

Beskrivning

Kortet 4002 är ett intelligent motorstyrningskort i serien DataBoard 4680. 4002 är ett I/O-kort som är designat för att arbeta i en industriell miljö och för att uppfylla industriella krav.

Kortet kan användas i ett brett fält av applikationer, allt från beräkningskontroll till alla typer av positionering och DC-styrning där en värddator, typ DataBoard 4680 eller ABC 80/800, handhar övervakningen av processen. Kortet kontrollerar positionering, acceleration, inbromsning, fastlåsning, ändlägen etc.

Några exempel på användningsområden:

- Styrning av transfervagnar, traverser och pallastningsrobotar.
- Rörliga montage- och testjigg.

Uppbyggnad

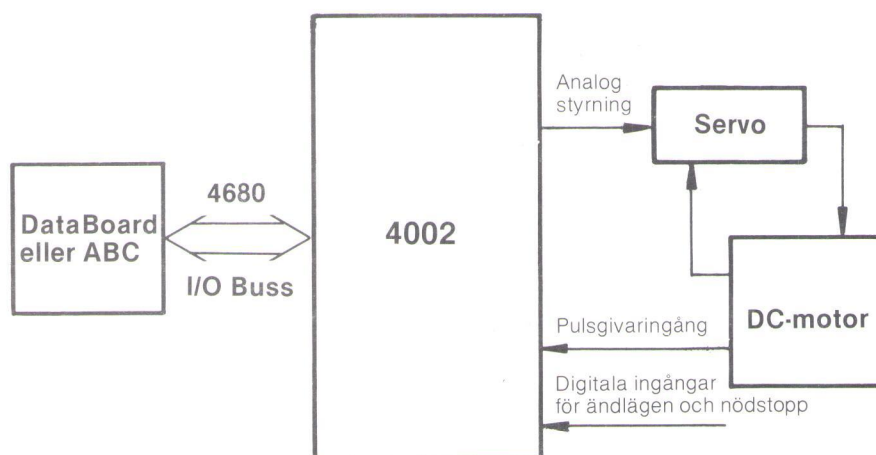
Kortet innehåller intelligens för styrning och övervakning och detta implementeras med hjälp av en enchipsprocessor, INTEL 8748.

Kortet medger styrning av en utrustning via en analog utgång, i reglerområdet ± 10 mVolt — ± 10 Volt. Positionsinformation till 4002 erhålls via pulsgivare.

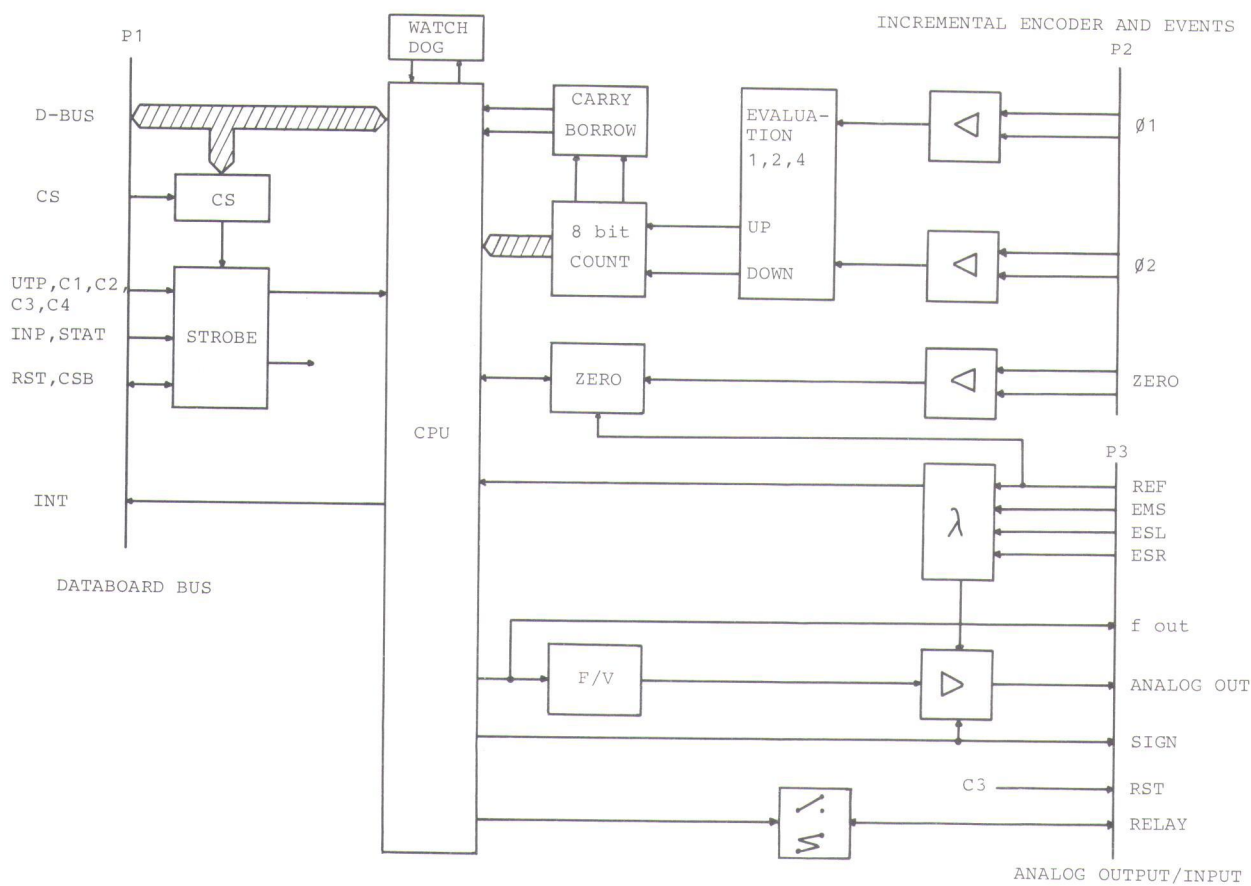
Riktningdetektor och räknare finns på kortet.

4002 kan antingen styra med maxfart eller med ramper för acceleration och retardation.

Direkt till kortet ansluts givare för ändstopp och nödstopp, dessa kontrolleras direkt av processorn på kortet.



Blockschema



Programfunktioner

Programvaran i 8748 handhar ett antal funktioner vilka nås via enkla kommandon på 4680-bussen. Exempel på dessa kommandon är:

Relativ positionering, absolut positionering, sätt hemmaläge, hastighet, acceleration, retardation, start, stopp, status, hysteres, låsning.

Reglering av hastighet sker i 1024 steg med hastighetskommando. När accelerations- och retardationsberäkningen görs på kortet regleras hastigheten i 24 + 24 intervall.

Tekniska data

Matningsspänning:	+ 5V	0,6A
	+ 12V	20mA
	- 12V	10mA
Arbetstemperatur:	+ 10 — + 40°C	
Storlek:	160 × 100 Europakort standard	
Anslutningsdon:	Till DataBoardbussen 1 st 64 pol Europadon	
	Till inkrementalgivare DB 15/p	
	Till analogutgång och ändstopp DB 25/p	
Ingångar:	Pulsgivare 5V Diff	
	Statusgivare 5V Optoisolerat	
Utgångar:	Motorstyrning, analog +/10 mV — +/10V	
	Motorstyrningsfrekvens 140Hz — 10kHz	
	(Option 25Hz — 100kHz) TTL	
	Reläväxling för analogutgången.	

4002-10

DataBoard

BESKRIVNING

MOTORSTYRNINGSKORT

87-4002-10 (A)

jan 85

**DATA
INDUSTRIER AB.**

Box 2029, 18302 TÄBY. Tel 08-768 06 60. Telex 10978.

DataBoard

85-01-20

Angående DataBoard Counter & positioning unit 4002.

Countposit kortet 4002 har i dagarna uppdaterats.

Uppdateringen består av ett kretsbyte som förbättrar analogutgångens prestanda. Kortets funktion påverkas ej.

Gällande revision är 4002-20.

Bifogade kortbeskrivning har beteckningen 4002-10.

Den gäller dock fortfarande på alla punkter.

Dataindustrier AB

Thomas Högenes

INNEHÅLLSFÖRTECKNING

Sammanfattning	2
Maskinvara	3
Kommandointerface	4
Räkneingångar	5
Val av arbetsmetod	5
Nollställning av räknare	7
Olika givares påverkan på räkneingångarna	7
Skydd mot magnetiska-, elektriska- och jordstörningar	12
Analogutgång	14
Regler mot störning	15
Referensingångar	17
Watchdog	20
Programvara	21
Aritmetik	21
Händelsehantering	22
Aktivering av händelse	22
Läsning av EVENT-ordet	24
Interrupt hantering	26
Beskrivning av portarnas användning	27
C2	27
C3	28
C4	28
STAT	28
Beskrivning av kommando	29
Beskrivning av slope tabellens innehåll	36
Uppstart av 4002	40
Programexempel	41
Verbal beskrivning	42
Program	43
Tekniska data	44
Schema	
Komponentplaceringsschema	
Kretsschema	
Kontaktplacering	
Strappningsfält	
APPENDIX	
Exempel på inkoppling av servosystem till 4002	
Inkoppling av diverse givare	

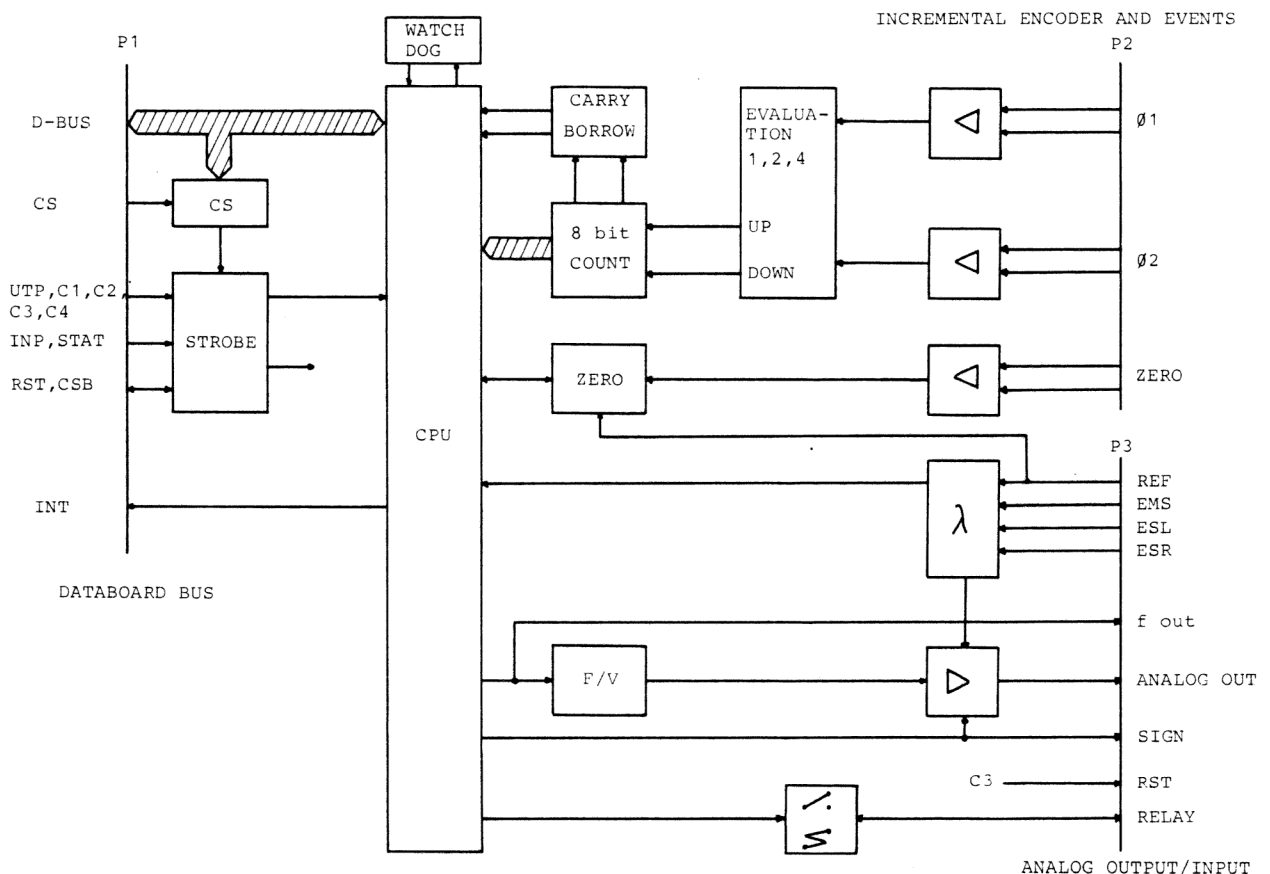
Sammanfattning

Kortet 4002 kan tillsammans med en värddator utföra motorstyrning och positionering av ett mekaniskt system. Kortet beräknar ärvärdet från den positioneringsinformation som den får via pulsgivare eller inkrementalgivare. Riktningdetektor och räknare finns på kortet. Kortet sköter styrningen via en analog utgång 10mV - 10V. Styrningen kan antingen ske med maxfart eller med ramper för acceleration och retardation. Direkt till kortet ansluts givare för ändstopp och nödstopp som kontrolleras direkt av processorn på kortet.

Något låst servosystem är det inte, utan den överordnande värddatorn har hela tiden kontroll över kortet 4002.

Kabeln till servot kan vid låg störmiljö vara 100m. Vid hög störmiljö bör ett yttre optoavskilt system användas.

Blockschema



Maskinvara

Maskinvaran på kortet 4002 kan delas in i fem delar, var och en med sin speciella funktion, se blockschema.

Ett kommandointerface som sköter den asynkrona kommunikationen mot DataBoard-bussen.

En räknardel som beräknar ärvärdet från information från yttre inkrementalpulsgivare.

En analogdel som skickar ut en börvärdessignal dels som en frekvens och dels som en proportionell spänning.

En referensdel som synkroniserar mot de externa villkoren.

Slutligen finns det en watchdog som övervakar att processorn inte har låst sig.

Kommandointerface

Kortet väljs in med kanalval. Utvalt kort ekar en CSB-signal till DataBoard-bussen.

Kommandona från värddatorn erhålles via DataBord-bussen. Denna buss består av en databuss (8 bitar) och 7 st strobe-signaler OUT, C1, C2, C3, C4, INP och STAT.

Reset kan antingen göras för hela I/O-sidan, via I/O-RST (INP 7), eller selektivt för enbart kortet, via signalen C3. Dessutom kan CPU-reset fås från kortets egen Watchdog.

4002-kortet kan sättas upp så att det genererar interrupt till värddator för vissa händelser på kortet.

Räkneingångarna

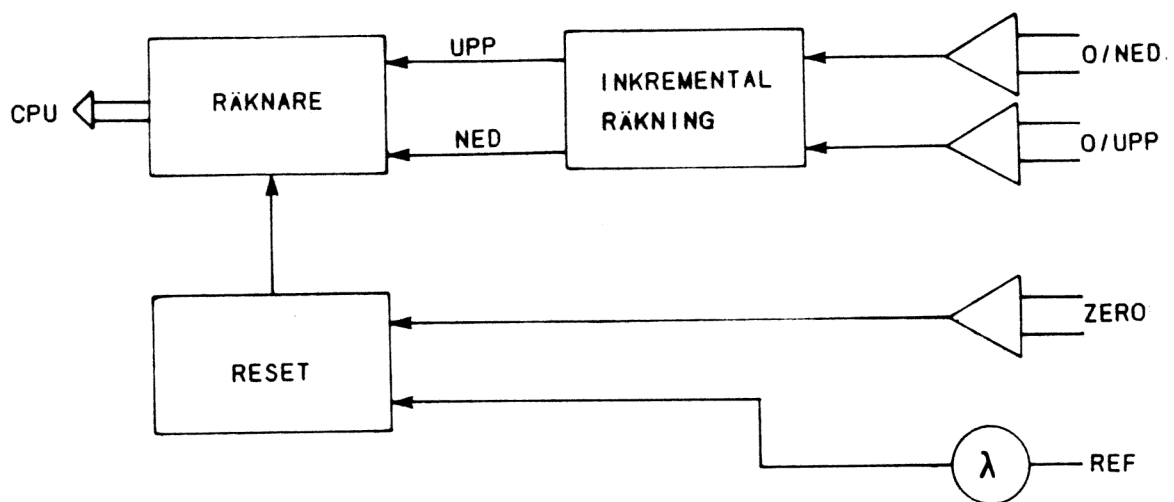
För att räkna yttre händelser finns två principiella arbetsmetoder.

Metod 1.

Pulsräkning med två ingångar, en för att räkna upp och en för att räkna ner.

Metod 2.

Inkremental räkning med hjälp av inkremental-pulsgivare med faldning (multiplikering) 1,2,4.

