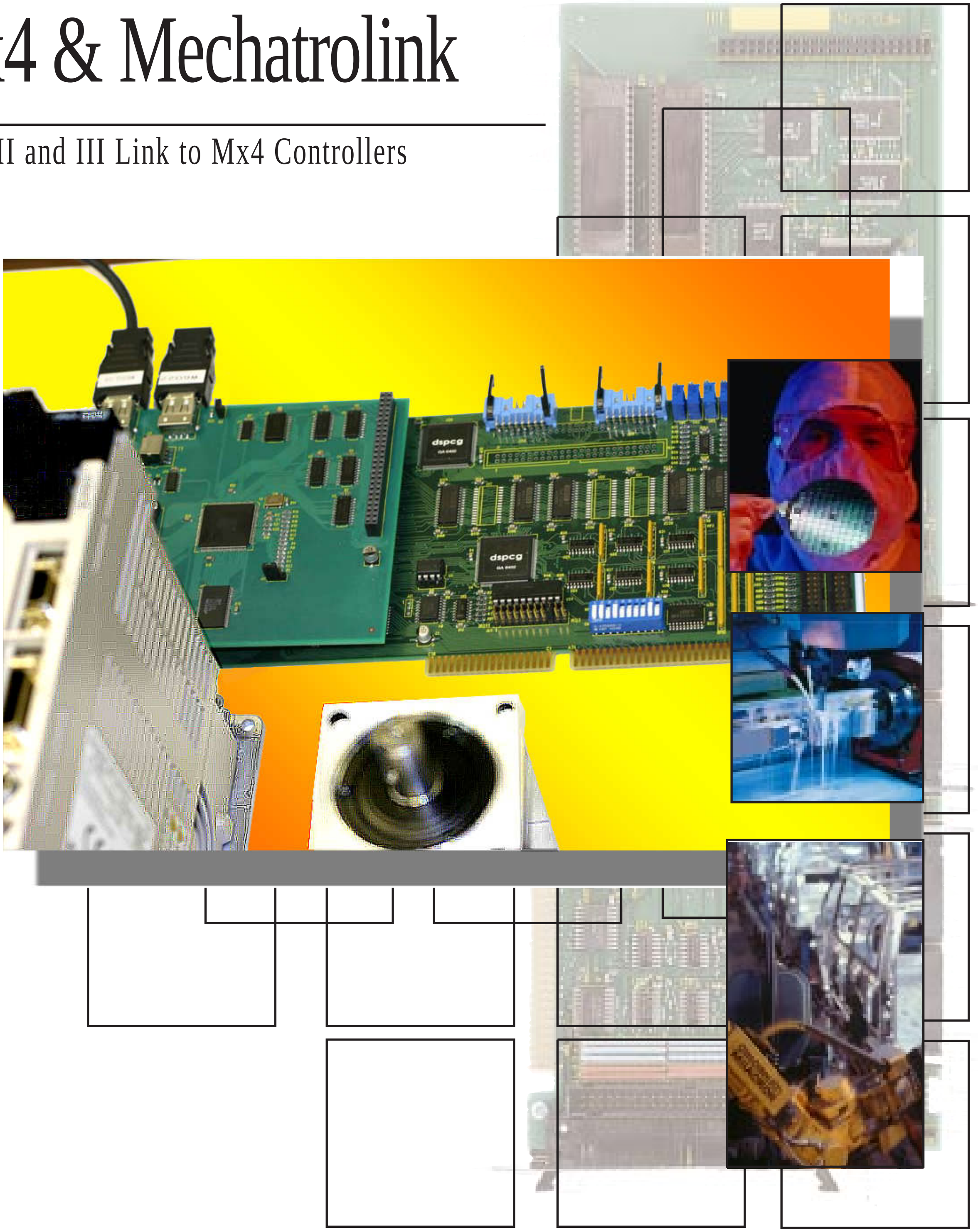
Mx4 & Mechatrolink

Sigma II and III Link to Mx4 Controllers



DSP Control Group, Inc.

