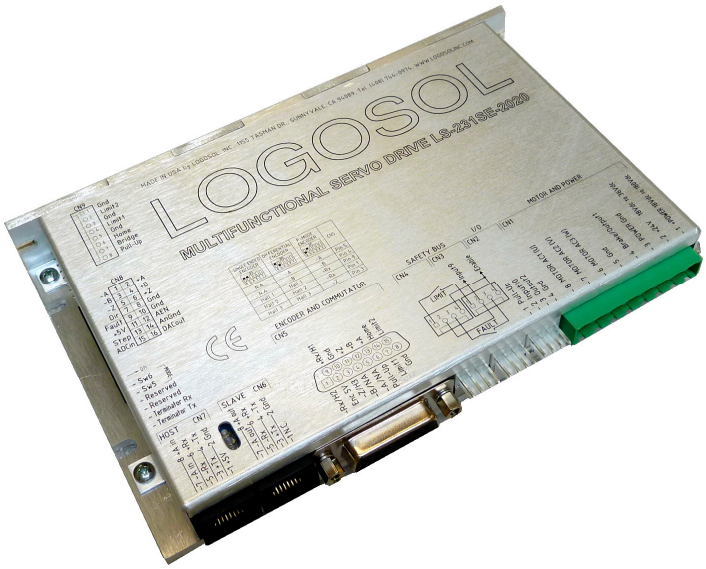
Logosol Multifunctional Servo Drive LS-231SE

Doc # 712231004 / Rev. A, 05/05/2011

Features

- **Controller modes:**
  - **Distributed Servo**
  - **Analog ±10V**
  - **Quadrature Encoder mode**
  - **Step & Direction**
  - **Step Positive/Step Negative**
- **Dual Encoder Loop**
- **Servo loop rate 51.2uS**
- **Sinusoidal motor phases commutation**
- **Motors supported:**
  - **Panasonic A and S series motors**
  - **Brushless 60/120° commutated**
  - **Linear motors**
  - **Voice coil and Brush motors**
- **Up to 20A peak / 12A continuous output current**
- **Up to 200V DC power supply**
- **Path point buffer for coordinated motion control**
- **32-bit position, velocity, acceleration, 16-bit PID filter gain values**
- **Torque loop**
- **Safety Bus**
- **Feedback loss protection**
- **Comprehensive motor output short-circuit protection:**
  - **Output to output**
  - **Output to ground**
- **Adjustable motor current and overload time limits**
- **Over/under voltage shutdown**
- **Overheating protection**
- **Forward and reverse over travel inputs**
- **Communication 19.2Kb/S to 1.25Mb/S**
- **PWM frequency 20 kHz**
- **Encoder rate 10.0MHz**



Description

LS-231SE is all digital multifunctional single-axis servo controller with integrated power amplifier designed for applications requiring sinusoidal control of Rotary or Linear brushless motors up to 1.5 HP. DC (brush) and Voice coil motors are also supported.

In *LDCN* mode the drive supports coordinated motion of several motors. Up to 31 devices can be controlled over a multi-drop full duplex RS-485 network. Standard RJ-45 connectors and commercially available cables are used for daisy chaining of the modules.

In *Analog* modes the drive supports precise speed control. *Analog* modes give flexible use of Analog, Enable and Direction inputs.

Three modes are provided in addition to the standard *Step & Direction* mode. In *Step Positive/Step Negative* mode the direction is controlled via two separate step inputs. In *Quadrature encoder* modes step and direction inputs can be connected to A and B phase of an incremental encoder.

Dual-Loop encoder mode supports two encoder inputs. In this mode the encoders for position and servo control are separated. High resolution encoders can be used for precise position control.

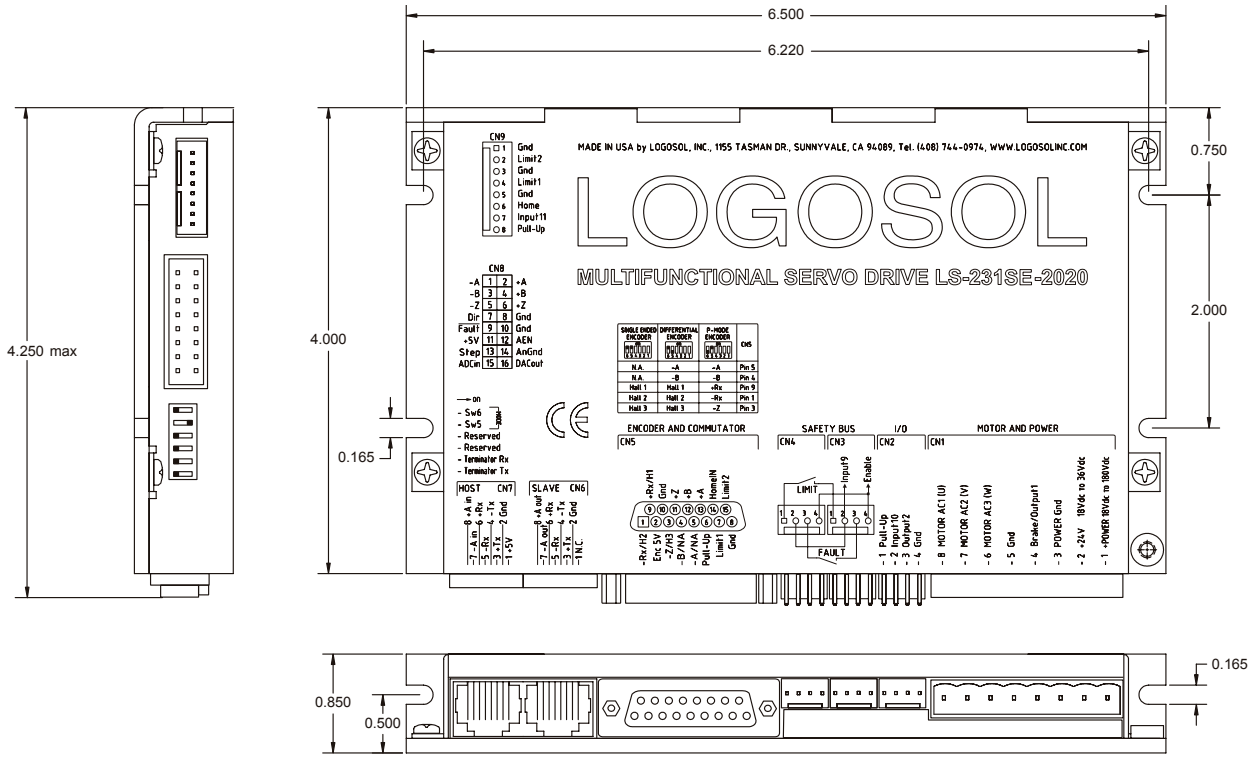
LS-231SE is equipped with various safety features such as short circuit protection for the motor and the drive, limit switch inputs, over/under voltage shutdown and encoder presence control. Safety bus can be connected for multi-drive system protection.

The maximum motor output current and overload time can be set.

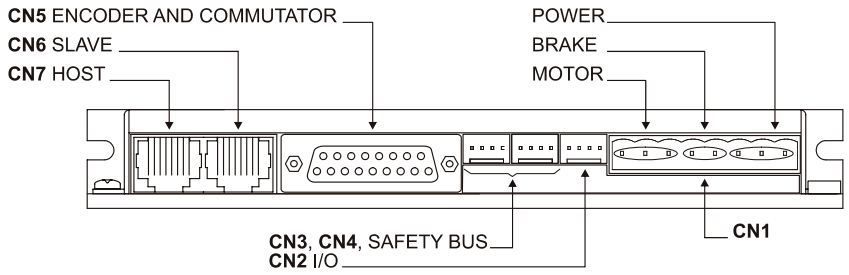
TECHNICAL SPECIFICATIONS rated at 25°C ambient, POWER (+)=60VDC, Load=250μH motor

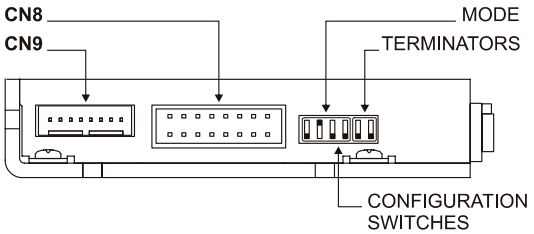
+POWER	18 to 180Vdc, 200V Absolute Maximum
+24V	18Vdc to 36Vdc I/O power supply
+24V Backup Current	0.25A Maximum
MAX MOTOR OUTPUT CURRENT Peak Continuous	 20A 12A
MIN LOAD INDUCTANCE	200μH
PWM SWITCHING FREQUENCY	19.5KHz
SERVO RATE	51.2 μSec
SERIAL BAUD RATE	19.2Kb/Sec to 1.25 Mb/Sec
DIGITAL OUTPUTS Source driver Brake/Output1 (CN1pin4) Pull-Up (CN2pin1, CN5pin6, CN9pin8) Output2 (CN2pin3) Fault (CN8pin9)	 24V (18Vdc to 36Vdc Maximum). Short protection. Output clamp diode. 0.3A 0.1A for all outputs combined 0.2A HImin=4V, LOmax=0.4V Iout=4mA
RELAY OUTPUTS LIMIT, FAULT	 Relay contact
ANALOG OUTPUT DACout	 0 to +5V/1mA
DIGITAL INPUTS +A, -A, +B, -B, +Z, -Z, +Rx/H1, -Rx/H2 Limit1, Limit2, HomeIN, Input9, Enable, Input10, Input11 Dir, AEN, Step (CN8pin7, pin12, pin13)	 HI=3.5V, LO=1.5V, 2K2 pull-up resistors LOmin=-0.5V<LO< 6.5V; 15V<HI<HImax=36V;Imax=8mA HI=3.5V, LO=1.5V, 4K7 pull-down resistors
ANALOG INPUT ADCin	 ±10V, 10K to Gnd
ENCODER	Quadrature with index. Encoder rate 10MHz max (5MHz max Enc Filter=ON) Panasonic encoder mode.
HALL SENSORS	60/120° or Panasonic
LEDs ORANGE, GREEN, RED	 Refer to *LS-231SE Diagnostic and I/O* section for details
PROTECTION Short circuit Over-temperature shut off	 Motor output to motor output, Motor output to GND Activated at 80°C
POWER DESSIPATION (max)	45W
THERMAL REQUIREMENTS Storage temperature range Operating temperature range	 –30 to +85°C 0 to 45°C
MECHANICAL Size Weight	 6.5” x 4.0” x 0.85” 0.8 lb. (0.36 kg)
+5V SOURCE Max output current	 200mA for all output pins combined
MATING CONNECTORS MOTOR AND POWER I/O SAFETY BUS CN9 ENCODER AND COMUTATOR CN8 NETWORK (HOST, SLAVE)	RECOMMENDED CONNECTOR TYPE EM2565-08-H Terminal Block 8 Poles 5.08mm Molex 22-01-3047 housing with 08-50-0114 pins (4 pcs.) Molex 22-01-3047 housing (2 pcs.) with 08-50-0114 pins (8 pcs.) Molex 22-01-3087 housing with 08-50-0114 pins (8 pcs.) AMP 205206-3 D-Shape 15 Pin M Crimp Housing with AMP 5-66507-7(15 pcs.) 3M 34526600 Socket16 pin, polarized 0.1" 8 pin RJ-45

DIMENSIONAL DRAWING



SERVO DRIVE LAYOUT

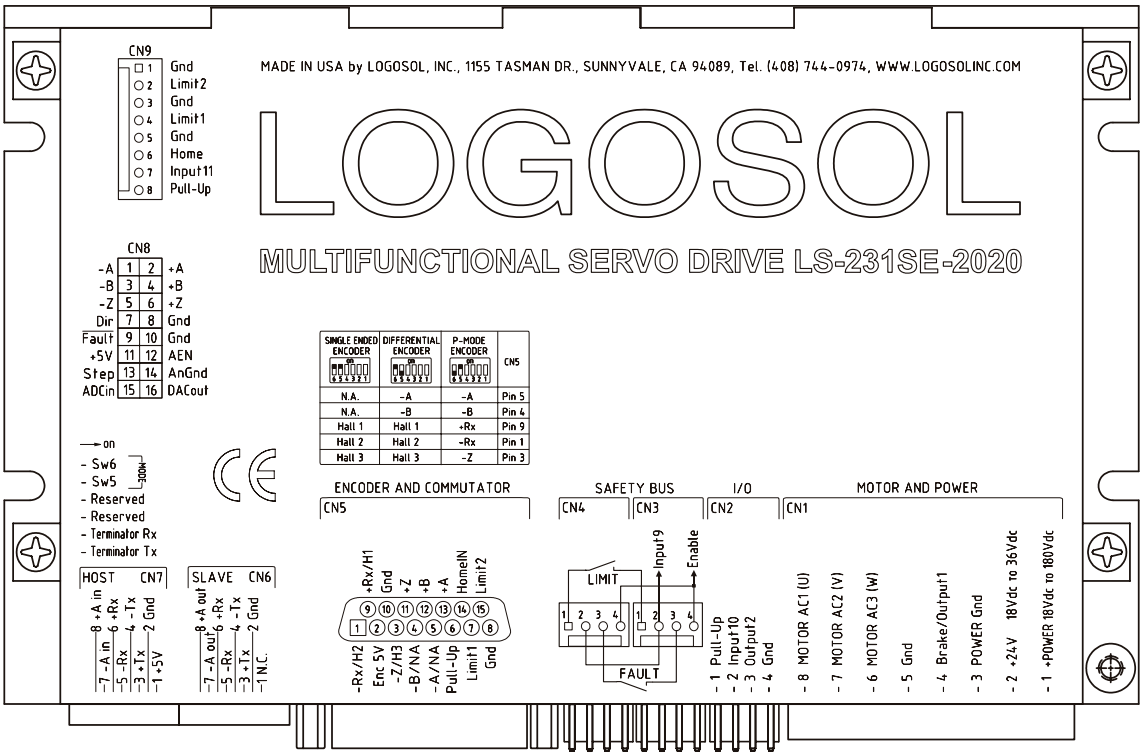




ORDERING GUIDE

PART NUMBER	MODEL	DESCRIPTION
912231010	LS-231SE- 2020	Multifunctional servo drive 20A/180V
230601051	LS-231-CN	Mating connector kit for LS-231SE
230601017	PAN-AS-CN	Mating connector kit for Panasonic A and S series motors
230601027	PAN-ASB-CN	Mating connector kit for Panasonic A and S series motors with brake
922231100	LS-2311	Dual-loop master encoder interface

CONECTORS AND PINOUT



DIP SWITCHES

SW	NAME	DESCRIPTION
1	Terminator Tx	Transmit line terminator
2	Terminator Rx	Receive line terminator
3	Reserved	Reserved must be off
4	Reserved	Must be off.
5	Sw5	Sw6=ON, Sw5=ON - Motor with single ended (non differential) encoder Sw6=ON, Sw5=OFF - Motor with differential encoder Sw6=OFF, Sw5=ON - Panasonic A or S series motor (P-mode) Sw6=OFF, Sw5=OFF - Reserved
6	Sw6	

CONNECTORS

CN1 – POWER AND MOTOR

PIN	SIGNAL	DESCRIPTION
1	+POWER 18Vdc TO 90Vdc	Power supply positive terminal
2	+24V 18Vdc TO 36Vdc	I/O power supply
3	POWER Gnd	Power supply ground
4	Brake/Output1	Brake output
5	Gnd	I/O ground
6	MOTOR AC3 (W)	Output to motor Phase 3 terminal for brushless motors Phase W for Panasonic A and S series motors Negative terminal for DC and Voice Coil motors
7	MOTOR AC2 (V)	Output to motor Phase 2 terminal for brushless motors Phase V for Panasonic A and S series motors Positive terminal for DC and Voice Coil motors
8	MOTOR AC1 (U)	Output to motor Phase 1 terminal for brushless motors Phase U for Panasonic A and S series motors Not connected for DC and Voice Coil motors

CN2 – I/O

PIN	SIGNAL	DESCRIPTION
1	Gnd	Signal ground
2	Output2	Multifunctional output (mode dependent)
3	Input10	Multifunctional input (mode dependent)
4	Pull-Up	Protected power output (+24V)

SAFETY BUS

CN3

PIN	SIGNAL	DESCRIPTION
1	LIMIT	Limit relay contact (CN3)
2	Input9	Limit input
3	FAULT	Fault relay contact (CN3)
4	Enable	Drive Enable/$\overline{\text{Stop}}$ input

CN4

PIN	SIGNAL	DESCRIPTION
1	LIMIT	Limit relay contact (CN4)
2	Input9	Limit input
3	FAULT	Fault relay contact (CN4)
4	Enable	Drive Enable/$\overline{\text{Stop}}$ input

CN5 – ENCODER AND COMMUTATOR

PIN	SIGNAL	DESCRIPTION
1	-Rx/H2	Hall input #2 for brushless motors Hall data for Panasonic A or S series motors Not connected for DC motors
2	Enc 5V	Encoder +5V power supply
3	-Z/H3	Hall input #3 for brushless motors Encoder index (-)Z for Panasonic A or S series motors and DC motors with differential encoder
4	-B/NA	Encoder phase (-)B for motors for motors with differential encoder
5	-A/NA	Encoder phase (-)A for motors for motors with differential encoder
6	Pull-Up	Protected power output (+24V)
7	Limit1	Limit1 input (connected to CN9-4 through 1K resistor)
8	Gnd	Signal ground
9	+Rx/H1	Hall input #1 for brushless motors Hall data for Panasonic A or S series motors Not connected for DC motors
10	Gnd	Encoder ground
11	+Z	Encoder index Z
12	+B	Encoder phase B
13	+A	Encoder phase A
14	HomeIN	Home input (connected to CN9-6 through 1K resistor)
15	Limit2	Limit2 input (connected to CN9-2 through 1K resistor)

CN6 –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 –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

