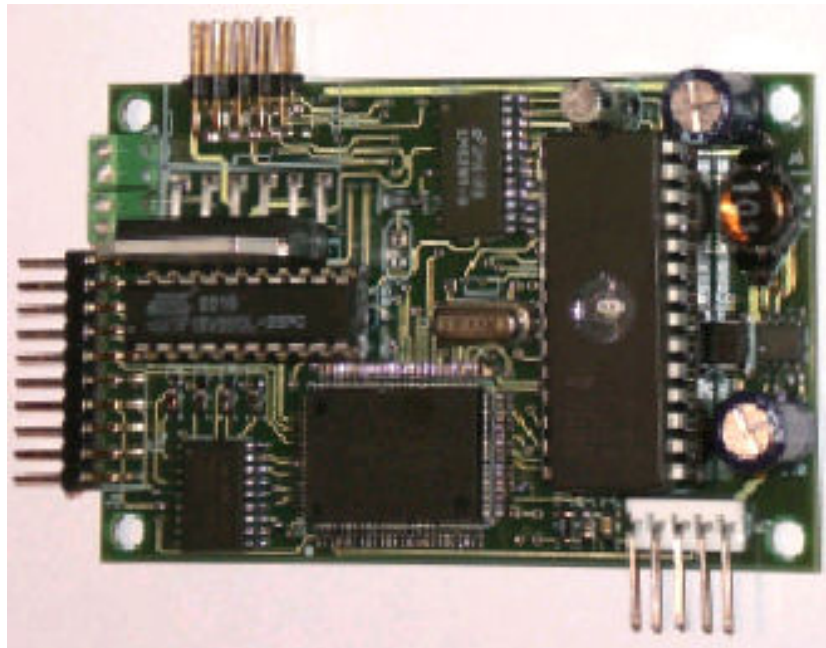


DIVA AUTOMATION  
*Innovation in Motion*

# SuprMotr Servo Controllers

# User Manual for SuprMotrX Controllers



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

# Contents

|  |           |
|--|-----------|
| <b>Key features of SuprMotr .....</b>      | <b>5</b>  |
| <b>Connectors.....</b>                     | <b>6</b>  |
| <i>J1: Motor, Encoder &amp; Limit.....</i> | <i>6</i>  |
| <i>J2: Analog and Digital I/O.....</i>     | <i>6</i>  |
| <i>J3: Power and RS-232.....</i>           | <i>7</i>  |
| <b>Communications .....</b>                | <b>8</b>  |
| <i>RS-232 Settings.....</i>                | <i>8</i>  |
| <i>Reply Termination.....</i>              | <i>8</i>  |
| <i>Single Command.....</i>                 | <i>9</i>  |
| <i>Compound Commands.....</i>              | <i>9</i>  |
| <i>Macro Commands.....</i>                 | <i>9</i>  |
| <i>Sub Commands.....</i>                   | <i>10</i> |
| <i>Control Characters:.....</i>            | <i>10</i> |
| <i>Address Selection.....</i>              | <i>10</i> |
| <b>Command Reference.....</b>              | <b>11</b> |
| <i>Tell Position.....</i>                  | <i>11</i> |
| <i>Tell Target.....</i>                    | <i>11</i> |
| <i>Tell Status.....</i>                    | <i>12</i> |
| <i>Move Status.....</i>                    | <i>13</i> |
| <i>Tell Following Error.....</i>           | <i>14</i> |
| <i>Tell Board Address.....</i>             | <i>14</i> |
| <i>Define Board Address.....</i>           | <i>14</i> |
| <i>Update.....</i>                         | <i>15</i> |
| <i>Define Velocity.....</i>                | <i>15</i> |
| <i>Get Velocity.....</i>                   | <i>15</i> |
| <i>Define Acceleration.....</i>            | <i>16</i> |
| <i>Get Acceleration.....</i>               | <i>16</i> |
| <i>Define P-Term.....</i>                  | <i>17</i> |
| <i>Get P-Term.....</i>                     | <i>17</i> |
| <i>Define I-Term.....</i>                  | <i>17</i> |
| <i>Get I-Term.....</i>                     | <i>18</i> |
| <i>Define Integral limit.....</i>          | <i>18</i> |
| <i>Get Integral Limit.....</i>             | <i>18</i> |
| <i>Define Following Error.....</i>         | <i>19</i> |
| <i>Define Baud Rate.....</i>               | <i>19</i> |
| <i>Reset Controller.....</i>               | <i>19</i> |
| <i>Echo Off.....</i>                       | <i>20</i> |
| <i>Echo On.....</i>                        | <i>20</i> |
| <i>Motor On.....</i>                       | <i>20</i> |
| <i>Motor Off.....</i>                      | <i>21</i> |
| <i>Abort Motion.....</i>                   | <i>21</i> |
| <i>Move Relative.....</i>                  | <i>22</i> |
| <i>Move Absolute.....</i>                  | <i>22</i> |
| <i>Repeat.....</i>                         | <i>22</i> |

