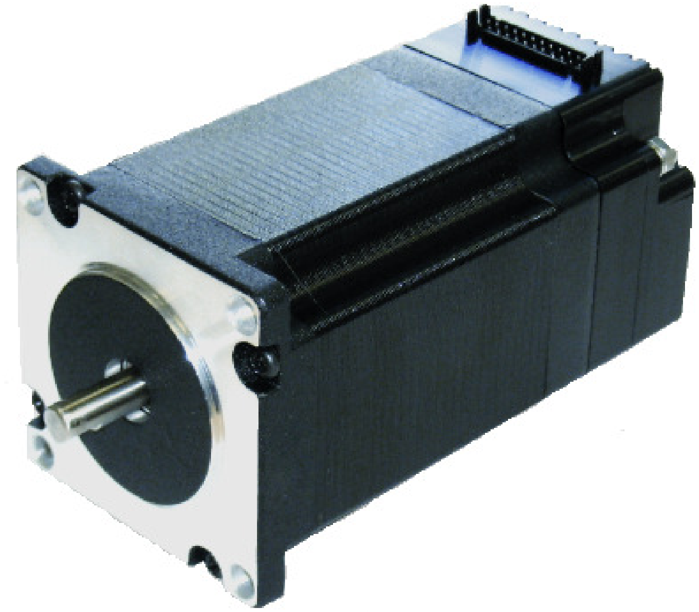
DMX-CAN-23
Integrated Step Motor
Encoder/Driver/Controller
Manual



First edition, December 2010

ARCUS makes no representations or warranties regarding the content of this document. We reserve the right to revise this document any time without notice and obligation.

Revision History:

1.4 – 1st release
1.5 – 2nd release
1.6 – 3rd release

Firmware Compatibility:

V117

Software Compatibility:

V108

Table of Contents

1. Introduction

DMX-CAN is an all-in-one integrated motor package that combines all the motion components in to one convenient package.

Communication to the DMX-CAN can be established over RS-232, RS-485 or CANbus.

Features

DMX-CAN-23

- RS-485 + RS-232 ASCII communication
  - 9600, 19200, 38400, 57600, 115200 bps
- CANbus communication over CANOpen implementing CiA DSP-402
  - 10K, 20K, 50K, 100K, 125K, 250K, 500K, 800K, 1M bps
- A/B/Z differential encoder inputs
  - StepNLoop closed loop control (position verification)
- Opto-isolated I/O
  - 6 x inputs (1 x high speed position capture latch input)
  - 3 x outputs (1 x position synchronized output)
  - +Limit/-Limit/Home inputs
- Homing routines:
  - Home input only
  - Limit only
  - Z-index encoder channel only
  - Home input + Z index encoder channel
- S-curve or trapezoidal acceleration profile control
- On-the-fly speed change
- 1000 line incremental encoder (4000 counts/rev with 4x quadrature decoding)
- Stepper driver
  - 12-35 VDC
  - 2.5 Amp max current setting (peak current)
  - 16 micro-step setting (fixed)
  - 400 KHz max pulse support
- Stepper motor
  - NEMA 23 motor sizes available in different stack sizes
  - 1.8° step angle

Model Numbers

Main Product

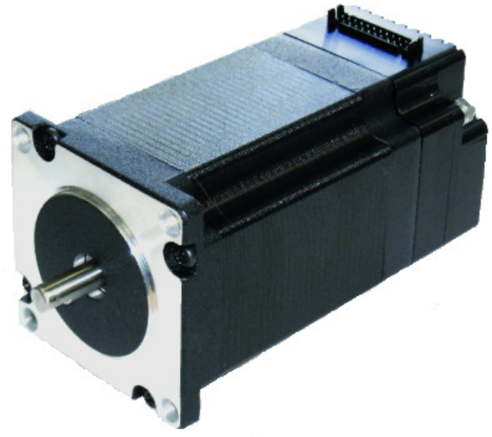
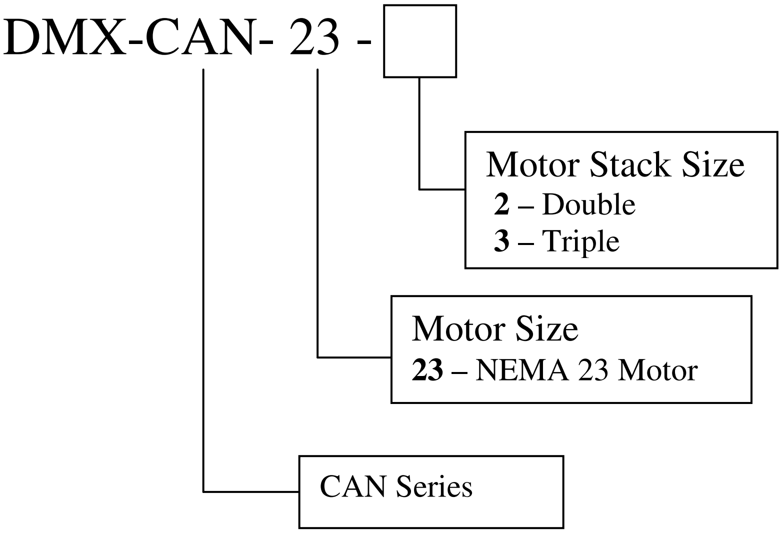


Figure 1.0



DMX-CAN- 23 - □

Motor Stack Size
2 – Double
3 – Triple

Motor Size
23 – NEMA 23 Motor

CAN Series

Contacting Support

For technical support contact: support@arcus-technology.com.

Or, contact your local distributor for technical support.

2. Electrical and Thermal Specifications

Power Requirement

Regulated Voltage:	**+12 to +35 VDC**
Current (Max):	**2.5 A (peak)**

Temperature Ratings †

Operating Temperature:	**0°C to +70°C**
Storage Temperature:	**-55°C to +150°C**

† Based on component ratings

Digital Inputs †

Type:	**Opto-isolated NPN inputs**
Opto-isolator supply:	**+12 to +24 VDC**
Maximum forward diode current:	**45 mA**

† Includes limit, home and latch

Digital Outputs

Type:	**Opto-isolated open-collector NPN outputs**
Max voltage at collector:	**+24 VDC**
Max source current at 24VDC	**†90 mA**

† A current limiting resistor is required

3. Dimensions

†All dimensions in inches

DMX-CAN-23

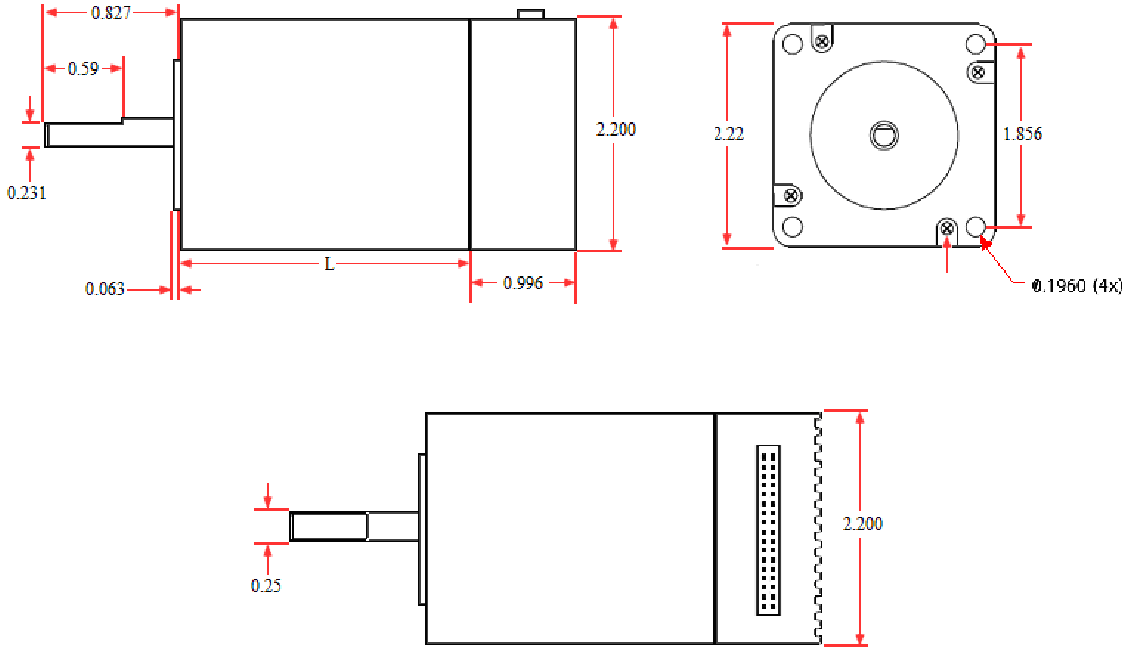


Figure 3.0

All dimensions in inches

Model	L (inches)
DMX-K-SA-23-2	2.2
DMX-K-SA-23-3	3.1

Table 3.0

4. Motor Specifications

Electrical Specifications

NEMA Size	Stack Size	Current / Phase †	Holding Torque	Resistance/ Phase	Inductance/ Phase	Inertia
23	Double	2.8A	0.95 N-m	0.9 Ω	2.5 mH	1.64 oz-in^2
	Triple	2.8A	1.41 N-m	1.13 Ω	3.6 mH	2.62 oz-in^2

Table 4.0

† Motor current specifications are in RMS form.

Torque Curve – NEMA 23

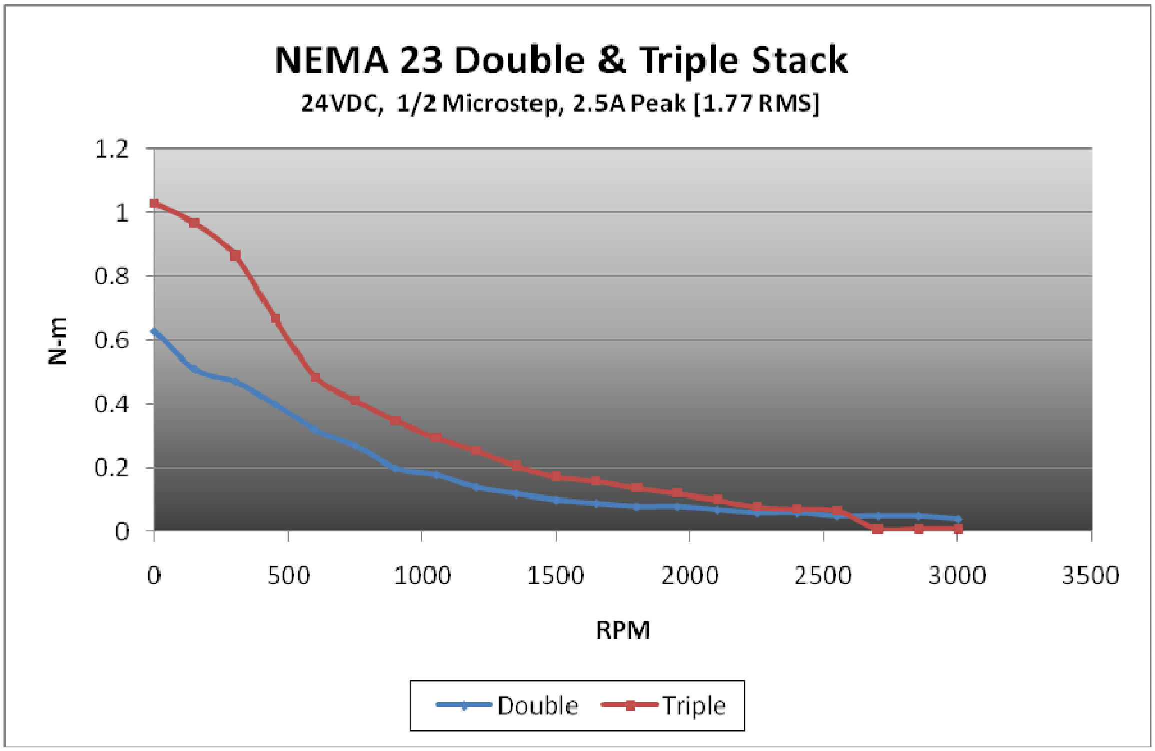


Figure 4.0

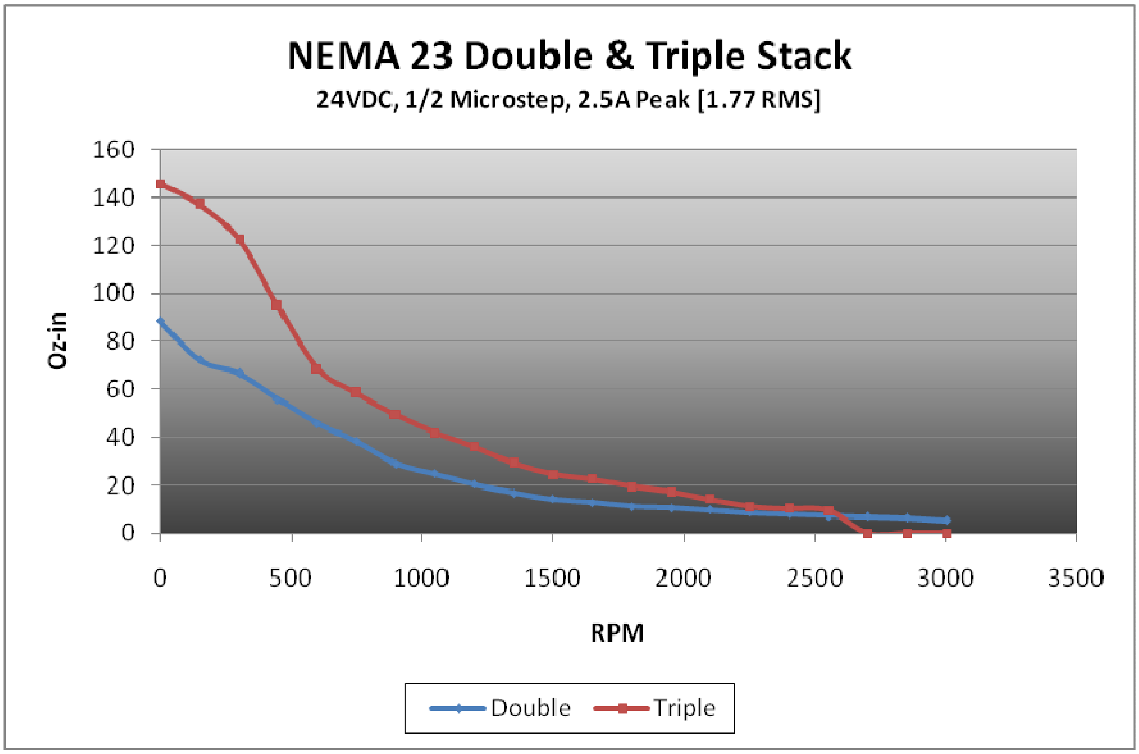

NEMA 23 Double & Triple Stack
24VDC, 1/2 Microstep, 2.5A Peak [1.77 RMS]
Oz-in
160
140
120
100
80
60
40
20
0
0
500
1000
1500
2000
2500
3000
3500
RPM
Double
Triple

Figure 4.1

5. Connections

24-Pin Connector (2mm)

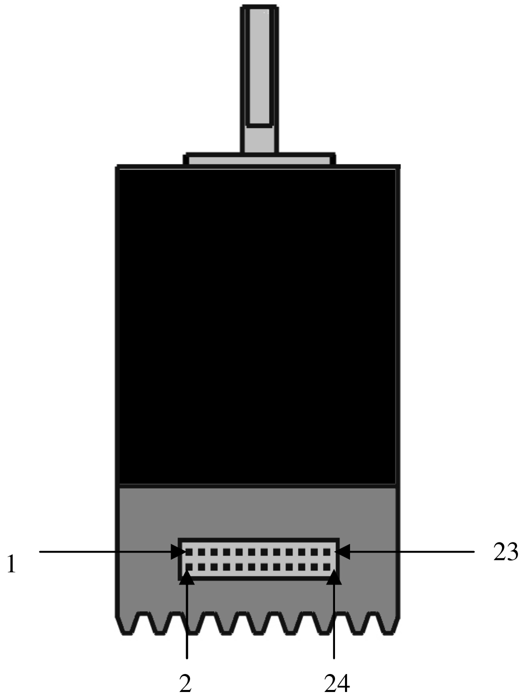


Figure 5.0

Pin #	In/Out	Name	Description
1	I	PWR	+12 to +35VDC power input
2	I	PWR	+12 to +35VDC power input. Shorted to pin 1
3	I	GND	Ground
4	I/O	485+	RS-485 plus signal
5	I	HOME	Home input
6	I/O	485-	RS-485 minus signal
7	O	DO1/INP	Digital Output 1
8	I	+LIM	Plus limit input
9	I/O	RXD	RS-232 RXD signal
10	I	-LIM	Minus limit input
11	I	DI1	Digital Input 1
12	I	DI3	Digital Input 3
13	I	DI2/LT	Digital Input 2 / Latch
14	I	DI4	Digital Input 4
15	I/O	CAN-H	CAN-H
16	I	DI5	Digital Input 5
17	I	OPTO	+12-24VDC opto-supply

