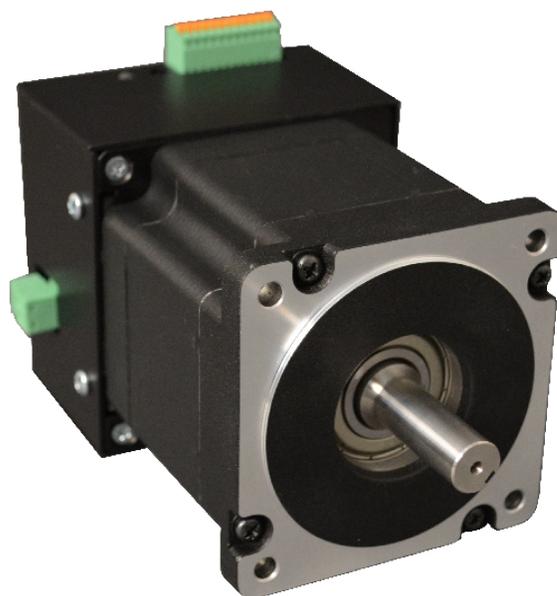


- **Stepper Motor / Microstep Driver / Controller**
- **Eliminates Motor Wires**
- **Encoder Options Available**
- **Microstep Divisor of 10**
- **Compact Package**
- **20-80V Power Requirement**
- **RS485 Communications**
- **TTL Logic or 24V Level Inputs Available**
- **Ideal for Precise Positioning**
- **0.18° Step Resolution**
- **Efficient and Durable**
- **Long Life Expectancy**



The 34MDSI Series has a compact construction that integrates a simple indexer/driver/controller, microstepping driver and a stepper motor in one streamline package. With the three parts combined into one casing, the need to include motor wires has been eliminated. The microstepping driver will operate off 20VDC minimum to 80VDC maximum. The 34MDSI Series has an output current capability of 1.5 Amps minimum to 10.0 Amps maximum (Peak Rating). The inputs are capable of running from either open collector, TTL level logic outputs or 24VDC outputs from PLCs and are all active low. The microstepping driver features resolution of 2000 steps/revolution, providing smooth rotary operation. The 34MDSI Series comes in either a single shaft versions or a double shaft version with optical encoder. Motor stack lengths of 1, 2, or 3 allow for varying amounts of start-up torque, dynamic torque and inertia. The 34MDSI Series features built in over temperature and short circuit shut down protection. It also has automatic 50% reduction in stepper motor current after indexing is complete and status LED's to indicate power on (green LED), motor running (yellow LED) and communication errors (red LED).



**ANAHEIM AUTOMATION**

4985 E. Landon Drive Anaheim, CA 92807  
e-mail: [info@anaheimautomation.com](mailto:info@anaheimautomation.com)

(714) 992-6990 fax: (714) 992-0471  
website: [www.anaheimautomation.com](http://www.anaheimautomation.com)

Input Pin Descriptions	
Pin #	Description
1	RS485 B(+)
2	RS485 A(-)
3	RS485 Gnd
4	Input 1
5	Input 2
6	On/Off
7	Direction In
8	Hard Limit +
9	Hard Limit -
10	Output 1
11	0VDC (Gnd)
12	0VDC (Gnd)

**Control Inputs (Pins 4, 5,6,7):**

Input 1,2: Logic "1" (Open) = Not Active  
 Logic "0" (0VDC) = Active  
 Direction: Logic "1" (Open) = CW  
 Logic "0" (0VDC) = CCW  
 On/Off: Logic "1" (Open) = On  
 Logic "0" (0VDC) = Off

**Limit Inputs (Pins 8,9):**

Limits: Logic "1" (Open) - Not Active, Motor Running  
 Logic "0" (0VDC) - Active, Motor stops in given Direction

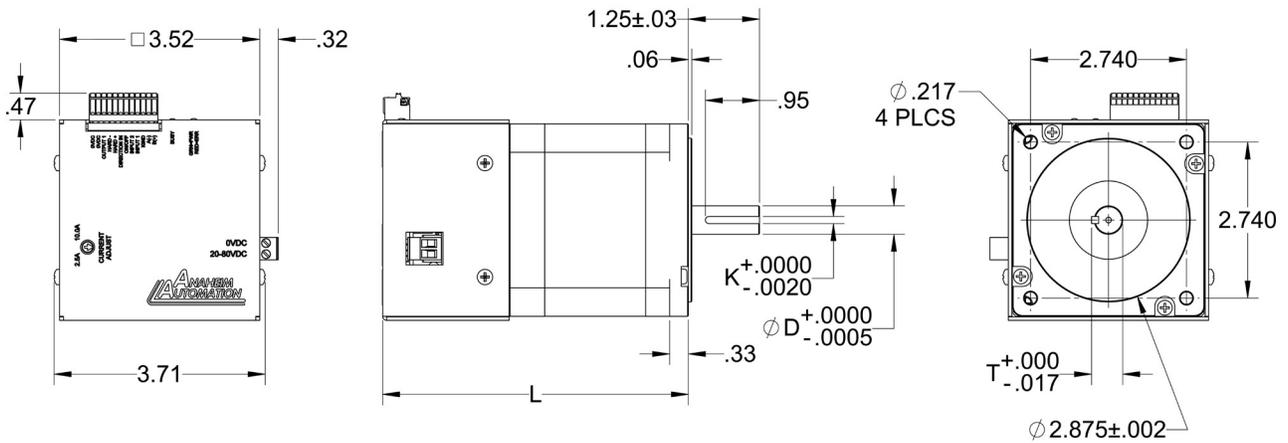
Note:

Open Inputs are inactive and internally pulled up to +5VDC for all 34MDSI models

**Control Outputs (Pin 10):**

Output 1: Open Drain Type Output, Vce (max) = 40VDC, Ic (max) = 50mA  
 This output will conduct current when active, and be open when not active. Current needs to be limited into this pin. For the 34MDSI Series, this output is a complete output.

