



Servo Controller 3.00

Einleitung

Der UHU Servocontroller 3.00 wurde auf dem Atmel ATtiny2313 implementiert, mit dem Ziel, für selbst aufgebaute Servosteuerungen eine preiswerte Alternative zu kommerziellen Produkten zu bieten. Dazu gehört eine unproblematische logische Integrierbarkeit und eine Benutzerschnittstelle, mit der auf einfache Weise ein möglichst weiter Einsatzbereich parametriert werden kann.

Er stellt eine erweiterte PID Regelung dar, bei der neben den üblichen Proportional- Integral- und Differential-Rückkoppelungen mit einem zusätzlichen Parameter auf Beschleunigung en gegengekoppelt wird. Dies ermöglicht die Stabilisierung schwieriger Regelkreise ohne nennenswerte Schleppfehler auch bei höheren Verfahrensgeschwindigkeiten.

Die Steuerung vom CNC-System erfolgt, wie auch bei Schrittmotorsteuerungen über eine Takt/Richtungs-Schnittstelle.

Der Prozessor kann über eine serielle Schnittstelle mit einem Klartext-Dialog in Echtzeit parametriert werden. Die Betriebsparameter werden im EEPROM des Prozessors geschützt abgelegt und sind immer reproduzierbar. Mehrere Parametersätze ermöglichen den schnellen Einsatz in unterschiedlichen Anwendungsszenarien.

Durch den Einsatz eines speziellen Terminalprogramms kann die Sprungantwort als Fehler über die Zeitachse graphisch dargestellt werden, so dass eine bestmögliche Optimierung der Einstellungen sichergestellt ist.

Es wurde eine Schleppfehlerabschaltung implementiert, die bei einstellbarem Fehlergrenzwert die Endstufe abschaltet. Durch einen Fehler-Eingang können Fehlerabschaltungen auch von anderen Komponenten übernommen werden.

Ein Watchdog schaltet die Endstufe ab, falls durch einen Fehler in der Software oder Parametrierung die Berechnungsroutine länger als 125 ms nicht angesprungen wurde.

Der Controller verarbeitet über 100.000 Schritte/s bei 4-Facher Encoder-Auswertung. Die Verarbeitung erfolgt vollständig im Interruptbetrieb.

Achtung: Maschinen, insbesondere mit automatischen Steuerungen und hohe elektrische Spannungen und Ströme sind potentiell gefährlich.

Aus rechtlichen Gründen wird der UHU-Servocontroller nur zum Aufbau von ungefährlichen Demonstrationsaufbauten abgegeben.

Die Steuerung von Maschinen mit dem Prozessor unterliegt der persönlichen Verantwortung des Anwenders, solange sie von mir nicht explizit freigegeben wurde.

Jede Haftung oder Gewährleistung wird hiermit explizit ausgeschlossen.

Voraussetzungen

Platine

Natürlich benötigt der Prozessor eine geeignete Platine mit einem sorgfältig aufgebauten Logik- und Leistungsteil. Die zurzeit wohl beste Platine stammt von ‚Stefan8051‘ aus Peters CNC-Ecke. Ein eigener Entwurf dürfte sich wohl nur lohnen, wenn besondere Eigenheiten berücksichtigt werden müssen.

Trotzdem ein paar Informationen zur Implementierung:

Der Prozessor setzt eine Endstufe voraus, zu dem Impulsteil der in Peters CNC-Ecke veröffentlichten 'Selbstbausteuerung' kompatibel ist. Dabei wird an PortB/3 (Pin15) des Prozessors ein 20 khz mittensymmetrisches PWM-Signal ausgegeben der der Strom der Endstufe über PortB/4 (Pin 16) freigegeben. Das PWM-Signal hat einen Regelbereich von 15% bis 85%, da die gewählten Mos-Fet Treiber IR2104 bzw. IR2184 für die eingebaute Ladungspumpe eine minimale Signalbreite voraussetzen. Dies bedeutet, dass die Eingangsspannung der Endstufe bei der ca. 1.5 fachen Motor -Nennspannung liegen muss, um die volle Leistung auszuschöpfen.

Optional können eine Betriebs-LED an PortB/5 und Fehler-LED an PortB/6 angeschlossen werden.

Eingänge:

Schritt:	PortD/2 Pin 6
Richtung:	PortD/3 Pin 7
Encoder A	PortB/0 Pin 12
Encoder B	PortB/1 Pin 13

Für die Verbindung zum PC wird eine Potentialtrennung über Optokoppler dringend empfohlen. Hier müssen unbedingt Typen zum Einsatz kommen, deren Ansprechstrom vom PC-Parallelport erbracht werden kann und deren max. Taktfrequenz deutlich über der gewünschten Schrittfrequenz liegt (10 -fach!). Gute Erfahrungen wurden mit den HCPL7601 oder HCPL22xx von HP/Agilent gemacht.

Für die Anpassung der serieller Schnittstelle an die Pegel der am PC vorhandenen RS232/V.24 kann eine einfache Schaltung über Widerstände und eine Invertierung der Signale über ein freies Gatter (z.B. CD 40106 oder 74HC14) zum Einsatz kommen

Soll eine eigene Platine entwickelt werden, ist in jedem Fall ein RS232 Pegelwandler vorzuziehen, beispielsweise ein MAX202.

