

MODBUS interface description

Table of Contents

General description	1
Introduction	1
Why a MODBUS Server on the MSX-E modules?	1
Technical details	1
FC3 (read multiple register) Functions	3
Function <u>GetLastCommandStatus</u>	5
For new application(s) or automate communication it is recommended to use the function	
<u>GetLastCommandStatusEx</u>	5
Description	6
Query frame layout	6
Response frame layout	6
Exception frame layout	7
Function <u>GetLastCommandStatusEx</u>	7
Description	8
Query frame layout	8
Response frame layout	8
Exception frame layout	9
Function <u>MXCommon GetModuleType</u>	9
For new application(s) or automate communication it is recommended to use the function	
<u>MXCommon GetModuleTypeEx</u>	10
Description	10
Query frame layout	10
Response frame layout	10
Exception frame layout	11
Function <u>MXCommon GetModuleTypeEx</u>	11
Description	11
Query frame layout	12
Response frame layout	12
Exception frame layout	13
Function <u>MXCommon GetTime</u>	13
For new application(s) or automate communication it is recommended to use the function	
<u>MXCommon GetTimeEx</u>	13
Description	13
Query frame layout	13
Response frame layout	14
Exception frame layout	14
Function <u>MXCommon GetTimeEx</u>	15
Description	15
Query frame layout	15
Response frame layout	16
Exception frame layout	16
Function <u>MXCommon TestCustomerID</u>	17
For new application(s) or automate communication it is recommended to use the function	
<u>MXCommon TestCustomerIDEx</u>	17
Description	17
Query frame layout	17
Response frame layout	18

Table of Contents

FC3 (read multiple register) Functions

Exception frame layout.....	18
Function MXCommon TestCustomerIDEx.....	19
Description.....	19
Query frame layout.....	19
Response frame layout.....	19
Exception frame layout.....	20
Function MX371x getNumberOfChannels.....	20
For new application(s) or automate communication it is recommended to use the function	
MX371x getNumberOfChannelsEx.....	21
Description.....	21
Query frame layout.....	21
Response frame layout.....	21
Exception frame layout.....	22
Function MX371x getNumberOfChannelsEx.....	22
Description.....	22
Query frame layout.....	23
Response frame layout.....	23
Exception frame layout.....	24
Function MX371x TransducerGetAutoRefreshValues.....	24
For new application(s) or automate communication it is recommended to use the function	
MX371x TransducerGetAutoRefreshValuesEx.....	24
Description.....	24
Query frame layout.....	25
Response frame layout.....	25
Exception frame layout.....	26
Function MX371x TransducerGetAutoRefreshValuesEx.....	26
Description.....	26
Query frame layout.....	26
Response frame layout.....	27
Exception frame layout.....	28
Function MSXE371x ExternalTemperatureRead.....	28
For new application(s) or automate communication it is recommended to use the function	
MSXE371x ExternalTemperatureReadEx.....	28
Description.....	28
Query frame layout.....	29
Response frame layout.....	29
Exception frame layout.....	30
Function MSXE371x ExternalTemperatureReadEx.....	30
Description.....	30
Query frame layout.....	31
Response frame layout.....	31
Exception frame layout.....	32
Function MX371x TransducerGetNbrOfType.....	32
For new application(s) or automate communication it is recommended to use the function	
MX371x TransducerGetNbrOfTypeEx.....	32
Description.....	32
Query frame layout.....	33

Table of Contents

FC3 (read multiple register) Functions

<u>Response frame layout</u>	33
<u>Exception frame layout</u>	34
<u>Function MX371x TransducerGetNbrOfTypeEx</u>	34
<u>Description</u>	34
<u>Query frame layout</u>	34
<u>Response frame layout</u>	35
<u>Exception frame layout</u>	36
<u>Function MX371x GetTransducerDatabaseCursor</u>	36
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x GetTransducerDatabaseCursorEx</u>	36
<u>Description</u>	36
<u>Query frame layout</u>	36
<u>Response frame layout</u>	37
<u>Exception frame layout</u>	37
<u>Function MX371x GetTransducerDatabaseCursorEx</u>	38
<u>Description</u>	38
<u>Query frame layout</u>	38
<u>Response frame layout</u>	39
<u>Exception frame layout</u>	39
<u>Function MX371x TransducerGetTypeInformation</u>	40
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerGetTypeInformationEx</u>	40
<u>Description</u>	40
<u>Query frame layout</u>	40
<u>Response frame layout</u>	41
<u>Exception frame layout</u>	42
<u>Function MX371x TransducerGetTypeInformationEx</u>	42
<u>Description</u>	43
<u>Query frame layout</u>	43
<u>Response frame layout</u>	44
<u>Exception frame layout</u>	45
<u>Function MX371x TransducerInitPrimaryConnectionTest</u>	45
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerInitPrimaryConnectionTestEx</u>	45
<u>Description</u>	45
<u>Query frame layout</u>	45
<u>Response frame layout</u>	46
<u>Exception frame layout</u>	46
<u>Function MX371x TransducerInitPrimaryConnectionTestEx</u>	47
<u>Description</u>	47
<u>Query frame layout</u>	47
<u>Response frame layout</u>	48
<u>Exception frame layout</u>	48
<u>Function MX371x TransducerTestPrimaryConnection</u>	49
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestPrimaryConnectionEx</u>	49
<u>Description</u>	49

Table of Contents

FC3 (read multiple register) Functions

<u>Query frame layout</u>	49
<u>Response frame layout</u>	50
<u>Exception frame layout</u>	50
Function MX371x <u>TransducerTestPrimaryConnectionEx</u>	51
<u>Description</u>	51
<u>Query frame layout</u>	51
<u>Response frame layout</u>	52
<u>Exception frame layout</u>	52
Function MX371x <u>TransducerTestPrimaryShortCircuit</u>	53
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestPrimaryShortCircuitEx</u>	53
<u>Description</u>	53
<u>Query frame layout</u>	53
<u>Response frame layout</u>	54
<u>Exception frame layout</u>	54
Function MX371x <u>TransducerTestPrimaryShortCircuitEx</u>	55
<u>Description</u>	55
<u>Query frame layout</u>	55
<u>Response frame layout</u>	56
<u>Exception frame layout</u>	56
Function MX371x <u>TransducerRearmPrimary</u>	57
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerRearmPrimaryEx</u>	57
<u>Description</u>	57
<u>Query frame layout</u>	57
<u>Response frame layout</u>	58
<u>Exception frame layout</u>	58
Function MX371x <u>TransducerRearmPrimaryEx</u>	59
<u>Description</u>	59
<u>Query frame layout</u>	59
<u>Response frame layout</u>	60
<u>Exception frame layout</u>	60
Function MX371x <u>TransducerTestSecondaryConnection0</u>	61
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestSecondaryConnection0Ex</u>	61
<u>Description</u>	61
<u>Query frame layout</u>	61
<u>Response frame layout</u>	62
<u>Exception frame layout</u>	62
Function MX371x <u>TransducerTestSecondaryConnection0Ex</u>	63
<u>Description</u>	63
<u>Query frame layout</u>	63
<u>Response frame layout</u>	64
<u>Exception frame layout</u>	64
Function MX371x <u>TransducerTestSecondaryConnection1</u>	65
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestSecondaryConnection1Ex</u>	65

Table of Contents

FC3 (read multiple register) Functions

<u>Description.....</u>	65
<u>Query frame layout.....</u>	65
<u>Response frame layout.....</u>	66
<u>Exception frame layout.....</u>	66
<u>Function MX371x TransducerTestSecondaryConnection1Ex.....</u>	67
<u>Description.....</u>	67
<u>Query frame layout.....</u>	67
<u>Response frame layout.....</u>	68
<u>Exception frame layout.....</u>	68
<u>Function MX371x TransducerTestSecondaryConnection2.....</u>	69
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestSecondaryConnection2Ex.....</u>	69
<u>Description.....</u>	69
<u>Query frame layout.....</u>	69
<u>Response frame layout.....</u>	70
<u>Exception frame layout.....</u>	70
<u>Function MX371x TransducerTestSecondaryConnection2Ex.....</u>	71
<u>Description.....</u>	71
<u>Query frame layout.....</u>	71
<u>Response frame layout.....</u>	72
<u>Exception frame layout.....</u>	72
<u>Function MX371x TransducerTestSecondaryConnection3.....</u>	73
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestSecondaryConnection3Ex.....</u>	73
<u>Description.....</u>	73
<u>Query frame layout.....</u>	73
<u>Response frame layout.....</u>	74
<u>Exception frame layout.....</u>	74
<u>Function MX371x TransducerTestSecondaryConnection3Ex.....</u>	75
<u>Description.....</u>	75
<u>Query frame layout.....</u>	75
<u>Response frame layout.....</u>	76
<u>Exception frame layout.....</u>	76
<u>Function MX371x TransducerTestSecondaryConnection4.....</u>	77
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestSecondaryConnection4Ex.....</u>	77
<u>Description.....</u>	77
<u>Query frame layout.....</u>	77
<u>Response frame layout.....</u>	78
<u>Exception frame layout.....</u>	78
<u>Function MX371x TransducerTestSecondaryConnection4Ex.....</u>	79
<u>Description.....</u>	79
<u>Query frame layout.....</u>	79
<u>Response frame layout.....</u>	80
<u>Exception frame layout.....</u>	80
<u>Function MX371x TransducerTestSecondaryConnection5.....</u>	81
<u>For new application(s) or automate communication it is recommended to use the function</u>	

Table of Contents

FC3 (read multiple register) Functions

<u>MX371x TransducerTestSecondaryConnection5Ex.....</u>	81
<u>Description.....</u>	81
<u>Query frame layout.....</u>	81
<u>Response frame layout.....</u>	82
<u>Exception frame layout.....</u>	82
<u>Function MX371x TransducerTestSecondaryConnection5Ex.....</u>	83
<u>Description.....</u>	83
<u>Query frame layout.....</u>	83
<u>Response frame layout.....</u>	84
<u>Exception frame layout.....</u>	84
<u>Function MX371x TransducerTestSecondaryConnection6.....</u>	85
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestSecondaryConnection6Ex.....</u>	85
<u>Description.....</u>	85
<u>Query frame layout.....</u>	85
<u>Response frame layout.....</u>	86
<u>Exception frame layout.....</u>	86
<u>Function MX371x TransducerTestSecondaryConnection6Ex.....</u>	87
<u>Description.....</u>	87
<u>Query frame layout.....</u>	87
<u>Response frame layout.....</u>	88
<u>Exception frame layout.....</u>	88
<u>Function MX371x TransducerTestSecondaryConnection7.....</u>	89
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestSecondaryConnection7Ex.....</u>	89
<u>Description.....</u>	89
<u>Query frame layout.....</u>	89
<u>Response frame layout.....</u>	90
<u>Exception frame layout.....</u>	90
<u>Function MX371x TransducerTestSecondaryConnection7Ex.....</u>	91
<u>Description.....</u>	91
<u>Query frame layout.....</u>	91
<u>Response frame layout.....</u>	92
<u>Exception frame layout.....</u>	92
<u>Function MX371x TransducerTestSecondaryShortCircuit0.....</u>	93
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerTestSecondaryShortCircuit0Ex.....</u>	93
<u>Description.....</u>	93
<u>Query frame layout.....</u>	93
<u>Response frame layout.....</u>	94
<u>Exception frame layout.....</u>	94
<u>Function MX371x TransducerTestSecondaryShortCircuit0Ex.....</u>	95
<u>Description.....</u>	95
<u>Query frame layout.....</u>	95
<u>Response frame layout.....</u>	96
<u>Exception frame layout.....</u>	96
<u>Function MX371x TransducerTestSecondaryShortCircuit1.....</u>	97

Table of Contents

FC3 (read multiple register) Functions

For new application(s) or automate communication it is recommended to use the function	
<u>MX371x TransducerTestSecondaryShortCircuit1Ex</u>	97
<u>Description</u>	97
<u>Query frame layout</u>	97
<u>Response frame layout</u>	98
<u>Exception frame layout</u>	98
Function <u>MX371x TransducerTestSecondaryShortCircuit1Ex</u>	99
<u>Description</u>	99
<u>Query frame layout</u>	99
<u>Response frame layout</u>	100
<u>Exception frame layout</u>	100
Function <u>MX371x TransducerTestSecondaryShortCircuit2</u>	101
For new application(s) or automate communication it is recommended to use the function	
<u>MX371x TransducerTestSecondaryShortCircuit2Ex</u>	101
<u>Description</u>	101
<u>Query frame layout</u>	101
<u>Response frame layout</u>	102
<u>Exception frame layout</u>	102
Function <u>MX371x TransducerTestSecondaryShortCircuit2Ex</u>	103
<u>Description</u>	103
<u>Query frame layout</u>	103
<u>Response frame layout</u>	104
<u>Exception frame layout</u>	104
Function <u>MX371x TransducerTestSecondaryShortCircuit3</u>	105
For new application(s) or automate communication it is recommended to use the function	
<u>MX371x TransducerTestSecondaryShortCircuit3Ex</u>	105
<u>Description</u>	105
<u>Query frame layout</u>	105
<u>Response frame layout</u>	106
<u>Exception frame layout</u>	106
Function <u>MX371x TransducerTestSecondaryShortCircuit3Ex</u>	107
<u>Description</u>	107
<u>Query frame layout</u>	107
<u>Response frame layout</u>	108
<u>Exception frame layout</u>	108
Function <u>MX371x TransducerTestSecondaryShortCircuit4</u>	109
For new application(s) or automate communication it is recommended to use the function	
<u>MX371x TransducerTestSecondaryShortCircuit4Ex</u>	109
<u>Description</u>	109
<u>Query frame layout</u>	109
<u>Response frame layout</u>	110
<u>Exception frame layout</u>	110
Function <u>MX371x TransducerTestSecondaryShortCircuit4Ex</u>	111
<u>Description</u>	111
<u>Query frame layout</u>	111
<u>Response frame layout</u>	112
<u>Exception frame layout</u>	112