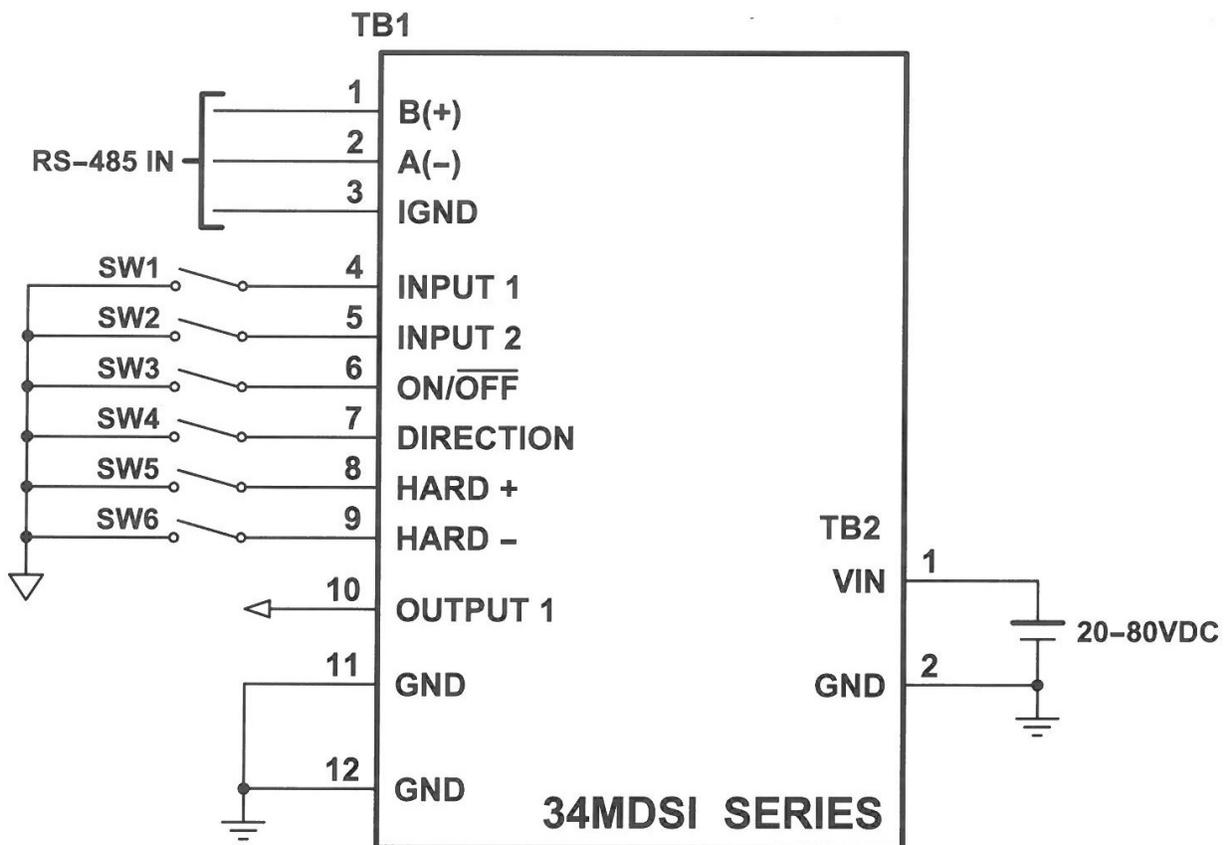
# Dimensions



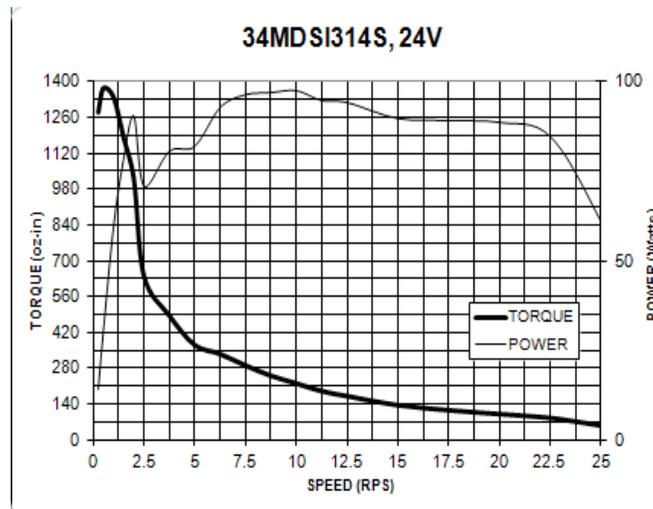
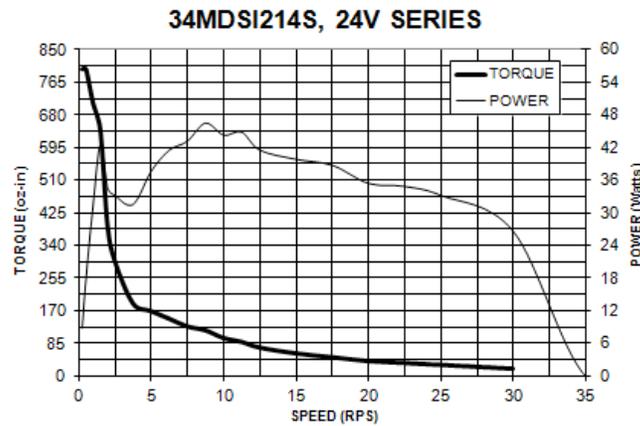
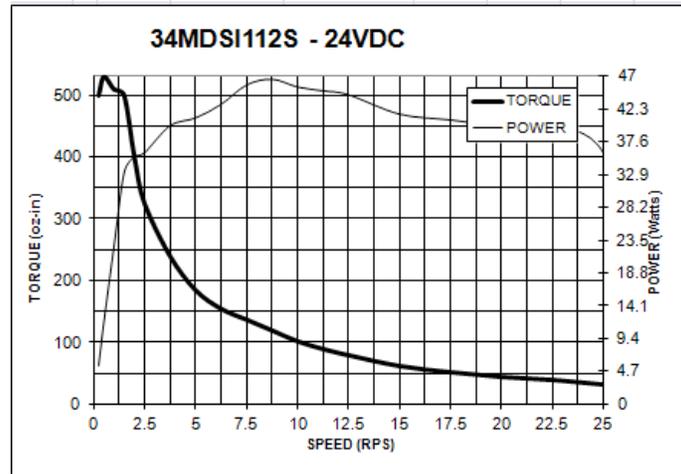
Units: (Inch)

