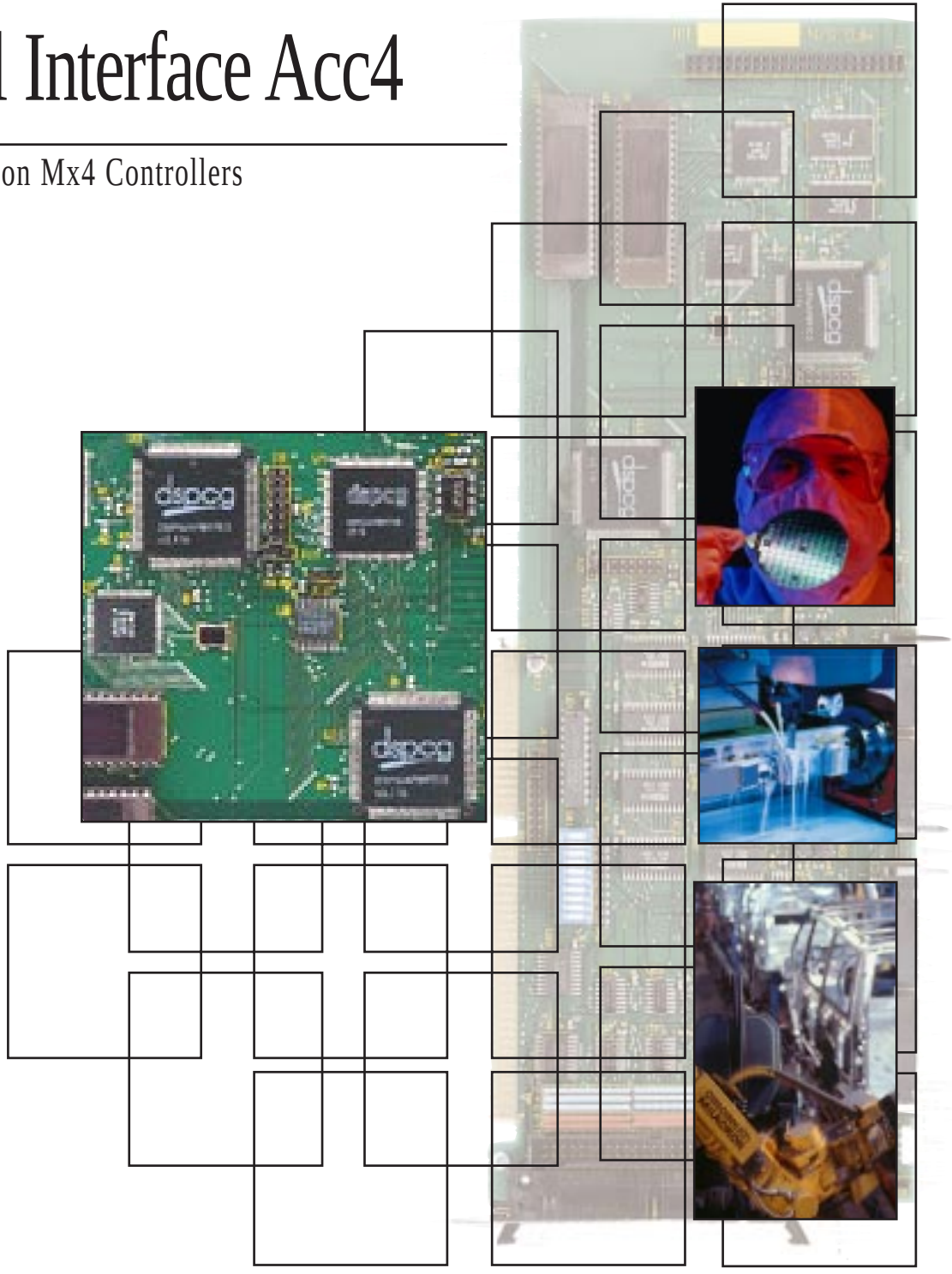
Serial Interface Acc4

Serial Link on Mx4 Controllers



DSP Control Group, Inc.

Acc4 Mx4 Options Daughterboard

User's Guide

V4.0

© Copyright 1991-1999 DSP Control Group, Inc.
PO Box 39331
Minneapolis, MN 55439
Phone: (612) 831-9556
FAX: (612) 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.

DSPL, Mx4 and Vx4++ are trademarks of DSP Control Group, Inc.

Other brand names and product names are trademarks of their respective holders.

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.

