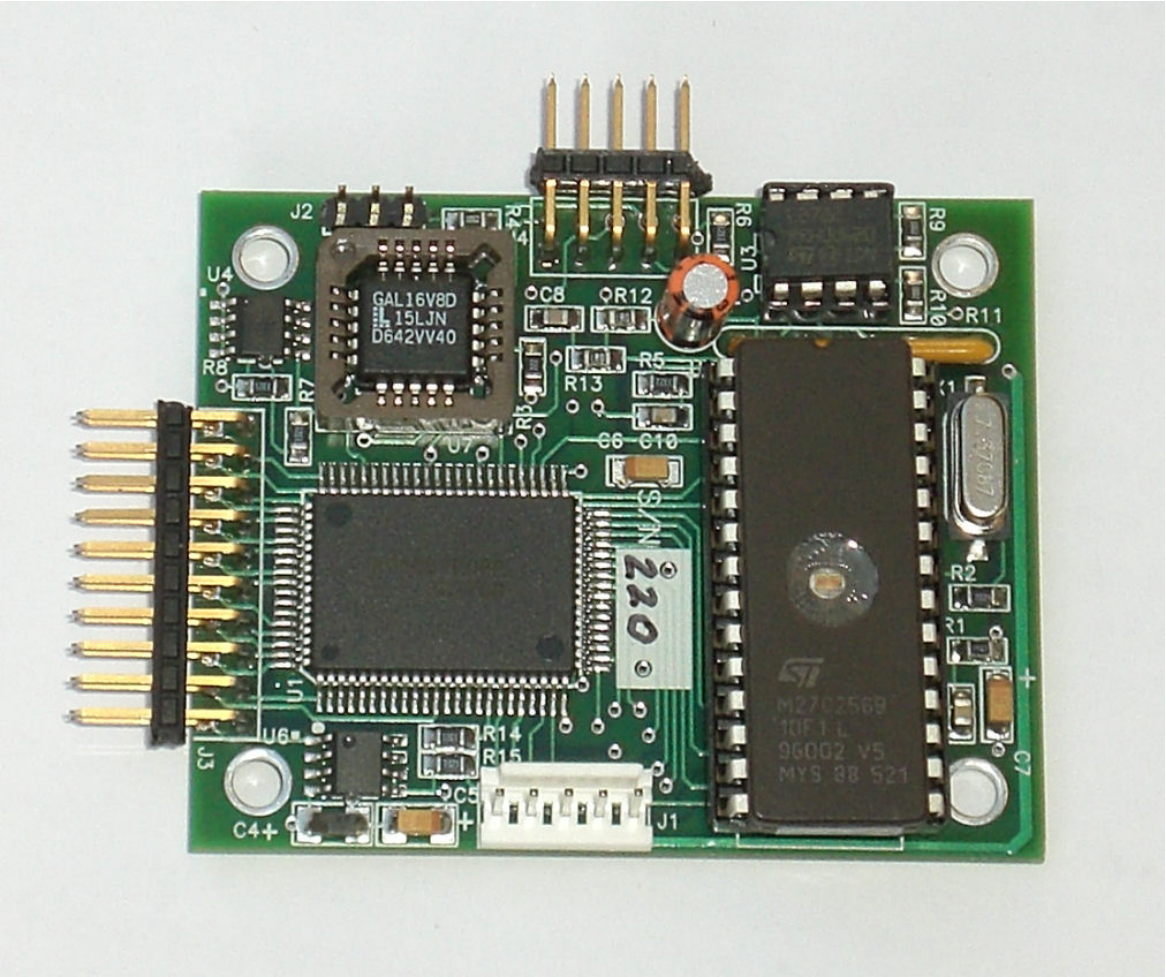
DIVA AUTOMATION

Innovation in Motion

SuprMotr Servo Controllers

User Manual for SuprMotrV9 Controllers



© DIVA Automation
803 Hermitage Dr., #211
Tel +01-800-984-DIVA • Fax +01-877-471-3437
Mail sales@divaauto.com

Contents

Key features of SuprMotrV9

- Servo control firmware for simultaneous control of position, velocity and acceleration, with programmable PID filtering.
- L2722 Motor Driver (1 A continuous current)
- On-board control language with over 50 commands.
- Up to 16 SuprMotr controllers may share a single RS-232 channel, with individual addressing..
- Operation from a single 9-12 volt power supply.
- Additional connector for general purpose input/output, fully supported by the integrated command structure.
- Eight analog inputs, 0-5 V, 8 bit resolution
- One analog output, 0-5 V, 8 bit resolution, 2 mA.
- Eighteen general purpose I/O signals. (Includes the nine analog I/O)

Connectors

J1: Power and RS-232

Molex 5-pin		
1	9-12 volts	
2	GND	DB-9 pin 5
3	Analog input 1	
4	Rx	DB-9 pin 3
5	Tx	DB-9 pin 2

Ensure that the power supply voltage does not exceed the maximum input voltage of the controller (Vmax = 12 VDC)!

Wrong polarity of the power supply will damage the controller!