Mx4 & Mechatrolink

User's Guide

V1.2

© Copyright 2000-2004 DSP Control Group, Inc.
4445 W 77th Street
Minneapolis, MN 55435
Phone: (952) 831-9556
FAX: (952) 831-4697

The authors and those involved in the manual's production have made every effort to provide accurate, useful information.

Use of this product in an electro-mechanical system could result in a mechanical motion that could cause harm. DSP Control Group, Inc. is not responsible for any accident resulting from misuse of its products.

DSPCG makes no warranty or condition, either expressed or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding the licensed materials.

Contents

1 Mechatrolink & Mx4

Mechatrolink - Mx4 System Description

Mechatrolink II is a digital link between DSP Control Group's Mx4 motion controller and Yaskawa's Sigma II or III drives. Other than making communication digital *and* your drives reachable by the Mx4 motion controller at a high bandwidth, this link offers no advantage.

Other digital links offered by DSP Control Group at high bandwidth are: Firewire (IEEE 1394), DSPNET (a high-speed IEEE 802.3 physical layer) and Ethernet.

In this section, we will describe what you must do to convert your familiar analog motion controller to a digitally linked system. We start with Mechatrolink System Wiring, then go on with the Drive Initialization DLL - through which you must execute a few instructions on the Mx4 motion controller (which in turn initializes the drives through the link). From that point on you may use all drives similar to any analog drive.

Finally, you may use a Windows application program that will let you set up, test and run your motion system before programming your application.

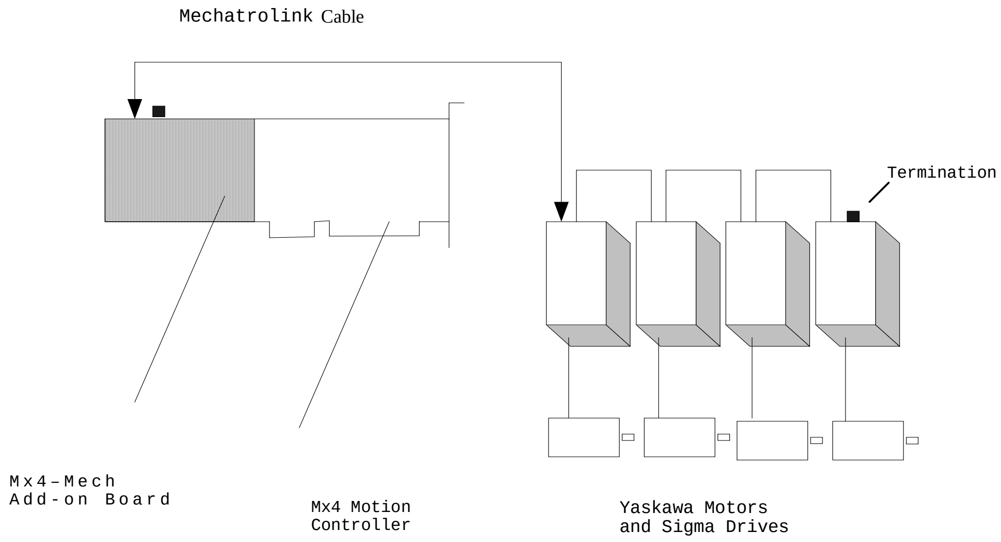
Mechatrolink System Wiring

Minimum System Requirement:

i) Mx4 Motion Controller With Mx4-Mech add-on board (ordered through DSPCG or their distributors)

ii) One to 4 Yaskawa motors and Sigma servo-drives with Mechatriolink II option (e.g. Sigma III products)

iii) Mechatrolink cables and Network Terminations. (ordered through Yaskawa or their distributors)



Mechatrolink System Wiring Diagram

Drive Initialization DLL

Drive Initialization commands, installed with Mx4pro software, enable the Sigma drives to accept basic motion control commands (e.g. velocity command). These commands in order of necessity and priority are:

