

MSX-E170x soap api functions

Generated by Doxygen 1.7.1

Tue Apr 12 2016 10:54:44

Contents

1	MSX-E170x SOAP functions documentation	1
1.1	Introduction	1
1.1.1	Networking	1
1.1.2	Unavailable functions	1
2	Module Documentation	3
2.1	MSX-E170x compatibility functions	3
2.2	MSX-E17xx functions	3
2.3	MSX-E17xx multifunction functions	3
2.4	Common functions	3
2.5	Common general functions	4
2.5.1	Function Documentation	5
2.5.1.1	MXCommon__GetModuleType	5
2.5.1.2	MXCommon__GetHostname	6
2.5.1.3	MXCommon__SetHostname	6
2.5.1.4	MXCommon__GetClientConnections	6
2.5.1.5	MXCommon__Sterror	7
2.5.1.6	MXCommon__Reboot	8
2.5.1.7	MXCommon__ResetAllIOFunctionalities	8
2.5.1.8	MXCommon__DataseverRestart	9
2.5.1.9	MXCommon__GetEthernetLinksStates	9
2.6	Common temperature functions	10
2.6.1	Detailed Description	10
2.6.2	Function Documentation	11
2.6.2.1	MXCommon__GetModuleTemperatureValueAndStatus	11
2.6.2.2	MXCommon__SetModuleTemperatureWarningLevels	11
2.7	Common hardware trigger functions	12
2.7.1	Function Documentation	12

2.7.1.1	MXCommon__SetHardwareTriggerFilterTime	12
2.7.1.2	MXCommon__GetHardwareTriggerFilterTime	13
2.7.1.3	MXCommon__GetHardwareTriggerState	13
2.8	Common security functions	14
2.8.1	Detailed Description	14
2.8.2	Function Documentation	15
2.8.2.1	MXCommon__SetCustomerKey	15
2.8.2.2	MXCommon__TestCustomerID	15
2.9	Common time functions	16
2.9.1	Detailed Description	16
2.9.2	Function Documentation	16
2.9.2.1	MXCommon__SetTime	16
2.9.2.2	MXCommon__SysToHardwareClock	17
2.9.2.3	MXCommon__HardwareClockToSys	17
2.9.2.4	MXCommon__GetTime	18
2.9.2.5	MXCommon__GetUpTime	18
2.10	Common I/O auto configuration functions	18
2.10.1	Detailed Description	19
2.10.2	Function Documentation	19
2.10.2.1	MXCommon__GetAutoConfigurationFile	19
2.10.2.2	MXCommon__SetAutoConfigurationFile	20
2.10.2.3	MXCommon__StartAutoConfiguration	20
2.11	Common synchronisation timer functions	20
2.11.1	Function Documentation	21
2.11.1.1	MXCommon__InitAndStartSynchroTimer	21
2.11.1.2	MXCommon__StopAndReleaseSynchroTimer	22
2.12	Set/Backup/Restore general system configuration	22
2.12.1	Detailed Description	23
2.12.2	Function Documentation	23
2.12.2.1	MXCommon__GetConfigurationBackupFile	23
2.12.2.2	MXCommon__ApplyConfigurationBackupFile	24
2.12.2.3	MXCommon__ChangePassword	24
2.13	System state management	25
2.13.1	Detailed Description	25
2.13.2	Function Documentation	25
2.13.2.1	MXCommon__GetSubSystemState	25

2.13.2.2	MXCommon__GetSubsystemIDFromName	26
2.13.2.3	MXCommon__GetStateIDFromName	26
2.13.2.4	MXCommon__GetSubsystemNameFromID	27
2.13.2.5	MXCommon__GetStateNameFromID	27
2.14	Customer option management	27
2.14.1	Function Documentation	28
2.14.1.1	MXCommon__GetOptionInformation	28
2.15	Synchronisation management	28
2.15.1	Function Documentation	28
2.15.1.1	MXCommon__SetToMaster	28
2.15.1.2	MXCommon__GetSynchronizationStatus	29
2.16	input filter Filter management	29
2.16.1	Function Documentation	30
2.16.1.1	MXCommon__SetFilterChannels	30
2.17	MSX-E170x digital I/O functions	30
2.17.1	Function Documentation	31
2.17.1.1	MSXE170x__DigitalIOGetNumber	31
2.17.1.2	MSXE170x__DigitalIOInitPortConfiguration	31
2.17.1.3	MSXE170x__DigitalIOReadChannelValue	32
2.17.1.4	MSXE170x__DigitalIOReadAllChannelsValue	32
2.17.1.5	MSXE170x__DigitalIOWriteChannelValue	32
2.17.1.6	MSXE170x__DigitalIOWriteAllChannelsValue	33
2.17.1.7	MSXE170x__DigitalIOReleasePortConfiguration	33
2.17.1.8	MSXE170x__DigitalIOTestShortCircuit	34
2.17.1.9	MSXE170x__DigitalIORearmShortCircuit	34
2.18	MSX-E170x IO watchdog functions	34
2.18.1	Function Documentation	35
2.18.1.1	MSXE170x__IOWatchdogInitAndStart	35
2.18.1.2	MSXE170x__IOWatchdogStopAndRelease	35
2.18.1.3	MSXE170x__IOWatchdogGetStatusAndValue	36
2.19	MSX-E170x multifunction common functions	36
2.19.1	Function Documentation	37
2.19.1.1	MSXE170x__MFCommonSetInputsFilter	37
2.19.1.2	MSXE170x__MFCommonReferenceVoltageActivation	38
2.20	MSX-E170x incremental counter functions	38
2.20.1	Function Documentation	40

2.20.1.1	MSXE170x__MFIncCounterGetNumber	40
2.20.1.2	MSXE170x__MFIncCounterInit	40
2.20.1.3	MSXE170x__MFIncCounterRelease	41
2.20.1.4	MSXE170x__MFIncCounterSetFIFO0Level	41
2.20.1.5	MSXE170x__MFIncCounterRead32BitValue	41
2.20.1.6	MSXE170x__MFIncCounterWrite32BitValue	42
2.20.1.7	MSXE170x__MFIncCounterClear	42
2.20.1.8	MSXE170x__MFIncCounterInitAndEnableCompareLogic	43
2.20.1.9	MSXE170x__MFIncCounterDisableAndReleaseCompareLogic	44
2.20.1.10	MSXE170x__MFIncCounterInitAndEnableLatchRegister	44
2.20.1.11	MSXE170x__MFIncCounterDisableAndReleaseLatchRegister	45
2.20.1.12	MSXE170x__MFIncCounterInitAndEnableFrequencyMeasurement	45
2.20.1.13	MSXE170x__MFIncCounterDisableAndReleaseFrequencyMeasurement	46
2.20.1.14	MSXE170x__MFIncCounterInitAndEnableIndex	47
2.20.1.15	MSXE170x__MFIncCounterDisableAndReleaseIndex	48
2.21	MSX-E17xx digital I/O functions	48
2.21.1	Function Documentation	49
2.21.1.1	MSXE17xx__DigitalIOGetNumber	49
2.21.1.2	MSXE17xx__DigitalIOInitPortConfiguration	49
2.21.1.3	MSXE17xx__DigitalIOReadChannelValue	50
2.21.1.4	MSXE17xx__DigitalIOReadAllChannelsValue	50
2.21.1.5	MSXE17xx__DigitalIOWriteChannelValue	51
2.21.1.6	MSXE17xx__DigitalIOWriteAllChannelsValue	51
2.21.1.7	MSXE17xx__DigitalIOReleasePortConfiguration	52
2.21.1.8	MSXE17xx__DigitalIOTestShortCircuit	52
2.21.1.9	MSXE17xx__DigitalIORearmShortCircuit	52
2.22	MSX-E17xx IO watchdog functions	53
2.22.1	Function Documentation	53
2.22.1.1	MSXE17xx__IOWatchdogInitAndStart	53
2.22.1.2	MSXE17xx__IOWatchdogStopAndRelease	54
2.22.1.3	MSXE17xx__IOWatchdogGetStatusAndValue	54
2.23	MSX-E17xx multifunction common functions	55
2.23.1	Function Documentation	55
2.23.1.1	MSXE17xx__MFCommonGetSubModuleFunctionality	55
2.23.1.2	MSXE17xx__MFCommonSetInputsFilter	56
2.23.1.3	MSXE17xx__MFCommonReferenceVoltageActivation	57

2.23.1.4	MSXE17xx__MFCommonEnableDisableTriggerGate	57
2.23.1.5	MSXE17xx__MFCommonSetFIFO0Level	58
2.24	MSX-E17xx pulse width modulation	59
2.24.1	Function Documentation	60
2.24.1.1	MSXE17xx__MFPWMGetNumber	60
2.24.1.2	MSXE17xx__MFPWMInitAndEnable	60
2.24.1.3	MSXE17xx__MFPWMSetNewTiming	62
2.24.1.4	MSXE17xx__MFPWMDisable	63
2.24.1.5	MSXE17xx__MFPWMDisableAndRelease	64
2.24.1.6	MSXE17xx__MFPWMGetConfiguration	65
2.24.1.7	MSXE17xx__MFPWMGetState	66
2.25	MSX-E17xx incremental counter functions	67
2.25.1	Function Documentation	70
2.25.1.1	MSXE17xx__MFIncCounterGetNumber	70
2.25.1.2	MSXE17xx__MFIncCounterInit	70
2.25.1.3	MSXE17xx__MFIncCounterRelease	71
2.25.1.4	MSXE17xx__MFIncCounterRead32BitValue	71
2.25.1.5	MSXE17xx__MFIncCounterWrite32BitValue	72
2.25.1.6	MSXE17xx__MFIncCounterClear	72
2.25.1.7	MSXE17xx__MFIncCounterInitAndEnableCompareLogic	73
2.25.1.8	MSXE17xx__MFIncCounterDisableAndReleaseCompareLogic	74
2.25.1.9	MSXE17xx__MFIncCounterInitHardwareTrigger	74
2.25.1.10	MSXE17xx__MFIncCounterReleaseHardwareTrigger	75
2.25.1.11	MSXE17xx__MFIncCounterInitIndex	76
2.25.1.12	MSXE17xx__MFIncCounterReleaseIndex	76
2.25.1.13	MSXE17xx__MFIncCounterInitAndEnableLatch	77
2.25.1.14	MSXE17xx__MFIncCounterDisableAndReleaseLatch	78
2.25.1.15	MSXE17xx__MFIncCounterInitAndEnableClear	79
2.25.1.16	MSXE17xx__MFIncCounterDisableAndReleaseClear	80
2.25.1.17	MSXE17xx__MFIncCounterInitAndEnableLatchRegister	80
2.25.1.18	MSXE17xx__MFIncCounterDisableAndReleaseLatchRegister	81
2.25.1.19	MSXE17xx__MFIncCounterInitAndEnableFrequencyMeasurement	82
2.25.1.20	MSXE17xx__MFIncCounterDisableAndReleaseFrequencyMeasurement	83
2.25.1.21	MSXE17xx__MFIncCounterInitAndEnableIndex	83
2.25.1.22	MSXE17xx__MFIncCounterDisableAndReleaseIndex	84

3 Data Structure Documentation

85

3.1	ByteArray Struct Reference	85
3.1.1	Field Documentation	85
3.1.1.1	__ptr	85
3.1.1.2	__size	85
3.1.1.3	__offset	85
3.2	DefaultResponse Struct Reference	85
3.2.1	Field Documentation	86
3.2.1.1	iReturnValue	86
3.2.1.2	syserrno	86
3.3	MSXE170x__DigitalIOGetNumberResponse Struct Reference	86
3.3.1	Detailed Description	86
3.3.2	Field Documentation	87
3.3.2.1	sResponse	87
3.3.2.2	ulNumberOfDigitalIO	87
3.4	MSXE170x__IOWatchdogGetStatusAndValueResponse Struct Reference	87
3.4.1	Field Documentation	87
3.4.1.1	sResponse	87
3.4.1.2	ulStatus	87
3.4.1.3	ulValue	87
3.4.1.4	ulInfo	87
3.5	MSXE170x__MFIncCounterGetNumberResponse Struct Reference	87
3.5.1	Detailed Description	88
3.5.2	Field Documentation	88
3.5.2.1	sResponse	88
3.5.2.2	ulNumberOfCounter	88
3.6	MSXE170x__Response Struct Reference	88
3.6.1	Field Documentation	88
3.6.1.1	iReturnValue	88
3.6.1.2	syserrno	88
3.7	MSXE170x__unsignedLongResponse Struct Reference	88
3.7.1	Field Documentation	89
3.7.1.1	sResponse	89
3.7.1.2	ulValue	89
3.8	MSXE170x__unsignedLongTimeStampResponse Struct Reference	89
3.8.1	Field Documentation	89
3.8.1.1	sResponse	89

3.8.1.2	ulValue	89
3.8.1.3	ulTimeStampLow	89
3.8.1.4	ulTimeStampHigh	89
3.9	MSXE17xx__DigitalIOGetNumberResponse Struct Reference	89
3.9.1	Field Documentation	90
3.9.1.1	sResponse	90
3.9.1.2	ulNumberOfDigitalIO	90
3.10	MSXE17xx__IOWatchdogGetStatusAndValueResponse Struct Reference	90
3.10.1	Field Documentation	90
3.10.1.1	sResponse	90
3.10.1.2	ulStatus	90
3.10.1.3	ulValue	90
3.10.1.4	ulInfo	90
3.11	MSXE17xx__MFIncCounterGetNumberResponse Struct Reference	90
3.11.1	Field Documentation	91
3.11.1.1	sResponse	91
3.11.1.2	ulNumberOfCounter	91
3.12	MSXE17xx__MFPWMGetConfigurationResponse Struct Reference	91
3.12.1	Field Documentation	92
3.12.1.1	sResponse	92
3.12.1.2	ulInitialised	92
3.12.1.3	ulTimeBase	92
3.12.1.4	ulLowTime	92
3.12.1.5	ulHighTime	92
3.12.1.6	ulEnabled	93
3.12.1.7	ulStartLevel	93
3.12.1.8	ulExternGate	93
3.12.1.9	ulStopMode	93
3.12.1.10	ulStopLevel	93
3.12.1.11	ulSynchroOut	93
3.12.1.12	ulDataFrame	94
3.12.1.13	ulInfo01	94
3.12.1.14	ulInfo02	94
3.13	MSXE17xx__MFPWMGetStateResponse Struct Reference	94
3.13.1	Field Documentation	95
3.13.1.1	sResponse	95

3.13.1.2	ulInitialised	95
3.13.1.3	ulEnabled	95
3.13.1.4	ulProgress	95
3.13.1.5	ulOutput	96
3.13.1.6	ulDiffInput	96
3.13.1.7	ulTriggerInput	96
3.13.1.8	ulCounter	96
3.13.1.9	ulInfo01	96
3.13.1.10	ulInfo02	96
3.14	MSXE17xx__Response Struct Reference	96
3.14.1	Field Documentation	96
3.14.1.1	iReturnValue	96
3.14.1.2	syserrno	97
3.15	MSXE17xx__unsignedLongResponse Struct Reference	97
3.15.1	Field Documentation	97
3.15.1.1	sResponse	97
3.15.1.2	ulValue	97
3.16	MSXE17xx__unsignedLongTimeStampResponse Struct Reference	97
3.16.1	Field Documentation	98
3.16.1.1	sResponse	98
3.16.1.2	ulValue	98
3.16.1.3	ulTimeStampLow	98
3.16.1.4	ulTimeStampHigh	98
3.17	MXCommon__ByteArrayResponse Struct Reference	98
3.17.1	Field Documentation	98
3.17.1.1	sResponse	98
3.17.1.2	sArray	98
3.18	MXCommon__FileResponse Struct Reference	98
3.18.1	Field Documentation	99
3.18.1.1	sResponse	99
3.18.1.2	sArray	99
3.18.1.3	ulEOF	99
3.19	MXCommon__GetAutoConfigurationFileResponse Struct Reference	99
3.19.1	Field Documentation	99
3.19.1.1	sResponse	99
3.19.1.2	bArray	99

3.19.1.3	ulEOF	99
3.20	MXCommon__GetEthernetLinksStatesResponse Struct Reference	99
3.20.1	Field Documentation	100
3.20.1.1	sResponse	100
3.20.1.2	sPort0	100
3.20.1.3	sPort1	100
3.21	MXCommon__GetHardwareTriggerFilterTimeResponse Struct Reference	100
3.21.1	Field Documentation	100
3.21.1.1	sResponse	100
3.21.1.2	ulFilterTime	100
3.21.1.3	ulInfo01	100
3.21.1.4	ulInfo02	100
3.22	MXCommon__GetHardwareTriggerStateResponse Struct Reference	100
3.22.1	Field Documentation	101
3.22.1.1	sResponse	101
3.22.1.2	ulState	101
3.22.1.3	ulInfo01	101
3.22.1.4	ulInfo02	101
3.23	MXCommon__GetModuleTemperatureValueAndStatusResponse Struct Reference	101
3.23.1	Field Documentation	102
3.23.1.1	sResponse	102
3.23.1.2	dTemperatureValue	102
3.23.1.3	ulTemperatureStatus	102
3.23.1.4	ulInfo	102
3.24	MXCommon__GetTimeResponse Struct Reference	102
3.24.1	Field Documentation	102
3.24.1.1	sResponse	102
3.24.1.2	ulLowTime	102
3.24.1.3	ulHighTime	102
3.25	MXCommon__GetUpTimeResponse Struct Reference	102
3.25.1	Field Documentation	103
3.25.1.1	sResponse	103
3.25.1.2	ulUpTime	103
3.26	MXCommon__Response Struct Reference	103
3.26.1	Field Documentation	103
3.26.1.1	iReturnValue	103

3.26.1.2	syserrno	103
3.27	MXCommon__TestCustomerIDResponse Struct Reference	103
3.27.1	Field Documentation	104
3.27.1.1	sResponse	104
3.27.1.2	bValueArray	104
3.27.1.3	bCryptedValueArray	104
3.28	MXCommon__unsignedLongResponse Struct Reference	104
3.28.1	Field Documentation	104
3.28.1.1	sResponse	104
3.28.1.2	ulValue	104
3.29	sGetEthernetLinksStatesPort Struct Reference	104
3.29.1	Field Documentation	105
3.29.1.1	ulState	105
3.29.1.2	ulSpeed	105
3.29.1.3	ulDuplex	105
3.29.1.4	ulInfo1	105
3.29.1.5	ulInfo2	105
3.30	UnsignedLongArray Struct Reference	105
3.30.1	Field Documentation	105
3.30.1.1	__ptr	105
3.30.1.2	__size	105
3.30.1.3	__offset	105
3.31	UnsignedShortArray Struct Reference	105
3.31.1	Field Documentation	106
3.31.1.1	__ptr	106
3.31.1.2	__size	106
3.31.1.3	__offset	106
3.32	xsd__base64Binary Struct Reference	106
3.32.1	Field Documentation	106
3.32.1.1	__ptr	106
3.32.1.2	__size	106
4	File Documentation	107
4.1	MSXE170x_public_doc.h File Reference	107
4.1.1	Define Documentation	120
4.1.1.1	MSXE170X_COUNTER_QUADRUPLE_MODE	120
4.1.1.2	MSXE170X_COUNTER_DOUBLE_MODE	120

4.1.1.3	MSXE170X_COUNTER_SIMPLE_MODE	120
4.1.1.4	MSXE170X_COUNTER_DIRECT_MODE	120
4.1.1.5	MSXE170X_COUNTER_HYSTERESIS_ON	120
4.1.1.6	MSXE170X_COUNTER_HYSTERESIS_OFF	121
4.1.1.7	MSXE170X_COUNTER_INCREMENT	121
4.1.1.8	MSXE170X_COUNTER_DECREMENT	121
4.1.1.9	MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_- COUNTER	121
4.1.1.10	MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_- COUNTER	121
4.1.1.11	MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER	121
4.1.1.12	MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER	121
4.1.2	Typedef Documentation	121
4.1.2.1	xsd_string	121
4.1.2.2	xsd_char	121
4.1.2.3	xsd_float	121
4.1.2.4	xsd_double	121
4.1.2.5	xsd_int	121
4.1.2.6	xsd_long	121
4.1.2.7	xsd_unsignedByte	121
4.1.2.8	xsd_unsignedInt	121
4.1.2.9	xsd_unsignedShort	121
4.1.2.10	xsd_unsignedLong	121
4.1.3	Function Documentation	121
4.1.3.1	MXCommon__GetModuleType	121
4.1.3.2	MXCommon__GetHostname	122
4.1.3.3	MXCommon__SetHostname	122
4.1.3.4	MXCommon__GetClientConnections	122
4.1.3.5	MXCommon__Sterror	123
4.1.3.6	MXCommon__Reboot	124
4.1.3.7	MXCommon__ResetAllIOFunctionalities	124
4.1.3.8	MXCommon__DataseverRestart	125
4.1.3.9	MXCommon__GetEthernetLinksStates	125
4.1.3.10	MXCommon__GetModuleTemperatureValueAndStatus	126
4.1.3.11	MXCommon__SetModuleTemperatureWarningLevels	127
4.1.3.12	MXCommon__SetHardwareTriggerFilterTime	127
4.1.3.13	MXCommon__GetHardwareTriggerFilterTime	128

4.1.3.14	MXCommon__GetHardwareTriggerState	128
4.1.3.15	MXCommon__SetCustomerKey	129
4.1.3.16	MXCommon__TestCustomerID	129
4.1.3.17	MXCommon__SetTime	129
4.1.3.18	MXCommon__SysToHardwareClock	130
4.1.3.19	MXCommon__HardwareClockToSys	130
4.1.3.20	MXCommon__GetTime	131
4.1.3.21	MXCommon__GetUpTime	131
4.1.3.22	MXCommon__GetAutoConfigurationFile	131
4.1.3.23	MXCommon__SetAutoConfigurationFile	132
4.1.3.24	MXCommon__StartAutoConfiguration	132
4.1.3.25	MXCommon__InitAndStartSynchroTimer	132
4.1.3.26	MXCommon__StopAndReleaseSynchroTimer	133
4.1.3.27	MXCommon__GetConfigurationBackupFile	134
4.1.3.28	MXCommon__ApplyConfigurationBackupFile	135
4.1.3.29	MXCommon__ChangePassword	135
4.1.3.30	MXCommon__GetSubSystemState	136
4.1.3.31	MXCommon__GetSubsystemIDFromName	136
4.1.3.32	MXCommon__GetStateIDFromName	136
4.1.3.33	MXCommon__GetSubsystemNameFromID	137
4.1.3.34	MXCommon__GetStateNameFromID	137
4.1.3.35	MXCommon__GetOptionInformation	138
4.1.3.36	MXCommon__SetToMaster	138
4.1.3.37	MXCommon__GetSynchronizationStatus	138
4.1.3.38	MXCommon__SetFilterChannels	139
4.1.3.39	MSXE170x__DigitalIOGetNumber	139
4.1.3.40	MSXE170x__DigitalIOInitPortConfiguration	139
4.1.3.41	MSXE170x__DigitalIOReadChannelValue	140
4.1.3.42	MSXE170x__DigitalIOReadAllChannelsValue	140
4.1.3.43	MSXE170x__DigitalIOWriteChannelValue	141
4.1.3.44	MSXE170x__DigitalIOWriteAllChannelsValue	141
4.1.3.45	MSXE170x__DigitalIOReleasePortConfiguration	141
4.1.3.46	MSXE170x__DigitalIOTestShortCircuit	142
4.1.3.47	MSXE170x__DigitalIORearmShortCircuit	142
4.1.3.48	MSXE170x__IOWatchdogInitAndStart	143
4.1.3.49	MSXE170x__IOWatchdogStopAndRelease	143

4.1.3.50	MSXE170x__IOWatchdogGetStatusAndValue	143
4.1.3.51	MSXE170x__MFCommonSetInputsFilter	144
4.1.3.52	MSXE170x__MFCommonReferenceVoltageActivation	145
4.1.3.53	MSXE170x__MFIncCounterGetNumber	146
4.1.3.54	MSXE170x__MFIncCounterInit	146
4.1.3.55	MSXE170x__MFIncCounterRelease	147
4.1.3.56	MSXE170x__MFIncCounterSetFIFO0Level	147
4.1.3.57	MSXE170x__MFIncCounterRead32BitValue	147
4.1.3.58	MSXE170x__MFIncCounterWrite32BitValue	148
4.1.3.59	MSXE170x__MFIncCounterClear	148
4.1.3.60	MSXE170x__MFIncCounterInitAndEnableCompareLogic	149
4.1.3.61	MSXE170x__MFIncCounterDisableAndReleaseCompareLogic	149
4.1.3.62	MSXE170x__MFIncCounterInitAndEnableLatchRegister	150
4.1.3.63	MSXE170x__MFIncCounterDisableAndReleaseLatchRegister	151
4.1.3.64	MSXE170x__MFIncCounterInitAndEnableFrequencyMeasurement	151
4.1.3.65	MSXE170x__MFIncCounterDisableAndReleaseFrequencyMeasurement	152
4.1.3.66	MSXE170x__MFIncCounterInitAndEnableIndex	152
4.1.3.67	MSXE170x__MFIncCounterDisableAndReleaseIndex	153
4.1.3.68	MSXE17xx__DigitalIOGetNumber	154
4.1.3.69	MSXE17xx__DigitalIOInitPortConfiguration	154
4.1.3.70	MSXE17xx__DigitalIOReadChannelValue	155
4.1.3.71	MSXE17xx__DigitalIOReadAllChannelsValue	155
4.1.3.72	MSXE17xx__DigitalIOWriteChannelValue	155
4.1.3.73	MSXE17xx__DigitalIOWriteAllChannelsValue	156
4.1.3.74	MSXE17xx__DigitalIOReleasePortConfiguration	156
4.1.3.75	MSXE17xx__DigitalIOTestShortCircuit	157
4.1.3.76	MSXE17xx__DigitalIORearmShortCircuit	157
4.1.3.77	MSXE17xx__IOWatchdogInitAndStart	157
4.1.3.78	MSXE17xx__IOWatchdogStopAndRelease	158
4.1.3.79	MSXE17xx__IOWatchdogGetStatusAndValue	158
4.1.3.80	MSXE17xx__MFCommonGetSubModuleFunctionality	159
4.1.3.81	MSXE17xx__MFCommonSetInputsFilter	159
4.1.3.82	MSXE17xx__MFCommonReferenceVoltageActivation	160
4.1.3.83	MSXE17xx__MFCommonEnableDisableTriggerGate	161
4.1.3.84	MSXE17xx__MFCommonSetFIFO0Level	162
4.1.3.85	MSXE17xx__MFPWMGetNumber	162

4.1.3.86	MSXE17xx__MFPWMInitAndEnable	163
4.1.3.87	MSXE17xx__MFPWMSetNewTiming	165
4.1.3.88	MSXE17xx__MFPWMDisable	166
4.1.3.89	MSXE17xx__MFPWMDisableAndRelease	166
4.1.3.90	MSXE17xx__MFPWMGetConfiguration	167
4.1.3.91	MSXE17xx__MFPWMGetState	169
4.1.3.92	MSXE17xx__MFIncCounterGetNumber	170
4.1.3.93	MSXE17xx__MFIncCounterInit	170
4.1.3.94	MSXE17xx__MFIncCounterRelease	171
4.1.3.95	MSXE17xx__MFIncCounterRead32BitValue	172
4.1.3.96	MSXE17xx__MFIncCounterWrite32BitValue	172
4.1.3.97	MSXE17xx__MFIncCounterClear	173
4.1.3.98	MSXE17xx__MFIncCounterInitAndEnableCompareLogic	173
4.1.3.99	MSXE17xx__MFIncCounterDisableAndReleaseCompareLogic	174
4.1.3.100	MSXE17xx__MFIncCounterInitHardwareTrigger	175
4.1.3.101	MSXE17xx__MFIncCounterReleaseHardwareTrigger	175
4.1.3.102	MSXE17xx__MFIncCounterInitIndex	176
4.1.3.103	MSXE17xx__MFIncCounterReleaseIndex	177
4.1.3.104	MSXE17xx__MFIncCounterInitAndEnableLatch	177
4.1.3.105	MSXE17xx__MFIncCounterDisableAndReleaseLatch	178
4.1.3.106	MSXE17xx__MFIncCounterInitAndEnableClear	179
4.1.3.107	MSXE17xx__MFIncCounterDisableAndReleaseClear	180
4.1.3.108	MSXE17xx__MFIncCounterInitAndEnableLatchRegister	181
4.1.3.109	MSXE17xx__MFIncCounterDisableAndReleaseLatchRegister	181
4.1.3.110	MSXE17xx__MFIncCounterInitAndEnableFrequencyMeasurement	182
4.1.3.111	MSXE17xx__MFIncCounterDisableAndReleaseFrequencyMeasurement	183
4.1.3.112	MSXE17xx__MFIncCounterInitAndEnableIndex	184
4.1.3.113	MSXE17xx__MFIncCounterDisableAndReleaseIndex	184

Chapter 1

MSX-E170x SOAP functions documentation

MainRevision:

1.1 Introduction

1.1.1 Networking

The module is accessed via a TCP/IP socket:

The Ethernet I/O module has the following two servers:

- Command server (SOAP) > to send commands (initialisation, etc.)
- Data server (TCP socket) > to obtain the values of the acquisition
- Event server (TCP socket) > to obtain event from the module

MSX-E170x server access information:

- SOAP server: Port number 5555
- Data server: Port number 8989
- Event server: Port number 6363

See the "Modules" chapter to view the functions

1.1.2 Unavailable functions

The following functions belong to the common MSX-E interface and are documented, but not actually available on the module.

