

Schrittmotormodul PCD2/3.H22

0	Inhaltsverzeichnis	
0.1	Dokument Historie	0-3
0.2	Warenzeichen	0-3
1	Einführung	
1.1	Funktion und Anwendung	1-1
1.2	Die wichtigsten Eigenschaften	1-2
1.3	Typische Einsatzgebiete	1-2
1.4	Mechanische Abmessungen	1-2
2	Technische Daten	
2.1	Technische Daten der Hardware.....	2-1
2.1.1	24 VDC Spannungsversorgung	2-1
2.1.2	5 VDC Stromversorgung ab PCD1/2/3-Bus.....	2-1
2.1.3	Eingänge	2-1
2.1.4	Ausgänge.....	2-1
2.1.5	Bidirektionelle Anschlüsse	2-2
2.1.6	Betriebsbedingungen	2-2
2.1.7	Bestellangaben	2-2
2.2	Anschlüsse.....	2-3
2.3	LED-Anzeige.....	2-4
3	Funktionsbeschreibung	
3.1	Parametrierung von Fahrprofilen	3-1
3.1.1	Automatische Profilkorrekturen	3-2
3.2	Fahrdistanzangabe	3-2
3.2.1	Relative Distanzangabe	3-2
3.2.2	Absolute Zielangabe	3-2
3.3	Verhalten bei Notstop im Betrieb (Emergency Stop)	3-2
3.4	Verhalten bei LimitSwitch im Betrieb.....	3-3
3.5	Eventgesteuertes Starten von Achsen.....	3-3
3.6	Positionierung auf Endschalter	3-3
3.7	Referenzfahrt	3-4
3.7.1	Suchen des Endschalters, anschliessendes Anfahren des Referenzschalters	3-4
3.7.2	Suchen des Referenzschalters ausserhalb der Endschalter	3-4
3.7.3	Direktes An- und Überfahren des Referenzschalters.....	3-5
3.8	Einzelschritte.....	3-5
3.9	Positionierung mit konstanter Geschwindigkeit	3-5
3.10	Positionierung mit Fahrprofil	3-5
3.11	Stoppen einer Fahrbewegung	3-5

4	Programmierbare Funktionen in PG5	
4.1	Zugriff mit Funktionsblöcken (FB)	4-1
4.1.1	Diagnose Flag und Register	4-2
4.1.2	FB init.....	4-2
4.1.3	FB initMP	4-3
4.1.3	FB exec.....	4-4
4.1.5	Kommandos.....	4-6
4.1.6	Schreibbefehle	4-7
4.1.7	Lesebefehle	4-8
4.2	Systemkonfiguration mit dem Device Configurator.....	4-11
4.3	Properties.....	4-13
4.4	Befehlssatz Direktzugriff	4-14
4.4.1	Schreibbefehle	4-14
4.4.2	Lesebefehle	4-16
5	Programmanpassungen bei Ablösung H210 auf H222	
6	Beispiele	
6.1	Hardware PCD2	6-1
6.2	Hardware PCD3	6-2
6.3	Programmbeschreibung.....	6-3
6.4	IL Programmcode.....	6-3
A	Anhang	
A.1	Icons	A-1
A.2	Kontakt.....	A-2

0.1 Dokument Historie

Version	Bearbeitung	Veröffentlichung	Bemerkungen
pDE01	2009-11-03	2009-11-30	Erste Ausgabe für "Launch-Page"
pDE01	2010-09-07	2010-09-07	Kapitel Ch04 neu
DE01	2010-10-31	2010-11-15	Div. Änderungen
DE01	2011-01-03	2011-04-15	Div. Änderungen und neues Cover
DE02	2011-08-25	2011-08-26	Neue Telefonnummer: +41 (0)26 / 672 72 72 Variablenname «IO.Slot0.IOAccess.X_AXIS_RD_ACTPOSABS» korrigiert
DE03	2013-06-14	2013-08-14	Logo und Namen geändert
DE04	2014-01-16	2014-02-19	Diverse Änderungen im ganzen Handbuch
GER05	2017-04-30	2017-04-30	Tabelle 4.4.2 - eine Zeile gelöscht

0.2 Warenzeichen

Saia PCD® und Saia PG5®
sind registrierte Warenzeichen der Saia-Burgess Controls AG.

Technische Änderungen sind abhängig vom Stand der Technik.

Saia-Burgess Controls AG, 2017. © All rights reserved.

Veröffentlicht in der Schweiz

1 Einführung

Dieses Modul lässt sich an jedem beliebigen E/A-Steckplatz einer PCD2 oder PCD3 einstecken. Es dient der Ansteuerung der Leistungsstufe zu einer Schrittmotorachse bis zu einer Frequenz von 20 kHz.



Die maximale Anzahl Module ist durch die Begrenzungen der Erweiterungen PCD2 respektive PCD3 gegeben. Die Basisadresse 255 darf durch kein Modul belegt werden.

1.1 Funktion und Anwendung

Mit dem Modul PCD2/3.H222 können zwei Schrittmotor-Endstufen angesteuert werden. Das Modul übernimmt dabei die Steuerung der gewünschten Fahrbewegungen und gibt die drei Signale MOTEN (Freigabe), DIR (Drehrichtung) und PUL (Impuls) an die Endstufe weiter. Neben Einzelschritten und mehreren Schritten mit konstanter Geschwindigkeit können auch Fahrprofile in Trapez- oder S-Form abgefahren werden, deren Hochlauf- und Bremsrampen asymmetrisch sind. Positionierungen können sowohl absolut als auch relativ gemacht werden.

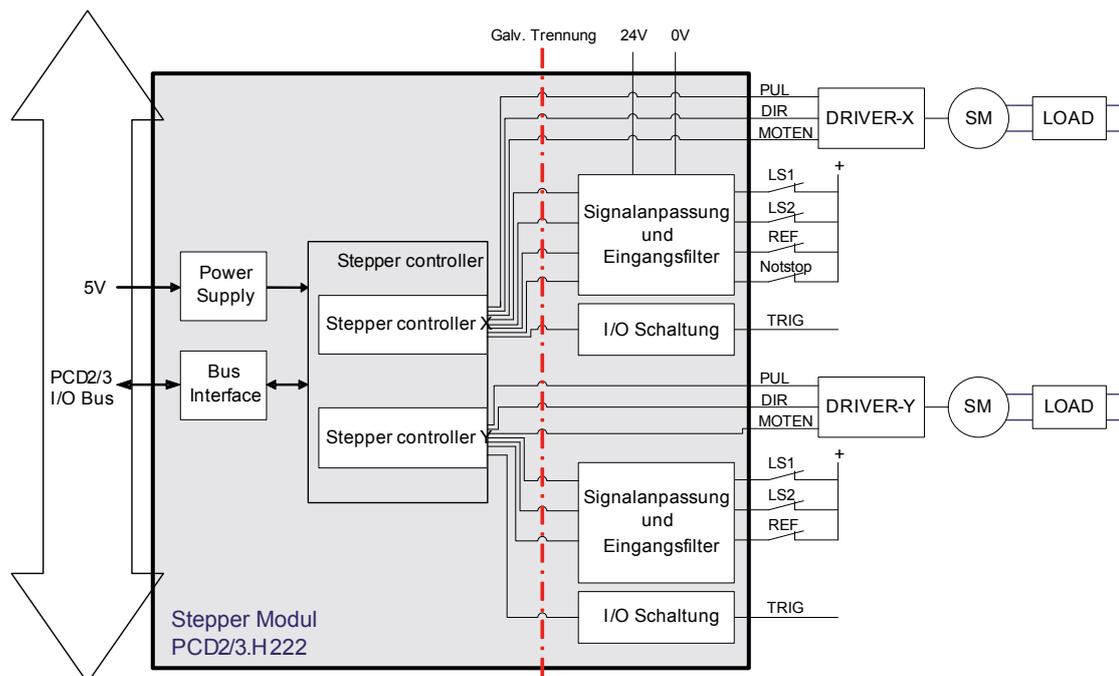
Pro Achse stehen 3 Eingänge für LS1 (Endschalter 1), LS2 (Endschalter 2) und REF (Referenzschalter) zur Verfügung. Dadurch kann mit einer einzigen Anweisung die Positionierung auf einen entsprechenden Schalter ausgelöst werden.

Über einen konfigurierbaren TRIG (trigger) Anschluss pro Achse können Fahraufträge über ein beliebiges 24V Signal gestartet werden. Verwendet man den TRIG-Anschluss bei einer Achse als Eingang und bei einer anderen Achse als Ausgang, können so Fahrprofile modulübergreifend synchron gestartet werden.

Ein weiterer Eingang pro Modul steht für die Notaus-Erkennung zur Verfügung.

Alle Ein- und Ausgänge sind galvanisch vom I/O-Bus getrennt.

Blockschaltbild eines Schrittmotorantriebes



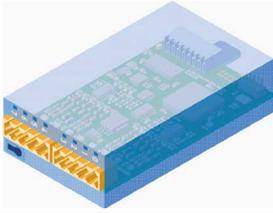
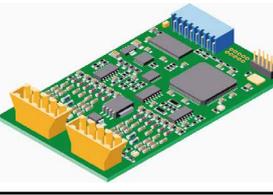
1.2 Die wichtigsten Eigenschaften

- Ein gemeinsamer Notaus-Eingang
- Pro Achse 3 Eingänge (1 Referenz- und 2 Endschalter)
- Pro Achse 3 Ausgänge (PUL, DIR, MOTEN)
- Ein konfigurierbarer Synchronisier-Ein-/Ausgang pro Achse
- Positioniermöglichkeiten: Einzelschritt, konstante Geschwindigkeit, Fahrprofile, automatisch auf Referenz- oder Endschalter fahren
- 16 Profile pro Achse für Parametrierung von S-Kurve oder Trapez mit asymmetrischer Start- und Bremsrampe
- absolute oder relative Positionsangabe
- endlose Fahrbewegungen

1.3 Typische Einsatzgebiete

- Handling- und Montageautomaten
- Pick- and Place-Funktionen
- Preisgünstige Palletierungs- und Montage-Antriebe
- Automatische Winkelsteuerung, z.B. von Kameras, Scheinwerfern, Antennen usw.
- Positionierung statischer Achsen (Set up)

1.4 Mechanische Abmessungen

<p>PCD3 Modul</p> 	<p>Abmessung: PCD3 I/O Modul</p> <p>Gehäuse Farbe: Blau</p> <p>Verbindung: 2× Steckbarer 10-poliger Federkraft-Klemmenblock</p>
<p>PCD2 Modul</p> 	<p>Abmessung: PCD2 I/O Modul</p> <p>Verbindung: 2× Steckbarer 10-poliger Federkraft-Klemmenblock</p>

