C-88 Main Manual

Valid for:
C11 – C88, F22 – F44,
D22 – D33, S22 – S44,
PCNC11 – 88

Newest Version of this Manual can be downloaded under
www.engelhardtgmbh.de/c88.pdf
www.engelhardtgmbh.de/c88de.pdf

CONTENTS

9 CONNECTORS 89

1. OPERATING MODES

1.0.1   Operating modes

After switching on, the CNC displays the menue of the operating modes. The following 8 modes can be selected:

MANUAL MODE   TEACH IN

AUTOMATIC MODE   MANUAL INPUT (MDI)

EXTERNAL DATA   CLEAR MEMORY

GRAPHICS   INPUT MODE

1.0.2   Function keys

REFERENCE   CLEAR   COOLING

RESET   SINGLE STEP

SEARCH BLOC   DELETE BLOC

SPINDLE CW   SPINDLE CCW   SPINDLE

These operating modes can be selected by pressing the corresponding key. Pushing the key $\boxed{T}$ or displays a list of implemented G-functions in the CNC.

A restart of the CNC can be initiated by pushing the key $\boxed{}$ (softstart) in the MENU or RESET (hardware-reset).
In the MENU, each key can call a programm with the number P98XX. If for example the programm P9801 is present in memory, it can be invoked by pushing the key $\boxed{1}$.

1.1 MANUAL MODE

1.1.1 Jogging Mode

After pushing 🖐, the desired moving direction can be selected. Then, after pushing `START` the CNC moves the selected axes in the desired direction. More than 3 or 4 axes can be selected by pushing `T` or ⬇.
The moving speed is controlled by the feedrate potentiometer. The Actual Value Counter monitors the movement. The key `STOP` immediately stops the movement and clears the Direction Keys.
Or push `START` and then for example **+X**. The axis starts moving as long as **+X** is pushed. This Jogging Mode can be left by pushing `STOP`.

1.1.2 Clearing the Actual Value counter

Clearing the Actual Value Counter is done as follows:
- Select the axis to be cleared.
- Clear the counter by pushing **C** once or twice (see P0 N905X).

If a G54 or a tool is active, the Actual Value counter will not display 0 but the programmed displacement of G54 or the tool offset.

1.1.3 Exiting the manual mode

To exit the manual mode, push `STOP` and then `MENU`. Pushing 🖐 or selects the mode MANUAL INPUT, `TEACHIN` selects the mode TEACH IN.

1.1.4 Handwheel, Joystick

The handwheel, if provided, can be activated by pressing for example **+X**. On the display appears H+...0,001 which indicates the step executed by each click of the handwheel. This step can be increased with **+X** and decreased with **-X**. The maximal step is set in P0 N904 Y16.
By turning the handwheel, the activated axis is moved forwards and backwards. This mode can be left by pushing `STOP`.
If a Joystick is connected and activated in the Machine Data P0, it is active immediatly when entering the Manual Mode.

1.1.5 Home Position

A HomePosition can be memorized by pushing 🔲 and then ⤳. The actual position is put into memory as HOME POSITION. The axes then can be moved to this Home Position by pushing 🔲 twice. To enable this function P9974 must be present in memory as described in chapter 8.

1.1.6 Reference point

For moving an axis to the reference point, select for example **+X** and 🔲 and then push `START`. The axis starts moving to the positive limitswitch, stops, reverts, looks eventually for the referencepulse of the encoder if programmed in P0 N790 and then sets the Actual Counter to 0. If P9974 is present in memory, then this program will be executed.

1.1.7 M03, M04, M08, M10, S, T

The key `M03` starts the milling aggregate and the cooling aggregate can be selected with `M08`. Pushing these keys one more time stops the aggregates. If P9936 (Tool Change) is present in memory, this program is called for execution of M03, M04, M08.

S (Spindle Gear) and T (Next Tool) are calling P9936 (with registers #81 and/or #82 set to –0,001). To enable this function P9936 as described in chapter 8 must be present in memory.

1.1.8   Display of Internal Counters

During a movement, certain internal variables describing the status of the axes can be displayed by pushing the following keys:

1   Real Position in the DILAG (increments) as it comes from the incremental transducer

2   Lag Distance in increments (result from target position - real position)

3   Target Position (Increments) as it comes from the Interpolator hardware

4   Distance (increments) of the Gantry axes, if available

5   Actual Value (mm) without corrections such as G54, T

6   Active G54 displacement (mm)

7   Absolute Actual Value (mm) since the last movement to reference

8   Programnumber and Blocnumber of the bloc being executed.

9   Way To Go of the current bloc (mm)

0

Most of these variables only are available on controllers for servosystems.

1.1.9   DNC-mode

With 🖐, EXT , the CNC is put into the DNC MODE and is waiting for a file in Binary Format to be sent through the Serial Interface. This allows very large data files to be executed immediatly without using the internal memory of the CNC.To use this function we offer a plug-in to our DIENSTPROGRAM which converts a CNC-file into Binary Format and sends it via Serial Interface to the CNC. However only G01,G02,G03, G11 are supported.

After having pushed EXT , the text DNC MODE V24 appears on the screen, signaling that this mode is activated. During datatransmission, the Interpolatorcache with 200 blocs is working. Datablocs will be loaded in the background while the contour is beeing executed. With the key STOP, at any moment the execution of the program can be interrupted and restarted again with START.

In order to initiate the DNC MODE in the DIENSTPROGRAM, push the key F5 MENU which switches to a Additional Functions. With ALT-3, the DNC MODE is invoked (PLOTMODE). You input the name of the CNC-file to be executed and push ENTER. The CNC-file will automatically be translated into Binary Format and sent to the CNC.

This function is only available for F-version, so not for turning or grinding versions.

1.2   MANUAL INPUT

In Linux-CNC the Manual Input Mode is includued in the Manual Mode!

By pushing ⊞ the MANUAL INPUT mode is entered. This operating mode allows to input and execute single G-functions and parametrical functions. Also cycles like G87 and programm calls can be executed.

The modal G-functions like G90/G91 or G75 or G54 as well as the Actual Values F, S, T, M will be displayed.The G-function to be executed always refers to these modal G-functions.

Underneath the modal functions, binary data blocs are displayed representing the state of the I/O cards 1 and 2. Each bloc consists of 2 lines. The first line displays the state of outputs 1 - 8, the second line displays the state of inputs 1 - 8. (This display is not available on all CNC's).

The CNC displays the input line:

G00 X........,... Y.......,... Z.......,... in which G is highlighted.

Pushing ➡ positions the cursor to the next data input area, in this case **X,** where new values can be input.The key |START| executes the programmed bloc. The execution of the bloc can be interrupted with |STOP|. A new G-function can be entered.

1.2.1   Tool Change

In the MANUAL INPUT MODE a tool change can be executed with: **G36 F..... S..... T.... M...**
➡ moves the cursor to T where for example 2 can be entered. With |START|, the tool T2 will be activated. The Tool Offset of T2 will be taken from the tooltable P9900 and the Actual Value counter will be actualized.

To use this function P9936 must be present in memory

1.2.2   Jogging Mode

- Switch to incremental mode with G91 by pushing the keys 9,1,➡, |START|
- Select G00 and input a distance in X, Y or Z.
- By pushing |START| the bloc will be executed. By pushing |START| one more time, the bloc will be executed again.

1.2.3   Move to a Position

- Switch to absolute mode with G90 by pushing the keys 9,1,➡, |START|
- Select G00 and input a position in X, Y or Z.
- Move to the desired position with |START|.

1.2.4   Starting a Semicircle

- Switch to incremental mode with G91.
- Select a feedrate with G11 F....... .
- Select G02 or G03 and input the endpoint and centerpoint of semicircle.
  G02 X+..50,000 Y...0,000 I+..25,000 J...0,000
- By pushing |START| the semicircle will be executed.
- A full circle can be programmed with:
  G02 X+...0,000 Y...0,000 I+..25,000 J...0,000

1.3   TEACH IN

1.3.1   Standard Teach In

Before entering the TEACH IN mode, G90 or G91 should be selected in MANUAL INPUT mode. Then TEACHIN can be pushed.The CNC requests the user to input a program number, which can be acknowledged by ➔ . A complete bloc input buffer for G01 will be displayed by pushing 1 and then ➔ .

Programming will be done as follows: Select the moving direction and push START . Move the axes under control of the feedrate potentiometer to the desired position and push STOP .

The Actual Position will be displayed in the bloc buffer. The bloc will be stored in memory by pushing the key .

By selecting the G-address, also another function can be chosen, for example G90 or G05.

After a movement followed by STOP , the X, Y and Z addresses will be updated.

With G05, the R-address can be accessed with ➔ , the desired radius can be input and the bloc can be memorized.

By turning the optional handwheel, the last axis activated with the Direction Keys can be moved. By pushing the same or another key, the Actual Value will be transferred into the bloc input buffer and can be memorized with ⇨ .

1.3.2   Playback

If the key TEACHIN is pressed one more time, the PLAYBACK mode is activated. However the analog joystick (option) is necessary for this operation.

The axes can be moved with the joystick. After each time slot (N904Z in the machine data), the actual position is memorized.

The optional Memory Card of 1 - 4 MB allows also very large programs to be memorized.

Attention: All the programs in memory apart P0 will be deleted !!!

1.4 AUTOMATIC MODE

1.4.1 Starting a CNC Program

When invoking the automatic mode ⬛, the CNC proposes a program number. The proposed program number will be the one processed at last. Another program number can be input if desired. The program will be executed by pushing START . If the program number is acknowledged by ➡, the CNC proposes a bloc number. A different bloc number to start with can be entered if desired.
To acknowledge the bloc number, push ➡. Then the first blocs of the program will be displayed in the lower part of the screen. The actual state of the modal functions as well as input and output states of the optional I/O cards (depending of the CNC used) will be displayed.
With START the automatic program execution is activated and the selected program is started.

1.4.2 Single Step Mode

Single Stepping can be activated by pushing ⬛ during program execution and will become active at the overnext bloc. At the end of each bloc, the CNC requests a new START . Single Step mode can be exited by pushing again.
This mode also can be invoked from the beginning by acknowledging the request for a program number with .

1.4.3 Exiting the AUTOMATIC Mode

The automatic mode is exited by pushing STOP and then MENU .

1.4.4 Error Messages, P9998

If a Limit Switch is detected during program execution or the input EXTERNAL INTERRUPT is activated, the CNC will stop all axes immediately and an error message will be displayed. In addition,P9998, if present in memory, will be invoked and executed. This allows for example to set the I/O's in a definite state or to execute an error handling programm.

1.4.5 Control of Program Execution

· M21 will stop screen updating in the AUTOMATIC mode, allowing a faster program execution between blocs.
· M22 same function as M21, but effective only within programm calls. The execution of the main program will be displayed. Single Step will be disabled during programm calls.
· The programmed speed F can be controlled with the Feedrate Potentiometer, as far as this feature has not been disabled by programming M23.
· M24 suppresses the execution of all following M-functions (except M20 - M28) as well as G04 (dwell time). This allows a program test without machine functions.

1.4.6 Autostart P9999

After switching on the CNC, a check for the presence of P9999 in memory is done. If it is present, it will be executed immediatly.
This program allows customizing the CNC to different needs of the user. For example if the actual display should not be set to zero but should contain the Actual Value of the time before the CNC was switched off, the following program will be used:
P9999
N10 G92 X#111 Y#112 Z#113
After switching on, P9999 will not be executed if ➡ is pressed and hold down.

1.4.7 Error Handling P9998

P9998is invoked if an error like **Limit Switch** or **External Interrupt** is generated in the Automatic Mode. So P9998 can react on errors and reset outputs or whatever is useful. The error number will be put into #40 and displayed with 94 #240.

1.4.8   Actual Value Display

When the axes are moving, the Actual Value is continously updated so that the Actual Position of the axes can be read out. If however the keys 1 - 9 are pressed, other information is represented as described in MANUAL MODE.

1.4.9   Move To Interrupted Progam Point (Option)

If there is a power failure or the CNC was switched off or the Automatic Mode was stopped, an indexnumber representing the just executed bloc is memorized and available again after switching on the CNC. When selecting the Automatic Mode and acknowledging P.... and N.... with , the indexnumber where the program was interrupted, is displayed.

With START , a quick execution of the program **without** movement of the axes is processed and stopped just before the interrupted bloc. Then Single Step is activated. The next START moves the axes with Rapid Traverse to the point corresponding to the Actual Value Display.

The keys M03 and M08 are active and can be selected. With START the execution of the next blocs can be started just as in the normal program run.

Conditions:
- No G92 in the program.
- The physical Program Starting Point of course must be the same
- The indexnumber is not incremented during execution of programs >= P9800.
- M2745 stops the internal counter sampling the indexnumbers, while M2755 enables it again.

1.4.10   Handwheel during Automatic Mode

While a programm is running, the handwheel can be activated by pushing X+, X-, Y+, Y- Z+ Z- keys. See also chapter 1.1.4

Turning the handwheel will generate a remaining offset to the actual position.

1.5   External Data

The operating mode $\boxed{\text{EXT}}$ permits input or output of programs via Serial Interface.
For security reasons, a code is requested if it was set previously in P0 N900.

1.5.1   Program Printout

Select $\boxed{1}$. The CNC proposes a program number for printout. To generate a printing of all the programs in memory, first push $\boxed{C}$ and then $\boxed{\rightarrow}$.
Data is transmitted through the Serial Interface. It can be memorized on a PC and then be printed on the local printer of the PC. The parallel interface port of the CNC is no more supported.

1.5.2   Serial Output Binary

Exporting data in an internal Binary Format through the Serial Interface will be done by selecting $\boxed{2}$. The program number of the program processed at last will be proposed by the CNC. Another program number can be entered. To output the whole memory contents, key $\boxed{C}$ and then $\boxed{\rightarrow}$ must be used.
Our DIENSTPROGRAM is able to recognize automatically this dataformat and to memorize the program with the extension .BIN.

1.5.3   Serial Input Binary

This function is invoked with $\boxed{6}$. The CNC asks for a programnumber. Input $\boxed{C}$ and then $\boxed{\rightarrow}$. The CNC awaits data through the Serial Interface in Binary Format.
BIN-files can be executed immediatly in the PLOTMODE which is available in MANUAL MODE, however only a reduced number of G-functions are allowed.

1.5.4   Serial Output ASCII

Exporting data in ASCII format through the Serial Interface will be done by selecting $\boxed{3}$. The program number of the program processed at last will be proposed by the CNC. Another program number can be entered. To output the whole memory contents, key $\boxed{C}$ and then $\boxed{\rightarrow}$ must be used.
Our DIENSTPROGRAM is able to recognize automatically this dataformat and to memorize the program with the extension CNC.

1.5.5   Serial Input ASCII

This function is selected with $\boxed{7}$. The CNC asks for a program number. Input the desired programm number and confirm with $\boxed{\rightarrow}$. The next programm coming via Serial Interface in ASCII format will be stored under this program number.
If several programs have to be read in, input $\boxed{C}$ and then $\boxed{\rightarrow}$. The CNC awaits data through the Serial Interface in ASCII format up to the last character which mustbe a %.
Data format:
First the programm number f.e P1 must be transmitted.
Then the programm blocs will follow, each MUST start with a bloc number. Each bloc must be terminated with CR (0DH).
After a complete programm or „%" (25H) or a character between (00H) to (08H) can follow.
If one of these characters was recognized, the program first read in will be memorized, an eventually existing program with the same number will be deleted, and the CNC will wait for additional data. If „%" was recognized (end of data transfer) the data transfer is finished.

Example:
P1 CR
N1 G0 X12Y-15 Z+4,15 CR
N2 ....;Commentar CR
....
N999 G1 Z0 CR
% CR
The blocs should be numbered correctly in ascendant order, otherwise they cannot be edited in the INPUT MODE. A commentar starting with ; is allowed, however this commentar is not memorized in the CNC.
The CNC uses the Grafic Ram as temporary storage, so that sometimes a binary image of the data is diplayed.
We offer a DIENSTPROGRAM running in a DOS-Window, allowing to receive, memorize, edit and send back CNC programs. Optional programs for translating CNC programs f.e. HPGL to CNC or ASCII to BIN, are also offered.

Postprocessor for CAD/CAM:
All CAD/CAM Systems allow to generate CNC blocs from a drawing. In order to do that they use a Postprocessor, which is nothing else than a program translating the drawing into CNC-code.
Because most CNC's need a slightly different code, the postprocessor can be configured.
In our case, the user has to define a new postprocessor and name it for example **Engelhardt-Postprocessor**. In the configuration of the Postprocessor the user must define the following rules:
- The CNC code must start with the character P followed by a program number of up to 4 digits defining the program number, followed by CR (Carriage Return). **Example:** P0100 CR. All the characters preceeding the charcter P are ignored
- After that are following the CNC blocs. They always must start with the character N followed by up to 4 digits defining the bloc number.
- After that follows the G-function with 2 digits. Every bloc must contain a G-function.
- After the bloc adresses with there values. Each bloc is terminated with CR. **Example:** N005 G0 X50,050 Y-30 CR. You can use ',' or '.' as decimal point.
- The last character must be % followed by CR without a bloc number.
- The middlepoint I, J of a circle is expected to be always in relative coordinates.
After definition of this **Engelhardt-Postprocessor**, a small drawing generated with the CAD/CAM system of the customer should be converted to CNC-code and transmitted to the CNC.For doing that, a Terminalprogram like Hyperterm can be used.
Preferable is our Dienstprogram. A small edition named MINIDI.EXE can be downloaded without fees from our website www.EngelhardtGmbh.de. This program allows to transmit code to the CNC and to receive code from the CNC for memorization on a PC.

1.5.6	Writing to Discette

Function no more supported

1.5.7	Reading from Discette

Function no more supported.

1.5.8	Writing to PC-DISC or NET-DISC

To support this function, we offer an additional PC-compatible floppydisc station which connects through the Serial Interface to the CNC. The baudrate is fixed to 9600. With $\boxed{5}$ the CNC proposes a program number to be stored on PC-DISC or NET-DISC. The selected program for example P9 is memorized under the file name P0009.CNC.

To output the complete memory contents, key [C] and then [→] must be used. This complete memory content is stored under the file name PALLE.CNC. NET-DISC has an additional Ethernet connection so that programs can also be stored and retrieved from a company server.

1.5.9   Reading from PC-DISC or NET-DISC

With [9], first a directory of the PC-DISC or NET-DISC is displayed in a window. With [→], the directory can be scrolled through. With [↓], the directory can be left and the CNC requests to input a program number which has to be acknowledged with [→]. This program then is read in and memorized. Programs with a program name like P0000.CNC or P0010.CNC or P9999.CNC can be read in this way.
Programs with any name like NAME123.CNC or PALLE.CNC can be read in by clearing the P-field (which then becomes P....) and then pushing [→].

1.5.10   FLASH-EPROM / STATIC RAM

With „0" a copy of the complete memory content is put in a FLASH EPROM (which is optional). If the Flash Eprom is not equipped, a copy of the memory is saved to STATIC Ram.
The programs can be recovered from the FLASH Eprom or STATIC Ram into the CNC memory, by holding the key [C] pressed while switching on the CNC until after approximately 5 seconds the message **CODE** appears on the screen. Input [0] followed by [→] and the programs are restored.
Note: All existing programs of the CNC memory are overwritten by this function. The programs from the FLASH Eprom or STATIC Ram are not added to the existing programs!

1.6   INPUT MODE

1.6.1   Entering Input Mode

The input mode ⧉ allows to input and edit programs. After selection of this mode, a program number will be proposed. However anther program number can be entered and acknowledged with ➜.

If a program with this number already exists in memory, the last blocs of this program will be displayed. By acknowledging with ⬇, the first blocs of the program will be displayed.

If the entered program number does not exist, **N001** will be proposed as a first bloc number. Acknow-ledgement will be made with ➜. The cursor moves to **G..** and a G-function can be entered.

After pushing ➜, the words corresponding to the G-function are displayed.

When all necessary words of the bloc have been entered, the bloc can be stored by pushing ⧉. The bloc number is incremented automatically. An error message will be displayed when trying to memorize an uncomplete bloc.

1.6.2   Alter Bloc

If a bloc already stored must be altered, it can be put into the editing buffer by typing the bloc number and then pushing .The cursor can be moved to the word to be corrected with ➜ .

Then the bloc must be memorized again by pushing ⧉.

1.6.3   Delete Bloc

The bloc to be deleted must be searched with . Then push .⧉ and the bloc is deleted.

1.6.4   Insert Bloc

Type in the new bloc number to be inserted and then push ➜. Select the G-function and complete the bloc. Push ⧉ to store the bloc.

It will be inserted automatically and the following bloc numbers will be incremented.

Note: Blocnumbers in JUMP's or CALL's are not changed automatically !

1.6.5   List Blocs

Entering a bloc number and then ⬇ will display the next blocs starting with the entered number.

1.6.6   List Programs

After selecting the INPUT MODE, an overview of all programs in memory can be displayed by answering the request for a programnumber by pushing **C** and then ➜.

If a program is marked with „!", this Programm has a "CHECKSUM ERROR". In that case the complete memory including P0000 and P9900 must be erased.

1.6.7   Program duplication

If the request for the program number is acknowledged with **C** and then ⧉ the CNC will ask for the program number to be duplicated and for the new program number.

Also P0000 can be duplicated.

1.6.8   Delete Program

List the program and then press the key .⧉. For security, the code defined in the machine data is requested.

1.6.9   Add a program name

To an existing program, a program name can be added. Therefore the program number can be ackowledged with the key ⇨, the program name can be typed in using the charcters on each key and then the key ⇨ pushed again.

1.6.10   Reserved program numbers:

- P0000 Machine data.
- P8000 Text for customer menues.
- P98XX are invoqued by a keystroke in the MENU
- P9900 Tool table
- P9901 - P9996 Cycles
- P9998 Error handling in the Automatic Mode.
- P9999 Autostart.

1.7   Clear Memory Mode

1.7.1   Delete a Program

This mode ✏ allows to delete single programs or to clear the complete memory.
First, the CNC requests the input of a code number. This code number can be defined in the machine data N900X. If the number 0 was selected in the machine data, the input of a code will not be requested. Single programs can be deleted by inputing the program number and then pushing ➡.

1.7.2   Delete Range of Blocs

To delete a range of blocs of a program, enter the program number and then ⬇. The CNC requests the starting and the ending bloc numbers. All blocs in this range will be deleted.

1.7.3   Clear Complete Memory

The whole memory can be cleared by pushing $\boxed{C}$ and then ➡. Here the code defined in the machine is requested, even if the code is 0. P0000 and P9900 however will remain in memory.They can only be deleted individually. The CNC requests the code defined in P0 N900X (default=0).

1.7.4   Clear Memory starting with a program number

If a program number is entered and the key ✋ is pushed, the rest of the memory starting with this program number is cleared.

1.8　Graphic Mode

1.8.1　Entering Grafic Mode

This mode ✳ allows to display the programmed tool path. The CNC asks for the starting program and bloc number. The program used at last and the first bloc within this program will be proposed from the CNC.

1.8.2　Viewing Plane

After that the viewing plane can be defined. All movements underneath this plane will be displayed on the screen. For example choice of plane Z = -10,000: all movements above Z = -10,000 are not displayed.

1.8.3　Zoom Value

Finally the Zoom value must be input. With a value of 2, the dimension of the drawing will be double, with a value of 0,5, the dimension will be half.
Then the display will be cleared and a cross wire appears. This cross wire defines the starting point of the program. It can be moved on the screen by the following keys:

up

2

left　4　　　6　right

8

down
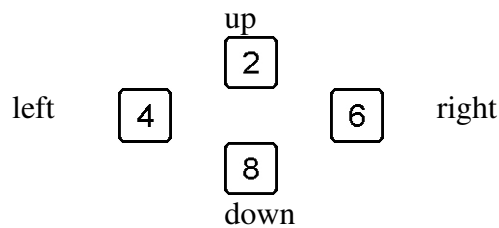
If the position of the start point on the screen is OK, push ➜ to display the contour. After that, the cross wire can be displaced to draw the picture once again.
With ✳ the GRAFIC MODE is invoked again and the starting parameters be changed.

1.8.4　Miscellaneous

- G04 and all M-functions won't be executed.
- Attention with G20! If for example in P0001 a jump to P0001 N001 is programmed, the graphic mode will execute this program continousely. The execution can be interrupted by pushing MENU.
- The graphic mode uses an internal Actual Value counter, which will be cleared by invoking the graphic mode. At the end of the contour, this counter will be displayed.
- First the programmed path is displayed. In a second run, the correct path can be displayed by pushing START.

1.8.5　Grafic Example

P9900	TOOL TABLE			
N001		X.....,...	Y.....,...	Z.....,...	R...10,000

P0001	TESTPROGRAM			
N001	G91	Incremental input		
N002	G11	F...200	T......1	
N003	G01	X+...30,000	Y+ ...20,000	
N004	G01	X	Y	Z-..10,000
N005	G01	X+...50,000	Y	
N006	G01	X	Y+....30,000	
N007	G03	X-....50,000	Y+......0,000	I-......25,000	J+......0,000
N008	G01	X	Y-.....30,000	

```
N009   G00    X.......         Y .......            Z+...10,000
N010   G00    X-....30,000     Y-.....20,000
```

2 PROGRAM STRUCTURE

Each program consists of a program number and up to 999 or 9999 blocs,depending on the version. Each bloc has a bloc number N and a G-function G. This function tells the CNC what to do in this bloc, for example a linear interpolation or a movement to the reference point.

