POSITIONING / MOTION CONTROL UNIT
ADDIPOS APCI-8001

G-code interface

Rev. 1/052005

1 Version information

This specification is valid for

McuWIN.EXE	from version 2.5.3.32
IniCfg.EXE	from version 2.5.3.15
MCFG.EXE	from version 2.5.3.40
MCUG3.DLL	from version 2.5.3.24
NCC.EXE	from version 2.5.3.13
RWMOS.ELF	from version 2.5.3.35

1.1 Changes in version 2.5.3.32

- S-command in same line as G-codes sometimes executed incorrectly
  Flow/requirements for S-command substantially changed (see doc.).

1.2 Changes in earlier versions

Version 2.5.3.31:
- Various changes in Editor display
- Main spindle switched off on program stop
- NoTriangle option allowed with LookAhead

Version 2.5.3.30:
- Case-insensitive mode now supported, configurable in IniCfg
- Program stop also possible in referencing and in single-step mode
- Position and time units for interpolation commands now selected in IniCfg
- New M150 command
- Cleared bug in G-code programs with no M30 at the end
- Module name no longer required in programs

Version 2.5.3.25:
When comments were used, translation failures sometimes occurred, particularly with comments in quotes in lines with G01, G02 or G03. It is now also possible to nest comments with round brackets.

Version 2.5.3.24:
- A velocity factor can be entered for freeing motions
  (FreeingVelFactor)
- Application-specific reference run can now be set in IniCfg.
- Configuration variable ReferenceSwitchLatch to disable check on reference switch latch
- Configuration variable NoRefToLimitSwitch to allow referencing only on the index track
- A customer-specific logo can now be incorporated into the program interface
- The need for line-numbers can be switched off in IniCfg

Version 2.5.3.23:
- New commands:
  - G153: Read zero offset
  - G154: Reprogram zero offset
  - Zero offset, particularly G54…G59, revised,
    G59 no longer exists, as there are only 5 zero offset registers
- Fixed bug in the use of non-consecutive interpolation axes;
- Program can be started in simulation mode;
- Counter monitoring by index latch and encoder error monitoring now selectable independently;
- Fixed bug in override setting at start of reference travel;

Version 2.5.3.19:
New commands G98, G99, G161, G162.
Commands G94 and G70/G71 were still sometimes handled incorrectly. With the updated version, there may be changes in velocity and acceleration. After an update, this must be investigated and corrected if necessary. (25.06.2004)

Version 2.5.3.18:
From this version onwards, it is possible to enter relative center coordinates even in absolute mode (G90).

Version 2.5.3.16:
It is possible to enter a LookAhead depth in IniCfg. Once the number of corresponding interpolation records has been programmed (G01, G02, G03), processing begins automatically.
In single-step mode, processing also starts automatically. It is no longer necessary to click within the editor screen to start the axes. Splines are only projected as straight-line segments.

Version 2.5.3.15:
Commands G21 / G22 / G23 and G24 for axis reflections are new. Also the use of positioning factors with G39.