18	I	DI6	Digital Input 6
19	I/O	TXD	RS-232 TXD signal
20	O	DO2	Digital Output 2
21	I	OPTOGND	Opto-supply ground
22	O	DO3/ALM	Digital Output 3 / Alarm
23	I/O	CAN-L	CAN-L
24	I	GND	Ground. Shorted to pin 3

Table 5.0

Mating Connector Description:	24 pin 2mm dual row connector
Mating Connector Manufacturer:	HIROSE
Mating Connector Housing Part Number:	DF11-24DS-2C
Mating Connector Pin Part Number:	DF11-2428SC

DMX-CAN-23 Interface Circuit

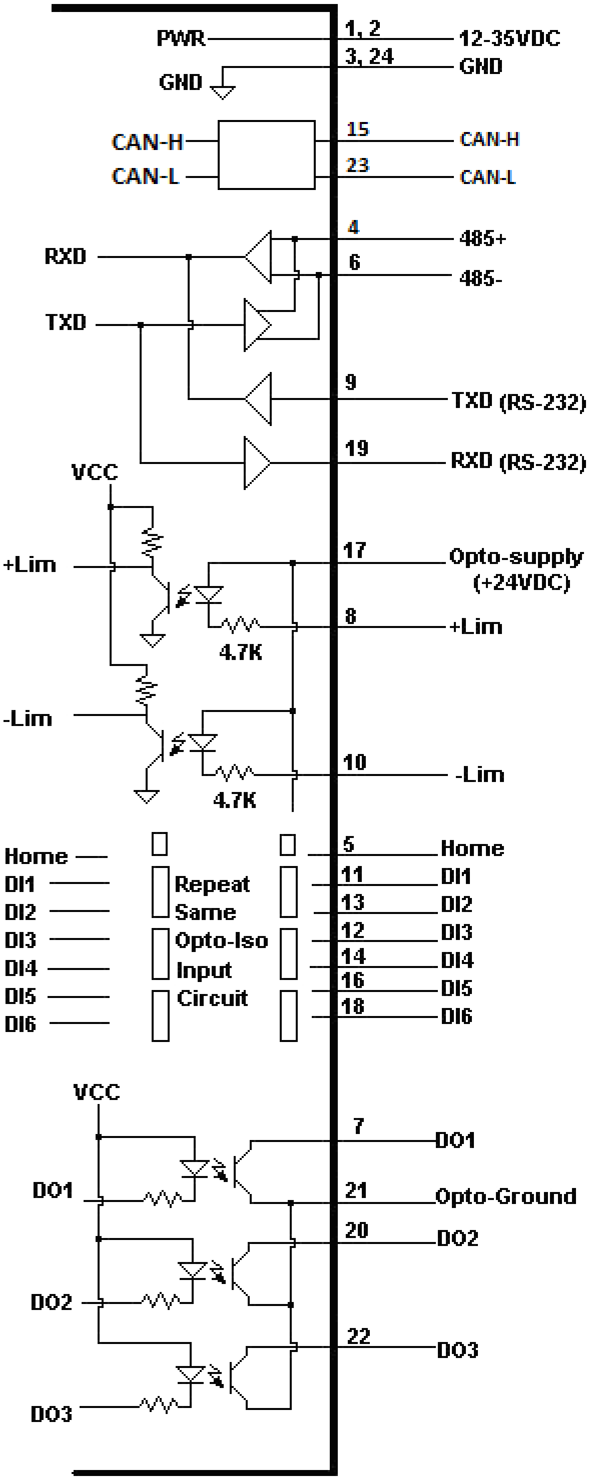


Figure 5.1

Digital Outputs

Figure 5.2 shows an example wiring to the digital output.

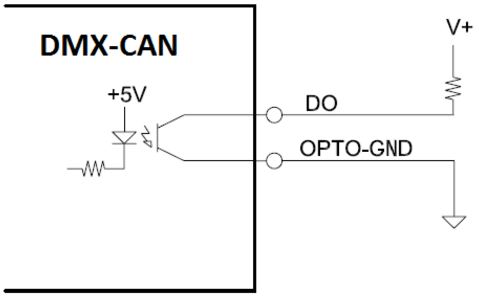


Figure 5.2

WARNING: The maximum sink current for digital outputs is 90 mA. Take caution to select the appropriate external supply and pull-up resistance to limit the sink current below this level.

Digital Inputs

Figure 5.3 shows the detailed schematic of the opto-isolated inputs.

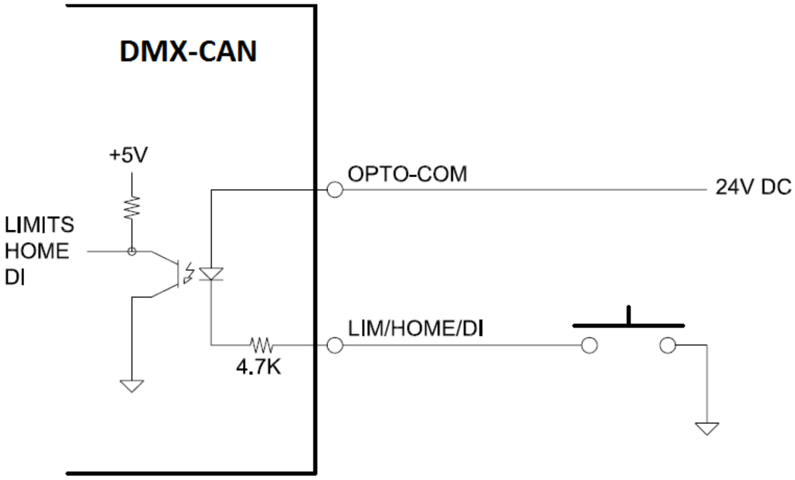


Figure 5.3

6. Getting Started

There are three ways to communicate with DMX-CAN series product: RS-232, RS-485, and CANopen.

RS-232 Communication

When the DMX-CAN unit is shipped from the factory, default communication setting is RS-232 at 9600 baud rate.

Note that RS-232 is a point-to-point protocol. See figure below:

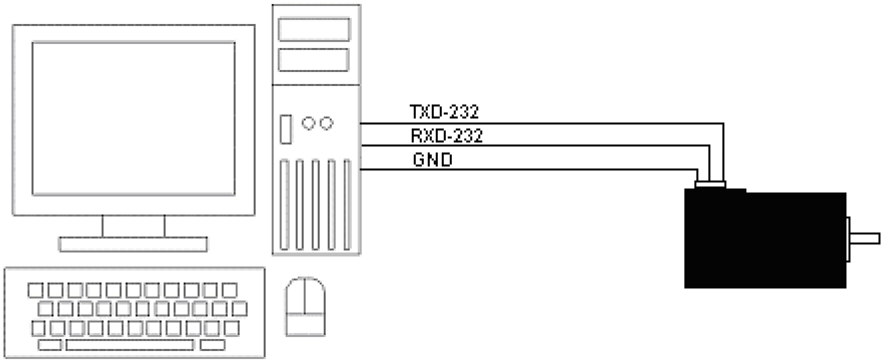


Figure 6.0

RS-485 Communication

If RS-485 communication is required, first you need to communicate using RS-232 and use the Windows program to change the communication method to RS-485, download the setup, and store to flash. Once communication method is changed, you need to reboot the module for the new parameter to take effect and then communicate through RS-485.

When communicating on RS-485, it is recommended to add 120 Ohm terminating resistor between 485+ and 485- signal on the last module.

Below is a typical RS-485 master and multi-slave multi-drop network.

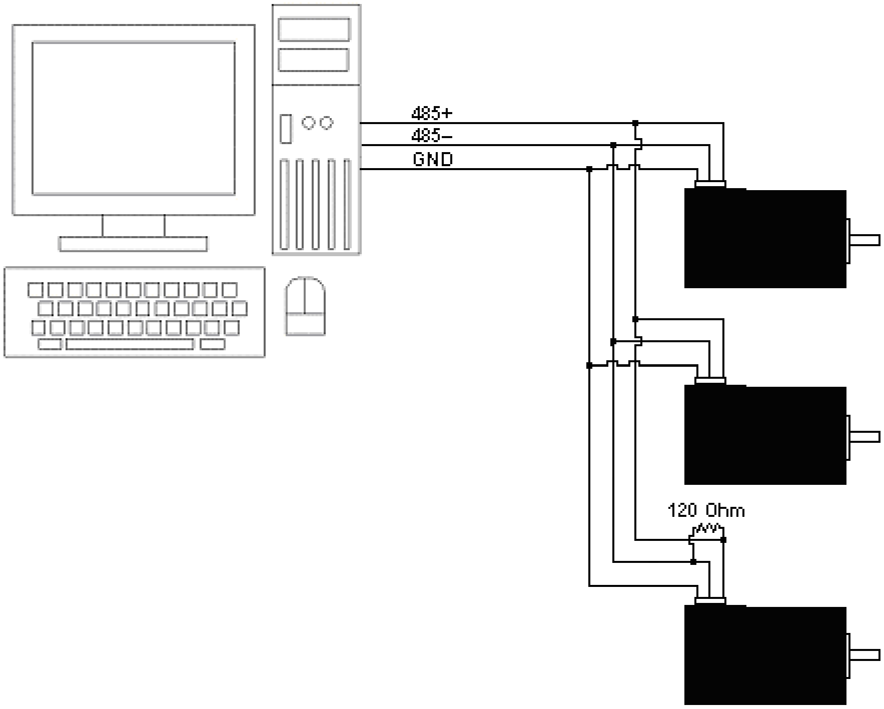


Figure 6.1

CANOpen Communication

If CANOpen communication is required, first you need to communicate using RS-232 and use the Windows program to change the communication method to CANOpen, download the setup, and store to flash. Once communication method is changed, you need to reboot the module for the new parameter to take effect and then communicate through CANopen.

When communicating on CANOpen, it is recommended to add 120 Ohm terminating resistor between CAN-H and CAN-L signal on the last module.

Below is a typical CANopen master and multi-slave multi-drop network.

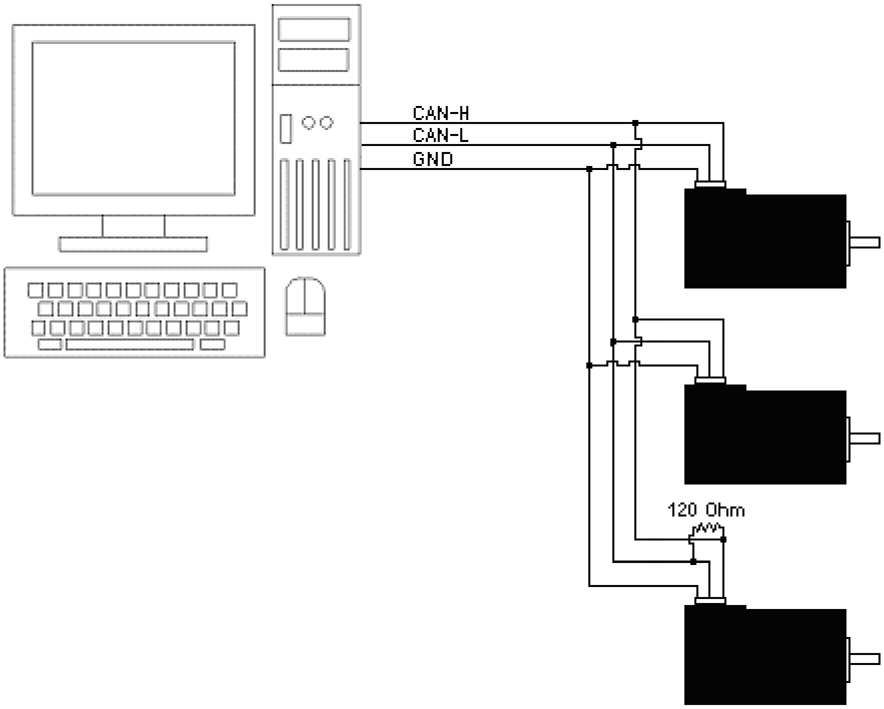
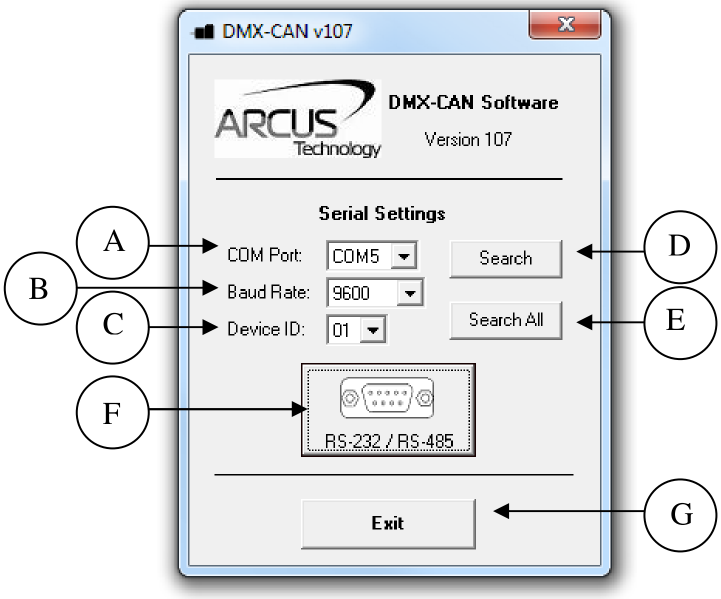


Figure 6.2

Windows GUI

DMX-CAN comes with user friendly Windows Program to quickly communicate, test, program, and debug the DMX-CAN unit.

Start the DMX-CAN program and following dialog box will show.



COM Port: COM5

Baud Rate: 9600

Device ID: 01

Figure 6.3

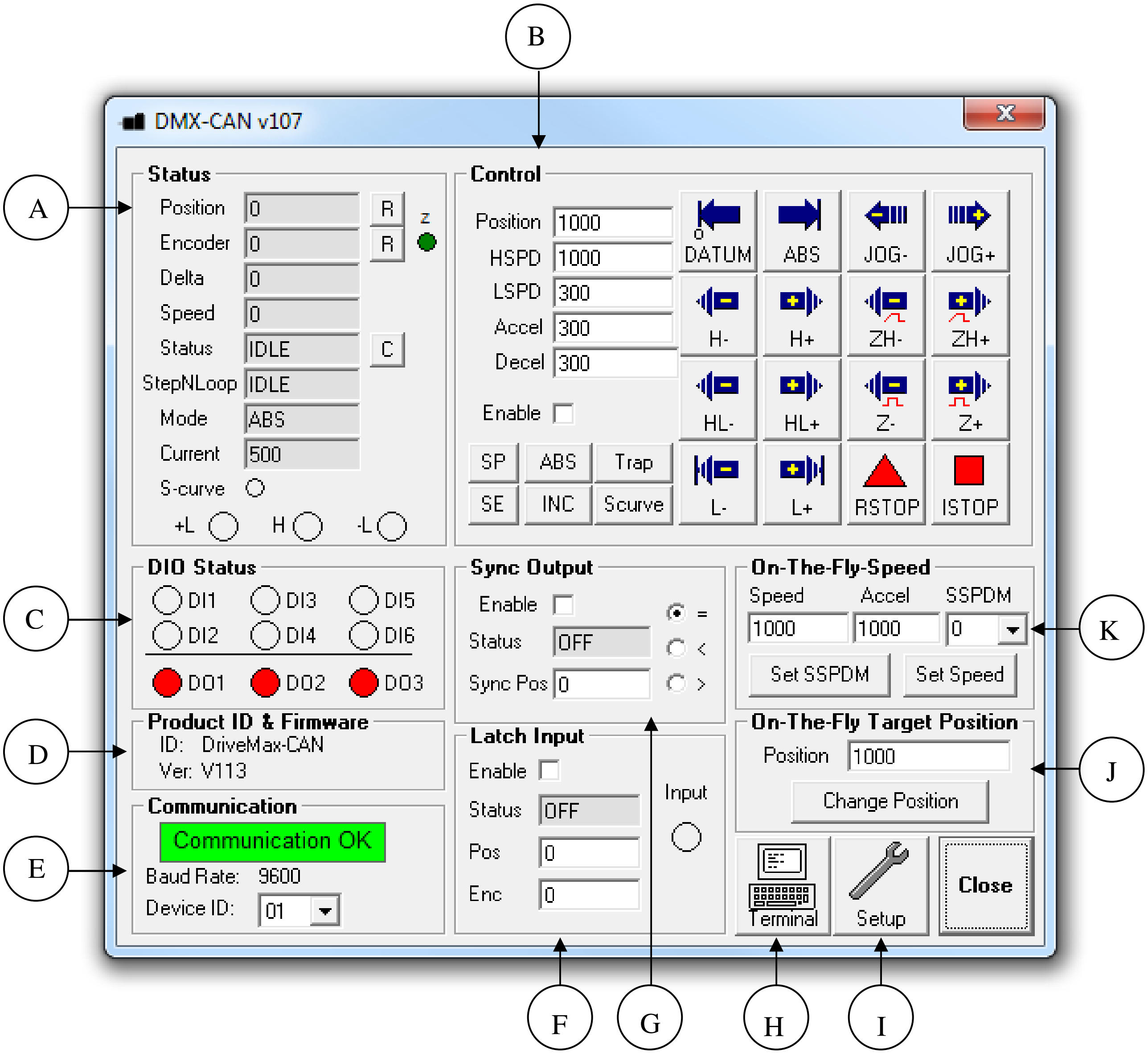
A. Serial Communication Port Number. This is a drop down combo box which has selection of serial port from COM1 to COM15.
B. 9600 baud rate is the default communication baud rate that is used. If you have set your DMX-CAN module to operate at a different baud rate, select the correct baud rate here.
C. Device name is used for RS-485 communication and to distinguish many devices on the RS-485 network.
D. Search button is used to search for any DMX-CAN module connected to an available serial communication port. Search looks for DMX-CAN from COM1 to COM15 using the device name.
E. Search All button is used to search for all DMX-CAN modules connected to the available serial communication port. This button should be used when more than one DMX-CAN are connected on a multi-drop RS-485 bus. Note that the DMX-CAN must be configured to use RS-485 communication.
F. Open connection uses the selected COM port and the device name to open communication with the DMX-CAN module.
G. Close the application program.

