DMX-K-SA
Integrated Step Motor Encoder/Driver/Controller Manual

Smart Motion Control Co.,Ltd.

Http://www.smmc.com.tw ;E-mail :smmc@seed.net.tw

+886-3-346-1082 FAX:+886-3-3753368

First edition, April 2007

Revision History:

2.0 – First revision
2.1 – Added RS-485 detailed transmission info
2.2 – Fixed pulse/rev feature
3.01 – Added RS-232 communication
Added Sync Output feature
3.02 – Added baud rate support
3.03 – Removed DO LOOPWHILE, added V commands
3.04 – Added limit error correction feature
3.05 – Simplified polarity
3.06 – Added alarm, in position polarity, broadcast address
3.07 – Added SSPDM, home limit correction, current commands

Firmware Compatibility:

v311

Software Compatibility:

DMX-K v311

Table of Contents

1. Introduction

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

DMX K Series has the following features:

- RS-485 or RS-232 ASCII communication with 9600, 19200, 38400, 57600, 115200 Baud rate
- Standalone programming using easy to use text based programming language
- Opto-isolated +Limit/-Limit/Home inputs
- Opto-isolated 6 digital inputs
- Opto-isolated 3 digital outputs
- High speed position capture Digital Input
- Position Synchronized Digital Output
- 1000 line incremental encoder (4000 counts/rev with 4x quadrature decoding)
- StepNLoop closed-loop control
- S-curve/Trapezoidal acceleration profile control
- Homing routine using:
    - Home input only
    - Z index encoder channel only
    - Home input and Z index encoder channel
- 16 micro-step driver with effective resolution with 1.8 degree motor of 3200 pulse/rev
- 12 to 35VDC Voltage Input
- Driver current from 100mA to 2.5A
- NEMA 17 motor sizes available in different stack sizes.
  (NEMA 23 size to be released by 3rd quarter of 2007)

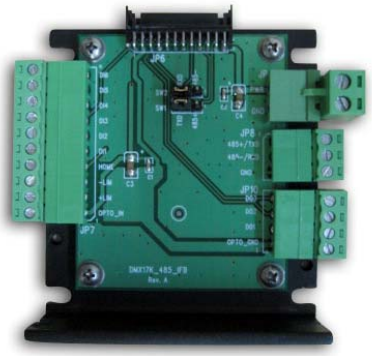
Model Numbers

Main Product

DMX-K-SA-17-[ ]

- [ ] Motor Stack Size
  - **2** – Double
  - **3** – Triple
  - **4** - Quad
- **17** – NEMA 17 Motor
- **SA** - Standalone
- **K** – K Series

Accessories



Part ID: DMX-K-SA-IFB
Description: Interface Board for the DMX-K-SA.

Part ID: DMX-K-24FF3
Description: 24 pin female to female cable 3 ft long

Default Settings

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

Communication Method	RS-232
Baud Rate	9600
Device ID	DMK01
Idle Current	1000 mA
Run Current	1600 mA
Idle Time	500 mSec
StepNLoop	Enabled
StepNLoop Maximum Attempt	10
StepNLoop Tolerance Range	20
StepNLoop Error Range	1000
Direction Polarity	CW
Limit Polarity	Active Low
Home Polarity	Active Low
Latch Polarity	Active Low
Limit Move Out Amount (for H+ or H- command only)	1000
Run Program on Power Up	Disabled

Contacting Support

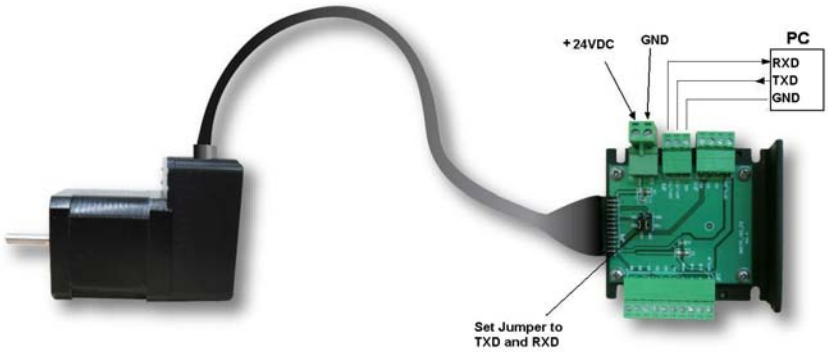
For any technical support, contact by email at support@arcus-technology.com or by phone 1-510-661-9100 from 9 AM to 5 PM PST (California, U.S.A.).

2. Quick Startup Guide

If you are a first time user and want to have the DMX-K unit up and running quickly, follow the recommended steps:

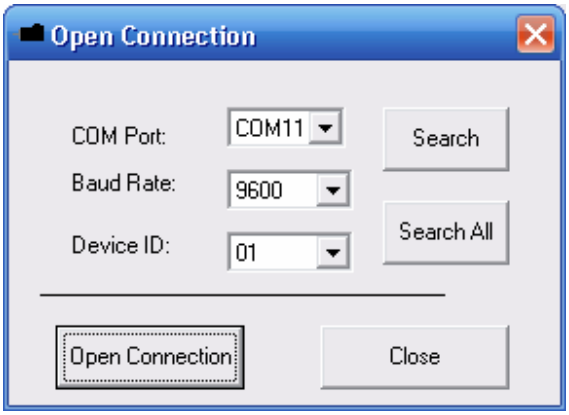
1) Prepare a Windows XP compatible PC with RS-232 communication port. If the PC does not have RS-232 communication port, USB to RS-232 converter is needed.

2) Connect DMX-K-SA to the Interface board using 24-pin cable. Connect RS-232 communication lines between the Junction board and PC. Supply +24V power and GND.



3) Download DMX-K Windows program from Arcus Technology website:
www.arcus-technology.com/support

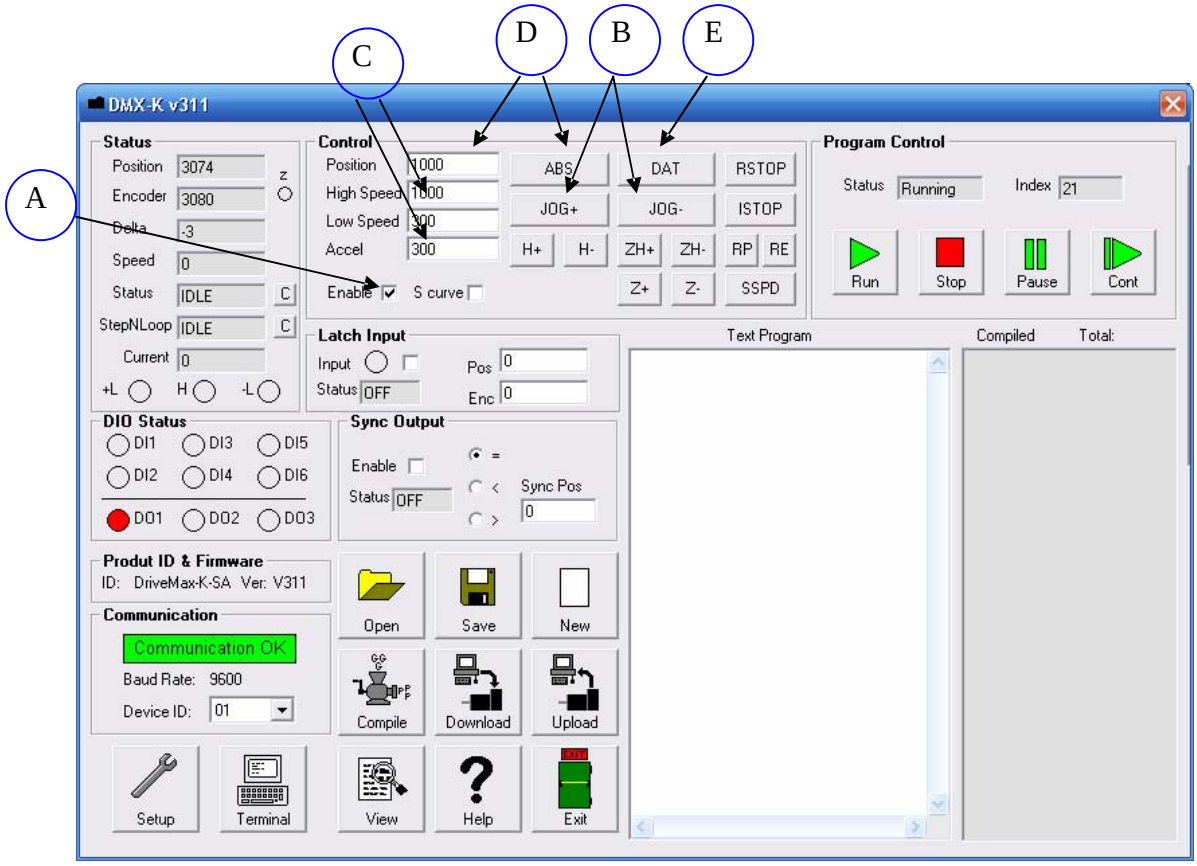
4) When DMX-K Windows program is started, you will see following dialog box:



Click on the Search button on the upper right corner. This search will check all available communication ports from COM1 to COM12 and check for communication with the DMX-K device. Note that the default baud rate and device name is **9600 bps**, and **01** respectively.

Click on the Search All button on the middle right corner if the DMX-K is configured for RS-485. RS-485 is a multi-drop protocol which allows many devices to be connected on a bus. This routine will search for all devices available on COM1 to COM12. Note that this routine does not apply if the DMX-K is configured for the point-to-point protocol: RS-232.

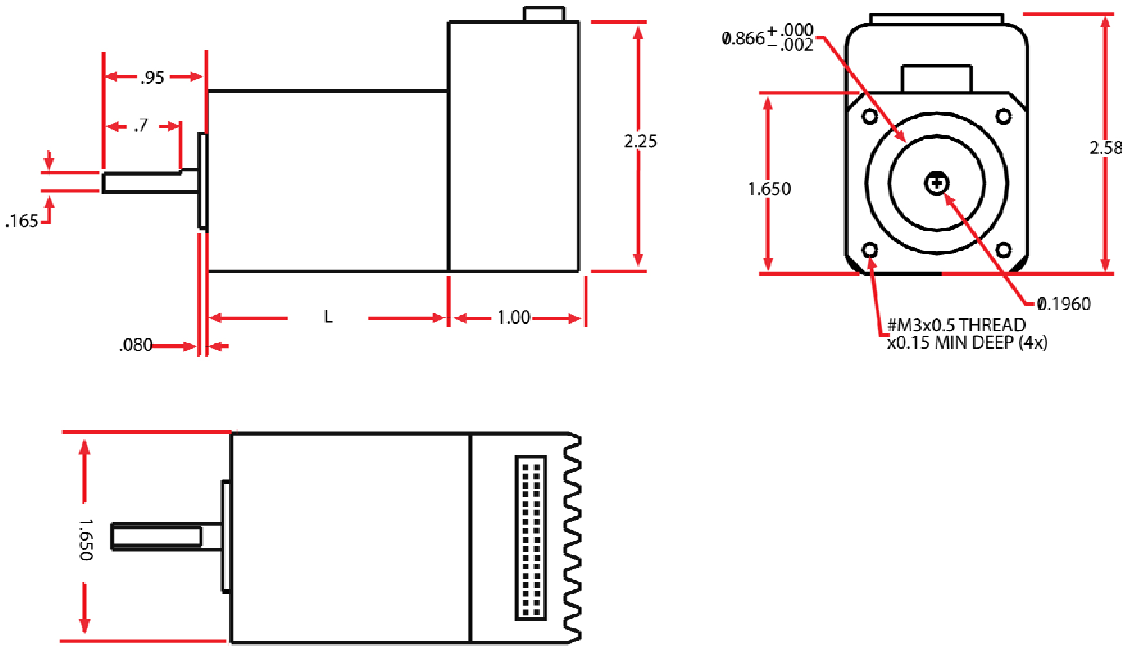
5) Once the desired device is found, the right port number will be selected and you can click on Open Connection button to start communicating with the DMX-K. The following screen will show:



A. To energize the motor, check the enable checkbox.
B. To move the motor, try jogging plus or minus.
C. Change speed and acceleration values to see moves at different speed.
D. To move to position, enter the target position and perform move ABS.
E. To move back to zero position, push DATUM button.