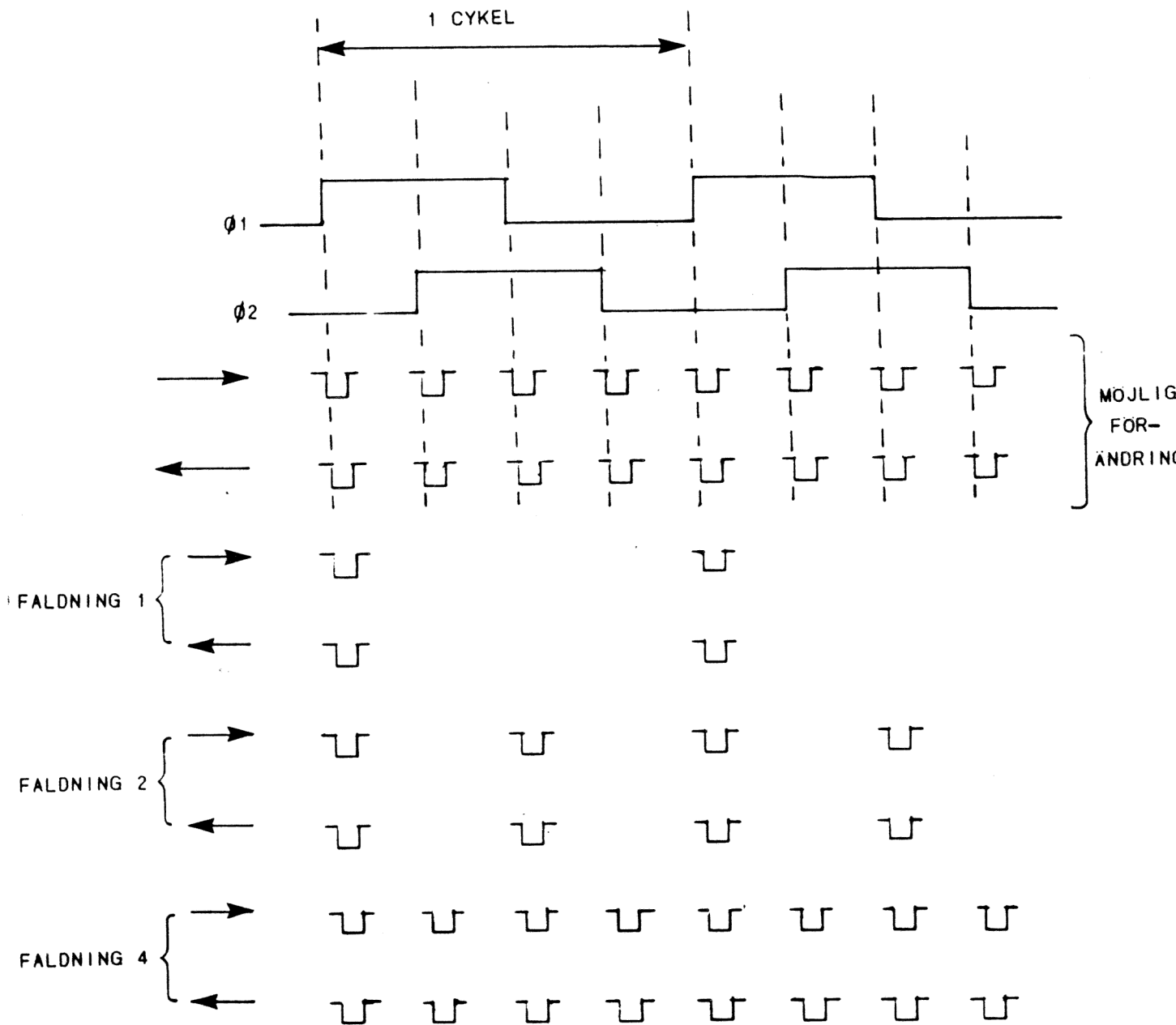


Val av arbetsmetod

Valet av arbetsmetod bestäms med ett strappfält/ bygelfält placerat mellan räknaren och ingångarna Ø1, Ø2.

Om arbetsmetod 1, pulsräkning, skall användas, sätts strapp J i för nedräkning (ingång Ø1) och strapp K i för uppräkning (ingång Ø2).

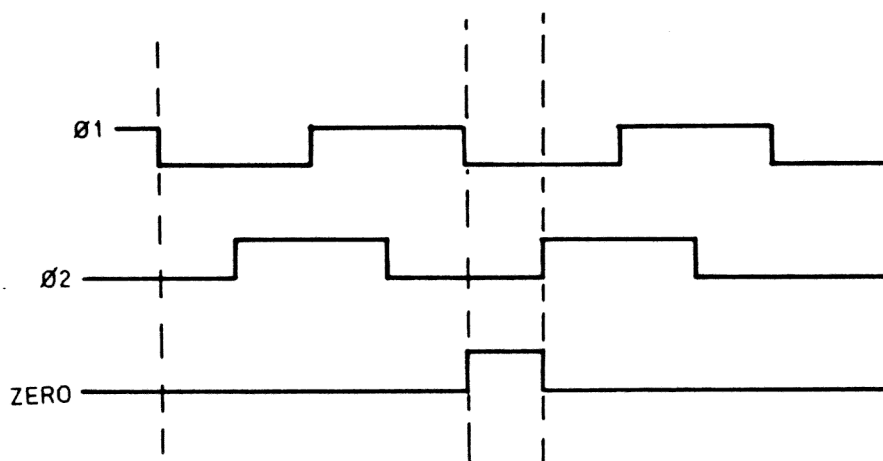
Vid val av arbetsmetod 2 skall strapparna A, B, ..., H användas. Den inkrementala givaren arbetar med två signaler som är 90 grader fasförskjutna till varandra, ur dess flanker kan en förflyttning och en riktning detekteras. När denna metod används måste en faldningsfaktor, multiplikationsfaktor väljas, i det här fallet kan den vara 1,2 eller 4. Med faldning menas att ur inkremental givarens två fasförskjutna signaler kan 4 + 4 händelser detekteras.



På kortet är strapparna placerade i position 4C. Sätt i strapparna enligt följande:

Strapp	Faldingspar							
	A	B	C	D	E	F	G	H
Faldning 1	x	x						
Faldning 2	x	x			x	x		
Faldning 4	x	x	x	x	x	x	x	x

Strappningsfältet för faldning är placerat mellan ingångsstegen och räknaren. Vid faldning med 1 eller 2 måste faldningsstrappar sättas i, så att ZERO-pulsen och räknepulsen stämmer överens. På inkrementalgivaren finns en punkt på varvet där en av flankerna hos O1 eller O2 stämmer överens med ZERO-pulsen. Eftersom det inte finns någon standard mellan givare kan det innebära att faldningsstrapparna A-B inte stämmer med ZERO-pulsen. Om så är fallet måste ett nytt läge på faldningsstrapparna prövas, C-D, E-D eller G-H.



Nollställning av räknare

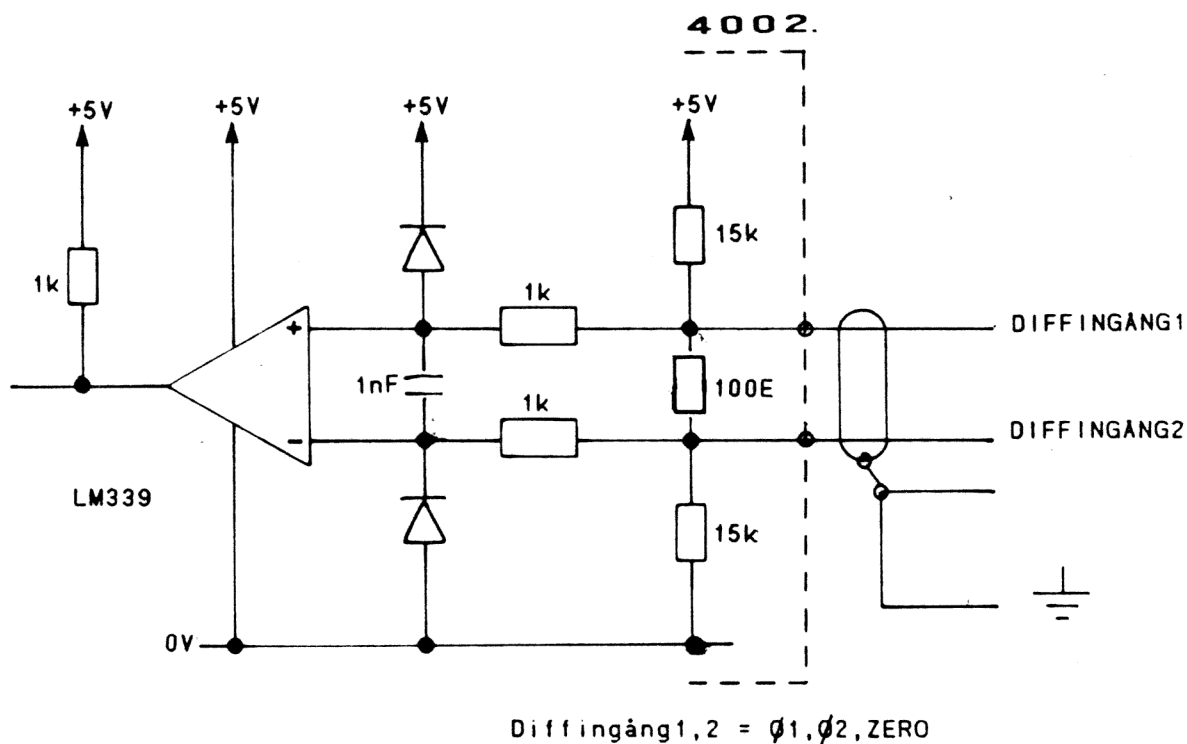
Räknaren kan nollställas med två olika ingångar, den differentiella ingången ZERO och den optoskilda ingången REF.

Nollställningen styrs av huvudprogrammet som i sin tur ger instruktioner till 4002-processorn. Nollställningen utförs sedan automatiskt antingen med bara ZERO eller REF eller med båda som ett OCH-villkor. När nollställningen är klar rapporterar 4002-processorn tillbaka till huvudprocessorn.

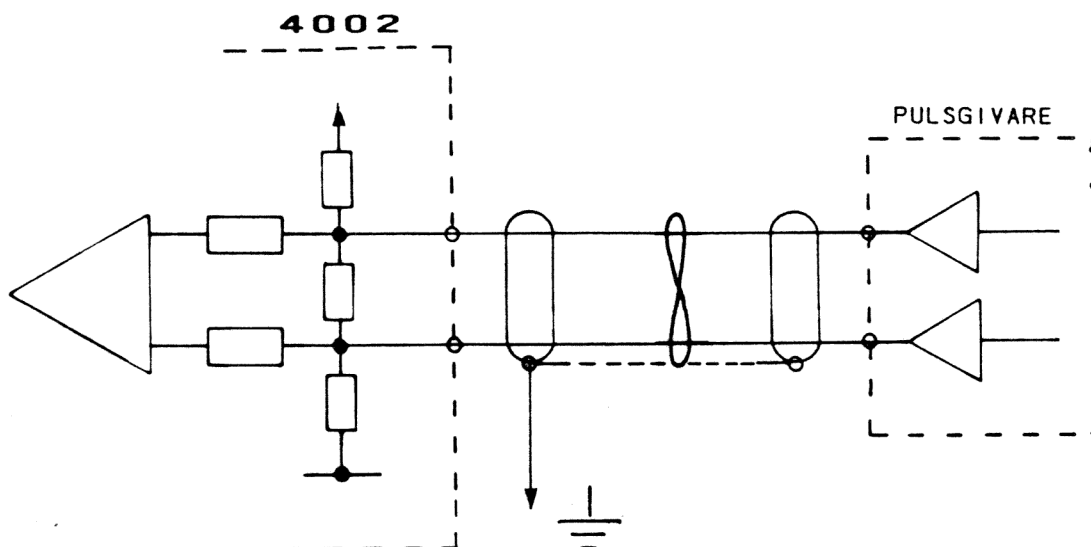
Olika givares påverkan på räknaringångarna

Ingångsförstärkarna till O1, O2 och ZERO är identiska och var och en består av en differentialförstärkare. Den karakteristiska ingångsimpedansen är 100 ohm. Värdet är valt för att matcha en partvinnad kabel. För att sänka gränsfrekvensen har en kondensator på 1 nF använts. Den praktiska övre gränsfrekvensen är satt till 100 kHz. Ingångsströmmen genom 100 ohms motståndet rekommenderas till minst 4 mA.

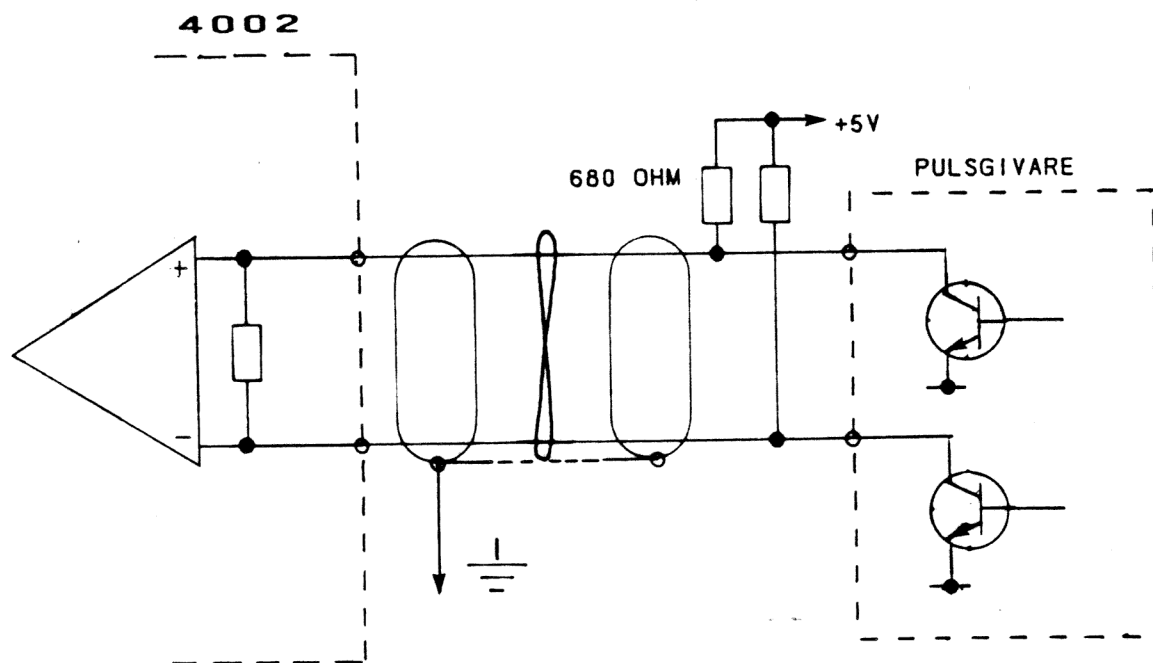
Optosnittet till ingången REF kräver 10 mA vid 5V. Om högre spänning önskas måste ett yttre motstånd kopplas in. Ingångarna är skyddade mot yttre störningar med skyddsdioder. Ingångarna är dimensionerade för 5V signalspänning.



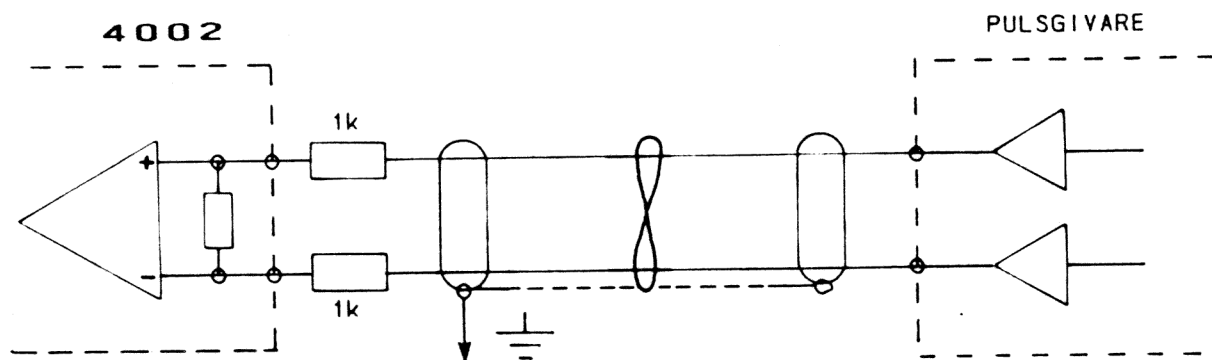
Vid drivning av ingångssteget från 5V logisk signal skall drivsteget aktivt kunna driva minst 4 mA.



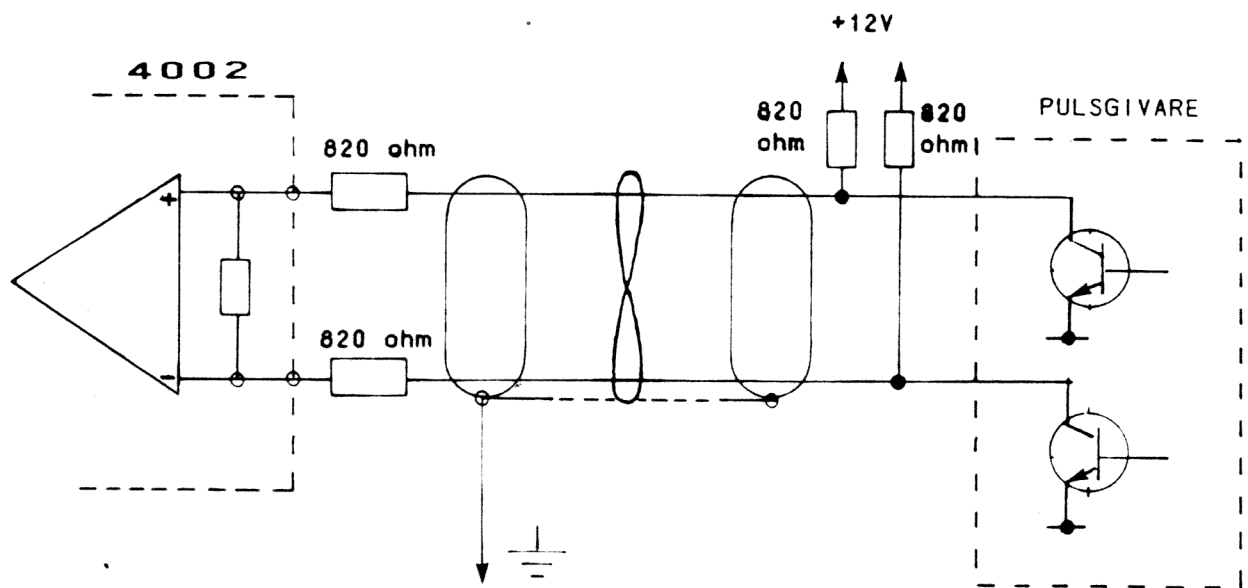
Om givaren har öppen kollektor och matnings-
spänningen är 5V skall yttre motstånd användas.