Um die Geschwindigkeit der seriellen Schnittstelle mit PC-Systemen kompatibel zu halten, muss der Servocontroller der richtigen Taktfrequenz (24 MHz) betrieben werden.

Bei Bedarf kann zusätzlich ein Fehler -Eingang an PortB/2 Pin 14 und Fehler-Ausgang an PortB/5 Pin 17 beschaltet werden.

Der Nothalt-Eingang muss im Normalbetrieb auf 5V liegen. Wird er auf Masse gezogen, erfolgt eine Abschaltung der Endstufe im nächsten Berechnungsintervall. Die Abschaltung über den Nothalt-Eingang aktiviert dabei die Fehler-LED nicht. Die Endstufe kann nur durch einen Hard - oder Soft-Reset wieder aktiviert werden.

Durch die vorgeschlagene Diode im Ausgang können mehrere Servocontroller mit ihren Fehlereingängen und Fehlerausgängen parallel geschaltet werden (wired-or).

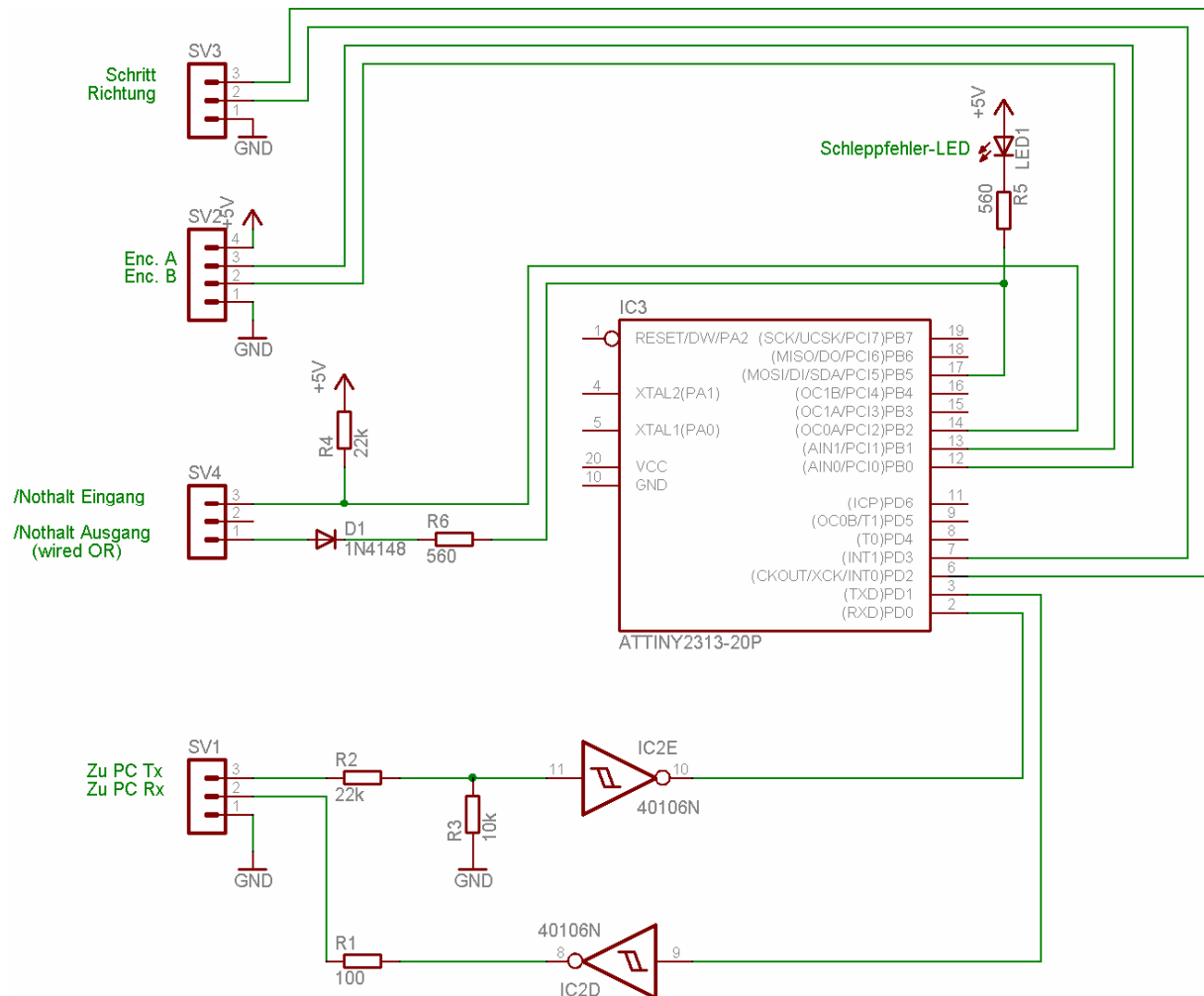
Soll der Nothalt-Eingang nicht verwendet werden, ist Pin 14 des Prozessors über eine 4k7 Widerstand auf +5V zu legen.

Da der AVR recht empfindlich auf elektrische Störungen reagiert, sind keramische Kondensatoren mit 100 nF über die Stromversorgung der ICs und parallel zu den ELKOS unverzichtbar. In neuen Layouts sollten sie auf keinen Fall vergessen werden. Generell erfordert die räumliche Nähe von schneller Logik und hohen getakteten Strömen ein EMV-gerechtes Design und sorgfältigen Aufbau.

