MSX-E171x soap api functions

Generated by Doxygen 1.7.1

Tue May 3 2016 10:14:18

Contents

Chapter 1

MSX-E171x SOAP functions documentation

MainRevision:

1.1 Introduction

The module is accessed via a TCP/IP socket:

The Ethernet I/O module has the following two servers: Command server (SOAP) > to send commands (initialisation, etc.)

Data server (TCP socket) > to obtain the values of the acquisition

Event server (TCP socket) > to obtain event from the module

MSX-E171x server access information:

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

See the "Modules" chapter to view the functions

Chapter 2

Module Documentation

2.1 MSX-E17xx functions

Modules

- MSX-E17xx multifunction functions
- MSX-E17xx digital I/O functions
- MSX-E17xx IO watchdog functions

2.2 MSX-E17xx multifunction functions

Modules

- MSX-E17xx multifunction common functions
- MSX-E17xx Sinus Cosinus

2.3 Common functions

Modules

- Common general functions

  Various utility functions, mainly to identify a remote system.

- Common temperature functions

  These functions deals with the internal temperature sub-system.

- Common hardware trigger functions

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

- Common security functions

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

- Common time functions

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

- Common I/O auto configuration functions

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

- Common synchronisation timer functions

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

- Set/Backup/Restore general system configuration

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

- System state management

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

- Customer option management

  Enable to get informations about the options of the system.

- Synchronisation management

  Manage the synchronisation state of the system.

- input filter Filter management

  Manages the analog input filters in the system.

2.4 Common general functions

Various utility functions, mainly to identify a remote system.

Functions

- int MXCommon__GetModuleType (void *_, struct MXCommon__ByteArrayResponse *Response)

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

- int MXCommon__GetHostname (void *_, struct MXCommon__ByteArrayResponse *Response)

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

- int MXCommon__SetHostname (struct xsd__base64Binary *bHostname, struct MXCommon__-Response *Response)

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

- int MXCommon__GetClientConnections (void *_, struct MXCommon__ByteArrayResponse *Response)

This function return the client connection list.

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

  Call the libc strerror() on the remote device (actually this is a call to strerror_r()).

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

  Ask the MSX-E module to reboot.

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

  Reset the I/O functionalities of the MSX-E system.

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

  Restart the data-server service.

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

  Get MSX-E Ethernet links states.

2.4.1 Function Documentation

2.4.1.1 int MXCommon__GetModuleType (void * _, struct MXCommon__ByteArrayResponse * *Response*)

Parameters

`[in]` _ : no input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.4.1.2 int MXCommon__GetHostname (void * _, struct MXCommon__ByteArrayResponse * *Response*)

Parameters

`[in]` _ : no input parameter

`[out]` ***Response*** • sArray : Hostname of the module
- iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)
- syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.4.1.3 int MXCommon__SetHostname (struct xsd__base64Binary * *bHostname*, struct MXCommon__Response * *Response*)

Parameters

`[in]` ***bHostname*** : Hostname

`[out]` ***Response*** • iReturnValue : Return value
- 0 : success
- -1: system error (see syserrno)

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.4.1.4 int MXCommon__GetClientConnections (void * _, struct MXCommon__ByteArrayResponse * *Response*)

Parameters

`[in]` _ : no input parameter

`[out]` ***Response*** • sArray : string containing the list of connected clients.
• sResponse Composed of iReturnValue and syserrno