They would return an error if called.

Functions relative to AutoConfiguration:

- [MXCommon__GetAutoConfigurationFile\(\)](#)
- [MXCommon__SetAutoConfigurationFile\(\)](#)
- [MXCommon__StartAutoConfiguration\(\)](#)

Chapter 2

Module Documentation

2.1 MSX-E170x compatibility functions

Modules

- [MSX-E170x digital I/O functions](#)
- [MSX-E170x IO watchdog functions](#)
- [MSX-E170x multifunction common functions](#)
- [MSX-E170x incremental counter functions](#)

2.2 MSX-E17xx functions

Modules

- [MSX-E17xx multifunction functions](#)
- [MSX-E17xx digital I/O functions](#)
- [MSX-E17xx IO watchdog functions](#)

2.3 MSX-E17xx multifunction functions

Modules

- [MSX-E17xx multifunction common functions](#)
- [MSX-E17xx pulse width modulation](#)
- [MSX-E17xx incremental counter functions](#)

2.4 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.

- [input filter Filter management](#)

Manages the analog input filters in the system.

2.5 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__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.

2.5.1 Function Documentation

2.5.1.1 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

2.5.1.2 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

2.5.1.3 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

2.5.1.4 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

2.5.1.5 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

2.5.1.6 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

2.5.1.7 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

2.5.1.8 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.

2.5.1.9 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.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

2.6 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.

2.6.1 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.

2.6.2 Function Documentation

2.6.2.1 `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__Strerror\(\)](#).
 - 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

2.6.2.2 `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__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.7 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.

2.7.1 Function Documentation

2.7.1.1 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**2.7.1.2 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**2.7.1.3 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__Sterror\(\)](#).

Return values*SOAP_OK* SOAP call success*otherwise* SOAP protocol error

2.8 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).

2.8.1 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.

2.8.2 Function Documentation

2.8.2.1 `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

2.8.2.2 `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

2.9 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).

2.9.1 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.

2.9.2 Function Documentation

2.9.2.1 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__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.9.2.2 `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.

2.9.2.3 `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.

2.9.2.4 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__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

2.9.2.5 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__Sterror\(\)](#).
 - ulUpTime : Number of seconds since the last boot of the system.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.10 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

2.10.1 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.

2.10.2 Function Documentation

2.10.2.1 `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

2.10.2.2 `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__Sterror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.10.2.3 `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__Sterror\(\)](#).
 - sArray : message returned by the auto configuration start

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.11 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)

2.11.1 Function Documentation

2.11.1.1 `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__Sterror\(\)](#). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

2.11.1.2 `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

2.12 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.

2.12.1 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"

2.12.2 Function Documentation

2.12.2.1 `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__Strerror\(\)](#).
 - `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;
}
*
```

2.12.2.2 `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__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

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

2.12.2.3 `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__Strerror\(\)](#).
 - 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.

2.13 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".

2.13.1 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.

2.13.2 Function Documentation

2.13.2.1 `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__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

2.13.2.2 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

2.13.2.3 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

2.13.2.4 `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

2.13.2.5 `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

2.14 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.

2.14.1 Function Documentation

2.14.1.1 `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

2.15 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.

2.15.1 Function Documentation

2.15.1.1 `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

2.15.1.2 `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

2.16 input filter Filter management

Manages the analog input filters in the system.

Functions

- `int MXCommon__SetFilterChannels (struct xsd__base64Binary *ChannelList, struct MXCommon__Response *Response)`

This function sets or resets a filter to a channel.

2.16.1 Function Documentation

2.16.1.1 `int MXCommon_SetFilterChannels (struct xsd__base64Binary * ChannelList, struct MXCommon_Response * Response)`

Parameters

[in] **ChannelList** Each index of the array represents a channel. A filter can be affected to each channel. If FilterID = 0, no filter is set (the filter is disabled on the corresponding channel). e.g.: ChannelList[0] = FilterID // Set FilterID on channel 0.

[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

2.17 MSX-E170x digital I/O functions

Data Structures

- struct [MSXE170x__DigitalIOGetNumberResponse](#)
Returns the number of digital IO available on the module.

Functions

- int [MSXE170x__DigitalIOGetNumber](#) (void __, struct [MSXE170x__DigitalIOGetNumberResponse](#) *Response)
- int [MSXE170x__DigitalIOInitPortConfiguration](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulPortConfiguration, struct [MSXE170x__Response](#) *Response)
Initialise a digital i/o port (2 channels).
- int [MSXE170x__DigitalIOReadChannelValue](#) (xsd__unsignedLong ulChannel, struct [MSXE170x__unsignedLongResponse](#) *Response)
Read a digital i/o channel value.
- int [MSXE170x__DigitalIOReadAllChannelsValue](#) (void __, struct [MSXE170x__unsignedLongResponse](#) *Response)
Read all digital i/o channels value.If channel is configured as output, then this function return the status of the output.
- int [MSXE170x__DigitalIOWriteChannelValue](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulChannelValue, struct [MSXE170x__Response](#) *Response)
write a digital i/o channel value

- `int MSXE170x__DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE170x__Response *Response)`
write all digital i/o channels value
- `int MSXE170x__DigitalIOReleasePortConfiguration (xsd__unsignedLong ulPort, struct MSXE170x__Response *Response)`
Release a digital i/o port (2 channels).
- `int MSXE170x__DigitalIOTestShortCircuit (xsd__unsignedLong ulOption, struct MSXE170x__Response *Response)`
Test short circuit status.
- `int MSXE170x__DigitalIORearmShortCircuit (xsd__unsignedLong ulOption, struct MSXE170x__Response *Response)`
Rearm digital outputs after a short circuit happened.

2.17.1 Function Documentation

2.17.1.1 `int MSXE170x__DigitalIOGetNumber (void * _, struct MSXE170x__DigitalIOGetNumberResponse * Response)`

2.17.1.2 `int MSXE170x__DigitalIOInitPortConfiguration (xsd__unsignedLong ulPort, xsd__unsignedLong ulPortConfiguration, struct MSXE170x__Response * Response)`

Parameters

[in] ***ulPort*** : Index of the digital i/o port (0 to 7)

[in] ***ulPortConfiguration*** : Define the port configuration

- 0 : input
- 1 : output

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Digital i/o port selection error
- -3: Port configuration selection error
- -100: Init dig i/o port kernel function error

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

Returns

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

2.17.1.3 `int MSXE170x__DigitalIOReadChannelValue (xsd__unsignedLong ulChannel, struct MSXE170x__unsignedLongResponse * Response)`

Parameters

[in] *ulChannel* : Index of the digital i/o channel (0 to 15)

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o channel selection error
- -100: Read dig i/o channel value kernel function error

syserrno : system-error code (the value of the libc "errno" code) *ulValue* : i/o channel value:

- 0
- 1

Returns

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

2.17.1.4 `int MSXE170x__DigitalIOReadAllChannelsValue (void * _, struct MSXE170x__unsignedLongResponse * Response)`

Parameters

[in] *_* : no input parameter

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -100: Read dig i/o channel value kernel function error

syserrno : system-error code (the value of the libc "errno" code) *ulValue* : i/o channels value(each bit correspond to one channel)

Returns

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

2.17.1.5 `int MSXE170x__DigitalIOWriteChannelValue (xsd__unsignedLong ulChannel, xsd__unsignedLong ulChannelValue, struct MSXE170x__Response * Response)`

Parameters

[in] *ulChannel* : Index of the digital i/o channel (0 to 15)

[in] *ulChannelValue* : Channel value

- 0
- 1

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o channel selection error
- -3: Channel value error
- -100: Write dig i/o channel value kernel function error

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

Returns

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

2.17.1.6 int MSXE170x_DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE170x__Response * Response)

Parameters

[in] **ulChannelValue** : Channels value (each bit corresponds to a channel)

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Channels value error
- -100: Write dig i/o channel value kernel function error

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

Returns

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

2.17.1.7 int MSXE170x_DigitalIOReleasePortConfiguration (xsd__unsignedLong ulPort, struct MSXE170x__Response * Response)

Parameters

[in] **ulPort** : Index of the digital i/o port (0 to 7)

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o port selection error

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

Returns

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

2.17.1.8 int MSXE170x__DigitalIOTestShortCircuit (xsd__unsignedLong ulOption, struct MSXE170x__unsignedLongResponse * Response)

Parameters

[in] *ulOption* : reserved

[out] *Response* :

iReturnValue :

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

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

ulValue : short circuit status: from 0 to 0xffff, one bit for each output

- 0 : no short circuit
- 1 : short circuit

Returns

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

2.17.1.9 int MSXE170x__DigitalIORearmShortCircuit (xsd__unsignedLong ulOption, struct MSXE170x__Response * Response)

Parameters

[in] *ulOption* : reserved

[out] *Response* :

iReturnValue :

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

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

Returns

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

2.18 MSX-E170x IO watchdog functions

Data Structures

- struct [MSXE170x__IOWatchdogGetStatusAndValueResponse](#)

Functions

- int [MSXE170x__IOWatchdogInitAndStart](#) (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct [MSXE170x__Response](#) *Response)

Init and start the digital output IO watchdog.

- `int MSXE170x__IOWatchdogStopAndRelease (xsd__unsignedLong ulOption, struct MSXE170x__Response *Response)`

Stop and release the digital output watchdog.

- `int MSXE170x__IOWatchdogGetStatusAndValue (xsd__unsignedLong ulOption, struct MSXE170x__IOWatchdogGetStatusAndValueResponse *Response)`

Get watchdog current status and value information.

2.18.1 Function Documentation

- 2.18.1.1** `int MSXE170x__IOWatchdogInitAndStart (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct MSXE170x__Response * Response)`

Parameters

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

[in] **ulTimeValue** : Time base of the watchdog delay (0 to 0xFFFF)

[in] **ulOption1** : Reserved

[in] **ulOption2** : Reserved

[out] **Response** :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -2: time base selection error
- -3: time value selection error
- -100: Init and start digital output watchdog kernel function error

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

Returns

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

- 2.18.1.2** `int MSXE170x__IOWatchdogStopAndRelease (xsd__unsignedLong ulOption, struct MSXE170x__Response * Response)`

Parameters

[in] **ulOption** : reserved

[out] **Response** :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -100: Stop and release digital output watchdog kernel function error

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

Returns

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

2.18.1.3 int MSXE170x__IOWatchdogGetStatusAndValue (xsd__unsignedLong ulOption, struct MSXE170x__IOWatchdogGetStatusAndValueResponse * Response)

Parameters

[in] *ulOption* : Reserved

[out] *Response* :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -2: channel selection error
- -100: Get diagnostic information kernel function error

ulStatus : current status information

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

ulValue : current value information (0 to 0xFFFF)

ulInfo : reserved

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

Returns

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

2.19 MSX-E170x multifunction common functions

Functions

- int [MSXE170x__MFCommonSetInputsFilter](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulInputAFilterValue, xsd__unsignedLong ulInputBFilterValue, xsd__unsignedLong ulInputCFilterValue, xsd__unsignedLong ulInputDFilterValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct [MSXE170x__Response](#) *Response)

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

- int [MSXE170x__MFCommonReferenceVoltageActivation](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulActivationFlag, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE170x__Response](#) *Response)

Permit to activate the reference voltage to pin D-.

2.19.1 Function Documentation

2.19.1.1 `int MSXE170x__MFCommonSetInputsFilter (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulInputAFilterValue, xsd__unsignedLong ulInputBFilterValue, xsd__unsignedLong ulInputCFilterValue, xsd__unsignedLong ulInputDFilterValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE170x__Response * Response)`

Parameters

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

[in] *ulInputAFilterValue* : Filter value for input A (0 to 262143)

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

[in] *ulInputBFilterValue* : Filter value for input B (0 to 262143)

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

[in] *ulInputCFilterValue* : Filter value for input C (0 to 262143)

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

[in] *ulInputDFilterValue* : Filter value for input D (0 to 262143)

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

[in] *ulOption01* : Set it to 0

[in] *ulOption02* : Set it to 0

[in] *ulOption03* : Set it to 0

[in] *ulOption04* : Set it to 0

[out] *Response* :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Input A filter value selection error

- -4: Input B filter value selection error
- -5: Input C filter value selection error
- -6: Input D filter value selection error

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

Returns

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

2.19.1.2 `int MSXE170x__MFCommonReferenceVoltageActivation (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulActivationFlag, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response * Response)`

Parameters

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

[in] **ulActivationFlag** :

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

[in] **ulOption01** : Set it to 0

[in] **ulOption02** : Set it to 0

[out] **Response** :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Activation flag selection error

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

Returns

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

2.20 MSX-E170x incremental counter functions

Data Structures

- struct [MSXE170x__MFIncCounterGetNumberResponse](#)

Returns the number of counter available on the module.

Functions

- `int MSXE170x__MFIncCounterGetNumber (void ___, struct MSXE170x__MFIncCounterGetNumberResponse *Response)`
- `int MSXE170x__MFIncCounterInit (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterMode, xsd__unsignedLong ulCounterOption, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE170x__Response *Response)`
Initialise the counter.
- `int MSXE170x__MFIncCounterRelease (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response *Response)`
Release the counter.
- `int MSXE170x__MFIncCounterSetFIFO0Level (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulFIFOLevel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response *Response)`
Use the function MSXE17xx__MFCommonSetFIFO0Level.
- `int MSXE170x__MFIncCounterRead32BitValue (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__unsignedLongTimeStampResponse *Response)`
Read the 32 bits counter value.
- `int MSXE170x__MFIncCounterWrite32BitValue (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterValue, struct MSXE170x__Response *Response)`
write a 32 bits counter value
- `int MSXE170x__MFIncCounterClear (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response *Response)`
Clear the 32 bits counter.
- `int MSXE170x__MFIncCounterInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response *Response)`
Init and enable a counter compare value.
- `int MSXE170x__MFIncCounterDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response *Response)`
Disable and Release a counter compare value.
- `int MSXE170x__MFIncCounterInitAndEnableLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulHardwareTriggerEdgeSelection, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulOption01, struct MSXE170x__Response *Response)`
Init and enable a counter latch register.
- `int MSXE170x__MFIncCounterDisableAndReleaseLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulOption01, struct MSXE170x__Response *Response)`
Disable and Release a counter latch register.

- int [MSXE170x__MFIncCounterInitAndEnableFrequencyMeasurement](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulTimingInterval, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE170x__Response](#) *Response)
Init and enable a counter frequency measurement.
- int [MSXE170x__MFIncCounterDisableAndReleaseFrequencyMeasurement](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE170x__Response](#) *Response)
Disable and Release a counter frequency measurement.
- int [MSXE170x__MFIncCounterInitAndEnableIndex](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulReferenceAction, xsd__unsignedLong ulIndexOperation, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, struct [MSXE170x__Response](#) *Response)
Init and enable a counter index.
- int [MSXE170x__MFIncCounterDisableAndReleaseIndex](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE170x__Response](#) *Response)
Disable and Release a counter index.

2.20.1 Function Documentation

2.20.1.1 int [MSXE170x__MFIncCounterGetNumber](#) (void * __, struct [MSXE170x__MFIncCounterGetNumberResponse](#) * *Response*)

2.20.1.2 int [MSXE170x__MFIncCounterInit](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterMode, xsd__unsignedLong ulCounterOption, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct [MSXE170x__Response](#) * *Response*)

Parameters

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

[in] **ulCounterMode** : Set the counter mode : either

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

[in] **ulCounterOption** : Set the counter option

if in QUADRUPLE/DOUBLE/SIMPLE mode : either

- [MSXE170X_COUNTER_HYSTERESIS_ON](#) (0x1)
- [MSXE170X_COUNTER_HYSTERESIS_OFF](#) (0x0)

if in DIRECT mode :

- [MSXE170X_COUNTER_INCREMENT](#) (0x0)
- [MSXE170X_COUNTER_DECREMENT](#) (0x1)

[in] **ulOption01** : Set it to 0

[in] **ulOption02** : Set it to 0

[in] *ulOption03* : Set it to 0

[in] *ulOption04* : Set it to 0

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Counter mode selection error
- -4: Counter option selection error
- -5: Multifunction sub module is not a incremental counter
- -100: Init counter kernel function error

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

Returns

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

2.20.1.3 int MSXE170x__MFIncCounterRelease (xsd__unsignedLong *ulMFModuleIndex*, struct MSXE170x__Response * *Response*)

Parameters

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

[out] *Response* :

iReturnValue :

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

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

Returns

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

2.20.1.4 int MSXE170x__MFIncCounterSetFIFO0Level (xsd__unsignedLong *ulMFModuleIndex*, xsd__unsignedLong *ulFIFOLevel*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, struct MSXE170x__Response * *Response*)

2.20.1.5 int MSXE170x__MFIncCounterRead32BitValue (xsd__unsignedLong *ulMFModuleIndex*, struct MSXE170x__unsignedLongTimeStampResponse * *Response*)

Parameters

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

[out] **Response** :

iReturnValue :

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

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

ulTimeStampLow : 32 bit low part of time stamp (us) **ulTimeStampHigh** : 32 bit high part of time stamp (s)

Returns

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

2.20.1.6 `int MSXE170x__MFIncCounterWrite32BitValue (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterValue, struct MSXE170x__Response * Response)`

Parameters

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

[in] **ulCounterValue** : Counter value

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Counter value error
- -4: Multifunction sub module is not a incremental counter
- -5: Incremental counter not initialised
- -100: Write counter value kernel function error

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

Returns

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

2.20.1.7 `int MSXE170x__MFIncCounterClear (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response * Response)`

Parameters

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

[out] **Response** :

iReturnValue :

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

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

Returns

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

2.20.1.8 `int MSXE170x__MFIncCounterInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response * Response)`

Parameters

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

[in] **ulValue** : compare value (0 to 0xFFFFFFFF included)

[in] **ulMode** : compare mode

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

[in] **ulSynchroTrigger** • 0 : no synchro trigger
 • 1 : generates a synchro trigger when condition is true

[in] **ulOption01** : set it to 0

[in] **ulOption02** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

2.20.1.9 int MSXE170x__MFIncCounterDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response * Response)

Parameters

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -5: Compare logic not initialised
- -100: Disable counter compare value kernel function error

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

Returns

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

2.20.1.10 int MSXE170x__MFIncCounterInitAndEnableLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulHardwareTriggerEdgeSelection, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulOption01, struct MSXE170x__Response * Response)

Parameters

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

[in] **ulLatchRegister** : 0: First Latch Register used with trigger input, 1: Second Lath Register used with synchro input

[in] **ulHardwareTriggerEdgeSelection** : not used for the synchro input

- 01 : rising front (Only if trigger input selected)
- 10 : falling front (Only if trigger input selected)
- 11 : Both front (Only if trigger input selected)

[in] **ulHardwareTriggerCount** : not used for the synchro input

Define the number of trigger events before the action occur

1 : all trigger event start the action

max value : 65535

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Latch register selection error
- -4: Trigger edge selection error

- -5: Trigger count selection error
- -6: Multifunction sub module is not a incremental counter
- -7: Incremental counter not initialised
- -8: Latch logic already initialised
- -100: Init and enable counter latch register kernel function error

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

Returns

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

2.20.1.11 `int MSXE170x__MFIncCounterDisableAndReleaseLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulOption01, struct MSXE170x__Response * Response)`

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
- [in] **ulLatchRegister** : 0: First Lath Register used with 24 V input, 1: Second Lath Register used with synchro input
- [in] **ulOption01** : set it to 0
- [out] **Response** :
- iReturnValue** :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Latch register selection error
 - -4: Multifunction sub module is not a incremental counter
 - -5: Incremental counter not initialised
 - -6: Latch logic not initialised
 - -100: Disable and release counter latchd register kernel function error
- syserrno** : system-error code (the value of the libc "errno" code)

Returns

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

2.20.1.12 `int MSXE170x__MFIncCounterInitAndEnableFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulTimingInterval, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response * Response)`

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
- [in] **ulTimingInterval** : Timing interval from 1 to 0xFFFF, one step correspond to 100 ns.

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

[in] **ulOption01** : Set the measurement mode

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

[in] **ulOption02** : Timebase selection

- 0: 100ns
- 1: 100us

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Timing interval selection error
- -4: Multifunction sub module is not a incremental counter
- -5: Incremental counter not initialised
- -6: Frequency measurement already initialised and started
- -7: Frequency measurement ulOption01 parameter wrong value
- -8: Frequency measurement ulOption02 parameter wrong value
- -100: Init and enable counter Frequency measurement kernel function error

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

Returns

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

2.20.1.13 int MSXE170x__MFIncCounterDisableAndReleaseFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response * Response)

Parameters

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

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -5: Frequency measurement not initialised
- -100: Disable and release counter latchd register kernel function error

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

Returns

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

2.20.1.14 `int MSXE170x__MFIncCounterInitAndEnableIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulReferenceAction, xsd__unsignedLong ulIndexOperation, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, struct MSXE170x__Response * Response)`

The index operation can be programmed. If used, the reference signal enable the index management. Rising edge on reference enable the index management. Rising edge on index reset the reference management.

Parameters

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

[in] *ulReferenceAction* : Reference action

- 0: do not use the D pin as reference,
- 1: use the D pin as reference

[in] *ulIndexOperation* : Index operation

- 0 = MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_COUNTER,
- 1 = MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_COUNTER,
- 2 = MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER,
- 3 = MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER,

[in] *ulAutoMode* : AutoMode 0: do not use auto mode (action is done only once), 1: use auto mode (action is done continuously)

[in] *ulOption01* : set it to 0

[out] *Response* :

iReturnValue :

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

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

Returns

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

2.20.1.15 `int MSXE170x__MFIncCounterDisableAndReleaseIndex (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response * Response)`

Parameters

- [in] *ulMFModuleIndex* : index of the multifunction sub module (0 to 3).
- [in] *ulOption01* : set it to 0
- [out] *Response* :
- iReturnValue* :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter
 - -4: Incremental counter not initialised
 - -5: Index logic not initialised
 - -100: Disable and release counter index kernel function error
- syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

2.21 MSX-E17xx digital I/O functions

Data Structures

- struct [MSXE17xx__DigitalIOGetNumberResponse](#)

Functions

- `int MSXE17xx__DigitalIOGetNumber (void ___, struct MSXE17xx__DigitalIOGetNumberResponse *Response)`
Returns the number of digital IO available on the module.
- `int MSXE17xx__DigitalIOInitPortConfiguration (xsd__unsignedLong ulPort, xsd__unsignedLong ulPortConfiguration, struct MSXE17xx__Response *Response)`
Initialise a digital i/o port (2 channels).
- `int MSXE17xx__DigitalIOReadChannelValue (xsd__unsignedLong ulChannel, struct MSXE17xx__unsignedLongResponse *Response)`
Read a digital i/o channel value.
- `int MSXE17xx__DigitalIOReadAllChannelsValue (void ___, struct MSXE17xx__unsignedLongResponse *Response)`
Read all digital i/o channels value.If channel is configured as output, then this function return the status of the output.

- `int MSXE17xx__DigitalIOWriteChannelValue (xsd__unsignedLong ulChannel, xsd__unsignedLong ulChannelValue, struct MSXE17xx__Response *Response)`
write a digital i/o channel value
- `int MSXE17xx__DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE17xx__Response *Response)`
write all digital i/o channels value
- `int MSXE17xx__DigitalIOReleasePortConfiguration (xsd__unsignedLong ulPort, struct MSXE17xx__Response *Response)`
Release a digital i/o port (2 channels).
- `int MSXE17xx__DigitalIOTestShortCircuit (xsd__unsignedLong ulOption, struct MSXE17xx__Response *Response)`
Test short circuit status.
- `int MSXE17xx__DigitalIORearmShortCircuit (xsd__unsignedLong ulOption, struct MSXE17xx__Response *Response)`
Rearm digital outputs after a short circuit happened.

2.21.1 Function Documentation

2.21.1.1 `int MSXE17xx__DigitalIOGetNumber (void * __, struct MSXE17xx__DigitalIOGetNumberResponse * Response)`

Parameters

[in] *None*

[out] *Response :*

sResponse.iReturn Value :

- 0: means the remote function performed OK
- -1: means an system error occured (check errno in this case)

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

Returns

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

2.21.1.2 `int MSXE17xx__DigitalIOInitPortConfiguration (xsd__unsignedLong ulPort, xsd__unsignedLong ulPortConfiguration, struct MSXE17xx__Response * Response)`

Parameters

[in] *ulPort* : Index of the digital i/o port (0 to 7)

[in] *ulPortConfiguration* : Define the port configuration

- 0 : input
- 1 : output

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o port selection error
- -3: Port configuration selection error
- -100: Init dig i/o port kernel function error

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

Returns

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

2.21.1.3 int MSXE17xx_DigitalIOReadChannelValue (xsd_unsignedLong ulChannel, struct MSXE17xx_unsignedLongResponse * Response)

Parameters

[in] **ulChannel** : Index of the digital i/o channel (0 to 15)

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o channel selection error
- -100: Read dig i/o channel value kernel function error

syserrno : system-error code (the value of the libc "errno" code) **ulValue** : i/o channel value:

- 0
- 1

Returns

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

2.21.1.4 int MSXE17xx_DigitalIOReadAllChannelsValue (void * _, struct MSXE17xx_unsignedLongResponse * Response)

Parameters

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -100: Read dig i/o channel value kernel function error

syserrno : system-error code (the value of the libc "errno" code) **ulValue** : i/o channels value(each bit correspond to one channel)

Returns

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

2.21.1.5 `int MSXE17xx__DigitalIOWriteChannelValue (xsd__unsignedLong ulChannel, xsd__unsignedLong ulChannelValue, struct MSXE17xx__Response * Response)`

Parameters

- [in] *ulChannel* : Index of the digital i/o channel (0 to 15)
- [in] *ulChannelValue* : Channel value
- 0
 - 1
- [out] *Response* :
- iReturnValue* :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Digital i/o channel selection error
 - -3: Channel value error
 - -100: Write dig i/o channel value kernel function error
- syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

2.21.1.6 `int MSXE17xx__DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE17xx__Response * Response)`

Parameters

- [in] *ulChannelValue* : Channels value (each bit corresponds to a channel)
- [out] *Response* :
- iReturnValue* :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Channels value error
 - -100: Write dig i/o channel value kernel function error
- syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

2.21.1.7 int MSXE17xx__DigitalIOReleasePortConfiguration (xsd__unsignedLong *ulPort*, struct MSXE17xx__Response * *Response*)

Parameters

- [in] *ulPort* : Index of the digital i/o port (0 to 7)
- [out] *Response* :
- iReturnValue* :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Digital i/o port selection error
- syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

2.21.1.8 int MSXE17xx__DigitalIOTestShortCircuit (xsd__unsignedLong *ulOption*, struct MSXE17xx__unsignedLongResponse * *Response*)

Parameters

- [in] *ulOption* : reserved
- [out] *Response* :
- iReturnValue* :
- 0 : means the remote function performed OK
 - -1: means an system error occurred
- syserrno* : system-error code (the value of the libc "errno" code)
- ulValue* : short circuit status: from 0 to 0xffff, one bit for each output
- 0 : no short circuit
 - 1 : short circuit

Returns

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

2.21.1.9 int MSXE17xx__DigitalIORearmShortCircuit (xsd__unsignedLong *ulOption*, struct MSXE17xx__Response * *Response*)

Parameters

- [in] *ulOption* : reserved
- [out] *Response* :
- iReturnValue* :
- 0 : means the remote function performed OK
 - -1: means an system error occurred

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

Returns

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

2.22 MSX-E17xx IO watchdog functions

Data Structures

- struct [MSXE17xx_IOWatchdogGetStatusAndValueResponse](#)

Functions

- int [MSXE17xx_IOWatchdogInitAndStart](#) (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct [MSXE17xx__Response](#) *Response)
Init and start the digital output IO watchdog.
- int [MSXE17xx_IOWatchdogStopAndRelease](#) (xsd__unsignedLong ulOption, struct [MSXE17xx__Response](#) *Response)
Stop and release the digital output watchdog.
- int [MSXE17xx_IOWatchdogGetStatusAndValue](#) (xsd__unsignedLong ulOption, struct [MSXE17xx_IOWatchdogGetStatusAndValueResponse](#) *Response)
Get watchdog current status and value information.

2.22.1 Function Documentation

2.22.1.1 int [MSXE17xx_IOWatchdogInitAndStart](#) (xsd__unsignedLong *ulTimeBase*, xsd__unsignedLong *ulTimeValue*, xsd__unsignedLong *ulOption1*, xsd__unsignedLong *ulOption2*, struct [MSXE17xx__Response](#) * *Response*)

Parameters

- [in] *ulTimeBase* : Time base of the watchdog delay (0 for mus, 1 for ms, 2 for s)
- [in] *ulTimeValue* : Time base of the watchdog delay (0 to 0xFFFF)
- [in] *ulOption1* : Reserved
- [in] *ulOption2* : Reserved
- [out] *Response* :
- iReturnValue* :
- 0: remote function performed OK
 - -1: an system error occurred
 - -2: time base selection error
 - -3: time value selection error
 - -100: Init and start digital output watchdog kernel function error

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

Returns

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

2.22.1.2 `int MSXE17xx_IOWatchdogStopAndRelease (xsd__unsignedLong ulOption, struct MSXE17xx_Response * Response)`

Parameters

[in] *ulOption* : reserved

[out] *Response* :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -100: Stop and release digital output watchdog kernel function error

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

Returns

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

2.22.1.3 `int MSXE17xx_IOWatchdogGetStatusAndValue (xsd__unsignedLong ulOption, struct MSXE17xx_IOWatchdogGetStatusAndValueResponse * Response)`

Parameters

[in] *ulOption* : Reserved

[out] *Response* :

iReturnValue :

- 0: remote function performed OK
 - -1: an system error occurred
 - -2: channel selection error
 - -100: Get diagnostic information kernel function error
- ulStatus* : current status information
- BIN XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXX0: is stopped,
 - BIN XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXX1: is running,
 - BIN XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXX0X: is not run down
 - BIN XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXX1X: is run down

ulValue : current value information (0 to 0xFFFF)

ulInfo : reserved

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

Returns

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

2.23 MSX-E17xx multifunction common functions

Functions

- int [MSXE17xx__MFCommonGetSubModuleFunctionality](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE17xx__unsignedLongResponse](#) *Response)

Get the selected sub module functionality.

- int [MSXE17xx__MFCommonSetInputsFilter](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulInputAFilterValue, xsd__unsignedLong ulInputBFilterValue, xsd__unsignedLong ulInputCFilterValue, xsd__unsignedLong ulInputDFilterValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct [MSXE17xx__Response](#) *Response)

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

- int [MSXE17xx__MFCommonReferenceVoltageActivation](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulActivationFlag, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE17xx__Response](#) *Response)

Permit to activate the reference voltage to pin D-.

- int [MSXE17xx__MFCommonEnableDisableTriggerGate](#) (xsd__unsignedLong ulTriggerConfiguration, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE17xx__Response](#) *Response)

Enable / disable trigger gate.

- int [MSXE17xx__MFCommonSetFIFO0Level](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulFIFOLevel, xsd__unsignedLong ulTimeOutTimeBase, xsd__unsignedLong ulReloadValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE17xx__Response](#) *Response)

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

2.23.1 Function Documentation

2.23.1.1 int [MSXE17xx__MFCommonGetSubModuleFunctionality](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE17xx__unsignedLongResponse](#) * Response)

Parameters

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

[out] **Response** :

ulValue :

- 0: Incremental counter
- -1: PWM

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred (check errno in this case)
- -2: Multifunction sub module index selection error

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

Returns

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

2.23.1.2 `int MSXE17xx_MFCommonSetInputsFilter (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulInputAFilterValue, xsd__unsignedLong ulInputBFilterValue, xsd__unsignedLong ulInputCFilterValue, xsd__unsignedLong ulInputDFilterValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx_Response * Response)`

Parameters

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

[in] *ulInputAFilterValue* : Filter value for input A (0 to 262143)

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

[in] *ulInputBFilterValue* : Filter value for input B (0 to 262143)

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

[in] *ulInputCFilterValue* : Filter value for input C (0 to 262143)

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

[in] *ulInputDFilterValue* : Filter value for input D (0 to 262143)

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

[in] *ulOption01* : Set it to 0

[in] *ulOption02* : Set it to 0

[in] *ulOption03* : Set it to 0

[in] *ulOption04* : Set it to 0

[out] *Response* :

iReturnValue :

- 0 : means the remote function performed OK

- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Input A filter value selection error
- -4: Input B filter value selection error
- -5: Input C filter value selection error
- -6: Input D filter value selection error

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

Returns

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

2.23.1.3 `int MSXE17xx__MFCommonReferenceVoltageActivation (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulActivationFlag, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

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

[in] **ulActivationFlag** :

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

[in] **ulOption01** : Set it to 0

[in] **ulOption02** : Set it to 0

[out] **Response** :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Activation flag selection error

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

Returns

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

2.23.1.4 `int MSXE17xx__MFCommonEnableDisableTriggerGate (xsd__unsignedLong ulTriggerConfiguration, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

[in] **ulTriggerConfiguration** : Trigger gate configuration:

Bit 0, Hardware trigger gate :

- 0 : Hardware trigger gate is disabled

- 1 : Hardware trigger gate is enabled

[in] ***ulOption01*** : Set it to 0

[in] ***ulOption02*** : Set it to 0

[out] ***Response*** :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occurred
- -2: ulTriggerConfiguration parameter is wrong
- -100: MSXE17xx__MFCommonEnableDisableTriggerGate kernel function error

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

Returns

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

2.23.1.5 `int MSXE17xx__MFCommonSetFIFO0Level (xsd_unsignedLong ulMFModuleIndex, xsd_unsignedLong ulFIFOLevel, xsd_unsignedLong ulTimeOutTimeBase, xsd_unsignedLong ulReloadValue, xsd_unsignedLong ulOption01, xsd_unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

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

[in] ***ulFIFOLevel*** : Define the FIFO level (1 to 200).

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

[in] ***ulReloadValue*** : Time out reload value (1 to 0xFFFF)

[in] ***ulOption01*** : reserved (Set it to 0).

[in] ***ulOption02*** : reserved (Set it to 0).

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: FIFO level value is wrong
- -4: Time out time base selection error
- -5: Time out value can not be null, if a time base is selected
- -100: Set FIFO level kernel function error

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

Returns

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

2.24 MSX-E17xx pulse width modulation

Data Structures

- struct [MSXE17xx__MFPWMGetConfigurationResponse](#)
- struct [MSXE17xx__MFPWMGetStateResponse](#)

Functions

- int [MSXE17xx__MFPWMGetNumber](#) (void *__, struct [MSXE17xx__unsignedLongResponse](#) *Response)

Returns the number of PWM sub modules available on the module.

- int [MSXE17xx__MFPWMInitAndEnable](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, xsd__unsignedLong ulStartLevel, xsd__unsignedLong ulExternGate, xsd__unsignedLong ulStopMode, xsd__unsignedLong ulStopLevel, xsd__unsignedLong ulSynchroOut, xsd__unsignedLong ulDataFrame, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct [MSXE17xx__Response](#) *Response)

Initialise and enable the selected PWM module.

- int [MSXE17xx__MFPWMSetNewTiming](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE17xx__Response](#) *Response)

Set a new PWM timing.

- int [MSXE17xx__MFPWMDisable](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE17xx__Response](#) *Response)

Disable the selected PWM.

- int [MSXE17xx__MFPWMDisableAndRelease](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE17xx__Response](#) *Response)

Disable and release the selected PWM.

- int [MSXE17xx__MFPWMGetConfiguration](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, struct [MSXE17xx__MFPWMGetConfigurationResponse](#) *Response)

Get the selected PWM initialisation.

- int [MSXE17xx__MFPWMGetState](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, struct [MSXE17xx__MFPWMGetStateResponse](#) *Response)

Get the selected PWM state.

2.24.1 Function Documentation

2.24.1.1 `int MSXE17xx__MFPWMGetNumber (void * __, struct MSXE17xx__unsignedLongResponse * Response)`

Parameters

[in] *None*

[out] *Response* :

ulValue : number of available PWM sub modules

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred (check errno in this case)

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

Returns

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

2.24.1.2 `int MSXE17xx__MFPWMInitAndEnable (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, xsd__unsignedLong ulStartLevel, xsd__unsignedLong ulExternGate, xsd__unsignedLong ulStopMode, xsd__unsignedLong ulStopLevel, xsd__unsignedLong ulSynchroOut, xsd__unsignedLong ulDataFrame, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulPWM* : Selected PWM 0 or 1

[in] *ulTimeBase* : PWM time base selection

- 0: 250ns (4MHz)
- 1: micro s (1MHz)
- 2: ms (1kHz)
- 3: s (Hz)
- 4: Diff input C for PWM0 and D for PWM1
- 5: Trigger input (24V)
- 6: Synchro trigger

[in] *ulLowTime* : PWM low time reload value (1 to 16777215)

- If the extern clock is used this define the number of clocks for the low level.
- If the internal time base is used this define the low level time : Low time = *ulTimeBase* * *ulLowTime*.

[in] *ulHighTime* : PWM high time reload value (1 to 16777215)

- If the extern clock is used this define the number of clocks for the high level.
- If the internal time base is used this define the high level time : High time = ulTimeBase * ulHighTime.

[in] **ulStartLevel** : PWM start level, time selection

- 0: PWM start with the low level
- 1: PWM start with the high level

[in] **ulExternGate** : Extern gate selection

- 0: Extern gate not used
- 1: Diff input C (PWM0) or D (PWM1) is used as gate
- 2: Input trigger (24V) is used as gate

[in] **ulStopMode** : Stop mode selection

- 0: Stop the PWM signal directly after a stop condition
- 1: Stop the PWM after the current period is stopped

[in] **ulStopLevel** : Stop level selection

- 0: The output signal keep the level after the stop condition
- 1: The output signal is set to low after the stop condition
- 2: The output signal is set to high after the stop condition

[in] **ulSynchroOut** : Give the possibility to generate a synchro trigger

- 0: Disabled
- 1: Generate a synchro trigger be each start of period
- 2: Generate a synchro trigger be each end of period
- 3: Generate a synchro trigger be each start and end of period

[in] **ulDataFrame** : Give the possibility to send via the data server PWM informations

- 0: Disabled
- 1: Enabled

For each start and end of period the data server send a 5 DWORD frame with following informations:

```

DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (1)
DWORD 3 : Event mask
    8: PWM0. First period begin
    9: PWM0. End of period
    10: PWM0. End of last period. Stop condition occur
    11: PWM1. First period begin
    12: PWM1. End of period
    13: PWM1. End of last period. Stop condition occur
DWORD 4 :
    D23-D0: Number of occurred periods
    D24: 0: Diff input C for PWM0 or D for PWM1 is low
         1: Diff input C for PWM0 or D for PWM1 is high
    D25: 0: Extern trigger input (24V) is low
         1: Extern trigger input (24V) is high
    D26: 0: PWM output is low
         1: PWM output is high
  
```

[in] **ulOption01** : Reserved.

- Set to 0

[in] **ulOption02** : Reserved.

- Set to 0

[in] *ulOption03* : Reserved.

- Set to 0

[in] *ulOption04* : Reserved.

- Set to 0

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -5: Time base selection wrong
- -6: Low time selection wrong
- -7: High time selection wrong
- -8: PWM already in progress
- -9: PWM timing not initialised
- -10: Start level selection wrong
- -11: Extern gate selection wrong
- -12: Extern gate selection identical to the external clock selection
- -13: Stop mode selection wrong
- -14: Stop level selection wrong
- -15: Synchro out trigger selection wrong
- -16: Module is slave and can not generate the syncho out trigger
- -17: Data frame selection wrong
- -18: Can not mixed synchro trigger clock and module synchronisation (ulSynchroOut)
- -100: Init and enable PWM kernel function error

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

Returns

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

2.24.1.3 `int MSXE17xx__MFPWMSetNewTiming (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulPWM* : Selected PWM 0 or 1

[in] *ulTimeBase* : PWM time base selection

- 0: 250ns (4MHz)
- 1: micro s (1MHz)
- 2: ms (1kHz)

- 3: s (Hz)
- 4: Diff input C for PWM0 and D for PWM1
- 5: Trigger input (24V)
- 6: Synchro trigger

[in] **ulLowTime** : PWM low time reload value (1 to 16777215)

If the extern clock is used this define the number of clocks for the low level.

If the internal time base is used this define the low level time : Low time = ulTimeBase * ulLowTime.

[in] **ulHighTime** : PWM high time reload value (1 to 16777215)

If the extern clock is used this define the number of clocks for the high level.

If the internal time base is used this define the high level time : High time = ulTimeBase * ulHighTime.

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

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -5: Time base selection wrong
- -6: Low time selection wrong
- -7: High time selection wrong
- -8: Can not change the current time base
- -9: PWM not initialised
- -10: PWM not enabled
- -100: Set new PWM timing kernel function error

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

Returns

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

2.24.1.4 `int MSXE17xx__MFPWMDisable (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

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

[in] **ulPWM** : Selected PWM 0 or 1

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

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -5: PWM not initialised
- -6: PWM already disabled
- -100: Disable PWM kernel function error

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

Returns

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

2.24.1.5 `int MSXE17xx_MFPWMDisableAndRelease (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx_Response * Response)`

If PWM in progress this stop directly the PWM and ignore the ulStopMode configuration (MSXE17xx_MFPWMInitAndEnable).

Parameters

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

[in] **ulPWM** : Selected PWM 0 or 1

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

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -5: PWM not initialised
- -100: Disable PWM kernel function error

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

Returns

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

2.24.1.6 `int MSXE17xx_MFPWMGetConfiguration (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, struct MSXE17xx_MFPWMGetConfigurationResponse * Response)`

Parameters

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

[in] ***ulPWM*** : Selected PWM 0 or 1

[out] ***Response*** :

ulInitialised Return the initialisation flag

- 0: PWN not initialised
- 1: PWM initialised

ulTimeBase PWM time base selection

- 0: 250ns (4MHz)
- 1: micro s (1MHz)
- 2: ms (1kHz)
- 3: s (Hz)
- 4: Diff input C for PWM0 and D for PWM1
- 5: Trigger input (24V)
- 6: Synchro trigger

ulLowTime PWM low time reload value (1 to 16777215)

- If the extern clock is used this define the number of clocks for the low level.
- If the internal time base is used this define the low level time : Low time = $ulTimeBase * ulLowTime$.

ulHighTime PWM high time reload value (1 to 16777215)

- If the extern clock is used this define the number of clocks for the high level.
- If the internal time base is used this define the high level time : High time = $ulTimeBase * ulHighTime$.

ulEnabled Return the enabled status

- 0: PWM disabled
- 1: PWN enabled

ulStartLevel Return the start level, time

- 0: PWM start with the low level
- 1: PWM start with the high level

ulExternGate Return the extern gate initialisation

- 0: Extern gate not used
- 1: Diff input C (PWM0) or D (PWM1) is used as gate
- 2: Input trigger (24V) is used as gate

ulStopMode Return the current stop mode

- 0: Stop the PWM signal directly after a stop condition
- 1: Stop the PWM after the current period is stopped

ulStopLevel Return the current selected stop level

- 0: The output signal keep the level after the stop condition
- 1: The output signal is set to low after the stop condition
- 2: The output signal is set to high after the stop condition

ulSynchroOut Give the possibility to generate a synchro trigger

- 0: Disabled
- 1: Generate a synchro trigger be each start of period
- 2: Generate a synchro trigger be each end of period
- 3: Generate a synchro trigger be each start and end of period

ulDataFrame Give the possibility to send via the data server PWM informations

- 0: Disabled
- 1: Enabled

For each start and end of period the data server send a 5 DWORD frame with following informations:

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (1)
DWORD 3 : Event mask
    8: PWM0. First period begin
    9: PWM0. End of period
    10: PWM0. End of last period. Stop condition occur
    11: PWM1. First period begin
    12: PWM1. End of period
    13: PWM1. End of last period. Stop condition occur
DWORD 4 :
    D23-D0: Number of occurred periods
    D24: 0: Diff input C for PWM0 or D for PWM1 is low
         1: Diff input C for PWM0 or D for PWM1 is high
    D25: 0: Extern trigger input (24V) is low
         1: Extern trigger input (24V) is high
    D26: 0: PWM output is low
         1: PWM output is high
```

ulInfo01 :

- Reserved. Return 0

ulInfo02 :

- Reserved. Return 0

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -100: Get PWM configuration kernel function error

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

Returns

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

2.24.1.7 `int MSXE17xx__MFPWMGetState (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, struct MSXE17xx__MFPWMGetStateResponse * Response)`

Parameters

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

[in] **ulPWM** : Selected PWM 0 or 1

[out] **Response** :

ulInitialised : Return the initialisation flag

- 0: PWN not initialised
- 1: PWM initialised

ulEnabled : Return the enabled status

- 0: PWM disabled
- 1: PWN enabled

ulProgress : Return the progress status

- 0: PWM not in progress
- 1: PWM in progress

ulOutput : Return the PWM output signal state

- 0: PWM output is low
- 1: PWM output is high

ulDiffInput : Return the diff input state

- 0: The diff input C (for PWM0) or F (for PWM1) is low
- 1: The diff input C (for PWM0) or F (for PWM1) is high

ulTriggerInput : Return the extern trigger state

- 0: The trigger input (24V) is low
- 1: The trigger input (24V) is high

ulCounter :

- Return the number of completed PWM periods

ulInfo01 :

- Reserved. Return 0

ulInfo02 :

- Reserved. Return 0

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -100: Get PWM state kernel function error

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

Returns

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

2.25 MSX-E17xx incremental counter functions

Data Structures

- struct [MSXE17xx__MFincrementCounterGetNumberResponse](#)

Functions

- `int MSXE17xx__MFIncCounterGetNumber (void ___, struct MSXE17xx__MFIncCounterGetNumberResponse *Response)`
Returns the number of counter available on the module.
- `int MSXE17xx__MFIncCounterInit (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterMode, xsd__unsignedLong ulCounterOption, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__Response *Response)`
Initialise the counter.
- `int MSXE17xx__MFIncCounterRelease (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)`
Release the counter.
- `int MSXE17xx__MFIncCounterRead32BitValue (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__unsignedLongTimeStampResponse *Response)`
Read the 32 bits counter value.
- `int MSXE17xx__MFIncCounterWrite32BitValue (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterValue, struct MSXE17xx__Response *Response)`
write a 32 bits counter value
- `int MSXE17xx__MFIncCounterClear (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)`
Clear the 32 bits counter.
- `int MSXE17xx__MFIncCounterInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`
Init and enable a counter compare value.
- `int MSXE17xx__MFIncCounterDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)`
Disable and Release a counter compare value.
- `int MSXE17xx__MFIncCounterInitHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulCount, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`
Init the hardware trigger configuration.
- `int MSXE17xx__MFIncCounterReleaseHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`
Release the hardware trigger.
- `int MSXE17xx__MFIncCounterInitIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`
Init the index configuration.

- `int MSXE17xx__MFIncCounterReleaseIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`
Release the index.
- `int MSXE17xx__MFIncCounterInitAndEnableLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`
Init and enable a counter latch logic.
- `int MSXE17xx__MFIncCounterDisableAndReleaseLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`
Disable and Release a counter latch logic.
- `int MSXE17xx__MFIncCounterInitAndEnableClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`
Init and enable a counter clear logic.
- `int MSXE17xx__MFIncCounterDisableAndReleaseClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`
Disable and Release a counter clear logic.
- `int MSXE17xx__MFIncCounterInitAndEnableLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulHardwareTriggerEdgeSelection, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`
Init and enable a counter latch register.
- `int MSXE17xx__MFIncCounterDisableAndReleaseLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`
Disable and Release a counter latch register.
- `int MSXE17xx__MFIncCounterInitAndEnableFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulTimingInterval, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`
Init and enable a counter frequency measurement.
- `int MSXE17xx__MFIncCounterDisableAndReleaseFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)`
Disable and Release a counter frequency measurement.
- `int MSXE17xx__MFIncCounterInitAndEnableIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulReferenceAction, xsd__unsignedLong ulIndexOperation, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`

Init and enable a counter index.

- int [MSXE17xx__MFIncCounterDisableAndReleaseIndex](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE17xx__Response](#) *Response)

Disable and Release a counter index.

2.25.1 Function Documentation

2.25.1.1 int [MSXE17xx__MFIncCounterGetNumber](#) (void * __, struct [MSXE17xx__MFIncCounterGetNumberResponse](#) * *Response*)

Parameters

[in] *None*

[out] *Response* :

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured (check errno in this case)

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

Returns

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

2.25.1.2 int [MSXE17xx__MFIncCounterInit](#) (xsd__unsignedLong *ulMFModuleIndex*, xsd__unsignedLong *ulCounterMode*, xsd__unsignedLong *ulCounterOption*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, xsd__unsignedLong *ulOption03*, xsd__unsignedLong *ulOption04*, struct [MSXE17xx__Response](#) * *Response*)

Parameters

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

[in] *ulCounterMode* : Set the counter mode : either

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

[in] *ulCounterOption* : Set the counter option

if in QUADRUPLE/DOUBLE/SIMPLE mode : either

- [MSXE170X_COUNTER_HYSTERESIS_ON](#) (0x1)
- [MSXE170X_COUNTER_HYSTERESIS_OFF](#) (0x0)

if in DIRECT mode :

- [MSXE170X_COUNTER_INCREMENT](#) (0x0)
- [MSXE170X_COUNTER_DECREMENT](#) (0x1)

[in] *ulOption01* : Set it to 0

[in] *ulOption02* : Set it to 0
 [in] *ulOption03* : Set it to 0
 [in] *ulOption04* : Set it to 0
 [out] *Response* :
 iReturnValue :
 • 0: means the remote function performed OK
 • -1: means an system error occurred
 • -2: Multifunction sub module index selection error
 • -3: Counter mode selection error
 • -4: Counter option selection error
 • -5: Multifunction sub module is not a incremental counter
 • -100: Init counter kernel function error
 syserrno : system-error code (the value of the libc "errno" code)

Returns

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

2.25.1.3 `int MSXE17xx__MFIncCounterRelease (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response * Response)`

Parameters

[in] *ulMFModuleIndex* : index of the multifunction sub module (0 to 3).
 [out] *Response* :
 iReturnValue :
 • 0: means the remote function performed OK
 • -1: means an system error occurred
 • -2: Multifunction sub module index selection error
 • -3: Multifunction sub module is not a incremental counter
 • -4: Incremental counter not initialised
 • -100: Release incremental counter kernel function error
 syserrno : system-error code (the value of the libc "errno" code)

Returns

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

2.25.1.4 `int MSXE17xx__MFIncCounterRead32BitValue (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__unsignedLongTimeStampResponse * Response)`

Parameters

[in] *ulMFModuleIndex* : index of the multifunction sub module (0 to 3).
 [out] *Response* :
 iReturnValue :

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

syserrno : system-error code (the value of the libc "errno" code) *ulValue* : counter value
ulTimeStampLow : 32 bit low part of time stamp (us) *ulTimeStampHigh* : 32 bit high part of time stamp (s)

Returns

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

2.25.1.5 `int MSXE17xx__MFIncCounterWrite32BitValue (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterValue, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulCounterValue* : Counter value

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Counter value error
- -4: Multifunction sub module is not a incremental counter
- -5: Incremental counter not initialised
- -100: Write counter value kernel function error

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

Returns

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

2.25.1.6 `int MSXE17xx__MFIncCounterClear (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response * Response)`

Parameters

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

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK

- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -100: Write counter value kernel function error

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

Returns

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

2.25.1.7 int MSXE17xx_MFIncCounterInitAndEnableCompareLogic (xsd_unsignedLong ulMFModuleIndex, xsd_unsignedLong ulValue, xsd_unsignedLong ulMode, xsd_unsignedLong ulSynchroTrigger, xsd_unsignedLong ulOption01, xsd_unsignedLong ulOption02, struct MSXE17xx_Response * Response)

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

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (0)
DWORD 3 : Event mask
    D30-D0 :
        0: Compare
        1: Frequency measurement
        2: Hardware trigger latch occur
        3: Synchro input latch occur
        4: Index input latch occur
    D31 :
        0: No error occur
        1: Amplitude or Frequency error occur.
DWORD 4 :
    D31-D0: Counter value (DWORD)
```

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
- [in] **ulValue** : compare value (0 to 0xFFFFFFFF included)
- [in] **ulMode** : compare mode
- 0: condition true when counter equals compare value
 - 1: condition true when counter equals a multiple of the compare value
- [in] **ulSynchroTrigger** • 0 : no synchro trigger
- 1 : generates a synchro trigger when condition is true
- [in] **ulOption01** : set it to 0
- [in] **ulOption02** : set it to 0
- [out] **Response** :
- iReturnValue** :
- 0: means the remote function performed OK
 - -1: means an system error occurred

- -2: Multifunction sub module index selection error
- -3: Compare value error
- -4: Compare mode error
- -5: Synchro trigger error
- -6: Multifunction sub module is not a incremental counter
- -7: Incremental counter not initialised
- -8: Compare logic already initialised
- -100: Init and enable counter compare kernel function error

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

Returns

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

2.25.1.8 `int MSXE17xx__MFIncCounterDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response * Response)`

Parameters

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

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -5: Compare logic not initialised
- -100: Disable counter compare value kernel function error

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

Returns

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

2.25.1.9 `int MSXE17xx__MFIncCounterInitHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulCount, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulEdgeSelection* : Front selection

- 01 : rising front
- 10 : falling front

- 11 : Both front

[in] **ulCount** : Define the number of trigger events before the action occur
 1 : all trigger event start the action
 max value : 65535

[in] **ulOption01** : set it to 0

[in] **ulOption02** : set it to 0

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Trigger edge selection error
- -5: Trigger count selection error
- -6: Incremental counter not initialised
- -7: Hardware trigger already initialised
- -100: Init hardware trigger kernel function error

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

Returns

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

2.25.1.10 int MSXE17xx_MFIncCounterReleaseHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx_Response * Response)

Parameters

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

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -5: Hardware trigger not initialised
- -6: Hardware trigger used and can not released
- -100: Release hardware trigger kernel function error

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

Returns

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

2.25.1.11 `int MSXE17xx_MFIncCounterInitIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx_Response * Response)`

Parameters

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

[in] *ulEdgeSelection* : Front selection

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

[in] *ulOption01* : set it to 0

[in] *ulOption02* : set it to 0

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Index edge selection error
- -5: Incremental counter not initialised
- -6: Index already initialised
- -100: Init index kernel function error

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

Returns

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

2.25.1.12 `int MSXE17xx_MFIncCounterReleaseIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx_Response * Response)`

Parameters

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

[in] *ulOption01* : set it to 0

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -5: Index not initialised
- -6: Index used and can not released
- -100: Release Index kernel function error

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

Returns

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

2.25.1.13 **int MSXE17xx_MFIncCounterInitAndEnableLatch (xsd_unsignedLong ulMFModuleIndex, xsd_unsignedLong ulLatchSource, xsd_unsignedLong ulCondition, xsd_unsignedLong ulAutoMode, xsd_unsignedLong ulOption01, xsd_unsignedLong ulOption02, struct MSXE17xx_Response * Response)**

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

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (0)
DWORD 3 : Event mask
    D30-D0 :
        0: Compare
        1: Frequency measurement
        2: Hardware trigger latch occur
        3: Synchro input latch occur
        4: Index input latch occur
    D31 :
        0: No error occur
        1: Amplitude or Frequency error occur.
DWORD 4 :
    D31-D0: Counter value (DWORD)
```

Parameters

- [in] **ulMFModuleIndex** : Index of the multifunction sub module (0 to 3).
- [in] **ulLatchSource** : Latch source.
- 0: Index input
 - 1: Hardware trigger
 - 2: Synchro input
- [in] **ulCondition** : Previously condition for accept the latch source
- 0: No previously condition required
 - 1: Index input condition required (Only if index input not selected selected for the latch source)
 - 2: Hardware trigger condition required (Only if hardware trigger not selected selected for the latch source)
 - 3: Synchro input condition required (Only if synchro input not selected selected for the latch source)
 - 4: Reference input (D) condition required
- [in] **ulAutoMode** : Action mode
- 0: Do not use auto mode (action is done only once)
 - 1: Use auto mode (action is done continuously)
- [in] **ulOption01** : set it to 0
- [in] **ulOption02** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

2.25.1.14 `int MSXE17xx__MFIncCounterDisableAndReleaseLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response * Response)`

Parameters

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

[in] **ulLatchSource** : Latch source to disable and release.

