

VORTEX UART Schnittstellendokumentation

Beschreibung der UART Kommandos für Geräte der VORTEX Serie

FREIGABE: KEINE, VORLÄUFIGE VERSION

Dokumentenrevision:

1.0 13.02.2020 HH Ersterstellung

Version	Datum	Betreff	Autor
1.0	13.02.2020	Ersterstellung	HH

Inhalt

Inhaltsverzeichnis

Schnittstellenparameter.....	2
Kommandostruktur	2
Status Abfragen	3
Allgemein.....	3
EEPROM Parameter.....	4
EEPROM Parameter Abfragen	4
EEPROM Parameter setzen	5
EEPROM Kommandos - Konfigurationsparameter speichern/laden	6
Steuerbefehle	6
Lageregler, Referenz, Micro Automation, Gantry.....	6
Micro Automation	7
Micro Automation auslesen/ abfragen	7
Micro Automation setzen.....	7

Schnittstellenparameter

Übertragungsrate: 38400 BAUD
Stopbits: 8N1
Flusssteuerung: Nein

Polling Intervall: nicht schneller als 15ms , damit Controller ausreichend Prozesszeit erhält

Kommandostruktur

Über das UART werden beide Richtungen nur lesbare Zeichen übertragen. Daher kann mit VORTEX Geräten auch via Terminalprogramm gearbeitet werden.

Alle Daten werden in Hexadezimaldarstellung übertragen. Daher wird für 1 Byte Daten, 2 Byte über das UART versendet. (Beispiel Wert 13477 = ,34A5' 2Byte Hex, 4 Byte über UART)

Jeder Kommandostring wird mit Carriage Return abgeschlossen und erst dann durch den Controller ausgeführt!

Kommandos über das UART sind Case Sensitive.

Es gibt nur 2 Arten von Kommandostrings

1. Get Befehle / Abfragen UART String beginnt mit ,?'
2. Set Befehle / Kommandos UART String beginnt mit ,!'

Der Aufbau eines Kommandostrings besteht in der Regel aus::

[GET/SET]+[CMD]+[PARAM1]...+[PARAMn]+CR

CMD kann mehrere Zeichen lang sein.

Für jeden Parameter als Hexbyte werden 2 Zeichen benötigt.

Status Abfragen

Allgemein

CMD	Befehl	Beschreibung	Antwort (Beispiel)
?v	Get Version	Gibt den Versionsstring zurück	vDC5 1.0
?s	Get Status	<p>Gibt den Gesamtstatus zurück. Die Antwort besteht aus folgenden 12 Bytes:</p> <ol style="list-style-type: none"> Schaltzustand Taster, Endlagenschalter, BIT - Bedeutung 0 Taster Rechts gedrückt [Low Active] 1 Taster Links gedrückt [Low Active] 2 Endlage Rechts [Low Active] 3 Endlage Links [Low Active] 4 DIP1 Mode End2End[0]/ Tippbetrieb[1] 5 DIP2 Mode Limit via Ends.[0]/ Überstrom[1] 6 DIP3 Mode Rampen Aktiv[0]/Inaktiv[1] 7 DIP4 Mode Advanced Control [Low Active] Speed Potentiometer (0-255) Current Potentiometer (0-255) Aktuell gemessener Brückenstrom (0-255) relativ zum Nennstrom, z.B. 5A I2T Register 0-100(%) relativ zum durch den I2T Parameter vorgegebenen Schwellwert Aktuelle Position (32Bit) MSB HIGH Aktuelle Position (32Bit) MSB LOW Aktuelle Position (32Bit) LSB HIGH Aktuelle Position (32Bit) LSB LOW Aktueller PWM Output (0-255=0-100%) Motor Status Register BIT - Bedeutung 0 Zielposition erreicht 1 Referenzierung erfolgt 2 Überstrom detektiert 3 Überstromabschaltung aktiv/ Overcurrent Lockout 4 Schleppfehlerabschaltung / Tracking Error Lockout 5 Referenzierungsfahrt aktiv 6 Gantry Modus aktiv / Slave Achse 7 NOT USED Aktueller Schritt der Micro Automation Schrittkette 	sF3D689.... [+18 Zeichen]
?p	Get Position	<p>Gibt die aktuelle Position zurück [32bit = 4 Bytes]</p> <ol style="list-style-type: none"> Aktuelle Position (32Bit) MSB HIGH Aktuelle Position (32Bit) MSB LOW Aktuelle Position (32Bit) LSB HIGH Aktuelle Position (32Bit) LSB LOW 	p00050A03
?i	Read Input Module +[ADR 0..7]	<p>Beispiel gesendet : ?i03</p> <p>Gibt den anliegend Bitwert eines Input Moduls mit der angegebenen Adresse zurück. Hinweis : Im Fehlerfall erfolgt die Antwort parameterlos (nur ,i')</p>	iF2

