COMPUTER NUMERICAL CONTROL OF MACHINE TOOLS

Laboratory for Manufacturing Systems and Automation
Department of Mechanical Engineering and Aeronautics
University of Patras, Greece

Dr. Dimitris Mourtzis

Associate professor

Patras, 2016

Chapter 12:
Do Loops and Subprograms

Table of Contents

Objectives of Chapter 12

- Describe a **do loop**

- Describe a **subprogram**
- Describe **nested loops**
- Write simple programs using **loops, subroutines and nested loops**

Do loops

- If an **operation is to be repeated over a number of equal steps**, it may be programmed in what is referred to as a **do loop**

- In a **do loop**, **the Machine Control Unit (MCU) is instructed to repeat an operation** (in this case, drill a hole five times) rather than be programmed for five separate hole locations

- A **do loop** simply instructs the **MCU** to **repeat a series of NC program statements a specified number of times**

Do loops

- Usually, **looping capability** on a CNC controller is an **optional item**, therefore **not all controllers have it**

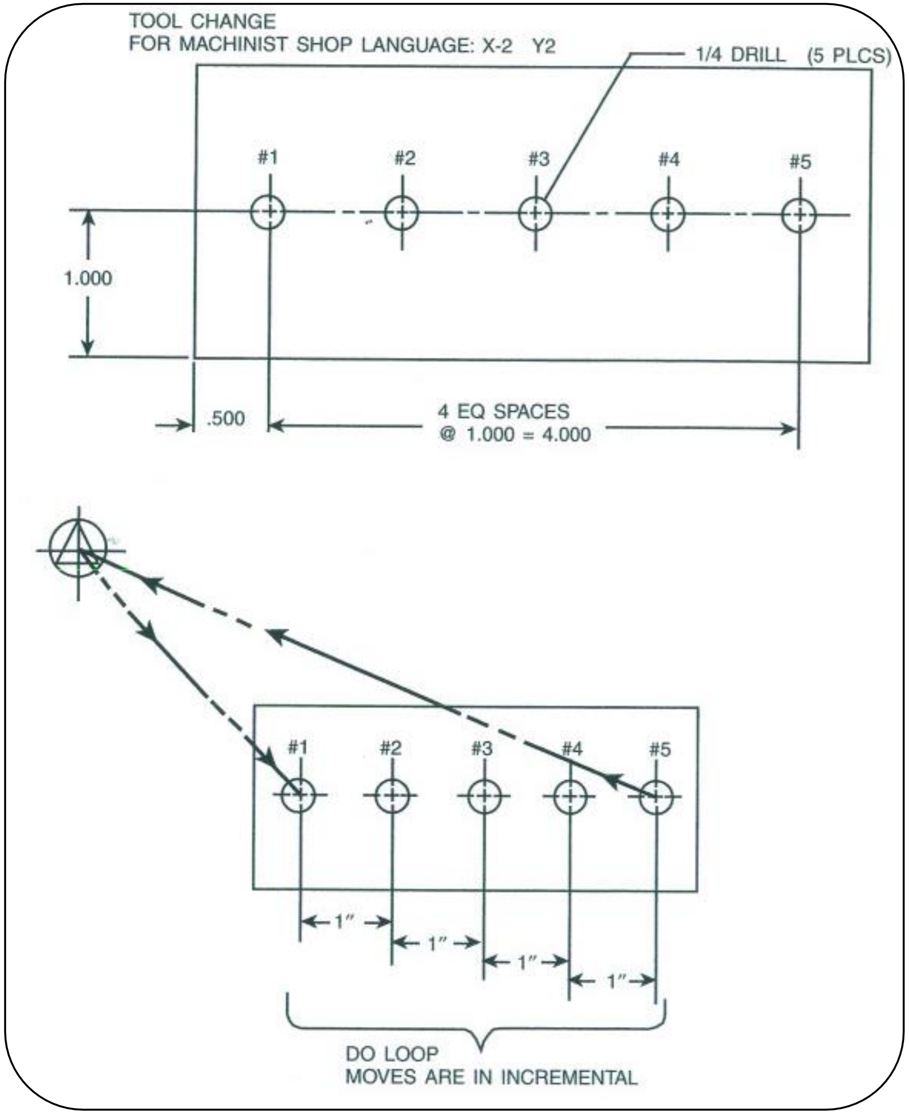
- The **looping feature is sometimes added to the controller** by the controller's manufacturer

- In other cases, it is **programmed into the controller** by the machine tool manufacturer

NOTE

The **NC codes used to initiate a do loop can vary** widely from machine to machine, even if they are all equipped with the same basic controller model

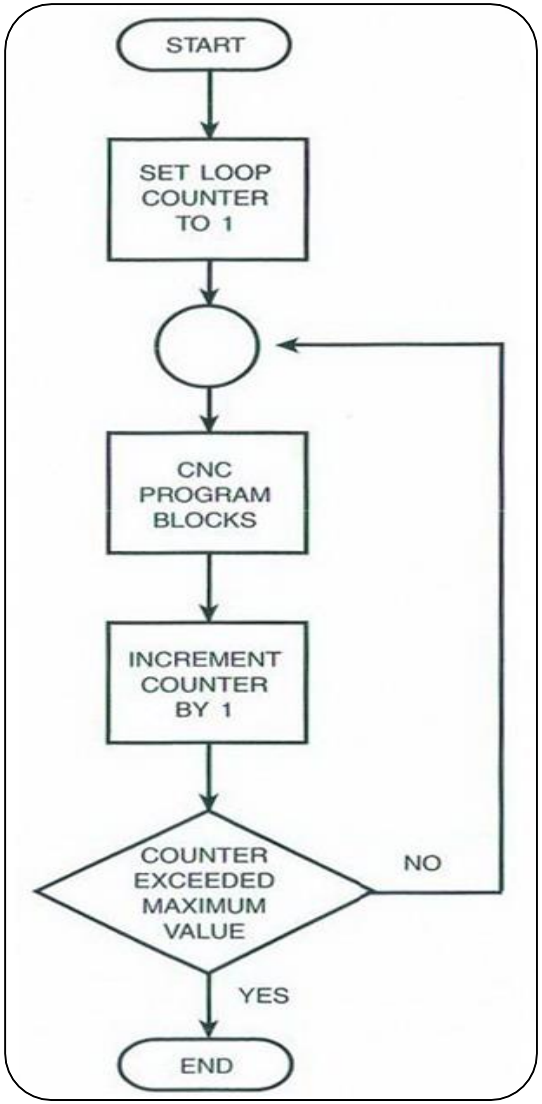
Do loops



- Figure 1 shows a part with a series of holes to be drilled, **equally spaced**

Figure 1: Part drawing and cutter path for do loop example *(Seams W., “Computer Numerical Control, Concepts & Programming”)*