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

Sample: 172.16.3.43:8989-5555 172.16.3.200:8989

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

Parameters

`[in]` ***errnum*** : Error number

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

```
STRERROR(3)                                                Linux Programmer's Manual
STRERROR(3)

NAME
strerror, strerror_r - return string describing error code

SYNOPSIS
#include <string.h>

char *strerror(int errnum);

#define _XOPEN_SOURCE 600
#include <string.h>

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

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

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

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

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

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

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

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

is a GNU extension used by glibc (since 2.0), and must be regarded as obsolete
in view of SUSv3.
The GNU version may, but need not, use the user-supplied buffer.
If it does, the result may be truncated in case the supplied buffer is too small.
The result is always NUL-terminated.

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

`[in]` _ : no input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

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

Parameters

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

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

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

Return values

0 SOAP_OK

Others See SOAP error

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

Parameters

`[in]` ***ulAction*** : action
- 0: normal restart
- 1: with cache file reset

- 2: with cache file deletion

[in] ***ulOption*** : Reserved

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

Note

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

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

Parameters

[in] _ : no input parameter

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

sResponse.iReturnValue

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

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

sPort0: Fisrt port informations

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

sPort1: Second port informations

- **ulState**
  - **0** Link down
  - **1** Link up
- **ulSpeed**

  - **10** 10 Mb/s
  - **100** 100 Mb/s
- **ulDuplex**
  - **0** Half duplex
  - **1** Full duplex
- **ulInfo1** Reserverd
- **ulInfo2** Reserverd

Return values

0 SOAP_OK

Others See SOAP error

2.5 Common temperature functions

These functions deals with the internal temperature sub-system.

Data Structures

- struct MXCommon__GetModuleTemperatureValueAndStatusResponse

Functions

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

  Read the temperature on the module.

- int MXCommon__SetModuleTemperatureWarningLevels (xsd__double dMinimalWarningLevel, xsd__double dMaximalWarningLevel, xsd__unsignedLong ulOption, struct MXCommon__-Response *Response)

  Set the temperature warning level on the module.

2.5.1 Detailed Description

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

2.5.2 Function Documentation

2.5.2.1 int MXCommon__GetModuleTemperatureValueAndStatus (xsd__unsignedLong *ulOption*, struct MXCommon__GetModuleTemperatureValueAndStatusResponse * *Response*)

Parameters

`[in]` ***ulOption*** : Reserved

[out] ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().
  - dValue : Temperature value in Degree Celsius

• ulTemperatureStatus : Temperature Status :
  - TEMPERATURE_INITIAL = 0 : Temperature not ready
  - TEMPERATURE_TOOLOW = 1 : Temperature too low !
  - TEMPERATURE_LOW = 2 : Temperature under the min warning value
  - TEMPERATURE_NOMINAL = 3 : Temperature in the nominal range
  - TEMPERATURE_HIGH = 4 : Temperature over the max warning value
  - TEMPERATURE_TOOHIGH = 5 : Temperature too high !

• ulInfo : Reserved

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

[in] ***dMinimalWarningLevel*** : Minimal temperature warning level in Degree : 5 to 60 Degree Celsius

[in] ***dMaximalWarningLevel*** : Maximal temperature warning level in Degree : 5 to 60 Degree Celsius

[in] ***ulOption*** : Reserved

[out] ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.6 Common hardware trigger functions

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

Data Structures

- struct MXCommon__GetHardwareTriggerFilterTimeResponse
- struct MXCommon__GetHardwareTriggerStateResponse

Functions

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

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

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

  Get the filter time for the hardware trigger input.

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

  Get the hardware trigger state after the filter.

2.6.1 Function Documentation

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

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

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

Parameters

`[in]` ***ulFilterTime*** Filter time for the hardware trigger input in steps of 250ns (max value : 65535).

- **0**: Disable the filter
- **1**: Sets the filter time to 250 ns
- **2**: Sets the filter time to 500 ns
- ...
- **65535**: Sets the filter time to 16 ms

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

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

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

Return values

0 SOAP_OK

Others See SOAP error

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

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

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

Parameters

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

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

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

Return values

0 SOAP_OK

Others See SOAP error

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

Parameters

`[in]` ***ulOption*** : Reserved

`[out]` ***Response***

- ulState : Hardware trigger input state.
  - 0: Hardware trigger input is low
  - 1: Hardware trigger input is high.
- sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.7 Common security functions

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

Data Structures

- struct MXCommon__TestCustomerIDResponse

Functions

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

  Set the Customer key.

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

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

2.7.1 Detailed Description

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

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

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

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

Detailed presentation of operations:

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

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

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

The user receives then two arrays of 16 bytes :

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

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

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

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

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

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

2.7.2 Function Documentation

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

Parameters

`[in]` ***bKey*** : Customer key (only writable on the module) [32 bytes containing a AES key]

`[in]` ***bPublicKey*** : IV (Initialisation vector) for the AES cryptography [16 bytes containing a AES key]

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.7.2.2 int MXCommon__TestCustomerID (void * _, struct MXCommon__TestCustomerIDResponse * *Response*)

Parameters

`[in]` _ : No Input

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

• bValueArray : non encrypted value array [16 bytes of random data]

• bCryptedValueArray : Encrypted value array [16 bytes of the encrypted random data]

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.8 Common time functions

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

Data Structures

- struct MXCommon__GetTimeResponse
- struct MXCommon__GetUpTimeResponse

Functions

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

2.8.1 Detailed Description

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

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

2.8.2 Function Documentation

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

Parameters

`[in]` ***ulLowTime*** : Number of microseconds since the begin of the second

`[in]` ***ulHighTime*** : Number of seconds since the Epoch (1st January,1970)

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.8.2.2 int MXCommon__SysToHardwareClock (void * _, struct MXCommon__Response * *Response*)

Parameters

[in] _ No input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

2.8.2.3 int MXCommon__HardwareClockToSys (void * _, struct MXCommon__Response * *Response*)

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

Parameters

[in] _ No input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

Parameters

[in] _ : No input parameter

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

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

`[in]` _ : no input parameter

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().
- ulUpTime : Number of seconds since the last boot of the system.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.9 Common I/O auto configuration functions

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

Data Structures

- struct MXCommon__GetAutoConfigurationFileResponse

Functions

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

  Get the auto configuration file of the module.

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

  Set the auto configuration file of the module.

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

  start/Restart the auto configuration

2.9.1 Detailed Description

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

This set of functions allows to:

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

2.9.2 Function Documentation

2.9.2.1 int MXCommon__GetAutoConfigurationFile (void * _, struct MXCommon__GetAutoConfigurationFileResponse * *Response*)

Parameters

`[in]` _ : No input parameter

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)
  - -100 : Error of the read of the auto configuration file

  • sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().
  • bArray : Array of Bytes of the file
  • ulEOF : End of file flag

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

`[in]` ***ByteArrayInput*** : Array of Bytes of the file

`[in]` ***ulEOF*** : End of file flag

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

[in] _ : No input parameter

[out] ***Response*** • sResponse.iReturnValue : Return value
  – 0 : success
  – -1: system error (see syserrno)
• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().
  – sArray : message returned by the auto configuration start

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.10 Common synchronisation timer functions

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

Functions

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

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

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

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

2.10.1 Function Documentation

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

Parameters

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

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

[in] ***ulNbrOfCycle*** : Number of timer cycle

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

[in] ***ulGenerateTriggerMode*** :

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

[in] ***ulOption01*** : Define the source of the trigger

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

[in] ***ulOption02*** : Define the edge of the hardware trigger who generates a trigger action

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

[in] ***ulOption03*** : Define the number of trigger events before the action occur

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

[in] ***ulOption04*** : Reserved

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

  - 0 : success
  - -1: system error (see syserrno)
  - -2: not available time base
  - -3: timer reload value can not be greater than 65535
  - -4: minimum time reload is 5 us
  - -5: Number of cycle can not be greater than 65535
  - -6: Generate trigger mode error
  - -100: Init timer error
  - -101: Start timer error

- sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror(). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

`[in]` ***ulOption01*** : Reserved

`[out]` ***Response*** • sResponse.iReturnValue : Return value
- 0 : success
- -1: system error (see syserrno)
- -100: Start/Stop timer error

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror(). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.11 Set/Backup/Restore general system configuration

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

Functions

- int MXCommon__GetConfigurationBackupFile (void *_, struct MXCommon__FileResponse *Response)

  Download a configuration backup file from the module.

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

  Upload a new configuration on the module.

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

  Set a new id/password.

2.11.1 Detailed Description

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

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

2.11.2 Function Documentation

2.11.2.1 int MXCommon__GetConfigurationBackupFile (void * _, struct MXCommon__FileResponse * *Response*)

Parameters

[in] _ : No input parameter

[out] ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno) (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

• bArray : Array of Bytes of the file

• ulEOF : End of file flag

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Below is an example in pseudo-C.

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

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

Parameters

[in] ***ByteArrayInput*** : Array of Bytes of the file

[in] ***ulEOF*** : End of file flag

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

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

The changes are immediately active.

Parameters

`[in]` _ : No input parameter

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: string PreviousUser is invalid
  - -2: string PreviousPassword is invalid
  - -3: string NewUser is invalid
  - -4: string NewPassword is invalid
  - -5: authentification failed
  - -100: system error while saving tokens (use syserrno for more information)
- sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().
- sArray : message returned by the auto configuration start

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

Warning

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

2.12 System state management

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

Functions

- int MXCommon__GetSubSystemState (xsd__unsignedLong SubsystemID, struct MXCommon__-unsignedLongResponse ∗Response)

  Returns the current state of the specified sub-system.

- int MXCommon__GetSubsystemIDFromName (struct xsd__base64Binary ∗SubsystemName, struct MXCommon__unsignedLongResponse ∗Response)

  Returns the ID of the sub-system of symbolic name "SubsystemName".

- int MXCommon__GetStateIDFromName (xsd__unsignedLong SubsystemID, struct xsd__-base64Binary ∗StateName, struct MXCommon__unsignedLongResponse ∗Response)

  Returns the ID of the state of symbolic name "StateName" of the sub-system of ID "SubsystemID".

- int MXCommon__GetSubsystemNameFromID (xsd__unsignedLong SubsystemID, struct MXCommon__ByteArrayResponse ∗Response)

  Returns the symbolic name of the sub-system of numerical ID "SubsystemName".

- int MXCommon__GetStateNameFromID (xsd__unsignedLong SubsystemID, xsd__unsignedLong StateID, struct MXCommon__ByteArrayResponse ∗Response)

  Returns the symbolic name of the state of numerical ID "StateID" of the sub-system of ID "SubsystemID".

2.12.1 Detailed Description

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

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

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

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

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

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

2.12.2 Function Documentation

2.12.2.1 int MXCommon__GetSubSystemState (xsd__unsignedLong *SubsystemID,* struct MXCommon__unsignedLongResponse ∗ *Response*)

Parameters

`[in]` ***SubsystemID*** sub-system numerical ID

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error while executing the request (see syserrno)
  - -2: invalid parameter SubsystemID

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

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

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

`[in]` ***SubsystemID*** sub-system numerical ID

`[in]` ***StateName*** state symbolic name.

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.12.2.4 int MXCommon__GetSubsystemNameFromID (xsd__unsignedLong *SubsystemID*, struct MXCommon__ByteArrayResponse * *Response*)

Parameters

`[in]` ***SubsystemID*** sub-system numerical ID.

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

`[in]` ***SubsystemID*** sub-system numerical ID.

`[in]` ***StateID*** sub-system numerical ID.

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.13 Customer option management

Enable to get informations about the options of the system.

Functions

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

  Enables to get information about the options available on the system.

2.13.1 Function Documentation

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

Parameters

[in] ***ulOption01,:*** not used, set it to 0

[in] ***ulOption02,:*** not used, set it to 0

[out] ***Response*** • sArray : Option information string
• sResponse Composed of iReturnValue and syserrno

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.14 Synchronisation management

Manage the synchronisation state of the system.

Functions

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

  Writes if the MSXE has to be always set to master The master mode (when enabled) make the system always detected as master.

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

  Reads the status of the synchronization for the corresponding MSXE The master mode (when enabled) make the system always detected as master.

2.14.1 Function Documentation

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

Parameters

[in] ***ulState*** State of the supermaster mode

- **0** automatic mode (default). The state of the system (master or slave) will be automatically detected by the system
- **1** Set to master mode at all time. The system will always be detected as master

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

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

[out] ***Response*** ***iReturnValue***

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

Return values

0 SOAP_OK

Others See SOAP error

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

Parameters

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

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

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

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

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

ulValue State of the supermaster mode

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

Return values

0 SOAP_OK

Others See SOAP error

2.15 input filter Filter management

Manages the analog input filters in the system.

Functions

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

  This function sets or resets a filter to a channel.

2.15.1 Function Documentation

2.15.1.1 int MXCommon__SetFilterChannels (struct xsd__base64Binary * *ChannelList,* struct MXCommon__Response * *Response*)

Parameters

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

`[out]` ***Response*** • sResponse.iReturnValue : Return value
- 0 : success
- -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

2.16 MSX-E17xx digital I/O functions

Data Structures

- struct MSXE17xx__DigitalIOGetNumberResponse

Functions

- int MSXE17xx__DigitalIOGetNumber (void *_, struct MSXE17xx__DigitalIOGetNumberResponse *Response)

  Returns the number of digital IO available on the module.

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

