

RCSL: AA52 -- 099

Author: FLH/EHL

Edited: Maj 1973

SYSTEMBESKRIVELSE FOR
SIM 200 INCL. DRIVERE

Keywords: RC 3500, data transmission, software description

Abstracts: Description of a system for communication with
CDC computers.

A/S REGNECENTRALEN
Guldsmedegade 3,
DK - 8000 Aarhus C.

1 kopi her. Dve.
25-1-73

INDHOLDSFORTEGNELSE

INDHOLD:	KAPITEL
STORAGE-LAYOUT	1.
SCANNER OG CDC-DRIVERE	2.
LOGGER	3.
STATIONS OG SITES	4.
STATION TABLES	5.
DEVICE-DRIVERS	6.
OPSTART AF SYSTEMET	7.
MONITORENS BEHANDLING AF MELDINGER	8.
PASSWORDS	9.
KOMMUNIKATION MELLEM MONITOR OG DRIVERE	10.
FORTEGNELSE OVER INDHOPPENE I DRIVERNE	11.
FORTEGNELSE OVER MONITORENS GLOBALE VARIABLE	12.
DRIVERSTRUKTUR	13.
LISTE OVER BENYTTED E J-NAVNE	14.
DET ASYNKRONE DRIVERSYSTEM	15.
Struktur af det asynkrone driversystem	15.1
Interruptnumre og kommunikationsflow	15.2
Den asynkrone timer	15.3
Opstart af det asynkrone driversystem	15.4
Struktur af drivere, der kører under det asynkrone driversystem	15.5
Struktur af station tables	15.6
Kommunikation mellem tegndrivere, der kører under det asynkrone driversystem og omverdenen	15.7

INDHOLD:

KAPITEL

APPENDIX:

SYSTEMBESKRIVELSE AF RC 810-DRIVER	A1
SYSTEMBESKRIVELSE AF POLLER	A2
Device-driverne	A2.1
Skrive scanner	A2.2
Poller timer	A2.3
Modtage scanner	A2.4
Message niveauet	A2.5
BRUGERBESKRIVELSE AF TELETYPES PÅ CDC VIA RC 3500	A3
GENERERING AF ET PROGRAMSYSTEM PÅ RC 4000	A4
Generel beskrivelse	A4.1
Editoringsprocessen	A4.2
Flere site-tabeller pr. program	A4.3
Eksempel på editering	A4.4

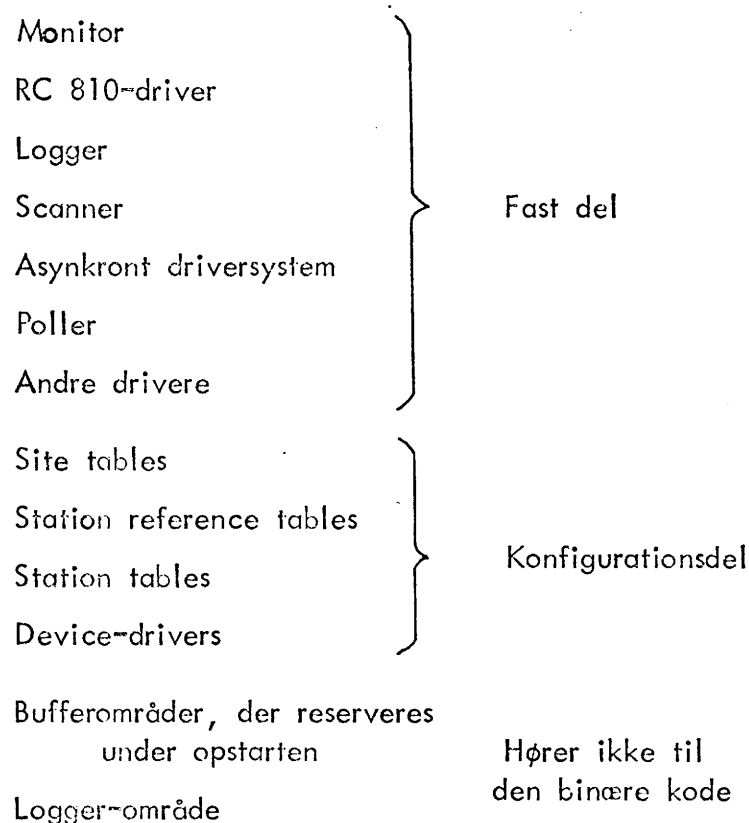
pp. 1 : 56

Programkomplekset består af en fast del og en variabel konfigurationsdel. Den faste del indeholder monitoren og alle driverne. Konfigurationsdelen indeholder konfigurationstabellerne og diverse "device drivere". Disse drivere eksisterer i lige så mange kopier, som der er tilsvarende hardware enheder, hvorimod driverne i den faste del højst eksisterer i én kopi.

Med undtagelse af drivere i det asynkrone driversystem ligger alle drivere sidestillet i monitoren.

Den faste og den variable del udgør tilsammen hele det binære program. Når programmet startes, vil det dog reservere bufferplads efter den binære kode. Når al reservation er sket, vil loggeren tage den resterende ledige lagerplads.

Storage-layout:



Normalt har monitoren intet at lave og kører så rundt i scanneren, hvor den hele tiden scanner de device drivere, der kommunikerer med CDC'en. Når en CDC-driver har afsendt en melding til CDC'en, vil scanneren starte CDC-inputdriveren op og scanne videre. Når en CDC-driver har modtaget en melding, vil scanneren gemme bufferadressen på meldingen i g37. Endvidere sættes g 70, g39 og g34. Derefter hopper scanneren til monitoren, der undersøger meldingen og eventuelt aktiverer driverne. Når dette er sket, kan monitoren enten hoppe til h52 eller h53 i scanneren, der svarende hertil enten starter CDC-output driveren eller CDC-input-driveren. Herefter vil scanneren scanne videre.

Scanneren og CDC-driverne aktiverer loggeren én gang for hver melding, der skal afsendes eller er modtaget.

Med regelmæssige mellemrum vil scanneren aktivere den asynkrone timer og polleren.

Som det er antydnet, foretager CDC-driverne indlæsning og udskrivning af hele meldinger. Ved læsning starter driverne med at synkronisere. Derefter skippes sync-tegn, og derefter forventes et SOH-tegn. Hvis dette ikke kommer, starter driveren forfra med at synkronisere. Ellers indlæses blokken, idet sync-tegn skippes. Ved bufferoverløb vil driveren blot starte forfra med at synkronisere.

Af hensyn til fejlfinding findes en logger, der kan registrere trafikken fra og til CDC'en. Hver eneste gang scanneren modtager en melding fra CDC-driveren, eller der bliver sendt en melding fra monitoren til CDC-driveren, vil loggeren gemme de første n bytes af meldingen i et cyklisk logger afsnit. En typisk værdi for n er 16 bytes, men alle lige værdier kan bruges. Man kan i øvrigt ændre n ved at rette i det oversatte program.

Loggeren bruger følgende variable:

- g44 startadresse for loggerområde
- g45 adresse på cellen efter det sidste ord, der blev gemt i loggerafsnittet
- g46 adresse på første celle efter loggerafsnittet

I hver eneste melding fra CDC'en findes en station adresse og en site adresse. Et par af disse adresser angiver tilsammen en station, der set fra CDC'en opfattes som en skærm eller en 200-user terminal.

En 200-user terminal består fysisk af flere input og output enheder, men monitorsystemet opfatter det hele som én station (ligesom CDC'en gør) og overlader det i nuværende udformning til den tilknyttede driver at skelne mellem de enkelte hardware enheder.

Til hver station findes en station table, der indeholder al nødvendig variabel og fast information om den pågældende station.