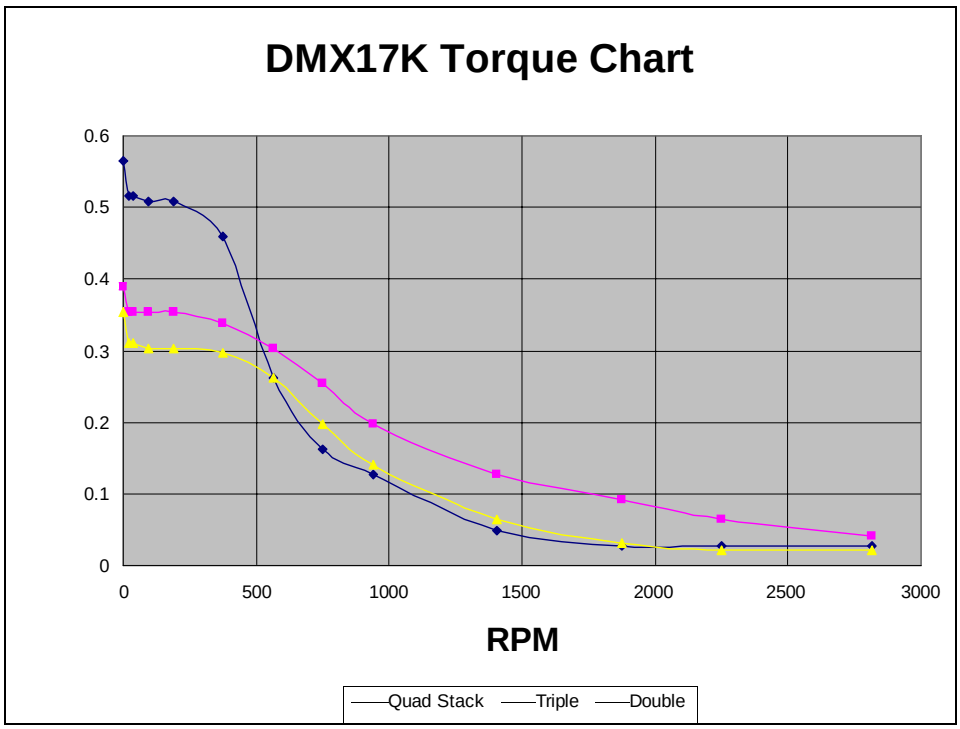
For detailed information on using the DMX-K Windows program, please see section 9.

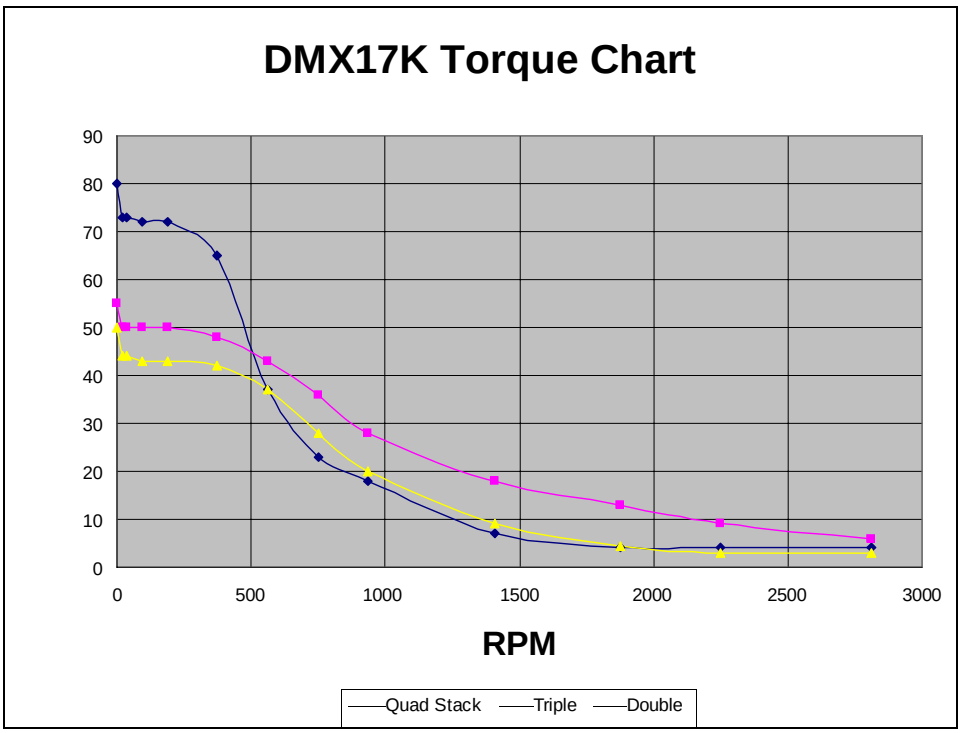
3. Dimensions and Motor Spec



NEMA 17 Models	L (inches)	Motor Inertia (oz-in^2)	Amp/Phase (amp)
DMX17K-SA-2	1.58	0.28	1.7
DMX17K-SA-3	1.89	0.37	2.0
DMX17K-SA-4	2.36	0.56	2.0

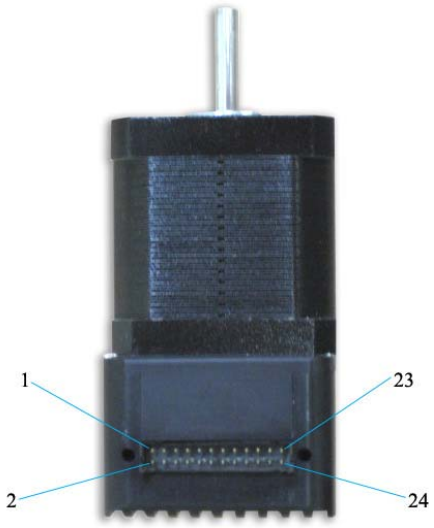
4. Torque Curve





5. Connector

DMX-K Series come with 24 pin connector as shown below.



Connector Information

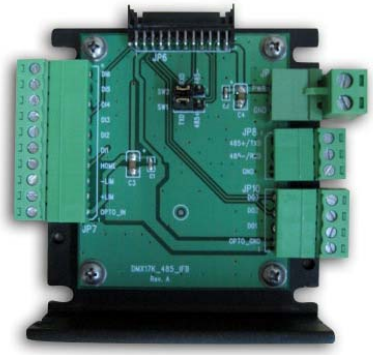
Description: Female 24 pin 2mm dual row
Manufacturer: HIROSE
Part Number: DF11-24DS-2C (24 pin female connector)
DF11-2428SC (female pin)

Connector Pin Outs

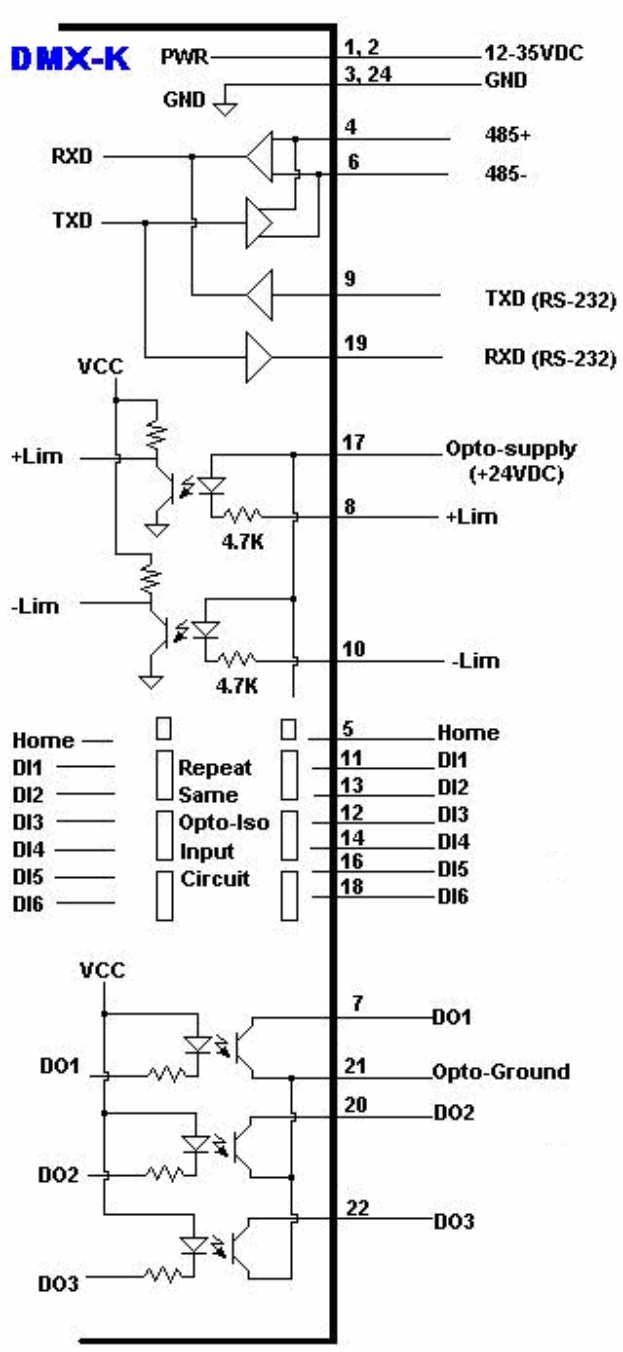
Description	Pin	Pin	Description
PWR	1	2	PWR
GND	3	4	485+
HOME	5	6	485-
DO1	7	8	+LIM
RS-232 TXD	9	10	-LIM
DI1	11	12	DI3
DI2	13	14	DI4
NC	15	16	DI5
OPTO SUPPLY	17	18	DI6
RS-232 RXD	19	20	DO2
OPTO GND	21	22	DO3
NC	23	24	GND

Junction Board

24 pin connector junction board which breaks out the 24 pin signals to screw terminals shown below is available for quick wiring and testing.



Interface Circuit



Note: All the digital inputs (Limits, Home, DI1 to DI6) have same opto-isolated circuitry)

Digital Input Connections

Limit, Home and Digital Inputs 1 to 6 are opto-isolated inputs.

Connect the Opto-Supply using 12-24VDC input. To trigger the Limit, Home, or Digital Inputs, sink the line to ground of the Opto-supply.

DI2 is used also as a high speed position capture digital input.

Digital Output Connections

Digital outputs are opto-isolated outputs using a Darlington transistor that can sink up to 100mA of current at maximum voltage of 24VDC.

DMX-K has 3 general purpose digital outputs.

When using StepNLoop closed loop control, DO1 is used as In-position output and DO3 is used as Alarm Output.

When using Sync Output feature, DO2 is used as the Sync Output signal.

6. Electrical Specifications

Power Requirement

Regulated Supply Voltage Range:	**+12 to +35 VDC**
Recommended Current for power supply:	**2.5 A**

Important Note:
Higher voltage recommended for high speed application

Communication Interface:

RS-232 (default)	**9600, 19200, 38400, 57600, 115K Baud RS-232 Arcus ASCII command support**
RS-485:	**9600, 19200, 38400, 57600, 115K Baud RS-485 Arcus ASCII command support**

Important Note:
Factory default setting is RS-232 at 9600 bps
To use RS-485, communicate first using RS-232,
change communication mode to RS-485, store to memory and
reboot the power and use RS-485 communication.

