9904 Series 2

User Manual

List of Tables

Introduction

The 9904 Series 2 controller is a micro-stepping indexer with 2.5A (peak) stepper motor driver. In addition to standard indexing applications, the 9904 Series 2 supports speed control, as well as support for “on the fly” motion parameter changes. This is different from previous models that only supported single move indexing.

Driver Features

Microstepping

Supports micro-stepping modes from full-step to 256x micro-stepping. Full stepping is a high-torque mode where both coils are always set to the maximum rms current in both directions with no zero current states.

Automatic Standby Current

Independently settable hold and run currents to reduce power consumption and heating heating.

Mid-Band and Low Speed Resonance Compensation

Resonances are automatically eliminated when using Net Controls Motors.

Protection

The MOSFET drivers are protected against short to ground and motor coil short circuit. Additional protection against ESD events and cable break.

NEVER HOT-PLUG MOTORS/CABLES. ALL CONNECTIONS SHOULD BE MADE BEFORE POWER IS APPLIED TO THE UNIT.

Power

The 9904 Series 2 is equipped with a 24-volt, 350W (200W Continuous) power supply, supporting 1-3 axis configurations. Convenient rear panel accessible 5A/250V fuse.

Indexer Features

Highly Configurable Motion Controller/Indexer

All motion parameters can be changed on the fly for maximum flexibility. Each axis has independent motion controllers, with up to 1MHz step rates.

Optional Touchscreen Interface

All 9904 Series 2 controllers can be ordered with a 3.5” for 4.3” touchscreen for manual control control or monitoring.

Intuitive ASCII Command Interface (RS485)

Command response architecture using printable ASCII commands make interfacing easy. Each axis is addressable. Global commands can be used to synchronize axis moves via set-points.

User Definable Set-Points

Each axis has 10 user-definable set-points for automated positioning. Each set-point has configurable acceleration and speed.

Full-Step (High Torque) Morphing

The 9904 Series 2 controllers feature a configurable step morphing mode to increase torque at higher speeds. When configured, the driver will automatically switch to square-wave full step mode and back to micro-stepping without step loss. This is completely transparent to the user motion profile.

Inputs/Outputs

4 inputs and 2 outputs on each axis. Inputs are configurable for rising or falling edges and can stop the motor on active signals. Outputs are configurable for motion diagnostics or can be set via ASCII commands.

Encoder Interface

For encoder-equipped motors, missed-step and stall detection algorithms.

Cable and Motor Compatibility

All controllers are compatible with any hybrid stepper motor. However, some functionality may be lost if encoders are not installed or the motor characteristics are not a match for the motor drivers. Below motors can be used as a guide in selecting compatible motors.

FIRE OR BURN HAZARD. ALWAYS CHECK THE CURRENT SETTINGS AND MOTOR SPECIFICATIONS PRIOR TO INITIALIZING THE AXIS. IF IMPROPERLY SET, MOTORS CAN GET EXTREMELY HOT!

The following motor and cables are compatible with the 9904 Series 2 Controllers:

Table 1: Compatible NEMA 34 Motors

Type	Length (L)	Shaft Dia. (D)	Rated Current (A)	Phase Resistance (Ω)	Inductance (mH)	Holding Torque (N. cm)	Detent Torque (N. cm)	Rotor Inertia (g. $cm^{2)}$	Motor Weight (lbs)
-2001	66mm	3/8"	2.0	2.9	6.6	310	5.5	850	4.4
-2701	66mm	3/8"	2.7	1.2	3.5	310	5.5	850	4.4
-2702	98mm	14mm	2.7	1.9	8.6	691	9.5	1550	6.6
-2703	126mm	14mm	2.7	2.5	12.5	902	18.5	2200	9.9
-2735	150mm	14mm	2.7	2.9	17.5	1198	24.5	2500	11

