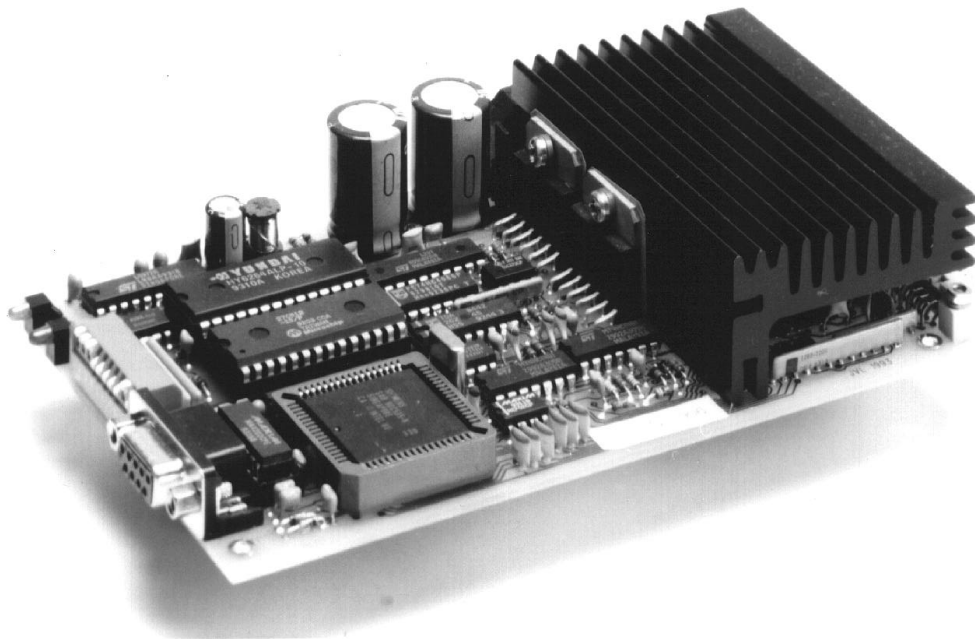


SMC20

Step Motor Controller

Bruger Manual



JVL Industri Elektronik A/S - August 1993

Indhold

1.1	INDLEDNING.....	2
1.2	TILSLUTNINGSMULIGHEDER.....	3
HARDWARE		
2.1	STRØMFORSYNING.....	5
2.2	TILSLUTNING AF MOTOR.....	7
2.3	BRUGERUDGANGE.....	10
2.4	BRUGERINDGANGE.....	11
2.5	ANALOGINDGANGE.....	12
2.6	STOPINDGANG.....	13
2.7	STATUSUDGANGE.....	14
2.8	STIKFORBINDELSER.....	15
INTERFACE		
3.1	INTERFACEFORBINDELSER.....	16
3.2	ADRESSERING.....	18
3.3	KOMMUNIKATIONSHASTIGHED.....	19
3.4	KOMMANDOFORMAT.....	20
3.5	CHECKSUM.....	21
SOFTWARE		
4.1	GENERELT OM SOFTWARE.....	22
4.2	KOMMANDOOVERSIGT.....	25
4.3	SYSTEMKOMMANDOER.....	28
4.4	MOTORKOMMANDOER.....	31
4.5	BRUGERINTERFACE.....	41
4.6	FLOWKOMMANDOER.....	44
APPENDIX		
5.1	ELEKTRISKE DATA.....	49
5.2	FYSISKE MÅL.....	50
5.3	PLADSUDNYTTELSE I HUKOMMELSE.....	51
5.4	TILBEHØR.....	52
5.5	ANVENDELSESEKSEMPEL FOR SMC20.....	53
5.6	MOTOR FORBINDELSER.....	59
5.7	STIKORDSREGISTER.....	60

1.1

Indledning

Stepmotor Controlleren type SMC20 er en lethåndterlig og prisbillig controller til styring af stepmotorer.

SMC20 kan bruges som stationær enhed eller kobles sammen med en terminal eller personal computer via RS232C/V24 interface. Controlleren er forsynet med ind- og udgange, som giver brugeren mulighed for en meget fleksibel tilpasning til opgaver såsom styring af mindre fræse/bore anlæg, håndteringsautomater og lignende, hvor hurtige og præcise manøvrer er ønskelige, uden at løsningen må fylde for meget eller være for kostbar.

Bemærk at JVL Industri Elektronik kan levere andre controllere med flere features, hvis der for eksempel er krav til galvanisk isolation, større motorstrøm etc.

SMC20 findes kun i een udgave i modsætning til JVL's øvrige program af controllere.

SMC20 har indbygget bipolar chopper driver på op til 2,5 Amp. pr. motorfase. Controlleren indeholder 3 digitale indgange og 3 digitale udgange til generel brug i forbindelse med den aktuelle opgave. 6 Analog-indgange kan benyttes hvor f.eks. en tryktransmitter eller vejecelle skal overføre måleværdier til controlle- ren.

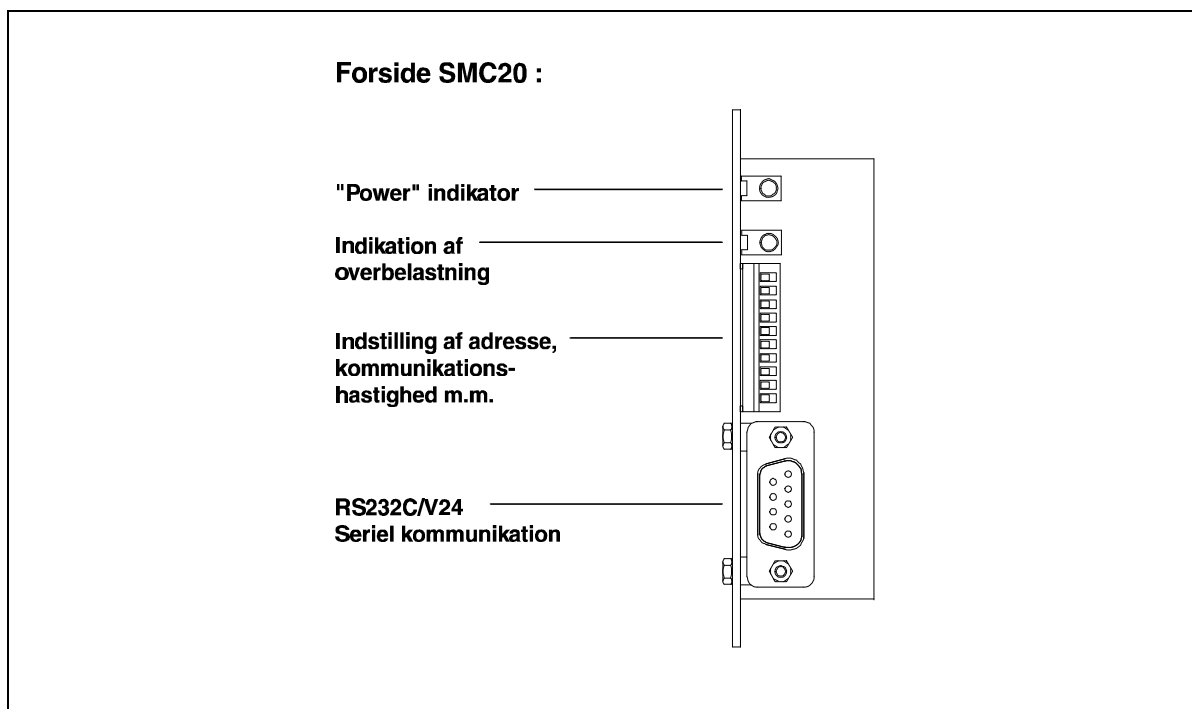
Bemærk at ind-/udgange ikke er beskyttet imod overbelastning. Dog er de vitale ind- og udgangskredsløb monteret i sokkel på en sådan måde at de hurtigt kan udskiftes hvis de er defekte. Motordriveren har dog kortslutnings- sikring, som gør motoren strømløs i tilfælde af kortslutning.

SMC20 har følgende grund-features:

- Forsyning 15-45VDC
- RS232C/V24 kommunikation.
- Enkel programmering.
- Max. kørefrekvens 15kHz.
- RS232 interfacelinie.
- Tilkobling af op til 7 controllere på samme RS232 interfacelinie
- Baudrate 110 - 9600.
- Små fysiske mål.
b34 x h100 x l160 mm
- Termisk sikring.
- 3 Brugerindgange.
- 6 Analogindgange.
- 1 Stopindgang.
- 3 Brugerudgange
- Tilslutning via DIN41612 stik

1.2

Tilslutningsmuligheder



Interface :

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

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 :

Der kræves kun én forsyningspænding. Denne kan vælges i området 15-45VDC eller 12-30VAC. Valget mellem DC eller AC forsyning foregår ved at flytte en jumper. Se afsnit 2.1 Strømforsyning.

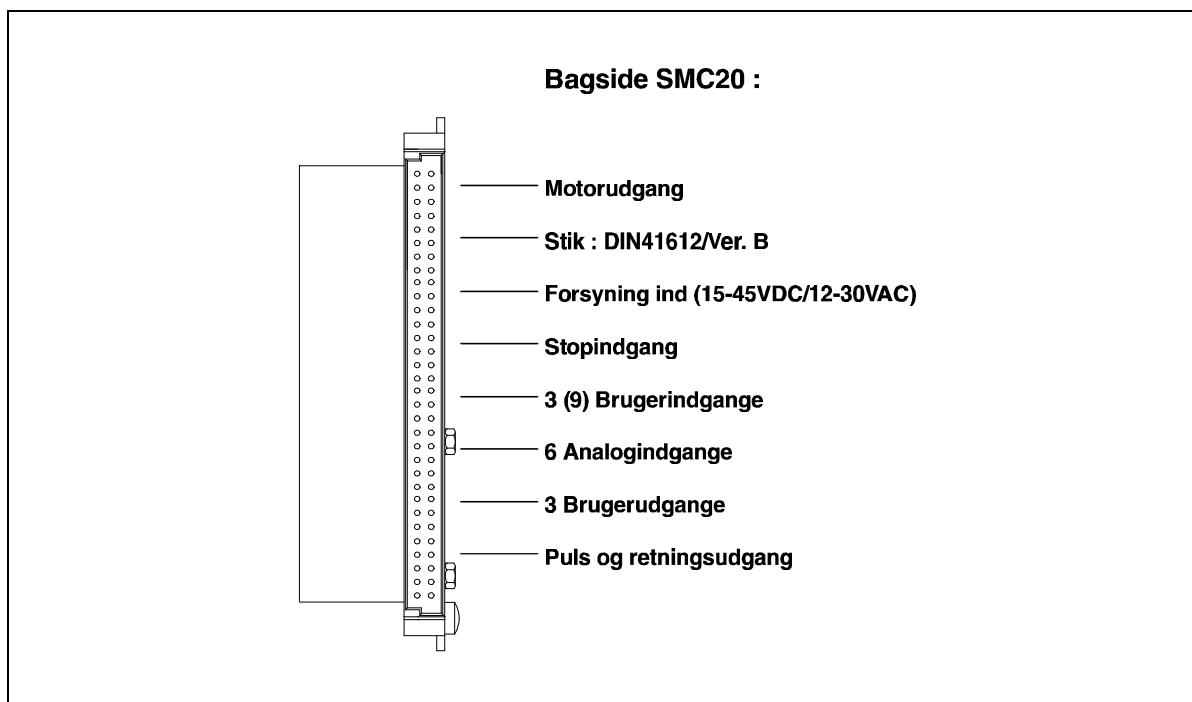
Brugerindgange :

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

Indgangene kan arbejde med TTL niveauer i området 0 til 5V. Bemærk at der ikke er optokoblere i indgangene.

1.2

Tilslutningsmuligheder



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 10mA og kan arbejde i spændingsområdet 0-5V.

Bemærk at udgangene ikke er kortslutningsbeskyttet og galvanisk isolerede fra det øvrige kredsløb.