2.1   THE G-FUNCTIONS

This part will explain the G-functions and the corresponding bloc structures. The CNC can be equipped with less G-functions depending on the control purchased.
A list of the implemented G-functions can be displayed with the key in the MENU of the CNC
If the G-funktion is only to be executed in the GRAPHIC mode, the key ⊛ can be pressed when the G-input field is active. The bloc then looks like that: **N0001*G...**
ATTENTION: The bloc stucture X,Y,Z,U ... can be different depending on the type of CNC. When turning or grinding, the structure XZYC often is used. For that, only the characters on the screen display are changed! It is important to know that in all cases the axes with the number 1,2,3,4 are referred.

2.1.1   G00 RAPID TRAVERSE

N... G00 X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
All axes can be moved together. The feedrate is determinded by Fmax in the machine data P0000.

2.1.2   G01 LINEAR INTERPOLATION

N... G01 X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
Up to 8 axes can be moved together. The feedrate (mm/min.) can be programmed before the bloc with G11.

2.1.3   G02 CIRCLE INTERPOLATION CLOCKWISE
2.1.4   G03 CIRCLE INTERPOLATION COUNTERCLOCKWISE

N... G02   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,... I.....,...J.....,...
XY, are the endpoint of the Circle Segment, IJ are the coordinates of the circle center. The circle center must be programmed relative to the starting point of the Circle Segment, even if the endpoint is defined in absolute coordinates. Of course, the programmed endpoint must be a part of the circle. This is the case, when $(X-I)^2 + (Y-J)^2 = I^2 + J^2 = R^2$.

A complete circle can be programmed as follows:
N... G02   X.....0,000 Y.....0,000 Z.....,... I.....20,000 J....0,000
A helix interpolation will result if in addition one or more linear axes are programmed.

Because only 4 digits can be programmed before the decimal point, the maximum programmable radius can be 9999,999 mm.
N... G02   X.....0,000 Y.....0,000 Z.....,... I9999,9999 J....0,000
The following programm will make a circle with a radius of 99 meters:
N... G75   X...10,000 Y...10,000
N... G02   X.....0,000 Y.....0,000 Z.....,... I9999,9999 J....0,000
N... G76

2.1.5   G04 DWELL TIME

N... G04 H....,...
Dwell time programming between 0,010 and 9999,990 seconds.

2.1.6	G05 CIRCLE CW WITH INPUT OF RADIUS
2.1.7	**G06 CIRCLE CCW WITH INPUT OF RADIUS**

N... G05   X.....,...   Y.....,...   Z.....,...   U.....,...   V.....,...   A.....,...   B.....,...   C.....,...   R.....,...

Input the desired endpoint in X and Y, the radius in R. The sign of R determines if a small or large Circle Segment is generated.

The transformation into a G02/G03 bloc needs calculation time during AUTOMATIC MODE. G05/G06 is not usable for fast bloc cycle times.

2.1.8   G07 CIRCLE DEFINED BY ANGLE

N... G07   X.....,...   Y.....,...   Z.....,...   U.....,...   V.....,...   A.....,...   B.....,...   C.....,...   R.....,...   W.....,...

The Circle Segment is defined by the radius R and the ending angle W. The starting point of the circle is tangent to the preceding bloc. X, Y have no meaning, the other axes can interpolate with X,Y. Only in G91 and in G40 !!

2.1.9   G08 ASYNCHRONOUS MOVEMENT

N... G08   X.....,...   Y.....,...   Z.....,...   U.....,...   V.....,...   A.....,...   B.....,...   C.....,...   F .....,...   W.....,...

At the same time as an interpolation in progress, the axes here defined begin to execute the programmed distance. F defines the speed, W the number of repetitions of the movement. Function G13 M91 interrupts G08. M820x waits until Asynchronous Axis x has finished. The programmed values are always in G91.

Examples:
- G08   X100   F100   W99   :   100 oscillations with F100
- G08   X100   F200   W0   :   asynchronous displacement by 100 mm
- G08   X0   F300   W0   :   X is running endlessly with F300 in positive direction
- G75   X-1

   G08   X0   F300   W0   :   X is running endlessly with F300 in negative direction

2.1.10   G09 SKIP REST OF TRAVEL

N... G09   X.....,...   Y.....,...   Z.....,...   U.....,...   V.....,...   A.....,...   B.....,...   C.....,...   M....

The linear interpolation will be done just like a G01 bloc depending on G90/G91. However, if the input programmed with M (M161-168, M171-178) becomes active, the interpolation will stop and the next program bloc will be executed. In addition the NZ Flag is set. This flag can the interrogated with the function 54.

Possible uses:
- recognization of tool fracture
- digitization of workpieces

2.1.11   G10 CORNER ROUNDING

N... G10 X....,...   Y....,...   X....,...   Y....,...   R....,...

Examples:

In the incremental mode G 91, the 1. XY input field is to be programmed with the first line, in the 2. XY the second line and in R the radius of the corner between these lines.

N001 G91
N002 G10 X.100,000 Y...0,000 X...0,000 Y.100,000 R..25,000
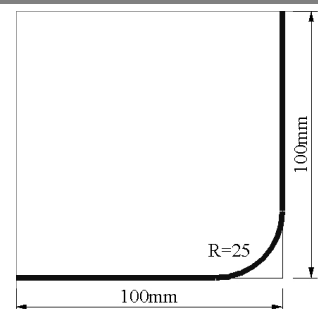
If the 2. XY input field is not programmed, R is interpreted as rotating angle for the 1. XY distance.

N001 G91
N002 G10 X.100,000 Y...0,000 X....,... Y....,... R..45,000

When memorizing G10, the resulting CNC blocs are calculated immediatly and memorized already in the INPUT MODE.

2.1.12   G11 ADDITIONAL FUNCTIONS F, S, T, M, B

N... G11 F...... S...... T..... M..... B.....
These functions allow programming of F, S, T, M and B. F is the speed of the consecutive movements.
- F is the feedrate of the next movements
- S is the spindle speed which will be active with the next M03 command.
- T is a tool from the tooltable P9900.
- M is M-function allowing for example to activate an output.
- B is factor defined in percent which will override the actual acceleration and deceleration values defined in P0000 N703 and N704. It can be disabled by G11 B100 giving 100% again.

For G11 Fxxxx, no Stop will be generated between 2 blocs, the movement will be continous. G94 has the same function but with a stop between the blocs.

2.1.13   G12 ADDITIONAL M-FUNCTION

N... G12 M.... M.... M.... M.... M.... M.... M.... M....
As G13, but output of M-function on the fly without stopping the actual movement. This function does not work when the Interpolator Cache M32 (see page 39) is activated. Also it is not senseful to program a Wait For Input Function!

2.1.14   G13 ADDITIONAL M-FUNCTION

N... G13 M.... M.... M.... M.... M.... M.... M.... M....
G13 allows to program several M-functions in one bloc.

2.1.15   G17 PLANE XY (Initial State)
2.1.16   G18 PLANE XZ
2.1.17   G19 PLANE YZ

N... G17 XY PLANE
This modal function will switch to the XY plane. After reset, G17 will be effective and will be overwritten by G18 or G19. During path compensation (G40 - G42), the plane must not be changed.
This function alows to define the plane in which a circle is executed. The center of the circle is always programmed in I, J, also in the plane XZ and YZ.

2.1.18   G20 JUMP TO PROGRAM

N... G20 P.... N....
This function executes a jump to the program **P** and continues with the starting bloc number **N**. If only **N** is programmed, the jump will be done into the actual programm. If only **P** is programmed, the jump will be done to the first bloc in program **P**.
This function also allows to jump to a bloc defined by a label, see page 54

2.1.19   G22 CALL PROGRAM WITH REPETITION FACTOR

N... G22 P.... N.... W....
The program „P" will be called, starting with bloc number „N". It will be repeated as programmed with „W". If the program is to be executed only one time, W00 must be programmed. Up to 6 programm calls can be stacked.
Note: An error „too many calls" will be reported in the following case:
P0010

N001 G..
....
N010 G22 P0010

2.1.20   G23 JUMP/CALL PROGRAM WITH REPEAT AND CONDITION

N... G23 P.... N.... W.... M....

Program **P** will be called, if condition **M** is true. If **W** is not programmed, a jump to program **P** will executed. The condition **M** could be for example M161 which will wait for input 1 to come active. Only then the jump or call will be executed.

2.1.21   G33 THREAD

N... G33 X....,... Y....,... Z....,... U....,... V....,... A....,... B....,... C....,... I....,... J....,...

G33 allows synchronisation of the programmed axes with an external pulse generator. I must be programmed with the pitch, J with the acceleration and deceleration distance.
If J is negative, G33 will not wait for the reference pulse for starting.
If I is negative, the pitch I is interpreted as beeing programmed in Inch.

Testprogramm:
N001 G91
N002 G33 X........   Z....10,000
N004 G00 X...5,000
N005 G00 X........   Z-...10,000
N006 G00 X-...5,000
N007 G20 N001

2.1.22   G36 TOOL CHANGE

N... G36   F...... S...... T...... M...... B......

The pogrammed values F,S,T,M,B are put into the registers #080 - #084 and then program P9936 is called. Here the customer can store his own tool change program.

2.1.23   G40 RADIUS CORRECTION OFF (Initial State)

N... G40 correction off This function will reset G41/G42. The next programmed **linear** interpolation in the XZ plane will be used to exit the tool path.

2.1.24   G41 RADIUS CORRECTION RIGHT
2.1.25   G42 RADIUS CORRECTION LEFT

For correct use of the toolpath compensation, the following notes must be observed:
- G41, G42 works in the XY plane, the tool length compensation works on the Z axis.
  Before using a path compensation, an appropriate tool (P9900) must be programmed.
- G41 compensates always lefthand, G42 always righthand in the moving direction of the tool.
- The compensation must be programmed one bloc before the bloc to be corrected. This bloc, which must be a linear interpolation, will then be used to enter the compensated path.
- G40 will switch off path compensation. The following linear movement in XY is used to exit the compensated path.This movement is a part of the contour!
- The inner compensation can only be done if the milling cutter is already placed within the programmed contour.
- During the compensation absolute or incremental input can be programmed. It is also possible to call subroutines (G22), however the called programm must contain at least one G01 movement.
- If the last program bloc is reached without encountering G40, the path compensation mode will be exited automatically.
- Jumps with condition (G23 M....) as well as parametrical functions should not be used during G41/G42

- The error message CORRECTION NOT POSSIBLE will be displayed if between 2 blocs there are more than 255 blocs including no movement are executed.
- During G41/G42 it is not allowed to move the Z axis together with X or Y at the same time. So, the following rest of a program with toolpathcorrection is not allowed:
  G40 COMPENSATION OFF
  G90 ABSOLUTE INPUT
  G00 X...0,000 Y...0,000 Z...0,000

Example for Tool Path Compensation:
P9900   Tool table
N001   X....0,000   Y....0,000   Z....0,000   R....10,000

P0001 Test program1
N0001 G11 F...200   S......   T...1
N0002 G42 RADIUS CORRECTION RIGHT
N0003 G01 X+..30,000 Y+..20,000
N0004 G01 X+..50,000 Y.....,...
N0005 G01 X.....,...   Y+..30,000
N0006 G03 X-..50,000 Y+...0,000 I-..25,000 J+..10,000
N0007 G01 X.....,...   Y-..30,000
N0008 G40 CORRECTION OFF
N0009 G01 X-..30,000 Y-..20,000

P0002 Circle with radius correction
N001 G11 F...200   S.........   T......1
N002 G42 Radius correction right
N003 G01 X-..10,000
N006 G02 X-....0,000 Y+.....0,000 I+..10,000 J+...0,000
N008 G40 Correction off
N009 G01 X+.10,000

In the graphic mode, P0001 and P0002 can be executed. First the programmed path is drawn with a straight line, by pushing START again, the corrected path is drawn with a dashed line.

2.1.26	**G53 DISPLACEMENT OFF (Initial State)**
2.1.27	**G54 DISPLACEMENT ON**

N... G54   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
With G90, the values programmed in G54 will be added to all the following movements. With G91, the displacement will only once be added to the first movement in the apropriate axis.
Example:
P0010
N001 G90 Absolute input
N002 G54 X.....0,000 Y....0,000
N003 G00 X.....0,000 Y....0,000
N004 G01 X...20,000 Y....0,000
N005 G01 X...20,000 Y...20,000
N006 G01 X.....0,000 Y.....0,000
If P0010 now should be executed in the position (100,50), bloc N002 should by changed to:
N002 G54 X...100,000 Y...50,000
During programexecution, the Actual Value Display shows the programmed values and not the real values.

2.1.28	**G55 DISPLACEMENT**

N... G55   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
As G54, however it must be switched off with G55 X0, Y0, Z0. This function cannot be used together with G86 in the TURNING MODE.

2.1.29	G58 MEMORIZE ZERO POINT

N... G58   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
With G58, the Zero Point of the workpiece can be memorized. After switching on the CNC, the axes then can move automatically to this Zero Point.
To enable this function, the following program must be put into memory:

P0074 REFERENCE POINT   ;It must be P0074
N001 G11 T 0   ;T0 must be selected with G11 or alternatively with G36
N010 G74 Z 0   ;The positive limitswitch must be moved to!!!
N020 G74 X 0   ;The positive limitswitch must be moved to!!!
N030 G74 Y 0   ;The positive limitswitch must be moved to!!!
N040 G92 X0 Y0 Z 0   ;This bloc number must be N40 !!!!

After switching on, P0074 must be called and a tool must be selected with G11 or G36. This can be done for example by programming in MANUAL INPUT MODE: G22 P0074 and then ⓢᵀᴬᴿᵀ followed by G11 T0001 and another ⓢᵀᴬᴿᵀ.
After that, one axis for example X, is moved to the Zero Point of the working piece. Then, in MANUAL INPUT MODE, G58 X0 is input and executed with ⓢᵀᴬᴿᵀ. After that, the same is done with the other axes and the workpiece Zero Point is memorized. The determined values are deposited in bloc N40. With the next call of P74, these values are finally put into the Actual Value display and the axes will be again at the desired Zero Point.

2.1.30	G59 PUT T IN MEMORY

N... G59   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...

Tool Measurement
First, in P9900, the tools are defined with X0, Y0, Z0 and R with the corresponding radius. The **longest** tool is taken as T001 and this is the reference tool with the lenght Z0.
In the Manual Mode this tool is moved down to the surface of the working piece by using the Jogging Mode and the handwheel. When the surface is touched, the Actual Value Display of Z is set to Zero with the key 'C'.
Now, with G36 T002 and ⓢᵀᴬᴿᵀ, the tool T002 is activated and moved down until the workpiece surface is also touched. Because this tool is shorter, the Actual Value diplays a negative value corresponding to the lenght difference to tool T001.
In MANUAL INPUT, G59 now must be input and the cursor positioned to Z. With START, the lengh difference in Z is transferred to P9900 T002 Z. The Actual Value displays now Z0, because the tool touches the working piece surface and the toollengh compensation is activated (by having programmed G36 T002 or G11 T002).
It is recommended to take the longest tool as T1 and to make the move to the reference point with T1 activated. So when switching to T2 which is smaller, this will lead to a correction movement into the direction of the working piece and not to the direction of the LIMITSWITCH which could be touched and would interrupt the program flow.

Hint: G59 calls P9959 in the MANUAL INPUT MODE and transmits the actual cursor position to #79. (The cursor position for X is 0,010, Y is 0,021, Z is 0,032...)
P9959 will put the actual position of the adressed axis into the corresponding adress of the activated tool. (only for X, Y, Z)

2.1.31	G67 SOFTWARE LIMIT SWITCH -
2.1.32	G68 SOFTWARE LIMIT SWITCH +

N... G67   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
If the limits which were programmed with G67/G68 are exceeded, the CNC will stop and display the error message "SOFTWARE LIMIT SWITCH".
If the destination point of a programmed bloc falls behind the software Limit Switches, also the error message is displayed and the CNC will not execute this bloc.
The software Limit Switch is disabled by programming G67 X0 Y0 Z0 .and G68 X0 Y0 Z0
During an active G67/G68, G92 must not be programmed!

2.1.33   G74 REFERENCE POINT

N... G74   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
This function moves the programmed axes to the corresponding Limit Switches while the direction will be determined by the sign of the programmed value. This value will be set into the Actual Value counter.
Example:
N... G74   X....0,000
N... G74   Y....0,000
N... G74   Z-...1,000
X, Y will be moved to the positive, Z to the negative Limit Switch.
Note: The axes always must be moved **separatly** to the reference point.
It is recommented to generate a program P0074, which will always be invoked to move X, Y and Z to their reference point and to put the actual counter to the correct value for X, Y and Z. See also G58.

2.1.34   G75 SCALE FACTOR ON
2.1.35   G76 SCALE FACTOR OFF (Initial State)

N... G75   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,... W.....,...
This modal function allows enlarge, decrease and mirror the following programs. All the following movements will be multiplied by the values programmed in X, Y and Z. A negative value will switch the polarity of the movement resulting in a mirroring of the contour.
Under W, a rotation angle can be programmed. A contur in X, Y will be rotated by this angle.In the Actual Value Display however, small rounding errors can appear at the end of the contour.

2.1.36   G78 FREE CYCLE

N... G78 X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
Will put the values X, Y, Z, U, V, A, B, C into the registers #80 - #87 and call P9978.

2.1.37   G79 FREE CYCLE

N... G79 X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
Will put the values X, Y, Z, U, V, A, B, C into the registers #80 - #87and call P9979.

2.1.38   G80 CYCLE OFF (Initial State)

Switches off G81 - G83 .

2.1.39   G81 FREE MODAL CYCLE

N... G81   X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
Will put the values X, Y, Z, U, V, A, B, C into the registers #70 - #77.
After each movement then the program P9981 will be invoked.
Example:
P9981
N005 G11 F1000   T0001
N010 G85 X...40,000 Y...20,000 Z-..10,000 Q-...4,000 V....1,000

P1
N010 G81 Z-....0,000
N011 G00 X10
N012 G00 X20
N013 G00 X30
N014 G80 CYCLE OFF

After each G0 a pocket defined by G85 will be milled.

2.1.40   G82 DEEP DRILLING

N... G82 Z.....,... Q.....,... V.....,... H.....,... F.....,...

Input Data:
- Z =   Depth calculated from work piece surface (negative value)
- Q =   Infeed (negative value)
- V =   Security Distance above work piece surface (positive value)
- H =   Dwell time
- F =   Feedrate

Cycle:
- with Rapid Traverse to the work piece surface - 0,5 mm
- with feedrate to the depth Q and dwell time H
- with Rapid Traverse back to Security Distance V
- with Rapid Traverse to the depth Q - 0,5 mm
- with feedrate to 2Q
- .
- .
- the remaining distance is executed
- dwell time H
- Rapid Traverse back to Security Distance V

Example:
N005 G00 Z1   ;Move to Security Distance
N010 G82 Z-...10,000 Q-..4,000 V....1,000 H....1,000 F.....,...
N011 G00 X10
N012 G00 X20
N013 G00 X30
N014 G80 CYCLE OFF

G82 is a modal cycle. A bloc with this function does not execute a Deep Drilling Cycle! So bloc N010 will not result in any movement. However after each following bloc N11 to N13, a Deep Drilling Cycle is executed until the cycle is switched off with G80 in bloc N14.

If a Deep Drilling Cycle is to be executed at the surface Z50, then the program looks like:
 N005 G00 Z51   ;Move to Security Distance above Z50
N010 G82 Z-...10,000 Q-..4,000 V....1,000 H....1,000 F.....,...
N011 G00 X10
N012 G00 X20
N013 G00 X30
N014 G80 CYCLE OFF

2.1.41   G83 TAP DRILLING

N... G83 Z.....,... K.....,... F......   Tapdrilling

Input Data:
Z: depth
K: pitch

F: feedrate

G83 calculates the corresponding spindle speed automatically according to the programmed feedrate F. If the resulting spindle speed is lower than 60 rot/min, an error message is displayed.

G83 is a modal function and can be switched off with G80.

2.1.42   G85 POCKET CYCLE

N... G85 X.....,... Y.....,... Z.....,... Q.....,... V.....,...

Input Data:

- X: Length of Pocket (value must be greater than 4*R of the activated tool)
- W: Width of Pocket (value must be greater than 4*R of the activated tool)
- Z: Depth of Pocket (negative value)
- Q: Infeed (negative value)
- V: Security Distance over working piece (positive value).The milling cutter must be positioned at this distance above the **center** of the pocket.

Cycle:

- move to starting point of the pocket
- Rapid Traverse to surface -0,5 mm
- with feedrate to Q
- mill outside frame of pocket
- move back to starting point +0,5 mm in X and Y
- mill pocket content in meander form
- move back to security distance
- repeat until desired depth is reached
- move to security distance

Example:

N005 G11 F1000   T0001

N010 G85 X...40,000 Y...20,000 Z-..10,000 Q-...4,000 V....1,000

The pocket can be milled in the opposite direction by first programming G75 X-...1,000 DIRECTLY before G85 and G76 DIRECTLY after G85.

If V is input with a negative value, only the outer contour is milled

2.1.43   G86 CIRCLE SEGMENTATION

N... G86 X.....,... Y.....,... Z.....,... D.....,... O...... P......

Input Data:

- X: Starting Angle of the Circle Segment
- Y: Ending Angle of the Circle Segment
- Z: Linear movement, which also will be divided in O segments
- D: Diameter of the circle. If D is input as negative value, P will also be called at the Starting Angle of the Circle Segment.
- O: Number of segments
- P: After each segment, the program P will be called. This program again may contain a cycle G82, G85, G87. If P0 is programmed, no program at all will be called at each segmentation.

Cycle:

- move to first segment
- call program P
- ....
- move to next segment
- call program P a last time

Example:
P0001
N010 G86 X....0,000 Y...90,000 Z....0,000 D...50,000 V...4,000 P 2

P0002
N001 G00 X....5,000
N002 G00 X-...5,000
Note:
The actual angle is stored in parameter #46. The program P can use it for further calculations. G86 can be used with an active scaling factor G75 (f.e. X....2,000). The circle segmentation is extended to an ellipse.

2.1.44   G87 CIRCLE POCKET

N... G87 D.....,... Z.....,... Q.....,... V.....,... A.....,...
Input Data:
- D: Diameter of the Circle Pocket
- Z: Depth of Pocket (negative value)
- Q: Infeed (negative value)
- V: Security Distance over working piece (positive value).The milling cutter must be positioned at this distance above the **center** of the pocket. If V is programmed with a negative value, only the outer contour of the circle pocket is milled.
- A: Starting diameter, used when f.e. an existing hole must be enlarged.

A tool with a radius > 0 must be activated.

Cycle:
- with Rapid Traverse to surface -0,5 mm
- with feedrate to Q
- mill circle pocket beginning at the center
- move to starting point
- with feedrate to Q
- and so on

Example:
G92 X....0,000 Y....0,000 Z....1,000
G11 F1000 T1
G87 ...50,000 Z-..10,000 Q-..10,000 V....1,000 A....0,000

The pocket can be milled in the opposite direction by first programming G75 X-...1,000 DIRECTLY before G87 and G76 DIRECTLY after G87.

2.1.45   G88 LINEAR SEGMENTATION

N... G88 X.....,... Y.....,... Z.....,... O...... P......
Input Data:
- X, Y, Z: Ending points of the Linear Segmnet
- O: Number of segments
- P: After each segment, the program P will be called. This program again may contain a cycle G82, G85, G87. If P0 is programmed, no program at all will be called at each segmentation.

Example:
P0001
N010 G88 X...50,000 Y...30,000 Z....0,000 O.....7 P.....2

P0002
N001 G00 X 5,000
N002 G00 X -...5,000

2.1.46   G89 MATRIX

N... G89 X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... P.....,...
- X,Y: distance of the matrix-positions in X and Y
- Z: Infeed in Z-direction
- U: Angle of the Matrix against the positive X axis
- V: Angle of the Matrix against the positive Y axis
- A: Number of positions in X direction
- B: Number of positions in Y direction
- P: Program to be called at each Matrix point. This program P can contain cycle like G86 and G88 which again calls a G82 or G85 or G87.

This function allows to arrange a program P at the intersection points of a Matrix whose rows and columns can have an angle against the X and Y axes. The starting point of the Matrix is the the position where G89 is invoked.
The smallest possible Matrix is a line with 2 positions (2 x 1 Matrix)

2.1.47   G90 ABSOLUTE INPUT

N... G90 Absolute Input
This function switches from incremental to absolute input. All the following blocs will be interpreted as absolute values.

2.1.48   G91 INCREMENTAL INPUT (Initial State)

N... G91 Incremental Input
This function switches from absolute to incremental input. All the following blocs will be interpreted as incremental values.

2.1.49   G92 SET ACTUAL VALUE

N... G92 X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
The programmed values are taken into the Actual Value counter. If G54 or a tool is active, these values are calculated to the actual counter. So, the value programmed in G92 will not necessarily appear on the screen.

2.1.50   G94 FEEDRATE IN MM/MIN

N... G94 F........ S........ T........ M........ B........
Similar to G11, but it is not executed on the fly.

2.1.51   G95 FEEDRATE IN µM/Rot

N... G95 F........ S........ T........ M........

2.1.52   G97 CONSTANT SPINDLE SPEED

N... G95 F........ S........ T........ M........
The spindle speed is programed in the rotations per min, which is the initial state when switching the CNC on.

2.1.53   G98 SPLINE (OPTION)

N... G98 X.....,... Y.....,... Z.....,... U.....,... V.....,... A.....,... B.....,... C.....,...
The Spline Function generates a smooth curve through all the programmed points. A Spline consists

of a starting bloc, the spline blocs and an ending bloc. In addition it is recommended to activate the Interpolator Cache with M2342.