I station tablen står bl.a., hvilken driver der varetager administration af den pågældende station; og hver eneste gang, monitoren vil den pågældende station noget, henvender den sig til driveren.

Når CDC'en bruger station address = 0 inden for en site, mener den hele siten. Disse "stations" behandles af monitoren på helt samme måde som normale stations.

Som anført i afsnittet "stations og sites" eksisterer der en station table for hver station. Alle station tables indeholder en fast del og en variabel del.

Den faste del har følgende opbygning:

<u>Relativcelle</u>	<u>Indhold</u>
g11	type: indeholder adressen på beskrivelsen af den driver, der administrerer stationen.
g15	Status: Denne celle kan indeholde status for stationen, men dette forlanges dog ikke af monitoren i dens nuværende udformning.
g13	Last station address: Bruges af monitoren, når den skal svare CDC'en.
g12	Devicenø: Kan eventuelt bruges af driveren.
g14	Interrupt level: Kan eventuelt bruges af driveren.
g16	Type bits: Angiver forskellig information om stationen. Se nedenfor.

I relativcelle g19 og de følgende celler har de enkelte drivere stående al information, der vedrører stationen. Bufferområder til stationerne vil driverne dog som regel reservere under opstarten og så nøjes med at have pegepinde til disse områder stående i station tablen.

Foreløbig er følgende bits defineret i type bits:

- | | |
|--------|--|
| 1 < 0: | OP-skærm, d.v.s. monitoren, sætter passwords efter meldinger startende med $\neq$ R. (Se passwords). |
| 1 < 1: | Async timer aktiv. |

Det bemærkes, at der i monitoren findes specielle med monitoren integrerede drivere, der administrerer non-existent stations og sites (i.e. stations med station address = 0).

Device-driverne findes i device-driverafsnittet i konfigurationsdelen af programkomplekset. Hver af device-driverne eksisterer i et eksemplar for hver eneste RC 3500 deviceno, hvorpå der er koblet enheder hørende til device-driveren.

Endvidere findes dummy device-driverne, der behandles som normale device-drivere; f.eks. findes en dummy device-driver, der starter scanneren. Når device-driverne oversættes, kræver de som regel definitioner indsat umiddelbart foran sig. Definitionerne udføres ved at definere nogle c-navne og kan f.eks. dreje sig om device-numre og itr-numre.

Alle device-drivere skal starte med følgende celler:

<u>Celle</u>	<u>Indhold</u>
0	Adressen + 2 på sidste celle af device-driveren (d.v.s. adressen på næste device-driver).
2	Det interrupt-niveau, som device-driveren hører til.
4	Reset adressen. D.v.s. den adresse, driverens interrupt-niveau skal startes i ved reset (se opstart af systemet).
6	Den adresse, der skal hoppes til ved start-up (se opstart af systemet).
8	Device-type. Denne type bruges af det øvrige system til at afgøre, om device-driveren kommer den pågældende systemdel noget ved.

Den sidste (dummy-) device-driver har følgende start:

<u>Celle</u>	<u>Indhold</u>
0	0
2	0
4	0
6	h 50
8	0

Den bruges til at indikere slut på device-driverne og til at starte scanneren.
Indtil videre er der benyttet følgende device-typer:

- 0: Diverse.
- 2: Multiplekser-driver.
- 4: MSSl-driver hørende til polleren.
- 6:
- 8: CDC-driver.

De forskellige typer device-driverne vil som regel have yderligere fast definerede celler umiddelbart efter det ovenomtalte faste "hoved".

Når reset-indgangen er aktiveret, kan device-driveren reservere bufferplads. Endvidere kan driveren stoppe devicen. Device-driveren skal slutte med:
Ice w0i k.

Når start-up-indgangen aktiveres, skal device-driveren starte devicen. Derefter skal den hoppe til næste device-drivers start-up adresse.

Se i øvrigt afsnittet: Opstart af systemet.

Systemet indlæses til celle 400 oktalt og fremefter. Systemet startes ved at starte et vilkårligt interrupt-niveau i første celle. Systemet kan genstartes på et vilkårligt tidspunkt. Opstarten af systemet forløber i følgende tempi:

1. Alle interrupt-niveauer resettes ved at sætte ordretællerne til at pege på en ordre:

Ice w0i k

Alle device-drivers aktiveres dog i deresreset indgang på det pågældende niveau.

Herunder kan alle device-drivers reservere bufferområder.

2. Alle drivere aktiveres i reset-indhoppet én gang for hver eneste station, der hører til driveren. Herunder kan der reserveres lagerplads til stationerne.
3. Loggerområdet reserveres.
4. Alle device-drivers aktiveres en gang i deres start-up indgang.
5. Scanneren startes, og systemet kører.

Under læsning af dette afsnit bør man hele tiden skæve til fortegnelsen over monitor-indhoppene i driverne. Endvidere skal man kende kommunikationen med f.eks. CDC 711 skærme, da det er dette kommunikationssystem, der benyttes med CDC'en, idet RC 3500 fungerer som low-level-processor.

Når monitoren får en melding fra scanneren, ved den, at meldingen starter med SOH og slutter med ETX, LRC. Testen af meldingen forløber derfor i følgende step:

1. Hvis der er fejl på site-adressen, eller hvis siten ikke er defineret, ignoreres meldingen, og der hoppes til scanneren, der starter CDC-input driveren op på ny.
2. Hvis der er fejl på stationsnummeret, eller hvis stationen ikke eksisterer, svares CDC'en med en nak-melding med station address for siten (d.v.s. 140 eller 160 oktalt).
3. Hvis (i) der er fejl på kontrolkoden, eller hvis (ii) stations adressen er 140/160 i meldinger, der ikke kan sendes til siten (d.v.s. alert- og skrivemeldinger), eller hvis (iii) meldingen ikke er tom i ikke-dataførende meldinger (d.v.s. poll, alert, status request), eller hvis (iv) der er tværgående eller langsgående paritetsfejl på ikke-skrivemeldinger, så sendes en nak-melding til CDC'en.

Monitorens reaktion afhænger herefter af meldingstypen.

4. **Skrivemeldinger:**
Write-indhoppet i den pågældende driver aktiveres. Driveren kan herefter sende nak eller rej til CDC'en ved at hoppe til monitortilbagehoppene nak, henholdsvis rej.

Hvis driveren imidlertid hopper til monitortilbagehoppene:
from write, save message eller return from write,

vil monitoren undersøge meldingen og udføre følgende aktioner:

4a. Hvis der ikke er registreret paritetsfejl på meldingen, eller hvis det er en diagnostic-write, vil monitoren gemme meldingens stationsadresse (og bruge den på efterfølgende meldinger til CDC'en). Derefter vil monitoren i tilfælde af en diagnostic-write svare med en read-melding med samme indhold som write-meldingen. I tilfælde af en ikke-diagnostic-write vil monitoren i stedet svare med en ack.

4b. Hvis der er registreret paritetsfejl, og meldingen ikke er en diagnostic-write, vil monitoren sende en nak til CDC'en.

5. Alert meldinger:

Monitoren vil aktivere alert-indgangen i den pågældende driver. Derefter vil den sende en ACK-melding til CDC'en.

6. Status-Request og Poll-meldinger:

Monitoren vil opbygge en status melding til CDC'en. Herunder vil den aktivere status-indhoppene i de drivere, meldingen vedrører (hvis stations-adressen er 140/160, vedrører meldingen alle stationer på siten).

6a. Hvis meldingen var en status-request, vil den opbyggede melding herefter sendes til CDC'en.

6b. Hvis meldingen var en poll-melding, og ingen af driverne har sat send-request-bit'en i status-ordet, vil monitoren sende en rej-melding til CDC'en.