Model #	NEMA Size	Bipolar Torque (oz-in)	Rotor Inertia (oz-in-sec <sup>2</sup> )	Weight (lbs)	L (in)	K (in)	T (in)	D (in)
34MDSI112S	34	637	0.0198	5.70	5.40	0.125	0.555	0.500
34MDSI214S	34	1200	0.038	8.94	6.90	0.125	0.555	0.500
34MDSI314S	34	1700	0.057	12.10	8.39	0.1875	0.705	0.625

# Hook-Up Drawings



# Torque Curves



## Accessories Ordering Information

Part Number	Description
PSEPM-0320-1-48V-6.7A	48V @ 6.7A Input Power Supply
PSEPM-0350-1-48V-7.3A	48V @ 7.3A Input Power Supply
PSA80V4A-1	80V @ 4.0A Input Power Supply

## Inputs and Outputs

**Inputs:** All inputs are pulled up to 5VDC. A logic “0” activates inputs that are pulled up. An unconnected input will always remain inactive.

**Direction:** When this input is not active, the motor will be moving in the clockwise or “+” direction. When this input is active, the motor will move in the counterclockwise or “-“ direction. This input is not read when a software index command is given. To change direction while using the software, change the direction option there. When two motors are used, the second motor will move in the opposite direction by default.

**On/Off:** When this input is not active, the motor will be enabled or energized. When this input is active, the motor will be disabled or de-energized.

**Input 1,2:** These inputs are used to select one of the two profiles. When one of the profiles is activated, the unit will change the speeds, acceleration, index number and complete time based on the pre-programmed values for that profile and begin the index. Activate only one of these inputs at once.

**Hard Limits:** These two inputs are controlled by the direction of the indexer. When the indexer is running in the positive direction only hard+ will work. When the indexer is running in the negative direction then only hard- will work. When pulled low the indexer will stop all pulses to the motor. To reverse off of a Hard Limit, change directions, and activate another profile input again to move in the opposite direction.

**Output 1 (Complete Output):** This is an open collector output that is capable of sinking 50mA. It is current sinking when the indexer finishes its index cycle for the pre-programmed amount of time.

## Microstep Mode

The 34MDSI Series only has one Microstep Mode, divisor of 10. This provides the motor with a microstep resolution of 2,000 steps/resolution for smooth rotary operation.

## Reducing Output Current

Reducing the output current is accomplished automatically. The amount of current per phase in the reduction mode is approximately 50% of the set motor current. When the current reduction circuit is activated, this lowers the per phase output current. This is done when the indexer is not running.

## LEDs

When powered and operated properly, the status LED will be green. When an error occurs, the LED will change to RED and an error code will be generated in the error code register. To read and clear the error with the software, click on the “Verify Parameters” button. To read and clear the error while in “Direct Mode” use the “!” command. Once the error has been read and cleared, the LED will return to green and the error code register will be cleared. Refer to the table in section 5 for a list of the error codes. When the indexer is running the yellow LED will be on. Refer to the dimension drawing for location of the LEDs. For more detail on “Direct Mode” refer to the Direct Talk Mode section of user’s guide.

## Baud Rates

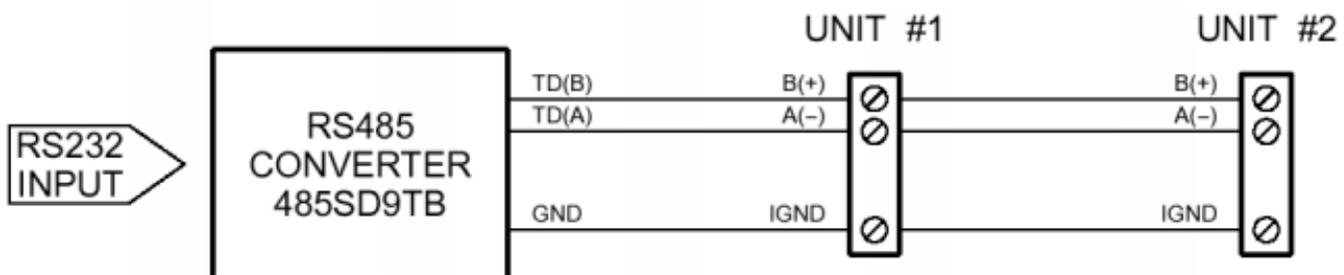
The 34MDSI Series only accepts a baud rate of 38400. A baud rate is a term used frequently in serial data communications. A “baud” is defined as the reciprocal of the shortest pulse duration in a data word signal, including start, stop, and parity bits. This is often taken to mean the same as “bits per second”, a term that expresses only the number of “data” bits per second. Very often, the parity bit is included as an information or data bit.