When connecting for the first time, we recommend that search is done to find out the COM port number where the DMX-CAN is connected.

If search cannot find the DMX-CAN device or open connection does not find the DMX-CAN, check the following:

1) Check power supply to DMX-CAN. Recommended power is from 12VDC to 35VDC.
2) Check communication wiring. If using RS-232, TXD from DMX-CAN should be connected to RXD of the serial port and RXD from DMX-CAN should be connected to TXD of serial port. If using RS-485, make sure that the 485+ from DMX-CAN is connected to 485+ of the master and 485- from DMX-CAN is connected to 485- of the master.
3) Confirm that the device name is set correctly. Default factory device name setting is “01”. If this name has been changed and stored to flash, enter the new name.

When Open Connection button is pressed and communication is successfully established, following screen will show:



Status

Position 0
Encoder 0
Delta 0
Speed 0
Status IDLE
StepNLoop IDLE
Mode ABS
Current 500
S-curve ○
+L ○ H ○ -L ○

Control

Position 1000
HSPD 1000
LSPD 300
Accel 300
Decel 300
Enable □

DIO Status

○ DI1 ○ DI3 ○ DI5
○ DI2 ○ DI4 ○ DI6
● DO1 ● DO2 ● DO3

Sync Output

Enable □
Status OFF
Sync Pos 0
● = ○ < ○ >

On-The-Fly-Speed

Speed 1000
Accel 1000
SSPDM 0

Product ID & Firmware

ID: DriveMax-CAN
Ver: V113

Latch Input

Enable □
Status OFF
Pos 0
Enc 0
Input ○

On-The-Fly Target Position

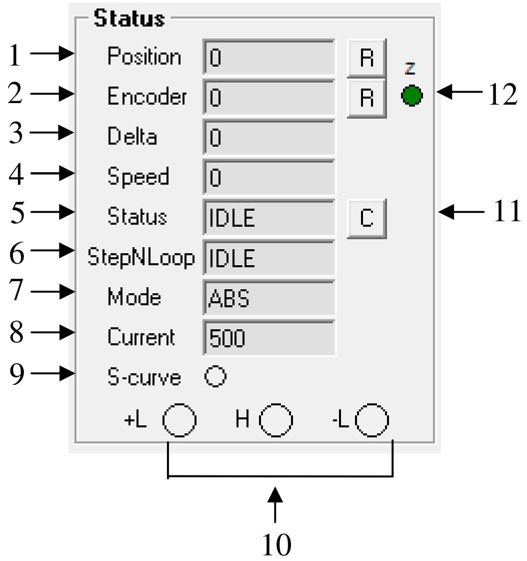
Position 1000

Communication

Communication OK
Baud Rate: 9600
Device ID: 01

Figure 6.4

A. Status



Status

Position 0

Encoder 0

Delta 0

Speed 0

Status IDLE

StepNLoop IDLE

Mode ABS

Current 500

S-curve ○

+L ○ H ○ -L ○

Figure 6.5

1. Current pulse position (when StepNLoop is enabled, pulse position is defined as the current target position)
2. Current encoder position
3. Delta position – this is the difference between the target position and actual position.
4. Speed – actual speed. When StepNLoop is enabled, this value is in encoder counts/sec. When StepNLoop is disabled, this value is in pulses/sec.
5. Motor status
   i. IDLE – motor is not moving.
   ii. ACCEL – motor is accelerating
   iii. CONST – motor is running in constant speed
   iv. DECEL – motor is decelerating
   v. ERROR – limit error
6. StepNLoop status
   i. NA – StepNLoop disabled
   ii. IDLE – motor is not moving
   iii. MOVING – motor is performing a target movement
   iv. CORRECTING – motor is correcting its position
   v. STOPPING – motor is decelerating to a stop
   vi. ABORTING – motor is immediately stopping
   vii. JOGGING – motor is jogging
   viii. HOMING – motor is homing
   ix. Z-HOMING – motor is homing using the Z-index
   x. ERR-RANGE – correction range error
   xi. ERR-ATMPT – correction attempt error
   xii. ERR-STALL – stall error
   xiii. ERR-LIM – limit error

7. Mode
    i. ABS – Target movement moves to the absolute target position
    ii. INC – Target movement increments/decrements by the target position amount
8. Current – displays active current value. Value is in mA.
9. S-curve enable status
10. –Limit, +Limit, and Home input status
11. Clear motor status – use this to clear the motor or StepNLoop error
12. Z encoder index channel status.

B. Control

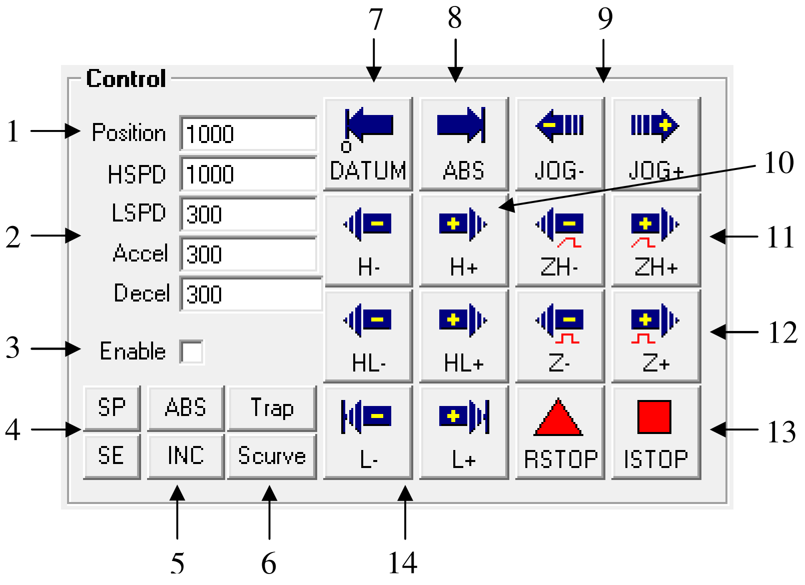


Figure 6.6

1. Target position entered here.
2. High speed, low speed, acceleration and deceleration parameters are entered here.
3. Enable – motor power is turned on or off using this check box. When motor is disabled, no motion is done.
4. SET POS/SET ENC – Set the pulse position or encoder position. Positions are set using the position box value.
5. INC/ABS – Select the Absolute Mode or Incremental Mode
6. Trap/Scurve – Select the acceleration profile to be used.
7. DAT – moves the motor to the zero target position.
8. ABS – moves the motor to the target absolute position using the high speed and the low speed and the acceleration values.
9. JOG+/JOG- - jogs the motor in positive and negative direction.
10. H+/H- - homing is done using only the home sensor. When the home sensor is triggered during homing, the position counter is reset to zero

and the motor decelerates to low speed and stops. After homing, the position is not necessarily zero due to deceleration after the trigger of the home switch.

11. ZH+/ZH- - Home sensor and encoder index channel is used to home.
12. Z+/Z- - Only the encoder index channel is used for homing.
13. RSTOP/ISTOP – motion is either stopped with deceleration using RSTOP or without deceleration using ISTOP.
14. L+/L- - Homing using the Limit inputs.

C. Digital Input and Output Status

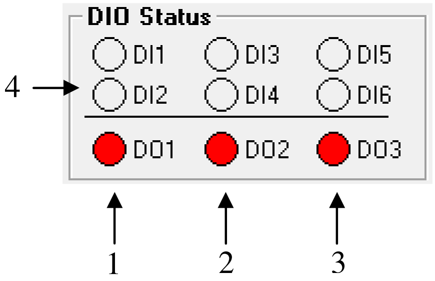


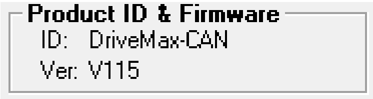
Figure 6.7

1. Digital Output bit 1 – this is general purpose digital output when StepNLoop is disabled. When StepNLoop is enabled, DO1 is used as In-Position output unless **EDO** is disabled.
2. Digital Output bit 2 – this is general purpose digital output when Sync Output is disabled. When Sync Output is enabled, DO2 is used for Sync Digital Output function.
3. Digital Output bit 3 – this is general purpose digital output when StepNLoop is disabled. When StepNLoop is enabled, DO3 is used as alarm output unless **EDO** is disabled.
4. Digital input 2 – when position capture feature is enabled, DI2 is used as the capture input where the encoder and pulse positions are recorded when DI2 is triggered.

Remaining DI's are general purpose digital inputs.

Digital outputs can be toggled by clicking on the digital output picture.

D. Product ID & Firmware

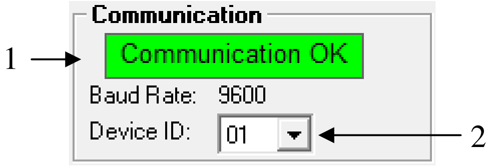


Product ID & Firmware
ID: DriveMax-CAN
Ver: V115

Figure 6.8

Displays the product ID of DMX-CAN as well as the firmware version of the module.

E. Communication

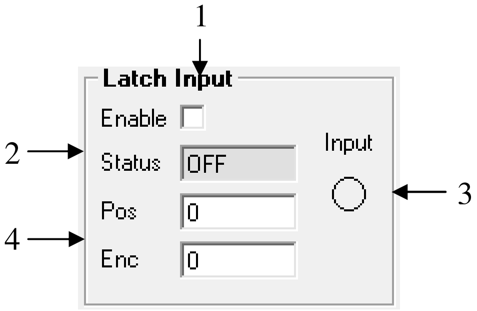


Communication
Communication OK
Baud Rate: 9600
Device ID: 01

Figure 6.9

1. Communication Status – Displays communication status with the selected device.
2. Device ID – Device ID of the communicating DMX-CAN. To communicate with a different DMX-CAN on-the-fly, select another ID number from this drop-down box.

F. Latch Input



Latch Input
Enable □
Status OFF
Pos 0
Enc 0
Input ○

Figure 6.10

1. Enable latch using this check box.
2. Status if either OFF, ON, or WAITING.
3. To trigger the latch, activate this latch input.
4. Once the latch is triggered the current pulse position and encoder position will be recorded here.

G. Sync Output

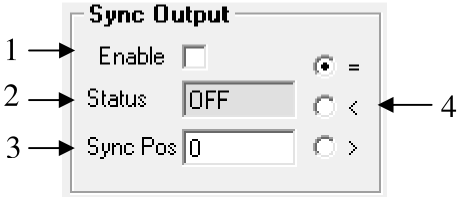
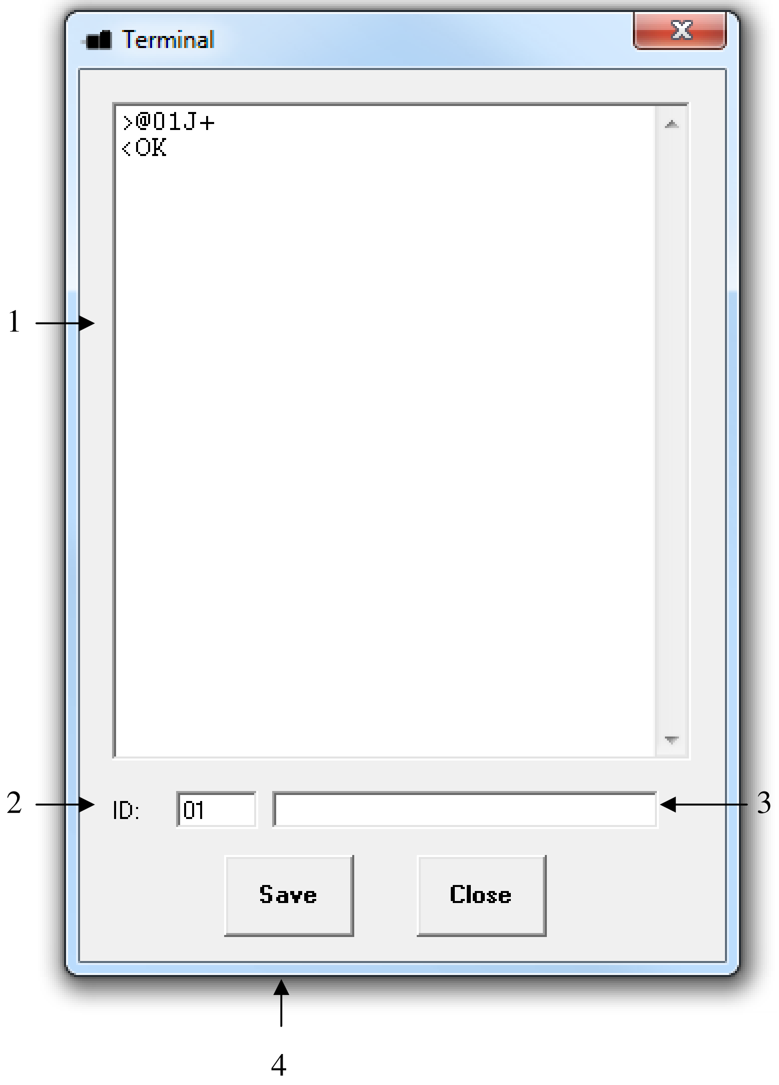


Figure 6.11

1. Enable Sync Output
2. Status is either OFF, ON, WAITING
3. The sync position used for comparison to the encoder position
4. Set the conditional used for the Sync output. Once this condition is met, the sync output (DO2) is activated.

H. Terminal

Send Terminal based commands to the DMX-CAN. Click on the button above to display the following window.



```
>@01J+
<OK
```

ID: 01

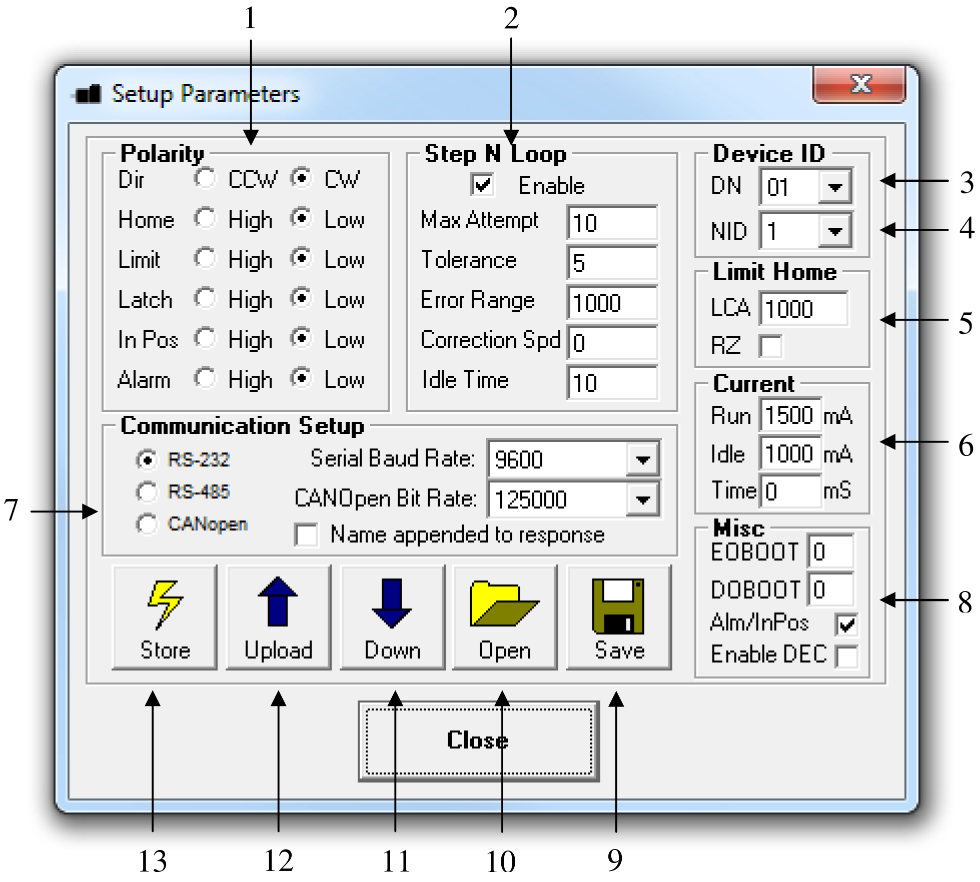
Figure 6.12

1. Response Window – Displays the response from the command line
2. Address – Select the address of the DMX-CAN module which you wish to communicate. Selecting address **‘00’** will send a broadcast command which will be received by all DMX-CAN modules on a RS-485 bus.
3. Send command – Type the commands here manually. When sending commands, you do not need to type the device name. For example, when you want to know what the motor status is, type MST and you will see a number on the reply that represents the status of the motor. Press the Enter key to send the command.
4. Save – Save the terminal text into a text file. When this button is pressed, typical Windows file save dialog box will open:

I. Setup

DMX-CAN configuration values are automatically loaded when the program is started. All the configuration changes are uploaded and download all at once. This means that in order for the configuration to become effective, download button must be pressed. In order for the configuration to be permanent, store to flash must be pressed.

