GDAŃSK UNIVERSITY
OF TECHNOLOGY

FACULTY OF ELECTRICAL AND CONTROL ENGINEERING
Introductory CDIO Project, PG_00049763

Exercise name:
Controlling a numerical machine tool

Instructions for the laboratory exercise

Arkadiusz Lewicki, Jarosław Guziński
ver 03.11.2016

1. Introduction

Numerically controlled machine tools (called. *Computer Numerical Control* CNC) are commonly used to process all types of materials in processes such as cutting, milling, turning, laser cutting for many different types materials, e.g. metals, wood, glass, stone, polystyrene or plastics. These machines are controlled numerically by a microprocessor unit. As a unit control also uses personal computers with the appropriate one installed control software.

The popularity of CNC machines is constantly growing. They have almost completely Supplanted conventional control machines. This is due to the high repeatability of the treatment accuracy and speed of work. Computer programs supporting design allow you to quickly prepare even very complex projects simultaneously generating a machining program for a CNC machine.

2. Description of the laboratory machine

The machine tool used in the laboratory exercise belongs to the group of machine tools gantries with a sliding table (Fig. 1a). Movement in three XYZ axes is possible. Motion table and cutting tool is implemented by appropriate control stepper motors.

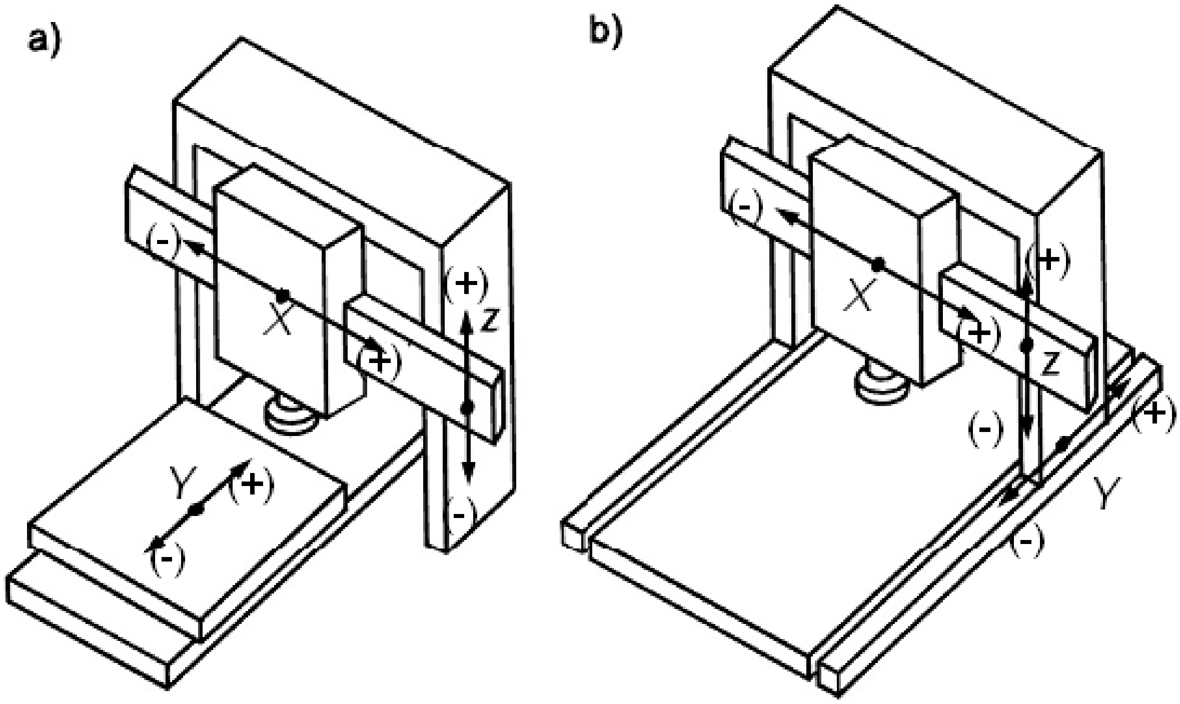


Fig. 1 Gantry machine with movable (a) and fixed (b) table

The view of the machine tool used in the laboratory is shown in Fig. 2

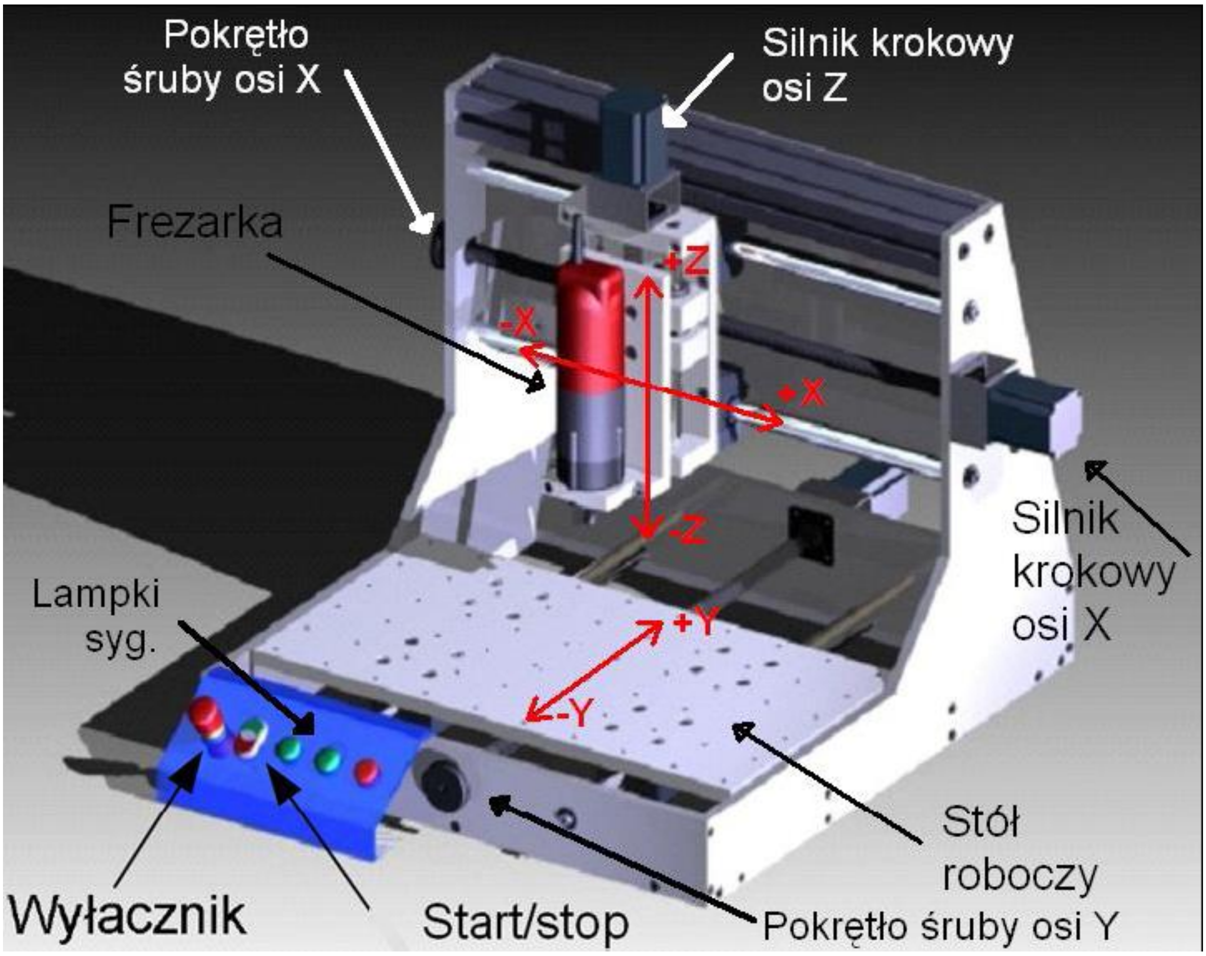


Fig. 2 View of the laboratory machine with mounted milling device

When processing materials, a power tool is mounted as a tool multifunctional with a drill or milling cutter. In the exercise, instead of this device is mounted recorder.

The machine tool must ensure that the user can work safely, therefore safety limit switches were used in it. They work on a layout like this same as the emergency stop, i.e. they cut off power completely to the machine, causing immediate stop. They are positioned to prevent the tool from falling out, either table outside the working area, which could damage the machine. To the situation this can occur in the event of a failure of the control electronics, suspension of the program control program or an incorrectly prepared part program (coordinates outside the working area).

The machine has no homing limit switches. Homing is carried out manually by the person operating the machine. Each axis of the machine is driven by one stepper motor. They were used FL57STH76-2804A stepper motors. They are two-winding motors with four leads.

