KOLLMORGEN®

Because Motion Matters™

This is a Discontinued Product

Contact Kollmorgen Customer Support at 1-540-633-3545 or email us at support.kollmorgen.com if assistance is required.

PROGRAMMING

PROGRAMMING

This section provides an overview to the process of programming a Control. Once the "logic" behind the programming commands is understood, and the mix of numbers and letters explained, programming your Control will be seen as a straightforward process.

Programming of any sort requires planning and forethought. Programming your Control is no exception. This section will provide aids to facilitate your planning process.

Be patient! Allow time for adjustments ("debugging") and experimentation.

WHAT IS PROGRAMMING?

A program is a list of discrete numbered lines or Command Strings that, taken together in sequence, provide the information needed to get a machine to perform your predetermined sequence of instructions. These instructions can, in the case of a Programmable Motion Control, cause the motor to move at certain speeds and for given distances, plus read various inputs or set outputs to accomplish different machine-related tasks.

MEMORY TYPES AND USAGE

A program is stored in Memory. There are two kinds of memory. RAM (Random Access Memory) is called "Volatile Memory" because when power is removed from the Control, the information stored in that memory is lost. The Control, for example, stores manual commands (H Codes) in RAM.

The second kind of memory is "Nonvolatile Memory", such as BBRAM (Battery Backed RAM). The electrical codes stored in this type of memory are retained when external power is removed.

WHAT'S IN A PROGRAM?

A program for your Control will have several elements for each line or command string. These elements are: Line Numbers, G Codes, X (or Y) Fields for move distance and other values, and an F Field for Feedrate (motor speed) or other data.

A program in your Control can have up to 400 lines. Each line may contain up to 30 characters, including letters, numbers, marks and spaces, comprising one command string. You may store several programs as well, as long as the total number of lines of all the programs you have stored does not exceed 400. These different programs may be accessed by "pointing" to the line number on which they start This will be described in full detail later. In addition to program lines, the controller needs and stores (separate from the commands) a series of set-up parameters called L Codes.

Commands and Parameters

The types of commands and/or parameters your Control can accept are pre-set. Parameters are set in your Control with "L Codes," which are covered in more detail in the Software Reference Section of this Manual. As an example of how a parameter works, let's consider L12, which sets the motor's starting or "base" speed in pulses/sec. When this code is set at, say 450 pulses/sec, the motor's starting speed will be at this 450 pulses/sec rate every time a move is commanded by the program.

1. A "parameter" is not a part of your program. Rather, it is a set-up command that is entered prior to programming the Control.

2. A "command" in your program specifies the action you wish your motor to accomplish (speed, motion direction, delay time, etc).

   Commands (G Codes) are performed via numbered lines in your program. The program is a sequence of commands that control the motor and motion-related events you want to happen in a particular period of time.

A line number (N[nnn]):

G codes, X/Y codes and F codes are assigned line numbers. Parameters (L Codes) and Serial Commands (H Codes) do not have line numbers.

Line numbers must be entered by you. They are not generated by the Control. They take the form "N001" (an "N" followed by a number occupying up to three places - you may enter N1 for example, and when the program is listed the Control will display N001). Your Control can store up to 400 lines of program.

X and Y Fields

An X or Y Field is an area designated for specific information. The information in the field is a "Value" for a specific action.

"X/Y Fields" have three different purposes:

1. X/Y Fields are used alone or with G Codes specifying motion commands, to determine direction "+" (CW)/"-" (CCW), and number of motor pulses desired for the move.

2. X/Y Fields can also be used to control program operation by directing the flow of command. With the G11 command, for example, the X field specifies the line number of a subroutine for the program to go to. (An "F Field" value specifies how many times to repeat the subroutine.) When used with G20, the X field specifies the input condition for a conditional branch for the program to go to. (An F Field specifies the branch line number.) Other G Codes are used in similar fashion for program control.

3. X/Y Fields are used to input values for variables such as delay times in a program. This use of the X/Y Field is called for by the G04 command. Many other commands obtain their data from the X/Y field.

F Fields

There are two uses for F Fields.

1. F Fields are used in conjunction with X/Y Fields and G Codes to specify the "Feedrate" of these motions.

2. F Fields used in conjunction with commands such as G11 and G20 tell the program how to use the information in the subroutines and branches.

Examples of a program line:

Listed below is a typical program line with its line number, G Codes, and respective X and F Fields and Values. Here we can see how a Command String, or program line is structured.

N020 G91 X -2500 F2400

This line is interpreted as follows:

N020	-	Program line number 20
G91	-	Sets Incremental Mode
X-2500	-	The motor moves 2500 pulses in the "-" direction
F2400	-	Feedrate (speed) is 2400 pulses/sec.

Another example: N030 G11 X50 F2

N030	-	Program line number 30
G11	-	Calls subroutine
X50	-	Location of the subroutine (at line number 50)
F2	-	Specifies number of subroutine repetitions after the initial execution of the line

The subroutine will be repeated three times: once at the initial call, plus two more repetitions called for by the value 2 in the F field.

THE CODES: OVERVIEW

A complete listing of the codes together with their meanings and uses is provided in the Software Reference Section of this Manual. It will be helpful to refer to that section as needed while you are reading this section.

To further aid you in programming your Control, Superior Electric has prepared a Motion Control Program Development Software Package: MS-2000. This software simplifies the task of programming the Motion Control using a PC.

L CODES

L Codes are system parameters. Parameters are instructions that are active throughout all the operations of the Control program.

L Codes are not considered part of program, but are used as reference throughout the entire program operation.

L Codes are usually entered first, before you begin entering the first line of your program. They can be entered in any order. Also, they can be changed at any time, even during or after programming the Control.

This section describes the general types of L Codes used to set up your Control. Each L Code is explained in full detail in the Software Reference Section of this Manual.

Kinds of L Codes

There are four kinds of L Codes: Communication Parameters, Motion Parameters, Input/Output Control Parameters, and Execution Parameters.

1. Communication Parameters: These parameters control the flow of information between your Motion Control and the "host" device used to program and operate it. Included in this group are alphanumeric messages and characters that the Control can send to the host, as well as program and parameter transfer control.

2. Motion Parameters: These parameters are very important to consider in preparing your program. Jog speed (pulses/sec), acceleration and deceleration (pulses/sec/sec), low speed, home speed, offset distance from home, CW and CCW travel limits and Mechanical Home directions (+ or -) are all set by means of L Codes, not by G Codes or other program (line numbered) commands. Some of these parameters have factory-set default values. The default values are listed in a later section of this Manual.

   In addition, the closed loop option is set up and controlled through the use of certain L Codes.

3. Input/Output Control Parameters: These set up the assignments for the way the various input and output terminals are used.

4. Execution Parameters: These parameters determine how your program will be executed in relation to the motions you are commanding. Delays and program line counts are set here.

Some important facts about certain L Codes.

L26: Command Acknowledge

L26 is used to augment the XON/XOFF handshaking protocol to indicate Control readiness for situations where XON/OFF are inappropriate (such as a dumb terminal). Regardless of the L26 setting (0, 1, 2 or 3), the Control will always respond with an "=" to indicate the Control is ready or a ":" to indicate the Control is busy, in response to host device sending the attention character "<" and a valid device address "nn" This address must correspond with the value set on the ID# select switches on top of the Control.

G CODES AND RELATED X/Y and F FIELDS AND VALUES

G Codes are used to control program execution and modify the way a program operates. For example, subroutine calls, go to's, looping, and program branching are all accomplished using G Codes. Other uses include programming delays, sending messages, strobing data from BCD switches, and boosting or reducing the motor current.

Some G codes use the X/Y and F fields, and others do not.

Here are examples of G Codes that use neither X/Y Field nor F fields (X/Y and F are ignored):

G30: Program End

G31: Stop Program

G32: Return from Subroutine

Other G Codes require data in the X/Y field. Some examples are:

G04: Program Dwell Time

The X/Y field MUST be programmed with the delay time: range 0-9999 milliseconds.

G22: Wait for Input

The X/Y field MUST contain the desired input state:
0 = inactive, 1 = active, 2 or 9 for a Don't Care condition.

G47: Set/Reset Output Condition

The X/Y field MUST contain the desired output state:
0 = inactive, 1 = active, 2 or 9 for a Don't Care condition.

In the above examples, F field data is optional. If you wish to change motor speed on a line where one of those G Codes is used, simply program a new F value on the same program line as the G code.

Refer to Appendix E for a complete summary of the G codes and their related fields.

H CODES

H Codes are instructions that are issued to the Control from outside either the embedded program (using G Codes) or parameters (L Codes) you have set. H Codes are used for manual control and operation, or for transmitting to a computer (via the serial communications port) information and status conditions of the Control.

IMMEDIATE CODES

These are codes issued from your computer or other programming device that immediately act on the Motion Control's actions. They are not part of the embedded program and they are not stored in any memory. They are listed and fully described in the Software Reference Section and Appendices of this Manual.

These include:

*	CLEAR (Uncontrolled stop); The indexer immediately halts all motor motion and program execution
$	FEED HOLD (Controlled stop); The indexer immediately brings motor motion to a controlled stop with the programmed deceleration.
#	CYCLE STOP (Stop Program Execution Cycle); The indexer will stop program execution after the present program line completes its task
<nn	Device Attention Character (nn is the device ID#)
^H (Control H)	Backspace and Delete
^X (Control X)	Delete Line
?	Send active device ID number

A number of the H Codes are also available as Immediate codes by preceding them with an exclamation point (!).

IMPORTANT TERMS USED IN PROGRAMMING

ASCII:
"ASCII" stands for American Standard Code for Information Interchange. It is means of standardizing what numbers represent what characters on a typewriter or computer keyboard. For example a letter "A" is assigned the ASCII code 65, while "a" is assigned the code 97. While you use letters or symbols on a keyboard or keypad to enter information, the computer only recognizes the numerical value, or ASCII code.

A complete listing of ASCII characters is provided in an Appendix of this Manual.

Binary Coded Decimal:

Binary Coded Decimal numbers, or BCD's, are another means of representing characters (in this case the decimal digits 0 through 9) in binary code (0's and 1's). For instance, the decimal number 1 is equal to the binary (and BCD) number 0001; the decimal number 5 is equal to the binary or BCD 0101; decimal 9 is binary or BCD 1001. To form numbers larger than decimal 9, the individual digits are combined as in the following examples: decimal 10 is expressed as BCD 0001 0000; decimal 15 is expressed as BCD 0001 0101; and decimal 91 is expressed as BCD 1001 0001.

