

Produkt Handbuch



Motion Coordinator MC 2000

Urheberrechte

© 2011 Metronix Meßgeräte und Elektronik GmbH. Alle Rechte vorbehalten.

Die Informationen und Angaben in diesem Dokument sind nach bestem Wissen zusammengestellt worden. Trotzdem können abweichende Angaben zwischen dem Dokument und dem Produkt nicht mit letzter Sicherheit ausgeschlossen werden. Für die Geräte und zugehörige Programme in der dem Kunden überlassenen Fassung gewährleistet Metronix den vertragsgemäßen Gebrauch in Übereinstimmung mit der Nutzerdokumentation. Im Falle erheblicher Abweichungen von der Nutzerdokumentation ist Metronix zur Nachbesserung berechtigt und, soweit diese nicht mit unangemessenem Aufwand verbunden ist, auch verpflichtet. Eine eventuelle Gewährleistung erstreckt sich nicht auf Mängel, die durch Abweichen von den für das Gerät vorgesehenen und in der Nutzerdokumentation angegebenen Einsatzbedingungen verursacht werden.

Metronix übernimmt keine Gewähr dafür, dass die Produkte den Anforderungen und Zwecken des Erwerbers genügen oder mit anderen von ihm ausgewählten Produkten zusammenarbeiten. Metronix übernimmt keine Haftung für Folgeschäden, die im Zusammenwirken der Produkte mit anderen Produkten oder aufgrund unsachgemäßer Handhabung an Maschinen oder Anlagen entstehen.

Metronix behält sich das Recht vor, das Dokument oder das Produkt ohne vorherige Ankündigung zu ändern, zu ergänzen oder zu verbessern.

Dieses Dokument darf weder ganz noch teilweise ohne ausdrückliche Genehmigung des Urhebers in irgendeiner Form reproduziert oder in eine andere natürliche oder maschinenlesbare Sprache oder auf Datenträger übertragen werden, sei es elektronisch, mechanisch, optisch oder auf andere Weise.

Warenzeichen

Alle Produktnamen in diesem Dokument können eingetragene Warenzeichen sein. Alle Warenzeichen in diesem Dokument werden nur zur Identifikation des jeweiligen Produkts verwendet.

ServoCommander™ ist ein eingetragenes Warenzeichen der Metronix Meßgeräte und Elektronik GmbH.



Die Metronix Meßgeräte und Elektronik GmbH macht darauf aufmerksam, dass es nach dem Stand der Technik nicht möglich ist, Softwareprodukte so zu erstellen, dass sie in allen Anwendungen und Kombinationen fehlerfrei arbeiten.

Die Metronix Meßgeräte und Elektronik GmbH übernimmt keine Haftung für die Fehlerfreiheit von Vorschlägen/Beispielen und/oder Software. Insbesondere übernimmt die Metronix Meßgeräte und Elektronik GmbH keine Gewähr dafür, dass die Vorschläge/Beispiele und/oder Softwareprodukte den Anforderungen und Zwecken des Erwerbers genügt oder mit anderen von ihm ausgewählten Produkten zusammenarbeiten.

Die Metronix Meßgeräte und Elektronik GmbH übernimmt keine Haftung für Folgeschäden, die im Zusammenwirken der Vorschläge/Beispiele und/oder Softwareprodukte mit anderen Produkten oder aufgrund unsachgemäßer Handhabung an Maschinen und Anlagen entstehen.



Diese Dokumentation erhebt keinen Anspruch auf Vollständigkeit.

| Verzeichnis der Revisionen | | | |
|-----------------------------------|--|--|--------------------|
| Ersteller: | | Metronix Meßgeräte und Elektronik GmbH | |
| Handbuchname: | | First Step MC 2000 | |
| Dateiname: | | P-HB_MC2000_2p0_DE | |
| Speicherort der Datei: | | | |
| Lfd. Nr. | Beschreibung | Revisions-Index | Datum der Änderung |
| 001 | Erstellung | 0.0 | 14.04.08 |
| 002 | Vorabversion (preliminary) | 0.1 | 06.10.08 |
| 003 | Freigabeversion | 1.0 | 04.08.2009 |
| 004 | Corporate Identity aktualisiert – Keine technischen Änderungen | 2.0 | 19.04.2011 |
| 005 | | | |
| 006 | | | |
| 007 | | | |
| | | | |
| | | | |

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | ALLGEMEINES | 11 |
| 1.1 | Dokumentation | 11 |
| 2 | SICHERHEITSHINWEISE FÜR ELEKTRISCHE ANTRIEBE UND STEUERUNGEN..... | 12 |
| 2.1 | Verwendete Symbole..... | 12 |
| 2.2 | Allgemeine Hinweise | 13 |
| 2.3 | Gefahren durch falschen Gebrauch..... | 14 |
| 2.4 | Sicherheitshinweise | 15 |
| 2.4.1 | Allgemeine Sicherheitshinweise..... | 15 |
| 2.4.2 | Sicherheitshinweise bei Montage und Wartung | 17 |
| 2.4.3 | Schutz gegen Berühren elektrischer Teile | 18 |
| 2.4.4 | Schutz durch Schutzkleinspannung (PELV) gegen elektrischen Schlag..... | 19 |
| 2.4.5 | Schutz vor gefährlichen Bewegungen..... | 20 |
| 2.4.6 | Schutz gegen Berühren heißer Teile | 20 |
| 2.4.7 | Schutz bei Handhabung und Montage | 21 |
| 3 | PRODUKTBESCHREIBUNG | 22 |
| 3.1 | Allgemeines | 22 |
| 3.2 | Benötigte Komponenten | 23 |
| 3.2.1 | Hardware-Komponenten | 23 |
| 3.2.2 | Software-Komponenten | 24 |
| 4 | TECHNISCHE DATEN..... | 25 |
| 5 | MECHANISCHE INSTALLATION | 26 |
| 6 | VERKABELUNG UND STECKERBELEGUNG | 27 |
| 6.1 | Anschluss: Serielle Schnittstellen [X2MC] | 28 |
| 6.1.1 | Ausführung am Gerät [X2MC]..... | 28 |
| 6.1.2 | Gegenstecker [X2MC]..... | 28 |
| 6.1.3 | Steckerbelegung [X2MC] | 28 |
| 6.1.4 | Art und Ausführung der RS232-Schnittstellenleitung | 29 |
| 6.1.5 | Anschluss Hinweise [X2MC]..... | 29 |
| 6.2 | Anschluss: CAN-Bus [X1MC] | 30 |
| 6.2.1 | Ausführung am Gerät [X1MC]..... | 30 |

| | | |
|----------|--|-----------|
| 6.2.2 | Gegenstecker [X1MC]..... | 30 |
| 6.2.3 | Steckerbelegung [X1MC] | 31 |
| 6.2.4 | Art und Ausführung des Kabels [X1MC] | 32 |
| 6.2.5 | Anschlusshinweise [X1MC]..... | 32 |
| 7 | AKTIVIERUNG VON MC 2000 UND CANOPEN (DEVICE PROFILE FOR DRIVES AND MOTION CONTROL)..... | 35 |
| 7.1 | Übersicht | 35 |
| 7.2 | Betriebsparameter | 36 |
| 7.2.1 | Mastersystem | 36 |
| 7.2.2 | Slavesystem..... | 37 |
| 7.3 | Diagnose | 39 |
| 7.3.1 | Steuer-/Statuswort | 39 |
| 7.3.2 | PDO-Daten | 40 |
| 8 | PROGRAMMIERUNG | 42 |
| 8.1 | Einleitung..... | 42 |
| 8.2 | Kommunikationsaufbau (RS232)..... | 42 |
| 8.3 | LED Statusanzeige..... | 44 |
| 8.4 | Aufteilung der Programmausführung (Tasks)..... | 45 |
| 8.5 | Wichtige Informationen vor dem Start | 45 |
| 8.6 | Informationen der Ein- und Ausgänge | 47 |
| 8.6.1 | Digitale Ein- und Ausgänge..... | 47 |
| 8.6.2 | Analoge Ein- und Ausgänge | 53 |
| 8.7 | Verwendung von Variablen..... | 56 |
| 8.7.1 | Allgemein gültiges (für lokale- und globale Variablen) | 56 |
| 8.7.2 | Globale Nummernvariable | 56 |
| 8.7.2.1 | Zugriff auf globale Variablen (Schreib- und Lesezugriff)..... | 56 |
| 8.7.2.2 | Referenz auf eine Nummernvariable über den Namen..... | 57 |
| 8.7.2.3 | Lokale Variablen | 58 |
| 8.7.2.4 | Kombination von lokalen- und globalen Variablen | 58 |
| 8.8 | Verwendung Konstanten | 59 |
| 8.8.1 | Allgemein gültiges..... | 59 |
| 8.8.2 | Konstanten..... | 59 |
| 8.9 | Verwendung von Bits..... | 60 |
| 8.9.1 | Allgemein | 60 |
| 8.9.2 | Zugriff auf Bits..... | 61 |

| | | |
|-----------|---|-----------|
| 8.10 | Drive Specific Commands | 62 |
| 8.10.1 | DRIVE_READ | 62 |
| 8.10.2 | DRIVE_WRITE | 63 |
| 8.10.3 | DRIVE_RESET | 64 |
| 8.10.4 | DRIVE_SAVE | 65 |
| 8.10.5 | DRIVE_CLEAR | 65 |
| 8.10.6 | DRIVE_HOME | 66 |
| 8.10.7 | DRIVE_SCAN | 68 |
| 9 | QUICK START..... | 69 |
| 9.1 | Voraussetzung..... | 69 |
| 10 | ANHANG | 71 |
| 10.1 | Bedeutung einzelner Zeichen und Kürzeln | 71 |
| 10.2 | Einzelne Kommunikations-Objekte | 72 |
| 10.2.1 | KO 042A _h : Festlegung der Funktionalität von Analogmonitor 0..... | 72 |
| 10.2.2 | KO 042B _h : Festlegung der Funktionalität von Analogmonitor 1..... | 72 |
| 10.2.3 | KO 042E _h : Darstellung einer festen Spannung an Analogmonitor 0..... | 72 |
| 10.2.4 | KO 042F _h : Darstellung einer festen Spannung an Analogmonitor 1..... | 72 |
| 10.2.5 | KO 0438 _h : Wert in digitale Ausgänge eintragen | 72 |
| 10.2.6 | KO 0439 _h : Maske für digitale Ausgänge..... | 73 |
| 10.2.7 | KO 044C _h : Zustand der digitale Eingänge..... | 74 |
| 10.3 | Referenz/Auflistung von schon angewendeten Befehlen..... | 75 |

Abbildungsverzeichnis

| | |
|---|----|
| Abbildung 1: Serielle Schnittstellen [X2MC]..... | 28 |
| Abbildung 2: Steckerbelegung RS232-Nullmodemkabel [X2MC]..... | 29 |
| Abbildung 3: Verkabelungsbeispiel für CAN-Bus..... | 33 |
| Abbildung 4: Beispiel einer CAN-Bus-Verdrahtung eines 4-Achssystems | 34 |
| Abbildung 5: RS232 Schnittstellenparameter | 44 |

Tabellenverzeichnis

| | |
|--|----|
| Tabelle 1: Technische Daten Motion Coordinator MC 2000 | 25 |
| Tabelle 2: Steckerbelegung: Serielle Schnittstellen [X2MC] | 28 |
| Tabelle 3: Steckerbelegung CAN-Bus [X1MC] | 31 |
| Tabelle 4: Prozessanzahl und Prozessnummerneinteilung | 45 |
| Tabelle 5: Technische Daten der digitale Ein- und Ausgänge [X1] | 48 |
| Tabelle 6: Bearbeitung der digitale Ein- und Ausgänge [X1] im Master | 49 |
| Tabelle 7: Bearbeitung der digitale Ein- und Ausgänge (EA88-Modul) im Master | 50 |
| Tabelle 8: Technische Daten der analogen Ein- und Ausgänge [X1] | 53 |
| Tabelle 9: Zuordnung Achsnummer/Achsen | 62 |
| Tabelle 10: Funktionszuordnung von Analogmonitor 0 | 72 |
| Tabelle 11: Funktionszuordnung/Bitzuordnung der digitalen Ausgänge | 73 |
| Tabelle 12: Funktionszuordnung/Bitzuordnung der digitalen Eingänge | 74 |
| Tabelle 13: Bewegungs- und Achsbefehle | 75 |
| Tabelle 14: Ein- und Ausgabebefehle | 75 |
| Tabelle 15: Programmschleifen und Programmstrukturen | 76 |
| Tabelle 16: Systemparameter und Systemkommandos | 76 |
| Tabelle 17: Mathematische Operationen und Kommandos | 77 |
| Tabelle 18: Konstanten | 77 |
| Tabelle 19: Achsparameter | 77 |

Beispielverzeichnis

| | |
|--|----|
| Beispiel 1: Einen digitalen Eingang auf einen digitalen Ausgang spiegeln (Master) | 51 |
| Beispiel 2: DIN0 bis DIN9 einlesen und in hexadezimaler Darstellung ausgeben (Master) | 51 |
| Beispiel 3: DIN0 bis DIN3 einlesen und in hexadezimaler Darstellung ausgeben (Slave 1) | 51 |
| Beispiel 4: Ansteuerung der digitalen Ausgängen eines EA88-Moduls (Slave 1) | 52 |
| Beispiel 5: Spannungswert am Analogeingang AIN0 einlesen und anzeigen (Master) | 55 |
| Beispiel 6: Spannungswert am Analogeingang AIN0 einlesen und anzeigen (Slave 1) | 55 |
| Beispiel 7: Wertezuweisung an eine globale Nummernvariable | 56 |
| Beispiel 8: Inhalt einer globale Nummernvariable ausgeben | 57 |
| Beispiel 9: Referenzierung (Namenszuweisung) einer globale Nummernvariablen | 57 |
| Beispiel 10: Referenzierung, Wertezuweisung und Inhaltsausgabe einer globale Variablen | 57 |
| Beispiel 11: Wertezuweisung an eine lokale Variable | 58 |
| Beispiel 12: Inhalt einer lokalen Variable ausgeben | 58 |
| Beispiel 13: Handhabung von lokalen Variablen | 58 |
| Beispiel 14: Zugriff auf globale Variablen über den Inhalt lokaler Variablen | 58 |
| Beispiel 15: Deklaration und Initialisierung einer Konstanten | 59 |
| Beispiel 16: Ausgabe des Inhaltes einer Konstanten | 59 |
| Beispiel 17: Bitzuweisungen | 61 |
| Beispiel 18: Auswertung eines Bitzustandes | 61 |
| Beispiel 19: Auslesen der Istposition von Slave 1 (DRIVE_READ) | 63 |
| Beispiel 20: Motornennstrom von Slave 2 eintragen (DRIVE_WRITE) | 64 |
| Beispiel 21: Reset von Servopositionierregler 1 (Slave 1) ausführen (DRIVE_RESET) | 64 |
| Beispiel 22: Parameterdatensatzes von Servopositionierregler 2 (Slave 2) sichern (DRIVE_SAVE) | 65 |
| Beispiel 23: Fehlermeldungen von Servopositionierregler 1 (Slave 1) löschen (DRIVE_CLEAR) | 66 |
| Beispiel 24: Referenzfahrt einer einzelnen Achse starten (DRIVE_HOME) | 67 |
| Beispiel 25: Referenzfahrt einer einzelnen Achse (mit nur einem Übergabeparameter) starten (DRIVE_HOME) | 67 |
| Beispiel 26: Referenzfahrt einer Achsgruppe (3 Achsen) starten (DRIVE_HOME) | 67 |
| Beispiel 27: Suche auf dem CAN-Bus nach angeschlossenen Slave-Teilnehmern (DRIVE_SCAN) | 68 |

1 Allgemeines

1.1 Dokumentation

Das vorliegende Handbuch beschreibt, wie das Technologiemodul Motion Coordinator MC 2000 für die Servopositionierregler der Gerätefamilie ARS 2000 in Betrieb genommen wird. Es soll den Anwender in die Lage versetzen, sich sehr schnell in der Welt der Antriebstechnik zurechtzufinden.

Es enthält Sicherheitshinweise, die beachtet werden müssen.

Um das Ziel (sehr schnelles zurechtfinden in der Antriebstechnik) erreichen zu können, enthält das Projekt auch unterschiedliche Beispiele für dieses Technologiemodul. Somit lässt sich innerhalb sehr kurzer Zeit ein Mehrachssystem steuern.

Falls es nicht anders angegeben wird, beziehen sich sämtliche Daten/Beispiele auf die Defaulteinstellungen der Servopositionierregler und des Technologiemodul Motion Coordinator MC 2000 (z.B. Zeit für einen Programmzyklus, Factor Group, ...).

In dieser Dokumentation wird selbst nicht auf einen Servopositionierregler der Gerätefamilie ARS 2000 oder der Beschreibung des Parametrierprogramms Metronix ServoCommander™ eingegangen. Des Weiteren wird auch nicht auf die benötigte Programmiersoftware „Motion Perfect“ von der Firma Trio Motion eingegangen, die zur Programmerstellung für das Technologiemodul benötigt wird.

Weitergehende Informationen finden sich in folgenden Handbüchern zur ARS 2000 Produktfamilie:

- ❖ **Produkthandbuch “Servopositionierregler ARS 2100”**: Beschreibung der technischen Daten und der Gerätefunktionalität sowie Hinweise zur Installation und Betrieb des Servopositionierregler ARS 2100.
- ❖ **Produkthandbuch “Servopositionierregler ARS 2302 - 2310”**: Beschreibung der technischen Daten und der Gerätefunktionalität sowie Hinweise zur Installation und Betrieb des Servopositionierregler ARS 2302, 2305 und 2310.
- ❖ **Produkthandbuch “Servopositionierregler ARS 2320 + 2340”**: Beschreibung der technischen Daten und der Gerätefunktionalität sowie Hinweise zur Installation und Betrieb des Servopositionierregler ARS 2320 und 2340.
- ❖ **Softwarehandbuch “Servopositionierregler ARS 2000”**: Beschreibung der Gerätefunktionalität und der Softwarefunktionen der Firmware einschließlich der RS232 Kommunikation. Beschreibung des Parametrierprogramms Metronix ServoCommander™ mit einer Anleitung für die Erstinbetriebnahme eines Servopositionierregler der Reihe ARS 2000.
- ❖ **CAN-Handbuch “Servopositionierregler ARS 2000”**: Beschreibung des implementierten CANopen Protokolls gemäß DSP402.

2 Sicherheitshinweise für elektrische Antriebe und Steuerungen

2.1 Verwendete Symbole



Information

Wichtige Informationen und Hinweise.