P0001 Testprogram Spline
N001 G13 M2342
N002 G01 X+.50,000 Y+.50,000
N003 G98 X+.50,000
N004 G98 Y+.50,000
N005 G98 X-.50,000
N006 G98 Y-.50,000
N007 G01 X-.50,000 Y-.50,000

The Spline can also be switched on with M2341 and switched off with M2351.

The axes for Spline Interpolation must be defined in P0 N790 (page 66) as Spline Axes and also as Main Axes.

2.2   M-FUNCTIONS

M-Functions will not be executed in the Grafic Mode

2.2.1   M00 Programmed Stop

Programexecution will be interrupted and the CNC waits for START to be pushed.

2.2.2   M01 Programmed Stop with acoustic signal

As M00, however in addition an acoustic signal will be generated (if available in the CNC).

2.2.3   M02 Program End

The program will be stopped. At the end of a program, it is not necessary to program M02. The CNC recognizes itself that there are no more blocs following.

2.2.4   M03 Spindle On clockwise
2.2.5   M04 Spindle On counterclockwise

Output M03 or M04 is set accordingly in order to switch the spindle.

2.2.6   M05 Spindle Stop

The output M05, if available, is set until the speed of the spindle becomes 0. To use this function, the spindle encoder must be connected.

M08   Cooling On
M09   Cooling Off
M10   Clamping On
M11   Clamping Off
M15   Acoustic Signal (if available)
M16   wait for „Input 1" to go active
M17   wait for „Input 1" to go inactive
M18   wait until no more key is pressed
M19   Wait until interpolating axes have stoped
              Hint: G13 M19 will stop the bloc look ahead function.
M21 - M28 See M22xx
M30   Program End
M31 - M38 See M23xx
M41/M51 Movement without acceleration or deceleration. Gives continous movement also with
              non- tangent blocs. It is the responsability of the user to switch on and off this function c
              orrectly. This function corresponds to [93 #41, which however is executed on the fly
M47/M57 Switch ON / Off the function AUTOSPEED
M48/M58 Disable /Activate P0 N905C in LINUX CNC

M61 - M89 Call of P0061 - P0089 (just like G22 P0061 to G22P0089)
M90 Oscillation (G08) stop at the next reversion
M91 Asynchronous Axis (G08) immediate stop
M97 Wait until all axes are In Position

2.2.7   M-Functions for I/O Cards

M0121 - M0128   test if output 1 - 8 is set and set NZ-Flag

M0131 - M0138	test if input 1 - 8 is active and set NZ-Flag

With M0121 to M0138 a jump NZ (□54) can be executed if the input or output is active.

M0140	set all outputs on I/O card 1
M0141 - M0148	set output 1 - 8 on I/O card 1
M0150	reset outputs 1-8 on I/O card 1
M0151 - M0158	reset output 1 - 8 on I/O card 1

M0160	wait for all inputs to go active
M0161 - M0168	wait for input 1 - 8 to go active
M0170	wait for all inputs to go inactive
M0171 - M0178	wait for input 1 - 8 to go inactive

The waiting functions M16, M0x60 - M0x68 and M0x70 - M0x78 cannot be skipped with START.
This however can be allowed by programming M2347.

M0180	invert all outputs on I/O card 1
M0181 - M0188	invert output 1 - 8 on I/O card 1

The I/O cards 2 - 8 can be programmed with M02xx to M08xx.
To use these additional functions the CNC must be equiped with the corresponding I/O cards.

2.2.8   Wait until Key Pressed

M0900 – M0949	Wait until key xx is pressed. See table on page 58

2.2.9   M-Functions for Leadscrew Error Compensation

M2101 - M2108	Set Machine Zero Point for Leadscrew Error Compensation for the axes X - C (page 63)

2.2.10   M-Functions for Servomotors

Use of the signals MOTOR ON for servoamplifiers

M2120	Activate the signal MOTOR ON to all axes with encoderinput **active**
M2121 - M2128	Activate the signal to the axes X - C with encoderinput **active**
M2130	Remove the signal MOTOR ON for all axes with encoderinput **inactive**
M2131 - M2138	Remove the signal for axes X - C with encoderinput **inactive**

M2140	Remove the signal MOTOR ON to all axes with encoderinput **active**
M2141 - M2148	Remove the signal to the axes X - C with encoderinput **active**
M2150	Activate the signal MOTOR ON for all axes with encoderinput **active**
M2151 - M2158	Activate the signal for axes X - C with encoderinput **active**

2.2.11   M-Functions for AUTOMATIC MODE

M2241	(M21)	bloc updating off in the automatic mode.
M2242	(M22)	bloc updating off in program calls
M2243	(M23)	Feedrate potentiometer off
M2244	(M24)	test run without G04 and M-functions
M2245	(M25)	test run with Rapid Traverse
M2246	(M26)	keyboard off
M2247	(M27)	wait for IN POSITION (see also P0 N803)

M2248	(M28)	Actual Value display off. Display remains active during Single Step. If in addition M2242 is set, the Actual Value of the interpolator is completely ignored speeding up the bloc update time.
M2251 - M2258		will reset the preceeding functions.

M2341	(M31)	Spline on (Option)
M2342	(M32)	Interpolator Cache for 200 blocs on. On the screen in the status line the number of blocs in the Interpolator is displayed. When the cache is full, M2241 and M2248 are ignored.
M2343	(M33)	feedrate potentiometer off with G00
M2344	(M34)	when pressing „MENU" P9999 is invoked
M2345	(M35)	
M2346	(M36)	
M2347	(M37)	the „wait for input" can be skipped by pressing „START" or "MENU"
M2348	(M38)	The Actual Value of the U-axis will be added to the Z- axis
M2351 - M2358		will reset the preceeding functions.

M24xx	refer to the machinedate P0 N902X, see page x
M25xx	refer to the machinedate P0 N902Y, see page x
M26xx	refer to the Grafic Flag

M2741	When M32 is active, the programmed movement is started only when the Interpolator Cache is full or M2751 was executed.
M2742	Display the destination register of parametrical functions continously
M2743	At the end of a program, the CNC does NOT wait until all axes are standing still
M2744	Interupt a G74 when a not programed axis has reached the limitswitch and continue with the next bloc. See also P0074 for Moving to Referencepoint of Gantry axes.
M2745	FPROP OFF/ON
M2746	Activate Bolzen (Option)

M4xxx	As M-Function programmable Dwell Time in 1/100 Seconds. (f.e. 1 s ~ M4100)

2.2.12   Gantry Axes

M80AB	axis **A** will be master, axis **B** will be slave and follow the axis **A**. Axis **B** must have have the same machine data as axis A.
Example:	M8023 will connect the Z-axis (3) to the Y-axis (2). Disconnection with M8033 (which will connect Z to itself).
Hint:	This function must not be invoked with G12.

2.2.13   Plane for Circular Interpolation

M81AB	A following G02/G03 takes place between the axes A and B. M8112 corresponds thus to G17.

2.2.14   Asynchronous Axes

M820x	Wait until Asynchronous Axis x has finished

2.2.15   S-Output

M9xxx	Value xxx from 0 to 255 is put directly to the spindle output generating a voltage between 0 and 10V.

2.3 FEEDRATE

The feedrate will be programmed with the F-function. Possible values are 1 to 999999 mm/min. The CNC will only move at values below or equal to Fmax defined in the machine data.

Example:

N... G11 F1000
N... G01 X..100,000 Z..100,000

The X and Z axes will not move with 1000 mm/min. Each, but only with 1000:1,4=714 mm/min. Because both axes are moving, the resulting feedrate will be 1000 mm/min.

2.4 SPINDLE SPEED

2.4.1 Spindle Speed

The spindle speed will be programmed with the S-function.Possible values are 0 to 60000 in r.p.m. The CNC will only accept values below or equal to Smax defined in the machine data memory. The output SPEED of connector X23B (option) gives a voltage between 0V (= S0000) and 10V (= SMAX) and proportional to the programmed spindle speed S. To update this output, program G11 S....... M03.

2.4.2 Analog Output Card

The optional Analog Output Card allows the programation of 4 analog outputs of 0-10V and 4 Enable Outputs (optocoupled, 10mA) by programing G11 Sxxxxx.
The first digit is the channel number 1-4, the last 5 digits are the corresponding spindle speed from 0 - 60000. The Enable Output is set when the programmed spindle speed exceeds 0 rpm. Sx00000 resets the Enable Output and sets the output voltage to 0V.
The maximal speed corresponding to 10V output voltage is set in P0 N913 XYZU.
Examples:

- G11 S102000   ;1. channel, S=2000 rpm
- G11 S400150   ;4. channel, S=150 rpm
- G11 S200000   ;2. channel disabled

2.5 TOOL CALL

2.5.1 Tool Call

With the T-function up to 99 tools (T01 - T99) can be programmed. These tools will be defined in P9900 with N0001 to N0099. Invoking G41, G42 the data of the just activated tool will be read out of the tool table P9900. If another tool is to be used, programming can be made with the T-function. The desired tool must be called before programming a path or length compensation with G11 T...... The T-function automatically activates the tool length compensation, which can be switched off with T00.

2.5.2 Tool table

The program number P9900 is reserved for the tool table. Up to 99 tools (T001 - T099) can be stored with radius and tool offset. These data will be called up by the T-word and are used by the path and the length compensation algorithms.

3   TURNING MODE

3.1   GENERAL INFORMATION

With the lathe 2 axis are used, the X and the Z-axis. Internally in the CNC these are however the axes X Y, whereby only on the display Y is exchanged with Z. That is why the electrical connection of the Z-axis is effected to the Y-connectors on the back of the CNC. Also in the machine data Y is exchanged with Z. The Z-axis runs in parallel to the spindle axis and the positive direction is from the workpiece to the tool. The X-axis is right-angled to the Z-axis and the positive direction goes to larger diameters.

The X-axis is programmed in diameter values when G90 is active and in radius values when G91 is active.

3.1.1   Reference points

The machine Zero Point is fixed by the centre of the spindle. Its coordinates are X = 0, Z = 0. Each axis of the machine has a firm point of reference, which is dependent on the machine construction. To this reference point, all movements of the axes refer. With G74 the reference point of the selected axis is searched automatically. Subsequently, the Actual Value on the display should correspond to the Actual Position. This is done by programming G92 X+(XMAX) Z0.

The position of the Zero Point of the program depends on the workpiece.

3.1.2   Graphic Mode

Always the programmed path is displayed. By pushing START, the corrected path is displayed in a second program execution run.

Between 2 blocs, a dashed line towards the center line is added so that a picture of the workpiece is generated.

The following example can be typed in:

N001   G11 F1000 S1000
N002   G74 X..200,000 Z...10,000
N003   G00 X...50,000 Z....0,000
N004   G01   Z-..20,000
N005   G03 X..150,000 Z-..70,000   I....0,000   J-..50,000
N006   G00 X..200,000 Z...10.000

3.1.3   The G-Functions for Turning

G00   RAPID TRAVERSE
G01   LINEAR INTERPOLATION
G02   CIRCLE CW
G03   CIRCLE CCW
G04   DWELL TIME
G05   CIRCLE WITH RADIUS CW
G06   CIRCLE WITH RADIUS CCW
G07   CIRCLE WITH ANGLE
G08   ASYNCHRONOUS MOVEMENT
G09   SKIP REST OF TRAVEL
G10   CORNER ROUNDING (OPTION)
G11   ADDITIONAL FUNCTIONS F, S, T
G12   ADDITIONAL M-FUNCTIONS
G13   ADDITIONAL M-FUNCTIONS
G17   PLANE XY
G18   PLANE XZ

G19	PLANE YZ
G20	JUMP TO PROGRAM
G22	CALL PROGRAMM
G23	CALL/ JUMP WITH CONDITION
G33	TAPPING
G36	TOOL CHANGE
G40	RADIUS CORRECTION OFF
G41	RADIUS CORRECTION LEFT
G42	RADIUS CORRECTION RIGHT
G53	DISPLACEMENT OFF
G54	DISPLACEMENT ON
G55	DISPLACEMENT
G58	MEMORIZE ZERO POINT
G59	PUT T IN MEMORY
G67	SOFTWARE LIMIT SWITCH -
G68	SOFTWARE LIMIT SWITCH +
G74	REFERENCE POINT
G75	SCALE FACTOR ON
G76	SCALE FACTOR OFF
G78	FREE CYCLE
G79	FREE CYCLE
G80	CYCLE OFF
G81	FREE MODAL CYCLE
G82	DEEP DRILLING (MODAL)
G83	TAP DRILLING
G84	TAPER CUTTING
G85	TAPER CUTTING
G86	CONTOUR TURNING
G87	THREAD CYCLE
G88	RELIEF GROOVE
G90	ABSOLUTE INPUT
G91	INCREMENTAL INPUT
G92	SET Actual Value
G94	FEEDRATE IN MM/MIN
G95	FEEDRATE IN MIC./ROT
G96	CONSTANT CUTT. SPEED
G97	CONSTANT SPINDLE SPEED

3.2   G-FUNCTIONS for TURNING

The following G-functions differ from the milling version:

3.2.1   G10   CORNER ROUNDING

N... G10 X....,... Z....,... X....,... Z....,... R....,...
In G 90, the 1. XZ input field is to be programmed with the first line, in the 2. XZ the second line
and in R the radius of the corner between these lines.
G00 X01 Z0
G01 X50 Z0
G10 X100 Z-10 X100 Z-50 R10
G01 Z-60
The bloc preceding G10 must contain a value for X **and** for Z

3.2.2   G32 THREAD with tangential ARC

N... G32 X.....,... Z.....,... K.....,... J.....,...
At the end of G32, the next bloc is appended using as feedrate the last calculated speed of G32.

P32 Thread cutting with exit movement
N01   G11 S100 M3
N02   G91
N10   G33   Z-10 K1 J1
N11   G03 X5 Z-5 K5   J0
N12   G00 Z15
N13   G00 X-10

3.2.3   G33 THREAD

N... G33 X.....,... Z.....,... K.....,... J.....,...
During thread cutting, the axes X and Z are synchronized with the spindle, so that slight variations in the spindle speed are automatically eliminated.
The ending point of the thread is defined in X, Z.
K is the pitch relative to the Z axis, J is the acceleration and deceleration distance.
For the execution of G33, the CNC waits for the reference pulse of the encoder connected to the spindle, then the acceleration distance is performed, then the thread itself, then the deceleration distance.
 If J is negative, the CNC does not wait for the reference pulse of the encoder.
If I is negative, the pitch I is interpreted as beeing programmed in Inch. The real pitch in mm is calculated with the formula 25.4/I.
Examples:
P33 Thread cutting without exit movement
N01   G11 S100 M3
N02   G91
N10   G33 Z-10 K1   J1
N11   G00 X5
N12   G00 Z10
N13   G00 X-5

P10 Depth of thread is 10 * 1mm
N01 G22   N10 W9
N10 G91
N11 G33 Z -100 K.... J....
N12 G00 X 10
N13 G00 Z 100
N14 G00 X -10
N15 G00 X -1

3.2.4   G59 PUT T IN MEMORY

N... G59   X.....,... Z.....,...

Tool Measurement
First, in P9900, the tools are defined with X0, Z0, R with the corresponding radius and O0. The **longest** tool is taken as T001 and this is the reference tool with the lenght X0, Z0.

In the MANUAL INPUT Mode this tool is activated by programing G36 T001 and [START].
Then in MANUAL Mode this tool is moved down in -X direction to the diameter of the working
piece by using the Jogging Mode and the handwheel. When the diameter is touched, the Actual
Value Display of X is set to Zero by pushing the key 'C' twice.
After that, in MANUAL INPUT Mode again, with G59 in X is entered the measured diameter of the
toolpiece. With [START] this diameter is transfered as reference for the definition of the next tools.
Now, with G36 T002 and [START], the tool T002 is activated and moved down in -X until the diameter
is also touched. Because this tool is shorter, the Actual Value diplays a negative value
corresponding to the lenght difference to tool T001.
In MANUAL INPUT Mode, G59 now must be input and the cursor positioned to X, where must be
input the actual diameter of the toolpiece. With [START], the lengh difference in X is transferred to P9900
T002 X. The Actual Value displays now the actual diameter, because the tool touches the working
piece surface and the toollengh compensation is activated (by having programmed G36 T002 or
G11 T002).
It is recommended to take the longest tool as T1 and to make the move to the reference point with
T1 activated. So when switching to T2 which is smaller, this will lead to a correction movement
into the direction of the working piece and not to the direction of the LIMITSWITCH which could
be touched and would interrupt the program flow.

Hint: G59 calls P9959 in the MANUAL INPUT MODE and transmits the actual cursor position to
#79. (The cursor position for X is 0,010, Z is 0,021)
P9959 will put the actual position of the adressed axis into the corresponding adress of the activated
tool. (only for X, Z)

3.2.5   G81 FREE CYCLE

3.2.6   G82 DEEP DRILLING

N... G82 Z.....,... Q.....,... V..... H.....,...
Input data:
Z = Endposition (Absolut)
Q = Infeed
V = Security distance
H = Dwell time
This function is not modal in the Turning Mode!
Example:
P0001
N001 G11 F1000 S1000 T01
N003 G00 X0 Z1   : move to the security distance in Z
N004 G82 Z-..20,000 Q-...4,000 V....1,000 H....0,100
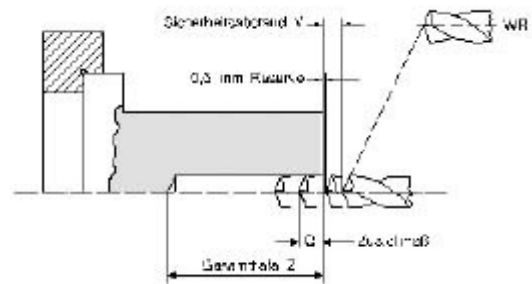
3.2.7   G83 CUT OFF

N... G83 X.....,... Z.....,... K.....,... Q.....,...
Input data:
X,Z = Starting position
K = Tool dimension in Z
Q = Infeed
Example:
P0001
N001 G00 X20 Z5
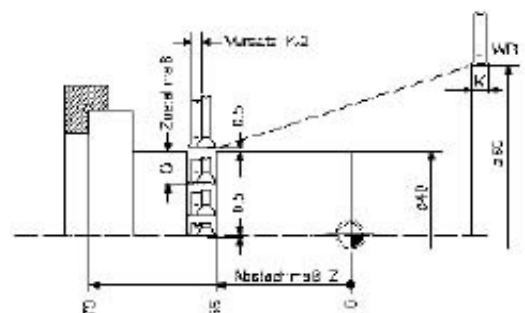
N002 G83 X...10,000 Z-..10,000 K....5,000 Q-...3,000
An example program can be executed with P9992 N83 in the Automatic Mode.

3.2.8   G84 TAPER CUTTING HORIZONTAL

N... G84 X.....,... Z.....,... E.....,... Q.....,... V.....,... K.....,...

Input data:
X = Endposition X
Z = Endposition of the large diameter
E = Endposition of the small diameter
Q = Infeed X
V = Lift off
K = Allowance
The allowance K remains at the end of the cycles.
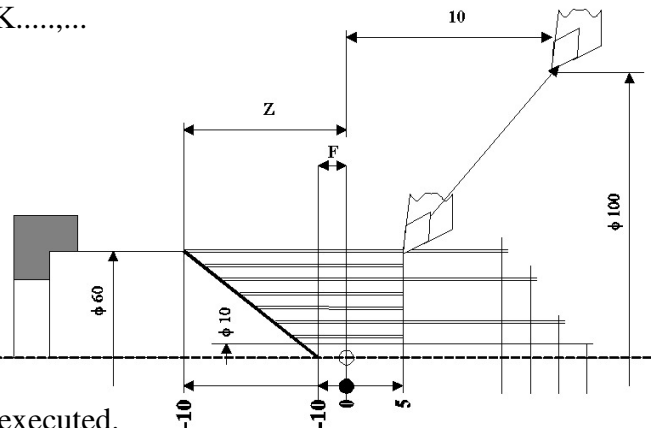
If the V value is negative, only a rough cutting is executed.
With a positive V value, also a finishing cut with half of the programmed speed is done.

Example:
P0084
N001 G11 F1000 S1000
N002 G92 X100 Z10
N003 G00 X60 Z5
N004 G84 X+..10,000 Z-..40,000 E-..10,000 Q-...4,000 V....1,000 K....1,000
An example program can be executed with P9992 N84 in the Automatic Mode.

3.2.9   G85 TAPER CUTTING VERTICAL

N... G85 X.....,... Z.....,... E.....,... Q.....,... V.....,...
I.....,...
Input data:
X = Endposition X
Z = Endposition Z at the small diameter
E = Endposition X at the large diameter
Q = Infeed X
V = Lift off
I = Allowance
The allowance I remains at the end of the cycles.
E must be greater than X.
Example:
P0085
N001 G90
N002 G11 F1000
N003 G92 X100 Z10
N004 G00 X62 Z0
N005 G85 X...20,000 Z-..40,000 E...40,000 Q-...2,000 V....1,000 I....1,000
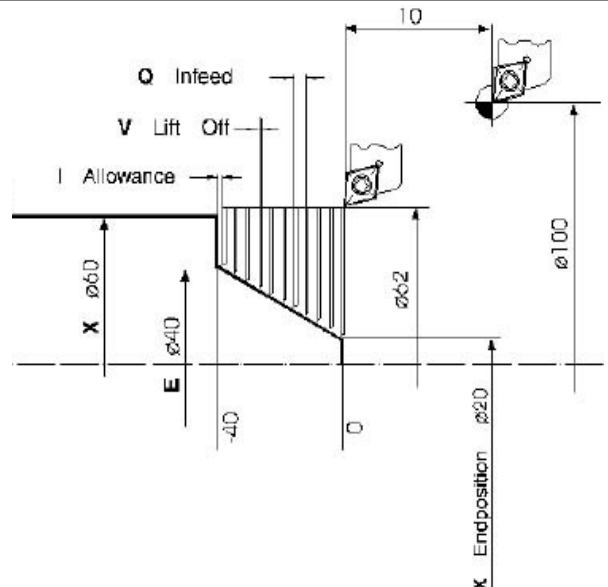
An example program can be executed with P9992 N85 in the Automatic Mode.

3.2.10   G86 CONTOUR TURNING

N... G86 X.....,... Z.....,... I.....,... K.....,... P.....,... V.....,...

Input data:

X: Allowance, (greater than the diameter of the tool)
Z: Allowance, (greater than the radius of the tool)
I, K: Infeed in X, Z
P; Programnumber (<8000) defining the contour
V = 1,000

Example:

P9900
T1 R0,4
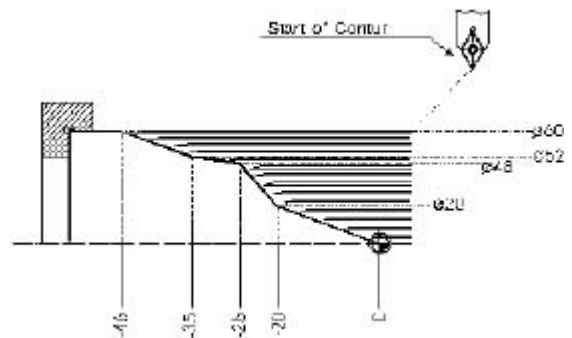
P0086
N001 G11 F1000 T1
N002 G90
N003 G92 X..100,000 Z ..10,000
N004 G00 X...60,000 Z 5,000
N005 G86 X.....1,000 Z.....0,500 I....0,000 K-...2,000 V1,000
;X greater than diameter of T1, Z geater than radius of T1
P186
N001 G90
N002 G42
N010 G00 X....0,000 Z 0,000   ;Contur definition starts with N10, both X and Z
N011 G01 X..20,000 Z-.. 20,000   ;must be programmed.
N012 G01 X .48,000 Z-...26,000
N013 G01 X..52,000 Z-...35,000
N014 G01 X..60,000 Z-...46,000
N015 G40   ;Endbloc must contain G40
N016 G00 Z 0,000

- Contur definition must start with bloc number N10 !
- Allowed are G01, G02, G03!
- The maximum diameter of the contour must be smaller or equal to the starting diameter at the beginning of the cycle.
- An example program can be executed with P9992 N86 in the Automatic Mode.

3.2.11   G87 THREAD CYCLE

N... G87 Z.....,... K.....,... I.....,... Q.....,... E.....,... J.....,... U.....,... V.....,...

Input data:

Z = Endpoint (Absolute coordinates)
K = Pitch
I = Depth of thread (Relative coordinates)
Q = Infeed X
E = Angle
J = Acceleration and deceleration distance
U = Endpoint in X of a tangential arc at the end of Thread.
This value must be greater than I absolutely seen (|U|>|I|)
V = Endpoint in Z of a tangential arc at the end of Thread
U and V can be programmed with 0 so no arc will be generated
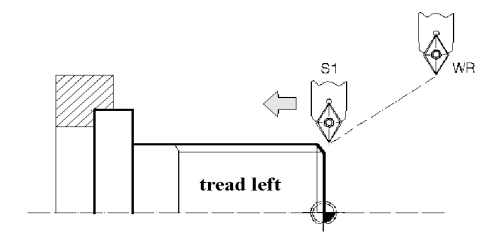
Before trying G87, it is recommended to try the test programm

mentioned with the Function G33 (page 45)!!