BCD's may be used to input data such as move distance or feedrate to your Control with thumbwheel switch panels.

NOTE: A more complete listing of BCD values is found in the Glossary.

WRITING YOUR OWN PROGRAMS

OVERVIEW

When writing your own program, there are several things to keep in mind:

1. You are providing instructions to a machine. **Keep it simple!**
2. Write down each program line as you have entered it, in sequence for later reference.
3. Be patient. Plan on going through several iterations.
4. Thoroughly familiarize yourself with this Instruction Manual, where the information is, what it is, and how it is presented, before you start.

WRITING AND LOADING THE PROGRAM:

1. Define your objectives. Write them down:

   "Move the belt 9 inches in 3 increments of 3 inches each, within a period of 6 seconds."

2. Break the objective down into discrete moves per axis. Write out the sequence for each axis. Make a Flow chart showing the sequence of the moves.

3. Find the G Codes that correspond to each of these moves in the table in the Software Reference Section, then read the detailed descriptions that are also listed in the Manual. Also consider the use of inputs and outputs, and code the appropriate program lines to control them.

4. Decide on your parameter settings. List the L Codes that are appropriate for this application (use the Programming Worksheet found at the back of this Manual). Write them down.

5. A sample "Programming Worksheet" has also been provided to help you prepare your program. It can be found at the back of this Manual. Please use it to help organize and record your program.

   The program can be written directly into the Control via the serial port, using either a terminal or a computer running a terminal emulation program (such as Superior Electric's MS-2000 disk). The program can also be written using a text editor, then downloaded to the Control via a communications program (such as "Crosstalk" or "Procomm" or MS-2000).

WRITING PROGRAMS FOR VARIOUS APPLICATIONS

This section explains various programming features of the control, the codes needed to accomplish typical applications, and examples of actual program lines which will make the control perform the required tasks. These examples are listed in order by complexity (from least complex to most) and by frequency of use (most frequent to least frequent). Many examples are included to aid you in learning how to program the control. It is important that you also consult the Software section for complete details on the use of each control parameter, code and command.

CONTROLLING SIMPLE MOVES

There are two basic types of moves: index and jog. Index moves specify a move distance, Jog moves do not. Jog moves the motor at a continuous controlled speed.

1. Setting the Basic Parameters

There are four parameters (L Codes) that specify the motor's velocity during a move. These are Base Speed, Acceleration, High Speed (or Feedrate) and Deceleration. During a simple move, the motor goes from standstill to base speed, accelerates to high speed, runs at high speed, decelerates to base speed and then stops.

Base Speed is the speed at which the motor starts and ends the move. It is typically set to a speed between 250 and 500 full steps/sec. L12 is the Base Speed parameter.

Acceleration and Deceleration specify the rate of change of motor speed in steps/sec/sec. They can differ by up to a factor of ten, i.e., the deceleration value can be between 1/10 and 10 times the acceleration value. Vertical lifting is an application where separate acceleration and deceleration can be used to advantage. L10 is the deceleration and L11 the acceleration parameter.

2. Setting the Move Distance

The move distance can be specified as Absolute or Incremental. An Incremental move specifies the distance to move from the present position. An Absolute move is specified in relation to electrical home, or absolute zero. The move distance is specified in the program or can be read in from thumbwheel switches or as BCD data from a PLC.

Example: N003 G91 X+200 F1000

In this example, G91 sets the X axis to incremental mode and calls for a move of 200 steps in the + direction with a high speed of 1000 steps/sec.

3. Choosing the Program Execution Mode

The program can be executed one line at a time (Single-Line), a fixed number of times (Automatic) or continuously (Continuous). This is controlled using parameter L06.

Program execution can be initiated by a cycle start command or automatically upon power up. There is a hardware input and a serial command for cycle start. For automatic startup upon power on, there is a 3 sec. delay built in to allow the motor drive to become ready. Program execution starts at the line number specified by L41.

Parameter L06, a 3 digit number, sets the execution mode. The Hundreds digit, 0 or 1, enables or disables motor motion. The Tens digit, 0 or 1, disables or enables autostart. The Units digit, 1,2 or 3, selects Single-Line, Automatic or Continuous program execution.

Applicable Codes:

- L06 Execution format
- L41 Program start line number
- L47 Repeat count

Examples:

L06 003	Motor motion enabled, program executes continuously upon receiving a cycle start
L06 102 L47 2	Motor motion disabled, program executes 3 times upon receiving a cycle start
L06 011 L41 22	Motor motion enabled, line 22 is executed upon power up.

USING THE BCD SWITCHES AND DATA SCALING FEATURE

The SS2000I Control has the capability of reading external data such as move distance, velocity and program line number from external BCD switches. This feature is quite useful for applications where the required data may vary and thus cannot be programmed with a fixed value. To take advantage of this ability the Control can be programmed using any of the G codes that instruct the it to read the data from the external BCD switches. When the Control executes one of these G codes the data that is present on the BCD switch is used as the data for that particular operation.

For example, under ideal conditions the Control would move a slide table a distance of 1 inch, however due to variances in the day to day process this distance must be adjusted to compensate for this variance. Rather than constantly reprogramming the Control with the adjusted distance value, an external BCD switch can be used to indicate the correct distance to move. The BCD value can then be adjusted as often as needed with out reprogramming the Control. Also, to facilitate the use of the external BCD data, a scale factor can be applied to compute the correct number steps to move a given distance.

In this example the slide table has a 10 pitch lead screw, meaning that 10 motor revolutions will cause 1 inch of table travel. Assuming there are 200 pulses per motor revolution, a total of 2000 pulses would be required to move the table 1 inch, with each pulse moving the table 0.0005 inches (1/2000). The requirement is to move in increments of 0.001 inches. This would require the operator to enter twice the desired value in order to obtain the proper move distance. By programming the scale factor with the correct value a direct 1 to 1 relationship can be obtained allowing the operator to enter the exact value to be moved. In this case a scale factor of 2 would be used achieve the desired move distance.

Applicable Codes:

L20	Power up configuration
L91	External BCD move distance scale factor
G52	Select BCD switch bank
G36	Read external move data and scale the value
G37	Read external line number
G38	Read external feedrate

Example:

N001 G36 F001000	Instructs the Control to read the BCD data, scale the data and move that distance at a velocity of 1000 pulses/sec
N001 G38	Instructs the Control to read the BCD data, and use that value as the new velocity

OPERATION FROM A PLC

The single point I/O facilitates connection to a PLC. The PLC can transfer data to the control using BCD format. The data is transferred 2 BCD digits at a time and the data length is 8 digits. The data range is 0 to +/- 79,999,999.

1. I/O setup

The transfer of data from the PLC requires 9 inputs and 4 outputs. Inputs 1-9 and Outputs 5-8 are configured using L58 and L59 respectively.

Applicable Codes:
L58 Input configuration
L59 Output configuration

Example: L58 "+0000BCCCCCCCC"
L59 "BBBB0000"

Inputs 1-8	configured as PLC Data ("C")
Input 9	configured as PLC Data Ready ("B")
Outputs 5-8	configured as PLC Strobes ("B")

The interface to the PLC uses handshaking. Refer to the detailed description of L58 for details. With L58 and L59 configured as above, the "PLC Mode" is enabled.

2. Data source selection

There are three sources of external data, PLC, BCD switch 1 and BCD switch 2. Code G52 selects the source of external data. Once selected, a given source will be used, to read all external data. When configured for PLC mode, the system defaults to select PLC data upon power up.

Applicable Codes: G52 Data source select

Examples: G52 X0 selects PLC data
G52 X1 selects BCD switch 1
G52 X2 selects BCD switch 2

3. Reading External Data

Applicable Codes: G10 Index from run
G29 Program L code
G36 Read move distance
G37 Read line number and branch
G38 Read feedrate

CHANGING SPEED DURING A MOVE

Changing speed during a move can be accomplished by 2 different methods. The first method involves issuing a speed control command via the serial port while the second method involves speed control commands that are part of the user's program.

Speed changes via the serial port requires the user to generate one of several different speed control commands to cause the Control to change the speed of the motor. These serial commands can be issued any time the motor is moving. All serial commands that control speed begin with an exclamation mark "!" and end with a carriage return and or linefeed. Speed control via the serial port is generally an synchronous approach to speed control since the timing of issuing the speed control commands can not always be repeated. For more precise control of speed changes, the programmed method of speed control should be used.

Speed changes under program control can be programmed to occur based on a variety of conditions. The most straightforward approach is to simply program the speed that you desire for each index. For example, "N001 X5000 F1000" would produce an indexed move of 5000 pulses at 1000 pulses per second. Speed changes while under program control may occur based on position, time or inputs.

Applicable Codes: G60 Enable continuous line execution mode
G22 Wait for input condition
G04 Delay
!H04 Set high speed mode
!H05 Set low speed mode
!H31 Increase speed
!H32 Decrease speed
!F Feedrate override

Speed change based on position

In this example the motor moves a total of 10,000 pulses. First it moves 2500 pulses at 1000 pulses/sec, then accelerates to 2000 pulses/sec. After 5000 pulses the speed changes to 750 pulses/sec which is held for the remainder of the move.

N001 G60 X+00010000 F001000
enable continuous execution mode, then move clockwise 10000 pulses at 1000 pulses/sec

N002 G62 X+00002500 F002000
wait for 2500 pulses to be moved then change speed to 2000 pulses/sec

N003 G62 X+00005000 F000750
wait for 5000 pulses to be moved then change speed to 750 pulses/sec

N004 G30
end program - program execution will wait here until the move is completed.

Speed change based on time

In this example the motor moves a total of 10,000 pulses. The motor first moves at 1000 pulses/sec for 2.5 seconds, then accelerates to 2000 pulses/sec. After an additional 3 seconds the motor then changes speed to 750 pulses/sec which is held for the remainder of the move.

N001 G60 X+00010000 F001000
enable continuous execution mode, then move clockwise 10000 pulses at 1000 pulses/sec

N002 G04 X+00002500 F002000
wait for 2.5 seconds to elapse, then change speed to 2000 pulses/sec

N003 G04 X+00003000 F000750
wait for 3 seconds to elapse, then change speed to 750 pulses/sec

N004 G30
end program - program execution will wait here until the move is completed.

