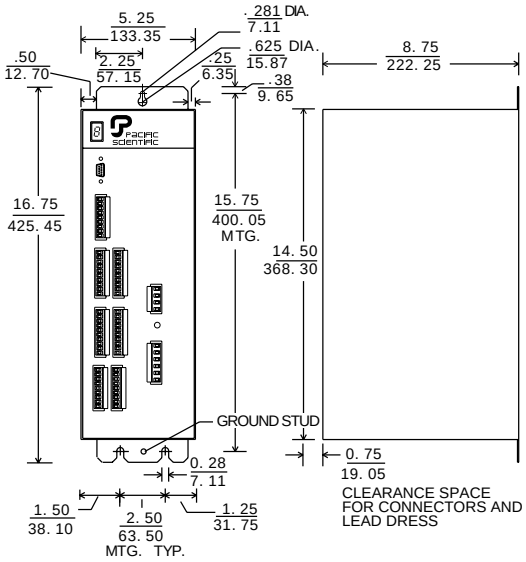
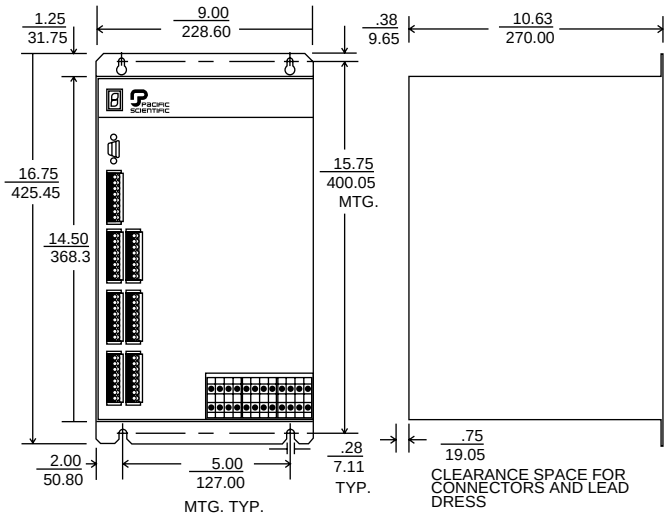
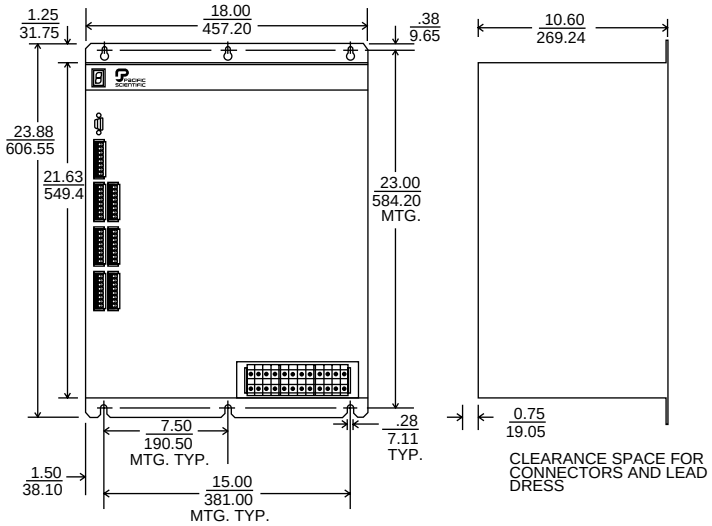
DIMENSIONS*



SC752/SC753



SC754/SC755



SC756

**See pages B-6 through B-26 for information on the next generation Servo Position Control.*

GENERAL . . .SC750 Series Controllers*

Efficiency (50% output voltage, rated cont. current)	>95%
Form factor	<1.01
Current loop bandwidth	1500 to 3000Hz
Input command	RS-232 or RS-485/422, PacLAN port
Output ripple frequency (±15%)	20 kHz
Position resolution	65,536 steps/rev (16 bit)
Position accuracy (drive only)	
Standard	± 22 arcmin
Optional	± 4 arcmin
Velocity range (drive only)	0 to 12,000 RPM
Analog input	±12 Vdc range, 16 bit @200 Hz bandwidth or 14 bit @1000 Hz bandwidth, 0.25% linearity, 50μV/°C drift, fully monotonic, 60 mV offset, 100 Kohm input impedance
Analog output	±5 Vdc range, 12 bit, updated every 0.4 mS, fully monotonic, 50 ohm source impedance
Control inputs	Enable Fault reset Inhibit + Inhibit -
Control outputs	Fault Current monitor Commutation signal Two +5V @250 mA max.
Programmable inputs	12 general purpose 2 general purpose/fast 2 general purpose/counter Optically isolated, 5 to 24 Vdc logic, sourcing; fast inputs have 5μS max. capture time
Programmable outputs	11 general purpose 1 general purpose/PWM 5 to 24 Vdc logic, open collector sinking, up to 100 mA
Encoder Emulation	
Modes	Encoder output Encoder input Step/direction input
Encoder output mode	Quadrature with marker pulse, differential TTL line driver, software selectable resolution—500, 512, 1000, 1024, 2000, 2048, 4096, 16,384 PPR, 750 kHz max.
Encoder input mode	Quadrature with marker pulse, differential TTL line receivers, 750 kHz max.
Step/direction mode	Differential TTL line receivers, 750 kHz max.
Mating connectors (furnished)	All plug-in mating connectors
Storage temperature	-55°C to 70°C
Operating temperature	
Full ratings	0°C to 50°C
Derated⚠	50°C to 60°C
Altitude	5000 ft. (1500 m)
Humidity	10% to 90%, non-condensing
Weight	SC752, 13 lbs. SC753, 16 lbs. SC754/755, 40 lbs SC756, 90 lbs.

⚠ Linearly derate the continuous current and power ratings to 70% at 60°C.

**See pages B-6 through B-26 for information on the next generation Servo Position Control.*

POWER DATA. . .SC750 Series Controllers*

	SC752	SC753	SC754	SC755	SC756
Input voltage					
Control logic power	90 to 264 Vac, 47 to 63 Hz, 1 phase				
Bus power	120 Vac (+ 10%, -15%)/ 240 Vac (+ 10%, -15%), 47 to 63 Hz, 1 or 3 phase		240 Vac (+ 10%, -15%), 47 to 63 Hz, 3 phase		
Bus Voltage					
(with 240 Vac input)	320 Vdc				
(with 120 Vac input)	160 Vdc				
Input current[3]					
Control logic power	500 mA max. @ 115 Vac, 250 mA max. @ 230 Vac.				
Bus current (RMS) (230 Vac, 3-phase)	4.5 A_{rms}	9 A_{rms}	18 A_{rms}	29 A_{rms} (@6kW)	38 A_{rms} (@ 12kW)
Output current @ 50°C[1]					
Peak (5 seconds)	7.5 A	15 A	30 A	60 A	120 A
Continuous (stall)	3.8 A	7.5 A	15 A	30 A[2]	60 A
Output power (min. @ 50°C)					
Peak (5 seconds)					
230 Vac, 3-phase	2.2 kW	4.5 kW	9 kW	18 kW	36 kW
230 Vac, 1-phase	2.0 kW	4.0 kW	n/a	n/a	n/a
Continuous					
230 Vac, 3-phase	1.1 kW	2.2 kW	4.5 kW	9 kW	18 kW
230 Vac, 1-phase	0.8 kW	1.6 kW	n/a	n/a	n/a
Shunt regulator power					
Peak	3 kW	6 kW	20 kW	20 kW	40 kW
Continuous	20 W	40 W	200 W	200 W	500 W

[1] Currents shown are the peak of a sinewave. Multiply by 0.707 to get RMS current value.

[2] The SC755 can be configured for 40 A continuous (stall).

[3] Input current is specified with unit operating at continuous rated output power.

