
POSITIONING AND CONTORUING CONTROL SYSTEME APCI-8001

User interface installation

McuWIN

1	Version information	7
2	Installation	7
3	Remarks for the installation.....	7
3.1	Creating a User directory	7
3.2	Installation of the axes with mcfg	8
3.3	First call of McuWIN	8
3.4	Installing McuWIN	8
3.5	Displaying a bit variable	9
3.6	Editing programs	9
3.7	Executing programs	9
3.8	Executing programs step by step.....	9
3.9	Stops in the program execution	9
3.10	Teaching of position values.....	9
3.11	Override.....	10
3.12	Further remarks.....	10
3.12.1	Used axes (usedAxis)	10
3.12.2	Axes in the interpolation connection (InterpolationAxis).....	10
3.12.3	Customer specific layout of the user interface.....	10
4	Hardware-preconditions.....	11
4.1	Main spindle	11
4.2	Output end of program	11
4.3	Output cooling	11
4.4	Overview assignment of the outputs	11
4.5	Reference switch.....	12
5	Variable in McuWIN.INI	12
5.1	[DESKTOP]	12
5.1.1	SingleStep	12
5.1.2	Override	12
5.1.3	OverrideEnable	13
5.1.4	OverrideMax.....	13
5.1.5	OverrideMin.....	13
5.2	[EDITOR].....	13
5.2.1	SrcFileName	13
5.2.2	AutoLoadOnActivate	13
5.2.3	DisableEdit	13
5.2.4	SrcFilter	13
5.3	[MCU]	13
5.3.1	AutoSetNum	13

5.3.2	usedAxis.....	14
5.3.3	InterpolationAxis.....	14
5.3.4	EECheck	14
5.3.5	FehlerVarCI, RefVarCI.....	14
5.3.6	RefAfterEO.....	14
5.3.7	BaseAdress.....	14
5.3.8	SystemFileName.....	14
5.3.9	BootFileName	14
5.3.10	Task?Aktiv.....	14
5.3.11	TRAC, TRVL	14
5.3.12	FreeingVelFactor.....	15
5.3.13	ReferenceSwitchLatch	15
5.3.14	NoRefToLimitSwitch.....	15
5.3.15	ReferenzUeberwachung (reference monitoring).....	15
5.3.16	UeberwachungsTask (monitoring task)	15
5.3.17	UserTask.....	15
5.3.18	DINGCodes.....	15
5.3.19	UserTaskAutoRun.....	15
5.3.20	UserTaskAutoStop	15
5.3.21	SProfile.....	16
5.3.22	JerkRel.....	16
5.3.23	LookAhead	16
5.3.24	LookAheadDeep	16
5.3.25	A?.mdvel	16
5.3.26	CenterPointRelative	16
5.3.27	RotatoricUnit	16
5.4	[TEACHMASK].....	17
5.4.1	GlyphLeftX=Y	17
5.4.2	GlyphRightX=Y	17
5.5	[SYSTEM].....	17
5.5.1	CompileAlways.....	17
5.5.2	RebootEnabled	17
5.5.3	Axis compensation	17
5.5.4	HWStart(n)	18
5.5.5	HWStop(n)	18
5.5.6	HWRef(n)	18
5.5.7	HWReset(n)	18
5.5.8	HWSingleStep(n)	18
5.6	[REFERENZFAHRT] – Reference travel	19
5.6.1	Order	19
5.6.2	ReferenzSchalter (reference switch).....	19
5.6.3	Indexsuche=0 (Index search=0)	19
5.6.4	Direction	19
5.6.5	GotoZeroAfterRef.....	19
5.6.6	ReferenzPosX / Y /	19
5.7	[FEHLERTEXTE] – Error texts.....	19
5.7.1	User specific error texts	20
5.7.2	[FEHLERINFO] – Error info	20
5.8	[WARNTEXTE] – Warning texts	20
	User specific warning texts	20
5.8.1	20	
5.9	[TOOLCOMPENSATION] tool compensation.....	21
5.9.1	TcFileName.....	21
5.10	[DDE] 21	

6.1	CD variable	22
6.2	CI variable	23

7 Description of the operating modes25

7.1	EditMode	25
7.2	RunMode.....	25
7.3	StepMode	25
7.4	HaltMode	26
7.5	ManMode	26
7.6	Reference travel.....	26
7.6.1	Referenciation on reference switch.....	26
7.6.2	Referenciation on limit switch	26
7.6.3	Fine positioning on zero track	26
7.6.4	Setting the home position	27

8 Description of the SAP task environment27

8.1	Background information for the APCI-8001	27
8.1.1	Compiler mode for G-Code programs.....	27
8.1.1.1	Command lines Compiler NCC.EXE	27
8.1.1.2	MCFG for MS-DOS	27
8.1.1.3	MCFG for 32bit-Windows	28
8.2	Supplementing of customer specific codes.....	28
8.2.1	G-codes.....	28
8.2.2	M-codes	28
8.2.3	Calling conventions.....	29
8.3	Task 0 – Command interpreter	29
8.3.1	Implementation of customer specific G-codes.....	29
8.3.2	Include-Files in TASK0.....	29
8.3.2.1	KdRefFahrt.INC.....	29
8.3.2.2	AppGCodes.INC.....	30
8.3.2.3	AppCommands.INC	31
8.3.2.4	AppMCodes.INC	31
8.3.2.5	Application.INC.....	31
8.4	Task 1 – Initialisation and monitoring task.....	31
8.4.1	Initialisations in module TASK1.SRC.....	31
8.4.1.1	Unit for interpolation commands.....	31
8.4.1.2	Acceleration for inpterpolaion commands	31
8.4.1.3	CI- and CD-variable.....	31
8.4.2	Monitorings.....	32
8.4.2.1	Position error	32
8.4.2.2	Hardware limit switch	32
8.4.2.3	Software limit switch	32
8.4.2.4	Emergency stop	32
8.4.2.5	Gain ready	32
8.4.2.6	Encoder-Error-Flag.....	32
8.4.2.7	Encoder verification.....	33
8.4.2.8	Further monitoring functions.....	33

9 Customer specific extensions	33
9.1 AppTask2.SRC	33
9.2 AppGCode.INC	33
9.3 AppMCodes.INC	33
9.4 AppCommands.INC	34
10 Examples for application specific amendments.....	34
10.1 Coolant On/Off (M08 / M09).....	34
10.2 Main spindle	34
11 DDE-communication.....	35
12 Error diagnostics	37

1 Version information

The following description is valid for:

McuWIN.EXE	from version 2.5.3.432
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	version 2.5.3.35

2 Installation

McuWIN runs under all 32-bit Windows platforms (Windows 95/98/ME, NT, 2000, XP). Please ensure that before the installation of McuWIN the programs Miniport, fwsetup and mcfg are installed. The installation of McuWIN is executed by calling SETUP.EXE. The user will be guided through the installation process. You can uninstall the package again with the menu or with the system control.

