EventMask Objects

Introduction

The **EventMask** determines which event types will be sent from the controller to the Notify object. By setting bits in the EventMask, specific events can be sent to the Notify object when they occur or by clearing specific bits, the events can be ignored. By default, no Events will occur until the EventMask is set.

The Event, EventMgr, and Notify objects are used with the EventMask to handle controller events.

Data Types

MPIEventMask	An array of unsigned longs

Macros

mpiEventMask**ALL** / meiEventMask**ALL**	Set all events within the event mask.
mpiEventMask**AND_ASSIGN**	Assign dst all events masked by both *src* and *dst*.
mpiEventMask**ASSIGN**	Assign the value of event mask *src* to the event mask *dst*.
mpiEventMask**AXIS** / meiEventMask**AXIS**	Set all MPIAxis events within the event mask.
mpiEventMask**BIT**	Set mask to only handle events of type *type*.
mpiEventMask**BITS_IN_ELEMENT**	Define the number of bits in each data element of MPIEventMask.
mpiEventMask**BitGET**	Reports if a mask is set to handle events of type type.
mpiEventMask**BitSET**	Sets mask to handle events of type *type*.
mpiEventMask**BIT_POSITION_MASK**	Returns an element's bit-mask for the specified event type.
mpiEventMask**CLEAR**	Set mask to handle no events.
mpiEventMask**COMPLEMENT**	Change the value of every bit within the event mask.
mpiEventMask**EXTERNAL**	Set external events within the event mask.
mpiEventMask**GET**	Reports if a mask is set to handle events of type *type*.
mpiEventMask**IS_EQUAL**	Tests the equality of two event masks.
mpiEventMask**MOTION** / meiEventMask**MOTION**	Set all MPIMotion events within the event mask.
mpiEventMask**MOTOR** / meiEventMask**MOTOR**	Set all MPIMotor events within the event mask.
mpiEventMask**OR_ASSIGN**	Add all events masked by src to the event mask dst.
mpiEventMask**RECORDER**	Sets all MPIRecorder events within the event mask.
mpiEventMask**SET**	Set mask to handle events of type *type*.
mpiEventMask**SET_ALL**	Set mask to handle all events whose type is enumerated less than type.

Constants

MPIEventMask**ELEMENTS**	Define the number of data elements in a MPIEventMask
MPIEventMask**ELEMENT_TYPE**	Define what the data type MPIEventMask is comprised of.

Copyright © 2002
Motion Engineering

MPIEventMask

MPIEventMask

```
#define MPIEventMaskELEMENTS    (2)
typedef unsigned long MPIEventMaskELEMENT_TYPE;
typedef MPIEventMaskELEMENT_TYPE MPIEventMask[MPIEventMaskELEMENTS];
```

Description

EventMask is an array of unsigned longs, with a length defined by MPIEventMaskELEMENTS. Each bit in the array represents a mask for a particular event. Be sure to always use the mpiEventMask…(…) macros to set or clear the event masks.

See Also

MPIEventType | mpiEventMaskBIT

mpiEventMaskALL / meiEventMaskALL

mpiEventMaskALL

Declaration

```
#define mpiEventMaskALL(mask),mpiEventMaskSET_ALL((mask)
                                MPIEventTypeLAST)
```

Required Header stdmpi.h

Description **EventMaskALL** sets all events within the event mask.

meiEventMaskALL

Declaration

```
#define meiEventMaskALL(mask)
 mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER0, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER1, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER2, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER3, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER4, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER5, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER6, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER7, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeMOTION_OUT_OF_FRAMES, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeMOTION_RESERVED0, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeIN_POSITION_COARSE, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeIN_POSITION_FINE, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeAT_TARGET, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeFRAME, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeAXIS_RESERVED0, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeAXIS_RESERVED1, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypePOWER_FAILURE, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_TORQUE, 1)
```

Required Header stdmei.h

Description **EventMaskALL** sets all MEI events within the event mask

See Also MPIEventType

mpiEventMaskAND_ASSIGN

Declaration

```
#define mpiEventMaskAND_ASSIGN(dst,src)((dst)[0] = ((dst)[0] & (src)[0])), \
```

Required Header

stdmpi.h

Description

EventMaskOR_ASSIGN assigns ***dst*** all events masked by both ***src*** and ***dst***.