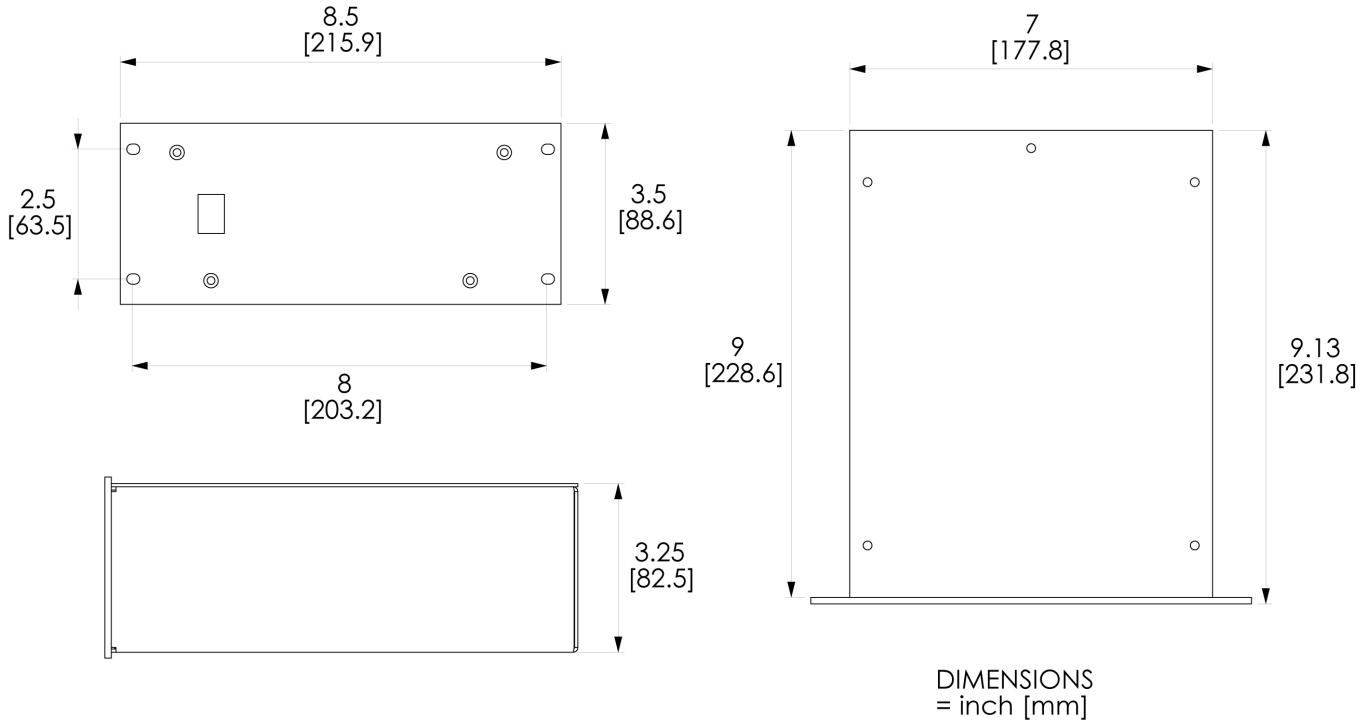
Table 2: Compatible NEMA 23 Motors

Type	Length (L)	Rated Current (A)	Phase Resistance (Ω)	Inductance (mH)	Holding Torque (N. cm)	Detent Torque (N. cm)	Rotor Inertia (g. $cm^{2)}$	Motor Weight
-4412	45mm	0.62	12	26	80	2.8	190	520g
-6415	56mm	1.5	3.6	13.8	110	3.5	280	680g
-6425	56mm	2.5	1.2	4.8	110	3.5	280	680g
-7425	64mm	2.5	1.5	4.5	150	5	380	850g
-8415	76mm	1.5	3.9	14.1	180	6	440	1050g
-8425	76mm	2.5	1.8	6.5	180	6	440	1050g

Table 3: Compatible Cables

Part Number	Description	Compatible Motors
71-08-0904	Cable, Motor Drive, 8'	80-xx-xxxx
71-08-0912	Cable, Motor, 8', w/ Encoder interface	82-xx-xxxx/83-xx-xxxx

Dimensions



Ordering Guide

Part Number	Description	Compatibility
91-01-9904	Controller, Single-Axis, 90-264VAC @ 125W, RS485	80, 82, 83 Series Motors
91-02-9904	Controller, Dual-Axis, 90-264VAC @ 200W, RS485	80, 82, 83 Series Motors
91-03-9904	Controller, Triple-Axis, 90-264VAC @ 200W, RS485	80, 82, 83 Series Motors

Part Number	Description	Compatibility
92-01-9904	Controller, Single-Axis, 90-264VAC @ 125W, RS485 w/Touch	80, 82, 83 Series Motors
92-02-9904	Controller, Dual-Axis, 90-264VAC @ 200W, RS485 w/Touch	80, 82, 83 Series Motors
92-03-9904	Controller, Triple-Axis, 90-264VAC @ 200W, RS485 w/Touch	80, 82, 83 Series Motors

Overview

Rear Panel Connectors

Table 4: Motor - DSUB-9

1 - Motor Phase A+	6 - Encoder Index
2 - Motor Phase A-	7 - Encoder Ch A
3 - Motor Phase B-	8 - Encoder Ch B
4 - Motor Phase B+	9 - Encoder +5VDC
5 - Encoder GND	

Table 5: I/O - DSUB-15

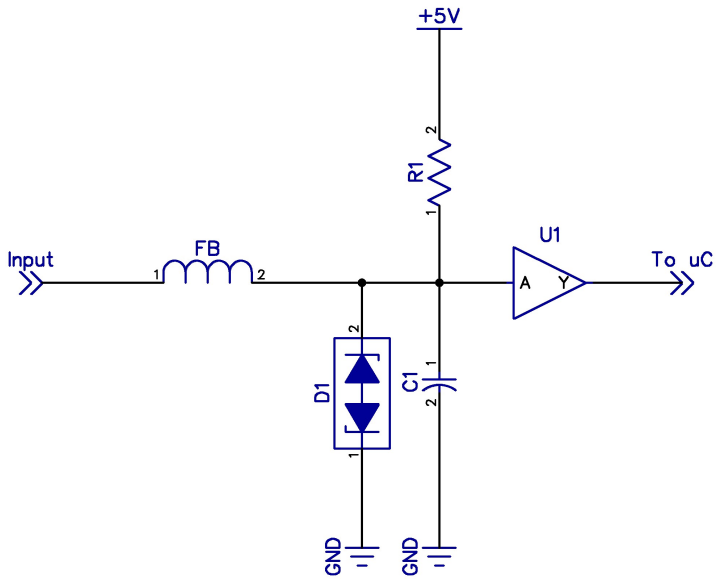
1 - +5 VDC Supply (100mA)	9 - +5 VDC Supply (100mA)
2 - Digital GND	10 - Digital GND
3 - Output 1 (Collector)	11 - Output 1 (Common/Emitter)
4 - Output 2 (Collector)	12 - Output 2 (Common/Emitter)
5 - Input 1 (+5 VDC Max)	13 - Digital GND
6 - Input 2 (+5 VDC Max)	14 - Digital GND
7 - Input 3 (+5 VDC Max)	15 - Chassis GND
8 - Input 4 (+5 VDC Max)	