Do loops



- The **flowchart** given in Figure 2 illustrates the basic **construct** of a **do loop**

Figure 2: Process flow of a do loop *(Seams W., "Computer Numerical Control, Concepts & Programming")*

Do loops

Programming a Loop

- Naturally, there is a **G code** to institute a **do loop**

- As mentioned previously, there are **no standard codes** for **do loops**

NOTE

The method described in this section **is only one of the schemes in use**

Seams W., “Computer Numerical Control, Concepts & Programming”

Do loops

```
%
O1103
(* **********)
(* X0/Y0 IS LOWER LEFT CORNER)
(* TOOL 1 - NO. 3 C-DRILL)
(* **********)
N001 G00 G40 G80 G70 G90 G98
N101 T01 M06
N102 G00 X.5 Y1. S3500 M03
N103 G43 Z0. H01 M08
N104G81 G99 X.5 Y1. Z-.162 R0. F7.
(BEGIN LOOP)
N105 G25 P106 Q106 L5
N106 G91 X1.
N107 G80 M09
N108 G00 G91 G30 Z0. M05
N109 G30 X0. Y0. M19
N110 M01
(* **********)
(* TOOL 2 - 1/4 DRILL)
(* **********)
N002 G00 G40 G80 G70 G90 G98
N201 T01 M06
N202 G00 X.5 Y1. S3500 M03
N203 G43 Z0. H01 M08
N204 G81 G99 X.5 Y1. Z-.375 R0. F7.
(BEGIN LOOP)
N205 G25 P206 Q20 6L5
N206 G91 X1.
N207 G80 M09
N208 G00 G91 G30 Z0. M05
N209 G30 X0. Y0. M19
N211 M30
%
```

Figure 3: Program to mill the part in Figure 1

- The **format** for a **do loop** is:

```
N…   G25 Pppp Qqqq Ll
Nppp X/Y/Z
N…   X/Y/Z
N…   X/Y/Z
Nqqq X/Y/Z
```

Where:

G25: Signals the start of a loop

P: Specifies the beginning block number of the loop

Q: Specifies the ending block number of the loop

L: Specifies the number of times to perform the loop

Do loops

Program Explanation

01103
Program number.
N001 through N103 is the tool change sequence N001
Program safety line.
N101
T01 — Places tool 1 in the tool change standby position. M06 — Initiates an automatic tool change.
N102
G00X.5Y1. — Positions the spindle at the X/Y starting position of the drill cycle at rapid traverse.
S3500M03 — Sets the spindle speed to 3,500 rpm, and turns on the spindle in a clockwise direction.
N103
G43 — Turns on tool length compensation.
Z0. — Positions the spindle .100 above the part at rapid.
H01 — Instructs the MCU to use the value in register 1 for tool length compensation.
M08 —Turns on the coolant.
N104 through N108 are the tool motion statements
N104
G81 —Turns on the canned drill cycle.
G99 — Instructs the MCU to return the spindle to the reference plane at the end of each cycle iteration.
X.5y1. — Y/Y coordinates of the first hole.
Z-.162 — The final feed depth for the Z axis.
R0. -Sets Z0 as the reference plane (feed engagement point)

Seams W., "Computer Numerical Control, Concepts & Programming"

Do loops

N105

G25 — Initiates the do loop. The loop counter is initialized to 1.

P106 — Tells the MCU to begin the loop in block N106.

Q106 —Tells the MCU that N106 is the end of the loop.

L5 — Instructs the MCU to perform the loop 5 times.

N106

G91 — Selects incremental positioning.

XL— Causes the spindle to move 1.000 inch along the X axis. Since the drill cycle is still turned on, a hole will be drilled at this location.

N108

G80 — Cancels the active drill cycle.

N108 through N109 are the tool cancel commands

N108

G00G91G30Z0. — Sends the Z axis to tool change position. M09 — Turns off the coolant.

N109

G30X0.Y0. — Sends the X and Y axes to tool change position. M19 — Orients the spindle for tool change gripper.

N110

M01 — Optional stop code.

N002 through N203 is the tool change sequence

N002

Program safety line. N201

T02 — Places tool 2 in the tool change standby position.

M06 — Initiates an automatic tool change,

Do loops

N202

G00X.5Y1. — Positions the spindle at the X/Y starting position of the drill cycle at rapid traverse.

S3500M03 — Sets the spindle speed to 3,500 rpm, and turns on the spindle in a clockwise direction.

M203

G43 — Turns on tool length compensation. Z0. — positions the spindle .100 above the part at rapid. **H02** — Instructs the MCU to use the value in register 2 for tool length compensation.

M08 — Turns on the coolant.

N204 through N208 are the tool motion statements

N204

G81 —Turns on the canned drill cycle.

G99 — Instructs the MCU to return the spindle to the reference plane at the end of each cycle iteration.

X.5y1. — Y/Y coordinates of the first hole.

Z-.375 — The final feed depth for the Z axis.

R0. — Sets Z0 as the reference plane (feed engagement point).

N205

G25 — Initiates the do loop.

P206 —Tells the MCU to begin the loop in block N106.

Q206 — Tells the MCU that N106 is the end of the loop.

L5 — Instructs the MCU to perform the loop 5 times.

N206

G91 — Selects incremental positioning.

X1. — Causes the spindle to move 1.000 inch along the X axis. Since the drill cycle is still turned on, a hole will be drilled at this location, the do loop instructions.

Do loops

N207

G80 — Cancels the active drill cycle.

M09 —Turns off the coolant.

N208 through N209 are the tool cancel commands

N208

G00G91G30Z0. — Sends the Z axis to tool change position.

M05 — Turns off the spindle.

N209

G30X0.Y0. — Sends the X and Y axes to tool change position. M19 — Orients the spindle for tool change gripper.

N210

M30 — End of program code.

Subprograms

- A ***subprogram*** is a separate program called by another program

- The use of subprograms can significantly **reduce** the amount of programming required on some parts

- One way to use a ***subprogram*** is **to place one or more do loops in the subprogram**. This is known as **nesting**

- Subprograms can also be **nested** in other subprograms, or ***nested within do loops***

- This gives the programmer a great deal of **flexibility** and a powerful programming tool