2 Technische Daten

2.1 Technische Daten der Hardware

2.1.1 24 VDC Spannungsversorgung

Parameter	Wert
Spannungsbereich	18 ... 32 VDC
Stromaufnahme bei 24 V	< 80 mA
Überspannungsschutz	Ja (39 V \pm 10%)
Verpolungsschutz	Nein

2.1.2 5 VDC Stromversorgung ab PCD1/2/3-Bus

Parameter	Wert
Stromaufnahme bei 5 V	< 85 mA

2.1.3 Eingänge

Parameter	Wert
Anzahl	7
Spannungsbereich	0 ... 32 V
Low-Bereich	0 ... 5 V
High-Bereich	15 ... 32 V
Schaltswelle Low-High	ca. 10.9 V
Schaltswelle High-Low	ca. 9.1 V
Hysterese	ca. 1.8 V
Eingangstrom (24 V)	ca. 5.2 mA
Einschaltverzögerung	ca 200 μ s
Ausschaltverzögerung	ca 200 μ s

2.1.4 Ausgänge

Parameter	Wert
Anzahl	6
Schritimpulse (PUL)	aktiv high
Pulsbreite für das Schrittsignal	16.5 μ sec
Richtungssignal (DIR)	low = vorwärts (cw) high = rückwärts (ccw)
Bewegungssignal (MOTEN)	high = Motor freigegeben low = Motor nicht freigegeben
Belastung pro Ausgang	< 20 mA
Betriebsart	Senkbetrieb (Minus wird geschaltet)
Einschaltverzögerung MOTEN, DIR	< 15 μ s
Ausschaltverzögerung MOTEN, DIR	< 25 μ s
Einschaltverzögerung PUL	< 1.5 μ s
Ausschaltverzögerung PUL	< 500 ns

2.1.5 Bidirektionelle Anschlüsse

Es stehen zwei bidirektionelle Anschlüsse zur Verfügung:

Input/Output TRIG Achse X.

Input/Output TRIG Achse Y.

Dieser Anschluss kann als Ausgang oder als Eingang verwendet werden. Dies erlaubt es, eine Master-Achse zu bestimmen, zu welcher die angeschlossenen Slave-Achsen synchronisiert werden können.

Kenndaten bei Verwendung als Eingang

Parameter	Wert
Spannungsbereich	0 ... 32 V
Low-Bereich	0 ... 5 V
High-Bereich	15 ... 32 V
Schaltswelle Low-High	ca. 9.3 V
Schaltswelle High-Low	ca. 11 V
Hysterese	ca. 1.7 V
Eingangsstrom (24V)	3.85 mA
Einschaltverzögerung	< 170 µs
Ausschaltverzögerung	< 184 µs

Kenndaten bei Verwendung als Ausgang

Parameter	Wert
Spannungsbereich	0 ... 32 V
Low-Bereich	0 ... 5 V
High-Bereich	15 ... 32 V
Belastung pro Ausgang	> 200 mA
Einschaltverzögerung	< 1.5 µs
Ausschaltverzögerung	< 28 µs
Betriebsart	Quellbetrieb

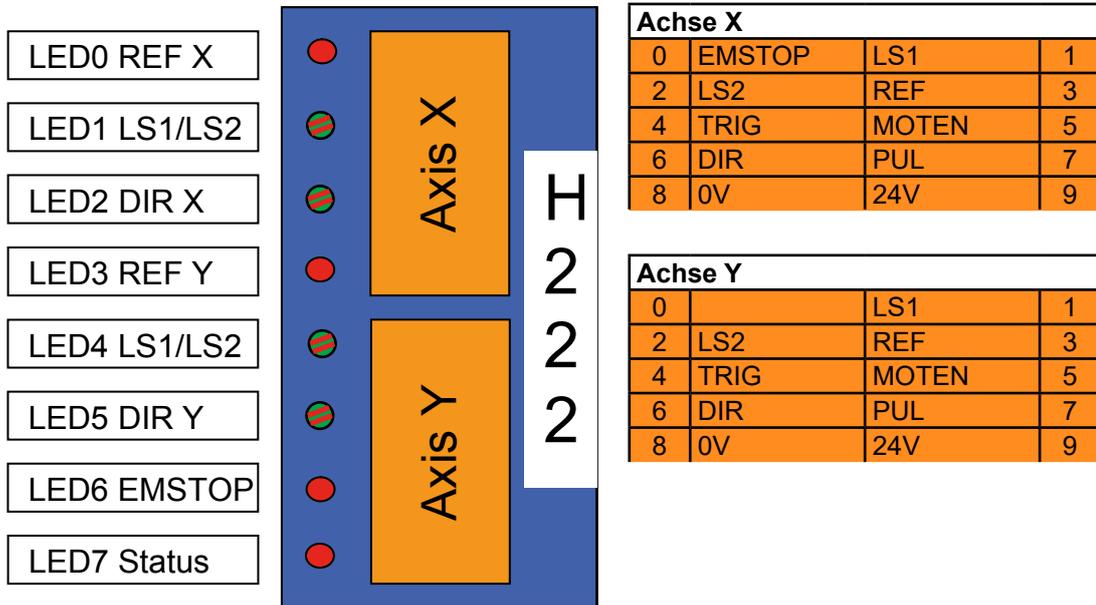
2.1.6 Betriebsbedingungen

Umgebungs-Temperatur	Betrieb: 0 °...+55 °C ohne Zwangsbelüftung, Lagerung: -20...+85 °C
----------------------	--

2.1.7 Bestellangaben

Typ	Beschreibung	Gewicht
PCD2.H222	Stepper Modul für 2 Achsen	27 g
PCD3.H222	Stepper Modul für 2 Achsen	70 g

2.2 Anschlüsse



Ein Notstoppschalter kann am Eingang EMSTOP angeschlossen werden. Die Pinbelegung ist aus der Abbildung erkennbar. Dieser Eingang dient nur zum Erkennen einer Not-Abschaltung und darf nicht für sicherheitsrelevante Funktionen eingesetzt werden. Die Anschlüsse LS1, LS2 und REF sind Eingänge, welche für die Positionierung auf einen der Schalter vorgesehen sind. LS1 und LS2 beschränken zudem den nutzbaren Bereich einer Schrittmotorenachse. Durch die Konfigurierbarkeit dieser Eingänge können sowohl Öffner- wie auch Schliesskontakte verwendet und nicht angeschlossene Schalter deaktiviert werden, was zu einer verbesserten Fehlererkennung führt.

Wird einer der Eingänge deaktiviert, so lässt sich dieser als digitaler Eingang verwenden. Dazu ist das IO-Status-Register durch das Anwenderprogramm auszulesen. Zusätzliche Informationen dazu sind im Kapitel 4.3.2 «Lesebefehle».

Der Anschluss TRIG kann als Eingang oder Ausgang konfiguriert werden und dient zum eventgesteuerten Starten verschiedener Achsen.

Ist der Anschluss aktiviert und als Eingang konfiguriert, so startet eine aus dem Anwenderprogramm gestartete Fahrbewegung erst, wenn am TRIG Anschluss 24V anliegen.

Verwendet man TRIG als Ausgang, wird dieser zeitgleich mit dem MOTEN Ausgangssignal ein- und auch wieder ausgeschaltet.

MOTEN, DIR und PUL sind im Senkbetrieb verwendete Ausgänge zum Ansteuern der Schrittmotor-Endstufe.

Die Stromversorgung des Modules erfolgt über den Anschluss +24V und 0V. Diese beiden Signale sind intern zwischen den Anschlussklemmen der X- und Y-Achse verbunden.

Mit Ausnahme des Signals EMSTOP, +24V und 0V sind alle Signale einmal für die X-Achse und einmal für die Y-Achse vorhanden.

2.3 LED-Anzeige

Achse X

LED 0	Spannung am Eingang REF
LED 1 (rot)	Spannung am Eingang LS1
LED 1 (grün)	Spannung am Eingang LS2
LED 2 (rot)	Spannung am Ausgang MOTEN und Ausgang DIR
LED 2 (grün)	Spannung am Ausgang MOTEN und 0V am Ausgang DIR

2

Achse Y

LED 3	Spannung am Eingang REF
LED 4 (rot)	Spannung am Eingang LS1
LED 4 (grün)	Spannung am Eingang LS2
LED 5 (rot)	Spannung am Ausgang MOTEN und Ausgang DIR
LED 5 (grün)	Spannung am Ausgang MOTEN und 0V am Ausgang DIR

Für beide Achsen

LED 6:	Spannung am Eingang EMSTOP
LED 7:	Visualisierung vom Betriebszustand und Fehlercodes

EMSTOP und REF werden durch eine einfache LED visualisiert. LS1 und LS2 werden auf einer zweifarbigen LED (LS1 rot und LS2 grün) angezeigt. Ebenfalls eine zweifarbige LED zeigt den Status von DIR und MOTEN. Ist MOTEN = 1, so unterscheidet sich die LED-Farbe in Abhängigkeit der Drehrichtung DIR (counterclockwise „ccw“ rot und clockwise „cw“ grün).

Die LED7 visualisiert einen allfällig aufgetretenen Fehler einer der beiden Achsen.



Abhängig von der Verdrahtung und dem Zustand von LS1 und LS2, können die LED 1 und die LED4 orange leuchten

3 Funktionsbeschreibung

Das Modul H222 wird in eine PCD2 oder PCD3 eingesetzt und mittels Peripheriezugriffe aus dem Anwenderprogramm angesteuert. Diese lösen eine Kommunikation zwischen PCD und H222 über den Standard I/O-Bus aus. Das Modul erkennt neue Fahrbefehle automatisch und leitet die richtigen Massnahmen ein, um diese abzuarbeiten.

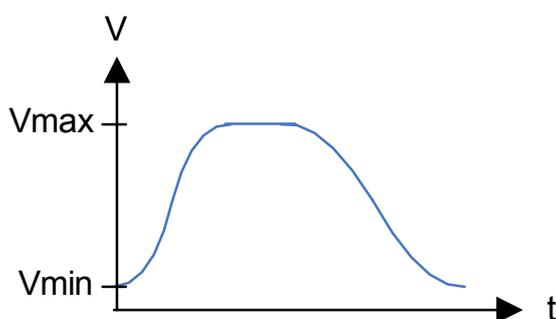
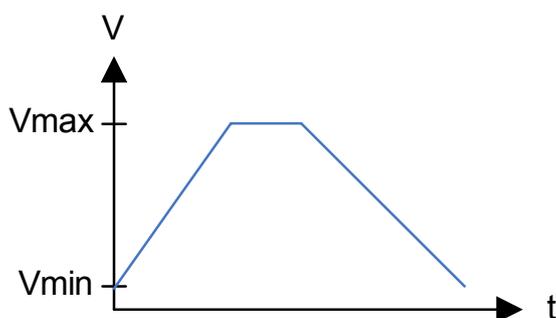
3.1 Parametrierung von Fahrprofilen

Damit nicht alle Parameter für ein Fahrprofil jedes Mal von der PCD zum Modul gesendet werden müssen, können 16 Fahrprofile definiert werden und, einmal abgespeichert, mit einem einzigen Kommando aktiviert werden.