J2: Special function

Molex 3-pin	
1	Signal
2	+5V
3	GND

The signal is connected to analog channel 2 and a 10K ohm pullup resistor and is available for general purpose use.

J3: Analog and Digital I/O

IDC-20	Function		IDC-20
1	External interrupt	I/O Channel 9	2
3	I/O Channel 3	Brake out	4
5	I/O Channel 8	I/O Channel 6	6
7	I/O Channel 5	I/O Channel 7	8
9	I/O Channel 2	I/O Channel 4	10
11	I/O Channel 16 or Analog Channel 8	I/O Channel 1	12
13	I/O Channel 14 or Analog Channel 6	I/O Channel 15 or Analog Channel 7	14
15	I/O Channel 12 or Analog Channel 4	I/O Channel 13 or Analog Channel 5	16
17	I/O Channel 10 or Analog output	I/O Channel 11 or Analog Channel 3	18
19	Ground	5 volts	20

J4: Motor, Encoder & Limit

IDC-10	Function		IDC-10
1	M+	5V	2
3	Encoder A	Encoder B	4
5	Ground	M-	6
7	Negative limit	Index/reference	8
9	Positive limit	Ground	10

The limit switch inputs are direction dependent. The negative limit stops motion in the negative direction, only. The positive limit stops only positive motion!

Do not connect or disconnect motor or power leads with power applied!

Wrong polarity of the motor connector may damage the motor encoder!

Communications

RS-232 Settings

Baud rate	**9600 (factory setting)** / 19200 / 38400
Stop bits	1
Parity	no
Protocol	no

How to communicate

The SuprMotr has an internal command interpreter. All data to or from the controller are ASCII strings. A command string starts with a 2 letter mnemonic and dependent on the command, one or more parameters! Over 80 commands are available for programming the SuprMotrV9.

Commands may be executed in various ways:

Single command -- One function executed immediately
Compound command -- Multiple functions executed immediately
Macro command -- Compound command stored for later execution
Sub-command-- Single-character commands
Address command—Network address selection

Color coding

For clarity, the following discussion will use **red** text to indicate data sent to the controller and **blue** text to indicate data returned from the controller. This will allow us to present the data in its exact form without the possibility of confusion caused by the use of quotation marks.

Character description

The following characters are allowed: (Lower case characters are translated internally to upper case, so they may be used interchangeably.)

Character(s)	Function	ASCII value (Decimal)	ASCII value (Hex)	PC key
Ctrl-a or Ctrl-A	Address selection identifier	01	01	Ctrl and A keys held simultaneously
Ctrl-c or Ctrl-C	Reply session terminator (ETX)	03	03	Not allowed for entry
Ctrl-h or Ctrl-H	Escape from macro or repeat loops	08	08	← (backspace)
Line feed (new line) (Ctrl-J)	Reply line terminator	0A	10	Not allowed for entry
Carriage return (Ctrl-M)	Command terminator	0D	13	Enter
Space	Ignored within commands	20	32	Space bar
Exclamation	AB subcommand	21	33	!
Pound sign	TC subcommand	23	35	#
Percent sign	TS subcommand	25	37	%
Single quote mark	TP subcommand	27	39	‘
Comma	Command line continuation	2C	44	Comma
0-9, (A-F or a-f hex)	Parameter entry	30-39	48-57	0-9, A-F
A-Z or a-z	Command entry	41-5A or 61-7A	65-90 or 97-122	A-Z or a-z
Question mark	TF subcommand	3F	63	?
Back slash	TI subcommand	5C	92	\

Command Termination

For communications to occur, an existing board address must first be selected. (*see Address Selection Chapter)* .

All commands except subcommands and address selection commands must be terminated by the *Carriage Return Char = dec 13 (*<CR>) (The Enter key on PC keyboards.)

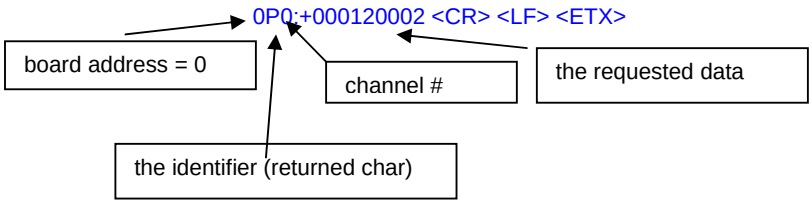
A <CR> entered before any other character, repeats the last command.

Reply Termination

All data sent from the SuprMotr are ASCII Strings. A report string starts with a command identifier which consists of:

the board address,
a one character identifier (return char),
the channel number,
a colon,
the requested data,
the reply termination CR LF ETX (decimal 13, 10, 3 or hex 0D, 0A, 03)

Example: TP<CR>



Single Command

A ***single command*** is executed immediately after the carriage return is received and will be repeated each time the carriage return is received, until a different command is entered. With this feature, it is very easy to continuously monitor the state of an input by simply holding down the "Enter" key and allowing the auto-repeat to send continuous characters.