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

2.25.1.15 `int MSXE17xx__MFINcCounterInitAndEnableClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

- [in] **ulMFModuleIndex** : Index of the multifunction sub module (0 to 3).
- [in] **ulClearSource** : Clear source.
- 0: Index input
 - 1: Hardware trigger
 - 2: Synchro input
- [in] **ulCondition** : Previously condition for accept the clear source
- 0: No previously condition required
 - 1: Index input condition required (Only if index input not selected selected for the clear source)
 - 2: Hardware trigger condition required (Only if hardware trigger not selected selected for the clear source)
 - 3: Synchro input condition required (Only if synchro input not selected selected for the clear source)
 - 4: Reference input (D) condition required
- [in] **ulAutoMode** : Action mode
- 0: Do not use auto mode (action is done only once)
 - 1: Use auto mode (action is done continuously)
- [in] **ulOption01** : set it to 0
- [in] **ulOption02** : set it to 0
- [out] **Response** :
- iReturnValue** :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter
 - -4: Incremental counter not initialised
 - -5: Clear logic already initialised
 - -6: Clear source selection error
 - -7: Previously condition selection error
 - -8: Auto mode selection error
 - -9: Hardware trigger not initialised. Refer to MSXE17xx__MFINcCounterInitHardwareTrigger
 - -10: Index input not initialised. Refer to MSXE17xx__MFINcCounterInitIndex
 - -100: Init and enable counter clear kernel function error
- syserrno** : system-error code (the value of the libc "errno" code)

Returns

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

2.25.1.16 `int MSXE17xx_MFIncCounterDisableAndReleaseClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulOption01, struct MSXE17xx_Response * Response)`

Parameters

- [in] *ulMFModuleIndex* : index of the multifunction sub module (0 to 3).
- [in] *ulClearSource* : Clear source to disable and release.
- [in] *ulOption01* : set it to 0
- [out] *Response* :
iReturnValue :
 - 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter
 - -4: Incremental counter not initialised
 - -5: Clear logic not initialised
 - -6: Clear source selection error
 - -100: Disable and release counter clear register kernel function error*syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

2.25.1.17 `int MSXE17xx_MFIncCounterInitAndEnableLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulHardwareTriggerEdgeSelection, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulOption01, struct MSXE17xx_Response * Response)`

It is recommended to use the "MSXE17xx_MFIncCounterInitAndEnableLatch" and "MSXE17xx_MFIncCounterInitAndEnableClear" functions

Parameters

- [in] *ulMFModuleIndex* : index of the multifunction sub module (0 to 3).
- [in] *ulLatchRegister* : 0: First Latch Register used with trigger input, 1: Second Lath Register used with synchro input
- [in] *ulHardwareTriggerEdgeSelection* : not used for the synchro input
 - 01 : rising front (Only if trigger input selected)
 - 10 : falling front (Only if trigger input selected)
 - 11 : Both front (Only if trigger input selected)
- [in] *ulHardwareTriggerCount* : not used for the synchro input
 Define the number of trigger events before the action occur
 1 : all trigger event start the action
 max value : 65535
- [in] *ulOption01* : set it to 0

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Latch register selection error
- -4: Trigger edge selection error
- -5: Trigger count selection error
- -6: Multifunction sub module is not a incremental counter
- -7: Incremental counter not initialised
- -8: Latch logic already initialised
- -100: Init and enable counter latch register kernel function error

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

Returns

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

2.25.1.18 `int MSXE17xx_MFIncCounterDisableAndReleaseLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulOption01, struct MSXE17xx_Response * Response)`

Parameters

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

[in] **ulLatchRegister** : 0: First Lath Register used with 24 V input, 1: Second Lath Register used with synchro input

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

2.25.1.19 int MSXE17xx_MFIncCounterInitAndEnableFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulTimingInterval, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx_Response * Response)

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

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 : Counter index
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (0)
DWORD 3 : Event mask
    0: Compare
    1: Frequency measurement
    2: Hardware trigger latch occur
    3: Synchro input latch occur
    4: Counter index
DWORD 4 :
    D31-D0: Counter value (DWORD)
```

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
- [in] **ulTimingInterval** : Timing interval from 1 to 0xFFFF, one step correspond to 100 ns.
 - 1: time interval = 100 ns
 - 2: time interval = 200 ns
 - ...
- [in] **ulOption01** : Set the measurement mode
 - 0: One shot, the frequency measure is done at each call of this function
 - 1: Triggered mode, this function has to be called only once (to initialize the measure). The synchro trigger is used to trigger the measure. The synchro trigger cycle time should be > frequency time interval
- [in] **ulOption02** : Timebase selection
 - 0: 100ns
 - 1: 100us
- [out] **Response** :
 - iReturnValue** :
 - 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Timing interval selection error
 - -4: Multifunction sub module is not a incremental counter
 - -5: Incremental counter not initialised
 - -6: Frequency measurement already initialised and started
 - -7: Frequency measurement ulOption01 parameter wrong value
 - -8: Frequency measurement ulOption02 parameter wrong value
 - -100: Init and enable counter Frequency measurement kernel function error
 - syserrno** : system-error code (the value of the libc "errno" code)

Returns

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

2.25.1.20 `int MSXE17xx__MFIncCounterDisableAndReleaseFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response * Response)`

Parameters

- [in] ***ulMFModuleIndex*** : index of the multifunction sub module (0 to 3).
- [in] ***ulOption01*** : set it to 0
- [out] ***Response*** :
- iReturnValue*** :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter
 - -4: Incremental counter not initialised
 - -5: Frequency measurement not initialised
 - -100: Disable and release counter latchd register kernel function error
- syserrno*** : system-error code (the value of the libc "errno" code)

Returns

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

2.25.1.21 `int MSXE17xx__MFIncCounterInitAndEnableIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulReferenceAction, xsd__unsignedLong ulIndexOperation, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, struct MSXE17xx__Response * Response)`

It is recommended to use the "MSXE17xx__MFIncCounterInitAndEnableLatch" and "MSXE17xx__MFIncCounterInitAndEnableClear" functions

The index operation can be programmed. If used, the reference signal enable the index management. Rising edge on reference enable the index management. Rising edge on index reset the reference management.

Parameters

- [in] ***ulMFModuleIndex*** : index of the multifunction sub module (0 to 3).
- [in] ***ulReferenceAction*** : Reference action
- 0: do not use the D pin as reference,
 - 1: use the D pin as reference
- [in] ***ulIndexOperation*** : Index operation
- 0 = MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_COUNTER,
 - 1 = MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_COUNTER,
 - 2 = MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER,
 - 3 = MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER,
- [in] ***ulAutoMode*** : AutoMode 0: do not use auto mode (action is done only once), 1: use auto mode (action is done continuously)
- [in] ***ulOption01*** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

2.25.1.22 `int MSXE17xx__MFIncCounterDisableAndReleaseIndex (xsd__unsignedLong
ulMFModuleIndex, struct MSXE17xx__Response * Response)`

Parameters

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

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

- 0: SOAP_OK
- <> 0: 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 MSXE170x__DigitalIOGetNumberResponse Struct Reference

Returns the number of digital IO available on the module.

Data Fields

- struct [DefaultResponse](#) *sResponse*

Default return values.

- [xsd__unsignedLong](#) *ulNumberOfDigitalIO*

Number of digital IO available on the module (up to 16).

3.3.1 Detailed Description

Parameters

[in] *None*

[out] *Response* :

sResponse.iReturnValue :

- 0: means the remote function performed OK
 - -1: means an system error occurred (check errno in this case)
- sResponse.syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

3.3.2 Field Documentation

3.3.2.1 struct DefaultResponse MSXE170x__DigitalIOGetNumberResponse::sResponse

3.3.2.2 xsd__unsignedLong MSXE170x__DigitalIOGetNumberResponse::ulNumberOfDigitalIO

3.4 MSXE170x__IOWatchdogGetStatusAndValueResponse Struct Reference

Data Fields

- struct [DefaultResponse sResponse](#)
Default return values.
- [xsd__unsignedLong ulStatus](#)
Watchdog current status information.
- [xsd__unsignedLong ulValue](#)
Watchdog current value information.
- [xsd__unsignedLong ulInfo](#)
reserved

3.4.1 Field Documentation

3.4.1.1 struct DefaultResponse MSXE170x__IOWatchdogGetStatusAndValueResponse::sResponse

3.4.1.2 xsd__unsignedLong MSXE170x__IOWatchdogGetStatusAndValueResponse::ulStatus

3.4.1.3 xsd__unsignedLong MSXE170x__IOWatchdogGetStatusAndValueResponse::ulValue

3.4.1.4 xsd__unsignedLong MSXE170x__IOWatchdogGetStatusAndValueResponse::ulInfo

3.5 MSXE170x__MFIncCounterGetNumberResponse Struct Reference

Returns the number of counter available on the module.

Data Fields

- struct [DefaultResponse sResponse](#)
Default return values.
- [xsd__unsignedLong ulNumberOfCounter](#)
Number of digital counter available on the module (up to 4).

3.5.1 Detailed Description

Parameters

[in] *None*

[out] *Response* :

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured (check errno in this case)

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

Returns

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

3.5.2 Field Documentation

3.5.2.1 struct DefaultResponse MSXE170x__MFIncCounterGetNumberResponse::sResponse

3.5.2.2 xsd__unsignedLong MSXE170x__MFIncCounterGetNumberResponse::ulNumberOfCounter

3.6 MSXE170x__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)

3.6.1 Field Documentation

3.6.1.1 xsd__int MSXE170x__Response::iReturnValue

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

3.6.1.2 xsd__int MSXE170x__Response::syserrno

3.7 MSXE170x__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

3.7.1 Field Documentation

3.7.1.1 struct [DefaultResponse MSXE170x__unsignedLongResponse::sResponse](#)

3.7.1.2 [xsd__unsignedLong MSXE170x__unsignedLongResponse::ulValue](#)

3.8 MSXE170x__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

3.8.1 Field Documentation

3.8.1.1 struct [DefaultResponse MSXE170x__unsignedLongTimeStampResponse::sResponse](#)

3.8.1.2 [xsd__unsignedLong MSXE170x__unsignedLongTimeStampResponse::ulValue](#)

3.8.1.3 [xsd__unsignedLong MSXE170x__unsignedLongTimeStampResponse::ulTimeStampLow](#)

3.8.1.4 [xsd__unsignedLong MSXE170x__unsignedLongTimeStampResponse::ulTimeStampHigh](#)

3.9 MSXE17xx__DigitalIOGetNumberResponse Struct Reference

Data Fields

- struct [DefaultResponse sResponse](#)

Default return values.

- [xsd__unsignedLong ulNumberOfDigitalIO](#)

Number of digital IO available on the module (up to 16).

3.9.1 Field Documentation

3.9.1.1 struct `DefaultResponse MSXE17xx__DigitalIOGetNumberResponse::sResponse`

3.9.1.2 `xsd__unsignedLong MSXE17xx__DigitalIOGetNumberResponse::ulNumberOfDigitalIO`

3.10 MSXE17xx__IOWatchdogGetStatusAndValueResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
Default return values.
- `xsd__unsignedLong ulStatus`
Watchdog current status information.
- `xsd__unsignedLong ulValue`
Watchdog current value information.
- `xsd__unsignedLong ulInfo`
reserved

3.10.1 Field Documentation

3.10.1.1 struct `DefaultResponse MSXE17xx__IOWatchdogGetStatusAndValueResponse::sResponse`

3.10.1.2 `xsd__unsignedLong MSXE17xx__IOWatchdogGetStatusAndValueResponse::ulStatus`

3.10.1.3 `xsd__unsignedLong MSXE17xx__IOWatchdogGetStatusAndValueResponse::ulValue`

3.10.1.4 `xsd__unsignedLong MSXE17xx__IOWatchdogGetStatusAndValueResponse::ulInfo`

3.11 MSXE17xx__MFIncCounterGetNumberResponse Struct Reference

Data Fields

- struct `DefaultResponse sResponse`
Default return values.
- `xsd__unsignedLong ulNumberOfCounter`
Number of digital counter available on the module (up to 4).

3.11.1 Field Documentation

3.11.1.1 struct `DefaultResponse` `MSXE17xx__MFINcCounterGetNumberResponse::sResponse`

3.11.1.2 `xsd__unsignedLong` `MSXE17xx__MFINcCounterGetNumberResponse::ulNumberOfCounter`

3.12 MSXE17xx__MFPWMGetConfigurationResponse Struct Reference

Data Fields

- struct `DefaultResponse` `sResponse`
Default return values.
- `xsd__unsignedLong` `ulInitialised`
Return the initialisation flag
.
- `xsd__unsignedLong` `ulTimeBase`
PWM time base selection
.
- `xsd__unsignedLong` `ulLowTime`
PWM low time reload value (1 to 16777215)
If the extern clock is used this define the number of clocks for the low level.
- `xsd__unsignedLong` `ulHighTime`
PWM high time reload value (1 to 16777215)
If the extern clock is used this define the number of clocks for the high level.
- `xsd__unsignedLong` `ulEnabled`
Return the enabled status
.
- `xsd__unsignedLong` `ulStartLevel`
Return the start level, time
.
- `xsd__unsignedLong` `ulExternGate`
Return the extern gate initialisation
.
- `xsd__unsignedLong` `ulStopMode`
Return the current stop mode
.
- `xsd__unsignedLong` `ulStopLevel`
Return the current selected stop level
.

- [xsd__unsignedLong ulSynchroOut](#)
Give the possibility to generate a synchro trigger
.
- [xsd__unsignedLong ulDataFrame](#)
Give the possibility to send via the data server PWM informations
.
- [xsd__unsignedLong ulInfo01](#)
Reserved.
- [xsd__unsignedLong ulInfo02](#)
Reserved.

3.12.1 Field Documentation

3.12.1.1 struct DefaultResponse MSXE17xx__MFPWMGetConfigurationResponse::sResponse

3.12.1.2 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulInitialised

- 0: PWN not initialised
- 1: PWM initialised

3.12.1.3 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulTimeBase

- 0: 250ns (4MHz)
- 1: micro s (1MHz)
- 2: ms (1kHz)
- 3: s (Hz)
- 4: Diff input C for PWM0 and D for PWM1
- 5: Trigger input (24V)
- 6: Synchro trigger

3.12.1.4 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulLowTime

If the internal time base is used this define the low level time : Low time = ulTimeBase * ulLowTime.

3.12.1.5 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulHighTime

If the internal time base is used this define the high level time : High time = ulTimeBase * ulHighTime.

3.12.1.6 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulEnabled

- 0: PWM disabled
- 1: PWN enabled

3.12.1.7 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulStartLevel

- 0: PWM start with the low level
- 1: PWM start with the high level

3.12.1.8 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulExternGate

- 0: Extern gate not used
- 1: Diff input C (PWM0) or D (PWM1) is used as gate
- 2: Input trigger (24V) is used as gate

3.12.1.9 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulStopMode

- 0: Stop the PWM signal directly after a stop condition
- 1: Stop the PWM after the current period is stopped

3.12.1.10 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulStopLevel

- 0: The output signal keep the level after the stop condition
- 1: The output signal is set to low after the stop condition
- 2: The output signal is set to high after the stop condition

3.12.1.11 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulSynchroOut

- 0: Disabled
- 1: Generate a synchro trigger be each start of period
- 2: Generate a synchro trigger be each end of period
- 3: Generate a synchro trigger be each start and end of period

3.12.1.12 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulDataFrame

- 0: Disabled
- 1: Enabled

For each start and end of period the data server send a 5 DWORD frame with following informations:

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (1)
DWORD 3 : Event mask
    8: PWM0. First period begin
    9: PWM0. End of period
    10: PWM0. End of last period. Stop condition occur
    11: PWM1. First period begin
    12: PWM1. End of period
    13: PWM1. End of last period. Stop condition occur
DWORD 4 :
    D23-D0: Number of occurred periods
    D24: 0: Diff input C for PWM0 or D for PWM1 is low
         1: Diff input C for PWM0 or D for PWM1 is high
    D25: 0: Extern trigger input (24V) is low
         1: Extern trigger input (24V) is high
    D26: 0: PWM output is low
         1: PWM output is high
```

3.12.1.13 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulInfo01

Return 0

3.12.1.14 xsd__unsignedLong MSXE17xx__MFPWMGetConfigurationResponse::ulInfo02

Return 0

3.13 MSXE17xx__MFPWMGetStateResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) sResponse
Default return values.
- [xsd__unsignedLong](#) ulInitialised
Return the initialisation flag
.
- [xsd__unsignedLong](#) ulEnabled
Return the enabled status
.
- [xsd__unsignedLong](#) ulProgress

Return the progress status

.

- [xsd__unsignedLong ulOutput](#)

Return the PWM output signal state

.

- [xsd__unsignedLong ulDiffInput](#)

Return the diff input state

.

- [xsd__unsignedLong ulTriggerInput](#)

Return the extern trigger state

.

- [xsd__unsignedLong ulCounter](#)

Return the number of completed PWM periods.

- [xsd__unsignedLong ulInfo01](#)

Reserved.

- [xsd__unsignedLong ulInfo02](#)

Reserved.

3.13.1 Field Documentation

3.13.1.1 struct DefaultResponse MSXE17xx__MFPWMGetStateResponse::sResponse

3.13.1.2 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulInitialised

- 0: PWN not initialised
- 1: PWM initialised

3.13.1.3 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulEnabled

- 0: PWM disabled
- 1: PWN enabled

3.13.1.4 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulProgress

- 0: PWM not in progress
- 1: PWM not in progress

3.13.1.5 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulOutput

- 0: PWM output is low
- 1: PWM output is high

3.13.1.6 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulDiffInput

- 0: The diff input C (for PWM0) or F (for PWM1) is low
- 1: The diff input C (for PWM0) or F (for PWM1) is high

3.13.1.7 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulTriggerInput

- 0: The trigger input (24V) is low
- 1: The trigger input (24V) is high

3.13.1.8 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulCounter**3.13.1.9 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulInfo01**

Return 0

3.13.1.10 xsd__unsignedLong MSXE17xx__MFPWMGetStateResponse::ulInfo02

Return 0

3.14 MSXE17xx__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)

3.14.1 Field Documentation**3.14.1.1 xsd__int MSXE17xx__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

3.14.1.2 xsd__int MSXE17xx__Response::syserrno

3.15 MSXE17xx__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

3.15.1 Field Documentation

3.15.1.1 struct DefaultResponse MSXE17xx__unsignedLongResponse::sResponse

3.15.1.2 xsd__unsignedLong MSXE17xx__unsignedLongResponse::ulValue

3.16 MSXE17xx__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

3.16.1 Field Documentation

3.16.1.1 struct `DefaultResponse MSXE17xx__unsignedLongTimeStampResponse::sResponse`

3.16.1.2 `xsd__unsignedLong MSXE17xx__unsignedLongTimeStampResponse::ulValue`

3.16.1.3 `xsd__unsignedLong MSXE17xx__unsignedLongTimeStampResponse::ulTimeStampLow`

3.16.1.4 `xsd__unsignedLong MSXE17xx__ - unsignedLongTimeStampResponse::ulTimeStampHigh`

3.17 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.

3.17.1 Field Documentation

3.17.1.1 struct `DefaultResponse MXCommon__ByteArrayResponse::sResponse`

3.17.1.2 struct `ByteArray MXCommon__ByteArrayResponse::sArray`

3.18 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.

3.18.1 Field Documentation

3.18.1.1 struct DefaultResponse MXCommon__FileResponse::sResponse

3.18.1.2 struct ByteArray MXCommon__FileResponse::sArray

3.18.1.3 xsd__unsignedLong MXCommon__FileResponse::ulEOF

3.19 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.

3.19.1 Field Documentation

3.19.1.1 struct DefaultResponse MXCommon__GetAutoConfigurationFileResponse::sResponse

3.19.1.2 struct ByteArray MXCommon__GetAutoConfigurationFileResponse::bArray

3.19.1.3 xsd__unsignedLong MXCommon__GetAutoConfigurationFileResponse::ulEOF

3.20 MXCommon__GetEthernetLinksStatesResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) sResponse

Default return values.

- struct [sGetEthernetLinksStatesPort](#) sPort0

- struct [sGetEthernetLinksStatesPort](#) sPort1

3.20.1 Field Documentation

3.20.1.1 struct `DefaultResponse MXCommon__GetEthernetLinksStatesResponse::sResponse`

3.20.1.2 struct `sGetEthernetLinksStatesPort MXCommon__GetEthernetLinksStatesResponse::sPort0`

3.20.1.3 struct `sGetEthernetLinksStatesPort MXCommon__GetEthernetLinksStatesResponse::sPort1`

3.21 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.

3.21.1 Field Documentation

3.21.1.1 struct `DefaultResponse MXCommon__GetHardwareTriggerFilterTimeResponse::sResponse`

3.21.1.2 `xsd__unsignedLong MXCommon__GetHardwareTriggerFilterTimeResponse::ulFilterTime`

3.21.1.3 `xsd__unsignedLong MXCommon__GetHardwareTriggerFilterTimeResponse::ulInfo01`

3.21.1.4 `xsd__unsignedLong MXCommon__GetHardwareTriggerFilterTimeResponse::ulInfo02`

3.22 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.

3.22.1 Field Documentation

3.22.1.1 struct DefaultResponse MXCommon__GetHardwareTriggerStateResponse::sResponse

3.22.1.2 [xsd__unsignedLong](#) MXCommon__GetHardwareTriggerStateResponse::ulState

3.22.1.3 [xsd__unsignedLong](#) MXCommon__GetHardwareTriggerStateResponse::ulInfo01

3.22.1.4 [xsd__unsignedLong](#) MXCommon__GetHardwareTriggerStateResponse::ulInfo02

3.23 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.

3.23.1 Field Documentation

- 3.23.1.1 struct `DefaultResponse MXCommon__ - GetModuleTemperatureValueAndStatusResponse::sResponse`
- 3.23.1.2 `xsd__double MXCommon__ - GetModuleTemperatureValueAndStatusResponse::dTemperatureValue`
- 3.23.1.3 `xsd__unsignedLong MXCommon__ - GetModuleTemperatureValueAndStatusResponse::ulTemperatureStatus`
- 3.23.1.4 `xsd__unsignedLong MXCommon__ - GetModuleTemperatureValueAndStatusResponse::ulInfo`

3.24 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).

3.24.1 Field Documentation

- 3.24.1.1 struct `DefaultResponse MXCommon__GetTimeResponse::sResponse`
- 3.24.1.2 `xsd__unsignedLong MXCommon__GetTimeResponse::ulLowTime`
- 3.24.1.3 `xsd__unsignedLong MXCommon__GetTimeResponse::ulHighTime`

3.25 MXCommon__GetUpTimeResponse Struct Reference

Data Fields

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

3.25.1 Field Documentation

3.25.1.1 struct DefaultResponse MXCommon__GetUpTimeResponse::sResponse

3.25.1.2 xsd__unsignedLong MXCommon__GetUpTimeResponse::ulUpTime

3.26 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\(\)](#)).

3.26.1 Field Documentation

3.26.1.1 xsd__int MXCommon__Response::iReturnValue

3.26.1.2 xsd__int MXCommon__Response::syserrno

3.27 MXCommon__TestCustomerIDResponse Struct Reference

Data Fields

- struct [DefaultResponse](#) sResponse

Default return values.

- struct [ByteArray](#) bValueArray

non encrypted value

- struct [ByteArray](#) bCryptedValueArray

encrypted value

3.27.1 Field Documentation

3.27.1.1 struct DefaultResponse MXCommon__TestCustomerIDResponse::sResponse

3.27.1.2 struct ByteArray MXCommon__TestCustomerIDResponse::bValueArray

3.27.1.3 struct ByteArray MXCommon__TestCustomerIDResponse::bCryptedValueArray

3.28 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.

3.28.1 Field Documentation

3.28.1.1 struct DefaultResponse MXCommon__unsignedLongResponse::sResponse

3.28.1.2 [xsd__unsignedLong](#) MXCommon__unsignedLongResponse::ulValue

3.29 sGetEthernetLinksStatesPort Struct Reference

Data Fields

- [xsd__unsignedLong](#) ulState
- [xsd__unsignedLong](#) ulSpeed
- [xsd__unsignedLong](#) ulDuplex
- [xsd__unsignedLong](#) ulInfo1
- [xsd__unsignedLong](#) ulInfo2

3.29.1 Field Documentation

3.29.1.1 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulState`

3.29.1.2 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulSpeed`

3.29.1.3 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulDuplex`

3.29.1.4 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulInfo1`

3.29.1.5 `xsd__unsignedLong sGetEthernetLinksStatesPort::ulInfo2`

3.30 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

3.30.1 Field Documentation

3.30.1.1 `xsd__unsignedLong* UnsignedLongArray::__ptr`

3.30.1.2 `int UnsignedLongArray::__size`

3.30.1.3 `int UnsignedLongArray::__offset`

3.31 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

3.31.1 Field Documentation

3.31.1.1 `xsd__unsignedShort* UnsignedShortArray::__ptr`

3.31.1.2 `int UnsignedShortArray::__size`

3.31.1.3 `int UnsignedShortArray::__offset`

3.32 `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

3.32.1 Field Documentation

3.32.1.1 `unsigned char* xsd__base64Binary::__ptr`

3.32.1.2 `int xsd__base64Binary::__size`

Chapter 4

File Documentation

4.1 MSXE170x_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 [MSXE170x__Response](#)
- struct [MSXE170x__unsignedLongResponse](#)
- struct [MSXE170x__unsignedLongTimeStampResponse](#)
- struct [MSXE170x__DigitalIOGetNumberResponse](#)

Returns the number of digital IO available on the module.

- struct [MSXE170x__IOWatchdogGetStatusAndValueResponse](#)
- struct [MSXE170x__MFIncCounterGetNumberResponse](#)

Returns the number of counter available on the module.

- struct [MSXE17xx__Response](#)
- struct [MSXE17xx__unsignedLongResponse](#)
- struct [MSXE17xx__unsignedLongTimeStampResponse](#)
- struct [MSXE17xx__DigitalIOGetNumberResponse](#)
- struct [MSXE17xx__IOWatchdogGetStatusAndValueResponse](#)
- struct [MSXE17xx__MFPWMGetConfigurationResponse](#)
- struct [MSXE17xx__MFPWMGetStateResponse](#)
- struct [MSXE17xx__MFIncCounterGetNumberResponse](#)

Defines

- #define [MSXE170X_COUNTER_QUADRUPLE_MODE](#) 0x4
In the quadruple mode, the edge analysis circuit generates a counting pulse from each edge of two signals which are phase-shifted in relation to each other.
- #define [MSXE170X_COUNTER_DOUBLE_MODE](#) 0x2
Same function as quadruple mode, except only 2 of the 4 edges are analysed.
- #define [MSXE170X_COUNTER_SIMPLE_MODE](#) 0x1
Same function as quadruple mode, except one of the 4 edges is analysed in each period.
- #define [MSXE170X_COUNTER_DIRECT_MODE](#) 0x0
In the direct mode both edge analysis circuits become inactive.
- #define [MSXE170X_COUNTER_HYSTERESIS_ON](#) 0x1
On both edge analysis circuit a hysteresis switch is available.
- #define [MSXE170X_COUNTER_HYSTERESIS_OFF](#) 0x0
The first pulse will not be suppressed after a change of rotation.
- #define [MSXE170X_COUNTER_INCREMENT](#) 0x0
The counter increments after each counting pulse.
- #define [MSXE170X_COUNTER_DECREMENT](#) 0x1
The counter decrements after each counting pulse.