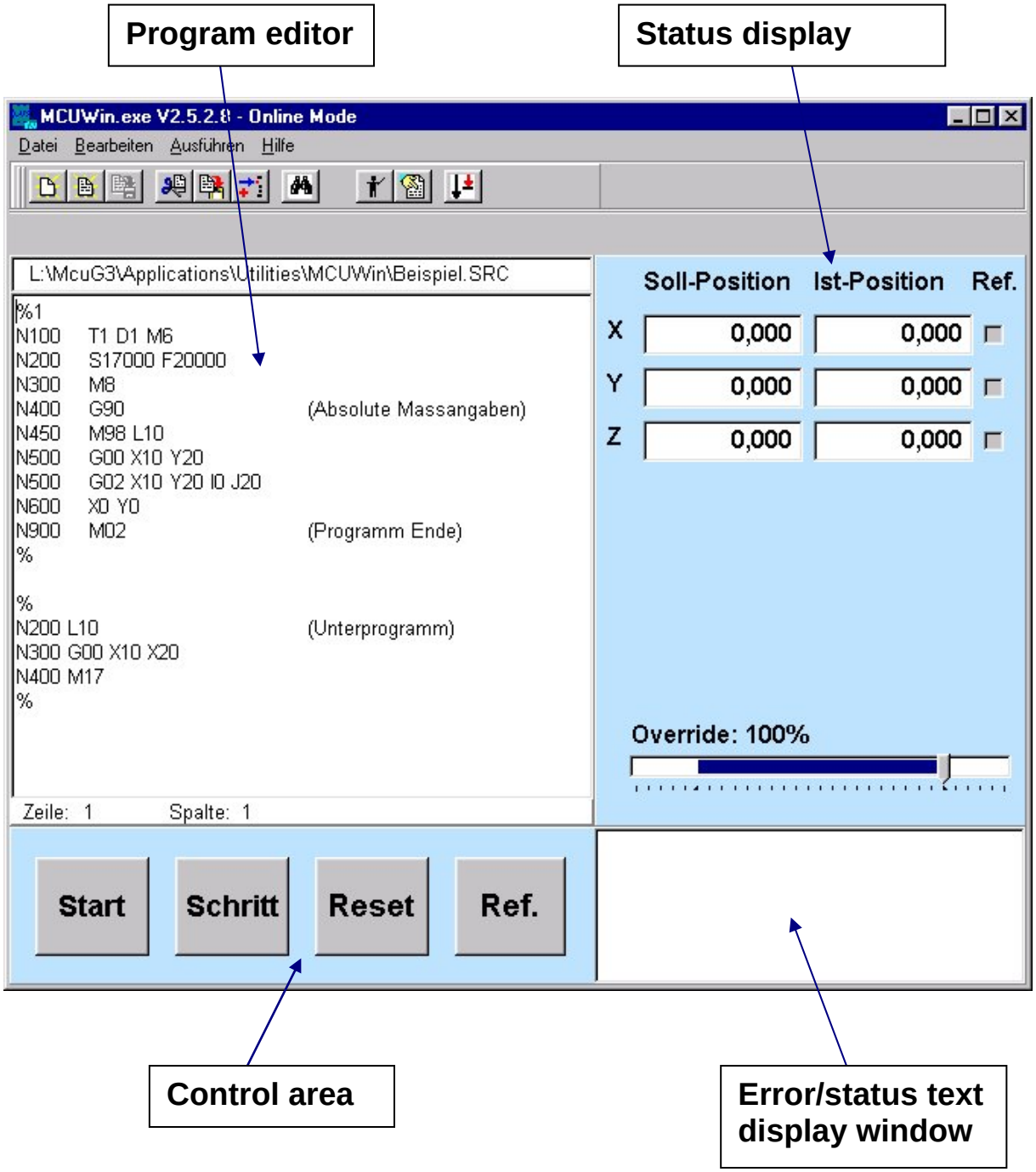
Version 2.5.3.14:
Command G54 now selects a zero offset from a zero offset table, as do G55 to G59. It is now possible to set the zero offset to any value with G92. The existing function of G54 has therefore been superseded by G92.

2 Procedure for processing G-code program files

2.1 The McuWIN user interface

The Windows application program McuWIN.exe provides the user with a user interface to run G-code programs. This interface can be used to edit programs, check their syntax and execute them.
It also displays target and actual positions, reference status and error messages. The required initialization steps for the controller are automatically executed by this program. Programs running in one-step mode can also be processed.
Additional information on processing G-code programs can be found in the file “McuWin Einrichten.PDF”. This will be helpful if McuWIN is not to be used because an in-house user interface is to be deployed or enhancements are needed.

