

Virtuelle Instrumente für MCC-Steuerungen



LabVIEW[®] Virtuelle Instrumente
für MCC-Steuerungen

ORIGINAL BETRIEBSANLEITUNG

© 2010

Alle Rechte bei:

Phytron GmbH

Industriestraße 12

82194 Gröbenzell, Deutschland

Tel.: +49(0)8142/503-0

Fax: +49(0)8142/503-190

Alle Angaben in diesem Handbuch erfolgen nach bestem Wissen, aber ohne Gewähr. Wir behalten uns im Interesse unserer Kunden vor, Verbesserungen und Berichtigungen an Hardware, Software und Dokumentation jeder Zeit ohne Ankündigung vorzunehmen. Für Anregungen und Kritik sind wir dankbar. E-Mail-Adresse: doku@phytron.de

Den neuesten Stand des Handbuchs finden Sie im Internet unter www.phytron.de.

Inhaltsverzeichnis

1	Einführung.....	6
1.1	Voraussetzungen.....	6
1.2	Lieferumfang.....	6
2	Vor der Installation zu beachten.....	7
2.1	Qualifiziertes Personal.....	7
2.2	Warnhinweise für den Einsatz von Software-Produkten.....	7
3	Allgemeine VI Beschreibung	8
4	Beschreibung der MCC.IIb	9
4.1	Init-MCC.vi.....	9
4.2	AD-MCC.VI.....	9
4.3	COMM-MCC.vi	10
4.4	Counter-MCC.vi.....	10
4.5	Directmode-MCC.vi	11
4.6	Drive-MCC.vi	12
4.7	Encoder-MCC.vi	13
4.8	Init-MCC.vi.....	14
4.9	Input-MCC.vi.....	15
4.10	Output-MCC.vi.....	16
4.11	Parameter-MCC.vi.....	17
4.12	Register-MCC.vi	18
4.13	Status-MCC.vi.....	19
5	Demo-MCC.vi.....	20
5.1	Allgemeine Beschreibung.....	20
5.2	A/D-Eingänge und Encoder-Zähler einlesen	21
5.3	Applikation für Direktbetrieb und Motor fahren	22
5.4	Internen Streckenzählers lesen und ausgeben	24
5.5	Ein / Ausgänge der MCC lesen / setzen.....	25
5.6	Parameter lesen / schreiben.....	26
5.7	Register lesen und schreiben	27
5.8	Initiator- und Steuerungs-Status lesen	28
6	Parameter.....	29
7	Copyright und Haftungsausschluss.....	35
8	Stichwortverzeichnis.....	36

1 Einführung

LabVIEW von National Instruments ist eine graphische Software-Entwicklungsumgebung, die auf einer symbolischen Darstellung beruht.

LabVIEW steht für „Laboratory Virtual Instrument Engineering Workbench“.

LabVIEW Programme werden als Virtuelle Instrumente oder einfach VIs bezeichnet. Sie bestehen aus dem Frontpanel, der Benutzerschnittstelle, und dem Blockdiagramm, dem graphischen Programmcode, der ähnlich anderer Hochsprachen kompiliert wird.

Dieses Manual beschreibt die LabVIEW Nutzung für die Phytron Steuerungen MCC-1, MCC-2 und MCC-2 LIN.

LabVIEW ist ein eingetragenes Warenzeichen von National Instruments Corporation. Siehe dazu Kap. 7.

1.1 Voraussetzungen

Um diese VIs für die MCC Steuerungen der Firma Phytron anwenden zu können, wird davon ausgegangen, dass der Anwender LabVIEW Erfahrung besitzt und ihm die Programmierumgebung vertraut ist. Grundlegende Kenntnisse in der Programmierung wie Datentypen, Schleifen usw. werden ebenfalls vorausgesetzt.

Die Systemvoraussetzung für die MCC VIs ist LabVIEW 8.0 oder aktueller.

1.2 Lieferumfang

Das Paket enthält zwei LabVIEW Bibliotheken:

• MCC.llb	enthält die zu verwendenden VIs für die MCC-Steuerungen
• Demo-MCC.llb	verdeutlicht durch Demo-Applikationen die VIs

2 Vor der Installation zu beachten



Lesen Sie vor Einbau und Inbetriebnahme alle Manuals gründlich durch. Beachten Sie die Sicherheitshinweise im folgenden Kapitel.

2.1 Qualifiziertes Personal

Projektierung, Inbetriebnahme und Wartung darf nur durch entsprechend geschultes Fachpersonal erfolgen.

Dieses Personal muss in der Lage sein, Gefahren zu erkennen, die durch mechanische, elektrische oder elektronische Geräte und Ausrüstungen verursacht werden können.

Das Fachpersonal muss den Inhalt dieses Manuals und alle zum Produkt gehörigen Unterlagen kennen und verstehen können. Sicherheitsunterweisungen sind vorzusehen.

Den Fachkräften müssen alle geltenden Normen, Bestimmungen und Unfallverhütungsvorschriften, die bei Arbeiten am und mit dem Produkt beachtet werden müssen, bekannt sein.

WARNUNG



Durch nicht ausreichend qualifiziertes Personal können schwere Schäden an Maschine und Antrieben, oder sogar Personenschäden verursacht werden!

2.2 Warnhinweise für den Einsatz von Software-Produkten



1. Bei jeder Anwendung kann die Funktionszuverlässigkeit von Software-Produkten durch entgegenwirkende Faktoren wie z. B. Spannungsunterschiede oder Hardwarefehler etc. beeinträchtigt werden. Um Schäden durch Systemfehler vorzubeugen, sollte der Nutzer angemessene Sicherheitsmaßnahmen ergreifen. Hierzu gehören unter anderem Sicherungs- und Abschaltmechanismen.



2. Beim Programmieren der Befehlscodes kann es zu Fehlfunktionen – z.B. Loslaufen des angeschlossenen Motors, Abbremsen,... – kommen. Deshalb den Programmablauf schrittweise testen!



3. Da jedes Endnutzersystem den Kundenbedürfnissen angepasst ist und sich vom Testumfeld unterscheidet, ist der Nutzer oder Anwendungsentwickler für die Eignung für diese Anwendung verantwortlich.

3 Allgemeine VI Beschreibung

In LabVIEW bezeichnet man Funktionsblöcke als Virtuelle Instrumente (VIs). Jedes Programm ist sowohl eigenständig lauffähig als auch als Unterprogramm anwendbar.

Der Anwender definiert den Datenfluss, indem er die VIs mit Verbindungslinien (Drähten) versieht.

Die Ausführung eines VIs beginnt, wenn alle Eingangsdaten vorhanden sind.

Die Ergebnisse liegen erst dann an den Ausgängen an, wenn das gesamte VI abgearbeitet ist. Die Reihenfolge der Abarbeitung wird durch die Abhängigkeit der Daten definiert. Es gibt keine vordefinierte Reihenfolge (z. B. von rechts nach links).

