

# SMC30B / SMC30C

## Step Motor Controller

### Bruger Manual



**JVL Industri Elektronik A/S - Marts 1991**

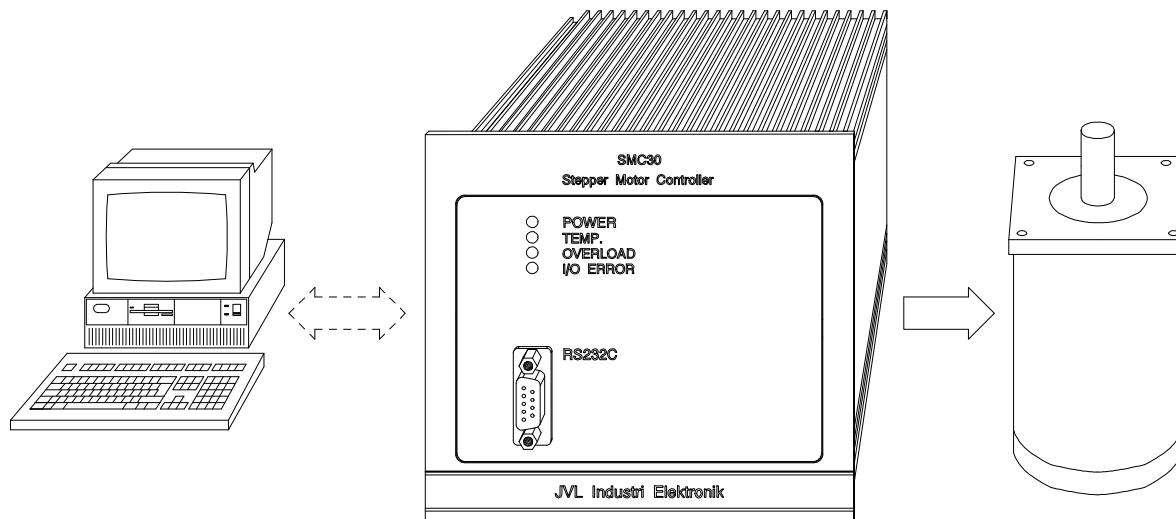
# Indhold

1.1	Introduktion	3
1.2	Tilslutningsmuligheder	4
<b>Driversektion</b>		
2.0	Blokdiagram	6
2.1	Strømforsyning	7
2.2	Motordriver	9
2.3	Brugerudgange	12
2.4	Steppuls-udgang	13
2.4	Brugerindgange	14
2.5	Analogindgange	16
2.6	Endestopindgange	17
2.7	Stopindgang	18
2.8	Stikforbindelser	19
<b>Controllersektion</b>		
3.1	Interfaceforbindelser	20
3.2	Adressering	21
3.3	Kommunikationshastighed	22
3.4	Kommandoformat	23
3.5	Checksum	24
3.6	Modulinterface	25
<b>Software beskrivelse</b>		
4.1	Generelt om software	27
4.2	Kommandooversigt	30
4.3	Systemkommandoer	33
4.4	Motorkommandoer	36
4.5	Brugerinterface	46
4.6	Flowkommandoer	49
<b>Appendix</b>		
5.1	Elektriske data	65
5.2	Fysiske mål	67
5.3	Pladsudnyttelse i hukommelse	68
5.4	Connectorboard for controller	69
5.5	Anvendelseseksempel	70
5.6	Motorforbindelser	76
5.7	Stikordsregister	77

## 1.1

# Stepmotor Controller SMC30

---



Stepmotor Controller type SMC30 er en let-håndterlig controller til styring af stepmotorer.

Den kan bruges som stationær enhed eller kobles sammen med en terminal eller personal computer via RS232C/V24 interface.

SMC30 er forsynet med ind- og udgange, som giver brugeren mulighed for en meget flexibel tilpasning til opgaver såsom styring af fræse/bore anlæg, håndteringsautomater og lignende, hvor hurtige og præcise manøvrer er ønskelige, uden at løsningen må fylde for meget eller være for kostbar.

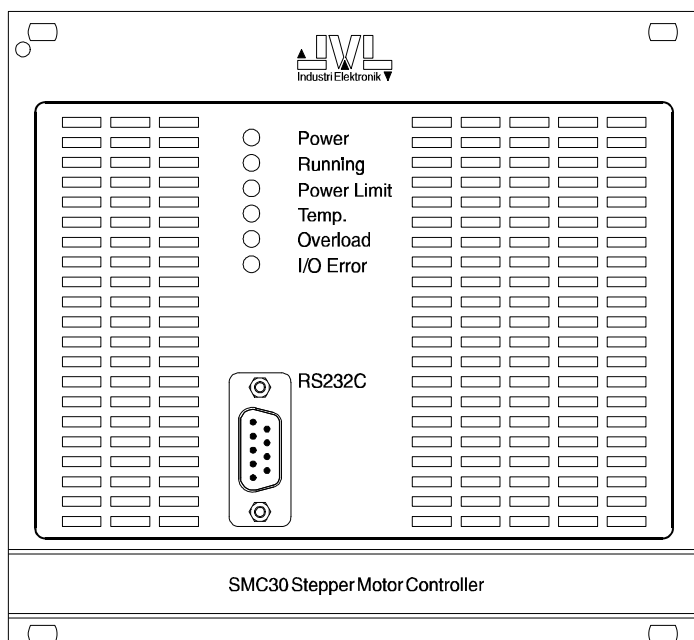
SMC30 kan leveres i 2 varianter, henholdsvis SMC30B og SMC30C der begge er baseret på "Bipolar chopper" princippet, og opererer med driverspændinger på 150V. Motorstrømmen kan ved SMC30B justeres fra 0 til 6 Amp. pr. fase og ved SMC30C kan strømmen justeres mellem 0 og 12Amp. pr. fase. Motordriveren i begge varianter er kortslutningssikret.

Brugerudgangene er beskyttet mod kortslutning og induktive tilbageslag.

Controlleren kan monteres i 19" rack eller fastspændes direkte på en flade.

### Features:

- Netforsyning 230V AC.
- Motormomenter op til 12Nm.
- Chopper-driver 150V med trinløs indstilling af motorstrøm (0-6 / 0-12A/fase).
- Termisk beskyttet.
- EMI-skærmet.
- RS232C/V24 kommunikation.
- Baud-rate 110 - 9600.
- Tilkobling af op til 7 controllere på samme interface linie.
- Enkel programmering.
- Kørefrekvens op til 15kHz.
- 3 Brugerindgange.
- 6 Analog/digital-indgange.
- 1 stopindgang.
- 3 Brugerudgange (hver 500mA).



Forside:

- "Power" indikator
- Indikation af kørsel/stilstand
- Indikation af effektforbrug
- Indikation af overophedning
- Indikation af korsluttet motorudgang
- Indikation af kortsluttet brugerudgang
- RS232C Interface

#### Motorudgang :

Der er mulighed for tilslutning af en 2-faset eller 4-faset stepmotor. Udgangen er kortslutningssikret. Motoren kan styres med indtil 15.000 halv/helstep i sekundet.

#### Strømforsyning :

SMC30 indeholder en komplet strømforsyning for nettilslutning (230VAC).

#### Brugerindgange :

Der er 4 støjbeskyttede indgange, hvoraf den ene er forbeholdt til drift-stop. De 3 andre kan bruges til f.eks. induktive følere eller til synkronisering af andre styringer.

Indgangene kan anvendes i området fra 5-30V.

Indgangene er alle galvanisk isolerede fra det øvrige kredsløb.

#### Interface :

RS 232C Interfacet gør det muligt at forbinde SMC30 til en computer eller terminal. Op til 7 controllere kan tilsluttes på samme interfacelinie.

#### Modulinterface :

Modulinterfacet udgøres af 2 terminaler der er galvanisk isoleret.

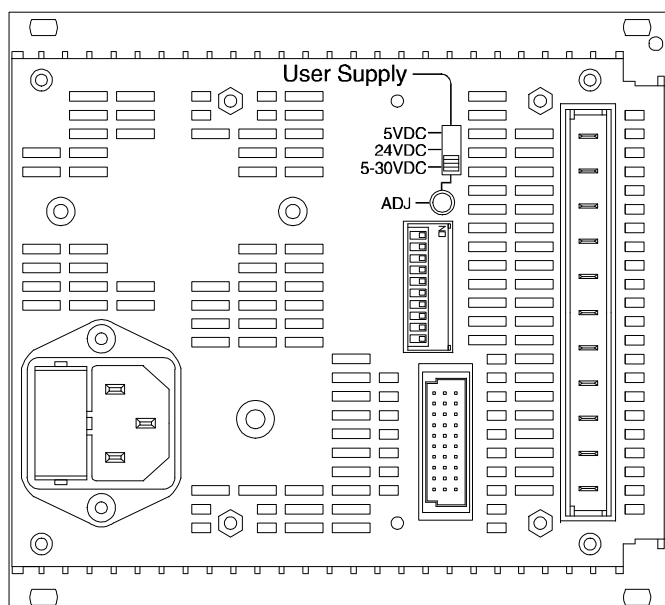
Disse 2 terminaler forbindes til alle eksterne moduler eksempelvis keyboard-/display-modul, ind-/udgangsmodul m.v.

#### Endestopindgange :

Endestopindgangene benyttes til formål hvor det er af yderste vigtighed at motoren ikke kører ud over nogle definerede grænser. Ved aktivering af endestop stoppes motorkørslen øjeblikkeligt. Endestopindgangene er sammen med brugerindgangene galvanisk isolerede fra det øvrige kredsløb.

## 1.2

## Tilslutningsmuligheder



### Bagside:

- 230V Tilslutning
- Motortilslutning
- Modulinterface
- 3 Brugerudgange
- 4 Brugerindgange
- Endestop-indgange
- 6 Analog-/digital-indgange
- RS232C Interface (sekundær)
- Valg af kommunikations-parametre m.v.
- Brugerforsyning 5-30VDC/500mA

### Brugerudgange :

Der er 3 udgange som f.eks. kan bruges til, at styre mindre DC motorer eller til at synkronisere enheden med andre styringer.

Hver brugerudgang kan levere op til 500mA og kan arbejde i spændingsområdet 5-30V.

Alle udgange er galvanisk isolerede fra det øvrige kredsløb. Endvidere er udgangene kortslutningsbeskyttede.

### Brugerforsyning :

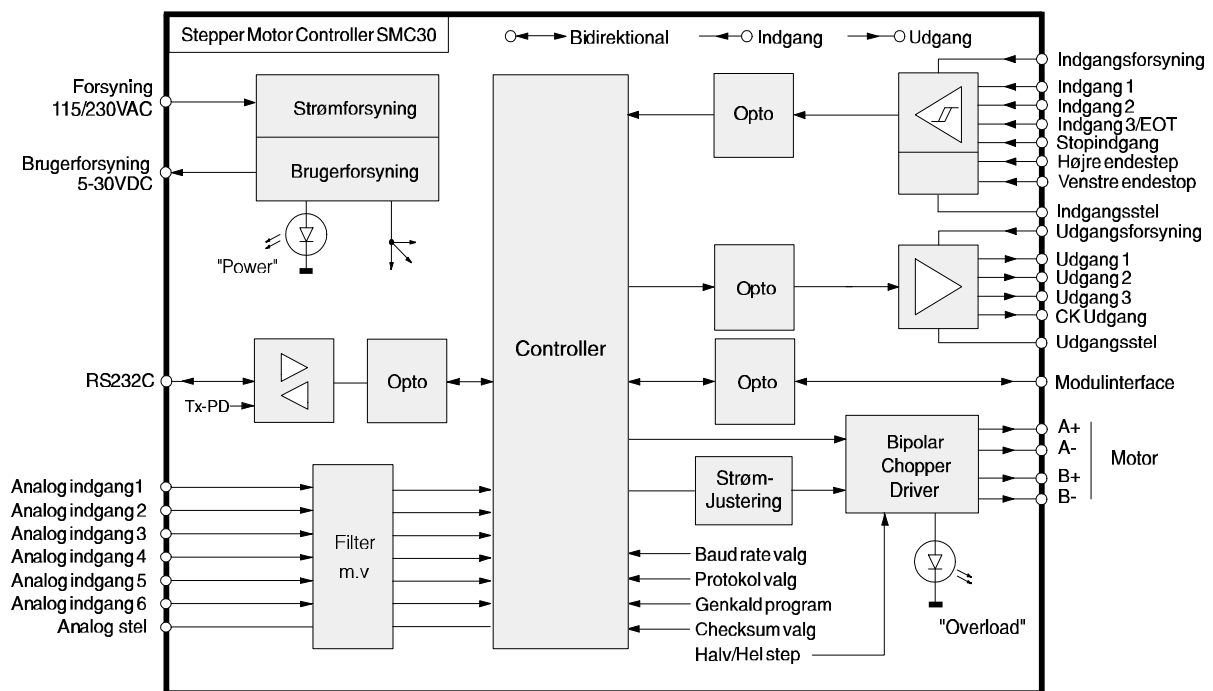
Til forsyning af følere, magnetventiler m.v. er der på bagsiden af controlleren placeret en brugerforsyningsudgang. Spændingen kan indstilles til henholdsvis 5VDC, 24VDC, eller justeres trinløst fra 5 til 30VDC.

Brugerforsyningen kan levere en strøm på 0,5A uanset spænding.

### Analogindgange :

De 6 analogindgange kan aftastes via et sæt af kommandoer. Det er f.eks. muligt, at styre motorens tophastighed ved hjælp af en spænding påtrykt en af de 6 analogindgange.

Indgangene skal tilføres en spænding i området 0 - 5,10V, og er beskyttet imod kortvarige overspændinger op til 45V.

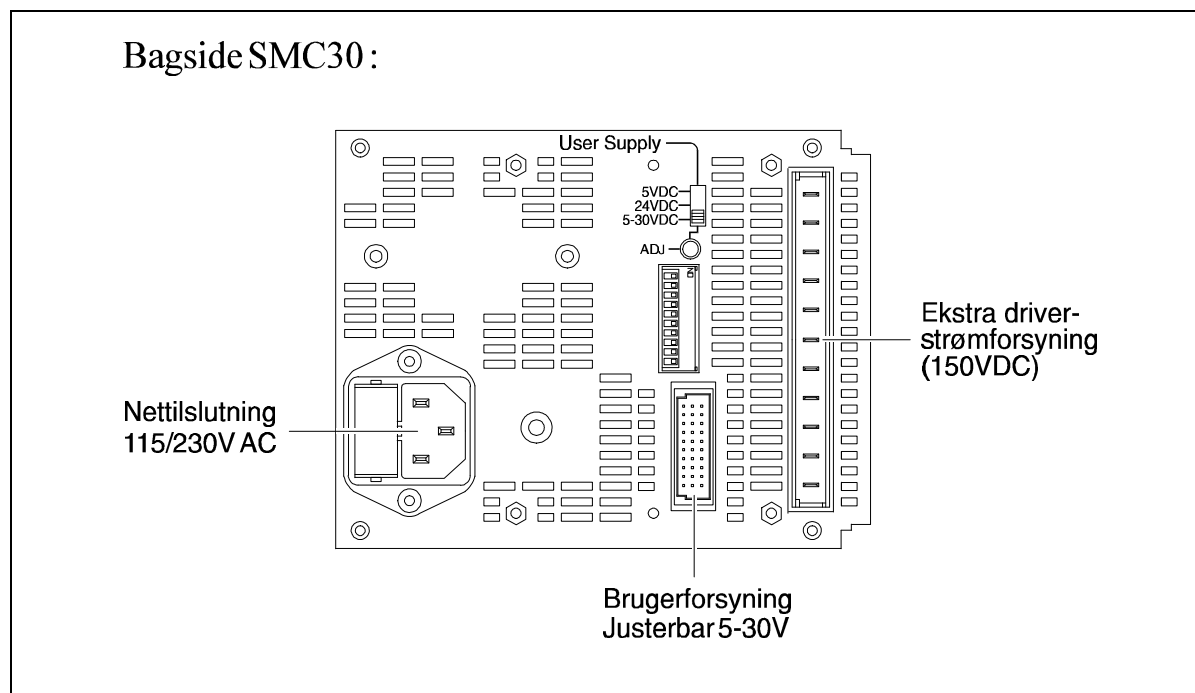


Ovenstående blokdigram viser de enkelte sektioner i Controlleren.

Dette kapitel omhandler de enkelte blokke og funktioner på hardware niveau.

Ønskes softwarefunktioner uddybet henvises til afsnit 4 hvor det komplette kommandosæt til f.eks., aktivering af brugerudgangene, er beskrevet.

Bemærk på blokdagrammet at alle ind- og udgange er optoisoleret fra hinanden, dog er analogindgangene ikke isoleret fra mikroprocessordelen.



### Generelt

Controlleren indeholder en komplet strømforsyning for netdrift.

Strømforsyningen er udvidet med en brugersforsyning der kan benyttes til eksterne formål. Denne brugersforsyning kan indstilles i 3 positioner (se ovenstående tegning).

Hvis omskifteren stilles i 1. position afgiver brugersforsyningen en fast spænding på 5VDC. Hvis omskifteren stilles i 2. position afgiver brugersforsyningen en fast spænding på 24VDC. Brugersforsyningen er i 3. position justerbar i intervallet 5 til 30VDC.

Brugersforsyningen kan i alle positioner levere en kontinuerlig strøm på 0,5A. Hvis denne strøm overskrides vil strømforsyningen skrue ned for spændingen på en sådan måde at brugersforsyningen aldrig lider overlast.

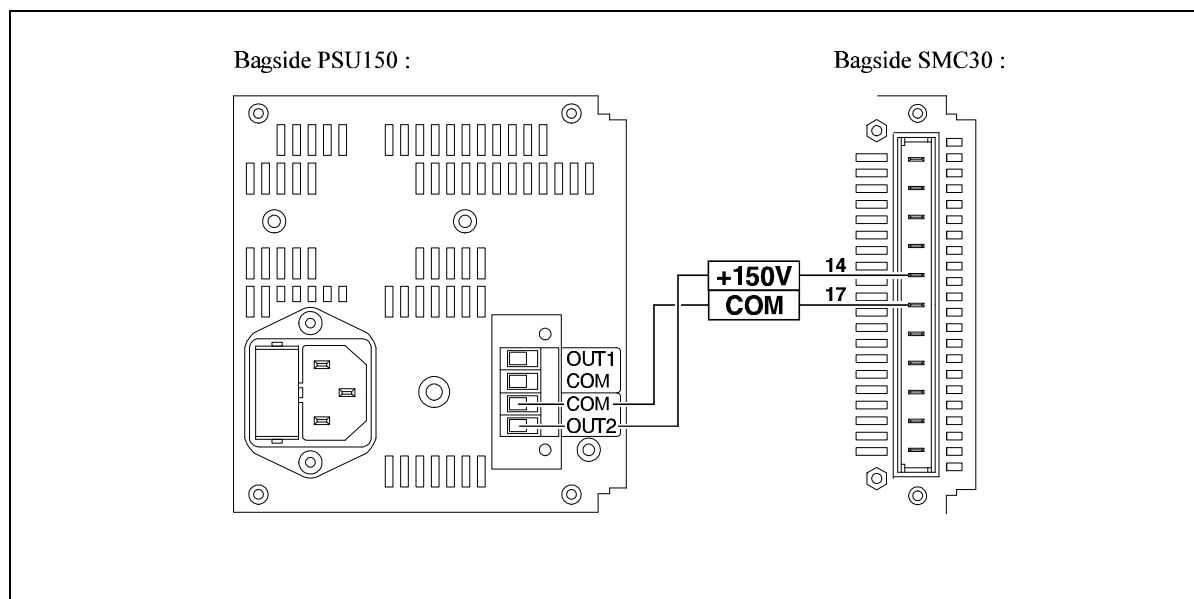
Brugersforsyningen tåler dermed kontinuerlig kortslutning.

Brugersforsyning samt controllerens interne spændingsforsyning (150VDC) er tilgængelig eksternt via de 2 stik.

Den interne forsyning kan benyttes til andre controllerer i det aktuelle system. Den røde lysdiode benævnt *Power Limit* på controllerens forplade vil lyse hvis effektforbruget fra controllerens strømforsyning overstiger de 200W som er controllerens maksimale ydeevne. Hvis der er behov for mere end 200W kan der tilkobles en ekstern strømforsyning.

I tilfælde af overspænding på nettet, vil controllerens overspændingskredsløb brænde sekundær eller primær sikring af. Er dette tilfældet, skal der afbrydes for forsyningen og sikringen udskiftes, hvorefter controlleren er funktionsdygtig igen.

Primær og sekundær sikring er placeret i netbrøndens lille skuffe.



### Ekstern Strømforsyning.

Controlleren indeholder en komplet strømforsyning for netdrift. Denne strømforsyning kan levere en permanent effekt på ca. 200W og en spidsbelastning på 300W. Er denne grænse nået vil lysdioden benævnt *Power Limit* på controllerens forplade lyse. Er dette tilfældet kan en ekstern strømforsyning tilsluttes - se ovenstående illustration. Dog skal man være opmærksom på at controlleren selv skruer ned for motorstrømmen hvis effektforbruget overstiger 200W, og derfor ikke lider overlast.

Størrelsen på den eksterne strømforsyning afgøres af de aktuelle krav til motorhastighed og moment.

JVL Industri Elektronik kan tilbyde følgende strømforsyninger:

PSU150-2 : 150VDC/200W  
 PSU150-4 : 150VDC/400W.

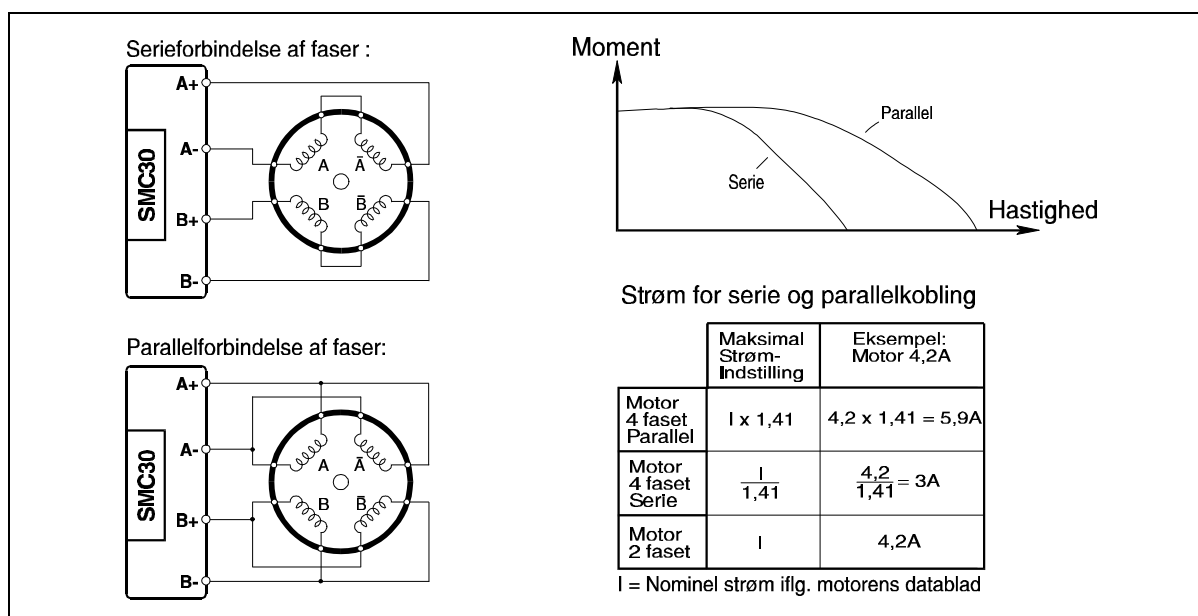
Hvis 150V indgangen bliver udsat for højere spændinger end ca. 180V vil et internt beskyttelseskredsløb afbryde for den udefra kommende spænding, via en sikring der er placeret internt i controlleren. Denne feature er kun tænkt som beskyttelse mod beskadigelse af selve controlleren, og er den interne sikring brændt af, skal controlleren til service.