Example:
N001 G11 F1000 S500 M03
N002 G00 X50   Z1
N003 G87 Z+ ...0,000 K....,... I.....,... Q....,... E......,... J....,... U.....,... V.....,...
N004 G87 Z-..50,000 K....1,000 I-...1,000 Q-...0,300 E...60,000 J....1,000 U5,000 V-5,000
This example program can be executed with P9992 N87 in the Automatic Mode.

Conical thread can be programmed as follow:
N10 G87 Z+.1,000 K....,... I.....,... Q....,... E......,... J....,... U.....,... V.....,...
N11 G87 Z-50,000 K1,000 I-1,000 Q-0,300 E+60,000 J1,000 U5,000. V-5,000
In the first G87 bloc, an additional X-movement in relative coordinates is programmable, in the example of 1mm.

3.2.12   G88 RELIEF GROOVE

N... G88 X....,... Z....,... R....,... W....,...
Input data:
X: Depth of relief
Z: Lenght of relief
W: Angle
R: Radius
Example:
N001 G00 X..40,000 Z......0,000
N010 G88 X-...5,000 Z-..20,000 R....5,000 W...45,000

An example program can be executed with P9992 N88 in the Automatic Mode.

3.2.13   G95 FEEDRATE IN MICROMETER / ROTATION

N... G95 F...... S...... T.... M.... B....
The calculation of the feedrate in mm/min is done by the following formula;
$F(mm/min) = F(µ/Rot) \times S(Rot/min) / 1000$
F can be calculated from the measured or programmed spindle speed S, depending on machine datum N905X page 73.
If the calculated feedrate becomes higher as FMAX in P0, then it is reduced to this feedrate without generating an error message.
G94 and G95 define how the CNC should interpret the F-word:
• In G94 F is a feedrate in mm/min
• In G95 F is a feedrate in µm/rot.
A movement programed in G0 will always be executed with the maximal feedrate defined in P0 independently of G95.
During G95 the feedrate potentiometer is not activated.

3.2.14   G96 CONSTANT CUTTING SPEED

N... G96 V...... S...... T....
With G96, the cutting speed is hold constant. The CNC calculates the appropriate spindle speed for each diameter. The cutting speed in m/min is programed under the adress V.
With each change in diameter the spindle speed is calculated and updated. The calculated speed could be to high , so a maximum spindle speed is input at the adress S.When G95 is programed in addition to G96, also the feedrate F is updated synchronously to the variation of the spindle speed.

$$S \text{ (Rot/min)} = \frac{V \text{ (m/min) x 1000}}{X \text{ (mm) x 3,14}}$$

where X is the actual diameter as it is displayed in the Actual Value Display of X.

P0001 Test G96
N001 G94
N002 G11 F..500
N003 G96 V..100 S..2000
N004 G90
N005 G92 X..50,000
N006 G01 X...0,000
N007 G01 X..50,000

3.2.15   G97 CONSTANT SPINDLE SPEED (Initial state)

N... G97   F.... S.... T... M... B...

The spindle speed is programed in rotations per min, which is the initial state when switching the CNC on.

3.3   Special Programms

3.3.1   P9990 CODE Check

3.3.2   P9991 Calculator

3.3.3   P9992 Testprograms for Cycles

The following programs starting with the indicated bloc can be executed in the GRAFIC Mode.

P9992 N83 demonstrates the example program of G83
P9992 N84 demonstrates the example program of G84
P9992 N85 demonstrates the example program of G85
P9992 N86 demonstrates the example program of G86. (A Tool N1 must be defined in P9900)
P9992 N87 demonstrates the example program of G87

4   PARAMETRICAL FUNCTIONS

The parametrical functions are an essential extension of the possibilities of a CNC. The user can develop cycles or make calculations within his program.
The CNC calculates internally with integer values, the number X + 1,000 is internally 1000, the number F100 ist internally 100. If the number of digits after the decimal point is 2, then X+1,00 internally is 100!
Hint:
Parametrical functions are executed on the fly during a movement. If the movement should be finished before the following parametrical functions are executed, a bloc G13 M19 must be programmed after the movement.

4.1   HOW TO USE PARAMETRICAL FUNCTIONS

4.1.1   Linear interpolation with parameters

Select linear interpolation (G01) and push ➡. The input buffer for X is active now.Push ⬇ and enter a 3-digit number representing a parameter register.
N001 G01 X......#004 Y......,... Z...10,000
The actual contents of parameter register #004 will be taken as endpoint for X and the value 10,000 will be used for Z.All addresses can be programmed in this way.

4.1.2   Calculation with parameters

100 parameter registers (000-099) are available to the user (200 starting from Version 4.02). They can be manipulated by mathematical functions. To select these functions (f.e. Addition), push ⬇ while the G-address is active.
The input line now looks as follows:
N002 ⬇..
Now the code for addition (01) can be typed in. After using ➡ the following display will appear:
N002 ⬇01 #... = #... + @.....,...

Now one can define f.e.:
N003 ⬇01 #001 = #002 + @....3,000
The new value of #001 is the result from the addition of the value in #002 and the value 3,000.
Also a register can be selected in the input field @ by pushing ⬇:
N002 ⬇01 #001 = #002 + @....#003
This means that the new value in parameter register #001 is the result from the addition of the values of register #002 plus #003.

4.1.3   Indirect programming

Also indirect programming is possible:
N004 ⬇01 #001 = #002 + @....#210
The new value is calculated from the contents of #002 and the contents of the register whose adress is defined in #010. #200 to #255 allow indirect programming with registers #000 to #055.
Or ⬇94 #210 means that the text, whose number is in #010, will be displayed.

4.1.4   Reserved parameters

The parameter registers #040-#099 can be changed by the cycles. If no cycles are used, they are available to the user.
A cycle G36,G84 - G89 will load #080 to #089 with the programmed values.

#090 will be loaded with the byte defining which axes (or Input Fields) have been programmed in the bloc.
#091 and #092 will be loaded with the programnumber and bloc number of the calling program.
The cycles G81 - G82 will load into #070 to #079.

#098 will be an OR-ed image of the inputs of I/O1 to I/O4. If #098 is reset to 0, a program can detect what inputs have been activated in the meantime.
#099 is used by the function ⬇55
#100 will be decremented in the background all 10 ms to zero.
#101 corresponds to the Execution Time and is resetted at the beginning of each program started.
Registers # 102, # 103, # 104 contain the position at what the CNC was switched off.
#105 Display of the programmed S-Value (see P0 N904 V2048)
#127 and #199 are charged with the value 0 after RESET

4.1.5	Jump to a Bloc Defined by Label

The following test program explains the function:
P0055
N001 ⬇53 JUMP TO ₦LBL

...
N046⬇80 LBL

...
The JUMP function looks in P0055 for the occurrence of the word ‚LBL', detects it in bloc N046 and program execution resumes at that point.This function can be used with all parametrical jumps.
How to input ‚LBL' in the N-field? When N is highlighted and the input field is empty then push the key which opens an input field for characters. Input LBL followed by .
The label can be between 2 and 4 characters long!

4.2	Mathematical Parametrical Functions

⬇00	#...	=		@.....,...	assign value
⬇01	#...	= #... +		@.....,...	*addition
⬇02	#...	= #... -		@.....,...	*substraction
⬇03	#...	= #...		@.....,...	*multiplication
⬇04	#...	= #... /		@.....,...	*division
⬇10	#...	= COPY	#...		copy contents
⬇11	#001	= ATN	#002		arcustangens of (#002)/(#003)
⬇12	#001	= PYTH	#002		#001 = SQRT ((#002)² + (#003)²)
⬇13	#...	= CPL	#...		calculate complement
⬇14	#...	= ABS	#...		calculate absolute value
⬇15	#...	= SQRT	#...		calculate the root
⬇16	#...	= SIN	#...		calculate sine (gives sine x1000)
⬇17	#...	= COS	#...		calculate cosine (value x1000)
⬇18	#...	= AND	#...		*logical AND function
⬇19	#...	= DIV2	#...		division by 2
⬇20	#...	= OR	#...		*logical OR function
⬇21	#...	= BIT	#...		*Test BIT.

Example: ⬇21   #001= BIT#128:
#001 is tested if the bit with the value 128 is present. If yes, the NOTZERO Flag is set.
Also a combination of bits can be tested. With the value 7, bits 0, 1, 2 can be checked.

⬇22   #...   = COMP   #...   *Compare.
Example: ⬇22   #001= COMP#005:
#001 is tested if the value 5 is present. If yes, the NOTZERO Flag is set.
  ⬇22   #001= COMP#000 tests if #001 is zero. Maximum value for comparing is 255!

↓50 (JUMP ZER TO) N...   jump if result zero
↓51 (JUMP POS TO) N...   jump if result positive
↓52 (JUMP NEG TO) N...   jump if result negative
↓53 (JUMP TO)   N...   jump without condition
↓54 (JUMP NZ TO) N...   jump if result not zero
↓55 (JUMP DEC TO) N...   decrement register #099 and jump if not equal.
Functions marked with * will influence the result register used for jumps with condition.

4.3   Parametrical Special Functions

The programmable values can be between 0 and 255!

4.3.1   ↓80   Text Input

For a space, press „±". Shift and then ☐ deletes the preceeding character. The last character of the line always should be a letter, a number or a space. If the last character is a „=", an input field will be opened in the Automatic or Graphic Mode. A value can be input which will be transfered into a parametrical register by pressing ☐ or ☐ or „START". The number of this register is the same as the bloc number in which the function ↓80 was programmed.

4.3.2   ↓81   Display Text From P8000

↓81 #*A*

Display or print the the text stored in program P8000 with the bloc number *A*. In addition the content of register #000 defines, how and where the text will be output:

- #000 = 0,000:
  Text output to display.
- #000 = 0,001 - 16,383:
  Content of register #000 defines the screen position of the text. The value 0 corresponds to the upper left position in the 1. row. The value 0,255 (or 0,240 with LCD displays) corresponds to the lowest left position in the 1. row.
- #000 = 16,384 - 32,767:
  As before, however an eventual request to input a value will be ignored
- #000 = 150,000
  Output to Serial Interface
- #000 =201 - 255:
  The value in one of the registers #001 - #055 defines the text number to be displayed.

↓81 #... #*A* #*B*

If „=" is present at the end of the text, an input field of the length *A* with *B* digits after the decimal point is opened. *A* can be 1 to 9, *B* can be 0 to 5, however *A* must be at least *B*+2! If a sign ± has to be displayed, *B* must be 16 to 21.
If a sign must be displayed, at least 1 digit after the decimal point must be specified!

↓83 #...

Like ↓81, but the textes will not be selected from P8000 but from the actual program in progress.

Example:
P5
N1 G13 M21 M28   ;Stop bloc scrolling and actual value display
N2 [91 #0   ;Clear Display

N10 [0 #0 = @0,010   ;Select line 1 for display
N11 [80 INPUT LENGTH   X=   ;Define text for display. N11 defines #11 for storage
                                                         ;of input value
N20 [83 #11   ;Put text N011 on the screen and wait for an input
                                                          ;which will be memorized in register #011.

4.3.3   ↓82   Subprogram call of the CNC operating system

↓82 #000   Subprogram call of the CNC operating system. #040 contains the address, #041,42,43,44 will be stored to HL,DE,BC,A.
- Change Operation Mode:
  #40 = @0,027, #44 = Keycode according to 89

-   - External Data Router:

  #40 = @0,035, #41 = Program Number (-0,001 for all programs)

  #44 = Number corresponding to the Menu of External Data.
- ↓82 #001

As #000, however when returning from the subprogramcall, HL, DE, BC, A will not be updated.

4.3.4   ↓82   Text Engraving

- ↓82 #003 The following bloc will be opened as text bloc.
- ↓82 #004. The next character will be fetched and put in #00. If the last character has been read, the ZERO flag will be set.

4.3.5   ↓82   Input/Output Via Serial Interface

- ↓82 #005 #A. Input Via Serial Interface. Read a number until CR from the Serial Interface and put the number into #A

  For example sending the string +0012,345 puts the number 12345 into #A. M2442 must be set!
- ↓82 #006 #A   Read a single character from the Serial Interface and put it into #A. If no character has been received, the Zero flag is set. M2442 must be set!
- ↓82 #007 #A #B #C Output Via Serial Interface. A defines the register, B defines the number of consecutiv registers to be sent.The final character sent as a CR. If C is 1, the CNC waits for the same characters to be sent back. They are checked if they are correct. If not or if a time out occurred, the ZERO flag is set. This function is in preparation.
- ↓82 #008 #A #B #C   Reserved function

4.3.6   ↓82   Function Calls of Operating system

- ↓82 #10 #A   Change Operation Mode:

  #A = Keycode according to ↓89.
- ↓82 #11 #A   A=0   Display a button, A=1 Invert a button, A=2, Display Start/Stop Buttons

4.3.7   ↓84 Read / Write

↓84 #A #B #C #D #E   Read/Write
- A: 0 = Memory access, 16 = I/O access, 64 = Interpolator access, 128 = DILAG access
- B: 1 = Read, 2 = Write, 5 = Read 4 Bytes, 6 = Write 4 Bytes to DILAG/Interpolator.

  8 = Reset Dilag
- C: parametrical register, where to read or from where to write.
- D: parametrical register with the memeory adress, from where to read or where to write.

  If A = 16, then D is directly the I/O adress for read/write.

  (The I/O cards 1-8 have the adresses 64 - 71)

  If A = 64, then D is directly the command number for the Interpolator.

  If A = 128, then D is directly the DILAG register for read/write.
- E: number of values to read or write.

  Example 1:

  Reading (#5) the Lag Error (#72) of the encoders X,Y,Z on the Dilag (#128) into #10:

  ↓84 #128 #5 #10 #72 #3

  ↓92 #10 #20   ;Display #10 in line 20

  Example 2:

  Reading (#5) the Actual Spindle Speed (#180) from the Interpolator (#64) into #10:

  ↓84 #64 #5 #10 #180 #1

  ↓92 #10 #20   ;Display #10 in line 20

4.3.8   ↓86   Axes Monitoring System On.

↓86 #A #B #C#D#E   Monitoring System On.

- **A**: 72-75 Lag Distance X-U
- **B**: 76-79 Actual Value X-U as coming from Interpolator
- **C**: 128 / 136 DILAG card bc,
- **D,E** Dividing Values for **A,B**

To display the Actual Value of X (#72) and the corresponding lag error (#76), the following program must be input:

N1 G13 M2241
N2 ↓86 #72 #76 #128 #1 #1
N3 G91
N10 G0 X10
N11 G0 X-10
N12 ↓84 #128 #001 #090 #032 #004   ;Read FeedForwardRegisters into #90
N13 ↓92 #090 #010   ;Display register 90 in line 10
N14 G20 N10

This function uses and alters registers #40 to #49.

During display, the keys 1 – 9 influence the amplitude of the curves displayed. (In fact they act on the Dividing Values)

X+ increments and X- decrements the FeedForwardRegister of the X-axis so that it can be changed online. The same for Y+ and Y-, Z+ and Z-.

4.3.9   ↓87 #A,#B   Get a bloc from memory.

- **B**=0: Program the desired program number in register **#A**, the bloc number in **#A**+1. The content of the bloc is put into **#A**+2 and the following registers.
- **B**=1: No error message is generated if P or N has not been found. The ZERO flag will be set.
- **B**=2: The G-function, the Bit Field describing the programmed axes in the bloc and the number of the following bloc are put into #90, #91, #92

4.3.10   ↓88   Put a bloc into memory.

as ↓87, but #**A**+2 and the following registers are put into memory.

4.3.11   ↓89   Keyboard scan.

↓89 #**A** #**B**   keyboard scan.

- **B**=0: Execute a keyboard scan. The code of the depressed key is put into register **A**. If no key is pressed, the ZERO flag will be set.
- **B**=1: The code of the last depressed key is put into register **A**. (without keyboardscan)
- **B**=2: The code of the just depressed key is put into register **A**. (without keyboardscan)
- **B**=3: Execute a keyboard scan and return only if a key is pressed which is put into register **A**.
- **B**=4: Like **B**=3: However before the keyboardscan, the CNC waits until no more key is pressed.

Code	Function	Code	Function
9	0 - 9	25	MANUAL MODE
10		26	GRAFIC MODE
11	+/-	27	AUTOMATIC MODE
12	MENU	28	MANUAL INPUT
13	-->	29	TEACH IN
14	CLEAR	30	REFERENCE
15	INPUT MODE	31	CLEAR MEMORY
16	+X	32	SEARCH BLOC
17	-X	33	CLEAR BLOC
18	+Y	34	EXTERNAL DATA
19	-Y	35	
20	+Z	36	SPINDLE ON
21	-Z	37	COOLING ON
22	SINGLE STEP	38	
23	START		
24	STOP		

4.3.12   ↓90 Insert line.

↓90 #000 insert blank line.

4.3.13   ↓91 Clear screen.

↓91 *#A #B #C #D #E* Clear screen.
- *A*=0: Clear the screen,
- *A*=255: Set all the display points
- *A*=1: Clear part of the screen,
- *A*=2: Invert part of the screen

The starting point B,C and the length/height in D,E. (values between 0 and 255)

4.3.14   ↓92 Display registers

- ↓92 *#A #B* Display registers *#A* to *#A*+4. (optionnally *B* can be programmed with 0)
- With *B*=1, the output will go to the printer.
- With *B*=2, the output will go to the serial interface.
- With *B*=3-254, B is the line where the registers will be displayed.
- *B*=255, the registers are displayed continously together with the Actual Value.

4.3.15   ↓93 Water Jet Cutting functions

- ↓93 #0 *#A #B* Arcs with a radius smaller than *A* (0-199mm) will be executed with a reduced speed of *B*% (0-100%)of the programmed feedrate. Switch off this function with ↓93 #0 #0 #0
- ↓93 #1 *#A #B* If between 2 consecutive blocs exists an angle of more or equal to *B,* the axes are stopped and a dwell time of **A** tenth of seconds is executed.**A** can be in the range between 0 and 199. The function can be switched off with **A**=0.
- ↓93 #2 *#A #B* Acceleration and deceleration ramp with lenght *A* in mm. The function can be switched off with =0 and is not applied with G00.

4.3.16   ↓93 Special Functions TNC135 Replacement

- ↓93 #3 *#A #B* Wait until a change in the Actual Value in X,Y,Z greater than the Value in B is detected and store the changes of X,Y,Z in #A, or exit with the key STOP.

- ↓93 #4 *#A #B* Loop until axis defined in *A* has reached the position defined in parametrical register #30 (exit with ZERO flag), or the bloc has reached the end (exit with NZ flag). The last 128 increments are executed with a ramp where *B* defines the lowest speed of the ramp.
- ↓93 #38 *#A* Count up the ONLINE Timer using Interpolator 2ms (A=1) or Dilag 2ms (A=2)

4.3.17   ↓93 Limit Switches

- ↓93 #39 *#A B* Switch off/on limitswitches programmed in **A. B** =0:OFF, =1:ON
- ↓93 #40 *#A* **Supervision of Limit Switches on while the axes are not moving.** *A*=1 for X, 2 for Y, 4 for Z and so on. If the Limit Switch of one of the selected axes is detected, the program execution is interrupted and the message LIMIT SWITCH AT REST is displayed

4.3.18   ↓93 Continous Movement

- ↓93 #41 Movement without acceleration or deceleration. Gives continous movement also with non- tangent blocs. It is the responsability of the user to switch on and off this function correctly. This function corresponds to M41, but is executed on the fly
- ↓93 #51 Cancels M41
- ↓93 #60
- ↓93 #61 Synchronize NCount
- ↓93 #51 Clear Memory Expansion of 4MB

4.3.19   ↓94 Display Internal Text

↓94 #A Display internal text number *A*.
In addition the content of register #000 defines, how and where the text will be output:
- #000 = 0,000:
  Text output to display.
- #000 = 0,001 - 16,383:
  Content of register #000 defines the screen position of the text. The value 0 corresponds to the upper left position in the 1. row. The value 0,255 (or 0,240 with LCD displays) corresponds to the lowest left position in the 1. row.
- #000 = 150,000
  Output to Serial Interface
- #000 =201 - 255:
  The value in one of the registers #001 - #055 defines the text number to be displayed.

4.3.20   ↓95 Display Error Message

↓95 #A Display error message *A* and stop Automatic or Graphic mode, f.e. *A*=58 displays the message „check bloc". If *A*=200, no message will be displayed, program execution will simply be stopped.

4.3.21   ↓96 Modal Functions

↓96 #0/1 Save/Restore state of G90/91, G94/95 and M21-M28
↓96 #2/3 as #001/002 however used in G81 - G83.
↓96 #4 *#A* Put actual T,S,F,R,M03 – M10, G90/G91 to register *#A* and following registers. M03 - M10 is a binary representation of the 8 bit ouput port with the following values:

1	Spindle running (M03 or M04)
2	Coolant activated (M08)
4	Spindle running CCW (M04)
8	Clamping (M10)

4.3.22   ↓96 Retrieve Internal Registers

↓96#5 #A #B Store the following values to register #A and following registers.

- *B*=0: Actual Value as it is on the display.
- *B*=1: Actual Value without taking into account G54, Toolcorrection...
- *B*=2: Activated offsets from G54
- *B*=3: Acivated offsets from G55
- *B*=4: Activated Tool Lenght Correction
- *B*=5: Activated G75
- *B*=6: Offset to Real Position produced by G92
- 96#6 *#A #0* Store the Actual Value of DILAG to register *#A* and following registers.
- 96#6 *#A #1* Store the Absolute Position since the last G74 to register *#A* and following registers. (Option not yet implemented)

4.3.23   ↓96 AD- and DA- Converters

↓96#7 *#A*   Store 8 analog inputs on Interpolator to register *#A* and following registers.

↓96#8 *#A #B*

- *B*=8:   Read 8 Bit AD-converter with 4 channels to register *#A* and following registers.
- *B*=12: Read 12 Bit AD-converter with 4 channels to register *#A* and following registers.

↓96#9 *#A #B*

- *B*=8:   Put values in register *#A* and following registers to 8 BIT DA converter
- *B*=12: Put values in register *#A* and following registers to 12 BIT DA converter

4.3.24   ↓96 Grafic Functions

↓96#10 *#A*

#*A*=0: GRAFIC PEN OFF, #*A*=1: GRAFIC PEN ON

#*A*=2: RADIUS LINES OFF, #*A*=3: RADIUS LINES ON

#*A*=19: Execute M19 in the Grafic Mode

4.3.25   ↓96 Limit Switches

↓96#11 *#A*

Read Hardware State of Limit switches and put it to *#A* and *#A*+1

↓96#12 *#A*

Logical Limit switches to *#A* and *#A*+1

↓96#13 *#A*

Actual Spindle Speed to *#A*

4.3.26   ↓96 Movement Control

↓96#14 *#A B*

A=0: Stops the actual movement of an axis

A=1: The next G33 bloc is started and program execution continues immediatly.

A=2: Display Actual Value.

A=3: Start the axes (MOTAN)

A=4: Set Z Flag if Asynchronous Axis *B* is at rest. B=1, 2, 4, 8, 16 ..

A=5: Display Buttons M03,M08

A=6: Display Buttons START/STOP with STOP activated

A=7: Display Buttons START/STOP with START activated

4.3.27   ↓96 SSI Interface

↓96#15 *#A #B #C #D*

- Read Absolute Position Encoder to *#A*.
- *B*=0: SSI Data Format is SINGLETURN, Binary. For This purpose Output O8 and input I8 on I/0 card 2 are used.

- **B**=1: SSI Data Format is SINGLETURN 13 bit Gray. **C and D** must not be programmed.
- **B**=16: SSI Multiturn Binary on input 8 of I/O card 2
- **B**=17: SSI Multiturn Binary on input 7 of I/O card 2
- **B**=18: SSI Multiturn Binary on input 6 of I/O card 2
- **C**= number of bits defining the rotations
- **D**= resolution in bits of encoder
- **E**= Number of I/O card used

Example: ↓96#15 #**A**#16 #14 #10

4.3.28   ↓96 TMS Monitor Call

96#16 #**A** #**B** #**C** #**D**

4.3.29   ↓96 Clock Set and Read

96#17 #**A**
A=0: Read Data into #40.....
A=1: Write Data from #40 (Year), #41 (Month), #42 (Day), #43 - #46 (must be 0)
 into clock chip

4.3.30   ↓96 Display state of IO1 and IO2

96#18 #**A, A**=0

4.3.31   ↓96 Change Display Color

96#19 #**A** #**B** #**C** #**D** #**E** (default is 6, 5, 3, 7, 1)
#A is the color for the text, #B for the Icons, #C for the border of the icons, #D for the contour in
Grafic Mode, #E for the toolpath in Grafic Mode.
Possible colors are: 1 Red, 2 Green, 3 Yellow, 4 Blue, 5 Magenta, 6 Cyan, 7 White

4.3.32   ↓96 Access to Moby interface

96#20 #**A**

4.3.33   ↓96 Change Baudrate during program execution

96#21 #**A**#**B** 9600 Baud, A=5, B=2 at 5Mhz, B=4 at 10Mhz
                19200 Baud, A=5, B=1 at 5Mhz, B=2 at 10Mhz

4.3.34   ↓96 Functions for Turning Mode

96#22 #0 Sets POSITIVE flag if tool is behind the Z-Axis

4.3.35   ↓98 Draw Line.