See Also

MPIEventMask

mpiEventMaskASSIGN

Declaration

```
#define mpiEventMaskASSIGN(dst,src)((dst)[0] = (src)[0]), \
```

Required Header stdmpi.h

Description **EventMaskASSIGN** assigns the value of event mask ***src*** to the event mask ***dst***.

See Also MPIEventMask

mpiEventMaskAXIS / meiEventMaskAXIS

mpiEventMaskAXIS

Declaration `#define mpiEventMaskAXIS(mask)(mask)`

Required Header stdmpi.h

Description **EventMaskAXIS** sets all MPIAxis events within the event mask.

meiEventMaskAXIS

Declaration

```
#define meiEventMaskAXIS(mask)
        mpiEventMaskBitSET((mask), MEIEventTypeIN_POSITION_COARSE, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeIN_POSITION_FINE, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeAT_TARGET, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeFRAME, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeAXIS_RESERVED0, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeAXIS_RESERVED1, 1)
```

Required Header stdmei.h

Description **EventMaskAXIS** sets all MEIAxis events within the event mask.

See Also MPIEventMask

mpiEventMaskBIT

Declaration

```
#define mpiEventMaskBIT(mask,type)(((mask)[0] = 0); ((mask)[1] = 0); \
((mask)[mpiEventMaskWORD(type)]= mpiEventMaskBIT_POSITION_MASK(type)))
```

Required Header

stdmpi.h

Description

EventMaskBIT sets mask to only handle events of type ***type***.

See Also

MPIEventMask | mpiEventMaskBIT | mpiEventMaskBitGET | mpiEventMaskBitSET | mpiEventMaskBITS_IN_ELEMENT | mpiEventMaskBIT_POSITION_MASK

mpiEventMaskBITS_IN_ELEMENT

mpiEventMaskBITS_IN_ELEMENT

```
#define mpiEventMaskBITS_IN_ELEMENT ((unsigned long)
(sizeof(MPIEventMaskELEMENT_TYPE) * 8))
```

Description **EventMaskBITS_IN_ELEMENT** defines the number of bits in each data element of MPIEventMask.

See Also MPIEventMask | mpiEventMaskBIT | mpiEventMaskBitGET | mpiEventMaskBitSET | mpiEventMaskBIT_POSITION_MASK

mpiEventMaskBitGET

Declaration

```
#define mpiEventMaskBitGET(mask,bit)(((mask)[mpiEventMaskWORD(bit)] &
        mpiEventMaskBIT_POSITION_MASK(bit)) ? 1 : 0)
```

Required Header

stdmpi.h

Description

EventMaskAXIS sets all MPIAxis events within the event mask.

Returns

TRUE if mask is set to handle events of type ***type***. Otherwise, it returns FALSE.

See Also

MPIEventMask | mpiEventMaskBIT | mpiEventMaskBitSET | mpiEventMaskBIT_POSITION_MASK | mpiEventMaskBITS_IN_ELEMENT

mpiEventMaskBitSET

Declaration

```
#define mpiEventMaskBitSET(mask,bit,value)      (((value) == 0) ? \
        ((mask)[mpiEventMaskWORD(bit)] &= ~mpiEventMaskBIT_POSITION_MASK(bit))
              : \
        ((mask)[mpiEventMaskWORD(bit)] |= mpiEventMaskBIT_POSITION_MASK(bit)))))
```

Required Header

stdmpi.h

Description

EventMaskBitSET sets mask to handle events of type ***type***.