CN8

PIN	SIGNAL	DESCRIPTION
1	-A	Encoder phase (-) A output. Wired to CN5pin5
2	+A	Encoder phase A output. Wired to CN5pin13
3	-B	Encoder phase (-) B output. Wired to CN5pin4
4	+B	Encoder phase B output. Wired to CN5pin12
5	-Z	Encoder phase (-) Z output
6	+Z	Encoder phase Z output. Wired to CN5pin11
7	Dir	*LDCN Single and Dual-loop* modes – NA *Analog input Single/Dual-loop* mode – NA *Analog input mode with direction invert input* – Direction input *Enable Positive/Enable Negative Analog input* mode – Enable + Direction input *Step & Direction mode* – Direction input *Step Positive/Step Negative mode* – Negative Step input *Quadrature encoder mode* – Counter A input
8	Gnd	Signal ground
9	$\overline{\text{Fault}}$	Fault output
10	Gnd	Signal ground
11	+5V	+5V power supply
12	AEN	*LDCN Single loop mode* – NA *LDCN Dual-loop mode* – Secondary encoder error All other modes – Enable input
13	Step	*LDCN Single and Dual-loop* modes – NA All Analog *modes* – NA *Step & Direction mode* – Step input *Step Positive/Step Negative mode* – Positive Step input *Quadrature encoder modes* – Counter B input
14	AnGnd	Analog ground
15	ADCin	Analog input 0 to ±10V
16	DACout	Analog output 0 to +5V

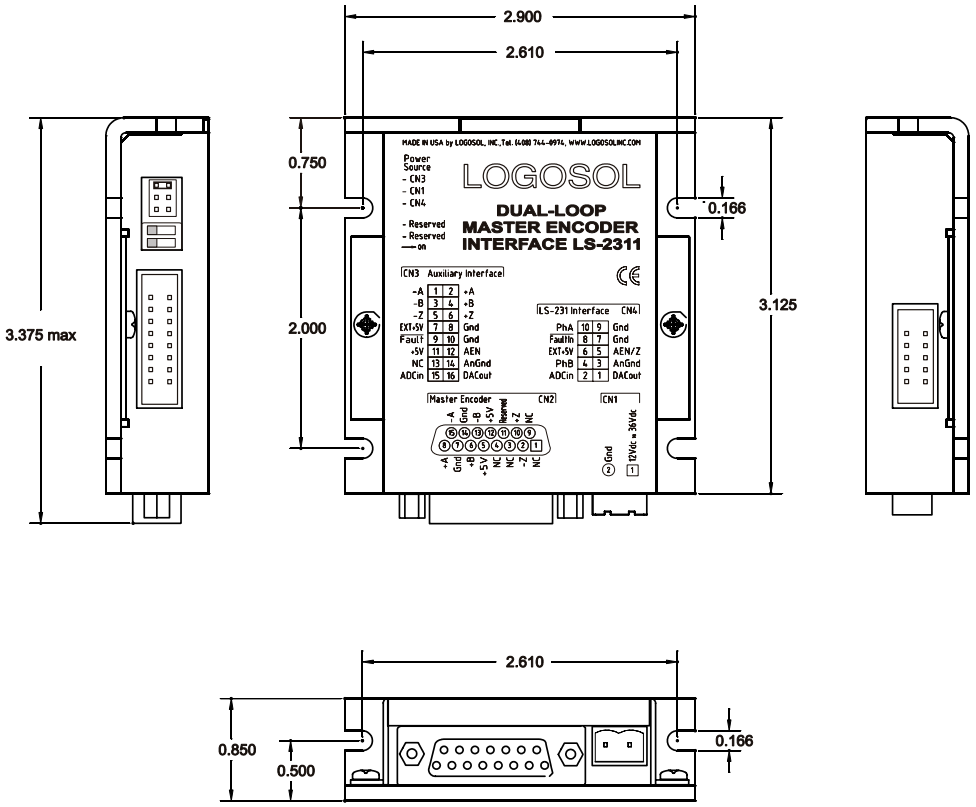
CN9

PIN	SIGNAL	DESCRIPTION
1	Gnd	Signal ground
2	Limit2	Limit2 (connected to CN5-15 through 1K resistor)
3	Gnd	Signal ground
4	Limit1	Limit1 (connected to CN5-7 through 1K resistor)
5	Gnd	Signal ground
6	Home	Home (connected to CN5-14 through 1K resistor)
7	Input11	Multifunctional input (mode dependent)
8	Pull-Up	Protected power output (+24V)

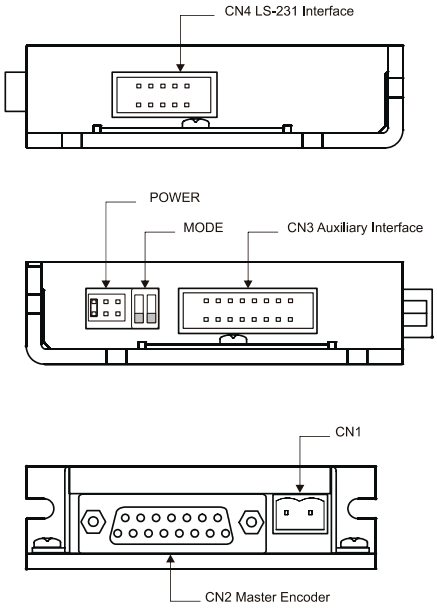
DUAL-LOOP MASTER ENCODER INTEFRACE

LS-231SE1 is master encoder interface for dual-loop applications with LS-231SE Multifunctional Servo Drive.

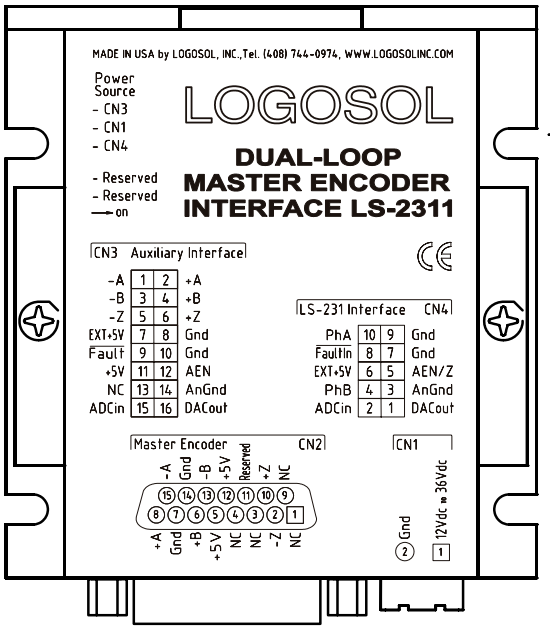
DUAL-LOOP MASTER ENCDOER INTERFACE DIMENSIONAL DRAWING



DUAL-LOOP MASTER ENCODER INTERFACE LAYOUT



DUAL-LOOP MASTER ENCODER INTERFACE LS-2311 CONECTORS AND PINOUT



JUMPERS

JUMPER	NAME	DESCRIPTION
1	CN4	EXT +5V Power supply from CN4pin11
2	CN1	Internal power supply. 12Vdc to 36Vdc power supply from CN1pin1
3	CN3	EXT +5V Power supply from CN3pin7

Note: Only one JUMPER must be shorted.

DIP SWITCHES

SWITCH	NAME	DESCRIPTION
1	Reserved	Reserved must be OFF
2	Reserved	Reserved must be OFF

CN1

PIN	SIGNAL	DESCRIPTION
1	12Vdc to 36Vdc	Power supply input. If Power source jumper CN1 is installed
2	Gnd	Power supply ground

CN2 – Master Encoder

PIN	SIGNAL	DESCRIPTION
1	NC	Not connected
2	-Z	Encoder phase –Z
3	NC	Not connected
4	NC	Not connected
5	+5V	Encoder power supply
6	+B	Encoder phase +B
7	Gnd	Power supply ground
8	+A	Encoder phase +A
9	NC	Not connected
10	+Z	Encoder phase +Z
11	Reserved	Reserved - DO NOT CONNECT
12	+5V	Encoder power supply
13	-B	Encoder phase –B
14	Gnd	Power supply ground
15	-A	Encoder phase –A

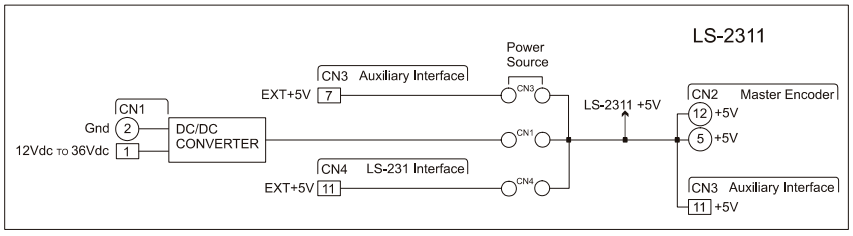
CN3 – Auxiliary Interface

PIN	SIGNAL	DESCRIPTION
1	-A	Encoder phase –A. Wired to CN2pin15
2	+A	Encoder phase +A. Wired to CN2pin8
3	-B	Encoder phase –B. Wired to CN2pin13
4	+B	Encoder phase +B. Wired to CN2pin6
5	-Z	Encoder phase –Z. Wired to CN2pin2
6	+Z	Encoder phase +Z. Wired to CN2pin10
7	EXT +5V	+5V power supply input. Power jumper CN3 must be installed
8	Gnd	Power supply ground
9	Fault	Fault output.
10	Gnd	Power supply ground
11	+5V	+5V Power Supply output
12	AEN	Amplifier Enable input
13	NC	Not connected
14	AnGnd	Analog ground. Wired to CN4pin14
15	ADCin	Analog input. Wired to CN4pin15
16	DACout	Analog output. Wired to CN4pin16

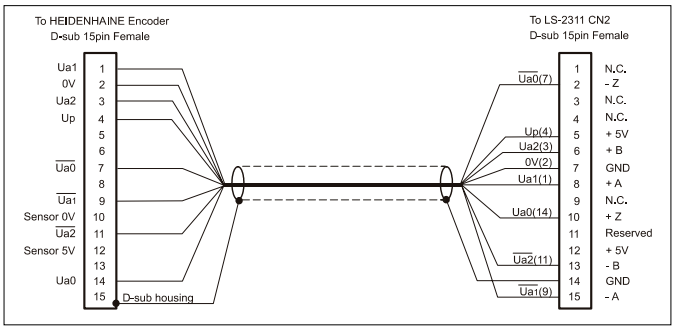
CN4 – LS-231SE interface

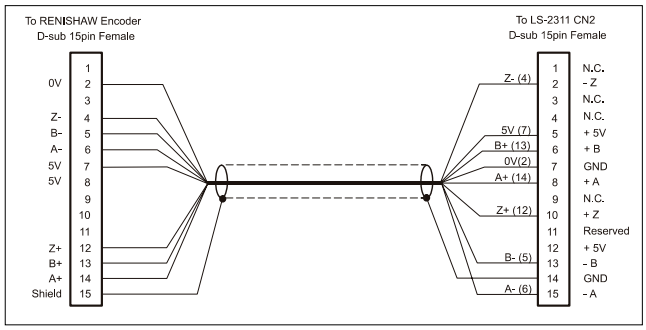
PIN	SIGNAL	DESCRIPTION
1	DACout	Analog output. Wired to CN3pin16
2	ADCin	Analog input. Wired to CN3pin15
3	AnGnd	Analog ground. Wired to CN3pin14
4	PhB	Master encoder phase B output to LS-231SE
5	AEN/Z	Amplifier Enable output
6	EXT +5V	+5V input Power Source jumper CN4 must be installed
7	Gnd	Power supply ground
8	Fault In	Fault input from LS-231SE
9	Gnd	Power supply ground
10	PhA	Master encoder phase A output to LS-231SE

LS-2311 Power Supply Source



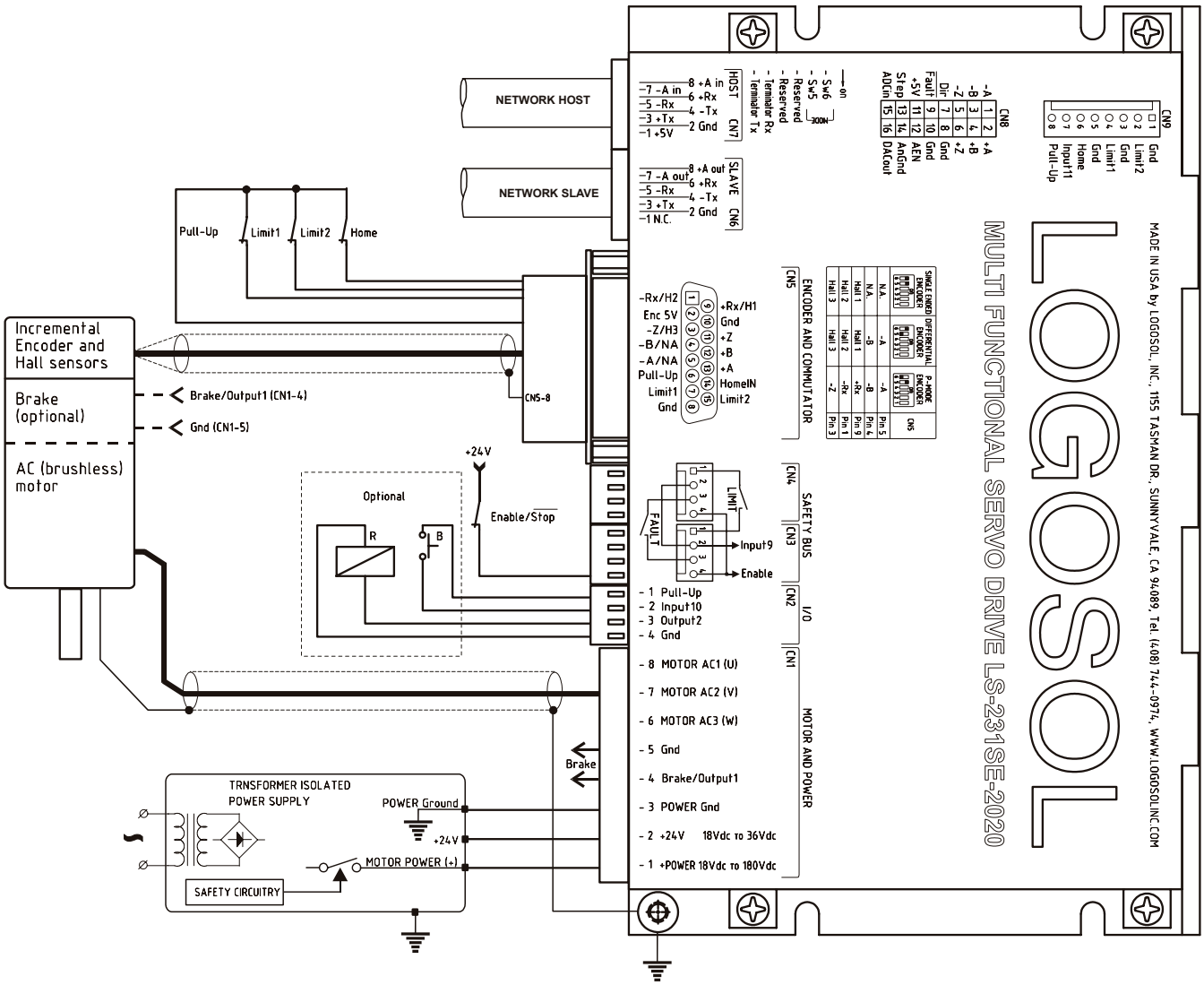
LS-231SE Master Encoder Cables





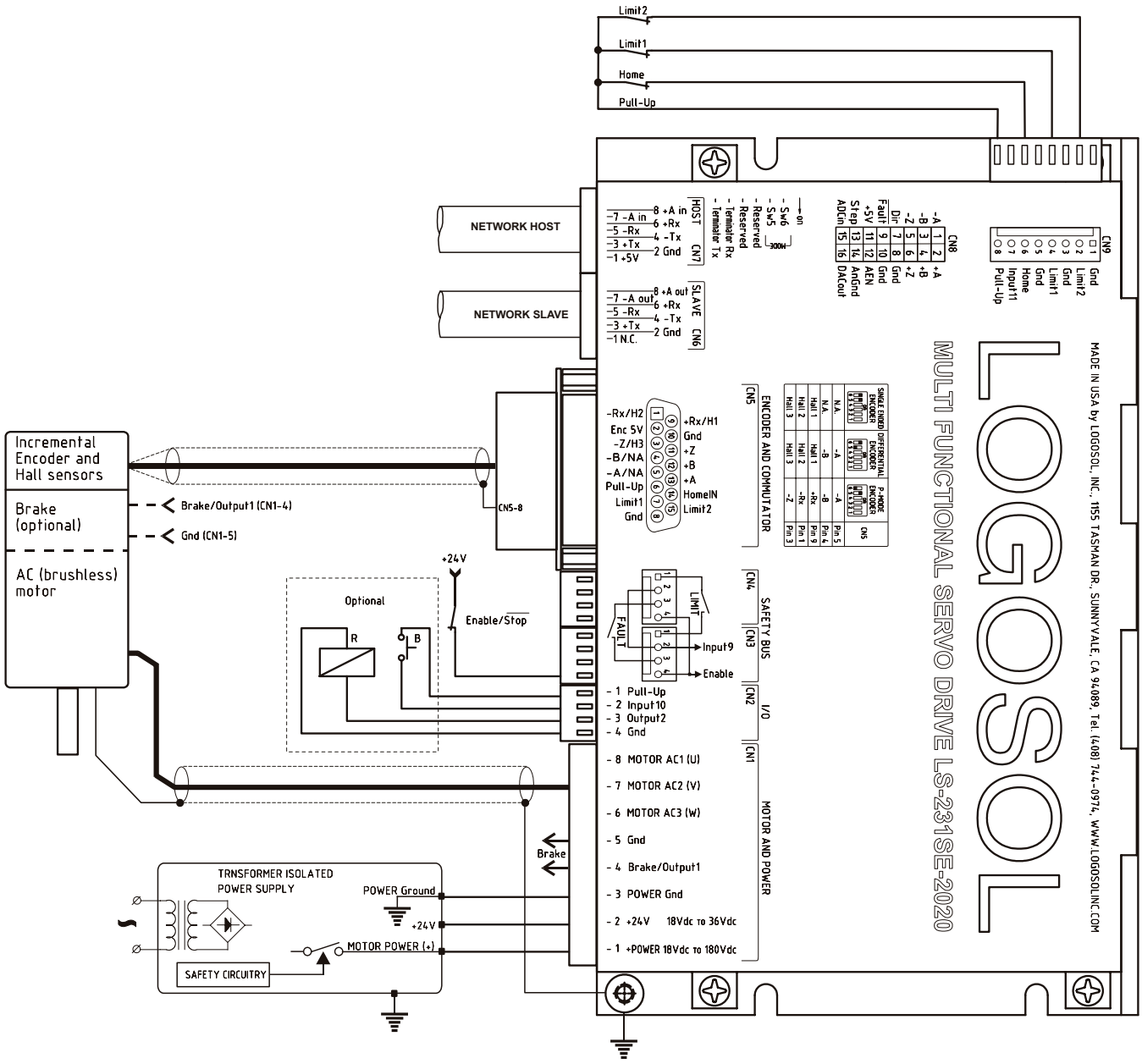
<u>SAMPLE APPLICATION</u> using AC (brushless) motor in *LDCN mode*

Home switch and limit switches are connected to CN5.



