CODEN: LUTFD2/(TFRT-5202)/1-174/(1977)

PROGRAM FOR LOGGING AND DATA ANALYIS ON PDP11/03

TYKE PÅLSSON
LEIF RADING

Department of Automatic Control
Lund Institute of Technology
November 1977

PROGRAMPAKET FÖR MÄTVÄRDESINSAMLING OCH MÄTVÄRDESANALYS

Examensarbete utfört av:

Tyke Pålsson
Leif Rading

Handledare: Ivar Gustavsson

Institutionen för Reglerteknik
vid Lunds Tekniska Högskola

Lund November 1977

SAMMANFATTNING

Rapporten beskriver ett programpaket som är utvecklat för en PDP11/03 dator. Det är avsett för mätvärdes-insamling,mätvärdesanalys samt för enklare regleruppgifter.

Programpaketet är delvis fråge/svars-styrt och delvis kommandostyrt och är utvecklat för att användas tillsammans med en RT11 förgrund/bakgrunds monitor.

Programpaketet är i första hand tänkt som en bas för vidare utveckling. En användare kan själv sätta in egna rutiner konstruerade för den aktuella användningen.

INNEHÅLLSFÖRTECKNING

INLEDNING

INMAT är ett interaktivt programpaket för mätvärdesinsamling och mätvärdesanalys. Det kan också användas för att utföra enklare regleruppgifter och som signalgenerator. Paketet är skrivet i FORTRAN.

INMAT är uppbyggt av två delar. Den ena delen, stommen, utför mätvärdesinsamling, sättning av styrsignaler, lagring och inhämtning av data från massminne samt organiserar operatörskommunikationen. Den andra delen består av ett antal rutiner som analyserar och utför beräkningar på mätdata. Dessa rutiner anropas av stommen. De är för närvarande få till antalet och främst avsedda som en demonstration av hur INMAT kan användas. Användaren kan sätta in egna rutiner konstruerade för den aktuella användningen.

INMAT är konstruerat för att användas tillsammans med en RT11 förgrunds/bakgrunds monitor. Erforderligt primärminne är 24 k ord. Som massminne har använts 2 floppy diskar med en kapacitet av vardera 120 k ord. Kommunikationen med datorn sker med hjälp av terminal och radskrivare. Samplingstiden är begränsad till 40 ms, antalet in- och utsignaler till 8 resp 4.

RT11 monitorn delar upp datorns minne i en förgrundsarea med hög prioritet och en bakgrundsarea med låg prioritet. I förgrundsarean ligger de tidskritiska reelltidsrutinerna för, t ex, mätvärdesinsamling och reglering och i bakgrundsarean ligger icke tidskritiska rutiner för, t ex, mätvärdesanalys och utskrift.

Rutinernas databas i respektive area består av två listor. Ena listans element innehåller informationen till en rutin (t ex samplingsperiod, rutintyp). Den andra listans element innehåller rutinens parametrar (t ex amplitud, frekvens). Information och parametrar för en rutin kallas tillsammans för en nod och informationsdelen för nodhuvud.

Paketet är kommandostyrt i den meningen att användaren under körningens gång kan ändra experimentvillkor såsom:

- Aktivering och deaktivering av program,
- Ändring av förgrundsrutinernas parametervärden,
- Listning av förgrundsrutinernas parametervärden,
- Listning av mätdata på bildskärmsterminal eller radskrivare.

Kapitel 1 till 3 behandlar programpaketets struktur, kapitel 4 beskriver befintliga rutiner och i kapitel 5 finns exempel på körningar. I appendix A och B visas gången vid insättning av nya rutiner i systemet och i appendix C finns samtliga program listade.

TECKENFÖRKLARING

Följande tecken har speciell betydelse i anslutning till de givna exemplen.

.....	- Understruken text skrivs ut av systemet.
↵	- Nedtryckning av tangenten return.
(.....)	- Parenteserna inklusive texten emellan kan utelämnas vid kommandot.
*	- Parameter som är märkt med en * är en status-parameter. Den kan ges initialvärde, i annat fall sätts den till 0.
#	- Parameter till uppstartrutin sätts fråga/svars-styrt.

1. FÖRGRUNDS/BAKGRUNDS SYSTEMET

I ett förgrunds/bakgrunds system disponerar två helt oberoende program "samtidigt" datorns resurser, förgrundsprogrammet är det tidskritiska och högprioriterade programmet. När förgrundsprogrammet vilar körs det lågprioriterade bakgrundsprogrammet tills förgrundsprogrammet på nytt kräver centralenheten. Vidare finns det i respektive avdelning faciliteten kompletterande rutiner. De har högre prioritet än övriga rutiner och startar asynkront när en speciell händelse inträffar. Kompletterande rutiner avbryter ej varandra utan läggs på kö och exekveras i ordningen först in - först ut.

1.1 FÖRGRUNDSPROGRAMMET

Förgrundsprogrammet har som huvuduppgift att:

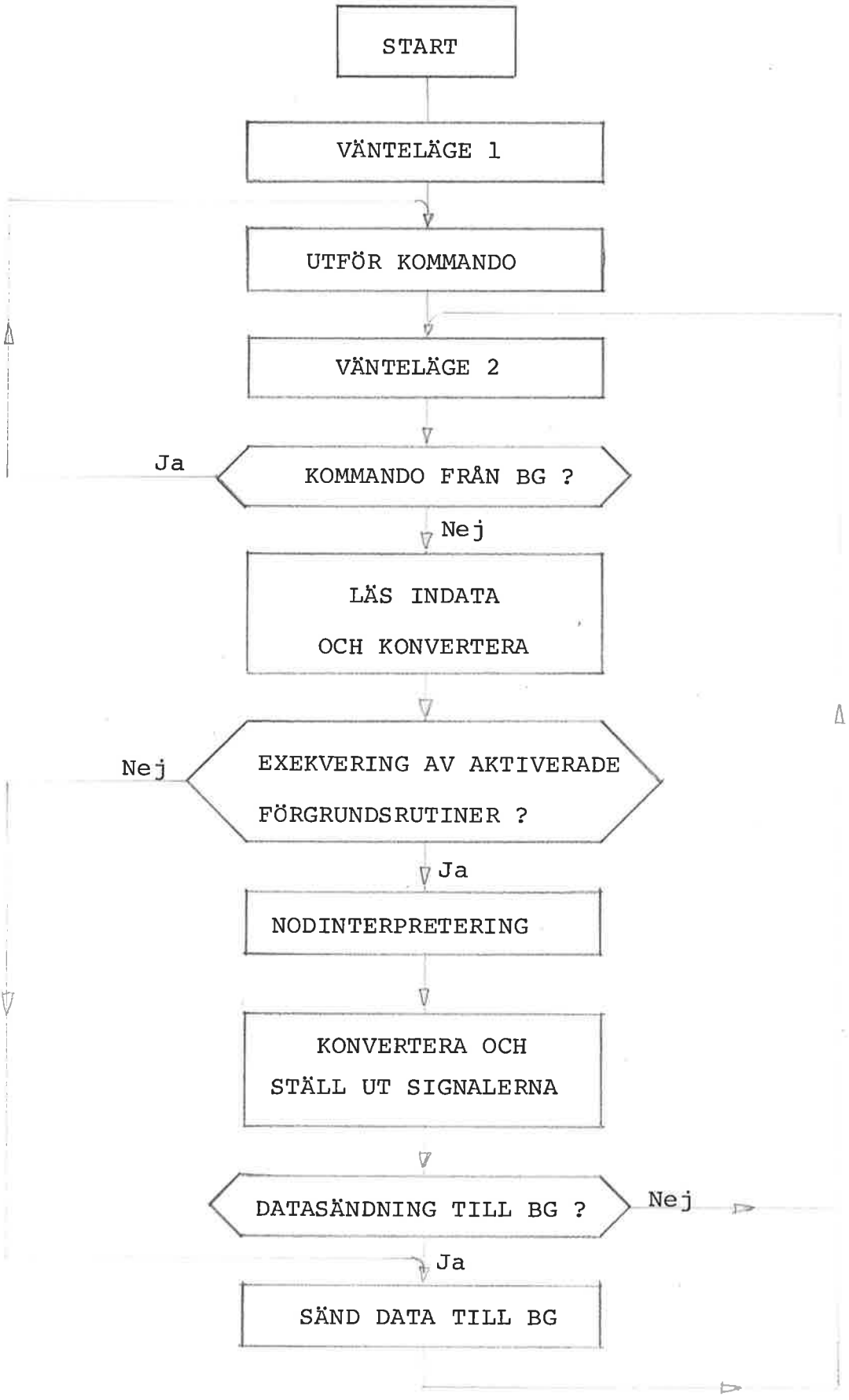
1) Samla in mätdata
2) Interpretera signalgenererings- och reglernoder
3) Ställa ut signaler
4) Sända insignaler och beräknade utsignaler till bakgrundsprogrammet
5) Verkställa kommandon från bakgrundsprogrammet

