

Anhang A

Verhaltensspezifikation der Motoren-Komponente

DOKUMENTENNAME:

Verhaltensspezifikation: XCTL-Steuerprogramm
Teil: Motorsteuerungs-Komponente (motors.dll)

Autor: S. Lützkendorf

Dokumentversion: 0.8 (Okt. 2001) Entwurf

Inhaltsverzeichnis

1.	Überblick	185
2.	Funktionale Beschreibung	185
F1	Bestimmung der absoluten Motorposition	185
F1.1	Systeminitialisierung und Bestimmung der Koordinatenverschiebung	186
F1.2	Verwaltung der Relativen Null	187
F1.3	Berücksichtigung des Motorspiels	188
F1.4	Umrechnung in physikalische Einheiten	188
F2	Grundstellung anfahren	189
F2.1	Kalibrierung	191
F2.2	Referenzpunktlauf	191
F2.3	Setzen der absoluten Null	193
F3	Motor-Parameter einstellen	193
F3.1	Parameter-Einstellung	194
F3.2	End-Schalter	195
F4	Verfahren nach Encoder-Position	195
F5	Bereichsmaximierung	196
F6	Optimieren	196
F6.1	Optimieren eines C-812er Motors	197
F6.2	Optimieren eines C-832er Motors	197
F6.3	Beschleunigungskurve ermitteln	198
F7	Motorenverwaltung	199
F7.1	Motor-Liste und Aktueller Motor	199
F7.2	Achsenzuordnung von Motoren	200
F7.3	Achsenspezifisches Motorenverhalten	201
3.	Daten	202
3.1	Benutzerschnittstelle (Dialoge)	202
3.2	Konfigurations-Parameter (INI-Einträge)	202
4.	Fehler	210
5.	Änderungswünsche	210
6.	Probleme, Offene Fragen	210
7.	Verwandte Dokumente	211
8.	Änderungen	212
9.	Quellen	212
10.	Glossar	212

1. Überblick

Die Motorsteuerungs-Komponente enthält die Kernfunktionen zur Initialisierung, Ansteuerung und Verwaltung von Motoren im XCTL-Programm. Sie realisiert eine auf relativ abstraktem Niveau definierte Schnittstelle, mit deren Hilfe andere Komponenten komplexere Steuerungsabläufe implementieren können.

2. Funktionale Beschreibung

F1 Bestimmung der absoluten Motorposition

Die vom Motorcontroller verwaltete interne Motorposition kann in eine absolute Motorposition umgerechnet werden, wenn die Verschiebung zwischen dem Koordinatensystem der absoluten und dem der internen Motorposition bekannt ist (F1.1). Außerdem ist zur Bestimmung der absoluten Motorposition das Motorspiel zu berücksichtigen (F1.2) und es sind die Motorencoder-Schritte in physikalische Einheiten umzurechnen (F1.3).

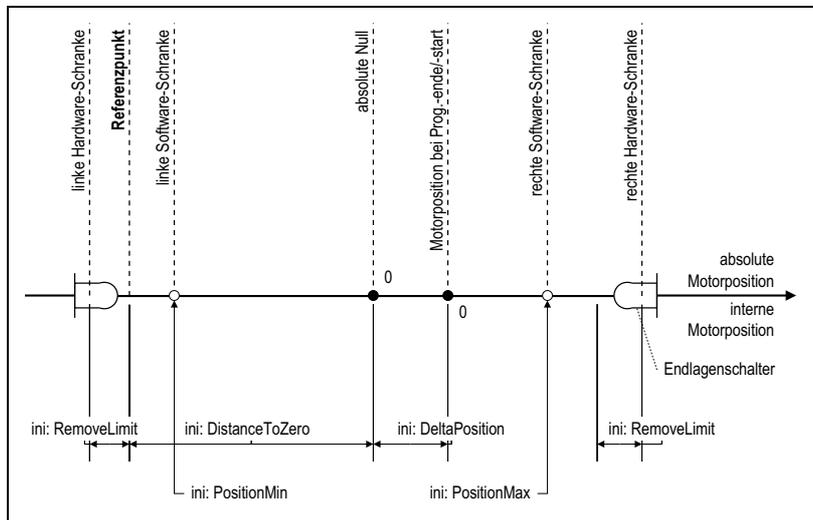


Abbildung A.1: Referenzpunkt- und Positionsbestimmung und Konfigurationsparameter (beim C-812, und beim C-832 ohne Verwendung des Index-Schalters).

F1.1 Systeminitialisierung und Bestimmung der Koordinatenverschiebung

Es gibt 2 Möglichkeiten nach dem Systemstart den Abstand zwischen der Absoluten Null und der Internen Null zu bestimmen:

1. Durch einen Referenzpunktlauf, bei dem eine Position angefahren wird, dessen Abstand zur Absoluten Null bekannt ist, und die zur internen Null gesetzt wird. Oder
2. über Konfigurationsdaten; dabei werden Informationen verwendet, die vor dem letzten Ausschalten des Systems gespeichert wurden.

Die Variante über Konfigurationsdaten kann nur dann fehlerfrei funktionieren, wenn sicher gestellt wird

1. dass in der Zeit in der die XCTL-Software nicht läuft, die absolute Motorposition nicht verändert wird, weder von Hand noch durch eine andere Software;
2. die XCTL-Software die Konfigurations-Daten korrekt abgespeichert hat; und
3. die Konfigurations-Daten nicht außerhalb des Programms verändert wurden.

Die Möglichkeit des Referenzpunktlafes dagegen ist weniger von aktuellen Konfigurationsdaten abhängig und damit weniger fehleranfällig.

Gründe dafür trotzdem auch die zweite Variante zu implementieren sind: a) dass ein Referenzpunktlauf Zeit kostet und b) dass es für einige Achsen wegen der Lebensdauer sinnvoll ist, größere Bewegungen zu vermeiden.

Die **Bestimmung der Koordinatenverschiebung aus den Konfigurationsdaten** erfolgt mit dem Parameter `DeltaPosition`. Dieser speichert den Abstand zwischen der Absoluten Null und der Absoluten Motorposition beim letzten ordnungsgemäßen Verlassen des Programms. Beim Neustart des Programms wird die Interne Motorposition auf Null gesetzt, damit ist `DeltaPosition` der Abstand zwischen der Absoluten und der Internen Motorposition, entspricht also der gesuchten Koordinatenverschiebung.

(*)	Die absolute Motorposition wird berechnet als Summe der vom Controller gelieferten Internen Motorposition und der Koordinatenverschiebung <code>DeltaPosition</code> .
-----	--

Ob die Konfigurationsdaten brauchbar sind, signalisiert der Konfigurationsparameter `RestartPossible`. Dieser wird bei der Initialisierung des Software-Systems auf `False` gesetzt, und wird erst beim ordnungsgemäßen Beenden des

Programmes und nach dem Speichern des aktuellen Wertes von `DeltaPosition` auf `True` gesetzt. Falls also das Programm nicht korrekt beendet wurde - z.B. durch einen Absturz - ist beim nächsten Programmstart die Ungültigkeit der Konfigurationsdaten erkennbar.

Wenn die Konfigurationsdaten nicht gültig sind, muss ein **Referenzpunktlauf** durchgeführt werden. Dabei wird eine Motorposition angefahren, deren Abstand zur Absoluten Null bekannt ist. Dieser Abstand wird während der Kalibrierung (s.u.) bestimmt und im Konfigurationsparameter `DistanceToZero` gespeichert. Wenn diese Position erreicht ist, wird die Interne Motorposition auf Null gesetzt. Damit entspricht `-DistanceToZero` der gesuchten Koordinatenverschiebung und wird in `DeltaPosition` gespeichert. Damit gilt dann wieder (*).

