

Vergessen Sie alles, was Sie je über CAN-Programmierung gehört oder gelernt haben.

Wenn Sie schon einmal eine CAN-Vernetzung programmiert haben, wissen Sie, wie viele Programmzeilen und komplizierter Sende- und Empfangsprotokolle es bedarf, um auch nur ein weiteres Modul im Netz anzusprechen!

Und selbst mit der Makrosprache APOS und den schon vereinfachten Makrobefehlen DEFCANOUT / DEFCANIN zum Definieren von Sende- und Empfangsobjekten und CANOUT und CANIN zum Senden und Empfangen mussten Sie immer noch viel über Sendeprotokolle wissen, um ein CAN-Netz aufzubauen.

```
Aposs - CPOS.m *
Datei Bearbeiten Entwicklung Steuerung Iestfahrt! Fenster Hilfe

/* Initialisieren des CAN-Moduls mit der eingestellten
Teilnehmernummer */
CANINI 1          /* Modul 1 wird angemeldet */
wert 1 = IN 257   /* liest von Modul 1 Eingang 1 */
                /* (= 1*256 + 1) */
wert 2 = IN 261   /* liest von Modul 1 Eingang 5 */
                /* (= 1*256 + 5) */
wert 3 = INB 256  /* liest von Modul 1 Eingangsbyte 0 */
                /* (= 1*256) */
wert 4 = INB 257  /* liest von Modul 1 Eingangsbyte 1 */
                /* (= 1*256 +1) */

OUT 513 1        /* setzt von Modul 2 den Ausgang 1 */
                /* auf Wert 1 (= 2*256 +1) */
OUT 514 0        /* setzt von Modul 2 den Ausgang 2 */
                /* auf Wert 0 (= 2*256 +2) */
```

Das können Sie jetzt alles vergessen (bzw. müssen es nicht erst lernen!)

Mit nur zwei Befehlen können Sie jetzt CAN-Module im Netz ansprechen, Ein- und Ausgänge setzen und Relais schalten, ohne dass Sie auch nur ein einziges Telegramm oder Protokoll verschicken müssen:

Alle CAN-Module werden genauso angesprochen wie Ein- und Ausgänge, nur mit höheren Ein- und Ausgangsnummern. Diese entstehen aber immer auf die gleiche Weise:

Modulnr. * 256 + I/O Nummer

Sogar Interrupt-Funktionen können Sie auf die CAN-Module setzen, z.B. um bei positiver Flanke einen Eingang zu setzen.

1 Sie wollen zusätzliche I/O-Module benutzen?

Dann stellen Sie für jedes Modul mit dem Drehschalter eine eindeutige Modulnummer ein, bei 1 beginnend.

Wenn das Modul mit der Nummer 1 z.B. 16 Eingänge hat, sprechen Sie diese an, indem Sie die Modulnummer mit 256 multiplizieren und die Eingangsnummer addieren.

- ◆ Digitale Ausgänge zum Ansteuern von Koppelrelais und Schaltern
- ◆ Digitale Eingänge zum Lesen von Tastern, Schaltern, Sensoren und Näherungsschaltern
- ◆ Analoge Eingänge zum Auswerten von Potentiometern, Temperatur- und Drucksensoren usw.

2 Sie wollen mehrere Steuerungen untereinander verbinden?

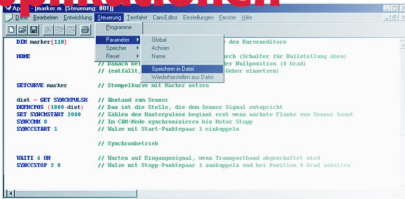
Dann verbinden Sie die Steuerungen, z.B. zwei MK1 mit dem CAN-Kabel und die Kommunikation wird – statt bisher über Ein- und Ausgänge – über den CAN-Bus durchgeführt.

Jede Steuerung ist dann in der Lage, Telegramme mit beliebigen Inhalten zu verschicken; die anderen können sie auswerten.