Must Run - before using Sigma drives with the Mx4 motion controller:

MECH_INIT	enables the mechatrolink link
DRIVEMAP	maps the drive number onto an axis
MECH_CONNECT	establishes communication between control and drives
MECH_FEEDBACK	enables feedback mechanism on mechatrolink drive
MECH_CLOOP	enables closed loop operation on the dive.
MECH_BLOCK	blocks/unblocks the above commands to go through

Note: You must observe the order of execution for the instructions listed above

Optionally Run - to either disconnect the drives, poll for or, clear a fault condition.

MECH_DISCONNECT	software disconnects the system elements
MECH_ALARM	reads the alarm (fault) from the drives
MECH_CLRALARM	clear the faults

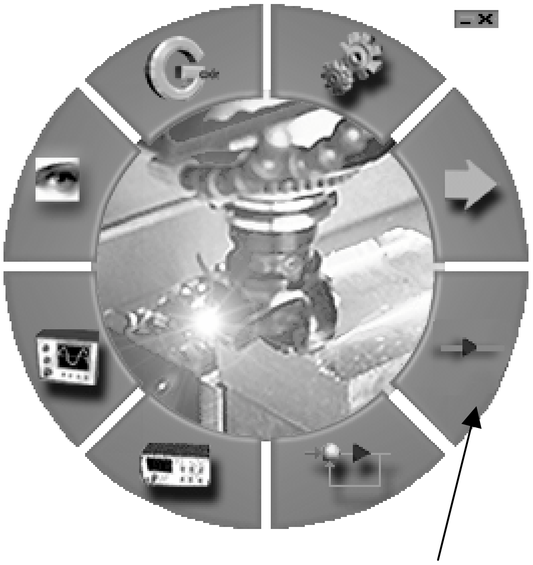
Further description for each command is available in the Instruction Set section of this short manual

Once the “Must Run” commands are executed, to the Mx4 motion controller, a Mechareolink drive appears as a common analog drive. Indeed, with the “Optionally Run” commands, unlike an analog drive, you may poll for a drive fault and clear it as needed.

Mechatrolink Windows Application Program

This Windows application program allows you to setup, test and run your Mechatrolink System before running your application. This test program consists of software connecting, initializing, checking for fault and status condition and, clearing the fault. This software is to be used in conjunction with Mx4pro tuning program provided by DSP Control Group. Further Information on this program is provided in the next chapter.

2 Mechatrolink Windows



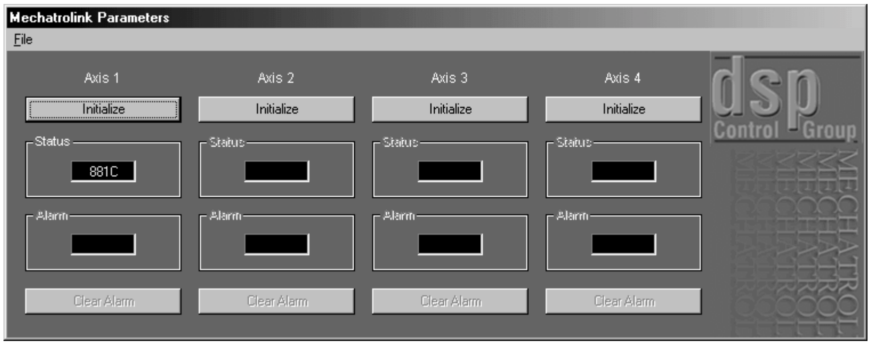
Click button to open the Mechatrolink Window

An important window in the Mx4pro Development Tools is the Mechatrolink Window. The Mechatrolink Window allows the user to setup and manage a Mechatrolink drive connected to a Mx4 system. This chapter will explain how to use all of the features of the Mechatrolink Window.