Vorsicht!

Die Nichtbeachtung kann hohe Sachschäden zur Folge haben.



GEFAHR!

Die Nichtbeachtung kann **Sachschäden** und **Personenschäden** zur Folge haben.



Vorsicht! Lebensgefährliche Spannung.

Der Sicherheitshinweis enthält einen Hinweis auf eine eventuell auftretende lebensgefährliche Spannung.



Die mit diesem Symbol gekennzeichneten Abschnitte stellen Beispiele dar, die das Verständnis und die Anwendung einzelner Objekte, Parameter und Programmbeispiele erleichtern.

2.2 Allgemeine Hinweise

Bei Schäden infolge von Nichtbeachtung der Warnhinweise in dieser Betriebsanleitung übernimmt die Metronix Meßgeräte und Elektronik GmbH keine Haftung.



Vor der Inbetriebnahme sind die Sicherheitshinweise für elektrische Antriebe und Steuerungen ab Seite 12 durchzulesen.

Wenn die Dokumentation in der vorliegenden Sprache nicht einwandfrei verstanden wird, bitte beim Lieferant anfragen und diesen informieren.

Der einwandfreie und sichere Betrieb des Servoantriebsreglers setzt den sachgemäßen und fachgerechten Transport, die Lagerung, die Montage, die Projektierung, unter der Beachtung der Risiken und Schutz- und Notfallmaßnahmen und die Installation sowie die sorgfältige Bedienung und die Instandhaltung voraus. Für den Umgang mit elektrischen Anlagen ist ausschließlich ausgebildetes und qualifiziertes Personal einsetzen:

AUSGEBILDETES UND QUALIFIZIERTES PERSONAL

im Sinne dieses Produkthandbuches bzw. der Warnhinweise auf dem Produkt selbst sind Personen, die mit der Projektierung, der Aufstellung, der Montage, der Inbetriebsetzung und dem Betrieb des Produktes sowie mit allen Warnungen und Vorsichtsmaßnahmen gemäß dieser Betriebsanleitung in diesem Produkthandbuch ausreichend vertraut sind und über die ihrer Tätigkeit entsprechenden Qualifikationen verfügen:

- ❖ Ausbildung und Unterweisung der Normen und Unfallverhütungsvorschriften, die in Zusammenhang der Anwendung stehen, bzw. Berechtigung, Geräte/Systeme gemäß den Standards der Sicherheitstechnik ein- und auszuschalten, zu erden und gemäß den Arbeitsanforderungen zweckmäßig zu kennzeichnen.
- ❖ Ausbildung oder Unterweisung gemäß den Standards der Sicherheitstechnik in Pflege und Gebrauch angemessener Sicherheitsausrüstung.
- ❖ Schulung in Erster Hilfe.

Die nachfolgenden Hinweise sind vor der ersten Inbetriebnahme der Anlage zur Vermeidung von Körperverletzungen und/oder Sachschäden zu lesen:



Diese Sicherheitshinweise sind jederzeit einzuhalten.



Versuchen Sie nicht, den Servopositionierregler zu installieren oder in Betrieb zu nehmen, bevor Sie nicht alle Sicherheitshinweise für elektrische Antriebe und Steuerungen in diesem Dokument sorgfältig durchgelesen haben. Diese Sicherheitsinstruktionen und alle anderen Benutzerhinweise sind vor jeder Arbeit mit dem Servopositionierregler durchzulesen.



Sollten Ihnen keine Benutzerhinweise für den Servopositionierregler zur Verfügung stehen, wenden Sie sich an Ihren zuständigen Vertriebsrepräsentanten. Verlangen Sie die unverzügliche Übersendung dieser Unterlagen an den oder die Verantwortlichen für den sicheren Betrieb des Servopositionierreglers.



Bei Verkauf, Verleih und/oder anderweitiger Weitergabe des Servopositionierreglers sind diese Sicherheitshinweise ebenfalls mitzugeben.



Ein Öffnen des Servopositionierreglers durch den Betreiber ist aus Sicherheits- und Gewährleistungsgründen nicht zulässig.



Die Voraussetzung für eine einwandfreie Funktion des Servopositionierreglers ist eine fachgerechte Projektierung!

**GEFAHR!**

Unsachgemäßer Umgang mit dem Servopositionierregler und Nichtbeachten der hier angegebenen Warnhinweise sowie unsachgemäße Eingriffe in die Sicherheitseinrichtung können zu Sachschaden, Körperverletzung, elektrischem Schlag oder im Extremfall zum Tod führen.

2.3 Gefahren durch falschen Gebrauch

**GEFAHR!**

Hohe elektrische Spannung und hoher Arbeitsstrom!

Lebensgefahr oder schwere Körperverletzung durch elektrischen Schlag!

**GEFAHR!**

Hohe elektrische Spannung durch falschen Anschluss!

Lebensgefahr oder Körperverletzung durch elektrischen Schlag!

**GEFAHR!**

Heiße Oberflächen auf Gerätegehäuse möglich!

Verletzungsgefahr! Verbrennungsgefahr!

**GEFAHR!****Gefahrbringende Bewegungen!**

Lebensgefahr, schwere Körperverletzung oder Sachschaden durch unbeabsichtigte Bewegungen der Motoren!

2.4 Sicherheitshinweise

2.4.1 Allgemeine Sicherheitshinweise



Der Servopositionierregler entspricht der Schutzklasse IP20, sowie der Verschmutzungsstufe 1. Es ist darauf zu achten, dass die Umgebung dieser Schutz- bzw. Verschmutzungsstufe entspricht.



Nur vom Hersteller zugelassene Zubehör- und Ersatzteile verwenden.



Die Servopositionierregler müssen entsprechend den EN-Normen und VDE-Vorschriften so an das Netz angeschlossen werden, dass sie mit geeigneten Freischaltmitteln (z.B. Hauptschalter, Schütz, Leistungsschalter) vom Netz getrennt werden können.



Der Servopositionierregler kann mit einem allstromsensitiven FI-Schutzschalter (RCD = Residual Current protective Device) 300mA abgesichert werden.



Zum Schalten der Steuerkontakte sollten vergoldete Kontakte oder Kontakte mit hohem Kontaktdruck verwendet werden.



Vorsorglich müssen Entstörungsmaßnahmen für Schaltanlagen getroffen werden, wie z.B. Schütze und Relais mit RC-Gliedern bzw. Dioden beschalten.



Es sind die Sicherheitsvorschriften und -bestimmungen des Landes, in dem das Gerät zur Anwendung kommt, zu beachten.



Die in der Produktdokumentation angegebenen Umgebungsbedingungen müssen eingehalten werden. Sicherheitskritische Anwendungen sind nicht zugelassen, sofern sie nicht ausdrücklich vom Hersteller freigegeben werden.



Die Hinweise für eine EMV-gerechte Installation sind aus dem jeweiligen Produkthandbuch des Servopositionierreglers zu entnehmen. Die Einhaltung der durch die nationalen Vorschriften geforderten Grenzwerte liegt in der Verantwortung der Hersteller der Anlage oder Maschine.



Die technischen Daten, die Anschluss- und Installationsbedingungen für den Servopositionierregler sind aus diesem Produkthandbuch zu entnehmen und unbedingt einzuhalten.



GEFAHR!

Es sind die Allgemeinen Errichtungs- und Sicherheitsvorschriften für das Arbeiten an Starkstromanlagen (z.B. DIN, VDE, EN, IEC oder andere nationale und internationale Vorschriften) zu beachten.

Nichtbeachtung können Tod, Körperverletzung oder erheblichen Sachschaden zur Folge haben.



Ohne Anspruch auf Vollständigkeit gelten unter anderem folgende Vorschriften:

| | |
|----------|--|
| VDE 0100 | Bestimmung für das Errichten von Starkstromanlagen bis 1000 Volt |
| EN 60204 | Elektrische Ausrüstung von Maschinen |
| EN 50178 | Ausrüstung von Starkstromanlagen mit elektronischen Betriebsmitteln |
| EN 292 | Sicherheit von Maschinen – Grundbegriffe, allg. Gestaltungsleitsätze |
| EN 1050 | Sicherheit von Maschinen – Leitsätze zur Risikobeurteilung |
| EN 1037 | Sicherheit von Maschinen – Vermeidung von unerwartetem Anlauf |
| EN 954 | Sicherheitsrelevante Teile von Steuerungen |

2.4.2 Sicherheitshinweise bei Montage und Wartung

Für die Montage und Wartung der Anlage gelten in jedem Fall die einschlägigen DIN, VDE, EN und IEC - Vorschriften, sowie alle staatlichen und örtlichen Sicherheits- und Unfallverhütungsvorschriften. Der Anlagenbauer bzw. der Betreiber hat für die Einhaltung dieser Vorschriften zu sorgen:



Die Bedienung, Wartung und/oder Instandsetzung des Servopositionierreglers darf nur durch für die Arbeit an oder mit elektrischen Geräten ausgebildetes und qualifiziertes Personal erfolgen.

Vermeidung von Unfällen, Körperverletzung und/oder Sachschaden:



Vertikale Achsen gegen Herabfallen oder Absinken nach Abschalten des Motors zusätzlich sichern, wie durch:

mechanische Verriegelung der vertikalen Achse,
externe Brems-/ Fang-/ Klemmeinrichtung oder
ausreichenden Gewichtsausgleich der Achse.



Die serienmäßig gelieferte Motor-Haltebremse oder eine externe, vom Antriebsregelgerät angesteuerte Motor-Haltebremse alleine ist nicht für den Personenschutz geeignet!



Die elektrische Ausrüstung über den Hauptschalter spannungsfrei schalten und gegen Wiedereinschalten sichern, warten bis der Zwischenkreis entladen ist bei:

Wartungsarbeiten und Instandsetzung
Reinigungsarbeiten
langen Betriebsunterbrechungen



Vor der Durchführung von Wartungsarbeiten ist sicherzustellen, dass die Stromversorgung abgeschaltet, verriegelt und der Zwischenkreis entladen ist.



Der externe oder interne Bremswiderstand führt im Betrieb und kann bis ca. 5 Minuten nach dem Abschalten des Servopositionierreglers gefährliche Zwischenkreisspannung führen, diese kann bei Berührung den Tod oder schwere Körperverletzungen hervorrufen.



Bei der Montage ist sorgfältig vorzugehen. Es ist sicherzustellen, dass sowohl bei Montage als auch während des späteren Betriebes des Antriebs keine Bohrspäne, Metallstaub oder Montageteile (Schrauben, Muttern, Leitungsabschnitte) in den Servopositionierregler fallen.



Ebenfalls ist sicherzustellen, dass die externe Spannungsversorgung des Servopositionierreglers (24 V) abgeschaltet ist.



Ein Abschalten des Zwischenkreises oder der Netzspannung muss immer vor dem Abschalten der 24 V Versorgungsspannung des Servopositionierreglers erfolgen.



Die Arbeiten im Maschinenbereich sind nur bei abgeschalteter und verriegelter Wechselstrom- bzw. Gleichstromversorgung durchzuführen. Abgeschaltete Endstufen oder abgeschaltete Reglerfreigabe sind keine geeigneten Verriegelungen. Hier kann es im Störfall zum unbeabsichtigten Verfahren des Antriebes kommen.



Die Inbetriebnahme mit leerlaufenden Motoren durchführen, um mechanische Beschädigungen, z.B. durch falsche Drehrichtung zu vermeiden.



Elektronische Geräte sind grundsätzlich nicht ausfallsicher. Der Anwender ist dafür verantwortlich, dass bei Ausfall des elektrischen Geräts seine Anlage in einen sicheren Zustand geführt wird.



Der Servopositionierregler und insbesondere der Bremswiderstand, extern oder intern, können hohe Temperaturen annehmen, die bei Berührung schwere körperliche Verbrennungen verursachen können.

2.4.3 Schutz gegen Berühren elektrischer Teile

Dieser Abschnitt betrifft nur Geräte und Antriebskomponenten mit Spannungen über 50 V. Werden Teile mit Spannungen größer 50 V berührt, können diese für Personen gefährlich werden und zu elektrischem Schlag führen. Beim Betrieb elektrischer Geräte stehen zwangsläufig bestimmte Teile dieser Geräte unter gefährlicher Spannung.



GEFAHR!

Hohe elektrische Spannung!

Lebensgefahr, Verletzungsgefahr durch elektrischen Schlag oder schwere Körperverletzung!

Für den Betrieb gelten in jedem Fall die einschlägigen DIN, VDE, EN und IEC - Vorschriften, sowie alle staatlichen und örtlichen Sicherheits- und Unfallverhütungsvorschriften. Der Anlagenbauer bzw. der Betreiber hat für die Einhaltung dieser Vorschriften zu sorgen:



Vor dem Einschalten die dafür vorgesehenen Abdeckungen und Schutzvorrichtungen für den Berührschutz an den Geräten anbringen. Für Einbaugeräte ist der Schutz gegen direktes Berühren elektrischer Teile durch ein äußeres Gehäuse, wie beispielsweise einen Schaltschrank, sicherzustellen. Die Vorschriften VBG 4 sind zu beachten!



Den Schutzleiter der elektrischen Ausrüstung und der Geräte stets fest an das Versorgungsnetz anschließen. Der Ableitstrom ist aufgrund der integrierten Netzfilter größer als 3,5 mA!



Nach der Norm EN60617 den vorgeschriebenen Mindest-Kupfer-Querschnitt für die Schutzleiterverbindung in seinem ganzen Verlauf beachten!



Vor Inbetriebnahme, auch für kurzzeitige Mess- und Prüfzwecke, stets den Schutzleiter an allen elektrischen Geräten entsprechend dem Anschlussplan anschließen oder mit Erdleiter verbinden. Auf dem Gehäuse können sonst hohe Spannungen auftreten, die elektrischen Schlag verursachen.



Elektrische Anschlussstellen der Komponenten im eingeschalteten Zustand nicht berühren.



Vor dem Zugriff zu elektrischen Teilen mit Spannungen größer 50 V das Gerät vom Netz oder von der Spannungsquelle trennen. Gegen Wiedereinschalten sichern.



Bei der Installation ist besonders in Bezug auf Isolation und Schutzmaßnahmen die Höhe der Zwischenkreisspannung zu berücksichtigen. Es muss für ordnungsgemäße Erdung, Leiterdimensionierung und entsprechenden Kurzschlusschutz gesorgt werden.



Das Gerät verfügt über eine Zwischenkreisschnellentladeschaltung gemäß EN60204 Abschnitt 6.2.4. In bestimmten Gerätekonstellationen, vor allem bei der Parallelschaltung mehrerer Servopositionierregler im Zwischenkreis oder bei einem nicht angeschlossenen Bremswiderstand, kann die Schnellentladung allerdings unwirksam sein. Die Servopositionierregler können dann nach dem Abschalten bis zu 5 Minuten unter gefährlicher Spannung stehen (Kondensatorrestladung).

2.4.4 Schutz durch Schutzkleinspannung (PELV) gegen elektrischen Schlag

Alle Anschlüsse und Klemmen mit Spannungen von 5 bis 50 V an dem Servopositionierregler sind Schutzkleinspannungen, die entsprechend folgender Normen berührungssicher ausgeführt sind:

international: IEC 60364-4-41

Europäische Länder in der EU: EN 50178/1998, Abschnitt 5.2.8.1.



GEFAHR!

Hohe elektrische Spannung durch falschen Anschluss!

Lebensgefahr, Verletzungsgefahr durch elektrischen Schlag!

An alle Anschlüsse und Klemmen mit Spannungen von 0 bis 50 V dürfen nur Geräte, elektrische Komponenten und Leitungen angeschlossen werden, die eine Schutzkleinspannung (PELV = Protective Extra Low Voltage) aufweisen.

Nur Spannungen und Stromkreise, die sichere Trennung zu gefährlichen Spannungen haben, anschließen. Sichere Trennung wird beispielsweise durch Trenntransformatoren, sichere Optokoppler oder netzfreien Batteriebetrieb erreicht.

2.4.5 Schutz vor gefährlichen Bewegungen

Gefährliche Bewegungen können durch fehlerhafte Ansteuerung von angeschlossenen Motoren verursacht werden. Die Ursachen können verschiedenster Art sein:

- ❖ unsaubere oder fehlerhafte Verdrahtung oder Verkabelung
- ❖ Fehler bei der Bedienung der Komponenten
- ❖ Fehler in den Messwert- und Signalgebern
- ❖ defekte oder nicht EMV-gerechte Komponenten
- ❖ Fehler in der Software im übergeordneten Steuerungssystem

Diese Fehler können unmittelbar nach dem Einschalten oder nach einer unbestimmten Zeitdauer im Betrieb auftreten.

Die Überwachungen in den Antriebskomponenten schließen eine Fehlfunktion in den angeschlossenen Antrieben weitestgehend aus. Im Hinblick auf den Personenschutz, insbesondere der Gefahr der Körperverletzung und/oder Sachschaden, darf auf diesen Sachverhalt nicht allein vertraut werden. Bis zum Wirksamwerden der eingebauten Überwachungen ist auf jeden Fall mit einer fehlerhaften Antriebsbewegung zu rechnen, deren Maß von der Art der Steuerung und des Betriebszustandes abhängen.



GEFAHR!

Gefahrbringende Bewegungen!

Lebensgefahr, Verletzungsgefahr, schwere Körperverletzung oder Sachschaden!

Der Personenschutz ist aus den oben genannten Gründen durch Überwachungen oder Maßnahmen, die anlagenseitig übergeordnet sind, sicherzustellen. Diese werden nach den spezifischen Gegebenheiten der Anlage einer Gefahren- und Fehleranalyse vom Anlagenbauer vorgesehen. Die für die Anlage geltenden Sicherheitsbestimmungen werden hierbei mit einbezogen. Durch Ausschalten, Umgehen oder fehlendes Aktivieren von Sicherheitseinrichtungen können willkürliche Bewegungen der Maschine oder andere Fehlfunktionen auftreten.

2.4.6 Schutz gegen Berühren heißer Teile



GEFAHR!

Heiße Oberflächen auf Gerätegehäuse möglich!

Verletzungsgefahr! Verbrennungsgefahr!



Gehäuseoberfläche in der Nähe von heißen Wärmequellen nicht berühren!
Verbrennungsgefahr!



Vor dem Zugriff Geräte nach dem Abschalten erst 10 Minuten abkühlen lassen.



Werden heiße Teile der Ausrüstung wie Gerätegehäuse, in denen sich Kühlkörper und Widerstände befinden, berührt, kann das zu Verbrennungen führen!

