IB-11A002

MOTION CONTROL SYSTEM, MSC SERIES DEC. 1996

Executive Port Serial Communications

Application Note

INDUSTRIAL INDEXING SYSTEMS, Inc.

Revision - E
Approved By:

MSC FAMILY SERIAL COMMANDS

The MSC family of Controllers can be interfaced to a host computer system using serial communications. The host can control the operation of the MSC in the following ways:

1. Read and Write Data Variables.

2. Read "Status Word".

3. Read Flags and I/O.

4. Set Flags and I/O.

5. Provide data for CAM Motions.

Serial commands are implemented using a **Packet** concept. A **Packet** consists of:

1. The **Packet Header** which is a 6 byte area defined by a **start of packet** character (1 byte), the **MSC unit number** (1 byte), the **packet body length** (2 bytes) and a **block sequence number** (2 bytes).

2. The **Packet Body** which is a variable length area containing ASCII characters and binary data representing addresses, values and other parameters.

3. The **Packet Checksum** which is a 2 byte area representing the checksum (unsigned) of the **Packet** beginning with the **MSC unit number** byte to and including the end of the **Packet Body**.

All communications are transferred at 9600 BAUD with 8 data bits and 1 stop bit. Parity is not used.

The host must initiate all serial communications as the MSC is a passive device with respect to Packet Protocol.

The **Packet Header**, **Packet Body** and **Packet Checksum** are generated using the protocol described in the remainder of this document.

DATA TRANSFER PROTOCOL

This protocol allows data to be sent between the MSC and a host computer.

In data communications, the protocol defines the rules for the electrical, physical and functional characteristics of the communication link. The protocol contains procedures required to ensure an orderly exchange of information through the link, to and from the executing programs.

The MSC is initially a “passive” device with respect to **Packet** transmission. It will wait for **Packet** requests from other devices and then respond to those requests where appropriate. It is the responsibility of the host to initiate communications with the MSC.

When the MSC receives a **Packet**, it responds to the host with either an ACK (acknowledgment) character or a NAK (negative acknowledgment) character. An ACK character is an indication that the **Packet** received was valid. A NAK character is an indication that the **Packet** received was invalid or a timeout occurred.

It is the responsibility of the host device to retransmit a **Packet** in the event that it receives a NAK character from the MSC.

DATA TRANSFER SEQUENCE

The following describes a typical sequence of events for the transfer of **Packets** between devices:

1. The host device creates and sends a **Packet** to the MSC.

2. The MSC receives the **Packet**. The **Packet** is examined for a valid **Packet Header**, **Packet Body** and **Packet Checksum**.

3. The MSC will respond with an ACK (acknowledgment) character if the **Packet Header**, **Packet Body** and **Packet Checksum** are correct.

   The MSC will respond with a NAK (negative acknowledgment) character if a problem was found with the **Packet** (for example, an incorrect **Packet Checksum** was received or transmission fails to complete within a given period of time resulting in a timeout).

4. If the **Packet** from the host is requesting information, the MSC will then retrieve the information, build a return **Packet** and transmit this **Packet** back to the host device.

   The MSC will also expect an ACK character (06H) or NAK (15H) character from the host device.

5. The MSC will retransmit the **Packet** up to 5 times if necessary.

PACKET DATA STRUCTURE

Please refer to Figure 1 for a diagram of the packet protocol.

1. The first byte is the **start of packet** character. It will always be a STX (02H) character and is used to signal the beginning of a P**acket**.

2. The second byte defines the **MSC unit number** (01H through 0fH). This byte should be set to the decimal value of 1 when using the MSC in RS-232 mode. For RS-485 communications, this byte should contain the decimal value 1 through 15 depending on the RS-485 address switch setting of the selected MSC that is being communicated with.

3. The next 2 bytes define the **packet body length** (the **start of packet** character, the **MSC unit number**, the **packet body length**, the **block sequence number** and the **Packet Checksum** are not included). The length is given in bytes and may be an even or odd value. The most significant byte of the length is transmitted first.