Mechatrolink Window

The Mechatrolink Window allows the user to select which axis will be connected to a Mechatrolink drive. It also allows the monitoring of the drive status and the ability to clear drive fault or alarm. To open the Mechatrolink Window click on the Mechatrolink Window button on the main Mx4pro Development Tools. The following window will appear:

Mx4 Mechatrolink Window



Status 881C

Fig. 2-1: Mechatrolink Window

Pressing the ***Initialize*** button is all that a user has to do to setup a Mechatrolink drive. The status word for the drive is displayed in hexadecimal format in the ***Status*** field (See the Mechatrolink User Guide for detailed status explanation). If a fault occurred the Mechatrolink Alarm code will be displayed in the ***Alarm*** field (See the Mechatrolink User Guide for detailed alarm explanation). To enable the drive after an alarm, press the ***Clear Alarm*** button, then the ***Initialize*** button if safe.

3 Integrating Mechatrolink With Executive

Using Mx4-Mechatrolink add-on card (and its associated firmware revisions) allows you to use a Yaskawa Sigma III amplifier to be used with an Mx4 similar to any analog amplifier. However, because the link between the Mx4 and Sigma III is now a digital network – as an OEM user of this system - you are responsible for its initialization and monitoring of the system in an executive program. You can perform these functions using the Mx4's Windows DLL or Real Time Commands (RTC). These functions let you:

1. Initialize the servo-amplifier
2. Program the combined amplifier/motion controller gains (through a single ctrl command)
3. Check and Clear if necessary the alarm or fault condition
4. Check the servo-amplifier status
5. Reset the system

Because of the added flexibility – you *MUST NOT* use this product unless you are fully aware of system safety issues and how added functions must be incorporated into your program to address these issues. DSP Control Group, Inc. is not responsible for any harm that may result from user's lack knowledge of the system safety or misuse of its product.

To help you understand these functions, at RTC or DLL level, we categorize them by a title. You must run the outlined command sequence to implement the function.

Initialize Servo-amplifier

MECH_INT ……………………… (RTC or DLL)
DRIVEMAP ……………………… (RTC or DLL)
MECH_CONNECT……………… (DLL – see conversion to RTC)
MECH_FEEDBACK ……………. (DLL – see conversion to RTC)
MECH_CLOOP …………………. (DLL – see conversion to RTC)

Read Alarm

MECH_ALARM …………………. (DLL – see conversion to RTC)

Clear Alarm

MECH_CLRALARM	(DLL – see conversion to RTC)
MECH_FEEDBACK	(DLL – see conversion to RTC)
MECH_CLOOP	(DLL – see conversion to RTC)

Reset System

MECH_DISCONNECT	(DLL – see conversion to RTC)
RESET	(RTC or DLL)

Read Amplifier Status

MECH_ALARM ……………..……. (DLL – see conversion to RTC)

Programing Amplifier Gains

CTRL ………………………………(RTC or DLL)

Programming the familiar Mx4 control law RTC (CTRL)
On an Mx4_Mech add-on card option defines the entire

tuning parameters. That is, one CTRL instruction provides combinecd tuning for both amplifier and motion controller.

Blocking Instructions To Get To The Amplifier

MECH_BLOCK ……………….. (RTC or DLL)

The next section in this chapter provides information to convert a DLL function to its respective RTC sequence. If you are not a Windows 2000/98/XP user, this section helps you write your own driver.

Converting DLL Functions to RTC

To properly initialize and enable a Mechatrolink drive connection, specific instruction sequences must be followed as was described in the previous section. The purpose of this section is to share necessary information on how each DLL function is constructed from a sequence of a RTCs. The variable n represents the axis.

DLL Function To Initialize Amplifier

To initialize a Mechatrolink connection, in addition to two RTCs, as described in previous section, the user must run three DLL commands. The following describes how these commands can be converted to RTC.