I tilfælde af overspænding på nettet, vil controllerens overspændingskredsløb brænde sekundær eller primær sikring af. Er dette tilfældet, skal der afbrydes for forsyningen og sikringen udskiftes, hvorefter controlleren er funktionsdygtig igen. Disse sikringer kan udskiftes af brugeren selv, og er placeret i netbrøndens lille skuffe.



## 2.2

## Tilslutning af motor



### Motortyper.

Stepmotorer kan leveres i flere varianter:

1. 2 Faser Bipolar (4 ledninger)
2. 4 Faser Bipolar/Unipolar (8 ledninger)
3. 4 Faser Unipolar (6 ledninger)-(ikke egnet)

Bemærk at pkt. 3 (Unipolar motor) ikke er egnet til drift med disse typer af controllere, idet de kører efter bipolar princippet. Bemærk også at et bipolar-system typisk giver 40% mere moment end et unipolar-system. 2 eller 4 fasede motorer kan forbindes på følgende måder:

### 2 Faset motor (4 ledninger).

Denne motortype kan forbindes direkte til controllerens udgangsterminaler.

Strømmen må maksimalt indstilles til den strøm der står anført i motorens datablad.

### 4 Faset motor (8 ledninger).

Denne motortype kan forbindes på 2 måder:

1. Faserne i serie.
2. Faserne i parallel.

Valget mellem disse 2 forbindelses-måder afgøres typisk ud fra kravet om systemets hastighed. Hvis det ønskes at køre langsomt, hvilket typisk er under 1 kHz, kan det vælges at forbinde faserne i serie. Hvis det ønskes at køre hurtigt, hvilket er over 1 kHz, kan det vælges at forbinde faserne i parallel.

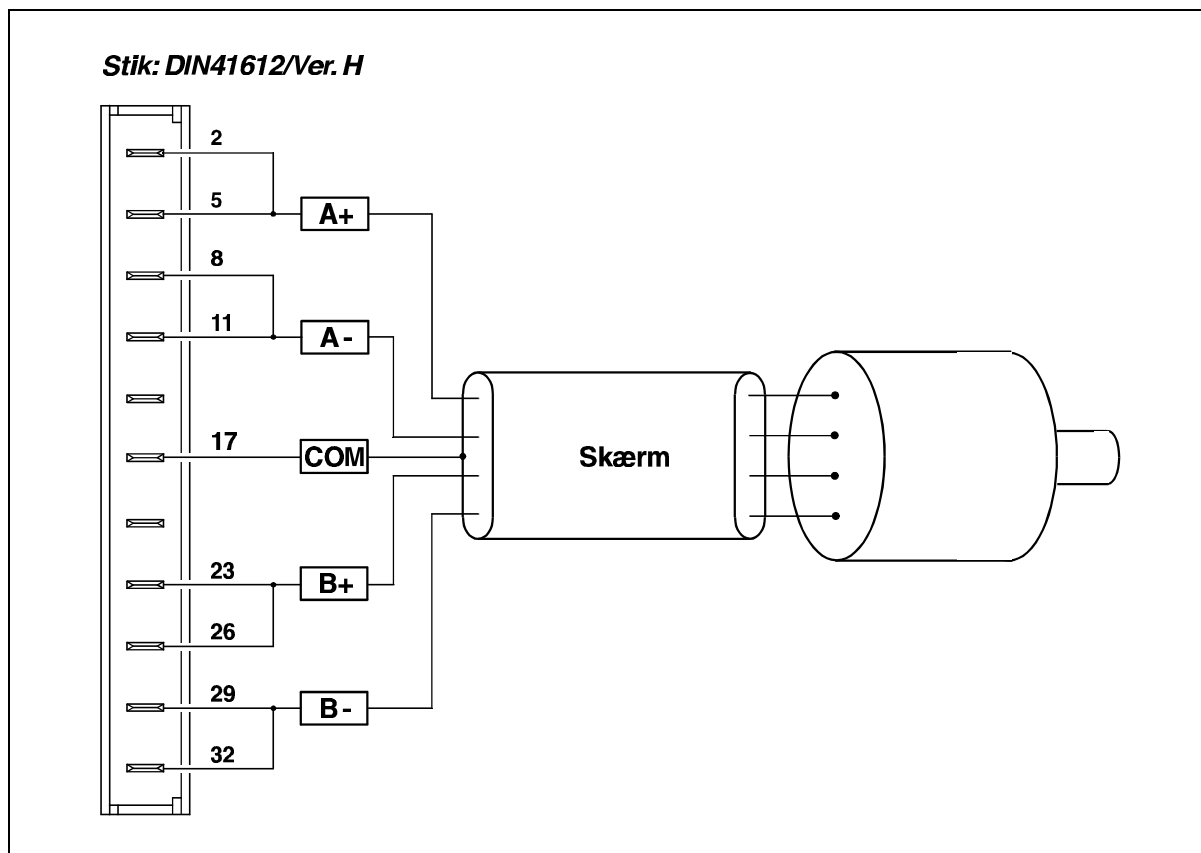
### Seriekobling:

Ved serieforbindelsen opnår man at motoren kan yde det samme moment (op til 1kHz) som ved parallel-forbindelsen dog med ca. den halve strøm. Dette kan have indflydelse på valget af controller idet det derved kun er nødvendigt at vælge en controller der kan levere den halve strøm. Se ovenstående illustration.

Når faserne på en 4 faset stepmotor kobles i serie skal motorens nominelle fasestrøm divideres med 1.41. Hvis der f.eks. i databladet for en 4 faset motor er angivet en fasestrøm på 4,2 Amp må strømmen maksimalt stilles på 3 Amp. når faserne forbindes i serie.

### Parallelkobling.

Ved parallelforbindelsen opnår man at motoren kan yde væsentligt mere ved frekvenser over 1kHz i forhold til serieforbindelsen dog med ca. den dobbelte strøm. Dette kan have indflydelse på valget af controller idet det derved er nødvendigt at vælge en controller der kan levere den dobbelte strøm set i forhold til serieforbindelsen. Se ovenstående illustration. Når faserne på en 4 faset stepmotor kobles i parallel skal databladets nominelle fasestrøm multipliceres med faktoren 1.41. Hvis der f.eks. i databladet for en 4 faset motor er angivet en fasestrøm på 4,2 Amp må strømmen maksimalt stilles på 5,9 Amp. når faserne forbindes i parallel.



Det bør bemærkes, at jo lavere selvinduktion motoren har, desto bedre, idet dette har en del indflydelse på drejningsmomentet ved høje hastigheder. Drejningsmomentet er proportional med den strøm, der bliver tilført motoren.

Følgende forhold gør sig gældende:

$$\text{Strøm} \approx \text{Moment} \approx \frac{\text{Påtrykt spænding}}{\text{Faseinduktion} \times \text{Kørefrekvens}}$$

Den påtrykte spænding reguleres i driveren, så fasestrømmen afpasses til den indstillede strøm.

I praksis vil dette sige, at vælges en motor med en stor faseinduktion på f.eks. 100mH, kan driveren ikke levere den ønskede fasestrøm ved høje hastigheder (høj kørefrekvens), idet udgangsspændingen er begrænset.

**Kabelføring.**

På SMC30B der kan afgive strømme i intervallet 0 til 6 Amp. kan det anbefales, at bruge min. 0,75 mm<sup>2</sup> kabel.

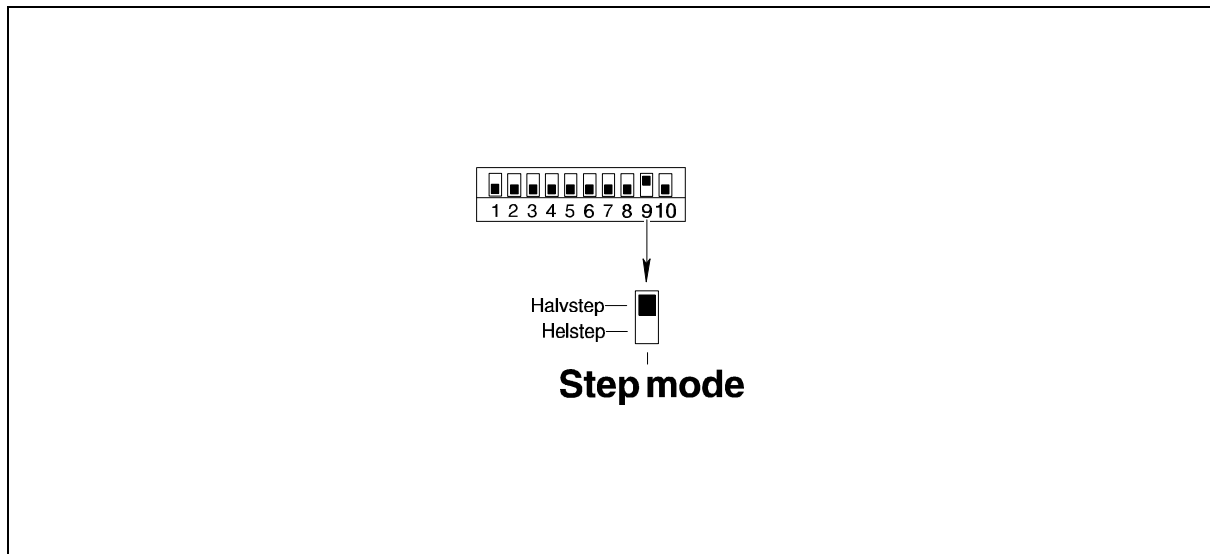
På SMC30C der kan afgive strømme i intervallet 0 til 12 Amp. kan det anbefales, at bruge min. 1,5mm<sup>2</sup> kabel.

Kabellængden bør ikke overstige 10 meter, eftersom kablets impedans vil give effekttab.

**Vigtigt !**

For at mindske støjstråling fra motorkablerne, skal der benyttes skærmet kabel.

Hvis der ikke benyttes skærmet kabel, kan anden elektronik i nærheden blive påvirket.



#### Stepopløsning.

Der kan vælges om controlleren skal styre motoren med halve eller hele step. Det kan ofte være en fordel at køre med halve step, eftersom man derved opnår den dobbelte opløsning pr. motoromdrejning. Dette kan muligvis overflødig gøre brugen af en mekanisk udveksling.

En anden fordel er, at man normalt undgår de resonansfænomener der kan forekomme når motoren køres i helstep, idet en stepmotor altid har en frekvens, hvor der opstår resonans. Denne frekvens kan ændre sig afhængigt af belastningen og vil give sig udslag i, at motoren taber al sin kraft. Ved større motorer, vil resonansfrekvensen som regel ligge uden for arbejdsområdet (det frekvensområde motoren accelereres og decelereres i).

Ovenstående tegning viser, hvorledes halv/hel step sættes:

#### Sikring mod overlast.

Controllerens driverkredsløb er kortslutningssikret, hvilket betyder, at motorudgangene bliver spændingsløse, når spidsstrømmen overstiger controllerens maksimale strøm plus 20% i mere end 2ms. En øjeblikkelig kortslutning mellem to vilkårlige motorterminaler vil controlleren ikke tage skade af, men blot indikere overbelastning via lysdioden benævnt *Overload*. For at få controlleren funktionsdygtig igen, skal forsyningsspændingen til controlleren blot fjernes i mindst 5 sek.

**Bemærk!** Controllerens motorudgange er ikke sikret imod kortslutning til stel (P-).

## 2.3

## Brugerudgange

For at gøre controlleren så let-anvendelig som muligt, rummer den 3 indgange og 3 udgange, som alle står frit til rådighed for brugeren.

En 4. indgang kan benyttes til drifts-stop.

Alle indgange er galvanisk isolerede.

Bruger ind- og udgangene skal alle strømforsynes eksternt.

### Brugerudgange.

Med mulighed for at kunne styre bifunktioner såsom aktuatorer, mindre motorer osv., kan brugerudgangene anvendes.

Dette giver mulighed for at synkronisere stepmotorens bevægelser med omverdenen.

De 3 udgange styres softwaremæssigt og til-lader alle en strøm op til 500mA.

De er beskyttet mod induktive tilbageslag og er kortslutningssikrede.

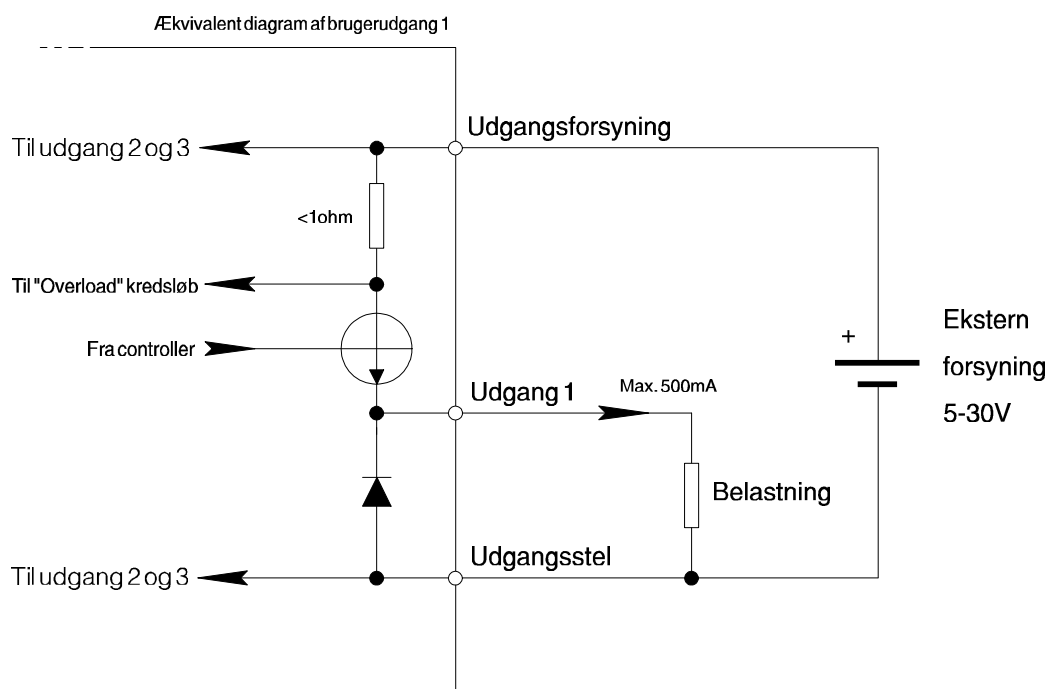
Hvis en given brugerudgang kortsluttes, eller hvis der trækkes en strøm større end 700mA, vil "Overload" dioden blinke. Yderligere vil den overbelastede brugerudgang blive spændingsløs.

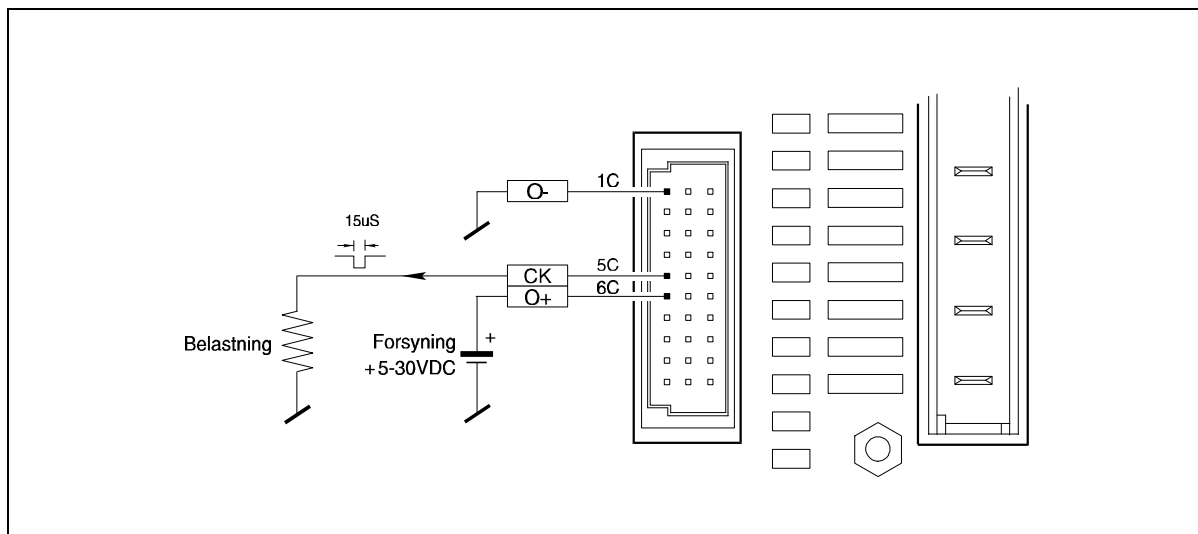
For at få controlleren funktionsdygtig igen, skal strømmen til controlleren blot fjernes i mindst 5 sek.

Hver af udgangene skal betragtes som en kontakt til + forsyningen. D.v.s. er en given udgang aktiveret, skal belastningen sættes mellem udgangsterminalen og stel (se diagram).

For at gøre udgangene kompatible med logik-kredsløb, skal der tilføjes en "pull down" modstand fra hver af udgangsterminalerne, til stel. Ved TTL skal denne modstand være 1kOhm, ved CMOS kan den sættes til ca. 10kOhm.

### Forbindelsesskema for brugerudgange





Til systemer hvor flere steppmotorer skal køre synkront er controlleren udstyret med en udgang hvor der udsendes en spændingsimpuls hver gang motoren tager 1 step.

Udgangen kan også benyttes for tilslutning af en ekstern tæller eller lignende som skal holde styr på hvor motoren befinder sig.

Udgangen har samme elektriske data som de 3 brugerudgange og tillader en belastningsstrøm op til 500mA. Udgangen deler ligeledes stel og forsyning med de 3 bruger-udgange.

Udgangen er beskyttet mod induktive tilbageslag og er kortslutningssikret.

Hvis steppulsudgangen kortsluttes, eller hvis der trækkes en strøm større end 700mA, vil "Overload" dioden blinke. Yderligere vil den overbelastede steppulsudgang blive spændingsløs.

For at få controlleren funktionsdygtig igen, skal strømmen til controlleren blot fjernes i mindst 5 sek.

Udgangen skal betragtes som en kontakt til + forsyningen. D.v.s. er den aktiveret, skal belastningen sættes mellem udgangsterminalen og stel (se diagram).

For at gøre udgangen kompatibel med logik-kredsløb, skal der tilføjes en "pull down" modstand fra udgangsterminalen, til stel. Ved TTL skal denne modstand være 1kOhm, ved CMOS kan den sættes til ca. 10kOhm.

## 2.5

# Brugerindgange

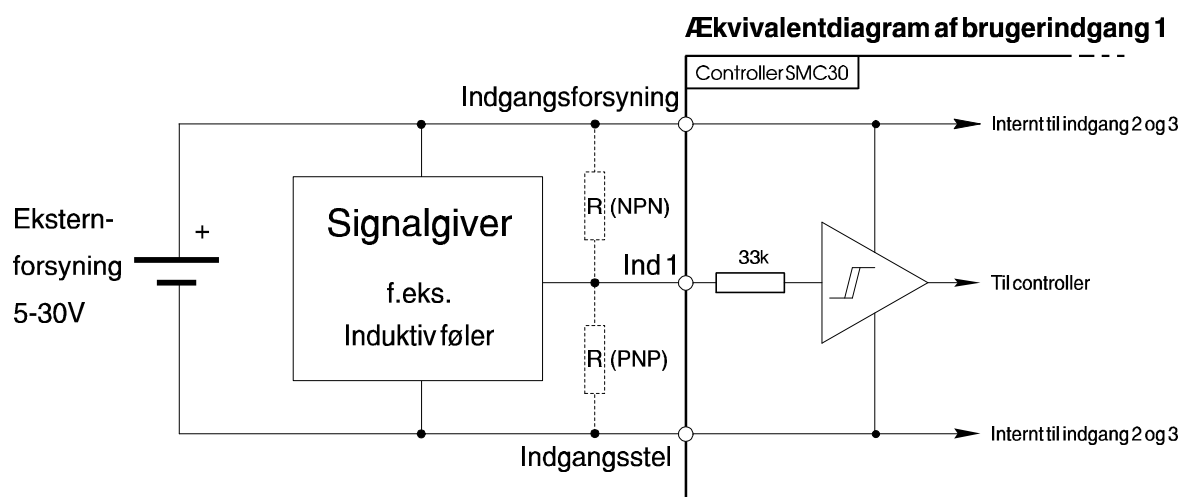
Der sidder et 1.orden lavpasfilter efter hver indgangsterminal, der afskærer frekvenser over 1kHz. Dette er gjort som sikkerhed, således at elektrisk støj fra startende motorer o.lign. ikke influerer på indgangssignalerne.

Det skal bemærkes, at de 3 brugerindgange vil indtage en udefineret stilling, hvis de ikke er forbundet til noget.

Alle indgangene er galvanisk isoleret fra det øvrige kredsløb. Dette medfører at indgangskredsløbet skal strømforsynes via en ekstern forsyning - se nedenstående tegning.

Visse induktive følere har åben kollektor udgang. Drejer det sig om følere med NPN udgang, skal der forbindes en ekstern modstand fra indgangen til +forsyningen og drejer det sig om PNP følere, skal en ekstern modstand forbindes fra indgangen til stel. Modstanden kan anbefales i størrelsen 500Ohm til 5kOhm, afhængigt af forsynings-spænding.

## Forbindelsesskema for brugerindgange



## 2.5

# Brugerindgange

### Indgangshysterese.

Samtlige indgange er støjbeskyttede og er kompatible med de gængse logiktyper, CMOS, TTL, osv.