Both uppercase and lower case characters are valid, and spaces are allowed.

Compound Commands

A compound command is a series of single commands separated by commas rather than by a carriage return. In this way, it is possible to string together several commands before terminating with the carriage return. These multiple commands will be executed sequentially.

The syntax for a compound command is:
CMD[n], CMD[n], ...,..., (Enter)

Once this command is entered, it remains in the buffer until replaced by another command and can be re-executed by transferring a carriage return. Compound commands may contain up to 15 single commands.

Example:

ma0,ws,mr500(Enter)
ma0,ws,mr2000,ws,rp10(Enter)

Macro Commands

Macros can be a most powerful tool for the programmer. A macro command is a grouping of commands to form a short program, implemented by a macro number.

To use macros for programming the **SuprMotr** controller, insert an **MD** (Macro Definition) command as the *first* instruction in the command string. The syntax for macro commands is:

MD(macro#), followed by a compound command string.

Example: MD3,MR1000,WS500,MR-1000,WS,RP5 (Enter)

In this example, MD3 defines macro #3. To call up this macro, just issue the command MC3**.**

Macro commands may be stored in any order, but you may prefer to number them sequentially as they are entered, because the system gives no warning if you define (and overwrite) an existing macro. You may wish to do this under many conditions, such as when one macro is called by another. It is sometimes desirable to define a complex motion in one macro and define key parameters such as torque, gain, or velocity, in another macro which is called by the main macro.

Macro commands can call another macro, without limit. For instance, MC1 could call MC2, and MC2 could then call MC3 and still be able to return to complete the remainder of
MC1.

Example: MD1,MC2,MC3,MC4,MC5,MC6 (Enter)

Macro commands may contain up to 15 single commands.

Sub-Commands

Sub-commands may be used at any time. They are most useful for interrogating variables without disturbing an operating program. An example would be a situation where a repetitive motion is in process, such as dispensing adhesive in a pattern. The operator would like to know the status of the command without stopping it. The sub-commands can be used to read the number of iterations in a loop, current system status, position, etc. A single character is also provided for emergency stop action. If a macro is running standard commands are not possible without interrupting the macros!

Sub-commands are one character commands without any command termination.

character	hex value	decimal value	standard command	function
Backspace	h08	08		Stops a running macro
‘	h27	39	tp	returns the actual position
?	h3f	63	tf	returns the actual position error
\	h5c	92	ti	returns the value of the iteration counter
%	h25	35	ts	returns the status of the system
!	h21	33	ab	stops motion (all controllers-whether addressed or not)
#	h23	35	tc	returns the status of digital inputs/outputs

Address Selection

The SuprMotr is unique among available controllers (and most RS-232 devices) in that it allows the connection of up to 32 controllers to a single RS-232 channel. This capability not only can save an enormous amount of cabling in larger systems, but also saves in only requiring a single communications port for each 32 axes of control. (Note that most laptop computers do not have sufficient drive for 32 receivers, but most can drive up to 16 receivers.)

Unlike many bussed systems, such as the IEEE-488, switching between devices is extremely simple. Only a two-character sequence must be transmitted to select a single controller on the bus and deselect all others. When the SuprMotr controller receives an ASCII 01 (Ctrl-a), it compares its own address stored in EEPROM with the next character following the Ctrl-a. If there is a match with the address, it enables itself to respond to subsequent commands. If not, it disables the response to any subsequent transmissions that are not address selection sequences.

Note: After power-up, the controller is not addressed. Send the address command first.

Example:	CtrlA5	Select Controller #5
	tp (Enter)	query position
	5p0:+000300	reply of controller #5
	CtrlA2	Select Controller #2
	tp (Enter)	query position
	2p0:+0002300	reply controller #2

There is an interesting feature which enables sending commands to all controllers simultaneously. *Ctrl-a+v+ (any standard command line).* Do not use this command with reporting commands. The results would be confusing and useless since all controllers would be transmitting on the same line at the same time.

To clarify, *Ctrl-a* means that the **Ctrl** key is held down while the **A** key is pressed, then the **Ctrl** key is released and the **V** key is pressed. This will select all controllers. Additional commands may be included on the same line, but all controllers remain addressed until a new address is sent, selecting a single controller. It is important to do this before attempting to communicate with a single controller.

Example:	Ctrl-AVDV2000 (Enter)	defines the velocity of all boards
	Ctrl-AVMR2000 (Enter)	all motors will move 2000 encoder increments relative to the present position.
	Ctrl-AVDP55,DI10,DL100,DP200 (Enter)	Sets gains for all controllers
	Ctrl-A1TG (Enter)	reports the PID gains for controller #1, only

Any motion sequence or operation begun prior to being de-selected will continue to be executed except for those that report data. In this manner, each SuprMotr on the bus can be addressed, programmed to execute any desired operation, and then de-selected before the sequence is repeated for any other motors selected.

Remember that each SuprMotr is de-selected when power is first applied, or after an **RT** command. An address selection sequence must be transmitted to begin control.