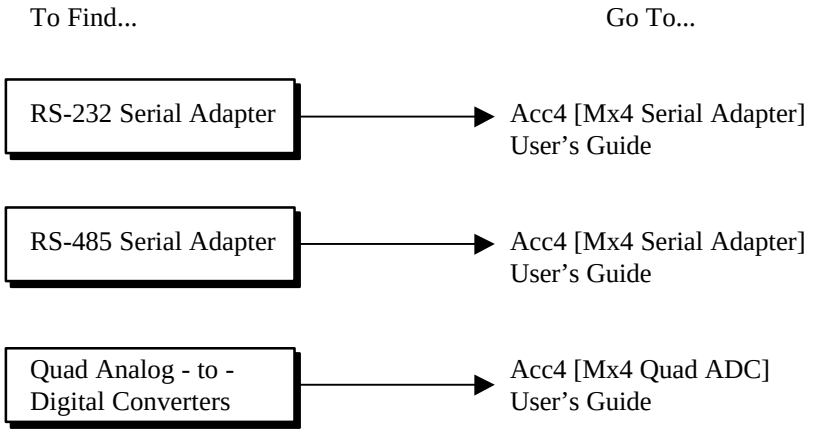
A Quick Reference

The Acc4 daughterboard includes these options:

Mx4 RS-232 Serial Adapter
Mx4 RS-485 Serial Adapter
Mx4 Quad Analog - to - Digital Converters

The Acc4 User's Guide consists of (2) separate documents: Acc4 [Mx4 Serial Adapter] User's Guide and the Acc4 [Mx4 Quad ADC] User's Guide.

The Acc4 User's Guide may include information on an option which is not pertinent to every Acc4 daughterboard, as the Acc4 is available in multiple option configurations.



Contents

This page intentionally blank.

This page intentionally blank.

1 Introduction

The Acc4 [Mx4 Serial Adapter] option enables you to use the Mx4 controller outside the computer rack as a stand-alone unit or serially programmable inside the computer. The Acc4 [Mx4 Serial Adapter] works with both PC/AT Mx4 and VME Mx4. In addition, the Acc4 [Mx4 Serial Adapter] may be used in conjunction with other Mx4 daughterboards such as VECTOR4.

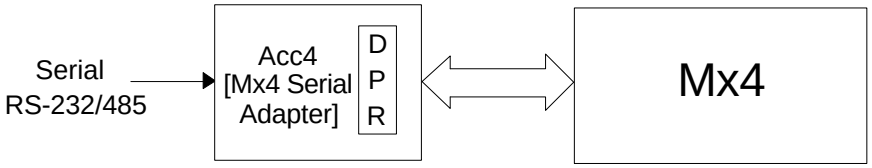


Fig. 1-1: Acc4 [Mx4 Serial Adapter] - Mx4 Communication Block Diagram

Both the PC/AT Mx4 and VME Mx4 (in their bus versions) communicate with the host computer through a Dual Port RAM (DPR) which is contained on the Mx4 board. If the user desires to communicate with the Mx4 across a RS-232 or RS-485 serial link, the Acc4 [Mx4 Serial Adapter] daughterboard must be used. It is important to note that the Acc4 [Mx4 Serial Adapter] daughter board includes a DPR to facilitate the host-serial-Mx4 communication (see Fig. 1-1).

Note : When the Acc4 [Mx4 Serial Adapter] is used with a non-stand alone Mx4 controller, the DPR on the Mx4 controller must be removed.

The DPR is located in a PLCC chip carrier socket. For PC/AT Mx4, the DPR is labeled as U20. For VME Mx4, the DPR is labeled as U28.

Note : Remember to replace the Mx4 DPR if the Mx4 is to be used in the “host bus communication” mode!

This page intentionally blank.

2 Installing the Hardware

Acc4 [Mx4 Serial Adapter] Mechanical Specifications

The Acc4 [Mx4 Serial Adapter] plugs onto the Mx4 controller via three connectors (see Fig. 2-1). The card is secured with these three connectors; however, if desired, a securing fastener may also be used.

Note : If the Acc4 [Mx4 Serial Adapter] is being used in conjunction with a VECTOR4 daughterboard, the Acc4 [Mx4 Serial Adapter] should be plugged onto the Mx4, and the VECTOR4 should be plugged onto the Acc4 [Mx4 Serial Adapter].

Important: For those users which are using the Acc4 [Mx4 Serial Adapter] with a non-standalone Mx4, the Mx4 DPR should be removed prior to installing the Acc4 [Mx4 Serial Adapter] (see *Introduction*).