MECH_CONNECT

```
MECH_WRITE( n, 1, 0 )
MECH_WRITE( n, 2, 0 )
MECH_WRITE( n, 3, 0 )
MECH_WRITE( n, 4, 0 )
MECH_WRITE( n, 5, 0 )
MECH_WRITE( n, 6, 0 )
MECH_WRITE( n, 7, 0 )
MECH_WRITE( n, 8, 0 )
MECH_WRITE( n, 9, 0 )
MECH_WRITE( n, 10, 0 )
MECH_WRITE( n, 11, 0 )
MECH_WRITE( n, 12, 0 )

MECH_WRITE( n, 3, 0021h )
MECH_WRITE( n, 4, 0001h )
MECH_WRITE( n, 1, 000Eh )
```

MECH_FEEDBACK

```
MECH_WRITE( n,  1,  0 )
MECH_WRITE( n,  2,  0 )
MECH_WRITE( n,  3,  0 )
MECH_WRITE( n,  4,  0 )
MECH_WRITE( n,  5,  0 )
MECH_WRITE( n,  6,  0 )

MECH_WRITE( n, 3, 0824h )
MECH_WRITE( n, 4, 1D02h )
MECH_WRITE( n, 1, 0002h )
```

Run this PARREAD repeatedly until the returned value is 0002h.

```
PARREAD( 41h, n, 01h )

MECH_WRITE( n, 3, 0000h )
MECH_WRITE( n, 4, 0000h )
MECH_WRITE( n,  1, 0000h )

MECH_WRITE( n, 3, 0825h )
MECH_WRITE( n, 4, 0302h )
MECH_WRITE( n,  1, 0002h )
```

Run this PARREAD repeatedly until the returned value is 0002h.

```
PARREAD( 41h, n, 01h )

MECH_WRITE( n, 3, 0000h )
MECH_WRITE( n, 4, 0000h )
MECH_WRITE( n, 1, 0000h )


MECH_WRITE( n, 7, 00FEh )
MECH_WRITE( n, 1, 0030h )
```

Run this PARREAD repeatedly until the returned value is 0030h.

```
PARREAD( 41h, n, 01h )
```

MECH_CLOOP

```
MECH_WRITE( n, 1, 0 )
MECH_WRITE( n, 2, 0 )
MECH_WRITE( n, 3, 0 )
MECH_WRITE( n, 4, 0 )
MECH_WRITE( n, 5, 0 )
MECH_WRITE( n, 6, 0 )

MECH_WRITE( n, 1, 0031h )
```

Run this PARREAD repeatedly until the returned value is 0031h.

```
PARREAD( 41h, n, 01h )

MECH_WRITE( n, 1, 0000h )
MECH_WRITE( n, 1, 0035h );
```

Run this PARREAD repeatedly until the returned value is 0035h.

```
PARREAD( 41h, n, 01h )
```

DLL/RTC Function To Block Instructions

Once the above routines have been completed for all Mechatrolink drives, and before any motion is begun, the user should run the MECH_BLOCK RTC. This prevents accidental modification of the Mechatrolink data buffer, which could cause corruption of motion commands.

DLL Function To Reset System

To reset a Mechatrolink system, the user should bring motion on all drives to a halt, and unblock the system using the MECH_BLOCK RTC. Then the following sequence should be run to disconnect all drives.

MECH_DISCONNECT

```
MECH_WRITE( n,  1,  0 )
MECH_WRITE( n,  2,  0 )
MECH_WRITE( n,  3,  0 )
MECH_WRITE( n,  4,  0 )
MECH_WRITE( n,  5,  0 )
MECH_WRITE( n,  6,  0 )
MECH_WRITE( n,  7,  0 )
MECH_WRITE( n,  8,  0 )
MECH_WRITE( n,  9,  0 )
MECH_WRITE( n,  10,  0 )
MECH_WRITE( n,  11,  0 )
MECH_WRITE( n,  12,  0 )

MECH_WRITE( n,  3, 0021h )
MECH_WRITE( n,  4, 0001h )
MECH_WRITE( n,  1, 000Fh )
```

Run this PARREAD repeatedly until the returned value is 000Fh.

```
PARREAD( 41h, n, 01h )
```

DLL Function To Read Alarm

MECH_ALARM

Mechatrolink drives provide error data for a variety of fault conditions. This data is displayed as an alarm number on the seven-segment display on the front of the drive, and is also made available to the host system over the Mectrolink network.