|  |           |
|--|-----------|
| <i>Find Edge</i> .....                   | 23        |
| <i>Find Index</i> .....                  | 23        |
| <i>Define Home</i> .....                 | 23        |
| <i>Go Home</i> .....                     | 24        |
| <i>Wait Stop</i> .....                   | 24        |
| <i>Macro Definition</i> .....            | 24        |
| <i>Macro Call</i> .....                  | 25        |
| <i>Tell Macro</i> .....                  | 25        |
| <i>Reset Macro</i> .....                 | 25        |
| <i>Set Analog Output</i> .....           | 26        |
| <i>Increment Analog Output</i> .....     | 26        |
| <i>Tell Analog Output</i> .....          | 26        |
| <i>Tell Analog Input</i> .....           | 27        |
| <i>Tell Channel</i> .....                | 27        |
| <i>Channel oFf</i> .....                 | 28        |
| <i>Channel oN</i> .....                  | 28        |
| <i>Wait oN</i> .....                     | 28        |
| <i>Wait oFf</i> .....                    | 29        |
| <i>eXecute if oN</i> .....               | 29        |
| <i>eXecute if oFf</i> .....              | 29        |
| <b>Trigger Functions: .....</b>          | <b>30</b> |
| <i>Set Breakpoint port</i> .....         | 30        |
| <i>Breakpoint Define</i> .....           | 30        |
| <i>define Breakpoint Increment</i> ..... | 31        |
| <i>define Pulse Width</i> .....          | 31        |
| <b>Some examples: .....</b>              | <b>32</b> |
| <b>Command Short Form .....</b>          | <b>34</b> |
| <b>Factory Settings : .....</b>          | <b>35</b> |

# Key features of SuprMotrX

- LM629 Servo Controller for simultaneous control of position, velocity and acceleration, with programmable PID filtering.
- LMD18200 Motor Driver (3 A continuous current, 6 A for 200 mSec)
- Non-volatile (permanent) storage of servo parameter setups and user programs.
- On-board control language with over 50 commands.
- Position breakpoint Trigger Output
- Up to 16 SuprMotr controllers may share a single RS-232 channel, with individual addressing..
- Operation from a single 9-32 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: 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!

## J2: Analog and Digital I/O

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

### J3: Power and RS-232

| Molex 5-pin |              |
|-------------|--------------|
| 1           | 9-36 volts   |
| 2           | GND          |
| 3           | Analog input |
| 4           | Rx           |
| 5           | Tx           |



Ensure that the power supply output voltage does not exceed the maximum input voltage of the controller ( $V_{max}=36$  VDC)!

Wrong polarity of the power supply will damage the controller!

# Communications

## RS-232 Settings

|           |                      |
|-----------|----------------------|
| Baud rate | 9600 / 19200 / 38400 |
| Stop bits | 1                    |
| Parity    | no                   |
| Protocol  | no                   |
|           |                      |

## How to communicate

The SuprMotr *series* 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 50 commands are available for programming the SuprMotr.