3 Remarks for the installation

After the installation of McuWIN, the environment must be adapted to the user specific requirements and conditions. In the following the single steps are described. For additional information, required supplements, and any other needs we are at your disposal:

info@addi-data.com

3.1 Creating a User directory

With the installation a user directory was created, that contains the files System.DAT and RWMOS.ELF. Updates and enthält. Updates and additions must be copied into this directory. The program mcfg.exe must be adjusted to the files under „File – Project Parameter“.

3.2 Installation of the axes with mcfg

Before you can use the program McuWIN, the complete system must be installed and configured with the installation program mcfg.exe . The following characteristics are of special significance:

- Axis name
Caution: When using McuWIN as G-Code interface, the axis names may not contain numeric values.
- Mechanic parameters
(encoder resolution, gear factor)
- Setting of the bearing controllers
- Maximum position error
- Configuration of the inputs and outputs, limit switch , reference switches, amplifier enable etc. See also chapter „Hardware-preconditions“
- Default of the Jog- and Home-accelerations and speeds
- Stop-Deceleration
(Target-Velocity must always be 0)
- Declaration of the software limit switch

The axes must be moved controlled with the Motion-Tools in mcfg. These values, which are set in mcfg.exe, will be stored in the indicated file SYSTEM.DAT during the storaction. It is important that McuWIN.EXE also accesses to this file as system file.

Caution: If the file SYSTEM.DAT was copied from a CD, it could be necessary that you must remove the file attribute ReadOnly, so that you can store it.

3.3 First call of McuWIN

Now the program McuWIN is called for the first time and terminated at once. Herewith the MCUWIN.INI is created and modified. After this, this file will be adapted with the program IniCfg.exe .

3.4 Installing McuWIN

The application specific adaptation of McuWIN will be executed with the program IniCfg, which was installed with McuWIN. The executed settings will be stored in the configuration file McuWIN.INI. Manual modifications in the file McuWIN.INI are obsolete. However, chapter 5 explains the contents of the INI-file.

3.5 Displaying a bit variable

The also installed program RegDisp allows displaying, grouping and marking bit information on the screen. Bit variables are for instance: Digital inputs, outputs, contents of CI-variables or also I/Os of additional PCI-groups. This program is very helpful during the system development.

The parameterizing of the several register cards and display elements is realized by menus that are opened with a click of the right mouse key on the specific display element.

3.6 Editing programs

In the Edit-mode in the editor window McuWIN programs can be loaded, saved and edited. Before the execution of the program a syntax test can be done. The result of the syntax test will be displayed in the error window.

3.7 Executing programs

With the „Start“-button you can start the program that is active in the editor. The editor then is switched to the trace mode.

3.8 Executing programs step by step

If you have selected in the INF file correctly, the program can be tested with the „step“-button in gradual operation. At spooler movement sequences (G01, G02, G03) the actual movement will be executed firstly after the execution of the last line that belongs to the corresponding section. By a mouse click in the editor window the traversing movement can be started at an earlier point at time.

3.9 Stops in the program execution

With M00 absolute stops can be set. When reaching M00, McuWIN switches to single step operation and then can be continued with „Step“ or „Forts“ – button.

With M01 absolute stops can be set. These stops can be enabled with the mouse on the button “Desired stop” („Wahlweiser Halt“).

3.10 Teaching of position values

You can open this with the mouse button „Teach In Maske“. Here the axes can be traversed manually by mouse button. By clicking on the key “Teach” the current position value is inserted at the cursor position in the source text file.

The displayed picture in the respecting button can be changed in the file File McuWIN.INI in the section [TEACHMASK] (see also section [TEACHMASK]).

3.11 Override

The Override-function is configured in the INI-File. With the shift controller the Override can be set during the execution of the program. The Override value also can be set by an external source. This functionality must be programmed from the user e.g. AppTask2.SRC in a closed loop.

3.12 Further remarks

After the first calling of McuWIN especially the following values shall be set by IniCfg :

3.12.1 Used axes (usedAxis)

Bit coded numeric value of the axes used in the system:

- At these axes the control loop will be closed
- For these axes positioning display windows are generated
- These axes are taken in consideration at the Teach-In
- These axes can be traversed manually
- These axes are effected by the Override-function
- Firstly, when all these axes were referenced, hardware limit switches will be displayed as errors in McuWIN

3.12.2 Axes in the interpolation connection (InterpolationAxis)

Bit coded numeric values of the axes that are connected with the interpolation. Here, it is possible that are less axes than indicated in usedAxis.

3.12.3 Customer specific layout of the user interface

From version 2.5.3.24 on you can integrate a bitmap file into the status display window. Herefore you must copy a bitmap file named „LOGO.BMP“ into the McuWIN-directory. The optimal resolution of the of the file is 529 x 285 pixel with a screen resolution of 1024 x 768 points. The relation of the pages shall be kept in order to avoid distortions of the pictures.

4 Hardware-preconditions

4.1 Main spindle

The main spindle is switched on or switched off with a digital output. The direction of the main spindle also is switched with a digital output. The assignment of these outputs must correspond with the programmed data in TASK0.SRC or in the corresponding Include-files.

4.2 Output end of program

When starting an automatic program, McuWIN calls the special command 1001 in Task 0. Herewith the output "Program – is running" is set. After the termination of the automatic program the output "Program - is running" is reset. This is done with the calling of the special command 1000 of McuWIN. The output "Program – is running" is O8 of the first axis channel as default.

With the first single step of a program of McuWIN the special command 1002 is called in Task 0. In this also the output "Program – is running" is set.

4.3 Output cooling

The assignment of the output cooling must correspond with the data programmed in TASK0.SRC or in the corresponding Include-files.

4.4 Overview assignment of the outputs

Table: Assignment of the outputs of the **APCI-8001**

Axis	Input	Pin / Board / Connector
1	O1	
1	O2	
1	O3	
1	O4	Cooling On
1	O5	
1	O6	Direction main spindle
1	O7	Main spindle On
1	O8	Display program finish

The declarations of the outputs can be found in GCode.INC and can be modified, if necessary.

4.5 Reference switch

If the axis referenciation shall be executed on a reference switch, the latch inputs that correspond with the axes **must** be used as reference switch inputs. Additionally, the reference switches must switch from 0 to + 24 V when recognizing the reference signal.

Table: Allocation axis / latch input

Axis	Input	Pin / Board / Connector
1	I14	47 / APCI-8001 / X1
2	I15	48 / APCI-8001/ X1
3	I16	49 / APCI-8001/ X1
4	I30	47 / OPMF / X1
5	I31	48 / OPMF / X1
6	I32	49 / OPMF / X1
7	I39	15 / OPMF / X2 (SUB-D ext.)
8	I40	16 / OPMF / X2 (SUB-D ext.)