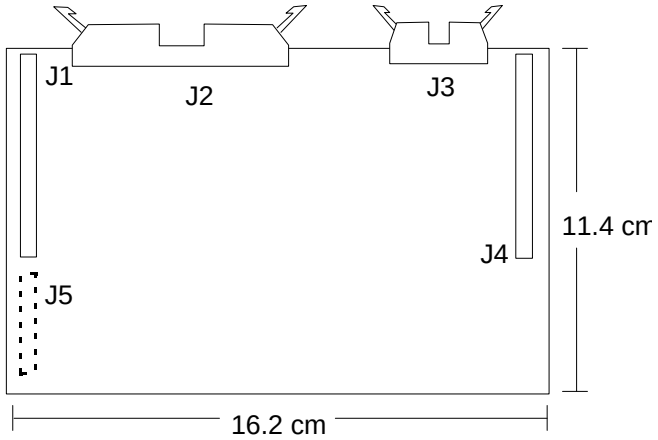


Fig. 2-1: Acc4 [Mx4 Serial Adapter] Dimensions and Connectors

Acc4 [Mx4 Serial Adapter]	To PC/AT Mx4	To VME Mx4
J1	J2	J1
J4	J4	J4
J5	J1	J6

Table 2-1: Mx4 - Acc4 [Mx4 Serial Adapter] Connections

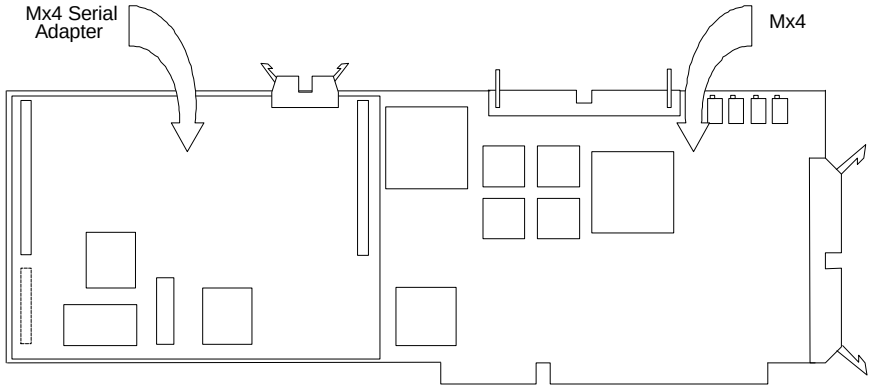


Fig. 2-2: Acc4 [Mx4 Serial Adapter] Mounted on PC/AT Mx4

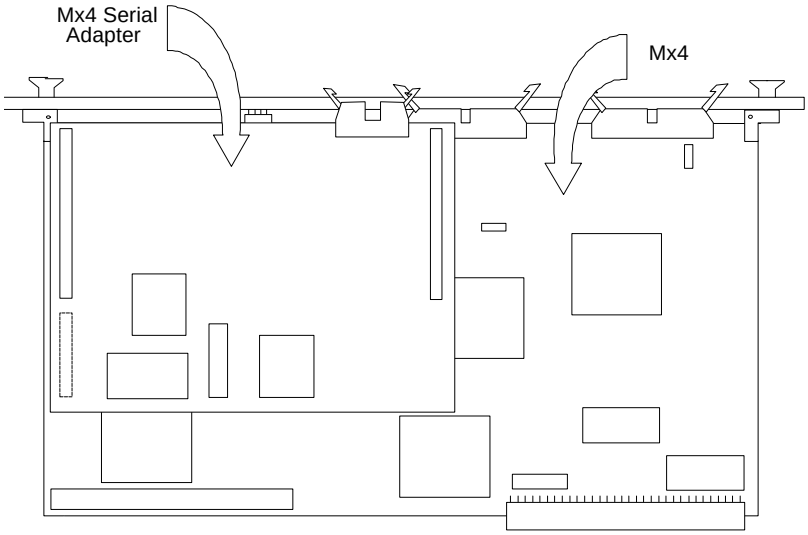


Fig. 2-3: Acc4 [Mx4 Serial Adapter] Mounted on VME Mx4

Acc4 [Mx4 Serial Adapter] Cabling

The Acc4 [Mx4 Serial Adapter] contains a single user-available connector as illustrated in Fig. 2-4. The J3 connector contains the RS-232/485 signals for interfacing the host to the Mx4 controller via serial link.