**See pages B-6 through B-26 for information on the next generation Servo Position Control.*

SC750 SERIES RECOMMENDED MOTOR/CONTROLLER SYSTEMS[6]

Peak stall torque T_{PS} [1] lb-in./Nm	Peak rated torque T_{PR} [2] lb-in./Nm	Cont. stall torque T_{CS} lb-in./Nm	Cont. rated torque T_{CR} lb-in./Nm	Rated Speed W_R [3] rpm	Inertia x 10^{-3} [4] lb-in-S^2/ kgm² J	Servo-motor Dia. or Width/height inch/mm	Servo-motor Length inch/mm	Inductance line-line L mH	Servo-motor model	Servo-Controller model [5]
With R Series (low inertia, rare earth magnet) motors										
10/ 1.1	6.5/ .7	4.8/ .5	2.5/ .3	8000	0.12/ .014	2.00/ 50.8	7.38/187.5	5.3	R22HENA-R1-NS-NV-01	SC752A-001-01
14/ 1.6	8/ .9	7.6/ .9	5.2/ .6	8000	0.15/ .017	2.00/ 50.8	8.38/212.9	6.5	R23HENA-R1-NS-NV-01	SC752A-001-01
17/ 1.9	10/ 1.1	9.2/ 1.0	7.1/ .8	6000	0.18/ .020	2.00/ 50.8	9.38/238.3	6.5	R24HENA-R1-NS-NV-01	SC752A-001-01
27/ 3.0	25/ 2.8	14/ 1.6	10/ 1.1	4000	0.55/ .062	3.25/ 82.6	7.09/180.1	23.0	R32GENC-R2-NS-NV-00	SC752A-001-01
27/ 3.0	25/ 2.8	14/ 1.6	6/ .7	7000	0.55/ .062	3.25/ 82.6	7.09/180.1	5.8	R32HENC-R2-NS-NV-00	SC753A-001-01
36/ 4.0	33/ 3.7	21/ 2.4	16/ 1.8	3000	0.74/ .084	3.25/ 82.6	8.09/205.5	22.0	R33GENC-R2-NS-NV-00	SC752A-001-01
36/ 4.0	25/ 2.8	21/ 2.4	10/ 1.1	6000	0.74/ .084	3.25/ 82.6	8.09/205.5	5.6	R33HENC-R2-NS-NV-00	SC753A-001-01
45/ 5.0	42/ 4.7	27/ 3.0	24/ 2.7	2800	0.92/ .104	3.25/ 82.6	9.09/230.9	30.0	R34JENC-R2-NS-NV-00	SC752A-001-01
45/ 5.0	38/ 4.3	27/ 3.0	16/ 1.8	4000	0.92/ .104	3.25/ 82.6	9.09/230.9	7.5	R34KENC-R2-NS-NV-00	SC753A-001-01
54/ 6.0	53/ 5.9	32/ 3.6	30/ 3.4	2000	1.11/ .125	3.25/ 82.6	10.09/256.3	42.0	R35JENC-R2-NS-NV-00	SC752A-001-01
54/ 6.0	48/ 5.4	32/ 3.6	23/ 2.6	3000	1.11/ .125	3.25/ 82.6	10.09/256.3	10.5	R35KENC-R2-NS-NV-00	SC753A-001-01
66/ 7.4	50/ 5.6	33/ 3.7	33/ 3.4	1800	1.90/ .215	4.25/108.0	8.34/211.8	53.0	R43GENA-R2-NS-NV-00	SC752A-001-01
66/ 7.4	60/ 6.7	33/ 3.7	31/ 3.5	3000	1.90/ .215	4.25/108.0	8.34/211.8	13.3	R43HENA-R2-NS-NV-00	SC753A-001-01
106/11.9	95/10.6	48/ 5.4	46/ 5.2	2000	2.70/ .305	4.25/108.0	9.84/249.9	4.9	R45GENA-R2-NS-NV-00	SC753A-001-01
106/11.9	95/10.6	48/ 5.4	43/ 4.8	4000	2.70/ .305	4.25/108.0	9.84/249.9	20.0	R45HENA-R2-NS-NV-00	SC754A-001-01
142/15.9	120/13.4	64/ 7.2	61/ 6.8	1500	3.50/ .395	4.25/108.0	11.34/288.1	25.0	R46GENA-R2-NS-NV-00	SC753A-001-01
142/15.9	120/13.4	64/ 7.2	50/ 5.6	3000	3.50/ .395	4.25/108.0	11.34/288.1	6.2	R46HENA-R2-NS-NV-00	SC754A-001-01
150/16.8	125/14.0	70/ 7.8	58/ 6.5	3000	7.10/ .818	5.75/146.1	9.36/237.7	8.9	R63GENA-R2-NS-NV-00	SC754A-001-01
150/16.8	142/15.9	70/ 7.8	38/ 4.3	6000	7.10/ .818	5.75/146.1	9.36/237.7	2.2	R63HENA-R2-NS-NV-00	SC755A-001-01
240/26.9	230/25.8	115/12.9	102/11.4	1700	11.10/1.28	5.75/146.1	11.36/288.5	13.7	R65GENA-R2-NS-NV-00	SC754A-001-01
240/26.9	200/22.4	115/12.9	90/10.1	3000	11.10/1.28	5.75/146.1	11.36/288.5	3.4	R65HENA-R2-NS-NV-00	SC755A-001-01
340/38.1	305/34.2	168/18.8	156/17.5	1000	15.10/1.74	5.75/146.1	13.36/339.3	18.2	R67GENA-R2-NS-NV-00	SC754A-001-01
340/38.1	330/37.0	168/18.8	146/16.4	2000	15.10/1.74	5.75/146.1	13.36/339.3	4.6	R67HENA-R2-NS-NV-00	SC755A-001-01
300/33.6	280/31.4	190/21.3	125/14.0	3800	39.10/4.50	7.50/190.5	10.93/277.6	3.2	R84GENA-R2-NS-NV-00	SC755A-001-01
300/33.6	270/30.2	190/21.3	45/ 5.0	6000	39.10/4.50	7.50/190.5	10.93/277.6	0.8	R84HENA-R2-NS-NV-00	SC756A-001-01
410/45.9	395/44.2	276/30.1	225/25.2	2000	58.10/6.69	7.50/190.5	12.93/328.4	3.6	R86GENA-R2-NS-NV-00	SC755A-001-01
410/45.9	365/40.9	276/30.1	120/13.4	4000	58.10/6.69	7.50/190.5	12.93/328.4	0.9	R86HENA-R2-NS-NV-00	SC756A-001-01
540/60.5	525/58.8	357/28.8	325/36.4	1500	76.10/8.77	7.50/190.5	14.93/379.2	4.0	R88GENA-R2-NS-NV-00	SC755A-001-01
540/60.5	500/56.0	357/28.8	205/23.0	3000	76.10/8.77	7.50/190.5	14.93/379.2	1.0	R88HENA-R2-NS-NV-00	SC756A-001-01
800/89.6	730/81.8	451/50.5	424/47.5	1200	95.10/10.96	7.50/190.5	16.93/430.0	7.4	R8AGENA-R2-NS-NV-00	SC755A-001-01
800/89.6	720/80.6	451/50.5	300/33.6	2000	95.10/10.96	7.50/190.5	16.93/430.0	1.9	R8AHENA-R2-NS-NV-00	SC756A-001-01
With F Series (medium inertia, ferrite magnet) motors [6]										
56/ 6.3	34/ 3.8	32/ 3.6	25/ 2.8	1500	9.20/1.04	4.25/108.0	8.34/211.8	49.0	F43GENA-R2-NS-NV-00	SC752A-001-01
60/ 6.7	30/ 3.4	32/ 3.6	22/ 2.5	3000	9.20/1.04	4.25/108.0	8.34/211.8	12.0	F43HENA-R2-NS-NV-00	SC753A-001-01
84/ 9.4	55/ 6.2	46/ 5.2	43/ 4.8	1000	13.70/1.55	4.25/108.0	9.84/249.9	69.0	F45FENA-R2-NS-NV-00	SC752A-001-01
84/ 9.4	45/ 5.0	46/ 5.2	38/ 4.3	2000	13.70/1.55	4.25/108.0	9.84/249.9	17.0	F45GENA-R2-NS-NV-00	SC753A-001-01
121/13.6	62/ 6.9	61/ 6.8	48/ 5.4	1500	17.70/2.00	4.25/108.0	11.34/288.1	24.0	F46GENA-R2-NS-NV-00	SC753A-001-01
121/13.6	60/ 6.7	61/ 6.8	43/ 4.8	3000	17.70/2.00	4.25/108.0	11.34/288.1	6.0	F46HENA-R2-NS-NV-00	SC754A-001-01

[1] See page B-73 for definitions of ratings.