EEPROM Parameter

EEPROM Parameter Abfragen

CMD	Befehl	Beschreibung	Antwort (Beispiel)
?Pi	Get Parameters I2T	Gibt den (gespeicherten/ aktuellen) Registerwert für Überstromdetektionsschwelle 10% Relativ zu Nennstrom pro s = 0-255 Beispiel 5A nenn 0-500mA/s= 0-255	PiA2
?Pr	Get Parameters Ramps	Gibt (gespeicherte/ aktuelle) Registerwerte für Rampensteuerung zurück. 1. Ramp Slope %PWM(0-255) /0.1s 2. Min PWM. Startwert PWM der Rampe	Pr05F0
?Pm	Get Parameters Motor Control	Gibt (gespeicherte/ aktuelle) Registerwerte für Motorsteuerung zurück. 1. Motor Control Register BIT - Bedeutung 0 NOT USED = 0 1 Drehrichtung Gantry Invertiert 2 Gantry automatisch aktiv/ autostart on powered 3 Drehrichtung Motor Invertiert 4 Zählrichtung Encoder Invertiert 5 Bremse aktiv, wenn Rampe [Modus] abgeschaltet 6 Bremse aktiv, wenn Rampe [Modus] eingeschaltet 7 Hohe [1] oder Niedrige[0] PWM Frequenz 2. Lageregler P-Anteil 3. Lageregler I-Anteil 4. Positionierfenster / erlaubte Abweichung für POS OK[inc] 5. Zeit zur Schleppfehlererkennung in0,1s. z.B. 10=1s ohne Encoderänderung erzeugt Schleppfehler	PmA40501030A
?Pq	Get Parameters Referenzfahrt	Gibt (gespeicherte/ aktuelle) Registerwerte für die Referenzierung zurück. 1. Referenzgeschwindigkeit 1 →Richtung Endlage [0-255 = 0-100% PWM] 2. Referenzgeschwindigkeit 2 →Von der Endlage herunter (nur bei Endlagenschalter, nicht bei Blockfahrt) [0-255 = 0-100% PWM] 3. Reference Control Register BIT - Bedeutung 0 Referenzier- Richtung invertieren --Restbyte nicht verwendet— 4. Offset Position int16 [MSB] 5. Offset Position int16 [LSB] Wert der aktuellen Position nach Referenzfahrt [-16000..+16000 INC] 6. Current Limit [0-255] Stromgrenze während der Referenzfahrt relativ zum Nennstrom(= I/255 * Wert). Wichtig für Blockfahrt!	PqD0A500000F2
?Pg	Get Parameters Gantry	Gibt (gespeicherte/ aktuelle) Registerwerte für denGantry Modus zurück. 1. Gantry Speed, maximal zulässige PWM im Gantry Modus [0-255 = 0-100%] 2. Gantry Current, maximal zulässiger Strom im Gantry Modus [0-255 = 0-100% I(nenn)]	PgFFFF