4 Beschreibung der MCC.IIb

4.1 Init-MCC.vi

Es gibt Eingänge und Ausgänge, die für alle VIs in der Bibliothek gleich sind:

Bezeichnung	I/O	Beschreibung
VISA resource name in	Eingang	Übergabe der Schnittstellenparameter
Error in	Eingang	Eingang des Fehlerclusters
VISA resource name out	Ausgang	Ausgabe der Schnittstellenparameter
Error out	Ausgang	Ausgang des Fehlerclusters

Diese Ein- bzw. Ausgänge sind immer mit der gleichen Funktion versehen, so dass sie hier nur einmal beschrieben wird.

Eine Bündelung unterschiedlicher Datentypen in LabVIEW nennt man Cluster. Es kann als Eingang oder Ausgang verwendet werden.

4.2 AD-MCC.VI

Der A/D Wert der MCC wird eingelesen.

Bei Ausführung dieses VIs wird der aktuelle A/D Wandler Wert als 16-Bit-Einheit am Ausgang zur Verfügung gestellt.

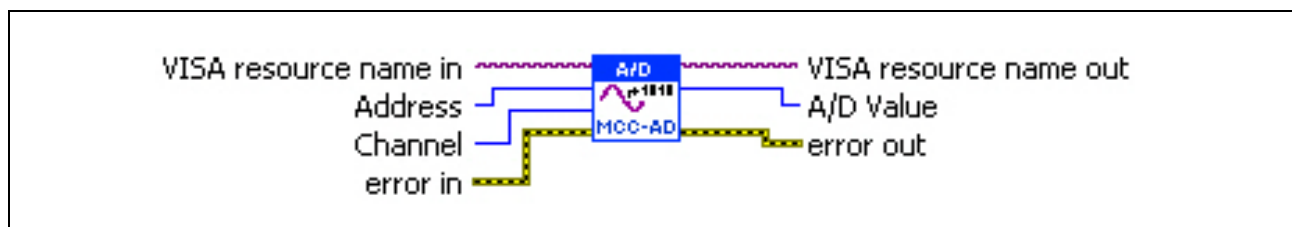


Abb. 1: AD-MCC.vi

Bezeichnung	I/O	Beschreibung
Address	Eingang	An der Steuerung eingestellte Adresse (0 -15 8-Bit Einheit)
Channel	Eingang	Kanal des A/D Wandlers, der gelesen werden soll (1 oder 2, 8-Bit Einheit)
A/D Value	Ausgang	A/D Wert in Inkrementen (0 – 1023, 10-Bit)

4.3 COMM-MCC.vi

Dieses VI wird von anderen VIs intern verwendet und sollte nicht zur Programmierung eigener Anwendungen benutzt werden.

4.4 Counter-MCC.vi

Dieses VI liest den Zählerstand der ausgewählten Achse.

Bei der Ausführung wird der Zählerwert der ausgewählten Achse ausgegeben. Dabei wird Parameter 20 (P20) der jeweiligen Achse gelesen.

Die Beschreibung der Parameter befindet sich im Kapitel 6.

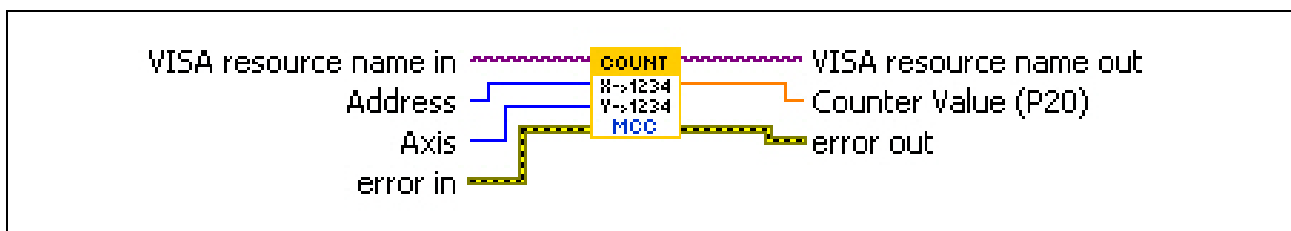


Abb. 2: Counter-MCC.vi

Bezeichnung	I/O	Beschreibung
Address	Eingang	An der Steuerung eingestellte Adresse (0 -15 8-Bit Einheit)
Axis	Eingang	Achse, deren Zählerstand gelesen werden soll (1 oder 2, 8-Bit Einheit)
Counter Value	Ausgang	Gelesener Zählerwert der Achse (Double)

4.5 Directmode-MCC.vi

Ein Befehl wird an die Steuerung geschickt.

Die VI überträgt den String am Eingang Send String an die Steuerung und holt die Antwort von der Steuerung ab.

Die Beschreibung der Steuerungsbefehle befindet sich im Kapitel 6.

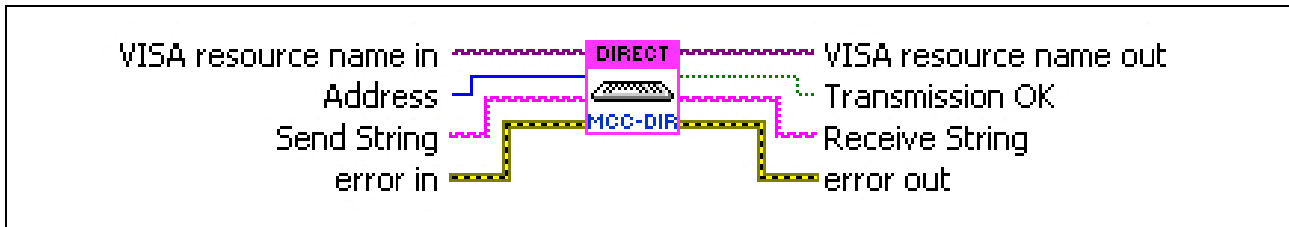


Abb. 3: Directmode-MCC.vi

Bezeichnung	I/O	Beschreibung
Address	Eingang	An der Steuerung eingestellte Adresse (0 -15 8- Einheit)
Send String	Eingang	Befehl, der an die Steuerung gesendet wird (z. B. X+1000 für 1000 Schritte fahren)
Transmission OK	Ausgang	True, wenn die Steuerung den Befehl verstanden hat (ACK) False, wenn der Befehl ungültig war (NAK)
Receive String	Ausgang	Antwort String der Steuerung (ohne Steuerzeichen und ACK) Bei Befehlen ohne Antwort leer

4.6 Drive-MCC.vi

Diese VI sendet Fahrbefehle an die MCC.

Dieses Funktionsmodul liest die im Cluster eingestellten Werte ein und generiert daraus für die MCC einen Fahrbefehl.