Inkoppling av en 12V's givare med aktiv
drivning.



12V's givare med öppen kollektorutgång.

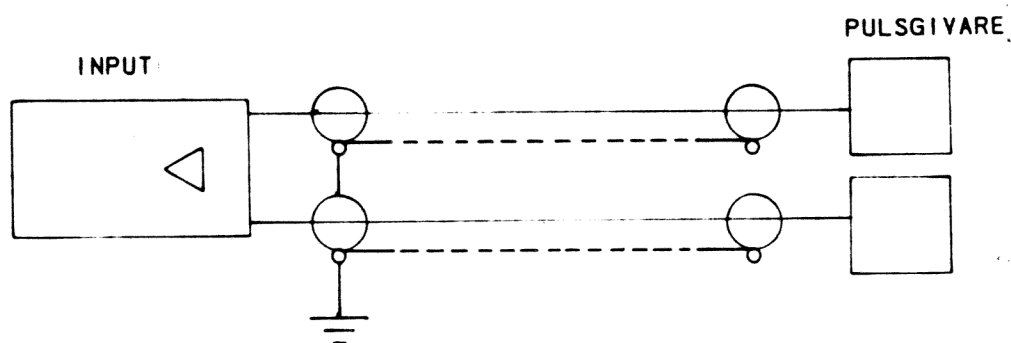


Ingångsimpedansen är korrekt för fig
Vid höga pulsfrekvenser samt långa kablar kan
ingångssteget modifieras så att den karakte-
ristiska ingångsimpedansen bättre stämmer överens
med kabelimpedansen. Med de ovan beskrivna
exemplen klars de flesta fall av inkopplingar.

Skydd mot magnetiska-, elektriska- och jordstörningar

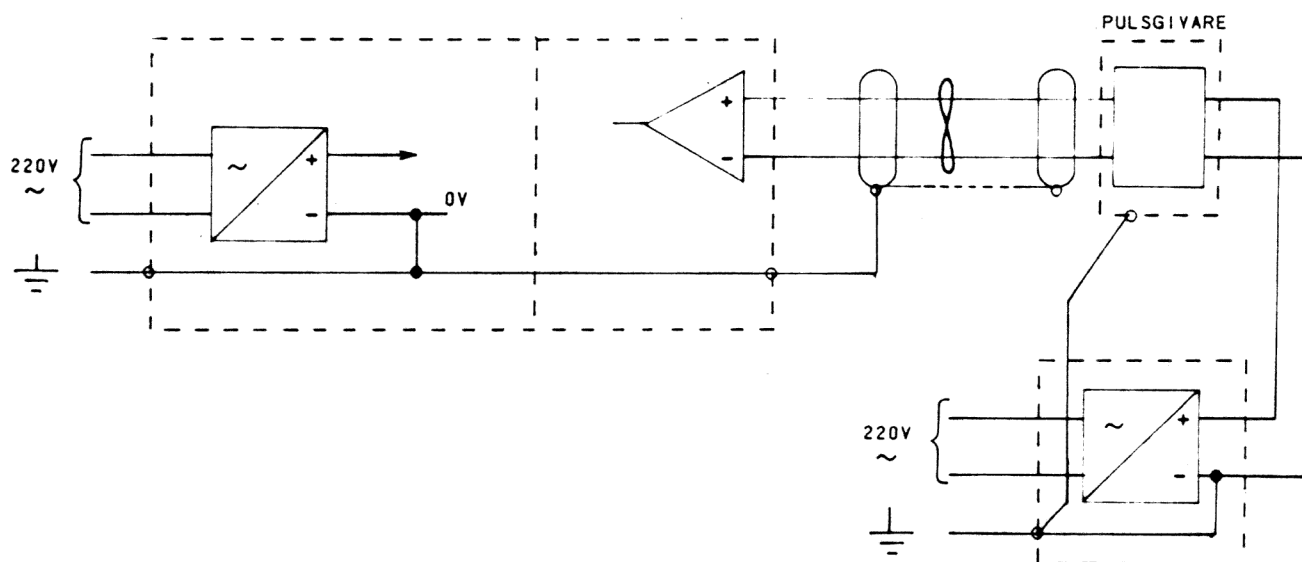
Vid ihopkoppling av olika elektriska apparater kan störningar av olika typer uppstå. Det finns tre olika metoder som kan användas för att minska inverkan av dess störningar, skydd mot magnetiska-, elektriska- och jordstörningar.

De magnetiska störningarna kan undvikas genom att en partvinnad kabel väljs och de elektriska störningarna genom att en skärmad kabel väljs. Men vid anslutning av skärmen kan jordströmmar uppkomma. För att undvika detta används principen att en skärm inte får anslutas i bägge ändar. Lättast uppfylls denna princip om alla skärmar ansluts i räknarkortet.

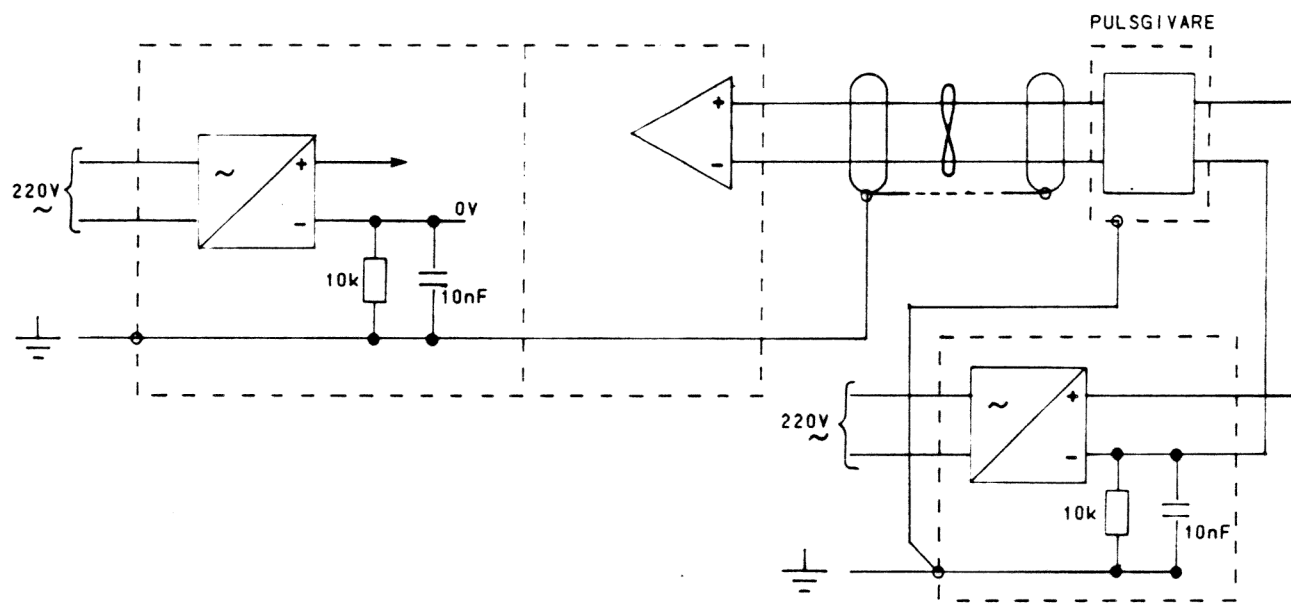


Vid jordning av apparater kan två olika metoder användas, direktjordning eller jordning via ett RC-nät.

Hos ett direktjordat system skall 0V hos varje kraftaggregat anslutas till chassijord.



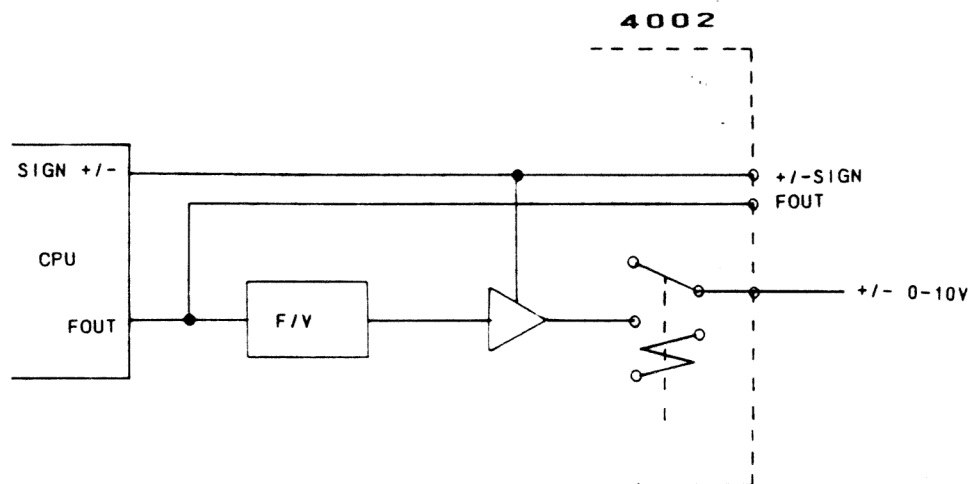
Ett bättre alternativ är att ansluta varje jordabonnent med 10 kohm//10nF mellan jord och 0V.



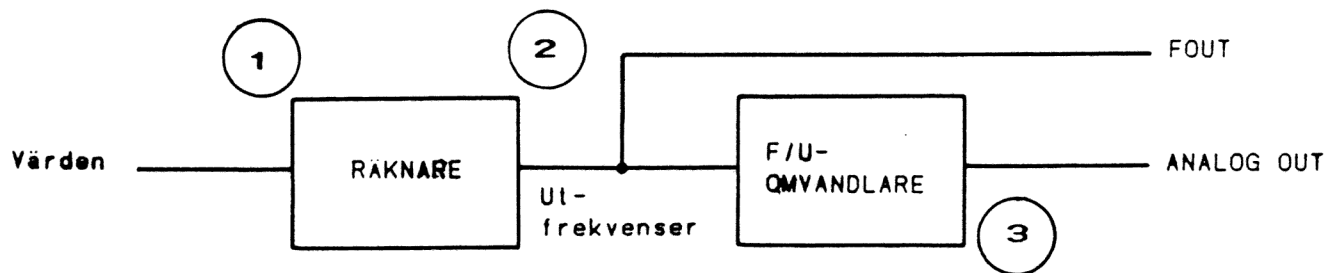
Vid en RC-kopplad ihoppkoppling av chassijord och 0V i system kommer jordpotentialerna att ligga på olika nivåer. Detta innebär att en jordström kommer att flyta genom motståndet R. Vid statisk elektricitet kommer laddningen att försvinna ur elektroniken via kondensatorn C. Det är på grund av denna statiska elektricitet som det är omöjligt att få chassijord och systemets 0V att ligga helt separerade. Men genom att koppla in motståndet R kommer jordpotentialen att hållas på en rimlig nivå.

Analogutgången

Kortet ger ut förflyttningsinformation både som en analogspänning (ANALOG OUT) och en digital frekvenssignal (FOUT) samt som en rörelseriktning (SIGN).



CPU'n har en 14 bitars timer/räknare som används som en programmerbar frekvensdelare.



Räknaren laddas med värden ① vilka ger ut-frekvenser ② från 25 Hz till 100kHz. Upp-lösningen i räknaren är omvänt proportionell mot frekvensen, eftersom det egentligen är periodtiden som är den programstyrda parametern. I denna applikation är detta snarare en fördel än en nackdel, därför att detta innebär att upplösningen ökar vid små värden på utspänningen, dvs låga hastigheter kan anges noggrannare än höga.

Den erhållna frekvensen ② går dels till en inbyggd frekvens/spänningsomvandlare, F/U ③, och dels till kortets utgångskontakt (FOUT). Det är därför möjligt att som alternativ använda en yttre frekvensomvandlare med hög precision. En fördel med detta är att det är lätt att galvanisk separera denna med hjälp av optokopplare.

Den inbyggda F/U-omvandlaren har ett arbetsområde mellan 140 Hz och 10 kHz. Det ger genom hela kedjan ett analogt utgångssving på 10 mV till 10 V. För att noggrannare kunna se förhållandet mellan delningstalet, här kallad rate, och utgångsspänningen se tabellen på sid 39.

I den efterföljande förstärkaren läggs riktningsinformationen, SIGN, till. Detta resulterar i att analogutgången antingen kan vara positiv eller negativ. Till förstärkaren är också ändlägesbrytarna (ESL och ESR) inkopplade, analogutgången ger 0V ut när ändlägesbrytarna har rätt polaritet eller när de indikerar stopp. Enda sättet att häva detta stopp är att en ny riktning indikeras. Utsignalen från förstärkaren går därefter vidare via ytterligare en förstärkare, ett buffertsteg, till ett 1 kohm motstånd, vilket skyddar förstärkaren från yttre störningar. Utgången är kortslutningssäker.

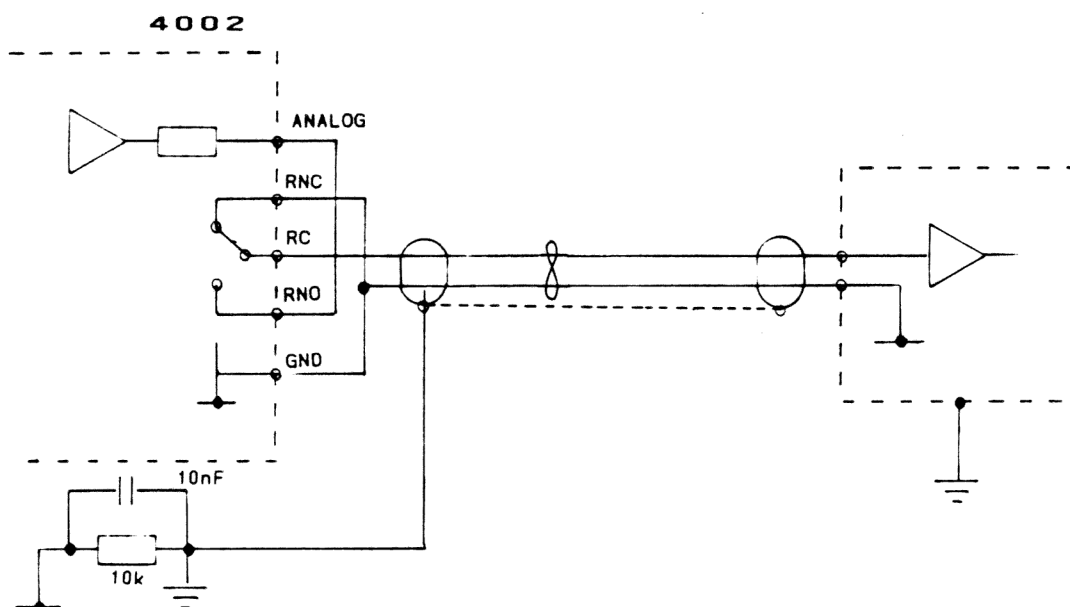
Externt kan den analoga utsignalen kopplas via det interna reläet på så sätt att servoingången erhåller en ren kortslutning när ett stopp indikeras från 4002-kortet. Servot bör i dessa fall stoppas och låsas. Det interna reläet påverkas direkt av signalen EMS, utan inblandning av programvara eller CPU.

Utgången RST aktiveras av kommandot C3 och används för att återställa yttre händelser som t ex felvippor.

Regler mot störning

För att undvika störningar mellan analogutgången och servoförstärkaren bör nedanstående regler följas.

* Använd partvinnad kabel med skärm.

