
POSITIONIER- UND BAHNSTEUERUNG APCI-8001 und APCI-8008

G-Code-Interface

Stand: 01.06.2015 McuWIN V 2.5.3.133
Rev. 10/052022

www.addi-data.de

1	Vorgehensweise zur Verarbeitung von G-Code-Programmdateien	7
1.1	Die Bedienoberfläche McuWIN	7
2	Funktionalität von McuWIN	9
2.1	Not-Aus Überwachung	9
2.2	Maximaler Schleppfehler.....	9
2.3	Hardware Endschalter.....	9
2.4	Software Endschalter	9
2.5	Überwachung des Encoder-Error-Flags	10
2.6	Überwachung der Positionszähler	10
2.7	Werkzeug-Radius-Korrektur.....	10
2.8	Spindelsteigungsfehler- und Winkelfehler-Kompensation	11
3	Beschreibung der realisierten G-Codes	12
3.1	G00 - Positionieren im Eilgang.....	12
3.2	G01 - Linearinterpolation.....	12
3.3	G02 – Kreis- / Helixinterpolation	12
3.4	G03 – Kreis- / Helixinterpolation	12
3.5	G04 - Verweilzeit.....	13
3.6	G17 - Ebenenauswahl.....	13
3.7	G18 - Ebenenauswahl.....	13
3.8	G19 - Ebenenauswahl.....	13
3.9	G21 – Spiegeln der Y-Achse.....	13
3.10	G22 – Spiegeln der X-Achse.....	13
3.11	G23 – Spiegeln der X- und der Y-Achse.....	13
3.12	G24 – Spiegeln aller Achsen deaktivieren	14
3.13	G39 – Positions-Faktor programmieren	14
3.14	G40 – Werkzeugradiuskorrektur aus	14
3.15	G41 – Werkzeugradiuskorrektur links	14
3.16	G42 – Werkzeugradiuskorrektur rechts	14
3.17	G51 – Effektiv-Radius programmieren.....	14
3.18	G53 – Nullpunktverschiebung ausschalten.....	15
3.19	G54..G58 – Nullpunktverschiebung setzen	15
3.20	G60 – Interpolationsachsen definieren	15
3.21	G70 – Maßangaben in inch.....	16
3.22	G71 – Maßangaben in mm	16
3.23	G74 – Referenzfahrt.....	16
3.24	G90 – Absolutmaßsystem.....	16
3.25	G91 – Relativmaßsystem.....	16
3.26	G92 – Nullpunktverschiebung setzen	16
3.27	G93 – Zeitreziproke Vorschubverschlüsselung	17
3.28	G94 – Zeiteinheit in Minute	17
3.29	G98 – Position für Softwareendschalter-Links setzen	17
3.30	G99 – Position für Softwareendschalter-Rechts setzen	17

3.31	G150 – Spline aus	17
3.32	G151 – Spline ein	17
3.33	G153 – Nullpunktverschiebung lesen	18
3.34	G154 – gelesene Nullpunktverschiebung reprogrammieren	18
3.35	G161 – Kreismittelpunktskoordinaten relativ oder absolut	18
3.36	G162 – Kreismittelpunktskoordinaten immer relativ	18

4 M-Codes19

4.1	M00 – Programm Halt	19
4.2	M01 – Wahlweiser Halt	19
4.3	M02 – Programm Ende	19
4.4	M03 – Spindel Rechtslauf	19
4.5	M04 – Spindel Linkslauf	19
4.6	M05 – Spindel Stop	20
4.7	M06 – Werkzeugwechsel	20
4.8	M08 – Kühlmittel Ein	20
4.9	M09 – Kühlmittel Aus	20
4.10	M17 – Unterprogramm Ende	20
4.11	M26 – Ausgang setzen	20
4.12	M27 – Ausgang rücksetzen	21
4.13	M30 – Programm Halt	21
4.14	M80 – Ausgänge beschreiben (wortweise)	21
4.15	M96 – Unbedingter Sprung	22
4.16	M98 – Unterprogramm aufrufen	22
4.17	M100 – Ausgang „Programm-Ende“ zurücksetzen	22
4.18	M150 – Aufzeichnung für grafische Systemanalyse starten	22
4.19	M901 – Common-Integer-Variable resident speichern	23
4.20	M902 – Resident gespeicherte Common-Integer-Variable rücklesen	23
4.21	M903 – Common-Double-Variable resident speichern	23
4.22	M904 – Resident gespeicherte Common-Double-Variable rücklesen	23
4.23	M910 – PC nach Programm-Ende ausschalten	24
4.24	M911 – McuWIN.exe nach Programm-Ende beenden	24

5 Sonstige Codes25

5.1	Labels	25
5.2	Realisierte Sonderfunktionen	25
5.3	F-Kommando	25
5.4	S-Kommando	26
5.5	T-Kommando	26
5.6	D-Kommando	26
5.7	String Ausgabe	26
5.7.1	WRITE Kommando	27
5.7.2	WRITELN Kommando	27

6 Kommentare28

7 Bedingte Programmausführung29

8 Schleifen	30
8.1 while-do Schleife	30
8.2 repeat-until Schleife	30
8.3 for Schleife	30
9 Einbinden von Include-Files	31
10 Einbindung von rw_SymPas Kommandos.....	32
11 Rechenparameter.....	33
12 Variable	34
13 Handhabung von McuWIN.....	35
13.1 Problembehandlung	35
13.1.1 Fehleranzeigen von McuWIN.....	35
13.1.1.1 Fehler # 1: Unbekannter Funktionscode!	35
13.1.1.2 Warnung # 1: Enkoder-Error!	35
13.1.1.3 Fehler # 2: Schleppfehler (Achse ?)!.....	35
13.1.1.4 Status # 2: Warte bis Tür geschlossen!	35
13.1.1.5 Fehler # 4: Hardware-Endschalter (links Achse ? / rechts Achse ?)!.....	35
13.1.1.6 Status # 4: Programm wurde gestartet!	36
13.1.1.7 Fehler # 8: Software-Endschalter (links Achse ? / rechts Achse ?)!	36
13.1.1.8 Status # 8: Programm wurde beendet!	36
13.1.1.9 Status # 10: Not-Aus	36
13.1.1.10 Warnung # 10: Positionsfehler erkannt!	36
13.1.1.11 Fehler # 20: Konflikt in den Konfigurationsdaten!	36
13.1.1.12 Status # 20: Referenzierung wurde gestartet!.....	36
13.1.1.13 Fehler 21 beim Laden von <<???>>	36
13.1.1.14 Fehler # 40: Fehler bei Referenzfahrt (Achse ?)!.....	37
13.1.1.15 Status # 40: Warten bis Spindeldrehzahl erreicht.....	37
13.1.1.16 Fehler # 80: Versionskonflikt McuWIN - SAP!.....	37
13.1.1.17 Status # 80: Fehlerbit in ErrorReg erkannt! ? hex.....	37
13.1.1.18 Fehler # 100: Konfigurationsfehler!	37
13.1.1.19 Status # 100: Fehlerbit in IFS-Register erkannt! ?? hex.....	37
13.1.1.20 Fehler 200: Fehler bei Werkzeug-Radius-Korrektur!	37
13.1.1.21 Status # 400: Vorschub Sperre Aktiv (UI)!	37

13.1.1.22	Fehler # 800: Verstärker nicht bereit!	38
13.1.1.23	Status # 1000: Laufzeitfehler in SAP-Task!	38
13.1.1.24	Status # 2000: Parameterfehler zur Laufzeit!.....	38
13.1.1.25	Fehler # 4000: Ungültiger S-Wert!.....	38
13.1.1.26	Fehler # 80000000: System wurde angehalten! Reset erforderlich	38

14 Versionsinformation39

14.1	Änderungen in früheren Versionen	39
------	--	----

1 Vorgehensweise zur Verarbeitung von G-Code-Programmdateien

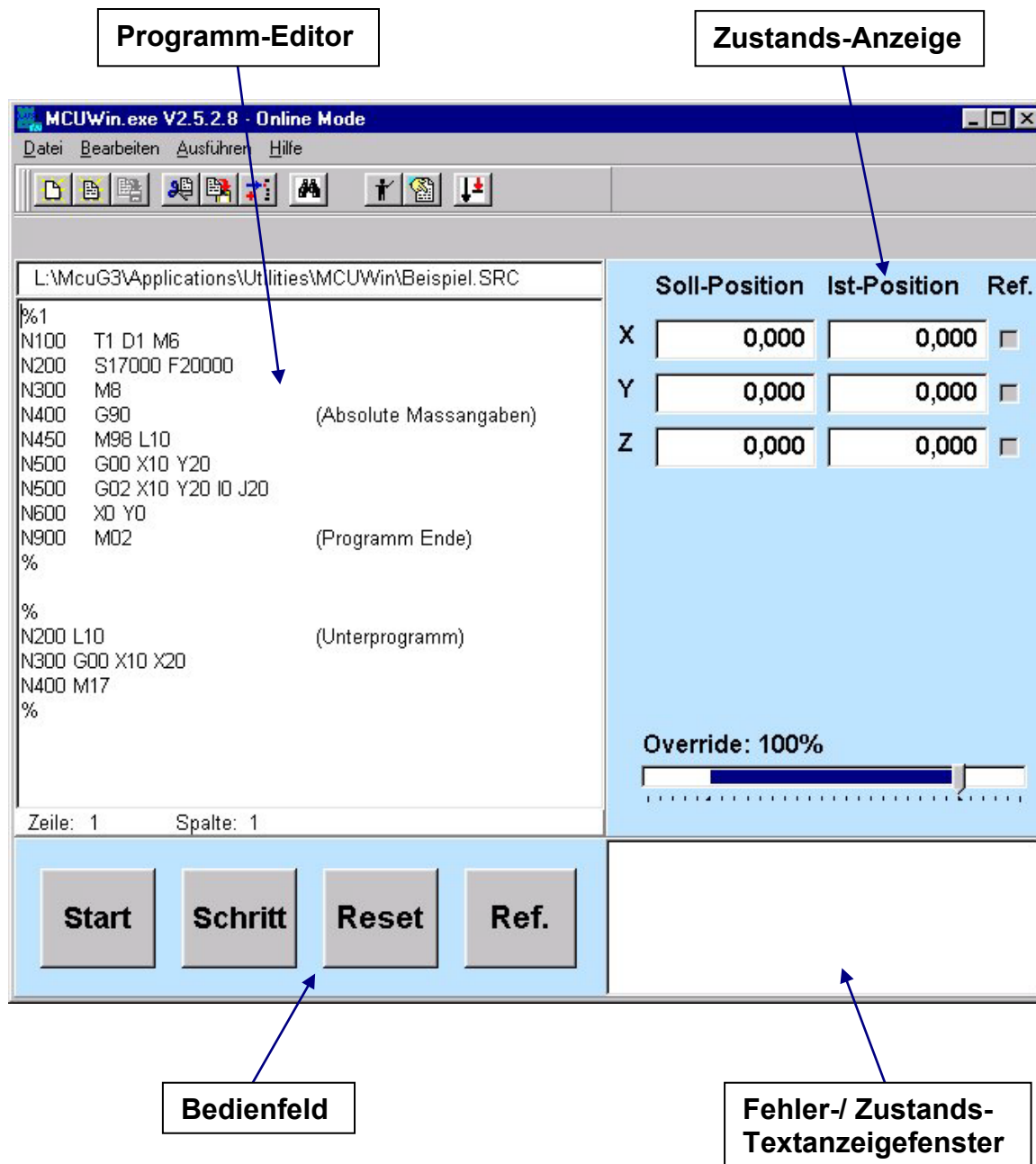
1.1 Die Bedienoberfläche McuWIN

Das Windows-Applikationsprogramm McuWIN.EXE stellt dem Anwender eine Bedienoberfläche zur Ausführung von G-Code-Programmen bereitstellt. In dieser Oberfläche können Programme editiert, einer Syntaxprüfung unterzogen und ausgeführt werden.