Table 6: RS485 Serial Communications - DSUB-9

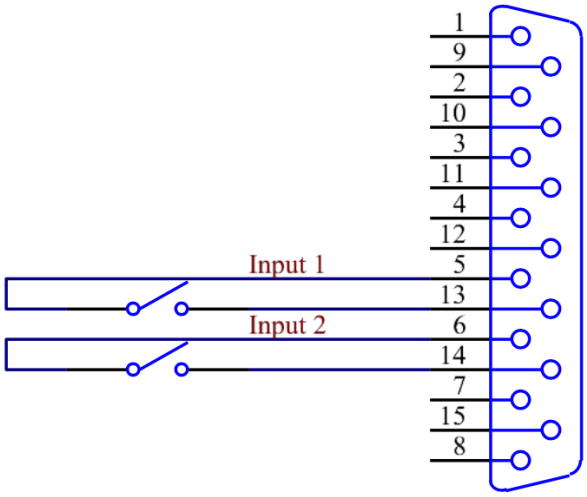
1 - RS485 B (Data -)	6 - N/C
2 - RS485 A (Data +)	7 - N/C
3 - N/C	8 - N/C
4 - N/C	9 - N/C
5 - Digital GND	

Input Circuit Diagram/Characteristics

All inputs are rated at 5V and are pulled up via a 10k resistor.

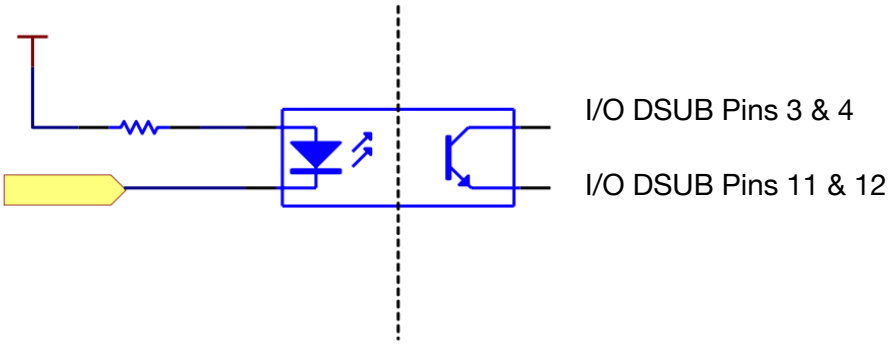


As a result, limit switches can be directly connected to the I/O connector using ground pins 13 and 14.



Output Circuits

Outputs are opto-isolators capable of driving 24V loads at 50mA. Each Axis has 2 outputs with independent connections for the collectors and emitters. Thus, each output can be independently configured for sink or sourcing applications.



Default Settings (Non-Volatile Memory NVM)

Motor/Motion Default Settings

All default settings are stored in non-volatile memory (NVM) and are loaded on power up. For driver-related parameters, changes are only re-loaded upon initialization (see Initialization Command). For motion-related parameters, power cycling the controller is required.

FIRE OR BURN HAZARD. ALWAYS CHECK THE CURRENT SETTINGS AND MOTOR SPECIFICATIONS PRIOR TO INITIALIZING THE AXIS. IF IMPROPERLY SET, MOTORS CAN GET EXTREMELY HOT!

Table 7: NVM Defaults - Motor/Motion Parameters

Defaults	Description	Setting
Run Current	Peak motor current during motion	16
Hold Current	Peak motor current during stand still	2
Motor Acceleration	Acceleration/Decleration in steps/sec2	5000
Motor Velocity (Max)	Max speed in steps/sec	10000
Current Range	High or Low Current Range	1
CW/CCW Direction	CW or CCW when position is increasing	CW
Encoder CPR	Counts/rev of encoder if equipped	800
Loading Error	Allowable Error when using encoder	12800
Initialization Loading Error	Separate Load Error when Homing the motor	32
Micro Step Resolution	Micro steps/step	256
Step Morphing	Speed to change to Torque Mode Full Stepping	6

NOTE: MOST CHANGES TO DEFAULT PARAMETERS REQUIRE POWER CYCLING.

I/O Default Settings

All I/O default settings are stored in non-volatile memory (NVM) and are loaded on power up only. Changes to configuration require a controller power cycle to properly load.

Table 8: NVM Defaults - I/O Parameters

Defaults	Description	Setting
Input 1	Input Pin configuration	3
Input 2	Input Pin configuration	3
Input 3	Input Pin configuration	3
Input 4	Input Pin configuration	3
Encoder Index	Encoder Index Configuration	1
Output 1	Output Configuration	1
Output 2	Output Configuration	1

Table 9: Input Configuration

Setting	Trigger	Function
0	Not Present	Disabled
1	Interrupt on Rising Edge	Motor Hard Stop (Use for Homing)
2	Interrupt on Rising Edge	Motor Soft Stop
3	Interrupt on Falling Edge	Motor Hard Stop (Use for Homing)
4	Interrupt on Falling Edge	Motor Soft Stop

Table 10: Output Configuration

Setting	Function
0	User Defined - ASCII Commands
1	Motor Error
2	Motor Active (In Motion)
3	Motor Stopped (Idle)

System Default Settings

System Settings are used to configure communications and compatibility modes. These setting require power cycling the controller.