1.1.1 FÖRKLARING AV FÖRGRUNDSPROGRAMMETS BLOCKSCHEMA

Nedan ges en detaljerad beskrivning av förgrundsprogrammet med utgångspunkt från figur 1.

Start

Operatören startar huvudprogrammet (programmet kallas FG) i förgrundsarean. FG exekverar tills vänteläge 1 nås. När bakgrundsprogrammet har startat och mottagit information om huvudsamplingsperioden och antalet in- och utsignaler från operatören sänds det till förgrundsprogrammet som därvid lämnar vänteläget.



Figur 1: Blockschema för förgrundsprogrammet

Utför kommando från BG

När bakgrundsprogrammet vill ha ett kommando utfört sänds ett meddelande till förgrundsprogrammet. Då meddelandet har mottagits startar omedelbart den kompletterande rutinen FLAGR som noterar att meddelandet har mottagits. Meddelandet finns nu i en vektor (kallas i programmet IB1) och har följande form:

Ord 1 - Antalet mottagna ord, första ordet ingår ej.
Ord 2 - Information om vilket kommando som ska utföras.
Ord 3-60 - Övrig information.

Vänteläge 2

FG lämnar vänteläge 2 om någon av följande händelser inträffar:

1) Dels är det dags för en ny sampling och dels har ett meddelande inkommit från bakgrundsprogrammet att data från förra samplingstillfället har mottagits. Båda dessa fall medför att rutinen FLAG exekveras en gång. FLAG måste alltså köras två gånger för att vänteläge 2 ska lämnas.

2) FLAGR har körts, alltså har information från bakgrundsprogrammet blivit mottagen (se ovan).

Läs indata och konvertera

Subrutinen LOGIN anropar ADIN, som läser mätsignalerna från A/D-omvandlaren, och konverterar dem till flyttal i intervallet (+1.0, -1.0). Därefter konverterar LOGIN mätsignalerna till fysikaliska värden med två konstanter från ingenjörsnoden, nollnivå och skalfaktor. Läsning och konvertering sker vid varje samplingstillfälle i följande ordning: insignal 1, insignal 2, ..., insignal NI, där antalet insignaler NI ges av operatören (se kap 5.1). Maximalt kan systemet ha 8 insignaler.

Nodinterpretering

Listan med nodhuvuden gås igenom. I varje nod med IR $\geq$ 0 nedräknas innehållet i räknaren IR med en enhet. Om räknaren blir noll påbörjas exekveringen av det program som är förknippat med noden. Därefter återställs räknaren till samplingsperioden IPE, vilken anger hur ofta rutinen ska exekvera.

Konvertera och ställ ut signalerna

Subrutinen LOGOUT konverterar de beräknade värdena (utsignalerna) med konstanterna nollnivå och skalfaktor från ingenjörsnoden. Därefter anropar LOGOUT subrutinen DAOUT som ställer ut signalerna till D/A-omvandlaren. Utställning av signalerna sker först då kommandot STAR har givits och utförs i följande ordning: utsignal 1, utsignal 2, ..., utsignal NO. NO står för antalet utsignaler och ges av operatören då bakgrundsprogrammet startas (se kap 5.1). Maximalt kan systemet ha 4 utsignaler.

Sänd data till bakgrundsprogrammet

Till bakgrundsprogrammet sänds tidpunkten för mätvärdesinsamlingen, mätvärdena och de beräknade värdena (utsignalerna).

1.2 BAKGRUNDSPROGRAMMET

Bakgrundsprogrammet har följande huvuduppgifter:

1) Mottaga mätdata och information från förgrundsprogrammet
2) Sända mätdata till massminne
3) Hämta mätdata från massminne
4) Interpretera analys- och utskriftsnoder
5) Organisera operatörskommunikationen

1.2.1 FÖRKLARING AV BAKGRUNDSPROGRAMMETS BLOCKSCHEMA

Nedan ges en detaljerad beskrivning av bakgrundsprogrammet med utgångspunkt från figur 2.

Start

Bakgrundsprogrammet BG startar systemet genom att:

1) Fråga operatören om huvudsamplingsperioden och antalet in- och utsignaler.
2) Öppna temporär datafil med namnet WORK.DAT på enheten DX1.
3) Organisera kommunikationen med massminne.
4) Informera förgrundsprogrammet om huvudsamplingsperioden och antalet in- och utsignaler.
5) Initiera den kompletterande rutinen DFBT.