[2] Peak torque ratings are for 5 seconds.

[3] Rated speeds are for 230 Vac, 3 phase operation.
Derate to approximately 85% for 240 Vac, 1 phase operation.
Derate to 40% for 115 Vac, 1 phase operation.

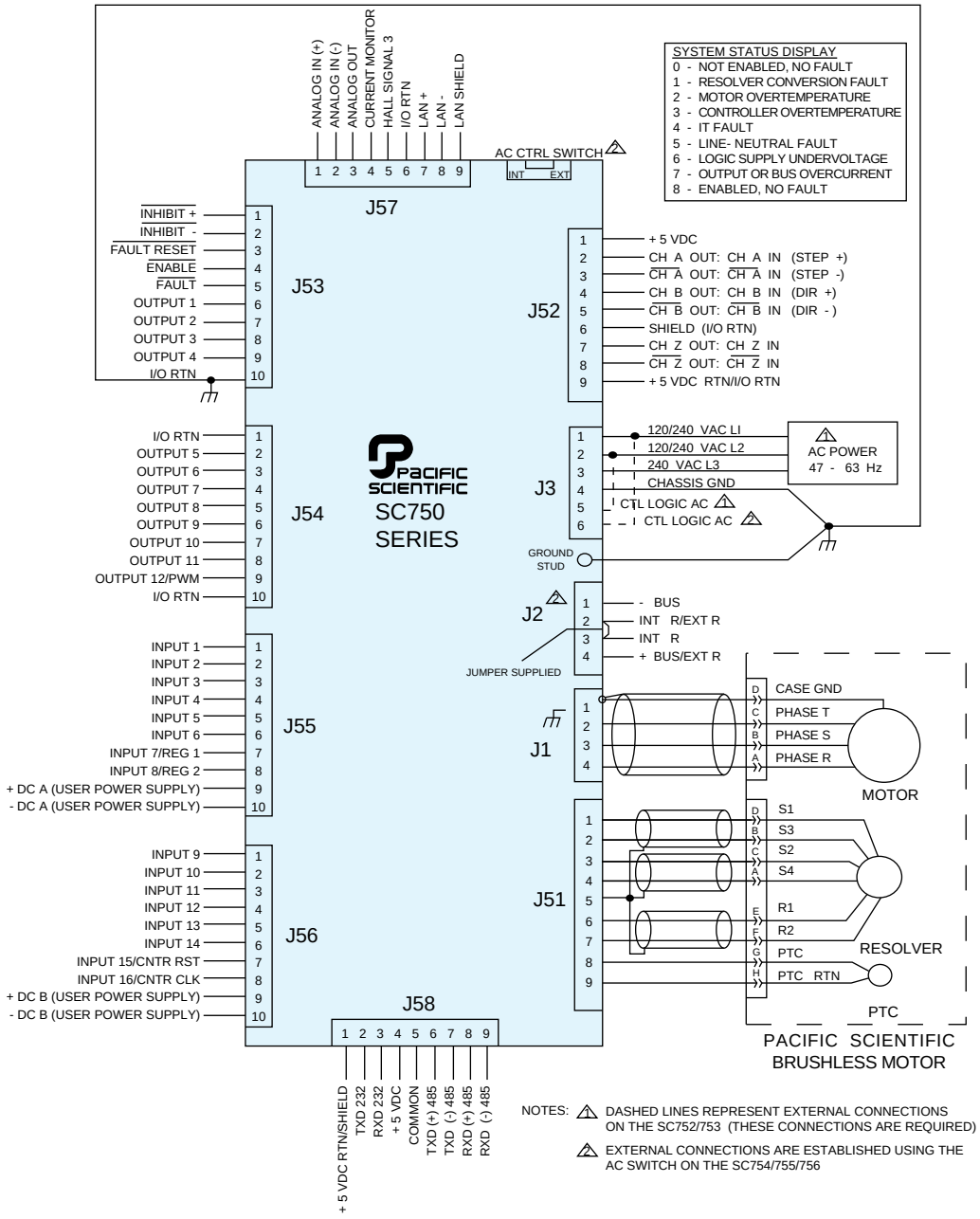
[4] Includes primary feedback inertia.

[5] Controller model numbers are for 12 bit resolution option.
See "How to Order" on page B-34.

[6] Each system requires one feedback and one motor power cable.

**See pages B-6 through B-26 for information on the next generation Servo Position Control.*

CONNECTION DIAGRAM . . . SC750 Series Controllers*



**See pages B-6 through B-26 for information on the next generation Servo Position Control.*

HOW TO ORDER . . . SC750 Series Controllers*

To order an SC750 Series brushless servo motor controller, designate the power level and position accuracy desired. The next three digits are a controller customization code, which is factory assigned. It is also used to specify any customer requested modifications. The final two digits specify the desired software option.

For a standard SC750 Series controller with a power rating of 15A cont., 30A peak, the following model number would be ordered:

MODEL NUMBER CODE

SC 7 5 4 A-001-01

Servo controller

Series

Function code
5=position controller (resolver)

Power level code
2=3.8A cont./7.5A peak
3=7.5A cont./15A peak
4=15A cont./30A peak
5=30A cont./60A peak
6=60A cont./120A peak

Option code
A=12 bit RDC (±22 arcmin, 1024 ppr)
B=14 bit RDC (±4 arcmin, 4096 ppr)

Customization code
Factory assigned, 001=standard unit

Suffix code
-01=PacSci ServoBASIC *Plus*™ programming language with 24K user memory
-02=PacSci ServoBASIC *Plus*™ programming language with 100K user memory and faster processor

HOW TO ORDER . . .SC750 Series Recommended Motor/Controller Systems

See the Recommended Motor/Controller Systems table on page B-32 for performance information and model numbers for servo motor/controller combinations. Order motors and controllers as separate part numbers. See brushless servo motor section for additional motor specifications and information.

**See pages B-6 through B-26 for information on the next generation Servo Position Control.*

PACIFIC SCIENTIFIC'S ServoBASIC *Plus*™

Pacific Scientific developed ServoBASIC™ as a subset of the popular GWBASIC by adding motion control commands.. With the ServoBASIC™ language, intelligent motion control became accessible to users with little or no programming experience.

First introduced in 1987, ServoBASIC™ has helped engineers do sophisticated motion and machine control in thousands of applications. The second generation ServoBASIC Plus™ was introduced with the SC750 family of drives. The SC750 ServoBASIC Plus™ was the logical progression of the original breakthrough, enhanced through field experience and faster microprocessors. The third generation of ServoBASIC Plus™ fits on the OC950 option card for the SC900 family of all digital servo drives. Each generation of ServoBASIC™ has expanded flexibility and functionality, increased execution speed, and simplified programming. Throughout each generation we have retained the familiar syntax and program flow of ServoBASIC, so you don't have to learn a new language to access the enhanced power and flexibility.

Advantages of the OC950 third generation ServoBASIC Plus™

- 5 times faster program execution than previous generation• 27 fully programable bidirectional digital inputs/outputs• 8 programmable position limit switches
- PacLAN™ inter drive communications is optional• Fully integrated Windows program development environment
- Enhanced program modularity:
  - Arguments to subroutines
  - Local variables
  - User defined functions
  - Fast multi-way branches
  - Multi dimensional array variables
- Full interactive debugger

ABORT.MOTION
Stops motor motion while allowing continued program execution.

ABS
Converts the associated value (x) to an absolute value.

ACCEL.GEAR
Sets the commanded acceleration rate when gearing is turned ON. The specified acceleration rate is used until GEARLOCK is achieved. ACCEL.GEAR is independent of DECEL.GEAR.

ACCEL.RATE
Sets the maximum commanded acceleration rate when speed is increased.

ACCEL.TYPE
Determines the use of constant acceleration or S-Curve velocity profiles.

AD.OFFSET
Specifies the level of a signal summed with the digitized value of the analog input channel, in volts.