The EEPROM address for each SuprMotr is set with the **DA*n*** (define address) command. All addressed SuprMotrs on the bus when this command is issued will be set to address ***n***.

When power is first applied, all boards are in the "addressed" state. If more than one board is connected, they will conflict if any command is given to return data. This includes the copyright notice. To select one board and deselect all others, send an ASCII ctrl-A (01) followed by the address number. Do not send an enter or carriage return code. After selecting the address, a TB command will display the address of the selected board. This makes it easy to poll through all addresses to detect which addresses have been connected.

The three highest numbered I/O channels (14, 15 and 16) were used as address selection inputs. This does not interfere with their normal use as input or output channels, except during the power up hase. When power is cycled, or when the SuprMotrV9 is reset for any reason (including use of the RT command), the state of these three pins is sampled and used to define the address of the board.

To test the controller addressing, set an address and give any reporting command, such as TT. The address of the board is reported before the data.

If you select address 5, then the TT command will report "5T:+0000000000", for instance. If you select address 1, TT will report "1T:+0000000000". In each case, the board address is reported with the data.

To select a particular board, send CTRL-A followed by the board address. This will select that address and deselect all others. To go back to the example of a board set to address 5, hold down the CTRL key and press A, then press 4. This will select board address 4, which is not the same as the board you are testing. If you then give a TT command, nothing will happen because there is no board connected with address 4. Now, hold down the CTRL key and press A, then press 5. TT will then report "5T:+0000000000".

Please note that you should not press the Enter key after the CTRL-A 5. The Enter key is used to enter commands, such as the TT. CTRL-A followed by a number is not a command, it is only an address selection.

If you have five boards connected and have assigned addresses to them of 1,2,3,4 and 5, then you could communicate with each one individually by sending:

CTRL-A1TT(Enter)
CTRL-A2TT(Enter)
CTRL-A3TT(Enter)
CTRL-A4TT(Enter)
CTRL-A5TT(Enter)

This would return:

1T:+0000000000
2T:+0000000000
3T:+0000000000
4T:+0000000000
5T:+0000000000

If you are sending the command from a program, the ASCII code for CTRL-A is 01, so the sequence of hex digits to send CTRL-A5TT Enter would be: 01 35 54 54 13. If you are controlling from a terminal emulator, such as HyperTerm, simply type the keys.

After the address selection has been sent, the boards that do not match the address will continue to monitor the commands that are sent, but will respond only to an address selection sequence. Only the board that was selected will respond. You may send any commands to it as if it were the only board connected. There is no need to send another address selection sequence until it is desired to select a different board.

Data formats:

I	integer (24 bit)
ui	unsigned integer (24 bit)
w	word (16bit)
b	byte (8 bit)

tp Tell Position

Description	Tell Position reports the absolute position in increments of the encoder. **TP** may be used to monitor motion during both motor on and motor off states.
Parameter	no
Return Char	P
Index	always 0
Return Value	I [increments]
Example	tp(Enter) 0P0:+0000020002

tt Tell Target

Description	Reports target position in encoder increments. This is the absolute position to which the servo loop will try to drive the motor any time the **MN** (Motor ON) state is in effect. The target position may be specified directly with the **MA** (Move Absolute) and several other commands, or indirectly with the **MR** (Move elative) command. If the system is in decimal mode, ten digits will be reported with a leading minus sign (-), if the position is less than the position defined as "home."
Parameter	no
Return Char	T
Index	always 0
Return Value	I [increments]
Example	tt(Enter) 0T0:+0000020002

ts

Tell Status

Description

When the **TS** command is given, the status of the system as well as the motion and limit switches is reported. The format is 0S:OO MM LL SS VV EE , where OO is the operating system status consisting of the flags listed below, MM are the motion control status flags as listed below, LL are the limit switch status flags as listed below, SS are the live status of the limits and reference switch, VV is the move mode and EE is the error number.

The data is presented in either decimal or hex form, depending on the mode selected. For those not familiar with hexadecimal, it will require some practice to make use of this command. A tutorial on the use of this type of data will be included in a planned appendix to this manual.

If an error has previously occurred (LED display flashing) the error will be reset by this command.

Parameter
no

Return Char
S

Index
always 0

Byte_1: **Operational flags**

Bit 0	Echo on
Bit 1	Wait in process
Bit 2	Command error
Bit 3	Leading zero suppression active
Bit 4	Macro command called
Bit 5	Leading zero suppression disabled
Bit 6	Number mode
Bit 7	Board addressed

Byte_2: **Motion flags**

Bit 0	Motor off
Bit 1	Polarity of dynamic error
Bit 2	Polarity of move direction 1= positive
Bit 3	Move Error
Bit 4	Polarity of move at start of deceleration
Bit 5	Encoder phase
Bit 6	Excessive position error
Bit 7	Trajectory complete

Byte_3: **Limit Switch Status**

Bit 0	Limit Switch enable flag
Bit 1	Limit Switch active state 1 = high
Bit 2	Find Edge in process
Bit 3	Find edge initial state
Bit 4	
Bit 5	
Bit 6	
Bit 7	