2.4.7 Schutz bei Handhabung und Montage

Die Handhabung und Montage bestimmter Teile und Komponenten in ungeeigneter Art und Weise kann unter ungünstigen Bedingungen zu Verletzungen führen.



GEFAHR!

Verletzungsgefahr durch unsachgemäße Handhabung!

Körperverletzung durch Quetschen, Scheren, Schneiden, Stoßen!

Hierfür gelten allgemeine Sicherheitshinweise:



Die allgemeinen Errichtungs- und Sicherheitsvorschriften zu Handhabung und Montage beachten.



Geeignete Montage- und Transporteinrichtungen verwenden.



Einklemmungen und Quetschungen durch geeignete Vorkehrungen vorbeugen.



Nur geeignetes Werkzeug verwenden. Sofern vorgeschrieben, Spezialwerkzeug benutzen.



Hebeeinrichtungen und Werkzeuge fachgerecht einsetzen.



Wenn erforderlich, geeignete Schutzausstattungen (zum Beispiel Schutzbrillen, Sicherheitsschuhe, Schutzhandschuhe) benutzen.



Nicht unter hängenden Lasten aufhalten.



Auslaufende Flüssigkeiten am Boden sofort wegen Rutschgefahr beseitigen.

3 Produktbeschreibung

3.1 Allgemeines

Das Technologiemodul Motion Coordinator MC 2000 ermöglicht es die Servopositionierreglerfamilie ARS 2000 um bis zu vier reale Servoachsen mehrachs koordiniert zu verfahren. Damit lassen sich komplexe Bewegungsabläufe wie z.B. elektronische Kurvenscheiben und Getriebe, Punkt-zu-Punkt-Positionieren und mehrere Interpolationsarten (Interpolation, Circular Interpolation, Helical Interpolation) schnell und einfach realisieren.

Das als Plug-In ausgeführte Technologiemodul Motion Coordinator MC 2000 wird direkt in dem Technologieschacht (TECH 2) des ARS 2000 Servopositionierreglers eingesteckt und steuert als Master über CANopen DSP402 (Device Profile for Drives and Motion Control) bis zu vier Servopositionierregler. Nach wie vor erfolgt die Reglerparametrierung (wie bisher gewohnt) über die komfortable Parametriersoftware „Metronix ServoCommander™“ (ab Version 2.8.0.1.2). Die funktionale Unterstützung des Technologiemodul Motion Coordinator MC 2000 wird ab der Firmware Version 3.5.0.1.7 gewährleistet.

Durch die Plug-In-Ausführung minimiert sich der externe Verdrahtungsaufwand. Zusätzlich kann ein externer Encoder direkt über den ARS 2000 angeschlossen und als weitere Achse vom MC 2000 ausgewertet werden. Die Anwendungsprogrammierung erfolgt über die vorhandene RS232 Schnittstelle durch die bewährte Trio Motion Software „Motion Perfect“, die eine Vielzahl vorgefertigter BASIC-Befehle bereithält. Ein HMI kann über die ebenfalls vorhandene RS485 Schnittstelle (MODBUS) angeschlossen werden. Für die Einbindung externer CAN I/Os steht ein zweites CAN-Interface nach DS401 (Device Profile for I/O modules) zur Verfügung.



Zusätzlich können Steuerungsaufgaben mit diesem Modul übernommen werden. Ob hierdurch sich sogar ein übergeordnetes Mastersystem einsparen lässt, ist im Einzelfall zu klären.



Das Technologiemodul Motion Coordinator MC 2000 wird von allen dreiphasigen Servopositionierreglern ARS 2300 und sämtlichen einphasigen Servopositionierreglern ARS 2100 mit integrierter Sicherheitstechnik „Sicherer Halt“ (erkennbar durch den Stecker [X3] auf der Frontseite) unterstützt.



Das Technologiemodul Motion Coordinator MC 2000 wird ausschließlich durch den Technologieschacht **TECH 2** unterstützt.

Zusätzlich zum MC2000 können im Technologieschacht TECH 1 das I/O-Erweiterungsmodul EA88 oder das Service-Speichermodul betrieben werden. Weitere Technologiemodule werden bei Nutzung des MC2000 nicht unterstützt.

Bitte wenden Sie sich bei weitergehenden Anforderungen an Ihren Vertriebspartner um eine Lösungsmöglichkeit für Ihren Anwendungsfall zu finden.

3.2 Benötigte Komponenten

An dieser Stelle werden die notwendigsten Komponenten aufgelistet, um eine Applikation/Anwendung in Kombination mit dem Technologiemodul Motion Coordinator MC 2000 zu realisieren.

Da sich mit diesem Produkt unterschiedlichste Applikationen/Anwendungen realisieren lassen, erhebt diese Auflistung keinerlei Anrecht auf Vollständigkeit. Sie soll lediglich als Leitfaden dazu dienen, keine wichtigen Komponenten zu übersehen.

3.2.1 Hardware-Komponenten

- ❖ Technologiemodul Motion Coordinator MC 2000 Bestell-Nr.: 9200-0008-00
- ❖ Schnittstellenleitung (PC → MC 2000)
für serielle RS232-Schnittstelle (von der Firma Trio Motion, Bestell-Code: P350)
- ❖ Bediengerät MC 2000
(optional, z.B. Mini-Membrane Keypad, 2x20Zeichen, 24VDC;
von der Firma Trio Motion, Bestell-Code.: P502)
- ❖ Servopositionierregler ARS 2000 (Bestellnummer abhängig vom Reglertyp, z.B. für ARS 2102)
mit entsprechendem Steckersatz
 - Power Connector Set
 - Signal Connector Set
- ❖ Bedienpult ARS 2000 Bestell-Nr.: 9200-0300-00
(optional)
- ❖ Schnittstellenkabel (PC → ARS 2000) für serielle
RS232-Schnittstelle (Nullmodem), 3-Adrig
- ❖ 24 V DC Stromversorgung
(z.B. einphasige Stromversorgung, Ausgangsspannung von 24VDC/10A)
- ❖ CANopen-Schnittstelle:

- D-Sub Steckverbinder, 9-polig Buchse mit Steckergehäuse
(z.B. von der Firma PHOENIX CONTACT: SUBCON-PLUS-CAN, Bestell-Nr.: 27 44 69 4)
 - CAN-Verbindungsleitung
(z.B. LAPP KABEL UNITRONIC BUS CAN; 2 x 2 x 0,22; Ø 7,6 mm,
mit Cu-Gesamtabschirmung)
 - 120Ω Bus-Abschlusswiderstand
(z.B. Metallschichtwiderstand mit 1% Toleranz, Baugröße 0207 der Firma BCC, Bestell-Nr.:
232215621201)
- ❖ Servomotor mit Motor- und Geberleitung
(z.B. Drehstrom-Synchron-Servomotoren, permanent erregt mit Resolver)

3.2.2 Software-Komponenten

- ❖ Metronix ServoCommander™
(Parametrierprogramm zur Reglereinstellung,
ab Version 2.8.0.1.2) bitte Vertriebspartner kontaktieren

- ❖ Motion Perfect
(Programmier- und Parametrierprogramm für das
Technologiemodul Motion Coordinator MC 2000,
ab Version 2.4.0.9) kostenloser Download von der
Trio Motion-Homepage
(<http://www.triomotion.com>)

- ❖ Bibliothek und Beispielapplikation zu "First Step MC 2000"

4 Technische Daten

Tabelle 1: Technische Daten Motion Coordinator MC 2000

| | |
|-------------------------------|---|
| Größe (L x B x H) | 92 mm x 65 mm x 19 mm |
| Temperaturbereich | 0° C bis 50° C |
| Stromverbrauch | max. 350 mA / 3,3 VDC und 150 mA / 5 VDC (intern über Servopositionierregler ARS 2000) |
| Max. Anzahl der Achsen | 8 (4x Servoantriebe, 1x Encoder, 3x virtuelle) |
| Zusätzlicher Gebereingang | Integrierter Geber / Schrittmotor Leitungstreiber Bi-direktionaler Anschluss (über Servopositionierregler ARS 2000) |
| Zykluszeit Servoantriebe | 1 ms |
| Integrierte digitale Eingänge | 6x 24 VDC (über Servopositionierregler ARS 2000) |
| Integrierte digitale Ausgänge | 3x 24 VDC (über Servopositionierregler ARS 2000) |
| Integrierte analoge Eingänge | 3x ±10 VDC über Servopositionierregler ARS 2000 (1x 16 Bit differentiell und 2x 10 Bit single ended) |
| Integrierte analoge Ausgänge | 2 x ±10 VDC, 9 Bit (über Servopositionierregler ARS 2000) |
| Serielle Anschlüsse | 1x RS232 (Programmierung) + 1x RS485 (z.B. HMI) |
| CAN Ports | 2x CAN Interfaces (1x Remote Drives 1 Mbaud und 1x Remote CAN I/Os 500 kBaud) |
| Erweiterungssteckkarten | EA88 IO-Erweiterungsmodul, Service Speichermodul (beides über Servopositionierregler ARS 2000) |
| Anwenderspeicher | 512 kBytes |
| Tabellenspeicher | 32000 Werte |
| Multi-tasking | 2 schnelle Tasks + 5 normale Tasks |
| EMV Standard | EN 61800-3 |
| Bestellnummer | 9200-0008-00 |

5 Mechanische Installation



GEFAHR!

Nichtbeachten der in Kapitel 2 Sicherheitshinweise für elektrische Antriebe und Steuerungen (Seite 12) beschriebenen Punkte können zu Sachschaden, Körperverletzung, elektrischem Schlag oder im Extremfall zum Tod führen.



GEFAHR !

Der Antriebsregler ist vor der Montage des Moduls von jeglichen stromführenden Leitungen zu trennen.

Mit einem geeigneten Kreuzschlitz-Schraubendreher wird das Frontblech über dem Technologieschacht (TECH 2) des Servopositionierreglers ARS 2000 abgeschraubt. Das Technologiemodul Motion Coordinator MC 2000 wird jetzt in den offenen Technologieschacht (TECH 2) so eingesteckt, dass die Platine in den seitlichen Führungen des Technologieschachtes läuft. Das Technologiemodul wird bis zum Anschlag eingeschoben. Abschließend wird das Technologiemodul mit der Kreuzschlitzschraube am Gehäuse des Servopositionierreglers angeschraubt. Es ist darauf zu achten, dass die Frontplatte leitenden Kontakt mit dem Gehäuse des Servopositionierreglers ARS 2000 hat (PE).



6 Verkabelung und Steckerbelegung

Das Technologiemodul Motion Coordinator MC 2000 verfügt über zwei Anschlüssen, die am Technologie-Steckmodul ausgeführt sind. Hierbei handelt es sich zum einen um die seriellen Anschlüsse (1xRS232 und 1xRS485) und zum anderen um die CAN-Anschlüsse (1x remote device mit 1MBaud und 1X remote I/O mit 500KBaud).

Die seriellen Schnittstellen sind als 8-poliger Mini DIN Steckverbinder ausgeführt.

Hierbei steht zur Programmierung des Technologiemodul Motion Coordinator MC 2000 die serielle RS232-Schnittstelle zur Verfügung.

Über die zweite serielle Schnittstelle (RS485) lassen sich Informationen von- und zur Außenwelt (z.B. HMI) austauschen.

Diese Schnittstelle wird als [X2MC] bezeichnet.

Bei den CAN-Schnittstellen handelt es sich hingegen bei der Ausführung um einen 9-poligen D-Sub Steckverbinder (Stift).

Für die Kommunikation zwischen den einzelnen Servopositionierreglern (max. 4

Servopositionierregler) wird die CAN-Schnittstelle mit einer Busgeschwindigkeit von max. 1MBaud eingesetzt.

Zusätzlich können über die zweite CAN-Schnittstelle (Busgeschwindigkeit von max. 500KBaud) dezentrale E/A's angeschlossen werden.

Diese Schnittstelle wird als [X1MC] bezeichnet.

6.1 Anschluss: Serielle Schnittstellen [X2MC]

6.1.1 Ausführung am Gerät [X2MC]

- ❖ Der serielle Anschluss ist als 8-poliger Mini DIN Steckverbinder reglerseitig als Buchse ausgeführt

6.1.2 Gegenstecker [X2MC]

- ❖ Mini DIN Steckverbinder, 8-polig Stift
- ❖ Gehäuse für 8-poligen Mini DIN Steckverbinder

6.1.3 Steckerbelegung [X2MC]

Tabelle 2: Steckerbelegung: Serielle Schnittstellen [X2MC]

| Pin Nr. | Bezeichnung | Spezifikation |
|---------|---------------------|--|
| 1 | RS485 Data in A | für optionalen RS485-Betrieb reserviert (z.B. Anbindung an ein HMI) |
| 2 | RS485 Data in B | für optionalen RS485-Betrieb reserviert (z.B. Anbindung an ein HMI) |
| 3 | TxD | Sendeleitung, RS232-Spezifikation |
| 4 | GND | Schnittstellen GND |
| 5 | RxD | Empfangsleitung, RS232-Spezifikation |
| 6 | RS485 Data out Z | für optionalen RS485-Betrieb reserviert (z.B. Anbindung an ein HMI) |
| 7 | RS485 Data out Y | für optionalen RS485-Betrieb reserviert (z.B. Anbindung an ein HMI) |
| 8 | - | Nicht belegt |

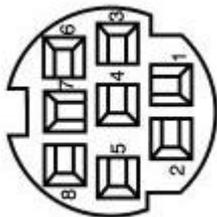


Abbildung 1: Serielle Schnittstellen [X2MC]

6.1.4 Art und Ausführung der RS232-Schnittstellenleitung

Die serielle RS232-Schnittstelle wird dazu verwendet, mittels einer „Nullmodemleitung“ eine Verbindung zwischen PC und dem Technologiemodul Motion Coordinator MC 2000 herzustellen. Hierfür wird eine 3-adrige, geschirmte Leitung (siehe Hardware-Komponenten) verwendet.



RS232 Steckerbelegung

Achtung: Die serielle RS232-Schnittstelle wurde über einen 8-poligen Mini DIN Steckverbinder realisiert. Die Pinbelegung (Pinnummern) stimmen nicht mit der RS232-Norm (Pinbelegung eines 25-poligen oder 9-poligen D-Sub Steckverbinders) überein!

6.1.5 Anschlusshinweise [X2MC]

MC 2000 Motion Control
Technologiemodul

PC

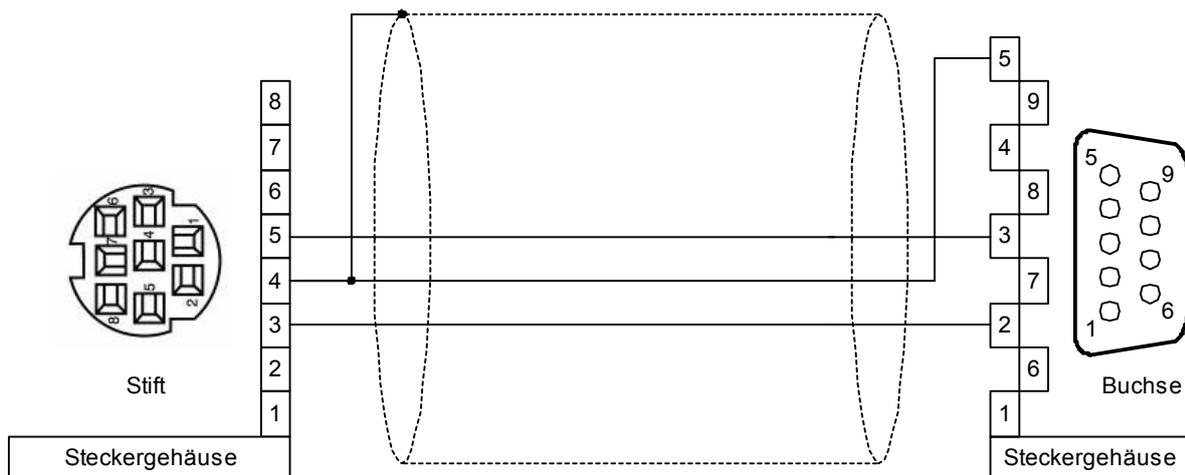


Abbildung 2: Steckerbelegung RS232-Nullmodemkabel [X2MC]

6.2 Anschluss: CAN-Bus [X1MC]

6.2.1 Ausführung am Gerät [X1MC]

- ❖ Der CAN-Bus-Anschluss ist normgemäß als 9-poliger D-Sub Steckverbinder (Stift) galvanisch getrennt ausgeführt

6.2.2 Gegenstecker [X1MC]

- ❖ D-Sub Steckverbinder, 9-polige Buchse
- ❖ Gehäuse für 9-poligen D-Sub Steckverbinder mit Verriegelungsschrauben 4/40 UNC

6.2.3 Steckerbelegung [X1MC]

Tabelle 3: Steckerbelegung CAN-Bus [X1MC]

| Pin Nr. | Bezeichnung | Wert | Spezifikation |
|---------|-------------|------|---|
| 1 | CANH B120R | **) | CAN-High Signalleitung (Schnittstelle B, intern über 120Ω-Widerstand geführt) |
| 6 | CAN GND | 0V | CAN-GND |
| 2 | CANL B | *) | CAN-Low Signalleitung (Schnittstelle B) |
| 7 | CANH B | *) | CAN-High Signalleitung (Schnittstelle B) |
| 3 | CAN GND | 0V | Siehe Pin Nr. 6 |
| 8 | CANH A | *) | CAN-High Signalleitung (Schnittstelle A) |
| 4 | CANL A | *) | CAN-Low Signalleitung (Schnittstelle A) |
| 9 | CANL A120R | ***) | CAN-Low Signalleitung (Schnittstelle A, intern über 120Ω-Widerstand geführt) |
| 5 | Schirm | PE | Anschluss für Kabelschirm |

- *) Externer Abschlusswiderstand 120Ω an den beiden Busenden erforderlich. Wir empfehlen die Verwendung von Metallschichtwiderständen mit 1% Toleranz in der Baugröße 0207, z.B. Firma BCC Art.Nr.: 232215621201.
- ***) CANB-High Signalleitung intern über einen 120Ω Widerstand geführt. Durch Anschlussverbindung dieser High-Signalleitung mit der zugehörigen Low-Signalleitung kann der Einsatz eines externen Abschlusswiderstandes eingespart werden (siehe **Abbildung 3: Verkabelungsbeispiel für CAN-Bus**).
- ***) CANA-Low Signalleitung intern über einen 120Ω Widerstand geführt. Durch Anschlussverbindung dieser Low-Signalleitung mit der zugehörigen High-Signalleitung kann der Einsatz eines externen Abschlusswiderstandes eingespart werden (Brück zwischen Pin 8 und Pin 9).