## RS485 Protocol

The 34MDSI Series uses a RS485 protocol for communication. The RS485 protocol mode is as follows; On board receivers will remain in active mode indefinitely. Transmitters must be turned off when the unit is not sending data, to prevent the line from sending and receiving data at the same time. Therefore when the PC is transmitting data its driver will be turned on and each of the units connected will have their drivers off. If they are requested to send data back to the PC, the selected unit will turn it's driver on to send the data and then turn it off after it has completed transmission. Note: The above protocol is done internally between the converter and the 34MDSI Series. The RS485 method of communication allows increased noise immunity and increased communication distance of up to 4000 feet without repeaters. RS485 repeaters allow an additional 4000 feet per repeater. The 34MDSI Series is designed for two wire configuration. The 2 wire configuration makes use of the tristate capabilities of RS485 to allow a single pair of wires to share transmit and receive signals for half duplex communications. This “two wire” configuration (note that an additional ground conductor must be used) reduces cabling cost. **NOTE: Keep control wiring separated from motor cable/wiring.**

## RS232 to RS485 for multiple units or cables longer than 50ft

The 34MDSI Series can be connected to your PC serial port via a RS485 converter (*model number: 485SD9TB sold separately*). This converter will convert the RS232 voltage signals to the compatible RS485 differential signals. Only one converter box is needed per serial port. Contact the factory or use the website [www.anaheimautomation.com](http://www.anaheimautomation.com).



## Terminating Resistor

To eliminate noise on the transmission lines or when using a 4000 ft. cable, a terminating resistor is suggested. If used, the termination resistor need only be added to the last (furthest from the RS485 converter box) 34MDSI Series motor/driver/controller on the network between pins A(-) and B(+) on the terminal block. The value of this resistor should be 120 ohms.

## Axis Selection

Each 34MDSI Series motor/driver/controller is addressed using a programmable register allowing the PC to address up to 99 units from one serial port. The Default axis is “0”. To change the axis, use the SMPG-SMSI software or the “~” command. To verify or check the axis, use the software or the “%” command. The axis designation is nonvolatile and will remain the same until changed by the user.

## Motion Profiles and Running the Indexer

The programmable simple indexer has the ability to store two different motion profiles. The profiles include a base speed (starting speed), a maximum speeds (running speed), acceleration/deceleration, and a number of steps (index) number. The base speed has a range of 1-5000 Hz, the maximum speed has a range of 1 Hz-50 kHz, the acceleration/deceleration has a range of 100 to 9,999,999 steps/sec<sup>2</sup> and the index has a range of 0-8,388,607 steps.

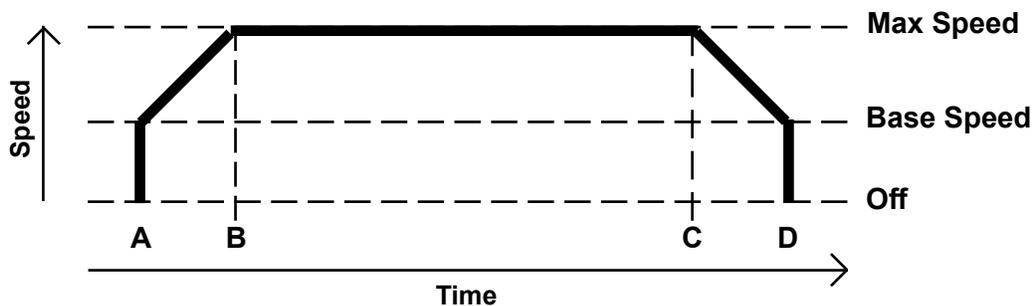
There are “two ways” to get the unit to index: Examples are shown on the next page.

1) The first way is to directly control it from the computer with the software provided. Once the motion profiles have been set, the unit is ready to index. From the software, the user can select which profile to run and then click the begin index button. If a soft limit command is sent or a soft limit input is activated, then the unit will immediately ramp down to base speed and continue running until the index has completed. To stop all motion before the index has completed, a hard limit command needs to be sent or a hard limit input must be activated.

2) The second way to make the unit run is to activate the individual index input. When an index input is activated, it selects from memory the speed profile that was selected. If a soft limit input is activated, then the unit will immediately ramp down to base speed and continue running until the index has completed. To stop all motion before the index has completed a hard limit input must be activated.

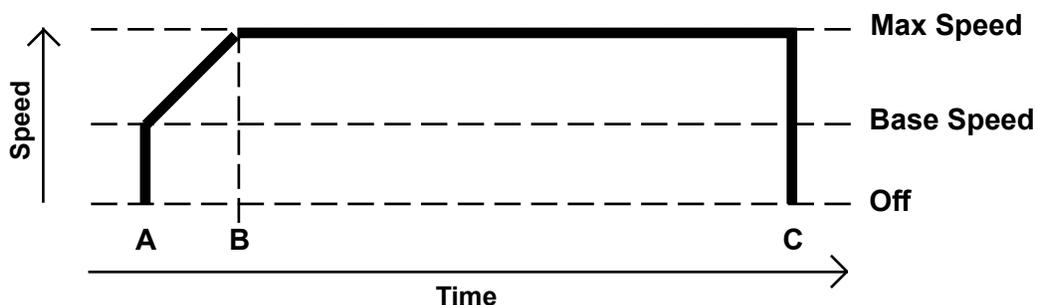
## Normal Operation

- A) A Profile input is activated; Ramps up to max speed.
- B) No limit is active; Max speed is reached (keeps running at max speed).
- C) No limit is active; Internal counter signals to ramp down to base speed.
- D) Internal Counter is complete; Base speed is reached. Pulses stop. Complete Output is set.