Figure: Screenshot of McuWIN user interface



3 Functionality of McuWIN

3.1 Emergency power-off monitoring

When the axes are set up with the configuration program mcfg, digital inputs can be configured as emergency power-off inputs. If an EPO input is activated, any programs that are running will be interrupted, referencing may be cancelled depending on the configuration, the control circuits will be opened and the EPO status will be displayed.
When the EPO condition has been cleared, the control circuits will be closed again. The reference run, or a G-code program, can now be restarted. This allows controlled operation to be resumed after the emergency power-off.
Warning: The EPO signal **must** act directly on the drives (DC link voltage). The relevant security instructions must be observed.

3.2 Maximum lag error

When the axes are set up with the configuration program mcfg, an axis-specific “maximum lag error” can be entered. If this is exceeded with the control circuit closed, the program will be terminated and the cause of the error displayed.

3.3 Hardware limit switches

When the axes are set up with the configuration program mcfg, digital inputs can be configured as hardware limit switches. When a limit switch is triggered, the program will be terminated and the cause of the error displayed. Monitoring of the hardware limit switch via McuWIN only becomes active after a successful reference run. However, the limit switches will always act on positioning profiles with the specified option.

3.4 Software limit switches

When the axes are set up with the configuration program mcfg, digital inputs can be configured as software limit switches. When a limit switch is triggered, the program will be terminated and the cause of the error displayed. Monitoring of the software limit switch via McuWIN only becomes active after a successful reference run. The limit switches only act on positioning profiles with the specified option after referencing.

3.5 Monitoring of the encoder error flag

This functionality enables errors in positioning with incremental encoder systems to be detected early. This monitoring can be enabled at an axis-specific level. If a problem is detected, a warning will be output to the user interface.

3.6 Monitoring of position counters

This monitoring can also be enabled at an axis-specific level. The position is verified on every zero pulse, giving a reliable indication of when the system is miscounting.
This option can only be used where:

- Monitoring of the encoder flag is enabled;
- Incremental encoder systems are used;
- The denominator unit for the system value slsp is set with deg;
- Servo axes (not steppers) are used

If a problem is detected, a warning will be output to the user interface.

3.7 Tool radius correction

Tool radius correction supports commands G01, G02 and G03. Circuits must be located in the respective primary plane.

4 Description of G-codes implemented

4.1 G00 – Rapid positioning

The target positions can be given as absolute or relative coordinates, depending on the mode selected (G90 / G91). Velocity and acceleration are the axis-specific default values entered in MCFG, or the values set in the initialization task.
This command is persistent, i.e. G00 does not need to be included in every subsequent line. After continuation lines, the command recognized as persistent is the last command in the source code preceding the relevant line, not the command last executed.

Example:
N0400 G00 X20 Y22
N0410 X30 Y23 Z-5

4.2 G01 – Linear interpolation

Execution of a linear interpolation with all positioning axes: the target positions can be given as absolute or relative coordinates, depending on the mode selected. The track speed can be defined e.g. with the F command. The acceleration is programmed in the initialization routine for a SAP task (0, 1 or 2).
This command is persistent.

4.3 G02 – Circular / helical interpolation

Circular or helical interpolation, clockwise: The target positions are given as absolute or relative coordinates, depending on the mode selected. The track speed can be defined e.g. with the F command. The acceleration is programmed in the initialization routine for a SAP task (0, 1 or 2). The target coordinates for the arc are entered as parameters. The required center coordinates are entered as parameters I and J. The center coordinates may be entered as relative coordinates even in absolute mode (G90). This option can be selected in the configuration module.
Command G02 is persistent.

4.4 G03 – Circular / helical interpolation

Circular interpolation, anti-clockwise: otherwise, as G02.

4.5 G04 – Dwell time

Dwell time in seconds. Floating point numeric values can also be entered.

Example: G04 3

4.6 G17 – Plane selection

Select X-Y plane for circular interpolation and tool radius correction.
The X-Y plane is the default value after system startup.

4.7 G18 – Plane selection

Select Z-X plane for circular interpolation and tool radius correction.