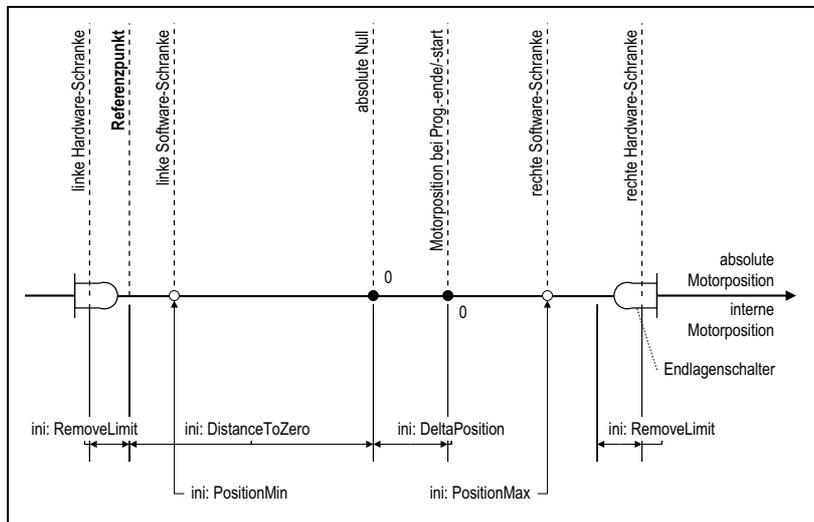


Abbildung A.2: Referenzpunkt- und Positionsbestimmung und Konfigurationsparameter (beim C-832 mit Verwendung des Index-Schalters, reference switches)

F1.2 Verwaltung der Relativen Null

Für jeden Motor kann temporär der Koordinatenursprung der absoluten Positionen verschoben werden. Der neue Koordinatenursprung heißt Relative Null.

Nach der Initialisierung ist bei allen Motoren die Relative Null gleich der Absoluten Null.

F1.3 Berücksichtigung des Motorspiels

Das Motorspiel ist ein systematischer Anteil am Positionierungsfehler der Motoren der durch notwendige Fertigungstoleranzen bei Getrieben entsteht. Da dieser Anteil systematisch ist, kann er durch die Software korrigiert werden.

Bei jeder *Änderung der Bewegungsrichtung* muss berücksichtigt werden, dass der Motor eine bestimmte Anzahl von Umdrehungen durchführt ohne, dass sich der Motorschlitten bewegt. D.h. der Motorcontroller bekommt Encoderimpulse, stellte also eine Änderung der Internen Motorposition fest, ohne dass sich die Absolute Motorposition ändert. Diese Differenz zwischen Interner und Absoluter Motorposition muss die Software bei den Positionierungskommandos verrechnen.

Fall 1) wenn ein Motor als letztes eine Linksbewegung zu Position X ausgeführt hat und dann um einen Betrag D nach rechts bewegt werden soll, muss die Interne Zielposition also durch $X+D+Spiel$, und die Absolute Zielposition durch $X+D$ berechnet werden.

Eine Richtungsänderung von links nach rechts entspricht also einer Verschiebung der Internen Null um $+D$.

Fall 2) wenn ein Motor als letztes eine Rechtsbewegung zu Position X ausgeführt hat und dann um einen Betrag D nach links bewegt werden soll, muss die Interne Zielposition also durch $X-D-Spiel$, und die Absolute Zielposition durch $X-D$ berechnet werden.

Eine Richtungsänderung von rechts nach links entspricht also einer Verschiebung der Internen Null um $-D$.

Das experimentell zu bestimmende Motorspiel wird als Parameter **Hysteresis** in der Konfigurationsdatei gespeichert. Der Konfigurationsparameter **Upwards** speichert die Richtung der letzten Motorbewegung.

FEHLER: Die aktuelle Implementation der Behandlung des Motorspiels ist fehlerhaft (z.B. wird der Parameter **Upwards** zwar in der Konfigurationsdatei gespeichert aber nie gelesen)

F1.4 Umrechnung in physikalische Einheiten

TODO (Positionen, Geschwindigkeiten, ...)

Falls **Correction=0** werden Motoreinheiten in physikalische Einheiten mit Hilfe des Parameters **Koeff_1** umgerechnet.

Falls **Unit** ein Bogenmaß (Grad, Minute, Sekunde) ist gibt **Koeff_1** das Verhältnis zwischen einer Motoreinheit und einer Winkelsekunde an
Andernfalls gibt **Koeff_1** das Verhältnis zwischen einer Motoreinheit und einer physikalischen Einheit an.

Beispiele

- bei `Unit=cm` und `Koeff_1=0.0025`
entspricht eine Motorposition in Motoreinheiten von 5000 einer Motorposition von 12.5cm.
- bei `Unit=Sekunden` und `Koeff_1=0.0025`
entspricht eine Motorposition in Motoreinheiten von 5000 einer Motorposition von 12.5 Sekunden.
- bei `Unit=Minuten` und `Koeff_1=0.0025`
entspricht eine Motorposition in Motoreinheiten von 5000 einer Motorposition von 12.5 Sekunden und damit 0.21 Minuten.

`Koeff_1`-Parameter können negative Werte enthalten. Dies führt zu einer Spiegelung des Koordinatensystems der Absoluten Positionen zu dem des Koordinatensystems der Internen Positionen. Mit anderen Worten zu einer Uminterpretation der Bewegungsrichtung des Motors. Am Beispiel: `Unit=sekunden` und `Koeff_1=-0.25` kann man dann so lesen: ein Encoder-Schritt in negativer Richtung entspricht einer viertel Winkelsekunde in positiver Richtung.

FRAGE: Was gilt für die Softwareschranken für den Fall, dass `Koeff_1` negativ ist?

Zur Umrechnung von Geschwindigkeiten aus *Motoreinheiten/Sekunde* in *Physikalische Einheiten/Sekunde* wird der Parameter `SpeedScale` verwendet. Dabei gilt
Geschwindigkeit in Physikalische Einheiten/Sekunde * `SpeedScale` = *Geschwindigkeit in Motoreinheiten/Sekunde*.

ANM: Scheinbar ist der Parameter `SpeedScale` redundant, da eine feste Relation zu `Koeff_1` bestehen sollte.

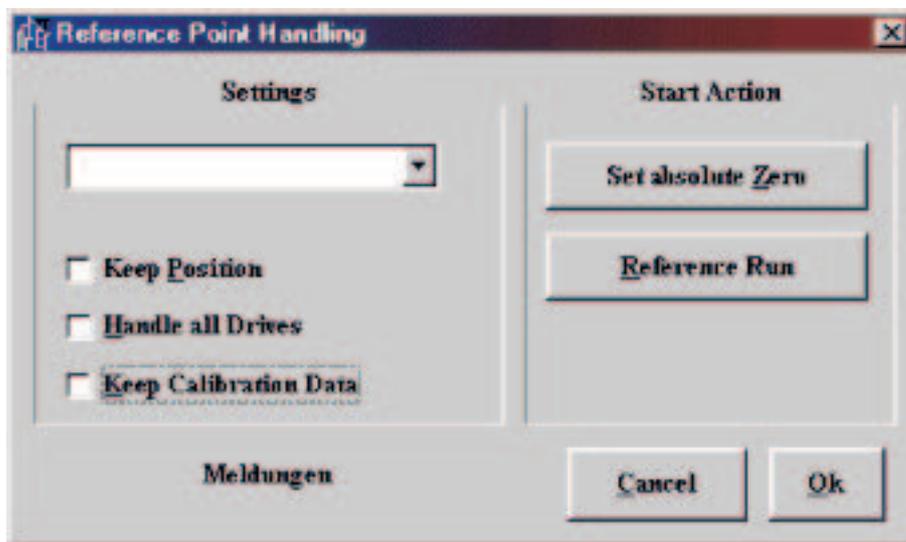
Beispiele

- Falls `SpeedScale=14400` und `MaxVelocity=128000` und `Unit=Grad`, so ist die max. erlaubte Geschwindigkeit gleich 8.89 Grad/Sekunde. (zu diesem `SpeedScale`-Wert gehört ein `Koeff_1` von 0,25 [Winkel-Sekunden / Motorschritt])

F2 Grundstellung anfahren

Die Funktion heißt auch **Reference Point Handling**.

DIALOGBOX



mit dieser Dialogbox lassen sich folgende Teilfunktionen ausführen:

	Option "Keep Position"	Option "Handle all Drives"	Option "Keep Calibration Data"	Button "Reference Run"	Button "Set Absolute Zero"
1. Kalibrierung eines Motors					
2.1 Referenzpunktlauf für einen Motor mit anschliessendem Anfahren der Grundstellung	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.2 Referenzpunktlauf für einen Motor mit anschliessendem Anfahren der Ausgangsstellung	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.3 Referenzpunktlauf für alle Motoren mit anschliessendem Anfahren der Grundstellung	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.4 Referenzpunktlauf für alle Motoren mit anschliessendem Anfahren der Ausgangsstellung	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. "Setzen der absoluten Null"					<input checked="" type="checkbox"/>

(Leere Felder bedeuten, dass das Steuerelement für die Teilfunktion keine Bedeutung hat. Der Button mit dem Bullet löst die Funktion aus.)