+Lim, -Lim, Home, Digital Inputs:

Type:	**Opto-isolated inputs**
Opto voltage supply input:	**+12 to +24 VDC**

Digital Outputs:

Type:	**Darlington Opto-isolated output**
Max sink current at 24VDC	**100 mA**

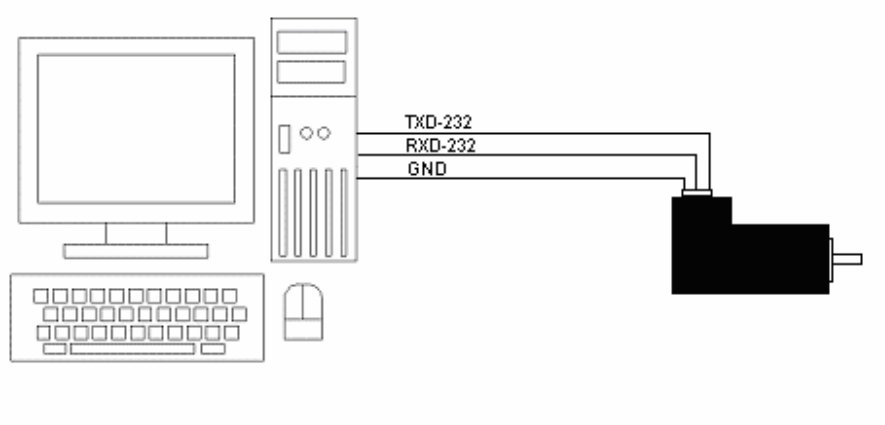
7. Communication

There are two ways to communicate with DMX-K series product: RS-232 and RS-485.

RS-232

When the DMX-K 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:

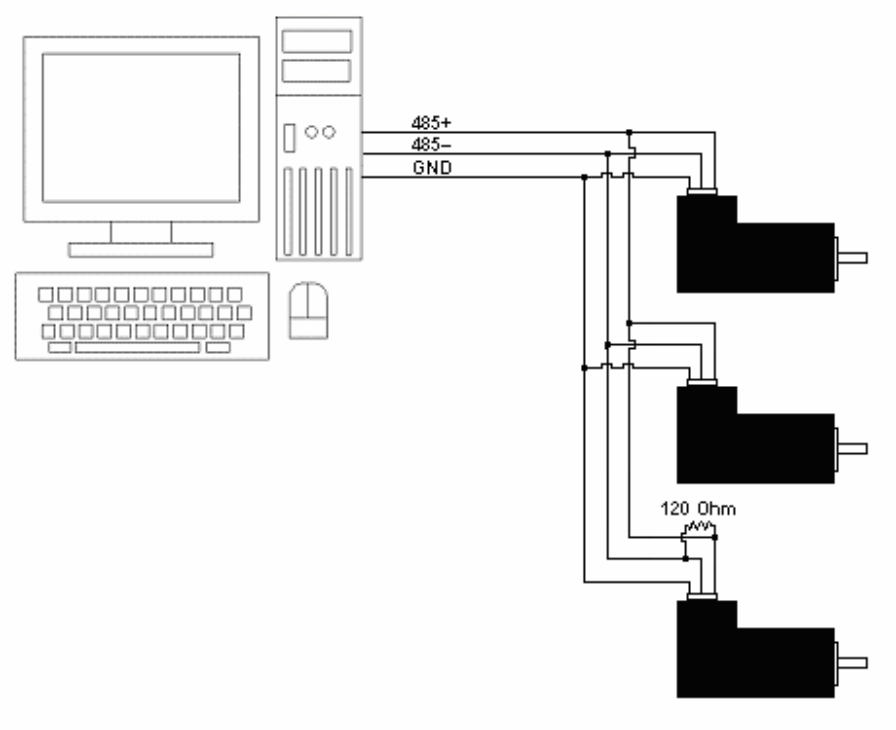


RS-485

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.



Communication Protocol

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

Sending Command to DMX-K

ASCII command string in the format of

@[DeviceName][ASCII Command][CR]

[CR] character has ASCII code 13.

Receiving Reply from DMX-K

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]

Note:

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

ASCII Commands

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
ACC	Returns current acceleration value in milliseconds.	Acceleration value in milliseconds
ACC=[Value]	Sets acceleration value in milliseconds	OK
CLR	Clears limit error	OK
CLRS	Clears StepNLoop error	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
DB	Return baud rate setting	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
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: DMK01 to DMK99 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
DX	Returns the delta value during StepNLoop control	32-bit number
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
EX	Returns current encoder counter value	32-bit number
EX=[Value]	Sets the current encoder counter value	OK
HSPD	Returns High Speed Setting	Value in PPS
HSPD=[Value]	Sets High Speed	OK
H+	Homes the motor in positive direction	OK
H-	Homes the motor in negative direction	OK
ID	Returns product ID	DriveMax-K-SA
J+	Jogs the motor in positive direction	OK
J-	Jogs the motor in negative direction	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
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
POL	Returns current polarity	Bit 0 – Bit 1 – Dir Bit 2 – Bit 3 – 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
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 Error. 9 – Correction Attempt Error. 10 – Stall Error 11 – Limit Error
SLT	Returns StepNLoop tolerance value	32-bit
SLT=[value]	Sets StepNLoop tolerance value	OK
STORE	Store values to flash memory	OK
SLOAD	Returns RunOnBoot parameter	
SLOAD=[0 or 1]	0 – Do NOT run standalone program on boot up 1 – Run standalone program on boot up	
SR=[Value]	Control standalone program: 0 – Stop standalone program 1 – Run standalone program 2 – Pause standalone program 3 – Continue standalone program	
SPC	Get program counter for standalone program	
SASTAT	Get standalone program status 0 – Stopped 1 – Running 2 – Paused 4 – In Error	
SA[LineNumber]	Get standalone line LineNumber: [0-1275]	
SA[LineNumber]=[Value]	Set standalone line LineNumber: [0-1275]	
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
V[VarNumber]	Get standalone variable value VarNumber: [0-99]	
V[VarNumber]=[Value]	Write standalone variable value VarNumber:[0-99]	
VER	Returns current firmware software version number	OK
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

8. Feature Overview

Built-in encoder

DMX-K 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 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

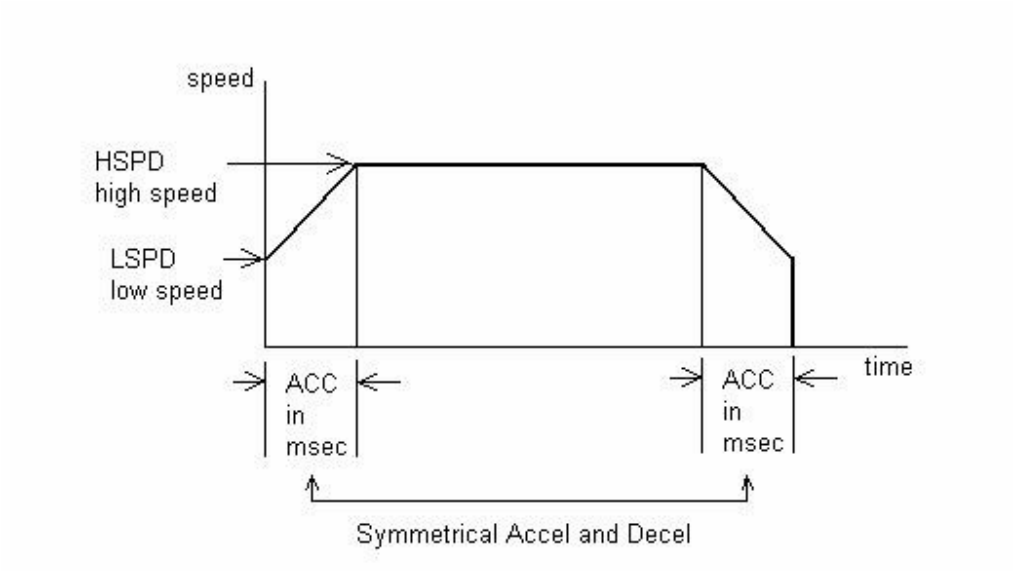
When StepNLoop closed-loop control is *disabled*:
EX command returns encoder position
PX command returns pulse position

Built-in Microstep Driver

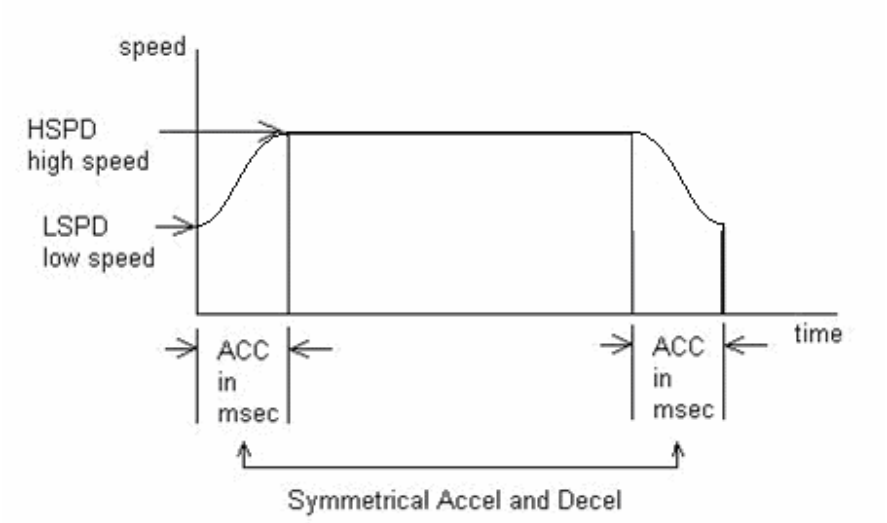
DMX-K has an integrated micro-step driver. The micro-step setting is fixed at 16. With a 1.8° motor, this results in a 3200 step/rev resolution.

Motion Profile

By default, DriveMax incorporates trapezoidal velocity profile as shown below.



S-curve velocity profile is shown below.



S-Curve reduces the jerk resulting in reduced vibration and wear on the mechanical system, and higher acceleration control.

Acceleration and deceleration time is in milliseconds and are symmetrical. Use the **ACC** command to set and get the acceleration/deceleration value.

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

Use **SCV** command to enable s-curve velocity profile instead of trapezoidal.

On-The-Fly Speed Change

On-the-fly speed change can be achieved with the **SSPD** command. **SSPD** command is only valid with trapezoidal acceleration.

During on-the-fly speed change operation, you must keep the initial and destination speeds within a certain window. See speed setting windows below:

SSPDM value	Lowest Speed [pps]	Highest Speed [pps]
0	SSPD not used	SSPD not used
1	1	15000
2	2	30000
3	5	80000
4	10	160000
5	20	300000
6	50	800000
7	100	3000000

8	400	5000000

To select a speed window, use the **SSPDM** command. At boot-up, the **SSPDM** value is equal to 0.

If you are to set your destination speed outside of your current window, the **SSPD** feature will not work correctly.

*Note: The lower the **SSPDM** value, the more accurate the pulse output speed will be. Therefore, it is recommended to choose the lowest **SSPDM** value as possible.*

To set acceleration of the on-the-fly speed change, use the **ACC** command. Set the acceleration before calling the **SSPD** command.

Note: The maximum acceleration value allowed depends on both the SSPDM value as well as the difference between the initial and destination speeds. See table below.

SSPDM value	Speed Delta (destination speed – initial speed) [pps]
0	SSPD not used
1	1250
2	2500
3	5000
4	10000
5	20000
6	40000
7	60000
8	120000

Speed Delta [destination speed – initial speed]: For every increment of speed delta, the maximum value of acceleration increases by 2500 ms (2.5 seconds).

Examples:

a) If **SSPDM**=4: when changing speed from 1000 pps to 11,000 pps , the maximum acceleration value allowable is 2500 ms × 10 = 2500 ms (2.5 sec).
b) If **SSPDM**=4: when changing speed from 1000 pps to 101,000 pps , the maximum acceleration value allowable is 2500 ms × 10 ms = 25,000 ms (25 sec).