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-commands** -- Single-character commands
- Address command** -- Network address selection

## Command Termination

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

A command must be terminated by the *Carriage Return Char = dec 13 (<CR>)*

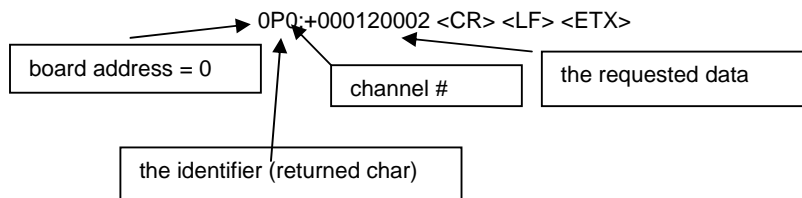
A <CR> without any other character, repeats the last command!

## Reply Termination

All data sent from the SuprMotr *series* are ASCII Strings. An answer 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 (dec 13, 10, 3 or hex D, E,3)

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 autorepeat 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], ..., (rtn)

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<CR>  
ma0,ws,mr2000,ws,rp10<CR>
```

## 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 (rtn)

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

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                                     |
|-----------|-----------|---------------|------------------|--|
| blank     | h20       | 32            |                  | Stops a running macro                        |
| '         | h27       | 39            | tp               | returns the actual position                  |
| ?         | h3f       | 63            | tf               | returns the actual position error            |
| \         | h5c       | 92            | ms               | returns Byte 1 of the commands TS            |
| %         | h25       | 35            | ts               | returns the status of the system 6 bytes     |
| !         | h21       | 33            | ab               | stops motion (all controllers)               |
| #         | h23       | 35            | tc               | returns the status of digital inputs/outputs |

## Control Characters:

| character | hex value | decimal value | function                   |
|-----------|-----------|---------------|----------------------------|
| CR        | \$E       | 13            | command /reply termination |
| LF        | \$D       | 10            | reply termination          |
| ETX       | \$3       | 3             | reply termination          |
| Ctrl A    | \$1       | 1             | address selection          |
|           |           |               |                            |
|           |           |               |                            |

## Address Selection

Up to 16 controllers can share one serial interface. The address selection is done by the command *Ctrl+A* + the controller address, defined by command *<db>*, all others are automatically de-addressed.

***Ctrl+A = ASCII table character 1***

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

|          |              |                        |
|----------|--------------|------------------------|
| Example: | CtrlA5       | address Controller #5  |
|          | tp           | query position         |
|          | 5p0:+000300  | reply of controller #5 |
|          | CtrlA2       | address Controller #2  |
|          | tp           | query position         |
|          | 2p0:+0002300 | reply controller #2    |

There is a interesting feature which enables sending commands to all available controllers.

The command *Ctrl-a+v+ standard command*

Do not use this command with reporting commands, the data will be chaotic!

|          |               |  |
|----------|---------------|--|
| Example: | CtrlAV DV2000 | defines the velocity of all boards     |
|          | CtrlAV MR2000 | all motors will move 2000 inc relative |

# Command Reference

Data formats:

|    |                           |
|----|---------------------------|
| l  | 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  
Return Char  
Index  
Return Value

no  
P  
always 0  
I [increments]

Example

tp  
OP0:+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 Relative) 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  
Return Char  
Index  
Return Value

no  
T  
always 0  
I [increments]

Example

tt  
OT0:+0000020002

ts

## Tell Status

Description

Parameter  
Return Char  
Index

When the **TS** command is given, the status of the system as well as the motion and limit switches is reported. The answer is a string with 7 Hex Bytes (B\_1 .. B\_7)

If an error occurs (LED Display) the error will be reset by this command!

no

S

always 0

b1 b2 b3 b4 b5 b6 b7

Byte\_1: LM629 Status Byte