Die meisten bislang aufgetretenen Probleme haben ihre Ursache in unsachgemäßem Layout oder schlampiger Verdrahtung insbesondere auf den Leitungen zu den Encodern und zum PC.

Trotz des im Tiny2313 eingebauten, guten Reset-Algorithmus und des Brown-Out-Detektors scheint die übliche Beschaltung des Reset-Einganges mit Widerstand und Kondensator nicht ideal, da sie unter bestimmten Bedingungen (z.B. besonders langsam steigende Spannung) versagt.

Es wird dringend empfohlen, statt der R-C-Kombination am Reset-Pin ein Resetbaustein wie z.B. den TL7757CLP oder MC34064P-5 oder MC33064P-5 einzubauen. Das ist mit ca. 1 Euro nicht teuer und bietet zusätzliche Sicherheit gegen Verlust der im EEPROM abgelegten Parameter. Der Baustein muss eine Schwellenspannung von min. 4.5V bis max. 5V haben, da der Tiny nur ab 4.5V stabil arbeitet. Die Betriebsspannung darf nie über 5.5V steigen.



Der Schaltplanausschnitt zeigt notwendige Modifikationen der ‚alten‘ Selbstbausteuerungsplatine. Der Schaltplan der vollständigen Steuerung wurde von ‚Stefan8051‘ in Peters CNC -Ecke veröffentlicht.

Terminalprogramm

Für den Betrieb des Terminalprogrammes Serv200.exe ist ein PC mit einer seriellen Schnittstelle com1 und DOS erforderlich. Alternativ kann das Programm im Win98 MS -DOS Modus gestartet werden. Ein Betrieb unter Win2000, WinNT oder WinXP ist nicht möglich.

Arbeitsweise

Der Servocontroller verrechnet Positions-Soll-Werte die er über die Schritt/Richtungs-Schnittstelle bekommt und die vom Encoder kommenden Positions-Ist-Werte unter Verwendung von konfigurierbaren Parametern zu einer Stellgröße, die als PWM-Signal ausgegeben wird. Diese steuert den Strom und dessen Richtung durch die Endstufe und dreht den Motor so lange, bis die Differenz ausgeglichen und die Sollposition erreicht ist. Der Regelalgorithmus sorgt für die Stabilität des Regelkreises und schnelles und präzises Ausregeln von Positionen und Positionsveränderungen. Er ist eine Erweiterung der PID-Regelung, bei der zusätzlich mit Beschleunigungswerten gegengekoppelt werden kann.

Das Intervall für den Berechnungszyklus ist einstellbar und wird unabhängig von den Betriebszuständen als festes Raster eingehalten. Dies garantiert einen verlässlichen Regelbetrieb.

Alle Eingangssignale werden durch HW-Interrupts erkannt und hierarchisch abgearbeitet.

Einstellungen

Die Parametrierung erfolgt ausschließlich über die serielle Schnittstelle. Idealerweise wird dafür das DOS-Programm Serv200.exe verwendet, das die nötige Terminalemulation für die COM1-Schnittstelle sowie eine grafische Anzeige der Sprungantwort bereitstellt.

Sollte ein eigenes Terminalprogramm zum Einsatz kommen, können natürlich nur die Parameter gesetzt/ausgelesen werden. Die Schnittstellenparameter dafür sind 38400,N,8,1 ohne Handshake.

Für den Normalbetrieb wird die serielle Schnittstelle nicht benötigt.

Da gesteuerte Maschinen potentiell gefährlich sind, ist dringend anzuraten, den Teil 'Einstellungen' komplett durchzulesen, bevor eine Steuerung mit dem Prozessor in Betrieb gesetzt wird.

Im Terminalbetrieb erfolgen alle Eingaben in einer Kommandozeile, die immer mit einer spitzen Klammer beginnt. Jede Eingabezeile wird mit der Enter-Taste, hier dargestellt als `>`, abgeschlossen.

```
UHU Servo Controller 3.00 (c) Snr: 134
EEPROM: OK
Hilfe mit (?)
>
```

Es wird grundsätzlich ein Buchstabe eingegeben, der die jeweilige Aktion oder den zu verändernden Parameter angibt. Die verfügbaren Kommandos und die dahinter stehenden Parameter können mit der Hilfefunktion durch Eingabe von '?' oder '#' abgezeigt werden.

Hier ein Beispiel:

```
(P)rop.      900
(I)nt.       24
(D)iff.      260
(B)eschl.    400
(G)eschw.    300
(M)oment     255
(T)raegh.    150
(F)ehler     2000
(V)erv.       0
(W)rap        0
(N)euber.     50
(L)esen      [0,1,2]
(S)peich.    [0,1,2]
(E)ilg.      [Wert]
(Z)aehl.
(R)eset
>
```

Die Veränderung von Parametern erfolgt dann durch Eingabe des Buchstabens, direkt gefolgt vom Konfigurationswert und wird vom Controller mit 'OK' bestätigt.

```
> P600 ?
OK
>
```

Gibt man diesen nicht an, wird der aktuelle Konfigurationswert dargestellt, mit der Möglichkeit einer neuen Eingabe:

```
> P ?
(P)rop.      900: 600?
OK
>
```

Eingabewerte sind grundsätzlich ganzzahlig.