4.8 G19 – Plane selection

Select Y-Z plane for circular interpolation and tool radius correction.

4.9 G21 – Reflection of Y axis

The coordinates programmed for the Y axis are reflected about the zero point. Any reflection of the X axis already active will not be affected by this command.

4.10 G22 – Reflection of X axis

The coordinates programmed for the X axis are reflected about the zero point. Any reflection of the Y axis already active will not be affected by this command.

4.11 G23 – Reflection of X and Y axes

The coordinates programmed for the X and Y axes are reflected about the zero point.

4.12 G24 – Disable reflection of all axes

All programmed reflection functions (G21, G22, G23) are switched off.

4.13 G39 – Program positioning factor

A positioning factor can be programmed for every axis in the system. Where axis reflection is selected, the positioning values for the axis concerned will be multiplied by this factor. The default value is -1.

Example:

N200 G39 X2 Y2

When G23 is active, the contour size will be doubled.

N500 G39 (Reset all positioning factors to -1)

4.14 G40 – Cancel tool radius correction

Switch off tool radius correction.

4.15 G41 – Tool radius correction, left

Switch on tool radius correction, left.

4.16 G42 – Tool radius correction, right

Switch on tool radius correction, right.

4.17 G51 – Program effective radius

G51 can be used to program an axis-specific radius in the specified interpolation path units. This will be used in the interpolation calculation where rotatory axes are involved in translatory interpolation runs. This supports processing of the surface of a cylindrical object turned about a rotatory axis. The unit of measure for the position values with rotatory axes is then not a rotatory value but the interpolation unit. This value is then converted using the radius entered into the angle to be traversed.

Example:

N200 G51 C120 D40

For this functionality, please refer to the version information in section 1. Version dated 16.07.2003 required as a minimum.

4.18 G53 – Disable zero offset

Switch off zero offset for all interpolation axes. This is not a spooler command and so cancels any associated contour programmed with spooler commands such as G01.

Example: N0100 G53

This instruction is equivalent to the following sequence of commands (for a 3-axis system):

```
N0100  G90              ' Absolute path information
N0101  G92 X0 Y0 Z0     ' Set zero offset to 0 for X and Y
```

4.19 G54 - G58 – Set zero offset

This command can be used to set the zero offset for all axes to the values in the associated zero offset table. The zero offset table is stored as part of the system data and can be edited with the commissioning program mcfg. With G54, record 0 will be selected, with G55 record 1, etc.
To set the zero offset to any desired value, command G92 should be used. This is not a spooler command and so cancels any associated contour programmed with spooler commands such as G01.

4.20 G60 – Define interpolation axes

Selection of the axes involved in interpolation commands (G01, G02, G03).

Example:

```
G60 X Y Z
```

This command defines the axes associated with the interpolation processed when the above commands are used. These details are used for initialization and are generally retrieved only once at program start.

4.21 G70 – Measurement in inches

Measurements for positions and velocities in inches.

4.22 G71 – Measurement in mm

Measurements for positions and velocities in mm (default value).

4.23 G74 – Reference run

Command G74 is designed to trigger axis referencing. This function has to be programmed / modified in Task 0 on an application-specific basis.

4.24 G90 – Absolute distance mode

Select absolute distance mode.

4.25 G91 – Relative distance mode

Select relative distance mode.

4.26 G92 – Set zero offset

Program zero offset for the specified axes to any desired value. This is not a spooler command and so cancels any associated contour programmed with spooler commands such as G01.

Example: N0100 G92 X100 Y-200

If G90 has been programmed, the zero offset will be relative to the absolute zero point. If G91 has been programmed, the zero offset will be relative to the zero point currently set.
A G53 call will reset the zero offset. Calling G54 – G58 will cancel a zero offset set with G92 and set a zero offset according to the table.
A G92 call without a positioning command will not cause any movement of the axes.

4.27 G93 – Reverse-time feed encoding

Reverse-time feed encoding. This function is not yet implemented, and should not be used.