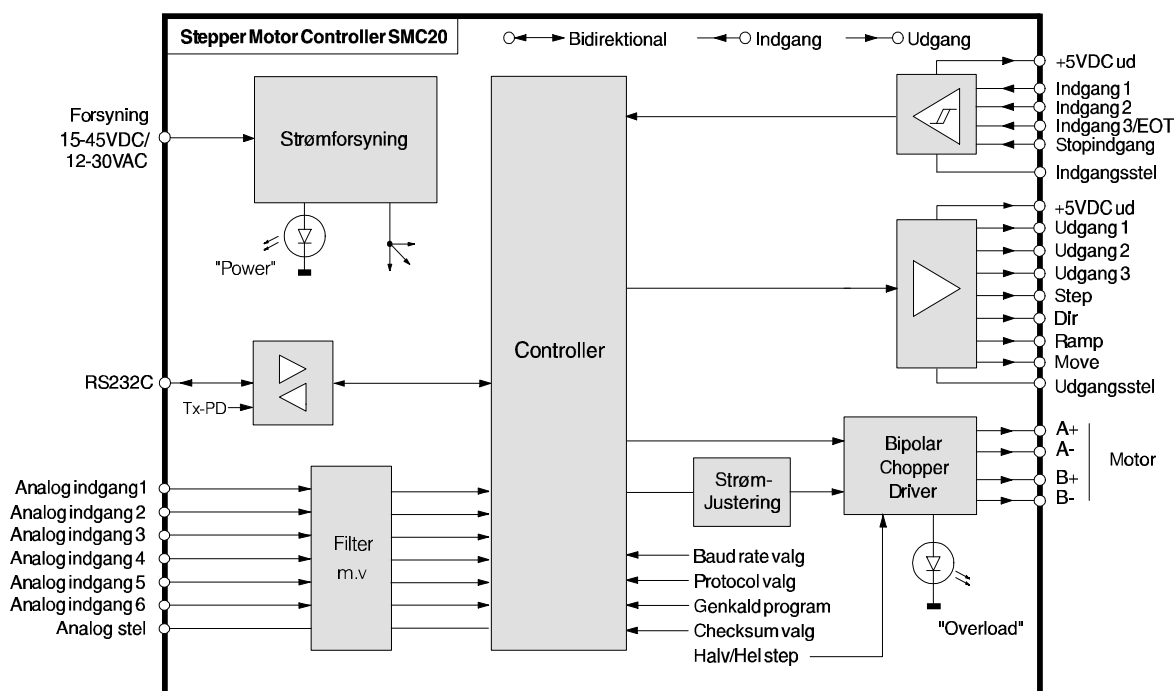
Analogindgange :

De 6 analogindgange kan aftastes via et sæt af kommandoer. Det er f.eks. muligt, at styre motorens tophastighed, absolut eller relativ kørelængde, 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.

2.1

Strømforsyning



For at gøre strømforsyningen så enkel som mulig, skal der kun tilsluttes én forsynings-spænding mellem 15 og 45VDC. Alternativt kan der tilsluttes en vekselspænding på 12-30VAC direkte. Omstilling mellem DC og AC forsyning sker ved at flytte jumper som vist på skitsen på efterfølgende side.

En indbygget konverter sørger for strømforsyning af kontrolkredsløb m.v.

Controlleren er sikret mod fejlpolarisering.

Af hensyn til controllerens virkemåde, anbefales det, at den eksterne strømforsyning har en kondensator på mindst 2000 μ F monteret mellem + og - ledning (P+ og P-) når der benyttes en jævnspændingsforsyning.

Ligeledes kan det anbefales, at ledningerne mellem strømforsyning og controller er mindst 0,75mm².

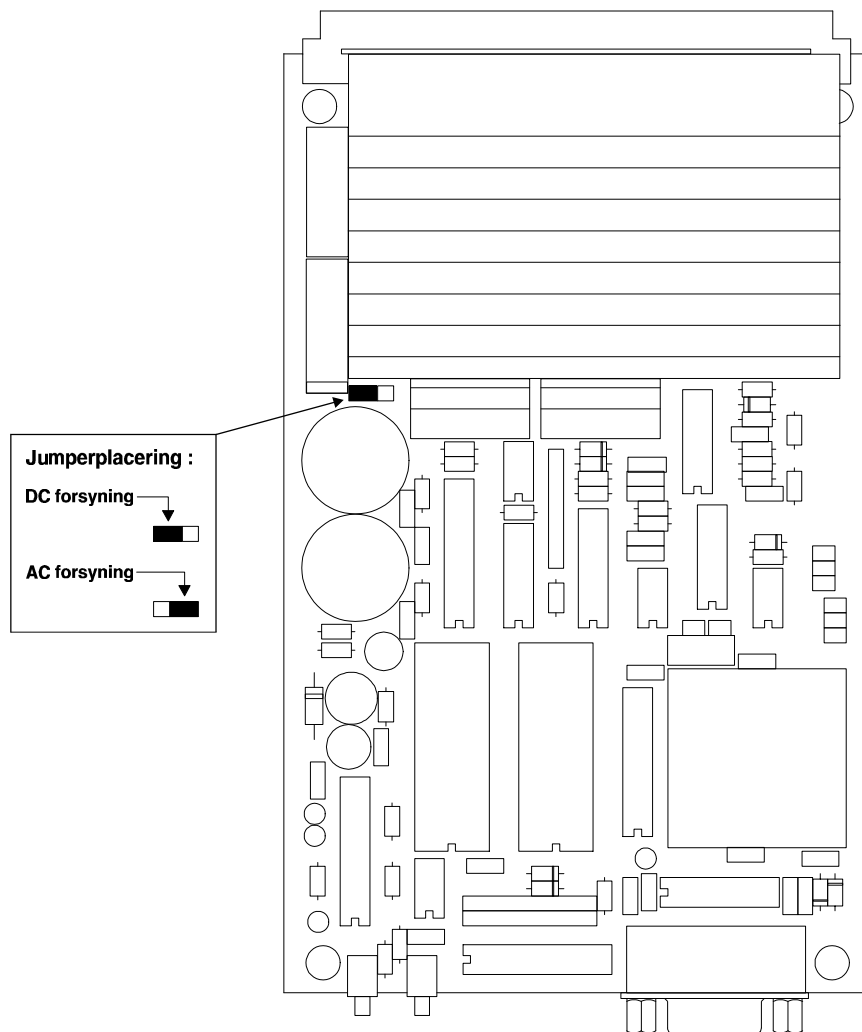
Hvis controllerens forsyningspænding kommer under 10V, vil det interne reset-kredsløb nulstille controlleren, og program m.v. vil i såfald være tabt. Af denne grund bør det sikres at forsyningspændingen forbliver minimum 15V, selvom f.eks. netspændingen falder.

Bemærk !!:

SMC20 ikke er overspændingssikret som de øvrige controllertyper fra JVL.

2.1

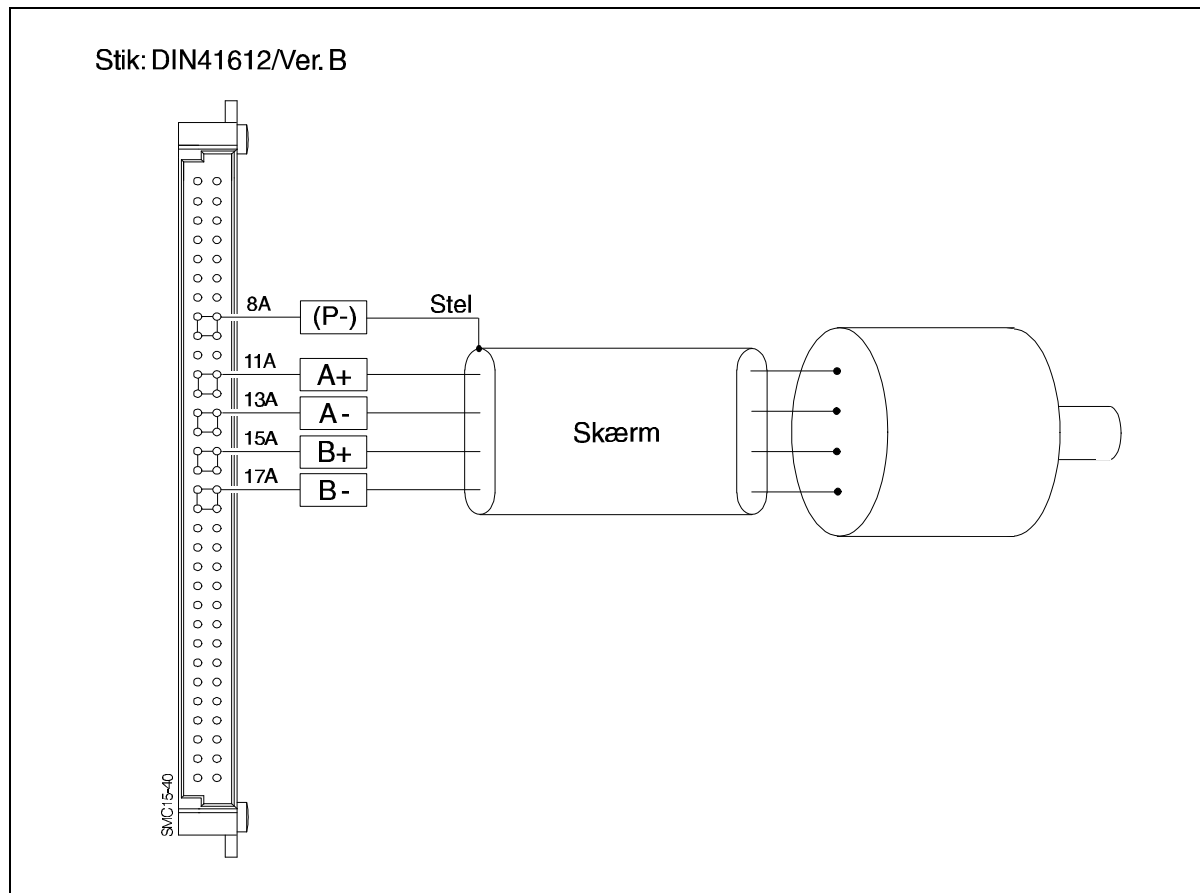
Strømforsyning



Ovenstående illustration viser hvorledes jumperen skal placeres hvis det ønskes at forsyne controlleren med en DC (jævnspænding) eller en AC (vekselspænding). Husk at jumperplaceringen kun må ændres når controlleren er strømløs.

2.2

Tilslutning af motor



Controlleren er beregnet for 2- eller 4-fasede stepmotorer. Controlleren kan yde en fasestrøm op til 2,5 Ampere.

Fasestrømmen er trinløs justerbar, og man har mulighed for at justere holdestrøm, acceleration/decceleration-strøm samt kørestrøm. Justering af strømmen foretages via et sæt af kommandoer - se beskrivelse i afsnit 4.4.

Driveren i controlleren udgøres af en 2-faset bipolar chopper driver. Denne drivertype giver en optimal udnyttelse af motoren, idet der kontinuerligt går strøm i begge steptomorens faser (når der køres helstep).

Chopperdriveren regulerer strømmen med en frekvens på 22kHz (nom.), hvilket sikrer at motoren ikke frembringer hørbare lyde som følge af strømreguleringen.