## Hard Limit Activated During Index

- A) A Profile input is activated; Ramps up to max speed.
- B) No limit is active; Max speed is reached (keeps running at max speed).
- C) Hard limit is activated; Pulses stop. Complete Output is set.



## Functions

**Hard Limit Switches:** When a hard limit switch is encountered, the pulses will stop. Hard limits are intended as either a home position or an emergency stop for your system.

**Profile Inputs:** These inputs are used to select and begin the specified motion profile. Only 1 input should be activated at a time.

**Motion Profiles:** The indexer will accept up to two different speed or motion profiles. Each profile has its own programmable Accel/Decel, Base speed, Max speed, Index distance value and Complete time. These values are stored in EEPROM for stand-alone use and must be programmed before the indexer is run.

**Acceleration/Deceleration:** The acceleration and deceleration are by default the same value. This function controls the time that the motor will take to move from base speed to max speed. The higher the value, the faster the motor will accelerate. The same principle applies for the deceleration which is controlling the time it takes to go from maximum speed to base speed. The higher the value, the faster the pulses will decelerate. The different accel/decel profiles are stored in EEPROM for stand-alone use. (Range: 100 to 9,999,999)

**Base Speed:** The base speed is the speed at which motion starts and stops. It is entered directly as the number of steps per second. This speed must always be less than the max speed. The different base profiles are stored in EEPROM for stand-alone use. (Range: 1 to 5000)

**Max Speed:** The max speed is the top speed the user wants the pulses to run at. This speed must always be equal or greater than the base speed. It is entered directly as the number of steps/second. The different max profiles are stored in EEPROM for stand-alone use. (Range: 1 to 50,000)

**Index Number:** The index number is the number of steps that the motor will take when the motion profile starts. It is entered directly as the number of steps to take. The different index profiles are stored in EEPROM for stand-alone use. (Range: 0 to 8,388,607)

**Direction Input:** If this input is open then the unit will be running in the clockwise direction. If this input is active then the unit will be running in the counterclockwise direction. This pin can be overridden by the programmable software direction. It will activate the direction output when the pin is changed. This means, that if you start the profile from the software, the unit will look at the software direction. If you start the indexer from the inputs, then the unit will look at the direction input.

**Complete Output:** This is an open collector output that is capable of sinking 50mA. It is current sinking when the indexer finishes its index cycle for the pre-programmed amount of time. The time is entered as microseconds. (Range: 1 to 1000)

## SMPG-SMSI Software

The SMPG-SMSI software is a handy utility that supports Anaheim Automation's programmable pulse generators and simple indexers. Connecting your PC to the 34MDSI Series stepper motor/driver/controller, via RS485, the software can easily perform the following tasks:

- Exercise and monitor the 34MDSI Series stepper motor/driver/controller
- Directly communicate with the 34MDSI Series stepper motor/driver/controller

## Installation

### Software

- SMPG-SMSI software is compatible with all versions of Windows including Windows 8, Windows 7, Windows Vista, and Windows XP
- The SMPG-SMSI software is available for download online on our website menu under Download Software. The link to this web page is provided below:  
<http://www.anaheimautomation.com/downloads/software/software.php>

### Windows 8/7/XP/Vista Installation

1. <http://www.anaheimautomation.com/downloads/software/software.php>
2. Scroll down to SMPG-SMSI Program and Download the .zip file by clicking on the download link. Choose a download destination folder and download the file.
3. Extract the SMPG-SMSI folder.
4. Enter the folder and double click setup.exe file and follow the prompts to install the software.

## Getting Started

- 1) Double click on the SMPG-SMSI icon to run the software.
- 2) Apply power to the 34MDSI unit.
- 3) Set the appropriate communication setting by selecting Setup | Communication Setting from the menu bar.
- 4) Establish communications with the 34MDSI by clicking on the Connect Icon, or select Setup | Connect. If the unit is connected properly, the program will notify you when communication has been established and the correct programming tab will be enabled to let you work with the unit.

## “The Unit is Connected” / “The Unit is NOT Connected”

On the right of the Toolbar, the user will find the communication status of the pulse generator. If communications is not established, please refer to the troubleshooting section.