<u>SAMPLE APPLICATION</u> using AC (brushless) motor in *LDCN mode*

Home switch and limit switches are connected to CN8.

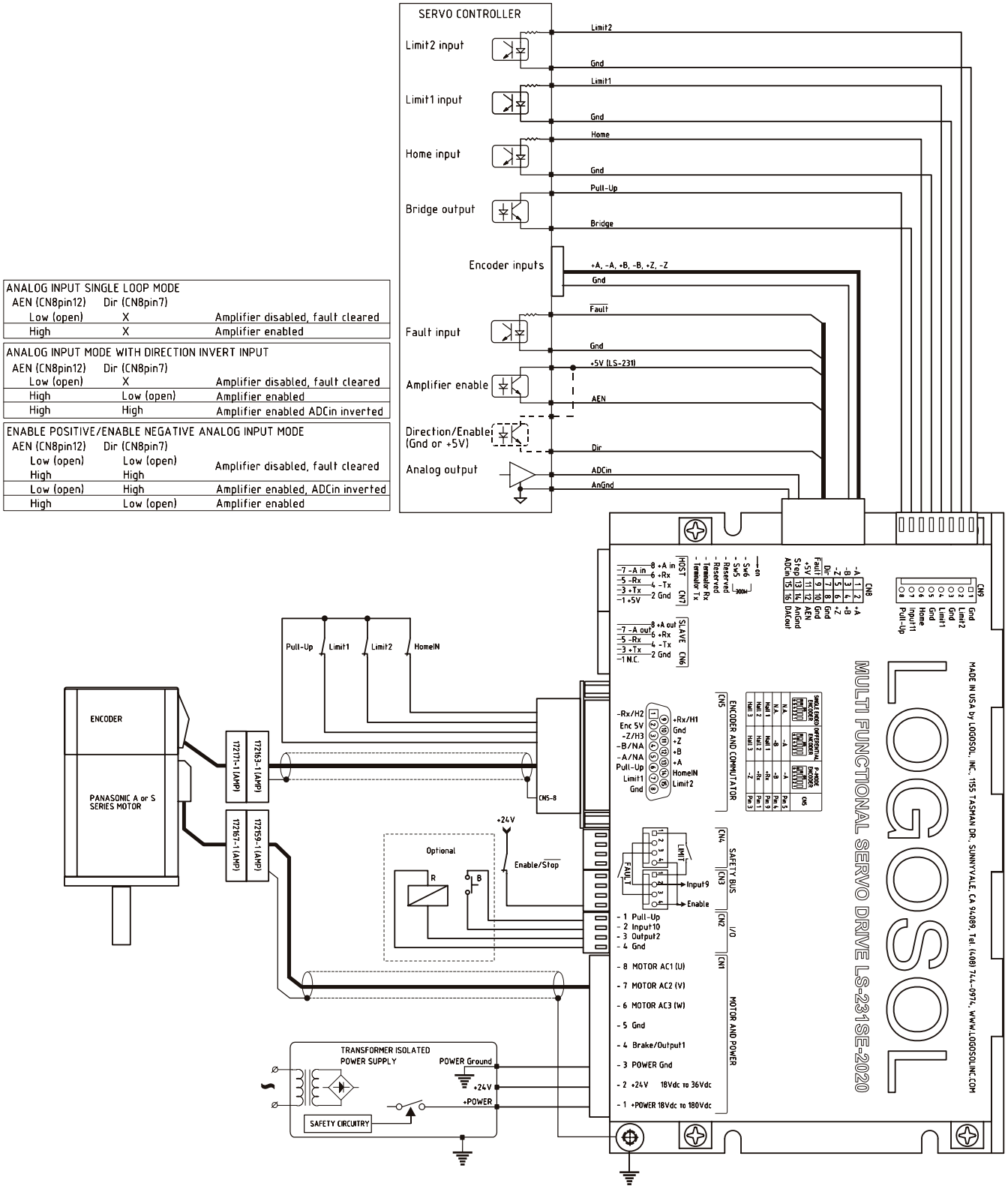


SAMPLE APPLICATION using Panasonic A or S series motors in *Analog input mode*

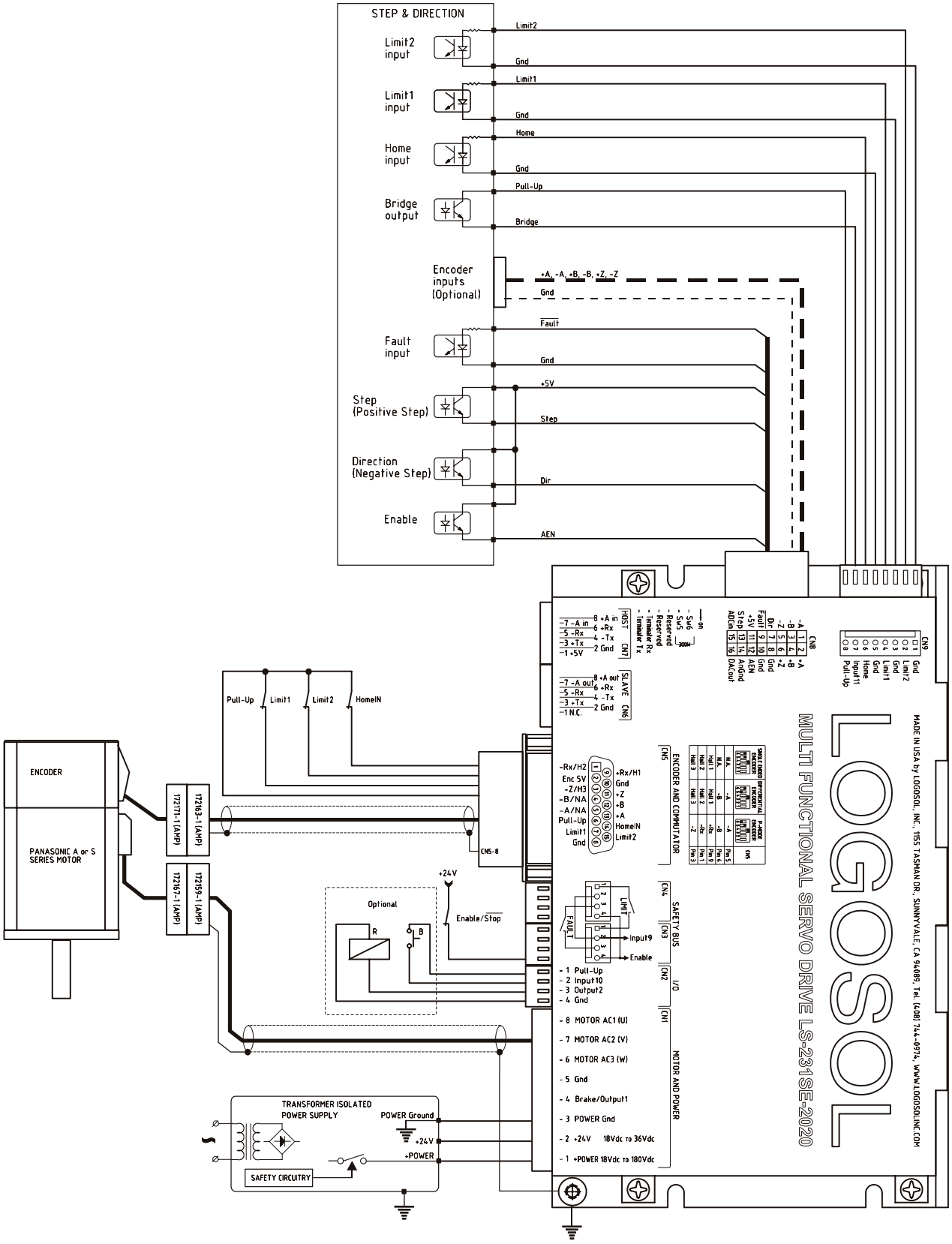
ANALOG INPUT SINGLE LOOP MODE		
AEN (CN8pin12)	Dir (CN8pin7)	
Low (open)	X	Amplifier disabled, fault cleared
High	X	Amplifier enabled

ANALOG INPUT MODE WITH DIRECTION INVERT INPUT		
AEN (CN8pin12)	Dir (CN8pin7)	
Low (open)	X	Amplifier disabled, fault cleared
High	Low (open)	Amplifier enabled
High	High	Amplifier enabled ADCin inverted

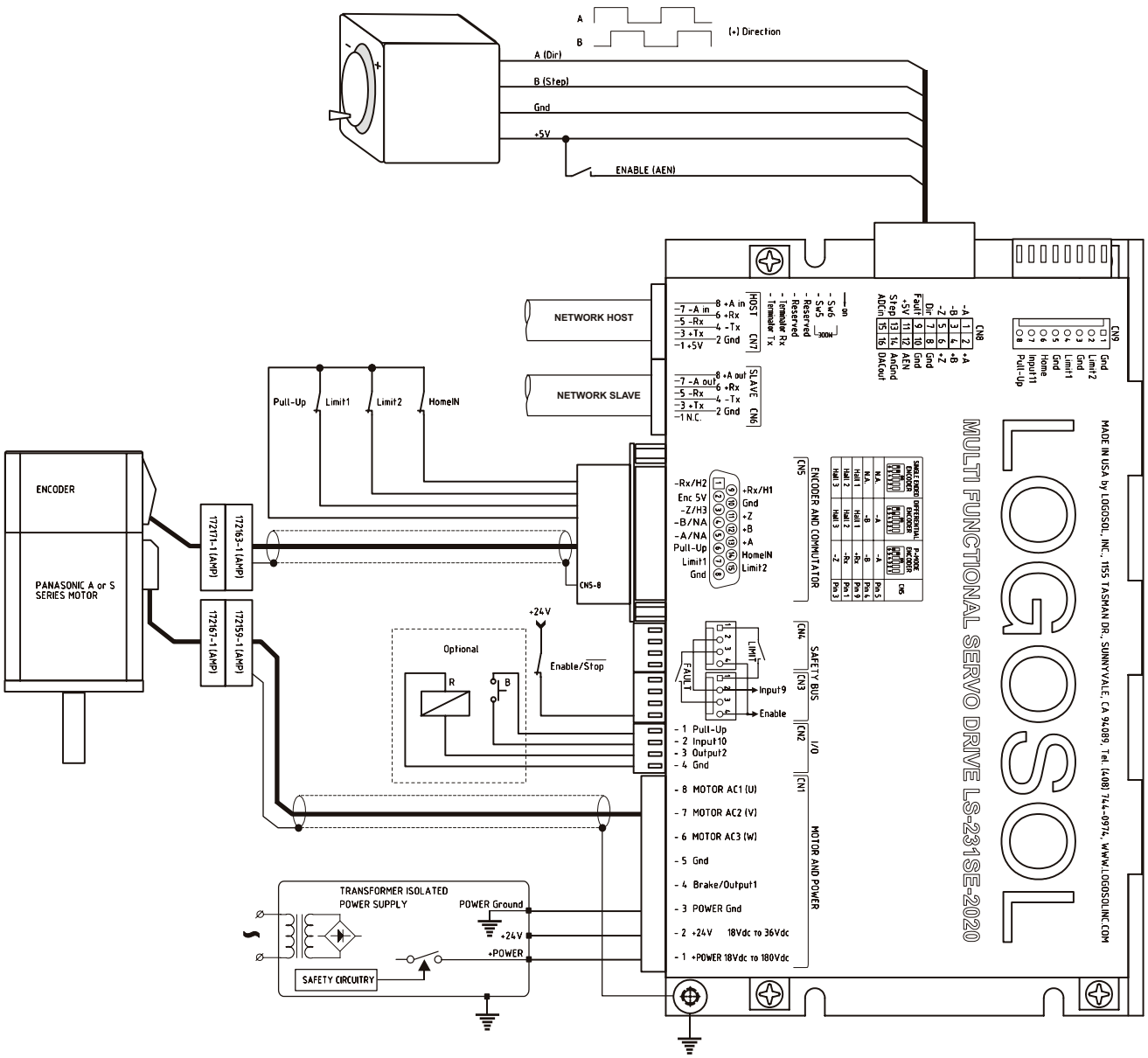
ENABLE POSITIVE/ENABLE NEGATIVE ANALOG INPUT MODE		
AEN (CN8pin12)	Dir (CN8pin7)	
Low (open)	Low (open)	Amplifier disabled, fault cleared
High	High	
Low (open)	High	Amplifier enabled, ADCin inverted
High	Low (open)	Amplifier enabled



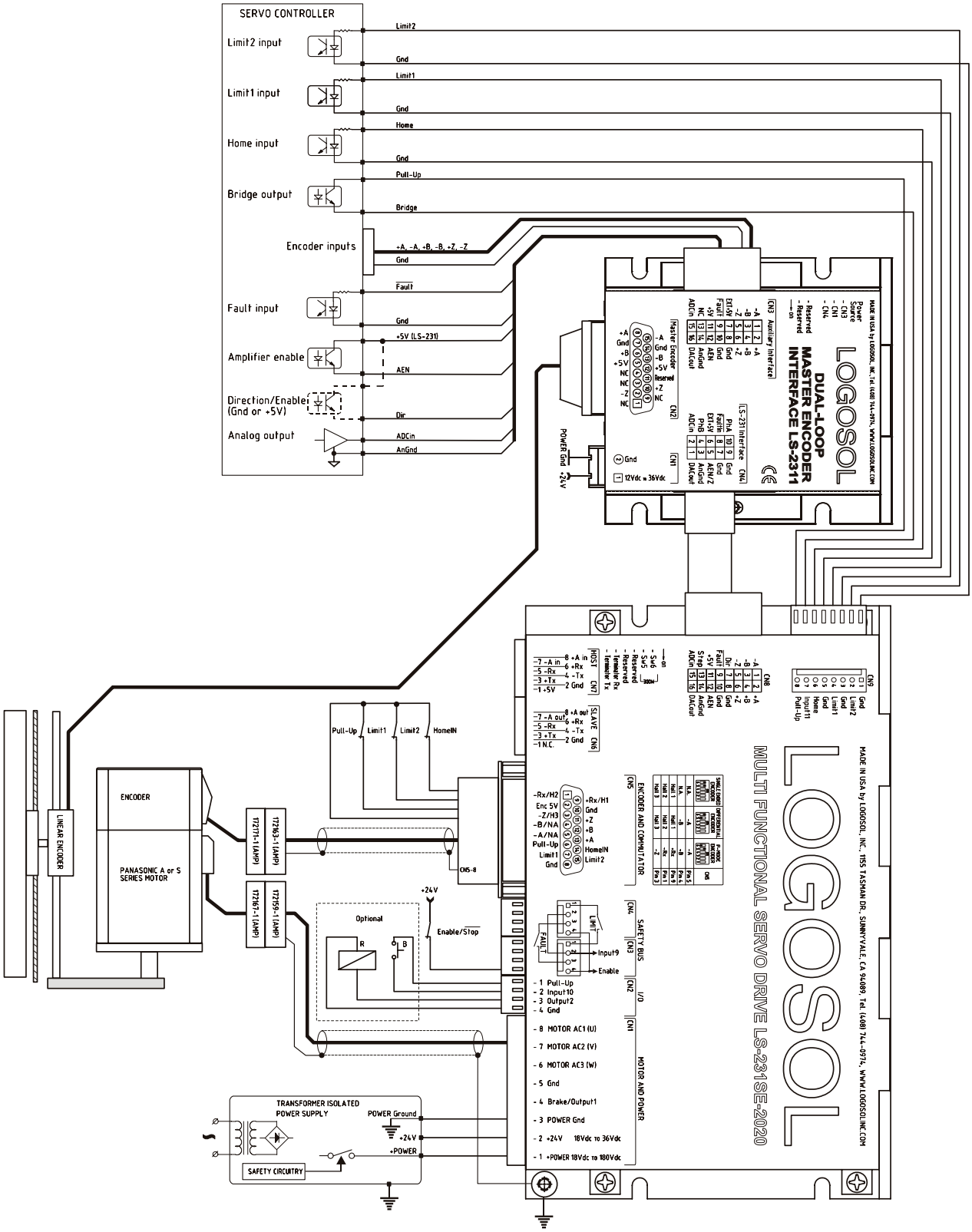
<u>SAMPLE APPLICATION</u> using Panasonic A or S series motors in *Step & Direction (Step Positive/Step Negative) mode*



<u>SAMPLE APPLICATION</u> using Panasonic A or S series motors in *Quadrature encoder mode*