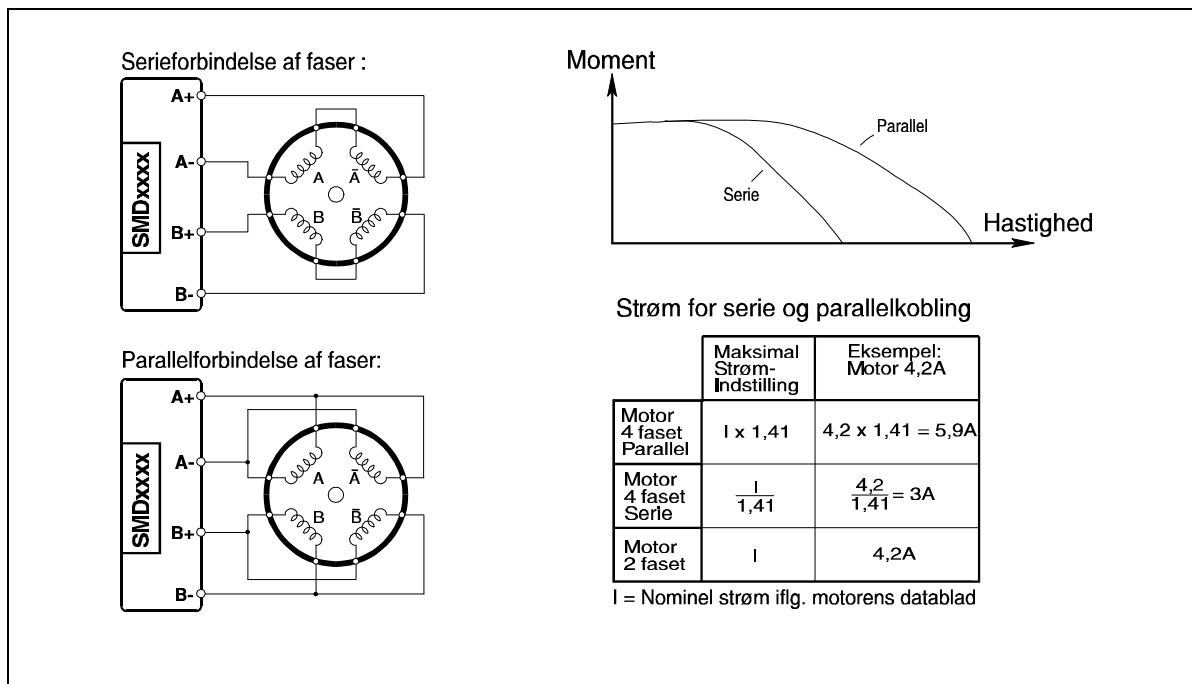
Driverens skiftetid er meget lille (<200ns), hvilket kan medføre at der opstår højfrekvente støjkomponenter på ledningerne mellem driveren og motoren.

Dette kan i visse tilfælde give anledning til uønsket påvirkning af andet elektronisk udstyr i nærheden af stepmotor-systemet. For at imødegå dette bør forbindelsen mellem controlleren og den tilsluttede motor foretages med skærmet kabel, som vist på ovenstående skitse. For at undgå unødige effekttab i ledninger til motoren, anbefales det at anvende 0,75 mm² ledninger.

Bemærk at controlleren ikke er beskyttet mod kortslutning mellem stel (-) og en af motorudgangene - A+, A-, B+ eller B-.

2.2

Tilslutning af motor



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 denne serie af drivere, idet disse drivere køre efter bipolar princippet.

Bemærk 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 driverens udgangsterminaler.

Driveren 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 forbindelser 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 (op til 1kHz) som ved parallelforbindelsen dog med ca. den halve strøm. Dette kan have indflydelse på valget af driver idet det derved kun er nødvendigt at vælge en driver 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 strømtrimmerene 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 driver idet det derved er nødvendigt at vælge en driver der kan levere den dobbelte strøm set i forhold til serieforbindelsen. Se ovenstående illustration. Når faserne på en 4 faset stepmotor kobles i serie 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.

2.2

Tilslutning af motor

Valg af hel eller halv-step

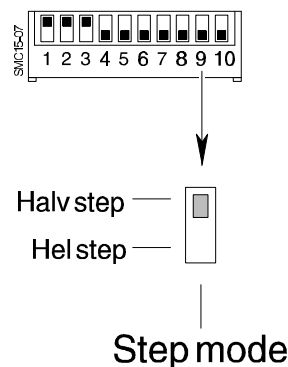
Der kan vælges, om drivertrinet skal køre med halve eller hele step. Det kan ofte være en fordel at køre med halve step, eftersom man derved opnår den dobbelte opløsning pr. motoromdrejning. Dette kan overflødig gøre brugen af mekaniske udvekslinger.

En anden fordel er, at man normalt undgår de resonansproblemer, der næsten uundgåelig vil være, når motoren køres i helstep.

En stepmotor har altid 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 acceleres og deceleres i).

Nedenstående tegning viser, hvorledes halv/hel step indstilles:

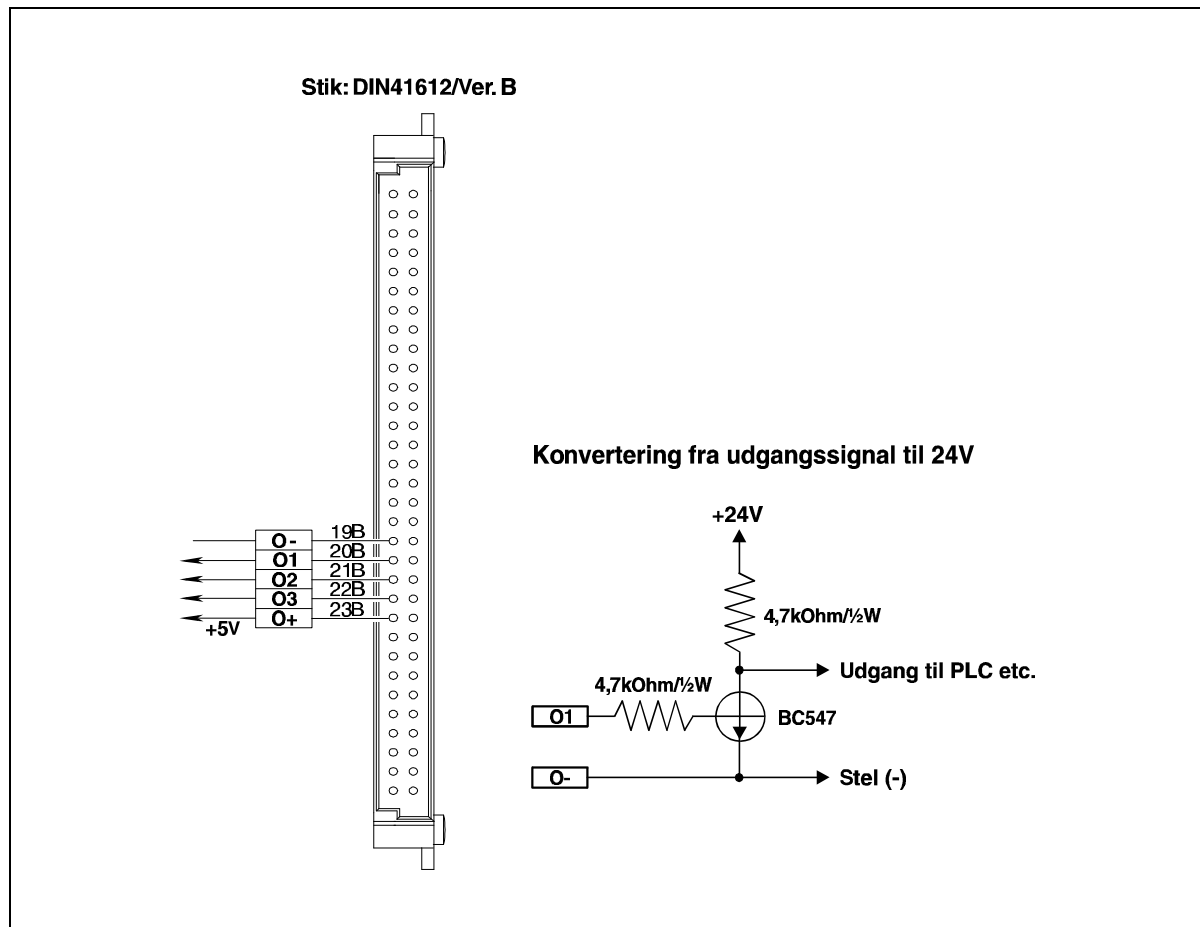


Sikring mod overlast.

Driveren er kortslutningssikret, hvilket betyder, at udgangene bliver spændingsløse i det øjeblik spidsstrømmen overstiger 3,2A i mere end 2mS. En øjeblikkelig kortslutning mellem to vilkårlige udgangsterminaler vil controlleren ikke tage skade af, men blot melde Overload. For at få driveren op at køre igen, skal strømmen til controlleren blot fjernes i mindst 5 sek.

2.3

Brugerudgange



Brugerudgangene O1, O2, og O3, kan bruges til at styre bifunktioner såsom aktuatorer, mindre motorer osv.

Dette giver mulighed for at synkronisere stepmotorens bevægelser med omverdenen.

De 3 udgange er *Push-Pull* typen, og styres software-mæssigt. Alle udgange tillader en strøm op til 10mA.

Udgangs-spændingen er nominelt 0 eller 5VDC.

Terminalen O- er udgangens stelforbindelse. Bemærk at O- har forbindelse til controllerens øvrige stelterminaler (I-, AG, m.v.).

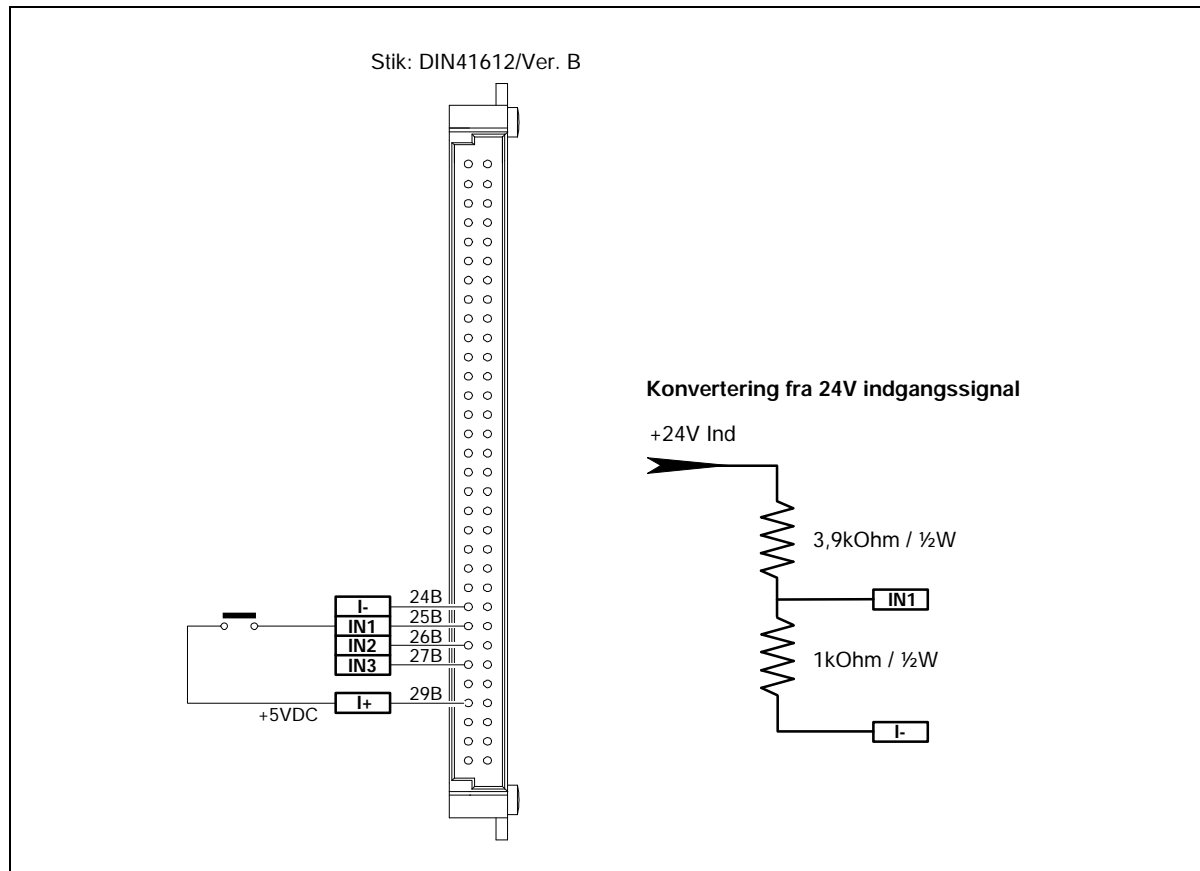
Der er +5VDC tilgængelig på terminalen O+. Denne forsyning kan benyttes til eksterne formål. Bemærk dog at der maksimalt kan trækkes en strøm på 50mA fra denne terminal.