4.28 G94 – Time unit in minutes

Time unit given in minutes, i.e. feed unit in [mm/min] or [inches/min].

4.29 G98 – Set position of software limit switch - left

The software limit switches are not activated with this command. They are activated automatically when the respective axes are referenced, assuming this functionality has been defined in mcfg.exe. If this command is not used, the limit switch position will be initialized to the default position entered in mcfg.exe.

4.30 G99 – Set position of software limit switch - right

As section 4.29, but for software limit switch, right.

4.31 G150 – Spline off

Deselect spline interpolation.

4.32 G151 – Spline on

Select spline interpolation.

4.33 G153 – Read zero offset

This command can be used to read the current absolute position of the zero offset for all axes in the system. The values returned will be stored in the system variables CD[50 + axis index] in the user-specific interpolation unit.

Example:

```
N100    G153    ' Read zero offset in CD50 ff.
```

4.34 G154 – Reprogram zero offset returned

This command can be used to reprogram the position values for the zero offset for any number of axes read in via G153. The values are taken from the system variables CD[50 + axis index].

Example:
N100 G154 ' Set zero offset (absolute)

4.35 G161 – Center coordinates, relative or absolute

When executing circuits in absolute mode (G90), the center coordinates are also given as absolute coordinates. This option can be set in the configuration program IniCfg.

4.36 G162 – Center coordinates always relative

Center coordinates are always given as relative coordinates.

5 M-codes

5.1 M00 – Program stop

Program stop, spindle, coolant and feed off: Setting/resetting the associated digital outputs must be implemented/modified in Task 0. In step mode, the instruction switches over. The program can be restarted from the point at which it stopped.

5.2 M01 – Optional stop

Optional stop: if the “Optional stop” button has been pressed, this instruction switches over to step mode.

5.3 M02 – End of program

End of program. The program stops. No further outputs will be changed!
This command does not trigger any call to Task 0. See also [section 5.13].

5.4 M03 – Spindle clockwise

This command is dependent on the type of machine and signifies e.g. <<Spindle on, clockwise>>. The relevant functionality must be programmed in Task 0.

5.5 M04 – Spindle anti-clockwise

This command is dependent on the type of machine and signifies e.g. <<Spindle on, anti-clockwise>>. The relevant functionality must be programmed in Task 0.

5.6 M05 – Spindle stop

This command is dependent on the type of machine and signifies e.g. <<Spindle stop>>. The relevant functionality must be programmed in Task 0.

5.7 M06 – Change tool

This command signifies <<Trigger change of tool>>. The relevant functionality must be programmed in Task 0.

5.8 M08 – Coolant on

This command signifies <<Coolant on>>. The relevant functionality must be programmed in Task 0.

5.9 M09 – Coolant off

This command signifies <<Coolant off>>. The relevant functionality must be programmed in Task 0.

5.10 M17 – End of sub-routine

End of sub-routine and return to calling program.

5.11 M26 – Set output

Set digital output.

Syntax: M26 Parameter

Parameter is a 3-digit number, with the first digit indicating the axis number and the second and third digits the output to be set. The value 00 in the second and third digits causes all outputs for this axis channel to be set.

Example: N0100 M26 102 // Set output 2 for axis 1

From version 3-00 onwards, this command is a spooler command and does not cause any intermediate stop when called during an interpolation run.

5.12 M27 – Reset output

Reset digital output.

Syntax: M27 Parameter

Parameter is a 3-digit number, with the first digit indicating the axis number and the second and third digits the output to be reset. The value 00 in the second and third digits causes all outputs for this axis channel to be reset.

Example: N0100 M27 102 // Reset output 2 for axis 1

From version 3-00 onwards, this command is a spooler command and does not cause any intermediate stop when called during an interpolation run.

5.13 M30 – Program stop

Program stop with reset, spindle, coolant and feed off (similar to M00): setting/resetting the associated digital outputs must be implemented/modified in Task 0.
Program execution is stopped and the program pointer returned to the start of the program. Task 0 is then called. Additional functions have to be processed at this point. The program can then be restarted with contcnct (3).