Byte_4: **Limit Switch inputs**

Bit 0	
Bit 1	Positive limit switch live status
Bit 2	Reference switch live status
Bit 3	Negative limit switch live status
Bit 4	
Bit 5	
Bit 6	
Bit 7	

Byte_5: **Error Codes**

01	command not available
02	first command character must be a letter
03	no command
04	negative not allowed
05	character following command must be a number
06	value too large
07	value too small
08	continuation character must be a comma
09	command buffer overflow
0A	macro storage overflow

tf

Tell Following Error

Description	Reports the difference between the dynamic target and the actual position. During motion, it is normal for the actual position to lag behind the target position by some amount, usually dependent on the programmed velocity. If the velocity is higher than physically possible for the system, or if it has encountered an impediment, the following error will increase. If the obstruction is temporary, the servo action will attempt to restore the error to zero when it is removed.
Parameter	no
Return Char	F
Index	always 0
Return Value	I [increments]
Example	tf(Enter) 0F0:+0000000022

tb

Tell Board Address

Description	Reports the board address
Parameter	no
Return Char	B
Index	always 0
Return Value	number 0.. F
Example	tb(Enter) 0B0:01

db

Define Board Address

Description	Defines the board address
Parameter	the address
Return Char	no
Index	no
Return Value	no
Example	db5(Enter)

ud

Update

Description	Stores all parameters like velocity and filter parameters in the non-volatile memory. When the SuprMotr powers up these parameters will be active!
Parameter	no
Return Value	no
Example	ud(Enter)

dv

Define Velocity

Description	Defines the desired velocity for any movement. Causes the motor to run at velocity ***n*** in subsequent motion command***s.*** per second. If the torque load changes on the motor, the controller attempts to maintain the velocity by varying the motor current. The value ***n*** may be in the range from 1000 to 255,000. However, the usable range of velocity settings is determined by the number of lines in the encoder and the maximum RPM of the motor in use. Typical systems will fall into a range of 50,000 to 250,000 counts/second.
Parameter	l [increments/s]
Parameter Range	0..1 000 000
Return Value	no
Example	dv20000(Enter)

gv

Get Velocity

Description	Returns the programmed velocity defined with the ***dv*** command.
Return Char	Y
Index	always 0
Return Value	l [increments/s]
Example	gv(Enter) 0Y0:+0000020000

da

Define Acceleration

Description	Defines the desired acceleration for any movement. The maximum value is limited by the motor, power supply and inertia of the system.
Parameter	I [icrements/s^2]
Parameter Range	0..10 000 000
Return Value	no
Example	da200000(Enter)

ga

Get Acceleration

Description	Returns the programmed acceleration, (defined with command *da*)
Return Char	A
Index	always 0
Return Value	I [icrements/s^2]
Example	ga(Enter) 0L0:+0000200000

dp

Define P-Term

Description	Defines the P-Term of the PID filter. This command sets the slope of the proportional relationship between the position error and the motor voltage. The higher the gain value is set, the higher is the stiffness of position coupling, so that an small error value causes a proportionally larger motor current driving the motor towards the target. The default gain value is usually stable. The optimum value depends on friction, inertia, motor power, and the resolution of the encoder. It must be determined by the user. If the error reported by an axis after completing its motion is excessive, the gain value may be increased in small increments until the error is within acceptable limits. If the axis becomes unstable and begins to oscillate, the gain must be reduced until the oscillation stops. If the error remains too high, use the *di* command.
Parameter	w
Parameter Range	0..255
Return Value	no
Example	dp80(Enter)

gp

Get P-Term

Description	Returns the programmed p-term, (defined with command *dp*)
Return Char	G
Index	always 0
Return Value	w
Example	GP(Enter) 0G0:+0000000080

di

Define I-Term

Description	Defines the I-Term of the PID filter. Sets the gain to be applied to the integral term in the PID algorithm. The primary function of this term is to overcome friction-induced static errors. Note: The I-term has no effect unless the dl command is used to set the limit of output>0
Parameter	w
Parameter Range	0..255
Return Value	no
Example	di200(Enter)

gi

Get I-Term

Description	Returns the programmed integral term, (defined with command *di*)
Return Char	I
Index	0
Return Value	w
Example	gi(Enter) 0I0:+0000000200

dI

Define Integral limit

Description	Defines the integral limit of the PID filter.
Parameter	n
Parameter Range	0..255
Return Value	no
Example	dI200(Enter)

gI

Get Integral Limit

Description	Returns the actual programmed integral limit, (defined with command *dI*)
Return Char	L
Return Value	w
Index	always 0
Example	gI(Enter) 0L0:+0000000200

sm

Set Maximum Following Error

Description	Sets the **maximum allowable** error between the dynamic target and the actual position. May be changed as often as desired to provide maximum protection to the system. The normal following error can be monitored during motion with the TF command. For maximum system safety, use the *DE* command to limit following error to a value slightly above that required for normal operation.
Parameter	w
Parameter Range	0..32767
Return Value	no
Example	SM2000(Enter)