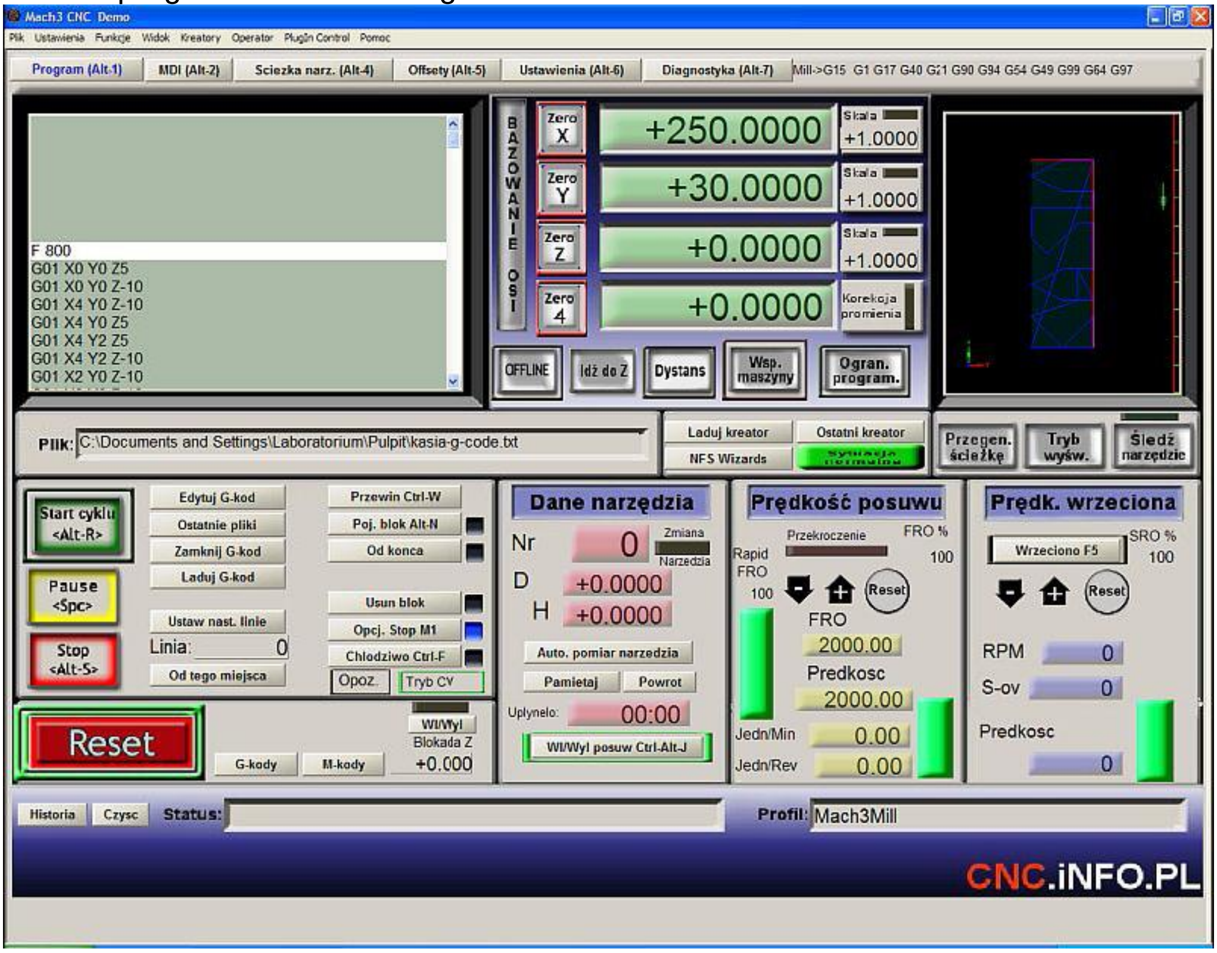
Manual control of the machine is implemented by means of a control panel with signal lamps and buttons. The buttons can be switched on and off machine power supply. In addition, there is a safety switch with interlock cutting off the power supply.

There are three indicator lights on the machine's control panel. They signal the successively supply the machine (green lamp), machine failure (red lamp) and spindle operation (green lamp). The fault light is on when it is started emergency stop switch or any of the limit switches.

The machine is locked after the limit switch has tripped. To her unlock for re-operation, turn the trapezoidal screw nuts by hand to do so way that the work table or gate with the tool unlock the limit switch. Then switch the machine on with the START button.