Click on the Setup button to show the following display box:



Setup Parameters

Polarity
Dir ○ CCW ● CW
Home ○ High ● Low
Limit ○ High ● Low
Latch ○ High ● Low
In Pos ○ High ● Low
Alarm ○ High ● Low

Step N Loop
☑ Enable
Max Attempt 10
Tolerance 5
Error Range 1000
Correction Spd 0
Idle Time 10

Device ID
DN 01
NID 1

Limit Home
LCA 1000
RZ □

Current
Run 1500 mA
Idle 1000 mA
Time 0 mS

Misc
EOBOOT 0
DOBOOT 0
Alm/InPos ☑
Enable DEC □

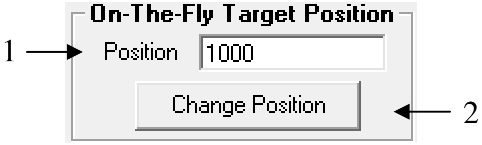
Communication Setup
● RS-232
○ RS-485
○ CANopen
Serial Baud Rate: 9600
CANOpen Bit Rate: 125000
□ Name appended to response

Figure 6.13

1. Polarity values – direction, home, limit, latch input, in position output, and alarm output values can be selected in this section.
2. StepNLoop configuration values – See detailed description in the StepNLoop feature section.
3. Device ID – Device name ranges from DMK01 to DMK99.

4. CAN Node ID – CAN Node ID ranges from 1 to 127.
5. Limit Correction Amount – Set limit correction amount. See detailed description in Home Limit Error Correction section.
6. Current values – Run current is used when the motor is running. Idle current is used when the Idle time expires. Idle time is in msec. Minimum current setting is 100 mA and maximum current setting is 2500 mA. Depending on the model of the motor, the current setting should not go above the recommended maximum rated current of the motor. If Idle current is set to 0 mA, the motor will become disabled once it is idle. When the motor starts running, the motor will be enabled again.
7. Communication mode selection – RS-232, RS-485, or CANopen is selected as the communication method.
    - a. For RS-232/RS-485, set baud rate ranging from 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps.
    - b. For CANopen, set bit rate ranging from 10000 bps, 20000 bps, 50000 bps, 100000 bps, 125000 bps, 250000 bps, 500000 bps, 800000 bps, 1000000 bps.
8. Miscellaneous Settings.
    - a. Set EOBOOT. Configure the status of the motor enable at boot-up
    - b. Set DOBOOT. Configure the status of the digital outputs at boot-up
    - c. Set EDO. Enable/Disable alarm/in position digital outputs.
    - d. Enable/disable deceleration. If disabled, the ACC parameter will be used for deceleration.
9. Save – Save the current parameters into a file. When this button is pressed, typical Windows file save dialog box will open.
10. Open – Open a file of saved parameters. Parameters will be loaded into the configuration box. Before parameters are made effective, you must perform a download procedure When this button is pressed, typical Windows file open dialog box will open.
11. Download – all the configuration values on the screen is downloaded.
12. Upload – all the configuration values on the controller is uploaded and updated on the screen.
13. Store to Flash – all configuration values in the DMX-CAN is stored to flash memory. In order to update any changes made, download the changes and then to make it permanent use Store to Flash button.

J. On-The-Fly-Position



Position: 1000

Figure 6.14

1. Enter new target position here.
2. Start on-the-fly-position change operation.

K. On-The-Fly-Speed

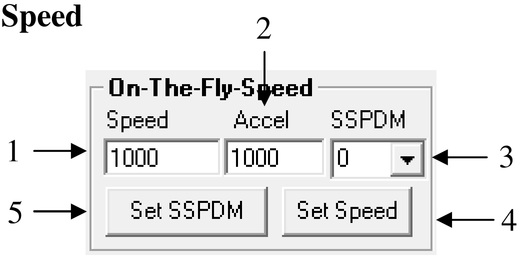


Figure 6.15

1. Desired Speed – Once the “Set Speed” button is clicked, the speed will change on-the-fly to the desired speed.
2. Desired Acc/Dec – The acceleration/deceleration use for the on-the-fly speed change operation
3. Before setting the DMX-CAN into motion, set the **SSPDM** parameter. To see which to use, see the on-the-fly speed change section.
4. Set the **SSPDM** parameter. Note that is an on-the-fly speed change operation is to be used, this parameter must be set before the controller goes to start motion.
5. Set **SSPD[value]** – Start the on-the-fly speed change operation.

7. Motion Control Overview

Important Note: All the commands described in this section are for ASCII communication over a RS-232/RS-485 network. For details on communication over CANOpen, please see the "Object Dictionary [CANOpen]" section.

Built-in encoder

DMX-CAN comes with a 1000 line encoder. With quadrature decoding, 4000 count/rev resolution is reached. Use the **EX** command to read and set the encoder position. Pulse position can be read and set using the **PX** command.

When StepNLoop closed-loop control is *enabled*:

EX command returns encoder position
PX command returns the real-time target position of your move

When StepNLoop closed-loop control is *disabled*:

EX command returns encoder position
PX command returns pulse position

Built-in Microstep Driver

DMX-CAN has an integrated micro-step driver. The micro-step setting is fixed at 16. With a 1.8° motor, this results in a 3200 step/rev resolution.

Motion Profile

By default, DMX-CAN incorporates trapezoidal velocity profile as shown below.

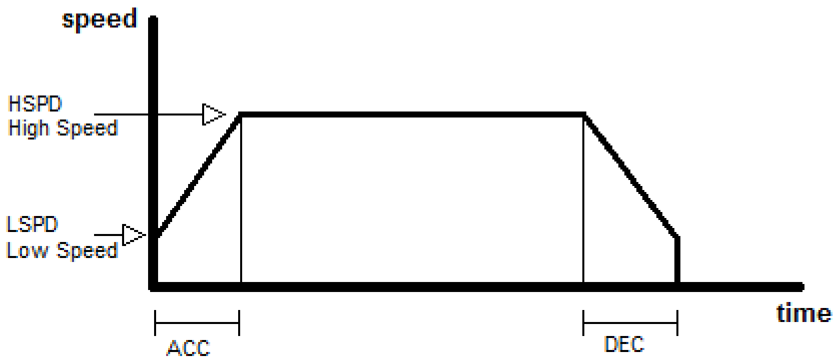


Figure 7.0

High speed and low speed are in pps (pulses/second). Use **HSPD/LSPD** commands to modify the high speed and low speed settings.

Acceleration and deceleration time are in milliseconds. Use the **ACC/DEC** command to modify the acceleration and deceleration values.

S-curve velocity profile can also be achieved by using the **SCV** command. See Figure 7.1.

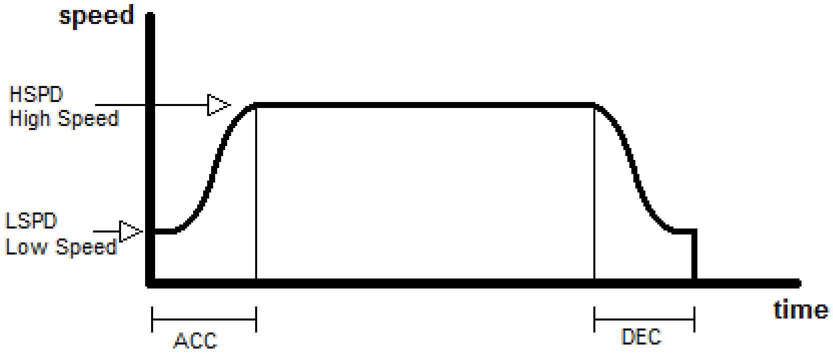


Figure 7.1

Notes:
The minimum and maximum acceleration values depend on the high speed and low speed settings. Refer to Table A.0 and Figure A.0 in **Appendix A** for details.

Digital Inputs/Outputs

DMX-CAN comes with 6 digital inputs and 3 digital outputs.

Inputs
Read digital input status using the **DI** command.

Digital input values can also be referenced one bit at a time by the **DI[1-6]** commands. Note that the indexes are 1-based for the bit references (i.e. DI1 refers to bit 0, not bit 1)

Bit	Description	Bit-Wise Command
0	Digital Input 1	DI1
1	Digital Input 2	DI2
2	Digital Input 3	DI3
3	Digital Input 4	DI4
4	Digital Input 5	DI5
5	Digital Input 6	DI6

Table 7.0

Outputs
Use the **DO** command to drive digital outputs. DO value must be within the range of 0-7. Digital output values can also be referenced one bit at a time by the **DO[1-3]** commands. Note that the indexes are 1-based for the bit references (i.e. DO1 refers to bit 0, not bit 1)

Bit	Description	Bit-Wise Command
0	Digital Output 1 (In Position)	DO1

1	Digital Output 2	DO2
2	Digital Output 3 (Alarm)	DO3

Table 7.1

If StepNLoop control and **EDO** are enabled, DO1 is used as an “In Position” status output, and DO3 is used as an “Alarm” output.

To use DO1 and DO3 as general purpose outputs while StepNLoop is enabled, set **EDO=0.**

The initial state of the digital outputs can be defined by setting the **DOBOOT** register to the desired initial digital output value. The value is stored to flash memory once the **STORE** command is issued.

Motor Power

Using the **EO** command, the motor power can be enabled or disabled.

The initial state the enable output can be defined by setting the **EOBOOT** register to the desired initial value. The value is stored to flash memory once the **STORE** command is issued.

Polarity

Using the **POL** command, polarity of following signals can be configured:

Bit	Description
0	Reserved
1	Direction
2	Reserved
3	Reserved
4	Limit
5	Home
6	Latch
7	In Position Output
8	Alarm Output

Table 7.2

Positional Moves

DMX-CAN can operate in either incremental or absolute move modes. Use **X** command to make moves. Use **INC** and **ABS** commands to set the move mode. Use **MM** command to read the current move mode.

Note: If a motion command is sent while the controller is already moving, the command is not processed. Instead, an error response is returned.

On-The-Fly Target Position Change

On-the-fly target position change can be achieved using the **T[value]** command. While the motor is moving, **T[value]** will change the final destination of the motor. If the motor has already passed the new target position, it will reverse direction once the target position change command is issued.

Note: If a **T** command is sent while the controller is not performing a target move, the command is not processed. Instead, an error response is returned.

Jogging

Jogging is available for continuous speed operation. Use **J+** and **J-** commands to jog in positive or negative direction.

Stopping Motor

When the motor is moving, jogging, or homing, using the **ABORT** command will immediately stop the motor. Using the **STOP** command will decelerate the motor to low speed before stopping.

Homing

Home search sequence involves moving the motor towards the home or limit switches and then stopping when the relevant input is detected. The DMX-CAN has four different homing routines:

Home Input Only (High speed only)

Use the **H+/H-** command. Figure 7.2 shows the homing routine.

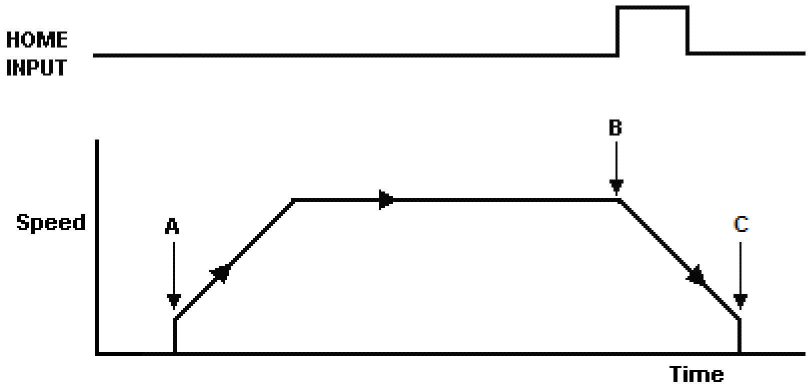


Figure 7.2

A. Starts the motor from low speed and accelerates to high speed.
B. As soon as the home input is triggered, the position counter is reset to zero and the motor begins to decelerate to low speed. As the motor decelerates, the position counter keeps counting with reference to the zero position.
C. Once low speed is reached, the motor stops. The position is non-zero.

Home Input and Z-index

Use the **ZH+/ZH-** command. Figure 7.3 shows the homing routine.

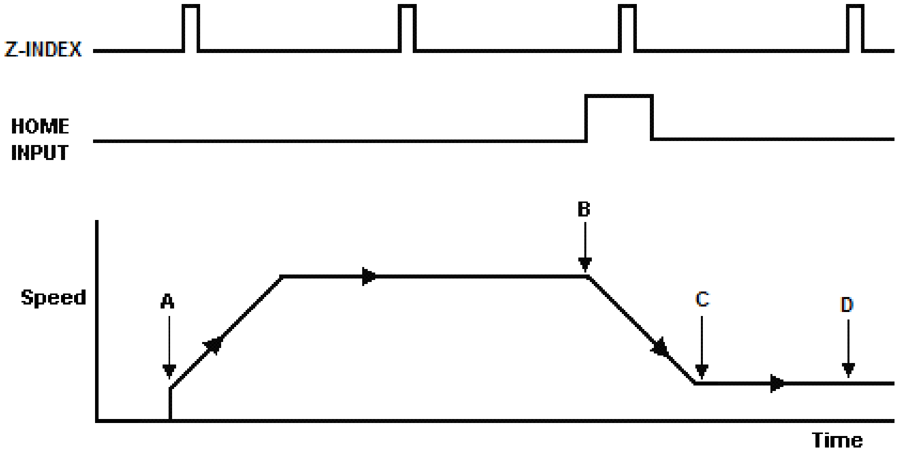


Figure 7.3

A. Issuing the command starts the motor from low speed and accelerates to high speed.
B. As soon as the home input is triggered, the motor decelerates to low speed
C. Once low speed is reached, the motor begins to search for the z-index pulse.
D. Once the z-index pulse is found, the motor stops and the position is set to zero.

Home Input Only (High speed and low speed)

Use the **HL+/HL-** command. Figure 7.4 shows the homing routine.

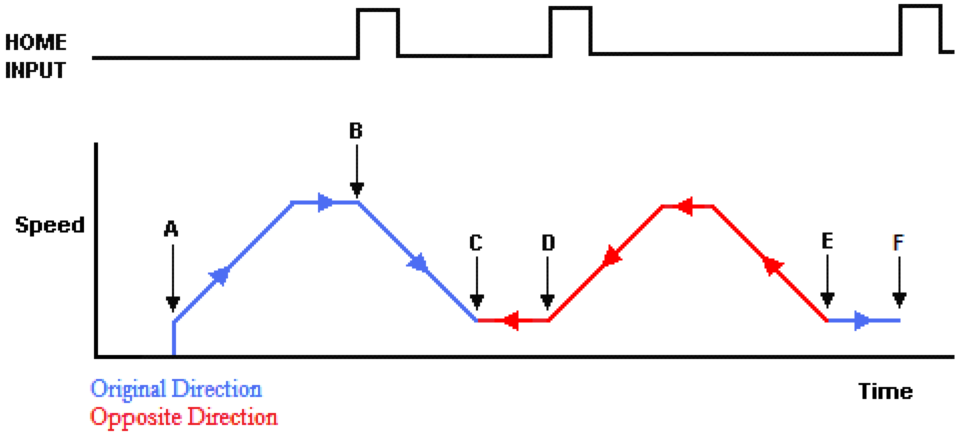


Figure 7.4

A. Starts the motor from low speed and accelerates to high speed.
B. As soon as the home input is triggered, the position counter is reset to zero and the motor decelerates to low speed.
C. Once low speed is reached, the motor reverses direction to search for the home switch.
D. Once the home switch is reached, it will continue past the home switch by the amount defined by the home correction amount (**HCA**) at high speed.

E. The motor is now past the home input by the amount defined by the home correction amount (**HCA**). The motor now moves back towards the home switch at low speed.
F. The home input is triggered again, the position counter is reset to zero and the motor stops immediately

Note: For **H**, **HL** homing routines, it is possible to have the motor automatically return to the zero position. To do so, set the **RZ=1**.

Limit Only

Use the **L+/L-** command. Figure 7.4 shows the homing routine.

