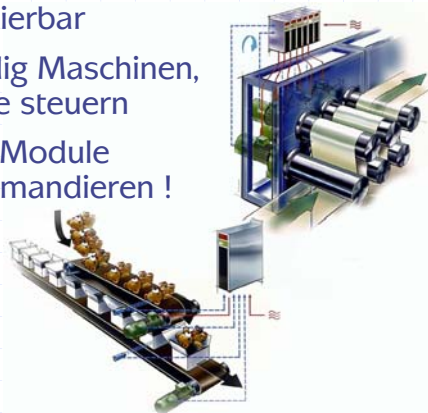


- ... Drehzahlregelung
- ... Positionierung
- ... Synchronisation

Motion-Control inklusive

Geräte mit der zub Motion-Control-Firmware ...

- ... sind frei programmierbar
- ... können selbstständig Maschinen, Anlagen und Geräte steuern
- ... können Peripherie-Module verwalten und kommandieren !
- ... sind 10'000-fach weltweit im Einsatz
- ... erfordern KEINE Runtime-Lizenzen !



zub Produktgruppe: MACS₃

MACS₃ = 1-,2-,3-Positionierung & Synchronisation







- Standalone oder mit SPS/PC vernetzbar
- Motion-Control inklusive & frei programmierbar
- 2 x CAN, 1 x RS232
- Digitale & analoge I/Os
- Alle Servos & FUs anschliessbar

zub Produktgruppe: MACS₄

MACS₄ = Motion-Control für Beckhoff-SPS



 + 

- Standalone oder mit SPS/PC vernetzbar
-> Optimiert für Beckhoff-SPS
- Motion-Control inklusive & frei programmierbar
- Ethercat In/Out, 1 x CAN, 1 x USB, 1 x RS485
- 6 Encoder-Eingänge
- Alle Servos & FUs anschliessbar

zub Produktgruppe: DSA

DSA = Digitale Servoverstärker



CANopen

- CANopen DS402 kompatibel
- Als Analogservo-Ersatz konfigurierbar
- Digitale Strom-, Drehzahl- & Positionsregelung
- PWM-Endstufe für bürstenlose & bürstenbehaftete Motoren
- Kompakte, schmale Bauform für die Hutschiene

zub Produktgruppe: OEM

OEM = Kundenspezifische Lösungen



CANopen

EtherCAT
Technology Group

- Standalone-Module für die autarke Gerätesteuerung
- Integrierte Lösungen aus Motion-Control-Unit und mehreren Servo-Endstufen
- Servo-Verstärker in 24V, 230V und 380V Technik
- Schnittstellen- & Protokoll-Konverter
- "Intelligente" I/O-Module

zub Produktgruppe: FU

FU = Danfoss-Frequenzumrichter



- Danfoss Frequenzumrichter optional mit Motion Control Unit
- Danfoss AutomationDrive FC300
- Danfoss MicroDrive FC051
- Mehrachs-Motion-Control mit MACS-Modulen und Frequenzumrichtern

Vernetzte SPS-Konzepte

Beckhoff-SPS mit MACS Motion Control Modul



- Low-Cost,
 - Integrierte Encoder-Schnittstellen
 - Ethercat on board
- Lizenzfreie Motion-Control
 - Positionierung
 - Synchronisation
Kurvenscheibe, Fliegende Säge
- Servoverstärker und Frequenzumrichter aller Hersteller anschliessbar