3. Control program

The machine tool is controlled using a personal computer program from Mach 3 program. Mach 3 program changes the computer into a control unit that through the LPT port controls the operation of each axle's propulsion systems. Main window view Mach 3 program is shown in Fig. 3.



Mach3 CNC Demo

Mill->G15 G1 G17 G40 G21 G90 G94 G54 G49 G99 G64 G97

```
F 800
G01 X0 Y0 Z5
G01 X0 Y0 Z-10
G01 X4 Y0 Z-10
G01 X4 Y0 Z5
G01 X4 Y2 Z5
G01 X4 Y2 Z-10
G01 X2 Y0 Z-10
```

Zero X +250.0000 Skala +1.0000

Zero Y +30.0000 Skala +1.0000

Zero Z +0.0000 Skala +1.0000

Zero 4 +0.0000

Plik: C:\Documents and Settings\Laboratorium\Pulpit\kasia-g-code.txt

Linia: 0

Blokada Z +0.000

Nr 0

D +0.0000

H +0.0000

Uplynelo: 00:00

FRO % 100

Rapid FRO 100

FRO 2000.00

Predkosc 2000.00

Jedn/Min 0.00

Jedn/Rev 0.00

SRO % 100

RPM 0

S-ov 0

Predkosc 0

Status:

Profil: Mach3Mill

Fig. 3. View of the Mach 3 main window

After starting Mach 3, press the button pulsing in red [Reset].

The next step is basing the machine. Homing consists of setting the pen in a given position. Homing is done using the AXIS BASE panel - Fig. 4.

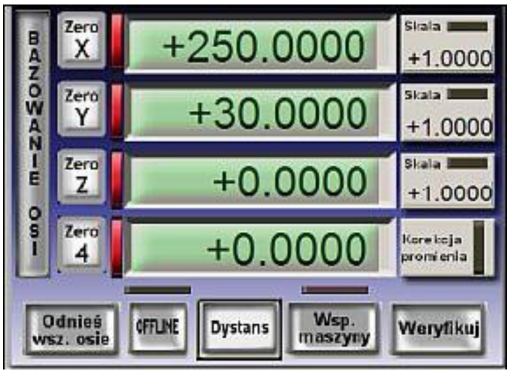


Fig. 4. Mach 3 homing panel view

When homing, the movements in the individual axes are selected with the keys:
- X axis - arrow keys ← →,
- Y axis - arrow keys ↑ ↓,
- Z axis - `[Page Up] [Page Down]` keys.

After setting the pen in a given position, set 0 coordinates for individual axes ([Zero X] [Zero Y] [Zero Z] buttons).

After homing, load the program prepared in G code.

4. CNC programming language - G code

Basic G-code commands are contained in the Polish Standard (PN-73M-55256, PN-93 / M-55251) which is a translation of the ISO standard (ISO 6983). In addition to the defined commands in the standards, a number of CNC machine manufacturers define additional instructions related to specificity of produced machines.

Programming of the CNC machine is done by text files containing a description motion trajectories and ancillary activities. A text file with the program may have any extension. In the Mach 3 program it is opened with the command [Load G-code].
After loading the program, the tool movement path appears in the preview window. To execute the program, scroll the cursor to the first line of the program and press the button [Start cycle]. After finishing the work, close the part program [Close G-code].
In the event of a malfunction in the operation of the device, the program may be executed paused by pressing the space key on the computer keyboard.

The basic G code functions useful for completing the exercise are given in Tab. 1.

Tab. 1. Basic G-code instructions used in the exercise

Syntax function/ Description

% %

Program name Designation of the beginning of the program with its name.
All of the text before the % sign is ignored by the control system. You can thanks to this put all necessary comments and explanations, before the actual program.

Ex .:

Program for making a PCB
Detal17%

Naming the program is not necessary

() (Comment text)

Comments inside the program are placed in round brackets
Ex .:

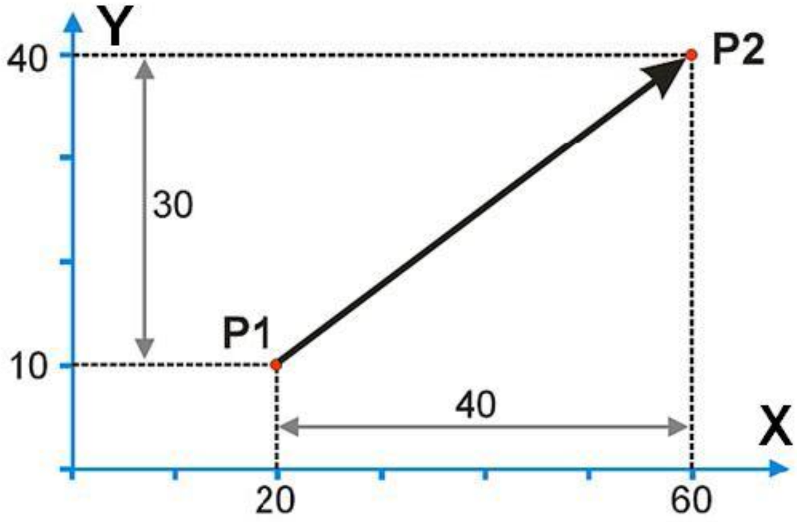
G3 X5 Y5 I-1 (circular motion)

F F_

Set the pen movement speed
Eg. the speed recommended for the pen in the exercise:
F2000

G90 90

Enter absolute location data - usually from the workpiece base.
Ex .:

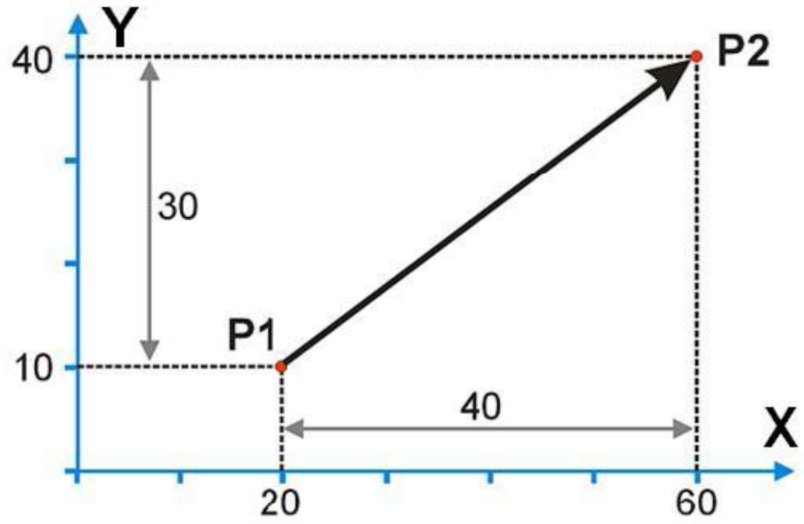


G00 G90 X60 Y40
displacement from any point to P2

G91 G91

Enter incremental location data - all tool displacements are determined relative to its previous position.

Ex .:



```
G0 G91 X40 Y30
```

displacement from point P1 to P2

G0 G0 X_ Y_ Z_

Setting movement function. Moves to a point indicated at maximum setting speed.

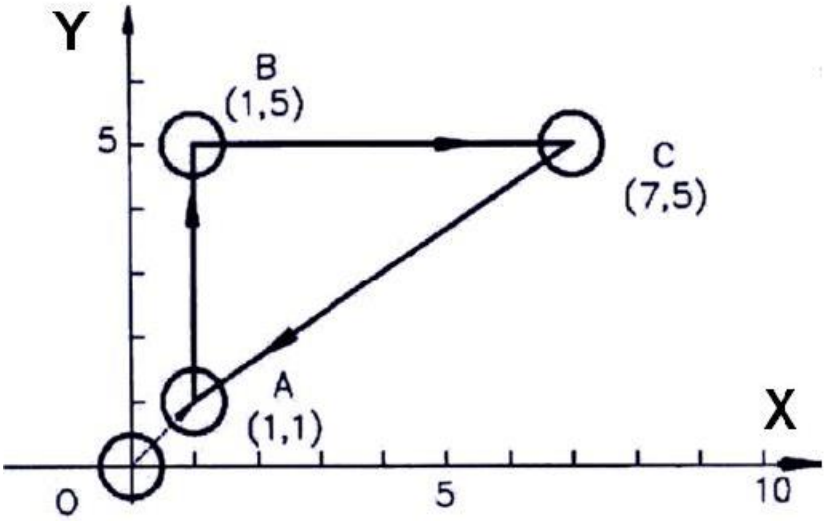
Ex .:

```
G00 X40 Y30 Z20
```

G1 G1 X_ Y_ Z_

Work movement function. It differs from the G0 function in that makes movement at steady operating speed parameter F.