Hysteresefunktionen i indgangene er afhængig af den tilsluttede forsyningspænding, hvilket fremgår af nedenstående kurve.

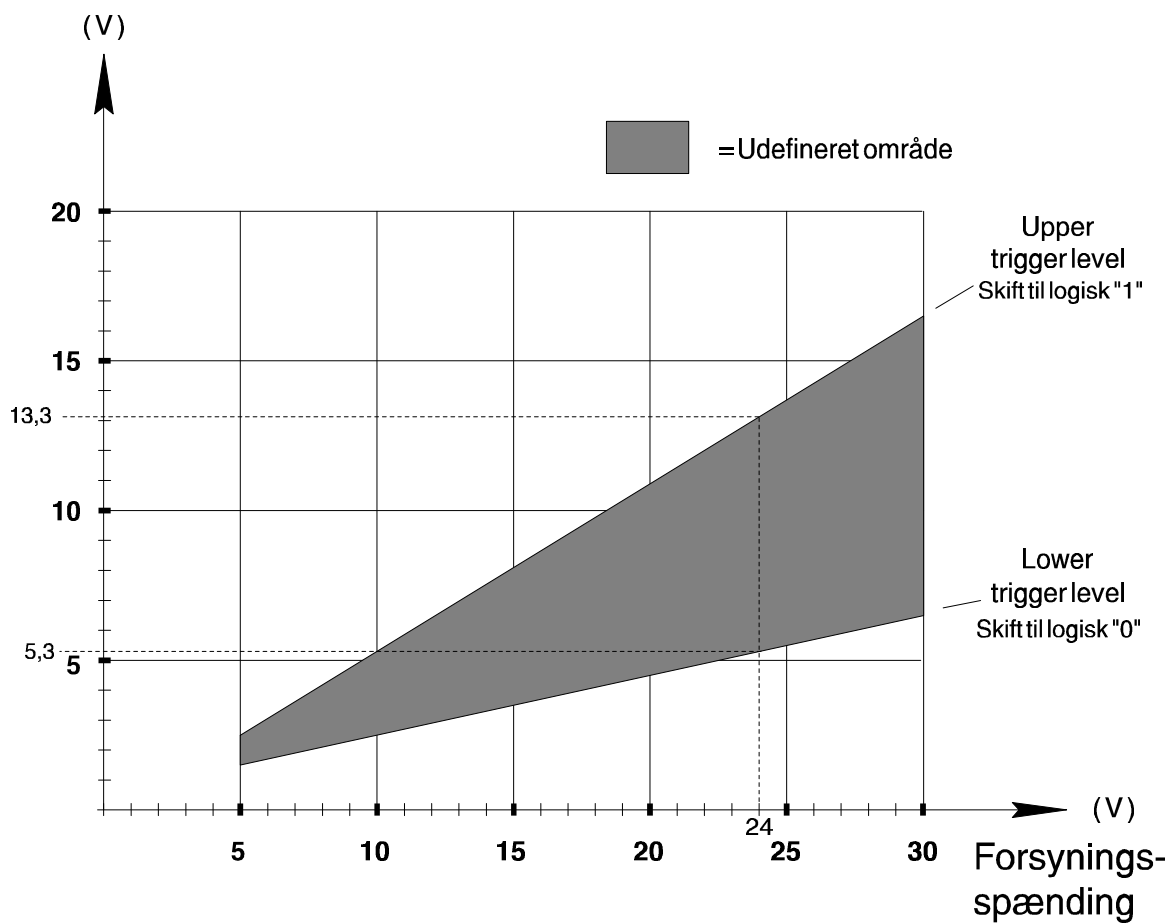
### Eksempel:

Indgang 1 ønskes anvendt, og forsyningspændingen er 24V.

Som det fremgår af nedenstående kurve, vil indgang 1 være logisk "1" ved at påtrykke en spænding større end 13,3 VDC. For at påtrykke indgangen logisk "0", skal indgangsspændingen sænkes til under ca. 5,3 VDC.

Tolerancen på disse spændinger er  $\pm 10\%$ .

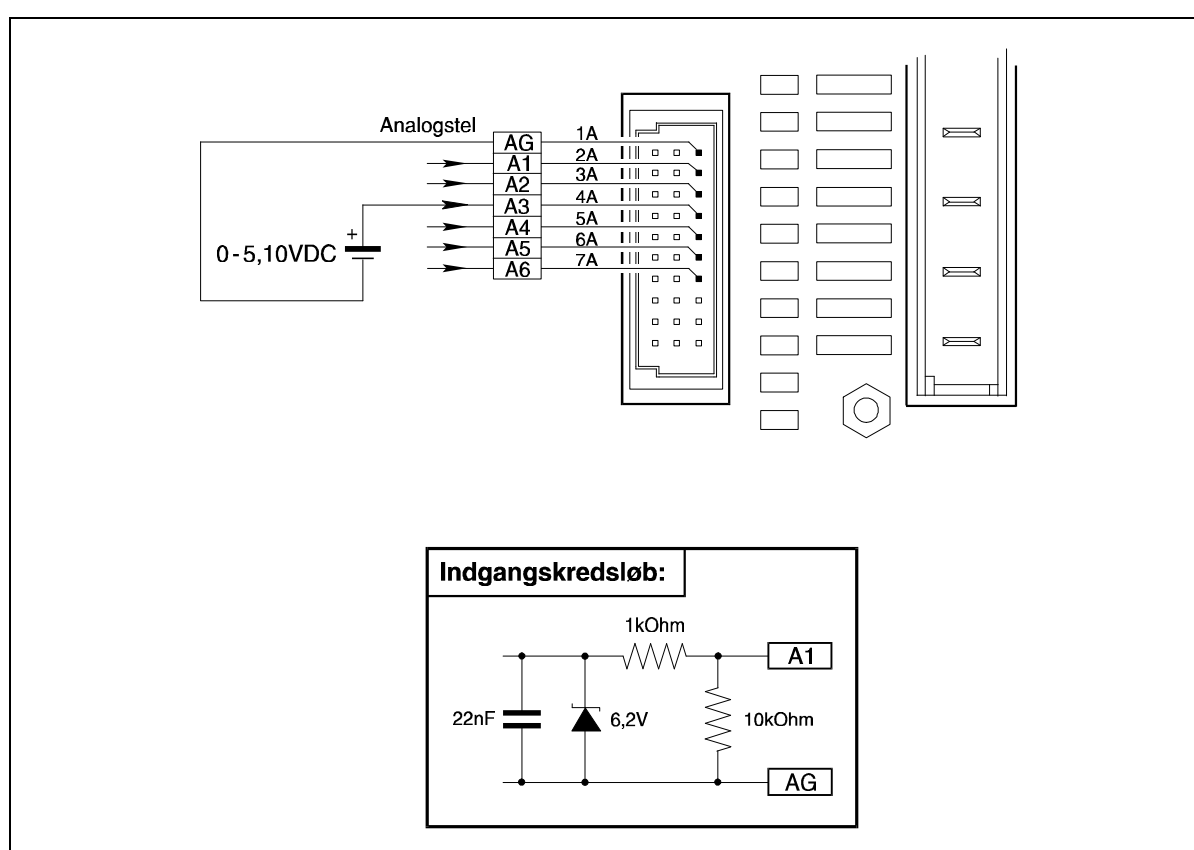
### Indgangsspænding



Se endvidere afsnit 4.5 (Brugerinterface) vedrørende bruger ind- og udgangene.

## 2.6

## Analogindgange



### Analogindgang

Controlleren rummer 6 analoge indgange, som kan aftastes via et sæt af kommandoer, der er beskrevet i afsnit 4.

De analoge indgange muliggør f.eks., at man kan styre stepmotorens hastighed ved hjælp af en analog spænding.

Indgangene er beskyttet imod kortvarige overspændinger op til 45V.

Hver gang controlleren foretager en måling på en given indgang, foretages i virkeligheden 16 målinger. Disse 16 målinger lægges sammen og midles. Dermed minimeres muligheden for, at en pludselige støjimpuls fra f.eks. stepmotordriveren, påvirker en måling.

Analogindgangene kan også benyttes som konventionelle brugerindgange (digitale niveauer), dog uden hysteres og galvanisk

isolation. Der skal ikke tages nogen forholdsregler for at udnytte dette, idet en given analogindgang på et vilkårligt tidspunkt kan benyttes enten som analogindgang eller som brugerindgang (digitale niveauer).

Se afsnit 4 - kommandoerne  $\pm A$ ,  $DA$ ,  $G\pm A$ ,  $JCA$ ,  $NA$ ,  $r$ ,  $s$ ,  $t$ ,  $U$ ,  $VA$ ,  $W$ .

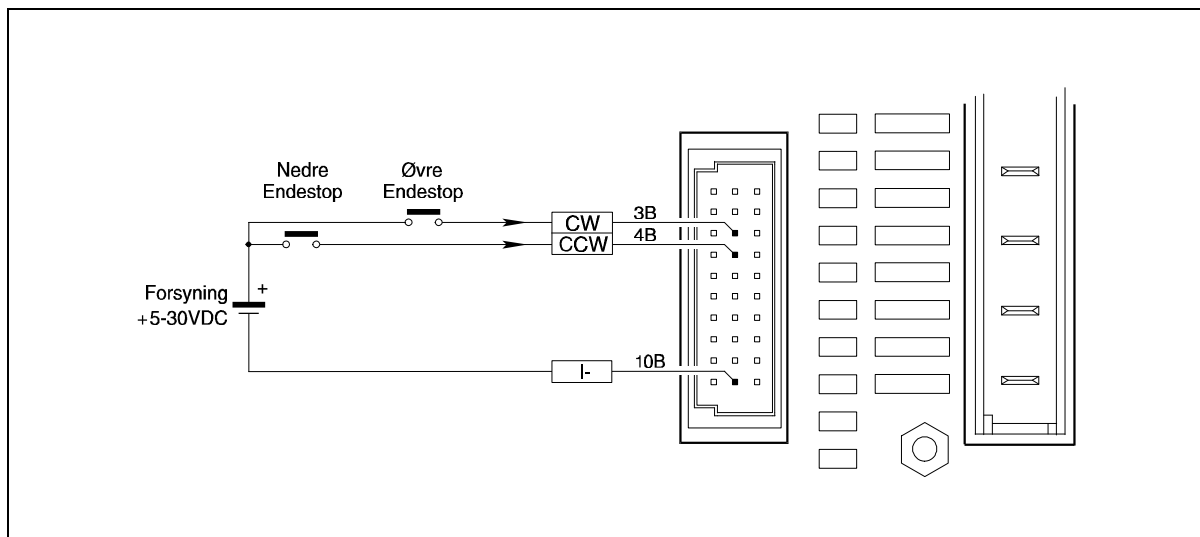
Indgangene accepterer spændinger fra 0V til 5,10V. Controllerens A/D konverter kører med 8bit, hvilket giver en opløsning på 256 trin. Hvert trin svarer dermed til 20,0mV på indgangen.

For at undgå fejlmålinger, skal stelbenet AGND (se stikskitse afsnit 2.7) benyttes sammen med de 6 analogindgange.

Se elektriske data afsnit 5.1 vedrørende yderligere informationer.

Efter hver analogindgangsterminal sidder et 1.orden lavpasfilter, der afskærer frekvenser over 10kHz.





### Endestopindgange

Ofte er det nødvendigt at opretholde nogle mekaniske yderpunkter i et steppermotorsystem, som steppermotoren under ingen omstændigheder må overskride.

Controlleren rummer til dette formål 2 indgange, benævnt *CCW* (nedre endestop) og *CW* (øvre endestop).

En af disse 2 indgange vil afhængigt af den aktuelle omdrejningsretning, stoppe motorkørslen hvis de aktiveres.

### Nedre endestop.

Hvis motoren kører tilbage og nedre endestopindgang aktiveres (logisk "1"), vil motorkørslen stoppe. Øvre endestop-indgang vil under tilbagekørsel ikke have nogen indflydelse.

### Øvre endestop.

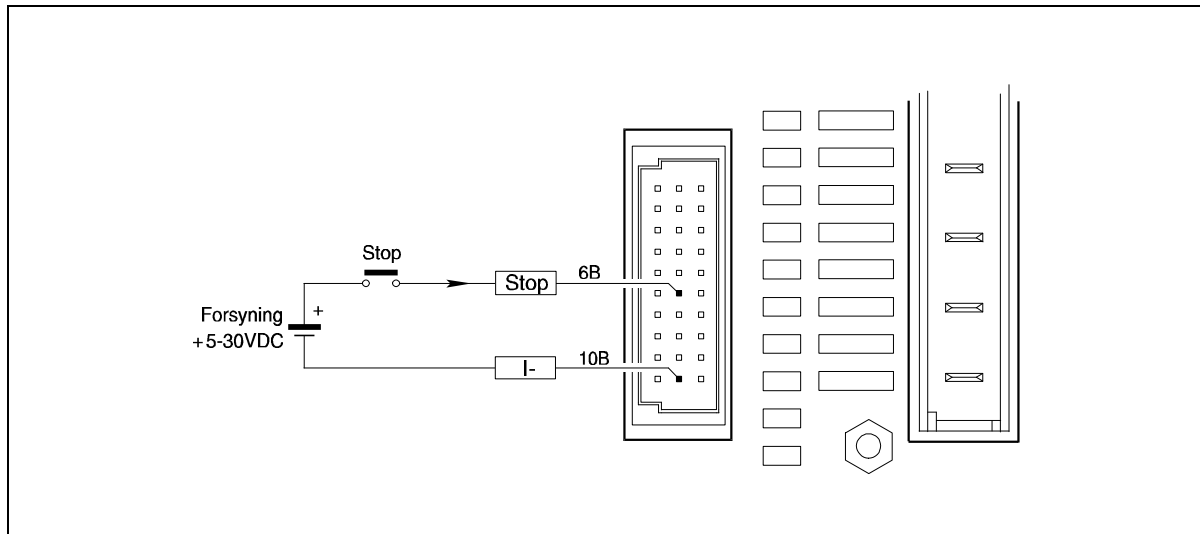
Hvis motoren kører fremad og øvre endestopindgang aktiveres (logisk "1"), vil motorkørslen stoppe. Nedre endestopindgang har ingen indflydelse under fremadkørsel.

Bemærk at en aktivering af en given endestopindgang vil medføre et øjeblikkeligt stop af motorkørslen uden hensyntagen til den indstillede deceleration.

Bemærk at program-udførelsen ikke bliver stoppet, kun selve motorkørslen.

## 2.8

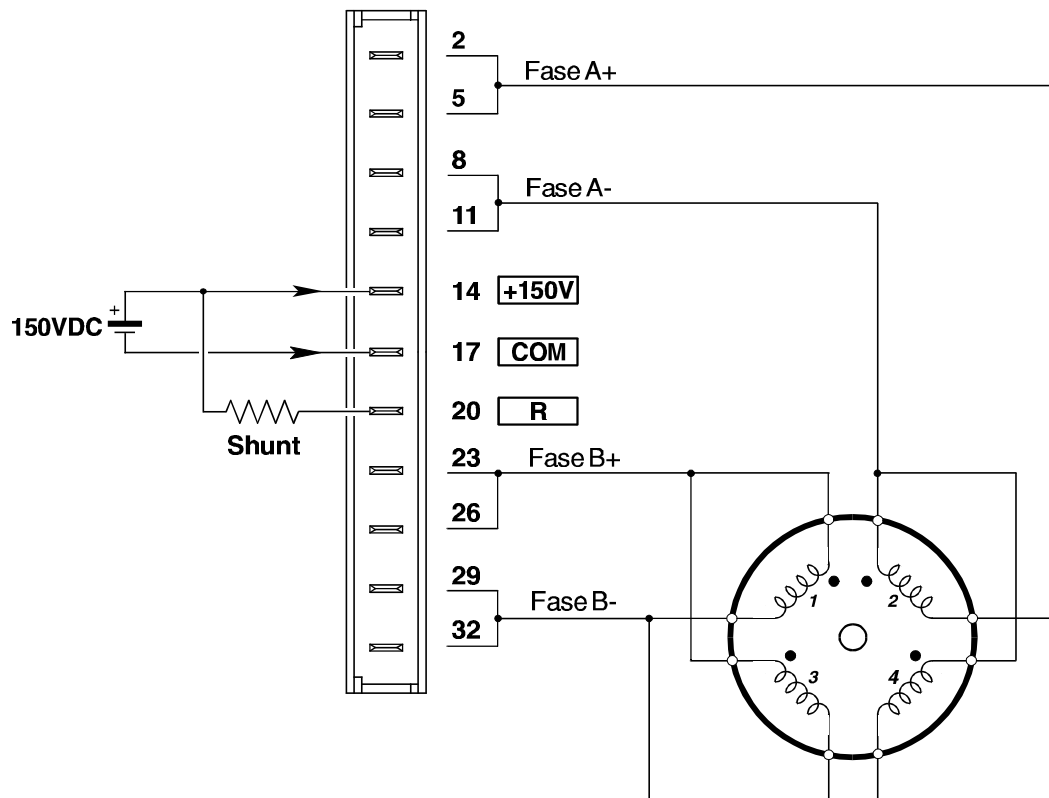
## Stopindgang



Ønskes det, at man med øjeblikkelig virkning skal stoppe programudførelsen og dermed motorkørslen, gøres dette ved at lægge stopindgangen til stel. Fjernes stelforbindelsen, vil programmet og dermed motor-kørslen fortsætte, og positionstællerens indhold vil ikke være tabt. Dog vil det pludselige stop sandsynligvis have medført, at motoren står i en udefineret position, eftersom der ved brug af stoppet ikke bliver taget højde for accelerations-/decelerations-ramperne (se under motorkommandoer afsnit 4.6).

### Forbindelse af motor og ekstern strømforsyning

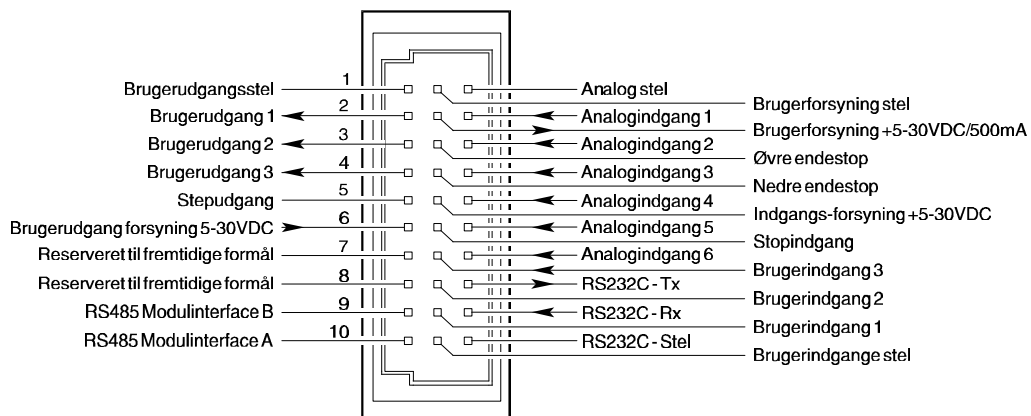
Stik: DIN41612 version H



Se endvidere afsnit 5.6 angående motorforbindelser

### Stikforbindelser for eksterne faciliteter

Stik: DIN41612 version C/3



## 3.1

# Interfaceforbindelser

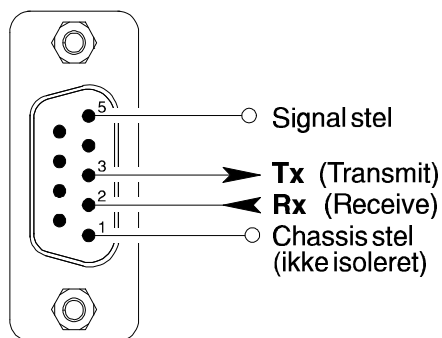
### Interfaceforbindelser.

Interfacet kører efter den meget anvendte RS232C standard, hvilket er en stor fordel, idet alle Personal Computers og standard terminaler har mulighed for at køre efter denne standard.

De 3 ledere Rx, Tx og stel anvendes.

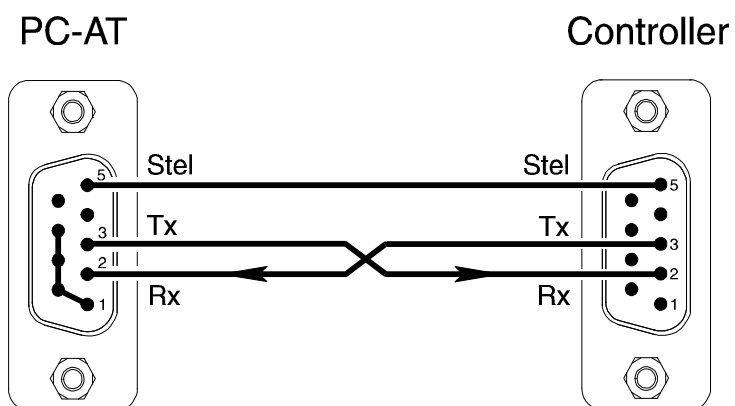
Kabellængden bør ikke overstige 10 meter, men bruges et længere kabel, kan der køres med checksum, se afsnit 3.5.

Forbindelser på controllerens interfacestik :



Skal man kommunikere fra en PC'er, kan de følgende 2 skemaer anvendes.

Forbindelser mellem controller og IBM AT, eller kompatibel :



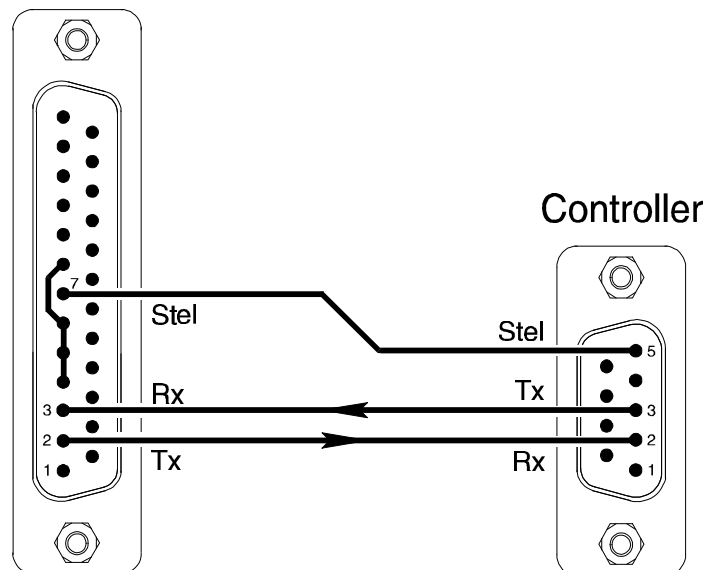
## 3.1

## Interfaceforbindelser

---

Forbindelser mellem controller og IBM XT/PS2 eller kompatibel :

PC-XT/PS2



## 3.2

## Adressering

Controlleren kan konfigureres til at reagere på alt, hvad der kommer på RS232 interfacelinien, også kaldet Point to Point. Der er mulighed for at koble op til 7 kontrollere på samme interfacelinie. For at opnå dette, skal dip-switchene stilles på en sådan måde, at hver controller får sin egen adresse, kaldet - Multipoint. Dette vil sige, at kontrolleren kun reagerer på kommandoer, der starter med den indstillede adresse. Hvis det ønskes at adressere, d.v.s have flere kontrollere koblet på samme interfacelinie, skal dip-switchen mærket *Tx-PD* stilles i stilling *ON* på en af de kontrollere, der indgår i multipoint konfigurationen og *OFF* på de resterende. *Tx-PD* stilles ligeledes i position *ON* ved Point to Point kommunikation.