Vernetzte Servo-Antriebe

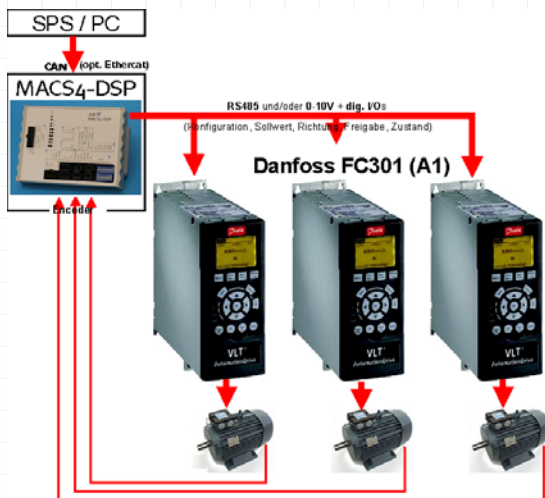


Steuerung /
Regelung:
MACS

Digitale
Endstufe:
DSA

Aktor &
Sensor

Vernetzte Frequenzumrichter



Steuerung /
Regelung:
MACS

Frequenz-
umrichter

Aktor &
Sensor

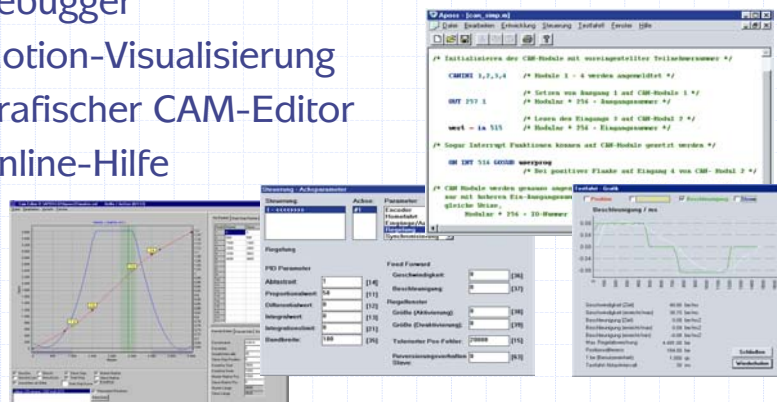
APOSS: Sprachmerkmale

- Strukturierte Programmiersprache mit Schleifen, Funktionen, Interrupts, ...
- Kapselung komplexer Antriebs- und Kommunikationsprobleme
 - Befehle analog bekannten Steuerungs Begriffen
 - Befehle zur Achs-Positionierung / -Synchronisation
 - Befehle zur CAN-Kommunikation
 - Wizards zur Erstellung von Standard-Applikationen

Tausendfach im industriellen Einsatz bewährt!

APOSS: Entwicklungsumgebung

- Kontextsensitiver Editor
- Debugger
- Motion-Visualisierung
- Grafischer CAM-Editor
- Online-Hilfe



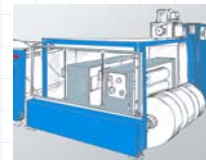
APOSS: Anwendungen

- Lagern
 - Warenkorb-Positionierung in automatisierten Lagern
- Positionieren
 - X/Y-Positionierung von Proben in Analysegeräten
- Dosieren
 - Hochkonstante und dosierbare Durchfluss- und Mengenregelung



APOSS: Anwendungen

- Etikettieren
 - Synchronisation der Etikettenzuführung auf den Warenfluss
- Wickeln
 - Synchronisation der Verlegung mit der Geschwindigkeit der Wickelguts
- Beschicken
 - Synchrone Zuführung von Komponenten, z.B. Zuführung von Zeitungsbeilagen



APOSS: Drehzahl regeln

```

Demo_Drehzahlregelung.m
// Beispielprogramm fuer Drehzahlregelung
// Eing.1: Drehzahl + 10% / Eing.2: Drehzahl - 10% / Eing.3: = STOP
gosub InitDrive
// Programm-Variablen & Drehzahlregelung
G_OldVel = 0 // Variable: Alte Soll-Geschwindigkeit [%]
G_CndVel = 0 // Variable: Neue Soll-Geschwindigkeit [%]
cvel x(1) G_CndVel // Geschwindigkeit setzen
cstart x(1) // Drehzahlmodus starten

// HAUPTSCHLEIFE *****
endless:
if (in 3) then // Eing. 3 => STOP
    G_CndVel = 0 // Soll-Geschwindigkeit = 0
elseif ((in 1) and (G_CndVel < 100)) then // Eing.1 => Drehzahl + 10%
    G_CndVel = G_CndVel + 10 // Soll = Soll + 10
elseif ((in 2) and (G_CndVel > -100)) then // Eing.2 => Drehzahl - 10%
    G_CndVel = G_CndVel - 10 // Soll = Soll - 10
endif

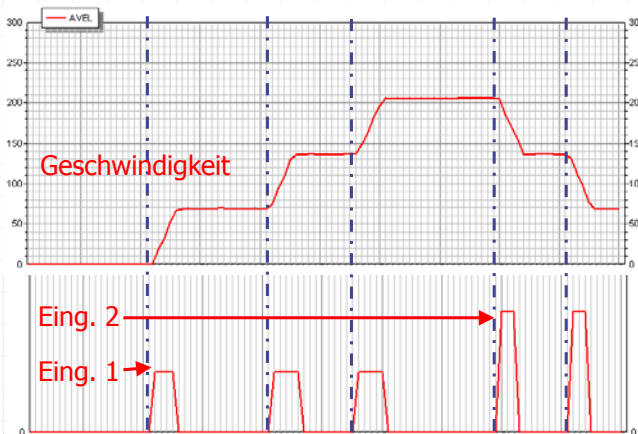
if (G_OldVel != G_CndVel) then // Soll-Geschwindigkeit uebernehmen
    cvel x(1) G_CndVel
    G_OldVel = G_CndVel
    print "Neue Soll-Geschwindigkeit: ",G_CndVel,"%
    waiti 1 off // Eing.1 entprellen
    waiti 2 off // Eing.2 entprellen
    delay 50 // 50ms warten
endif
goto endless
    
```

