Xmp Module

Introduction

The **Xmp** module provides the low-level interface between the controller and the MPI library. It defines the shared memory access between the controller's processor and the host CPU. It also contains several hardware constants and maximum values for resources. ALL data transactions between the MPI and controller are defined by this module.

The MPI provides a layer between the application code and the controller. It protects the application from *xmp.h* changes, hides controller complexity, handles semaphore locking, performs data validation and range checking, plus many other features. Normally, an application does not need the Xmp module. An application should ALWAYS use the MPI to access the controller. In some cases, the MPI may not have methods/structures to support a controller feature. For example, a custom feature may require support for direct access to the MPI, but it has not been yet been developed. In these cases, an application can use xmp.h and mpiControlMemoryGet/Set to directly access the controller.

WARNING!
The *xmp.h* file is version dependent. Make sure to ONLY use the xmp.h that was included with the MPI and controller firmware software package. Using mismatched xmp.h defines can cause unexpected and potentially dangerous behavior!

Be aware that the xmp.h is always changing. It is an internal file, used by the controller firmware and MPI. It is optimized for memory allocation and performance for the controller's processor. As new features are developed and improved, the xmp.h is modified. If you use xmp.h defines in your application code, make sure to check for changes when upgrading to new software releases.

Data Types

MEIXmp**SwitchType**

MEIXmpSwitchType

Declaration

```
typedef enum {
    MEIXmpSwitchTypeNONE,
    MEIXmpSwitchTypeMOTION_ONLY,
    MEIXmpSwitchTypeWINDOW,
    MEIXmpSwitchTypeUSER,
} MEIXmpSwitchType;
```

Required Header: xmp.h

Description

MEIXmpSwitchType is an enumeration for gain scheduling that determines the gain scheduling mode. Only MEIXmpSwitchTypeNONE and MEIXmpSwitchTypeMOTION_ONLY are available in standard firmware types.

Gain Scheduling is a feature that switches filter gains for the acceleration, deceleration, constant velocity, and idle states of motion. The post filters are not affected by gain scheduling. Standard algorithms are used with gain scheduling (PID, PIV). To change the gain scheduling type from *none* (uses only the gains in gain table index 0), use MEIFilterConfig. GainSwitchType is set with mpiFilterConfigSet(...).

When setting filter gain parameters using mpiFilterGainGet(...) and mpiFilterGainSet(...), use the gain index value to write to a gain index of your choosing.

MEIXmpSwitchTypeNONE	Default value in factory default firmware. This mode uses mpiFilterGainIndexSet() and mpiFilterGainIndexGet() to manipulate the gain index manually, if desired.
MEIXmpSwitchTypeMOTION_ONLY	Switch gains based on controller's switching algorithm.

MEIFilterGainIndex (go to MEIFilterGainIndex)

MEIFilterGainIndexNO_MOTION	When command velocity = 0
MEIFilterGainIndexACCEL	When command acceleration > 0
MEIFilterGainIndexDECEL	When command acceleration < 0
MEIFilterGainIndexVELOCITY	When command velocity > 0 and command acceleration = 0 The firmware automatically takes care of this switching. Be aware when checking the gain index, that the firmware can change the gain index in real time.

Description

Gain Scheduling is a feature that switches filter gains for the acceleration, deceleration, constant velocity, and idle states of motion. The post filters are not affected by gain scheduling. Standard algorithms are used with gain scheduling (PID, PIV). To change the gain scheduling type from NONE (uses only the gains in gain table index 0), use MEIFilterConfig.GainSwitchType, which is set with mpiFilterConfigSet(...).

Use mpiFilterConfigSet(...) to change MEIFilterConfig.GainSwitchType to one of the MEIXmpSwitchType enumerations to change the gain scheduling mode.

See Also

MPIFilterConfig | mpiFilterConfigGet | mpiFilterConfigSet | MEIFilterGainIndex | mpiFilterGainIndexSet | mpiFilterGainIndexGet | mpiFilterGainGet | mpiFilterGainSet