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

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

  Read a digital i/o channel value.

- int MSXE17xx__DigitalIOReadAllChannelsValue (void *_, struct MSXE17xx__unsignedLongResponse *Response)

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

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

  write a digital i/o channel value

- int MSXE17xx__DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE17xx__Response *Response)

  write all digital i/o channels value

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

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

- int MSXE17xx__DigitalIOTestShortCircuit (xsd__unsignedLong ulOption, struct MSXE17xx__-unsignedLongResponse *Response)

  Test short circuit status.

- int MSXE17xx__DigitalIORearmShortCircuit (xsd__unsignedLong ulOption, struct MSXE17xx_-_Response *Response)

  Rearm digital outputs after a short circuit happened.

2.16.1 Function Documentation

2.16.1.1 int MSXE17xx__DigitalIOGetNumber (void * _, struct MSXE17xx__DigitalIOGetNumberResponse * *Response*)

Parameters

`[in]` ***None***

`[out]` ***Response*** :

sResponse.iReturnValue :

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

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

Returns

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

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

Parameters

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

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

- 0 : input
- 1 : output

`[out]` ***Response*** :

iReturnValue :

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

- -2: Digital i/o port selection error
- -3: Port configuration selection error
- -100: Init dig i/o port kernel function error

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

Returns

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

2.16.1.3 int MSXE17xx__DigitalIOReadChannelValue (xsd__unsignedLong *ulChannel*, struct MSXE17xx__unsignedLongResponse * *Response*)

Parameters

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

`[out]` ***Response*** :

iReturnValue :

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

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

- 0
- 1

Returns

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

2.16.1.4 int MSXE17xx__DigitalIOReadAllChannelsValue (void * _, struct MSXE17xx__unsignedLongResponse * *Response*)

Parameters

`[in]` _ : no input parameter

`[out]` ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

[in] ***ulChannelValue*** : Channel value

- 0
- 1

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

2.16.1.6 int MSXE17xx__DigitalIOWriteAllChannelsValue (xsd__unsignedLong *ulChannelValue,* struct MSXE17xx__Response * *Response*)

Parameters

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

[in] ***ulOption*** : reserved

[out] ***Response*** :

iReturnValue :

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

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

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

- 0 : no short circuit
- 1 : short circuit

Returns

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

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

Parameters

[in] ***ulOption*** : reserved

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

2.17 MSX-E17xx IO watchdog functions

Data Structures

- struct MSXE17xx__IOWatchdogGetStatusAndValueResponse

Functions

- int MSXE17xx__IOWatchdogInitAndStart (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct MSXE17xx__-Response *Response)

  Init and start the digital output IO watchdog.

- int MSXE17xx__IOWatchdogStopAndRelease (xsd__unsignedLong ulOption, struct MSXE17xx_-_Response *Response)

  Stop and release the digital output watchdog.

- int MSXE17xx__IOWatchdogGetStatusAndValue (xsd__unsignedLong ulOption, struct MSXE17xx__IOWatchdogGetStatusAndValueResponse *Response)

  Get watchdog current status and value information.

2.17.1 Function Documentation

2.17.1.1 int MSXE17xx__IOWatchdogInitAndStart (xsd__unsignedLong *ulTimeBase*, xsd__unsignedLong *ulTimeValue*, xsd__unsignedLong *ulOption1*, xsd__unsignedLong *ulOption2*, struct MSXE17xx__Response * *Response*)

Parameters

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

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

`[in]` ***ulOption1*** : Reserved

`[in]` ***ulOption2*** : Reserved

`[out]` ***Response*** :

iReturnValue :

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

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

Returns

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

2.17.1.2 int MSXE17xx__IOWatchdogStopAndRelease (xsd__unsignedLong *ulOption*, struct MSXE17xx__Response * *Response*)

Parameters

[in] ***ulOption*** : reserved

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

2.17.1.3 int MSXE17xx__IOWatchdogGetStatusAndValue (xsd__unsignedLong *ulOption*, struct MSXE17xx__IOWatchdogGetStatusAndValueResponse * *Response*)

Parameters

[in] ***ulOption*** : Reserved

[out] ***Response*** :

iReturnValue :

- 0: remote function performed OK
- -1: an system error occured
- -2: channel selection error
- -100: Get diagnostic information kernel function error
  ulStatus : current status information
- BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXX0: is stopped,
- BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXX1: is running,
- BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXX0X: is not run down
- BIN XXXXXXXX XXXXXXXX XXXXXXXX XXXXXX1X: is run down
  ulValue : current value information (0 to 0xFFFF)
  ulInfo : reserved
  syserrno : system-error code (the value of the libc "errno" code)

Returns

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

2.18 MSX-E17xx multifunction common functions

Functions

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

Get the selected sub module functionality.

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

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

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

  Permit to activate the reference voltage to pin D-.

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

  Enable / disable trigger gate.

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

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

2.18.1 Function Documentation

2.18.1.1 int MSXE17xx__MFCommonGetSubModuleFunctionality (xsd__unsignedLong *ulMFModuleIndex,* struct MSXE17xx__unsignedLongResponse * *Response*)

Parameters

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

`[out]` ***Response*** :

ulValue :

- 0: Incremental counter
- -1: PWM

sResponse.iReturnValue :

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

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

Returns

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

2.18.1.2 int MSXE17xx__MFCommonSetInputsFilter (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulInputAFilterValue,* xsd__unsignedLong *ulInputBFilterValue,* xsd__unsignedLong *ulInputCFilterValue,* xsd__unsignedLong *ulInputDFilterValue,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__Response * *Response*)

Parameters

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

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

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

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

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

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

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

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

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

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

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

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

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

[out] ***Response*** :

iReturnValue :

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

- -6: Input D filter value selection error

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

Returns

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

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

Parameters

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

[in] ***ulActivationFlag*** :

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

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

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

Bit 0, Hardware trigger gate :

- 0 : Hardware trigger gate is disabled
- 1 : Hardware trigger gate is enabled

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

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

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

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

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

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

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

`[out]` ***Response*** :
iReturnValue :

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

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

Returns

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

2.19 MSX-E17xx Sinus Cosinus

Data Structures

- struct MSXE17xx__MFSinCosInitResponse
- struct MSXE17xx__MFSinCosInitExResponse
- struct MSXE17xx__MFSinCosReadResponse
- struct MSXE17xx__MFSinCosReadExResponse

Functions

- int MSXE17xx__MFSinCosInit (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulMode, xsd__unsignedLong ulSignalPeriod, xsd__unsignedLong ulResolution, xsd__unsignedLong ulValueFormat, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__MFSinCosInitResponse *Response)

  Use the function MSXE17xx__MFSinCosInitEx.

- int MSXE17xx__MFSinCosInitEx (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulMode, xsd__double dSignalPeriod, xsd__unsignedLong ulResolution, xsd__unsignedLong ulValueFormat, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__MFSinCosInitExResponse *Response)

  Initialize the selected Sinus / Cosinus module.

- int MSXE17xx__MFSinCosRead (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__MFSinCosReadResponse *Response)

  Use the function MSXE17xx__MFSinCosReadEx.

- int MSXE17xx__MFSinCosReadEx (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__MFSinCosReadExResponse *Response)

  Read measured value on the selected Sinus / Cosinus module.

- int MSXE17xx__MFSinCosClear (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Clear the selected Sinus / Cosinus module.

- int MSXE17xx__MFSinCosRelease (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Release the selected Sinus / Cosinus module.

- int MSXE17xx__MFSinCosInitHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulCount, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init the hardware trigger configuration.

- int MSXE17xx__MFSinCosReleaseHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)

  Release the hardware trigger.

- int MSXE17xx__MFSinCosInitIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init the index configuration.

- int MSXE17xx__MFSinCosReleaseIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)

  Release the index.

- int MSXE17xx__MFSinCosInitAndEnableLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init and enable a counter latch logic.

- int MSXE17xx__MFSinCosDisableAndReleaseLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)

  Disable and Release a counter latch logic.

- int MSXE17xx__MFSinCosInitAndEnableClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init and enable a counter clear logic.

- int MSXE17xx__MFSinCosDisableAndReleaseClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)

  Disable and Release a counter clear logic.

- int MSXE17xx__MFSinCosInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__double dValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init and enable a counter compare value.

- int MSXE17xx__MFSinCosDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)

  Disable and Release a counter compare value.

2.19.1 Function Documentation

2.19.1.1 int MSXE17xx__MFSinCosInit (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulMode,* xsd__unsignedLong *ulSignalPeriod,* xsd__unsignedLong *ulResolution,* xsd__unsignedLong *ulValueFormat,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__MFSinCosInitResponse * *Response*)

2.19.1.2 int MSXE17xx__MFSinCosInitEx (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulMode,* xsd__double *dSignalPeriod,* xsd__unsignedLong *ulResolution,* xsd__unsignedLong *ulValueFormat,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__MFSinCosInitExResponse * *Response*)

Parameters

[in] ***ulModuleIndex*** : The module to initialize (0 to 3).

[in] ***ulMode*** : Measure mode

- 0 : Fast measure. The measure is fast,
  always 250 kHz but the measure range is smaller.
- 1 : Full range. The measure is slow but the maximal range is used.
  See in the table called
  "Max. input frequency in corresponding with the input ulResolution"
  in order to set the measure frequency.

[in] ***dSignalPeriod*** : Signal period.

[in] ***ulResolution*** : Reslution to use for the measure (binary value)

Max. input frequency in corresponding with the input ulResolution:

Resolution	Max. Freq. Hz.	Compatible with
16	250000	fast mode and full range mode
25	26000	fast mode
32	162500	fast mode and full range mode
40	16300	fast mode and full range mode
50	26000	fast mode
64	81300	fast mode and full range mode
80	16300	fast mode and full range mode
100	26000	fast mode and full range mode
125	20800	fast mode
128	40600	fast mode and full range mode
160	16300	fast mode and full range mode
200	26000	fast mode and full range mode
250	20800	fast mode
256	20300	fast mode and full range mode
320	16300	fast mode and full range mode
400	13000	fast mode and full range mode
500	10400	fast mode and full range mode
512	10200	fast mode and full range mode
800	6500	fast mode and full range mode
1000	5200	fast mode and full range mode
1024	5100	fast mode and full range mode
1600	3300	fast mode and full range mode
2000	2600	fast mode and full range mode
2048	2540	fast mode and full range mode
4096	1270	fast mode and full range mode
8192	635	fast mode and full range mode

*

[in] ***ulValueFormat*** : Output format of the measure

- 0 : Raw data.
- 1 : Standardized in mm.

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

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

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

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

[out] ***Response*** : ***ulMaxInputFrequency :*** Return the maximal input frequency that can be used (in Hz).

sResponse.iReturnValue :

- 0 : No error.
- -1 : means an system error occured
- -2 : Multifunction sub module index selection error.
- -3 : Multifunction sub module is not a SinCos module.
- -4 : Wrong mode.
- -5 : Wrong signal period.
- -6 : Wrong resolution.
- -7 : The resolution is not supported by the selected mode.
- -8 : Wrong format.
- -9 : Auto gain calibration error.
- -100 : Kernel function error (see syserrno).

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

Returns

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

2.19.1.3 int MSXE17xx__MFSinCosRead (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__MFSinCosReadResponse * *Response*)

2.19.1.4 int MSXE17xx__MFSinCosReadEx (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__MFSinCosReadExResponse * *Response*)

Parameters

[in] ***ulModuleIndex*** : The module to use (0 to 3).

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

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

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

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

[out] ***Response*** : ***iReturnValue :*** Return an error number

- 0 : No error.
- -1 : means an system error occured
- -2 : Multifunction sub module index selection error.
- -3 : Multifunction sub module is not a SinCos module.
- -4 : Sinus / Cosinus module not initialised
- -100 : Read Sinus Cosinus kernel function error (see syserrno).

dValue : Measured value, as raw or converted in the selected format.

ulMeasureError : Measure error.

0 : No error.

1 : Amplitude error.

2 : Frequency error (in fast mode is this error not relevant).

3 : Amplitude and frequency error.

Returns

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

2.19.1.5 int MSXE17xx__MFSinCosClear (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* struct MSXE17xx__Response * *Response*)

Parameters

[in] ***ulModuleIndex*** : The module to clear (0 to 3).

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

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

[out] ***Response*** : ***iReturnValue :*** Return an error number

- 0 : No error.
- -1 : means an system error occured
- -2 : Multifunction sub module index selection error.
- -3 : Multifunction sub module is not a SinCos module.
- -4 : Sinus / Cosinus module not initialised
- -100 : Clear Sinus Cosinus kernel function error (see syserrno).

Returns

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

2.19.1.6 int MSXE17xx__MFSinCosRelease (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* struct MSXE17xx__Response * *Response*)

Parameters

[in] ***ulModuleIndex*** : The module to clear (0 to 3).

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

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

[out] ***Response*** : ***iReturnValue :*** Return an error number

- 0 : No error.
- -1 : means an system error occured
- -2 : Multifunction sub module index selection error.
- -3 : Multifunction sub module is not a SinCos module.
- -4 : Sinus / Cosinus module not initialised
- -100 : Release Sinus Cosinus kernel function error (see syserrno).

Returns

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

2.19.1.7 int MSXE17xx__MFSinCosInitHardwareTrigger (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulEdgeSelection,* xsd__unsignedLong *ulCount,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulEdgeSelection*** : Front selection

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

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

[in] ***ulOption01*** : Hardware trigger gate, if enabled hardware trigger is not activated until MSXE17xx__EnableDisableHardwareTriggerGate is called

- 0 : Hardware trigger gate is not used
- 1 : Hardware trigger gate is used

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

[out] ***Response*** :
iReturnValue :

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

- -3: Multifunction sub module is not a SinCos module.
- -4: Trigger edge selection error
- -5: Trigger count selection error
- -6: Sinus / Cosinus module not initialised
- -7: Hardware trigger already initialised
- -8: ulOption01 parameter is wrong must be 0 or 1
- -100: Init hardware trigger kernel function error

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

Returns

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

2.19.1.8 int MSXE17xx__MFSinCosReleaseHardwareTrigger (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulOption01,* struct MSXE17xx__Response * *Response*)

Parameters

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Hardware trigger not initialised
- -6: Hardware trigger used and can not released
- -100: Release hardware trigger kernel function error

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

Returns

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

2.19.1.9 int MSXE17xx__MFSinCosInitIndex (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulEdgeSelection,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulEdgeSelection*** : Front selection

- 01 : rising front

- 10 : falling front
- 11 : Both front

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Index edge selection error
- -5: Sinus / Cosinus module not initialised
- -6: Index already initialised
- -100: Init index kernel function error

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

Returns

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

2.19.1.10 int MSXE17xx__MFSinCosReleaseIndex (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulOption01,* struct MSXE17xx__Response * *Response*)

Parameters

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Index not initialised
- -6: Index used and can not released
- -100: Release Index kernel function error

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

Returns

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

2.19.1.11 int MSXE17xx__MFSinCosInitAndEnableLatch (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulLatchSource,* xsd__unsignedLong *ulCondition,* xsd__unsignedLong *ulAutoMode,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* struct MSXE17xx__Response * *Response*)