Programm-
beispiel:

Über digitale
Eingänge
gesteuerte
Drehzahl-
regelung

Eing.1: +10%
Eing.2: -10%
Eing.3: Stop

APOSS: Drehzahl regeln



Geschwindig-
keitsverlauf
 $v = f(t)$

Eingangs-
signale

Eing.1 = +10%
Eing.2 = - 10%

APOSS: Positionierung 1

```

Demo_PositionCycle.m
// Beispielprogramm fuer zyklisches Positionieren
// 5 Umdrehungen vorwaerts + 1 Sekunde warten
// 5 Umdrehungen rueckwaerts + 1 Sekunde warten

gosub InitDrive

// Programm-Variablen
G_RelDist = 5 * 4 * (get_encoder x(1)) // Distanz fuer 5 Umdrehungen berechnen

// Nullpunkt setzen
deforigin x(1)
// Beschleunigungs- / Bremsrampe und Geschwindigkeit definieren
acc 100 // 100% Beschleunigung
dec 30 // 30% Bremsrampe
vel 30 // 30% Geschwindigkeit

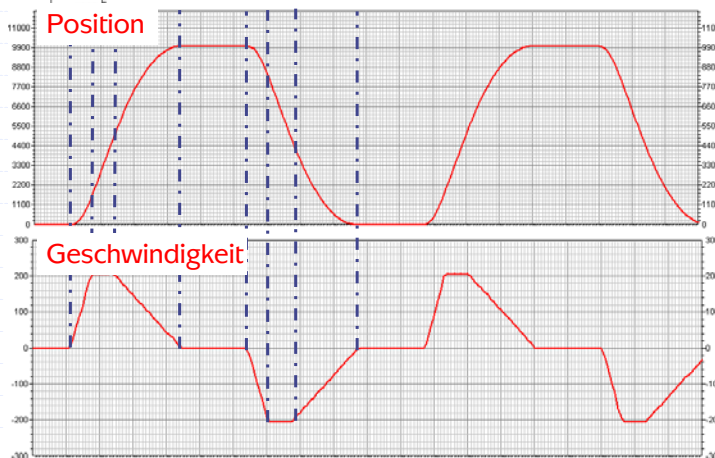
// HAUPTSCHLEIFE *****
endless:
  posr G_RelDist // 5 Umdrehungen vorwaerts
  print "Position: ",cpos x(1)
  delay 1000 // 1 Sekunde [ms] warten
  posr -G_RelDist // 5 Umdrehungen rueckwaerts
  print "Position: ",cpos x(1)
  delay 1000 // 1 Sekunde [ms] warten
goto endless
  
```

Programm-
beispiel:

Zyklische
Vor-/Rückwärts
Positionierung

Bremsrampe
= 30% Beschl.

APOSS: Positionierung 1



Position
 $s = f(t)$

0 -> 5 Umdr.
1 Sek. warten
5 Umdr. -> 0
1 Sek. warten

Geschwin-
digkeit
 $v = f(t)$

Bremsrampe
= 30% Beschl.