5.14 M96 – Unconditional jump

Unconditional jump to the label specified. Labels are entered with the prefix L.

Example:

```
N220 M96 L114          // Unconditional jump to label 114
N221 ...
...
N348 L114              // Label 114
N349 ....
```

5.15 M98 – Call sub-routine

Sub-routine call when L label specified. An O counter can optionally be used to enter the number of sub-routine cycles.

Example:

```
N220 M98 L114 O5       // Call sub-routine 114 (5 times)
N221 ...               // Further program
...
N348 L114              // Sub-routine 114
N349 ....              // Sub-routine body
N400 M17               // Sub-routine end
```

Sub-routines can be inserted locally into the relevant source code after the end of the program module. Global sub-routines can also be added to the source code in include files. Further details can be found in the section on “Integrating include files”.

5.16 M100 – Reset “Program end” output

Application-specific command: Reset “Program end” output. This function must be programmed in Task 0.

5.17 M150 – Start recording for graphical systems analysis

Calling this command records the change of position in the first three axes over 5 seconds. The change of position can then be displayed in mcfg using the graphic screen.

6 Other codes

6.1 Labels

Labels are used to identify sub-routines and jump destinations, and are entered as numbers prefixed with the letter L.

Example:
N220 L114 // Label 114

6.2 Special functions supported

Instruction code	Instruction no	Parameter	Comments
S	1	Parameter in CD0	Main spindle revolutions
T	2	Parameter in CD0	Tool change

These codes (instruction numbers) are passed to Task 0 in variable CI2 and executed as sub-routines.

6.3 F-command

Set feed velocity for G01, G02, G03 commands in the unit selected in the initialization tasks. If a line with a G-command also contains an F-command, the newly-programmed velocity will apply for this profile.
Any braking of the axis will be executed in the preceding profile section. Any acceleration of the axis will be executed in the current profile.

6.4 S-command

Setting the main spindle revolutions: This functionality must be programmed in Task 0. The function code for this command is 1. The parameter is passed in CD0. The S-command programmed in Task 0 interrupts the interpolation contour and executes the preceding lines using the SSMSIW command. This is a command specifically for this purpose, which is not used in standard rw_SymPas programming.
If S-commands are to be implemented that do not interrupt the interpolation contour, this command must be omitted. In this case, appropriate spooler commands must be executed to control the main spindle of the processing tool (SSF).

6.5 T-command

Tool change: This functionality must be programmed in Task 0. The function code for this command is 2. The parameter is passed in CD0.
To correct the tool radius, this command is used to select a correction record defined in the tool table. By default, tool no 0 is selected at system startup.

6.6 D-command

Select tool correction table: This parameter is currently ignored and so not evaluated.

7 Comments

Comments can be enclosed in round brackets. It is also possible to mark the end of a line as comments with a single quote. Comments can be nested.

Examples:

```
N210 G00 X10 Y-20     (This is a comment)
N220 G90              ‘ This is a comment
```

Round brackets can still be used within expressions without them being treated as comments. In the example below, the contents of the brackets are **not** a comment.

```
N230   G01 X(CD601 - Y.rp)
```

8 Conditional program execution

Based on Boolean values, sections of programs can be executed conditionally. The standards for expressions and operators are based on APCI-8001 SAP programming.

Example:

```
N350   $if (expression) then begin
N400   .....                        (instructions)
N450   $end
```

The (expression) is a Boolean expression. Some examples:

```
(CI600 < 20)          // Result of a comparison
BOOLEAN (CI600)       // Conversion of a numeric value
(X.digib.5)           // Query digital input 5 for the X axis
```

The conditionally executable program block must be terminated with $end. If required, a $else branch can be declared.

Example:

```
N350   $if (expression) then begin
N400   .....                        (instructions)
N450   $end else begin
N500   ....                         (instructions)
N550   $end
```

It is also possible to construct if-else-if chains.