6c. Hvis meldingen var en poll-melding, og der findes en driver, der har sat send-request-bit'en i status-ordet, vil monitoren i stedet aktivere read-indgangen i den driver, der svarer til den station, der har mindst stationsnummer, og som samtidig havde send-request-bit'en sat.

Den således tilvejebragte read-melding vil herefter blive sendt til CDC'en.

Af hensyn til sikkerhed mod opdateringer i CDC's registre kan systemet i visse meldinger til CDC'en sætte "OP-Passwords".

Disse passwords sættes, hvis følgende betingelser er opfyldt:

1. Type-bits (se station tables) indikerer, at stationen er en officiel OP-station.
2. De første to tegn i meldingen er $\neq$ R.
3. Meldingen er på mindst 20 data tegn.

De første 20 tegn må ikke indeholde ESC-tegn.

De indsatte passwords består af en 10-tegns karakterkombination, der indsættes som data-tegn nr. 11-20 i meldingen.

Karakterkombinationen er fælles for alle skærme og findes i programmet.

Monitoren ligger på itr. niveau 0. Al kommunikation mellem monitoren og driverne foregår ved, at monitoren hopper til forskellige indgange i driveren (d.v.s. den del af driveren, der ligger på itr. niveau 0). Driveren kan herefter (afhængig af den indgang, monitoren benyttede) referere til forskellige variable i monitoren.

Når niveau 0-delen af driveren har udført sit arbejde, skal den hoppe tilbage til monitoren. De mulige tilbagehop afhænger naturligvis også af den indgang, monitoren benyttede. Det pointeres kraftigt, at monitoren's variable kun må benyttes, så længe man ikke har hoppet tilbage til monitoren. Nødvendig information må altså kopieres fra monitoren's variable før tilbagehoppet.

Medmindre andet er skrevet, må driverne hverken ændre monitoren's arbejds-celler eller dens buffer.

Indhop 0: Read

Benyttelse:

Dette indhop benyttes af monitoren, når den vil læse en melding fra en station. Indhoppet vil kun blive foretaget, hvis monitoren ved et tidligere statusindhop (indhop 6) har fået et statusord med send-request-bit'en sat lig 1.

Funktion:

Driveren skal tilvejebringe en melding i monitorens buffer (der i øvrigt er adresseret med g37). Det er tilladt for driveren at bytte monitorens buffer ud (ved at ændre g37) med en buffer, der er lige så lang (og hvor cellerne 0-8 er hentet fra monitorens oprindelige buffer).

Tilbagehop:

1. GOTO Store-Station:

ldr w0i h54

Ved dette tilbagehop vil monitoren i bufferen sætte korrekt:

Korrekt lrc

Address of lrc

Station address

paritet på alle tegn (fra og med soh, til og med lrc),

endvidere vil monitoren evt. sætte OP-passwords (se Passwords).

Alt andet i bufferen skal være i orden før tilbagehoppet.

2. Andre tilbagehop er mulige, hvis man selv sørger for at få samme virkning som i punkt 1.

Indhop 2: Write

Benyttelse:

Dette benyttes af monitoren, hvis der er kommet en skrivemelding, også selv om der er fejl i meldingen ($g39 \neq 0$).

Funktion:

Driveren skal skrive meldingen på stationen, medmindre den ønsker at rejecte den eller skippe den. Som ved Read (indhop 0) er det tilladt at bytte buffere. Det er tilladt driveren at provokere fejl i write-meldingen ved at sætte $g39 \neq 0$. Dette vil få betydning for tilbagehop 1 og 2.

Tilbagehop:

1. GOTO Return-from-write:

ldr w0i h55

Dette er det normale tilbagehop.

2. GOTO From-write, save-message:

w3:= adresse på ledig buffer.

ldr w0i h61

Hvis driveren vil bytte buffer med monitoren, kan dette tilbagehop benyttes. G37 skal være uændret ved tilbagehoppet. Den ledige buffer skal have samme længde som monitorens buffer, men indholdet i den ledige buffer er ligegyldigt, da dette kopieres fra monitorens egen buffer. Til sidst vil monitoren gemme w3 i g37, hvorved bytningen af bufferne er tilendebragt.

3. Goto REJ:

ldr w0i h58

Dette tilbagehop bruges, hvis en driver vil rejecte en skrivemelding.

Indhop 4: Alert

Benyttelse:

Dette indhop benyttes af monitoren, når den har modtaget en korrekt Alert-melding fra cdc'en. Sådanne meldinger kan komme fra cdc'en, hvis den vil af med output. Normalt vil cdc'en herefter vente med at skrive, indtil den har læst en eller anden melding fra den samme station.

Funktion:

Driveren kan eventuelt sætte sendrequest-bit'en til 1 og senere svare med en tom read-melding (efter et read-indhop).

Tilbagehop:

1. GOTO ACK :
ldr w0i h56

Indhop 6: Status

Benyttelse:

Dette indhop benyttes, når monitoren vil hente status fra stationen, eller hvis den eventuelt vil læse fra den (med indhop 0, Read).

Funktion:

Driveren skal generere et statusord med følgende betydning:

bit 15: 1 = power ok
bit 14: 1 = send request
bit 9: 1 = ready
bit 10-13 og 0-8: 0

Tilbagehop:

GOTO RETURN ADDRESS:

w3:= status

ldr w0, #40

Indhop 8: Release keyboard

Benyttelse:

Dette indhop vil altid benyttes umiddelbart efter et write-indhop, hvis dette blev besvaret med returhop 1 eller 2, og hvis samtidig g39 var 0 ved tilbagehoppet, eller hvis det samtidig var en diagnostic-write.

Funktion:

Når hele write-meldingen er blevet udskrevet på stationen, skal dettes tastatur låses op.

Tilbagehop:

1. GOTO RETURN ADDRESS:

ldr w0 g40

Indhop 10: Reset Device

Benyttelse:

Dette indhop benyttes af monitoren én gang for hver station ved hver programopstart (d.v.s. når programmet er blevet loadet eller ved genstart).

Funktion:

Driveren skal initialisere de variable, der hører til stationen. Bufferplads kan reserveres ved at tælle g44 op med det ønskede antal bytes (altid et lige antal). g37 vil under opstarten hele tiden pege på første ledige celle i det lagerområde, hvori bufferområderne udlægges.

Tilbagehop:

GOTO RETURN ADDRESS:

ldr w0 g40

I følgende tabel betyder et kryds, at variabelen må bruges. Et æ betyder, at den både må bruges og ændres.

<u>Indhop nr.</u>	<u>Navn</u>	<u>Betydning</u>
0 2 4 6 8 10		
+ + + +	g30	site address (a. 8.377)
+ + +	g31	station address (a. 8.377)
+ + + +	g32	control code (a. 8.177)
+ + + +	g33	control code bits
æ æ	g37	buffer address
æ	g39	parity error (0 = ok)
+ + +	g40	return address
+ + + + + +	g43	station table address
æ	g44	global buffer

"Buffer address" peger på første celle af monitorens buffer, der i øvrigt har længden g0.

"Station table address" peger på celle 0 i den tabel, der hører til beskrivelsen af stationen.

Monitorens buffer har følgende opbygning:

0: Adresse på lrc	}	Administreres af monitoren
2: SYNC SYNC		
4: SYNC SYNC		
6: SOH Station address		
8: Site address Control code evt. texttegn		
Blokslut: ETX, LRC		
eller: EXT		
LRC		

Hver eneste driver skal være omgivet af en blok og have en driver-beskrivelse. Denne driverbeskrivelse skal være placeret inde i driveren og skal have følgende udseende:

jX: e0, e2, e4, e6, e8, e10,

hvor jX er en j-etikette (se g11 i afsnittet om station tables). Efter j-etiketten følger adressen på de seks monitorindhop i driveren (eY svarer til indhop Y).