4. The next two bytes define the **block sequence number**. This value is not presently used.

5. The variable length **Packet Body** is next. The maximum length of the **Packet Body** is 240 bytes.

6. Following the **Packet Body**, there is a two byte field which contains the checksum for the **Packet**. The checksum is calculated by summing the contents of each **byte** of the **Packet**, beginning with the **MSC unit number** and continuing through the last byte of the **Packet Body**. Unsigned addition is performed. The high order byte of the checksum is transmitted first.

MSC Packet Protocol Format

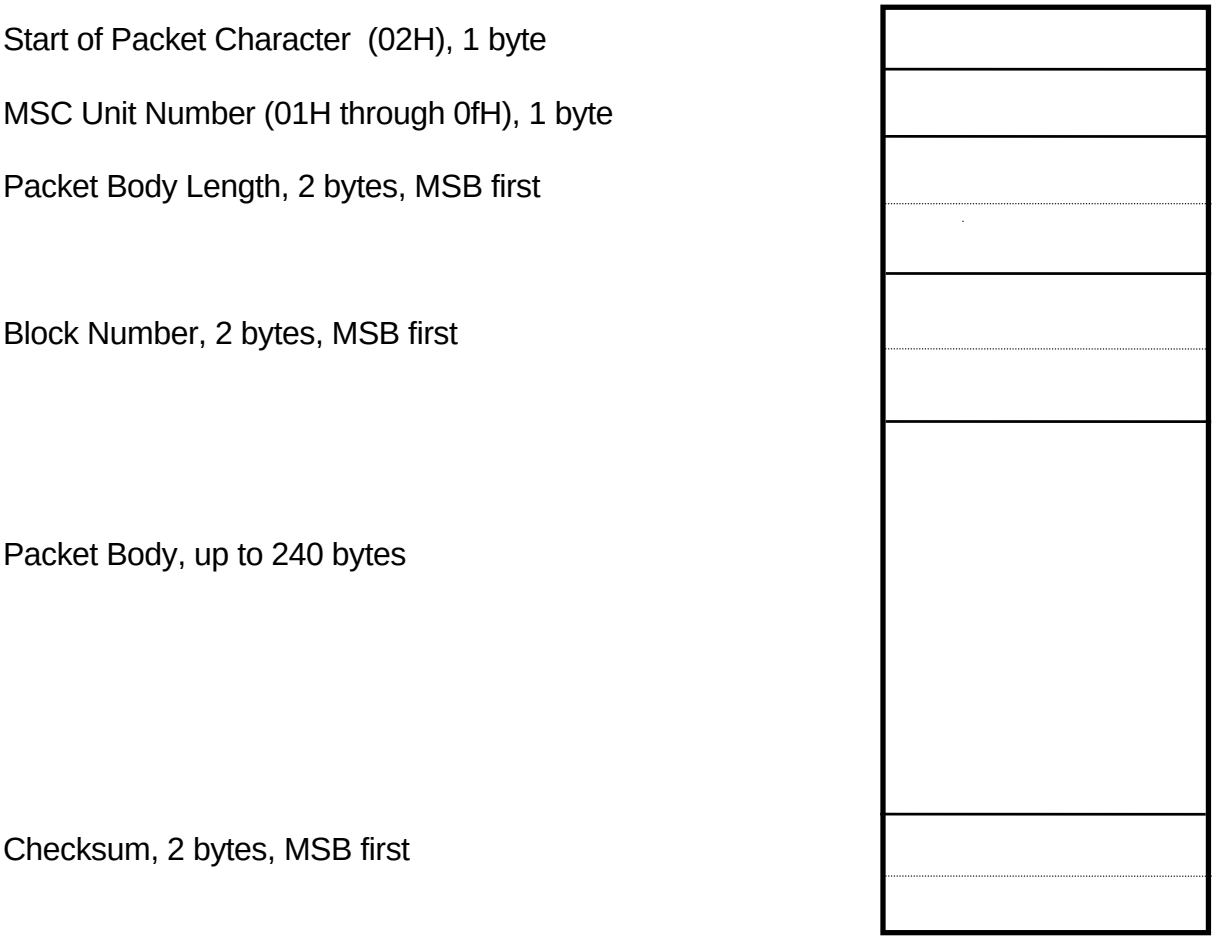

Figure 1 - Packet Protocol Format

EXAMPLE PACKET

The following is an example of a **Packet** that would be created to write the value 1234 (decimal) to address 2008 (decimal) in an MSC:

BYTE	ASCII VALUE	HEX VALUE	DECIMAL VALUE	DESCRIPTION
0		02	2	Start of Packet
1		01	1	MSC Unit Number (1 - 15)
2		00	0	Packet Body Length
3		0D	13	"
4		00	0	Block Sequence Number
5		00	0	"
6	D	44	68	MSC Serial Command
7	W	57	87	"
8	R	52	82	"
9	I	49	73	"
10	T	54	84	"
11	E	45	69	"
12	,	2C	44	"
13		07	7	Address of Data (2008 dec.)
14		D8	216	"
15		00	0	Data Value (1234 dec.)
16		00	0	"
17		04	4	"
18		D2	210	"
19		03	3	Checksum of bytes 1 - 18 (958 dec.)
20		BE	190	"

Please refer to Figure 1 - Packet Protocol Format.

GENERAL SERIAL COMMAND FORMAT

Serial commands consist of two basic types. Some commands serve only to transmit information to the MSC and receive no response, other than the acknowledge character. Other commands request information from the MSC after it has responded with its acknowledgment to the command.

Each serial command consists of ASCII characters, followed by a comma, and optionally followed by one or more data values. The complete command must be formatted into a **Packet** by placing the appropriate header information in the packet, followed by the command, which is in turn followed by the appropriate checksum.

If the command is one that requires a response from the MSC, then the MSC will return the response in the same packet format. Further, the MSC will place the command name and the comma in the packet. Details for each serial command follow. In the format description for each packet, [Header] represents the packet header, and [Checksum] represents the packet checksum. The brackets themselves are for purposes of clarity and are NOT part of the actual packet. All numbers in binary format are most significant byte first, followed by least significant byte(s).

The BREAD Command

The BREAD (block read) serial command serves to read a group of 32 data values from the MSC. Each data value is stored sequentially in the Macroprogram data area beginning at the address specified in the BREAD command.

Packet sent to the MSC:

[Header]BREAD,[AA][Checksum]

where [AA] is the address within the Macroprogram data area where the first of the 32 values is to be read. The value [AA] is stored as a 16 bit, 2 byte number.

Packet returned by the MSC:

[Header]BREAD,[V1][V2]...[V32][Checksum]

where [V1] through [V32] are 32 bit, 4 byte signed numbers.

The BWRITE Command

The BWRITE (block write) serial command serves to transmit a group of 32 data values to the MSC. Each data value is stored sequentially in the Macroprogram data area, beginning at the address specified in the BWRITE command. Caution should be used to ensure that sufficient space has been reserved in the data area to receive the 32 values. If fewer than 32 values are to be transmitted, the user must "pad" the command with a sufficient number of dummy values to make a total of 32.

Data arrays longer than 32 values can be transmitted by using multiple BWRITE commands.

Packet sent to the MSC:

[Header]BWRITE,[AA][V1][V2]...[V32][Checksum]

where [AA] is the address within the Macroprogram data area where the first of the 32 values is to be stored, and [V1] through [V32] represent the 32 data values to be transmitted. The value [AA] is stored as a 16 bit, 2 byte number. Each data value is stored as a 32 bit, 4 byte number. When placing numeric values in the packet, the most significant byte goes first.

The value for [AA] can be determined from the Data Table portion of the Macroprogram listing.

For example, if a Macroprogram contains the statement

```
array    dim    32
```

then the Data Table will contain an entry for the symbol called "array" and will indicate its address within the data area. This is the value that should be used for [AA].

Packet returned by the MSC:

None.

The CFLAG Command

The CFLAG serial command can be used to turn off a specified MSC output flag or MSC user flag.

Packet sent to the MSC:

[Header]CFLAG,[FF][Checksum]

where [FF] is a 16 bit, 2 byte number representing the desired flag.

Packet returned by the MSC:

None.

The DREAD Command

The DREAD function is used to read the contents of a specified address in the Macroprogram data storage area. The requested value is returned as a 4 byte, 32 bit signed value.