Zu einem Fahrprofil gehören die folgenden Parameter:

Parameter	Beschreibung	Bereich, Einheit	Auflösung, Bemerkung
Vmin	Start- Stoppgeschwindigkeit	10...10'000 Hz	1 Hz (16 bit)
Vmax	Maximale Geschwindigkeit	20...20'000 Hz	1 Hz (16 bit)
Acc	mittlere Beschleunigung	1...1000 kHz/s	1 kHz/s (16 bit)
Dec	mittlere Verzögerung	1...1000 kHz/s	1 kHz/s (16 bit)
Jerk_acc	Jerk Startrampe in Prozent	0...50%	1% (6 bit)
Jerk_dec	Jerk Bremsrampe in Prozent	0...50%	1% (6 bit)

Die beiden Jerk-Werte geben an, nach welchem prozentualen Anteil der Zeit für die Beschleunigungs- bzw. Bremsphase die lineare Beschleunigung oder Verzögerung erreicht werden soll. Die lineare Phase dauert dann $100\% - 2 \cdot \text{Jerk-Wert}$. Ein Jerk-Wert von 0% entspricht also einem trapezförmigen Fahrprofil, während bei 50% eine S-Kurve abgefahren wird.



Die Zeitdauer zum Abfahren des jeweiligen Profils ergibt sich aus diesen Parametern und der Fahrdistanz.

3.1.1 Automatische Profilkorrekturen

Werden Werte eingestellt, welche nicht im zulässigen Bereich liegen, so korrigiert das Modul diese automatisch auf den entsprechenden Minimal- respektive Maximalwert ab. Es wird eine Warnung abgesetzt. Sollte einer der Werte V_{min} , V_{max} , Acc oder Dec gleich 0 sein, so wird das Profil nicht gestartet und eine Fehlermeldung abgesetzt.

Zwischen gewissen Eingabeparametern bestehen Abhängigkeiten, welche eingehalten werden müssen. Werden diese verletzt, kann es vorkommen, dass z.B. eine eingestellte S-Kurve auf einen Trapezverlauf reduziert wird, um Schrittfehler zu verhindern.

3.2 Fahrdistanzangabe

Die Fahrdistanzangabe erfolgt unabhängig vom Fahrprofil entweder relativ, absolut oder als endlose Fahrbewegung. Wobei letztere ein Spezialfall der relativen Distanzangabe bildet. Fahrdistanzangaben haben einen Einfluss auf die Positionierungsarten „Positionierung mit konstanter Geschwindigkeit“ und „Positionierung mit Fahrprofil“.

3.2.1 Relative Distanzangabe

Bei der relativen Distanzangabe wird die Anzahl abzufahrende Schritte angegeben. Beim Starten der Fahrbewegung ist dann entsprechend die Drehrichtung anzugeben. Die zulässige Anzahl Schritte liegt zwischen 0 und $16'777'215$, wobei $16'777'215$ zu einer endlosen Fahrbewegung führt. Alle anderen Werte führen zur entsprechenden Anzahl Schritte.

3.2.2 Absolute Zielangabe

Absolute Positionierungen erlauben es, den Endpunkt anzugeben ohne die aktuelle Position zu kennen. Das Modul berechnet daraus die relative Fahrdistanz und die Drehrichtung in Abhängigkeit der aktuellen Absolutposition selber. Beim Starten der Fahrbewegung wird der Übergabeparameter Drehrichtung vernachlässigt.

3.3 Verhalten bei Notstop im Betrieb (Emergency Stop)

Der Notstop-Eingang erlaubt es, eine laufende Bewegung beider Achsen des Moduls mit der vordefinierten Bremsrampe zu stoppen. Da die effektive Position nach einer extern erfolgten Schnellabschattung nicht mehr übereinstimmt, muss nach einem Notstop einer der Endschalter oder Referenzschalter angefahren werden.



Sicherheitsrelevante Schnellabschaltungen müssen vom Modul unabhängig vorgenommen werden.

3.4 Verhalten bei LimitSwitch im Betrieb

Wenn ‚LS1‘ oder ‚LS2‘ aktiv werden, wird automatisch eine Bremsrampe ausgelöst. Die aktuelle Absolutposition geht dabei nicht verloren. Allerdings wird eine Fehlermeldung erzeugt, welche zu quittieren ist. Der aktive Bereich eines Endschalters kann nur durch eine Positionierung auf den entsprechenden Endschalter oder allenfalls durch die Referenzfahrt verlassen werden.

3.5 Eventgesteuertes Starten von Achsen

Konfiguriert man den TRIG Anschluss als Eingang und aktiviert diesen, können Fahrbefehle auf eine Achse abgesetzt werden, welche erst gestartet werden, wenn 24V am TRIG Anschluss anliegen. Dazu kann ein beliebiges digitales 24V Signal verwendet werden.

Sollen zwei oder mehr Achsen synchronisiert starten, dann ist der TRIG Achschluss für Folgeachsen als Eingang zu konfigurieren und zu aktivieren. Bei der Steuerachse wird der Anschluss als Ausgang aktiviert. Alle Anschlüsse sind extern durch ein Kabel miteinander zu verbinden. Zuerst werden die Fahraufträge für alle Folgeachsen abgesetzt. Diese warten nun auf das Startsignal am TRIG Anschluss. Dieses erfolgt, wenn nun die Steuerachse zum entsprechenden Zeitpunkt gestartet wird.

3.6 Positionierung auf Endschalter

Bei der Fahrt auf einen Endschalter gibt es zwei verschiedene Startpositionen zu unterscheiden. Entweder befindet man sich zwischen den beiden Endschaltern und fährt in Richtung Endschalter mit dem eingestellten Geschwindigkeitsprofil, bremst ab bis zum Stillstand sobald der Endschalter aktiv wird und fährt mit V_{min} in Gegenrichtung bis der Endschalter inaktiv wird. Oder aber man befindet sich im aktiven Bereich des Endschalters -Flag. Hier wird direkt mit V_{min} zum gültigen Bereich zwischen den Endschaltern hin gefahren bis der Endschalter inaktiv wird.

Das Fahren auf den näher liegenden Endschalter aus dem ungültigen Bereich (außerhalb der beiden Endschalter) ist nicht direkt möglich. Hierzu muss vorgängig in den gültigen Bereich gefahren werden.

Die Positionierung auf einen der beiden Endschalter wird mit dem Peripheriezugriff und je nach gewünschter Achse dem Befehl `X_AXIS_WR_LSREFMOVE` oder `Y_AXIS_WR_LSREFMOVE` gestartet. Auf welchen der beiden Endschalter positioniert werden soll, wird mit dem Übergabeparameter festgelegt (1 = LS1, 2 = LS2).

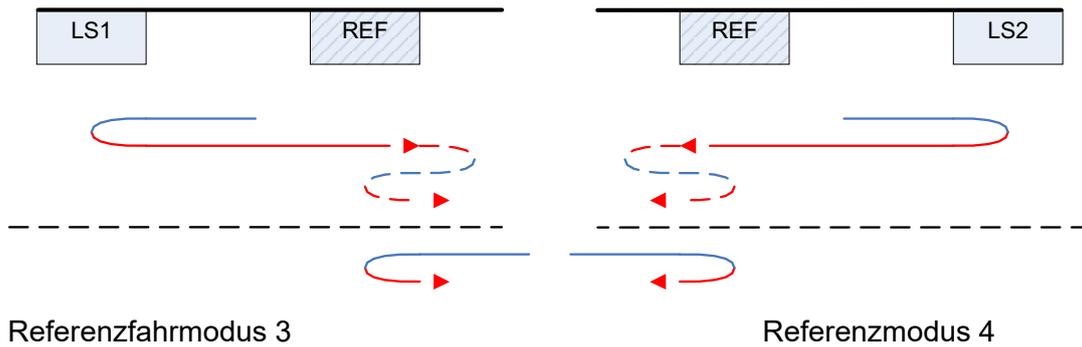
3.7 Referenzfahrt

Zum Positionieren auf den angeschlossenen Referenzschalter stehen unterschiedliche Modi zur Verfügung. Unabhängig vom Modus muss der Referenzschalter aktiviert sein, ansonsten wird eine Fehlermeldung abgesetzt.

3.7.1 Suchen des Endschalters, anschliessendes Anfahren des Referenzschalters

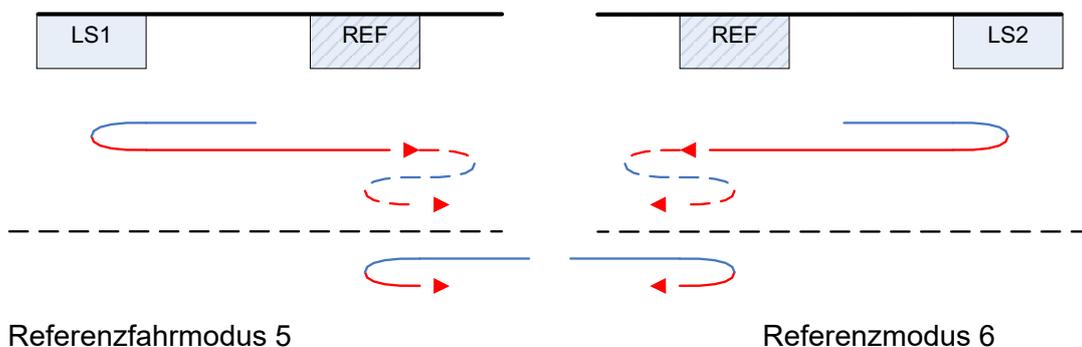
Referenziert man in Vorwärtsrichtung (Referenzfahrmodus 3), so wird beginnt die Fahrt mit dem aktiven Geschwindigkeitsprofil in Rückwärtsrichtung bis der Referenz- oder der Endschalter aktiv wird. Dies löst den Bremsvorgang aus. Nach dem Stillstand wird die Fahrtrichtung geändert und mit dem Geschwindigkeitsprofil bis zum Referenzschalter gefahren. Wird der Referenzschalter aktiv, beginnt die Verzögerung. Sollte der andere Endschalter aktiv werden, stoppt die Referenzfahrt und ein Fehler wird angezeigt.

Ist der Referenzschalter nach dem Verzögerungsrampe aktiv, so wird ohne Unterbruch mit V_{min} weiter gefahren, bis der Referenzschalter inaktiv wird. Andererseits wird erst in Gegenrichtung mit V_{min} der Referenzschalter angefahren und erst dann mit V_{min} in Vorwärtsrichtung aus dem aktiven Bereich vom Referenzschalter gefahren.