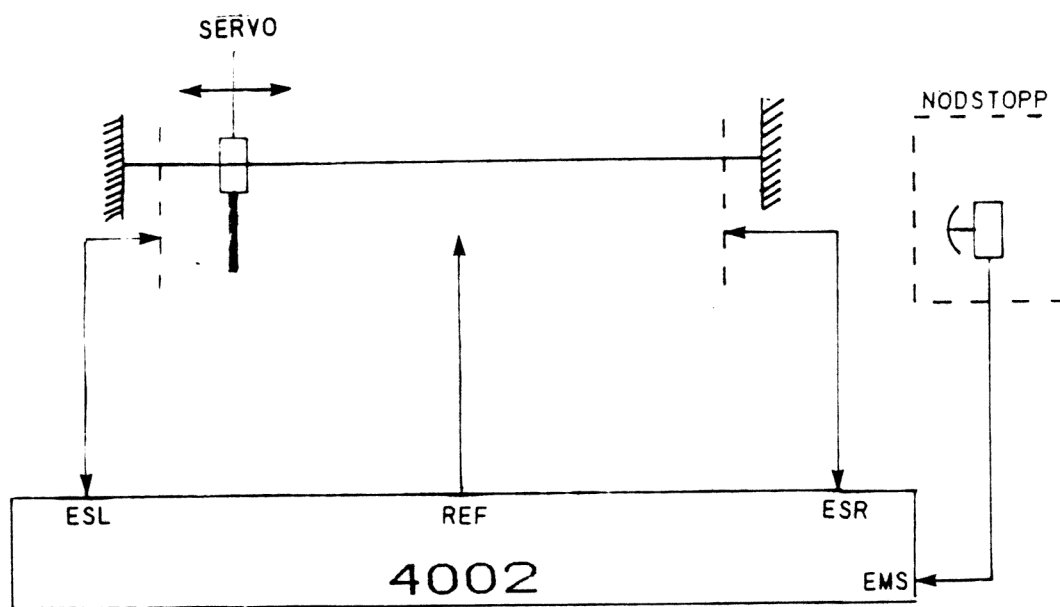


- * För att undvika jordströmmar skall skärmen bara anslutas i en enda punkt, om det finns flera jordar skall dessa anslutas i ett stjärnformat nät. En punkt skall väljas som huvudjordpunkt, lämpligen 4002.
- * För att ytterligare minska jordstörningarna kan de båda systemens jord förbindas med en skyddskabel med en minsta area på 6 mm.
- * Se till att kabeln läggs så att den inte går parallellt med andra störande kablar. Om detta inte kan undvikas måste denna kabel placeras så att den har en distans på minst 5 cm till den/de störande kablarna. Skulle kabeln i alla fall ta upp transienter utifrån kan ett RC-filter kopplas in på servoförstärkarens ingång.

Referensingångar

Det finns fyra statiska ingångar på motorstyrningskortet:

- REF - REFERenspunkt för nollställning
- EMS - EMergency-Stop, nödstopp
- ESL - End-Stop-Left, vänster ändläge
- ESR - End-Stop-Right, höger ändläge



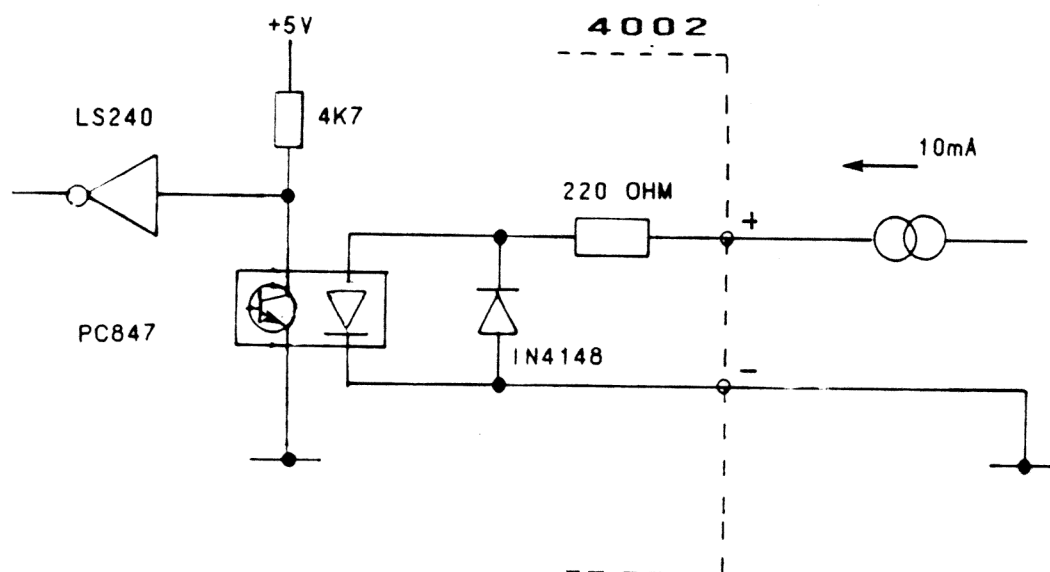
REF-ingången används tillsammans med ZERO-ingången för att nollställa räknaren. Den inkopplade inkrementalgivaren ger ut en ZERO-puls per varv, detta innebär att en hel rad ZERO-pulser kommer att genereras vid förflyttning längs med axeln. För att rätt ZERO-puls ska kunna väljas ut används REF-ingången som en grov inställning av var på axeln nollställningen skall ske.

Vid aktivering av EMS-ingången genereras omedelbart stopp i analogutgången, reläet vid utgången faller. Samtidigt avbryts 4002-programmet och huvudprogrammet får en indikering på att något har hänt. Om reläet inte används kommer inte EMS-signalen att påverka analogutgången.

Signalerna ESL och ESR är de båda gränslägena för axelns vänster- resp högersida.

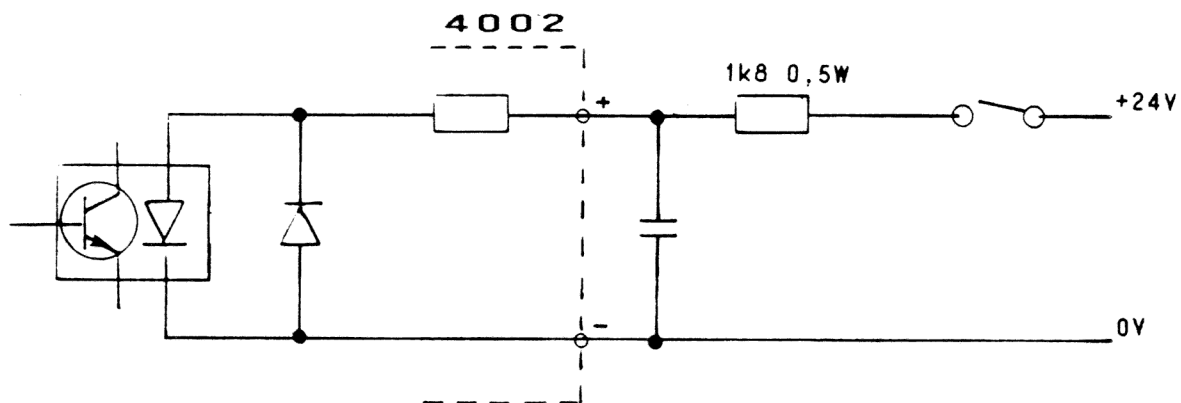
När ESR påverkas och analogutgången har positiv polaritet (riktning) antar utgången värdet 0V. För att komma ur denna situation måste programmet ändra polaritet. Omvända förhållande gäller för ESL.

Ingångsstegen är optoskilda och dimensionerade för att fungera för 3mA. Men optokopplare åldras så därför rekommenderas en inström på ca 10 mA vid 5 Volt. Vid högre inspänning måste ett serie-motstånd användas för att ge strömbegränsning. Ingångsstegets har även ett polvändningsskydd i form av en diod. Observera att förhållandena kan förändras om läckström uppstår.

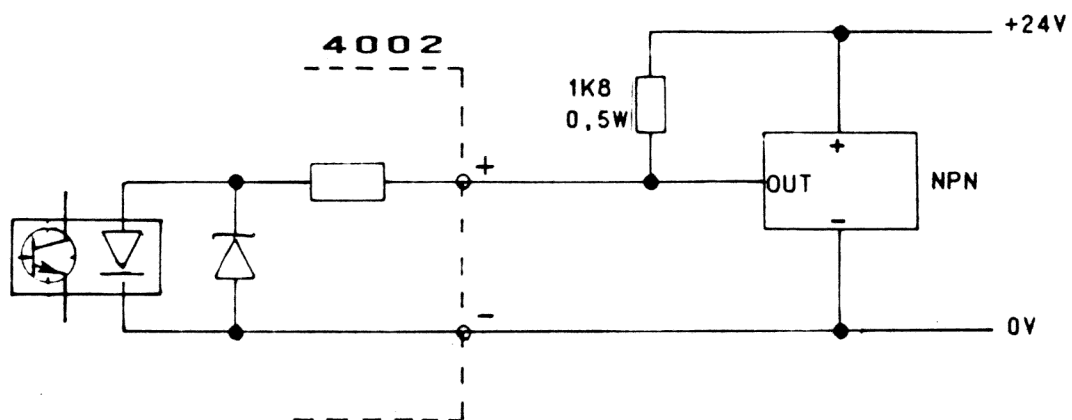


I vila kräver ingångsstegets ström, detta betyder att steget är aktivt om ingen ström finns närvarande. När en ingång inte används måste en ström läggas genom den optokopplare som inte används. Strömmen måste tas från ett yttre kraftaggregat.

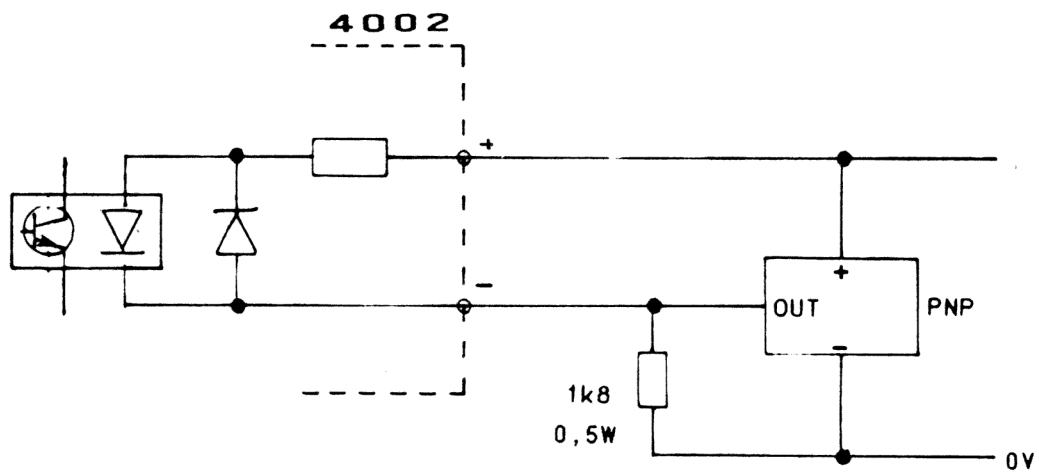
Exempel på en enkel inkoppling av en gränslägesbrytare som har 24V matningsspänning. Induktiva givare används ofta som ändlägesgivare och REF-givare. Givarna, typ NPN eller PNP, antingen sänker eller lyfter i påverkat tillstånd, t ex när referenspunkten har uppnåtts. Det är möjligt att koppla in givare med antingen aktiv sänkning, NPN-givare, eller med aktiv lyftning, PNP-givare, eftersom bägge sidor på ingångsstegets finns tillgängliga i kontakten.



Exempel på en inkoppling av givare av NPN-typ med systemspänning på 24V.



Exempel på inkoppling av givare av PNP-typ med systemspänning på 24V.



Watchdog

En övervakningsfunktion har implementeras och denna kräver att 8748:an klappar den minst varje sekund. I annat fall återstartas programmet, vilket bl a innebär att analogutgången faller till 0 V och enheten åter blir redo för kommandon.

Programvara

Aritmetik

Aritmetiken på kortet 4002 beräknas med binär notation, detta innebär att argumenten till kommandona också är binära. För att få ett tillräckligt stort område för positionering valdes 3 bytes. Därmed sträcker sig det maximala positioneringsintervallet från -8 388 608 till + 8 388 607, eftersom teckenaritmetik används. Observera att den totala sträckan mellan ändlägena inte får överskrida 23 bits och det är användaren som är ansvarig för att detta inte inträffar. Tecknet är valt på så sätt att all positionering över nollpositionen är positiv, se figur.

Hastighetsargumentet (rate) är alltid positivt och ligger i området 13568 - 16383 för den interna F/U-omvandlaren. Den analoga utgången ger maximal utspänning när hastigheten sätts lika med 16383 och reläet stängs av när hastigheten sätts lika med noll. Åkriktningen beräknas av den yttre enheten med hjälp av information om position och storlek på utspänningen. Om tecknet på utspänningen är negativt antas det att motorn går baklänges. Rate kontra utspänningen är en icke-linjär funktion som kommer att beskrivas senare.

OBS! Positionering från minvärde till maxvärde går inte att utföra, -8 388 608 - 8 388 607, eftersom det behövs 23 bitar för att definiera området 0 - 8 388 608.

Beskrivning av portarnas användning

4002 kortet supporterar följande standardportar i DataBoard 4680:

STAT	Innehåller information om processorn och de statiska signalerna.
C1	Används inte.
C2	Förmedlar kommandon till 4002.
C3	Hårdvarureset.
C4	Händelse- och interrupthantering.
UTP	förmedla argument till kommandona
INP	t ex sätta upp ny position (INP) eller avläsa nuvarande position (UTP). Gemensam benämning för dessa två portar är DATA-porten.

STAT

Fem av flaggorna i STAT-porten används för att avkoda den aktuella statusen på ändläges, nödstopps-, referens- och nollsignalerna. Res-terande 3 flaggor ger information från processorn. Ready indikerar när 4002 är klar att ta emot data från någon av portarna DATA, C2 och C4. Användaren måste alltid vänta på att Ready flaggan ska vara satt, för att vara helt säker på att 4002 reagerar på ett korrekt sätt på de olika portarna. Interrupt flaggan visar att ett interrupt är aktivt eller 'pending'. Event enable flaggan visar att Event enable ordet har uppfyllits.

STAT flaggmönster

xxxxxxx

.....1	Interrupt 'pending'
.....1.	Event enable flagga 'pending'
.....1..	Nollpulser
...1....	Nödstopp
..1.....	Ändstopp 1
.1.....	Ändstopp 2
1.....	Enheten klar för att ta emot kommando (Ready)

Exempel:

```
CALL WAITRDY
```

WAITRDY EQU *	Subrutin
INP STAT	Läser STAT-porten
NI 080H	Maskar fram bit Ready
JZS WAITRDY	Väntar på Ready
RET	Return

C2

C2-porten används till att via kommandon styra 4002 till att utföra önskad handling och DATA-porten används till att läsa eller skriva argumenten till dessa kommandon.

Kommandona beskrivs senare.

Exempel: 4002 skall positionera sig 10 pulser åt vänster (negativ riktning). Antag att hastigheten redan är specificerad.

```
CALL WAITRDY  Vänta på att kortet skall
               acceptera kommando.
LI  A, 'N'    Relativ positionering.
OUT C2
LDI HL, -10   Ladda antal pulser.
CALL WAITRDY  Anropa subrutin.
LR  A, L      Minst signifikanta byten
               (byte 0) av antalet
               pulser.

OUT DATA
CALL WAITRDY
LR  A, H      Byte 1.
OUT DATA
CALL WAITRDY
LI  A, -1     Mest signifikanta byten
               (byte 2).

OUT DATA
CALL WAITRDY
LI  A, 'G'
OUT C2        Starta positioneringen.
```

C3

C3 utför reset av 4002. Denna reset är en hårdvarureset och både processorn och all logik på kortet återställs, processorn på 4002 startar om från början av sina program.

C4

Via C4-porten är det möjligt för användaren att använda 4002 i interruptmod och att detektera vissa händelser. Beskrivs i detalj i avsnittet Händelsehantering sid 24 och Interrupt hantering sid 28.

Händelsehantering

Användaren kan med hjälp av händelsehantering sätta upp motorstyrningskortet 4002 så att det reagerar på vissa händelser. En sådan händelse kan t ex vara att den önskade positionen är uppnådd. En av de stora fördelarna med denna möjlighet är att programmeringen i värddatorn förenklas genom att det inte är nödvändigt att kontrollera den aktuella positionen för att se om rätt position är uppnådd. Förutom den beskrivna händelsen kan också noll- och referenshändelserna detekteras, antingen en och en eller båda samtidigt. Slutligen kan ändstopphändelsen detekteras, denna uppstår antingen när en nödstopps- eller ändstoppspuls känns av.

Om någon händelse är redo för aktivering är motsvarande signal i STAT-porten aktiv. Naturligtvis kan mer än en händelse vara redo för aktivering samtidigt, men för att få reda på vilken av de aktiverade händelserna som har inträffat, måste användaren läsa EVENT-ordet. Samtidigt som händelsen inträffar latchas EVENT-ordet och denna latch frisläpps inte förrän användaren skriver till C4-porten (OUT C4 instruktion).

Aktivering av händelse (Event enabling)

För att de olika händelserna ska kunna aktiveras eller inaktiveras måste ett flaggmönster sändas med OUT till C4-porten. Samtidigt placeras EVENT-ordet på DATA-porten (beskrivs senare), latchen för uppdatering frisläpps, flaggan som är satt för position uppnådd i EVENT-ordet tas bort och Event enable flaggan i STAT-porten frisläpps. Endast de händelser kan inträffa vars flagga är aktiv i Event enabling ordet.

Flaggmönster för ordet Event enabling, C4*

..xxxxxx

.....1 Uppnådd position
1. Nollpuls
1.. Referenspuls
1... Reserverad
 ...1.... Reserverad
 ..1..... Ändstopp
 yy..... Interrupt, beskrivs på sid 26

Exempel:

WAITRDY	EQU	*	
	INP	STAT	
	NI	080H	
	JZS	WAITRDY	Vänta på ready.
	LI	A, 1	Redo för aktivering vid uppnådd position.
	OUT	C4	Skriv till C4 porten.

Läsning av EVENT-ordet

EVENT-ordet kan läsas på DATA-porten (INP*) efter det att en OUT C4 instruktion har utfärdats och efter det att den yttre enheten har indikerat på STATporten (bit 1) att den är klar. De sex minst signifikanta flaggorna i EVENT-ordet är reserverade för olika händelser och de har följande flaggmönster.