Subprograms

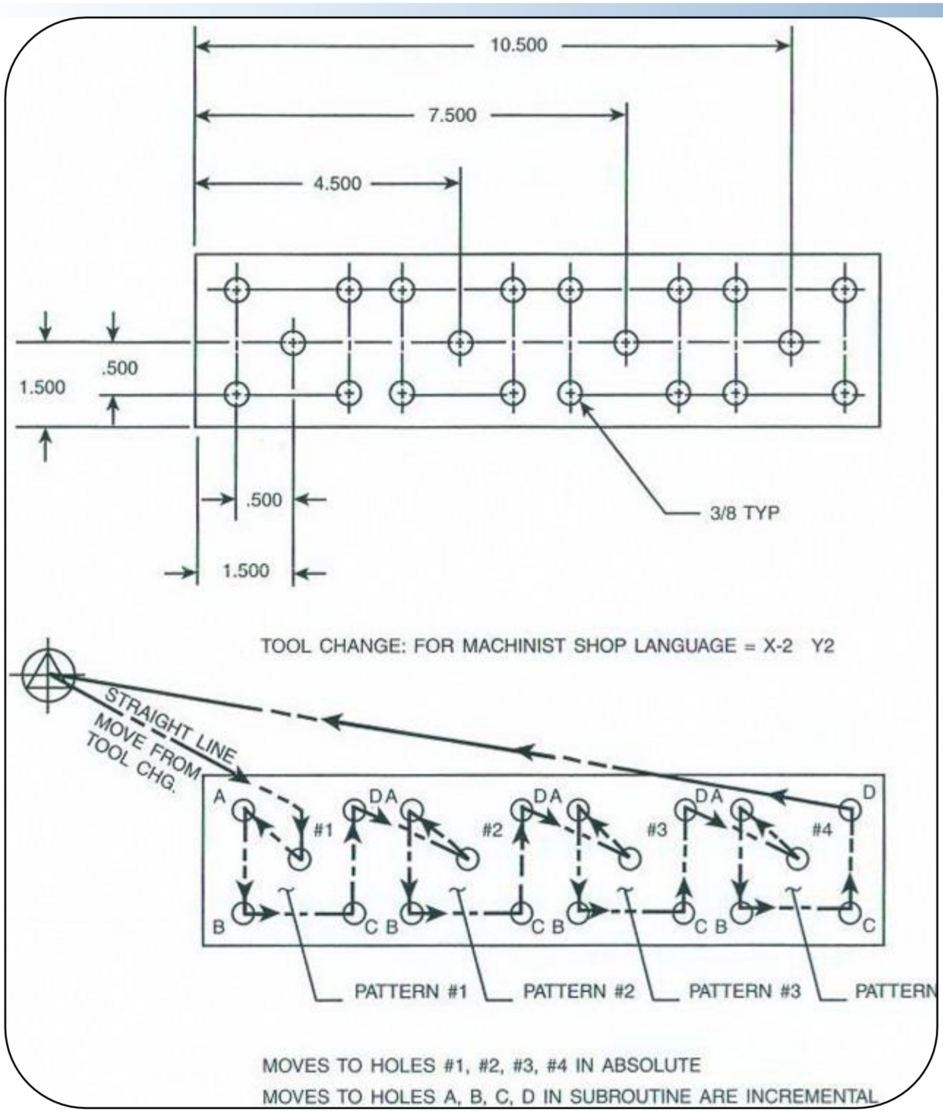

10.500
7.500
4.500
.500
1.500
.500
1.500
3/8 TYP
TOOL CHANGE: FOR MACHINIST SHOP LANGUAGE = X-2 Y2
STRAIGHT LINE MOVE FROM TOOL CHG.
A
B
C
D
#1
#2
#3
#4
PATTERN #1
PATTERN #2
PATTERN #3
PATTERN
MOVES TO HOLES #1, #2, #3, #4 IN ABSOLUTE
MOVES TO HOLES A, B, C, D IN SUBROUTINE ARE INCREMENTAL

Figure 4: Part drawing and tool path for subprogram example

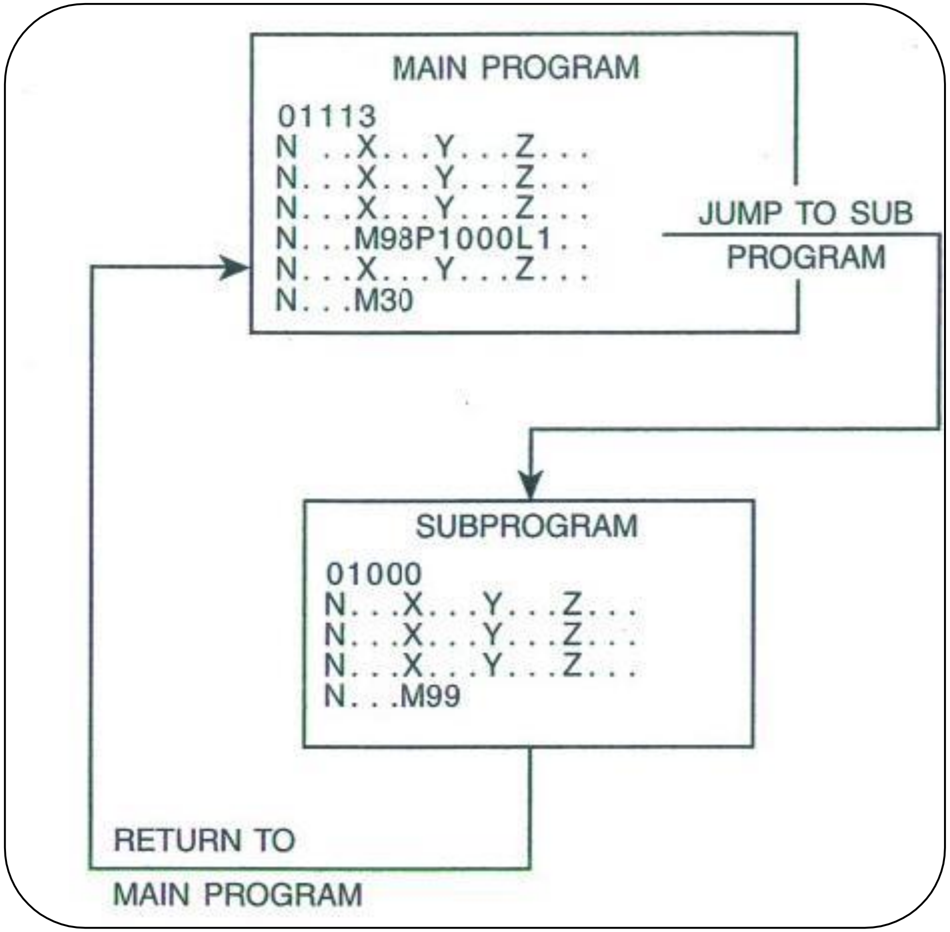
Seams W., "Computer Numerical Control, Concepts & Programming"

- For example, on the part in Figure 4, note that the holes occur in the ***same geometric and dimensional pattern*** in four different locations

- A **do loop** could be programmed to drill the holes, but programming steps can be **minimized** by placing the pattern in a **subprogram**

- The drill can be sent to **hole #1** and the subprogram called to drill the four holes A, B, C, and D. **Hole #2** can then be positioned and the subprogram called again, and so on