EEPROM Parameter setzen

CMD	Befehl	Params	Beschreibung	Antwort (Beispiel)
!Pi	Set Parameter I2T	1 Byte	<p>Beispiel gesendet : !PiF3</p> <p>Setzt den Registerwert für Überstromdetektionsschwelle 10% Relativ zu Nennstrom pro s = 0-255</p> <p>Beispiel 5A nenn 0-500mA/s= 0-255</p>	PiF3
!Pr	Set Parameter Ramps	2 Bytes	<p>Beispiel gesendet : !PrA00D</p> <p>Setzt Registerwerte für Rampensteuerung.</p> <ol style="list-style-type: none"> Ramp Slope %PWM(0-255) /0.1s Min PWM. Startwert PWM der Rampe 	PrA00D
!Pm	Set Motor Control Parameters	5 Bytes	<p>Beispiel gesendet : !Pm00030105A0</p> <p>Setzt Registerwerte für Motorsteuerung</p> <ol style="list-style-type: none"> Motor Control Register BIT - Bedeutung 0 NOT USED = 0 1 Drehrichtung Gantry Invertiert 2 Gantry automatisch aktiv/ autostart on powered 3 Drehrichtung Motor Invertiert 4 Zählrichtung Encoder Invertiert 5 Bremse aktiv, wenn Rampe [Modus] abgeschaltet 6 Bremse aktiv, wenn Rampe [Modus] eingeschaltet 7 Hohe [1] oder Niedrige[0] PWM Frequenz Lageregler P-Anteil Lageregler I-Anteil Positionierfenster / erlaubte Abweichung für POS OK[inc] Zeit zur Schleppfehlererkennung in0,1s. z.B. 10=1s ohne Encoderänderung erzeugt Schleppfehler 	Pm00030105A0
!Pq	Set Reference Parameters	6 Bytes	<p>Beispiel gesendet : !PqD0A5000000F2</p> <p>Setzt Registerwerte für die Referenzierung.</p> <ol style="list-style-type: none"> Referenzgeschwindigkeit 1 →Richtung Endlage [0-255 = 0-100% PWM] Referenzgeschwindigkeit 2 →Von der Endlage herunter (nur bei Endlagenschalter, nicht bei Blockfahrt) [0-255 = 0-100% PWM] Reference Control Register BIT - Bedeutung 0 Referenzier- Richtung invertieren --Restbyte nicht verwendet-- Offset Position int16 [MSB] Offset Position int16 [LSB] Wert der aktuellen Position nach Referenzfahrt [-16000..+16000 INC Current Limit [0-255] Stromgrenze während der Referenzfahrt relativ zum Nennstrom(= I/255 * Wert). Wichtig für Blockfahrt! 	PqD0A5000000F2
!Pg	Set Gantry Parameters	2 Bytes	<p>Beispiel gesendet : !PgFFFF</p> <p>Setzt Registerwerte für den Gantry Modus.</p> <ol style="list-style-type: none"> Gantry Speed, maximal zulässige PWM im Gantry Modus [0-255 = 0-100%] Gantry Current, maximal zulässiger Strom im Gantry Modus [0-255 = 0-100% I(nenn)] 	PgFFFF

EEPROM Kommandos - Konfigurationsparameter speichern/laden

CMD	Befehl	Beschreibung	Antwort (Beispiel)
!Es	Save All	Schreibt die aktuellen Werte der Parameter Variablen ins EEPROM. Siehe Tabelle Parameter Kommandos	Es
!Er	Read All	Überschreibt alle Parameter Variablen mit den Daten aus EEPROM	Er
!Ec	Reset To Default	Überschreibt alle Parameter Variablen UND die Daten des EEPROM mit Standardwerten	Ec

