Section 5: Subroutines & Subprograms

Subroutines

Subroutines are used for contours, hole patterns, or any actions that repeat or are used in many locations. Typically subroutines will contain only positional moves. Feed rates, tool changes, spindle speeds, rotation, and other codes are reserved for the main program. However, most codes can be in a subroutine.

All user defined subroutines must be at the beginning of the program before the main section of the program. Only the O word and comments may be used prior to the first subroutine. **Subroutines cannot be defined in a subprogram.** However, if the START macro command is used, subroutines can be defined in subprograms (see Section Eighteen, Macros).

Beginning a Subroutine

The format of the L word for a subroutine definition is LNNKK.

NN is the subroutine number (01-89).

KK will always be 00 (zero, zero).

EXAMPLE: *L0100 (or L100) This would define the beginning of subroutine number 1*
L2300 This would define the beginning of subroutine number 23

The maximum number allowed for NN is 89. Subprograms 90 - 99 are used by the control for Fixed subroutines (see Section Six, Fixed Subroutines). The line with the L word that defines the subroutine can only have a parenthesis or an asterisk for a comment. No other codes are permitted.

Calling a Subroutine

The format for the subroutine call is LNNKK. NN is the subroutine number (01-99). KK is the number of repetitions (01-99).

EXAMPLE: *L101 This would call, or use, subroutine number 1, one time*
L2315 This would call, or use, subroutine number 23, fifteen times

The LNNKK word must be the only word in the block in which it appears with the exception of R parameter definitions, G66, and a parenthesis or an asterisk for a comment. After a subroutine has been executed, it will return to the line where it was called and the program will continue from that line.

Ending a Subroutine

A subroutine ends with the L word that starts the next subroutine or with an M17. The M17 must be the only word in the block in which it appears.

The last subroutine in the program MUST have an M17 coded at the end.

Main Program

An M30 marks the end of the subroutine section and the start of the main program. An M17 marks the end of the last subprogram, which must be on a line before an M30. The M30 must be the only word in the block in which it appears.

When the operator presses the auto button the control will process the program. The control will recognize the existence of subroutines by the L #00 at the beginning of the program. The control will then recognize the beginning of the main program by the M30 code. When the control is ready to run, the line just after the M30 will be the first line to appear on the screen of the pendent.

When the M30 is used to end the subroutine section, an M2 is used to end the program. At the end of the program, the M2 will cause the program to begin again at the line after the M30 code.

EXAMPLE:

N10 M17 This marks the end a sub
N11 M30
N12 G0 G90 S2000 M3 E1 X0 Y0 End of subroutine section
N13 H1 M7 Z.1 Beginning of the Main program
N14 L201 Sub #2 one time. When sub #2 is complete it will return here
N15 M5 M9 G80

Nesting

A subroutine may be called for execution from another subroutine. This is called subroutine "nesting."

Subroutines may be nested as many as seven deep. This means that at some point in a subroutine another subroutine can be called, and then from that subroutine another can be called and so forth up to seven times.

Subroutines cannot be defined in a subprogram. However if the START marco command is used, subroutines can be defined in subprograms (see Section Eighteen, Macros).

EXAMPLE:

```
N1 O1234 (SUBROUTINE EXAMPLE PROGRAM
N2 L100  This marks the beginning of sub #1
N3 X.5 Y.5
N4 X-.5
N5 G80
N6 M17  This marks the end a sub
N7 L200  This marks the beginning of sub #2
N8 G81 G99 R0+.1 Z-.1 F35.
N9 L101  Sub#1 is being called from sub #2. Sub #1 is nested inside sub#2
N10 M17  This marks the end a subroutine
N11 M30  End of subroutine section
N12 G0 G90 S2000 M3 E1 X0 Y0  Beginning of the Main program
N13 H1 M7 Z.1
N14 L201  Sub #2 one time. When sub #2 is complete it will return here
N15 M5 M9 G80
N16 G0 G49 G90 Z0
N17 M2  End of the Main program. In the auto mode the program will rerun
from line N12
```

EXAMPLE: *Drill and Tap 2 holes using subroutine to define positions.*

```
N1 O1  (SAMPLE PROGRAM
N2 L100  Define Subroutine 1
N3 X.5 Y.5
N4 X-.5
N5 G80
N6 M17  End Subroutine
N7 M30  End of Subroutine definition
N8 M6 T1
N9  Tool #1 drill
N10 G0 G90 S3500 M3 E1 X0 Y0
N11 H1 M7 Z.25  Start main program
```

```
  Drill cycle
N12 G81 G99 R0+.1 Z-.475 F20.
N13 L101  Call Subroutine 1
N14 N15 M6 T2  Tool #2 tap
N16 G0 G90 S600 M3 E1 X0 Y0
N17 H2 M7 Z.25
N18 G84 G98 R0+.1 Z-.5 F600. Q.05  Tap Cycle
N19 L101  Call Subroutine
N20 M5 M9
N21 G0 G49 G90 Z0
N22 E0 X0 Y0
N23 M6 T1
N24 M2
```

- Block N2 uses the L word to identify the beginning of the subroutine.
- Block N3 through N4 identify the X and Y locations.
- Block N6 uses M17 to define the end of the subroutine.
- Block N7 uses M30 to define the end of subroutine definition and the beginning of the main program.