- #define [MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_COUNTER](#) 0x0
After an index signal (Low level), the counter value (32-bit) is latched into the first latch register and then deleted (32-bit).
- #define [MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_COUNTER](#) 0x1
After an index signal (High level), the counter value (32-bit) is latched into the first latch register and then deleted (32-bit).
- #define [MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER](#) 0x2
After an index signal (Low level), the counter value (32-bit) is latched into the first latch register.
- #define [MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER](#) 0x3
After an index signal (High level), the counter value (32-bit) is latched into the first latch register.

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 [MXCommon__SetFilterChannels](#) (struct [xsd__base64Binary](#) *ChannelList, struct [MXCommon__Response](#) *Response)

This function sets or resets a filter to a channel.

- int [MSXE170x__DigitalIOGetNumber](#) (void ___, struct [MSXE170x__DigitalIOGetNumberResponse](#) *Response)
- int [MSXE170x__DigitalIOInitPortConfiguration](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulPortConfiguration, struct [MSXE170x__Response](#) *Response)

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

- int [MSXE170x__DigitalIOReadChannelValue](#) (xsd__unsignedLong ulChannel, struct [MSXE170x__unsignedLongResponse](#) *Response)

Read a digital i/o channel value.

- int [MSXE170x__DigitalIOReadAllChannelsValue](#) (void ___, struct [MSXE170x__unsignedLongResponse](#) *Response)

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

- int [MSXE170x__DigitalIOWriteChannelValue](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulChannelValue, struct [MSXE170x__Response](#) *Response)

write a digital i/o channel value

- int [MSXE170x__DigitalIOWriteAllChannelsValue](#) (xsd__unsignedLong ulChannelValue, struct [MSXE170x__Response](#) *Response)

write all digital i/o channels value

- int [MSXE170x__DigitalIOReleasePortConfiguration](#) (xsd__unsignedLong ulPort, struct [MSXE170x__Response](#) *Response)

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

- int [MSXE170x__DigitalIOTestShortCircuit](#) (xsd__unsignedLong ulOption, struct [MSXE170x__unsignedLongResponse](#) *Response)

Test short circuit status.

- int [MSXE170x__DigitalIORearmShortCircuit](#) (xsd__unsignedLong ulOption, struct [MSXE170x__Response](#) *Response)

Rearm digital outputs after a short circuit happened.

- int [MSXE170x__IOWatchdogInitAndStart](#) (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct [MSXE170x__Response](#) *Response)

Init and start the digital output IO watchdog.

- int [MSXE170x__IOWatchdogStopAndRelease](#) (xsd__unsignedLong ulOption, struct [MSXE170x__Response](#) *Response)

Stop and release the digital output watchdog.

- int [MSXE170x__IOWatchdogGetStatusAndValue](#) (xsd__unsignedLong ulOption, struct [MSXE170x__IOWatchdogGetStatusAndValueResponse](#) *Response)

Get watchdog current status and value information.

- `int MSXE170x__MFCommonSetInputsFilter (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulInputAFilterValue, xsd__unsignedLong ulInputBFilterValue, xsd__unsignedLong ulInputCFilterValue, xsd__unsignedLong ulInputDFilterValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE170x__Response *Response)`

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

- `int MSXE170x__MFCommonReferenceVoltageActivation (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulActivationFlag, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response *Response)`

Permit to activate the reference voltage to pin D-.

- `int MSXE170x__MFIncCounterGetNumber (void *__, struct MSXE170x__MFIncCounterGetNumberResponse *Response)`
- `int MSXE170x__MFIncCounterInit (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterMode, xsd__unsignedLong ulCounterOption, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE170x__Response *Response)`

Initialise the counter.

- `int MSXE170x__MFIncCounterRelease (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response *Response)`

Release the counter.

- `int MSXE170x__MFIncCounterSetFIFO0Level (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulFIFOLevel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response *Response)`

Use the function MSXE17xx__MFCommonSetFIFO0Level.

- `int MSXE170x__MFIncCounterRead32BitValue (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__unsignedLongTimeStampResponse *Response)`

Read the 32 bits counter value.

- `int MSXE170x__MFIncCounterWrite32BitValue (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterValue, struct MSXE170x__Response *Response)`

write a 32 bits counter value

- `int MSXE170x__MFIncCounterClear (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response *Response)`

Clear the 32 bits counter.

- `int MSXE170x__MFIncCounterInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response *Response)`

Init and enable a counter compare value.

- `int MSXE170x__MFIncCounterDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response *Response)`

Disable and Release a counter compare value.

- int [MSXE170x__MFINcCounterInitAndEnableLatchRegister](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulHardwareTriggerEdgeSelection, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulOption01, struct [MSXE170x__Response](#) *Response)

Init and enable a counter latch register.

- int [MSXE170x__MFINcCounterDisableAndReleaseLatchRegister](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulOption01, struct [MSXE170x__Response](#) *Response)

Disable and Release a counter latch register.

- int [MSXE170x__MFINcCounterInitAndEnableFrequencyMeasurement](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulTimingInterval, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE170x__Response](#) *Response)

Init and enable a counter frequency measurement.

- int [MSXE170x__MFINcCounterDisableAndReleaseFrequencyMeasurement](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE170x__Response](#) *Response)

Disable and Release a counter frequency measurement.

- int [MSXE170x__MFINcCounterInitAndEnableIndex](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulReferenceAction, xsd__unsignedLong ulIndexOperation, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, struct [MSXE170x__Response](#) *Response)

Init and enable a counter index.

- int [MSXE170x__MFINcCounterDisableAndReleaseIndex](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE170x__Response](#) *Response)

Disable and Release a counter index.

- int [MSXE17xx__DigitalIOGetNumber](#) (void ___, struct [MSXE17xx__DigitalIOGetNumberResponse](#) *Response)

Returns the number of digital IO available on the module.

- int [MSXE17xx__DigitalIOInitPortConfiguration](#) (xsd__unsignedLong ulPort, xsd__unsignedLong ulPortConfiguration, struct [MSXE17xx__Response](#) *Response)

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

- int [MSXE17xx__DigitalIOReadChannelValue](#) (xsd__unsignedLong ulChannel, struct [MSXE17xx__unsignedLongResponse](#) *Response)

Read a digital i/o channel value.

- int [MSXE17xx__DigitalIOReadAllChannelsValue](#) (void ___, struct [MSXE17xx__unsignedLongResponse](#) *Response)

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

- int [MSXE17xx__DigitalIOWriteChannelValue](#) (xsd__unsignedLong ulChannel, xsd__unsignedLong ulChannelValue, struct [MSXE17xx__Response](#) *Response)

write a digital i/o channel value

- `int MSXE17xx__DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE17xx__Response *Response)`
write all digital i/o channels value
- `int MSXE17xx__DigitalIOReleasePortConfiguration (xsd__unsignedLong ulPort, struct MSXE17xx__Response *Response)`
Release a digital i/o port (2 channels).
- `int MSXE17xx__DigitalIOTestShortCircuit (xsd__unsignedLong ulOption, struct MSXE17xx__unsignedLongResponse *Response)`
Test short circuit status.
- `int MSXE17xx__DigitalIORearmShortCircuit (xsd__unsignedLong ulOption, struct MSXE17xx__Response *Response)`
Rearm digital outputs after a short circuit happened.
- `int MSXE17xx__IOWatchdogInitAndStart (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct MSXE17xx__Response *Response)`
Init and start the digital output IO watchdog.
- `int MSXE17xx__IOWatchdogStopAndRelease (xsd__unsignedLong ulOption, struct MSXE17xx__Response *Response)`
Stop and release the digital output watchdog.
- `int MSXE17xx__IOWatchdogGetStatusAndValue (xsd__unsignedLong ulOption, struct MSXE17xx__IOWatchdogGetStatusAndValueResponse *Response)`
Get watchdog current status and value information.
- `int MSXE17xx__MFCommonGetSubModuleFunctionality (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__unsignedLongResponse *Response)`
Get the selected sub module functionality.
- `int MSXE17xx__MFCommonSetInputsFilter (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulInputAFilterValue, xsd__unsignedLong ulInputBFilterValue, xsd__unsignedLong ulInputCFilterValue, xsd__unsignedLong ulInputDFilterValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__Response *Response)`
Set a filter to the input of a multifunction sub module.
- `int MSXE17xx__MFCommonReferenceVoltageActivation (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulActivationFlag, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`
Permit to activate the reference voltage to pin D-.
- `int MSXE17xx__MFCommonEnableDisableTriggerGate (xsd__unsignedLong ulTriggerConfiguration, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`
Enable / disable trigger gate.

- `int MSXE17xx__MFCommonSetFIFO0Level (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulFIFOLevel, xsd__unsignedLong ulTimeOutTimeBase, xsd__unsignedLong ulReloadValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`

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

- `int MSXE17xx__MFPWMGetNumber (void *_ , struct MSXE17xx__unsignedLongResponse *Response)`

Returns the number of PWM sub modules available on the module.

- `int MSXE17xx__MFPWMInitAndEnable (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, xsd__unsignedLong ulStartLevel, xsd__unsignedLong ulExternGate, xsd__unsignedLong ulStopMode, xsd__unsignedLong ulStopLevel, xsd__unsignedLong ulSynchroOut, xsd__unsignedLong ulDataFrame, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__Response *Response)`

Initialise and enable the selected PWM module.

- `int MSXE17xx__MFPWMSetNewTiming (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulLowTime, xsd__unsignedLong ulHighTime, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`

Set a new PWM timing.

- `int MSXE17xx__MFPWMDisable (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`

Disable the selected PWM.

- `int MSXE17xx__MFPWMDisableAndRelease (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`

Disable and release the selected PWM.

- `int MSXE17xx__MFPWMGetConfiguration (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, struct MSXE17xx__MFPWMGetConfigurationResponse *Response)`

Get the selected PWM initialisation.

- `int MSXE17xx__MFPWMGetState (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, struct MSXE17xx__MFPWMGetStateResponse *Response)`

Get the selected PWM state.

- `int MSXE17xx__MFINcCounterGetNumber (void *_ , struct MSXE17xx__MFINcCounterGetNumberResponse *Response)`

Returns the number of counter available on the module.

- `int MSXE17xx__MFINcCounterInit (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterMode, xsd__unsignedLong ulCounterOption, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__Response *Response)`

Initialise the counter.

- `int MSXE17xx_MFIncCounterRelease (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)`

Release the counter.

- `int MSXE17xx_MFIncCounterRead32BitValue (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__unsignedLongTimeStampResponse *Response)`

Read the 32 bits counter value.

- `int MSXE17xx_MFIncCounterWrite32BitValue (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterValue, struct MSXE17xx__Response *Response)`

write a 32 bits counter value

- `int MSXE17xx_MFIncCounterClear (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)`

Clear the 32 bits counter.

- `int MSXE17xx_MFIncCounterInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchronroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`

Init and enable a counter compare value.

- `int MSXE17xx_MFIncCounterDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)`

Disable and Release a counter compare value.

- `int MSXE17xx_MFIncCounterInitHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulCount, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`

Init the hardware trigger configuration.

- `int MSXE17xx_MFIncCounterReleaseHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`

Release the hardware trigger.

- `int MSXE17xx_MFIncCounterInitIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`

Init the index configuration.

- `int MSXE17xx_MFIncCounterReleaseIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)`

Release the index.

- `int MSXE17xx_MFIncCounterInitAndEnableLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)`

Init and enable a counter latch logic.

- int [MSXE17xx__MFIncCounterDisableAndReleaseLatch](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulOption01, struct [MSXE17xx__Response](#) *Response)

Disable and Release a counter latch logic.

- int [MSXE17xx__MFIncCounterInitAndEnableClear](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE17xx__Response](#) *Response)

Init and enable a counter clear logic.

- int [MSXE17xx__MFIncCounterDisableAndReleaseClear](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulOption01, struct [MSXE17xx__Response](#) *Response)

Disable and Release a counter clear logic.

- int [MSXE17xx__MFIncCounterInitAndEnableLatchRegister](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulHardwareTriggerEdgeSelection, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulOption01, struct [MSXE17xx__Response](#) *Response)

Init and enable a counter latch register.

- int [MSXE17xx__MFIncCounterDisableAndReleaseLatchRegister](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulOption01, struct [MSXE17xx__Response](#) *Response)

Disable and Release a counter latch register.

- int [MSXE17xx__MFIncCounterInitAndEnableFrequencyMeasurement](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulTimingInterval, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct [MSXE17xx__Response](#) *Response)

Init and enable a counter frequency measurement.

- int [MSXE17xx__MFIncCounterDisableAndReleaseFrequencyMeasurement](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE17xx__Response](#) *Response)

Disable and Release a counter frequency measurement.

- int [MSXE17xx__MFIncCounterInitAndEnableIndex](#) (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulReferenceAction, xsd__unsignedLong ulIndexOperation, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, struct [MSXE17xx__Response](#) *Response)

Init and enable a counter index.

- int [MSXE17xx__MFIncCounterDisableAndReleaseIndex](#) (xsd__unsignedLong ulMFModuleIndex, struct [MSXE17xx__Response](#) *Response)

Disable and Release a counter index.

4.1.1 Define Documentation

4.1.1.1 #define MSXE170X_COUNTER_QUADRUPLE_MODE 0x4

4.1.1.2 #define MSXE170X_COUNTER_DOUBLE_MODE 0x2

4.1.1.3 #define MSXE170X_COUNTER_SIMPLE_MODE 0x1

4.1.1.4 #define MSXE170X_COUNTER_DIRECT_MODE 0x0

The inputs A and B in 32-Bit mode or A, B and C,D in 16-Bit mode present, each, one clock pulse gate circuit. Thereby frequency and pulse duration measurements can be done.

4.1.1.5 #define MSXE170X_COUNTER_HYSTERESIS_ON 0x1

It suppresses the first counting pulse after a change of rotation.

4.1.1.6 **#define MSXE170X_COUNTER_HYSTERESIS_OFF 0x0**

4.1.1.7 **#define MSXE170X_COUNTER_INCREMENT 0x0**

4.1.1.8 **#define MSXE170X_COUNTER_DECREMENT 0x1**

4.1.1.9 **#define MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_COUNTER 0x0**

4.1.1.10 **#define MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_COUNTER 0x1**

4.1.1.11 **#define MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER 0x2**

4.1.1.12 **#define MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER 0x3**

4.1.2 Typedef Documentation

4.1.2.1 **typedef char* xsd__string**

4.1.2.2 **typedef char xsd__char**

4.1.2.3 **typedef float xsd__float**

4.1.2.4 **typedef double xsd__double**

4.1.2.5 **typedef int xsd__int**

4.1.2.6 **typedef long xsd__long**

4.1.2.7 **typedef unsigned char xsd__unsignedByte**

4.1.2.8 **typedef unsigned int xsd__unsignedInt**

4.1.2.9 **typedef unsigned short int xsd__unsignedShort**

4.1.2.10 **typedef unsigned long xsd__unsignedLong**

4.1.3 Function Documentation

4.1.3.1 **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

4.1.3.2 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

4.1.3.3 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

4.1.3.4 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

4.1.3.5 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) Linux Programmer's Manual
 STRERROR(3)

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

4.1.3.6 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

4.1.3.7 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

4.1.3.8 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.

4.1.3.9 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

4.1.3.10 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_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

4.1.3.11 `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

4.1.3.12 `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

4.1.3.13 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

4.1.3.14 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

4.1.3.15 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__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.3.16 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__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

4.1.3.17 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__Strerror\(\)](#).

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.3.18 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__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.

4.1.3.19 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__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.

4.1.3.20 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__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

4.1.3.21 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__Strerror\(\)](#).
 - ulUpTime : Number of seconds since the last boot of the system.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.22 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__Strerror\(\)](#).
 - bArray : Array of Bytes of the file

- `ulEOF` : End of file flag

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.23 `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

4.1.3.24 `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

4.1.3.25 `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__Strerror\(\)](#). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.26 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