Weiterhin werden Sollpositionen, Istpositionen, Referenzstatus und Fehlermeldungen angezeigt. Die erforderlichen Initialisierungen der Steuerung werden von diesem Programm automatisch durchgeführt. Optional können auch Programme im Einzelschrittbetrieb abgearbeitet werden.

Ergänzende Informationen zur Abarbeitung von G-Code-Programmen werden im File „McuWIN-Bedienoberfläche“ zur Verfügung gestellt. Diese sind hilfreich, wenn McuWIN nicht verwendet werden soll, weil eine eigene Programmieroberfläche eingesetzt werden soll oder wenn Ergänzungen nötig sind.

Hinweis: Die Schriftgröße im Editor kann durch Drehen des Mausekkrads bei gedrückter linker Maustaste verändert werden.

Bild: Ansicht der Bedieneroberfläche McuWIN

2 Funktionalität von McuWIN

2.1 Not-Aus Überwachung

Bei der Einrichtung der Achsen mit dem Konfigurationsprogramm mcfg können digitale Eingänge als Not-Aus-Eingänge konfiguriert werden. Falls ein Not-Aus-Eingang aktiviert wird, werden laufende Programme unterbrochen, die Referenzierung wird, je nach Konfiguration aufgehoben, die Regelkreise werden geöffnet und der Not-Aus-Zustand wird angezeigt.

Sobald die Not-Aus-Bedingung wieder beseitigt ist, werden die Regelkreise wieder geschlossen. Nun kann die Referenzfahrt bzw. ein G-Code-Programm wieder gestartet werden. Dadurch kann der Betrieb nach dem Not-Aus wieder kontrolliert aufgenommen werden.

Vorsicht: Das Not-Aus Signal **muss** auf jeden Fall direkt auf die Antriebe wirken (Zwischenkreis-Spannung). Hier sind die einschlägigen Sicherheitsvorschriften zu beachten.

2.2 Maximaler Schleppfehler

Bei der Einrichtung der Achsen mit dem Konfigurationsprogramm mcfg kann ein achsspezifischer „Maximaler Schleppfehler“ angegeben werden. Wenn dieser bei geschlossenem Regelkreis überschritten wird, wird das Programm beendet und die Fehlerursache angezeigt.

2.3 Hardware Endschalter

Bei der Einrichtung der Achsen mit dem Konfigurationsprogramm mcfg können digitale Eingänge als Hardware-Endschalter konfiguriert werden. Beim Ansprechen eines Endschalters wird das Programm beendet und die Fehlerursache angezeigt. Die Überwachung durch McuWIN der Hardware-Endschalter wird erst aktiv nach erfolgreicher Referenzfahrt. Die Endschalter wirken jedoch immer auf Verfahrprofile mit der vorgegebenen Option.

2.4 Software Endschalter

Bei der Einrichtung der Achsen mit dem Konfigurationsprogramm mcfg können achsspezifisch Software-Endschalter konfiguriert werden. Beim Ansprechen eines Endschalters wird das Programm beendet und die Fehlerursache angezeigt. Die Überwachung durch McuWIN der Software-Endschalter wird erst aktiv nach erfolgreicher Referenzfahrt. Die Endschalter wirken auch erst nach der Referenzierung auf Verfahrprofile mit der vorgegebenen Option.

2.5 Überwachung des Encoder-Error-Flags

Mit Hilfe dieser Funktionalität ist es möglich, Probleme bei der Positionserfassung bei Inkrementalgebersystemen rechtzeitig zu erkennen. Diese Überwachung kann achsspezifisch aktiviert werden. Wenn ein Problem erkannt wird, wird eine Warnung an die Oberfläche ausgegeben.

2.6 Überwachung der Positionszähler

Diese Überwachung kann ebenfalls achsspezifisch aktiviert werden. Hierbei wird bei jedem Null-Impuls die Position verifiziert. Somit kann sicher erkannt werden, wenn sich das System verzählt.

Diese Option kann nur genutzt werden, wenn:

- die Überwachung des Encoder-Flags aktiviert ist,
- Inkrementalgebersysteme verwendet werden,
- die Nennereinheit der Systemgröße slsp mit deg angegeben wurde,
- Servo-Achse (keine Stepper) verwendet werden

Wenn ein Problem erkannt wird, wird eine Warnung an die Oberfläche ausgegeben.

2.7 Werkzeug-Radius-Korrektur

Die Werkzeug-Radius-Korrektur unterstützt die Befehle G01, G02 und G03. Kreise müssen in der jeweiligen Hauptebene liegen. Die Korrekturdaten müssen in einer INI-Datei abgelegt sein. Der Dateiname kann im Konfigurations-Programm IniCFG auf der Registerkarte "Dateien" gewählt werden. **ToolComp.INI** ist der Standardwert für diesen Dateinamen.

Diese Datei benötigt einen fest vorgeschriebenen Aufbau. Für jede Korrekturebene existiert eine Sektion, die nach folgendem Muster gekennzeichnet werden muss:

[ToolsXY]

Die beiden Buchstaben am Ende, geben die Korrekturebene an. Weitere Möglichkeiten sind:

[ToolsYZ]

[ToolsZX]

Die Buchstaben X, Y und Z sind fix und unabhängig von evtl. gewählten anderen Achsnamen.

Nach dem Bezeichner der Sektion erfolgt für jeden Werkzeugindex ein Maß für die Radius- und für die Längenkorrektur, z.B. in folgender Form:

[ToolsXY]

R0=0.0

L0=0.0

R1=5.0

L1=2.0

Hinweis: Die Werkzeugradius-Korrekturtablette wird nach jedem Neustart von McuWIN oder nach einem Reboot der Steuerung aus der Konfigurationsdatei geladen. Die Bearbeitung dieser Datei kann mit einem normalen Texteditor oder mit dem Zusatzprogramm **ToolEdit.exe** erfolgen. Wenn keine Werkzeugradius-Korrekturtablette existiert, dann wird die Funktionalität der Werkzeugradiuskorrektur nicht verwendet. In diesem Fall wird auch keine Zuordnung der Kreisebenen durchgeführt. Die Anwahl eines Werkzeugindexes oder anderer Funktionen der Werkzeugradiuskorrektur sind in diesem Fall nicht möglich. Die Bereitschaft der Werkzeugradiuskorrektur wird bit-codiert in CI30 angezeigt.

2.8 Spindelsteigungsfehler- und Winkelfehler-Kompensation

McuWIN stellt dem Anwender die Möglichkeiten zur Kompensation von Spindelsteigungsfehlern und Winkelfehlern bei kartesischer Achsanordnung zur Verfügung. Hierzu muss das McuWIN-Verzeichnis mit einfachen Textdateien ergänzt werden, welche die Kompensationstabellen enthalten. Aktiviert wird die Kompensation durch das Vorhandensein der entsprechenden Dateien. Hinweise zur Erstellung dieser Dateien sind im Handbuch „McuWIN-Bedienoberfläche“ zu finden.

3 Beschreibung der realisierten G-Codes

3.1 G00 - Positionieren im Eilgang

Die Übergabe der Zielpositionen erfolgt in Absolut- oder Relativkoordinaten, je nach angewähltem Modus (G90 / G91). Geschwindigkeit und Beschleunigung sind die in MCFG angegebenen achsspezifischen Standardwerte (File – System Data – Registerkarte "Motion Parameters" – {jvl} und {jac}) bzw. die in der Initialisierungstask gesetzten Werte. Die Positionseinheiten für G00 sind immer die in mcfg gesetzten, achsspezifischen Einheiten, unabhängig von der Wahl der Interpolationseinheit.

Dieser Befehl ist selbsthaltend, d.h. G00 braucht nicht in jeder Folgezeile angegeben zu werden. Nach Sprungzielen gilt als selbstgehaltener Befehl, der Befehl, der im Quelltext zuletzt vor der entsprechenden Zeile erscheint, nicht derjenige, der zuletzt ausgeführt wurde.

Beispiel:

```
N0400 G00 X20 Y22  
N0410 X30 Y23 Z-5
```

3.2 G01 - Linearinterpolation

Ausführung einer Linearinterpolation mit allen Positionierachsen: die Übergabe der Zielpositionen erfolgt in Relativ- oder Absolutkoordinaten, je nach angewähltem Modus. Die Bahngeschwindigkeit kann z.B. mit dem F-Befehl definiert werden. Die Programmierung der Beschleunigung erfolgt in der Initialisierungsroutine einer SAP-Task (0,1 oder 2).

Dieser Befehl ist selbsthaltend.

3.3 G02 – Kreis- / Helixinterpolation

Kreis- bzw. Helixinterpolation im Uhrzeigersinn: die Übergabe der Zielpositionen erfolgt in Relativ- oder Absolutkoordinaten, je nach angewähltem Modus. Die Bahngeschwindigkeit kann z.B. mit dem F-Befehl definiert werden. Die Programmierung der Beschleunigung erfolgt in der Initialisierungsroutine einer SAP-Task (0,1 oder 2). Als Parameter werden die Zielkoordinaten des Kreisbogens angegeben. Die erforderlichen Mittelpunktskordinaten werden in den Parametern I und J angegeben. Die Mittelpunktskordinaten können auch im Absolutmodus (G90) als Relativkoordinaten übergeben werden. Diese Option kann in der Konfigurationsoberfläche angewählt werden.