Speed change based on Inputs

In this example, the motor moves a total of 10,000 pulses. The motor will run at 1000 pulses/sec until input 1 is activated. After activating input 1 the motor will then accelerate to 2000 pulses/sec and will run at this speed until input 2 is activated. After activating input 2 the speed changes to 750 pulses/sec which is held for the remainder of the move.

N001 G60 X+00010000 F001000
enable continuous execution mode, then move clockwise 10000 pulses at 1000 pulses/sec

N002 G22 X+22222221 F002000
wait for input 1 to become active, then change speed to 2000 pulses/sec

N003 G22 X+22222212 F000750
wait for input 2 to become active, then change speed to 750 pulses/sec

N004 G30
end program - program execution will wait here until the move is complete

USING BREAKPOINT PROGRAMMING

Breakpoint programming is used in the continuous line execution mode. This allows the Control to start a move on one line then continue to execute the next line. The lines that are executed while the motor is moving can be used to set up breakpoints. These breakpoints can be based on distance, time or velocity. For example, let's assume that the Control is in the continuous line execution mode and that a move of 25,000 pulses has been started at a velocity of 1000 pulses/sec. The application requires that once 3000 pulses have been moved the velocity needs to be changed to 5000 pulses/sec and when 16000 pulses have been moved the velocity needs to be changed to 1000 pulses/sec. This application would use position breakpoints to tailor the move profile.

Applicable Codes:
- G60 Enable continuous line execution mode
- G61 Disable continuous line execution mode
- G62 Wait for distance to be achieved
- G63 Wait for velocity to be achieved

Example:

N001 G60 X+0025000 F001000
continuous line execution mode: move 25000 pulses clockwise at a velocity of 1000 pulses/sec

N002 G62 X+0003000 F005000
wait for 3000 pulses to be moved, then change velocity to 5000 pulses/sec

N003 G62 X+0016000 F001000
wait for 16000 pulses to be moved, then change velocity to 1000 pulses/sec

N004 G30 end of program

PERFORMING I/O OPERATIONS DURING INDEXED MOTION

To perform input and output operations during indexed motion the continuous mode of execution must first be enabled. With this mode enabled, a line calling for indexed motion is executed, motion will start and then program execution will continue with the next programmed line. This line may call for a time delay or for waiting until a position or velocity is reached or an input is activated before continuing with program execution. By using these commands to control program execution, external devices may be signalled at the appropriate time by the setting of outputs.

Applicable Codes:
- L58 Input configuration
- L59 Output configuration
- G04 Delay
- G22 Wait for input condition
- G47 Set outputs
- G60 Enable continuous execution mode
- G61 Disable continuous execution mode
- G62 Wait for move distance
- G63 Wait for velocity condition

Example 1. In this example continuous execution is enabled. The motor then moves a total of 10,000 pulses at a velocity of 1,000 pulses/second. Output 1 is activated when the motor reaches a velocity of 1000 pulses/sec. The output remains active for 1 second and is then turned off.

N001 G60 X+00010000 F001000
enable continuous mode, then move 10,000 pulses at 1000 pulses/sec

N002 G63 F001000
wait for a velocity of 1000 pulses/sec to be reached

N003 G47 X+00000001
turn on output 1, outputs 2 thru 8 are turned off

N004 G04 X+00001000
delay for 1 second (1000 milliseconds)

N005 G47 X+00000000
turn off all outputs

N006 G30
wait for move to finish then end program

Example 2. In this example continuous execution is enabled. The motor then moves a total of 10,000 pulses at a velocity of 1,000 pulses/second. Output 1 is activated when the motor reaches a position of 2000 pulses. The output remains active until the motor reaches a position of 9000 pulses and is then turned off.

N001 G60 X+00010000 F001000
enable continuous mode, then move 10,000 pulses at 1000 pulses/sec

N002 G62 X+00001000
wait for a position of 1000 pulses to be reached

N003 G47 X+00000001
turn on output 1, outputs 2 thru 8 are turned off

N002 G62 X+00009000
wait for a position of 9000 pulses to be reached

N005 G47 X+00000000
turn off all outputs

N006 G30
wait for move to finish, then end program

USING ADDITIONAL INPUTS AND OUTPUTS

The Control has the capability to provide 8 program testable inputs and 8 program controlled outputs. Depending on the number of inputs and outputs assigned to functions other than program I/O, the expansion port which is normally used for accessing BCD data may instead be used for expanded program controlled I/O. An additional 8 Inputs and 8 outputs are available on this port. This allows for a total of 16 inputs and 16 outputs if the application requires it.

Expansion I/O is optional. The user may purchase industry-standard I/O boards such as "OPTO-22" type PB16A or "Potter & Brumfield" type 2IO-16. The expansion card is connected to the Control's top-mounted BCD / Expansion I/O connector via a 25-conductor cable (male 25-pin subminiature "D" connector at each end), by means of the accessory adapter board, Superior Electric part 221158-001 which attaches to the edge connector on the I/O card. On the I/O card, slots 0 to 7 must be inputs, and slots 8 to 15 must be outputs.

To access the additional I/O the L20 parameter would be modified to L20 nn1nn. This tells the Control that the expansion port is to be configured for additional I/O. The same G codes G20, G22 and G47 that access the standard I/O will now be able to access the expanded I/O. To have the Control use the expanded I/O simply use Y instead of X when programming the Control. For example "G20 Xnnnnnnnn" would instruct the Control to base a branch decision the state of the standard I/O while "G20 Ynnnnnnnn" would instruct the Control to base a branch decision on the state of the expanded I/O. Likewise "G47 Xnnnnnnnn" would instruct the Control to set or reset outputs using the standard I/O while G47 "Ynnnnnnnn" would instruct the Control to set or reset outputs using the expanded I/O.

Applicable Codes:

L20	Power up configuration
L59	Output configuration
G20	Branch on input condition
G22	Wait for input condition
G47	Set/reset outputs

Example:

N001 G22 X+00000111 Instructs the Control to wait for standard inputs 1, 2 and 3 to be active and standard inputs 4 through 8 to be inactive before continuing with program execution

N002 G22 Y+00000111 Instructs the Control to wait for expansion inputs 1, 2, and 3 to be active and expansion inputs 4 through 8 to be inactive before continuing with program execution

N003 G20 X+00000111 F000050
Instructs the Control to go to line 50 if standard inputs 1, 2 and 3 are active and standard inputs 4 through 8 are inactive.

N004 G20 Y+00000111 F000050
Instructs the Control to go to line 50 if expansion inputs 1, 2 and 3 are active and expansion inputs 4 through 8 are inactive.

N005 G47 X+00000001 Instructs the Control to turn on standard output 1 and turn off standard outputs 2 through 8.

N006 G47 Y+00000001 Instructs the Control to turn on expansion output 1 and turn off expansion outputs 2 through 8.

SENDING ASCII MESSAGES

In some applications it may be required to prompt an operator to select or start a specific process. In some cases this prompt may be something as simple as turning on a light and waiting for a button to be pressed before continuing with the program. In other cases more information must be presented to the operator before a decision can be made. If the application requires the ability to prompt the operator with detailed messages, the SLO-SYN 2000 Control's ability to send messages to a display may be used. The Control provides the user with the capability of programming 5 different messages with each message being 20 characters long. The messages may be transmitted one at a time or several messages may transmitted in sequence to be combined into longer messages. The messages are transmitted via the serial port, so the device connected to the serial port must be capable of receiving and displaying standard ASCII characters. After each message is transmitted, a carriage return and line feed will automatically be sent.

Applicable Codes: 1

L01	Message 1
L02	Message 2
L03	Message 3
L04	Message 4
L05	Message 5
G05	Transmit message 1 - 5

Example:

L01 "Load a part now"

N001 G05 X+00000001 send message #1

Example:

L01 "This is an example"
L02 "of combining text"
L03 "to send messages"
L04 "longer than 20"
L05 "characters."

N001 G05 X+00000001 send message #1
N002 G05 X+00000002 send message #2
N003 G05 X+00000003 send message #3
N004 G05 X+00000004 send message #4
N005 G05 X+00000005 send message #5
N006 G30 end program

USE OF THE SECOND AXIS

The SLO-SYN 2000 Control has the capability of driving a second axis of motion. The second axis has all the features of the primary axis except closed loop operation. The second axis is independent of the primary axis. A separate position counter and status register is maintained; also various L codes support two fields which allow for independent parameters. The only operation that is not supported, is for both the primary and secondary axes to move at the same time. Assuming that the default parameters are still loaded and that two drives are properly connected to the Control it is simple matter to program both axes for motion.

Example:

N001 G91 X+00001000 F001000
enable incremental positioning mode for the X axis, move the X axis clockwise 1000 pulses at 1000 pulses/sec

N002 G90 Y-00002300 F000750
enable the absolute positioning mode for the Y axis, move the Y axis to an absolute position of 2300 pulses at 750 pulses/sec

N003 X+00001000
move the X axis clockwise an additional 1000 pulses at 1000 pulses/sec

N004 Y+00002000
move the Y axis to an absolute position of 2000 pulses at 750 pulses/sec

N005 G30 end of program

MARK REGISTRATION (INDEX FROM RUN)

Mark registration (sometimes called Index From Run) is a means by which an indexed move can be performed when the total move distance is unknown. This incredible ability is actually quite simple, by connecting an external sensor to the Control's trigger input the Control can be made to run until this sensor detects a registration mark. At this point the Control will switch to an Index mode and will then move a preset distance and stop. This feature is useful when the distance from the registration mark is known but the distance between registration marks is unknown. The accuracy for this type of motion is typically +/- 1 pulse.

Applicable Codes: L16 Mark registration travel limit
L20 Power up configuration
G10 Mark registration mode

Example:

N001 G10 X+00001000 F001000
enable mark registration mode then start running clockwise at 1000 pulses/sec, wait for trigger to occur, and when trigger occurs move 1000 pulses at 1000 pulses/sec.