Bemærk !!:

Udgangene er ikke beskyttet mod induktive tilbageslag og er ikke kortslutningssikret.

2.4

Brugerindgange



Der sidder et 1.orden lavpasfilter efter hver indgangsterminal, der afskærer frekvenser over 6kHz. 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 niveauet logisk "0", hvis de ikke er forbundet til noget.

Terminalen I- er indgangenes stelforbindelse. Bemærk at I- har forbindelse til controllerens øvrige stelterminaler (O-, AG, m.v.).

Der er +5VDC tilgængelig på terminalen I+. Denne forsyning kan benyttes til eksterne formål. Bemærk dog at der maksimalt kan trækkes en strøm på 50mA fra denne terminal.

Bemærk !

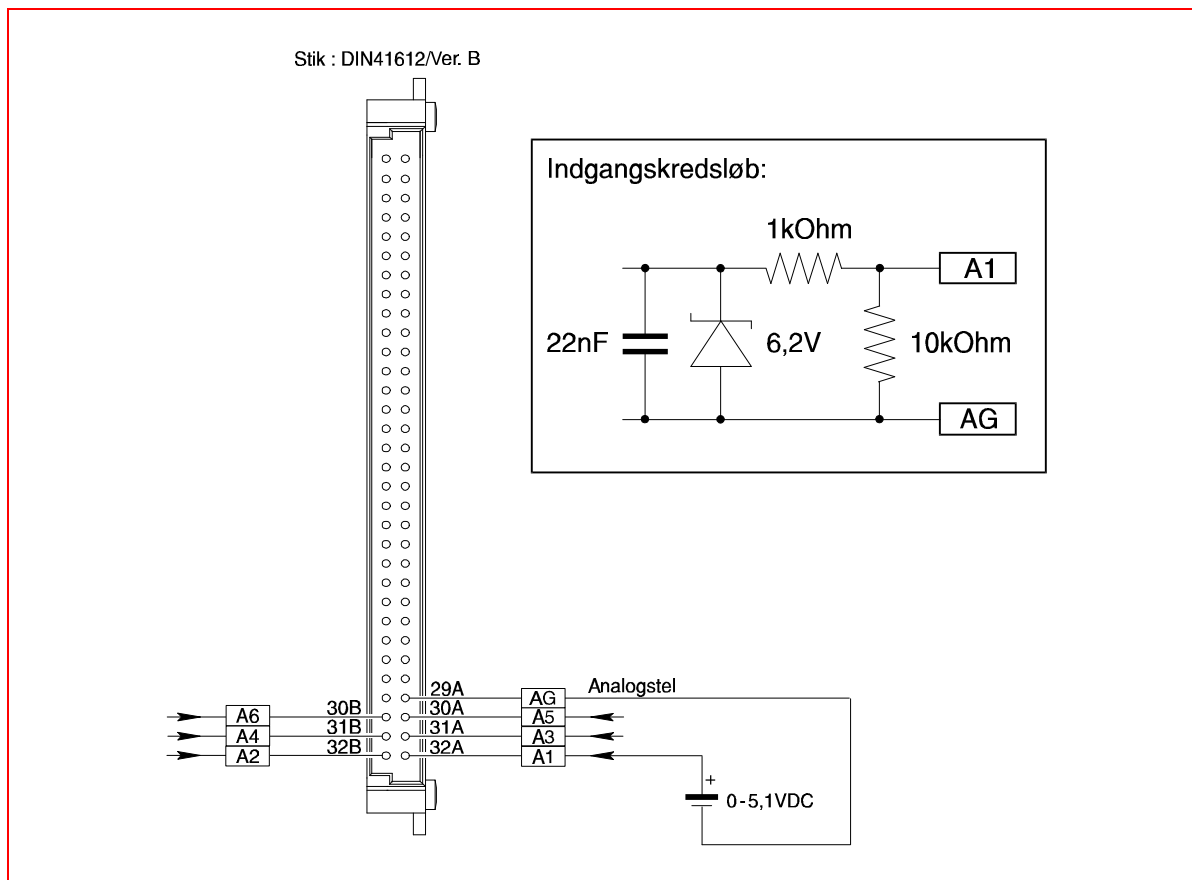
Indgangene må maksimalt påtrykkes spændinger på +5V. Indgangene brænder af hvis de påtrykkes spændinger uden for intervallet 0-5V.

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

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

2.5

Analogindgange



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, 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 ana-

logindgang 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.

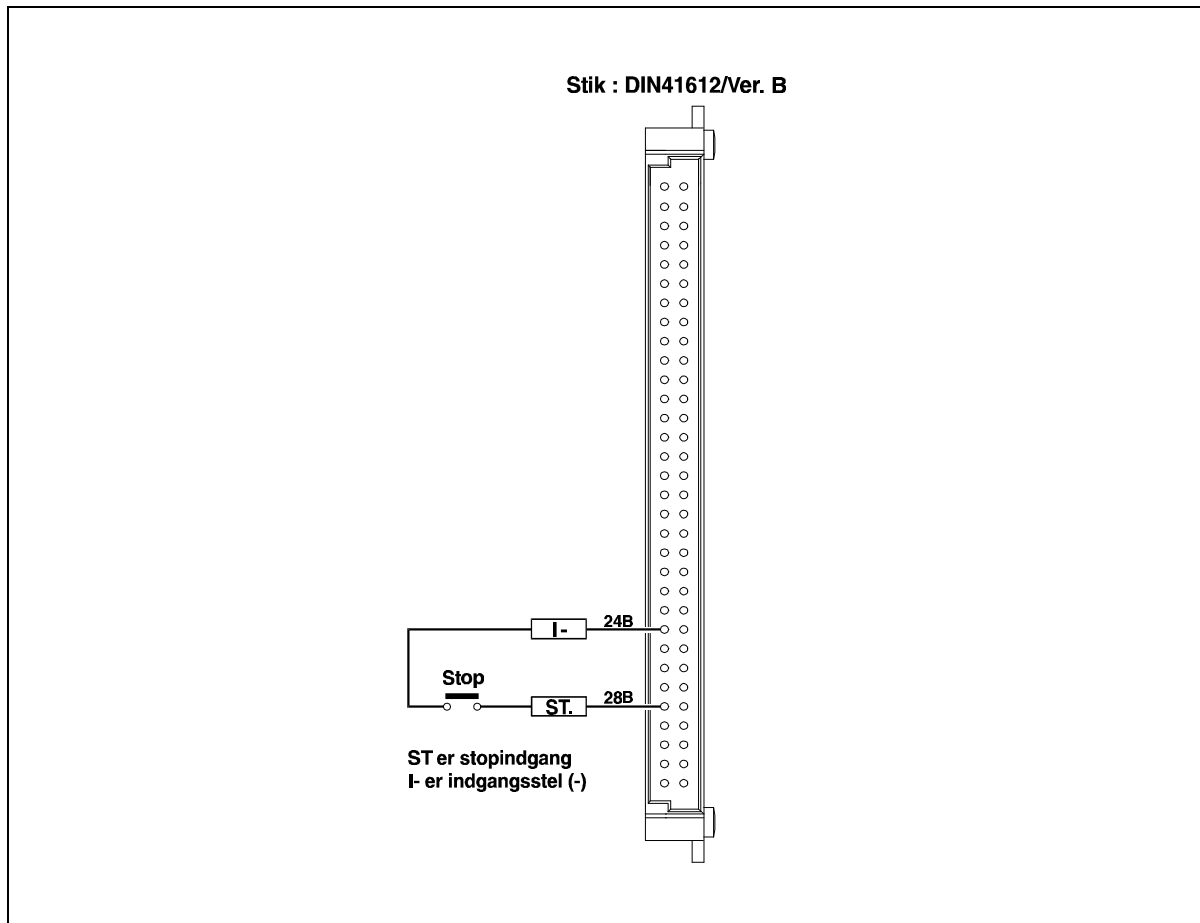
For at undgå fejlmålinger, skal stel benet AGND (se stikskitse afsnit 2.7) benyttes sammen med de 6 analogindgange.

Se elektriske data afsnit 5.1 vedrørende yderligere informationer.

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

2.6

Stopindgang



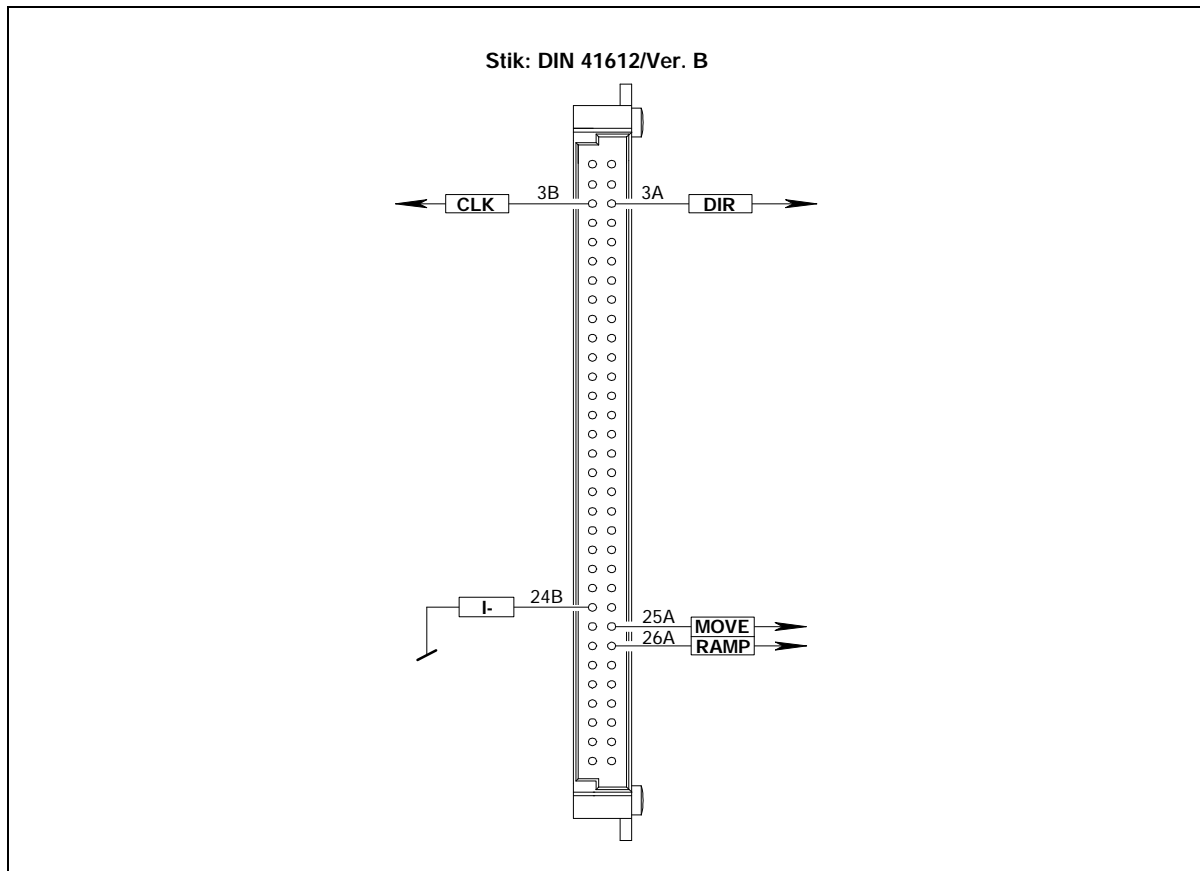
Ønskes det at man med øjeblikkelig virkning skal stoppe motorkørslen, gøres dette ved at lægge stopindgangen til stel.

Fjernes stelforbindelsen, vil motoren 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-/decelerationsramperne.

(se også under motorkommandoer afsnit 4.6).
Bemærk at stopindgangen er inaktiv hvis den ikke er forbundet.