Der Befehl G02 ist selbsthaltend.

3.4 G03 – Kreis- / Helixinterpolation

Kreisinterpolation im Gegen-Uhrzeigersinn: sonst wie G02

3.5 G04 - Verweilzeit

Verweilzeit in Sekunden. Hier können Gleitpunkt Zahlenwerte, Gleitpunkt-Variable oder Berechnungsausdrücke angegeben werden.

Beispiele:

```
N100 G04 3  
N100 G04 CD601
```

Vorsicht: Mit G04 wird eine eventuelle Selbsthaltung von G-Befehlen aufgehoben.

3.6 G17 - Ebenenauswahl

Ebene X-Y für Kreisinterpolation und Werkzeugradiuskorrektur auswählen.
Die Ebene X-Y ist der Defaultwert nach dem Systemstart.

3.7 G18 - Ebenenauswahl

Ebene Z-X für Kreisinterpolation und Werkzeugradiuskorrektur auswählen.

3.8 G19 - Ebenenauswahl

Ebene Y-Z für Kreisinterpolation und Werkzeugradiuskorrektur auswählen.

3.9 G21 – Spiegeln der Y-Achse

Die für die Y-Achse programmierten Koordinaten werden am Nullpunkt gespiegelt. Eine bereits aktive Spiegelung der X-Achse wird von diesem Befehl nicht beeinflusst.

3.10 G22 – Spiegeln der X-Achse

Die für die X-Achse programmierten Koordinaten werden am Nullpunkt gespiegelt. Eine bereits aktive Spiegelung der Y-Achse wird von diesem Befehl nicht beeinflusst.

3.11 G23 – Spiegeln der X- und der Y-Achse

Die für die X- und Y-Achse programmierten Koordinaten werden am Nullpunkt gespiegelt.

3.12 G24 – Spiegeln aller Achsen deaktivieren

Alle programmierten Spiegelfunktionen (G21, G22, G23) werden ausgeschaltet.

3.13 G39 – Positions-Faktor programmieren

Für jede gespiegelte Achse im System kann ein Positionsfaktor programmiert werden. Bei angewählter Achsspiegelung, werden die Positionswerte der entsprechenden Achse mit diesem Faktor multipliziert. Standardwert für diesen Faktor ist -1 (Achsspiegelung). Das Kommando G39 ist also nur im Zusammenhang mit einer angewählten Achsspiegelung (G21, G22 oder G23) wirksam.

Beispiel:

N200 G39 X2 Y2

Die Konturgröße wird bei aktivem G23 verdoppelt.

N500 G39 (Rücksetzen aller Positions-Faktoren auf -1)

3.14 G40 – Werkzeugradiuskorrektur aus

Werkzeugradiuskorrektur ausschalten.

3.15 G41 – Werkzeugradiuskorrektur links

Werkzeugradiuskorrektur links einschalten.

3.16 G42 – Werkzeugradiuskorrektur rechts

Werkzeugradiuskorrektur rechts einschalten.

3.17 G51 – Effektiv-Radius programmieren

Mit G51 kann ein achsspezifischer Radius in der angegebenen Interpolations-Wegeinheit programmiert werden. Dieser wird zur Interpolationsberechnung herangezogen, wenn rotatorische Achse an translatorischen Linear-Interpolationsfahrten teilnehmen. Somit wird z.B. die Bearbeitung der Mantelfläche eines zylindrischen Werkstücks ermöglicht, welches mit einer rotatorischen Achse zur Bearbeitung gedreht wird. Die Maßeinheit der Positionswerte bei rotatorischen Achsen ist in diesem Fall keine rotatorische Größe, sondern die Interpolationseinheit. Diese Größe wird dann über den angegebenen Radius auf den entsprechenden, zu verfahrenen Winkel umgerechnet.

Beispiel:

N200 G51 C120 D40

Für diese Funktionalität sind die Versionsinformationen in Kapitel 14 zu beachten. Mindestversion ab 16.07.2003 ist erforderlich.

3.18 G53 – Nullpunktverschiebung ausschalten

Nullpunktverschiebung ausschalten bei allen Interpolationsachsen. Dieser Befehl ist kein Spoolerbefehl und beendet somit eine zusammenhängende Kontur, die mit Spoolerbefehlen, wie z.B. mit G01 programmiert wird. Falls bei anderen Achsen als den Interpolations-Achsen die Nullpunktverschiebung ausgeschaltet werden muss, kann z.B. der Befehl G92 im Absolutmodus verwendet werden.

Beispiel: N0100 G53

Diese Anweisung ist gleichwertig wie folgende Befehlssequenz (bei einem 3-Achsen System):

```
N0100 G90          ' Absolute Weginformation
N0101 G92 X0 Y0 Z0 ' Nullpunktverschiebung bei X und Y auf 0 setzen
```

3.19 G54..G58 – Nullpunktverschiebung setzen

Mit diesem Befehl wird die Nullpunktverschiebung aller Interpolationsachsen auf die in der entsprechenden Nullpunkt-Verschiebungstabelle stehenden Werte gesetzt. Die Nullpunktverschiebungstabelle ist in den Systemdaten abgelegt und kann mit Hilfe des Inbetriebnahmeprogramms mcfg editiert werden (Zero Offset). Hierbei wird mit G54 der Satz 0, G55 der Satz 1 usw. angewählt.

Für ein freies Setzen der Nullpunktverschiebung steht der Befehl G92 zur Verfügung. Dieser Befehl ist kein Spoolerbefehl und beendet somit eine zusammenhängende Kontur, die mit Spoolerbefehlen, wie z.B. mit G01 programmiert wird.

3.20 G60 – Interpolationsachsen definieren

Auswahl der an Interpolationsbefehlen (G01, G02, G03) teilnehmenden Achsen.

Beispiel:

G60 X Y Z

Mit diesem Befehl werden die Achsen definiert, die beim Verfahren mit den o.g. Befehlen im Interpolationszusammenhang stehen sollen. Diese Angabe dient der Initialisierung und wird i.A. nur einmal am Programmanfang aufgerufen.

Falls die Interpolationsachsen während des Programmablaufs geändert werden sollen, kann dies auch in einer rw_SymPas Task, z.B. in Task 0 im Rahmen eines M- oder G-Befehls geändert werden. Hierzu sind die niederwertigsten Bits der Systemvariablen **IPOLMODE** zu beschreiben. Bitte beachten Sie hierzu die entsprechende Dokumentation „McuWIN-Bedienoberfläche“.

3.21 G70 – Maßangaben in inch

Maßangaben für Positionen und Geschwindigkeiten in inch.

3.22 G71 – Maßangaben in mm

Maßangaben für Positionen und Geschwindigkeiten in mm (Defaultwert).

3.23 G74 – Referenzfahrt

Das Kommando G74 soll die Achsreferenzierung auslösen. Diese Funktion muss applikationsspezifisch in Task 0 programmiert / angepasst werden.

3.24 G90 – Absolutmaßsystem

Absolutmaßsystem auswählen.

3.25 G91 – Relativmaßsystem

Relativmaßsystem auswählen.

3.26 G92 – Nullpunktverschiebung setzen

Nullpunktverschiebung bei den angegebenen Achsen frei programmieren. Dieser Befehl ist kein Spoolerbefehl und beendet somit eine zusammenhängende Kontur, die mit Spoolerbefehlen, wie z.B. mit G01 programmiert wird. Dieser Befehl ist selbsthaltend.

Beispiel: N0100 G92 X100 Y-200

Ist G90 programmiert so wird die Nullpunktverschiebung auf den absoluten Nullpunkt bezogen. Ist G91 programmiert, so wird die Nullpunktverschiebung relativ zum derzeit gesetzten Nullpunkt verschoben.

Ein Aufruf von G53 setzt die Nullpunktverschiebung zurück, jedoch nur bei den selektierten Achsen (Interpolationsachsen). Ein Aufruf von G54 – G58 hebt eine mit G92 gesetzte Nullpunktverschiebung auf und setzt eine Nullpunktverschiebung laut Tabelle.

Ein Aufruf von G92 ohne Verfahrbefehl bewirkt keine Bewegung der Achsen.

In rw_SymPas kann auf die aktuell gesetzte Nullpunktverschiebung mit Hilfe des Achsenqualifizierers **ZEROOFFSET** zugegriffen werden.

3.27 G93 – Zeitreziproke Vorschubverschlüsselung

Zeitreziproke Vorschubverschlüsselung. Diese Funktion ist noch nicht implementiert und darf nicht verwendet werden.

3.28 G94 – Zeiteinheit in Minute

Angabe der Zeiteinheit in Minute, d.h. Einheit des Vorschubs in [mm/min] bzw. [inch/min].

G96 – Konstante Schnittgeschwindigkeit aktivieren

G97 – Konstante Schnittgeschwindigkeit aufheben

3.29 G98 – Position für Softwareendschalter-Links setzen

Die Softwareendschalter werden durch dieses Kommando nicht aktiviert. Die Aktivierung erfolgt automatisch bei der Referenzierung der jeweiligen Achse, soweit diese Funktionalität in mcfg.exe überhaupt definiert ist. Wenn dieses Kommando nicht verwendet wird, ist die Endschalterposition mit der Defaultposition initialisiert, die in mcfg.exe erfasst wurde.

3.30 G99 – Position für Softwareendschalter-Rechts setzen

Wie Kapitel 3.29, hier jedoch für Software-Endschalter rechts.

3.31 G150 – Spline aus

Abwahl Spline-Interpolation

3.32 G151 – Spline ein

Anwahl Spline-Interpolation

3.33 G153 – Nullpunktverschiebung lesen

Mit diesem Kommando kann der aktuell gesetzte absolute Positionswert der Nullpunktverschiebung aller Achsen im System gelesen werden. Die zurückgelesenen Werte, werden in den Systemvariablen CD[50 + Achsindex] abgelegt in der benutzerspezifischen Interpolationseinheit.

Beispiel:

N100 G153 ' Nullpunktverschiebung auslesen in CD50 ff.

3.34 G154 – gelesene Nullpunktverschiebung reprogrammieren

Mit diesem Kommando können die per G153 gelesenen Positionswerte der Nullpunktverschiebung beliebiger Achsen reprogrammiert werden. Die Werte, werden den Systemvariablen CD[50 + Achsindex] entnommen.