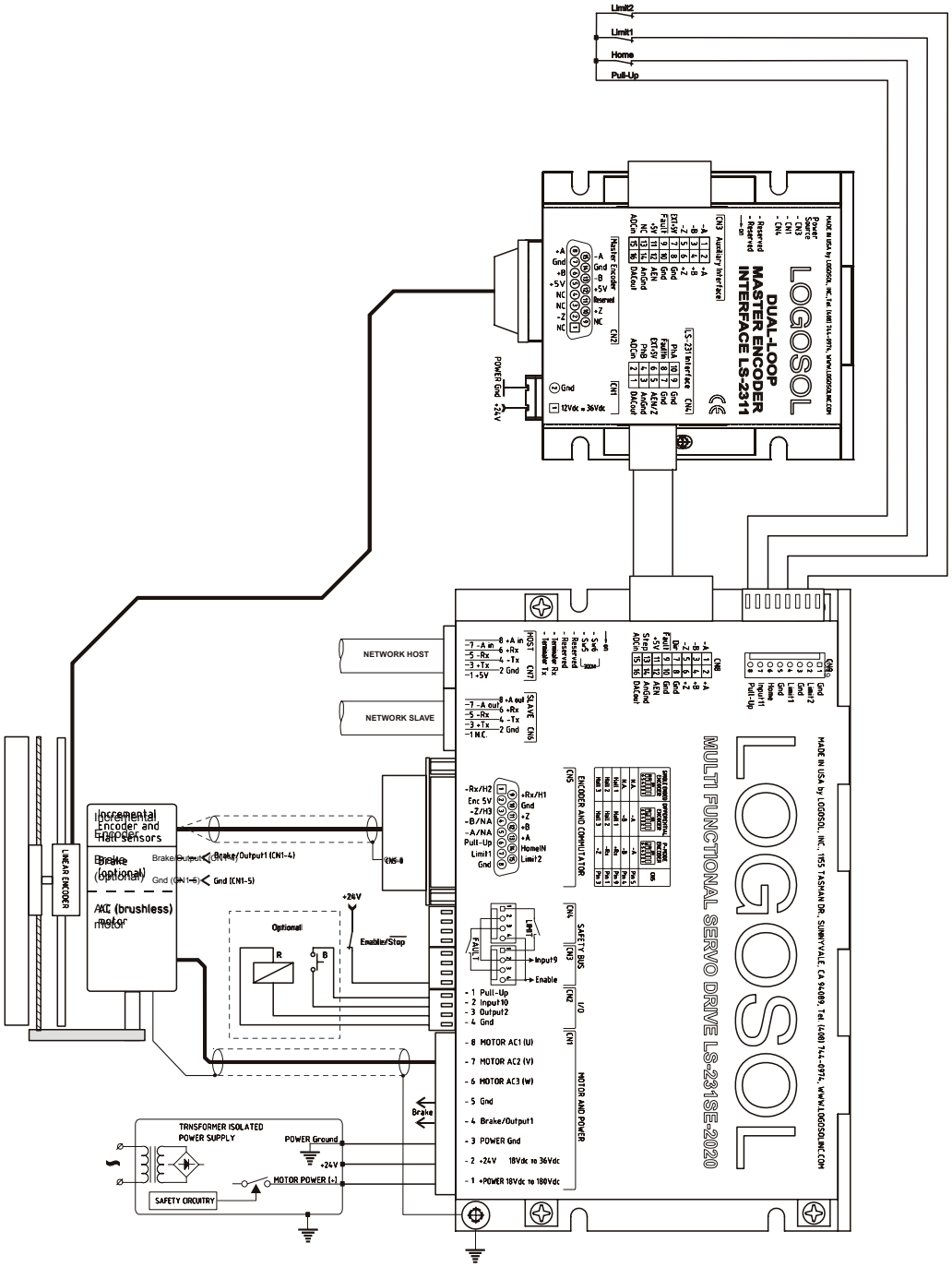


SAMPLE APPLICATION ***Analog Input Dual-Loop*** **mode using Panasonic A or S series motors**



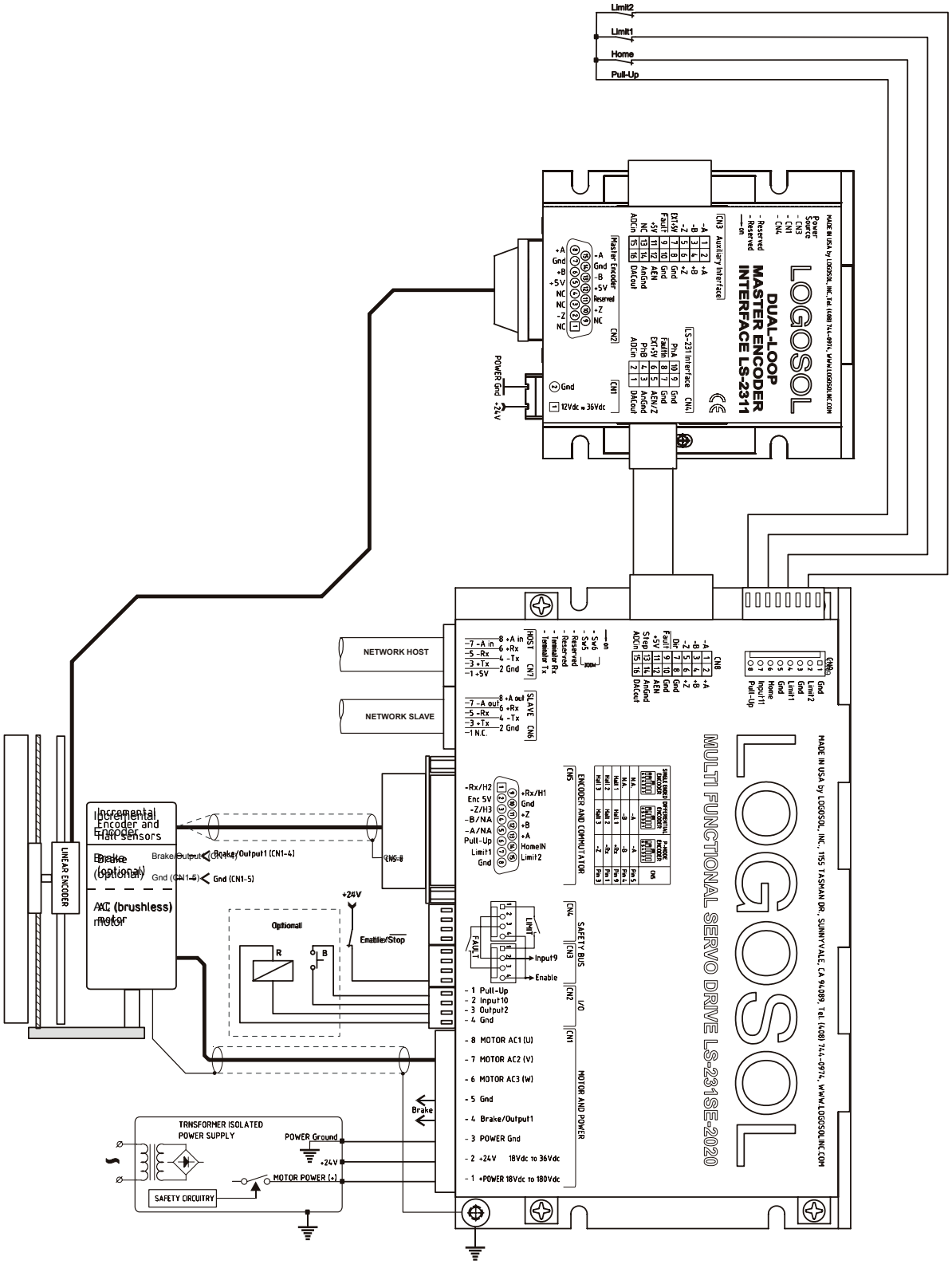
SAMPLE APPLICATION ***LDCN Dual-Loop*** **mode using AC (brushless) motor**

Home switch and limit switches are connected to CN5.

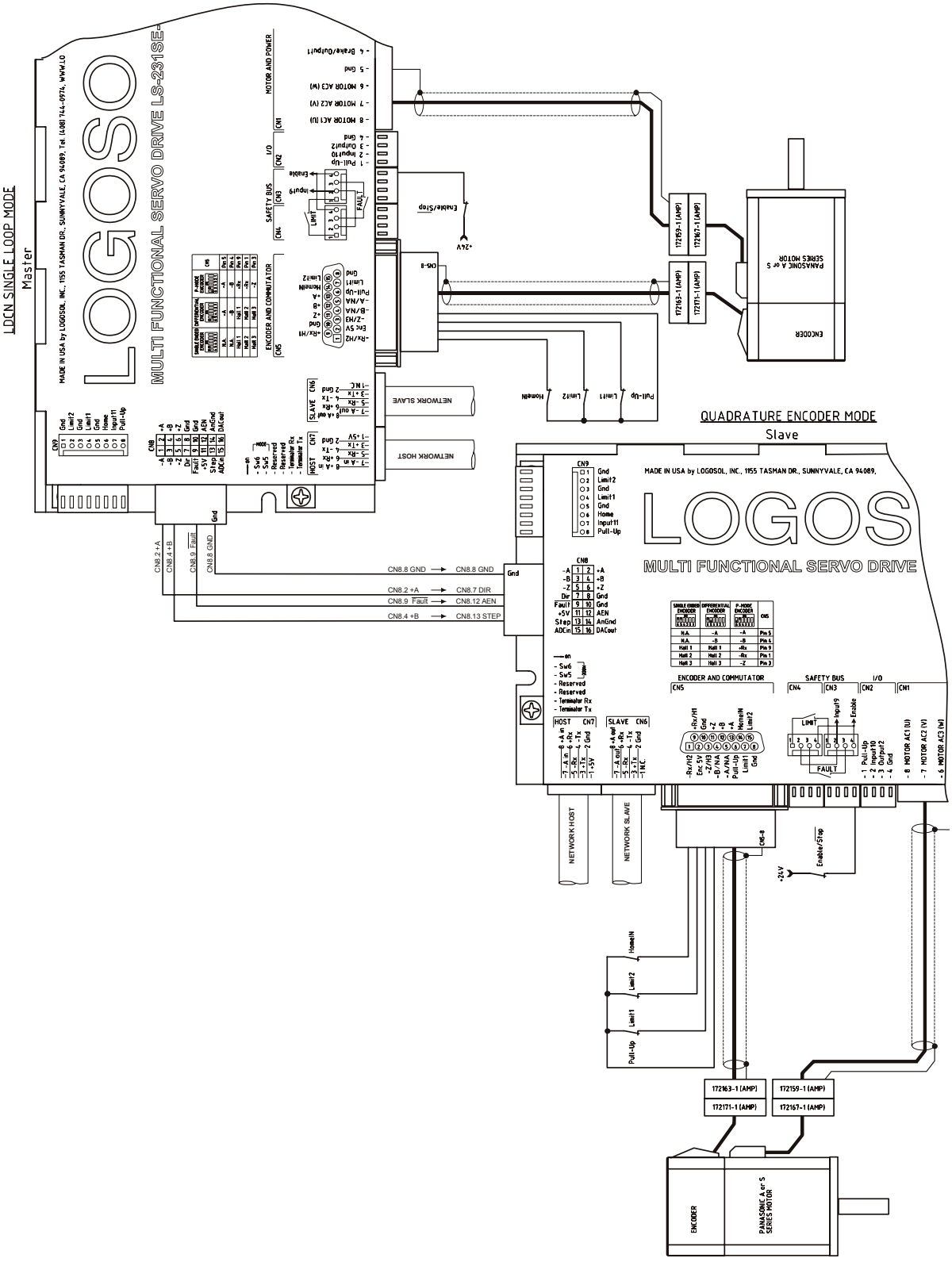


SAMPLE APPLICATION - ***LDCN Dual-Loop*** **mode using AC (brushless) motor**

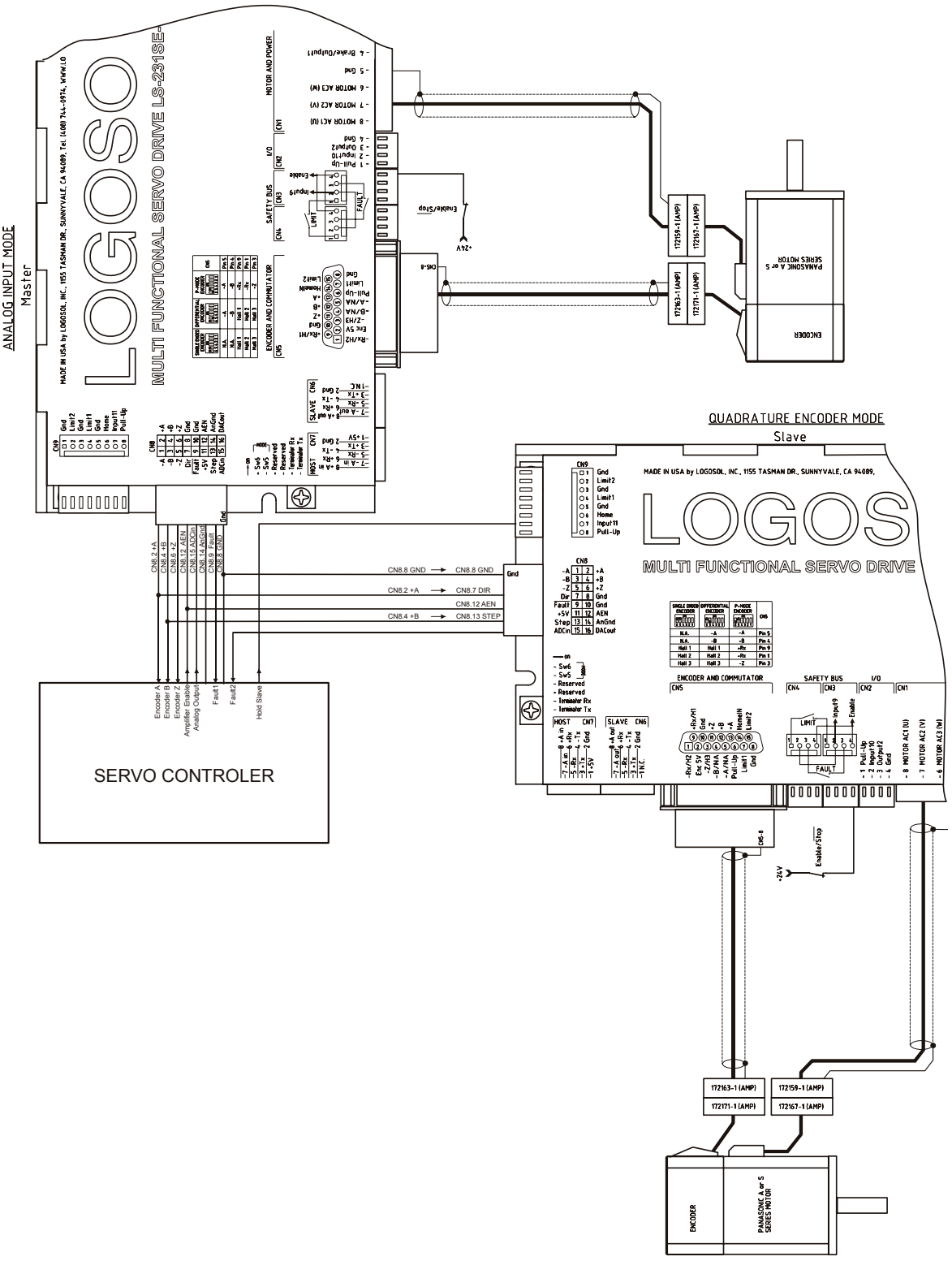
Home switch and limit switches are connected to CN8.



