POSITIONIER- UND BAHNSTEUERUNG
MCU-3000 / MCU-3100
MCU-3400C
BEDIENUNGS-HANDBUCH / BHB

Stand: 22.04.2015, ab Disk V2.53VD
Board-Revision: MCU-3000 Rev. B/C/D MCU-3400C Rev. A MCU-3100 Rev. C

1 Einführung 4

2 System-Hardware 5

2.1 Die Baugruppe MCU-3000 / MCU-3100 5
2.1.1 Die MCU-3x00 - Interface-Logik 5
2.1.2 Anschaltung der externen Systemkomponenten 6
2.2 Externe Systemkomponenten 6
2.3 Typen von Leistungsverstärkern 7
2.3.1 Drehzahlregler 7
2.3.2 Stromverstärker 7
2.3.3 Spannungsverstärker 7
2.3.4 Schrittmotorleistungsverstärker 7
2.4 Positions-Latch-Funktionalität 8
2.4.1 Hardware Latch Strobe 8
2.4.2 Software Latch Strobe 8
2.4.3 Nullspur Hardware Latch Strobe 8

3 Einführung in die Bedienung / Programmierung der MCU-3x00 9

3.1 Manuelle Bedienung 9
3.2 Programmierung durch ein PC-Applikations-Programm (PCAP) 9
3.3 Programmierung als Standalone-System (SAP) 10

4 System-Software 11

4.1 Die Betriebssystemsoftware rw_MOS (rwmos.elf) 11
4.1.1 PIDF-Filter-Modul 13
4.1.2 Rampen- und Interpolations-Modul 13
4.1.3 PC-Schnittstellen-Modul 13
4.1.4 Standalone-CNC-Modul 13
4.1.5 Booten von rw_MOS 14
4.2 Das Hilfsprogramm *mcfg.exe* 14
4.3 Das Hilfsprogramm *ncc.exe* 15
4.4 Das Hilfsprogramm *fwsetup.exe* 15
4.5 Das Hilfsprogramm McuControlInit.exe 17

1 Einführung

Die Baugruppen MCU-3000 / MCU-3100 / MCU-3400C (Motion-Control-Unit) sind universelle Positionier- und Bahn-Steuerungen für Werkzeugmaschinen, Roboter, Handhabungsgeräte und Sondermaschinen. Über die Eingabe von CNC-Programmen und Parametern können Bewegungsabläufe und Prozessabläufe automatisiert werden.
Die MCU-3x00 ist als PCI-Erweiterungskarte für die Rechnerfamilie IBM-AT, IPC (Industrial Personal Computer) oder Kompatible ausgeführt und dient zur Ansteuerung von 1 bis 3 CNC-Achsen mit wahlweise Servo- oder Schritt-Motoren. Die MCU-3400C ist eine Compact-PCI-Variante mit 1 bis 4 Achsen.
Bei der Entwicklung wurde besonderes Augenmerk auf die Tauglichkeit für den Einsatz in industrieller Umgebung gelegt, da gerade in diesem Bereich der Personal-Computer immer größere Akzeptanz findet.
Das vorliegende Handbuch beschreibt im wesentlichen die Hard- und Software-Komponenten des im Standard-Lieferumfang enthaltenen Zubehörs und gliedert sich in drei Teile:

- **BHB**: Bedienungs-Handbuch
- **PHB**: Programmier- und Referenzhandbuch
- **IHB**: Inbetriebnahme-Handbuch

Im *BHB* werden die im Standard-Lieferumfang verfügbaren MCU-3x00 Hard- und Softwarekomponenten beschrieben. Der Anwender erhält einen Überblick über die Funktionsweise, Betriebsarten und Programmiermethoden der MCU-3x00.

Im *PHB* werden die verschiedenen Arten der Programmierung beschrieben. Eine Befehls- bzw. Referenzliste vervollständigt diesen Abschnitt.

Im *IHB* erfolgt eine Beschreibung zur Installation und Inbetriebnahme der im Lieferumfang enthaltenen Systemkomponenten.

Zum besseren Verständnis wird empfohlen, zuerst das Bedienungs-Handbuch durchzuarbeiten.

2 System-Hardware

2.1 Die Baugruppe MCU-3000 / MCU-3100

Die MCU-3x00 ist eine PCI-Einsteckkarte für Personalcomputer oder Kompatible des Typs AT oder IPC (Industrial Personal Computer mit AT-Schnittstellenfunktionalität) und stellt die Intelligenz der Steuerung dar. Die minimale Ausbaustufe, bestehend aus einer MCU-3x00, unterstützt drei CNC-Achskanäle mit wahlweise Schritt- (Stepper) oder Servo-Antrieben (DC oder AC/EC). Die maximale Ausbaustufe unterstützt acht Achskanäle. Hierzu wird ein Tochterboard (OPMF-3001 / OPMF-3100) für maximal 5 weitere CNC-Achsen in Sandwichtechnik auf die MCU-3x00 aufgesteckt.

Die MCU-3x00 ist auf einer kurzen PCI-Einsteckkarte für IBM AT oder Kompatible aufgebaut. Die Kommunikation zwischen PC und MCU-3x00 erfolgt über den PCI-Bus. Dadurch wird ein hoher Datendurchsatz für die Ausgabe von Befehlen und das Einlesen von Statusinformationen gewährleistet. Die Adresse der MCU-3x00 wird vom Plug+Play Bios des PCs verwaltet.
Die MCU-3400C ist auf einer Compact PCI-Karte 100x160mm in Einschubtechnik mit einem 78poligen SUB-D Anschlussstecker ausgeführt. Die Ausführungen dieses Handbuchs gelten für beide Geräte. Bei Unterschieden wird auf diese hingewiesen.