Table of Contents

FC3 (read multiple register) Functions

<u>Function MX371x TransducerTestSecondaryShortCircuit5</u>	113
For new application(s) or automate communication it is recommended to use the function	
<u>MX371x TransducerTestSecondaryShortCircuit5Ex</u>	113
<u>Description</u>	113
<u>Query frame layout</u>	113
<u>Response frame layout</u>	114
<u>Exception frame layout</u>	114
<u>Function MX371x TransducerTestSecondaryShortCircuit5Ex</u>	115
<u>Description</u>	115
<u>Query frame layout</u>	115
<u>Response frame layout</u>	116
<u>Exception frame layout</u>	116
<u>Function MX371x TransducerTestSecondaryShortCircuit6</u>	117
For new application(s) or automate communication it is recommended to use the function	
<u>MX371x TransducerTestSecondaryShortCircuit6Ex</u>	117
<u>Description</u>	117
<u>Query frame layout</u>	117
<u>Response frame layout</u>	118
<u>Exception frame layout</u>	118
<u>Function MX371x TransducerTestSecondaryShortCircuit6Ex</u>	119
<u>Description</u>	119
<u>Query frame layout</u>	119
<u>Response frame layout</u>	120
<u>Exception frame layout</u>	120
<u>Function MX371x TransducerTestSecondaryShortCircuit7</u>	121
For new application(s) or automate communication it is recommended to use the function	
<u>MX371x TransducerTestSecondaryShortCircuit7Ex</u>	121
<u>Description</u>	121
<u>Query frame layout</u>	121
<u>Response frame layout</u>	122
<u>Exception frame layout</u>	122
<u>Function MX371x TransducerTestSecondaryShortCircuit7Ex</u>	123
<u>Description</u>	123
<u>Query frame layout</u>	123
<u>Response frame layout</u>	124
<u>Exception frame layout</u>	124
<u>Function MSXE371x InternalTemperatureRead</u>	125
For new application(s) or automate communication it is recommended to use the function	
<u>MSXE371x InternalTemperatureReadEx</u>	125
<u>Description</u>	125
<u>Query frame layout</u>	125
<u>Response frame layout</u>	126
<u>Exception frame layout</u>	126
<u>Function MSXE371x InternalTemperatureReadEx</u>	126
<u>Description</u>	127
<u>Query frame layout</u>	127
<u>Response frame layout</u>	127

Table of Contents

FC3 (read multiple register) Functions

<u>Exception frame layout</u>	128
<u>Function MSXE371x IncCounterRead32BitValue</u>	128
<u>Description</u>	128
<u>Query frame layout</u>	129
<u>Response frame layout</u>	129
<u>Exception frame layout</u>	130

FC16 (write multiple register) Functions.....131

<u>Function MXCommon SetHardwareTriggerFilterTime</u>	132
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon SetHardwareTriggerFilterTimeEx</u>	132
<u>Description</u>	132
<u>Query frame layout</u>	133
<u>Response frame layout</u>	133
<u>Exception frame layout</u>	134
<u>Function MXCommon SetHardwareTriggerFilterTimeEx</u>	134
<u>Description</u>	135
<u>Query frame layout</u>	135
<u>Response frame layout</u>	136
<u>Exception frame layout</u>	136
<u>Function MXCommon InitAndStartSynchroTimer</u>	137
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon InitAndStartSynchroTimerEx</u>	137
<u>Description</u>	137
<u>Query frame layout</u>	138
<u>Response frame layout</u>	139
<u>Exception frame layout</u>	139
<u>Function MXCommon InitAndStartSynchroTimerEx</u>	140
<u>Description</u>	140
<u>Query frame layout</u>	141
<u>Response frame layout</u>	142
<u>Exception frame layout</u>	142
<u>Function MXCommon StopAndReleaseSynchroTimer</u>	143
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon StopAndReleaseSynchroTimerEx</u>	143
<u>Description</u>	143
<u>Query frame layout</u>	143
<u>Response frame layout</u>	144
<u>Exception frame layout</u>	144
<u>Function MXCommon StopAndReleaseSynchroTimerEx</u>	145
<u>Description</u>	145
<u>Query frame layout</u>	145
<u>Response frame layout</u>	146
<u>Exception frame layout</u>	146
<u>Function MXCommon Reboot</u>	147
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon RebootEx</u>	147

Table of Contents

FC16 (write multiple register) Functions

<u>Description</u>	147
<u>Query frame layout</u>	147
<u>Response frame layout</u>	148
<u>Exception frame layout</u>	148
<u>Function MXCommon RebootEx</u>	149
<u>Description</u>	149
<u>Query frame layout</u>	149
<u>Response frame layout</u>	150
<u>Exception frame layout</u>	150
<u>Function MXCommon SetCustomerKey</u>	151
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon SetCustomerKeyEx</u>	151
<u>Description</u>	151
<u>Query frame layout</u>	151
<u>Response frame layout</u>	152
<u>Exception frame layout</u>	153
<u>Function MXCommon SetCustomerKeyEx</u>	153
<u>Description</u>	153
<u>Query frame layout</u>	153
<u>Response frame layout</u>	154
<u>Exception frame layout</u>	155
<u>Function MXCommon SetFilterChannels</u>	155
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MXCommon SetFilterChannelsEx</u>	155
<u>Description</u>	155
<u>Query frame layout</u>	156
<u>Response frame layout</u>	156
<u>Exception frame layout</u>	157
<u>Function MXCommon SetFilterChannelsEx</u>	157
<u>Description</u>	157
<u>Query frame layout</u>	158
<u>Response frame layout</u>	158
<u>Exception frame layout</u>	159
<u>Function MX371x TransducerInitAndStartAutoRefresh</u>	159
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerInitAndStartAutoRefreshEx</u>	159
<u>Description</u>	159
<u>Query frame layout</u>	161
<u>Response frame layout</u>	163
<u>Exception frame layout</u>	163
<u>Function MX371x TransducerInitAndStartAutoRefreshEx</u>	164
<u>Description</u>	164
<u>Query frame layout</u>	166
<u>Response frame layout</u>	167
<u>Exception frame layout</u>	168
<u>Function MX371x TransducerStopAndReleaseAutoRefresh</u>	168
<u>For new application(s) or automate communication it is recommended to use the function</u>	

Table of Contents

FC16 (write multiple register) Functions

<u>MX371x TransducerStopAndReleaseAutoRefreshEx.....</u>	168
<u>Description.....</u>	168
<u>Query frame layout.....</u>	169
<u>Response frame layout.....</u>	169
<u>Exception frame layout.....</u>	170
<u>Function MX371x TransducerStopAndReleaseAutoRefreshEx.....</u>	170
<u>Description.....</u>	170
<u>Query frame layout.....</u>	171
<u>Response frame layout.....</u>	171
<u>Exception frame layout.....</u>	172
<u>Function MX371x TransducerInitAndStartSequence.....</u>	172
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerInitAndStartSequenceEx.....</u>	172
<u>Description.....</u>	172
<u>Query frame layout.....</u>	175
<u>Response frame layout.....</u>	177
<u>Exception frame layout.....</u>	177
<u>Function MX371x TransducerInitAndStartSequenceEx.....</u>	178
<u>Description.....</u>	178
<u>Query frame layout.....</u>	180
<u>Response frame layout.....</u>	182
<u>Exception frame layout.....</u>	183
<u>Function MX371x TransducerStopAndReleaseSequence.....</u>	183
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x TransducerStopAndReleaseSequenceEx.....</u>	183
<u>Description.....</u>	183
<u>Query frame layout.....</u>	183
<u>Response frame layout.....</u>	184
<u>Exception frame layout.....</u>	185
<u>Function MX371x TransducerStopAndReleaseSequenceEx.....</u>	185
<u>Description.....</u>	185
<u>Query frame layout.....</u>	185
<u>Response frame layout.....</u>	186
<u>Exception frame layout.....</u>	187
<u>Function MSXE371x ExternalTemperatureInit.....</u>	187
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MSXE371x ExternalTemperatureInitEx.....</u>	187
<u>Description.....</u>	187
<u>Query frame layout.....</u>	188
<u>Response frame layout.....</u>	189
<u>Exception frame layout.....</u>	190
<u>Function MSXE371x ExternalTemperatureInitEx.....</u>	190
<u>Description.....</u>	190
<u>Query frame layout.....</u>	191
<u>Response frame layout.....</u>	192
<u>Exception frame layout.....</u>	193
<u>Function MSXE371x ExternalTemperatureRelease.....</u>	193

Table of Contents

FC16 (write multiple register) Functions

<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MSXE371x ExternalTemperatureReleaseEx.....</u>	193
<u>Description.....</u>	193
<u>Query frame layout.....</u>	194
<u>Response frame layout.....</u>	194
<u>Exception frame layout.....</u>	195
<u>Function MSXE371x ExternalTemperatureReleaseEx.....</u>	195
<u>Description.....</u>	195
<u>Query frame layout.....</u>	195
<u>Response frame layout.....</u>	196
<u>Exception frame layout.....</u>	197
<u>Function MX371x SetTransducerDatabaseCursor.....</u>	197
<u>For new application(s) or automate communication it is recommended to use the function</u>	
<u>MX371x SetTransducerDatabaseCursorEx.....</u>	197
<u>Description.....</u>	197
<u>Query frame layout.....</u>	198
<u>Response frame layout.....</u>	198
<u>Exception frame layout.....</u>	199
<u>Function MX371x SetTransducerDatabaseCursorEx.....</u>	199
<u>Description.....</u>	199
<u>Query frame layout.....</u>	199
<u>Response frame layout.....</u>	200
<u>Exception frame layout.....</u>	201
<u>Function MSXE371x IncCounterInit.....</u>	201
<u>Description.....</u>	201
<u>Query frame layout.....</u>	202
<u>Response frame layout.....</u>	203
<u>Exception frame layout.....</u>	203
<u>Function MSXE371x IncCounterRelease.....</u>	204
<u>Description.....</u>	204
<u>Query frame layout.....</u>	204
<u>Response frame layout.....</u>	205
<u>Exception frame layout.....</u>	205
<u>Function MSXE371x IncCounterClear.....</u>	206
<u>Description.....</u>	206
<u>Query frame layout.....</u>	206
<u>Response frame layout.....</u>	207
<u>Exception frame layout.....</u>	207
<u>Function MSXE371x IncCounterWrite32BitValue.....</u>	208
<u>Description.....</u>	208
<u>Query frame layout.....</u>	208
<u>Response frame layout.....</u>	209
<u>Exception frame layout.....</u>	209
<u>Function MSXE371x IncCounterInitAndEnableCompareLogic.....</u>	210
<u>Description.....</u>	210
<u>Query frame layout.....</u>	210
<u>Response frame layout.....</u>	211

Table of Contents

<u>FC16 (write multiple register) Functions</u>	
<u>Exception frame layout</u>	212
<u>Function MSXE371x IncCounterDisableAndReleaseCompareLogic</u>	212
<u>Description</u>	212
<u>Query frame layout</u>	212
<u>Response frame layout</u>	213
<u>Exception frame layout</u>	214
<u>Function MSXE371x IncCounterInitAndEnableIndex</u>	214
<u>Description</u>	214
<u>Query frame layout</u>	215
<u>Response frame layout</u>	215
<u>Exception frame layout</u>	216
<u>Function MSXE371x IncCounterDisableAndReleaseIndex</u>	216
<u>Description</u>	217
<u>Query frame layout</u>	217
<u>Response frame layout</u>	218
<u>Exception frame layout</u>	218
<u>FC23 (read/write registers) Functions</u>	219
<u>Query frame layout</u>	219
<u>Response frame layout</u>	220
<u>Exception frame layout</u>	220
<u>Exception code description</u>	221
<u>Siemens Step 7 compatibility information (AWL/SDF code)</u>	222

General description

[Top](#)

Introduction

This document describes the protocol used by the MODBUS server of the module. The OPEN MODBUS protocol is based on the widely known MODBUS protocol. OPEN MODBUS is an open protocol and is not manufacturer dependent. It is mainly used to connect PLC and I/O devices.

Why a MODBUS Server on the MSX-E modules?

Thanks to the MODBUS server, it is possible to manage an MSX-E module with e.g.: a Siemens S7 PLC. The S7 PLC can start acquisitions and read data from the MSX-E module!

Technical details

Please note that only MODBUS over TCP is standardized. Nonetheless in this present version the server implements OPEN MODBUS/TCP class 0 and one function of the class 2 even on UDP sockets.

The MODBUS/TCP class 0 defines two types of query: FC3 and FC16.

- **FC3 functions** read register content from the memory of the remote system
- **FC16 functions** write new register content on the memory of the remote system

The MODBUS/TCP server implement the following query of the class 2 : FC23.

