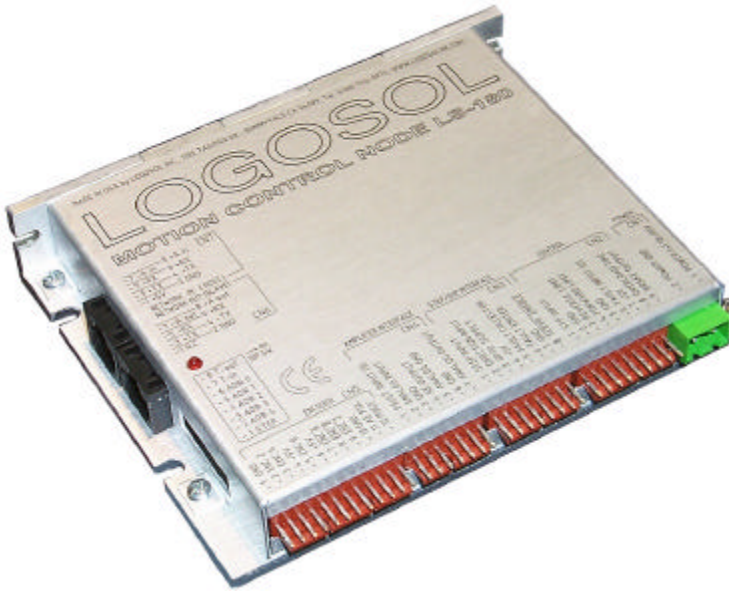
Logosol Motion Control Node with Step & Direction Interface LS-180

Doc # 712180001 / Rev. 1.08, 01/16/2001

Features

- **Encoders supported**
  - **Differential**
  - **Single ended**
- **SERVO AMPLIFIER interface**
  - **11 bit 0 ÷ ±10V analog output**
  - **8 bit 0.5 ÷ ±20V analog input**
  - **FAULT input**
  - **AMPLIFIER ENABLE output**
- **32-bit position, velocity, acceleration, 16-bit PID filter gain values**
- **Servo rate 2 kHz**
- **Encoder transition rate up to 2MHz**
- **Optoisolated STEP/DIRECTION interface**
  - **STEP input**
  - **DIRECTION input**
  - **SERVO ENABLE input**
  - **FAULT output**
- **Emergency stop input**
- **Forward and reverse over travel inputs**
- **RS-485 command interface**
- **Communication speed 19.2 - 115.2 KBps**
- **Command rate up to 1000/sec**
- **Small footprint (5” x 4” x 0.85”)**



Description

LS-180 is a single-axis motion control node with ±10V analog output, designed for applications using a standard servo amplifier. The node can work in NETWORK or STEP/DIRECTION modes.

Up to 31 motion control nodes can be controlled over a multi-drop full duplex RS-485 network in a distributed motion control environment. Standard RJ-45 connectors and commercially available cables are used for daisy chaining of the modules.

LS-180 is equipped with various safety features such as an over travel limit switch inputs, emergency stop input, encoder presence control and motor current monitoring.

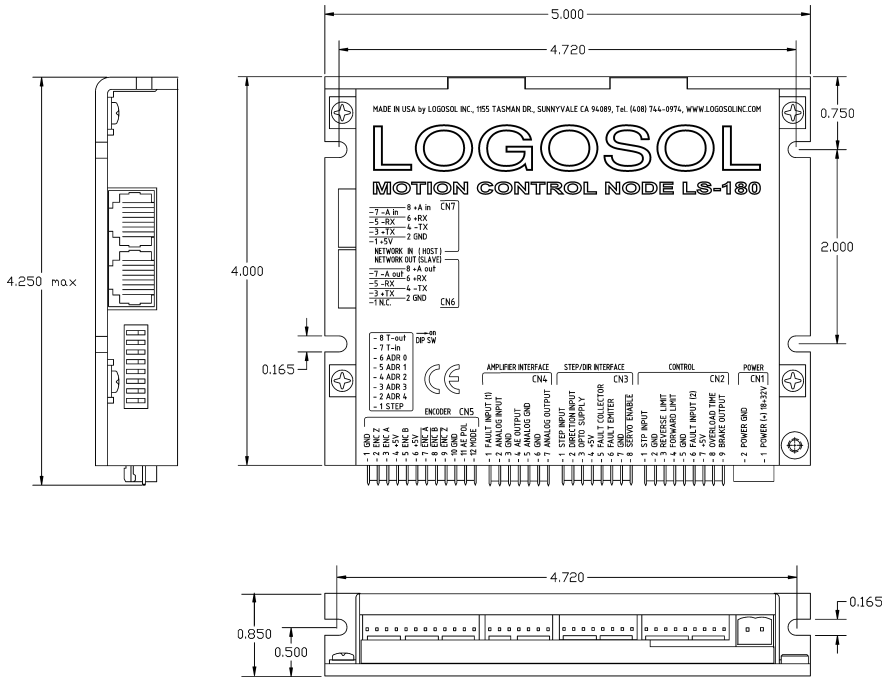
TECHNICAL SPECIFICATIONS rated at 25°C ambient, POWER (+)=24VDC

POWER SUPPLY VOLTAGE	18 to 32V DC, 35V Absolute Maximum
SERVO RATE	0.512 msec
SERIAL BAUD RATE	19.2 – 115.2 Kbps (Faster communication rates are possible at lower servo rates)
ANALOG INPUT Voltage Resolution Input impedance Mode	 ±0.5 ÷ ±20V, Adjustable 8bit 10K Asynchronous
ANALOG OUTPUT Voltage Resolution Output short current Adjustments	 0 ÷ ±10V 11bit 10mA 0 ÷ ±12V max output voltage, ±100mV offset
DIGITAL INPUTS CONTROL: FORWARD LIMIT, REVERSE LIMIT, STP IN, STEP/DIRECTION Interface: STEP INPUT, DIRECTION INPUT, $\overline{\text{SERVO ENABLE}}$ AMPLIFIER Interface: FAULT INPUT(1), FAULT INPUT(2)	 LOmin=-1V, HImax=48V, Imax=5mA Optoisolated, Imax=10mA LOmin=-0.5V HImax=5V, Imax=5mA
ENCODER Inputs	Quadrature with index (Differential or Single Ended) TTL with 1K pull-up to 5V
DIGITAL OUTPUTS BRAKE OUTPUT: Max voltage applied to output Max current load FAULT OUTPUT: Max voltage applied to the output Max current load AE OUTPUT	 Open collector 48V 0.3A Optoisolated 48V 3mA TTL output with 100 Ohm resistor in series
INDICATORS Red LED (two intensity levels)	Power 'ok' – low intensity Servo Amplifier 'on' – high intensity
THERMAL REQUIREMENTS Storage temperature range Operating temperature range	 –30 to +85 °C 0 to 45 °C
MECHANICAL Size Weight	 L=5.00", H=4.00", D=0.85" 0.66lb. (300gr.)
MATING CONNECTORS Power Encoder Amplifier interface Step/Dir Interface Control Communication	 Magnum EM2565-02-VL or Phoenix MSTB 2.5/2-ST-5.08 Molex 22-01-3127 housing with 08-50-0114 pins (12 pcs.) Molex 22-01-3077 housing with 08-50-0114 pins (7 pcs.) Molex 22-01-3087 housing with 08-50-0114 pins (8 pcs.) Molex 22-01-3097 housing with 08-50-0114 pins (9 pcs.) 8 pin RJ-45
+5V SOURCE Max current	200mA for all three output pins combined

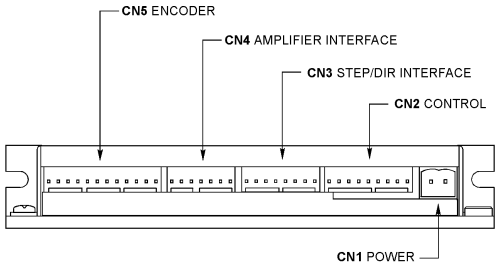
ORDERING GUIDE

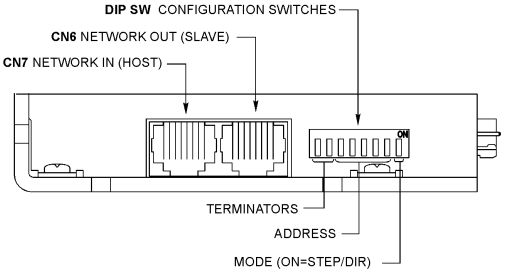
PART NUMBER	MODEL	DESCRIPTION
912180001	LS-180	Motion Control Node
230601009	LS-180-CN	Mating connector kit

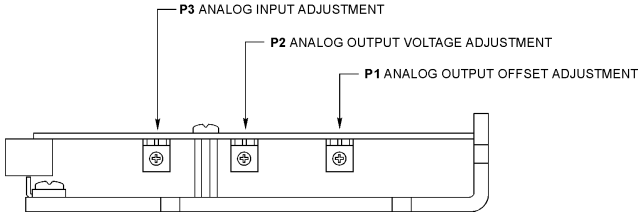
DIMENSIONAL DRAWING



SERVO DRIVE LAYOUT







CONNECTORS AND PINOUT

DIP SW – DIP SWITCH

SW	SIGNAL	DESCRIPTION
1	STEP	Mode select switch / 'on'=STEP/DIR mode
2	ADR4	Address select switch 4
3	ADR3	Address select switch 3
4	ADR2	Address select switch 2
5	ADR1	Address select switch 1
6	ADR0	Address select switch 0
7	T-in	Transmit line terminator
8	T-out	Receive line terminator

CN1 – POWER AND MOTOR CONNECTOR

PIN	SIGNAL	DESCRIPTION
1	POWER (+)	18 ÷ 32V power supply, positive terminal
2	POWER GND*	Power supply ground

* *POWER GND and GND are electrically connected. Drive's case is isolated from drive circuitry and can be grounded externally.*

CN2 – CONTROL

PIN	SIGNAL	DESCRIPTION
1	STP IN	Stop input (disables servo amplifier)
2	GND*	Signal ground
3	REVERSE LIMIT	Reverse limit input
4	FORWARD LIMIT	Forward limit input
5	GND*	Signal ground
6	PEAK CURRENT	Overcurrent limit
7	+5V**	Signal power supply
8	OVERLOAD TIME	Overcurrent limit timeout
9	BRAKE OUTPUT	Brake output. Open collector 48V/0.3A

CN3 – STEP/DIR INTERFACE

PIN	SIGNAL	DESCRIPTION
1	STEP INPUT	Step input
2	DIRECTION INPUT	Direction input
3	OPTO SUPPLY	STEP/DIR interface power supply (+)
4	+5V**	Signal power supply
5	FAULT COLLECTOR	Fault output (+)
6	FAULT EMITTER	Fault output (-)
7	GND*	Signal ground
8	SERVO ENABLE	Servo enable input (active low)

CN4 – AMPLIFIER INTERFACE

PIN	SIGNAL	DESCRIPTION
1	FAULT INPUT (1)	Input, connected to Servo Amplifier FAULT output
2	ANALOG INPUT	Input, connected to Servo Amplifier current monitor output
3	GND*	Signal ground
4	AE OUTPUT	Output for enabling Servo Amplifier
5	ANALOG GND*	Analog output ground
6	GND*	Signal ground
7	ANALOG OUTPUT	±10V analog output

* *POWER GND and GND are electrically connected. Drive's case is isolated from drive circuitry and can be grounded externally.*

** *200mA Max current for all +5V outputs combined.*

CN5 – ENCODER

PIN	SIGNAL	DESCRIPTION
1	GND*	Encoder ground
2	ENC Z	Encoder index (+)Z
3	ENC A	Encoder phase (+)A
4	+5V**	Encoder power supply
5	ENC B	Encoder phase (+)B
6	+5V**	Commutator power supply
7	$\overline{\text{ENC A}}$	Encoder phase (-)A
8	$\overline{\text{ENC B}}$	Encoder phase (-)B
9	$\overline{\text{ENC Z}}$	Encoder index (-)Z
10	GND*	Encoder shield
11	AE POL	AE output polarity
12	MODE	Encoder mode – differential if open

CN6 – NETWORK OUT (SLAVE)