br

Define Baud Rate

Description	The SuprMotr is able to communicate with 3 standard Baud Rates 0 : 9600 1: 19200 2: 38400
Parameter	b
Parameter Range	0..2
Return Value	no
Example	br0(Enter) ; 9600 is active

rt

Reset Controller

Description	Restarts the internal firmware operation as if from a power off condition. All default values are restored. If Macro 0 exists, it will be executed.
Parameter	no
Return Value	no
Example	rt(Enter)

ef

Echo Off

Description	Disables echoing. When control is from a computer program, it is sometimes easier to program if there is no echo, unless the program uses it for verification of successful transmission.
Parameter	no
Return Value	no
Example	ef(Enter)

en

Echo On

Description	Enables echoing of command characters as they are entered. Each character received is echoed unchanged. This is a very useful feature when the SuprMotr is being controlled manually from a terminal..
Parameter	no
Return Value	no
Example	en(Enter)

mn

Motor On

Description	This is the normal system control mode, where the SuprMotr controls the axis position continuously. Any deviation between actual and target position causes the motor to be driven toward the target and possibly with full force, depending on the distance moved during the motor off condition. Use caution when turning the motor back on. The SuprMotr remembers its position when it received the *MF* command and it will try to return there unless the target position is redefined.
Parameter	no
Return Value	no
Example	mn(Enter)

mf

Motor Off

Description	When this command is issued, the motor is no longer held in position control and may be moved freely. The *MF* command is used to prevent unwanted movement or to allow for manual positioning of the unit. When manually positioned, however, the motor position is still monitored in the ***MF*** status and may be reported by the ***TP*** command. Use caution when turning the motor back on (***MN***). The target position is still the same as when the MF command was issued and it will try to return there unless the target position is redefined.
Parameter	no
Return Value	no
Example	mf(Enter)

ab

Abort Motion

Description	This command stops a motion! The target position is changed to be equal to the present position. n=0 stop abrupt n=1 decelerated stop
Parameter	n
Parameter Range	0 ..1
Return Value	no
Example	ab1(Enter) decelerated stop

mr

Move Relative

Description	This command generates a motion of relative distance of ***n*** counts in the specified direction from the actual motor position. **n** may be either a positive or negative number up to a total target position +/-1 073 741 843
Parameter	I
Parameter Range	Depends on the absolute target, should not exceed the maximum position +/-1 073 741 843.
Return Value	no
Example	mr5000(Enter)

ma

Move Absolute

Description	This command generates a motion to the absolute position ***n.*** The zero, or home position, may be defined by the **DH** (Define Home) statement. If not otherwise defined, it is the position where the controller was when powered on.
Parameter	I
Parameter Range	+/-1 073 741 843
Return Value	no
Example	ma2000(Enter)

rp

Repeat

Description	This command causes the command string to repeat ***I+1*** times. If ***n*** is not specified, the commands are repeated 65,535 times. The repeat loop may be interrupted by transferring a backspace character (Ctrl-H). (To repeat forever, use two **RP** commands in sequence.)
Parameter	n
Parameter Range	0..65535
Return Value	no
Example	ma0,ws,ma2000,ws,rp99(Enter) (performs 100 iterations)

fe

Find Edge

Description	This command is used to initialize the system at a given position. The motor runs at a programmed speed until a change of state occurs on the limit input line. The direction of motion is controlled by the parameter, n. Example: **F**E0 (Enter) : Causes motor to move in a positive direction until the E2 input is activated. **F**E1 (Enter) : Causes motor to move in a negative direction until the E1 input is activated. **F**E4 (Enter) : Causes motor to move in a positive direction until the E1 input is deactivated.
Parameter	n
Parameter Range	0,1,4
Return Value	no
Example	fe1(Enter) search for E1 in negative direction

dh

Define Home

Description	Defines the current motor position as zero position (home position).
Parameter	no
Return Value	no
Example	dh(Enter)

gh

Go Home

Description	The Go Home command causes the motor to move to absolute zero position. Equivalent to an **MA0** (Move Absolute axis a to zero) command.
Parameter	no
Return Value	no
Example	gh(Enter)

ws

Wait for Stop

Description	Wait until axis has reached the end of the trajectory, and waits another *n* milliseconds before the execution of the next command. Without the parameter, the default wait time is 1000 ms!
Parameter	n
Parameter Range	0..65535
Return Value	no
Example	mr2000,ws500,ma0(Enter)

md

Macro Definition

Description	This command is used to define a new macro command. Any duplication of numbers will simply result in the loss of any previously defined macro using that number. To define a macro, choose any number in the allowable range for the new macro and enter MD followed by this number and a comma before entering the function you wish the macro to perform, in the normal manner. Macro 0 is an auto start macro, executed after booting the controller or after an RT command.
Parameter	macro number + command string
Parameter Range	0..14 (macro number)
Return Value	no
Example	md1,ma0,ws,ma2000,ws(Enter)