N002 G47 X+00000001
turn on output 1, all other outputs off

N003 G30 end of program

SOFTWARE REFERENCE

This page intentionally left blank

L CODES

CONTROL PARAMETER (L CODE) FUNCTIONAL LISTING

System Parameters

- L20 System Configuration
- L58 Input Configuration
- L59 Output Configuration

Travel Limit Parameters

- L16 Mark Registration Travel Limit
- L18 Clockwise Software Travel Limit
- L19 Counterclockwise Software Travel Limit

Serial Port Parameters

- L26 Transmission Protocol
- L48 Program Line Count List/Clear
- L49 First Parameter To List
- L50 Parameter List Count
- L98 Delay Between Automatic Transmissions

Serial Port Messages

- L01 Message #1
- L02 Message #2
- L03 Message #3
- L04 Message #4
- L05 Message #5
- L52 Buffer Full Warning Message
- L53 Following Error Warning Message
- L54 Unable To Reach Position Warning Message
- L55 Line Done Message
- L56 Program Done Message

Execution Parameters

- L06 Program Execution Format
- L41 Program Start Line Number
- L44 Delay After Motion
- L47 Program Execution Repeat Count
- L91 External Move Distance Scale Factor

Manual Motion Parameters

- L09 Jog Speed
- L13 Step Distance
- L73 Speed Increment

Motion Parameters

- L10 Deceleration
- L11 Acceleration
- L12 Low Speed
- L71 Maximum Speed
- L72 Velocity Profile

Home Parameters

- L08 Mechanical Home Direction
- L14 Home Speed
- L17 Offset From Home Switch

Backlash Parameters

- L43 Backlash Compensation Delay
- L66 Backlash Compensation

Closed Loop Parameters

- L87 Following Error
- L90 Closed Loop Configuration
- L93 Position Deadband
- L94 Encoder Counter Direction
- L95 Encoder Scale Factor
- L96 Position Correction Attempts
- L97 Delay Between Correction Attempts

```
L01  Message #1
L02  Message #2
L03  Message #3
L04  Message #4
L05  Message #5
```

Modified per Rev. B

Function

Defines the messages that are transmitted when a Transmit Message (G05) command is executed.

Command Format

L01 "aaaaaaaaaaaaaaaaaaaa"
L02 "aaaaaaaaaaaaaaaaaaaa"
L03 "aaaaaaaaaaaaaaaaaaaa"
L04 "aaaaaaaaaaaaaaaaaaaa"
L05 "aaaaaaaaaaaaaaaaaaaa"

Range

ASCII characters from 32 to 126 decimal with the following exceptions:

" decimal 34
$ decimal 36
* decimal 42
< decimal 60

Maximum message length is 20 characters.

Defaults

L01 = "Message #1 goes here"
L02 = "Message #2 goes here"
L03 = "Message #3 goes here"
L04 = "Message #4 goes here"
L05 = "Message #5 goes here"

Notes

Quotes must be used when programming L01 - L05. The designated message will be transmitted via the serial port when a G05 command is executed.

The message is sent only if the unit's ID is active.

An "*" is substituted for a "~" when the message is sent and "CRLF" is inhibited.

An "LF" is sustituted for a "^" when the message is sent and "CRLF" is inhibited

See Appendix H for further details.

L06 Program Execution Format

Function

Defines the way in which a program is executed.

Command Format

L06 nnn

n--	0 =	Program Debug Mode off, program executes as specified and motor motion will occur.
	1 =	Program Debug Mode on, program executes as specified but motor motion is not allowed.
-n-	0 =	AutoStart disabled, execution begins with a cycle start command.
	1 =	AutoStart enabled, execution begins upon power up.
--n	1 =	Single-Line Execution
	2 =	Complete Program Execution
	3 =	Continuous Program Execution

Default

L06 002 Program Debug Mode off
Autostart disabled
Complete Program Execution

Notes

In the Program Debug Mode (L06 1nn), the pulse output is disabled, thereby inhibiting all motion. This mode may be useful when debugging a program.

If AutoStart is enabled (L06 n1n), program execution commences upon power up, otherwise a cycle start command is required.

In the Single-Line Execution Mode (L06 nn1), each program line requires a Cycle Start for execution.

In the Complete Execution Mode (L06 nn2), the entire program is executed once plus the number of times indicated by the Program Execution Repeat Count parameter (L47).

In the Continuous Execution Mode (L06 nn3), the entire program is executed continuously.

Related parameters

L47 (Program Execution Repeat Count)
L41 (Program Start Line Number)

L08 Mechanical Home Direction

Function Sets motor rotation direction for a Mechanical Home Cycle.

Command Format
L08 s,s Set X and Y Home Directions.
L08 s Set X axis Home Direction.
L08 ,s Set Y axis Home Direction.

Range s + or - (CW or CCW rotation)

Default +,+ (CW rotation for X and Y)

Related parameters
L14 (Home Speed)
L17 (Offset From Home Switch)

Examples
L08 -,+ X Home Direction is - (CCW), Y is + (CW)

L08 ,- Y Home Direction is - (CCW)

L09 Jog Speed

Function Feedrate used to Jog or Step motor.

Command Format

L09 nnnnnn,nnnnnn	Set X and Y jog speeds.
L09 nnnnnn	Set X axis jog speed.
L09 ,nnnnnn	Set Y axis jog speed.

Range nnnnnn 1 - 999,999 pulses/sec

Default Value equivalent to 1000 Full Steps/sec, exact number depends on selected microstepping resolution. See Default table in Appendix A.

Examples

L09 2000,2000	Sets X and Y axis Jog Speed to 2000 pulses/sec
L09 2000	Sets X axis Jog Speed to 2000 pulses/sec

L10 Deceleration
L11 Acceleration

Function

L10 defines the rate at which the motor speed is decreased and L11 the rate at which it is increased.

Command Format

L10 nnnnnnnn,nnnnnnnn	Set Deceleration for both axes.
L10 nnnnnnnn	Set X axis Deceleration.
L10 ,nnnnnnnn	Set Y axis Deceleration.
L11 nnnnnnnn,nnnnnnnn	Set Acceleration for both axes.
L11 nnnnnnnn	Set X axis Acceleration.
L11 nnnnnnnn	Set Y axis Acceleration.

Range

nnnnnnnn 1 - 99,999,999 pulses/sec/sec
Deceleration (L10) 0.1 to 10 times Acceleration (L11)

Default

Value equivalent to 1000 Full Steps/sec/sec, exact number depends on selected microstepping resolution. See Default table in Appendix A.

Notes

The Deceleration value can be from 0.1 to 10 times the Acceleration value (L11).

If a Deceleration value is entered that is outside this range it is discarded, the old value retained and the Illegal L code program error status bit set.

If the Acceleration (L11) value is changed, the Deceleration value is checked against the range based on the new Acceleration. If the Deceleration value is outside this new range, it is set equal to the Acceleration value.

Related parameters

L72 (Velocity Profile)

Examples

L10 2000,2000	Set the X and Y Deceleration to 2000 pulses/sec/sec.
L11 ,2000	Set the Y axis Acceleration to 2000 pulses/sec/sec.

L12 Low Speed

Function

The starting/stopping speed in High Speed mode and the feedrate for all moves in Low Speed mode.

Command Format

L12 nnnnnn,nnnnnn	Sets X and Y axes Low Speed.
L12 nnnnnn	Sets X axis Low Speed.
L12 ,nnnnnn	Sets Y axis Low Speed.

Range

nnnnnn 0 - 999,999 pulses/sec

Default

300 Full Step/sec equivalent, exact number depends on selected microstepping resolution. See Default table in Appendix A.

Notes

During a typical move, the motor starts at Low Speed, accelerates to the target speed then decelerates to Low Speed and stops. When the target speed is less than the low speed, the motor will start and stop at the target speed.

When Low Speed mode is selected, all programmed feed rates are ignored and the Low Speed value is used for all motion.

Example

L12 320,320	Sets X and Y Low Speed to 320 pulses/sec.
L12 320	Sets X axis Low Speed to 320 pulses/sec.

L13 Step Distance

Function Distance moved in response to a step command.

Command Format

L13 nnnnnnnn,nnnnnnnn	Sets X and Y axes step distance.
L13 nnnnnnnn	Sets X axis step distance.
L13 ,nnnnnnnn	Sets Y axis step distance.

Range nnnnnnnn 1 - 99,999,999 pulses

Default 1 Full step equivalent, exact number depends on selected microstepping resolution. See Default table in Appendix A.

Notes A step command can be initiated by hardware inputs or by commands (H Codes) via the serial port.

Example

L13 200,200	Set X and Y step distance to 200 pulses
L13 200	Set X axis step distance to 200 pulses

L14 Home Speed

Function Speed used for Electrical (G76) or Mechanical Home (G78) cycles.

Command Format

L14 nnnnnn,nnnnnn	Set X and Y axis home speed.
L14 nnnnnn	Set X axis home speed.
L14 ,nnnnnn	Set Y axis home speed.

Default 1000 Full Steps/sec equivalent, exact number depends on selected microstepping resolution. See Default table in Appendix A.

Notes The feedrate during an Electrical or Mechanical Home cycle is either the Home Speed (L14) or the Low Speed (L12). The Low Speed is used when Low Speed Mode is active.

Related parameters
L08 (Mechanical Home Direction)
L17 (Offset From Home Switch)

Example:

L14 3000,3000	sets the X and Y home speed to 3000 pulses/sec
L14 ,3000	sets the Y axis home speed to 3000 pulses/sec

L16 Mark Registration Travel Limit

Function Defines the maximum move distance allowed in a Mark Registration Cycle.

Command Format L16 nnnnnnnn Set mark registration travel limit

Range nnnnnnnn 0 - 99,999,999 pulses

Default 0 (limit disabled)

Notes There is only one mark registration travel limit, it is used with both X and Y. The travel limit is an incremental distance, relative to the start of the move. A zero value denotes no limit (disabled).

The travel limit will terminate a Mark Registration cycle, should the registration input fail to occur. When the L16 distance is exceeded, the motion is decelerated to a controlled stop and bit 8 of System Error Status set.

Example L16 2000 Sets the Mark Registration Travel Limit to 2000 pulses.

L17 Offset From Home Switch

Function Defines the final position of a mechanical home cycle, as an offset (distance and direction) from the position, where the home input is activated.

Command Format

L17 snnnnnnnn,snnnnnnnn	Sets Offset From Home Switch for both X and Y.
L17 snnnnnnnn	Sets Offset From Home Switch for the X axis.
L17 ,snnnnnnnn	Sets Offset From Home Switch for the Y axis.