- **FC23 functions** read/write registers content from/to the memory of the remote system

The MODBUS server offer a virtual memory organisation: registers (functions) are mapped to be equivalent to SOAP functions.

Characteristics of this communication channel as the standardisation document describes it are:

- The default port used by the server is **512** in both UDP/IP and TCP/IP. You can change this via the web server.
- Data are sent in network order, i.e. **big endian (Motorola formata)**. Use the standard C functions `atons/atohl` and `ntohs/ntohl` to convert values bigger than 1 bytes.
- Datastructures used to describe parameters that are embedded in on-wire frames **must** be packed. How to do that is compiler-dependant.

The ADDI-DATA MSX-E Modbus server offers the following extension to the standard:

- It is possible to configure the server to accept data sent in **little endian (Intel format)** (native order)
- In this case, the default port used is **215**. You can change this via the web server.

MODBUS interface description

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

If an exception or an error occurred, you can use the GetLastCommandStatus command to get the real error number (from the remote server).

Real error numbers are described for each command in the "Returns" field.

The chapter below describes the available functions and their parameters.

It also contains the precise description of all frames implied in a given action.

FC3 (read multiple register) Functions

[Top](#)

Functions in this group are used to read values on the module.

• <u>GetLastCommandStatus</u>	Register: 0
• <u>GetLastCommandStatusEx</u>	Register: 10000
• <u>MXCommon_GetModuleType</u>	Register: 1
• <u>MXCommon_GetModuleTypeEx</u>	Register: 10200
• <u>MXCommon_GetTime</u>	Register: 2
• <u>MXCommon_GetTimeEx</u>	Register: 10500
• <u>MXCommon_TestCustomerID</u>	Register: 3
• <u>MXCommon_TestCustomerIDEx</u>	Register: 10550
• <u>MX371x_getNumberOfChannels</u>	Register: 100
• <u>MX371x_getNumberOfChannelsEx</u>	Register: 1000
• <u>MX371x_TransducerGetAutoRefreshValues</u>	Register: 101
• <u>MX371x_TransducerGetAutoRefreshValuesEx</u>	Register: 1050
• <u>MSXE371x_ExternalTemperatureRead</u>	Register: 102
• <u>MSXE371x_ExternalTemperatureReadEx</u>	Register: 1200
• <u>MX371x_TransducerGetNbrOfType</u>	Register: 103
• <u>MX371x_TransducerGetNbrOfTypeEx</u>	Register: 1212
• <u>MX371x_GetTransducerDatabaseCursor</u>	Register: 104
• <u>MX371x_GetTransducerDatabaseCursorEx</u>	Register: 1216
• <u>MX371x_TransducerGetTypeInformation</u>	Register: 105
• <u>MX371x_TransducerGetTypeInformationEx</u>	Register: 1220

MODBUS interface description

• <u>MX371x TransducerInitPrimaryConnectionTest</u>	Register: 106
• <u>MX371x TransducerInitPrimaryConnectionTestEx</u>	Register: 1218
• <u>MX371x TransducerTestPrimaryConnection</u>	Register: 107
• <u>MX371x TransducerTestPrimaryConnectionEx</u>	Register: 1222
• <u>MX371x TransducerTestPrimaryShortCircuit</u>	Register: 108
• <u>MX371x TransducerTestPrimaryShortCircuitEx</u>	Register: 1224
• <u>MX371x TransducerRearmPrimary</u>	Register: 109
• <u>MX371x TransducerRearmPrimaryEx</u>	Register: 1226
• <u>MX371x TransducerTestSecondaryConnection0</u>	Register: 110
• <u>MX371x TransducerTestSecondaryConnection0Ex</u>	Register: 1228
• <u>MX371x TransducerTestSecondaryConnection1</u>	Register: 111
• <u>MX371x TransducerTestSecondaryConnection1Ex</u>	Register: 1230
• <u>MX371x TransducerTestSecondaryConnection2</u>	Register: 112
• <u>MX371x TransducerTestSecondaryConnection2Ex</u>	Register: 1232
• <u>MX371x TransducerTestSecondaryConnection3</u>	Register: 113
• <u>MX371x TransducerTestSecondaryConnection3Ex</u>	Register: 1234
• <u>MX371x TransducerTestSecondaryConnection4</u>	Register: 114
• <u>MX371x TransducerTestSecondaryConnection4Ex</u>	Register: 1236
• <u>MX371x TransducerTestSecondaryConnection5</u>	Register: 115
• <u>MX371x TransducerTestSecondaryConnection5Ex</u>	Register: 1238
• <u>MX371x TransducerTestSecondaryConnection6</u>	Register: 116
• <u>MX371x TransducerTestSecondaryConnection6Ex</u>	Register: 1240
• <u>MX371x TransducerTestSecondaryConnection7</u>	Register: 117

MODBUS interface description

• <u>MX371x TransducerTestSecondaryConnection7Ex</u>	Register: 1242
• <u>MX371x TransducerTestSecondaryShortCircuit0</u>	Register: 118
• <u>MX371x TransducerTestSecondaryShortCircuit0Ex</u>	Register: 1244
• <u>MX371x TransducerTestSecondaryShortCircuit1</u>	Register: 119
• <u>MX371x TransducerTestSecondaryShortCircuit1Ex</u>	Register: 1246
• <u>MX371x TransducerTestSecondaryShortCircuit2</u>	Register: 120
• <u>MX371x TransducerTestSecondaryShortCircuit2Ex</u>	Register: 1248
• <u>MX371x TransducerTestSecondaryShortCircuit3</u>	Register: 121
• <u>MX371x TransducerTestSecondaryShortCircuit3Ex</u>	Register: 1250
• <u>MX371x TransducerTestSecondaryShortCircuit4</u>	Register: 122
• <u>MX371x TransducerTestSecondaryShortCircuit4Ex</u>	Register: 1252
• <u>MX371x TransducerTestSecondaryShortCircuit5</u>	Register: 123
• <u>MX371x TransducerTestSecondaryShortCircuit5Ex</u>	Register: 1254
• <u>MX371x TransducerTestSecondaryShortCircuit6</u>	Register: 124
• <u>MX371x TransducerTestSecondaryShortCircuit6Ex</u>	Register: 1256
• <u>MX371x TransducerTestSecondaryShortCircuit7</u>	Register: 125
• <u>MX371x TransducerTestSecondaryShortCircuit7Ex</u>	Register: 1258
• <u>MSXE371x InternalTemperatureRead</u>	Register: 126
• <u>MSXE371x InternalTemperatureReadEx</u>	Register: 1260
• <u>MSXE371x IncCounterRead32BitValue</u>	Register: 1600

Function GetLastCommandStatus

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

Description

Return the result of the last remote function call

Parameters:

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

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

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

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

Query frame layout

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

Response frame layout

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

MODBUS interface description

			server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	112	0x7000	0x0070
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	108	0x6C00	0x006C
Return Value	4	32-bit integer	See the description above	0x????????	0x????????
Syserrno	4	32-bit integer	See the description above	0x????????	0x????????
Errstr	100	8-bit integer array	See the description above	0x??[100]	0x??[100]

Exception frame layout

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

Function GetLastCommandStatusEx

Description

Return the result of the last remote function call

Parameters:

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

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

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

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

Query frame layout

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

Response frame layout

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

MODBUS interface description

			server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	111	0x6F00	0x006F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	108	0x6C	0x6C
Return Value	4	32-bit integer	See the description above	0x????????	0x????????
Syserrno	4	32-bit integer	See the description above	0x????????	0x????????
Errstr	100	8-bit integer array	See the description above	0x??[100]	0x??[100]

Exception frame layout

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

Function MXCommon_GetModuleType

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

Description

Returns the type of the MSX-E Module

Parameters:

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

Query frame layout

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

Response frame layout

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

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

MODBUS interface description

protocol identifier		16-bit integer			
length	2	16-bit integer	204	0xCC00	0x00CC
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	200	0xC800	0x00C8
str	200	8-bit integer array	See the description above	0x??[200]	0x??[200]

Exception frame layout

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

Function MXCommon__GetModuleTypeEx

Description

Returns the type of the MSX-E Module

Parameters:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	10200	0xD827	0x27D8
word count	2	16-bit integer	100	0x6400	0x0064

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	203	0xCB00	0x00CB
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	200	0xC8	0xC8
str	200	8-bit	See the	0x??[200]	0x??[200]

MODBUS interface description

		integer array	description above		
--	--	------------------	----------------------	--	--

Exception frame layout

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

Function MXCommon__GetTime

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

Description

Get the time on the module

Parameters:

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

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

Query frame layout

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

Response frame layout

MODBUS interface description

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

Response frame layout

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

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MXCommon__GetTimeEx

Description

Get the time on the module

Parameters:

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

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

Query frame layout

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

Exception frame layout

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	10500	0x0429	0x2904
word count	2	16-bit integer	4	0x0400	0x0004

Response frame layout

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

Exception frame layout

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

MODBUS interface description

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

Function MXCommon__TestCustomerID

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

Description

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

Parameters:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number	2	16-bit integer	3	0x0300	0x0003

Exception frame layout

MODBUS interface description

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

Response frame layout

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

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MXCommon__TestCustomerIDEx

Description

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

Parameters:

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

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

Query frame layout

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

Response frame layout

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

Exception frame layout

MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	35	0x2300	0x0023
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	32	0x20	0x20
bValueArray	16	8-bit integer array	See the description above	0x??[16]	0x??[16]
bCryptedValueArray	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

Exception frame layout

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

Function MX371x__getNumberOfChannels

For new application(s) or automate communication it is recommended to use the function MX371x__getNumberOfChannelsEx.

Description

Return the number of transducer channels on the module (4,8 or 16)

Parameters:

[Response frame layout] **ChannelNumber:** Number of channels

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	100	0x6400	0x0064
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

For new application(s) or automate communication it is recommended to use the function MX371x21_getNun

MODBUS interface description

protocol identifier		16-bit integer			
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ChannelNumber	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

Function MX371x__getNumberOfChannelsEx

Description

Return the number of transducer channels on the module (4,8 or 16)

Parameters:

[Response frame layout] **ChannelNumber**: Number of channels

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1000	0xE803	0x03E8
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

			above		
--	--	--	-------	--	--

Exception frame layout

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

Function MX371x__TransducerGetAutoRefreshValues

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerGetAutoRefreshValuesEx.

Description

This function get the auto refresh counter value an the channels values

Parameters:

[Response frame layout] **Value:** Array that contain the counter and channels values

- ◆ **Values [0]:** Auto refresh counter value
- ◆ **Values [1]:** Channel 0 value
- ◆ ...
- ◆ **Values [16]:** Channel 15 value

Returns:

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

- ◆ 0 : means the remote function performed OK
- ◆ -100 : GetAutoRefreshAllValues kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	101	0x6500	0x0065
word count	2	16-bit integer	34	0x2200	0x0022

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	72	0x4800	0x0048
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	68	0x4400	0x0044
Value	68	32-bit integer	See the description	0x???????[17]	0x???????[17]

		array	above		
--	--	-------	-------	--	--

Exception frame layout

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

Function MX371x__TransducerGetAutoRefreshValuesEx

Description

This function get the auto refresh counter value an the channels values

Parameters:

[Response frame layout] **Value:** Array that contain the counter and channels values

- ◆ **Values [0]:** Auto refresh counter value
- ◆ **Values [1]:** Channel 0 value
- ◆ ...
- ◆ **Values [16]:** Channel 15 value

Returns:

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

- ◆ 0 : means the remote function performed OK
- ◆ -100 : `GetAutoRefreshAllValues` kernel function error

Query frame layout

Field	Size	Type	Value	little	big endian
-------	------	------	-------	--------	------------

Response frame layout

MODBUS interface description

	(Bytes)			endian (Intel)	(Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1050	0x1A04	0x041A
word count	2	16-bit integer	34	0x2200	0x0022

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	71	0x4700	0x0047
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	68	0x44	0x44
Value	68	32-bit integer array	See the description above	0x???????[17]	0x???????[17]

Exception frame layout

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

Function MSXE371x__ExternalTemperatureRead

For new application(s) or automate communication it is recommended to use the function MSXE371x__ExternalTemperatureReadEx.

Description

Read the external temperature

Parameters:

[Response frame layout] **Value:** External temperature value. Format depend from the initialisation

[Response frame layout] **TimeStampLow:** Time stamp low dword (us)

[Response frame layout] **TimeStampHigh:** Time stamp high dword (s)

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Power save enabled and no transducer acquisition started
- ◆ -3: Power save enabled and transducer acquisition started but hardware trigger used
- ◆ -100: Read external temperature kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	102	0x6600	0x0066
word count	2	16-bit integer	6	0x0600	0x0006

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	16	0x1000	0x0010
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	12	0x0C00	0x000C
Value	4	32-bit integer	See the description	0x???????	0x???????