Flaggmönster i EVENT-ordet, INP*

. XXXXXXXX

```

.....1  Position uppnådd eller passerad
.....1.  Nollpuls detekterad
.....1.. Referenspuls detekterad
....1... Reserverad
...1.... Reserverad
..1..... Ändstopp detekterad
.1..... Reserverad

```

Den mest signifikanta flaggan användas till att detektera om kortet har blivit återställt (reset). Flaggan nollställs med watch out kommandot.

```
1..... Watch out bit, satt vid reset.
```

Exempel: Anlag att uppnådd position är redo för aktivering.

```
WAITEVT EQU *  
INP STAT  
NI 002H Vänta på att en händelse ska inträffa.  
  
JNZS WAITEVT  
  
WATRDY1 EQU *  
INP STAT  
NI 080H  
JZS WAITRDY1 Vänta på Ready.  
LI A, 021H Uppnådd position och ändstopp sätts redo för aktivering.  
  
OUT C4 Skriv till porten.  
  
WAITRDY2 EQU *  
INP STAT  
NI 080H  
JZS WAITRDY2 Vänta på Ready för att läsa EVENT-ordet.  
INP DATA EVENT-ordet läggs i reg A.  
NI 001H Kontrollera att positionen är uppnådd.  
JZ ERROR
```

ANM. Om både noll- (ZERO) och referenssignalerna (REF) är specificerade används en logisk AND funktion på båda, dvs båda signalerna måste vara närvarande vid samma tidpunkt för att en händelse skall inträffa.

Interrupt hantering

I en del applikationer är det för tidskrävande att sätt Ready och Eventflaggorna i STAT-porten, därför erbjuder 4002 användarna en interrupt facilitet. Denna ger användarna möjlighet att ge ett interrupt på Ready- och Event-flaggorna. För värddatorn reducerar detta väntetiden på att 4002 skall ha exekverat kommandona. För att aktivera ett interrupt måste användaren sätta de två mest signifikanta flaggorna i EVENT enable ordet och skriva ut det till C4-porten.

Interrupt flaggmönster, C4*

xx.....

.1..... Interrupt på Ready
 1..... Interrupt på händelse
 ..yyyyyy Event enable flaggor, är
 tidigare beskrivna i avsnittet
 Händelsehantering, se sid 22

Exempel:

WAITRDY	EQU	*	
	INP	STAT	
	NI	080H	
	JZS	WAITRDY	Väntar på Ready.
	LI	A, 081H	Aktivera interrupt vid uppnådd position.
	OUT	C4	Skriv till porten.

Beskrivning av kommando

Nedan följer en beskrivning över de kommando som existerar till 4002. Kommandona skall skrivas till port C2 och eventuella argument till port DATA. För varje kommando anges dessutom ASCII-värdet, binärvärdet och om kommandot har några argument anges dess storlek.

Athome

	ASCII	Binärt	Decimalt
C2	A	0100 0001	65

Kommandot Athome definierar den nuvarande positionen som hemma position. Denna position är den absoluta nollpunkten och den är referens till alla positionerings kommandon.

Athome utförs automatiskt vid uppstart av 4002.

Deviation

	ASCII	Binärt	Decimalt	
C2	D	0100 0100	68	
DATA			Låg byte	Avvikelse In/Ut Binärt
DATA				
DATA			Hög byte	

Kommandot Deviation kan användas både som in- och utinstruktion.

In: Kommandot Deviation läser den relativa differansen mellan den aktuella och den begärda positionen. De efterföljande 3 sekventiella INP DATA ger avvikelsen på binärform.

Ut: Om 4002 har satts redo för aktivering utav kommandot Lock anger Deviation kommandot den maximala (absoluta) avvikelsen från den begärda innan en ompositionering utförs. Avvikelsen anges i de tre sekventiellt efterföljande OUT DATA, denna information används vid Lock hanteringen. En ompositionering m h a Lock utförs på samma sätt som en positionering dvs med kommandona Rate och Slope.

Go

	ASCII	Binärt	Decimalt
C2	G	0100 0111	71

Go kommandot startar upp den positionering som är specificerad av kommandona Number, Position och Rate. Om regleringen är aktiverad kommer den hastighet som är specificerad av Rate kommandot att bytas mot den hastighet som specificeras av kommandot Slope. Biten 'Position uppnådd' i EVENT ordet nollställs. Go kommandot stänger av kommandot Lock.

OBS! Om någon av kommandona Athome, Go, Number, Position (ut) eller Slope anges innan positioneringen är slutförd och efter ett Go kommando kan oförutsedda saker inträffa.

Lock

	ASCII	Binärt	Decimalt
C2	L	0100 1100	76

Kommandot Lock startar funktionen fastlåsning på position. Denna funktion mäter kontinuerligt avvikelsen mellan den begärda och den aktuella positionen och om så behövs startar den en ompositionering. Den maximalt tillåtna avvikelsen, innan en ompositionering sker, definieras av kommandot Deviation. I kommandot Lock innefattas ingen servoreglering utan en reglering kan bara ske inom det definierade intervallet. Om detta intervall är för litet i förhållande till den definierade hastigheten, kan systemet komma i självsvängning.

Manual

	ASCII	Binärt	Decimalt
C2	M	0100 1101	77

Kommandot Manual stänger av regleringen dvs upphäver funktionen hos kommandona On regulation och Lock, 4002 kommer att börja arbeta i en icke-regleringsmod. Vid nästa Go kommando kommer den hastighet att användas som är specificerad av kommandot Rate, default vid uppstart.

Number

	ASCII	Binärt	Decimalt	
C2	N	0100 1110	78	
DATA			Låg byte	Relativ pos. Ut Binärt
DATA				
DATA			Hög byte	

Med kommandot Number specificeras det antal steg som används i den relativa moden. De efterföljande tre sekventiella OUT DATA anger den nya relativa positionen men ingen åtgärd kommer att vidtagas förrän ett Go kommando anges.

On regulation

	ASCII	Binärt	Decimalt
C2	O	0100 1111	79

Kommandot On regulation startar upp regleringen och 4002 börjar arbeta i regleringsmod. Vid nästa Go kommando kommer hastigheten att hämtas från den slope tabell som är specificerad av kommandot Slope. Detta kommando sätter också Rate kommandot till sitt maximala värde, för att försäkra sig om att alla hastigheter hämtas från slope tabellen. Se Slope kommandot.

Position

	ASCII	Binärt	Decimalt	
C2	P	0101 0000	80	
DATA			Låg byte	Absolut pos. In/Ut Binärt
DATA				
DATA			Hög byte	

Kommandot Position anger att 4002 börjar arbeta i absolutmod. De tre sekventiellt efterföljande INP DATA anger den aktuella positionen. Vid skrivning till 4002 anger de tre sekventiellt efterföljande OUT DATA målets absoluta position och denna position anges relativt nollpositionen (hemma läget). Den yttre enheten startar inte upp positioneringen förrän kommandot Go anges.

Anm. Nollpositionen (hemma läget) definieras med kommandot Athome.

Quit

	ASCII	Binärt	Decimalt
C2	Q	0101 0001	81

Med kommandot Quit avbryts nuvarande positionering och den nya hastigheten sätts till noll dvs positioneringsregistret nollas. Detta kommando påverkar inte den hastighet som är specificerad av Rate kommandot.

Rate

	ASCII	Binärt	Decimalt	
C2	R	0101 0010	82	
DATA			Låg byte	Läsa/Sätta hastighet
DATA			Hög byte	In/Ut Binärt

Kommandot Rate används till att både läsa aktuell hastighet och sätta hastigheten i en icke-regleringsmod (se kommandot Manual). De 2 sekventiellt efterföljande INP DATA används för indata och de anger den aktuella hastigheten i binärnotation, om den är negativ kommer motorn att arbeta baklänges.

När en hastighet anges används de 2 sekventiellt efterföljande OUT DATA för utdata, totalt används endast 14 bits, 8 bits i låg byte och 6 bits i hög byte. Om hastigheten = 0 anges frisläpps reläet. Den specificerade hastigheten laddas in omedelbart och det finns inga restriktioner för hur detta kommando kan användas under positioneringen. Om ett stopp, ändläge- eller nödstopp, inträffar sätts hastigheten inte till noll utan användaren måste själv se till att rätt åtgärd vidtages, reläet frisläpps i detta fall av hårdvaran.

Hastigheten kontra utspänningen är en icke-linjär funktion. Om en begränsad hastighet skall användas trots att reglering enligt slope tabell är vald måste Rate anges efter On regulation.

Slope

	ASCII	Binärt	Decimalt	
C2	S	0101 0011	83	
DATA				Hastighet Ut 240 bytes Binärt
DATA				
DATA				

Med kommandot Slope kan användaren själv definiera olika lutningar på hastighetskurvan (slopes). Denna möjlighet används normalt om höga hastig-

heter skall användas och den initiala lasten på motorn förhindrar en ögonblicklig positionering. I dessa fall accelereras lasten från vila till maximal hastighet och retarderas därefter till stopp. Användaren definierar antingen ett intervall från start (acceleration) eller till stopp (retardation) och den hastighet som skall användas i intervallet.

I regleringsmod startas 4002 upp från vila, hastigheten ökas, enligt specifikationen i slope tabellen, tills den maximala hastigheten uppnås eller tills den hastighet uppnås som är specificerad av kommandot Rate. Kortet beräknar själv vid vilken punkt retardationen skall börja. Lutningen på accelerations- och retardationsramporna behöver nödvändigtvis inte vara lika. Det sista intervallet i slope tabellen innehåller den maximala hastighet som motorn kan arbeta med. Denna hastighet används när den relativa positionen befinner sig utanför slope tabellen.

Det totala antalet intervall som kan specificeras är 48 stycken och varje intervall innehåller 5 bytes, detta innebär att de 240 sekventiellt efterföljande bytes innehåller en komplett tabell, se Beskrivning över slope tabellens innehåll sid 36.

Anm. Användaren ansvarar för att innehållet i slope tabellen inte ändras under positioneringen.

Version

	ASCII	Binärt	Decimalt
C2	V	0101 0110	86
DATA			

Versionsnr
In, Binärt

Kommandot Version placerar versionsnumret på det aktuella 4002 programmet på porten INP DATA, för att läsas vid nästa inläsning.

Watch out

	ASCII	Binärt	Decimalt
C2	W	0101 0111	87

Kommandot Watch out nollställer den mest signifikanta flaggan i Event-ordet. Eftersom denna flagga alltid sätts vid start av 4002, kan den användas till att testa om 4002 har återstartats.

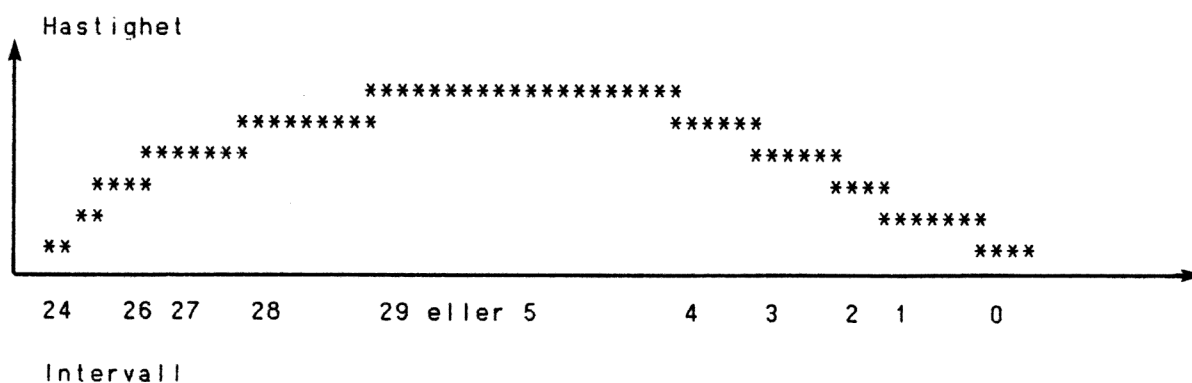
Anm. Läsning av Event-ordet se kap Händelse hantering. Vid läsning/skrivning av argumenten behandlas den minst signifikanta byten först och sist den minst signifikanta.

Beskrivning över slope tabellens innehåll

En av fördelarna med 4002 är möjligheten att specificera olika lutningar (slopes) för acceleration och retardation. Denna information lagras i en tabell hos 4002. Användaren specificerar varje intervalls längd och hastighet. Positionen för varje intervall definieras med 3 bytes och hastigheten med 2 bytes, totalt för varje intervall åtgår det 5 bytes och antalet intervall som kan specificeras är 48 stycken, sammanlagt åtgår det 240 bytes för en komplett tabell. Interpositionen är alltid positiv.

Datastrukturen ser ut på följande sätt:

Adr	Intervall	Användning
0	0	Hastighet vid retardation.
2		Intervall position (relativt slutet)
5	1	Hastighet vid retardation.
7		Intervall position (relativt slutet).
10	2	
	!	
	!	
	!	
120	25	Hastighet vid acceleration.
122		Intervall position (relativt slutet).
125	24	Hastighet vid acceleration.
127		Intervall position (relativt slutet).
130	26	
	!	
	!	
	!	
235	47	
237		Maximal hastighet om utanför tabellen.



Figur: Diagram över intervall kontra hastigheten.

Anm. Den maximala hastighet som används är den hastighet som är lägst av max hastigheten i slopetabellen och den hastighet som specificerats i kommandot Rate.

Exempel på hur en lutningstabell (slope tabell) kan sättas upp.

Assembler:

```

CALL WAITDRY
LI A, 'S'      Säg till yttre enhet att
                spara slopetabellen.

OUT C2
LA HL, TABLE
LI B, 240      Antal bytes som skall
                sparas.

STORELOP EQU *

CALL WAITDRY   Vänta på mottagande av
                nästa byte.

L A, (HL)
INCD HL
OUT DATA
DJNZ STORELOP
RET

*
TABLE DA 1000   Hastighet i första
                intervallet, retard.
      DB 100,0,0 Position.
      DA 1000   Hastighet i andra
                intervallet, retard.
      DB 200,0,0 Position.

      DA ....
      DB ....
      .
      .

```

Basic:

Följande program sparar en lutningstabell (slope tabell). Lutningarna för acceleration och retardation är olika. Funktionen FNWaitdry väntar på att signalen Ready ska bli aktiv. Antag dessutom att kortet redan har blivit valt med OUT 1,cs.

```
DEF FNStoreslope
```

```
Q7=FNWAITRDY : OUT 3,ASCII("S")! Give slope command.
```

```
FOR I=1 TO 2
```

```
  IF I=1 RESTORE 23341 ELSE RESTORE 23351
```

```
  FOR MEMADR=1 TO 24
```

```
    READ SPEED, POSITION
```

```
    Q71=FNWAITRDY : OUT 0, SPEED
```

```
    Q71=FNWAITRDY : OUT 0, SPEED/256
```

```
    Q71=FNWAITRDY : OUT 0, 0 ! ZERO IN LSB OF POSITION
```

```
    Q71=FNWAITRDY : OUT 0, POSITION
```

```
    Q71=FNWAITRDY : OUT 0, POSITION/256
```

```
  NEXT MEMADR
```

```
NEXT I
```

```
RETURN
```

```
FNEND
```

```
DATA 14288,1,15892,6,16002,8,16116,15,16178,30,16216,35
```

```
DATA 16244,40,16262,45,16278,50,16288,60,16298,70,16305,85
```

```
DATA 16312,100,16317,115,16322,130,16326,145,16330,160
```

```
DATA 16333,180,16336,200,16337,225,16338,250,16339,285
```

```
DATA 16340,320,16341,370
```

```
!
```

```
! ACC SLOPE
```

```
DATA 14288,1,15892,12,16002,16,16116,22,16178,30,16216,35
```

```
DATA 16244,40,16262,50,16278,60,16288,70,16298,80,16305,105
```

```
DATA 16312,130,16317,150,16322,180,16326,215,16330,250
```

```
DATA 16333,290,16336,330,16337,365,16338,400,16339,450
```

```
DATA 16340,500,16341,600
```

Exempel på en tabell över utspänning kontra rate
för ett intrimmat 4002-kort.

Volt	Rate
0.1	13799
0.025	14051
0.05	14339
0.075	14565
0.1	15745
0.15	15014
0.2	15207
0.25	15351
0.3	15463
0.35	15547
0.4	15628
0.5	15750
0.75	15921
1.0	16021
1.5	16130
2.0	16189
2.5	16225
3.0	16250
3.5	16269
4.0	16282
4.5	16293
5.0	16302
5.5	16310
6.0	16316
6.5	16321
7.0	16325
7.5	16330
8.0	16333
8.5	16335
9.0	16338
9.5	16341
10.0	16345

Komplett tabell = Trimresultat medföljer varje kort

Uppstart av 4002

Den uppstart som beskrivs här nedan är en funktionstest där Basic II används i direktmod. OBS! Detta är ingen generell uppstart.