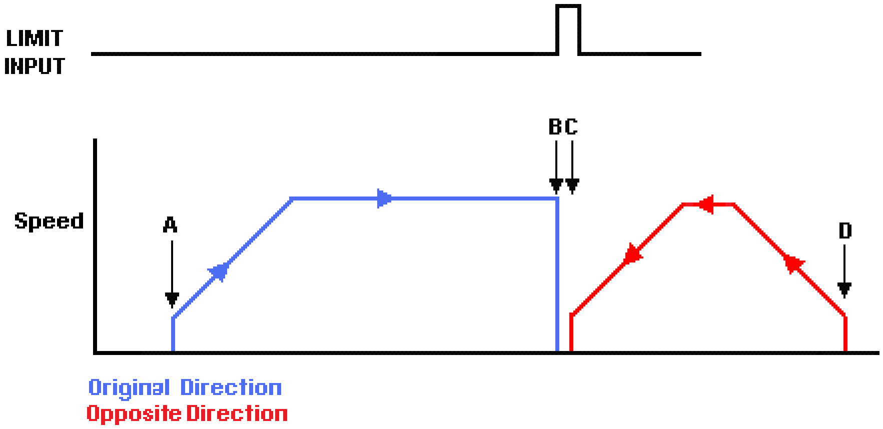


Figure 7.5

A. Issuing the command starts the motor from low speed and accelerates to high speed.
B. The corresponding limit is triggered and the motor stops immediately.
C. The motor reverses direction by the amount defined by the limit correction amount (**LCA**) at high speed.
D. The zero position is reached.

Z-index only

Use the **Z+/Z-** command. Figure 7.5 shows the homing routine.

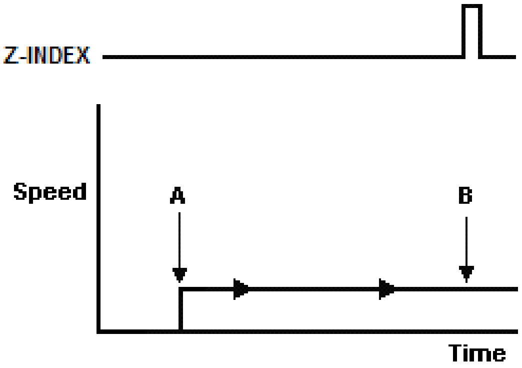


Figure 7.6

A. Issuing the command starts the motor at low speed.
B. Once the z-index pulse is found, the motor stops and the position is set to zero.

Motor Position

Motor position can be set and read by using the **PX** command.
Encoder position can be set and read by using the **EX** command.

Motor Status

Motor status can be read anytime by reading the response to the **MST** command. The following is the bit representation of motor status:

Bit	Description
0	Motor running at constant speed
1	Motor in acceleration
2	Motor in deceleration
3	Home input switch status
4	Minus limit input switch status
5	Plus limit input switch status
6	Minus limit error. This bit is latched when minus limit is hit during motion. This error must be cleared using the **CLR** command before issuing any subsequent move commands.
7	Plus limit error. This bit is latched when plus limit is hit during motion. This error must be cleared using the **CLR** command before issuing any subsequent move commands.
8	Latch input status
9	Z-index status

Table 7.3

Examples:

- When motor status value is 0, motor is idle and all input switches are off.
- When motor status value is 2, motor is in acceleration.
- When motor status value is 9, motor is moving in constant high speed and home input switch is on.
- When motor status value is 64, motor is in minus limit error. Use **CLR** command to clear the error before issuing any more move commands.

Limit Inputs

If positive limit switch is triggered while moving in positive direction, the motor will immediately stop and the motor status bit for positive limit error is set. The same is for the negative limit while moving in the negative direction. Once the limit error is set, use the **CLR** command to clear the error. Once the error is cleared, move the motor out of the limit switch.

Latch Input

DMX-CAN has high speed position latch input DI2.

This input performs high speed position capture of both pulse and encoder positions but does not reset the pulse or encoder position counters.

Note: When StepNLoop mode is enabled, the position value should be ignored.
Use the **LT** command to enable and disable latch feature. To read the latch status, use **LTS** command.

Following are return value description for **LTS** command.

Return Value	Description
0	Latch off
1	Latch on and waiting for latch trigger
2	Latch triggered

Table 7.4

Once the latch is triggered, the triggered position can be retrieved using **LTP** (latched pulse position) and **LTE** (latched encoder position) commands.

Sync Output

DMX-CAN has a designated synchronization digital output (DO2). The synchronization signal output is triggered when the encoder position value meets the set condition.

While this feature is enabled, the designated digital output (DO2) cannot be controlled by user.

Use **SYNO** to enable the synchronization output feature.

Use **SYNF** to disable the synchronization output feature.

Use **SYNP** to read and set the synchronization position value. (28-bit signed number)

Use **SYNC** to set the synchronization condition.

1 – Turn the output on when the encoder position is EQUAL to sync position.
If the synchronization output is done during motion, the sync output pulse will turn on only when the encoder position and sync position are equal.
2 – Turns output on when the encoder position is LESS than the sync position.
3 – Turns output on when the encoder position is GREATER than sync position.

Use **SYNS** to read the synchronization output status.

0 – Sync output feature is off
1 – Waiting for sync condition
2 – Sync condition occurred

StepNLoop Closed Loop Control

DMX-CAN features a closed-loop position verification algorithm called StepNLoop (SNL). The algorithm requires the use of an incremental encoder.

SNL performs the following operations:

1) Position Verification: At the end of any targeted move, SNL will perform a correction if the current error is greater than the tolerance value.
2) Delta Monitoring: The delta value is the difference between the actual and the target position. When delta exceeds the error range value, the motor is stopped and the SNL Status goes into an error state. Delta monitoring is performed during moves – including homing and jogging. To read the delta value, use the **DX** command.

See Table 7.5 for a list of the SNL control parameters.

SNL Parameter	Description	Command
Tolerance	Maximum error between target and actual position that is considered "In Position". In this case, no correction is performed. Units are in encoder counts.	**SLT**
Error Range	Maximum error between target and actual position that is not considered a serious error. If the error exceeds this value, the motor will stop immediately and go into an error state.	**SLE**
Correction Attempt	Maximum number of correction tries that the controller will attempt before stopping and going into an error state.	**SLA**
Idle Tolerance	After correction, if the remaining error is greater than this value, an additional correction will be performed using idle current instead of run current.	**SLM**

Table 7.5

To enable/disable the SNL feature use the **SL** command. To read the SNL status, use **SLS** command to read the status.

See Table 7.6 for a list of the **SLS** return values.

Return Value	Description
0	Idle

1	Moving
2	Correcting
3	Stopping
4	Aborting
5	Jogging
6	Homing
7	Z-Homing
8	Correction range error. To clear this error, use **CLRS or CLR** command.
9	Correction attempt error. To clear this error, use **CLRS or CLR** command.
10	Stall Error. **DX** value has exceeded the correction range value. To clear this error, use **CLRS or CLR** command.
11	Limit Error
12	N/A (i.e. SNL is not enabled)
13	Limit homing

Table 7.6

See Table 7.7 for SNL behavior within different scenarios.

Condition	**SNL behavior (motor is moving)**	**SNL behavior (motor is idle)**
δ <= SLT	Continue to monitor the **DX**	In Position. No correction is performed.
δ > SLT AND δ < SLE	Continue to monitor the **DX**	Out of Position. A correction is performed.
δ > SLT AND δ > SLE	Stall Error. Motor stops and signals and error.	Error Range Error. Motor stops and signals and error.
Correction Attempt > SLA	NA	Max Attempt Error. Motor stops and signals and error.

Table 7.7

Key

[δ]:	Error between the target position and actual position
SLT:	Tolerance range
SLE:	Error range
SLA:	Max correction attempt

Notes:

Once SNL is enabled, position move commands are in term of encoder position. For example, X1000 means to move the motor to encoder 1000 position.

Once SNL is enabled, the speed is in encoder speed. For example HSPD=1000 when SNL is enabled means that the target high speed is 1000 encoder counts per second.

If **EDO** is enabled while SNL is enabled, DO1 is dedicated as the "In Position" output and DO3 is dedicated as the "Alarm" output. In order to use the digital outputs for general purpose, disable **EDO** by setting **EDO=0**.

Idle Current and Run Current

DMX-CAN allows for two current settings.

Run Current: Current used while the motor is running. Set using the **CURR** command
Idle Current: Current used when the motor is idle. Set using the **CURI** command

To set the amount of time the motor needs to be idle before changing to idle current, use the **CURT** command. Units are in ms.

To read the actual current at anytime, use the **CUR** command

When setting idle and run current, the range must be within 100mA to 2500mA, unless the user wishes to have the motor become disabled during idle state. To do this, set the idle current to 0.

RS-232/RS-485/CANopen Selection:

In order to choose between RS-232, RS-485, or CANopen communication, use the **CM** command.

A value of 0 corresponds to RS-232 communication. A value of 1 corresponds to RS-485 communication. A value of 2 corresponds to CANopen communication. By default, the **CM** value is set to 0.

To write the value to flash memory, use the **STORE** command. After a complete power cycle, the new communication method will be used. Note that before a power cycle is done, the setting will not take effect.

Node ID, Device Number, and Baud Rate:

DMX-CAN comes with following default factory communication setting:

Baud Rate:	**9600**
Device Name:	**DMC01**
Node ID:	**1**

DMX-CAN module provides the user with the ability to set the device number for RS-485 multi-drop applications. In order to make these changes, first set the desired device number using the **DN** command. Please note that this value must be within the range [DMC01,DMC99].

The node ID of the DMX-CAN module can be set by the user for using multiple units on the CANopen bus. To change the node ID, set the desired value using the **NID** command. Please not that this value must be within the range [1,127].

DMX-CAN module provides the user with the ability to change the baud rate for RS-232 and RS-485 communication. In order to make these changes, first set the desired baud rate using the **DB** command. Please note the following baud rate codes:

Return Value	Description
1	9600 bps
2	19200 bps
3	38400 bps
4	57600 bps
5	115200 bps

Table 7.8

DMX-CAN module also provides the user with the ability to change the baud rate for CANOpen communication. To make these changes set the desired rate using the **BR** command. Please note the following bit rate codes:

Return Value	Description
1	10000 bps
2	20000 bps
3	50000 bps
4	100000 bps
5	125000 bps
6	250000 bps
7	500000 bps
8	800000 bps
9	1000000 bps

Table 7.9

To write the values to the device number and baud rate permanently to flash memory, use the **STORE** command. After a complete power cycle, the new device ID will be used. Note that before a power cycle is done, the settings will not take effect.

Broadcasting over RS-485

The address **'00'** is reserved for broadcasting over an RS-485 bus. Any ASCII command prefixed by **'@00'** will be processed by all DMX-CAN modules on the RS-485 bus. When a broadcast command is received by a DMX-CAN module, no response is sent back to the master.

Response type selection

It is possible to choose between two types of response string formats. This parameter can be set using the **RT** command.

Format 1 (default): [Response][Null]

```
Examples:
    For querying the encoder position
        Send:   @01EX[CR]
        Reply:  1000[Null]

    For jogging the motor in positive direction
        Send:   @01J+[CR]
        Reply:  OK[Null]

    For aborting any motion in progress
        Send:   @01ABORT[CR]
```

To achieve this response string type, send command **RT=0**.

Format 2: #[DeviceName][Response][Null]

```
Examples:
    For querying the encoder position
        Send:   @01EX[CR]
        Reply:  #011000[Null]

    For jogging the motor in positive direction
        Send:   @01J+[CR]
        Reply:  #01OK[Null]

    For aborting any motion in progress
        Send:   @01ABORT[CR]
        Reply:  #01OK[Null]
```

To achieve this response string type, send command **RT=1**.
To write the response type parameter to flash memory, use the **STORE** command. After a complete power cycle, the new response type will take effect. Note that before a power cycle is done, the setting will not take effect.

Storing to Flash

The following items are stored to flash:

ASCII Command	Description
BR	Baud rate (for CANOpen)
CM	Communication method (RS-232/RS-485)
CURI,CUR,CURT	Driver current settings
DB	Serial communication baud rate

DN	Device name
DOBOOT	DO configuration at boot-up
EDO	Enable in-pos/alarm outputs
EOBOOT	EO configuration at boot-up
LCA	Limit correction amount
POL	Polarity settings
RT	ASCII response type
SL, SLR, SLE, SLT, SLA, SLM	StepNLoop parameters

Table 7.10

Default Settings

Following are the factory default settings when then unit is shipped from the factory.

Command	Parameter Description	Value
BR	Baud-rate (for CANOpen)	125000 bps
CM	Communication Method	RS-232
CURI	Idle Current	1000 mA
CURR	Run Current	1600 mA
CURT	Idle Time	500 mSec
DB	Baud Rate (for RS-232/RS-485)	9600 bps
DN	Device ID	DMC01
DOBOOT	Digital Output Boot-up State	7
EDO	Alarm/ In Pos Output Mode	Enabled
EOBOOT	Enable Output Boot-up State	0
LCA	Limit Correction Amount	1000
NID	Node ID (for CANOpen)	1
POL (bit 1)	Direction Polarity	CW
POL (bit 4)	Limit Polarity	Active Low
POL (bit 5)	Home Polarity	Active Low
POL (bit 6)	Latch Polarity	Active Low
POL (bit 7)	In Position Output Polarity	Active Low
POL (bit 8)	Alarm Output Polarity	Active Low
RT	Response type	Do Not Append
SL	StepNLoop	Enabled
SLA	StepNLoop Maximum Attempt	10
SLE	StepNLoop Error Range	1000
SLM	StepNLoop Idle Tolerance	5
SLT	StepNLoop Tolerance Range	20

Table 7.11

8. Object Dictionary [CANOpen]

CANOpen contains a set of profiles that can be used for various different applications. DMX-CAN implements the CiA DSP-402 Device Profile for Drives and Motion Control.

The object dictionary stores all the parameters of the DMX-CAN. It is divided into the following sections.

Index	Description
0001_h-$009F_h$	Data Type Definitions
$00A0_h$-$0FFF_h$	Reserved
1000_h-$1FFF_h$	Communication Profile Area (CiA 301)
2000_h-$5FFF_h$	Manufacturer Specific Profile Area
6000_h-$9FFF_h$	Standardized Profile Area (CiA 402)
$A000_h$-$FFFF_h$	Reserved

Table 8.0

Manufacturer Specific Objects [2000_h-$5FFF_h$]

All manufacturer specific objects in this section only pertain to Arcus Technology motion products.

Object 2000_h: Alarm/In Position Enable

This object disables or enables the alarm/in position feature available on the DMX-CAN. When enabled, this feature will monitor and indicate the status of the closed loop algorithm StepNLoop by using the available digital outputs. DO1 will be used as an "in-position" status output and DO3 will be used as an "alarm" output.

Object Description

Index	Name	Data Type	Value Range	Access
2000_h	Alarm/In Position Enable	Unsigned 8	0 - 1	Read/Write

Table 8.1

Data Description

Value	Description
0	Disable
1	Enable

Table 8.2

By default, the alarm/in position feature is enabled.

Object 2001_h: StepNLoop Parameters

DMX-CAN module has closed loop position control algorithm called StepNLoop control for accurate positioning of the motor using the integrated encoder. This object configures all the required parameters.

Object Description

Index	Name	Data Type
2001_h	StepNLoop Parameters	Array - Unsigned 32

Table 8.3

Sub Index Description

Sub Index	Name	Value Range	Access
00_h	Number of elements	5	Read Only

Table 8.4

Sub Index	Name	Description	Value Range	Access
01_h	StepNLoop Enable	Enable or disable StepNLoop	0 - 1	Read/Write

Table 8.5

Value	Description
0	Disable
1	Enable

Table 8.6

Sub Index	Name	Description	Value Range	Access
02_h	Max Attempt	Maximum allowable correction attempts	0 - (2^{32}-1)	Read/Write

Table 8.7

Sub Index	Name	Description	Value Range	Access
03_h	Tolerance Range	Allowable error before a correction attempt is made while the motor is moving.	0 - (2^{32}-1)	Read/Write

Table 8.8

Sub Index	Name	Description	Value Range	Access
04_h	Idle Tolerance Range	Allowable error before a correction attempt is made while the motor is idle.	0 - (2^{32}-1)	Read/Write

Table 8.9

Sub Index	Name	Description	Value Range	Access
05_h	Error Range	Maximum allowable error before motor enters error state.	0 - (2^{32}-1)	Read/Write

Table 8.10

The default StepNLoop parameters are shown in Table 8.11

Sub Index	Default Value
StepNLoop Enable (01_h)	1

Max Attempt (02_h)	10
Tolerance Range (03_h)	10
Idle Tolerance Range (04_h)	5
Error Range (05_h)	1000

Table 8.11

The Tolerance Range, Idle Tolerance Range, and Error Range sub indices are all in units of pulses.

Object 2002_h: StepNLoop Delta

Delta position is the difference between the actual and the target position in terms of encoder counts. When the delta goes over the allowed Error Range, the motor is stopped and the StepNLoop status goes into the "stall" error state.

Object Description

Index	Name	Data Type	Value Range	Access
2002_h	StepNLoop Delta	Signed 32	$-(2^{31})$ - $(2^{31}-1)$	Read Only

Table 8.12

Object 2003_h: StepNLoop Status

This object returns the current StepNLoop status.