MODBUS interface description

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

Exception frame layout

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

Function MSXE371x__ExternalTemperatureReadEx

Description

Read the external temperature

Parameters:

[Response frame layout] **Value:** External temperature value. Format depend from the initialisation

[Response frame layout] **TimeStampLow:** Time stamp low dword (us)

[Response frame layout] **TimeStampHigh:** Time stamp high dword (s)

Returns:

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

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

Response frame layout

MODBUS interface description

- ◆ -2: Power save enabled and no transducer acquisition started
- ◆ -3: Power save enabled and transducer acquisition started but hardware trigger used
- ◆ -100: Read external temperature kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1200	0xB004	0x04B0
word count	2	16-bit integer	6	0x0600	0x0006

Response frame layout

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

MODBUS interface description

		8-bit integer			
Value	4	32-bit integer	See the description above	0x????????	0x????????
TimeStampLow	4	32-bit integer	See the description above	0x????????	0x????????
TimeStampHigh	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

Function MX371x__TransducerGetNbrOfType

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerGetNbrOfTypeEx.

Description

Returns the number of transducer types currently defined in the database.

Parameters:

[Query frame layout] **NumberOfTransducerTypes:** number of transducer types currently defined.

Returns:

Response frame layout

MODBUS interface description

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

- 0 : success
- otherwise : internal error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	103	0x6700	0x0067
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03

MODBUS interface description

Byte count	2	16-bit integer	4	0x0400	0x0004
NumberOfTransducerTypes	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

Function MX371x__TransducerGetNbrOfTypeEx

Description

Returns the number of transducer types currently defined in the database.

Parameters:

[Query frame layout] **NumberOfTransducerTypes**: number of transducer types currently defined.

Returns:

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

- 0 : success
- otherwise : internal error

Query frame layout

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

Response frame layout

MODBUS interface description

				(Intel)	
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1212	0xBC04	0x04BC
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__GetTransducerDatabaseCursor

For new application(s) or automate communication it is recommended to use the function MX371x__GetTransducerDatabaseCursorEx.

Description

Returns the current cursor of the transducer database.

Parameters:

[Query frame layout] **TransducerDatabaseCursor**: Current cursor. This is an integer from 0 .. (NumberOfTransducerTypes-1)

Returns:

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

- 0 : success
- otherwise : internal error

Query frame layout

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

MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	104	0x6800	0x0068
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
TransducerDatabaseCursor	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MX371x__GetTransducerDatabaseCursorEx

Description

Returns the current cursor of the transducer database.

Parameters:

[Query frame layout] **TransducerDatabaseCursor:** Current cursor. This is an integer from 0 .. (NumberOfTransducerTypes-1)

Returns:

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

- 0 : success
- otherwise : internal error

Query frame layout

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

Exception frame layout

MODBUS interface description

protocol identifier		16-bit integer			
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1216	0xC004	0x04C0
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MX371x__TransducerGetTypeInformation

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerGetTypeInformationEx.

Description

Returns the information stored in the database about the type selected by the current TransducerDatabaseCursor.

Parameters:

SelectionIndex : Identifier. Value to use for the transducer type selection in the other SOAP functions.

Name : Name of the transducer type

CalibrationStatus : Calibration status \li 0 : Transducer type is not calibrated

\li 1 : Transducer type is calibrated

CalibratedChannels : Bitmask of currently calibrated channels (D0 => channel 1, D1 => channel 1, ...)

Type : Type (0: HB 1: LVDT 2:Knaebel 3:HB-Mahr 4:LVDT-Mahr) **Frequency** : Frequency (Hz)

Impedance : Impedance (Ohm)

Veff : Nominal voltage (Vrms)

Sensibility : Sensibility (mv/V/mm)

Range : Range (mm)

Returns:

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

- 0 : success
- otherwise : internal error

Query frame layout

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

Exception frame layout

MODBUS interface description

			- copied by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	105	0x6900	0x0069
word count	2	16-bit integer	67	0x4300	0x0043

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	137	0x8900	0x0089
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	133	0x8500	0x0085
SelectionIndex	4	32-bit integer	See the description above	0x????????	0x????????
Name	100	8-bit integer array	See the description above	0x??[100]	0x??[100]
CalibrationStatus	1	8-bit integer	See the description above	0x??	0x??

MODBUS interface description

CalibratedChannels	4	32-bit integer	See the description above	0x????????	0x????????
Type	4	32-bit integer	See the description above	0x????????	0x????????
Frequency	4	32-bit integer	See the description above	0x????????	0x????????
Impedance	4	32-bit integer	See the description above	0x????????	0x????????
NominalVoltage	4	32-bit floating point	See the description above	0x????????	0x????????
Sensibility	4	32-bit floating point	See the description above	0x????????	0x????????
Range	4	32-bit floating point	See the description above	0x????????	0x????????

Exception frame layout

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

Function MX371x__TransducerGetTypeInformationEx

Description

Returns the information stored in the database about the type selected by the current TransducerDatabaseCursor.

Parameters:

SelectionIndex : Identifier. Value to use for the transducer type selection in the other SOAP functions.

Name : Name of the transducer type

CalibrationStatus : Calibration status \li 0 : Transducer type is not calibrated

\li 1 : Transducer type is calibrated

CalibratedChannels : Bitmask of currently calibrated channels (D0 => channel 1, D1 => channel 1, ...)

Type : Type (0: HB 1: LVDT 2:Knaebel 3:HB-Mahr 4:LVDT-Mahr) **Frequency** : Frequency (Hz)

Impedance : Impedance (Ohm)

Veff : Nominal voltage (Vrms)

Sensibility : Sensibility (mv/V/mm)

Range : Range (mm)

Returns:

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

- 0 : success
- otherwise : internal error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1220	0xC404	0x04C4

MODBUS interface description

word count	2	16-bit integer	67	0x4300	0x0043
------------	---	----------------	----	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	136	0x8800	0x0088
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	133	0x85	0x85
SelectionIndex	4	32-bit integer	See the description above	0x????????	0x????????
Name	100	8-bit integer array	See the description above	0x??[100]	0x??[100]
CalibrationStatus	1	8-bit integer	See the description above	0x??	0x??
CalibratedChannels	4	32-bit integer	See the description above	0x????????	0x????????
Type	4	32-bit integer	See the description above	0x????????	0x????????
Frequency	4	32-bit integer	See the description above	0x????????	0x????????
Impedance	4	32-bit integer	See the description above	0x????????	0x????????
NominalVoltage	4	32-bit floating point	See the description above	0x????????	0x????????
Sensibility	4	32-bit floating point	See the description above	0x????????	0x????????

Query frame layout

MODBUS interface description

Range	4	32-bit floating point	See the description above	0x????????	0x????????
-------	---	-----------------------	---------------------------	------------	------------

Exception frame layout

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

Function MX371x__TransducerInitPrimaryConnectionTest

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerInitPrimaryConnectionTestEx.

Description

Initialise the primary connection test

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

- 0: success
- -1: otherwise : internal error
- -100> Primary connection test initialization kernel function error

Query frame layout

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

Response frame layout

MODBUS interface description

			by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	106	0x6A00	0x006A
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
TransducerInitPrimaryConnection	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MX371x__TransducerInitPrimaryConnectionTestEx

Description

Initialise the primary connection test

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

- 0: success
- -1: otherwise : internal error
- -100> Primary connection test initialization kernel function error

Query frame layout

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

Exception frame layout

MODBUS interface description

code					
Reference number (=register)	2	16-bit integer	1218	0xC204	0x04C2
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Query frame layout

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

Function MX371x__TransducerTestPrimaryConnection

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestPrimaryConnectionEx.

Description

Test primary connection

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

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

MODBUS interface description

Reference number (=register)	2	16-bit integer	107	0x6B00	0x006B
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

Query frame layout

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

Function MX371x__TransducerTestPrimaryConnectionEx

Description

Test primary connection

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1222	0xC604	0x04C6
word count	2		2	0x0200	0x0002

Exception frame layout

MODBUS interface description

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

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestPrimaryShortCircuit

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestPrimaryShortCircuitEx.

Description

Test primary short circuit

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	108	0x6C00	0x006C
word count	2		2	0x0200	0x0002

MODBUS interface description

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

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulStatus	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function MX371x__TransducerTestPrimaryShortCircuitEx

Description

Test primary short circuit

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1224	0xC804	0x04C8
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerRearmPrimary

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerRearmPrimaryEx.

Description

Rearm primary

Parameters:

[Response frame layout] **ulValue:** Rearm status

- 0: Rearm not ok
- 1: Rearm ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	109	0x6D00	0x006D
word count	2		2	0x0200	0x0002

MODBUS interface description

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

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function MX371x__TransducerRearmPrimaryEx

Description

Rearm primary

Parameters:

[Response frame layout] **ulValue:** Rearm status

- 0: Rearm not ok
- 1: Rearm ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1226	0xCA04	0x04CA
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryConnection0

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryConnection0Ex.

Description

Test the secondary connection for channel 0 (you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	110	0x6E00	0x006E

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryConnection0Ex

Description

Test the secondary connection for channel 0 (you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1228	0xCC04	0x04CC
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryConnection1

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryConnection1Ex.

Description

Test the secondary connection for channel 1(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	111	0x6F00	0x006F

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryConnection1Ex

Description

Test the secondary connection for channel 1(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1230	0xCE04	0x04CE
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryConnection2

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryConnection2Ex.

Description

Test the secondary connection for channel 2(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	112	0x7000	0x0070

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryConnection2Ex

Description

Test the secondary connection for channel 2(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1232	0xD004	0x04D0
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryConnection3

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryConnection3Ex.

Description

Test the secondary connection for channel 3(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	113	0x7100	0x0071

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryConnection3Ex

Description

Test the secondary connection for channel 3(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1234	0xD204	0x04D2
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryConnection4

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryConnection4Ex.

Description

Test the secondary connection for channel 4(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	114	0x7200	0x0072

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryConnection4Ex

Description

Test the secondary connection for channel 4(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1236	0xD404	0x04D4
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryConnection5

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryConnection5Ex.

Description

Test the secondary connection for channel 5(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	115	0x7300	0x0073

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryConnection5Ex

Description

Test the secondary connection for channel 5(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1238	0xD604	0x04D6
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryConnection6

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryConnection6Ex.

Description

Test the secondary connection for channel 6(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	116	0x7400	0x0074

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryConnection6Ex

Description

Test the secondary connection for channel 6(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1240	0xD804	0x04D8
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryConnection7

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryConnection7Ex.

Description

Test the secondary connection for channel 7(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	117	0x7500	0x0075

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
Value	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryConnection7Ex

Description

Test the secondary connection for channel 7(you must call the MX371X__TransducerSelectChannel function before to select the channel you want to use)

Parameters:

[Response frame layout] **ulValue:** Connection status

- 0: connection error
- 1: connection ok

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1242	0xDA04	0x04DA
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryShortCircuit0

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryShortCircuit0Ex.

Description

Test the secondary short circuit status for channel 0(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	118	0x7600	0x0076

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryShortCircuit0Ex

Description

Test the secondary short circuit status for channel 0(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1244	0xDC04	0x04DC
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryShortCircuit1

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryShortCircuit1Ex.

Description

Test the secondary short circuit status for channel 1(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	119	0x7700	0x0077

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryShortCircuit1Ex

Description

Test the secondary short circuit status for channel 1(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1246	0xDE04	0x04DE
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryShortCircuit2

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryShortCircuit2Ex.

Description

Test the secondary short circuit status for channel 2(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	120	0x7800	0x0078

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryShortCircuit2Ex

Description

Test the secondary short circuit status for channel 2(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1248	0xE004	0x04E0
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryShortCircuit3

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryShortCircuit3Ex.

Description

Test the secondary short circuit status for channel 3 (You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	121	0x7900	0x0079

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryShortCircuit3Ex

Description

Test the secondary short circuit status for channel 3 (You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1250	0xE204	0x04E2
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryShortCircuit4

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryShortCircuit4Ex.

Description

Test the secondary short circuit status for channel 4(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	122	0x7A00	0x007A

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryShortCircuit4Ex

Description

Test the secondary short circuit status for channel 4(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1252	0xE404	0x04E4
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryShortCircuit5

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryShortCircuit5Ex.

Description

Test the secondary short circuit status for channel 5(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	123	0x7B00	0x007B

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryShortCircuit5Ex

Description

Test the secondary short circuit status for channel 5(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1254	0xE604	0x04E6
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryShortCircuit6

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryShortCircuit6Ex.

Description

Test the secondary short circuit status for channel 6(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	124	0x7C00	0x007C

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryShortCircuit6Ex

Description

Test the secondary short circuit status for channel 6(You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1256	0xE804	0x04E8
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MX371x__TransducerTestSecondaryShortCircuit7

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerTestSecondaryShortCircuit7Ex.

Description

Test the secondary short circuit status for channel 7 (You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	125	0x7D00	0x007D

MODBUS interface description

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
ulValue	4	32-bit integer	See the description above	0x???????	0x???????

Exception frame layout

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

Function

MX371x__TransducerTestSecondaryShortCircuit7Ex

Description

Test the secondary short circuit status for channel 7 (You must call the MX371X__TransducerSelectChannel function before to select the channel)

Parameters:

[Response frame layout] **ulValue:** short circuit status

- 0: short circuit
- 1: no short circuit

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1258	0xEA04	0x04EA
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

Exception frame layout

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

Function MSXE371x__InternalTemperatureRead

For new application(s) or automate communication it is recommended to use the function MSXE371x__InternalTemperatureReadEx.

