

Funktionsbeschreibung

Plug&Drive-Interface

Erweiterung zum technischen Handbuch folgender Produkte:

C5, C5-E, N5, CL3-E, CL4-E, CPB-x, NP5, PDx-C, PDx-E

Inhalt

1	Einleitung.....	3
1.1	Versionshinweise.....	3
1.2	Urheberrecht, Kennzeichnung und Kontakt.....	3
1.3	Hinweise und verwendete Symbole.....	3
1.4	Hervorhebungen im Text.....	4
1.5	Zahlenwerte.....	4
2	Beschreibung der Schnittstelle.....	5
2.1	Aufbau der verwendeten Objekte.....	5
2.2	Steuerkommando <i>PDI-Cmd</i> (2291 _h :04 _h).....	6
2.3	<i>PDI-Status</i> (2292 _h :01 _h).....	13
3	Umsetzung auf dem Feldbus.....	15
3.1	CANopen.....	15
3.2	Modbus.....	15
3.3	EtherNet/IP™	16

1 Einleitung

Mit dem *Plug&Drive-Interface (PDI)* können Sie bei den Antrieben von Nanotec einfache Fahrten realisieren. Das Interface wird von allen Steuerungen und Plug&Drive-Motoren unterstützt und gehört zu ihrer Firmware, unabhängig von der vorhandenen Kommunikationsschnittstelle.

Dieses Dokument beschreibt die Betriebsarten und Funktionen des *Plug&Drive-Interface* sowie die Umsetzung auf dem Feldbus. Für Informationen zur Inbetriebnahme und Bedienung des jeweiligen Nanotec-Produkts nutzen Sie das zugehörige technische Handbuch.

Beachten Sie die Sicherheits- und Warnhinweise sowie die Hinweise zum Gebrauch der Nanotec-Produkte und zur Gewährleistung. Diese finden Sie unter www.nanotec.de in den technischen Handbüchern der Produkte.

1.1 Versionshinweise

Version Handbuch	Datum	Änderungen	Version Firmware
1.0.0	06/2018	erste Veröffentlichung	FIR-v1825
1.0.1	03/2020	Neues Kapitel <u>EtherNet/IP</u>	FIR-v2013

1.2 Urheberrecht, Kennzeichnung und Kontakt

© 2013 – 2020 Nanotec Electronic GmbH & Co. KG. Alle Rechte vorbehalten.

Nanotec Electronic GmbH & Co. KG

Kapellenstraße 6

85622 Feldkirchen

Deutschland

Tel.+49 89 900 686-0

Fax +49 89 900 686-50

www.nanotec.de

1.3 Hinweise und verwendete Symbole

Alle Hinweise sind in einheitlicher Form. Der Grad der Gefährdung wird in die nachfolgenden Klassen eingeteilt.

VORSICHT



Der Hinweis VORSICHT verweist auf eine möglicherweise gefährliche Situation.

Die Missachtung des Hinweises führt **möglicherweise** zu mittelschweren Verletzungen.

► Beschreibt, wie Sie die gefährliche Situation vermeiden.

Hinweis

- Weist auf eine Fehlerquelle oder Verwechslungsgefahr hin.
- Die Missachtung des Hinweises führt **möglicherweise** zu Beschädigungen an diesem Gerät oder anderen Geräten.
- Beschreibt, wie Sie Geräteschäden vermeiden können.

Tipp

Zeigt einen Tipp zur Anwendung oder Aufgabe.

1.4 Hervorhebungen im Text

Im Dokument gelten folgende Konventionen:

Ein *kursiv* hervorgehobener Text markiert benannte Objekte:

- Lesen Sie das *Installationshandbuch*.
- Benutzen Sie die Software *Plug & Drive Studio*, um das Auto-Setup durchzuführen.
- Für Software: Im Tab *Operation* finden Sie die entsprechenden Informationen.

Ein Text in "Anführungszeichen" markiert Benutzereingaben:

- NanoJ-Programm starten durch Beschreiben von Objekt 2300_h , Bit 0 = "1".
- Schreiben Sie den Wert "203B_h" in $2291_h:02_h$ (*PDI-SetValue2*).

1.5 Zahlenwerte

Zahlenwerte werden grundsätzlich in dezimaler Schreibweise angegeben. Sollte eine hexadezimale Notation verwendet werden, wird das mit einem tiefgestellten *h* am Ende der Zahl markiert.

Die Objekte im Objektverzeichnis werden mit Index und Subindex folgendermaßen notiert:

<Index>:<Subindex>

Sowohl der Index als auch der Subindex werden in hexadezimaler Schreibweise angegeben. Sollte kein Subindex notiert sein, gilt der Subindex 00_h .