2.1.1 Die MCU-3x00 - Interface-Logik

Ein wesentliches Merkmal der MCU-3x00-Baugruppe ist der Einsatz von modernster PLD und FPGA-Technik. FPGA-Bausteine sind im Betrieb beliebig oft umprogrammierbare Logikbausteine mit einer hohen Funktionalität. In diesen befindet sich die Logik für die Istwerterfassung der Meßsysteme. Dazu gehört u.a. die Verarbeitung von vielen gängigen Inkrementalenkodern, die Verarbeitung von SSI-Absolutwertgebern, das präzise echtzeitfähige Zwischenspeichern von Positionswerten aufgrund verschiedener Bedingungen. Natürlich können auch individuell auf die Applikation abgestimmte Anpassungen realisiert werden.
Die komplette I/O-Peripherie ist galvanisch von der Systemelektronik (PC-Logik) getrennt. Die Anschaltung der externen Komponenten erfolgt nach gängigen Richtlinien der Industrie-Elektronik.

2.1.2 Anschaltung der externen Systemkomponenten

Der Anschluss an die in [Kapitel 2.2] erwähnten externen Systemkomponenten wie Leistungsverstärker, Inkrementalgeber, Endschalter, Ein- bzw. Ausgänge usw. erfolgt an einem 50poligen SUB-D-Stiftstecker an der Frontseite der MCU-3x00.
An diesem Steckverbinder können drei CNC-Achsen mit wahlweise Schritt- oder Gleichstrom-Motoren angeschlossen werden.
Die MCU-3x00 stellt drei Analog-Kanäle mit einem Ausgangsspannungsbereich von +/-10V und 16bit Auflösung zur Verfügung. Diese sind von der digitalen Stromversorgung der MCU-3x00 (PC-Stromversorgung) galvanisch getrennt und dienen zur Ansteuerung von handelsüblichen Leistungsverstärkern, die als Drehzahl- oder Stromregler geschaltet sind.
Drei Schritt- und Richtungssignale mit jeweils antivalentem Signalpegel und einer Steuerspannung von 5V (RS422) werden zur Ansteuerung von Schrittmotorleistungsverstärkern zur Verfügung gestellt.
Drei Impulserfassungskanäle mit jeweils 32bit-Registerbreite dienen zur Positionserfassung mit Hilfe von Inkrementalenkodern bzw. Längenmaßstäben. Die Impulserfassung erfolgt wahlweise für TTL-Geber bzw. für Geber mit symmetrischen Ausgängen (RS-422). Die von diesen Gebern erzeugten, um 90 Grad phasenverschobenen Quadratursignale, werden elektronisch vervierfacht und sind ebenfalls galvanisch von der Systemelektronik getrennt werden. Die maximale Impulseingangsfrequenz beträgt 2.00MHz. Ein Nullimpuls kann von der MCU-3x00 ebenfalls ausgewertet werden. Die RS-422 Eingangslogik der MCU-3100 ist mit einer Leitungsbrucherkennung ausgestattet.
Die Istposition des Meßsystems kann mit Hilfe des Nullimpulses oder eines digitalen Eingangs in Registern zwischengespeichert werden. Diese Meßmethoden gestatten u.a. eine einfache Enkoderverifikation, das Setzen von Referenzmarken oder das echtzeitfähige Positionslatchen bei Messmaschinen.
Alternativ zu den oben erwähnten Inkrementalgebern können auch alle marktgängigen SSI-Absolutwertgeber ausgewertet werden.
Die MCU-3x00 gestattet den Anschluss von 16 optisch entkoppelten Eingängen und 8 optisch entkoppelten Ausgängen. Die Ein- und Ausgänge sind nicht achsspezifisch gruppiert. Eine achsspezifische Zuordnung kann jedoch durch softwaremäßige Projektierung der Ein- und Ausgänge wahlweise mit speziellen Funktionen wie Endschalterfunktion, Referenzschalterfunktion, Leistungsverstärker-Freigabe usw. vorgegeben werden. Alle nicht projektierten Ein- und Ausgänge können frei programmiert werden.
Eine Watchdogschaltung sorgt für sichere Betriebszustände auch in Ausnahmesituationen. Ein Flash-Speicher dient zur Aufnahme von verschiedenen Betriebsparametern. Dazu gehört beispielsweise der Sollzustand der Ausgänge nach der Initialisierung des Systems.

2.2 Externe Systemkomponenten

Aufgrund der digitalen Signalverarbeitung und den auf der MCU-3x00 vorhandenen Standard-Soll- und Ist-Wert-Schnittstellen, kann die MCU-3x00 unabhängig vom Leistungsbereich mit verschiedenen Motortypen und Leistungsverstärkern eingesetzt werden.
Die Auswahl der externen Komponenten richtet sich nach der Applikation und kann unter Berücksichtigung der Gesichtspunkte Leistungsklasse, Funktionalität, Sicherheit und Wirtschaftlichkeit erfolgen.
Es können alle handelsüblichen Leistungsverstärker mit Hilfe eines +/-10V Sollwertkanals angesteuert werden. Die Leistungsverstärker können als Drehzahlregler oder als Stromverstärker geschaltet sein.
Bei der Auswahl der Motoren kommen sowohl Schritt- als auch Servomotoren in Frage. Alle Motoren mit Positionsrückmeldung werden dabei als Servomotoren bezeichnet. Dazu gehören bürstenlose (AC oder EC) bzw. bürstenbehaftete Gleichstrommotoren (DC) oder Hydraulikmotoren. Zur Positionierung dieser Motortypen ist eine Positionserfassung notwendig. Im allgemeinen dienen hierzu Drehgeber oder Linearmaßstäbe. Zur Ansteuerung von Schrittmotorleistungsendstufen stehen Puls- und Richtungssignale zur Verfügung. Im Gegensatz zum Servo-Antrieb kann der Schrittmotor-Antrieb ohne Positionsrückmeldung verfahren werden.
Weitere externe Systemkomponenten können über digitale Ein- oder Ausgänge mit der MCU-3x00 in Verbindung stehen. Durch softwaremäßige Projektierung dieser Ein- bzw. Ausgänge können z.B. Endschalter-, Referenzschalter-, Not-Aus-, Verstärkerfreigabe- und andere Funktionen realisiert werden.