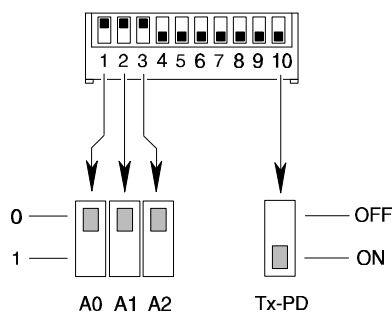
Af hensyn til kommunikationssikkerheden er det tilrådeligt at anvende checksum faciliteten. Dette gøres ved at sætte CHS switchen i stilling *ON*. Se endvidere afsnit 3.5.

Bemærk at modulinterfacet (RS485) også benytter denne adresse.

Til adresseindstillingen bruges følgende skema:

A0	A1	A2	Adresse	Protocol
0	0	0	-	Point to point
1	0	0	1	Multipoint
0	1	0	2	Multipoint
1	1	0	3	Multipoint
0	0	1	4	Multipoint
1	0	1	5	Multipoint
0	1	1	6	Multipoint
1	1	1	7	Multipoint

**VIGTIGT!** : Hvis adresse switchenes stilling er blevet ændret, skal der slukkes for kontrolleren, og dernæst tændes igen. Først da har kontrolleren registreret ændringen.



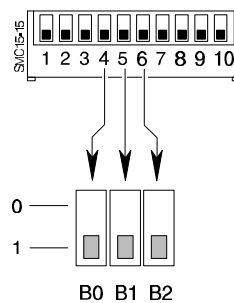
## 3.3

# Kommunikationshastighed

---

Der kan vælges kommunikationshastigheder (Baud rate) på mellem 110 og 9600 Baud. Denne hastighed skal sættes på de 3 dip switches, som er vist på nedenstående figur.

B0	B1	B2	Baud Rate
0	0	0	110
1	0	0	150
0	1	0	300
1	1	0	600
0	0	1	1200
1	0	1	2400
0	1	1	4800
1	1	1	9600



**VIGTIGT!** Hvis baudrate switchenes stilling er blevet ændret, skal der slukkes for controlleren, og dernæst tændes igen. Først da har controlleren registreret ændringen.

Hastigheden skal sættes på den terminal eller PC'er, som bruges til kommunikationen. Desuden skal denne sættes efter følgende format:

**(1 startbit)      7 databit      Odd paritet      1 Stop bit**

() Der er altid en startbit i RS232C/V24 protokollen.

Når man sender kommandoer til controlleren, er det en forudsætning, at man følger et bestemt format:

**Adresse**            **Kommando**            **Argument**            **Checksum**            **Return**

**Adresse :**    Adressen skal kun bruges, hvis der anvendes flere controllere på samme kommunikationslinie. Den tillægges værdien 1 - 7.

**Kommando :** Selve kommandoen. Se under software afsnit.

**Argument :**    Den efterfølgende talværdi for kommandoen.  
Visse kommandoer skal ikke bruge noget argument.  
Eks. *K* eller *Z* (se under software beskrivelser).

**Checksum :** Kan bruges i tilfælde af lange kommunikationslinier. Dette er blot en ekstra sikkerhed for, at kommandoen modtages korrekt. I tilfælde af fejl vil man modtage en fejlmeddelelse (*E1*), og man er derfor nødsaget til at sende kommandostrengen en gang til. (se næste side)

**Return:**        Har værdien 13 (ASCII). Dette fortæller controlleren, at kommandostrengen er fuldendt, og en oversættelse kan påbegyndes.



## 3.5

## Checksum

---

I industriel sammenhæng vil der ofte optræde elektrisk støj fra f.eks. elektromotorer. Denne støj kan komme helt vilkårligt, og selv en effektiv elektrisk filtrering vil ikke kunne fjerne denne støj 100%. Da det må anses for at være alt afgørende, at systemet fungerer helt efter hensigten, bør man gardere sig ved at vælge en kommunikationshastighed, der ikke ligger for højt. Endvidere bør kabellængden ikke overstige 10 meter. En typisk kommandostreng vil se ud som følgende:

**1A3%**

Der køres med adressering, og controller nummer 1 ønskes adresseret. Kommandoen lyder på, at udgang 3 skal aktiveres. Udregningen af checksumkarakteren foregår på følgende måde. Først findes ASCII værdien for hver af de karakterer, der indgår i kommandostrengen. Disse lægges sammen, og resultatet divideres med 128. Heltals resultatet smides væk, og resten ganges med 128 og udgør den endelige checksum. Det egentlige regnestykke ser således ud:

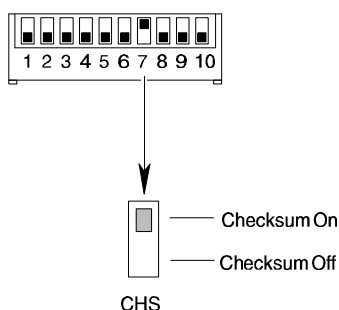
Adressekarakter	1	=	ASCII	49
Kommandokarakter	A	=	-	65
Argumentkarakter	3	=	-	51

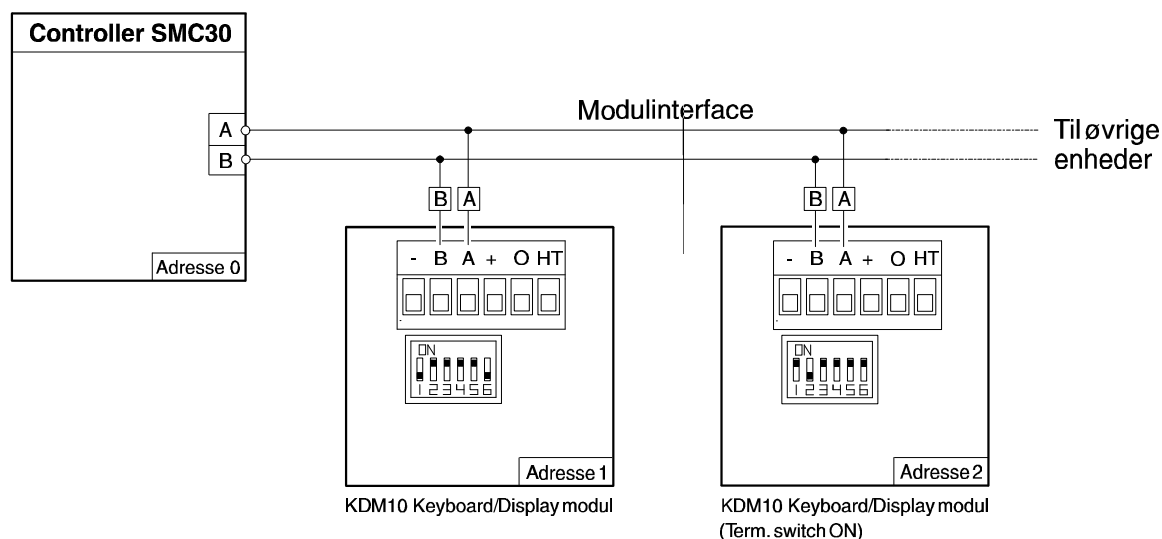
Resultat + rest =  $(49+65+51)/128 = 1,289..$

Checksum = Rest \* 128 =  $0,289 * 128 = 37$

I tilfælde af, at kommandostrengen bliver påvirket under transmissionen, vil checksummen ikke passe, og controlleren vil melde "E1", hvilket betyder, at controlleren ikke har kunnet forstå den indkomne kommandostreng. Kommandostrengen skal i så fald repeteres en gang til, og melder controlleren stadig "E1", bør kommunikationshastigheden sættes ned, eller ledningslængden mellem computer og controller gøres kortere. Checksumsfunktionen aktiveres ved at sætte dip-switchen CHS i stilling *Checksum On*.

**VIGTIGT!** : Hvis checksum-switchens stilling er blevet ændret, skal der slukkes for controlleren, og dernæst tændes igen. Først da har controlleren registreret ændringen.





Controlleren kan tilsluttes eksterne moduler som f.eks. keyboard/display-modul, ind-/udgangs-modul m.m.

Forbindelsen til disse eksterne moduler foretages via controllerens serielle RS485 interface. Interfacet udgøres af de 2 terminaler benævnt "A" og "B".

Via disse 2 terminaler styres alle de eksterne modulers funktioner. Op til 31 moduler og mindst 1 motorcontroller kan kobles på samme interface linie.

RS485 Interfacet tilbyder flere fordele idet det kører balanceret og kommunikationslinien er lav-impedant. Endvidere er de 2 interface-terminaler galvanisk isoleret fra de øvrige tilslutninger på controlleren.

Interfacet er beskyttet mod eventuelle transienter der kan forekomme på interfaceledningerne fra controller til modulet. Disse faktorer gør at der kan kommunikeres over store afstande, på trods af elektrisk støj.

Det kan anbefales at ledningsføringen mellem controlleren og de øvrige enheder på kommunikationslinien foretages med par-snoede ledninger.

I et system hvor kommunikationslængden overstiger 25 meter mellem 2 enheder skal dipswitchen mærket *TERM* stilles i position *ON* på de moduler der sidder mere end 25 meter væk fra de øvrige.

Se dipswitchens placering i brugermanualen for det aktuelle modul.

#### Adressering af moduler:

I et kommunikationssystem hvor flere enheder er koblet sammen, skal hver enhed indstilles til sin egen adresse. Denne adresse kan vælges i området 1-31.

Ovenstående skitse viser hvordan disse adresser i et typisk system er indstillet.

Det er ikke tilladt at lade flere moduler benytte den samme adresse. Hvis flere moduler benytter den samme adresse vil controlleren stoppe programafviklingen og melde fejl.

Bemærk at controllerens adresse til modulinterfacet er den samme som benyttes til RS232 interfacet. Se afsnit 3.2.

Det enkelte moduls adresse indstilles iflg. modulets brugermanual.

Før de enkelte kommandoer beskrives nærmere, er det en forudsætning at vide lidt om controllerens struktur.

Der findes 2 lagre i controlleren, der begge er tilgængelige for brugeren. Lagrenes formål er at huske de programmer og køreparametre, der bliver sendt til controlleren fra en computer eller terminal.

Første lager benævnes på de følgende sider som "*arbejdslager*". Det er et lager, der bruges ved kontinuerlig sammenkobling med en computer eller terminal. Alt hvad der lægges i dette lager, bliver slettet, når der slukkes for controlleren. Det kan også bruges som lager for et program under opbygning.

Det andet lager udgøres af en E<sup>2</sup>PROM, d.v.s et lager, der ikke bliver slettet, når der slukkes for controlleren. Dette lager benævnes "*arkiv*" på de følgende sider. Dette lager er tænkt anvendt, når controlleren er konfigureret som "*Stand alone unit*", d.v.s ikke er koblet sammen med nogen computer eller terminal.

Controlleren kan i så fald selv starte afviklingen af det program, der ligger gemt i "*arkivet*" uden nogen form for indblanding udefra.

Arkivet kan også bruges, selvom controlleren er forbundet til en computer. I såfald kunne dets formål f.eks. være at huske programsekvenser, som hyppigt bliver brugt.

#### **Positionstæller.**

Positionstælleren er det register, der holder trit med motorens position. Positionstælleren kan nulstilles ved at bruge *I* -(Initialiser) eller *H* (Home) kommandoen ( se under 4.3 og 4.4). Positionstælleren indhold kan endvidere aflæses/eller ændres via kommandoerne *V1* og *f[±n]*

Når positionstælleren når sit maximum på +8.388.607/-8.388.608, vil motoren stoppe automatisk.

#### **Kommandobeskrivelser.**

På de følgende sider (afsnit 4.3 - 4.6) er diverse kommandoer m.v. beskrevet.

For at undgå misforståelser skal det kort nævnes, hvad der menes med diverse tekstformater.

Alle kommandoerne er benævnt med et bogstav efterfulgt af et ord i parentes.

Den egentlige kommando er bogstavet, ordet i parentes er blot en hjælp for gøre det nemmere at huske, hvad kommandoen står for.

Stort set alle kommandoerne er efterfulgt af en talværdi, et + eller - tegn. Det er vigtigt, at man overholder de angivne rammer for talværdierne, eftersom controlleren ikke vil kendes ved talstørrelser, der er udenfor de angivne rammer.

Se endvidere afsnit 3.4 vedr. kommandoformat.

## 4.1

## Generelt om software

---

### Modes.

Der findes 3 modes controlleren kan stå i:

1) *Standby mode.*

Dette vil ske efter *K* (Kill), *Z* (Smooth stop) el. *PX* (Program exit) kommandoen. Ellers efter endt programudførelse.

2) *Programmeringsmode.*

Denne mode bruges, når et program indlæses. Brug *PO* (program) kommandoen for at komme i denne mode.

3) *Execute mode.*

Brug *E* (Execute) kommandoen og det indprogrammerede program kører.

Det stopper først, når alle kommandoer er udført, eller hvis der afbrydes med *K* eller *Z* kommandoen.

### Programmering.

Når et program opbygges, startes altid med kommandoen *PO*. Dette vil sige, at controlleren stilles i programmeringsmode. Dernæst kan det egentlige program indtastes.

Til slut skrives *E*, hvilket medfører, at man går fra programmeringsmode over til *Execute mode*. Programmet vil nu blive afviklet, og ønskes det evt. lagret, gøres dette med *M* (Memory save) kommandoen, efter programmet er stoppet. Programmeringsmode kan også afbrydes ved hjælp af *PX* (Program exit), i såfald vil controlleren stå i standby-mode.

Et programmeringsforløb er som følgende.

- |   | Mode:          |
|---|----------------|
| 1) Controlleren tændes.   | <i>Standby</i> |
| 2) <i>PO</i> (Program) kommandoen indtastes.  | <i>Program</i> |
| 3) Den ønskede programsekvens indtastes.  | <i>Program</i> |
| 4) <i>PX</i> (Program exit) kommandoen indtastes, hvorved controlleren går i standbymode.                 | <i>Standby</i> |
| 5) <i>E</i> (Execute) kommandoen indtastes.   | <i>Execute</i> |
| * Program afvikles færdigt, eller der afbrydes med <i>K</i> (Kill) el. <i>Z</i> (Smooth stop) kommandoen. |                |
| 6) Program kan gemmes ved at indtaste <i>M</i> - (Memory save) kommandoen.                                | <i>Standby</i> |

Det bør tilføjes, at fejlmeddelelsen *E1* sandsynligvis vil forekomme, første gang der kommunikeres. Dette skyldes de transienter, der vil forekomme på interface ledningerne, når computer eller controller tændes.

## 4.1

## Controllermeddelelser

---

Hver gang controlleren modtager en kommando eller en forespørgsel, vil den svare tilbage med en kort meddelelse.

Formatet på den streng, der bliver returneret, ser ud som følgende :

Svarkode	Argument	Checksum	Carriage-Return
----------	----------	----------	-----------------

Svarkoden - er den egentlige meddelelse og er en af følgende:

- Y** = (Yes) Kommando er modtaget og vil blive eller er efterkommet.
  
- B** = (Busy) Controlleren er optaget med programudførelse og er ikke klar til at modtage kommandoen eller forespørgslen.
  
- R** = (Ready) Controlleren er parat til at efterkomme kommandoen.
  
- V** = (Verify) Status vedr. position eller brugerind/udgange. Denne meddelelse vil kun forekomme, hvis controlleren er blevet forespurgt vedr. dette.  
Se beskrivelserne af kommandoerne V1 og V2, afsnit 4.3 og 4.5.
  
- E** = (Error) Der er fundet fejl i den modtagne kommandostreng, og controlleren er ikke istand til at efterkomme denne.  
Argumentet specificerer, hvilke fejltypen det drejer sig om.
  - E1** Paritetsfejl efter at have modtaget en eller flere karakterer. Checksumsfejl. Den sendte kommandostreng har været for lang.
  - E2** Det argument, der efterfulgte kommandoen, har været for langt eller er ikke nødvendigt.
  - E3** Der er ikke plads i arbejdslageret.
  - E4** Der har været anvendt en kommando, controlleren ikke kender eller ikke har været istand til at efterkomme.
  - E5** Positionstælleren ville have overskredet sit maksimum på -8.388.608 el. +8.388.607 step, hvis motorkørslen ikke var blevet stoppet.  
Fejl i køreparametre ( R, S, T ).
  - E6** Der har været fejl under transmission til/fra arkiv lageret.

**Argumentet** - vil kun forekomme efter E (error) eller V (Verify) meddelelsen og kan være på 1 til 7 karakterer.

**Checksum** Vil kun indgå i strengen, såfremt checksumswitchen er i stilling *ON*.  
Se checksumbeskrivelse, afsnit 3.5.

**Carriage-Return** - afslutter strengen og har ASCII-værdien 13.

## 4.2

## Kommandooversigt

---

### Systemkommandoer :

E	(Execute)	Starter afvikling af program.
f [ $\pm$ nnnnnnn]	(Forcing pos.)	Indlæser ny position.
F	(Feedback)	Status forespørgsel til controller.
I [n]	(Initialize)	Nulstiller controllerregistre (software reset).
K	(Kill)	Stopper afvikling af programudførelse.
M	(Memory)	Gemmer arbejdsprogram i arkiv (E <sup>2</sup> PROM).
PE	(Program enter)	Sætter controlleren i programmeringsmode uden at slette gammelt program.
PO	(Program)	Sætter controlleren i programmeringsmode. Kommandoen sletter et evt. gammelt program.
PX	(Program exit)	Forlad programmeringsmode.
TP	(Temperature)	Udlæser den aktuelle temperatur i controlleren.
V1	(Verify)	Viser positiontællerens indhold.
X	(Recall)	Henter program fra arkiv til arbejdslager.
Z	(Smooth stop)	Stopper langsomt programafvikling med hensyntagen til decelerationsrampe.

### Motorkommandoer :

[ $\pm$ nnnnnnn]		Relativ positionering angiver omdrejnings retning (+/-) og antal step.
$\pm$ A [n].[n1-n2]		Relativ positionering kontrolleres af spænding på analogindgang.
CR [nnnn]	(Current Ramp)	Fastsætter motorstrøm under acceleration.
CS [nnnn]	(Current Start)	Fastsætter motorstrøm under stilstand.
CT [nnnn]	(Current Top)	Fastsætter motorstrøm ved tophastighed.
g [ $\pm$ ]	(Velocity)	Kontinuerlig kørsel frem/tilbage.
G [ $\pm$ nnnnnnn]	(Goto)	Absolut positionering.
G $\pm$ A [n].[n1-n2]	(Goto)	Absolut positionering kontrolleres af spænding på analogindgang.
H [ $\pm$ ]	(Home)	Nulstiller motor og elektronik.
N [n1n2.n3n4]	(Input setup)	Starter/stopper motor ifølge brugerindgange.
NA [p1.p2]	(Input setup)	Starter/stopper motor ifølge analogindgange.
R [nnnnn]	(Ramp)	Acceleration/decelerationparameter (1-10000step).
RT [nnnn]	(Ramp time)	Acceleration/decelerationparameter (0,01-10sek).
RS [nnnnn]	(Ramp slope)	Acceleration/decelerationparameter (10-30000step/s <sup>2</sup> ).
S [nnnn]	(Start rate)	Min. hastighed.
T [nnnnn]	(Top rate)	Max. hastighed.
r [n1.n2]	(A/D Ramp)	Som R, styret af spænding på analogindgang.
s [n1.n2]	(A/D Start rate)	Som S, styret af spænding på analogindgang.
t [n1.n2]	(A/D Top rate)	Som T, styret af spænding på analogindgang.
VR	(Verify Ramp)	Viser det aktuelle acceleration/decelerationparameter.
VS	(Verify St.rate)	Viser det aktuelle start rate parameter.
VT	(Verify Top rate)	Viser det aktuelle top rate parameter.

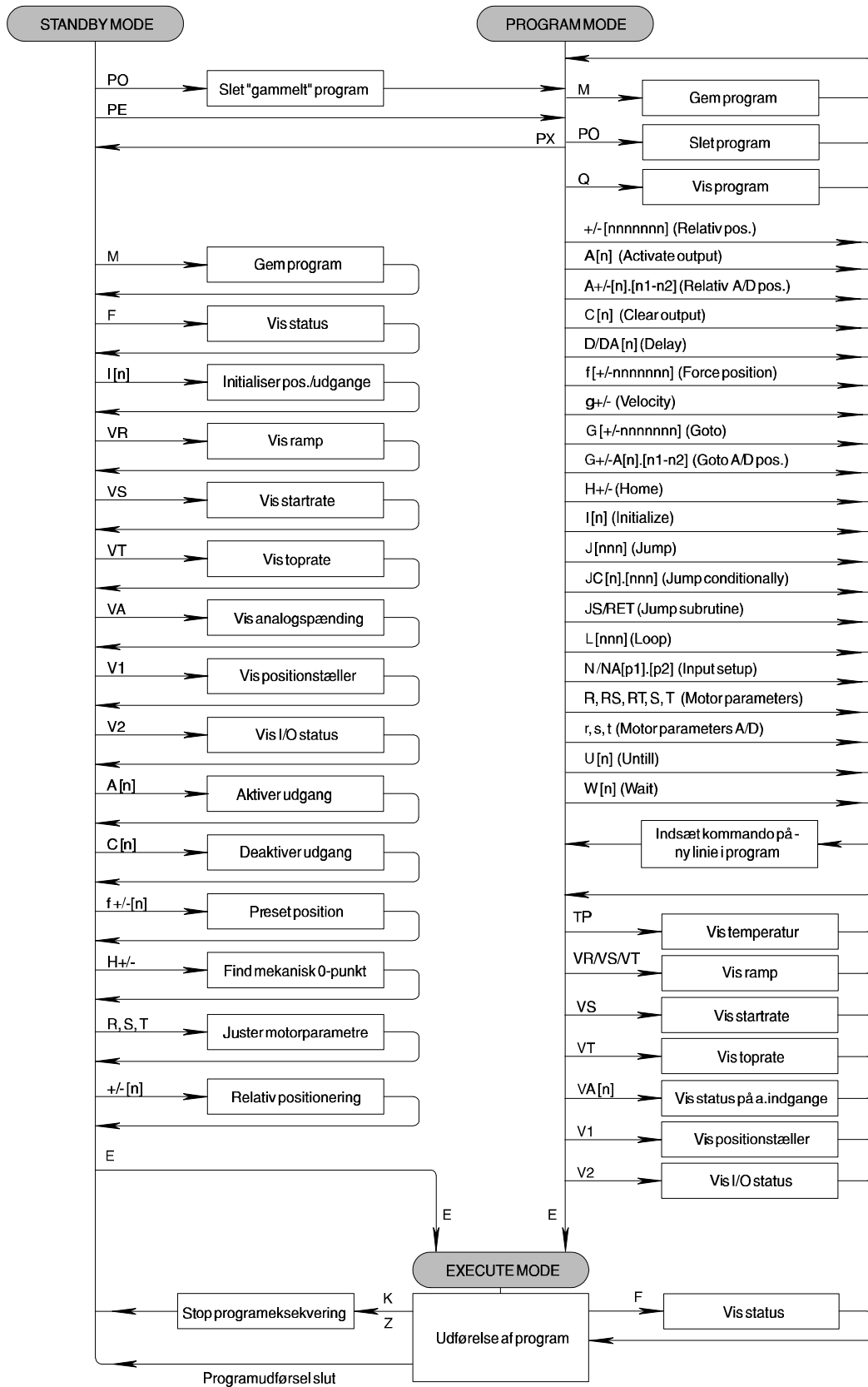
(Fortsættelse)

**Brugerinterface :**

A [n]	(Activate)	Tænder en udgang.
C [n]	(Clear)	Slukker en udgang.
U [n]	(Until)	Gentager program(del), indtil en given indgang aktiveres.
VA [n]	(Verify ainput)	Viser spændingen målt på en af de 6 analogindgange
VA		Viser logiske niveauer på analogindgange.
V2	(Verify)	Viser brugerinterface status.
W [n]	(Wait for)	Holder en pause i programafviklingen, indtil en given indgang aktiveres.

