Preliminary

Animatics SmartMotor™ *with* DeviceNet Specifications:

Mode Velocity Example

Revision 0 4/09/03 Jason Hyatt

APPENDIX: MODE VELOCITY EXAMPLE

This example uses I/O Polling Messaging only.

Following will be the sequence of values contained in the 8 bytes that are the DeviceNet output (command) registers. Also displayed are the values contained in the 8 bytes that are the DeviceNet input (response) registers.

This example is the equivalent of a SmartMotor™ host command sequence, or downloaded user program segment, consisting of the SmartMotor™ commands:

MV	‘ set mode velocity
A=255	‘ set acceleration to 255 which is hex 00 00 00 FF
V=100000	‘ set velocity to 100,000 which is hex 00 01 86 A0
G	‘ go, or begin motion trajectory

1. Set up packet to change from power-up default mode position to velocity mode.

Operating Mode is in the Position Controller Class, Attribute number 3.
Use the Command Message Type 1B since it is Position Controller and not Position Controller Supervisor (which would be Type 1A).

Command Bytes (0 through 7)

00	bit 0, load data will not be set until all other bytes have the desired value
03	GET attribute 3, Operating Mode, for the response message data
1B	Command Type Position Controller Attribute
03	SET attribute number 3, Operating Mode
01	value for Operating Mode velocity is 1
00	higher order bytes of data are 0
00	
00	

2. Now that all these bytes are set up, set the first byte, load data bit to cause the Operating Mode Attribute to be SET to the data value of 1, velocity mode.

Command Bytes 0-7

01	bit 0, load data is NOW set to cause data to be accepted
03	GET attribute 3, Operating Mode, for the response message data
1B	Command Type Position Controller Attribute
03	SET attribute number 3, Operating Mode
01	value for Operating Mode velocity is 1
00	higher order bytes of data are 0
00	
00	

3. If we look at the Response Message, we expect to see:

Response Bytes 0-7

01	echo of command byte 0 becoming a 1 proves command received
03	GET attribute 3
80	Load of data complete
1B	Response Type Position Controller Attribute
01	new value of Position Controller Attribute 3, Operating Mode
00	
00	
00	

4. Now that the response byte 0 displays the load data bit 0 set to 1 (value 01), we know that our data has been received and loaded. We now set the load data bit to 0 so we can begin again.

Command Bytes 0-7

00	bit 0, load data is set to zero to begin handshaking again
03	GET attribute 3, Operating Mode, for the response message data
1B	Command Type Position Controller Attribute
03	SET attribute number 3, Operating Mode
01	value for Operating Mode velocity is 1
00	higher order bytes of data are 0
00	
00	

5. Now that velocity mode is set, we set the acceleration, leaving the load data bit zero until we are ready. We will use the I/O Command message Type 03, set Acceleration, to set the acceleration value.

Command Bytes 0-7

00	bit 0, load data is set to zero to begin handshaking again
00	unused
03	Command Type 03 Acceleration
03	ask for Response message Type 03, report Actual Velocity
FF	value for Acceleration is 255 which is hex FF
00	higher order bytes of data are 0
00	
00	

6. We are ready to set acceleration, so we set the Load Data bit.

Command Bytes 0-7

01	bit 0, load data is set to one to load the data
00	unused
03	Command Type 03 Acceleration
03	ask for Response message Type 03, report Actual Velocity
FF	value for Acceleration is 255 which is hex FF
00	higher order bytes of data are 0
00	
00	

7. We check the response message to be sure the data has been loaded.

Response Bytes 0-7

01	echo of command byte 0 becoming a 1 proves command received
00	not used
80	Load of data complete
03	Response Type 03 Report Actual Velocity
00	velocity is zero, motion not begun
00	
00	
00	

8. Now that the response byte 0 displays the load data bit 0 set to 1 (value 01), we know that our data has been received and loaded. We now set the load data bit to 0 so we can begin again.

Command Bytes 0-7

00	bit 0, load data is set to one to load the data
00	unused
03	Command Type 03 Acceleration
03	ask for Response message Type 03, report Actual Velocity
FF	value for Acceleration is 255 which is hex FF
00	higher order bytes of data are 0
00	
00	

9. Now that the acceleration has been set, we are ready to set the target velocity and simultaneously begin our move. First we will place our values in bytes 1 to 7, leaving byte 0, with the load data bit, unchanged until we are ready. We will use the I/O Command message Type 02, set Target Velocity, to set the velocity value.

Command Bytes 0-7

00	bit 0, load data is set to zero to begin handshaking again
00	unused
02	Command Type 02 set Target Velocity
03	ask for Response message Type 03, report Actual Velocity
A0	value for Velocity is 100000 which is hex 00 01 86 A0
86	
01	
00	

10. Simultaneously load the target velocity by setting load data bit 0 of byte 0 to cause the data load, and begin motion by setting enable bit 7 of byte 0 to cause power to the motor to be enabled, by placing hex 81 in byte 0.

Command Bytes 0-7

81	enable bit 7 plus load data bit 0 to begin the move
00	unused
02	Command Type 02 set Target Velocity
03	ask for Response message Type 03, report Actual Velocity
A0	value for Velocity is 100000 which is hex 00 01 86 A0
86	
01	
00	

11. We check the response message to be sure it worked.

Response Bytes 0-7

81	echo of command byte 0 becoming a 1 proves command received
00	not used
80	Load of data complete
03	Response Type 03 Report Actual Velocity
A0	velocity is 100,000 = hex 00 01 86 A0 (once acceleration is complete)
86	
01	
00	