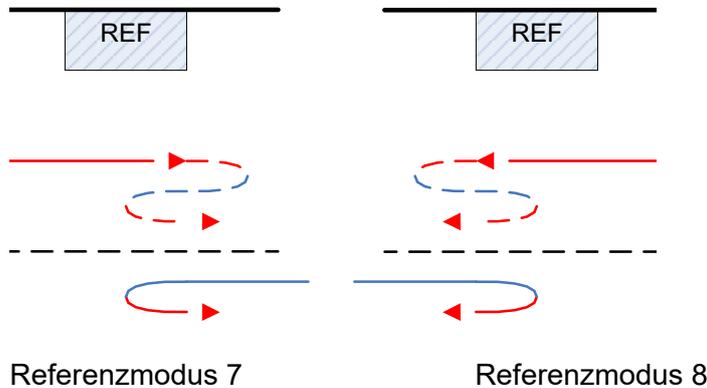
3.7.2 Suchen des Referenzschalters ausserhalb der Endschalter

Mit dem Referenzfahrmodus 5 ist es möglich den Referenzschalter ausserhalb der Endschalter zu montieren und trotz aktivierten Endschaltern den Referenzschalter anzufahren.



3.7.3 Direktes An- und Überfahren des Referenzschalters

Bei dieser Art der Referenzierung wird direkt mit der finalen Drehrichtung gestartet und der Referenzschalter überfahren. Sollte beim Referenzmodus 7 der LS1 aktiv werden (sofern dieser aktiviert ist), wird die Referenzfahrt gestoppt und eine Fehlermeldung abgesetzt. Selbes gilt beim Modus 8 für den LS2.



3.8 Einzelschritte

Jeder Einzelschritt bedingt ein Kommando von der PCD. Dieser wird mit dem eingestellten Wert für V_{min} ausgegeben. Sollte dieser nicht festgelegt sein, wird mit der Frequenz von 10 Hz verschoben. Die maximal mögliche Schrittfolge im Einzelschrittbetrieb liegt bei 350 Hz.

3.9 Positionierung mit konstanter Geschwindigkeit

Beim Starten der Bewegung mit V_{min} werden die Parameter V_{min} und die Anzahl zu absolvierender Schritte verwendet. Wird die Anzahl Schritte auf $16'777'215$ ($2^{24}-1$) gesetzt, so startet die Bewegung mit V_{min} und stoppt nur beim Erreichen des entsprechenden Endschalters oder durch einen Stop-Befehl aus dem Anwenderprogramm.

3.10 Positionierung mit Fahrprofil

Diese, wohl üblichste, Positionierungsart ist die schnellste Variante, um eine Zielposition zu erreichen. Dabei werden die aktuellen Fahrprofilparameter verwendet und je nach Einstellungen der Parameter und der Positionierdistanz eine S-Kurve, ein Trapez, ein Dreieck oder eine konstante Geschwindigkeit abgefahren.

3.11 Stoppen einer Fahrbewegung

Gestartete Fahrbewegungen können mit einer Stop-Instruktion abgebrochen werden. Dabei wird die vordefinierte Bremsrampe berücksichtigt und die Achse zum Stillstand gebracht. Diese Instruktion löst eine Fehlermeldung aus, welche vor dem nächsten Fahrbefehl quittiert werden muss.

4 Programmierbare Funktionen in PG5



Achten sie auf die unterschiedliche Ansteuerungsmöglichkeiten des Moduls über Direktzugriff (Media Mapping) oder Funktionsblöcke (FB)

4.1 Zugriff mit Funktionsblöcken (FB)

Das Modul H222 wird in eine PCD2 oder PCD3 eingesetzt und mittels FB's im Programmierwerkzeug PG5 angesteuert. Diese lösen eine Kommunikation zwischen PCD und H222 über den Standard IO-Bus aus. Dabei werden die Bewegungsparameter und Fahrbefehle im entsprechenden Register des Mikrocontrollers auf dem Erweiterungsmodul gesetzt.

Die Modul-Firmware erkennt neue Fahrbefehle automatisch und leitet die Richtigen Massnahmen ein, um diese abzuarbeiten. Damit nicht alle Parameter für ein Fahrprofil jedes Mal von der SPS zum Modul gesendet werden müssen, können 16 Fahrprofile definiert werden und, einmal abgespeichert mit einem einzigen Kommando aktiviert werden.

Das Modul ermöglicht das Anschliessen von Notstop-, End- und Referenzschaltern. Sicherheitstechnische Schnellabschaltungen müssen aber extern vorgenommen werden. Durch die Konfigurierbarkeit der Eingänge können sowohl Öffner- wie auch Schliesskontakte verwendet werden und nicht angeschlossene Schalter deaktiviert werden, was zu einer erhöhten Sicherheit durch verbesserte Fehlererkennung führt.

Mit einem Funktionsblock „Init“ wird das Modul einmalig initialisiert. Der „InitMP“ Funktionsblock erlaubt das Übermitteln eines kompletten Fahrprofils mit einem FB-Aufruf. Der FB „Exec“ erlaubt das Übergeben eines so genannten „command“. Abhängig von der Art der Anweisung wird ein Datenbyte mitgegeben oder als Rückgabeparameter zurück gelesen. Eine Liste aller zulässigen „Commands“ und deren Beschreibung befindet sich im Kapitel

4.1.1 Diagnose Flag und Register

Das Diagnose Register und die Diagnose Flag werden pro initialisierte Achse nach den jeweiligen Aktionen aufdatiert.

Struktur von Diagnose Register rDiag:

```

;          31.....24 23.....16 15.....8 7.....0
;          \ Fehler / \ FB-Nr. / \ Par.-Nr. / \ Modul-Nr./
;
;          Fehler:      00h -> Kein Sonderfehler
;                       01h -> H222 FB wird bereits genutzt
;          FB-Nr.:      01h -> H222.Init
;                       02h -> H222.InitMP
;                       03h -> H222.Exec
;          Par.-Nr.:    00h -> Kein Parameterfehler
;                       01h -> Modulnummer nicht korrekt
;                       02h -> Achs-Nummer nicht korrekt
;                       03h -> Befehlscode nicht korrekt (<0 oder >255)
;          Modul-Nr.:   Nummer des Moduls (FB-Parameter 1), das
;                       den Fehler verursacht

```

4

4.1.2 FB init

Dieser FB initialisiert alle H222-Module, welche in dem entsprechenden .equ-File definiert wurden. fDiag zeigt, ob irgendein Fehler während der Initialisierung erfolgte oder nicht. rDiag dient zu dessen Identifizierung.

Der Aufruf des FB «INIT» präsentiert sich wie folgt:

```

CFB   H222.Init
R rDiag      ; Par. 1: Diagnose Register
F fDiag      ; Par. 2: Status Flag (0: Init OK, 1: Fehler)

```

Oder als FB-Aufruf dargestellt:



4.1.3 FB initMP

Dieser FB initialisiert ein Bewegungsprofil (MP: motion profile) und konfiguriert die End-, Referenz- und Notstopsschalter. rDiag liefert eine Rückmeldung, welche im Fehlerfall hilft die Ursache zu bestimmen. fDiag zeigt, ob irgendein Fehler während der Initialisierung erfolgte oder nicht.

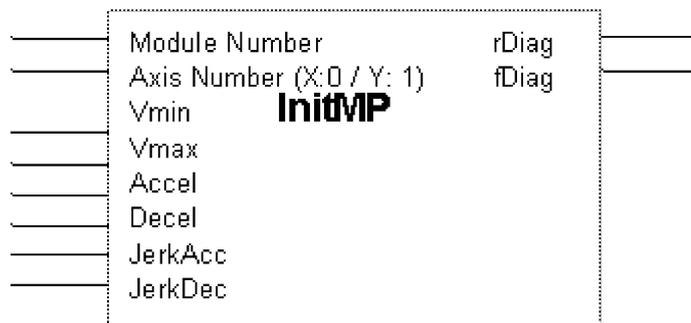
Der Aufruf des FB «INITMP» präsentiert sich wie folgt:

CFB H222.InitMP

R modNr ; Par. 1: Modul-Basisadresse
 R axisNr ; Par. 2: Achs-Nummer (0 = X-Achse, 1 = Y-Achse)
 R Vmin ; Par. 3: Vmin in Hz
 R Vmax ; Par. 4: Vmax in Hz
 R Accel ; Par. 5: Beschleunigung in kHz/s
 R Decel ; Par. 6: Bremsung in kHz/s
 R JerkAcc ; Par. 7: Jerk-Prozent Beschleunigung (0 bis 50%)
 R JerkDec ; Par. 8: Jerk-Prozent Bremsung (0 bis 50%)
 R rDiag ; Par. 9: Diagnose Register
 F fDiag ; Par.10: Status Flag (0: InitMP OK, 1: Fehler)

4

Oder als FB-Aufruf dargestellt:



4.1.3 FB exec

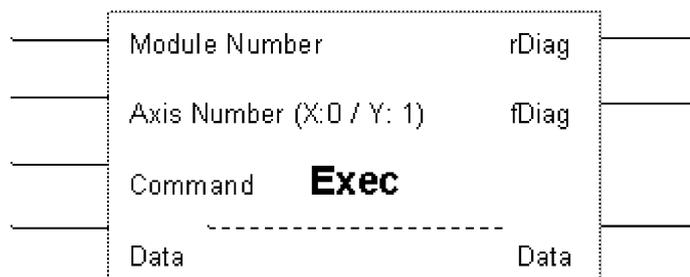
Der individuelle Baustein „exec“ ermöglicht es die folgenden Befehle auszuführen und allenfalls weitere Daten wie zum Beispiel die Drehrichtung oder ähnliches mitzugeben. Andere Commands liefern in Data Rückgabewerte (siehe Beschreibung der Befehle). fDiag signalisiert einen aufgetretenen Fehler und rDiag hilft diesen einer Ursache zuzuordnen.