**Flowkommandoer :**

D [nnn]	(Delay)	Venter en specificeret tid.	
DA [n].n1-n2]	(Analog Delay)	Venter en specificeret tid, kontrolleret af analog spænding.	
J [n1]	(Jump)	Ubetinget spring til programlinie.	
JC [n].[n1]	(Jump con.)	Betinget spring til programlinie.	JCA [p].[n1] (Jump con.)
Retur fra subrutine. L [nnn]	(Loop)	Gentager programdel et specificeret antal gange.	Udvidet kommandosæt + - / * De fire regnearter VR[0-510]
Angiver skaleringsforhold mellem aktuel distance og step. registre til eksternt modul. IF [p1 m p2]	(IF)	Hvis parameter er opfyldt, udføres efterfølgende linie.	PRINT[n1.n2.n3] (PRINT) Udskriver
INPUT[n1.n2.n3] (INPUT)		Henter data fra eksternt modul til register.	AO[a].[o] (Ver





## 4.3

# Systemkommandoer

---

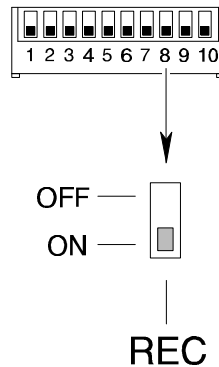
- E**  
(Execute) Starter programafvikling. Kommandoen kan også bruges til at slutte en programmeringssekvens. Kommandoen kan både bruges, når controlleren er i standbymode og i programmeringsmode.
- f+/-[nnnnnnn]**  
(Forcing pos.) Tillægger positionstælleren en specificeret position. Denne position kan defineres i området fra -8.388.608 til og med +8.388.607. Kommandoen kan både bruges, når controlleren er i standbymode og i programmeringsmode.
- Eksempel:*  
f+100 vil medføre, at positionstælleren får værdien +100.
- F**  
(Feedback) Statusforespørgsel til controlleren. Der vil blive returneret 1 ud af 3 svar.
- 1) Er controlleren klar til at udføre kommandoer, vil svaret være R (Ready).
  - 2) Er controlleren optaget, vil den returnere svaret B (Busy).
  - 3) Hvis motoren er blevet stoppet automatisk p.g.a overflow i positionstælleren, vil svaret være E5 (Error 5).
- I [1-3]**  
(Initialize) Kommandoen bruges til at nulstille enten positionstæller og/eller brugerudgange
- I1** = Nulstiller kun positionstæller.
- I2** = Brugerudgange bliver nulstillet.
- I3** = Positionstæller og brugerudgange bliver nulstillet.
- Se også afsnit 4.7 - kommandoerne I4 - I7.
- K**  
(Kill) Denne kommando har højeste prioritet, idet den stopper programudførelsen, uanset om motoren kører eller ej. Kommandoen vil have øjeblikkelig virkning, d.v.s. den sætter controlleren i standby, og en ny start af programmet skal foretages med E (Execute) kommandoen. Programmet vil da blive afviklet helt fra begyndelsen. Det kan være nødvendigt at bruge H (Home) kommandoen inden en ny start, eftersom motoren vil stå i en vilkårlig position på grund af det pludselige stop, der opstår ved brug af Kill kommandoen.

## 4.3

# Systemkommandoer

---

- M** (Memory save) For at undgå, at det færdige program går tabt, når controlleren er slukket, bør denne kommando benyttes, eftersom den gemmer programmet i et "arkiv". Der kan kun ligge et program i arkivet ad gangen. Når controlleren senere tændes, vil programmet automatisk blive eksekveret, hvis dip-switchen mærket *REC* er i stilling "ON".



- PE** (Program enter) Denne kommando bruges til at gå i programmeringsmode uden at slette et evt. gammelt program, der i forvejen ligger i arbejdslageret.  
Kommandoen bruges primært, når man er igang med at opbygge et program og man ønsker at bygge videre på programmet uden at indtaste det hele igen.
- PO** (Program) Programkommandoen sætter controlleren i programmeringsmode.  
Dette vil sige, at controlleren er parat til at modtage de programkommandoer, der indlæses.  
Hver gang kommandoen bruges, slettes det program, der måtte ligge i arbejdslageret. Se endvidere omtalen af kommandoen i starten af dette kapitel.
- PX** (Program exit) Controlleren vil forlade programmeringsmode og stille sig i standbymode.  
Programmet kan nu afvikles, eller et nyt program kan indtastes.

## 4.3

# Systemkommandoer

---

- TP**  
(Temperature) Viser den aktuelle temperatur inde i controlleren.  
Den kan bruges til at verificere, om controlleren arbejder indenfor sit temperaturområde på 0-50°C. Hvis der, under "worst-case" betingelser i det opbyggede system måles temperaturer over ca. 60°C, skal kølingen af controlleren forbedres.
- V1**  
(Verify pos.) Via denne kommando er det muligt at aflæse positionstællerens indhold.  
Værdien der bliver returneret, refererer til 0, det vil sige Home positionen.  
(Se under Home kommandoen)
- X**  
(Recall prog.) Det program, der evt. måtte ligge i arkivet, kan hentes ind i arbejdslageret via X kommando. Dette kan f.eks være en fordel, hvis controlleren med jævne mellemrum skal udføre det samme stykke program. Send først X og dernæst E kommando. Det vil starte programmet i arkivet øjeblikkeligt.  
Husk, at hver gang et program hentes fra arkivlageret til arbejdslageret, vil det program der ligger i arbejdslageret blive slettet.
- Z**  
(Smooth stop) Denne kommando svarer fuldstændigt til Kill kommando, bortset fra, at motoren decelerer iflg. de indkodede parametre ( R, S, T ).  
Man opnår derved, at motoren ikke stopper i en udefinerbar position som ved brug af *K* (kill).

## 4.4

# Motorkommandoer

**+/- [nnnnnnn]** Ligner Goto kommandoen. I stedet for at positionere i forhold til 0, positioneres i forhold til den position motoren står i, før kommandoen bliver udført også kaldet relativ positionering. Der specificeres, hvilken vej positioneringen skal foregå, ved at bruge + eller - . Den efterfølgende værdi angiver, hvor mange step motoren skal dreje i den pågældende retning. Værdien kan sættes fra 1 til 8.388.607 step.

*Eksempel :*

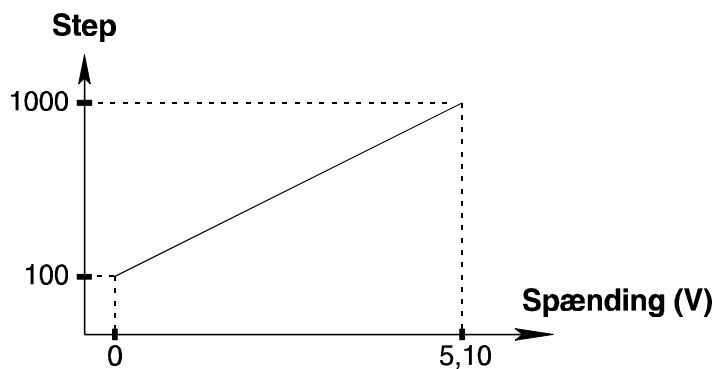
Specificeres +15 , vil motoren gå 15 step frem i forhold til, hvor den stod.

**±A [n].[n1-n2]** Skal motoren køre et bestemt antal step afhængig af en analog spænding, kan denne kommando anvendes. n specificerer, hvilken analogindgang (1-6) der benyttes som måleindgang. n1 og n2 angiver step-intervallet, hvor n1 angiver antal step ved 0V, og n2 angiver antal step ved 5,10V. n1 og n2 kan sættes fra 1 til 65000 step.

*Eksempel:*

+A1.100-1000

Denne kommando vil medføre, at motoren kører 100 step, hvis der påtrykkes 0V på analogindgang 1, og 1000 step hvis der påtrykkes 5,10V. Der vil ved spændinger imellem 0 og 5,10V blive foretaget en linær interpolering imellem 100 og 1000 step afhængig af spændingen.



Strømmen til stepmotoren kan justeres henholdsvis for stilstand, acceleration/deceleration og kørsel. Som regel kan man nøjes med en lille motorstrøm, når motoren står stille, eftersom en typisk stepmotors holdemoment er væsentligt større end køremomentet, alt afhængigt af det hastighedsområde stepmotoren opererer i.

En stepmotors moment er ligefrem proportionalt med den tilførte strøm. Dette forhold gør sig kun gældende, indtil strømmen overstiger den aktuelle motors specificerede fasestrøm (se den pågældende motors datablad).

Overstiges motorens nominelle strøm, vil motoren blive overophedet, og den overskydende strøm vil kun i ringe grad give sig udslag i forøget motormoment.

Nedenstående kommandoer benyttes til at specificere motorstrømmen. De kan indsættes et vilkårligt sted i programmet. Alle 3 parametre kan fastsættes/ændres løbende igennem hele programmet.

Hvis der i et givent program ikke er specificeret en eller flere af strømparametrene, vil det/de undladte parametre indtage værdien 1000mA.

<b>CS [0-12000]</b>	(Current Standby) Fastsættelse af motorstrømmen når motoren ikke kører.
<b>CR [0-12000]</b>	(Current Ramp) Fastsættelse af motorstrømmen ved acceleration/deceleration.
<b>CT [0-12000]</b>	(Current Top) Fastsættelse af motorstrømmen når motoren kører med topfart.

Bemærk at alle 3 parametre kun kan specificeres i intervallet 0 til 6000 mA, hvis programmet køres i stepmotor controller SMC30B (6Amp. version).

*Eksempel:*

*(Program)*

.

CS500	Fastsætter motorstrøm ved stilstand til 500mA (0,5A).
CR6000	Fastsætter motorstrøm under acceleration/deceleration til 6000mA (6A).
CT4000	Fastsætter motorstrøm ved tophastighed til 4000mA (4A).
+100	Motor kører 100 step frem.
CT5500	Fastsættelse af ny motorstrøm ved tophastighed til 5500mA (5,5A).

.

.

- g+/-**  
(Velocity mode) Ønskes det, at motoren drejer kontinuerligt i en bestemt retning, anvendes denne kommando.  
Kommandoen efterfølges af + eller -, der specificerer retningen. Den eneste måde, at stoppe motoren på, er ved brug af Z eller K kommandoen.  
Hvis N kommandoen er udført før  $g_{\pm}$  kommandoen, kan N kommandoens betingelser også stoppe motorkørslen (se beskrivelse af N kommando).  
Det bør bemærkes, at positionstælleren bliver ajourført under afviklingen af denne kommando. Kommandoen kan kun afvikles, når den indgår i et program.
- G+/- [nnnnnnn]**  
(Goto) Denne kommando udfører en absolut positionering. Værdien, der anføres efter kommandoen, refererer til positionstælleren. Den anførte position kan tillægges en værdi i området -8.388.608 og +8.388.607.
- G±A [n].[n1-n2]**  
(Goto) Kommandoen udfører en absolut positionering som  $G_{\pm}[n]$ , dog er den ønskede position bestemt af en analog spænding påtrykt en given analogindgang. n fastsætter hvilken analogindgang (1-6) der skal benyttes som måleindgang. n1 og n2 angiver den ønskede position ved henholdsvis 0V og 5,10V. Den ønskede position kan vælges i området +0 til +65000.  
Se endvidere kommandoen  $\pm A$ .
- Eksempel:*  
G±A2.0-800
- Kommandoen medfører at motoren kører til position +0, hvis der påtrykkes 0V på analogindgang 2, og +800 hvis der påtrykkes 5,10V.  
Der vil ved spændinger mellem 0 og 5,10V blive foretaget en lineær interpolering mellem positionen +0 og +800. Se endvidere beskrivelse af kommandoen  $\pm A$ .
- H+/-**  
(Home) Denne funktion gør det muligt at nulstille systemet til en referenceposition både mekanisk og elektrisk. I samme øjeblik controlleren modtager denne kommando, vil motoren køre i den specificerede retning enten H+ eller H-.  
I samme øjeblik EOT (End of travel) indgangen bringes lav, vil motoren stoppe. Motoren er nu sat i sin udgangsposition. Hastigheden, hvormed der bliver nulstillet, svarer til den hastighed, der er specificeret i S ( Start rate ) kommandoen.  
Efter Home kommandoen er eksekveret, vil positionstælleren have indholdet +0 .

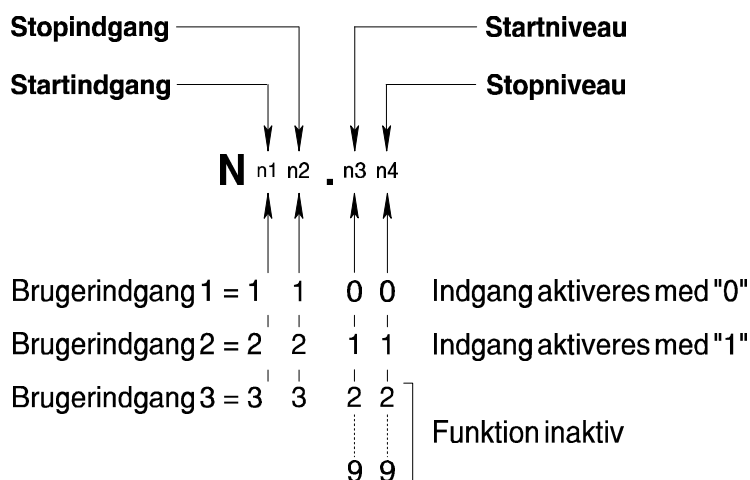
## 4.4

# Motorkommandoer

**N [n1n2.n3n4]** Kommandoen gør det muligt, at starte og stoppe motorkørslen via styrespænding-er på brugerindgangene.

Kommandoen selv starter ikke motorkørslen. N kommandoen har kun indflydelse på den førstkommende motorkommando bestemt af motorkommandoerne  $\pm[n]$  /  $g[\pm]$  /  $G\pm[n]$ . Kommandoen vil herefter være inaktiv. Hvis det ønskes at starte/stoppe motoren igen via betingelser i en N kommando, skal N kommandoen udføres før den pågældene motorkommando.

*Kommandoformat:*



**n1:** Henviser til, hvilken brugerindgang (1-3) der skal bruges til at starte motorkørslen.

**n2:** Henviser til, hvilken brugerindgang (1-3) der skal bruges til at stoppe motorkørslen.

**n3:** Hænger sammen med n1, idet n3 bestemmer, hvilket logisk niveau der skal påtrykkes den valgte brugerindgang for at starte motoren. Hvis n3 sættes til 0, vil motoren starte, når den valgte brugerindgang påtrykkes det logiske niveau 0. Hvis n3 sættes til 1, vil motoren starte, når den valgte brugerindgang påtrykkes det logiske niveau 1.

Hvis n3 tillægges et tal fra 2 til 9, vil startfunktionen være inaktiv, og motoren vil starte med det samme.

**n4:** Hænger sammen med n2, idet n4 bestemmer, hvilket logisk niveau der skal påtrykkes den valgte brugerindgang for at stoppe motoren. Hvis n4 sættes til 1, vil motoren stoppe, når den valgte brugerindgang påtrykkes det logiske niveau 1.

Hvis n4 sættes til 0, vil motoren stoppe, når niveauet går fra logisk 1 til logisk 0.

Hvis n4 tillægges et tal fra 2 til 9, vil stopfunktionen være inaktiv, og motorkørslen vil først blive stoppet, når en af de førnævnte motorkommandoer, der har startet motoren, kræver det.

(Fortsættes)

(Fortsættelse)

**N [n1n2.n3n4]**

(Input setup)

Når N kommandoen er impliceret i et kørselsforløb, vil positionstælleren blive ajourført, som var det et normalt kørselsforløb.

Når motoren kører, kan man bruge Z (Smooth stop) eller K (Kill) kommandoen til at stoppe motorkørslen. Hvis indexeren står og venter på startsignal fra en given brugerindgang kan programafviklingen stoppes med K (Kill) eller Z (Smooth stop).

*Eksempel 1:*

N13.01 vil, når g+ kommandoen bliver udført, starte motoren, når brugerindgang 1 bliver tilført en spænding, der svarer til logisk 0. Bemærk, at det er logisk niveau "0", og ikke et skift fra "1" til "0", der aktiverer start.

Motoren vil køre efter de specificerede køreparametre og køre med normal hastighed, indtil brugerindgang 3 bliver tilført en spænding, der svarer til logisk "1". Derefter vil motoren decelerere, indtil den stopper, og næste kommando udføres.

*Eksempel 2:*

N21.10 vil, når +10000 bliver udført, starte motoren, når brugerindgang 2 bliver "1" og køre med normal hastighed indtil 10000 step (-dec.step) er kørt, eller indtil brugerindgang 1 bliver udsat for et skift fra logisk 1 til logisk 0.

Motoren vil herefter decelerere, og næste kommando udføres.

*Eksempel 3:*

N11.19 vil, når G+3500 bliver udført, starte motoren, når brugerindgang 1 bliver logisk "1" og stoppe, når positionen +3500 er nået. Motoren vil køre efter de angivne parametre sat i R,S og T kommandoen.

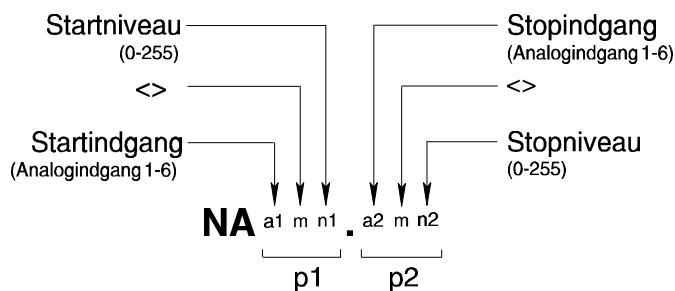


**NA [p1.p2]**  
(Input setup)

Kommandoen gør det muligt at starte og stoppe motorkørslen via analogindgange. Princippet i kommandoen er som N [n1n2.n3n4].

Kommandoens to parametre (p1 og p2) angiver henholdsvis start og stop-betingelsen. Kommandoen selv starter ikke motorkørslen. NA kommandoen har kun indflydelse på den førstkommande motorkommando bestemt af motorkommandoerne  $\pm[n]$  /  $g[\pm]$  /  $G\pm[n]$ . Kommandoen vil herefter være inaktiv. Hvis det ønskes at starte/stoppe motoren igen, via betingelser i en NA kommando, skal NA kommandoen udføres før den pågældende motorkommando.

Kommandoens totale format ser således ud:



**p1:** Hvis startbetingelsen er opfyldt, startes motorkørslen. Hvis p1 tillægges karakteren X, vil en start af motorkørslen være ubetinget, og motoren starter med det samme.

**p2:** Hvis stopbetingelsen er opfyldt, stoppes motorkørslen. Hvis p2 tillægges karakteren X, vil et stop af motorkørslen være ubetinget, og motoren vil køre, indtil den valgte motorkommando ( $\pm[n]$ ,  $g\pm$ ,  $G\pm[n]$ ) kræver det.

**a1:** Angiver den ønskede startindgang. a1 kan angives i området A1 til A6, svarende til analogindgang 1-6.

**a2:** Angiver den ønskede stopindgang. a2 kan angives i området A1 til A6, svarende til analogindgang 1-6.

**n1:** Referenceværdi for start. Referenceværdien sammenlignes med det målte på den pågældende analogindgang. Referencen kan tillægges en værdi i området 0 til 255.

**n2:** Referenceværdi for stop. Referenceværdien sammenlignes med det målte på den pågældende analogindgang. Referencen kan tillægges en værdi i området 0 til 255.

**m:** Sammenligning mellem referenceværdi og det målte. Hvis man ønsker, at motoren skal starte/stoppe, når det målte på analogindgangen er mindre end referenceværdien skal, "<" anvendes. Hvis motoren skal starte/stoppe, når det målte er større eller lig med referenceværdien, skal ">" tegnet anvendes.

## 4.4

# Motorkommandoer

---

(Fortsættelse)

**NA [p1.p2]**  
(Input setup)

Da n1 og n2 bliver indsat som en værdi i intervallet 0-255, og spændingen der måles ligger i området 0-5,10V, skal man ved bestemmelse af n1/n2 konvertere værdien til en spænding eller visa-versa iflg. nedenstående formler:

$$V_{ref} = 0,02 \times n \quad \text{eller} \quad n = 50 \times V_{ref}$$

*Eksempel:*

Man ønsker en referenceværdi på 1,20V.

$$n = 50 \times 1,2 = 60$$

*Eksempel 1:*

```
.  
.   
NAA1<60.A6>100  
+10000  
.
```

Hvis spændingen på analogindgang 1 er mindre end 1,2 Volt (se ovenstående beregning), starter motoren, ellers udføres ingenting. Når motoren kører, stoppes den enten når de 10000 step er kørt eller når spændingen på analogindgang A6 er større eller lig med 2V (n2=100).