SAMPLE APPLICATION - LDCN mode using Master and Slave motors

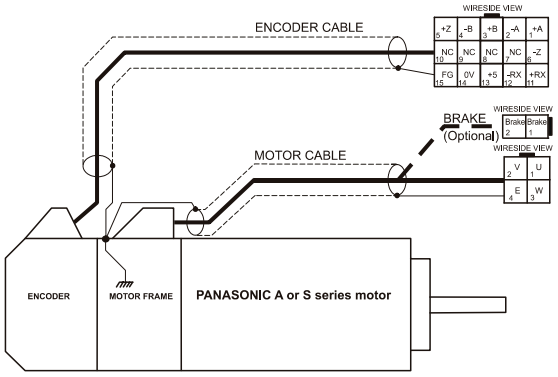


SAMPLE APPLICATION – ***Analog input using Master and Slave motors***



PANASONIC A AND S SERIES MOTORS

WIRING DIAGRAM



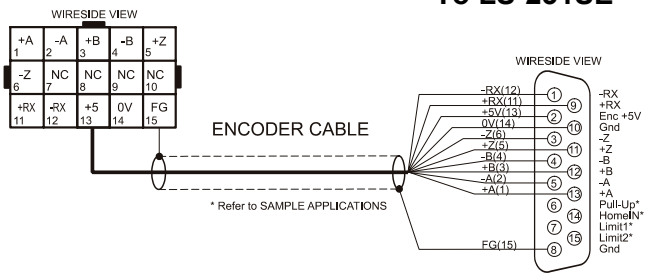
ENCODER CONNECTOR		
PIN#	SIGNAL NAME	WIRE COLOR
1	+ A channel output	Red
2	- A channel output	Pink
3	+ B channel output	Green
4	- B channel output	Blue
5	+ Z channel output	Yellow
6	- Z channel output	Orange
7	NC	NA
8	NC	NA
9	NC	NA
10	NC	NA
11	+RX	Light blue
12	-RX	Purple
13	+5V	White
14	0V	Black
15	FG = motor frame	Black

MOTOR CONNECTOR		
PIN#	SIGNAL NAME	WIRE COLOR
1	U phase	Red
2	V phase	White
3	W phase	Black
4	E = motor frame	Green/yellow

BRAKE CONNECTOR (Optional)		
PIN#	SIGNAL NAME	WIRE COLOR
1	Brake	Yellow
2	Brake	Yellow

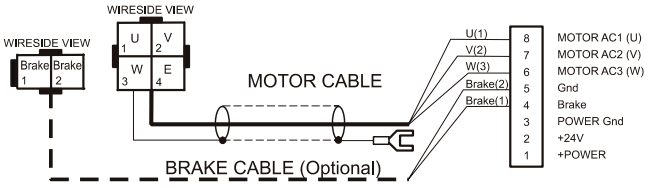
EXTENSION CABLE

To LS-231SE



MOTOR CONNECTOR AMP CAP 172163-1 10 pins 170365-1		LS-231SE CONNECTOR AMP 205206-3 D-Shape 15 Pin Male Crimp Housing 15 pins AMP 5-66507-7	
PIN#	SIGNAL NAME	PIN#	SIGNAL NAME
1	+ A channel output	13	+A
2	- A channel output	5	-A
3	+ B channel output	12	+B
4	- B channel output	4	-B
5	+ Z channel output	11	+Z
6	- Z channel output	3	-Z
11	+RX	9	+RX
12	-RX	1	-RX
13	+5V	2	Enc +5V
14	0V	10	Gnd
FG	SHIELD	8	Gnd (SHIELD)
NA	NA	6	Pull-Up
NA	NA	14	HomeIN
NA	NA	7	Limit1
NA	NA	15	Limit2

To LS-231SE



MOTOR CONNECTOR AMP CAP 172159-1 4 pins 170366-1		LS-231SE CONNECTOR PHOENIX CONTACT MSTB2.5/8-ST-5.08	
PIN#	SIGNAL NAME	PIN#	SIGNAL NAME
1	U phase	8	MOTOR AC1 (U)
2	V phase	7	MOTOR AC2 (V)
3	W phase	6	MOTOR AC3 (W).
4	E=motor frame		⏚ (SHIELD)

BRAKE (OPTIONAL) AMP CAP 172157-1 2 pins 170366-1		LS-231SE CONNECTOR PHOENIX CONTACT MSTB2.5/8-ST-5.08	
PIN#	SIGNAL NAME	PIN#	SIGNAL NAME
1	Brake	4	Brake/Output1
2	Brake	5	Gnd

LOGOSOL LS-231SE QUICK START GUIDE

LDCN mode

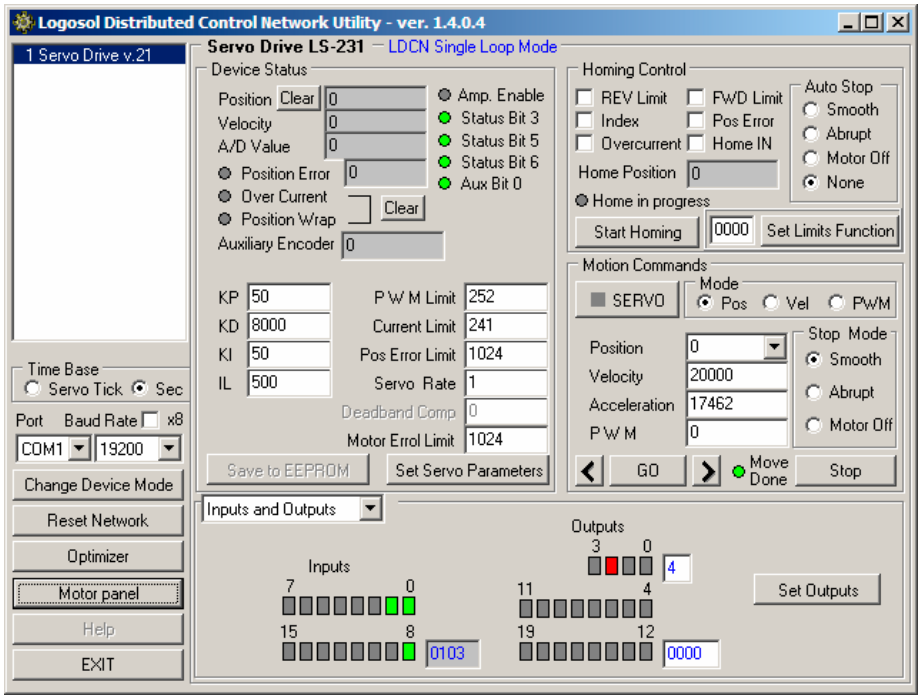
1. Set DIP-switches 5 and 6 depending on the motor type.
2. Connect power supply to LS-231SE.
3. Connect your motor, encoder, hall sensors, and any other I/O you may have.
4. Connect RS-232 adapter and RJ-45 network cable between LS-231SE and your host computer.

Software installation

1. Insert Logosol Distributed Control Network Utility installation disk into the floppy drive.
2. Select RUN from the Windows 95/98/2000/NT/XP 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.

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 **Reset network** button.
5. If the motor is not initialized complete **Motor Initialization** procedure.
6. Click **SERVO** button.
7. 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.



Motor Initialization

1. Complete steps 1 to 4 from *Initial Connection to the Host* section.
2. Set the drive in LDCN mode by clicking Change Device Mode button and select LDCN mode.
3. Click **Motor Panel** button.
4. Set the motor parameters.

   Peak current, Peak time – Peak current is % of the MAX MOTOR OUTPUT CURRENT (20A for LS-231SE-2020). The output current will be limited to the Peak current for Time=Peak time. After that the current will be limited to the Continuous current. Peak current (%) and Peak time should be set depending on **motor and application** parameters.

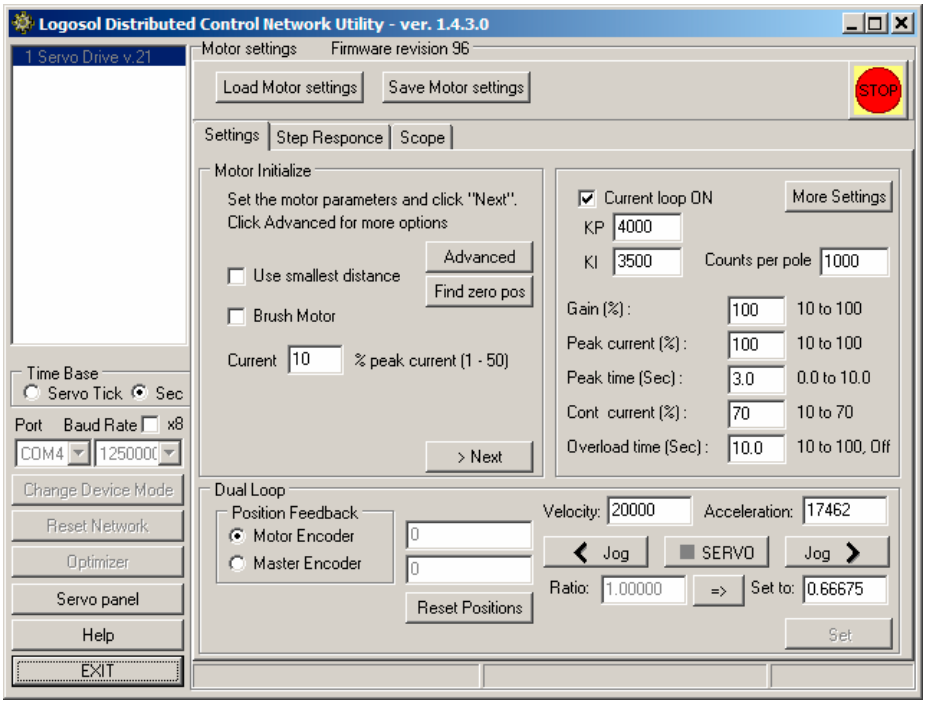
   Continuous current, Overload time – the drive output will be limited to this value for overload time. After the expiring of the Overload time the output will be disabled. If "Off" is set the drive output will not be affected. Continuous current (%) and Overload time should be set depending on **motor and application** parameters.

   Gain – set Gain=100%. This parameter sets the ratio between command value and the drive PWM output. Gain=50% will limit the PWM output value to 1/2 of its maximum.

   Current – motor current will be limited to this value during the **initialization procedure only**. If the current is too low the initialization procedure will not work. Higher values may damage the motor. Start with values 5%-20% and increase the current depending on the results.

   More settings – click More Settings to select encoder resolution.

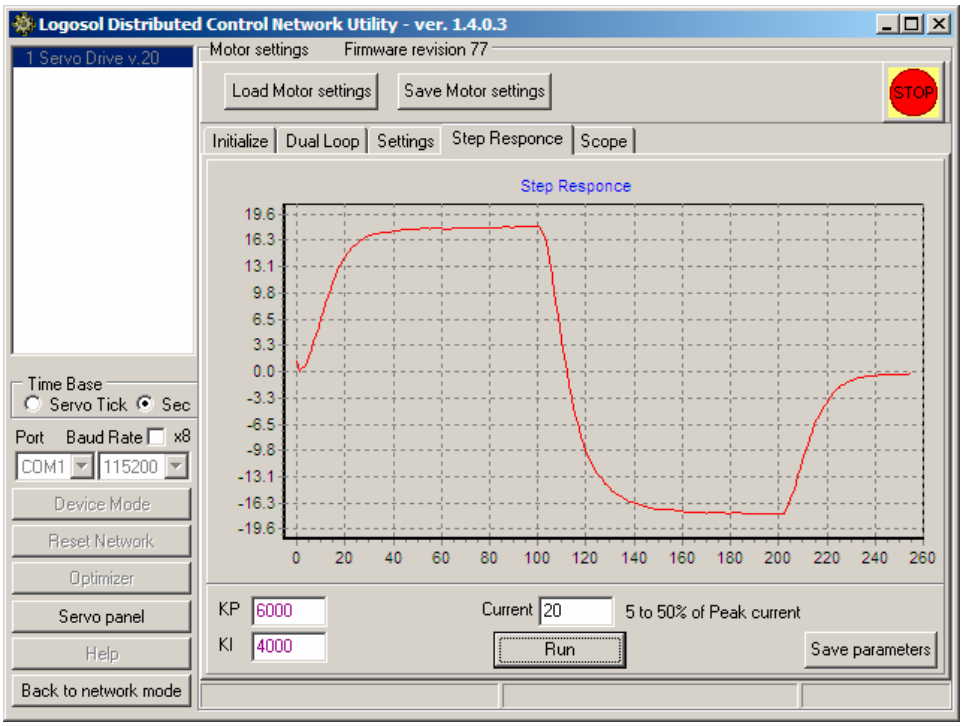
5. Select the motor type (brush or brushless).
6. To start the initialization click **Next** and follow the instructions on the screen.



Motor Tuning – all modes

Setting Current loop parameters

In some applications using "current loop" mode will improve the drive performance. Check **Current loop** ON box and click Set button to activate this mode. To set **KP** and **KI** values select **Step Response** window. Try to get the current chart as close as it is possible to square wave with minimum overshooting and oscillation.



Setting the drive in dual servo loop mode

The drive should be initialized in single servo loop mode before setting it to dual loop mode.

1. Turn off the power supply.
2. Connect the master encoder and the dual loop master encoder interface module LS-2311 accordingly to one of the sample applications described earlier.
3. Turn on the power supply and run LDCN Utility.
4. While still in single loop mode, run the motor and make sure the auxiliary encoder position on the screen changes.
5. Click **Motor Panel** button.
6. Calculate the ratio between the master encoder and the motor encoder and set the value in **Set To** edit box. If the directions of the two encoders do not match, type in a negative value. To make sure the ratio is correct, click **Reset Positions** button and using the **Jog** buttons on the screen, move the motor. The utility calculates the ratio and displays its value on the screen. This value should be approximately the same as the calculated ratio.
7. Select **Master Encoder** in the **Position Feedback** radio group and click **Set** button.

Note: The dual loop mode is available only in LDCN mode and in the standard Analog Input mode. Switching between these two modes does not change the single/dual loop mode. However, setting the drive in any other mode sets the drive in single loop mode.

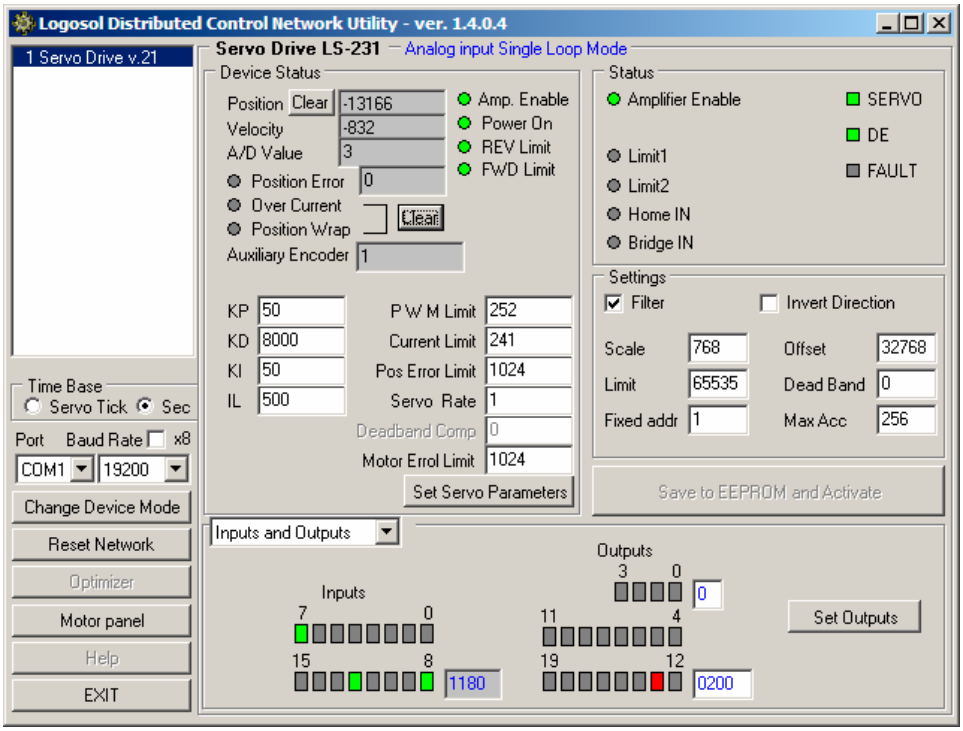
Mode Selection

Various modes can be set using **Change Device Mode** button**.** Clicking this button activates a menu with all possible modes of the controller. After setting the device mode, the PID parameters and all other mode specific parameters should be set and saved to the non-volatile memory of the drive.

When an analog input mode is selected, set analog input parameters. **Filter** enables/disables filtering of the analog input. **Invert direction** can be used to invert the motor direction. **Scale** is the factor used to convert the analog input value (-32768 to 32767) to velocity. For example, scale of 10 produces maximum velocity of 327670 or 97653 encoder counts per second (327670 / 3.3554432 = 97653). The **offset** determines the zero analog input and is usually set to 32767 – middle of the whole range. **Limit** is the maximum analog value. **Deadband** limits the minimum non-zero value thus reducing the oscillations at zero velocity. **Acceleration** value determines how quickly the commanded velocity will be reached. For example, a value of 256 corresponds to 1,490,116 encoder count/s/s (256 / 0.00017179869).

In all step modes, the **step rate** should be set. It represents the factor between steps and encoder counts.

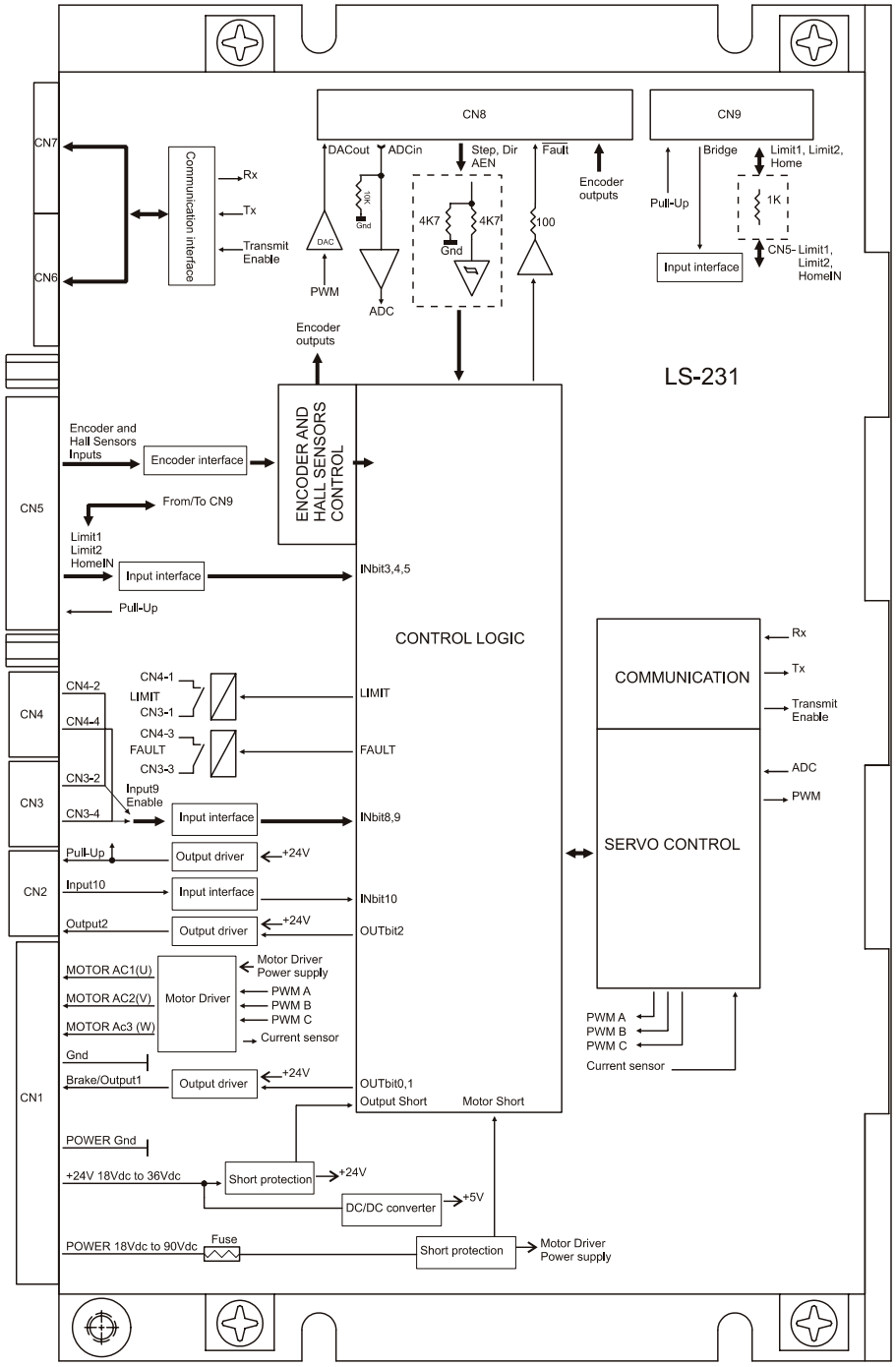
Fixed address should be set to a value between 1 and 31. It is recommended to use different values for each drive in the network.