Subprograms



```
01113
N . .X. . .Y. . .Z. . .
N. . .X. . .Y. . .Z. . .
N. . .X. . .Y. . .Z. . .
N. . .M98P1000L1. .
N. . .X. . .Y. . .Z. . .
N. . .M30
```

```
01000
N. . .X. . .Y. . .Z. . .
N. . .X. . .Y. . .Z. . .
N. . .X. . .Y. . .Z. . .
N. . .M99
```

Figure 5: Subprogram call mechanics

Seams W., "Computer Numerical Control, Concepts & Programming"

- The **flowchart** in Figure 5 illustrates how a **subprogram** works. The calling program is referred to as the *main,* or **parent program**. The **subprogram** is sometimes referred to as the **subroutine** or **child program**

- When the **subprogram** call statement is issued in the main NC program**, the MCU switches to the subprogram**. The subprogram then **executes**

- At the end of the subprogram is a command that causes the **MCU to switch back to the main program.** The MCU returns to the NC block in the main program immediately following the command that called the subprogram in the first place

Calling Subprograms

Main Program

O0001

```
N001 X/Y/Z
N002   -
N003   -
N004 M98P2000L1
N005   -
N006   -
N007   -
N008 M30
```

Subprogram

O2000

```
N001 X/Y/Z
N002   -
N003   -
N004 M99
```

- Where:

M98 – Instructs the MCU to **jump** to a subprogram

P2000 – Tells the MCU that O2000 is the **subprogram ID**

L1 – Instructs the MCU to **execute** the subprogram one time

Calling Subprograms

```
%
O1106
(* **********)
(* X0/Y0 - LOWER LEFT CORNER)
(* Z0 - .100 ABOVE TOP OF PART)
(* **********)
(* TOOL 1 - NO. 3 C-DRILL)
(* **********)
N001 G17 G90 G40 G80 G98
N101 T01 M06
N102 G00 G90 X1.5 Y1.5 S3500 M03
N103 G43 G90 Z3. H01 M08
N104 G81 G99 X1.5 Y1.5 Z-.162 R0. F10.5
N105 P1000 M98 L1
N106 X4.5 Y1.5
N107 P1000 M98 L1
N108 X7.5 Y1.5
N109 P1000 M98 L1
N110 X10.5 Y1.5
N111 P1000 M98 L1
N112 G80 M09
N113 G00 G91 G28 Z0. M05
N114 G28 X0. Y0. M01
(* **********)
(* TOOL 1 - NO. 3 C-DRILL)
(* **********)
N002 G90 G40 G98 G80
N201 T02 M06
N202G00 X1.5 Y1.5 S3500 M03
N203 G43 G90 Z3. H01 M08
N204 G81 G99 X1.5 Y1.5 Z-.375 R0. F10.5
```

```
N205 P1000 M98 L1
N206 X4.5 Y1.5
N207 P1000 M98 L1
N208 X7.5 Y1.5
N209 P1000 M98 L1
N210 X10.5 Y1.5
N211 P1000 M98 L1
N212 G80 M09
N213 G00 G91 G28 Z0. M05
N214 G28 X0. Y0.
N215 M30


01000
(* **********)
(* START OF SUBPROGRAM 1000)
(* **********)
N001 G91 X-.5 Y.5
N002 Y-1.
N003 X1.
N004 Y1.
N004 G90
N006 M99
%
```

Figure 6: Program to machine the part in Figure 4 *(Seams W., “Computer Numerical Control, Concepts & Programming”)*

Calling Subprograms

Program Explanation

O1106
Program id number
N001 through N103 is the tool change sequence
N001
T01 — Put tool 1 in tool change standby position.
M06 — Initiate a tool change. Tool 1 will be placed in the spindle.
N102
G00G90X1.5Y1.5 — Move to hole #1 at rapid, using absolute positioning.
S3500M03 — Turns on the spindle clockwise at 3,500 rpm.
N103
G43 —Turns on tool length compensation.
Z3. — Positions the spindle to Z3.000
H01 — Instructs the MCU to use the values in offset register 1 for tool length compensation.
MOB — Turns on the flood coolant.
N104 through N112 are the tool motion blocks
N104
G81 —Turns on the canned drilling cycle.
G99 — Specifies a retract to the reference plane at the end of each drill cycle iteration.
X1.5Y1.5 — X/Y coordinates of the first hole.
Z-.162 —Final drilling depth.
R0. — The reference plane. This is the Z coordinate where the MCU will begin feeding the tool into the workpiece.
F10.5 — Specifies a drilling feedrate of 10.5 inches per minute

Seams W., "Computer Numerical Control, Concepts & Programming"

Calling Subprograms

N105

This block is a jump to subprogram call.

P1000 — Specifies program O1000 as the target of the subroutine jump.

M98 — Instructs the MCU to jump to the target program (in this case O1000).

L1 — Tells the MCU to execute the subprogram 1 time.

N106

X4.5Y1.5 — Positions the spindle over hole #2.

N108

X7.5Y1.5 — Positions the spindle over hole #3.

N109

P100M98L1 — Jump to subprogram O1000 call.

N110

X10.5Y1.5 — Positions the spindle over hole #4.

N111

P100M98L1 —Jump to subprogram O1000 call.

N112

G80 — Cancels the canned drilling cycle.

M09 — Turns off the flood coolant.

N113 through N114 is the tool cancel sequence

N113

G00G91G2820. — Sends the spindle to the Z axis home position in rapid traverse mode.

M05 —Turns off the spindle.

N114

G28X0.Y0. — Sends the spindle to the X/Y home position.

M01 — Optional program stop code. Provided for operator convenience.

Calling Subprograms

N002 through N203 is the tool change sequence

N002

Safety block

N201

T02 — Put tool 2 in tool change standby position.

M06 — Initiate a tool change. Tool 2 will be placed in the spindle.

N202

G00G90X1.5Y1.5 — Move to hole #1 at rapid, using absolute positioning.

S3500M03 — Turns on the spindle clockwise at 3,500 rpm.

N203

G43 — Turns on tool length compensation.

Z3. — Positions the spindle to Z3.000

HO2 – Instructs the MCU to use the values in offset register 2 for tool length compensation.

M08 — Turns on the flood coolant

N204 through N212 are the tool motion blocks

N204

G81 —Turns on the canned drilling cycle.

G99 — Specifies a retract to the reference plane at the end of each drill cycle iteration.

X1.5Y1.5 — X/Y coordinates of the first hole.

Z-.375 — Final drilling depth.

RO. — The reference plane. This is the Z coordinate where the MCU will begin feeding the tool into the workpiece.

F10.5 — Specifies a drilling feedrate of 10.5 inches per minute.