Denne liste skal ajourføres for hver ny driver.

j0	non existing device
j1	site (i.e. station address = 0)
j2	RC 810-driver
j3	TTY-driver
j4	Poller
j5	RC 811-driver
j6	
j7	
j8	
j9	
j10	
j11	TTY-char-driver
j12	RC 811-char-driver
j13	
j14	

Alle drivere, der skal kommunikere med enheder, der er koblet på RC 3500 via asynkrone multipleksere, skal køre under det asynkrone driversystem.

Dette system formidler kommunikationen mellem multiplekserne og driverne. Endvidere indeholder systemet en timer, der for hver enkelt device kan sende time out til driverne, når der er gået et vist stykke tid, siden timeren for den pågældende device blev resat.

Det asynkrone driversystem er på enkelte punkter integreret med monitoren (async timer, g-navne, opstart).

Systemet består af en kopi af følgende programdele:

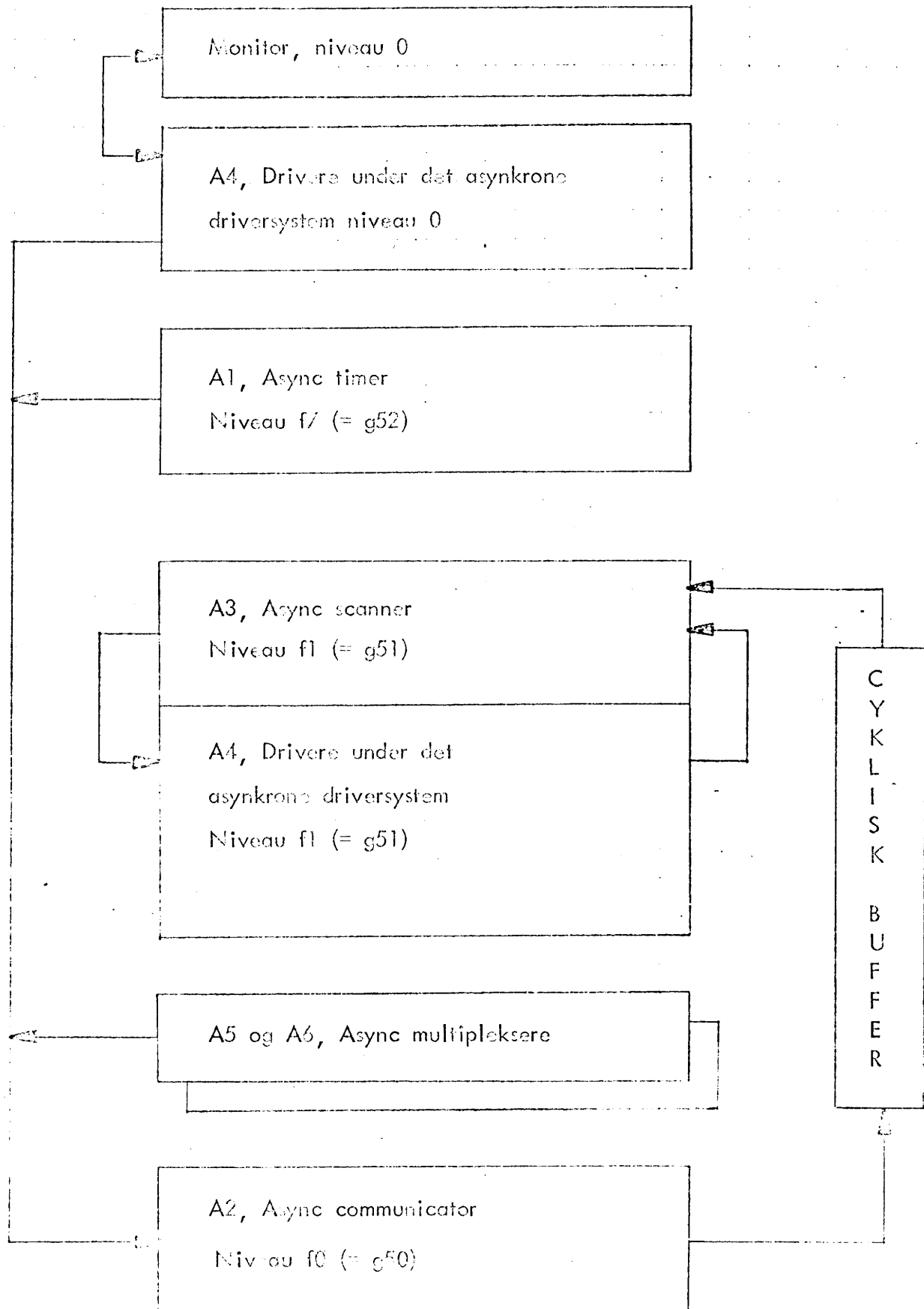
- A1. Async timer
- A2. Async communicator
- A3. Async scanner
- A4. Drivere, der kører under systemet

Endvidere er der lige så mange kopier af følgende device drivere, som der er tilsvarende hardware-multipleksere:

- A5. Async bit mux
- A6. Async char mux

Leksikografisk placeres A1-A4 i samme blok i monitorens permanente driver-afsnit. A5 og A6 placeres blandt monitorens øvrige devicedrivere i den variable konfigurationsdel. Hver driver i A4 står logisk set på lige fod med alle andre drivere, der kører under monitoren. De har således en niveau 0-del (til kommunikation med monitoren). Endvidere skal de have en tegn-niveau-del, der først og fremmest varetager udskrift og indlæsning af tegn. Af hensyn til udelelighed af funktioner i niveau 0-delen er der endvidere åbnet mulighed for, at niveau 0-delen kan aktivere tegn-niveau-delen, når udelelige funktioner skal udføres. Da tegn-niveauet kun kører i én inkarnation, vil funktioner i tegn-niveauet nemlig foregå udeleligt.

Som sædvanlig bør langvarige funktioner ikke udføres i tegnniveauet, men i så lave interrupt-niveauer som muligt.



A5 og A6 skal have interruptnumre mellem g51 og g50. Disse interrupt-numre skal i øvrigt naturligvis svare til de interrupt-niveauer, multiplerne reelt er koblet på.

Kommunikationen fra A2 til A3 foregår ved hjælp af elementer i en cyklisk buffer. Ved buffer-overløb går systemet i dets nuværende udformning i bag-lås, men det vil dog være muligt blot at skippe information fra A5 og A6 (A1 og A4 vil ikke kunne forårsage buffer-overløb, da de kun kører, når den cykliske buffer er tom, da A3 ellers spærrer for dem).

Hvert element i den cykliske buffer består af 2 celler og har følgende opbygning:

celle 1:	station table address
celle 2:	input char and function

Celle 2 skal normalt indeholde funktionen i position 15. Funktionen skal altid være lige. Hvis celle 2 indeholder et tegn, skal dette stå i position 0-7, og position 15 skal være et-stillet. Den asynkrone scanner vil så opfatte det som funktion 0.

Den asynkrone timer er i den nuværende udformning en programmeret timer uden nærmere specificeret time-out tid. Detaljer vil blive fastlagt, når behovet viser sig.

For at den asynkrone driver i det hele taget skal være aktiv, skal "type-bits" i station tablen have bit 1 < 1 et-stillet.

Først aktiveres monitorindhop nr. 10 (reset) for hver eneste driver. Dernæst startes multiplekserne. Derefter startes den asynkrone timer. Til sidst startes cdc-driveren, således at monitorhoppene 0-8 kan blive brugt.

Niveau 0-delen har den sædvanlige struktur for drivere.