mc

Macro Call

Description	This command may be used to implement a previously defined macro command. If there is no macro defined by the number n, no action will be taken. **A running macro can be stopped by sending the <backspace>** dec=08 hex =$08
Parameter	n, macro number
Parameter Range	0.14
Return Value	no
Example	mc1(Enter)

tm

Tell Macro

Description	Displays all previously stored macro commands. If ***n*** = 0 or, if ***n*** is not specified, all macros will be displayed. Since macros may be defined in any sequence, the **TM** command is useful for confirming the existence of, as well as displays all previously stored macro commands. If ***n*** = 0 or, if ***n*** is not specified, all macros will be displayed. Since macros may be defined in any sequence, the **TM** command is useful for confirming the existence of, as well as
Parameter	n, macro number 0..14
Return Value	the contents of the macro(s)
Example	tm1(Enter) mc1,mr555,ws,gh

rm

Reset Macro

Description	Resets macro memory
Parameter	0..macro number)
Example	rm(Enter) reset all macros rm1(Enter) reset only macro 1

so

Set Analog Output

Description	Sets the analog output to a value. The output is from 0 to 5 V with 8 bit resolution. One bit is 19.5 mV
Parameter	b -> value of the output 8 bit 0..255
Example	so255(Enter) the output is 5 V

io

Increment Analog Output

Description	Increments the analog output . The output is a 0 to 5 V 8 bit output. One bit is 19.5 mV.
Parameter	I -> incremental value -255..255
Example	io2(Enter) ; new analog value = old value + 2 mc2,io1,wa1,rp(Enter) ; generates an analog sawtooth

to

Tell Analog Output

Description	Returns the actual value of the analog output
Parameter	no
Return Char	o
Index	always 0
Return Value	b [0..255]
Example	to(Enter) 0o0:255

ta

Tell Analog Input

Description	Returns the current value of the analog inputs 1 and 2
Parameter	b channel [0 ..2] 0 = report both inputs 1..2 = reports channel #
Return Char	A
Index	b [0..2]
Return Value	n m -> value of the analog inputs 8 bit n = channel 1 m = channel 2 0..255
Example	ta(Enter) 0A0: 127 230 ta1(Enter) 0A1 :127 ta2(Enter) 0A2 :230

tc

Tell Channel

Description	Returns the current status of the digital inputs and outputs.
Parameter	b [0..16] 0 = report all outputs and inputs 1..16 = report input channel #
Return Char	H
Index	0 ..16
Return Value	n m -> value of the inputs /outputs 8 bit n = outputs m = inputs
Example	tc(Enter) 0H0: 00 00 ; all inputs not active tc1(Enter) 0H1: 1 ; reports channel 1

cf

Channel oFf

Description	Reset digital output(s)
Parameter	b, 0..16 0 = all outputs 1..16 = only specified channel #
Return Value	no
Example	cf 0(Enter) or cf (Enter) resets all 16 digital outputs cf1(Enter) resets channel 1

cn

Channel oN

Description	Set digital output (s)
Parameter	b, 0..16 0 = all outputs 1..16 =only specified channel #
Return Value	no
Example	cn0(Enter) or cn(Enter) sets all 16 digital outputs cn1(Enter) sets channel 1

wn

Wait oN

Description	Stops Command Execution until Input Channel on
Parameter	b [1..16] 1..16 = input channel #
Return Char Index	1 ..16
Return Value	no
Example	wn1,ma1000(Enter)

wf

Wait oFf

Description	Stops Command Execution until Input Channel off
Parameter	b [1..16] 1..16 = input channel #
Return Char Index Return Value	 1 ..16 no
Example	wn1,ma1000,wf1,ma0(Enter)

xn

eXecute if oN

Description	Executes Commands if Input Channel on
Parameter	b [1..16] 1..16 = input channel #
Return Char Index Return Value	 1 ..16 no
Example	xn1,mr555,ws10(Enter) move if Input1 = on xn1,xn2,mr599,ws10(Enter) move if Input 1 & 2 = on

xf

eXecute if oFf

Description	Executes commands if input channel off
Parameter	b [1..16] 1..16 = input channel #
Return Char Index Return Value	 1 ..16 no
Example	xn1,mr555,ws10(Enter) move if Input1 = on xn1,xn2,mr599,ws10(Enter) move if Input 1 & 2 = on xn1,xn2,xf3,mr599,ws10(Enter) move if In 1 & 2 = on & In 3 = off

Some examples:

Homing

This procedure can be used for calibrating the position feedback after power has been applied. When power is removed, the position information is lost. Restoring power causes the position reading to be reset to zero. To restore the absolute position to a correct value, it is necessary to return to a reference point. These commands can be entered on a single line, separated by commas, as a compound command or macro. They may also be entered as single commands, but then there is no need to use the WS commands.

DV10000	Reduce velocity
FE1	Search in negative direction
WS	Wait until after motor stops at limit switch 1000 ms
DV500	Reduce velocity more
FE4	Search for release of the limit switch in forward direction
WS	Wait until after move complete 1000 ms
DH	Define home, actual position = 0

Demo

Similar comments apply to these commands. If entered on a single line, separated by commas, the sequence will be performed 50 times. If entered individually, the RP49 command has no function.

