Application of the C-axis

COMMAND	MEANING
M13	DRIVEN TOOL ON clockwise
M14	DRIVEN TOOL ON counterclockw.
M15	DRIVEN TOOL OFF
M52	Round axis operation (C-axis ON)
M53	Spindle operation (C-axis OFF)

Note

The original FANUC 21T control does not support an operation with the C-axis (round axis) and with driven tools.

To be able to use this function still on the respective EMCO machines (EMCO TURN 325/II, PC TURN 155, CONCEPT TURN 155), EMCO WinNC FANUC 21T has been expanded by that scope of commands of FANUC 21i which supports the C-axis with driven tools.

For that purpose the adjoining M-functions have been taken over.

For milling with polar coordinate interpolation the functions G12.1/G13.1 have been taken over. The cycles G83 and G84 have been modified correspondingly for the operation with driven tools. Programming of G83 and G84 is identical with the one in the original FANUC 21T control.

Axial working with driven tools

Note:
When using driven tools (EMCO Turn 325/II, PC Turn 155, Concept Turn 155) an offset of X-20mm must be entered in the tool data for the driven tool in X-direction . This offset results from the different positions of driven tools versus rigid tools.

Deep-hole drilling axial with driven tools, G83

Format

N... G83 Z-15 Q.... F...

G83 ..... call drilling cycle
Z-15 ..... final drilling depth absolute (here 15)
Q [µm] ..... infeed depth to withdrawal
F ..... drilling feed

Prior to the call of the drilling cycle the tool in the X- and C-axis must be positioned on drilling centre and in the Z-axis at a safety distance. After the cycle end the tool is positioned in rapid motion to the last position before cycle call (safety distance). The number of revolutions and the rotational direction must be programmed prior to the cycle call.

General notes

- X must always be programmed even if in the previous block the tool has already been traversed to the turning centre (N... G00 X Z...)
  If X is progammed, in the previous block only the start position for the Z-axis must be approached (N.. G00 Z3)
- If Q is not specified, no cut division is carried out, i.e. drilling till Z-endpoint in one movement.

Tapping axial with driven tool, G84

Format
N... G84 Z-10 F... M...

G84 ..... call drilling cycle
Z-10 ..... tapping depth absolute (here 10)
F [mm] ..... thread pitch
M ..... rotational direction of spindle (M13 or M14)

Note:
When using driven tools (EMCO Turn 325/II, PC Turn 155, Concept Turn 155) an offset of X-20mm must be entered in the tool data for the driven tool in X-direction . This offset results from the different positions of driven tools versus rigid tools.

Prior to the call of the drilling cycle the tool in the X- and C-axis must be positioned on drilling centre and in the Z-axis on safety distance.

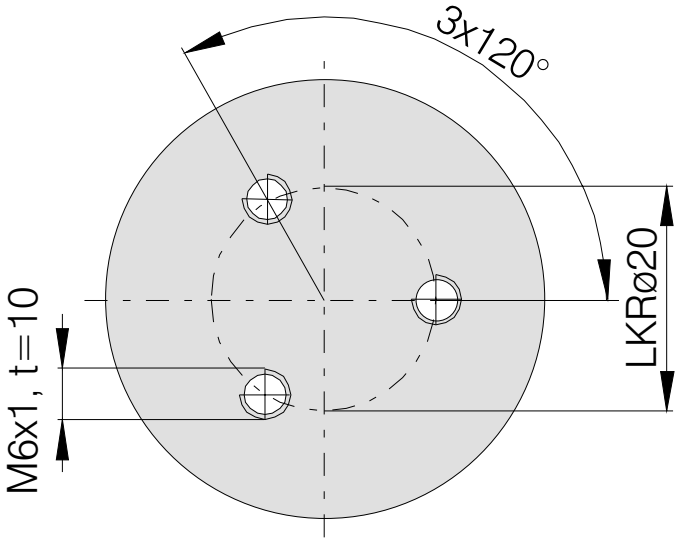
General notes

- X must always be programmed even if in the previous block the tool has already been traversed to the turning centre (N... G00 X Z...)
  If X is progammed, in the previous block only the start position for the Z-axis must be approached (N.. G00 Z3)
- The tapping cycle is started with the respective M-function (M13 or M14). At the target point the sense of rotation of the spindle is automatically reversed for returning. When the starting position is reached again, the system switches automatically to the original sense of rotation.
- Tapping with driven tool (M13, M14) only with axial tappings, which are outside of the turning centre. For that purpose also the C-axis must previously be activated and correspondingly positioned (M52).