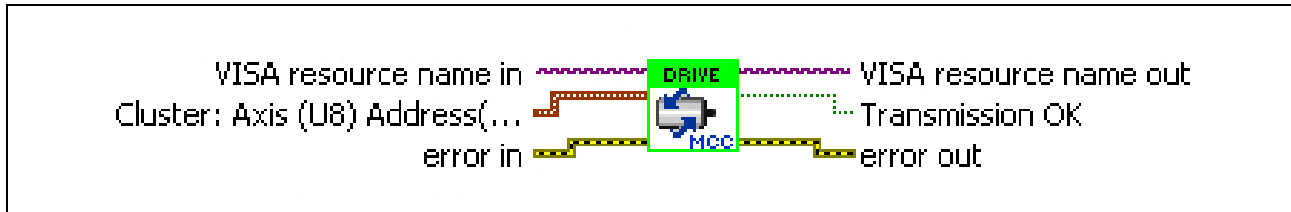


Abb. 4: Drive-MCC.vi

Name	I/O	Beschreibung
Cluster: Axis Address Position Mode Distance	Eingang	<p>Besteht aus folgenden Dateitypen:</p> <ul style="list-style-type: none"> • Axis (8-Bit Einheit): Achse, an die der Fahrbefehl ausgegeben wird (1 oder 2) • Address (8-Bit Einheit): Steuerungsadresse (0-15) • Position Mode (ENUM): Folgende Einstellungen sind verfügbar: <ul style="list-style-type: none"> ○ Relative, ein relativer Fahrbefehl wird erstellt und übertragen ○ Absolute, ein absoluter Fahrbefehl wird erstellt und übertragen ○ Initialisation Plus, eine Referenzfahrt in positive Richtung wird erstellt und übertragen ○ Initialisation Minus, eine Referenzfahrt in negative Richtung wird erstellt und übertragen ○ Free Run Plus, ein freier Lauf in die positive Richtung wird gestartet ○ Free Run Minus, ein freier Lauf in die negative Richtung wird gestartet • Distance (DLB): Für die Fahrbefehle relative und absolut wird diese Eingabe als Strecke verwendet
Transmission OK	Ausgang	<p>True: die Steuerung akzeptiert den Befehl False: Befehl ungültig</p>

4.7 Encoder-MCC.vi

Der Encoderzähler liest die ausgewählte Achse.
 Parameter 22 (P22) wird für die jeweilige Achse ausgelesen.
 Die Beschreibung der Parameter befindet sich unter Kapitel 6).

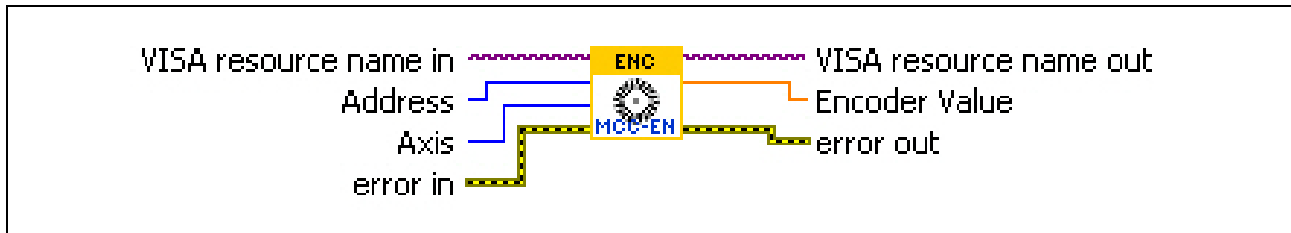


Abb. 5: Encoder-MCC.vi

Name	I/O	Beschreibung
Address	Eingang	An der Steuerung eingestellte Adresse (0 -15 8-Bit Einheit)
Axis	Eingang	Achse, deren Zählerstand gelesen werden soll (1 oder 2, 8-Bit Einheit)
Encoder Value	Ausgang	Gelesener Encoder-Zählerwert der Achse (Double)

4.8 Init-MCC.vi

Der Zustand der Initiatoren wird ausgegeben.

Der Initiator Status der MCC wird eingelesen und als BOOL-Cluster ausgegeben.

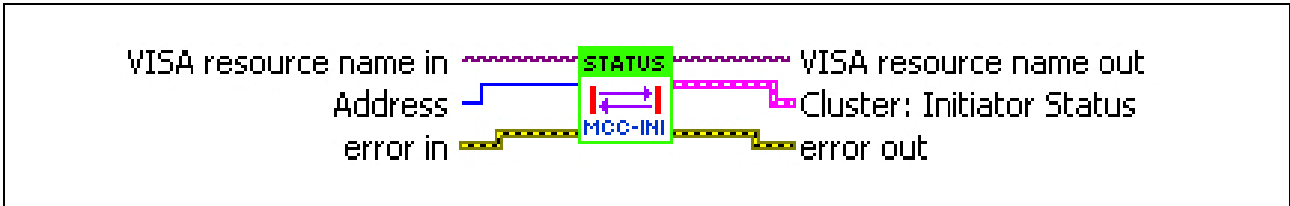


Abb. 6: Init-MCC.vi

Name	I/O	Beschreibung
Address	Eingang	An der Steuerung eingestellte Adresse (0 -15 8-Bit Einheit)
Cluster: Initiator Status	Ausgang	Initiator Status besteht aus vier Elementen (BOOL) <ul style="list-style-type: none"> • Axis 1+, angesprochen = TRUE, frei = FALSE • Axis 1-, angesprochen = TRUE, frei = FALSE • Axis 2+, angesprochen = TRUE, frei = FALSE • Axis 2 -, angesprochen = TRUE, frei = FALSE

4.9 Input-MCC.vi

Der Eingangszustand der MCC wird eingelesen.
Dieser Zustand der MCC wird als BOOL-Cluster ausgegeben.

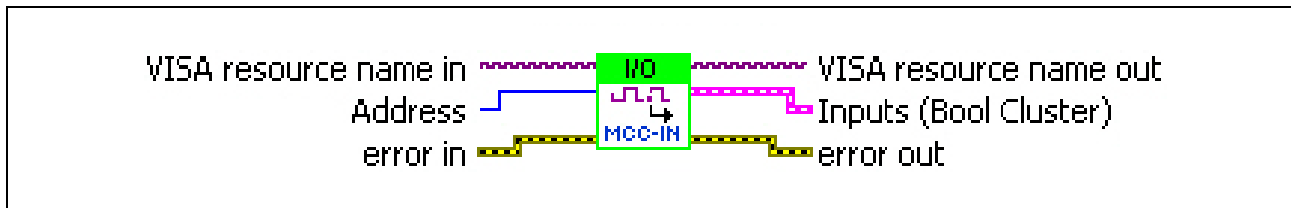


Abb. 1: Input-MCC.vi

Name	I/O	Beschreibung
Address	Eingang	An der Steuerung eingestellte Adresse (0 -15 8-Bit Einheit)
Cluster: Inputs	Ausgang	Inputs, besteht aus acht Elementen (BOOL) TRUE = Eingang High FALSE = Eingang Low

4.10 Output-MCC.vi

Diese VI setzt die Ausgänge an der MCC.

Bei der Ausführung werden die am Eingang anliegenden Zustände als Ausgangszustände gesetzt.