4.1.3.27 `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;
}
*
```

4.1.3.28 `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.

4.1.3.29 `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.

4.1.3.30 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__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

4.1.3.31 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__Strerror\(\)](#).
 - Value The numerical ID of the sub-system "SubsystemName".

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.32 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

4.1.3.33 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

4.1.3.34 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__Sterror\(\)](#).
 - sArray The symbolic name associated with the state numerical ID.

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

4.1.3.35 `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

4.1.3.36 `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

4.1.3.37 `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

4.1.3.38 int MXCommon_SetFilterChannels (struct xsd__base64Binary * ChannelList, struct MXCommon_Response * Response)

Parameters

[in] *ChannelList* Each index of the array represents a channel. A filter can be affected to each channel. If FilterID = 0, no filter is set (the filter is disabled on the corresponding channel). e.g.: ChannelList[0] = FilterID // Set FilterID on channel 0.

[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

4.1.3.39 int MSXE170x_DigitalIOGetNumber (void * _, struct MSXE170x_DigitalIOGetNumberResponse * Response)

4.1.3.40 int MSXE170x_DigitalIOInitPortConfiguration (xsd__unsignedLong ulPort, xsd__unsignedLong ulPortConfiguration, struct MSXE170x_Response * Response)

Parameters

[in] *ulPort* : Index of the digital i/o port (0 to 7)

[in] *ulPortConfiguration* : Define the port configuration

- 0 : input
- 1 : output

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o port selection error
- -3: Port configuration selection error
- -100: Init dig i/o port kernel function error

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

Returns

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

4.1.3.41 int MSXE170x__DigitalIOReadChannelValue (xsd__unsignedLong ulChannel, struct MSXE170x__unsignedLongResponse * Response)

Parameters

[in] **ulChannel** : Index of the digital i/o channel (0 to 15)

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o channel selection error
- -100: Read dig i/o channel value kernel function error

syserrno : system-error code (the value of the libc "errno" code) **ulValue** : i/o channel value:

- 0
- 1

Returns

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

4.1.3.42 int MSXE170x__DigitalIOReadAllChannelsValue (void * __, struct MSXE170x__unsignedLongResponse * Response)

Parameters

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -100: Read dig i/o channel value kernel function error

syserrno : system-error code (the value of the libc "errno" code) **ulValue** : i/o channels value(each bit correspond to one channel)

Returns

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

4.1.3.43 `int MSXE170x__DigitalIOWriteChannelValue (xsd__unsignedLong ulChannel, xsd__unsignedLong ulChannelValue, struct MSXE170x__Response * Response)`

Parameters

- [in] *ulChannel* : Index of the digital i/o channel (0 to 15)
- [in] *ulChannelValue* : Channel value
- 0
 - 1
- [out] *Response* :
- iReturnValue* :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Digital i/o channel selection error
 - -3: Channel value error
 - -100: Write dig i/o channel value kernel function error
- syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.44 `int MSXE170x__DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE170x__Response * Response)`

Parameters

- [in] *ulChannelValue* : Channels value (each bit corresponds to a channel)
- [out] *Response* :
- iReturnValue* :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Channels value error
 - -100: Write dig i/o channel value kernel function error
- syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.45 `int MSXE170x__DigitalIOReleasePortConfiguration (xsd__unsignedLong ulPort, struct MSXE170x__Response * Response)`

Parameters

- [in] *ulPort* : Index of the digital i/o port (0 to 7)

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Digital i/o port selection error

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

Returns

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

4.1.3.46 `int MSXE170x__DigitalIOTestShortCircuit (xsd__unsignedLong ulOption, struct MSXE170x__unsignedLongResponse * Response)`

Parameters

[in] **ulOption** : reserved

[out] **Response** :

iReturnValue :

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

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

ulValue : short circuit status: from 0 to 0xffff, one bit for each output

- 0 : no short circuit
- 1 : short circuit

Returns

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

4.1.3.47 `int MSXE170x__DigitalIORearmShortCircuit (xsd__unsignedLong ulOption, struct MSXE170x__Response * Response)`

Parameters

[in] **ulOption** : reserved

[out] **Response** :

iReturnValue :

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

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

Returns

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

4.1.3.48 `int MSXE170x_IOWatchdogInitAndStart (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct MSXE170x__Response * Response)`

Parameters

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

[in] *ulTimeValue* : Time base of the watchdog delay (0 to 0xFFFF)

[in] *ulOption1* : Reserved

[in] *ulOption2* : Reserved

[out] *Response* :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -2: time base selection error
- -3: time value selection error
- -100: Init and start digital output watchdog kernel function error

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

Returns

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

4.1.3.49 `int MSXE170x_IOWatchdogStopAndRelease (xsd__unsignedLong ulOption, struct MSXE170x__Response * Response)`

Parameters

[in] *ulOption* : reserved

[out] *Response* :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -100: Stop and release digital output watchdog kernel function error

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

Returns

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

4.1.3.50 `int MSXE170x_IOWatchdogGetStatusAndValue (xsd__unsignedLong ulOption, struct MSXE170x_IOWatchdogGetStatusAndValueResponse * Response)`

Parameters

[in] *ulOption* : Reserved

[out] **Response** :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -2: channel selection error
- -100: Get diagnostic information kernel function error

ulStatus : current status information

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

ulValue : current value information (0 to 0xFFFF)

ulInfo : reserved

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

Returns

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

4.1.3.51 `int MSXE170x__MFCommonSetInputsFilter (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulInputAFilterValue, xsd__unsignedLong ulInputBFilterValue, xsd__unsignedLong ulInputCFilterValue, xsd__unsignedLong ulInputDFilterValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE170x__Response * Response)`

Parameters

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

[in] **ulInputAFilterValue** : Filter value for input A (0 to 262143)

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

[in] **ulInputBFilterValue** : Filter value for input B (0 to 262143)

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

[in] **ulInputCFilterValue** : Filter value for input C (0 to 262143)

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

- 262143 : 26,2143 ms

[in] ***ulInputDFilterValue*** : Filter value for input D (0 to 262143)

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

[in] ***ulOption01*** : Set it to 0

[in] ***ulOption02*** : Set it to 0

[in] ***ulOption03*** : Set it to 0

[in] ***ulOption04*** : Set it to 0

[out] ***Response*** :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Input A filter value selection error
- -4: Input B filter value selection error
- -5: Input C filter value selection error
- -6: Input D filter value selection error

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

Returns

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

4.1.3.52 **int** MSXE170x__MFCommonReferenceVoltageActivation (xsd__unsignedLong ***ulMFModuleIndex***, xsd__unsignedLong ***ulActivationFlag***, xsd__unsignedLong ***ulOption01***, xsd__unsignedLong ***ulOption02***, struct MSXE170x__Response * ***Response***)

Parameters

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

[in] ***ulActivationFlag*** :

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

[in] ***ulOption01*** : Set it to 0

[in] ***ulOption02*** : Set it to 0

[out] ***Response*** :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Activation flag selection error

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

Returns

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

4.1.3.53 `int MSXE170x__MFIncCounterGetNumber (void * __, struct MSXE170x__MFIncCounterGetNumberResponse * Response)`

4.1.3.54 `int MSXE170x__MFIncCounterInit (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterMode, xsd__unsignedLong ulCounterOption, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE170x__Response * Response)`

Parameters

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

[in] *ulCounterMode* : Set the counter mode : either

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

[in] *ulCounterOption* : Set the counter option

if in QUADRUPLE/DOUBLE/SIMPLE mode : either

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

if in DIRECT mode :

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

[in] *ulOption01* : Set it to 0

[in] *ulOption02* : Set it to 0

[in] *ulOption03* : Set it to 0

[in] *ulOption04* : Set it to 0

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Counter mode selection error
- -4: Counter option selection error
- -5: Multifunction sub module is not a incremental counter
- -100: Init counter kernel function error

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

Returns

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

4.1.3.55 `int MSXE170x__MFIncCounterRelease (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response * Response)`

Parameters

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

[out] *Response* :

iReturnValue :

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

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

Returns

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

4.1.3.56 `int MSXE170x__MFIncCounterSetFIFO0Level (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulFIFOLevel, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response * Response)`

4.1.3.57 `int MSXE170x__MFIncCounterRead32BitValue (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__unsignedLongTimeStampResponse * Response)`

Parameters

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

[out] *Response* :

iReturnValue :

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

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

ulTimeStampLow : 32 bit low part of time stamp (us) *ulTimeStampHigh* : 32 bit high part of time stamp (s)

Returns

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

4.1.3.58 `int MSXE170x__MFIncCounterWrite32BitValue (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterValue, struct MSXE170x__Response * Response)`

Parameters

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

[in] *ulCounterValue* : Counter value

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Counter value error
- -4: Multifunction sub module is not a incremental counter
- -5: Incremental counter not initialised
- -100: Write counter value kernel function error

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

Returns

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

4.1.3.59 `int MSXE170x__MFIncCounterClear (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response * Response)`

Parameters

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

[out] *Response* :

iReturnValue :

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

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

Returns

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

4.1.3.60 `int MSXE170x__MFIncCounterInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response * Response)`

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
- [in] **ulValue** : compare value (0 to 0xFFFFFFFF included)
- [in] **ulMode** : compare mode
- 0: condition true when counter equals compare value
 - 1: condition true when counter equals a multiple of the compare value
- [in] **ulSynchroTrigger** • 0 : no synchro trigger
- 1 : generates a synchro trigger when condition is true
- [in] **ulOption01** : set it to 0
- [in] **ulOption02** : set it to 0
- [out] **Response** :
- iReturnValue** :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Compare value error
 - -4: Compare mode error
 - -5: Synchro trigger error
 - -6: Multifunction sub module is not a incremental counter
 - -7: Incremental counter not initialised
 - -8: Compare logic already initialised
 - -100: Init and enable counter compare kernel function error
- syserrno** : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.61 `int MSXE170x__MFIncCounterDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response * Response)`

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
- [out] **Response** :
- iReturnValue** :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter

- -4: Incremental counter not initialised
- -5: Compare logic not initialised
- -100: Disable counter compare value kernel function error

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

Returns

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

4.1.3.62 `int MSXE170x__MFIncCounterInitAndEnableLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulHardwareTriggerEdgeSelection, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulOption01, struct MSXE170x__Response * Response)`

Parameters

- [in] *ulMFModuleIndex* : index of the multifunction sub module (0 to 3).
- [in] *ulLatchRegister* : 0: First Latch Register used with trigger input, 1: Second Latch Register used with synchro input
- [in] *ulHardwareTriggerEdgeSelection* : not used for the synchro input
- 01 : rising front (Only if trigger input selected)
 - 10 : falling front (Only if trigger input selected)
 - 11 : Both front (Only if trigger input selected)
- [in] *ulHardwareTriggerCount* : not used for the synchro input
Define the number of trigger events before the action occur
1 : all trigger event start the action
max value : 65535
- [in] *ulOption01* : set it to 0
- [out] *Response* :
- iReturnValue* :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Latch register selection error
 - -4: Trigger edge selection error
 - -5: Trigger count selection error
 - -6: Multifunction sub module is not a incremental counter
 - -7: Incremental counter not initialised
 - -8: Latch logic already initialised
 - -100: Init and enable counter latch register kernel function error
- syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.63 `int MSXE170x__MFIncCounterDisableAndReleaseLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulOption01, struct MSXE170x__Response * Response)`

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
- [in] **ulLatchRegister** : 0: First Lath Register used with 24 V input, 1: Second Lath Register used with synchro input
- [in] **ulOption01** : set it to 0
- [out] **Response** :
- iReturnValue** :
- 0: means the remote function performed OK
 - -1: means an system error occured
 - -2: Multifunction sub module index selection error
 - -3: Latch register selection error
 - -4: Multifunction sub module is not a incremental counter
 - -5: Incremental counter not initialised
 - -6: Latch logic not initialised
 - -100: Disable and release counter latchd register kernel function error
- syserrno** : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.64 `int MSXE170x__MFIncCounterInitAndEnableFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulTimingInterval, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE170x__Response * Response)`

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
- [in] **ulTimingInterval** : Timing interval from 1 to 0xFFFF, one step correspond to 100 ns.
- 1: time interval = 100 ns
 - 2: time interval = 200 ns
 - ...
- [in] **ulOption01** : Set the measurement mode
- 0: One shot, the frequency measure is done at each call of this function
 - 1: Triggered mode, this function has to be called only once (to initialize the measure). The synchro trigger is used to trigger the measure. The synchro trigger cycle time should be > frequency time interval
- [in] **ulOption02** : Timebase selection
- 0: 100ns
 - 1: 100us

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Timing interval selection error
- -4: Multifunction sub module is not a incremental counter
- -5: Incremental counter not initialised
- -6: Frequency measurement already initialised and started
- -7: Frequency measurement ulOption01 parameter wrong value
- -8: Frequency measurement ulOption02 parameter wrong value
- -100: Init and enable counter Frequency measurement kernel function error

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

Returns

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

4.1.3.65 `int MSXE170x__MFIncCounterDisableAndReleaseFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, struct MSXE170x__Response * Response)`

Parameters

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

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

4.1.3.66 `int MSXE170x__MFIncCounterInitAndEnableIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulReferenceAction, xsd__unsignedLong ulIndexOperation, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, struct MSXE170x__Response * Response)`

The index operation can be programmed. If used, the reference signal enable the index management. Rising edge on reference enable the index management. Rising edge on index reset the reference management.

Parameters

- [in] ***ulMFModuleIndex*** : index of the multifunction sub module (0 to 3).
- [in] ***ulReferenceAction*** : Reference action
- 0: do not use the D pin as reference,
 - 1: use the D pin as reference
- [in] ***ulIndexOperation*** : Index operation
- 0 = MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_COUNTER,
 - 1 = MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_COUNTER,
 - 2 = MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER,
 - 3 = MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER,
- [in] ***ulAutoMode*** : AutoMode 0: do not use auto mode (action is done only once), 1: use auto mode (action is done continuously)
- [in] ***ulOption01*** : set it to 0
- [out] ***Response*** :
- iReturnValue*** :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Reference action selection error
 - -4: Index operation selection error
 - -5: Auto mode selection error
 - -6: Multifunction sub module is not a incremental counter
 - -7: Incremental counter not initialised
 - -8: Index logic already initialised
 - -100: Init and enable counter index kernel function error
- syserrno*** : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.67 `int MSXE170x_MFIncCounterDisableAndReleaseIndex (xsd__unsignedLong
ulMFModuleIndex, struct MSXE170x__Response * Response)`

Parameters

- [in] ***ulMFModuleIndex*** : index of the multifunction sub module (0 to 3).
- [in] ***ulOption01*** : set it to 0
- [out] ***Response*** :
- iReturnValue*** :
- 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Multifunction sub module is not a incremental counter
 - -4: Incremental counter not initialised

- -5: Index logic not initialised
- -100: Disable and release counter index kernel function error

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

Returns

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

4.1.3.68 int MSXE17xx__DigitalIOGetNumber (void * __, struct MSXE17xx__DigitalIOGetNumberResponse * Response)

Parameters

[in] *None*

[out] **Response** :

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred (check errno in this case)

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

Returns

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

4.1.3.69 int MSXE17xx__DigitalIOInitPortConfiguration (xsd__unsignedLong ulPort, xsd__unsignedLong ulPortConfiguration, struct MSXE17xx__Response * Response)

Parameters

[in] **ulPort** : Index of the digital i/o port (0 to 7)

[in] **ulPortConfiguration** : Define the port configuration

- 0 : input
- 1 : output

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o port selection error
- -3: Port configuration selection error
- -100: Init dig i/o port kernel function error

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

Returns

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

4.1.3.70 `int MSXE17xx_DigitalIOReadChannelValue (xsd__unsignedLong ulChannel, struct MSXE17xx__unsignedLongResponse * Response)`

Parameters

[in] *ulChannel* : Index of the digital i/o channel (0 to 15)

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o channel selection error
- -100: Read dig i/o channel value kernel function error

syserrno : system-error code (the value of the libc "errno" code) *ulValue* : i/o channel value:

- 0
- 1

Returns

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

4.1.3.71 `int MSXE17xx_DigitalIOReadAllChannelsValue (void * _, struct MSXE17xx__unsignedLongResponse * Response)`

Parameters

[in] *_* : no input parameter

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -100: Read dig i/o channel value kernel function error

syserrno : system-error code (the value of the libc "errno" code) *ulValue* : i/o channels value(each bit correspond to one channel)

Returns

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

4.1.3.72 `int MSXE17xx_DigitalIOWriteChannelValue (xsd__unsignedLong ulChannel, xsd__unsignedLong ulChannelValue, struct MSXE17xx__Response * Response)`

Parameters

[in] *ulChannel* : Index of the digital i/o channel (0 to 15)

[in] *ulChannelValue* : Channel value

- 0
- 1

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o channel selection error
- -3: Channel value error
- -100: Write dig i/o channel value kernel function error

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

Returns

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

4.1.3.73 int MSXE17xx_DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE17xx_Response * Response)

Parameters

[in] **ulChannelValue** : Channels value (each bit corresponds to a channel)

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Channels value error
- -100: Write dig i/o channel value kernel function error

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

Returns

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

4.1.3.74 int MSXE17xx_DigitalIOReleasePortConfiguration (xsd__unsignedLong ulPort, struct MSXE17xx_Response * Response)

Parameters

[in] **ulPort** : Index of the digital i/o port (0 to 7)

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Digital i/o port selection error

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

Returns

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

4.1.3.75 `int MSXE17xx_DigitalIOTestShortCircuit (xsd__unsignedLong ulOption, struct MSXE17xx__unsignedLongResponse * Response)`

Parameters

[in] *ulOption* : reserved

[out] *Response* :

iReturnValue :

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

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

ulValue : short circuit status: from 0 to 0xffff, one bit for each output

- 0 : no short circuit
- 1 : short circuit

Returns

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

4.1.3.76 `int MSXE17xx_DigitalIORearmShortCircuit (xsd__unsignedLong ulOption, struct MSXE17xx__Response * Response)`

Parameters

[in] *ulOption* : reserved

[out] *Response* :

iReturnValue :

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

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

Returns

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

4.1.3.77 `int MSXE17xx_IOWatchdogInitAndStart (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulTimeValue* : Time base of the watchdog delay (0 to 0xFFFF)

[in] *ulOption1* : Reserved

[in] *ulOption2* : Reserved

[out] *Response* :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -2: time base selection error
- -3: time value selection error
- -100: Init and start digital output watchdog kernel function error

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

Returns

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

4.1.3.78 int MSXE17xx_IOWatchdogStopAndRelease (xsd__unsignedLong *ulOption*, struct MSXE17xx__Response * *Response*)

Parameters

[in] *ulOption* : reserved

[out] *Response* :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occurred
- -100: Stop and release digital output watchdog kernel function error

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

Returns

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

4.1.3.79 int MSXE17xx_IOWatchdogGetStatusAndValue (xsd__unsignedLong *ulOption*, struct MSXE17xx__IOWatchdogGetStatusAndValueResponse * *Response*)

Parameters

[in] *ulOption* : Reserved

[out] *Response* :

iReturnValue :

- 0: remote function performed OK
 - -1: an system error occurred
 - -2: channel selection error
 - -100: Get diagnostic information kernel function error
- ulStatus* : current status information
- BIN XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXX0: is stopped,
 - BIN XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXX1: is running,
 - BIN XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXX0X: is not run down

- BIN XXXXXXXXXX XXXXXXXXXX XXXXXXXXXX XXXXXXXX1X: is run down
ulValue : current value information (0 to 0xFFFF)
ulInfo : reserved
syserrno : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.80 int MSXE17xx__MFCommonGetSubModuleFunctionality (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__unsignedLongResponse * Response)

Parameters

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

[out] *Response* :

ulValue :

- 0: Incremental counter
- -1: PWM

sResponse.iReturn Value :

- 0: means the remote function performed OK
- -1: means an system error occured (check errno in this case)
- -2: Multifunction sub module index selection error

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

Returns

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

4.1.3.81 int MSXE17xx__MFCommonSetInputsFilter (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulInputAFilterValue, xsd__unsignedLong ulInputBFilterValue, xsd__unsignedLong ulInputCFilterValue, xsd__unsignedLong ulInputDFilterValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__Response * Response)

Parameters

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

[in] *ulInputAFilterValue* : Filter value for input A (0 to 262143)

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

[in] *ulInputBFilterValue* : Filter value for input B (0 to 262143)

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

[in] ***ulInputCFilterValue*** : Filter value for input C (0 to 262143)

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

[in] ***ulInputDFilterValue*** : Filter value for input D (0 to 262143)

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

[in] ***ulOption01*** : Set it to 0

[in] ***ulOption02*** : Set it to 0

[in] ***ulOption03*** : Set it to 0

[in] ***ulOption04*** : Set it to 0

[out] ***Response*** :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Input A filter value selection error
- -4: Input B filter value selection error
- -5: Input C filter value selection error
- -6: Input D filter value selection error

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

Returns

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

4.1.3.82 **int** MSXE17xx__MFCommonReferenceVoltageActivation (xsd_unsignedLong *ulMFModuleIndex*, xsd_unsignedLong *ulActivationFlag*, xsd_unsignedLong *ulOption01*, xsd_unsignedLong *ulOption02*, struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulActivationFlag*** :

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

[in] *ulOption01* : Set it to 0

[in] *ulOption02* : Set it to 0

[out] *Response* :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Activation flag selection error

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

Returns

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

4.1.3.83 `int MSXE17xx__MFCommonEnableDisableTriggerGate (xsd__unsignedLong ulTriggerConfiguration, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

[in] *ulTriggerConfiguration* : Trigger gate configuration:

Bit 0, Hardware trigger gate :

- 0 : Hardware trigger gate is disabled
- 1 : Hardware trigger gate is enabled

[in] *ulOption01* : Set it to 0

[in] *ulOption02* : Set it to 0

[out] *Response* :

iReturnValue :

- 0 : means the remote function performed OK
- -1: means an system error occurred
- -2: ulTriggerConfiguration parameter is wrong
- -100: MSXE17xx__MFCommonEnableDisableTriggerGate kernel function error

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

Returns

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

4.1.3.84 `int MSXE17xx__MFCommonSetFIFO0Level (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulFIFOLevel, xsd__unsignedLong ulTimeOutTimeBase, xsd__unsignedLong ulReloadValue, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

- [in] *ulMFModuleIndex* : index of the multifunction sub module (0 to 3).
- [in] *ulFIFOLevel* : Define the FIFO level (1 to 200).
- [in] *ulTimeOutTimeBase* : Define a Time out : permit to receive the data from the FIFO before the FIFO level is reached.
Time base of the timer (0: disabled, 1 for us, 2 for ms, 3 for s)
- [in] *ulReloadValue* : Time out reload value (1 to 0xFFFF)
- [in] *ulOption01* : reserved (Set it to 0).
- [in] *ulOption02* : reserved (Set it to 0).
- [out] *Response* :
iReturnValue :
 - 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: FIFO level value is wrong
 - -4: Time out time base selection error
 - -5: Time out value can not be null, if a time base is selected
 - -100: Set FIFO level kernel function error*syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.85 `int MSXE17xx__MFPWMGetNumber (void * _, struct MSXE17xx__unsignedLongResponse * Response)`

Parameters

- [in] *None*
- [out] *Response* :
ulValue : number of available PWM sub modules

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred (check errno in this case)

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

Returns

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

4.1.3.86 `int MSXE17xx_MFPWMInitAndEnable (xsd_unsignedLong ulMFModuleIndex, xsd_unsignedLong ulPWM, xsd_unsignedLong ulTimeBase, xsd_unsignedLong ulLowTime, xsd_unsignedLong ulHighTime, xsd_unsignedLong ulStartLevel, xsd_unsignedLong ulExternGate, xsd_unsignedLong ulStopMode, xsd_unsignedLong ulStopLevel, xsd_unsignedLong ulSynchroOut, xsd_unsignedLong ulDataFrame, xsd_unsignedLong ulOption01, xsd_unsignedLong ulOption02, xsd_unsignedLong ulOption03, xsd_unsignedLong ulOption04, struct MSXE17xx_Response * Response)`

Parameters

- [in] **ulMFModuleIndex** : Index of the multifunction sub module (0 to 3).
- [in] **ulPWM** : Selected PWM 0 or 1
- [in] **ulTimeBase** : PWM time base selection
 - 0: 250ns (4MHz)
 - 1: micro s (1MHz)
 - 2: ms (1kHz)
 - 3: s (Hz)
 - 4: Diff input C for PWM0 and D for PWM1
 - 5: Trigger input (24V)
 - 6: Synchro trigger
- [in] **ulLowTime** : PWM low time reload value (1 to 16777215)
 - If the extern clock is used this define the number of clocks for the low level.
 - If the internal time base is used this define the low level time : Low time = ulTimeBase * ulLowTime.
- [in] **ulHighTime** : PWM high time reload value (1 to 16777215)
 - If the extern clock is used this define the number of clocks for the high level.
 - If the internal time base is used this define the high level time : High time = ulTimeBase * ulHighTime.
- [in] **ulStartLevel** : PWM start level, time selection
 - 0: PWM start with the low level
 - 1: PWM start with the high level
- [in] **ulExternGate** : Extern gate selection
 - 0: Extern gate not used
 - 1: Diff input C (PWM0) or D (PWM1) is used as gate
 - 2: Input trigger (24V) is used as gate
- [in] **ulStopMode** : Stop mode selection
 - 0: Stop the PWM signal directly after a stop condition
 - 1: Stop the PWM after the current period is stopped
- [in] **ulStopLevel** : Stop level selection
 - 0: The output signal keep the level after the stop condition
 - 1: The output signal is set to low after the stop condition
 - 2: The output signal is set to high after the stop condition
- [in] **ulSynchroOut** : Give the possibility to generate a synchro trigger
 - 0: Disabled
 - 1: Generate a synchro trigger be each start of period
 - 2: Generate a synchro trigger be each end of period

- 3: Generate a synchro trigger be each start and end of period

[in] **ulDataFrame** : Give the possibility to send via the data server PWM informations

- 0: Disabled
- 1: Enabled

For each start and end of period the data server send a 5 DWORD frame with following informations:

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (1)
DWORD 3 : Event mask
    8: PWM0. First period begin
    9: PWM0. End of period
    10: PWM0. End of last period. Stop condition occur
    11: PWM1. First period begin
    12: PWM1. End of period
    13: PWM1. End of last period. Stop condition occur
DWORD 4 :
    D23-D0: Number of occurred periods
    D24: 0: Diff input C for PWM0 or D for PWM1 is low
          1: Diff input C for PWM0 or D for PWM1 is high
    D25: 0: Extern trigger input (24V) is low
          1: Extern trigger input (24V) is high
    D26: 0: PWM output is low
          1: PWM output is high
```

[in] **ulOption01** : Reserved.

- Set to 0

[in] **ulOption02** : Reserved.

- Set to 0

[in] **ulOption03** : Reserved.

- Set to 0

[in] **ulOption04** : Reserved.

- Set to 0

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -5: Time base selection wrong
- -6: Low time selection wrong
- -7: High time selection wrong
- -8: PWM already in progress
- -9: PWM timing not initialised
- -10: Start level selection wrong
- -11: Extern gate selection wrong
- -12: Extern gate selection identical to the external clock selection
- -13: Stop mode selection wrong
- -14: Stop level selection wrong
- -15: Synchro out trigger selection wrong

- -16: Module is slave and can not generate the syncho out trigger
- -17: Data frame selection wrong
- -18: Can not mixed synchro trigger clock and module synchronisation (ulSynchroOut)
- -100: Init and enable PWM kernel function error

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

Returns

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

4.1.3.87 `int MSXE17xx_MFPWMSetNewTiming (xsd_unsignedLong ulMFModuleIndex, xsd_unsignedLong ulPWM, xsd_unsignedLong ulTimeBase, xsd_unsignedLong ulLowTime, xsd_unsignedLong ulHighTime, xsd_unsignedLong ulOption01, xsd_unsignedLong ulOption02, struct MSXE17xx_Response * Response)`

Parameters

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

[in] **ulPWM** : Selected PWM 0 or 1

[in] **ulTimeBase** : PWM time base selection

- 0: 250ns (4MHz)
- 1: micro s (1MHz)
- 2: ms (1kHz)
- 3: s (Hz)
- 4: Diff input C for PWM0 and D for PWM1
- 5: Trigger input (24V)
- 6: Synchro trigger

[in] **ulLowTime** : PWM low time reload value (1 to 16777215)

If the extern clock is used this define the number of clocks for the low level.

If the internal time base is used this define the low level time : Low time = ulTimeBase * ulLowTime.

[in] **ulHighTime** : PWM high time reload value (1 to 16777215)

If the extern clock is used this define the number of clocks for the high level.

If the internal time base is used this define the high level time : High time = ulTimeBase * ulHighTime.

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

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -5: Time base selection wrong
- -6: Low time selection wrong

- -7: High time selection wrong
- -8: Can not change the current time base
- -9: PWM not initialised
- -10: PWM not enabled
- -100: Set new PWM timing kernel function error

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

Returns

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

4.1.3.88 `int MSXE17xx__MFPWMDisable (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

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

[in] **ulPWM** : Selected PWM 0 or 1

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

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -5: PWM not initialised
- -6: PWM already disabled
- -100: Disable PWM kernel function error

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

Returns

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

4.1.3.89 `int MSXE17xx__MFPWMDisableAndRelease (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

If PWM in progress this stop directly the PWM and ignore the ulStopMode configuration (MSXE17xx__MFPWMInitAndEnable).

Parameters

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

[in] **ulPWM** : Selected PWM 0 or 1

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

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

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -5: PWM not initialised
- -100: Disable PWM kernel function error

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

Returns

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

4.1.3.90 **int** MSXE17xx__MFPWMGetConfiguration (xsd__unsignedLong **ulMFModuleIndex**, xsd__unsignedLong **ulPWM**, struct MSXE17xx__MFPWMGetConfigurationResponse * **Response**)

Parameters

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

[in] **ulPWM** : Selected PWM 0 or 1

[out] **Response** :

ulInitialised Return the initialisation flag

- 0: PWN not initialised
- 1: PWM initialised

ulTimeBase PWM time base selection

- 0: 250ns (4MHz)
- 1: micro s (1MHz)
- 2: ms (1kHz)
- 3: s (Hz)
- 4: Diff input C for PWM0 and D for PWM1
- 5: Trigger input (24V)
- 6: Synchro trigger

ulLowTime PWM low time reload value (1 to 16777215)

- If the extern clock is used this define the number of clocks for the low level.
- If the internal time base is used this define the low level time : Low time = ulTimeBase * ulLowTime.

ulHighTime PWM high time reload value (1 to 16777215)

- If the extern clock is used this define the number of clocks for the high level.

- If the internal time base is used this define the high level time : High time = ulTimeBase * ulHighTime.

ulEnabled Return the enabled status

- 0: PWM disabled
- 1: PWN enabled

ulStartLevel Return the start level, time

- 0: PWM start with the low level
- 1: PWM start with the high level

ulExternGate Return the extern gate initialisation

- 0: Extern gate not used
- 1: Diff input C (PWM0) or D (PWM1) is used as gate
- 2: Input trigger (24V) is used as gate

ulStopMode Return the current stop mode

- 0: Stop the PWM signal directly after a stop condition
- 1: Stop the PWM after the current period is stopped

ulStopLevel Return the current selected stop level

- 0: The output signal keep the level after the stop condition
- 1: The output signal is set to low after the stop condition
- 2: The output signal is set to high after the stop condition

ulSynchroOut Give the possibility to generate a synchro trigger

- 0: Disabled
- 1: Generate a synchro trigger be each start of period
- 2: Generate a synchro trigger be each end of period
- 3: Generate a synchro trigger be each start and end of period

ulDataFrame Give the possibility to send via the data server PWM informations

- 0: Disabled
- 1: Enabled

For each start and end of period the data server send a 5 DWORD frame with following informations:

```

DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (1)
DWORD 3 : Event mask
    8: PWM0. First period begin
    9: PWM0. End of period
    10: PWM0. End of last period. Stop condition occur
    11: PWM1. First period begin
    12: PWM1. End of period
    13: PWM1. End of last period. Stop condition occur
DWORD 4 :
    D23-D0: Number of occurred periods
    D24: 0: Diff input C for PWM0 or D for PWM1 is low
         1: Diff input C for PWM0 or D for PWM1 is high
    D25: 0: Extern trigger input (24V) is low
         1: Extern trigger input (24V) is high
    D26: 0: PWM output is low
         1: PWM output is high

```

ulInfo01 :

- Reserved. Return 0

ulInfo02 :

- Reserved. Return 0

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -100: Get PWM configuration kernel function error

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

Returns

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

4.1.3.91 `int MSXE17xx__MFPWMGetState (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulPWM, struct MSXE17xx__MFPWMGetStateResponse * Response)`

Parameters

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

[in] ***ulPWM*** : Selected PWM 0 or 1

[out] ***Response*** :

ulInitialised : Return the initialisation flag

- 0: PWN not initialised
- 1: PWM initialised

ulEnabled : Return the enabled status

- 0: PWM disabled
- 1: PWN enabled

ulProgress : Return the progress status

- 0: PWM not in progress
- 1: PWM in progress

ulOutput : Return the PWM output signal state

- 0: PWM output is low
- 1: PWM output is high

ulDiffInput : Return the diff input state

- 0: The diff input C (for PWM0) or F (for PWM1) is low
- 1: The diff input C (for PWM0) or F (for PWM1) is high

ulTriggerInput : Return the extern trigger state

- 0: The trigger input (24V) is low
- 1: The trigger input (24V) is high

ulCounter :

- Return the number of completed PWM periods

ulInfo01 :

- Reserved. Return 0

ulInfo02 :

- Reserved. Return 0

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Sub module selection wrong
- -3: Sub module is not a PWM
- -4: PWM selection wrong
- -100: Get PWM state kernel function error

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

Returns

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

4.1.3.92 `int MSXE17xx__MFIncCounterGetNumber (void * __, struct MSXE17xx__MFIncCounterGetNumberResponse * Response)`

Parameters

[in] *None*

[out] *Response :*

sResponse.iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred (check errno in this case)

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

Returns

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

4.1.3.93 `int MSXE17xx__MFIncCounterInit (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterMode, xsd__unsignedLong ulCounterOption, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulCounterMode* : Set the counter mode : either

- MSXE170X_COUNTER_QUADRUPLE_MODE (0x4)
- MSXE170X_COUNTER_DOUBLE_MODE (0x2)

- MSXE170X_COUNTER_SIMPLE_MODE (0x1)
- MSXE170X_COUNTER_DIRECT_MODE (0x0)

[in] **ulCounterOption** : Set the counter option
 if in QUADRUPLE/DOUBLE/SIMPLE mode : either

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

if in DIRECT mode :

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

[in] **ulOption01** : Set it to 0
 [in] **ulOption02** : Set it to 0
 [in] **ulOption03** : Set it to 0
 [in] **ulOption04** : Set it to 0
 [out] **Response** :
iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Counter mode selection error
- -4: Counter option selection error
- -5: Multifunction sub module is not a incremental counter
- -100: Init counter kernel function error

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

Returns

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

4.1.3.94 `int MSXE17xx_MFIncCounterRelease (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx_Response * Response)`

Parameters

[in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
 [out] **Response** :
iReturnValue :

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

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

Returns

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

4.1.3.95 `int MSXE17xx__MFIncCounterRead32BitValue (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__unsignedLongTimeStampResponse * Response)`

Parameters

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

[out] *Response* :

iReturnValue :

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

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

ulTimeStampLow : 32 bit low part of time stamp (us) *ulTimeStampHigh* : 32 bit high part of time stamp (s)

Returns

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

4.1.3.96 `int MSXE17xx__MFIncCounterWrite32BitValue (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulCounterValue, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulCounterValue* : Counter value

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Counter value error
- -4: Multifunction sub module is not a incremental counter
- -5: Incremental counter not initialised
- -100: Write counter value kernel function error

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

Returns

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

4.1.3.97 `int MSXE17xx__MFIncCounterClear (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response * Response)`

Parameters

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

[out] *Response* :

iReturnValue :

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

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

Returns

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

4.1.3.98 `int MSXE17xx__MFIncCounterInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

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

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (0)
DWORD 3 : Event mask
    D30-D0 :
        0: Compare
        1: Frequency measurement
        2: Hardware trigger latch occur
        3: Synchro input latch occur
        4: Index input latch occur
    D31 :
        0: No error occur
        1: Amplitude or Frequency error occur.
DWORD 4 :
    D31-D0: Counter value (DWORD)
```

Parameters

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

[in] *ulValue* : compare value (0 to 0xFFFFFFFF included)

[in] *ulMode* : compare mode

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

[in] *ulSynchroTrigger* • 0 : no synchro trigger

- 1 : generates a synchro trigger when condition is true

[in] *ulOption01* : set it to 0

[in] *ulOption02* : set it to 0

[out] *Response* :

iReturnValue :

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

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

Returns

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

4.1.3.99 `int MSXE17xx__MFIncCounterDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response * Response)`

Parameters

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

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -5: Compare logic not initialised
- -100: Disable counter compare value kernel function error

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

Returns

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

4.1.3.100 `int MSXE17xx__MFIncCounterInitHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulCount, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulEdgeSelection* : Front selection

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

[in] *ulCount* : Define the number of trigger events before the action occur

1 : all trigger event start the action

max value : 65535

[in] *ulOption01* : set it to 0

[in] *ulOption02* : set it to 0

[out] *Response* :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Trigger edge selection error
- -5: Trigger count selection error
- -6: Incremental counter not initialised
- -7: Hardware trigger already initialised
- -100: Init hardware trigger kernel function error

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

Returns

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

4.1.3.101 `int MSXE17xx__MFIncCounterReleaseHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response * Response)`

Parameters

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

[in] *ulOption01* : set it to 0

[out] *Response* :

iReturnValue :

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

- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -5: Hardware trigger not initialised
- -6: Hardware trigger used and can not released
- -100: Release hardware trigger kernel function error

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

Returns

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

4.1.3.102 `int MSXE17xx__MFIncCounterInitIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response * Response)`

Parameters

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

[in] **ulEdgeSelection** : Front selection

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

[in] **ulOption01** : set it to 0

[in] **ulOption02** : set it to 0

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Index edge selection error
- -5: Incremental counter not initialised
- -6: Index already initialised
- -100: Init index kernel function error

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

Returns

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

4.1.3.103 int MSXE17xx_MFIncCounterReleaseIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx_Response * Response)

Parameters

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

[in] *ulOption01* : set it to 0

[out] *Response* :

iReturn Value :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a incremental counter
- -4: Incremental counter not initialised
- -5: Index not initialised
- -6: Index used and can not released
- -100: Release Index kernel function error

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

Returns

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

4.1.3.104 int MSXE17xx_MFIncCounterInitAndEnableLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx_Response * Response)

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

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 :
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (0)
DWORD 3 : Event mask
    D30-D0 :
        0: Compare
        1: Frequency measurement
        2: Hardware trigger latch occur
        3: Synchro input latch occur
        4: Index input latch occur
    D31 :
        0: No error occur
        1: Amplitude or Frequency error occur.
DWORD 4 :
    D31-D0: Counter value (DWORD)
```

Parameters

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

[in] *ulLatchSource* : Latch source.

- 0: Index input
- 1: Hardware trigger

- 2: Synchro input

[in] **ulCondition** : Previously condition for accept the latch source

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

[in] **ulAutoMode** : Action mode

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

[in] **ulOption01** : set it to 0

[in] **ulOption02** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

4.1.3.105 `int MSXE17xx__MFIncCounterDisableAndReleaseLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response * Response)`

Parameters

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

[in] **ulLatchSource** : Latch source to disable and release.

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

4.1.3.106 `int MSXE17xx_MFIncCounterInitAndEnableClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx_Response * Response)`

Parameters

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

[in] **ulClearSource** : Clear source.