Beispiel:

N100 G154 ' Nullpunktverschiebung setzen (absolut)

3.35 G161 – Kreismittelpunktskoordinaten relativ oder absolut

Bei Ausführung von Kreisen im Absolutmodus (G90) werden die Kreismittelpunktskoordinaten ebenfalls in Absolutkoordinaten angegeben. Diese Option ist im Konfigurationsprogramm IniCFG einstellbar.

3.36 G162 – Kreismittelpunktskoordinaten immer relativ

Kreismittelpunktskoordinaten werden immer in Relativkoordinaten angegeben.

4 M-Codes

Die nachfolgend aufgeführten M-Codes sind in McuWIN bekannt. Weitere M-Codes können vom Anwender definiert werden. Informationen hierzu ist im Handbuch „McuWIN-Bedienoberfläche“ zu finden. Kommentare in runden Klammern, sind nach M-Codes nicht erlaubt, da die Möglichkeit besteht M-Codes mit Parametern aufzurufen. Diese Parameter können auch Ausdrücke in runden Klammern sein.

4.1 M00 – Programm Halt

Programm Halt, Spindel, Kühlmittel und Vorschub aus: das Rücksetzen/Setzen der entsprechenden Digitalausgänge muss in Task 0 realisiert / angepasst werden. Die Anweisung schaltet in den Step-Mode um. Das Programm kann an der Stelle, an der angehalten wurde, fortgesetzt werden.

4.2 M01 – Wahlweiser Halt

Wahlweiser Halt: Wenn der Button „Wahlweiser Halt“ aktiviert wurde, wird mit dieser Anweisung in den Step-Mode umgeschaltet.

4.3 M02 – Programm Ende

Programm Ende. Das Programm wird angehalten. Es werden keine Ausgänge mehr verändert! Dieser Befehl bewirkt einen Aufruf in Task 0. Beachten Sie hierzu auch [Abschnitt 4.13]. Der Programmzeiger wird jedoch nicht auf den Programmanfang zurückgesetzt.

4.4 M03 – Spindel Rechtslauf

Dieses Kommando ist abhängig von der Art der Maschine und hat z.B. die Bedeutung <<Spindel Ein, Rechtslauf>>. Die entsprechende Funktionalität muss in Task 0 programmiert werden.

4.5 M04 – Spindel Linkslauf

Dieses Kommando ist abhängig von der Art der Maschine und hat z.B. die Bedeutung <<Spindel Ein, Linkslauf>>. Die entsprechende Funktionalität muss in Task 0 programmiert werden.

4.6 M05 – Spindel Stop

Dieses Kommando ist abhängig von der Art der Maschine und hat z.B. die Bedeutung <<Spindel Stop>>. Die entsprechende Funktionalität muss in Task 0 programmiert werden.

4.7 M06 - Werkzeugwechsel

Dieses Kommando hat die Bedeutung <<Auslösen des Werkzeugwechsels>>. Die entsprechende Funktionalität muss in Task 0 programmiert werden.

4.8 M08 – Kühlmittel Ein

Dieses Kommando hat die Bedeutung <<Kühlmittel Ein>>. Die entsprechende Funktionalität muss in Task 0 programmiert werden.

4.9 M09 – Kühlmittel Aus

Dieses Kommando hat die Bedeutung <<Kühlmittel Aus>>. Die entsprechende Funktionalität muss in Task 0 programmiert werden.

4.10 M17 – Unterprogramm Ende

Unterprogramm Ende und Rücksprung ins aufrufende Programm. Ab Version 2.5.3.41 bewirkt dieser Befehl keinen Zwischenstopp bei einer Interpolationsfahrt mehr.

4.11 M26 – Ausgang setzen

Digitalausgang setzen.

Syntax: M26 Parameter

Parameter ist eine Ganzzahl, bei welcher der Wert der Hunderterstelle die Achs-Nummer und der Wert der Einer- und Zehnerstelle den zu setzenden Ausgang angibt. Mit dem Wert 00 in der Einer-/Zehnerstelle werden alle Ausgänge dieses Achskanals gesetzt.

Beispiel: N0100 M26 102 Ausgang 2 von Achse 1 setzen

Dieser Befehl ist ein Spooler-Befehl und bewirkt beim Aufruf während einer Interpolationsfahrt keinen Zwischenstopp.

4.12 M27 – Ausgang rücksetzen

Digitalausgang rücksetzen.

Syntax: M27 Parameter

Parameter ist eine Ganzzahl, bei welcher der Wert der Hunderterstelle die Achs-Nummer und der Wert der Einer- und Zehnerstelle den rückzusetzenden Ausgang angibt. Mit dem Wert 00 in der Einer-/Zehnerstelle werden alle Ausgänge dieses Achskanals rückgesetzt.

Beispiel: N0100 M27 102 ' Ausgang 2 von Achse 1 rücksetzen

Dieser Befehl ist ein Spooler-Befehl und bewirkt beim Aufruf während einer Interpolationsfahrt keinen Zwischenstopp.

4.13 M30 – Programm Halt

Programm Halt mit Rücksetzen, Spindel, Kühlmittel und Vorschub aus (ähnlich M00): Das Rücksetzen/Setzen der entsprechenden Digitalausgänge muss in Task 0 realisiert werden.

Die Programmausführung wird angehalten und der Programmzeiger wird auf den Programmbeginn zurückgesetzt. Dann wird Task 0 aufgerufen. Hier müssen zusätzliche Funktionalitäten abgehandelt werden. Das Programm kann mit contcnct (3) dann wieder neu gestartet werden!

4.14 M80 – Ausgänge beschreiben (wortweise)

Mit diesem Kommando kann ein Ausgangsregister wortweise beschrieben werden. Hierbei ist zu beachten, dass unter Umständen vom System aktualisierte Ausgangsinformationen (z.B. PAE Ausgang) überschrieben werden. Des Weiteren ist zu beachten, dass die Ausgänge bei jeweils einer Achsgruppe nur gemeinsam gesetzt werden können.

Beispiel:

N220 M80 X 4 ' Bei der X-Achse wird der 3. Ausgang gesetzt, alle anderen Ausgänge werden rückgesetzt.

4.15 M96 – Unbedingter Sprung

Unbedingter Sprung zum angegebenen Label! Labels werden mit dem Zeichen L angegeben.
Ab Version 2.5.3.41 bewirkt dieser Befehl keinen Zwischenstopp bei einer Interpolationsfahrt mehr.

Beispiel:

```
N220 M96 L114          ' Unbedingter Sprung zum Label 114
N221 ...
...
N348 L114              ' Label 114
N349 ....
```

4.16 M98 – Unterprogramm aufrufen

Unterprogrammaufruf beim angegebenen Label L. Optional kann Zähler O für die Anzahl der Unterprogrammdurchläufe angegeben werden.

Beispiel:

```
N220 M98 L114 O5        ' Unterprogramm 114 aufrufen (5-mal)
N221 ...                ' weiteres Programm
...
%114                   ' Unterprogramm 114
N100 ...                ' Unterprogramm-Rumpf
N900 M17                ' Unterprogramm Ende
%
```

Unterprogramme können lokal im jeweiligen Quelltext nach dem Ende des Programmmoduls angefügt werden. Weiterhin können globale Unterprogramme in Include-Files dem Quelltext hinzugefügt werden. Weitere Anmerkungen hierzu in Kapitel 9.

Ab Version 2.5.3.41 bewirkt dieser Befehl keinen Zwischenstopp bei einer Interpolationsfahrt mehr.

4.17 M100 – Ausgang „Programm-Ende“ zurücksetzen

Applikationsspezifischer Befehl: Ausgang Programm-Ende zurücknehmen! Diese Funktionalität ist in Task 0 zu programmieren!

4.18 M150 – Aufzeichnung für grafische Systemanalyse starten

Durch Aufruf dieses Kommandos, wird der Positionsverlauf der ersten drei Achsen über 5 Sekunden durchgeführt. Der Positionsverlauf kann in mcfg mit Hilfe des Grafikbildschirms dargestellt werden.

4.19 M901 – Common-Integer-Variable resident speichern

Mit diesem Kommando kann eine Ganzzahlvariable des Common-Integer-Bereichs resident gespeichert werden. Diese Information kann dann per M902 zu einem beliebigen Zeitpunkt wiederhergestellt werden, auch nach einem Programm-Neustart. Die Ablage der Information erfolgt im INI-File von McuWIN. Als Parameter muss der Index der zu speichernden Variablen angegeben werden (0..999).

Beispiel:

```
N020 M901 601
```

4.20 M902 – Resident gespeicherte Common-Integer-Variable rücklesen

Mit diesem Kommando kann eine Ganzzahlvariable des Common-Integer-Bereichs wiederhergestellt werden, die zuvor mit dem Kommando M901 gesichert wurde. Wenn die Variable zuvor noch nicht gesichert wurde, dann wird sie durch das Kommando auf 0 gesetzt. Wenn Common-Variable mit Indices < 600 geschützt sind, können diese auch nicht wiederhergestellt werden. Als Parameter muss der Index der zu speichernden Variablen angegeben werden (0..999).

Beispiel:

```
N020 M902 601
```

4.21 M903 – Common-Double-Variable resident speichern

Mit diesem Kommando kann eine Gleitpunktvariable des Common-Double-Bereichs resident gespeichert werden. Diese Information kann dann per M904 zu einem beliebigen Zeitpunkt wiederhergestellt werden, auch nach einem Programm-Neustart. Die Ablage der Information erfolgt im INI-File von McuWIN. Als Parameter muss der Index der zu speichernden Variablen angegeben werden (0..999).

4.22 M904 – Resident gespeicherte Common-Double-Variable rücklesen

Mit diesem Kommando kann eine Gleitpunktvariable des Common-Double-Bereichs wiederhergestellt werden, die zuvor mit dem Kommando M903 gesichert wurde. Wenn die Variable zuvor noch nicht gesichert wurde, dann wird sie durch das Kommando auf 0,0 gesetzt. Wenn Common-Variable mit Indices < 600 geschützt sind, können diese auch nicht wiederhergestellt werden. Als Parameter muss der Index der zu speichernden Variablen angegeben werden (0..999).

4.23 M910 – PC nach Programm-Ende ausschalten

Ab Windows XP kann mit einem M910 vor dem Programm-Ende der PC nach Beendigung des G-Code-Programms ausgeschaltet werden.

Beispiel:

```
....  
M901  
M30  
%
```

4.24 M911 – McuWIN.exe nach Programm-Ende beenden

Ab McuWIN V2.5.3.117 kann mit einem M911 vor dem Programm-Ende das Programm nach Beendigung des G-Code-Programms beendet werden.