Die Teilfunktionen die für einem Motor gelten, arbeiten auf dem in der Listbox ausgewähltem Motor. Die Teilfunktionen die auf allen Motoren arbeiten,

ignorieren die Listbox-Einstellung und führen die Funktion für alle Motoren parallel durch. FRAGE1

Die Teilfunktion "Setzen der absoluten Null" arbeitet nur auf dem in der Listbox ausgewähltem Motor. FRAGE2

F2.1 Kalibrierung

Aufgabe der Kalibrierung ist es, das Software-System mit einer neuen bzw. veränderten Motorenkonfiguration abzustimmen. Ziel ist es den Abstand zwischen Referenzpunkt und Absoluter Null zu bestimmen. Dieser Abstand wird in dem Konfigurationsparameter `DistanceToZero` gespeichert. Mit Hilfe dieses Wertes lässt sich die Lage des Ursprungs des Koordinatensystems der absoluten Positionen bestimmen.

Die Kalibrierung ist zu wiederholen, wenn sich an der Motorenkonfiguration etwas verändert hat; z.B. Veränderung der Hardwareschranken, der Referenz-Schalter,

Ablauf (unvollständig)

1. (Einrichtung der Motorenkonfiguration entsprechend dem Versuchsaufbau)
2. Bestimmung wichtiger Konfigurationsparameter:
 - Parameter zur Motoransteuerung
 - linke und rechte Software-Schranke (`PositionMin` und `PositionMax`)
 - `Koeff_1`, `SpeedScale`
3. eventuell Optimierung des Motors, s. F6
4. Positionierung des Motors auf der Absoluten Null
5. Referenzpunktlauf mit der Option "`Keep Calibration Data`" == `false`.

(aus Dokumentation: "`DistanceToZero`: ... Diese Größe ist für viele Antriebe durch explizites Messen einer Kalibrierungskurve festgelegt. In anderen Fällen kann die physikalische Null mit dem Referenzpunkt in Beziehung gesetzt werden." FRAGE3: was für Kalibrierungskurven sind gemeint?)

F2.2 Referenzpunktlauf

Ziel des Referenzpunktlaufes ist es, eine Motorposition anzufahren, von welcher der Abstand zur Absoluten Null bekannt ist. Setzt man diese Position zur Internen Null, kann man mit dem bekannten Abstand die Internen Motorpositionen in Absolute umrechnen. Am Referenzpunkt gilt: `DeltaPosition==DistanceToZero`.

Ablauf

1. Speichern der aktuellen Internen Position
2. Anfahren des Referenzpunktes; der Referenzpunkt wird Motor/Controller-spezifisch angefahren:

C-812

Der Motor wird angewiesen eine Position anzufahren, von der sicher ist, dass sie links der linken Hardware-Schranke liegt.

D.h. der Motor fährt gegen die linke Hardware-Schranke. Dies löst ein Signal aus, auf das der Motor automatisch vom Controller um eine bestimmte Distanz zurückgefahren wird (s. Parameter `RemoveLimit`). Die Position an der der Motor stehen bleibt, ist per definitionem der Referenzpunkt. (s. Abb. 1)

C-832 allgem.

Der C-832-Controller hat eine spezielle Index-Line (Reference switch input). D.h. man kann einen extra Schalter verwenden, der dem Controller des Erreichen einer speziellen Position signalisiert. (s. Abb. 2)

Ob ein solcher Schalter in der aktuellen Versuchsanordnung verwendet wird, legt der Konfigurationsparameter `IndexLine` fest.

C-832 (`IndexLine == false`)

In diesem Fall wird der Referenzpunkt wie beim C-812er Controller bestimmt; mit dem Unterschied, dass das Zurückfahren nicht automatisch vom Controller ausgelöst wird, sondern von der Software. (FRAGE: Das ist etwas problematisch, da die Software abfragen muss ob eine Hardware-Schranke erreicht worden ist, und der Zeitpunkt (bzw. die Position) wann dies erfolgt, nicht immer die gleiche sein muss bzw. kann.)

C-832 (`IndexLine == true`)

Falls `MoveFirstToLimit == true` wird zuerst die linke Hardware-Schranke angefahren, in dem eine Ziel-Position links davon angegeben wird; und nachdem diese erreicht wurde, wird der Motor nach rechts in Bewegung gesetzt (in dem eine Position sehr weit rechts als Ziel angegeben wird). Der Motor wird gestoppt sobald das Index-Signal empfangen wurde. Dass dieses Signal kommt ist sichergestellt, da von der "linkesten" Position mit der Suche begonnen wird. `MoveFirstToLimit == true` stellt also sicher, dass der Index-Schalter stets von der gleichen Seite angefahren wird. (FRAGE4: was passiert wenn es gar keine Index-Line gibt?)

Ist `MoveFirstToLimit == false` wird der Motor in Richtung der linken Hardware-Schranke bewegt und gestoppt sobald das Index-Signal empfangen wurde. (FRAGE5: was passiert wenn der Referenzpunktlauf links vom Referenzpunkt beginnt?).

3. Falls der Referenzpunktlauf der Kalibrierung dient ("`Keep Calibration Data`" `== false`), kann `DistanceToZero` aus der Distanz zwischen der gespeicherten Internen Position (s. Schritt 1) und der aktuellen Internen Position bestimmt werden.

4. Die Interne Position wird auf Null gesetzt (Home-Position); und `DeltaPosition` auf `-DistanceToZero`; von nun an gilt wieder (*).
5. Falls der Referenzpunktlauf mit der Option "Hold Position" == true gestartet wurde, wird nun die Motorposition angefahren, die der Motor vor dem Referenzpunktlauf inne hatte; andernfalls wird die Position angefahren die im Konfigurationsparameter `InitialAngle` steht.

Der Referenzpunktlauf wird nicht für Motoren zugelassen mit den Konfigurations-Parameter `InitialMove==0`.

F2.3 Setzen der absoluten Null

Legt die aktuelle Motorposition als Absolute und Interne Null fest. D.h. `DeltaPosition` wird auf Null gesetzt und die Interne Motorposition wird ebenfalls auf Null gesetzt (Define Home).

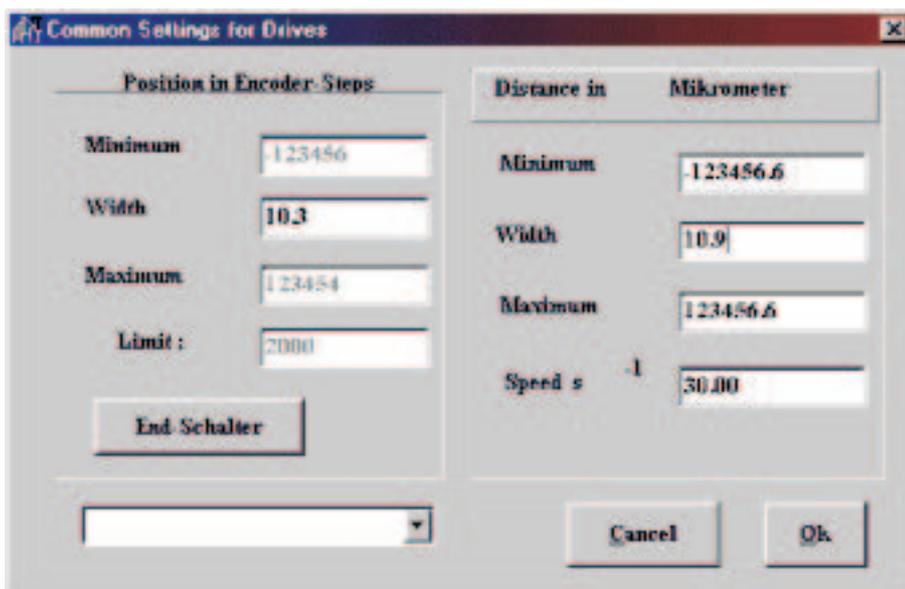
Nach dem "Setzen der absoluten Null" ist der entsprechende Motor im Zustand "nicht kalibriert". FRAGE6

FRAGE7: Wann wird diese Funktion verwendet?

F3 Motor-Parameter einstellen

Die Funktion heißt auch **Common Settings for Drives**

DIALOGBOX



F3.1 Parameter-Einstellung

Über diesen Dialog können folgende allgemeine Konfigurationsparameter für den Aktuellen Motor eingestellt werden:

- `PositionMin`
- `PositionWidth`
- `PositionMax`
- `RemoveLimit`
- `AngleMin`
- `AngleWidth`
- `AngleMax`
- `Velocity`

Für die Konfigurationsparameter gelten die im Kapitel 3.2 aufgeführten Angaben. Es ist sicher zu stellen, dass die dort angegebenen Einschränkungen nicht verletzt werden.

