

MSXE3317 soap api functions

Generated by Doxygen 1.7.1

Mon Oct 27 2014 09:47:51

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Remark: SOAP functions prototypes	1
2	Module Documentation	3
2.1	MSX-E3317 functions	3
2.2	Common functions	3
2.3	Common general functions	4
2.3.1	Function Documentation	5
2.3.1.1	MXCommon__GetModuleType	5
2.3.1.2	MXCommon__GetHostname	5
2.3.1.3	MXCommon__SetHostname	6
2.3.1.4	MXCommon__GetClientConnections	6
2.3.1.5	MXCommon__Sterror	6
2.3.1.6	MXCommon__Reboot	8
2.3.1.7	MXCommon__ResetAllIOFunctionalities	8
2.3.1.8	MXCommon__DataseverRestart	8
2.3.1.9	MXCommon__GetEthernetLinksStates	9
2.4	Common temperature functions	10
2.4.1	Detailed Description	10
2.4.2	Function Documentation	10
2.4.2.1	MXCommon__GetModuleTemperatureValueAndStatus	10
2.4.2.2	MXCommon__SetModuleTemperatureWarningLevels	11
2.5	Common hardware trigger functions	11
2.5.1	Function Documentation	12
2.5.1.1	MXCommon__SetHardwareTriggerFilterTime	12
2.5.1.2	MXCommon__GetHardwareTriggerFilterTime	12
2.5.1.3	MXCommon__GetHardwareTriggerState	13

2.6	Common security functions	13
2.6.1	Detailed Description	14
2.6.2	Function Documentation	15
2.6.2.1	MXCommon__SetCustomerKey	15
2.6.2.2	MXCommon__TestCustomerID	15
2.7	Common time functions	15
2.7.1	Detailed Description	16
2.7.2	Function Documentation	16
2.7.2.1	MXCommon__SetTime	16
2.7.2.2	MXCommon__SysToHardwareClock	17
2.7.2.3	MXCommon__HardwareClockToSys	17
2.7.2.4	MXCommon__GetTime	17
2.7.2.5	MXCommon__GetUpTime	18
2.8	Common I/O auto configuration functions	18
2.8.1	Detailed Description	19
2.8.2	Function Documentation	19
2.8.2.1	MXCommon__GetAutoConfigurationFile	19
2.8.2.2	MXCommon__SetAutoConfigurationFile	19
2.8.2.3	MXCommon__StartAutoConfiguration	20
2.9	Common synchronisation timer functions	20
2.9.1	Function Documentation	21
2.9.1.1	MXCommon__InitAndStartSynchroTimer	21
2.9.1.2	MXCommon__StopAndReleaseSynchroTimer	22
2.10	Set/Backup/Restore general system configuration	22
2.10.1	Detailed Description	22
2.10.2	Function Documentation	23
2.10.2.1	MXCommon__GetConfigurationBackupFile	23
2.10.2.2	MXCommon__ApplyConfigurationBackupFile	23
2.10.2.3	MXCommon__ChangePassword	24
2.11	System state management	24
2.11.1	Detailed Description	25
2.11.2	Function Documentation	25
2.11.2.1	MXCommon__GetSubSystemState	25
2.11.2.2	MXCommon__GetSubsystemIDFromName	26
2.11.2.3	MXCommon__GetStateIDFromName	26
2.11.2.4	MXCommon__GetSubsystemNameFromID	27

2.11.2.5	MXCommon__GetStateNameFromID	27
2.12	Customer option management	27
2.12.1	Function Documentation	28
2.12.1.1	MXCommon__GetOptionInformation	28
2.13	Synchronisation management	28
2.13.1	Function Documentation	28
2.13.1.1	MXCommon__SetToMaster	28
2.13.1.2	MXCommon__GetSynchronizationStatus	29
2.14	MSX-E3317 Acquisition functions	29
2.15	MSX-E3317 Acquisition information functions	30
2.15.1	Function Documentation	30
2.15.1.1	MSXExxxx__AcquisitionGetNumberOfChannels	30
2.15.1.2	MSXExxxx__AcquisitionGetChannelInfo	31
2.16	MSX-E3317 Autorefresh functions	31
2.16.1	Detailed Description	32
2.16.2	Function Documentation	33
2.16.2.1	MSXExxxx__AcquisitionAutoRefreshInitAndStart	33
2.16.2.2	MSXExxxx__AcquisitionAutoRefreshGetValues	35
2.16.2.3	MSXExxxx__AcquisitionAutoRefreshStopAndRelease	35
2.17	MSX-E3317 Sequence functions	36
2.17.1	Detailed Description	36
2.17.2	Function Documentation	37
2.17.2.1	MSXExxxx__AcquisitionSequenceInitAndStart	37
2.17.2.2	MSXExxxx__AcquisitionSequenceStopAndRelease	39
2.18	MSX-E3317 Pressure functions	40
2.19	MSX-E3317 Pressure initialisation/information functions	40
2.19.1	Function Documentation	41
2.19.1.1	MSXExxxx__PressureGetNumberOfChannels	41
2.19.1.2	MSXExxxx__PressureSetChannelConfiguration	41
2.19.1.3	MSXExxxx__PressureSetSamplingRate	42
2.19.1.4	MSXExxxx__PressureGetConfiguration	43
2.20	MSX-E3317 Digital I/O functions	43
2.21	MSX-E3317 Digital I/O information, configuration functions	44
2.21.1	Function Documentation	44
2.21.1.1	MSXExxxx__DigitalIOGetNumberOfChannels	44
2.21.1.2	MSXExxxx__DigitalIOGetNumberOfPorts	45

2.21.1.3	MSXExxxx__DigitalIOGetPortAvailableDirections	45
2.21.1.4	MSXExxxx__DigitalIOSetPortDirections	46
2.21.1.5	MSXExxxx__DigitalIOGetPortDirections	47
2.22	MSX-E3317 Digital I/O filter functions	47
2.22.1	Function Documentation	48
2.22.1.1	MSXExxxx__DigitalIOSetInputsFilterTime	48
2.22.1.2	MSXExxxx__DigitalIOEnableDisableInputsFilter	48
2.22.1.3	MSXExxxx__DigitalIOGetInputsFilterConfiguration	49
2.23	MSX-E3317 Digital I/O diagnostic functions	49
2.23.1	Function Documentation	50
2.23.1.1	MSXExxxx__DigitalIOTestOutputsShortCircuit	50
2.23.1.2	MSXExxxx__DigitalIORearmOutputsShortCircuit	51
2.23.1.3	MSXExxxx__DigitalIOTestOutputsPowerSupply	51
2.24	MSX-E3317 Digital I/O read/write functions	52
2.24.1	Function Documentation	52
2.24.1.1	MSXExxxx__DigitalIOReadChannel	52
2.24.1.2	MSXExxxx__DigitalIOReadPort	53
2.24.1.3	MSXExxxx__DigitalIOWriteChannel	53
2.24.1.4	MSXExxxx__DigitalIOWritePort	54
2.25	MSX-E3317 Sine/Cosine functions	54
2.26	MSX-E3317 Sine/Cosine initialisation/information functions	55
2.26.1	Function Documentation	55
2.26.1.1	MSXExxxx__SinCosGetNumberOfChannels	55
2.26.1.2	MSXExxxx__SinCosInit	56
2.26.1.3	MSXExxxx__SinCosRelease	57
2.26.1.4	MSXExxxx__SinCosGetConfiguration	58
2.27	MSX-E3317 Sine/Cosine read and clear functions	59
2.27.1	Function Documentation	59
2.27.1.1	MSXExxxx__SinCosRead	59
2.27.1.2	MSXExxxx__SinCosClear	60
2.27.1.3	MSXExxxx__SinCosClearError	60
2.28	MSX-E3317 Sine/Cosine Index functions	61
2.28.1	Function Documentation	61
2.28.1.1	MSXExxxx__SinCosInitIndex	61
2.28.1.2	MSXExxxx__SinCosReleaseIndex	62
2.29	MSX-E3317 Sine/Cosine Compare functions	63

2.29.1	Function Documentation	63
2.29.1.1	MSXExxxx__SinCosInitAndEnableCompareLogic	63
2.29.1.2	MSXExxxx__SinCosDisableAndReleaseCompareLogic	64
3	Data Structure Documentation	67
3.1	ByteArray Struct Reference	67
3.1.1	Field Documentation	67
3.1.1.1	__ptr	67
3.1.1.2	__size	67
3.1.1.3	__offset	67
3.2	DefaultResponse Struct Reference	67
3.2.1	Field Documentation	68
3.2.1.1	iReturnValue	68
3.2.1.2	syserrno	68
3.3	MSXExxxx__AcquisitionAutoRefreshGetValuesResponse Struct Reference	68
3.3.1	Field Documentation	69
3.3.1.1	sResponse	69
3.3.1.2	ulTimeStampLow	69
3.3.1.3	ulTimeStampHigh	69
3.3.1.4	ulCounterValue	69
3.3.1.5	dValue	69
3.4	MSXExxxx__AcquisitionGetChannelInfoResponse Struct Reference	69
3.4.1	Field Documentation	69
3.4.1.1	sResponse	69
3.4.1.2	ulType	69
3.4.1.3	ulHwPosition	70
3.4.1.4	ulChannelIndex	70
3.5	MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam Struct Reference	70
3.5.1	Field Documentation	70
3.5.1.1	ulChannelList	70
3.6	MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse Struct Reference	70
3.6.1	Field Documentation	71
3.6.1.1	sResponse	71
3.6.1.2	ulFilterTime	71
3.6.1.3	ulFilter	71
3.7	MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse Struct Reference	71
3.7.1	Field Documentation	71

3.7.1.1	sResponse	71
3.7.1.2	ulInputs	71
3.7.1.3	ulOutputs	72
3.8	MSXExxxx__FileResponse Struct Reference	72
3.8.1	Field Documentation	72
3.8.1.1	sResponse	72
3.8.1.2	sArray	72
3.8.1.3	ulEOF	72
3.9	MSXExxxx__PressureGetConfigurationResponse Struct Reference	72
3.9.1	Field Documentation	73
3.9.1.1	sResponse	73
3.9.1.2	dSensorSensibility	73
3.9.1.3	dSensorOffset	73
3.9.1.4	ulBaseSamplingRate	73
3.10	MSXExxxx__Response Struct Reference	73
3.10.1	Field Documentation	73
3.10.1.1	iReturnValue	73
3.10.1.2	syserrno	73
3.11	MSXExxxx__SinCosGetConfigurationResponse Struct Reference	73
3.11.1	Field Documentation	75
3.11.1.1	sResponse	75
3.11.1.2	ulInitialisationState	75
3.11.1.3	dSignalPeriod	75
3.11.1.4	ulResolution	75
3.11.1.5	ulIndexState	75
3.11.1.6	ulIndexEdge	75
3.11.1.7	ulIndexAction	75
3.11.1.8	ulCompareLogicState	75
3.11.1.9	dCompareLogicValue	75
3.11.1.10	ulCompareLogicValueFormat	75
3.11.1.11	ulCompareLogicMode	75
3.11.1.12	ulCompareLogicSynchro	75
3.11.1.13	ulInfo01	75
3.11.1.14	ulInfo02	75
3.12	MSXExxxx__SinCosInitResponse Struct Reference	75
3.12.1	Field Documentation	76

3.12.1.1	sResponse	76
3.12.1.2	ulMaxInputFrequency	76
3.12.1.3	ulInfo01	76
3.12.1.4	ulInfo02	76
3.13	MSXExxxx__SinCosReadResponse Struct Reference	76
3.13.1	Field Documentation	76
3.13.1.1	sResponse	76
3.13.1.2	dValue	76
3.13.1.3	ulValue	76
3.13.1.4	ulMeasureError	76
3.13.1.5	ulInfo01	77
3.13.1.6	ulInfo02	77
3.14	MSXExxxx__unsignedLongResponse Struct Reference	77
3.14.1	Field Documentation	77
3.14.1.1	sResponse	77
3.14.1.2	ulValue	77
3.15	MSXExxxx__unsignedLongTimeStampResponse Struct Reference	77
3.15.1	Field Documentation	78
3.15.1.1	sResponse	78
3.15.1.2	ulValue	78
3.15.1.3	ulTimeStampLow	78
3.15.1.4	ulTimeStampHigh	78
3.16	MXCommon__ByteArrayResponse Struct Reference	78
3.16.1	Field Documentation	78
3.16.1.1	sResponse	78
3.16.1.2	sArray	78
3.17	MXCommon__FileResponse Struct Reference	78
3.17.1	Field Documentation	79
3.17.1.1	sResponse	79
3.17.1.2	sArray	79
3.17.1.3	ulEOF	79
3.18	MXCommon__GetAutoConfigurationFileResponse Struct Reference	79
3.18.1	Field Documentation	79
3.18.1.1	sResponse	79
3.18.1.2	bArray	79
3.18.1.3	ulEOF	79

3.19	MXCommon__GetEthernetLinksStatesResponse Struct Reference	79
3.19.1	Field Documentation	80
3.19.1.1	sResponse	80
3.19.1.2	sPort0	80
3.19.1.3	sPort1	80
3.20	MXCommon__GetHardwareTriggerFilterTimeResponse Struct Reference	80
3.20.1	Field Documentation	80
3.20.1.1	sResponse	80
3.20.1.2	ulFilterTime	80
3.20.1.3	ulInfo01	80
3.20.1.4	ulInfo02	80
3.21	MXCommon__GetHardwareTriggerStateResponse Struct Reference	80
3.21.1	Field Documentation	81
3.21.1.1	sResponse	81
3.21.1.2	ulState	81
3.21.1.3	ulInfo01	81
3.21.1.4	ulInfo02	81
3.22	MXCommon__GetModuleTemperatureValueAndStatusResponse Struct Reference	81
3.22.1	Field Documentation	82
3.22.1.1	sResponse	82
3.22.1.2	dTemperatureValue	82
3.22.1.3	ulTemperatureStatus	82
3.22.1.4	ulInfo	82
3.23	MXCommon__GetTimeResponse Struct Reference	82
3.23.1	Field Documentation	82
3.23.1.1	sResponse	82
3.23.1.2	ulLowTime	82
3.23.1.3	ulHighTime	82
3.24	MXCommon__GetUpTimeResponse Struct Reference	82
3.24.1	Field Documentation	83
3.24.1.1	sResponse	83
3.24.1.2	ulUpTime	83
3.25	MXCommon__Response Struct Reference	83
3.25.1	Field Documentation	83
3.25.1.1	iReturnValue	83
3.25.1.2	syserrno	83

3.26	MXCommon__TestCustomerIDResponse Struct Reference	83
3.26.1	Field Documentation	84
3.26.1.1	sResponse	84
3.26.1.2	bValueArray	84
3.26.1.3	bCryptedValueArray	84
3.27	MXCommon__unsignedLongResponse Struct Reference	84
3.27.1	Field Documentation	84
3.27.1.1	sResponse	84
3.27.1.2	ulValue	84
3.28	sGetEthernetLinksStatesPort Struct Reference	84
3.28.1	Field Documentation	85
3.28.1.1	ulState	85
3.28.1.2	ulSpeed	85
3.28.1.3	ulDuplex	85
3.28.1.4	ulInfo1	85
3.28.1.5	ulInfo2	85
3.29	UnsignedLongArray Struct Reference	85
3.29.1	Field Documentation	85
3.29.1.1	__ptr	85
3.29.1.2	__size	85
3.29.1.3	__offset	85
3.30	UnsignedShortArray Struct Reference	85
3.30.1	Field Documentation	86
3.30.1.1	__ptr	86
3.30.1.2	__size	86
3.30.1.3	__offset	86
3.31	xsd__base64Binary Struct Reference	86
3.31.1	Field Documentation	86
3.31.1.1	__ptr	86
3.31.1.2	__size	86
4	File Documentation	87
4.1	MSXE3317_public_doc.h File Reference	87
4.1.1	Typedef Documentation	95
4.1.1.1	xsd__string	95
4.1.1.2	xsd__char	95
4.1.1.3	xsd__float	95

4.1.1.4	xsd__double	95
4.1.1.5	xsd__int	95
4.1.1.6	xsd__long	95
4.1.1.7	xsd__unsignedByte	95
4.1.1.8	xsd__unsignedInt	95
4.1.1.9	xsd__unsignedShort	95
4.1.1.10	xsd__unsignedLong	95
4.1.2	Function Documentation	95
4.1.2.1	MXCommon__GetModuleType	95
4.1.2.2	MXCommon__GetHostname	96
4.1.2.3	MXCommon__SetHostname	96
4.1.2.4	MXCommon__GetClientConnections	96
4.1.2.5	MXCommon__Strerror	97
4.1.2.6	MXCommon__Reboot	98
4.1.2.7	MXCommon__ResetAllIOFunctionalities	98
4.1.2.8	MXCommon__DataseverRestart	99
4.1.2.9	MXCommon__GetEthernetLinksStates	99
4.1.2.10	MXCommon__GetModuleTemperatureValueAndStatus	100
4.1.2.11	MXCommon__SetModuleTemperatureWarningLevels	101
4.1.2.12	MXCommon__SetHardwareTriggerFilterTime	101
4.1.2.13	MXCommon__GetHardwareTriggerFilterTime	102
4.1.2.14	MXCommon__GetHardwareTriggerState	102
4.1.2.15	MXCommon__SetCustomerKey	103
4.1.2.16	MXCommon__TestCustomerID	103
4.1.2.17	MXCommon__SetTime	104
4.1.2.18	MXCommon__SysToHardwareClock	104
4.1.2.19	MXCommon__HardwareClockToSys	104
4.1.2.20	MXCommon__GetTime	105
4.1.2.21	MXCommon__GetUpTime	105
4.1.2.22	MXCommon__GetAutoConfigurationFile	106
4.1.2.23	MXCommon__SetAutoConfigurationFile	106
4.1.2.24	MXCommon__StartAutoConfiguration	106
4.1.2.25	MXCommon__InitAndStartSynchroTimer	107
4.1.2.26	MXCommon__StopAndReleaseSynchroTimer	108
4.1.2.27	MXCommon__GetConfigurationBackupFile	108
4.1.2.28	MXCommon__ApplyConfigurationBackupFile	109

4.1.2.29	MXCommon__ChangePassword	109
4.1.2.30	MXCommon__GetSubSystemState	110
4.1.2.31	MXCommon__GetSubsystemIDFromName	110
4.1.2.32	MXCommon__GetStateIDFromName	111
4.1.2.33	MXCommon__GetSubsystemNameFromID	111
4.1.2.34	MXCommon__GetStateNameFromID	112
4.1.2.35	MXCommon__GetOptionInformation	112
4.1.2.36	MXCommon__SetToMaster	112
4.1.2.37	MXCommon__GetSynchronizationStatus	113
4.1.2.38	MSXExxxx__AcquisitionGetNumberOfChannels	113
4.1.2.39	MSXExxxx__AcquisitionGetChannelInfo	114
4.1.2.40	MSXExxxx__AcquisitionAutoRefreshInitAndStart	114
4.1.2.41	MSXExxxx__AcquisitionAutoRefreshGetValues	116
4.1.2.42	MSXExxxx__AcquisitionAutoRefreshStopAndRelease	117
4.1.2.43	MSXExxxx__AcquisitionSequenceInitAndStart	118
4.1.2.44	MSXExxxx__AcquisitionSequenceStopAndRelease	120
4.1.2.45	MSXExxxx__PressureGetNumberOfChannels	120
4.1.2.46	MSXExxxx__PressureSetChannelConfiguration	121
4.1.2.47	MSXExxxx__PressureSetSamplingRate	121
4.1.2.48	MSXExxxx__PressureGetConfiguration	122
4.1.2.49	MSXExxxx__DigitalIOGetNumberOfChannels	123
4.1.2.50	MSXExxxx__DigitalIOGetNumberOfPorts	124
4.1.2.51	MSXExxxx__DigitalIOGetPortAvailableDirections	124
4.1.2.52	MSXExxxx__DigitalIOSetPortDirections	125
4.1.2.53	MSXExxxx__DigitalIOGetPortDirections	125
4.1.2.54	MSXExxxx__DigitalIOSetInputsFilterTime	126
4.1.2.55	MSXExxxx__DigitalIOEnableDisableInputsFilter	127
4.1.2.56	MSXExxxx__DigitalIOGetInputsFilterConfiguration	127
4.1.2.57	MSXExxxx__DigitalIOTestOutputsShortCircuit	128
4.1.2.58	MSXExxxx__DigitalIORearmOutputsShortCircuit	128
4.1.2.59	MSXExxxx__DigitalIOTestOutputsPowerSupply	129
4.1.2.60	MSXExxxx__DigitalIOReadChannel	130
4.1.2.61	MSXExxxx__DigitalIOReadPort	130
4.1.2.62	MSXExxxx__DigitalIOWriteChannel	131
4.1.2.63	MSXExxxx__DigitalIOWritePort	131
4.1.2.64	MSXExxxx__SinCosGetNumberOfChannels	132

4.1.2.65	MSXExxxx__SinCosInit	132
4.1.2.66	MSXExxxx__SinCosRelease	134
4.1.2.67	MSXExxxx__SinCosGetConfiguration	134
4.1.2.68	MSXExxxx__SinCosRead	135
4.1.2.69	MSXExxxx__SinCosClear	136
4.1.2.70	MSXExxxx__SinCosClearError	136
4.1.2.71	MSXExxxx__SinCosInitIndex	137
4.1.2.72	MSXExxxx__SinCosReleaseIndex	138
4.1.2.73	MSXExxxx__SinCosInitAndEnableCompareLogic	138
4.1.2.74	MSXExxxx__SinCosDisableAndReleaseCompareLogic	139

Chapter 1

Introduction

MainRevision:

1.1 Introduction

This documentation describes the SOAP functions and gives software hints to work with the MSX-E systems. Following documentations can be found under **Modules**.

SOAP means Simple Object Access Protocol. This protocol enables to use the MSX-E software functions over Ethernet. It is providing **Web Services** that can easily be consumed in many programming languages like C, C++, C#, VB.Net... With the SOAP functions, all functionalities of the MSX-E system can be managed / configured / monitored.

1.2 Remark: SOAP functions prototypes

In some programming languages, SOAP functions names and parameters could be different as those described in this documentation. Please see to `software_hints`

Chapter 2

Module Documentation

2.1 MSX-E3317 functions

Modules

- [MSX-E3317 Acquisition functions](#)
Contain the acquisition information and initialisation functions.
- [MSX-E3317 Pressure functions](#)
Contain the Pressure initialisation and diagnostic functions.
- [MSX-E3317 Digital I/O functions](#)
Contain the digital I/O functions.
- [MSX-E3317 Sine/Cosine functions](#)
Contain the Sine/Cosine functions.

2.2 Common functions

Modules

- [Common general functions](#)
Various utility functions, mainly to identify a remote system.
- [Common temperature functions](#)
These functions deals with the internal temperature sub-system.
- [Common hardware trigger functions](#)
These functions allow to set and request the current value of the hardware trigger.
- [Common security functions](#)
The "customer key" feature may for instance be used by a customer to be sure that his application communicates only with certified MSX-E modules.

- [Common time functions](#)

A MSX-E module provides a "system clock" that may be in the simplest case set by the function [MXCommon__SetTime\(\)](#).

- [Common I/O auto configuration functions](#)

On the web site of some MSX-E module, there is the possibility to define an auto-configuration and auto start of the I/O.

- [Common synchronisation timer functions](#)

When modules are linked through a "synchronisation bus", the master can run a timer that generate a "synchro signal" on the slaves when overrun.

- [Set/Backup/Restore general system configuration](#)

Distinct of the I/O auto-configuration/auto-start functionality, these functions allows to manipulate the general system configuration.

- [System state management](#)

Every MSX-E modules are composed of several sub-systems that work together to provide the system functionalities.

- [Customer option management](#)

Enable to get informations about the options of the system.

- [Synchronisation management](#)

Manage the synchronisation state of the system.

2.3 Common general functions

Various utility functions, mainly to identify a remote system.

Functions

- [int MXCommon__GetModuleType](#) (void ___, struct [MXCommon__ByteArrayResponse](#) *Response)

This function return the type of the MSX-E Module.

- [int MXCommon__GetHostname](#) (void ___, struct [MXCommon__ByteArrayResponse](#) *Response)

This function return the hostname of the MSX-E Module.

- [int MXCommon__SetHostname](#) (struct [xsd__base64Binary](#) *bHostname, struct [MXCommon__Response](#) *Response)

This function allows to set the hostname of the MSX-E Module.

- [int MXCommon__GetClientConnections](#) (void ___, struct [MXCommon__ByteArrayResponse](#) *Response)

This function return the client connection list.

- `int MXCommon__Strerror (xsd__int errnum, struct MXCommon__ByteArrayResponse *Response)`
Call the libc `strerror()` on the remote device (actually this is a call to `strerror_r()`).
- `int MXCommon__Reboot (void *_ , struct MXCommon__Response *Response)`
Ask the MSX-E module to reboot.
- `int MXCommon__ResetAllIOFunctionalities (xsd__unsignedLong ulOption, struct MXCommon__Response *Response)`
Reset the I/O functionalities of the MSX-E system.
- `int MXCommon__DataserverRestart (xsd__unsignedLong ulAction, xsd__unsignedLong ulOption, struct MXCommon__Response *Response)`
Restart the data-server service.
- `int MXCommon__GetEthernetLinksStates (void *_ , struct MXCommon__GetEthernetLinksStatesResponse *Response)`
Get MSX-E Ethernet links states.

2.3.1 Function Documentation

2.3.1.1 `int MXCommon__GetModuleType (void * _ , struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] `_` : no input parameter
- [out] **Response** • `sArray` : Module type string
 • `sResponse` Composed of `iReturnValue` and `syserrno`

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.3.1.2 `int MXCommon__GetHostname (void * _ , struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] `_` : no input parameter
- [out] **Response** • `sArray` : Hostname of the module
 • `iReturnValue` : Return value
 – 0 : success
 – -1: system error (see `syserrno`)
 • `syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.3.1.3 `int MXCommon__SetHostname (struct xsd__base64Binary * bHostname, struct MXCommon__Response * Response)`

Parameters

- [in] *bHostname* : Hostname
- [out] *Response*
 - *iReturnValue* : Return value
 - 0 : success
 - -1: system error (see *syserrno*)
 - *syserrno* : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.3.1.4 `int MXCommon__GetClientConnections (void * __, struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] *_* : no input parameter
- [out] *Response*
 - *sArray* : string containing the list of connected clients.
 - *sResponse* Composed of *iReturnValue* and *syserrno*

The *sArray* string is of the form IP-Address:first connection-second connection---- IP-Address:first connection-second connection----

Sample: 172.16.3.43:8989-5555 172.16.3.200:8989

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.3.1.5 `int MXCommon__Strerror (xsd__int errnum, struct MXCommon__ByteArrayResponse * Response)`

Usually SOAP functions return this value in a variable named *syserror*, which is meaningful only when the function return value, usually called *iReturnValue*, indicate an error (that is, have a value of -1 or -100, depending of the case).

Parameters

- [in] *errnum* : Error number
- [out] *Response*
 - *sArray* : See the description below.
 - *sResponse.iReturnValue* : Return value

- 0 : success
- -1: system error (see syserrno).
- `sResponse.syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

STRERROR(3)
STRERROR(3)

Linux Programmer's Manual

NAME

`strerror`, `strerror_r` - return string describing error code

SYNOPSIS

```
#include <string.h>
```

```
char *strerror(int errnum);
```

```
#define _XOPEN_SOURCE 600
#include <string.h>
```

```
int strerror_r(int errnum, char *buf, size_t n);
```

DESCRIPTION

The `strerror()` function returns a string describing the error code passed in the argument `errnum`, possibly using the `LC_MESSAGES` part of the current locale to select the appropriate language. This string must not be modified by the application, but may be modified by a subsequent call to `perror()` or `strerror()`. No library function will modify this string.

The `strerror_r()` function is similar to `strerror()`, but is thread safe. It returns the string in the user-supplied buffer `buf` of length `n`.

RETURN VALUE

The `strerror()` function returns the appropriate error description string, or an unknown error message if the error code is unknown. The value of `errno` is not changed for a successful call, and is set to a non-zero value upon error. The `strerror_r()` function returns 0 on success and -1 on failure, setting `errno`.

ERRORS

EINVAL The value of `errnum` is not a valid error number.

ERANGE Insufficient storage was supplied to contain the error description string.

CONFORMING TO

SVID 3, POSIX, 4.3BSD, ISO/IEC 9899:1990 (C89). `strerror_r()` with prototype as given above is specified by SUSv3, and was in use under Digital Unix and HP Unix. An incompatible function, with prototype

```
char *strerror_r(int errnum, char *buf, size_t n);
```

is a GNU extension used by glibc (since 2.0), and must be regarded as obsolete in view of SUSv3.

The GNU version may, but need not, use the user-supplied buffer. If it does, the result may be truncated in case the supplied buffer is too small. The result is always NUL-terminated.

SEE ALSO

`errno(3)`, `perror(3)`, `strsignal(3)`

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.3.1.6 int MXCommon__Reboot (void * __, struct MXCommon__Response * *Response*)

Parameters

- [in] *__* : no input parameter
- [out] *Response* • iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
 - syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.3.1.7 int MXCommon__ResetAllIOFunctionalities (xsd__unsignedLong *ulOption*, struct MXCommon__Response * *Response*)

The behavior of the function depends on the MSX-E system that is used.

On MSX-E3511: Stop the watchdogs and stop the generators
 On MSX-E3601: Stop the sequence acquisition and stop the calibration
 On MSX-E3701: Stop the acquisition

Parameters

- [in] *ulOption* Reserved. Set to 0
- [out] *Response* *iReturnValue*
- **0** The remote function performed OK
 - **-1** Internal system error occurred. See value of syserrno
 - **-100** Function not supported by the system
- syserrno* system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.3.1.8 int MXCommon__DataseverRestart (xsd__unsignedLong *ulAction*, xsd__unsignedLong *ulOption*, struct MXCommon__Response * *Response*)

Parameters

- [in] *ulAction* : action
- 0: normal restart
 - 1: with cache file reset
 - 2: with cache file deletion
- [in] *ulOption* : Reserved
- [out] *Response* • iReturnValue : Return value

- 0 : success
- -1: system error (see syserrno)
- syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

Note

(revision>6386) Depending on the system type, can be used to restart the data-recv service as well. In this case, parameter action is ignored.