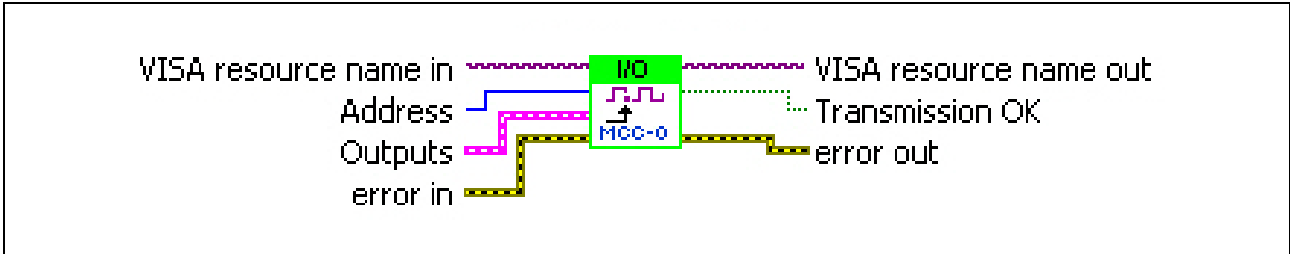


Abb. 2: Output-MCC.vi

Name	I/O	Beschreibung
Address	Eingang	An der Steuerung eingestellte Adresse (0 -15 8-Bit Einheit)
Cluster: Outputs	Eingang	Outputs, besteht aus acht Elementen (BOOL) TRUE = Ausgang High FALSE = Ausgang Low
Transmission OK	Ausgang	True: die Steuerung akzeptiert den Befehl False: Befehl ungültig

4.11 Parameter-MCC.vi

Mit diesem VI setzt oder liest man die Parameter der MCC.

Der in Parameter Number eingestellte Parameter wird von der MCC gelesen oder an die MCC übertragen.

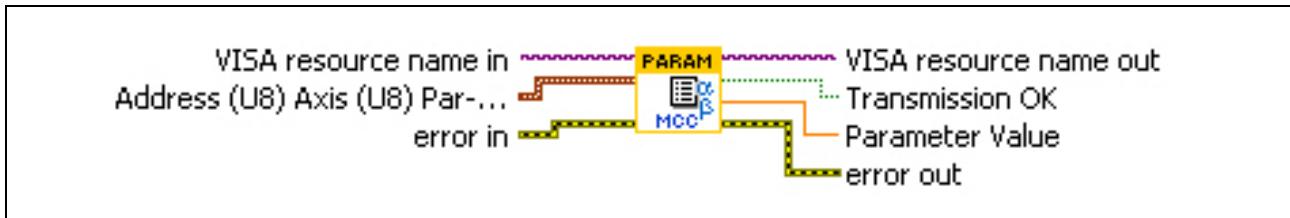


Abb. 3: Parameter-MCC.vi

Name	I/O	Beschreibung
Cluster: Address Axis Parameter Number Parameter Value Read / Write	Eingang	Besteht aus folgenden Datentypen: <ul style="list-style-type: none"> • Address (8-Bit Einheit): Steuerungsadresse (0-15) • Axis (8-Bit Einheit): Achse, deren Parameter gelesen/geschrieben wird (1 oder 2) • Parameter Number (8-Bit Einheit): Parameternummer, die gelesen oder geschrieben werden soll • Parameter Value (Double): Parameterwert, der geschrieben werden soll. Nur bei Auswahl ‚write‘! • Read / Write (ENUM): Enthält die Einträge Read und Write. Read: Parameter wird gelesen und am Ausgang Parameter Value ausgegeben Write: Parameter wird mit dem Wert aus dem Eingang Parameter Value beschrieben.
Transmission OK	Ausgang	True: die Steuerung akzeptiert den Befehl False: Befehl ungültig (BOOL)
Parameter Value	Ausgang	Bei Funktion Read wird der Parameterwert des ausgewählten Parameters ausgegeben (Double).

4.12 Register-MCC.vi

Die Register der MCC werden gesetzt oder gelesen.

Das in Register Number eingestellte Register wird von der MCC gelesen oder an diese übertragen.

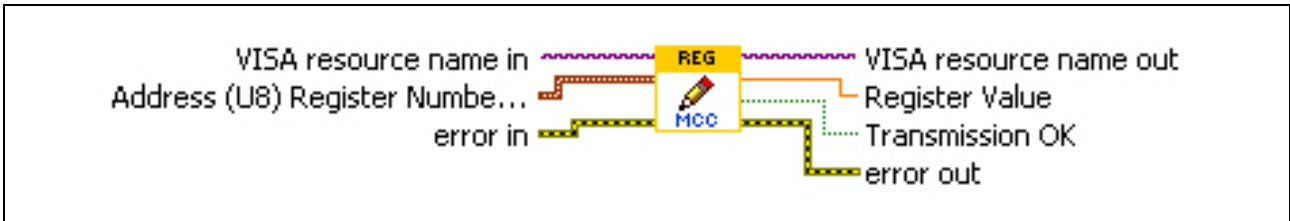


Abb. 4: Register-MCC.vi

Name	I/O	Beschreibung
Cluster: Address Register Number Register Value Read / Write	Eingang	Besteht aus folgenden Datentypen: <ul style="list-style-type: none"> Address (8-Bit Einheit): Steuerungsadresse (0-15) Register Number (16-Bit Einheit): Registernummer, die gelesen oder geschrieben werden soll. Register Value (Double): Registerwert, der geschrieben werden soll. Nur bei Auswahl ‚write‘! Read / Write (ENUM): Enthält die Einträge Read und Write. Read: Parameter wird gelesen und am Ausgang Register Value ausgegeben Write: Parameter wird mit dem Wert aus dem Eingang Register Value beschrieben.
Transmission OK	Ausgang	True: Steuerung akzeptiert den Befehl False: Befehl ungültig (BOOL)
Register Value	Ausgang	Bei Funktion Read wird der Registerwert des ausgewählten Parameters ausgegeben (Double).

4.13 Status-MCC.vi

Der Status der MCC wird gelesen und als BOOL-Cluster ausgegeben.

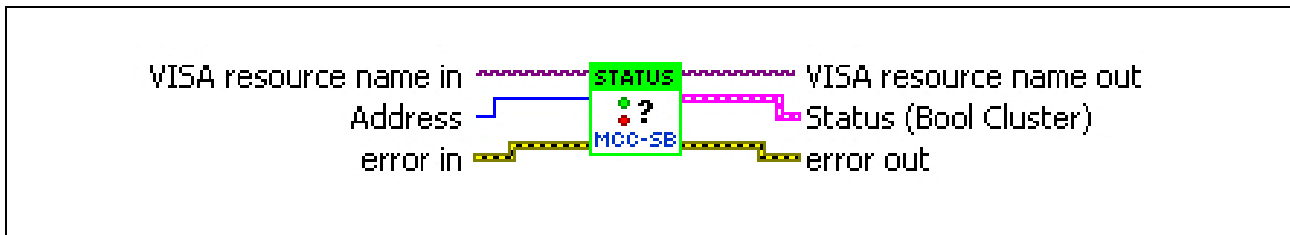


Abb. 5: Status-MCC.vi

Name	I/O	Beschreibung
Address	Eingang	An der Steuerung eingestellte Adresse (0-15, 8-Bit Einheit)
Cluster: Status	Ausgang	Gibt den Status der MCC als BOOL-Cluster aus. Der Status wird binär gelesen. Weitere Informationen sind im MINILOG Programmier-Manual für MCC unter dem Befehl SB zu finden.