6.2.4 Art und Ausführung des Kabels [X1MC]

Die aufgeführten Kabelbezeichnungen beziehen sich auf Kabel der Firma Lapp. Sie haben sich in der Praxis bewährt und befinden sich in vielen Applikationen erfolgreich im Einsatz. Es sind aber auch vergleichbare Kabel anderer Hersteller, z.B. der Firma Lütze oder der Firma Helukabel, verwendbar.



Technische Daten CAN-Bus-Kabel: 2 Paare mit je 2 verdrehten Adern, $d \geq 0,22 \text{ mm}^2$, geschirmt, Schleifenwiderstand $< 0,2 \Omega/\text{m}$, Wellenwiderstand 100-120 Ω

- ❖ LAPP KABEL UNITRONIC BUS CAN; 2 x 2 x 0,22; \varnothing 7,6 mm, mit Cu-Gesamtabschirmung

Für hochflexible Anwendungen:

- ❖ LAPP KABEL UNITRONIC BUS-FD P CAN UL/CSA; 2 x 2 x 0,25; \varnothing 8,4 mm, mit Cu-Gesamtabschirmung

6.2.5 Anschlusshinweise [X1MC]

Der CAN-Bus bietet eine einfache und störungssichere Möglichkeit alle Komponenten einer Anlage miteinander zu vernetzen. Voraussetzung dafür ist allerdings, dass alle nachfolgenden Hinweise für die Verkabelung beachtet werden.



Bei der Verkabelung der Servopositionierregler über den CAN-Bus sollten unbedingt die nachfolgenden Informationen und Hinweise beachten werden, um ein stabiles, störungsfreies System zu erhalten. Bei einer nicht sachgemäßen Verkabelung können während des Betriebs Störungen auf dem CAN-Bus auftreten, die dazu führen, dass der Regler aus Sicherheitsgründen mit einem Fehler abschaltet.

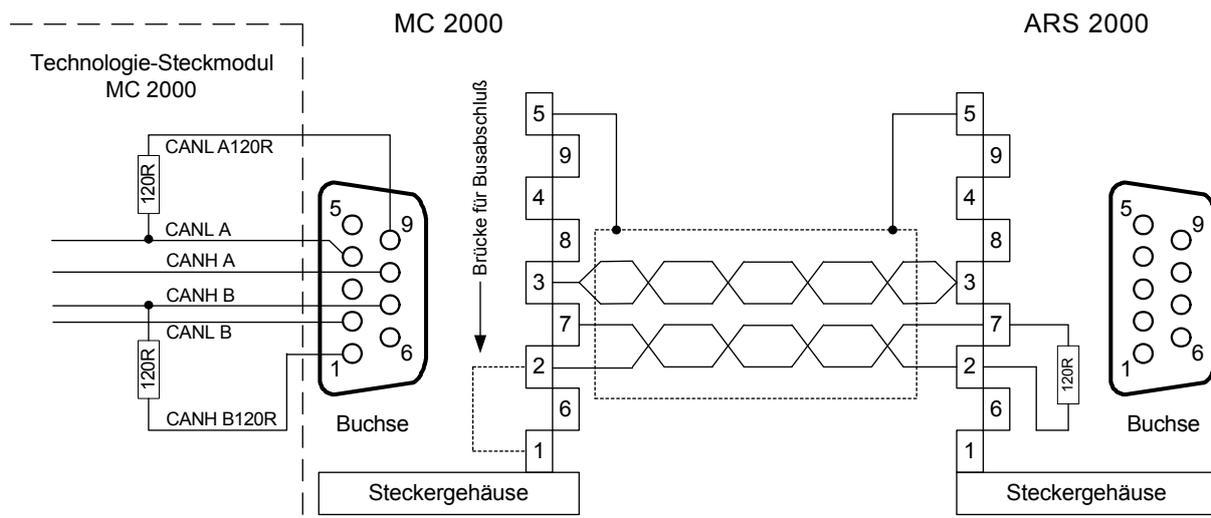


Abbildung 3: Verkabelungsbeispiel für CAN-Bus

- ❖ Die einzelnen Knoten des Netzwerkes werden grundsätzlich linienförmig miteinander verbunden, so dass das CAN-Kabel von Regler zu Regler durchgeschleift wird.
- ❖ An beiden Enden des CAN-Bus-Kabels muss jeweils genau ein Abschlusswiderstand von $120\Omega \pm 5\%$ vorhanden sein. Häufig ist in CAN-Karten oder in einer SPS bereits ein solcher Abschlusswiderstand eingebaut, der entsprechend berücksichtigt werden muss. Das Technologiemodul Motion Coordinator MC 2000 wurde in dieser Beziehung sehr flexibel ausgeführt. Somit hat der Anwender die Möglichkeit, die bereits integrierten 120Ω Widerstände oder extern angebrachte Widerstände zu verwenden.
- ❖ Für die Verkabelung muss geschirmtes Kabel mit genau zwei verdrehten Aderpaaren verwendet werden.
- ❖ Ein verdrehtes Aderpaar wird für den Anschluss von CAN-H und CAN-L verwendet.
- ❖ Die Adern des anderen Paares werden gemeinsam für CAN-GND verwendet.
- ❖ Der Schirm des Kabels wird bei allen Knoten an die CAN-Shield-Anschlüsse geführt.
- ❖ Geeignete und von Metronix empfohlene Kabel finden sie im Kapitel 6.2.4 Art und Ausführung des Kabels [X1MC]
- ❖ Von der Verwendung von Zwischensteckern bei der CAN-Bus-Verkabelung wird abgeraten. Sollte dies dennoch notwendig sein, ist zu beachten, dass metallische Steckergehäuse verwendet werden, um den Kabelschirm zu verbinden.
- ❖ Um die Störeinkopplung so gering wie möglich zu halten, sollten grundsätzlich
 - Motorkabel nicht parallel zu Signalleitungen verlegt werden.
 - Motorkabel gemäß der Spezifikation von Metronix ausgeführt sein.
 - Motorkabel ordnungsgemäß geschirmt und geerdet sein.

Für weitere Informationen zum Aufbau einer störungsfreien CAN-Bus-Verkabelung verweisen wir auf die Controller Area Network protocol specification, Version 2.0 der Robert Bosch GmbH, 1991.



CAN-Bus-Verkabelung

Um ein stabiles, störungsfreies System zu erhalten bitte hierzu die Informationen im CAN-Handbuch "Servopositionierregler ARS 2000" beachten.

Die integrierte CAN-Bus-Schnittstelle der Servopositionierreglerfamilie ARS 2000 besitzt keine galvanische Trennung.



Abbildung 4: Beispiel einer CAN-Bus-Verdrahtung eines 4-Achssystems

7 Aktivierung von MC 2000 und CANopen (Device Profile for Drives and Motion Control)

7.1 Übersicht

Zur Herstellung einer funktionsfähigen Kommunikation zwischen dem Servopositionierregler ARS 2000 und dem Technologiemodul Motion Coordinator MC 2000, sowie zur Herstellung der Kommunikation zwischen den einzelnen Servopositionierreglern über CANopen (DSP402) sind mehrere Schritte erforderlich.

Einige dieser Einstellungen sollten bzw. müssen vor der Aktivierung des Technologiemoduls ausgeführt werden. Dieses Kapitel liefert eine Übersicht über die entsprechenden Schritte.



Als Mastersystem wird die Antriebseinheit bezeichnet, welches das Technologiemodul Motion Coordinator MC 2000 beinhaltet.



Eine Grundvoraussetzung für alle Servopositionierregler ist die Übertragung (bzw. das Vorhandensein) der korrekten Firmware (siehe Kapitel 3.1 Allgemeines).



Eine CAN-Kommunikation über den CAN-Bus des Grundgerätes ist bei einem gesteckten Technologiemodul Motion Coordinator MC 2000 nicht mehr möglich.

7.2 Betriebsparameter

7.2.1 Mastersystem

Die Aktivierung des Datenaustausches zwischen dem Technologiemodul Motion Coordinator MC 2000 und dem Servopositionierer erfolgt automatisch bei einem Neustart/Reset des Servopositionierreglers. Somit entfällt z.B. eine einmalig Aktivierung über die serielle Schnittstelle des Servopositionierreglers.

Natürlich lässt sich der Datenaustausch zwischen dem Technologiemodul und dem Servopositionierer temporär deaktivieren. Dieses temporäre deaktivieren/aktivieren erfolgt über das Motion Coordinator MC2000-Fenster des Metronix ServoCommander™ (Parameter\Feldbus\MC2000\Betriebsparameter...).



Die effektive Knotennummer des Mastersystems am CAN-Bus kann nicht parametrierbar sein und ist von Haus aus auf „1“ eingestellt.

7.2.2 Slavesystem

Die Aktivierung des CAN-Interface mit dem Protokoll CANopen eines Slavesystems erfolgt einmalig über die serielle Schnittstelle des Servopositionierreglers. Das CAN-Protokoll wird über das CAN-Bus-Fenster des Metronix ServoCommander™ aktiviert

(Parameter\Feldbus\CANopen\Betriebsparameter...).

Es müssen insgesamt 3 verschiedene Parameter eingestellt werden:

❖ Basis-Knotennummer

Zur eindeutigen Identifizierung im Netzwerk muss jedem Teilnehmer eine Knotennummer zugeteilt werden, die nur einmal im Netzwerk vorkommen darf. Über diese Knotennummer wird das Gerät adressiert.

Als zusätzliche Option besteht die Möglichkeit die Knotennummer des Servopositionierreglers von der äußeren Beschaltung abhängig zu machen. Zur Basis-Knotennummer wird einmalig nach einem Neustart/Reset des Servopositionierreglers die Eingangskombination der digitalen Eingänge DIN0...DIN3 und/oder der analogen Eingänge AIN1 und AIN2 addiert. Dabei wird AIN1 mit einer Wertigkeit von 32 und AIN2 mit einer Wertigkeit von 64 hinzuaddiert, wenn der jeweilige Eingang auf $V_{ref} = 10V$ gebrückt ist.



Die effektive Knotennummer des Mastersystems am CAN-Bus kann nicht parametrieren werden und ist von Haus aus auf „1“ eingestellt.

❖ Baudrate

Dieser Parameter bestimmt die auf dem CAN-Bus verwendete Baudrate in kBaud.



Die Baudrate sollte beim Einsatz des Technologiemodul Motion Coordinator MC 2000 auf 1000 kBaud eingestellt werden!



Nach einem Neustart/Reset des Master-Servopositionierreglers wird die Baudrate auf dem Bus vom Technologiemodul Motion Coordinator MC 2000 automatisch ermittelt.

Beispiel: Alle Slave-Servopositionierregler wurden mit einer Baudrate von 500kBaud gesichert und aktiviert. Somit erkennt das Technologiemodul Motion Coordinator MC 2000 nach einem Neustart/Reset automatisch diese Busgeschwindigkeit und stellt diese auf 500kBaud ein.

❖ Optionen

Alle in einem CANopen-Netzwerk vorhandenen Geräte senden eine Einschaltmeldung (Bootup-Message) über den Bus, die die Knotennummer des Senders enthält. Empfängt der Servopositionierer eine solche Einschaltmeldung, die seiner eigenen Knotennummer entspricht, wird der Fehler 12-0 ausgelöst.

Letztlich kann das CANopen-Protokoll im Servopositionierer aktiviert werden. Beachten Sie, dass Sie die genannten Parameter nur ändern können, wenn das Protokoll deaktiviert ist.



Bitte beachten Sie, dass die Parametrierung der CANopen-Funktionalität nach einem Neustart/Reset des Servopositionierreglers nur erhalten bleibt, wenn der Parametersatz des Reglers gesichert wurde.

7.3 Diagnose

7.3.1 Steuer-/Statuswort

Unter dem Menü **Parameter/Feldbus/MC2000/Diagnose/Steuer-/Statuswort...** (beim Mastersystem) bzw. **Parameter/Feldbus/CANopen/Diagnose/Steuer-/Statuswort...** (beim Slavesystem) werden die aktuellen Werte des CANopen Steuer-/Statuswortes angezeigt.

Weitere Informationen hierzu finden Sie im CAN-Handbuch "Servopositionierregler ARS 2000".

The screenshot shows a software window titled "CANopen: Steuer-/Statuswort". It contains two main panels: "Steuerwort" (Control Word) and "Statuswort" (Status Word). Both panels have a "Wert" (Value) field showing hexadecimal values (0x001F and 0x1227 respectively) and a "Bitbelegung" (Bit Assignment) section with 16 radio buttons representing bits 0-15. The "Steuerwort" panel also includes a "Kommandos" (Commands) section with bits [7, 3, 2, 1, 0] set to "Enable operation / Switch on". The "Statuswort" panel includes a "Zustände" (States) section with bits [6, 5, 3, 2, 1, 0] set to "OPERATION_ENABLE". At the bottom, there are "OK" and "Hilfe" (Help) buttons.

Das Fenster enthält die folgenden Elemente/Funktionen:

❖ **Steuerwort anzeigen**

Über dieses Kontrollkästchen kann die Anzeige des Steuerwortes ein- bzw. ausgeschaltet werden. Die Anzeige des Steuerwortes kann nicht ausgeschaltet werden, wenn das Statuswort nicht angezeigt wird.

❖ **Statuswort anzeigen**

Über dieses Kontrollkästchen kann die Anzeige des Statuswortes ein- bzw. ausgeschaltet werden. Die Anzeige des Statuswortes kann nicht ausgeschaltet werden, wenn das Steuerwort nicht angezeigt wird.

❖ Der aktuelle Wert des Steuer- bzw. Statuswortes wird als hexadezimale Zahl dargestellt.

❖ Die einzelnen Bits werden als LEDs dargestellt. Die oberste LED stellt Bit 0 und die unterste LED Bit 15 des jeweiligen Wortes dar. Da das Steuer- und Statuswort in den verschiedenen

Betriebstarten unterschiedlich belegt sind, werden die dargestellten Texte je nach Betriebsart und Bitbedeutung angepasst.

Zusätzlich wird beim Steuerwort das zuletzt empfangene Kommando und beim Statuswort der aktuelle Zustand in Klartext dargestellt. Die Bits, die zur Bestimmung des Kommandos/Zustandes ausgewertet werden, sind in eckigen Klammern angegeben.

| | |
|--|--|
| Kommandos Bits [7, 3, 2, 1, 0]: Enable operation / Switch on | Zustände Bits [6, 5, 3, 2, 1, 0]: OPERATION_ENABLE |
|--|--|

7.3.2 PDO-Daten

Unter dem Menü **Parameter/Feldbus/MC2000/Diagnose/PDO-Daten...** (beim Mastersystem) bzw. **Parameter/Feldbus/CANopen/Diagnose/PDO-Daten ...** (beim Slavesystem) werden die aktuellen Daten der RPDOs bzw. TPDOs angezeigt.

Weitere Informationen hierzu finden Sie im CAN-Handbuch "Servopositionierregler ARS 2000".

Anhand der dargestellten Zeichen ist es z.B. möglich, Zuordnungen von Daten bzw. Vertauschungen in Datenfeldern zu erkennen. Die Reihenfolge der Bytes für 2 Byte bzw. 4 Byte Datentypen entspricht stets „Low Byte ... High Byte“. Zur Interpretation der Daten empfiehlt sich der Vergleich mit dem eingestellten PDO-Mapping.

Falls die gemappten Objekte weniger als 64 Bits belegen, werden die dargestellten PDO-Daten mit Nullen aufgefüllt, bis die 64-Bits erreicht sind.

Das Fenster enthält folgende Elemente:

❖ RPDOs

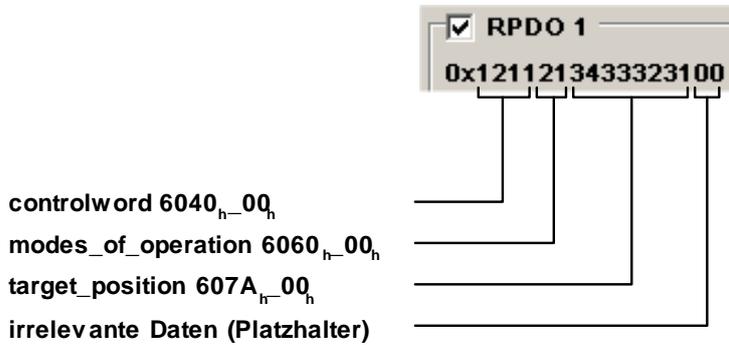
Anzeige der Daten der RPDOs (Servopositionierregler empfängt Daten). Die Anzeige kann für jedes RPDO über das entsprechende Kontrollkästchen **RPDO x** aktiviert bzw. deaktiviert werden. Die dargestellten Daten sind folgendermaßen zu interpretieren.

Eingestelltes RPDO-Mapping:

Übersicht: Receive-PDOs

| RPDO | Identifizier | Länge | Objekt 1 | Objekt 2 | Objekt 3 | Objekt 4 | Übertragungsart |
|------|--------------|---------|-----------------------|--------------------|---------------------------|----------|-----------------|
| 1. | 0x201 | 7 Bytes | controlword (6040_00) | modes_of_operation | target_position (607A_00) | | 0 SYNC |
| 2. | 0x301 | 0 Bytes | | | | | 0 SYNC |
| 3. | 0x401 | 0 Bytes | | | | | 0 SYNC |
| 4. | 0x501 | 0 Bytes | | | | | 0 SYNC |

Darstellung der empfangenen Daten mit Zuordnung zu den CANopen-Objekten:



❖ **TPDOs**

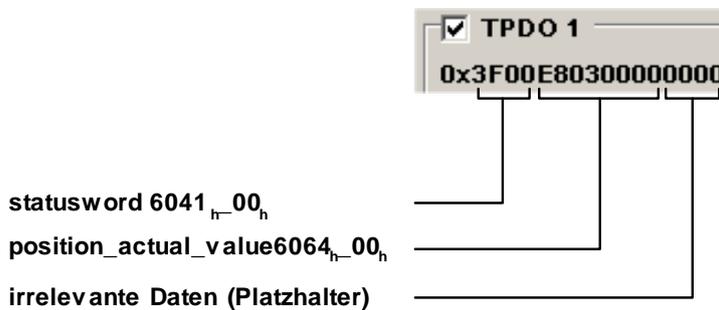
Anzeige der Daten der TPDOs (Servopositionierregler sendet Daten). Die Anzeige kann für jedes TPDO über das entsprechende Kontrollkästchen **TPDO x** aktiviert bzw. deaktiviert werden. Die dargestellten Daten sind folgendermaßen zu interpretieren.