LS-231SE ARCHITECTURE

Overview

LS-231SE Logosol Multifunctional Servo Drive is a highly integrated servo control module for Brushless motors including motion controller, servo amplifier with integrated sinusoidal output, Serial communication interface, Amplifier interface, Step & direction interface, Safety bus and protection circuit. LS-231SE can be configured to work in 8 different modes.



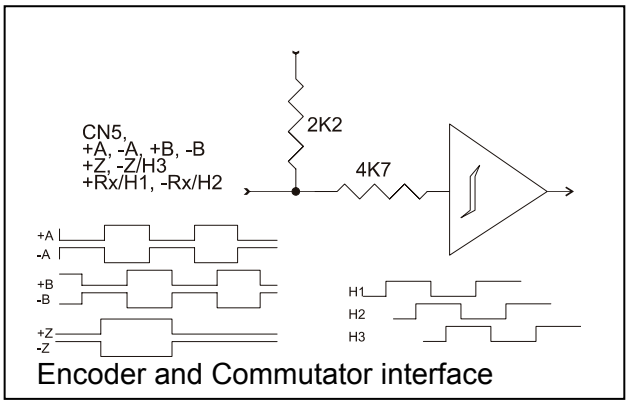
Block diagram

Encoder and Commutator interface

The Encoder interface accepts two square wave inputs – +A, +B (+A, -A, +B, and -B for differential encoders) from an incremental encoder. Ideally these square waves are 50% duty cycle and +/-90 degrees out of phase. The time between encoder state transitions is limited and should not be less than 0.1uS when encoder filter is off and 0.2uS when encoder filter is on. With ideally formed encoder pulses, this would correspond to a 2500 line encoder (10000 counts/rev) rotating at 60,000/30,000 rpm. Refer to Logosol Quick start guide / Motor initialization to select the encoder mode.

Encoder Index +Z (+Z and –Z for differential encoders) is used by Servo control module to capture Home position.

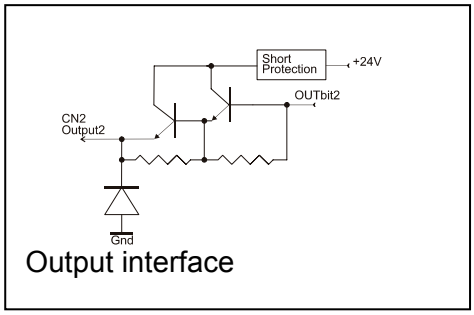
Hall sensors provide the motor position. They are used by motor control module to determine motor position after power up and during the motor initialization procedure.



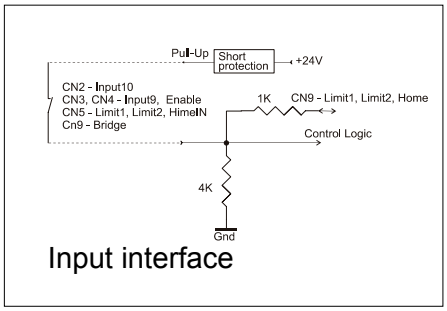
Encoder and Commutator interface

Besides the position control, encoder phases, index and hall sensors are used to determine the motor commutation synchronizing the output PWM every motor rotation according to hall and encoder data.

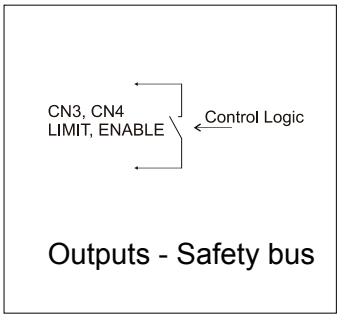
Control interface



Output interface



Input interface



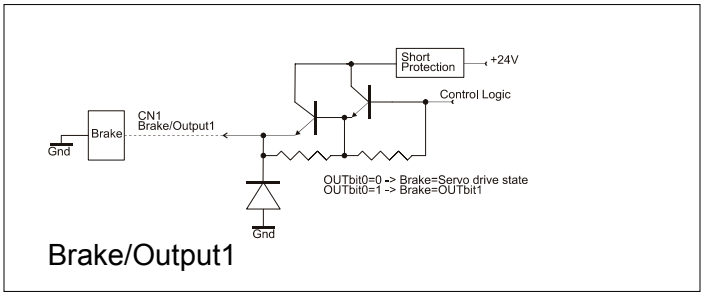
Outputs - Safety bus

Refer to *Safety Features - LS-231SE diagnostic and I/O* section for inputs/outputs descriptions.

Brake/Output1

When OUTbit0=1 Brake/Output1 follows the state of OUTbit1.

When OUTbit0=0 Brake/Output1 follows the Servo drive state described in *LS-231SE Diagnostic and I/O*. Brake current must not exceed 0.3A



Brake/Output1

Amplifier interface and Step & Direction interface.

If Analog mode is selected, ADC (analog input) is differential and may be adjusted between ±2.5V to ±10.0V. The load is 10K.

"Analog offset" is a constant added to the amplifier's input control voltage and can be used in applications with asymmetric load. Refer to *Safety Features - LS-231SE diagnostic and I/O* section for Analog mode description.

Step & Direction mode uses the same interface. In this mode Step, Dir (direction) and AEN (enable) inputs are active. Refer to *Safety Features - LS-231SE diagnostic and I/O* section for Step & Direction mode description.

Fault output and Encoder (+A, -A, +B, -B, +Z, -Z) outputs are TTL compatible.

All digital inputs are HC (high speed CMOS) compatible and with 4K7 pull-down resistors.

Serial Command Interface

Serial communication with the LS-231SE 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 supports full-duplex multi drop RS-485 interface that allows multiple drives to be controlled over a single RS-485 port. The host sends commands over it's RS-485 transmit line and receives all status data over shared RS-485 receive line.

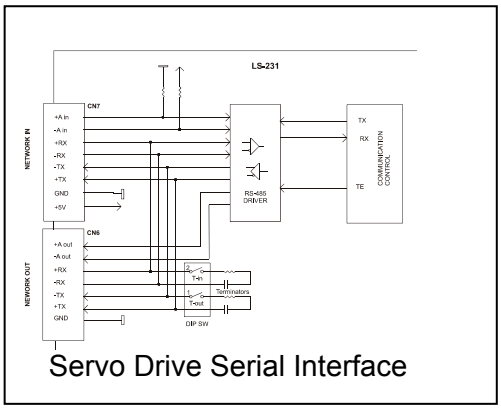
The command protocol is a strict master/slave protocol in which the host sends a command packet to a specific slave. The data are stored in the slave buffer until the end of the servo cycle (51.2μS) and then the command is executed. Then the slave Drive sends back a status packet. Typically the host 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 consist 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-bit 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 drive will send back a status packet consisting of:

Status byte
0-23 optional bytes of status data
Checksum byte



Servo Drive Serial Interface

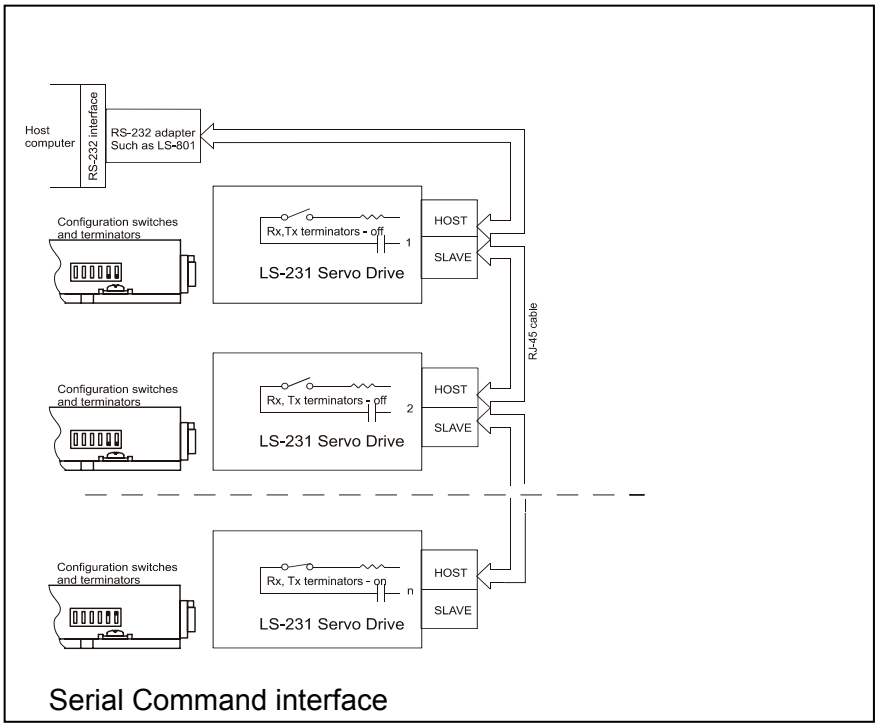
The status byte contains basic status information about the Drive, including a checksum error flag for the command just received. The optional data bytes may include data such as the position, velocity, etc. and are programmable by the host. The checksum byte is the 8-bit sum of the status byte and the additional status data bytes.

The transmission of all 16-bit and 32-bit data is always with the least significant byte first.

Addressing

Rather than hard-wired or switch-selected address, the host dynamically sets the address of each LS-231SE with the aid of the daisy-chained *A in* and *A out* lines. This allows additional drives to be added to a RS-485 network with no hardware changes. *A in* of the first Drive is pulled low, its communication is enabled and the default address is 0x00. When the "*Set Address"* command is issued to give this Drive new unique address, it will lower it's *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.

In any non-LDCN modes, each LS-231SE should have a unique fixed address. In this case, the host can establish communication with a controller without sending Hard Reset command.



Serial Command interface

Group Addresses

In addition to the individual address, each controller has a secondary group address. Several LS-231SE 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-231SE 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. When 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.

Changing communication rates

The default baud rate after power-up is 19.2Kb/S. Baud rates up to 1250Kb/S 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-231SE uses a "proportional-integral-derivative", or PID filter. The position, velocity and acceleration are programmed as 32-bit quantities in units of encoder counts per servo ticks. For example, a velocity of 10 revolutions per second of a motor with a 500 line encoder (2000 counts/rev) at a tick time of 51.2 µSec. would correspond to a velocity of 1.024 counts/tick. Velocities and accelerations use the lower 16 bits as a fractional component so the actual programmed velocity would be 1.024 x 65536 or 67109. An acceleration of 40 rev/sec/sec (which would bring us up to the desired speed in ¼ sec) would be 0.00021 counts/tick/tick; with the lower 16 bits the fractional component, this would be programmed as 0.00021 x 65536 or 14. 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.

PWM mode

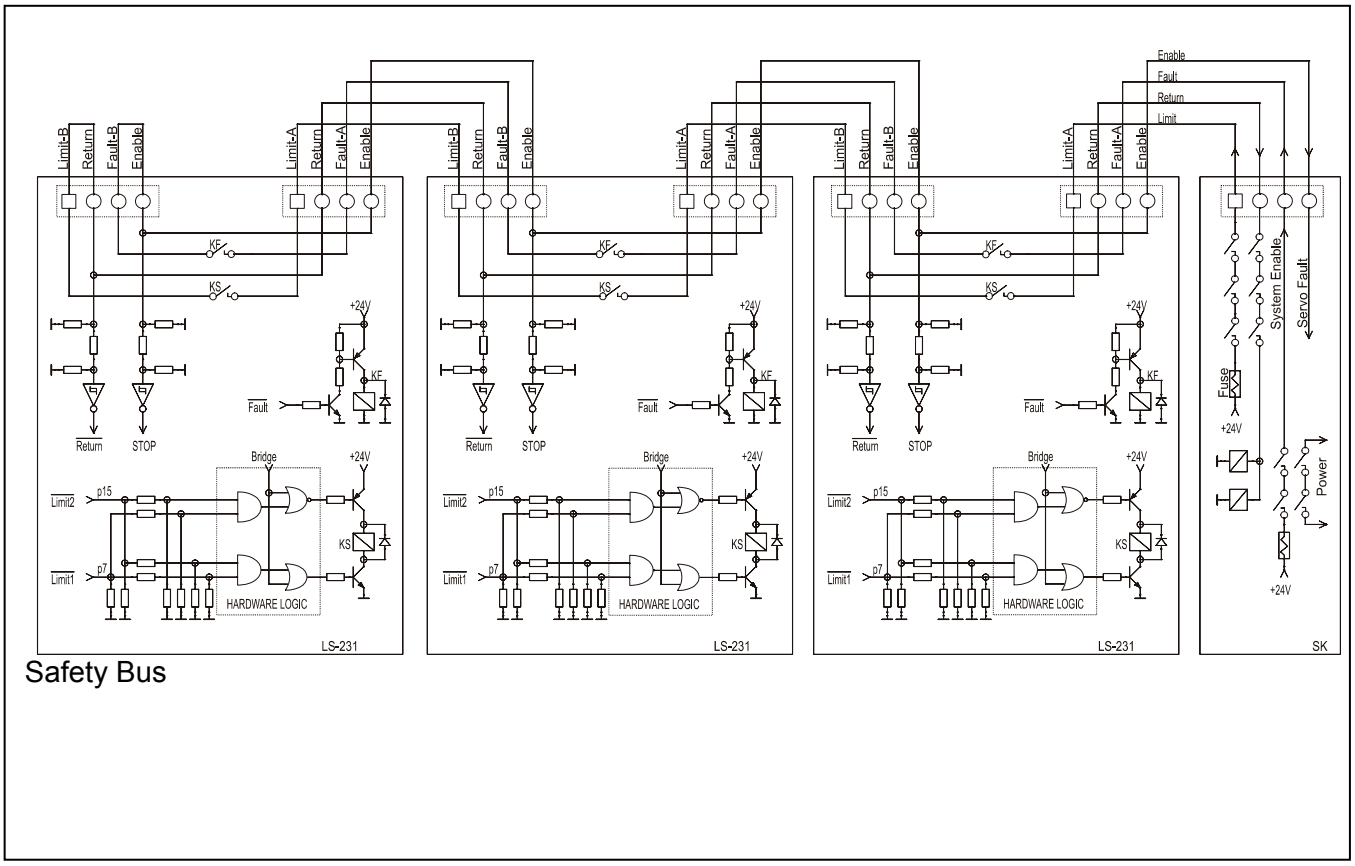
If the position servo is disabled, the motor is operated in a raw PWM output mode and no trapezoidal or velocity profiling is performed. In this mode, a user specified PWM value is outputted directly to the amplifier. A PWM value of 255 corresponds to 100% and a value of 0 corresponds to 0%. Command 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.

STATUS and SAFETY features

Safety Features

Safety Bus

Safety Bus interface is provided for safety operation for several drives. If the bus is connected any drive **FAULT** will disable all the drives connected to the network. The drive LEDs and the Status bits (*LS-231SE Diagnostic and I/O*) are used for diagnostic. For better control Safety bus can be connected to SK-2310 or other controller with similar functions.



Safety Bus

FAULT output depends on internal drive status. Refer to *LS-231SE Diagnostic and I/O* for details.
LIMIT output is closed when:

- **Limit1** and **Limit2** (CN5, CN9) are High;
- **BridgeSTA** bit is High.

To operate without *Safety bus* - connect **Enable** input to +24Vdc.
Motor short protection – the drive is protected against short circuit output to output or output to ground.
Overvoltage protection –The drive will be in Fault Condition and the drive output will be disabled when the input voltage is above specified limits (Refer to *Technical Specifications* section of this document).