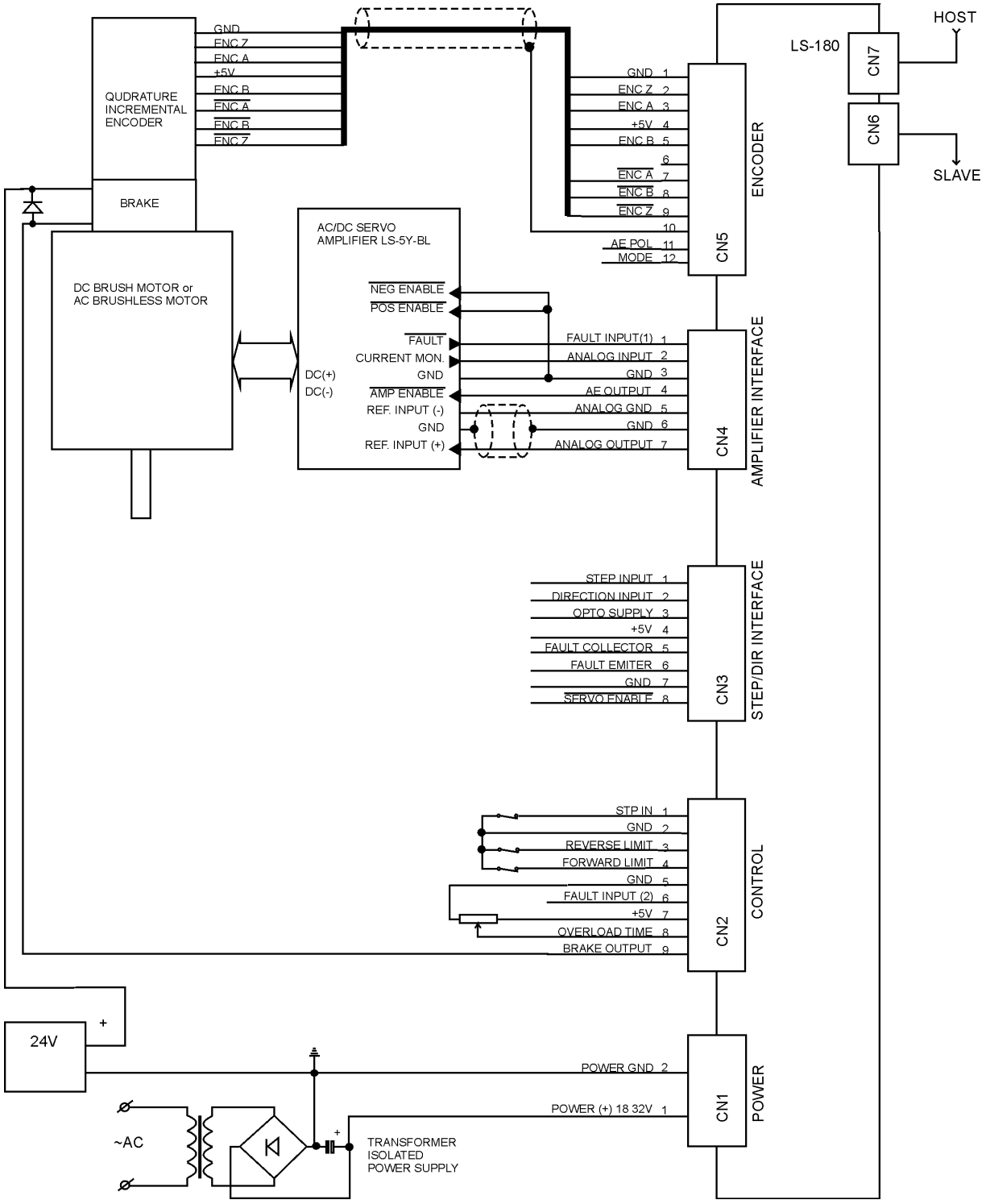
PIN	SIGNAL	DESCRIPTION
1	N.C.	Not connected
2	GND*	Interface ground
3	+TX	(+) Transmit data
4	-TX	(-) Transmit data
5	-RX	(-) Receive data
6	+RX	(+) Receive data
7	-A out	(-) Address output
8	+A out	(+) Address output

CN7 – NETWORK IN (HOST)

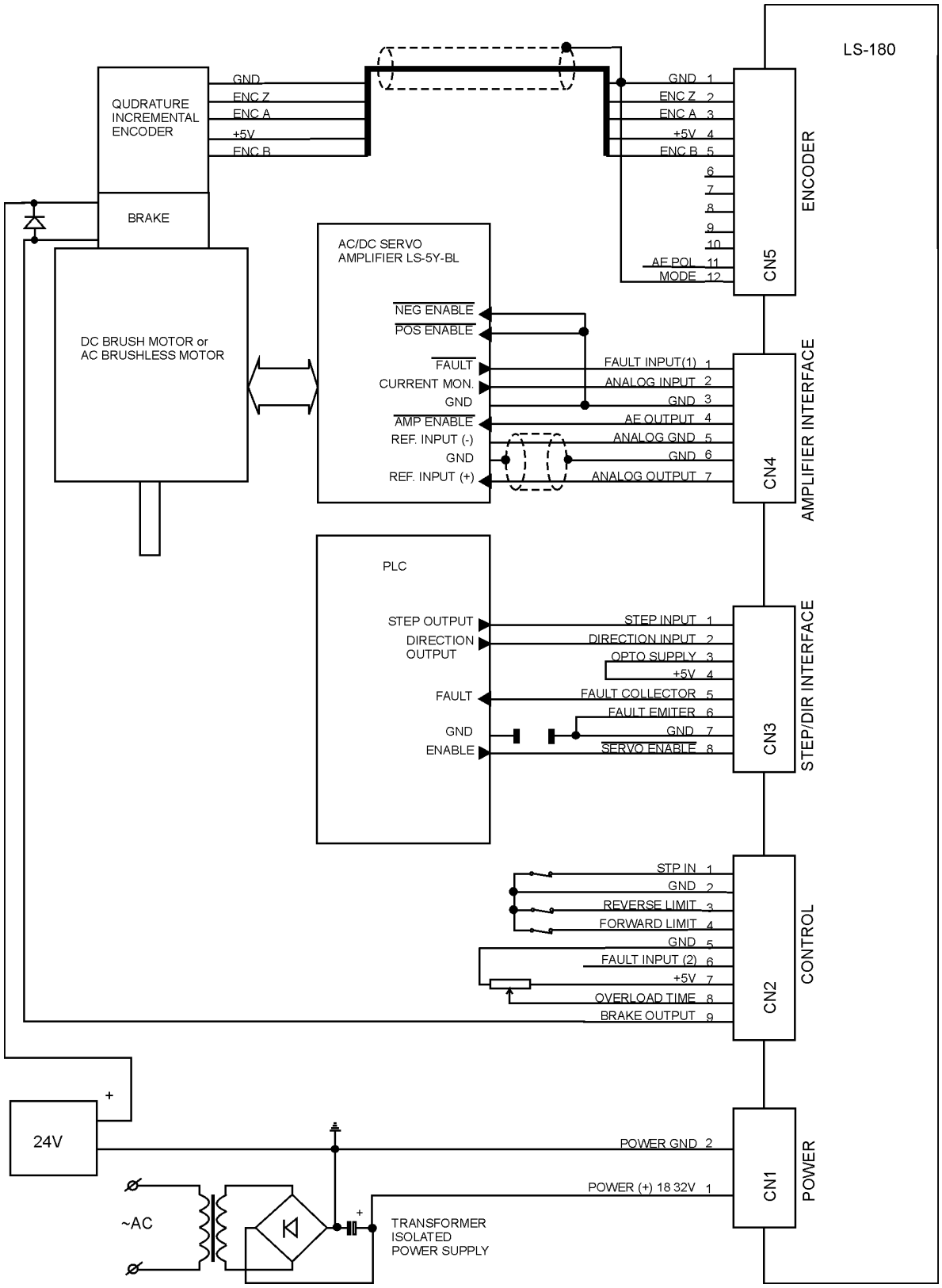
PIN	SIGNAL	DESCRIPTION
1	+5V**	RS-232 adapter power supply
2	GND*	Interface ground
3	+TX	(+) Transmit data
4	-TX	(-) Transmit data
5	-RX	(-) Receive data
6	+RX	(+) Receive data
7	-A in	(-) Address input
8	+A in	(+) Address input

* *POWER GND and GND are electrically connected. Drive's case is isolated from drive circuitry and can be grounded externally.*
** *200mA Max current for all +5V outputs combined.*

SAMPLE APPLICATION in RS-485 network command mode



SAMPLE APPLICATION in STEP/DIRECTION mode



LOGOSOL LS-180 QUICK START GUIDE

Hardware Setup

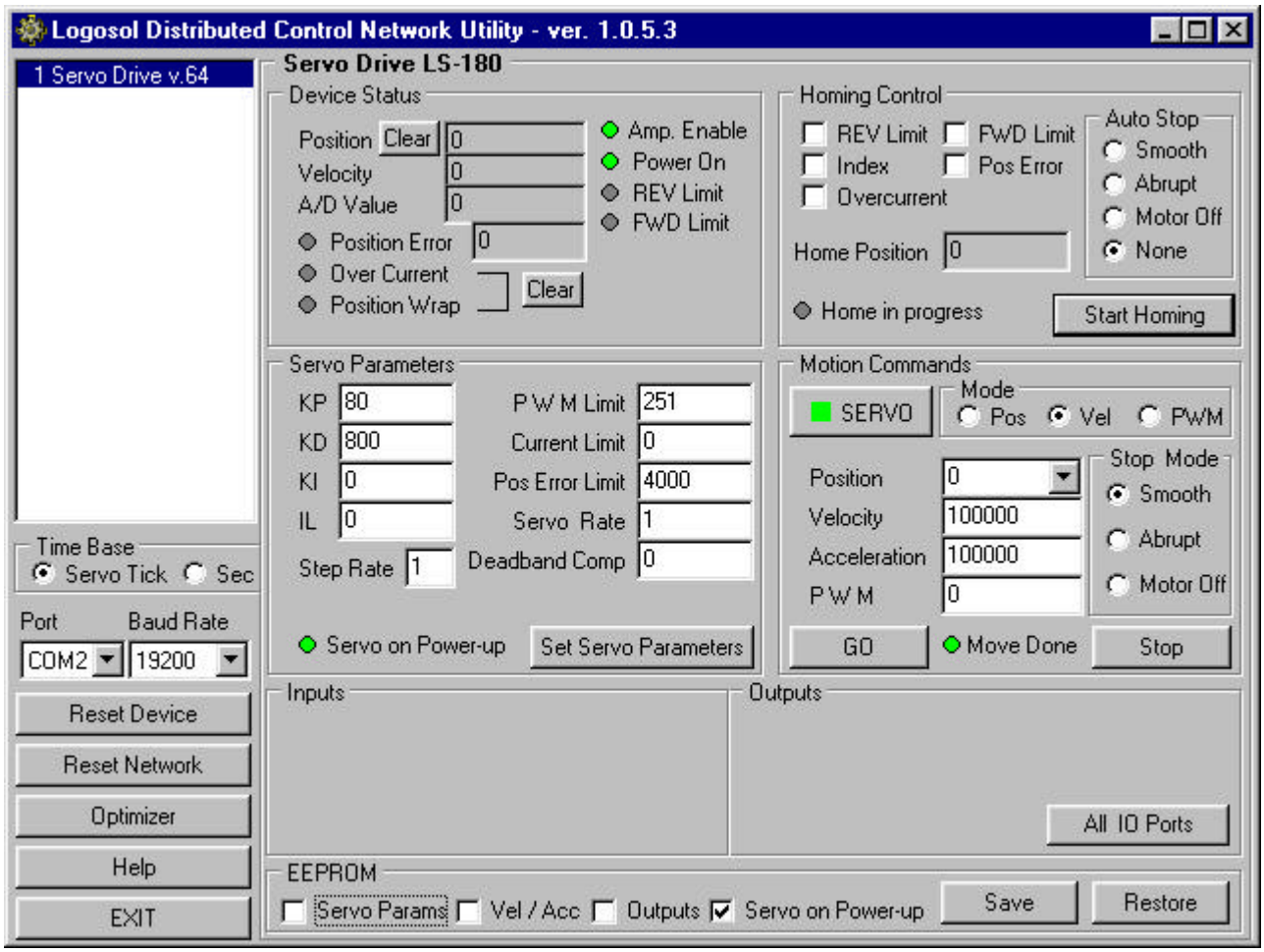
1. Connect power supply to LS-180.*
2. Connect your servo amplifier, encoder, and any other device you may have.
3. Connect RS-232 adapter and RJ-45 network cable between LS-180 and host computer.