↓98 #**A** #**B** #**C** #**D** Draw Line.
Starting point in #A,#B and the ending point in #C,#D. The upper left is position 0,0 and the lower
right position is 255,255. A control with a LCD-display has a lower right position of 170, 240.

g180

g210 #A	Set #A into empty state
g220 #0 #A	Stop Timer in #A
#1 #A	Start Timer in #A, counting in ms

5 MACHINE DATA

The machine data allow an easy adaptation of the CNC to different mechanics. The machine data memory is addressed by P0000.
The data concerning one axis can be programmed for each axis separatly. An interpolation will be done with for example the lowest feedrate F of all participating axes.
Not documented bloc numbers are not allowed to be used!
The value in brackets is the default. This value is already memorized in the CNC. Only if a **different** value for an axis is needed, it must be programmed in P0000.
In the TURNING or GRINDING versions, the axes are named X,Z,C. X is the 1., Z the 2., C the 3. axis. The machine data for Z must be entered in the 2. data field (Y), the data for C in the 3. data field (Z)!!!!

5.1 Leadscrew Error Compensation

N100XR Leadscrew errorcompensation in X
N200YR Leadscrew errorcompensation in Y
N300ZR Leadscrew errorcompensation in Z
This compensation is used to correct errors in the leadscrew. The axis is moved to the negative Limit Switch with G74, the Actual Value is set to 0.
This is the beginning of a compenstion table with the Actual Value equal to 0 and the measured value also 0. Now the axis must be measured and the points where the pitch changes are noticed. After that one has f.e. the following diagram which must be memorized in P0:

```
P0
N100   X ...0,000  R ....0,000      ;Start of table
N101   X.123,000  R.120,000        ;1. measured point with error of +3mm, desired point
N102   X.182,000  R.180,000        ;2. measured point with error of +2mm, desired point
N103   X.362,000  R.360,000        ;3. measured point with error of +2mm, desired point
N104   X.459,000  R.460,000        ;4. measured point with error of -1mm, desired point
N105   X.578,000  R.580,000        ;5. measured point with error of -2mm, desired point
N106   X 900,000..R 900,000        ;Last value is at the end of he moving range
```
(The values can also be entered under X and Z for example. It is important that the first value is the measured position and the second value is the desired position!!!!)

Accordingly, the tables for Y (or Z in TURNING MODE) start at N200, the one for Z at N300. For activating the table, the Limit Switche wich was used as starting point for the table (in this case the negative one), must be moved to with G74 everytime the CNC is switched on and M210x must be executed. After that the compensation is active. The compensation always starts with M210x.
There are 32 table points allowed for each axis. This function reduces the number of blocs available for storage in the Interpolator Cache
With the following program the above table can be verified:
```
P1
N1 G74 X-1
N2 G13 M2101
N10 G92 X0
N11 G0 X120
N12 G0 X180
N13 G0 X360
N14 G0 X460
N15 G0 X580
```

The following testprogram may help understanding and verifying this function:

P0	P1	Measured position
	G74 X-1, G92 X0, G90 G13 M2101	
N100 X0 R0	N10 G0 X0	X0
N101 X1 R1	N11 G0 X1	X1
N102 X2 R2,1	N12 G0 X2	X1,9
N103 X3 R3	N13 G0 X3	X3
N104 X4 R3,9	N14 G0 X4	X4,1
N105 X5 R5	N15 G0 X5	X5

In P0 it is good to put only the following data:
N790 X771 (or 787 for Servomotors), N813 X1, and select N706 so that the motor axis makes 1 Rotation when G0 X1 is programmed.

5.2   Machine data for Movement

5.2.1   N698XYZUVABC   WAY OFF REFERENCE PULSE

Reserved for N790, value 64.

5.2.2   N699XYZUVABC   CORRECTION REFERENCE POSITION (0)

This value will be moved to after the referencepulse during G74 with the feedrate programmed in N902A. R = 0.

5.2.3   N700XYZUVABC   F MAX (1000)

Maximal feedrate in millimeter/min. The maximal interpolationfrequency depends on the CNC.
- CNC in 19" rack for stepping motors Standard: 30 Khz
- CNC in 19" rack for stepping motors with Microstepping: 120 KHz
- CNC in 19" rack for servo motors: 600 KHz
- CNC Singleboard (D22 - F33) for stepping motors: 120 Khz
- CNC Singleboard (D22 - F33) for servo motors: 300 Khz

The maximum feedrate is calculated as follows: $\boxed{\text{FMAX} = 60 \times f(Hz) / (\text{STEPS/MM})}$

So a standard CNC with 30 Khz and 100 steps per mm can move at 18 m per minute.

5.2.4   N701XYZUVABC   F START (100
5.2.5   N702XYZUVABC   F STOP (100)

Start and Stop frequency in mm/min of an interpolation. The smallest programmable value is 1.

5.2.6   N703XYZUVABC   B START (500)
5.2.7   N704XYZUVABC   B STOP (500)

Acceleration and deceleration in multiples of 10 mm/sec².
Hint:
Too small values (f.e. <10) in N 701 to N 704 together with a small value (f.e. 50) in N 706 (steps/mm) can result in a problem that the axes won´t start moving.

5.2.8   N705XYZUVABC   F OFF LIMIT SWITCH (200)

After a reference movement, this feedrate is used for moving the axis off the Limit Switch.

5.2.9   N706XYZUVABC   STEPS
5.2.10   N707XYZUVABC   MM or DEGREE(1)

(N706 has no default value, so it always must be programed!)

These 2 parameters define both together the resolution for each axis.The CNC needs the following information: How many steps (N706) gives a movement of how many units, for example millimeters (N707)? There is no limitation to these values. For example there could be 9000 steps per 7 mm, or whatever.

If the number of steps per mm is smaller than 100, it is recommended to choose cm as unit. All machinedata whith the unit mm are to be replaced with the unit cm.So you input steps/cm and F in cm/min and all valued programmed are interpreted as being in cm.

Linear Axis with Stepping Motor: (In N790 the value 16 is not programmed)

A stepping motor for the X axis makes 1000 steps per rotation and is connected to a 5 mm spindle. The resulting values are: N706 X1000 and N707 X5

For 120KHz output (F- and D-Version), the calculated value in N 706 must be divided by 4.

Rotary Table: (In N790 the value 2 is not programmed)

With a rotary table, the number of steps per rotation are to be input. A stepping motor with 800 steps per rotation drives a rotary table with a gear of 18:1. The resulting values are 800 steps x 18 = 14400 steps for 360 degrees, giving N706 X14400 and N707 X....360.

Servoaxis with Encoder:(In N790 the value 16 is programmed)

For servomotors the number of pulses of the encoder or linear measuring system are to be multiplyed by 4. A transducer on the X axis gives 250 pulses per (1) mm.The resolution of the measuring system multiplied by 4 must be input.Resulting values are N706 X1000 and N707 X1.

5.2.11   N708XYZUVABC MODULO (0)

Modulo function for the Actual Position display. Linear axes: 0, rotary axes: 360000.

5.2.12   N709XYZUVABC LIMIT SWITCH DEBOUNCE TIME (10)

During this time in ms at least the Limit Switch signal must be stable.
Maximal value is 255 ms.

5.2.13   N710XYZUVABC F REFERENCE (500)

Feedrate in mm/min for G74for each axis.

5.2.14   N711XYZUVABC WAY OFF FROM LIMIT SWITCH (1000)

In G74, this value in µm will be moved off the Limit Switch with the feedrate programmed in N705.

5.2.15   N712XYZUVABC MAXIMAL WAY OFF LIMIT SWITCH (50000)

If the Limit Switch is not desactivated in between this value in µm while moving away from the Limit Switch during G74, the CNC stops and displays an error message.

5.2.16   N713XYZUVABC MAXIMAL DISTANCE FOR STOP (0)

If this value is programmed in µm, a movement is stoped within this maximal distance when a Limit Switch is reached.

5.2.17   N714XYZUVABC BACKSLASH COMPENSATION (0)

This value in µm will be added to the following movement at each change in the moving direction.

5.2.18   N716XYZUVABC SOFTWARE LIMIT SWITCH -   (0)

5.2.19   N717XYZUVABC SOFTWARE LIMIT SWITCH + (0)

Reserved for Software Limit Switches. Use G67 and G68 instead!!

5.2.20   N722XYZUVABC F FOR BACKSLASH COMPENSATION (0)

With a value of 0, the feedrate of the compensation is the value from N701 else the value programmed here will be used.

5.2.21   N750XYZUVABC F DURING REFERENCE PULSE SEARCH (0)

N902A is not active when N70 is programmed. The speed for Reference Pulse Search can be adjusted individually for each axis.

5.2.22   N790XYZUVABC .AXISDEFINITION (771)

This value is composed for each axis separately from the sum of the following options:

1	Main Axis. A main axis will influence the feedrate during an interpolation. In most cases X,Y,Z will be main axes.
2	Linear Axis" with + and - Limit Switches which always will be active.A Rotary Axis will react to the Limit Switch only during G74.
4	A Rotary Axis will move off the Limit Switch always in negative direction.
8	Axis with spline interpolation.
16	Axis driven by a servomotor and not a by a stepping motor.
32	Search of referencepulse. With G74 and after moving to and then off the Limit Switch, the distance programmed in N711 will be moved in the same direction with the feedrate N705. Then the axis moves on with the speed F REFPULS (N902A) until the reference pulse of the transducer is detected. Here the internal counters of the CNC are zeroed.
64	Search of Referencepulse without first going to the Limitswitch. The axis moves with N710 to the reference pulse and overrides it, moves the distance in N698, and then moves thereference pulse again with the feedrate defined in N902A. This can only be done at low speed, so that the referencepulse can be recognized from the CNC.
128	Gantry axis is allowed to move also separately from main axis.
256	Limitswitch + connected
512	Limitswitch - connected
1024	Limitswitch + is normally open
2048	Limitswitch - is normally open
4096	Invert moving direction
8192	Normally if only one Limit Switch is programmed with the value 256 or 512, this Limit Switch will be accepted if the axis is comimg from the positive **OR** the negative direction. However if 8192 is programmed this feature is switched off.

SUM

The SUM for each axis can be calculated by adding the values of the desired functions.

SUM for stepping motor:	771 = (1+2+256+512),
SUM necessary for servoaxis:	787 = (1+2+16+256+512)
Servoaxis with search for reference pulse:	819 = (1+2+16+32+256+512)

5.3   MACHINE DATA FOR SERVOAXES

The machinedata N800-N813 are necessary for Servomotors. (N790 must include the value 16).

5.3.1   N800XYZUVABC P-FACTOR (20)

The Lag Distance is multiplied with the P-Factor and gives the output voltage to the servoamplifier. This output voltage **ALWAYS** is proportional to the Lag Distance. The standard value of 20 works with most applications. If the axis starts vibrating when switching on, this value must be decreased. Acordingly N804 LAGMAX should be adapted.

The maximal output voltage of +/- 10V will be reached with a Lag Distance of:

$$\frac{32000 \text{ (constant)}}{20 \text{ (P-factor)}} = 1600 \text{ increments.}$$

5.3.2   N801XYZUVABC I-FACTOR (0)
5.3.3   N802XYZUVABC D-FACTOR (0)

With the I- and D-factor, constant current regulators can be used as servoamplifiers. A tachogenerator is not needed. However the system is diffucult to adjust and should be used only for high quantity applications.

5.3.4   N803XYZUVABC IN POSITION (10)

When M27 is active, the CNC waits at the end of a movement until the Lag Distance has become smaller than the value IN POSITION. This value must be reached within 2 seconds, else an error message will be displayed.

5.3.5   N804XYZUVABC LAGMAX (1600)

If the Lag Distance in increments becomes graeter than this value, the CNC stops and displays an error message. The maximal programmed value should be 32000/P-Factor. The Actual Position is not lost.

5.3.6   N805XYZUVABC DIVIDER LAG DISTANCE (0)

Before the processing of the Lag Distance, this one can be divided by this value in order to reduce the sensitivity of the Closed Loop. However the positioning accuracy will be reduced by that same value.

The programmable values are 1 to 7, corresponding to dividers of 2, 4, 8, 16, 32, 64, 128.

5.3.7   N806XYZUVABC F FEEDFORWARD. (0)

5.3.8   N807XYZUVABC ISTFAKT..(1)

Pulses coming from the encoder are multiplied with this value.

5.3.9   N808XYZUVABC SOLLFAKT..(1)

Pulses coming from the interpolator are multiplied with this value.

5.3.10   N809Z DISTANCE GANTRY AXES (0)

The function M80xx allows to connect 2 axes, while one is master and the other is slave. If the distance in increments between both becomes greater than this value, the axes will stop and the error GANTRY AXES will be displayed. The Actual Position is not lost. A value of 0 will switch off this supervision function.

5.3.11   N809U FATAL ERROR GANTRY AXES (0)

If the distande of the Gantry Axes exceeds this value in increments, all the axes are switched off and an error message is displayed.

5.3.12   N812XYZUVABC   ZERO OFFSET (0)

When an axis stands still and the Lag Distance cannot be corrected to 0 with the trimmer "Offset" of the servoamplifier, then an offset can be defined here.
For example a value of 35 will give +15mV, a value of 65550 gives –15mV.

5.3.13   N813X   SERVO ON (0)

Activates the SERVO ON output of connetor X11.

1	2	4	8	16	32	64	128
X	Y	Z	U	V	A	B	C

Example: Milling Machine with axes X, Y, Z = 7
                Turning machine with axes X, Z   = 3

With X,Y,Z,U,V,A,B,C each axis can be activated by programming a 1 and disabled by programming 0.

5.3.14   N813Z   FATAL LAGERROR (32000)

If the Lag Distance of one axis exceeds this value in increments, the servoamplifiers are disabled. This value should always be greater (min. 30%) than N804. This value is common to all axes. The maximum input value is 32000!

5.4   MISCELLANEOUS MACHINE DATA

5.4.1   N900X   CODE (0)

When going to INPUT MODE, EXTERNAL DATA, TEACH IN or CLEAR MEMORY, the CNC requests an usercode, which can be determined here. A value of 0 disables this request.

5.4.2   N900Y   CODE (0)

User code necessary to access P0000 (Machinedata), all programs higher or equal to P9800. A value of 0 disables this request. (With M904V64 some G-functions also can be disabled.)

5.4.3   N900A   SPINDLEAXIS (0)

0   None of the axes X - C is a spindleaxis.

1, 2, 3, 4..   Axis X or Y.. or C is a spindleaxis.
            M03 or M04 activates the spindleaxis, the speed of the spindle can be programmed with G11 S..... .
            The spindleaxis must be defined in P0 N790 with the value 16 as a servoaxis and must be activated in P0 N813. If it is not activated in N813, then it acts like a normal spindle without closed loop position control.
            M03/M04 takes the spindle out of the Closed Loop position control and makes it run with the programmed speed S. M05 takes the spindle again under Closed Loop position control allowing the axis to be positioned with G00. In order to reference the spindleaxis with G74, in P0 N790 the option 64 should be activated which enables referencing without additional Limit Switch using only the reference pulse of the encoder. After an M05 command the spindle must be referenced! The actual spindlespeed can not be displayed. In order to do that, the pulses must also be connected to G33 interface!

Add 16 The output voltage always is positive

5.4.4 N900B MOTOR ON/OFF (0)

At the beginning of a movement of an axis in the MANUAL MODE, the signal MOTOR ON will be activated. The programmed value defines a time in ms which will pass before the axis starts to move.

5.4.5 N900C ACCELERATION TIME of the SPINDLE (0)

5.4.6 N901X S MAX (3000)

Maximal spindle speed in r.p.m. The S-output generates a voltage between 0 and 10V, according to S0000 to S3000. The programmed value must be a rounded up multiple of 250. The maximal programmable value is 60000.

5.4.7 N901Y BAUDRATE (9600)

Defines the baudrate of the Serial Interface. Values of 1200, 2400, 4800, 9600, 19200 are selectable.

5.4.8 N901Z DIGITS AFTER DECIMAL POINT (3)

Number of digits between 1 and 5. However one axis can be fixed to 3 digits by adding 256 for X, 512 for Y, 1024 for Z, 2048 for U.

5.4.9 N901U DIVIDER F PROP (0)

The actual feed rate is divided by this value (which usually is the up rounded value FMAX/256) and output to the optional „F PROP" card (0-255 = 0-10V). Also the output A1 on the I/O card 9 will be set when the actual speed corresponds to the programmed speed. A value of 0 inhibits this function.

5.4.10 N901V I/O-CALL CARD-NR. (0)

Number of the I/O card which is allowed to call subprograms at the end of a bloc in the Automatic Mode. (1= IO card 1, 2= IO card 2 ....)
If the value 16 is programmed, the decimal value of the last pressed key is used for calling the subprogram. For example if N901C is 100 and the key 6 which must be defined with N901A6, is pressed during program execution, then Program P106 will be called at the end of the next movement. This applies only if M2342 (M32) is not active! See also page 80

5.4.11 N901A I/O-CALL ACTIVE INPUTS F (0)

Bit pattern to indicate what input triggers an I/O subprogram call. (E1=1, E2=2, E3=4, E4=8 ....)

5.4.12 N901B I/O-CALL POLARITY OF INPUT (0)

Selects if input for I/O subprogram call is active high or active low.(E1 is active high = 1, E2 is active high = 2, E3 is active high = 4, E4 is active high = 8, ...)

5.4.13 N901C I/O-CALL PROGRAMMNUMBER (0)

Program number which will be called from input 1. The following inputs call accordingly the next higher programmnumber. (f.e. 10, then P11 with E1, P12 with E2, P14 with E3, P18 with E4 .... will be called)

5.4.14 N902X CONTROLCODES I (0)

The date N902X is composed from the following numbers:

1	Ignore error message after Reset or Switching on the CNC
2	Disable keybord simulation via Serial Interface.
4	Positioning (Refer also to connector X8, I/O4)
8	Set Single Step in the AUTOMATIC MODE.
16	In the INPUT MODE, G79 automatically generates G01, G01, G03 blocs and puts them into memory. The program number which is calling must be greater than P9979!
32	Do not display on the screen the last key pressed.
64	XON-XOFF protocol in EXTERNAL DATA for Serial Interface.
128	STOP input (E2) on I/O 4 or input defined according to P0 N905C is active, when no signal is applied.
256	No cycles, G81 to G99 can be freely used for own programs.
512	EXT. INTERRUPT input (E1) on I/O 4 is active, when no signal is applied.
1024	Reserved function for Positioning Loop Control on Interpolator card.
2048	In JOG MODE during MANUAL MODE an axis is started without delay and without generating a movement of 1 increment at the beginning.

The control codes 1 - 128 can also be set with M-functions. M2441 corresponds to 01, M2442 to 02, M2443 to 04, M2444 to 08 and so on.

5.4.15 N902Y CONTROLCODES II (0)

1	In the positioning mode at the end of a bloc, the signal MOTOR ON remains active.
2	In G90 and with Rotary Axis, only the programmed sign defines the direction of the movement of the X-axis
4	In the Manual Mode, only Jogmode is activ.
8	An active G54 or tool will not influence the Actual Value counter.
16	No delay when changing from M03 to M04 or to M05.
32	In Positioning mode only a START pulse is necessary.
64	Generate „PARITY EVEN" for serial output in EXTERNAL DATA.
128	Stop and Single Step in the Automatic Mode will reset outputs A1, A2 on I/O Card 4.
256	The function [83 takes the textes from P8001.
512	Tool is behind Z-axis.
1024	Initial state G90 in Automatic Mode
2048	Do not reset G54 when changing to Menu.
4096	Do not reset G75 when changing to Menu.
8192	M03, M04, M08 in the Manual Mode will call P9936.

The control codes 1 - 128 can also be set with M-functions. M2541 corresponds to 01, M2542 to 02, M2543 to 04, M2544 to 08 and so on.

5.4.16 N902Z LANGUAGE (0)

0: German 1: English 2: French 3: Turkish 4: Italian 5: Spanish....6: Potoguese 7: Finnish
8: Polnish
If the value 16 is added, chinese textes are generated where available.
If the value 32 is added, persian textes are generated where available.

5.4.17 N902U INITIAL VALUE M23xx (0)

M2341 corresponds to the value 1, M2342 ¸ 2, M2343 ¸ 4, M2344 ¸ 8. With the value 256, no initialisation will be done.

1	Spline on (Option)
2	Interpolatorcache for approx. 200 blocs on
4	Feedrate potentiometer off with G00
8	When pressing „MENU" P9999 is invoked
16	
32	
64	The „wait for input" can be skipped by pressing „START" or "MENU"
128	The Actual Value of the U-axis will be added to the Z- axis

5.4.18   N902V INITIAL VALUE   M22xx (0)

M2241 corresponds to the value 1, M2242 ¸ 2, M2243 ¸ 4, M2244 ¸ 8. With the value 256, no initialisation will be done.

1	Bloc updating off in the automatic mode.
2	Bloc updating off in program calls
4	Feedrate potentiometer off
8	Test run without G04 and M-functions
16	Test run with Rapid Traverse
32	Keyboard off
64	Wait for IN POSITION (see also P0 N803)
128	Actual Value Display Off. Display remains active during Single Step. If in addition M2242 is set, the Actual Value of the interpolator is completely ignored speeding up the bloc update time.

5.4.19   N902A F REFPULS (20)

Federate in mm/min. Used for Search of Reference pulse of an encoder or transducer, if N790 contains the value 32.

5.4.20   N902B MULTIPLIER FOR G75 (1000)
5.4.21   N902C MULTIPLIER FOR PARAMETRICAL CALCULATION (1000)

5.4.22   N903XYZUVABC I / O - INITIALVALUES (0)

These values will be put to the outputs of the I/O cards 1-8, when switching on or when changing to MENU. Example: A value of 3 in N903X sets the Outputs 1 and 2 on I/O 1.
If a value of 256 is programmed, the corresponding IO card will not be initialized.
Hint:
In Positioning Mode, when N902X4 is active, these machine data represent special functions described in page 79.

5.4.23   N904X JOYSTICK (0)

1: Joystick on X   2: Joystick on Y   4: Joystick on Z

5.4.24   N904Y CONTROL CODES (0)

1	Activate supervision of spindle during a movement
2	X-values of tool correction table are in diameter
4	Signal EXT STOP calls P9996
8	EXT. INTERRUPT only active in Automatic Mode
16	Range of hand wheel factor goes up to 1 mm per step
32	When LAGERROR is displayed, then do not disable the servomotors
64	The X value of an active tool has NO influence on G96
128	Activate X+, X-, Y+.... while program is stopped in AUTOMATIC Mode
256	Change from MANUAL mode to MANUAL INPUT only is possible over MENU
512	Texts for ↓81 are taken from P9031 (which can reside in the expansion memory)
1024	Texts for ↓81 are taken from P9000 (which can reside in the expansion memory)
2048	Change from MANUAL to AUTOMAT is allowed without going over MENU
4096	Signal „Programm Running" of I/O4 is put on output M10

The control codes 1 - 128 can also be set with M-functions. M2841 corresponds to 01, M2842 to 02, M2843 to 04, M2844 to 08 and so on.

5.4.25 N904Z TIME INTERVAL PLAYBACK (0)

Time interval in multiples of 10 ms after which the next position is memorized. The lowest input value is 10 corresponding to 100 ms.

5.4.26 N904U F FOR M23 (128)

Value between 0 and 128 corresponding to a speed of 0 to 100%.

5.4.27 N904V CONTROL CODES (0)

1	Do not display Actual Value of the slave gantry axis.
2	Slave gantry axis is allowed to move alone if programmed.
4	In Playback Mode memorize the Actual Position only if it has changed.
8	G81 to G83 are not modal.
16	Hand wheel over optional hand wheel input.
32	Clear Serial Input Buffer when not in External Data and when keyboard simulation is off (N902X2)
64	Do not propose G11, G12, G13 in INPUT and MANUAL INPUT Mode
128	Spindle output 0-10V is only activated when M3 or M4 active.
256	While sending blocs through the Serial Interface in External Data, an additional LF is sent after each CR.
512	Exchange the Keys Y+,Y- with Z+Z- (second row with third row)
1024	While waiting for an input to come active, the corresponding M-Function will be displayed on the screen.
2048	Display of the programmed S-Value will come from parametrical register #105.
4096	Generate screen border for C88-series
8192	

The control codes 1 - 128 can also be set with M-functions. M2941 corresponds to 01, M2942 to 02, M2943 to 04, M2944 to 08 and so on.

5.4.28 N904A G-Function for MANUAL INPUT and TEACH IN (0)

This G-function will be used as bloc when entering MANUAL INPUT or TEACH IN.

5.4.29 N904BC Automatical Distance Adjustment

5.4.30 N905X CONTROL CODES (0)

1	Switch off Spindle and Coolant with STOP and Single Step in the Automatic Mode
2	Switch off Spindle and Coolant at the end of a program in the Automatic Mode.
4	The in the Input Mode lastly processed program number will be proposed in the Automatic and Graphic Mode.
8	During program run the key MENU does not stop the program.
16	Activate collision control while G41/G42 (option)
32	In Manual Mode there is no need to push the keys REF and C twice.
64	No delay at the end of G09
128	With G95, the federate depends on the measured speed of the spindle and not of the programmed spindle speed.
256	With G95 the feed rate potentiometer is switched off.
512	During a movement, an External Interrupt can be generated from input E8 of I/O card 1.
1024	An External Start can be generated from input E7 of I/O card 1

5.4.31 N905Z F in Manual Mode (0)

Maximal speed in mm/min in the Manual Mode. With a value of 0 the FMAX in the Machine Data will be taken.

5.4.32 N905UV F Reduction in the Setup Mode (0)

During Setup Mode, the maximal speed is reduced to a value of 0 t0 128 corresponding to 0 to 100% of FMAX. This value is defined in N905V. The reduction depends on the Input of the I/O Card 4 which is defined in N905V:

0: No reduction.
8: Speed reduction when input I4 of I/O card 4 is active
16, 32, 64, 128: reduction when input I5, I6, I7, I8 is active.