Calling Subprograms

N205
P100M98L1 — Jump to subprogram O1000 call.
N206
X4.5Y1.5 — Positions the spindle over hole #2.
N208
X7.5Y1.5 — Positions the spindle over hole #3.
N209
P100M98L1 — Jump to subprogram O1000 call.
N210
X10.5Y1.5 — Positions the spindle over hole #4.
N211
P100M98L1 — Jump to subprogram O1000 call.
N212
G80 — Cancels the canned drilling cycle.
M09 — Turns off the flood coolant.
N213 through N214 is the tool cancel sequence
N213
G00G91G28Z0. — Sends the spindle to the Z axis home position in rapid traverse mode.
M05 — Turns off the spindle.
N214
G28X0.Y0. — Sends the spindle to the X/Y home position.
N214
M30 — End of program memory reset code

Calling Subprograms

Subprogram Explanation

- Notice that a **subprogram** has ***its own program ID number***, in this case **O1000**
- The **sequence blocks also are numbered independently** from the main program
- The only **difference** between the **subprogram** and an **independent program** is the **return to calling program command** (**M99**) at the end of the program

Seams W., "Computer Numerical Control, Concepts & Programming"

Calling Subprograms

Subprogram Explanation

O1000

Program ID number.

N001

G91 — Selects incremental positioning mode. Incremental moves are used throughout the subprogram.

X-.5Y.5 — Moves the spindle to hole A. Since the drill cycle was turned on in the main program, a hole will be drilled at this location.

N0D2

Y-1. — Moves the spindle from hole A to hole B.

N003

X1. — Moves the spindle from hole B to hole C.

N004

Y1. — Moves the spindle from hole C to hole D.

N005

G90 — Selects absolute positioning mode. Since the main program uses absolute mode, operator confusion is minimized by placing the machine back into absolute mode before returning from the subprogram.

N006

M99 — Return to calling program code. The MCU Will return to the main program when this code is received. If this were an independent program rather than a subprogram, an **M30** would have been used here instead of M99

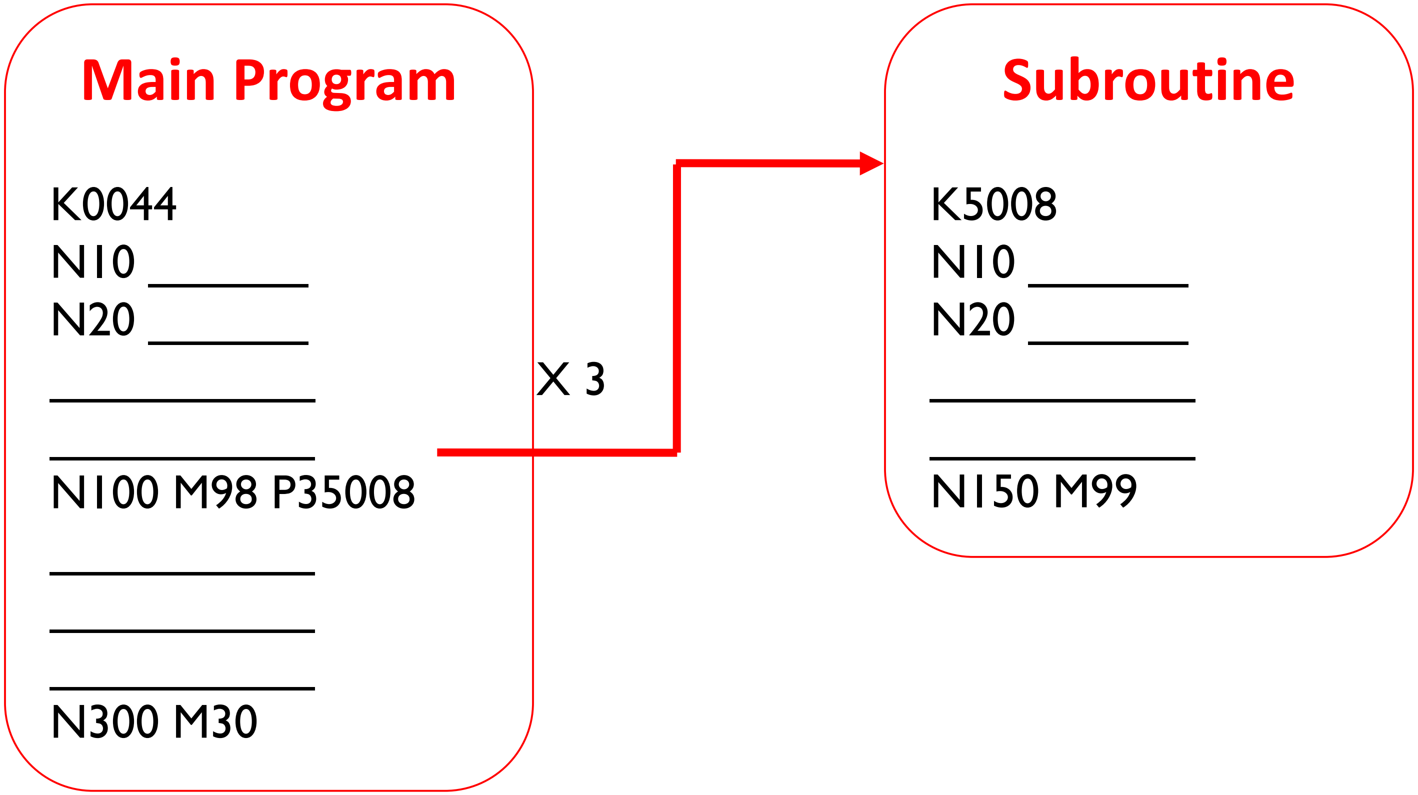
Subroutines

Subroutines

- **A subroutine is an independent program** which is called within the program
- Used when there is **need for repeating** a sequence of commands
- Programming **time saving**
- **Register** orders once, **recall** anywhere in the program and **repeat** as many times as needed
- **M98**, **M99** commands