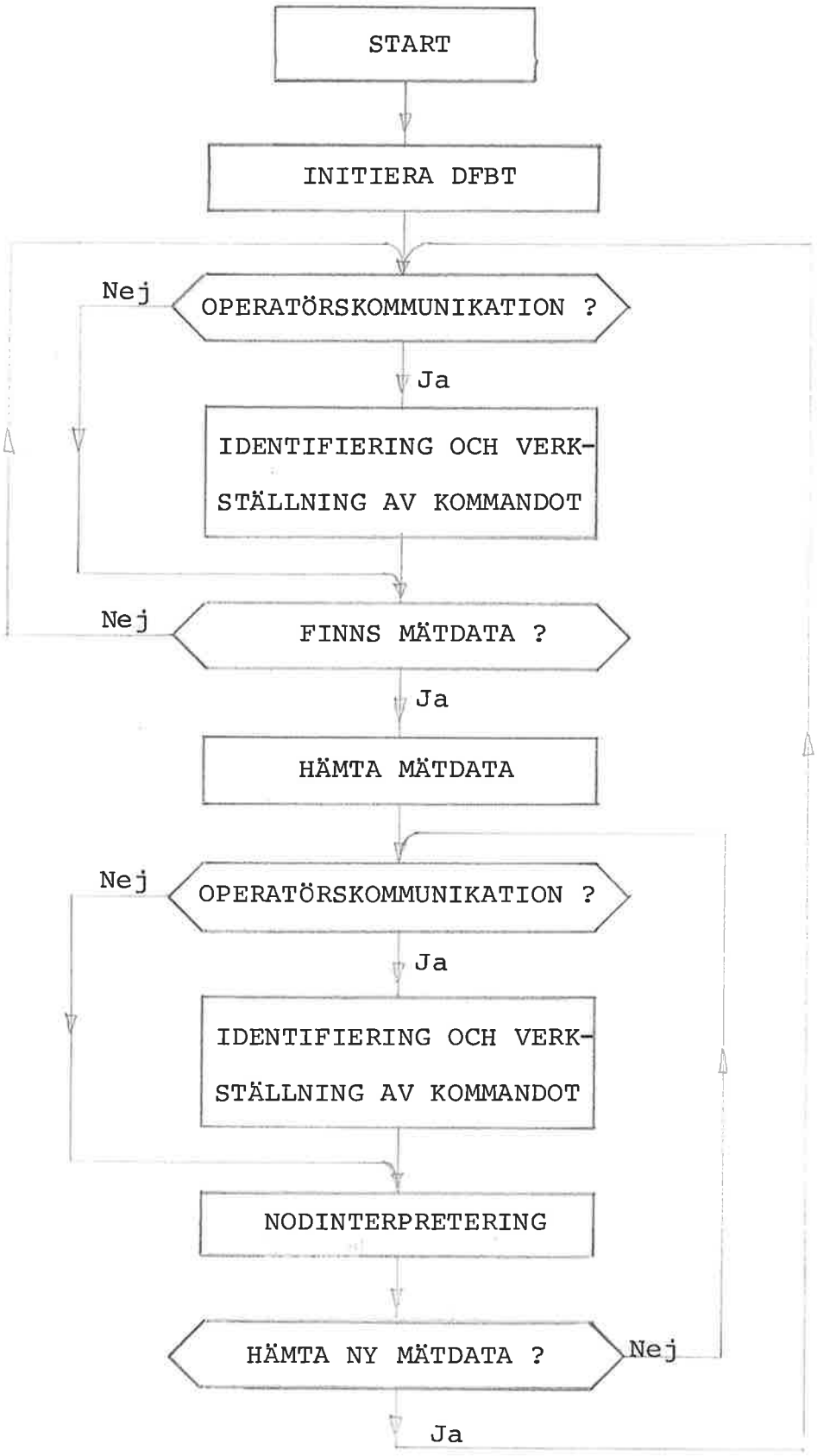
Initiera DFBT

Initiering av DFBT innebär att den placeras i en exekveringskö och startar först då information eller data mottagits från förgrundsprogrammet. DFBT har till uppgift att organisera lagring av mätdata på massminne och i buffertarna IB2, IB3 i primärminnet.

Innan rutinen lämnas läggs den åter i exekveringskön, varvid den åter är redo att mottaga data.

Förfrågan om operatörskommunikation

Subrutinen TCMD känner av om operatören vill kommunicera, dvs om operatören har slagit return. Om så är fallet startas subrutinen OPCOM som skriver ut tecknet > och är berett att ta emot ett kommando. Det bör observeras att bakgrundsprogrammets tid delas mellan operatörskommunikationen och mätvärdesanalysen, vilket innebär att mätvärdesanalysen avstannar fullständigt då operatörskommunikation pågår. Mottagning och lagring av data från förgrundsprogrammet fortgår dock ostört.



Figur 2 : Blockschema för bakgrundsprogrammet.

Identifiering och verkställning av kommandot

OPCOM startar subrutinen TECK som identifierar kommandot, varefter ett hopp sker till satser som verkställer kommandot.

Hämta beräkningsdata

Den kompletterande rutinen COP hämtar beräkningsdata till beräknings- och analysprogrammen. Beroende på tidsskillnaden mellan insamling och analys sker inhämtningen antingen från buffertarna IB2, IB3 eller från massminne. Det senare sker om tidsskillnaden är stor mellan insamling och analys.

Nodinterpretering

Listan med nodhuvuden gås igenom. I varje nod med IR $\geq$ 0 nedräknas innehållet i räknaren IR med en enhet. Om räknaren blir noll påbörjas exekveringen av det program som är förknippat med noden. Därefter återställs räknaren till samplingsperioden IPE, vilken anger hur ofta rutinen ska exekvera.

Exempel: Antag att IPE sätts lika med 5, då kommer rutinen att exekvera vart femte samplingstillfälle.

2. OPERATÖRSKOMMUNIKATION

Operatörskommunikationen sköts av en interaktiv rutin OPCOM i bakgrundsprogrammet som delvis är fråga/svar-styrt och delvis kommandostyrt.

Rutinen TCMD känner vid olika tillfällen av om operatören vill kommunicera, dvs har slagit return. Om så är fallet startas subrutinen OPCOM som skriver ut > och är berett att ta emot ett kommando. Ett kommando består av fyra tecken. Subrutinen TECK identifierar kommandot. Operatörskommunikationen är avslutat när # # # # # skrivs ut och exekveringen av beräkningsrutinerna återupptas då.

2.1 KOMMANDON FÖR FÖRGRUNDSPROGRAMMET

OPEN:TYPE NR

Initierar sättningen av en nod som är förknippad med en rutin som har namnet TYPE. NR anger nodens nummer, tal i intervallet 1 till 20 är tillåtna nodnummer, se kap 4.1. Om noden redan är aktiv hämtas den in till en arbetsarea. I annat fall körs nodtypens eventuella uppstartrutin och därefter startar rutinen OPCOM subrutinen SNOD som sätter nodhuvudet fråga/svar-styrt. Parametrarna sätts och ändras av subrutinen SPAR. Rutinen skriver ut * och följande tre kommandon kan ges:

NM=VALUE — Parametern NM i arbetsarean får det nya värden VALUE. Parameternamnet består av max två tecken.

LI(/NR) — Listar namn och värden på parametrarna i arbetsarean på radskrivaren om NR=6, i annat fall på terminalen.

CL — Noden i arbetsarean sänds till förgrundsprogrammet. Operatörskommunikationen är avslutad.

Exempel, se kap 4.

ACTI:TYPE NR,...

Aktiverar de uppräknade noderna, som därvid blir synkroniserade. Max 6 noder.

Exempel:

≥ACTI:SINU1,PIDR20 ↵

LSTF(/NR)

Subrutinen LSTF listar aktiva och upptagna noder, på radskrivare om NR=6, i annat fall på terminalen.