*Note: In order to begin normal operation after on-the-fly speed moves, it is required to first set **SSPDM** to 0.*

Motor Power

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

Polarity

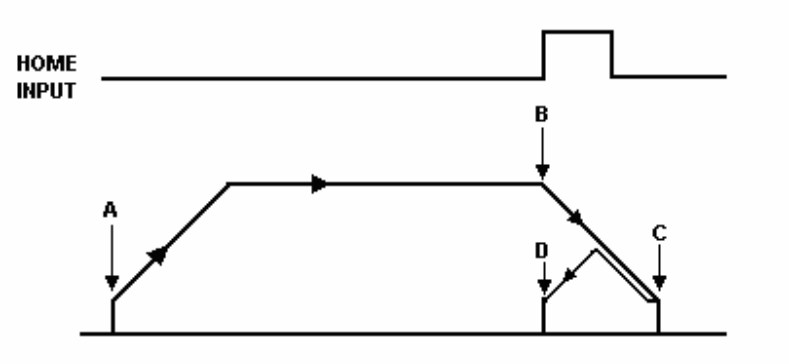
Using **POL** command, polarity of following signals can be configured:

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

Homing

Three types of automatic homing are available.

1) Homing using only the HOME input switch:



A. Issuing home command 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. If the home sensor is trigged during acceleration, position is immediately set to zero and decelerates to low speed.

C. After the deceleration, the motor stops

D. Motor is moved back to zero position where the home input was triggered.

To trigger the home input switch, supply the opto-supply voltage with 12 to 24VDC and connect the home input signal to opto-supply ground. Use **H+** or **H-** commands for this type of homing.

*Note: This type of homing has a feature to take into account the possibility of hitting a limit switch during the H+ or H- operation. See "**Home Limit Error Correction**" section.*

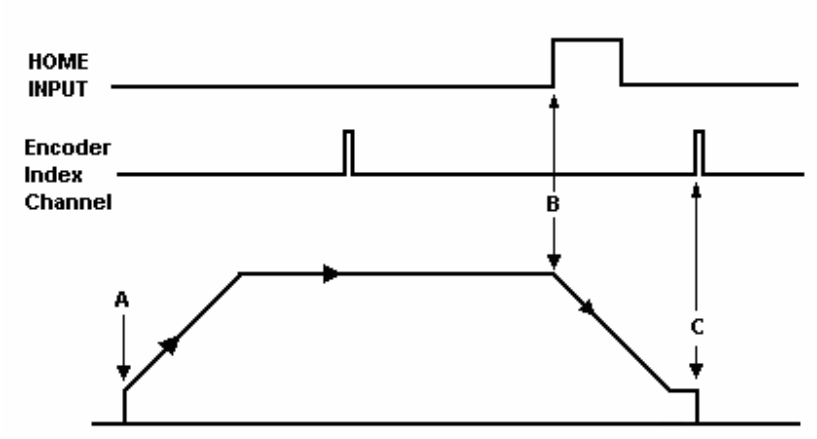
2) Homing using only the Z index encoder channel:

Z index channel occurs once per revolution of the motor. Homing can be done using only the Z index channel.

A. Z Index homing command is issued and the motor uses only the low speed.
B. When the Z index channel trigger is found, position is set to zero and the motor stop immediately.

Use the **Z+** or **Z-** commands for this type of homing.

3) Homing using the Z index encoder channel and HOME input switch:
Homing can be done using both the HOME switch input and Z index encoder channel to get accurate and consistent home position.



A. When home command is issued, pulse ramps up from low to high speed.
B. As soon as the home input is triggered, the pulse rate ramps down to low speed. If the home input is triggered during acceleration, the motor immediately decelerates.
C. Low speed is maintained until the Z index channel of the encoder is triggered at which time the position is reset to zero and motor stops. If the Index channel trigger occurs before reaching the low speed, the motor position is set to zero and stops immediately before reaching the low speed.

Use **ZH+** or **ZH-** commands to issue a Z and HOME input home command.

Limit Homing:
Limit switch can be used for homing by jogging the motor to the limit switch (using **J+** or **J-** command), clearing the limit error (using **CLR** command) and then resetting the position counter (using **PX** and **EX** command).

Jogging

Jogging is available for continuous speed operation. Use **J+** and **J-** commands to jog in positive or negative direction. To stop the motor, use the **STOP** or **ABORT** commands.

Stopping Motor

When motor is moving, jogging, or homing, **ABORT** command will immediately stop the motor. **STOP** command will decelerate the motor to low speed and then stop.

Motor Position

Pulse 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 are 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 negative direction motion. This error must be cleared (using **CLR** command) before issuing any subsequent move
7	Plus limit error. This bit is latched when plus limit is hit during positive direction motion. This error must be cleared (using **CLR** command) before issuing any subsequent move commands
8	Latch input switch status. This is also DI2 which can also be read with DI command
9	Z encoder index channel status

Limit Inputs

If positive limit switch is triggered while moving in positive direction, motor will immediately stop and the motor status bit for positive limit error is set. Same is for negative limit while moving in negative direction.

*Note: If limit switch is triggered during a H+ or H- command, the motor will move in the other direction a certain number of steps/counts set by the **LCA** command. In this case, the controller will not go into a limit error state.*

Once limit error is set, use **CLR** command to clear the error. Once the error is cleared, move the motor out of the limit switch. The limit switch is an opto-isolated input. Supply

the opto-supply voltage 12 to 24VDC. To trigger the limit input switch, connect the input signal to ground of opto-supply.

Digital Inputs / Outputs

DMX-K has 6 digital inputs and 3 digital outputs.

To read and set digital outputs, use **DO** command to for all three bits. For individual bits, use **DO[bit #]** command. For example, to set all three bits to 1 use **DO=7** command. For setting only second bit use **DO2=1**. DO command is also used to read the digital output status.

To read the 6 digital inputs, use **DI** command. To read individual bits, use **DI[bit #]** command. For example, to read all 6 bits, use **DI** command. To read the first bit use **DI1** command.

When StepNLoop control is enabled, DO1 is used as an "In Position" status output, and DO3 is used as an "Alarm" output. To use these two outputs as general purpose, you must first disable StepNLoop by using **SL** command.

When Sync Output feature is enabled, DO2 is used as Sync Digital Output. To use DO2 as general purpose output, use **SYNO and SYNF** command to enable and disable sync output.

Latch Input

DMX-K has high speed position latch input DI2.

This input is similar to Home input in that it does high speed position capture of both pulse and encoder positions but without resetting the pulse or encoder position counters.

Use **LT** command to enable and disable latch feature.

To read the latch status, use **LTS** command. Following are return value description for **LTS** command:

0 – Latch off
1 – Latch on and waiting for latch trigger
2 – Latch triggered.

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

Sync Output

DMX-K has synchronization digital output for triggering the output when the encoder position value meets the set condition. DO2 is used as synchronization digital output when the feature is enabled.

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.

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

1 – Turn the DO2 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 DO2 on when the encoder position is LESS than the sync position.

3 – Turns DO2 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

Home Limit Error Correction

DMX-K has a home limit error correction feature. During a **H+** or **H-** operation, a limit switch triggering will not cause the controller to go to an error state. Instead, the controller will move the motor to a configurable distance in the opposite direction. This will result in moving the motor away from the limit.

Use **LCA** to set/read the correction amount.

When StepNLoop closed loop control is enabled: the **LCA** value refers to number of encoder counts.

When StepNLoop closed loop control is disabled: the **LCA** value refers to number of pulse counts.

StepNLoop Closed Loop Control

DMX-K module has closed loop position control algorithm called StepNLoop control for accurate positioning of the motor using the integrated encoder.

StepNLoop control does following operations:

1) Position Delta monitoring: Delta position is the difference between the actual and the target position. When the Delta goes over the allowed Correction Range, the motor is stopped and the StepNLoop Status goes into the "stall" error state. Delta monitoring is done for all moves including homing and jogging. View the Delta value by using the **DX** command.

2) Position Correction at the end of the move: Correction of the motor position is done at the end of any targeted move.

Following are configuration required for StepNLoop control:

StepNLoop Parameters	Description
Tolerance Range	When the actual encoder position is within desired encoder position by this tolerance range, no position correction is done. Use **SLT** command to set the tolerance range.
Correction Range	When the actual encoder position is within desired encoder position by this correction range, position correction is done when idle. If the actual encoder position is outside of correction range, the motor status goes to error state. Use **SLE** command to set the correction range.
Correction Attempt Number	This is the maximum number of correction tries that the controller will attempt. If the correction cannot be done within this number of tries, the motor status goes to error state. Use **SLA** command to set the maximum correction attempt number.

To enable and disable the StepNLoop feature use **SL** command.

To read the StepNLoop status, use **SLS** command to read the status.

Following are the StepNLoop status values:

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** command.
9	Correction Attempt Error. To clear this error, use **CLRS** command.
10	Stall Error. **DX** value has exceeded the **SLE** value. To clear this error, use To clear this error, use **CLRS** command.
11	Limit Error. To clear this error, use **CLRS** command.

Notes:

Once StepNLoop is enabled, position move commands are in term of encoder position.

For example, X1000 means to move the motor to encoder 1000 position.

Once StepNLoop is enabled, the speed is in encoder speed.

For example HSPD1000 when StepNLoop is enabled means that the target high speed is 1000 encoder counts per second.

StepNLoop correction is done only when the pulse rate is idle. For example, when the motor is moving, correction is not done. Once the pulse rate is idle, StepNLoop correction is done.

Idle Current and Run Current

DMX-K 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 rung 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.

Device Number and Baud Rate:

DMX-K comes with following default factory communication setting:

Baud Rate: **9600**
Device Name: **DMK01**

DMX-K 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 [DMK01,DMK99].

DMX-K 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:

Device Baud Value	Baud Rate (bps)
1	9600
2	19200
3	38400
4	57600
5	115200

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-K modules on the RS-485 bus. When a broadcast command is received by a DMX-K module, no response is sent back to the master.

Standalone Support

DMX-K supports up to 1275 lines of compiled standalone code. See section 10 for detailed specifications on programming language.

Storing to Flash

The following items are stored to flash:

- Device Number (used for RS-485 ASCII communication)
- Baud rate
- Polarity settings
- StepNLoop enable
- StepNLoop parameters
- Run Current
- Idle Current
- Idle Time
- Automatic program run on power up
- RS-232 or RS-485 communication
- Limit error correction amount

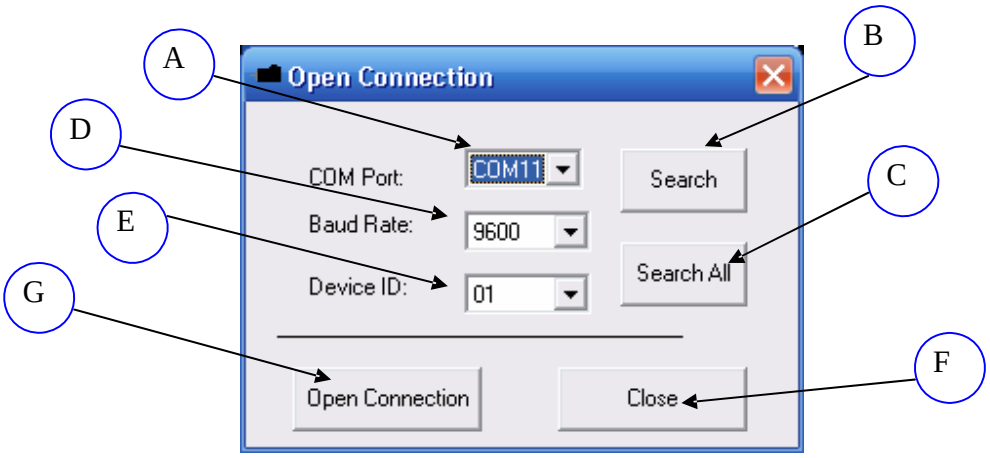
When standalone program is downloaded, the program is immediately written on the flash memory.

9. DMX-K Windows Program

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

From Arcus Technology support web site download the DMX-K installation program.

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



A. Serial Communication Port Number. This is a drop down combo box which has selection of serial port from COM1 to COM12.

B. Search button is used to search for any DMX-K module connected to an available serial communication port. Search looks for DMX-K from COM1 to COM12 using the device name.

C. Search All button is used to search for all DMX-K modules connected to the available serial communication port. This button should be used when more than one DMX-K are connected on a multi-drop RS-485 bus. Note that the DMX-K must be configured to use RS-485 communication.

D. 9600 baud rate is the default communication baud rate that is used. If you have set your DMX-K module to operate at a different baud rate, select the correct baud rate here.