|       |                          |
|-------|--------------------------|
| Bit 0 | Busy                     |
| Bit 1 | Command error            |
| Bit 2 | Trajectory complete      |
| Bit 3 | Index found              |
| Bit 4 | Position limit exceeded  |
| Bit 5 | Excessive position error |
| Bit 6 | Breakpoint reached       |
| Bit 7 | Motor off                |

Byte\_2: Operational Flags 1

|       |                                   |
|-------|-----------------------------------|
| Bit 0 |                                   |
| 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 |                                   |
| Bit 7 | Echo on                           |

Byte\_3: Operational Flags 2 ( only for factory use)

|       |  |
|-------|--|
| Bit 0 |  |
| Bit 1 |  |
| Bit 2 |  |
| Bit 3 |  |
| Bit 4 |  |
| Bit 5 |  |
| Bit 6 |  |
| Bit 7 |  |

Byte\_4: Motion Flags

|       |   |
|-------|---|
| Bit 0 |   |
| 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 | Velocity mode                             |
| Bit 6 | Excessive position error                  |
| Bit 7 | LM629 access in process                   |

Byte\_5: 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 | Brake on Flag                      |
| Bit 4 | Find limit switch flag             |
| Bit 5 |                                    |
| Bit 6 |                                    |
| Bit 7 |                                    |

Byte\_6: Limit Switch inputs

|       |                      |
|-------|----------------------|
| Bit 0 | Limit reverse active |
| Bit 1 |                      |
| Bit 2 | Limit forward active |
| Bit 3 |                      |
| Bit 4 |                      |
| Bit 5 |                      |
| Bit 6 |                      |
| Bit 7 |                      |

Byte\_7: 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                       |



## Move Status

Description

Reports only the LM629 status (Byte 1 of the **TS** command )

Parameter  
Return Char  
Return Value  
Index  
Example

no  
M  
b  
always 1  
ms

0M1:4C

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  
Return Char  
Index  
Return Value

no  
F  
always 0  
I [increments]

Example

tf  
OF0:+0000000022

tb

## Tell Board Address

Description

Reports the board address

Parameter  
Return Char  
Index  
Return Value

no  
B  
always 0  
number 0.. F

Example

tb  
0B0:01

db

## Define Board Address

Description

Defines the board address

Parameter  
Return Char  
Index  
Return Value

the address  
no  
no  
no

Example

db5

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   |

dv

## Define Velocity

|                 |  |
|-----------------|--|
| Description     | Defines the desired velocity for any movement. Causes the motor to run at velocity <i>n</i> in subsequent motion commands. per second. If the torque load changes on the motor, the controller attempts to maintain the velocity by varying the motor current. The value <i>n</i> 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       | I [increments/s]   |
| Parameter Range | 0..1 000 000   |
| Return Value    | no   |
| Example         | dv20000  |

gv

## Get Velocity

|              |   |
|--------------|---|
| Description  | Returns the programmed velocity defined with the <i>dv</i> command. |
| Return Char  | Y   |
| Index        | always 0  |
| Return Value | I [increments/s]  |
| Example      | gv<br>  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 [increments/s <sup>2</sup> ]  |
| Parameter Range | 0..10 000 000   |
| Return Value    | no  |
| Example         | da200000  |

ga

## Get Acceleration

|              |  |
|--------------|--|
| Description  | Returns the programmed acceleration, (defined with command <i>da</i> ) |
| Return Char  | A  |
| Index        | always 0   |
| Return Value | I [increments/s <sup>2</sup> ]   |
| Example      | ga<br>  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 a 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 <i>di</i> command. |
| Parameter       | w  |
| Parameter Range | 0..32767   |
| Return Value    | no   |
| Example         | dp800  |

gp

## Get P-Term

|              |  |
|--------------|--|
| Description  | Returns the programmed p-term, (defined with command <i>dp</i> ) |
| Return Char  | G  |
| Index        | always 0   |
| Return Value | w  |
| Example      | gp   0G0:+0000000800   |