```
For each latch the data server send a 5 DWORD frame with following informations:
  DWORD 0 : Time stamp micro s
  DWORD 1 : Time stamp s
  DWORD 2 :
     D1-D0   : Sub module index (0 to 3)
     D31-D16 : Sub module functionality (2)
  DWORD 3 : Event mask
     D30-D0 :
        2: Hardware trigger latch occur
        3: Synchro input latch occur
        4: Index input latch occur
     D31 :
        0: No error occur
        1: Amplitude or Frequency error occur.
  DWORD 4 :
     D31-D0: Counter value (DWORD) if the selected output format of the measure i
      s raw data
     D31-D0: Current position in mm (FLOAT) if the selected output format of the
      measure is standardized in mm.
```

Parameters

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

[in] ***ulLatchSource*** : Latch source.

- 0: Index input
- 1: Hardware trigger
- 2: Synchro input

[in] ***ulCondition*** : Previously condition for accept the latch source

- 0: No previously condition required
- 1: Index input condition required (Only if index input not selected selected for the latch source)
- 2: Hardware trigger condition required (Only if hardware trigger not selected selected for the latch source)
- 3: Synchro input condition required (Only if synchro input not selected selected for the latch source)

[in] ***ulAutoMode*** : Action mode

- 0: Do not use auto mode (action is done only once)
- 1: Use auto mode (action is done continously)

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Latch logic already initialised

- -6: Latch source selection error
- -7: Previously condition selection error
- -8: Auto mode selection error
- -9: Hardware trigger not initialised. Refer to MSXE17xx__MFSinCosInitHardwareTrigger
- -10: Index input not initialised. Refer to MSXE17xx__MFSinCosInitIndex
- -11: Can not be used for the "fast measure" mode
- -100: Init and enable counter latch kernel function error

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

Returns

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

2.19.1.12 int MSXE17xx__MFSinCosDisableAndReleaseLatch (xsd__unsignedLong *ulMFModuleIndex*, xsd__unsignedLong *ulLatchSource*, xsd__unsignedLong *ulOption01*, struct MSXE17xx__Response * *Response*)

Parameters

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

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

- 0: Index input
- 1: Hardware trigger
- 2: Synchro input

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

`[out]` ***Response*** :

iReturnValue :

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

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

Returns

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

2.19.1.13 int MSXE17xx__MFSinCosInitAndEnableClear (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulClearSource,* xsd__unsignedLong *ulCondition,* xsd__unsignedLong *ulAutoMode,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulClearSource*** : Clear source.

- 0: Index input
- 1: Hardware trigger
- 2: Synchro input

[in] ***ulCondition*** : Previously condition for accept the clear source

- 0: No previously condition required
- 1: Index input condition required (Only if index input not selected selected for the clear source)
- 2: Hardware trigger condition required (Only if hardware trigger not selected selected for the clear source)
- 3: Synchro input condition required (Only if synchro input not selected selected for the clear source)

[in] ***ulAutoMode*** : Action mode

- 0: Do not use auto mode (action is done only once)
- 1: Use auto mode (action is done continously)

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Clear logic already initialised
- -6: Clear source selection error
- -7: Previously condition selection error
- -8: Auto mode selection error
- -9: Hardware trigger not initialised. Refer to MSXE17xx__MFSinCosInitHardwareTrigger
- -10: Index input not initialised. Refer to MSXE17xx__MFSinCosInitIndex
- -11: Can not be used for the "fast measure" mode
- -100: Init and enable counter clear kernel function error

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

Returns

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

2.19.1.14 int MSXE17xx__MFSinCosDisableAndReleaseClear (xsd__unsignedLong *ulMFModuleIndex*, xsd__unsignedLong *ulClearSource*, xsd__unsignedLong *ulOption01*, struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulClearSource*** : Clear source to disable and release.

- 0: Index input
- 1: Hardware trigger
- 2: Synchro input

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Clear logic not initialised
- -6: Clear source selection error
- -100: Disable and release counter clear register kernel function error

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

Returns

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

2.19.1.15 int MSXE17xx__MFSinCosInitAndEnableCompareLogic (xsd__unsignedLong *ulMFModuleIndex*, xsd__double *dValue*, xsd__unsignedLong *ulMode*, xsd__unsignedLong *ulSynchroTrigger*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, struct MSXE17xx__Response * *Response*)