Beispiel: Der Subindex 5 des Objekts 1003_h wird adressiert mit $1003_h:05_h$, der Subindex 00 des Objekts 6040_h mit 6040_h .

2 Beschreibung der Schnittstelle

Das *Plug&Drive-Interface* stellt eine Nanotec-spezifische Variante zur Ansteuerung eines Antriebs dar und bietet Ihnen eine Alternative zum *Device Profile*, das im CiA 402 beschrieben ist.

Beim *Plug&Drive-Interface* können Kommandos unmittelbar Antriebsbefehle auslösen. Dadurch entfällt die Notwendigkeit, die *State Machine* zu durchlaufen.

2.1 Aufbau der verwendeten Objekte

Das *Plug&Drive-Interface* nutzt zwei Objekte (*PDI-Objekte*), die sechs Einträge umfassen. Ein zusätzliches Objekt dient dazu, das *Plug&Drive-Interface* zu aktivieren.

2.1.1 Objekt zur Aktivierung des *Plug&Drive-Interface*

Über das Objekt 2290_h:00_h (*PDI-Ctrl*) können Sie das *Plug&Drive-Interface* aktivieren. Um das *Plug&Drive-Interface* zu aktivieren, setzen Sie das Bit 0 auf "1".

Objekt-Name	Objekt	Datentyp	Beschreibung
PDI-Ctrl	2290 _h :00 _h	UNSIGNED08	Bit 0: <ul style="list-style-type: none"> ■ Wert = "0": <i>Plug&Drive-Interface</i> deaktiviert ■ Wert = "1": <i>Plug&Drive-Interface</i> aktiviert

Das *Plug&Drive-Interface* ist im Auslieferungszustand der Steuerungen bereits aktiviert.

2.1.2 PDI-Objekte

Beim *Plug&Drive-Interface* ist die komplette Schnittstellenfunktionalität über zwei Record-Objekte abgebildet. Das Objekt 2291_h nutzen Sie für die Eingabe der Parameter. Das Objekt 2292_h wird von der Steuerung für die Ausgabe von Werten genutzt.

Mit dem Objekt 2291_h können Sie den Betriebsmodus wählen und starten sowie die entsprechenden Zielwerte einstellen (Zielposition, Geschwindigkeit usw.). Um Objekte zu Lesen/Schreiben, können Sie mit diesem Objekt auch auf das gesamte Objektverzeichnis (alle Parameter) der Steuerung zugreifen. Das Objekt umfasst das Steuerkommando (*PDI-Cmd*) und drei Einträge (*PDI-SetValue*), die in den einzelnen Betriebsarten unterschiedlich verwendet werden.

Das Objekt 2292_h umfasst den Status (*PDI-Status*) und einen Rückgabewert (*PDI-ReturnValue*), der von der verwendeten Betriebsart abhängt (z. B. den Wert eines Objekts oder einen aktuellen Istwert).

Alle Einträge beider Objekte nutzen den Datentyp SIGNED.

Die nachfolgende Tabelle listet alle Einträge.

Objekt-Name	Objekt-Adresse	Datentyp	Beschreibung
PDI-SetValue1	2291 _h :01 _h	SIGNED32	Wird in den einzelnen Betriebsarten unterschiedlich verwendet, meistens für Zielwerte
PDI-SetValue2	2291 _h :02 _h	SIGNED16	Wird in den einzelnen Betriebsarten unterschiedlich verwendet
PDI-SetValue3	2291 _h :03 _h	SIGNED08	Wird für das Lesen und Schreiben von Objekten

Objekt-Name	Objekt-Adresse	Datentyp	Beschreibung
			verwendet sowie zum Setzen der Homing-Methode
PDI-Cmd	2291 _h :04 _h	SIGNED08	Fungiert als universelles Steuerkommando
PDI-Status	2292 _h :01 _h	SIGNED16	Liefert den Status der Steuerung als Bitmaske
PDI-ReturnValue	2292 _h :02 _h	SIGNED32	Liefert einen Rückgabewert, der von der verwendeten Betriebsart abhängt (z. B. den Wert eines Objekts oder einen aktuellen Istwert)

2.2 Steuerkommando *PDI-Cmd* (2291_h:04_h)

Mit dem *PDI-Cmd* können Sie die Kommandos ausführen, die in der nachfolgenden Tabelle gelistet sind. Um ein bestimmtes Kommando auszuführen, tragen Sie im Objekt 2291_h:04_h den zugehörigen Wert ein. Das Kommando wird unmittelbar ausgeführt.