5 Variable in McuWIN.INI

The following settings are executed by the program IniCfg.exe and are used from McuWIN.exe for the configuration of and adaptation to the application. A manual editing of this file usually is not required. However, you can use this section in order to know the several settings more comprehensively.

5.1 [DESKTOP]

In this section the appearance of the user interface is administrated. The values WindowState, Left, Top, Height, and Width save the position of the user interface and are administrated of McuWIN. Further parameters are described in the following part:

5.1.1 SingleStep

Marker for showing/not showing the singlestep button (Default: 1 = On).

5.1.2 Override

Marker for showing/not showing the Override-trackbar (Default: 0 = Off).

5.1.3 OverrideEnable

Marker, if the Override track bar can be modified in the user interface by the mouse (Default: TRUE). If the Override value shall be set e.g. in the task environment OverrideEnable=0. In this case the Override-track bar is only for display.

5.1.4 OverrideMax

Maximum value of the Override track bar in % (Default: 125).

5.1.5 OverrideMin

Minimum of the Override track bar in % (Default: 0). Here also negative values are possible. Then you can drive back within a traversing profile.

5.2 [EDITOR]

5.2.1 SrcFileName

In Parameter SrcFileName the name and path of the user task is saved. This parameter is administrated by McuWIN.

5.2.2 AutoLoadOnActivate

When this parameter is set on 1, the indicated source text file is loaded newly every time when activating McuWIN under Windows, as soon as the system is in the edit mode, i.e. when no user program is executed.

5.2.3 DisableEdit

When this parameter is set on 1, user programs can be loaded and executed, but not edited.

5.2.4 SrcFilter

Here default extension for source text files can be indicated. This extension will always be proposed after the first selection when opening a file (when a source text file shall be opened).

5.3 [MCU]

In this section the control specific parameters can be found.

5.3.1 AutoSetNum

If this marker is set ($\neq 0$), then the line numbers (Nxxx), which are normally required in the G-Code mode, are not needed anymore.

5.3.2 usedAxis

Indication of the actual used axes (bit coded). With this variable, the axes of the operation system, which are processed by McuWIN, can be limited (Default FFh)

Example: You have a system with 3 axes (X, Y and Z)

Then here the value 7 must be entered because here the three bits with the lowest value are set.

5.3.3 InterpolationAxis

Indication of the axes that are in the connection of interpolation!

5.3.4 EECheck

Indication of the axes (bit coded), at which the encoder-error is tested. In step motor systems this variable can be set on 0. With this variable also the position monitoring is activated by index-latch function.

5.3.5 FehlerVarCI, RefVarCI

Indication of the CI-variable for error display and for reference status display. These variables are preset by the programs TASK0.SRC and TASK1.SRC and may not be changed.

5.3.6 RefAfterEO

This marker indicates, if the system must be referenced newly after an emergency stop (default value=1).

5.3.7 BaseAdress

In MCUG2-systems (PA8000, PS840) here the base address of the control must be entered (Default = 768 = 300H).

5.3.8 SystemFileName

Path and name of the system file (Default: SYSTEM.DAT in the program directory).

5.3.9 BootFileName

Path and name of the boot file (Default: RWMOS.ELF in the program directory).

Caution: In MCUG2 systems (PA8000, PS840) here usually RWTOS.BTL.

5.3.10 Task?Aktiv

? is the index of task 0..3. With the respective marker it is indicated if the respective task shall be loaded at the program start. If yes, under **Task?CncFileName** the name of the file to be loaded is indicated with the path (Default: TASK?.CNC in the program directory).

5.3.11 TRAC, TRVL

Here the default values for the track speed and track acceleration for interpolation commands are entered. The units for these values shall be set in the monitoring task. If they should not be set there, mm/sec and mm/sec² are selected.

5.3.12 FreeingVelFactor

Free movements, e.g. at the referenciation of the axes, are executed normally with the rapid traverse – speed / acceleration (Jog-velocity, Jog-Acceleration). With this parameter, these values can be reduced or increased.

5.3.13 ReferenceSwitchLatch

In this marker the axes are entered bit coded, where the reference switch input is at the same time a fast latch input. If this is not the case or when the latch signal is at the moment a fast latch input. If this is not the case or if the latch signal recognition shall be checked, this can be deactivated axis specific with help of the marker.

5.3.14 NoRefToLimitSwitch

If not to one reference switch is referenced, the referenciation is executed on a limit switch. If this should be not wanted, this can be deactivated (bit coded) with the marker. In this case the referenciation is only to the index signal or there is no referenciation.

5.3.15 ReferenzUeberwachung (reference monitoring)

In this parameter it is indicated bit coded which axes must be referenced so that a program can be traversed (Default 0xFF);

As reference switch always the latch inputs of the respective inputs must be used (inputs 14, 15 and 16 at the first three axes).

5.3.16 UeberwachungsTask (monitoring task)

Here the task must be indicated, which monitors the system. If there is no monitoring task available, here a negative value must be indicated (Default: -1).

The monitoring task must always be activated with the task Task?Aktiv=1 and started automatically with the task Task?Autorun=1. With Task?CncFileName= the file name and path are indicated (Default: TASK?.CNC).

5.3.17 UserTask

Here the task must be entered in which the program shall be executed in the editor. Default is 0.

5.3.18 DINGCodes

With the value 1 for this parameter the option G-Code processing can be activated. Then the UserTask must be Task 3. Hereto a command interpreter task must be loaded in Task 0 and a monitoring task in Task 1.

5.3.19 UserTaskAutoRun

With this marker it is indicated if the UserTask shall be started automatically after the program start.

5.3.20 UserTaskAutoStop

With this marker it is indicated if the system shall be reset after finishing McuWIN. If yes, the file **StopUserTask.CNC** in the User-Task is loaded and started.

Caution: **StopUserTask.CNC** must be compiled for the correct task.

5.3.21 SProfile

Indicator, shows if the S-profile or the trapezoid-speed-profiles shall be traversed (Default = 0).

5.3.22 JerkRel

Factor for jolt at S-profiles (Default = 0.5):

0.0 = max. allowed jolt, corresponds trapezoid-speed-profile

1.0 = min. jolt, corresponds triangular acceleration phase

Values between 0 and 1 correspond trapezoid acceleration phase

Here a floating point value between 0.0 and 1.0 can be entered, if SProfile = 1.

5.3.23 LookAhead

Indicator, indicates if the Look-Ahead speed limitation is active (Default = 1). In the active Look-Ahead for the several axes the maximum allowed velocity jumps is indicated, which is indicated in A?.mdvel.

5.3.24 LookAheadDeep

Whole number variable indicates in how many interpolation-traverse commands in the spooler the spooler processing begins. So this number is always the max. number of traverse profiles, with which a LookAhead calculation can be realized.