To determine if an error has been returned, the user should run the following RTC command:

PARREAD(41h, n, 2).

If the least significant bit of the returned word is set, an alarm is present. The user should then run the following command to determine the alarm number:

PARREAD(41h, n, 1).

The high byte of the returned word contains the high two digits of the alarm number in binary coded decimal format.

DLL Function To Clear Alarm

MECH_CLRALARM

To clear an alarm, all axes are to be brought to a stop, the system is unblocked using the MECH_BLOCK RTC, and the following sequence is run.

```
MECH_WRITE( n,  1,  0 )
MECH_WRITE( n,  2,  0 )
MECH_WRITE( n,  3,  0 )
MECH_WRITE( n,  4,  0 )
MECH_WRITE( n,  5,  0 )
MECH_WRITE( n,  6,  0 )
```

Run Mechatrolink disconnect command.
MECH_WRITE(n, 1, 0006h)

Run this PARREAD repeatedly until the returned value is 0006h.
PARREAD (41h, n, 01h)

MECH_ALARM (a DLL Function)

FUNCTION Read status and return alarm code if one occurred

SYNTAX `MECH_ALARM n, status, error`

If used as a function, the function will return (long) zero if successful, nonzero if error.

ARGUMENTS

n	long value specifying the axis
status	pointer to long value status returned from Mechatrolink
error	pointer to long value error returned from Mechatrolink

error = 0	No alarm
error	Mechatrolink alarm code

DESCRIPTION

This command gets the status word from a Mechatrolink drive and returns the alarm code if one occurred.

SEE ALSO `MECH_CLRALARM, MECH_INIT, DRIVEMAP, MECH_CONNECT`

APPLICATION

Command Sequence Example

1) MECH_INIT command is required before running this instruction.

2) DRIVEMAP command is required before running this instruction.

3) MECH_CONNECT command is required before running this instruction.

EXAMPLE

Read the status word and alarm code from Mechatrolink for axis 2.

```
MECH_ALARM 2, &STATUS, &ERROR
```

MECH_CLRALARM (a DLL Function)

FUNCTION Clears Mechatrolink alarm

SYNTAX `MECH_CLRALARM n`

If used as a function, the function will return (long) zero if successful, nonzero if error.

ARGUMENTS

n	long value specifying the axis

DESCRIPTION

This command will clear an alarm on a Mechatrolink drive.

SEE ALSO `MECH_ALARM, MECH_INIT, DRIVEMAP, MECH_CONNECT`

APPLICATION

Command Sequence Example

1) MECH_INIT command is required before running this instruction.

2) DRIVEMAP command is required before running this instruction.

3) MECH_CONNECT command is required before running this instruction.

EXAMPLE

Clear alarm on axis 2.

```
MECH_CLRALARM 2
```

MECH_INIT (a DLL Function)

FUNCTION Enables Mechatrolink syetm

SYNTAX `MECH_INIT`

If used as a function, the function will return a (long) zero if successful, nonzero if error.

ARGUMENTS

none

DESCRIPTION

This command enables the Mechatrolink system for all axes in a system using an Mx4 system. It must always be sent before any other Mechatrolink commands.

SEE ALSO `DRIVEMAP,MECH_SCALE,MECH_BLOCK`

APPLICATION

A necessary function before all Mechatrolink commands.

Command Sequence Example

No preparation is required before running this instruction.