2.3 Typen von Leistungsverstärkern

Zur Bereitstellung der Energie für den elektrischen Antrieb ist der Einsatz verschiedener Typen von Leistungsendstufen möglich. Alle nachfolgend aufgeführten Typen können von der MCU-3x00 direkt angesteuert werden.
Die unterschiedlichen Typen von Leistungsverstärkern bestimmen unter anderem auch das charakteristische Regelverhalten der Regelstrecken, welches bei der Einstellung der Filterparameter [IHB / Kapitel 6.2] berücksichtigt werden muß.
Anmerkung: Die meisten Leistungsverstärker, außer den Schrittmotorleistungsverstärkern, werden üblicherweise durch die Analog-Sollwert-Kanäle der MCU-3x00-Baugruppe angesteuert. Für den Einsatz von Schrittmotorleistungsverstärkern stehen Puls- und Richtungssignale sowie deren invertierte Impulsfolgen zur Verfügung. Diese Schnittstelle, wird jedoch manchmal auch bei Servosystemen mit digitalen Leistungsverstärkern eingesetzt. In diesem Fall muss auf dem Leistungsverstärker ein Lageregler dimensioniert werden. Die Projektierung der MCU-3x00 muss dann wie bei einem Schrittmotor erfolgen.

2.3.1 Drehzahlregler

Dies sind die üblichen, am Markt erhältlichen Geräte mit denen die Drehzahl eines Gleichstrom- oder bürstenlosen Gleichstrommotors geregelt werden kann. Eingangsgröße ist hier meist eine Spannung von +/- 10V die einer Drehzahl von +/- Maximaldrehzahl entspricht. Der Drehzahlistwert wird, meist als Tachosignal, direkt dem Drehzahlregler zugeführt. Bei freigegebenem Verstärker entwickelt der Drehzahlregler auch ohne Sollwertsignal ein Haltemoment. I.a. driftet die Motorachse durch den Eingangsoffset mehr oder weniger langsam weg. Der Eingangsoffset kann normalerweise eingestellt werden, ist jedoch temperaturabhängig.

2.3.2 Stromverstärker

Im Gegensatz zum Drehzahlregler benötigt der Stromverstärker kein Tachosignal als Istwertrückführung vom Motor. Der Eingangssollwert hat hier die Bedeutung eines Ankerstromes. Viele Hersteller bieten ihre Leistungsverstärker als Stromverstärker oder Drehzahlregler an. Üblicherweise lassen sich die handelsüblichen Drehzahlregler mit wenigen Handgriffen zum Stromverstärker umbauen. Hierbei sind die jeweiligen Herstellerangaben zu beachten. Der Verstärkungsfaktor sollte so eingestellt sein, daß der gewünschte Maximalstrom dem Eingangssollwert 10V entspricht.

2.3.3 Spannungsverstärker

Im Gegensatz zum Drehzahlregler benötigt der Spannungsverstärker kein Tachosignal als Istwertrückführung vom Motor. Der Eingangssollwert hat hier die Bedeutung einer Ankerspannung. Spannungsverstärker haben in der Praxis keine wesentliche Bedeutung und werden allenfalls bei kleinen Leistungen eingesetzt, wenn der Motorstrom durch den Ankerwiderstand begrenzt wird. Hierbei muß beachtet werden, daß die verfügbare Beschleunigung zu höheren Drehzahlen hin abnimmt.

2.3.4 Schrittmotorleistungsverstärker

Schrittmotoren haben meist 2, 3 oder 5 Phasen bzw. anzusteuernde Wicklungen. Damit eine Drehbewegung erfolgt, muß in den einzelnen Motorwicklungen in einem festgelegten Zyklus Strom fließen. Alle gängigen am Markt erhältlichen Schrittmotorleistungsendstufen erfüllen diese Anforderungen üblicherweise und werden mit den Eingangssignalen Puls (pro Puls ein Motorschritt) und Richtung (legt Drehrichtung der Motorwelle fest) angesteuert. Aus dieser Eingangsinformation resultiert der oben beschriebene Stromfluss.

2.4 Positions-Latch-Funktionalität

2.4.1 Hardware Latch Strobe

Die MCU-3x00 stellt verschiedene Funktionen zur Verfügung um Positions-Istwerte per Hardwaresignal zwischenzuspeichern. Bei Servosystemen mit Inkrementalgeber-Interface steht hierzu der sogenannte Hardware-Latch (Latch-Strobe) zur Verfügung. Jedem Achskanal ist ein schneller Digital-Eingang zugeordnet. Die Aktivierung dieses Eingangs bewirkt ein Latchen des aktuellen Zählerstandes im Positionsregister LP (siehe [PHB / Kapitel 4.4.90]). Gleichzeitig wird das Bit STRBL im Register der digitalen Eingänge beim jeweiligen Achskanal gesetzt. Die Erkennung des Hardwaresignals erfolgt flankengesteuert, d.h. auch kurze Impulse (bis ca. 50ns) werden sicher erkannt. Der eigentliche Latchvorgang erfolgt nahezu verzögerungsfrei (< 200ns). Das Signalbit STRBL muss ggf. manuell zurückgesetzt werden durch beschreiben des Registers der digitalen Eingänge (SAP-Programmierung) oder mit Hilfe der PCAP-Funktion rdigi. Die Polarität der Flankenerkennung kann in mcfg invertiert werden.