*Eksempel 2:*

```
.  
.   
NAX.A5>220  
G+100000  
.
```

X henviser til at startbetingelsen er deaktiveret, og motoren vil derfor starte med det samme og først stoppe, når positionen +100000 er nået, eller hvis spændingen på analogindgang 5 er større end eller lig med 4,4V (n2=220).

K og Z kommandoerne kan bruges til at afbryde programeksekveringen.

## 4.4

# Motorkommandoer

En stepmotor er elektrisk kommuteret i modsætning til en normal DC motor, som er "Selvkommuterende". Dette vil sige at Stepmotoren drives frem af magnetfelter, der kontrolleres elektronisk. Belastes motoren, vil magnetfelterne før eller siden ikke være kraftige nok til at kunne blive ved med at trække akslen (rotoren). Motoren vil gå i stå, men elektronikken vil fortsat blive ved med at flytte magnetfelterne med samme hastighed, som om intet var hændt. Det er derfor vigtigt, at motoren accelereres og decelereres i et passende tempo, eftersom magnetfelterne skal have en chance for at trække akslen (rotoren) med sig op eller ned i hastighed. Ligeledes har en stepmotor en maksimal kørehastighed, og kommer man over denne, yder motoren ikke længere den samme kraft og vil gå helt i stå.

Der findes 3 grundlæggende parametre at tage hensyn til:

### S [16-2000] (Start rate step/sek)

Dette er den hastighed, som motoren startes ved. Sættes denne for højt, vil motoren ikke følge med, men blot stå stille i en vilkårlig position.

Den ønskede værdi kan sættes fra 16 til 2000 step/sek. Defaultværdi: 100step/sek

### T [16-15000] (Top rate step/sek)

Den maksimale hastighed motoren skal køre ved. Sættes denne for højt, vil motoren ikke yde nok kraft og vil gå i stå i en vilkårlig stilling.

Værdien kan specificeres i området 16 til 15000 step/sek. Defaultværdi: 1000step/sek

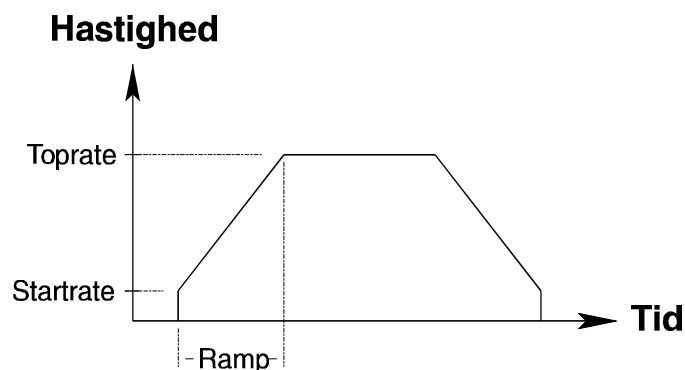
### R [1-10000] (Ramp) / RT [1-1000] (Ramp time) / RS [10-30000] (Ramp slope)

Denne værdi angiver, hvordan motoren skal accelerere og decelerere. Dette kan angives på 3 måder. R bruges, hvis acc./dec. ønskes angivet i step. RT bruges, hvis acc./dec. ønskes angivet i tid, og RS bruges, hvis acc./dec. ønskes angivet i step/sek<sup>2</sup>. RS kan med fordel anvendes, hvis der i et program gentagne gange ændres top- eller start-hastighed, eftersom accelerationen pr. tidsenhed forbliver den samme. Vælges en for hurtig accelerationen/decelerationen, vil motoren gå i stå. R [n] kan specificeres fra 1 til 10000 step. RT [n] kan specificeres fra 1 til 1000, hvilket angiver fra 0,01 til 10 sek. RS [n] kan specificeres fra 10 til 30000 step/sek<sup>2</sup>.

#### Eksempler:

R100 giver en acceleration/deceleration på 100 step. RT50 giver en acceleration/deceleration på 0,5 sek.

RS900 giver en acceleration/deceleration på 900 step/sek<sup>2</sup>. Defaultværdi for R : 100step.



Alle 3 parametre skal fastsættes i programmet og kan ændres løbende igennem hele programmet. Hvis T (toprate) vælges mindre end S (startrate), vil motoren køre med hastigheden T uden at accelerere eller decelerere.

## 4.4

## Fejl! Henvisningskilde ikke fundet. Motorkommandoer

**r [n1.n2]** (A/D ramp step) Det er muligt via disse 3 kommandoer at fastlægge køreparametrene R, S og T ved hjælp af en analog spænding, som påtrykkes en af de 6 analogindgange.

Kommandoen r [n1.n2] fastsætter ramp step parameteren.

**s [n1.n2]** (A/D start rate) Kommandoen s [n1.n2] fastsætter start rate parameteren, og kommandoen [n1.n2] fastsætter top rate parameteren.

**t [n1.n2]**

(A/D top rate) n1 henviser til, hvilken analogindgang der skal bruges til den givne step/frekvens bestemmelse. n1 skal være mellem 1-6, og spændingen, der påtrykkes den aktuelle analogindgang, skal være mellem 0 og 5,10V.

n1	Analogindgang
1	AN1
2	AN2
3	AN3
4	AN4
5	AN5
6	AN6

SMC15-29

n2 henviser til, hvad fuld udstyring (5,10V) skal svare til i step - kommandoen r [n1.n2] eller frekvens - kommandoen s [n1.n2] eller t [n1.n2].

n2 kan være et tal mellem 1 og 10, se nedenstående tabel.

n2	r[n1,n2] step	s[n1,n2] step/sek	t[n1,n2] step/sek
1	100	100	1000
2	200	200	2000
3	300	300	3000
4	400	400	4000
5	500	500	5000
6	600	600	6000
7	700	700	7000
8	800	800	8000
9	900	900	9000
10	1000	1000	10000

SMC15-29

Ved 0V på en given analogindgang vil det altid svare til 16 step ved kommandoen r [n1.n2], og 16 step/sek hvis kommandoen s [n1.n2] eller t [n1.n2] er valgt.

Hvis der er tvivl om, hvilke parametre motoren har kørt efter, kan VR, VS og VT anvendes - se omtale andetsteds i dette kapitel.

(Fortsættes)

## 4.4

# Motorkommandoer

---

(Fortsættelse)

*Eksempel:*

**r [n1.n2]**  
(A/D ramp step) Kommandoen t1.4 anvendes i et program. Ved eksekvering af programmet vil controlleren måle spændingen 2,5V på analogindgang 1 (AN1). Denne spænding vil blive konverteret til en frekvens på:

**s [n1.n2]**  
(A/D start rate)

$$\frac{2,5 \times 4000}{5,10} = 1953 \text{ Hz} = 1953 \text{ step/sek}$$

**t [n1.n2]**  
(A/D top rate) Denne frekvens vil så blive top rate ved næste motorkørsel. 0V vil svare til 16Hz, og 5,10V til 4000Hz.

**VR**  
(Verify ramp) Med disse 3 kommandoer kan man få oplyst den aktuelle værdi af de 3 køreparametre R,S og T.

**VS**  
(Verify start rate) VR henviser til ramp step "R" angivet i step. VS henviser til start rate "S" angivet i step/sek. VT henviser til top rate angivet i step/sek.

**VT**  
(Verify top rate) *Eksempel:* Det ønskes at få oplyst, hvad den aktuelle top rate (tophastighed) er, derfor sendes følgende streng til controlleren:

*VT (carriage return)*

Controlleren vil herefter svare:

*T1000 (carriage return)*

Dette angiver, at den aktuelle top rate er 1000 step/sek.

VR, VS og VT kan også bruges til at få oplyst de køreparametre der er bestemt via kommandoerne *r [n1.n2]*, *s [n1.n2]* og *t [n1.n2]*.

## 4.5

# Brugerinterface

---

**A [1-3]** Sætter en af de 3 brugerudgange til logisk "1". Kommandokarakteren (Activate output) efterfølges af et tal mellem 1 og 3, der specificerer, hvilken udgang der ønskes aktiveret.

*Eks. A2 sætter udgang 2 til logisk "1".*

**C [1-3]** Sætter en af de 3 brugerudgange til logisk "0". Kommandokarakteren (Clear output) efterfølges af et tal mellem 1 og 3, der specificerer, hvilken udgang der ønskes deaktiveret.

*Eks. C1 sætter udgang 1 til logisk "0".*

**U [1-3]** Den programdel, kommandoen kombineres med, vil blive repeteret, indtil  
**U [A1-A6]** den specificerede indgang bliver lagt til logisk "0" (se elektriske data).  
(Until) Hele programmet kan blive gentaget, eller kun en bestemt del. Brugerindgangene benævnes 1 til 3, og analogindgangene benævnes A1 til A6.

*Eksempel*

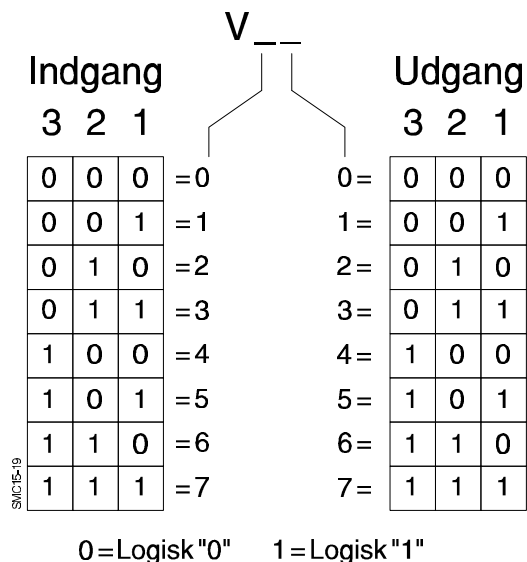
```
.  
.
G+50
A2
*   U3   -   Gentager programdel helt fra starten indtil brugerindgang 3
        bliver lagt til logisk "0".
A1
D25
C1
**  UA1  -   Gentager programmet mellem * og **, indtil analogindgang 1
        bliver påtrykt logisk "0"
```

## 4.5

# Brugerinterface

### V2 (Verify I/O)

Via denne kommando er det muligt at aflæse bruger-indgange og udgange. Kommandoen V2 sendes til controlleren, og svaret vil blive returneret som et V efterfulgt af to tal mellem 0 og 7. Første tal angiver spændingsniveauerne på indgangene. Andet tal angiver spændingsniveauerne på udgangene. Tallene kan oversættes iflg. nedenstående skema:



#### Eksempel:

Modtages V25, betyder det, at indgang 2 er logisk "1", og udgang 1 og 3 er logisk "1".

Kommandoen er udelukkende anvendelig til formål, hvor der er konstant interfaceforbindelse til en computer/terminal. Den kan ikke bruges i selve programmet.

### VA [1-6] (Verify ainput)

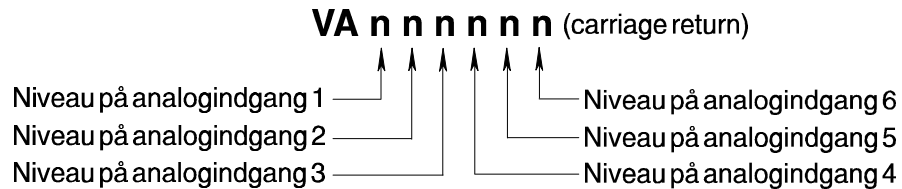
Denne kommando viser spændingen målt på en af de 6 analoge indgange. Der måles med 8bit opløsning, hvilket medfører, at spændingen vil vises i spring på 20mV med 5,10V som det maksimale. Kommandoen kan bruges i standby-mode. Man kan evt. koble de 3 brugerindgange til 3 analogindgange og derved verificere, om spændingerne er som forventet.

## 4.5

# Brugerinterface

---

**VA** Denne kommando viser de digitale niveauer på analogindgangene.  
(Verify ainput) Den returnerede streng har følgende format:



*Eksempel:*

Strengen "VA" sendes til indexeren.  
Følgende streng modtages:

VA101001 (Carriage return)

Dette indikerer at analogindgang 1, 3, og 6 er logisk 1 (>2,5V).  
Analogindgang 2, 4, og 5 er logisk "0" (<2,5V).

**W [1-3]** Stopper programudførelsen, indtil en specificeret indgang bliver påtrykt  
**W [A1-A6]** logisk "1". Efterfølges kommandokarakteren af et tal mellem 1 og 3, henvises til  
(Wait for) en af brugerindgangene, og efterfølges kommandokarakteren af A1 til A6, henvises til en af analogindgangene.

*Eksempel:*

```
.  
A3  
G+372  
W1 - Venter her, indtil brugerindgang 1 bliver påtrykt logisk "1".  
G+46  
C1  
D20  
WA5 - Venter her, indtil analogindgang 5 bliver påtrykt logisk "1".  
.  
.
```



## 4.6

## Flowkommandoer

---

**D [1-32000]** Sætter programmet i pausetilstand. Kommandoen skal efterfølges af et tal  
(Delay) mellem 1 og 32000, hvilket angiver pausen i 1/100 sekunder.

*Eksempel:*

*D27* - giver en forsinkelse på 0,27 sekund.

**DA[n].[n1-n2]** Sætter programmet i pausetilstand. Pausens varighed kan via en analog-  
(Analog delay) indgang kontrolleres mellem 0,01 og 320sek.

*Kommandoformat:*

**n:** Specificerer den analogindgang, der ønskes anvendt som måleindgang.

**n1-n2:** Specificerer det ønskede pauseinterval. n1 er den nedre grænse og svarer til den pausetid, der benyttes, når analogindgangen bliver påtrykt 0V. n2 er den øvre grænse og svarer til den pausetid, der benyttes, når analogindgangen er påtrykt 5,10V.

*Eksempel.*

*DA2.10-100* - Muliggør en pause på 0,1-1,0sek kontrolleret af en spænding på 0 til 5,10V tilsluttet analogindgang 2.

**J [n1]** Jump kommandoen medfører et ubetinget spring til det linienummer der speci-  
(Jump) ceres efter kommandoen.

Det specificerede linienummer kan angives i området 0-255.

*Eksempel:*

Linienr.:

0	A1
1	+1000
2	A2
3	G+5
4	C2
5	<b>J2</b>

Jump-kommandoen medfører, at kommandoerne A2 - G+5 - C2 bliver gentaget igen og igen. En afbrydelse kan kun finde sted ved brug af Z eller K kommandoen.

## 4.6

## Flowkommandoer

---

**JC [0-7].[0-255]** I modsætning til *J* (Jump) medfører *JC* (Jump con.) et betinget spring til det linienummer der specificeres efter kommandoen. Linienummeret kan angives i området 0-255. Betingelsen for om springet skal finde sted, afgøres af niveauerne på brugerindgangene. Indgangsniveauerne kan oversættes iflg. dette skema:

*Eksempel:*

Kommandoen *JC5.10* vil medføre et spring til linie 10, hvis brugerindgang 1 og 3 er logisk "1".

**JC\_. [nnn]**

Indgang  
**3 2 1**

0	0	0	=0
0	0	1	=1
0	1	0	=2
0	1	1	=3
1	0	0	=4
1	0	1	=5
1	1	0	=6
1	1	1	=7

0=Logisk "0"  
1=Logisk "1"

*Eksempel:*

Linienr.:

0	S450
1	R200
2	A1
3	+1200
4	<b>JC5.4</b> - Springer til linie 4 hvis brugerindgang 1 og 3 er logisk "1"
5	G+0
6	C1
7	<b>JC3.2</b> - Springer til linie 2, hvis brugerindgang 1 og 2 er logisk "1".

**JC [p].[n1]**  
(Jump con.)

Kommandoen fungerer i princippet som JC[0-7] - betinget af et indgangs-niveau, springes til en specificeret linie.

I modsætning til JC[0-7] foretager JC[p] et betinget spring, hvis et specifikt indgangsniveau opfylder kommandoens betingelse. JC[0-7] foretager dette betingede spring, hvis et logisk mønster på alle 3 indgange er opfyldt.

JC[p] kan både benyttes sammen med brugerindgangene og analogindgangene.

*Kommandoformat:*

**JC [i]=[n].[n1]**

- i:** Specificerer den indgang, der ønskes anvendt som måleindgang .  
Hvis der specificeres 1 til 3, svarer det til brugerindgang 1 til 3. Hvis der specificeres A1 til A6, svarer det til analogindgang 1 til 6.
- n:** Referenceniveauet som det målte indgangsniveau skal sammenlignes med. n kan tillægges 0 eller 1. Når indgangsniveauet på indgang i er lig med referenceniveauet, bliver springet foretaget.

**n1:** Specificerer den linie der skal springes til, hvis betingelsen er opfyldt.

*Program eksempel:*

Linienr.:

0	S750	
1	R700	
2	C3	
3	+500	
4	<b>JC2=0.3</b>	- Springer til linie 3, hvis brugerindgang 2 er logisk "0".
5	G+0	
6	T1000	
7	A2	
8	<b>JCA4=1.5</b>	- Springer til linie 5, hvis analogindgang 4 er logisk "1".

Når analogindgangene, som her, benyttes som logiske indgange, svarer logisk "1" til spændinger større end 2,5V, og logisk "0" svarer til spændinger mindre end 2,5V.

**JCA [p].[n1]**  
(Jump con.)

Kommandoen fungerer i princippet som JC, betinget af et indgangsniveau springes til en specificeret linie.

I modsætning til JC, foretager JCA et betinget spring, hvis en analog indgangsspænding opfylder kommandoens betingelse. JC foretager dette betingede spring hvis et logisk niveau (0 eller 1) er opfyldt.

*Kommandoformat:*

**JCA [a1mn].[n1]**

**a1:** Specificerer den analogindgang, der ønskes anvendt som måleindgang.

**n:** Specificerer referenceværdien som den målte indgangsspænding skal sammenlignes med.

**m :** Tillægges ">", hvis springet skal foretages, når den målte spænding er større end eller lig med referenceværdien.

m tillægges "<", hvis springet skal foretages når den målte spænding er mindre end referenceværdien. n kan sættes i området 0 til 255.

**n1:** Specificerer den linie, der skal springes til, hvis betingelsen er opfyldt.

Da *n* bliver indsat som en værdi i intervallet 0-255, og spændingen, der måles, ligger i området 0-5,10V, skal man ved bestemmelse af *n* konvertere værdien til en spænding eller visa-versa iflg. nedenstående formler:

$$V_{ref} = 0,02 \times n \quad \text{eller} \quad n = 50 \times V_{ref}$$

*Eksempel:*

Man ønsker en referenceværdi på 3,00V.

$$n = 50 \times 3,00 = 150$$

*Programeksempel:*

Linienr.:

0	S450	
1	R600	
2	D2	
3	+342	
4	<b>JCA3&gt;150.2</b>	- Springer til linie 2, hvis analogindgang 3 er større end eller lig med 3,00V.
5	G+0	

**JS [n1]**  
(Jump sub.)

I modsætning til *J* (Jump) medfører *JS* (Jump sub) et ubetinget spring til en subrutine. Det vil sige, at når *JS* bliver eksekveret, gemmer controlleren det næstkommende linienummer. Programmet springer derefter til det linienummer der står specificeret efter *JS*. Når *RET* senere bliver eksekveret returneres til det gemte linienummer, og programafviklingen fortsætter herfra.

*JS* kan benyttes ialt 32 gange svarende til subrutiner i 32 lag.

**L [0-255]**  
(Loop)

Ønskes en programdel gentaget flere gange, anvendes denne kommando.

Den værdi der efterfølger kommandokarakteren, angiver hvor mange gange programdelen skal gentages. Programdelen kan gentages fra 1 til 255 gange.

Det stykke program der ønskes gentaget, skal afgrænses ved at starte med *L0* og afsluttes med f.eks. *L5*.

*Eksempel*

```
L0
.
.
.
(Program)
.
.
.
L5
.
```

Programdelen mellem *L0* og *L5* vil da blive repeteret 5 gange. Skrives der ikke *L0* i starten, bliver alt repeteret helt fra starten af programmet.

**Registerbeskrivelse.**

Der findes 510 registre, som frit kan anvendes af brugeren. Disse registre betegnes R1-R510 og kan bruges til mellemresultater m.m. Desuden findes der 7 prædefinerede registre, som kun kan anvendes til specifikke formål. Eksempelvis register T som anvendes til at bestemme topfarten med. Via disse registre er der mulighed for at ændre parametre løbende igennem et program-forløb. Det er f.eks. muligt at ændre længde, hastighed, acceleration, pausetid, delprogram gentagelser m.m.

Registrene kan desuden gemmes permanent i EEPROM via I kommandoen. F.eks. behøver parametre, som en gang er indtastet gennem KDM10 (keyboard display modul) ikke at blive indtastet, hver gang styringen startes op.

Følgende registre er til rådighed :

- R :** Antal step, der skal bruges på at acc/dec motoren. Bemærk, dette register har ikke noget at gøre med register R1 til R510.
- S :** Starthastighed.
- T :** Tophastighed.
- D :** Pause i program.
- L :** Delprogram tæller (Loopcounter).
- n :** Positions tæller i step.
- A1-A6 :** Analog indgangenes værdi går fra 0-255 svarende til spændinger fra 0-5.10V.
- R1-R510 :** 510 brugerregistre til mellemresultater m.m.

*Eksempel 1:*

R2 = 3000 ; Sætter register R2 til værdien 3000  
 T = R2 + 100 ; Sætter tophastigheden til 3100 step/s ved næste kørselsforløb

*Eksempel 2:*

R34 = 400  
 D = R34 + A1 ; venter (400 + værdi fra AD konvertering) x 10mS

*Eksempel 3:*

R1 = n + R2 ; sætter R1 til værdien af positionstælleren i step + R2

*Eksempel 4:*

R1 = 350 + 700 ; R1 tildeles værdien 1050  
 +(R1) ; Der køres 1050 step med uret

*Eksempel 5:*

R30 = 100 ; R30 tildeles værdien 100  
 R31 = 200 ; R31 tildeles værdien 200  
 R34 = R30 + R31; R34 tildeles værdien af R30+R31  
 G+(R34) ; Kør til position R34

**Register beskrivelse.**

Der er desuden mulighed for at opbygge arrays, så et register kan pege på andre registre og derved flytte registerindhold. Dette gør at man f.eks hurtigt kan kopiere en blok af registre til en anden blok.

*Eksempel 6:* Der ønskes kopieret ARRAY1 (register 100-199) til ARRAY2 (register 300-399)

```
R1=100      ; Opret "pegepind1" til array1
R2=300      ; Opret "pegepind2" til array2
L0          ; Der er 100 registre, der skal kopieres
R(R2)=R(R1) ; Kopier et register ad gangen
R1=R1+1    ; Peg på næste register i array1
R2=R2+1    ; Peg på næste register i array2
L99        ; Bliv ved, indtil alle 100 registre er kopieret
```

**Regler for registerbrug:**

1: Lighedstegn bruges som tildelings operator. Der må maksimalt anvendes 3 registre eller tal i en ligning.