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

[in] **ulCondition** : Previously condition for accept the clear source

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

[in] **ulAutoMode** : Action mode

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

[in] **ulOption01** : set it to 0

[in] **ulOption02** : set it to 0

[out] **Response** :

iReturnValue :

- 0: means the remote function performed OK

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

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

Returns

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

4.1.3.107 `int MSXE17xx__MFIncCounterDisableAndReleaseClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response * Response)`

Parameters

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

[in] **ulClearSource** : Clear source to disable and release.

[in] **ulOption01** : set it to 0

[out] **Response** :

iReturnValue :

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

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

Returns

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

4.1.3.108 `int MSXE17xx__MFIncCounterInitAndEnableLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulHardwareTriggerEdgeSelection, xsd__unsignedLong ulHardwareTriggerCount, xsd__unsignedLong ulOption01, struct MSXE17xx__Response * Response)`

It is recommended to use the "MSXE17xx__MFIncCounterInitAndEnableLatch" and "MSXE17xx__MFIncCounterInitAndEnableClear" functions

Parameters

- [in] **ulMFModuleIndex** : index of the multifunction sub module (0 to 3).
 - [in] **ulLatchRegister** : 0: First Latch Register used with trigger input, 1: Second Lath Register used with synchro input
 - [in] **ulHardwareTriggerEdgeSelection** : not used for the synchro input
 - 01 : rising front (Only if trigger input selected)
 - 10 : falling front (Only if trigger input selected)
 - 11 : Both front (Only if trigger input selected)
 - [in] **ulHardwareTriggerCount** : not used for the synchro input
 Define the number of trigger events before the action occur
 1 : all trigger event start the action
 max value : 65535
 - [in] **ulOption01** : set it to 0
 - [out] **Response** :
iReturnValue :
 - 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Latch register selection error
 - -4: Trigger edge selection error
 - -5: Trigger count selection error
 - -6: Multifunction sub module is not a incremental counter
 - -7: Incremental counter not initialised
 - -8: Latch logic already initialised
 - -100: Init and enable counter latch register kernel function error
- syserrno** : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.109 `int MSXE17xx__MFIncCounterDisableAndReleaseLatchRegister (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchRegister, xsd__unsignedLong ulOption01, struct MSXE17xx__Response * Response)`

Parameters

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

[in] ***ulLatchRegister*** : 0: First Lath Register used with 24 V input, 1: Second Lath Register used with synchro input

[in] ***ulOption01*** : set it to 0

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

4.1.3.110 int MSXE17xx_MFIncCounterInitAndEnableFrequencyMeasurement (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulTimingInterval, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx_Response * Response)

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

```
DWORD 0 : Time stamp micro s
DWORD 1 : Time stamp s
DWORD 2 : Counter index
    D1-D0 : Sub module index (0 to 3)
    D31-D16 : Sub module functionality (0)
DWORD 3 : Event mask
    0: Compare
    1: Frequency measurement
    2: Hardware trigger latch occur
    3: Synchro input latch occur
    4: Counter index
DWORD 4 :
    D31-D0: Counter value (DWORD)
```

Parameters

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

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

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

[in] ***ulOption01*** : Set the measurement mode

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

[in] ***ulOption02*** : Timebase selection

- 0: 100ns
- 1: 100us

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Multifunction sub module index selection error
- -3: Timing interval selection error
- -4: Multifunction sub module is not a incremental counter
- -5: Incremental counter not initialised
- -6: Frequency measurement already initialised and started
- -7: Frequency measurement ulOption01 parameter wrong value
- -8: Frequency measurement ulOption02 parameter wrong value
- -100: Init and enable counter Frequency measurement kernel function error

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

Returns

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

4.1.3.111 int MSXE17xx_MFIncCounterDisableAndReleaseFrequencyMeasurement (xsd_unsignedLong *ulMFModuleIndex*, struct MSXE17xx_Response * *Response*)

Parameters

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

[in] ***ulOption01*** : set it to 0

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

4.1.3.112 `int MSXE17xx__MFIncCounterInitAndEnableIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulReferenceAction, xsd__unsignedLong ulIndexOperation, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, struct MSXE17xx__Response * Response)`

It is recommended to use the "MSXE17xx__MFIncCounterInitAndEnableLatch" and "MSXE17xx__MFIncCounterInitAndEnableClear" functions

The index operation can be programmed. If used, the reference signal enable the index management. Rising edge on reference enable the index management. Rising edge on index reset the reference management.

Parameters

- [in] *ulMFModuleIndex* : index of the multifunction sub module (0 to 3).
- [in] *ulReferenceAction* : Reference action
 - 0: do not use the D pin as reference,
 - 1: use the D pin as reference
- [in] *ulIndexOperation* : Index operation
 - 0 = MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_COUNTER,
 - 1 = MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_COUNTER,
 - 2 = MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER,
 - 3 = MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER,
- [in] *ulAutoMode* : AutoMode 0: do not use auto mode (action is done only once), 1: use auto mode (action is done continuously)
- [in] *ulOption01* : set it to 0
- [out] *Response* :
 - iReturnValue* :
 - 0: means the remote function performed OK
 - -1: means an system error occurred
 - -2: Multifunction sub module index selection error
 - -3: Reference action selection error
 - -4: Index operation selection error
 - -5: Auto mode selection error
 - -6: Multifunction sub module is not a incremental counter
 - -7: Incremental counter not initialised
 - -8: Index logic already initialised
 - -100: Init and enable counter index kernel function error
 - syserrno* : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.113 `int MSXE17xx__MFIncCounterDisableAndReleaseIndex (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response * Response)`

Parameters

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

[in] ***ulOption01*** : set it to 0

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

Index

- __offset
 - ByteArray, 85
 - UnsignedLongArray, 105
 - UnsignedShortArray, 106
 - __ptr
 - ByteArray, 85
 - UnsignedLongArray, 105
 - UnsignedShortArray, 106
 - xsd__base64Binary, 106
 - __size
 - ByteArray, 85
 - UnsignedLongArray, 105
 - UnsignedShortArray, 106
 - xsd__base64Binary, 106
- Analog
 - MXCommon__SetFilterChannels, 30
- bArray
 - MXCommon__-
 - GetAutoConfigurationFileResponse, 99
- bCryptedValueArray
 - MXCommon__TestCustomerIDResponse, 104
- bValueArray
 - MXCommon__TestCustomerIDResponse, 104
- ByteArray, 85
 - __offset, 85
 - __ptr, 85
 - __size, 85
- Common functions, 3
- Common general functions, 4
- Common hardware trigger functions, 12
- Common I/O auto configuration functions, 18
- Common security functions, 14
- Common synchronisation timer functions, 20
- Common temperature functions, 10
- Common time functions, 16
- 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, 8
 - MXCommon__ResetAllIOFunctionalities, 8
 - MXCommon__SetHostname, 6
 - MXCommon__Strerror, 6
- Common_hardware_trigger
 - MXCommon__-
 - GetHardwareTriggerFilterTime, 13
 - MXCommon__GetHardwareTriggerState, 13
 - MXCommon__-
 - SetHardwareTriggerFilterTime, 12
- Common_security
 - MXCommon__SetCustomerKey, 15
 - MXCommon__TestCustomerID, 15
- Common_synchrotimer
 - MXCommon__InitAndStartSynchroTimer, 21
 - MXCommon__-
 - StopAndReleaseSynchroTimer, 22
- Common_temperature
 - MXCommon__-
 - GetModuleTemperatureValueAndStatus, 11
 - MXCommon__-
 - SetModuleTemperatureWarningLevels, 11
- Common_time
 - MXCommon__GetTime, 17
 - MXCommon__GetUpTime, 18
 - MXCommon__HardwareClockToSys, 17
 - MXCommon__SetTime, 16
 - MXCommon__SysToHardwareClock, 17
- Customer option management, 27
- CustomerOption
 - MXCommon__GetOptionInformation, 28
- DefaultResponse, 85
 - iReturnValue, 86

- syserrno, [86](#)
- dTemperatureValue
 - MXCommon__-
 - GetModuleTemperatureValueAndStatusResponse, [102](#)
- input filter Filter management, [29](#)
- iReturnValue
 - DefaultResponse, [86](#)
 - MSXE170x__Response, [88](#)
 - MSXE17xx__Response, [96](#)
 - MXCommon__Response, [103](#)
- MSX-E170x compatibility functions, [3](#)
- MSX-E170x digital I/O functions, [30](#)
- MSX-E170x incremental counter functions, [38](#)
- MSX-E170x IO watchdog functions, [34](#)
- MSX-E170x multifunction common functions, [36](#)
- MSX-E17xx digital I/O functions, [48](#)
- MSX-E17xx functions, [3](#)
- MSX-E17xx incremental counter functions, [67](#)
- MSX-E17xx IO watchdog functions, [53](#)
- MSX-E17xx multifunction common functions, [55](#)
- MSX-E17xx multifunction functions, [3](#)
- MSX-E17xx pulse width modulation, [59](#)
- MSXE170x__DigitalIOGetNumber
 - MSXE170x_DigIO, [31](#)
 - MSXE170x_public_doc.h, [139](#)
- MSXE170x__DigitalIOGetNumberResponse, [86](#)
 - sResponse, [87](#)
 - ulNumberOfDigitalIO, [87](#)
- MSXE170x__DigitalIOInitPortConfiguration
 - MSXE170x_DigIO, [31](#)
 - MSXE170x_public_doc.h, [139](#)
- MSXE170x__DigitalIOReadAllChannelsValue
 - MSXE170x_DigIO, [32](#)
 - MSXE170x_public_doc.h, [140](#)
- MSXE170x__DigitalIOReadChannelValue
 - MSXE170x_DigIO, [31](#)
 - MSXE170x_public_doc.h, [140](#)
- MSXE170x__DigitalIORearmShortCircuit
 - MSXE170x_DigIO, [34](#)
 - MSXE170x_public_doc.h, [142](#)
- MSXE170x__DigitalIOReleasePortConfiguration
 - MSXE170x_DigIO, [33](#)
 - MSXE170x_public_doc.h, [141](#)
- MSXE170x__DigitalIOTestShortCircuit
 - MSXE170x_DigIO, [33](#)
 - MSXE170x_public_doc.h, [142](#)
- MSXE170x__DigitalIOWriteAllChannelsValue
 - MSXE170x_DigIO, [33](#)
 - MSXE170x_public_doc.h, [141](#)
- MSXE170x__DigitalIOWriteChannelValue
 - MSXE170x_DigIO, [32](#)
- MSXE170x_public_doc.h, [140](#)
- MSXE170x__IOWatchdogGetStatusAndValue
 - MSXE170x_public_doc.h, [143](#)
- MSXE170x__Watchdog, [36](#)
- MSXE170x__IOWatchdogGetStatusAndValueResponse, [87](#)
 - sResponse, [87](#)
 - ulInfo, [87](#)
 - ulStatus, [87](#)
 - ulValue, [87](#)
- MSXE170x__IOWatchdogInitAndStart
 - MSXE170x_public_doc.h, [142](#)
 - MSXE170x_Watchdog, [35](#)
- MSXE170x__IOWatchdogStopAndRelease
 - MSXE170x_public_doc.h, [143](#)
 - MSXE170x_Watchdog, [35](#)
- MSXE170x__MFCommonReferenceVoltageActivation
 - MSXE170x_MF_Common, [38](#)
 - MSXE170x_public_doc.h, [145](#)
- MSXE170x__MFCommonSetInputsFilter
 - MSXE170x_MF_Common, [37](#)
 - MSXE170x_public_doc.h, [144](#)
- MSXE170x__MFIncCounterClear
 - MSXE170x_MF_IncCounter, [42](#)
 - MSXE170x_public_doc.h, [148](#)
- MSXE170x__MFIncCounterDisableAndReleaseCompareLogic
 - MSXE170x_MF_IncCounter, [43](#)
 - MSXE170x_public_doc.h, [149](#)
- MSXE170x__MFIncCounterDisableAndReleaseFrequencyMeasurement
 - MSXE170x_MF_IncCounter, [46](#)
 - MSXE170x_public_doc.h, [152](#)
- MSXE170x__MFIncCounterDisableAndReleaseIndex
 - MSXE170x_MF_IncCounter, [47](#)
 - MSXE170x_public_doc.h, [153](#)
- MSXE170x__MFIncCounterDisableAndReleaseLatchRegister
 - MSXE170x_MF_IncCounter, [45](#)
 - MSXE170x_public_doc.h, [150](#)
- MSXE170x__MFIncCounterGetNumber
 - MSXE170x_MF_IncCounter, [40](#)
 - MSXE170x_public_doc.h, [146](#)
- MSXE170x__MFIncCounterGetNumberResponse, [87](#)
 - sResponse, [88](#)
 - ulNumberOfCounter, [88](#)
- MSXE170x__MFIncCounterInit
 - MSXE170x_MF_IncCounter, [40](#)
 - MSXE170x_public_doc.h, [146](#)
- MSXE170x__MFIncCounterInitAndEnableCompareLogic
 - MSXE170x_MF_IncCounter, [43](#)
 - MSXE170x_public_doc.h, [148](#)
- MSXE170x__MFIncCounterInitAndEnableFrequencyMeasurement
 - MSXE170x_MF_IncCounter, [45](#)
 - MSXE170x_public_doc.h, [151](#)
- MSXE170x__MFIncCounterInitAndEnableIndex

- MSXE170x_MF_IncCounter, [47](#)
- MSXE170x_public_doc.h, [152](#)
- MSXE170x__MFIncCounterInitAndEnableLatchRegister
 - MSXE170x_MF_IncCounter, [44](#)
 - MSXE170x_public_doc.h, [150](#)
- MSXE170x__MFIncCounterRead32BitValue
 - MSXE170x_MF_IncCounter, [41](#)
 - MSXE170x_public_doc.h, [147](#)
- MSXE170x__MFIncCounterRelease
 - MSXE170x_MF_IncCounter, [41](#)
 - MSXE170x_public_doc.h, [146](#)
- MSXE170x__MFIncCounterSetFIFO0Level
 - MSXE170x_MF_IncCounter, [41](#)
 - MSXE170x_public_doc.h, [147](#)
- MSXE170x__MFIncCounterWrite32BitValue
 - MSXE170x_MF_IncCounter, [42](#)
 - MSXE170x_public_doc.h, [147](#)
- MSXE170x__Response, [88](#)
 - iReturnValue, [88](#)
 - syserrno, [88](#)
- MSXE170x__unsignedLongResponse, [88](#)
 - sResponse, [89](#)
 - ulValue, [89](#)
- MSXE170x__unsignedLongTimeStampResponse, [89](#)
 - sResponse, [89](#)
 - ulTimeStampHigh, [89](#)
 - ulTimeStampLow, [89](#)
 - ulValue, [89](#)
- MSXE170X_COUNTER_DECREMENT
 - MSXE170x_public_doc.h, [121](#)
- MSXE170X_COUNTER_DIRECT_MODE
 - MSXE170x_public_doc.h, [120](#)
- MSXE170X_COUNTER_DOUBLE_MODE
 - MSXE170x_public_doc.h, [120](#)
- MSXE170X_COUNTER_HIGH_EDGE_-
 - LATCH_AND_CLEAR_COUNTER
 - MSXE170x_public_doc.h, [121](#)
- MSXE170X_COUNTER_HIGH_EDGE_-
 - LATCH_COUNTER
 - MSXE170x_public_doc.h, [121](#)
- MSXE170X_COUNTER_HYSTERESIS_OFF
 - MSXE170x_public_doc.h, [120](#)
- MSXE170X_COUNTER_HYSTERESIS_ON
 - MSXE170x_public_doc.h, [120](#)
- MSXE170X_COUNTER_INCREMENT
 - MSXE170x_public_doc.h, [121](#)
- MSXE170X_COUNTER_LOW_EDGE_LATCH_-
 - AND_CLEAR_COUNTER
 - MSXE170x_public_doc.h, [121](#)
- MSXE170X_COUNTER_LOW_EDGE_LATCH_-
 - COUNTER
 - MSXE170x_public_doc.h, [121](#)
- MSXE170X_COUNTER_QUADRUPLE_MODE
 - MSXE170x_public_doc.h, [120](#)
- MSXE170X_COUNTER_SIMPLE_MODE
 - MSXE170x_public_doc.h, [120](#)
- MSXE170x_DigIO
 - MSXE170x__DigitalIOGetNumber, [31](#)
 - MSXE170x__DigitalIOInitPortConfiguration, [31](#)
 - MSXE170x__-
 - DigitalIOReadAllChannelsValue, [32](#)
 - MSXE170x__DigitalIOReadChannelValue, [31](#)
 - MSXE170x__DigitalIORearmShortCircuit, [34](#)
 - MSXE170x__-
 - DigitalIOReleasePortConfiguration, [33](#)
 - MSXE170x__DigitalIOTestShortCircuit, [33](#)
 - MSXE170x__-
 - DigitalIOWriteAllChannelsValue, [33](#)
 - MSXE170x__DigitalIOWriteChannelValue, [32](#)
- MSXE170x_MF_Common
 - MSXE170x__-
 - MFCommonReferenceVoltageActivation, [38](#)
 - MSXE170x__MFCommonSetInputsFilter, [37](#)
- MSXE170x_MF_IncCounter
 - MSXE170x__MFIncCounterClear, [42](#)
 - MSXE170x__-
 - MFIncCounterDisableAndReleaseCompareLogic, [43](#)
 - MSXE170x__-
 - MFIncCounterDisableAndReleaseFrequencyMeasurement, [46](#)
 - MSXE170x__-
 - MFIncCounterDisableAndReleaseIndex, [47](#)
 - MSXE170x__-
 - MFIncCounterDisableAndReleaseLatchRegister, [45](#)
 - MSXE170x__MFIncCounterGetNumber, [40](#)
 - MSXE170x__MFIncCounterInit, [40](#)
 - MSXE170x__-
 - MFIncCounterInitAndEnableCompareLogic, [43](#)
 - MSXE170x__-
 - MFIncCounterInitAndEnableFrequencyMeasurement, [45](#)
 - MSXE170x__-
 - MFIncCounterInitAndEnableIndex, [47](#)
 - MSXE170x__-
 - MFIncCounterInitAndEnableLatchRegister, [44](#)
 - MSXE170x__MFIncCounterRead32BitValue, [41](#)

- MSXE170x__MFIncCounterRelease, [41](#)
- MSXE170x__MFIncCounterSetFIFO0Level, [41](#)
- MSXE170x__-
 - MFIncCounterWrite32BitValue, [42](#)
- MSXE170x_public_doc.h, [107](#)
- MSXE170x__DigitalIOGetNumber, [139](#)
- MSXE170x__DigitalIOInitPortConfiguration, [139](#)
- MSXE170x__-
 - DigitalIOReadAllChannelsValue, [140](#)
- MSXE170x__DigitalIOReadChannelValue, [140](#)
- MSXE170x__DigitalIORearmShortCircuit, [142](#)
- MSXE170x__-
 - DigitalIOReleasePortConfiguration, [141](#)
- MSXE170x__DigitalIOTestShortCircuit, [142](#)
- MSXE170x__-
 - DigitalIOWriteAllChannelsValue, [141](#)
- MSXE170x__DigitalIOWriteChannelValue, [140](#)
- MSXE170x__-
 - IOWatchdogGetStatusAndValue, [143](#)
- MSXE170x__IOWatchdogInitAndStart, [142](#)
- MSXE170x__IOWatchdogStopAndRelease, [143](#)
- MSXE170x__-
 - MFCommonReferenceVoltageActivation, [145](#)
- MSXE170x__MFCommonSetInputsFilter, [144](#)
- MSXE170x__MFIncCounterClear, [148](#)
- MSXE170x__-
 - MFIncCounterDisableAndReleaseCompareLogic, [149](#)
- MSXE170x__-
 - MFIncCounterDisableAndReleaseFrequencyMeasurement, [152](#)
- MSXE170x__-
 - MFIncCounterDisableAndReleaseIndex, [153](#)
- MSXE170x__-
 - MFIncCounterDisableAndReleaseLatchRegister, [150](#)
- MSXE170x__MFIncCounterGetNumber, [146](#)
- MSXE170x__MFIncCounterInit, [146](#)
- MSXE170x__-
 - MFIncCounterInitAndEnableCompareLogic, [148](#)
- MSXE170x__-
 - MFIncCounterInitAndEnableFrequencyMeasurement, [151](#)
- MSXE170x__-
 - MFIncCounterInitAndEnableIndex, [152](#)
- MSXE170x__-
 - MFIncCounterInitAndEnableLatchRegister, [150](#)
- MSXE170x__MFIncCounterRead32BitValue, [147](#)
- MSXE170x__MFIncCounterRelease, [146](#)
- MSXE170x__MFIncCounterSetFIFO0Level, [147](#)
- MSXE170x__-
 - MFIncCounterWrite32BitValue, [147](#)
- MSXE170X_COUNTER_DECREMENT, [121](#)
- MSXE170X_COUNTER_DIRECT_MODE, [120](#)
- MSXE170X_COUNTER_DOUBLE_MODE, [120](#)
- MSXE170X_COUNTER_HIGH_EDGE_-LATCH_AND_CLEAR_COUNTER, [121](#)
- MSXE170X_COUNTER_HIGH_EDGE_-LATCH_COUNTER, [121](#)
- MSXE170X_COUNTER_HYSTERESIS_-OFF, [120](#)
- MSXE170X_COUNTER_HYSTERESIS_-ON, [120](#)
- MSXE170X_COUNTER_INCREMENT, [121](#)
- MSXE170X_COUNTER_LOW_EDGE_-LATCH_AND_CLEAR_COUNTER, [121](#)
- MSXE170X_COUNTER_LOW_EDGE_-LATCH_COUNTER, [121](#)
- MSXE170X_COUNTER_QUADRUPLE_-MODE, [120](#)
- MSXE170X_COUNTER_SIMPLE_MODE, [120](#)
- MSXE17xx__DigitalIOGetNumber, [154](#)
- MSXE17xx__DigitalIOInitPortConfiguration, [154](#)
- MSXE17xx__-
 - DigitalIOReadAllChannelsValue, [155](#)
- MSXE17xx__DigitalIOReadChannelValue, [154](#)
- MSXE17xx__DigitalIORearmShortCircuit, [157](#)
- MSXE17xx__-
 - DigitalIOReleasePortConfiguration, [156](#)
- MSXE17xx__DigitalIOTestShortCircuit, [156](#)
- MSXE17xx__-
 - DigitalIOWriteAllChannelsValue, [156](#)
- MSXE17xx__DigitalIOWriteChannelValue, [156](#)

- 155
- MSXE17xx__-
 - IOWatchdogGetStatusAndValue, 158
- MSXE17xx__IOWatchdogInitAndStart, 157
- MSXE17xx__IOWatchdogStopAndRelease, 158
- MSXE17xx__-
 - MFCCommonEnableDisableTriggerGate, 161
- MSXE17xx__-
 - MFCCommonGetSubModuleFunctionality, 159
- MSXE17xx__-
 - MFCCommonReferenceVoltageActivation, 160
- MSXE17xx__MFCCommonSetFIFO0Level, 161
- MSXE17xx__MFCCommonSetInputsFilter, 159
- MSXE17xx__MFIncCounterClear, 172
- MSXE17xx__-
 - MFIncCounterDisableAndReleaseClear, 180
- MSXE17xx__-
 - MFIncCounterDisableAndReleaseCompareLogic, 174
- MSXE17xx__-
 - MFIncCounterDisableAndReleaseFrequencyMeasurement, 183
- MSXE17xx__-
 - MFIncCounterDisableAndReleaseIndex, 184
- MSXE17xx__-
 - MFIncCounterDisableAndReleaseLatch, 178
- MSXE17xx__-
 - MFIncCounterDisableAndReleaseLatchRegister, 181
- MSXE17xx__MFIncCounterGetNumber, 170
- MSXE17xx__MFIncCounterInit, 170
- MSXE17xx__-
 - MFIncCounterInitAndEnableClear, 179
- MSXE17xx__-
 - MFIncCounterInitAndEnableCompareLogic, 173
- MSXE17xx__-
 - MFIncCounterInitAndEnableFrequencyMeasurement, 182
- MSXE17xx__-
 - MFIncCounterInitAndEnableIndex, 183
- MSXE17xx__-
 - MFIncCounterInitAndEnableLatch, 177
- MSXE17xx__-
 - MFIncCounterInitAndEnableLatchRegister, 180
- MSXE17xx__-
 - MFIncCounterInitHardwareTrigger, 174
- MSXE17xx__MFIncCounterInitIndex, 176
- MSXE17xx__MFIncCounterRead32BitValue, 171
- MSXE17xx__MFIncCounterRelease, 171
- MSXE17xx__-
 - MFIncCounterReleaseHardwareTrigger, 175
- MSXE17xx__MFIncCounterReleaseIndex, 176
- MSXE17xx__-
 - MFIncCounterWrite32BitValue, 172
- MSXE17xx__MFPWMDisable, 166
- MSXE17xx__MFPWMDisableAndRelease, 166
- MSXE17xx__MFPWMGetConfiguration, 167
- MSXE17xx__MFPWMGetNumber, 162
- MSXE17xx__MFPWMGetState, 169
- MSXE17xx__MFPWMInitAndEnable, 162
- MSXE17xx__MFPWMSetNewTiming, 165
- MXCommon__-
 - ApplyConfigurationBackupFile, 134
 - ChangePassword, 135
 - DataserverRestart, 124
 - GetAutoConfigurationFile, 131
 - GetClientConnections, 122
 - GetConfigurationBackupFile, 134
 - GetEthernetLinksStates, 125
 - GetHardwareTriggerFilterTime, 127
 - GetHardwareTriggerState, 128
 - GetHostname, 121
 - GetModuleTemperatureValueAndStatus, 126
 - GetModuleType, 121
 - GetOptionInformation, 137
 - GetStateIDFromName, 136
 - GetStateNameFromID, 137
 - GetSubsystemIDFromName, 136
 - GetSubsystemNameFromID, 137
 - GetSubSystemState, 135
 - GetSynchronizationStatus, 138