Upon execution of the program, the CNC always begins processing from the first block. When the first block contains the L word, the CNC examines each following block, until the M17, M30 codes are encountered. The execution begins with the block following the M30.

The example program begins execution from block N8. Block N13 causes program execution from block N2 until the M17 is encountered at block N6.

After completing the subroutine call, the program execution is returned to the next block following the subroutine call (N14).

Parametric Programming

Generalized subroutines can be written with the use of subroutine parameters. In a generalized subroutine, the numerical value of the A, B, E, F, G, H, I, J, K, L, M, P, Q, R, S, T, X, Y, Z words need not be specified directly. Values that are to be determined at the time of the subroutine call are specified indirectly by the use of the parametric reference "R". There are ten parameters, R0 through R9.

X+R1 directs the CNC to take the current value of parameter R1 as the value for the X word. X-R1 directs the CNC to take the negative of the current value of parameter R1 as the value for the X word.

The values of the parameters are modal. They are modified by programming an R word in a line of code. For example:

R0+.137 defines the value of parameter R0 to +.137. This value is used by any R0 in the program until it is redefined.

In the example below, subroutine L100 is a generalized subroutine to create a “D” pattern. Block N7 of the example calls the subroutine with the parameters R0 and R1 set to 2.0 and 1.0 respectively.

EXAMPLE:

```
N1 L100  (DEFINE SUBROUTINE 1
N2 G1 Y+R0  (FIRST LEG OF “D” PATTERN
N3 G2 X+R1 Y-R1 J-R1
N4 G2 X-R1 Y-R1 I-R1  (BACK TO BEGINNING
N5 M17
N6 M30  (END OF SUBROUTINE DEFINITION
N7 L101 R0+2. R1+1.  (CALL SUBROUTINE 1, (EXECUTE 1 TIME
```

All R values are modal, and are not cleared at the beginning of a program. The values are cleared at power on and are zero until they are defined.

Indefinite Subroutine Repetitions

In some cases a subroutine needs to be repeated an indefinite number of times. This is accomplished by using a .1 extension at the end of a subroutine call.

N1 L100
N2 E1 X.45 Y-1.05
N3 G81 G99 R0+.1 Z-.75 F80. X.5 Y-1.
N4 X2.5
N5 G80
N6 Z1.
N7 E0 X0 Y0
N8 G4 P66000 The machine is in the Waiting state, spindle & coolant on
N9 M17
N10 M30
N11 G90 G0 S10000 E1 X.45 Y-1.05
N12 H1 Z1. M7
N13 L101.1 The .1 extension repeats sub #1 an indefinite number of times
N14 M2

Subprograms

Subprograms function for the same purpose as a subroutine. They can be used instead of subroutines. For program editing purposes a program that uses subroutines is easier to edit. The operator can edit both the main and subroutine section of the program without switching to another program. Editing a subprogram requires that the operator first switch to the subprogram, edit, then switch back to the main program. The main program must be currently active to execute the program.

Subprograms are generally used in Format 2 style programs. A main program is identified by the use of the M30 code at the end, which functions like the M2 would in a Format 1 style program. The subprogram is identified by using the M99 code at the end. The M99 functions as the point to either return to the beginning of the subprogram for a repeat, or to return to the line where the subprogram was called.

Subprograms cannot contain subroutines. However a subprogram can be called from a subroutine. If START macro command is used, subroutines can be defined in subprograms (see Section Eighteen, Macros).

EXAMPLE:

O1 (MAIN PROGRAM
G90 G0 G17 G80 G40 G49 Z0
M6 T1 (TOOL #1
S2000 M3 G54 X0 Y0

H1 D1 Z1. M8
G82 G99 R0+.1 Z-.25 F45.

M98 P2 L1 Call subprogram #2 one time. This is where the sub returns after execution
M5 M9
G80
G90 G0 G49 Z0
M6 T2 (TOOL #2
G90 G0 S2000 M3 G54 X0 Y0
H2 D2 Z1. M8
G83 G99 R0+.1 Z-2.1 F37. Q.3143

M98 P2 L1 Call subprogram #2 one time. This is where the sub returns to after execution
M5 M9
G80
G0 G90 G49 Z0
G59 X0 Y0
M30
O2 (SUBPROGRAM FOR HOLE LOCATIONS
X2. Y1.
X3. Y1.
M99 End of subprogram and return to main program

This page intentionally left blank.