Beispiel:

```
....  
M911  
M30  
%
```


5 Sonstige Codes

5.1 Labels

Labels dienen zur Kennzeichnung von Unterprogrammen und Sprungzielen und werden mit dem Buchstaben L mit einer kennzeichnenden Nummer angegeben. Die Definition eines Labels bewirkt keinen Zwischenstopp bei einer Interpolationsfahrt. Der Modulname eines Unterprogramms wird gleichzeitig als Label angelegt und kann für Unterprogrammaufrufe verwendet werden.

Beispiel:

```
N220 L114          ' Label 114

%UP114             (Unterprogramm UP114)
...
M17
%
```

5.2 Realisierte Sonderfunktionen

Befehlscode	Befehlsnummer	Parameter	Bemerkung
S	1	Parameter in CD0	Hauptspindeldrehzahl
T	2	Parameter in CD0	Werkzeugwechsel

Diese Codes (Befehlsnummern) werden in der Variable CI2 an die Task 0 übergeben, und dort als Unterprogramme ausgeführt.

5.3 F-Kommando

Vorschubgeschwindigkeit setzen für G01, G02, G03 Befehle in der Einheit, die in der Initialisierungs-Task angewählt wurde. Wenn in einer Zeile mit einem G-Kommando auch ein F-Kommando steht, gilt die neu programmierte Geschwindigkeit bereits für dieses Profil.

Eine evtl. notwendige Achsverzögerung wird im vorherigen Profilabschnitt durchgeführt. Eine evtl. notwendige Achsbeschleunigung wird im aktuellen Profil durchgeführt. Diese Anweisung bewirkt keinen Zwischenstopp bei einer Interpolationsfahrt.

5.4 S-Kommando

Einstellung der Hauptspindeldrehzahl: Die Funktionalität dieser Funktion muss in Task 0 programmiert werden. Der Funktionscode dieses Kommandos ist 1. Der Parameter wird in CD0 übergeben. Das S-Kommando, welches in Task 0 ausprogrammiert ist, unterbricht die Interpolationskontur und führt die vorherigen Zeilen aus durch den Befehl SSMSIW. Dies ist ein Befehl speziell für diesen Anwendungszweck, welcher in der rwSymPas-Standardprogrammierung nicht verwendet wird.

Falls S-Kommandos ohne Unterbrechung der Interpolationskontur realisiert werden sollen, muss dieser Befehl weggelassen werden. In diesem Fall müssen dann entsprechende Spooler-Kommandos ausgeführt werden, um die Hauptspindel / das Bearbeitungswerkzeug zu steuern (SSF).

5.5 T-Kommando

Werkzeugwechsel: Die Funktionalität dieser Funktion ist in Task 0 programmiert. Der Funktionscode dieses Kommandos ist 2. Der Parameter wird in CD0 übergeben.

Für die Werkzeugradiuskorrektur wird mit diesem Kommando ein Korrektur-Datensatz ausgewählt, der in der Werkzeugtabelle definiert ist. Standardmäßig ist nach dem Systemstart Werkzeug Nr. 0 ausgewählt. Eine Werkzeuganwahl ist nur möglich, wenn beim Start von McuWIN eine Werkzeugtabelle mit Inhalten auf die Steuerung geladen wurde, d.h. wenn die Werkzeugdatei (Standardname: ToolComp.ini) auch tatsächlich Werkzeuge enthält.

5.6 D-Kommando

Werkzeugkorrekturspeicher anwählen: Die Funktionalität dieser Funktion muss in Task 0 programmiert werden. Der Funktionscode dieses Kommandos ist 3 (in C12). Der Parameter wird in CD0 übergeben.

5.7 String Ausgabe

Mit Hilfe der Kommandos WRITE bzw. WRITELN ist es möglich, Anzeige-Informationen zur Laufzeit eines Anwenderprogramms zu generieren. Hierbei wird der Anzeigestring in einem quittierbaren Fenster über McuWIN gelegt. Durch Ausgabe eines Leerzeichens (Blank bzw. ' ') kann dieses Fenster auch wieder automatisch geschlossen werden.

Ab McuWIN V2.5.3.101 kann aus jeder Task das Fenster derselben oder einer anderen Task durch Ausgabe des Strings '!0' bzw. '!3' geschlossen werden, wobei die Zahl den Index der Tasknummer darstellt, deren Fenster zu schließen ist.

Ab McuWIN V2.5.3.131 kann mit dem Merker MessageTask?Continue = 1 in der Section [MCU] der Datei McuWIN.ini veranlasst werden, dass durch Quittieren des Fensters über die Schaltfläche „OK“ die jeweilige Task aus dem SingleStep-Modus fortgesetzt wird.

5.7.1 WRITE Kommando

Mit diesem Kommando wird eine Zeichenkette zur Anzeige generiert und gegebenenfalls an eine noch nicht abgeschlossene Zeichenkette angehängt, aber nicht abgeschlossen, d.h., weitere Ausgaben können noch angehängt werden.

Parameter können Stringkonstanten, Integer, Double und Boolesche Variable sowie Berechnungsausdrücke sein.

Beispiel:

```
WRITE "Istposition: " X.rp
```

5.7.2 WRITELN Kommando

Mit diesem Kommando wird eine Zeichenkette zur Anzeige generiert und gegebenenfalls an eine noch nicht abgeschlossene Zeichenkette angehängt und abgeschlossen, d.h., weitere Ausgaben können nicht mehr angehängt werden.

Parameter können Stringkonstanten, Integer, Double und Boolesche Variable sowie Berechnungsausdrücke sein. Die Ausgabe eines Leerzeichens schließt das Ausgabefenster.

Beispiel:

```
WRITELN "Istposition: " X.rp
```

6 Kommentare

Kommentare können mit runden Klammern umschlossen werden. Weiterhin besteht die Möglichkeit, den Rest einer Zeile durch ein Hochkomma als Kommentar zu kennzeichnen. Die Schachtelung von Kommentaren ist möglich.

Beispiele:

```
N210 G00 X10 Y-20   (Dies ist ein Kommentar)
N220 G90            ' Dies ist ein Kommentar
```

Vorsicht: In Ausdrücken ist die Verwendung von runden Klammern jedoch möglich, ohne dass der Ausdruck ein Kommentar darstellt. In nachfolgendem Beispiel ist der Klammerinhalt **kein** Kommentar:

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

Ein Berechnungsausdruck wird immer auch dort erwartet, wo ein Parameter möglich ist, z.B. nach M-Befehlen. In diesem Fall wird der Compiler-Fehler 2074 angezeigt.

Im Zweifelsfall ist die Kennzeichnung von Kommentaren per Hochkomma zu bevorzugen. Kommentare dürfen nicht geschachtelt werden.

7 Bedingte Programmausführung

Abhängig von Booleschen Werten können bedingte Programmteile ausgeführt werden. Die Schreibweise von Ausdrücken und Operatoren ist der SAP-Programmierung der APCI-8001 angelehnt. Bei dieser Schreibweise ist insbesondere zu beachten, dass Programmanweisungen mit Semikolon abgeschlossen werden müssen (nicht begin) und dass Kommentare mit // anstatt mit Hochkomma bzw. mit geschweiften Klammern anstatt mit runden Klammern gekennzeichnet werden.

Beispiel:

```
N350  $if (Ausdruck) then begin
N400  .....                (Anweisungen)
N450  $end
```

Ausdruck ist ein Boolescher Ausdruck. Einige Beispiele hierzu:

```
(CI600 < 20)           // Ergebnis eines Vergleichs
BOOLEAN (CI600)        // Konvertierung eines Zahlenwertes
(X.digib.5)            // Abfrage des Digital-Eingangs 5 der X-Achse
```

Der bedingt auszuführende Programmblock muss mit \$end abgeschlossen werden. Im Bedarfsfall kann ein \$else Zweig deklariert werden.

Beispiel:

```
N350  $if (Ausdruck) then begin
N400  .....                (Anweisungen)
N450  $end else begin
N500  ....                  (Anweisungen)
N550  $end
```

Weiterhin ist es möglich If-else-If Ketten aufzubauen.

Beispiel:

```
N350  $if (Ausdruck 1) then begin
N400  .....                (Anweisungen)
N450  $end else if (Ausdruck 2) begin
N500  ....                  (Anweisungen)
N550  $end else if (Ausdruck 3) begin
N600  ....                  (Anweisungen)
N650  $end else begin
N700  ....                  (Anweisungen)
N750  $end
```

8 Schleifen

8.1 while-do Schleife

Beispiel:

```
N350 $while (Ausdruck) do begin
N400 ..... (Anweisungen)
N450 $end;
```

Ausdruck muss einen booleschen Wert zurückliefern (siehe IF). Die Schleife wird so lange ausgeführt, bis Ausdruck false wird. Der Wert von Ausdruck wird jeweils am Schleifenanfang abgeprüft, d.h. wenn der Wert beim Erreichen der Schleife false ist, wird die Schleife übersprungen.

8.2 repeat-until Schleife

Beispiel:

```
N350 $repeat begin
N400 ..... (Anweisungen)
N450 $end until (Ausdruck);
```

Ausdruck muss einen booleschen Wert zurückliefern (siehe IF). Die Schleife wird so lange ausgeführt, bis Ausdruck true wird. Der Wert von Ausdruck wird jeweils am Schleifenende abgeprüft, d.h. die Schleife wird mindestens einmal ausgeführt.