5.3.25 A?.mdvel

An axis specific max. elocity jump in the LookAhead mode (Default = 0,1). For „?“ the respective axis number (1..n) is significant. The unit is the respective interpolation unit.

5.3.26 CenterPointRelative

Marker with which can be indicated if the circle centre coordinates are indicated as relative or absolute coordinates in the absolute mode (G90) .

5.3.27 RotatoricUnit















Marker with which can be indicated if rotatory axes (axes with a rotatory user unit) are programmed in the selected compiling unit or in the user specific rotatory unit when used within translatory defines interpolation travels (G01, G02, G03). The translation between translatory and rotatory units is realized with the axis specific effective radius, which can be programmed e.g. with the command G51.

5.4 [TEACHMASK]

In this section each traversing button can be assigned to a bitmap.















5.4.1 GlyphLeftX=Y

This command assigns the buttons of the left column to a picture. X is the line of the button (counting from 0 upwards). Y is the index of the available buttons (according to the table).

0	1	2	3	4	5	6	7	8	9	10	11	12	13
													

5.4.2 GlyphRightX=Y

This command assigns the buttons of the right columns to a picture. X is the line of the button (counting from 0 upwards). Y is the index of the available buttons (according to the table).

0	1	2	3	4	5	6	7	8	9	10	11	12	13
													

5.5 [SYSTEM]

5.5.1 CompileAlways

If this marker is <> 0 at each program start the active SRC-file is compiled firstly and then loaded to the control. If CompileAlways = 0 is set, it is firstly checked, if the CNC-file already exists. If yes, only the SCRC-file is compiled, if the date of the SCRC-file is more current than the date of the CNC-file.

Herewith in a lot of cases time can be saved for the (redundant) compilation process. This is especially useful with slow PCs or large programs.

5.5.2 RebootEnabled

With this marker it can be controlled, if the system is rebooted by double-clicking on the error window.

5.5.3 Axis compensation

With this marker axis compensation can be loaded. Hereto it is accessed to a separate INI-file, whose name and paths are indicated in *KompensationsFile* (Default: CP.INI).

This INI-File can be created with the program WrParams.EXE. If this option is used, RWMOS must be used with the option GEAR.

Caution: In the Ini-file the path must be indicated, otherwise the file cannot be found!

5.5.4 HWStart(n)

With the marker HWStart one or several digital inputs can be indicated by bits, at which a hardware program start can be realised. The number of the axis channel is n. This function is only effective, if in the editor are no modifications available that were not saved.

Example: Input 2 of the X-axis (first axis) shall start the user program:

HWStart1=2

5.5.5 HWStop(n)

With the marker HWStop one or several digital inputs can be indicated by bits, at which a hardware program start can be realised. The number of the axis channel is n.

Example: Input 3 of the X-axis (first axis) shall stop the user program:

HWStop1=4

5.5.6 HWRef(n)

With the marker HWRef one or several digital inputs can be indicated by bits, at which a hardware start of the reference travel can be realised. The number of the axis channel is n.

Example: Input 3 X-axis (first axis) shall start the reference travel.

HWRef1=4

5.5.7 HWReset(n)

With the marker HWReset one or several digital inputs can be indicated by bits, with which a hardware-activation of the reset-buttons can be realised. The number of the axis channel is n.

Example: Input 4 of the X-axis (first axis) shall cause a reset:

HWReset1=8

5.5.8 HWSingleStep(n)

With the marker HWSingleStep one or several digital inputs can be indicated by bits, with which a hardware-activation of the SingleStep-buttons can be realised. The number of the axis channel is n.

Example: Input 5 of the X-axis (first axis) shall cause a reset:

HWSingleStep1=\$10

5.6 [REFERENZFAHRT] – Reference travel

5.6.1 Order

Order of the referenciation. The axis number is filed in the respective decimal places (counting from 1 upwards). The axis with in the place with the lowest value is referenciated firstly (units digits).

Example:

Order = 123

Note: When commissioning it is useful if you firstly enter only the axis that shall be referenciated firstly, and realizes with this axis a successful reference travel (e.g. Z-axis = 3. axis)

Order = 3

Then you add the following axis **before** this number (e.g. X-axis = 1. axis).

Order = 13

Continue this until all axes are successfully referenciated.

5.6.2 ReferenzSchalter (reference switch)

Indication of the axes, which are referenciated on the reference switch (bit coded).

5.6.3 Indexsuche=0 (Index search=0)

Indication of the axes, which are referenciated on the index pulse (bit coded).

5.6.4 Direction

Reference – search direction (bit coded). 1 causes a referenciation to the negative direction.

5.6.5 GotoZeroAfterRef

With this variable it is indicated if the respective axis shall travel to its. The default value is 0.

5.6.6 ReferenzPosX / Y / ...

Reference position in the reference point.

5.7 [FEHLERTEXTE] – Error texts

In this section the display texts for the error cases are defined. A few errors are already predefined by McuWIN. You can firstly copy this section into the file "FehlerTexte.INI". Further errors can be added application specifically. The error texts can be called with the variable C110 (bit coded).

5.7.1 User specific error texts

Section [FEHLERTEXTE] (error texts) section error number (decimal).
Here the text must be entered. This text must contain %X for the error number.

5.7.2 [FEHLERINFO] – Error info

Section variable number (decimal)

Here a CI-address must be entered that specifies one or two axes that caused the error. If two axes are to be indicated, the left axis must be entered *100 (e.g. 1213 – axis left in C112 and axis right C113)

For each axis that is specified, an error string is added automatically, in which the axis name is shown.

Example: Two axes are selected

[FEHLERTEXTE] – Error texts

1=Fehler # %i: Unknown function code!

2=Fehler # %i: Position error

[FEHLERINFO] – error info

1=0

2=14

5.8 [WARNTEXTE] – Warning texts

In this section warning and status texts are defined. These are bit coded and are be output when in C115 the respecting bit is set. With the text output the corresponding bit in C115 is removed.

If in addition the bit with the highest value is set in C115, the error window is deleted before the output.

5.8.1 User specific warning texts

Section [FEHLERTEXTE] (error texts) section error number (decimal)

Here a text must be entered. This text must contain %X for the warning / status number.

5.9 [TOOLCOMPENSATION] tool compensation

5.9.1 TcFileName

With this variable an Ini-File with information about the tool radius correction can be indicated. Sample data can be found in „ToolComp Demo.ini“.

5.10 [DDE]

With McuWIN you can communicate with a DDE master. Hereto “DDE communication” (“DDE Kommunikation”) must be observed. The settings for the DDE communication are realized in section DDE.