Tegndriver-delen skal indeholde en j-etikette det sted, hvortil den asynkrone scanner skal hoppe. Værdien af j-etiketten er i øvrigt placeret i hver eneste station table (relativ celle f19), der hører til driveren.

Niveau 0-delen og tegn-niveau-delen skal være placeret i en blok inden i det asynkrone driversystem (af hensyn til f-navnene, der er lokale til det asynkrone driversystem).

Foruden den faste del, der forlanges af monitoren, bruger det asynkrone system følgende celler:

f12 = g12 ; deviceno
g16 ; type bits
f11 = g19 + 0 ; mux type
; 0 = bit mux
; 2 = char mux
f13 = g19 + 2 ; mux address no
f18 = g19 + 4 ; counts pr time-out
; = 100
f19 = g19 + 6 ; char driver, entering address
f14 = g19 + 8 ; partial input char
(only for bit mux)
f15 = g19 + 10 ; partial output char
(only for bit mux)
f16 = g19 + 12 ; time count
f17 = g19 + 14 ; time-out limit

Den del, der bruges af driverne, skal være placeret i f20 (= g19 + 16) og de følgende celler.

Al kommunikation til disse drivere foregår gennem den asynkrone scanner.

Ved indhop er:

w3 = station table address

w2 = function

w1 = input char (kun ved function 2)

Function kan være:

0, når multiplekseren er færdig med at skrive sidste tegn ud

2, når multiplekseren har læst et tegn ind

4, ved time-out

Endvidere kan andre værdier af function benyttes ved kommunikation fra niveau 0-delen af driveren til tegndriveren.

Når tegndriveren er færdig med at behandle functionen, skal den hoppe tilbage til den asynkrone scanner ved hjælp af følgende tilbagehop:

1. Write Return:
w3 uændret
w2 := output char
ldr w0i f52
2. Reset Timer Return:
w3 uændret
ldr w0i f51
3. Dummy Return:
w3 uændret
ldr w0i f50
4. Reset Return:
w3 uændret
ldr w0i f53

Tilbagehop 3 er det normale tilbagehop. De andre tilbagehop vil altid resette timeren. Tilbagehop 1 benyttes, når der ønskes skrevet et tegn. Dette tilbagehop må kun benyttes, hvis der siden sidste tilbagehop 1 eller 4 er modtaget en funktion 0 eller 4. Tilbagehop 4 skal benyttes én gang for hver station, når programmet startes op (d.v.s. ved monitor-indhop 10, reset device).

Kommunikation fra Niveau 0 til tegndriveren kan ske på følgende måde:

```
start:      ldr w1i e1
            ist wls x0 f0

e0:         ---
            ---
            ---

e1:         w3:= station table address
            w2:= function
            ldr w0i f54
```

eller tilsvarende.

Driveren i niveau 0 vil vente i e0, indtil tegndriveren har udført funktionen (og evt. andre funktioner 0, 2 eller 4).

Låsning af tastatur og send request

RC 810 virker sådan, at operatøren kan danne et skærbillede uden medvirken af RC 3500. Når operatøren er færdig, trykker han på en send-knap, og først på dette tidspunkt sendes interrupt til RC 3500. RC 810-driveren vil her- efter sætte send-request-bit'en i statusordet og disable tastaturet, herunder send-knapperne.

Endvidere disables tastaturet, når RC 3500 begynder at læse fra skærmen, eller når en skrivemelding er ankommet til den.

Efter en skrivemelding med korrekt paritet enables tastaturet, hvis send-request-bit'en ikke er sat, eller hvis det ikke var en almindelig write.

Send-request-bit'en slettes altid efter hver eneste skrivemelding (også selv om meldingen ikke skrives på skærmen, f.eks. på grund af paritetsfejl).

Skrivemeldinger

En skrivemelding kommer kun ud på skærmen, hvis pariteten er ok. Hvis meldingen er en almindelig write, og hvis send-request-bit'en samtidig er sat, kommer meldingen dog alligevel ud.

En clear-write med korrekt paritet starter med at slette hele skærmen.

Clear-write, Reset-write og Diagnostic-write vil altid starte med at skrive oppe i venstre hjørne, medens almindelig write vil starte skrivningen, hvor sidste skrivning eller læsning holdt op.

Under skrivning på skærmen vil driveren skrive 64 tegn, derefter skippe 16 tegn o.s.v. Efter NL startes dog forfra med tællingen.

Der foretages følgende tegnkonverteringer (oktaltalværdier):

137	→	37
37	→	137
12	→	NL
33 + 101	→	NL
33 + 42	→	Skip
33 + 3	→	ET X
33 + andet	→	137
3	→	ET X

0-2, 4-11, 13-32, 34-36, 177 → 137.

Andre tegn konverteres normalt ikke. Under oversættelsen af driveren kan man dog definere, at små bogstaver skal konverteres til store (140-176 → 100-136).

Alert meldinger

Disse meldinger besvares blot med ack.

Status-request meldinger

Statusordene har følgende opbygning (binært): 10000x1, hvor x angiver send-request-bit'en.

Læsemeldinger

Ved læsning startes altid i øverste venstre hjørne. Hvis der er trykket på "send én linie", læses kun 64 tegn. Ellers læses 1024 tegn. I begge tilfælde stopper læsningen dog, hvis der læses et tegn med oktaltærdi = 76 (d.v.s. >).

I øvrigt foretages følgende tegnkonverteringer (oktaltærdier):

137	→	40
0-37, 177	→	137
140-176	→	100-136
76	→	stop

Resten konverteres ikke.

For at forstå dette system skal man kende CDC 711.

Polleren er simpelthen en CDC 711 driver.

Pollersystemet består af følgende programdele:

- A. Message niveau
- B. Poller timer
- C. Modtage scanner
- D. Skrive scanner
- E. Device-drivere

Del A foretager kommunikationen med monitoren. Med mellemrum vil scanneren aktivere pollersystemet, der herefter aktiverer programdelene B, C og D. For at der ikke skal bruges for lang tid i træk polleren vil C, D og E dog returnere til scanneren efter en langvarig operation. Programdel E startes af D og sender en melding af sted til en CDC-skærm og venter derefter på svar fra skærmen. Når E er færdig, vil svaret undersøges af C, der blot registrerer, hvad der er opnået ved svaret.

Hver device-driver har en lille buffer på 30 tegn, hvori den kan sende og modtage korte meldinger. Hver station under polleren har derudover en stor buffer, hvori der kan stå read- og write-meldinger. Device-driveren virker som slave for det øvrige system.

Den har en semaphore, der angiver:

- 0: ikke aktiv.
- 1: aktiv, d.v.s. den er i en skrive/læse cyklus.
- 2: den har fra CDC-skærmen modtaget en melding,
der endnu ikke er undersøgt af C.
- 3: timeren har afbrudt device-driveren, og dette er endnu
ikke registreret af C.

Semaphoren benyttes af det øvrige system.

Når device-driveren skal læse en melding, tester den blot, at denne starter med soh og slutter med etx. Herudover registrerer den paritetsfejl, men reagerer ikke på en eventuel paritetsfejl.

For ikke at overbelaste RC 3500 har hver device-driver endvidere en boolean, der angiver, om D må starte device-driveren.

Denne programdel scanner device-driverne, og hvis den finder en driver, der ikke kører (d.v.s. hvis semaphore er lig 0), og hvis denne driver må køre, så vil D udføre følgende forsøg på at starte device-driveren:

I device-driveren står en reference ind i "station ref table". Denne angiver, hvilken station driveren sidst kørte for (referencen kan eventuelt være til starten af tabellen). Denne reference tælles frem, indtil der findes en station, der hører til device-driveren. Hvis en sådan findes, inden tabellen er slut, vil D starte driveren op for denne station (se nedenfor). Ellers vil referencen resettes til starten af tabellen, og D vil fjerne device-driverens indikation om at måtte køre. Herefter vil D endvidere scanne device-driverne cyklisk, indtil den finder en device-driver, der hører til polleren, og som ikke må køre. Denne driver vil nu få sat en indikation om at måtte køre. Ethvert af disse cykliske scan starter, hvor det forrige slap, hvorved der sikres en ret ligelig tildeling af køretid til hver station.