*Eksempel:*

R3=R23+T er ok, mens R3=R23+T-100 ikke er tilladt.

2: Følgende 4 regningsarter kan anvendes:

- + : Addition
- : Subtraktion
- \* : Multiplikation
- / : Division

3: Alle tal og registerindhold skal være heltal i området 0-65535. Dog må værdierne ved division maksimalt være 32767 for at få et korrekt resultat.

4: Ved division rundes altid nedad.

*Eksempel:*

R2=289/10 ; Udregner divisionen 289/10.

PRINT(5.R2) ; Resultatet blev rundet ned, der vil blive udskrevet tallet 28.

5: R, S, T, L og D registre kan tildeles værdier via ligheds tegn, men en hurtigere måde, både programopbygnings- og udførsels-mæssigt, vil være at tildele værdier direkte uden lighedstegn. Hvis man kender værdien på forhånd som en konstant værdi, kan værdien indsættes direkte i registret.

*Eksempel:*

T2000 vil svare til T=2000

D200 vil svare til D=200

S500 vil svare til S=500

6: Ved "if" sammenligninger må tallene der sammenlignes højst være 32767. Det gælder både for registerindhold og tal.



**VR[0-510]** Indholdet af brugerregistrene kan verificeres ved hjælp af verify kommandoen.  
(Verify)

*Eksempel:*

**VR9** ; Udskriver indholdet af register R9

**I[5-7]** Kommandoen bruges til at gemme alle brugerregistre i EEPROM eller hente registre  
(Initialize) fra EEPROM. Desuden kan man nulstille alle registre på en gang.  
Kommandoen kan bruges både i standbymode og programmeringsmode.

*Eksempel:*

- 15** Nulstiller alle 510 brugerregistre.  
Bemærk at brugerregistrene i EEPROM'en bliver ikke påvirket af denne kommando.
- 16** Gemmer alle 510 registre i EEPROM. De prædefinerede registre (R,S,T m.m.) bliver ikke gemt i EEPROM
- 17** Henter alle 510 registre fra EEPROM.

**con=n**

Hvis man ønsker et omsætningsforhold mellem step og længde/position bruges denne kommando. n angiver, hvor mange step der går på en længde eller positions-enhed. Enheden kunne være mm, cm, ml, dl osv. Omsætningsfaktoren kan specificeres som decimaltal med op til 4 decimaler fra 0,0001 til 1600,0000.

Kommandoen indsættes i programmet i starten og vil virke, indtil der evt. under programafviklingen specificeres en ny omsætningsfaktor. Når motoren skal til at køre, udregnes det antal enheder der er specificeret, gange omsætningsfaktoren, og det antal step vil motoren køre.

Hvis man f.eks. sætter omsætningsfaktoren til 2.3456 med *con=2.3456* kommando, og der skal køres feks 450 ml, vil  $450 * 2,3456$  give 1055.52 step. Motoren vil derfor køre 1055 step og gemme resten på 0,52 step. Ved næste kørselsforløb, anvendes denne rest, så der korrigeres for det ikke kørte 0,52. step.

*Eksempel:*

Man har et system, der kræver, at der køres 14.654 step for at få 1 ml doseret.

```
con=14.654      ; Omsætningsfaktor 14.654 step per milliliter
R1=290         ; Tildel 290 til register R1.
+(R1)         ; Kør 290 ml. Antal kørte step er  $290 * 14.654=4249$ .
               ; Reststep er 0.66
D100          ; Hold pause i 1 sek.
+18           ; Kør 18 ml. Antal step der køres er  $18*14.654+reststep=264$ .
               ; Den nye reststep er 0.432
```

Efter "Home" søgning med H kommando vil rest i step blive sat til 0, da motoren befinder sig på sit absolutte nulpunkt.

**PRINTn1.n2.n3** Print kommandoen kan anvendes til at udskrive indholdet af registre til eksterne moduler. På nuværende tidspunkt er det muligt at udskrive til 5 eksterne moduler, nemlig til en PC'er el.lign. gennem RS232 og JVL moduler type DIS10, KDM10, MCM10, og IOM10 moduler gennem RS485.

Kommandoformat :

**n1** : Specificerer adressen på modulet, der skal skrives til (1-31).  
Hvis RS232 interfacet benyttes, specificeres værdien 255.

**n2** : Specificerer, hvilket register eller cursorposition i modulet der skal skrives til.

**n3** : Specificerer, hvilket register, talværdi eller tekststreng i stepmotor controlleren der skal udskrives.

*Eksempel 1:*

PRINT1.0.R23

Udskriver indholdet af register R23 til det modul, der har adressen 1. Da der sendes via RS485 balanceret, vil det være muligt at placere modulerne op til 500 meter fra controlleren.

*Eksempel 2:*

PRINT255.0.R2

Udskriver indholdet af register R2 til PC'er gennem RS232 interface. Kommandoen kan bruges til at udskrive indholdet af registre under programafviklingen. Den er specielt velegnet under fejlfinding i program. Hvis man anvender JVL's terminalprogram "*Editor2*", kan man, efter at programmet er overført med F5, skifte til kommunikationsvinduet ved at trykke F6. Her vil register indholdet blive vist, når PRINT linien nås. Adressen 255 er fast defineret til at være PC'er.

*Eksempel 3 :*

PRINT3.41."Indtast Antal: "

I forbindelse med KDM10 (keyboard-display modul) er det ofte ønskeligt at udskrive information til brugeren. Ovenstående eksempel viser, hvordan man skriver tekst på LCD displayet. Adressen på modulet, 3. 41, står for cursorposition 41, som er 1 tegn på linie 2.

*Eksempel 4 :*

```
R1=5555           ; Tildel register R1 værdien 5555
R30=333          ; Tildel register R30 værdien 333
PRINT5.41.R1     ; Udskriv til KDM10 på cursorposition 41 indholdet af R1
PRINT2.0.R30     ; Udskriv til DIS10 indhold af register R30.
```

Når man anvender DIS10 eller KDM10, skal man ofte udskrive registerindhold på de 2 modulers respektive displays. Dette foregår nemmest som ovenstående med PRINT kommandoen, hvor et register udskrives til en cursorposition eller som på DIS10 direkte på LED displayet. Adresserne 5 og 2 er de respektive adresser sat på modulernes dipswitch.

## 4.7

## Udvidet kommandosæt

---

**IF [p1 m p2]** IF kommandoen bruges til at sammenligne 2 talværdier, p1 og p2. Disse kan være indholdet af et register eksempelvis R1,A1,T m.m. eller blot et heltal som 10500. Alle registre, beskrevet under "Register beskrivelse", kan bruges i IF sætningen. Sammenligningsoperatoren m kan være en af følgende :

m	Betingelse opfyldt hvis
<	Mindre end
>	Større end
=	Lig med
<=	Mindre end eller lig med
>=	Større end eller lig med
<>	Forskellig fra

Hvis betingelsen er opfyldt, hoppes til næste linie, og programlinierne udføres derfra. Hvis betingelsen ikke er opfyldt, springes næste linie over, og programmet udføres derfra.

*Eksempel 1:*

```
:START IF R10 < 9800 ; Hvis indholdet af register R10 er mindre
      J:PROG1 ; end 9800 hop til label PROG1.
      J:PROG2 ; ellers hop til label PROG2.
```

*Eksempel 2:*

```
T=100 ; Sæt Topfarten til 100 step/sek
:START T=T+50 ; Tæl topfarten op med 50
      IF T>4000 ; Hvis topfarten bliver større end 4000step/s
      J:FARTOK ; Hop til label FARTOK
      +1000 ; Kør 1000 step i urets retning
      J:START ; Hop til linie 11, hvor farten bliver sat op
```

Ovenstående program får motoren til at køre 1000 step med hastigheden 150 step/sek og vil øge tophastighed med 50 step/sek, indtil tophastighed på 4000 step/sek opnåes.

**INPUTn1.n2** INPUT kommandoen bruges til at hente data fra de eksterne moduler, som er koblet på RS485 bussen. Af eksterne moduler kan nævnes Keyboard, Display, thumbwheel, BCD cifre fra PLC, printer, ekstra indgange, digital til analog moduler m.m.

Alle modulerne er intelligente og vil derfor indeholde registre. Disse registres indhold kan via INPUT kommandoen overføres til stepmotor controllerens registre. Registrenes størrelse og antal kan variere, men alle moduler har mindst et register.

*Kommando format :*

**n1:** Specificerer adressen på det modul, der ønskes input fra. Tallet skal være mellem 0-255. På RS485 bussen er der mulighed for at have op til 32 moduler tilkoblet samtidigt. De enkelte moduler skal, via dipswitch på modulet, sættes til den adresse som er specificeret i n1.

**n2:** Specificerer, hvilket register i det eksterne modul stepmotor controlleren ønsker at få overført. n2 skal være mellem 0-255.

*Eksempel 1:*

IOM10 modulet med 16 indgange og 8 udgange benyttes. Modulet har adresse 5. Der ønskes at læse alle 16 indgange og teste, om den binære værdi er 255. Hvis det er tilfældet, skal tælleren læses, og programmet fortsættes.

I IOM10's manual finder man, at tæller's registre hedder 3, og registret for alle 16 indgange er 2.

```
:LÆSINDG      R10=INPUT5.2      ; Læs alle 16 indgange og overfør til R10.
               IF R10<>255      ; Hvis indgange forskellig fra 255 læs igen
               J:LÆSINDG
               J:LÆS_TÆLLER; Ellers læs tællerværdi og fortsæt program

:LÆS_TÆLLER   R30=INPUT5.3      ; Læs tællerværdi og overfør til R30
               R(R1)=R30        ; Overfør tællerværdi til array med R1 som
                               ; pegepind.
```

**AO[a].[o]** Kommandoen aktiverer et flag på modulet med adressen a.  
(Activate) Flaget har nummer det nummer der specificeres af "o". Flaget kan f.eks. være en udgang som på IOM10 modulet. Det vil medføre, at når et flag på IOM10 modulet bliver aktiveret, vil en udgang blive aktiveret. Hvis et flag i et andet modul bliver aktiveret, kan det betyde noget helt andet. F.eks. hvis man aktiverer flag 3 i KDM10, vil det medføre, at cursoren på LCD displayet begynder at blinke. D.v.s at flag med samme nummer på de forskellige moduler ikke har samme funktion. For at se funktionen af flagene, henvises til de respektive modulers brugermanual.

**Format: AO{1<=a<=31}.{1<=o<=255}**

*Eksempel 1:*

Modulet KDM10 (Keyboard display modul) har adressen 4. Displayet ønskes slettet, så ny tekst kan udskrives. Nedenstående vil slette displayet og placere cursoren i øverste venstre hjørne.

AO4.1 ; Sletter displayet

*Eksempel 2:*

IOM10 modulet og SMC30B sidder koblet sammen. IOM10 har adressen 10. Udgang 4 ønskes aktiveret.

AO10.4

**CO[a].[o]** I de respektive moduler er der nogle flag, som kan slettes med denne kommando.  
(Clear) Antallet af flag i de forskellige moduler er forskelligt, men alle moduler har mindst et flag. I KDM10 (keyboard display modul) kan kommandoen f.eks. bruges til at slette LCD displayet. I IOM10 (I/O modul) kan kommandoen bruges til at deaktivere en af udgangene m.m.

**Format:** CO {1<=a<=31}.{1<=o<=255}

*Eksempel 1:*

SMC30 og KDM10 er forbundet sammen via RS485 bussen. SMC30 har adressen 1 og KDM10 har adressen 3. (Adressen bliver sat på de respektive dipswitches). Cursoren på LCD displayet skal slukke. Hvis cursor er tændt under tekst udskrivning med PRINT kommando, kan displayet flimre. Dette undgås ved at slette cursoren.

CO3.3 ; Sletter cursor

*Eksempel 2:*

SMC30C og IOM10 er forbundet sammen via RS485 bus. IOM10's adresse er 5. Man ønsker at deaktivere udgang 7 på IOM10.

CO5.7 ; Deaktiverer udgang 7 på IOM10 med adresse 5.



## 5.1

## Elektriske data

	Min.	Typ.	Max.	Enhed
<b>Strømforsyning :</b>				
Forsyningsspænding	207	230	242	VAC
Effektforbrug (uden motor)		7		W
Brugerforsyning udgang				
Udgangsspænding (justerbar)	5,1		30,2	VDC
Belastningsstrøm			500	mADC
<b>Motordriver :</b>				
Udgangsstrøm (pr. fase)	0,0		6/(12)	ADC
Udgangsspænding (uden motor)		150		VDC
Chopperfrekvens		22		kHz
<b>Interface :</b>				
Rx mark position	-1		-12 V	
Rx space position	2.5		12	V
Tx mark position	-3		-12	V
Tx space position	5		12	V
Kommunikationshastighed	110		9600	Baud
Isolationspænding*			500	V (Max.)
<b>Modulinterface :</b>				
Kommunikationslængde 0		500	m	
Kommunikationshastighed		50		kbit/sek
Isolationspænding		* 1000V	(Max.)	
<b>Brugerindgange + stopindgang :</b>				
Indgangsimpedans	10			kohm
Forsyning :				
spænding	5		32	VDC
strøm v/5V	-		10	mA DC
v/12V	-		20	-
v/30V	-		45	-
Logisk "0"				
v/5V	<1,7		-	VDC
v/12V	<3,0		-	-
v/30V	<6,7		-	-
Logisk "1"				
v/5V	-		>2,7	-
v/12V	-		>6,5	-
v/30V	-		>16,2	-

\* Målt fra strømforsyningsstel til interfacestel

() Værdier gældende for SMC30C.

(Fortsættes næste side)

## 5.1

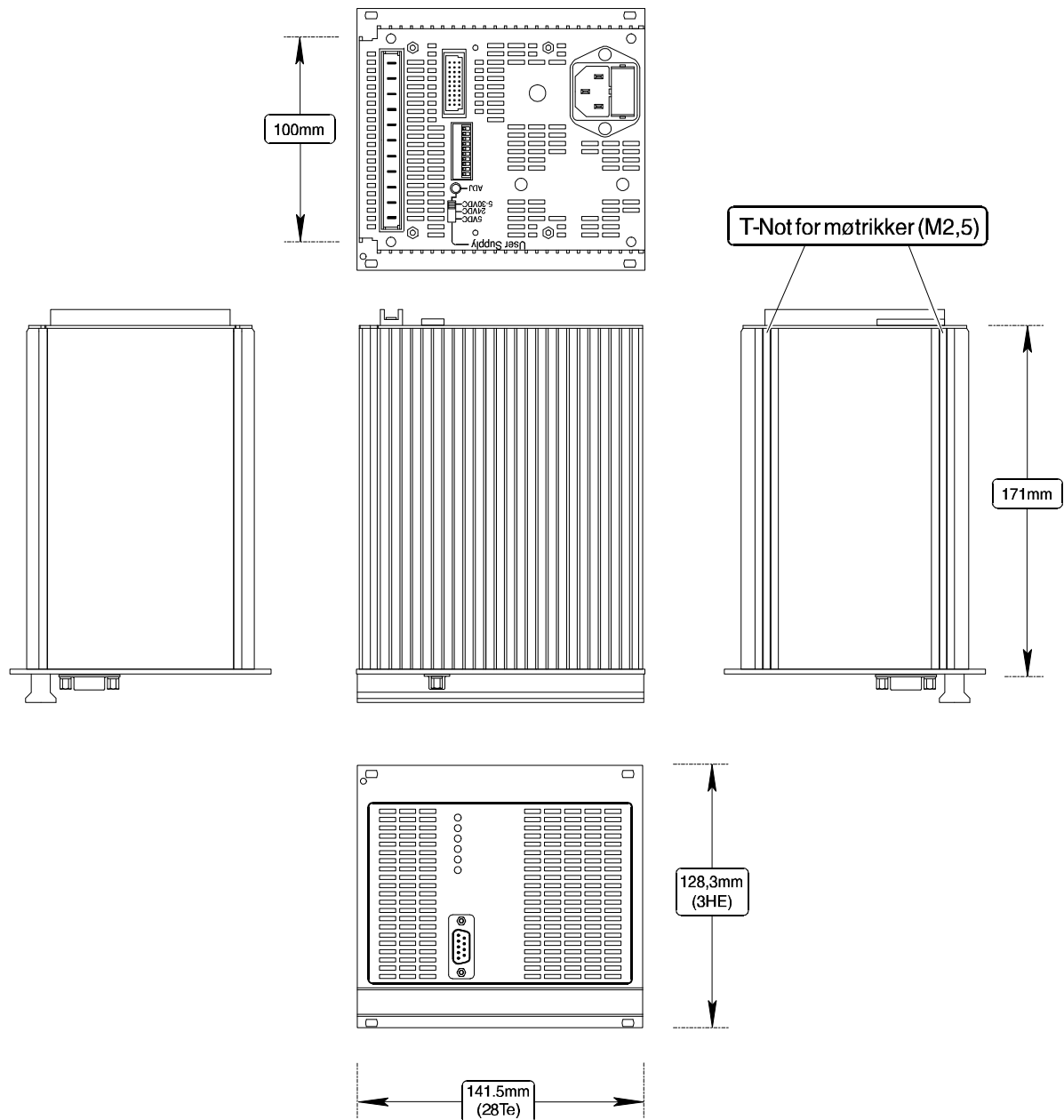
## Elektriske data

	Min.	Typ.	Max.	Enhed
<b>Analogindgange :</b>				
Opløsning	-		8	Bit
Indgangsspænding (Max tilladt)	-20		** 45V	DC
Indgangsspænding (Nominel)	0.00		5,10	V DC
Offset error	-	±½	±1	LSB
Gain error	-	±½	±1	LSB
Temperaturdrift @ 0-50 °C	-	±¼	±½	LSB
Logisk "0"	<2,5V		-	DC
Logisk "1"	-		>2,5	-
<b>Endestopindgange :</b>				
Indgangsimpedans	2,4		3,3	kOhm
Logisk "0" (inaktiv)	(-30)		3,0	V DC
Logisk "1" (aktiv)	3,8		30	V DC
<b>Brugerudgange og steppuls-udgang:</b>				
Spændingsforsyning	5		30	V DC
Belastningsstrøm pr. udgang				
1 udgang aktiveret @25 °C			700	mA DC
2 udgange aktiveret @25 °C			460	mA DC
3 udgange aktiveret @25 °C			300	mA DC
<b>Diverse :</b>				
Arbejdstemperatur (SMC30B)	0		45	°C
Arbejdstemperatur (SMC30C)	0		40	°C

\*\* Absolut max. tid < 1 sek.

## 5.2

## Fysiske mål



Såfremt kontrolleren monteres i et lukket chassis, bør der monteres en blæser eller lignende køleforanstaltning. Controlleren er dog sikret imod overophedning, idet en indbygget termosikring afbryder drivertrinnene ved ca. 80°C.

## 5.3

## Pladsudnyttelse i hukommelse

---

Arkiv-hukommelsen i controlleren udgøres af en 8192 byte E<sup>2</sup>PROM.

Der bruges 1192 byte til at korigere A/D konverterens offset-fejl, og justering af termometer-funktionen (TP) samt de 500 regneregistre. Dermed er der 7000 byte tilbage til programlager.

For at kunne udnytte de 7000 byte optimalt, fremgår det af nedenstående tabel, hvor meget plads de enkelte kommandoer optager. Det samlede program må ikke fylde mere end de 7000 byte.

Forsøges et program større end 7000 bytes indlæst, vil controlleren melde "E3".

<b>1 byte :</b>	g± H± RET (Blank linie)	<b>6 byte :</b>	DA[n].[n1-n2] ±A[n].[n1-n2] G±A[n].[n1-n2]
<b>2 byte :</b>	A[n] C[n] I[n] r,s,t[n.n] U[n] W[n] L[nnn] J[nnn] JS[nnn]	<b>7 byte :</b>	NA[p1-p2]
		<b>10 byte :</b>	PRINT(typ.) R[n]=x
		<b>12 byte :</b>	INPUT[n.n.n]
		<b>14 byte :</b>	R[n]=x+x
		<b>17 byte :</b>	IF(max.)
<b>3 byte :</b>	AO[n.n] CO[n.n] CR[nnnn] CS[nnnn] CT[nnnn] R[nnnnn] RS[nnnn] RT[nnnnn] S[nnnn] T[nnnn] N[nn.nn] D [nnnnn]		
<b>4 byte :</b>	f [±nnnnnnn] ± [nnnnnnn] G [±nnnnnnn] JC[n.nnn] JCA[ρ].[n1]		
<b>5 byte :</b>	con=[nnnn.nnnn]		

**Jord :**

± Forbindelse til kobberareal på print

**RS232C Interface:**

IG Interface stel  
RX Receive indgang  
TX Transmitter udgang

**ModulInterface:**

A Terminal A  
B Terminal B

**Brugerforsyning:**

U+ + Forsyning ud, 5 til 30V  
U- Forsyningsstel

**Analogindgange:**

AG Analogindgangs-stel  
A1 Analogindgang 1  
A2 Analogindgang 2  
A3 Analogindgang 3  
A4 Analogindgang 4  
A5 Analogindgang 5  
A6 Analogindgang 6

**Brugerindgange:**

I+ Indgangsforsyning +5-30VDC  
CCW Nedre endestopindgang  
CW Øvre endestopindgang  
ST Stopindgang  
I3 Brugerindgang 3  
I2 Brugerindgang 2  
I1 Brugerindgang 1  
EB Ingen anvendelse - til fremtidig brug  
EA Ingen anvendelse - til fremtidig brug  
I- Brugerindgangs stel

**Brugerudgange:**

O- Udgangsforsyning stel  
O1 Brugerudgang 1  
O2 Brugerudgang 2  
O3 Brugerudgang 3  
CK Sløppuls udgang  
O+ Udgangsforsyning +5-30VDC

**Motorudgang:**

A+ Fase 1 + terminal  
A- Fase 1 - terminal  
B+ Fase 2 + terminal  
B- Fase 2 - terminal  
COM Stel - forbindes til skærm på motorkabel

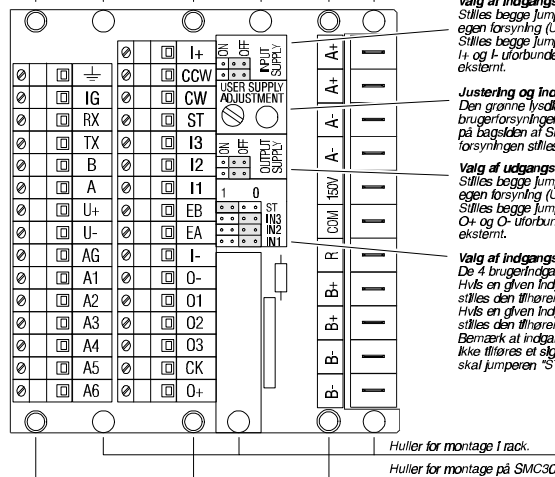
**Ekstern strømforsyning:**

150V Tilslutning fra ekstern strømforsyning +150VDC  
COM Stel

### Forbindelsesskema Connectorboard til SMC30B/C Type "CON30"

Huller for montage i rack.