Description

Get the MSXE internal temperature in degree

Parameters:

[Response frame layout] ***InternalTemperature:*** internal temperature in degree

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	126	0x7E00	0x007E
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	8	0x0800	0x0008
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	2	16-bit integer	4	0x0400	0x0004
fInternalTemperature	4	32-bit floating point	See the description above	0x???????	0x???????

Exception frame layout

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

Function MSXE371x__InternalTemperatureReadEx

Description

Get the MSXE internal temperature in degree

Parameters:

[Response frame layout] ***InternalTemperature***: internal temperature in degree

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1260	0xEC04	0x04EC
word count	2	16-bit integer	2	0x0200	0x0002

Response frame layout

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

MODBUS interface description

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

Exception frame layout

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

Function MSXE371x__IncCounterRead32BitValue

Description

Read the 32 bits counter value.

Parameters:

[Response frame layout] **ulValue** : Counter value.

[Response frame layout] **ulTimeStampLow** : 32 bit low part of time stamp (us)

Response frame layout

MODBUS interface description

[Response frame layout] **ulTimeStampHigh** : 32 bit high part of time stamp (s)

Returns:

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Reference number (=register)	2	16-bit integer	1600	0x4006	0x0640
word count	2	16-bit integer	6	0x0600	0x0006

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F

MODBUS interface description

unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x03	0x03	0x03
Byte count	1	8-bit integer	12	0x0C	0x0C
ulValue	4	32-bit integer	See the description above	0x????????	0x????????
ulTimeStampLow	4	32-bit integer	See the description above	0x????????	0x????????
ulTimeStampHigh	4	32-bit integer	See the description above	0x????????	0x????????

Exception frame layout

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

FC16 (write multiple register) Functions

[Top](#)

Functions in this group are used to set value on the module.

- [MXCommon_SetHardwareTriggerFilterTime](#) Register: **100**
- [MXCommon_SetHardwareTriggerFilterTimeEx](#) Register: **11000**
- [MXCommon_InitAndStartSynchroTimer](#) Register: **101**
- [MXCommon_InitAndStartSynchroTimerEx](#) Register: **11050**
- [MXCommon_StopAndReleaseSynchroTimer](#) Register: **102**
- [MXCommon_StopAndReleaseSynchroTimerEx](#) Register: **11100**
- [MXCommon_Reboot](#) Register: **103**
- [MXCommon_RebootEx](#) Register: **11150**
- [MXCommon_SetCustomerKey](#) Register: **104**
- [MXCommon_SetCustomerKeyEx](#) Register: **11200**
- [MXCommon_SetFilterChannels](#) Register: **105**
- [MXCommon_SetFilterChannelsEx](#) Register: **11250**
- [MX371x_TransducerInitAndStartAutoRefresh](#) Register: **1**
- [MX371x_TransducerInitAndStartAutoRefreshEx](#) Register: **1250**
- [MX371x_TransducerStopAndReleaseAutoRefresh](#) Register: **2**
- [MX371x_TransducerStopAndReleaseAutoRefreshEx](#) Register: **1300**
- [MX371x_TransducerInitAndStartSequence](#) Register: **3**
- [MX371x_TransducerInitAndStartSequenceEx](#) Register: **1350**
- [MX371x_TransducerStopAndReleaseSequence](#) Register: **4**
- [MX371x_TransducerStopAndReleaseSequenceEx](#) Register: **1400**

MODBUS interface description

- MSXE371x ExternalTemperatureInit Register: **5**
- MSXE371x ExternalTemperatureInitEx Register: **1450**
- MSXE371x ExternalTemperatureRelease Register: **6**
- MSXE371x ExternalTemperatureReleaseEx Register: **1500**
- MX371x SetTransducerDatabaseCursor Register: **7**
- MX371x SetTransducerDatabaseCursorEx Register: **1504**
- MSXE371x IncCounterInit Register: **1650**
- MSXE371x IncCounterRelease Register: **1700**
- MSXE371x IncCounterClear Register: **1750**
- MSXE371x IncCounterWrite32BitValue Register: **1800**
- MSXE371x IncCounterInitAndEnableCompareLogic Register: **1850**
- MSXE371x IncCounterDisableAndReleaseCompareLogic Register: **1900**
- MSXE371x IncCounterInitAndEnableIndex Register: **2000**
- MSXE371x IncCounterDisableAndReleaseIndex Register: **2100**

Function MXCommon__SetHardwareTriggerFilterTime

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

Description

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

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

Parameters

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

- ◆ ...
- ◆ **65535**: filter of 16ms
- [Query frame layout] **ulOption** Reserved. Set to 0

Returns

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

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

Query frame layout

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

Response frame layout

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

MODBUS interface description

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

Exception frame layout

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

Function MXCommon__SetHardwareTriggerFilterTimeEx

Description

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

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

Parameters

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

Returns

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	15	0x0F00	0x000F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11000	0xF82A	0x2AF8
word count	2	16-bit integer	4	0x0400	0x0004
byte count	1		8	0x08	0x08

MODBUS interface description

		8-bit integer			
ulFilterTime	4	32-bit integer	See the description above	0x????????	0x????????
Reserved	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MXCommon__InitAndStartSynchroTimer

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

Description

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

Parameters:

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

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

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

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

[Query frame layout] **ulGenerateTriggerMode:**

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

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

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

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

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

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

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

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

Returns:

Exception frame layout

MODBUS interface description

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	40	0x2800	0x0028
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	101	0x6500	0x0065
word count	2	16-bit integer	16	0x1000	0x0010
byte count	2	16-bit integer	32	0x2000	0x0020
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulReloadValue	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrOfCycle	4	32-bit integer	See the description above	0x????????	0x????????
ulGenerateTriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4			0x????????	0x????????

MODBUS interface description

		32-bit integer	See the description above		
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????
ulOption03	4	32-bit integer	See the description above	0x????????	0x????????
ulOption04	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MXCommon__InitAndStartSynchroTimerEx

Description

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

Parameters:

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

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

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

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

[Query frame layout] **ulGenerateTriggerMode:**

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

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

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

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

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

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

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

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

Returns:

Exception frame layout

MODBUS interface description

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	39	0x2700	0x0027
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11050	0x2A2B	0x2B2A
word count	2	16-bit integer	16	0x1000	0x0010
byte count	1	8-bit integer	32	0x20	0x20
ulTimeBase	4	32-bit integer	See the description above	0x????????	0x????????
ulReloadValue	4	32-bit integer	See the description above	0x????????	0x????????
ulNbrOfCycle	4	32-bit integer	See the description above	0x????????	0x????????
ulGenerateTriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4			0x????????	0x????????

MODBUS interface description

		32-bit integer	See the description above		
ulOption02	4	32-bit integer	See the description above	0x????????	0x????????
ulOption03	4	32-bit integer	See the description above	0x????????	0x????????
ulOption04	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11050	0x2A2B	0x2B2A
word count	2	16-bit integer	16	0x1000	0x0010

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MXCommon__StopAndReleaseSynchroTimer

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

Description

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

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	12	0x0C00	0x000C
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01

Exception frame layout

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	102	0x6600	0x0066
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	102	0x6600	0x0066
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

MODBUS interface description

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

Function MXCommon__StopAndReleaseSynchroTimerEx

Description

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

Parameters:

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

Returns:

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

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

Query frame layout

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

Exception frame layout

MODBUS interface description

Function code		integer			
Reference number (=register)	2	16-bit integer	11100	0x5C2B	0x2B5C
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11100	0x5C2B	0x2B5C
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Query frame layout

MODBUS interface description

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

Function MXCommon__Reboot

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

Description

Ask the MSX-E module to reboot

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	12	0x0C00	0x000C
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01

Exception frame layout

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	103	0x6700	0x0067
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
Dummy	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	103	0x6700	0x0067
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

MODBUS interface description

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

Function MXCommon__RebootEx

Description

Ask the MSX-E module to reboot

Parameters:

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

Returns:

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

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

Query frame layout

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

Exception frame layout

MODBUS interface description

code					
Reference number (=register)	2	16-bit integer	11150	0x8E2B	0x2B8E
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
Dummy	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11150	0x8E2B	0x2B8E
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

MODBUS interface description

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

Function MXCommon__SetCustomerKey

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

Description

Permit to set the Customer key

Parameters:

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

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2		56	0x3800	0x0038

Exception frame layout

MODBUS interface description

		16-bit integer			
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	104	0x6800	0x0068
word count	2	16-bit integer	24	0x1800	0x0018
byte count	2	16-bit integer	48	0x3000	0x0030
bKey	32	8-bit integer array	See the description above	0x??[32]	0x??[32]
bPublicKey	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

Response frame layout

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

Exception frame layout

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

Function MXCommon__SetCustomerKeyEx

Description

Permit to set the Customer key

Parameters:

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

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

Returns:

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

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

Query frame layout

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

MODBUS interface description

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

Response frame layout

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

MODBUS interface description

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

Exception frame layout

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

Function MXCommon__SetFilterChannels

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

Description

Permit to set a filter per channel

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	24	0x1800	0x0018
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	105	0x6900	0x0069
word count	2	16-bit integer	8	0x0800	0x0008
byte count	2	16-bit integer	16	0x1000	0x0010
ChannelList	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

Response frame layout

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

MODBUS interface description

Function code		integer			
Reference number (=register)	2	16-bit integer	105	0x6900	0x0069
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function MXCommon__SetFilterChannelsEx

Description

Permit to set a filter per channel

Parameters:

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

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	11250	0xF22B	0x2BF2
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ChannelList	16	8-bit integer array	See the description above	0x??[16]	0x??[16]

Response frame layout

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

MODBUS interface description

Function code		integer			
Reference number (=register)	2	16-bit integer	11250	0xF22B	0x2BF2
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function MX371x__TransducerInitAndStartAutoRefresh

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerInitAndStartAutoRefreshEx.

Description

Initialise and start the transducer auto refresh acquisition mode

Parameters:

[Query frame layout] **TransducerSelection:** Transducer type selection

[Query frame layout] **ChannelMask:** Mask of the channel to acquire by the auto refresh (1 bit = 1 Channel)

[Query frame layout] **AverageMode:** Set the average mode :

◆ 0: not used

Response frame layout

MODBUS interface description

- ◆ 1: average per channel

[Query frame layout] **AverageValue:** Set the average value (only used, when average is used)

- ◆ 0: average not used
- ◆ max value: 255

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

- ◆ 0 : Trigger disabled
- ◆ 1 : Enable Hardware Digital Input Trigger
- ◆ 2 : Enable Synchro Trigger
- ◆ 4 : Enable Incremental Counter Compare Interrupt

[Query frame layout] **TriggerMode:** Define the trigger mode

- ◆ 1 : One shot trigger : After the software start, the module is waiting for a trigger signal to start the acquisition. After this the trigger signal is ignored.
- ◆ 2 : Sequence trigger : After the software start the module is waiting for the trigger signal and acquires x sequences (also adjustable) and then wait again.

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

- ◆ 1 : Rising front (Only if hardware trigger selected)
- ◆ 2 : Falling front (Only if hardware trigger selected)
- ◆ 3 : Both front (Only if hardware trigger selected)

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

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

[Query frame layout] **ByTriggerNbrOfSeqToAcquire:** define the number of sequences to acquire by each trigger event

[Query frame layout] **DataFormat:** Data format option

- ◆ ***D0 : Time stamp information***

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

- ◆ ***D1 : Data format***

- ◇ 0: Digital value
- ◇ 1: Analog value (in V)

- ◆ ***D2 : invert value***

- ◇ 0 : don't invert the channel value
- ◇ 1 : invert the channel value (-2 mm -> + 2mm)

- ◆ ***D3 : Temperature value***

- ◇ 0 : don't acquire the temperature value
- ◇ 1 : acquire the temperature value

- ◆ ***D4 : Incremental counter value***

- ◇ 0 : don't acquire the incremental counter value

MODBUS interface description

◇ 1 : acquire the incremental counter value

◆ **D5 : Diff. mode**

◇ 0 : Diff mode disabled

◇ 1 : Diff mode enabled : Channel X value = Channel (X) value + Channel (X + 4) value

◆ **D5 : Diff. mode**

◇ 0 : Diff mode disabled

◇ 1 : Diff mode enabled : Channel X value = Channel (X) value + Channel (X + 4) value

[Query frame layout] **Option1:** Reserved

[Query frame layout] **Option2:** Reserved

[Query frame layout] **Option3:** Reserved

[Query frame layout] **Option4:** Reserved

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Transducer selection error
- ◆ -3: The channel mask cannot be null
- ◆ -4: Channel Mask error
- ◆ -5: not available average mode
- ◆ -6: not available average value
- ◆ -7: Trigger source : 2 or more different source cannot be simultaneously be activated
- ◆ -8: Trigger mode selection error
- ◆ -9: Hardware trigger : front definition error
- ◆ -10: Hardware trigger count value not available
- ◆ -11: Nbr of sequence to acquire by trigger mode not available
- ◆ -12: Data format not available
- ◆ -13: Incremental counter not initialised
- ◆ -14: Incremental counter compare logic not initialised
- ◆ -15: Temperature channel not initialised
- ◆ -100: Auto refrseh initialisation and start kernel function error

Query frame layout

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

MODBUS interface description

			copied by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	64	0x4000	0x0040
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1	0x0100	0x0001
word count	2	16-bit integer	28	0x1C00	0x001C
byte count	2	16-bit integer	56	0x3800	0x0038
TransducerSelection	4	32-bit integer	See the description above	0x????????	0x????????
ChannelMask	4	32-bit integer	See the description above	0x????????	0x????????
AverageMode	4	32-bit integer	See the description above	0x????????	0x????????
AverageValue	4	32-bit integer	See the description above	0x????????	0x????????
TriggerMask	4	32-bit integer	See the description above	0x????????	0x????????
TriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
HardwareTriggerEdge	4	32-bit integer	See the description above	0x????????	0x????????
HardwareTriggerCount	4	32-bit integer	See the description above	0x????????	0x????????
ByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????	0x????????
DataFormat	4	32-bit integer	See the description above	0x????????	0x????????
Option1	4	32-bit integer	See the description	0x????????	0x????????