Der Aktuelle Motor wird in der Listbox ausgewählt. Dieser Motor bleibt der Aktuelle Motor auch nach Beendigung der Funktion.

FRAGE8: Warum sind die Steuerelemente für `PositionMin`, `PositionMax` und `RemoveLimit` disabled?

Positions-Softwareschranken

Das XCTL-System verwaltet für die absolute Position 2 x 2 Softwareschranken. Zum ersten `PositionMin` und `PositionMax` deren Werte in Motorencoder-Schritten angegeben werden und zum zweiten `AngleMin` und `AngleMax`, deren Werte in physikalische Einheiten gemäß `Unit` angegeben werden.

FRAGE9: Wozu die doppelte Buchführung die nur zu Fehlern führen kann? Die Werte für `AngleMin` und `AngleMax` können problemlos aus `PositionMin` und `PositionMax` berechnet werden. Genau dies stellt auch die Funktion "Bereichsmaximierung" (s.u.) zur Verfügung. `AngleMin` und `AngleMax` werden in der Implementation nur während der Umrechnung von Nutzer-Einheiten in Motor-Schritte verwendet, und das Ergebnis der Umrechnung wird dann noch mal gegen `PositionMin` und `PositionMax` getestet.

Wenn `AngleMin` und `AngleMax` doch benötigt werden, sollte das Verhältnis zu `PositionMin` und `PositionMax` definiert werden; etwa: das Intervall `AngleMin` und `AngleMax` kann nur im Intervall `PositionMin` und `PositionMax` liegen.

Ähnliches gilt für `PositionWidth` und `AngleWidth`.

F3.2 End-Schalter

Diese Funktion veranlasst den aktuellen Motor in regelmäßigen Abständen zu überprüfen ob eine der Hardware-Schranken erreicht wurde. Der Motor informiert (via Event) den Anwendungsrahmen über das Testen der Hardware-Schranke und gegebenenfalls über das Erreichen einer Hardware-Schranke.

Der erste Button-Click startet diese Überwachungsfunktion, der zweite Click beendet sie. (das sollte in der Button-Beschriftung erkennbar gemacht werden.)

Diese Überwachungsfunktion ist nicht an diesen Dialog gebunden. D.h. nach dem Schließen des Dialoges (und eigentlich erst dann) bleibt diese Funktion aktiv.

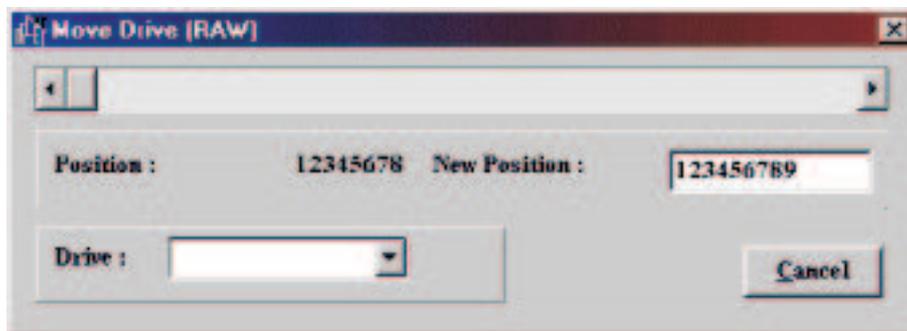
FRAGE10: wann wird diese Funktion benötigt?

ANM: Die entsprechenden Methoden sind nur für den C-832er Controller implementiert. D.h. beim C-812er passiert beim Button-Click nichts.

F4 Verfahren nach Encoder-Position

Die Funktion heißt auch **Direct Steering, Move Drive (RAW)**

DIALOGBOX



Diese Funktion ermöglicht den Aktuellen Motor über die Angabe einer internen Motorposition in Encoderschritten zu bewegen. Die möglichen Motorpositionen werden dabei durch die Softwareschranken `PositionMin` und `PositionMax` beschränkt.

Der Aktuelle Motor wird in der Listbox ausgewählt. Dieser Motor bleibt der Aktuelle Motor auch nach Beendigung der Funktion.

Die Ziel-Position, d.h. die interne Motorposition die angefahren werden soll, kann entweder über die Textbox "New Position" oder über eine Bewegung des Scrollbars bestimmt werden.

Wird die Ziel-Position über die Textbox angegeben, wird getestet ob die

angegebene Ziel-Position jenseits einer Software-Schranke liegt. Ist dies der Fall wird die Ziel-Position durch die Position dieser Software-Schranke ersetzt. Dadurch wird sichergestellt, dass die Zielposition immer zwischen den Software-Schranken liegt. Dann wird der Motor angewiesen die Zielposition anzufahren. Der Motor kann um jeden beliebig kleinen positiven Betrag bewegt werden.

Über den Scrollbar kann die Positionsänderung nur in Vielfachen des Parameters `PositionWidth` erfolgen. Eine "kleine" Bewegung entspricht $1 \times \text{PositionWidth}$, eine "große" entspricht $5 \times \text{PositionWidth}$. FEHLER1

Während der Motor in Bewegung ist kann keine neue Ziel-Position angegeben werden.

Das Schließen des Dialoges über den "Cancel"-Button oder über den "X"-Button während einer Bewegung stoppt den aktiven Motor unmittelbar. WUNSCH1

FEHLER2: die Auswahl eines anderen Motors während der aktuell gewählte Motor in Bewegung ist, sollte nicht möglich sein; in der gegenwärtigen Implementation ist dies möglich; wenn der aktuelle Motor in Bewegung ist, läuft er 'im Hintergrund' weiter; er wird auch nicht gestoppt wenn der Dialog geschlossen wird.

FRAGE11: Unter welchen Umständen wird diese Funktion verwendet?

WUNSCH2: Die Einheit der Positionsangaben sollte deutlich werden.

ANM: Der Menüpunkt, der diese Funktion aufruft (Einstellungen/Antriebe/Direkte Steuerung) steht nur im Expertenmodus zur Verfügung.

F5 Bereichsmaximierung

Diese Funktion wird durch die Methode `m1SetAngleDefault()` realisiert.

Setzt für alle Motoren die Relative Null gleich der Absoluten Null (`AngleBias==0`). Berechnet aus den Software-Schranken für die Internen Positionen (`PositionMin`, `PositionMax`) die Werte für die Software-Schranken der Absoluten Positionen (`AngleMin`, `AngleMax`).

F6 Optimieren

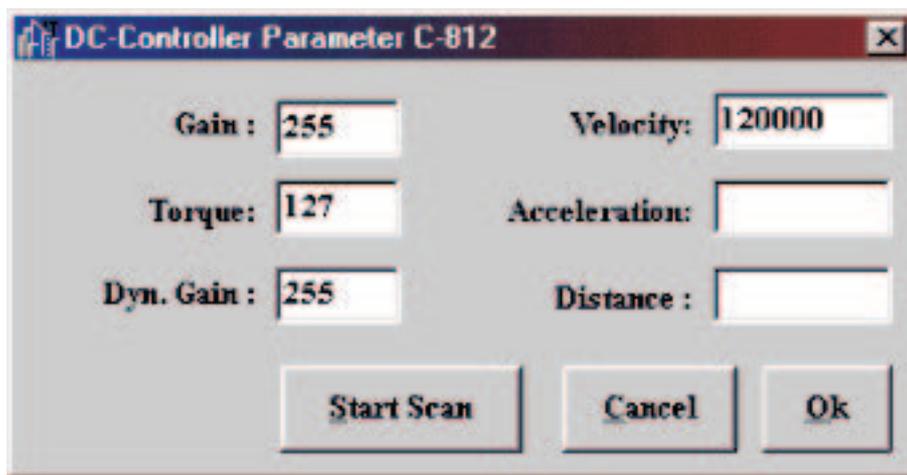
Unter dieser Funktion werden Optionen zur Einstellung Motor- bzw. Controllerspezifischer Parameter angeboten bzw. die Möglichkeit diese Parameter durch einen Probelauf zu testen.

Die Funktion arbeitet stets auf dem aktuellen Motor. Der Name des aktuellen Motors wird in der Titel-Leiste des Dialoges angegeben.

Vor dem Öffnen des jeweiligen Dialoges wird der Anwendungsrahmen darüber informiert (via Windows-Event), dass eventuell eine Diagrammfenster (für die Anzeige der Beschleunigungskurve) benötigt wird. WUNSCH3

F6.1 Optimieren eines C-812er Motors

DIALOGBOX



Über diesen Dialog können folgende Konfigurationsparameter für einen C-812er Motorcontroller eingestellt werden.