Eingestelltes TPDO-Mapping:

Übersicht Transmit-PDOs

| TPDO | Identifizier | Länge | Objekt 1 | Objekt 2 | Objekt 3 | Objekt 4 | Übertragungsart | Wert |
|------|--------------|---------|----------------------|-----------------------|----------|----------|-----------------|--------|
| 1. | 0x181 | 6 Bytes | statusword (6041_00) | position_actual_value | | | Änderung | 0,0 ms |
| 2. | 0x281 | 0 Bytes | | | | | Änderung | 0,0 ms |
| 3. | 0x381 | 0 Bytes | | | | | Änderung | 0,0 ms |
| 4. | 0x481 | 0 Bytes | | | | | Änderung | 0,0 ms |

Darstellung der gesandten Daten mit Zuordnung zu den CANopen-Objekten:



8 Programmierung

8.1 Einleitung

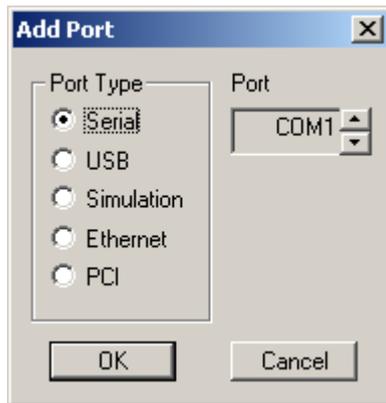
Die Programmierung der Applikation erfolgt über die vorhandene RS232 Schnittstelle durch die bewährte Trio Motion Software „Motion Perfect“, die eine Vielzahl vorgefertigter BASIC-Befehle bereithält. An dieser Stelle wurde bewusst darauf verzichtet, eine Beschreibung dieses Softwaretools mit in dieses Dokument zu integrieren. Vielmehr wird für den Umgang mit diesem Softwaretool auf die mitgelieferte HTML-Hilfedatei (MotionPerfect2.chm) verwiesen.

8.2 Kommunikationsaufbau (RS232)

Bei der Kommunikation über die serielle Schnittstelle (RS232) benötigt die Programmiersoftware (Motion Perfect) einige Informationen über die verwendete Schnittstelle. Durch Öffnen des Konfigurationsfensters für die Schnittstelle (dieser Menüpunkt befindet sich unter „**Options/Communications**“) kann eine bereits ausgewählt bzw. eine neue Schnittstelle hinzugefügt werden.

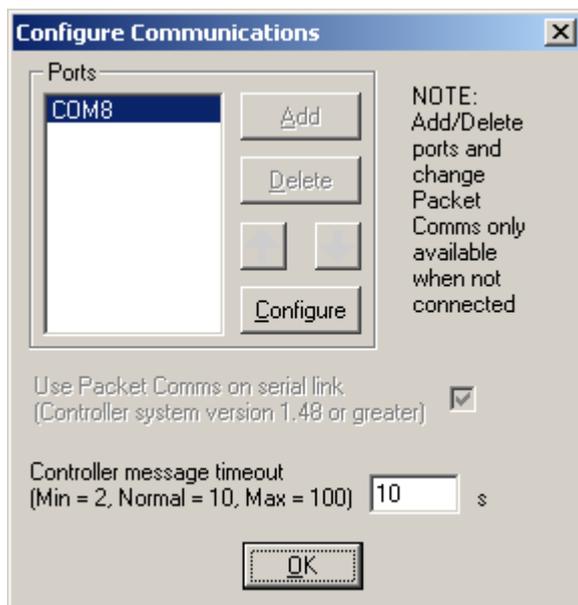


Durch Betätigen der Schaltfläche „Add“ kann eine weitere Schnittstelle hinzugefügt werden.



Eine Kommunikation zum Technologiemodul Motion Coordinator MC 2000 (für die Modulprogrammierung) wird über die serielle Schnittstelle (RS232) hergestellt.

Wurde bereits eine Verbindung zwischen der Programmiersoftware und dem Technologiemodul Motion Coordinator MC 2000 erfolgreich hergestellt (und man befindet sich noch online), so kann keine weitere Schnittstelle mehr hinzugefügt werden.



Durch Betätigen der Schaltfläche „OK“ (bei Hinzufügen einer Schnittstelle) bzw. durch Betätigen der Schaltfläche „Configure“ (im Konfigurationsfenster der Schnittstellen) gelangt man ins Menü der Schnittstellenparameter. Die zugehörigen Parameter für eine erfolgreiche Kommunikation können der **„Abbildung 5: RS232 Schnittstellenparameter“** entnommen werden.

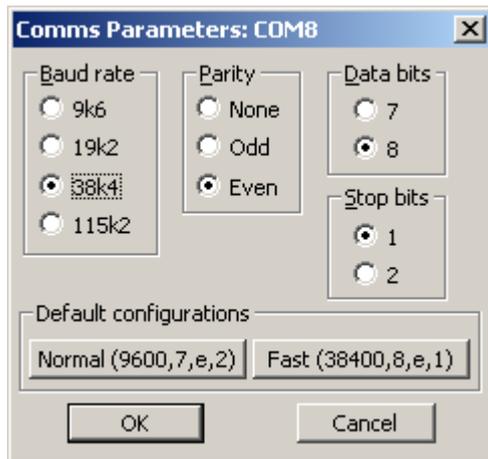


Abbildung 5: RS232 Schnittstellenparameter

8.3 LED Statusanzeige

Das Technologiemodul Motion Coordinator MC 2000 verfügt an der Frontseite über zwei LED Statusanzeigen. Anhand dieser lassen sich folgende Informationen entnehmen:

- 1) Grüne LED
Spannungsversorgung vorhanden und Software aktiv (Systemsoftware läuft)
- 2) Rote LED
Diese LED beinhaltet insgesamt 2 unterschiedliche Meldungen:
 - a) Kein Fehler vorhanden.
Hierbei repräsentiert diese LED den WDOG-Status (LED aus → WDOG=OFF, LED an → WDOG=ON)
 - b) Fehler vorhanden.
LED blinkt, bis Fehler behoben wurde.

8.4 Aufteilung der Programmausführung (Tasks)

Als klassisches Programm wird eine Aufgabe bezeichnet, die man dem Technologiemodul Motion Coordinator MC 2000 übergibt, um es auszuführen. Dieses Programm ist einfach eine Liste von Anweisungen, welche in einer für das Technologiemodul verständlichen Sprache (Trio BASIC language) erstellt wurde.

Da es sich um ein quasi multitasking fähiges Technologiemodul handelt, können somit mehrere Programme (Tasks) scheinbar gleichzeitig bearbeitet werden. Diese Tasks werden in „Prozesse“ (Vorgänge) eingeteilt, welche über Nummern vom System identifiziert werden. Dem Anwender stehen insgesamt 2 schnelle Prozesse (Tasks) und 5 normale Prozesse (Tasks) zur Verfügung. Hierbei ergibt sich folgende Zuordnung:

Tabelle 4: Prozessanzahl und Prozessnummerneinteilung

| | |
|--------------------------------|---------|
| Maximale Anzahl der Prozesse: | 7 |
| Prozessnummer (normale Tasks): | 1 bis 5 |
| Prozessnummer schnelle Tasks): | 6 und 7 |



Der Prozess 0 wird automatisch zugewiesen und für die Befehls-Eingabezeile (Terminal-Kanal 0) von Motion Perfect verwendet. Über diese Befehls-Eingabezeile können einzelne Trio BASIC-Kommandos abgesetzt (ausgeführt) werden.

Der Anwender hat somit die Möglichkeit, einzelnen Tasks (Prozesse) Nummern und somit auch Prioritäten zuzuweisen. Der Unterschied zwischen „schnellen Tasks“ und „normalen Tasks“ besteht darin, dass den schnellen Tasks ein fester Zeitschlitz zugeteilt wird. Somit wird eine Abarbeitung dieser Programmteile in immer gleichen Zeitintervallen gewährleistet. Dies kann z.B. für die Abfrage von Sicherheitseinrichtungen (Not-Aus) verwendet werden.

Allen anderen Tasks (einschließlich der Befehls-Eingabezeile) müssen sich den/die verbleibende(n) Zeitschlitz(e) gemeinsam aufteilen. Dies bedeutet, dass eine zeitlich exakte Abarbeitung der einzelnen Tasks nicht gewährleistet ist. Für die meisten Anwendungen jedoch dürfte diese Abarbeitungsgeschwindigkeit dieser normalen Tasks ausreichend sein. In solchen Tasks könne z.B. Ein- und Ausgänge, start von Positionierung oder Abläufe zur Steuerung von HMI-Sequenzen bearbeitet werden (HMI = Human Machine Interface).

8.5 Wichtige Informationen vor dem Start

Das Technologiemodul Motion Coordinator MC 2000 wurde in Zusammenarbeit mit der Firma Trio Motion entwickelt. Der Befehlsumfang dieses Technologiemoduls entspricht deshalb in etwa dem bekannten Motion Coordinator MC206X der Firma Trio Motion.



Bei dem Technologiemodul Motion Coordinator MC 2000 handelt es sich um einen speziell entwickelten Motion Coordinator für die Servopositionierregler der ARS 2000 Familie. Deshalb kann die Befehlsbeschreibung (08 Command Reference.pdf) vom Motion Coordinator MC206X als Hilfe herangezogen werden, jedoch sind nicht alle dort aufgeführten Befehle im Befehlsumfang des Technologiemoduls enthalten. Auch können dort aufgeführte Befehle evtl. nur teilweise bzw. nur unter bestimmten Bedingungen wie beschrieben ausgeführt werden. Desweiteren kann es vorkommen, dass man durch das Anwenden eines Befehles ungewollt eine Aktion ausführt, die an dieser Stelle nicht explizit beschrieben wurde.



Im Anhang befindet sich eine Auflistung/Referenz der Befehle, die bereits mit dem Technologiemodul Motion Coordinator MC 2000 zum Teil verifiziert wurden.

8.6 Informationen der Ein- und Ausgänge

8.6.1 Digitale Ein- und Ausgänge

Mit dem Technologiemodul Motion Coordinator MC 2000 haben Sie Zugriff auf die Ein- und Ausgänge des Servopositionierreglers ARS 2000. Durch den Einsatz des IO-Erweiterungs-Interface EA88 kann der Servopositionierregler um zusätzliche 8 digitale Ein- und Ausgänge erweitert werden. Bei den Slavesystemen besteht die Möglichkeit, insgesamt zwei solcher IO-Erweiterungsmodule zu integrieren. Der Maximalausbau (bei einem 4-Achs-System) würde somit 7 EA88-Module beinhalten. Bei solch einem Anwendungsfall würde das System um insgesamt 56 Ein- und Ausgänge erweitert werden.



Einen direkten Zugriff auf das IO-Erweiterungs-Interface EA88 über ein CAN-Objekt ist nur möglich, wenn es sich im Technologieschacht 1 befindet.



Das Bearbeiten von digitalen Ein- und Ausgängen eines EA88-Moduls im Technologieschacht 2 kann über eine Tunnelung erfolgen (siehe Beispiel 4). Ein Tunneln bezeichnet in der Elektronischen Datenverarbeitung (EDV) das Übertragen der Daten eines Netzwerkprotokolls (hier die Kommunikations-Objekte des RS232-Protokolls), eingebettet in ein anderes Netzwerkprotokoll (hier das CANopen-Protokoll).



Achtung: Der Anwender hat darauf zu achten, dass noch genügend Buskapazitäten für den konfigurierten PDO Datenaustausch zur Verfügung steht. Falls dies nicht mehr gegeben sein sollte, greift ein Schutzmechanismus des Technologiemodul Motion Coordinator MC 2000 ein und entzieht z.B. die Kontrolle des Servoantriebes (SERVO=OFF).

Aus nachfolgender Tabelle können die technischen Daten der digitalen Ein- und Ausgänge eines Servopositionierreglers ARS 2000 [X1] entnommen werden.

Tabelle 5: Technische Daten der digitale Ein- und Ausgänge [X1]

| Digitale Ein- und Ausgänge | Werte |
|-----------------------------------|---|
| Logikeingänge | 24V (8V...30V) aktiv high, EN 1131-2 konform |
| DIN 0 | frei verfügbar |
| DIN 1 | frei verfügbar |
| DIN 2 | frei verfügbar |
| DIN 3 | frei verfügbar |
| DIN 4 | Steuereingang Endstufenfreigabe bei High (Flankenwertung) |
| DIN 5 | Regler frei bei High, Fehler quittieren bei Low (Flankenwertung) |
| DIN 6 | Endschalteneingang 0 |
| DIN 7 | Endschalteneingang 1 |
| DIN 8 | Steuersignal Start Positionierung oder Referenzschalter für Referenzfahrt oder speichern von Positionen |
| DIN 9 | Steuersignal Start Positionierung oder Referenzschalter für Referenzfahrt oder speichern von Positionen |
| Logikausgänge | Galvanisch getrennt, 24V (8V...30V) aktiv high |
| DOUT 0 | betriebsbereit 24 V, max. 100 mA |
| DOUT 1 | frei konfigurierbar 24 V, max. 100 mA |
| DOUT 2 | frei konfigurierbar 24 V, max. 100 mA |
| DOUT 3 | frei konfigurierbar 24 V, max. 100 mA |
| DOUT 4 | Haltebremse [X6] 24 V, max. 1 A |

| Analoge Ein- und Ausgänge | Werte |
|----------------------------------|--|
| AIN 0 | Hochauflösender Analogeingang, $\pm 10V$ Eingangsbereich, 16 Bit, differentiell, $< 250\mu s$ Verzögerungszeit |
| AIN 1 | $\pm 10V$ Eingangsbereich, 10 Bit, single ended, $< 250\mu s$ Verzögerungszeit |
| AIN 2 | $\pm 10V$ Eingangsbereich, 10 Bit, single ended, $< 250\mu s$ Verzögerungszeit |
| AOUT 0 | $\pm 10V$ Ausgangsbereich, 9 Bit, $f_{Grenz} > 1kHz$ |
| AOUT 1 | $\pm 10V$ Ausgangsbereich, 9 Bit, $f_{Grenz} > 1kHz$ |

Tabelle 6: Digitale Ein- und Ausgänge [X1] im Master

| Digitale Ein- und Ausgänge | Werte | Adresse (Bit) | Beispielbefehl |
|----------------------------|---|-------------------|----------------|
| Logikeingänge | 24V (8V...30V) aktiv high, EN 1131-2 konform | | |
| DIN 0 | frei verfügbar | 0 | IN(0) |
| DIN 1 | frei verfügbar | 1 | IN(1) |
| DIN 2 | frei verfügbar | 2 | IN(2) |
| DIN 3 | frei verfügbar | 3 | IN(3) |
| DIN 4 | Steuereingang Endstufenfreigabe bei High (Flankenwertung) | 4 | IN(4) |
| DIN 5 | Regler frei bei High, Fehler quittieren bei Low (Flankenwertung) | 5 | IN(5) |
| DIN 6 | Endschaltereingang 0 | 6 | IN(6) |
| DIN 7 | Endschaltereingang 1 | 7 | IN(7) |
| DIN 8 | Steuersignal Start Positionierung oder Referenzschalter für Referenzfahrt oder speichern von Positionen | 8 | IN(8) |
| DIN 9 | Steuersignal Start Positionierung oder Referenzschalter für Referenzfahrt oder speichern von Positionen | 9 | IN(9) |
| Logikausgänge | Galvanisch getrennt, 24V (8V...30V) aktiv high | | |
| DOUT 0 | betriebsbereit | 24 V, max. 100 mA | --- |
| DOUT 1 | frei konfigurierbar | 24 V, max. 100 mA | 11 OP(11, 1) |
| DOUT 2 | frei konfigurierbar | 24 V, max. 100 mA | 12 OP(12, 1) |
| DOUT 3 | frei konfigurierbar | 24 V, max. 100 mA | 13 OP(13, 1) |
| DOUT 4 | Haltebremse [X6] | 24 V, max. 1 A | --- |

Tabelle 7: Digitale Ein- und Ausgänge (EA88-Modul) im Master

| Digitale Ein- und Ausgänge | Werte | Adresse (Bit) | Beispielbefehl | |
|----------------------------|----------------------------|---------------|----------------|-----------|
| Logikeingänge | 24V (8V...30V) aktiv high | | | |
| EA88_DIN 0 | frei verfügbar | 14 | IN(14) | |
| EA88_DIN 1 | frei verfügbar | 15 | IN(15) | |
| EA88_DIN 2 | frei verfügbar | 16 | IN(16) | |
| EA88_DIN 3 | frei verfügbar | 17 | IN(17) | |
| EA88_DIN 4 | frei verfügbar | 18 | IN(18) | |
| EA88_DIN 5 | frei verfügbar | 19 | IN(19) | |
| EA88_DIN 6 | frei verfügbar | 20 | IN(20) | |
| EA88_DIN 7 | frei verfügbar | 21 | IN(21) | |
| Logikausgänge | 24V (18V...30V) aktiv high | | | |
| EA88_DOUT 0 | frei verfügbar | 24 V, 100mA | 22 | OP(22, 1) |
| EA88_DOUT 1 | frei verfügbar | 24 V, 100mA | 23 | OP(23, 1) |
| EA88_DOUT 2 | frei verfügbar | 24 V, 100mA | 24 | OP(24, 1) |
| EA88_DOUT 3 | frei verfügbar | 24 V, 100mA | 25 | OP(25, 1) |
| EA88_DOUT 4 | frei verfügbar | 24 V, 100mA | 26 | OP(26, 1) |
| EA88_DOUT 5 | frei verfügbar | 24 V, 100mA | 27 | OP(27, 1) |
| EA88_DOUT 6 | frei verfügbar | 24 V, 100mA | 28 | OP(28, 1) |
| EA88_DOUT 7 | frei verfügbar | 24 V, 100mA | 29 | OP(29, 1) |



Zugriffe auf digitale Ein- und Ausgänge eines Slavesystems können nicht direkt (wie beim Mastersystem) über die Kommandos „IN(xx)“ und „OP(xx)“ erfolgen.

Zum Einlesen der digitalen Eingänge eines Slavesystems müssen Sie hierfür das CAN-Objekt „60FD, digital_inputs“ verwenden. Zur Ansteuerung der digitalen Ausgänge eines Slavesystems verwenden Sie bitte das CAN-Objekt „60FE, digital_outputs“.

Diese CAN-Objekte werden mit Hilfe der Drive Specific Commands bearbeitet (DRIVE_READ und DRIVE_WRITE).