MODBUS interface description

			above		
Option2	4	32-bit integer	See the description above	0x????????	0x????????
Option3	4	32-bit integer	See the description above	0x????????	0x????????
Option4	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1	0x0100	0x0001
word count	2	16-bit integer	28	0x1C00	0x001C

Exception frame layout

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

MODBUS interface description

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

Function MX371x__TransducerInitAndStartAutoRefreshEx

Description

Initialise and start the transducer auto refresh acquisition mode

Parameters:

[Query frame layout] **TransducerSelection:** Transducer type selection

[Query frame layout] **ChannelMask:** Mask of the channel to acquire by the auto refresh (1 bit = 1 Channel)

[Query frame layout] **AverageMode:** Set the average mode :

- ◆ 0: not used
- ◆ 1: average per channel

[Query frame layout] **AverageValue:** Set the average value (only used, when average is used)

- ◆ 0: average not used
- ◆ max value: 255

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

- ◆ 0 : Trigger disabled
- ◆ 1 : Enable Hardware Digital Input Trigger
- ◆ 2 : Enable Synchro Trigger
- ◆ 4 : Enable Incremental Counter Compare Interrupt

[Query frame layout] **TriggerMode:** Define the trigger mode

- ◆ 1 : One shot trigger : After the software start, the module is waiting for a trigger signal to start the acquisition. After this the trigger signal is ignored.
- ◆ 2 : Sequence trigger : After the software start the module is waiting for the trigger signal and acquires x sequences (also adjustable) and then wait again.

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

- ◆ 1 : Rising front (Only if hardware trigger selected)
- ◆ 2 : Falling front (Only if hardware trigger selected)
- ◆ 3 : Both front (Only if hardware trigger selected)

MODBUS interface description

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

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

[Query frame layout] **ByTriggerNbrOfSeqToAcquire:** define the number of sequences to acquire by each trigger event

[Query frame layout] **DataFormat:** Data format option

- ◆ **D0 : Time stamp information**
 - ◇ 0: no time stamp information
 - ◇ 1: time stamp information
- ◆ **D1 : Data format**
 - ◇ 0: Digital value
 - ◇ 1: Analog value (in V)
- ◆ **D2 : invert value**
 - ◇ 0 : don't invert the channel value
 - ◇ 1 : invert the channel value (-2 mm -> + 2mm)
- ◆ **D3 : Temperature value**
 - ◇ 0 : don't acquire the temperature value
 - ◇ 1 : acquire the temperature value
- ◆ **D4 : Incremental counter value**
 - ◇ 0 : don't acquire the incremental counter value
 - ◇ 1 : acquire the incremental counter value
- ◆ **D5 : Diff. mode**
 - ◇ 0 : Diff mode disabled
 - ◇ 1 : Diff mode enabled : Channel X value = Channel (X) value + Channel (X + 4) value
- ◆ **D5 : Diff. mode**
 - ◇ 0 : Diff mode disabled
 - ◇ 1 : Diff mode enabled : Channel X value = Channel (X) value + Channel (X + 4) value

[Query frame layout] **Option1:** Reserved

[Query frame layout] **Option2:** Reserved

[Query frame layout] **Option3:** Reserved

[Query frame layout] **Option4:** Reserved

Returns:

MODBUS interface description

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Transducer selection error
- ◆ -3: The channel mask cannot be null
- ◆ -4: Channel Mask error
- ◆ -5: not available average mode
- ◆ -6: not available average value
- ◆ -7: Trigger source : 2 or more different source cannot be simultaneously be activated
- ◆ -8: Trigger mode selection error
- ◆ -9: Hardware trigger : front definition error
- ◆ -10: Hardware trigger count value not available
- ◆ -11: Nbr of sequence to acquire by trigger mode not available
- ◆ -12: Data format not available
- ◆ -13: Incremental counter not initialised
- ◆ -14: Incremental counter compare logic not initialised
- ◆ -15: Temperature channel not initialised
- ◆ -100: Auto refrseh initialisation and start kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	63	0x3F00	0x003F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1250	0xE204	0x04E2
word count	2	16-bit integer	28	0x1C00	0x001C
byte count	1	8-bit integer	56	0x38	0x38
TransducerSelection	4	32-bit integer	See the description above	0x????????	0x????????
ChannelMask	4	32-bit integer	See the description	0x????????	0x????????

MODBUS interface description

			above		
AverageMode	4	32-bit integer	See the description above	0x????????	0x????????
AverageValue	4	32-bit integer	See the description above	0x????????	0x????????
TriggerMask	4	32-bit integer	See the description above	0x????????	0x????????
TriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
HardwareTriggerEdge	4	32-bit integer	See the description above	0x????????	0x????????
HardwareTriggerCount	4	32-bit integer	See the description above	0x????????	0x????????
ByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????	0x????????
DataFormat	4	32-bit integer	See the description above	0x????????	0x????????
Option1	4	32-bit integer	See the description above	0x????????	0x????????
Option2	4	32-bit integer	See the description above	0x????????	0x????????
Option3	4	32-bit integer	See the description above	0x????????	0x????????
Option4	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1250	0xE204	0x04E2
word count	2	16-bit integer	28	0x1C00	0x001C

Exception frame layout

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

Function

MX371x__TransducerStopAndReleaseAutoRefresh

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerStopAndReleaseAutoRefreshEx.

Description

Stop and release the transducer auto refresh acquisition mode

Parameters:

Exception frame layout

MODBUS interface description

[Query frame layout] **Dummy:** Is not used

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -100: StopAutoRefresh kernel function error

Query frame layout

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

Response frame layout

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

MODBUS interface description

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

Exception frame layout

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

Function

MX371x__TransducerStopAndReleaseAutoRefreshEx

Description

Stop and release the transducer auto refresh acquisition mode

Parameters:

[Query frame layout] **Dummy:** Is not used

Response frame layout

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -100: StopAutoRefresh kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1300	0x1405	0x0514
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
Dummy	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

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

MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1300	0x1405	0x0514
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function MX371x__TransducerInitAndStartSequence

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerInitAndStartSequenceEx.

Description

Initialise and start the transducer sequence acquisition mode

Parameters:

[Query frame layout] **TransducerSelection** : Transducer type selection

Response frame layout

MODBUS interface description

[Query frame layout] **NbrOfChannel** : Number of channel in the sequence

[Query frame layout] **ChannelList** : List of the channel index (0 to MaxChannel-1) who compose the sequence

[Query frame layout] **NbrOfSequence** : Number of sequence to acquire :

- ◆ 0 : continuous mode
- ◆ > 0 : number of sequence

[Query frame layout] **NbrMaxSequenceToTransfer** : This parameter defined the minimal number of sequences to acquire between each send of data by the modul.

Warning : They are two possibilities that the number of sequences sent doesn't reach the minimal number:

- ◆ By the end of the acquisition.
- ◆ If the memory capacity is not big enough.

[Query frame layout] **DelayMode** : Delay Mode :

- ◆ ADDIDATA_DELAY_NOT_USED 0 : Delay is not used.
- ◆ ADDIDATA_DELAY_MODE1_USED 1 : The delay time defines the time between 2 sequence beginnings.

[Query frame layout] **DelayTimeUnit** : Selection of the delay time unit

- ◆ 0: ms
- ◆ 1: s

[Query frame layout] **DelayValue** : Delay Value (max value: 65535)

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

- ◆ 0 : Trigger disabled
- ◆ 1 : Enable Hardware Digital Input Trigger
- ◆ 2 : Enable Synchro Trigger
- ◆ 4 : Enable Incremental Counter Compare Interrupt

[Query frame layout] **TriggerMode**: Define the trigger mode

- ◆ 1 : One shot trigger : After the software start, the module is waiting for a trigger signal to start the acquisition. After this the trigger signal is ignored.
- ◆ 2 : Sequence trigger : After the software start the module is waiting for the trigger signal and acquires x sequences (also adjustable) and then wait again.

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

- ◆ 1 : Rising front (Only if hardware trigger selected)
- ◆ 2 : Falling front (Only if hardware trigger selected)
- ◆ 3 : Both front (Only if hardware trigger selected)

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

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

MODBUS interface description

[Query frame layout] **ByTriggerNbrOfSeqToAcquire:** define the number of sequences to acquire by each trigger event

[Query frame layout] **DataFormat:** Data format option

◆ **D0 : Time stamp information**

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

◆ **D1 : Sequence counter information**

- ◇ 0: no sequence counter
- ◇ 1: sequence counter

◆ **D2 : Data format**

- ◇ 0: Digital value
- ◇ 1: Analog value (in V)

◆ **D3 : invert value**

- ◇ 0 : don't invert the channel value
- ◇ 1 : invert the channel value (-2 mm -> + 2mm)

◆ **D4 : Temperature value**

- ◇ 0 : don't acquire the temperature value
- ◇ 1 : acquire the temperature value

◆ **D5 : Incremental counter value**

- ◇ 0 : don't acquire the incremental counter value
- ◇ 1 : acquire the incremental counter value

◆ **D6 : Diff. mode**

- ◇ 0 : Diff mode disabled
- ◇ 1 : Diff mode enabled : Channel X value = Channel (X) value + Channel (X + 4) value

[Query frame layout] **Option1:** Reserved

[Query frame layout] **Option2:** Reserved

[Query frame layout] **Option3:** Reserved

[Query frame layout] **Option4:** Reserved

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Transducer selection error
- ◆ -3: Number of channel error

MODBUS interface description

- ◆ -4: Channel array selection error
- ◆ -5: Division factor error
- ◆ -6: Incorrect value for Hardware Trigger Mode
- ◆ -7: Incorrect value for Hardware Trigger Front
- ◆ -8: Incorrect value for Synchro Trigger Mode
- ◆ -9: Incorrect value for Hardware Trigger Count
- ◆ -10: Incorrect value for Hardware Trigger filter time
- ◆ -11: Incorrect value for "trigger number of sequences to acquire"
- ◆ -12: Delay Mode selection error
- ◆ -13: Delay time unit selection error
- ◆ -14: Delay value
- ◆ -15: Wrong data format parameter (Option1)
- ◆ -16: A value for Hardware Trigger front was defined but Hardware Trigger Mode is not set
- ◆ -17: Cannot use both triggers at the same time
- ◆ -18: Incorrect value for the hardware trigger stop front
- ◆ -19: Hardware trigger stop can not be used by this configuration of hardware trigger start
- ◆ -100: TransducerInit kernel function error
- ◆ -101: InitConvertTimeDivisionFactor kernel function error
- ◆ -102: InitEnableDisableSequenceDelay kernel function error
- ◆ -103: InitDigitalInputFilter kernel function error
- ◆ -104: InitEnableDisableHardwareTrigger kernel function error
- ◆ -105: InitEnableSynchroTrigger kernel function error
- ◆ -106: DisableSynchroTrigger kernel function error
- ◆ -107: SetTriggerSequenceCount kernel function error
- ◆ -108: InitSequence kernel function error
- ◆ -109: StartStopSequence kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	108	0x6C00	0x006C
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	3	0x0300	0x0003
word count	2	16-bit integer	50	0x3200	0x0032
byte count	2	16-bit	100	0x6400	0x0064

MODBUS interface description

		integer			
TransducerSelection	4	32-bit integer	See the description above	0x????????	0x????????
NbrOfChannel	4	32-bit integer	See the description above	0x????????	0x????????
ChannelList	32	32-bit integer array	See the description above	0x????????[8]	0x????????[8]
NbrOfSequence	4	32-bit integer	See the description above	0x????????	0x????????
NbrMaxSequenceToTransfer	4	32-bit integer	See the description above	0x????????	0x????????
DelayMode	4	32-bit integer	See the description above	0x????????	0x????????
DelayTimeUnit	4	32-bit integer	See the description above	0x????????	0x????????
DelayValue	4	32-bit integer	See the description above	0x????????	0x????????
TriggerMask	4	32-bit integer	See the description above	0x????????	0x????????
TriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
HardwareTriggerEdge	4	32-bit integer	See the description above	0x????????	0x????????
HardwareTriggerCount	4	32-bit integer	See the description above	0x????????	0x????????
ByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????	0x????????
DataFormat	4	32-bit integer	See the description above	0x????????	0x????????
Option1	4	32-bit integer	See the description above	0x????????	0x????????
Option2	4	32-bit integer	See the description above	0x????????	0x????????