Intern werden die Regelungskonstanten als 16 Bit Festkommazahl behandelt. dabei geben die ersten 8 Byte den Wert vor dem Komma, die zweiten 8 Byte den Wert hinter dem Komma als Vielfache von 1/256.

Faktisch kann man bei man auf diese Feinheiten verzichten, wenn man den Regler 'nach Gefühl' einstellt. Hier ist es schneller und einfacher, die Werte als entsprechend große, ganze Zahlen zu betrachten

Proportional-Konstante (P)

Sie erzeugt eine Regelgröße, die wie der Name schon sagt, proportional zu der aktuelle Wegdifferenz ist. Je größer der Wert, desto ‚schärfer‘ reagiert die Steuerung auf neue Soll-Positionen.

Zu hoher Werte führen zu Schwingungen niedriger Frequenz und konstanter oder langsam steigender oder fallender Amplitude.

Sinnvolle Werte liegen bei ca.50 bis ca. 5000, je nach Auflösung des Encoders und Dynamik des Motors.

Integral-Konstante (I)

Sie erzeugt eine Regelgröße, die mit jedem nicht ausgeglichenen Fehler über die Zeit anwächst. So wird sichergestellt, dass auch ein kleiner Fehler (beispielsweise eine einzige Encoderposition) ausgeregelt wird, weil mit jedem Berechnungszyklus die Regelgröße um I/256 der angegebenen I-Konstante erhöht wird, bis es zum Anlauf des Motors ausreicht.

Zu hohe Werte führen zu Schwingungen hoher Frequenz und schnell wachsender Amplitude.

Sinnvolle Werte liegen bei 1 bis ca. 120, je nach Auflösung des Encoders und Dynamik des Motors.

Achtung: Die Wirkung des I-Parameters wurde gegenüber der vorherigen Versionen um einen Faktor 4 reduziert (Eingegebene 40 entspricht 10 bei der Version 2.02t). Durch diese Skalierung kann jetzt der meist sehr kleine Wert genauer und einfacher eingestellt werden.

Differential-Konstante (D)

Sie erzeugt eine Gegenkopplung, die proportional zur aktuellen Geschwindigkeit ist.

Dies stabilisiert den Regelkreis, führt jedoch zu temporären Abweichungen von der Idealposition während schneller Positionsveränderungen (Schleppfehler). Braucht man die Stabilisierung nicht, ist ein Wert von 255 einzutragen. Grundsätzlich ist der D-Parameter zur Stabilisierung vorzuziehen.

Zu niedrige Werte führen zu Schwingungen höherer Frequenz und der Tendenz zur Selbstverstärkung. Sinnvolle Werte liegen bei ca. 255 bis ca. 2000 je nach Auflösung des Encoders und Dynamik des Motors.

Beschleunigungs-Konstante (B)

Sie erzeugt eine Gegenkopplung, die proportional zu der aktuellen Geschwindigkeitsdifferenz der Soll- und Ist-Werte im letzten Berechnungsintervall ist. Der Parameter wird zur Stabilisierung des Regelkreises eingesetzt und bringt üblicherweise bessere Resultate als ein hoher D-Parameter.

Ein hoher B-Wert kompensiert bei oberflächlicher Betrachtung nur einen hohen P-Wert, wirkt sich tatsächlich aber stärker auf die Dynamik aus. Daher ist ein zu hoher B-Wert nicht wünschenswert.

Es ist ein Optimum anzustreben, bei dem die Sprungantwort die geringste Abweichung von der Nulllinie hat.

Sinnvolle Werte liegen bei ca. 200 bis ca. 20000, je nach Auflösung des Encoders und Dynamik des Motors.

Die Wirkung des B-Parameters wurde gegenüber der Version 2.02t deutlich erhöht.

Maximalgeschwindigkeit (G)

Der Parameter begrenzt die maximale Geschwindigkeit und ist nur für spezielle Anwendungen interessant.

Die normale Einstellung von 1000 reicht in den meisten Fällen aus und sollte nicht verändert werden.

Sinnvolle Werte liegen bei ca. 0 bis ca. 5000, je nach Auflösung des Encoders und Dynamik des Motors. Zu geringe Werte führen zum zu frühen Ansprechen der Schleppfehlerabschaltung.

Maximalmoment (M)

Der Parameter begrenzt den maximalen Motorstrom durch Begrenzung des PWM Tastverhältnisses. Dies kann für den Einsatz niedrigerer Motoren an höheren Spannungen verwendet werden.

Vorsicht: Wird ein Motor verwendet, der nicht für die Betriebsspannung geeignet ist, zerstört ein falsch eingetragener M-Wert den Motor oder die Endstufe. Idealerweise bleibt dieser Wert unverändert.

Sinnvolle Werte liegen bei ca. 50 bis max. 255 (entspricht vollem Strom). Bei passender Motor-Betriebsspannung sollte bei der Einstellung mit 255 begonnen und nur bei unkontrollierbaren Schwingungen in kleinen Schritten reduziert werden.

Systemträgheit (T)

Der Parameter beeinflusst die Dynamik der Gesamt-Regelgröße. Er sollte nur für das Feintuning verwendet werden und wird meist nicht benötigt.

Höhere Werte führen zu 'schärferen' Regelcharakteristiken. Die Wirksamkeit nimmt bei hohen P- und B-Werten deutlich ab.