LS-231SE Diagnostic and I/O

LDCN mode state and diagnostics

MODEbit[C,B,A] = 000

Status **Bit 6** Limit2	Status **Bit 5** Home Source	Status **Bit 4** Pos_error	Status **Bit 3** Power	Status **Bit 0** Move_done	Auxiliary Bit 2 Servo	Auxiliary **Bit 0** Index	Stop Cmd **Bit 0** Pic_ae≡DE	**CONDITION**	**ORANGE**	**GREEN**	**RED**	**FAULT**	**BRAKE** (OUTbit0=0)
0	1	1	0	1	0	1	0	**No Motor Power after LDCN Init**	ON	ON	OFF	RELAY Closed	Engaged
1	1	X	1	1	0	1	0	**AxisOFF**	OFF	Blink	OFF		
Limit2 Source	Home Source	0	1	X	1	Encoder	1	**ServoON**		ON			Released
					0			**ServoOFF**					
X	X	X	X	1	0	0	0	**ErrHALL**	ON	ON	Blink	CN8pin9 High	Engaged
				0	0	1	0	**ErrEEPROM**			Fast Blink		
0	1	1	0	1	0	1	0	**No Motor Power**	ON	ON	ON		
								Overheat	ON	OFF	Blink		
1	0	X	1	1	0	1	0	**Disabled**	Alternate		Blink		
1	0	X	1	1	0	0	0	**Master EncoderERROR**	Blink	OFF	Blink		
1	0	X	1	1	0	1	0	**Brake or Output Short**	Alternate		Blink	RELAY Open	
1	0	1	0	X	0	1	1	**Stopped**	Alternate		ON		
0	1	1	0	X	0	1	1	**MotorShort**	ON	OFF	Blink		
1	1	1	0	X	0	1	1	**Motor PowerDROP**	ON Blink	ON Blink	ON Blink	CN8pin9 Low	
0	0	1	0	X	0	1	1	**OverLOAD**	Blink	Blink	Blink		
1	0	1	0	X	0	0	1	**EncoderERR (Reset required)**	Blink	OFF			
X	X	1	1	1	0	X	1	**PositionERROR**	Blink	ON			

Amplifier mode state and diagnostics

MODEbit[C,B,A] ≠ 000

Status **Bit 6** Limit2	Status **Bit 5** Home Source	Status **Bit 4** Pos_error	Status **Bit 3** Power	Status **Bit 0** Move_done	Auxiliary Bit 2 Servo	Auxiliary **Bit 0** Index	**DE** INbit12	**CONDITION**	**ORANGE**	**GREEN**	**RED**	**FAULT** *RELAY*	**BRAKE** (OUTbit0=0)
Limit2 Source	Home Source	X	1	X	1	Encoder	1	**ServoON**	OFF	ON	OFF	Closed	Released
0	1	1	0	1	0	1	0	**No Motor Power after Power-UP**	ON	ON	OFF	Closed	Engaged
0	1	1	0	0	0	1	0	**EEPROM Checksum Error on PowerUP**	ON	Fast Blink	Fast Blink	Closed	
0	1	1	0	1	0	0	0	**Invalid HALL on PowerUP**	ON	Blink	Blink	Closed	
X	1	1	0	1	0	0	0	**PowerUP EncoderERR (Reset required)**	Blink	OFF	OFF	Closed	
0	1	1	0	X	0	X	0	**No Motor Power**	ON	ON	ON	Closed	
1	1	1	0	1	0	1	0	**Ready after PowerUP**	Blink	Blink	OFF	Closed	
								Ready	OFF	Blink	OFF	Closed	
								Ready after PowerDROP	Blink	Blink	Blink	Closed	
								Ready after PositionERROR	Blink	ON	Blink	Closed	
								Ready after MotorShort, Overheat or OverVoltage	ON	OFF	Blink	Closed	
0	1	1	0	1	0	1	0	**Overheat or OverVoltage**					
1	0	1	0	1	0	0	0	**EncoderERR (Reset required)**	Blink	OFF	Blink ON	Closed	
1	0	1	0	1	0	1	0	**Triggered Stop**	Alternate		Blink ON	Closed	
								Triggered Output short	Alternate		Blink ON	Open	
1	0	1	0	0	0	1	0	**EEPROM Checksum Error**	ON	Fast Blink	ON	Closed	
1	0	1	0	1	0	0	0	**Invalid HALL**	ON	Blink	ON	Closed	
1	1	1	0	0	0	1	1	**EEPROM Checksum Error**	ON	Fast Blink	ON	Open	
1	1	1	0	1	0	0	1	**Invalid HALL**	ON	Blink	ON	Open	
0	1	1	0	1	0	1	1	**Triggered MotorShort or Overheat**	ON	OFF	Blink	Open	
1	1	1	0	1	0	1	1	**Triggered Motor PowerDROP**	Blink	Blink	Blink	Open	
								No Motor Power	ON	ON	ON		
1	1	1	0	1	0	0	1	**Triggered PositionERROR**	Blink	ON	Blink	Open	
1	0	1	0	1	0	1	1	**Triggered Stop or Output short**	Alternate		ON	Open	
1	0	1	0	1	0	0	1	**EncoderERR (Reset required)**	Blink	OFF	Blink ON	Open	

INPUTS

Bit	Name	Description
INbit0	StatusBit5	Diagnostic / Limit1 / HomeIN — See HomeSEL1 and HomeSEL2 for details
INbit1	StatusBit6	Diagnostic Limit2 / Input10 / Input11
INbit2	StatusBit3	Power_on diagnostic bit.
INbit3	HomeIN	Home Input. (CN9pin6/CN5pin14). Pin=HIGH, Bit='1'
INbit4	Limit1	Reverse Limit. (CN9pin4/CN5pin7) Pin=HIGH, Bit='1'
INbit5	Limit2	Forward Limit. (CN9pin2/CN5pin15) Pin=HIGH, Bit='1'
INbit6	BridgeSTA	1= Limit Switches BRIDGED.
INbit7	AEN	Amplifier enable input. (CN8pin12) Pin=HIGH, Bit='1' (Mode dependant function)
INbit8	Enable (Enable/Stop)	Hardware Enable/Stop input. (CN3pin4 & CN4pin4) Pin=HIGH, Bit='1' = Drive enable. Pin=LOW, Bit='0' = Hardware stop.
INbit9	Input9	None dedicated input. (CN3pin6/CN4pin6) Pin=HIGH, Bit='1'
INbit10	Input10	None dedicated input. (CN2 pin2) Pin=HIGH, Bit='1'
INbit11	Input11	LDCN mode - None dedicated input. (CN9pin7) Pin=HIGH, Bit='1' Amplifier modes - Hardware Bridge input. (CN9pin7) Pin=HIGH, Bit='1' Limit Switches bridged.
INbit12	DE	Drive enable status bit. LDCN mode = PIC_AE, Other modes = Amplifier Enable state.
INbit13	Reserved.	N.A.
INbit14	Dir	Multifunctional input. (CN8pin7) Pin=HIGH, Bit='1' (Mode dependant function)
INbit15	FAULT	0 = FAULT relay contact closed. 1 = FAULT relay contact open.

OUTPUTS

Bit	Name	Description
OUTbit0	BrakeMODE	0 = Brake/Output1 depends on Servo drive state. See diagnostic tables. 1 = Brake/Output1 is following Output1 bit: Bit='0', Brake/Outpu1(CN1pin4) = OFF Bit='1', Brake/Outpu1(CN1pin4) = HIGH
OUTbit1	Output1	
OUTbit2	Output2 (CN2pin3)	Bit='0', Output2=OFF Bit='1', Output2=HIGH
OUTbit3	N.A.	Reserved – clear to '0'
OUTbit4	HomeSEL1	Select homming input source, Motor and Master encoders latch source.
OUTbit5	Bridge	Limit relay contact
OUTbit6	UserREL	
OUTbit7	SmartSTOP	SmartSTOP 1= Enables the SmartSTOP timer and hardware logic.
OUTbit8	HomeSEL2	Select homming input source, Motor and Master encoders latch source.
OUTbit9	N.A.	Reserved – clear to '0'
OUTbit10	N.A.	Reserved – clear to '0'
OUTbit11	N.A.	Reserved – clear to '0'
OUTbit12	MODEbitA	See Mode tables.
OUTbit13	MODEbitB	
OUTbit14	MODEbitC	
OUTbit15	MODEbitD	
OUTbit16	N.A.	Reserved – clear to '0'
OUTbit17	N.A.	Reserved – clear to '0'
OUTbit18	N.A.	Reserved – clear to '0'
OUTbit19	N.A.	Reserved – clear to '0'

HomeSEL1 / HomeSEL2

HomeSEL2 OUTbit8	HomeSEL1 OUTbit4	StatusBit5 INbit0	StatusBit6 INbit1	Motor encoder latch source	Master encoder latch source
0	0	Limit1	Limit2	Motor Encoder phase Z	Master Encoder phase Z
0	1	HomeIN	Input10		
1	0	HomeIN	Input10	Input10 (change)	
1	1	HomeIN	Input11	Input11 (change)	

Bridge / UserREL

UserREL OUTbit6	Bridge OUTbit5	Input11	LDCN mode Limit Relay	Amplifier mode Limit Relay
0	0	0	Depends on Limit1 and Limit2.	
0	1	0	Closed. BridgeSTA = 1	Depends on Limit1 and Limit2.
0	0	1	Depends on Limit1 and Limit2.	Closed. BridgeSTA = 1
1	0	X	General purpose user relay - Open	
1	1	X	General purpose user relay - Closed	

HomeSEL2

HomeSEL2 OUTbit8	HomeSEL1 OUTbit4	StatusBit5 INbit0	StatusBit6 INbit1	Motor encoder latch source	Master encoder latch source
0	0	Limit1	Limit2	Motor Encoder phase Z	Master Encoder phase Z
0	1	HomeIN	Input10		
1	0	HomeIN	Input10	Input10 (change)	
1	1	HomeIN	Input11	Input11 (change)	

LDCN Single loop mode

MODEbit[C,B,A] = 000

Name	Bit	Function
DE	INbit12	Returns Drive Enable bit (PIC_AE). Bit='1' when PIC_AE bit is SET (Power driver is enabled).
AEN CN8pin12	INbit7	None dedicated input. Pin=HIGH, Bit='1' Routed to Master encoder Latch strobe (CAP6).
Dir CN8pin7	INbit14='0'	Master encoder counter phase A (CAP4)
Step CN8pin13	N.A.	Master encoder counter phase B (CAP5)
$\overline{\text{Fault}}$ CN8pin9	N.A.	SW4=OFF HIGH when PIC_AE bit is CLEARED (Power driver is disabled) or NO fault condition is present. LOW when PIC_AE bit is SET (Power driver is enabled) and fault condition is present. SW4=ON HIGH when PIC_AE bit is SET (Power driver is enabled) and NO fault condition is present. LOW when PIC_AE bit is CLEARED (Power driver is disabled) or fault condition is present. If MODEbitD = 0 Inserted 0.8 µS wide pulse with 51.2 µS period. Used by LS-2311 Master Encoder interface. If MODEbitD = 1 No pulse. Slave drive Enable output.

LDCN Dual loop mode

MODEbit[C,B,A] = 001

Name	Bit	Function
DE	INbit12	Returns Drive Enable bit (PIC_AE). Bit='1' when PIC_AE bit is SET (Power driver is enabled).
AEN CN8pin12	INbit7	LOW = Master encoder error. Routed to Master encoder Latch strobe (CAP6). Master encoder index rising edge is represented with 0.4 µS wide pulse.
Dir CN8pin7	INbit14='0'	Master encoder counter phase A (CAP4)
Step CN8pin13	N.A.	Master encoder counter phase B (CAP5)
$\overline{\text{Fault}}$ CN8pin9	N.A.	SW4=**OFF** HIGH when PIC_AE bit is CLEARED (Power driver is disabled) or NO fault condition is present. LOW when PIC_AE bit is SET (Power driver is enabled) and fault condition is present. SW4=ON HIGH when PIC_AE bit is SET (Power driver is enabled) and NO fault condition is present. LOW when PIC_AE bit is CLEARED (Power driver is disabled) or fault condition is present. Inserted 0.8 µS wide pulse with 51.2 µS period. Used by LS-2311 Master Encoder interface.

Analog input Single/Dual loop mode

MODEbit[C,B,A] = 010

Name	Bit	Function
DE	INbit12	Returns AEN input (CN8pin12). Pin=HIGH, Bit='1'
AEN CN8pin12	INbit7	0 = Amplifier disabled, fault conditions clear. 1 = Amplifier enable.
Dir CN8pin7	INbit14='0'	Master encoder counter phase A (CAP4)
Step CN8pin13	N.A.	Master encoder counter phase B (CAP5)
$\overline{\text{Fault}}$ CN8pin9	N.A.	SW4=**OFF** LOW = fault. HIGH = NO fault. SW4=ON LOW when DE is LOW or fault condition is present. HIGH when DE is HIGH and NO fault condition is present.

Analog input mode with direction invert input

MODEbit[C,B,A] = 011

Name	Bit	
DE	INbit12	
AEN CN8pin12	INbit7	
Dir CN8pin7	INbit14	
Step CN8pin13	N.A.	Routed to CAP6.
$\overline{\text{Fault}}$ CN8pin9	N.A.	SW4=**OFF** LOW = fault. HIGH = NO fault. SW4=ON LOW when DE is LOW or fault condition is present. HIGH when DE is HIGH and NO fault condition is present.

AEN (CN8.12)	Dir (CN8.7)	Function	DE (INbit12)
0	X	Amplifier disabled, fault conditions clear.	0
1	0	Amplifier enable.	1
1	1	Amplifier enable, analog input (ADCin) inverted.	1

Enable Positive/Enable Negative Analog input mode

MODEbit[C,B,A] = 100

Name	Bit	
DE	INbit12	
AEN CN8pin12	INbit7	
Dir CN8pin7	INbit14	
Step CN8pin13	N.A.	Routed to CAP6.
____ Fault CN8pin9	N.A.	SW4=**OFF** LOW = fault. HIGH = NO fault. SW4=ON LOW when DE is LOW or fault condition is present. HIGH when DE is HIGH and NO fault condition is present.

AEN (CN8.12)	Dir (CN8.7)	Function	DE (INbit12)
0	0	Amplifier disabled, fault conditions clear.	0
1	1		
0	1	Amplifier enable, analog input (ADCin) inverted.	1
1	0	Amplifier enable.	1

Quadrature encoder mode

MODEbit[C,B,A] = 101

Name	Bit	Function	
DE	INbit12	Returns AEN input (CN8pin12).	
AEN CN8pin12	INbit7	0 = Amplifier disabled, fault conditions clear. 1 = Amplifier enable. Routed to Master encoder Latch strobe (CAP6).	
Dir CN8pin7	INbit14='0'	Master encoder counter phase A (CAP4)	If MODEbitD = 1 OR Input11 = 'High' Master Encoder is stopped.
Step CN8pin13	N.A.	Master encoder counter phase B (CAP5)	
____ Fault CN8pin9	N.A.	SW4=**OFF** LOW = fault. HIGH = NO fault. SW4=ON LOW when DE is LOW or fault condition is present. HIGH when DE is HIGH and NO fault condition is present.	

Step & Dir mode

MODEbit[C,B,A] = 110

Name	Bit	
DE	INbit12	
AEN CN8pin12	INbit7	
Dir CN8pin7	INbit14='0'	
Step CN8pin13	N.A.	0->1 transition = Step. 1->0 transition = NO change.
____ Fault CN8pin9	N.A.	SW4=**OFF** LOW = fault. HIGH = NO fault. SW4=ON LOW when DE is LOW or fault condition is present. HIGH when DE is HIGH and NO fault condition is present.

AEN (CN8.12)	Dir (CN8.7)	Function	DE (INbit12)
0	X	Amplifier disabled, fault conditions clear.	0
1	0	Amplifier enable, direction positive.	1
1	1	Amplifier enable, direction negative.	1

Step Positive/Step Negative mode

MODEbit[C,B,A] = 111

Name	Bit	Function
DE	INbit12	Returns AEN input (CN8.12).
AEN CN8pin12	INbit7	0 = Amplifier disabled, fault conditions clear. 1 = Amplifier enable.
Dir CN8pin7	INbit14='0'	
Step CN8pin13	N.A.	
____ Fault CN8pin9	N.A.	SW4=**OFF** LOW = fault. HIGH = NO fault. SW4=ON LOW when DE is LOW or fault condition is present. HIGH when DE is HIGH and NO fault condition is present.

Dir (CN8.7)	Step (CN8.13)	Action
0	0	No action
0	0->1 transition	Positive step
1	0->1 transition	No action
0->1 transition	0	Negative step
0->1 transition	1	No action
1	1	No action

All modes can be selected using LDCN (Logosol Distributed Control Network) Utility. Refer to *LOGOSOL LS-231SE QUICK START GUIDE - Mode Selection* section for details.

Power-up and Reset Conditions

After 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 (51.2 µSec servo rate)
The PWM value is set to zero
The controller is 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 Kb/S
In the status byte, the move_done and pos_error flags will be set and home_in_progress flag will be cleared.
In the auxiliary status byte, the pos_wrap, servo_on, accel_done, slew_done and servo_overrun flags will be cleared.

Coordinated motion control – Theory of operation

LS-231SE contains a path point buffer with room for 256 entries. Each entry is a goal position for the motor. When the *Servo Drive* enters its special path mode, it will automatically move from one point to the next at a user selectable rate in steps of 51.2 µSec. The Servo Drive moves the motor between goal points at a constant velocity such that it always arrives at the next path point in exactly the pre-calculated time. When sets of path points are downloaded into multiple controllers, and then the paths started simultaneously, the individual axes will execute their paths with exact* synchronization.
If, for example, the time between the points is set to 5.12 mSec, the path point buffer has room for about 1.3 seconds worth of motion. Typically, the host computer downloads the first part of a path to the LS-231SE buffers and then starts the path mode. As the buffers becomes depleted, additional path points are dynamically added while the axes are still in motion, until the path is complete. The timing requirements for the host require that it be able to dynamically download new path points before the path point buffers empties completely. With a path point buffer size of 1.3 seconds or even more with lower frequency, even a non-real time host, such as a PC running Windows, can easily keep up with the task of re-filling the path point buffers as needed.
The actual multi-axis paths, which are downloaded into the LS-231SE path point buffers, are calculated by the host computer. In addition to creating the geometry desired (arcs, lines, etc.), the path should be smooth, adhering to the physical acceleration and velocity limits of the motors being controlled. Because the host computer actually creates the paths, any path the user can create can be executed, and paths can involve up to 31 axes. Most typically, coordinated straight-line motions, 2-axis circular motions, or S-curve profiling motions are created.
Note that motions created with the path mode are independent of any acceleration or velocity values loaded using the Load Trajectory command.

Path Accuracy

The path accuracy of the LS-231SE Servo Drive is more than adequate for most CNC machine control or robot control applications. For very high speed or very high accuracy applications, however, there are two types of path errors to consider: absolute path errors and timing errors.

Absolute Path Errors

Absolute path accuracy is the accuracy with which a series of calculated path points with straight line segments between them matches the actual curved path desired. For example, a circle, which is approximated by only 5 path points, will form a pentagon rather than a circle. The maximum error between the side of the pentagon and the circle may be quite large. A larger number of path points will

* *The exactness of the synchronization is subject to crystal frequency accuracy and other timing factors discussed later.*

produce a smaller error. In general, accuracy of an approximated path will be a function of the number of path points used, and the radius of the curve.
Because LS-231SE uses a fixed number of points per second, moving more slowly will result in a more accurate path than moving quickly. Also, a higher frequency path will be more accurate than a lower frequency path. The main advantages of using a slower path, however, are that fewer path points need to be calculated, less data needs to be sent to the controllers, and the path point buffer will last longer.
The maximum absolute path error can be approximated by the formula:

Error = R x (1 - cos(V / (2xFxR)))*

where **R** is the radius of the curve (in inches), **V** is the velocity of the motion (in inches/sec), and **F** is the path point frequency. For example, a one-inch diameter circle with a velocity of 1 inch per second and a path frequency of 30 Hz would have a maximum error of 0.00028 inches.