Kommando	Wert	Beschreibung
NOP (No Operation)	0	Führt nichts aus
Switch Off	1	Schaltet die Regelung ab; Die Leistungsstufe wird gesperrt und der Motor ist stromlos.
Clear Error	2	Löscht einen eventuellen Fehler, vorausgesetzt, die Fehlerursache ist inzwischen behoben
Quickstop	3	Veranlasst einen Schnellstopp des Motors; Der Motor bremsst mit der Quickstop-Rampe auf die Drehzahl 0 ab.
OD-Read	14	Liest ein Objekt aus
OD-Write	15	Schreibt einen Wert in ein Objekt
Auto setup	16	startet das Auto-Setup
Homing on current position	17	Setzt die aktuelle Position auf einen beliebigen Wert
Homing	18	Startet eine Referenzfahrt
Profile Position absolute	20	Startet eine absolute Positionierung
Profile Position relative	21	Startet eine relative Positionierung
Profile Velocity	23	Aktiviert den Drehzahlmodus
Profile Torque	25	Aktiviert den Drehmomentmodus
Halt Bit	Bit 6 (64)	Bremst den Antrieb mit einer Rampe auf Drehzahl 0 herunter und wird von folgenden Kommandos unterstützt: 20, 21, 23 und 25
ToggleCmd Bit	Bit 7 (128)	Durch Setzen/Zurücksetzen (Umschalten) des Bits können Sie ein Kommando mehrmals nacheinander ausführen.

2.2.1 Kommando NOP (No Operation) – PDI-Cmd 0

Dieses Kommando führt nichts aus, sondern löscht nur das bestehende Kommando. Das Bit *PdiStatusToggleCmd* wird trotzdem bedient (umgeschaltet).

2.2.2 Kommando Switch Off – PDI-Cmd 1

Dieses Kommando sperrt die Leistungsstufe des Antriebs. Die Regelung schaltet ab und der Motor ist stromlos. Ein bestehendes Fahrkommando bricht ab und die Regelabweichung wird gelöscht.

Tipp



Um den Antrieb in Regelung zu versetzen, ohne ihn zu bewegen, können Sie ein Kommando 23 (*Profile Velocity*) mit Zielgeschwindigkeit "0" senden oder eine Positionierung (Kommando 20) auf die gegenwärtige Position starten.

2.2.3 Kommando Clear Error – PDI-Cmd 2

Mit diesem Kommando löschen Sie einen eventuellen Fehler, vorausgesetzt, die Fehlerursache ist inzwischen behoben.

Mit *PDI-Status* können Sie am Bit *PdiStatusFault* ablesen, ob aktuell ein Fehler vorliegt.

2.2.4 Kommando Quickstop – PDI-Cmd 3

Mit diesem Kommando veranlassen Sie einen Schnellstopp des Motors. Für weitere Informationen siehe technisches Handbuch der Steuerung, Kapitel *CiA 402 Power State Machine*.

Der Motor bremsst auf die Drehzahl 0 ab. Für diesen Übergang können Sie in Objekt 605A_h eine der folgenden Optionen einstellen:

Wert in Objekt 605A _h	Beschreibung
0	Soforthalt mit anschließendem Zustandswechsel in <i>Switch on disabled</i>
1	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Switch on disabled</i>
2	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Switch on disabled</i>
5	Abbremsen mit <i>slow down ramp</i> (Bremsrampe je nach Betriebsmodus) und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.
6	Abbremsen mit <i>quick stop ramp</i> und anschließendem Zustandswechsel in <i>Quick Stop Active</i> ; Die Regelung schaltet nicht ab und der Motor bleibt bestromt. Sie können wieder in den Zustand <i>Operation enabled</i> schalten.

2.2.5 Kommando OD-Read – PDI-Cmd 14

Mit diesem Kommando können Sie ein beliebiges Objekt aus dem Objektverzeichnis der Steuerung auslesen.

Sie schreiben den Index des Objekts in 2291_h:02_h (*PDI-SetValue2*) und den Subindex in 2291_h:03_h (*PDI-SetValue3*).

Parameter	Objekt-Name	Objekt
Index	PDI-SetValue2	2291 _h :02 _h
Subindex	PDI-SetValue3	2291 _h :03 _h

Anschließend senden Sie das Kommando *OD-Read*.

Der *PDI-ReturnValue* (2292_h:02_h) liefert den Wert des Objekts zurück. Im Fehlerfall setzt die Steuerung das Bit 15 (*PDIStatusError*) des *PDI-Status* (2292_h:01_h) zurück und legt den Fehlercode in *PDI-ReturnValue* ab.