2.4.2 Software Latch Strobe

Ergänzend zu dieser Hardware-Funktionalität kann auch ein digitaler Eingang der Eingangsgruppe eines Achskanals als Software-Latcheingang deklariert werden (im Programm mcfg.exe). In diesem Fall wird der jeweilige Eingang einmal pro Abtastintervall statisch abgefragt. Diese Funktionalität ist auch bei Schrittmotorsystemen und bei anderen Impulserfassungssystemen als Inkrementalgebern z.B. SSI verwendbar. Allerdings handelt es sich hierbei um eine reine Softwarefunktion, d.h. hier ist eine Verzögerung von max. einem Abtastinterfall zwischen Impulsflanke und Latchzeitpunkt möglich. Weiterhin ist es möglich, dass kurze Impulse (< 1 Abtastintervall) nicht erkannt werden. Die Erkennung eines Latchsignals wird im Bit LPSF des achsspezifischen AXST-Registers angezeigt. Durch Auslesen des gelatchten Positionswertes im Positionsregister LP wird dieses Bit automatisch gelöscht.
Vorsicht: Bei gleichzeitiger Verwendung von Hard- und Software-Positionslatch bei einer Achse sind Konflikte möglich, weil immer das Positionsregister LP zum Auslesen der Latchposition verwendet wird. Bei Servosystemen mit Inkementalenkoder-Interface muss deshalb immer die Hardware-Latch-Funktionalität verwendet werden, da in diesem Fall diese immer aktiv ist und nicht deaktiviert werden kann.

2.4.3 Nullspur Hardware Latch Strobe

Bei Servosystemen mit Inkrementalgeber-Interface steht zusätzlich die Möglichkeit des sogenannten Nullspur-Latch zur Verfügung. Die Aktivierung des Nullspur-Signals des Inkrementalgebers bewirkt ein Latchen des aktuellen Zählerstandes im Positionsregister LPNDX (siehe [PHB / Kapitel 4.4.91]) Gleichzeitig wird das Bit NDXL im Register der digitalen Eingänge beim jeweiligen Achskanal gesetzt. Die Erkennung des Hardwaresignals erfolgt flankengesteuert, d.h. auch kurze Impulse (bis ca. 50ns) werden sicher erkannt. Der eigentliche Latchvorgang erfolgt nahezu verzögerungsfrei (< 200ns). Das Signalbit NDXL muss ggf. manuell zurückgesetzt werden durch beschreiben des Registers der digitalen Eingänge (SAP-Programmierung) oder mit Hilfe der PCAP-Funktion rdigi. Die Polarität der Flankenerkennung kann in mcfg invertiert werden.

3 Einführung in die Bedienung / Programmierung der MCU-3x00

Die Bedienung und Programmierung der MCU-3x00 kann auf mehrere Arten erfolgen. Alle Tools und Treiber sind für die 32-bit Windows-Betriebssysteme von Microsoft zugeschnitten (Windows 98, NT, 2000, XP, VISTA und Windows 7).

3.1 Manuelle Bedienung

Die manuelle Bedienung der MCU-3x00 erfolgt im wesentlichen mit dem Hilfsprogramm *mcfg.exe*, welches Bestandteil der MCU-3000 / MCU-3100 TOOLSET Software [TSW] ist [Kapitel 2.2].
Dieses bietet dem Anwender eine Vielzahl von Betriebsarten zur Bedienung der MCU-3x00, unter anderem auch die manuelle Bedienung des kompletten Achssystems.

3.2 Programmierung durch ein PC-Applikations-Programm (PCAP)

Der Anwender erstellt ein auf dem PC ablauffähiges Anwenderprogramm in einer höheren Programmiersprache wie *C* oder *Pascal*. Mit Hilfe von Funktionenbibliotheken werden aus diesem Applikationsprogramm über einen 32-bit DLL-Treiber mcug3.dll Kommandos auf der MCU-3x00 ausgeführt. Diese DLL benötigt wiederum einen Hardwaretreiber (Miniporttreiber), welcher die Zugriffe über den PCI-Bus ermöglicht. Das PC-Anwenderprogramm ist in dieser Betriebsart für den koordinierten Ablauf der einzelnen Achssysteme verantwortlich. Das PC-Applikations-Programm wird im weiteren mit PCAP bezeichnet.
Die Befehlsübergabe erfolgt mit Hilfe vordefinierter Befehle. Diese wiederum sind als DLL-Funktionen für 32bit Windows realisiert. Für die oben genannten Programmiersprachen existiert jeweils eine Source- oder Header-Datei und ggf. entsprechende Lib-Files.
Aus Anwendersicht stellen die DLL-Funktionen lediglich eine Funktionserweiterung der entsprechenden Programmiersprache dar. Die eigentliche „Intelligenz“ der Funktionen liegt im DLL-Treiber bzw. auf der Steuerung.

3.3 Programmierung als Standalone-System (SAP)

Eine weitere Möglichkeit ist die Programmerstellung für die sogenannte Standalone-Betriebsart. Diese Betriebsart gestattet es, ein zuvor auf die MCU-3x00 geladenes Betriebsprogramm automatisch, also ohne Unterstützung durch ein PCAP, auszuführen. Dies bedeutet, daß der PC andere Aufgaben übernehmen kann. Dieses Standalone-Applikations-Programm wird im weiteren mit SAP bezeichnet.
Das SAP wird durch den Anwender mit Hilfe eines Editors oder mit Hilfe des im Hilfsprogramm *mcfg.exe* integrierten Editors CNC-Edit erstellt. Die Syntax dieses Anwenderprogramms ist *Pascal*-ähnlich und erlaubt eine einfache und flexible Programmerstellung. Aus dem so erstellten Anwenderprogramm wird mit dem Compiler *NCC*, welcher zum einen als Kommandozeilencompiler zur Verfügung steht [Kapitel 4.3] und zum anderen im Hilfsprogramm *mcfg.exe* integriert ist, ein sogenanntes Autocode-File erzeugt. Dieses Autocode-File kann auf die MCU-3x00 übertragen und dort automatisch, d.h. ohne Unterstützung durch den PC, ausgeführt werden. Hierbei ist zu beachten, dass das jeweilige CNC File für jeden Steuerungstyp übersetzt sein muss. Der Dateityp ergibt sich automatisch aus dem Typ der Systemdatei SYSTEM.DAT, welcher während des Übersetzungsvorgangs vorliegt.
Ist eine Synchronisation zu einem evtl. parallel ablaufenden PCAP notwendig, so kann diese mit Hilfe gemeinsamer vordefinierter Variablen, welche sowohl für den PC als auch für die MCU-3x00 schreibend und lesend zugänglich sind, erfolgen.