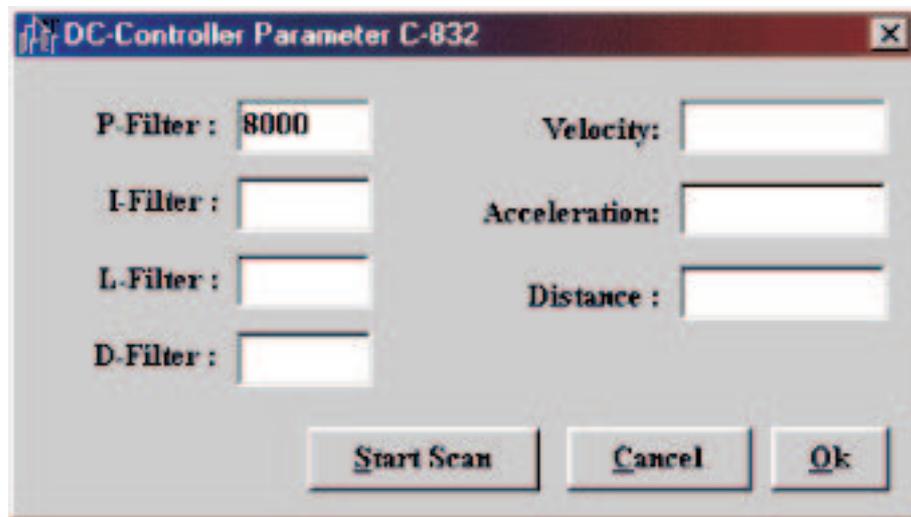
- StaticGain (Box Gain)
- Torque
- DynamicGain (Box Dyn. Gain)
- MaxVelocity (Box Velocity)
- Acceleration
- PositionWidth (Box Distance)

Für die Konfigurationsparameter gelten die im Kapitel 3.2 aufgeführten Angaben. Es ist sicher zu stellen, dass die dort angegebenen Einschränkungen nicht verletzt werden.

Der Button "Start Scan" löst die Funktion F6.3 aus.

F6.2 Optimieren eines C-832er Motors

DIALOGBOX



Über diesen Dialog können folgende Konfigurationsparameter für einen C-832er Motorcontroller eingestellt werden.

- DynamicGain (Box P-Filter)
- IntegralGain (Box I-Filter)
- IntegralLimit (Box L-Filter)
- Gain (Box F-Filter)
- MaxVelocity (Box Velocity)
- Acceleration
- PositionWidth (Box Distance !!)

Für die Konfigurationsparameter gelten die im Kapitel 3.2 aufgeführten Angaben. Es ist sicher zu stellen, dass die dort angegebenen Einschränkungen nicht verletzt werden.

Der Button "Start Scan" löst die Funktion F6.3 aus.

F6.3 Beschleunigungskurve ermitteln

Die Funktion taucht auch unter den Bezeichnungen **Start Scan**, **CheckScan**, **MoveScan** auf.

Ziel dieser Funktion ist es die aktuellen Bewegungsparameter aus dem "Optimierungs"-Dialog zu testen, sicher zu stellen, "dass die Endposition gleichmäßig und ohne Schwingungen angefahren wird".

Dies erfolgt durch einen **MoveScan**, d.h. in dem der aktuelle Motor um eine bestimmte Strecke verfahren wird, und in gleichmäßigen Abständen die Position ausgelesen wird. Damit erhält man die Daten für ein Ort-Zeit-Diagramm.

Nach Durchführung des MoveScans wird der Anwendungsrahmen (via Windows-Event) über die Beendigung informiert, und die Daten werden zur Verfügung gestellt. Der Anwendungsrahmen ist dafür verantwortlich diese Daten auszuwerten. (Im gegenwärtigen XCTL-System wird ein Weg-Zeit-Diagramm in einem "ScanFenster" angezeigt).

Der MoveScan wird wie folgt durchgeführt:

1. die aktuell im Dialog eingestellten Spannungs- und Bewegungsparameter werden aktiviert
2. der Motor wird in Bewegung gesetzt, mit der Zielposition: `AktuellePosition+PositionWidth`
3. im Abstand von 15 ms (C-812ISA), 10 ms (C-832) bzw. 20ms (C-812GPIB) wird die Position des Motors ausgelesen und gespeichert; die Position wird bis zum Stillstand des Motors, maximal aber 150 mal ausgelesen.

Die Daten eines MoveScans stehen solange zur Verfügung bis ein neuer MoveScan durchgeführt wird.

F7 Motorenverwaltung

F7.1 Motor-Liste und Aktueller Motor

Die Motorenkomponente verwaltet eine **Liste** aller in einer Konfiguration des XCTL-Systems verfügbaren Motoren.

Es wird von der Annahme ausgegangen, dass es in jeder Konfiguration wenigstens einen Motor gibt. Die Anzahl der Motoren ist nach oben durch die Software nicht beschränkt.

Die Motorenkomponente unterstützt folgende Steuerkarten für die Motoransteuerung:

- C-812
- C-832

Außerdem stellt die Motorenkomponente einen *Testmotor* zu Verfügung, der die Verwendung der Software ohne tatsächlich vorhandene Steuerkarten bzw. Motoren ermöglicht.

Wie viele Motoren, welchen Typs und mit welchen Parametern in einer Versuchsanordnung existieren, wird in einer Konfigurationsdatei gespeichert. Die Konfigurationsdatei wird im "Windows"-Verzeichnis erwartet.

Ein Motor aus der Motorenliste ist stets als aktiv gekennzeichnet, und heißt **Aktueller Motor**. Es gibt immer einen Aktuellen Motor. Nach der Initialisierung ist der erste Motor der Konfiguration der Aktuelle Motor.

F7.2 Achsenzuordnung von Motoren

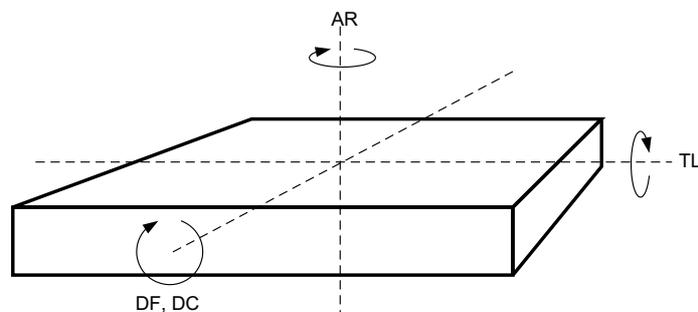
Die von der Motorenkomponente verwalteten Motoren haben, in der verschiedenen Versuchen in denen die Software verwendet wird, unterschiedliche Rollen: sie führen eine Bewegung entlang einer bestimmten Achse aus.

Im XCTL-System gibt es im Moment folgende Achsen:

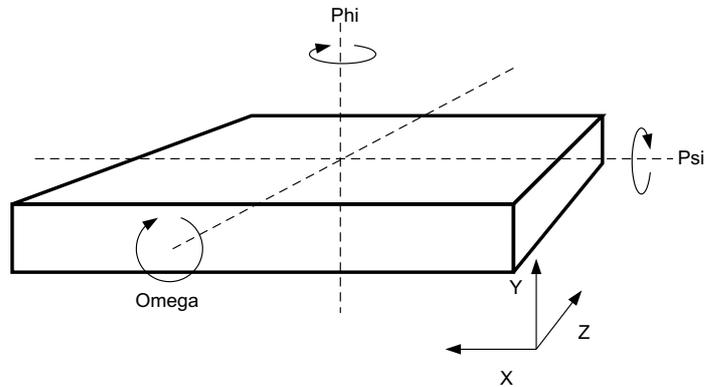
1. Absorber
2. Encoder
3. CC (Collimator, Kollimator)
4. DC (Beugung grob, Diffraction Coarse)
5. DF (Beugung fein, Diffraction Fine)
6. Monochromator
7. Omega
8. Phi
9. Psi
10. AR (Rotation)
11. Theta
12. TL (Tilt)
13. X
14. Y
15. Z

Die Bedeutung der Achsen in den verschiedenen Versuchsaufbauten ist folgende:

1. für die Topographie: DF, DC, AR, TL für den Probenhalter sowie CC



2. für die Diffraktometrie/Reflektometrie: Omega, Phi, Psi, X, Y, Z für den Probenhalter sowie CC



3. für ???: Absorber, Monochromator, Encoder, Theta; FRAGE13

Über den Namen eines Motors wird seine Zuordnung zu einer Achse bestimmt.