Exempel:

≥LSTF/6 ↵

STOP:TYPE NR,...

Deaktiverar och sätter de uppräknade noderna lediga. Max 6 noder.

Exempel:

≥STOP:SINU1,PIDR20 ↵

STAR

Startar interpreteringen av nodlistan.

Exempel:

≥STAR ↵

STOR

Stoppar interpreteringen av nodlistan.

Exempel:

≥STOR ↵

STAC

Startar mätvärdesinsamlingen och sändningen av mätdata till bakgrundsprogrammet.

Exempel:

≥STAC ↵

STOC

Stoppar mätvärdesinsamlingen och sändningen av mätdata till bakgrundsprogrammet.

Anm. Innan bakgrundsprogrammet stannas måste operatören övertyga sig om att data ej sänds från förgrundsprogrammet till bakgrundsprogrammet, vilket kan göras med detta kommando.

Exempel:

≥STOC ↵

2.2 KOMMANDON FÖR BAKGRUNDSPROGRAMMET

ACTI:TYPE NR

Sättter och aktiverar en nod som är förknippad med en rutin i bakgrundsprogrammet som har namnet TYPE. NR är ett ordningsnummer i intervallet 1 till 9 vars enda uppgift är att skilja på noder som är förknippade med samma rutintyp. Om noden redan är aktiv informeras operatören därom. I annat fall sker följande:

1) Subrutinen FRENOD letar upp första lediga nod i nodlistan.

2) Subrutinen TFD frågar efter nodens insignaler, max fyra. En insignal är:
   - Mätsignal från förgrundsprogrammet: INPU1, INPU2, ..., INPU8.
   - Utsignal från förgrundsprogrammet: OUTP1, OUTP2, OUTP3, OUTP4.
   - Parametrar i en annan rutin (nod) i bakgrundsprogrammet: TYPE NR/0,.....,TYPE NR/4.
     Exempel, se kap 4.2.3.

3) Nodens uppstartrutin körs, denna sätter bl a nodens övriga informations- och beräkningsparametrar.

4) Noden aktiveras.

Noder kan synkroniseras genom att:

1) Aktivera noderna i den följd de skall exekveras utan att använda kommandot STOP dessemellan.

2) Ge som insignal en parameter till den nod TYPE NR skall synkroniseras med.

STOP:TYPE NR

Noden TYPE NR deaktiveras och sätts ledig. Max en nod.
Exempel:
≥STOP:GRAF2 ↵

LSTB(/NR)

Listar aktiva noder i bakgrundsprogrammet, på radskrivare om NR=6, i annat fall på terminalen.
Exempel:
≥LSTB/6 ↵

CRNF

Beroende på antalet aktiva datafiler behandlar subrutinen CRNF kommandot på följande sätt:

1) Om vid anropet två datafiler är aktiva, vilket är det maximala antalet, skrivs ett felmeddelande ut.

2) Om en datafil är klar, då permanentas den och * skrivs ut, se vidare nedan.

3) En datafil är aktiv och då skrivs * ut, se vidare nedan.

När * skrivs ut är programmet berett att aktivera en ny datafil på kommandot

DEV:NAME.EXT[N1] =

Storlek på den aktiverade datafilen bestäms av N1 enligt:

1) N1 utelämnat eller N1=0
   - hälften av största eller näst största segmentet på enheten

2) N1=-1
   - största segmentet

3) N1=heltal (0)
   - segment med N1 block

Exempel 1:
*DX1:AAAA.DAT= ↵

På enhet DX1 aktiveras en datafil med extension DAT av storlek (se ovan).

Exempel 2:
*DX:BBB.CCC[-1]= ↵

På enhet DX aktiveras en datafil med extension CCC av storlek (se ovan).

Följande meddelanden är möjliga:

1) NO MORE SPACE AVAILABLE IN OPENED FILES
   - öppnade filer fyllda.

2) TWO FILES ALREADY ACTIVE
   - Operatören försöker aktivera en tredje datafil medan två redan är aktiva.

3) NR MISSING BLOCKS
   - Om en ny datafil aktiveras efter det att meddelandet 1 skrivits ut, kommer detta meddelande att skrivas ut och NR står för antalet block som har gått förlorade för analysprogrammen.

4) OLD FILE PERMANENT
   - Klar datafil permanentas.

   Anm. Då bakgrundsprogrammet avbrytes på grund av fel eller då två CTRL C ges, kan aktiva datafiler permanentas med kommandot: .CLOSE
   Detta måste göras efter avslutat experiment.

3. LISTSTRUKTUR

För att uppfylla kravet på flexibilitet och tillfredsställande minnesutnyttjande används listor för lagring av programinformation och parametrar.

Listorna i FÖRGRUNDSAREAN är lagrade i vektorerna LF (nodhuvud) och IPF (parametrar). I IPF lagras förutom parameterlistan även in- och utsignalareorna och tidpunkten för dess insamling. I BAKGRUNDSAREAN lagras listorna i vektorerna LIST respektive IPA. I övrigt som i förgrundsarean.

Nodhuvudena är av fix längd. Tolkningen av innehållet är lika för alla programtyper i samma area men en viss skillnad förekommer mellan areorna. Listorna med parametrarna består av element med varierande längd och innehållet är beroende på rutintypen.

3.1 INNEHÅLL I NODHUVUD I FÖRGRUNDSPROGRAMMET

Ordnr	Namn	Innebörd
0	IR	räknare
1	IPE	period
2	ITYP	programtyp
3	K1	adress till parameterlista
4	K2	" -
5	K3	" -
6	K4	" -
7	K5	" -
8	K6	adress till insignalarea
9	K7	" -
10	K8	" -
11	K9	om K9>0: utsignalens läge i utsignalarean, i annat fall K=0.

Kommentar:

1. Nodstatus: 1. nod ledig om IR=1 och IPE=0
   2. nod upptagen men inaktiv om IR=-1 och IPE=-period
   3. nod aktiv om IR≥0 och IPE=period

2. För aktiva noder räknas IR ned vid varje samplingstillfälle. När IR=0 interpreteras noden varefter IR återställes till IPE.

3. Perioden anges i antal samplingar. Den verkliga samplingstiden fås ur sambandet:
   ST=IPE*TSM*0.02
   där TSM är huvudsamplingstiden.