Table 11: NVM Defaults - System Parameters

Defaults	Description	Setting
Baud	System Baud Rate Configuration	0
Homing	Homing Compatability Mode	1
I/O	I/O Compatability Mode	1
Address	Axis address in HEX	Factory Preset

ASCII Serial Command Interface

All 9904 Series 2 controllers serial user interface based based on the RS485 physical Layer specification. This RS485 interface operates in half-duplex mode, or a standard two-wire interface. All commands are printable ASCII characters for ease of development.

NOTE: DEFAULT SERIAL PORT PARAMETERS: 38400, 8, N, 1. THE BAUD RATE IS CONFIGURABLE FROM 9600 - 115200. THERE IS ALSO A SPECIAL 250K BAUD MODE FOR INTER-CONNECTED AXIS AND IS AVAILABLE TO THE USER FOR SHORT, HIGH-QUALITY, WIRING LESS THAN 10'.

Command Structure - Get Commands

: a c CR

Where:

:	Start of Command
a	Axis Address
c	ASCII Command
CR	Carriage Return, displayed as '\r'

Example:

To read Axis 1 position:

:1p\r

Controller responds with:

:1p12800\r

Command Structure - Set Commands

: a c n CR

Where:

:	Start of Command
a	Axis Address (ASCII HEX)
c	ASCII Command
n	Option or Value
CR	Carriage Return, displayed as '\r'

Example:

To send Axis 1 motor to position 12800:

:1p12800\r

Controller does not respond.

Unavailable Commands or Command Errors

When a command is either unavailable or is entered in error, the controller will respond with the input command followed by an exclamation point (!) and CR

Example:

:1x\r

Controller responds with:

:1x!\r

Special Command Responses

Some command responses are enabled to report success of certain actions. These include NVM save status and Address unlock status.

:1k\r (Address unlock requested)
:1k>>\r (Controller completed unlock sequence)

:1Q\r (Update Flash with new NVM data)
:1>\r (NVM parameters updated and stored successfully)

Global Commands

Global commands are supported for various tasks. The following global commands are supported:

Command	Notes
Homing	All Axis must be configured to home to the same I/O
Execute Set Point	All set points must be configured for each Axis
STOP/HALT	Hard or Soft stop of all motors
Initialize	All Axis initialized to same value
Address Unlock*	Factory or single Axis use
Address Set*	Factory or single Axis use
FW Update**	Ready update and Silent mode for non-updating Axis

***NOTE: ADDRESS UNLOCK AND SET SHOULD NEVER BE USED IN MULTI-AXIS CONTROLLERS. AXIS ADDRESSES ARE PRESET AT THE FACTORY AND SHOULD NOT BE CHANGED**

****NOTE: FW UPDATES ARE DONE USING THE GLOBAL ADDRESS (ADDRESS 0) FOLLOWED BY THE AXIS TO UPDATE. THIS WAY, ALL AXIS RECEIVE THE COMMAND AND GO INTO SILENT MODE UNTIL FW UPDATE IS COMPLETE AND POWER IS CYCLED**

ASCII Serial Commands - NVM Settings

Table 12: NVM ASCII Command Set - Motor/Motion Parameters

Parameter	Address (hex)	ASCII Printable Equivalent	Command
Acceleration	0x41	A	:*abxxxx* where: *a* = Address *b* = 'A' *xxxx* = Acceleration in steps/sec^2
Direction	0x43	C	:*abxxxx* where: *a* = Address *b* = 'C' *x* = 1 = CCW *x* = 0 = CW
Encoder CPR	0x45	E	:*abxxxx* where: *a* = Address *b* = 'E' *xxxx* = Encoder CPR or PPS x4
Holding Current	0x48	H	:*abxxxx* where: *a* = Address *b* = 'H' *x* = 0-31 (31 = 2A/Phase)
Initialization Load Error	0x49	I	:*abxxxx* where: *a* = Address *b* = 'I' *xxx* = Allowable Error before hard stop is detected
Load Error	0x4C	L	:*abxxxx* where: *a* = Address *b* = 'L' *xxxx* = Allowable Following Error before faulting
Microsteps/Step	0x4D	M	:*abxxxx* where: *a* = Address *b* = 'M' *xxx* = 2-256
Current Range Select	0x4F	O	:*abxxxx* where: *a* = Address *b* = 'O' *x* = 0 = High Range (2A) *x* = 1 = Low Range (1A)

Full Step Morphing	0x50	P	:*abs* where: *a* = Address *b* = 'R' x = 0 = Disable Morphing *x* = 1 = 60 RPM transition to full step x = 2 = 120 RPM transition to full step x = 3 = 180 RPM transition to full step x = 4 = 240 RPM transition to full step x = 5 = 300 RPM transition to full step x = 6 = 360 RPM transition to full step x = 7 = 420 RPM transition to full step x = 8 = 480 RPM transition to full step x = 9 = 500 RPM transition to full step
Run Current	0x52	R	:*abxxxx* where: *a* = Address *b* = 'R' *x* = 0-31 (31 = 2A/Phase)
MAX Speed	0x53	S	:*abxxxx* where: *a* = Address *b* = 'S' *xxxx* = Max Speed in Steps/Sec