Opstart af device-driverne:

Den melding, der skal sendes til CDC-skærmen, afhænger af kontrolvariablerne i stations-tabellen.

Allerførst slettes "last was write". Derefter undersøges, om "det stat addr" er sat. Hvis dette er tilfældet, sendes en status request melding.

Hvis denne boolean ikke var sat, men "out pending" er sat, sendes i stedet den skrive-melding, der står i stationens buffer. Samtidig sættes "last was write".

Hvis heller ikke denne boolean var sat, men "alert pending" er sat, sendes en alert melding.

Hvis ingen af ovenstående tre booleans var sat, sendes en station-poll-melding, hvis send-request bit'en ikke er sat.

Samtidig med, at device-driveren bliver startet op, sættes dens semaphore til 1, og dens "time count" sættes til den værdi, der er angivet i stations-tabellen.

Hver n'ite gang denne aktiveres, vil den for hver device-driver hørende til polleren tælle "time count" ned med 1, hvis dens semaphore er lig 1. Hvis tælleren derefter er blevet nul, vil B resette device-driveren og sætte dens semaphore til 3.

I programmet er n sat til 100, men denne værdi kan eventuelt ændres.

Denne programdel scanner device-driverne, indtil den finder en driver hørende til polleren, og hvis semaphore er lig 2 eller 3.

Hvis semaphore er 3, eller hvis stations og site adresserne er forkerte, indikeres fejl på meldingen.

1. Hvis der er fejl på meldingen, hoppes til step 8. Hvis "last was write" er sat, sættes endvidere "det stat addr".
2. Hvis "det stat addr" er sat, sammenlignes den afsendte meldings stations adresse med den modtages. Hvis de er ens, slettes "outpending". Endvidere slettes altid "det stat addr".
3. Hvis kontrolkoden er nak eller rej, hoppes til step 8.
4. Hvis "last was write" er sat, slettes "outpending", og der hoppes til step 8.
5. Hvis kontrolkoden i den afsendte melding var status-request, hoppes til step 8.
6. Hvis kontrolkoden i den afsendte melding var alert, slettes "alert pending", og der hoppes til step 8.
7. Hvis kontrolkoden i den læste melding er read, slettes "alert pending", og "send request" sættes.
8. Under alle omstændigheder sættes "ready" :=, "outpending". Og semaphore sættes til 0.

Denne programdel varetager kommunikationen med monitoren (se "kommunikation mellem monitor og drivere").

Read:

Stationens buffer kopieres blot over i monitoren.

Write:

Hvis der er paritetsfejl på meldingen, eller hvis det er en diagnostic-write, returneres blot til monitoren.

Hvis "busy" er sat, hoppes til rej-tilbagehoppet. Herefter cleares "send request" og "ready".

Hvis der er en device-driver, der kører for stationen, resettes den, og semaphoren sættes til 0.

"outpending" sættes, og "alert pending" cleares.

I den stationsadresse, der står i stationstabellen og bruges i hver melding til CDC-skærmen, ændres den bit, der angiver intervallerne 141-157 og 161-177.

Herefter kopieres monitoren buffer ind i stationens buffer, idet der dog indsættes nogle sync efter kontrolkoden. Til sidst returneres til monitoren.

Alert:

Hvis "read request" ikke er sat, sættes "alert pending".

Derefter returneres der til monitoren.

Status:

Virker efter forskrifterne.

Release:

Der returneres efter at have clet "send request".

Reset:

Initialiserer stations tabellen; herunder sættes stationsadressen til intervallet 141-157.

Når teletypes kobles på CDC'en via RC 3500, vil CDC'en opfatte dem som CDC-skærme. RC 3500 sørger for at samle et "skærbillede" op fra TTY'en og sende dette til CDC'en. Ligeledes modtager RC 3500 et skærbillede fra CDC'en og trykker det på TTY'en. Se i øvrigt beskrivelsen af intercoms behandling af CDC-skærmene.

Skærmstørrelse

For at spare på bufferpladsen i RC 3500 er skærlængden sat til én linie á 80 tegn. Hvis der sendes mere end én linie fra CDC'en, vil RC 3500 acceptere billedet, men kun sende de første 80 tegn ud på TTY'en (idet evt. linieskift-karakter ignoreres).

For at få CDC'en til at nøjes med at sende én linie ud ad gangen, er der lavet et Intercom program, der kan definere "skærmen" om til kun at bestå af én linie. Programmet kaldes ved i commandmode at skrive:

size CR.

Herefter vil size bede om antallet af linier, og man taster: 1 CR, hvorefter CDC'en kun sender én linie ad gangen.

Linieadministration

For at CDC'en ikke for hvert billede skal vente, til man sender en tom blok til den, er der lavet en automatisk svarmekanisme i RC 3500, som selv genererer sådanne blokke. Ligeledes er der lavet en editeringsmekanisme, der administrerer linieskift, således at der ikke kommer alt for mange blanke linier hidrørende fra de kvitteringer, CDC'en sender.

Outputmeldinger

Hvis CDC'en kommer med et billede til TTY'en, vil dette billede altid komme ud på TTY'en, og RC 3500 vil glemme alt om et eventuelt billede, man er ved at taste ind over TTY'en, eller som endnu ikke er sendt til CDC'en.

Hvis man imidlertid interrupter RC 3500, medens den er ved at skrive på TTY'en, vil den ovenomtalte automatiske svarmekanisme ikke træde i funktion, når billedet er skrevet ud, og CDC'en vil så vente på TTY'en.

Repetition af tegn

I modsætning til CDC'en sender RC 3500 alle indtastede og accepterede data-tegn tilbage til TTY'en. Denne skal altså være forberedt på full-duplex drift. Operatøren kan således se, om RC 3500 har modtaget tegnene eller skippet dem (f.eks. hvis den endnu ikke havde fået sendt sidste linie til CDC'en).

Hvis man forsøger at skrive linier på mere end 80 tegn, vil RC 3500 skippe alle efterfølgende tegn, der ville bevirke, at linien blev på mere end 80 tegn. Man kan i så fald f.eks. slette hele linien eller sende den til CDC'en.

Tegnbehandling

For at betjeningen af TTY'erne skal ligne betjeningen af TTY'er koblet direkte på CDC'en, vil RC 3500 administrere interrupt på samme måde som CDC'en. Endvidere administreres følgende tegn på samme måde som hos CDC'en:

CTRL H, SHIFT O, CTRL X

CTRL Z (herunder A, S, CR)

*) CR, LF, SPACE

Herudover vil alle tegn med oktalværdier 40-136 opfattes som alfanumeriske tegn og små bogstaver (oktalværdier 140-176) konverteres til tilsvarende store bogstaver. Alle andre tegn skippes.

Strimler

Driveren er ikke designet til at kunne læse strimler.

*) RC 3500 vil dog altid svare med LF, CR, DC 3, når man indtaster LF eller CR.

Generel beskrivelse

A4.1

Alle programkomplekser består som nævnt i afsnittet "Storage Layout" af en fast del og en variabel konfigurationsdel.

Programteksten dannes ved editering ud fra to tilsvarende tekster, der foreligger på hulstrimmel i flexowriterkode.

Den første tekst indeholder monitor og dele af samtlige drivere, som er programmeret til systemet.