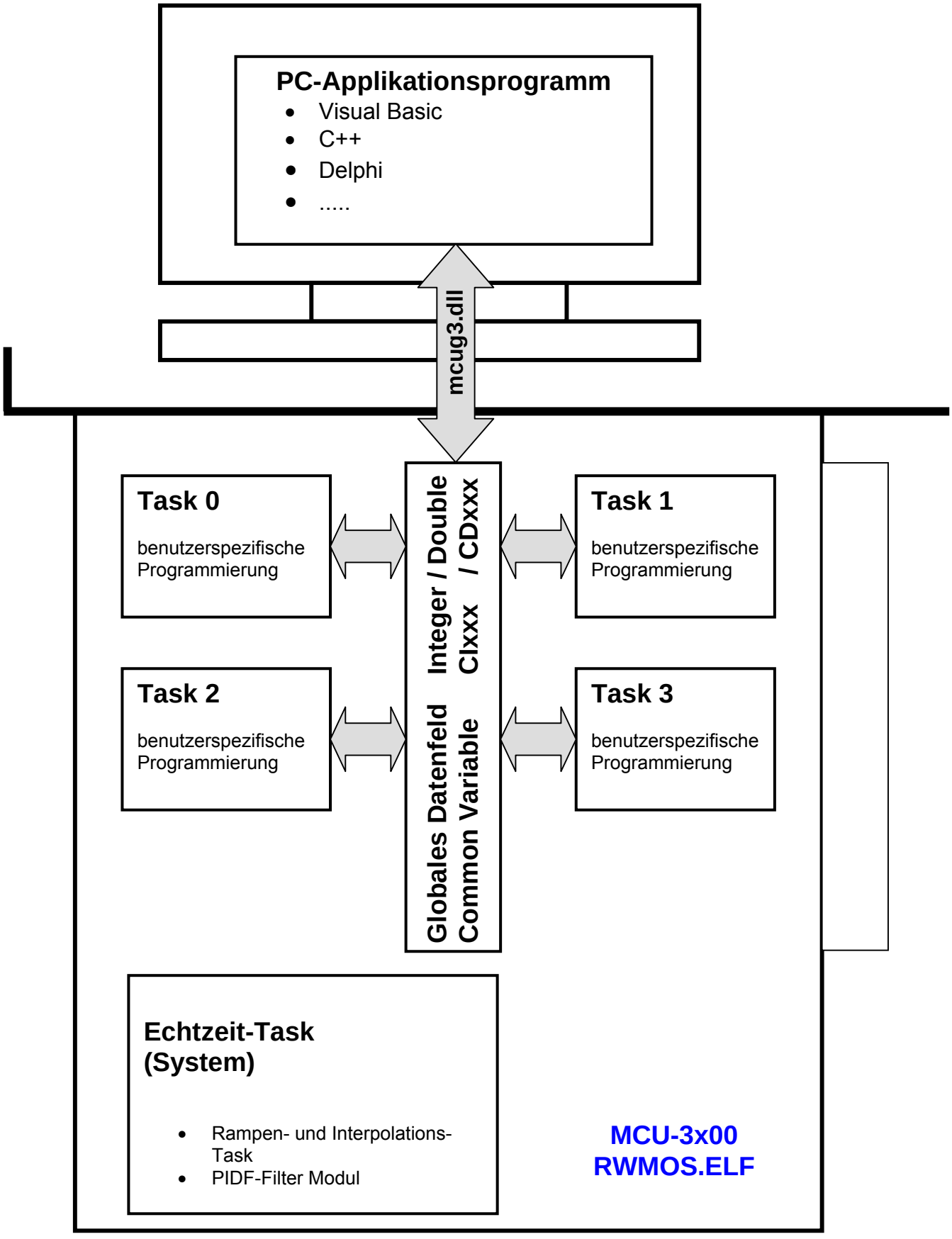
4 System-Software

In diesem Kapitel werden die Hilfsprogramme der im Standard-Lieferumfang enthaltenen MCU-3000 / MCU-3100 TOOLSET Software [TSW] beschrieben. Diese Informationen sind wichtig für die korrekte Bedienung der Steuerung und zur Erstellung von Anwenderprogrammen.

4.1 Die Betriebssystemsoftware rw_MOS (rwmos.elf)

Die Betriebssystemsoftware *rw_MOS* ist in der Datei rwmos.elf enthalten und beinhaltet den ausführbaren Maschinencode für das MCU-3000 bzw. MCU-3100 CPU-System. In *rw_MOS* sind verschiedene Softwaremodule enthalten, welche für die Funktionalität der Steuerung benötigt werden. Zum tieferen Verständnis werden für den interessierten Anwender die wichtigsten Softwaremodule in den nachfolgend aufgeführten Kapiteln erläutert. Für jeden Steuerungstyp ist ein geeignetes RWMOS.ELF File notwendig, d.h. diese Files sind für die verschiedenen Steuerungsvarianten nicht austauschbar.

Bild: MCU-3000 / MCU-3100 / MCU-3400C Betriebssystem



4.1.1 PIDF-Filter-Modul

Dieses Modul führt die Regelung der Motoren auf eine gewünschte Sollwert-Position oder -Drehzahl mit Hilfe eines digitalen PIDF-Filters durch. Dieser Filter wird durch Abarbeiten in kurzen zeitgleichen Abständen dem analogen Filter nachempfunden. Die Regelzykluszeit ist auf 1.28ms für alle Achskanäle eingestellt, kann jedoch in gewissen Grenzen variiert werden (100µs – 5ms). Alle für die Filterberechnung notwendigen Istwert-Daten werden ebenfalls in diesem Abtastzyklus eingelesen und verarbeitet. Anschließend erfolgt die Sollwertausgabe auf die entsprechenden Sollwertkanäle.
Anmerkung: Das PIDF-Filter wird sowohl für Servo- als auch für Schrittmotoren ausgeführt. D.h. auch bei Schrittmotoren muss vor der Verwendung der Achsen der jeweilige Regelkreis geschlossen werden. Hier erfolgt die Lageregelung auf die tatsächlich ausgegebenen Schrittimpulse. Eine externe Positionsrückmeldung ist nicht erforderlich.