Object Description

Index	Name	Data Type	Value Range	Access
2003_h	StepNLoop Status	Unsigned 16	0 - 13	Read Only

Table 8.13

Data Description

Value	Description
0	Idle
1	Moving
2	Correcting
3	Stopping
4	Aborting
5	Jogging
6	Homing
7	Z-Homing
8	Correction Range Error: Delta has exceeded error range while the motor is idle.
9	Correction Attempt Error: Motor has exceeded maximum correction attempts.
10	Stall Error: Delta has exceeded error range while motor is moving.
11	Limit Error
12	N/A (StepNLoop disabled)
13	Limit Homing

Table 8.14

Object 2004_h: Driver Settings

The DMX-CAN has a built in micro-step driver. This object configures all the available driver settings.

Object Description

Index	Name	Data Type
2004_h	Driver Settings	Array - Unsigned 8

Table 8.15

Sub Index Description

Sub Index	Name	Value Range	Access
00_h	Number of elements	4	Read Only

Table 8.16

Sub Index	Name	Description	Value Range	Access
01_h	Real-time Current	Return the real-time current of the driver.	100 - 2500 mA	Read Only

Table 8.17

Sub Index	Name	Description	Value Range	Access
02_h	Run Current	Current used while the motor is moving	100 - 2500 mA	Read/Write

Table 8.18

Sub Index	Name	Description	Value Range	Access
03_h	Idle Current	Current used while the motor is idle	100 - 2500 mA	Read/Write

Table 8.19

Sub Index	Name	Description	Value Range	Access
04_h	Idle Time	Time the motor waits before changing from run current to idle current	1 - 10000 ms	Read/Write

Table 8.20

The default driver settings are shown in table 8.21.

Sub Index	Default Value
Run Current (02_h)	1500 mA
Idle Current (03_h)	500 ms
Idle Time (04_h)	500 ms

Table 8.21

Object 2005_h: Digital Output Bootup Status

The initial state of the digital outputs can be defined by configuring this object.

Object Description

Index	Name	Data Type	Value Range	Access
2005_h	Digital Output Boot	Unsigned 8	0 - 7	Read/Write

Table 8.22

To toggle the initial state of a digital output, the corresponding bit must be changed. Table 8.23 shows the bit mapping for digital outputs.

Bit	Description
0	Digital Output 1
1	Digital Output 2
2	Digital Output 3

Table 8.23

Object 2006_h: Enable Output Bootup Status

The initial state of the enable output can be defined by configuring this object.

Object Description

Index	Name	Data Type	Value Range	Access
2006_h	Enable Output Boot	Unsigned 8	0 - 1	Read/Write

Table 8.24

Object 2007_h: Latch Enable

The latch operation performs high speed position capture of both the pulse and encoder positions. Digital input 2 is used as the latch input trigger.

Note: When StepNLoop mode is enabled, the position value should be ignored.

Object Description

Index	Name	Data Type	Value Range	Access
2007_h	Latch Enable	Unsigned 8	0 - 1	Read/Write

Table 8.25

Data Description

Value	Description
0	Disable
1	Enable

Table 8.26

Object 2008_h: Latch Status

This object returns the current latch status.

Object Description

Index	Name	Data Type	Value Range	Access
2008_h	Latch Status	Unsigned 8	0 - 1	Read Only

Table 8.27

Data Description

Value	Description
0	Latch off
1	Latch on and waiting for latch trigger
2	Latch triggered

Table 8.28

Object 2009_h: Latch Positions

Once the latch is triggered, the pulse/encoder positions can be retrieved by this object.

Object Description

Index	Name	Data Type
2009_h	Latch Positions	Array - Signed 32

Table 8.29

Sub Index Description

Sub Index	Name	Value Range	Access
00_h	Number of elements	2	Read Only

Table 8.30

Sub Index	Name	Description	Value Range	Access
01_h	Pulse Position	The latched pulse position	$-(2^{31}) - (2^{31}-1)$	Read Only

Table 8.31

Sub Index	Name	Description	Value Range	Access
02_h	Encoder Position	The latched encoder position	$-(2^{31}) - (2^{31}-1)$	Read Only

Table 8.32

Object $200A_h$: Communication Type

This object determines if the device communicates over CAN, RS-232, or RS-485. If configured to communicate over RS-232 or RS-485, details on the ASCII protocol can be found in Section 7.

Object Description

Index	Name	Data Type	Value Range	Access
$200A_h$	Communication Type	Unsigned 8	0 - 2	Read/Write

Table 8.33

Data Description

Value	Description
0	RS-232
1	RS-485
2	CAN

Table 8.34

Object 200B$_h$: Bit Rate

This object configures the bit rate for CAN communication.

Object Description

Index	Name	Data Type	Value Range	Access
200B$_h$	Bit Rate	Unsigned 8	1 - 8	Read/Write

Table 8.35

Data Description

Value	Description
1	10,000
2	20,000
3	50,000
4	100,000
5	125,000
6	250,000
7	500,000
8	800,000
9	1,000,000

Table 8.36

Object 200C$_h$: Node ID

The node ID of the CANOpen device can be configured by this object.

Object Description

Index	Name	Data Type	Value Range	Access
200C$_h$	Node ID	Unsigned 8	1-127	Read/Write

Table 8.37

Object 200D$_h$: Clear Error

This object clears the motor and any and all error states. To clear an error, write a 1 to this object. The device well then clear the error once the trigger has occurred and the object will be reset to zero, indicating that the error has been cleared. See below for a timing diagram:

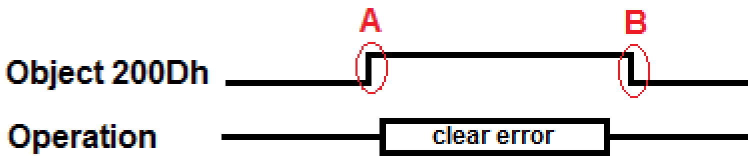


Figure 8.1

A: Object 200D$_h$ activated, indicating that an error need to be cleared.
B: The device has reset 200D$_h$ after all errors have been cleared.

Object Description

Index	Name	Data Type	Value Range	Access
200D$_h$	Clear Error	Unsigned 8	0 - 1	Read/Write

Table 8.38

Object 200E$_h$: Perform Store

This object stores all relevant settings and parameters to non-volatile flash. To store to flash, write a 1 to this object. The device well then save all parameters and store them to flash and the object will be reset to zero, indicating that the store has been performed. See below for a timing diagram:

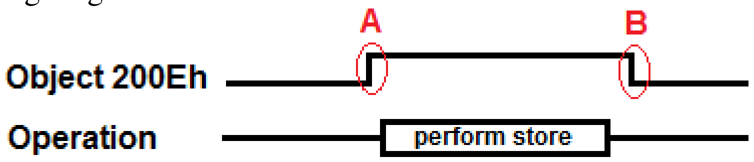


Figure 8.2

A: Object 200E$_h$ activated, indicating that all parameters need to be stored to flash.
B: The device has reset 200E$_h$ after all parameters have been saved successfully.

Object Description

Index	Name	Data Type	Value Range	Access
200E$_h$	Perform Store	Unsigned 8	0 - 1	Read/Write

Table 8.39

The following objects are saved to flash.

Object	Description
2000$_h$	Alarm/In Position Enable
2001$_h$	StepNLoop Parameters
2004$_h$	Driver Settings
2005$_h$	Digital Output Boot
2006$_h$	Enable Output Boot
200A$_h$	Communication Type

200B$_h$	Bit Rate
200C$_h$	Node ID
607C$_h$	Home Offset
607E$_h$	Polarity
6086$_h$	Motion Profile Type

Table 8.40

CiA DSP-402 [6000$_h$-9FFF$_h$]

The DMX-CAN implements the CiA DSP-402 protocol for drives and motion control. Of the available operation modes in this protocol, the DMX-CAN utilizes the Homing, Profile Velocity, and Profile Positions modes.

The DSP-402 state machine allows the user to enable the motor, recover from faults, and enter or exit the various modes of the DMX-CAN. Figure 8.1 shows the DSP-402 state machine.

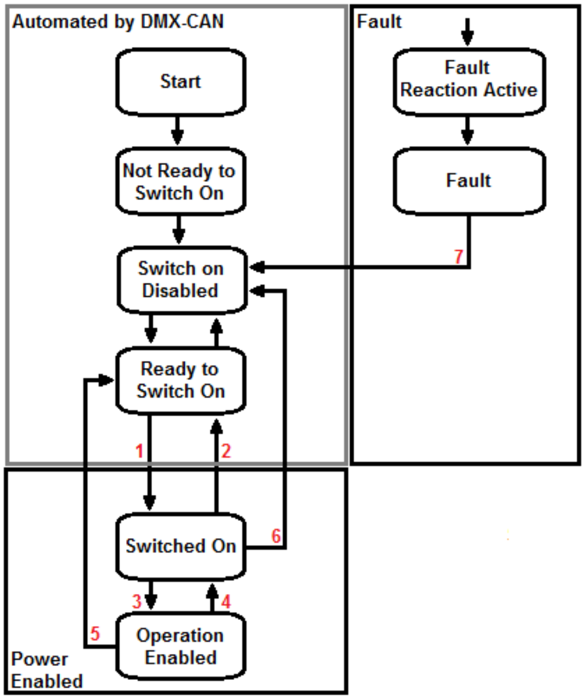


Figure 8.3

The DSP-402 state machine changes states by writing to the Controlword. The Statusword returns the current state of the DSP-402 state machine.

Transition 1: Ready to Switch On → Switched On

The "Switch On" control bit (Controlword bit 0) has been activated. The motor has been enabled.

Transition 2: Switched On→Ready to Switch On

The "Switch On" control bit (Controlword bit 0) has been deactivated. The motor has been disabled.

Transition 3: Switch On → Operation Enabled

The "Enable Operation" control bit (Controlword bit 3) has been activated. The DMX-CAN enters the specified Mode of Operation.

Transition 4: Operation Enabled → Switched On

The "Enable Operation" control bit (Controlword bit 3) has been deactivated. The DMX-CAN exits the current Mode of Operation. The motor remains enabled.

Transition 5: Operation Enabled → Ready to Switch On

The "Enable Operation" and "Switch On" control bits (Controlword bits 0 and 3) have been deactivated. The DMX-CAN exits the current Mode of Operation and the motor is disabled.

Transition 6: Switched On → Switch On Disabled

The "Disable Voltage" control bit (Controlword bit 1) has been activated. The motor is disabled.

Transition 6: Fault → Switch On Disabled

The "Reset Fault" control bit (Control word bit 7) has been activated. The DMX-CAN recovers from the current fault and prepares the state machine for further operation.

All other transitions of the DSP-402 state machine are automated by DMX-CAN.

The DMX-CAN has three modes of operation available that can be used in the Operation Enabled state.

Homing Mode allows the DMX-CAN to perform various homing routines to find the desired home position. Homing routines can be done using the home inputs, limit inputs, or the z-index of the encoder.

By using the Profile Velocity Mode, the DMX-CAN can move in either direction when given a target velocity.

The Profile Position Mode allows the DMX-CAN to move to desired target positions. Various speeds and motion profiles can be used when performing this movement.

Object 6040_h: Controlword

The Controlword is used to control all aspects and operations of the DSP-402 library including controlling the DSP-402 state machine, entering modes of operation, and moving or stopping the motor.

Object Description

Index	Name	Data Type	Value Range	Access
6040_h	Controlword	Unsigned 16	0 - 65535	Read/Write

Table 8.41

See table 8.44 to see the bit description of the Controlword.

Bit	Name
0	Switch On
1	Disable Voltage
2	Quick Stop
3	Enable Operation
4	Operation Mode Specific
5	Operation Mode Specific
6	Operation Mode Specific
7	Reset Fault
8	Halt

Table 8.42

Bit 8 of the Controlword, the Halt bit, will start and stop motion for all operation modes.

Bits 4, 5, and 6 of the Controlword perform different behaviors specific to the current Operation Mode. See table 8.45 for the description of these bits.

Bit	Homing Mode	Profile Velocity Mode	Profile Position Mode
4	Homing Operation Start	Not Used	new_set_point
5	Not Used	Not Used	change_set_immediately
6	Not Used	Not Used	0: Absolute 1: Incremental

Table 8.43

Object 6041$_h$: Statusword

The Statusword returns the current status of the DMX-CAN.

Object Description

Index	Name	Data Type	Value Range	Access
6041$_h$	Statusword	Unsigned 16	0 - 65535	Read Only

Table 8.44

See table 8.46 below for the bit description of the Statusword.

Bit	Name
0	Ready to Switch On
1	Switched On
2	Operation Enabled
3	Fault
4	Voltage Disabled
5	Quick Stop
6	Switch On Disabled
7	Warning
8	N/A
9	Remote

10	Target Reached
11	Internal Limit Active
12	Operation Mode Specific
13	Operation Mode Specific

Table 8.45

Object 6060$_h$: Modes of Operation

The Modes of Operation changes the desired mode of operation. The DMX-CAN supports the *Homing Mode*, *Profile Velocity Mode*, and the *Profile Position Mode*. For more details regarding these modes please see their respective sections.

Object Description

Index	Name	Data Type	Value Range	Access
6060$_h$	Modes of Operation	Signed 8	-128 - 127	Write Only

Table 8.46

See table 8.49 for the description of the Modes of Operation object. Also note that the DMX-CAN initiates and terminates a mode of operation via the Controlword object (6040$_h$).

Value	Mode of Operation
0	Reserved
1	Profile Position Mode
2	N/A
3	Profile Velocity Mode
4	N/A
5	Reserved
6	Homing Mode

Table 8.47

Object 6061$_h$: Modes of Operation Display

This object returns the current mode of operation of the DMX-CAN. A mode of operation is initiated and terminated by the Controlword object (6040$_h$).

Object Description

Index	Name	Data Type	Value Range	Access
6061$_h$	Modes of Operation Display	Signed 8	-128 - 127	Read Only

Table 8.48

See table 8.49 for details on the Mode of Operation Display object.

Object 607E$_h$: Polarity

The polarity object multiplies the position and/or velocity depending on the polarity value.

Object Description

Index	Name	Data Type	Value Range	Access
607E$_h$	Polarity	Unsigned 8	0 - 255	Read/Write

Table 8.49

Bit	Description
0...5	N/A
6	Velocity polarity 0: multiply by 1 1: multiply by -1
7	Position polarity 0: multiply by 1 1: multiply by -1

Table 8.50

Object 60FD$_h$: Digital Inputs

This object returns the current status of all limit, home, and general purpose digital inputs.

Object Description

Index	Name	Data Type	Value Range	Access
60FD$_h$	Digital Inputs	Unsigned 32	0 - (2^{32}-1)	Read Only

Table 8.51

See table 8.54 for the bit description of object 60FD$_h$.

Bit	Description
0	-Limit
1	+Limit
2	Home
3 - 15	Reserved
16	Digital Input 1
17	Digital Input 2
18	Digital Input 3
19	Digital Input 4
20	Digital Input 5
21	Digital Input 6

Table 8.52

Object 60FE$_h$: Digital Outputs

This object give read and write access to all available digital outputs.

Object Description

Index	Name	Data Type	Value Range	Access
$60FE_h$	Digital Outputs	Unsigned 32	$0 - (2^{32}-1)$	Read/Write

Table 8.53

See table 8.56 for the bit description for object $60FE_h$.

Bit	Description
0 - 15	Reserved
16	Digital Output 1
17	Digital Output 2
18	Digital Output 3

Table 8.54

Object 6402_h: Motor Type

This object returns the type of motor that is being driven by the controller.

Object Description

Index	Name	Data Type	Value Range	Access
6402_h	Motor Type	Unsigned 16	0 - 65535	Read Only

Table 8.55

Since the DMX-CAN is an integrated micro-step stepper motor solution, this object is read only and will return the value 768. See the table below for the value description.

Bit	Motor Type
0	Non-Standard Motor
1	Phase Modulated DC Motor
2	Frequency Controlled DC Motor
3	PM Synchronous Motor
4	FC Synchronous Motor
5	Switched Reluctance Motor
6	Wound Rotor Induction Motor
7	Squirrel Cage Induction Motor
8	Stepper Motor
9	Micro-Step Stepper Motor
10	Sinusoidal PM BL Motor
11	Trapezoidal PM BL Motor

Table 8.56

Object 6502_h: Supported Drive Modes

The DMX-CAN supports three modes of operation, the Profile Position Mode, Profile Velocity Mode, and Homing Mode. This information can be read using the Supported Drive Modes object.

Object Description

Index	Name	Data Type	Value Range	Access
6502$_h$	Supported Drive Modes	Unsigned 32	0 - (2^{32}-1)	Read Only

Table 8.57

This object is a read only constant and will always return 37. See table 8.60 for the bit description of this object.

Bit	Description
0	Profile Position Mode
1	Velocity Mode
2	Profile Velocity Mode
3	Profile Torque Mode
4	Reserved
5	Homing Mode
6	Interpolated Position Mode

Table 8.58

Object 607C$_h$: Home Offset

The home offset is the difference between the home switch position and the desired zero position. During homing, the motor will find the home switch location and then move to the zero position which is determined by the value of the home offset. The home offset is defined by pulses if running in open loop and encoder counts if running in StepNLoop mode.