Range snnnnnnnn -99,999,999 to +99,999,999 pulses

Default +0,+0 (no offset)

Notes When the final position is achieved during a mechanical home cycle, the absolute position is set to 0.

The Offset From Home Switch parameter is not used during an Electrical Home cycle.

Related parameters L08 (Mechanical Home Direction)
L14 (Home Speed)

Example L17 +1000,+1000 Sets X and Y Offset From Home +1000 pulses for Switch values.

L17 +1000 Sets the X axis Offset From Home +1000 pulses for Switch value.

L18 Clockwise Software Travel Limit
L19 Counterclockwise Software Travel Limit

Function

CW motion is terminated or not allowed when absolute position is more positive or equal to L18. CCW motion is terminated or not allowed when the absolute position is more negative than L19.

Command Format

L18 snnnnnnnn,snnnnnnnn	Sets X and Y CW software travel limit.
L18 snnnnnnnn	Sets X axis CW software travel limit.
L18 ,snnnnnnnn	Sets Y axis CW software travel limit.
L19 snnnnnnnn,snnnnnnnn	Sets X and Y CCW software travel limit.
L19 snnnnnnnn	Sets X axis CCW software travel limit.
L19 ,snnnnnnnn	Sets Y axis CCW software travel limit.

Range

s	+ or -	
nnnnnnnn	0 - 99,999,999	pulses

Default

L18: -0, -0 CW (limit disabled)
L19: +0, +0 (CCW limit disabled)

Notes

A minus L18 value disables the CW software limit.

CW motion is not allowed unless the absolute position is less positive than the L18 value. When, during CW motion, the absolute position becomes more positive or equal to L18, the motion is decelerated to a controlled stop and bit 5 of Motion Error Status is set.

A positive L19 value disables the CCW software limit.

CCW motion is not allowed unless the absolute position is less negative than the limit. When, during CCW motion, the absolute position becomes more negative than L19, the motion is decelerated to a controlled stop and bit 4 of Motion Error Status is set.

Examples

L18 +100000,-100000	X CW software limit at +100000 pulses, Y disabled.
L19 -100000,+100000	X CCW software limit at -100000 pulses, Y disabled
X absolute position =	90000, X axis CW motion is allowed.
X absolute position =	100000, X axis CW motion is not allowed.
X absolute position =	-90000, X axis CCW motion is allowed.
X absolute position =	-100000, X axis CCW motion is not allowed.

L20 System Configuration

Modified per Rev. C

Function

Configures inputs TR1, TR2 and the expansion port. Specifies the electrical position, encoder position and feedrate defaults at power up.

Command Format

L20 nnnnn

TR1	n—	0 -	X axis Registration or Home input, positive edge triggered.
		1 -	X axis Registration or Home input, negative edge triggered.
		2 -	Y axis Registration or Home input, positive edge triggered.
		3 -	Y axis Registration or Home input, negative edge triggered.
		4 -	X axis Home input, encoder index and TR1 active.
TR2	-n—	0 -	X axis Registration or Home input, positive edge triggered.
		1 -	X axis Registration or Home input, negative edge triggered.
		2 -	Y axis Registration or Home input, positive edge triggered.
		3 -	Y axis Registration or Home input, negative edge triggered.
		4 -	X axis Home input, encoder index and TR2 active.
	—n—	0 -	Expansion Port interfaced to BCD switches.
		1 -	Expansion Port interfaced to expansion I/O.
		2 -	Expansion Port interfaced to SS2000 Expansion I/O board
	—n-	0 -	X and Y absolute position = 0. General purpose registers are set to zero on power up.
		1 -	The X and Y absolute positions saved at power down are restored. General purpose registers are set to zero on power up.
		2 -	(used with closed loop option) The encoder position saved at power down is restored. X absolute position = encoder position. Y absolute position = 0. General purpose registers are set to zero on power up.
		3 -	X and Y absolute position = 0. General purpose registers saved at power down are restored.
		4 -	The X and Y absolute positions saved at power down are restored. General purpose registers saved at power down are restored.
		5 -	(used with closed loop option) The encoder position saved at power down is restored. X absolute position = encoder position. Y absolute position = 0. General purpose registers saved at power down are restored.
	---n	0 -	Power up feedrate = L09 value.
		1 -	Power up feedrate = last index feedrate

Default

L20 02000

TR1 X axis positive edge
TR2 Y axis positive edge
Expansion port - BCD switches
X and Y abs. position = 0
Power up feedrate = L09
General purpose registers are set to zero on power up.

Notes

When a mechanical home or registration cycle is executed, the input signal source is determined by checking the assignment for TR1 first, then TR2. For example if TR1 is X positive and TR2 is X negative, TR1 X positive will be used.

Positive Edge Trigger definition:
Input transitions from inactive to active.
Negative Edge Trigger definition:
Input transitions from active to inactive.

The expansion port can be interfaced to BCD switches or expansion I/O, not both.

Related parameters

L58 (Input Configuration)
L59 (Output Configuration)

L26 Transmission Protocol

Function Selects the serial communication protocol

Command Format L26 n

Range n 0 - 7 (see below)

Default 0 (normal transmission mode, Xon/Xoff protocol)

Notes

Xon/Xoff Protocol Enabled

- n=0 normal transmission mode (no "EOT" or "=" characters)
- n=1 "EOT" follows each complete data transmission
- n=2 "=" is transmitted when ready for more commands
- n=3 "EOT" follows each complete data transmission and "=" is transmitted when ready for more commands

Xon/Xoff Protocol Disabled

- n=4 normal transmission mode (no "EOT" or "=" characters)
- n=5 "EOT" follows each complete data transmission
- n=6 "=" is transmitted when ready for more commands
- n=7 "EOT" follows each complete data transmission and "=" is transmitted when ready for more commands

EOT: ASCII 04 decimal.

see Xon/Xoff description for details

L41 Program Start Line Number

Function Line number from which program execution starts following a Power up, a Clear command, execution of an End of Program command or the occurance of a program error condition.

Command Format L41 nnn

Range nnn 0 - 400 line number

Default line 1

Example L41 100 Program Starts at line 100.

L43 Backlash Compensation Delay

Function Delay between an index move and its backlash cycle. Also the delay between the jog and index portions of a mechanical home cycle.

Command Format L43 nnnn

Range nnnn 0 - 9,999 milliseconds

Default 50 milliseconds

Notes The delay time should be sufficient to allow the motor to settle before the backlash correction move begins.

Related parameters L66 (Backlash Compensation)

Example L43 100 Sets Backlash Compensation Delay to 100 milliseconds.

L44 Delay After Motion

Function Delay that occurs following motion.

Command Format L44 nnnn

Range nnnn 0 - 9,999 milliseconds.

Default 50 milliseconds

Notes The delay time should be sufficient to allow the motor to settle before the next move begins.

Example L44 100 Sets Delay After Motion to 100 milliseconds.

L47 Program Execution Repeat Count

Function Number of times a program will be executed in the Complete Execution Mode. (L06 nn2)

Command Format L47 nnnn

Range nnnn 0 - 9999 times

Default 0 times

Notes This parameter only applies in the Complete Execution Mode (L06 nn2). In this mode, the program is executed once plus L47 times.

Related parameters L06 (Program Execution Format)

Example L47 50 Sets the Program Execution Repeat Count to 50.

L48 Program Line Count List/Clear

Function Number of program lines to be listed or cleared in response to serial port commands H14 and H12.

Command Format L48 nnn

Range: nnn 0 - 400 lines

Default: 20 lines

Notes: Starting at the present line number, transmit via the serial port the specified number of programmed lines in response to an H14 (List) command.

Starting at the present line number, clear the specified number of consecutive program lines in response to an H12 command. (clear)

If L48 = 0 then all lines will be listed or cleared.

Related parameters: L49 (First Parameter to List)
L50 (Parameter List Count)

Examples L48 0 All programmed lines listed or cleared.

L49 First Parameter to List
L50 Parameter List Count

Function

Control the listing of parameters in response to serial port command H16 (transfer parameters).

Command Format

L49 nn
L50 nn

Range

nn 0, valid L code number
nn 0 - 99

Default

L49 = 0	List all parameters, 3 per
L50 = 0	line format.

Notes

Attempts to program L49 with a non-existent parameter number, other than 0, are ignored.

The response to a transfer parameter (H16) command is:

L49 = 0 L50 = 0	List all parameters, three per line format. (3 columns)
L49 = 0 L50 non zero	List all parameters, one per line format. (1 column)
L49 non zero L50 = 0	List single parameter specified by L49.
L49 non zero L50 non zero	List L50 parameters, starting with the parameter specified by L49.

Related parameters

L48 (Program Line Count List/Clear).

Example

L50 = 4, L49 = 9 The response to an H16 command is:

L09 nnnnnn,nnnnnn
L10 nnnnnnnn,nnnnnnnn
L11 nnnnnnnn,nnnnnnnn
L12 nnnnnn,nnnnnn

L52 Buffer Full Warning Message

Function

Defines the message that is transmitted when either buffer is within 10 characters of overflowing.

Command Format

L52 "aaa"

Range

ASCII characters from 32 to 127 decimal except for the following:
" decimal 34
$ decimal 36
* decimal 42
< decimal 60

Maximum message length is three characters

Default

L52 "" (null string, no message sent)

Notes

Quotes must be used when programming L52.

The message is transmitted only if the unit's ID is active.

The message will be transmitted via the serial port when either buffer is within 10 characters of overflowing. A "CR LF" is transmitted following the message.

If the message is the null string (""), then nothing, not even "CR LF", is transmitted.

Example

L52 "Buf" Sets "BufCRLF" as the characters to be transmitted when either buffer is near to overflowing.

L53 Following Error Warning Message

Function

Defines the message that is transmitted when the position error exceeds the following error (L87).

Command Format

L53 "aaa"

Range

ASCII characters from 32 to 127 decimal except for the following:
" decimal 34
$ decimal 36
* decimal 42
< decimal 60

Maximum message length is three characters

Default

L53 "" (null string, no message sent)

Notes

This L code is only available with the closed loop option.

Quotes must be used when programming L53.

The message is transmitted only if the unit's ID is active.

The message will be transmitted via the serial port when the position error, during motion, exceeds the following error (L87) value. A "CR LF" is transmitted following the message.

If the message is the null string (""), then nothing, not even "CR LF", is transmitted.