MA0	Move to absolute position 0
WS500	Wait until after move complete 500 ms
MR500	Move relative 500 increments
WS100	Wait until after move complete 100 ms
RP49	Do the command line 50 times

Some more macros

Macro 2: This macro calls other macros (Commands EM and MC can be used equally.)

MD2, EM10,EM11,EM12,MA,WS500,RP(Enter)

Macro 10: Motor moves 20000 increments relative, after trajectory complete, output 1 goes high for 200 ms, then output goes low

MD10,MR20000,WS1,CN1,WA200,CF1(Enter)

Macro 11: Motor moves 500 increments relative, after trajectory complete wait for 100 ms , repeat this macro 9 times for a total of 10 executions

MD11, MR500,WS100,RP9(Enter) (format as entered)

MD11, MR+0000000500,WS+0000000100,RP+000000009(Enter) (format as reported)

Macro 12: Same as macro 11 in negative direction

MD12, MR500,WS100,RP5(Enter)

Command Short Form

Reporting command	Reply ID character	Set command	Subcommand character
TS **T**ell **S**tatus	**S:**		**%**
TQ **T**ell tor**Q**ue	**Q:**	SQ **S**et tor**Q**ue	
HE **HE**lp: Print list of commands			
TE **T**ell **E**rror (distance from target)	**E:**		
TP **T**ell **P**osition	**P:**		‘ (single quote)
TT **T**ell **T**arget	**T:**		
TF **T**ell **F**ollowing error (distance between position and dynamic target)	**F:**		**?**
TD **T**ell **D**ynamic target	**N:**		
TV **T**ell actual **V**elocity	**V:**		
TY **T**ell velocit**Y**	**Y:**	SV **S**et **V**elocity	
TL **T**ell acce**L**eration	**A:**	SA **S**et **A**cceleration	
GP **G**et **P**roportional gain	**G:**	DP **D**efine **P**roportional gain	
GI **G**et **I**ntegral gain	**I:**	DI **D**efine **I**ntegral gain	
GD **G**et **D**erivative gain	**D:**	DD **D**efine **D**erivative gain	
GL **G**et Integral Limit	**L:**	DL **D**efine integral **L**imit	
		SM **S**et **M**aximum following error	
VE Display **VE**rsion number			
TI **T**ell the **I**teration number	**X:**		****
TB **T**ell **B**oard address	**B:**	DB **D**efine **B**oard address	
CS Perform self-test **C**heck**S**um	**C:**		
TJ **T**ell **J**og	**J:**	SJ **S**et **J**og	
		LN **L**imit switch operation o**n**	
		LF **L**imit switch operation o**ff**	
		LL **L**imits switches active **l**ow	
		LH **L**imit switches active **h**igh	
TG **T**ell **G**ains (GP, GI, GD and GL)	**P,I,D,L**		
		UD **U**p**D**ate	
		DM Input and output in **D**ecimal for**m**at	
		HM Input and output in **H**ex for**m**at	
		RT **R**ese**T** all parameters to default and do power-up start	
		EF Turn of**f** **e**cho to RS-232 port	
		EN Turn o**n** **e**cho to RS-232 port	
DU **Du**mp (TT, TP, TD and TS)	**T,P,D,S**		
		AB **Ab**ort Motion	**!** (exclamation)
		MN **M**otor O**n**	
		MF **M**otor o**ff**	
		WA **Wa**it specified time	
		MR **M**ove **R**elative	
		MA **M**ove **A**bsolute	
		DH **D**efine **H**ome	
		GH **G**o **H**ome	
		RP **Rep**eat from beginning of line	
		WS **W**ait until end of trajectory (**S**top)	
		FE **F**ind **E**dge	
		CI Set **C**hannel as **I**nput	
		CT Set **C**hannel as outpu**T**	
TM **T**ell **M**acro (1..31) TZ **T**ell Macro **Z**ero		MD **M**acro **D**efinition	
		MC **M**acro **C**ommand (or EM **E**xecute **M**acro)	
		RM **R**eset **M**acro	
		RZ **R**eset **M**acro 0	
TO **T**ell Analog **O**utput	**O:**	SO **S**et analog **O**utput	
TA **T**ell **A**nalog Input	**A:**		
TC **T**ell **C**hannel	**H:**	CF **C**hannel **O**ff	#
		CN **C**hannel o**N**	
		CP Set **C**hannel **P**attern	
		PH Set encoder **PH**ase	
		ST **ST**op	

Factory Settings:

DV **D**efine **Velocity**	DV20000
DA **D**efine **A**cceleration	DA200000
DP **D**efine **P**roportional gain	DP75
DI **D**efine **I**ntegral gain	DI0
DD **D**efine **D**erivative gain	DD0
DL **D**efine integral **L**imit	DL100
SM **D**efine maximum following **E**rror	SM10000
EN Turn o**n** **e**cho to RS-232 port	EN
LN	
LL	