Timing Errors

If the timing of multiple axes is not perfectly synchronized, there will be a deviation from the desired path from the fact that one axis will be ahead or behind in time. The exact deviation will depend on the path geometry.

The first type of timing error results from multiple axes not starting at exactly the same time. When a "start path" command is issued to a group of controllers, they will all start within +/- 0.000025 seconds of one another.

The second type of timing error results from inaccuracies in the frequencies of the oscillators running on each *LS-231SE* controller. (If all Servo Drives are timed from the same oscillator, this error is zero.) Typical oscillator variations (for the same operating temperature) are about 10 parts per million. Therefore, after running a path for 10 seconds, for example, the timing error would be about +/-0.0001 seconds.
By adding both of these timing errors together, and then multiplying by the path velocity, we get the total distance that one axis can be ahead of another axis. For a 10 second motion, while moving at 1 inch per second, we could have one axis moving ahead of another by at most 0.000125 inches. The actual worst-case deviation (moving along a 45 degree angle) will produce an error from the ideal path of 0.000125 inches. Over a total distance of 10 inches traveled, this gives a basic accuracy of ±0.0000125 inches per inch of travel. Other examples, of course, will produce different accuracy figures.
Note that errors due to timing only accumulate during a coordinated motion and are, in essence, reset with each new move. Therefore, if errors due to timing do become a problem, the paths should be broken up into shorter moves.

Eliminating those errors is possible by enabling the hardware synchronization mode using *Enable / Disable hardware synchronization mode* command.

* *The cosine function should be executed for an angle in radians.*

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	1-2	Defines which data should be sent in every status packet	Yes
Read status	0x3	1-2	Causes particular status data to be returned just once	Yes
Load trajectory	0x4	1-15	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	Stops the motor in one of three manners	Yes
I/O control	0x8	1	Controls the “Brake out” mode and sets the path point mode frequency	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
Add path points	0xD	0-n	Adds up to 7 path points to the device buffer and starts the path point mode motion	Yes
Nop	0xE	0	Simply causes the defined status data to be returned	Yes
Extended commands	0xE	1-n	Extended commands (n > 0)	No
Hard reset	0xF	0	Resets the controller t o 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.

Command Description

Reset Position

Command value: *0x0*
Number of data bytes: *0*
Command byte: ***0x00***
Data bytes:
None

Description:
Resets the 32-bit encoder counter to zero. 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)
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 module 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 default to being a group member.) 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 or 2*
Command byte: ***0x12 or 0x22***
Data bytes:
1. Status items: (default: 0x0000)

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)* *(First byte - Motor controller device ID = 0,* *Second byte - version number =20 to 29 (decimal))*
	6:	*send current position error (2 bytes)*
	7:	*send number of points in the path buffer (1 byte)*
	8:	*send digital inputs (2 bytes)*
	9:	*send analog inputs (2 bytes)*
	10, 11:	*reserved. Clear to 0*
	12:	*send watchdog status (2 bytes)*
	13:	*send motor position and position error (6 bytes)*
	14, 15:	*reserved. Clear to 0*

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 aux. 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.

Read Status

Command value: *0x3*
Number of data bytes: *1 or 2*
Command byte: ***0x13 or 0x23***
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)*
(First byte - Motor controller device ID = 0,
Second byte - version number =20 to 29 (decimal))
6: *send current position error (2 bytes)*
7: *send number of points in the path buffer (1 byte)*
8: *send digital inputs (2 bytes)*
9: *send analog inputs (2 bytes)*
10, 11: *reserved. Clear to 0*
12: *send watchdog status (2 bytes)*
13: *send motor position and position error (6 bytes)*
14, 15: *reserved. Clear to 0*

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.

Load Trajectory

Command value: *0x4*
Number of data bytes: *n = 1-15*
Command byte: ***0xn5***
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 or 2 bytes)**
4: *servo mode - 0 = PWM mode, 1 = position servo*
5: *profile mode - 0 = trapezoidal profile, 1 = velocity profile*
6: *in velocity/PWM mode - direction flag 0 = FWD, 1 = REV*
7: *start motion now*

PWM value can be 1 or 2 bytes. If only 1 byte is supplied, the most significant byte is set to 0.
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

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

value (range 0x0000 - 0xFFFF), used only when the position servo is not operating, sends a raw PWM value directly to the amplifier. 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 if the PWM mode should be entered. Bit 5 specifies whether a trapezoidal profile motion should be initiated or if 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 has been completed, or in velocity mode, when the command velocity has been reached.) Bit 6 indicates the velocity or PWM direction. 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 load only 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.

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 Drives with trajectory information and then starting them simultaneously with a group command.

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. Not used
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 $0<PWM\leq OL$ in position servo modes. In PWM mode OL is ignored. CL is used for motor current limitation (refer to *Motor current monitoring* in *Safety Features* for detailed information). 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 51.2 uSec (19.531) KHz). For example SR=3 gives a servo rate of 6510 Hz. The servo tick rate is also used as the profiling time base, although command processing is always performed at the maximum tick rate.

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: Not used. Clear to 0
6: Not used. Clear to 0
7: Not used. Clear 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 (Limit1 (Reverse)), and bit 6 (Limit 2 (Forward)) of status byte (refer to *Status Bits* and *Safety Features* section 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 are 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*
Command byte: ***0x18***
Data bytes:
Control byte
Bit 0: "Brake out" mode.
If this bit is set to 0 the "Brake out" is controlled according the *"Status bits and LEDs"* section of this document. If this bit is set to 1 the "Brake out" is controlled by *BIT1.*
1: Brake out control.
If *BIT0* is set to 0 - not used.
If *BIT0* is set to 1: *BIT1*=0 - Brake out=off;
BIT1=1 - Brake out =on.
2: Not used. Must be set to 0.
3: Not used. Must be set to 0.
4: Not used. Must be set to 0.
5: Not used. Must be set to 0.
6: If this bit is set, the Path Points Buffer counter will be set to the value of the next 2 data bytes (requires 2 data bytes, range 0000h – 7FFFh)
7: Not used. Set to 0.
Description:

Depending on *BIT0 "Brake out"* can be controlled by device status (Refer to *"Status bits and LEDs"* section of this document) or by *BIT1* (Refer to *"Brake out"* section of this document).
This command with bit 6 set to 1 is used to set the time base for the path points. The path point buffer counter multiplied by the servo tick (51.2 uSec) gives the time between the points. For example if the path point buffer counter is set to 100, the time between the points will be 5.12 mSec (100 x 51.2 uSec).

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 Limit 1 (Reverse)
1: Capture home position on change of Limit 2 (Forward)
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 then 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.

Set Baud Rate

Command value: *0xA*
Number of data bytes: *1*
Command byte: ***0x1A***
Data bytes:
1. Baud rate divisor, *BRD*

sample values:

9600	BRD = 0x81
19200	BRD = 0x3F
57600	BRD = 0x14
115200	BRD = 0x0A
125000	BRD = 0x27
312500	BRD = 0x0F
625000	BRD = 0x07
1250000	BRD = 0x03

Description:
Sets the communication baud rate. All 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.

Add path points
Command value: *0xD*
Number of data bytes: *n = 0, 2, 4, 6, 8, 10, 12 or 14*
Command byte: ***0xnD***
Data bytes:
1, 2: *Incremental data for path point 1 (n >= 2)*
3, 4: *Incremental data for path point 2 (n >= 4)*
…
13, 14: *Incremental data for path point 7 (n = 0xE) or*
None Starts execution of path point mode (n = 0)
Description:
The data format of the points is a 2-byte signed value. The most significant byte is the integer part and the least significant byte is fractional part (1/256 of encoder count). The whole value is added to the desired position each servo tick. The same value is applied Path Points Buffer counter times, thus forming the desired frequency between the path points.

No Operation
Command value: *0xE*
Number of data bytes: *0*
Command byte: ***0x0E***
Data bytes: None

Description:
The No Operation command does nothing except cause a status packet with the currently defined status data to be returned.

Extended commands
Command value: *0xE*
Number of data bytes: *1 to n*
Command byte: ***0x1E*** *to* ***0xnE***
Data bytes:
1: sub command code
2 to n: sub-command specific data

Sub-command ***0x00*** *Stop on limit switches.*
Data bytes:
1: control byte for limit 1 (Reverse)
2: control bite for limit 2 (Forward)

Limit switch control bytes bit description:
Bit 0 servo in one direction
1 turn motor off
2 stop abruptly
3 stop smoothly
4 to 7 not used

Description
Setting one of bits 0 to 3 enables Stop on limits function and specifies how to stop the motor. If bit 0 is set, the position servo will be enabled in the direction of the limit only. If bit 1 is set, the position servo will be disabled, the PWM output value will be set to 0, and bits 2 and 3 are ignored. If bit 2 is set, the motor will simply start servoing to its current position. Setting bit 3 enters a more graceful stop mode – the controller sets the goal velocity to 0 and enters velocity mode, causing the motor to decelerate to a stop at the current acceleration rate.
Clearing bits 0 to 3 disables Stop on limits function.
This function is disabled by default.

Sub-command ***0x01*** *Read hall sensors and initialize the angle.*

Description
This command makes the controller to read the hall sensors state and to calculate the initial angle. This angle will be overwritten when the first index comes.

Sub-command ***0x02*** *repeat the last answer.*

Description
This command makes the controller to send the last sent answer again.

Sub-command ***0x04*** *Enable / Disable hardware synchronization mode.*
Data byte 1 0 – Disable; 1 – Enable.

Description
This command enables / disables hardware synchronization mode. When enabled, several LS-231SE drives connected to each other synchronize their servo ticks. This eliminates any differences in the actual velocity, which otherwise can be caused by the slightly different oscillators.

Sub-command ***0x05*** *Set watchdog mode mode.*

Data byte 1 *mode: 0 – watchdog off, 1 – disable the amplifier, 2 – stop smoothly and disable the amplifier, 3 – stop smoothly.*
Data byte 2 *watchdog timeout in multiples of 8192 microseconds.*

Description
This command sets the watchdog mode and timeout. The watchdog is refreshed by any command sent to the driver. Upon expiration, the watchdog can disable the amplifier, stop smoothly and disable the amplifier, or stop smoothly and keep the amplifier enabled. After the watchdog expires, the drive stops executing any motion commands. This command should be sent again to reset the watchdog.
The watchdog status can be obtained using Read Status or Define Status commands with bit 12 set to 1. If the watchdog is not activated, the status will be 65535 (0xFFFF). If the watchdog has expired, the status will be 0. Any other value means that the watchdog is working and represents the remaining time before the watchdog expires in multiples of 8192 microseconds.

*Sub-command **0x10*** *Set motor error limit.*
Data bytes 1 and 2 *motor error limit.*

Description
In dual loop mode, after power up, the motor error limit is set to the same value as the master error limit. Set gain commands also sets the motor error limit to the same value as the master error limit. This command can be used to set the motor error limit.

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.
Hard reset command sent at address 0xFF resets the controller even if its group address is different than 0xFF.

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 just received command packet
2	Current_limit	Set if current limiting has exceeded. Must be cleared by user with *Clear Sticky Bits* command
3	Power_on/diag. bit	Refer to *LS-231SE Diagnostic and I/O* section in this document
4	Pos_error	Set if the position error has exceeded the position error limit. It is also set whenever the position servo is disabled (Power_on=0). Must be cleared by user with *Clear Sticky Bits* command
5	Home source/ diag. bit	Home source or diagnostic bit (refer *LS-231SE Diagnostic and I/O* section in this document).
6	Limit2/diag. bit	Forward Limit or diagnostic bit (refer to *LS-231SE Diagnostic and I/O* 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 *LS-231SE Diagnostic and I/O* 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 completed. 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 51.2 uSec, 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	Path mode	Set when the drive is currently executing a path. Cleared when buffer is emptied or Stop Motor or Load Trajectory command is sent.

INITIALIZING PROCEDURE AND PROGRAMMING EXAMPLES FOR SERVO DRIVES

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

1. Reset all modules with *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 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).

Understanding the Serial Communication with Servo drives

The Serial Communication with Servo drives is strictly master-slave and matches repeatedly two elements:

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

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

Commands

There are 16 commands managing Servo drives (refer to 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 FF	21
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 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 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

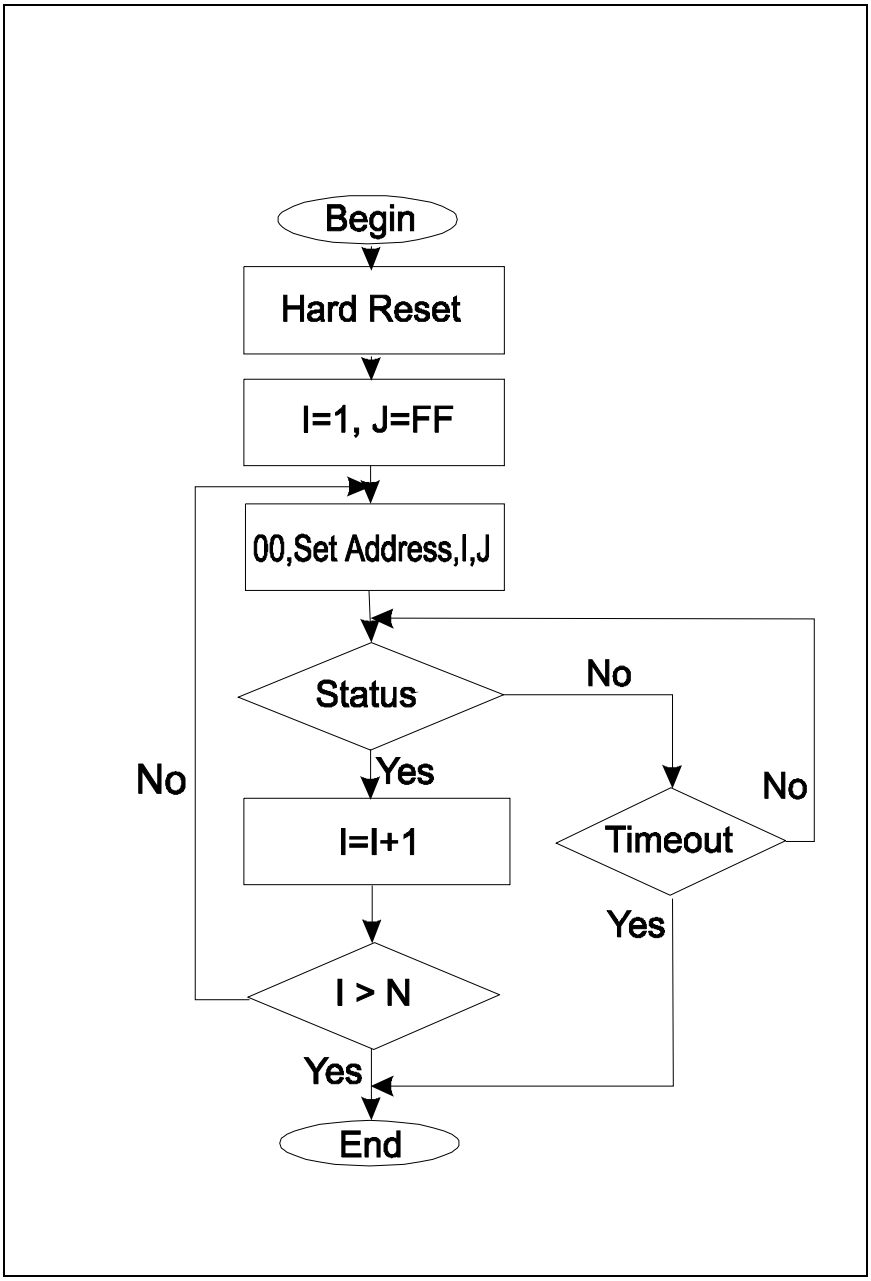
After power-up and *Hard Reset* command all drives have their address set to 00h and only the first drive (starting from the host) has its communication enabled. Consecutive *Set Address* commands are sent to address 00h until all drives are addressed. This procedure can be executed once after *Hard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

Example of sequential addressing for three Servo 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 *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 cycle.

Examples of Managing Two Servo Drives

1 – Resets all modules with group command.
2 and # 3 - Set the addresses of drives 1 and 2.
4 and # 6 - 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, KD=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, KD=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 until no response received
AA 01 13 20 34	Read Device ID and Version number
AA 01 13 FF 13	Read all status data
AA 01 E6 64 00 00 04 00 00 00 00 FF 00 00 08 01 00 57	Set Gain parameters
AA 01 E4 9F 00 00 00 00 00 00 00 00 01 00 00 00 00 85	Set Trajectory parameters
AA 01 17 05 1D	Close servo loop

Procedure FindHomePosition

AA 01 E6 C8 00 20 03 46 00 28 00 FF 00 40 1F 01 00 9F	Set 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	Close the servo loop (Stop smoothly and amplifier enable)
AA 01 94 36 25 06 01 00 22 00 00 00 19	Load trajectory: Velocity mode, Forward direction, Velocity=1 revolution per second (67109 programmed velocity for 5000 line encoder), Acceleration = 10 revolutions per second^2 (34 programmed acceleration for 5000 line encoder)
AA 01 19 12 2C	Set home mode - capture home position on change of Limit 1 and stop abruptly
AA 01 05 06	Start motion
wait while home_in_progress bit=1	Home position is found on change of Limit 2
AA 01 19 18 32	Set 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	Load trajectory: Velocity mode, Reverse direction
AA 01 05 06	Start 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 revolutions per second) x 3.3554432
Acc = (encoder counts per revolution) x (number of revolutions per second^2) x 0.00017179869

For this example (5000 lines encoder – 20000 encoder counts per revolution):

Vel	= 20000 x 1	x 3.3554432	= 67109	= 00010625h
Acc	= 20000 x 10	x 0.00017179869	= 34	= 00000022h