Setting Run and Hold Current

Using the NVM setting commands listed in Table 12, set up the motor current as follows:

1. Set the current range Low or High.
   a. Example :1O1\r Sets the current range of Axis 1 to Low, or 1A/phase
   b. Example :2O0\r Sets the current range of Axis 2 to High, or 2A/phase
2. Set the Run current to match the motor Max current or less depending on the load conditions.
   a. Example :1R16\r Sets the Run current for Axis 1 to 16, or 1A/phase in High Range
3. Set the Hold Current for power/heat reduction.
   a. Example :2H2\r Sets the Hold current for Axis 2 to 2, or .13A/phase

To calculate motor currents:

Current Setting * 2 / 31 (High Range)
Current Setting * 1 / 31 (Low Range)

Example:

:1R8\r Sets the Run current to 8 * 2 / 31 or .52A/phase in High Range.

Table 13: NVM ASCII Command Set - I/O Parameters

Parameter	Address (hex)	ASCII Printable Equivalent	Command
Out 1 Configuration	0X4A	J	:*abxxxx* where: *a* = Address *b* = 'J' *x* = Config Parameter
Out 2 Configuration	0x4B	K	:*abxxxx* where: *a* = Address *b* = 'K' *x* = Config Parameter
Input 1 Configuration	0x54	T	:*abxxxx* where: *a* = Address *b* = 'T' *x* = Config Parameter
Input 2 Configuration	0x55	U	:*abxxxx* where: *a* = Address *b* = 'U' *x* = Config Parameter
Input 3 Configuration	0x56	V	:*abxxxx* where: *a* = Address *b* = 'V' *x* = Config Parameter
Input 4 Configuration	0x57	W	:*abxxxx* where: *a* = Address *b* = 'W' *x* = Config Parameter
Index Configuration	0x5A	Z	:*abxxxx* where: *a* = Address *b* = 'Z' *x* = Config Parameter

Table 14: NVM ASCII Command Set - System Parameters

Parameter	Address (hex)	ASCII Printable Equivalent	Command
Baud	0x42	B	:*abxxxx* where: *a* = Address *b* = 'B' *x* = 1 = 9600 *x* = 2 = 19200 *x* = 3 = 38400 *x* = 4 = 250k *x* = 5 = 115200
Address	0x44	D	:*abxxxx* where: *a* = Address *b* = 'D' *xxxx* = Device address 1 - F (hex)
Home Compatability	0x46	F	:*abx* where: *a* = Address *b* = 'F' x = 0 = Not Compatible x = 1 = Full compatibility with 9904 series 1 x = 2 = Homing direction compatibility only
I/O Compatability	0x47	G	:*abx* where: *a* = Address *b* = 'G' *x* = 0 = Not Compatible x = 1 = Full compatibility with 9904 series 1
NVM Save	0x51	Q	:*ab* where: *a* = Address *b* = 'Q'
FW Update Mode Set	0x58	X	:*abx* where: *a* = Address *b* = 'X' *x* = Axis to be updated (1 - F)

Table 15: NVM ASCII Command Set - Set Point Parameters

Parameter	Address (hex)	ASCII Printable Equivalent	Command
Set Point 0	0x30	0	:*abxx,yy,zz* where: *a* = Address *b* = '0' *xx* = Position assigned to set-point 0 yy = Velocity assigned to set-point 0 zz = Acceleration assigned to set-point 0
Set Point 1	0x31	1	:*abxx,yy,zz* where: *a* = Address *b* = '1' *xx* = Position assigned to set-point 1 yy = Velocity assigned to set-point 1 zz = Acceleration assigned to set-point 1
Set Point 2	0x32	2	:*abxx,yy,zz* where: *a* = Address *b* = '2' *xx* = Position assigned to set-point 2 yy = Velocity assigned to set-point 2 zz = Acceleration assigned to set-point 2
Set Point 3	0x33	3	:*abxx,yy,zz* where: *a* = Address *b* = '3' *xx* = Position assigned to set-point 3 yy = Velocity assigned to set-point 3 zz = Acceleration assigned to set-point 3
Set Point 4	0x34	4	:*abxx,yy,zz* where: *a* = Address *b* = '4' *xx* = Position assigned to set-point 4 yy = Velocity assigned to set-point 4 zz = Acceleration assigned to set-point 4
Set Point 5	0x35	5	:*abxx,yy,zz* where: *a* = Address *b* = '5' *xx* = Position assigned to set-point 5 yy = Velocity assigned to set-point 5 zz = Acceleration assigned to set-point 5

Set Point 6	0x36	6	:*abxx,yy,zz* where: *a* = Address *b* = '6' *xx* = Position assigned to set-point 6 yy = Velocity assigned to set-point 6 zz = Acceleration assigned to set-point 6
Set Point 7	0x37	7	:*abxx,yy,zz* where: *a* = Address *b* = '7' *xx* = Position assigned to set-point 7 yy = Velocity assigned to set-point 7 zz = Acceleration assigned to set-point 7
Set Point 8	0x38	8	:*abxx,yy,zz* where: *a* = Address *b* = '8' *xx* = Position assigned to set-point 8 yy = Velocity assigned to set-point 8 zz = Acceleration assigned to set-point 8
Set Point 9	0x39	9	:*abxx,yy,zz* where: *a* = Address *b* = '9' *xx* = Position assigned to set-point 9 yy = Velocity assigned to set-point 9 zz = Acceleration assigned to set-point 9