4.1.2 Rampen- und Interpolations-Modul

Die Motoren werden mit sogenannten Rampenfunktionen mit definierter Beschleunigung und vorgegebener Maximalgeschwindigkeit in eine gewünschte Zielposition gebracht. Diese Aufgabe übernimmt das Rampen- und Interpolations-Modul. Dieses Modul ist mit dem PIDF-Filter-Modul synchronisiert und wird ebenfalls für alle Achsen einmal pro Abtastzyklus (1.28ms) ausgeführt.
Eine weitere Aufgabe dieses Moduls ist es, mehrere Achsen zu synchronisieren um eine Bahn- oder Raumkurve interpoliert abarbeiten zu können. Es sind verschieden Interpolationsverfahren möglich. Dies sind Linear-, Zirkular-, Helix- und Spline-Interpolation.
Bezüglich der Auswahl der an einer Interpolation beteiligten Achsen gibt es weder bei der Standard-Version (bis 3 Achsen) als auch in der Multi-Achs-Version (mehr als 3 Achsen) keine Einschränkungen.
Der Geschwindigkeitsverlauf ist normalerweise trapezförmig und wird mit der Beschleunigung, Maximalgeschwindigkeit und Zielgeschwindigkeit parametriert. Durch Anwahl von S-Profilen kann auch ein S-förmiger Drehzahlverlauf mit trapezförmigem Beschleunigungsverlauf eingestellt werden.
Eine weitere wichtige Eigenschaft des Rampen- und Interpolations-Moduls ist das sogenannte Dynamik-Modul zweiter Ordnung. Die bei vielen konventionellen Positioniersteuerungen implementierte Rampengenerierung verlangt einen Bewegungsstopp in der gewünschten Zielposition. Diese Einschränkung wurde durch die Implementierung dieses Dynamik-Moduls aufgehoben. Der Anwender kann für die gewünschte Zielposition eine beliebige Zielgeschwindigkeit programmieren. Das Achssystem wird unter Berücksichtigung von Maximalgeschwindigkeit und Maximalbeschleunigung in die gewünschte Zielposition verfahren und erreicht genau in diesem Punkt die programmierte Zielgeschwindigkeit. Mit dieser Methode können beliebig komplizierte Bewegungsprofile erzeugt werden.
Bei der Verwendung von S-förmigen Beschleunigungsprofilen ist zu beachten, dass bei einer Unterbrechung eines Verfahrprofils z.B. durch einen erneuten Aufruf eines Jog-Verfahrkommandos, der aktuelle erreichte Wert der Beschleunigung erneut von 0 an hochgefahren wird.

4.1.3 PC-Schnittstellen-Modul

Dieses Modul ist Grundlage für die PC-Applikations-Programmierung (PCAP-Programmierung) und führt die vom Benutzer aufgerufenen Funktionen und Befehle aus. Hierzu steht eine Vielfalt von Kommandos zur Verfügung. Der Aufruf dieser Kommandos erfolgt aus dem PC-Applikations-Programm, welches in einer höheren Programmiersprache erstellt wird.

4.1.4 Standalone-CNC-Modul

Dieses Modul ist Grundlage für die Standalone-Applikations-Programmierung (SAP-Programmierung) und eine der leistungsfähigen Eigenschaften der MCU-3x00. Mit dem im Lieferumfang enthaltenen Compiler *NCC* können bis zu vier Standalone-Anwenderprogramme (SAP-Programme) erstellt werden. Diese werden durch die vier CNC-Tasks automatisch im Multitasking-Betrieb abgearbeitet. Da die Antriebskontrolle auch komplett durch ein SAP-Programm übernommen werden kann, steht der PC in diesem Fall für andere Aufgaben zur Verfügung.

4.1.5 Booten von rw_MOS

Die MCU-3x00-Betriebssystemsoftware *rw_MOS* muss nach jedem Systemstart an die MCU-3x00 übertragen werden und bewirkt eine System-Initialisierung (Hardware-Reset). Dieser Ladevorgang ist mindestens einmal pro Systemstart notwendig und innerhalb weniger Sekunden abgeschlossen. Der Vorteil des Bootens gegenüber einem ROM-residenten Betriebssystem ist, daß kundenspezifische Änderungswünsche des MCU-3x00-Betriebsprogrammes leicht berücksichtigt werden können. Sobald der Bootvorgang erfolgreich abgeschlossen wird, ist die MCU-3x00 ab diesem Zeitpunkt betriebsbereit.

4.2 Das Hilfsprogramm *mcfg.exe*

Das Hilfsprogramm *mcfg.exe* ist als 32bit-MDI-Applikation für Windows ausgeführt und bietet eine komfortable Umgebung zur SAP-Programmentwicklung sowie eine leistungsfähige Inbetriebnahme-, Diagnose- und Konfigurations-Oberfläche.
Eine detaillierte Beschreibung erfolgt in den jeweiligen Handbüchern bzw. Kapiteln. Nachfolgend wird jedoch die Einstellung der Systemdaten beschrieben, da dieser für die Erstinbetriebnahme der Steuerung unumgänglich ist.
Das Hilfsprogramm *mcfg.exe* benötigt die Dateien *system.dat* und rwmos.elf. In *system.dat* sind die benutzerspezifischen Einstellungen für das jeweilige Antriebssystem abgelegt. Bei der Installation wird eine Defaultdatei von der Installations-CD verwendet oder mit dem Hilfsprogramm sysgen.exe erzeugt.
Beim Start von *mcfg.exe* werden alle Achskanäle, bei welchen das *cef*-Error-Flag gesetzt ist, angezeigt. Dieser Fehler resultiert aus der Dateninkonsistenz zwischen System-Datei (*system.dat*) und MCU-3x00 Baugruppe(n). Der Fehler kann durch Speichern im bei aktivem [System Data]-Fenster behoben werden.
Eine ausführliche Beschreibung des Programms mcfg.exe ist im Handbuch mcfg.pdf zu finden.