ADF0
Sets the analog input channel's filter break frequency.

ANALOG.IN
Contains the digitized value of the analog input channel, in volts, of J57-1 relative to J57-2.

ANALOG.OUT
Sets the voltage level of the analog output channel.

ARF0
The first anti-resonant filter break frequency.

ARF1
The second anti-resonant filter break frequency.

ASC( )
Returns a numeric value that is the ASCII code for the first character of the string expression.

ATAN
Returns the arctangent of x in radians.

AUTOSTART
Specifies the automatic execution of a user program as soon as the servo controller has AC power applied.

AXIS.ADDR
Indicates the RS-485 or multidrop address set by switch S1 on the controller.

AXIS.INTR
Indicates the axis address number of the source address of a LANINTERRUPT.

BEEP
Transmits a speaker beep command to the serial port.

BLKTYPE
Specifies configuration as a position, velocity or torque block.

CALL
Transfers program execution to a BASIC subroutine.

CCWINH
Indicates the current state of the CCWINH (INH-) input.

CCWOT
Sets the counter-clockwise overtravel limit.

CHR$( )
Converts an ASCII code to its equivalent character.

CINT( )
Converts x to an integer by rounding the fractional portion.

CLS
Clears the screen display of a terminal.

CMDGAIN
Controls the scale factor of the analog input signal to the servo loops for BLKTYPE=0, 1, or 3.

CONST
Declares numeric constants to be used in place of numeric values.

COS( )
Returns the cosine of its argument which must be in radians.

COUNTER
Specifies the current count of the hardware event counter feature using discrete input 16.

COUNTSPERREV
Specifies the resolution of the position control.

CWINH
Indicates the current state of the CWINH (INH+) input.

CWOT
Sets the clockwise overtravel limit.

DACMAP
Specifies the signal sent to the monitor DAC driving the analog output channel.

DACMON
Contains the value of the selected, filtered variable output to the analog output channel.

DECEL.GEAR
Sets the deceleration rate commanded when gearing is turned OFF. The specified deceleration rate is used until geared motion has stopped. ACCEL.GEAR is independent of DECEL.GEAR.

DECEL.RATE
Sets the maximum deceleration rate commanded when speed is decreased.

DIM
Specifies and allocates storage for variables and arrays.

DIR
Sets the direction the motor turns when a GO.VEL function is executed.

DMF0
Sets the analog output channel's filter break frequency.

DMGAIN
Specifies the multiplicative scale factor applied to analog output signal when DACMAP=0.

ENABLE
Allows or prevents power flow to the motor.

ENABLED
Indicates whether the controller is enabled.

ENC.FREQ
Contains the frequency in quadrature pulses per second of the external encoder input averaged over a 128 msec interval for filtering.

ENC.IN
Specifies the line count of the encoder being used for electronic gearing.

ENC.OUT
Selects the direction of encoder ports and selects emulated encoder line count when the bidirectional encoder ports are set to “transmit.”

ENCPOS
Indicates the position of the external encoder.

END
Marks the end of a program, subroutine, IF...THEN...ELSE block structure or interrupt service routine.

EXIT
Used to exit a subroutine, an interrupt, FOR...NEXT or WHILE...WEND block structure.

FAULTCODE
Indicates the status of the controller.

FIX( )
Returns the truncated integer part of x.

FOR...NEXT
Allows a series of statements to be executed in a loop a given number of times.

FVEL.ERR
Indicates the velocity servo error signal, in RPM, after it has been processed by the anti-resonant filter section.

FWV
Indicates the controller firmware version number.

GEARERROR
Specifies the amount of position lag that accumulates when electronic gearing is turned on.

GEARING
Turns electronic gearing on or off and sets allowed direction of motion.

GEARLOCK
Indicates slave axis velocity is synchronized with the master, when performing electronic gearing.

GO.ABS
Causes the motor to move to the position specified by TARGET.POS.

GO.HOME
Moves the motor shaft to the electrical home position.

GO.INCR
Moves the motor shaft an incremental index from the current position.

GO.VEL
Moves the motor shaft at a constant speed.

GOSUB...RETURN
Branches program execution to a subroutine, executes it, and returns.

GOTO
Causes software to jump to a specific label and continue executing.

HEX$( )
Converts a long integer to a hexadecimal ASCII string.

ICMD
Indicates the commanded motor torque current in amperes.

IFB
Indicates the measured motor current amplitude in amperes.

IF...THEN...ELSE
Statements control program execution based on the evaluation of numeric expressions.

ILC
Sets the current loop proportional gain.

ILMT.MINUS
Sets the maximum allowable current in the counter-clockwise direction.

ILMT.PLUS
Sets the maximum allowable current in the clockwise direction.

INDEX.DIST
Sets the distance which the motor rotates during each incremental index.

INKEY$( )
Returns one character read from the serial port’s buffer.

IN.POS.LIMIT
Specifies the tolerance of commanded position minus action position within which region the position limit flag will be set.

IN.POSITION
Indicates whether or not the motor has achieved commanded position.

INPn
Reads the state of an individual discrete input.

INPUT
Reads a character string received by the serial communications port, terminated by a carriage return.

INPUTS
Reads the state of discrete inputs.

INSTR( )
Provides the location of a substring within a string.

INT( )
Truncates an expression to a whole number.

INTERRUPT
Marks the beginning and the end of an interrupt service routine.

INTR.{source label}
Used to enable and disable the interrupts.

IPEAK
Contains the peak current rating of the controller in amperes.

IFT0
Specifies the corner frequency of the low pass filter implementing the IT controller thermal protection circuit.

IT.FILT
Contains the output of the I*T controller thermal protection filter circuit.

IT.THRESH
Designates the threshold (trip level) at which the IT thermal protection triggers an overcurrent fault.

KPP
Sets the proportional gain of the position loop in Hz.

KTEFF
Torque constant of the system in lb-in/amp.

KVFF
Sets the proportion of velocity feedforward signal added to the velocity command from differentiated position command.

KVI
Sets the integral gain of the velocity servo loop.

KVP
Sets the proportional gain of the velocity servo loop.

LANFLT(1-32)[axis]
Shared user floating point variables for inter-axis communication. 32 variables per controller on the PacLAN network.

LANFLT(n) [axis#]
An array of 32 floating point variables globally accessible over PacLAN.

LANINT(1-32) [axis]
Shared user integer variables for inter-axis communication. 32 variables per controller on the PacLAN network.

LANINT(n) [axis#]
An array of 32 integer variables globally accessible over PacLAN.

LANINTERRUPT [axis#]
Invokes an interrupt to the PacLAN controller specified by [AXIS#].

LCASE$( )
Converts expression to lower case characters.

LEFT$( )
Returns a string of the leftmost characters of a string.

LEN( )
Returns the number of characters in the string.

LTRIM$( )
Removes the leading blank characters from a string expression.

LOG( )
Returns natural logarithm value.

LOG10( )
Returns base 10 logarithm of expression.

LOGGEDON
Indicates that the controller has been selected as the enabled multidrop subsystem by the multidrop master.

MID$( )
Replaces a portion of a string with another string.

MODEL
Indicates the servo controller model number.

MOVING
Indicates when a commanded motion profile has been completed.

OUTn
Sets the state of the individual discrete output.

OUTPUTS
Specifies the state of the discrete outputs.

PAUSE
Causes the program to pause the amount of time specified by the PAUSE.TIME variable.

PAUSE.TIME
Defines the length of time delay during a pause statement.

POLECOUNT
Matches the controller for the appropriate motor pole count.

POS.CHKn
Specifies the position at which outputs 1, 2 and 3 are switched to the polarity designated by POS.CHKn.OUT.

POS.CHKn.OUT
Used in conjunction with POS.CHKn to implement position check n.

POS.COMMAND
Contains the current position command.

POS.ERROR
Equal to the difference between the position command and the actual position.

POSITION
Indicates the actual motor position.

PRINT
Displays output on the terminal screen while the program is running.

PULSES.IN
Specifies the number of input encoder counts used for selecting an exact gear ratio.

PULSES.OUT
Specifies the number of resolver counts the motor will move for a specified number of encoder counts.