Den anden strimmel, "basiskonfigurationsdelen", indeholder de forskellige devicedrivere til systemet. Hver devicedriver udgør en programblok.

Til hjælp for editeringen er der før og efter hver driver-blok indsat en kommentar til at søge på.

Kommentarerne, som er indsat før driverteksten, har følgende form:

```
; begin <drivernavn>  
    og efter driveren:  
; end <drivernavn>
```

Listen over drivernavne er følgende:

(Der benyttes udelukkende små bogstaver i søgeteksterne).

cdc 48k baud driver

cdc 2400 baud driver

async multiplexer

sync drivers

Til sidst på konfigurationsstrimmelen findes to dummy drivere:

- a) Driver til opstart af det asynchrone driversystem
- b) Driver, der starter hele systemet ved at starte CDC-scanneren

Derefter følger programafslutningen.

Hele denne sidste del, som altid skal indgå i systemet, indledes med søgetexten:

```
; begin text 2
```

og afsluttes med texten:

```
; end text 2
```

Hele basiskonfigurationstabellen udgør en blok, hvori navnene: a20, b20, c99, e99 er erklærede.

Efter erklæringerne følger en tabel af a-navne, sitetabellen:

```
g5:    a0, a1, a2, a3, a4, a5, a6, a7, a10, a11, a12, a13, a14, a15,  
        a16, a17, - 1.
```

Denne første del af konfigurationstabellen indrammes af søgeteksterne:

```
; begin configurationtable text 1
```

```
; end configurationtable text 1
```

og den skal altid inkluderes i systemet.

a-navnene er adresser på stationreferencetabeller, som indeholder adresserne på systemets stationtabeller. (Hver device adresseres fra CDC-maskinen ved et site- og et station-nr. Se yderligere under beskrivelsen af site og stations).

Site 0 refererer til a0, site 1 til a1 o.s.v.

Der benyttes lige så mange a-navne, som der er sites koblet på systemet. De ikke benyttede a-navne får af programmet tildelt værdien 0.

Hvert a-navn er adressen på en stationreferencetabel, som har følgende form:

```
      8  
ax:   bx  
      ex1  
      ex2  
      ex3  
      .  
      .  
      .  
      ex8
```

exi er adressen på station xi (adresseret ved siteno. x, stationno. i). Det er muligt at adressere 16 stations pr. site, men 8-tallet øverst i tabellen angiver, at der kun er afsat plads til 8 adresser. Dette er normalt rigeligt, men kan i øvrigt vælges efter behov.

(I øjeblikket er det kun muligt fra CDC-maskinen at adressere én station pr. site).

For de ikke benyttede stations indsættes e8, som er adressen på en stationtabel for et dummy (not existant) device. (Se eksemplet bilag 1).

bx refererer til station 0, der som nævnt i beskrivelsen refererer til alle stations under site x.

Station table-formater:

(Se beskrivelsen af station table og det asynchrone driver-system side 8).

jx er indhopsadresser til de tilsvarende drivere i den faste del.

Site (station 0)

bx: j1, 0, 0, 0 ; site x

RC 810-skærme:

ex: j2, 1, 0, 0, <deviceno.>, <itr.level>

TTY:

ex: j3, 2, 0, 0, <deviceno.>, <itr.level>,
<multiplexertype>, <address.no>,
<counts pr. timeout-interval>, j11, 0, r.20

CDC 711-skærme:

ex: j4, 0, 0, 0, <deviceno.>, <interrupt level>,
<site adr.><station adr.><buffer length>,
<counts pr. timeout-interval>, 0, r,4

non existant device:

e8: j0, 0, 0, 0

I den følgende beskrivelse refereres ofte til et eksempel bilag 1, hvor en programtext er editet ved hjælp af RC 4000's editor.

I den første text fil (den faste del) skal der defineres nogle navne, som bestemmer systemet.

På det sted i programmet, hvor disse definitioner skal indføres, findes følgende søgetext:

```
; define system
```

Følgende navne skal defineres:

g90 = bitmønster, der definerer, hvilke drivere man vil lade indgå i det oversatte program:

bit 15 = 1 => konverter til små bogstaver

= 0 => ingen konvertering

bit 14 = 1 => inkluder logger (til testoutput)

bit 13 = 1 => inkluder RC 810 driver

bit 12 = 1 => inkluder TTY driver

bit 11 = 1 => inkluder poller driver (til CDC 711-skærm)

g0 = bufferlængde i bytes

(Buffer til CDC-messages, g0 = 1500 er normalt benyttet)

g48 = topadresse i bytes

(Angiver lagerstørrelsen i moduler ca 8192 bytes)

Hvis TTY-driveren er inkluderet, skal følgende navne defineres:

g50 = Async communicator level

(Dette interruptniveau skal ligge over de niveauer, som de asynchrone multiplexdrivere kører på. I eksemplet bilag 1 er niveau 21 benyttet).

g51 = async scanner level
(Niveauet skal være lavere end de asynchrone multiplexer-drivers.
I eksemplet er g51 = 18).

g52 = async timer level
(Niveauet skal ligge under scannerniveauet. I eksemplet er g52 = 17).

Hvis CDC 711-driveren er inkluderet, skal også g64 defineres:

g64 = run claim
(I eksemplet er g64 = 3).

Efter at disse definitioner er foretaget, kopieres resten af den faste del.

Derefter følger konfigurationsdelen:

1. Først kopieres konfigurationstabeltexten fra
 ; begin configuration table text 1
 til
 ; end configuration table text 1
2. Derefter indsættes stationreferencetabellen efterfulgt af station-
 tabellerne

Den første får følgende form (se eksemplet):

```
g8:            ; stationreferencetable (g8 er global)
8, a0:        b0, e0, e8, r.7
8, a1:        b1, e1, e8, r.7
8, a2:        b2, e2, e8, r.7
          :
          :
8, ax:        bx, ex, e8, r.7
0            ; end stationref.table
```

Tabellen skal afsluttes med 0 (nul).

Stationtabellerne får følgende form:

```
e8:   j0, 0, 0, 0           ; non existing device
b0:   j1, 0, 0, 0           ; site 0
e0:   j2, 1, 0, 0, 5, 2     ; RC 810 skærm med deviceno. 5,
b1:                                   ; interruptniveau 2 og type 1,
e1:                                   ; d.v.s tegn-orienteret
:
b3:   j1, 0, 0, 0           ; site 3
e3:   j3, 2, 0, 0, 3, 20, 0, 1, 1000, j11, 0, r.20; TTY-
                                   ; driver med deviceno. 3, inter-
                                   ; ruptniveau 20, multiplexer-
                                   ; adresse 1 og type 2, d.v.s.
                                   ; bitorienteret, timeout count = 1000

b4:   j0, 0, 0, 0
e4:   j3, 2, 0, 0, 3, 20, 0, 2, 1000, j11, 0, r.20
:                                   ; TTY koblet på multiplexeradresse 2
:
b7:   j1, 0, 0, 0           ; site 7
e7:   j4, 0, 0, 0, 4, 22, 0, 1, 1300, 1000, 0, r.4
:
:                                   ; CDC 711- driver, deviceno. 4,
:                                   ; interrupt level 22, site adr. = 0
:                                   ; station adr. = 1, bufferlængde =
o.s.v.                               ; 1300 bytes, timeout count = 1000
```

Derefter skal devicedriverne kopieres ind. Der skal altid findes en driver til kommunikation med CDC-maskinen.

Man søger forfra i configurationstabellen til søgetexten med navnet på driveren.

Foran første driver sættes adressen

g49: ; adressen på device driveren.

Derefter defineres de nødvendige c-navne.