```
For each compare the data server send a 5 DWORD frame with following informations
      :
  DWORD 0 : Time stamp micro s
  DWORD 1 : Time stamp s
  DWORD 2 :
     D1-D0   : Sub module index (0 to 3)
     D31-D16 : Sub module functionality (2)
  DWORD 3 : Event mask
     D30-D0 :
        0: Compare occur
     D31 :
        0: No error occur
        1: Amplitude or Frequency error occur.
  DWORD 4 :
     D31-D0: Counter value (DWORD) if the selected output format of the measure i
      s Raw data
     D31-D0: Current position in mm (FLOAT) if the selected output format of the
      measure is standardized in mm.
```

Parameters

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

[in] ***dValue*** : compare value :

- 0 to 0xFFFFFFFF if raw data seleced
- Position in mm if the measure is standardized in mm

[in] ***ulMode*** : compare mode

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

[in] ***ulSynchroTrigger*** • 0 : no synchro trigger

- 1 : generates a synchro trigger when condition is true

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Compare value error
- -4: Compare mode error
- -5: Synchro trigger error
- -6: Multifunction sub module is not a SinCos module.
- -7: Sinus / Cosinus module not initialised
- -8: Compare logic already initialised
- -9: Can not be used for the "fast measure" mode
- -100: Init and enable counter compare kernel function error

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

Returns

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

2.19.1.16 int MSXE17xx__MFSinCosDisableAndReleaseCompareLogic (xsd__unsignedLong *ulMFModuleIndex*, struct MSXE17xx__Response * *Response*)

Parameters

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Compare logic not initialised

- -100: Disable counter compare value kernel function error

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

Returns

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

Chapter 3

Data Structure Documentation

3.1 ByteArray Struct Reference

Dynamic Array of byte - encapsulates C-type strings.

Data Fields

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

3.1.1 Field Documentation

3.1.1.1 xsd__unsignedByte* ByteArray::__ptr

3.1.1.2 int ByteArray::__size

3.1.1.3 int ByteArray::__offset

3.2 DefaultResponse Struct Reference

Data Fields

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

3.2.1 Field Documentation

3.2.1.1 xsd__int DefaultResponse::iReturnValue

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

3.2.1.2 xsd__int DefaultResponse::syserrno

3.3 MSXE17xx__DigitalIOGetNumberResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__unsignedLong ulNumberOfDigitalIO

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

3.3.1 Field Documentation

3.3.1.1 struct DefaultResponse MSXE17xx__DigitalIOGetNumberResponse::sResponse

3.3.1.2 xsd__unsignedLong MSXE17xx__DigitalIOGetNumberResponse::ulNumberOfDigitalIO

3.4 MSXE17xx__IOWatchdogGetStatusAndValueResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__unsignedLong ulStatus

  Watchdog current status information.

- xsd__unsignedLong ulValue

  Watchdog current value information.

- xsd__unsignedLong ulInfo

  reserved

3.4.1 Field Documentation

3.4.1.1 struct DefaultResponse MSXE17xx__-IOWatchdogGetStatusAndValueResponse::sResponse

3.4.1.2 xsd__unsignedLong MSXE17xx__IOWatchdogGetStatusAndValueResponse::ulStatus

3.4.1.3 xsd__unsignedLong MSXE17xx__IOWatchdogGetStatusAndValueResponse::ulValue

3.4.1.4 xsd__unsignedLong MSXE17xx__IOWatchdogGetStatusAndValueResponse::ulInfo

3.5 MSXE17xx__MFSinCosInitExResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__unsignedLong ulMaxInputFrequency

  The maximal input frequency that can be used (in Hz).

- xsd__unsignedLong ulInfo01

  Reserved.

- xsd__unsignedLong ulInfo02

  Reserved.

3.5.1 Field Documentation

3.5.1.1 struct DefaultResponse MSXE17xx__MFSinCosInitExResponse::sResponse

3.5.1.2 xsd__unsignedLong MSXE17xx__MFSinCosInitExResponse::ulMaxInputFrequency

3.5.1.3 xsd__unsignedLong MSXE17xx__MFSinCosInitExResponse::ulInfo01

3.5.1.4 xsd__unsignedLong MSXE17xx__MFSinCosInitExResponse::ulInfo02

3.6 MSXE17xx__MFSinCosInitResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__unsignedLong ulMaxInputFrequency

  The maximal input frequency that can be used (in Hz).

3.6.1 Field Documentation

3.6.1.1 struct DefaultResponse MSXE17xx__MFSinCosInitResponse::sResponse

3.6.1.2 xsd__unsignedLong MSXE17xx__MFSinCosInitResponse::ulMaxInputFrequency

3.7 MSXE17xx__MFSinCosReadExResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__double dValue

  Measured value, as raw or converted in the selected format.

- xsd__unsignedLong ulMeasureError

  Measure erros :
  0 : No error.

- xsd__unsignedLong ulInfo01

  Reserved.

- xsd__unsignedLong ulInfo02

  Reserved.

3.7.1 Field Documentation

3.7.1.1 struct DefaultResponse MSXE17xx__MFSinCosReadExResponse::sResponse

3.7.1.2 xsd__double MSXE17xx__MFSinCosReadExResponse::dValue

3.7.1.3 xsd__unsignedLong MSXE17xx__MFSinCosReadExResponse::ulMeasureError

1 : Amplitude error.

2 : Frequency error (in fast mode is this error not relevant).

3 : Amplitude and frequency error.

3.7.1.4 xsd__unsignedLong MSXE17xx__MFSinCosReadExResponse::ulInfo01

3.7.1.5 xsd__unsignedLong MSXE17xx__MFSinCosReadExResponse::ulInfo02

3.8 MSXE17xx__MFSinCosReadResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

Default return values.

- xsd__unsignedLong ulValue

  Measured value, as raw or converted in the selected format.

- xsd__unsignedLong ulMeasureError

  Measure erros :
  0 : No error.

3.8.1 Field Documentation

3.8.1.1 struct DefaultResponse MSXE17xx__MFSinCosReadResponse::sResponse

3.8.1.2 xsd__unsignedLong MSXE17xx__MFSinCosReadResponse::ulValue

3.8.1.3 xsd__unsignedLong MSXE17xx__MFSinCosReadResponse::ulMeasureError

1 : Amplitude error.

2 : Frequency error (in fast mode is this error not relevant).

3 : Amplitude and frequency error.

3.9 MSXE17xx__Response Struct Reference

Data Fields

- xsd__int iReturnValue

  return value of the call :

- xsd__int syserrno

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

3.9.1 Field Documentation

3.9.1.1 xsd__int MSXE17xx__Response::iReturnValue

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

3.9.1.2 xsd__int MSXE17xx__Response::syserrno

3.10 MSXE17xx__unsignedLongResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__unsignedLong ulValue

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

3.10.1 Field Documentation

3.10.1.1 struct DefaultResponse MSXE17xx__unsignedLongResponse::sResponse

3.10.1.2 xsd__unsignedLong MSXE17xx__unsignedLongResponse::ulValue

3.11 MSXE17xx__unsignedLongTimeStampResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__unsignedLong ulValue

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

- xsd__unsignedLong ulTimeStampLow

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

- xsd__unsignedLong ulTimeStampHigh

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

3.11.1 Field Documentation

3.11.1.1 struct DefaultResponse MSXE17xx__unsignedLongTimeStampResponse::sResponse

3.11.1.2 xsd__unsignedLong MSXE17xx__unsignedLongTimeStampResponse::ulValue

3.11.1.3 xsd__unsignedLong MSXE17xx__unsignedLongTimeStampResponse::ulTimeStampLow

3.11.1.4 xsd__unsignedLong MSXE17xx__unsignedLongTimeStampResponse::ulTimeStampHigh

3.12 MXCommon__ByteArrayResponse Struct Reference

Response containing a C-type string.

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- struct ByteArray sArray

  Dynamic Array of byte - encapsulates C-type strings.

3.12.1 Field Documentation

3.12.1.1 struct DefaultResponse MXCommon__ByteArrayResponse::sResponse

3.12.1.2 struct ByteArray MXCommon__ByteArrayResponse::sArray

3.13 MXCommon__FileResponse Struct Reference

Response containing a chunk of a file.

Data Fields

- struct DefaultResponse sResponse

  return values.

- struct ByteArray sArray

  Dynamic Array of byte.

- xsd__unsignedLong ulEOF

  flag indicating end of file.

3.13.1 Field Documentation

3.13.1.1 struct DefaultResponse MXCommon__FileResponse::sResponse

3.13.1.2 struct ByteArray MXCommon__FileResponse::sArray

3.13.1.3 xsd__unsignedLong MXCommon__FileResponse::ulEOF

3.14 MXCommon__GetAutoConfigurationFileResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- struct ByteArray bArray

  Array of byte of the file.

- xsd__unsignedLong ulEOF

  End of file flag.

3.14.1 Field Documentation

3.14.1.1 struct DefaultResponse MXCommon__GetAutoConfigurationFileResponse::sResponse

3.14.1.2 struct ByteArray MXCommon__GetAutoConfigurationFileResponse::bArray

3.14.1.3 xsd__unsignedLong MXCommon__GetAutoConfigurationFileResponse::ulEOF

3.15 MXCommon__GetEthernetLinksStatesResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- struct sGetEthernetLinksStatesPort sPort0
- struct sGetEthernetLinksStatesPort sPort1

3.15.1 Field Documentation

3.15.1.1 struct DefaultResponse MXCommon__GetEthernetLinksStatesResponse::sResponse

3.15.1.2 struct sGetEthernetLinksStatesPort MXCommon__-GetEthernetLinksStatesResponse::sPort0

3.15.1.3 struct sGetEthernetLinksStatesPort MXCommon__-GetEthernetLinksStatesResponse::sPort1

3.16 MXCommon__GetHardwareTriggerFilterTimeResponse Struct Reference

Data Fields

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

3.16.1 Field Documentation

3.16.1.1 struct DefaultResponse MXCommon__-GetHardwareTriggerFilterTimeResponse::sResponse

3.16.1.2 xsd__unsignedLong MXCommon__-GetHardwareTriggerFilterTimeResponse::ulFilterTime

3.16.1.3 xsd__unsignedLong MXCommon__GetHardwareTriggerFilterTimeResponse::ulInfo01

3.16.1.4 xsd__unsignedLong MXCommon__GetHardwareTriggerFilterTimeResponse::ulInfo02

3.17 MXCommon__GetHardwareTriggerStateResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse
  Default return values.
- xsd__unsignedLong ulState

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

- xsd__unsignedLong ulInfo01

  Reserved.

- xsd__unsignedLong ulInfo02

  Reserved.

3.17.1 Field Documentation

3.17.1.1 struct DefaultResponse MXCommon__GetHardwareTriggerStateResponse::sResponse

3.17.1.2 xsd__unsignedLong MXCommon__GetHardwareTriggerStateResponse::ulState

3.17.1.3 xsd__unsignedLong MXCommon__GetHardwareTriggerStateResponse::ulInfo01

3.17.1.4 xsd__unsignedLong MXCommon__GetHardwareTriggerStateResponse::ulInfo02

3.18 MXCommon__GetModuleTemperatureValueAndStatusResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return value.

- xsd__double dTemperatureValue

  Temperature value.

- xsd__unsignedLong ulTemperatureStatus

  Temperature status.

- xsd__unsignedLong ulInfo

  Reserved.

3.18.1 Field Documentation

3.18.1.1 struct DefaultResponse MXCommon__-GetModuleTemperatureValueAndStatusResponse::sResponse

3.18.1.2 xsd__double MXCommon__-GetModuleTemperatureValueAndStatusResponse::dTemperatureValue

3.18.1.3 xsd__unsignedLong MXCommon__-GetModuleTemperatureValueAndStatusResponse::ulTemperatureStatus

3.18.1.4 xsd__unsignedLong MXCommon__-GetModuleTemperatureValueAndStatusResponse::ulInfo

3.19 MXCommon__GetTimeResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__unsignedLong ulLowTime

  Number of microseconds since the begin of the second.

- xsd__unsignedLong ulHighTime

  Number of seconds since the Epoch (1st January,1970).

3.19.1 Field Documentation

3.19.1.1 struct DefaultResponse MXCommon__GetTimeResponse::sResponse

3.19.1.2 xsd__unsignedLong MXCommon__GetTimeResponse::ulLowTime

3.19.1.3 xsd__unsignedLong MXCommon__GetTimeResponse::ulHighTime

3.20 MXCommon__GetUpTimeResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return value.

- xsd__unsignedLong ulUpTime

  Reserved.

3.20.1 Field Documentation

3.20.1.1 struct DefaultResponse MXCommon__GetUpTimeResponse::sResponse

3.20.1.2 xsd__unsignedLong MXCommon__GetUpTimeResponse::ulUpTime

3.21 MXCommon__Response Struct Reference

contains return values

Data Fields

- xsd__int iReturnValue

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

- xsd__int syserrno

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

3.21.1 Field Documentation

3.21.1.1 xsd__int MXCommon__Response::iReturnValue

3.21.1.2 xsd__int MXCommon__Response::syserrno

3.22 MXCommon__TestCustomerIDResponse Struct Reference

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- struct ByteArray bValueArray

  non encrypted value

- struct ByteArray bCryptedValueArray

  encrypted value

3.22.1 Field Documentation

3.22.1.1 struct DefaultResponse MXCommon__TestCustomerIDResponse::sResponse

3.22.1.2 struct ByteArray MXCommon__TestCustomerIDResponse::bValueArray

3.22.1.3 struct ByteArray MXCommon__TestCustomerIDResponse::bCryptedValueArray

3.23 MXCommon__unsignedLongResponse Struct Reference

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

Data Fields

- struct DefaultResponse sResponse

  Default return values.

- xsd__unsignedLong ulValue

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

3.23.1 Field Documentation

3.23.1.1 struct DefaultResponse MXCommon__unsignedLongResponse::sResponse

3.23.1.2 xsd__unsignedLong MXCommon__unsignedLongResponse::ulValue

3.24 sGetEthernetLinksStatesPort Struct Reference

Data Fields

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

3.24.1 Field Documentation

3.24.1.1 xsd__unsignedLong sGetEthernetLinksStatesPort::ulState

3.24.1.2 xsd__unsignedLong sGetEthernetLinksStatesPort::ulSpeed

3.24.1.3 xsd__unsignedLong sGetEthernetLinksStatesPort::ulDuplex

3.24.1.4 xsd__unsignedLong sGetEthernetLinksStatesPort::ulInfo1

3.24.1.5 xsd__unsignedLong sGetEthernetLinksStatesPort::ulInfo2

3.25 UnsignedLongArray Struct Reference

Dynamic Array of unsigned long.

Data Fields

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

3.25.1 Field Documentation

3.25.1.1 xsd__unsignedLong* UnsignedLongArray::__ptr

3.25.1.2 int UnsignedLongArray::__size

3.25.1.3 int UnsignedLongArray::__offset

3.26 UnsignedShortArray Struct Reference

Dynamic Array of unsigned short.

Data Fields

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

- int __offset
  not used

3.26.1 Field Documentation

3.26.1.1 xsd__unsignedShort* UnsignedShortArray::__ptr

3.26.1.2 int UnsignedShortArray::__size

3.26.1.3 int UnsignedShortArray::__offset

3.27 xsd__base64Binary Struct Reference

Dynamic Array of byte for input use.

Data Fields

- unsigned char * __ptr
  pointer of byte
- int __size
  size of the byte array

3.27.1 Field Documentation

3.27.1.1 unsigned char* xsd__base64Binary::__ptr

3.27.1.2 int xsd__base64Binary::__size

Chapter 4

File Documentation

4.1 MSXE171x_public_doc.h File Reference

Data Structures

- struct xsd__base64Binary

  Dynamic Array of byte for input use.

- struct UnsignedShortArray

  Dynamic Array of unsigned short.

- struct UnsignedLongArray

  Dynamic Array of unsigned long.

- struct ByteArray

  Dynamic Array of byte - encapsulates C-type strings.

- struct DefaultResponse
- struct MXCommon__Response

  contains return values

- struct MXCommon__ByteArrayResponse

  Response containing a C-type string.

- struct MXCommon__FileResponse

  Response containing a chunk of a file.

- struct MXCommon__unsignedLongResponse