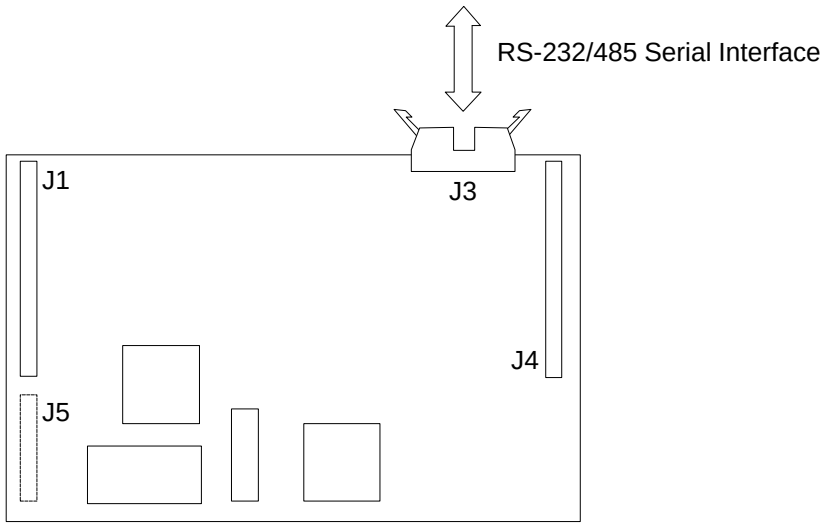


Fig. 2-4: Acc4 [Mx4 Serial Adapter] Connector Signals

J3 Connector ... RS-232/485 Serial Interface

The Acc4 [Mx4 Serial Adapter] J3 connector is a standard 0.100" 10-pin dual row header with short ejector tabs. This connector contains the Mx4-Host serial interface signals.

Table 2-2A specifies the pinout for the Acc4 [Mx4 RS-232 Serial Adapter] J3 10-pin header. Table 2-2B specifies the pinout for the Acc4 [Mx4 RS-485 Serial Adapter] J3 10-pin header. The tables include signal level (type) and I/O functionality (with respect to the Acc4 [Mx4 Serial Adapter]).

J3 Connector Pinout

PIN	SIGNAL	LEVEL	I/O	DESCRIPTION
1	RS232HIGH	RS-232 HIGH	O	high RS-232 line, may be tied to DCD
2	RS232HIGH	RS-232 HIGH	O	high RS-232 line, may be tied to DSR
3	DT	RS-232	O	RS-232 data transmit
4	nc	-	-	no connection
5	DR	RS-232	I	RS-232 data receive
6	RS232HIGH	RS-232 HIGH	O	high RS-232 line, may be tied to CTS
7	nc	-	-	no connection
8	nc	-	-	no connection
9	Digital GND	-	O	-
10	nc	-	-	no connection

Table 2-2A: Acc4 [Mx4 RS-232 Serial Adapter] J3 Connector Pinout

Note 1: The Acc4 [Mx4 RS-232 Serial Adapter] requires only the RS-232 signals RX and TX to operate. The J3 connector includes three permanently HIGH RS-232 lines that may be interfaced to the host RS-232 signals DCD, DSR and CTS in order to enable RS-232 data transmissions (if required).

Note 2: The signals that compromise the J3 connector pinout are arranged for convenient DB-9 RS-232 cable construction. On the Acc4 [Mx4 RS-232 Serial Adapter] side, a standard 10-pin ribbon cable socket connector can be used. On the host side, strip down the pin 10 signal of the ribbon so that signals 1-9 are left. Using a ribbon cable DB-9 connector, install the connector such that ribbon cable pin 1 corresponds to DB-9 pin 1. The DB-9 now includes the proper pinout for standard PC/AT-type serial port connections.

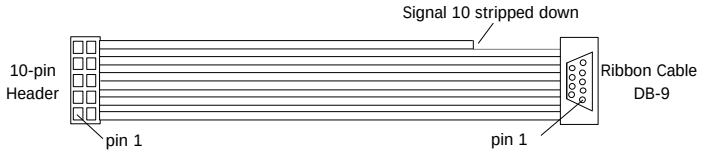


Fig. 2-5: DB-9 RS-232 Cable

J3 Connector Pinout

PIN	SIGNAL	LEVEL	I/O	DESCRIPTION
1	nc	-	-	no connection
2	nc	-	-	no connection
3	nc	-	-	no connection
4	nc	-	-	no connection
5	nc	-	-	no connection
6	nc	-	-	no connection
7	RS485B	RS-485	IO	RS-485 differential B
8	RS485A	RS-485	IO	RS-485 differential A
9	Digital GND	-	O	-
10	nc	-	-	no connection

Table 2-2B: Acc4 [Mx4 RS-485 Serial Adapter] J3 Connector Pinout

3 Mx4 Software Utilities/DSPL Support

The Acc4 [Mx4 RS-232 Serial Adapter] is supported by a complete set of software utilities which facilitate RS-232 communication. The utilities are located on the Mx4 Utilities 3.5" diskette. The utilities include:

MX4PRO.EXE	Mx4 testing and tuning utility
DSPLD.EXE	DSPL downloader
IMC.EXE	'Issue Mx4 Command" utility
DOWN_POS.EXE	position compensation table download utility
DOWN_VEL.EXE	velocity compensation table download utility
DOWN_VAR.EXE	DSPL variable table download utility
DOWN_CAM.EXE	CAM data table download utility
DOWN_CUB.EXE	internal cubic spline data download utility

The operation of the utilities is described in the README file located in the root directory of the Mx4 Utilities diskette.