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.<br>Note: The I-term has no effect unless the <i>di</i> command is used to set the limit of output>0 |
| Parameter       | w   |
| Parameter Range | 0..32767  |
| Return Value    | no  |
| Example         | di200   |

gi

## Get I-Term

Description

Returns the programmed integral term, (defined with command *dI*)

Return Char  
Index  
Return Value

I  
0  
w

Example

gi | 0I0:+0000000400

dI

## Define Integral limit

Description

Defines the integral limit of the PID filter.

Parameter  
Parameter Range  
Return Value

n  
0..32767  
no

Example

dI2000

gI

## Get Integral Limit

Description

Returns the actual programmed integral limit, (defined with command *dI*)

Return Char  
Return Value  
Index  
Example

L  
w  
always 0  
gI

| 0L0:+0000002000

de

## Define Following Error

|                 |  |
|-----------------|--|
| Description     | Sets the <b>maximum allowable</b> 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 <i>DE</i> 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         | de2000   |

br

## Define Baud Rate

|                 |  |
|-----------------|--|
| Description     | The SuprMotr is able to communicate with 3 standard Baud Rates<br>0 : 9600<br>1: 19200<br>2: 38400 |
| Parameter       | b  |
| Parameter Range | 0..2   |
| Return Value    | no   |
| Example         | br0 ; 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  |

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<br>Return Value | no<br>no  |
| Example                   | ef  |

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<br>Return Value | no<br>no   |
| Example                   | en   |

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 <i>MF</i> command and it will try to return there unless the target position is redefined. |
| Parameter<br>Return Value | no<br>no   |
| Example                   | mn   |

mf

## Motor Off

|              |   |
|--------------|---|
| Description  | <p>When this command is issued, the motor is no longer held in position control and may be moved freely. The <i>MF</i> 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 <i>MF</i> status and may be reported by the <i>TP</i> command.</p> <p>Use caution when turning the motor back on (<i>MN</i>). The target position is still the same as when the <i>MF</i> command was issued and it will try to return there unless the target position is redefined.</p> |
| Parameter    | no  |
| Return Value | no  |
| Example      | mf  |

ab

## Abort Motion

|                 |  |
|-----------------|--|
| Description     | <p>This command stops a motion ! The target position is changed to be equal to the present position.</p> <p>n=0 stop abrupt<br/>n=1 decelerated stop</p> |
| Parameter       | n  |
| Parameter Range | 0 ..1  |
| Return Value    | no   |
| Example         | ab1            decelerated stop  |

mr

## Move Relative

|                 |  |
|-----------------|--|
| Description     | This command generates a motion of relative distance of <i>n</i> counts in the specified direction from the actual motor position. <i>n</i> 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   |

ma

## Move Absolute

|                 |   |
|-----------------|---|
| Description     | This command generates a motion to the absolute position <i>n</i> . The zero, or home position, may be defined by the <b>DH</b> (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  |

rp

## Repeat

|                 |  |
|-----------------|--|
| Description     | This command causes the command string to repeat <b>I+1</b> times. If <i>n</i> is not specified, the commands are repeated 65,535 times. The repeat loop may be interrupted by transferring any character. This character may not be the first character of a new command because it will be discarded. (To repeat forever, use two <b>RP</b> commands in sequence.) |
| Parameter       | n  |
| Parameter Range | 0..65535   |
| Return Value    | no   |
| Example         | ma0,ws,ma2000,ws,rp99 (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 Example: **FE0 (rtn)** : Causes motor to move in a positive direction until the E2 input is activated.  
**FE1 (rtn)** : Causes motor to move in a negative direction until the E1 input is activated.  
**FE4 (rtn)** : Causes motor to move in a positive direction until the E1 input is deactivated.

Parameter  
Parameter Range  
Return Value

n  
0,1,4  
no

Example