2.3.1.9 int MXCommon__GetEthernetLinksStates (void * __, struct MXCommon__GetEthernetLinksStatesResponse * *Response*)

Parameters

[in] *__* : no input parameter

[out] *Response* Structure that contains the MSX-E Ethernet links states and errors:

sResponse.iReturn Value

- **0** The remote function performed OK
- **-1** System error occurred
- **-2** Fail to get Ethernet links states
- **-100** Internal system error occurred. See value of syserrno

sResponse.syserrno system error code (the value of the libc "errno" code)

sPort0: Fisrt port informations

- **ulState**
 - **0** Link down
 - **1** Link up
- **ulSpeed**
 - **10** 10 Mb/s
 - **100** 100 Mb/s
- **ulDuplex**
 - **0** Half duplex
 - **1** Full duplex
- **ulInfo1** Reserverd
- **ulInfo2** Reserverd

sPort1: Second port informations

- **ulState**
 - **0** Link down
 - **1** Link up
- **ulSpeed**
 - **10** 10 Mb/s
 - **100** 100 Mb/s
- **ulDuplex**

- 0 Half duplex
- 1 Full duplex
- **ulInfo1** Reserved
- **ulInfo2** Reserved

Return values

0 SOAP_OK

Others See SOAP error

2.4 Common temperature functions

These functions deals with the internal temperature sub-system.

Data Structures

- struct [MXCommon__GetModuleTemperatureValueAndStatusResponse](#)

Functions

- int [MXCommon__GetModuleTemperatureValueAndStatus](#) (xsd__unsignedLong ulOption, struct [MXCommon__GetModuleTemperatureValueAndStatusResponse](#) *Response)

Read the temperature on the module.

- int [MXCommon__SetModuleTemperatureWarningLevels](#) (xsd__double dMinimalWarningLevel, xsd__double dMaximalWarningLevel, xsd__unsignedLong ulOption, struct [MXCommon__Response](#) *Response)

Set the temperature warning level on the module.

2.4.1 Detailed Description

The role of this sub-system is to monitor the internal temperature of a module and issue a warning if it is below or above a threshold. If the internal temperature reaches a domain where the system is endangered, it switches automatically in a degraded working mode.

2.4.2 Function Documentation

2.4.2.1 int [MXCommon__GetModuleTemperatureValueAndStatus](#) (xsd__unsignedLong *ulOption*, struct [MXCommon__GetModuleTemperatureValueAndStatusResponse](#) * *Response*)

Parameters

[in] *ulOption* : Reserved

[out] *Response* • sResponse.iReturnValue : Return value

- 0 : success
- -1: system error (see syserrno)

- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - dValue : Temperature value in Degree Celsius
- ulTemperatureStatus : Temperature Status :
 - TEMPERATURE_INITIAL = 0 : Temperature not ready
 - TEMPERATURE_TOLOW = 1 : Temperature too low !
 - TEMPERATURE_LOW = 2 : Temperature under the min warning value
 - TEMPERATURE_NOMINAL = 3 : Temperature in the nominal range
 - TEMPERATURE_HIGH = 4 : Temperature over the max warning value
 - TEMPERATURE_TOOHIGH = 5 : Temperature too high !
- ulInfo : Reserved

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.4.2.2 `int MXCommon__SetModuleTemperatureWarningLevels (xsd__double dMinimalWarningLevel, xsd__double dMaximalWarningLevel, xsd__unsignedLong ulOption, struct MXCommon__Response * Response)`

Parameters

- [in] *dMinimalWarningLevel* : Minimal temperature warning level in Degree : 5 to 60 Degree Celsius
- [in] *dMaximalWarningLevel* : Maximal temperature warning level in Degree : 5 to 60 Degree Celsius
- [in] *ulOption* : Reserved
- [out] *Response* • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.5 Common hardware trigger functions

These functions allow to set and request the current value of the hardware trigger.

Data Structures

- struct [MXCommon__GetHardwareTriggerFilterTimeResponse](#)
- struct [MXCommon__GetHardwareTriggerStateResponse](#)

Functions

- `int MXCommon__SetHardwareTriggerFilterTime (xsd__unsignedLong ulFilterTime, xsd__unsignedLong ulOption, struct MXCommon__Response *Response)`
Sets the filter time for the hardware trigger input in steps of 250 ns (max value: 65535).
- `int MXCommon__GetHardwareTriggerFilterTime (xsd__unsignedLong ulOption, struct MXCommon__GetHardwareTriggerFilterTimeResponse *Response)`
Get the filter time for the hardware trigger input.
- `int MXCommon__GetHardwareTriggerState (xsd__unsignedLong ulOption, struct MXCommon__GetHardwareTriggerStateResponse *Response)`
Get the hardware trigger state after the filter.

2.5.1 Function Documentation

2.5.1.1 `int MXCommon__SetHardwareTriggerFilterTime (xsd__unsignedLong ulFilterTime, xsd__unsignedLong ulOption, struct MXCommon__Response * Response)`

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

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

Parameters

- [in] ***ulFilterTime*** Filter time for the hardware trigger input in steps of 250ns (max value : 65535).
- **0**: Disable the filter
 - **1**: Sets the filter time to 250 ns
 - **2**: Sets the filter time to 500 ns
 - ...
 - **65535**: Sets the filter time to 16 ms
- [in] ***ulOption*** Reserved. Set to 0
- [out] ***Response*** Response of the system
- ***sResponse.iReturnValue***
 - **0**: The remote function performed OK
 - **-1**: Internal system error occurred. See value of syserrno
 - ***sResponse.syserrno*** system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.5.1.2 `int MXCommon__GetHardwareTriggerFilterTime (xsd__unsignedLong ulOption, struct MXCommon__GetHardwareTriggerFilterTimeResponse * Response)`

Get 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

- [in] *ulOption* Reserved. Set to 0
- [out] *Response* Response of the system
- *ulFilterTime* filter time for the hardware trigger input
 - 0: filter disabled
 - 1: filter of 250ns
 - 2: filter of 500ns
 - ...
 - 65535: filter of 16ms
 - *sResponse.iReturnValue*
 - 0: The remote function performed OK
 - -1: Internal system error occurred. See value of syserrno
 - *sResponse.syserrno* system error code (the value of the libc "errno" code)

Return values

- 0 SOAP_OK
- Others* See SOAP error

2.5.1.3 `int MXCommon__GetHardwareTriggerState (xsd__unsignedLong ulOption, struct MXCommon__GetHardwareTriggerStateResponse * Response)`

Parameters

- [in] *ulOption* : Reserved
- [out] *Response* • *ulState* : Hardware trigger input state.
- 0: Hardware trigger input is low
 - 1: Hardware trigger input is high.
 - *sResponse.iReturnValue* : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - *sResponse.syserrno* : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

- SOAP_OK* SOAP call success
- otherwise* SOAP protocol error

2.6 Common security functions

The "customer key" feature may for instance be used by a customer to be sure that his application communicates only with certified MSX-E modules.

Data Structures

- struct [MXCommon__TestCustomerIDResponse](#)

Functions

- int [MXCommon__SetCustomerKey](#) (struct [xsd__base64Binary](#) *bKey, struct [xsd__base64Binary](#) *bPublicKey, struct [MXCommon__Response](#) *Response)

Set the Customer key.

- int [MXCommon__TestCustomerID](#) (void *_ , struct [MXCommon__TestCustomerIDResponse](#) *Response)

Test the Customer ID (if the module has the right customer Key).

2.6.1 Detailed Description

A "customer key" consists of two strings of data stored on the certified MSX-E module, to be used by the function [MXCommon__TestCustomerID\(\)](#) to encrypt data.

These strings can not be read back. They are supposed to be kept secret by the user of this functionality.

To test if the MSX-E module you use is certified, you can request the MSX-E module to provide a set of randomly generated data and the result of the encryption (through the use of the stored "customer key") of the same data. Then your application must encrypt the delivered random data with its own "customer key" and compare it with the encrypted data delivered by the MSX-E module.

If the results are matching, the MSX-E module is certified for this application.

Detailed presentation of operations:

The user generates and stores on the module two keys (thanks to the software function : [MXCommon__SetCustomerKey\(\)](#)). This needs only to be done once:

- A public Key K1 (16 Bytes)
- A private Key K2 (32 Bytes)

When requested (with the software function : [MXCommon__TestCustomerID\(\)](#)), the module generates a 16 bytes random value and do an encryption of this value using the two saved keys and the AES algorithm (Rijndael).

The user receives then two arrays of 16 bytes :

- one with a random value [A]
- the second with encrypted random value [B]

[B]=AES([A], K1, K2)

The user performs then the same computation from [A],K1,K2 and compares his result with [B]. If it is the same, it means that the module he is using was already configured with the correct identification token.

The security of the method comes from that even knowing [A] and [B] no one can deduce K1 and K2 back in practical times. ADDI-DATA is not aware of a practical way to remotely retrieve the value of the key stored on a module.

It is the responsibility of the developer of the application to ensure that these tokens are suitably protected. The authorisation of the change of the "customer key" on the MSX-E module can be managed with the web interface.

The use of the "customer key" don't have an impact of the other functionalities of the MSX-E module.

2.6.2 Function Documentation

2.6.2.1 `int MXCommon__SetCustomerKey (struct xsd__base64Binary * bKey, struct xsd__base64Binary * bPublicKey, struct MXCommon__Response * Response)`

Parameters

- [in] *bKey* : Customer key (only writable on the module) [32 bytes containing a AES key]
- [in] *bPublicKey* : IV (Initialisation vector) for the AES cryptography [16 bytes containing a AES key]
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.6.2.2 `int MXCommon__TestCustomerID (void * _, struct MXCommon__TestCustomerIDResponse * Response)`

Parameters

- [in] _ : No Input
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).
 - bValueArray : non encrypted value array [16 bytes of random data]
 - bCryptedValueArray : Encrypted value array [16 bytes of the encrypted random data]

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.7 Common time functions

A MSX-E module provides a "system clock" that may be in the simplest case set by the function [MXCommon__SetTime\(\)](#).

Data Structures

- struct [MXCommon__GetTimeResponse](#)
- struct [MXCommon__GetUpTimeResponse](#)

Functions

- `int MXCommon__SetTime (xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, struct MXCommon__Response *Response)`
Set the time on the module.
- `int MXCommon__SysToHardwareClock (void *_ , struct MXCommon__Response *Response)`
Set the hardware clock (if present) to the current system time.
- `int MXCommon__HardwareClockToSys (void *_ , struct MXCommon__Response *Response)`
Set the system time from the hardware clock (if present).
- `int MXCommon__GetTime (void *_ , struct MXCommon__GetTimeResponse *Response)`
Get the time on the module.
- `int MXCommon__GetUpTime (void *_ , struct MXCommon__GetUpTimeResponse *Response)`
Ask the MSX-E module uptime (number of seconds since the last boot).

2.7.1 Detailed Description

If the module is configured to use NTP, the time received by the NTP server will have a greater priority. If the module is linked to another through a "synchronization bus" and is slave, then the time received from the master is the one taken into account.

Recent models also provide a "hardware clock", a component whose role is to track the time between reboots.

2.7.2 Function Documentation

2.7.2.1 `int MXCommon__SetTime (xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, struct MXCommon__Response * Response)`

Parameters

- [in] **ulLowTime** : Number of microseconds since the begin of the second
- [in] **ulHighTime** : Number of seconds since the Epoch (1st January,1970)
- [out] **Response**
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

Return values

- SOAP_OK** SOAP call success
- otherwise** SOAP protocol error

2.7.2.2 `int MXCommon__SysToHardwareClock (void * _, struct MXCommon__Response * Response)`

Parameters

- [in] _ No input parameter
- [out] **Response**
- sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

If this function fails, it means the module does not have a hardware RTC, or the hardware is not functional. Check the "hwclock" subsystem status.

2.7.2.3 `int MXCommon__HardwareClockToSys (void * _, struct MXCommon__Response * Response)`

When the hardware clock is present, the system time is automatically set to it when the module becomes master on the inter-module synchronisation bus.

Parameters

- [in] _ No input parameter
- [out] **Response**
- sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

If this function fails, it means the module does not have a hardware RTC, or the hardware is not functional. Check the "hwclock" subsystem status.

2.7.2.4 `int MXCommon__GetTime (void * _, struct MXCommon__GetTimeResponse * Response)`

Parameters

- [in] _ : No input parameter
- [out] **Response**
- sResponse.iReturnValue : Return value
 - 0 : success

- -1: system error (see syserrno)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).
- ulLowTime : Number of microseconds since the begin of the second
- ulHighTime : Number of seconds since the Epoch (1st January,1970)

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.7.2.5 int MXCommon__GetUpTime (void * _, struct MXCommon__GetUpTimeResponse * Response)

Parameters

- [in] _ : no input parameter
- [out] **Response** • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).
 - ulUpTime : Number of seconds since the last boot of the system.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.8 Common I/O auto configuration functions

On the web site of some MSX-E module, there is the possibility to define an auto-configuration and auto start of the I/O.

Data Structures

- struct [MXCommon__GetAutoConfigurationFileResponse](#)

Functions

- int [MXCommon__GetAutoConfigurationFile](#) (void *_, struct [MXCommon__GetAutoConfigurationFileResponse](#) *Response)
Get the auto configuration file of the module.
- int [MXCommon__SetAutoConfigurationFile](#) (struct [xsd__base64Binary](#) *ByteArrayInput, [xsd__unsignedLong](#) ulEOF, struct [MXCommon__Response](#) *Response)
Set the auto configuration file of the module.

- `int MXCommon__StartAutoConfiguration (void *__, struct MXCommon__ByteArrayResponse *Response)`
start/Restart the auto configuration

2.8.1 Detailed Description

- Auto-configuration means the system configures the I/O automatically at boot time.
- Auto-start means the system starts an acquisition automatically at boot time (this may no make sense for some systems). It implies auto-configuration.

This set of functions allows to:

- get the auto-configuration/start currently set on module, as a read-only binary file.
- set a auto-configuration/start on the module, using a previously saved file.
- start or restart the auto-configuration/start on the module, using the current configuration saved on the module.

2.8.2 Function Documentation

2.8.2.1 `int MXCommon__GetAutoConfigurationFile (void * ___, struct MXCommon__GetAutoConfigurationFileResponse * Response)`

Parameters

- [in] `__` : No input parameter
- [out] **Response** • `sResponse.iReturnValue` : Return value
- 0 : success
 - -1: system error (see `syserrno`)
 - -100 : Error of the read of the auto configuration file
 - `sResponse.syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).
 - `bArray` : Array of Bytes of the file
 - `ulEOF` : End of file flag

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.8.2.2 `int MXCommon__SetAutoConfigurationFile (struct xsd__base64Binary * ByteArrayInput, xsd__unsignedLong ulEOF, struct MXCommon__Response * Response)`

Parameters

- [in] **ByteArrayInput** : Array of Bytes of the file
- [in] **ulEOF** : End of file flag

- [out] **Response**
- sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.8.2.3 int MXCommon__StartAutoConfiguration (void * _, struct MXCommon__ByteArrayResponse * Response)

Parameters

- [in] _ : No input parameter
- [out] **Response**
- sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).
 - sArray : message returned by the auto configuration start

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.9 Common synchronisation timer functions

When modules are linked through a "synchronisation bus", the master can run a timer that generate a "synchro signal" on the slaves when overrun.

Functions

- int [MXCommon__InitAndStartSynchroTimer](#) (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulReloadValue, xsd__unsignedLong ulNbrOfCycle, xsd__unsignedLong ulGenerateTriggerMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct [MXCommon__Response](#) *Response)
Initialises and starts the synchronisation timer of the module (not already available on all module).
- int [MXCommon__StopAndReleaseSynchroTimer](#) (xsd__unsignedLong ulOption01, struct [MXCommon__Response](#) *Response)
start/Restart the synchronisation timer (not already available on all module)

2.9.1 Function Documentation

2.9.1.1 `int MXCommon__InitAndStartSynchroTimer (xsd_unsignedLong ulTimeBase, xsd_unsignedLong ulReloadValue, xsd_unsignedLong ulNbrOfCycle, xsd_unsignedLong ulGenerateTriggerMode, xsd_unsignedLong ulOption01, xsd_unsignedLong ulOption02, xsd_unsignedLong ulOption03, xsd_unsignedLong ulOption04, struct MXCommon__Response * Response)`

Parameters

- [in] **ulTimeBase** : Time base of the timer (0 for us, 1 for ms, 2 for s)
- [in] **ulReloadValue** : Timer reload value (0 to 0xFFFF), minimum reload time is 5 us
- [in] **ulNbrOfCycle** : Number of timer cycle
 - 0: continuous
 - > 0: defined number of cycle
- [in] **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
- [in] **ulOption01** : Define the source of the trigger
 - 0 : Trigger disabled
 - 1 : Enable the hardware digital input trigger
- [in] **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)
- [in] **ulOption03** : Define the number of trigger events before the action occur
 - 1 : all trigger event start the action
 - max value : 65535
- [in] **ulOption04** : Reserved
- [out] **Response**
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - -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
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.9.1.2 int MXCommon__StopAndReleaseSynchroTimer (xsd__unsignedLong ulOption01, struct MXCommon__Response * Response)

Parameters

- [in] *ulOption01* : Reserved
- [out] *Response* • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
 - -100: Start/Stop timer error
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#). May be ENOSYS : Function not implemented.

Return values

- SOAP_OK* SOAP call success
- otherwise* SOAP protocol error

2.10 Set/Backup/Restore general system configuration

Distinct of the I/O auto-configuration/auto-start functionality, these functions allows to manipulate the general system configuration.

Functions

- int [MXCommon__GetConfigurationBackupFile](#) (void *__, struct [MXCommon__FileResponse](#) *Response)
Download a configuration backup file from the module.
- int [MXCommon__ApplyConfigurationBackupFile](#) (struct [xsd__base64Binary](#) *ByteArrayInput, [xsd__unsignedLong](#) ulEOF, struct [MXCommon__Response](#) *Response)
Upload a new configuration on the module.
- int [MXCommon__ChangePassword](#) (struct [xsd__base64Binary](#) *PreviousUser, struct [xsd__base64Binary](#) *PreviousPassword, struct [xsd__base64Binary](#) *NewUser, struct [xsd__base64Binary](#) *NewPassword, struct [MXCommon__Response](#) *Response)
Set a new id/password.

2.10.1 Detailed Description

It includes the network configuration, and generally everything that can be set up through the web interface.

These functions have been included to ease the automation of module customisation. They may be disabled using the web interface, in "Security/Remote general system configuration authorisation/remote sysconf changes"

2.10.2 Function Documentation

2.10.2.1 `int MXCommon__GetConfigurationBackupFile (void * _, struct MXCommon__FileResponse * Response)`

Parameters

- [in] `_` : No input parameter
- [out] ***Response*** • `sResponse.iReturnValue` : Return value
- 0 : success
 - -1: system error (see `syserrno`) (see `syserrno`)
 - `sResponse.syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - `bArray` : Array of Bytes of the file
 - `ulEOF` : End of file flag

Return values

- SOAP_OK*** SOAP call success
- otherwise*** SOAP protocol error

This function is designed to be called repeatedly until no more data is available. At this point the flag `ulEOF` is set.

Below is an example in pseudo-C.

```
int dummy;
struct MXCommon__FileResponse Response;
while(1)
{
    if ( MXCommon__GetConfigurationBackupFile(&dummy, &Response) != SOAP_OK)
    {
        // handle soap error
    }
    if (Response.iReturnValue)
    {
        // handle remote error (Response.syserrno contains more information)
    }
    // do something with the data, for example save it in a file
    write(fd, Response.bArray.__ptr, Response.bArray.__size);
    // if this is the end of the file, quit the loop
    if(Response.ulEOF)
        break;
}
*
```

2.10.2.2 `int MXCommon__ApplyConfigurationBackupFile (struct xsd__base64Binary * ByteArrayInput, xsd__unsignedLong ulEOF, struct MXCommon__Response * Response)`

Parameters

- [in] ***ByteArrayInput*** : Array of Bytes of the file
- [in] ***ulEOF*** : End of file flag
- [out] ***Response*** • `sResponse.iReturnValue` : Return value

- 0 : success
- -1: system error (see syserrno)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

This function is designed to be called repeatedly until all data is transfered. At this point the flag uEOF must be set to 1. The new configuration is then applied.

2.10.2.3 int MXCommon__ChangePassword (struct xsd__base64Binary * *PreviousUser*, struct xsd__base64Binary * *PreviousPassword*, struct xsd__base64Binary * *NewUser*, struct xsd__base64Binary * *NewPassword*, struct MXCommon__Response * *Response*)

The changes are immediately active.

Parameters

- [in] _ : No input parameter
- [out] *Response* • sResponse.iReturnValue : Return value
- 0 : success
 - -1: string PreviousUser is invalid
 - -2: string PreviousPassword is invalid
 - -3: string NewUser is invalid
 - -4: string NewPassword is invalid
 - -5: authentication failed
 - -100: system error while saving tokens (use syserrno for more information)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - sArray : message returned by the auto configuration start

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

Warning

The parameters transit in clear text. Use this functionality only on trusted networks.
 Given that ADDI-DATA GmbH takes security seriously, there is no way to change the password without knowing it. No "hidden back-door". This function makes it all too easy to lock a module, if you don't remember the password you set on it.

2.11 System state management

Every MSX-E modules are composed of several sub-systems that work together to provide the system functionalities.

Functions

- `int MXCommon__GetSubSystemState (xsd__unsignedLong SubsystemID, struct MXCommon__unsignedLongResponse *Response)`
Returns the current state of the specified sub-system.
- `int MXCommon__GetSubsystemIDFromName (struct xsd__base64Binary *SubsystemName, struct MXCommon__unsignedLongResponse *Response)`
Returns the ID of the sub-system of symbolic name "SubsystemName".
- `int MXCommon__GetStateIDFromName (xsd__unsignedLong SubsystemID, struct xsd__base64Binary *StateName, struct MXCommon__unsignedLongResponse *Response)`
Returns the ID of the state of symbolic name "StateName" of the sub-system of ID "SubsystemID".
- `int MXCommon__GetSubsystemNameFromID (xsd__unsignedLong SubsystemID, struct MXCommon__ByteArrayResponse *Response)`
Returns the symbolic name of the sub-system of numerical ID "SubsystemName".
- `int MXCommon__GetStateNameFromID (xsd__unsignedLong SubsystemID, xsd__unsignedLong StateID, struct MXCommon__ByteArrayResponse *Response)`
Returns the symbolic name of the state of numerical ID "StateID" of the sub-system of ID "SubsystemID".

2.11.1 Detailed Description

These sub-systems have a state that, for example, indicate if it functions nominally.

A sub-system is identified by its ID (a positive integer) and its symbolic name. Each state in the set of possible states for a given sub-system has also an ID and a symbolic name.

Names are less likely to change between releases of the MSX-E operating system. That is why manipulating names should be preferred against indexes in an application. Still, manipulating ID is more efficient.

The functions in this section provide a way to retrieve the association between names and indexes. `MXCommon__GetSubSystemState()` requests the state of a given sub-system.

Notice that the event manager is the recommended way to be warned of a change of state.

The list of sub-systems and their ID and associated name can be consulted on the web site of the module.

2.11.2 Function Documentation

2.11.2.1 `int MXCommon__GetSubSystemState (xsd__unsignedLong SubsystemID, struct MXCommon__unsignedLongResponse * Response)`

Parameters

- [in] **SubsystemID** sub-system numerical ID
- [out] **Response**
- `sResponse.iReturnValue` : Return value
 - 0 : success
 - -1: system error while executing the request (see `syserrno`)
 - -2: invalid parameter `SubsystemID`
 - `sResponse.syserrno` : System-error code. The value of the libc "errno" code, see `MXCommon__Strerror()`.

- Value The state of the sub-system "Id" at the moment of the execution of the request.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.11.2.2 `int MXCommon__GetSubsystemIDFromName (struct xsd__base64Binary * SubsystemName, struct MXCommon__unsignedLongResponse * Response)`

Parameters

[in] *SubsystemName* sub-system symbolic name.

[out] *Response* • sResponse.iReturnValue :Return value

- 0 : success
- -1: system error while executing the request (see syserrno)
- -2: invalid parameter SubsystemName

- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
- Value The numerical ID of the sub-system "SubsystemName".

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.11.2.3 `int MXCommon__GetStateIDFromName (xsd__unsignedLong SubsystemID, struct xsd__base64Binary * StateName, struct MXCommon__unsignedLongResponse * Response)`

Parameters

[in] *SubsystemID* sub-system numerical ID

[in] *StateName* state symbolic name.

[out] *Response* • sResponse.iReturnValue : Return value

- 0 : success
- -1: system error while executing the request (see syserrno)
- -2: invalid parameters SubsystemID or StateName

- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
- Value The numerical ID of the state "StateName".

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.11.2.4 `int MXCommon__GetSubsystemNameFromID (xsd__unsignedLong SubsystemID, struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] **SubsystemID** sub-system numerical ID.
- [out] **Response**
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error while executing the request (see syserrno)
 - -2: invalid parameter SubsystemName
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - sArray : The symbolic name associated with the ID.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.11.2.5 `int MXCommon__GetStateNameFromID (xsd__unsignedLong SubsystemID, xsd__unsignedLong StateID, struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] **SubsystemID** sub-system numerical ID.
- [in] **StateID** sub-system numerical ID.
- [out] **Response**
 - sResponse.iReturnValue : Return value
 - 0 success
 - -1 system error while executing the request (see syserrno)
 - -2 invalid parameters SubsystemID or StateID
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - sArray The symbolic name associated with the state numerical ID.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.12 Customer option management

Enable to get informations about the options of the system.

Functions

- `int MXCommon__GetOptionInformation (void *, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__ByteArrayResponse *Response)`
Enables to get information about the options available on the system.

2.12.1 Function Documentation

2.12.1.1 `int MXCommon__GetOptionInformation (void * __, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] *ulOption01*,: not used, set it to 0
- [in] *ulOption02*,: not used, set it to 0
- [out] *Response*
 - sArray : Option information string
 - sResponse Composed of iReturnValue and syserrno

Return values

- SOAP_OK* SOAP call success
- otherwise* SOAP protocol error

2.13 Synchronisation management

Manage the synchronisation state of the system.

Functions

- `int MXCommon__SetToMaster (void * __, xsd__unsignedLong ulState, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__Response *Response)`
Writes if the MSXE has to be always set to master The master mode (when enabled) make the system always detected as master.
- `int MXCommon__GetSynchronizationStatus (void * __, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__unsignedLongResponse *Response)`
Reads the status of the synchronization for the corresponding MSXE The master mode (when enabled) make the system always detected as master.

2.13.1 Function Documentation

2.13.1.1 `int MXCommon__SetToMaster (void * __, xsd__unsignedLong ulState, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__Response * Response)`

Parameters

- [in] *ulState* State of the supermaster mode
 - **0** automatic mode (default). The state of the system (master or slave) will be automatically detected by the system
 - **1** Set to master mode at all time. The system will always be detected as master
- [in] *ulOption01* Reserved. Set to 0
- [in] *ulOption02* Reserved. Set to 0
- [out] *Response* *iReturnValue*

- **0** The remote function performed OK
- **-1** System error occurred
- **-2** The PLD is not working
- **-3** The ulFilterTime parameter is wrong
- **-100** Internal system error occurred. See value of syserrno *syserrno* system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.13.1.2 `int MXCommon__GetSynchronizationStatus (void * __, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__unsignedLongResponse * Response)`

Parameters

[in] *ulOption01* Reserved. Set to 0

[in] *ulOption02* Reserved. Set to 0

[out] *Response sResponse.iReturnValue*

- **0** The remote function performed OK
- **-1** System error occurred
- **-2** The PLD is not working
- **-100** Internal system error occurred. See value of syserrno

sResponse.syserrno system error code (the value of the libc "errno" code)

ulValue State of the supermaster mode

- **0** Automatic mode (default). The state of the system (master or slave) will be automatically detected by the system
- **1** MSXE is always set as a master. The system will always be detected as master

Return values

0 SOAP_OK

Others See SOAP error

2.14 MSX-E3317 Acquisition functions

Contain the acquisition information and initialisation functions.

Modules

- [MSX-E3317 Acquisition information functions](#)

Contain the acquisition information functions.

- [MSX-E3317 Autorefresh functions](#)

In the auto refresh mode the measurement value is updated automatically after each acquisition.

- [MSX-E3317 Sequence functions](#)

A sequence is a list of channels (max 16) that are acquired.

2.15 MSX-E3317 Acquisition information functions

Contain the acquisition information functions.

Data Structures

- struct [MSXExxxx__AcquisitionGetChannelInfoResponse](#)

Functions

- int [MSXExxxx__AcquisitionGetNumberOfChannels](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)

Return the number of acquisition channels.

- int [MSXExxxx__AcquisitionGetChannelInfo](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, struct [MSXExxxx__AcquisitionGetChannelInfoResponse](#) *Response)

Return the selected acquisition channel type and hardware position.

2.15.1 Function Documentation

2.15.1.1 int [MSXExxxx__AcquisitionGetNumberOfChannels](#) (xsd__unsignedLong *ulOption1*, struct [MSXExxxx__unsignedLongResponse](#) * *Response*)

Parameters

[in] *ulOption1* : Reserved. Set to 0

[out] *Response* :

sResponse.iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -100: Internal system error occurred. See value of syserrno

sResponse.syserrno : system-error code (the value of the libc "errno" code)

ulValue : Number of available acquisition channels

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

2.15.1.2 `int MSXExxxx__AcquisitionGetChannelInfo (xsd__unsignedLong ulChannel,
xsd__unsignedLong ulOption1, struct MSXExxxx__AcquisitionGetChannelInfoResponse
* Response)`

Parameters

[in] *ulChannel* : Selected acquisition channel

[in] *ulOption1* : Reserved. Set to 0

[out] *Response* :

sResponse.iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Channel selection wrong
- -100: Internal system error occurred. See value of syserrno

sResponse.syserrno : system-error code (the value of the libc "errno" code)

ulType : Acquisition channel type

- 0 : Not available
- 1 : Temperature channel
- 2 : Pressure channel
- 3 : Transducer channel
- 4 : Analog input channel
- 5 : Analog input ICP channel
- 6 : Digital I/O port

ulHwPosition : Hardware position index (0 to 7) *ulChannelIndex* : Return the functionality channel index

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

2.16 MSX-E3317 Autorefresh functions

In the auto refresh mode the measurement value is updated automatically after each acquisition.

Data Structures

- struct [MSXExxxx__AcquisitionAutoRefreshGetValuesResponse](#)

Functions

- `int MSXExxxx__AcquisitionAutoRefreshInitAndStart (xsd__unsignedLong ulChannelMask, xsd__unsignedLong ulAverageValue, xsd__unsignedLong ulRefreshTime, xsd__unsignedLong ulRefreshTimeUnit, xsd__unsignedLong ulTriggerMask, xsd__unsignedLong ulTriggerMode, xsd__unsignedLong ulHardwareTriggerEdge, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulByTriggerNbrOfSeqToAcquire, xsd__unsignedLong ulDataFormat, xsd__unsignedLong ulForceStart, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, xsd__unsignedLong ulOption3, struct MSXExxxx__Response *Response)`

Starts an autorefresh acquisition using provided configuration.

- int `MSXExxxx__AcquisitionAutoRefreshGetValues` (xsd__unsignedLong ulBlocking, struct `MSXExxxx__AcquisitionAutoRefreshGetValuesResponse` *Response)

Reads the values acquired in auto refresh mode.

- int `MSXExxxx__AcquisitionAutoRefreshStopAndRelease` (void *_ , struct `MSXExxxx__Response` *Response)

Stops the current auto refresh acquisition.