ASCII Serial Commands - Controller Operations

Table 16: ASCII Command Set - Controller Operations

Parameter	Address (hex)	ASCII Printable Equivalent	Command
Acceleration	0x61	a	:*abx* where: *a* = Address *b* = 'a' *x* = Acceleration in steps/sec^2
Following Error	0x62	b	:*abx* where: *a* = Address *b* = 'b' *x* = Following Error expressed as micro-steps relative to the encoder count (ratio)
Home	0x63	c	:*abx* where: *a* = Address *b* = 'c' *x* = 0 = Home to Index - CCW x = 1 = Home to input 1 - CCW x = 2 = Home to input 2 - CW x = 3 = Home to input 3 - CCW x = 4 = Home to input 4 - CW x = 5 = Home to following error - CCW
Execute Set Point	0x64	d	:*abx* where: *a* = Address *b* = 'd' *x*= Set-point Number 0 - 9
Motor Status	0x67	g	:*abxy* where: *a* = Address *b* = 'g' *x* = Motor Status: 0 = Motor Ready 1 = Motor Not Homed 2 = Motor Not Initialized 3 = Motor Error *y* = 1 = Motor Running *y*= 0 = Motor Stopped/finished

Halt/Stop	0x68	h	:*abx* where: *a* = Address *b* = 'h' *x* = 1 = Hard Stop *x*= 2 = Soft Stop
Initialize/Enable	0x69	i	:*abx* where: *a* = Address *b* = 'i' *x* = 1 = Initialize/Reset Driver x = 2 = Disable Driver x = 3 = Enable Driver (does not load parameters)
Jog	0x6A	j	:*abxxxx* where: *a* = Address *b* = 'j' *xxxx* = Number of steps to jog (can be -)
Input Status	0x6C	l	Returns status of all inputs 4 + Index :abvwxyz *a* = Address *b* = 'l' v = Input 1 Status w = Input 2 Status x = Input 3 Status y = Input 4 Status z = Encoder Index Status 1 = High 0 = Low i.e. :*ab*00001 indicates Index is High
Output 1	0x6E	n	:*abx* where: *a* = Address *b* = 'n' *x* = 1 = On *x* = 0 = Off
Output 2	0x6F	o	:*abx* where: *a* = Address *b* = 'o' *x* = 1 = On *x* = 0 = Off
Position	0x70	p	:*abxxxx* where: *a* = Address *b* = 'p' *xxxx* = Position in micro-steps
Current Speed	0x73	s	:*abxxxx* where: *a* = Address *b* = 's' *xxxx* = Speed in steps/sec

Global Velocity	0x76	v	:*abxxxx* where: *a* = Address *b* = 'v' *xxxx* = Max Velocity in steps/sec
Encoder Position	0x79	y	:*abxxxx* where: *a* = Address *b* = 'y' *xxxx* = Encoder Counts (Can be negative)
Software Revision	0x7A	z	:*abxyz* where: *a* = Address *b* = 'z' *xyz* = Series Revision-Date

Controller Operation

Motion Controller

Controller Initialization

Minimum settings/configurations required for motor movement:
1. Current Range Select set to 0/1 for High/Low range
2. Run Current set
3. Hold Current set
4. Acceleration (or default acceleration) set
5. Max Speed (or default max speed) set
6. Optional encoder CPR setting if motor is encoder-equipped
7. Optional Load Error Setting if motor is encoder-equipped

Once the minimum settings are set, the controller is ready for the initialization sequence. This sequence moves the motor slightly and calibrates the encoder (if equipped). During the initialization sequence, the following actions are completed:

1. If homing is active, the controller will immediately kill the homing task
2. The Stepper driver is initialized/re-initalialzed
   a. Run/Hold Current is loaded/re-loaded
   b. Current Range Select is loaded/re-loaded
   c. Micro-step resolution is loaded/re-loaded
3. Motion controller direction is set/re-set
4. Acceleration/Max Speed are set/re-set
5. If Step Morphing is enabled, the setting is loaded/re-loaded
6. Active Motion parameters (i.e. speed/position targets) are re-set to 0
7. Encoder is set to 0 (even if connected motor is not equipped)

Initialization Options:

:1i1\r	Initializes the controller (Axis 1) using the sequencing above
:2i2\r	Kills Homing task and disables MOSFET driver for Axis 2
:3i3\r	Kills Homing task and re-enables MOSFET driver for Axis 3

Upon successful initialization, the controller will report a ready status for each Axis. Status is reported using the ‘g’ command, or ASCII 0x67 HEX.

All Axis (motors) can be initialized simultaneously. This is done using the “global” address.

Example:

:0i1\r	Initializes all connected Axis using the sequencing above

Motion/Indexer Operations

With the controller initialized, the motors can be commanded to move. Indexer operations are done using the Position command, ‘p’ or ASCII 0x70 HEX for absolute positioning. Relative positioning can be done using the Jog command, ‘j’ or ASCII 0x6A HEX.