fe1 search for E1 in negative direction

fi

## Find Index

Description

This command is used to define a very exact homing position. The motor runs at the programmed speed until the index of the encoder occurs.  
Bit 3 indicates if the index occurs (status & \$08)  
Normal rotary encoders have one index per motor revolution!

Parameter  
Parameter Range  
Return Value

no

Example

fi

dh

## Define Home

Description

Defines the current motor position as zero position (home position).

Parameter  
Return Value

no  
no

Example

dh

## gh

### Go Home

|              |   |
|--------------|---|
| Description  | The Go Home command causes the motor to move to absolute zero position. Equivalent to an <b>MA0</b> (Move Absolute axis a to zero) command. |
| Parameter    | no  |
| Return Value | no  |
| Example      | gh  |

## WS

### Wait Stop

|                 |  |
|-----------------|--|
| Description     | Wait until axis has reached the end of the actual movement, and waits another <i>n</i> milliseconds before the execution of the next command.<br>Without parameter the wait time is 1000 ms! |
| Parameter       | n  |
| Parameter Range | 0..65535   |
| Return Value    | no   |
| Example         | mr2000,ws500,ma0   |

## 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.<br><br>Macro 0 is an auto start macro, executed after booting the controller |
| Parameter       | macro number + command string  |
| Parameter Range | 0..14 (macro number)   |
| Return Value    | no   |
| Example         | md1,ma0,ws,ma2000,ws   |

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 <i>n</i> , no action will be taken.<br><br><b>A running macro can be stopped by sending the &lt;space&gt;</b><br>dec=32 hex =\$20 |
| Parameter       | <i>n</i> , macro number  |
| Parameter Range | 0..14  |
| Return Value    | no   |
| Example         | mc1  |

tm

## Tell Macro

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

rm

## Reset Macro

|             |  |
|-------------|--|
| Description | Resets macro memory                            |
| Parameter   | 0..14macro number)                             |
| Example     | rm reset all macros<br>rm 1 reset only macro 1 |

so

## Set Analog Output

|             |  |
|-------------|--|
| Description | Sets the analog output to a value. The output is a 0 to 5 V 8 bit output. One bit is 19.5 mV |
| Parameter   | b -> value of the output 8 bit<br>0..255   |
| Example     | so255<br>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   | l -> incremental value<br>-255..255  |
| Example     | io2 ; new analog value = old value + 2<br>mc2,io1,ws1,rp ; generates a 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<br>0o0:255                                 |

ta

## Tell Analog Input

|                      |   |
|----------------------|---|
| Description          | Returns the current value of the analog inputs 1 and 2  |
| Parameter            | b channel [0 ..2]<br>0 = report both inputs<br>1..2 = reports channel #                         |
| Return Char<br>Index | A   |
| Return Value         | b [0..2]<br>n m -> value of the analog inputs 8 bit<br>n = channel 1<br>m = channel 2<br>0..255 |
| Example              | ta<br>OA0: 127 230<br>ta1<br>OA1 :127<br>ta2<br>OA2 :230  |

tc

## Tell Channel

|                      |   |
|----------------------|---|
| Description          | Returns the current status of the digital inputs and outputs.                             |
| Parameter            | b [0..4]<br>0 = report all outputs and inputs<br>1..4 = report input channel #            |
| Return Char<br>Index | H   |
| Return Value         | 0 ..4<br>n m -> value of the inputs /outputs 8 bit<br>n = outputs<br>m = inputs           |
| Example              | tc<br>OH0: 0 255 ; all outputs and inputs not active<br>tc1<br>OH1: 1 ; reports channel 1 |

cf

## Channel oFf

|              |   |
|--------------|---|
| Description  | Reset digital output(s)   |
| Parameter    | b, 0..4<br>0 = all outputs<br>1..4 = only channel #             |
| Return Value | no  |
| Example      | cf 0 or cf resets all 4 digital outputs<br>cf1 resets channel 1 |

cn

## Channel oN

|              |  |
|--------------|--|
| Description  | Set digital output (s)                                     |
| Parameter    | b, 0..4<br>0 = all outputs<br>1..4 =only channel #         |
| Return Value | no   |
| Example      | cn0 or cn sets all 4 digital outputs<br>cn1 sets channel 1 |