2.16.1 Detailed Description

The acquisition is initialised and the values of each channels are stored in memory on the Ethernet E/A module MSX-E3317.

The PC reads the data asynchronously to the acquisition via the data socket or a SOAP function.

You can define a mask of all channels that should be acquired.

In the auto refresh mode you can activate the channel average value computation on the module

You can start the acquisition by a hardware trigger or a synchro trigger.

The hardware trigger can react to a rising wave, falling wave or both edges.

You have the following possibility:

- Defining a number of edges before a trigger action is generated

There are two trigger modes:(for the hardware or synchro trigger)

a) One shot

b) Sequence

a) One shot:

After the software start, the module is waiting for a trigger signal to start the acquisition. After this the trigger signal is ignored.

b) Sequence:

After the software start the module is waiting for the trigger signal and acquires x sequences (also adjustable) and then wait again.

2.16.2 Function Documentation

2.16.2.1 `int MSXExxxx_AcquisitionAutoRefreshInitAndStart (xsd__unsignedLong ulChannelMask, xsd__unsignedLong ulAverageValue, xsd__unsignedLong ulRefreshTime, xsd__unsignedLong ulRefreshTimeUnit, xsd__unsignedLong ulTriggerMask, xsd__unsignedLong ulTriggerMode, xsd__unsignedLong ulHardwareTriggerEdge, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulByTriggerNbrOfSeqToAcquire, xsd__unsignedLong ulDataFormat, xsd__unsignedLong ulForceStart, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, xsd__unsignedLong ulOption3, struct MSXExxxx_Response * Response)`

Parameters

- [in] **ulChannelMask** Mask of the channel to acquire by the auto refresh (1 bit = 1 Channel). 0 for all available acquisition channels
- [in] **ulAverageValue** Set the average value :
 - 1 : not used
 - max value : 255
- [in] **ulRefreshTimeUnit** Refresh Time Unit
 - 0 : microsecond
 - 1 : millisecond
 - 2 : second
- [in] **ulRefreshTime** Refresh Time
 - range from min 1000 to 65535 when the unit is the microsecond
 - range from min 1 to 65535 when the unit is the millisecond
 - range from min 1 to 65535 when the unit is the second
- [in] **ulTriggerMask** Define the source of the trigger
 - 0 : trigger disabled
 - 1 : Enable Hardware Digital Input Trigger
 - 2 : Enable Synchro Trigger
 - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/-Cosine input)
 - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- [in] **ulTriggerMode** Define the trigger mode
 - 1 : One shot trigger
 - 2 : Sequence trigger
- [in] **ulHardwareTriggerEdge** Define the edge of the hardware trigger who generates a trigger action
 - 1 : rising edge (Only if hardware trigger selected)
 - 2 : falling edge (Only if hardware trigger selected)
 - 3 : Both front (Only if hardware trigger selected)
- [in] **ulHardwareTriggerCount** Define the number of trigger events before the action occur
 - 1 : all trigger event start the action
 - max value : 65535
- [in] **ulByTriggerNbrOfSeqToAcquire** Define the number of sequence to acquire by each trigger event
 - 0 : continuous mode

- <> 0 : number of sequence : (1..0xFFFFFFFF)

D0 : Absolute Time stamp information

[in] **ulDataFormat** • 0 : no time stamp information

- 1 : time stamp information

D1 : Relative Time stamp information

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

D2 : Auto refresh counter information

- 0 : No auto refresh counter information
- 1 : Auto refresh counter information

D3 : System status information

- 0 : No system status information required
- 1 : System status information required

D4 : Data format

- 0: Digital value (see technical description)
- 1: Analog value (see technical description)

You can not select both absolute and relative time stamp simultaneously

[in] **ulForceStart** :

- 0 : Function return a error if any acquisition already in progress
- 1 : If any acquisition in progress then stop this

[in] **ulOption1** Reserved. Set to 0

[in] **ulOption2** Reserved. Set to 0

[in] **ulOption3** Reserved. Set to 0

[out] **Response** :

iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Any acquisition already in progress
- -3: Any selected channel not OK, call the diagnostic function for more information
- -4: Channel Mask error
- -5: Not available average value
- -6: Not available refresh time unit
- -7: The minimal refresh time is 1000 us !
- -8: The maximal refresh time is 65535 !
- -9: Trigger mask not available
- -10: Trigger mask : 2 different trigger source cannot be simultaneously be activated
- -11: Trigger mode not available
- -12: Trigger mask : 2 trigger mode cannot be simultaneously activated
- -13: Hardware trigger : front definition error
- -14: Hardware trigger count value not available
- -15: Nbr of sequence to acquire by trigger mode not available
- -16: Data format not available
- -17: Selected channels combination not available
- -100: Kernel function error

syserrno : system-error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

2.16.2.2 int MSXExxxx__AcquisitionAutoRefreshGetValues (xsd__unsignedLong ulBlocking, struct MSXExxxx__AcquisitionAutoRefreshGetValuesResponse * Response)

Reads the values acquired in auto refresh mode.

Parameters

[in] *ulBlocking* Wait a new value or read the actual value

- 0: Get the current auto refresh values
- 1: Wait a new auto refresh value cycle

[out] *Response* Response of the system

- *iReturnValue*
 - 0: The remote function performed OK
 - -1: Means an system error occurred
 - -2: No Acquisition in progress
 - -3: 2s timeout occurred. This if you have enabled the blocking mode.
 - -4: 2s timeout occurred. This if you do not have enabled the blocking mode, and if the first value is not available.
 - -100: Internal system error occurred. See value of *syserrno*
- *syserrno* system error code (the value of the libc "errno" code)
- *ulTimeStampLow* number of microseconds since the Epoch
- *ulTimeStampHigh* number of seconds since the Epoch
- *ulCounterValue* counter value
- *dValue* Array that contains the channels values
 - *dValue[0]* Value of channel 0
 - ...
 - *dValue[15]* Value of channel 15

Return values

0 SOAP_OK

Others See SOAP error

2.16.2.3 int MSXExxxx__AcquisitionAutoRefreshStopAndRelease (void * _, struct MSXExxxx__Response * Response)

Parameters

[in] _ Dummy parameter

[out] *Response* :

iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -100: Kernel function error

syserrno : system-error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

Must be called before any another call to MSXExxxx__AcquisitionAutoRefreshInitAndStart.

2.17 MSX-E3317 Sequence functions

A sequence is a list of channels (max 16) that are acquired.

Data Structures

- struct [MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam](#)

Functions

- int [MSXExxxx__AcquisitionSequenceInitAndStart](#) (xsd_unsignedLong ulNbrOfChannel, struct [MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam](#) *psChannelList, xsd_unsignedLong ulAcquisitionTime, xsd_unsignedLong ulAcquisitionTimeUnit, xsd_unsignedLong ulNbrOfSequence, xsd_unsignedLong ulNbrMaxSequenceToTransfer, xsd_unsignedLong ulTriggerMask, xsd_unsignedLong ulTriggerMode, xsd_unsignedLong ulHardwareTriggerEdge, xsd_unsignedLong ulHardwareTriggerCount, xsd_unsignedLong ulByTriggerNbrOfSeqToAcquire, xsd_unsignedLong ulDataFormat, xsd_unsignedLong ulForceStart, xsd_unsignedLong ulOption1, xsd_unsignedLong ulOption2, xsd_unsignedLong ulOption3, struct [MSXExxxx__Response](#) *Response)

Initialises and starts the sequence acquisition mode.

- int [MSXExxxx__AcquisitionSequenceStopAndRelease](#) (void *_ , struct [MSXExxxx__Response](#) *Response)

Stop and release the sequence acquisition mode.

2.17.1 Detailed Description

It can be any order of the channels in this list.

There are different sequence modes:

- Certain number of sequences / continuous

a) Certain number of sequences:

After the acquisition of the defined number of sequences, the acquisition is stopped automatically.

b) Continuous:

The sequences are acquired continuously until a software-stop-command occurs.

You can start the acquisition by a hardware or synchro trigger.

The hardware trigger can react to a rising, falling or both edges.

You have the following possibility:

- Defining a number of edges before a trigger action is generated

There are two trigger modes (for the hardware or synchro trigger):

a) One shot

b) Sequence

a) One shot:

After the software start, the module is waiting for a trigger signal to start the acquisition. After this the trigger signal is ignored.

b) Sequence:

After the software start the module is waiting for the trigger signal and acquires x sequences (also adjustable) and then wait again.

2.17.2 Function Documentation

2.17.2.1 `int MSXExxxx_AcquisitionSequenceInitAndStart (xsd__unsignedLong ulNbrOfChannel, struct MSXExxxx_AcquisitionSequenceInitAndStartChannelListParam * psChannelList, xsd__unsignedLong ulAcquisitionTime, xsd__unsignedLong ulAcquisitionTimeUnit, xsd__unsignedLong ulNbrOfSequence, xsd__unsignedLong ulNbrMaxSequenceToTransfer, xsd__unsignedLong ulTriggerMask, xsd__unsignedLong ulTriggerMode, xsd__unsignedLong ulHardwareTriggerEdge, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulByTriggerNbrOfSeqToAcquire, xsd__unsignedLong ulDataFormat, xsd__unsignedLong ulForceStart, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, xsd__unsignedLong ulOption3, struct MSXExxxx_Response * Response)`

Initialises and starts the sequence acquisition mode.

Parameters

- [in] *ulNbrOfChannel* : Nbr of channel in the sequence
- [in] *psChannelList* : List of the channel who compose the sequence.
- [in] *ulAcquisitionTime* : Acquisition Time
 - range from min 1000 to 65535 when the unit is the microsecond
 - range from min 1 to 65535 when the unit is the millisecond
 - range from min 1 to 65535 when the unit is the second
- [in] *ulAcquisitionTimeUnit* : Acquisition Time Unit
 - 0 : us
 - 1 : ms
 - 2 : s
- [in] *ulNbrOfSequence* : Number of sequence to acquire :

- 0 : continuous mode
 - <> 0 : number of sequence
- [in] ***ulNbrMaxSequenceToTransfer*** : Max nbr of sequence to acquire before a data transfer : (1,4096)
- [in] ***ulTriggerMask*** : Define the source of the trigger
- 0 : trigger disabled
 - 1 : Enable Hardware Digital Input Trigger
 - 2 : Enable Synchro Trigger
 - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/-Cosine input)
 - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- [in] ***ulTriggerMode*** : Define the trigger mode
- 1 : One shot trigger
 - 2 : Sequence trigger
- [in] ***ulHardwareTriggerEdge*** : Define the edge of the hardware trigger who generate a trigger action
- 1 : rising front (Only if hardware trigger selected)
 - 2 : falling front (Only if hardware trigger selected)
 - 3 : Both front (Only if hardware trigger selected)
- [in] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
- 1 : all trigger event start the action
 - max value : 65535
- [in] ***ulByTriggerNbrOfSeqToAcquire*** : define the number of sequence to acquire by each trigger event
- 0 : continuous mode
 - <> 0 : number of sequence : (1..0xFFFFFFFF)
- [in] ***ulDataFormat*** : Data format option
- D0 : Absolute Time stamp information
- 0 : no time stamp information
 - 1 : time stamp information
- D1 : Relative Time stamp information
- 0 : no time stamp information
 - 1 : time stamp information
- D2 : Sequence counter information
- 0 : No sequence counter information
 - 1 : Sequence counter information
- D3 : System status information
- 0 : No system status information required
 - 1 : System status information required
- D4 : Data format
- 0: Digital value (see technical description)
 - 1: Analog value (see technical description)

You can not select both absolute and relative time stamp simultaneously

[in] **ulForceStart** :

- 0 : Function return a error if any acquisition already in progress
- 1 : If any acquisition in progress then stop this

[in] **ulOption1** : Reserved. Set to 0

[in] **ulOption2** : Reserved. Set to 0

[in] **ulOption3** : Reserved. Set to 0

[out] **Response** :

iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Any acquisition already in progress
- -3: The nbr of channel in the sequence is null or too high
- -4: Channel index selection error
- -5: Channel already selected
- -6: Any selected channel not OK, call the diagnostic function for more information
- -7: Not available acquisition time unit
- -8: The minimal acquisition time is 1000 us !
- -9: The maximal acquisition time is 65535 !
- -10: Transfer sequence size error (1 to 4096) !
- -11: The total number of sequences is not a multiple from number of sequences to transfer
- -12: Trigger mask not available
- -13: Trigger mask : 2 different trigger source cannot be simultaneously be activated
- -14: Trigger mode not available
- -15: Trigger mask : 2 trigger mode cannot be simultaneously be activated
- -16: Hardware trigger : front definition error
- -17: Hardware trigger count value not available
- -18: Nbr of sequence to acquire by trigger mode not available
- -19: Data format not available
- -20: Selected channels combination not available
- -100: Start sequence kernel function error

syserrno : system-error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.17.2.2 int MSXExxxx__AcquisitionSequenceStopAndRelease (void * __, struct MSXExxxx__Response * *Response*)

Parameters

[in] **_** : no input parameter

[out] **Response** :

iReturnValue :

- 0: Means the remote function performed OK

- -1: Means an system error occurred
- -2: No sequence acquisition in progress
- -100: Kernel function error

syserrno : system-error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

2.18 MSX-E3317 Pressure functions

Contain the Pressure initialisation and diagnostic functions.

Modules

- [MSX-E3317 Pressure initialisation/information functions](#)
Contain the Pressure initialisation/information functions.

2.19 MSX-E3317 Pressure initialisation/information functions

Contain the Pressure initialisation/information functions.

Data Structures

- struct [MSXExxxx__PressureGetConfigurationResponse](#)

Functions

- int [MSXExxxx__PressureGetNumberOfChannels](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Return the number of pressure channels.
- int [MSXExxxx__PressureSetChannelConfiguration](#) (xsd__unsignedLong ulChannel, xsd__double dSensorSensibility, xsd__double dSensorOffset, xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)
Pressure sensor configuration for the selected channel.
- int [MSXExxxx__PressureSetSamplingRate](#) (xsd__unsignedLong ulChannelGroup, xsd__unsignedLong ulBaseSamplingRate, xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)
Pressure acquisition sampling rate selection.
- int [MSXExxxx__PressureGetConfiguration](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, struct [MSXExxxx__PressureGetConfigurationResponse](#) *Response)
Get the selected pressure channel current configuration.

2.19.1 Function Documentation

2.19.1.1 `int MSXExxxx__PressureGetNumberOfChannels (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Parameters

- [in] *ulOption1* : Reserved. Set to 0
- [out] *Response* :
- sResponse.iReturnValue* :
- 0: Means the remote function performed OK
 - -1: Means an system error occurred
- sResponse.syserrno* : system-error code (the value of the libc "errno" code)
- ulValue* : Number of available pressure channels

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

2.19.1.2 `int MSXExxxx__PressureSetChannelConfiguration (xsd__unsignedLong ulChannel, xsd__double dSensorSensibility, xsd__double dSensorOffset, xsd__unsignedLong ulOption1, struct MSXExxxx__Response * Response)`

Remarks

For MSXE with pressure functionality :

- Before revision 6982 the dSensorOffset parameter is given in mV.

$$\text{Pressure value (in Unit)} = (\text{AnalogValue (mV)} - \text{dSensorOffset (mV)}) / (\text{BridgeSupply (V)} * \text{dSensorSensibility(mV / V / Unit)})$$
- Since revision 6982 the dSensorOffset parameter is given in Unit instead of mV.

$$\text{Pressure value (in Unit)} = (\text{AnalogValue (mV)}) / (\text{BridgeSupply (V)} * \text{dSensorSensibility(mV / V / Unit)}) - \text{dSensorOffset (Unit)}$$

Parameters

- [in] *ulChannel* : Channel selection (0 to 15 or 255 for all channels)
- [in] *dSensorSensibility* : Sensor sensibility (mV/V/bar or mV/V/mbar or mV/V/Pa, ...). Refer to sensor documentation
- [in] *dSensorOffset* : Sensor offset in unit (bar/Newton/Pa/Psi...) that depends on the sensor Refer to sensor documentation
- [out] *Response* :
- iReturnValue* :
- 0: Means the remote function performed OK
 - -1: Means an system error occurred
 - -2: Channel selection wrong
 - -3: Sensor sensibility selection wrong
 - -4: Acquisition in progress. Can not change the configuration

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

2.19.1.3 `int MSXExxxx_PressureSetSamplingRate (xsd__unsignedLong ulChannelGroup,
xsd__unsignedLong ulBaseSamplingRate, xsd__unsignedLong ulOption1, struct
MSXExxxx_Response * Response)`

Parameters

[in] *ulChannelGroup* : Channel group selection.

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

[in] *ulBaseSamplingRate* : Sampling rate selection

- 5 for 5Hz
- 10 for 10Hz
- 20 for 20Hz
- 40 for 40Hz
- 80 for 80Hz
- 160 for 160Hz
- 320 for 320Hz
- 640 for 640Hz
- 1000 for 1000Hz
- 2000 for 2000Hz

If only one channel is used then the real sampling rate is $ulBaseSamplingRate / 2$

If all 2 channels are used then the real sampling rate is $ulBaseSamplingRate / 3$

[in] *ulOption1* : Reserved. Set to 0

[out] *Response* :

iReturnValue :

- 0 : Means the remote function performed OK
- -1 : Means an system error occurred
- -2 : Channel group selection error
- -3 : Sampling rate selection error
- -4 : Acquisition in progress. Can not change the sampling rate
- -100 : IOCTL system call error

syserrno : system-error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

2.19.1.4 `int MSXExxxx_PressureGetConfiguration (xsd_unsignedLong ulChannel,
xsd_unsignedLong ulOption1, struct MSXExxxx_PressureGetConfigurationResponse
* Response)`

Parameters

[in] *ulChannel* : Channel selection (0 to 15)

[in] *ulOption1* : Reserved. Set to 0

[out] *Response* :

sResponse.iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Channel selection wrong

sResponse.syserrno : system-error code (the value of the libc "errno" code)

dSensorSensibility : Sensor sensibility (mV/V/bar or mV/V/mbar or mV/V/Pa, ...). Refer to sensor documentation

dSensorOffset : Sensor V offset for 0 mV/V/bar, 0 mV/V/mbar, ... Refer to sensor documentation

ulBaseSamplingRate : Sampling rate selection

- 5 for 5Hz
- 10 for 10Hz
- 20 for 20Hz
- 40 for 40Hz
- 80 for 80Hz
- 160 for 160Hz
- 320 for 320Hz
- 640 for 640Hz
- 1000 for 1000Hz
- 2000 for 2000Hz

Returns

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Channel selection wrong

2.20 MSX-E3317 Digital I/O functions

Contain the digital I/O functions.

Modules

- [MSX-E3317 Digital I/O information, configuration functions](#)
Contain the digital I/O information, configuration functions.
- [MSX-E3317 Digital I/O filter functions](#)
Contain the digital I/O filter functions.

- [MSX-E3317 Digital I/O diagnostic functions](#)

Contain the digital I/O diagnostic functions.

- [MSX-E3317 Digital I/O read/write functions](#)

Contain the digital I/O read/write functions.

2.21 MSX-E3317 Digital I/O information, configuration functions

Contain the digital I/O information, configuration functions.

Data Structures

- struct [MSXExxxx_DigitalIOGetPortAvailableDirectionsResponse](#)

Functions

- int [MSXExxxx_DigitalIOGetNumberOfChannels](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Returns the number of digital I/O channels.
- int [MSXExxxx_DigitalIOGetNumberOfPorts](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Returns the number of digital I/O ports.
- int [MSXExxxx_DigitalIOGetPortAvailableDirections](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct [MSXExxxx_DigitalIOGetPortAvailableDirectionsResponse](#) *Response)
Returns the available directions for the selected port (input or output).
- int [MSXExxxx_DigitalIOSetPortDirections](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulDirection, xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)
Write the current digital I/O direction for the selected port.
- int [MSXExxxx_DigitalIOGetPortDirections](#) (xsd__unsignedLong ulPort, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Reads the current digital I/O direction for the selected port.

2.21.1 Function Documentation

2.21.1.1 int [MSXExxxx_DigitalIOGetNumberOfChannels](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) * Response)

Returns the number of digital I/O channels.

Parameters

[in] *ulOption1* Reserved. Set to 0.

[out] **Response** *sResponse.iReturnValue*

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-100**: Internal system error occurred. See value of syserrno

sResponse.syserrno system error code (the value of the libc "errno" code)

ulValue Number of available digital I/O channels

Return values

0 SOAP_OK

Others See SOAP error

2.21.1.2 int MSXExxxx_DigitalIOGetNumberOfPorts (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)

Returns the number of digital I/O ports.

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

Parameters

[in] *ulOption1* Reserved. Set to 0

[out] **Response** *sResponse.iReturnValue*

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-100**: Internal system error occurred. See value of syserrno

sResponse.syserrno system error code (the value of the libc "errno" code)

ulValue Number of available digital I/O ports

Return values

0 SOAP_OK

Others See SOAP error

2.21.1.3 int MSXExxxx_DigitalIOGetPortAvailableDirections (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx_DigitalIOGetPortAvailableDirectionsResponse * Response)

Returns the available directions for the selected port (input or output).

Parameters

[in] *ulPort* Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)

Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.

[in] *ulOption1* Reserved. Set to 0

[out] **Response** *sResponse.iReturnValue*

- **0**: The remote function performed OK

- **-1**: System error occurred
- **-2**: The `ulPort` parameter is wrong
- **-100**: Internal system error occurred. See value of `syserrno`

sResponse.syserrno system error code (the value of the libc "errno" code)

ulInputs Digital inputs availability. Each bit indicates if the channel can be used as an input.
Example:

- **1**: I/O 0 of the selected port can be used as an input
- **4**: I/O 2 of the selected port can be used as an input
- **3**: I/Os 0 and 1 of the selected port can be used as input

ulOutputs Digital outputs availability. Each bit indicates if the channel can be used as an output.
Example:

- **1**: I/O 0 of the selected port can be used as an output
- **4**: I/O 2 of the selected port can be used as an output
- **3**: I/Os 0 and 1 of the selected port can be used as output

Return values

0 SOAP_OK

Others See SOAP error

2.21.1.4 `int MSXExxxx_DigitalIOSetPortDirections (xsd__unsignedLong ulPort, xsd__unsignedLong ulDirection, xsd__unsignedLong ulOption1, struct MSXExxxx_Response * Response)`

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

Parameters

[in] ***ulPort*** Selected digital I/O port (0 to `MSXExxxx_DigitalIOGetNumberOfPorts`)

Please read the documentation of the `MSXExxxx_DigitalIOGetNumberOfPorts` for the description of a port.

[in] ***ulDirection*** Digital I/O direction. Each bit indicates if the channel is used as an input or an output. Example:

- **1**: I/O 0 of the selected port is configured as output, all the other I/Os of the selected port are configured as input
- **3**: I/Os 0 and 1 of the selected port are configured as output, all the other I/Os of the selected port are configured as input

[in] ***ulOption1*** Reserved. Set to 0

[out] ***Response iReturnValue***

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-2**: The `ulPort` parameter is wrong
- **-3**: The `ulDirection` parameter is wrong
- **-100**: Internal system error occurred. See value of `syserrno`

syserrno system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.21.1.5 int MSXExxxx_DigitalIOGetPortDirections (xsd__unsignedLong ulPort, struct MSXExxxx__unsignedLongResponse * Response)

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

Parameters

- [in] **ulPort** Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)
Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.
- [in] **ulOption1** Reserved. Set to 0
- [out] **Response sResponse.iReturnValue**
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulPort parameter is wrong
 - **-100**: Internal system error occurred. See value of syserrno
- sResponse.syserrno** system error code (the value of the libc "errno" code)
- ulValue** Current digital I/O direction. Each bit indicates if the channel is used as an input or an output. Example:
- **1**: I/O 0 of the selected port is configured as output, all the other I/Os of the selected port are configured as input
 - **3**: I/Os 0 and 1 of the selected port are configured as output, all the other I/Os of the selected port are configured as input

Return values

0 SOAP_OK

Others See SOAP error

2.22 MSX-E3317 Digital I/O filter functions

Contain the digital I/O filter functions.

Data Structures

- struct [MSXExxxx_DigitalIOGetInputsFilterConfigurationResponse](#)

Functions

- int [MSXExxxx_DigitalIOSetInputsFilterTime](#) (xsd__unsignedLong ulFilterTime, xsd__unsignedLong ulOption1, struct [MSXExxxx_Response](#) *Response)
Sets the filter time for the digital inputs in steps of 250 ns (max value: 16777215).
- int [MSXExxxx_DigitalIOEnableDisableInputsFilter](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulFilter, xsd__unsignedLong ulOption1, struct [MSXExxxx_Response](#) *Response)
Enables/disables the digital input filter for the selected port.

- `int MSXExxxx__DigitalIOGetInputsFilterConfiguration (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse *Response)`

Reads the digital inputs filter configuration for the selected port.

2.22.1 Function Documentation

2.22.1.1 `int MSXExxxx__DigitalIOSetInputsFilterTime (xsd__unsignedLong ulFilterTime, xsd__unsignedLong ulOption1, struct MSXExxxx__Response * Response)`

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

Parameters

[in] ***ulFilterTime*** Filter time for the digital inputs in steps of 250 ns (max value: 16777215)

- **0**: Disable the filter
- **1**: Sets the filter time to 250 ns
- **2**: Sets the filter time to 500 ns
- ...
- **16777215**: Sets the filter time to 4 s

[in] ***ulOption1*** Reserved. Set to 0

[out] ***Response*** Response of the system

- ***iReturnValue***
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The *ulFilterTime* parameter is wrong
 - **-100**: Internal system error occurred. See value of *syserrno*
- ***syserrno*** system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.22.1.2 `int MSXExxxx__DigitalIOEnableDisableInputsFilter (xsd__unsignedLong ulPort, xsd__unsignedLong ulFilter, xsd__unsignedLong ulOption1, struct MSXExxxx__Response * Response)`

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

Parameters

[in] ***ulPort*** Selected digital I/O port (0 to `MSXExxxx__DigitalIOGetNumberOfPorts`)

Please read the documentation of the `MSXExxxx__DigitalIOGetNumberOfPorts` for the description of a port.

[in] ***ulFilter*** Digital input filter selection. Each bit indicates if the filter is enabled on the input. Example:

- **1**: Filter only enabled on input 0

- **3**: Filter enabled on inputs 0 and 1

[in] **ulOption1** Reserved. Set to 0

[out] **Response iReturnValue**

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-2**: The ulPort parameter is wrong
- **-3**: The ulFilter parameter is wrong
- **-4**: Any selected input is not an input or a bidirectional channel
- **-100**: Internal system error occurred. See value of syserrno

syserrno system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.22.1.3 int MSXExxxx_DigitalIOGetInputsFilterConfiguration (xsd_ unsignedLong ulPort, xsd_ unsignedLong ulOption1, struct MSXExxxx_DigitalIOGetInputsFilterConfigurationResponse * Response)

Reads the digital inputs filter configuration for the selected port.

Parameters

[in] **ulPort** Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)

Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.

[in] **ulOption1** Reserved. Set to 0

[out] **Response sResponse.iReturnValue**

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-2**: The ulPort parameter is wrong
- **-100**: Internal system error occurred. See value of syserrno

sResponse.syserrno system error code (the value of the libc "errno" code)

ulFilterTime Filter time value (from 1 to 16777215) 1 corresponds to 250 ns, 2 corresponds to 500 ns, ...

ulFilter Digital inputs filter selection. Each bit indicate the filter state for one digital input channel.

Return values

0 SOAP_OK

Others See SOAP error

2.23 MSX-E3317 Digital I/O diagnostic functions

Contain the digital I/O diagnostic functions.

Functions

- `int MSXExxxx_DigitalIOTestOutputsShortCircuit (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx_unsignedLongResponse *Response)`
Get the short-circuit status of the outputs of the selected port.
- `int MSXExxxx_DigitalIORearmOutputsShortCircuit (xsd__unsignedLong ulOption1, struct MSXExxxx_Response *Response)`
Rearm the digital outputs short circuit.
- `int MSXExxxx_DigitalIOTestOutputsPowerSupply (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx_unsignedLongResponse *Response)`
Reads the current power supply status of the selected port.

2.23.1 Function Documentation

2.23.1.1 `int MSXExxxx_DigitalIOTestOutputsShortCircuit (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx_unsignedLongResponse * Response)`

Get the short-circuit status of the outputs of the selected port.

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

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

Parameters

- [in] **ulPort** Selected digital I/O port (0 to `MSXExxxx_DigitalIOGetNumberOfPorts`)
Please read the documentation of the `MSXExxxx_DigitalIOGetNumberOfPorts` for the description of a port.
- [in] **ulOption1** Reserved. Set to 0
- [out] **Response sResponse.iReturnValue :**
- **0:** The remote function performed OK
 - **-1:** System error occurred
 - **-2:** The ulPort parameter is wrong
 - **-100:** Internal system error occurred. See value of syserrno
- sResponse.syserrno :** system error code (the value of the libc "errno" code)
- ulValue :** Digital outputs short circuit state. Each bit represents the short-circuit state of one digital output channel.
- B0 : 0: Digital I/O 0/32 no short circuit. 1: Digital I/O 0/32 short circuit
 - ...
 - D31 : 0: Digital I/O 31/63 no short circuit. 1: Digital I/O 31/63 short circuit

Return values

0 SOAP_OK

Others See SOAP error

2.23.1.2 `int MSXExxxx__DigitalIORearmOutputsShortCircuit (xsd__unsignedLong ulOption1, struct MSXExxxx__Response * Response)`

Rearm the digital outputs short circuit.

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

Parameters

[in] ***ulOption1*** Reserved. Set to 0

[out] ***Response iReturnValue***

- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-100**: Internal system error occurred. See value of `syserrno`
- syserrno*** system error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

2.23.1.3 `int MSXExxxx__DigitalIOTestOutputsPowerSupply (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Reads the current power supply status of the selected port.

The digital outputs need an external power supply. This function checks the state of the power supply.

Parameters

[in] ***ulPort*** Selected digital I/O port (0 to `MSXExxxx__DigitalIOGetNumberOfPorts`)

Please read the documentation of the `MSXExxxx__DigitalIOGetNumberOfPorts` for the description of a port.

[in] ***ulOption1*** Reserved. Set to 0

[out] ***Response sResponse.iReturnValue***

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-2**: The PLD is not working
- **-3**: The `ulPort` parameter is wrong
- **-100**: Internal system error occurred. See value of `syserrno`

sResponse.syserrno system error code (the value of the libc "errno" code)

ulValue Current digital I/O power supply state. Each bit indicates the power supply state of the output. Example:

- **1**: No external supply voltage for the output 0
- **3**: No external supply voltage for the outputs 0 and 1

Return values

0 SOAP_OK

Others See SOAP error

2.24 MSX-E3317 Digital I/O read/write functions

Contain the digital I/O read/write functions.