Example

L53 "Fol" Sets "FolCRLF" as the characters to be to be sent transmitted when a following error occurs.

L54 Unable To Reach Position Warning Message

Function

Defines the message that is transmitted when the position error is excessive and could not be corrected.

Command Format

L54 "aaa"

Range

ASCII characters from 32 to 127 decimal except for the following:
" decimal 34
$ decimal 36
* decimal 42
< decimal 60

Maximum message length is three characters

Default

L54 "" (null string, no message sent)

Notes

This L code is only available with the closed loop option.

Quotes must be used when programming L54.

The message is transmitted only if the unit's ID is active.

The message will be transmitted via the serial port when excessive position error could not be corrected. A "CR LF" is transmitted following the message.

If the message is the null string (""), then nothing, not even "CR LF", is transmitted.

Example

L54 "Utc" Sets "UtcCRLF" as the characters to be transmitted when the excessive position error can not be corrected.

L55 Line Done Message

Function Defines the message that is transmitted when a program line is completed.

Command Format L55 "aaa"

Range ASCII characters from 32 to 127 decimal except for the following:
" decimal 34
$ decimal 36
* decimal 42
< decimal 60

Maximum message length is three characters

Default L55 "" (null string, no message sent)

Notes Quotes must be used when programming L55.

The message is transmitted only if the unit's ID is active.

The message will be transmitted via the serial port upon completion of a program line (line 1 thru 400). A "CR LF" is transmitted following the message.

If the message is the null string (""), then nothing, not even "CR LF", is transmitted.

Example L55 "Lin" Sets "LinCRLF" as the characters to be to be sent transmitted when a program line is completed.

L56 Program Done Message

Function Defines the message that is transmitted when a the program is completed.

Command Format L56 "aaa"

Range ASCII characters from 32 to 127 decimal except for the following:
" decimal 34
$ decimal 36
* decimal 42
< decimal 60

Maximum message length is three characters

Default L56 "" (null string, no message sent)

Notes Quotes must be used when programming L56.

The message is transmitted only if the unit's ID is active.

The message will be transmitted via the serial port upon completion of a the program. A "CR LF" is transmitted following the message.

If the message is the null string (""), then nothing, not even "CR LF", is transmitted.

Example L56 "Prg" Sets "PrgCRLF" as the characters to be to be transmitted when the program has been executed.

L58 Input Configuration

Function

Assigns specific functions to the 13 configureable inputs and selects the logic convention for both inputs and outputs.

Command Format

L58 "shhhhhhhhhhhhh"

Range

s + or - (logic convention)
h 0 - 9, A, B, C, D, E, F (function designator)

Default

+2187600000000 positive logic
In13 CCW limit X axis
In12 CW limit X axis
In11 Step/Jog
In10 CCW direction
In9 CW direction
In1-8 Program Testable Inputs

Notes

The sign determines the logic convention for the inputs and outputs (+ = positive, - = negative). With positive logic, an active input or output is denoted by a logic 1.

With negative logic, an active input or output is denoted by a logic 0.

Each digit in the L58 field represents a specific hardware input. The value programmed for a given digit determines the function assigned to that corresponding input.

```
L58   "shhhhhhhhhhhhh"
       s------------- Logic convention + pos, - neg.
       -h------------ Function for hardware Input 13
       --h----------- Function for hardware Input 12
       ---h---------- Function for hardware Input 11
       ----h--------- Function for hardware Input 10
       -----h-------- Function for hardware Input 9
       ------h------- Function for hardware Input 8
       -------h------ Function for hardware Input 7
       --------h----- Function for hardware Input 6
       ---------h---- Function for hardware Input 5
       ----------h--- Function for hardware Input 4
       -----------h-- Function for hardware Input 3
       ------------h- Function for hardware Input 2
       -------------h Function for hardware Input 1
```

Descriptions

Function 0 — Program Testable Input
Input state is checked when G20 (branch) or G22 (wait) command is executed.

Function 1 — Clockwise Travel Limit X Axis
When this input is activated, X axis CW motion is stopped immediately, further CW motion is not allowed while this input is active.

Function 2 — Counterclockwise Travel Limit X Axis
When this input is activated, X axis CCW motion is stopped immediately, further CCW motion is not allowed while this input is active.

Function 3 — Feed Hold
When this input is active motion comes to a controlled stop, if uncompleted, the remainder of the move will be finished when a cycle start is commanded.

Function 4 — Cycle Start
When this input transitions from inactive to active program execution starts or resumes from a Feed Hold state.

Function 5 — Cycle Stop
When this input becomes active, program execution stops at the completion of the current line.

Function 6 — Clockwise Direction (manual control)
Initiates Step or Jog motion in the CW direction. The commanded motion type is determined by the Step/Jog mode status of the active axis. Motion is initiated on the inactive to active transition of this input. The input must remain active to maintain Jog motion.

This input is ignored during program execution.

Function 7 — Counterclockwise Direction (manual control)
Initiates Step or Jog motion in the CCW direction. The commanded motion type is determined by the Step/Jog mode status of the active axis. Motion is initiated on the inactive to active transition of this input. The input must remain active to maintain Jog motion.

This input is ignored during program execution.

Function 8 — Step/Jog (manual control)
Sets the Step/Jog mode of the active axis. an inactive to active input transition selects Step Mode and an active to inactive transition selects Jog Mode.

This input is ignored during program execution, while motion is occurring or when in a Feed Hold state.

Function 9 — Low/High
Sets the High/Low speed mode of the active axis. When this input transitions from inactive to active, Low Speed Mode is selected, an active to inactive transition selects High Speed Mode.

Function A — Motor Windings Off/On
Sets the Windings Off/On mode of the active axis. When this input transitions from inactive to active, Windings Off Mode is selected, an active to inactive transition selects Windings On Mode.

This input is ignored during program execution, while motion is occurring or when in a Feed Hold state.

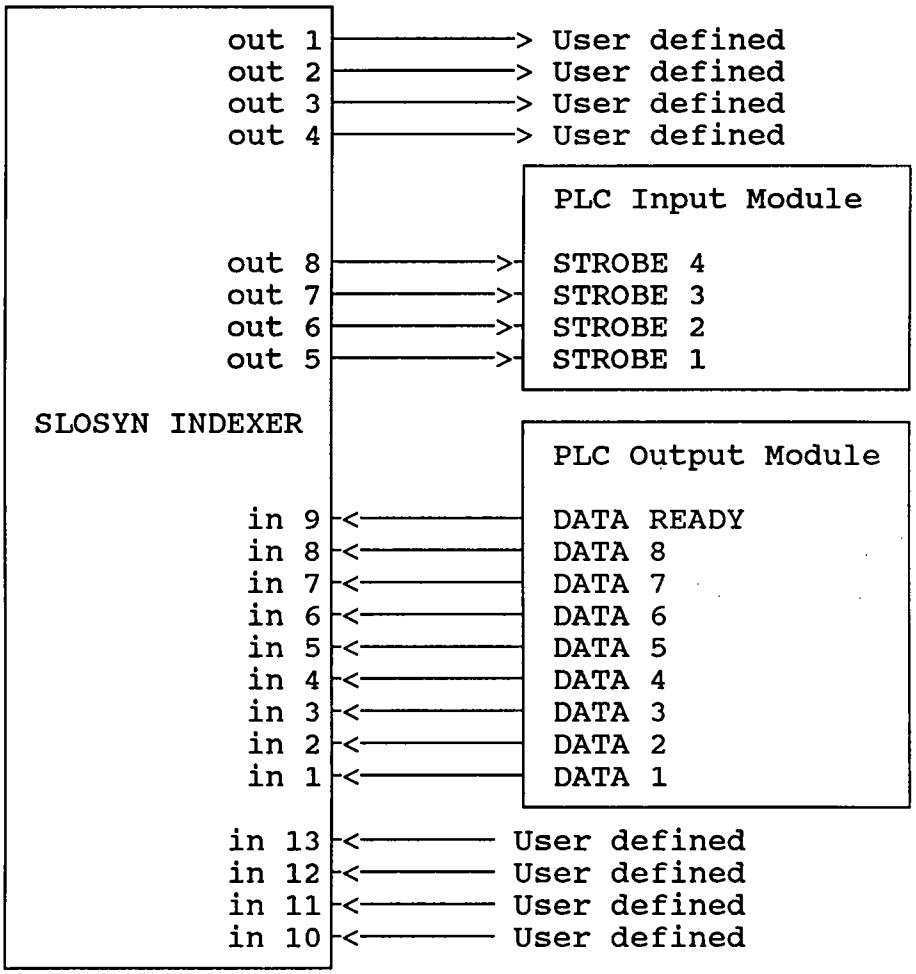
Function B — PLC DATA READY
Used for handshaking when reading data from a PLC. When active indicates that the PLC data is valid. This function can only be assigned to IN9.

Function C — PLC DATA
Reading PLC data requires 8 data inputs. The data inputs can only be assigned to IN1 - IN8. The data is coded as 2 BCD digits. The msb is IN8.

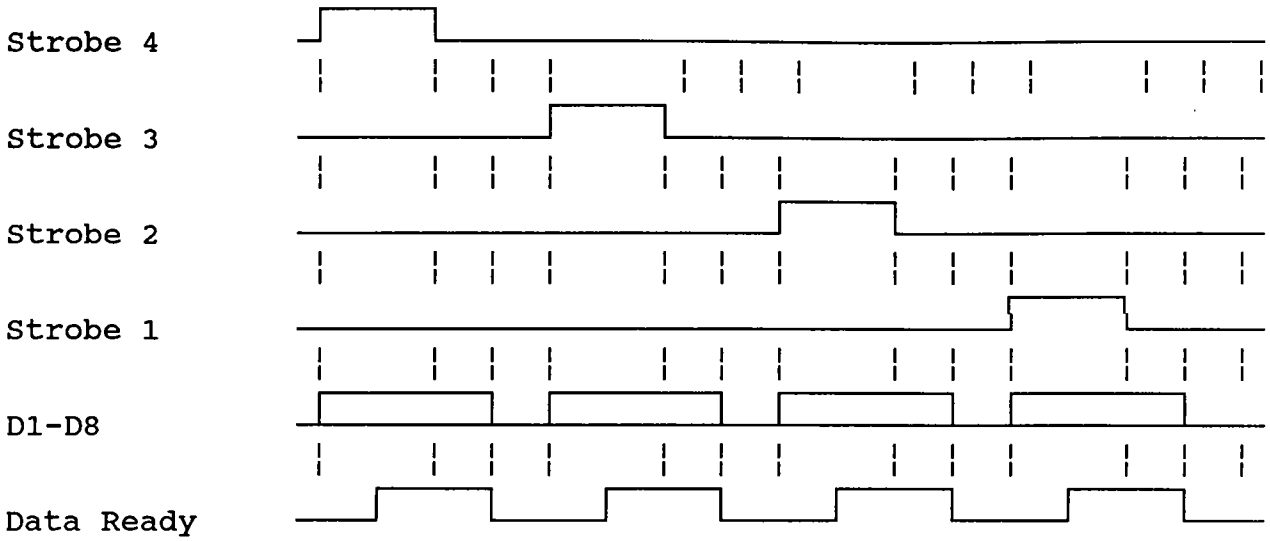
Function D — Y axis/ X axis select (manual control)
Selects the active axis. When this input transitions from inactive to active, Y becomes the active axis, an active to inactive transition selects X as the active axis. This input is ignored during program execution, while motion is occurring or when in a Feed Hold state.