Der Aufruf des FB «EXEC» präsentiert sich wie folgt:

```
CFB    H222.Exec
R modNr    ; Par. 1: Modul-Basisadresse
R axisNr   ; Par. 2: Achs-Nummer (0 = X-Achse, 1 = Y-Achse)
R rCommand ; Par. 3: Befehl (siehe Befehlsliste unten)
R rData    ; Par. 4: Übergabe- oder Rückleseparameter (z.T optional)
R rDiag    ; Par. 5: Diagnose Register
F fDiag    ; Par. 6: Status Flag (0: Exec OK, 1: Fehler)
```

4

Oder als FB-Aufruf dargestellt:



Das Baustein FB exec unterstützt die folgenden Befehle (Command):

Komandos:

- Einzelschritt (SingleStep)
- Referenzfahrt oder LS suchen und verlassen oder Overrun-Bereich verlassen (LsRefMove)
- Positionieren mit konst. Geschwindigkeit (MoveVconst)
- Positionieren mit Geschwindigkeitsprofil (StartMP)
- Fahrbefehl abbrechen (Stop)
- Freigabe der Fahrbefehle nach „Stop“ (Cont)
- Fahrprofil abspeichern (SaveMP)
- Fahrprofil laden (ActMP)
- Modulfirmware neu starten (ResMC)
- Fehlerregister löschen (ClrError)

Parametrierbefehle:

- mittlere Anfahrbeschleunigung eingeben (LdAcc)
- mittlere Bremsverzögerung eingeben (LdDec)
- Maximalgeschwindigkeit eingeben (LdVmax)
- Minimalgeschwindigkeit eingeben (LdVmin)
- Jerk-Wert für Beschleunigungsbereich eingeben (LdJPa)
- Jerk-Wert für Bremsbereich eingeben (LdJPd)
- Sollposition relativ eingeben (LdDestRel)
- Sollposition absolut eingeben (LdDestAbs)
- Absolutposition für momentanen Standort der Achse setzen (LdActPosAbs)
- Endschalterlogik eingeben (LdIoConf)

4

Rücklesebefehle:

- mittlere Anfahrbeschleunigung auslesen (RdAcc)
- mittlere Bremsverzögerung auslesen (RdDec)
- Maximalgeschwindigkeit auslesen (RdVmax)
- Minimalgeschwindigkeit auslesen (RdVmin)
- Jerk-Wert für Beschleunigungsbereich auslesen (RdJPa)
- Jerk-Wert für Bremsbereich auslesen (RdJPd)
- Sollposition relativ auslesen (RdDestRel)
- Sollposition absolut auslesen (RdDestAbs)
- Aktuelle Istposition relativ auslesen (RdActPos)
- Aktuelle Positionierzeit auslesen (RdActRuntime)
- Aktuelle Geschwindigkeit auslesen (RdActSpeed)
- Aktuelle Absolutposition auslesen (RdActPosAbs)
- Fehler auslesen (RdError)
- Status auslesen (RdStatus)
- I/O Konfiguration auslesen (RdIoConf)
- Firmwareversion auslesen (RdFwVersion)

4.1.5 Kommandos

Par.3 (rCommand)	Par.4 (rData)
StartMP	Bewegung initialisieren und starten Par.4 = 0: Starten des Ist-Bewegungsprofils cw 1: Starten des Ist-Bewegungsprofils ccw (endlose Bewegung, wenn LdDestRel mit Wert 0xFFFFFFFF (16777215) festgelegt wurde)
Stop	Stoppt die Bewegung (diese Anweisung definiert einen HALT-Fehler im Fehlerregister, der vor der nächsten Bewegungsanweisung durch Cont oder ClrError beseitigt werden muss)
Cont	Beseitigt HALT-Fehler aus Fehlerregister, der beim Fortführen der Stop-Anweisung festgelegt wurde
SingleStep	Einzelschritt mit Ist-Vmin Par.4 . = 0: Richtung cw 1: Richtung ccw
MoveVconst	Positionierung mit Mindestfrequenz Par.4 . = 0: Richtung cw 1: Richtung ccw (endlose Bewegung, wenn LdDestRel mit Wert 0xFFFFFFFF (16777215) festgelegt wurde)
LsRefMove	Positionierung am Endschalter oder Referenzschaltergrenze Par.4 . = 1: LS1 2: LS2 3: REF cw (LS1 oder REF in Richtung ccw suchen) 4: REF ccw (LS2 oder REF in Richtung cw suchen) 5: REF cw in ungültigem Bereich unter LS1(ccw zuerst) 6: REF ccw in ungültigem Bereich über LS2 (cw zuerst) 7: REF cw (direkt in Richtung REF) 8: REF ccw (direkt in Richtung REF)
SaveMP	Bewegungsprofil x speichern (max. 16) Par.4 . = 0: Profil 0 1: Profil 1 .. 15: Profil 15
ActMP	Bewegungsprofil x aktivieren (max. 16) Par.4 . = 0: Profil 0 1: Profil 1 .. 15: Profil 15
ResMC	Neustart der Modul-Firmware, alle Profile werden gelöscht, Fehler werden beseitigt (Par.2 und Par.4 haben keinen Einfluss)
ClrError	Leert Fehlerregister

4.1.6 Schreibbefehle

Par.3 (rCommand)	Par.4 (rData)
LdVmin	Laden der Mindestfrequenz in Hz Par.4 = Wert (2 Bytes) (Bereich: 10...10'000)
LdVmax	Laden der Höchstfrequenz in Hz Par.4 = Wert (2 Bytes) (Bereich: 20...20'000)
LdAcc	Laden der Beschleunigung in kHz/s Par.4 = Wert (2 Bytes) (Bereich: 1...1'000)
LdDec	Laden der Verzögerung in kHz/s Par.4 = Wert (2 Bytes) (Bereich: 1...1'000)
LdJPa	Laden von Jerk-Wert in Prozent für Beschleunigung Par.4 = Wert (2 Bytes) (Bereich: 0...50)
LdJPd	Laden von Jerk-Wert in Prozent für Verzögerung Par.4 = Wert (2 Bytes) (Bereich: 0...50)
LdIoConf	Laden von Input-/Output-Konfigurierung Bit 0: Input LS1 an/aus (0 = an/1 = aus) Bit 1: Input LS2 an/aus Bit 2: Input REF an/aus Bit 3: Input Notstop an/aus Bit 4: Input LS1 Mode (0=Öffner/1=Schliesser) Bit 5: Input LS2-Modus Bit 6: Input REF-Modus Bit 7: Input Notstop Bit 8: Benutzen von TRIG als Input (0)/Output (1) Bit 9: TRIG (0 = TRIG nicht benutzt/1 = TRIG benutzt)
LdDestRel	Laden von relativem Ziel in Schritten Par.4 = Wert (3 Bytes) (Bereich: 0...16777216)
LdDestAbs	Laden von Ziel als absolute Position Par.4 = Wert (4 Bytes vorzeichenbehaftet) (Bereich: -2'147'483'648...+2'147'483'647, aber beachten Sie, dass das relative Ziel kleiner als 16'777'216 ist!) LdDestAbs muss durch MoveVconst oder StartMP gefolgt werden. Durch Stop, SingleStep, LsRefMove wird die Anweisung LdDestAbs abgebrochen.
LdActAbsPos	Laden der absoluten Position für momentane Motorposition Par.4 = Wert (4 Bytes vorzeichenbehaftet) (Bereich: -2'147'483'648...+2'147'483'647)

4.1.7 Lesebefehle

Par.3 (rCommand)	Par.4 (rData)
RdVmin	Gibt die Mindestfrequenz in Hz zurück Par.4 = Wert (2 Bytes)
RdVmax	Gibt die Höchsthfrequenz in Hz zurück Par.4 = Wert (2 Bytes)
RdAcc	Gibt die Beschleunigung in kHz/s zurück Par.4 = Wert (2 Bytes)
RdDec	Gibt Verzögerung in kHz/s zurück Par.4 = Wert (2 Bytes)
RdJPa	Gibt Jerk-Wert in Prozent für Beschleunigung zurück Par.4 = Wert (2 Bytes)
RdJPd	Gibt Jerk-Wert in Prozent für Verzögerung zurück Par.4 = Wert (2 Bytes)
RdIoConf	Gibt Input-/Output-Konfigurierung zurück Par.4 = Wert (2 Bytes) Bit 0: Input LS1 an/aus (0 = an/1 = aus) Bit 1: Input LS2 an/aus Bit 2: Input REF an/aus Bit 3: Input Notstop an/aus Bit 4: Input LS1 Mode (0=Öffner/1=Schliesser) Bit 5: Input LS2-Modus Bit 6: Input REF-Modus Bit 7: Input Notstop Modus Bit 8: Benutzen von TRIG als Input (0)/Output (1) Bit 9: TRIG (0 = TRIG nicht benutzt/1 = TRIG benutzt)
RdStatus	Gibt Modulstatus zurück Bit 0: Status von Input LS1 Bit 1: Status von Input LS2 Bit 2: Status von Input REF Bit 3: Status von Input Notstop Bit 4: Status von Input/Output TRIG Bit 5: Status von Output DIR Bit 6: Status von Output MOTEN Bit 7: - Bit 8: Warten auf TRIG Flag Bit 9: Bei Ziel Flag
RdDestRel	Gibt relatives Ziel in Schritten zurück, das für die nächsten Bewegungsbefehle verwendet wird Par.4 = Wert (3 Bytes)
RdDestAbs	Gibt absolutes Ziel in Schritten zurück, das für die nächsten Bewegungsbefehle verwendet wird Par.4 = Wert (3 Bytes)
RdActSpeed	Gibt Ist-Geschwindigkeit in Hz zurück Par.4 = Wert (2 Bytes)
RdActPos	Gibt Ist-Position in Schritten zurück, die seit Beginn der Bewegung gemacht wurden Par.4 = Wert (3 Bytes)
RdActPosAbs	Gibt die absolute Ist-Position zurück Par.4 = Wert (4 Bytes vorzeichenbehaftet)

RdActRuntime	Gibt die Ist-Laufzeit zurück Par.4 = Wert (3 Bytes)
RdError	<p>Lesefehler Par.4 = Wert (2 Bytes)</p> <p>Byte 0: Fehlercode</p> <ul style="list-style-type: none"> 0: Kein Fehler 1: undefinierter Fehler 2: Kommunikationsfehler 3: Unzulässige Parameter benutzt 4: Input deaktiviert, aber für Bewegung benutzt 5: Keine REF gefunden zwischen LS1 und LS2 6: LS oder REF Bewegung muss zuerst durchgeführt werden, wenn gültiger Bereich verlassen wurde 7: Stop-Anweisung wird fortgeführt 8: EMSTOP ist/war aktiv 9: fortgeführte Anzahl von Schritten entsprechen nicht Vorschau 10: mindestens zwei Parameter für Bewegungsprofil passen nicht zusammen <p>Byte 1: Warn-Code</p> <ul style="list-style-type: none"> 1: Jerk-Wert auf Null gesetzt 2: Vmax nicht erreicht 3: Schritt am Ende des Bewegungsprofils hinzugefügt, um korrekte Anzahl an Schritten zu erreichen 4: Schritte an das Ende des Bewegungsprofils zurückverlegt, um korrekte Anzahl an Schritten zu erreichen 5: Vstop zu gross ($V_{stop} > 2 \cdot V_{min}$) 6: Vmin war grösser als Vmax -> Vmin auf Vmax reduziert 7: mindestens ein Parameter für das Bewegungsprofil ausserhalb des Bereichs 8: Keine Bewegungsparameter nach Modulrückstellung geladen
RdFwVersion	Gibt Firmware-Version zurück Par.4 = Wert (4 Byte/Zeichen) Format x.yy

Statusanzeige auf Moduladresse

Kommuniziert die PCD nicht mit dem Modul (kein H222 FB aktiv), so sind gewisse Statusbits direkt an den Moduladressen als Eingänge zu lesen. Das LSB entspricht der Modulbasisadresse.