Functions

- int `MSXExxxx_DigitalIOReadChannel` (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, struct `MSXExxxx_unsignedLongResponse` *Response)
Read the selected digital I/O channel.
- int `MSXExxxx_DigitalIOReadPort` (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct `MSXExxxx_unsignedLongResponse` *Response)
Read the selected digital I/O port.
- int `MSXExxxx_DigitalIOWriteChannel` (xsd__unsignedLong ulChannel, xsd__unsignedLong ulState, xsd__unsignedLong ulOption1, struct `MSXExxxx_Response` *Response)
Set the selected digital output channel to on or off.
- int `MSXExxxx_DigitalIOWritePort` (xsd__unsignedLong ulPort, xsd__unsignedLong ulState, xsd__unsignedLong ulOption1, struct `MSXExxxx_Response` *Response)
Write a value to the selected digital I/O port.

2.24.1 Function Documentation

2.24.1.1 int `MSXExxxx_DigitalIOReadChannel` (xsd__unsignedLong *ulChannel*, xsd__unsignedLong *ulOption1*, struct `MSXExxxx_unsignedLongResponse` * *Response*)

Read the selected digital I/O channel.

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

Parameters

- [in] ***ulChannel*** Selected digital I/O channel (0 to `MSXExxxx_DigitalIOGetNumberOfChannels`)
- [in] ***ulOption1*** Reserved. Set to 0
- [out] ***Response sResponse.iReturnValue*** :
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulChannel parameter is wrong
 - **-100**: Internal system error occurred. See value of syserrno
- sResponse.syserrno*** system error code (the value of the libc "errno" code)
- ulValue*** : Digital I/O channel state
- 0: Digital I/O channel is low
 - 1: Digital I/O channel is high

Return values

0 SOAP_OK

Others See SOAP error

2.24.1.2 `int MSXExxxx_DigitalIOReadPort (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Read the selected digital I/O port.

Parameters

- [in] **ulPort** Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)
Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.
- [in] **ulOption1** Reserved. Set to 0
- [out] **Response sResponse.iReturnValue**
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulPort parameter is wrong
 - **-100**: Internal system error occurred. See value of syserrno
- sResponse.syserrno** system error code (the value of the libc "errno" code)
- ulValue** Digital I/O state. Each bit indicates the state of one digital I/O channel.
- D0 : 0: Digital I/O 0/32 is low. 1: Digital I/O 0/32 is high
 - ...
 - D31 : 0: Digital I/O 31/63 is low. 1: Digital I/O 31/63 is high

Return values

0 SOAP_OK

Others See SOAP error

2.24.1.3 `int MSXExxxx_DigitalIOWriteChannel (xsd__unsignedLong ulChannel, xsd__unsignedLong ulState, xsd__unsignedLong ulOption1, struct MSXExxxx__Response * Response)`

Set the selected digital output channel to on or off.

Parameters

- [in] **ulChannel** Selected digital I/O channel (0 to MSXExxxx_DigitalIOGetNumberOfChannels)
- [in] **ulState** Digital I/O channel state
- **0**: Set the digital I/O output channel to low
 - **1**: Set the digital I/O output channel to high
- [in] **ulOption1** Reserved. Set to 0
- [out] **Response iReturnValue**
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulChannel parameter is wrong
 - **-3**: The ulState parameter is wrong
 - **-4**: The selected digital I/O is not an output
 - **-100**: Internal system error occurred. See value of syserrno

syserrno system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.24.1.4 int MSXExxxx__DigitalIOWritePort (xsd__unsignedLong ulPort, xsd__unsignedLong ulState, xsd__unsignedLong ulOption1, struct MSXExxxx__Response * Response)

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

Parameters

- [in] **ulPort** Selected digital I/O port (0 to MSXExxxx__DigitalIOGetNumberOfPorts)
Please read the documentation of the MSXExxxx__DigitalIOGetNumberOfPorts for the description of a port.
 - [in] **ulState** Digital I/O state. Each bit set the state for one digital I/O channel (0: off, 1: on).
 - [in] **ulOption1** Reserved. Set to 0
 - [out] **Response iReturnValue**
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulPort parameter is wrong
 - **-3**: The ulState parameter is wrong
 - **-4**: Any digital I/O set to 1 is not an output channel
 - **-100**: Internal system error occurred. See value of syserrno
- syserrno** system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.25 MSX-E3317 Sine/Cosine functions

Contain the Sine/Cosine functions.

Modules

- [MSX-E3317 Sine/Cosine initialisation/information functions](#)
Contain the Sine/Cosine initialisation/information functions.
- [MSX-E3317 Sine/Cosine read and clear functions](#)
Contain the Sine/Cosine read and clear functions.
- [MSX-E3317 Sine/Cosine Index functions](#)
Contain the Sine/Cosine Index functions.
- [MSX-E3317 Sine/Cosine Compare functions](#)
Contain the Sine/Cosine Compare functions.

2.26 MSX-E3317 Sine/Cosine initialisation/information functions

Contain the Sine/Cosine initialisation/information functions.

Data Structures

- struct [MSXExxxx__SinCosInitResponse](#)
- struct [MSXExxxx__SinCosGetConfigurationResponse](#)

Functions

- int [MSXExxxx__SinCosGetNumberOfChannels](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Return the number of Sine/Cosine channels.
- int [MSXExxxx__SinCosInit](#) (xsd__unsignedLong ulChannel, xsd__double dSignalPeriod, xsd__unsignedLong ulResolution, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct [MSXExxxx__SinCosInitResponse](#) *Response)
Initialise the selected Sine/Cosine channel.
- int [MSXExxxx__SinCosRelease](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXExxxx__Response](#) *Response)
Release the selected Sine / Cosine channel.
- int [MSXExxxx__SinCosGetConfiguration](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption02, struct [MSXExxxx__SinCosGetConfigurationResponse](#) *Response)
Get the current configuration of the selected Sine/Cosine channel.

2.26.1 Function Documentation

2.26.1.1 int [MSXExxxx__SinCosGetNumberOfChannels](#) (xsd__unsignedLong *ulOption1*, struct [MSXExxxx__unsignedLongResponse](#) * *Response*)

Return the number of Sine/Cosine channels.

Parameters

[in] *ulOption1* Reserved. Set to 0

[out] *Response* Response of the system

- *sResponse*
 - *iReturnValue* Function return code
 - * **0**: The remote function performed OK
 - * **-1**: System error occurred
 - * **-100**: Internal system error occurred. See value of syserrno
 - *syserrno* System error code (the value of the libc "errno" code)
- *ulValue* Number of available Sine / Cosine channels

Return values*0* SOAP_OK*Others* See SOAP error

2.26.1.2 `int MSXExxxx__SinCosInit (xsd__unsignedLong ulChannel, xsd__double dSignalPeriod, xsd__unsignedLong ulResolution, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXExxxx__SinCosInitResponse * Response)`

Initialise the selected Sine/Cosine channel.

Parameters

[in] *ulChannel* Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)

[in] *dSignalPeriod* Signal period (Unit/period)

[in] *ulResolution* Resolution to use for the measure (steps/period)

Maximum input frequency corresponding to the parameter *ulResolution*:

Resolution	Max. Freq. Hz.
16	250000
32	162500
40	16300
64	81300
80	16300
100	26000
128	40600
160	16300
200	26000
256	20300
320	16300
400	13000
500	10400
512	10200
800	6500
1000	5200
1024	5100
1600	3300
2000	2600
2048	2540

4096	1270	
+-----+	+-----+	+-----+
8192	635	
+-----+	+-----+	+-----+

*

[in] **ulOption01** Reserved. Set it to 0
[in] **ulOption02** Reserved. Set it to 0
[in] **ulOption03** Reserved. Set it to 0
[in] **ulOption04** Reserved. Set it to 0
[out] **Response** Response of the system

- **sResponse**
 - **iReturnValue** Function return code
 - * **0**: The remote function performed OK
 - * **-1**: System error occurred
 - * **-2**: The PLD is not working
 - * **-3**: The ulChannel parameter is wrong
 - * **-4**: The current status is not correct. The selected Sine/Cosine channel has already been initialised. Please call the MSXExxxx__SinCosRelease and retry
 - * **-5**: The dSignalPeriod parameter is wrong
 - * **-6**: The ulResolution parameter is wrong
 - * **-7**: Gain calibration error occurred. Please check the connection with the sensor
 - * **-21**: Timeout while initialising the Sine/Cosine sensor. Please check the connection with the sensor
 - * **-100**: Internal system error occurred. See value of syserrno
 - **syserrno** System error code (the value of the libc "errno" code)
- **ulMaxInputFrequency** Return the maximal input frequency that can be used (in Hz).

Return values

0 SOAP_OK

Others See SOAP error

2.26.1.3 `int MSXExxxx__SinCosRelease (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

Release the selected Sine / Cosine channel.

Parameters

[in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
[in] **ulOption01** Reserved. Set it to 0
[in] **ulOption02** Reserved. Set it to 0
[out] **Response** Response of the system

- **iReturnValue** Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working

- **-3**: The `ulChannel` parameter is wrong
- **-4**: The current status is not correct. The selected Sine/Cosine channel is not initialised
- **-100**: Internal system error occurred. See value of `syserrno`
- ***syserrno*** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.26.1.4 `int MSXExxxx__SinCosGetConfiguration (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption02, struct MSXExxxx__SinCosGetConfigurationResponse * Response)`

Get the current configuration of the selected Sine/Cosine channel.

Parameters

- [in] ***ulChannel*** Channel selection (0 to `MSXExxxx__SinCosGetNumberOfChannels - 1`)
- [in] ***ulOption01*** Reserved. Set it to 0
- [in] ***ulOption02*** Reserved. Set it to 0
- [out] ***Response*** Response of the system
 - ***sResponse***
 - ***iReturnValue*** Function return code
 - * **0**: The remote function performed OK
 - * **-1**: System error occurred
 - * **-2**: The PLD is not working
 - * **-3**: The `ulChannel` parameter is wrong
 - * **-100**: Internal system error occurred. See value of `syserrno`
 - ***syserrno*** System error code (the value of the libc "errno" code)
 - ***ulInitialisationState*** Initialisation state of the Sine/Cosine channel (0: uninitialised, 1: initialised)
 - ***dSignalPeriod*** Signal period (Unit/period)
 - ***ulResolution*** Resolution to use for the measure (steps/period)
 - ***ulIndexState*** State of the index logic (0: uninitialised, 1: initialised)
 - ***ulIndexEdge*** Edge selected for the index logic
 - ***ulIndexAction*** Action to do when the index signal occurs
 - ***ulCompareLogicState*** State of the compare logic (0: uninitialised, 1: initialised)
 - ***dCompareLogicValue*** Compare value of the compare logic
 - ***ulCompareLogicValueFormat*** Format of the compare value of the compare logic (0: Raw, 1: Standardised)
 - ***ulCompareLogicMode*** Compare mode of the compare logic (0: Simple, 1: Modulo)
 - ***ulCompareLogicSynchro*** Synchro trigger generation of the compare logic (0: No, 1: Yes)

Return values

0 SOAP_OK

Others See SOAP error

2.27 MSX-E3317 Sine/Cosine read and clear functions

Contain the Sine/Cosine read and clear functions.

Data Structures

- struct `MSXExxxx__SinCosReadResponse`

Functions

- int `MSXExxxx__SinCosRead` (`xsd__unsignedLong` ulChannel, `xsd__unsignedLong` ulOption01, `xsd__unsignedLong` ulOption02, `xsd__unsignedLong` ulOption03, `xsd__unsignedLong` ulOption04, struct `MSXExxxx__SinCosReadResponse` *Response)
Read the measured value of the selected Sine/Cosine channel.
- int `MSXExxxx__SinCosClear` (`xsd__unsignedLong` ulChannel, `xsd__unsignedLong` ulOption01, `xsd__unsignedLong` ulOption02, struct `MSXExxxx__Response` *Response)
Clear the measured value of the selected Sine/Cosine channel.
- int `MSXExxxx__SinCosClearError` (`xsd__unsignedLong` ulChannel, `xsd__unsignedLong` ulOption01, `xsd__unsignedLong` ulOption02, struct `MSXExxxx__Response` *Response)
Clear the error flag of the selected Sine/Cosine channel.

2.27.1 Function Documentation

2.27.1.1 int `MSXExxxx__SinCosRead` (`xsd__unsignedLong` ulChannel, `xsd__unsignedLong` ulOption01, `xsd__unsignedLong` ulOption02, `xsd__unsignedLong` ulOption03, `xsd__unsignedLong` ulOption04, struct `MSXExxxx__SinCosReadResponse` * Response)

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

Parameters

- [in] **ulChannel** Channel selection (0 to `MSXExxxx__SinCosGetNumberOfChannels` - 1)
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [in] **ulOption03** Reserved. Set it to 0
- [in] **ulOption04** Reserved. Set it to 0
- [out] **Response** Response of the system
 - **sResponse**
 - **iReturnValue** Function return code
 - * **0**: The remote function performed OK
 - * **-1**: System error occurred
 - * **-2**: The PLD is not working
 - * **-3**: The ulChannel parameter is wrong
 - * **-4**: The current status is not correct. The selected Sine/Cosine channel is not initialised

- * **-100**: Internal system error occurred. See value of `syserrno`
- *syserrno* System error code (the value of the libc "errno" code)
- *dValue* Measured value (Unit)
- *ulValue* Measured value (Raw value)
- *ulMeasureError* Status of the Sine/Cosine measurement. If the value is 1, then an error occurred. When an error occurred, the reference point is corrupted. To correct it, you must first reset the error flag using the function `MSXExxxx__SinCosClearError`, and then reset your reference point.
- *ulInfo01* Reserved
- *ulInfo02* Reserved

Return values

0 SOAP_OK

Others See SOAP error

2.27.1.2 `int MSXExxxx__SinCosClear (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

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

Parameters

- [in] *ulChannel* Channel selection (0 to `MSXExxxx__SinCosGetNumberOfChannels - 1`)
- [in] *ulOption01* Reserved. Set it to 0
- [in] *ulOption02* Reserved. Set it to 0
- [out] *Response* Response of the system
 - *iReturnValue* Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The *ulChannel* parameter is wrong
 - **-4**: The current status is not correct. The selected Sine/Cosine channel is not initialised
 - **-100**: Internal system error occurred. See value of `syserrno`
 - *syserrno* System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.27.1.3 `int MSXExxxx__SinCosClearError (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

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

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
 - **iReturnValue** Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The ulChannel parameter is wrong
 - **-4**: The current status is not correct. The selected Sine/Cosine channel is not initialised
 - **-100**: Internal system error occurred. See value of syserrno
 - **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.28 MSX-E3317 Sine/Cosine Index functions

Contain the Sine/Cosine Index functions.

Functions

- int [MSXExxxx__SinCosInitIndex](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulEdge, xsd__unsignedLong ulAction, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXExxxx__Response](#) *Response)
Initialise the Index configuration.
- int [MSXExxxx__SinCosReleaseIndex](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXExxxx__Response](#) *Response)
Release the Index configuration.

2.28.1 Function Documentation

2.28.1.1 int [MSXExxxx__SinCosInitIndex](#) (xsd__unsignedLong *ulChannel*, xsd__unsignedLong *ulEdge*, xsd__unsignedLong *ulAction*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, struct [MSXExxxx__Response](#) * *Response*)

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

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)

- [in] **ulEdge** Edge selection
- **0b01 = 1**: Rising edge
 - **0b10 = 2**: Falling edge
 - **0b11 = 3**: Both edges
- [in] **ulAction** Action to do when the index trigger occurs
- **0b00 = 0**: Do nothing (but index trigger can also be used to trigger the acquisition)
 - **0b01 = 1**: Clear the value of the sensor
 - **0b10 = 2**: Generate a synchro trigger
 - **0b11 = 3**: Clear the value of the sensor AND Generate a synchro trigger
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
- **iReturnValue** Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The ulChannel parameter is wrong
 - **-4**: The ulEdge parameter is wrong
 - **-5**: The ulAction parameter is wrong
 - **-6**: The current status is not correct. The selected Sine/Cosine channel is not initialised
 - **-7**: The current status is not correct. The selected Sine/Cosine channel Index Logic has already been initialised. Please call the function MSXExxxx__SinCosReleaseIndex
 - **-100**: Internal system error occurred. See value of syserrno
 - **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.28.1.2 `int MSXExxxx__SinCosReleaseIndex (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

Release the Index configuration.

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
- **iReturnValue** Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working

- **-3**: The ulChannel parameter is wrong
- **-4**: The current status is not correct. The selected Sine/Cosine channel Index Logic is not initialised
- **-100**: Internal system error occurred. See value of syserrno
- **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.29 MSX-E3317 Sine/Cosine Compare functions

Contain the Sine/Cosine Compare functions.

Functions

- `int MSXExxxx__SinCosInitAndEnableCompareLogic (xsd__unsignedLong ulChannel, xsd__double dValue, xsd__unsignedLong ulValueFormat, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response *Response)`

Initialise and enable a Sine/Cosine Compare Logic.

- `int MSXExxxx__SinCosDisableAndReleaseCompareLogic (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response *Response)`

Disable and Release the Sine/Cosine Compare Logic.

2.29.1 Function Documentation

2.29.1.1 `int MSXExxxx__SinCosInitAndEnableCompareLogic (xsd__unsignedLong ulChannel, xsd__double dValue, xsd__unsignedLong ulValueFormat, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

Initialise and enable a Sine/Cosine Compare Logic.

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] **dValue** Compare value. Possible value depends on the variable ulValueFormat
 - **If ulValueFormat is set to 0**: 0 to 0xFFFFFFFF
 - **If ulValueFormat is set to 1**: Standardised value (Unit). The value cannot be negative
- [in] **ulValueFormat** Format of the value dValue
 - **0**: Raw value
 - **1**: Standardised value. The result of the operation $dValue * (ulResolution / dSignalPeriod)$ must be an integer

[in] **ulMode** Compare mode

- **0**: Simple mode. As soon as the counter value corresponds to the compare value (dValue), a trigger or synchro trigger is released.
- **1**: Modulo mode. When the counter value corresponds to the compare value (dValue) or a multiple of it, a trigger or synchro trigger is released. In that mode, the value dValue cannot be 0.

[in] **ulSynchroTrigger** Generate synchro trigger, which can be used to trigger another MSX-E system, when the compare trigger is released

- **0**: No
- **1**: Yes

[in] **ulOption01** Reserved. Set it to 0

[in] **ulOption02** Reserved. Set it to 0

[out] **Response** Response of the system

- **iReturnValue** Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The ulChannel parameter is wrong
 - **-4**: The ulValueFormat parameter is wrong
 - **-5**: The ulMode parameter is wrong
 - **-6**: The ulSynchroTrigger parameter is wrong
 - **-7**: In Modulo mode, the parameter dValue cannot be 0
 - **-8**: The dValue parameter is wrong
 - **-9**: The current status is not correct. The selected Sine/Cosine channel is not initialised
 - **-10**: The current status is not correct. The selected Sine/Cosine channel Compare Logic has already been initialised. Please call the function MSXExxxx__SinCosDisableAndReleaseCompareLogic
 - **-100**: Internal system error occurred. See value of syserrno
- **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

2.29.1.2 `int MSXExxxx__SinCosDisableAndReleaseCompareLogic (xsd_unsignedLong ulChannel, xsd_unsignedLong ulOption01, xsd_unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

Disable and Release the Sine/Cosine Compare Logic

Parameters

[in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)

[in] **ulOption01** Reserved. Set it to 0

[in] **ulOption02** Reserved. Set it to 0

[out] **Response** Response of the system

- *iReturnValue* Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The ulChannel parameter is wrong
 - **-4**: The current status is not correct. The selected Sine/Cosine channel Compare Logic is not initialised
 - **-100**: Internal system error occurred. See value of syserrno
- *syserrno* System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

Chapter 3

Data Structure Documentation

3.1 ByteArray Struct Reference

Dynamic Array of byte - encapsulates C-type strings.

Data Fields

- `xsd__unsignedByte * __ptr`
pointer of byte
- `int __size`
size of the byte array in bytes
- `int __offset`
not used

3.1.1 Field Documentation

3.1.1.1 `xsd__unsignedByte* ByteArray::__ptr`

3.1.1.2 `int ByteArray::__size`

3.1.1.3 `int ByteArray::__offset`

3.2 DefaultResponse Struct Reference

Data Fields

- `xsd__int iReturnValue`
return value of the call :
- `xsd__int syserrno`
system-error code (the value of the libc "errno" code)

3.2.1 Field Documentation

3.2.1.1 xsd__int DefaultResponse::iReturnValue

- 0 means the remote function performed OK
- -1 means a system error occurred, the meaning of other values is function dependant and should be defined in the related header

3.2.1.2 xsd__int DefaultResponse::syserrno

3.3 MSXExxxx__AcquisitionAutoRefreshGetValuesResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) sResponse

Default return values.

- [xsd__unsignedLong](#) ulTimeStampLow

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong](#) ulTimeStampHigh

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong](#) ulCounterValue

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__double](#) dValue [16]

The meaning of this field is defined in the related header of the function who use this type.

3.3.1 Field Documentation

- 3.3.1.1 **struct DefaultResponse MSXExxxx__AcquisitionAutoRefreshGetValuesResponse::sResponse**
- 3.3.1.2 **xsd__unsignedLong MSXExxxx__AcquisitionAutoRefreshGetValuesResponse::ulTimeStampLow**
- 3.3.1.3 **xsd__unsignedLong MSXExxxx__AcquisitionAutoRefreshGetValuesResponse::ulTimeStampHigh**
- 3.3.1.4 **xsd__unsignedLong MSXExxxx__AcquisitionAutoRefreshGetValuesResponse::ulCounterValue**
- 3.3.1.5 **xsd__double MSXExxxx__AcquisitionAutoRefreshGetValuesResponse::dValue[16]**

3.4 MSXExxxx__AcquisitionGetChannelInfoResponse Struct Reference

Data Fields

- struct [DefaultResponse sResponse](#)
Default return values.
- [xsd__unsignedLong ulType](#)
Acquisition channel type
.
- [xsd__unsignedLong ulHwPosition](#)
Hardware position index (0 to 7).
- [xsd__unsignedLong ulChannelIndex](#)
Return the functionality channel index.

3.4.1 Field Documentation

- 3.4.1.1 **struct DefaultResponse MSXExxxx__AcquisitionGetChannelInfoResponse::sResponse**
- 3.4.1.2 **xsd__unsignedLong MSXExxxx__AcquisitionGetChannelInfoResponse::ulType**
 - 0 : Not available
 - 1 : Temperature channel
 - 2 : Pressure channel
 - 3 : Transducer channel
 - 4 : Analog input channel
 - 5 : Analog input ICP channel

- 6 : Digital I/O port
- 7 : Incremental counter channel

3.4.1.3 `xsd__unsignedLong MSXExxxx__AcquisitionGetChannelInfoResponse::ulHwPosition`

3.4.1.4 `xsd__unsignedLong MSXExxxx__AcquisitionGetChannelInfoResponse::ulChannelIndex`

3.5 MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam Struct Reference

Data Fields

- `xsd__unsignedLong ulChannelList [16]`

The meaning of this field is defined in the related header of the function who use this type.

3.5.1 Field Documentation

3.5.1.1 `xsd__unsignedLong MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam::ulChannelList[16]`

3.6 MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`

Default return values.

- `xsd__unsignedLong ulFilterTime`

Filter time value (from 1 to 16777215) 1 corresponds to 250 ns, 2 corresponds to 500 ns, ...

- `xsd__unsignedLong ulFilter`

Digital inputs filter selection.

3.6.1 Field Documentation

- 3.6.1.1** struct `DefaultResponse MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse::sResponse`
- 3.6.1.2** `xsd__unsignedLong MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse::ulFilterTime`
- 3.6.1.3** `xsd__unsignedLong MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse::ulFilter`

Each bit indicate the filter state for one digital input channel.

- D0 : 0: Digital I/O 0/32 filter disabled. 1: Digital I/O 0/32 filter enabled
- ...
- D31 : 0: Digital I/O 31/63 filter disabled. 1: Digital I/O 31/63 filter enabled

3.7 MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
Default return values.
- `xsd__unsignedLong ulInputs`
Digital input directions availability.
- `xsd__unsignedLong ulOutputs`
Digital output directions availability.

3.7.1 Field Documentation

- 3.7.1.1** struct `DefaultResponse MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse::sResponse`
- 3.7.1.2** `xsd__unsignedLong MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse::ulInputs`

Each bit indicates the availability for one channel.

- D0 : 0: Digital I/O 0/32 can not be used for input. 1: Digital I/O 0/32 can be used for input
- ...
- D31 : 0: Digital I/O 31/63 can not be used for input. 1: Digital I/O 31/63 can be used for input

3.7.1.3 `xsd__unsignedLong MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse::ulOutputs`

Each bit indicates the availability for one channel.

- D0 : 0: Digital I/O 0/32 can not be used for output. 1: Digital I/O 0/32 can be used for output
- ...
- D31 : 0: Digital I/O 31/63 can not be used for output. 1: Digital I/O 31/63 can be used for output

3.8 `MSXExxxx__FileResponse` Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
return values.
- struct `ByteArray sArray`
Dynamic Array of byte.
- `xsd__unsignedLong ulEOF`
flag indicating end of file.

3.8.1 Field Documentation

3.8.1.1 struct `DefaultResponse MSXExxxx__FileResponse::sResponse`

3.8.1.2 struct `ByteArray MSXExxxx__FileResponse::sArray`

3.8.1.3 `xsd__unsignedLong MSXExxxx__FileResponse::ulEOF`

3.9 `MSXExxxx__PressureGetConfigurationResponse` Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
Default return values.
- `xsd__double dSensorSensibility`
The meaning of this field is defined in the related header of the function who use this type.
- `xsd__double dSensorOffset`
The meaning of this field is defined in the related header of the function who use this type.
- `xsd__unsignedLong ulBaseSamplingRate`
The meaning of this field is defined in the related header of the function who use this type.

3.9.1 Field Documentation

3.9.1.1 struct DefaultResponse MSXExxxx__PressureGetConfigurationResponse::sResponse

3.9.1.2 xsd__double MSXExxxx__PressureGetConfigurationResponse::dSensorSensibility

3.9.1.3 xsd__double MSXExxxx__PressureGetConfigurationResponse::dSensorOffset

3.9.1.4 xsd__unsignedLong MSXExxxx__PressureGetConfigurationResponse::ulBaseSamplingRate

3.10 MSXExxxx__Response Struct Reference

Data Fields

- [xsd__int iReturnValue](#)
return value of the call :
- [xsd__int syserrno](#)
System-error code (the value of the libc "errno" code).

3.10.1 Field Documentation

3.10.1.1 xsd__int MSXExxxx__Response::iReturnValue

- 0 means the remote function performed OK
- -1 means a system error occurred, the meaning of other values is function dependant and should be defined in the related header

3.10.1.2 xsd__int MSXExxxx__Response::syserrno

3.11 MSXExxxx__SinCosGetConfigurationResponse Struct Reference

Data Fields

- struct [DefaultResponse sResponse](#)
Default return values.
- [xsd__unsignedLong ulInitialisationState](#)
The meaning of this field is defined in the related header of the function who use this type.
- [xsd__double dSignalPeriod](#)
The meaning of this field is defined in the related header of the function who use this type.
- [xsd__unsignedLong ulResolution](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulIndexState](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulIndexEdge](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulIndexAction](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulCompareLogicState](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__double dCompareLogicValue](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulCompareLogicValueFormat](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulCompareLogicMode](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulCompareLogicSynchro](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulInfo01](#)

The meaning of this field is defined in the related header of the function who use this type.

- [xsd__unsignedLong ulInfo02](#)

The meaning of this field is defined in the related header of the function who use this type.

3.11.1 Field Documentation

3.11.1.1 struct [DefaultResponse](#) [MSXExxxx__SinCosGetConfigurationResponse::sResponse](#)

3.11.1.2 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulInitialisationState](#)

3.11.1.3 [xsd__double](#) [MSXExxxx__SinCosGetConfigurationResponse::dSignalPeriod](#)

3.11.1.4 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulResolution](#)

3.11.1.5 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulIndexState](#)

3.11.1.6 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulIndexEdge](#)

3.11.1.7 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulIndexAction](#)

3.11.1.8 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulCompareLogicState](#)

3.11.1.9 [xsd__double](#) [MSXExxxx__SinCosGetConfigurationResponse::dCompareLogicValue](#)

3.11.1.10 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulCompareLogicValueFormat](#)

3.11.1.11 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulCompareLogicMode](#)

3.11.1.12 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulCompareLogicSynchro](#)

3.11.1.13 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulInfo01](#)

3.11.1.14 [xsd__unsignedLong](#) [MSXExxxx__SinCosGetConfigurationResponse::ulInfo02](#)

3.12 MSXExxxx__SinCosInitResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) [sResponse](#)
Default return values.
- [xsd__unsignedLong](#) [ulMaxInputFrequency](#)
The maximal input frequency that can be used (in Hz).
- [xsd__unsignedLong](#) [ulInfo01](#)
Reserved.
- [xsd__unsignedLong](#) [ulInfo02](#)
Reserved.

3.12.1 Field Documentation

3.12.1.1 struct `DefaultResponse MSXExxxx__SinCosInitResponse::sResponse`

3.12.1.2 `xsd__unsignedLong MSXExxxx__SinCosInitResponse::ulMaxInputFrequency`

3.12.1.3 `xsd__unsignedLong MSXExxxx__SinCosInitResponse::ulInfo01`

3.12.1.4 `xsd__unsignedLong MSXExxxx__SinCosInitResponse::ulInfo02`

3.13 MSXExxxx__SinCosReadResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`

Default return values.

- `xsd__double dValue`

Measured value (mm).

- `xsd__unsignedLong ulValue`

Measured value (raw value).

- `xsd__unsignedLong ulMeasureError`

Measure error.

- `xsd__unsignedLong ulInfo01`

Reserved.

- `xsd__unsignedLong ulInfo02`

Reserved.

3.13.1 Field Documentation

3.13.1.1 struct `DefaultResponse MSXExxxx__SinCosReadResponse::sResponse`

3.13.1.2 `xsd__double MSXExxxx__SinCosReadResponse::dValue`

3.13.1.3 `xsd__unsignedLong MSXExxxx__SinCosReadResponse::ulValue`

3.13.1.4 `xsd__unsignedLong MSXExxxx__SinCosReadResponse::ulMeasureError`

0 no error, 1 error occurred

3.13.1.5 xsd__unsignedLong MSXExxxx__SinCosReadResponse::ulInfo01

3.13.1.6 xsd__unsignedLong MSXExxxx__SinCosReadResponse::ulInfo02

3.14 MSXExxxx__unsignedLongResponse Struct Reference

Data Fields

- struct [DefaultResponse sResponse](#)

Default return values.

- [xsd__unsignedLong ulValue](#)

The meaning of this value is defined in the related header of the function who use this type.

3.14.1 Field Documentation

3.14.1.1 struct [DefaultResponse MSXExxxx__unsignedLongResponse::sResponse](#)

3.14.1.2 [xsd__unsignedLong MSXExxxx__unsignedLongResponse::ulValue](#)

3.15 MSXExxxx__unsignedLongTimeStampResponse Struct Reference

Data Fields

- struct [DefaultResponse sResponse](#)

Default return values.