Example:

```
N350   $if (expression 1) then begin
N400   .....                        (instructions)
```

```
N450   $end else if (expression 2) begin
N500   ....                          (instructions)
N550   $end else if (expression 3) begin
N600   ....                          (instructions)
N650   $end else begin
N700   ....                          (instructions)
N750   $end
```

9 Loops

9.1 Do-while loop

Example:

N350 $while (expression) do begin
N400 ..... (instructions)
N450 $end

(Expression) must return a Boolean value (see IF). The loop will be executed until (expression) is false. The value of (expression) is checked at the start of the loop, i.e. if the value is false when the loop is reached, the loop is by-passed.

9.2 Repeat-until loop

Example:

N350 $repeat begin
N400 ..... (instructions)
N450 $end until (expression)

(Expression) must return a Boolean value (see IF). The loop will be executed until (expression) is true. The value of (expression) is checked at the end of the loop, i.e. the loop is executed at least once.

9.3 For loop

Examples:

N350 $for CI600:=1 to 10 do begin
- N400 ..... (Instruction Case-Insensitive mode is possible, configurable in IniCfg
- Program stop also possible in referenciatio and in the single step mode.
- Position an time unit of interpolation commands now is selected in IniCfg
- New command M150
- Bug at G-Code program without M30 at the end corrected
- Modul name in programs not required anymoreProgrammen nicht mehr erforderlich

ngen)
N450 $end

N750 $for CI600:=10 downto 1 do begin
N800 ..... (instructions)
N850 $end

```
N850    $end
```

The run-time variable (here CI600) must be an integer variable. The definition of the “for” loop includes a start and an end value. Start and end values may also be expressions.

10 Integrating include files

The $I instruction can be used to incorporate include files into a source code. Include files may contain further include files.

Example:

$I UP2.SRC

This function can be used to insert a series of sub-routines consisting of standalone programs into a piece of source code.

Example:

(in the main program)

```
N400   M98 L2000 O3          (Call sub-routine L2000 3 times)
N500   ...
N900   M02                   (End main program)
%
```

(Definition of a local sub-routine)

```
%107
N0100  L107                  (Local sub-routine 107)
N0200  ...                   (Program body)
N0300  M98    L1000 O2       (Call sub-routine L1000 2 times)
N1000  M17                   (Return)
%

$I Unterprogramme.inc        (Include library of local sub-routines)
```

The file **Unterprogramme.inc** may contain e.g. the following text:

```
$I UP1.SRC    (Module 1000: Bore out keyhole)
$I UP2.SRC    (Module 2000: Add tapped hole for compressed air)
$I UP3.SRC    (Module 3000: Bevel top edge)
```

Another example of the global sub-routine UP1.SRC:

```
%1000                        (Module name)
N100   L1000                 (Label for sub-routine call)
N200   G00 X0 Y0 Z0
N300   G00 X10 Y10 Z-10
N400   M17                   (Return from sub-routine)
N500   %
```

11 Incorporation of rw_SymPas commands

In programs conforming to DIN 66025, rw_SymPas source code elements can be incorporated. In this case, DIN mode is terminated with the instruction $DINEND.

Example:
N200 $DINEND

After this instruction, rw_SymPAs instructions can be inserted without line numbering. To switch back to DIN mode, the compiler instruction {$DINSTART} is used.

Example:

{$DINSTART}
N220 G00 X10

12 Calculation parameters

For mathematical operations like calculating target positions, calculation parameters can be used. These calculation parameters can also be used to decide on conditional jump functions.
Calculation parameters are referenced as CD<n> for floating-point numbers and CI<n> for integer values. Values for n of 600 – 699 are reserved for the user.

Warning: The system also allows other values of n. However, any values outside the specified range may be system variables, and using them could cause instability in the system.

Example:

N220 CD620 := 100
N230 G00 X2*CD620