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

- struct sGetEthernetLinksStatesPort
- struct MXCommon__GetEthernetLinksStatesResponse
- struct MXCommon__GetModuleTemperatureValueAndStatusResponse
- struct MXCommon__GetHardwareTriggerFilterTimeResponse
- struct MXCommon__GetHardwareTriggerStateResponse

- struct MXCommon__TestCustomerIDResponse
- struct MXCommon__GetTimeResponse
- struct MXCommon__GetUpTimeResponse
- struct MXCommon__GetAutoConfigurationFileResponse
- struct MSXE17xx__Response
- struct MSXE17xx__unsignedLongResponse
- struct MSXE17xx__unsignedLongTimeStampResponse
- struct MSXE17xx__DigitalIOGetNumberResponse
- struct MSXE17xx__IOWatchdogGetStatusAndValueResponse
- struct MSXE17xx__MFSinCosInitResponse
- struct MSXE17xx__MFSinCosInitExResponse
- struct MSXE17xx__MFSinCosReadResponse
- struct MSXE17xx__MFSinCosReadExResponse

Defines

- #define MSXE170X_COUNTER_QUADRUPLE_MODE 0x4

  In the quadruple mode, the edge analysis circuit generates a counting pulse from each edge of two signals which are phase-shifted in relation to each other.

- #define MSXE170X_COUNTER_DOUBLE_MODE 0x2

  Same function as quadruple mode, except only 2 of the 4 edges are analysed.

- #define MSXE170X_COUNTER_SIMPLE_MODE 0x1

  Same function as quadruple mode, except one of the 4 edges is analysed in each period.

- #define MSXE170X_COUNTER_DIRECT_MODE 0x0

  In the direct mode both edge analysis circuits become inactive.

- #define MSXE170X_COUNTER_HYSTERESIS_ON 0x1

  On both edge analysis circuit a hysteresis switch is available.

- #define MSXE170X_COUNTER_HYSTERESIS_OFF 0x0

  The first pulse will not be suppressed after a change of rotation.

- #define MSXE170X_COUNTER_INCREMENT 0x0

  The counter increments after each counting pulse.

- #define MSXE170X_COUNTER_DECREMENT 0x1

  The counter decrements after each counting pulse.

- #define MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_COUNTER 0x0

  After an index signal (Low level), the counter value (32-bit) is latched into the first latch register and then deleted (32-bit).

- #define MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_COUNTER 0x1

  After an index signal (High level), the counter value (32-bit) is latched into the first latch register and then deleted (32-bit).

- #define MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER 0x2

After an index signal (Low level), the counter value (32-bit) is latched into the first latch register.

- #define MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER 0x3
  After an index signal (High level), the counter value (32-bit) is latched into the first latch register.

Typedefs

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

Functions

- int MXCommon__GetModuleType (void *_, struct MXCommon__ByteArrayResponse *Response)
  This function return the type of the MSX-E Module.
- int MXCommon__GetHostname (void *_, struct MXCommon__ByteArrayResponse *Response)
  This function return the hostname of the MSX-E Module.
- int MXCommon__SetHostname (struct xsd__base64Binary *bHostname, struct MXCommon__-Response *Response)

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

- int MXCommon__GetClientConnections (void *_, struct MXCommon__ByteArrayResponse *Response)

  This function return the client connection list.

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

  Call the libc strerror() on the remote device (actually this is a call to strerror_r()).

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

  Ask the MSX-E module to reboot.

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

  Reset the I/O functionalities of the MSX-E system.

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

  Restart the data-server service.

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

  Get MSX-E Ethernet links states.

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

  Read the temperature on the module.

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

  Set the temperature warning level on the module.

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

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

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

  Get the filter time for the hardware trigger input.

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

  Get the hardware trigger state after the filter.

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

  Set the Customer key.

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

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

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

  Set the time on the module.

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

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

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

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

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

  Get the time on the module.

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

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

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

  Get the auto configuration file of the module.

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

  Set the auto configuration file of the module.

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

  start/Restart the auto configuration

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

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

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

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

- int MXCommon__GetConfigurationBackupFile (void *_, struct MXCommon__FileResponse *Response)

  Download a configuration backup file from the module.

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

  Upload a new configuration on the module.

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

  Set a new id/password.

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

  Returns the current state of the specified sub-system.

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

  Returns the ID of the sub-system of symbolic name "SubsystemName".

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

  Returns the ID of the state of symbolic name "StateName" of the sub-system of ID "SubsystemID".

- int MXCommon__GetSubsystemNameFromID (xsd__unsignedLong SubsystemID, struct MXCommon__ByteArrayResponse *Response)

  Returns the symbolic name of the sub-system of numerical ID "SubsystemName".

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

  Returns the symbolic name of the state of numerical ID "StateID" of the sub-system of ID "SubsystemID".

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

  Enables to get information about the options available on the system.

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

  Writes if the MSXE has to be always set to master The master mode (when enabled) make the system always detected as master.

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

  Reads the status of the synchronization for the corresponding MSXE The master mode (when enabled) make the system always detected as master.

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

  This function sets or resets a filter to a channel.

- int MSXE17xx__DigitalIOGetNumber (void *_, struct MSXE17xx__DigitalIOGetNumberResponse *Response)

  Returns the number of digital IO available on the module.

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

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

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

  Read a digital i/o channel value.

- int MSXE17xx__DigitalIOReadAllChannelsValue (void *_, struct MSXE17xx__unsignedLongResponse *Response)

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

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

  write a digital i/o channel value

- int MSXE17xx__DigitalIOWriteAllChannelsValue (xsd__unsignedLong ulChannelValue, struct MSXE17xx__Response *Response)

  write all digital i/o channels value

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

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

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

  Test short circuit status.

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

  Rearm digital outputs after a short circuit happened.

- int MSXE17xx__IOWatchdogInitAndStart (xsd__unsignedLong ulTimeBase, xsd__unsignedLong ulTimeValue, xsd__unsignedLong ulOption1, xsd__unsignedLong ulOption2, struct MSXE17xx__Response *Response)

  Init and start the digital output IO watchdog.

- int MSXE17xx__IOWatchdogStopAndRelease (xsd__unsignedLong ulOption, struct MSXE17xx__Response *Response)

  Stop and release the digital output watchdog.

- int MSXE17xx__IOWatchdogGetStatusAndValue (xsd__unsignedLong ulOption, struct MSXE17xx__IOWatchdogGetStatusAndValueResponse *Response)

  Get watchdog current status and value information.

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

  Get the selected sub module functionality.

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

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

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

  Permit to activate the reference voltage to pin D-.

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

  Enable / disable trigger gate.

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

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

- int MSXE17xx__MFSinCosInit (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulMode, xsd__unsignedLong ulSignalPeriod, xsd__unsignedLong ulResolution, xsd__unsignedLong ulValueFormat, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__MFSinCosInitResponse *Response)

  Use the function MSXE17xx__MFSinCosInitEx.

- int MSXE17xx__MFSinCosInitEx (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulMode, xsd__double dSignalPeriod, xsd__unsignedLong ulResolution, xsd__unsignedLong ulValueFormat, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__MFSinCosInitExResponse *Response)

  Initialize the selected Sinus / Cosinus module.

- int MSXE17xx__MFSinCosRead (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__MFSinCosReadResponse *Response)

  Use the function MSXE17xx__MFSinCosReadEx.

- int MSXE17xx__MFSinCosReadEx (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, xsd__unsignedLong ulOption03, xsd__unsignedLong ulOption04, struct MSXE17xx__MFSinCosReadExResponse *Response)

  Read measured value on the selected Sinus / Cosinus module.

- int MSXE17xx__MFSinCosClear (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Clear the selected Sinus / Cosinus module.

- int MSXE17xx__MFSinCosRelease (xsd__unsignedLong ulModuleIndex, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Release the selected Sinus / Cosinus module.

- int MSXE17xx__MFSinCosInitHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulCount, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init the hardware trigger configuration.

- int MSXE17xx__MFSinCosReleaseHardwareTrigger (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)

  Release the hardware trigger.

- int MSXE17xx__MFSinCosInitIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulEdgeSelection, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init the index configuration.

- int MSXE17xx__MFSinCosReleaseIndex (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)

  Release the index.

- int MSXE17xx__MFSinCosInitAndEnableLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init and enable a counter latch logic.

- int MSXE17xx__MFSinCosDisableAndReleaseLatch (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulLatchSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)

  Disable and Release a counter latch logic.

- int MSXE17xx__MFSinCosInitAndEnableClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulCondition, xsd__unsignedLong ulAutoMode, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init and enable a counter clear logic.

- int MSXE17xx__MFSinCosDisableAndReleaseClear (xsd__unsignedLong ulMFModuleIndex, xsd__unsignedLong ulClearSource, xsd__unsignedLong ulOption01, struct MSXE17xx__Response *Response)

  Disable and Release a counter clear logic.

- int MSXE17xx__MFSinCosInitAndEnableCompareLogic (xsd__unsignedLong ulMFModuleIndex, xsd__double dValue, xsd__unsignedLong ulMode, xsd__unsignedLong ulSynchroTrigger, xsd__unsignedLong ulOption01, xsd__unsignedLong ulOption02, struct MSXE17xx__Response *Response)

  Init and enable a counter compare value.

- int MSXE17xx__MFSinCosDisableAndReleaseCompareLogic (xsd__unsignedLong ulMFModuleIndex, struct MSXE17xx__Response *Response)

  Disable and Release a counter compare value.

4.1.1 Define Documentation

4.1.1.1 #define MSXE170X_COUNTER_QUADRUPLE_MODE 0x4

4.1.1.2 #define MSXE170X_COUNTER_DOUBLE_MODE 0x2

4.1.1.3 #define MSXE170X_COUNTER_SIMPLE_MODE 0x1

4.1.1.4 #define MSXE170X_COUNTER_DIRECT_MODE 0x0

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

4.1.1.5 #define MSXE170X_COUNTER_HYSTERESIS_ON 0x1

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

4.1.1.6 #define MSXE170X_COUNTER_HYSTERESIS_OFF 0x0

4.1.1.7 #define MSXE170X_COUNTER_INCREMENT 0x0

4.1.1.8 #define MSXE170X_COUNTER_DECREMENT 0x1

4.1.1.9 #define MSXE170X_COUNTER_LOW_EDGE_LATCH_AND_CLEAR_COUNTER 0x0

4.1.1.10 #define MSXE170X_COUNTER_HIGH_EDGE_LATCH_AND_CLEAR_-COUNTER 0x1

4.1.1.11 #define MSXE170X_COUNTER_LOW_EDGE_LATCH_COUNTER 0x2

4.1.1.12 #define MSXE170X_COUNTER_HIGH_EDGE_LATCH_COUNTER 0x3

4.1.2 Typedef Documentation

4.1.2.1 typedef char* xsd__string

4.1.2.2 typedef char xsd__char

4.1.2.3 typedef float xsd__float

4.1.2.4 typedef double xsd__double

4.1.2.5 typedef int xsd__int

4.1.2.6 typedef long xsd__long

4.1.2.7 typedef unsigned char xsd__unsignedByte

4.1.2.8 typedef unsigned int xsd__unsignedInt

4.1.2.9 typedef unsigned short int xsd__unsignedShort

4.1.2.10 typedef unsigned long xsd__unsignedLong

4.1.3 Function Documentation

4.1.3.1 int MXCommon__GetModuleType (void * _, struct MXCommon__ByteArrayResponse * *Response*)

Parameters

`[in]` _ : no input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.2 int MXCommon__GetHostname (void * _, struct MXCommon__ByteArrayResponse * *Response*)

Parameters

[in] _ : no input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.3 int MXCommon__SetHostname (struct xsd__base64Binary * *bHostname*, struct MXCommon__Response * *Response*)

Parameters

[in] ***bHostname*** : Hostname

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.4 int MXCommon__GetClientConnections (void * _, struct MXCommon__ByteArrayResponse * *Response*)

Parameters

[in] _ : no input parameter

[out] ***Response*** • sArray : string containing the list of connected clients.
- sResponse Composed of iReturnValue and syserrno

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

Sample: 172.16.3.43:8989-5555 172.16.3.200:8989

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

Parameters

[in] ***errnum*** : Error number

[out] ***Response*** • sArray : See the description below.

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