4. Nodens parametrar är lagrade i ordningsföljden:
   1. ändringsbara parametrar
   2. statusparametrar
   3. uppstartrutinens parametrar (om några)

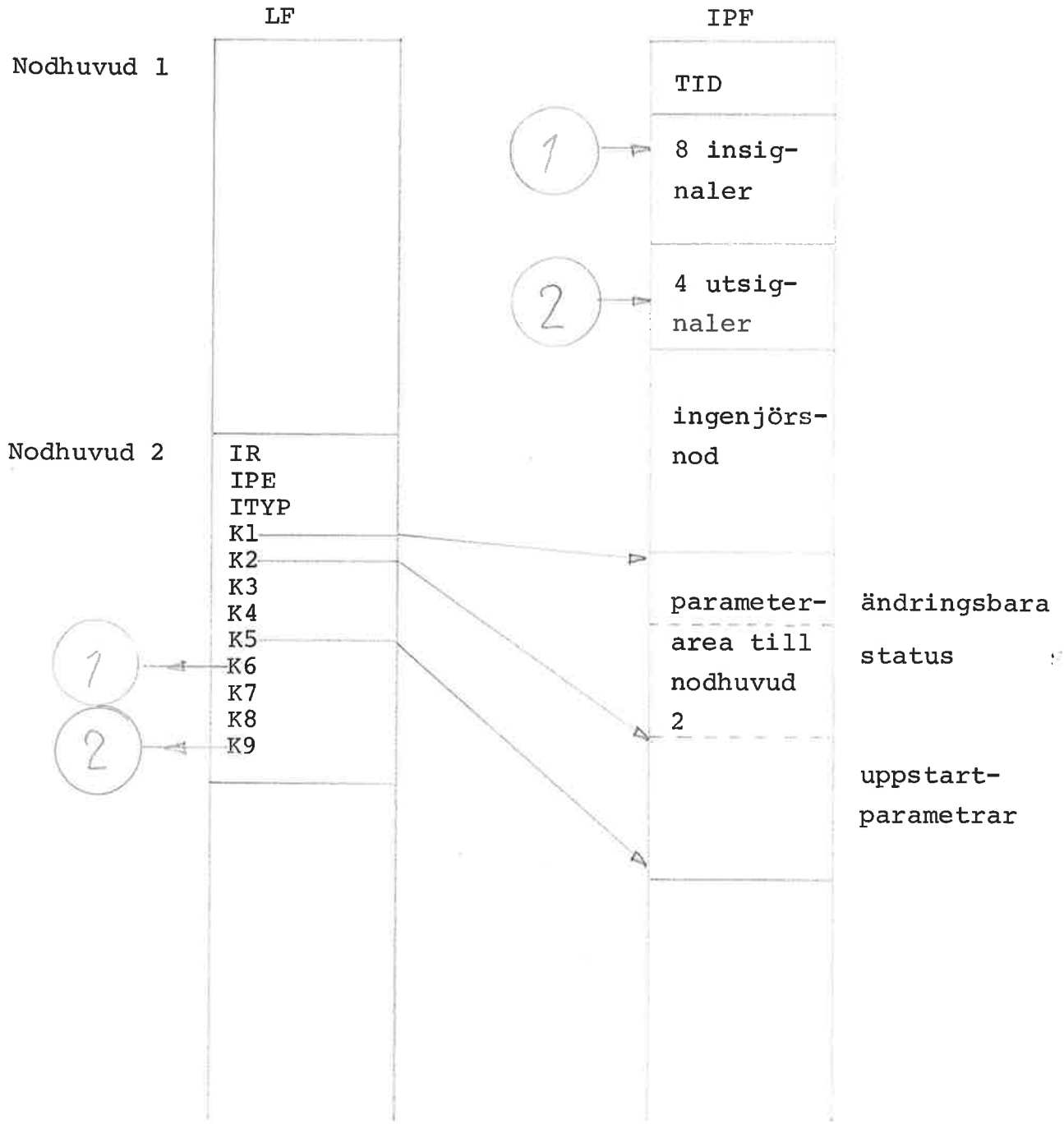
   i parameterlistan mellan pekarna K1 och K5.

5. K2 innehåller adressen till uppstartrutinens första parameter.

6. Användningen av K3 och K4 varierar med rutintyp och bestäms vid insättningen av en ny rutintyp.

Se figur 3.

Figur 3 : Figuren visar vektorerna LF och IPF. Vidare illustreras växelverkan mellan ett nodhuvud och dess parameterarea. Noden har en ut- och en insignal och en uppstartrutin.



3.2 INNEHÅLL I NODHUVUD I BAKGRUNDSPROGRAMMET

3.2.1 NODHUVUDINNEHÅLL - BAKGRUND

Ordnr	Namn	Innebörd
0	IR	räknare
1	IPE	period
2	ITYP	identifierare
3	K1	adress till parameterlista
4	K2	" -
5	K3	" -
6	K4	" -
7	K5	" -
8	K6	adress till insignalarea
9	K7	" -
10	K8	" -
11	K9	" -

Tolkningen av nodinnehållet skiljer sig på följande punkter från förgrundsarean:

1. Om IPE sätts ≤0 vid aktiveringen körs noden en gång, varefter den deaktiveras och noden sätts ledig.

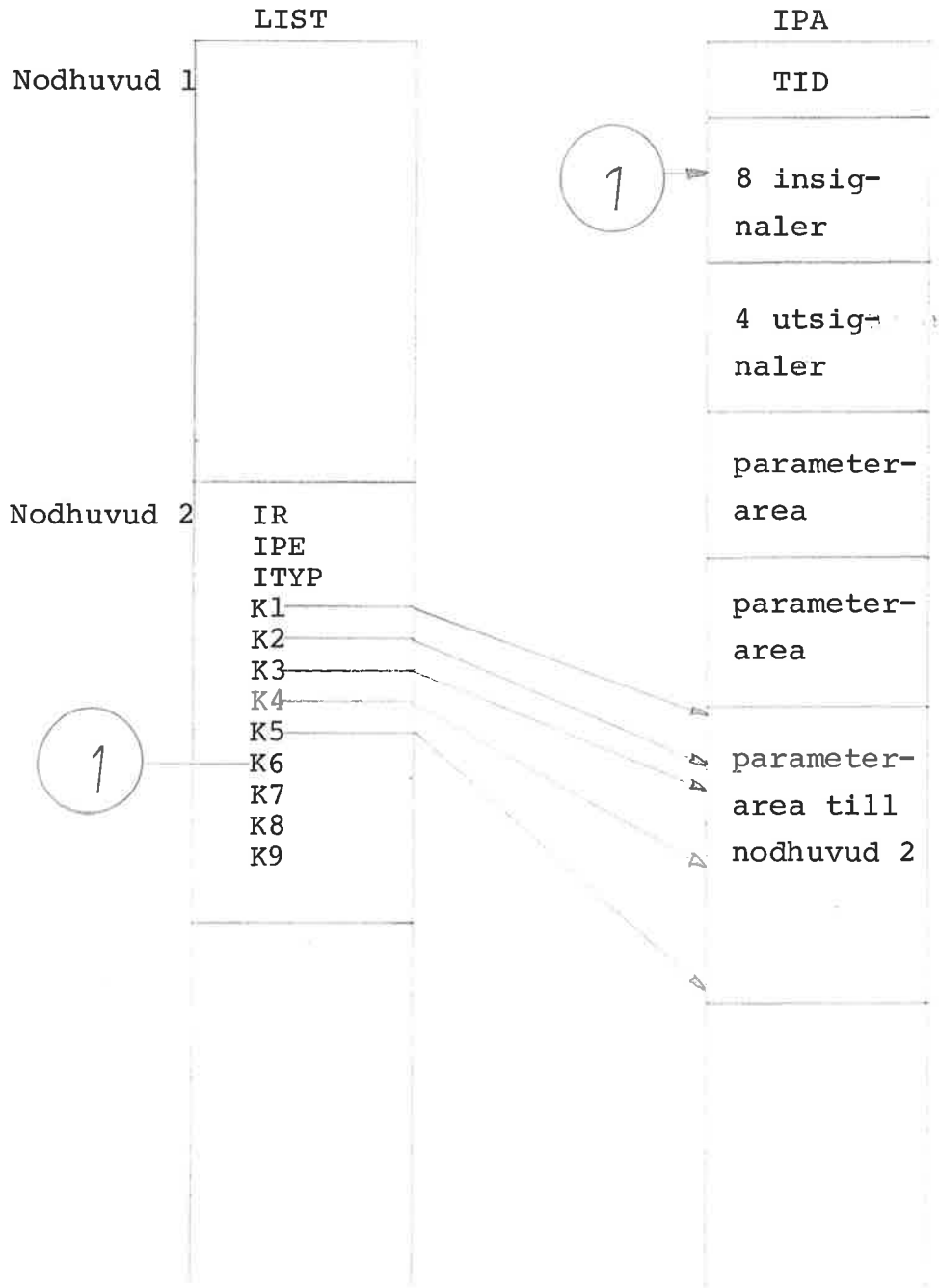
2. ITYD=1000 * programtyp + ordningsnummer
   Exekveringsordningen påverkas ej av ordningsnumret. Dess enda uppgift är att skilja på noder förknippade med samma programtyp.