Ein Zugriff auf die digitalen Eingänge DIN 10 und DIN 11 aus der Software heraus ist nicht möglich. Dies gilt für das Master- und das Slavesystem.

Die Einstellung (über den Metronix ServoCommander™) „DOUT 2 als digitalen Eingang DIN 10 nutzen“ und/oder „DOUT 3 als digitalen Eingang DIN 11 nutzen“ ist an dieser Stelle (für eine Abfrage aus dem Programm heraus) ohne Bedeutung.



Ein Zugriff auf die digitalen Eingänge DIN AIN 1 und DIN AIN 2 aus der Software heraus ist nicht möglich. Dies gilt für das Master- und das Slavesystem.

Die Einstellung (über den Metronix ServoCommander™) „AIN 1 als digitalen Eingang DIN AIN 1 nutzen“ und/oder „AIN 2 als digitalen Eingang DIN AIN 2 nutzen“ ist an dieser Stelle (für eine Abfrage aus dem Programm heraus) ohne Bedeutung.



Beispiel 1: Einen digitalen Eingang auf einen digitalen Ausgang spiegeln (Master)

Digitalen Eingang DIN8 vom Mastersystem auf digitalen Ausgang DOUT2 des Mastersystems spiegeln.

```
OP(12, IN(8)) ' Einlesen des digitalen Einganges DIN8 'IN(8)' mit
              ' Zuweisung an den digitalen Ausgang DOUT2 'OP(12)'
```



Beispiel 2: DIN0 bis DIN9 einlesen und in hexadezimaler Darstellung ausgeben (Master)

Aktuellen Zustand der digitalen Eingänge DIN0 bis DIN9 vom Mastersystem einlesen und über die serielle Schnittstelle (Terminal-Kanal 0) in hexadezimaler Darstellung ausgeben.

```
PRINT HEX(IN(0, 9)) ' Einlesen und anzeigen (in hexadezimaler Darstellung)
                   ' der digitalen Eingänge DINO bis DIN9
```



Beispiel 3: DIN0 bis DIN3 einlesen und in hexadezimaler Darstellung ausgeben (Slave 1)

Aktuellen Zustand der digitalen Eingänge DIN0 bis DIN3 vom Slavesystem 1 einlesen und über die serielle Schnittstelle (Terminal-Kanal 0) in hexadezimaler Darstellung ausgeben.

```
IF DRIVE_READ(1, $60FD, 0, 1000) = 0 THEN
    PRINT "Fehler beim Einlesen der digitalen Eingaenge von Slavesystem 1"
    PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
ELSE
    ' Benoetigte Eingaenge ausmaskieren (DINO bis DIN3 = Bit 24 bis 27) und
    ' hexadezimal ueber die serielle Schnittstelle (Terminal 0) ausgeben
    i32_output = VR(1002) AND $FOO ' Ausmaskierte Bits lokal zwischenspeichern
    IF i32_output > 255 THEN
        i32_output = i32_output / 256 ' Ausmaskierte Bits um 8 Stellen nach
        ' rechts schieben
    ELSE
        i32_output = 0
    ENDIF
    PRINT HEX(i32_output)
ENDIF
```



Beispiel 4: Ansteuerung der digitalen Ausgänge eines EA88-Moduls (Slave 1)

Ansteuerung der digitalen Ausgänge eines EA88-Moduls (Slavesystem 1) in Technologieschacht 2 über eine Tunnelung. Nur die anzusteuernenden Ausgänge (DOUT_EA88_0 bis DOUT_EA88_7) werden durch Eintrag einer Maske für die Ansteuerung freigeschaltet (Bit 14 bis 21). Die Ausgänge 0, 2, 4, 6 und 8 werden für 2 Sekunden angesteuert und danach wieder gelöscht.

```
' Adresse fuer Kommunikationsobjekt (KO) eintragen (Maske)
IF DRIVE_WRITE(1, $2200, $0, 32, $0439, 0, 1000) = 0 THEN
  PRINT "Fehler bei Eintrag der KO-Adresse (Maske)"
  PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
ELSE
  ' Maske eintragen (Bit 14 bis 21) dürfen bearbeitet werden
  IF DRIVE_WRITE(1, $2214, $0, 32, $C000, $3F, 1000) = 0 THEN
    PRINT "Fehler beim Schreiben auf ein KO (Maske eintragen)"
    PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
  ELSE
    ' KO fuer Zugriff auf EA88-Modul eintragen
    IF DRIVE_WRITE(1, $2200, $0, 32, $0438, 0, 1000) = 0 THEN
      PRINT "Fehler bei Eintrag der KO-Adresse (EA88-Modul)"
      PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
    ELSE
      ' Ausgang 0, 2, 4, 6 und 8 von EA88-Modul in Technologieschacht 2 setzen
      IF DRIVE_WRITE(1, $2214, $0, 32, $4000, $15, 1000) = 0 THEN
        PRINT "Fehler beim Schreiben auf ein KO (Ausgange bearbeiten)"
        PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
      ELSE
        PRINT "Ausgang 0, 2, 4, 6 und 8 von EA88-Modul wurden gesetzt"
        WA(2000) ' Zeitverzögerung (in ms)
        ' Alle Ausgänge vom EA88-Modul in Technologieschacht 2 löschen
        IF DRIVE_WRITE(1, $2214, $0, 32, $0, $0, 1000) = 0 THEN
          PRINT "Fehler beim Schreiben auf ein KO (Ausgange bearbeiten)"
          PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
        ELSE
          PRINT "Alle Ausgaenge vom EA88-Modul wurden geloescht"
          WA(2000) ' Zeitverzögerung (in ms)
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
```



Natürlich kann man auch für das Eintragen des KO, der Maske und den Zugriff auf das EA88-Interface eine Subroutine erstellen. Somit würde ein Aufruf dieser Subroutine erfolgen, danach könnte man immer die gewünschten Ausgänge mit einem SDO-Befehl beschreiben.

Achtung: Dies gilt natürlich nur, wenn nicht in der Zwischenzeit evtl. andere KO eingetragen wurden (z.B. Einlesen der digitalen Eingänge eines EA88-Interfaces).

8.6.2 Analoge Ein- und Ausgänge

Ebenso wie auf die digitalen Ein- und Ausgänge können Sie über das Technologiemodul Motion Coordinator MC 2000 direkt auf die analogen Ein- und Ausgänge eines Servopositionierreglers zugreifen.

Aus der nachfolgenden Tabelle (**Tabelle 8: Technische Daten der analogen Ein- und Ausgänge [X1]**) können die benötigten Informationen für das Mastersystem entnommen werden.

Tabelle 8: Technische Daten der analogen Ein- und Ausgänge [X1]

| Analoge Ein- und Ausgänge | Werte |
|---------------------------|--|
| AIN 0 | Hochauflösender Analogeingang, ±10V Eingangsbereich, 16 Bit, differentiell, < 250µs Verzögerungszeit |
| AIN 1 | ±10V Eingangsbereich, 10 Bit, single ended, < 250µs Verzögerungszeit |
| AIN 2 | ±10V Eingangsbereich, 10 Bit, single ended, < 250µs Verzögerungszeit |
| AOUT 0 | ±10V Ausgangsbereich, 9 Bit, $f_{Grenz} > 1\text{kHz}$ |
| AOUT 1 | ±10V Ausgangsbereich, 9 Bit, $f_{Grenz} > 1\text{kHz}$ |



Zugriffe auf analoge Ein- und Ausgänge eines Slavesystems können nicht direkt (wie beim Mastersystem) über die Kommandos „AIN(xx)“ und „AOUT(xx)“ erfolgen.

Zum Einlesen der analogen Eingänge eines Slavesystems müssen Sie hierfür das CAN-Objekt „2400, analog_input_voltage“ verwenden. Die Analogen Ausgänge eines Slavesystems können über eine Tunnelung von KO bedient werden (siehe Kapitel 10.2 Einzelne Kommunikations-Objekte).

Diese Objekte werden mit Hilfe der Drive Specific Commands bearbeitet (DRIVE_READ und DRIVE_WRITE).



Die eingelesenen Werte eines Mastersystems (z.B. von AIN(0)) werden in mV dargestellt und liegen somit im Bereich zwischen -10000 und +10000 (entspricht ±10V).
Achtung: Ein evtl. eingetragener Offset bei den Analogeingängen wirkt sich auf die eingelesenen Werte aus.

Beispiel: Offset von AIN(0) = 2,000 V, somit liegen nun die eingelesenen Werte im Bereich zwischen -12000 und 8000.



Die eingelesenen Werte (z.B. von AIN(0)) eines Slavesystems über CANopen (DRIVE_READ) liegen im Bereich zwischen 0 und 65535 (entspricht $\pm 10V$).

Achtung: Ein evtl. eingetragener Offset bei den Analogeingängen wirkt sich auf die eingelesenen Werte aus.

Beispiel: Offset von AIN(0) = 2,000 V, somit liegen nun die eingelesenen- und umgerechneten Werte im Bereich zwischen -12000 und 8000.

Der eigentliche Wertebereich (0 bis 65535) verändert sich dadurch nicht. Es erfolgt in diesem Falle nur eine Verschiebung des Nullpunktes (ohne Offset entspricht der Wert 0 = 0,000V; bei einem Offset von 2,000V entspricht der Wert 0 = 2,000V → Eine Eingangsspannung von 0,000V würde somit einem Wert von 63537 entsprechen → Berechnung negativer Werte: 65536-Wert+1).

Analoge Eingänge

AIN 0 | AIN 1 | AIN 2

AIN 0 - Skalierung:

Eine Eingangsspannung von + 10 V entspricht ...

| | | |
|---------------------|----------------------|----------------------|
| Drehmomentsollwert | 7,07 A | 7,07 A |
| Momentenbegrenzung | <input type="text"/> | <input type="text"/> |
| Drehzahlsollwert | 1000 U/min | 1000 U/min |
| Drehzahlgrenze | <input type="text"/> | <input type="text"/> |
| Lagesollwert | 10,00 U | 10,00 U |
| Filterzeitkonstante | 0,13 ms | 0,13 ms |

Offset:

Automatischer Offsetabgleich

Sichere Null



Beispiel 5: Spannungswert am Analogeingang AIN0 einlesen und anzeigen (Master)

Auswertung und Anzeige von Analogeingang 0 (AIN0) des Mastersystems (Ausgabe des Spannungswertes in mV mit dazugehöriger Polarität über die serielle Schnittstelle (Terminal 0)).

```
IF AIN(0) >= 0 THEN
  PRINT "Der Eingangswert ist positiv, eingelesener Wert: ";
ELSE
  PRINT "Der Eingangswert ist negativ, eingelesener Wert: ";
ENDIF
PRINT AIN(0); "mV" ' Ausgabe des eingelesenen Spannungswertes (mV)
```



Beispiel 6: Spannungswert am Analogeingang AIN0 einlesen und anzeigen (Slave 1)

Auswertung und Anzeige von Analogeingang 0 (AIN0) des Slavesystems 1 (Ausgabe des Spannungswertes in mV mit dazugehöriger Polarität über die serielle Schnittstelle (Terminal 0)).

```
' Einlesen der aktuellen Eingangsspng. von Analogeingang 0 (Slavesystem 1)
IF DRIVE_READ(1, $2400, 1, 1000) = 0 THEN
  PRINT "Fehler beim Einlesen des Analogeinganges 0 von Slavesystem 1"
  PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
ELSE
  ' Prüfen, ob eingelesener Zahlenwert sich im positiv- oder negativ Bereich
  ' befindet. Da es sich hierbei um einen 16 Bitwert handelt wird nur die
  ' Nummernvariable VR(1001) benötigt
  IF VR(1001) > 32767 THEN
    ' Ausgabe des eingelesenen Spannungswertes (mV)
    PRINT "Der Eingangswert ist negativ, eingelesener Wert: "; 65536-VR(1001)+1;
  ELSE
    ' Ausgabe des eingelesenen Spannungswertes (mV)
    PRINT "Der Eingangswert ist positiv, eingelesener Wert: "; VR(1001);
  ENDIF
  PRINT "mV"
ENDIF
```

8.7 Verwendung von Variablen

8.7.1 Allgemein gültiges (für lokale- und globale Variablen)

Die Variableninhalte sind nicht batteriegepuffert. Dies bedeutet, dass nach einem Reset des Servopositionierreglers bzw. nach Spannungswiederkehr der Dateninhalt überschrieben wird. Somit ist darauf zu achten, dass nach einem Reset bzw. nach Spannungswiederkehr die verwendeten Variablen korrekt initialisiert (beschrieben) werden, bevor eine Abfrage auf diese erfolgt.



Für globale Variablen (Nummernvariable 0 bis 1023) besteht die Möglichkeit, diese in einem nichtflüchtigen Speicher (Flash EPROM) auf dem Modul zu hinterlegen.

Die Namensunterscheidung ist auf 16 Zeichen beschränkt. Es können längere Namen angegeben (verwendet) werden, die Unterscheidung jedoch erfolgt anhand der ersten 16 Zeichen.

Beispiel: Das Modul kann nicht zwischen ‚schleifenzaehler1‘ und ‚schleifenzaehler2‘ unterscheiden.

Für die Namensvergabe sind (in Kleinschrift) alphanumerische Zeichen und Unterstrich „_“ zulässig.

8.7.2 Globale Nummernvariable

8.7.2.1 Zugriff auf globale Variablen (Schreib- und Lesezugriff)

Schreibweise:

VR(Ausdruck)

Beschreibung:

Abfrage oder Zuweisung von globalen Nummernvariablen. Diese können auch sehr einfach als ein Feld (array) oder als eine Anzahl von Feldern verwendet werden.

Insgesamt stehen dem Anwender 1024 dieser Variablen zur Verfügung (0 bis 1023).

Die Nummernvariablen können im Trio BASIC an mehreren Stellen (Programme/Tasks) verwendet werden. Somit ist ein Datenaustausch zwischen einzelnen Programmteilen sehr einfach zu realisieren. Wird jedoch kein Datenaustausch zwischen einzelnen Programmteilen benötigt, sollte man an Stelle von globalen Nummernvariablen besser lokale Variablen mit einem aussagefähigen Namen verwenden.



Beispiel 7: Wertezuweisung an eine globale Nummernvariable

```
VR(10) = 100 ' Der Nummernvariablen 10 wird der Wert 100 zugewiesen
```



Beispiel 8: Inhalt einer globale Nummernvariable ausgeben

```
PRINT VR(2) ' Inhalt der Nummernvariablen 2 über die serielle Schnittstelle
           ' (Terminal-Kanal 0) ausgeben
```

8.7.2.2 Referenz auf eine Nummernvariable über den Namen

Schreibweise:

GLOBAL "name", VR-Nummer

Beschreibung:

Deklariert den Namen ("name") als Referenz auf eine Nummernvariable (VR-Nummer). Somit kann aus einem beliebigen Programm (Task) heraus auf diese Nummernvariable über den Namen zugegriffen werden.



Bei der Verwendung von Nummernvariablen über eine Namensreferenz ist darauf zu achten, dass diese Namenszuweisung (Deklaration) vor dem ersten Namenszugriff zu erfolgen hat.



Es können maximal 128 Referenzen (globale Namenszuweisungen) realisiert werden.



Beispiel 9: Referenzierung (Namenszuweisung) einer globale Nummernvariablen

```
GLOBAL "i32_second", 1 ' Dem Namen "i32_second" wird die globale
                       ' Nummernvariable 1 zugewiesen
```



Beispiel 10: Referenzierung, Wertezuweisung und Inhaltsausgabe einer globale Variablen

```
GLOBAL "schleifenzaehler", 4 ' Dem Namen "schleifenzaehler" wird die
                              ' Nummernvariable 4 zugewiesen
schleifenzaehler = 10 ' Zuweisung des Wertes 10 an die Nummernvariable 4
                      ' (Zugriff über den zuvor Deklarierten Namen)
PRINT schleifenzaehler ' Inhalt der Nummernvariablen 4 über die serielle
                       ' Schnittstelle (Terminal-Kanal 0) ausgeben
```

8.7.2.3 Lokale Variablen

Schreibweise:

„name“ = „Variablenzuweisung“

Beschreibung:

Mit dieser Syntax wird der lokalen Variable mit dem Namen „name“ der Inhalt (Wert) „Variablenzuweisung“ zugewiesen. Sollte diese lokale Variable dem System noch nicht bekannt sein, so erfolgt eine automatische Deklaration dieser Variable.

Insgesamt stehen dem Anwender 1024 dieser Variablen pro Programm/Task zur Verfügung.



Auf lokale Variablen kann nur innerhalb eines Programms (Task) zugegriffen werden.



Beispiel 11: Wertezuweisung an eine lokale Variable

```
schleife1 = 100 ' Zuweisung des Werts 100 an die lokale Variable "schleife1".
                ' Sollte diese lokale Variable dem System noch nicht bekannt
                ' sein, so erfolgt eine automatische Deklaration dieser Variable
```



Beispiel 12: Inhalt einer lokalen Variable ausgeben

```
PRINT schleife1 ' Inhalt der lokalen Variablen "schleife1" über die serielle
                 ' Schnittstelle (Terminal-Kanal 0) ausgeben
```



Beispiel 13: Handhabung von lokalen Variablen

```
schleife1 = schleife1 + 10 ' Zum Inhalt der lokalen Variablen "schleife1"
                           ' wird der Wert 10 addiert
```

8.7.2.4 Kombination von lokalen- und globalen Variablen



Beispiel 14: Zugriff auf globale Variablen über den Inhalt lokaler Variablen

```
schleife1 = 2 ' Zuweisung des Werts 2 an die lokale Variable "schleife1"
PRINT VR(schleife1) ' Inhalt einer Nummernvariablen über die serielle
                    ' Schnittstelle (Terminal-Kanal 0) ausgeben, welche durch
                    ' den Inhalt der lokale Variable "schleife1" repräsentiert
                    ' wird (in diesem Beispiel die Nummernvariable 2)
```

8.8 Verwendung Konstanten

8.8.1 Allgemein gültiges

Das Technologiemodul Motion Coordinator MC 2000 verfügt über keinen batteriegepufferten Speicher. Dies bedeutet, dass nach einem Reset des Servopositionierreglers bzw. nach Spannungswiederkehr eine deklarierte Konstante dem System nicht mehr bekannt ist. Somit ist darauf zu achten, dass nach einem Reset bzw. nach Spannungswiederkehr bei Verwendung von Konstanten diese zuerst deklariert werden.

Für die Namensvergabe sind (in Kleinschrift) alphanumerische Zeichen und Unterstrich „_“ zulässig.

Es können bis zu 128 Konstanten verwendet werden.