Ex .:



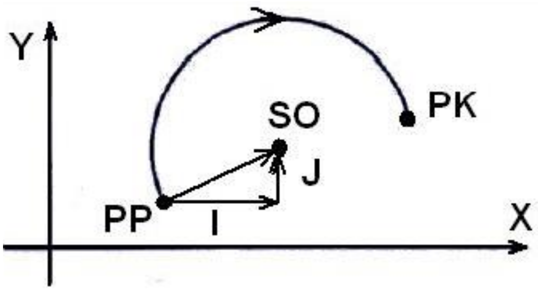
```
G0 X1 Y1 (positioning motion)
G1 X1 Y5 (working motion from A to B)
G1 X7 Y5 (working motion from B to C)
G1 X1 Y1 (working motion from C to A)
```

or

G0 X1 Y1 (positioning motion)
G1 Y5 (working motion from A to B)
G1 X7 (working motion from B to C)
G1 X1 Y1 (working motion from C to A)

G2   G2 X_ Y_ I_ J_

Circle motion function. Programs the circular motion in clockwise in the plane defined by G17, G18, G19 (XY plane by default). X, Y, Z coordinates specify the coordinates of the destination. I, J, K are incremental values of the center of the circle after which is the trajectory.

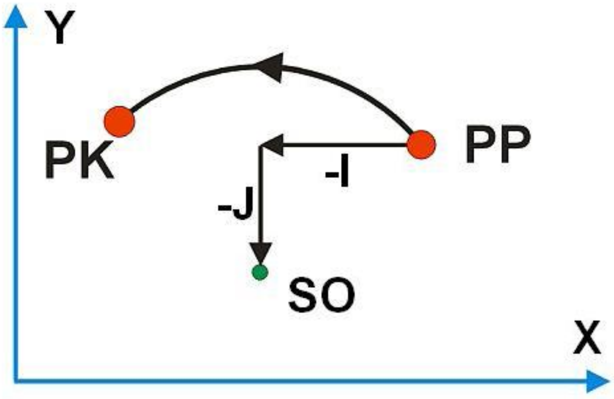


PP - starting point of movement
PK - end point of movement
SO - center of the circle

Ex .:
G17
G2 X60 Y25 I18 J8

G3   G3 X_ Y_ I_ J_

Circle motion function. Programs the circular motion in counterclockwise in the plane defined by G17, G18, G19 (XY plane by default). X, Y, Z coordinates specify the coordinates of the destination. I, J, K are incremental values of the center of the circle after which trajectory runs.



PP - starting point of movement
PK - end point of movement
SO - center of the circle

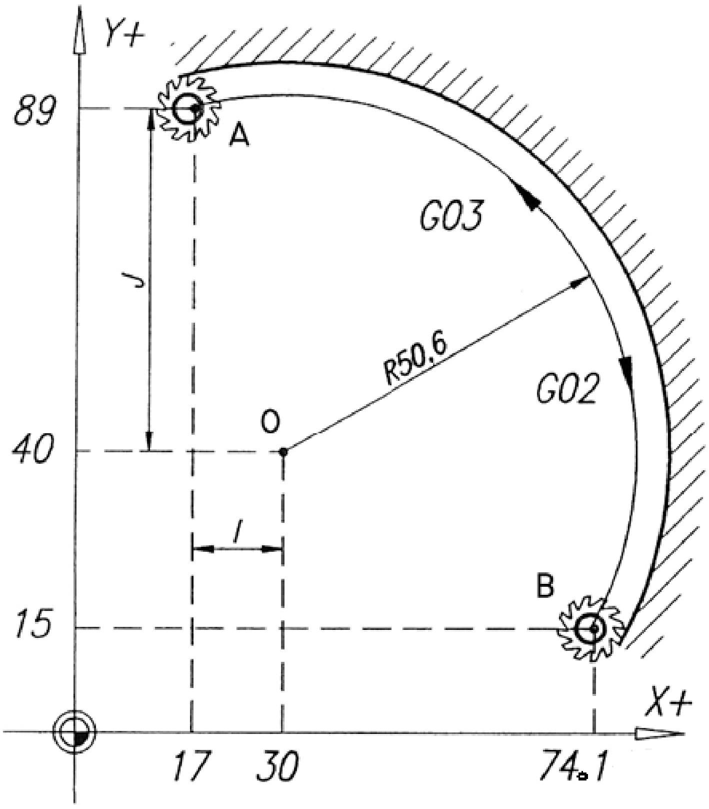
Ex .:

```
G17
G3 X10 Y40 I-20 J-15
```

G2   G2 X_ Y_ R_
G3   G2 X_ Y_ R_

Circular motion functions. saved in format with specifying the circle radius (parameter R)