Object Description

Index	Name	Data Type	Value Range	Access
607C$_h$	Home Offset	Signed 32	(-2^{31}) - (2^{31}-1)	Read/Write

Table 8.59

Object 6098$_h$: Homing Method

The DMX-CAN has several homing routines available. The homing method object determines which homing routine will be performed during homing.

Object Description

Index	Name	Data Type	Value Range	Access
6098$_h$	Homing Method	Signed 8	-128 - 127	Read/Write

Table 8.60

See the available homing methods below:

Method 7 & 11: Homing using the home input and z-index.

Method 7 and 11 will home the motor using the home input and z-index in the positive and negative directions, respectively. When the homing routine is started, the motor ramps to the Home Switch Speed until the home switch is triggered which, at that time, the motor decelerates to the Zero Position Speed. At Zero Position Speed the motor continues to move until the Z index channel is triggered and at that time the motor changes direction and moves by the defined home offset. The final position is marked as the home position.

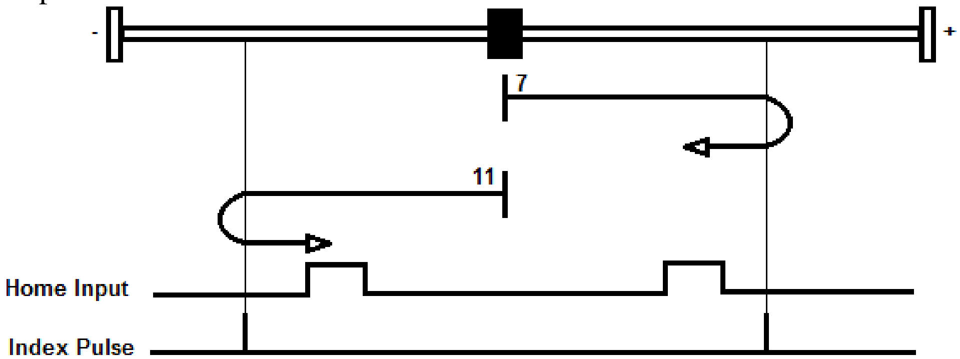


Figure 8.4

Method 17 & 18: Homing using the limit inputs only.

Method 17 and 18 will home the motor using the limit inputs only in the positive and negative direction, respectively. When the homing routine is started**,** the motor will accelerate to the Home Switch Speed. As soon as the limit input is triggered, the motor ramps down to the Zero Position Speed and moves away from the limit input triggered position by the home offset, depending on the home limit error correction amount. The final position at the end of the move will be referenced as zero.

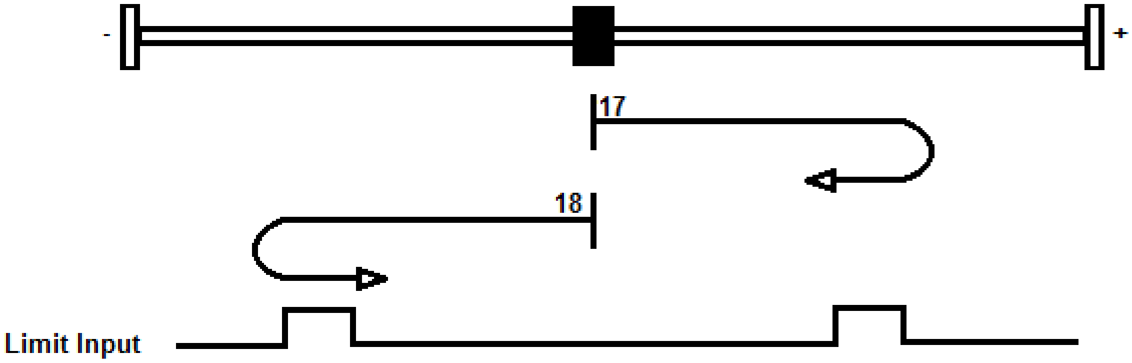


Figure 8.5

Method 19 & 20: Homing using the home input only.

Method 19 and 20 will home the motor using the home input only in the positive and negative direction, respectively. When the homing routine is started**,** the motor will ramp up to the Home Switch Speed. As soon as the home input is triggered, the motor ramps down to the Zero Position Speed and changes direction. When ramp down is done, the motor will move back by the Home Offset amount.

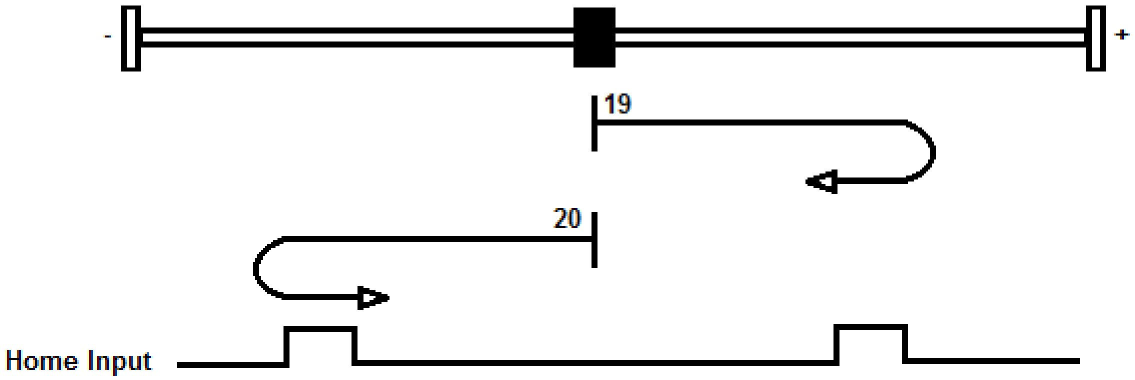


Figure 8.6

Method 33 & 34: Homing using the z-index only.

Methods 33 and 34 will home the motor using the z-index only in the positive and negative direction, respectively. Z-index channel pulse occurs once per revolution of the motor. When homing with only the z-index channel, only the Zero Position Speed is used. Once the z-index has been triggered, the motor will continue moving by the home offset amount.

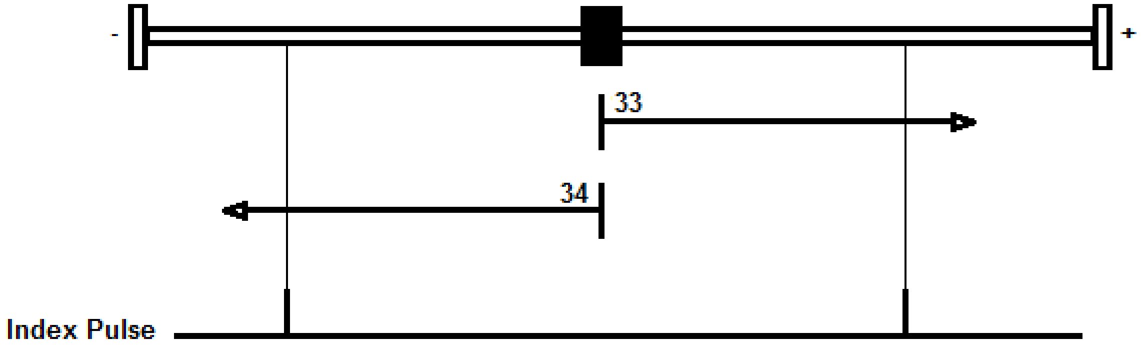


Figure 8.7

Method 35: Homing using the current position.

This method uses the current position of the DMX-CAN as the home position.

<u>Object 6099$_h$: Homing Speeds</u>

This object will define the speeds that are used when performing a homing routine. The home switch speed is the speed that will be used when the motor is searching for the switch. The zero position speed is the speed that will be used when the motor is moving from the home switch to the zero position (the distance between the home switch and the zero position is defined by object 607C$_h$).

If the motor is running in open-loop mode, all speed are in pulses per second. If StepNLoop is enabled, all speeds are in encoder counts per second.

Object Description

Index	Name	Data Type
6099_h	Homing Speeds	Array - Unsigned 32

Table 8.61

Sub Index Description

Sub Index	Name	Value Range	Access
00_h	Number of elements	2	Read Only

Table 8.62

Sub Index	Name	Description	Value Range	Access
01_h	Home Switch Speed	Speed used when searching for home switch	0 - (2^{32}-1)	Read/Write

Table 8.63

Sub Index	Name	Description	Value Range	Access
02_h	Zero Position Speed	Speed used when searching for zero position	0 - (2^{32}-1)	Read/Write

Table 8.64

Object 609A$_h$: Homing Acceleration

The homing acceleration defines the acceleration and deceleration that is used when the motor is starting or stopping its motion profile. Acceleration time is in milliseconds.

Object Description

Index	Name	Data Type	Value Range	Access
$609A_h$	Homing Acceleration	Unsigned 32	0 - (2^{32}-1)	Read/Write

Table 8.65

Object 606C$_h$: Velocity Actual Value

This object returns the current velocity of the motor. If the motor is running in open-loop mode, the velocity is in pulses per second. If the motor is running in StepNLoop mode, the velocity is in encoder counts per second.

Object Description

Index	Name	Data Type	Value Range	Access
$606C_h$	Velocity Actual Command	Signed 32	(-2^{31}) - (2^{31}-1)	Read Only

Table 8.66

Object 60FF$_h$: Target Velocity

This object defines the target velocity when the DMX-CAN is in Profile Velocity Mode. If the motor is in open-loop mode, the velocity is in pulses per second. If the motor is running in StepNLoop mode, the velocity is in encoder counts per second.

Object Description

Index	Name	Data Type	Value Range	Access
60FF$_h$	Target Velocity	Signed 32	(-2^{31}) - $(2^{31}-1)$	Read/Write

Table 8.67

Object 6064$_h$: Position Actual Value

This object will return the current position of the DMX-CAN. If the motor is in open-loop mode the position will be in pulses. If the motor is in StepNLoop mode, the position will be in encoder counts.

Object Description

Index	Name	Data Type	Value Range	Access
6064$_h$	Position Actual Value	Signed 32	(-2^{31}) - $(2^{31}-1)$	Read Only

Table 8.68

Object 607A$_h$: Target Position

This object defines the target position when the DMX-CAN is in Profile Position Mode. If the motor is in open-loop mode the position will be in pulses. If the motor is in StepNLoop mode, the position will be in encoder counts.

Object Description

Index	Name	Data Type	Value Range	Access
607A$_h$	Target Position	Signed 32	(-2^{31}) - $(2^{31}-1)$	Read/Write

Table 8.69

Object 6081$_h$: Profile Velocity

The profile velocity is the desired high speed when the DMX-CAN is in Profile Position Mode. The profile is achieved at the end of the acceleration ramp in the motion profile. If the motor is in open-loop mode, the velocity is in pulses per second. If the motor is running in StepNLoop mode, the velocity is in encoder counts per second.

Object Description

Index	Name	Data Type	Value Range	Access
6081$_h$	Profile Velocity	Unsigned 32	0 - $(2^{32}-1)$	Read/Write

Table 8.70

Object 6082$_h$: End Velocity

The end velocity is the desired low speed when the DMX-CAN is in Profile Position Mode. The end velocity is achieved at the end of a move, when the motor has reached the target position. If the motor is in open-loop mode, the velocity is in pulses per second. If the motor is running in StepNLoop mode, the velocity is in encoder counts per second.

Object Description

Index	Name	Data Type	Value Range	Access
6082$_h$	End Velocity	Unsigned 32	0 - (2^{32}-1)	Read/Write

Table 8.71

Object 6083$_h$: Profile Acceleration

The profile acceleration is the time the motor takes to accelerate to the profile velocity. This object is in milliseconds.

Object Description

Index	Name	Data Type	Value Range	Access
6083$_h$	Profile Acceleration	Unsigned 32	0 - (2^{31}-1)	Read/Write

Table 8.72

Object 6084$_h$: Profile Deceleration

The profile deceleration is the time the motor takes to decelerate from the profile velocity to the end velocity. This object is in milliseconds.

Object Description

Index	Name	Data Type	Value Range	Access
6084$_h$	Profile Deceleration	Unsigned 32	0 - (2^{31}-1)	Read/Write

Table 8.73

Object 6086$_h$: Motion Profile Type

The DMX-CAN has two available motion profile types, trapezoidal and sinusoidal. See the trapezoidal motion profile below:

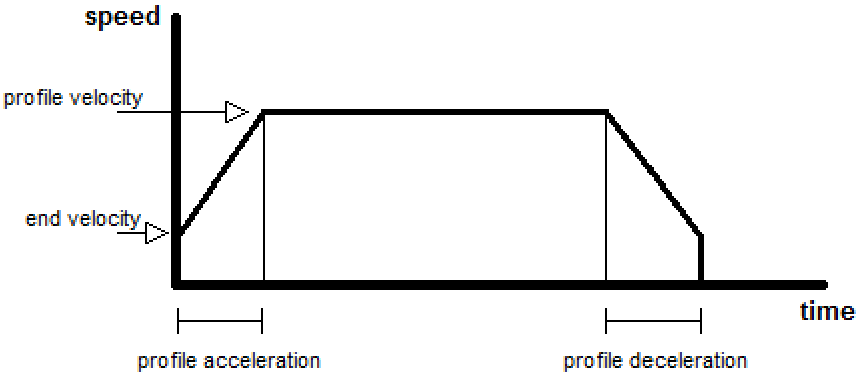


Figure 8.9

Figure 8.9 shows a sinusoidal motion profile.

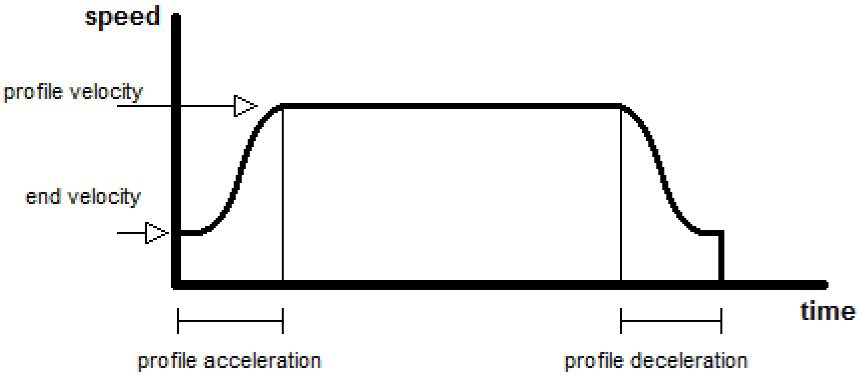


Figure 8.10

Object Description

Index	Name	Data Type	Value Range	Access
6086_h	Motion Profile Type	Signed 16	-32768 - 32767	Read/Write

Table 8.74

See table 8.77 for the motion profile code.

Value	Motion Profile
0	Trapezoidal Ramp
1	Sinusoidal Ramp

Table 8.75

Notes

Profile Velocity Mode

The Profile Velocity mode is used to move the DMX-CAN in either direction when given a target velocity. In this mode, moves are made in regards to speed and do not take the motor position into consideration

Profile Position Mode

The Profile Position Mode will move the DMX-CAN to various target positions. There are two ways to process new target position depending on the change_set_immediately bit in the Control word.

1. Single Set-point (change_set_immediately=0): The DMX-CAN will move to a given target position. Once the move is completed, it will indicate the move is complete and receive a new target position from the CANOpen master. The motor will ramp down to a stop after each move.
2. Set of Set-points (change_set_immediately=1): The DMX-CAN will move to a given target position. Upon completion, it will immediately process the next target position in the target position buffer. The motor will ramp down to a stop after each move.

The motion behavior for the Single Set-point mode is shown below:

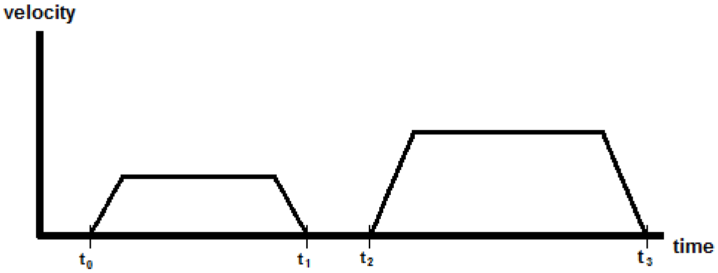


Figure 8.11

See below for the motion behavior of the Set of Set-points mode.

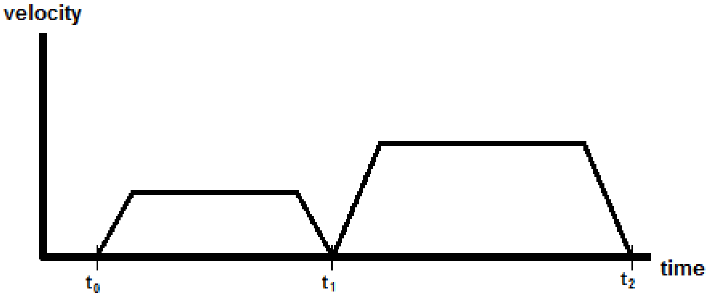


Figure 8.12

9. Communication Protocol [ASCII]

Communication protocol and commands are the same for both RS-232 and RS-485.