Statusbits: Zugriff RO

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	FlagsValid	x	x	x	Error Y	Error X	OnDest Y	OnDest X
MSB															LSB

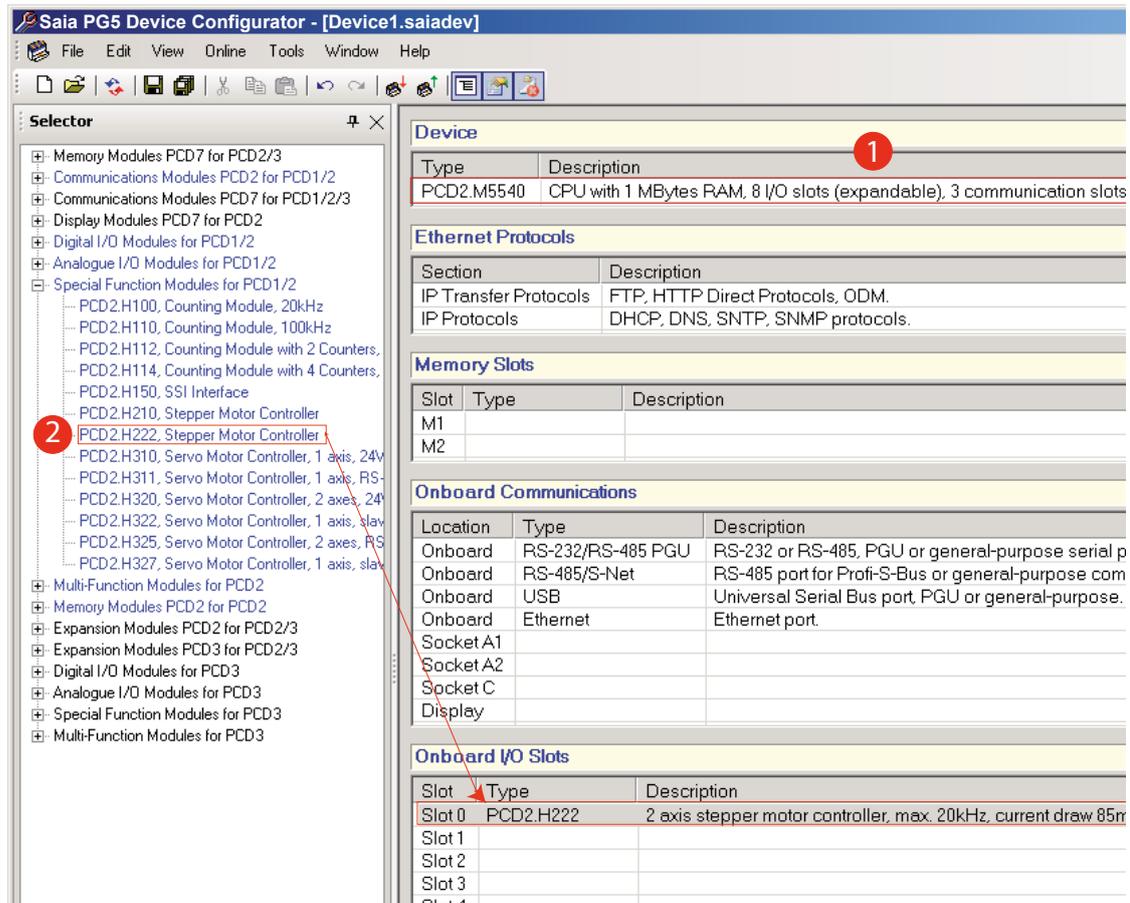


- OnDest X und Y: Diese Bits sind 0 vom Moment wo ein Fahrbefehl abgesetzt wird, bis dieser ausgeführt wurde. Wird ein Fahrbefehl abgesetzt und auf das Trigger-Signal gewartet, so ist das „on destination“ Bit ebenfalls 0.
- Error X und Y: Wird ein Fehler für die jeweilige Achse erkannt, so wird auch das Error-Bit gesetzt.
- FlagsValid: Dieses Bit signalisiert, ob die Flags auf den Input-Adressen 0...7 gültig sind (1) oder nicht (0).

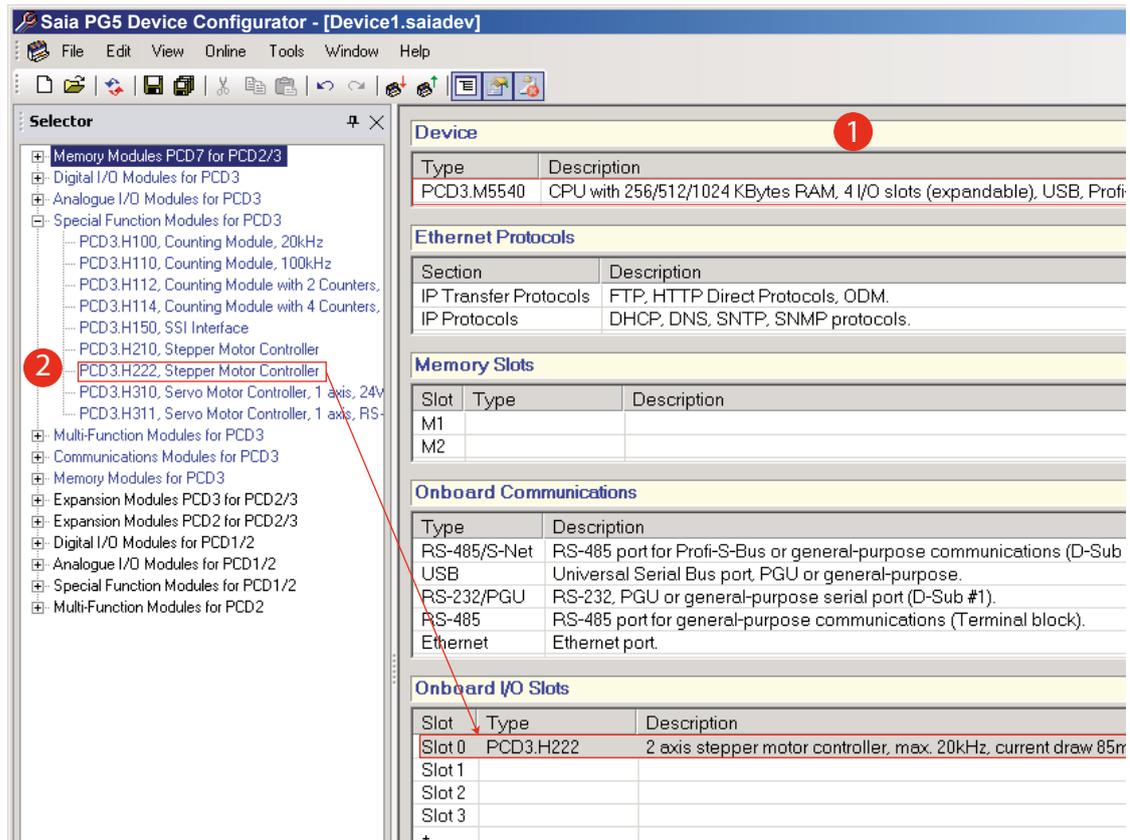
4.2 Systemkonfiguration mit dem Device Configurator

Zu Beginn eines neuen Projektes ist zuerst die Zentralprozessoreinheit (CPU) ❶ und das entsprechende Modul mit dem Gerätekonfigurator (device configurator) auszuwählen. Das Modul ist unter dem Reiter „Spezielle Funktionsmodule für PCD 1/2“ oder „Spezielle Funktionsmodule für PCD 3“ ganz links ❷ zu finden und kann mit „drag and drop“ auf den entsprechenden „slot“ gezogen werden.

Unter dem rechten Reiter ist das Eigenschaftfenster mit dem für die Ein- und Ausgänge der Zugriff zur X- und Y-Achse definiert werden kann. Es besteht die Möglichkeit die Triggerfunktion einem Ein- oder Ausgang zuzuordnen. Mit dieser speziellen Triggerfunktion ist die Bewegung der Achsen konfigurierbar.



Gerätekonfigurator: Beispiel mit PCD2



4

Gerätekonfigurator: Beispiel mit PCD3

4.3 Properties

Die vorgegebene Konfiguration wird hier unter dem rechten Reiter angezeigt:

In dieser Konfiguration kann der Ablauf beider Achsen sowie die Auslösefunktion definiert werden. Wenn ein Endschalter oder ein Referenzschalter erforderlich ist, kann er mit diesen Eigenschaften aktiviert werden.

Properties

Slot 0 : PCD3.H222, Stepper Motor Controller

- General**
 - Base Address: 0
 - Connector Type: Type A, Spring Term
- Power Consumption**
 - Power Consumption 5V [mA]: 85
- Media Mapping Status**
 - Media Mapping For Status Enabled: No
 - Media Type For Status: Register
 - Number Of Media For Status: 4
 - Media Address For Status: 0
 - Base Symbol Definitions For Status: S.IO.Slot0.Status
 - Symbol Definitions For Status: (Custom)
- Media Mapping Position/Speed**
 - Media Type For Position/Speed: Register
 - Number Of Media For Position/Speed: 6
 - Media Address For Position/Speed: 0
 - Base Symbol Definitions For Position/Sp: S.IO.Slot0.SpeedF
 - Symbol Definitions For Position/Speed: (Custom)
- Media Mapping Outputs**
 - Media Mapping For Outputs Enabled: No
- X Axis**
 - Input LS1 Enabled For X Axis: **Yes**
 - Input LS1 Polarity For X Axis: Not Inverted
 - Input LS2 Enabled For X Axis: No
 - Input LS2 Polarity For X Axis: Not Inverted
 - Input REF Enabled For X Axis: No
 - Input REF Polarity For X Axis: Not Inverted
 - Input EMSTOP Enabled For X Axis: No
 - Input EMSTOP Polarity For X Axis: Not Inverted
 - Trigger IO For X Axis: Input
 - Trigger Enabled For X Axis: No
- Y Axis**
 - Input LS1 Enabled For Y Axis: No
 - Input LS1 Polarity For Y Axis: Not Inverted
 - LS2 Input Enabled For Y Axis: No
 - Input LS2 Polarity For Y Axis: Not Inverted
 - Input REF Enabled For Y Axis: No
 - Input REF Polarity For Y Axis: Not Inverted
 - Input EMSTOP Enabled For Y Axis: No
 - Input EMSTOP Polarity For Y Axis: Not Inverted
 - IO Trigger For Y Axis: Input