Deep-hole drilling, G83 and tapping, G84 axial with driven tool

Example:

```
( LKR DMR.20 )
( M6 10mm tief )
( 3x 120GRAD )
G10 P0 Z-100
T0000 G0 X100 Z150
G90 G40 G95
T0505
M52
G28 G0 C0
M13
G97 S2000
G0 X20 Z2
G83 Z-15 Q5000 F0.15
G0 C120
G83 Z-15
G0 C240
G83 Z-15
G0 Z20 M15
T0707 ( AXIAL TAPPING )
N90 G97 S300
N95 G0 X20 Z5
G0 C0
G84 Z-10 F1 M13
G0 C120
G84 Z-10 F1 M13
G0 C240
G84 Z-10 F1 M13
G0 Z20 M15
M53
M30
```



Radial working with driven tools

Note:
When using driven tools (EMCO Turn 325/II, PC Turn 155, Concept Turn 155) an offset of X-20mm must be entered in the tool data for the driven tool in X-direction . This offset results from the different positions of driven tools versus rigid tools.

Deep-hole drilling radial with driven tool, G77

Format
N... G77 R1
N... G77 X-4 P... F...

G77 ..... call drilling cycle
R1 [mm] .... withdrawal (here 1)
X-4 ..... final drilling depth (here 4)
P [µm] ..... infeed to withdrawal
F ..... drilling feed

Prior to the call of the drilling cycle the tool in the Z- and C-axis must be positioned on drilling centre and in the X-axis at a safety distance. After the cycle end the tool is positioned in rapid motion to the last position before cycle call (safety distance). The number of revolutions and the rotational direction must be programmed prior to the cycle call.

General notes

- If P is not specified, no cut division is carried out i.e. drilling till Z-endpoint in one movement.

Tapping radial with driven tool, G33

Format
N... G33 X2 F... M13
N... G33 X24 F... M14

G33 ..... tapping
X2 [mm] ..... thread depth absolute (here 2)
X24 [mm] ... starting point
F [mm] ..... thread pitch drilling/withdrawal
M13 ..... rotational direction of spindle drilling
M14 ..... rotational direction of spindle withdrawal

Prior to the call of the drilling cycle the tool in the Z- and C-axis must be positioned on drilling centre and in the X-axis at safety distance.

General notes

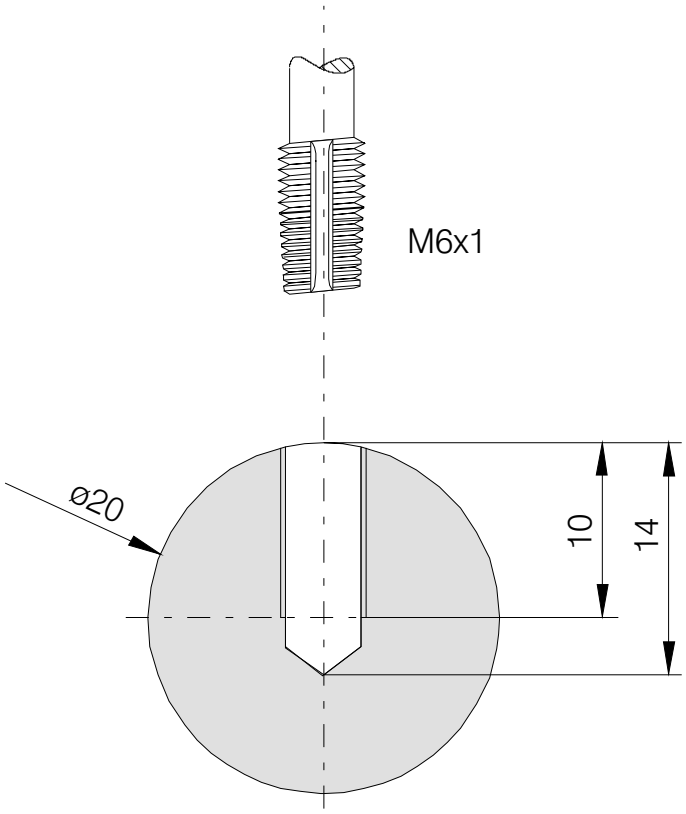
- The tapping cycle is started with the respective M-function (M13 or M14). At the target point the sense of rotation of the spindle is automatically reversed for returning. When the starting position is reached again, the system switches automatically to the original sense of rotation.