8.3 for Schleife

Beispiele:

```
N350 $for CI600:=1 to 10 do begin
N400 ..... (Anweisungen)
N450 $end;

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

Die Laufvariable (hier CI600) muss eine Ganzzahlvariable sein. Die Definition der for-Schleife schließt einen Start und einen Endwert mit ein. Start und Endwert können auch Ausdrücke sein.

9 Einbinden von Include-Files

Mit der Anweisung \$I können Include-Files in einen Quelltext eingebunden werden. Include-Files können weitere Include-Files enthalten. Der Dateiname kann auch eine Laufwerks- und Pfadangabe enthalten. In Dateiname und Pfadangabe darf aber kein Leerzeichen enthalten sein.

Beispiel:

\$I UP2.SRC

Mit Hilfe dieser Funktion ist es möglich eine Unterprogrammsammlung bestehend aus eigenständigen Programmen zu einem Quelltext hinzuzufügen.

Beispiel:

(im Hauptprogramm)

```
N400  M98 L2000 O3      (Unterprogramm L2000 3-mal aufrufen)
N500  ...
N900  M02              ' Ende Hauptprogramm
%
```

(Definition eines lokalen Unterprogramms)

```
%107
N0100 L107              (Lokales Unterprogramm 107)
N0200 ...               (Programmrumpf)
N0300 M98  L1000 O2      (Unterprogramm L1000 2-mal aufrufen)
N1000 M17              ' Rücksprung
%
```

\$I Unterprogramme.inc (Library der globalen Unterprogramme einbinden)

Das File Unterprogramme.INC kann nun z.B. folgenden Text enthalten:

```
$I UP1.SRC  (Modul 1000: Schlüsselloch ausfräsen)
$I UP2.SRC  (Modul 2000: Sackbohrung für Pressluft anbringen)
$I UP3.SRC  (Modul 3000: Oberkante fäsen)
```

Weiterhin noch ein Beispiel für das globale Unterprogramm UP1.SRC:

```
%1000              (Modulname = Label)
N200  G00 X0 Y0 Z0
N300  G00 X10 Y10 Z-10
N400  M17          ' UP Rücksprung
N500  %
```

10 Einbindung von rw_SymPas Kommandos

In Programme nach DIN 66025 können rw_SymPas Quelltextabschnitte eingebunden werden. Dazu wird der DIN-Modus mit der Anweisung \$DINEND beendet.

Beispiel:

```
N200 $DINEND
```

Nach dieser Anweisung lassen sich rw_SymPas Anweisungen einfügen ohne Zeilennummerierung. Die Zurückschaltung in den DIN-Modus erfolgt mit der Compileranweisung {\$DINSTART}.

Beispiel:

```
{$DINSTART}  
N220 G00 X10
```

Weiterhin ist es möglich innerhalb eines GCode-Programms eine einzelne Programmzeile mit dem Zeichen \$ auf die rw_SymPas-Syntax umzuschalten.

Beispiel:

```
$wt(2000);
```

Bevor ein GCode-Programm beginnt (vor dem ersten %-Zeichen), kann bereits ein rw_SymPas Programmblock eingebunden werden, in dem Variable, Konstante, Labels und Prozeduren deklariert werden. Diese Objekte können dann im GCode-Programm mit den oben beschriebenen Mechanismen verwendet werden. Hierbei ist zu beachten, dass Konstanten- und Variablennamen in G-Code-Programmen keine numerischen Zeichen enthalten dürfen.

11 Rechenparameter

Für mathematische Operationen wie z.B. Berechnung von Zielpositionen können Rechenparameter verwendet werden. Diese Rechenparameter können auch zur Entscheidung bei bedingten Sprungfunktionen verwendet werden.

Die Rechenparameter werden mit CD<n> für Gleitpunktzahlen und mit CI<n> für Ganzzahlwerte referenziert. Für den Benutzer sind die Werte für n = 600..699 reserviert.

Vorsicht: Das System lässt auch andere Werte für n zu. Alle Werte außerhalb des angegebenen Bereiches können Systemvariable sein, und können beim Beschreiben zur Instabilität des Systems führen.

Beispiel:

```
N220  CD620 := 100
N230  G00 X2*CD620
```

12 Variable

Durch Umschaltung auf die `rw_SymPas` Syntax können auch Variable deklariert werden, die im G-Code-Programm verwendet werden können. Ein Beispielprogramm ist nachfolgend aufgeführt. Dieses Programm ist im Lieferumfang enthalten.

' G-Code Programmbeispiel
' - Deklaration von Variablen und Konstanten
' - Verwendung mathematischer Funktionen in G-Code-Programmen

\$DINEND

// rw_SymPas Programmblock zur Deklaration von Variablen}

var

sRadius, step : double; { Gleitpunktvariable }
cntr : integer; { Ganzzahlvariable }

const

Loopcntr = 500; { Konstante }

{ \$DINSTART }

%SAMPLE_PROGRAM

G90

G00 X100 Y200 Z-10

Z-40

step := 0.5 ' Zuweisung an eine Variable

G90 F20

\$for cntr := 0 to Loopcntr do begin // for-Schleife in rw_SymPas Syntax

 C1601 := cntr

 C1602 := Loopcntr

 G01 X(X.tp + step) Y (200.0 + 5.0 * sin (X.tp / 10.0)) ' Verwendung der Sinus-Funktion

\$end;

M30

%

Ab McuWIN V2.5.3.90 kann ein derartiger Initialisierungsbereich (fettgedruckt) per PreInclude-File in einen für den Anwender unsichtbaren Bereich ausgelagert werden. Ebenso kann ein am Ende angehängter Bereich per PostInclude-File ausgelagert werden.

13 Handhabung von McuWIN

13.1 Problembehandlung

13.1.1 Fehleranzeigen von McuWIN

Im Fehlerfenster von McuWIN werden eventuelle Status- oder Fehlerinformationen für den Anwender sichtbar gemacht. Hierbei muss unterschieden werden zwischen Übersetzungsfehlern und Laufzeitfehlern. Übersetzungsfehler werden angezeigt, während das Programm übersetzt bzw. gestartet wird. Laufzeitfehler treten während der Programmabarbeitung auf. Teilweise lassen sich Fehler durch einen Klick auf das Fehlerfenster quittieren. In den meisten Fällen ist ein Reset des Programms notwendig. Dadurch ist auch die Referenzierung des Systems aufgehoben. Im Allgemeinen muss die Fehlerursache, z.B. im Programmquelltext, beseitigt werden.

13.1.1.1 Fehler # 1: Unbekannter Funktionscode!

Im Programm wird ein undefinierter Befehl (z.B. G oder M) verwendet. Unter Umständen muss der Befehlsinterpreter applikationsspezifisch angepasst werden.

13.1.1.2 Warnung # 1: Enkoder-Error!

Das Enkoder-Error-Flag von mindestens einer Achse wurde erkannt. In diesem Fall kann die exakte Position der betreffenden Achse nicht mehr garantiert werden. Dieses Flag deutet auf ein Hardware- oder Verkabelungsproblem hin.

13.1.1.3 Fehler # 2: Schleppfehler (Achse ?)!

Eine Überschreitung des vordefinierten maximalen Schleppfehlers bei einer Achse wurde erkannt. Die verursachende Achse wird angezeigt.

13.1.1.4 Status # 2: Warte bis Tür geschlossen!

Warten, bis die Schutztür geschlossen wurde.

13.1.1.5 Fehler # 4: Hardware-Endschalter (links Achse ? / rechts Achse ?)!

Mindestens ein Hardware-Endschalter wurde erkannt. Achskanal und Rechts/Links wird in der Meldung spezifiziert. Diese Überwachung ist nur aktiv, wenn das System vollständig referenziert ist, da ein Anfahren von Endschaltern während der Referenzierung durchaus beabsichtigt sein kann.

13.1.1.6 Status # 4: Programm wurde gestartet!

Ein Applikationsprogramm wurde gestartet! – normaler Betriebszustand

13.1.1.7 Fehler # 8: Software-Endschalter (links Achse ? / rechts Achse ?)!

Mindestens ein Software-Endschalter wurde erkannt. Achskanal und Rechts/Links wird in der Meldung spezifiziert. Diese Überwachung ist nur aktiv, wenn das System vollständig referenziert ist, da eine Positionsüberwachung nur bei einem referenzierten System sinnvoll ist.

13.1.1.8 Status # 8: Programm wurde beendet!

Ein Applikationsprogramm wurde beendet! – normaler Betriebszustand

13.1.1.9 Status # 10: Not-Aus

Mit Hilfe eines als EO deklarierten Digitaleingangs wird ein laufendes Programm abgebrochen und ein Neustart verhindert. Dieser Fehler wird automatisch quittiert, wenn der entsprechende Eingang inaktiv wird. Auf diese Weise reagiert die Steuerung auf ein Not-Aus-Signal. Hierbei ist zu beachten, dass der Not-Aus-Zustand immer direkt auf die Leistungselektronik einwirken muss, um den einschlägigen Sicherheitsvorschriften zu genügen.

13.1.1.10 Warnung # 10: Positionsfehler erkannt!

Die Positionsüberwachung per Nullspursignal hat einen Positionsfehler diagnostiziert und deutet auf ein Problem mit einem Positionsmesssystem hin. Voraussetzung für eine zuverlässige Anzeige ist jedoch die richtige Konfiguration der Positionsüberwachung.

13.1.1.11 Fehler # 20: Konflikt in den Konfigurationsdaten!

Die Einrichtung von McuWIN ist fehlerhaft. Dieser Fehler kann unterschiedliche Ursachen haben und z.B. auf die Deklaration unerlaubter Achsen oder die Angabe von 0 in Standard-Geschwindigkeits- oder Beschleunigungswerten hindeuten. Eine weitere Möglichkeit sind falsch definierte Software-Endschalterbereiche.

13.1.1.12 Status # 20: Referenzierung wurde gestartet!

Die Referenzfahrt wurde gestartet! – normaler Betriebszustand.

13.1.1.13 Fehler 21 beim Laden von <<??>>>

Dieser Übersetzungsfehler kann auftreten, wenn ein übersetztes Programm (.cnc Datei) auf die Steuerung geladen werden soll, z.B. nach Betätigung der Start- oder Schritt-Schaltfläche. In diesem Fall kann die gewünschte Datei nicht auf die Steuerung geladen werden.

Dieser Fehler tritt manchmal bei sehr großen Dateien auf, die z.B. durch automatische Generierung aus CAD-Daten erzeugt werden. Standardmäßig können CNC-Files bis zu einer Größe von 100.000 Bytes verarbeitet werden (Größe des übersetzten Files mit der Endung .CNC). Falls größere Dateien erzeugt werden, kann die steuerungsspezifische Umgebungsvariable „SZTSK3“ auf einen höheren Wert gesetzt werden (max. bis ca. 12 MByte). Falls diese Größe ebenfalls nicht ausreicht, sollten die Dateien mit verringerter Auflösung generiert werden.

13.1.1.14 Fehler # 40: Fehler bei Referenzfahrt (Achse ?)!

Laufzeitfehler während des Referenzierens. Kann z.B. auftreten, wenn die Bedingungen für die Referenzfahrt falsch gesetzt sind.

13.1.1.15 Status # 40: Warten bis Spindeldrehzahl erreicht...

Warten, bis die Spindel die erforderliche Drehzahl erreicht hat. – Normaler Betriebszustand, kann aber auch auf ein Problem mit der Spindel bzw. Spindelrückmeldung hindeuten.

13.1.1.16 Fehler # 80: Versionskonflikt McuWIN - SAP!

Die Versionen von McuWIN.EXE, RWMOS.ELF oder den Taskprogrammen sind nicht zusammenpassend. Ein Update wurde nicht ordnungsgemäß durchgeführt. Unter Umständen müssen die Taskprogramme neu übersetzt werden. Vielleicht muss der PC vor Wiederinbetriebnahme neu gebootet werden, damit alle Programme reinitialisiert werden.

13.1.1.17 Status # 80: Fehlerbit in ErrorReg erkannt! ? hex

Laufzeitfehler, der ein Fehlerbit im steuerungsspezifischen Register ErrorReg der APCI-8001 anzeigt. 20 hex deutet auf ein Verfahrprofil ohne Verfahrweg hin, 200 hex zeigt an, dass eine Fehlermeldung im fwsetup-Monitor-Screen angezeigt wird. Weitere Hexwerte sind im Programmierhandbuch PHB der Steuerung beschrieben.

13.1.1.18 Fehler # 100: Konfigurationsfehler!

Die Konfigurationsdaten der Steuerung sind nicht gespeichert. Dies kann z.B. nach Änderungen an der Konfiguration oder nach dem Ersatz der Steuerungsbaugruppe auftreten. In diesem Fall sind die Systemdaten in mcfg bei gebootetem System zu speichern.

13.1.1.19 Status # 100: Fehlerbit in IFS-Register erkannt! ?? hex

Im IFS-Register der Steuerungsbaugruppe wurde ein Fehlerbit erkannt. Eine genauere Diagnose ist mit Hilfe von mcfg möglich. Eine fehlerfreie Ausführung von Programmen ist nicht mehr gewährleistet.

13.1.1.20 Fehler 200: Fehler bei Werkzeug-Radius-Korrektur!

Durch ein Konfigurationsproblem wurde die Werkzeugradiuskorrektur in einen ungültigen Zustand gebracht.

13.1.1.21 Status # 400: Vorschub Sperre Aktiv (UI)!

Mit Hilfe eines als UI deklarierten Digitaleingangs kann der Achsvorschub gesperrt werden. Dieser Fehler wird automatisch quittiert, wenn der entsprechende Eingang inaktiv wird.

13.1.1.22 Fehler # 800: Verstärker nicht bereit!

Das Signal DNR (Drive Not Ready) von mindestens einer Achse zeigt an, dass ein Leistungsverstärker nicht betriebsbereit ist.

13.1.1.23 Status # 1000: Laufzeitfehler in SAP-Task!

In einer Stand-Alone-Task ist ein Laufzeitfehler aufgetreten; das System ist nicht mehr betriebsbereit. Mit Hilfe von mcfg kann die Ursache ermittelt werden. Hierbei kann ein Systemprogrammfehler die Ursache sein. Diese Fehler können im Allgemeinen nicht vom Anwender selbst behoben werden.

13.1.1.24 Status # 2000: Parameterfehler zur Laufzeit!

Einem Befehl im Anwenderprogramm wurden falsche Parameter übergeben.

13.1.1.25 Fehler # 4000: Ungültiger S-Wert!

Das S-Kommando wurde mit einem unerlaubten Drehzahlwert verwendet.

13.1.1.26 Fehler # 80000000: System wurde angehalten! Reset erforderlich ...

Das System wurde durch einen undefinierten Vorgang angehalten. Dies kann z.B. passieren, wenn mit einer anderen Applikation als mcfg auf die Steuerung zugegriffen wird, während McuWIN in Betrieb ist.

14 Versionsinformation

Die nachfolgende Beschreibung ist gültig für

McuWIN.EXE	ab Version 2.5.3.111
IniCfg.EXE	ab Version 2.5.3.87
MCFG.EXE	ab Version 2.5.3.91
MCUG3.DLL	ab Version 2.5.3.90
NCC.EXE	ab Version 2.5.3.63
RWMOS.ELF	ab Version 2.5.3.116
ToolEdit.EXE	ab Version 2.5.3.8

14.1 Änderungen in früheren Versionen

Version 2.5.2.122

- Im Single-Step-Modus ist nun auch Teach In möglich (Zeilen einfügen oder Zeilen modifizieren).
- Ergänzungen für Standardbenutzer ohne Schreibrechte für McuWIN.ini
- Ergänzung / Vervollständigung von Sprach-Tags
- Warnausgaben und Ausgaben in Protokolldatei überarbeitet

Version 2.5.2.117

- TeachIn-Button-Aktivierung flackerte
- M911 neu: McuWIN bei Programm-Ende beenden
- Optionaler automatischer Start von Anwenderprogrammen erst nach Beendigung der Initialisierung in Task 1
- M00 bewirkt Aktivierung des „Continue“-Button
- Alive-Counter in CI59: Hängen von McuWIN.EXE wird nun in Task1 erkannt und bearbeitet.

Version 2.5.3.112

- Zustandsüberwachung im Single-Step-Modus verbessert

Version 2.5.3.111

- TC-Werkzeugverwaltung überarbeitet:
ToolEdit V2.5.3.8
Schreib-/Lesezugriff auf ToolCompensation-Group 0: greift auf die Gruppe der aktiv angewählten Ebene zu

Version 2.5.3.110

- Task-Programmierung von McuWIN: Konstante MyTaskNr neu für Debug-Anzeigen, z.B. in Schleifen
- Task1.src: Einbindung von appeo_off.inc verlegt, da hierzu die geschlossenen Regelkreise notwendig sein können

Version 2.5.3.108

- Lesen von IniFile unabhängig von Ländereinstellungen (DecimalSeparator)
IniCFG – V2.5.3.89
ToolEdit - V2.5.3.7
- RegDisp V2.5.3.6: mit Read-only INI File ist nun möglich
- T-Parameter kann nun auch Variable sein

Version 2.5.3.107

- Neue Message-Fenster auch für Task 1 und Task 2 hinzugefügt
- Beim Starten von Task1 eine Meldung "Initialisierung ..." aufgelegt
- Schreiben aller Fehlermeldungen und Warnungen in eine Protokolldatei, damit gelöschte Meldungen im Nachhinein verifiziert werden können --> IniFile Variable [MCU] ProtocolFileName: Hier muss ein Dateiname mit Laufwerk und Pfad eingetragen werden. Die Einstellung kann nur manuell im Ini-File angepasst werden, nicht per IniCfg.
- Zeichengröße im Editor einstellbar per IniFile
- [EDITOR] EditorFontSize (Standard 10)
- Die Einstellung kann nur manuell im Ini-File angepasst werden, nicht per IniCfg.
- Aufbereitung der Fehlertexte, wenn Achsen spezifiziert werden, nun aus Inifile
- [FEHLERTEXTE_EXT] Zusatzinfo????
- Die Einstellung kann nur manuell im Ini-File angepasst werden, nicht per IniCfg.

Version 2.5.3.106

- Selbsthaltung von Kommandos wird nun von DINEND und DINSTART aufgehoben