Subroutines

- Programming with **FANUK MCU**



Main Program

K0044
N10 ______
N20 ______
____
____
N100 M98 P35008
____
____
____
N300 M30

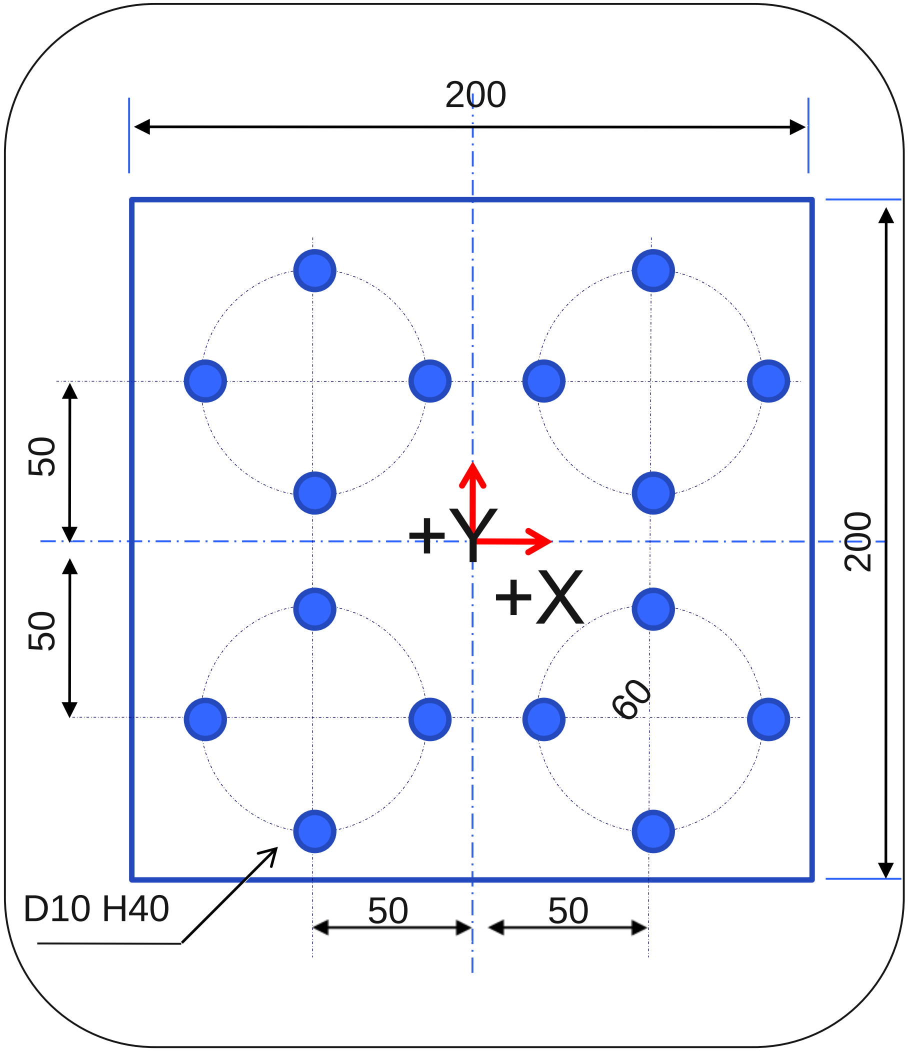
X 3

Subroutine

K5008
N10 ______
N20 ______
____
____
N150 M99

(source: Σύγχρονες μέθοδοι κατεργασίας υλικών και προγραμματισμός με Ηλεκτρονικό Υπολογιστή (Η/Υ) ,Δ. Μούρτζης ,κ.α.)

Subroutines example program



(source: Σύγχρονες μέθοδοι κατεργασίας υλικών και προγραμματισμός με Ηλεκτρονικό Υπολογιστή (Η/Υ) ,Δ. Μούρτζης ,κ.α.)

- In aluminum plate 200 x 200 x 50 mm to 16 planned drilling holes as shown

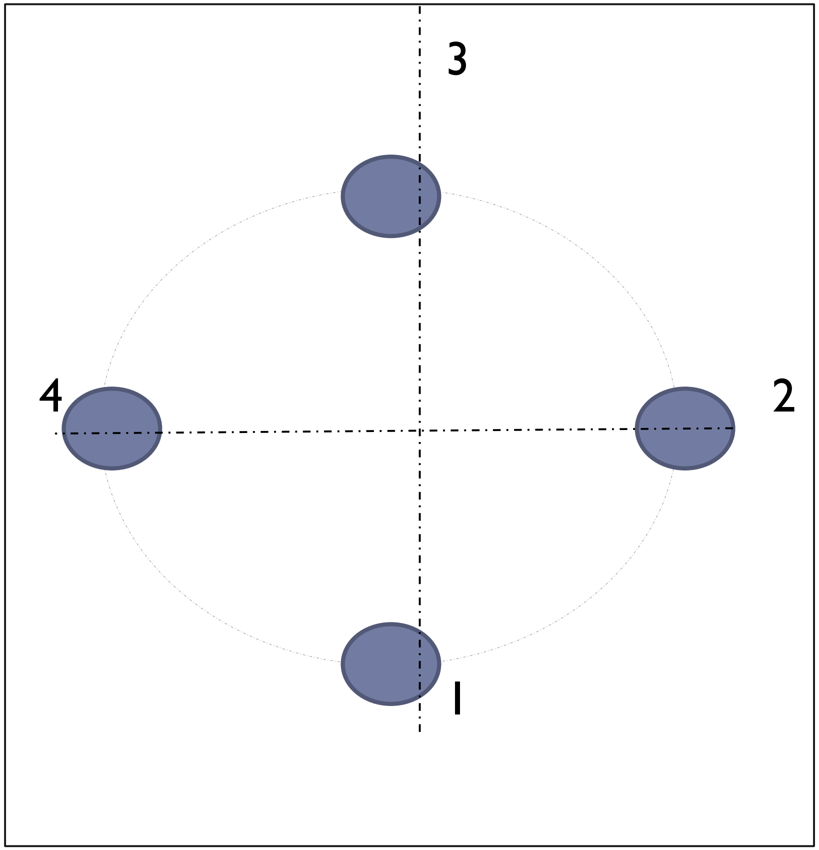
- The **depth of penetration** is 40 mm
- The available **tool T1** is drill D10 with the following **cutting conditions**:

I. Feed-rate speed when drilling F = 60 mm / min

II. Spindle tool-holder **rotation speed** S = 800 rpm

III. Cooling operation on

Subroutines example program



- Subroutine **P2222**

```
P2222
N10 G91 X30 Y30          Hole 2
N20 X-30 Y30             Hole 3
N30 X-30 Y-30            Hole 4

N50 M99                  Return to program
```

(source: Σύγχρονες μέθοδοι κατεργασίας υλικών και προγραμματισμός με Ηλεκτρονικό Υπολογιστή (Η/Υ) ,Δ. Μούρτζης ,κ.α.)