5.4.33 N905A Waiting time for InPosition M27 (0)

Applies only for servo systems. A value of 100 waits around 10 seconds for InPosition.

5.4.34 N905B STOP Input (0)

In LINUX-CNC, this function allows to select up to 8 inputs of an I/O card by adding the values 1, 2, 4, 8 ...to the number of the I/O card multiplied by 256. If one of the inputs becomes active, one of the error message2 128 -135 from the file error.gr or error.en is displayed. The CNC stops and can be restarted with START. The polarity of the inputs can be changed to openers by adding the value 10000.
Example: Value for input 1and input 3 of IoCard 2: 2*256 +1 + 4
M48/M58 Disables /Activates N905B and N905C in the LINUX CNC

5.4.35 N905C STOP Input (0)

An additional STOP Input can be programmed on an I/O card. The input must be programmed, for example 1 for I1, 2 for I2, 4 for I3, 8 for I4. The number of the I/O card multiplied by 256 must be added.
Example: Input I4 (= value 8) on I/O card 1 (= value 256) should work as STOP input. The value to

enter is 264.

🐧 In LINUX-CNC, this function is the same as N905B, but the error messages are from 136 - 143

5.4.36 N906X CONTROL CODES (0)

1	Activate Hand box with LC-Display (9600 Baud!!) over Serial Interface Number 1
2	Select Serial Interface Number 2 for the Hand box
4	Supervision on of I5, I6, I7, I8 in Positioning Mode.
8	G08 stops on key STOP
16	Hand wheel is active also when X+, X-, Y+ ... is not hold pushed
32	Minipos with Feedratepoti
64	Actual Value Display in 10' of Inch, (divided by 2,54)
128	
256	Activate Error detection of Servo amplifier and Encoder input on DILAG
512	Activate Error detection of Servo amplifier on IO2 Inputs 5-8

5.4.37 N906Y CONTROL CODES (0)

5.4.38 N906Z FPROP (0)

Divider is (FMAX *256) / 255

5.4.39 N906U FPROP Basis (0)

5.4.40 N906V Maximal way to reference pulse (0)

🐧 If the reference pulse is not found within this distance in mm, the axes are stopped and an error message is generated. This function is implemented in the PCNC under Linux.

5.4.41 N906A Lubrification Pulse (0)

The time interval in minutes between 1 and 255 will be programmed here. The pulse is output on O8 on I/O card 1, however only while an axis is moving. The pulselenght is 1 second.

5.4.42 N906B Initial State of Gantry axes (0)

For example 8014 which makes axis 1 a master and axis 4 a slave.

5.4.43 N906C Safety function for M03 (0)

🐧 With this function activated, the M03 command is expanded with some safety functions. The value to input is of the format vxyzzzz.
The value zzzz defines a time in milliseconds with which the M03 output is toggled. With zzzz = 1000 the M03 output is on for 1 second and off again for 1 second. In case of

5.4.44 N910XYZU Joystick X (127,3,3,127)
5.4.45 N911XYZU Joystick X (127,3,3,127)
5.4.46 N912XYZU Joystick X (127,3,3,127)

X= left maximum, Y= left starting point, Z= right starting point, U= right maximum.
Between the left and right starting point is a zone where a movement of the joystick does not result in a movement of the axis. When the joystick is at the maximum and when the resulting federate is to small, then the value of X or U must be **decreased**.

5.4.47   N913XYZUVABC SMAX for optional S-Outputs 1-4 and 5-8 (0)

All values must be defined in order to work properly!

5.4.48   N914X Range for FPROP Pulse card

5.4.49   N914Y Delay after RESET (0)

Delay in 10 ms units after Reset until MOTOR ON is activated.

5.4.50   N914Z Bolz Flight time (0)

5.4.51   N915 Supervision of Analog Output Voltage going to Servodrivers(0)

5.4.52   N916 X, Y Lower and Upper Limit of Steps for Handwheel(0, 0.100)

5.4.53   N920Z Maximal number of blocs in the Interpolator Cache

Default is the number which fits in the available RAM and is typically about 200 blocs. If necessary it can for example be set to 50 blocs.

5.4.54   N921XYZU Screen Adaptation

X,Y: Pixel number in X and Y direction
Z,U: Length of the screen in X and Y direction

Example:
5"   LCD Milling: X=320, Y=240, Z=114, U=85
5"   LCD Turning: X=640, Y=240, Z=114, U=85
9"   LCD Milling: X=640, Y=240, Z=211, U=156
10" LCD Milling: X=640, Y=340, Z=260, U=175

5.4.55   N922U Maximal error of circle (1)

Normally the feed rate is reduced so that a maximal circle error of 1 increment is accepted. With a value of 0, always the highest possible feed rate is used taking into account that the circle error is higher.

5.4.56   N923Y Smooth Factor for feed rate potentiometer (5)

The maximum value is 8, which however makes the reaction of the potentiometer very slow.

5.4.57   N923U Feed rate Potentiometer / External Hand wheel (16)

16   A change of this value in the internal or external feed rate potentiometer is recognized and control over the feed rate is given to the potentiometer which was changed. This value can be changed between 1 and 255.

4096   will select a divider of 4 for the External Handwheel

When not wanting to change the setting of the feedrate potentiometer, a value of 4112 must be input for selecting a divider of 4 for the External Handwheel!

5.4.58	**N925X**	**Pulses of spindle encoder ....**
5.4.59	**N925Y**	**per rotation**

The number of pulses per rotation of the spindle can be programmed. The maximal input frequency is 60 Khz, so if a motor with 3000 rot/min must be synchronized, the encoder can have a maximum of 1200 pulses/rot.

5.4.60	**Offset Spindle Reference Pulse (0)**

Offset in number of spindle pulses.

5.4.61	**N925B**	**Integration Value for Spindle Speed Display (32000)**

When the spindle speed is displayed on the screen, this value is used as timeperiod for integrating the spindle pulses coming from the encoder. Values can be 1000 µs, 2000, 4000, 8000.....

5.4.62	**N925C**	**Integration Value for G33 Spindle Pulses (32000)**

The same as with N925B applies, however this value is used for smoothing the speed used for G33.

5.4.63	**N927V**	**Memory Reservation Moving Back On Contour (0)**

In Preparation.

6   GENERAL INFORMATIONS

6.0.1   Initialization of CNC

During StartUp after Switching On or after a RESET, the CNC checks if the Key ↓ is pressed. If yes, the message **CODE** is displayed. After inputting $\boxed{0}$ followed by ➡, the complete memory incuding everything is cleared.

6.0.2   Checksum Error

Each bloc in the memory of the CNC has a checksum byte. When the bloc is readout from the memory, this byte is checked. If an error is detected the message **CHECKSUM ERROR** is displayed, and the bloc where the error occured is marked with „!".The AUTOMATIC MODE will be interrupted. Possible causes for the occurance of this error:
* spike on the powerline during the moment of memorizing.
* dataloss caused by a voltage too low of the buildin accumulator.
* memorizing a bloc when the memory contents are already bad.
* erroneous memory chip.

Help:
* clear the complete memory as often as possible.
* If only one bloc has an error, the bloc can be erased and then memorized one more time.
* filter the powerline.

6.0.3   Error handling P9998

If P9998 is in memory, it will be invoked in the AUTOMATIC MODE if an error has occured (Limit Switch, external interrupt..). In the register #40 is the error code which can be displayed with ↓94 #240.

6.0.4   CNC displays „LAG ERROR"

* at the end of a displacement: The CNC does not succeed to reduce the Lag Distance of an axis to a value lower than „IN POSITION".
* during a movement: The value FMAX in the machine data is to high, or the P-FACTOR is not correctly input.

6.0.5   Keyboard simulation with V24 interface

The CNC keyboard can be simulated on an external computer using the Serial Interface.

External computer	Generated function	External computer	Generated function
@	+X	O	CLEAR MEMORY
A	-X	P	SEARCH bloc
B	+Y	Q	DELETE bloc
C	-Y	R	EXTERNAL DATA
D	+Z	S	
E	-Z	T	SPINDLE
F	SINGLE bloc	U	COOLING
G	START	H	STOP
I	MANUAL	<	MENU

External computer	Generated function	External computer	Generated function
J	GRAPHIC	=	**->**
K	AUTOMATIC	>	CLEAR
L	MANUAL INPUT	.	.
M	TEACH IN	0-9	0-9
N	REFERENCE POINT $		

- ENTER, ? INPUT MODE or STORE into memory
- /   returns given position, status, operating mode, error status. The status corresponds to the output of I/O card 4, values are ASCII.
- !   returns Actual Value from the DILAG cards. This is the Actual Value of the axes XYZU and VABC with a time jitter of aproxymatly 500 ns.
- &   returns given position in HEX format, 8 values of 8 characters, 2 characters forming 1 byte, LSB first.
- „   returns the numbers of the programs in memory.
- (state of the inputs and outputs of the i/o cards 1 - 8.
-)   returns the following values: POT%, programmed F, actual F, T
- *   returns Software Version of the CNC

Programs also can be sent to the CNC over the Serial Interface (X21) by Remote Control.This can be done for example by sending the following data to the CNC:
R7>=
P0001 CR
N1G0X55 CR
% CR
- R Switch to EXTERNAL DATA
- 7 Select input through Serial Interface
- > Clear input field
- = ➡
- Then follows the program to be transmitted.

These functions can be simulated and tesed with each TERMINAL program like Hyperterminal or of course our DIENSTPROGRAM.

The following BASIC program allows to send data from the keyboard of a PC to the CNC:
10 CLS:OPEN „com1:9600,n,8,1,RS,CS,DS,CD" AS #1
20 REM Keyboard scan.
30 A$=INKEY$:IF A$=""THEN 30
40 REM Wait until CNC is ready for receiving data.
50 IF (INP(&H3FD)AND 64)=0 THEN 50
60 IF (INP(&H3FE)AND 16)=0 THEN 60
70 PRINT #1, A$
80 GOTO 30

If COM2 is to be used, correct line 10 (COM2), line 50 (&H2FD) and line 60 (&H2FE).

6.0.6	Continuous movement

The continuous movement of the CNC will be interrupted when:
- CNC blocs cannot be loaded fast enough into the Interpolator. That is why, the Actual Value display and bloc scrolling should be disabled with G13 M21 M22 M28.

- successive blocs are not tangent like the sides of a rectangle. Depending on the machine data (F Start/Stop, acceleration) a certain difference of the tangents is allowed for moving with constant speed.
- more than 50 blocs have to be executed per second. At a speed of F=10000 mm/min, the movements must be at least 2 mm long.
- the following bloc is not at least so long that the CNC can stop without position loss, when the key STOP is pressed. That is why the machine data FSTART/STOP and ACCELERATION must be adjusted to their optimum. Alternatively the Interplator Cache can be activated with M32.
- a long linear movement joins a short Circle Segment with small radius. The radial acceleration for one axis then mostly is higher than the acceleration defined in the machine data. Here the feedrate should be reduced before the Circle Segment.

P0001 TESTPROGRAM FOR CONTINOUS MOVEMENT
N001 G91
N002 G13 M21 M22 M28
N003 G00 X10
N004 G00 X10
....
N010 G00 X-10
N011 G00 X-10
....
N100 G20 P0001 N001

By slowly opening the feedrate potentiometer, a feedrate can be detected where the movement will no more be continous and which allows to evaluate the possibilities of th CNC..

6.0.7   Positioning Control

This mode is intended for moving to a position, then to activate a motor brake, and to send a signal „Position reached" or „End of bloc" to a master control.
The the CNC waits for an „External Start", then opens the brake and moves to the next position.
In order to select this mode, the machine datum N902 X12 V2 must be programmed.
The inputs and outputs on the I/O card number 4 have the following meaning:

In 1	External Interrupt	O 1	Clamping X
In 2	External Stop	O 2	Clamping Y
In 3	External Start	O 3	Positioning=1 / End of bloc=0
In 4	Keyboard off, only ext. Start/Stop activ	O 4	Brake X
In 5	Wait for pressure Y at end of movement	O 5	Brake Y
In 6	Wait for pressure X at end of movement	O 6	Pulse if position reached
In 7	Wait for pressure X at beginning of movement	O 7	Programm started = 1 / end = 0
In 8	Wait for pressure Y at beginning of movement	O 8	Error output

- O3 Positioning ON
- Activate MOTOR ON Signal
- O1 Clamping ON
Variable Delay defined in N903Z
O4 Brake ON
Wait for I7 Pressure present
Wait until I3 (EXT. START) is inactive

Execute the programmed movement

Wait until Lag Distance has stabilized the number of checks defined in N903A. If N903A is 0, then terminate.

Compensate the Lag Error until the Timeout in ms defined in N903B has been reached. After this Timeout, the message ERROR "MECHANIK BLOCKED" is generated.

6.0.8   Steppingmotor Supervision with Encoder

Select Steppingmotor with N790 and activate the DILAG Card with N813X.
The DILAG receives from the INTERPOLATOR the desired position and from the encoder input the actual position. If the LAG ERROR in N804 is exceeded, an error message will be diplayed. If the steps per revolution of the encoder and the stepping motor differ, then the number of steps of the motor can be adapted with N808, and the steps of the encoder with N808.

6.0.9   Memory Expansion

The memory expansion uses the programnumbers P9000 to P9031, and allocates 32 Kbyte of memory to each of these programs giving in all 1MB. With 4 MB, programs P9000 – P9127 are possible.
If in EXTERNAL DATA, a program of f.e. 100 Kbyte is to be transmitted to the CNC, the programnumber P9000 is input in the CNC und the transmission is started. The CNC automatically detects if 32Kbyte are transmitted, inserts a G20 P9001 at the end of P9000, generates a new program P9001 and inserts there the following blocs which are transmitted.
In the AUTOMATIC mode, the complete program starting with P9000 is executed followed by P9001 and so on.
Hint:
Programs P90xx cannot be copied and can only be output as single programs in EXTERNAL DATA.

6.0.10   SubProgramCall over I/O-CALL

If, during programexecution, a subprogram has to be called when an input is activated, this function is suitable. However when M2342 (M32) is active, the subprogram will be called only when the Interpoaltorcache is empty!

Example: Programnumber P0011 (N901C10) should be called when Input1 (N901A1) on I/Ocard1 (N901V1) becomes active (N901B1) during programexecution. See also page 69

In P0011, the Z axis must move up 100mm, wait until START is pressed, so that for example a new tool can be inserted, then move down again and continue in the place where it was interrupted.

```
P0011
N1 [96 #0                ;Save Status G90/G91
N2 G91                   ;Switch to G91
N3 G0 Z100               ;Move up Z-axis
N4 G13 M19               ;Wait until all movement finished
N5 G13 M9 M5 M0          ;Switch off Spindle and Coolant and wait for START
N6 G13 M3 M8             ;Switch on Spindle and Coolant
N7 G0 Z-100              ;Move down
N8 [96 #1                ;Restore Status G90/G91
N9 G13 M19               ;wait until all movement finished
```

6.0.11   Sample programs for training

Here is how to program a simple calculator that takes 3 inputs from user as the first number X, second number Y, and the operator (1 for addition, 2 for subtraction, 3 for multiplication and 4 for division) and prints out the result to screen.

```
P0099 A SIMPLE CALCULATOR       ;Program number and name
N0001 G13 M2241 M2248 M.... M.... ;Bloc updating and actual value displays are off
N0010 [80 INPUT1      X=        ;Text inputs for variables
N0011 [80 INPUT2      Y=
N0012 [80 OPERATOR       0=
N0013 [80 RESULT       R=
N0050 [91   #000  #...  #...    ;Clear the screen
N0051 [00   #000 =     @+...0,050  ;Assign value 50 as the text start line
N0052 [83   #010  #008  #003  #...  ;Open input field with 8 characters of which 3 are after
                                    ;decimal point. The value typed in will be stored in #010
N0053 [00   #000 =     @+...0,060  ;Line 6
N0054 [83   #011  #008  #003  #...  ;Open input field with 8 characters of which 3 are after
                                    ;decimal point. The value typed in will be stored in #011
N0055 [00   #000 =     @+...0,070  ;Line 7
N0056 [83   #012  #002  #000  #...  ;Open input field with 2 characters. The value typed in will be
                                    ;stored in #012
```

From now on program starts to check the operator(#012) and makes the suitable calculation.
For this we occupy a free register #009.

```
N0057 [02   #009 = #012 - @+...0,001 ;Subtract 1 from the operator input(#012) and store it to #009

N0058 [54   (JUMP NZ TO) N0061  ;If the result is not equal to zero jump to line 61
N0059 [01   #013 = #010 + @.....#011 ;Add X and Y, assign the value to RESULT (#013)
N0060 [53   (JUMP   TO) N0500   ;Jump to line 500
N0061 [02   #009 = #012 - @+...0,002 ;Subtract 2 from the operator input(#012) and store it to #009
N0062 [54   (JUMP NZ TO) N0065  ;If the result is not equal to zero jump to line 65
N0063 [02   #013 = #010 - @.....#011 ;Subtract Y from X, assign the value to RESULT (#013)
N0064 [53   (JUMP   TO) N0500   ;Jump to line 500
N0065 [02   #009 = #012 - @+...0,003 ;Subtract 3 from the operator input(#012) and store it to #009
N0066 [54   (JUMP NZ TO) N0069  ;If the result is not equal to zero jump to line 69
N0067 [03   #013 = #010 * @.....#011 ;Multiply X and Y, assign the value to RESULT (#013)
N0068 [53   (JUMP   TO) N0500   ;Jump to line 500
N0069 [02   #009 = #012 - @+...0,004 ;Subtract 4 from the operator input(#012) and store it to #009
N0070 [54   (JUMP NZ TO) N0900  ;If the result is not equal to zero jump to line 900
N0071 [04   #013 = #010 / @.....#011 ;Divide X to Y, assign the value to RESULT (#013)
```

```
N0500 [00    #000 =      @+...0,080      ;Line 8
N0501 [83    #013  #...  #...  #...       ;Print RESULT (#013) as 8 character (3 after coma as default)
N0900 [80 END                            ;Simply end the program
%
```

This main program P0300 seeks for an input signal from user and recalls the suitable
sub-program when the signal is received from I/O card:

```
P0300 MAIN LED SWITCH                    ;Program number and name
N0001 G13 M2241 M2248 M.... M....        ;Bloc updating and actual value displays are off
N0002 [00    #010 =      @+...0,500      ;Set register #010 to 0.5
N0010 G23 P0301  N....   W0000  M0161    ;When input 1 signalled recall program P0301
N0011 G23 P0302  N....   W0000  M0162    ;When input 2 signalled recall program P0302
N0012 G20 P0300  N....                   ;Jump to first line
%
```

This is the first sub-program that switches on 8 leds one by one and then switches
them off in reverse order:

```
P0301 SCHEME 1
N0001 G13 M0141 M.... M.... M....        ;Switch on led1
N0002 G04 H+....#010                     ;Dwell time 0.5sec as stored in #010
N0003 G13 M0142 M.... M.... M....        ;Switch on led2
N0004 G04 H+....#010                     ;Dwell time 0.5sec as stored in #010
N0005 G13 M0143 M.... M.... M....        ;Switch on led3
N0006 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0007 G13 M0144 M.... M.... M....        ;Switch on led4
N0008 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0009 G13 M0145 M.... M.... M....        ;Switch on led5
N0010 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0011 G13 M0146 M.... M.... M....        ;Switch on led6
N0012 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0013 G13 M0147 M.... M.... M....        ;Switch on led7
N0014 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0015 G13 M0148 M.... M.... M....        ;Switch on led8
N0016 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0017 G13 M0158 M.... M.... M....        ;Switch off led8
N0018 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0020 G13 M0157 M.... M.... M....        ;Switch off led7
N0021 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0022 G13 M0156 M.... M.... M....        ;Switch off led6
N0023 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0024 G13 M0155 M.... M.... M....        ;Switch off led5
N0025 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0026 G13 M0154 M.... M.... M....        ;Switch off led4
N0027 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0028 G13 M0153 M.... M.... M....        ;Switch off led3
N0029 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0030 G13 M0152 M.... M.... M....        ;Switch off led2
N0031 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0032 G13 M0151 M.... M.... M....        ;Switch off led1
N0033 G04 H.....#000                     ;Dwell time 0.5sec as stored in #010
%
```

And this is the second sub-program that switches on and then off
all the leds in order:

```
P0302
N0001 G13 M0141 M.... M.... M....        ;Switch on led1
N0002 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0003 G13 M0151 M.... M.... M....        ;Switch off led1
N0004 G13 M0142 M.... M.... M....        ;Switch on led2
N0005 G04 H.....#010                     ;Dwell time 0.5sec as stored in #010
N0006 G13 M0152 M.... M.... M....        ;Switch off led2
N0007 G13 M0143 M.... M.... M....        ;Switch on led3
```

```
N0008 G04 H.....#010                    ;Dwell time 0.5sec as stored in #010
N0009 G13 M0153 M.... M.... M....       ;Switch off led3
N0010 G13 M0144 M.... M.... M....       ;Switch on led4
N0011 G04 H.....#010                    ;Dwell time 0.5sec as stored in #010
N0012 G13 M0154 M.... M.... M....       ;Switch off led4
N0013 G13 M0145 M.... M.... M....       ;Switch on led5
N0014 G04 H.....#010                    ;Dwell time 0.5sec as stored in #010
N0015 G13 M0155 M.... M.... M....       ;Switch off led5
N0016 G13 M0146 M.... M.... M....       ;Switch on led6
N0017 G04 H.....#010                    ;Dwell time 0.5sec as stored in #010
N0018 G13 M0156 M.... M.... M....       ;Switch off led6
N0019 G13 M0147 M.... M.... M....       ;Switch on led7
N0020 G04 H.....#010                    ;Dwell time 0.5sec as stored in #010
N0021 G13 M0157 M.... M.... M....       ;Switch off led7
N0022 G13 M0148 M.... M.... M....       ;Switch on led8
N0023 G04 H.....#010                    ;Dwell time 0.5sec as stored in #010
N0024 G13 M0158 M.... M.... M....       ;Switch off led8
%
```

7   SETUP OF CNC

The CNC needs a power supply of 230V. For a quick first test, the CNC can be connected to the mains and switched on. After some seconds, the CNC will display a DOWNLOAD message and then enter the main MENU.

At the first start, the CNC should be initialized (see page 77).

Now in the INPUT MODE, a short programm can be entered and displayed in the GRAPHIC MODE.(programm example on page 22)

7.0.1   Stepping motors

If the drivers for the Stepping Motors were bought from Dipl.-Ing. Engelhardt GmbH, the motors need only to be connected to the exits MOTOR X, MOTOR Y and so on. The connection can be found on page 106, and differs for motors with 2 phases, 3 phases and 5 phases.

Then the plug X84 of the CNC is connected to the plug X84 of the driver. X84 delivers the stepping pulses and the stepping direction to the drivers.

If the drivers are from another manufacturer, the plug X84 (page 102) must be wired, first only the X-motor. For this, **0V Internal**, **Clock X** and **Right/Left X** are to be used.

In the machine data P0, the following bloc must be inserted: **N790 X3**. This activates stepping motor mode and disables the Limit Switches.

Then in Manual Mode, X+ and START is pressed, the motor now can be moved under control of the feedrate potentiometer.

NOTE: To acomplish this test, only P0 N790 N3 must be programmed, nothing else!!!

Next, the steps/mm in P0 N706 and N707 (page 64) must be adjusted.

After that, N700 to N704 must be determined. For this the following program can help:

P1
N1 G00 X100
N2 G00 X-100
N3 G20 P1

If it is started in the Automatic Mode, the effect of the individual machine data during the run of the motor can be observed. A goal of this procedure is it to obtain a troublefree movement of the motor without loss of steps at all speeds adjustable with the feed potentiometer.

7.0.2   Servomotors

If the servodrivers were purchased from our company, then the servomotors have only to be connected to the corresponding outputs MOTOR X, MOTOR Y .... The pin connections can be found on page 106 and differs for DC- and AC- servomotors. After that the connector X90A of the CNC is wired to X90A on the servodriver unit. X90 delivers **±10V** and **MOTOR ENABLE** to the servodriver.

If the servo driver was purchased from an other company, the connector X90A (page 105) must be wired, first only the X-motor. For that pins **DC-OUT+**, **DC-OUT-**, **MOTOR ENABLE+** and **MOTOR ENABLE-** must be used.

The servoamplifier must have a Differential Input. Those pins must be connected to **DC-OUT+**, **DC-OUT-**. With 0 - 10V at the input, the speed of the servomotor must be adjustable between 0 and FMAX in P0. The maximal speed of the motor is not necessarily 3000 rot/min, but can be much lower f.e. 600 rot/min. With a spindle pitch of 5mm this will result in a feedrate of 3000 mm/min!

Each servoamplifier must have 2 pins, which, when shorted, activate the amplifier. It is important to get to know, which of these 2 pins is the positive one! It must be wired to **MOTOR ENABLE+** whereas the negative one is wired to **MOTOR ENABLE-**.

In the machine data P0 the following blocs must be inserted:

· N790 X19:   Servomode on, Limit Switches disabled.
· N813 X1   X-axis activated

After that change to Manual Mode, select X+, and START. By opening the feedratepotentiometer for a short time, a small Lag Distance is generated in the DILAG resulting in a small output voltage going to the servoamplifier. The servomotors starts moving with the corresponding speed. Because there is no feedback through the encoder, the Lag Distance will stay constant and the servomotor will move at constant speed. The Lag Distance can be displayed by pushing the key „2".