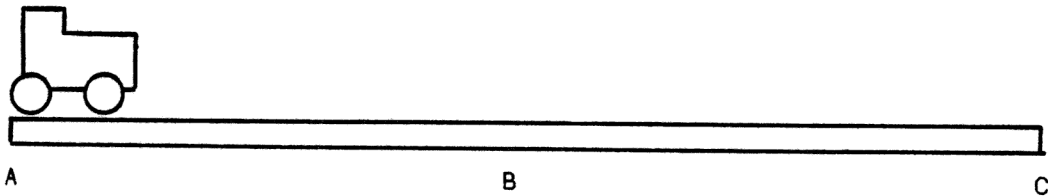
1. Kontrollera spänningen på busskontakten, stift 2 = 0 V och stift 31 = 5 V. Observera att de tre opto-isolerade ingångarna
 - EMS nödstopp
 - ESL ändstopp, vänster
 - ESR ändstopp, höger
 har normalt drifttillstånd spänningssatta. Detta betyder att vissa stift i kontakt 3P måste vara spänningssatta vid provkörning och drift, stiftnummer inom parentes:
 0V - EMS- (10) ESL- (11) ESR- (12)
 +5V - EMS+ (23) ESL+ (24) ESR+ (25)
 Se vidare kap Referensingångar.
2. Välj kort genom att skriva OUT 1,CS där CS står för kortets kortvalsadress. Lysdioden på kortet kommer att lysa upp när rätt kortvalsadress har angivits.
3. Läs STAT-porten och kontrollera att den mest signifikanta flaggan är satt, indikerar att enheten är klar.
4. Skriv OUT 5,0 och läs därefter STAT-porten för att kontrollera att enheten är klar.
5. Läs DATA-porten (EVENT-ordet läses). Det värde som läses ska vara 80H, indikerar att systemet är återställt.
6. När enheten har indikerat att den är klar (punkt 3), skriv OUT 3,57H, kommandot Watch out placeras på C2-porten.
7. Vänta på att enheten är klar, skriv OUT 5,0. Vänta på att enheten är klar, läs DATA-porten. Svaret ska vara lika med noll, indikerar att watch out flaggan i EVENT-ordet är nollställd.
8. Om du har lyckats genomföra alla punkter utan några problem, kan du mata in vilket kommando du vill och rätt åtgärd kommer att bli utförd.

```

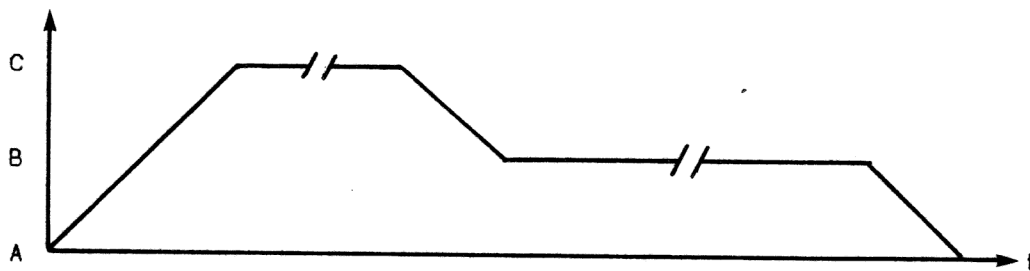
RST=INP(7)
OUT=OUT 0,x%
CS=OUT 1,x%
C1=OUT 2,x%
C2=OUT 3,x%
C3=OUT 4,x%
C4=OUT 5,x%
STAT=INP(1)
  
```

Programexempel 4002

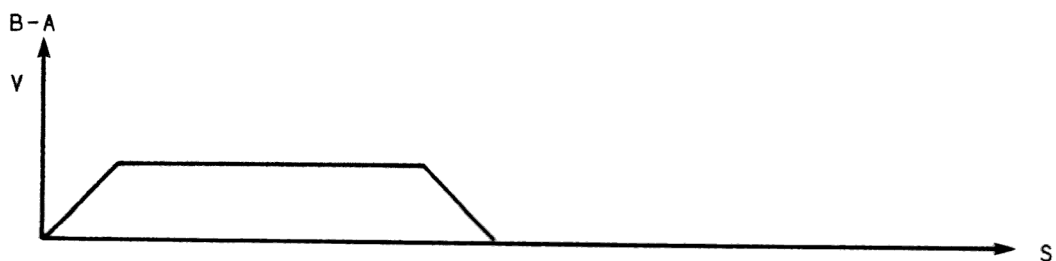
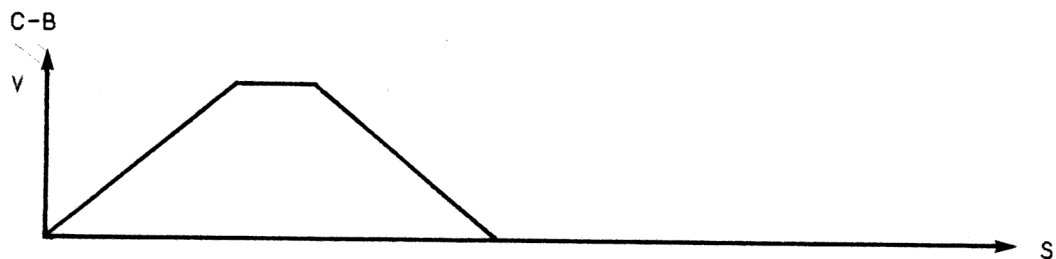
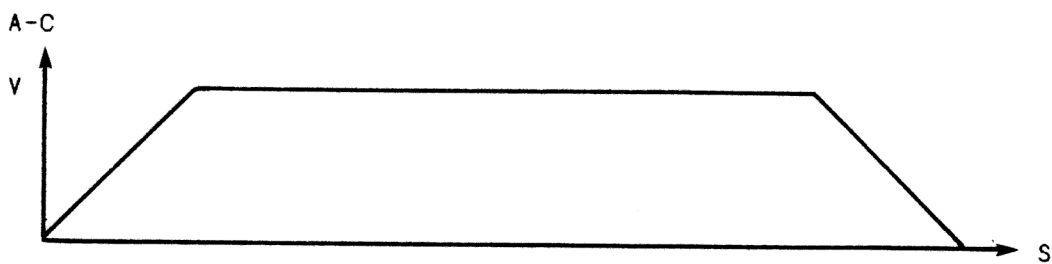
I detta tänkta exempel skall en transfervagn flyttas enligt nedanstående väg/tid diagram.



$$\text{Avstånd A-B} = \text{B-C} = (\text{A-C})/2$$



De olika hastighetsprofilerna vid förflyttning mellan punkterna A-C, C-B och B-A har följande utseende:



Vid förflyttningen mellan A-C utnyttjas hög acceleration och retardation samt att vagnen tillåts åka med full fart.

Vid förflyttningen mellan C-B och B-A har acceleration och retardation minskats.

Maximalfart tillåts mellan, A-C och C-B men däremot har en begränsad toppfart angivits för sträckan B-A.

Verbal beskrivning

En verbal beskrivning av styrproblemet kan låta så här:

- 1 - Den tomma vagnen står i hemmaläge. (A)
- 2 - Manuell startorder ges, i ett verkligt fall utgör startorder och alla kvittenser av gränsläges signaler, tryckknappar etc. I detta exempel ges kvittens vid nedtryckning av en godtycklig tangent på bildskärmsterminalen.
- 3 - Förflyttning A-C med tom vagn tillåter hög acceleration och retardation samt hög hastighet.
- 4 - Aktivera positionskontroll så att lastläge (C) hålls låst inom angiven tolerans.
- 5 - Invänta lastningskvittens (simulerad) vid lastläge (C).
- 6 - Förflyttning C-B sker med lägre acceleration och retardation eftersom vagnen delvis är lastad. Full fart är däremot tillåten.
- 7 - Invänta lastningkvittens (simulerad) vid lastläge (B).
- 8 - Förflyttning B-A med oförändrad acceleration och retardation, maxhastigheten är sänkt.
- 9 - Hemmaläge (A)

Program

Detta problem är löst i de efterföljande exemplen med:

- Basic II
- Assembler

Programmen är körda i DataBoard 4 MHz system under OS8 mot 4002-simulator.

```

1000 ! *****
1020 ! *
1040 ! * DEMO PROGRAM FOR 4002 BOARD.
1060 ! * CARD ADDRESS SHOULD BE 15Q.
1080 ! *
1100 ! * Ver 1.00 84-03-27 CE First try.
1120 ! * Ver 1.01 84-11-27 TH Minor changes.
1140 ! *
1160 Q7=FNInitdemo ! SET UP SOME VARIABLES.
1180 WHILE 1
1200 !
1220 Q7=FNMaxspeed(63*256+240) ! SET MAX SPEED.
1240 !
1260 Q7=FNStoreslope(1) ! SLOPE TYPE 1 FIRST.
1280 !
1300 Q7=FNGoPosition(1000) ! POSITION TO 1000
1320 !
1340 Q7=FNOut(C2,ASCII("L"))
1360 !
1380 Q7=FNQuit
1400 !
1420 Q7=FNOut(C2,ASCII("M")) ! SLOPE AND LOCK OFF.
1440 !
1460 Q7=FNStoreslope(2) ! STORE SLOPE TYPE 2 WHEN GOING BACK.
1480 !
1500 Q7=FNMaxspeed(63*256+240) ! SET MAX SPEED.
1520 !
1540 Q7=FNGoPosition(500) ! POSITION BACK TO 500.
1560 !
1580 Q7=FNQuit
1600 !
1620 Q7=FNMaxspeed(63*256) ! SET NEW MAX SPEED
1640 !
1660 Q7=FNGoPosition(0) ! BACK TO START POINT.
1680 !
1700 Q7=FNQuit
1720 !
1740 WEND
1760 ! FNEND
1780 ! *****
1800 ! *
1820 ! * DO POSITIONING TO REQUESTED POSITION.
1840 ! *
1860 DEF FNGoPosition(Position)
1880 Q7=FNOut(C2,ASCII("P")) ! ABSOLUTE POSITIONING.
1900 Q7=FNOut(Data,0) ! LOW BYTE OF POS ARGUMENT.
1920 Q7=FNOut(Data,Position)
1940 Q7=FNOut(Data,Position/256) ! HIGH BYTE.
1960 Q7=FNOut(C4,1+32) ! ENABLE IN POSITION AND END-STOP EVENTS.
1980 Event=FNInp(Data) ! DUMMY READ TO GET RID OF OLD EVENTS.
2000 Q7=FNOut(C2,ASCII("G")) ! START POSITIONING.
2020 Event=0

```



```

2040  WHILE (Event AND 1)=0 ! WAIT UNTIL POSITION REACHED.
2060      WHILE (INP(Stat) AND 2)=0 : WEND
2080      Q7=FNOut(C4,1+32) ! READ EVENT WORD.
2100      Event=FNInp(Data)
2120      WHILE Event AND 32 ! END STOP OCCURED.
2140          Status=FNInp(Stat) ! READ STATUS
2160          ; CUR(10,20) SPACE$(20)
2180          IF Status AND 64 ; CUR(10,20) "LEFT MARGIN STOP."
2200          IF Status AND 32 ; CUR(10,20) "RIGHT MARGIN STOP."
2220          IF Status AND 16 ; CUR(10,20) "EMERGENCY STOP."
2240          Event=Event AND (-1-32)
2260      WEND
2280  WEND
2300  RETURN F
2320 FNEND
2340 ! *****
2360 ! *
2380 ! * SET MAXIMUM POSITION SPEED.
2400 ! *
2420 DEF FNMaxspeed(Maxspeed)
2440  Q7=FNOut(C2,ASCII("R")) ! GIVE COMMAND.
2460  Q7=FNOut(Data,Maxspeed) ! LOW BYTE.
2480  Q7=FNOut(Data,Maxspeed/256) ! HIGH BYTE.
2500  RETURN F
2520 FNEND
2540 ! *****
2560 ! *
2580 ! * WAIT ON USER QUIT
2600 ! *
2620 DEF FNQuit
2640  ; CUR(10,20) SPACE$(20)
2660  ; CUR(10,20) CHR$(7)+"IN POSITION."
2680  GET Q7$ ! WAIT USER KEY.
2700  ; CUR(10,20) SPACE$(20)
2720  RETURN F
2740 FNEND
2760 ! *****
2780 ! *
2800 ! *      WAIT FOR READY SIGNAL UNTIL TIMEOUT.
2820 ! *
2840 DEF FNWaitrdy
2860  Q7=0 ! RESET TIMEOUT COUNTER.
2880  WHILE Q7<>100
2900      IF INP(Stat) AND 128 THEN RETURN F ! UNIT READY.
2920      Q7=Q7+1 ! UPDATE TIMEOUT COUNTER.
2940  WEND
2960  ; "FATAL ERROR !! UNIT NEVER READY !" : STOP ! RETURN T
2980 FNEND

```

```

3000 ! *****
3020 ! *
3040 ! *          STORE ACCELERATION SLOPE.
3060 DEF FNStoreslope(Slope)
3080   Q7=FNOut(C2,ASCII("S")) ! GIVE COMMAND.
3100   FOR I=1 TO 2
3120     IF I=1 RESTORE 3420 ELSE RESTORE 3560
3140     FOR Memadr=1 TO 24
3160       READ Speed,Position
3180       IF Slope=2 THEN Position=Position*2 ! DOUBLE POS ON SLOPE 2.
3200       Q7=FNOut(Data,(Speed AND 255))
3220       Q7=FNOut(Data,Speed/256)
3240       Q7=FNOut(Data,0)
3260       Q7=FNOut(Data,Position)
3280       Q7=FNOut(Data,SWAP%(Position))
3300     NEXT Memadr
3320   NEXT I
3340   Q7=FNOut(C2,ASCII("O")) ! TURN REGULATION ON.
3360   RETURN F
3380 FNEND
3400 ! RET SLOPE
3420 DATA 14000,1,14500,6,15002,8,15216,15,15678,30,16016,35
3440 DATA 16144,40,16262,45,16278,50,16288,60,16298,70,16304,85
3460 DATA 16312,100,16317,115,16322,130,16326,145,16330,160
3480 DATA 16332,180,16334,200,16336,225,16338,250,16340,285
3500 DATA 16342,320,16344,370
3520 !
3540 ! ACC SLOPE
3560 DATA 14000,1,14500,12,15402,16,15682,22,15778,30,15900,35
3580 DATA 16000,40,16262,50,16284,60,16292,70,16298,80,16304,105
3600 DATA 16312,130,16316,150,16322,180,16326,215,16330,250
3620 DATA 16332,290,16334,330,16336,365,16338,400,16340,450
3640 DATA 16342,500,16344,600
3660 !
3680 ! *****
3700 ! *
3720 ! * SOME HELP ROUTINES.
3740 ! *
3760 DEF FNOut(Strobe,Data)
3780   Q7=FNWaitrdy
3800   OUT Strobe,Data
3820   RETURN F
3840 FNEND
3860 ! *
3880 ! *
3900 ! *
3920 DEF FNInp(Strobe)
3940   Q7=FNWaitrdy
3960   RETURN INP(Strobe)
3980 FNEND

```

```
4000 ! *****
4020 ! *
4040 ! *          INIT DATA
4060 ! *
4080 DEF FNInitdemo
4100   F=0 : T=-1
4120   ; CHR$(26) ! CLEAR SCREEN
4140   LET Data=0 : Cs=1 : C1=2 : C2=3 : C3=4 : C4=5 : Stat=1
4160   OUT Cs,13 ! DO CARD SELECT.
4180   Q7=FNOut(C2,ASCII("I")) ! INIT BOARD
4200   Q7=FNOut(C2,ASCII("D")) ! SET UP MAXIMUM DEVIATION.
4220   Q7=FNOut(Data,25) : Q7=FNOut(Data,0) : Q7=FNOut(Data,0)
4240   RETURN F
4260 FNEND
```

```

DEMO4002  PROG  ** DEMONSTRATION FOR 4002 BOARD. **
*
*
*      VER 1.00  CE      84-03-27
*      VER 1.01  TH      84-11-27
*
*      DEMONSTRATION PROGRAM FOR 4002 BOARD
*      RUNNING UNDER OS8 WITH FLAG TEST.
*
*      CARD ADDRESS SHOULD BE 15Q.
*
*      RADIX 16
*      SQUEZ
*      EJECT
*
*      START UP.
*
START      =      *
CALL      INIT
MAINLOOP   =      *
LDI        HL,63*256+240      SET MAX SPEED.
CALL      MAXSPEED
LI         A,1
CALL      STORESLO           STORE SLOPE TYPE 1.
LDI        HL,1000           DO POSITIONING TO POS 1000.
CALL      GOPOSIT
LDI        HL,C2:'L'         TURN LOCK ON.
CALL      OUTSTROB
CALL      QUIT               WAIT FOR USER KEY.
LDI        HL,C2:'M'         SET MANUAL MODE.
CALL      OUTSTROB           AND LOCK OFF TOO.
LI         A,2               STORE SLOPE TYPE 2.
CALL      STORESLO
LDI        HL,63*256+240      SET MAX SPEED.
CALL      MAXSPEED
LDI        HL,500            GO TO POSITION 500
CALL      GOPOSIT
CALL      QUIT
LDI        HL,63*256          SET NEW MAX SPEED.
CALL      MAXSPEED
LDI        HL,0
CALL      GOPOSIT            BACK TO POSITION 0.
CALL      QUIT
JMP        MAINLOOP
EJECT
*
*      DO POSITIONING TO REQUESTED POSITION IN HL.
*
GOPOSIT    =      *
PUSH      HL
LDI        HL,C2:'P'         ABSOLUTE POSITIONING.
CALL      OUTSTROB
LDI        HL,DATA:0         0 IN LOW BYTE OF ARGUMENT.
CALL      OUTSTROB