- [xsd__unsignedLong ulValue](#)

the meaning of this value is defined in the related header of the function who use this type

- [xsd__unsignedLong ulTimeStampLow](#)

the meaning of this value is defined in the related header of the function who use this type

- [xsd__unsignedLong ulTimeStampHigh](#)

the meaning of this value is defined in the related header of the function who use this type

3.15.1 Field Documentation

3.15.1.1 struct `DefaultResponse MSXExxxx__unsignedLongTimeStampResponse::sResponse`

3.15.1.2 `xsd__unsignedLong MSXExxxx__unsignedLongTimeStampResponse::ulValue`

3.15.1.3 `xsd__unsignedLong MSXExxxx__unsignedLongTimeStampResponse::ulTimeStampLow`

3.15.1.4 `xsd__unsignedLong MSXExxxx__ -
unsignedLongTimeStampResponse::ulTimeStampHigh`

3.16 MXCommon__ByteArrayResponse Struct Reference

Response containing a C-type string.

Data Fields

- struct `DefaultResponse sResponse`
Default return values.
- struct `ByteArray sArray`
Dynamic Array of byte - encapsulates C-type strings.

3.16.1 Field Documentation

3.16.1.1 struct `DefaultResponse MXCommon__ByteArrayResponse::sResponse`

3.16.1.2 struct `ByteArray MXCommon__ByteArrayResponse::sArray`

3.17 MXCommon__FileResponse Struct Reference

Response containing a chunk of a file.

Data Fields

- struct `DefaultResponse sResponse`
return values.
- struct `ByteArray sArray`
Dynamic Array of byte.
- `xsd__unsignedLong ulEOF`
flag indicating end of file.

3.17.1 Field Documentation

3.17.1.1 struct DefaultResponse MXCommon__FileResponse::sResponse

3.17.1.2 struct ByteArray MXCommon__FileResponse::sArray

3.17.1.3 xsd__unsignedLong MXCommon__FileResponse::ulEOF

3.18 MXCommon__GetAutoConfigurationFileResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) sResponse

Default return values.

- struct [ByteArray](#) bArray

Array of byte of the file.

- [xsd__unsignedLong](#) ulEOF

End of file flag.

3.18.1 Field Documentation

3.18.1.1 struct DefaultResponse MXCommon__GetAutoConfigurationFileResponse::sResponse

3.18.1.2 struct ByteArray MXCommon__GetAutoConfigurationFileResponse::bArray

3.18.1.3 xsd__unsignedLong MXCommon__GetAutoConfigurationFileResponse::ulEOF

3.19 MXCommon__GetEthernetLinksStatesResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) sResponse

Default return values.

- struct [sGetEthernetLinksStatesPort](#) sPort0

- struct [sGetEthernetLinksStatesPort](#) sPort1

3.19.1 Field Documentation

3.19.1.1 struct `DefaultResponse MXCommon__GetEthernetLinksStatesResponse::sResponse`

3.19.1.2 struct `sGetEthernetLinksStatesPort MXCommon__GetEthernetLinksStatesResponse::sPort0`

3.19.1.3 struct `sGetEthernetLinksStatesPort MXCommon__GetEthernetLinksStatesResponse::sPort1`

3.20 MXCommon__GetHardwareTriggerFilterTimeResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
Default return values.
- `xsd__unsignedLong ulFilterTime`
Hardware filter time (step of 250ns).
- `xsd__unsignedLong ulInfo01`
Reserved.
- `xsd__unsignedLong ulInfo02`
Reserved.

3.20.1 Field Documentation

3.20.1.1 struct `DefaultResponse MXCommon__GetHardwareTriggerFilterTimeResponse::sResponse`

3.20.1.2 `xsd__unsignedLong MXCommon__GetHardwareTriggerFilterTimeResponse::ulFilterTime`

3.20.1.3 `xsd__unsignedLong MXCommon__GetHardwareTriggerFilterTimeResponse::ulInfo01`

3.20.1.4 `xsd__unsignedLong MXCommon__GetHardwareTriggerFilterTimeResponse::ulInfo02`

3.21 MXCommon__GetHardwareTriggerStateResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
Default return values.
- `xsd__unsignedLong ulState`

0 : Trigger input is low / 1 : Trigger input is high

- [xsd__unsignedLong ulInfo01](#)

Reserved.

- [xsd__unsignedLong ulInfo02](#)

Reserved.

3.21.1 Field Documentation

3.21.1.1 struct DefaultResponse MXCommon__GetHardwareTriggerStateResponse::sResponse

3.21.1.2 [xsd__unsignedLong](#) MXCommon__GetHardwareTriggerStateResponse::ulState

3.21.1.3 [xsd__unsignedLong](#) MXCommon__GetHardwareTriggerStateResponse::ulInfo01

3.21.1.4 [xsd__unsignedLong](#) MXCommon__GetHardwareTriggerStateResponse::ulInfo02

3.22 MXCommon__GetModuleTemperatureValueAndStatusResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) sResponse

Default return value.

- [xsd__double](#) dTemperatureValue

Temperature value.

- [xsd__unsignedLong](#) ulTemperatureStatus

Temperature status.

- [xsd__unsignedLong](#) ulInfo

Reserved.

3.22.1 Field Documentation

- 3.22.1.1 struct `DefaultResponse MXCommon__ - GetModuleTemperatureValueAndStatusResponse::sResponse`
- 3.22.1.2 `xsd__double MXCommon__ - GetModuleTemperatureValueAndStatusResponse::dTemperatureValue`
- 3.22.1.3 `xsd__unsignedLong MXCommon__ - GetModuleTemperatureValueAndStatusResponse::ulTemperatureStatus`
- 3.22.1.4 `xsd__unsignedLong MXCommon__ - GetModuleTemperatureValueAndStatusResponse::ulInfo`

3.23 MXCommon__GetTimeResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
Default return values.
- `xsd__unsignedLong ulLowTime`
Number of microseconds since the begin of the second.
- `xsd__unsignedLong ulHighTime`
Number of seconds since the Epoch (1st January,1970).

3.23.1 Field Documentation

- 3.23.1.1 struct `DefaultResponse MXCommon__GetTimeResponse::sResponse`
- 3.23.1.2 `xsd__unsignedLong MXCommon__GetTimeResponse::ulLowTime`
- 3.23.1.3 `xsd__unsignedLong MXCommon__GetTimeResponse::ulHighTime`

3.24 MXCommon__GetUpTimeResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
Default return value.
- `xsd__unsignedLong ulUpTime`
Reserved.

3.24.1 Field Documentation

3.24.1.1 struct DefaultResponse MXCommon__GetUpTimeResponse::sResponse

3.24.1.2 xsd__unsignedLong MXCommon__GetUpTimeResponse::ulUpTime

3.25 MXCommon__Response Struct Reference

contains return values

Data Fields

- [xsd__int iReturnValue](#)

return value of the call :

- 0 success
- -1 a system error occurred, the meaning of other values is function dependent and should be defined in the related header.

- [xsd__int syserrno](#)

system-error code (the value of the libc "errno" code, see [MXCommon__Strerror\(\)](#)).

3.25.1 Field Documentation

3.25.1.1 xsd__int MXCommon__Response::iReturnValue

3.25.1.2 xsd__int MXCommon__Response::syserrno

3.26 MXCommon__TestCustomerIDResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) sResponse

Default return values.

- struct [ByteArray](#) bValueArray

non encrypted value

- struct [ByteArray](#) bCryptedValueArray

encrypted value

3.26.1 Field Documentation

3.26.1.1 struct `DefaultResponse` `MXCommon__TestCustomerIDResponse::sResponse`

3.26.1.2 struct `ByteArray` `MXCommon__TestCustomerIDResponse::bValueArray`

3.26.1.3 struct `ByteArray` `MXCommon__TestCustomerIDResponse::bCryptedValueArray`

3.27 `MXCommon__unsignedLongResponse` Struct Reference

Response containing a numerical value (ex: return code).

Data Fields

- struct `DefaultResponse` `sResponse`

Default return values.

- `xsd__unsignedLong` `ulValue`

The meaning of this value is defined in the related header of the function who use this type.

3.27.1 Field Documentation

3.27.1.1 struct `DefaultResponse` `MXCommon__unsignedLongResponse::sResponse`

3.27.1.2 `xsd__unsignedLong` `MXCommon__unsignedLongResponse::ulValue`

3.28 `sGetEthernetLinksStatesPort` Struct Reference

Data Fields

- `xsd__unsignedLong` `ulState`
- `xsd__unsignedLong` `ulSpeed`
- `xsd__unsignedLong` `ulDuplex`
- `xsd__unsignedLong` `ulInfo1`
- `xsd__unsignedLong` `ulInfo2`

3.28.1 Field Documentation

3.28.1.1 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulState`

3.28.1.2 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulSpeed`

3.28.1.3 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulDuplex`

3.28.1.4 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulInfo1`

3.28.1.5 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulInfo2`

3.29 UnsignedLongArray Struct Reference

Dynamic Array of unsigned long.

Data Fields

- `xsd__unsignedLong * __ptr`
pointer of unsigned Long
- `int __size`
size of the unsigned Long array in Bytes
- `int __offset`
not used

3.29.1 Field Documentation

3.29.1.1 `xsd__unsignedLong* UnsignedLongArray::__ptr`

3.29.1.2 `int UnsignedLongArray::__size`

3.29.1.3 `int UnsignedLongArray::__offset`

3.30 UnsignedShortArray Struct Reference

Dynamic Array of unsigned short.

Data Fields

- `xsd__unsignedShort * __ptr`
pointer of unsigned short
- `int __size`
size of the unsigned short array in Bytes

- int [__offset](#)
not used

3.30.1 Field Documentation

3.30.1.1 `xsd__unsignedShort* UnsignedShortArray::__ptr`

3.30.1.2 `int UnsignedShortArray::__size`

3.30.1.3 `int UnsignedShortArray::__offset`

3.31 `xsd__base64Binary` Struct Reference

Dynamic Array of byte for input use.

Data Fields

- unsigned char * [__ptr](#)
pointer of byte
- int [__size](#)
size of the byte array

3.31.1 Field Documentation

3.31.1.1 `unsigned char* xsd__base64Binary::__ptr`

3.31.1.2 `int xsd__base64Binary::__size`

Chapter 4

File Documentation

4.1 MSXE3317_public_doc.h File Reference

Data Structures

- struct [xsd__base64Binary](#)
Dynamic Array of byte for input use.
- struct [UnsignedShortArray](#)
Dynamic Array of unsigned short.
- struct [UnsignedLongArray](#)
Dynamic Array of unsigned long.
- struct [ByteArray](#)
Dynamic Array of byte - encapsulates C-type strings.
- struct [DefaultResponse](#)
- struct [MXCommon__Response](#)
contains return values
- struct [MXCommon__ByteArrayResponse](#)
Response containing a C-type string.
- struct [MXCommon__FileResponse](#)
Response containing a chunk of a file.
- struct [MXCommon__unsignedLongResponse](#)
Response containing a numerical value (ex: return code).
- struct [sGetEthernetLinksStatesPort](#)
- struct [MXCommon__GetEthernetLinksStatesResponse](#)
- struct [MXCommon__GetModuleTemperatureValueAndStatusResponse](#)
- struct [MXCommon__GetHardwareTriggerFilterTimeResponse](#)
- struct [MXCommon__GetHardwareTriggerStateResponse](#)

- struct [MXCommon__TestCustomerIDResponse](#)
- struct [MXCommon__GetTimeResponse](#)
- struct [MXCommon__GetUpTimeResponse](#)
- struct [MXCommon__GetAutoConfigurationFileResponse](#)
- struct [MSXExxxx__Response](#)
- struct [MSXExxxx__unsignedLongResponse](#)
- struct [MSXExxxx__FileResponse](#)
- struct [MSXExxxx__unsignedLongTimeStampResponse](#)
- struct [MSXExxxx__AcquisitionGetChannelInfoResponse](#)
- struct [MSXExxxx__AcquisitionAutoRefreshGetValuesResponse](#)
- struct [MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam](#)
- struct [MSXExxxx__PressureGetConfigurationResponse](#)
- struct [MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse](#)
- struct [MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse](#)
- struct [MSXExxxx__SinCosInitResponse](#)
- struct [MSXExxxx__SinCosGetConfigurationResponse](#)
- struct [MSXExxxx__SinCosReadResponse](#)

Typedefs

- typedef char * [xsd__string](#)
encode xsd__string value as the xsd:string schema type
- typedef char [xsd__char](#)
encode xsd__string value as the xsd:char schema type
- typedef float [xsd__float](#)
encode xsd__float value as the xsd:float schema type
- typedef double [xsd__double](#)
encode xsd__double value as the xsd:double schema type
- typedef int [xsd__int](#)
encode xsd__int value as the xsd:int schema type
- typedef long [xsd__long](#)
encode xsd__long value as the xsd:long schema type
- typedef unsigned char [xsd__unsignedByte](#)
encode xsd__unsignedByte value as the xsd:unsignedByte schema type
- typedef unsigned int [xsd__unsignedInt](#)
encode xsd__unsignedInt value as the xsd:unsignedInt schema type
- typedef unsigned short int [xsd__unsignedShort](#)
encode xsd__unsignedShort value as the xsd:unsignedShort schema type
- typedef unsigned long [xsd__unsignedLong](#)
encode xsd__unsignedLong value as the xsd:unsignedLong schema type

Functions

- `int MXCommon__GetModuleType (void *__, struct MXCommon__ByteArrayResponse *Response)`
This function return the type of the MSX-E Module.
- `int MXCommon__GetHostname (void *__, struct MXCommon__ByteArrayResponse *Response)`
This function return the hostname of the MSX-E Module.
- `int MXCommon__SetHostname (struct xsd__base64Binary *bHostname, struct MXCommon__Response *Response)`
This function allows to set the hostname of the MSX-E Module.
- `int MXCommon__GetClientConnections (void *__, struct MXCommon__ByteArrayResponse *Response)`
This function return the client connection list.
- `int MXCommon__Sterror (xsd__int errnum, struct MXCommon__ByteArrayResponse *Response)`
Call the libc strerror() on the remote device (actually this is a call to strerror_r()).
- `int MXCommon__Reboot (void *__, struct MXCommon__Response *Response)`
Ask the MSX-E module to reboot.
- `int MXCommon__ResetAllIOFunctionalities (xsd__unsignedLong ulOption, struct MXCommon__Response *Response)`
Reset the I/O functionalities of the MSX-E system.
- `int MXCommon__DataseverRestart (xsd__unsignedLong ulAction, xsd__unsignedLong ulOption, struct MXCommon__Response *Response)`
Restart the data-server service.
- `int MXCommon__GetEthernetLinksStates (void *__, struct MXCommon__GetEthernetLinksStatesResponse *Response)`
Get MSX-E Ethernet links states.
- `int MXCommon__GetModuleTemperatureValueAndStatus (xsd__unsignedLong ulOption, struct MXCommon__GetModuleTemperatureValueAndStatusResponse *Response)`
Read the temperature on the module.
- `int MXCommon__SetModuleTemperatureWarningLevels (xsd__double dMinimalWarningLevel, xsd__double dMaximalWarningLevel, xsd__unsignedLong ulOption, struct MXCommon__Response *Response)`
Set the temperature warning level on the module.
- `int MXCommon__SetHardwareTriggerFilterTime (xsd__unsignedLong ulFilterTime, xsd__unsignedLong ulOption, struct MXCommon__Response *Response)`
Sets the filter time for the hardware trigger input in steps of 250 ns (max value: 65535).
- `int MXCommon__GetHardwareTriggerFilterTime (xsd__unsignedLong ulOption, struct MXCommon__GetHardwareTriggerFilterTimeResponse *Response)`

Get the filter time for the hardware trigger input.

- `int MXCommon__GetHardwareTriggerState (xsd__unsignedLong ulOption, struct MXCommon__GetHardwareTriggerStateResponse *Response)`

Get the hardware trigger state after the filter.

- `int MXCommon__SetCustomerKey (struct xsd__base64Binary *bKey, struct xsd__base64Binary *bPublicKey, struct MXCommon__Response *Response)`

Set the Customer key.

- `int MXCommon__TestCustomerID (void *_ , struct MXCommon__TestCustomerIDResponse *Response)`

Test the Customer ID (if the module has the right customer Key).

- `int MXCommon__SetTime (xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, struct MXCommon__Response *Response)`

Set the time on the module.

- `int MXCommon__SysToHardwareClock (void *_ , struct MXCommon__Response *Response)`

Set the hardware clock (if present) to the current system time.

- `int MXCommon__HardwareClockToSys (void *_ , struct MXCommon__Response *Response)`

Set the system time from the hardware clock (if present).

- `int MXCommon__GetTime (void *_ , struct MXCommon__GetTimeResponse *Response)`

Get the time on the module.

- `int MXCommon__GetUpTime (void *_ , struct MXCommon__GetUpTimeResponse *Response)`

Ask the MSX-E module uptime (number of seconds since the last boot).

- `int MXCommon__GetAutoConfigurationFile (void *_ , struct MXCommon__GetAutoConfigurationFileResponse *Response)`

Get the auto configuration file of the module.

- `int MXCommon__SetAutoConfigurationFile (struct xsd__base64Binary *ByteArrayInput, xsd__unsignedLong ulEOF, struct MXCommon__Response *Response)`

Set the auto configuration file of the module.

- `int MXCommon__StartAutoConfiguration (void *_ , struct MXCommon__ByteArrayResponse *Response)`

start/Restart the auto configuration

- `int MXCommon__InitAndStartSynchroTimer (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulReloadValue, xsd__unsignedLong ulNbrOfCycle, xsd__unsignedLong ulGenerateTriggerMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MXCommon__Response *Response)`

Initialises and starts the synchronisation timer of the module (not already available on all module).

- `int MXCommon__StopAndReleaseSynchroTimer (xsd__unsignedLong ulOption01, struct MXCommon__Response *Response)`

start/Restart the synchronisation timer (not already available on all module)

- `int MXCommon__GetConfigurationBackupFile (void *, struct MXCommon__FileResponse *Response)`
Download a configuration backup file from the module.
- `int MXCommon__ApplyConfigurationBackupFile (struct xsd__base64Binary *ByteArrayInput, xsd__unsignedLong ulEOF, struct MXCommon__Response *Response)`
Upload a new configuration on the module.
- `int MXCommon__ChangePassword (struct xsd__base64Binary *PreviousUser, struct xsd__base64Binary *PreviousPassword, struct xsd__base64Binary *NewUser, struct xsd__base64Binary *NewPassword, struct MXCommon__Response *Response)`
Set a new id/password.
- `int MXCommon__GetSubSystemState (xsd__unsignedLong SubsystemID, struct MXCommon__unsignedLongResponse *Response)`
Returns the current state of the specified sub-system.
- `int MXCommon__GetSubsystemIDFromName (struct xsd__base64Binary *SubsystemName, struct MXCommon__unsignedLongResponse *Response)`
Returns the ID of the sub-system of symbolic name "SubsystemName".
- `int MXCommon__GetStateIDFromName (xsd__unsignedLong SubsystemID, struct xsd__base64Binary *StateName, struct MXCommon__unsignedLongResponse *Response)`
Returns the ID of the state of symbolic name "StateName" of the sub-system of ID "SubsystemID".
- `int MXCommon__GetSubsystemNameFromID (xsd__unsignedLong SubsystemID, struct MXCommon__ByteArrayResponse *Response)`
Returns the symbolic name of the sub-system of numerical ID "SubsystemName".
- `int MXCommon__GetStateNameFromID (xsd__unsignedLong SubsystemID, xsd__unsignedLong StateID, struct MXCommon__ByteArrayResponse *Response)`
Returns the symbolic name of the state of numerical ID "StateID" of the sub-system of ID "SubsystemID".
- `int MXCommon__GetOptionInformation (void *, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__ByteArrayResponse *Response)`
Enables to get information about the options available on the system.
- `int MXCommon__SetToMaster (void *, xsd__unsignedLong ulState, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__Response *Response)`
Writes if the MSXE has to be always set to master The master mode (when enabled) make the system always detected as master.
- `int MXCommon__GetSynchronizationStatus (void *, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__unsignedLongResponse *Response)`
Reads the status of the synchronization for the corresponding MSXE The master mode (when enabled) make the system always detected as master.
- `int MSXExxxx__AcquisitionGetNumberOfChannels (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse *Response)`

Return the number of acquisition channels.

- int [MSXExxxx__AcquisitionGetChannelInfo](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, struct [MSXExxxx__AcquisitionGetChannelInfoResponse](#) *Response)

Return the selected acquisition channel type and hardware position.

- int [MSXExxxx__AcquisitionAutoRefreshInitAndStart](#) (xsd__unsignedLong ulChannelMask, xsd__unsignedLong ulAverageValue, xsd__unsignedLong ulRefreshTime, xsd__unsignedLong ulRefreshTimeUnit, xsd__unsignedLong ulTriggerMask, xsd__unsignedLong ulTriggerMode, xsd__unsignedLong ulHardwareTriggerEdge, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulByTriggerNbrOfSeqToAcquire, xsd__unsignedLong ulDataFormat, xsd__unsignedLong ulForceStart, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, xsd__unsignedLong ulOption3, struct [MSXExxxx__Response](#) *Response)

Starts an autorefresh acquisition using provided configuration.

- int [MSXExxxx__AcquisitionAutoRefreshGetValues](#) (xsd__unsignedLong ulBlocking, struct [MSXExxxx__AcquisitionAutoRefreshGetValuesResponse](#) *Response)

Reads the values acquired in auto refresh mode.

- int [MSXExxxx__AcquisitionAutoRefreshStopAndRelease](#) (void *_ , struct [MSXExxxx__Response](#) *Response)

Stops the current auto refresh acquisition.

- int [MSXExxxx__AcquisitionSequenceInitAndStart](#) (xsd__unsignedLong ulNbrOfChannel, struct [MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam](#) *psChannelList, xsd__unsignedLong ulAcquisitionTime, xsd__unsignedLong ulAcquisitionTimeUnit, xsd__unsignedLong ulNbrOfSequence, xsd__unsignedLong ulNbrMaxSequenceToTransfer, xsd__unsignedLong ulTriggerMask, xsd__unsignedLong ulTriggerMode, xsd__unsignedLong ulHardwareTriggerEdge, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulByTriggerNbrOfSeqToAcquire, xsd__unsignedLong ulDataFormat, xsd__unsignedLong ulForceStart, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, xsd__unsignedLong ulOption3, struct [MSXExxxx__Response](#) *Response)

Initialises and starts the sequence acquisition mode.

- int [MSXExxxx__AcquisitionSequenceStopAndRelease](#) (void *_ , struct [MSXExxxx__Response](#) *Response)

Stop and release the sequence acquisition mode.

- int [MSXExxxx__PressureGetNumberOfChannels](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)

Return the number of pressure channels.

- int [MSXExxxx__PressureSetChannelConfiguration](#) (xsd__unsignedLong ulChannel, xsd__double dSensorSensibility, xsd__double dSensorOffset, xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)

Pressure sensor configuration for the selected channel.

- int [MSXExxxx__PressureSetSamplingRate](#) (xsd__unsignedLong ulChannelGroup, xsd__unsignedLong ulBaseSamplingRate, xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)

Pressure acquisition sampling rate selection.

- int [MSXExxxx__PressureGetConfiguration](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, struct [MSXExxxx__PressureGetConfigurationResponse](#) *Response)
Get the selected pressure channel current configuration.
- int [MSXExxxx__DigitalIOGetNumberOfChannels](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Returns the number of digital I/O channels.
- int [MSXExxxx__DigitalIOGetNumberOfPorts](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Returns the number of digital I/O ports.
- int [MSXExxxx__DigitalIOGetPortAvailableDirections](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct [MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse](#) *Response)
Returns the available directions for the selected port (input or output).
- int [MSXExxxx__DigitalIOSetPortDirections](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulDirection, xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)
Write the current digital I/O direction for the selected port.
- int [MSXExxxx__DigitalIOGetPortDirections](#) (xsd__unsignedLong ulPort, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Reads the current digital I/O direction for the selected port.
- int [MSXExxxx__DigitalIOSetInputsFilterTime](#) (xsd__unsignedLong ulFilterTime, xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)
Sets the filter time for the digital inputs in steps of 250 ns (max value: 16777215).
- int [MSXExxxx__DigitalIOEnableDisableInputsFilter](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulFilter, xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)
Enables/disables the digital input filter for the selected port.
- int [MSXExxxx__DigitalIOGetInputsFilterConfiguration](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct [MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse](#) *Response)
Reads the digital inputs filter configuration for the selected port.
- int [MSXExxxx__DigitalIOTestOutputsShortCircuit](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Get the short-circuit status of the outputs of the selected port.
- int [MSXExxxx__DigitalIORearmOutputsShortCircuit](#) (xsd__unsignedLong ulOption1, struct [MSXExxxx__Response](#) *Response)
Rearm the digital outputs short circuit.
- int [MSXExxxx__DigitalIOTestOutputsPowerSupply](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct [MSXExxxx__unsignedLongResponse](#) *Response)
Reads the current power supply status of the selected port.

- `int MSXExxxx__DigitalIOReadChannel (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse *Response)`
Read the selected digital I/O channel.
- `int MSXExxxx__DigitalIOReadPort (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse *Response)`
Read the selected digital I/O port.
- `int MSXExxxx__DigitalIOWriteChannel (xsd__unsignedLong ulChannel, xsd__unsignedLong ulState, xsd__unsignedLong ulOption1, struct MSXExxxx__Response *Response)`
Set the selected digital output channel to on or off.
- `int MSXExxxx__DigitalIOWritePort (xsd__unsignedLong ulPort, xsd__unsignedLong ulState, xsd__unsignedLong ulOption1, struct MSXExxxx__Response *Response)`
Write a value to the selected digital I/O port.
- `int MSXExxxx__SinCosGetNumberOfChannels (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse *Response)`
Return the number of Sine/Cosine channels.
- `int MSXExxxx__SinCosInit (xsd__unsignedLong ulChannel, xsd__double dSignalPeriod, xsd__unsignedLong ulResolution, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXExxxx__SinCosInitResponse *Response)`
Initialise the selected Sine/Cosine channel.
- `int MSXExxxx__SinCosRelease (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response *Response)`
Release the selected Sine / Cosine channel.
- `int MSXExxxx__SinCosGetConfiguration (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption02, struct MSXExxxx__SinCosGetConfigurationResponse *Response)`
Get the current configuration of the selected Sine/Cosine channel.
- `int MSXExxxx__SinCosRead (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXExxxx__SinCosReadResponse *Response)`
Read the measured value of the selected Sine/Cosine channel.
- `int MSXExxxx__SinCosClear (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response *Response)`
Clear the measured value of the selected Sine/Cosine channel.
- `int MSXExxxx__SinCosClearError (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response *Response)`
Clear the error flag of the selected Sine/Cosine channel.
- `int MSXExxxx__SinCosInitIndex (xsd__unsignedLong ulChannel, xsd__unsignedLong ulEdge, xsd__unsignedLong ulAction, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response *Response)`

Initialise the Index configuration.

- int [MSXExxxx__SinCosReleaseIndex](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXExxxx__Response](#) *Response)

Release the Index configuration.

- int [MSXExxxx__SinCosInitAndEnableCompareLogic](#) (xsd__unsignedLong ulChannel, xsd__double dValue, xsd__unsignedLong ulValueFormat, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXExxxx__Response](#) *Response)

Initialise and enable a Sine/Cosine Compare Logic.

- int [MSXExxxx__SinCosDisableAndReleaseCompareLogic](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXExxxx__Response](#) *Response)

Disable and Release the Sine/Cosine Compare Logic.

4.1.1 Typedef Documentation

4.1.1.1 typedef char* xsd__string

4.1.1.2 typedef char xsd__char

4.1.1.3 typedef float xsd__float

4.1.1.4 typedef double xsd__double

4.1.1.5 typedef int xsd__int

4.1.1.6 typedef long xsd__long

4.1.1.7 typedef unsigned char xsd__unsignedByte

4.1.1.8 typedef unsigned int xsd__unsignedInt

4.1.1.9 typedef unsigned short int xsd__unsignedShort

4.1.1.10 typedef unsigned long xsd__unsignedLong

4.1.2 Function Documentation

4.1.2.1 int [MXCommon__GetModuleType](#) (void * _, struct [MXCommon__ByteArrayResponse](#) * *Response*)

Parameters

[in] _ : no input parameter

[out] *Response* • sArray : Module type string
 • sResponse Composed of iReturnValue and syserrno

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.2 `int MXCommon__GetHostname (void * _, struct MXCommon__ByteArrayResponse * Response)`

Parameters

[in] `_` : no input parameter
[out] *Response* • sArray : Hostname of the module

- iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
- syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.3 `int MXCommon__SetHostname (struct xsd__base64Binary * bHostname, struct MXCommon__Response * Response)`

Parameters

[in] *bHostname* : Hostname
[out] *Response* • iReturnValue : Return value

- 0 : success
- -1: system error (see syserrno)

- syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).
Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.4 `int MXCommon__GetClientConnections (void * _, struct MXCommon__ByteArrayResponse * Response)`

Parameters

[in] `_` : no input parameter
[out] *Response* • sArray : string containing the list of connected clients.

- sResponse Composed of iReturnValue and syserrno

The sArray string is of the form IP-Address:first connection-second connection---- IP-Address:first connection-second connection----

Sample: 172.16.3.43:8989-5555 172.16.3.200:8989

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.5 int MXCommon__Strerror (xsd__int errnum, struct MXCommon__ByteArrayResponse * Response)

Usually SOAP functions return this value in a variable named syserror, which is meaningful only when the function return value, usually called iReturnValue, indicate an error (that is, have a value of -1 or -100, depending of the case).

Parameters

- [in] **errnum** : Error number
- [out] **Response** • sArray : See the description below.
- sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno).
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

STRERROR(3)
 STRERROR(3)

Linux Programmer's Manual

NAME

strerror, strerror_r - return string describing error code

SYNOPSIS

```
#include <string.h>
```

```
char *strerror(int errnum);
```

```
#define _XOPEN_SOURCE 600
#include <string.h>
```

```
int strerror_r(int errnum, char *buf, size_t n);
```

DESCRIPTION

The `strerror()` function returns a string describing the error code passed in the argument `errnum`, possibly using the `LC_MESSAGES` part of the current locale to select the appropriate language. This string must not be modified by the application, but may be modified by a subsequent call to `perror()` or `strerror()`. No library function will modify this string.

The `strerror_r()` function is similar to `strerror()`, but is thread safe. It returns the string in the user-supplied buffer `buf` of length `n`.

RETURN VALUE

The `strerror()` function returns the appropriate error description string, or an unknown error message if the error code is unknown. The value of `errno` is not changed for a successful call, and is set to a non-zero value upon error.

The `strerror_r()` function returns 0 on success and -1 on failure, setting `errno`.

ERRORS

`EINVAL` The value of `errnum` is not a valid error number.