See Also

MPIEventMask | mpiEventMaskBitGET | mpiEventMaskBIT | mpiEventMaskBIT_POSITION_MASK | mpiEventMaskBITS_IN_ELEMENT

mpiEventMaskBIT_POSITION_MASK

Declaration

```
#define mpiEventMaskBIT_POSITION_MASK (type)
                    ((MPIEventMaskELEMENT_TYPE)(0x1 <<
                      mpiEventMaskBIT_POSITION(type)))
```

Required Header stdmpi.h

Description

type	the event type (MPIEventType / MEIEventType)

Returns an element's bit-mask for the specified event type.

See Also MPIEventMask | mpiEventMaskBIT | mpiEventMaskBitGET | mpiEventMaskBitSET | mpiEventMaskBITS_IN_ELEMENT

mpiEventMaskCLEAR

Declaration

```
#define mpiEventMaskCLEAR(mask)((mask)[0] = 0), \ ((mask)[1] = 0)
```

Required Header

stdmpi.h

Description

EventMaskCLEAR sets mask to handle no events.

See Also

MPIEventMask

mpiEventMaskCOMPLEMENT

Declaration

```
#define mpiEventMaskCOMPLEMENT(mask)((mask)[0] = ~(mask)[0]),
                                \ ((mask)[1] = ~(mask)[1])
```

Required Header

stdmpi.h

Description

EventMaskCOMPLEMENT() changes the value of every bit within the event mask. If the ***mask*** is origininally set to handle an event, then ***mask*** will be set to not handle the event after the call to mpiEventMaskCOMPLEMENT(). Likewise, If the ***mask*** is origininally set to not handle an event, then mask will be set to handle the event after the call to mpiEventMaskCOMPLEMENT().

See Also

MPIEventMask

mpiEventMaskEXTERNAL

Declaration

```
#define mpiEventMaskEXTERNAL(mask)mpiEventMaskBitSET((mask),
                                MPIEventTypeEXTERNAL, 1)
```

Required Header stdmpi.h

Description **EventMaskEXTERNAL** sets external events within the event mask.

See Also MPIEventMask

mpiEventMaskGET

Declaration

```
#define mpiEventMaskGET(mask,type)
mpiEventMaskBitGET(mask,type)
```

Required Header

stdmpi.h

Description

Returns	TRUE if mask is set to handle events of type ***type***. Otherwise, it returns FALSE.

See Also

MPIEventMask | mpiEventMaskSET

mpiEventMaskIS_EQUAL

Declaration

```
#define mpiEventMaskIS_EQUAL(mask1,mask2)
        (((mask1)[0] == (mask2)[0]) &&
        ((mask1)[1] == (mask2)[1])))
```

Required Header stdmpi.h

Description **EventMaskIS_EQUAL** tests the equality of two event masks.

Returns TRUE if mask1 is the same as mask2. Otherwise it returns FALSE.

See Also MPIEventMask

mpiEventMaskMOTION / meiEventMaskMOTION

mpiEventMaskMOTION

Declaration

```
#define mpiEventMaskMOTION(mask)
        mpiEventMaskBitSET((mask), MPIEventTypeMOTION_DONE, 1), \
        mpiEventMaskBitSET((mask), MPIEventTypeMOTION_AT_VELOCITY, 1)
```

Required Header

stdmpi.h

Description

EventMaskMOTION sets all MPIMotion events within the event mask.

meiEventMaskMOTION

Declaration

```
#define meiEventMaskMOTION(mask)        mpiEventMaskBitSET((mask),
        MEIEventTypeMOTION_OUT_OF_FRAMES, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeMOTION_RESERVED0, 1)
```

Required Header

stdmei.h

Description

EventMaskMOTION sets all MEIMotion events within the event mask.