PWM12
Specifies the function of discrete output 12/PWM as a Pulse Width Modulated (PWM) signal and controls the duty cycle.

RATIO
Sets the electronic gearing ratio (rev/rev) between the encoder shaft and the motor shaft.

REG.DIST
Sets the distance that is moved automatically when a resolver registration input is applied.

REG.ENCPOS
Records the encoder position when an encoder registration input triggers.

REG.FLAG
Indicates that the registration input has triggered.

REG.FUNC
Specifies whether REG.DIST is the distance that is moved automatically when a resolver registration input is applied.

REG.MODE
Specifies the discrete input signal used to trigger and latch the encoder and resolver positions during registration.

REG.POS
Records the motor position when a resolver registration input triggers.

REG.RESPOS
Records the resolver position relative to the motor housing when a resolver registration input triggers.

REM or '
Designates explanatory remarks or comments in the program.

RESPOS
Contains the mechanical orientation of the resolver relative to the motor housing.

RESTART
Re-initializes the servo controller and commences execution after encountering an interrupt or program execution has become nested within levels of subroutines.

RIGHT$( )
Returns the rightmost characters of the string.

RUN.SPEED
Sets the maximum speed used in making an incremental or an absolute move.

RVEL
Indicates the actual speed at which the motor shaft is rotating.

SGN( )
Returns the sign of x.

SIN( )
Returns the sine of its argument in radians.

SPACE$( )
Returns a string of spaces.

SQR( )
Returns the square root of an expression.

STATUS[axis#]
A read-only software variable that can be used within a program to determine if an external axis is connected to PacLAN.

STEPDIR
Specifies response to either quadrature encoder signals or step and direction input signals when electronic gearing is in use.

STOP
Stops the execution of the program.

STR$( )
Returns a string representation of the value of a numeric expression.

STRING$( )
Returns a string containing the specified number of occurrences of a character.

SUB
Defines the beginning and ending of a subroutine.

SWAP
Exchanges the values of two variables.

TAN( )
Returns the tangent of its argument in radians.

TARGET.POS
Sets the target position used with the GO.ABS function.

TIME
Contains the current value, in seconds, of a free running timer that is maintained by the internal software.

TMENABLEn
Enables or disables the programmable timers.

TMOUTn
Indicates the state of an output controlled by a programmable timer.

TMRSET
Specifies the operation of the programmable timers.

UCASE$( )
Converts value to upper case.

UPD.MOVE
Updates a move in progress with new variables.

VAL( )
Returns the numerical value of the string.

VEL.CMD
Indicates the net velocity servo loop command signal in RPM.

VEL.ERR
Indicates the velocity servo error signal in RPM.

VELOCITY
Indicates the actual speed at which the motor shaft is rotating averaged over a 128 msec interval.

WHEN
Used for very fast output response to certain input conditions.

WHEN.ANALOG.IN
Records the digitized value of the analog input channel, in volts, at the time the WHEN statement is satisfied.

WHEN.DACMON
Records the value of the selected, filtered variable output to the analog output channel at the time the WHEN statement is satisfied.

WHEN.ENCPOS
Records the encoder position at the time the WHEN statement is satisfied.

WHEN.ICMD
Records the commanded motor torque current, in amperes, at the time the WHEN statement is satisfied.

WHEN.IFB
Records the measured motor current amplitude, in amperes, at the time the WHEN statement is satisfied.

WHEN.PCMD
Specifies the motor position when the WHEN condition is satisfied.

WHEN.POS
Set to the value of POSITION when the WHEN condition is satisfied.

WHEN.RESPOS
Records the resolver's mechanical orientation relative to the motor housing at the time the WHEN statement is satisfied.

WHEN.RVEL
Records the raw motor velocity at the time the WHEN statement is satisfied.

WHEN.TIME
Records the variable TIME at the time the WHEN statement is satisfied.

WHEN.VELCMD
Records the net velocity servo loop command signal, in rpm, at the time the WHEN statement is satisfied.

WHILE...WEND
Tells the program to execute a series of statements as long as an expression after the WHILE statement is true.

WVSHP
Integer composed of the ASCII codes for the characters in the name of the motor back EMF wave shape that the controller is using to shape the motor current.

The following arithmetic operators are supported:

ADDITION +

DIVISION /

EXPONENTIATION ^

MODULO DIVISION %

MULTIPLICATION *

SUBTRACTION -

PROGRAMMING EXAMPLES. . .PacSci ServoBASIC *Plus*™

These are actual program elements demonstrating the power of PacSci ServoBASIC *Plus*™ to easily program complex motion profiles.

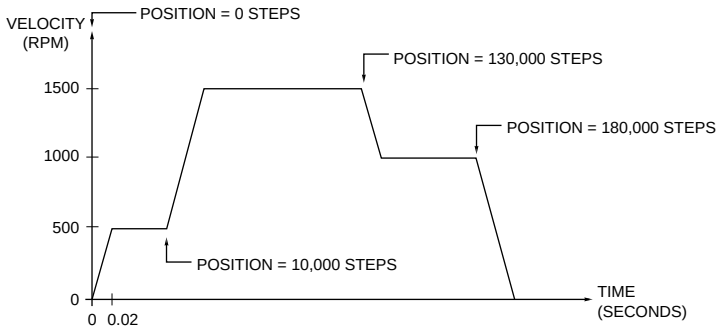
There's so much functionality in PacSci ServoBASIC *Plus* that you may never need to tap its full potential. If there's a software solution to your problem, our applications engineering people will find it.

BASIC PROGRAMMING EXAMPLES

EXAMPLE 1...Velocity change on position events

This program produces a multiple level velocity profile. Velocity changes are initiated based upon position events.

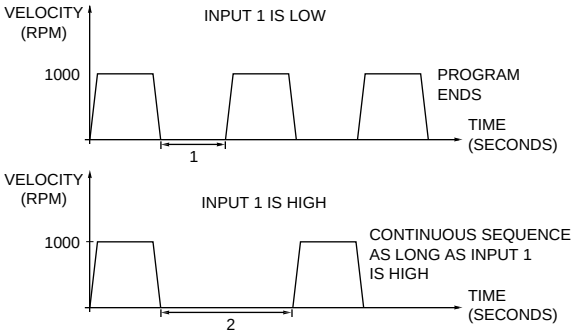
Program line	**Comments**
SETUP:	
ACCEL.RATE = 25000	SET ACCEL RATE TO 25,000 RPM/S
DECEL.RATE = 25000	SET DECEL RATE TO 25,000 RPM/S
RUN.SPEED = 500	SET RUN SPEED TO 500 RPM
IN.POS.LIMIT = 10	SET IN POSITION LIMIT TO ± 10 STEPS
HOME:	
GO.HOME	GO TO HOME POSITION
WHILE IN.POSITION < >1	WAIT FOR HOMING COMPLETE
WEND	
PROFILE:	
GO.VEL	START PROFILE, GO TO 500 RPM
RUN.SPEED = 1500	SET RUN SPEED TO 1500 RPM
WHILE POSITION <10000	WAIT FOR POSITION OF 10,000 STEPS
WEND	
GO.VEL	GO TO 1500 RPM
RUN.SPEED = 1000	SET RUN SPEED TO 1000 RPM
WHILE POSITION <130000	WAIT FOR POSITION OF 130,000 STEPS
WEND	
GO.VEL	GO TO 1000 RPM
RUN.SPEED = 0	SET RUN SPEED TO 0 RPM
WHILE POSITION <180000	WAIT FOR POSITION OF 180,000 STEPS
WEND	
GO.VEL	GO TO 0 RPM
END	END PROGRAM



BASIC PROGRAMMING EXAMPLES (Cont.)
Example 2. . .Jump on input

Here, the IF. . .THEN command is used to test a discrete input and take action depending upon the state of the input. The jumps in the program depend upon the state of Input 1.