The Acc4 [Mx4 Serial Adapter] includes an optional ASCII terminal interface feature. The option allows the serial adapter to communicate in either of two modes,

1. standard protocol mode
2. ASCII mode

The ASCII mode option is supported in DSPL with the PRINT and INPUT commands. These commands allow for information transfer between an executing DSPL program and an ASCII terminal. The PRINT and INPUT DSPL commands are listed as follows.

INPUT

FUNCTION Input Data to DSPL Program via ASCII Interface

SYNTAX INPUT (message, var)

USAGE DSPL (Motion)

ARGUMENTS

message	a character string, enclosed in quotes
var	DSPL Variable (VAR1-VAR64) in which to place the input data

DESCRIPTION

The INPUT command allows the DSPL program to prompt an ASCII terminal for data, and place the input data in a DSPL Variable. The DSPL motion program execution is halted at the INPUT command until data is received from the ASCII terminal.

Note: The INPUT command requires an Acc4 [Mx4 Serial Adapter], with the ASCII mode option.

SEE ALSO PRINT

APPLICATION

This command may be used in a stand-alone application, where the operator requires an ASCII terminal for data entry to a pre-loaded DSPL program.

EXAMPLE

Request from the ASCII terminal operator the axis target position valve, Px. The value should be placed in VAR31.

```
INPUT ("Enter Px: ?", VAR31)
```

PRINT

FUNCTION Output Character String to ASCII Interface

SYNTAX PRINT (message)

USAGE DSPL (Motion)

ARGUMENTS

message a character string, enclosed in quotes

DESCRIPTION

The PRINT command allows the DSPL program to output message prompts to an ASCII terminal.

Note: The PRINT command requires an Acc4 [Mx4 Serial Adapter], with the ASCII mode option.

SEE ALSO INPUT

APPLICATION

This command may be used in a stand-alone application, where the operator interface is an ASCII terminal.

EXAMPLE

Orientate a new operator to the machine interface.

PRINT (“To halt the X-axis, press the STOPX button.”)

etc.

Appendix A:

Advanced Programming

The Acc4 [Mx4 RS-232 Serial Adapter] includes a complete set of software utilities (see Chapter 3, *Software Utilities*) which provide the user with serial communication - supported utilities including Mx4Pro, the DSPL program downloader, and a set of table data download utilities. for the majority of Acc4 [Mx4 Serial Adapter] users, these utilities will provide sufficient application support.

For those users whose applications require user - customized software support (i.e.: the host-Mx4 serial communication must be an integral portion of the system's executable software), appendices A-1 and A-2 detail the Acc4 [Mx4 Serial Adapter]'s communication protocol and command set.

Important: The information contained in Appendix A-1 and Appendix A-2 is not required in order to operate the Acc4 [Mx4 RS-232 Serial Adapter] with the supplied software utilities.

This page intentionally blank.

Appendix A-1:

Serial Communication Protocol

The Host and Acc4 [Mx4 Serial Adapter] communicate via a full duplex serial port operating at a baud rate of 9600 bps. What follows describes the communication between the Host and Acc4 [Mx4 Serial Adapter] from both perspectives.

Framing

Packets and Frames

A packet is an unrestricted variable length sequence of 8-bit data bytes. The serial protocol consists of a set of rules for the exchange of packets between a master and slave. A packet is transmitted over an asynchronous serial communication channel as a frame. A frame is a sequence of 8-bit characters. Special characters are used to indicate the beginning of the frame and the end of the frame. A technique known as a byte stuffing is used to prevent these special characters from occurring within the body of the frame.

Framing Characters

The three framing characters:

- `0x80` -- Escape Character (ESC)
- `0x81` -- Start of Message (SOM)
- `0x82` -- End of Message (EOM)

Byte Stuffing

A frame always starts with a SOM character and ends with an EOM character. Each byte of the packet is encoded in the frame as either one or two characters. If the packet byte is one of the special characters it is encoded in the frame using an ESC character followed by the packet byte with the most significant bit cleared. For example `0x81` would be encoded with the characters `0x80,0x01`.

Example

Consider the following: The packet to transmit consists of these data bytes:

```
0x21, 0x82, 0x17
```

This packet contains an 0x82 which is one of the special characters and must be escaped. It is encoded into the frame below:

```
0x81, 0x21, 0x80, 0x02, 0x17, 0x82
```

Packet Format

A packet consists of a single byte header, zero or more data bytes, and a two byte CRC as shown in the figure below.

Header (1)	Data (n)	CRC (2)

The Header

The packet header specifies both the packet type and the address of a slave device. This is a multi-drop protocol which allows for a single master and multiple slaves.