Huller for montage på SMC30

**Valg af indgangsforsyning :**

Stilles begge Jumpere i position "ON", føres controllerens egen forsyning (U+/U-) til indgangsterminalerne I+ og I-. Stilles begge Jumpere i position "OFF" er indgangsterminalerne I+ og I- uforbunde, og forsyning af indgangskredsløb skal ske eksternt.

**Justering og indikation af brugerforsyning:**

Den grønne lysdiodes intensitet angiver hvilken spænding brugerforsyningen afgiver. Er switchen mærket "User supply" på bagsiden af SMC30 stillet i position "5-30V", kan brugerforsyningen stilles fra 5 til 30VDC via trimmepotentlometret.

**Valg af udgangsforsyning :**

Stilles begge Jumpere i position "ON", føres controllerens egen forsyning (U+/U-) til udgangsterminalerne O+ og O-. Stilles begge Jumpere i position "OFF" er udgangsterminalerne O+ og O- uforbunde, og forsyning af udgangskredsløb skal ske eksternt.

**Valg af indgangstype :**

De 4 brugerindgange IN1, IN2, IN3 og ST har hver en Jumper. Hvis en given indgang får signal fra en PNP (Source) udgang stilles den tilhørende Jumper i position "0". Hvis en given indgang får signal fra en NPN (Sink) udgang stilles den tilhørende Jumper i position "1". Bemærk at indgang "ST" (stopindgang) aktiveres hvis indgangen ikke tilføres et signal der er logisk "1". Hvis indgangen er ubenyttet skal Jumperen "ST" derfor sættes i position "1".

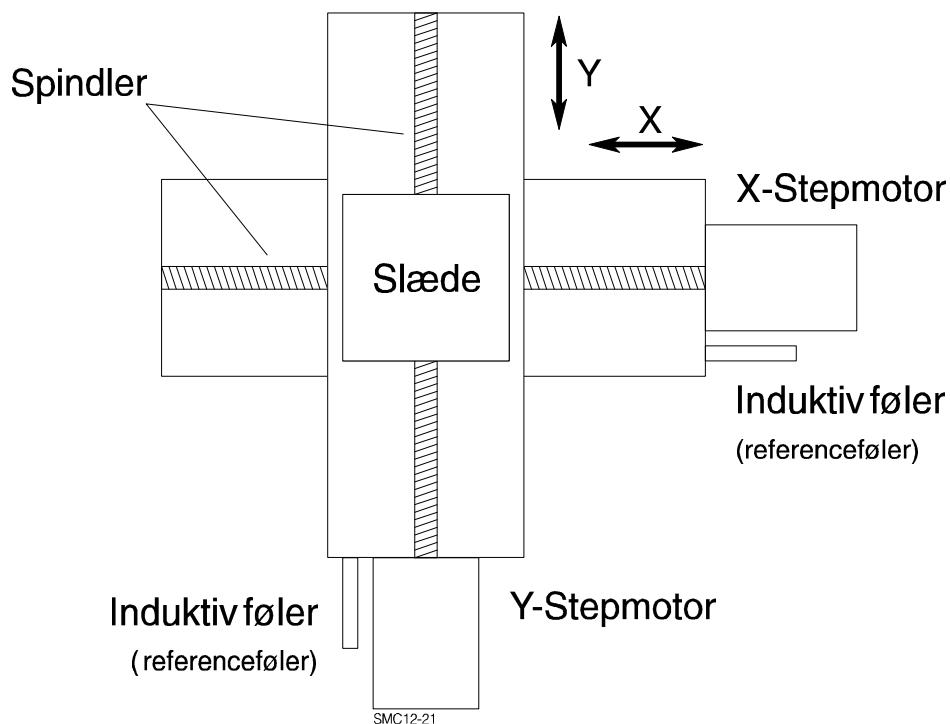
Huller for montage i rack.

Huller for montage på SMC30

Som en ekstra hjælp, vil her blive vist et eksempel på, hvordan 2 Controllere kan bruges til at styre et krydsbord. Problemstillingen er, at der skal bores 2 huller, i en aluminiumsklods. Der anvendes et krydsbord med to påmonterede stepmotorer, (en til hver akse). Endvidere monteres der en induktiv føler til hver akse. Disse induktive følere, skal have til opgave, at registrere, hvornår henholdsvis x og y akse, er i sin mekaniske referenceposition (0 position). De anvendte følere bør have en så lille tasteafstand som muligt (typisk 1-2mm), eftersom tasteafstanden nøje hænger sammen med repetitionsnøjagtigheden.

Dette er vigtigt, idet de induktive følere afgør, hvor referencepunktet, for henholdsvis x og y akse skal ligge, og dermed hele systemets nøjagtighed. Typisk kan en føler med 1mm tasteafstand, klare en repetitionsnøjagtighed på  $\pm 1/100\text{mm}$ , hvilket burde række i langt de fleste tilfælde. Selve boremaskinen vil ikke blive nærmere beskrevet her, men kunne eksempelvis, bestå af en DC-motor, der bliver ført ned af en hydraulik/pneumatik cylinder.

### Krydsbord set fra oven:



## 5.5

# Anvendelseksempel for SMC30

---

### Elektrisk tilslutning.

Før den egentlige programopbygning påbegyndes er det en god ide, at gøre sig klart hvilke indgangs/udgangssignaler, systemet skal håndtere.

For at kunne kontrollere systemet, er der forbundet en startkontakt til Y-controlleren.

Følgende skal håndteres:

- 1) 2 Stepmotorer.
- 2) 2 Induktive følere med NPN udgang (0V på udgangen når de er aktiveret).
- 3) En boreenhed. Denne startes ved at sende en spændingsimpuls på 0,1 sek., og afgiver 12V når den er færdig med at bore, og gået tilbage til sin udgangsposition.
- 4) Startkontakt.
- 5) Interne synkroniseringssignaler mellem de to controllerer.  
(Se programbeskrivelse, næste side)

På næste side er skitseret, hvordan det samlede system er elektrisk forbundet.

Bruger ud-/indgange er anvendt på følgende måde:

### X-controller.

#### Indgang 1

Bruges til at modtage klarmeddelelse fra Y-controller.

#### Indgang 2

Bruges til at modtage klarmeddelelse fra boreenhed.

#### Indgang 3

Induktiv føler (X-akse)

#### Udgang 1

Sender klarmeddelelse til Y-controller.

#### Udgang 2

Sender startimpuls til boreenhed.

#### Udgang 3

Bruges ikke

### Y-controller.

#### Indgang 1

Bruges til at modtage klarmeddelelse fra X-controller.

#### Indgang 2

Bruges til startkontakt

#### Indgang 3

Induktiv føler (Y-akse).

#### Udgang 1

Sender klarmeddelelse til X-controller.

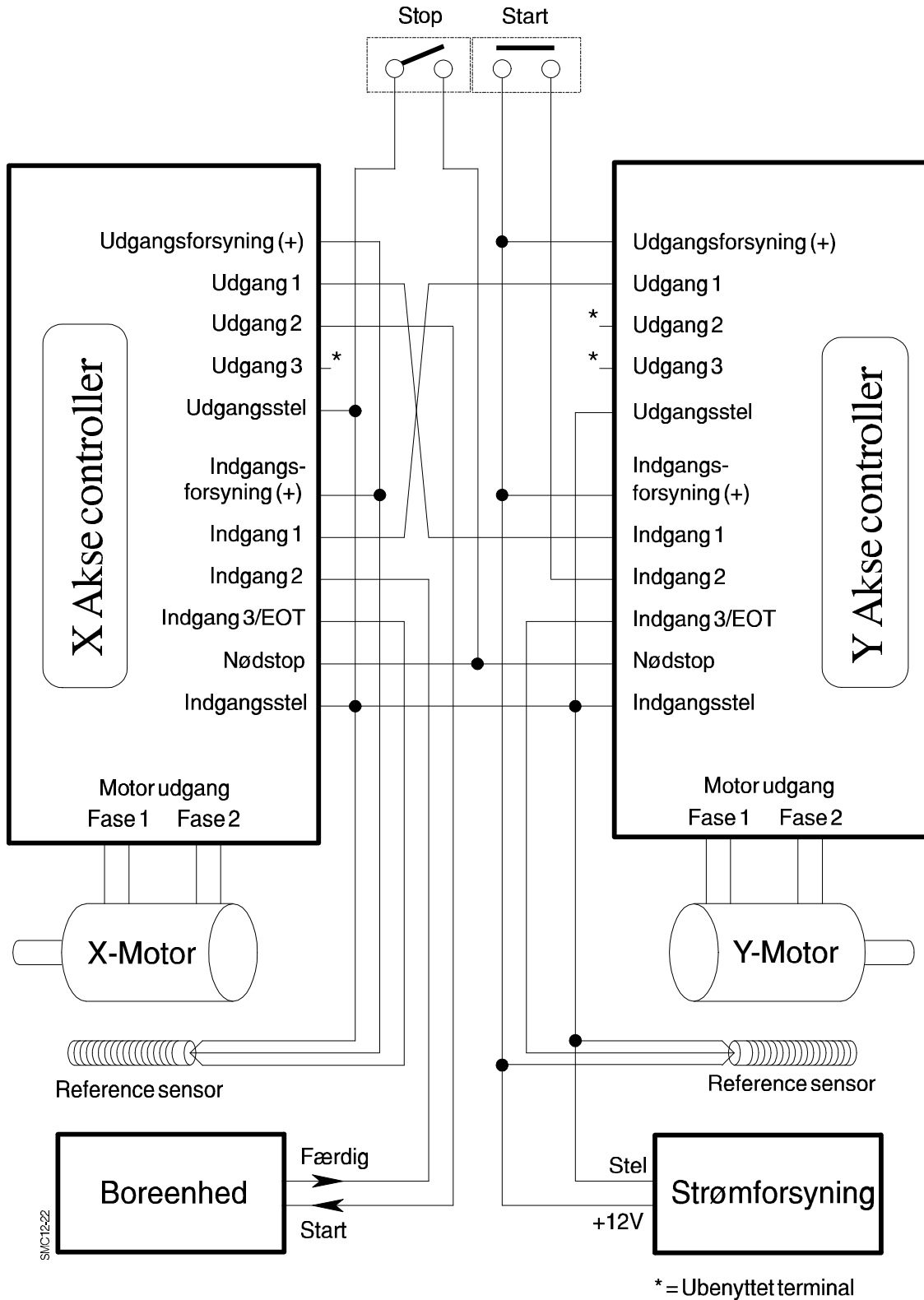
#### Udgang 2

Bruges ikke

#### Udgang 3

Bruges ikke

Vedr. placeringen af de enkelte terminaler, henvises til afsnit 2.6.



SMC1222



**Program til styring af krydsbord.**

Før programopbygningen påbegyndes, bør alle involverede parametre kendes.

Det forudsættes, at der ønskes to huller boret på følgende korodinatorer:

*1 Hul - 4013,7387*

*2 Hul - 5164,1949*

Koordinaterne er angivet i step, med reference til de induktive føleres aktiveringspunkt.

Boreenheden skal have en spændingspuls i 0,1sek, for at aktiveres. Den borer hullet, og sørger selv for at returnerer til sin udgangsposition.

Herefter afgiver den en konstant spænding på 12V, hvilket indikerer, at den er færdig med at bore.

De handlinger der skal foregå kan beskrives på følgende måde:

1. Hvis startkontakten er aktiveret fortsættes til pkt.2
2. Gå til position 4013,7387
3. Bor hul
4. Gå til position 5164,1949
5. Bor hul
6. Slut, Gå til pkt.1

Følges denne procedure, vil systemmet vente indtil startkontakten aktiveres, bore de to huller, og vente på en ny aktivering af startkontakten.

Inden programmerne til de to controllere laves, kan det være en god ide at lave et flowchart, eftersom det giver et bedre overblik. Det resterende arbejde, vil stort set kun være, at finde de tilhørende kommandoer.

Se flowchartet på næste side.

## 5.5

## Anvendelseksempel for SMC30

---

### Program til styring af krydsbord.

#### X-Controller.

Start:

- 1) Fortsæt når Y-controller har givet klarmelding.
- 2) -
- 3) Kør til position 4013.

#### Y-Controller.

Start:

- 1) Hvis startkontakt er aktiveret, fortsættes til pkt.2.
- 2) Giv klarmelding til X-controller.
- 3) Kør til position 7387.

X-controlleren skal køre kortest, og vil derfor være først færdig. Den vil som følge af dette, skulle vente på, at Y-controlleren bliver færdig.

- 4) Fortsæt når Y controller har givet klarmelding.
- 5) Send startsignal til boreenhed.
- 6) Fortsæt når boreenhed er færdig.
- 7) Giv klarmelding til Y-controller.
- 8) Kør til position 5164.

- 4) Giv klarmelding til X-controller.
- 5) Fortsæt når X-controller har givet klarmelding.
- 6) -
- 7) -
- 8) Kør til position 1949.

Y-controlleren skal køre kortest, og vil derfor være først færdig. X-controlleren skal både køre til positionen, og sørger for at hullet bliver boret. Y-controlleren må, som følge af dette, skulle vente på at X-controlleren bliver færdig.

- 9) Send startsignal til boreenhed.
- 10) Fortsæt når boreenhed er færdig.
- 11) Giv klarmelding til Y-controller.
- 12) Gå til pkt.1.

- 9) Fortsæt når X-controlleren har givet klarmelding.
- 10) -
- 11) -
- 12) Gå til pkt.1

#### Bemærkning:

De pladser der er markeret med en streg, er når en af controllerne venter på, at den anden controller bliver færdig, og giver klarmelding.

## 5.5

## Anvendelseksempel for SMC30

---

### Program til styring af krydsbord.

Nu er der dannet grundlag for det egentlige program, som kan laves på følgende måde. Tallene i venstre side af de to kolloner, refererer til flowchartet på forrige side. Det er ikke alle flowchartets punkter der kan oversættes direkte til en kommando.

Hver gang X-controlleren giver klarmelding til Y-controlleren, eller visa-versa, er man nødsaget til at bruge 3 kommandoer.

Hver gang en klarmeddelelse bliver overført mellem de to controllere, er denne udformet som en spændingsimpuls. Impulsens længde er sat til 1 sek, idet modtageren af klarmeddelelsen (X eller Y-controlleren), skal kunne nå at opfange denne. Impulsen bliver udformet ved, at senderen aktiverer sin udgang. Dernæst holdes der en pause på 1 sek., hvorefter udgangen deaktiveres. Denne manøvre involverer 3 kommandoer: A1 - D10 - C1.

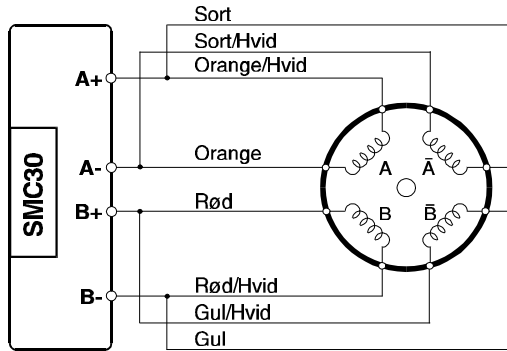
#### X-Controller.

- 1) H- (Nulstil)
- 2) W1 (Vent på indgang 1)
- 
- 
- 
- 4) G+4013 (kør til pos.+4013)
- 5) A1 (Aktiver udgang 1)
- 
- 
- 6) A2 (Aktiver udgang 2)
- D1 (Vent 0,1 sek)
- C2 (Deaktiver udgang 2)
- 7) W2 (Vent på indgang 2)
- 8) A1 (Aktiver udgang 1)
- D10 (Vent 1 sek)
- C1 (Deaktiver udgang 1)
- 9) G+5164 (kør til pos.+5164)
- 10)A2 (Aktiver udgang 2)
- D1 (Vent 0,1 sek)
- C2 (Deaktiver udgang 2)
- 11)W2 (Vent på indgang 2)
- 12)A1 (Aktiver udgang 1)
- D10 (Vent 1 sek)
- C1 (Deaktiver udgang 1)
- 13)J1 (Hop til pkt.2)

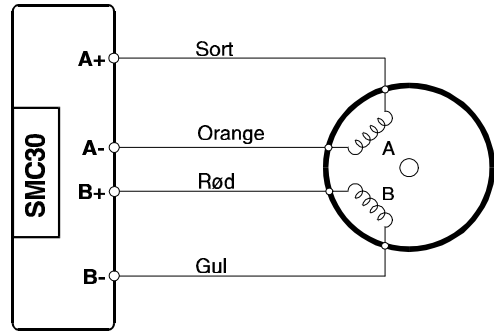
#### Y-Controller.

- 1) H- (Nulstil)
- 2) W2 (Vent på indgang 2)
- 3) A1 (Aktiver udgang 1)
- D10 (Vent 1sek)
- C1 (Deaktiver udgang 1)
- 4) G+7387 (Kør til pos.+7387)
- 5) A1 (Aktiver udgang 1)
- D10 (Vent 1 sek)
- C1 (Deaktiver udgang 1)
- 6) W1 (Vent på indgang 1)
- 
- 
- 
- 
- 9) G+1949 (Kør til pos.+1949)
- 10)W1 (Vent på indgang 1)
- 
- 
- 
- 
- 13)J1 (Hop til pkt.2)

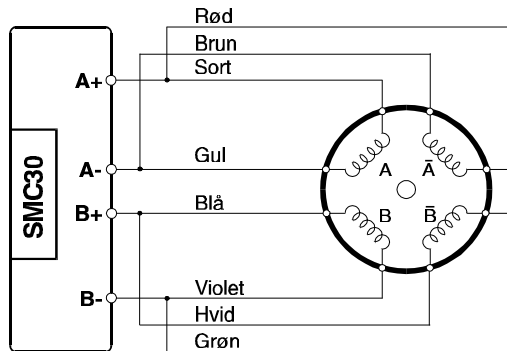
**Forbindelse af MAE motor  
Type HY200-xxxx-xxx-x8**



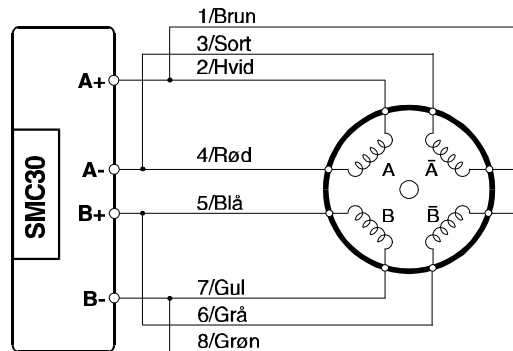
**Forbindelse af MAE motor  
Type HY200-xxxx-xxx-x4**



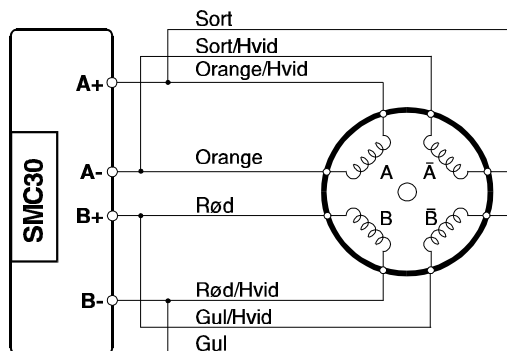
**Forbindelse af Phytron motor  
Type ZSx . xxx.x,x**



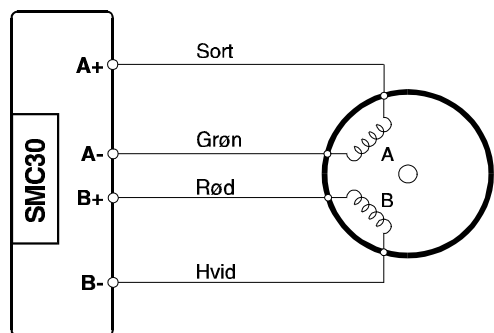
**Forbindelse af Zebotronics motor  
Type: SMxx.x.xx.x**



**Forbindelse af Wexta stepmotor  
Type: PH2xx-xxx**



**Forbindelse af Teco stepmotor  
Type: 4Hxxxx**



- +/- kommando**, 36
- ±A kommando**, 36
- A kommando**, 46
- Addition, 56
- Adressering, 22
- Analogindgange**, 16
- AO**, 63
- Arkiv*, 34
- Brugerindgange, 14
- Brugerinterface, 46; 48
- Brugerudgange**, 12; 33
- byte, 68
- C kommando**, 46
- Checksum, 25
- CO**, 64
- con=n**, 58
- controlleren, 75
- CR kommando**, 37
- CS kommando**, 37
- CT kommando**, 37
- D kommando**, 49
- DA kommando**, 49
- Division, 56
- E kommando**, 33
- E<sup>2</sup>PROM, 68
- Elektriske data, 65; 66
- Endestopindgange, 17; 18
- Error, 29
- F kommando**, 33
- f+/- kommando**, 33
- Flowkommandoer, 49; 50; 51; 52; 53
- g+/-**, 38
- G±A**, 38
- H+/-**, 38
- hukommelsen, 68
- I [1-3] kommando**, 33
- I[5-7]**, 57
- IF**, 61
- Indgangshysterese**, 15
- INPUT**, 62
- Interfaceforbindelser**, 20; 21
- J kommando**, 49
- JC kommando**, 50; 51
- JCA kommando**, 52
- JS kommando**, 53
- K kommando**, 33
- Kommandoformat**, 24
- Kommunikationshastighed, 23
- L kommando**, 53
- Lagre, 27
- M kommando**, 34
- Modes, 28
- moduler**, 26
- Motorkommandoer, 36; 37; 38; 40; 41; 42; 43; 44; 45
- Multiplikation, 56
- N kommando**, 39; 40
- NA kommando**, 41; 42
- PE kommando**, 34
- PO kommando**, 34
- Positionstæller, 27; 33
- PRINT**, 59
- PX kommando**, 34
- R kommando**, 43; 44; 45
- RS kommando**, 43
- RS485, 26
- RT kommando, 43
- S kommando**, 43; 44; 45
- Sikring, 7; 8
- Stop, 33
- Stopindgang**, 19
- Subtraktion, 56
- Systemkommandoer, 33; 34; 35
- T kommando**, 43; 44; 45
- Temperatur, 35
- TP kommando**, 35
- U kommando**, 46
- V1 kommando, 35
- V2 kommando**, 47
- VA kommando**, 47; 48
- VR kommando**, 45
- VR[0-510]**, 57
- VS kommando**, 45
- VT kommando**, 45
- W kommando**, 48
- X kommando**, 35
- Z kommando**, 35