Function E — Clockwise Travel Limit Y Axis
When this input is activated, Y axis CW motion is stopped immediately, further CW motion is not allowed while this input is active.

Function F — Counterclockwise Travel Limit Y Axis
When this input is activated, Y axis CCW motion is stopped immediately, further CCW motion is not allowed while this input is active.



PLC INTERFACE BLOCK DIAGRAM
FIGURE L-1



PLC TIMING DIAGRAM
FIGURE L-2

PLC Handshaking Sequence

1)	Control	Asserts Strobe 4 (OUT8)
2)	PLC	Outputs data (Sign, 10M's & 1M's) and asserts Data Ready
3)	Control	Reads data then removes Strobe 4
4)	PLC	Removes Data Ready and data
5)	Control	Asserts Strobe 3 (OUT7)
6)	PLC	Outputs data (100K's & 10K's) and asserts Data Ready
7)	Control	Reads data then removes Strobe 3
8)	PLC	Removes Data Ready and data
9)	Control	Asserts Strobe 2 (OUT6)
10)	PLC	Outputs data (1K's & 100's) and asserts Data Ready
11)	Control	Reads data then removes Strobe 2
12)	PLC	Removes Data Ready and data
13)	Control	Asserts Strobe 1 (OUT5)
14)	PLC	Outputs data (10's & 1's) and asserts Data Ready
15)	Control	Reads data then removes Strobe 1
16)	PLC	Removes Data Ready and data

PLC data Strobe Assignments

Strobe	In #8	In #7	In #6	In #5	In #4	In #3	In #2	In #1
Out 1	80	40	20	10	8	4	2	1
Out 2	8,000	4,000	2,000	1,000	800	400	200	100
Out 3	800,000	400,000	200,000	100,000	80,000	40,000	20,000	10.000
Out 4	- sign	40,000,000	20,000,000	10,000000	8,000,000	4,000,000	2,000,000	1,000,000

Related Parameters L20 (System Configuration)
L59 (Output Configuration)

Example:

L58 "-FE21BCCCCCCCC" Negative logic convention

Input #13,	Y axis CCW Travel Limit
Input #12,	Y axis CW Travel Limit
Input #11,	X axis CCW Travel Limit
Input #10,	X axis CW Travel Limit.
Input #9,	PLC Data Ready
Input #1-#8,	PLC Data

L58 "+3456789AD0000" Positive logic convention

Input #13,	Feedhold
Input #12,	Cycle Start
Input #11,	Cycle Stop
Input #10,	CW direction (manual control)
Input #9,	CCW direction (manual control)
Input #8,	Step/Jog
Input #7,	Low/High
Input #6,	Motor Windings On/Off
Input #5,	X axis/ Y axis select
Inputs #1-#4,	program testable inputs

L59 Output Configuration

Function Assigns specific functions to the 8 configurable outputs. See "Descriptions" below.

Command Format L59 "hhhhhhhh"

Range h 0 - 9, A, B, C, D, E, F (function designator)

Default 00000000 Out1 - 8 are programmable outputs

Notes Each digit in the L59 field represents a specific hardware output. The value programmed for a given digit determines the function assigned to that corresponding output.

L59 "hhhhhhhh"
h------ Function for hardware Output #8
-h----- Function for hardware Output #7
--h---- Function for hardware Output #6
---h--- Function for hardware Output #5
----h-- Function for hardware Output #4
-----h- Function for hardware Output #3
------h Function for hardware Output #2
-------h Function for hardware Output #1

Descriptions

Function 0 Programmable Output set by user program (G47 code).
This output is inactive on Power-On or following a "Clear" command.

Function 1 Programmable Output set by user program (G47 code).
This output is active on Power-On or following a "Clear" command.

Function 2 Motion busy
This output is active if either the X or Y axis is moving.

Function 3 System busy
This output is active when either axis is moving or program execution is occurring.

Function 4 At hard limit
This output is active if motion was terminated or inhibited by a travel limit and the source of that limit was a hardware input.

Function 5 At clockwise limit
This output is active if motion was terminated or inhibited by a CW travel limit. The travel limit can be from an input or the CW software travel limit (L18).

Function 6 At counter clockwise limit
This output is active if motion was terminated or inhibited by a CCW travel limit. The travel limit can be from an input or the CCW software travel limit (L19).

Function 7 At home position (electrical zero)
This output is active if the selected axis is at absolute position 0.

Function 8 Position error (closed loop option)
This output is active when the magnitude of the position error exceeds the position deadband (L93) value at standstill.

Function 9 X axis active
This output is active when the X axis is moving or selected.

Function A — Y axis active
This output is active when the Y axis is moving or selected.

Function B — PLC Strobe Output
Reading PLC data requires 4 strobes, one for each byte (2 BCD digits) of data. When used, this function must be assigned to Outputs #5-#8.

Function C — Fault
This output is active when the fault lamp is on.

Function D — Following error (closed loop option)
This output is active when the magnitude of the position error exceeds the following error (L87) value during motion.

Function E — Closed Loop Correction Cycle (closed loop option)
This output is active during the motion cycle that attempts to correct a following error.

Function F — Unable to Correct (closed loop option)
This output is active when all the allotted (L96) attempts to correct the position or following error have failed.

Related parameters L20 (System Configuration)
L58 (Input Configuration)

Example

L59 "BBBB0000"	Outputs #5-#8, PLC Strobes Outputs #1-#4, Programmable Outputs, inactive on Power-On or Clear.
L59 "11110000"	Outputs #5-#8, Programmable Outputs, active on Power-On or Clear. Outputs #1-#4, Programmable Outputs, inactive on Power-On or Clear.

L66 Backlash Compensation

Function Move distance to compensate for backlash.

Command Format

L66 snnnnnnnn,snnnnnnnn	Set X and Y axis backlash compensation distance.
L66 snnnnnnnn	Set X axis backlash compensation distance.
L66 ,snnnnnnnn	Set Y axis backlash compensation distance.

Range snnnnnnnn -99,999,999 to +99,999,999 pulses

Default +0,+0 (disabled)

Notes Backlash compensation is not performed with Mark Registration, step or jog motion.

Backlash Compensation is only performed when the move direction and the backlash compensation (L66) sign are the same. When this occurs, the motor is rotated past its target position by the backlash compensation distance. Following a delay time (L43) the motor is rotated back, in the opposite direction, the backlash compensation distance.

Related parameters L43 (Backlash Compensation Delay)

Examples

L66 +100,+100	Sets X and Y backlash compensation distance to 100 pulses in the CW direction.
L66 -100	Sets X axis backlash compensation distance to 100 pulses in the CCW direction.

L71 Maximum Speed

Function Maximum speed for all motion.

Command Format L71 nnnnnn

Range nnnnnn 1 - 999,999 pulse/sec.

Default Value equivalent to 10,000 Full Steps/sec, for microstepping resolution up to 100. Exact number depends on selected microstepping resolution. See Default table in Appendix A.

Notes

1) The maximum speed value applies to both X and Y. The speed of any motion is not allowed to exceed this value.

2) For best motor performance, set L71 as low as possible. When L71≤ 115,000, an internal pulse-smoothing algorithm is enabled. Therefore, when using drives with resolutions of full, half, 1/5, and 1/10 step it is best to keep L71 set no higher than 115,000.

Example L71 20000 Sets the maximum speed to 20,000 pulses/sec.

L72 Velocity Profile

Function Selects the velocity profile used during all motion.

Command Format L72 n

Range
0 = trapezoidal profile
1 = hyperbolic profile
2 = "S" shaped profile

Default: L72 0 (trapezoidal profile)

Notes The velocity profile determines the manner in which the accel/decel varies with speed, refer to illustrations of each profile.

Trapezoidal profile - The accel/decel is constant.

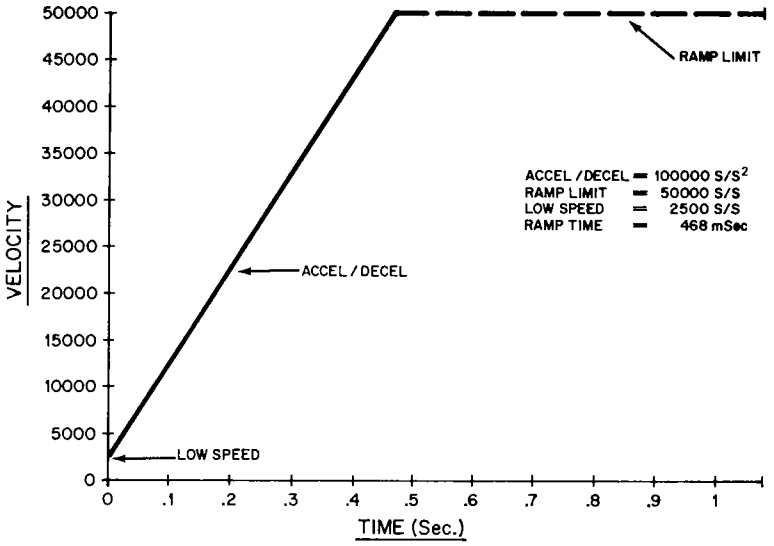
Hyperbolic profile - The accel/decel is at its maximum value at low speed and decreases to its minimum value at high speed.