An example specifying the radius of a circle:



```
G0 X17 Y89 (positioning movement to point A)
G02 X74.1 Y15 R50.6 (circular motion from A to B)
```

An example with the coordinates of the center of the circle:

```
G0 X17 Y89 (positioning movement to A)
G02 X74.1 Y15 I13 J-49 (movement from A to B)
```

G4   G4 P_

Time delay function.
This function performs stopping the program for P seconds

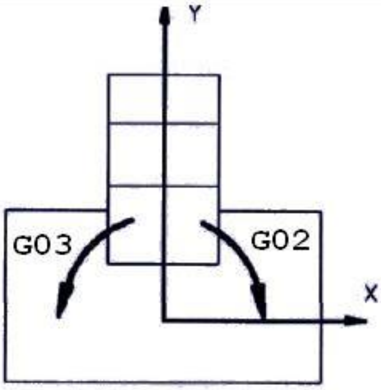
Ex .:

```
G4 P2
```

G17 G17

Set active plane for the circular motion to the XY plane.

Ex .:

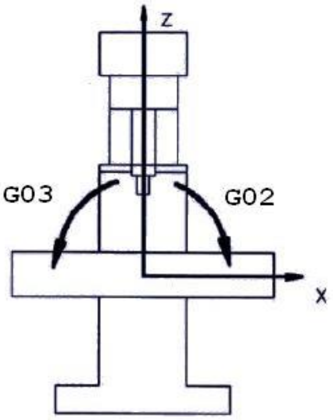


G17

G18 G18

Set active plane for the circular motion to the XZ plane.

Ex .:

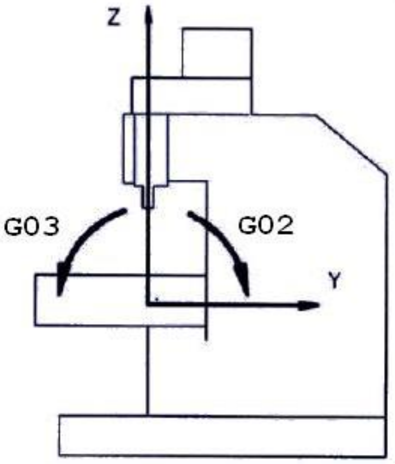


G18

G19 G19

Set active plane for the circular motion to the YZ plane.

Ex .:



G19

M02 M02
End of program

M03 M03
Switching on spindle revolutions (clockwise)

M04 M04
Switching on spindle revs (counterclockwise)

M30 M30
End of the program and scrolling to the beginning of the code

G code as a language has a specific syntax that provides great flexibility.
No restrictions on space characters makes syntax analysis a more difficult task, than standard structures. Keywords search task in the case when a space or Enter can be inserted anywhere, comes down to parse the code character by character. Space marks during program execution and end of line characters are omitted.

Another feature of the G code that introduces some complication is the fact that no repetition of command is required. After calling the G function once, parameters in the next lines of code they are automatically treated as parameters for the called function previously.

This requires remembering previously used commands. So it's correct full version of the program:

```
G1 X10 Y10 Z0
G1 X20 Y10 Z0
G1 X20 Y20 Z0
G1 X20 Y20 Z10
```

as well as in the short version:

```
g1z0
x20
Y20
z10
```

because both versions lead to the same trajectory of movement. In both cases, the spindle

I am at the starting point with the coordinates (X, Y, Z) = (10,10,10), it is omitted to position Z = 0 and then makes two linear movements, first to X = 20 and then to Y = 20. Finally, the spindle is raised back to the Z = 10 position.

There is also some freedom in the matter of parameter transfer. Order entering parameters is irrelevant, i.e. you can e.g. write:

```
G0 X17 Y89
```

as well as

```
G0 Y89 X17
```

Unused or unchanged function parameters can be omitted. There is also no case-sensitive meaning the correct entry is e.g.