## File Menu



## Setup Menu

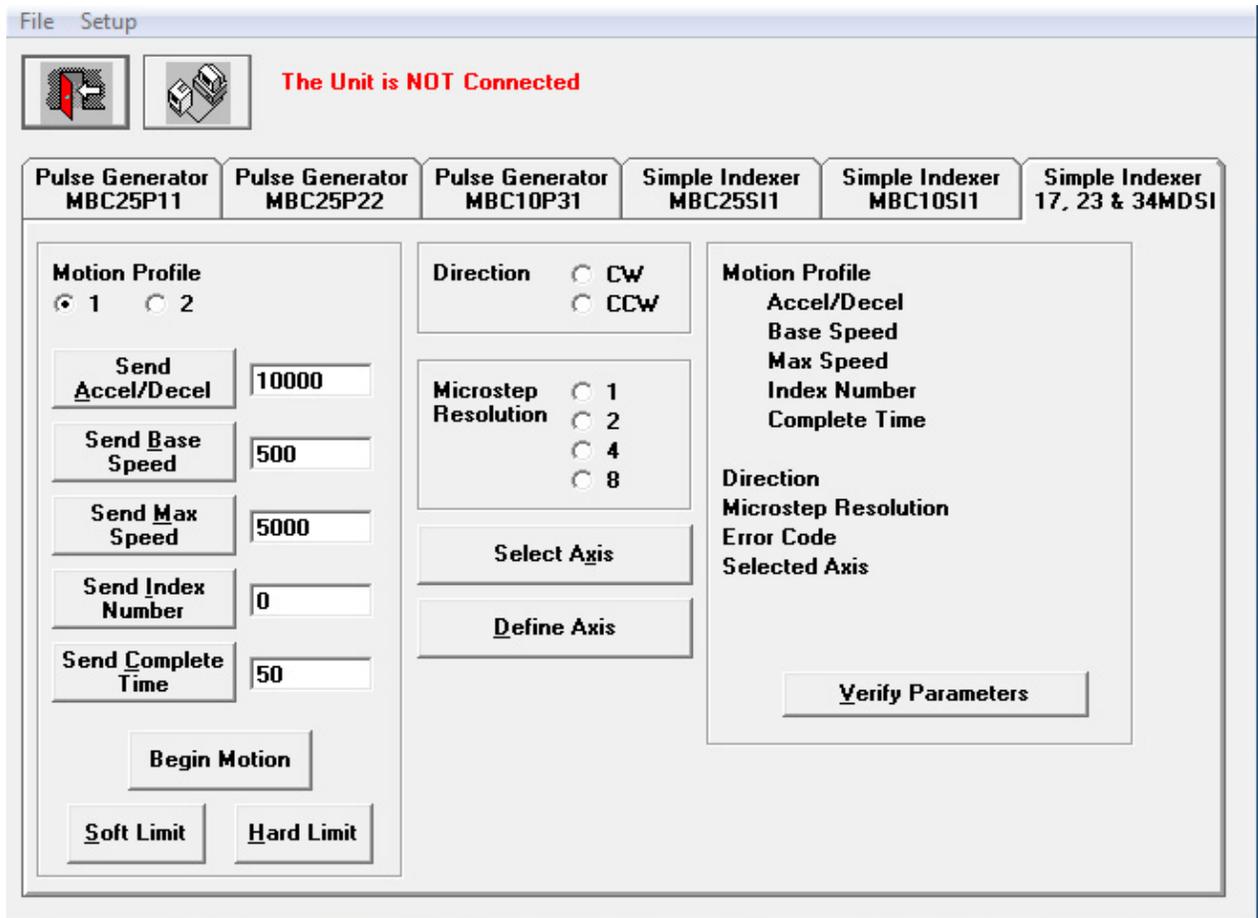


Exit	Exit the SMPG10WIN software
Connect	Establish communications with the controller.
Disconnect	Discontinue communications and release the comport for use by other devices.
Communication Settings...	COM port Selection (Ports 1, 2, 3, 4, 5, 6 or 7)

## Toolbar



# Program Window



Motion Profile	Select motion profiles 1 or 2.
Send Accel/Decel	Send the acceleration & deceleration parameter to the indexer. (steps/sec)
Send Base Speed	Send the base speed parameter to the indexer. (steps/sec)
Send Max Speed	Send the maximum speed parameter to the indexer. (steps/sec)
Send Index Number	Send the index number parameter to the indexer. (steps)
Send Complete Time	Send the complete time parameter to the indexer. (useconds)
Begin Index	Motor will ramp up to maximum speed and keep moving until a limit switch is triggered or the index is complete.
Soft Limit	Motor will ramp down to base speed and continue running until the index is complete.
Hard Limit	Stop any motor motion.
Direction	Set direction to clockwise or counter-clockwise.
Select Axis	Set the axis in the software for which 34MDSI unit to communicate with
Define Axis	Sets the axis in the 34MDSI.
Verify Parameters	Updates and displays controllers parameters and resets the error codes.

## Direct Talk Mode

Direct mode is used to directly control the motion for real time movements through serial communication. This mode is used mainly to program the parameters and for checking the movement of the motor. The 34MDSI Series has 16 commands which are easy to remember for direct movement of a step motor.

## COM Port Settings

**Baud Rate:** 38400  
**Parity:** None  
**Data Bits:** 8  
**Stop Bits:** 1  
**Flow Control:** Xon/Xoff

## Unit Selection

In order to select a unit the @ command followed by 0 (address of the unit) must be sent.

NOTE: There should be no spaces between the @ and the 0.

### How to select the unit:

@0 (Unit is selected)

### How to get a response from the unit:

@0\$ (Carriage Return)

After the \$ command, the pulse generator will return a SMSI30 + the current version number.

Note: In direct talk mode each command is followed by a carriage return.

The unit communicates in half duplex mode, therefore proper setup of hyper terminal is necessary to view characters, if characters are to be echoed back to the screen.

## Instructions

All instructions require that no spaces be sent between the command and the parameter followed by a carriage return. The commands are also case sensitive and are all sent as capitals.

### Command Summary:

**A** - Acceleration/Deceleration  
**B** - Base Speed  
**G** - Go (Index)  
**H** - Hard Limit  
**M** - Max Speed  
**N** - Index Number  
**S** - Soft Limit

**T** - Complete Time  
**V** - Verify  
**+** - Clockwise Direction  
**-** - Counterclockwise Direction  
**\$** - Version Number Register  
**!** - Error Codes Register  
**%** - Verify Controller Address  
**~** - Set Controller Address

## **\$ - Version Number Register**

Format:        \$

Description:   This command requests the 34MDSI to return the version number.

## **! - Error Codes Register**

Format:        !

Description:   This command requests the 34MDSI to get the current error code and print it to the screen.

## **+/- - Direction**

Format:        + or -

Description:   This command sets the direction output. A “+” sets the output to clockwise, and a “-” sets the output to counterclockwise. This must be done when the indexer is not busy. This value is saved in the EEPROM for stand-alone use.