`ERANGE` Insufficient storage was supplied to contain the error description string.

CONFORMING TO

SVID 3, POSIX, 4.3BSD, ISO/IEC 9899:1990 (C89).

`strerror_r()` with prototype as given above is specified by SUSv3, and was in use under Digital Unix and HP Unix. An incompatible function, with prototype

```
char *strerror_r(int errnum, char *buf, size_t n);
```

is a GNU extension used by glibc (since 2.0), and must be regarded as obsolete in view of SUSv3.

The GNU version may, but need not, use the user-supplied buffer.

If it does, the result may be truncated in case the supplied buffer is too small.

The result is always NUL-terminated.

SEE ALSO

`errno(3)`, `perror(3)`, `strsignal(3)`

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.2.6 int MXCommon__Reboot (void * _, struct MXCommon__Response * *Response*)

Parameters

[in] `_` : no input parameter

[out] ***Response*** • `iReturnValue` : Return value

– 0 : success

– -1 : system error (see `syserrno`)

• `syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.2.7 int MXCommon__ResetAllIOFunctionalities (xsd__unsignedLong *ulOption*, struct MXCommon__Response * *Response*)

The behavior of the function depends on the MSX-E system that is used.

On MSX-E3511: Stop the watchdogs and stop the generators

On MSX-E3601: Stop the sequence acquisition and stop the calibration

On MSX-E3701: Stop the acquisition

Parameters

[in] ***ulOption*** Reserved. Set to 0

[out] **Response** *iReturnValue*

- **0** The remote function performed OK
- **-1** Internal system error occurred. See value of `syserrno`
- **-100** Function not supported by the system

syserrno system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.8 int MXCommon__DataserverRestart (xsd__unsignedLong ulAction, xsd__unsignedLong ulOption, struct MXCommon__Response * Response)

Parameters

[in] *ulAction* : action

- 0: normal restart
- 1: with cache file reset
- 2: with cache file deletion

[in] *ulOption* : Reserved

[out] **Response** • *iReturnValue* : Return value

- 0 : success
- -1: system error (see `syserrno`)
- `syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

Note

(revision>6386) Depending on the system type, can be used to restart the data-recv service as well. In this case, parameter action is ignored.

4.1.2.9 int MXCommon__GetEthernetLinksStates (void * _, struct MXCommon__GetEthernetLinksStatesResponse * Response)

Parameters

[in] *_* : no input parameter

[out] **Response** Structure that contains the MSX-E Ethernet links states and errors:

sResponse.iReturnValue

- **0** The remote function performed OK
- **-1** System error occurred
- **-2** Fail to get Ethernet links states
- **-100** Internal system error occurred. See value of `syserrno`

sResponse.syserrno system error code (the value of the libc "errno" code)

sPort0: Fisrt port informations

- **ulState**
 - 0 Link down
 - 1 Link up
- **ulSpeed**
 - 10 10 Mb/s
 - 100 100 Mb/s
- **ulDuplex**
 - 0 Half duplex
 - 1 Full duplex
- **ulInfo1** Reserved
- **ulInfo2** Reserved

sPort1: Second port informations

- **ulState**
 - 0 Link down
 - 1 Link up
- **ulSpeed**
 - 10 10 Mb/s
 - 100 100 Mb/s
- **ulDuplex**
 - 0 Half duplex
 - 1 Full duplex
- **ulInfo1** Reserved
- **ulInfo2** Reserved

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.10 `int MXCommon__GetModuleTemperatureValueAndStatus (xsd__unsignedLong
ulOption, struct MXCommon__GetModuleTemperatureValueAndStatusResponse *
Response)`

Parameters

[in] *ulOption* : Reserved

[out] *Response* • sResponse.iReturnValue : Return value

- 0 : success
- -1: system error (see syserrno)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - dValue : Temperature value in Degree Celsius
- ulTemperatureStatus : Temperature Status :
 - TEMPERATURE_INITIAL = 0 : Temperature not ready
 - TEMPERATURE_TOOLOW = 1 : Temperature too low !

- TEMPERATURE_LOW = 2 : Temperature under the min warning value
- TEMPERATURE_NOMINAL = 3 : Temperature in the nominal range
- TEMPERATURE_HIGH = 4 : Temperature over the max warning value
- TEMPERATURE_TOOHIGH = 5 : Temperature too high !

- ulInfo : Reserved

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.11 `int MXCommon__SetModuleTemperatureWarningLevels (xsd__double dMinimalWarningLevel, xsd__double dMaximalWarningLevel, xsd__unsignedLong ulOption, struct MXCommon__Response * Response)`

Parameters

- [in] **dMinimalWarningLevel** : Minimal temperature warning level in Degree : 5 to 60 Degree Celsius
- [in] **dMaximalWarningLevel** : Maximal temperature warning level in Degree : 5 to 60 Degree Celsius
- [in] **ulOption** : Reserved
- [out] **Response** • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.12 `int MXCommon__SetHardwareTriggerFilterTime (xsd__unsignedLong ulFilterTime, xsd__unsignedLong ulOption, struct MXCommon__Response * Response)`

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

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

Parameters

- [in] **ulFilterTime** Filter time for the hardware trigger input in steps of 250ns (max value : 65535).
- **0**: Disable the filter
 - **1**: Sets the filter time to 250 ns
 - **2**: Sets the filter time to 500 ns
 - ...
 - **65535**: Sets the filter time to 16 ms

[in] *ulOption* Reserved. Set to 0

[out] *Response* Response of the system

- *sResponse.iReturnValue*
 - 0: The remote function performed OK
 - -1: Internal system error occurred. See value of *syserrno*
- *sResponse.syserrno* system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.13 int MXCommon__GetHardwareTriggerFilterTime (xsd__unsignedLong *ulOption*, struct MXCommon__GetHardwareTriggerFilterTimeResponse * *Response*)

Get 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

[in] *ulOption* Reserved. Set to 0

[out] *Response* Response of the system

- *ulFilterTime* filter time for the hardware trigger input
 - 0: filter disabled
 - 1: filter of 250ns
 - 2: filter of 500ns
 - ...
 - 65535: filter of 16ms
- *sResponse.iReturnValue*
 - 0: The remote function performed OK
 - -1: Internal system error occurred. See value of *syserrno*
- *sResponse.syserrno* system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.14 int MXCommon__GetHardwareTriggerState (xsd__unsignedLong *ulOption*, struct MXCommon__GetHardwareTriggerStateResponse * *Response*)

Parameters

[in] *ulOption* : Reserved

[out] *Response* • *ulState* : Hardware trigger input state.

- 0: Hardware trigger input is low
- 1: Hardware trigger input is high.
- *sResponse.iReturnValue* : Return value

- 0 : success
- -1: system error (see `syserrno`)
- `sResponse.syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.15 `int MXCommon__SetCustomerKey (struct xsd__base64Binary * bKey, struct xsd__base64Binary * bPublicKey, struct MXCommon__Response * Response)`

Parameters

- [in] *bKey* : Customer key (only writable on the module) [32 bytes containing a AES key]
- [in] *bPublicKey* : IV (Initialisation vector) for the AES cryptography [16 bytes containing a AES key]
- [out] *Response* • `sResponse.iReturnValue` : Return value
- 0 : success
 - -1: system error (see `syserrno`)
 - `sResponse.syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.16 `int MXCommon__TestCustomerID (void * _, struct MXCommon__TestCustomerIDResponse * Response)`

Parameters

- [in] `_` : No Input
- [out] *Response* • `sResponse.iReturnValue` : Return value
- 0 : success
 - -1: system error (see `syserrno`)
 - `sResponse.syserrno` : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - `bValueArray` : non encrypted value array [16 bytes of random data]
 - `bCryptedValueArray` : Encrypted value array [16 bytes of the encrypted random data]

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.17 `int MXCommon_SetTime (xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, struct MXCommon_Response * Response)`

Parameters

- [in] *ulLowTime* : Number of microseconds since the begin of the second
- [in] *ulHighTime* : Number of seconds since the Epoch (1st January,1970)
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon_Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.18 `int MXCommon_SysToHardwareClock (void * _, struct MXCommon_Response * Response)`

Parameters

- [in] *_* No input parameter
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon_Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

If this function fails, it means the module does not have a hardware RTC, or the hardware is not functional. Check the "hwclock" subsystem status.

4.1.2.19 `int MXCommon_HardwareClockToSys (void * _, struct MXCommon_Response * Response)`

When the hardware clock is present, the system time is automatically set to it when the module becomes master on the inter-module synchronisation bus.

Parameters

- [in] *_* No input parameter
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)

- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

If this function fails, it means the module does not have a hardware RTC, or the hardware is not functional. Check the "hwclock" subsystem status.

4.1.2.20 int MXCommon__GetTime (void * _, struct MXCommon__GetTimeResponse * Response)

Parameters

[in] _ : No input parameter

[out] *Response* • sResponse.iReturnValue : Return value

– 0 : success

– -1: system error (see syserrno)

- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

- ulLowTime : Number of microseconds since the begin of the second

- ulHighTime : Number of seconds since the Epoch (1st January,1970)

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.2.21 int MXCommon__GetUpTime (void * _, struct MXCommon__GetUpTimeResponse * Response)

Parameters

[in] _ : no input parameter

[out] *Response* • sResponse.iReturnValue : Return value

– 0 : success

– -1: system error (see syserrno)

- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

- ulUpTime : Number of seconds since the last boot of the system.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.2.22 int MXCommon__GetAutoConfigurationFile (void * __, struct MXCommon__GetAutoConfigurationFileResponse * Response)

Parameters

- [in] *__* : No input parameter
- [out] **Response** • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
 - -100 : Error of the read of the auto configuration file
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - bArray : Array of Bytes of the file
 - ulEOF : End of file flag

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.23 int MXCommon__SetAutoConfigurationFile (struct xsd__base64Binary * ByteArrayInput, xsd__unsignedLong ulEOF, struct MXCommon__Response * Response)

Parameters

- [in] *ByteArrayInput* : Array of Bytes of the file
- [in] *ulEOF* : End of file flag
- [out] **Response** • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.24 int MXCommon__StartAutoConfiguration (void * __, struct MXCommon__ByteArrayResponse * Response)

Parameters

- [in] *__* : No input parameter
- [out] **Response** • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

- sArray : message returned by the auto configuration start

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.25 `int MXCommon__InitAndStartSynchroTimer (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulReloadValue, xsd__unsignedLong ulNbrOfCycle, xsd__unsignedLong ulGenerateTriggerMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MXCommon__Response * Response)`

Parameters

- [in] *ulTimeBase* : Time base of the timer (0 for us, 1 for ms, 2 for s)
- [in] *ulReloadValue* : Timer reload value (0 to 0xFFFF), minimum reload time is 5 us
- [in] *ulNbrOfCycle* : Number of timer cycle
 - 0: continuous
 - > 0: defined number of cycle
- [in] *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
- [in] *ulOption01* : Define the source of the trigger
 - 0 : Trigger disabled
 - 1 : Enable the hardware digital input trigger
- [in] *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)
- [in] *ulOption03* : Define the number of trigger events before the action occur
 - 1 : all trigger event start the action
 - max value : 65535
- [in] *ulOption04* : Reserved
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0 : success
 - -1: system error (see syserrno)
 - -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
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.26 `int MXCommon__StopAndReleaseSynchroTimer (xsd__unsignedLong ulOption01, struct MXCommon__Response * Response)`

Parameters

- [in] *ulOption01* : Reserved
- [out] *Response* • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno)
 - -100: Start/Stop timer error
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.27 `int MXCommon__GetConfigurationBackupFile (void * _, struct MXCommon__FileResponse * Response)`

Parameters

- [in] *_* : No input parameter
- [out] *Response* • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error (see syserrno) (see syserrno)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
- bArray : Array of Bytes of the file
- ulEOF : End of file flag

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

This function is designed to be called repeatedly until no more data is available. At this point the flag ulEOF is set.

Below is an example in pseudo-C.

```
int dummy;
struct MXCommon__FileResponse Response;
while(1)
{
```

```

if ( MXCommon__GetConfigurationBackupFile(&dummy, &Response) != SOAP_OK)
{
    // handle soap error
}
if (Response.iReturnValue)
{
    // handle remote error (Response.syserrno contains more information)
}
// do something with the data, for example save it in a file
write(fd, Response.bArray.__ptr, Response.bArray.__size);
// if this is the end of the file, quit the loop
if(Response.ulEOF)
break;
}
*

```

4.1.2.28 `int MXCommon__ApplyConfigurationBackupFile (struct xsd__base64Binary * ByteArrayInput, xsd__unsignedLong ulEOF, struct MXCommon__Response * Response)`

Parameters

- [in] *ByteArrayInput* : Array of Bytes of the file
- [in] *ulEOF* : End of file flag
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0: success
 - -1: system error (see syserrno)
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

This function is designed to be called repeatedly until all data is transfered. At this point the flag ulEOF must be set to 1. The new configuration is then applied.

4.1.2.29 `int MXCommon__ChangePassword (struct xsd__base64Binary * PreviousUser, struct xsd__base64Binary * PreviousPassword, struct xsd__base64Binary * NewUser, struct xsd__base64Binary * NewPassword, struct MXCommon__Response * Response)`

The changes are immediately active.

Parameters

- [in] *_* : No input parameter
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0: success
 - -1: string PreviousUser is invalid
 - -2: string PreviousPassword is invalid
 - -3: string NewUser is invalid
 - -4: string NewPassword is invalid

- -5: authentication failed
- -100: system error while saving tokens (use syserrno for more information)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
- sArray : message returned by the auto configuration start

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

Warning

The parameters transit in clear text. Use this functionality only on trusted networks.
 Given that ADDI-DATA GmbH takes security seriously, there is no way to change the password without knowing it. No "hidden back-door". This function makes it all too easy to lock a module, if you don't remember the password you set on it.

4.1.2.30 int MXCommon__GetSubSystemState (xsd__unsignedLong SubsystemID, struct MXCommon__unsignedLongResponse * Response)

Parameters

- [in] *SubsystemID* sub-system numerical ID
- [out] *Response* • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error while executing the request (see syserrno)
 - -2: invalid parameter SubsystemID
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - Value The state of the sub-system "Id" at the moment of the execution of the request.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.31 int MXCommon__GetSubsystemIDFromName (struct xsd__base64Binary * SubsystemName, struct MXCommon__unsignedLongResponse * Response)

Parameters

- [in] *SubsystemName* sub-system symbolic name.
- [out] *Response* • sResponse.iReturnValue :Return value
- 0 : success
 - -1: system error while executing the request (see syserrno)
 - -2: invalid parameter SubsystemName
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
 - Value The numerical ID of the sub-system "SubsystemName".

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.32 `int MXCommon__GetStateIDFromName (xsd__unsignedLong SubsystemID, struct xsd__base64Binary * StateName, struct MXCommon__unsignedLongResponse * Response)`

Parameters

- [in] *SubsystemID* sub-system numerical ID
- [in] *StateName* state symbolic name.
- [out] *Response* • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error while executing the request (see syserrno)
 - -2: invalid parameters SubsystemID or StateName
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
- Value The numerical ID of the state "StateName".

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.33 `int MXCommon__GetSubsystemNameFromID (xsd__unsignedLong SubsystemID, struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] *SubsystemID* sub-system numerical ID.
- [out] *Response* • sResponse.iReturnValue : Return value
- 0 : success
 - -1: system error while executing the request (see syserrno)
 - -2: invalid parameter SubsystemName
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Sterror\(\)](#).
- sArray : The symbolic name associated with the ID.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.34 `int MXCommon__GetStateNameFromID (xsd__unsignedLong SubsystemID, xsd__unsignedLong StateID, struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] *SubsystemID* sub-system numerical ID.
- [in] *StateID* sub-system numerical ID.
- [out] *Response*
 - sResponse.iReturnValue : Return value
 - 0 success
 - -1 system error while executing the request (see syserrno)
 - -2 invalid parameters SubsystemID or StateID
 - sResponse.syserrno : System-error code. The value of the libc "errno" code, see [MXCommon__Strerror\(\)](#).
 - sArray The symbolic name associated with the state numerical ID.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.35 `int MXCommon__GetOptionInformation (void * _, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__ByteArrayResponse * Response)`

Parameters

- [in] *ulOption01*,: not used, set it to 0
- [in] *ulOption02*,: not used, set it to 0
- [out] *Response*
 - sArray : Option information string
 - sResponse Composed of iReturnValue and syserrno

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.2.36 `int MXCommon__SetToMaster (void * _, xsd__unsignedLong ulState, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__Response * Response)`

Parameters

- [in] *ulState* State of the supermaster mode
 - **0** automatic mode (default). The state of the system (master or slave) will be automatically detected by the system
 - **1** Set to master mode at all time. The system will always be detected as master
- [in] *ulOption01* Reserved. Set to 0
- [in] *ulOption02* Reserved. Set to 0
- [out] *Response iReturnValue*
 - **0** The remote function performed OK

- **-1** System error occurred
- **-2** The PLD is not working
- **-3** The ulFilterTime parameter is wrong
- **-100** Internal system error occurred. See value of syserrno *syserrno* system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.37 `int MXCommon__GetSynchronizationStatus (void * _, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MXCommon__unsignedLongResponse * Response)`

Parameters

[in] *ulOption01* Reserved. Set to 0

[in] *ulOption02* Reserved. Set to 0

[out] *Response sResponse.iReturnValue*

- **0** The remote function performed OK
- **-1** System error occurred
- **-2** The PLD is not working
- **-100** Internal system error occurred. See value of syserrno

sResponse.syserrno system error code (the value of the libc "errno" code)

ulValue State of the supermaster mode

- **0** Automatic mode (default). The state of the system (master or slave) will be automatically detected by the system
- **1** MSXE is always set as a master. The system will always be detected as master

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.38 `int MSXExxxx__AcquisitionGetNumberOfChannels (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Parameters

[in] *ulOption1* : Reserved. Set to 0

[out] *Response* :

sResponse.iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -100: Internal system error occurred. See value of syserrno

sResponse.syserrno : system-error code (the value of the libc "errno" code)

ulValue : Number of available acquisition channels

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

4.1.2.39 `int MSXExxxx__AcquisitionGetChannelInfo (xsd__unsignedLong ulChannel,
xsd__unsignedLong ulOption1, struct MSXExxxx__AcquisitionGetChannelInfoResponse
* Response)`

Parameters

[in] *ulChannel* : Selected acquisition channel

[in] *ulOption1* : Reserved. Set to 0

[out] *Response* :

sResponse.iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Channel selection wrong
- -100: Internal system error occurred. See value of syserrno

sResponse.syserrno : system-error code (the value of the libc "errno" code)

ulType : Acquisition channel type

- 0 : Not available
- 1 : Temperature channel
- 2 : Pressure channel
- 3 : Transducer channel
- 4 : Analog input channel
- 5 : Analog input ICP channel
- 6 : Digital I/O port

ulHwPosition : Hardware position index (0 to 7) *ulChannelIndex* : Return the functionality channel index

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

4.1.2.40 `int MSXExxxx__AcquisitionAutoRefreshInitAndStart (xsd__unsignedLong
ulChannelMask, xsd__unsignedLong ulAverageValue, xsd__unsignedLong
ulRefreshTime, xsd__unsignedLong ulRefreshTimeUnit, xsd__unsignedLong
ulTriggerMask, xsd__unsignedLong ulTriggerMode, xsd__unsignedLong
ulHardwareTriggerEdge, xsd__unsignedLong ulHardwareTriggerCount,
xsd__unsignedLong ulByTriggerNbrOfSeqToAcquire, xsd__unsignedLong ulDataFormat,
xsd__unsignedLong ulForceStart, xsd__unsignedLong ulOption1, xsd__unsignedLong
ulOption2, xsd__unsignedLong ulOption3, struct MSXExxxx__Response * Response)`

Parameters

[in] *ulChannelMask* Mask of the channel to acquire by the auto refresh (1 bit = 1 Channel). 0 for all available acquisition channels

- [in] ***ulAverageValue*** Set the average value :
- 1 : not used
 - max value : 255
- [in] ***ulRefreshTimeUnit*** Refresh Time Unit
- 0 : microsecond
 - 1 : millisecond
 - 2 : second
- [in] ***ulRefreshTime*** Refresh Time
- range from min 1000 to 65535 when the unit is the microsecond
 - range from min 1 to 65535 when the unit is the millisecond
 - range from min 1 to 65535 when the unit is the second
- [in] ***ulTriggerMask*** Define the source of the trigger
- 0 : trigger disabled
 - 1 : Enable Hardware Digital Input Trigger
 - 2 : Enable Synchro Trigger
 - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/-Cosine input)
 - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- [in] ***ulTriggerMode*** Define the trigger mode
- 1 : One shot trigger
 - 2 : Sequence trigger
- [in] ***ulHardwareTriggerEdge*** Define the edge of the hardware trigger who generates a trigger action
- 1 : rising edge (Only if hardware trigger selected)
 - 2 : falling edge (Only if hardware trigger selected)
 - 3 : Both front (Only if hardware trigger selected)
- [in] ***ulHardwareTriggerCount*** Define the number of trigger events before the action occur
- 1 : all trigger event start the action
 - max value : 65535
- [in] ***ulByTriggerNbrOfSeqToAcquire*** Define the number of sequence to acquire by each trigger event
- 0 : continuous mode
 - <> 0 : number of sequence : (1..0xFFFFFFFF)
- D0 : Absolute Time stamp information
- [in] ***ulDataFormat*** • 0 : no time stamp information
- 1 : time stamp information
- D1 : Relative Time stamp information
- 0 : no time stamp information
 - 1 : time stamp information
- D2 : Auto refresh counter information
- 0 : No auto refresh counter information
 - 1 : Auto refresh counter information
- D3 : System status information
- 0 : No system status information required

- 1 : System status information required

D4 : Data format

- 0: Digital value (see technical description)
- 1: Analog value (see technical description)

You can not select both absolute and relative time stamp simultaneously

[in] **ulForceStart** :

- 0 : Function return a error if any acquisition already in progress
- 1 : If any acquisition in progress then stop this

[in] **ulOption1** Reserved. Set to 0

[in] **ulOption2** Reserved. Set to 0

[in] **ulOption3** Reserved. Set to 0

[out] **Response** :

iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Any acquisition already in progress
- -3: Any selected channel not OK, call the diagnostic function for more information
- -4: Channel Mask error
- -5: Not available average value
- -6: Not available refresh time unit
- -7: The minimal refresh time is 1000 us !
- -8: The maximal refresh time is 65535 !
- -9: Trigger mask not available
- -10: Trigger mask : 2 different trigger source cannot be simultaneously be activated
- -11: Trigger mode not available
- -12: Trigger mask : 2 trigger mode cannot be simultaneously activated
- -13: Hardware trigger : front definition error
- -14: Hardware trigger count value not available
- -15: Nbr of sequence to acquire by trigger mode not available
- -16: Data format not available
- -17: Selected channels combination not available
- -100: Kernel function error

syserrno : system-error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

4.1.2.41 `int MSXExxxx__AcquisitionAutoRefreshGetValues (xsd__unsignedLong ulBlocking, struct MSXExxxx__AcquisitionAutoRefreshGetValuesResponse * Response)`

Reads the values acquired in auto refresh mode.

Parameters

[in] **ulBlocking** Wait a new value or read the actual value

- **0**: Get the current auto refresh values
- **1**: Wait a new auto refresh value cycle

[out] **Response** Response of the system

- **iReturnValue**
 - **0**: The remote function performed OK
 - **-1**: Means an system error occurred
 - **-2**: No Acquisition in progress
 - **-3**: 2s timeout occurred. This if you have enabled the blocking mode.
 - **-4**: 2s timeout occurred. This if you do not have enabled the blocking mode, and if the first value is not available.
 - **-100**: Internal system error occurred. See value of syserrno
- **syserrno** system error code (the value of the libc "errno" code)
- **ulTimeStampLow** number of microseconds since the Epoch
- **ulTimeStampHigh** number of seconds since the Epoch
- **ulCounterValue** counter value
- **dValue** Array that contains the channels values
 - **dValue[0]** Value of channel 0
 - ...
 - **dValue[15]** Value of channel 15

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.42 int MSXExxxx__AcquisitionAutoRefreshStopAndRelease (void * __, struct MSXExxxx__Response * *Response*)

Parameters

[in] **_** Dummy parameter

[out] **Response** :

iReturnValue :

- **0**: Means the remote function performed OK
- **-1**: Means an system error occurred
- **-100**: Kernel function error

syserrno : system-error code (the value of the libc "errno" code)

Returns

- **0**: SOAP_OK
- **<> 0**: See SOAP error

Must be called before any another call to MSXExxxx__AcquisitionAutoRefreshInitAndStart.

4.1.2.43 `int MSXExxxx__AcquisitionSequenceInitAndStart (xsd__unsignedLong ulNbrOfChannel, struct MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam * psChannelList, xsd__unsignedLong ulAcquisitionTime, xsd__unsignedLong ulAcquisitionTimeUnit, xsd__unsignedLong ulNbrOfSequence, xsd__unsignedLong ulNbrMaxSequenceToTransfer, xsd__unsignedLong ulTriggerMask, xsd__unsignedLong ulTriggerMode, xsd__unsignedLong ulHardwareTriggerEdge, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulByTriggerNbrOfSeqToAcquire, xsd__unsignedLong ulDataFormat, xsd__unsignedLong ulForceStart, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, xsd__unsignedLong ulOption3, struct MSXExxxx__Response * Response)`

Initialises and starts the sequence acquisition mode.

Parameters

- [in] *ulNbrOfChannel* : Nbr of channel in the sequence
- [in] *psChannelList* : List of the channel who compose the sequence.
- [in] *ulAcquisitionTime* : Acquisition Time
 - range from min 1000 to 65535 when the unit is the microsecond
 - range from min 1 to 65535 when the unit is the millisecond
 - range from min 1 to 65535 when the unit is the second
- [in] *ulAcquisitionTimeUnit* : Acquisition Time Unit
 - 0 : us
 - 1 : ms
 - 2 : s
- [in] *ulNbrOfSequence* : Number of sequence to acquire :
 - 0 : continuous mode
 - <> 0 : number of sequence
- [in] *ulNbrMaxSequenceToTransfer* : Max nbr of sequence to acquire before a data transfer : (1,4096)
- [in] *ulTriggerMask* : Define the source of the trigger
 - 0 : trigger disabled
 - 1 : Enable Hardware Digital Input Trigger
 - 2 : Enable Synchro Trigger
 - 4 : Enable Compare Trigger (only useful if your system has incremental counter or Sine/-Cosine input)
 - 8 : Enable Index Trigger (only useful if your system has Sine/Cosine input)
- [in] *ulTriggerMode* : Define the trigger mode
 - 1 : One shot trigger
 - 2 : Sequence trigger
- [in] *ulHardwareTriggerEdge* : Define the edge of the hardware trigger who generate a trigger action
 - 1 : rising front (Only if hardware trigger selected)
 - 2 : falling front (Only if hardware trigger selected)
 - 3 : Both front (Only if hardware trigger selected)
- [in] *ulHardwareTriggerCount* : Define the number of trigger events before the action occur

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

[in] ***ulByTriggerNbrOfSeqToAcquire*** : define the number of sequence to acquire by each trigger event

- 0 : continuous mode
- <> 0 : number of sequence : (1..0xFFFFFFFF)

[in] ***ulDataFormat*** : Data format option

D0 : Absolute Time stamp information

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

D1 : Relative Time stamp information

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

D2 : Sequence counter information

- 0 : No sequence counter information
- 1 : Sequence counter information

D3 : System status information

- 0 : No system status information required
- 1 : System status information required

D4 : Data format

- 0: Digital value (see technical description)
- 1: Analog value (see technical description)

You can not select both absolute and relative time stamp simultaneously

[in] ***ulForceStart*** :

- 0 : Function return a error if any acquisition already in progress
- 1 : If any acquisition in progress then stop this

[in] ***ulOption1*** : Reserved. Set to 0

[in] ***ulOption2*** : Reserved. Set to 0

[in] ***ulOption3*** : Reserved. Set to 0

[out] ***Response*** :

iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Any acquisition already in progress
- -3: The nbr of channel in the sequence is null or too high
- -4: Channel index selection error
- -5: Channel already selected
- -6: Any selected channel not OK, call the diagnostic function for more information
- -7: Not available acquisition time unit
- -8: The minimal acquisition time is 1000 us !
- -9: The maximal acquisition time is 65535 !
- -10: Transfer sequence size error (1 to 4096) !
- -11: The total number of sequences is not a multiple from number of sequences to transfer

- -12: Trigger mask not available
- -13: Trigger mask : 2 different trigger source cannot be simultaneously be activated
- -14: Trigger mode not available
- -15: Trigger mask : 2 trigger mode cannot be simultaneously be activated
- -16: Hardware trigger : front definition error
- -17: Hardware trigger count value not available
- -18: Nbr of sequence to acquire by trigger mode not available
- -19: Data format not available
- -20: Selected channels combination not available
- -100: Start sequence kernel function error

syserrno : system-error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.44 `int MSXExxxx__AcquisitionSequenceStopAndRelease (void * _, struct MSXExxxx__Response * Response)`

Parameters

[in] **_** : no input parameter

[out] **Response** :

iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: No sequence acquisition in progress
- -100: Kernel function error

syserrno : system-error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

4.1.2.45 `int MSXExxxx__PressureGetNumberOfChannels (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Parameters

[in] **ulOption1** : Reserved. Set to 0

[out] **Response** :

sResponse.iReturnValue :

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

sResponse.syserrno : system-error code (the value of the libc "errno" code)

ulValue : Number of available pressure channels

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

4.1.2.46 `int MSXExxxx_PressureSetChannelConfiguration (xsd__unsignedLong ulChannel, xsd__double dSensorSensibility, xsd__double dSensorOffset, xsd__unsignedLong ulOption1, struct MSXExxxx_Response * Response)`

Remarks

For MSXE with pressure functionality :

- Before revision 6982 the *dSensorOffset* parameter is given in mV.

$$\text{Pressure value (in Unit)} = (\text{AnalogValue (mV)} - \text{dSensorOffset (mV)}) / (\text{BridgeSupply (V)} * \text{dSensorSensibility(mV / V / Unit)})$$
- Since revision 6982 the *dSensorOffset* parameter is given in Unit instead of mV.

$$\text{Pressure value (in Unit)} = (\text{AnalogValue (mV)}) / (\text{BridgeSupply (V)} * \text{dSensorSensibility(mV / V / Unit)}) - \text{dSensorOffset (Unit)}$$

Parameters

- [in] *ulChannel* : Channel selection (0 to 15 or 255 for all channels)
- [in] *dSensorSensibility* : Sensor sensibility (mV/V/bar or mV/V/mbar or mV/V/Pa, ...). Refer to sensor documentation
- [in] *dSensorOffset* : Sensor offset in unit (bar/Newton/Pa/Psi...) that depends on the sensor Refer to sensor documentation
- [out] *Response* :
- iReturnValue* :
- 0: Means the remote function performed OK
 - -1: Means an system error occurred
 - -2: Channel selection wrong
 - -3: Sensor sensibility selection wrong
 - -4: Acquisition in progress. Can not change the configuration

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

4.1.2.47 `int MSXExxxx_PressureSetSamplingRate (xsd__unsignedLong ulChannelGroup, xsd__unsignedLong ulBaseSamplingRate, xsd__unsignedLong ulOption1, struct MSXExxxx_Response * Response)`

Parameters

- [in] *ulChannelGroup* : Channel group selection.
- 0 for channels 0 and 1
 - 1 for channels 2 and 3
 - 2 for channels 4 and 5

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

[in] ***ulBaseSamplingRate*** : Sampling rate selection

- 5 for 5Hz
- 10 for 10Hz
- 20 for 20Hz
- 40 for 40Hz
- 80 for 80Hz
- 160 for 160Hz
- 320 for 320Hz
- 640 for 640Hz
- 1000 for 1000Hz
- 2000 for 2000Hz

If only one channel is used then the real sampling rate is $ulBaseSamplingRate / 2$

If all 2 channels are used then the real sampling rate is $ulBaseSamplingRate / 3$

[in] ***ulOption1*** : Reserved. Set to 0

[out] ***Response*** :

iReturnValue :

- 0 : Means the remote function performed OK
- -1 : Means an system error ocured
- -2 : Channel group selection error
- -3 : Sampling rate selection error
- -4 : Acquisition in progress. Can not change the sampling rate
- -100 : IOCTL system call error

syserrno : system-error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

4.1.2.48 `int MSXExxxx__PressureGetConfiguration (xsd__unsignedLong ulChannel,
xsd__unsignedLong ulOption1, struct MSXExxxx__PressureGetConfigurationResponse
* Response)`

Parameters

[in] ***ulChannel*** : Channel selection (0 to 15)

[in] ***ulOption1*** : Reserved. Set to 0

[out] ***Response*** :

sResponse.iReturnValue :

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Channel selection wrong

sResponse.syserrno : system-error code (the value of the libc "errno" code)

dSensorSensibility : Sensor sensibility (mV/V/bar or mV/V/mbar or mV/V/Pa, ...). Refer to sensor documentation

dSensorOffset : Sensor V offset for 0 mV/V/bar, 0 mV/V/mbar, ... Refer to sensor documentation

ulBaseSamplingRate : Sampling rate selection

- 5 for 5Hz
- 10 for 10Hz
- 20 for 20Hz
- 40 for 40Hz
- 80 for 80Hz
- 160 for 160Hz
- 320 for 320Hz
- 640 for 640Hz
- 1000 for 1000Hz
- 2000 for 2000Hz

Returns

- 0: Means the remote function performed OK
- -1: Means an system error occurred
- -2: Channel selection wrong

4.1.2.49 `int MSXExxxx_DigitalIOGetNumberOfChannels (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Returns the number of digital I/O channels.