G0 X17 Y89

as well as

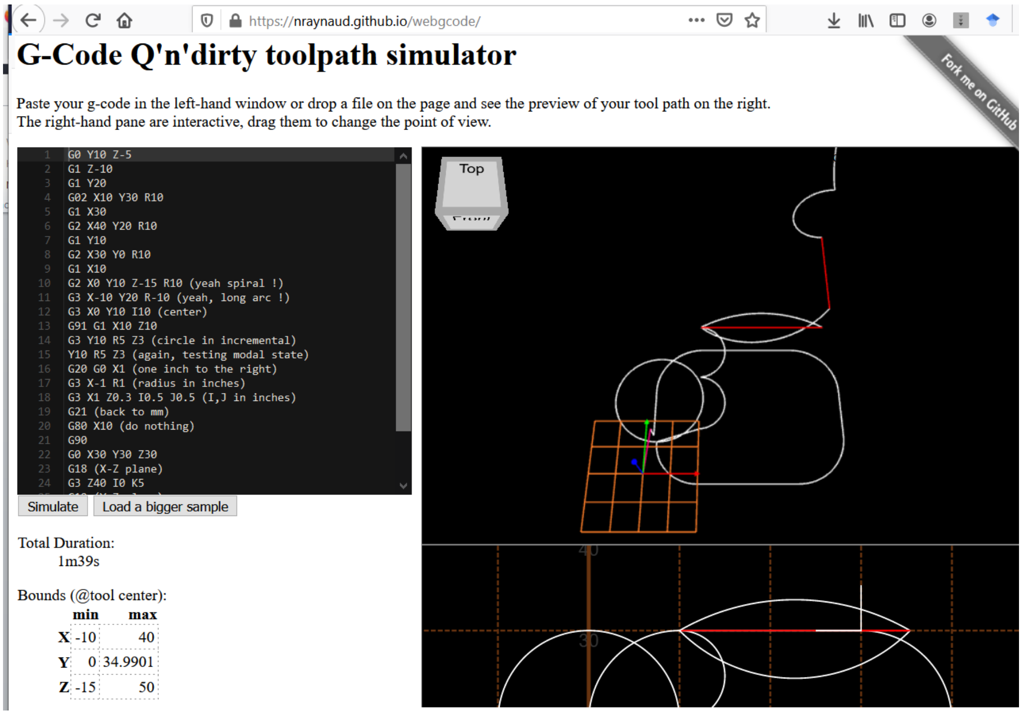
g0 y89 x17

5. The course of the exercise

1. Prepare a text file with the program in G code for writing **author's surname**. Linear and circular motions should be used. Z coordinate it will be set so that for Z = 0 the pen will be above the page, while for Z = -10 the pen it will be pressed against the paper and writing will be possible.

Set typing speed to F2000. The text must fit on an A4 sheet of paper. Units are given in millimeters. The decimal sign is a period.

2. Use the G-code viewer (G-Code Q'n'dirty toolpath simulator): https://nraynaud.github.io/webgcode/



G-Code Q'n'dirty toolpath simulator

Paste your g-code in the left-hand window or drop a file on the page and see the preview of your tool path on the right.
The right-hand pane are interactive, drag them to change the point of view.

```
G0 Y10 Z-5
G1 Z-10
G1 Y20
G02 X10 Y30 R10
G1 X30
G2 X40 Y20 R10
G1 Y10
G2 X30 Y0 R10
G1 X10
G2 X0 Y10 Z-15 R10 (yeah spiral !)
G3 X-10 Y20 R-10 (yeah, long arc !)
G3 X0 Y10 I10 (center)
G91 G1 X10 Z10
G3 Y10 R5 Z3 (circle in incremental)
Y10 R5 Z3 (again, testing modal state)
G20 G0 X1 (one inch to the right)
G3 X-1 R1 (radius in inches)
G3 X1 Z0.3 I0.5 J0.5 (I,J in inches)
G21 (back to mm)
G80 X10 (do nothing)
G90
G0 X30 Y30 Z30
G18 (X-Z plane)
G3 Z40 I0 K5
```

Total Duration:
1m39s

Bounds (@tool center):

	min	max
X	-10	40
Y	0	34.9901
Z	-15	50

2. Load the developed G-code into the **online G-code viewer** and check the correctness of the code in tool movement preview window. Paste your g-code in the left-hand window or drop a file on the page and see the preview of your tool path on the right.

4. Run the program.

The right-hand pane are interactive, drag them to change the point of view.

5. Include your G-code program in the report. Include also obtained results from the online G-code viewer – as printscreens to confirm the correct operation of your program.

The CNC machine prototype in the laboratory was built as a master's thesis: Łukasz Bałdyga, Mariusz Górski "Three-axis CNC numerical machine tool" Politechnika Gdansk, Gdansk 2010.
This manual uses some of the material derived from this work.