5 Demo-MCC.vi

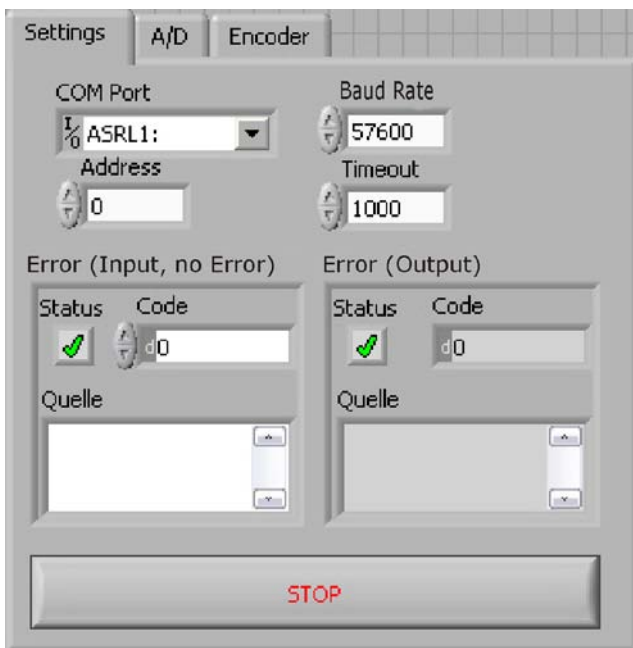
5.1 Allgemeine Beschreibung

In den Demo VIs werden die VIs aus der MCC.lib verwendet. Diese Demos bestehen aus Registerkarten mit verschiedenen Inhalten.

Die erste Registerkarte (Settings) ist bei allen Demos gleich und wird deshalb hier nur einmal beschrieben.

Registerkarte Settings:

Hier werden allgemeine Einstellungen zur Schnittstelle vorgenommen:

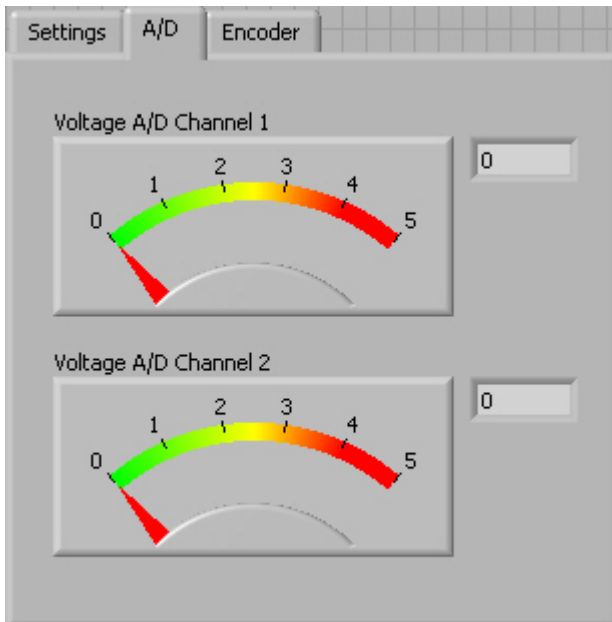


COM Port	Stellt die verwendete Schnittstelle ein. Hier die serielle Schnittstelle COM4.
Address	An der Steuerung eingestellte Adresse
Baud Rate	Baud Rate der Steuerung wird eingestellt, z. B. 57 600.
Timeout	Zeit, so lange auf eine Antwort gewartet wird. Kommt keine Antwort in der angegebenen Zeit, löst das VISA-VI einen Fehler aus.
Fehler	Dieser Ausgang gibt Fehlermeldungen aus, die während der Kommunikation auftreten können.
Stop	Beendet das Programm.

5.2 DEMO
A/D-ENC
MCC **A/D-Eingänge und Encoder-Zähler einlesen**

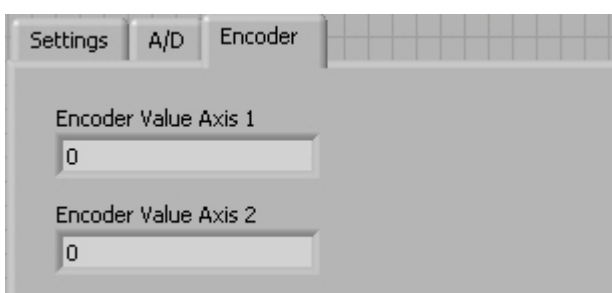
Diese Demo besteht aus einer Registerkarte mit drei Registern, Settings, A/D und Encoder. (Beschreibung Registerkarte Settings siehe oben.)

Registerkarte A/D:



Hier werden die Spannungen, die am Kanal 1 und 2 der A/D Eingänge der MCC anliegen, angezeigt. Diese werden sowohl graphisch dargestellt als auch als Zahl im Textfeld ausgegeben.

Registerkarte Encoder:



Die Encoder Zähler der MCC werden eingelesen und der Zählerwert wird in den Textfeldern angezeigt.

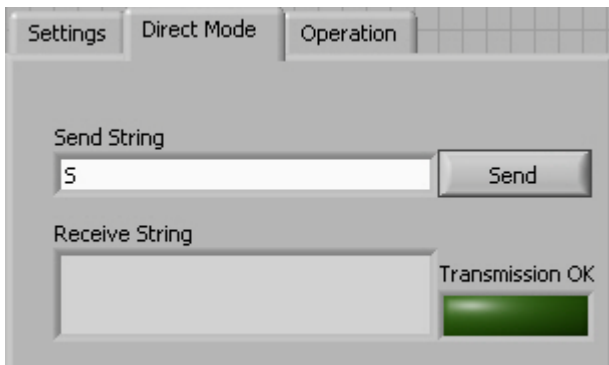


5.3 Applikation für Direktbetrieb und Motor fahren

Diese Demo ist eine kleine Applikation zur Verdeutlichung der

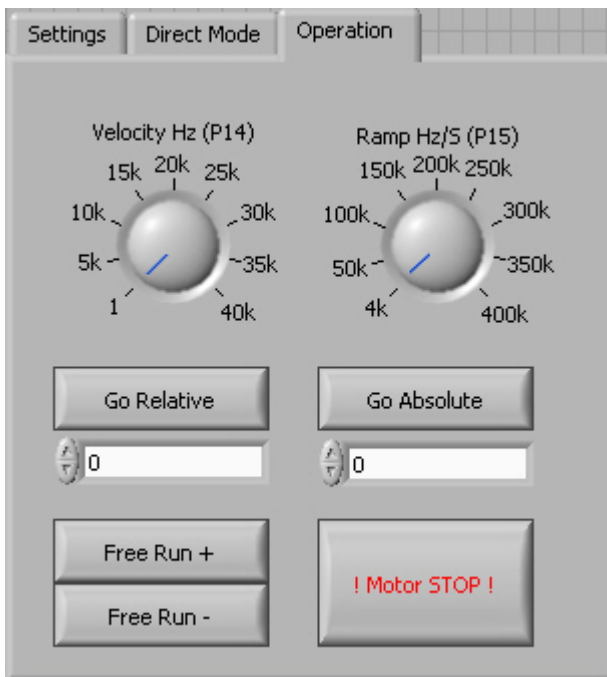
- **Directmode-MCC.vi**
- **Parameter-MCC.vi** und
- **Drive-MCC.vi** Dateien.