Note:
When using driven tools (EMCO Turn 325/II, PC Turn 155, Concept Turn 155) an offset of X-20mm must be entered in the tool data for the driven tool in X-direction . This offset results from the different positions of driven tools versus rigid tools.

Deep-hole drilling, G77 and tapping, G33 radial with driven tool

Example

```
( M6 10mm deep)
G10 P0 Z-100
T0000 G0 X100 Z150
G90 G40 G95
T0909
M52
G28 G0 C0
M13
G97 S2000
G0 X24 Z-10
G77 R1
G77 X-8 P5000 F0.15
G0 C120
G0 X80 M15
T0909
N90 G97 S300
N95 G0 X26 Z-10
G0 C0
G33 X0 F1 M13
G33 F1 X24 M14
G0 X80 Z20 M15
M53
M30
```



G: Flexible NC programming

Variable number	Variable type	Function
#0	Always zero system variable	This variable has always the value zero. Not changeable
#1-33	Local variable	At disposal for calculations in the program
#100-149	Global variables	At disposal for calculations in the program
#500-531	System variable	At disposal for calculations in the program
#1000	System variable	Loading magazine: bar end reached
#1001	System variable	Loading magazine: loader has advanced
#1002	System variable	Loading magazine: first part after bar change
#3901	System variable	Nominal piece number
#3901	System variable	Actual piece number

Variables and arithmetic parameters

By using variables instead of fixed values, a program can be configured more flexibly. Thus, you can react to signals, such as e.g. measuring values, or the same program can be used for different geometries by using variables as nominal value.
Together with variable calculation and program jumps you get the possibility to create a highly-flexible program archive and thus save programming time.

Local and global variables can be read and written. All other variables can only be read.

Local variables can only be used in that macro in which they are defined.
Global variables can be used in every macro irrespective of the macro in which they were defined.

Calculating with variables

With the four basic arithmetic operations the usual mathematic notation is valid.

Function	Example
=	#1=2
+	#1=#2+#3
-	#1=#2-#3
*	#1=#2*#3
/	#1=#2/#3

The term at the operator's right can contain constants and/or variables combined by functions.

Each variable can be replaced again by an arithemetic term in square brackets or by a constant.

Example

```
#1=#[#2]
```

During the calculation the limitation is valid that the execution of the calculation is carried out from left to right without observance of the calculation rule point before line.

Example

```
#1=#2*3+#5/2
```

Control structures

In programs the control sequence can be changed by IF and GOTO instructions. Three types of branchings are possible:

- IF[<condition>] THEN
- IF[<condition>] GOTO <n>
- GOTO <destination>

IF[<Condition>] THEN

After IF a provisory term must be indicated. If the provisory term applies, a determined macro instruction is carried out. Only one macro instruction can be carried out.

Example

With equal values of #1 and #2 the value 5 is allocated to #3.

IF [#1 EQ #2] THEN#3=5

IF[<Condition>] GOTO <n>

After IF a provisory term must be indicated. If the provisory term applies, the branching is carried out to block number n. Otherwise the subsequent block is carried out.

Example

If the value of the variable #1 is greater than 10, the branching is carried out to block number N4. Otherwise the subsequent block is carried out.

IF [#1 GT 10] GOTO 4

GOTO <n>

The jump command GOTO can also be programmed without condition. A variable or constant can be used as a branch destination. With a variable the number can be replaced again by a calculation term in square brackets.

Example

Jump to block number 3

GOTO 3

Example

Jump to variable #6

GOTO#6

Relational operators

Relational operators consist of two letters and are used to determine, in comparison with two values, if these are equal or if one value is greater and/or less than the other.

Operator	Meaning
EQ	Equal (=)
NE	Unequal (≠)
GT	Greater than (>)
GE	Greater than or equal (=)
LT	Less than (<)
LE	Less than or equal (=)

The expressions to be compared can be variable n or constants. A variable can be replaced again by a calculation term in square brackets.

Example

IF[#12 EQ 1] GOTO10

Comprising macro programming examples:

IF[#1000 EQ 1] GOTO10

IF[#[10]] NE #0] GOTO#[#1]

IF[1 EQ 1] THEN#2 =5

IF[#[#4+#[#2/2]] GT #20] THEN#[#10]] =#1*5+#7