Parameters

[in] ***ulOption1*** Reserved. Set to 0.

[out] ***Response sResponse.iReturnValue***

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-100**: Internal system error occurred. See value of syserrno

sResponse.syserrno system error code (the value of the libc "errno" code)

ulValue Number of available digital I/O channels

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.50 `int MSXExxxx_DigitalIOGetNumberOfPorts (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Returns the number of digital I/O ports.

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

Parameters

[in] *ulOption1* Reserved. Set to 0

[out] *Response sResponse.iReturnValue*

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-100**: Internal system error occurred. See value of `syserrno`

sResponse.syserrno system error code (the value of the libc "errno" code)

ulValue Number of available digital I/O ports

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.51 `int MSXExxxx_DigitalIOGetPortAvailableDirections (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse * Response)`

Returns the available directions for the selected port (input or output).

Parameters

[in] *ulPort* Selected digital I/O port (0 to `MSXExxxx_DigitalIOGetNumberOfPorts`)

Please read the documentation of the `MSXExxxx_DigitalIOGetNumberOfPorts` for the description of a port.

[in] *ulOption1* Reserved. Set to 0

[out] *Response sResponse.iReturnValue*

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-2**: The *ulPort* parameter is wrong
- **-100**: Internal system error occurred. See value of `syserrno`

sResponse.syserrno system error code (the value of the libc "errno" code)

ulInputs Digital inputs availability. Each bit indicates if the channel can be used as an input.
Example:

- **1**: I/O 0 of the selected port can be used as an input
- **4**: I/O 2 of the selected port can be used as an input
- **3**: I/Os 0 and 1 of the selected port can be used as input

ulOutputs Digital outputs availability. Each bit indicates if the channel can be used as an output.
Example:

- **1**: I/O 0 of the selected port can be used as an output
- **4**: I/O 2 of the selected port can be used as an output

- **3:** I/Os 0 and 1 of the selected port can be used as output

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.52 `int MSXExxxx_DigitalIOSetPortDirections (xsd_unsignedLong ulPort, xsd_unsignedLong ulDirection, xsd_unsignedLong ulOption1, struct MSXExxxx_Response * Response)`

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

Parameters

[in] **ulPort** Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)

Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.

[in] **ulDirection** Digital I/O direction. Each bit indicates if the channel is used as an input or an output. Example:

- **1:** I/O 0 of the selected port is configured as output, all the other I/Os of the selected port are configured as input
- **3:** I/Os 0 and 1 of the selected port are configured as output, all the other I/Os of the selected port are configured as input

[in] **ulOption1** Reserved. Set to 0

[out] **Response iReturnValue**

- **0:** The remote function performed OK
- **-1:** System error occurred
- **-2:** The ulPort parameter is wrong
- **-3:** The ulDirection parameter is wrong
- **-100:** Internal system error occurred. See value of syserrno

syserrno system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.53 `int MSXExxxx_DigitalIOGetPortDirections (xsd_unsignedLong ulPort, struct MSXExxxx_unsignedLongResponse * Response)`

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

Parameters

[in] **ulPort** Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)

Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.

[in] *ulOption1* Reserved. Set to 0

[out] *Response* *sResponse.iReturnValue*

- **0**: The remote function performed OK
- **-1**: System error occurred
- **-2**: The *ulPort* parameter is wrong
- **-100**: Internal system error occurred. See value of *syserrno*

sResponse.syserrno system error code (the value of the libc "errno" code)

ulValue Current digital I/O direction. Each bit indicates if the channel is used as an input or an output. Example:

- **1**: I/O 0 of the selected port is configured as output, all the other I/Os of the selected port are configured as input
- **3**: I/Os 0 and 1 of the selected port are configured as output, all the other I/Os of the selected port are configured as input

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.54 int MSXExxxx_DigitalIOSetInputsFilterTime (xsd__unsignedLong ulFilterTime, xsd__unsignedLong ulOption1, struct MSXExxxx_Response * Response)

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

Parameters

[in] *ulFilterTime* Filter time for the digital inputs in steps of 250 ns (max value: 16777215)

- **0**: Disable the filter
- **1**: Sets the filter time to 250 ns
- **2**: Sets the filter time to 500 ns
- ...
- **16777215**: Sets the filter time to 4 s

[in] *ulOption1* Reserved. Set to 0

[out] *Response* Response of the system

- *iReturnValue*
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The *ulFilterTime* parameter is wrong
 - **-100**: Internal system error occurred. See value of *syserrno*
- *syserrno* system error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.55 `int MSXExxxx__DigitalIOEnableDisableInputsFilter (xsd__unsignedLong ulPort, xsd__unsignedLong ulFilter, xsd__unsignedLong ulOption1, struct MSXExxxx__Response * Response)`

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

Parameters

- [in] **ulPort** Selected digital I/O port (0 to MSXExxxx__DigitalIOGetNumberOfPorts)
Please read the documentation of the MSXExxxx__DigitalIOGetNumberOfPorts for the description of a port.
- [in] **ulFilter** Digital input filter selection. Each bit indicates if the filter is enabled on the input.
Example:
- **1**: Filter only enabled on input 0
 - **3**: Filter enabled on inputs 0 and 1
- [in] **ulOption1** Reserved. Set to 0
- [out] **Response iReturnValue**
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulPort parameter is wrong
 - **-3**: The ulFilter parameter is wrong
 - **-4**: Any selected input is not an input or a bidirectional channel
 - **-100**: Internal system error occurred. See value of syserrno
- syserrno** system error code (the value of the libc "errno" code)

Return values

- 0** SOAP_OK
- Others** See SOAP error

4.1.2.56 `int MSXExxxx__DigitalIOGetInputsFilterConfiguration (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse * Response)`

Reads the digital inputs filter configuration for the selected port.

Parameters

- [in] **ulPort** Selected digital I/O port (0 to MSXExxxx__DigitalIOGetNumberOfPorts)
Please read the documentation of the MSXExxxx__DigitalIOGetNumberOfPorts for the description of a port.
- [in] **ulOption1** Reserved. Set to 0
- [out] **Response sResponse.iReturnValue**
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulPort parameter is wrong
 - **-100**: Internal system error occurred. See value of syserrno

sResponse.syserrno system error code (the value of the libc "errno" code)
ulFilterTime Filter time value (from 1 to 16777215) 1 corresponds to 250 ns, 2 corresponds to 500 ns, ...
ulFilter Digital inputs filter selection. Each bit indicate the filter state for one digital input channel.

Return values

0 SOAP_OK
Others See SOAP error

4.1.2.57 `int MSXExxxx__DigitalIOTestOutputsShortCircuit (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Get the short-circuit status of the outputs of the selected port.

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

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

Parameters

- [in] *ulPort* Selected digital I/O port (0 to MSXExxxx__DigitalIOGetNumberOfPorts)
 Please read the documentation of the MSXExxxx__DigitalIOGetNumberOfPorts for the description of a port.
- [in] *ulOption1* Reserved. Set to 0
- [out] *Response sResponse.iReturnValue* :
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulPort parameter is wrong
 - **-100**: Internal system error occurred. See value of syserrno
- sResponse.syserrno* : system error code (the value of the libc "errno" code)
ulValue : Digital outputs short circuit state. Each bit represents the short-circuit state of one digital output channel.
- B0 : 0: Digital I/O 0/32 no short circuit. 1: Digital I/O 0/32 short circuit
 - ...
 - D31 : 0: Digital I/O 31/63 no short circuit. 1: Digital I/O 31/63 short circuit

Return values

0 SOAP_OK
Others See SOAP error

4.1.2.58 `int MSXExxxx__DigitalIORearmOutputsShortCircuit (xsd__unsignedLong ulOption1, struct MSXExxxx__Response * Response)`

Rearm the digital outputs short circuit.

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

Parameters

- [in] *ulOption1* Reserved. Set to 0
- [out] *Response iReturnValue*
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-100**: Internal system error occurred. See value of *syserrno*
- syserrno* system error code (the value of the libc "errno" code)

Returns

- 0: SOAP_OK
- <> 0: See SOAP error

4.1.2.59 `int MSXExxxx_DigitalIOTestOutputsPowerSupply (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Reads the current power supply status of the selected port.

The digital outputs need an external power supply. This function checks the state of the power supply.

Parameters

- [in] *ulPort* Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)
- Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.
- [in] *ulOption1* Reserved. Set to 0
- [out] *Response sResponse.iReturnValue*
- **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The ulPort parameter is wrong
 - **-100**: Internal system error occurred. See value of *syserrno*
- sResponse.syserrno* system error code (the value of the libc "errno" code)
- ulValue* Current digital I/O power supply state. Each bit indicates the power supply state of the output. Example:
- **1**: No external supply voltage for the output 0
 - **3**: No external supply voltage for the outputs 0 and 1

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.60 `int MSXExxxx_DigitalIOReadChannel (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Read the selected digital I/O channel.

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

Parameters

- [in] **ulChannel** Selected digital I/O channel (0 to MSXExxxx_DigitalIOGetNumberOfChannels)
- [in] **ulOption1** Reserved. Set to 0
- [out] **Response sResponse.iReturnValue** :
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulChannel parameter is wrong
 - **-100**: Internal system error occurred. See value of syserrno
- sResponse.syserrno** system error code (the value of the libc "errno" code)
- ulValue** : Digital I/O channel state
 - **0**: Digital I/O channel is low
 - **1**: Digital I/O channel is high

Return values

- 0** SOAP_OK
- Others** See SOAP error

4.1.2.61 `int MSXExxxx_DigitalIOReadPort (xsd__unsignedLong ulPort, xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Read the selected digital I/O port.

Parameters

- [in] **ulPort** Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)
Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.
- [in] **ulOption1** Reserved. Set to 0
- [out] **Response sResponse.iReturnValue**
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The ulPort parameter is wrong
 - **-100**: Internal system error occurred. See value of syserrno
- sResponse.syserrno** system error code (the value of the libc "errno" code)
- ulValue** Digital I/O state. Each bit indicates the state of one digital I/O channel.
 - **D0** : 0: Digital I/O 0/32 is low. 1: Digital I/O 0/32 is high
 - ...
 - **D31** : 0: Digital I/O 31/63 is low. 1: Digital I/O 31/63 is high

Return values

- 0** SOAP_OK
- Others** See SOAP error

4.1.2.62 `int MSXExxxx_DigitalIOWriteChannel (xsd__unsignedLong ulChannel, xsd__unsignedLong ulState, xsd__unsignedLong ulOption1, struct MSXExxxx_Response * Response)`

Set the selected digital output channel to on or off.

Parameters

- [in] ***ulChannel*** Selected digital I/O channel (0 to MSXExxxx_DigitalIOGetNumberOfChannels)
 - [in] ***ulState*** Digital I/O channel state
 - 0: Set the digital I/O output channel to low
 - 1: Set the digital I/O output channel to high
 - [in] ***ulOption1*** Reserved. Set to 0
 - [out] ***Response iReturnValue***
 - 0: The remote function performed OK
 - -1: System error occurred
 - -2: The ulChannel parameter is wrong
 - -3: The ulState parameter is wrong
 - -4: The selected digital I/O is not an output
 - -100: Internal system error occurred. See value of syserrno
- syserrno*** system error code (the value of the libc "errno" code)

Return values

- 0 SOAP_OK
- Others*** See SOAP error

4.1.2.63 `int MSXExxxx_DigitalIOWritePort (xsd__unsignedLong ulPort, xsd__unsignedLong ulState, xsd__unsignedLong ulOption1, struct MSXExxxx_Response * Response)`

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

Parameters

- [in] ***ulPort*** Selected digital I/O port (0 to MSXExxxx_DigitalIOGetNumberOfPorts)
Please read the documentation of the MSXExxxx_DigitalIOGetNumberOfPorts for the description of a port.
 - [in] ***ulState*** Digital I/O state. Each bit set the state for one digital I/O channel (0: off, 1: on).
 - [in] ***ulOption1*** Reserved. Set to 0
 - [out] ***Response iReturnValue***
 - 0: The remote function performed OK
 - -1: System error occurred
 - -2: The ulPort parameter is wrong
 - -3: The ulState parameter is wrong
 - -4: Any digital I/O set to 1 is not an output channel
 - -100: Internal system error occurred. See value of syserrno
- syserrno*** system error code (the value of the libc "errno" code)

Return values

- 0 SOAP_OK
- Others*** See SOAP error

4.1.2.64 `int MSXExxxx__SinCosGetNumberOfChannels (xsd__unsignedLong ulOption1, struct MSXExxxx__unsignedLongResponse * Response)`

Return the number of Sine/Cosine channels.

Parameters

[in] *ulOption1* Reserved. Set to 0

[out] *Response* Response of the system

- *sResponse*
 - *iReturnValue* Function return code
 - * **0**: The remote function performed OK
 - * **-1**: System error occurred
 - * **-100**: Internal system error occurred. See value of syserrno
 - *syserrno* System error code (the value of the libc "errno" code)
- *ulValue* Number of available Sine / Cosine channels

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.65 `int MSXExxxx__SinCosInit (xsd__unsignedLong ulChannel, xsd__double dSignalPeriod, xsd__unsignedLong ulResolution, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXExxxx__SinCosInitResponse * Response)`

Initialise the selected Sine/Cosine channel.

Parameters

[in] *ulChannel* Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)

[in] *dSignalPeriod* Signal period (Unit/period)

[in] *ulResolution* Resolution to use for the measure (steps/period)

Maximum input frequency corresponding to the parameter ulResolution:

+-----+-----+		
Resolution	Max. Freq. Hz.	
+-----+-----+		
16	250000	
+-----+-----+		
32	162500	
+-----+-----+		
40	16300	
+-----+-----+		
64	81300	
+-----+-----+		
80	16300	
+-----+-----+		
100	26000	
+-----+-----+		
128	40600	
+-----+-----+		
160	16300	
+-----+-----+		
200	26000	

256	20300	
320	16300	
400	13000	
500	10400	
512	10200	
800	6500	
1000	5200	
1024	5100	
1600	3300	
2000	2600	
2048	2540	
4096	1270	
8192	635	

*

[in] ***ulOption01*** Reserved. Set it to 0

[in] ***ulOption02*** Reserved. Set it to 0

[in] ***ulOption03*** Reserved. Set it to 0

[in] ***ulOption04*** Reserved. Set it to 0

[out] ***Response*** Response of the system

- ***sResponse***

- ***iReturnValue*** Function return code

- * **0**: The remote function performed OK

- * **-1**: System error occurred

- * **-2**: The PLD is not working

- * **-3**: The ulChannel parameter is wrong

- * **-4**: The current status is not correct. The selected Sine/Cosine channel has already been initialised. Please call the MSXExxxx__SinCosRelease and retry

- * **-5**: The dSignalPeriod parameter is wrong

- * **-6**: The ulResolution parameter is wrong

- * **-7**: Gain calibration error occurred. Please check the connection with the sensor

- * **-21**: Timeout while initialising the Sine/Cosine sensor. Please check the connection with the sensor

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

- ***syserrno*** System error code (the value of the libc "errno" code)

- ***ulMaxInputFrequency*** Return the maximal input frequency that can be used (in Hz).

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.66 `int MSXExxxx__SinCosRelease (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

Release the selected Sine / Cosine channel.

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
 - **iReturnValue** Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The ulChannel parameter is wrong
 - **-4**: The current status is not correct. The selected Sine/Cosine channel is not initialised
 - **-100**: Internal system error occurred. See value of syserrno
 - **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.67 `int MSXExxxx__SinCosGetConfiguration (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption02, struct MSXExxxx__SinCosGetConfigurationResponse * Response)`

Get the current configuration of the selected Sine/Cosine channel.

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
 - **sResponse**
 - **iReturnValue** Function return code
 - * **0**: The remote function performed OK
 - * **-1**: System error occurred
 - * **-2**: The PLD is not working
 - * **-3**: The ulChannel parameter is wrong
 - * **-100**: Internal system error occurred. See value of syserrno
 - **syserrno** System error code (the value of the libc "errno" code)
 - **ulInitialisationState** Initialisation state of the Sine/Cosine channel (0: uninitialised, 1: initialised)

- *dSignalPeriod* Signal period (Unit/period)
- *ulResolution* Resolution to use for the measure (steps/period)
- *ulIndexState* State of the index logic (0: uninitialised, 1: initialised)
- *ulIndexEdge* Edge selected for the index logic
- *ulIndexAction* Action to do when the index signal occurs
- *ulCompareLogicState* State of the compare logic (0: uninitialised, 1: initialised)
- *dCompareLogicValue* Compare value of the compare logic
- *ulCompareLogicValueFormat* Format of the compare value of the compare logic (0: Raw, 1: Standardised)
- *ulCompareLogicMode* Compare mode of the compare logic (0: Simple, 1: Modulo)
- *ulCompareLogicSynchro* Synchro trigger generation of the compare logic (0: No, 1: Yes)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.68 `int MSXExxxx__SinCosRead (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXExxxx__SinCosReadResponse * Response)`

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

Parameters

- [in] *ulChannel* Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] *ulOption01* Reserved. Set it to 0
- [in] *ulOption02* Reserved. Set it to 0
- [in] *ulOption03* Reserved. Set it to 0
- [in] *ulOption04* Reserved. Set it to 0
- [out] *Response* Response of the system
 - *sResponse*
 - *iReturnValue* Function return code
 - * **0**: The remote function performed OK
 - * **-1**: System error occurred
 - * **-2**: The PLD is not working
 - * **-3**: The ulChannel parameter is wrong
 - * **-4**: The current status is not correct. The selected Sine/Cosine channel is not initialised
 - * **-100**: Internal system error occurred. See value of syserrno
 - *syserrno* System error code (the value of the libc "errno" code)
 - *dValue* Measured value (Unit)
 - *ulValue* Measured value (Raw value)
 - *ulMeasureError* Status of the Sine/Cosine measurement. If the value is 1, then an error occurred. When an error occurred, the reference point is corrupted. To correct it, you must first reset the error flag using the function MSXExxxx__SinCosClearError, and then reset your reference point.
 - *ulInfo01* Reserved

- *ulInfo02* Reserved

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.69 `int MSXExxxx__SinCosClear (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

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

Parameters

- [in] *ulChannel* Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] *ulOption01* Reserved. Set it to 0
- [in] *ulOption02* Reserved. Set it to 0
- [out] *Response* Response of the system
 - *iReturnValue* Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The *ulChannel* parameter is wrong
 - **-4**: The current status is not correct. The selected Sine/Cosine channel is not initialised
 - **-100**: Internal system error occurred. See value of *syserrno*
 - *syserrno* System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.70 `int MSXExxxx__SinCosClearError (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

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

Parameters

- [in] *ulChannel* Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] *ulOption01* Reserved. Set it to 0
- [in] *ulOption02* Reserved. Set it to 0
- [out] *Response* Response of the system
 - *iReturnValue* Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred

- **-2:** The PLD is not working
- **-3:** The ulChannel parameter is wrong
- **-4:** The current status is not correct. The selected Sine/Cosine channel is not initialised
- **-100:** Internal system error occurred. See value of syserrno
- **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.71 `int MSXExxxx__SinCosInitIndex (xsd__unsignedLong ulChannel, xsd__unsignedLong ulEdge, xsd__unsignedLong ulAction, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

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

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] **ulEdge** Edge selection
 - **0b01 = 1:** Rising edge
 - **0b10 = 2:** Falling edge
 - **0b11 = 3:** Both edges
- [in] **ulAction** Action to do when the index trigger occurs
 - **0b00 = 0:** Do nothing (but index trigger can also be used to trigger the acquisition)
 - **0b01 = 1:** Clear the value of the sensor
 - **0b10 = 2:** Generate a synchro trigger
 - **0b11 = 3:** Clear the value of the sensor AND Generate a synchro trigger
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
 - **iReturnValue** Function return code
 - **0:** The remote function performed OK
 - **-1:** System error occurred
 - **-2:** The PLD is not working
 - **-3:** The ulChannel parameter is wrong
 - **-4:** The ulEdge parameter is wrong
 - **-5:** The ulAction parameter is wrong
 - **-6:** The current status is not correct. The selected Sine/Cosine channel is not initialised
 - **-7:** The current status is not correct. The selected Sine/Cosine channel Index Logic has already been initialised. Please call the function MSXExxxx__SinCosReleaseIndex
 - **-100:** Internal system error occurred. See value of syserrno
 - **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.72 `int MSXExxxx__SinCosReleaseIndex (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

Release the Index configuration.

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
 - **iReturnValue** Function return code
 - **0**: The remote function performed OK
 - **-1**: System error occurred
 - **-2**: The PLD is not working
 - **-3**: The ulChannel parameter is wrong
 - **-4**: The current status is not correct. The selected Sine/Cosine channel Index Logic is not initialised
 - **-100**: Internal system error occurred. See value of syserrno
 - **syserrno** System error code (the value of the libc "errno" code)

Return values

- 0** SOAP_OK
- Others** See SOAP error

4.1.2.73 `int MSXExxxx__SinCosInitAndEnableCompareLogic (xsd__unsignedLong ulChannel, xsd__double dValue, xsd__unsignedLong ulValueFormat, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

Initialise and enable a Sine/Cosine Compare Logic.

Parameters

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

- **1:** Modulo mode. When the counter value corresponds to the compare value (dValue) or a multiple of it, a trigger or synchro trigger is released. In that mode, the value dValue cannot be 0.
- [in] **ulSynchroTrigger** Generate synchro trigger, which can be used to trigger another MSX-E system, when the compare trigger is released
- **0:** No
 - **1:** Yes
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
- **iReturnValue** Function return code
 - **0:** The remote function performed OK
 - **-1:** System error occurred
 - **-2:** The PLD is not working
 - **-3:** The ulChannel parameter is wrong
 - **-4:** The ulValueFormat parameter is wrong
 - **-5:** The ulMode parameter is wrong
 - **-6:** The ulSynchroTrigger parameter is wrong
 - **-7:** In Modulo mode, the parameter dValue cannot be 0
 - **-8:** The dValue parameter is wrong
 - **-9:** The current status is not correct. The selected Sine/Cosine channel is not initialised
 - **-10:** The current status is not correct. The selected Sine/Cosine channel Compare Logic has already been initialised. Please call the function MSXExxxx__SinCosDisableAndReleaseCompareLogic
 - **-100:** Internal system error occurred. See value of syserrno
 - **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

4.1.2.74 `int MSXExxxx__SinCosDisableAndReleaseCompareLogic (xsd__unsignedLong ulChannel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXExxxx__Response * Response)`

Disable and Release the Sine/Cosine Compare Logic

Parameters

- [in] **ulChannel** Channel selection (0 to MSXExxxx__SinCosGetNumberOfChannels - 1)
- [in] **ulOption01** Reserved. Set it to 0
- [in] **ulOption02** Reserved. Set it to 0
- [out] **Response** Response of the system
- **iReturnValue** Function return code
 - **0:** The remote function performed OK
 - **-1:** System error occurred

- **-2:** The PLD is not working
- **-3:** The ulChannel parameter is wrong
- **-4:** The current status is not correct. The selected Sine/Cosine channel Compare Logic is not initialised
- **-100:** Internal system error occurred. See value of syserrno
- **syserrno** System error code (the value of the libc "errno" code)

Return values

0 SOAP_OK

Others See SOAP error

Index

- `__offset`
 - ByteArray, 67
 - UnsignedLongArray, 85
 - UnsignedShortArray, 86
 - `__ptr`
 - ByteArray, 67
 - UnsignedLongArray, 85
 - UnsignedShortArray, 86
 - `xsd__base64Binary`, 86
 - `__size`
 - ByteArray, 67
 - UnsignedLongArray, 85
 - UnsignedShortArray, 86
 - `xsd__base64Binary`, 86
- `bArray`
 - `MXCommon__-`
 - `GetAutoConfigurationFileResponse`, 79
- `bCryptedValueArray`
 - `MXCommon__TestCustomerIDResponse`, 84
- `bValueArray`
 - `MXCommon__TestCustomerIDResponse`, 84
- `ByteArray`, 67
 - `__offset`, 67
 - `__ptr`, 67
 - `__size`, 67
- Common functions, 3
- Common general functions, 4
- Common hardware trigger functions, 11
- Common I/O auto configuration functions, 18
- Common security functions, 13
- Common synchronisation timer functions, 20
- Common temperature functions, 10
- Common time functions, 15
- `Common_autoconf`
 - `MXCommon__GetAutoConfigurationFile`, 19
 - `MXCommon__SetAutoConfigurationFile`, 19
 - `MXCommon__StartAutoConfiguration`, 20
- `Common_configuration`
 - `MXCommon__-`
 - `ApplyConfigurationBackupFile`, 23
 - `MXCommon__ChangePassword`, 24
 - `MXCommon__GetConfigurationBackupFile`, 23
- `Common_general`
 - `MXCommon__DataseverRestart`, 8
 - `MXCommon__GetClientConnections`, 6
 - `MXCommon__GetEthernetLinksStates`, 9
 - `MXCommon__GetHostname`, 5
 - `MXCommon__GetModuleType`, 5
 - `MXCommon__Reboot`, 7
 - `MXCommon__ResetAllIOFunctionalities`, 8
 - `MXCommon__SetHostname`, 6
 - `MXCommon__Strerror`, 6
- `Common_hardware_trigger`
 - `MXCommon__-`
 - `GetHardwareTriggerFilterTime`, 12
 - `MXCommon__GetHardwareTriggerState`, 13
 - `MXCommon__-`
 - `SetHardwareTriggerFilterTime`, 12
- `Common_security`
 - `MXCommon__SetCustomerKey`, 15
 - `MXCommon__TestCustomerID`, 15
- `Common_synchrotimer`
 - `MXCommon__InitAndStartSynchroTimer`, 21
 - `MXCommon__-`
 - `StopAndReleaseSynchroTimer`, 21
- `Common_temperature`
 - `MXCommon__-`
 - `GetModuleTemperatureValueAndStatus`, 10
 - `MXCommon__-`
 - `SetModuleTemperatureWarningLevels`, 11
- `Common_time`
 - `MXCommon__GetTime`, 17
 - `MXCommon__GetUpTime`, 18
 - `MXCommon__HardwareClockToSys`, 17
 - `MXCommon__SetTime`, 16
 - `MXCommon__SysToHardwareClock`, 16
- Customer option management, 27
- `CustomerOption`
 - `MXCommon__GetOptionInformation`, 28
- `dCompareLogicValue`
 - `MSXExxxx__-`
 - `SinCosGetConfigurationResponse`,

- 75
- DefaultResponse, 67
- iReturnValue, 68
- syserrno, 68
- dSensorOffset
 - MSXExxxx_-
 - PressureGetConfigurationResponse, 73
- dSensorSensibility
 - MSXExxxx_-
 - PressureGetConfigurationResponse, 73
- dSignalPeriod
 - MSXExxxx_-
 - SinCosGetConfigurationResponse, 75
- dTemperatureValue
 - MXCommon_-
 - GetModuleTemperatureValueAndStatusResponse, 82
- dValue
 - MSXExxxx_-
 - AcquisitionAutoRefreshGetValuesResponse, 69
 - MSXExxxx__SinCosReadResponse, 76
- iReturnValue
 - DefaultResponse, 68
 - MSXExxxx__Response, 73
 - MXCommon__Response, 83
- MSX-E3317 Acquisition functions, 29
- MSX-E3317 Acquisition information functions, 30
- MSX-E3317 Autorefresh functions, 31
- MSX-E3317 Digital I/O diagnostic functions, 49
- MSX-E3317 Digital I/O filter functions, 47
- MSX-E3317 Digital I/O functions, 43
- MSX-E3317 Digital I/O information, configuration functions, 44
- MSX-E3317 Digital I/O read/write functions, 52
- MSX-E3317 functions, 3
- MSX-E3317 Pressure functions, 40
- MSX-E3317 Pressure initialisation/information functions, 40
- MSX-E3317 Sequence functions, 36
- MSX-E3317 Sine/Cosine Compare functions, 63
- MSX-E3317 Sine/Cosine functions, 54
- MSX-E3317 Sine/Cosine Index functions, 61
- MSX-E3317 Sine/Cosine initialisation/information functions, 55
- MSX-E3317 Sine/Cosine read and clear functions, 59
- MSXE3317_Acquisition_Info
 - MSXExxxx__AcquisitionGetChannelInfo, 30
- MSXExxxx_-
 - AcquisitionGetNumberOfChannels, 30
- MSXE3317_Autorefresh
 - MSXExxxx_-
 - AcquisitionAutoRefreshGetValues, 35
 - MSXExxxx_-
 - AcquisitionAutoRefreshInitAndStart, 33
 - MSXExxxx_-
 - AcquisitionAutoRefreshStopAndRelease, 35
- MSXE3317_DigIO_Access
 - MSXExxxx__DigitalIOReadChannel, 52
 - MSXExxxx__DigitalIOReadPort, 52
 - MSXExxxx__DigitalIOWriteChannel, 53
 - MSXExxxx__DigitalIOWritePort, 54
- MSXE3317_DigIO_Diagnostic
 - MSXExxxx_-
 - DigitalIORearmOutputsShortCircuit, 50
 - MSXExxxx_-
 - DigitalIOTestOutputsPowerSupply, 51
 - MSXExxxx_-
 - DigitalIOTestOutputsShortCircuit, 50
- MSXE3317_DigIO_Filter
 - MSXExxxx_-
 - DigitalIOEnableDisableInputsFilter, 48
 - MSXExxxx_-
 - DigitalIOGetInputsFilterConfiguration, 49
 - MSXExxxx__DigitalIOSetInputsFilterTime, 48
- MSXE3317_DigIO_Info
 - MSXExxxx_-
 - DigitalIOGetNumberOfChannels, 44
 - MSXExxxx__DigitalIOGetNumberOfPorts, 45
 - MSXExxxx_-
 - DigitalIOGetPortAvailableDirections, 45
 - MSXExxxx__DigitalIOGetPortDirections, 46
 - MSXExxxx__DigitalIOSetPortDirections, 46
- MSXE3317_Pressure_Init
 - MSXExxxx__PressureGetConfiguration, 42
 - MSXExxxx_-
 - PressureGetNumberOfChannels, 41
 - MSXExxxx_-
 - PressureSetChannelConfiguration, 41