It can vary between +/- 1600 increments. At higher values an error message „LAG ERROR" will be displayed.

If this test is finished successfully, the motor can be fixed to the mechanical axis. Then the encoder is wired according to X29, page 101. If the encoder is wired correctly, the X-axis can be moved in the MANUAL MODE under control of the feedratepotentiometer. It could be that the Counting Direction of the Encoder is not correct and must be reversed by rewiring according to page 101.

As next the steps/mm are adjusted with the machinedata P0 N706 and N707 (page 64). After that N700 - N704 can be adjusted with the help of the following program:

P1
N1 G00 X100
N2 G04 H1
N3 G00 X-100
N4 G04 H1
N5 G20 P1

This program can be started in the Automatic Mode. The effect of the different machine data on the run of the axis can be observed. The goal of this adjustment is to get a smooth movement of the axis at all speeds adjustable with the feedratepotentiometer.

If the feedratepotentiometer is fully opened, the Lag Distance should display between 1000 and 1400 increments (to be viewed by pushing the key „2"), with a feedrate of 0, the Lag Distance also should be 0. If not, it can be adjusted with the OFFSET trimmer on the servo driver until the Lag Distance oscillates slowly between 0 and 1.

7.0.3   Coordinate System Milling

If the X axis is commanded to move to the positive direction, a milling action should take place to the right, during positive counting of the Y axis, the milling action should take place to the rear, thus away from the viewer. During negative counting of the Z-axis, the milling should go towards the workpiece. This corresponds to the Cartesian Coordinate System.

7.0.4   Coordinate System Turning

During negative counting of the X and the Z-axis, the Tool should go towards the workpiece.

7.0.5   Limit Switches

The Limit Switches will be connected to X9 (page 100) or X85 (page 103). As always, first only the X-axis will be wired. An external 24V- supply must be used for powering the Limit Switches. The **Positive Direction Limit Switch** will be activated when the axis is moving in the positive direction. At this moment must be distinguished between **Closers** or **Openers** as Limit Switches. A Closer is switch outputting 24V when activated whereas an Opener outputs 0V when activated. Normally Openers are used because they simulate an activation if a wire has broken.

If it is not known if the switch is Opener or Closer, the voltage between the **output** and **0V Extern** can be measured with a voltmeter. If the voltage is +24V and the switch is not activated, there is an Opener.

According to the above result, the machinedata P0 N790 must be corrected. Add 256 for the X+ limitswitch and 512 for X- limitswitch. With a steppingmotor system we have a total of 771, with servomotors 878. If closers were used we will have to add 1024 plus 2048. See also page 66.

If the X axis is moved slowly in the Manual Mode to a limit switch, it will stop immediatly when the Limit Switch is reached and an Error Message will be displayed. The axis can only be moved back in the opposite direction

7.0.6   Offset Alignment of Servomotors

At stop the Lag Distance of all axes must be equal to zero. It can be observed in the Manual Mode
by pushing 2

Due to offset voltages in the D/A converters of the CNC and in the input stages of the servo drivers,
this Lag Distance however mostly is not equal to 0. It must be adjusted with a trimmer in the front
plate of the servo driver. If however the adjustment range is not sufficient, an individual offset for
each axis can also be programmed in P0 N812. (See page 68)

Additionally, a program P9993 is implemented into the CNC, which can accomplish the offset
alignment automatically. P9993 is started in the Automatic Mode. With 1 , 2 , 3 , the axes 1, 2, 3
can be adjusted automatically. The determined offset value is then stored in P0 N812.

8   INSTALLATION PROGRAMS

These programs demonstarte the installation of a turning machine with stepping motors and a Tool Changer. It is assumed that the following inputs and outputs are used:

I1 - I6:   Toolposition 1 to 6 is recognized. If only 4 toolpositions are available, I5 and I6 must both be connected to +24V.
O1:   Toolchange Motor move forward
O2:   Toolchange Motor move backward

8.0.1   P0000 MACHINEDATA STEPPING MOTORS

N700	X0004000 Z0006000	;Maximal feedrate for X and Z
N701	X0000050 Z0000050	;Starting feedrate
N702	X0000050 Z0000050	;Stopping feedrate
N703	X0000100 Z0000100	;acceleration of movement
N704	X0000100 Z0000100	;deceleration of movement
N706	X0000500 Z0001250	;in X there is 500 steps for 2 mm,
N707	X0000002 Z0000006	;in Z there is 1250 steps for 6 mm
N710	X0004000 Z0004000	;feedrate for G74
N790	X0000771 Z0000771	;1 (Main Axis) +2 (Linear Axis) +256 (Limit Switch in ;positive direction) +512 (Limit Switch in negative direction)
N813	X3	;Servoaxes X, Z are activated
N902	Z1	;Textdisplay in english=1 (german=0)
N903	X0000256 Y0000256	;IO1 and IO2 are not reset when changing to MENU
N904	A000001	;G01 is presented in MANUAL INPUT
N905	X0000128	;when G95 is active, the feedrate is proportional to speed of the ;spindle and not to the programmed spindlespeed.
N921	X640 Y240 Z114 U85	;resolution of LC-display
N925	X0001024 Y0000001	;1024 pulses for 1 rotation of spindlemotor

8.0.2   P0074 REFERENCE POINT

P0074 is used for referencing all the axes.
The following program must be in memory:
P0074 REFERENCE POINT

N001	G11 T 0	;T0 must be selected with G11 or G36
N010	G74 X 0	;The positiv limit switches
N020	G74 Z 0	;must be moved to!!!
N040	G92 X0 Z 0	;This bloc number must be N40 !!!!

This program is used by G58. (see page 30) which determines the offset from the Reference Point to the Program Starting Point.
These values are deposit in bloc N40. With the next call of P74, these values are finally put into the Actual Value display.

8.0.3   P9900 TOOL TABLE

P9900 is the Tool Table. It must be present in memory when a tool is invoked with G11 T... or with G36 T....
T001 is the Reference Tool, it must be the longest tool of all tools physically used. It should have the length of 0 in X, and Z.
The first pair of X, Z of the remaining tools should contain the lenght difference to T1. The second pair of X,Z are used to correct the tool when the lenght chnages during work. In fact both pairs of X,Z are added together in order to generate the actual toollenghtcompensation.
O is the orientation between 0 and 9 of the tool (0 and 5 disables the toolorientation)
T001   X ....0,000 Z ..0,000 X ....0,000 Z ..0,000 R ...0,000 O..0

```
T002    X ....4,000  Z+..2,000  X ....0,000  Z ..0,000  R ...0,000  O..0
T003    X ....6,000  Z+..3,000  X ....0,000  Z ..0,000  R ...0,000  O..0
T004    X ....8,000  Z+..4,000  X ....0,000  Z ..0,000  R ...0,000  O..0
T005    X ..10,000  Z+..5,000  X ....0,000  Z ..0,000  R ...0,000  O..0
T006    X ..12,000  Z+..6,000  X ....0,000  Z ..0,000  R ...0,000  O..0
```

8.0.4　P9936 Tool Change

P9936 is sample Tool Change Programm which can be adapted to the needs of the user. When the user programms the bloc **G36 F100 S2 T3 M3**, he wants in this case to have a feedrate of **F100**, the spindle gear number **S2**, the tool number **T3** and the spindle on clockwise **M3**. When **G36 F100 S2 T3 M3** is executed, P9936 is invoked and the values of **F,S,T,M** are put into the CNC registers #80, #81 #82, #83. #90 contains a value indicating wether F or S or T or M have been programmed in G36.

;Definition of the Spindle Speed Ranges for each Spindle Gear.

```
N001  [00  #071 = @+...0,180       ;S-gear 1 from 0 to 180 rpm
N002  [00  #072 = @+...0,500       ;S-gear 2 from 181 to 500 rpm
N003  [00  #073 = @+...1,000       ;S-gear 3 from 501 to 1000 rpm
N004  [00  #074 = @+...1,800       ;S-gear 4 from 1001 to 1800 rpm
```

;Test if F has been programmed

```
N010  [00  #092 = @+...0,128
N011  [18  #092 = AND   #090
N012  [50  (JUMP ZER TO) N20       ;if F was not programmed then JUMP
N013  G11  F..#080                  ;else use the programmed F for the next movement
```

;Test if S has been programmed

```
N020  [00  #092 = @+...0,064
N021  [18  #092 = AND #090
N022  [54  (JUMP NZ  TO) N800       ;S was programmed, JUMP to Gear Changer
```

;Test if T has been programmed

```
N030  [00  #092 = @+...0,032
N031  [18  #092 = AND #090
N032  [54  (JUMP NZ  TO) N60        ;T was programmed, JUMP to Tool Changer
```

;Test if M has been programmed

```
N040  [00  #092 = @+...0,016
N041  [18  #092 = AND #090
N042  [50  (JUMP ZER TO) N990       ;M was not programmed, JUMP to the end ofP9936
```

;M has been programmed

If M03 or M04 was programmed, execute the function with G11 and ;then wait in N050 for the input 5 of IO2. This input should become active ;when the spindle motor starts working.

```
N043 G11  F...... S...... T....   M#083
N044  [02  #082 = #083 - @+...0,003
N045  [50  (JUMP ZER TO) N0050
N046  [02  #082 = #083 - @+...0,004
N047  [54  (JUMP NZ  TO) N0990
N050 G13  M0265
N051  [53  (JUMP    TO) N0990
```

;T has been programmed, make tool change!

```
N060  [04  #080 = #082 / @+..10,000
N061  [54  (JUMP NZ  TO) N0065
N062 G22  N0200                          ;Change to requested tool
N063 G11  T#082
N064  [53  (JUMP    TO) N0040
N065  [10  #089 = COPY   #082
N066  [10  #082 = COPY   #080
N067 G22  N0200                          ;Change to requested tool
```

```
N068 G11  T#089
N069 [53 (JUMP    TO) N0040
N100 [02  #080 = #081 - @+...0,001
N101 [54 (JUMP NZ  TO) N0110
N102 G13  M0251 M0252 M0253 M0254
N103 G13  M0241 M0243 M0261 M0263
N104 [53 (JUMP    TO) N0190
N110 [02  #080 = #081 - @+...0,002
N111 [54 (JUMP NZ  TO) N0120
N112 G13  M0251 M0252 M0253 M0254
N113 G13  M0242 M0243 M0262 M0263
N114 [53 (JUMP    TO) N0190
N120 [02  #080 = #081 - @+...0,003
N121 [54 (JUMP NZ  TO) N0130
N122 G13  M0251 M0252 M0253 M0254
N123 G13  M0241 M0244 M0261 M0264
N124 [53 (JUMP    TO) N0190
N130 [02  #080 = #081 - @+...0,004
N131 [54 (JUMP NZ  TO) N0140
N132 G13  M0251 M0252 M0253 M0254
N133 G13  M0242 M0244 M0262 M0264
N134 [53 (JUMP    TO) N0190
N140 [02  #080 = #081 - @+...0,000
N142 [54 (JUMP NZ  TO) N0150
N143 G13  M0251 M0252 M0253 M0254
N144 [53 (JUMP    TO) N0190
N150 [51 (JUMP POS TO) N0190
N153 [96  #004    #085
N154 [01  #081 = #086 + @+...0,001
N155 [02  #080 = #081 - @+...0,005
N156 [52 (JUMP NEG TO) N0159
N157 [00  #081 =       @+...0,001
N159 [53 (JUMP    TO) N0100
N190 G11  S..#081
N199 [53 (JUMP    TO) N0030
```

;Change to requested tool
;Was tool 1 requested?
```
N200 [02  #081 = #082 - @+...0,001
N201 [54 (JUMP NZ  TO) N0220
```
;change to tool #1
```
N202 G23  N0490  M0161              ;Jump to N490 if tool #1 already present
N203 G13  M0141  M0161 M0151 ;Toolchange motor forward, wait for tool #1, motor off
N204 [53 (JUMP    TO) N0480
```
;Was tool 2 requested?
```
N220 [02  #081 = #082 - @+...0,002
N221 [54 (JUMP NZ  TO) N0240
```
;change to tool #2
```
N222 G23  N0490  M0162              ;Jump to N490 if tool #2 already present
N223 G13  M0141 M0162 M0151    ;Toolchange motor forward, wait for tool #2, motor off
N224 [53 (JUMP    TO) N0480
```
;Was tool 3 requested?
```
N240 [02  #081 = #082 - @+...0,003
```

```
N241  [54  (JUMP NZ  TO) N0260
;change to tool #3
N242 G23  N0490  M0163          ;Jump to N490 if tool #3 already present
N243 G13  M0141 M0163 M0151     ;Toolchange motor forward, wait for tool #3, motor off
N244  [53  (JUMP    TO) N0480
;Was tool 4 requested?
N260  [02  #081 = #082 - @+...0,004
N261  [54  (JUMP NZ  TO) N0280
;change to tool #4
N262 G23  N0490  M0164          ;Jump to N490 if tool #4 already present
N263 G13  M0141 M0164 M0151     ;Toolchange motor forward, wait for tool #4, motor off
N264  [53  (JUMP    TO) N0480
;Was tool 5 requested?
N280  [02  #081 = #082 - @+...0,005
N281  [54  (JUMP NZ  TO) N0300
;change to tool #5
N282 G23  N0490  M0165          ;Jump to N490 if tool #5 already present
N283 G13  M0141 M0165 M0151     ;Toolchange motor forward, wait for tool #5 motor off
N284  [53  (JUMP    TO) N0480
;Was tool 6 requested?
N300  [02  #081 = #082 - @+...0,006
N301  [54  (JUMP NZ  TO) N0320
;change to tool #6
N302 G23  N0490  M0166          ;Jump to N490 if tool #6 already present
N303 G13  M0141 M0166 M0151     ;Toolchange motor forward, wait for tool #6, motor off
N304  [53  (JUMP    TO) N0480
N320 G22  N0900                 ;What is active tool?
N321  10  #082 = COPY   #081
N322  [53  (JUMP    TO) N0200
;toolchanger backwards, wait for lock-in, wait 0,5 seconds, toolchanger off
N480 G13  M0142 M0167
N481 G04  H+...0,500 DWELL
N482 G13  M0152
;End of toolchange
N490  [53  (JUMP    TO) N0990


;S was programmed
N800 G13  M9000                 ;output voltage to S is 0V
N801  [02  #080 = #081 - @+...0,004
N802  [51  (JUMP POS TO) N0810  ;if programmed S-function is > 4
N803 G22  N0100                 ;change the S-gear stage 1 - 4
N804 G13  M9255                 ;give full S-speed
N805  [53  (JUMP    TO) N0030   ;next function


N810  [02  #080 = #081 - @.....#071
N811  [51  (JUMP POS TO) N0820  ;if programmed S is > 1. gear stage
N812  [03  #080 = #081 * @+...0,255
N813  [04  #080 = #080 / @.....#071
N814  [00  #081 =        @+...0,001    ;gear stage 1
N815  [53  (JUMP    TO) N0890


N820  [02  #080 = #081 - @.....#072
N821  [51  (JUMP POS TO) N0840  ;if programmed S is > 2. gear stage
N822  [03  #080 = #081 * @+...0,255
```

```
N823 [04  #080 = #080 / @.....#072
N824 [00  #081 =      @+...0,002
N825 [53  (JUMP    TO) N0890


N840  02  #080 = #081 - @.....#073
N841 [51  (JUMP POS TO) N0860                ;if programmed S is > 3. gear stage
N842 [03  #080 = #081 * @+...0,255
N843 [04  #080 = #080 / @.....#073
N844 [00  #081 =      @+...0,003
N845 [53  (JUMP    TO) N0890


N860 [02  #080 = #081 - @.....#074
N861 [52  (JUMP NEG TO) N0863
N862 [00  #081 =      @.....#074
N863 [03  #080 = #081 * @+...0,255
N864 [04  #080 = #080 / @.....#074
N865 [00  #081 =      @+...0,004


N890 [01  #079 = #080 + @+...9,000           ;giving M9xxx
N891 G22  N0100  W....   CALL PROGRAM        ;change S-gear
N892 G13  M #079                             ;output M9xxx 0-10V
N893 [53  (JUMP    TO) N0030                 ;process next function
```
;What is active tool?
The routine starting with N900 looks what tool is active at the moment and returns the active tool number in #080. In #081, the next available toolnumber is returned. When input 5 and 6 of IO1 are both active (=connected to 24V), then #081 can contain the values 0,001 to 0,004, else 0,001 to 0,006.
```
N900 [00  #080 =      @+...0,000
N901 [00  #081 =      @+...0,001
N910 G23  N0914  W....  M0171                ;jump to N914 if tool #1 not active
N911 [00  #080 = @+...0,001
N912 [00  #081 =      @+...0,002
N913 [53  (JUMP    TO) N0950
N914 G23  N0918  W....  M0172                ;jump to N918 if tool #2 not active
N915 [00  #080 =      @+...0,002
N916 [00  #081 =      @+...0,003
N917 [53  (JUMP    TO) N0950
N918 G23  N0922  W....  M0173                ;jump to N922 if tool #3 not active
N919 [00  #080 =      @+...0,003
N920 [00  #081 =      @+...0,004
N921 [53  (JUMP    TO) N0950
N922 G23  N0930  W....  M0174                ;jump to N930 if tool #4 not active
N923 [00  #080 =      @+...0,004
N924 [00  #081 =      @+...0,005
N925 [00  #081 =      @+...0,001
N926 G22  N0960  W....   CALL PROGRAM
N927 [50  (JUMP ZER TO) N0950
N928 [00  #081 =      @+...0,005
N929 [53  (JUMP    TO) N0950
N930 G22  N0960  W....   CALL PROGRAM
N931 [50  (JUMP ZER TO) N0990
N940 G23  N0944  W....  M0175                ;jump to N944 if tool #5 not active
N941 [00  #080 =      @+...0,005
```

N942 [00 #081 =   @+...0,006
N943 [53 (JUMP TO) N0950
N944 G23 N0950 W.... M0176   ;jump to N950 if tool #6 not active
N945 [00 #080 =   @+...0,006
N946 [00 #081 =   @+...0,001
N950 G11 T#080 M....
N953 [53 (JUMP TO) N0990
N960 [84 #016 #001 #088 #064 #001
N961 [00 #089 =   @+...0,048
N962 [18 #088 = AND   #089
N990 [80 END

8.0.5   P9974 Home Position

P9974 is called when [key icon] in the MANUAL MODE is pushed. P9974 is programmed so that a
Home Position can be memorized and moved to from any point.
N001 [80 xx set HOME POSITION
N002 [80 xx move HOME POSITION
;Display text N001 and N002 on the screen
N050 [00   #000 =   @+...1,254
N051 [83   #001
N053 [00   #043 =   @+..49,372
N054 [00   #044 =   @+...0,027
N055 [82   #011 #000
N060 [00   #000 =   @+...7,910
N061 [83   #002
N063 [00   #043 =   @+..56,028
N064 [00   #044 =   @+...0,086
N065 [82   #011 #0000
;wait for a key to be pushed
N102 [89   #080
N103 [50   (JUMP ZER TO) N0102
N104 [02   #081 = #080 - @+...0,030
N105 [50   (JUMP ZER TO) N0700
N106 [02   #081 = #080 - @+...0,015
N107 [54   (JUMP NZ TO) N0900
;key „INPUT" was pushed
N108 [96   #005 #082 #001
N120 [00   #081 =   @+..29,184
N123 [84   #000 #002 #082 #081 #002 #000 #000
N124 [00   #081 =   @+..29,188
N125 [84   #000 #002 #083 #081 #002 #000 #000
N130 [53   (JUMP   TO) N0900
;Key „REFERENCE POINT" was pushed
N700 G90 ABSOLUTE INPUT
N710 [00   #085 =   @+..29,184
N711 [84   #000 #001 #086 #085 #002
N713 [01   #086 = #086 + @.....#086
N715 G00 X.....#086 Z.....#087 RAPID TRAVERSE
N716 [53   (JUMP   TO) N0900
;wait for all axes to have stopped
N900 G13 M0019
;switch back to MANUAL MODE
N901 [00   #040 =   @+...0,027

N902 [00   #044 =   @+...0,025
N903 [82   #000

8.0.6   P9999 Autostart

P9999 is executed when the CNC is switched on.
;#102, #103 and #104 contain the Actual Value of X, Y, and Z before the CNC was switched off.

P9999 AUTOSTART
N004 G92 X.....#102 Y.....#103 Z.....#104   ;SET Actual Value
;P9936 N900 looks what tool is active and and puts it in the TOOL register.
N005 G20 P9936 N0900   ;JUMP PROGRAM P9936 bloc N900

8.0.7   P0074 REFERENCE POINT FOR GANTRY AXES

We supose that U is a gantry axis to Y. So we program P0 N906 B8024 which couples U to Y.
The following program must be in memory:
P0074 REFERENCE POINT FOR GANTRY AXES
N001 G13 M8024
N002 G13 M2744   ; N005 is interupted in case the U axis reaches the limit switch first!
N005 G74 Y-1   ; Y and U start moving to the limit switch
; Here we have 2 posibilities: Y reaches the limit switch first, then N005 is terminated normally.
;Or U reaches the limitswitch first, then N005 is aborted with U-axis staying on the limitswitch.

N006 G74 U-1   ;Move U to the reference point
N007 G74 Y-1   ;Move Y to the reference point

9   CONNECTORS

A CNC is not necessaryly equipped with all the connectors mentioned here. The number of connectors depends upon the purchased CNC and options. All connectors mentioned are SUB-D connectors.

Attention:

• **When connecting the CNC, the national security requirements must be fullfilled.**
• **Especially, the CNC must be switched off in an emergency situation.**
• **The power supply must be 230V +/- 5%.**
• **The CNC - housing must not be totally closed. Air circulation must be available.**
• **All connections to the CNC must be shielded.**
  The shield must be firmly connected to the metallic case of the connector.
  This prescription applies to ALL connections to the CNC!
  Else noise can be injected into the CNC causing spontaneous resets and similar errors!
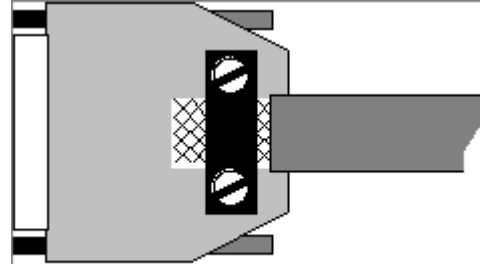
9.0.1   X1 INPUT

Pin	Signal	Pin	Signal	M-Function
1	IN 1.1 +	20	IN 1.1 -	M161
2	IN 1.2 +	21	IN 1.2 -	M162
3	IN 1.3 +	22	IN 1.3 -	M163
4	IN 1.4 +	23	IN 1.4 -	M164
5	IN 1.5 +	24	IN 1.5 -	M165
6	IN 1.6 +	25	IN 1.6 -	M166
7	IN 1.7 +	26	IN 1.7 -	M167 (reserved Ext.Start)
8	IN 1.8 +	27	IN 1.8 -	M168 (reserved EXT Inter.)
9	IN 2.1 +	28	IN 2.1 -	M261
10	IN 2.2 +	29	IN 2.2 -	M262
11	IN 2.3 +	30	IN 2.3 -	M263
12	IN 2.4 +	31	IN 2.4 -	M264
13	IN 2.5 +	32	IN 2.5 -	M265
14	IN 2.6 +	33	IN 2.6 -	M266
15	IN 2.7 +	34	IN 2.7 -	M267
16	IN 2.8 +	35	IN 2.8 -	M268
17		36		
18		37		
19				

All inputs are optocoupled.
They are activated by a signal of 24V, +/- 10%, 5mA.
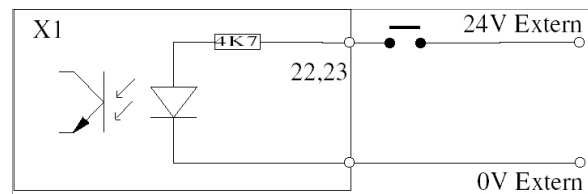
9.0.2   X2 OUTPUT

Pin	Signal	Pin	Signal	M-Function
1	OUT 1.1 +	20	OUT 1.1 -	M141
2	OUT 1.2 +	21	OUT 1.2 -	M142
3	OUT 1.3 +	22	OUT 1.3 -	M143
4	OUT 1.4 +	23	OUT 1.4 -	M144
5	OUT 1.5 +	24	OUT 1.5 -	M145

6	OUT 1.6 +	25	OUT 1.6 -	M146
7	OUT 1.7 +	26	OUT 1.7 -	M147
8	OUT 1.8 +	27	OUT 1.8 -	M148 (Lubrification Pulse page 74)
9	OUT 2.1 +	28	OUT 2.1 -	M241
10	OUT 2.2 +	29	OUT 2.2 -	M242
11	OUT 2.3 +	30	OUT 2.3 -	M243
12	OUT 2.4 +	31	OUT 2.4 -	M244
13	M03 +	32	M03 -	M3
14	M04 +	33	M04 -	M4
15	M05 +	34	M05 -	M5
16	M08 +	35	M08 -	M8
17	M10 +	36	M10 -	M10
18	SPEED +	37	SPEED -	S-Function (page 40)
19				

All outputs are optocoupled.
They can source 0,5A max.
A clamp diode (1N4001) should be connected in parallel to the load.
The overall capacity is 2A.
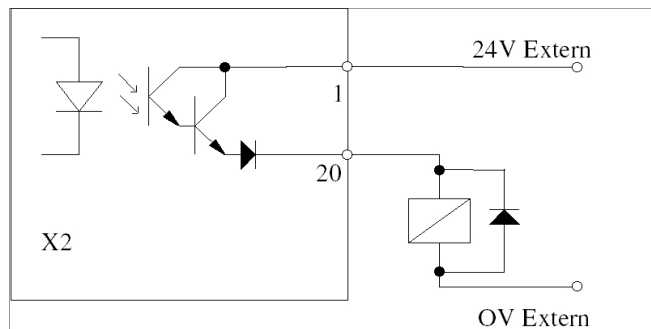
9.0.3   X3 M-FUNCTION for C88

Pin	Signal	Pin	Signal
1	Spindle CW M03+	9	Spindle CW M03-
2	Cooling M08+	10	Cooling M08-
3	Spindle CCW M04+	11	Spindle CCW M04-
4	Clamping M10+	12	Clamping M10-
5	Spindel Speed+ (Option)	13	Spindel Speed- (Option)
6	Input M16+	14	Input M16-
7		15	Code Pin

Outputs:
SPINDLE CW: Switched on with M03, off with M05.
SPINDLE CCW: Switched on with M04, off with M05
COOLING: Switched on with M08, switched off with M09.
CLAMPING: Switched on with M10, switched off with M11.
Spindle Speed: Analog output 0-10V, corresponding to the programmed Spindle speed S.
Inputs:
INPUT 1:   Is read in with M16.
The outputs can switch 20mA at 24 V, the inputs need 24V, 5mA..
All inputs and outputs are optocoupled.