MODBUS interface description

Option3	4	32-bit integer	See the description above	0x????????	0x????????
Option4	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	3	0x0300	0x0003
word count	2	16-bit integer	50	0x3200	0x0032

Exception frame layout

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

MODBUS interface description

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

Function MX371x__TransducerInitAndStartSequenceEx

Description

Initialise and start the transducer sequence acquisition mode

Parameters:

[Query frame layout] **TransducerSelection** : Transducer type selection

[Query frame layout] **NbrOfChannel** : Number of channel in the sequence

[Query frame layout] **ChannelList** : List of the channel index (0 to MaxChannel-1) who compose the sequence

[Query frame layout] **NbrOfSequence** : Number of sequence to acquire :

- ◆ 0 : continuous mode
- ◆ > 0 : number of sequence

[Query frame layout] **NbrMaxSequenceToTransfer** : This parameter defined the minimal number of sequences to acquire between each send of data by the modul.

Warning : They are two possibilities that the number of sequences sent doesn't reach the minimal number:

- ◆ By the end of the acquisition.
- ◆ If the memory capacity is not big enough.

[Query frame layout] **DelayMode** : Delay Mode :

- ◆ ADDIDATA_DELAY_NOT_USED 0 : Delay is not used.
- ◆ ADDIDATA_DELAY_MODE1_USED 1 : The delay time defines the time between 2 sequence beginnings.

[Query frame layout] **DelayTimeUnit** : Selection of the delay time unit

- ◆ 0: ms
- ◆ 1: s

[Query frame layout] **DelayValue** : Delay Value (max value: 65535)

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

- ◆ 0 : Trigger disabled
- ◆ 1 : Enable Hardware Digital Input Trigger
- ◆ 2 : Enable Synchro Trigger
- ◆ 4 : Enable Incremental Counter Compare Interrupt

[Query frame layout] **TriggerMode**: Define the trigger mode

MODBUS interface description

- ◆ 1 : One shot trigger : After the software start, the module is waiting for a trigger signal to start the acquisition. After this the trigger signal is ignored.
- ◆ 2 : Sequence trigger : After the software start the module is waiting for the trigger signal and acquires x sequences (also adjustable) and then wait again.

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

- ◆ 1 : Rising front (Only if hardware trigger selected)
- ◆ 2 : Falling front (Only if hardware trigger selected)
- ◆ 3 : Both front (Only if hardware trigger selected)

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

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

[Query frame layout] **ByTriggerNbrOfSeqToAcquire:** define the number of sequences to acquire by each trigger event

[Query frame layout] **DataFormat:** Data format option

◆ **D0 : Time stamp information**

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

◆ **D1 : Sequence counter information**

- ◇ 0: no sequence counter
- ◇ 1: sequence counter

◆ **D2 : Data format**

- ◇ 0: Digital value
- ◇ 1: Analog value (in V)

◆ **D3 : invert value**

- ◇ 0 : don't invert the channel value
- ◇ 1 : invert the channel value (-2 mm -> + 2mm)

◆ **D4 : Temperature value**

- ◇ 0 : don't acquire the temperature value
- ◇ 1 : acquire the temperature value

◆ **D5 : Incremental counter value**

- ◇ 0 : don't acquire the incremental counter value
- ◇ 1 : acquire the incremental counter value

◆ **D6 : Diff. mode**

- ◇ 0 : Diff mode disabled
- ◇ 1 : Diff mode enabled : Channel X value = Channel (X) value + Channel (X + 4) value

[Query frame layout] **Option1:** Reserved

MODBUS interface description

[Query frame layout] **Option2:** Reserved

[Query frame layout] **Option3:** Reserved

[Query frame layout] **Option4:** Reserved

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Transducer selection error
- ◆ -3: Number of channel error
- ◆ -4: Channel array selection error
- ◆ -5: Division factor error
- ◆ -6: Incorrect value for Hardware Trigger Mode
- ◆ -7: Incorrect value for Hardware Trigger Front
- ◆ -8: Incorrect value for Synchro Trigger Mode
- ◆ -9: Incorrect value for Hardware Trigger Count
- ◆ -10: Incorrect value for Hardware Trigger filter time
- ◆ -11: Incorrect value for "trigger number of sequences to acquire"
- ◆ -12: Delay Mode selection error
- ◆ -13: Delay time unit selection error
- ◆ -14: Delay value
- ◆ -15: Wrong data format parameter (Option1)
- ◆ -16: A value for Hardware Trigger front was defined but Hardware Trigger Mode is not set
- ◆ -17: Cannot use both triggers at the same time
- ◆ -18: Incorrect value for the hardware trigger stop front
- ◆ -19: Hardware trigger stop can not be used by this configuration of hardware trigger start
- ◆ -100: TransducerInit kernel function error
- ◆ -101: InitConvertTimeDivisionFactor kernel function error
- ◆ -102: InitEnableDisableSequenceDelay kernel function error
- ◆ -103: InitDigitalInputFilter kernel function error
- ◆ -104: InitEnableDisableHardwareTrigger kernel function error
- ◆ -105: InitEnableSynchroTrigger kernel function error
- ◆ -106: DisableSynchroTrigger kernel function error
- ◆ -107: SetTriggerSequenceCount kernel function error
- ◆ -108: InitSequence kernel function error
- ◆ -109: StartStopSequence kernel function error

Query frame layout

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

MODBUS interface description

protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	107	0x6B00	0x006B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1350	0x4605	0x0546
word count	2	16-bit integer	50	0x3200	0x0032
byte count	1	8-bit integer	100	0x64	0x64
TransducerSelection	4	32-bit integer	See the description above	0x????????	0x????????
NbrOfChannel	4	32-bit integer	See the description above	0x????????	0x????????
ChannelList	32	32-bit integer array	See the description above	0x????????[8]	0x????????[8]
NbrOfSequence	4	32-bit integer	See the description above	0x????????	0x????????
NbrMaxSequenceToTransfer	4	32-bit integer	See the description above	0x????????	0x????????
DelayMode	4	32-bit integer	See the description above	0x????????	0x????????
DelayTimeUnit	4	32-bit integer	See the description above	0x????????	0x????????
DelayValue	4	32-bit integer	See the description above	0x????????	0x????????
TriggerMask	4	32-bit integer	See the description above	0x????????	0x????????
TriggerMode	4	32-bit integer	See the description above	0x????????	0x????????
HardwareTriggerEdge	4	32-bit integer	See the description above	0x????????	0x????????
HardwareTriggerCount	4	32-bit integer	See the description	0x????????	0x????????

MODBUS interface description

			above		
ByTriggerNbrOfSeqToAcquire	4	32-bit integer	See the description above	0x????????	0x????????
DataFormat	4	32-bit integer	See the description above	0x????????	0x????????
Option1	4	32-bit integer	See the description above	0x????????	0x????????
Option2	4	32-bit integer	See the description above	0x????????	0x????????
Option3	4	32-bit integer	See the description above	0x????????	0x????????
Option4	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1350	0x4605	0x0546
word count	2	16-bit integer	50	0x3200	0x0032

Exception frame layout

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

Function MX371x__TransducerStopAndReleaseSequence

For new application(s) or automate communication it is recommended to use the function MX371x__TransducerStopAndReleaseSequenceEx.

Description

Stop and release the transducer sequence acquisition mode

Parameters:

[Query frame layout] **Dummy:** Is not used

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -100: StartStopSequence kernel function error

Query frame layout

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

MODBUS interface description

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

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	4	0x0400	0x0004
word count	2		2	0x0200	0x0002

Query frame layout

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

Exception frame layout

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

Function

MX371x__TransducerStopAndReleaseSequenceEx

Description

Stop and release the transducer sequence acquisition mode

Parameters:

[Query frame layout] **Dummy:** Is not used

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -100: StartStopSequence kernel function error

Query frame layout

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

Response frame layout

MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1400	0x7805	0x0578
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
Dummy	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1400	0x7805	0x0578
word count	2		2	0x0200	0x0002

Query frame layout

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

Exception frame layout

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

Function MSXE371x__ExternalTemperatureInit

For new application(s) or automate communication it is recommended to use the function MSXE371x__ExternalTemperatureInitEx.

Description

Initialise the external temperature

Parameters:

[Query frame layout] **ConnectedTempSensor** : Connected sensor type

- ◆ 100 for PT100
- ◆ 500 for PT500
- ◆ 1000 for PT1000

[Query frame layout] **ConvertMode** : Converting mode

- ◆ 0: Disabled. Return the value into the mOhm form
- ◆ 1: oC
- ◆ 2: oF

[Query frame layout] **GainSelection** : Gain selection (0 for auto mode, 1, 2, 4, 8, 16, 32, 64 or 128)

MODBUS interface description

[Query frame layout] **FrequencySelection** : Acquisition frequency selection (10, 20, 30, 40, 50, 60, 80, 100, 120, 200, 300, 400, 600, 800, 1200 or 2400)

[Query frame layout] **PowerSaveMode** : Power save mode selection

- ◆ 0 : Disable the power save mode
- ◆ 1 : Enable the power save mode. Value can only be read if any transducer acquisitions enabled

[Query frame layout] **Option01** : Set it to 0

[Query frame layout] **Option02** : Set it to 0

[Query frame layout] **Option03** : Set it to 0

[Query frame layout] **Option04** : Set it to 0

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Connected temperature sensor selection error
- ◆ -3: Convert mode selection error
- ◆ -4: Gain selection error
- ◆ -5: Frequency selection error
- ◆ -6: Power save mode selection error
- ◆ -100: Init external temperature kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	44	0x2C00	0x002C
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	5	0x0500	0x0005
word count	2		18	0x1200	0x0012

MODBUS interface description

		16-bit integer			
byte count	2	16-bit integer	36	0x2400	0x0024
ConnectedTempSensor	4	32-bit integer	See the description above	0x????????	0x????????
ConvertMode	4	32-bit integer	See the description above	0x????????	0x????????
GainSelection	4	32-bit integer	See the description above	0x????????	0x????????
FrequencySelection	4	32-bit integer	See the description above	0x????????	0x????????
PowerSaveMode	4	32-bit integer	See the description above	0x????????	0x????????
Option01	4	32-bit integer	See the description above	0x????????	0x????????
Option02	4	32-bit integer	See the description above	0x????????	0x????????
Option03	4	32-bit integer	See the description above	0x????????	0x????????
Option04	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	5	0x0500	0x0005
word count	2	16-bit integer	18	0x1200	0x0012

Exception frame layout

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

Function MSXE371x__ExternalTemperatureInitEx

Description

Initialise the external temperature

Parameters:

[Query frame layout] **ConnectedTempSensor** : Connected sensor type

- ◆ 100 for PT100
- ◆ 500 for PT500
- ◆ 1000 for PT1000

[Query frame layout] **ConvertMode** : Converting mode

- ◆ 0: Disabled. Return the value into the mOhm form
- ◆ 1: oC
- ◆ 2: oF

[Query frame layout] **GainSelection** : Gain selection (0 for auto mode, 1, 2, 4, 8, 16, 32, 64 or 128)

Response frame layout

MODBUS interface description

[Query frame layout] **FrequencySelection** : Acquisition frequency selection (10, 20, 30, 40, 50, 60, 80, 100, 120, 200, 300, 400, 600, 800, 1200 or 2400)

[Query frame layout] **PowerSaveMode** : Power save mode selection

- ◆ 0 : Disable the power save mode
- ◆ 1 : Enable the power save mode. Value can only be read if any transducer acquisitions enabled

[Query frame layout] **Option01** : Set it to 0

[Query frame layout] **Option02** : Set it to 0

[Query frame layout] **Option03** : Set it to 0

[Query frame layout] **Option04** : Set it to 0

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Connected temperature sensor selection error
- ◆ -3: Convert mode selection error
- ◆ -4: Gain selection error
- ◆ -5: Frequency selection error
- ◆ -6: Power save mode selection error
- ◆ -100: Init external temperature kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	43	0x2B00	0x002B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1450	0xAA05	0x05AA
word count	2		18	0x1200	0x0012

MODBUS interface description

		16-bit integer			
byte count	1	8-bit integer	36	0x24	0x24
ConnectedTempSensor	4	32-bit integer	See the description above	0x????????	0x????????
ConvertMode	4	32-bit integer	See the description above	0x????????	0x????????
GainSelection	4	32-bit integer	See the description above	0x????????	0x????????
FrequencySelection	4	32-bit integer	See the description above	0x????????	0x????????
PowerSaveMode	4	32-bit integer	See the description above	0x????????	0x????????
Option01	4	32-bit integer	See the description above	0x????????	0x????????
Option02	4	32-bit integer	See the description above	0x????????	0x????????
Option03	4	32-bit integer	See the description above	0x????????	0x????????
Option04	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

Query frame layout

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1450	0xAA05	0x05AA
word count	2	16-bit integer	18	0x1200	0x0012

Exception frame layout

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

Function MSXE371x__ExternalTemperatureRelease

For new application(s) or automate communication it is recommended to use the function MSXE371x__ExternalTemperatureReleaseEx.

Description

Release the external temperature

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	12	0x0C00	0x000C
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	6	0x0600	0x0006
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
Dummy	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

MODBUS interface description

Function code		integer			
Reference number (=register)	2	16-bit integer	6	0x0600	0x0006
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function MSXE371x__ExternalTemperatureReleaseEx

Description

Release the external temperature

Returns:

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

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

Query frame layout

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

Response frame layout

MODBUS interface description

transaction identifier		16-bit integer	User defined - copied by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1500	0xDC05	0x05DC
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
Dummy	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1500	0xDC05	0x05DC
word count	2		2	0x0200	0x0002

Query frame layout

MODBUS interface description

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

Exception frame layout

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

Function MX371x__SetTransducerDatabaseCursor

For new application(s) or automate communication it is recommended to use the function MX371x__SetTransducerDatabaseCursorEx.