Hvis 2400-baud-driveren benyttes, defineres:

```
c0 = <deviceno.>
c1 = <interrupt level>
c2 = 0 ; d.v.s. half duplex (c2 = 1 for
        ; full duplex)
c5 = g5 ; adresse på sitetabellen
```

2400-baud-driveren er en bitorienteret driver, som benytter samme interrupt-niveau for input og output.

48K-baud-driveren benytter to interruptniveauer, et for input og et for output. Driveren er byte-orienteret.

Følgende c-navne skal defineres:

```
c0 = <deviceno. for input linien>
c1 = <interruptniveau for input linien>
c2 = <deviceno. for output linien>
c3 = <interruptniveau for output linien>
c5 = g5 ; adresse på sitetabel
```

Efter disse definitioner kopieres drivertexten.

Hvis man benytter flere kommunikationskanaler til CDC, kopieres lige så mange drivere (på hvert sit interruptniveau).

Den asynchrone multiplexerdriver benyttes, hvis der er koblet teletypes på systemet. Det er en bitorienteret driver.

Der søges forfra frem til søgetexten.

```
; begin async multiplexer
```

Derefter defineres følgende c-navne:

I navne c0-c7 overføres adressen på de tilhørende stationtables. De ikke benyttede c-navne sættes lig 0 som i vort eksempel:

```
c0 = 0, c1 = e3, c2 = e4, c3 = 0, c4 = 0, c5 = 0, c6 = 0, c7 = 0
c11 = 3 ; devicno. for multiplexer
c12 = 20 ; interruptniveau for multiplexer
```

Derefter kopieres drivertexten.
099

Er der flere multiplexere, gentages proceduren.

Den synchrone poller-bitdriver benyttes af CDC 711-skærme.

Man søger frem til teksten:

```
; begin sync drivers
```

Derefter indsættes definitionerne:

```
c0 = <deviceno.>
```

```
c1 = <interruptniveau>
```

```
c2 = 0                                ; d.v.s. half duplex (idet  
                                      ; full duplex for c2 = 1)
```

Derefter kopieres drivertexten.

Som afslutning på programmet kopieres de to dummy drivere mellem

```
; begin text 2
```

```
  og
```

```
; end text 2
```

Disponering af interruptniveauer

Systemets monitor kører på niveau 0, og timeren beslaglægger niveau 1.

Transmissionslinierne til CDC skal kobles på interruptniveauer over de øvrige drivere i systemet. (48K-baud linien beslaglægger to niveauer).

Drivere til CDC 711 skal ligge på de næsthøjeste niveauer.

Derunder kommer niveauerne for de asynchrone multiplexere. Disse benytter et niveau til tegnbehandlingsprogram, som ligger over multiplexerniveauet (men under CDC 711-niveauerne), og to lavere niveauer, et til scanner og et til timer for multiplexerdriverne.

RC 810-skærm-driverne skal ligge på de laveste interruptniveauer.

Hvis man benytter strimmellæser til programindlæsning, kan det være bekvemt at reservere niveau 31 hertil.

Hvis man bruger flere cdc-drivere og ønsker, at de skal have hver sin site-tabel, kan dette opnås ved at ændre site-tabellen i "basiskonfigurationsdelen".

Site-tabellen skal blot bestå af: Et eller flere "site-afsnit" adskilt med tallet-2. Efter site-tabellen skal der som afslutning stå tallet - 1.

Hvert "site-afsnit" skal bestå af 16 "station refs" (for site 0-15).

Hvis en site ikke er defineret, skal den pågældende "station ref" være lig 0. Ellers skal den som sædvanligt pege ned på den tilhørende "station ref table".

Der må kun findes én "station ref", der peger på en given "station ref table".

Exempel på editering af et programkomplex.

; RC 3500 config. 7 med 2x200 baud tilslutning til CDC

rcpr1 = set 100

rcpr2 = set 100

rccf1 = set 100

rccf2 = set 100

rcpr2 = copy trf ; indlæs fast del

rccf2 = copy trf ; indlæs configurations-del

rcpr1 = edit rcpr2

1./define system:/,

i/

g0 = 1500 ; bufferlength

g90 = 1<1+1<2+1<3+1<4 ; include logger, RC 810 driver, TTY driver, poller driver

g48 = 16384 ; topaddress

g50 = 21 ; communicator-level

g51 = 18 ; scanner-level

g52 = 17 ; async timer-level

g64 = 3 ; runclaim

/,

f

rccf1 = edit rccf2

d./;begin configuration table text1/,

1./;end configuration table text1/,

i/

g8: ;stationreferencetable:

8,a0:b0,e0,e8,r.7

8,a1:b1,e1,e8,r.7

8,a2:b2,e2,e8,r.7

8,a3:b3,e3,e8,r.7

8,a4:b4,e4,e8,r.7

8,a5:b5,e5,e8,r.7

8,a6:b6,e6,e8,r.7

8,a7:b7,e7,e8,r.7

8,a10:b10,e10,e8,r.7

0 ;end stationreferencetable

;stationtable:

e8:j0,0,0,0; non existing device

b0:j1,0,0,0; site 0

e0:j2,1,0,0,7,3; RC 810 driver

b1:j1,0,0,0; site 1

e1:j2,1,0,0,8,4; RC 810 driver

b2:j1,0,0,0; site 2

e2:j3,2,0,0,3,20,0,0,1000,j11,0,r.20; TTY driver

b3:j1,0,0,0; site 3

e3:j3,2,0,0,3,20,0,1,1000,j11,0,r.20; TTY driver

b4:j1,0,0,0; site 4

e4:j3,2,0,0,3,20,0,2,1000,j11,0,r.20; TTY driver

b5:j1,0,0,0; site 5

e5:j3,2,0,0,3,20,0,3,1000,j11,0,r.20; TTY driver

b6:j1,0,0,0; site 6

e6:j3,2,0,0,3,20,0,4,1000,j11,0,r.20; TTY driver

b7:j1,0,0,0; site 7

e7:j4,0,0,0,4,22,0,1,1300,1000,0,r.4; CDC 711 driver

b10:j1,0,0,0; site 8

e10:j4,0,0,0,5,23,0,1,1300,1000,0,r.4; CDC 711 driver

;end stationtable

```

/,

s 1,
d./;begin cdc 2400 baud driver/,
i/

g49:      ; devicecodes:

c0=1      ; deviceno 1. 2400 baud line
c1=28     ; interruptlevel
c2=0      ; half duplex
c5=g5     ; address of sitetable

/,

l./;end cdc 2400 baud driver/,
s 1,
d./;begin cdc 2400 baud driver/,
i/

c0=2      ;deviceno 2. 2400 baud line
c1=29     ;interruptlevel
c2=0      ;half duplex
c5=g5     ;address of sitetable

/,

l./;end cdc 2400 baud driver/,
s 1,
d./;begin async multiplexer/,
i/

c0=e2, c1=e3, c2=e4, c3=e5, c4=e6, c5=0, c6=0, c7=0

c11=3     ;deviceno TTY multiplexer
c12=20    ;interruptlevel -

/,

l./;end async multiplexer/,
s 1,
d./;begin sync drivers/,
i/

```

c0=4 ;deviceno CDC 711

c1=22 ;interruptlevel

c2=0 ;half duplex

/,

l./;end sync drivers/,

s l,

d./;begin sync drivers/,

i/

c0=5 ;deviceno CDC 711

c1=23 ;interruptlevel

c2=0 ;half duplex

/,

l./;end sync drivers/,

s l,

d./;begin text2/,

l./;end text2/,

f

;derefter kan programmet oversættes: (oversat kode i rcpr2)

o tp

rcpr2=slam list.yes names.yes print.yes rcpr1 rccf1

o c