## **A - Acceleration/Deceleration**

Format:        A#\_[value] - where # is the motion profile number 1 or 2

Sample:        A1\_10000     Accel of profile 1 equals 10000

Description:   This command sets the acceleration profile which can be an integer value between 100 and 9,999,999. These values are saved in the EEPROM for stand-alone use.

Range:         100 - 9,999,999

## **B - Base Speed**

Format:        B#\_[value] - where # is the motion profile number 1 or 2

Sample:        B2\_500        Base Speed of profile 2 equals 500

Description:   This command sets the base (start) speed for motion. This value must be set before motion begins and be less than the maximum speed. The pulses will ramp down to this speed after a soft limit is triggered and run at this speed until a hard limit is triggered or the index has finished. These values are saved in the EEPROM for stand-alone use.

Range:         1 - 5000

## **G - Go Index (Run)**

Format:        G# - where # is the speed profile number 1 or 2

Description:   This command will send clocks out to the pulse generator. The only command that can stop the clocks is H (stop motion). The S (soft limit) command will make the pulses go from max speed to base speed. Motion can also be stopped by using the limit switch inputs. The ramp profile is specified by the B (base speed), M (max speed), and A (acceleration/deceleration) commands.

## H - Hard Limit or Stop Motion

Format: H

Description: This command will stop all motion. It can only be used when pulses are running.

## M - Max Speed

Format: M#\_[value] - where # is the motion profile number 1 or 2

Sample: M2\_10000 Max Speed of profile 2 equals 10000

Description: This command sets the maximum (running) speed for motion. This value must be set before motion begins and be equal or greater than the base speed. The motor will run at this speed until a soft limit or a hard limit is triggered. These values are saved in the EEPROM for stand-alone use.

Range: 1 - 50,000

## N - Index Number

Format: N#\_[value] - where # is the motion profile number 1 or 2

Sample: N1\_10000 Index number of profile 1 equals 10000

Description: This command sets the number of pulses to index for the motion profile. This value must be set before motion begins. These values are saved in the EEPROM for stand-alone use.

Range: 0 - 8,388,607

## S - Stop Soft

Format: S

Description: This command will cause the indexer to ramp down to base speed and run until the index is complete or a hard limit is activated. It can only be used when pulses are running.

## T - Complete Time

**Format:** T#[value] - where # is the motion profile number 1 or 2

**Sample:** T2\_100 Complete time of profile 2 equals 100 uSeconds

**Description:** This command sets the time for an active complete signal after the unit has finished indexing for the motion profile. The number is entered as microsecond. This value must be set before motion begins. These values are saved in the EEPROM for standalone use.

**Range:** 0 - 1000

## V - Verify

**Description:** This command can be used with most commands to verify the register contents. This is a read only command. Valid Commands are: A, B, D, M, N, R, T and +.

**Format:** V[command]

This format is good for R and +.

**R -** If a 1 is sent back then the driver is in Full step mode. If a 2 is sent back then the driver is in Half step mode. If a 4 is sent back then the driver is in Quarter step mode. If an 8 is sent back then the driver is in Eighth step mode.

**+ -** If a 1 is sent back then the direction is clockwise. If a 0 is sent back then the direction is counterclockwise.

**Sample:** V+ Verification of Direction is prompted.

**Format:** V[command]# - where # is the speed profile number 1 or 2

This format is good for A, B, M, N and T.

**A# -** Verify Acceleration/Deceleration for given speed profile.

**B# -** Verify Base speed for given speed profile.

**M# -** Verify Max speed for given speed profile.

**N# -** Verify Index number for given speed profile.

**T# -** Verify Complete Time for given speed profile.

## % - Verify address register

**Format:** % (No address is needed before this function. @% will return the address)

**Description:** This command requests the 34MDSI Series controller to return its internal address number to the PC or PLC.

## ~ - Set address register

**Format:** ~[value] (No address is needed before this function. @~[value] will set the address)

**Description:** This command sets the address for communication inside the 34MDSI Series controller.

**Range:** 0 - 99

## Visual Basic Direct Mode Programming Examples

### Example 1: This Example is for Axis=0, and Profile=1

```
Dim Const DefaultTimeout As Single = 0.5
frmMain.MSComm1.Output = "@0A1_100000" & Chr$(13) 'Set Acceleration
Pause DefaultTimeout
frmMain.MSComm1.Output = "@0B1_1000" & Chr$(13) 'Set Base Speed
Pause DefaultTimeout
frmMain.MSComm1.Output = "@0M1_4000" & Chr$(13) 'Set Maximum Speed
Pause DefaultTimeout
frmMain.MSComm1.Output = "@0N1_500" & Chr$(13) 'Set Index Number
Pause DefaultTimeout
frmMain.MSComm1.Output = "@0T1_100" & Chr$(13) 'Set Complete Time
Pause DefaultTimeout
frmMain.MSComm1.Output = "@0R8" & Chr$(13) 'Set Microstep Resolution
Pause DefaultTimeout
frmMain.MSComm1.Output = "@0+" & Chr$(13) 'Set Direction CW
Pause DefaultTimeout
frmMain.MSComm1.Output = "@0G1" & Chr$(13) 'Start the Index
Pause DefaultTimeout
```

### Example 2: This Example receives the error code and Version Number from Axis3

```
Dim Const DefaultTimeout As Single = 0.5
frmMain.MSComm1.Output = "@3!" & Chr$(13)
Pause ShortTimeOut
MsgBox ReceiveAscii()
frmMain.MSComm1.Output = "@3$" & Chr$(13)
Pause ShortTimeOut
MsgBox ReceiveAscii()
```