Sinnvolle Werte liegen bei ca. 50 bis 1000

Fehlerabschaltung (F)

Der Parameter begrenzt die maximal zulässige Abweichung in 4-fach ausgewerteten Encoderschritten von Soll- und Ist-Position. Wird sie überschritten, macht der Prozessor die Endstufe stromlos und aktiviert die Fehler-LED. Der Strom kann durch einen harten Reset (Stromabschaltung oder Reset-Taster) oder einen Soft-Reset (R) wieder aktiviert werden.

Falls ein Nothalt-Ausgang an Pin 17 beschaltet ist, können benachbarte Systeme mit abgeschaltet werden.

Niedrigere Werte sind besser, höhere riskanter.

Sinnvolle Werte liegen bei ca. 100 bis 5000.

Schrittvervielfacher (V)

Dieser Wert wird bei jedem Eingangsschritt vom PC zusätzlich auf die Soll-Position addiert. Ist z.B. 2 eingestellt, fährt die Steuerung bei jedem Schritt vom PC genau 3 Schritte vom Encoder.

Damit können Systeme mit hoher Encoderauflösung auf höhere Geschwindigkeiten gebracht werden, obwohl der steuernde Rechner nur vergleichsweise geringe Schritt-Raten erzeugen kann.

Braucht man die Auflösung der Encoder nicht, ist es immer sinnvoll, diesen Wert hoch zu setzen, da dadurch der Prozessor deutlich entlastet wird.

Vorsicht: Eine hohe Schrittvervielfachung führt definitionsgemäß zu größeren Schleppfehlern innerhalb eines Abtastintervalls. Dies ist bei Einstellung des F-Wertes zu berücksichtigen.

Grundeinstellung ist 0, d.h. jeder Eingangsschritt entspricht einem 4 -fach ausgewerteten Encodertakt. Sinnvolle Werte liegen bei 0 bis max. 20

Neuberechnungsintervall (N)

Dieser Parameter steuert das Intervall der Neuberechnungen der PID Regelgröße.

Grundeinstellung ist 63 (Intervall 666µs bzw. 1.5 kHz).

Je kleiner der Wert, desto kürzer ist das Intervall und desto kleiner sind die in die PID -Berechnung einfließenden Soll-Ist-Differenzen. Damit können kritische Systeme mit hoher Encoder -Auflösung besser stabilisiert werden, da insbesondere der I-Wert weniger Auswirkung auf die Regelgröße bekommt.

Größere Werte entlasten den Prozessor und sind bei Systemen geringerer Dynamik leichter einzustellen.

Da die Geschwindigkeiten als Positionsdifferenzen über ein Neuberechnungsintervall ermittelt werden, ändert sich mit diesem Parameter auch die Wirkung der P-, I-, D-, B-, G- und T-Konstanten.

Erlaubte Werte liegen bei min 30 bis max. 200 (entspricht 0.3 ms bis 2 ms).

Achtung: Zu kleine Werte führen zu einer sequenziellen Abarbeitung ohne sicheres Zeitraster und damit zu nicht verlässlichen Regelungseigenschaften.

'Wrap-Zähler' (W)

Dies ist kein Regelungsparameter, sondern ein Zähler für nicht erkannte Encoder -Bewegungen (Abweichungen vom Gray-Code des Encoders / Wechsel der Polarität beider Eingangssignale zwischen zwei Abtastungen. Die Ursache dafür kann sein:

- ? zu hohe Verfahrgeschwindigkeit (unwahrscheinlich bei üblichen Konfigurationen)
- ? unsaubere Flanken oder gestörte Eingangssignale
- ? defekter Encoder oder defekter Eingangs-Pin des Prozessors.

Der Zähler kann durch die Eingabe von '0' rückgesetzt werden.

Abspeichern der Konfigurationen

Alle Parameter können im EEPROM abgespeichert werden. Dazu stehen 3 Parametersätze zur Verfügung. Sie sind mit 0,1 und 2 bezeichnet können wahlfrei belegt werden. Die Werte werden mit einer Prüfsumme und einer Forward-Error-Correction versehen, um bei Datenverlust keine unvorhersehbaren Reaktionen der Regelung zu bekommen.

Beim Start des Controllers wird immer der Parametersatz '0' aktiviert.

Schreiben (S)

erlaubt sind S0, S1 und S2 für die drei Parametersätze.

Beim Schreiben wird immer die Prüfsumme für den Parametersatz neu berechnet und abgelegt. Eine inhaltliche Prüfung der Werte selbst erfolgt nicht.

Vorsicht: Das bedeutet, dass auch grob fehlerhafte Parameter mit S0 abgespeichert beim Start des Controllers aktiv werden.

Es macht Sinn, eine erprobte und unter S0 abgelegte Konfiguration auch unter S1 oder S2 abzuspeichern, damit man bei Änderungen der Konfiguration die alten Werte wiederherstellen kann.

Lesen (L)

erlaubt sind L0, L1 und L2 für die drei Parametersätze.

Wenn trotz Forward-Error-Correction die Prüfsumme nicht korrekt ist, wird eine Fehlermeldung ausgegeben, der Strom der Endstufe abgeschaltet und die Fehler-LED aktiviert.