Subroutines example program

```
%                                       Program launch
:0008                                   Program number
N10 G21                                 Dimension in mm
N20 G40 G80                             Cancellation block
N30 M06 T01                             Cutter change, T01
N40 M13 S700                            Rotation CW – cooling on
N50 G90 G00 X-50 Y-80 Z5                Rapid approach of the hole (bottom-left)
N60 G83 G99 X-50 Y-80 Z-40 Q10 R2 F50   Machining cycle G83
N70 M98 P2222                           Subroutine call P2222
N80 G90 X50 Y-80                        Hole(bottom-right)
N90 M98 P2222                           Subroutine call P2222
N100 G90 X50 Y20                        Hole (upper-right)
N110 M98 P2222                          Subroutine call P2222
N120 G90 X-50 Y20                       Hole (upper-left)
N130 M98 P2222                          Subroutine call P2222
N140 G90 G00 Z50                        Rapid return
N150 M30                                End of program
```

Subroutines for Cutter Diameter Compensation

- As pointed out in chapter 10, **subprograms** are often used with **cutter diameter compensation**. Figure 7 presents a program written for the part shown in Figure 8. The program utilizes a subprogram to mill the part periphery

- The first time the **subprogram** is called, **CRO register D11 is used**. If .520 is placed in the register, .010 stock per side of the part will be left for finishing

- The second time the subprogram is called, **register D11 is used**. Placing a value of .500 in D12 results in a finish pass. Remark statements are included in the program listing that explain the program execution in detail

- The machining sequence used in this program is **identical** to that used in Chapter 10. The **difference** is that the ***duplication of coordinate locations is eliminated*** by using the subprogram

Subroutines for Cutter Diameter Compensation

```
%
O1107
(* **********)
(* X0/Y0 - LOWER LEFT CORNER)
(* Z0 - .100 ABOVE TOP OF PART)
(* THIS PROGRAM CALLS SUBPROGRAM 1007)
(* **********)
N001 G00 G40 G70 G90          (SAFETY LINE)
N101 T01 M06                  (TOOL CHANGE)
N102 G00X7. Y.875             (MOVE CUTTER TO CRO RAMP-ON POSITION)
N103 G45 Z0. H01 M08          (PICK UP LENGTH OFFSET/COOLANT ON)
N104 G01 Z-.89                (POSITION CUTTER TO DEPTH)
N105 G17 G42 D11              (TURN ON THE CRO - USE REGISTER 11)
N106 P1007 M98 L1             (JUMP TO SUBROUTINE TO MILL PART)
N107 G00 X7. Y.875            (POSITION THE CUTTER FOR CRO RAMP-ON)
N108 G01 Z-.89                (POSITION CUTTER TO DEPTH)
N109 G17 G42                  (TURN ON THE CRO - USE REGISTER 12)
N110 P1007 M98 L1             (JUMP TO SUBROUTINE TO MILL PART)
N111 G00 G91 G28 Z0. M09      (RAPID Z TO HOME/COOLANT OFF)
N112 G28 X0. Y0. M05          (RAPID X/Y TO HOME/SPINDLE OFF)
N113 M30                      (END OF PROGRAM)
01007
(* **********)
(* SUBPROGRAM CALLED BY PROGRAM 1107)
(* **********)
N001 G01 X6.                   (CRO ALREADY ON - RAMP-ON TO#1))
N002 Y4.                       (FEED TO #2)
N003 X2.                       (FEED TO #3)
N004 X0. Y2.                   (FEED TO #4)
N005 Y0.                       (FEED TO #5)
N006 X4.875                    (FEED TO #6)
N007 G02 X6. Y1.125 I0. J1.125 (CUT ARC FROM #6 TO #1)
N008 G40 X7.                   (CANCEL CRO AND RAMP-OFF)
N009 G00 Z0.                   (RETRACT THE SPINDLE)
N010 M99                       (RETURN TO CALLING PROGRAM)
%
```

Figure 7: Program to machine the part in Figure 8

Seams W., “Computer Numerical Control, Concepts & Programming”

Nested Loops

- **Do loops** may ***nest*** inside other do loop or subprograms

- Similarly, **subprograms** may ***nest*** inside other subprograms

- This **concept** will be demonstrated using the part illustrated in Figure 8 and the program in Figure 9. This program features **two loops nested inside a subprogram**

NOTE

In writing a CNC program, a **reference sketch** such as this in Figure 8,is a valuable **aid** in developing a **machining strategy** and provides a way for the programmer to **check** his or her work

Nested Loops

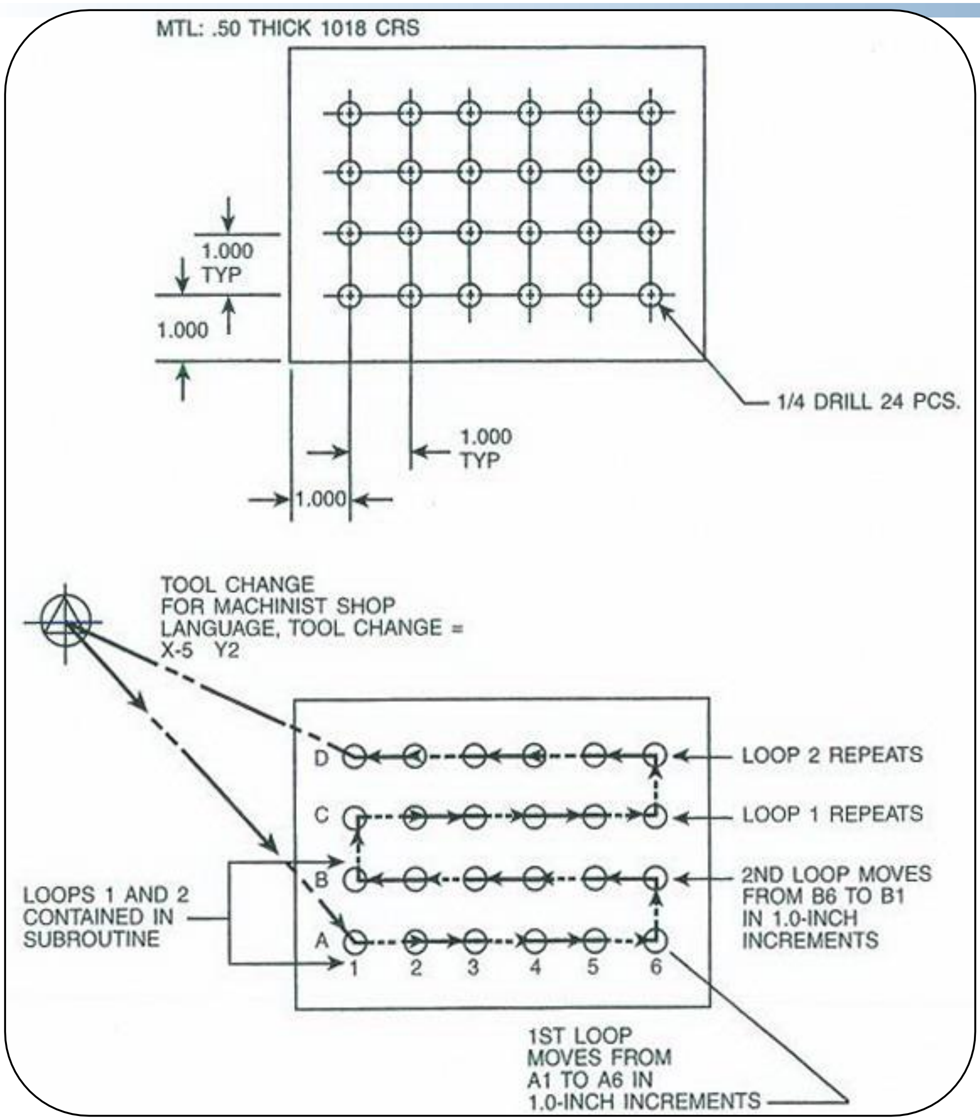


Figure 8: Part drawing and tool path for nested loop example

Seams W., "Computer Numerical Control, Concepts & Programming"

- In Figure 8, the **rows of holes** have been labeled for easy reference.