Software Installation

1. Installation and using Logosol Distributed Control Network Utility

A. Installation

1. Insert the Logosol Distributed Control Network Utility installation disk into the floppy drive.
2. Select Run from the Windows 95/98/NT Start menu.
3. Type a:\dcnsetup and then click OK (a: represents the drive letter).
4. The installation wizard will guide you through the setup process.



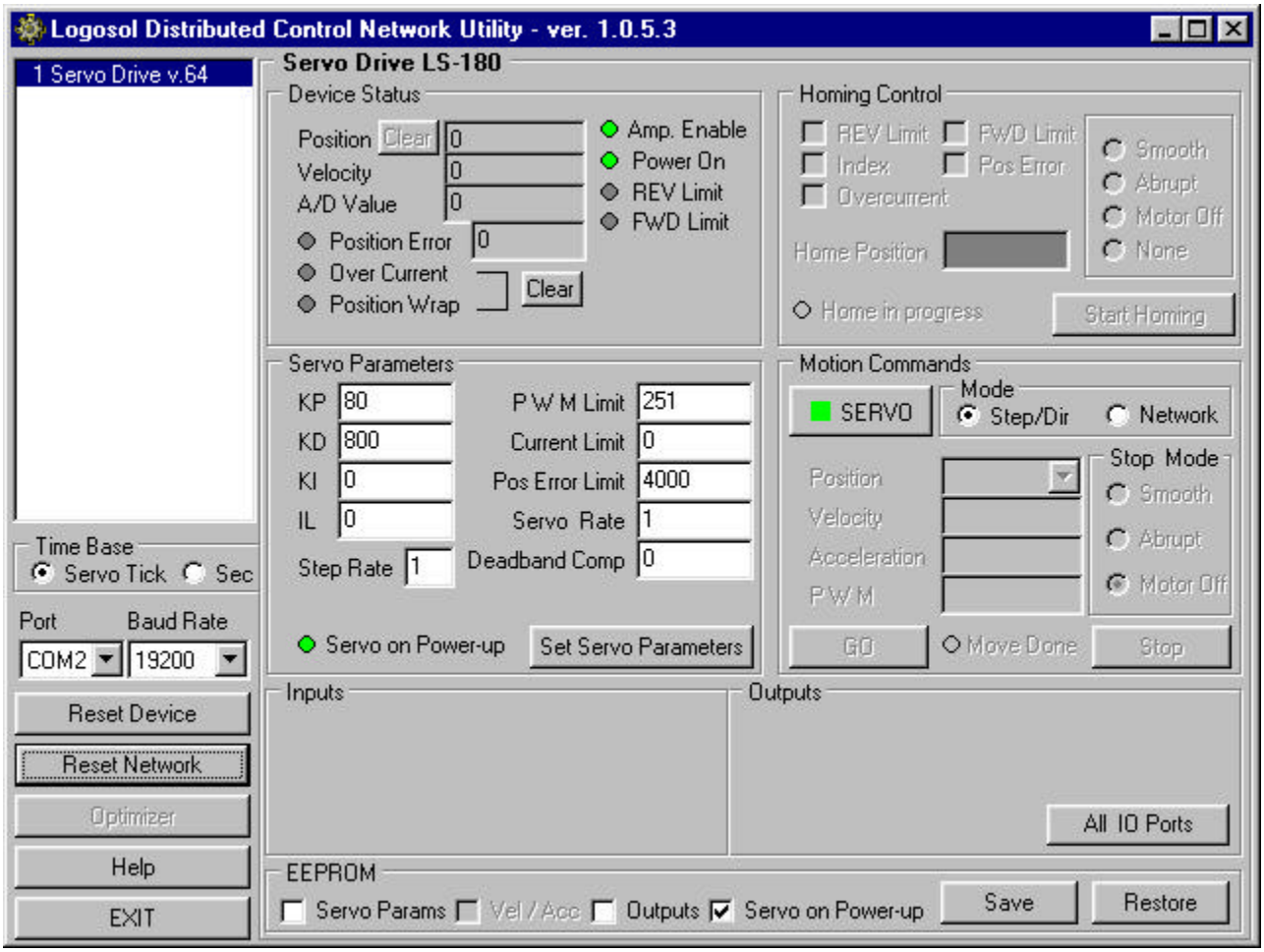
* *When installing multiple drives, each drive should have its own twisted pair cable running to terminals of the power supply. Don't "daisy-chain" cables from one drive to the next. This will aggravate cable noise and cause the noise from one drives to add to each other. The "star"-wiring configuration will minimize wiring noise.*

B. Initial Connection to the Host

1. Turn on the power supply.
2. Run the Logosol Distributed Control Network Utility.
3. Choose the proper COM port
4. Click “SERVO” button.
5. Click “GO” button. The motor should rotate slowly in positive direction. Click “Stop” to interrupt the motion. More information about using LDCN utility is available in LDCN Help.

C. STEP/DIRECTION mode quick start

1. Execute steps **A** and **B**.
2. Set “Step rate” edit box to 1 to set STEP multiplier value.
3. Check “Servo params” check box and click “Save” button in “EEPROM” panel.
4. Turn off the power supply.
5. Set STEP/DIR mode (STEP switch ‘on’).
6. Turn on the power supply and you can control the drive using STEP/DIR interface.



When the controller is in step and direction mode, most of the controls are disabled. Clicking on “Network” radio button in “Mode” panel performs switching to RS-485 network command mode. Switching back to step and direction mode is possible only by resetting the controller.

2. Installation and using Logosol Motion Control Center

A. Installation

1. Insert the Logosol Motion Control Center installation disk into the floppy drive.
2. Select Run from the Windows 95/98/NT Start menu.
3. Type a:\mccsetup and then click OK (a: represents the drive letter).
4. The installation wizard will guide you through the setup process.

B. Initial Connection to the Host

1. Turn on the power supply.
2. Run the Logosol Motion Control Center software.
3. From the **Connection** menu select **Terminal**. This will open a terminal window. From the **Target** pull-down list select either RS-485-COM1 or RS-485-COM2 corresponding to the one used to communicate with LS-180. Press the Return key to verify that the connection is established and the command prompt (>) appears on the terminal window.
4. Type **INI** at the command prompt followed by Return to initialize the controller. It may take few seconds to complete the process.
5. Type **KP A1=20** to set the proportional coefficient, then set the velocity, acceleration and error limit by the following commands: **VEL A1=100**, **ACC A1=1000** and **MAX A1=8000**.
6. Type **SER** to close the servo loop, then **FOR A1** to instruct the controller to rotate the motor forward. Type **GO A1** to initiate the motion. The motor should rotate slowly in positive direction. Type **STO** to interrupt the motion and refer to the following MCL* Terminal Interpreter Command Set for executing various motion commands and parameters settings.

C. STEP/DIRECTION mode quick start

1. Execute steps **A** and **B.**
2. Type **WRP A1=00 00 00 01** to set STEP multiplier value.
3. Type **SVP A1** to save all parameters in EEPROM.
4. Turn off the power supply.
5. Set STEP/DIR mode (STEP switch ‘on’).
6. Turn on the power supply and you can control the drive using STEP/DIR interface.

* The information about MCL Terminal Interpreter command set is also available in MCC Help Menu Index (Advanced Features /Command set for Distributed Servo Drive maintenance).

MCL TERMINAL INTERPRETER COMMAND SET

The following commands are available from the Terminal prompt:

POS – returns and sets current position:
1) "**POS**" returns the positions of all the axes;
2) "**POS A1**" returns the position of axis A1;
3) "**POS A1=Y**" sets the position of axis A1 to the specified value Y.

MAX – returns and sets maximal error (the difference between calculated and real positions):
1) **"MAX"** returns the maximal errors of all the axes;
2) "**MAX A1**" returns the maximal error of axis A1;
3) "**MAX A1=Y**" sets the error limit of axis A1 to the specified value 0<Y<16383.

VEL – returns and sets velocity in velocity mode or goal velocity in trapezoidal mode:
1) "**VEL**" returns the velocities of all the axes;
2) "**VEL A1**" returns the velocity of axis A1;
3) "**VEL A1**=**Y**" sets the velocity of axis A1 to the specified value Y.

ACC – returns and sets acceleration:
1) "**ACC**" returns the accelerations of all the axes;
2) "**ACC A1**" returns the acceleration of axis A1;
3) "**ACC A1=Y**" sets the acceleration of axis A1 to the specified value Y.

XST – displays status information:
1) "**XST**" displays status info for all the modules in the network;
2) "**XST A1**" displays status info for module A1.

Status information for module LS-180 includes:
1) Address
2) Status byte
3) Auxiliary status byte
4) Encoder position
5) Device ID
6) Version number
7) Auxiliary input bytes
8) ADC value.

STA – displays MCL compatible status:
1) "**STA**" returns MCL compatible status of all the axes;
2) "**STA A1**" returns MCL compatible status of axis A1.

Note: Status 00000400 - position reached;
00000080 - servo off.

INI – resets the network and assigns individual and group addresses (individual addresses are assigned for modules without fixed addresses).

STO – stops abruptly:
1) "**STO**" stops the movement of all the axes;
2) "**STO A1**" stops the movement of axis A1.

HAL – stops smoothly with specified acceleration:
1) "**HAL**" stops the movement of all the axes;
2) "**HAL A1**" stops the movement of axis A1.

KP – returns and sets proportional PID filter parameter:
1) "**KP**" returns KP parameters for all the axes;
2) "**KP A1**" returns KP parameter for axis A1;
3) "**KP A1=Y**" sets KP of axis A1 to the specified value Y.

Note: KP is set to 0 after power-up.

KI – returns and sets integral PID filter parameter:
1) "**KI**" returns KI parameters for all the axes;
2) "**KI A1**" returns KI parameter for axis A1;
3) "**KI A1=Y**" sets KI of axis A1 to the specified value Y.

Note: KI is set to 0 after power-up.

KD – returns and sets deferential PID filter parameter:
1) "**KD**" returns KD parameters for all the axes;
2) "**KD A1**" returns KD parameter for axis A1;
3) "**KD A1=Y**" sets KD of axis A1 to the specified value Y.

Note: KD is set to 0 after power-up.

IL – returns and sets integral limit PID filter parameter:
1) "**IL**" returns IL parameters for all the axes;
2) "**IL A1**" returns IL parameter for axis A1;
3) "**IL A1=Y**" sets IL of axis A1 to the specified value Y.

Note: IL is set to 0 after power-up.

ABS – sets absolute motion position for specified axis – "**ABS A1=Y**".

REL – sets relative motion position for specified axis – "**REL A1=Y**".

PWM – returns and sets PWM value in range -255÷255:
1) "**PWM**" returns the PWM values of all modules;
2) "**PWM A1**" returns the PWM value of axis A1;
3) "**PWM A1=Y**" sets PWM of axis A1 to value Y.

FOR – sets forward motion in velocity mode:
1) "**FOR A1**" sets the motion of axis A1 with the current speed;
2) "**FOR A1=Y**" sets the motion of axis A1 with the specified speed Y.

REV – sets reverse motion in velocity mode:
1) "**REV A1**" sets the motion of axis A1 with the current speed;
2) "**REV A1=Y**" sets the motion of axis A1with the specified speed Y.

GO – starts motion in previously defined mode and parameters:
1) "**GO**" starts motion of all axes;
2) "**GO A1**" starts motion of axis A1.

CLI – returns and sets current limit:
1) "**CLI**" returns the current limits of all the axes;
2) "**CLI A1=0**" disables current limit of axis A1;
3) "**CLI A1=X**" sets current limit of axis A1 to value X=0÷127.

Note: CL=2X+1. CL – current limit parameter of Set Gain command.

SER – enables servo:
1) "**SER**" enables servo of all connected modules;
2) "**SER A1**" enables servo of module A1.