- MSXExxxx__PressureSetSamplingRate, [41](#)
- MSXE3317_public_doc.h, [87](#)
- MSXExxxx__-
 - AcquisitionAutoRefreshGetValues, [116](#)
- MSXExxxx__-
 - AcquisitionAutoRefreshInitAndStart, [114](#)
- MSXExxxx__-
 - AcquisitionAutoRefreshStopAndRelease, [117](#)
- MSXExxxx__AcquisitionGetChannelInfo, [114](#)
- MSXExxxx__-
 - AcquisitionGetNumberOfChannels, [113](#)
- MSXExxxx__-
 - AcquisitionSequenceInitAndStart, [117](#)
- MSXExxxx__-
 - AcquisitionSequenceStopAndRelease, [120](#)
- MSXExxxx__-
 - DigitalIOEnableDisableInputsFilter, [126](#)
- MSXExxxx__-
 - DigitalIOGetInputsFilterConfiguration, [127](#)
- MSXExxxx__-
 - DigitalIOGetNumberOfChannels, [123](#)
- MSXExxxx__DigitalIOGetNumberOfPorts, [123](#)
- MSXExxxx__-
 - DigitalIOGetPortAvailableDirections, [124](#)
- MSXExxxx__DigitalIOGetPortDirections, [125](#)
- MSXExxxx__DigitalIOReadChannel, [129](#)
- MSXExxxx__DigitalIOReadPort, [130](#)
- MSXExxxx__-
 - DigitalIORearmOutputsShortCircuit, [128](#)
- MSXExxxx__DigitalIOSetInputsFilterTime, [126](#)
- MSXExxxx__DigitalIOSetPortDirections, [125](#)
- MSXExxxx__-
 - DigitalIOTestOutputsPowerSupply, [129](#)
- MSXExxxx__-
 - DigitalIOTestOutputsShortCircuit, [128](#)
- MSXExxxx__DigitalIOWriteChannel, [130](#)
- MSXExxxx__DigitalIOWritePort, [131](#)
- MSXExxxx__PressureGetConfiguration, [122](#)
- MSXExxxx__-
 - PressureGetNumberOfChannels, [120](#)
- MSXExxxx__-
 - PressureSetChannelConfiguration, [121](#)
- MSXExxxx__PressureSetSamplingRate, [121](#)
- MSXExxxx__SinCosClear, [136](#)
- MSXExxxx__SinCosClearError, [136](#)
- MSXExxxx__-
 - SinCosDisableAndReleaseCompareLogic, [139](#)
- MSXExxxx__SinCosGetConfiguration, [134](#)
- MSXExxxx__SinCosGetNumberOfChannels, [131](#)
- MSXExxxx__SinCosInit, [132](#)
- MSXExxxx__-
 - SinCosInitAndEnableCompareLogic, [138](#)
- MSXExxxx__SinCosInitIndex, [137](#)
- MSXExxxx__SinCosRead, [135](#)
- MSXExxxx__SinCosRelease, [133](#)
- MSXExxxx__SinCosReleaseIndex, [137](#)
- MXCommon__-
 - ApplyConfigurationBackupFile, [109](#)
- MXCommon__ChangePassword, [109](#)
- MXCommon__DataseverRestart, [99](#)
- MXCommon__GetAutoConfigurationFile, [105](#)
- MXCommon__GetClientConnections, [96](#)
- MXCommon__GetConfigurationBackupFile, [108](#)
- MXCommon__GetEthernetLinksStates, [99](#)
- MXCommon__-
 - GetHardwareTriggerFilterTime, [102](#)
- MXCommon__GetHardwareTriggerState, [102](#)
- MXCommon__GetHostname, [96](#)
- MXCommon__-
 - GetModuleTemperatureValueAndStatus, [100](#)
- MXCommon__GetModuleType, [95](#)
- MXCommon__GetOptionInformation, [112](#)
- MXCommon__GetStateIDFromName, [111](#)
- MXCommon__GetStateNameFromID, [111](#)
- MXCommon__GetSubsystemIDFromName, [110](#)
- MXCommon__GetSubsystemNameFromID, [111](#)
- MXCommon__GetSubSystemState, [110](#)
- MXCommon__GetSynchronizationStatus, [113](#)
- MXCommon__GetTime, [105](#)
- MXCommon__GetUpTime, [105](#)
- MXCommon__HardwareClockToSys, [104](#)

- MXCommon__InitAndStartSynchroTimer, 107
- MXCommon__Reboot, 98
- MXCommon__ResetAllIOFunctionalities, 98
- MXCommon__SetAutoConfigurationFile, 106
- MXCommon__SetCustomerKey, 103
- MXCommon__-
 - SetHardwareTriggerFilterTime, 101
- MXCommon__SetHostname, 96
- MXCommon__-
 - SetModuleTemperatureWarningLevels, 101
- MXCommon__SetTime, 103
- MXCommon__SetToMaster, 112
- MXCommon__StartAutoConfiguration, 106
- MXCommon__-
 - StopAndReleaseSynchroTimer, 108
- MXCommon__Sterror, 97
- MXCommon__SysToHardwareClock, 104
- MXCommon__TestCustomerID, 103
- xsd__char, 95
- xsd__double, 95
- xsd__float, 95
- xsd__int, 95
- xsd__long, 95
- xsd__string, 95
- xsd__unsignedByte, 95
- xsd__unsignedInt, 95
- xsd__unsignedLong, 95
- xsd__unsignedShort, 95
- MSXE3317_Sequence
 - MSXExxxx__-
 - AcquisitionSequenceInitAndStart, 37
 - MSXExxxx__-
 - AcquisitionSequenceStopAndRelease, 39
- MSXE3317_SinCos_Compare
 - MSXExxxx__-
 - SinCosDisableAndReleaseCompareLogic, 64
 - MSXExxxx__-
 - SinCosInitAndEnableCompareLogic, 63
- MSXE3317_SinCos_Index
 - MSXExxxx__SinCosInitIndex, 61
 - MSXExxxx__SinCosReleaseIndex, 62
- MSXE3317_SinCos_Init
 - MSXExxxx__SinCosGetConfiguration, 58
 - MSXExxxx__SinCosGetNumberOfChannels, 55
 - MSXExxxx__SinCosInit, 56
 - MSXExxxx__SinCosRelease, 57
- MSXE3317_SinCos_Read
 - MSXExxxx__SinCosClear, 60
 - MSXExxxx__SinCosClearError, 60
 - MSXExxxx__SinCosRead, 59
- MSXExxxx__AcquisitionAutoRefreshGetValues
 - MSXE3317_Autorefresh, 35
 - MSXE3317_public_doc.h, 116
- MSXExxxx__AcquisitionAutoRefreshGetValuesResponse, 68
 - dValue, 69
 - sResponse, 69
 - ulCounterValue, 69
 - ulTimeStampHigh, 69
 - ulTimeStampLow, 69
- MSXExxxx__AcquisitionAutoRefreshInitAndStart
 - MSXE3317_Autorefresh, 33
 - MSXE3317_public_doc.h, 114
- MSXExxxx__AcquisitionAutoRefreshStopAndRelease
 - MSXE3317_Autorefresh, 35
 - MSXE3317_public_doc.h, 117
- MSXExxxx__AcquisitionGetChannelInfo
 - MSXE3317_Acquisition_Info, 30
 - MSXE3317_public_doc.h, 114
- MSXExxxx__AcquisitionGetChannelInfoResponse, 69
 - sResponse, 69
 - ulChannelIndex, 70
 - ulHwPosition, 70
 - ulType, 69
- MSXExxxx__AcquisitionGetNumberOfChannels
 - MSXE3317_Acquisition_Info, 30
 - MSXE3317_public_doc.h, 113
- MSXExxxx__AcquisitionSequenceInitAndStart
 - MSXE3317_public_doc.h, 117
 - MSXE3317_Sequence, 37
- MSXExxxx__AcquisitionSequenceInitAndStartChannelListParam, 70
 - ulChannelList, 70
- MSXExxxx__AcquisitionSequenceStopAndRelease
 - MSXE3317_public_doc.h, 120
 - MSXE3317_Sequence, 39
- MSXExxxx__DigitalIOEnableDisableInputsFilter
 - MSXE3317_DigIO_Filter, 48
 - MSXE3317_public_doc.h, 126
- MSXExxxx__DigitalIOGetInputsFilterConfiguration
 - MSXE3317_DigIO_Filter, 49
 - MSXE3317_public_doc.h, 127
- MSXExxxx__DigitalIOGetInputsFilterConfigurationResponse, 70
 - sResponse, 71
 - ulFilter, 71
 - ulFilterTime, 71
- MSXExxxx__DigitalIOGetNumberOfChannels
 - MSXE3317_DigIO_Info, 44
 - MSXE3317_public_doc.h, 123

- MSXExxxx__DigitalIOGetNumberOfPorts
 - MSXE3317_DigIO_Info, [45](#)
 - MSXE3317_public_doc.h, [123](#)
- MSXExxxx__DigitalIOGetPortAvailableDirections
 - MSXE3317_DigIO_Info, [45](#)
 - MSXE3317_public_doc.h, [124](#)
- MSXExxxx__DigitalIOGetPortAvailableDirectionsResponse
 - [71](#)
 - sResponse, [71](#)
 - ulInputs, [71](#)
 - ulOutputs, [71](#)
- MSXExxxx__DigitalIOGetPortDirections
 - MSXE3317_DigIO_Info, [46](#)
 - MSXE3317_public_doc.h, [125](#)
- MSXExxxx__DigitalIOReadChannel
 - MSXE3317_DigIO_Access, [52](#)
 - MSXE3317_public_doc.h, [129](#)
- MSXExxxx__DigitalIOReadPort
 - MSXE3317_DigIO_Access, [52](#)
 - MSXE3317_public_doc.h, [130](#)
- MSXExxxx__DigitalIORearmOutputsShortCircuit
 - MSXE3317_DigIO_Diagnostic, [50](#)
 - MSXE3317_public_doc.h, [128](#)
- MSXExxxx__DigitalIOSetInputsFilterTime
 - MSXE3317_DigIO_Filter, [48](#)
 - MSXE3317_public_doc.h, [126](#)
- MSXExxxx__DigitalIOSetPortDirections
 - MSXE3317_DigIO_Info, [46](#)
 - MSXE3317_public_doc.h, [125](#)
- MSXExxxx__DigitalIOTestOutputsPowerSupply
 - MSXE3317_DigIO_Diagnostic, [51](#)
 - MSXE3317_public_doc.h, [129](#)
- MSXExxxx__DigitalIOTestOutputsShortCircuit
 - MSXE3317_DigIO_Diagnostic, [50](#)
 - MSXE3317_public_doc.h, [128](#)
- MSXExxxx__DigitalIOWriteChannel
 - MSXE3317_DigIO_Access, [53](#)
 - MSXE3317_public_doc.h, [130](#)
- MSXExxxx__DigitalIOWritePort
 - MSXE3317_DigIO_Access, [54](#)
 - MSXE3317_public_doc.h, [131](#)
- MSXExxxx__FileResponse, [72](#)
 - sArray, [72](#)
 - sResponse, [72](#)
 - ulEOF, [72](#)
- MSXExxxx__PressureGetConfiguration
 - MSXE3317_Pressure_Init, [42](#)
 - MSXE3317_public_doc.h, [122](#)
- MSXExxxx__PressureGetConfigurationResponse, [72](#)
 - dSensorOffset, [73](#)
 - dSensorSensibility, [73](#)
 - sResponse, [73](#)
 - ulBaseSamplingRate, [73](#)
- MSXExxxx__PressureGetNumberOfChannels
 - MSXE3317_Pressure_Init, [41](#)
 - MSXE3317_public_doc.h, [120](#)
- MSXExxxx__PressureSetChannelConfiguration
 - MSXE3317_Pressure_Init, [41](#)
 - MSXE3317_public_doc.h, [121](#)
- MSXExxxx__PressureSetSamplingRate
 - MSXE3317_Pressure_Init, [41](#)
 - MSXE3317_public_doc.h, [121](#)
- MSXExxxx__Response, [73](#)
 - iReturnValue, [73](#)
 - syserrno, [73](#)
- MSXExxxx__SinCosClear
 - MSXE3317_public_doc.h, [136](#)
 - MSXE3317_SinCos_Read, [60](#)
- MSXExxxx__SinCosClearError
 - MSXE3317_public_doc.h, [136](#)
 - MSXE3317_SinCos_Read, [60](#)
- MSXExxxx__SinCosDisableAndReleaseCompareLogic
 - MSXE3317_public_doc.h, [139](#)
 - MSXE3317_SinCos_Compare, [64](#)
- MSXExxxx__SinCosGetConfiguration
 - MSXE3317_public_doc.h, [134](#)
 - MSXE3317_SinCos_Init, [58](#)
- MSXExxxx__SinCosGetConfigurationResponse, [73](#)
 - dCompareLogicValue, [75](#)
 - dSignalPeriod, [75](#)
 - sResponse, [75](#)
 - ulCompareLogicMode, [75](#)
 - ulCompareLogicState, [75](#)
 - ulCompareLogicSynchro, [75](#)
 - ulCompareLogicValueFormat, [75](#)
 - ulIndexAction, [75](#)
 - ulIndexEdge, [75](#)
 - ulIndexState, [75](#)
 - ulInfo01, [75](#)
 - ulInfo02, [75](#)
 - ulInitialisationState, [75](#)
 - ulResolution, [75](#)
- MSXExxxx__SinCosGetNumberOfChannels
 - MSXE3317_public_doc.h, [131](#)
 - MSXE3317_SinCos_Init, [55](#)
- MSXExxxx__SinCosInit
 - MSXE3317_public_doc.h, [132](#)
 - MSXE3317_SinCos_Init, [56](#)
- MSXExxxx__SinCosInitAndEnableCompareLogic
 - MSXE3317_public_doc.h, [138](#)
 - MSXE3317_SinCos_Compare, [63](#)
- MSXExxxx__SinCosInitIndex
 - MSXE3317_public_doc.h, [137](#)
 - MSXE3317_SinCos_Index, [61](#)
- MSXExxxx__SinCosInitResponse, [75](#)
 - sResponse, [76](#)

- ulInfo01, 76
- ulInfo02, 76
- ulMaxInputFrequency, 76
- MSXExxxx__SinCosRead
 - MSXE3317_public_doc.h, 135
 - MSXE3317_SinCos_Read, 59
- MSXExxxx__SinCosReadResponse, 76
 - dValue, 76
 - sResponse, 76
 - ulInfo01, 76
 - ulInfo02, 77
 - ulMeasureError, 76
 - ulValue, 76
- MSXExxxx__SinCosRelease
 - MSXE3317_public_doc.h, 133
 - MSXE3317_SinCos_Init, 57
- MSXExxxx__SinCosReleaseIndex
 - MSXE3317_public_doc.h, 137
 - MSXE3317_SinCos_Index, 62
- MSXExxxx__unsignedLongResponse, 77
 - sResponse, 77
 - ulValue, 77
- MSXExxxx__unsignedLongTimeStampResponse, 77
 - sResponse, 78
 - ulTimeStampHigh, 78
 - ulTimeStampLow, 78
 - ulValue, 78
- MXCommon__ApplyConfigurationBackupFile
 - Common_configuration, 23
 - MSXE3317_public_doc.h, 109
- MXCommon__ByteArrayResponse, 78
 - sArray, 78
 - sResponse, 78
- MXCommon__ChangePassword
 - Common_configuration, 24
 - MSXE3317_public_doc.h, 109
- MXCommon__DataseverRestart
 - Common_general, 8
 - MSXE3317_public_doc.h, 99
- MXCommon__FileResponse, 78
 - sArray, 79
 - sResponse, 79
 - ulEOF, 79
- MXCommon__GetAutoConfigurationFile
 - Common_autoconf, 19
 - MSXE3317_public_doc.h, 105
- MXCommon__GetAutoConfigurationFileResponse, 79
 - bArray, 79
 - sResponse, 79
 - ulEOF, 79
- MXCommon__GetClientConnections
 - Common_general, 6
- MSXE3317_public_doc.h, 96
- MXCommon__GetConfigurationBackupFile
 - Common_configuration, 23
 - MSXE3317_public_doc.h, 108
- MXCommon__GetEthernetLinksStates
 - Common_general, 9
 - MSXE3317_public_doc.h, 99
- MXCommon__GetEthernetLinksStatesResponse, 79
 - sPort0, 80
 - sPort1, 80
 - sResponse, 80
- MXCommon__GetHardwareTriggerFilterTime
 - Common_hardware_trigger, 12
 - MSXE3317_public_doc.h, 102
- MXCommon__GetHardwareTriggerFilterTimeResponse, 80
 - sResponse, 80
 - ulFilterTime, 80
 - ulInfo01, 80
 - ulInfo02, 80
- MXCommon__GetHardwareTriggerState
 - Common_hardware_trigger, 13
 - MSXE3317_public_doc.h, 102
- MXCommon__GetHardwareTriggerStateResponse, 80
 - sResponse, 81
 - ulInfo01, 81
 - ulInfo02, 81
 - ulState, 81
- MXCommon__GetHostname
 - Common_general, 5
 - MSXE3317_public_doc.h, 96
- MXCommon__GetModuleTemperatureValueAndStatus
 - Common_temperature, 10
 - MSXE3317_public_doc.h, 100
- MXCommon__GetModuleTemperatureValueAndStatusResponse, 81
 - dTemperatureValue, 82
 - sResponse, 82
 - ulInfo, 82
 - ulTemperatureStatus, 82
- MXCommon__GetModuleType
 - Common_general, 5
 - MSXE3317_public_doc.h, 95
- MXCommon__GetOptionInformation
 - CustomerOption, 28
 - MSXE3317_public_doc.h, 112
- MXCommon__GetStateIDFromName
 - MSXE3317_public_doc.h, 111
 - SystemStatemanagement, 26
- MXCommon__GetStateNameFromID
 - MSXE3317_public_doc.h, 111
 - SystemStatemanagement, 27

- MXCommon__GetSubsystemIDFromName
 - MSXE3317_public_doc.h, [110](#)
 - SystemStatemanagement, [26](#)
- MXCommon__GetSubsystemNameFromID
 - MSXE3317_public_doc.h, [111](#)
 - SystemStatemanagement, [26](#)
- MXCommon__GetSubSystemState
 - MSXE3317_public_doc.h, [110](#)
 - SystemStatemanagement, [25](#)
- MXCommon__GetSynchronizationStatus
 - MSXE3317_public_doc.h, [113](#)
 - Synchronisation, [29](#)
- MXCommon__GetTime
 - Common_time, [17](#)
 - MSXE3317_public_doc.h, [105](#)
- MXCommon__GetTimeResponse, [82](#)
 - sResponse, [82](#)
 - ulHighTime, [82](#)
 - ulLowTime, [82](#)
- MXCommon__GetUpTime
 - Common_time, [18](#)
 - MSXE3317_public_doc.h, [105](#)
- MXCommon__GetUpTimeResponse, [82](#)
 - sResponse, [83](#)
 - ulUpTime, [83](#)
- MXCommon__HardwareClockToSys
 - Common_time, [17](#)
 - MSXE3317_public_doc.h, [104](#)
- MXCommon__InitAndStartSynchroTimer
 - Common_synchrotimer, [21](#)
 - MSXE3317_public_doc.h, [107](#)
- MXCommon__Reboot
 - Common_general, [7](#)
 - MSXE3317_public_doc.h, [98](#)
- MXCommon__ResetAllIOFunctionalities
 - Common_general, [8](#)
 - MSXE3317_public_doc.h, [98](#)
- MXCommon__Response, [83](#)
 - iReturnValue, [83](#)
 - syserrno, [83](#)
- MXCommon__SetAutoConfigurationFile
 - Common_autoconf, [19](#)
 - MSXE3317_public_doc.h, [106](#)
- MXCommon__SetCustomerKey
 - Common_security, [15](#)
 - MSXE3317_public_doc.h, [103](#)
- MXCommon__SetHardwareTriggerFilterTime
 - Common_hardware_trigger, [12](#)
 - MSXE3317_public_doc.h, [101](#)
- MXCommon__SetHostname
 - Common_general, [6](#)
 - MSXE3317_public_doc.h, [96](#)
- MXCommon__SetModuleTemperatureWarningLevels
 - Common_temperature, [11](#)
 - MSXE3317_public_doc.h, [101](#)
- MXCommon__SetTime
 - Common_time, [16](#)
 - MSXE3317_public_doc.h, [103](#)
- MXCommon__SetToMaster
 - MSXE3317_public_doc.h, [112](#)
 - Synchronisation, [28](#)
- MXCommon__StartAutoConfiguration
 - Common_autoconf, [20](#)
 - MSXE3317_public_doc.h, [106](#)
- MXCommon__StopAndReleaseSynchroTimer
 - Common_synchrotimer, [21](#)
 - MSXE3317_public_doc.h, [108](#)
- MXCommon__Sterror
 - Common_general, [6](#)
 - MSXE3317_public_doc.h, [97](#)
- MXCommon__SysToHardwareClock
 - Common_time, [16](#)
 - MSXE3317_public_doc.h, [104](#)
- MXCommon__TestCustomerID
 - Common_security, [15](#)
 - MSXE3317_public_doc.h, [103](#)
- MXCommon__TestCustomerIDResponse, [83](#)
 - bCryptedValueArray, [84](#)
 - bValueArray, [84](#)
 - sResponse, [84](#)
- MXCommon__unsignedLongResponse, [84](#)
 - sResponse, [84](#)
 - ulValue, [84](#)
- sArray
 - MSXExxxx__FileResponse, [72](#)
 - MXCommon__ByteArrayResponse, [78](#)
 - MXCommon__FileResponse, [79](#)
- Set/Backup/Restore general system configuration, [22](#)
- sGetEthernetLinksStatesPort, [84](#)
 - ulDuplex, [85](#)
 - ulInfo1, [85](#)
 - ulInfo2, [85](#)
 - ulSpeed, [85](#)
 - ulState, [85](#)
- sPort0
 - MXCommon__-GetEthernetLinksStatesResponse, [80](#)
- sPort1
 - MXCommon__-GetEthernetLinksStatesResponse, [80](#)
- sResponse
 - MSXExxxx__-AcquisitionAutoRefreshGetValuesResponse, [69](#)
 - MSXExxxx__-AcquisitionGetChannelInfoResponse,

- 69
- MSXExxxx__-
 - DigitalIOGetInputsFilterConfigurationResponse, MXCommon__GetSubSystemState, 25
 - 71
- MSXExxxx__-
 - DigitalIOGetPortAvailableDirectionsResponse, MSXExxxx__-
 - 71
 - PressureGetConfigurationResponse, 73
- MSXExxxx__FileResponse, 72
- MSXExxxx__-
 - PressureGetConfigurationResponse, 73
- MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- MSXExxxx__SinCosInitResponse, 76
- MSXExxxx__SinCosReadResponse, 76
- MSXExxxx__unsignedLongResponse, 77
- MSXExxxx__-
 - unsignedLongTimeStampResponse, 78
- MXCommon__ByteArrayResponse, 78
- MXCommon__FileResponse, 79
- MXCommon__-
 - GetAutoConfigurationFileResponse, 79
- MXCommon__-
 - GetEthernetLinksStatesResponse, 80
- MXCommon__-
 - GetHardwareTriggerFilterTimeResponse, 80
- MXCommon__-
 - GetHardwareTriggerStateResponse, 81
- MXCommon__-
 - GetModuleTemperatureValueAndStatusResponse, 82
- MXCommon__GetTimeResponse, 82
- MXCommon__GetUpTimeResponse, 83
- MXCommon__TestCustomerIDResponse, 84
- MXCommon__unsignedLongResponse, 84
- Synchronisation
 - MXCommon__GetSynchronizationStatus, 29
 - MXCommon__SetToMaster, 28
- Synchronisation management, 28
- syserrno
 - DefaultResponse, 68
 - MSXExxxx__Response, 73
 - MXCommon__Response, 83
- System state management, 24
- SystemStatemanagement
 - MXCommon__GetStateIDFromName, 26
 - MXCommon__GetStateNameFromID, 27
 - MXCommon__GetSubsystemIDFromName, 26
- MXCommon__GetSubsystemNameFromID, 26
- MXCommon__GetSubSystemState, 25
- ulBaseSamplingRate
 - MSXExxxx__-
 - PressureGetConfigurationResponse, 73
- ulChannelIndex
 - MSXExxxx__-
 - AcquisitionGetChannelInfoResponse, 70
- ulChannelList
 - MSXExxxx__-
 - AcquisitionSequenceInitAndStartChannelListParam, 70
- ulCompareLogicMode
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- ulCompareLogicState
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- ulCompareLogicSynchro
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- ulCompareLogicValueFormat
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- ulCounterValue
 - MSXExxxx__-
 - AcquisitionAutoRefreshGetValuesResponse, 69
- ulDuplex
 - sGetEthernetLinksStatesPort, 85
- ulEOF
 - MSXExxxx__FileResponse, 72
 - MXCommon__FileResponse, 79
 - MXCommon__-
 - GetAutoConfigurationFileResponse, 79
- ulFilter
 - MSXExxxx__-
 - DigitalIOGetInputsFilterConfigurationResponse, 71
- ulFilterTime
 - MSXExxxx__-
 - DigitalIOGetInputsFilterConfigurationResponse, 71
 - MXCommon__-
 - GetHardwareTriggerFilterTimeResponse, 80

- 80
- ulHighTime
 - MXCommon__GetTimeResponse, 82
- ulHwPosition
 - MSXExxxx__-
 - AcquisitionGetChannelInfoResponse, 70
- ulIndexAction
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- ulIndexEdge
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- ulIndexState
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- ulInfo
 - MXCommon__-
 - GetModuleTemperatureValueAndStatusResponse, 82
- ulInfo01
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
 - MSXExxxx__SinCosInitResponse, 76
 - MSXExxxx__SinCosReadResponse, 76
 - MXCommon__-
 - GetHardwareTriggerFilterTimeResponse, 80
 - MXCommon__-
 - GetHardwareTriggerStateResponse, 81
- ulInfo02
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
 - MSXExxxx__SinCosInitResponse, 76
 - MSXExxxx__SinCosReadResponse, 77
 - MXCommon__-
 - GetHardwareTriggerFilterTimeResponse, 80
 - MXCommon__-
 - GetHardwareTriggerStateResponse, 81
- ulInfo1
 - sGetEthernetLinksStatesPort, 85
- ulInfo2
 - sGetEthernetLinksStatesPort, 85
- ulInitialisationState
 - MSXExxxx__-
 - SinCosGetConfigurationResponse,
- 75
- ulInputs
 - MSXExxxx__-
 - DigitalIOGetPortAvailableDirectionsResponse, 71
- ulLowTime
 - MXCommon__GetTimeResponse, 82
- ulMaxInputFrequency
 - MSXExxxx__SinCosInitResponse, 76
- ulMeasureError
 - MSXExxxx__SinCosReadResponse, 76
- ulOutputs
 - MSXExxxx__-
 - DigitalIOGetPortAvailableDirectionsResponse, 71
- ulResolution
 - MSXExxxx__-
 - SinCosGetConfigurationResponse, 75
- ulSpeed
 - sGetEthernetLinksStatesPort, 85
- ulState
 - MXCommon__-
 - GetHardwareTriggerStateResponse, 81
 - sGetEthernetLinksStatesPort, 85
- ulTemperatureStatus
 - MXCommon__-
 - GetModuleTemperatureValueAndStatusResponse, 82
- ulTimeStampHigh
 - MSXExxxx__-
 - AcquisitionAutoRefreshGetValuesResponse, 69
 - MSXExxxx__-
 - unsignedLongTimeStampResponse, 78
- ulTimeStampLow
 - MSXExxxx__-
 - AcquisitionAutoRefreshGetValuesResponse, 69
 - MSXExxxx__-
 - unsignedLongTimeStampResponse, 78
- ulType
 - MSXExxxx__-
 - AcquisitionGetChannelInfoResponse, 69
- ulUpTime
 - MXCommon__GetUpTimeResponse, 83
- ulValue
 - MSXExxxx__SinCosReadResponse, 76
 - MSXExxxx__unsignedLongResponse, 77

- MSXExxxx__-
 - unsignedLongTimeStampResponse,
78
- MXCommon__unsignedLongResponse, 84
- UnsignedLongArray, 85
 - __offset, 85
 - __ptr, 85
 - __size, 85
- UnsignedShortArray, 85
 - __offset, 86
 - __ptr, 86
 - __size, 86
- xsd__base64Binary, 86
 - __ptr, 86
 - __size, 86
- xsd__char
 - MSXE3317_public_doc.h, 95
- xsd__double
 - MSXE3317_public_doc.h, 95
- xsd__float
 - MSXE3317_public_doc.h, 95
- xsd__int
 - MSXE3317_public_doc.h, 95
- xsd__long
 - MSXE3317_public_doc.h, 95
- xsd__string
 - MSXE3317_public_doc.h, 95
- xsd__unsignedByte
 - MSXE3317_public_doc.h, 95
- xsd__unsignedInt
 - MSXE3317_public_doc.h, 95
- xsd__unsignedLong
 - MSXE3317_public_doc.h, 95
- xsd__unsignedShort
 - MSXE3317_public_doc.h, 95