3. K9 är en pekare till en insignal, ej till en utsignal som i förgrundsarean.

4. K2-K4 innehåller adresser till parametrar. Användningen bestäms vid insättningen av en ny rutintyp.

Exempel se figur 4.

Figur 4 : Figuren visar vektorerna LIST och IPA. Vidare illustreras växelverkan mellan ett nodhuvud och dess parameterarea. Noden har en insignal.



4. RUTINER

4.1 RUTINER I FÖRGRUNDSPROGRAMMET

I förgrundsprogrammet ligger tidskritiska reelltidsrutiner såsom reglering och signalgenerering.

Samtliga rutiner har följande gemensamt:

1. Nodnummer (absolut) i intervallet [1,20]. Exekveringen sker i nodnummerordning.

2. Samplingstiden ges som heltal och anger antal huvud-samplingstider.
   Absolut samplingstid=IPE * huvudsamplingstid * 0.02 [sek]

3. Insignaler är av två slag:
   - mätsignal: INPU1,...,INPU8
   - utsignal från annan nod: TYPE NR

4. Utsignalen lagras alltid i nodens parameterarea. Om signalen skall ställas ut anger operatören var:
   OUTP1,...,OUTP4

5. Parametrarnas initialvärde = 0.0

Efter varje rutinbeskrivning ges ett exempel.

4.1.1 ADDS

Rutinen ADDS adderar sina insignaler.

Anrop: ≥OPEN:ADDS NR

Insignaler: max 3

Argument: W1 - viktfaktor för insignal 1
W2 - viktfaktor för insignal 2
W3 - viktfaktor för insignal 3
Y - utsignal *

Exempel:

```
≥OPEN:ADDS 19 ↵
SAMPLE PERIOD:12↵
INPUT SIGNAL:RAMP1,INPU2,PULS5 ↵
OUTPUT SIGNAL:OUTP2 ↵
*W1=0.1 ↵
*W2=0.2 ↵
*W3=0.3 ↵
*CL ↵
```

4.1.2 NODI

Rutinen NODI genererar ett normalfördelat brus.

Anrop: ≥OPEN:NODI NR

Insignaler: saknas

Argument: XM - medelvärde
SG - standardavvikelse
Y - utsignal *

Metod: NODI adderar 12 rektangelfördelade tal i intervallet [0,1]. Summan subtraheras med talet 6.

Exempel:

```
≥OPEN:NODI8 ↵
SAMPLE PERIOD:2 ↵
INPUT SIGNAL: ↵
OUTPUT SIGNAL:OUTP1 ↵
*XM=2.2 ↵
*SG=2.0 ↵
*LI ↵
XM=0.22000E+01
SG=0.20000E+01
Y==0.00000E+00

*CL ↵
```

4.2.3 PIDR

Rutinen PIDR utför PID-reglering med hjälp av följande algoritm:

$$u(t) = G\left[\left(1 + \frac{TS}{TI(1-q^{-1})}\right)(y_r(t) - y(t)) - \frac{TD(1-q^{-1})(1-\beta)}{TS(1-\beta q^{-1})}\, y(t)\right]$$

där

$$\beta = \frac{TD}{TD + GD \cdot TS}$$

u(t) = utsignalen från regulatorn

y(t) = insignalen till regulatorn

$y_r(t)$ = referenssignalen

Anrop: >OPEN:PIDR NR

Insignaler: insignal, referenssignal (i nämnd ordning)

Argument:

G	- förstärkning	
TI	- integrationskonstant	
TD	- deriveringskonstant	
GD	- tidskonstant i derivatafilter	
YD	- gamla utsignalen	*
RI	- integrationsterm	*
D	- derivationsterm	*
Y	- utsignal	*

Anm: ingen gränsvärdeskontroll göres.

Exempel:

```
>OPEN:PIDR 15 ↵
SAMPLE PERIOD:1 ↵
INPUT SIGNAL:INPU1,SINU7 ↵
OUTPUT SIGNAL: OUTP3 ↵
*G=1.0 ↵
*TD=0.7 ↵
*CL ↵
```

4.1.4 PRBG

Rutinen PRBG genererar en PRBS-signal.

Anrop: >OPEN:PRBG NR

Insignaler: saknas

Argument: AM - amplitud
Y - utsignal *

Rutinen har en uppstartrutin. Denna sätter antal bitar i generatorn, i intervallet [3,15], fråga/svars-styrt

Exempel:

```
*OPEN:PRBG20 ↵
NUMBER OF BITS:5 ↵                    #
SAMPLE PERIOD:18 ↵
INPUT SIGNAL: ↵
OUTPUT SIGNAL:OUTP4 ↵
*AM=0.2 ↵
*CL ↵
```

4.1.5 PULS

Rutinen PULS genererar en fyrkantsignal med variabel pulslängd.

Anrop: >OPEN:PULS NR

Insignaler: saknas

Argument: AM - amplitud
FR - frekvens (rad/sek)
PL - pulslängd (sek)
X - tillstånd *
Y - utsignal *

Anm: PL<1/FR

Exempel:

```
>OPEN:PULS 1 ↵
SAMPLE PERIOD:2 ↵
INPUT SIGNAL: ↵
OUTPUT SIGNAL:OUTP4 ↵
*AM=0.4 ↵
*PL=3.0 ↵
*FR=0.1 ↵
*CL ↵
```

4.1.6 RAMP

Rutinen RAMP genererar en rampsignal.