NOS – disables servo:
1) "**NOS**" disables servo of all connected modules;
2) "**NOS A1**" disables servo of module A1.

INX – finds index:
1) "**INX A1 F**" finds index of axis A1 in forward direction;
2) "**INX A1 R**" finds index of axis A1 in reverse direction.

Note: if the motor makes more than one revolution to find index, decrease the speed.

FLS – finds limit switch:
1) "**FLS A1 F**" finds limit switch of axis A1 in forward direction;
2) "**FLS A1 R**" finds limit switch of axis A1 in reverse direction.

HEX – hex command mode – sends a low-level command written in hexadecimal format. For more information about command format refer to "Command Description" section in this document. Start byte (AA) and checksum byte are generated by the MCL interpreter:
"**HEX 01 05**" sends *Start Motion* command (code 0x05) for module with address 1.

BDR – sets baud rate. Possible baud rate values are 9600, 19200, 57600 and 115200:
"**BDR 115200**" – sets baud rate to 115.2 Kbps.

Note: baud rate is set to 19.2 Kbps after power-up.

HIS – shows the history of the recently used commands, their hexadecimal codes and the returned status packets.

EXE – executes a text file containing sequence of MCL commands:
"**EXE control.dat**" – executes the command sequence from "*control.dat*" file in current directory.

VER – returns MCL interpreter version.

NET – displays the number and types of all modules in the network and their addresses.

BRL – releases the motor brake to enable manually positioning:
1) "**BRL**" releases the brakes of all connected modules;
2) "**BRL A1**" releases the brake of module A1.

WRP – returns and sets four bytes to auxiliary output ports:

1) "**WRP**" returns the values of auxiliary output ports of all modules;
2) "**WRP A1**" returns the values of auxiliary output ports of module A1;
3) "**WRP A1=Y0 Y1 Y2 Y3**" sets the values of auxiliary output ports of module A1 to values Y0, Y1, Y2 and Y3 (Yn = 0 ÷ FFh).

RDP – returns three bytes of auxiliary input ports in hexadecimal format:

1) "**RDP**" returns the values of auxiliary input ports of all modules;
2) "**RDP A1**" returns the values of auxiliary input ports of module A1.

SVP – saves the gain and trajectory parameters and auxiliary output ports values in EEPROM memory of LS-180:

1) "**SVP**" saves the parameters of all modules;
2) "**SVP A1**" saves the parameters of module A1.

Note. Trajectory parameters (velocity and acceleration) are saved after a motion with these parameters was started.

RSP – restores the gain and trajectory parameters and auxiliary output ports values from EEPROM memory of LS-180:

1) "**RSP**" restores the parameters of all modules;
2) "**RSP A1**" restores the parameters of module A1.

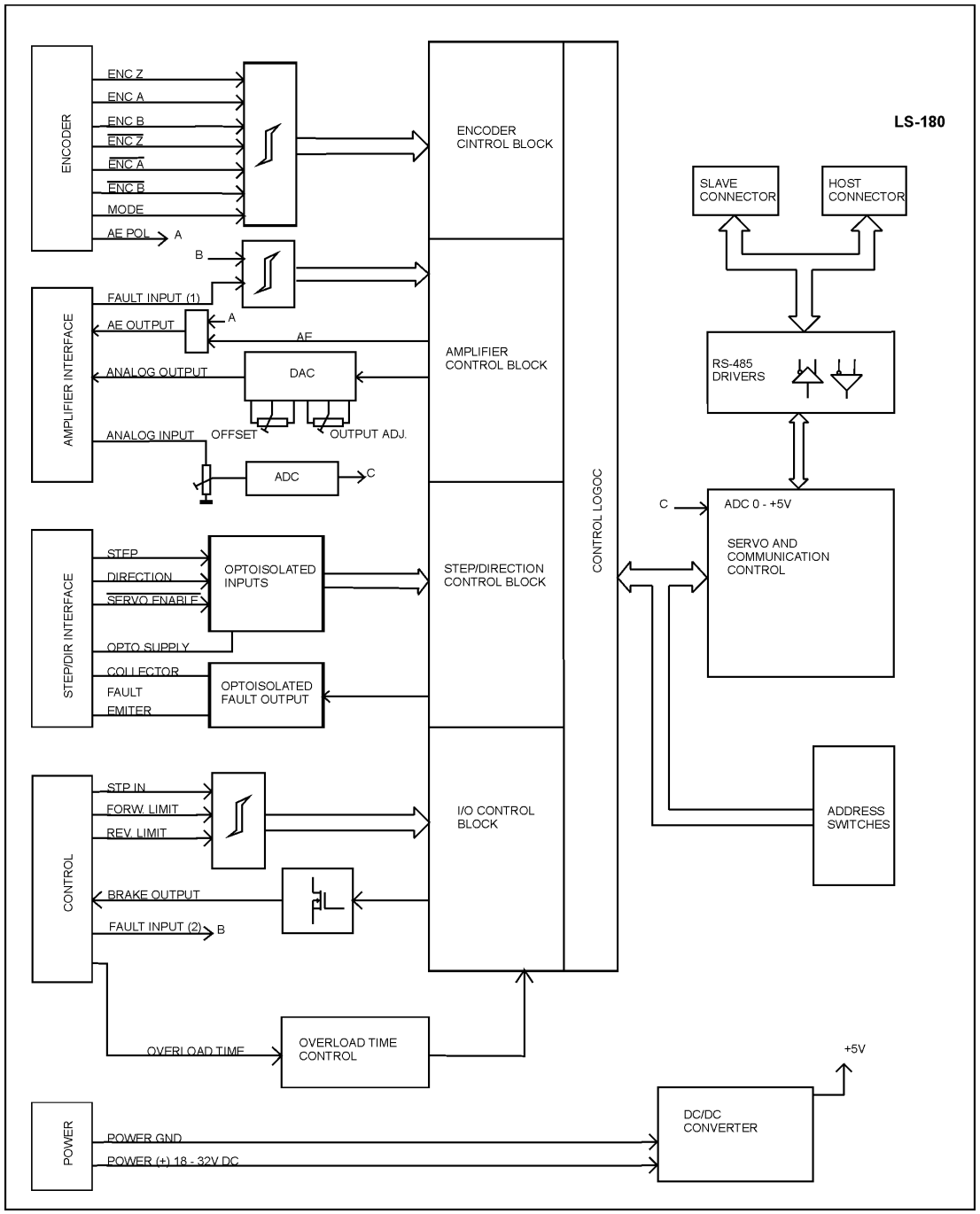
DBC – returns and sets amplifier deadband compensation value:

1) "**DBC**" returns the deadband compensation values of all the axes;
2) "**DBC A1**" returns the deadband compensation value of axis A1;
3) "**DBC A1**=**Y**" sets the deadband compensation value of axis A1 to the value Y.

LS-180 ARCHITECTURE

Overview

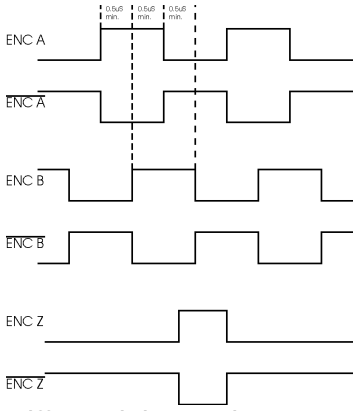
The LS-180 motion control node is a highly integrated module including a motion controller, analog voltage output, analog voltage input, serial communication interface, step and direction interface, incremental encoder interface and limit switch inputs. The motion control node is designed so that up to 31 controllers can be daisy-chained and connected directly to a single standard serial port (RS-232 adapter may be necessary).



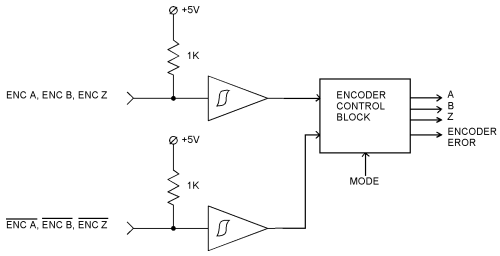
Functional Diagram

Encoder Interface

LS-180 is designed to work with quadrature incremental encoder. Both differential and nondifferential encoder types may be used. Because of that LS-180 encoder inputs are designed with two separate receivers for each phase instead with one differential. MODE input located at the ENCODER connector selects encoder type. For nondifferential encoders MODE must be tied to ground. For differential encoder MODE input must be left unconnected.



Differential encoder CW rotation

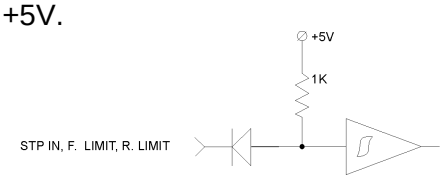


Encoder Interface

Using encoders with differential wiring is highly recommended because of the better noise immunity. In addition, with differential encoder LS-180 provides protection against motor windup caused by missing or broken encoder wire. A dedicated logic controls the presence of all encoder signals.

Control inputs

There are 3 control inputs - STP IN, FORWARD LIMIT and REVERSE LIMIT. STP IN may be used only as “STOP” input. Limit inputs may be used as “HOME” switches, limit switches or as general-purpose inputs. (Refer to *“I/O Control”* and *“Set Homing Mode”* commands in the *Command Description* section in this document) All the inputs are implemented with pull-up resistors 1K to +5V.

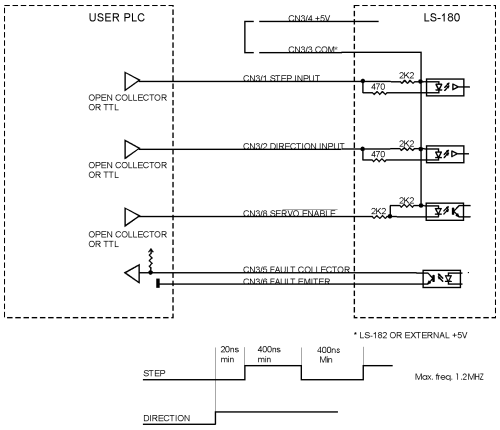


Limit Switches and Stop Input

In STEP/DIR mode FORWARD LIMIT and REVERSE LIMIT are active high limit switches. If any of limit switches is open the movement in this direction is disabled.

STEP/DIRECTION interface

Three optoisolated inputs are available for control in STEP/DIRECTION mode. OPTO SUPPLY pin must be connected to LS-180 +5V or to external source. STEP and DIRECTION inputs are equipped with high-sped opto-coplers. Open (high) DIRECTION input corresponding to CW rotation of the motor. SERVO ENABLE input is active low input and must be ON=low to enable motor rotation. If it is open the STEP input is disabled. If motion control node is in Fault condition (refer to *Status Bits and Led* section in this document) the normal operation will be restored by negative (high to low) transition of SERVO ENABLE input.



STEP/DIRECTION interface

FAULT is optoisolated output and will be open when the motion control node is in Fault Condition.

Amplifier interface

Amplifier interface is provided for Servo Amplifier control. The interface include:

- FAULT INPUT (1) – connected to Servo Amplifier FAULT output. FAULT INPUT (1) and FAULT INPUT (2) are used for Servo Amplifier control as follows:

FAULT INPUT (1)	FAULT INPUT (2)	AMPLIFIER STATUS
1 (OPEN)	1	OK
1	0 (TIED TO GND)	AMPLIFIER FAULT
0	1	AMPLIFIER FAULT
0	0	OK

Note: For AMPLIFIER FAULT code description - refer to Status Bits and Led section in this document.

- ANALOG INPUT - Adjustable (P1) analog input (±0.5 ÷ ±20V). The A/D value of *Read Status* command is proportional to the voltage applied to this input;
- AE OUTPUT - Enables Servo Amplifier (Refer to *AE OUTPUT ON condition* of *Status Bits and Led* section in this document);
  The polarity of AE OUTPUT may be set using AE POL input (ENCODER connector – pin 11):
  AE POL - open =AE OUTPUT active Low;
  AE POL - tied to GND =AE OUTPUT active High
- ANALOG OUTPUT – 11 bit resolution, 0 ÷ ±10V. Adjustable in the range of 0 ÷ ±12V (P2). The offset may be zeroed using P3.