Example:

:1p12800\r	Axis 1 Motor will move to absolute position 12800
:2p-6400\r	Axis 2 Motor will move to absolute position -6400
:1j12800\r	Axis 1 Motor will move 12800 steps in the positive direction
:2j-3200\r	Axis 2 Motor will move 3200 steps in the negative direction

To read back controller position, the Position command is used.
Example:

:1p\r	Axis 1 Motor Position is reported as :1p12800\r

NOTE: POSITION IS ONLY REPORTED AS ABSOLUTE POSITION. THERE IS NO MECHANISM TO REPORT RELATIVE POSITIONING

Motion Rotate (Speed) Operations

With the controller status ready (or initialized), the motor can be rotated in either direction. This is accomplished using the Set Speed command, ’s’ or ASCII 0x73 HEX.

:1s51200\r	Axis 1 will ramp to 51200 step/s.
:1s\r	Axis 1 will respond with current motor speed, ie. :1s51200\r

All speed is reported in steps per second. To get revolutions per second, RPS, use the following formula:

Controller Micro-Step Setting * 200 / Current Speed

With a micro-step setting of of 256x, in the above example the speed would be 1 RPS. To get RPM, you can multiply by 60. As a result, the above example would have a speed of 60 RPM.

Motion Homing Operations

With the Axis in the ready state, the motor is ready to operation or to be homed to a sensor, encoder index, or following error load.

NOTE: BEFORE HOMING IS INITIATED, ALL SENSORS MUST BE CONNECTED AND CONFIGURED FOR RISING OR FALLING EDGE TRIGGERING. WHEN HOMING TO A FOLLOWING ERROR, A HARD STOP MUST BE PRESENT WITH SUFFICIENT RIGIDITY.

Homing is done using the Home To command, ‘c’ or ASCII 0x63 HEX, followed by the home to sensor. The user can select from 6 different sensors with the following properties:

Table 17: Homing Behavior

Home To	Sensor	Direction *	Notes
0	Encoder Index	CCW	Must be configured as Hard Stop
1	Input 1	CCW	Must be configured as Hard Stop
2	Input 2	CW	Must be configured as Hard Stop
3	Input 3	CCW	Must be configured as Hard Stop
4	Input 4	CW	Must be configured as Hard Stop
5	Following Error	CCW	Init Load must be configured =>2

Examples:

:1c0\r Axis 1 will rotate CCW until the Index pulse is detected**
:2c2\r Axis 2 will rotate CW until sensor input 2 is triggered**

Using the “global” addressing:

:0c1\r All connected Axis rotate CCW until sensor input 1 is triggered**

***NOTE: DIRECTION IS FIXED. HOWEVER, THIS BEHAVIOR CAN BE REVERSED USING THE COMPATIBILITY COMMAND, “F1” OR “F2”.**

****NOTE: WHEN THE COMPATIBILITY SETTING IS “F0”, THE CONTROLLER WILL FIND THE TRIGGER EDGE. THIS CAN BE USEFUL FOR HIGH SPEED HOMING OPERATIONS.**

When Homing is complete, the controller will indicate the “Ready” state. This can be polled via the ‘g’, or ASCII 0x67 HEX, command.

Motion Encoder Operation

For motors equipped with encoders, the controller can be configured to:

1. Home to Encoder Index (See Table 17).
2. Home to Hard-Stop Detection (See Table 17 “Following Error”).
3. Monitor Following Error to determine load conditions.
4. Halt on Following Error.
5. Monitor Encoder Position.

Monitoring Following Error can give an indication of load conditions. It should be noted, however, that following errors do not mean there is a loss of position. Thus, some amount of following error is always present and even desirable from a torque efficiency perspective. As long as synchronization is maintained, stepper motors will not loose position and errors will be dynamically corrected. For a stepper motor to loose synchronization, the following error must be equivalent to 2 Full Steps.

Following Error is calculated as:

(Motor Position / Encoder Ratio) - Encoder Position

Encoder Ratio is determined automatically based on Encoder CPR and Micro-stepping resolution. It is assumed that the motor has 200 full steps per round.

To poll the following error, the ‘b’ or ASCII 0x62 HEX command is used.

Example:

:1b\r Following error is reported for Axis 1, as “:1b3\r” for a following error of 3

To monitor the encoder position independently from the following error, use the ‘y’ or ASCII 0x79 HEX command. This will report the actual encoder position in encoder counts.

Example:

:2y\r Encoder position is reported for Axis 2, as “:2y800\r” for 800 counts

Controller I/O Operation

All 9904 controllers have 4 inputs and 2 outputs that can be configured to use as inputs/ outputs or to perform certain actions.
Inputs can be configured to trigger in rising or falling edges, thus making easy to implement sinking or sourcing configurations. Also, outputs can be electrically configured as sinking or sourcing based on connection.

For manual operation, inputs or outputs should be set to ‘0’ using the NVM configuration. In this way, inputs can simply be read as high or low using the ‘I’ or ASCII 0x6C HEX command. In the same way, outputs can be set using the ’n’ or ASCII 0x6E HEX command and the ‘o’ or ASCII 0x6F HEX command for output 1 and 2 respectively.

To poll the inputs:

:1I\r Controller returns the status of all inputs. 1 is active and 0 is inactive

Controller Response:

:1I00001\r Controller responds inputs all inactive and Encoder Index Active

Table 18: Controller Input Reporting Structure

Command Echo	Input 1	Input 2	Input 3	Input 4	Encoder Index	Delimeter
:1I	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	\r

Outputs can be polled and well as set.