Anrop: >OPEN:RAMP NR

Insignaler: saknas

Argument:	XK	- lutningskoefficient	*
	X	- tillstånd	*
	Y	- utsignal	*

Exempel:

```
>OPEN:RAMP 10 ↵
SAMPLE PERIOD:11 ↵
INPUT SIGNAL: ↵
OUTPUT SIGNAL:OUTP1 ↵
*X=0.5 ↵
*XK=0.1 ↵
*LI ↵

XK=0.10000E+00
X==0.50000E+00

*CL ↵
```

4.1.7 SINU

Rutinen SINU genererar en sinussignal med fasförskjutning.

Anrop: >OPEN:SINU NR

Insignaler: saknas

Argument:

AM	- amplitud	
FR	- frekvens (rad/sek)	
DF	- fasförskjutning (rad)	
X	- fasvinkel (rad)	*
Y	- utsignal	*

Exempel:

```
>OPEN:SINU7 ↵
SAMPLE PERIOD:5 ↵
INPUT SIGNAL: ↵
OUTPUT SIGNAL:OUTP2 ↵
*AM=0.1 ↵
*FR=0.2 ↵
*DF=0.3 ↵
*CL ↵
```

4.1.8 STEP

Rutinen STEP genererar en trappstegssignal.

Anrop: >OPEN:STEP NR

Insignaler: saknas

Argument:

AM	- amplitud	
X	- tillstånd	*
Y	- utsignal	*

Exempel:

```
>OPEN:STEP 4 ↵
SAMPLE PERIOD:2 ↵
INPUT SIGNAL: ↵
OUTPUT SIGNAL:OUTP4 ↵
*AM=0.01 ↵
*X=-0.5 ↵
*LI ↵

AM=0,10000E-01
X==-0.50000E+00
Y==0.00000E+00

*CL ↵
```

4.2 RUTINER I BAKGRUNDSPROGRAMMET

I bakgrundsprogrammet ligger rutiner som ej kräver omedelbar respons såsom analys- och utskriftsrutiner.

Samtliga rutiner har följande gemensamt:

1. Nodnummer i intervallet [1,9]

2. Samplingsperioden ges som heltal och anger antal huvudsamplingstider.
   Absolut samplingstid=IPE * huvudsamplingstid * 0.02 [sek]

3. Insignaler är:
   - data från förgrundsprogrammet: INPU1,...,INPU8,
     OUTP1,...,OUTP4
   - utsignaler från en annan nod i bakgrundsprogrammet:
     TYPE NR/0,...,TYPE NR/4

4. Varje rutin har en individuell startrutin, som i fråga/svars-form erhåller erforderlig information.

5. En nod innehåller bl a de fem adresserna K1-K5. En adress anger läget för en parameter eller första elementet i en "vektor" (area) i parameterlistan. Användaren refererar till dessa med följande kommandon:

Kommando	Adress
TYPE NR(/0)	-K1
TYPE NR/1	-K2
TYPE NR/2	-K3
TYPE NR/3	-K4
TYPE NR/4	-K5

(se rutinen XWRI)

4.2.1 GRAF

Rutinen GRAF plottar ett given antal värden från en vektor på radskrivaren eller på terminalen.

Anrop: >ACTI:GRAF NR

Antal insignaler: 1

Rutintypens uppstartrutin GRAFST frågar efter:

1. antal värden (element)
2. insignalernas undre gräns
3. insignalernas övre gräns
4. utskriftenhet, radskrivare (LP) eller terminal (TT)

Tillgänglig data i rutinens parameterarea är lagrad enligt följande:

Adress	Data
K1	antal element
K2	övre gräns
K3	undre gräns
K4	-
K5	logisk enhet (LP=6, TT=7)

Exempel:

```
>ACTI:GRAF2  ↵
SAMPLE PERIOD:100 ↵
INPUT SIGNAL:STAF 3/0 ↵
NUMBER OF ELEMENTS:1 ↵                    #
HIGHER LIMIT:0.1 ↵                        #
LOWER LIMIT:-0.1 ↵                        #
OUTPUT ON DEVICE:LP ↵                     #
```

4.2.2 GROU

Rutinen GROU grupperar data.

Anrop: >ACTI:GROU NR

Antal insignaler: 1

Rutintypens uppstartrutin GROUST frågar efter:

1. antal klasser
2. undre klassgräns
3. klassbredd

Tillgänglig data i rutinens parameterarea är lagrad enligt följande:

Adress	Data
K1	resultatvektor (adress till 1:a element)
K2	undre gräns
K3	klassbredd
K4	-
K5	antal klasser

Exempel:

```
>ACTI:GROU 1 ↵
SAMPLE PERIOD:12 ↵
INPUT SIGNAL:INPU 1 ↵
NUMBER OF CLASSES:100 ↵                   #
LOWER LIMIT:-0.9 ↵                        #
CLASS LENGTH:0.01 ↵                       #
```

4.2.3 IWRI, XWRI

Rutinerna IWRI och XWRI skriver ut heltal respektive reella tal.

Anrop: >ACTI:IWRI NR
XWRI NR

Antal insignaler: max 4

Rutintypernas uppstartrutin WRIST frågar efter:

1. På vilket yttre enhet utskrift skall ske, radskrivare (LP) eller terminal (TT).

Tillgänglig data i rutinens parameterarea är lagrad enligt följande:

Adress	Data
K1	-
K2	-
K3	-
K4	-
K5	logisk enhet (LP=6, TT=7)

Exempel: Säg att största och minsta värdet, beräknade av MAXM 3, och insignalen 5 ska skrivas ut på terminalen.

```
>ACTI:XWRI 2 ↵
SAMPLE PERIOD:10 ↵
INPUT SIGNAL:MAXM 3/1, MAXM 3/3, INPU 5 ↵
OUTPUT ON DEVICE:TT ↵
```

4.2.4 MAXM

Rutinen MAXM beräknar största och minsta värde samt dess samplingsnummer.

Antal insignaler: 1

Rutintypens uppstartrutin MAXMST frågar efter: inget

Tillgänglig data i rutinens parameterarea är lagrad enligt följande:

Adress	Data
K1	maxvärdets samplingsnummer
K2	maxvärdet
K3	minvärdets samplingsnummer
K4	minvärdet
K5	aktuellt samplingsnummer

Exempel:

```
>ACTI:MAXM 1 ↵
SAMPLE PERIOD:4 ↵
INPUT SIGNAL:INPU8 ↵
```

4.2.5 STAF

Rutinen STAF beräknar medelvärde och standardavvikelse med varierande viktfaktor.

Anrop: >ACTI:STAF NR

Antal insignaler: 1

Rutinens uppstartrutin STAFST frågar efter:

1. viktfaktorns initialvärde (WF)
2. viktfaktorns förändringsfaktor (WCF)

Viktfaktorn varierar enligt formeln:

$WF = WFC * (WF - 1.0) + 1.0$

Tillgängliga data i rutinens parameterarea är lagrad enligt följande:

Adress	Data
K1	medelvärde
K2	varians
K3	standardavvikelse
K4	viktfaktor
K5	antal data

Exempel:

```
>ACTI:STAF3 ↵
SAMPLE PERIOD:1 ↵
INPUT SIGNAL:OUTP1 ↵
WEIGHTING FACTOR:0.9 ↵                    #
WEIGHTING CHANGE FACTOR:0.9 ↵             #
```

5. KÖRNINGAR

I detta kapitel visas två exempel på körningar.