4.3 Das Hilfsprogramm *ncc.exe*

Mit dem Kommandozeilen-Compiler *ncc.exe* können Stand-Alone-Applikationsprogramme (SAP-Programme) direkt aus der DOS-Ebene (Eingabeaufforderung) von Windows-32 Systemen generiert werden. Der Compiler ist identisch mit dem in *mcfg.exe* enthaltenen Compiler NCC und wird wie folgt aufgerufen:

NCC filename [Option]

Die Textdatei mit dem Namen *filename* muß ein SAP-Programm enthalten. Als Dateierweiterungsnamen wird automatisch die Endung *'.src'* angenommen. Die verschiedenen Optionen *[Option]* werden weiter unten erläutert und müssen durch Leerzeichen voneinander getrennt werden.

Sofern eine fehlerfreie Kompilierung möglich war, wird ein Autocode-File mit dem Dateinamen *filename.cnc* im aktuellen Verzeichnis angelegt. Bei Auftreten eines Fehlers wird die Kompilierung jedoch abgebrochen und die fehlerhafte Zeilennummer nebst Fehlertext am Bildschirm angezeigt.

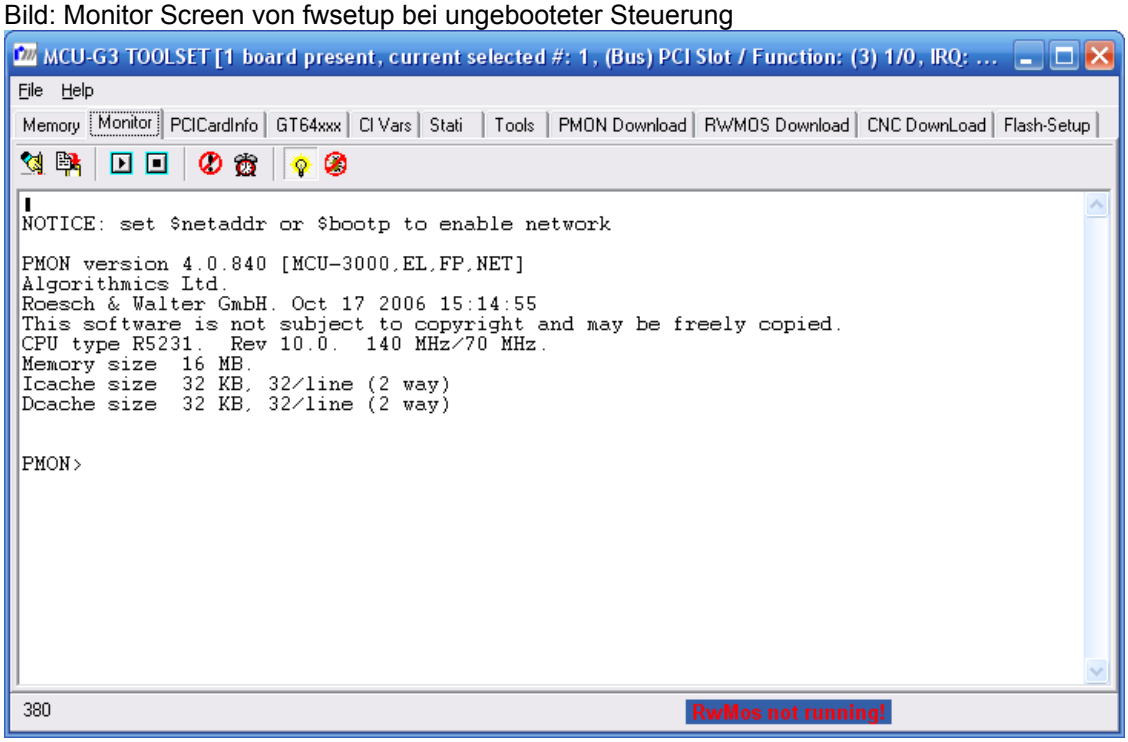
NCC Kommandozeilencompiler-Optionen *[Option]*:

Option	Funktion
FS	(Full System) es wird nicht geprüft wie viele Achsen tatsächlich im System vorhanden sind, d.h. es können bis zu 18 verschiedene Achsenbezeichner referenziert werden. Die Defaultnamen dieser Achsenbezeichner sind dabei A1 .. A18.
SC	(Syntax Check) Es wird lediglich ein Compilerlauf ohne Generierung eines CNC-Files durchgeführt. Dies dient vor allem zum syntaktischen Überprüfen des SAP-Quelltextprogramms.
TSK x	(Task Selection, x = 0..3) Die Erzeugung des Autocodefiles wird für Task x vorgenommen. Sofern das SAP-Programm jedoch einen {$TASK} Compilerbefehl enthält, ist der Parameter TSK x hinfällig.