8.8.2 Konstanten

Schreibweise:

CONSTANT "name", KonstanterWert

Beschreibung:

Deklariert den Namen ("name") als eine Konstante mit dem Wert (Inhalt) „KonstanterWert“. Somit kann in einem Programm an Stelle eines festen Wertes immer der Name "name" eingesetzt werden. Dies hat den Vorteil, dass bei einer Werteänderung nicht das komplette Programm durchsucht werden muss (und alle Werte abgeändert müssen), sondern dies zentral nur an einer Stelle (wo diese Konstante deklariert wurde) geschieht.



Bei der Verwendung von Konstanten ist darauf zu achten, dass diese zuerst deklariert werden, bevor ein Zugriff erfolgt.



Beispiel 15: Deklaration und Initialisierung einer Konstanten

```
CONSTANT "sqrt2", 1.4142 ' Deklaration und Initialisierung einer Konstanten mit
                          ' dem Namen "sqrt2" und dem Inhalt (Wert) 1.4142
                          ' (sqrt ' square root = Quadratwurzel)
```



Beispiel 16: Ausgabe des Inhaltes einer Konstanten

```
PRINT sqrt2 ' Inhalt der Konstanten über die serielle Schnittstelle
            ' (Terminal-Kanal 0) ausgeben
```

8.9 Verwendung von Bits

8.9.1 Allgemein

Unter einem Bit (binary digit) versteht man die Bezeichnung einer Binärziffer. Dabei ist ein Bit die kleinste darstellbare Datenmenge, die beispielsweise durch eine Binärziffer dargestellt werden kann.

Mit Hilfe von einzelnen Bits können sehr leicht Programminformationen (z.B. Maschinenzustände) von einem Programm (Task) zum anderen übermittelt werden.

Auch andere Funktionen (z.B. Taktgenerator → findet häufig Anwendung in speicherprogrammierbare Steuerungen) können hierdurch realisiert werden.

Es stehen dem Anwender insgesamt 24 Bits zur Verfügung (0 bis 23).



Die Anzahl der zur Verfügung stehenden Bits kann (z.B. durch selbst definierte Variablen → bitweise kodiert) durch den Anwender jeder Zeit erhöht werden.

Das Technologiemodul Motion Coordinator MC 2000 verfügt über keinen batteriegepufferten Speicher. Dies bedeutet, dass nach einem Reset des Servopositionierreglers bzw. nach Spannungswiederkehr der Status der Bits nicht mehr gültig ist (logisch „0“ oder logisch „1“). Somit ist darauf zu achten, dass nach einem Reset bzw. nach Spannungswiederkehr die Bits vor einem Lesezugriff zuerst beschrieben (initialisiert) werden.

8.9.2 Zugriff auf Bits

Schreibweise:

FLAG (Flagnummer [,Wert])

Beschreibung:

Das FLAG-Kommando wird verwendet, um den Status (logisch „0“ oder logisch „1“) von einzelnen Bits abzufragen oder zu beeinflussen. Dieser Befehl kann mit ein- oder zwei Parametern ausgeführt werden.

Mit einem Parameter wird der aktuelle Status des angegebenen Bits abgefragt.

Bei Verwendung von zwei Parametern erfolgt eine Zuweisung des Bits. Hierbei gibt der zweite Parameter den Status (logisch „0“ oder logisch „1“) an, mit dem das Bit beschrieben werden soll.

Flagnummer: Bit-Nummer (0 bis 23)

Wert: Status, der dem Bit zugewiesen werden soll (logisch „0“ oder logisch „1“). Dies kann mit dem Wert ‚0‘ oder ‚1‘, bzw. durch ‚OFF‘ oder ‚ON‘ erfolgen



Beispiel 17: Bitzuweisungen

```
FLAG(10, 1) ' Setzen von Bit 10 auf logisch 1
FLAG(13, OFF) ' Setzen von Bit 13 auf logisch 0
```



Beispiel 18: Auswertung eines Bitzustandes

```
IF FLAG(10) = 1 THEN ' Abfrage des Zustandes von Bit 10. Anhand dem Status von
                    ' Bit 10 wird eine Verzweigung ausgeführt
    PRINT "Zustand von Bit 10 = 1" ' Ausgabe von "Zustand von Bit 10 = 1" über die
                                ' serielle Schnittstelle (Terminal-Kanal 0),
                                ' wenn der Zustand von Bit 10 logisch 1 ist
ELSE
    PRINT "Zustand von Bit 10 = 0" ' Ausgabe von "Zustand von Bit 10 = 0" über die
                                ' serielle Schnittstelle (Terminal-Kanal 0),
                                ' wenn der Zustand von Bit 10 logisch 0 ist
ENDIF
```

8.10 Drive Specific Commands

Die „Drive Specific Commands“ sind zusätzlich integrierte Befehle für die Servoreglerfamilie ARS 2000. Mit diesen können reglerspezifische Aktivitäten durchgeführt werden (z.B. Sicherung des aktuellen Parameterdatensatzes).

8.10.1 DRIVE_READ

Schreibweise:

DRIVE_READ (<Achsnnummer>, Index, Sub-Index, Start VR)

Beschreibung:

Über diesen Befehl können Reglerparameter ausgelesen werden. Zur Anwendung dieses Befehles müssen vier Parameter mit übergeben werden, wobei der erste Parameter optional ist. Als ersten Parameter wird die gewünschte Achse (Achsnnummer) übergeben. Wird an dieser Stelle keine Achsnnummer angegeben, so bezieht sich das Kommando automatisch auf die Achse 0 (Masterachse). Die Zuordnung der Achsnummern zu den einzelnen Achsen ist wie folgt:

Tabelle 9: Zuordnung Achsnnummer/Achsen

| Achsnnummer | Achse |
|-------------|---------|
| 0 | Master |
| 1 | Slave 1 |
| 2 | Slave 2 |
| 3 | Slave 3 |

Als weitere Parameter müssen nun (wie unter CANopen bekannt) der Index und Sub-Index des gewünschten, auszulesenden Objekts angegeben werden. Als letzter Parameter wird der Ablageort für den Rückgabewert dem Befehl übergeben. Die Ablage des Rückgabewertes erfolgt bei den Nummernvariablen. Für diesen Rückgabewert werden maximal drei Nummernvariablen verwendet. Dies ist zum einen vom auszulesenden Objekt abhängig (8-, 16- oder 32Bitwert) und zum anderen, ob dieser Befehl fehlerfrei ausgeführt werden konnte. In einem Fehlerfall werden drei Werte zurückgeliefert. Wert 1 enthält hierbei die Fehlerkennung (80h), Wert 2 und 3 enthalten den Fehlercode (LSW/MSW). Bei einer korrekten Befehlsausführung enthält die zweite bzw. die zweite und die dritte Nummernvariable den ausgelesenen Parameter (abhängig 8-, 16- oder 32Bitwert). Dieser ausgelesene Wert (wie auch bei einem Fehlercode) wird in der Reihenfolge LSW/MSW zurückgegeben.

Beispiel: Start VR = 1000, somit enthält VR(1001) das LSW und VR(1002) das MSW.

Ob bei einer Anwendung dieser Befehl korrekt ausgeführt werden konnte, kann anhand einer Befehls-Rückgabeinformation überprüft werden. Bei einer erfolgreichen Ausführung wird der Wert „-1“ zurückgeliefert, hingegen der Wert „0“ in einem Fehlerfall.



Beispiel 19: Auslesen der Istposition von Slave 1 (DRIVE_READ)

Auslesen der Istposition von Slave 1 mit Auswertung des Rückgabewertes. Die Ablage der Werte erfolgt ab Nummernvariable 1000. In einem Fehlerfall wird eine Textmeldung über die serielle Schnittstelle (Terminal-Kanal 0) ausgegeben, ansonsten wird der ausgelesene Wert in die Anwendereinheit „Umdrehungen“ umgerechnet und ausgegeben.

```
' Auslesen und umrechnen der Istposition von Slave 1
IF DRIVE_READ(1, $6064, 0, 1000) = 0 THEN
  PRINT "Fehler beim Auslesen der Slaveposition"
  PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
ELSE
  ' Umrechnung der aktuellen Position auf Umdrehungen
  VR(1003) = VR(1002) + (VR(1001) / 65536)
  ' Ausgabe der Istposition über die serielle Schnittstelle (Terminal-Kanal 0)
  PRINT "Aktuelle Slaveposition: " ; VR(1003)
ENDIF
```

8.10.2 DRIVE_WRITE

Schreibweise:

DRIVE_WRITE (<Achsnnummer>, Index, Sub-Index, Länge, LSW-Daten, MSW-Daten, Start VR)

Beschreibung:

Mit dem Kommando „DRIVE_WRITE“ können Reglerparameter beschrieben werden. Für die Beschreibung müssen folgende Parameter dem Kommando mitgeteilt/übergeben werden:

- ❖ **Achsnnummer**
Dieser Parameter ist für das Mastersystem optional (keine Angabe bezieht sich auf das Mastersystem). Die Zuordnung der Achsnnummern zu den einzelnen Achsen ist der **Tabelle 9: Zuordnung Achsnnummer/Achsen** zu entnehmen.
- ❖ **Index, Sub-Index**
Über diese zwei Werte wird das zu beschreibende CAN-Objekt eindeutig identifiziert.
- ❖ **Länge**
Längenangabe des zu übermittelnden Parameters (8-, 16- oder 32Bitwert)
- ❖ **LSW- und MSW-Daten**
Zu übertragender Daten (Werte), mit denen der Servopositionierregler beschrieben werden soll.
- ❖ **Start VR**
Beginn des Ablageorts für Werterückgabe. Der Rückgabewert hat eine feste Länge von drei Worten (drei Nummernvariablen). Wird dieses Kommando fehlerfrei abgearbeitet, so werden diese drei Nummernvariablen mit dem Wert ,0' beschrieben. Wurde das ausgeführte Kommando mit einem Fehler abgearbeitet, so enthält die erste Nummernvariable (Beginn des Ablageorts) die Fehlerkennung. Darauf folgend wird der Fehlercode (LSW und MSW) eingetragen.

Ebenfalls wie das Lesekommando (DRIVE_READ) ist dieser Befehl mit einer Befehls-Rückgabeinformation ausgestattet. Bei einer korrekt Ausführung dieses Befehls wird der Wert „-1“ zurückgeliefert, hingegen der Wert „0“ in einem Fehlerfall.



Beispiel 20: Motornennstrom von Slave 2 eintragen (DRIVE_WRITE)

In diesem Beispiel wird der Motornennstrom von Servopositionierer 2 (Slave 2) über eine lokale Variable auf 0,6A eingetragen (Einheit in mA).

```
' Slave 2: Motornennstrom von 0,6A über lokale Variable eintragen (Einheit: mA)
i32_tmp_motor_i = 600 ' Motornennstrom in mA
IF DRIVE_WRITE(2, $6075, $0, 32, i32_tmp_motor_i, 0, 1000) = 0 THEN
  PRINT "Fehler beim Schreiben des Motornennstromes (Slave 2)"
  PRINT "Fehlernummer: " ; HEX(VR(1002)) ; " " ; HEX(VR(1001))
ELSE
  PRINT "Motornennstrom (Slave 2) eingetragen"
ENDIF
```

8.10.3 DRIVE_RESET

Schreibweise:

DRIVE_RESET (<Achsnnummer>)

Beschreibung:

Mit diesem Befehl kann ein Reset eines Servopositionierreglers ausgelöst werden. Um welchen Servopositionierer es sich handelt wird anhand der Achsnnummer mitgeteilt. Wie bei den bisherigen Drive Specific Commands erhält man auch hier dieselbe Befehls-Rückgabeinformation für eine Befehlsausführung (Befehlsausführung in Ordnung = „-1“, Befehlsausführung fehlerhaft = „0“).



Beispiel 21: Reset von Servopositionierer 1 (Slave 1) ausführen (DRIVE_RESET)

Durchführung eines Resets von Servopositionierer der Achse 1 (Slave 1)

```
' Reset von Slave 1
IF DRIVE_RESET(1) = 0 THEN
  PRINT "Fehler Reglerreset (Slave 1)"
ELSE
  PRINT "Reglerreset (Slave 1) ausgeführt"
ENDIF
```



Achtung: Bei Durchführung eines Resets des Servopositionierreglers der Masterachse geht eine evtl. vorhandene Verbindung (Kommunikation) zum Technologiemodul Motion Coordinator MC 2000 verloren!

8.10.4 DRIVE_SAVE

Schreibweise:

DRIVE_SAVE (<Achsnnummer>)

Beschreibung:

Der aktuelle Parametersatz des Servopositionierreglers ist im RAM-Speicher (RAM = Random Access Memory) vorhanden. Das RAM verliert seinen Speicherinhalt, sobald die Versorgungsspannung abgeschaltet bzw. ein Reset des Servopositionierreglers ausgeführt wird. Um den Parametersatz dauerhaft zu sichern, kann er mittels dieses Befehls in einen nicht flüchtigen Speicher (Flash-Memory) kopiert werden. Das Flash-Memory verliert seinen Speicherinhalt auch dann nicht, wenn die Spannung abgeschaltet wird. Bei jedem Reset des Servopositionierreglers wird der Inhalt des Flash-Memory in das RAM kopiert.

Bei einer Befehlsausführung wird als Übergabeparameter die Achsnnummer erwartet (siehe hierzu **Tabelle 9: Zuordnung Achsnnummer/Achsen**). Wird dieser Übergabeparameter nicht verwendet, so bezieht sich dieser Befehl auf die Achse 0 (Master). Auch dieses Kommando ist mit einer Befehls-Rückgabeinformation versehen (Befehlsausführung in Ordnung = „-1“, Befehlsausführung fehlerhaft = „0“).



Beispiel 22: Parameterdatensatzes von Servopositionierregler 2 (Slave 2) sichern (DRIVE_SAVE)

```
' Sichern des aktuellen Parameterdatensatzes (Slave 2)
IF DRIVE_SAVE(2) = 0 THEN
  PRINT "Fehler beim Sichern des Parameterdatensatzes (Slave 2)"
ELSE
  PRINT "Parameterdatensatz (Slave 2) gesichert"
ENDIF
```

8.10.5 DRIVE_CLEAR

Schreibweise:

DRIVE_CLEAR (<Achsnnummer>)

Beschreibung:

Durch Ausführung dieses Kommandos werden anstehende Fehlermeldungen im Servopositionierregler gelöscht (wenn möglich). Das Löschen von anstehenden Fehlermeldungen kann nur erfolgreich durchgeführt werden, wenn zuvor die Fehlerursachen beseitigt worden sind. An welchem Servopositionierreglers die Fehlerquittierung durchgeführt werden soll, wird dem Kommando über den Parameter <Achsnnummer> mitgeteilt (siehe hierzu **Tabelle 9: Zuordnung Achsnnummer/Achsen**). Dieser Befehl enthält keine Befehls-Rückgabeinformation.



Beispiel 23: Fehlermeldungen von Servopositionierregler 1 (Slave 1) löschen (DRIVE_CLEAR)

```
DRIVE_CLEAR(1) ' Löschen anstehender Fehlermeldungen von
                ' Servopositionierregler 1 (Slave 1)
```

8.10.6 DRIVE_HOME

Schreibweise:

DRIVE_HOME (<Achsnnummer>, Methode, <Achsnnummer>, Methode, <Achsnnummer>, Methode, <Achsnnummer>, Methode)

Beschreibung:

Jede Positioniersteuerung benötigt beim Betriebsbeginn einen wiederholbaren definierten Nullpunkt, der durch eine Referenzfahrt ermittelt wird. Diese Referenzfahrt kann der Servopositionierregler eigenständig ausführen. Bei diesem Prozess wird zwischen dem Ziel und dem Bezugspunkt unterschieden. Ziel können z.B. der Endschalter, ein Referenzschalter, der Nullimpuls oder der Endanschlag sein. Als Bezugspunkt kann das Ziel selbst oder z.B. der Nullimpuls ausgewählt werden.



Nach Einschalten der 24 VDC-Spannungsversorgung oder nach einem Reset des Servopositionierreglers wird die aktuelle Istposition als „Nullpunkt“ verwendet. Bei Winkelgebern, die über eine Umdrehung eine absolute Lageinformation liefern (z.B. Resolver), wird der Nullpunkt auf die Nullposition dieses Winkelgebers gelegt. Damit existiert zwar eine Referenzposition, der Antrieb gilt aber nicht als „referenziert“. Die Referenzposition bleibt unabhängig von Regler- oder Endstufenfreigaben und von der Betriebsart erhalten. Sie geht nur mit Unterbrechung der Spannungsversorgung, einem Reset oder einem Winkelgeberfehler verloren.

Mittels „DRIVE_HOME“ kann eine Referenzfahrt ausgelöst (gestartet) werden. Dieses Kommando wurde so flexibel gestaltet, dass der Anwender eine einzelne Referenzfahrt oder ein Referenzfahrt von verschiedenen Achsgruppen (max. alle vier Achsen) gleichzeitig starten kann. Die Übergabeparameter werden in folgender Reihenfolge erwartet: Als ersten Übergabeparameter wird die zu referenzierende Achse angegeben (siehe hierzu **Tabelle 9: Zuordnung Achsnnummer/Achsen**). Als zweiten Übergabeparameter teilt man der Achse die Referenzfahrtmethode mit. Aufgrund des so flexibel gestalteten Kommandos kann somit diese Sequenz der Übergabeparameter (Achsnnummer und Referenzfahrtmethode) insgesamt vier mal erfolgen.

Spezialfall: Wird nur ein einzelner Übergabeparameter mit angegeben, so bezieht sich dieser Parameter auf die Referenzfahrtmethode der angewählten Basisachse.

Das Kommando „DRIVE_HOME“ verfügt ebenfalls über eine Befehls-Rückgabeinformation (Befehlsausführung in Ordnung = „-1“, Befehlsausführung fehlerhaft = „0“).



Wenn eine Referenzfahrt mit einer nicht vorhandenen Referenzfahrtmethode (z.B. Referenzfahrtmethode 4) gestartet wird, führt der ausgewählte Antrieb die im Servopositionierregler hinterlegte Referenzfahrtmethode durch.

Beispiel: Das Mastersystem (Antrieb 0) soll eine Referenzfahrt nach der Referenzfahrtmethode 4 (diese Methode gibt es nicht) durchführen. Die aktuelle Einstellung der Referenzfahrtmethode im Servopositionierregler ist die Methode 17 (Referenzfahrt auf den negativen Endschalter). Nach Absetzen des Kommandos wird die Referenzfahrt des Masters gestartet. Jedoch erfolgt keine Fehlermeldung (wie evtl. erwartet), sondern das Mastersystem startet eine Referenzfahrt mit der Referenzfahrtmethode 17. Falls die Referenzfahrt erfolgreich durchgeführt werden konnte, erhält der Anwender an dieser Stelle den Rückgabewert „-1“.