Registerkarte Directmode:



Send String	Eingabe des zu übertragenden Befehls
Send	Übertragen des Befehls
Receive String	Anzeigen der MCC-Antwort
Transmission OK	Befehl verstanden: LED an (ACK) Befehl nicht verstanden: LED aus (NAK)

Registerkarte Operation:



- Velocity Hz (P14) Einstellen der Fahrgeschwindigkeit der MCC. Mit Free Run + oder – lässt sich die Geschwindigkeit auch während der Fahrt ändern.

- Ramp Hz/S (P15) Einstellen der Beschleunigungs- und Achsenrampe. Der Wert wird nur bei stehendem Motor übernommen.

- Go Relative Mit der im Feld darunter eingegebenen Strecke wird relativ verfahren.

- Go Absolute Mit der im Feld darunter liegenden Strecke wird absolut verfahren.

- Free Run + Startet freien Lauf in positive Richtung

- Free Run - Startet freien Lauf in negative Richtung

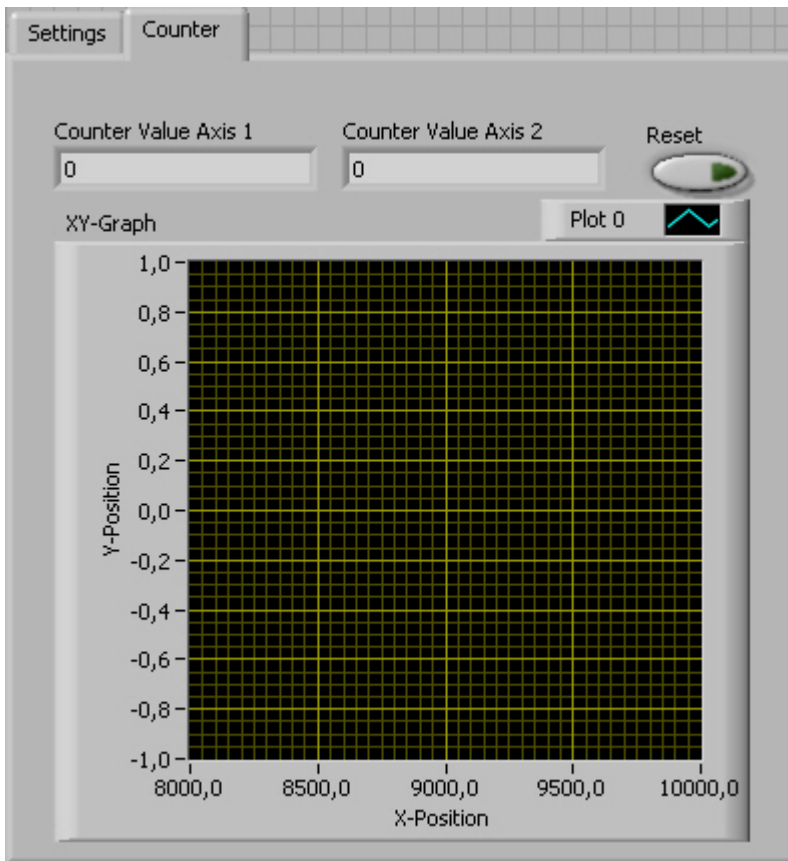
- ! Motor STOP ! Jede laufende Positionierung wird abgebrochen und der Motor gestoppt.



5.4 Internen Streckenzählers lesen und ausgeben

Diese Demo liest den internen Streckenzähler der MCC (P20) und gibt ihn sowohl als Zählerwert als auch als graphischen Plot aus.

Registerkarte Counter:



Counter Value Axis1 Anzeige des Zählerwerts von Achse 1 (X)

Counter Value Axis2 Anzeige des Zählerwerts von Achse 2 (Y)

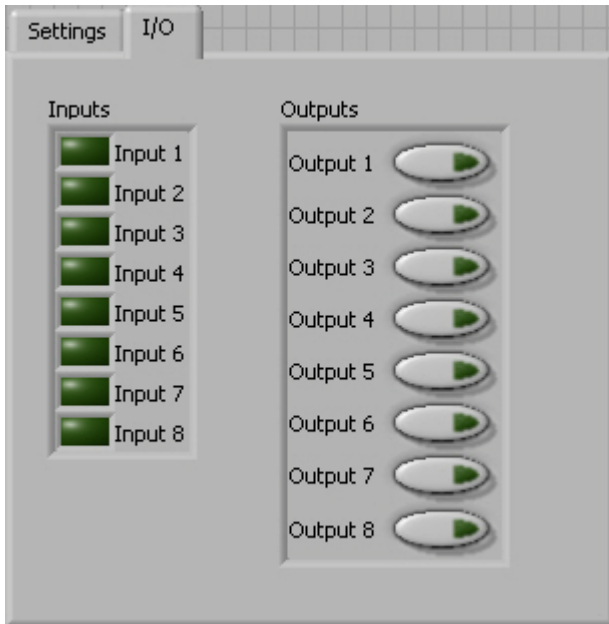
Reset Löscht den Graphen

XY Graph Graphische Darstellung der sich ändernden Zählerwerte an Achse 1 und in einem Koordinatensystem.

5.5  **Ein / Ausgänge der MCC lesen / setzen**

Diese Demo liest die Eingänge der MCC ein, zeigt sie an und schaltet die Ausgänge.

Registerkarte I/O:



Inputs Anzeige des Eingangszustandes der MCC:
 LED an = Zustand High
 LED aus = Zustand Low

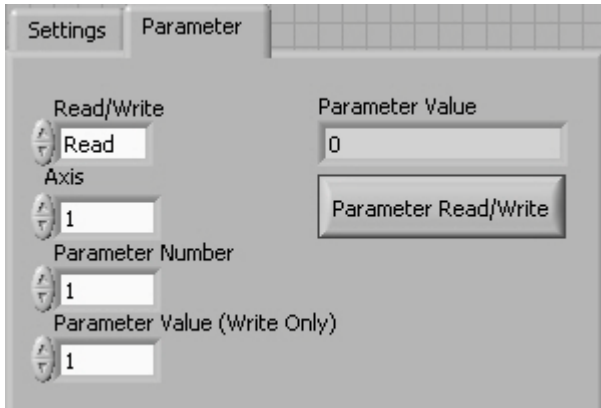
Outputs Die jeweiligen Ausgänge der MCC werden geschaltet.