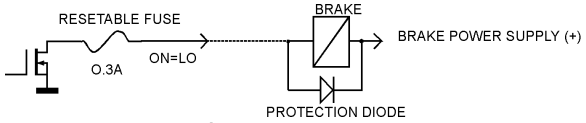
Brake output

If Network Command mode is selected brake will be released after Pic_ae 0 to 1 transition. If the motion control node is in STEP/DIRECTION mode the brake will be released after high to low transition of *SERVO ENABLE* input.

Brake will be engaged (Brake output is OFF) if:

- STP IN is open;
- Servo Amplifier FAULT;
- Overcurrent;
- Position error exceeds the position error limit.

Note: For additional information refer to Status bits and LED, Status byte and Auxiliary status byte and Stop command description sections of this document.



Brake Output

Dip Switches

The dip switches are divided in three groups: terminators, address select and mode select. T-in and T-out are used for connecting terminators to receive and transmit lines. Only the last drive (starting form host) has its T-in and T-out ON. Setting the address select switches (ADR0÷ADR4) to 0 (ON) sets the drive in daisy-chain address mode. If one or more of ADR0÷ADR4 are 1 (OFF) drive address is fixed to the selected value (refer to *Addressing* section of this document). STEP is mode select switch and if it is ON the drive is in STEP/DIRECTION mode. In this mode the drive is controlled by STEP/DIRECTION interface (refer to *STEP/DIRECTION interface* section of this document). Serial command interface in STEP/DIRECTION mode is used only for gain parameters setting and for diagnostics. In Network Command mode the drive is controlled by RS-485 serial command interface.

Serial Command Interface

Serial communication with the LS-180 drives adheres to a full-duplex (4 wire) 8 bit asynchronous protocol with one start bit, followed by 8 data bits (lsb first), followed by a single stop bit.

The communication protocol of the LS-180 also supports a full-duplex multi-drop RS-485 interface that allows multiple LS-180 motion control nodes to be controlled over a single RS-485 port. In this case, the host sends commands over its RS-485 transmit line and receives all status data back over the shared RS-485 receive line.

The command protocol is a strict master/slave protocol in which the host master sends a command packet over the command line to a specific LS-180 slave. The data are stored in the buffer of the LS-180 until the end of the current servo cycle (0.512 msec max.) and then the command is executed. The servo drive then sends back a status packet. Typically, the host does not send another command until a status packet has been received to insure that it does not overwrite any previous command data still in use.

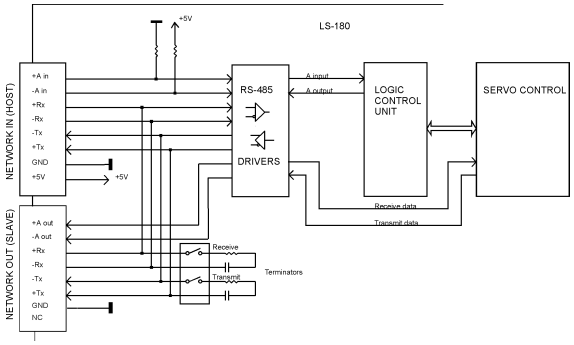
Each command packet consists of:

Header byte (0xAA)
Address byte - individual or group (0x00 - 0xFF)
Command byte
0 - 15 data bytes
Checksum byte

The command byte is divided into upper and lower nibbles: the lower nibble is the command value; the upper nibble is the number of additional data bytes, which will follow the command byte. The checksum byte is 8 bits sum of the address byte, the command byte and the data bytes. The number of data bytes depends on the particular command chosen. After a command is issued, the corresponding controller will send back a status packet consisting of:

Status byte
0-19 optional bytes of status data
Checksum byte

The status byte contains basic status information about the LS-180, including a checksum error flag for the command just received. The optional data bytes may include data such as the position, velocity, etc. and are programmable by the host. The checksum byte is the 8-bit sum of the status byte and the additional optional status data bytes. The transmission of all 16-bit and 32-bit data is always with the least significant byte first.
There is only one exception – when using *EEPROM control* command with control byte 0 (*Read EEPROM Memory Data*), 29 bytes of EEPROM content are returned before the normal status packet.



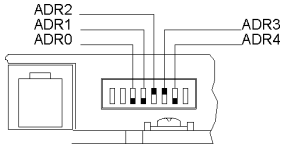
Motion Control Node Serial Interface

Addressing

LS-180 has two addressing modes: fixed address when one or more of ADR0 ÷ 4 are OFF and daisy-chain when ADR0 ÷ 4 are ON.
In daisy-chain mode the host dynamically sets the address of each LS-180 with the aid of the daisy-chained *A in* and *A out* lines. This allows additional LS-180 controllers to be added to an RS-485 network with no hardware changes. *A in* of the first LS-180 is pulled low, its communication is enabled and the default address is 0x00. When the *Set Address* command is issued to give this LS-180 new unique address, it will lower its *A out* pin. Connecting *A out* pin to the *A in* pin of the next servo drive in the network will enable its communication at default address of 0x00. Repeating this process allows a variable number of controllers present to be given unique addresses.
If fixed addressing mode is used the Drive address is fixed to:

DRIVE ADDRESS=ADR0x1+ADR1x2+ADR2x4+ADR3x8+ADR4x16
Where: ADR0 (1,2,3,4)=0 if switch is ON
ADR0 (1,2,3,4)=1 if switch is OFF

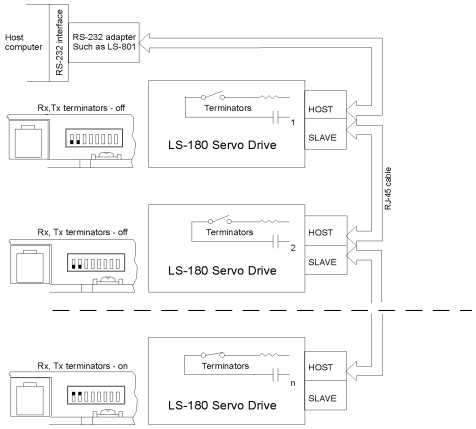
Example:



DRIVE ADDRESS=1x1+1x2+0x4+0x8+1x16=19

Note: Only one addressing mode must be used into a single network.

In addition to the individual address, each controller has a *secondary group address*. Several LS-180 controllers may share a common group address. This address is useful for sending commands, which must be performed simultaneously by a number of drivers (e.g. *Start motion, Set Baud Rate*, etc.). When a LS-180 receives a command sent to its group address, it will execute the command but not send back a status packet. This prevents data collisions on the shared response line. In programming group addresses, however, the host can specify that one member of the group is the "group leader". The group leader will send back a status packet just like it would for a command sent to its individual address. The group address is programmed at the same time as the unique individual address using the *Set Address* command.



Multiple Controller Configuration

Changing Communication Rates

The default baud rate after power-up is 19.2 Kbps. Baud rates up to 115.2 Kbps may be used at maximum servo rate. After communication has been established with all servo drives on a single network, the baud rate may be changed to a higher value with the *Set Baud Rate* command.

Servo Control

LS-180 uses a "proportional-integral-derivative", or PID filter. The position, velocity and acceleration are programmed as 32-bit quantities in units of encoder counts for servo ticks. For example, a velocity of one revolution per second of a motor with a 500 line encoder (2000 counts/rev) at a tick time of 0.512 msec would correspond to a velocity of 1.0240 counts/tick. Velocities and accelerations use the lower 16 bits as a fractional component so that the actual programmed velocity would be 1.024×2^{16} or 67,109. An acceleration of 4 rev/sec/sec (which would bring us up to the desired speed in ¼ sec) would be 0.0021 counts/tick/tick; with the lower 16 bits the fractional component, this would be programmed as 0.0021×2^{16} or 137. Position is programmed as a straight 32-bit quantity with no fractional component. Note that if the servo rate divisor is modified, the time dependent velocity and acceleration parameters will also have to be modified.

DAC Mode Operation

If the position servo is disabled, the motion control node is operated in a raw DAC output mode and no trapezoidal or velocity profiling is performed. In this mode, a user specified PWM value is converted directly to an output voltage. Setting the PWM (refer to *Load Trajectory* command in *Command Description* section in this document) value in the range of 0 ÷ ±255 corresponding to output voltage in the range of 0 ÷ ±10V. Command position is continually updated to match the actual position of the motor and there will be no abrupt jump in the motor's position when position or velocity modes are entered. Also while the position servo is disabled, the command velocity is continually updated to match the actual velocity of

motor. Thus, when velocity mode is entered, there will be no discontinuity in the motor's velocity. (Trapezoidal profile motions, however, will still force the motor to begin at zero velocity).

Safety Features and Diagnostics

To protect both the user device and the controller, LS-180 is equipped with various safety features.

STP IN – Stop Input

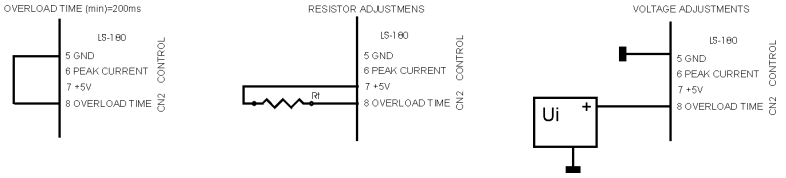
For normal operation STP IN signal must be LOW. If it is HIGH=open the AE OUTPUT will be disabled (set to inactive value) and status byte bit 3 (Power_on) will be set to zero.

Current Limit Control*

An analog input value can be read using *Read Status* command (refer to *Command description* section on this document).
If analog input is connected to the amplifier current monitor output, the software current limit control may be used.
Current limit parameter (CL parameter of *Set Gain* command) is compared each servo tick with A/D value (proportional to the motor current). The actual ANALOG OUTPUT value is:

DAC=DACcalc – DACadj

Where: *DAC* is the ANALOG OUTPUT value; *DACcalc* is the motion command calculated value; *DACadj* (0<*DACadj*≤*DACcalc*) is an internal parameter. If CL<A/D *DACadj* is incremented by 1 each servo tick. If CL>A/D *DAC*adj is decremented by 1 to 0. Bit 2 (Current_Limit) of status byte will be set until CL<A/D. CL is in the range of 0 ÷ 255 and only odd values must be used. "Logosol MCL Interpreter for LS-180" CLI command (refer to *MCL Terminal Interpreter Command Set* section in this document) is provided for current limit control. To disable the function set CL=0 of *Set Gain* command.
If the *DAC* value remains limited by *Current limit for Time=Overload Time* the drive will be disabled.



Overload Time Setting

Overload Time resistors adjustments

Overload time / s	0.65	1.0	2.0	3.0	4.0	6.0	8.0	12.0
Rt	OPEN	300K	75K	36K	24K	11K	7K5	0

Overload Time voltage adjustments

Overload time / s	0.2	0.65	1.0	2.0	3.0	4.0	6.0	8.0	12.0
Ut / V	0.0	0.3	0.5	1.0	1.5	2.0	3.0	3.5	5.0

Encoder Control

Using differential encoder provides protection against motor windup caused by missing or broken encoder wire. A dedicated logic controls the presence of all encoder signals. Detecting a missing signal is considered as Encoder Error (refer to *Status Bits and Led* section in this document).
If nondifferential encoder is connected to the motion control node *MODE* (pin 12 of Encoder connector) must be connected to GND (pin 1).

* *Many amplifiers have an analog voltage output proportional to the motor current.*

Status Bits and LED

Bit 3 (Power_on), bit 5 (Reverse Limit) and bit 6 (Forward Limit) of Status Byte and bit 0 (Index) of Auxiliary Status byte are used for reading input signals and node diagnostics as shown on tables below.