See Also

MPIEventMask

mpiEventMaskMOTOR / meiEventMaskMOTOR

mpiEventMaskMOTOR

Declaration

```
#define mpiEventMaskMOTOR(mask)
        mpiEventMaskBitSET((mask), MPIEventTypeAMP_FAULT,1), \
        mpiEventMaskBitSET((mask), MPIEventTypeHOME, 1), \
        mpiEventMaskBitSET((mask), MPIEventTypeLIMIT_ERROR, 1), \
        mpiEventMaskBitSET((mask), MPIEventTypeLIMIT_HW_NEG, 1), \
        mpiEventMaskBitSET((mask), MPIEventTypeLIMIT_HW_POS, 1), \
        mpiEventMaskBitSET((mask), MPIEventTypeLIMIT_SW_NEG, 1), \
        mpiEventMaskBitSET((mask), MPIEventTypeLIMIT_SW_POS, 1), \
        mpiEventMaskBitSET((mask), MPIEventTypeENCODER_FAULT, 1)
```

Required Header stdmpi.h

Description **EventMaskMOTOR** sets all MPIMotor events within the event mask.

meiEventMaskMOTOR

Declaration

```
#define meiEventMaskMOTOR(mask)
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER0, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER1, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER2, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER3, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER4, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER5, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER6, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_USER7, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypePOWER_FAILURE, 1), \
        mpiEventMaskBitSET((mask), MEIEventTypeLIMIT_TORQUE, 1)
```

Required Header stdmei.h

Description **EventMaskMOTOR** sets all MEIMotor events within the event mask.

See Also MPIEventMask

mpiEventMaskOR_ASSIGN

Declaration

```
#define mpiEventMaskOR_ASSIGN(dst,src)  ((dst)[0] = ((dst)[0] | (src)[0])), \
                                        ((dst)[1] = ((dst)[1] | (src)[1]))
```

Required Header

stdmpi.h

Description

EventMaskOR_ASSIGN adds all events masked by ***src*** to the event mask ***dst***.

See Also

MPIEventMask

mpiEventMaskRECORDER

Declaration

```
#define mpiEventMaskRECORDER(mask)      mpiEventMaskBitSET((mask),
        MPIEventTypeRECORDER_FULL, 1), \
        mpiEventMaskBitSET((mask), MPIEventTypeRECORDER_DONE, 1)
```

Required Header

stdmpi.h

Description

EventMaskRECORDER sets all MPIRecorder events within the event ***mask***.

See Also

MPIEventMask

mpiEventMaskSET

Declaration

```
#define mpiEventMaskSET(mask,type)
        mpiEventMaskBitSET(mask,type,1)
```

Required Header stdmpi.h

Description **mpiEventMaskSET** sets mask to handle events of type ***type***.

See Also MPIEventMask | mpiEventMaskGET

mpiEventMaskSET_ALL

Declaration

```
#define mpiEventMaskSET_ALL(mask,type)  ((mpiEventMaskWORD(type) == 0) ? \
                        ((mask)[0] |= mpiEventMaskBIT_POSITION_MASK(type) - 1) : \
                        ((mask)[0] |= 0xFFFFFFFF)), \
                        ((mpiEventMaskWORD(type) == 0) ? \
                        ((mask)[1] |= 0x0) : \
                        ((mask)[1] |= mpiEventMaskBIT_POSITION_MASK(type) - 1))
```

Required Header

stdmpi.h

Description

EventMaskSET_ALL sets ***mask*** to handle all events whose type is enumerated less than ***type***.

See Also

MPIEventMask

MPIEventMaskELEMENTS

MPIEventMaskELEMENTS

```
#define MPIEventMaskELEMENTS    (2)
```

Description

EventMaskELEMENTS defines the number of data elements in a MPIEventMask.

See Also

MPIEventMask

MPIEventMaskELEMENT_TYPE

MPIEventMaskELEMENT_TYPE

```
typedef unsigned long MPIEventMaskELEMENT_TYPE
```

Description

EventMaskELEMENT_TYPE defines what the data type MPIEventMask is comprised of.

See Also

MPIEventMask