Weitere Funktionen

Folgende weitere Funktionen sind über die serielle Schnittstelle aktivierbar:

Eilgang (E)

Mit dieser Funktion kann die Eingabe einer neuen Soll-Position Encoderschritten direkt erfolgen. Der Prozessor steuert dann zu dieser Position mit der unter (G) eingegebenen maximalen Geschwindigkeit. Dadurch wird definitionsgemäß ein Fehler in Höhe der Positionsänderung erzeugt. Ist die eingegebene Fehlergrenze kleiner als die Positionsdifferenz erfolgt eine Fehlerabschaltung der Endstufe und Aktivierung der Fehler-LED.

Die Funktion wurde nur zur Unterstützung einer künftigen Auto-Tuning-Funktion implementiert und hat im Moment keine Bedeutung.

Zulässige Eingaben sind E-32000 bis E+32000

Soft-Reset (R)

Die Eingabe von 'R' (ohne Parameter) setzt den Controller mit dem aktuellen (ungespeicherten) Parametersatz zurück. Die Positionszähler und der Wrap-Zähler werden genullt.

Der Strom für die Endstufe wird eingeschaltet und die Fehler-LED gelöscht. Der Analysemodus wird deaktiviert.

Der Befehl wird mit 'OK' quittiert.

Hard-Reset (+)

Die Eingabe von '+' (ohne Parameter) startet den Controller wie durch einen echten Reset oder Unterbrechen der Stromzufuhr. Es wird der Parametersatz 0 ausgelesen und aktiviert.

Der Strom für die Endstufe wird eingeschaltet und die Fehler-LED gelöscht. Der Analysemodus wird deaktiviert.

Analysemodus starten (A)

Die Funktion wird ausschließlich für das Terminalprogramm Serv2000.exe benötigt. Sie darf nicht von einem normalen ASCII-Terminal aus gestartet werden, da sonst die laufenden übertragenen Zeichen die Kommunikation verhindern. Falls jemand eine eigene Software schreiben möchte (z.B. für 32Bit Windows... ;-), hier nur die groben Eckwerte. Details oder DOS-Code in Pascal gerne auf Anfrage.

Die Eingabe von 'A' (ohne Parameter) startet die Ausgabe von Schleppfehlerwerten mit einem Zeitraster das dem Fünffachen des Neuberechnungsintervalls entspricht.

Die Schleppfehler werden als einzelne 8-Bit Zeichen ausgegeben, wobei ein gesetztes Bit 7 signalisiert, daß es sich nicht um ein ASCII-Zeichen handelt. Der Wert ist also ein 7-Bit-Wert (0..127) mit einem Nullwert bei 63 und durch das gesetzte Bit 7 das Ganze um 128 nach oben geschoben.

Der Befehl wird nicht quittiert.

Analysemodus beenden (X)

Schaltet den Analysemodus ab. Praktisch, wenn man ihn versehentlich aktiviert hat....

Inbetriebnahme

Terminalprogramm

Das Terminalprogramm Serv200.exe ist ein reines DOS-Programm und nicht unter Windows lauffähig, da es auf die Hardware des Videocontrollers und die serielle Schnittstelle zugreift. Es wird zusammen mit der Datei egavge.exe in ein beliebiges Verzeichnis kopiert und von dort gestartet. Einstellungen sind nicht erforderlich. Im Moment wird ausschließlich COM1 unterstützt.

Das Programm reicht alle Eingaben zum Controller weiter. Alle Zeichen werden vom Controller als Echo zurückgeschickt und nur diese werden dargestellt. So lässt sich die erfolgreiche Verbindung leicht testen.

Durch Drücken der Leertaste wird der Analysemodus aktiviert. Dieser zeigt in einem grafischen Fenster die Abweichung der Soll- zur Ist-Position an. Die Beobachtung dieses Verlaufes ermöglicht eine genaue Einstellung der Servo-Parameter.

Das Programm wird mit ESC beendet.

Controller

Achtung: Trennen Sie für die erste Inbetriebnahme jeder Steuerung unbedingt die Motorspannung von der Platine. Sie vermeiden damit gefährliche Bewegungen ihrer Maschine, die durch HW-Fehler oder falsche Konfiguration entstehen könnten.

Beginnen Sie falls möglich mit einer geringen Motorspannung, idealer Weise mit Strombegrenzung aus einem Labornetzteil und steigern Sie die Spannung dann vorsichtig bis auf den Nennwert.

Installieren Sie unbedingt einen Notaus-Schalter, mit dem man die Motorspannung im Gefahrenfall unterbrechen kann.

Unmittelbar nach Anlegen der Betriebsspannung führt der Prozessor einen Reset durch, bei dem die Betriebsparameter aus dem EEPROM geladen werden und der Regelungsprozess initialisiert wird.

Am Ende der Initialisierung meldet sich der Prozessor über die serielle Schnittstelle mit:

```
UHU Servo Controller 3.00 (c) Snr: 134
EEPROM: OK
Hilfe mit (?)
>
```

Im Lieferzustand sind die Parameter auf in der Regel sichere Werte voreingestellt. Trotzdem wurde bewusst eine falsche Prüfsumme gesetzt, damit die Steuerung nicht ohne vorherige Kontrolle in Betrieb gehen kann. Dies führt beim Start zur Ausgabe

```
UHU Servo Controller 3.00 (c) Snr: 134
EEPROM: Fehler!
Hilfe mit (?)
>
```

Nach überprüfen der Einstellungen kann der Controller aktiviert werden indem man einen Soft -Reset durchgeführt.