Format:

Reserved (7:7)	Packet Type (4:6)	Node Address (0:3)

bit 7: Always zero.

bits 4-6: Packet Type. The four packet types and their encoding is given in Table A-1.

bits 0-3: Node Address.

Packet Type	Description
0	I0 -- Information Packet with sequence number 0
1	I1 -- Information Packet with sequence number 1
2	RESET -- Used by master to force slave to reset its sequence number
3	UA -- Unnumbered Acknowledge. Used by slave to acknowledge a RESET command.

Table A-1: Packet Type Encoding

Data

The packet can contain from zero or to 64 data bytes. The data section of the packet is used for application level commands and responses.

CRC

A 16-bit CRC is used to ensure the integrity of the packet. The last two bytes of the packet are selected so as to make the overall CRC equal to zero. The CRC is generated by the standard CRC polynomial $X^{16}+X^{12}+X^{5}+1$.

Link-Level Protocol

Overview

The link-level protocol is a master/slave protocol. It is intended to be used for multi-drop communication channels like RS-485. Slaves only transmit packets in response to command packets from the master. Each slave on the line has a unique 4-bit address. The slave must ignore packets addressed to other slaves.

The master and slave must also ignore any packet whose CRC is not zero. A single bit sequence number is used to ensure reliable communication between master and slave. An initialization procedure is used to synchronize the master and slave with the same sequence number. The slave always saves its last response to the master in a buffer. When it receives a packet from the master it checks the sequence number. If the packet does not have the expected sequence number it ignores the command and will transmit the buffered packet. If the packet has the correct sequence number it processes the command, transmits a response packet, (with the same sequence number) and increments (module 2) the sequence number it is expecting. The master follows much the same protocol. After transmitting a command packet to the slave, the master waits for a certain period of time for a response packet. If it does not see a valid response, (correct framing, node address, and CRC) within the time-out period, it will retransmit the command packet.

Note: The link-level protocol requires that in response to every command packet (I0 or I1) from the master, the slave sends back a response packet (I0 or I1).

Master State Machine

One method of specifying the protocol is to give the state machine a description of the master and the slave. See the master state transition diagram (Fig. A-1). The machine events are the reception of valid frames and a time-out. These events are listed below.

- I0 -- Reception of valid packet of type I0
- I1 -- Reception of valid packet of type I1
- U1 -- Reception of valid packet of type UA
- Time-out -- A time-out occurs

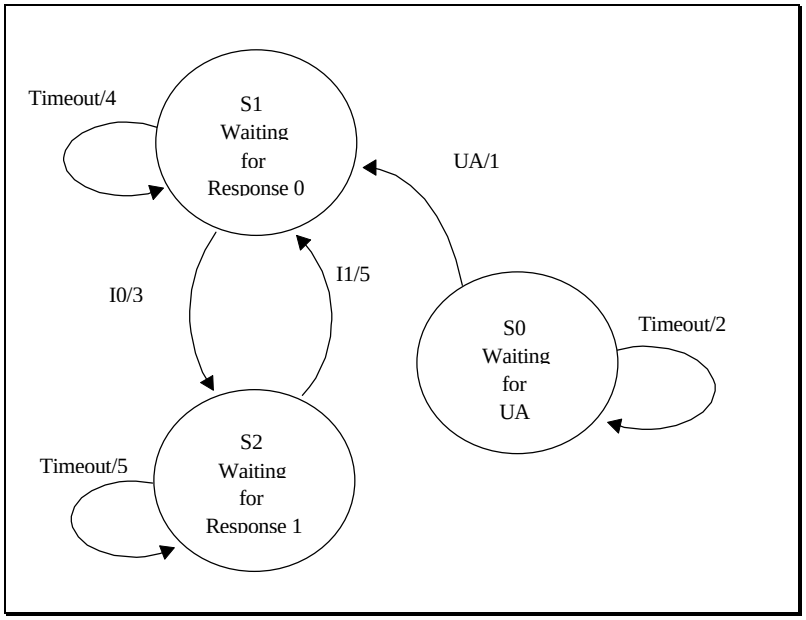


Fig. A-1: Master State Transition Diagram

State	Event	Next State	Action
S0	I0	S0	0
	I1	S0	0
	UA	S1	1
	Time-out	S0	2
S1	I0	S2	3
	I1	S1	0
	UA	S1	0
	Time-out	S1	4
S2	I0	S2	0
	I1	S1	5
	A	S2	0
	Time-out	S2	5

Table A-2: Master State Transition Table

Action Number	Description
0	Do nothing
1	Generate first command and transmit using I0
2	Transmit RESET packet
3	Process response, generate next command, transmit using I1
4	transmit command using I0
5	Process response, generate next command transmit I0
6	Transmit command using I1

Table A-3: Master Action Table

Slave State Machine

The slave state machine responds as follows:

- I0 Reception of a valid packet of type I0
- I1 Reception of a valid packet of type I1
- RESET Reception of a valid packet of type RESET

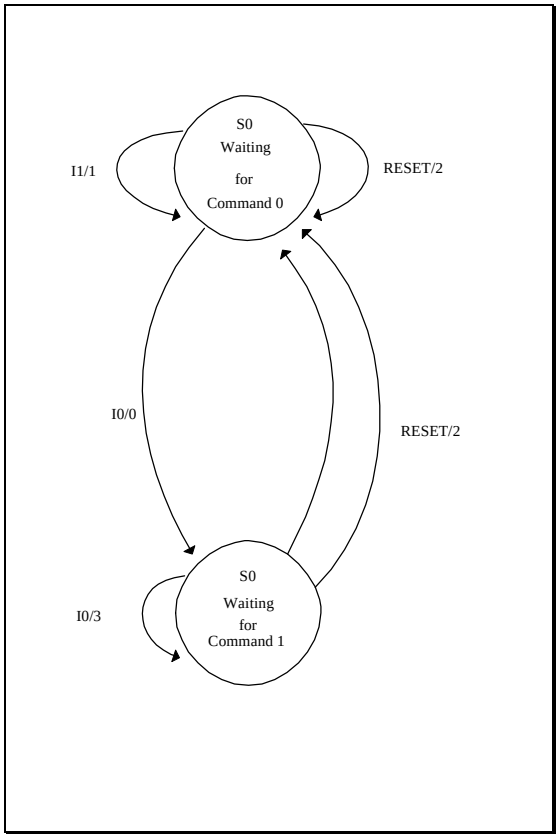


Fig. A-2: Slave State Transition Diagram

State	Event	Next State	Action
S0	I0	S1	0
	I1	S0	1
	RESET	S0	2
S1	I0	S1	3
	I1	S0	4
	RESET	S0	2

Table A-4: Slave State Transition Table

Action Number	Description
0	Process command, generate response, transmit response using I0
1	Transmit last response using I1
2	Transmit UA packet
3	Transmit last response using I0
4	Process command, generate response, transmit response using I1

Table A-5: Slave Action Table

An Example Exchange

This section contains an example exchange between the master and slave (node1) using the protocol. It should help to clarify the protocol definition. It shows the characters (in hexadecimal) making up the frames transmitted by the master and slave.

Master resets the link by sending a RESET (Type=2) packet. Slave responds with UA (Type=3) packet.

```
Master:   81 21 24 43 82
Slave:    81 31 26 72 82
```

Read Mx4 signature bytes

```
Master:   81 01 02 03 15 01 F2 CE 82
Slave:    81 01 02 4D 58 34 E9 04 82
```

Issue CTRL RTC. (Ki=100, Kp=4096, Kf=4096, Kd=1024)

```
Master:   81 11 05 62 64 00 00 10 00 10 04 00 CA BD 82
Slave:    81 11 05 60 E7 82
```

Issue MAXACC RTC. The specified acceleration is 0x8000 so byte stuffing must be used to transmit the most significant byte.

```
Master:   81 01 05 71 01 00 80 00 F4 8E 82
Slave:    81 01 05 60 E7 82
```

Issue VELMODE RTC. The specified velocity is .5 (00008000h) so byte stuffing must be used to transmit the 0x80 byte.

```
Master:   81 11 05 70 01 00 80 00 00 00 D7 57 82
Slave:    81 11 05 60 E7 82
```

Read Position, Error, and Velocity of Axis-1 with one command

```
Master:   81 01 01 04 D3 00 04 E3 00 04 F3 00 9C 5E 82
Slave:    81 01 01 01 00 00 00 02 00 00 00 03 00 00 00 29 0D 82
```

Use write command to send an RTC.

```
Master:   81 11 03 01 C3 03 01 01 C2 03 6E BD 7A 82
Slave:    81 11 03 00 21 82
```

This page intentionally blank.

Appendix A-2:

Serial Communication Commands

Command Summary

The five available application level commands are given in the table below:

Code	Description
MT_READ1	This read command allows specified sections of the Mx4 dual port ram to be read. The serial adapter checks the specified addresses to see whether they are protected by access bytes. If they are, it follows the appropriate access byte protocol.
MT_READ2	This read command allows specified sections of the Mx4 dual port ram to be read. It does not check to see whether the specified areas of RAM are protected by access bytes.
MT_WRITE1	This command is used to write data into the dual port RAM. It waits for the RTC byte to clear before transferring the data to the RAM. It is useful for loading the ring-buffer.
MT_WRITE2	This command is used to write data into the dual port RAM. It does not check the RTC byte.
MT_RTC	This command issues an RTC. It waits for the RTC byte to clear before issuing the RTC.