9.0.4   X4 EXTERNAL SYNCHRONISATION FOR G33

1	+5V Internal	6	B*, Ua2-
2	0V	7	C, Ua0+, Refpulse
3	A, Ua1+	8	C*, Ua0-, Refpulse*
4	B, Ua2+	9	
5	A*, Ua1-		

Here the encoder of the spindle is to be connected. N925X must be adjusted, see page 76.

9.0.5   X6 COM 1

Pin	Signal		Pin	Signal
1	GND		6	+12V
2	RxD (Input)		7	RTS (Output)
3	TxD (Output)		8	CTS (Input)
4	External Feedrate Potentiometer		9	External Handwheel
5	GND			

Output RTS = „L" = -12V: The peripheral unit must stop sending data.

Input CTS = „L" = -12V: Transmissing of data via TxD is stopped.

The baudrate can be programmed with N901Y. The dataformat is 8 databits, 1 startbit, 1 stopbit, no parity.

9.0.6   X7 COM 2

Connection for Handbox

9.0.7   X8   INPUT/OUTPUT

Pin	Signal	Pin	Signal
1	Output 1 (white)	14	Input 1 (white)
2	Output 2 (yellow)	15	Input 2 (yellow)
3	Output 3 (green)	16	Input 3 (green)
4	Output 4 (brown)	17	Input 4 (brown)
5	Output 5 (grey)	18	Input 5 (grey)
6	Output 6 (black)	19	Input 6 (black)
7	Output 7 (rose)	20	Input 7 (rose)
8	Output 8 (violet)	21	Input 8 (violet)
9	EXTERNAL 24V	22	EXTERNAL 0V
10	EXTERNAL 24V	23	EXTERNAL 0V
11		24	
12	Code Pin	25	
13			
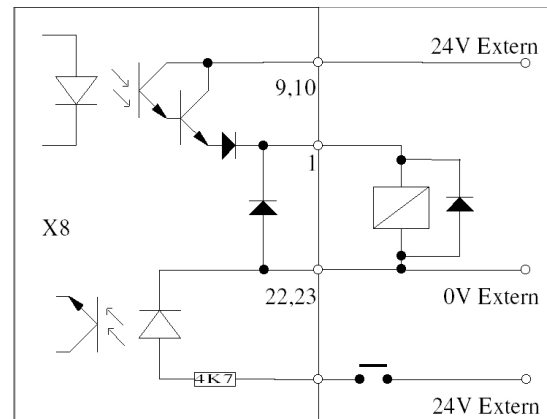
All 8 inputs and outputs are optocoupled.

The outputs can source 0,5A max.

A clamp diode (1N4001) should be connected in parallel to the relay.

The overall capacity of one I/O card is 2A.

The inputs are activated by a signal of 24V, +/- 10%, 5mA.

The CNC can manage up to 8 I/O cards. The address of the I/O card can be set on the board.

The card I/O4 has the following predifined inputs and outputs:

Output 1: Clamping X	Input 1: External Interrupt *
Output 2: Clamping Y	Input 2: External Stop *
Output 3: Positioning =1/ bloc end =0 *	Input 3: External Start *
Output 4: Brake X	Input 4: Keyboard off, only External Start is active.
Output 5: Brake Y	Input 5: Wait for pressure down Y
Output 6: Pulse when Position Reached	Input 6: Wait for pressure down X
Output 7: Programm running =1, finished =0 *	Input 7: Wait until pressure X present.
	Input 8: Wait until pressure Y present.

* These signals are generally available.The other signals are active only when P0 N902 X04 is activated. Additionally N902 X08 can be programmed, which selects automatically Single Step: the CNC waits for a START before each movement.

Input1 must not be used as emergency input without switching off the main power supply.

I/O card 8:
Output 8:   Manual Mode is active.
Input 8:   Manual Mode is inhibited.

9.0.8   X9 Limit Switches

Pin	Signal	Pin	Signal
1	Negative Direction U +	14	Negative Direction U -
2	Negative Direction Z +	15	Negative Direction Z -
3	Negative Direction Y +	16	Negative Direction Y -
4	Negative Direction X +	17	Negative Direction X -
5	Positive Direction U +	18	Positive Direction U -
6	Positive Direction Z +	19	Positive Direction Z -
7	Positive Direction Y +	20	Positive Direction Y -
8	Positive Direction X +	21	Positive Direction X -
9		22	
10		23	
11		24	
12		25	
13			

The inputs need 24V, 5mA and are optocoupled. The machinedatum N790 (page 66) defines for each axis separately if the limitswitches used are Opener or Closer and if 1 or 2 limitswitches are connected.
If only 1 Limit Switch is present for example for the Positive X direction, the input for the Negative X should be connected to +24V
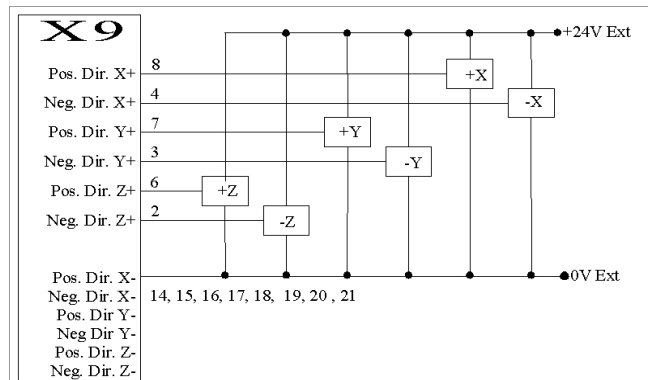
If the CNC should fail, no danger causing movement is allowed to result. Therefore, Safety Limit Switches generating an emergency stop with disable the amplifiers and switch off the CNC should be located after the normal Limit Switches.
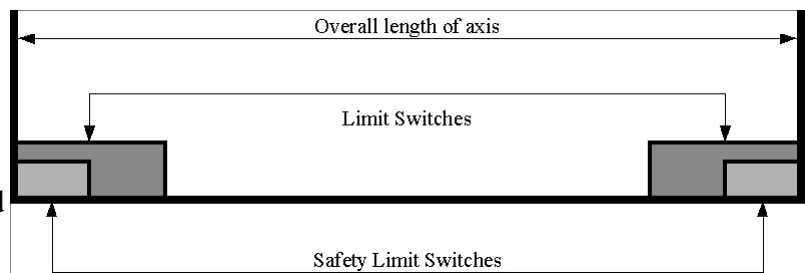
In Turning Mode, the Z limitswitches must be connected to the Y inputs!

9.0.9   X11 SERVO / ENCODER

Pin	Signal	Pin	Signal
1	+5V INTERN	9	MOTOR ENABLE +
2	0V INTERN	10	MOTOR ENABLE -
3	A, Ua1+	11	DC+
4	B, Ua2+	12	DC-
5	A*, Ua1-	13	
6	B*, Ua2-	14	
7	C, Ua0+	15	Code pin
8	C*, Ua0-		

For each axis exists a connector X11. They are named X11-1 for X, X11-2 for Y (or Z in Turning mode), X11-3 for Z, X11-4 for U.

The ouput Motor Enable is optocoupled and can switch 24V, 20mA.. In addition, a relay activating a motorbrake can be connected.

The servoamplifier connected to DC must have a **differential input** +/- 10V.

The connection from the CNC to the amplifier should be shielded and grounded on one side to the CNC. The other side should be left open.

The maximim input frequency for the encoders is 150KHz

Hints for selection of machine data for servo mode:
- Switch N790 (page 66) to servo mode (add the value 16).
- Activate N813X (page 68).The axis now can now be moved at low speed if connector X11 is correctly cabled.
- Input N700, N706, N707 (page 64) correctly for each axis.
- By pushing the key „2" in the MANUAL MODE, the Lag Distance for each axis is displayed.

9.0.10 X21 Serial Interface

Pin	Signal	Pin	Signal
1	GND	6	+12V (Output)
2	TxD (Output)	7	GND
3	RxD (Input)	8	External Feedrate Potentiometer
4	RTS (Output)	9	External Handwheel
5	CTS (Intput)		

Connection of X21 to a PC connector with 25 pins or 9 pins:

X21		PC-25 pin	PC-9 pin
1	GND	7	5
2	TxD (O)	3	2
3	RxD (I)	2	3
4	RTS (O)	5	8
5	CTS (I)	4	7
8	Code pin		

Output RTS = „L" = -12V: The peripheral unit must stop sending data.

Input CTS = „L" = -12V: Transmissing of data via TxD is stopped.

The baudrate can be programmed with N901Y. The dataformat is 8 databits, 1 startbit, 1 stopbit, no parity.

9.0.11 X23 CNC Signal

Pin	Signal	Pin	Signal
7	S-Signal +	20	S-Signal -

Between S-Signal + and S-Signal - a voltage of 0-10V is generated depending on the programmed spindlespeed S.

9.0.12 X26 VIDEO TTL

Pin	Signal	Pin	Signal
1	INTERNAL GND	6	+12V, 0,5A
2	VIDEO	7	VIDEO *
3	HOR. SYNC *	8	HOR. SYNC
4	VERT. SYNC *	9	VERT. SYNC
5	BAS COMPOSITE VIDEO		
*	Signals are negativ switching		

9.0.13 X29 TRANSDUCER (Measuring System)

CNC			Heidenhain encoder		CNC
Pin	Signal		Pin	Color	
1	+5V	(red)	2,12	blue, brown/yello	CLOCK X
2	0V	(grey)	10,11	White, white/yell	14
3	Ua1, A	(brown)	5	brown	74 LS 04 TTL
4	Ua2, B	(violet)	8	grey	
5	Ua1*, A*	(white)	6	green	RIGHT/LEFT X
6	Ua2*, B*	(blue)	1	rose	2
7	Ua0, C,	(green)	3	red	
8	Ua0*, C*,	(yellow)	4	black	GND
9	Code Pin				1
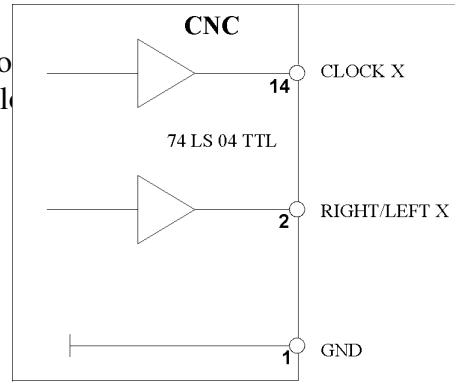
Within the CNC the inputs A - A*, B - B*, C - C* are connected to the inputs of an optocoupler. If the transducers are powered externally, they are completely galvanically isolated.

The pins 1 and 2 (+5V und 0V) MUST NOT be connected to supply power to the encodersimulation of a servoamplifier. The encodersimulation must always be powered internally from the servoamplifer in order to insure the galvanic isolation.

The maximim input frequency for the encoders is 600KHz

- Use shielded cable. Connect the shield to the case of the CNC.
- Use transducer with TTL output!
- With a 12V transducer, additional external resistors of 220 Ohm must be connected in series with A, B, C.
- All connected transducers together can use a maximum of 1A from the internal 5V line.
- To invert the counting direction of the measuring system, exchange A with B and A* with B*.

9.0.14 X31 Handbox (Option)

Pin	Signal		Pin	Signal	
1	Tas 0	(1)	14	Tas 1	(2)
2	Tas 2	(3)	15	Tas 3	(4)
3	Tas 4	(5)	16	Tas 5	(6)
4	Tas 6	(7)	17	Tas 7	(8)
5	Strobe 0	(9)	18	Strobe 1	(10)
6	Strobe 2	(11)	19	Strobe 3	(12)
7	Strobe 4	(13)	20	Strobe 5	(14)
8	Strobe 6	(15)	21	Strobe 7	(16)
9	* Notaus+	(17)	22	* Notaus-	(18)
10	* Shield (19)	(20)	23	(internal Handwheel CNC)	(20)
11	Feedrate Handbox	(21)	24	(internal Feedrate CNC)	(22)
12	+5V	(23)	25	Handwheel	(24)
13	GND	(25)			

* Pin 9, 10 und 22 are not connected internally in the CNC!!!! Numbers in braquets refer to the 26-pin Latch-connector going to the keyboard inside the CNC and to which this connector is in parallel. The cables NOTAUS+ and - must be connected to a **clean** 24V-power supply in order not to interfere with the keyboard cables.

9.0.15 X84 SM SIGNAL

| Pin	Signal	Pin	Signal
1	0V Internal	14	Clock X
2	Right/Left X	15	Clock Y
3	Right/Left Y	16	Clock Z
4	Right/Left Z	17	Clock U
5	Right/Left U	18	Clock V
6	Right/Left V	19	Clock A
7	Right/Left A	20	Clock B
8	Right/Left B	21	Clock C
9	Right/Left C	22	Code Pin
10		23	
11		24	
12		25	
13			

- All signal outputs are TTL 5V with positive polarity.
- It is recommended to connect only stepping motor drivers with optocoupled inputs.
- Shielded cable must be used. Over X84 Pin 1 external noise can be injected into the CNC resulting in unpredictable behaviour.

9.0.16   X85 Limit Switches

Pin	Signal	Pin	Signal
1	Internal use (Out IPO0+)	14	Internal use (Out IPO0-)
2	Positive Direction X	15	Positive Direction Y
3	Positive Direction Z	16	Positive Direction U
4	Positive Direction V	17	Positive Direction A
5	Positive Direction B	18	Positive Direction C
6	Negative Direction X	19	Negative Direction Y
7	Negative Direction Z	20	Negative Direction U
8	Negative Direction V	21	Negative Direction A
9	Negative Direction B	22	Internal Use
10	0V External	23	0V External
11	Code Pin	24	Internal Use
12	Internal Use	25	Internal Use
13	Internal Use		

The inputs need 24V, 5mA and are optocoupled. The machinedatum N790 defines for each axis separately if the limitswitches used are Opener or Closer and if 1 or 2 limitswitches are connected. If only 1 Limit Switch is present for example for the **Positive Direction X**, the input for the **Negative Direction X** should be connected to +24V
If the CNC should fail, no danger causing movement is allowed to result. Therefore, Safety Limit Switches generating an emergency stop which disable the amplifiers and switch off the CNC should be located after the normal Limit Switches.
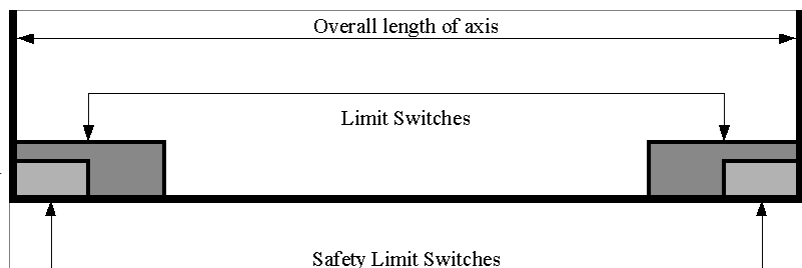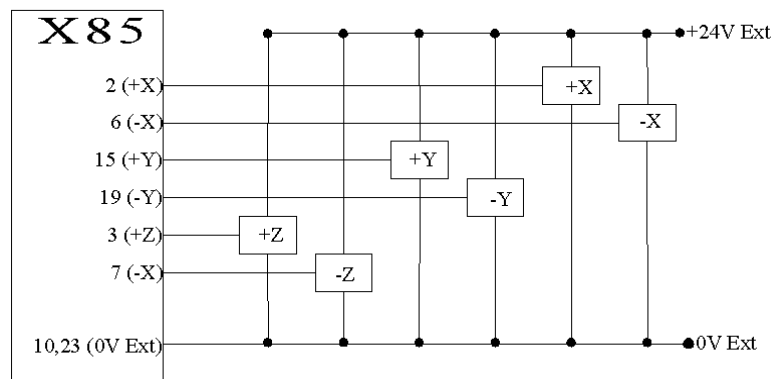
Hint:
On a D-Vsersion, the limitswitches Z must be connected to the Y inputs!

9.0.17   X85 on CNC for steppingmotors with Referencepulse (Option)

For increasing the accuracy of the limitswitches when used as reference position with G74, we offer on special request additional inputs on the CNC's DRSM and DRSM-MINI which must be connected as follows on the connector X85:

Pin	Signal	Pin	Signal
2	Positive Direction X	15	Positive Direction Y
3	Positive Direction Z	16	Positive Direction U
4	**Reference Pulse X** (Positive Direction V)	17	**Reference Pulse Y** (Positive Direction A)
5	**Reference Pulse Z** (Positive Direction B)	18	
6	Negative Direction X	19	Negative Direction Y
7	Negative Direction Z	20	Negative Direction U
8	**Reference Pulse X** (Negative Direction V)	21	**Reference Pulse Y** (Negative Direction A)
9	**Reference Pulse Z** (Negative Direction B)	22	Internal Use

The inputs **Positive Direction** and **Negative Direction** must have 24V and must be Normally Closed, that means when not activated they must bring 24V. The inputs **Reference Pulse** must go shortly to 5V when the reference pulse comes. Under this conditions the G74 is executed as follows: The axis moves to the limitswitch, stops down, co0mes to stop maybe after the limit switch, inverts the moving direction, moves back to the limit switch, and from this point on, THE NEXT REFERENCEPULSE IS RECOGNIZED and the CNC stops.

9.0.18   X85 Handwheel (Option)

The following pins of X85 are used to connect an optional handwheel

11	Code Pin	24	+5V
12	Ub+(B)	25	0V
13	Ua+ (A)		

9.0.19   X86 EXT SYNC (Option)

1	+5V Internal	6	C, Ua0, Refpulse
2	0V	7	
3	A, Ua1	8	
4	B, Ua2	9	
5	Code Pin		

A transducer can be connected to A, B, C in order to synchronize G33 to a spindle (EXT SYNC). The input X85 Pin 22 (-C) must not be connected and N790C must not be programmed with 512 (Limit switch- present). With N925X, the number of pulses per rotation of the spindle can be programmed. The maximal input frequency is 60 Khz, so if a motor with 3000 rot/min must be synchronized, the encoder can have a maximum of 1200 pulses/rot. Softwareversions produced after 1.10.2002 are able to count up to 255 Khz, giving the possibilty to use encoders with 5000 pulses per rotation can be used for better accuracy.

9.0.20   X86A EXT SYNC (Option)

Function like X86, however Pin Connection like X29 Standard.

9.0.21   X87 JOYSTICK (Option)

1	+5V Internal	6	

|				
2	0V		7	Joystick X
3			8	Joystick Y
4			9	Joystick Z
5	Code Pin			

In restposition of the joystick, a voltage of 2,5V must be applied to pin 7,8,9. With +/- 2V the feedrate in positive/negative direction can be influenced.

9.0.22   X88 ANALOG CARD   (Option)

Pin	Signal	Pin	Signal
1	AD DC 1+ (Input)	14	AD DC 1- (Input)
2	AD DC 2+ (Input)	15	AD DC 2- (Input)
3	AD DC 3+ (Input)	16	AD DC 3- (Input)
4	AD DC 4+ (Input)	17	AD DC 4- (Input)
5	Enable 1+	18	Enable 1-
6	Enable 2+	19	Enable 2-
7	Enable 3+	20	Enable 3-
8	DA DC 1+ (Output)	21	DA DC 1- (Output)
9	DA DC 2+ (Output)	22	DA DC 2- (Output)
10	DA DC 3+ (Output)	23	DA DC 3- (Output)
11	DA DC 4+ (Output)	24	DA DC 4- (Output)
12	Enable 4+	25	
13	Enable 4-		

9.0.23   X90A DC-OUT

Pin	Signal	Pin	Signal
1		14	Out1-
2	DC1+	15	DC1-
3	DC2+	16	DC2-
4	DC3+	17	DC3-
5	DC4+	18	DC4-
6	Out0+	19	Inp0+
7		20	Code Pin
8	Motor 1 Enable+	21	Motor 2 Enable+
9	Motor 3 Enable+	22	Motor 4 Enable+
10	Motor 1 Enable-	23	Motor 2 Enable-
11	Motor 3 Enable-	24	Motor 4 Enable-
12	Out0-	25	Inp0-
13	Out1+		
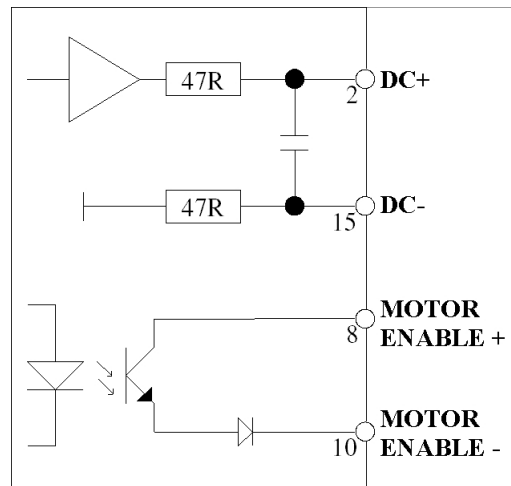
The ouputs Motor Enable are optocoupled and can switch 24V, 20mA.. They allow in addition to connect a relay activating a motorbrake.

The servoamplifier connected to DC must have a **differential input** +/- 10V.

The connection from the CNC to the amplifier should be shielded and grounded on one side to the CNC. The other side should be left open.

Hints for selection of machine data for servo mode:
• Switch N790 to servo mode (add the value 16).
• Activate N813X f.e. 7 = 1 + 2 + 4 = axes X,Y,Z.The axes now can now be moved at low speed if connector X90 and X29 are correctly cabled.
• Input N700, N706, N707 correctly for each axis.
• By pushing the key „2" in the MANUAL MODE, the Lag Distance for each axis is displayed.

9.0.24   Resolver connector for Motors EBLx

1		6	Thermoswitch-
2	Thermoswitch+	7	Cosine-
3	Cosine+	8	Sine-
4	Sine+	9	Supply-
5	Supply+		

9.0.25   Motor connector for Motors EBLx:

1 = U
4 = V
3 = W
2 = Case (ground)
A =
B =
C = *(Brake +)
D = *(Brake -)
1 =Shielding

9.0.26   STEPPING MOTOR 5 phases

1	X1*	(yellow)	2	X1	(white)
3	X2*	(blue)	4	X2	(red)
5	X3*	(orange)	6	X3	(green)
7	X4*	(grey)	8	X4	(black)
9	X5*	(brown)	10	X5	(violet)
11	GND		12	BRAKE+	
13	BRAKE-				

Reversing of the moving direction: Interchange pin 1 and 10, 2 and 9, 3 and 8, 4 and 7, 5 and 6.

9.0.27   STEPPING MOTOR 2 phases

1	A	yellow/green
2	A*	grey/blue
3	B	white/brown
4	B*	orange/red
5	GROUND	

9.0.28   STEPPING MOTOR 3 phases

1 U
2 V
3 W
Case Ground, Shield

9.0.29   POWER SUPPLY 380V

1	L1
2	L2
3	L3
4	N
CASE	GROUND

9.0.30   DC-Servomotor

1	MOTOR +
2	MOTOR -
3	TACHO +
4	TACHO -
5	BRAKE + (Option)
6	BRAKE -

CASE GROUND

9.0.31   AC-Servomotor

1	U
2	V
3	W
4	CABLESHIELD
5	BRAKE +
6	BRAKE -

CASE   GROUND

Motors allways should be connected with a SHIELDED cable. The shield should be connected only on one side to the ground of the amplifier.

9.0.32   I/O CONNECTION

a1 OUTPUT 1	b2 EMERGENCY+ (OPTION)	c1 INPUT 1
a2 OUTPUT 2	b3 EMERGENCY- (OPTION)	c2 INPUT 2
a3 OUTPUT 3	b4 BRAKE + (OPTION)	c3 INPUT 3
a4 OUTPUT 4	b5 BRAKE - (OPTION)	c4 INPUT 4
a5 OUTPUT 5		c5 INPUT 5
a6 OUTPUT 6		c6 INPUT 6
a7 OUTPUT 7		c7 INPUT 7
a8 OUTPUT 8		c8 INPUT 8
a9 24V EXTERN		c9 0V EXTERN

KEYWORDS