Alternativ kann der Parametersatz mit 'S0' abgespeichert werden. Der Controller startet dann nach dem nächsten Hard-Reset oder nach dem nächsten Einschalten mit der überprüften Konfiguration.

Einstellstrategie

Glücklicherweise ist das Einstellen eines PID-Reglers bis zu einer stabilen Regelung nicht wirklich kompliziert, vorausgesetzt, daß die Regelgrenzen für das System ausreichend sind. Bei dem hier angewandten Algorithmus ist es fast noch einfacher, da die Gegenkoppelung besser greift. Es kann und wird aber auch Konfigurationen geben, bei denen der Regler versagt, z.B. wenn zwischen Motor und Encoder zu viel Spiel auftritt.

Das Feintuning des Systems ist dann doch wieder eine Sache, bei der man einiges Gefühl mitbringen sollte.

Am besten findet man die Grundeinstellung ohne ext. Takt heraus, indem man das System händisch auslenkt und die Reaktionen beobachtet (Motorwelle verdrehen und auslassen). Genauso geht es natürlich mit schnellen Verfahrensschritten vom CNC-PC oder auch mit dem 'E' Kommando aus dem Terminal.

Schritt 1

Anschluss des Steuer-PCs mit dem CNC-Programm. Starten des Analysemodus im Terminalprogramm.

Schritt 2

Setzen der Parameter auf 'relativ ungefährliche' Startwerte.

P = 1000

I = 25

D = 255

B = 1000

G = 3000

M=255

T=100

Schritt 3

Schrittweises Erhöhen des P-Werts bis das System nach Bewegungen zu schwingen beginnt und Dämpfung der Schwingungen durch Erhöhung des B -Parameters (ggf. auch D-Parameter. Die Erhöhung von B ist wegen der geringeren Auswirkung auf den Schleppfehler vorzuziehen. D eignet sich zum Feintuning und zur Stabilisierung besonders nervöser Systeme.)

Es gibt ein Optimum, wo durch den P-Parameter schnelle Reaktionen des Systems erzeugt werden und sich die dämpfende Wirkung des B-Parameters noch nicht zu stark auf die Dynamik auswirkt. Dies ist am besten experimentell zu bestimmen, indem man das System schnelle Bewegungen ausführen lässt und die entstehende Kurve am Analyzer beobachtet. Sie soll möglichst glatt sein, keine Schwingungen aufweisen und geringe Ausschläge beim Beschleunigen zeigen. Hier sollte auch beachtet werden, dass im CNC - Programm eine dem Motor angemessene Rampe definiert werden muss, die auf die maximal mögliche Beschleunigung des eingebauten Motors (incl. Mechanik) abgestimmt ist. Man kann einen Regelkreis nur im endgültigen Aufbau abstimmen!

Hat man einen stabilen Arbeitspunkt gefunden, P zur Sicherheit 10% niedriger setzen.

Sollte die Motorkennlinie gar nicht passen, kann man M heruntersetzen, allerdings verliert der Motor damit Drehmoment und Maximalgeschwindigkeit.

Schritt 4

Schrittweises Erhöhen des I-Werts bis das System nach Bewegungen zu schwingen beginnt. Dann I 10% niedriger setzen. **Vorsicht: Es können sehr vehemente Schwingungen auftreten! Unbedingt einen Not-Aus-Schalter in Reichweite halten!**

Schritt 5

Weiteres Optimieren nach Schritt 3 und 4 und mit dem T-Parameter falls nötig.

Schritt 6

Speichern der Konfiguration mit S0 und ggf. S1 oder S2.

Änderungshistorie

;Rev 3.00

; Freigabeversion

;Rev 2.17

- ; Konditionale Assemblierung Watchdog, Initialisierung auf 125ms
- ; Konditionale Assemblierung Spacing, bei deaktiviertem 'N' Intervall auf 672ms fixiert
- ; Timer Prescaler von 64 auf 256 umgestellt

;Rev 2.16

- ; Umstellung der Initialisierungssequenz
- ; Preset des N-Timers als Sicherheit für Soft-Reset
- ; Korrektur Ausgabestring bei Soft-Reset
- ; WD-int sichert Register
- ; 'Optische' Verbesserungen am Code

;Rev 2.15

- ;WD per Interrupt statt Reset aktiviert

;Rev 2.14

- ; Fehler bei multiplem EEPROM-Read behoben.

;Rev 2.13

- ; Diverse Aufräumarbeiten (LPM + MOVW)
- ; Zähler und Analyzer entkoppelt
- ; Umbau I-Wert auf DIV 4

;Rev 2.12

- ; Problem bei der Überlappung Z und Analyzer beseitigt.

;Rev 2.11

- ; Fehler bei Nutzung R0 in RS232 Input beseitigt
- ; Anzeige Seriennummer

;Rev 2.10

- ; Analyzer in Z wieder eingebaut
- ; Fehler in Reihenfolge Berechnung Grundparameter beseitigt
- ; Encodereingänge gleichlaufend zur 2.02t (Pin8 -->13, Pin9-->12)

;Rev 2.09

- ; Fehler in Z-Funktion beseitigt.