Beispiel

Um das Objekt 3320_h:01_h (Analogeingang 1) auszulesen, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "13088 (3320_h)" in 2291_h:02_h (*PDI-SetValue2*).
2. Schreiben Sie den Wert "1" in 2291_h:03_h (*PDI-SetValue3*).
3. Senden Sie das Kommando *OD-Read*, indem Sie den Wert "14" in 2291_h:04_h (*PDI-Cmd*) schreiben.

Den aktuellen Wert des Analogeingangs können Sie in *PDI-ReturnValue* (2292_h:02_h) lesen.

Um einen zweiten Wert auszulesen, müssen Sie das *ToggleCmd Bit* umschalten.

2.2.6 Kommando OD-Write – PDI-Cmd 15

Mit diesem Kommando können Sie einen Wert in ein Objekt des Objektverzeichnisses der Steuerung schreiben.

Sie schreiben den Index des Objekts in 2291_h:02 (*PDI-SetValue2*) und den Subindex in 2291_h:03_h (*PDI-SetValue3*). Den Wert tragen Sie in 2291_h:01_h (*PDI-SetValue1*) ein.

Parameter	Objekt-Name	Objekt
Index	PDI-SetValue2	2291 _h :02 _h
Subindex	PDI-SetValue3	2291 _h :03 _h
Wert	PDI-SetValue1	2291 _h :01 _h

Anschließend senden Sie das Kommando *OD-Write*.

Im Fehlerfall setzt die Steuerung das Bit 15 (*PDIStatusError*) des *PDI-Status* (2292_h:01_h) zurück und legt den Fehlercode in *PDI-ReturnValue* ab.

Beispiel

Um den Wert "1000" ins Objekt 203B_h:01_h zu schreiben (Nennstrom auf 1000 mA setzen), gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "8251 (203B_h)" in 2291_h:02_h (*PDI-SetValue2*).
2. Schreiben Sie den Wert "1" in 2291_h:03_h (*PDI-SetValue3*).
3. Schreiben Sie den Wert "1000" in 2291_h:01_h (*PDI-SetValue1*).
4. Senden Sie das Kommando *OD-Write*, indem Sie den Wert "15" in 2291_h:04_h (*PDI-Cmd*) schreiben.

Um einen zweiten Wert zu schreiben, müssen Sie das *ToggleCmd Bit* umschalten.

2.2.7 Kommando Auto setup – PDI-Cmd 16

Mit diesem Kommando starten Sie das Auto-Setup. Für weitere Informationen siehe technisches Handbuch der Steuerung, Kapitel *Auto-Setup*.

Das Bit *PdiStatusAutosetupDone* in *PDI-Status* (2292_h:01_h) signalisiert das Ende des Auto-Setups. Wenn Sie ein neues Kommando senden, bleibt dieses Bit gesetzt.

VORSICHT



Unkontrollierte Motorbewegungen!

Das interne Koordinatensystem ist nach dem Auto-Setup nicht mehr gültig. Es kann zu unvorhersehbaren Reaktionen kommen.

- ▶ Starten Sie das Gerät nach einem Auto-Setup neu. Homing alleine genügt nicht.

2.2.8 Kommando Homing on current position – PDI-Cmd 17

Mit diesem Kommando können Sie die aktuelle Istposition auf einen beliebigen Wert setzen. Damit können Sie Referenzfahrten frei programmieren, wenn beispielsweise kein Anschließen eines Referenzschalters möglich ist. Sie können auch nach Einschalten der Steuerspannung einen zuvor gespeicherten Positionswert in den Antrieb schreiben.

Sie tragen die gewünschte Istposition in 2291_h:01_h (*PDI-SetValue1*) ein und führen das Kommando aus, indem Sie den Wert "17" in 2291_h:04_h (*PDI-Cmd*) schreiben.

Das Bit *PdiStatusHomingDone* in *PDI-Status* (2292_h:01_h) signalisiert das Ende des Homings. Wenn Sie ein neues Kommando senden, bleibt dieses Bit gesetzt.

2.2.9 Kommando Homing – PDI-Cmd 18

Mit diesem Kommando können Sie eine Referenzfahrt durchführen. Für weitere Informationen zum Homing-Modus und den Referenzfahrt-Methoden siehe technisches Handbuch der Steuerung, Kapitel *Homing*.

Sie tragen die gewünschte Referenzfahrt-Methode in 2291_h:03_h (*PDI-SetValue3*) ein und führen das Kommando aus, indem Sie den Wert "18" in 2291_h:04_h (*PDI-Cmd*) schreiben.

Das Bit *PdiStatusHomingDone* in *PDI-Status* (2292_h:01_h) signalisiert das Ende des Homings. Wenn Sie ein neues Kommando senden, bleibt dieses Bit gesetzt.