2.7

Statusudgange



SMC20 indeholder 4 udgange der indikerer følgende. Bemærk at alle 4 udgange er TTL niveau (0/5V) og ikke er kortslutningsbeskyttede).

Move:

Udgang der indikerer når motoren kører. Udgangen vil være logisk "1" når motoren starter, og vil først være logisk "0" når motoren holder stille.

Ramp:

Udgang der indikere når motoren accelererer eller decelererer. Udgangen vil være logisk "1" når motoren accelererer eller decelererer, og vil kun være logisk "0" når motoren holder stille eller kører med konstant hastighed.

CLK:

Hver gang motoren kører et step, fremkommer en positiv impuls på denne udgang. Pulsens bredde er ca. 16 μ S.

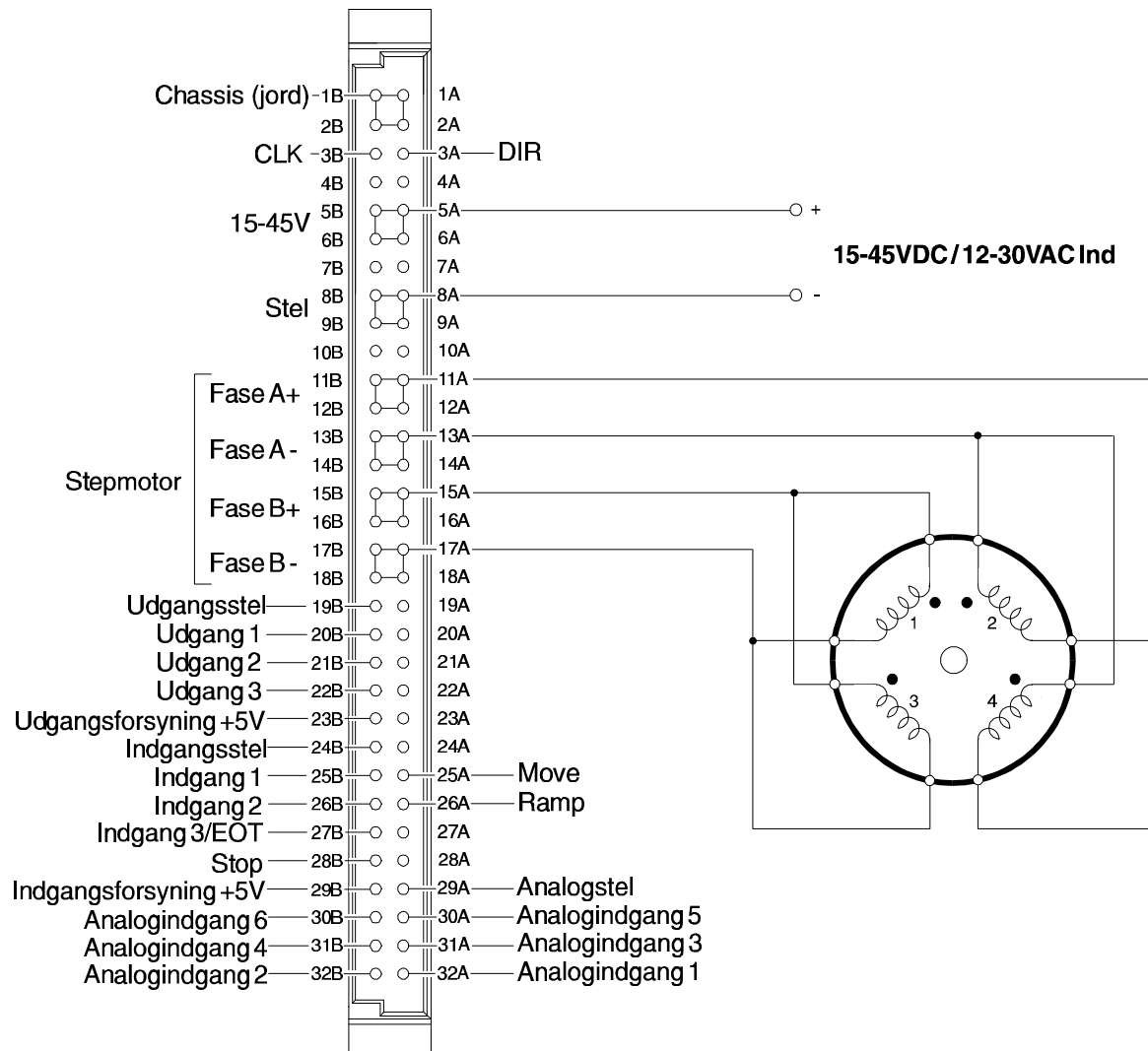
DIR:

Udgang der indikerer hvilken vej motoren kører. Udgangen vil være logisk "1" når motoren kører i positiv retning og logisk "0" når motoren kører i negativ retning.

2.8

Stikforbindelser

(DIN 41612 Ver.B)



Indgangsstel 24B, udgangsstel 19B og analogstel 29A, er internt forbundet sammen, men af støjhensyn anbefales det at benytte de enkelte stelforbindelser separat - indgangsstel sammen med indgangene o.s.v.

3.1 Interfaceforbindelser

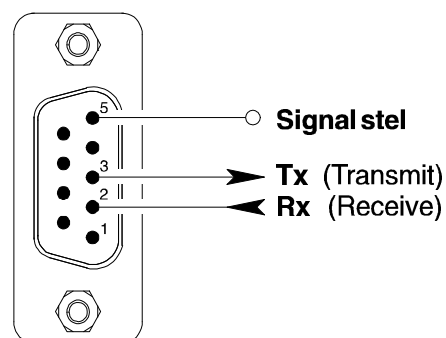
Interfaceforbindelser.

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.

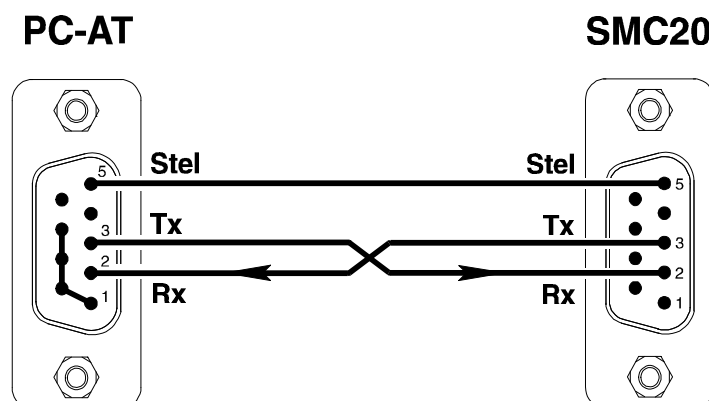
Kabellængden bør ikke overstige 10 meter, men bruges et længere kabel, kan der køres med checksum, se afsnit 3.5.

Controller interface:



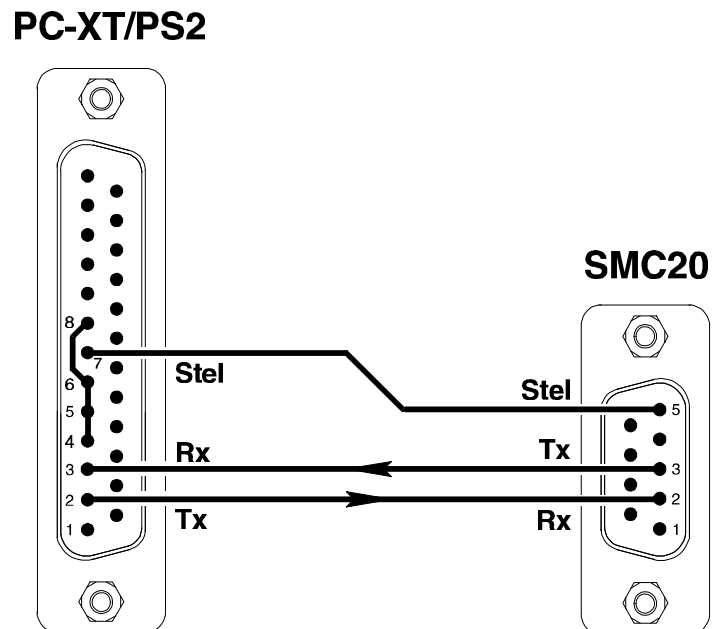
Skal man kommunikere fra en PC'er, kan disse skemaer anvendes.

Forbindelser mellem controller og IBM AT, eller kompatibel :



3.1 Interfaceforbindelser

Forbindelser mellem controller og IBM XT/PS2 eller kompatibel :



3.2

Adressering

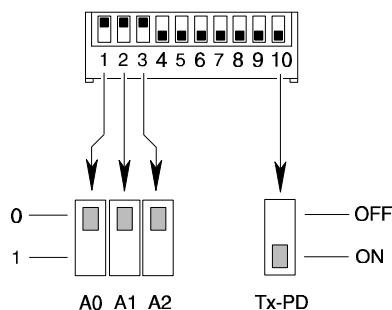
Controlleren kan konfigureres til at reagere på alt, hvad der kommer på interfacelinien, også kaldet Point to point. Der er mulighed for at koble op til 7 kontrollere på samme interfacelinie. For at opnå dette, skal dip-switchene stilles på en sådan måde, at hver controller får sin egen adresse, kaldet - 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.

Til adresseindstillingen bruges følgende skema:

A0	A1	A2	Adresse	Protokol
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

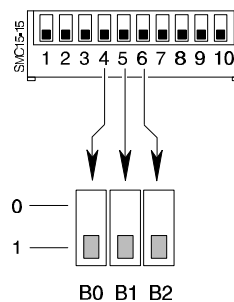
VIGTIGT! : Hvis adresse switchenes stilling er blevet ændret, skal der først slukkes for kontrolleren,



3.3 Kommunikations hastighed

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 først 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.

3.4 Kommandoformat

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 el. 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 altafgørende, 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. Det egentlige resultat smides væk, og resten beholdes som den endelige checksum. Det egentlige regnestykke ser således ud:

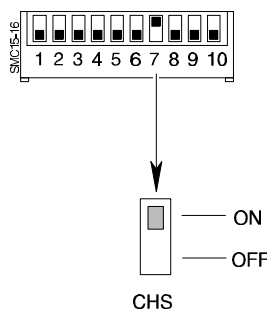
Adressekarakter	1	=	ASCII 49
Kommandokarakter	A	=	- 65
Argumentkarakter	3	=	- 51

Resultat + rest = $(49+65+51)/128$
Checksum = Rest * 128

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 ON.

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



4.1

Generelt om software

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 E²PROM, 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) vil diverse kommandoer m.v. være 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 parantes.

Den egentlige kommando er bogstavet, ordet i parantesen 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 uden for de angivne rammer.

Se endvidere afsnit 3.4 vedr. kommandofor-
mat.

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 indprogramerede 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 standbymode.

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 i stand til at efterkomme.

E5 Positionstælleren ville have overskredet sit maksimum på -8.388.607 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 ² 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.
Q	(Query)	Viser programmet, der ligger i arbejdslager.
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 omdrejningsretning (+/-) 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 ²).
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.