Beispiel 24: Referenzfahrt einer einzelnen Achse starten (DRIVE_HOME)

Start einer Referenzfahrt der Achse 0 (Master) mit der Referenzfahrt-Methode 23 (Referenzfahrt auf den Referenzschalter in positiver Richtung).

```
IF DRIVE_HOME(0, 23) = -1 THEN
    PRINT "Referenzfahrt (Master) erfolgreich ausgeführt"
ELSE
    PRINT "Fehler bei Referenzfahrt (Master)"
ENDIF
```



Beispiel 25: Referenzfahrt einer einzelnen Achse (mit nur einem Übergabeparameter) starten (DRIVE_HOME)

Start einer Referenzfahrt der Achse 1 (Slave 1) mit der Referenzfahrt-Methode 23 (Referenzfahrt auf den Referenzschalter in positiver Richtung).

Sonderfall: Ausführung dieses Kommandos mit nur einem Übergabeparameter.

```
BASE(1) ' Auswahl der Basisachse (Slave 1) fuer nachfolgende Befehle
IF DRIVE_HOME(23) = -1 THEN
    PRINT "Referenzfahrt der Achsgruppe erfolgreich ausgeführt"
ELSE
    PRINT "Fehler bei Referenzfahrt"
ENDIF
```



Beispiel 26: Referenzfahrt einer Achsgruppe (3 Achsen) starten (DRIVE_HOME)

Referenzfahrt einer Achsgruppe (3 Achsen) starten. Hierbei wird die Achse 0 (Master) mit der Referenzfahrt-Methode 23 (Referenzfahrt auf den Referenzschalter in positiver Richtung), die Achse 1 (Slave 1) mit der Referenzfahrt-Methode 17 (Referenzfahrt auf den negativen Endschalter) und die Achse 2 (Slave 2) mit der Referenzfahrt-Methode 34 (Referenzfahrt auf die aktuelle Position) gestartet.

```

IF DRIVE_HOME(0, 23, 1, 17, 2, 34) = -1 THEN
  PRINT "Referenzfahrt der Achsgruppe erfolgreich ausgefuehrt"
ELSE
  PRINT "Fehler bei Referenzfahrt"
ENDIF

```

8.10.7 DRIVE_SCAN

Schreibweise:

DRIVE_SCAN (<Achsnnummer>)

Beschreibung:

Dieser Befehl dient zum Aufbau der CAN-Kommunikation (Neuinitialisierung des CAN-Controllers). Falls es zum einem Kommunikationsabbruch zwischen dem Mastersystem (Technologiemodul Motion Coordinator MC 2000) und den Slavesystemen gekommen sein sollte, kann mit Hilfe dieses Kommandos der Bus nach Slave-Teilnehmern neu abgesucht werden (re-scan). Dies gilt für alle am CAN-Bus angeschlossenen Slave-Teilnehmer (das Technologiemodul Motion Coordinator MC 2000 führt keinen re-scan des eigenen DPR durch → Mastersystem).

Der Befehl „DRIVE_SCAN“ entspricht dem unter CANopen bekannten NMT-Service 82h (Reset Communication).

Dieser Befehl ist mit keiner Befehls-Rückgabeinformation ausgestattet.



NMT-Telegramme dürfen nicht in einem Burst (unmittelbar hintereinander) gesendet werden!

Zwischen zwei aufeinanderfolgenden NMT-Nachrichten auf dem Bus (auch für verschiedene Knoten!) muss mindestens die doppelte Lagereglerzykluszeit liegen, damit der Regler die NMT-Nachrichten korrekt verarbeitet.



Der NMT Befehl „Reset Communication“ wird gegebenenfalls so lange verzögert, bis ein laufender Speichervorgang abgeschlossen ist, da ansonsten der Speichervorgang unvollständig bleiben würde (defekter Parametersatz).

Die Verzögerung kann im Bereich einiger Sekunden liegen.



Beispiel 27: Suche auf dem CAN-Bus nach angeschlossenen Slave-Teilnehmern (DRIVE_SCAN)

```
DRIVE_SCAN
```

9 Quick Start

Anhand dieser Kurzbeschreibung werden Sie (unter gewissen Bedingungen) in die Lage versetzt, innerhalb weniger Minuten (ca. 30 Minuten) eine interpolierende Bewegung von zwei Achsen durchzuführen (Kontur der Ziffer „0“ abfahren).



Als Mastersystem wird die Antriebseinheit bezeichnet, welches das Technologiemodul Motion Coordinator MC 2000 beinhaltet.

9.1 Voraussetzung

- ❖ Der Servopositionierregler der Reihe ARS 2000 ist Ihnen nicht unbekannt
- ❖ Der Umgang mit dem Metronix ServoCommander™ ist Ihnen vertraut
- ❖ Alle benötigten Hardware-Komponenten sind vorhanden (siehe Kapitel 3.2.1 Hardware-Komponenten). In diesem Falle zwei Servopositionierregler der Gerätefamilie ARS 2000
- ❖ Sie sind im Besitz der benötigten Software-Komponenten (siehe Kapitel 3.2.2 Software-Komponenten)
- ❖ Alle benötigten Hardware-Komponenten wurden fachgerecht und unter Beachtung der jeweils dazugehörigen Dokumentation aufgebaut
- ❖ Die benötigten Software-Komponenten wurden installiert (Metronix ServoCommander™ und Motion Perfect)
- ❖ Verbindung (Kommunikation) zwischen Rechner (Metronix ServoCommander™) und ARS 2000 (Master-System) herstellen
- ❖ Verbindung (Kommunikation) zwischen Rechner (Motion Perfect) und Technologiemodul Motion Coordinator MC 2000 herstellen (siehe hierzu die Hinweise in Kapitel 8.2 Kommunikationsaufbau (RS232))
- ❖ Metronix Beispielprogramm „QuickStartWithTwoAxes_Zero“ aus dem Paket von „First Step MC 2000“ öffnen (Motion Perfect)
- ❖ Die Antriebe sind frei drehbar

**GEFAHR!**

Antriebe müssen mechanisch fixiert und frei drehbar sein. Eine geeignete Schutzvorrichtung gegen Berührung muss ebenfalls vorhanden sein. Der Aufbau muss so ausgeführt werden, dass es selbst bei unbeabsichtigten Bewegungen der Antriebe zu keinen **Sachschäden** und **Personenschäden** kommen kann!

Alle Antriebe erhalten nach einem Neustart (Spannungswiederkehr) eine Endstufen- und Reglerfreigabe (DIN4 und DIN5 auf +24 V DC legen, z.B. mittels Bedienpult ARS 2000)



Ab hier beginnt die Zeit zum Zählen!

Alle nun folgenden Angaben beziehen sich auf das Mastersystem.

- ❖ Übertragen der Firmware an den Servopositionierregler (Version 3.5.0.1.7 oder höher)
- ❖ Übertragung der Parameterdatei (FirstStep_MasterAxis0_V1_0.DCO)
- ❖ Durchführung der Erstinbetriebnahme (hierzu den evtl. aktivierten Datenaustausch zwischen Technologiemodul und dem Servopositionierregler temporär deaktivieren)
- ❖ Sichern der Daten im Servopositionierregler
- ❖ Metronix Beispielprogramm „QuickStartWithTwoAxes_Zero“ aus den Paket von “First Step MC 2000“ zum Technologiemodul Motion Coordinator MC 2000 übertragen
- ❖ Sicherung des Programms auf dem Technologiemodul

Alle nun folgenden Angaben beziehen sich auf das Slavesystem.

- ❖ Übertragen der Firmware an den Servopositionierregler (Version 3.5.0.1.7 oder höher)
- ❖ Übertragung der Parameterdatei (FirstStep_SlaveAxis1_V1_0.DCO)
- ❖ Durchführung der Erstinbetriebnahme (hierzu ein evtl. aktiviertes CAN-Interface temporär deaktivieren)
- ❖ Sichern der Daten im Servopositionierregler (Achtung: Falls die CAN-Kommunikation temporär deaktiviert wurde, diese bitte vor dem Sichern wieder aktivieren).

Nun bitte einen gleichzeitigen Neustart der Servopositionierregler durchführen. **Fertig!**

10 Anhang

10.1 Bedeutung einzelner Zeichen und Kürzeln

| | |
|-----------|--|
| \$ bzw. h | Wert in hexadezimaler Darstellung |
| DPR | Dual Ported Ram. Ein Dual-Port-RAM ist ein RAM-Speicher, bei dem von zwei Seiten gleichzeitig Lesezugriffe möglich sind. |
| HMI | Human Machine Interface |
| KO | Kommunikationsobjekt |
| LSW | Least Significant Word |
| MSW | Most Significant Word |
| NMT | Network Management. Alle CANopen-Geräte können über das Netzwerkmanagement angesteuert werden. |
| PDO | Process-Data-Objects |
| RAM | Random Access Memory |
| RPDO | Receive-PDO |
| TPDO | Transmit-PDO |

10.2 Einzelne Kommunikations-Objekte

10.2.1 KO 042A_n: Festlegung der Funktionalität von Analogmonitor 0

Die Funktionalität von Analogmonitor 0 wird anhand eines Dezimalwertes (32 Bit) festgelegt. Hierbei stehen folgende Auswahlmöglichkeiten zur Verfügung:

Tabelle 10: Funktionszuordnung von Analogmonitor 0

| Dezimalwert | Funktionalität |
|-------------|-------------------------------------|
| 1 | Drehzahlsollwert |
| 2 | Drehzahlistwert |
| 3 | Wirkstromsollwert |
| 4 | Wirkstromistwert |
| 5 | Blindstromsollwert |
| 6 | Blindstromistwert |
| 7 | Phasenstrom U |
| 8 | Geberwinkel |
| 9 | Fester Spannungswert |
| 10 | Frei wählbares Kommunikationsobjekt |
| 11 | Lagesollwert |
| 12 | Lageistwert |
| 13 | Zwischenkreisspannung |

10.2.2 KO 042B_n: Festlegung der Funktionalität von Analogmonitor 1

Die Funktionalität von Analogmonitor 1 wird ebenfalls anhand eines Dezimalwertes (32 Bit) festgelegt. Hierbei stehen die selben Auswahlmöglichkeiten wie bei Analogmonitor 1 zur Verfügung (siehe **Tabelle 10: Funktionszuordnung von Analogmonitor 0**).

10.2.3 KO 042E_n: Darstellung einer festen Spannung an Analogmonitor 0

Dieses KO wird verwendet, um einen festen Spannungswert über den Analogmonitor 0 auszugeben. Die Basiseinheit für dieses KO ist die Spannung, welche im Servopositionierregler als 32 Bitwert interpretiert wird. Somit kann eine max. Auflösung von 1/216V realisiert werden. Der hieraus resultierende Wertebereich umfasst somit $\pm 215V$.

10.2.4 KO 042F_n: Darstellung einer festen Spannung an Analogmonitor 1

Siehe hierzu „KO 042E_n: Darstellung einer festen Spannung an Analogmonitor 0“. Dieses KO bezieht sich jedoch auf den Analogmonitor 1.

10.2.5 KO 0438_n: Wert in digitale Ausgänge eintragen

Mit diesem KO kann ein Wert (Bitfeld von 32 Bit) zur Ansteuerung der digitalen Ausgänge eingetragen werden. Diese Ansteuerung umfasst nicht nur die vorhandenen digitalen Ein- und Ausgänge eines

Servopositionierreglers, sonder erstreckt sich auch noch auf die Ansteuerung der digitalen Ein- und Ausgänge eines EA88-Interfaces sowie der LEDs für CAN und ARS Ready.



Das direkte Schreiben eines Wertes (32 Bit) auf dieses KO bewirkt eine Änderung der Funktionalität aller betroffenen Bits (Bitfeld von 32 Bit). Um ein evtl. ungewolltes Überschreiben einzelner Bits (und somit einzelner Ausgänge) zu verhindern, kann das KO 0439_h (Eintrag einer Maske) verwendet werden.

Tabelle 11: Funktionszuordnung/Bitzuordnung der digitalen Ausgänge

| Adresse (Bit) | Funktionalität |
|---------------|----------------|
| 0 | DOUT0 |
| 1 | DOUT1 |
| 2 | DOUT2 |
| 3 | DOUT3 |
| 4 | EA88_DOUT 0_0 |
| 5 | EA88_DOUT 0_1 |
| 6 | EA88_DOUT 0_2 |
| 7 | EA88_DOUT 0_3 |
| 8 | EA88_DOUT 0_4 |
| 9 | EA88_DOUT 0_5 |
| 10 | EA88_DOUT 0_6 |
| 11 | EA88_DOUT 0_7 |
| 12 | CAN LED |
| 13 | ARS Ready LED |
| 14 | EA88_DOUT 1_0 |
| 15 | EA88_DOUT 1_1 |
| 16 | EA88_DOUT 1_2 |
| 17 | EA88_DOUT 1_3 |
| 18 | EA88_DOUT 1_4 |
| 19 | EA88_DOUT 1_5 |
| 20 | EA88_DOUT 1_6 |
| 21 | EA88_DOUT 1_7 |

10.2.6 KO 0439_h: Maske für digitale Ausgänge

Mit Hilfe des KO 0439h kann eine Bitmaske erstellt werden, mit deren Hilfe man ein unbeabsichtigtes überschreiben (Ansteuern) der digitalen Ausgänge verhindern kann. Somit können nur digitale Ausgänge beschrieben (angesteuert) werden, bei denen das Bit (innerhalb dieser Bitmaske) auf logisch „1“ gesetzt wurde.



Beim zyklischen Auffrischen der digitalen Ausgänge wird diese Maske nicht verwendet.

10.2.7 KO 044C_h: Zustand der digitale Eingänge

Anhand dieses KO können die Zustände der digitalen Eingänge eingelesen werden. Diese werden ebenfalls (wie die digitalen Ausgänge) durch einen 32 Bitwert (Bitfeld) repräsentiert. Anhand der unten dargestellten Tabelle kann man die Zuordnung der Eingänge zu den einzelnen Bits entnehmen.



Bei den Eingängen DIN10, DIN11, DIN_AIN1, DIN_AIN2 und den Eingängen des EA88-Interfaces handelt es sich um optionale Eingänge.

Tabelle 12: Funktionszuordnung/Bitzuordnung der digitalen Eingänge

| Adresse (Bit) | Funktionalität |
|---------------|----------------|
| 0 | fester Wert |
| 1 | DIN0 |
| 2 | DIN1 |
| 3 | DIN2 |
| 4 | DIN3 |
| 5 | DIN4 |
| 6 | DIN5 |
| 7 | DIN6 |
| 8 | DIN7 |
| 9 | DIN8 |
| 10 | DIN9 |
| 11 | DIN10 |
| 12 | DIN11 |
| 13 | EA88_DIN 0_0 |
| 14 | EA88_DIN 0_1 |
| 15 | EA88_DIN 0_2 |
| 16 | EA88_DIN 0_3 |
| 17 | EA88_DIN 0_4 |
| 18 | EA88_DIN 0_5 |
| 19 | EA88_DIN 0_6 |
| 20 | EA88_DIN 0_7 |
| 21 | DIN_AIN1 |
| 22 | DIN_AIN2 |
| 23 | EA88_DIN 1_0 |
| 24 | EA88_DIN 1_1 |
| 25 | EA88_DIN 1_2 |
| 26 | EA88_DIN 1_3 |
| 27 | EA88_DIN 1_4 |
| 28 | EA88_DIN 1_5 |
| 29 | EA88_DIN 1_6 |
| 30 | EA88_DIN 1_7 |

10.3 Referenz/Auflistung von schon angewendeten Befehlen

Tabelle 13: Bewegungs- und Achsbefehle

| |
|-----------|
| BASE |
| CAM |
| CANCEL |
| CONNECT |
| DATUM |
| DEFPOS |
| FORWARD |
| MOVE |
| MOVEABS |
| MOVECIRC |
| RAPIDSTOP |
| REGIST |
| REVERSE |

Tabelle 14: Ein- und Ausgabebefehle

| |
|---------------------|
| AIN |
| AIN0..3 / AINBIO..3 |
| AOUT0...3 |
| CHR |
| FLAG |
| GET |
| GET# |
| HEX |
| IN()/IN |
| INVERT_IN |
| KEY |
| LINPUT |
| OP |
| PRINT |
| PRINT# |
| READ_OP() |

Tabelle 15: Programmschleifen und Programmstrukturen

| |
|----------------------|
| BASICERROR |
| ELSE |
| ELSEIF |
| ENDIF |
| FOR..TO.. STEP..NEXT |
| GOSUB |
| GOTO |
| ON.. GOSUB |
| ON.. GOTO |
| THEN |
| UNTIL |
| WA |
| WAIT IDLE |
| WAIT UNTIL |
| WEND |
| WHILE |

Tabelle 16: Systemparameter und Systemkommandos

| |
|----------------|
| CLEAR |
| DIR |
| ERROR_AXIS |
| FEATURE_ENABLE |
| FLASHVR |
| MOTION_ERROR |
| RUN |
| RUNTYPE |
| SERVO_PERIOD |
| STOP |
| TABLE |
| TIME |
| TSIZE |
| VR |
| VRSTRING |
| WDOG |
| # |
| \$ |
| INDEVICE |
| PROC_LINE |
| PROC_STATUS |
| RUN_ERROR |
| TICKS |

Tabelle 17: Mathematische Operationen und Kommandos

| |
|-----------------------|
| + Add |
| - Subtract |
| * Multiply |
| / Divide |
| = Equals |
| <> Not Equal |
| > Greater Than |
| <= Less Than or Equal |
| ABS |
| AND |
| CONSTANT |
| GLOBAL |
| MOD |
| OR |
| READ_BIT |
| SGN |
| SQR |
| XOR |

Tabelle 18: Konstanten

| |
|-------|
| OFF |
| ON |
| FALSE |
| PI |
| TRUE |

Tabelle 19: Achsparameter

| |
|--------------|
| ACCEL |
| ATYPE |
| AXIS_ENABLE |
| AXISSTATUS |
| CREEP |
| DECEL |
| DRIVE_CLEAR |
| DRIVE_HOME |
| DRIVE_READ |
| DRIVE_RESET |
| DRIVE_STATUS |
| DRIVE_WRITE |
| ERRORMASK |
| FE_LIMIT |
| FE_RANGE |
| MERGE |
| REG_POS |
| SERVO |
| SPEED |
| UNITS |