2.2.10 Kommando Profile Position absolute – PDI-Cmd 20

Mit diesem Kommando starten Sie eine absolute Positionierung. Für weitere Informationen zu diesem Betriebsmodus siehe technisches Handbuch der Steuerung, Kapitel *Profile Position*.

Sie können jederzeit einen neuen Fahrauftrag starten: durch Senden des Kommandos *NOP* und erneutes Senden des Kommandos 20 oder durch Umschalten des *ToggleCmd Bit*. Der Antrieb setzt die Fahrt sofort um.

Sie tragen die Zielposition in 2291_h:01_h (*PDI-SetValue1*) und die maximale Geschwindigkeit in 2291_h:02_h (*PDI-SetValue2*) ein.

Parameter	Objekt-Name	Objekt
Zielposition in benutzerdefinierten Einheiten (Werkseinstellung: Zehntelgrad)	PDI-SetValue1	2291 _h :01 _h
Maximale Geschwindigkeit in benutzerdefinierten Einheiten (Werkseinstellung: Umdrehungen pro Minute)	PDI-SetValue2	2291 _h :02 _h

Anschließend senden Sie das Kommando *Profile Position absolute* (20).

Wenn die Zielposition erreicht ist, setzt die Steuerung das Bit *PdiStatusTargetReached* in *PDI-Status* (2292_h:01_h).

Die Steuerung legt die aktuelle Position in *PDI-ReturnValue* ab.

Beispiel

Um eine Fahrt auf Position 2000 mit einer Geschwindigkeit von 200 zu starten, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "2000" in 2291_h:01_h (*PDI-SetValue1*).
2. Schreiben Sie den Wert "200" in 2291_h:02_h (*PDI-SetValue2*).
3. Senden Sie das Kommando *Profile Position absolute*, indem Sie den Wert "20" in 2291_h:04_h (*PDI-Cmd*) schreiben.

Hinweis



Der Eintrag *PDI-SetValue2* ist auf 16 Bit beschränkt. Berücksichtigen Sie das bei der Wahl der Geschwindigkeitseinheit. Für Details lesen Sie das Kapitel *Benutzerdefinierte Einheiten* im technischen Handbuch der Steuerung.

2.2.11 Kommando Profile Position relative – PDI-Cmd 21

Mit diesem Kommando starten Sie eine relative Positionierung. Für weitere Informationen zu diesem Betriebsmodus siehe technisches Handbuch der Steuerung, Kapitel *Profile Position*.

Sie können jederzeit einen neuen Fahrauftrag starten, durch Senden des Kommandos *NOP* und erneutes Senden des Kommandos 21 oder durch Umschalten des *ToggleCmd Bit*. Der Antrieb setzt die Fahrt sofort um.

Sie tragen die Zielposition in 2291_h:01_h (*PDI-SetValue1*) und die maximale Geschwindigkeit in 2291_h:02_h (*PDI-SetValue2*) ein.

Parameter	Objekt-Name	Objekt
Zielposition in benutzerdefinierten Einheiten (Werkseinstellung: Zehntelgrad)	PDI-SetValue1	2291 _h :01 _h
Maximale Geschwindigkeit in benutzerdefinierten Einheiten (Werkseinstellung: Umdrehungen pro Minute)	PDI-SetValue2	2291 _h :02 _h

Anschließend senden Sie das Kommando *Profile Position relative* (21).

Wenn die Zielposition erreicht ist, setzt die Steuerung das Bit *PdiStatusTargetReached* in *PDI-Status* (2292_h:01_h).

Die Steuerung legt die aktuelle Position in *PDI-ReturnValue* ab.

Beispiel

Um eine Fahrt auf Position 2000 mit einer Geschwindigkeit von 200 zu starten, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "2000" in $2291_{\text{h}}:01_{\text{h}}$ (*PDI-SetValue1*).
2. Schreiben Sie den Wert "200" in $2291_{\text{h}}:02_{\text{h}}$ (*PDI-SetValue2*).
3. Senden Sie das Kommando *Profile Position relative*, indem Sie den Wert "21" in $2291_{\text{h}}:04_{\text{h}}$ (*PDI-Cmd*) schreiben.

Hinweis



Der Eintrag *PDI-SetValue2* ist auf 16 Bit beschränkt. Berücksichtigen Sie das bei der Wahl der Geschwindigkeitseinheit. Für Details lesen Sie das Kapitel *Benutzerdefinierte Einheiten* im technischen Handbuch der Steuerung.