EXAMPLE

Enable the Mechatrolink system for all axes.

```
MECH_INIT
```

DRIVEMAP (DLL Function also see RTC)

FUNCTION Maps an axis number to a Mechatrolink drive number

SYNTAX `DRIVEMAP n, drv`

If used as a function, the function will return (long) zero if successful, nonzero if error.

ARGUMENTS

n	long value specifying the axis to be mapped
drv	byte value specifying the Mechatrolink drive number

DESCRIPTION

This command must be executed for each axis connected to a Mechatrolink drive. To unmap a drive set the drive number to zero.

Note – the axis and drive numbers are commonly the same. A drive number is what is set on the front pannel of a Yaskawa Sigma drive via a rotary numerically marked switch. So, if this switch is set to 3, the drive value, drv is equal to 3 and you may use the same value (i.e. 3) for n. This command must be used after MECH_INIT.

SEE ALSO `MECH_INIT, MECH_SCALE, MECH_BLOCK`

DRIVEMAP cont.

APPLICATION

A necessary function before a Mechatrolink command can be send to a drive.

Command Sequence Example

1) MECH_INIT command is required before running this instruction.

EXAMPLE

Map axes 1 and 3 to Mechatrolink drive numbers 4 and 10. Also, unmap axis 2, which would return it to normal mode.

```
BEGIN_RTC
       DRIVEMAP 1, 4
       DRIVEMAP 2, 0
       DRIVEMAP 3, 10
END_RTC
```

MECH_BLOCK (a DLL Function also see RTC)

FUNCTION Blocks any new Mechatrolink command

SYNTAX `MECH_BLOCK n, blk`

If used as a function, the function will return a (long) zero if successful and nonzero if drive error.

ARGUMENTS

n	long value specifying the axis number
blk	byte value block code

blk = 0	Mechatrolink block disabled
blk = 1	Mechatrolink block enabled

DESCRIPTION

This command is used to block some of the Mechatrolink commands so that those commands may not be accidentally executed. The user is responsible to disable the block command in order to execute one of the commands listed below (SEE ALSO).

SEE ALSO `MECH_CLOOP, MECH_CLRALARM, MECH_ENCO, MECH_CONNECT, MECH_DISCONNECT, CTRL`

APPLICATION

A necessary function to prevent the accidental change of parameters during Mechatrolink operation. This should be used before any motion commands are sent.

Command Sequence Example

No preparation is required before running this instruction.

EXAMPLE

Enable the Mechatrolink command blocking for axes 3 and 4. Also disable the Mechatrolink command blocking for axis 1.

```
BEGIN_RTC
       MECH_BLOCK 1, 0
       MECH_BLOCK 3, 1
       MECH_BLOCK 4, 1
END_RTC
```

MECH_INIT (an RTC Function)

FUNCTION Mechatrolink Enable

DPR ORDER command code

USAGE Host (command code: 91h)

ARGUMENTS

none

DESCRIPTION

This command enables the Mechatrolink system for all axes in the Mx4 firmware. It must be send before any other Mechatrolink commands.

SEE ALSO DRIVEMAP, MECH_WRITE, MECH_BLOCK

EXAMPLE

No arguments needed.

DRIVEMAP (an RTC Function)

FUNCTION Map an Axis to a Mechatrolink Drive Number

DPR ORDER command code, n , drv_1, … , drv_4

USAGE Host (command code: 92h)

ARGUMENTS

n	a single byte, bit coding the axes involved
drv_x	a single byte value the drive number to map to axis x

DESCRIPTION

This command must be executed for each axis connected to a Mechatrolink drive. It maps a Mx4 axis number to a Mechatrolink drive number. This allows the normal motion commands to be executed without any modification. To unmap an axis, send a drive number of zero.