E. Device name is used for RS-485 communication and to distinguish many devices on the RS-485 network.

F. Close the application program.

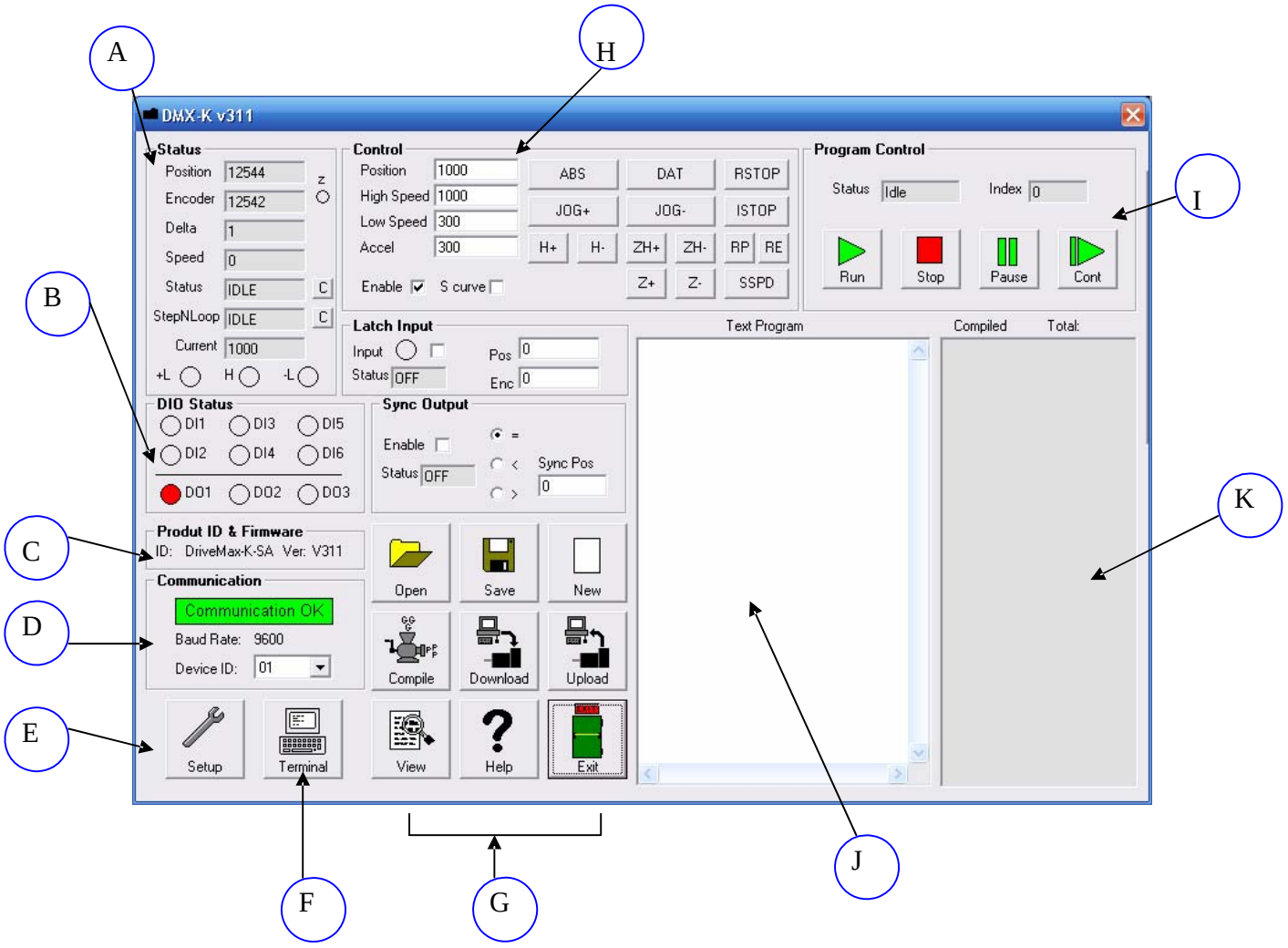
G. Open connection uses the selected COM port and the device name to open communication with the DMX-K module.

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

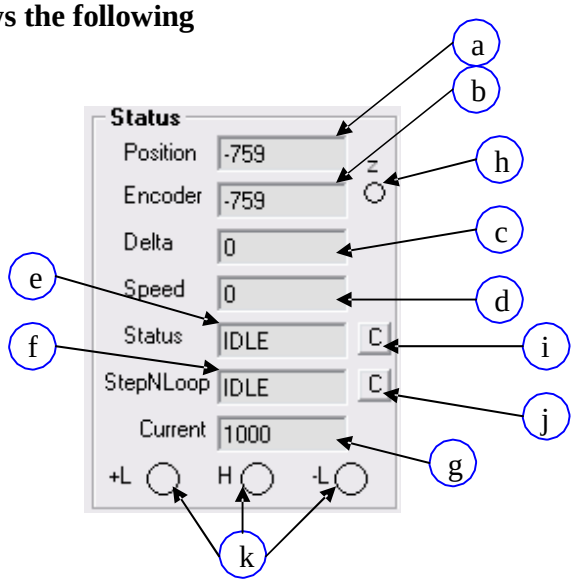
If search cannot find the DMX-K device or open connection does not find the DMX-K, check the following:

1) Check power supply to DMX-K. Recommended power is from 12VDC to 35VDC.
2) Check communication wiring. If using RS-232, TXD from DMX-K should be connected to RXD of the serial port and RXD from DMX-K should be connected to TXD of serial port. If using RS-485, make sure that the 485+ from DMX-K is connected to 485+ of the master and 485- from DMX-K 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:



A. Status box shows the following



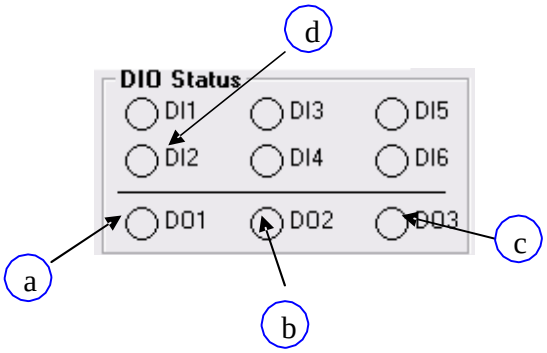
a. Current pulse position (when StepNLoop is enabled, pulse position is encoder position is shown)
b. Current encoder position
c. Delta position – this is the difference between the target position and actual position.
d. Speed – actual speed. When StepNLoop is enabled, this value is in encoder counts/sec. When StepNLoop is disabled, this value is in pulses/sec.
e. 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 f. StepNLoop status

   i. Idle – motor is not moving
   ii. Correcting – correction is in progress iii. Jogging – jogging in progress
   iv. Homing – homing in progress
   v. Stopping – decal and stop in progress

g. Current – displays active current value. Value is in mA. h. Z encoder index channel status.
i. Clear motor status – use this to clear the motor error
j. Clear StepNLoop status – use this to clear StepNLoop error
k. –Limit, + Limit, and Home input status

B. Digital Input and Output Status



a. 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.

b. 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.

c. Digital Output bit 3 – this is general purpose digital output when StepNLoop is disabled. When StepNLoop is enabled, DO3 is used as alarm output.

d. 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.

C. Product ID & Firmware

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

D. Communication

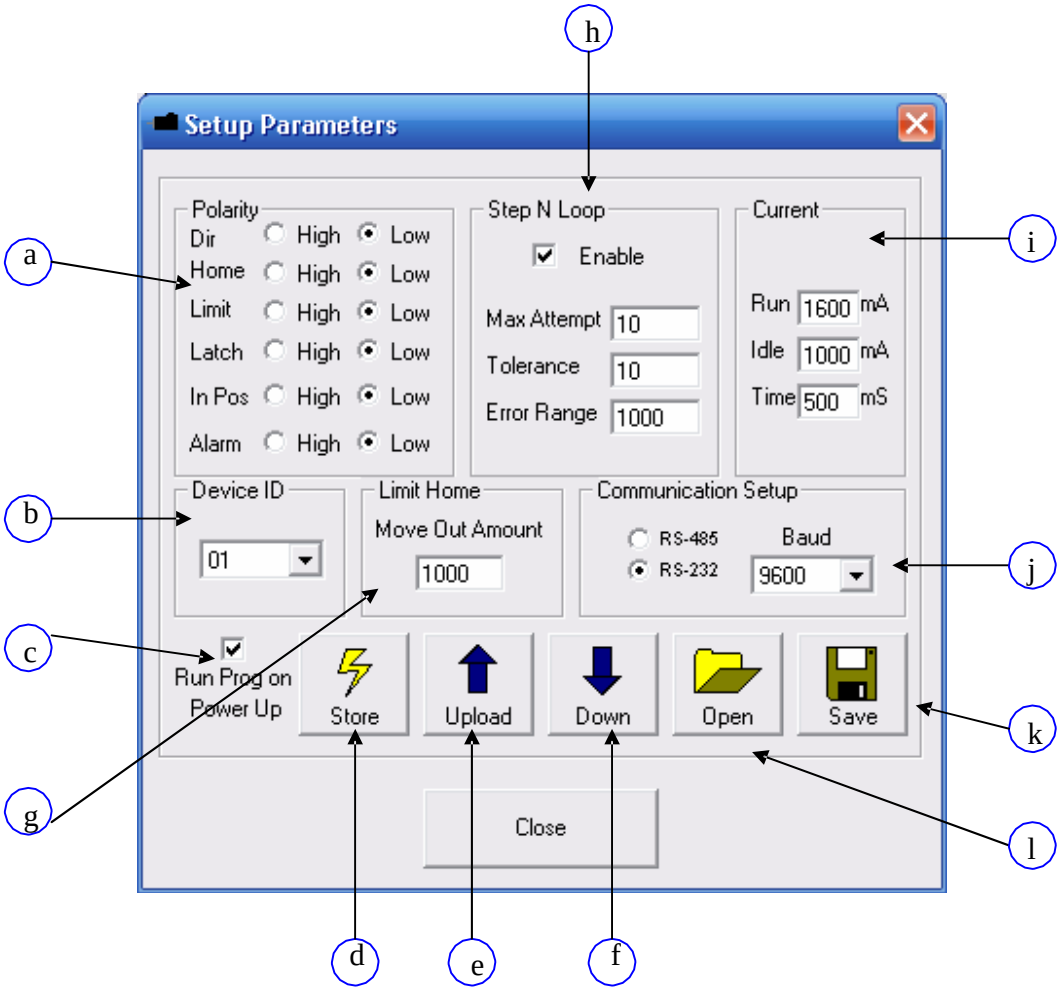
a. Communication Status – Displays communication status with the selected device.

b. Device ID – Device ID of the communicating DMX-K. To communicate with a different DMX-K on-the-fly, select another ID number from this drop-down box.

E. Setup

DMX-K 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:



a. Polarity values – direction, pulse, home, limit, latch input, in position output, alarm output values can be selected in this section.

b. Device Name – Device name ranges from DMK01 to DMK99.

c. Run Program on Power Up – this feature enables automatic run of the program when the device is powered up. This feature enables automatic run without the intervention of the external device to issue program run command.

d. Store to Flash – all configuration values in the DMX-K 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.

e. Upload – all the configuration values on the controller is uploaded and updated on the screen.

f. Download – all the configuration values on the screen is downloaded.

g. Limit Home Amount – Set limit correction amount. See detailed description in Home Limit Error Correction section.