Program line	Comments
SETUP:	
ACCEL.RATE = 10000	SET ACCEL TO 10,000 RPM/S
DECEL.RATE = 10000	SET DECEL RATE TO 10,000 RPM/S
RUN.SPEED = 1000	SET RUN SPEED TO 1,000 RPM
INDEX.DIST = 70000	SET INDEX DISTANCE TO 70,000 STEPS
READINPUT:	
IF INP1 THEN	IF INPUT 1 IS HIGH
PAUSE.TIME = 2	SET PAUSE TIME TO 2 SECONDS
GO.INCR	DO INDEX MOVE
WHILE MOVING = 1:WEND	WAIT TO COMPLETE MOVE
PAUSE	PAUSE 2 SECONDS
GOTO READINPUT	GO BACK TO READ INPUT 1 AGAIN
ELSE	IF INPUT 1 IS LOW
PAUSE. TIME = 1	SET PAUSE TIME TO 1 SECOND
FOR X = 1 TO 3	SET UP LOOP TO COUNT 3 INDEXES
GO.INCR	DO INDEX MOVE
WHILE MOVING = 1:WEND	WAIT TO COMPLETE MOVE
PAUSE	PAUSE 1 SECOND
NEXT	LOOP UNTIL 3 INDEXES COMPLETE
END IF	END IF. . .THEN. . .ELSE BLOCK
END	END PROGRAM



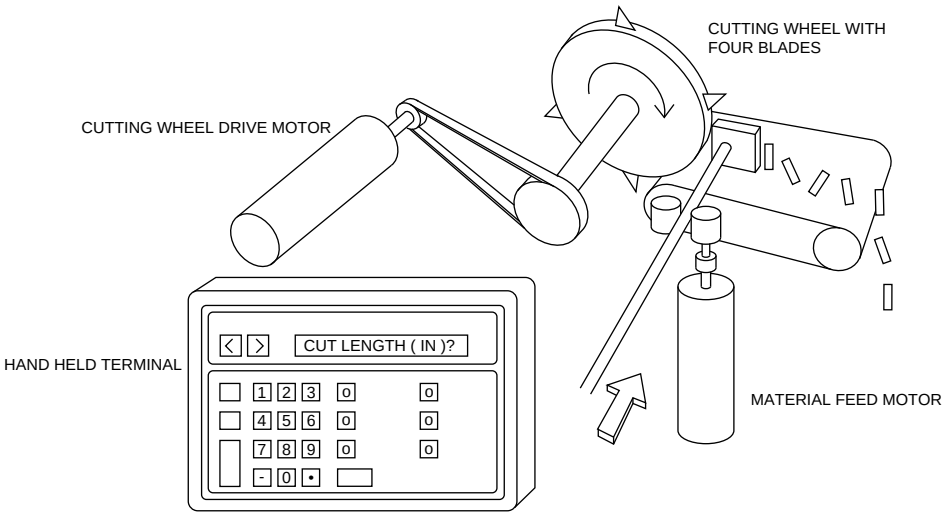
BASIC PROGRAMMING EXAMPLES (Cont.)
EXAMPLE 3. . .Cut-to-length operation

In this application, a rotary cut-to-length machine uses a short PacSci ServoBASIC *Plus*™ program to control independent cutting wheel and material feed axes.

This program drives an industrial terminal, such as the PacSci T10. The user is prompted for feed rate in inches per second and desired cut length in inches. The program then computes the required feed roll motor velocity based on feed roller diameter.

The cutting wheel motor runs at the computed speed as long as Input 1 is True. If Input 1 is brought False, the motor stops and the user is prompted for a new value.

Program line	**Comments**
SETUP:	
ACCEL.RATE = 10000	SET ACCEL TO 10,000 RPM/S
DECEL.RATE = 10000	SET DECEL RATE TO 10,000 RPM/S
DIR = 1	SET DIRECTION TO COUNTER CLOCKWISE
ENABLE = 1	SOFTWARE MOTOR ENABLE ON
PROMPTINPUTS:	
INPUT “FEED RATE (IN/SEC)”;FEEDRATE	PROMPT USER FOR INPUT
INPUT “CUT LENGTH (IN)”;CUTLENGTH	PROMPT USER FOR INPUT
PROFILE:	
RUN.SPEED = 75*FEEDRATE/CUTLENGTH	CALCULATE RUN SPEED
WHILE INP = 0	WHILE INPUT 1 IS LOW
GO.VEL	RUN AT CALCULATED SPEED
WEND	
ABORT.MOTION	STOP WHEN INPUT 1 IS HIGH
GOTO PROMPTINPUTS	PROMPT USER FOR NEW PARAMETERS



Cut-to-Length Machine

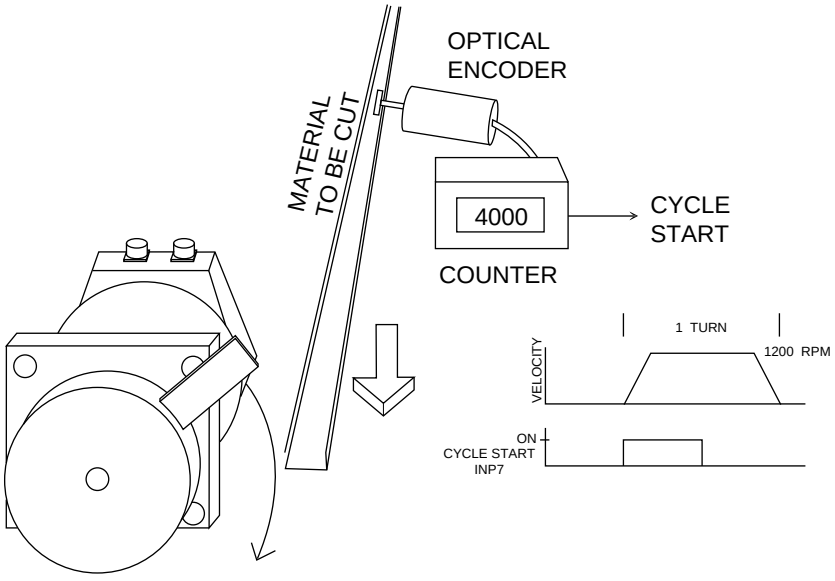
BASIC PROGRAMMING EXAMPLES (Cont.)
EXAMPLE 4. . .Using WHEN for rapid response

The PacSci ServoBASIC *Plus*™ instruction WHEN will help you achieve high speed response to inputs.

This program rotates a cutting blade one full revolution on the off-to-on transition of input INP7. WHEN ensures that the incremental move is initiated within one millisecond of the input transition.

Typically, the cut command is derived from an encoder and counter, which measure length of material passing the blade.

Program line	**Comments**
SETUP:	
ACCEL.RATE = 40000	SET ACCEL TO 40,000 RPM/S
DECEL.RATE = 40000	SET DECEL RATE TO 40,000 RPM/S
RUN.SPEED =1200	SET RUN SPEED TO 1200 RPM
INDEX.DIST = 4096	SET INDEX DISTANCE TO 4096 STEPS
IN.POS.LIMIT = 100	SET IN POSITION LIMIT TO ± 100 STEPS
OUTPUTS = 4095	SET ALL OUTPUTS OFF (HIGH)
PAUSE.TIME = 1	SET PAUSE TIME TO 1 SECOND
ENABLE = 1	SOFTWARE MOTOR ENABLE ON
NOTENABLED:	
WHILE ENABLE = 0	IF ENABLE INPUT IS NOT ACTIVE
OUT1 = 0	BLINK OUTPUT 1 ON (LOW)
PAUSE	FOR 1 SECOND
OUT1 = 1	AND OFF (HIGH)
PAUSE	FOR 1 SECOND
WEND	UNTIL ENABLE INPUT IS ACTIVATED
MOVE:	
OUT7 = 0	TURN OUTPUT 7 ON (LOW)
WHILE INP7 = 0:WEND	WAIT FOR INPUT 7 TO GO HIGH
WHEN INP7 = 0, GO.INCR	MOVE WHEN INPUT 7 GOES LOW
OUT7 = 1	TURN OUTPUT 7 OFF (HIGH)
WHILE IN.POSITION < >1:WEND	WAIT FOR MOVE COMPLETE
GOTO MOVE	GO BACK TO MOVE



ADVANCED PROGRAMMING EXAMPLES (Cont.)
EXAMPLE 5. . .Programmable limit switches (cont. on next page)

The following program generates a motion profile typical of transfer line slide drives.