SEE ALSO MECH_INIT, MECH_WRITE, MECH_BLOCK

EXAMPLE

Set the axes 1 and 3 to Mechatrolink drive numbers 1 and 4.

drv_1	=	1
drv_3	=	4

The values of the RTC arguments are:

n	:	05h
drv_1	:	01h
drv_3	:	04h

MECH_BLOCK (an RTC Function)

FUNCTION Map an Axis to a Mechatrolink Drive Number

DPR ORDER command code, n, blk_1, … , blk_4

USAGE Host (command code: 95h)

ARGUMENTS

n	a single byte, bit coding the axes involved	
blk_x	a single byte value block code	
	blk = 0	Mechatrolink block disabled
	blk = 1	Mechatrolink block enabled

DESCRIPTION

This command is used to block some of the Mechatrolink commands so that those commands may not be accidentally executed. The user is responsible to disable the block command in order to execute one of the commands listed below (SEE ALSO).

SEE ALSO MECH_WRITE

EXAMPLE

Enable the Mechatrolink command blocking for axis 2 and disable the Mechatrolink command blocking for axis 1.

blk_1	=	0
blk_2	=	1

The values of the RTC arguments are:

n	:	03h
blk_1	:	00h
blk_2	:	01h

MECH_WRITE (an RTC Function)

FUNCTION Write a word to Mechatrolink transmit buffer

DPR ORDER command code, n, offset, value

USAGE Host (command code: 93h)

ARGUMENTS

n	a single byte, bit coding the axes involved
offset	a single byte the offset to write in the axis transmit buffer
value	a single word the value to write into the transmit buffer

DESCRIPTION

This command is used to write a word to the Mechatrolink transmit buffer. It is used to send Mechatrolink commands (See Mechatrolink User Guide for command format) to the drive that is mapped an axis.

SEE ALSO DRIVEMAP, MECH_INIT, MECH_BLOCK

EXAMPLE

For the Mechatrolink drive mapped to axis 1 write 0Fh to offset 2.

offset	=	2
value	=	0Fh

The values of the RTC arguments are:

n	:	01h
offset	:	02h
value	:	0Fh

PARREAD (an RTC Function)

FUNCTION Parameter Readback

DPR ORDER command code, n, m, offset

USAGE Host (command code: 5Eh)

ARGUMENTS

n a single byte, which indicates the parameters to echo.

n=40h	Read word from Mechatrolink transmit buffer
n=41h	Read word from Mechatrolink receive buffer
n=42h	Drive mapping for axes 1, 2, 3, and 4

m a byte value, which codes the axis, involved

offset a byte value the offset to read

DESCRIPTION

Upon the execution of this command, Mx4 echoes the desired parameters to DPR locations 0B8h - 0BFh. "m" is echoed to DPR location 0B7h if the parameters are ready in the DPR. Parameters may take more than 5ms to echo back to the DPR. Host can use the following algorithm:

1. write m to DPR location 3C3h
2. write 0 to DPR location 0B7h
3. write RTC command code to DPR location 3C2h
4. poll DPR location 0B7h until m is echoed read the data from DPR location 0B8h - 0BFh

PARREAD cont.

DATA FORMAT

For each type of parameter, DPR locations 0B8h - 0BFh are interpreted differently. The following shows the format for each type of parameter:

1. Mechatrolink transmit buffer (n=40h)

0B8h	low byte of value at offset of axis m
0B9h	high byte of value at offset of axis m
0BAh	
:	not used
0BFh	

2. Mechatrolink receive buffer (n=41h)

0B8h	low byte of value at offset of axis m
0B9h	high byte of value at offset of axis m
0BAh	
:	not used
0BFh	

3. Drive Mapping (n=42h)

0B8h	drive number mapped to axis 1
0B9h	drive number mapped to axis 2
0BAh	drive number mapped to axis 3
0BBh	drive number mapped to axis 4
0BCh	
:	not used
0BFh	

PARREAD cont.

SEE ALSO none

EXAMPLE

Read a word from the Mechatrolink receive buffer of axis 2 at offset 3.

offset = 3

The values of the RTC arguments are:

n	:	40h
m	:	02h
offset	:	03h

Read drive mapping .

The values of the RTC arguments are:

n	:	42h