6 Variable assignment

6.1 CD variable

No.	Meaning	Program
0-17	Parameter of G, T or other codes The used variables are coded by bits in the system variable AXSEL	McuWIN, TASK0
20	TRAC – Default track acceleration	McuWIN, TASK1
21	TRVL – Default track acceleration	McuWIN, TASK1
23	Buffered value of override at Value saved at advance lock	TASK0, TASK1
24	Velocity factor for free traversing	TASK0
30	Start position value A1 for position monitoring	TASK1
31	Start position value A2 for position monitoring	TASK1
32	Start position value A3 for position monitoring	TASK1
33	Start position value A4 for position monitoring	TASK1
34	Start position value A5 for position monitoring	TASK1
35	Start position value A6 for position monitoring	TASK1
36	Start position value A7 for position monitoring	TASK1
37	Start position value A8 for position monitoring	TASK1
CD40	Value max. velocity jump {mdvel} for axis A1	TASK1, McuWIN
CD41	Value max. velocity jump {mdvel} for axis A2	TASK1, McuWIN
CD42	Value max. velocity jump {mdvel} for axis A3	TASK1, McuWIN
CD43	Value max. velocity jump {mdvel} for axis A4	TASK1, McuWIN
CD44	Value max. velocity jump {mdvel} for axis A5	TASK1, McuWIN
CD45	Value max. velocity jump {mdvel} for axis A6	TASK1, McuWIN
CD46	Value max. velocity jump {mdvel} for axis A7	TASK1, McuWIN
CD47	Value max. velocity jump {mdvel} for axis A8	TASK1, McuWIN
CD48	JerkRel	TASK1, McuWIN
CD50	Return currently set zero shift axis A1	TASK0
CD51	Return currently set zero shift axis A2	TASK0
CD52	Return currently set zero shift axis A3	TASK0
CD53	Return currently set zero shift axis A4	TASK0
CD54	Return currently set zero shift axis A5	TASK0
CD55	Return currently set zero shift axis A6	TASK0
CD56	Return currently set zero shift axis A7	TASK0
CD57	Return currently set zero shift axis A8	TASK0
90	Version of TASK0.SRC	TASK0
91	Version of TASK1.SRC	TASK1
94	Version of McuWIN.EXE	McuWIN
95	Version of RWMOS.ELF	TASK1

No.	Meaning	Program
600 to 699	Application specific	

6.2 CI variable

No.	Meaning	Program	
0	Command transfer for command interpreter G	McuWIN.EXE	x
1	Command transfer for command interpreter M	McuWIN.EXE	x
2	Command transfer for command interpreter Sonder (special)	McuWIN.EXE	x
9	Axes in the system are bit coded (reduced)	McuWIN.EXE T0	x
10	Error variable This variable is bit coded. The individual errors are defined in GCODE.INC.	McuWIN.EXE T0	x
11	Axes referenciated (bit coded)	McuWIN.EXE T0 T1	x
12	Marker: Record protocol data	T0 T1	x
13	Axes for EE-monitoring (bit coded) and for counter monitoring by index (bit 16 .. 23)	McuWIN.EXE T1	
14	(WW: MPE causing axis) Marker: Do not start Task3 after command execution	(McuWIN.EXE) McuWIN.EXE T0	x
15	Warning No. MSB not set means: Delete error window before display	MCUWIN.EXE T1	x
16	Number of the actually available axes after boot	McuWIN.EXE T1	x
17	Axis No. (bit coded) at: Axis at error reference travel Axis at error MPE Axis at configuration error Axis at encoder error Axis at counter error (detected by index)	T0 T1 McuWIN.EXE	x
18	Axis at ESL Hard + Soft (bit coded)	T1	x
19	Axis at ESR Hard + Soft (bit coded)	T1	x
20	Status of counter monitoring (bit coded)	T1	
21	Order of the reference travel (decimal coded)	McuWIN.EXE T0	x
22	Referenciation on reference switch? (bit coded)	McuWIN.EXE T0	x
23	Index search at reference travel? (bit coded)	McuWIN.EXE T0	x
24	Direction of the reference travel (bit coded)	McuWIN.EXE T0	x
25	Index of the X-axis (for ToolCompensation)	T1 McuWIN.EXE	x
26	Index of the Y-axis (for ToolCompensation)	T1 McuWIN.EXE	x
27	Index of the Z-axis (for ToolCompensation)	T1 McuWIN.EXE	x
28	Realize a customer specific reference travel, if <> 0	T1 McuWIN.EXE	
29	Traverse zero point after reference travel? (bit coded)	T1 McuWIN.EXE	

No.	Meaning	Program	
30	Indicates bit coded options of RWMOS.ELF Bit 0 = Resource-interface available Bit 1 = TC-interface available		
31	Status information for absolute stop or desired stop	McuWIN.EXE	x
32	TC-error word	TASK1.SRC	
33	Configuration word for MODEREG - S-profile - Look-Ahead		
34	Control generation MCUG2 = 2, MCUG3 (ADDIPOS) = 3	McuWIN.EXE All tasks according to requirements.	
35	LookAhead-depth	McuWIN.EXE	
36	Marker if the system is dereferenced after emergency stop	McuWIN.EXE TASK1.SRC	
37	Information if the program is running: 0 – Program is not running 1 – Program is running 2 – Program is running in the single step mode	McuWIN.EXE	
38			
39	Bits to traverse axes manually	T2 McuWIN.EXE	
40ff	Oelmaier CNC-7A: Output variable for AWS		
50ff	Oelmaier CNC-7A: Input variable for AWS		
50	At MBAWIN: Switch the ToolButton manually	MbaWIN.EXE	
60	Synchronisation variable for program start	McuWIN.EXE, TASK1.SRC	
61	Reference switch is latch input (bit coded)	McuWIN.EXE, TASK0.SRC	
62	No referenciation on limit switch (bit coded)	McuWIN.EXE, TASK0.SRC	
90	Specify version error: 1 = Version of Task 0 is too small for Task 1 2 = Version of McuWIN is too small for Task 1 6 = Version of RWMOS is too small for Task 1 11 = Version of Task 0 is too small for McuWIN.EXE 12 = Version of Task 1 is too small for McuWIN.EXE 13 = Version of RWMOS is too small for McuWIN.EXE		
100ff	DDE writing the status variable	McuWIN.EXE	
200ff	DDE reading the status variable	McuWIN.EXE	
500	At MBAWIN: Buttons Visible	MbaWIN.EXE	
501	At MBAWIN: Buttons Enable	MbaWIN.EXE	
600 to 699	Application specific		

7 Description of the operating modes

7.1 EditMode

This is the operation mode for editing programs. Program start is possible if the axes are referenciated (if this characteristic is not disabled).

In the EditMode the axes can be traversed manually by the TeachIn-window. The manual procedure is still possible with the CI-variable CI39.

Bit coding in CI39:

Bit	Axis	Direction
0	1	positive
1	1	negative
2	2	positive
3	2	negative
4	3	positive
5	3	negative
	etc.	

The description of CI39, depending on digital inputs, can be realized e.g. in Task2 of a closed loop.