4.2

Kommandooversigt

(Fortsættelse)

Brugerinterface :

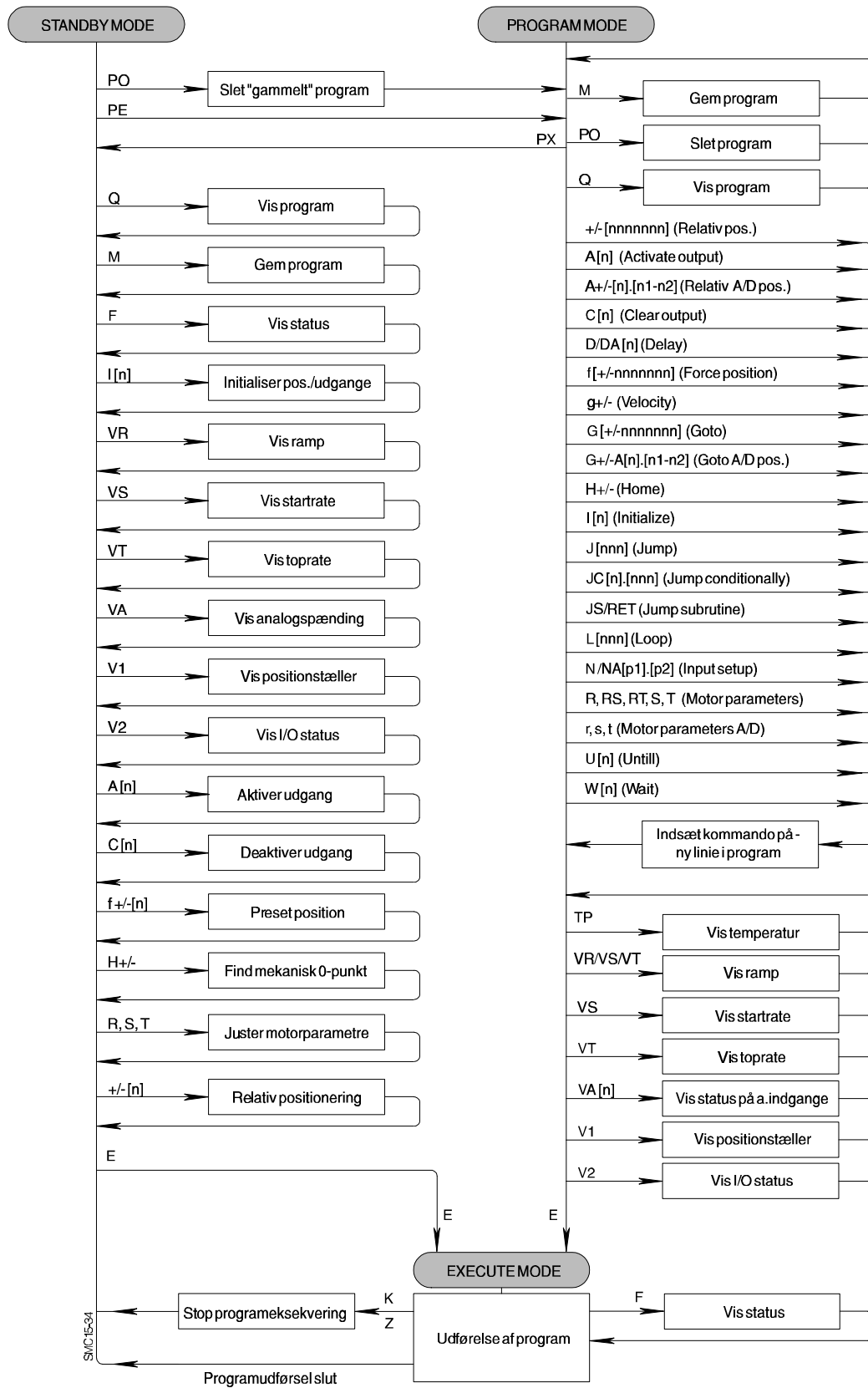
A [n]	(Activate)	Tænder en udgang.
C [n]	(Clear)	Slukker en udgang.
U [n]	(Until)	Gentager program(del), indtil en given indgang aktiveres.
VA [n]	(Verify ainput)	Viser spændingen målt på en af de 6 analogindgange
VA		Viser logiske niveauer på analogindgange.
V2	(Verify)	Viser brugerinterface status.
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.)	Betinget spring til programlinie når analogindgang er påtrykt en given spænding.
JS [n1]	(Jump sub)	Ubetinget spring til subrutine.
RET	(Return)	Retur fra subrutine.
L [nnn]	(Loop)	Gentager programdel et specificeret antal gange.

4.2

Kommandooversigt



4.3

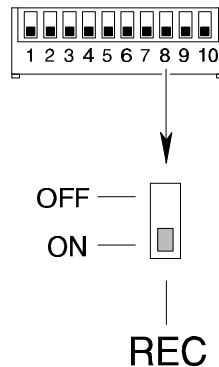
Systemkommandoer

- 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 -8.388.607 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.

Q (Query) Viser det pågældende program, som ligger i arbejdslageret incl. køreparametre. Ønskes en udskrift af programmet, der ligger i arkivet, bruges først X kommandoen. Vær da opmærksom på, at programmet i arbejdslageret bliver erstattet af arkivprogrammet.

4.3

Systemkommandoer

- TP**
(Temperature) Viser den aktuelle temperatur inde i controlleren.
Den kan bruges til at verificere, om controlleren arbejder indenfor sit temperatur-område 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 kommandoen. Dette kan f.eks være en fordel, hvis controlleren med jævne mellemrum skal udføre det samme stykke program. Send først X og dernæst E kommandoen. 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 kommandoen, 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]
(Relativ)

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.

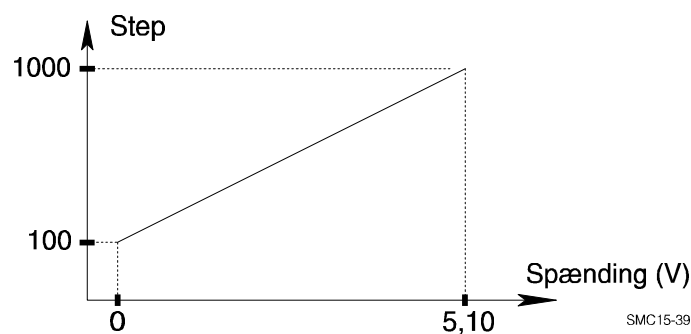
±A [n].[n1-n2]

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.



4.4 Motorkommandoer

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 hvilket som helst 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.

CS [0-2500] (Current Standby)

Fastsættelse af motorstrømmen når motoren ikke kører.

CR [0-2500](Current Ramp)

Fastsættelse af motorstrømmen ved acceleration/deceleration.

CT [0-2500] (Current Top)

Fastsættelse af motorstrømmen når motoren kører med tophastighed.

Eksempel:

(Program)

.

CS500	Fastsætter motorstrøm ved stilstand til 500mA (0,5A).
CR2000	Fastsætter motorstrøm under acceleration/deceleration til 2000mA (2A).
CT1500	Fastsætter motorstrøm ved tophastighed til 1500mA (1,5A).
+100	Motor kører 100 step frem.
CT2100	Fastsættelse af ny motorstrøm ved tophastighed til 2100mA (2,1A).

.

.

4.4

Motorkommandoer

- 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 -8.388.607 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 ±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.
- 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

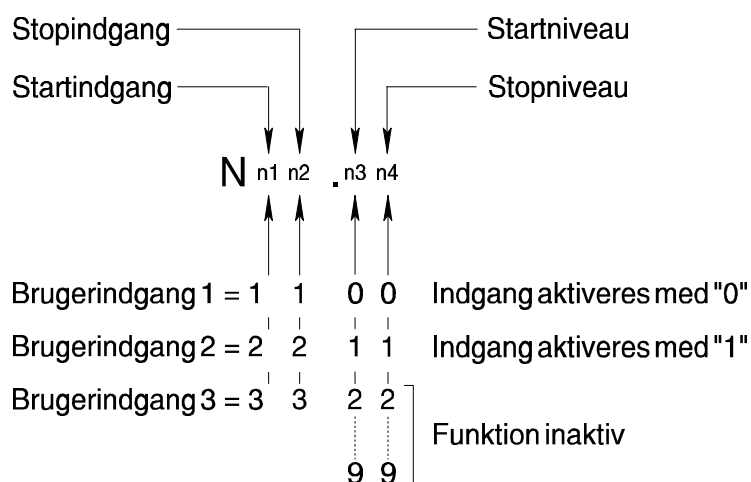
Motorkommandoer

N [n1n2.n3n4]
(Input setup)

Kommandoen gør det muligt, at starte og stoppe motorkørslen via styrespændinger på brugerindgangene.

Kommandoen selv starter ikke motorkørslen. N 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 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 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)

4.4

Motorkommandoer

(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:

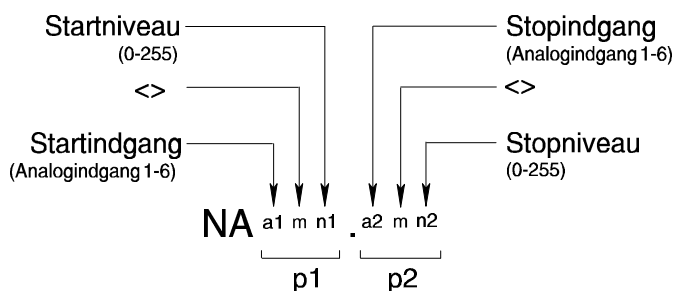
N11.19 vil, når *G+3500* bliver udført, starte motoren, når brugerindgang 1 bliver logisk "1" og stoppe, når positionen +3500 er nået. Motoren vil køre efter de angivne parametre sat i R,S og T kommandoen.

4.4

Motorkommandoer

NA [p1.p2] (Input setup)

Kommandoen gør det muligt at starte og stoppe motorkørslen via analogindgangene. Princippet i kommandoen er som N [n1n2.n3n4]. Kommandoens to parametre (p1 og p2) angiver henholdsvis start og stopbetingelsen. 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 kommandoen, skal NA kommando udføres før den pågældne 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 pågældne analogindgang. Referencen kan tillægges en værdi i området 0 til 255.
- n2:** Referenceværdi for stop. Referenceværdien sammenlignes med det målte på den pågældne 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 totalt 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.

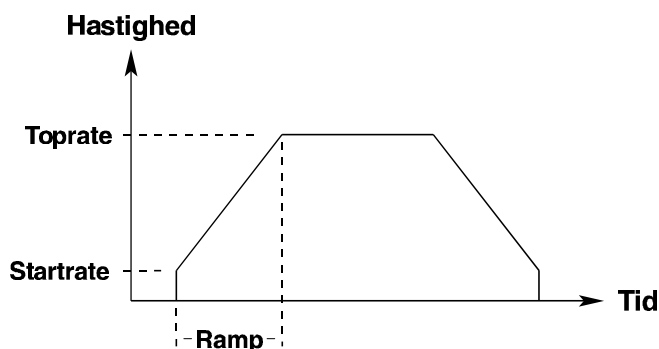
Værdien kan specificeres i området 16 til 15000 step/sek. Defaultværdi: 1000step/sek

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/sek². 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/sek².

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/sek². 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

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.

s [n1.n2] (A/D start rate) Kommandoen s [n1.n2] fastsætter start rate parameteret, og kommandoent [n1.n2] fastsætter top rate parameteret.