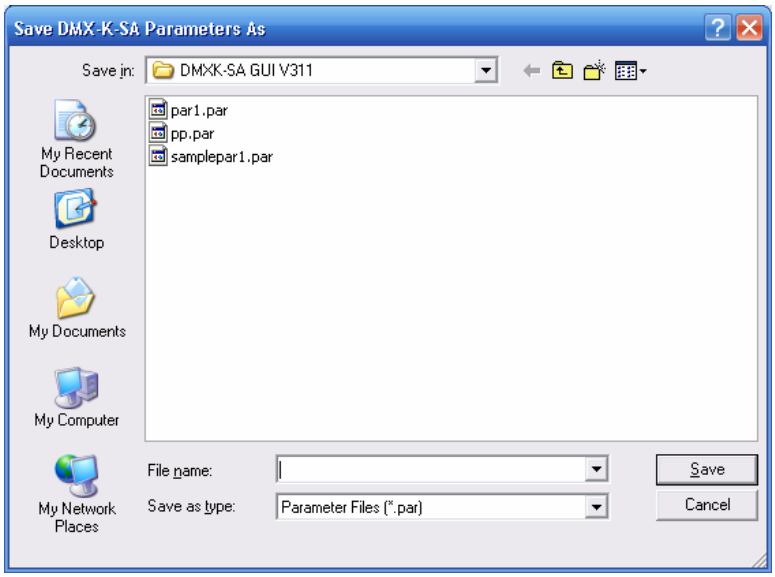
h. StepNLoop configuration values – See detailed description in the StepNLoop feature section.

i. 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.

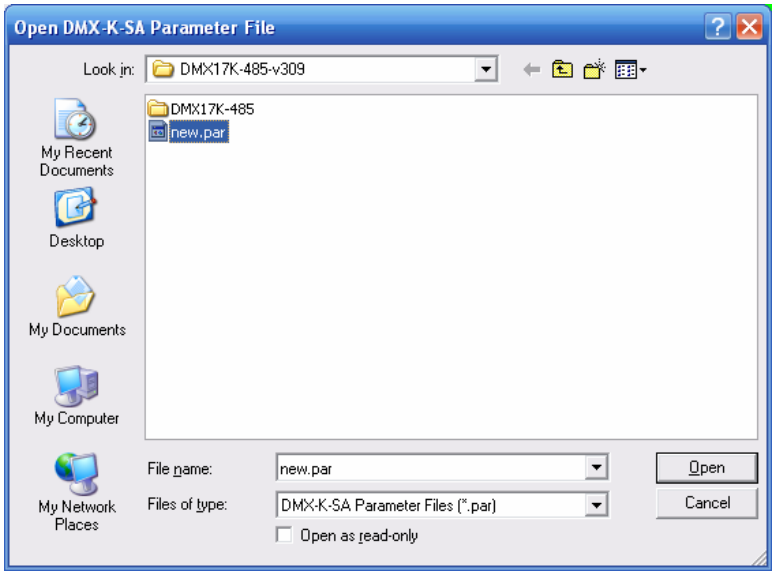
   If Idle current is set to 0 mA, the motor will be come disabled once it is idle. When the motor starts running, the motor will be enabled again.

j. Communication mode selection – RS-232 or RS-485 is selected as the communication method. Baud Rate – Set baud rate ranging from 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps.

k. Save – Save the current parameters into a file. When this button is pressed, typical Windows file save dialog box will open:

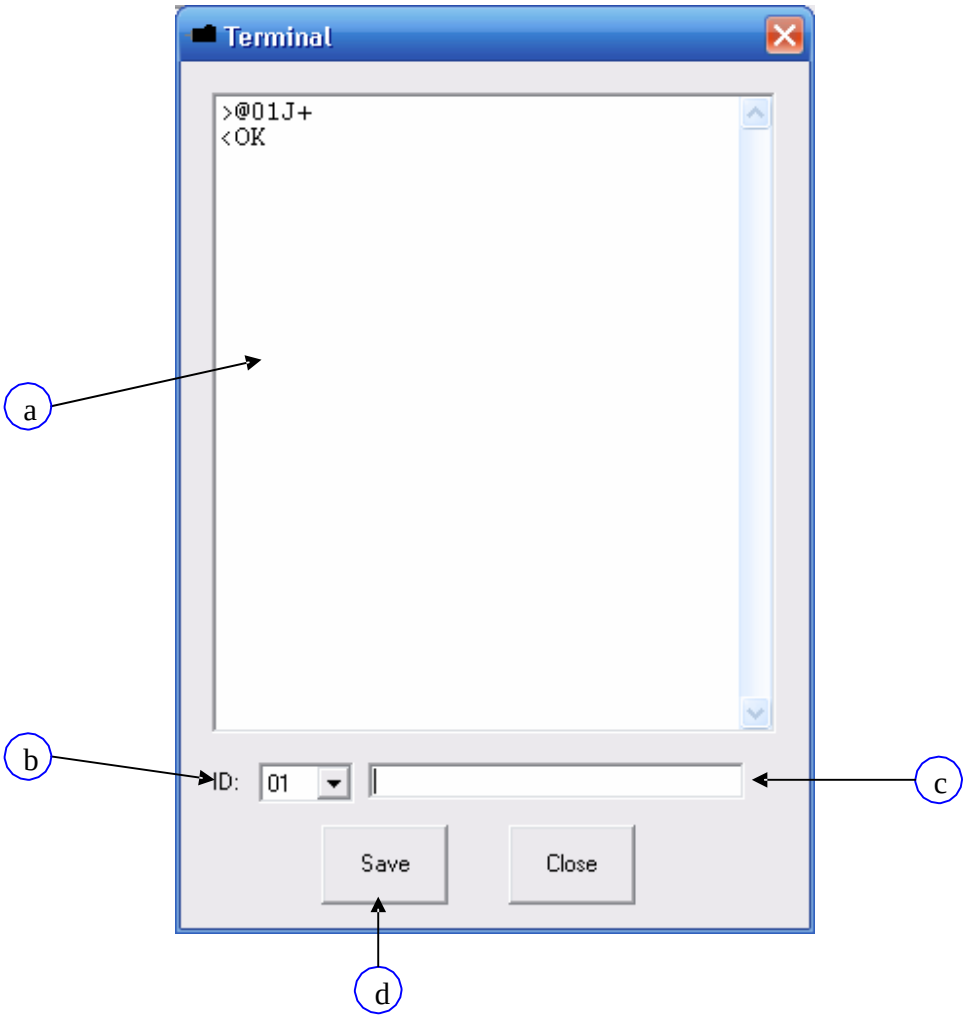


l. Open – Open a file of saved parameters. Parameters will be loaded into the configuration box. Before parameters are made effective, you must perform a download procedure When this button is pressed, typical Windows file open dialog box will open:



F. Terminal

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



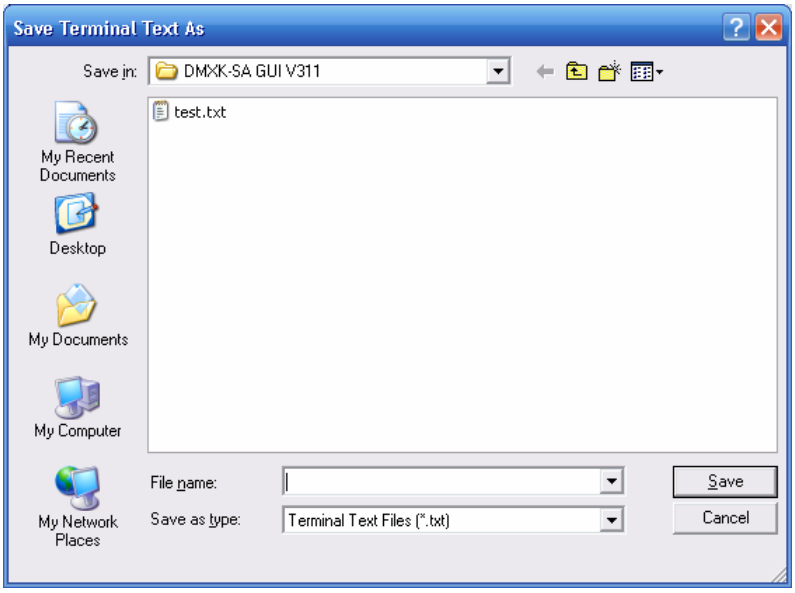
a. Response Window – Displays the response if any from the command line

b. Address – Select the address of the DMX-K module which you wish to communicate. Selecting address **'00'** will send a

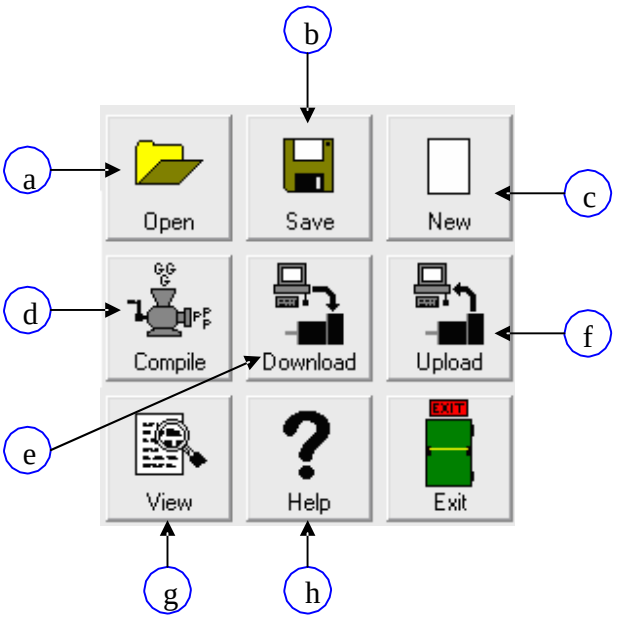
broadcast command which will be received by all DMX-K modules on a RS-485 bus.

c. 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.

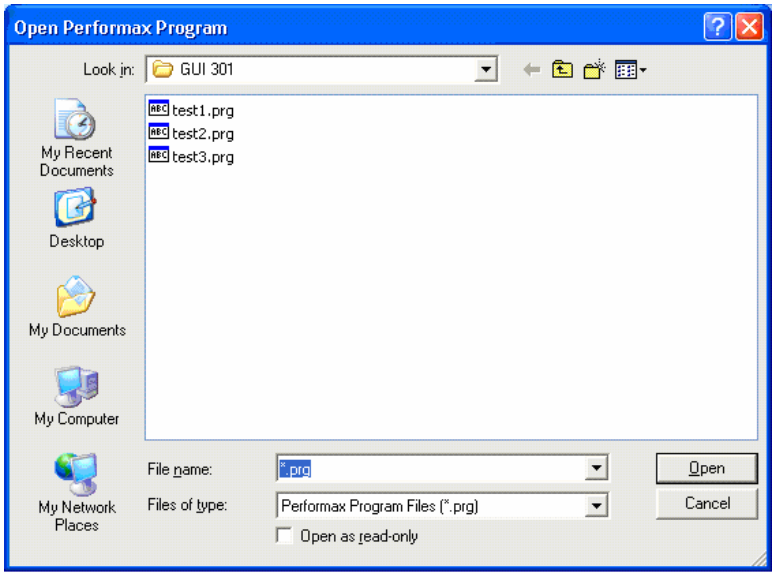
d. Save – Save the terminal text into a text file. When this button is pressed, typical Windows file save dialog box will open:



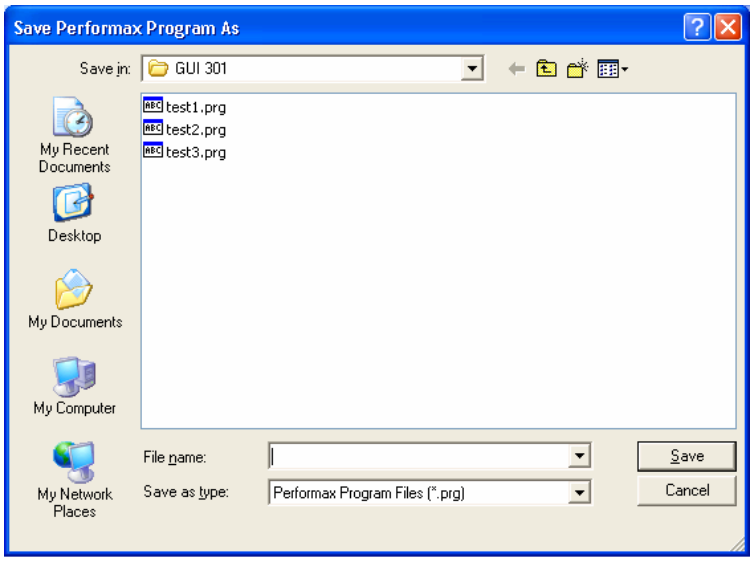
G. Program File and Standalone Program Process



a. Open - standalone program is loaded to the editor box. When this button is pressed, typical Windows file open dialog box will open:



b. Save – standalone program in the text edit is saved to a file. When this button is pressed, typical Windows file save dialog box will open:



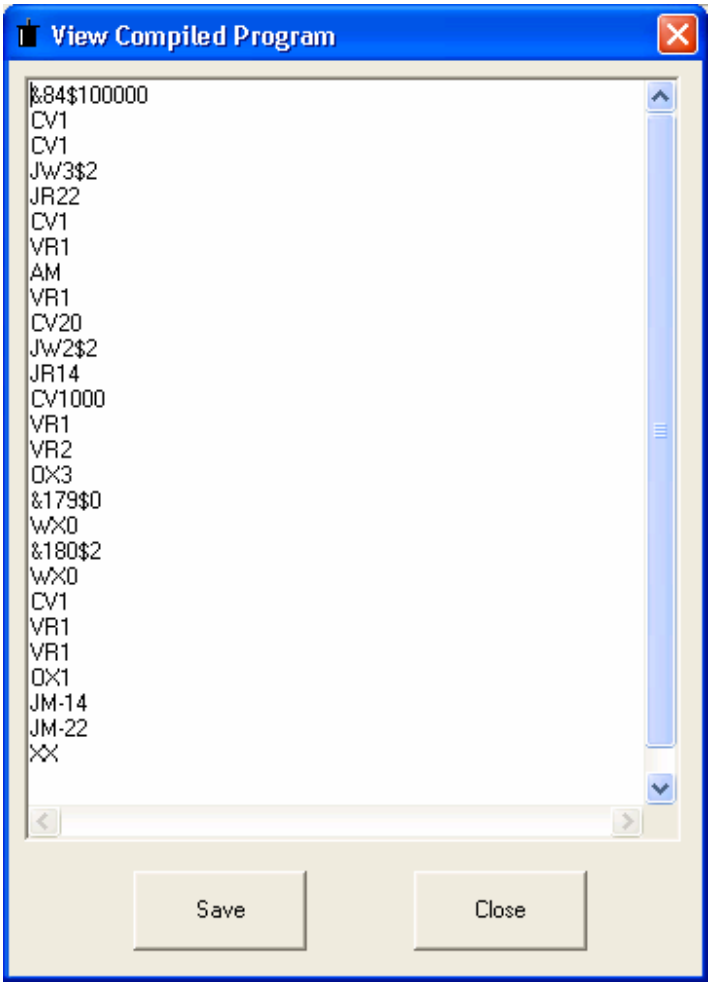
c. New – when this button is pressed, the text editor is cleared.

d. Compile – use this button to compile the text program.

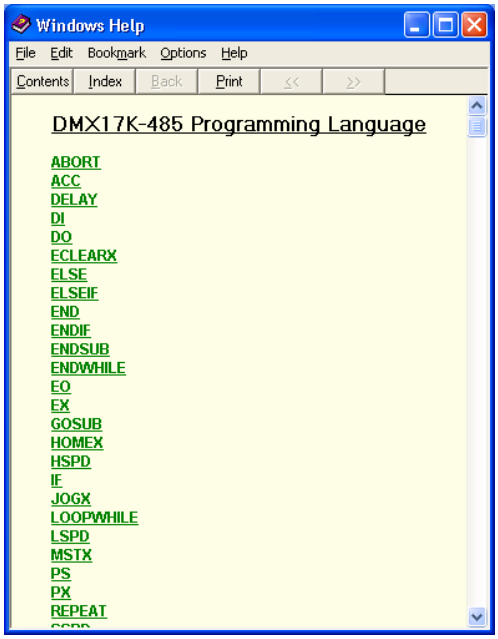
e. Download – use this button to download the compiled low level program to the controller.

f. Upload – use this button to upload the compiled low level program from the controller to the text editor.

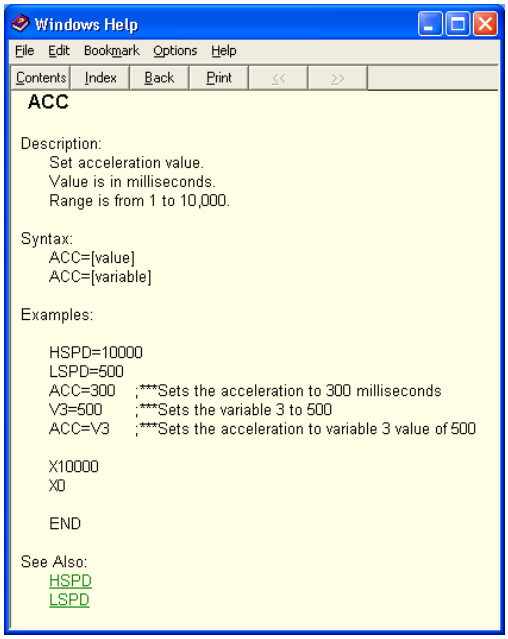
g. Low Level Program Viewer - Low level program can be viewed in a text editor. When this button is pressed, the low level program on the list is transferred to the text editor which allows you to save the low level program to a file.



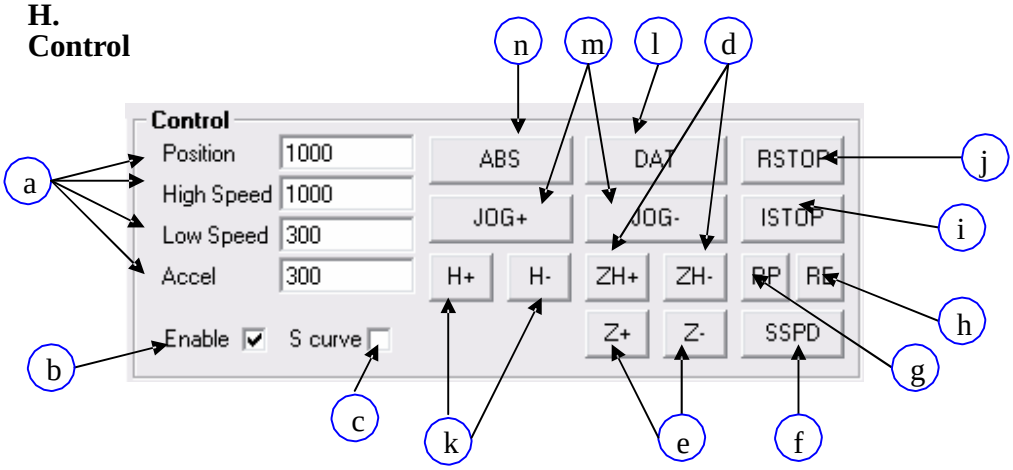
h. DMX-K Windows program has online help for the high level language for the standalone programming. When help button is pressed, following help page will show:



This is typical help file where you can click each command and see the description, syntax, example, and related commands as shown below.



H.
Control



a. Target position, high speed, low speed, and acceleration is entered here

b. Enable – motor power is turned on or off using this check box. When motor is disabled, no motion is done.

c. S-Curve – when this check box is checked, S-curve profile is used. If this check box is cleared, trapezoidal acceleration profile is used.

d. ZH+/ZH- - Home sensor and encoder index channel is used to home.

e. Z+/Z- - Only the encoder index channel is used for homing.

f. SSPD – On-the-fly speed change is issued with this button. SSPD is valid only when the motion profile is trapezoidal. When clicked, speed accelerate/decelerate to the value in "High Speed"

g. RP – Resets the current pulse position to zero.

h. RE – Resets the current encoder position to zero.

i. ISTOP – the motion is immediately stopped without deceleration.

j. RSTOP – the motion is stopped with deceleration.

k. 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.

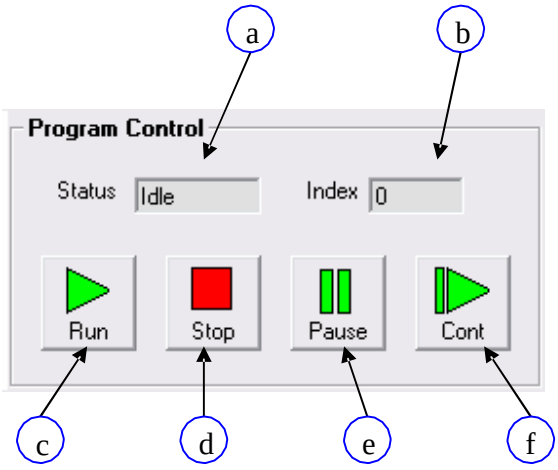
l. DAT – moves the motor to the zero target position.

m. JOG+/JOG- - jogs the motor in positive and negative direction.

n. ABS – moves the motor to the target absolute position using the high speed and the low speed and the acceleration values.

I. Program Control

DMX-K has standalone program capability. Standalone program is written using an easy to use and understand text based programming language, it is then compiled, downloaded and run.



a. Program Status – program status shows here. Following are possible program status: Idle, Running, and Paused.

b. Index – program that is downloaded is in the form of low-level code. Each line of the low level code has a line number which shows here.

c. Run – program is run.

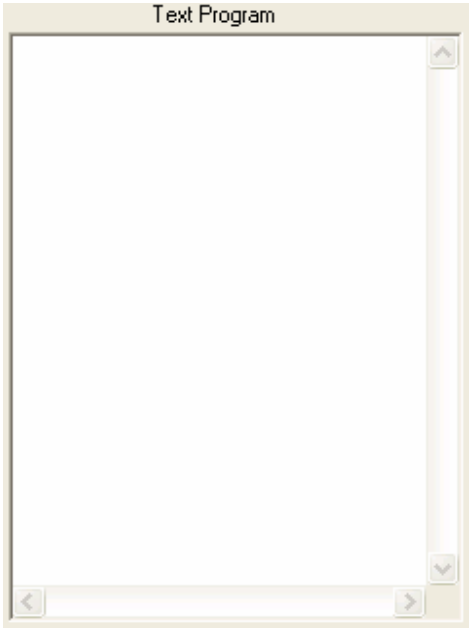
d. Stop – program is stopped.

e. Pause – program that is running can be paused.

f. Continue – program that is paused can be continued

J. Text Program Editor

DMX-K has standalone program capability. Standalone program can be loaded and written in this area.



K. Low Level Program List

Once the high level program is compiled, the low level code is shown in this list. This list is not editable.

10. Standalone Language Specification

Following are the details of the language that can be used to program standalone program.

;

Description:

Comment notation

Note: In programming, comment must be in its own line.

Syntax:

; [Comment Text]

Examples:

```
; ***This is a comment

J+                ;***Jogs X axis to positive direction
DELAY=100         ;***Wait 1 second
ABORT             ;***Stop immediately all axes including X axis
```

ABORT

Description:

Immediately stops all axes if in motion. Also clears StepNLoop error.

Syntax:

ABORT

Examples:

```
JOGX+             ;***Jogs X axis to positive direction
DELAY=100         ;***Wait 1 second
ABORT             ;***Stop immediately all axes including X axis
```

ACC

Description:
Set acceleration value.
Value is in milliseconds.
Range is from 1 to 10,000.