wn

## Wait oN

|                                      |  |
|--------------------------------------|--|
| Description                          | Stops Command Execution until Input Channel on |
| Parameter                            | b [ 1..4]<br>1..4 = input channel #            |
| Return Char<br>Index<br>Return Value | 1 ..4<br>no                                    |
| Example                              | wn1,ma1000                                     |

wf

## Wait oFf

|                                      |   |
|--------------------------------------|---|
| Description                          | Stops Command Execution until Input Channel off |
| Parameter                            | b [ 1..4]<br>1..4 = input channel #             |
| Return Char<br>Index<br>Return Value | 1 ..4<br>no                                     |
| Example                              | wn1,ma1000,wf1,ma0                              |

xn

## eXecute if oN

|                                      |   |
|--------------------------------------|---|
| Description                          | Executes Commands if Input Channel on   |
| Parameter                            | b [ 1..4]<br>1..4 = input channel #   |
| Return Char<br>Index<br>Return Value | 1 ..4<br>no   |
| Example                              | xn1,mr555,ws10 move if Input1 = on<br>xn1,xn2,mr599,ws10 move if Input 1 & 2 = on |

xf

## eXecute if oFf

|                                      |  |
|--------------------------------------|--|
| Description                          | Executes Commands if Input Channel off   |
| Parameter                            | b [ 1..4]<br>1..4 = input channel #  |
| Return Char<br>Index<br>Return Value | 1 ..4<br>no  |
| Example                              | xn1,mr555,ws10 move if Input1 = on<br>xn1,xn2,mr599,ws10 move if Input 1 & 2 = on<br>xn1,xn2,xf3,mr599,ws10 move if In 1 & 2 = on & In 3 = off |

# Trigger Functions:

The SuprMotr controller is able to generate position breakpoint trigger outputs. This helps to synchronize lasers or other measurement devices. The accuracy is perfectly synchronized with the encoder, with the limitation that multiple triggers are limited to a maximal frequency of **500 Hz!**

Example: The application needs every 5  $\mu\text{m}$  a trigger signal. The first trigger should occur at absolute position 10  $\mu\text{m}$ .  
1  $\mu\text{m}$  = 2 encoder increments  
The output for the trigger is Out 1, the pulse width = 2.5  $\mu\text{sec}$

actual position is : 0  
sb1  
pw0  
bd20  
bi10  
mr500

Maximum speed should be: 500 [1/s] x 10 [increments] = 5000 [Increments/s]

sb

## SetBreakpoint port

|              |   |
|--------------|---|
| Description  | Assigns the trigger output port (digital output 1..4) |
| Parameter    | b, 1..4<br>1..4 = port number                         |
| Return Value | no  |
| Example      | sb1 ; defines output 1 as trigger output              |

bd

## Breakpoint Define

|              |  |
|--------------|--|
| Description  | Defines the absolute position for the first breakpoint<br>Please note, the stage should be a little before the first trigger!<br>It also sets bit 0 of the 4th byte reported in TS |
| Parameter    | l<br>+/-1 073 741 843  |
| Return Value | no   |
| Example      | ma19900,sb1,bi100,bd20000,ma30000<br>first trigger at position 20000, next every 100 increments  |

bi

## define Breakpoint Increment

|              |  |
|--------------|--|
| Description  | Defines the relative distance (breakpoint increment) between the trigger points.   |
| Parameter    | I<br>+/-1 073 741 843  |
| Return Value | no   |
| Example      | ma19900,ws,sb1,bi100,db20000,ma30000<br>first trigger at position 20000, next every 100 increments<br><br>ma20100,ws,sb1,bi-100,db20000,ma0<br>first trigger at position 20000, next every -100 increments |

pw

## define Pulse Width

|              |   |
|--------------|---|
| Description  | Defines the pulse width of the trigger signal<br>0= 2.5µsec<br>1..255 = 1..255 ms |
| Parameter    | b<br>0..255   |
| Return Value | no  |
| Example      | pw1 ; pulse width = 1ms   |

## Some examples:

### Homing

These commands should be entered, separated by commas, as a single compound or macro. They may also be entered as single commands, but then there is no need to use the WS commands.

DV10000  
FE1  
WS  
DV500  
FE4  
WS  
DH

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