AE OUTPUT OFF condition (Pic_ae=0)

Diagnostic bits

Status byte					
Bit 6 (Forward Limit)	Bit 5 (Reverse Limit)	Bit 3 (Power_on)	Condition		LED intensity
1	1	0	A	AMPLIFIER FAULT	Low
1	0	1	B	STP IN ACTIVATED	Low
1	0	0	D	A+B	Low
1	1	1	H	OK CONDITION	Low

Diagnostic codes

Status byte				
Pos_error =1	Pos_error =0	Fault code description		Index
71h	61h	A	AMPLIFIER FAULT	1
59h	49h	B	STP IN ACTIVATED	1
51h	41h	D	A+B	1
79h	69h	H	NO Fault	1

AE OUTPUT ON condition (Pic_ae=1)

Status byte					
Bit 6 (Forward Limit)	Bit 5 (Reverse Limit)	Bit 3 (Power_on)	Drive inputs CN2 (Control)		LED intensity
0	1	1	A	FORWARD LIMIT=HIGH REVERSE LIMIT=LOW	High
1	0	1	B	FORWARD LIMIT=LOW REVERSE LIMIT=HIGH	High
0	0	1	C	FORWARD LIMIT=HIGH REVERSE LIMIT=HIGH	High
1	1	1	D	FORWARD LIMIT=LOW REVERSE LIMIT=LOW	High

Motion Control Node Fault condition (Pic_ae=1)

Diagnostic bits

Status byte					
Bit 6 (Forward Limit)	Bit 5 (Reverse Limit)	Bit 3 (Power_on)	Condition		LED intensity
1	0	0	A	STP IN (LATCHED) OR ENCODER ERR	Low
0	1	0	B	AMPLIFIER FAULT	Low
0	0	0	C	OVERCURRENT	Low

Diagnostic codes

Status byte			
	Fault code description		Index
51h	A	STP IN	1
		ENCODER ERROR	0
31h	B	AMPLIFIER FAULT	1
15h		SOFTWARE CURRENT LIMIT	1

Brake follows the LED intensity:

- low intensity <-> brake is engaged;
- high intensity <-> brake is released.

To restore the normal operation, if bit 3 (Power_on) is set to 0, Pic_ae must be cycled to 0 and 1. If there are no more fault conditions Power_on will be set to 1.

Power-up and Reset Conditions

On Power-up or reset, the following state is established:

Motor position is reset to zero
Velocity and acceleration values are set to zero
All gain parameters and limit values are set to zero
The servo rate divisor is set to 1 (0.512 msec servo rate)
The PWM value is set to zero
The controller is placed in PWM mode
The default status data is the status byte only
The individual address is set to 0x00 and the group address to 0xFF (group leader not set)
Communications are disables pending a low value of "A in"
The baud rate is set to 19.2 KBPS
In the status byte, the move_done and pos_error flags will be set and the current_limit and home_in_progress flags will be cleared
In the auxiliary status byte, the pos_wrap, servo_on, accel_done, slew_done and servo_overrun flags will be cleared.

COMMAND SPECIFICATION

List of Commands

Command	CMD Code	# Data bytes	Description	While Moving?
Reset position	0x0	0	Sets position counter to zero	No
Set address	0x1	2	Sets the individual and group addresses	Yes
Define status	0x2	1	Defines which data should be sent in every status packet	Yes
Read status	0x3	1	Causes particular status data to be returned just once	Yes
Load trajectory	0x4	1-14	Loads motion trajectory parameters	Maybe*
Start motion	0x5	0	Executes the previously loaded trajectory	Maybe**
Set gain	0x6	14	Sets the PID gains and operating limits	Yes
Stop motor	0x7	1 or 5	Stops the motor in one of three manners	Yes
I/O control	0x8	1-5	Sets the direction and values of the LIMIT pins and auxiliary output ports values	Yes
Set home mode	0x9	1	Sets conditions for capturing the home position	Yes
Set baud rate	0xA	1	Sets the baud rate (group command only)	Yes
Clear bits	0xB	0	Clears the sticky status bits	Yes
Save as home	0xC	0	Saves the current position in the home position register	Yes
EEPROM control	0xD	1	Saves, restores and returns EEPROM memory content	No***
Nop	0xE	0	Simply causes the defined status data to be returned	Yes
Hard reset	0xF	0	Resets the controller to its power-up state.	Yes

**Only allowed while moving if the "start motion now" bit of the trajectory control word is not set or if the "profile mode" bit is set for velocity mode.*

***Only allowed while moving if the previously loaded trajectory has the "profile mode" bit set for velocity mode.*

****This command can be used with amplifier disabled (Pic_ae=0) only.*

LS-180 Command Description

Reset Position

Command value: *0x0*
Number of data bytes: *0*
Command byte: ***0x00***

Description:
Resets the 32-bit encoder counter to 0. Also resets the internal command position to 0 to prevent the motor from jumping abruptly if the position servo is enabled. Do not issue this command while executing a trapezoidal profile motion.

Set Address

Command value: *0x1*
Number of data bytes: *2*
Command byte: ***0x21***
Data bytes:

1. *Individual address: 0x01-0x7F (initial address 0x00)*
2. *Group Address: 0x80-0xFF (initial value 0xFF)*

Description:
Sets the individual address and group address. Group addresses are always interpreted as being between 0x80 and 0xFF. If a Drive is to be a group leader, clear bit 7 of the desired group address in the second data byte. The LS-180 will automatically set bit 7 internally after flagging the Drive as a group leader (If bit 7 of the second data byte is set, the module will be a group member by default). The first time this command is issued after power-up or reset, it will also enable communications for the next Drive in the network chain by lowering it's "A out" signal.

Define Status

Command value: *0x2*
Number of data bytes: *1*
Command byte: ***0x12***
Data bytes:
1. Status items: (default: 0x00)

Bit 0: *send position (4 bytes)*
1: *send A/D value (1 byte)*
2: *send actual velocity (2 bytes - no fractional component)*
3: *send auxiliary status byte (1 byte)*
4: *send home position (4 bytes)*
5: *send device ID and version number (2 bytes)*
(motor controller device ID = 0, version number =60 - 69)
6: *send current position error (2 bytes)*
7: *send auxiliary input values (3 bytes)*

Description:
Defines what additional data will be sent in the status packet along with the status byte. Setting bits in the command's data byte will cause the corresponding additional data bytes to be sent after the status byte. The status data will always be sent in the order listed. For example if bits 0 and 3 are set, the status packet will consist of the status byte followed by

four bytes of position data, followed by the auxiliary status byte, followed by the checksum. The status packet returned in response to this command will include the additional data bytes specified. On power-up or reset, the default status packet will include only the status byte and the checksum byte.
Note: The actual velocity is a positive number when moving in reverse direction and a negative number when moving in forward direction.

Read Status

Command value: *0x3*
Number of data bytes: *1*
Command byte: ***0x13***
Data bytes:
1.Status items:

Bit 0: *send position (4 bytes)*
1: *send A/D value (1 byte)*
2: *send actual velocity (2 bytes - no fractional component)*
3: *send auxiliary status byte (1 byte)*
4: *send home position (4 bytes)*
5: *send device ID, version number (2 bytes)*
(Motor controller device ID = 0, version number = 60 - 69)
6: *send current position error (2 bytes)*
7: *send auxiliary input values (3 bytes)*

Description:
This is a non-permanent version of the *Define Status* command. The status packet returned in response to this command will incorporate the data bytes specified, but subsequent status packets will include only the data bytes previously specified with the *Define Status* command.
Note: The actual velocity is a positive number when moving in reverse direction and a negative number when moving in forward direction.

Load Trajectory

Command value: *0x4*
Number of data bytes: *n = 1-14*
Command byte: ***0xn4***
Data bytes:
1.Control byte:

Bit 0: *load position data (n = n + 4 bytes)*
1: *load velocity data (n = n + 4 bytes)*
2: *load acceleration data (n = n + 4 bytes)*
3: *load PWM value (n = n + 1 bytes)*
4: *servo mode - 0 = PWM mode, 1 = position servo*
5: *profile mode - 0 = trapezoidal profile, 1 = velocity profile*
6: *velocity/PWM direction - 0 = FWD, 1 = REV*
7: *start motion now*

Description:
All motion parameters are set with this command. Setting one of the first four bits in the control byte will require additional data bytes to be sent (as indicated) in the order listed. The

position data (range[*]* +/- 0x7FFFFFFF) is only used as the goal position in trapezoidal profile mode. The velocity data (range 0x00000000 to 0x7FFFFFFF) is used as the goal velocity in velocity profile mode or as the maximum velocity in trapezoidal profile mode. Velocity is given in encoder counts per servo tick, multiplied by 65536. The acceleration data (range 0x00000000 to 0x7FFFFFFF) is used in both trapezoidal and velocity profile mode. Acceleration is given in encoder counts per servo tick per servo tick, multiplied by 65536. The PWM value (range 0x00 - 0xFF), used only when the position servo is not operating, directly sets an ANALOG OUTPUT voltage (range 0 - +/-10V). The PWM value is reset to 0 internally on any condition, which automatically disables the position servo. Bit 4 of the control byte specifies whether the position servo should be used or the PWM mode should be entered. Bit 5 specifies whether a trapezoidal profile motion should be initiated or the velocity profiler is used. Trapezoidal profile motions should only be initialized when the motor velocity is 0. (Bit 0 of the status byte indicates when a trapezoidal profile motion is complete, or in velocity mode, when the command velocity has been reached.) Bit 6 indicates the velocity or PWM direction and is ignored in trapezoidal profile mode. If bit 7 is set, the command will be executed immediately. If bit 7 is clear, the command data will be buffered and it will be executed when the *Start Motion* command is issued. For example to only load new position data and acceleration data but not to start the motion yet, the command byte would be 0x94, the control byte would be 0x15, followed by 4 bytes of position data (least significant byte first), followed by 4 bytes of acceleration data.

If in the middle of a trapezoidal position move, a new *Load Trajectory* command is issued with new position data downloaded, new position data will be used as a relative offset to modify the goal position. For example, if in the middle of a move to position 50,000, a new *Load Trajectory* command with new position data of 10,000 is loaded, the motor will stop at final position of 60,000. The relative offset can be either positive or negative. The new *Load Trajectory* command must be issued while the motor is running at a constant velocity – issuing the command while accelerating or decelerating will cause a position error to occur. If more than one *Load Trajectory* is issued before the end of move, the goal position will be modified by the sum of relative offsets.

Start Motion

Command value:	*0x5*
Number of data bytes:	*0*
Command byte:	***0x05***

Description:

Causes the trajectory information loaded with the most recent Load Trajectory command to execute. This is useful for loading several LS-180 chips with trajectory information and then starting them simultaneously with a group command.

[*] *While the position may range from -0x7FFFFFFF to +0x7FFFFFFF, the goal position should not differ from the current position by more then 0x7FFFFFFF.*

Set Gain

Command value: *0x6*
Number of data bytes: *14*
Command byte: ***0xE6***
Data bytes:

1,2. Position gain Kp (0 - 0x7FFF)
3,4. Velocity gain Kd (0 - 0x7FFF)
5,6 Integral gain Ki (0 - 0x7FFF)
7,8. Integration limit IL (0 - 0x7FFF)
9. Output limit OL (0 - 0xFF) (typically recommended 0xFA)
10. Current limit CL (0 - 0xFF) (only odd values)
11,12 Position error limit EL (0 - 0x3FFF)
13. Servo rate divisor SR (1 - 0xFF)
14. Amplifier deadband compensation (0 - 0xFF) (typical value is between 0x03 and 0x05)

Description:

Sets all parameters and limits governing the behavior of the position servo. KP, KD, KI and IL are PID filter parameters. OL limits the maximal PWM output value to 0<PWM≤OL in position servo modes. In PWM mode OL is ignored. CL is used for motor current limitation (refer to *Current Limit control* in *Safety Features*). Setting CL=0 effectively disables current limiting. The position error limit (EL) will cause the position servo to be disabled should the position error grow beyond the limit. The servo rate divisor sets the servo tick time to be a multiple of 0.512 msec (1.953 KHz). For example SR=3 defines a servo rate of 651 Hz. The servo tick rate is also used as the profiling time base, although command processing and current limiting are always performed at the maximum tick rate. Some times it is necessary to compensate for the deadband region around zero PWM output exhibited by some amplifier/motor combinations. The deadband compensation value will be added to the magnitude of the PWM output to force the amplifier into its active region.