"S" profile - The accel/decel is at its minimum value at low speed, increases to its maximum and then decreases back to its minimum value at high speed.

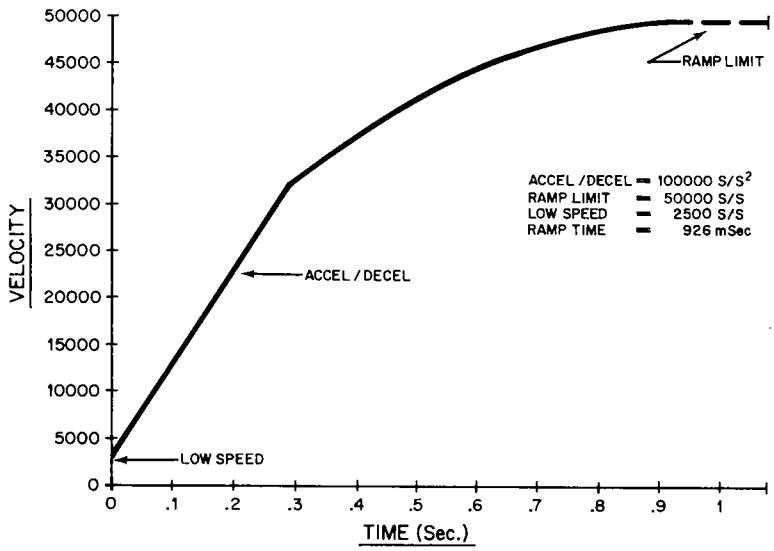
With some loads, a variable acceleration ("S" or Hyperbolic) will permit the load to achieve higher speeds than would be possible with constant acceleration (Trapezoidal).

For a given accel (L11) or decel (L10), the trapezoidal profile results in the shorter accel/ decel times.

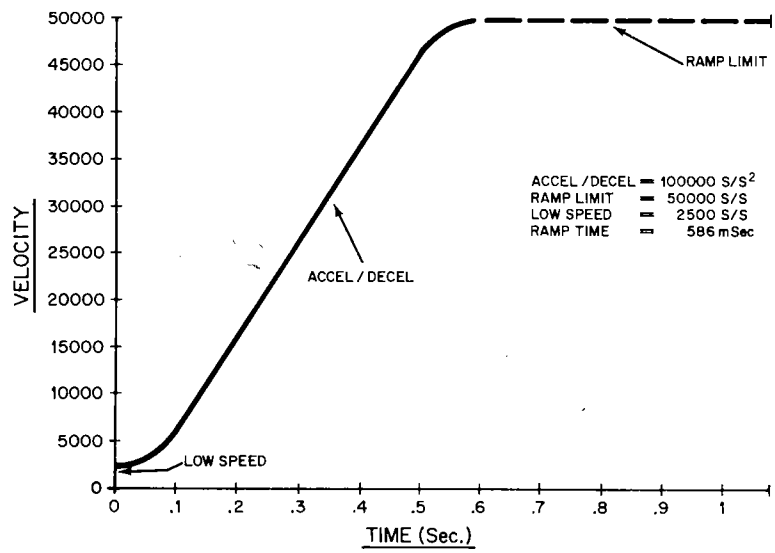
Related parameters
L10 (Deceleration)
L11 (Acceleration)



TRAPEZOIDAL PROFILE



HYPERBOLIC PROFILE



"S" PROFILE

L73 Speed Increment

Function Size of the speed change that occurs in response to serial port commands Target Velocity Increase (H31) and Target Velocity Decrease (H32).

Command Format L73 nnnnnn

Range nnnnnn 1 - 999,999 pulses/sec.

Default Value equivalent to 100 Full Steps/sec, exact number depends on selected microstepping resolution. See Default table in Appendix A.

Notes The speed can not be increased above the maximum speed (L71) or decreased below 0.

Example L73 100 Sets the speed increment to 100 pulses/sec.

L91 External Move Distance Scale Factor

Function Scale factor used with external BCD move data.

Command Format

L91 nnnnnnnn	(scale factor for X axis)
L91 ,nnnnnnnn	(scale factor for Y axis)
L91 nnnnnnnn,nnnnnnnn	(scale factors for X and Y)

Range nnnnnnnn 0 - 99999999

Default 00010000,00010000 (X and Y scale factor = 1.0000)

Notes The scale factor has 4 fractional and 4 whole digits. (10000 = 1.0000).

The scale factor is only used with the following G codes:

- G10 (Mark Registration Cycle)
- G36 (Read Move Distance)
- G79 (Set Absolute Position)

When one of the above codes is executed and the distance is not specified, the external data is read and multiplied by the scale factor to form the distance.

Examples L91 105000,15000 X axis scale factor = 10.5000
Y axis scale factor = 1.5000

L98 Delay Between Automatic Transmissions

Function Sets the time between repeated transmissions of data called for by continuous (%) H code commands.

Command Format L98 nnnn

Range nnnn 0 - 9,999 milliseconds

Default 100 milliseconds

Notes The continuous H code commands are:

%H13 Transfer Current Program Line
%H15 Transfer Current Line Number
%H17 Transfer Active Axis Absolute Position
%H19 Transfer Mode Status
%H20 Transfer Output Status
%H21 Transfer Input Status
%H22 Transfer Encoder Position
%H26 Transfer Error Count
%H27 Transfer Position Verification Status
%H60 Transfer Present Velocity
%H88 Transfer Execution Status

The remaining L codes pertain to the closed loop option and its operation.

L87 Following Error

Function Magnitude of position error, while motion is occurring, that signals a following error.

Command Format L87 nnnn

Range nnnn 0 - 9,999 pulses

Default Value equivalent to 10 Full Steps, exact number depends on selected microstepping resolution. See Default table in Appendix A.

Notes Closed loop operation is only available on the X axis.

The Following Error should be greater than four full steps (Drive Resolution/50).

200/50 = 4 (four full steps)
25000/50 = 500 (four full steps)

Related parameters L93 (Position Deadband)

Example L87 200 Sets the Following Error at 200 pulses.

L90 Closed Loop Configuration

Function Sets the closed loop operating mode.

Command Format L90 n

Range n 0 - 3

Default 0 (Open Loop Mode)

Notes The Encoder position and the position error are calculated in all operating modes. The encoder position can be examined with a H22 command.

A Following Error occurs when the magnitude of the position error exceeds the Following Error (L87) during motion.

L90=0 Encoder feedback has no effect on operation. (Open Loop Mode)

L90=1 Position maintenance is enabled. When a Following Error occurs, motion is terminated and the fault light turned on.

L90=2 Position maintenance is enabled. When a Following Error occurs, motion is terminated, then restarted in an attempt to complete the original move. If after L96 attempts, the move is not completed, the cycle is terminated and the fault light turned on. Position Verification Status will indicate an "unable to correct" condition.

L90=3 Position maintenance is enabled. When a Following Error occurs, indexed motion is terminated, the motor is returned to the position from which it started, then the original move retried. Up to L96 attempts are allowed for returning to the starting position and also for completing the original move. If either the return to starting position or subsequent completion of the original move is unsuccessful, the active cycle is terminated and the fault light turned on. Position Verification Status will indicate an "unable to correct" condition.

A Following Error occurring during jog or any motion without a target position, will be handled in the same manner as L90=2 mode.

Position maintenance

Maintenance is only active while motion is not being commanded (standstill). If the magnitude of the position error exceeds the Position Deadband (L93), the motor is moved so as to produce zero position error.

Attempts to correct the position error will be limited to the Position Corrections Attempts (L96) number, at which point the fault light is turned on. Position Verification Status will indicate an "unable to correct" condition.

L93 Position Deadband

Function Magnitude of position error, at standstill, that signals a position error.

Command Format L93 nnnn

Range nnnn 0 - 9,999 pulses

Default see default table Appendix A.

Note If L95 (Encoder Quadrature Scale Factor) is set to 5, the L93 value should be a multiple of 5.

Related parameters L87 (Following Error)

Example L93 1 Sets the position deadband to 1 pulse.

L94 Encoder Counter Direction

Function Sets the encoder position counter direction (up or down) vs. encoder output Ch.A & Ch.B phasing.

Command Format L94 n

Range
n = 0 count up if Ch.A leads Ch.B
n = 1 count down if Ch.A leads Ch.B

Default 1 (count down, Ch.A leads Ch.B)

Notes For the closed loop to work, the encoder position and X absolute position counters must both count in the same direction.

Related parameters L95 (Encoder Scale Factor)

Example L94 1 count down if, Ch.A leads Ch.B

L95 Encoder Scale Factor

Function Scales the encoder counts/rev. to match the motor pulses/rev.

Command Format L95 n

Range
0 = no scaling
2 = divide encoder counts/rev by 2
4 = divide encoder counts/rev by 4
5 = multiply encoder counts/rev by 5
8 = divide encoder counts/rev by 8

Default see default table Appendix A.

Related parameters L94 (Encoder Counter Direction)

Examples A 400 line Encoder (1600 pulses/rev) is used with a half step drive (400 pulses/rev). The scale factor required is divide by 4. (L95 = 4)

A 1250 line Encoder (5000 pulses/rev) is used with a 1/125 microstepping drive (25000 pulses/rev). The scale factor required is multiply by 5. (L95 = 5)

L96 Position Correction Attempts

Function Number of consecutive correction attempts allowed for Position Maintenance or for the correction of a Following Error.

Command Format L96 nnnn

Range nnnn 0 - 9,999 attempts

Default 100 attempts

Notes When the number of attempts to correct a position error equals the L96 value, an unable to correct flag is set in Position Verification status. No further attempts will be allowed until a new motion is started.

Related parameters L97 (Delay Between Correction Attempts)

Example L96 200 Sets number of correction attempts to 200.

L97 Delay Between Correction Attempts

Function Delay between correction attempts for Position Maintenance or Following Error correction.

Command Format L97 nnnn

Range nnnn 0 - 9,999 milliseconds

Default 100 milliseconds

Notes When a following error or a motion clear occurs, the encoder position is transferred to the X absolute position, L97 milliseconds later.

Related parameters L96 (Position Correction Attempts)

Example L97 1000 Sets the delay between correction attempts to 1000 milliseconds.

This page intentionally left blank