Function ReceiveAscii() As String

'wait for the incoming data to get to the buffer

Dim BeginTime As Single

Dim A As Integer

Dim B As Integer

BeginTime = Timer

While Main.MSComm1.InBufferCount = 0

DoEvents

If Timer - BeginTime > 0.1 Then GoTo ExitRoutine1

Wend

A = 0: B = Main.MSComm1.InBufferCount

While A <> B

A = Main.MSComm1.InBufferCount

Pause 0.02

' Fixed time value

B = Main.MSComm1.InBufferCount

Wend

ReceiveAscii = Main.MSComm1.Input

Exit Function

## Troubleshooting

### Problem:

*Can not establish communications with the indexer.*

### Possible Solutions:

- 1) Make sure the indexer has power. Is the Green LED on.
- 2) Check RS232 connections.
- 3) Check for loose cable connection either on the pulse generator or COM Port.
- 4) Was the software installed successfully?
- 5) Go to **Setup | Communication Settings** and verify COM port settings.
- 6) Click on **Connect** icon to communicate with the indexer.
- 9) If problems still exist, contact Anaheim Automation Tech Support.

### Problem:

*There is no power to the indexer.*

### Possible Solutions:

- 1) Is the indexer connected to the appropriate power supply?
- 2) Check for any blown fuses in line with the indexer.
- 3) If problems still exist, contact Anaheim Automation at 714-992-6990.

### Problem:

*The indexer has a fault condition.*

### Possible Solutions:

- 1) To clear an error use either the SMPG-SMSI software or the direct mode command.
- 2) The SMPG-SMSI software can clear an error in the motion tab section by clicking on the Verify Parameters button.
- 3) The direct mode command “!” can clear an error by prompting indexer to serially send the error code back to the user.

Example: @0! (carriage return)

Description: Address the unit by typing @ followed by a 0 (address number) an ! (Error Codes Register) and a carriage return.

Note: The error code is returned in binary coded decimal format. If two errors were received their binary values would be added together.

## Error Codes

1	Recieve Over-flow Error	The serial communications had a recieving error. This is an internal error caused by the computer.
2	Range Error	There was an invalid number of characters sent to the pulse generator. Check to see if the parameters are invalid for the command that was sent.
4	Command Error	A bad command was sent to the pulse generator. Please check to see that the command being sent is valid, or that the pulse generator is not running.
8	Transmitt Error	To many parameters sent back to the PC. This is an internal error caused by the eeprom.
16	Motor Error	Motor speed profiles are set incorrectly. Please make sure that the base speed is less than the max speed and that the speeds are within their valid ranges.
32	Zero Parameters Error	There were no parameters sent to the pulse generator. A command was sent to the pulse generator that expected to see parameters after the command.

## ASCII Table for Direct Mode

ASCII Symbol	Hex Value	ASCII Symbol	Hex Value
Carriage Return	0D	H	48
0	30	M	4D
1	31	N	4E
2	32	R	52
3	33	S	53
4	34	T	54
5	35	V	56
6	36	!	21
7	37	\$	24
8	38	%	25
9	39	+	2B
A	41	-	2D
B	42	_	5F
G	47	~	7E

## Multiple Axes Setup for 34MDSI Series

1. Power up the 1st unit **ONLY** and connect the RS485 connection to the 1st unit.
2. Click the "Define Axis" button and enter 0.
3. Click the "Select Axis" button and enter 0.
4. Under Setup menu click "Connect".
5. Text appears on top of the SMPG-SMSI Software "The Unit is Connected" and the version of the firmware is displayed as well. This text should turn blue when connected.
6. Under Setup menu click "Disconnect".
7. Text appears on top of the SMPG-SMSI Software "The Unit is NOT Connected". This text should turn red when disconnected.
8. POWER DOWN and Physically disconnect the 1st Unit from both power and RS485 Connections.
9. Power up the 2nd unit **ONLY** and connect the RS485 connection to the 2nd unit.
10. Click the "Define Axis" button and enter 1.
11. Click the "Select Axis" button and enter 1.
12. Under Setup menu click "Connect".
13. Text appears on top of the SMPG-SMSI Software "The Unit is Connected" and the version of the firmware is displayed as well. This text should turn blue when connected.
14. Under Setup menu click "Disconnect".
15. Text appears on top of the SMPG-SMSI Software "The Unit is NOT Connected". This text should turn red when disconnected.
16. POWER DOWN and Physically disconnect the 2nd Unit from both power and RS485 Connections.
17. For more then two units, continue steps 9 thru 16 incrementing the axis number for each unit.
18. Now reconnect the power and RS485 connections to **ALL** units.
19. Power up **ALL** Units.
20. You should now be able to connect individually to any Axis with the RS485 converter from one Serial Port. Use the "Select Axis" button to connect to each individual axis.

**Copyright**

Copyright 2005 by Anaheim Automation. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of Anaheim Automation, 4985 E Landon Drive, Anaheim, CA 92807. The only exception to this would be use of the program examples in this manual.

**Disclaimer**

Though every effort has been made to supply complete and accurate information in this manual, the contents are subject to change without notice or obligation to inform the buyer. In no event will Anaheim Automation be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation.

**Limited Warranty**

All Anaheim Automation products are warranted against defects in workmanship, materials and construction, when used under Normal Operating Conditions and when used in accordance with specifications. This warranty shall be in effect for a period of twelve months from the date of purchase or eighteen months from the date of manufacture, whichever comes first. Warranty provisions may be voided if the products are subjected to physical damage or abuse.

Anaheim Automation will repair or replace at its option, any of its products which have been found to be defective and are within the warranty period, provided that the item is shipped freight prepaid, with RMA (return material authorization), to Anaheim Automation's plant in Anaheim, California.

**Trademarks**

Control Link and Driver Pack are registered trademarks of Anaheim Automation.

IBM PC is a registered trademark of International Business Machines, Inc.

***ANAHEIM AUTOMATION***