;Rev. 20.8

- ; Umbau I-Wert auf DIV 256

;Rev. 2.07

- ; Lin-Encoder ausgebaut (Platz...)
- ; Counter-Pos auf Register geändert
- ; unbenötigte Registerzuordnungen gelöscht
- ; Fehlerbehandlung statt Interpolation in den Encoder-Zähler eingebaut
- ; Einführung des (W)rap Zählers für verschlafene oder falsche Encoder-Flanken
- ; Alle Interrupts für die Dauer der Berechnung der Grundwerte der Regelberechnung deaktiviert
- ; Reihenfolge der Variablen verändert (und damit der EEPROM-Parameter) für fail safe
- ; Fehler Startwert bei der Initialisierung 20/24MHZ behoben
- ; Fehler ZL aus Zähler-Int behoben
- ; Busy Flag in Berechnung eingeführt
- ; Neuberechnungswert mit NEG umgerechnet

;Rev. 2.06

- ; Experimentell: Änderung Interrupt -Steuerung / verworfen
- Stand 25.9.05

;Rev. 2.04, 2.05
; Experimentell: Temporäre Änderung Zähler -Ausgabe / verworfen

;Rev 2.03
; Umbau auf Encoder Interruptbetrieb auf Port B0/1 (Pin 12 und 13)
; Achtung: Parameter für Neuberechnung bezieht sich jetzt nicht mehr auf 1/ 192 Clk sondern auf 1/64 Clk

;Rev. 2.02t24
; Umbau auf 24 MHz Quarz
; Phase Correct PWM 25 kHz

;Rev. 2.02t
; Watchdog 16ms eingebaut

;Rev. 2.01
; Slave shutoff auf PINB2 / Pin 14 active low eingebaut.
; Prüfsumme und Forward Error Correction für EEPROM eingebaut

;Rev 2.0 **; Freigabeversion**

;Rev 1.9.0dual4
; Schleifenabfolge / Timeslot für Neuberechnung programmierbar
; Prüfsumme für EEPROM eingebaut
; Nothalt Ein- und Ausgang in Software vorgefertigt.
;

;Rev 1.9.0dual3
; Timeslot auf 2ms verändert
; Z setzt nun immer erst den Zähler durch Reset zurück
; Lin. Encoder-Abtastung aktiviert (auf gleichem Port wie rot. / nur Perf. Test)
; Version läuft mit V=1 sehr rund.

;Rev 1.9.0dual2
; Fehler beim Limiter-Aufruf beseitigt
; Eingangsports mit Pullup
; Limiter auf Analysemodus
; Ripplefilter ausgebaut. Bringt nichts.

;Rev 1.9.0dual1
; Versionsnummer nachgezogen auf 1.9
; Ripplefilter (24 Takte) in Step Input eingebaut
; Register Zuordnungen verändert: _analyzeDiv, _mul, _0

;Rev 1.8.9dual4
; Aufrufe für Help und Selektion der Parametereingabe verkürzt
; Trigger für Analysemodus umgebaut
; Analysemodus auf 16 Bit
; Flag gegen Reentry der Berechnung eingebaut

;Rev 1.8.9dual3
; Massive Umbauten.
;- Mul1616 auf indirekte Adressierung umgebaut
;- AddMul1616 eingebaut
;- 16 bit Limiter eingebaut

;Rev 1.8.9dual
; Eingang für Linearencoder eingebaut (unfertig, Differenzierung der Encoder fehlt...)
;Dazu:
; Variablen in R- und L- gedoppelt
; Alle Counter auf Memory umgebaut bis auf Pos und Comm
; Geschwindigkeitsbegrenzer ausgebaut

;Rev 1.8.9
; Schleifendurchläufe gecheckt
; Kommentare vereinheitlicht
; xmit2 ausgebaut

;Rev 1.8.8.y
; Einbau B-Kompensation
; PC-Software neu skaliert: 16 Zeilen Text, Display schmaler.

;Rev 1.8.8.x
; Reduzierung des Rom Bedarfs durch bessere Stringverwaltung

;Rev 1.8.8
; seltsame Probleme beim Reset gelöst: nicht initialisierter V_Par
; Interrupt-Tabelle vervollständigt

;Rev 1.8.7
; PC-Software auf reinen Grafik-Betrieb umgestellt, ID und (C) eingefügt
; Störender Balken beim Scrollen beseitigt
; Umschaltung Analysemodus mit <Leertaste>

; Rev 1.8.6
; Hilfe für Analyse und Rückmeldung entfernt
; Analyse a=on x=off
; Eilgang-Steps mit Cursor (32) und Ctrl -Cursor(64) in PC-Software eingebaut

;Rev 1.8.5
; Test neuer Schleppfehler mit Basis alte Soll Pos
; SEI versetzt hinter Schleppfehlerausgabe
; Schleppfehler-Led ausgebaut, zeigt jetzt nur mehr Fehlerabschaltung an
; Umbenennung alle Variablen auf xxxxxL/M/H

;Rev 1.8.4
; Reset schaltet Analysemodus mit ab.
; falscher Text bei 'Vervielfacher' behoben

;Rev 1.8.3
; Texte Komprimiert
; (M)oment und (T)rägheit eingeführt
; (Multiplikator in (V)ervielfacher umbenannt
; senden der Fehlerwerte als Positiver Wert (addi64)