```
STRERROR(3)                                                    Linux Programmer's Manual
STRERROR(3)

NAME
strerror, strerror_r - return string describing error code

SYNOPSIS
#include <string.h>

char *strerror(int errnum);

#define _XOPEN_SOURCE 600
#include <string.h>

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

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

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

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

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

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

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

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

```
is a GNU extension used by glibc (since 2.0), and must be regarded as obsolete
in view of SUSv3.
The GNU version may, but need not, use the user-supplied buffer.
If it does, the result may be truncated in case the supplied buffer is too small.
The result is always NUL-terminated.

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

[in] _ : no input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

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

Parameters

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

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

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

Return values

0 SOAP_OK

Others See SOAP error

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

Parameters

[in] ***ulAction*** : action

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

[in] ***ulOption*** : Reserved

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

Note

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

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

Parameters

[in] _ : no input parameter

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

sResponse.iReturnValue

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

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

sPort0: Fisrt port informations

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

- **ulInfo1** Reserverd
- **ulInfo2** Reserverd

sPort1: Second port informations

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

Return values

0 SOAP_OK

Others See SOAP error

4.1.3.10 int MXCommon__GetModuleTemperatureValueAndStatus (xsd__unsignedLong *ulOption*, struct MXCommon__GetModuleTemperatureValueAndStatusResponse * *Response*)

Parameters

`[in]` ***ulOption*** : Reserved

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

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

- ulInfo : Reserved

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

[in] ***dMinimalWarningLevel*** : Minimal temperature warning level in Degree : 5 to 60 Degree Celsius

[in] ***dMaximalWarningLevel*** : Maximal temperature warning level in Degree : 5 to 60 Degree Celsius

[in] ***ulOption*** : Reserved

[out] ***Response*** • sResponse.iReturnValue : Return value
- 0 : success
- -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

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

Parameters

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

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

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

Return values

0 SOAP_OK

Others See SOAP error

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

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

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

Parameters

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

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

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

Return values

0 SOAP_OK

Others See SOAP error

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

Parameters

`[in]` ***ulOption*** : Reserved

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

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

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

[out] ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.16 int MXCommon__TestCustomerID (void * _, struct MXCommon__TestCustomerIDResponse * *Response*)

Parameters

[in] _ : No Input

[out] ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

• bValueArray : non encrypted value array [16 bytes of random data]

• bCryptedValueArray : Encrypted value array [16 bytes of the encrypted random data]

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

[in] ***ulLowTime*** : Number of microseconds since the begin of the second

[in] ***ulHighTime*** : Number of seconds since the Epoch (1st January,1970)

[out] ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.18 int MXCommon__SysToHardwareClock (void * _, struct MXCommon__Response * *Response*)

Parameters

`[in]` _ No input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

4.1.3.19 int MXCommon__HardwareClockToSys (void * _, struct MXCommon__Response * *Response*)

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

Parameters

`[in]` _ No input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

Parameters

`[in]` _ : No input parameter

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

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

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

`[in]` _ : no input parameter

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.22 int MXCommon__GetAutoConfigurationFile (void * _, struct MXCommon__GetAutoConfigurationFileResponse * *Response*)

Parameters

`[in]` _ : No input parameter

`[out]` ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)
  - -100 : Error of the read of the auto configuration file

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

• bArray : Array of Bytes of the file

- ulEOF : End of file flag

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

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

Parameters

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

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

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

Parameters

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

Return values

SOAP_OK SOAP call success
otherwise SOAP protocol error

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

Parameters

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

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

[in] ***ulNbrOfCycle*** : Number of timer cycle

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

[in] ***ulGenerateTriggerMode*** :

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

[in] ***ulOption01*** : Define the source of the trigger

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

[in] ***ulOption02*** : Define the edge of the hardware trigger who generates a trigger action

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

[in] ***ulOption03*** : Define the number of trigger events before the action occur

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

[in] ***ulOption04*** : Reserved

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

  - 0 : success
  - -1: system error (see syserrno)
  - -2: not available time base
  - -3: timer reload value can not be greater than 65535
  - -4: minimum time reload is 5 us
  - -5: Number of cycle can not be greater than 65535
  - -6: Generate trigger mode error
  - -100: Init timer error
  - -101: Start timer error

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror(). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

[in] ***ulOption01*** : Reserved

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

  - 0 : success
  - -1: system error (see syserrno)
  - -100: Start/Stop timer error

- sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror(). May be ENOSYS : Function not implemented.

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.27 int MXCommon__GetConfigurationBackupFile (void * _, struct MXCommon__FileResponse * *Response*)

Parameters

`[in]` _ : No input parameter

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Below is an example in pseudo-C.

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

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

Parameters

[in] ***ByteArrayInput*** : Array of Bytes of the file

[in] ***ulEOF*** : End of file flag

[out] ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: system error (see syserrno)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

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

The changes are immediately active.

Parameters

[in] _ : No input parameter

[out] ***Response*** • sResponse.iReturnValue : Return value
  - 0 : success
  - -1: string PreviousUser is invalid
  - -2: string PreviousPassword is invalid
  - -3: string NewUser is invalid
  - -4: string NewPassword is invalid
  - -5: authentification failed
  - -100: system error while saving tokens (use syserrno for more information)

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

• sArray : message returned by the auto configuration start

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

Warning

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

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

Parameters

[in] ***SubsystemID*** sub-system numerical ID

[out] ***Response*** • sResponse.iReturnValue : Return value
- 0 : success
- -1: system error while executing the request (see syserrno)
- -2: invalid parameter SubsystemID

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

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

[out] ***Response*** • sResponse.iReturnValue :Return value
- 0 : success
- -1: system error while executing the request (see syserrno)
- -2: invalid parameter SubsystemName

• sResponse.syserrno : System-error code. The value of the libc "errno" code, see MXCommon__Strerror().

• Value The numerical ID of the sub-system "SubsystemName".

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

[in] ***SubsystemID*** sub-system numerical ID

[in] ***StateName*** state symbolic name.

[out] ***Response*** • sResponse.iReturnValue : Return value
- 0 : success
- -1: system error while executing the request (see syserrno)
- -2: invalid parameters SubsystemID or StateName

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.33 int MXCommon__GetSubsystemNameFromID (xsd__unsignedLong *SubsystemID*, struct MXCommon__ByteArrayResponse * *Response*)

Parameters

`[in]` ***SubsystemID*** sub-system numerical ID.

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

`[in]` ***SubsystemID*** sub-system numerical ID.

`[in]` ***StateID*** sub-system numerical ID.

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

[in] ***ulOption01,:*** not used, set it to 0

[in] ***ulOption02,:*** not used, set it to 0

[out] ***Response*** • sArray : Option information string
- sResponse Composed of iReturnValue and syserrno

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

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

Parameters

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

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

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

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

Return values

0 SOAP_OK

Others See SOAP error

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

Parameters

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

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

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

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

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

ulValue State of the supermaster mode

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

Return values

0 SOAP_OK

Others See SOAP error

4.1.3.38 int MXCommon__SetFilterChannels (struct xsd__base64Binary * *ChannelList,* struct MXCommon__Response * *Response*)

Parameters

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

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

Return values

SOAP_OK SOAP call success

otherwise SOAP protocol error

4.1.3.39 int MSXE17xx__DigitalIOGetNumber (void * _, struct MSXE17xx__DigitalIOGetNumberResponse * *Response*)

Parameters

[in] ***None***

[out] ***Response*** :

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

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

Returns

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

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

Parameters

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

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

- 0 : input
- 1 : output

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

4.1.3.41 int MSXE17xx__DigitalIOReadChannelValue (xsd__unsignedLong *ulChannel*, struct MSXE17xx__unsignedLongResponse * *Response*)

Parameters

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

[out] ***Response*** :

iReturnValue :

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

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

- 0
- 1

Returns

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

4.1.3.42 int MSXE17xx__DigitalIOReadAllChannelsValue (void * _, struct MSXE17xx__unsignedLongResponse * *Response*)

Parameters

[in] _ : no input parameter

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

[in] ***ulChannelValue*** : Channel value

- 0
- 1

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

4.1.3.44 int MSXE17xx__DigitalIOWriteAllChannelsValue (xsd__unsignedLong *ulChannelValue,* struct MSXE17xx__Response * *Response*)

Parameters

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

[in] ***ulOption*** : reserved

[out] ***Response*** :

iReturnValue :

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

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

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

- 0 : no short circuit
- 1 : short circuit

Returns

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

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

Parameters

[in] ***ulOption*** : reserved

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

4.1.3.48 int MSXE17xx__IOWatchdogInitAndStart (xsd__unsignedLong *ulTimeBase*, xsd__unsignedLong *ulTimeValue*, xsd__unsignedLong *ulOption1*, xsd__unsignedLong *ulOption2*, struct MSXE17xx__Response * *Response*)

Parameters

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

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

[in] ***ulOption1*** : Reserved

[in] ***ulOption2*** : Reserved

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

4.1.3.49 int MSXE17xx__IOWatchdogStopAndRelease (xsd__unsignedLong *ulOption*, struct MSXE17xx__Response * *Response*)

Parameters

[in] ***ulOption*** : reserved

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

4.1.3.50 int MSXE17xx__IOWatchdogGetStatusAndValue (xsd__unsignedLong *ulOption,* struct MSXE17xx__IOWatchdogGetStatusAndValueResponse ∗ *Response*)

Parameters

[in] ***ulOption*** : Reserved

[out] ***Response*** :

iReturnValue :

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

ulStatus : current status information

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

ulValue : current value information (0 to 0xFFFF)
ulInfo : reserved
syserrno : system-error code (the value of the libc "errno" code)

Returns

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

4.1.3.51 int MSXE17xx__MFCommonGetSubModuleFunctionality (xsd__unsignedLong *ulMFModuleIndex,* struct MSXE17xx__unsignedLongResponse ∗ *Response*)

Parameters

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

[out] ***Response*** :

ulValue :

- 0: Incremental counter
- -1: PWM

sResponse.iReturnValue :

- 0: means the remote function performed OK

- -1: means an system error occured (check errno in this case)
- -2: Multifunction sub module index selection error

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

Returns

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

4.1.3.52 int MSXE17xx__MFCommonSetInputsFilter (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulInputAFilterValue,* xsd__unsignedLong *ulInputBFilterValue,* xsd__unsignedLong *ulInputCFilterValue,* xsd__unsignedLong *ulInputDFilterValue,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__Response * *Response*)

Parameters

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

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

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

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

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

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

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

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

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

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

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

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

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

[in] ***ulActivationFlag*** :

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

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

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

Bit 0, Hardware trigger gate :

- 0 : Hardware trigger gate is disabled
- 1 : Hardware trigger gate is enabled

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

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

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

Parameters

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

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

[in] ***ulTimeOutTimeBase*** : Define a Time out : permit to receive the data from the FIFO before the FIFO level is reached.

Time base of the timer (0: disabled, 1 for us, 2 for ms, 3 for s)

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

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: FIFO level value is wrong
- -4: Time out time base selection error

- -5: Time out value can not be null, if a time base is selected
- -100: Set FIFO level kernel function error

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

Returns

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

4.1.3.56 int MSXE17xx__MFSinCosInit (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulMode,* xsd__unsignedLong *ulSignalPeriod,* xsd__unsignedLong *ulResolution,* xsd__unsignedLong *ulValueFormat,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__MFSinCosInitResponse ∗ *Response*)

4.1.3.57 int MSXE17xx__MFSinCosInitEx (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulMode,* xsd__double *dSignalPeriod,* xsd__unsignedLong *ulResolution,* xsd__unsignedLong *ulValueFormat,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__MFSinCosInitExResponse ∗ *Response*)

Parameters

[in] ***ulModuleIndex*** : The module to initialize (0 to 3).

[in] ***ulMode*** : Measure mode

- 0 : Fast measure. The measure is fast,
  always 250 kHz but the measure range is smaller.
- 1 : Full range. The measure is slow but the maximal range is used.
  See in the table called
  "Max. input frequency in corresponding with the input ulResolution"
  in order to set the measure frequency.

[in] ***dSignalPeriod*** : Signal period.

[in] ***ulResolution*** : Reslution to use for the measure (binary value)

Max. input frequency in corresponding with the input ulResolution:

Resolution	Max. Freq. Hz.	Compatible with
16	250000	fast mode and full range mode
25	26000	fast mode
32	162500	fast mode and full range mode
40	16300	fast mode and full range mode
50	26000	fast mode
64	81300	fast mode and full range mode
80	16300	fast mode and full range mode
100	26000	fast mode and full range mode
125	20800	fast mode