X Axis

4.4 Befehlssatz Direktzugriff

4.4.1 Schreibbefehle

Grösse	Konstante/ Variable:	Beschreibung:
B	IO.Slot0.IOAccess.X_AXIS_WR_STARTMP	Initialisieren und starten Sie die Fahrbewegung: 0: Starten Sie das aktive Fahrprofil cw 1: Starten Sie das aktive Fahrprofil ccw
B	IO.Slot0.IOAccess.X_AXIS_WR_SINGLESTEP	Einzelschritt mit Vmin: 0: Richtung cw 1: Richtung ccw
B	IO.Slot0.IOAccess.X_AXIS_WR_MOVEVCONST	Schritte mit konstanter Geschwindigkeit 0:Richtung cw 1:Richtung ccw
B	IO.Slot0.IOAccess.X_AXIS_WR_LSREFMOVE	Positionierung am Endschalter oder Referenzschalter: 1: LS1 2: LS2 3: REF cw (suchen nach LS1 oder REF in Richtung ccw) 4: REF cw (suchen nach LS2 oder REF in Richtung ccw) 5: REF cw in ungültigem Bereich unter LS1 (ccw zuerst) 6: REF cw in ungültigem Bereich unter LS2 (cw zuerst) 7: REF cw (direkt in Richtung REF) 8: REF ccw (direkt in Richtung REF)
B	IO.Slot0.IOAccess.X_AXIS_WR_STOP	Stoppt die Fahrbewegung (Diese Anweisung definiert einen HALT-Fehler im Fehlerverzeichnis, der vor der nächsten Bewegungsanweisung durch CONT beseitigt werden muss)
B	IO.Slot0.IOAccess.X_AXIS_WR_CONT	Beseitigt HALT-Fehler im Fehlerregister, der durch die Stop-Anweisung verursacht wurde.
W	IO.Slot0.IOAccess.X_AXIS_WR_VMIN	Mindestfrequenz in Hz laden: Bereich: 10...10'000
W	IO.Slot0.IOAccess.X_AXIS_WR_VMAX	Höchstfrequenz in Hz laden: Bereich: 20...20'000
W	IO.Slot0.IOAccess.X_AXIS_WR_ACC	Beschleunigung in kHz/s laden: Bereich: 1...1'000
W	IO.Slot0.IOAccess.X_AXIS_WR_DEC	Verzögerung in kHz/s laden: Bereich: 1...1'000
W	IO.Slot0.IOAccess.X_AXIS_WR_JERKACC	Jerk-Wert in Prozent für Beschleunigung laden: Bereich: 0...50
W	IO.Slot0.IOAccess.X_AXIS_WR_JERKDEC	Jerk-Wert in Prozent für Verzögerung laden: Bereich: 0...50
B	IO.Slot0.IOAccess.X_AXIS_WR_SAVEMP	Fahrprofil x speichern (max. 16): 0: Profil 0 1: Profil 1 ... 15: Profil 15
DW	IO.Slot0.IOAccess.X_AXIS_WR_DESTREL	Relatives Ziel in Schritten laden: Bereich: 0...16777215 (3 Bytes) (Endlose Bewegung, falls der Wert 0xFFFFF (16777215) gesetzt wird)

DW	IO.Slot0.IOAccess.X_AXIS_WR_DESTABS	Ziel als absolute Position laden: Bereich: -2'147'483'648... +2'147'483'647 (4 Bytes) (Das relative Ziel muss niedriger sein als 16'777'215 Schritte). WR_DESTABS muss durch WR_MOVECONST oder WR_STARTMP gefolgt werden. WR_STOP, WR_SINGLESTEP und WR_LSREFMOVE löscht diese Anweisung.
DW	IO.Slot0.IOAccess.X_AXIS_WR_ACTPOSABS	Absolute Position für momentane Achsposition definieren: Bereich: -2'147'483'648... +2'147'483'647 (4 Bytes)
B	IO.Slot0.IOAccess.X_AXIS_WR_IOCONF	Inputkonfiguration laden: Bit 0: Input LS1 an/aus (0 = an/1 = aus) Bit 1: Input LS2 an/aus Bit 2: Input REF an/aus Bit 3: Input Notstop an/aus Bit 4: Input LS1 Modus (0 = öffnen/1 = schliessen Kont.) Bit 5: Input LS2 Modus Bit 6: Input REF Modus Bit 7: Input Notstop
B	IO.Slot0.IOAccess.X_AXIS_WR_TRIGCONF	Trigger Input-/Outputkonfiguration laden: Bit 0: Benutzen von TRIG als Input (0)/ Output (1) Bit 1: Benutzen von TRIG (0 = TRIG nicht benutzt/1 = TRIG benutzt)
B	IO.Slot0.IOAccess.X_AXIS_WR_ACTMP	Fahrprofil x laden (max. 16): 0: Profil 0 1: Profil 1 ... 15: Profil 15

4.4.2 Lesebefehle

Grösse	Konstante/ Variable:	Beschreibung:
B	IO.Slot0.IOAccess.X_AXIS_RD_IOSR	Bit 7: Reserviert Bit 6: Zustand von Output-MOTEN 0: Achse angehalten 1: Achse in Bewegung Bit 5: Zustand von Output DIR 0: Achse läuft CW 1: Achse läuft CCW Bit 4: Zustand von In-/Output TRIG Bit 3: Zustand des Input-EMSTOP Bit 2: Zustand des REF-Inputs Bit 1: Zustand des LS2-Input Bit 0: Zustand des LS1-Inputs
B	IO.Slot0.IOAccess.X_AXIS_RD_SR	Bit 7:2 Reserviert Bit 1: OnDest Zielmarkierung 0: nicht auf Endposition 1: Endposition erreicht Bit 0: Warten auf TRIG 0: Positionierung läuft, nicht gestartet 1: Positionierung startet, wenn TRIG=1
W	IO.Slot0.IOAccess.X_AXIS_RD_ERR_WARN	Der Lesezugriff sendet den letzten Fehler- und Warningcode aus dem Modul zurück. Nach dem Lesen wird dieser quittiert.
DW	IO.Slot0.IOAccess.X_AXIS_RD_SPEED	Der Lesezugriff gibt die momentane Geschwindigkeit (Hz) zurück.
DW	IO.Slot0.IOAccess.X_AXIS_RD_RUN-TIME	Gibt die aktuelle Positionierzeit (seit Beginn des Fahrauftrags) in ms zurück (3 Bytes)
DW	IO.Slot0.IOAccess.X_AXIS_RD_ACT-POSREL	Der Lesezugriff gibt die Anzahl Schritte, gemacht seit Beginn des Fahrauftrages zurück. (3 Bytes)
W	IO.Slot0.IOAccess.X_AXIS_RD_ACT-POSABS	Der Lesezugriff gibt die momentane Position zurück. (4 Bytes)
W	IO.Slot0.IOAccess.X_AXIS_RD_VMIN	Gibt die Mindestfrequenz in Hz zurück
W	IO.Slot0.IOAccess.X_AXIS_RD_VMAX	Gibt die Höchsthfrequenz in Hz zurück
W	IO.Slot0.IOAccess.X_AXIS_RD_ACC	Gibt die Beschleunigung in kHz/s zurück
W	IO.Slot0.IOAccess.X_AXIS_RD_DEC	Gibt die Verzögerung in kHz/s zurück
W	IO.Slot0.IOAccess.X_AXIS_RD_JER-KACC	Gibt die Beschleunigung in Prozent zurück.
W	IO.Slot0.IOAccess.X_AXIS_RD_JERK-DEC	Gibt die Verzögerung in Prozent zurück.
DW	IO.Slot0.IOAccess.X_AXIS_RD_DEST-REL	Gibt die für Fahrbefehle benutzte Anzahl Schritte zurück (3 Bytes)
DW	IO.Slot0.IOAccess.X_AXIS_RD_DE-STABS	Gibt die absolute Zielposition zurück, welche für den nächsten Fahrbefehl verwendet wird. (4 Bytes)
B	IO.Slot0.IOAccess.X_AXIS_RD_IOCONF	Konfiguration der Eingänge. (Beschreibung siehe WR_IOCONF)
B	IO.Slot0.IOAccess.X_AXIS_RD_TRIG-CONF	Konfiguration des TRIG-Anschlusses. (Beschreibung siehe WR_TRIGCONFIG)
DW	IO.Slot0.IOAccess.RD_MODULEFW-VERSION	Gibt die Firmware-Version im Format x.yy in ASCII zurück.

Zusätzliche Informationen dazu sind im Kapitel 4.3.2 «Lesebefehle».



Abhängig von der Verdrahtung und dem Zustand von LS1 und LS2, können die LED 1 und die LED4 orange leuchten

5 Programmanpassungen bei Ablösung H210 auf H222

Unten aufgelistet die wichtigsten Unterschiede im Programm bei einem Wechsel vom Modul H210 auf H222:

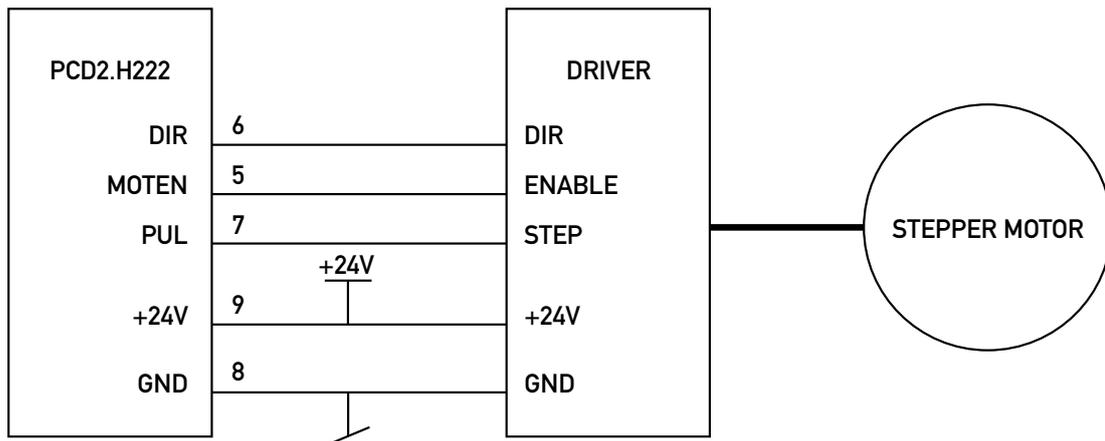
- H222.Init initialisiert nur das Modul nicht aber das Fahrprofil
→ Es ist H222.Init und H222.InitMP nötig, um die Achse in Bewegung zu bringen
- Auswahl von Frequenzbereich entfällt à Vereinfachung von Init
- Vmin, Vmax und Acceleration sind nicht mehr Abhängig vom Frequenzbereich sondern direkt als Frequenzwert zu interpretieren
→ Eine Umrechnung ist zwingend
- End-, Referenz- und Notausschalter müssen mit neuen Kommandos aktiviert werden
- Referenzfahrt wird nicht mit H210.Home sondern H222.Exec und entsprechendem Referenzfahrparameter gestartet. H222.Exec Kommandos und Anzahl Parameter sind nicht identisch mit Kommandos für H210.Exec
→ alle Kommandos müssen überprüft werden