... ohne viele Ein- und Ausgänge oder unzähligen Kabeln?

Wenn Sie z.B. ein Werkstück verschieben und das Erreichen der Position melden wollen, mussten Sie bisher entweder viele Kabel ziehen oder viele Ein- und Ausgänge umständlich anschliessen, um die verschiedenen Meldungen und die Positionen abzusetzen.

Applikationen



Und so einfach geht das:

Mit folgenden wenigen Befehlen kann die MK1 das Werkstück verschieben und dann ein Telegramm verschicken, in dem neben der „Ankunftsmeldung“ auch die Position stehen kann. Die zweite MK1 kann darauf sofort reagieren und den nächsten Arbeitsgang auslösen, z.B. eine Bohrung.

3 Sie wollen viele Steuerungen miteinander verbinden und den Komfort des PC nutzen?

Verbinden Sie die Steuerungen mit einem CAN-Kabel mit dem PC, z.B. die 64 MK1 einer Bühnensteuerung. Nun können Sie auch die C-Bibliothek des PCs nutzen, um

- ◆ Nachrichten an alle Teilnehmer im Netz zu verschicken,
- ◆ den Status von allen Teilnehmern abzufragen oder
- ◆ zyklisch abzufragen, ob noch alle Teilnehmer im Netz sind.

So können Sie im PC die Beleuchtung für die Aufführung oder die Verwandlung beim Szenenwechsel speichern und dann beliebig abrufen.

Zusätzlich kann man mit APOSS die Fahrprogramme beim Probeaufbau mit der Teach-in-Funktion erarbeiten und einfach speichern.

```
co = DEFSCANOUT 1000 8
                                /* Definiere Telegramm 1000 Länge 8 */

start:
  POSA 10000                    /* Positioniere nach 10000 */
  lo = 1                        /* Kennung für 'angekommen' */
  hi = APOS x(1)                /* Aktuelle Position */
  CANOUT co hi lo              /* Telegramm mit hi/lo verschicken */
  WAITI 1 ON                    /* Auf neues Teil warten */
  POSA 0
  WAITI 1 OFF                   /* Teil angekommen */
  GOTO start
```

4 Sie wollen den CAN-Bus nur benutzen, um mit APOSS komfortabel zu programmieren?

Dann schliessen Sie die Steuerungen an einen CENTRONICS-Adapter an und Sie können mit allen Steuerungen gleichzeitig arbeiten:

Zum Beispiel können Sie mit jedem Teilnehmer einzeln eine Testfahrt durchführen oder in jede Steuerung ein anderes Programm laden.

Alle Befehle und Grundfunktionen von APOSS stehen für das komfortable Programmieren der Steuerung zur Verfügung.

Oder wollen Sie weiter auf unterster Ebene programmieren?

... etwa so ...

(* Rücksetzen von 3DO und DIAcan *)
(* Clear*)

```
LD 0
ST OutputsI_8
LD 0
ST DIAcan_Out
RET
```

(* Addition (8DI, bit2) *)

```
CheckAdd:
LD InputsI_8
AND 1
BYTE_TO_BOOL
JMPCN CheckSub
R Dir_Sub
```

(* Subtraktion (8DI, bit7) *)

```
CheckSub:
LD InputsI_8
AND 64
BYTE_TO_BOOL
JMPCN Run
S Dir_Sub
```

```
Run:
LD Dir_Sub
JMPC Subtract
```

(* Addition *)

```
CAL Add_One(
  Input :=OutputsI_8 )
```

```
CAL Add_One(
  Input :=DIAcan_Out )
```

```
RET
```

(* Subtraktion *)

```
Subtract:
```

Weblinks: ◆ www.aposs.ch ◆ www.zub.ch ◆



zub machine control AG
Alpenblickstrasse 10 · CH-8853 Lachen
Telefon +41-55-462 11 17 · Fax +41-55-462 12 86
info@zub.ch

www.zub.ch

Alle Produkt- und Firmennamen
eingetragene Warenzeichen der
Hersteller.