### Demo

The same comments apply to these commands.

MA0  
WS500  
MR500  
WS100  
RP49

Move to absolute position 0  
Wait until after move complete 500 ms  
Move relative 500 increments  
Wait until after move complete 100 ms  
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
- 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
- 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 (format as entered)  
MD11, MR+0000000500,WS+0000000100,RP+000000009 (format as reported)
- Macro 12:** Same as macro 11 in negative direction  
MD12, MR500,WS100,RP5

# Command Short Form

| Read command   | Reply char | Set command  |
|--|------------|--|
| TS Tell Status   | S:         |  |
| MS Move Status   | M:         |  |
| HE Print list of commands  |            |  |
| TE Tell Error (distance from target)                                   | E:         |  |
| TP Tell Position   | P:         |  |
| TT Tell Target   | T:         |  |
| TF Tell Following error (distance between position and dynamic target) | F:         |  |
| TD Tell Dynamic target   | N:         |  |
| TV Tell actual Velocity  | V:         |  |
| GV Get Velocity  | Y:         | DV Define Velocity                                       |
| GA Get Acceleration  | A:         | DA Define Acceleration                                   |
| GP Get Proportional gain   | G:         | DP Define Proportional gain                              |
| GI Get Integral gain   | I:         | DI Define Integral gain                                  |
| GD Get Derivative gain   | D:         | DD Define Derivative gain                                |
| GL Get Integral Limit  | L:         | DL Define integral Limit                                 |
|  |            | DE Define maximum following Error                        |
| VE Display Version number  |            |  |
| TI Tell the Iteration number   | X:         |  |
| TB Tell Board address  | B:         | DB Define board address                                  |
| CS Perform self-test checksum  | C:         |  |
|  |            | LN Limit switch operation on                             |
|  |            | LF Limit switch operation off                            |
|  |            | LL Limits switches active low                            |
|  |            | LH Limit switches active high                            |
|  |            |  |
|  |            | UD Update  |
|  |            | DM Input and output in Decimal format                    |
|  |            | HM Input and output in Hex format                        |
|  |            | RT Reset all parameters to default and do power-up start |
|  |            | EF Turn off echo to RS-232 port                          |
|  |            | EN Turn on echo to RS-232 port                           |
|  |            |  |
|  |            | AB Abort Motion  |
|  |            | MN Motor On  |
|  |            | MF Motor off   |
|  |            | WA Wait specified time                                   |
|  |            | MR Move Relative   |
|  |            | MA Move Absolute   |
|  |            | DH Define Home   |
|  |            | GH Go Home   |
|  |            | RP Repeat from beginning of line                         |
|  |            | WS Wait until end of trajectory                          |
|  |            | FE Find Edge   |
|  |            | FI Find Index  |
|  |            |  |
| TM Tell Macro (1..31)  |            | MD Macro Definition                                      |
| TZ Tell Macro Zero   |            |  |
|  |            | MC Macro Command (or EM)                                 |
|  |            | RM Reset Macro   |
|  |            | RZ Reset Macro 0   |
| TO Tell Analog Output  | O          | SO Set Analog Output                                     |
| TA Tell Analog Input   | A          |  |
| TC Tell Channel  | H          | CF Channel Off   |
|  |            | CN Channel on  |
|  |            |  |

## Factory Settings :

|  |  |
|--|--|
| DV Define <b>V</b> elocity                     |  |
| DA Define <b>A</b> cceleration                 |  |
| DP Define <b>P</b> roportional gain            |  |
| DI Define <b>I</b> ntegral gain                |  |
| DD Define <b>D</b> erivative gain              |  |
| DL Define integral <b>L</b> imit               |  |
| DE Define maximum following <b>E</b> rror      |  |
|  |  |
| EN Turn <b>o</b> n <b>e</b> cho to RS-232 port |  |
|  |  |
|  |  |
|  |  |