2.2.12 Kommando Profile Velocity – PDI-Cmd 23

Mit diesem Kommando aktivieren Sie den Drehzahlmodus (*Profile Velocity*). Für weitere Informationen zu diesem Betriebsmodus siehe technisches Handbuch der Steuerung, Kapitel *Profile Velocity*.

Sie tragen die Zielgeschwindigkeit in $2291_{\text{h}}:01_{\text{h}}$ (*PDI-SetValue1*) ein.

Parameter	Objekt-Name	Objekt
Zielgeschwindigkeit in benutzerdefinierten Einheiten (Werkseinstellung: Umdrehungen pro Minute)	PDI-SetValue1	$2291_{\text{h}}:01_{\text{h}}$

Anschließend senden Sie das Kommando *Profile Velocity* (23).

Die Steuerung legt die aktuelle Geschwindigkeit in *PDI-ReturnValue* ab.

Beispiel

Um eine Fahrt mit Zielgeschwindigkeit 300 zu starten, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "300" in $2291_{\text{h}}:01_{\text{h}}$ (*PDI-SetValue1*).
2. Senden Sie das Kommando *Profile Velocity*, indem Sie den Wert "23" in $2291_{\text{h}}:04_{\text{h}}$ (*PDI-Cmd*) schreiben.

2.2.13 Kommando Profile Torque – PDI-Cmd 25

Mit diesem Kommando aktivieren Sie den Drehmomentmodus (*Profile Torque*). Für weitere Informationen zu diesem Betriebsmodus siehe technisches Handbuch der Steuerung, Kapitel *Profile Torque*.

Sie tragen das Zieldrehmoment in $2291_{\text{h}}:01_{\text{h}}$ (*PDI-SetValue1*) und die maximale Geschwindigkeit in $2291_{\text{h}}:02_{\text{h}}$ (*PDI-SetValue2*) ein.

Parameter	Objekt-Name	Objekt
Zieldrehmoment in Tausendstel des Nenn Drehmoments	PDI-SetValue1	$2291_{\text{h}}:01_{\text{h}}$
Maximale Geschwindigkeit in benutzerdefinierten Einheiten (Werkseinstellung: Umdrehungen pro Minute)	PDI-SetValue2	$2291_{\text{h}}:02_{\text{h}}$

Anschließend senden Sie das Kommando *Profile Torque* (25).

Hinweis



Sie müssen vorher die Rampe für das Drehmoment (Objekt 6087_h, Torque Slope) konfigurieren, da der Vorgabewert "0" ist.

Die Steuerung legt das aktuelle Drehmoment in *PDI-ReturnValue* ab.

Beispiel

Um eine Fahrt mit 50 % des Nenndrehmoments und einer maximaler Geschwindigkeit von 100 zu starten, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "100" in 2291_h:02_h (*PDI-SetValue2*).
2. Schreiben Sie den Wert "500" in 2291_h:01_h (*PDI-SetValue1*). Damit setzen Sie das Zieldrehmoment auf 50 % des Nennstroms.
3. Senden Sie das Kommando *Profile Torque*, indem Sie den Wert "25" in 2291_h:04_h (*PDI-Cmd*) schreiben.

2.2.14 Halt Bit – Bit 6 (PDI-Cmd 64)

Wenn Sie das *Halt-Bit* setzen, bremst der Motor mit einer Rampe auf Drehzahl 0 herunter. Die Regelung bleibt aktiv und der Motor bestromt. Das *Halt-Bit* wird für alle Fahrkommandos unterstützt (*Profile Position* (*absolute/relative*), *Profile Velocity*, *Profile Torque*).

Sie müssen zum Wert des Fahrkommandos den Wert 64 (Bit 6) addieren und das Ergebnis in 2291_h:04_h (*PDI-Cmd*) schreiben.

Um die Fahrt fortzusetzen, haben Sie zwei Möglichkeiten:

- Die Fahrt mit den vorher gesetzten Zielwerten fortsetzen: Dazu müssen Sie das *Halt-Bit* zurücksetzen.
- Einen neuen Zielwert setzen, der beim Zurücksetzen des *Halt-Bits* übernommen werden soll: Dazu müssen Sie das *Halt-Bit* zurücksetzen und gleichzeitig das *ToggleCmd Bit* umschalten.

Beispiel

Um eine absolute Positionierung (Kommando *Profile Position absolute*, 20) anzuhalten und wieder mit der gleichen Zielposition und Zielgeschwindigkeit fortzusetzen, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "84" in 2291_h:04_h (*PDI-Cmd*).
Der Motor bremst ab.
2. Schreiben Sie den Wert "20" in 2291_h:04_h (*PDI-Cmd*).
Der Motor fährt weiter auf die vorher gesetzte Zielposition.