Einige Achsen (verschiedener Versuchsanordnungen) werden im XCTL-System einander *gleichgesetzt*; und zwar:

- AR und Phi
- DF und Omega
- TL und Psi

Es *sollte* in einer Konfiguration stets nur ein Motor für eine bestimmte (bzw. 2 gleichgesetzte Achsen) Achse existieren. Wenn es in einer Konfiguration mehrere Motoren gibt, die der gleichen Achse zugeordnet sind, ist die Zuordnung nur für den im Ini-File letzten Motor gültig. In der Motorliste sind aber trotzdem alle enthalten.

F7.3 Achsenspezifisches Motorenverhalten

F7.3.1 Abhängigkeiten zwischen DF und DC

TODO

Achtung: an einigen Stellen der Implementation wird implizit die Annahme gemacht, dass das Vorhandensein der DF-Achse das der DC-Achse impliziert (und umgekehrt); s. z.B. `TMotor::SetCorrectionState(B00L)`.

3. Daten

3.1 Benutzerschnittstelle (Dialoge)

Die Motorenkomponente implementiert folgende Dialoge:

1. Referenzpunkt
2. Allgemeine Motorparameter
3. Verfahren nach Encoder-Schritten
4. Optimieren eines C-812er Motors
5. Optimieren eines C-832er Motors

Die Dialoge sind jeweils bei den entsprechenden Funktionen abgebildet und erläutert.

3.2 Konfigurations-Parameter (INI-Einträge)

Einige INI-Einträge sind in ihrer Bedeutung bzw. Gültigkeit vom Parameter Type abhängig. Wenn dies der Fall ist, steht der entsprechende Type in Klammern nach dem Parameternamen.

Abschnitte in Hochkommata (") sind i.d.R. aus den vorhandenen Dokumentationsfragmenten.

Allgemein

Für jeden Motor ist im Konfiguration-File eine Sektion anzulegen. Die Sektionen müssen [MotorX] heißen wobei X eine Zahl ist. X muss bei 0 beginnen und die Nummerierung der vorhandenen Sektionen muss durchgehend sein; d.h. die letzte Motor-Sektion, die gelesen wird, ist die [MotorX]-Sektion für die es keine nachfolgende Sektion [MotorX+1] mehr gibt. Es gibt keine Beschränkung für die Anzahl der Motor-Sektionen.

Die Parameter in den Sektionen sind folgende:

Name

Ein im Prinzip beliebiger Bezeichner der der Identifikation des Motors in der Nutzeroberfläche dient.

Werte: Zeichenkette

Default: "Motor"

Es gibt einige vordefinierte Bezeichner die einem Motor eine spezielle Rolle als Achse in einem bestimmten Versuchsaufbau zuweisen. Einige Achsen werden miteinander identifiziert. Bei der Achsenzuordnung wird Groß/Kleinschreibung des Name berücksichtigt.

vordefinierter Name	zugeordnete Achse
AzimutalRotation AZ AR Rotation Azimute	AR = Phi
X, x Horizontal x-Achse x-Axis	X
Y, y y-Achse y-Axis	Y
Z, z Vertical z-Achse z-Axis	Z
Theta	Theta
Omega	Omega = DF
Tilt TL	Tilt = Psi
DC Diffraction coarse Beugung grob	DC
DF Diffraction fine Beugung fein	DF = Omega
Psi	Psi = Tilt
Phi	Phi = AR
CC Collimator Kollimator	CC
Absorber	Absorber
Monochromator	Monochromator
Encoder	Encoder

Type

Hardware/Implementations-Typ des Motor-Controller.

Werte: [C-812ISA | C-812GPIB | C-832 | TMotor]

Default: TMotor

Einträge zur Hardware-Ansteuerung**BoardId**

Controller-Nummer des angeschlossenen Motors; "Ordnet dem logischen

- Motor einen Kanal auf der Steuerkarte zu”
Werte: [0 | 1] (C-832); [1 .. 4] (C-812ISA, C-812GBIB)
Default: 1
- RamAddr** (C-812ISA)
Speicheradresse, für den Direktspeicherzugriff zur Kommunikation Software-Hardware.
Werte: Hex-Zahl aus Bereich: C800 ... DE00
Default: 0xD000
- IoAddr** (C-832, C-812GBIB)
Port-Adresse für die Software-Hardware-Kommunikation.
Werte: Hex-Zahl ”above 200(hex)”(C-832)
Default: 0x210 (C-832); 0x200 (C-812GBIB)
- GPIBAddr** (C-812GBIB)
Default: 1 oder 15 (??)
- DifferentialEncoder** (C-832)
”Die Motorsteuerung ist in der Lage, neben einfachen auch differentielle Encodersignale zu verarbeiten”
Werte: [0 | 1]
Default: 0
- EnableInterrupts** (C-832)
??
Werte: [0 | 1]
Default: 0
- InquireStatus** (C-812GPIB)
Flag das angibt ob der Stillstand eines Motors über das Statusbit ermittelt wird.
Werte: [0 | 1]
1: der Stillstand eines Motors wird über das Statusbit des Controllers ermittelt
0: der Stillstand eines Motors wird durch Test ermittelt ob ist der Positionsfehler kleiner gleich **DeathBand** ist
Default: 0

Einträge zum Referenzpunkt und zur Positionsbestimmung

- DistanceToZero**
Distanz zwischen Referenzpunkt und absoluter Null.
Wird während der Kalibrierung bestimmt.
Werte: [?? .. ??]
Einheit: Motorencoderschritte
Default: 0
Anm.: die Absolute Null kann auch außerhalb des verfahrbaren Bereichs liegen, und kann auch negativ sein.
- IndexLine** (C-832)
Vorhandensein eines speziellen Index-Schalters/Inputs beim Motor/Controller. Dieser Schalter kennzeichnet eine als Referenzpunkt verwendbare Position.

Der C-832 hat eine solche Index-Line ("die Antriebe Omega und Theta des HRM-Diffraktometer besitzen Indizes für Referenzpunktläufe").

Dieser Parameter wird nur beim C-832 berücksichtigt. Werte: [0 | 1]

0: keine Index-Line

1: Index-Line vorhanden

Default: 0

MoveFirstToLimit (C-832)

Bei Existenz einer Index-Line, d.h. `IndexLine=1`, gibt der Parameter an ob der Motor erst die linke Endlage anfahren soll, bevor die Indexposition gesucht wird. Das stellt sicher, dass der Referenzpunkt immer von der gleichen Seite angefahren wird.

Werte: [0 | 1]

0: gleich Indexposition suchen

1: erste linke Hardwareschranke und dann Indexposition anfahren

Default: 0

InitialMove

"Gibt an, ob mit dem Antrieb die Grundstellung programmgesteuert angefahren werden kann," das ist z.B. sinnvoll wenn ein neuer Motor zwar schon im System integriert ist, aber noch keine Hardwareschranken bzw. Indexschalter hat, die für den Referenzpunktlauf erforderlich sind.

Dieser Konfigurationsparameter kann nur über das ini-File gesetzt werden.

Werte: [0 | 1]

0: verhindert die Nutzung/Durchführung der Funktionen Kalibrierung und Referenzpunktlauf.

Default: 0

InitialAngle

Gibt an, welche absolute Position nach dem Referenzpunktlauf angefahren werden soll.

(Wird nicht berücksichtigt falls der Referenzpunktlauf mit der Option "Hold Position" durchgeführt wird. Häufig ist dies die absolute Null.)

Werte: [real] ???

Einheit: ???

Default: 0.0

RemoveLimit

Distanz um die der Motor bei Ansprechen einer Hardwareschranke automatisch zurückgefahren wird.

(Beim C-812er Controller übernimmt das automatische Zurückfahren der Controller selbst; beim C-832er muss dafür die Software sorgen.)

Werte: [0 .. ???]; sollte so gewählt werden, dass der Endlagenschalter entspannt wird.

Einheit: Motorencoderschritte.

Default: 4000

Hysteresis

Motorspiel

Werte: [0 .. ???]

Einheit: Motorencoderschritte

Default: 0

DeltaPosition

absolute Motorposition beim Verlassen des Programms.

Einheit: Motorencoderschritte

Werte: [?? .. ??]

Default: 0 ??

Upwards

Richtung der zuletzt durchgeführten Bewegung.

Werte: [0, 1]

1: "Aufwärts", Bewegung nach rechts

0: "Abwärts", Bewegung nach links

Default: 1

RestartPossible

Gibt an, ob das Programm ordnungsgemäß verlassen wurde, d.h. dass die Kalibrierungsdaten im ini-File korrekt gespeichert wurden.