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 -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)

r [n1.n2]
(A/D ramp step)

Eksempel:

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 brugerind-
          -  gang 3 bliver lagt til logisk "0".
A1
D25
C1
** UA1    -  Gentager programmet mellem * og **, indtil analogind-
          -  gang 1 bliver påtrykt logisk "0"
```

4.5

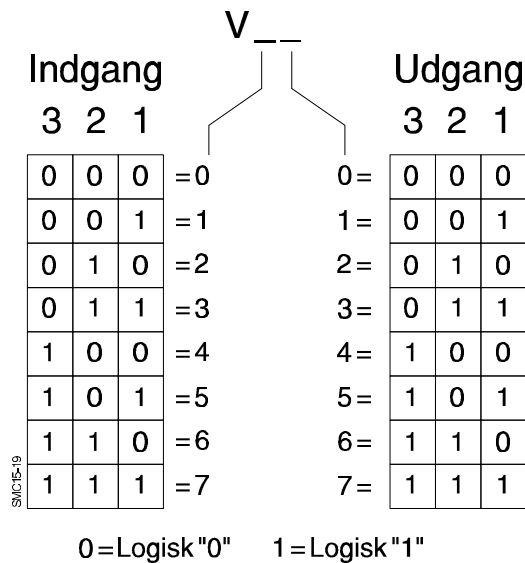
Brugerinterface

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 interface forbindelse til en Computer/terminal. Den kan ikke bruges i selve programmet.

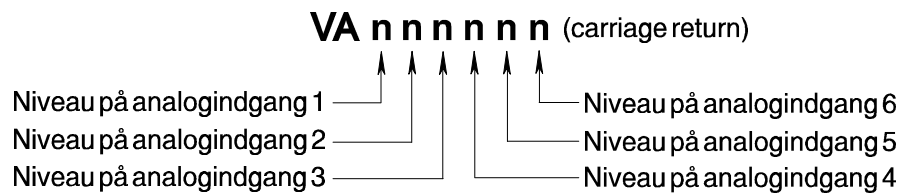
VA [1-6] (Verify ainput)

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 (Verify ainput) Denne kommando viser de digitale niveauer på analogindgangene. 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,
(Wait for) henvises til 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 liniernr. der specificeres
(Jump) efter kommandoen.
Det specificerede liniernr. kan angives i området 0-255.

Eksempel:

Liniernr.:

0	A1
1	+1000
2	A2
3	G+5
4	C2
5	J2

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 linie nummer 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 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	JC5.4	- Springer til linie 4 hvis brugerindgang 1 og 3 er logisk "1"
5	G+0	
6	C1	
7	JC3.2	- Springer til linie 2, hvis brugerindgang 1 og 2 er logisk "1".

4.6

Flowkommandoer

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

Kommandoen fungerer i princippet som JC[0-7] - betinget af et indgangsniveau, 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.

Programseksempel:

Linienr.:

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

4.6

Flowkommandoer

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

Kommandoen fungerer i princippet som JC, betinget af et indgangsniveauspringes 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].[n]

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

mn: Referenceværdien (n), 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 formlerne:

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

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	JCA3>150.2	- Springer til linie 2, hvis analogindgang 3 er større end eller lig med 3,00V.
5	G+0	

4.6

Flowkommandoer

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 forsætter 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 gentages, 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.

5.1

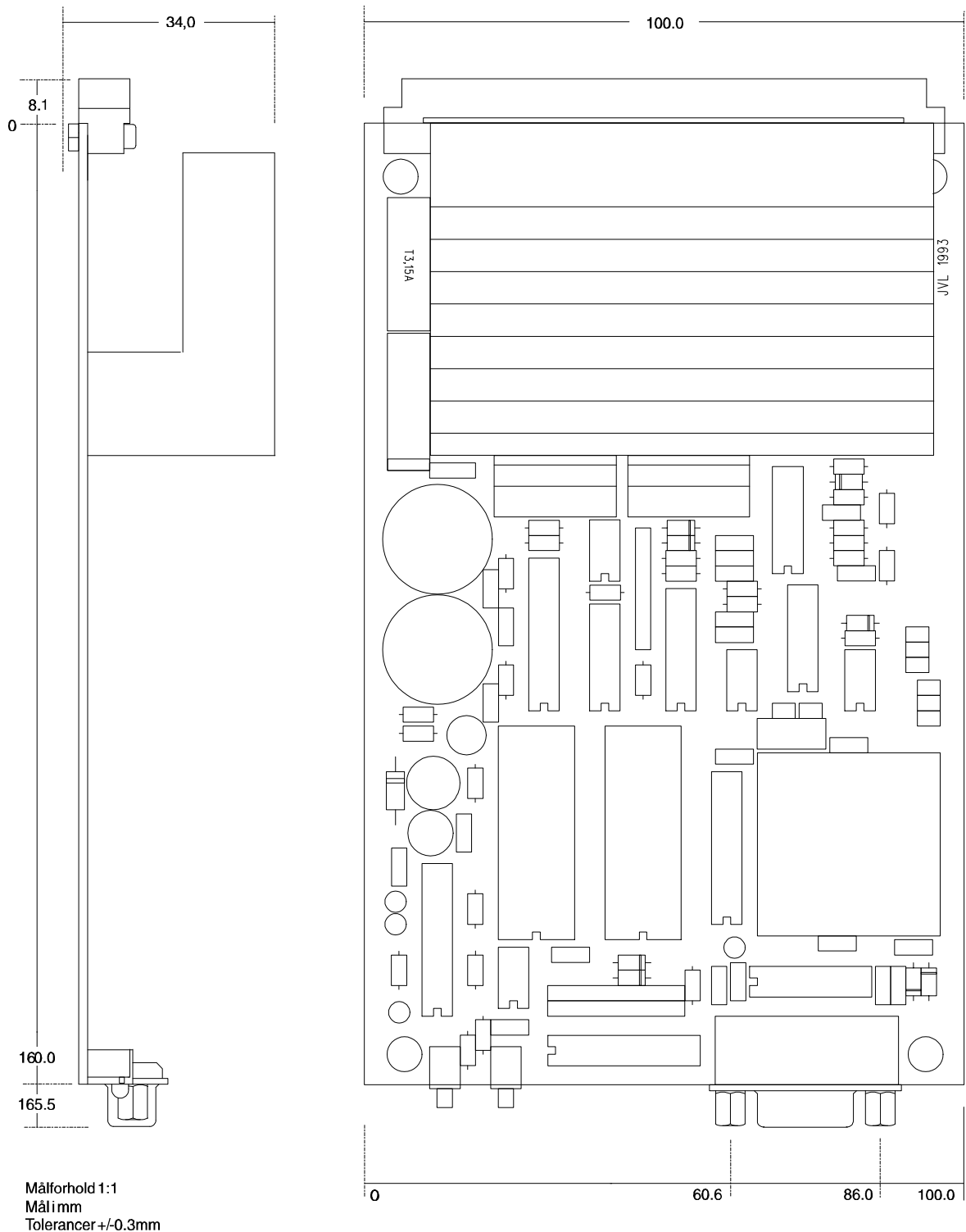
Elektriske data

	Min.	Typ.	Max.	Enhed
Strømforsyning				
Forsyningsspænding (DC)	15		45	VDC
Forsyningsspænding (AC)	12		30	VAC
Effektforbrug (uden motor)		4		W
Motordriver :				
Udgangsstrøm (pr. fase)	0,2		2,5	ADC
Udgangsspænding	15		45	VDC
Chopperfrekvens		22		KHz
Interface :				
on	-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
Brugerindgange + stopindgang :				
Indgangsimpedans	33			kohm
Forsyning : spænding	-	5	-	VDC
Logisk "0"	<1,35		-	VDC
Logisk "1"	-		>3,5	-
Analogindgange :				
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		-	V DC
Logisk "1"	-		>2,5	-
Indgangsimpedans		10		kOhm
Brugerudgange :				
Spændingsforsyning			0,33	V DC
Logisk "0"	-	-	-	V DC
Logisk "1"	3,85	-	-	V DC
Belastningsstrøm pr. udgang			10	mA DC
Diverse :				
Arbejdstemperatur	0		50	°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. Kontrolleren 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 4kbit E²PROM (4096 bit eller 512 byte).

Der bruges 4 byte til at korigere A/D konverterens offset-fejl, og justering af termometer-funktionen (TP). Dermed er der 508 byte tilbage til program m.v.

For at kunne udnytte de 508 byte optimalt, fremgår det af nedenstående tabel, hvor meget plads de enkelte kommandoer optager. Det samlede program må ikke fylde mere end de 508 byte.

Forsøges et program større end 508 bytes indlæst, vil controlleren melde "E3".

1 byte :	g± H± RET (Blank linie)	6 byte :	DA[n].[n1-n2] ±A[n].[n1-n2] G±A[n].[n1-n2]
2 byte :	A[n] C[n] I[n] r,s,t[n.n] U[n] W[n] L[nnn] J[nnn] JS[nnn]	7 byte :	NA[p1-p2]
3 byte :	CR[nnnn] CS[nnnn] CT[nnnn] R[nnnnn] RS[nnnn] RT[nnnnn] S[nnnn] T[nnnn] N[nn.nn] D [nnnnn]		
4 byte :	f [±nnnnnnn] ± [nnnnnnn] G [±nnnnnnn] JC[n.nnn] JCA[p].[n1]		

5.4

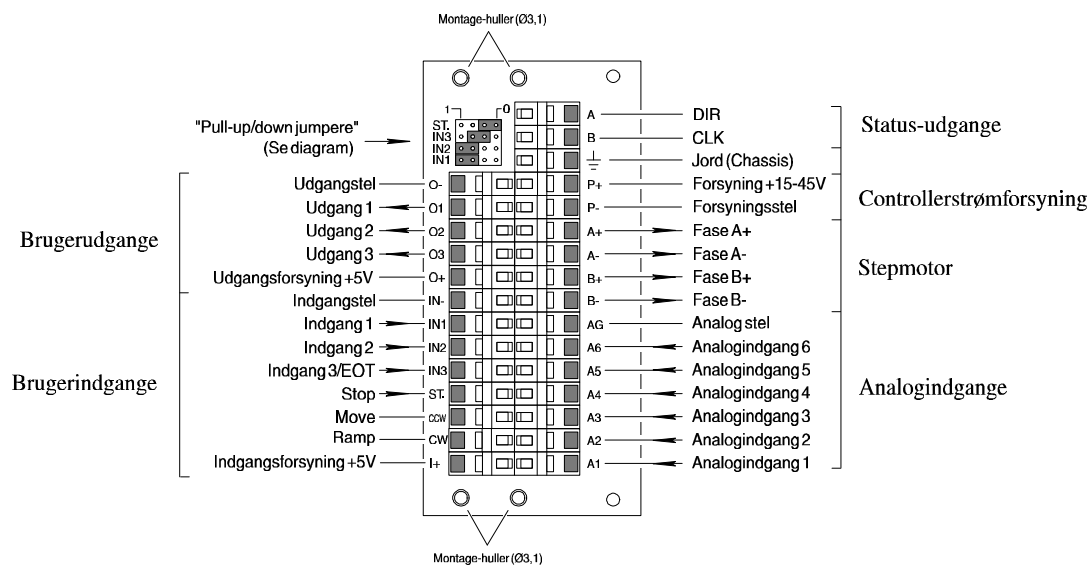
Tilbehør

Connectorboard.

Connectorboard type CON10 er et konverteringsprint der gør ledningsmontage enklere og mere overskuelig. Alle tilslutningerne foregår via skrueterminaler.

Forbindelses-skema

Type: "CON10"



Interfacekabel.

Der findes 2 kabeltyper til programmering via RS232 fra en PC.