4.4 Das Hilfsprogramm *fwsetup.exe*

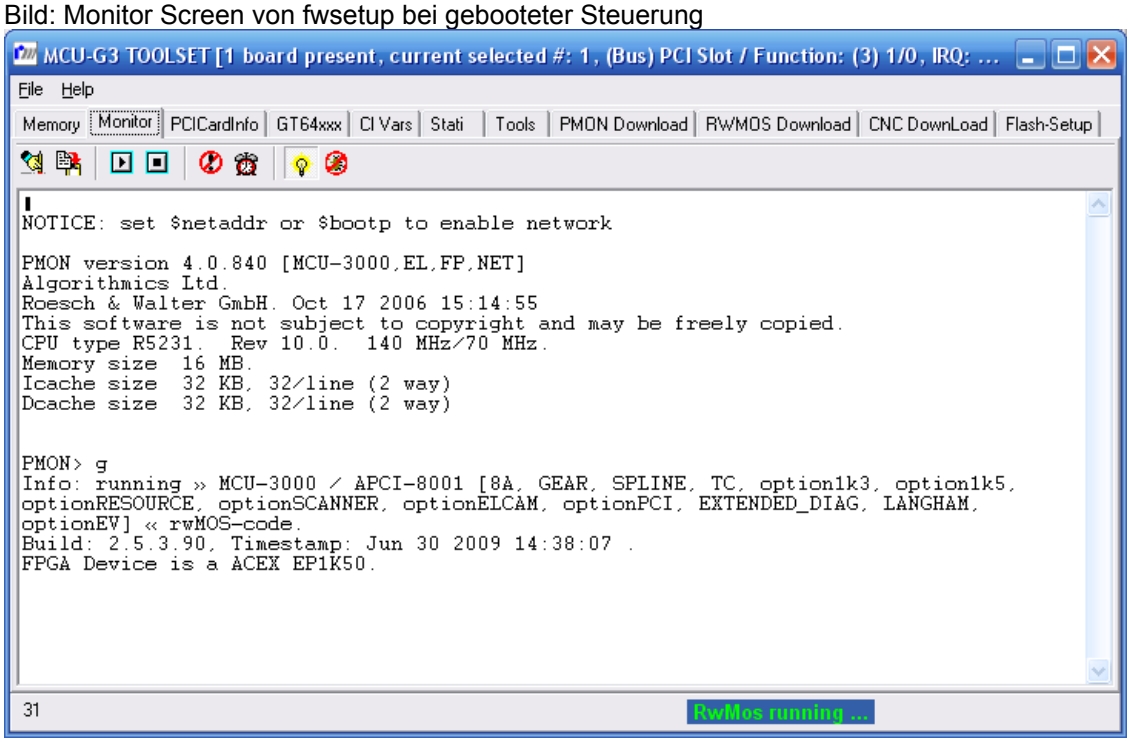
Das Programm fwsetup ist ein Monitorprogramm zum direkten Zugriff auf die Steuerungsbaugruppe. Im ungebooteten Zustand können hier Informationen über den Bootloader PMON der Steuerung angezeigt werden. Weiterhin lassen sich Umgebungsvariable zur Konfiguration der Steuerung anzeigen und verändern. So können z.B. Speicherbereiche verwaltet, Achsspezifikationen und die Abtastrate gesetzt werden.

Bild: Monitor Screen von fwsetup bei ungebooteter Steuerung



Beim Booten der Steuerung gibt RWMOS.ELF Meldungen über die Eigenschaften und Optionen aus, die im Monitorscreen verifiziert werden können. Dieser Bildschirm kann insbesondere bei Fehlerzuständen und bei Programmierproblemen hilfreiche Informationen liefern, weil hier u.U. RWMOS-interne Fehlermeldungen oder Hinweise angezeigt werden.
Auf der Registerkarte „Stati“ werden RWMOS-interne Laufzeiten und Kommunikationszustände dargestellt. Hinweise zur Handhabung von fwsetup.exe für bestimmte Anwendungsfälle, insbesondere bei der Neuinstallation sind im Handbuch PHB beschrieben.

Bild: Monitor Screen von fwsetup bei gebooteter Steuerung



4.5 Das Hilfsprogramm McuControlInit.exe

Mit Hilfe des Programmes McuControlInit.exe können die Controller-Flags der rwmos.elf Lageregler testweise zur Laufzeit angezeigt und modifiziert werden. Somit lassen sich Regleroptimierungen und Parameterermittlungen auf einfache Weise ohne Programmänderungen durchführen. Das Programm ist auf der McuG3-Toolset CD enthalten. Die Beschreibung der Flags erfolgt im Handbuch PHB im Kapitel "ControllerFlags".

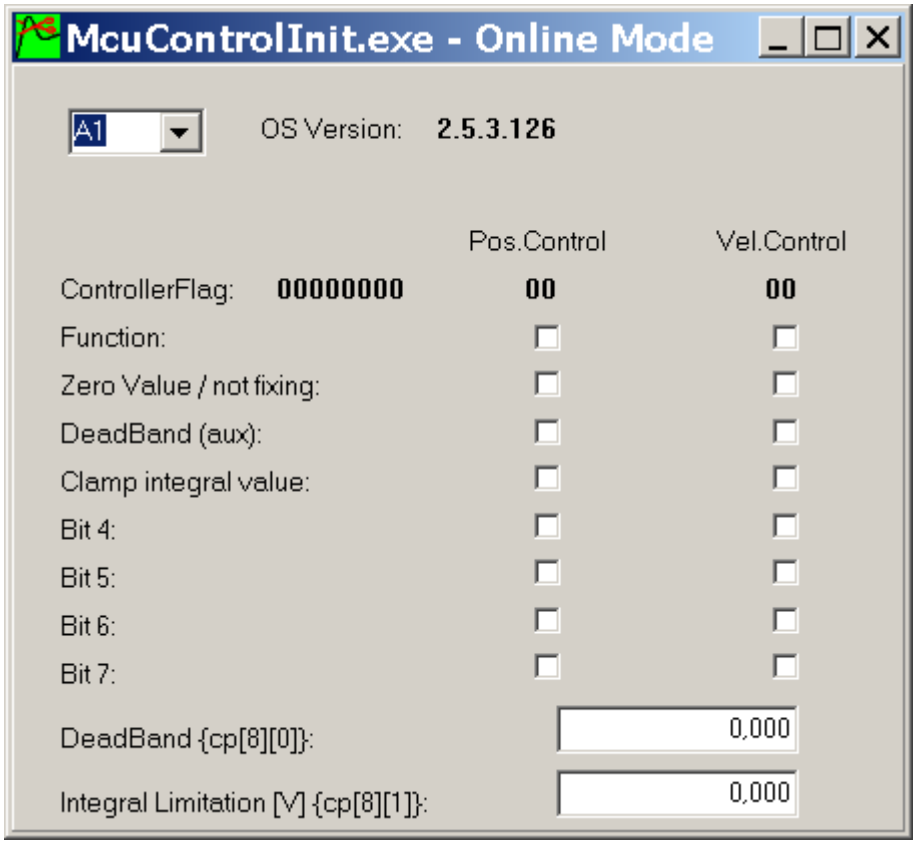


Bild: Ansicht des Hilfsprogramms McuControlInit.exe