Um die absolute Positionierung anzuhalten und mit einer neuen Zielposition fortzusetzen, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "84" in 2291_h:04_h (*PDI-Cmd*).
Der Motor bremst ab.
2. Tragen Sie die neue Zieposition in 2291_h:01_h (*PDI-SetValue1*) ein.
3. Schreiben Sie den Wert "148" in 2291_h:04_h (*PDI-Cmd*).
Damit wird das *Halt-Bit* zurücksetzt und das *ToggleCmd Bit* umgeschaltet. Der Motor fährt weiter auf die neue Zielposition.

2.2.15 ToggleCmd Bit – Bit 7 (PDI-Cmd 128)

Durch Umschalten (Setzen/Zurücksetzen) des *ToggleCmd Bits* können Sie ein Kommando mehrmals nacheinander ausführen. Jedes Umschalten des *ToggleCmd Bits* bewirkt automatisch das Umschalten des Bits *PdiStatusToggleCmd* in *PDI-Status* (2292_h:01_h).

Beispiel

Um eine relative Positionierung (Kommando *Profile Position relative*) dreimal nacheinander auszuführen, gehen Sie wie folgt vor:

1. Schreiben Sie den Wert "21" in 2291_h:04_h (*PDI-Cmd*).
Die erste Fahrt wird gestartet.
2. Falls gewünscht, ändern Sie die Zielposition oder Zielgeschwindigkeit.
3. Um die zweite Fahrt zu starten, schreiben Sie den Wert "149" in 2291_h:04_h (*PDI-Cmd*).
4. Um die dritte Fahrt zu starten, schreiben Sie den Wert "21" in 2291_h:04_h (*PDI-Cmd*).

2.3 PDI-Status (2292_h:01_h)

Der *PDI-Status* liefert den Status des Antriebs und des *PDI-Interface* als Bitmaske zurück. Details bezüglich des Status des Antriebs finden Sie im Kapitel *CiA 402 Power State Machine* des technischen Handbuchs Ihres Produkts.

Der *PDI-Status* nutzt folgende Bits:

Bit 0: *PdiStatusOperationEnabled*

Dieses Bit wird gesetzt, wenn der Antrieb sich im Zustand *Operation enabled* befindet. Der Antrieb ist in diesem Zustand bestromt.

Bit 1: *PdiStatusWarning*

Dieses Bit wird gesetzt, wenn im Antrieb Warnungen auftreten; Im Objekt 603F_h:00_h können Sie die Warnung auslesen.

Bit 2: *PdiStatusFault*

Dieses Bit wird gesetzt, wenn im Antrieb Fehler auftreten; Im Objekt 603F_h:00_h können Sie den Fehlercode auslesen.

Bit 3: *PdiStatusTargetReached*

Dieses Bit wird gesetzt, wenn der Zielwert (Position, Geschwindigkeit) erreicht ist. Es gibt für den jeweiligen Betriebsmodus ein Toleranzfenster (siehe entsprechendes Kapitel zu dem Betriebsmodus im technischen Handbuch des jeweiligen Nanotec-Produkts).

Bit 4: *PdiStatusFollowingError*

Dieses Bit wird gesetzt, wenn ein Schleppfehler im Modus *Profile Position* oder ein Schlupffehler im Modus *Profile Velocity* aufgetreten ist. Die Grenzen, ab denen ein Fehler registriert wird, stellen Sie ein in den Objekten 6065_h/6066_h (Schleppfehler) bzw. 6078_h/203F_h (Schlupffehler). Die auszuführende Aktion (z. B. Bremsung) definieren Sie in 3700_h (der Vorgabewert entspricht keiner Aktion).

Bit 5: *PdiStatusLimitReached*

Dieses Bit wird gesetzt, wenn ein Limit (Position, Geschwindigkeit) überschritten wird.

Bit 7: *PdiStatusQsHaltBit*

Dieses Bit wird gesetzt, wenn der Antrieb sich im Zustand *Quick stop active* befindet oder wenn das *Halt Bit* gesetzt wurde.

Bit 11: *PdiStatusHomingDone*

Dieses Bit wird gesetzt, wenn ein *Homing* erfolgreich durchgeführt wurde; Das Bit bleibt solange gesetzt, bis Sie die Steuerung ausschalten.

Bit 12: *PdiStatusAutosetupDone*

Dieses Bit wird gesetzt, wenn ein *Auto-Setup* erfolgreich durchgeführt wurde; Das Bit bleibt solange gesetzt, bis Sie die Steuerung ausschalten.

Bit 14: *PdiStatusToggleCmd*

Dieses Bit wird umgeschaltet, wenn ein neues Kommando erkannt wurde. In Verbindung mit dem Bit *PdiStatusError* können Sie erkennen, ob das neue Kommando akzeptiert wurde.