:1n\r Returns status of Axis 1 output 1, as “:1n0\r” for output 1 not active

:1o1\r Sets Axis 1 output 2 as active

NOTE: OUTPUTS CONFIGURED TO TAKE ACTION, IE. STOP MOTOR ON ERROR, CANNOT BE USED IN MANUAL OPERATION MODE AS THEY WILL ALWAYS INDICATE STATUS OF CONTROLLER AS CONFIGURED.

Controller Compatibility Modes

The 9904 Series 2 controllers are mostly compatible with previous generation models, with a few differences that can be configured.

I/O Compatibility Modes

One of the most important differences are the way I/O's are configured and read. Previous generations had a software bug that read the inverted status of the inputs, thus rising and falling edges were actually inverted. The series 2 controllers fixed this issue, but can be configured for compatibility in certain applications.

To Configure I/O compatibility mode:

:1G1\r	Configures the controller to invert the Inputs for Axis 1
:2G0\r	Configures Axis 2 to improved Input reporting and handling

Homing Compatibility Modes

Homing sequencing is also changed. Series 2 controllers corrected an issue with the direction being inverted, as well as improved the homing detection algorithm to find the edge of the inputs for more a precise home position. This also allows for high speed homing operations, which older generation controllers did not support.

To configure the Homing Compatibility Mode, the 'F' or ASCII 0x46 HEX command is used.

Examples:

:1F0\r	Homing Compatibility Mode is disabled for Axis 1
:1F1\r	Homing Compatibility Mode is enabled for Axis 1
:2F2\r	Homing Compatibility Mode direction reversed but using the improved detection

Updating Non-Volatile Memory (NVM) Settings

The Series 2 9904 controllers use a new method for storing parameters. Thus, an additional step is required when changing these parameters. For example, when setting the I/O Compatibility Mode using:

:1G1\r

This will be loaded into the RAM table but not the NVM table until the NVM Update command is sent. This allows for all updates to be done followed by a single write to flash memory. To load the RAM table contents to NVM, use the 'Q' or ASCII 0x51 HEX command.

Example:

:1Q\r	Save RAM table to NVM for Axis 1
:1Q>\r	Controller response to NVM Update indicating success
:1Q!\r	Controller response to NVM Update indicating failure

Using Set Points

The 9904 Series 2 controllers have 10 user configurable set points available to automate and synchronize positioning tasks. Each set point has a definable Position, Acceleration, and Speed.

Defining/Configuring Set Points

Set points are defined using a single bulk Set Point Command, number 0 to 9. Settings are comma delimited.

Example:

:1012800,4000,10000\r

Defines Axis 1 Set Point 0 (ASCII 0x30 HEX) as:
Position: 12800 absolute position in micro-steps
Speed: 4000 steps per second
Accel: 10000 steps per second per second

Using Set Points

With set points defined, the set point can be executed as a single Axis command or a global command where all Axis execute the set point simultaneously. Set points can be saved in the NVM table and loaded at boot-up. However, set points can also be used in volatile mode by simply configuring the set point as above and not issuing the NVM Update command.

To execute configured set point(s) use the ‘d’ or ASCII 0x64 HEX command.

:1d5\r Axis 1 will move the motor to the position defined for set point 5
:0d1\r All connected Axis will move to the position defined for set point 1

Contact and Warranty Information

Contact Information

Net Controls, LLC
1151 W Robinhood Drive
Suite C8
Stockton, CA 95207
Phone: 530-852-3221
www.n-controls.com
sales@n-contols.com
support@n-controls.com

Warranty Information

Each product sold by Net Controls, LLC (Net Controls) is warranted to be free from manufacturing defects that adversely affect the normal functioning thereof during the twelve (12) month period immediately following delivery thereof by NET CONTROLS (or in the case of products or components of any product purchased by NET CONTROLS from another for any lesser period of time that such manufacturer warrants said product or component to NET CONTROLS), provided that the same is properly operated under conditions of normal use and that regular, periodic maintenance and service is performed or replacements made, in accordance with the instructions provided by NET CONTROLS. The foregoing warranty shall not apply to any product or component that has been repaired or altered by anyone other than an authorized NET CONTROLS representative or that has been subject to improper installation or abuse, misuse, negligence or accident. NET CONTROLS shall not be liable for any damage, loss, or expense, whether consequential, special, incidental, direct or otherwise, caused by, arising out of or connected with the manufacture, delivery (including any delay in or failure to deliver), packaging, storage or use of any product sold or delivered by NET CONTROLS shall fail to conform to the foregoing warranty or to the description thereof contained herein, the purchaser thereof, as its exclusive remedy, shall upon prompt notice to NET CONTROLS of any such defect or failure and upon the return of the product, part or component in question to NET CONTROLS at it's factory, with transportation charges prepaid, and upon NET CONTROLS's inspection confirming the existence of any defect inconsistent with said warranty of any such failure, be entitled to have such defect or failure cured at the NET CONTROLS factory and at no charge therefore, by replacement or repair of said product, as NET CONTROLS may elect. NET CONTROLS MAKES NO WARRANTY OR REPRESENTATION OF ANY KIND, EXPRESS OR IMPLIED, (INCLUDING NO WARRANTY OR MERCHANTABILITY), EXCEPT FOR THE FOREGOING WARRANTY AND THE WARRANTY THAT EACH PRODUCT SHALL CONFORM TO THE DESCRIPTION THEREOF CONTAINED HEREIN, and no warranty shall be implied by law.