```

	POP	HL	
	PUSH	HL	MEDIMUM BYTE OF ARGUMENT.
	LI	H,DATA	
	CALL	OUTSTROB	
	POP	HL	
	LR	L,H	MOST SIGNIFICANT BYTE OF ARGUMENT.
	LI	H,DATA	
	CALL	OUTSTROB	
	LDI	HL,C4:32+1	ENABLE IN POSITION AND END-STOP EVENTS.
	CALL	OUTSTROB	
	LI	H,DATA	
	CALL	INSTROB	DUMMY READ TO GET RID OF OLD EVENETS.
	LDI	HL,C2:'G'	SAY GO.
	CALL	OUTSTROB	
GOLOOP	=	*	
	INP	STAT	
	NI	2	WAIT FOR EVENT.
	JZS	GOLOOP	
*			
	LDI	HL,C4:32+1	EVENT OCCURED,READ EVENT WORD.
	CALL	OUTSTROB	
	LI	H,DATA	
	CALL	INSTROB	
	TBT	32,A	CHECK ENDSTOP.
	JZS	NOENDSTO	
*			
	PUSH	A	SAVE FOR IN POSITION RESEARCH.
	INP	STAT	WHAT KIND OF END-STOP?
	LA	HL,S.LEFTM	
	TBT	64,A	
	CNZ	PRINTSTR	
	LA	HL,S.RIGHTM	
	TBT	32,A	
	CNZ	PRINTSTR	
	TBT	16,A	
	LA	HL,S.EMERG	
	CNZ	PRINTSTR	
	POP	A	
NOENDSTO	=	*	
	NI	1	IN POSITION?
	JZS	GOLOOP	NO !
*			
	RET		
	EJECT		
*			
*			
*			
	SET	MAXIMUM SPEED.	
MAXSPEED	=	*	
	PUSH	HL	
	LDI	HL,C2:'R'	GIVE COMMAND.
	CALL	OUTSTROB	
	POP	HL	
	PUSH	HL	
	LI	H,DATA	

```

CALL OUTSTROB      WRITE ARGUMENT.
POP HL
LR L,H
LI H,DATA
CALL OUTSTROB
RET
EJECT

*
*
*
QUIT = *
LA HL,S.CLEAR
CALL PRINTSTR      CLEAR SCREEN.
LA HL,S.INPOS
CALL PRINTSTR      SAY 'IN POSITION,'
SVC 1,S1RDKEY      READ ONE KEY FROM KEYBOARD.
LA HL,S.CLEAR
CALL PRINTSTR
RET
EJECT

*
*
*
WAITRDY = *
PUSH HL
LDI HL,10000      SET TIME OUT COUNTER.
WAITLOOP = *
INP STAT
NI 128
JNZS WAITEXIT    UNIT READY.

*
DECD HL
LR A,L
OR H
JNZS WAITLOOP

*
WAITEXIT = *
POP HL
RNZ      EXIT IF NOT TIMEOUT.
LA HL,S.FATAL      FATAL ERROR,DEVICE NEVER READY.
CALL PRINTSTR
SVC 6,S6CAN      ABORT JOB !!!!!!!!!!!!!!!
EJECT

*
*
*
STORESLO = *
STORE ACCECLARATION SLOPE.

*
*
*
LDI HL,C2:'S'
CALL OUTSTROB
LI C,1

```

```

STORE1  =      *
         LR     A,C
         CI     1
         LA     DE,SLOPEDEC      DECELERATING SLOPE?
         JZS    STORE2

*
         LA     DE,SLOPEACC
STORE2  =      *
         LI     B,24              24 INTERVALS.
         LI     H,DATA
STORE3  =      *
         L      A,(DE)
         INCD   DE
         LR     L,A
         CALL   OUTSTROB          SPEED LOW BYTE.
         L      A,(DE)
         INCD   DE
         LR     L,A
         CALL   OUTSTROB          SPEED HIGH BYTE.
         LI     L,0
         CALL   OUTSTROB          POSITION LOW BYTE = 0.
         L      A,(DE)
         INCD   DE
         LR     L,A
         CALL   OUTSTROB          POSITION MEDIUM BYTE.
         L      A,(DE)
         INCD   DE
         LR     L,A
         CALL   OUTSTROB          POSITION HIGH BYTE.
         DJNZ   STORE3            LOOP ALL 24 INTERVALS.

*
         INCR   C
         LI     A,3
         CR     C                  BOTH SLOPES, DEC AND ACC.
         JNZS   STORE1

*
         LDI    HL,C2:'0'         REGULATION ON.
         CALL   OUTSTROB
         RET

*
*
*
SLOPEDEC =      *
         DA     14000,1,14500,6,15002,8,15216,15,15678,30,16016,35
         DA     16144,40,16262,45,16278,50,16288,60,16298,70,16304,85
         DA     16312,100,16317,115,16322,130,16326,145,16330,160
         DA     16332,180,16334,200,16336,225,16338,250,16340,285
         DA     16342,320,16344,370

```

```

SLOPEACC = *
DA 14000,1,14500,12,15402,16,15682,22,15778,30,15900,35
DA 16000,40,16262,50,16284,60,16292,70,16298,80,16304,105
DA 16312,130,16316,150,16322,180,16326,215,16330,250
DA 16332,290,16334,330,16336,365,16338,400,16340,450
DA 16342,500,16344,600
EJECT

*
* SOME HELP ROUTINES.
*
OUTSTROB = *
PUSH BC
PUSH HL
CALL WAITRDY          DEVICE MUST BE READY.
POP HL
LR C,H
OUTR L
POP BC
RET

*
*
*
INSTROB = *
PUSH BC
PUSH HL
CALL WAITRDY          DEVICE MUST BE READY BEFORE READING.
POP HL
LR C,H
INPR A
POP BC
RET
EJECT

*
* INIT SOME PARAMETERS.
*
INIT = *
SVC 7,S7ASGCON        ASSIGN CONSOLE.
SVC 1,S1PRINT          CLEAR SCREEN.
LI A,13
OUT CS                DO CARD SELECT.
LDI HL,C2:'I'          INITIALIZE BOARD.
CALL OUTSTROB
LDI HL,C2:'D'          SET UP MAX DEVIATION FOR LOCK FUNCTION.
CALL OUTSTROB
LDI HL,DATA:25          ALLOW 25 PULSES BEFORE REPOSITIONING.
CALL OUTSTROB
LDI HL,DATA:0
CALL OUTSTROB
CALL OUTSTROB
RET
EJECT

```



```

*
*      PRINT ROUTINE.
*
PRINTSTR =      *
          L      A,(HL)
          ST      A,S1PRINT+S1.BSZ  SET UP LENGTH.
          INCD    HL
          STD      HL,S1PRINT+S1.BAD
          LI      A,S1F.WRIT+S1F.RND+S1F.FASC  RANDOM BIT FOR CURSOR ADDRESSING.
          ST      A,S1PRINT+S0.FC
          SVC      1,S1PRINT
          RET

*
S1PRINT  =      *
          DA      S1F.WRIT+S1F.IBIN+S1F.RND  CURSOR ADDRESSING.
          DA      1                          LU.
          DA      FFSTR
          DA      1                          LENGTH.
          DA      0                          SIZE.
          DA      800AH,8014H                CURSOR ADDRESS.

*
FFSTR    DB      26                          CLEAR SCREEN FOR ADM3A.
          EJECT

*
*      SOME SVC DATA AND OTHER IMPORTANT STUFF.
*
DATA      =      0
STAT      =      1
CS        =      1
C1        =      2
C2        =      3
C3        =      4
C4        =      5

*
S7ASGCON  DA      S7F.ASGN,1,CONFD,0,0,0,0,0
CONFD     DB      'CON '
          DMB      24,' '

*
S1RDKEY   DA      S1F.READ+S1F.IBIN,1,READBUF,1,0,0,0,0
READBUF   DB      0

*
S6CAN     DA      S6F.CAN,0,0,0,0,0,0,0

*
S.CLEAR   DB      S.CLEARE--1
          DMB      20,' '
S.CLEARE  =      *

*
S.EMERG    DB      S.EMERGE--1
          DB      'EMERGENCY STOP.'
S.EMERGE  =      *

*
S.LEFTM   DB      S.LEFTME--1
          DB      'LEFT MARGIN.'

```

```
S.LEFTME  =      *
*
S.RIGHTM  DB      S.RIGHTE--1
          DB      'RIGHT MARGIN.'
S.RIGHTE  =      *
*
S.FATAL   DB      S.FATALE--1
          DB      'FATAL ERROR, UNIT NOT READY!!!!!!'
S.FATALE  =      *
*
S.INPOS   DB      S.INPOSE--1
          DB      'IN POSITION.'
S.INPOSE  =      *
          END      START
```

Kommandon för att kunna köra assemblerprogrammet under OS8.

Källfilen antas heta DEMO4002_ASM .

```
-Asmz,,10 DEMO4002_ASM,DEMO4002_OBJ,,pr:   eller tex ,,con:  
<ASMZ Z80 Macro Assembler Rx-xx>  
<End of task 0>
```

```
-estab,,10                                starta      ren  
<ESTlänk AB Ver x-xx>
```

```
print pr:                                om lista önskas  
rel                                       laddas under OS8  
inc DEMO4002_OBJ                         fil som asmz skapat  
lib,a OS40BJ.M4DEFLB/O                  innehåller variabler  
ta DEMO4002                             namn på slutresultat  
check                                    kontrollera  
end                                       tack & adjö  
<End of task 0>  
-  
-DEMO4002                               programmet startas
```

Tekniska data

Matningsspänning: + 5V 0,6A
+ 12V 20mA
- 12V 10mA

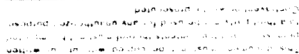
Arbetstemperatur: +10 - +40 grader C

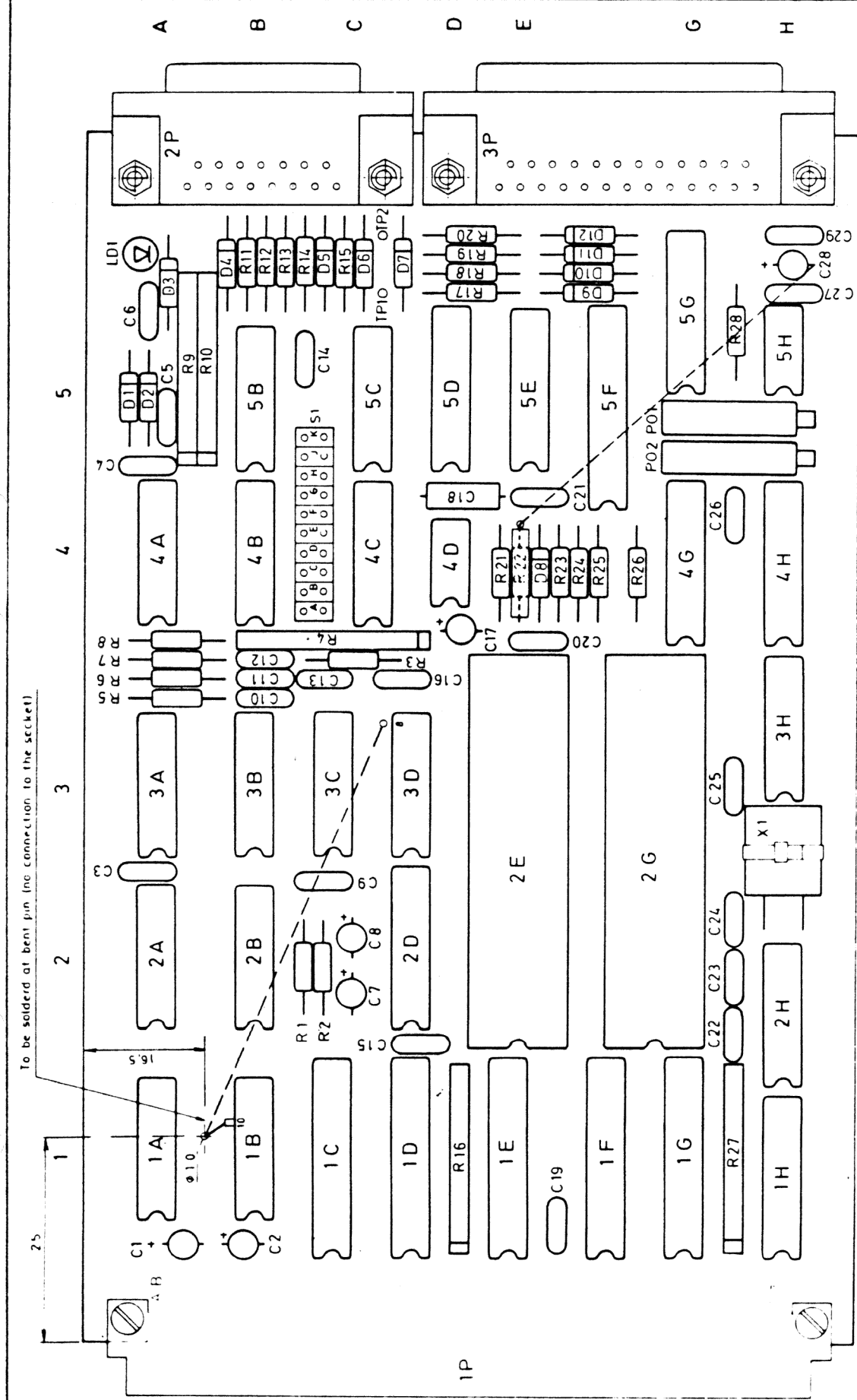
Storlek: 160x100 Europakort standard

Anslutningsdon: Till DataBoard-bussen 1 st 64 pol Europadon
Till inkrementalgivare DB 15/p
Till analogutgång och ändstopp DB 25/p

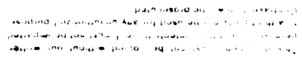
Ingångar: Pulsgivare 5V Diff
Statusgivare 5V Optoisolerat

Utgångar: Motorstyrning, analog 10mV - 10V
Motorstyrningsfrekvens 140Hz - 10kHz
(Option 25Hz - 100kHz)TTL
Reläväxling för analogutgången

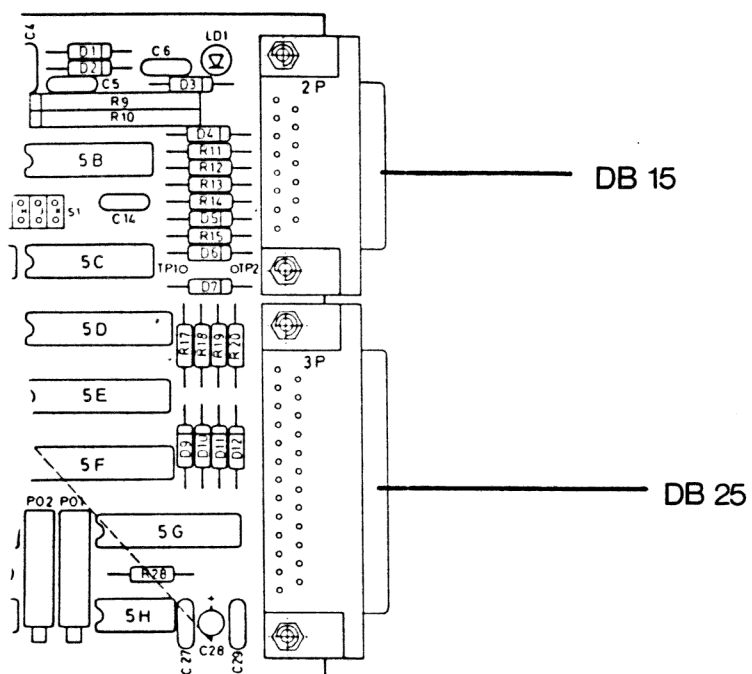




NOTE Lift the connected leg on R22 from the PC-board
Shrinkable tubing length 12 mm on R22