7.2 RunMode

A program is realized in the automatic mode. Referenciation and new program start is not possible. The program also cannot be edited.

With "Stop" you can stop, with "Step" („Schritt“) you can switch to the StepMode and with "Reset" you can reset the control.

The RunMode is finished automatically when the UserTask and function task are fixed.

With the commands M00 and M01 you can switch into the Step-Mode. M01 is only working when the button "Desired stop" („Wahlweiser Halt) is activated.

7.3 StepMode

Single step mode, which finishes automatically, if after one step no new line is reached anymore.

7.4 HaltMode

Transition status between RunMode and EditMode.

7.5 ManMode

Manual procedure is active (Tech-In, reference travel). In this mode no automatic cycles may be started.

7.6 Reference travel

The sequence of the reference travel can be configured with the corresponding variables in McuWIN.INI. If the reference travel cannot be configured with the predetermined variation possibilities, in the file „KdRefFahrt.INC“ an application specific sequence can be programmed. In order to activate this, in the variable CI28 the value 1 must be entered (e.g. in the file „AppCommands.INC“).

The axes are always referenced sequentially in the order that is indicated in the variable „sequence“.

Firstly, the set values for Jog-velocity, Jog-acceleration, Home-velocity and Home-acceleration are checked. If here a error is detected (value ≤ 0) the error „conflict in the configuration data!“ („Konflikt in den Konfigurationsdaten!“) (20H) will be displayed.

If within the sequence an axis is indicated several times or if an axis is indicated, which is not available in the system, this error also is displayed. Then the axis will be travelled free by the limit switch, if necessary. Now the axis is started into the search direction, which is indicated in the variable „direction“ („Richtung“) in the bit corresponding with the axis. 1 means search in negative direction (with smaller position values).

7.6.1 Referenciation on reference switch

This variant is selected axis specific, if the bit corresponding with the axis is set in the variable „reference switch“. Then the axis is traversed in the selected search direction to the reference switch. As reference switch input the „fast latch input“ of the corresponding axis **must** be used (1. axis = I14, 2. axis= I15, 3. axis = I16).

7.6.2 Referenciation on limit switch

This variant is selected axis specific if the bit corresponding with the axis is not set in the variable „reference switch“. Then the axis is travelled on the hardware limit switch and is stopped by this and then released again.

7.6.3 Fine positioning on zero track

By setting the axis specific bit in the variable „index search“ („Indexsuche“) now a fine positioning to the zero track of the incremental encoder is realized.

7.6.4 Setting the home position

In the position that is found now, the home position is set on a value, which is indicated in the variable „ReferenzPos?“ for the corresponding axis. After all axes have been referenciated, the hardware and software limit switch monitoring is activated.

8 Description of the SAP task environment

8.1 Background information for the APCI-8001

G-Code programs according to DIN 66025 can be processed with the **APCI-8001** in a SAP-task. Hereto the programs must be compiled in a CNC-file, loaded onto the control and started. Here the commands that are available must be taken into consideration, which can be extended application specifically.

As described in the manual, four tasks for the processing of NC-programs are available for the user. G-Code programs have to be executed always in Task 3. Basically, for the handling the same conventions are valid as for SAP programs that were written in RW-SymPas. The compilation can be realized with the compiler NCC.EXE, with function calls in NCC.DLL or in the commissioning program MCFG.EXE for DOS or Windows-32 (from version 1.06). The respective compiler must be operated in the corresponding mode. Parallel to this task (Task 3) an application specific command interpreter should be loaded in Task 0, that is called from Task 3, if necessary. In Task 1 still an initialisation and monitoring program shall be started, which executes the allocation of default values, e.g. track acceleration, start values for CI- and CD-variable and the monitoring of the system status, e.g. handling of the limit switch, emergency stop handling, etc. A sample program for the G-Code programming is available under the name BEISPIEL.SRC.

8.1.1 Compiler mode for G-Code programs

8.1.1.1 Command lines Compiler NCC.EXE

The command line program NCC.EXE must be called with the parameter DIN.

Example:

NCC PrgName DIN

8.1.1.2 MCFG for MS-DOS

The program MCFG.EXE for DOS must be converted once to the DIN-mode. This is to be done as follows:

Start of the program with the parameter PLC

Select in the menu <<Editor>> <<Setup>> <<Set Compiler Mode>>
the option DIN-code by mouseclick.

(Switch off by selecting rw_SymPas)

This information will be stored in the file MCFG.DAT!

8.1.1.3 MCFG for 32bit-Windows

If an editor window is opened and activated, in the menu <<Compile>> <<Options>>, the option <<Select DIN 66025 Language>> can be selected. This information will not be stored and must be activated after each new call. After this action in the menu <<Trace>> the additional menu option <<Trace DIN66025 Task>> is available, with which corresponding programs can be compiled, loaded into Task 3 and started.

8.2 Supplementing of customer specific codes

8.2.1 G-codes

The compiler processes only several G-codes directly. G-codes that are not processed directly by the compiler, cause a calling of Task 0 with the transfer of parameters in CI-variables and, if necessary, in CD-variables. In Task 0, the corresponding functionality of the command can be realized, if this is not already provided by the control.

The G-codes listed below are processed directly by the operating system software RWMOS.ELF and do not cause the call of Task 0:

G00, G01, G02, G03, G04, G53 – G60, G7, G71, G90 – G94, G150, G151.

All other G-commands cause a calling of Task 0. Its functionality must be tested there. The commands

G17-G19, G21 – G24, G39 – G42, G50, G51, G74, G98, G99, G161 and G162

are programmed in Task0.SRC and can, if required, be reprogrammed user specifically. A few G-codes cause the execution of a movement contour with the support of spooler commands. These are the commands

G01, G02, G03, G17, G18, G19, G40, G41, G42, G90, G91, G200 - G299.

With all other G-commands a contour is finished and executed. When the user programs out these G-codes, it must be observed, that these commands are in fact assigned with spooler commands, especially at the functions that are still free

G200 to G299.

8.2.2 M-codes

Also a few M-codes are processed directly by RWMOS.ELF. These are the commands

M17, M26, M27, M80, M96, M97, M98.

All other m-commands cause the calling of Task 0. Its functionality must be tested there. The commands

M00 - M06, M08, M09, M30

are programmed in Task0.SRC and can, if required, be reprogrammed user specifically. The execution of a movement contour is not interrupted with the commands

M00, M01, M26, M27 und M200 – M299

When the user programs out these m-codes, it must be observed that these commands are in fact assigned with spooler commands, especially at the functions that are still free.

M200 to M299.

8.2.3 Calling conventions

During the execution of an unknown G-code, Task 0 is called with the G-code number in CI0. An allocation to X, Y or Z is transferred in CD0, CD1 or CD2. The axes, which were indicated at the calling of the command, are shown bit coded in the system variable AXSEL. So the parameter list can be checked. During the execution of an unknown m-code Task 0 is called with the m-code number CI1. During the execution of an unknown special command, Task 0 is called with the command number in CI2. A parameter is transferred in CD0.