- MXCommon__GetTime, [130](#)
- MXCommon__GetUpTime, [131](#)
- MXCommon__HardwareClockToSys, [130](#)
- MXCommon__InitAndStartSynchroTimer, [132](#)
- MXCommon__Reboot, [124](#)
- MXCommon__ResetAllIOFunctionalities, [124](#)
- MXCommon__SetAutoConfigurationFile, [132](#)
- MXCommon__SetCustomerKey, [128](#)
- MXCommon__SetFilterChannels, [139](#)
- MXCommon__-
 - SetHardwareTriggerFilterTime, [127](#)
- MXCommon__SetHostname, [122](#)
- MXCommon__-
 - SetModuleTemperatureWarningLevels, [126](#)
- MXCommon__SetTime, [129](#)
- MXCommon__SetToMaster, [138](#)
- MXCommon__StartAutoConfiguration, [132](#)
- MXCommon__-
 - StopAndReleaseSynchroTimer, [133](#)
- MXCommon__Sterror, [122](#)
- MXCommon__SysToHardwareClock, [130](#)
- MXCommon__TestCustomerID, [129](#)
- xsd__char, [121](#)
- xsd__double, [121](#)
- xsd__float, [121](#)
- xsd__int, [121](#)
- xsd__long, [121](#)
- xsd__string, [121](#)
- xsd__unsignedByte, [121](#)
- xsd__unsignedInt, [121](#)
- xsd__unsignedLong, [121](#)
- xsd__unsignedShort, [121](#)
- MSXE170x__Watchdog
 - MSXE170x__-
 - IOWatchdogGetStatusAndValue, [36](#)
 - MSXE170x__IOWatchdogInitAndStart, [35](#)
 - MSXE170x__IOWatchdogStopAndRelease, [35](#)
- MSXE17xx__DigitalIOGetNumber
 - MSXE170x_public_doc.h, [154](#)
 - MSXE17xx_DigIO, [49](#)
- MSXE17xx__DigitalIOGetNumberResponse, [89](#)
 - sResponse, [90](#)
 - ulNumberOfDigitalIO, [90](#)
- MSXE17xx__DigitalIOInitPortConfiguration
 - MSXE170x_public_doc.h, [154](#)
 - MSXE17xx_DigIO, [49](#)
- MSXE17xx__DigitalIOReadAllChannelsValue
 - MSXE170x_public_doc.h, [155](#)
 - MSXE17xx_DigIO, [50](#)
- MSXE17xx__DigitalIOReadChannelValue
 - MSXE170x_public_doc.h, [154](#)
 - MSXE17xx_DigIO, [50](#)
- MSXE17xx__DigitalIORearmShortCircuit
 - MSXE170x_public_doc.h, [157](#)
 - MSXE17xx_DigIO, [52](#)
- MSXE17xx__DigitalIOReleasePortConfiguration
 - MSXE170x_public_doc.h, [156](#)
 - MSXE17xx_DigIO, [51](#)
- MSXE17xx__DigitalIOTestShortCircuit
 - MSXE170x_public_doc.h, [156](#)
 - MSXE17xx_DigIO, [52](#)
- MSXE17xx__DigitalIOWriteAllChannelsValue
 - MSXE170x_public_doc.h, [156](#)
 - MSXE17xx_DigIO, [51](#)
- MSXE17xx__DigitalIOWriteChannelValue
 - MSXE170x_public_doc.h, [155](#)
 - MSXE17xx_DigIO, [51](#)
- MSXE17xx__IOWatchdogGetStatusAndValue
 - MSXE170x_public_doc.h, [158](#)
 - MSXE17xx_Watchdog, [54](#)
- MSXE17xx__IOWatchdogGetStatusAndValueResponse, [90](#)
 - sResponse, [90](#)
 - ulInfo, [90](#)
 - ulStatus, [90](#)
 - ulValue, [90](#)
- MSXE17xx__IOWatchdogInitAndStart
 - MSXE170x_public_doc.h, [157](#)
 - MSXE17xx_Watchdog, [53](#)
- MSXE17xx__IOWatchdogStopAndRelease
 - MSXE170x_public_doc.h, [158](#)
 - MSXE17xx_Watchdog, [54](#)
- MSXE17xx__MFCommonEnableDisableTriggerGate
 - MSXE170x_public_doc.h, [161](#)
 - MSXE17xx_MF_Common, [57](#)
- MSXE17xx__MFCommonGetSubModuleFunctionality
 - MSXE170x_public_doc.h, [159](#)
 - MSXE17xx_MF_Common, [55](#)
- MSXE17xx__MFCommonReferenceVoltageActivation
 - MSXE170x_public_doc.h, [160](#)
 - MSXE17xx_MF_Common, [57](#)
- MSXE17xx__MFCommonSetFIFO0Level
 - MSXE170x_public_doc.h, [161](#)
 - MSXE17xx_MF_Common, [58](#)
- MSXE17xx__MFCommonSetInputsFilter
 - MSXE170x_public_doc.h, [159](#)
 - MSXE17xx_MF_Common, [56](#)
- MSXE17xx__MFIncCounterClear
 - MSXE170x_public_doc.h, [172](#)
 - MSXE17xx_MF_IncCounter, [72](#)
- MSXE17xx__MFIncCounterDisableAndReleaseClear
 - MSXE170x_public_doc.h, [180](#)
 - MSXE17xx_MF_IncCounter, [79](#)
- MSXE17xx__MFIncCounterDisableAndReleaseCompareLogic

- MSXE170x_public_doc.h, [174](#)
- MSXE17xx_MF_IncCounter, [74](#)
- MSXE17xx_MFIncCounterDisableAndReleaseFrequencyMSXE17xx_MF_IncCounter, [75](#)
- MSXE170x_public_doc.h, [183](#)
- MSXE17xx_MF_IncCounter, [82](#)
- MSXE17xx_MFIncCounterDisableAndReleaseIndexMSXE17xx_MF_IncCounter, [76](#)
- MSXE170x_public_doc.h, [176](#)
- MSXE17xx_MF_IncCounter, [76](#)
- MSXE17xx_MFIncCounterWrite32BitValueMSXE17xx_MF_IncCounter, [72](#)
- MSXE170x_public_doc.h, [172](#)
- MSXE17xx_MF_IncCounter, [72](#)
- MSXE17xx_MFIncCounterDisableMSXE17xx_MF_PWM, [63](#)
- MSXE170x_public_doc.h, [166](#)
- MSXE17xx_MF_PWM, [63](#)
- MSXE17xx_MFIncCounterDisableAndReleaseMSXE17xx_MF_PWM, [64](#)
- MSXE170x_public_doc.h, [166](#)
- MSXE17xx_MF_PWM, [64](#)
- MSXE17xx_MFPWMGetConfigurationMSXE17xx_MF_PWM, [64](#)
- MSXE170x_public_doc.h, [167](#)
- MSXE17xx_MF_PWM, [64](#)
- MSXE17xx_MFIncCounterGetNumberResponse, [90](#)
- sResponse, [91](#)
- ulNumberOfCounter, [91](#)
- MSXE17xx_MFIncCounterInitMSXE170x_public_doc.h, [170](#)
- MSXE17xx_MF_IncCounter, [70](#)
- MSXE17xx_MFIncCounterInitAndEnableClearMSXE170x_public_doc.h, [179](#)
- MSXE17xx_MF_IncCounter, [78](#)
- MSXE17xx_MFIncCounterInitAndEnableCompareLogicMSXE170x_public_doc.h, [173](#)
- MSXE17xx_MF_IncCounter, [73](#)
- MSXE17xx_MFIncCounterInitAndEnableFrequencyMeasurementMSXE170x_public_doc.h, [182](#)
- MSXE17xx_MF_IncCounter, [81](#)
- MSXE17xx_MFIncCounterInitAndEnableIndexMSXE170x_public_doc.h, [183](#)
- MSXE17xx_MF_IncCounter, [83](#)
- MSXE17xx_MFIncCounterInitAndEnableLatchMSXE170x_public_doc.h, [177](#)
- MSXE17xx_MF_IncCounter, [77](#)
- MSXE17xx_MFIncCounterInitAndEnableLatchRegisterMSXE170x_public_doc.h, [180](#)
- MSXE17xx_MF_IncCounter, [80](#)
- MSXE17xx_MFIncCounterInitHardwareTriggerMSXE170x_public_doc.h, [174](#)
- MSXE17xx_MF_IncCounter, [74](#)
- MSXE17xx_MFIncCounterInitIndexMSXE170x_public_doc.h, [176](#)
- MSXE17xx_MF_IncCounter, [75](#)
- MSXE17xx_MFIncCounterRead32BitValueMSXE170x_public_doc.h, [171](#)
- MSXE17xx_MF_IncCounter, [71](#)
- MSXE17xx_MFIncCounterReleaseMSXE170x_public_doc.h, [171](#)
- MSXE17xx_MF_IncCounter, [71](#)
- MSXE17xx_MFIncCounterReleaseHardwareTriggerMSXE170x_public_doc.h, [175](#)
- MSXE17xx_MF_IncCounter, [75](#)
- MSXE17xx_MFIncCounterReleaseIndexMSXE170x_public_doc.h, [176](#)
- MSXE17xx_MF_IncCounter, [76](#)
- MSXE17xx_MFIncCounterWrite32BitValueMSXE17xx_MF_IncCounter, [72](#)
- MSXE170x_public_doc.h, [172](#)
- MSXE17xx_MF_IncCounter, [72](#)
- MSXE17xx_MFPWMDisableMSXE17xx_MF_PWM, [63](#)
- MSXE170x_public_doc.h, [166](#)
- MSXE17xx_MF_PWM, [63](#)
- MSXE17xx_MFPWMDisableAndReleaseMSXE17xx_MF_PWM, [64](#)
- MSXE170x_public_doc.h, [166](#)
- MSXE17xx_MF_PWM, [64](#)
- MSXE17xx_MFPWMGetConfigurationMSXE17xx_MF_PWM, [64](#)
- MSXE170x_public_doc.h, [167](#)
- MSXE17xx_MF_PWM, [64](#)
- MSXE17xx_MFPWMGetConfigurationResponse, [91](#)
- sResponse, [92](#)
- ulDataFrame, [93](#)
- ulEnabled, [92](#)
- ulExternGate, [93](#)
- ulHighTime, [92](#)
- ulInfo01, [94](#)
- ulInfo02, [94](#)
- ulInitialised, [92](#)
- ulLowTime, [92](#)
- ulStartLevel, [93](#)
- ulStopLevel, [93](#)
- ulStopMode, [93](#)
- ulSynchroOut, [93](#)
- ulTimeBase, [92](#)
- MSXE17xx_MFPWMGetNumberMSXE170x_public_doc.h, [162](#)
- MSXE17xx_MF_PWM, [60](#)
- MSXE17xx_MFPWMGetStateMSXE170x_public_doc.h, [169](#)
- MSXE17xx_MF_PWM, [66](#)
- MSXE17xx_MFPWMGetStateResponse, [94](#)
- sResponse, [95](#)
- ulCounter, [96](#)
- ulDiffInput, [96](#)
- ulEnabled, [95](#)
- ulInfo01, [96](#)
- ulInfo02, [96](#)
- ulInitialised, [95](#)
- ulOutput, [95](#)
- ulProgress, [95](#)
- ulTriggerInput, [96](#)
- MSXE17xx_MFPWMInitAndEnableMSXE170x_public_doc.h, [162](#)
- MSXE17xx_MF_PWM, [60](#)

- MSXE17xx__MFPWMSetNewTiming
 - MSXE170x_public_doc.h, 165
 - MSXE17xx_MF_PWM, 62
- MSXE17xx__Response, 96
 - iReturnValue, 96
 - syserrno, 96
- MSXE17xx__unsignedLongResponse, 97
 - sResponse, 97
 - ulValue, 97
- MSXE17xx__unsignedLongTimeStampResponse, 97
 - sResponse, 98
 - ulTimeStampHigh, 98
 - ulTimeStampLow, 98
 - ulValue, 98
- MSXE17xx_DigIO
 - MSXE17xx__DigitalIOGetNumber, 49
 - MSXE17xx__DigitalIOInitPortConfiguration, 49
 - MSXE17xx__-
 - DigitalIOReadAllChannelsValue, 50
 - MSXE17xx__DigitalIOReadChannelValue, 50
 - MSXE17xx__DigitalIORearmShortCircuit, 52
 - MSXE17xx__-
 - DigitalIOReleasePortConfiguration, 51
 - MSXE17xx__DigitalIOTestShortCircuit, 52
 - MSXE17xx__-
 - DigitalIOWriteAllChannelsValue, 51
 - MSXE17xx__DigitalIOWriteChannelValue, 51
- MSXE17xx_MF_Common
 - MSXE17xx__-
 - MFCommonEnableDisableTriggerGate, 57
 - MSXE17xx__-
 - MFCommonGetSubModuleFunctionality, 55
 - MSXE17xx__-
 - MFCommonReferenceVoltageActivation, 57
 - MSXE17xx__MFCommonSetFIFO0Level, 58
 - MSXE17xx__MFCommonSetInputsFilter, 56
- MSXE17xx_MF_IncCounter
 - MSXE17xx__MFINcCounterClear, 72
 - MSXE17xx__-
 - MFINcCounterDisableAndReleaseClear, 79
 - MSXE17xx__-
 - MFINcCounterDisableAndReleaseCompareLogic, 74
 - MSXE17xx__-
 - MFINcCounterDisableAndReleaseFrequencyMeasurement, 82
- MSXE17xx__-
 - MFINcCounterDisableAndReleaseIndex, 84
- MSXE17xx__-
 - MFINcCounterDisableAndReleaseLatch, 78
- MSXE17xx__-
 - MFINcCounterDisableAndReleaseLatchRegister, 81
- MSXE17xx__MFINcCounterGetNumber, 70
- MSXE17xx__MFINcCounterInit, 70
- MSXE17xx__-
 - MFINcCounterInitAndEnableClear, 78
- MSXE17xx__-
 - MFINcCounterInitAndEnableCompareLogic, 73
- MSXE17xx__-
 - MFINcCounterInitAndEnableFrequencyMeasurement, 81
- MSXE17xx__-
 - MFINcCounterInitAndEnableIndex, 83
- MSXE17xx__-
 - MFINcCounterInitAndEnableLatch, 77
- MSXE17xx__-
 - MFINcCounterInitAndEnableLatchRegister, 80
- MSXE17xx__-
 - MFINcCounterInitHardwareTrigger, 74
- MSXE17xx__MFINcCounterInitIndex, 75
- MSXE17xx__MFINcCounterRead32BitValue, 71
- MSXE17xx__MFINcCounterRelease, 71
- MSXE17xx__-
 - MFINcCounterReleaseHardwareTrigger, 75
- MSXE17xx__MFINcCounterReleaseIndex, 76
- MSXE17xx__-
 - MFINcCounterWrite32BitValue, 72
- MSXE17xx_MF_PWM
 - MSXE17xx__MFPWMDisable, 63
 - MSXE17xx__MFPWMDisableAndRelease, 64
 - MSXE17xx__MFPWMGetConfiguration, 64
 - MSXE17xx__MFPWMGetNumber, 60
 - MSXE17xx__MFPWMGetState, 66
 - MSXE17xx__MFPWMInitAndEnable, 60
 - MSXE17xx__MFPWMSetNewTiming, 62
- MSXE17xx_Watchdog
 - MSXE17xx__-
 - IOWatchdogGetStatusAndValue, 54

- MSXE17xx__IOWatchdogInitAndStart, [53](#)
- MSXE17xx__IOWatchdogStopAndRelease, [54](#)
- MXCommon__ApplyConfigurationBackupFile
 - Common_configuration, [23](#)
 - MSXE170x_public_doc.h, [134](#)
- MXCommon__ByteArrayResponse, [98](#)
 - sArray, [98](#)
 - sResponse, [98](#)
- MXCommon__ChangePassword
 - Common_configuration, [24](#)
 - MSXE170x_public_doc.h, [135](#)
- MXCommon__DataseverRestart
 - Common_general, [8](#)
 - MSXE170x_public_doc.h, [124](#)
- MXCommon__FileResponse, [98](#)
 - sArray, [99](#)
 - sResponse, [99](#)
 - ulEOF, [99](#)
- MXCommon__GetAutoConfigurationFile
 - Common_autoconf, [19](#)
 - MSXE170x_public_doc.h, [131](#)
- MXCommon__GetAutoConfigurationFileResponse, [99](#)
 - bArray, [99](#)
 - sResponse, [99](#)
 - ulEOF, [99](#)
- MXCommon__GetClientConnections
 - Common_general, [6](#)
 - MSXE170x_public_doc.h, [122](#)
- MXCommon__GetConfigurationBackupFile
 - Common_configuration, [23](#)
 - MSXE170x_public_doc.h, [134](#)
- MXCommon__GetEthernetLinksStates
 - Common_general, [9](#)
 - MSXE170x_public_doc.h, [125](#)
- MXCommon__GetEthernetLinksStatesResponse, [99](#)
 - sPort0, [100](#)
 - sPort1, [100](#)
 - sResponse, [100](#)
- MXCommon__GetHardwareTriggerFilterTime
 - Common_hardware_trigger, [13](#)
 - MSXE170x_public_doc.h, [127](#)
- MXCommon__GetHardwareTriggerFilterTimeResponse, [100](#)
 - sResponse, [100](#)
 - ulFilterTime, [100](#)
 - ulInfo01, [100](#)
 - ulInfo02, [100](#)
- MXCommon__GetHardwareTriggerState
 - Common_hardware_trigger, [13](#)
 - MSXE170x_public_doc.h, [128](#)
- MXCommon__GetHardwareTriggerStateResponse, [100](#)
 - sResponse, [101](#)
 - ulInfo01, [101](#)
 - ulInfo02, [101](#)
 - ulState, [101](#)
- MXCommon__GetHostname
 - Common_general, [5](#)
 - MSXE170x_public_doc.h, [121](#)
- MXCommon__GetModuleTemperatureValueAndStatus
 - Common_temperature, [11](#)
 - MSXE170x_public_doc.h, [126](#)
- MXCommon__GetModuleTemperatureValueAndStatusResponse, [101](#)
 - dTemperatureValue, [102](#)
 - sResponse, [102](#)
 - ulInfo, [102](#)
 - ulTemperatureStatus, [102](#)
- MXCommon__GetModuleType
 - Common_general, [5](#)
 - MSXE170x_public_doc.h, [121](#)
- MXCommon__GetOptionInformation
 - CustomerOption, [28](#)
 - MSXE170x_public_doc.h, [137](#)
- MXCommon__GetStateIDFromName
 - MSXE170x_public_doc.h, [136](#)
 - SystemStatemanagement, [26](#)
- MXCommon__GetStateNameFromID
 - MSXE170x_public_doc.h, [137](#)
 - SystemStatemanagement, [27](#)
- MXCommon__GetSubsystemIDFromName
 - MSXE170x_public_doc.h, [136](#)
 - SystemStatemanagement, [26](#)
- MXCommon__GetSubsystemNameFromID
 - MSXE170x_public_doc.h, [137](#)
 - SystemStatemanagement, [26](#)
- MXCommon__GetSubSystemState
 - MSXE170x_public_doc.h, [135](#)
 - SystemStatemanagement, [25](#)
- MXCommon__GetSynchronizationStatus
 - MSXE170x_public_doc.h, [138](#)
 - Synchronisation, [29](#)
- MXCommon__GetTime
 - Common_time, [17](#)
 - MSXE170x_public_doc.h, [130](#)
- MXCommon__GetTimeResponse, [102](#)
 - sResponse, [102](#)
 - ulHighTime, [102](#)
 - ulLowTime, [102](#)
- MXCommon__GetUpTime
 - Common_time, [18](#)
 - MSXE170x_public_doc.h, [131](#)
- MXCommon__GetUpTimeResponse, [102](#)
 - sResponse, [103](#)

- ulUpTime, [103](#)
- MXCommon__HardwareClockToSys
 - Common_time, [17](#)
 - MSXE170x_public_doc.h, [130](#)
- MXCommon__InitAndStartSynchroTimer
 - Common_synchrotimer, [21](#)
 - MSXE170x_public_doc.h, [132](#)
- MXCommon__Reboot
 - Common_general, [8](#)
 - MSXE170x_public_doc.h, [124](#)
- MXCommon__ResetAllIOFunctionalities
 - Common_general, [8](#)
 - MSXE170x_public_doc.h, [124](#)
- MXCommon__Response, [103](#)
 - iReturnValue, [103](#)
 - syserrno, [103](#)
- MXCommon__SetAutoConfigurationFile
 - Common_autoconf, [19](#)
 - MSXE170x_public_doc.h, [132](#)
- MXCommon__SetCustomerKey
 - Common_security, [15](#)
 - MSXE170x_public_doc.h, [128](#)
- MXCommon__SetFilterChannels
 - Analog, [30](#)
 - MSXE170x_public_doc.h, [139](#)
- MXCommon__SetHardwareTriggerFilterTime
 - Common_hardware_trigger, [12](#)
 - MSXE170x_public_doc.h, [127](#)
- MXCommon__SetHostname
 - Common_general, [6](#)
 - MSXE170x_public_doc.h, [122](#)
- MXCommon__SetModuleTemperatureWarningLevels
 - Common_temperature, [11](#)
 - MSXE170x_public_doc.h, [126](#)
- MXCommon__SetTime
 - Common_time, [16](#)
 - MSXE170x_public_doc.h, [129](#)
- MXCommon__SetToMaster
 - MSXE170x_public_doc.h, [138](#)
 - Synchronisation, [28](#)
- MXCommon__StartAutoConfiguration
 - Common_autoconf, [20](#)
 - MSXE170x_public_doc.h, [132](#)
- MXCommon__StopAndReleaseSynchroTimer
 - Common_synchrotimer, [22](#)
 - MSXE170x_public_doc.h, [133](#)
- MXCommon__Strerror
 - Common_general, [6](#)
 - MSXE170x_public_doc.h, [122](#)
- MXCommon__SysToHardwareClock
 - Common_time, [17](#)
 - MSXE170x_public_doc.h, [130](#)
- MXCommon__TestCustomerID
 - Common_security, [15](#)
 - MSXE170x_public_doc.h, [129](#)
- MXCommon__TestCustomerIDResponse, [103](#)
 - bCryptedValueArray, [104](#)
 - bValueArray, [104](#)
 - sResponse, [104](#)
- MXCommon__unsignedLongResponse, [104](#)
 - sResponse, [104](#)
 - ulValue, [104](#)
- sArray
 - MXCommon__ByteArrayResponse, [98](#)
 - MXCommon__FileResponse, [99](#)
- Set/Backup/Restore general system configuration, [22](#)
- sGetEthernetLinksStatesPort, [104](#)
 - ulDuplex, [105](#)
 - ulInfo1, [105](#)
 - ulInfo2, [105](#)
 - ulSpeed, [105](#)
 - ulState, [105](#)
- sPort0
 - MXCommon__ -
 - GetEthernetLinksStatesResponse, [100](#)
- sPort1
 - MXCommon__ -
 - GetEthernetLinksStatesResponse, [100](#)
- sResponse
 - MSXE170x__DigitalIOGetNumberResponse, [87](#)
 - MSXE170x__ -
 - IOWatchdogGetStatusAndValueResponse, [87](#)
 - MSXE170x__ -
 - MFIncCounterGetNumberResponse, [88](#)
 - MSXE170x__unsignedLongResponse, [89](#)
 - MSXE170x__ -
 - unsignedLongTimeStampResponse, [89](#)
 - MSXE17xx__DigitalIOGetNumberResponse, [90](#)
 - MSXE17xx__ -
 - IOWatchdogGetStatusAndValueResponse, [90](#)
 - MSXE17xx__ -
 - MFIncCounterGetNumberResponse, [91](#)
 - MSXE17xx__ -
 - MFPWMGetConfigurationResponse, [92](#)
 - MSXE17xx__MFPWMGetStateResponse, [95](#)
 - MSXE17xx__unsignedLongResponse, [97](#)
 - MSXE17xx__ -
 - unsignedLongTimeStampResponse, [97](#)

- 98
- MXCommon__ByteArrayResponse, 98
- MXCommon__FileResponse, 99
- MXCommon__-
 - GetAutoConfigurationFileResponse, 99
- MXCommon__-
 - GetEthernetLinksStatesResponse, 100
- MXCommon__-
 - GetHardwareTriggerFilterTimeResponse, 100
- MXCommon__-
 - GetHardwareTriggerStateResponse, 101
- MXCommon__-
 - GetModuleTemperatureValueAndStatusResponse, 102
- MXCommon__GetTimeResponse, 102
- MXCommon__GetUpTimeResponse, 103
- MXCommon__TestCustomerIDResponse, 104
- MXCommon__unsignedLongResponse, 104
- Synchronisation
 - MXCommon__GetSynchronizationStatus, 29
 - MXCommon__SetToMaster, 28
- Synchronisation management, 28
- syserrno
 - DefaultResponse, 86
 - MSXE170x__Response, 88
 - MSXE17xx__Response, 96
 - MXCommon__Response, 103
- System state management, 25
- SystemStatemanagement
 - MXCommon__GetStateIDFromName, 26
 - MXCommon__GetStateNameFromID, 27
 - MXCommon__GetSubsystemIDFromName, 26
 - MXCommon__GetSubsystemNameFromID, 26
 - MXCommon__GetSubSystemState, 25
- ulCounter
 - MSXE17xx__MFPWMGetStateResponse, 96
- ulDataFrame
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, 93
- ulDiffInput
 - MSXE17xx__MFPWMGetStateResponse, 96
- ulDuplex
 - sGetEthernetLinksStatesPort, 105
- ulEnabled
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, 92
 - MSXE17xx__MFPWMGetStateResponse, 95
- ulEOF
 - MXCommon__FileResponse, 99
 - MXCommon__-
 - GetAutoConfigurationFileResponse, 99
- ulExternGate
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, 93
- ulFilterTime
 - MXCommon__-
 - GetHardwareTriggerFilterTimeResponse, 100
- ulHighTime
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, 92
 - MXCommon__GetTimeResponse, 102
- ulInfo
 - MSXE170x__-
 - IOWatchdogGetStatusAndValueResponse, 87
 - MSXE17xx__-
 - IOWatchdogGetStatusAndValueResponse, 90
 - MXCommon__-
 - GetModuleTemperatureValueAndStatusResponse, 102
- ulInfo01
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, 94
 - MSXE17xx__MFPWMGetStateResponse, 96
 - MXCommon__-
 - GetHardwareTriggerFilterTimeResponse, 100
 - MXCommon__-
 - GetHardwareTriggerStateResponse, 101
- ulInfo02
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, 94
 - MSXE17xx__MFPWMGetStateResponse, 96
 - MXCommon__-
 - GetHardwareTriggerFilterTimeResponse, 100
 - MXCommon__-
 - GetHardwareTriggerStateResponse, 101
- ulInfo1
 - sGetEthernetLinksStatesPort, 105
- ulInfo2

- sGetEthernetLinksStatesPort, [105](#)
- ulInitialised
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, [92](#)
 - MSXE17xx__MFPWMGetStateResponse, [95](#)
- ulLowTime
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, [92](#)
 - MXCommon__GetTimeResponse, [102](#)
- ulNumberOfCounter
 - MSXE170x__-
 - MFIncCounterGetNumberResponse, [88](#)
 - MSXE17xx__-
 - MFIncCounterGetNumberResponse, [91](#)
- ulNumberOfDigitalIO
 - MSXE170x__DigitalIOGetNumberResponse, [87](#)
 - MSXE17xx__DigitalIOGetNumberResponse, [90](#)
- ulOutput
 - MSXE17xx__MFPWMGetStateResponse, [95](#)
- ulProgress
 - MSXE17xx__MFPWMGetStateResponse, [95](#)
- ulSpeed
 - sGetEthernetLinksStatesPort, [105](#)
- ulStartLevel
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, [93](#)
- ulState
 - MXCommon__-
 - GetHardwareTriggerStateResponse, [101](#)
 - sGetEthernetLinksStatesPort, [105](#)
- ulStatus
 - MSXE170x__-
 - IOWatchdogGetStatusAndValueResponse, [87](#)
 - MSXE17xx__-
 - IOWatchdogGetStatusAndValueResponse, [90](#)
- ulStopLevel
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, [93](#)
- ulStopMode
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, [93](#)
- ulSynchroOut
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, [93](#)
- ulTemperatureStatus
 - MXCommon__-
 - GetModuleTemperatureValueAndStatusResponse, [102](#)
- ulTimeBase
 - MSXE17xx__-
 - MFPWMGetConfigurationResponse, [92](#)
- ulTimeStampHigh
 - MSXE170x__-
 - unsignedLongTimeStampResponse, [89](#)
 - MSXE17xx__-
 - unsignedLongTimeStampResponse, [98](#)
- ulTimeStampLow
 - MSXE170x__-
 - unsignedLongTimeStampResponse, [89](#)
 - MSXE17xx__-
 - unsignedLongTimeStampResponse, [98](#)
- ulTriggerInput
 - MSXE17xx__MFPWMGetStateResponse, [96](#)
- ulUpTime
 - MXCommon__GetUpTimeResponse, [103](#)
- ulValue
 - MSXE170x__-
 - IOWatchdogGetStatusAndValueResponse, [87](#)
 - MSXE170x__unsignedLongResponse, [89](#)
 - MSXE170x__-
 - unsignedLongTimeStampResponse, [89](#)
 - MSXE17xx__-
 - IOWatchdogGetStatusAndValueResponse, [90](#)
 - MSXE17xx__unsignedLongResponse, [97](#)
 - MSXE17xx__-
 - unsignedLongTimeStampResponse, [98](#)
 - MXCommon__unsignedLongResponse, [104](#)
- UnsignedLongArray, [105](#)
 - __offset, [105](#)
 - __ptr, [105](#)
 - __size, [105](#)
- UnsignedShortArray, [105](#)
 - __offset, [106](#)
 - __ptr, [106](#)
 - __size, [106](#)

- xsd__base64Binary, [106](#)
 - __ptr, [106](#)
 - __size, [106](#)
- xsd__char
 - MSXE170x_public_doc.h, [121](#)
- xsd__double
 - MSXE170x_public_doc.h, [121](#)
- xsd__float
 - MSXE170x_public_doc.h, [121](#)
- xsd__int
 - MSXE170x_public_doc.h, [121](#)
- xsd__long
 - MSXE170x_public_doc.h, [121](#)
- xsd__string
 - MSXE170x_public_doc.h, [121](#)
- xsd__unsignedByte
 - MSXE170x_public_doc.h, [121](#)
- xsd__unsignedInt
 - MSXE170x_public_doc.h, [121](#)
- xsd__unsignedLong
 - MSXE170x_public_doc.h, [121](#)
- xsd__unsignedShort
 - MSXE170x_public_doc.h, [121](#)