5.6 Parameter lesen / schreiben

Diese Demo liest und schreibt die Parameter der MCC.

Registerkarte Parameter:

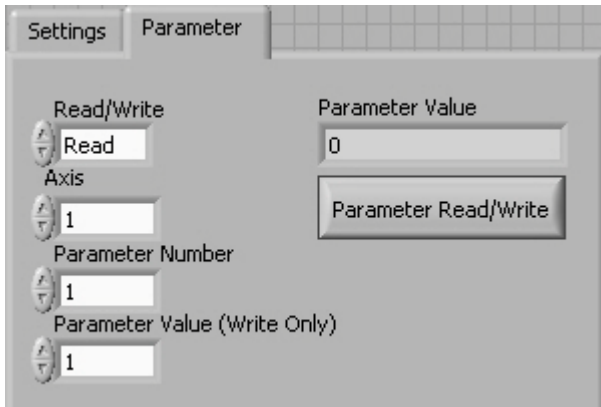


Read/Write	Parameter lesen oder schreiben
Axis	Achse, deren Parameter geändert werden
Parameter Number	Parameternummer, die geändert wird
Parameter Value (Write Only)	Wert, auf den der Parameter geändert wird (nur schreiben)
Parameter Value	Gelesener Parameter von der Steuerung (nur bei lesen)

5.7  **Register lesen und schreiben**

Diese Demo liest und schreibt die Register der MCC.

Registerkarte Read/Write Registers:



Read/Write	Register lesen oder schreiben
Register Number	Nummer des Registers, das gelesen oder geschrieben wird
Register Value (Write Only)	Wert, der in das Register geschrieben wird. (nur schreiben)
Register Value	Wert, der aus dem Register gelesen wird. (nur lesen)



5.8 Initiator- und Steuerungs-Status lesen

Diese Demo liest den Initiator Status und den allgemeinen Status der MCC und zeigt diesen an.

Registerkarte Status:



Initiator Status Initiator Status der Steuerung.
LED leuchtet bei Aktivierung des entsprechenden Initiators.
Hier wird der Befehl SUI verwendet.

Status Allgemeiner Status der Steuerung.
LEDs zeigen den Status an.
Hier wird der Befehl SB verwendet.

Die Farbzustände der LEDs sind im MiniLog-Programmiermanual beschrieben.

6 Parameter

Zum Betrieb der Steuerung werden verschiedene Voreinstellungen wie Frequenzen, Beschleunigungsrampen oder Wartezeiten benötigt, die als **Parameter** bezeichnet werden.

Bei Auslieferung sind Grundparameter hinterlegt, mit denen die Steuerung in vielen Anwendungen betrieben werden kann. Diese Parameter können in LabVIEW mit Parameter-VI oder in MiniLog-Comm ausgelesen und editiert werden.

Zu den Parametern gehören auch Zähler, die vom Programm fortlaufend aktualisiert werden. Es ist möglich, die Zähler auszulesen und z.T. auch zu editieren.


- Die Parameter gibt man für jede Achse separat ein. Zur Kennzeichnung der Achse muss vor der Parameternummer ein X bzw. Y eingefügt werden (1 oder 2 auch möglich).
- Beispiel: XP15 ist die Beschleunigungsrampe für die X-Achse.
- Auch innerhalb des Programms ist es möglich, Parameter zu ändern.
- Parameter können entweder beschrieben oder ausgelesen werden.
- P48 und P49 können nur ausgelesen werden.
- P19 bis P22 sind Zähler, die vom Programm bei Verfahren der Achsen laufend aktualisiert werden.
- P27 bis P49 sind spezielle Parameter für MCC-2.

Parameterliste

Nr.	Bedeutung	Auslieferungszustand
P01	Art der Bewegung 0 = rotatorisch Rundtisch, 1 Endschalte für Initialisierung 1 = linear Lineartisch, 2 Endschalte: Initialisierung und Begrenzung –Richtung Begrenzung +Richtung	0
P02	Maßeinheit der Bewegung 1 = Schritt 2 = mm 3 = Zoll 4 = Grad	1
P03	Umrechnungsfaktor Spindelsteigung 1 Schritt entspricht ... Bei P03 = 1 (Schritte) ist der Umrechnungsfaktor 1 Berechnung des Umrechnungsfaktors: $\text{Umrechnungsfaktor} = \frac{\text{Spindelsteigung}}{\text{Motorschrittzahl pro Umdrehung}}$ Beispiel: 4 mm Spindelsteigung 200-schrittiger Motor = 400 Schritte/U im Halbschrittbetrieb $\text{Umrechnungsfaktor} = \frac{4}{400} = 0,01$	1
P04	Start-/Stoppfrequenz Die Start-/Stoppfrequenz ist die maximale Frequenz, bei der der Schrittmotor noch ohne Rampe starten oder stoppen kann, ohne dass Schrittverluste auftreten. Die Start-/Stoppfrequenz ist abhängig von verschiedenen Größen wie Motortyp, Last, Mechanik, Endstufe. Eingabe der Frequenz in Hz	400
P05 P06	nicht belegt	
P07	Achsenrampe für Nothalt Eingabe in 4000-Hz/s-Schritten	100 000

Nr.	Bedeutung	Auslieferungszustand
P08	f_{\max} MØP, Fahrfrequenz beim Initialisieren Eingabe in Hz (ganzzahlige Werte)	4000
P09	Rampe MØP für Initialisierung, zugehörig zu Parameter P08 Eingabe in 4000-Hz/s-Schritten	4000
P10	f_{\min} MØP, Fahrfrequenz beim Verlassen der Endschalter Eingabe in Hz	400
P11	MØP Offset für Endschalter Plusrichtung Abstand des mechanischen Nullpunkts MØP (Referenzpunkt) vom Schaltpunkt des Endschalters. Einheit: wie in Parameter P02 festgelegt	0
P12	MØP Offset für Endschalter Minusrichtung Abstand des mechanischen Nullpunkts MØP vom Schaltpunkt des Endschalters. Einheit: wie in Parameter P02 festgelegt	0
P13	Beruhigungszeit MØP Wartezeit bei Initialisierung Eingabe in ms	20
P14	f_{\max} Lauffrequenz bei Positionierbefehlen Eingabe in Hz (ganzzahlige Werte) (max. 40 000)	4000
P15	Rampe für Lauffrequenz (P14) Eingabe in 4000-Hz/s-Schritten (4000 bis 500 000 Hz/s)	4000
P16	Beruhigungszeit Position Wartezeit nach Ausführung eines Fahrbefehls Eingabe in ms	20