After calling Task 0, the execution is stopped in Task 3. Task 0 must process the command that is now indicated. After the termination of the execution, the command must be confirmed and Task 3 must be continued with the command

contcnct (3);

The SRC-sample program still contains the following convention:

- If after the execution of the command Task 3 shall be started again, in CI14 the bit 1 must be reset.
- If after the execution of the command Task 3 shall not be started again, in CI14 the value 1 is entered. This value is always reset again after the execution of the command.

8.3 Task 0 – Command interpreter

In general, in Task 0 a task is executed that provides the functionality of G-codes, M-codes and of special functions, which cannot be handled directly from the compiler. Under the name TASK0.SRC there is a program available that can be amended application specifically.

8.3.1 Implementation of customer specific G-codes

When calling a G-code command, which is not directly processed from the control, a function call in Task 0 is executed. Here the number of the G-code is entered in CI0. Values that are indicated with the axis names, are entered in the common-double-variable 0.1,... whereas the index of the CD-variable corresponds with the axis index. The axes, which are entered in the calling command, are shown bit coded in the system variable AXSEL. After this the G-code task (Task 3) is stopped and Task 0 is started. So the amendment of the interpreter in Task 0 allows the amending of own G-codes. The user usually receives an adequately adapted version of the file TASK0.SRC according to the requirement of amendment. However, the user also can realize own amendments. Hereto please observe the following chapter.

Note: After reinstalling McuWIN, TASK0.SRC must be compiled newly if one of the following Includefiles contains customer specific amendments.

8.3.2 Include-Files in TASK0

8.3.2.1 KdRefFahrt.INC

The sequence of the reference travel is programmed in TASK0.SRC in the procedure "reference travel" („Referenzfahrt"). If this sequence shall be adapted customer specific, this sequence shall be programmed in the Includefile „KdRefFahrt.INC". Then this function must be activated by entering a value in CI28 <> 0.

This option can be set in the installation program IniCFG.EXE. Before McuWIN V2.5.3.24 this entry had to be done in AppTask2.SRC.

This file is kept at a reinstallation of McuWIN (new installation without previous deinstallation). So user specific amendments are also available after updates at this place.

8.3.2.2 AppGCodes.INC

In this file application specific G-codes are programmed out.

This file is kept at a reinstallation of McuWIN (new installation without previous deinstallation). So user specific amendments are also available after updates at this place.

8.3.2.3 AppCommands.INC

In this file application specific special commands, e.g. S-command, are programmed out. This file remains preserved at a reinstallation of McuWIN (new installation without previous deinstallation). So user specific amendments are also available after updates at this place.

8.3.2.4 AppMCodes.INC

In this file application specific m-codes are programmed out. This file is kept at a reinstallation of McuWIN (new installation without previous deinstallation). So user specific amendments are also available after updates at this place.

8.3.2.5 Application.INC

In this file user specific constants and variable declarations can be entered.

This file is kept at a reinstallation of McuWIN (new installation without previous deinstallation). Thus, user specific amendments are also available after updates at this place.

8.4 Task 1 – Initialisation and monitoring task

Task 1 is available for the execution of an initialisation and monitoring task. Hereto a sample program under the name TASK1.SRC is available. This task can be amended application specifically. Firstly, this task processes an initialisation list and then remains in a closed loop, in which cyclic monitoring functions are processed.

The user usually receives an adequately adapted version of the file TASK1.SRC, if wanted, according to the required amendments.

8.4.1 Initialisations in module TASK1.SRC

The following descriptions are only important for the system adaptation. The user of the program will not get in contact with these system files!

8.4.1.1 Unit for interpolation commands

Hereto a time or position unit is allocated to the system variables TU or PU (see „Programming Manual“ chapter 6.2.1.2)

8.4.1.2 Acceleration for inptpolation commands

In order to set the acceleration value for interpolation commands, a corresponding value is allocated to the system variable (see „Programming Manual“ chapter 6.3.1). The unit of this value is defined in the system variables PU and TU.

8.4.1.3 CI- and CD-variable

These are application specific variables, which can contain system variables, status and error variables. These variables can be read directly from a PC interface and evaluated. The following variables are used from the system, e.g. for parameter transfer at code-calls:

Common variables that are used for the parameter transfer:

Variable	Use
CI0	G-code No.
CI1	M-code No.
CI2	Special command No.
CI10	Error variable
CI11	Referenciated axes (bit coded)
CD0	Parameter for several commands
CD1	Parameter for several commands
CD2	Parameter for several commands
CD3..7	Parameter for several commands

8.4.2 Monitorings

The monitoring functions are executed in Task 1, which is always active. Individual monitorings are, if required, activated or deactivated by system statuses, (see e.g. software limit switch).

A few system statuses are monitored by the RWMOS-operating software and generate a corresponding event-routine, when occurring. These routines must be activated before the use (e.g. EVMPE). Hereto please see the descriptions in the „Programming Manual“ (chapter 6.4).

Caution: The task, in which events are managed, may not contain long Wait commands, as during a Wait, e.g. wt (2000) the corresponding task is inactive.

8.4.2.1 Position error

The axis specific max. position error is defined in the program mcfg. Exceeding this position error can be treated in the event procedure EVMPE.

8.4.2.2 Hardware limit switch

A hardware limit switch can be treated in the event procedure EVLSH.

8.4.2.3 Software limit switch

A software limit switch event can be treated in the event procedure EVLSS. This event firstly will be monitored after the command SHP at the respective axis, as at not referenciated axes the monitoring of a software limit switch normally is not useful.

8.4.2.4 Emergency stop

If an input is active, to which the function EO was allocated, will be branched out into the event processing routine EVEO.

This routine may process only indirectly an emergency stop event. Please observe the corresponding safety regulations for the direct effects of the emergency stop to the drives.

8.4.2.5 Gain ready

The event routine EVDNR is executed if an input is inactive, which was declared as DR.

8.4.2.6 Encoder-Error-Flag

This flag can be monitored cyclically for an increased system safety.

8.4.2.7 Encoder verification

With the index-latch-function the counter value of the encoder logic can be monitored cyclically.

8.4.2.8 Further monitoring functions

In this chapter further monitoring functions could be added. For this, please refer to the documentation of the **APCI-8001**.

9 Customer specific extensions

The source tasks of the CNC-task files are delivered with the program package and therefore can be used from the user for amendments and adaptations. But in order to guarantee the compatibility of standard applications with future updates, user specific adaptations may be limited to specific files. So it can be guaranteed that the updates of the most important files (TASK0.SRC und TASK1.SRC) can be activated at any time.

Please remember: In task 0 the program TASK0.SRC is executed. It contains a command interpreter in which G-codes, M-codes and further functions are programmed.