1. Type RS232-9-9 Til PC-AT med 9 polet stik.
2. Type RS232-9-25 Til PC-XT eller PC-PS2 med 25 polet stik.

Editor-software til PC.

Til programmering og indkøring af SMC20 findes editor-software type "Editor2".

Denne editor muliggør opbygning af et program med kommentarer m.v.

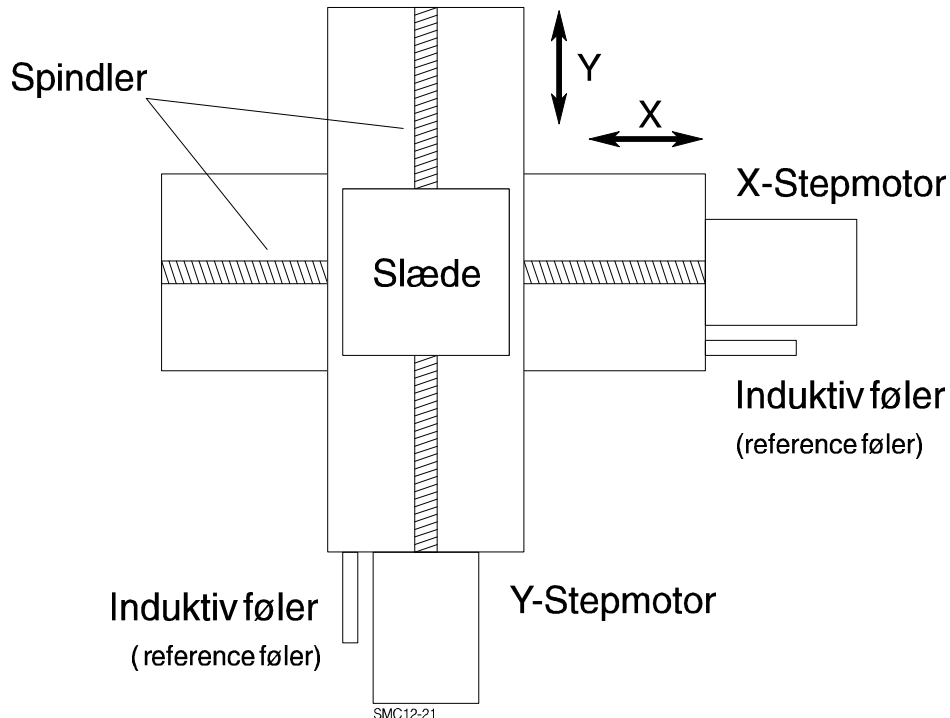
Programmet kan overføres til SMC20 og eksekveres. Status på position, ind- og udgange m.v. kan overvåges via editoren.

5.5 Anvendelseseksempel for SMC20

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 registrerer, 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, være udgjort af en DC-motor, der bliver ført ned af en hydraulik/pneumatik cylinder.

Krydsbord set fra oven:



5.5 Anvendelseksempel for SMC20

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. Den afgiver 12V når den er færdig med at bore, og står i sin udgangsposition.
- 4) Startkontakt.
- 5) Interne synkroniseringssignaler mellem de to controllere.
(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

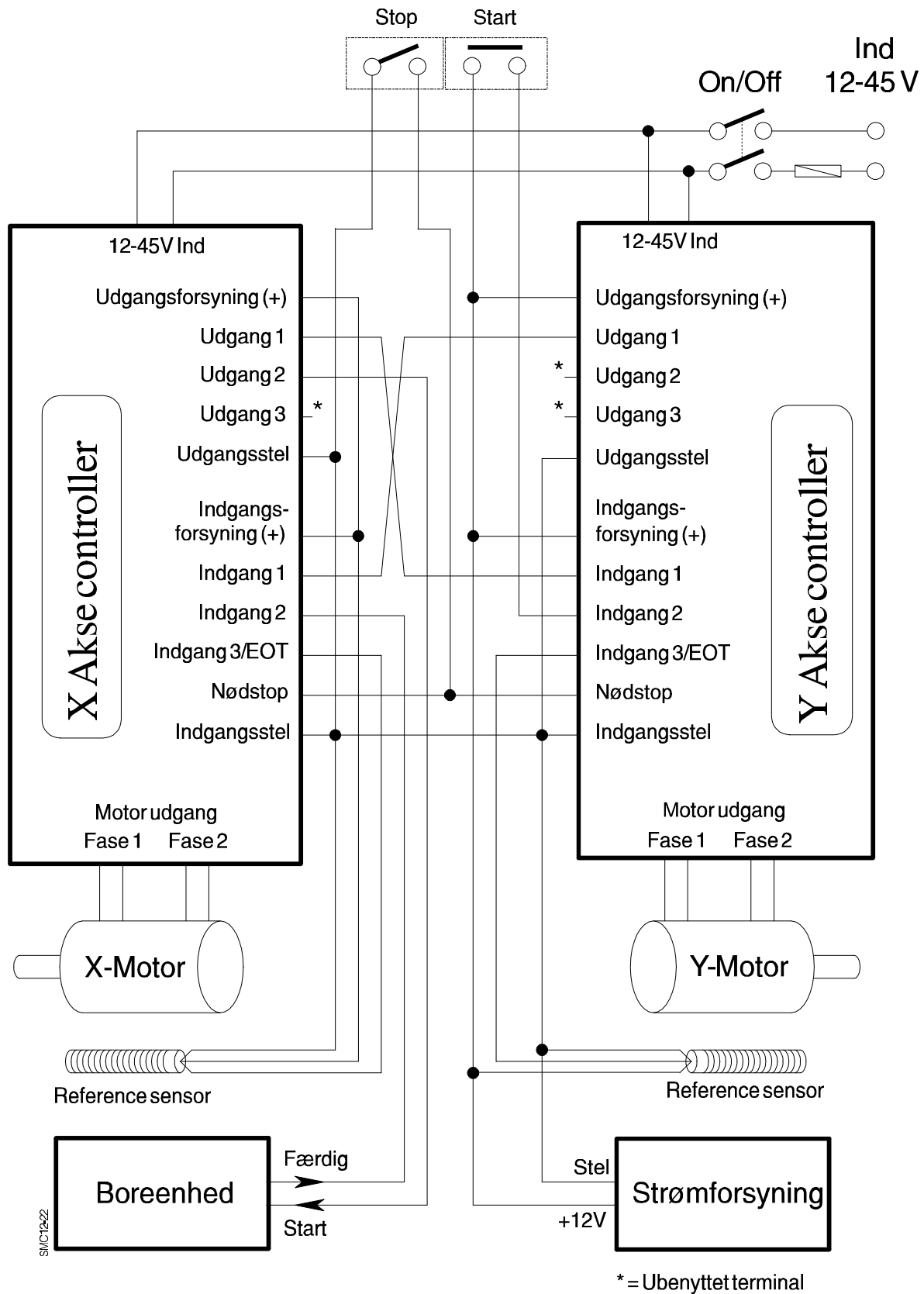
Bruges ikke

Udgang 3

Bruges ikke

5.5 Anvendelseksempel for SMC20

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



5.5 Anvendelseksempel for SMC20

Program til styring af krydsbord.

Før programopbygningen påbegyndes, bør alle involverede parametre kendes.

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

1 Hul - 4013,7387

2 Hul - 5164,1949

Korodinatorerne 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 Anvendelseseksempel for SMC20

Program til styring af krydsbord.

X-Controller.

Start:

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

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.

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.

Y-Controller.

Start:

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

- 4) Giv klarmelding til X-controller.

- 5) Fortsæt når X-controller har givet klarmelding.

-

-

- 8) Kør til position 1949.

- 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 controlleren bliver færdig, og giver klarmelding.

5.5 Anvendelseseksempel for SMC20

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 colloner, 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.

Hvergang 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)

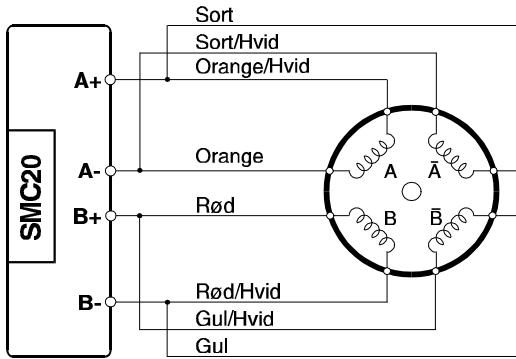
Y-Controller.

- 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)

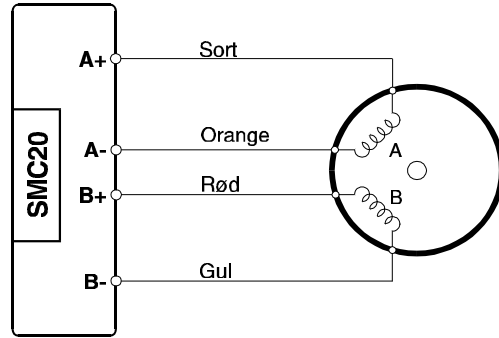
5.6

Motor forbindelser

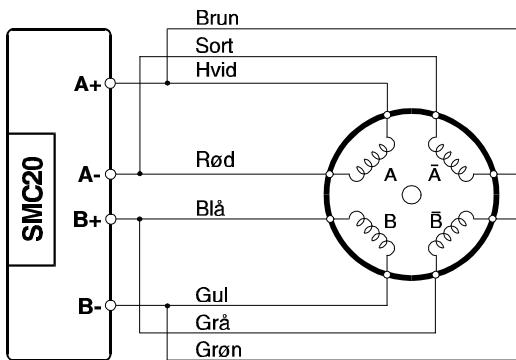
**Forbindelse af MAE motor
Type HY200-xxxx-xxx-x8**



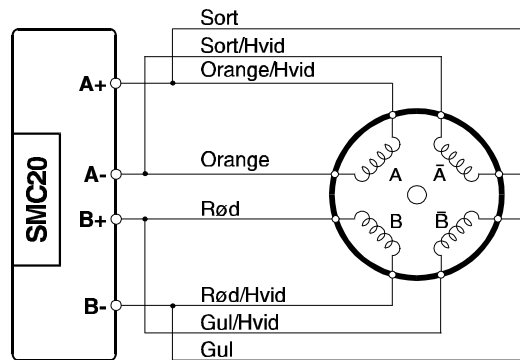
**Forbindelse af MAE motor
Type HY200-xxxx-xxx-x4**



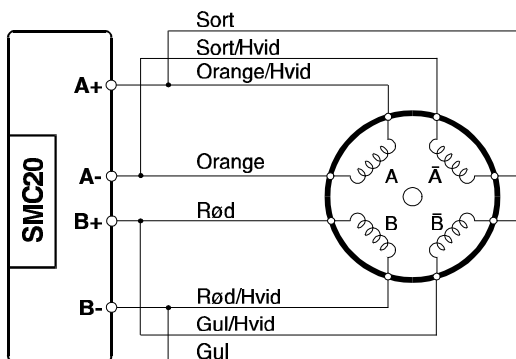
**Forbindelse af Zebotronics motor
Type SMxx.x.xxx.x**



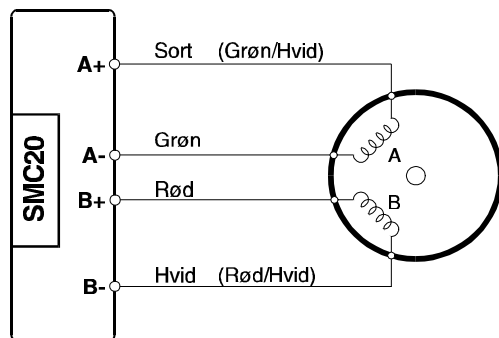
**Forbindelse af Pasific Scientific
Type: Powermax serie**



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



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



5.7

Stikordsregister

- +/- kommando**, 31
- ±A kommando**, 31
- A kommando**, 41
 - Adressering, 18
 - Analogindgange, 12
 - Arkiv, 29
- Brugerindgange, 11
- Brugerinterface, 41; 43
- Brugerudgange, 10; 28
- byte, 51
- C kommando**, 41
 - Checksum, 21
 - Connectorboard, 52
- CR kommando**, 32
- CS kommando**, 32
- CT kommando**, 32
- D kommando**, 44
- DA kommando**, 44
- E kommando**, 28
 - E²PROM, 51
 - Elektriske data, 49
 - Error, 24
- F kommando**, 28
- f+/- kommando**, 28
- Flowkommandoer, 44; 45; 46;
47; 48
- Fysiske mål, 50
- g+/-**, 33
- G±A**, 33
- H+/-**, 33
- hukommelsen, 51
- I [1-3] kommando**, 28
- Interfaceforbindelser**, 16; 17
- J kommando**, 44
- JC kommando**, 45; 46
- JCA kommando**, 47
- JS kommando**, 48
- K kommando**, 28
 - Kommandoformat, 20
 - Kommunikationshastighed, 19
- L kommando**, 48
 - Lagre, 22
- M kommando**, 29
 - Modes, 23
 - Motor, 7
 - Motorkommandoer, 31; 32;
33; 35; 36; 37; 38; 39; 40
- N kommando**, 34; 35
- NA kommando**, 36; 37
- PE kommando**, 29
- PO kommando**, 29
- Positionstæller, 22; 28
- PX kommando**, 29
- Q kommando**, 29
- R kommando**, 38; 39; 40
 - Resonans, 9
- RS kommando**, 38
- RT kommando, 38
- S kommando**, 38; 39; 40
 - Stikforbindelser, 15
 - Stop, 28
 - Stopindgang, 13
 - Strømforsyning, 5
 - Systemkommandoer, 28; 29;
30
- T kommando**, 38; 39; 40
 - Temperatur, 30
 - Tilbehør, 52
- TP kommando**, 30
- U kommando**, 41
 - V1 kommando, 30
- V2 kommando**, 42
- VA kommando**, 42; 43
- VR kommando**, 40
- VS kommando**, 40
- VT kommando**, 40
- W kommando**, 43
- X kommando**, 30
- Z kommando**, 30