Stop Motor

Command value: *0x7*
Number of data bytes: *1 or 5*
Command byte: ***0x17 or 0x57***
Data bytes:
1. Stop control byte

Bit 0: Pic_ae (Power Driver enable)
1: Turn motor off
2: Stop abruptly
3: Stop smoothly
4: Stop here
5-7: Clear all to 0

2-5. Stopping position (only required if bit 4 above is set)

Description:

Stops the motor in the specified manner. If bit 0 of the Stop Control Byte is set, Power Driver will be enabled. If bit 0 is cleared Power Driver will be disabled, regardless of the state of the other bits. Pic_ae also controls the meaning of bit 3 (Power_on), bit 5 (Reverse Limit), and bit 6 (Forward Limit) of status byte (refer to “Status Bits” section of “Safety Features” in this document). If bit 1 is set, the position servo will be disabled, the PWM output value will be set to 0, and bits 2, 3 and 4 will be ignored. If bit 2 is set, the current command velocity and the

goal velocity will be set to 0, the position servo will be enabled, and velocity mode will be entered. If the velocity servo was previously disabled, the motor will simply start servoing to its current position. If the motor was previously moving in one of the profiling modes, it will stop moving abruptly and servo to its current position. This stopping mode should only be used as an emergency stop where the motor position needs to be maintained. Setting bit 3 enters a more graceful stop mode - this sets the goal velocity to 0 and enters velocity mode, causing the motor to decelerate to a stop at the current acceleration rate. If bit 4 is set, the motor will move to the specified stopping position abruptly with no profiling. This mode can be used to cause the motor to track a continuous string of command positions. Note that if the stopping position is too far from the current position, a position error will be generated. Only one of the bits 1, 2, 3 or 4 should be set at the same time. The *Stop Motor* command must be issued initially to set Pic_ae before other motion commands are issued.

I/O Control

Command value: *0x8*
Number of data bytes: *n=1 - 5*
Command byte: ***0xn8***
Data bytes:
1. I/O control byte

Bit	*0:*	*Output value of Reverse Limit (not used)*
	1:	*Output value of Forward Limit (not used)*
	2:	*Direction of Reverse Limit (must be set to 1 = input)*
	3:	*Direction of Forward Limit (must be set to 1 = input)*
	4:	*Write next data byte to AUX Port 0 (n = n + 1 byte)*
	5:	*Write next data byte to AUX Port 1 (n = n + 1 byte)*
	6:	*Write next data byte to AUX Port 2 (n = n + 1 byte)*
	7:	*Write next data byte to AUX Port 3 (n = n + 1 byte)*

Description:
Sets the four values of auxiliary ports of LS-180. Setting one of the bits 4 to 7 in the control byte will require additional data byte to be sent (as indicated) in the order listed.
When the controller is in STEP/Direction mode, this command with bit 0 set to 1 switches the controller in RS-485 network command mode. Switching back to STEP/Direction mode is performed with *Hard Reset* command. This function works with version 64 and above.
After power-up Reverse Limit and Forward Limit are inputs.
Note: Write ports command is implemented in software level only and is intended for further use.

Set Homing Mode

Command value: 0x9
Number of data bytes: 1
Command byte: **0x19**
Data bytes:
1. Homing control byte

Bit	0:	Capture home position on change of Reverse Limit
	1:	Capture home position on change of Forward Limit
	2	Turn motor off on home
	3:	Capture home on change of Index
	4:	Stop abruptly on home
	5:	Stop smoothly on home
	6:	Capture home position when an excess position error occurs
	7:	Capture home position when current limiting occurs

Description:
Causes the Drive to monitor the specified conditions and capture the home position when any of the flagged conditions occur. The home_in_progress bit in the status byte is set when this command is issued and it is lowered when the home position has been found. Setting one (and only one) of bits 2, 4 or 5 will cause the motor to stop automatically in the specified manner once the home condition has been triggered. This feature can also be used as a safety shutoff.
Note: For homing with Index signal, use low velocities, which ensure the time of the Index pulse is at least one servo tick (0.512 msec). The maximum theoretical homing velocity is 65536 (one encoder count per servo tick). Depending of motor vibrations, the homing velocity should be less than 65536. A recommended homing velocity is 16384 (0.25 encoder counts per servo tick).

Set Baud Rate

Command value:	*0xA*	**sample values**:	
Number of data bytes:	*1*	9600	BRD = 0x81
Command byte:	***0x1A***	19200	BRD = 0x3F
Data bytes:		57600	BRD = 0x14
1. Baud rate divisor,	*BRD*	115200	BRD = 0x0A

Description:
Sets the communication baud rate. All LS-180 drives on the network must have their baud rates changed at the same time; therefore this command should only be issued to a group including all of the controllers on the network. A status packet returned from this command would be at the new baud rate, so typically (unless the host's baud rate can be accurately synchronized) there should be no group leader when this command is issued.

Clear Sticky Bits

Command value:	*0xB*
Number of data bytes:	*0*
Command byte:	***0x0B***

Description:
The overcurrent and position error bits in the status byte and the position wrap and servo timer overrun bits in the auxiliary status byte will stay set unless cleared explicitly with this command.

Save Current Position as Home

Command value:	*0xC*
Number of data bytes:	*0*
Command byte:	***0x0C***

Description:
Causes the current position to be saved as the home position. This command is typically issued to a group of controllers to cause their current positions to be stored synchronously. The stored positions can then be read individually by reading the home position

EEPROM Control

Command value: *0xD*
Number of data bytes: *n = 1 - 5*
Command byte: ***0xnD***

Actually there are 8 different EEPROM Control commands. All of these commands have to be used when servo is disabled (Pic_ae=0):

Read EEPROM memory data

1D 00

This command causes the module to return 29 bytes of EEPROM data and a normal status packet after them. The meanings of these 29 bytes follow:
KP – bytes 0 and 1; KD – bytes 2 and 3; KI – bytes 4 and 5; IL – bytes 6 and 7; Output limit – byte 8; Current limit – byte 9; Maximal error limit – bytes 10 and 11; Servo rate divisor – byte 12; Amplifier deadband compensation – byte 13; Velocity – bytes 14, 15, 16 and 17; Acceleration – bytes 18, 19, 20 and 21; Byte 22 is not used; Servo init state – byte 23 (if this byte is 0x01 LS-180 will turn on servo loop on power-up); Auxiliary output ports 0 to 3 – bytes 24, 25, 26 and 27; Mode – byte 28 (0x01 – step and direction mode, 0x00 – network mode).

Save gain parameters

1D 01

Saves previously loaded servo gain parameters (see *Servo Gain* Command)

Restore gain parameters

1D 02

Restores servo gain parameters from EEPROM memory. This command has the same effect as *Set Gain* command with corresponding values.

Save velocity and acceleration

1D 04

Saves previously loaded goal velocity and acceleration used in the last motion.

Restore velocity and acceleration

1D 08

Restores goal velocity and acceleration. This command has the same effect as *Load trajectory* command with corresponding velocity and acceleration.

Save auxiliary output values

5D 10 N0 N1 N2 N3

Saves the values N0, N1, N2 and N3 (0-0xFF) as auxiliary output values in EEPROM.

Restore auxiliary output values

1D 20

Restores auxiliary output values from EEPROM to the outputs. This command has the same effect as *I/O Control* command with corresponding values for auxiliary outputs.

Save “servo off power-up state”

1D 40

Causes LS-183 to save “no servo” state as a starting state after Power-up.

Save “servo on power-up state”.

1D C0

Causes LS-183 to save “servo” state as a starting state after Power-up.

Note: These commands normally need more time to be executed than single execution loop time and causes servo_overrun bit in auxiliary status byte to be set. This bit can be cleared with *Clear Sticky Bits* Command.

No Operation

Command value:	*0xE*
Number of data bytes:	0
Command byte:	*0x0E*

Description:
Does nothing except cause a status packet with the currently defined status data to be returned.

Hard Reset

Command value:	*0xF*
Number of data bytes:	*0*
Command byte:	***0x0F***

Description:
Resets the control module to its power-up state. No status will be returned. Typically, this command is issued to all the modules on the network, although if the baud rate is set at the default, it is possible to reset and re-initialize the addresses of a contiguous sub-chain of modules.
Note: *Hard Reset* command sent at address 0xFF will be executed by all LS-180 Drives, regardless of their own group address.

STATUS BYTE AND AUXILIARY STATUS BYTE DEFINITIONS

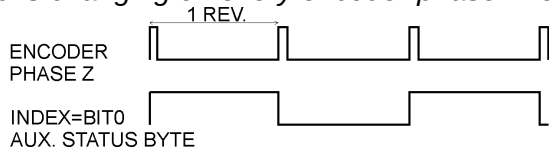
Status Byte

Bit	Name	Definition
0	Move_done	Clear when in the middle of a trapezoidal profile move or in velocity mode, when accelerating from one velocity to the next. This bit is set otherwise, including while the position servo is disabled
1	Cksum_error	Set if there was a checksum error in the command packet received
2	Current_limit	Set if current limiting occurred (refer to "Motor Current Monitoring" section in this document). Must be cleared by user with *Clear Sticky Bits* command.
3	Power_on/diag. bit	Refer to "Status Bits and LED" section in this document
4	Pos_error	Set if the position error exceeds the position error limit. It is also set whenever the position servo is disabled. Must be cleared by user with *Clear Sticky Bits* command.
5	Reverse Limit/ diag. bit	Reverse Limit or diagnostic bit (refer to "Status Bits and LED" section in this document).
6	Forward Limit/ diag. bit	Forward Limit or diagnostic bit (refer to "Status Bits and LED" section in this document).
7	Home_in_progress	Set while searching for a home position. Reset to zero once the home position has been captured.

Auxiliary Status Byte

Bit	Name	Definition
0	Index/diag. bit	Compliment of the value of the index* input or diagnostic bit (refer to "Status Bits and LED" section in this document).
1	Pos_wrap	Set if the 32-bit position counter wraps around. Must be cleared with the Clear Sticky Bits command
2	Servo_on	Set if the position servo is enabled, clear otherwise
3	Accel_done	Set when the initial acceleration phase of a trapezoidal profile move is complete. Cleared when the next move is started.
4	Slew_done	Set when the slew portion of a trapezoidal profile move is complete. Cleared when the next move is started.
5	Servo_overrun	At the highest baud rate and servo rate, certain combinations of calculations may cause the servo, profiling, and command processing to take longer than 0.512 msec, in which case, this bit will be set. This is typically not serious, only periodically introducing a small fraction of a millisecond delay to the servo tick time. Cleared with the Clear Sticky Bits command.
6	Reserved	
7	Reserved	

The logic level of **Index=bit0 is changing on every encoder phase Z low to high transition.*



INITIALIZING PROCEDURE AND PROGRAMMING EXAMPLES FOR LS-180

To ensure a proper operation of all LS-180 drives connected to the network, the following initializing steps should be executed:

1. Reset all modules using *Hard Reset* command.
2. Set the addresses for all connected drives.
3. Set the individual gains (KP, KD, KI, IL, OL, CL, EL, SR and DB). Minimal requirements are: KP <> 0, EL <> 0, and SR <> 0.
4. Use *Load trajectory* command to set the target position, velocity and acceleration with start motion now in trapezoidal mode. Minimal requirements are acceleration <> 0 and target position = 0. This command does not start any motion. It is necessary to initialize internal registers of the module.
5. Close the servo loop by using *Stop Motor* command (Pic_ae=1 and Stop abruptly=1).

Note: Steps 3, 4 and are necessary only if EEPROM data do not fit minimal requirements. Step 5 is necessary if servo init byte in EEPROM is 0.

Understanding the Serial Communication with LS-180

The Serial Communication with LS-180 is strictly master-slave and matches repeatedly two elements:

- Sending a command to the specified drive's address;
- Receiving the answer to the command sent – Status Byte(s).

Note: *During the communication all bytes are sent with LSB first.*

Commands

There are 16 commands controlling LS-180 drives (refer to LS -180 Command Description). Each command as shown in the following two tables includes header, address, command, data bytes and one checksum byte. Checksum does not include header byte.

Structure of *Read Status* command

Byte 1	Byte 2	Byte 3		Byte 4	Byte 5
Header	Address (Individual or Group)	Command Code		Data Byte	CheckSum = Byte 2 + Byte 3 + Data Byte
		High 4 bits No. of data bytes	Low 4 bits command code		
AA	01	1	3	01	15

Examples

Cmd. Bytes	Byte 1	Byte 2	Byte 3	Byte 4 - N	Byte N+1
Command	Header	Address	Cmd. Code	Data Byte(s)	Checksum
Reset position	AA	01	0 0		01
Define status	AA	05	1 2	05	1C
Set address	AA	01	2 1	07 F0	19
Load trajectory	AA	01	5 4	91 00 28 00 00	0E
Set gain	AA	01	E 6	64 00 00 04 00 00 00 00 FF 00 00 08 01 00	57

Status Data

The structure of the returned status information depends on *Define Status* or *Read Status* commands (refer to LS-180 Command Description). By default only the Status byte and Checksum are returned to the host.