Because absolute positioning is required, homing to a mechanical home limit switch is initiated by an off-to-on transition of input INP7. The mechanical home limit switch is connected to INP1. The SEEK_HOME subroutine supports the mechanical homing operation and establishes the zero position (POS.COMMAND = 0) electrical reference.

Outputs OUT2 and OUT3 are programmed as “programmable limit switches” using the POS.CHK commands. They indicate, within two milliseconds, whether actual position is above or below specified values.

For example, OUT2 could signal the bed drive that the tool is fully withdrawn and the workpiece can now be transferred to the next station.

OUT4 is programmed to be on whenever the tool is in the workpiece. Such an output might be used to control cooling fluid.

The HOME block initiates the homing function on a high to low transition of input 7. In the SETLIMITS block, the function of the limit switches is established.

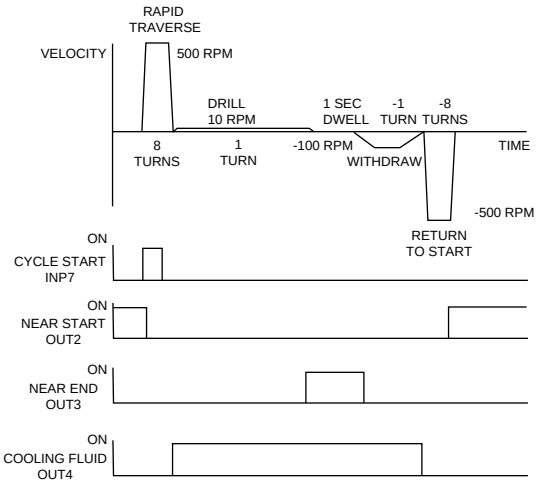
TRANSFER LINE SLIDE DRIVE PROGRAM

Program line	**Comments**
INITIALSETUP:	
ACCEL.RATE = 10000	SET ACCEL TO 10,000 RPM/S
DECEL.RATE = 10000	SET DECEL RATE TO 10,000 RPM/S
RUN.SPEED = 100	SET RUN SPEED TO 100 RPM
OUTPUTS = 4095	SET ALL OUTPUTS OFF (HIGH)
PAUSE.TIME = 1	SET PAUSE TIME TO 1 SECOND
POS.CHK1.OUT = 0	TURN OFF POSITION CHECK 1
POS.CHK2.OUT = 0	TURN OFF POSITION CHECK 2
POS.CHK3.OUT = 0	TURN OFF POSITION CHECK 3
DIR = 1	SET DIRECTION TO COUNTER CLOCKWISE
ENABLE = 1	SOFTWARE MOTOR ENABLE ON
NOTENABLED:	
WHILE ENABLED < > 1	IF ENABLE INPUT IS NOT ACTIVE
OUT1 = 0	BLINK OUTPUT 1 ON (LOW)
PAUSE	FOR 1 SECOND
OUT1 = 1	AND OFF (HIGH)
PAUSE	FOR 1 SECOND
WEND	UNTIL ENABLE INPUT IS ACTIVATED
HOME:	
OUT7 = 0	TURN ON (LOW) OUTPUT 7
WHILE INP7 = 0:WEND	WAIT FOR INPUT 7 TO GO HIGH
WHILE INP7 = 1:WEND	WAIT FOR INPUT 7 TO RETURN LOW
OUT7 = 1	TURN OFF (HIGH) OUTPUT 7
DIR = 1	
RUN.SPEED = 1000	
CALL SEEK__HOME	
SET LIMITS:	
POS.CHK2 = 8392	SET POSITION CHECK 2 AT 8392 STEPS
POS.CHK3 = 44856	SET POSITION CHECK 3 AT 44,856 STEPS
POS.CHK2.OUT = 11	OUTPUT 2 ON WHEN POSITION < POS.CHK2
POS.CHK3.OUT = 10	OUTPUT 3 ON WHEN POSITION > POS.CHK3
MOVETOSTART:	
ACCEL.RATE = 10000	SET ACCEL TO 10,000 RPM/S
RUN.SPEED = 500	SET RUN SPEED TO 500 RPM
TARGET.POS = 8192	SET TARGET POSITION TO 8192 STEPS
IN.POS.LIMIT = 10	SET IN POSITION LIMIT TO ± 10 STEPS
OUT4 = 1	TURN OUTPUT 4 OFF (HIGH)
GOSUB ABSMOVE	GO TO MOVE SUBROUTINE
OUT7 = 0	TURN OUTPUT 7 ON (LOW)
WHILE INP7 = 0:WEND	WAIT FOR INPUT 7 TO GO HIGH
OUT7 = 1	TURN OUTPUT 7 OFF (HIGH)
RAPIDTRAVERSE:	
ACCEL.RATE = 10000	SET ACCEL RATE TO 10,000 RPM/S
RUN.SPEED = 500	SET RUN SPEED TO 500 RPM
TARGET.POS = 40960	SET TARGET POSITION TO 40,960 STEPS
IN.POS.LIMIT = 50	SET IN POSITION LIMIT TO ± 50 STEPS
OUT4 = 1	TURN OUTPUT 4 OFF (HIGH)
GOSUB ABSMOVE	GO TO MOVE SUBROUTINE

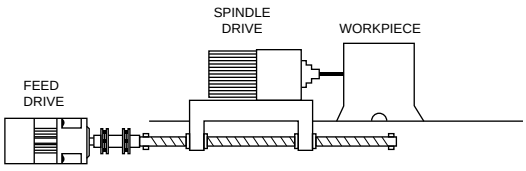
EXAMPLE 5. . .Programmable limit switches (cont. from prev. page)

Program line	Comments
DRILL:	
ACCEL.RATE = 1000	SET ACCEL RATE TO 1000 RPM/S
RUN.SPEED = 10	SET RUN SPEED TO 10 RPM
TARGET.POS = 45056	SET TARGETED POSITION TO 45,056 STEPS
IN.POS.LIMIT = 5	SET IN POSITION LIMIT TO ± 5 STEPS
OUT4 = 0	TURN OUTPUT 4 ON (LOW)
GOSUB ABSMOVE	GO TO MOVE SUBROUTINE
PAUSE	PAUSE FOR 1 SECOND
WITHDRAW:	
ACCEL.RATE = 1000	SET ACCEL RATE TO 1000 RPM/S
RUN.SPEED = 100	SET RUN SPEED TO 100 RPM
TARGET.POS = 40910	SET TARGET POSITION TO 40,910 STEPS
IN.POS.LIMIT = 50	SET IN POSITION LIMIT TO ± 50 STEPS
OUT4 = 0	TURN OUTPUT 4 ON (LOW)
GOSUB ABSMOVE	GO TO MOVE SUBROUTINE
GOTO MOVETOSTART	GO BACK TO START
GO.ABS	DO ABSOLUTE MOVE
WHILE NOT IN.POSITION:WEND	WAIT UNTIL MOVE COMPLETE
RETURN	RETURN TO GOSUB
END	

```
SIMPLIFIED SEEK__HOME ROUTINE
REM HOME SWITCH IS ACTIVE ON FALLING EDGE (CLOSES). THIS ROUTINE
REM COMMANDS MOTION UNTIL SWITCH SIGNAL GOES LOW, THEN REVERSES
REM DIRECTION AND DETERMINES LOCATION OF LOW-TO-HIGH
REM SWITCH SIGNAL (OPENING SWITCH) AND
REM ESTABLISHES ELECTRICAL HOME (POS.COMMAND = 0).
SUB SEEK__HOME
  GO.VEL
  RUN.SPEED = 0
  WHEN INP1 = 0, GO.VEL
  WHILE MOVING = 1:WEND
  IF DIR = 1 THEN DIR = 0 ELSE DIR = 1
  RUN.SPEED = 5
  GO.VEL
  RUN.SPEED = 0
  WHEN INP1 = 1, GO.VEL
  POS.COMMAND = POS.COMMAND = WHEN.PCMD
  END SUB
```



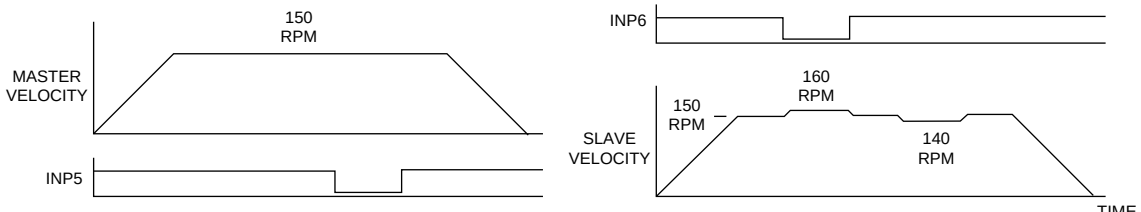
SLIDE DRIVE (MACHINE TOOL FEED)



EXAMPLE 6. . .Electronic gearing

In electronic gearing, a slave axis is electronically geared to a master. The master has a 1024 line encoder and is used in quadrature to produce 4096 transitions per revolution. The slave accepts 4096 transitions per revolution for a one to one ratio with the master. Electronic gearing is activated by an off to on transition of INP7.