6 Beispiele

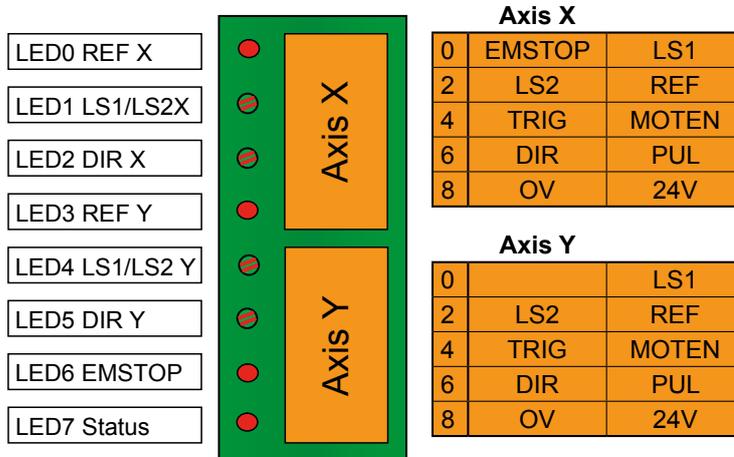
6.1 Hardware PCD2

- PCD2.H222
- PCD2.M5540
- Schrittmotor-Endstufe (Driver)

Dieses Beispiel, zeigt eine einfache Anwendung des PCD2.H222-Moduls:



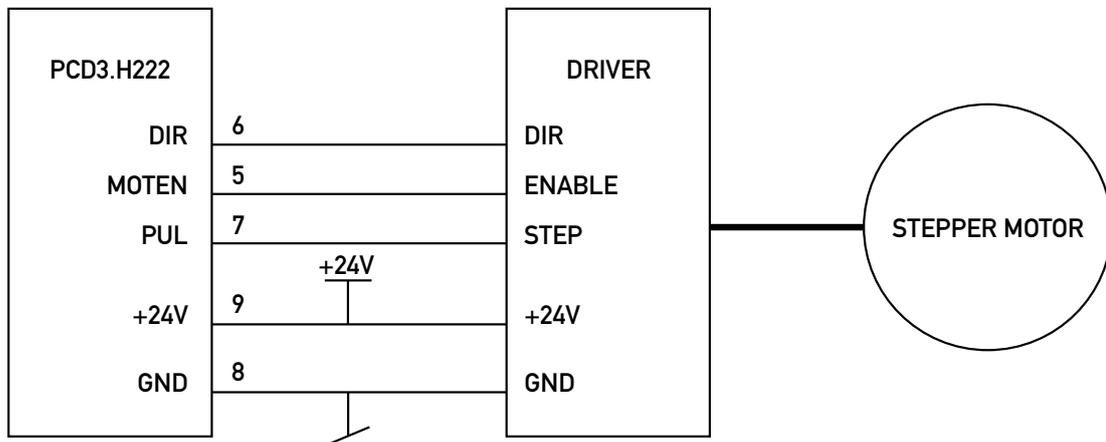
6



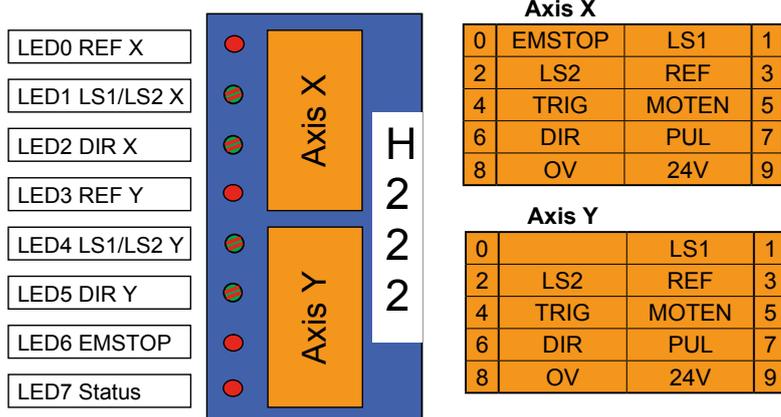
6.2 Hardware PCD3

- PCD3.H222
- PCD3.M5340
- Schrittmotor-Endstufe (Driver)

Dieses Beispiel, zeigt eine einfache Anwendung des PCD3.H222-Moduls:



6



6.3 Programmbeschreibung

Im ersten Teil des IL-Programms werden verschiedene Register definiert, welche im Programmlauf verwendet werden.

Der XOB 16 definiert die Werte für ein Fahrprofil und schreibt diese zum H222 Modul im IO-Slot 0.

Im Hauptprogramm, dem COB 0, wird überprüft, ob sich die Achse auf Endposion befindet. Sobald dies der Fall ist, wird ein neuer Fahrauftrag mit dem vorgegebenen Fahrprofil, der Anzahl Schritte in rPosition und der Drehrichtung in rDir gestartet.

6.4 IL Programmcode

Definitionen

rUmin	EQU	R	; minimal speed
rUmax	EQU	R	; maximal speed
rAccel	EQU	R	; acceleration value
rDecel	EQU	R	; deceleration value
rJerkAcc	EQU	R	; jerk vaule for acceleration
rJerkDec	EQU	R	; jerk vaule for deceleration
rPosition	EQU	R	; number of steps to move
rDir	EQU	R	; 0=Vorwärts, 1=Rückwärts
rStatus	EQU	R	

Initialisierung

```

;*****
;THIS IS A DEMONSTRATION PROGRAM FOR STEPPING MOTOR CONTROLLER
;
      XOB      16

; definiere Motionprofil-Werte X-Achse

      LD      rUmin          ; Umin: 100Hz
              100
      LD      rUmax          ; Umax: 1kHz
              1000
      LD      rAccel         ; acceleration: 100kHz/s
              100
      LD      rDecel        ; deceleration: 10kHz/s
              10
      LD      rJerkAcc       ; max. S-Shape in acceleration
              50
      LD      rJerkDec       ; no S-Shape in deceleration
              0
      LD      rPosition      ; run 5000 steps
              5000
      LD      rDir           ; direction: clockwise
              0

; configure module in Slot0, axis x
      WRPW    IO.Slot0.IOAccess.X_AXIS_WR_UMIN
              rUmin
      WRPW    IO.Slot0.IOAccess.X_AXIS_WR_UMAX
              rUmax
      WRPW    IO.Slot0.IOAccess.X_AXIS_WR_ACC
              rAccel
      WRPW    IO.Slot0.IOAccess.X_AXIS_WR_DEC
              rDecel
      WRPW    IO.Slot0.IOAccess.X_AXIS_WR_JERKACC
              rJerkAcc
      WRPW    IO.Slot0.IOAccess.X_AXIS_WR_JERKDEC
              rJerkDec

; configure module in Slot0, axis y
      WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_UMIN
              rUmin
      WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_UMAX
              rUmax
      WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_ACC
              rAccel
      WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_DEC
              rDecel
      WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_JERKACC
              rJerkAcc
      WRPW    IO.Slot0.IOAccess.Y_AXIS_WR_JERKDEC
              rJerkDec

      EXOB

```

```

-----
;*****
      COB      0
              0

; check and start module in Slot0, axis x
RDPB  IO.Slot0.IOAccess.X_AXIS_RD_SR
      rStatus      ; read modul status register
SHIR  rStatus      ; OnDest flag in Accu
      2
JR    L Y_READY    ; if axis x still in move, jump to Y_READY

WRP   IO.Slot0.IOAccess.X_AXIS_WR_DESTREL
      rPosition    ; set number of steps to go
WRPB  IO.Slot0.IOAccess.X_AXIS_WR_STARTMP ; start motion profile
      rDir         ; set direction (0: clockwise, 1: conterclockwise

; check and start module in Slot0, axis y
Y_READY: RDPB  IO.Slot0.IOAccess.Y_AXIS_RD_SR
      rStatus      ; read modul status register
SHIR  rStatus      ; OnDest flag in Accu
      2
JR    L END_COB    ; if axis y still in move, jump to END_COB

WRP   IO.Slot0.IOAccess.Y_AXIS_WR_DESTREL
      rPosition    ; set number of steps to go
WRPB  IO.Slot0.IOAccess.Y_AXIS_WR_STARTMP ; start motion profile
      rDir         ; set direction (0: clockwise, 1: conterclockwise

END_COB:  ECOB

```

A Anhang

A.1 Icons

	<p>Dieses Symbol verweist den Leser innerhalb eines Handbuches auf weiterführende Informationen in diesem oder einem anderen Handbuch, oder in technischen Informationsbroschüren. In der Regel besteht kein direkter Link zu diesen Dokumenten.</p>
	<p>Dieses Symbol warnt den Leser vor dem Risiko elektrischer Entladung durch Berühren. Empfehlung: Bevor Sie in Kontakt mit elektronischen Bauteilen kommen, sollten Sie zumindest vorher den Minuspol des Systems (Gehäuse der PGU-Buchse) berühren. Besser ist es, permanent mit einer Erdungslasche am Handgelenk mit dem Minuspol verbunden zu sein.</p>
	<p>Dieses Zeichen steht neben Anweisungen, die befolgt werden müssen.</p>
	<p>Erklärungen neben diesem Zeichen sind nur für die Saia PCD® Classic Serie gültig.</p>
	<p>Erklärungen neben diesem Zeichen sind nur für die Saia PCD® xx7 Serie gültig.</p>

A.2 Kontakt

Saia-Burgess Controls AG

Bahnhofstrasse 18
3280 Murten/Switzerland

Telephon +41 26 580 30 00

Fax..... +41 26 580 34 99

E-Mail Support: support@saia-pcd.com

Supportseite: www.sbc-support.com

SBC Seite: www.saia-pcd.com

Internationale Vertretungen &

SBC Verkaufsgesellschaften: www.saia-pcd.com/contact

Postadresse für Rücksendungen von Produkten, durch Kunden des Verkaufs Schweiz:

Saia-Burgess Controls AG

Service Après-Vente
Bahnhofstrasse 18
CH-3280 Murten/Switzerland