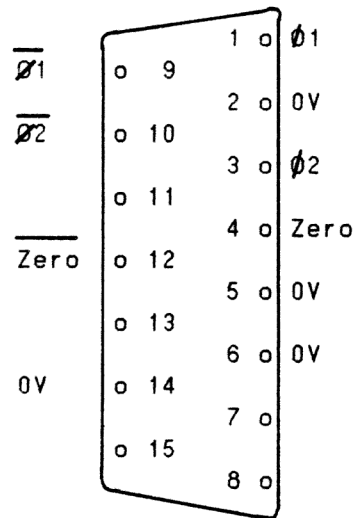


Kontaktplacering 2P - 3P



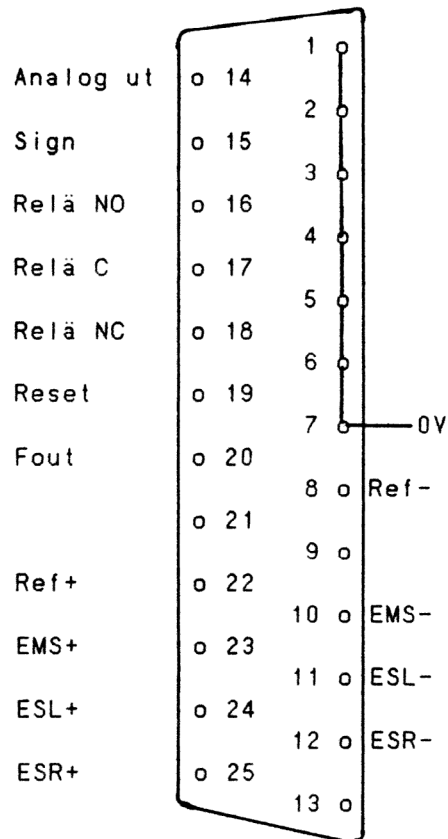
Kontakt 2P

Pulsgivare



Kontakt 3P

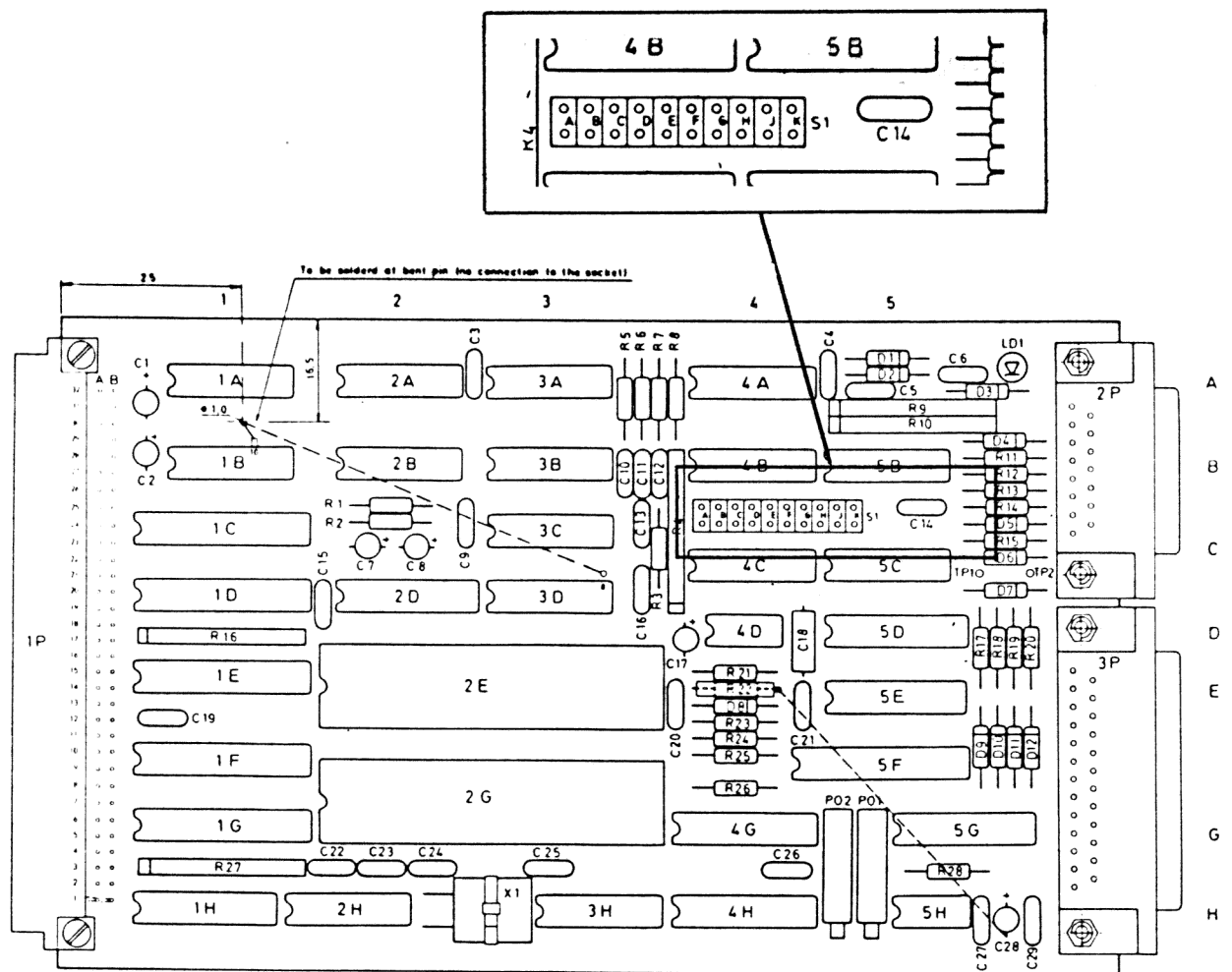
Digitala givarsignaler, frekvens och analog ut



Strappningsfält

Vid val av arbetsmetod 1, använd strapparna J och K i faldningsfältet. Vid val av arbetsmetod 2 använd strapparna A-H, för vidare information se sid 5.

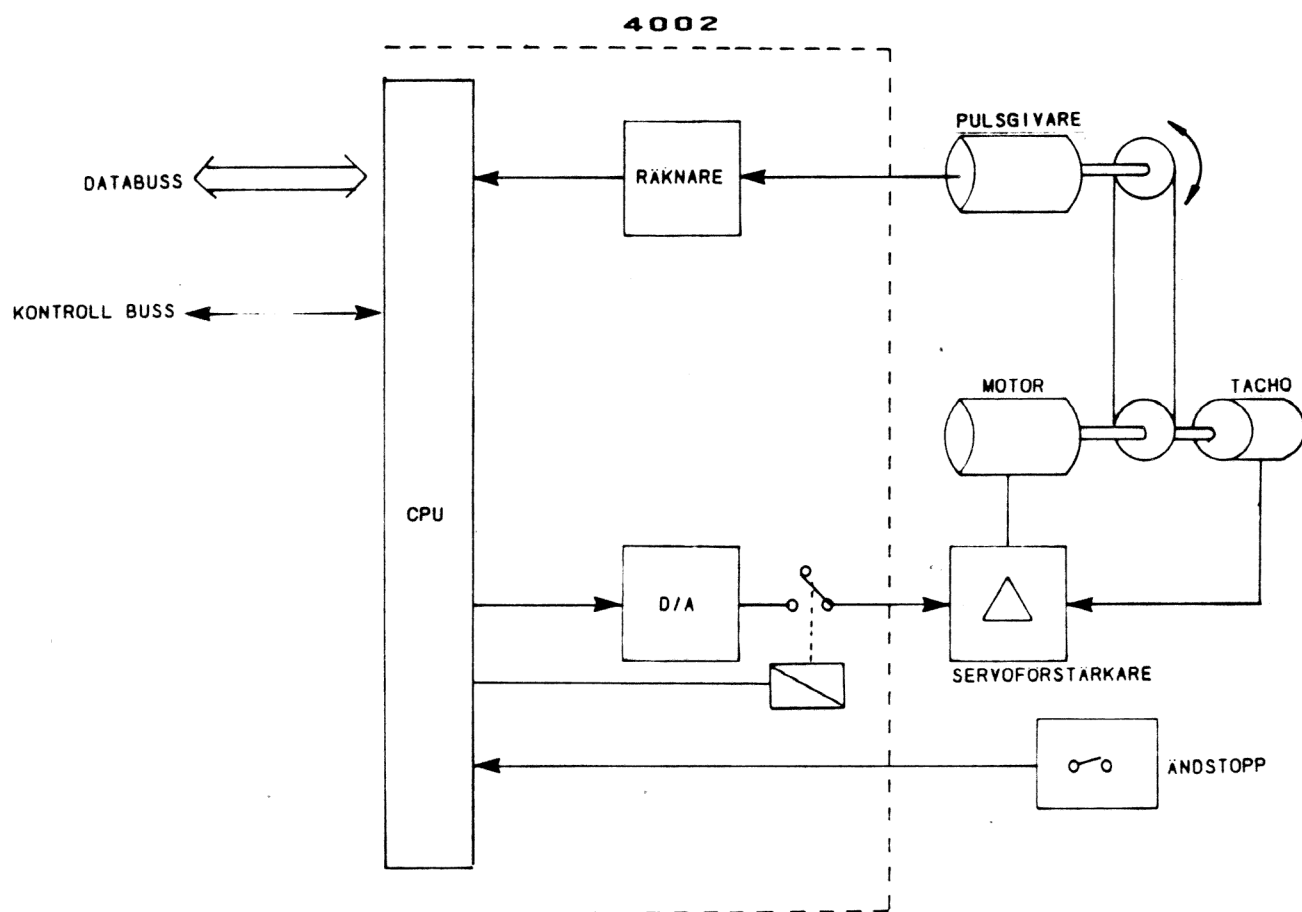
Faldningsfält



APPENDIX 1

Exempel på inkoppling av servosystem till 4002

För att åskådliggöra ett praktiskt arbetande servo har en uppkoppling gjorts.



Givaren är av fabrikat Haidenhaim, typ RoD 420.
 Servoförstärkaren är av fabrikat AXODYN typ O5LV.
 Motorn är av fabrikat BBC typ O1 LB4 3Volt taco.

Kontakten, DB15, till inkrementalgivaren följer ett standardiserat gränssnitt. Kontakten, DB25, innehåller hastighetsinformation till servot samt information från gränslägesbrytarna.

Signalen FOUT, den digitala hastighetsinformationen till servot, finns utdragen i en BNC-kontakt.

Polariteten på riktningsinformationen indikeras med hjälp av en lysdiod, kopplad till signalen SIGN.

Via en omkopplare väljs om servot skall ta sin analoga signal via det inbyggda kortslutningsreläet på 4002-kortet eller om analogsignalen skall gå direkt.

För att simulera brytarna REF, EMS, ESL och ESR vinns 4 st strömbrytare som via lysdioder ger information till 4002-kortet.

I uppkopplingen har ett analogt visarinstrument kopplas in för att visa den analoga spänningen till servot.

Vid inkoppling av systemet måste inkopplingen och val av jordning ske enligt tidigare diskussion, se avsnittet om analogutgången.

Vid inkoppling av ett servosystem kan många felbilder uppkomma. Ofta kan varje ingående enhet fungera perfekt var för sig men vid ihopkopplingen av systemet kan det uppstå problem.

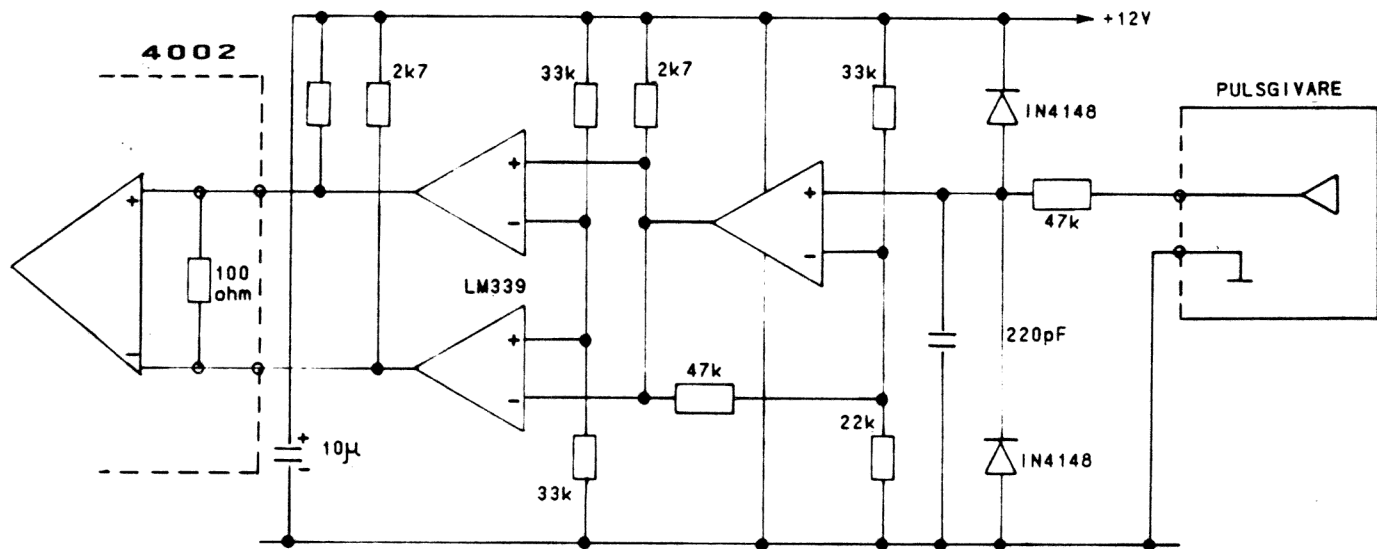
En av frågeställningarna är om servo-loopen skall vara öppen eller sluten. 4002-kortet är i grunden utfört som ett öppet servosystem. Med detta menas att när positionen är uppnådd ges stopp som är lika med ingen utspänning till servot.

Om systemet kopplas som en sluten servo-loop kommer minsta fel i positioneringen att resultera i en hastighetsangivelse till servot. Regulatorn som övervakar detta slutna system kan vara av mer eller mindre komplex art, PID-regulatorer, hastighetsservo etc. Alla dessa regulatorer arbetar med ett litet fel från lägesgivarna. Beroende på typ av regulatorer och beroende på den mekaniska massan i systemet kan utrustningen börja självsvänga, eller så kan onödigt stora belastningar uppstå i det mekaniska systemet.

APPENDIX 2

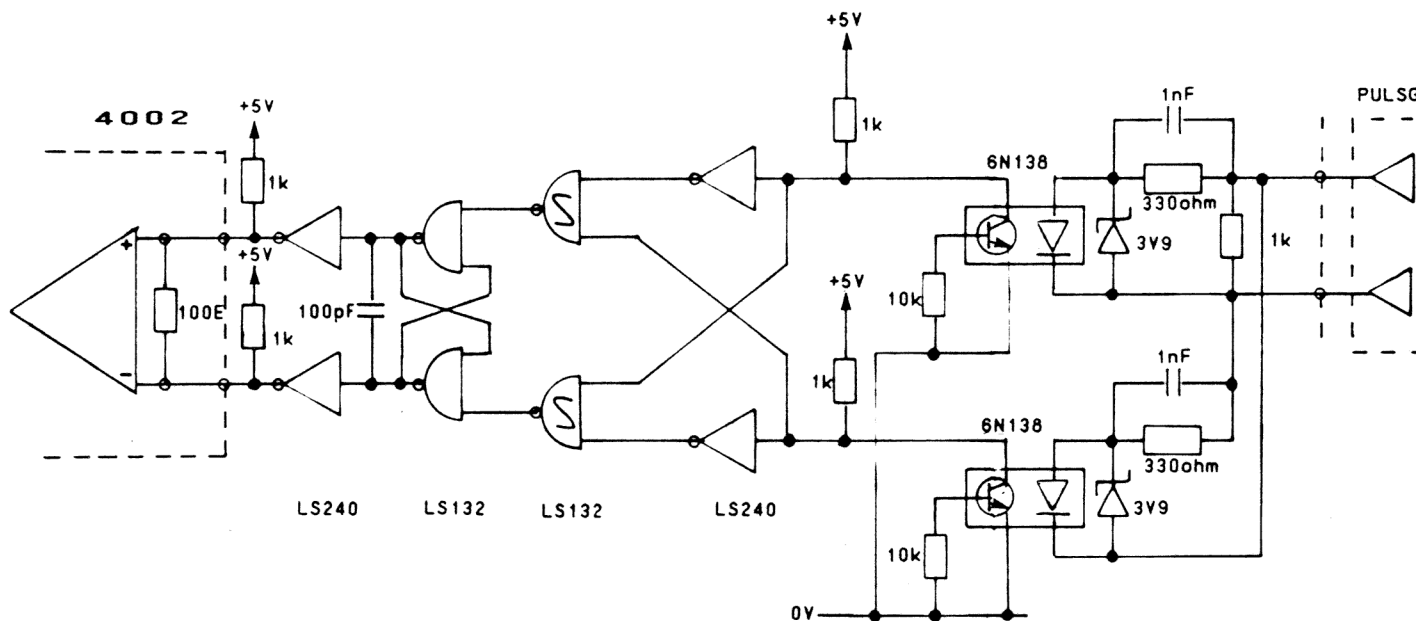
Inkoppling av diverse givare

Inkoppling av givare som inte har differentiell drivning.

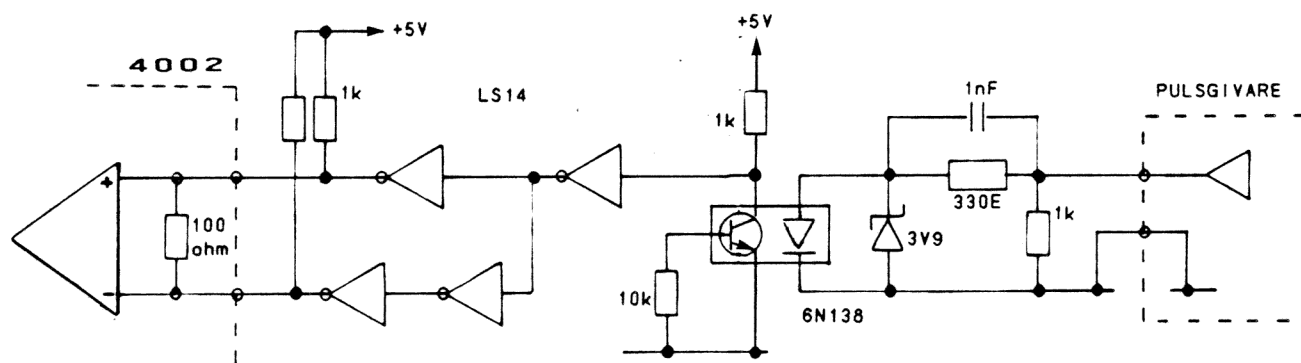


Ingången är inte matchad till linjeimpedansen. Den övre praktiska gränsfrekvensen är 50 kHz.

Exempel på inkoppling av optokopplare i de fall det är troligt att jordströmmar kan uppstå eller störande transienter kan uppträda.



Givare med differentiell matning. Gränsfrekvensen är över 100 kHz. Om givaren har högre spänning än 5V krävs seriemotstånd.



Givare utan differentiell matning. Den övre gränsfrekvensen ligger vid 50 kHz. Om spänningen är högre än 5V krävs seriemotstånd.