In task 1 the monitoring program TASK1.SRC is executed.

The user has the possibility to realize in the files described below customer specific extensions / adaptations

9.1 AppTask2.SRC

This file is loaded and executed in Task2. Here customer specific single initialisations and in a closed loop cyclic monitorings can be implemented.

9.2 AppGCode.INC

This is an Includefile for TASK0.SRC. In this file application specific G-codes can be programmed out.

9.3 AppMCodes.INC

This is an Includefile for TASK0.SRC. In this file application specific M-codes can be programmed out.

9.4 AppCommands.INC

This is an Includefile for TASK0.SRC. In this file application specific special command, e.g. S- or T-command can be programmed out.

10 Examples for application specific amendments

10.1 Coolant On/Off (M08 / M09)

With the command M08 the coolant shall be switched on. This is realized by using output 5 of the first axis channel. Hereto AppMCodes.INC is amended as follows:

```
// application specific m-commands
if (CI1 = 8) then begin          // M08 - Coolant On
    X.digob.5 := TRUE;          // Switch on output 5 of the X-axis
end else if (CI1 = 9) then begin // M09 - Coolant Off
    X.digob.5 := FALSE;         // Switch off output 5 of the X-axis
end else
```

German original version :

```
// Applikationsspezifische M-Befehle
if (CI1 = 8) then begin          // M08 - Kühlmittel Ein
    X.digob.5 := TRUE;          // Ausgang 5 von X-Achse einschalten
end else if (CI1 = 9) then begin // M09 - Kühlmittel Aus
    X.digob.5 := FALSE;         // Ausgang 5 von X-Achse ausschalten
end else
```

10.2 Main spindle

With the S-command the spindle speed shall be set. Here a minimum and a maximum value shall be monitored. The output of the spindle speed is realized through the analog output of the fourth axis channel (H). With the command M03 the spindle with run to the right shall be switched on. This is realized through output 6 = EIN (Spindle On) and output 7 = 0 (run to the right). With M04 the spindle with run to the left shall be switched on. This is realized through output 6 = EIN (spindle On) and output 7 = 1 (run to the left). With M05 the spindle shall be switched off.

Here the reaching of the spindle speed is not monitored. Switching from run to the right to run to the left or reverse is not possible. When something like this is called, an error is issued with 00010000hex.

Hereto AppMCodes.INC is amended as follows:

In order to output with the S-command analog values, the AppCommands.INC is amended as follows:

```
if (CI2 = 1) then begin          // S-command: Analog value at main spindle
```

```

//issue (axis H)
// Firstly, checking the range of the analog value
// Minimum value can be found in CI601
// Maximum value can be found in CI602
if CD0 < CI601 then begin
    CI10 := CI10 or $00020000;    // Indicate error in CI10
end;
if CD0 > CI602 then begin
    CI10 := CI10 or $00020000;    // Indicate error in CI10
end;
if CI10 = 0 then begin // If there is no error
    SSMSIW;                // Execute previous interpolation commands
    H.mcp := integer (CD0);    // Issue spindle speed on
                                // analog output
end;
end else

```

German original version:

```

if (CI2 = 1) then begin          // S-Kommando: Analogwert an Hauptspindel
                                //(Achse H) ausgeben
    // Zunächst Bereich des Analogwertes überprüfen
    // Minimalwert steht in CI601
    // Maximalwert steht in CI602
    if CD0 < CI601 then begin
        CI10 := CI10 or $00020000;    // Fehler in CI10 anzeigen
    end;
    if CD0 > CI602 then begin
        CI10 := CI10 or $00020000;    // Fehler in CI10 anzeigen
    end;
    if CI10 = 0 then begin // Wenn kein Fehler ansteht
        SSMSIW;                // vorhergehende Interpolationskommandos ausführen
        H.mcp := integer (CD0);    // Spindeldrehzahl auf
                                    // Analogausgang ausgeben
    end;
end else

```

11 DDE-communication

The program McuWIN can exchange data with a DDE-server application as DDE-client. In order to activate this functionality, the following values must be defined in the file McuWIN.INI:

Section [DDE]

DDE Server- and Topic-name:

ServerName=

TopicName=

Items for sending data (E0.? = examples)

OUT00=E0.0

OUT01=E0.1

OUT02=E0.2

...

Items for receiving data (A0.? = examples)

IN00=A0.0

IN01=A0.1

IN02=A0.2

...

The definition of IN and OUT must begin at 00 (zero and not the letter O) and be continuous. The data to be sent must be allocated to the common-integer-variables CI100 and the following. You can access to the data to be received with the die common-integer-variable CI200 and the following.

The validity of the DDE-channel is displayed in the status window of McuWIN.

At the moment the DDE-channels are provided with a clock rate of 20 ms.

12 Error diagnostics

- **When starting a program after the compilation process Error 21 is displayed:**
In this case the compiled program is too big for the main memory of the user task (TASK 3). In this case the system variable SZTSK3 of the control must be set to an adequate value. This value will be indicated in bytes and can have a few MB. Default is 100000.
How to set environment variable is described in the chapter „Environment variable and control hardware“ of the „Commissioning manual“.
- **When compiling Task1.SRC in mcfg Error 89 is displayed:**
In order to compile this file, the option „Full System“ must be set.
- **At the reference travel, the axis drives cyclically to the reference switch and again away:**
As reference switch input not the axis specific input for position latch is used (see chapter “hardware preconditions” / reference switch). The reference switch of the corresponding axes must be wired differently or the reference travel routine TASK0.SRC must be changed.
- **Task 0 or Task 1 cannot be compiled:**
In the first source text line appears the error message: Error 4, Duplicate Identifier.
The cause for this is that a reserved system name is used as axis name. Check of all axis names.
- **During the referenciation the error „Conflict in the configuration data!“ (20hex) occurs:**
The cause for this can be:
 - at least one value of the JOG-velocities or accelerations is initialised with 0 (jvl, jac, hvl, hac)
 - in the Ini-File in the section „reference travel“ [REFERENZFAHRT] under the value order one axis is indicated several times under the value order
 - in the Ini-File in the section “reference travel” [REFERENZFAHRT] under the value order an axis is indicated that is either not available on the APCI-8001 or that is in the File GCODE.INC in the constant BitAll not indicated.
- **Other causes for the error „Conflict in the configuration data!“ (20hex):**
 - in GCode.INC the axis allocations are not correct.
(NrXAxis, NrYAxis, NrZAxis)
- **Status # 1000: Runtime error in SAP Task!**
This error shows an error in the programming of the APCI-8001 task environment. So, it is possible that a program option is used, which is not supported by the current RWMOS-operating software. Using the following procedure, this error can be localised:
Calling mcfg.exe and opening of a window „Cnc Task Status“, e.g. check with Shift-F5 if there is a runtime error at a task. If yes, here the error line and the program file can be detected.