Bit 15: *PdiStatusError*

Dieses Bit wird gesetzt, wenn beim Durchführen eines Kommandos Fehler auftreten. In *PDI-ReturnValue(2292_h:02_h)* können Sie die Fehlercodes auslesen.

3 Umsetzung auf dem Feldbus

Das *Plug&Drive-Interface* funktioniert auf allen Feldbussen, die von den Nanotec-Steuerungen unterstützt werden. Für weitere Informationen zu den Besonderheiten des jeweiligen Feldbusses lesen Sie im technischen Handbuch der Steuerung das entsprechende Kapitel.

Sie brauchen mindestens zwei PDO (Prozessdatenobjekte).

Auf der Empfangsseite mappen Sie folgendes RX-PDO:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
2291 _h :01 _h (<i>PDI-SetValue1</i>)				2291 _h :02 _h (<i>PDI-SetValue2</i>)	2291 _h :03 _h (<i>PDI-SetValue3</i>)	2291 _h :04 _h (<i>PDI-Cmd</i>)	

Auf der Sendeseite mappen Sie folgendes TX-PDO:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
2292 _h :01 _h (<i>PDI-Status</i>)	603F _h :00 _h (<i>Error Code</i>)		2292 _h :02 _h (<i>PDI-ReturnValue</i>)				

Falls Sie weitere Prozesswerte des Antriebs benötigen, können Sie weitere PDO konfigurieren. Nanotec empfiehlt, die Objekte 6040_h (Controlword) und 6060_h (Modes Of Operation) nicht zu mappen, da die Kommandos des *Plug&Drive-Interfaces* diese Objekte überschreiben. Ebenso weitere Objekte, die durch die verwendeten *PDI-Cmd*-Kommandos überschrieben werden, wie Zielposition (607A_h), Geschwindigkeiten (6081_h, 60FF_h) usw..

3.1 CANopen

Bei CANopen empfiehlt Nanotec, die PDO auf azyklische Verarbeitung zu konfigurieren, mit einer realistischen *Inhibit-Time*.

Wenn Sie im Netzwerk einen CAN- bzw. CANopen-Master einsetzen, der keinen NMT-Service unterstützt, gehen Sie nach der PDO-Konfiguration folgendermaßen vor: Setzen Sie das Objekt 1F80_h:00_h bei dem CANopen-Slave auf 08_h. Dadurch wechselt der Slave direkt nach dem Start in den NMT-Zustand *Operational* und die PDOs lassen sich verwenden.

3.2 Modbus

Bei Modbus empfiehlt Nanotec, mit den Funktionscodes 23 (Read/Write Multiple Registers) bzw. 4 (Read Input Register) und 22 (Write Multiple Registers) zu arbeiten. Nur mit diesen Funktionscodes können Sie die gemappten Daten des *Plug&Drive-Interface* in einem Arbeitsschritt übertragen.

Die PDI-Objekte sind im Auslieferungszustand bereits gemappt. Um auf die PDI-Objekte zuzugreifen, nutzen Sie die Modbus-Register mit den folgenden Adressen:

Auf der Empfangsseite (Master → Slave):

Register-Adresse	Objekt
5996	2291 _h :01 _h (<i>PDI-SetValue1</i>)
5997	
5998	2291 _h :02 _h (<i>PDI-SetValue2</i>)
5999	<ul style="list-style-type: none"> ■ Byte 0: 2291_h:03_h (<i>PDI-SetValue3</i>) ■ Byte 1: 2291_h:04_h (<i>PDI-Cmd</i>)

Auf der Sendeseite (Slave → Master):

Register-Adresse	Objekt
4996	2292 _h :01 _h (<i>PDI-Status</i>)
4997	603F _h :00 _h (<i>Error Code</i>)
4998 und 4999	2292 _h :02 _h (<i>PDI-ReturnValue</i>)

Diese Adressen-Bereiche grenzen direkt an die Bereiche der normalen PDO (ab Adresse 6000 für RX-PDO bzw. 5000 für TX-PDO). Sie können mit einem Modbus-Kommando gleichzeitig das *PDI-Interface* bedienen und auf andere gemappten Objekte zugreifen (z. B. mit dem Funktionscode Read/Write Multiple Registers).

3.3 EtherNet/IP™

Firmware-Versionen ab FIR-v2013 erhalten bereits die *PDI-Objekte* in den vorkonfigurierten *Assemblies*.

Weitere Details entnehmen Sie dem Kapitel *Assembly-Objekte* des jeweiligen technischen Handbuchs.