- Abschließendes % wurde bei automatischer Satznummernvergabe nicht richtig erkannt
- BoardType in CI67 ablegen, wenn eine Karte erfolgreich initialisiert wurde
- Checks in Task1.src abschaltbar per Register SwitchOffChecks; Bit-Deklarationen SwitchOffCheck_xxx hierzu in GCode.inc
- Bei APCI-8008-Leitungsbruch Flags überwachen

Version 2.5.3.105

- McuWIN konnte mit RWMOS.ELF zwischen 1.2.2013 und 27.2.2013 einen Deadlock beim Programmstart bewirken
- Referenzschalter freifahren, gegebenenfalls beliebig oft wiederholen
- Fehler # hex 8000 0000 erzeugte einen negativen Eintrag im Ini File --> Anzeigeproblem

Version 2.5.3.103

- Fehleranzeigen ergänzt von CI10 auf CI10 und CI48: In CI48 sind nun benutzerspezifische Fehlerbits möglich.
Hierzu neue Sektion [FEHLERTEXTE48] in mcuwin.ini

Version 2.5.3.101

- Meldungsfenster in McuWIN (Task writeln) per "!" schließen, wobei ? = Task Nr.

Version 2.5.3.100

- Werkzeugradiuskorrektur und Werkzeuglängenkorrektur überarbeitet:
Vorher Fehlverhalten bei Werkzeuglänge und bei Anwahl anderer Ebenen als G17
Programm ToolEdit V2.5.3.6 überarbeitet: Nun sind auch Werkzeuge für alle Ebenen möglich
Bei Initialisierung: Änderungen Nullpunktverschiebung und TC
- Verwendung von G04 ohne Parameter erzeugte eine Exception in RWMOS –
nun Compiler-Fehler 2075

Version 2.5.3.99

- In McuWIN.ini Sektion [Project data]: Mit SelectedCardNr kann eine andere als die 1. Karte im System aktiviert werden.
- Kompatibilitätshinweis: McuWIN mit WebServices muss mit mcug3.dll zusammenpassen
- Verbesserungen bei Web-Services Verbindungsaufbau: Timeout / zusätzliche Fehlerangaben in Log-Messages

Version 2.5.3.96

- Meldung "Ein deaktiviertes oder unsichtbares Fenster kann nicht den Focus erhalten!" verhindert
- Undo Button für TeachIn
- Manuelles Verfahren beim Programmstart sicher rücksetzen
- TASK1: Hardware-Endschalter-Handling geändert
- Message von Task 0 bei "Forts." schließen
- ToolFile kann nun in IniCFG auch abgeschaltet werden.
- G-Code-Parameterauswertung war teilweise fehlerhaft.

Version 2.5.3.95

- InhibitProfileRefuse verwenden für Warn-Anzeige; Beschleunigung oder Geschwindigkeit = 0 stoppt nun den Programmablauf; bisher wurde ein derartiges Verfahrprofil verworfen.

Vorsicht: unter Umständen Inkompatibilität

Version 2.5.3.94

- appeo_off.inc wurde zuvor an falscher Stelle eingebunden:
Vorsicht: Unter Umständen Kompatibilitätsproblem nach Update, wenn diese Datei verwendet wurde!
- M-Codes heben Selbsthaltung von G-Befehlen nicht mehr auf

Version 2.5.3.93

- Möglichkeit zum Schließen von Textausgabefenstern
- M910 – PC nach Programmende herunterfahren

Version 2.5.3.90

- Möglichkeit der Angabe von Pre- und Post-Include-Files im G-Code Programm
- Fehlendes M02 / M30 / M17 in G-Code Programmen bewirkt Fehler 190 beim Übersetzen
- Parameterinfo AXSEL und NUMPARAMS überarbeitet

Version 2.5.3.89

- Selektion von Bearbeitungsebenen (G17 / 18 / 19) nur, wenn auch reale Werkzeugtabellen aktiv sind

Version 2.5.3.88

- Kompensation von Spindelsteigungsfehler und Winkelfehler implementiert
- In Sollposition kann alternativ die Zielposition des aktiven Verfahrprofils angezeigt werden
- Beim manuellen Verfahren von Achsen kann nun eine Schrittweite angegeben werden.
- In der Maske für manuelles Verfahren kann per Schaltfläche die Ruheposition angefahren werden.
- Mit den Kommandos M901 – M904 können Common-Variable resident auf die Festplatte geschrieben und wiederhergestellt werden.
- Wenn rw_SymPas in G-Code-Programmen verwendet wurde, war die automatische Initialisierung der Interpolationsachsen fehlerhaft.
- Nullpunktverschiebung wird nach Programmstopp und Neustart gelöscht.
- Beenden von McuWIN ist nur möglich, wenn nicht im Automatikmodus.
- Interne Handhabung von Fehler- und Warnmeldungen geändert.
- Vorzeichen S-Befehl bei Änderung der Spindeldrehzahl berücksichtigen.
- Stopp beim manuellen Verfahren nicht mehr mit sdec, sondern mit jac, dadurch sanfteres Verhalten beim manuellen Verfahren
- Spindeldrehzahl-Sollwert beim Programmstart abnullen.
- Übersetzen von Dateien und Syntaxcheck ist nun auch im Offline-Modus möglich.
- Override-Korrektur beim Abbremsen verbessert.

Version 2.5.3.82

- Ergänzungen bei Nutzung von Web-Services, Verwendung unter Linux/WINE möglich
- Neue applikationsspezifische Datei AppEO_Off.inc für zusätzliche Gestaltungsmöglichkeiten bei der Einrichtung von McuWIN.

Version 2.5.3.78

- G60, G90/91 werden nun zur Laufzeit von Programmen, nicht mehr beim Übersetzen berücksichtigt. Hierzu ist mindestens RWMOS V2.5.3.89 notwendig.
- Möglichkeit der Schnittgeschwindigkeits-Interpolation neu
- Im Single-Step wurden Verfahrbewegungen nicht immer sicher gestartet.

Version 2.5.3.76

- In IniCfg.EXE sind zusätzliche Einstellungen möglich:
Registerkarte „Achs Def.“ – hier kann für einzelne Achsen ein Open-Loop-Betrieb angewählt werden.
Registerkarte „Hardware Ausgänge“ – hier kann ein Hilfsspannungsausgang definiert werden.
- Nach dem Programmstart oder nach einem Fehler muss ein evtl. definierter Hilfsspannungsausgang in einer Maske manuell gesetzt werden.
- In Task eins wird die neue Datei AppStartChecks.inc eingebunden, welche beim Start eine einmalige applikationsspezifische Systemüberprüfung ermöglicht.

Version 2.5.3.75

- Für rotatorische Achsen kann in IniCfg.EXE das Anfahren der Zielposition auf dem kürzesten Verfahrweg angewählt werden (Registerkarte „Desktop / System“).

Version 2.5.3.61

- Das Programm ToolEdit.exe zum Editieren der Werkzeugtabelle schreibt den aktuellen Stand der Werkzeugtabelle sofort beim Speichern auf die Steuerung. Änderungen werden somit online wirksam.
- McuWIN.exe und IniCfg.exe: Behandlung der Hauptspindel überarbeitet. Unter anderem können nun auch Soll- und Istwert der Spindeldrehzahl in McuWIN angezeigt werden.

Version 2.5.3.52:

- D-Kommando wird nun auch unterstützt. Diese Funktion kann in Appcommands.inc unter Funktion = 3 ausprogrammiert werden.
- Zeilentrace verbessert, zeigt nun bei G01/02/03 immer die Zeile an, die auch tatsächlich abgefahren wird.
- Compiler-Fehleranzeige verbessert
- Standardpfad bei Include-Files war bisher Installationsverzeichnis, jetzt Pfad der Quelltextdatei
- Bug bei SAP-Anweisungen per \$-Operator behoben (\$end)

Version 2.5.3.47:

- Anfahren einer Ruheposition ungleich 0 nach Referenzierung ist nun möglich
- Bug bei der Verwendung von G01/02/03 nach Sprüngen und in Unterprogrammen behoben.
- Bug, wenn AutoSetNum gesetzt ist und Befehle mit Selbsthaltung verwendet wurden
- Ab RWMOS V2.5.3.47 werden Hardware-Referenzschalter im axst-Register angezeigt. Diese Eigenschaft wird in der Referenzfahrt ab sofort verwendet. Deshalb wurde die Mindestversion erhöht auf 2.5.3.46.
- Probleme bei Kommandos mit mehreren Parametern behoben
Runde Klammern wurden nicht immer als Kommentar erkannt (z.B. nach G04)
Leerzeichen nach L und O nun nicht mehr erforderlich
- Referenzfahrt ergänzt zum Freifahren von Referenzschalter bei Bedarf.
- Verbesserungen in Bezug auf Spooler-Abarbeitung, insbesondere beim Aufruf von Unterprogrammen und bei Ausführung von G200.299 / M200..299
- Fehler in McuWIN behoben: Die Einstellungen der Referenzposition wurde in bestimmten Fällen zerstört.
- Teach-in ist nun auch ohne Zeilennummern möglich.
- Probleme bei Verwendung von unären Operatoren (+/-) behoben.

Version 2.5.3.41:

Die Sprungbefehle M96, M98 sowie M17 und Deklaration von Labels unterbrechen ab dieser Version eine Spoolerkontur (G01, G02, G03) nicht mehr.

Version 2.5.3.40:

Die in IniCfg eingestellten Referenzpositionen wurden bisher über den sogenannten Common-Buffer an die Steuerung übertragen. Dies wird nun über die Common-Variablen CD60 – CD67 gemacht. Deshalb muss unbedingt gewährleistet sein, dass diese Variable vom Anwender nicht benutzt werden.

Version 2.5.3.38:

- neue Kommandos WRITE und WRITELN für Stringausgabe
- Anzeige der Stringausgabe in Oberfläche von McuWIN
- vollständige Einbindung von rw_SymPas-Mechanismen in G-Code Programme möglich
- Definition eigener M-Codes ist nun auch mit Parameterauswertung möglich
- G04 Parameter kann nun auch per Variable oder Ausdruck angegeben werden

Version 2.5.3.32:

- S-Kommando in einer Zeile mit G-Codes wurde z.T. fehlerhaft ausgeführt
Ablauf / Anforderungen bei S-Kommando wesentlich geändert (siehe Doku).

Version 2.5.3.31:

- div. Änderungen in Editor Anzeige
- Hauptspindel wird bei Programm-Stopp ausgeschaltet
- Option NoTriangle bei Look-Ahead eröffnet

Version 2.5.3.30:

- Case-Insensitive Modus ist nun möglich, einstellbar in IniCfg
- Programmstopp auch in Referenzierung und im Single-Step Modus möglich
- Position- und Time-Unit von Interpolationsbefehlen wird nun in IniCfg gewählt
- neuer Befehl M150
- Bug bei G-Code Programmen ohne M30 am Ende behoben
- Modulname in Programmen nicht mehr erforderlich

Version 2.5.3.25:

Bei der Verwendung von Kommentaren traten z.T. Übersetzungsfehler auf, insbesondere bei Kommentaren per Hochkomma in Zeilen mit G01, G02 oder G03. Des Weiteren ist nun die Schachtelung von Kommentaren mit runden Klammern möglich.

Version 2.5.3.24:

- für Freifahrbewegungen kann ein Geschwindigkeitsfaktor angegeben werden (FreeingVelFactor)
- Applikationsspezifische Referenzfahrt kann nun in IniCfg angewählt werden.
- Konfigurationsvariable ReferenceSwitchLatch um Überprüfung des Referenzschalter-Latch zu deaktivieren
- Konfigurationsvariable NoRefToLimitSwitch um eine Referenzierung nur auf die Indexspur zu ermöglichen
- Ein kundenspezifisches Logo kann nun in die Programm-Oberfläche eingebunden werden
- Die Notwendigkeit von Zeilennummern kann in IniCFG abgeschaltet werden

Version 2.5.3.23:

- Neue Kommandos:

- G153: Nullpunktverschiebung lesen
- G154: Nullpunktverschiebung reprogrammieren
- Nullpunktverschiebung, insbesondere G54..G59 überarbeitet, G59 existiert nicht mehr, da nur 5 Nullpunktverschiebungsregister existieren

- Bug bei Verwendung nicht fortlaufender Interpolationsachsen behoben;
- Programmstart im Simulationsmodus möglich;
- Zählerüberwachung per Index-Latch und Encoder-Error-Überwachung sind nun gesondert selektierbar;
- Bug in Override-Einstellung beim Start der Referenzfahrt behoben;

Version 2.5.3.19:

Neue Kommandos G98, G99, G161, G162.

Weiterhin wurden die Kommandos G94 und G70/G71 teilweise fehlerhaft behandelt. Hier können sich mit der aktualisierten Version Änderungen in Geschwindigkeit und Beschleunigung ergeben. Dies ist nach einem Update unbedingt zu untersuchen und ggf. zu berichtigen. (25.06.2004)

Version 2.5.3.18:

Ab dieser Version ist es möglich auch im Absolutmodus (G90) Kreismittelpunktskoordinaten relativ anzugeben.

Version 2.5.3.16:

In IniCfg ist die Angabe einer LookAhead-Tiefe möglich. Sobald die Anzahl der entsprechenden Interpolationsdatensätze programmiert wurde (G01, G02, G03) wird automatisch mit der Abarbeitung begonnen.

Weiterhin wird bei einer Einzelschrittausführung die Abarbeitung sofort gestartet. Ein Klick in das Editorfenster zum Start der Achsen ist nicht mehr notwendig. Splines werden dadurch aber nur noch als Geradenstücke abgefahren.

Version 2.5.3.15:

Die Kommandos G21 / G22 / G23 und G24 für Achsspiegelungen sind neu. Ebenso die Verwendung von Positionsfaktoren mit G39.

Version 2.5.3.14:

Das Kommando G54 wählt nun eine Nullpunktverschiebung, aus einer Nullpunktverschiebungstabelle, ebenso G55 bis G59. Ein freies Setzen der Nullpunktverschiebung ist nun mit G92 möglich. Die bisherige Funktion von G54 wurde also durch G92 ersetzt.