```
| 128        | 40600          | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 160        | 16300          | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 200        | 26000          | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 250        | 20800          | fast mode                     |
+------------+----------------+-------------------------------+
| 256        | 20300          | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 320        | 16300          | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 400        | 13000          | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 500        | 10400          | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 512        | 10200          | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 800        | 6500           | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 1000       | 5200           | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 1024       | 5100           | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 1600       | 3300           | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 2000       | 2600           | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 2048       | 2540           | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 4096       | 1270           | fast mode and full range mode |
+------------+----------------+-------------------------------+
| 8192       | 635            | fast mode and full range mode |
+------------+----------------+-------------------------------+
*
```

[in] ***ulValueFormat*** : Output format of the measure

- 0 : Raw data.
- 1 : Standardized in mm.

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

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

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

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

[out] ***Response*** : ***ulMaxInputFrequency :*** Return the maximal input frequency that can be used (in Hz).

sResponse.iReturnValue :

- 0 : No error.
- -1 : means an system error occured
- -2 : Multifunction sub module index selection error.
- -3 : Multifunction sub module is not a SinCos module.
- -4 : Wrong mode.
- -5 : Wrong signal period.
- -6 : Wrong resolution.

- -7 : The resolution is not supported by the selected mode.
- -8 : Wrong format.
- -9 : Auto gain calibration error.
- -100 : Kernel function error (see syserrno).

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

Returns

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

4.1.3.58 int MSXE17xx__MFSinCosRead (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__MFSinCosReadResponse * *Response*)

4.1.3.59 int MSXE17xx__MFSinCosReadEx (xsd__unsignedLong *ulModuleIndex,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* xsd__unsignedLong *ulOption03,* xsd__unsignedLong *ulOption04,* struct MSXE17xx__MFSinCosReadExResponse * *Response*)

Parameters

[in] ***ulModuleIndex*** : The module to use (0 to 3).

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

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

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

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

[out] ***Response*** : ***iReturnValue :*** Return an error number

- 0 : No error.
- -1 : means an system error occured
- -2 : Multifunction sub module index selection error.
- -3 : Multifunction sub module is not a SinCos module.
- -4 : Sinus / Cosinus module not initialised
- -100 : Read Sinus Cosinus kernel function error (see syserrno).

dValue : Measured value, as raw or converted in the selected format.

ulMeasureError : Measure error.

0 : No error.

1 : Amplitude error.

2 : Frequency error (in fast mode is this error not relevant).

3 : Amplitude and frequency error.

Returns

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

4.1.3.60 int MSXE17xx__MFSinCosClear (xsd__unsignedLong *ulModuleIndex*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, struct MSXE17xx__Response * *Response*)

Parameters

[in] ***ulModuleIndex*** : The module to clear (0 to 3).

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

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

[out] ***Response*** : ***iReturnValue :*** Return an error number

- 0 : No error.
- -1 : means an system error occured
- -2 : Multifunction sub module index selection error.
- -3 : Multifunction sub module is not a SinCos module.
- -4 : Sinus / Cosinus module not initialised
- -100 : Clear Sinus Cosinus kernel function error (see syserrno).

Returns

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

4.1.3.61 int MSXE17xx__MFSinCosRelease (xsd__unsignedLong *ulModuleIndex*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, struct MSXE17xx__Response * *Response*)

Parameters

[in] ***ulModuleIndex*** : The module to clear (0 to 3).

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

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

[out] ***Response*** : ***iReturnValue :*** Return an error number

- 0 : No error.
- -1 : means an system error occured
- -2 : Multifunction sub module index selection error.
- -3 : Multifunction sub module is not a SinCos module.
- -4 : Sinus / Cosinus module not initialised
- -100 : Release Sinus Cosinus kernel function error (see syserrno).

Returns

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

4.1.3.62 int MSXE17xx__MFSinCosInitHardwareTrigger (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulEdgeSelection,* xsd__unsignedLong *ulCount,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulEdgeSelection*** : Front selection

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

[in] ***ulCount*** :Define the number of trigger events before the action occur

1 : all trigger event start the action

max value : 65535

[in] ***ulOption01*** : Hardware trigger gate, if enabled hardware trigger is not activated until MSXE17xx__EnableDisableHardwareTriggerGate is called

- 0 : Hardware trigger gate is not used
- 1 : Hardware trigger gate is used

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Trigger edge selection error
- -5: Trigger count selection error
- -6: Sinus / Cosinus module not initialised
- -7: Hardware trigger already initialised
- -8: ulOption01 parameter is wrong must be 0 or 1
- -100: Init hardware trigger kernel function error

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

Returns

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

4.1.3.63 int MSXE17xx__MFSinCosReleaseHardwareTrigger (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulOption01,* struct MSXE17xx__Response * *Response*)

Parameters

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Hardware trigger not initialised
- -6: Hardware trigger used and can not released
- -100: Release hardware trigger kernel function error

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

Returns

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

4.1.3.64 int MSXE17xx__MFSinCosInitIndex (xsd__unsignedLong *ulMFModuleIndex*, xsd__unsignedLong *ulEdgeSelection*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulEdgeSelection*** : Front selection

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

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Index edge selection error
- -5: Sinus / Cosinus module not initialised
- -6: Index already initialised
- -100: Init index kernel function error

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

Returns

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

4.1.3.65 int MSXE17xx__MFSinCosReleaseIndex (xsd__unsignedLong *ulMFModuleIndex*, xsd__unsignedLong *ulOption01*, struct MSXE17xx__Response * *Response*)

Parameters

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Index not initialised
- -6: Index used and can not released
- -100: Release Index kernel function error

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

Returns

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

4.1.3.66 int MSXE17xx__MFSinCosInitAndEnableLatch (xsd__unsignedLong *ulMFModuleIndex*, xsd__unsignedLong *ulLatchSource*, xsd__unsignedLong *ulCondition*, xsd__unsignedLong *ulAutoMode*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, struct MSXE17xx__Response * *Response*)

```
For each latch the data server send a 5 DWORD frame with following informations:
  DWORD 0 : Time stamp micro s
  DWORD 1 : Time stamp s
  DWORD 2 :
     D1-D0   : Sub module index (0 to 3)
     D31-D16 : Sub module functionality (2)
  DWORD 3 : Event mask
     D30-D0 :
        2: Hardware trigger latch occur
        3: Synchro input latch occur
        4: Index input latch occur
     D31 :
        0: No error occur
        1: Amplitude or Frequency error occur.
  DWORD 4 :
     D31-D0: Counter value (DWORD) if the selected output format of the measure i
      s raw data
     D31-D0: Current position in mm (FLOAT) if the selected output format of the
      measure is standardized in mm.
```

Parameters

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

[in] ***ulLatchSource*** : Latch source.

- 0: Index input

- 1: Hardware trigger
- 2: Synchro input

[in] ***ulCondition*** : Previously condition for accept the latch source

- 0: No previously condition required
- 1: Index input condition required (Only if index input not selected selected for the latch source)
- 2: Hardware trigger condition required (Only if hardware trigger not selected selected for the latch source)
- 3: Synchro input condition required (Only if synchro input not selected selected for the latch source)

[in] ***ulAutoMode*** : Action mode

- 0: Do not use auto mode (action is done only once)
- 1: Use auto mode (action is done continously)

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Latch logic already initialised
- -6: Latch source selection error
- -7: Previously condition selection error
- -8: Auto mode selection error
- -9: Hardware trigger not initialised. Refer to MSXE17xx__MFSinCosInitHardwareTrigger
- -10: Index input not initialised. Refer to MSXE17xx__MFSinCosInitIndex
- -11: Can not be used for the "fast measure" mode
- -100: Init and enable counter latch kernel function error

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

Returns

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

4.1.3.67 int MSXE17xx__MFSinCosDisableAndReleaseLatch (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulLatchSource,* xsd__unsignedLong *ulOption01,* struct MSXE17xx__Response * *Response*)

Parameters

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

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

- 0: Index input

- 1: Hardware trigger
- 2: Synchro input

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

[out] ***Response*** :

iReturnValue :

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

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

Returns

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

4.1.3.68 int MSXE17xx__MFSinCosInitAndEnableClear (xsd__unsignedLong *ulMFModuleIndex,* xsd__unsignedLong *ulClearSource,* xsd__unsignedLong *ulCondition,* xsd__unsignedLong *ulAutoMode,* xsd__unsignedLong *ulOption01,* xsd__unsignedLong *ulOption02,* struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulClearSource*** : Clear source.

- 0: Index input
- 1: Hardware trigger
- 2: Synchro input

[in] ***ulCondition*** : Previously condition for accept the clear source

- 0: No previously condition required
- 1: Index input condition required (Only if index input not selected selected for the clear source)
- 2: Hardware trigger condition required (Only if hardware trigger not selected selected for the clear source)
- 3: Synchro input condition required (Only if synchro input not selected selected for the clear source)

[in] ***ulAutoMode*** : Action mode

- 0: Do not use auto mode (action is done only once)
- 1: Use auto mode (action is done continously)

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Clear logic already initialised
- -6: Clear source selection error
- -7: Previously condition selection error
- -8: Auto mode selection error
- -9: Hardware trigger not initialised. Refer to MSXE17xx__MFSinCosInitHardwareTrigger
- -10: Index input not initialised. Refer to MSXE17xx__MFSinCosInitIndex
- -11: Can not be used for the "fast measure" mode
- -100: Init and enable counter clear kernel function error

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

Returns

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

4.1.3.69 int MSXE17xx__MFSinCosDisableAndReleaseClear (xsd__unsignedLong *ulMFModuleIndex*, xsd__unsignedLong *ulClearSource*, xsd__unsignedLong *ulOption01*, struct MSXE17xx__Response * *Response*)

Parameters

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

[in] ***ulClearSource*** : Clear source to disable and release.

- 0: Index input
- 1: Hardware trigger
- 2: Synchro input

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Clear logic not initialised
- -6: Clear source selection error
- -100: Disable and release counter clear register kernel function error

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

Returns

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

4.1.3.70 int MSXE17xx__MFSinCosInitAndEnableCompareLogic (xsd__unsignedLong *ulMFModuleIndex*, xsd__double *dValue*, xsd__unsignedLong *ulMode*, xsd__unsignedLong *ulSynchroTrigger*, xsd__unsignedLong *ulOption01*, xsd__unsignedLong *ulOption02*, struct MSXE17xx__Response * *Response*)

```
For each compare the data server send a 5 DWORD frame with following informations
      :
  DWORD 0 : Time stamp micro s
  DWORD 1 : Time stamp s
  DWORD 2 :
     D1-D0   : Sub module index (0 to 3)
     D31-D16 : Sub module functionality (2)
  DWORD 3 : Event mask
     D30-D0 :
        0: Compare occur
     D31 :
        0: No error occur
        1: Amplitude or Frequency error occur.
  DWORD 4 :
     D31-D0: Counter value (DWORD) if the selected output format of the measure i
      s Raw data
     D31-D0: Current position in mm (FLOAT) if the selected output format of the
      measure is standardized in mm.
```

Parameters

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

[in] ***dValue*** : compare value :

- 0 to 0xFFFFFFFF if raw data seleced
- Position in mm if the measure is standardized in mm

[in] ***ulMode*** : compare mode

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

[in] ***ulSynchroTrigger*** • 0 : no synchro trigger

- 1 : generates a synchro trigger when condition is true

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

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Compare value error
- -4: Compare mode error
- -5: Synchro trigger error
- -6: Multifunction sub module is not a SinCos module.
- -7: Sinus / Cosinus module not initialised
- -8: Compare logic already initialised
- -9: Can not be used for the "fast measure" mode
- -100: Init and enable counter compare kernel function error

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

Returns

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

4.1.3.71 int MSXE17xx__MFSinCosDisableAndReleaseCompareLogic (xsd__unsignedLong *ulMFModuleIndex,* struct MSXE17xx__Response * *Response*)

Parameters

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

[out] ***Response*** :

iReturnValue :

- 0: means the remote function performed OK
- -1: means an system error occured
- -2: Multifunction sub module index selection error
- -3: Multifunction sub module is not a SinCos module.
- -4: Sinus / Cosinus module not initialised
- -5: Compare logic not initialised
- -100: Disable counter compare value kernel function error

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

Returns

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

Index