Syntax:
ACC=[long value]
ACC=[variable]

Examples:

```
ACC=300      ;***Sets the acceleration to 300 milliseconds
V3=500       ;***Sets the variable 3 to 500
ACC=V3       ;***Sets the acceleration to variable 3 value of 500
```

DELAY

Description:
Set a delay (1 ms units)

Syntax:
Delay=[Number] (1 ms units)

Examples:

```
JOGX+            ;***Jogs X axis to positive direction
DELAY=1000       ;***Wait 1 second
ABORT            ;***Stop with deceleration all axes including X axis
```

DI

Description:
Gets the digital input value
DriveMax-K-485 has 6 digital inputs

Syntax:
[variable] = DI

Examples:

```
DO=7                    ;***Turn first 3 bits on and rest off

IF DI=0
    DO=1                ;***If no digital inputs are triggered, set DO=1
ENDIF
```

DI[1-6]

Description:
Gets the individual digital input value
DriveMax-K-485 has 6 digital inputs

Syntax:
[variable] = DI[1-6]

Examples:

```
DO=7                    ;***Turn first 3 bits on and rest off

IF DI1=0
    DO=1                ;***If digital input 1 is off, turn digital output 1 on
ENDIF
```

DO

Description:

Sets the digital output value
DriveMax-K-485 has 3 digital outputs

Syntax:

DO = [value]
DO = [variable]

Examples:

```
DO=7                    ;***Turn 3 bits on

V1=0
WHILE V1<7
        DO=V1           ;***Output DO to value variable 1
        V1=V1+1
ENDWHILE
END
```

DO[1-3]

Description:

Sets the individual digital output value
DriveMax-K-485 has 3 digital outputs

Syntax:

DO[1-3] = [value]
DO[1-3] = [variable]

Examples:

```
DO3=7                   ;***Turn digital output 3 on
```

EX

Description:
Sets the current encoder position

Syntax:
EX

Examples:

```
JOGX+           ;***Jogs X axis to positive direction
DELAY=100       ;***Wait 1 second
ABORT           ;***Stop with deceleration all axes including X axis
```

ECLEARX

Description:
Clears error status

Syntax:
ECLEARX

Examples:

```
ECLEARX             ;***Clears error of axis X
```

ELSE

Description:

Perform ELSE condition check as a part of IF statement

Syntax:

ELSE

Examples:

```
IF V1=1
        X1000           ;***If V1 is 1, then move to 1000
ELSE
        X-1000          ;***If V1 is not 1, then move to -1000
ENDIF
```

ELSEIF

Description:

Perform ELSEIF condition check as a part of the IF statement

Syntax:

ELSEIF [Argument 1] [Comparison] [Argument 2]

[Argument] can be any of the following:

- Numerical value
- Pulse or Encoder Position
- Digital Output
- Digital Input
- Enable Output
- Motor Status

[Comparison] can be any of the following

=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
!=	Not Equal to

Examples:

```
IF V1=1
        X1000
ELSEIF V1=2
        X2000
ELSEIF V1=3
        X3000
ELSE
        X0
ENDIF
```

END

Description:

Indicate end of program.
Program status changes to idle when END is reached.

Note: Subroutine definitions should be written AFTER the END statement

Syntax:

END

Examples:

```
X0
X1000
END
```

ENDIF

Description:

Indicates end of IF operation

Syntax:

ENDIF

Examples:

```
IF V1=1
        X1000
ENDIF
```

ENDSUB

Description:

Indicates end of subroutine
When ENDSUB is reached, the program returns to the previously called subroutine.

Note: Subroutine definitions should be written AFTER the END statement

Syntax:

ENDSUB

Examples:

```
GOSUB 1
END

SUB 1
    X0
    X1000
ENDSUB
```

ENDWHILE

Description:

Indicate end of WHILE loop

Syntax:

ENDWHILE

Examples:

```
WHILE V1=1          ;***While V1 is 1 continue to loop
    X0
    X1000
ENDWHILE            ;***End of while loop so go back to WHILE
```

EO

Description:
Sets the enable output value

Syntax:
EO= [value]
EO = [variable]

Examples:

```
EO=1                    ;***Turn on enable output
```

GOSUB

Description:
Perform go to subroutine operation
Subroutine range is from 1 to 32.

Note: Subroutine definitions should be written AFTER the END statement

Syntax:
GOSUB [subroutine number]

[Subroutine Number] range is 1 to 32

Examples:

```
GOSUB 1
END

SUB 1
        X0
        X1000
ENDSUB
```

HOMEX

Description:
Perform homing using current high speed, low speed, and acceleration.

Syntax:
HOMEX[+ or -]

Examples:
```
HOMEX+      ;***Homes X axis in positive direction
```

HSPD

Description:
Sets high speed. Value is in pulses/second.
Range is from 1 to 6,000,000.

Syntax:
HSPD=[long value]
HSPD=[variable]

Examples:
```
HSPD=10000   ;***Sets the high speed to 10,000 pulses/sec

V1=2500      ;***Sets the variable 1 to 2,500
HSPD=V1      ;***Sets the high speed to variable 1 value of 2500
```

IF

Description:

Perform IF condition check

Syntax:

IF [Argument 1] [Comparison] [Argument 2]

[Argument] can be any of the following:

- Numerical value
- Pulse or Encoder Position
- Digital Output
- Digital Input
- Enable Output
- Motor Status

[Comparison] can be any of the following

=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
!=	Not Equal to

Examples:

```
IF V1=1
    X1000
ENDIF
```

JOGX

Description:

Perform jogging using current high speed, low speed, and acceleration.

Syntax:

JOGX[+ or -]

Examples:

```
JOGX+        ;***Jogs X axis in positive direction

JOGX-        ;***Jogs X axis in negative direction
```

LSPD

Description:

Set low speed.
Value is in pulses/second.
Range is from 1 to 6,000,000.

Syntax:

LSPD=[long value]
LSPD=[variable]

Examples:

```
LSPD=1000    ;***Sets the start low speed to 1,000 pulses/sec

V1=500       ;***Sets the variable 1 to 500
LSPD=V1      ;***Sets the start low speed to variable 1 value of 500
```

MSTX

Description:

Get motor status of axis

Syntax:

MST[Axis]

Examples:

```
IF MSTX=0
        DO=6
ENDIF
```

PS

Description:
Gets the current speed of the controller

Syntax:
[variable] = PS

Examples:

```
IF PS=2999
        DO=1                ;***If pulse speed is 2999, set DO=1
ENDIF
```

PX

Description:
Sets the current pulse position

Syntax:
PX=[long value]
PX=[variable]

Examples:

```
JOGX+           ;***Jogs X axis to positive direction
DELAY=100       ;***Wait 1 second
ABORT           ;***Stop with deceleration all axes including X axis
PX=0            ;***Sets the current pulse position to 0
```

SCV

Description:
S-curve is enabled and disabled.

Syntax:
SCV=[0 or 1]

Examples:

```
SCV=1          ;***Enables S Curve
SCV=0          ;***Disables S Curve
```

SSPD

Description:
Change speed of motor on the fly

Syntax:
SSPD=[long value]
SSPD=[variable]

Examples:

```
JOGX+          ;***Jogs X axis to positive direction
DELAY=100      ;***Wait 1 second
SSPD=1000      ;***Change X axis speed on the fly
DELAY=100      ;***Wait 1 second
SSPD=2000      ;***Change X axis speed on the fly
```

SSPDM

Description:

Change on-the-fly speed window

Syntax:

SSPDM=[long value]
SSPDM=[variable]

Examples:

```
SSPDM=5          ;***Change on-the-fly speed window to 5
HSPD=1000        ;***Set high speed to 1000
JOGX+            ;***Jogs X axis to positive direction
DELAY=100        ;***Wait 1 second
ACC=20000        ;***Set acceleration to 20 sec
SSPD=180000      ;***Change X axis speed on the fly with acc of 20 sec
```

STOPX

Description:

Stop all axes if in motion with deceleration.
Previous acceleration value is used for deceleration.

Syntax:

STOP[Axis]

Examples:

```
JOGX+            ;***Jogs X axis to positive direction
DELAY=100        ;***Wait 1 second
STOPX            ;***Stop with deceleration all axes including X axis
```

SUB

Description:

Indicates start of subroutine

Syntax:

SUB [subroutine number]

[Subroutine Number] range is 1 to 32

Examples:

```
GOSUB 1
END

SUB 1
    X0
    X1000
ENDSUB
```

SYNCCFG

Description:

Sets the synchronization output condition

1 – Encoder position is EQUAL to sync position
2 – Encoder position is LESS than sync position
3 – Encoder position is GREATER than sync position

Syntax:

SYNCCFG=[1,2,3]

Examples:

```
EX=0
EO=1
SYNCPOS= -1000
SYNCCFG=2          ;***When encoder is less than -1000 sync output done
SYNCON
JOGX-              ;***Jog negative
WHILE SYNCSTAT!=2
ENDWHILE
STOPX
SYNCOFF
END
```

SYNCOFF

Description:

Turns off synchronization output feature

Syntax:

SYNCOFF

Examples:

```
EX=0               ;***Reset the encoder position to zero
EO=1               ;***Enable the motor
SYNCPOS=10000      ;***Set sync position to 10000
SYNCCFG=1          ;***Trigger output when position equal
SYNCON             ;***Turn the sync feature on
JOGX+              ;***Jog the motor
WHILE SYNCSTAT!=2      ;***Wait until trigger happened
ENDWHILE
```

```
STOPX           ;***Stop the motor
SYNCOFF         ;***Turn off synchronization
END
```

SYNCON

Description:

Turns on synchronization output feature

Syntax:

```
SYNCON
```

Examples:

```
SYNCON
END
```

SYNCPOS

Description:

Sets synchronization output position

Syntax:

```
SYNCPOS=[value]
SYNCPOS=[variable]
```

Examples:

```
V1=1000
SYNCPOS=V1
```

SYNCSTAT

Description:

Reads the sync output status.

0 – idle
1 – waiting for sync
2 – sync occurred.

Syntax:

V[var #]=SYNCSTAT
SYNCSTAT can be used with IF, WHILE statements.

Examples:

```
V1=SYNCSTAT

WHILE SYNCSTAT=1
ENDWHILE

IF SYNCSTAT=2
ENDIF
```

V[1-99]

Description:

Assign to variable.
DriveMax-K-SA has 99 variables [V1-V99]

Syntax:

V[Variable Number] = [Argument]
V[Variable Number] = [Argument1][Operation][Argument2]

Special case for BIT NOT:
V[Variable Number] = ~[Argument]

[Argument] can be any of the following:

- Numerical value
- Pulse or Encoder Position
- Digital Output
- Digital Input
- Enable Output
- Motor Status

[Operation] can be any of the following

+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
>>	Bit Shift Right
<<	Bit Shift Left
&	Bit AND
|	Bit OR
~	Bit NOT

Examples:

```
V1=12345        ;***Set Variable 1 to 123

V2=V1+1         ;***Set Variable 2 to V1 plus 1

V3=DI           ;***Set Variable 3 to digital input value

V4=DO           ;***Sets Variable 4 to digital output value
```

```
V5=~EO          ;***Sets Variable 5 to bit NOT of enable output value

V6=DO & 8       ;***Bit AND digital output and 8 and save to var 6

V7=MSTX >> 2    ;***Bit shift right motor X status by 2 and save to var 7
```

WAITX

Description:

Tell program to wait until move on the certain axis is finished before executing next line.

Syntax:

```
WAITX
X[variable]
```

Examples:

```
X10000      ;***Move X Axis to position 10000
WAITX       ;***Wait until X Axis move is done
DO=5        ;***Set digital output
X3000       ;***Move X Axis to 3000
WAITX       ;***Wait until X Axis move is done
```

WHILE

Description:

Perform WHILE loop

Syntax:

WHILE [Argument 1] [Comparison] [Argument 2]

[Argument] can be any of the following:

- Numerical value
- Pulse or Encoder Position
- Digital Output
- Digital Input
- Enable Output
- Motor Status

[Comparison] can be any of the following

=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
!=	Not Equal to

Examples:

```
WHILE V1=1          ;***While V1 is 1 continue to loop
    X0
    X1000
ENDWHILE
```

X

Description:
Perform X axis move to target location
With other Axis moves in the same line, linear interpolation move is done.

Syntax:
X[value]
X[variable]

Examples:
```
X10000          ;***Move X Axis to position 10000

V10 = 1200      ;***Set variable 10 value to 1200
XV10            ;***Move X Axis to variable 10 value
```

ZHOMEX

Description:
Perform Z-homing using current high speed, low speed, and acceleration.

Syntax:
ZHOMEX[+ or -]

Examples:
```
ZHOMEX+     ;***Z Homes X axis in positive direction
```

ZOMEX

Description:
Homes using low speed to the Z-index.

Syntax:
ZOMEX[+ or -]

Examples:
```
ZOMEX+      ;***Z Homes X axis in positive direction
```

Contact Information

Arcus Technology, Inc.

48501 Warm Springs Blvd #104
Fremont, CA 94539
510-661-9100
510-661-9200

www.arcus-technology.com