Werte: [0 | 1]

Default: 0

DeathBand (C-812)

Zulässiger Positionsfehler. Wenn der Positionsfehler (Distanz zwischen Soll- und Ist-Position) kleiner als DeathBand ist, wird nicht versucht den Motor noch genauer zu positionieren. "Regelung setzt innerhalb dieses Bereiches aus".

Nur der C-812er Controller hat diesen Parameter.

Der Parameter wird u.U. auch verwendet um den Stillstand eines Motors zu überprüfen, s. `InquireStatus`.

Werte: [1, 126]

Einheit: Encoderschritte

Default: 1

Einträge zu Software-Schranken

PositionMin

absolute Position der linken Software-Schranke

Einheit: Motorencoderschritte

Default: -100

PositionMax

absolute Position der rechten Software-Schranke

Einheit: Motorencoderschritte

Default: 100

MinimalWidth

"legt die kleinste Schrittweite fest, um ein sicheres Anfahren des Motors zu gewährleisten"

Einheit: Motorencoderschritte??

Werte: [0 .. ???]

Default: 10

MaximalWidth

Werte: [0 .. ???]

Default: 10

MaxVelocity

Maximale Geschwindigkeit des Motors

Einheit: Motorencoderschritte / sec (?)

Default: 8000

Werte: 0 .. 8.388.680 (C812); 0..1073741823 = 0x3FFFFFFF (C832) ?

Einträge für das Rückspeichern von verschiedenen Einstellungen

PositionWidth

Inkonsistente Mehrfachverwendung (FEHLER3)

wird verwendet in den Implementationen von TMotor::StartCheckScan, zur Realisierung der Funktion F6 Optimieren, ("Schrittweite zum Messen des Anfahrverhaltens")

legt die kleinste Schrittweite in der Funktion F4 Verfahren nach Encoder-Position fest (wenn dort der ScrollBar funktionieren würde)

Default: 10

Werte: > 0

AngleMin

"Nutzerbereich links"

Default: -1

AngleMax

"Nutzerbereich rechts"

Default: 1

AngleBias

"absolute relative Winkel"; Offset zwischen Absoluter und Relativer Null;
ist wohl obsolet; s.a. F1.2

AngleWidth

"Nutzerschrittweite"

Default: 0.1

Velocity

"aktuelle Geschwindigkeit"

Werte: [1, MaxVelocity]

Default: 8000

Falls der Wert 0 ist wird er automatisch auf $\text{MaxVelocity} / 2$ gesetzt.

Sonstige Parameter zur Motorsteuerung

Torque (C812)

maximaler Motorstrom

Werte: 1.. 127

Default: 110

Gain

statische Verstärkung; KD?

Werte: 1 .. 255 (C812); 1 .. 32767 (C832)

Default: 100

DynamicGain

dynamische Verstärkung; KP

Werte: 1 .. 255 (C812); 1 .. 32767 (C832)

Default: 37

IntegralGain (C-832)

KI
Werte: 1 .. 32767
Default: 10

IntegralLimit (C-832)

KL
Werte: 1 .. 32767
Default: 10

Acceleration

Beschleunigung. Wird auch als Entschleunigung verwendet. Ein Wert von 0 ist zwar zulässig aber wohl nicht wirklich sinnvoll.

Werte: 0 .. 8.388.680 (C812); 0..1073741823 = 0x3FFFFFFF (C832)
?Default: 10

DecelerationPoint (C-812ISA?)

Default: 20

TODO

Unit

"Einheit des Koordinatensystems. Die Korrekturfaktoren sind i.A. für Winkelsekunden berechnet. Der Nutzer hat bei einige Antrieben die Wahl zwischen Grad, Minuten, oder Winkelsekunden. Die Einheiten werden dann entsprechend umgerechnet. Ansonsten können beliebige Einheiten angegeben werden"

Werte: ["Grad" | "Sekunden" | "Minuten" | "Minuts" | Name]
Default: "Unit"

Gross- und Kleinschreibung wird nicht berücksichtigt.

MaxFailure

Default: 30.0

Digits

"Gibt die Anzahl der Nachkommastellen für die Anzeige der Position in verschiedenen Dialogen und Koordinatensystemen an. Für die Schrittweite wird i.A. eine weitere Nachkommastelle verwendet"

Werte: [0 ..]
Default: 2

SpeedScale

"Faktor zur Angabe der Geschwindigkeit in der verständlichen Form"

Werte: real, > 0
Einheit: "Motoreinheiten / Phys. Einheiten"
Default: 10.0

Correction

Default: 0

Koeff_1

Verhältnis zwischen Internen Positionen (in Encoderschritten) zu Absoluten Positionen (in physikalischen Einheiten, s. Unit).

Einheit: Winkel-Sekunden / Motorschritt
Default: 1.0

Koeff_2
Default: 0.0

Koeff_3
Default: 0.0

Obsolete Parameter

Diese Parameter tauchen in einigen "echten" INI-Files auf, werden aber von der Software nicht verwendet. Wahrscheinlich Überreste älterer Versionen.

MoveFirstToZero; Orientation; Direction; Position; AngleZero; Koeff_0.

Beziehungen/Bedingungen (unvollständig)

Der folgende Abschnitt enthält eine Reihe von Beziehungen bzw. Bedingungen, die zwischen den Konfigurationenparameter gelten, bzw. von diesen einzuhalten sind.

PositionMin

falls Type=C-812 oder Type=C-832 und IndexLine=0, d.h. der linke Endlagenschalter ist der Referenzpunkt

keine Positionen jenseits (kleiner) des Referenzpunktes erlaubt

DistanceToZero $\geq 0 \implies$ PositionMin \leq DistanceToZero

besser

DistanceToZero $\geq 0 \implies$ PositionMin \leq DistanceToZero - RemoveLimit

bzw.

DistanceToZero $< 0 \implies$ PositionMin \geq abs(DistanceToZero)

besser

DistanceToZero $< 0 \implies$ PositionMin \geq abs(DistanceToZero) + RemoveLimit

PositionMin, PositionMax

PositionMin $<$ PositionMax (? , was wenn Koeff_1 $<$ 0)

AngleMin

AngleMin (umgerechnet in Motorpos.) \geq PositionMin, d.h. "Nutzerbereich" sollte kleiner als der "Hardwarebereich" sein

AngleMax

AngleMax (umgerechnet in Motorpos.) \leq PositionMax, d.h. "Nutzerbereich" sollte kleiner als der "Hardwarebereich" sein

AngleMin, AngleMax

AngleMin $<$ AngleMax

InitialAngle

InitialAngle \geq AngleMin und InitialAngle \leq AngleMax, d.h. die "Grundstellung" muss im verfahrbaren Bereich liegen

AngleWidth

AngleWidth (umgerechnet in Motorpos.) \geq MaximalWidth and

AngleWidth (umgerechnet in Motorpos.) \leq MinimalWidth and

Eindeutigkeit der Hardwareansteuerung

Für jeden realen Motor darf es nur einen Motor in der Konfiguration geben d.h. die Konfigurationsparameterpaare (`BoardId`, `RamAddr`) bzw. `BoardId`, `IoAddr`) dürfen in allen Motor-Abschnitten höchstens einmal auftreten.

Eindeutigkeit der Achsenzuordnung

jede Achse darf in einer Konfiguration höchstens einmal vorkommen

4. Fehler

1. Im Dialog "Verfahren nach Encoder-Position" funktioniert der Scrollbar zur Angabe der Zielposition nicht, obwohl Teile des Codes dafür vorhanden sind. Kontext
2. im Dialog "Verfahren nach Encoder-Position": zu einem Zeitpunkt darf nur ein Motor in Bewegung sein. Kontext
3. Inkonsistente Mehrfachverwendung des Parameters `PositionWidth`
4. Fehler hafte Behandlung des `Motorspiels`

5. Änderungswünsche

1. Der "Cancel"-Button im Dialog "Move Drive" sollte während der Bewegung "Stop" heißen, und nur die Bewegung beenden, nicht aber das Fenster schließen.
2. Im Dialog der Funktion F4 sollte die Einheit der Positionsangaben erkennbar sein.
3. zu F6: Aus Softwarestrukturgünden kann die Motorenkomponente eigentlich nicht wissen das der Anwendungsrahmen die Daten des MoveScans als Diagramm darstellen will, d.h. das Event sollte nicht ausgelöst werden und der Anwendungsrahmen sollte das Fenster vor Anfordern der Funktion selbst öffnen

6. Probleme, Offene Fragen

Geklärte Fragen oder Probleme sind durchgestrichen.