Nr.	Bedeutung	Auslieferungszustand
P17	<p>Boost (definiert in P42)</p> <p>0 = aus 1 = ein während der Motor fährt 2 = ein bei Hochlauf und Absenkung der Fahrfrequenz (Rampe)</p> <p><u>Anmerkungen:</u></p> <p>Der Booststrom kann in Parameter P42 programmiert werden.</p> <p>Mit dem Parameter P17 wird festgelegt, wann die Steuerung auf Booststrom umschaltet.</p> <p>P17 = 1 bedeutet, dass bei fahrendem Motor immer der Booststrom fließt. Bei Stillstand des Motors wird auf Stoppstrom umgeschaltet.</p>	0
P18	nicht belegt	
P19	<p>Elektronischer-Nullpunkt-Zähler</p> <p>Dient zur Bestimmung von Arbeitspunkten, kann bei Achsenstillstand gesetzt und ausgelesen werden.</p>	0
P20	<p>Mechanischer-Nullpunkt-Zähler</p> <p>Zählt Impulse bezogen auf den mechanischen Nullpunkt. Kann bei Achsenstillstand ausgelesen werden. Am MØP wird P20 automatisch null gesetzt.</p>	0
P21	<p>Absolutwertzähler</p> <p>Gibt die aktuelle Position an. P21 kann jeder Zeit abgefragt, beschrieben oder geändert werden. P21 wird am MØP <u>nicht</u> automatisch null gesetzt.</p>	0
P22	<p>Encoderzähler</p> <p>Gibt die aktuelle Encoderposition an.</p>	0
P23	<p>Achsenbegrenzung pos. Richtung +</p> <p>Bei Erreichen dieser Schrittzahl wird der Lauf in +Richtung abgebrochen.</p> <p>0 = keine Begrenzung</p>	0

Nr.	Bedeutung	Auslieferungszustand
P24	Achsenbegrenzung neg. Richtung - Bei Erreichen dieser Schrittzahl wird der Lauf in –Richtung abgebrochen. 0 = keine Begrenzung	0
P25	Spielausgleich Gibt die Schrittzahl an, um die die Sollposition in der gewählten Richtung überfahren und anschließend in umgekehrter Richtung angefahren wird. 0 = kein Spielausgleich	0
P26	nicht belegt	
P27	Initiatorotyp 0 = PNP-Öffner 1 = PNP-Schließer	0
P28 bis P33 nicht belegt		
P34	Encodertyp 0 = keiner 1 = inkrementell 2 = serielle Schnittstelle SSI Binär Code 3 = serielle Schnittstelle SSI Gray Code  Achtung: Encodertyp korrekt eingeben! Falsches Parametrieren führt zu Beschädigungen!	0
P35	Auflösung bei Absolut-Encoder (SSI) Eingabe: maximale Auflösung in Bit (max. 31 Bit)	10
P36	Encoderfunktion 0 = Zähler	0
P37	nicht belegt	
P38	Encoder Vorzugsdrehrichtung 0 = plus 1 = minus	0
P39	Encoder Umrechnungsfaktor 1 Inkrement entspricht ...	1

7 Copyright und Haftungsausschluss

Die Kommunikationssoftware LabVIEW sowie die dazugehörige Dokumentation sind urheberrechtlich geschützt. Das Handbuch darf ohne Zustimmung von National Instruments weder in Teilen noch im Ganzen kopiert, fotokopiert, reproduziert, in eine maschinenlesbare Form gebracht oder auf andere Weise vervielfältigt werden. Von den Phytron eigenen VIs als Freeware Produkt ist es erlaubt Sicherungskopien für den persönlichen Gebrauch zu erstellen. Jedoch darf das Programm weder verändert noch verkauft werden

Einschränkung der Gewährleistung

Die Phytron spezifischen VIs und das zugehörige Handbuch wurden mit größter Sorgfalt erstellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Jede CD-ROM wird vor Auslieferung mit einem bekannten Scanner-Programm auf Viren aller Art überprüft. Trotzdem können Fehler existieren.

Phytron übernimmt keine Gewährleistung, dass diese Software frei von Fehlern ist. Phytron übernimmt keine Haftung für Schäden, die aus der Verwendung dieser Software entstehen. Der Benutzer verwendet die Software auf eigene Verantwortung.

Wir verweisen hier auf unsere Liefer- und Zahlungsbedingungen, insbesondere auf Punkt VII Haftung und Punkt IX Softwarenutzung.

Indem Sie das Softwareprodukt installieren, kopieren oder anderweitig verwenden, erklären Sie sich mit unseren Liefer- und Zahlungsbedingungen einverstanden. Falls Sie den Bestimmungen dieser AGB's nicht zustimmen, sind Sie nicht berechtigt, das Softwareprodukt zu installieren oder zu verwenden.

Geschützte Warenzeichen

Wir nehmen in diesem Handbuch auf geschützte Warenzeichen Bezug, die innerhalb des laufenden Textes nicht mehr explizit als solche gekennzeichnet sind. Aus dem Fehlen einer Kennzeichnung kann also nicht geschlossen werden, dass der entsprechende Produktname frei von Rechten Dritter ist:

LabVIEW ist ein eingetragenes Warenzeichen von National Instruments Corporation.

Microsoft ist ein eingetragenes Warenzeichen, und Windows ist eine Kennzeichnung der Microsoft Corporation in den Vereinigten Staaten und anderen Ländern.

8 Stichwortverzeichnis

A

A/D Wandler 9
A/D-Eingänge 21

B

BOOL-Cluster 15

C

Cluster 9
Copyright 35
Counter 10, 24

D

Demo 20
Demo-MCC.Ilb 6
Directmode 11, 22
Drive 12

E

Ein / Ausgänge 25
Eingang 15
Encoder 13
Encoder-Zähler 21

F

Fahrbefehl 12
Fehlercluster 9
Freeware 35
Funktionsblock 8

G

Gewährleistung 35

I

I/O 25
Inbetriebnahme 7
Init 14
Initiator-Status 14

Input 15

L

LabVIEW 6

M

MCC.Ilb 6
MINILOG 19

O

Operation 23

P

Parameter 17, 26, 29
Parameterliste 30
Parameter-Number 17

R

Read / Write Register 27
Register 18
Registerkarte 20
Register-Number 18

S

Schnittstellenparameter 9
Settings 20
Status 19, 28

V

VI 6
Virtuelle Instrumente 6

W

Warenzeichen 35
Warnhinweise 7

Z

Zählerwert 10

Phytron GmbH • Industriestraße 12 • 82194 Gröbenzell, Germany
Tel. +49(0)8142/503-0 • Fax +49(0)8142/503-190 • E-Mail info@phytron.de • www.phytron.de

Phytron, Inc. • 600 Blair Park Road Suite 220 • Williston, VT 05495 USA
Tel. +1-802-872-1600 • Fax +1-802-872-0311 • Email info@phytron.com • www.phytron.com