Sending Command to DMX-CAN
ASCII command string in the format of
@[DeviceName][ASCII Command][CR]

[CR] character has ASCII code 13.

Receiving Reply from DMX-CAN
The response will be in the format of
[Response][Null]

[Null] character has ASCII code 0.

Examples:
For querying the encoder position
Send: @01EX[CR]
Reply: 1000[Null]

For jogging the motor in positive direction
Send: @01J+[CR]
Reply: OK[Null]

For aborting any motion in progress
Send: @01ABORT[CR]
Reply: OK[Null]

Notes:

The address **'00'** is reserved for broadcasting over a RS-485 bus. Any ASCII command prefixed by **'@00'** will be processed by all DMX-CAN modules on the RS-485 bus. When a broadcast command is received by a DMX-CAN module, no response is sent back to the master.

There is an option to have the device name appended to the beginning of the response string. This can be achieved by using the **RT** command. See "Response Type Selection"

10. Communication Protocol [CANOpen]

Object Dictionary

The object dictionary stores all parameters of the DMX-CAN and is divided into different sections.

Index	Description
0x0001-0x009F	Data Type Definitions
0x00A0-0x0FFF	Reserved
0x1000-0x1FFF	Communication Profile Area (CiA 301)
0x2000-0x5FFF	Manufacturer Specific Profile Area
0x6000-0x9FFF	Standardized Profile Area (CiA 402)
0xA000-0xFFFF	Reserved

Table 10.0

The object dictionary of DMX-CAN has the following format.

Index	Sub-Index	Access	Name	Description
0x1000	None	Constant	Device Type	Returns the device profile number
0x1001	None	Read Only	Error Register	Returns an error code if applicable
0x1008	None	Read Only	Manufacturer Device Name	Return DMX-CAN device name – "DMC"
0x1018	0x01	Read Only	Manufacturer Number	N/A
	0x02	Read Only	Product Code	N/A
	0x03	Read Only	Revision Number	112
	0x04	Read Only	Serial Number	N/A
0x2000	None	Read/Write	Alarm/In Pos Enable	Enable/disable the alarm/in pos feature
0x2001	0x01	Read/Write	StepNLoop Enable	Enable/disable StepNLoop
	0x02	Read/Write	Max Attempt	Max number of correction attempts
	0x03	Read/Write	Run Tolerance	Minimum allowable error before correction while the motor is moving
	0x04	Read/Write	Idle Tolerance	Minimum allowable error before correction while the motor is idle
	0x05	Read/Write	Error Range	Maximum allowable error
0x2002	None	Read Only	StepNLoop Delta	Current delta error
0x2003	None	Read Only	StepNLoop Status	See table 8.14
0x2004	0x01	Read Only	Current	Motor current
	0x02	Read/Write	Run Current	Current used while motor is moving
	0x03	Read/Write	Idle Current	Current used while motor is idle
	0x04	Read/Write	Idle Time	Time required before switching to idle current
0x2005	None	Read/Write	Digital Output Boot	Digital output status on boot-up
0x2006	None	Read/Write	Enable Output Boot	Enable output status on boot-up
0x2007	None	Read/Write	Latch Enable	Enable/Disable latch operation
0x2008	None	Read/Write	Latch Status	Current latch status
0x2009	0x01	Read Only	Latch Pulse Position	The latched pulse position
	0x02	Read Only	Latch Encoder Position	The latched encoder position

0x200A	None	Read/Write	Communication Type	Specify communication type between RS-232, RS-485, and CANOpen. See table 8.34.
0x200B	None	Read/Write	Bit Rate	Specify CANOpen bit rate
0x200C	None	Read/Write	Node ID	Specify CANOpen node ID
0x200D	None	Read/Write	Clear Error	Clear a motor or StepNLoop error
0x200E	None	Read/Write	Perform Store	Store all non-volatile parameters
0x6040	None	Read/Write	Controlword	Control the CiA-DSP402 state machine. See table 8.42.
0x6041	None	Read Only	Statusword	Current status of the CiA-DSP402 state machine. See table 8.45.
0x6060	None	Write Only	Mode of Operation	Set mode of operation. See table 8.47
0x6061	None	Read Only	Mode of Operation Display	Current mode of operation. See table 8.47
0x6064	None	Read Only	Position Actual Value	Current motor position
0x606C	None	Read Only	Velocity Actual Value	Current motor velocity
0x607A	None	Read/Write	Target Position	Specify target position when in Profile Position Mode
0x607C	None	Read/Write	Home Offset	Distance between home trigger and zero position
0x607E	None	Read/Write	Polarity	Direction and target position polarity
0x6081	None	Read/Write	Profile Velocity	Target high speed of the motor
0x6082	None	Read/Write	End Velocity	Target low speed of the motor
0x6083	None	Read/Write	Profile acceleration	Target acceleration
0x6084	None	Read/Write	Profile deceleration	Target deceleration
0x6086	None	Read/Write	Motion Profile Type	Trapezoidal or sinusoidal motion profile
0x6098	None	Read/Write	Homing Method	Homing routine
0x6099	0x01	Read/Write	Home Switch Speed	Speed used when searching for home switch
	0x02	Read/Write	Home Zero Speed	Speed used when searching for zero position
0x609A	None	Read/Write	Home Acceleration	Acceleration/deceleration used during homing
0x60FD	None	Read Only	Digital Inputs	See table 8.52.
0x60FE	None	Read/Write	Digital Outputs	See table 8.54.
0x60FF	None	Read/Write	Target Velocity	Target velocity when in Profile Velocity Mode
0x6402	None	Read Only	Motor Type	See table 8.56
0x6502	None	Read Only	Supported Modes	See table 8.58.

Table 10.1

All data in the object dictionary will be accessed using Service Data Objects, called SDO's.

Service Data Objects

SDO's are used for read and write access to all entries within object dictionary. Each SDO transfer consists of a SDO client and a SDO server. The SDO client initiates the transfer and the SDO server responds to the request. DMX-CAN will only function as a SDO server. See below for the SDO data package.

Domain protocol	16 bit index	8 bit sub-index	4 bytes data

Figure 10.0

In order to successfully transfer a SDO request to the DMX-CAN, an appropriate COB-ID has to be transferred with each message. The following formula can be used to calculate the correct COB-IB:

COB-ID = 0x0600 + <Node ID>

DMX-CAN will respond to the SDO request with the appropriate data or confirmation along with the following COB-ID:

COB-ID = 0x0580 + <Node ID>

11. Language Specification [ASCII]

Important Note: All the commands described in this section are for ASCII communication over a RS-232/RS-485 network. For details on communication over CANOpen, please see the "Object Dictionary [CANOpen]" section.

DMX-CAN language is case sensitive. All command should be in capital letters. Invalid command is returned "?". Always check for proper reply when command is sent.

Command	Description	Return
ABORT	Immediately stops the motor if in motion. For decelerate stop, use STOP command. This command is used for clearing the StepNLoop error status	OK
ABS	Set move mode to absolute	OK
ACC	Returns current acceleration value in milliseconds.	Acceleration value in milliseconds
ACC=[Value]	Sets acceleration value in milliseconds.	OK
CLR	Clears limit and StepNLoop error	OK
BR	Return bit rate setting (CANopen)	1,2,3,4,5,6,7,8
BR=[Value]	Set bit rate Range: 1 – 10000 2 – 20000 3 – 50000 4 – 100000 5 – 125000 6 – 250000 7 – 500000 8 – 800000 9 – 1000000	OK
CLR	Clears limit error and StepNLoop error	OK
CM	Get RS-232/RS-485/CANopen communication mode	0-2
CM=[0-2]	Set RS-232/RS-485/CANopen communication mode 0 – RS-232 1 – RS-485 2 – CANopen	OK
CUR	Get real-time current	0mA, 100mA to 2500mA
CURI	Get idle current setting	0mA, 100mA to 2500mA
CURI=[Value]	Set idle current. To have motor become disabled during idle state, set this value to 0.	OK
CURR	Get run current setting	0mA, 100mA to 2500mA
CURR=[Value]	Set run current	OK
CURT	Get driver idle time setting	milliseconds
CURT=[Value]	Set driver idle time setting	OK
DB	Return baud rate setting (RS-232/RS-485)	1,2,3,4,5
DB=[Value]	Set baud rate. Range: 1 – 9600 bps 2 – 19200 bps 3 – 38400 bps 4 – 57600 bps 5 – 115200 bps	OK

DEC	Get deceleration value in milliseconds. Only used if EDEC=1	Milli-seconds
DEC=[Value]	Set deceleration value in milliseconds. Only used if EDEC=1	OK
DI	Return status of digital inputs	6-bit number in decimal.
DI[1-6]	Return status of individual input	1 or 0
DN	Return device name	DMK00-DMK99
DN=[Device ID]	Set device name. Range: DMC01 to DMC99 Note: Address 00 is reserved for broadcasting over RS-485	OK
DO	Return status of digital outputs	3-bit number
DO=[Value]	Set digital output 3 bit number. Note that DO1 and DO2 can only be used as general purpose when StepNLoop is disabled	OK
DO[1-3]	Return status of individual digital output	1 or 0
DO[1-3] = [Value]	Set individual digital output	OK
DOBOOT	Get DO boot-up state	See Table 7.1
DOBOOT=[Value]	Set DO boot-up state	OK
DX	Returns the delta value during StepNLoop control	32-bit number
EDEC	Get unique deceleration enable	0 or 1
EDEC=[Value]	Set unique deceleration enable	OK
EDO	Returns enable alarm/in pos mode status	1 – enabled 0 – disabled
EDO=[0 or 1]	Enables (value 1) or disable (value 0) alarm/in pos mode	OK
EO	Returns driver power enable status.	1 – Motor power enabled 0 – Motor power disabled
EO=[0 or 1]	Enables (value 1) or disable (value 0) motor power.	OK
EOBOOT	Get EO boot-up state	0 or 1
EOBOOT=[Value]	Set EO boot-up state	OK
EX	Returns current encoder counter value	32-bit number
EX=[Value]	Sets the current encoder counter value	OK
H+	Homes the motor in positive direction	OK
H-	Homes the motor in negative direction	OK
HL+	Homes the motor in positive direction (with with low speed)	OK
HL-	Homes the motor in negative direction (with low speed)	OK
HSPD	Returns High Speed Setting	Value in PPS
HSPD=[Value]	Sets High Speed.	OK
ID	Returns product ID	DriveMax-CAN
INC	Set move mode to incremental	OK
J+	Jogs the motor in positive direction	OK
J-	Jogs the motor in negative direction	OK
L+	Homes the motor in the positive direction using the limit switch.	OK
L-	Homes the motor in the negative direction using the limit switch.	OK
LCA	Return the limit correction amount	32-bit number
LCA=[Value]	Set the limit correction amount	OK
LSPD	Returns Low Speed Setting	Value in PPS
LSPD=[Value]	Sets Low Speed	OK
LT=[0 or 1]	Enable or disable position latch feature	OK
LTE	Returns latched encoder position	32-bit number
LTP	Returns latched pulse position	32-bit number

LTS	Returns latch status.	0 – Latch off 1 – Latch on and waiting for latch trigger 2 – Latch triggered
MM	Get move mode status	0 – Absolute move mode 1 – Incremental move mode
MST	Returns motor status	Bit 0 – constant speed Bit 1 – accelerating Bit 2 – decelerating Bit 3 – home input status Bit 4 – -limit input status Bit 5 – +limit input status Bit 6 – minus limit error Bit 7 – plus limit error Bit 8 – latch input status Bit 9 – Z encoder channel
NID	Return node ID (CANopen)	1-127
NID=[Value]	Set node ID (CANopen) Range: 1 – 127	OK
POL	Returns current polarity	Bit 0 – don't care Bit 1 – Dir Bit 2 – don't care Bit 3 – don't care Bit 4 – Limit Bit 5 – Home Bit 6 – Latch Bit 7 – In Position Output Bit 8 – Alarm Output
POL=[value]	Sets polarity.	OK
PS	Returns current pulse speed	Value in PPS
PX	Returns current position value	Position value in 32 bit
PX=[value]	Sets the current position value	OK
RT	Get response type parameter	0 or 1
RT=[0 or 1]	Set response type parameter	0 – Do not append device name to response string 1 – Append device name to response string
SCV	Returns the s-curve accel/decal control	0 or 1
SCV=[0 or 1]	Enable or disable s-curve. If disabled, trapezoidal acceleration/ deceleration will be used.	OK
SL	Returns StepNLoop control status	0 – StepNLoop Off 1 – StepNLoop On
SL=[0 or 1]	Enable or disable StepNLoop Control	OK
SLA	Returns maximum number of StepNLoop control attempt	32-bit
SLA=[value]	Sets maximum number of StepNLoop control attempt	OK
SLE	Returns StepNLoop correction value.	32-bit
SLE=[value]	Sets StepNLoop correction value.	OK
SLS	Returns current status of StepNLoop control	0 – Idle 1 – Moving 2 – Correcting 3 – Stopping 4 – Aborting 5 – Jogging

		6 – Homing 7 – Z Homing 8 – Correction Range Err 9 – Correction Attempt Err 10 – Stall Err 11 – Limit Err
SLT	Returns StepNLoop tolerance value	32-bit
SLT=[value]	Sets StepNLoop tolerance value.	OK
STORE	Store values to flash memory.	**OK**
SSPD[value]	Set on-the-fly speed change. In order to use this command, S-curve control must be disabled. Use SCV command to disable s-curve control.	**OK**
SSPDM	Return on-the-fly speed change mode	1,2,3,4,5,6,7,8
SSPDM=[value]	Set on-the-fly speed change mode	OK
STOP	Decelerated to stop the motor if in motion. For immediate stop, use ABORT command	OK
VER	Returns current firmware software version number	VXXX
X[value]	Moves the motor to absolute position value using the HSPD, LSPD, and ACC values. Maximum allowed incremental move amount is 262143. For example, if current position is 100000, target move must be between 362143 and -162143	OK
Z+	Homes the motor in positive direction using the Z index encoder channel ONLY.	OK
Z-	Homes the motor in negative direction using the Z index encoder channel ONLY.	OK
ZH+	Homes the motor in positive direction using the home switch and then Z index encoder channel.	OK
ZH-	Homes the motor in negative direction using the home switch and then Z index encoder channel.	OK

Table 11.0

Error Codes

If an ASCII command cannot be processed by the DMX-CAN, the controller will reply with an error code. See below for possible error responses:

Error Code	Description
?[Command]	The ASCII command is not understood by the controller
?Bad SSPD Command	SSPD move parameter is invalid
?Enable RS-485 Mode	Modbus mode cannot be set because communication needs to first be set to RS-485.
?Index out of Range	The index for the command sent to the controller is not valid.
?Motor is not moving	T[] command is invalid because a target position move is not in operation
?Moving	A move or position change command is sent while the controller is outputting pulses.
?SCV ON	Cannot perform SSPD move because s-curve is enabled
?Speed out of range	SSPD move parameter is out of the range of the SSPDM speed window.
?State Error	A move command is issued while the controller is in error state.

Table 11.1

Appendix A: Speed Settings

HSPD value [PPS] †	Speed Window [SSPDM]	Min. LSPD value	Min. ACC [ms]	δ	Max ACC setting [ms]
1 - 16 K	0,1	10	2	500	((HSPD – LSPD) / δ) × 1000
16K - 30 K	2	10	1	1 K	
30K - 80 K	3	15	1	2 K	
80K - 160 K	4	25	1	4 K	
160K - 300 K	5	50	1	8 K	
300K - 800 K	6	100	1	18 K	
800K - 1.6 M	7	200	1	39 K	
1.6 M - 3.0 M	8	400	1	68 K	
3.0 M – 6.0 M	9	500	1	135 K	

Table A.0

†If StepNLoop is enabled, the [HSPD range] values needs to be transposed from PPS (pulse/sec) to EPS (encoder counts/sec) using the following formula:

EPS = PPS / Step-N-Loop Ratio

Acceleration/Deceleration Range

The allowable acceleration/deceleration values depend on the **LSPD** and **HSPD** settings. The minimum acceleration/deceleration setting for a given high speed and low speed is shown in Table A.0.

The maximum acceleration/deceleration setting for a given high speed and low speed can be calculated using the formula:

Note: The ACC parameter will be automatically adjusted if the value exceeds the allowable range.

Max ACC = ((HSPD – LSPD) / δ) × 1000 [ms]

Figure A.0

Examples:

a) If **HSPD** = 20,000 pps, **LSPD** = 100 pps:
   a. Min acceleration allowable: **1 ms**
   b. Max acceleration allowable:
      ((20,000 – 100) / 1,000) x 1,000 ms = **19900 ms** (19.9 sec)

b) If **HSPD** = 900,000 pps, **LSPD** = 1000 pps:
   a. Min acceleration allowable: **1 ms**
   b. Max acceleration allowable:
      ((900,000 – 1000) / 39,000) x 1000 ms = **23050** ms (23.05 sec)

Contact Information

Arcus Technology, Inc.

3159 Independence Drive
Livermore, CA 94551
925-373-8800

www.arcus-technology.com