Nested Loops

```
%
O1109
(* **********)
(* X0/Y0 - LOWER LEFT CORNER)
(* Z0 - .100 ABOVE TOP OF PART)
(* **********)
(* TOOL 1 -  NO. 3 C-DRILL)
(* **********)
N001 G00 G90 G80 G40 G98            (SAFETY LINE)
N101 T01 M06                        (TOOL CHANGE)
N102 G00 X1. Y1. S1700 M03          (POSITION AT A-1)
N103 Z-.163 H01 M08                 (TURN ON TOOL LENGTH OFFSET AND COOLANT)
N104 G81 G98 X1. Y1. Z-.162 R0. F5.1 (TURN ON DRILL CYCLE - DRILL A-1)
N105 P1000 M98 L1                   (JUMP TO SUBPROGRAM 1000)
N106 Y1.                            (POSITION TO AND DRILL C-1)
N107 P1000 M98 L1                   (JUMP TO SUBPROGRAM 1000)
N108 G00 G91 G28 Z0. M09            (RAPID Z HOME/COOLANT OFF)
N109 G28 X0. Y0. M05                (RAPID X/Y HOME/SPINDLE OFF)
N110 M01                            (OPTIONAL STOP CODE)
(* **********)
(* TOOL 2 -  1/4 DRILL)
(* **********)
N002 G00 G90 G80 G40 G98            (SAFETY LINE)
N201 T01 M06                        (TOOL CHANGE)
N202 G00 X1. Y1. S1700 M03          (POSITION AT A-1)
N203 Z-.163 H01 M08                 (TURN ON TOOL LENGTH OFFSET AND COOLANT)
N204 G81 G98 X1. Y1. Z-.162 R0. F5.1 (TURN ON DRILL CYCLE- DRILL A-1)
N205 P1000 M98 L1                   (JUMP TO SUBPROGRAM 1000)
N206 Y1.                            (POSITION TO AND DRILL C-1)
N207 P1000 M98 L1                   (JUMP TO SUBPROGRAM 1000)
N208 G00 G91 G28 Z0. M09            (RAPID Z HOME/COOLANT OFF)
N209 G28 X0. Y0. M05                (RAPID X/Y HOME/SPINDLE OFF)
N210 M30                            (END OF PROGRAM - MEMORY RESET)
```

```
O1000
(* **********)
(* SUBPROGRAM 1000 CALLED BY)
(* PROGRAM NUMBER 1109)
(* **********)
N101 G25 P112 Q102 L5         (BEGIN DO LOOP - REPEAT 5 TIMES)
N102 G91 X1.                  (MOVE INCREMENTALLY 1.000 IN X)
N103 Y1.                      (POSITION TO C-1)
N104 G25 P104 Q104 L5         (BEGIN A LOOP - REPEAT 5 TIMES)
N105 G91 X-1.                 (MOVE INCREMENTALLY -1.000 IN X)
N106 G90                      (SWITCH TO ABSOLUTE POSITIONING)
N107 M99                      (RETURN TO CALLING PROGRAM)
%
```

Figure 9: Program to machine the part in Figure 8

Seams W., "Computer Numerical Control, Concepts & Programming"

Summary 1/2

The important **concepts** presented in this chapter are:

- A **do loop** instructs the MCU to **repeat** a series of instructions a specified number of times

- The **format** for a do loop is: **G25P...Q..-L.**

  - **Where**:.

G25 **turns on** the loop

P is the **beginning block** number of the loop.

Q is the **ending block** number of the loop.

L is the **number of times** to repeat the loop.

Summary 2/2

- A **subprogram** is a program called by another program in **a parent-child**
- The **format** for calling a subprogram is: **P....M98L**
  - **Where**:

P - is the **program number** of the subprogram.
M98 - causes subprogram **P** to **execute.**
L - specifies the **number of times** subprogram **P** executes.

- **Nested loops** are placed inside other loops or **inside** subprograms
- The **codes for subprograms** and **do loops** **vary** from controller
- To program a particular machine, it will be necessary to ***consult the programming manual*** for the machine in question.

Vocabulary Introduced in this chapter

- Do loop
- Main program
- Nested loop
- Subprogram
- Subroutine

References

1. Chryssolouris G., «Manufacturing Systems: Theory and Practice», 2nd Edition, 2006, Springer-Verlag
2. http://www.dptechnology.com/
3. Kalpakjian S., «Manufacturing Engineering and Technology», 2nd Edition, 1992, Addison-Wesley Publishing company
4. Mattson M., “CNC Programming, Principles and Applications”, Delmar, 2002
5. Moriwaki T., “Multi-Functional Machine Tool”, CIRP Annals Manufacturing Technology, Vol. 57/2, 2008, pp. 736-749
6. Seams W., “Computer Numerical Control, Concepts & Programming”, 4th Edition, Delmar, 2002
7. Γ. Χρυσολούρης, «Συστήματα Παραγωγής Θεωρία και Πράξη» Μέρος I και II, Εκπαιδευτικές Σημειώσεις, Πανεπιστήμιο Πατρών, 2001,
8. Γ. Χρυσολούρης, Δ. Μούρτζης, Κ. Τσίρμπας, Σ. Καραγιάννης, “Ορθογωνική Κοπή”, Εκπαιδευτικές Σημειώσεις, Πανεπιστήμιο Πατρών, 2000

References

9. Γ. Χρυσολούρης, Δ. Μούρτζης, και άλλοι, “Εργαστήρια Μηχανουργικής Τεχνολογίας Ι και ΙΙ”», Εκπαιδευτικές Σημειώσεις για το εργαστήριο του αντιστοίχου μαθήματος, Πανεπιστήμιο Πατρών, 2008 (4η Έκδοση)

10. Δ. Μούρτζης, “Αριθμητικός Έλεγχος Εργαλειομηχανών” Εκπαιδευτικές Σημειώσεις, Πανεπιστήμιο Πατρών, 2011 (3η Έκδοση)

11. Πετρόπουλου Π.Γ., «Μηχανουργική Τεχνολογία – ΙΙ. Τεχνολογία κατεργασιών κοπής των μετάλλων», 1998, Εκδόσεις Ζήτη

12. Σύγχρονες μέθοδοι κατεργασίας υλικών και προγραμματισμός με Ηλεκτρονικό Υπολογιστή (Η/Υ) ,Δ. Μούρτζης ,Κ.Σαλωνίτης