Command Listing

The Mx4 serial communication commands are listed in alphabetical order. The command listing follows this format:

COMMAND FORMAT structure of command messages from host to serial adapter

DESCRIPTION explanation of command operation, functionality

FUNCTION indicates the command function

MESSAGE TYPE command code

RESPONSE FORMAT structure of response message from serial adapter to host

MT_READ1

FUNCTION High Level Read

MESSAGE TYPE 0x01

DESCRIPTION

This command is used to read one or more segments of the Mx4 dual port RAM. The serial adapter checks to see whether any segments overlap which are protected by access bytes. If they do, it locks the section before reading it.

COMMAND FORMAT

Offset	Size	Description
0	1	Message Type (1)
1	1	Size of segment #1 (m_1)
2	2	Address of segment #1
4	1	Size of segment #2 (m_2)
5	2	Address of segment #2
.	.	
.	.	
.	.	
(n-1)*3+1	1	Size of segment #n (m_n)
(n-1)*3+2	2	Address of segment #n

RESPONSE FORMAT

Offset	Size	Description
0	1	Message Type (1)
1	m_1	segment #1 data
$1+m_1$	m_2	segment #2 data
.	.	.
.	.	.
.	.	.
$1+m_1+...+m_{n-1}$	m_n	segment #n data

MT_READ2

FUNCTION Low Level Read

MESSAGE TYPE 0x02

DESCRIPTION This command is used to read one or more segments of the Mx4 dual port RAM. Unlike the previous command it does not check access bytes.

COMMAND FORMAT

Offset	Size	Description
0	1	Message Type (=2)
1	1	Size of segment #1 (m_1)
2	2	Address of segment #1
4	1	Size of segment #2 (m_2)
5	2	Address of segment #2
.	.	
.	.	
.	.	
(n-1)*3+1	1	Size of segment #n (m_n)
(n-1)*3+2	2	Address of segment #n

RESPONSE FORMAT

Offset	Size	Description
0	1	Message Type (1)
1	m_1	segment #1 data
$1+m_1$	m_2	segment #2 data
.	.	.
.	.	.
.	.	.
$1+m_1+...+m_{n-1}$	m_n	segment #n data

MT_WRITE1

FUNCTION High Level Write

MESSAGE TYPE 0x03

DESCRIPTION

This command is used to write data to one or more segments of Mx4 dual port RAM. When this write command is used, the serial adapter waits for the RTC byte to clear before transferring the data. This is useful for loading the ring-buffer when using RTCs like LOAD_CUBIC.

COMMAND FORMAT

Offset	Size	Description
0	1	Message Type (=3)
1	1	Size of segment #1 (m_1)
2	2	Address of segment #1
4	m_1	Data for segment #1
6	1	Size of segment #2 (m_2)
7	2	Address of segment #2
9	m_2	Data for segment #2
.	.	
.	.	
$(n-1)*3+m_1+...+m_{n-1}+1$	1	Size of segment #n (m_n)
$(n-1)*3+m_1+...+m_{n-1}+2$	2	Address of segment #n
$(n-1)*3+m_1+...+m_{n-1}+4$	m_n	Data for segment #n

RESPONSE FORMAT

Offset	Size	Description
0	1	Message Type (=3)

MT_WRITE2

FUNCTION Low Level Write

MESSAGE TYPE 0x04

DESCRIPTION

This command is used to write data to one or more segments of the Mx4 dual port RAM. Unlike the previous command it does not wait for the RTC byte to clear before transferring the data to the dual port RAM.

COMMAND FORMAT

Offset	Size	Description
0	1	Message Type (=3)
1	1	Size of segment #1 (m_1)
2	2	Address of segment #1
4	m_1	Data for segment #1
6	1	Size of segment #2 (m_2)
7	2	Address of segment #2
9	m_2	Data for segment #2
.	.	
.	.	
$(n-1)*3+m_1+...+m_{n-1}+1$	1	Size of segment #n (m_n)
$(n-1)*3+m_1+...+m_{n-1}+2$	2	Address of segment #n
$(n-1)*3+m_1+...+m_{n-1}+4$	m_n	Data for segment #n

RESPONSE FORMAT

Offset	Size	Description
0	1	Message Type (=3)

MT_RTC

FUNCTION Issue Real Time Command

MESSAGE TYPE 0x05

DESCRIPTION This command is used to issue an RTC. The serial adapter waits for the RTC byte to clear, transfers the arguments to the RTC buffer, and finally sets the command byte.

COMMAND FORMAT

Offset	Size	Description
0	1	Message Type (=5)
1	1	RTC Code
2	n	RTC Arguments

RESPONSE FORMAT

Offset	Size	Description
0	1	Message Type (=5)

This page intentionally blank.