Description

Change the active transducer database cursor

Parameters:

[Query frame layout] **TransducerDatabaseCursor**: New cursor value. This is an integer from 0 .. (NumberOfTransducerTypes-1)

Returns:

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	12	0x0C00	0x000C
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	7	0x0700	0x0007
word count	2	16-bit integer	2	0x0200	0x0002
byte count	2	16-bit integer	4	0x0400	0x0004
TransducerDatabaseCursor	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

MODBUS interface description

Reference number (=register)	2	16-bit integer	7	0x0700	0x0007
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function MX371x__SetTransducerDatabaseCursorEx

Description

Change the active transducer database cursor

Parameters:

[Query frame layout] **TransducerDatabaseCursor:** New cursor value. This is an integer from 0 .. (NumberOfTransducerTypes-1)

Returns:

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

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

Query frame layout

Field		Type	Value		
-------	--	------	-------	--	--

Response frame layout

MODBUS interface description

	Size (Bytes)			little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1504	0xE005	0x05E0
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
TransducerDatabaseCursor	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1504	0xE005	0x05E0

word count	2	16-bit integer	2	0x0200	0x0002
------------	---	----------------	---	--------	--------

Exception frame layout

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

Function MSXE371x__IncCounterInit

Description

Initialise the counter

[Query frame layout]

- **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)

[Query frame layout]

- **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)

[Query frame layout]

- **ulOption01** : Set it to 0 [Query frame layout]
- **ulOption02** : Set it to 0 [Query frame layout]
- **ulOption03** : Set it to 0 [Query frame layout]
- **ulOption04** : Set it to 0

Response frame layout

MODBUS interface description

[Response frame layout]

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

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Counter mode selection error
- -3: Counter option selection error
- -100: Init counter kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	31	0x1F00	0x001F
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1650	0x7206	0x0672
word count	2	16-bit integer	12	0x0C00	0x000C
byte count	1	8-bit integer	24	0x18	0x18
ulCounterMode	4	32-bit integer	See the description above	0x???????	0x???????
ulCounterOption	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????
ulOption02	4	32-bit integer	See the description above	0x???????	0x???????
ulOption03	4	32-bit integer	See the description above	0x???????	0x???????

MODBUS interface description

ulOption04	4	32-bit integer	See the description above	0x????????	0x????????
------------	---	----------------	---------------------------	------------	------------

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1650	0x7206	0x0672
word count	2	16-bit integer	12	0x0C00	0x000C

Exception frame layout

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

code		integer	corresponding chapter		
------	--	---------	--------------------------	--	--

Function MSXE371x__IncCounterRelease

Description

Release the counter

[Query frame layout]

- **ulOption01** : Set it to 0

[Response frame layout]

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

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1700	0xA406	0x06A4
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1700	0xA406	0x06A4
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function MSXE371x__IncCounterClear

Description

Clear the 32 bits counter

[Query frame layout]

- **ulOption01** : Set it to 0

[Response frame layout]

- **Possible return value on the remote system (read them with GetLastErrorStatusEx):**

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

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1750	0xD606	0x06D6
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1750	0xD606	0x06D6
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function MSXE371x__IncCounterWrite32BitValue

Description

write a 32 bits counter value

[Query frame layout]

- **ulCounterValue** : Counter value

[Response frame layout]

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

- 0: means the remote function performed OK
- -1: means an system error occurred
- -2: Counter value error
- -100: Kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1800	0x0807	0x0708
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulCounterValue	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1800	0x0807	0x0708
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

Function

MSXE371x__IncCounterInitAndEnableCompareLogic

Description

Init and enable a counter compare value

Parameters:

[Query frame layout] **ulValue** / **ulValueLow** : compare value (double). ulValueLow

[Query frame layout] **ulMode** : compare mode

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

[Query frame layout] **ulSynchroTrigger**

- ◆ 0 : no synchro trigger
- ◆ 1 : generates a synchro trigger when condition is true

[Query frame layout] **ulOption01** : set it to 0

[Query frame layout] **ulOption02** : set it to 0

Returns:

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

- ◆ 0: means the remote function performed OK
- ◆ -1: means an system error occurred
- ◆ -2: Compare value error
- ◆ -3: Compare mode error
- ◆ -4: Synchro trigger error
- ◆ -100: Init and enable counter compare kernel function error

Query frame layout

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

MODBUS interface description

MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1850	0x3A07	0x073A
word count	2	16-bit integer	10	0x0A00	0x000A
byte count	1	8-bit integer	20	0x14	0x14
ulValue	4	32-bit integer	See the description above	0x???????	0x???????
ulMode	4	32-bit integer	See the description above	0x???????	0x???????
ulSynchroTrigger	4	32-bit integer	See the description above	0x???????	0x???????
ulOption01	4	32-bit integer	See the description above	0x???????	0x???????
ulOption02	4	32-bit integer	See the description above	0x???????	0x???????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1850	0x3A07	0x073A
word count	2		10	0x0A00	0x000A

MODBUS interface description

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

Exception frame layout

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

Function

MSXE371x__IncCounterDisableAndReleaseCompareLogic

Description

Disable and Release a counter compare value

Parameters:

[Query frame layout]

- **ulOption01** : Set it to 0

Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
 - ◆ 0: means the remote function performed OK
 - ◆ -1: means an system error occurred
 - ◆ -100: Disable counter compare value kernel function error

Query frame layout

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

MODBUS interface description

transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1900	0x6C07	0x076C
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	1900	0x6C07	0x076C
word count	2		2	0x0200	0x0002

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

Exception frame layout

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

Function MSXE371x__IncCounterInitAndEnableIndex

Description

Init and Enable the counter index

Parameters:

[Query frame layout] **ulReferenceAction** : Reference action

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

[Query frame layout] **ulIndexOperation** : Index operation

- ◆ 0: latch and clear counter at low edge
- ◆ 1: latch and clear counter at high edge

[Query frame layout] **ulAutoMode** : Auto mode

- ◆ 0: do not use automode (action is done only once)
- ◆ 1: use automode (action is done continuously)

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

Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
 - ◆ 0: Means the remote function performed OK

MODBUS interface description

- ◆ -1: Means an system error occurred
- ◆ -2: Reference action selection error
- ◆ -3: Index operation selection error
- ◆ -4: Automode selection error
- ◆ -5: Index logic already initialized
- ◆ -100: Disable counter compare value kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	23	0x1700	0x0017
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2000	0xD007	0x07D0
word count	2	16-bit integer	8	0x0800	0x0008
byte count	1	8-bit integer	16	0x10	0x10
ulReferenceAction	4	32-bit integer	See the description above	0x????????	0x????????
ulIndexOperation	4	32-bit integer	See the description above	0x????????	0x????????
ulAutoMode	4	32-bit integer	See the description above	0x????????	0x????????
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

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

MODBUS interface description

transaction identifier		16-bit integer	User defined - copied by server - usually 0		
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2000	0xD007	0x07D0
word count	2	16-bit integer	8	0x0800	0x0008

Exception frame layout

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

Function MSXE371x__IncCounterDisableAndReleaseIndex

Description

Disable and Release the counter index

Parameters:

[Query frame layout]

- **ulOption01** : Set it to 0

Returns:

- **Possible return value on the remote system (read them with GetLastCommandStatusEx):**
 - ◆ 0: Means the remote function performed OK
 - ◆ -1: Means an system error occurred
 - ◆ -2: Index logic already initialized
 - ◆ -100: Disable counter compare value kernel function error

Query frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	11	0x0B00	0x000B
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2100	0x3408	0x0834
word count	2	16-bit integer	2	0x0200	0x0002
byte count	1	8-bit integer	4	0x04	0x04
ulOption01	4	32-bit integer	See the description above	0x????????	0x????????

Response frame layout

Field	Size (Bytes)	Type	Value	little endian (Intel)	big endian (Motorola)
transaction identifier	2	16-bit integer	User defined - copied by server - usually 0	0x0000	0x0000
protocol identifier	2	16-bit integer	0	0x0000	0x0000
length	2	16-bit integer	6	0x0600	0x0006
unit identifier	1	8-bit integer	0 or 1	0x00 or 0x01	0x00 or 0x01
MODBUS Function code	1	8-bit integer	0x10	0x10	0x10
Reference number (=register)	2	16-bit integer	2100	0x3408	0x0834
word count	2	16-bit integer	2	0x0200	0x0002

Exception frame layout

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

FC23 (read/write registers) Functions

[Top](#)

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

Query frame layout

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

Response frame layout

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

Exception frame layout

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

Exception code description

[Top](#)

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

Siemens Step 7 compatibility information (AWL/SDF code)

[Top](#)

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

Function/Parameter	Renamed as
MXCommon__GetModuleType	GetModuleType
MXCommon__GetTime	GetTime
MXCommon__TestCustomerID	TestCustomerID
MX371x__getNumberOfChannels	371x_GetNbrOfChannels
MX371x__TransducerGetAutoRefreshValues	371x_GetAutoRefVal
MSXE371x__ExternalTemperatureRead	371x_ExtTempRead
MX371x__TransducerGetNbrOfType	371x_GetNbrOfType
MX371x__GetTransducerDatabaseCursor	371x_GetDataBaseCursor
TransducerDatabaseCursor	TransducerDBCursor
MX371x__TransducerGetTypeInformation	371x_GetTypeInfo
Type	TransducerType
MX371x__TransducerInitPrimaryConnectionTest	MX371x_TransInitPrCoTe
TransducerInitPrimaryConnection	TransducerInitPrimConn
MX371x__TransducerTestPrimaryConnection	MX371x_TransTestPrCo
MX371x__TransducerTestPrimaryShortCircuit	MX371x_TransTestPrShCi
MX371x__TransducerRearmPrimary	MX371x_TransRearPrim
MX371x__TransducerTestSecondaryConnection0	MX371x_TransTestSeC0
MX371x__TransducerTestSecondaryConnection1	MX371x_TransTestSeC1
MX371x__TransducerTestSecondaryConnection2	MX371x_TransTestSeC2
MX371x__TransducerTestSecondaryConnection3	MX371x_TransTestSeC3
MX371x__TransducerTestSecondaryConnection4	MX371x_TransTestSeC4
MX371x__TransducerTestSecondaryConnection5	MX371x_TransTestSeC5
MX371x__TransducerTestSecondaryConnection6	MX371x_TransTestSeC6
MX371x__TransducerTestSecondaryConnection7	MX371x_TransTestSeC7
MX371x__TransducerTestSecondaryShortCircuit0	MX371x_TransTstSeShC0
MX371x__TransducerTestSecondaryShortCircuit1	MX371x_TransTstSeShC1
MX371x__TransducerTestSecondaryShortCircuit2	MX371x_TransTstSeShC2
MX371x__TransducerTestSecondaryShortCircuit3	MX371x_TransTstSeShC3
MX371x__TransducerTestSecondaryShortCircuit4	MX371x_TransTstSeShC4
MX371x__TransducerTestSecondaryShortCircuit5	MX371x_TransTstSeShC5
MX371x__TransducerTestSecondaryShortCircuit6	MX371x_TransTstSeShC6
MX371x__TransducerTestSecondaryShortCircuit7	MX371x_TransTstSeShC7
MSXE371x__InternalTemperatureRead	371x_IntTempRead
MSXE371x__IncCounterRead32BitValue	371xIncCntRead32
MXCommon__SetHardwareTriggerFilterTime	SetHwTrigFiltTime
MXCommon__InitAndStartSynchroTimer	InitStartSyncTimer

MODBUS interface description

MXCommon__StopAndReleaseSynchroTimer	StopRelSyncTimer
MXCommon__Reboot	Reboot
MXCommon__SetCustomerKey	SetCustomerKey
MXCommon__SetFilterChannels	SetFilterChannels
MX371x__TransducerInitAndStartAutoRefresh	371x_InitStartAutoRef
HardwareTriggerEdge	HwTrigCount
ByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSeqToAcq
MX371x__TransducerStopAndReleaseAutoRefresh	371x_StopRelAutoRef
MX371x__TransducerInitAndStartSequence	371x_InitStartSeq
NbrMaxSequenceToTransfer	NbrMaxSeqToTransfer
HardwareTriggerCount	HwTrigCount
ByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSeqToAcq
MX371x__TransducerStopAndReleaseSequence	371x_StopRelSeq
MSXE371x__ExternalTemperatureInit	371x_SetDataBaseCursor
MSXE371x__ExternalTemperatureRelease	MX371x_ExtTempRelease
MX371x__SetTransducerDatabaseCursor	MX371x_SetTransDbCurs
TransducerDatabaseCursor	TransducerDBCursor
MSXE371x__IncCounterInit	371xIncCntInit
MSXE371x__IncCounterRelease	371xIncCntRelease
MSXE371x__IncCounterClear	371xIncCntClear
MSXE371x__IncCounterWrite32BitValue	371xIncCntWrite32
MSXE371x__IncCounterInitAndEnableCompareLogic	371xIncCntInitCL
MSXE371x__IncCounterDisableAndReleaseCompareLogic	371xIncCntReleaseCL
MSXE371x__IncCounterInitAndEnableIndex	371xIncCntInitI
MSXE371x__IncCounterDisableAndReleaseIndex	371xIncCntReleaseI