APOSS: Positionierung 2

```

Demo_PositionOnInput.m
// Beispielprogramm fuer Positionierung auf Eingangssignal
// Eing.1: 5 Umdr. vorwaerts / Eing.2: 5 Umdr. rueckwaerts

gosub InitDrive

// Programm-Variablen
G_RelDist = 5 * 4 * (get_encoder x(1)) // Distanz fuer 5 Umdrehungen berechnen

// Nullpunkt setzen
deforigin x(1)

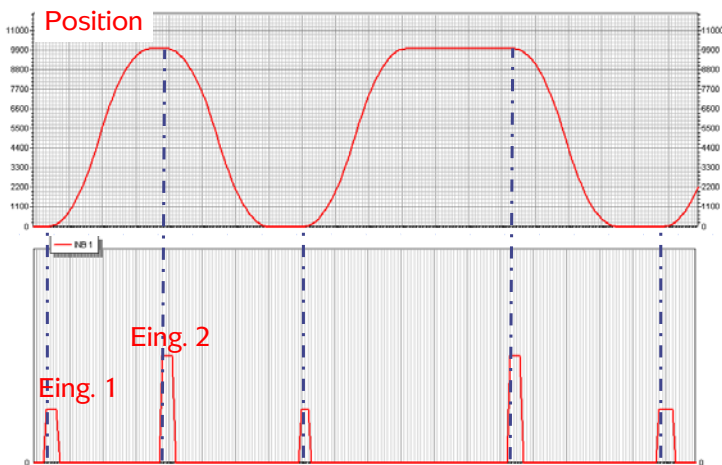
// HAUPTSCHLEIFE *****
endless:
  if ((in 1) or (in 2)) then // Eingang 1 oder 2 aktiv ?
    print "Positionierung von ",cpos x(1)," -> ";
    if (in 1) then // Eing.1 => 5 Umdr. vorwaerts
      posr G_RelDist
    elseif (in 2) then // Eing.2 => 5 Umdr. rueckwaerts
      posr -G_RelDist
    endif
    print cpos x(1)
    waiti 1 off // Eing.1 entprellen
    waiti 2 off // Eing.2 entprellen
    delay 50 // 50ms warten
  endif
  goto endless
  
```

Programm-
beispiel:

Über digitale
Eingänge
gesteuerte
Vor-/Rückwärts
Positionierung

Eing.1: +5 Umdr.
Eing.2: -5 Umdr.

APOSS: Positionierung 2



Position
 $s = f(t)$

Eingangs-
signale

1 = vorw.
2 = rückw.

APOSS: Synchronisation

```

Demo_PositionsSynchronisation.m
// Beispielprogramm fuer Positionssynchronisation

gosub InitDrive

// Nullpunkte definieren
deforigin // Leitachse
deforigin x(1) // Geregelter Antrieb

// Synchronisationsfaktoren definieren
set syncfacts 2 // Geregelter Antrieb doppelte Distanz wie Leitachse
set syncfactn 1

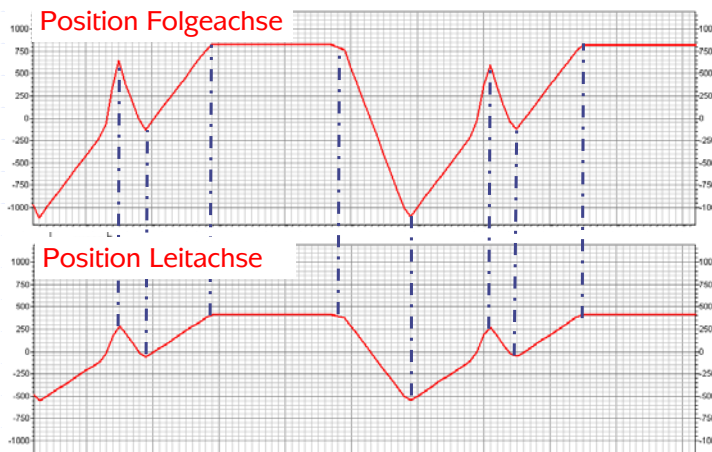
// Synchronisation aktivieren
syncp

// HAUPTSCHLEIFE *****
endless:
  print "Leitachse: ",mapos," / Folgeachse: ",cpas x(1)
  // Informationsausgabe alle 5 Sekunden
  delay 5000 // [ms]
goto endless
  
```

Programm-
beispiel:

Positions-
synchronisation
mit Faktor 2:1

APOSS: Synchronisation



Position
Folgeachse

=

2 x
Position
Leitachse

Resüme

APOSS Motion Control ist die ideale Steuerungs-
basis für die ...

- ... Positions-Regelung
- ... Geschwindigkeits-Regelung
- ... Achs-Synchronisation
- ... Prozesssteuerung

APOSS Motion Control lässt sich auf nahezu allen
8-, 16-, 32-Bit und DSP-Systemen implementieren !

zub-Knowhow nutzen

- Consulting
- Hardware-Entwicklung
 - Mikroprozessorlogik & Controller-Boards
 - Leistungselektronik
in 24V, 48V, 230V- und 380V-Technik
- APOSS-Integration & Software-Entwicklung
- Applikations-Engineering
- Produktion

zub - Unser Motto

Kein Antriebsproblem
ohne Lösung!

zub  machine control AG

Kastaniensteig 7
CH - 6047 Kastanienbaum / LU

Tel.: +41 - (0)41 - 348 00 30

e-mail: info@zub.ch

Fax: +41 - (0)41 - 348 00 39

Website: www.zub.ch