Examples

Byte 1	Optional Bytes 0-16	CheckSum
Status Byte	Additional Status Bytes as position, velocity, home position, A/D auxiliary byte, version and position error.	CheckSum = Byte 1+ Optional Bytes
09	no additional status bytes requested	09
09	00 28 00 00 – four additional status bytes	31

Addressing

Each drive in the daisy-chained network has two addresses:

- Individual - for individual control of each drive. Its range is from 01h to 7Fh.
- Group - for simultaneous control of all group members by sending a single command to their group address. It is in the range of 80h to FFh.

Both these addresses have to be set during the initialization process.

The group may have a Group leader responsible to send status data. Its address is:

Group leader address = Group address - 80h.

If there is no group leader - no status data will be send after a group command.

Set Baud Rate command must be sent only as a group command with no group leader, otherwise communication problems may occur.

***Set Address* command format**

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Header	Preset Address	Command code	Individual Address	Group Address	Checksum
AA	00	21	01	FF	21

Setting the Addresses

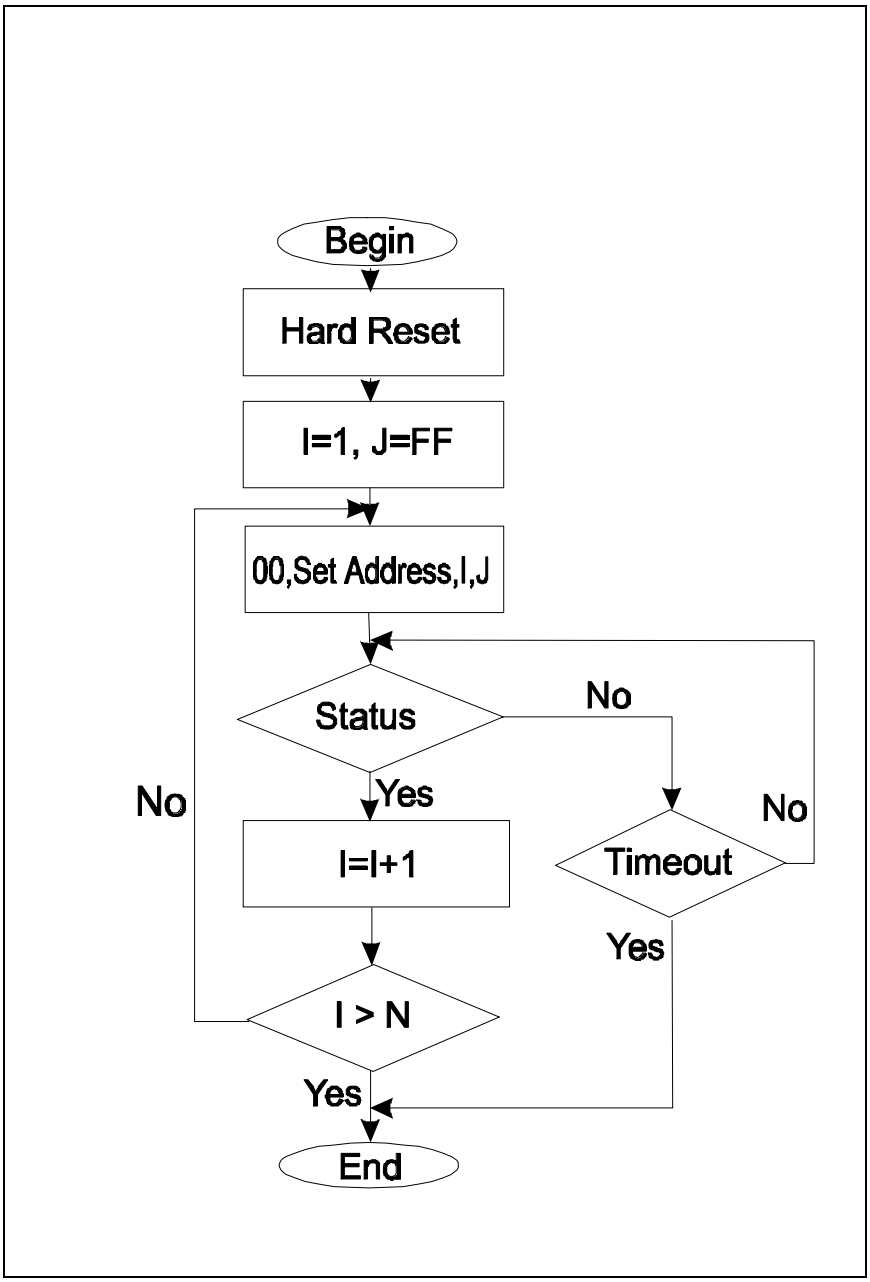
After power-up and *Hard Reset* command all drives have their address set to 00h and only the first drive (starting from the host) has its communication enabled. Consecutive *Set Address* commands are sent to address 00h until all drives are addressed. This procedure can be executed once after *Hard Reset*. The table below shows the steps to address 3-drives network.

Example of sequential addressing for three LS-180 drives

Step	Command	Set address Hexadecimal Code	Drive 1		Drive 2		Drive 3	
			Individual address	Group address	Individual address	Group address	Individual address	Group address
0	Power-up							
1	Hard Reset	AA FF 0F 0E	address=00 communication *enabled*		address=00 communication *disabled*		address=00 communication *disabled*	
2	Set Address Drive1 = 01	AA 00 21 01 FF 21	01	FF	address=00 communication *enabled*		address=00 communication *disabled*	
3	Set Address Drive2 = 02	AA 00 21 02 FF 22	01	FF	02	FF	address=00 communication *enabled*	
4	Set Address Drive3 = 03	AA 00 21 03 FF 23	01	FF	02	FF	03	FF

Note: Before start addressing always *Hard Reset* command must be issued.

The flowchart shows the addressing procedure of N drives network. There is no group leader and the group address is FF.



I - Individual Address; *J* - Group Address = FF;
Status - Status Data sent to the Host; *Timeout* - Greater than one servo circle.

Examples of Managing Two LS-180 Drives

1 – Resets all modules with group command.
2 and # 3 - Set the addresses of drives 1 and 2.
4 and # 5 - Set PID parameters of drives 1 and 2.
6 and # 7 - Starts motion in trapezoidal mode with target position=0, velocity=0, acceleration=1 and PWM=0.
8 and # 9 - Close servo loops of drives 1 and 2. Initialization is complete at this point.
10 and # 10 - Load trajectories (positions, velocities and accelerations) for drives 1 and 2.
12 and # 13 - Load and execute new trajectory for drive 1.
14 and # 15 - Read additional status bytes from drives 1 and 2.
16, # 17 and #18 - Load new trajectories for drives 1 and 2 and execute them with one command sent to the drives' group address.

Examples

#	Hexadecimal code of command	Comments
1	AA FF 0F 0E	*Hard Reset*
2	AA 00 21 01 FF 21	*Set Address* 01h for drive 1. Group address=FFh.
3	AA 00 21 02 FF 22	*Set Address* 02h for drive 2. Group address=FFh.
4	AA 01 E6 64 00 00 04 00 00 00 00 FF 00 00 08 01 00 57	*Set Gains* of drive 1 – defines PID parameters: KP=64h, KI=400h, KI=00h, IL=00h, OL=FFh, CL=00h, EL=800h, SR=01h, DC=00h.
5	AA 02 E6 64 00 00 04 00 00 00 00 FF 00 00 08 01 00 58	*Set Gains* of drive 2 – defines PID parameters: KP=64h, KI=400h, KI=00h, IL=00h, OL=FFh, CL=00h, EL=800h, SR=01h, DC=00h.
6	AA 01 E4 9F 00 00 00 00 00 00 00 00 01 00 00 00 00 85	*Load trajectory* for drive 1 – target position=0, velocity=0, acceleration=1, PWM=0 and start motion now
7	AA 02 E4 9F 00 00 00 00 00 00 00 00 01 00 00 00 00 86	*Load trajectory* for drive 2 – target position=0, velocity=0, acceleration=1, PWM=0 and start motion now
8	AA 01 17 05 1D	*Stop Motor* - closes servo loop of drive 1 with *Power Driver enable* and *Stop Abruptly* in Command byte.
9	AA 02 17 05 1E	*Stop Motor* - closes servo loop of drive 2 with *Power Driver enable* and *Stop Abruptly* in Command byte.
10	AA 01 E4 9F 00 00 00 00 00 80 01 00 64 00 00 00 00 69	*Load Trajectory* of drive 1 with Pos=0000h, Vel=18000h, Acc=6400h, PWM=00h, servo mode=1.
11	AA 02 E4 9F 00 00 00 00 00 80 01 00 64 00 00 00 00 6A	*Load Trajectory* of drive 2 with Pos=0000h, Vel=18000h, Acc=6400h, PWM=00h, servo mode=1.
12	AA 01 54 11 00 28 00 00 8E	*Load Trajectory* of drive 1 with new position=2800h.
13	AA 01 05 06	*Start Motion* - executes previously loaded trajectory.
14	AA 01 13 05 19	*Read Status* from drive 1 (plus position and velocity).
15	AA 02 13 05 1A	*Read Status* from drive 2 (plus position and velocity).
16	AA 01 54 11 20 4E 00 00 D4	*Load Trajectory* of drive 1 with new position=4E20h.
17	AA 02 54 11 E0 B1 FF FF F6	*Load Trajectory* of drive 2 with new position=FFFFB1E0h (-4E20h).
18	AA FF 05 04	*Start Motion* – executes previously loaded trajectories. The command is sent to the drives' group address FFh.

Procedure Initialize

AA FF 0F 0E	*Hard reset*
AA 00 21 01 FF 21	*Set address*
AA 00 21 02 FF 22	Search for more modules...
AA 01 13 20 34	Reads Device ID and Version number
AA 01 13 FF 13	Reads all status data
AA 01 1D 00 1E*	*Reads EEPROM memory data*
AA 01 E6 64 00 00 04 00 00 00 00 FF 00 00 08 01 00 57	Sets Gain parameters
AA 01 E4 9F 00 00 00 00 00 00 00 00 01 00 00 00 00 85	Sets Trajectory parameters
AA 01 17 09 21	Closes the servo loop

*Initialization is complete after this command if gain and trajectory parameters are present in EEPROM memory and servo init byte is 1.

Procedure FindHomePosition

AA 01 E6 C8 00 20 03 46 00 28 00 FF 00 40 1F 01 00 9F	Sets gain parameters: KP=200, KD=800, KI=70, IL=40, Output limit=255, current limit =0, Position error limit=8000, Servo rate divisor=1 amplifier deadband compensation=0
AA 01 17 09 21	Closes the servo loop (Stop smoothly and amplifier enable)
AA 01 94 37 25 06 01 00 58 01 00 00 51	Loads trajectory: Velocity mode, Forward direction, Velocity=1 round per second (67109 programmed velocity for 500 line encoder), Acceleration = 10 round per second^2 (344 programmed acceleration for 500 line encoder)
AA 01 19 12 2C	Sets home mode - capture home position on change of Forward Limit and stop abruptly
AA 01 05 06	Starts motion
wait while home_in_progress bit=1	Home position is found on change of Forward limit
AA 01 19 18 32	Sets home mode - capture home position on change of Index and stop abruptly
AA 01 94 77 25 06 01 00 58 01 00 00 91	Loads trajectory: Velocity mode, Reverse direction
AA 01 05 06	Starts motion
wait while home_in_progress bit=1	Home position is found on change of Index

Calculation of programmed velocity and acceleration for servo rate divisor = 1:
Vel = (encoder counts per revolution) x (number of revolution per second) x 33.554432
Acc = (encoder counts per revolution) x (number of revolution per second^2) x 0.017179869184

For this example:

Vel	= 2000 x 1	x 33.554432	= 67109	= 00010625h
Acc	= 2000 x 10	x 0.017179869184	= 344	= 00000158h