Steuerbefehle

Lageregler, Referenz, Micro Automation, Gantry

CMD	Befehl	Params	Beschreibung	Antwort (Beispiel)
!Cz	Zero Current Pos	-	Setzt die aktuelle Position des Antriebs auf Null Stoppt Lageregler, falls aktiv	Cz
!Cs	STOPP	-	Stoppt Lageregler Bricht Referenzfahrt ab Motorstopp	Cs
!Cp	Start Positioning	6 Bytes	Beispiel gesendet : !Cp0000AD03BF0D Startet den Lageregler um den Motor zu positionieren. <ol style="list-style-type: none"> 1. Zielposition (32Bit) MSB HIGH 2. Zielposition (32Bit) MSB LOW 3. Zielposition (32Bit) LSB HIGH 4. Zielposition (32Bit) LSB LOW 5. Max Geschwindigkeit PWM [0-255 = 0-100%] 6. Max Strom [0-255 = 0-100%] relativ zu Nennstrom 	Cp0000AD03BF0D
!Cq	Start Reference	-	Stoppt Lageregler Startet Referenzfahrt	Cq
!Cm	Start Micro Automation Program	1 Byte	Beispiel gesendet : !Cm02 Startet den Micro Automation Programmablauf auf dem Controller, beginnend mit der im Parameter angegeben Schritt ID	Cm
!Cn	Stop Micro Automation Program	-	Stoppt den Micro Automation Programmablauf auf dem Controller.	Cn
!Co	Set Output Module	2 Bytes	Beispiel gesendet : !Co02AD Setzt ein externes Ausgangsmodul mit der angegeben Adresse auf die angegebene Bitkonfiguration <ol style="list-style-type: none"> 1. Adresse [0..7] 2. Wert [0..255] 	Co02AD
!Cge	Enable Gantry	-	Startet den Lageregler im Gantry Modus	Cge
!Cgd	Disable Gantry	-	Stoppt Gantry Modus	Cgd

Micro Automation

Micro Automation auslesen/ abfragen

CMD	Befehl	Beschreibung	Antwort (Beispiel)
?m	Aktuelles Micro Automation Programm auslesen	<p>Gibt das aktuell gespeicherte MA Programm aus</p> <ol style="list-style-type: none"> Anzahl der Datensätze mit einem ‚#‘ davor <i>ACHTUNG: Hier erfolgt bereits ein CR</i> MA Datensatz Bestehend aus folgenden Bytes ‚m‘+ +[BYTE ID] +[BYTE CMD] +[DWORD P32] +[BYTE P1] +[BYTE P2] +[BYTE NXT] +CR → Diese Ausgabe erfolgt für jeden vorhandenen Datensatz, also maximal bis zu 100 mal Funktionsbelegung der Tasten und jeweilige ID des Initialen Datensatzes ‚mx‘ +[BYTE BTNFNC] +[BYTE InitStateS1] +[BYTE InitStateS2] +CR <p>BTNFNC kann folgende Werte annehmen: 0 = None/OFF 1 = Start – Stop 2 = Start – Restart 3 = Hold to run</p>	<p>Micro Automation Programm mit 2 Datensätzen</p> <p>#2 m010700000000053502 m02090000A359037A01 mx010102</p>

Micro Automation setzen

CMD	Befehl	Params	Beschreibung	Antwort (Beispiel)
!Ms	Modify/ Set MA Command	9 Bytes	<p>Beispiel gesendet : !Ms01070000DD6G000203</p> <p>Setzt einen MA Datensatz bestehend aus folgenden Bytes [BYTE ID] +[BYTE CMD] +[DWORD P32]+[BYTE P1]+[BYTE P2] +[BYTE NXT]</p> <p>Der übertragene Datensatz wird direkt im EEPROM gespeichert!</p>	Msm01070000DD6G000203
!Mc	Erase/ Clear MA Program	-	Löscht das MA Programm aus dem EEPROM	Mc
!Mx	Set MA Button Functions	3 Bytes	<p>Beispiel gesendet : !Mx010102</p> <p>Setzt Tasterfunktionen für die Micro Automation Steuerung und den Initialen Datensatz, der mit dem Taster verknüpft wird. [BYTE BTNFNC] +[BYTE InitStateS1] +[BYTE InitStateS2]</p> <p>BTNFNC kann folgende Werte annehmen: 0 = None/OFF 1 = Start – Stop 2 = Start – Restart 3 = Hold to run</p>	Mx010102