Packet sent to the MSC:

[Header]DREAD,[AA][Checksum]

where [AA] is a 16 bit number that represents the address in the Macroprogram data area where the desired data is stored.

The value for [AA] can be determined from the Data Table portion of the Macroprogram listing.

Packet returned by the MSC:

[Header]DREAD,[VVVV][Checksum]

where [VVVV] is a 32 bit, 4 byte signed number.

The DWRITE Command

The DWRITE serial command transfers a single data value to the specified address in the Macroprogram data area.

Packet sent to the MSC:

[Header]DWRITE,[AA][VVVV][Checksum]

where [AA] is a 16 bit number that represents the address in the Macroprogram listing.

The data value [VVVV] is a 4 byte, 32 bit signed number.

Packet returned by the MSC:

None.

The RFLAG Command

The RFLAG serial command can be used to read the status of MSC inputs, outputs, timers, motor status flags, and user flags.

Packet sent to the MSC:

[Header]RFLAG,[FF][Checksum]

where [FF] is a 16 bit, 2 byte number representing the flag of interest.

Packet returned by the MSC:

[Header]RFLAG,[NN][Checksum]

where [NN] is a 16 bit, 2 byte number representing the state of the flag of interest. If the flag is ON (set), [NN] will be 1. If the flag is OFF (cleared), [NN] will be zero.

The RFLAGS Command

The RFLAGS serial command can be used to read the status of all MSC inputs, outputs, timers, motor status flags, and user flags with a single instruction.

Packet sent to the MSC:

[Header]RFLAGS,[Checksum]

Packet returned by the MSC:

[Header]RFLAGS,[B1][B2]...[B32][Checksum]

where [B1] through [B32] are 8 bit unsigned values representing the state of the 256 MSC internal flags. The flag states are returned as follows:

BYTE	BIT #	FLAG #
B1	7 - 0	7 - 0
B2	7 - 0	15 - 8
B3	7 - 0	23 - 16
B4	7 - 0	31 - 24
	.	
	.	
	.	
	.	
B32	7 - 0	255 - 248

The RFLAGS serial command is **only** supported by the MSC-850/32 and MSC-250 controllers.

The RQSTAT Command

The RQSTAT serial command requests the MSC controller to transmit its status word.

Packet sent to the MSC:

[Header]RQSTAT,[Checksum]

Packet returned by the MSC:

[Header]RQSTAT,[SS][FF][AAAA][Checksum]

where [SS] is the 16 bit Macroprogram status word, and [FF] and [AAAA] are not currently used. The Macroprogram status word definition is shown below.

As an example, if the returned Macroprogram Status Word is 65, the program is running (1) and auto start is enabled (64).

MACROPROGRAM STATUS WORD

BIT	VALUE	DEFINITION
0	1	PROGRAM RUNNING
1	2	BATTERY FOR MEMORY LOW
2	4	RAM MEMORY NOT RELIABLE
3	8	SYSTEM HAS BEEN RESET, PROGRAM WAS CLEARED
4	16	SYSTEM RETURN INTERRUPT ENCOUNTERED
5	32	SYSTEM FAULT INTERRUPT ENCOUNTERED
6	64	AUTO START ENABLED, PROGRAM STARTS ON POWER-UP
7	128	SYSTEM TEST IN PROGRESS
8	256	ERROR - LOADING OVER RUNNING PROGRAM
9	512	ERROR - CHECKSUM IN INCORRECT
10	1024	PROGRAM LOAD IN PROGRESS
11	2048	ERROR - PROGRAM LOAD OUT OF SEQUENCE
12	4096	ERROR - BAD OPCODE ENCOUNTERED
13	8192	ERROR - STACK OVERFLOW ENCOUNTERED
14	16384	ERROR - STACK UNDERFLOW ENCOUNTERED
15	32768	ERROR - SOFTWARE INTERRUPT TABLE FULL

The SFLAG Command

The SFLAG serial command can be used to turn on a specified MSC output flag or MSC user flag.

Packet sent to the MSC:

[Header]SFLAG,[FF][Checksum]

where [FF] is a 16 bit, 2 byte number representing the desired MSC flag.

Packet returned by the MSC:

None.