1. Kalibrierung auch für alle Motoren möglich? Die beiden Optionen "Keep Position" und "Handle all Drives" werden zwar disabled wenn "Keep Calibration Data" auf false gesetzt wird, aber die Werte der Optionen werden trotzdem verwendet (denke ich). Kontext

2. Welche Funktion haben "Cancel" und "OK" bei laufendem Referenzpunktlauf und sonst? Diese Buttons sollten "Stop" und "Close" heißen. Stop sollte nur bei laufendem Referenzpunktlauf enabled sein und Close sonst. Kontext
3. aus Dokumentation: "DistanceToZero: ... Diese Größe ist für viele Antriebe durch explizites Messen einer Kalibrierungskurve festgelegt. In anderen Fällen kann die physikalische Null mit dem Referenzpunkt in Beziehung gesetzt werden." FRAGE was für Kalibrierungskurven sind gemeint? Kontext
4. Was passiert wenn die Konfigurationsparameter (für C832) sagen es gibt eine Index-Line, es aber gar keine gibt, bzw. diese nicht anspricht? Kontext
5. ~~Was passiert wenn MoveFirstToLimit == false und der Referenzpunktlauf links vom Referenzpunkt beginnt, also der Referenzpunkt nicht in Fahrtrichtung gefunden werden kann? Kontext s. Testpaket m_rpl~~
6. Warum macht das Setzen der Absoluten Null die Kalibrierung bzw. den Referenzpunktlauf ungültig? Kontext
7. Wann wird die Funktion "Setzen der Absoluten Null" verwendet? Kontext
8. Warum sind im Dialog "Common Settings ..." die Steuerelemente für PositionMin, PositionMax und RemoveLimit disabled? Kontext
9. Frage zum Verhältnis Position[Min|Max|...] zu Angle[Min|Max|...]. Kontext
10. Wann wird die Funktion "F3.2 End-Schalter" benötigt?
11. Unter welchen Umständen wird die Funktion "Verfahren nach Encoder-Schritten" verwendet?
12. ~~Welche Funktion hat der Konfigurations-Parameter InitialMove?~~
13. Welche Bedeutung haben die Achsen Absorber, Monochromator, Encoder, Theta? s. F7.2

7. Verwandte Dokumente

1. Beschreibung der C-Schnittstelle für die Motorenkomponente

8. Änderungen

Version	Änderung / Anmerkung
0.8	Abschnitt "Beziehungen/Bedingungen" eingefügt kleinere Korrekturen/Eränzungen
0.7.1	kleinere Korrekturen/Eränzungen HTML/Format-Anpassungen für TeXifizierung
0.7	Angaben zu Konf.-Parametern ergänzt/korrigiert Definitionen erweitert (insb. zur genaueren Unterscheidung von Motor und Achse) Funktionen F7 eingefügt Funktion F6.3 ausgeführt
0.6	Änderung der Struktur entsprechend Vorgaben für Verhaltensspezifikation; Korrekturen nach Review durch Herrn Sacklowski
0.5	erste öffentliche Fassung

9. Quellen

1. Dokumentationsfragmente der Programmierers
2. Physik Instrumente (PI) GmbH, Waldbronn, DE: *C-812 DC-Motor Controller. Operating Manual, Programming Reference Product Documentation*, Dec. 1993.
3. Physik Instrumente (PI) GmbH, Waldbronn, DE: *C-832 DC-Motor Controller. Programming Manual*, April 1993.

10. Glossar

Absolute Motorposition

(Physikalische) Stellung des vom Motor bewegten Schlitten/Tisch/... . Das Koordinatensystem zur Angabe absoluter Motorpositionen hat seinen Ursprung in der Absoluten Null.

Es ist Aufgabe der Software aus der internen Motorposition die absolute Motorposition zu bestimmen.

Die Einheit der vom Motorcontroller gelieferten internen Position sind Motorencoderschritte. Diese müssen je nach Achse in physikalische Einheiten umgerechnet werden. (z.B. Sekunden, Grad, Zentimeter)

Absolute Null (Syn.: Physikalische Null)

Absolute Motorposition, die per definitionem dem Ursprung des verwendeten Koordinatensystems der Absoluten Positionen entspricht. Diese Position wird während der Einrichtung des Motors festgelegt.

Z.B. könnte man festlegen, dass die Absolute Null der Tilt-Achse gerade dort ist, wo der Tisch waagrecht ist.

Achse

1. Ein Freiheitsgrad in einem bestimmten Versuchsaufbau.
2. Rolle die ein Motor in einem bestimmten Versuchsaufbau einnimmt, in dem über ihn die Bewegung entlang einer Achse (1) erfolgt.

s.a. F7.2

Aktueller Motor, Aktiver Motor

Ausgezeichneter Motor, der für einige Funktionen eine Voreinstellung und für einige Funktionen der Motor ist, auf dem die Funktion arbeitet. Einige Funktionen können den Aktuellen Motor bestimmen. s.a. F7.1

Controller (Syn.: Steuerkarte)

Hardware. Schnittstelle zur Geräten der Peripherie (Motoren, Detektoren). Die Software steuert ein Gerät stets indirekt über einen Controller.

Hardware-Schranke (Endlage, Endlagenschalter)

Absolute Motorposition die die Bewegung eines Motors in einer Richtung beschränkt. Erreicht der Motor diese Position löst die Hardware - z.B. über einen Endlagenschalter - ein Signal aus, auf das der Motorcontroller bzw. die Steuersoftware reagiert.

Bei dem Controller C-812 wird der Motor automatisch um eine konfigurierbare Anzahl von Motorschritten zurückgefahren um diese kritische Position zu verlassen und ein Status-Flag gesetzt.

Beim C-832 wird nur ein Status-Flag gesetzt, das von der Software abzufragen und zu behandeln ist.

s.a. ini-Eintrag `RemoveLimit`.

Interne Motorposition

Vom Motorcontroller bestimmte Position, relativ zu einer beliebig gesetzten Null-Position. Ein Motorcontroller kann nur Positionsänderungen registrieren und mit Hilfe dieser die interne Motorposition fortschreiben. Die Einheit der Internen Motorposition ist der Encoderschritt/Encoderimpuls. Nach dem Einschalten der Hardware ist diese interne Position undefiniert oder Null. Sie kann zu jedem Zeitpunkt per Befehl auf Null gesetzt werden. Die Position an der die interne Position gleich null ist, wird Home-Position oder auch Interne Nullgenannt.

Kalibrierung

Allgemein: Abstimmung zwischen Soft- und Hardware. Die Software soll dem konkreten Zustand der Hardware entsprechend konfiguriert werden. Konkret soll das Verhältnis zwischen den Koordinatensystemen der Internen und der Absoluten Motorpositionen bestimmt werden um die Umrechnung ineinander zu ermöglichen.

links - rechts

Eine Position ist links von einer anderen wenn ihr Wert kleiner ist, bzw. rechts wenn ihr Wert größer ist. (s.a. Abb. 1)

Motor (Syn.: Antrieb)

...

Motorencoderschritte

Einheit in der ein Motorcontroller Distanzen und Positionen angibt. Die Umrechnung in physikalische Einheiten ist abhängig von der Auflösung des verwendeten Motors (in Encoderimpulsen pro Umdrehung) und der Übersetzung des verwendeten Getriebes.

Motorspiel (Syn.: Hysteresis)

Differenz zwischen zwei absoluten Positionen um die der Schlitten ohne eine Drehung der Motorschraube bewegt werden kann

bzw. umgekehrt

Winkel um den die Motorschraube (nach Richtungswechsel) gedreht werden kann, ohne dass sich der Schlitten bewegt.

Das Motorspiel muss experimentell bestimmt werden. s. ini-Eintrag Hysteresis

Referenzpunkt (Syn.: Index-Position, Grundstellung)

Position die der Motor reproduzierbar anfahren kann, unabhängig davon, an welcher Ausgangsposition er sich befindet (d.h. die hardwareseitig erkennbar ist). Die Lage dieses Punktes im Koordinatensystem der absoluten Motorposition ist durch die Kalibrierung bekannt.

z.B. wird beim C-812-Controller als R. die Position verwendet, auf die der Motor automatisch gefahren wird, wenn die linke Hardware-Schranke erreicht wird.

Relative Motorposition

Absolute Motorposition in einem alternativen Koordinatensystem, das sich von dem Koordinatensystem der Absoluten Positionen nur durch eine Verschiebung (Offset) des Nullpunktes unterscheidet. Sie dient der temporären Koordinatenverschiebung.

Relative Null

Koordinatenursprung der Relativen Motorpositionen. Bestimmt durch ein Offset zur Absoluten Null. (s.a. ini-Eintrag `AngleBias`)

Software-Schranke

Parameter der die Werte eines anderen Parameters in einer Richtung begrenzt.