Program line	Comments
SETUP:	
OUTPUTS = 4095	TURN ALL OUTPUTS OFF (HIGH)
GEARING = 0	TURN ALL ELECTRONIC GEARING OFF
ENC.OUT = 1024	SET FOR 1024 PPR ENCODER
RATIO = 1	SET ELECTRONIC GEAR RATIO TO 1:1
ACCEL.RATE = 10000	SET ACCEL TO 10,000 RPM/S
DECEL.RATE = 100000	SET DECEL RATE TO 100,000 RPM/S
RUN.SPEED = 10	SET RUN SPEED TO 10 RPM
OUT7 = 0	TURN OUTPUT 7 ON (LOW)
PAUSE.TIME = 1	SET PAUSE TIME TO 1 SECOND
ENABLE = 1	SOFTWARE MOTOR ENABLE ON
NOTENABLED:	
WHILE ENABLED < > 1	IF ENABLE INPUT IS NOT ACTIVE
OUT1 = 0	BLINK OUTPUT 1 ON (LOW)
PAUSE	FOR 1 SECOND
OUT1 = 1	AND OFF (HIGH)
PAUSE	FOR 1 SECOND
WEND	UNTIL ENABLE INPUT IS ACTIVATED
GEARINGON:	
WHILE INP7 = 0:WEND	WAIT FOR INPUT 7 TO GO HIGH
WHILE INP7 = 1:WEND	WAIT FOR INPUT 7 TO RETURN LOW
GEARING = 1	TURN ELECTRONIC GEARING ON
OUT7 = 1	TURN OUTPUT 7 OFF (HIGH)
OUT5 = 0	TURN OUTPUT 5 ON (LOW)
OUT6 = 0	TURN OUTPUT 6 ON (LOW)
DIRECTION:	
IF INP6 = 0 THEN	IF INPUT 6 IS LOW, SET
DIR = 0	DIRECTION OF 10 RPM VELOCITY TO CW
GO.VEL	SUPERIMPOSE 10 RPM ONTO GEARING
GOTO DIRECTION	GO BACK TO LOOK AT INPUT 6
END IF	END IF. . .THEN BLOCK
IF INP5 = 0 THEN	IF INPUT 5 IS LOW, SET
DIR = 0	DIRECTION OF 10 RPM VELOCITY TO CCW
GO.VEL	SUPERIMPOSE 10 RPM ONTO GEARING
GOTO DIRECTION	GO BACK TO LOOK AT INPUT 6
END IF	END IF. . .THEN BLOCK
ABORT.MOTION	STOP SUPERIMPOSED 10 RPM VELOCITY
GOTO DIRECTION	GO BACK TO LOOK AT INPUT 6



Go.Vel Superimposed On Gearing

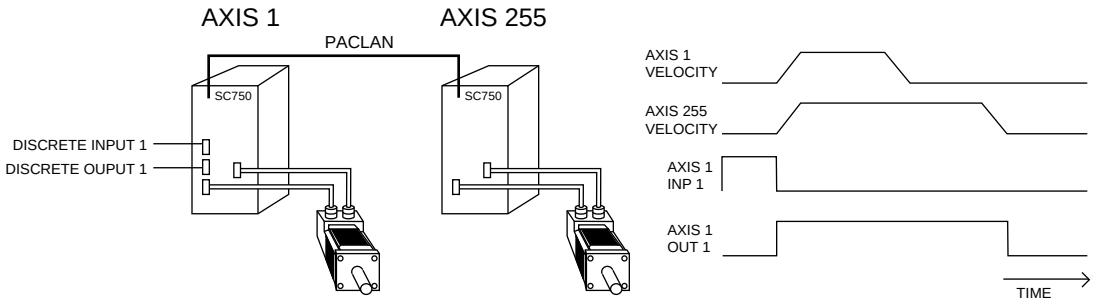
ADVANCED PROGRAMMING EXAMPLES (Cont.)

EXAMPLE 7. . .PacLAN™ Multi-axis position control capability using SC750 single axis digital position controllers

The PacLAN™ enhancement is a local area network (LAN) system that allows multiple single axis SC750 controllers to be integrated to provide effective and economical solutions to many multi-axis motion control applications. SC750 controllers networked via PacLAN provide distributed motion control and replace centralized multi-axis controllers. Using PacLAN, information can be exchanged with any SC750 controller on the network. Easy application monitoring and adjustments are made using a single operator interface or a host computer.

The 2.5 megabaud PacLAN allows up to 255 axes (nodes) to communicate and share information at distances up to 500 feet. PacLAN is a token passing communications port that utilizes CRC error detection, message acknowledgement, rugged twinaxial cabling and a fully transformer-isolated cable driver for reliable communications in high noise industrial environments.

This example shows the ease of synchronized motion in two axes with an input to the first axis.



The program for axis 1 waits for a one-to-zero transition of its discrete input 1 before initiating a ten turn incremental move. Similarly, axis 255 waits for the same transition to initiate a 20 turn incremental move. The program in axis 255 indicates that it is looking at an input to axis 1 by using the variable name “INP1[1].” The bracketed “1” is a call to PacLAN to fetch the data from axis 1 over the network. The network is also used to insure that both axes are in position before another move can be initiated.

AXIS 1

Program line	Comments
OUTPUTS = 4095	TURN OFF ALL OUTPUTS
ACCEL.RATE = 10000	SETUP AXIS 1 MOVE PARAMETERS
DECEL.RATE = 10000	
RUN.SPEED = 500	
INDEX.DIST = 10 * 4096	
IN.POS.LIMIT = 10	
PAUSE.TIME = .1	ENABLE AXIS 1 CONTROLLER
ENABLE = 1	
PAUSE	
WHILE 1 = 1	MAIN PROGRAM LOOP
OUT1 = 0	TELL THE WORLD BOTH AXES ARE HOLDING POSITION
WHILE INP1 = 0:WEND	DO INCREMENTAL MOVE ON HIGH-TO-LOW TRANSITION OF INP1
WHILE INP1 = 1:WEND	
OUT1 = 1	INDICATE MOTION IN PROCESS
GO.INCR	INITIATE MOVE
WHILE((IN.POSITION = 0) OR (IN.POSITION [255] = 0))	WAIT TILL BOTH AXES ARE AT POSITION
WEND	
WEND	
END	MAIN PROGRAM

AXIS 255

Program line	Comments
ACCEL.RATE = 10000	SETUP AXIS 255 MOVE PARAMETERS
DECEL.RATE = 10000	
RUN.SPEED = 500	
INDEX.DIST = 20 * 4096	
IN.POS.LIMIT = 10	
PAUSE.TIME = .1	ENABLE AXIS 255 CONTROLLER
ENABLE = 1	
PAUSE	
WHILE 1 = 1	MAIN PROGRAM LOOP
WHILE INP1[1] = 0:WEND	DO INCREMENTAL MOVE ON HIGH-TO-LOW TRANSITION OF AXIS 1’S INP1
WHILE INP1[1] = 1:WEND	
GO.INCR	
WHILE((IN.POSITION = 0) OR (IN.POSITION [1] = 0))	WAIT TILL BOTH AXES ARE AT POSITION
WEND	
WEND	
END	MAIN PROGRAM