5.1 KÖRNING 1

I denna körning visas tillvägagångssättet vid insamling och lagring av mätdata samt utskrift.

```
.SET USR NO SWAP ↵                    ;Förhindrar att systemrutinen
                                       USR swappas.
.FRUN DX1:FG/NI500 ↵                  ;Start av förgrundsprogrammet
                                       samt allokering av extra
                                       minnesutrymme (250 ord)
.RUN DX1:BG ↵                         ;Start av bakgrundsprogrammet
MAIN SAMPLE PERIOD:5 ↵                ;Sätter huvudsamplingsperioden,
                                       0.02*5=0.1s
NUMBER OF INPUT SIGNALS:3 ↵           ;De tre första signalerna läses
                                       från A/D-omvandlaren vid varje
                                       samplingstillfälle
NUMBER OF OUTPUT SIGNALS:3 ↵          ;De tre första signalerna i ut-
                                       signalarean ställs ut till
                                       D/A-omvandlaren
↵                                     ;Anrop av operatörskommunikations-
                                       rutinen
>ACTI:XWRI1 ↵                         ;Aktivering av en skrivrutin
SAMPLE PERIOD:100 ↵                   ;Utskrift var hundrade sampling(10s)
INPUT SIGNAL: INPU1,INPU2,INPU3 ↵
OUTPUT ON DEVICE:LP ↵                 ;Utskrift sker på radskrivare
# # # # #                             ;Operatörskommunikationen avslutad
↵
>STAC ↵                               ;Initiering av datasändning från
                                       förgrundsprogrammet till bakgrunds-
                                       programmet
# # # # #
↵
>CRNF ↵                               ;Öppna ny datafil på massminne
*DX1:MAT.DAT[-1]= ↵                   ;En datafil med namnet MAT.DAT
                                       öppnas för data
# # # # #
↵
```

```
>STOP:XWRI1 ↵                    ;Deaktivering av XWRI1
# # # # #
↵
>STOC ↵                          ;Avbryter datasändningen från
                                  förgrundsprogrammet till bak-
                                  grundsprogrammet
# # # # #
^C                               ;Bakgrundsprogrammet avbryts
.
F>
^C                               ;Förgrundsprogrammet avbryts
^C

B>
CLOSE ↵                          ;Alla datafiler göres permanenta
.
```

5.2 KÖRNING 2

I denna körning visas tillvägagångssättet vid insamling, lagring, generering av styrsignal och utskrift av mätsignal.

```
.SET USR NO SWAP ↵                    ;Se föregående körning
.FRUN DX1:FG/NI500 ↵
.RUN DX1:BG ↵
MAIN SAMPLE PERIOD:5 ↵
NUMBER OF INPUT SIGNALS:3 ↵
NUMBER OF OUTPUT SIGNALS:2 ↵
↵
>OPEN:PULS1 ↵                         ;Initierar sättning av en puls-
                                       generator
SAMPLE PERIOD:10 ↵                    ;Rutinen kommer att köras var
                                       tionde sampling (varje sekund)
INPUT SIGNAL: ↵                       ;Saknas
OUTPUT SIGNALS:OUTP1 ↵                ;Signalen ställs ut som första
                                       utsignal
*AM=1.0 ↵                             ;Rutinens parametrar sätts
*FR=0.01 ↵
*PL=5.0E1 ↵
*LI ↵                                 ;Rutinens parametrar listas
AM= 0.10000E+01
FR= 0.10000E-01
PL= 0.50000E+02
X== 0.00000E+00
Y== 0.00000E+00
*CL ↵                                 ;Noden stängs och sänds till
                                       förgrundsprogrammet
# # # # #
↵
>OPEN:IGJN ↵                          ;Initierar ändring i ingenjörsnoden
*LI ↵                                 ;Listning av ingenjörsnoden
0.000000E+00   0.100000E+01           ;Insignal 1   (nollnivå/skalfaktor)
0.000000E+00   0.100000E+01           ;"         2
0.000000E+00   0.100000E+01           ;"         3
0.000000E+00   0.100000E+01           ;"         4
0.000000E+00   0.100000E+01           ;"         5
0.000000E+00   0.100000E+01           ;"         6
0.000000E+00   0.100000E+01           ;"         7
0.000000E+00   0.100000E+01           ;"         8
```

```
0.000000E+00   0.100000E+01     ;Utsignal 1
0.000000E+00   0.100000E+01     ;"        2
0.000000E+00   0.100000E+01     ;"        3
0.000000E+00   0.100000E+01     ;"        4
*OUTP1/1=-0.5 ↵                 ;Ändring av nollnivån till
                                 utsignal 1
*OUTP1/2=0.1 ↵                  ;Ändring av skalfaktor till
                                 utsignal 1
*LI ↵                           ;Listning av ingenjörsnoden
0.000000E+00   0.1000000E+01
0.000000E+00   0.1000000E+01
0.000000E+00   0.1000000E+01
0.000000E+00   0.1000000E+01
0.000000E+00   0.1000000E+01
0.000000E+00   0.1000000E+01
0.000000E+00   0.1000000E+01
0.000000E+00   0.1000000E+01
-0.500000E+00  0.1000000E+00
 0.000000E+00  0.1000000E+00
 0.000000E+00  0.1000000E+00
 0.000000E+00  0.1000000E+00
*CL ↵                           ;Noden stängs och sänds till
                                 förgrundsprogrammet
# # # # #
↵
>ACTI:GRAF1 ↵                   ;Aktivering av plottrutin
SAMPLE PERIOD:10 ↵              ;Rutinen körs var tionde
                                 sampling
INPUT SIGNAL:INPU1 ↵            ;Plottar första mätsignalen
NUMBER OF ELEMENT:1 ↵
HIGHER LIMIT:1.0 ↵
LOWER LIMIT:-1.0 ↵
OUTPUT ON DEVICE:LP ↵           ;Plottas på radskrivare
# # # # #
↵
>ACTI:PULS1                     ;Aktivering av pulsgeneratorn
# # # # #
↵
>STAR ↵                         ;Startar nodinterpreteringen i
                                 förgrundsprogrammet
# # # # #
↵
>STAC ↵                         ;Se föregående körning
# # # # #
↵
```

```
>OPEN:PULS1 ↵                    ;Öppnar noden för parameterändring
THE NODE IS ALREADY ACTIVE
*LI ↵
AM= 0.10000E+01
FR= 0.10000E-01
PL= 0.50000E+02
X== 0.69000E+02
Y== 0.00000E+00
*FR=0.5E-2 ↵
*PL=100 ↵
*CL ↵
# # # # #
↵
>STOC ↵
# # # # #
↵
>STOP:PULS1 ↵                    ;Deaktiverar pulsgeneratorn
# # # # #
↵
>STOR ↵                          ;Stoppar nodinterpreteringen
# # # # #
↵
>^C
.CLOSE ↵
.
F>
^C
^C
B>
```

APPENDIX A

INSÄTTNING AV EN NY RUTINTYP I BAKGRUNDSPROGRAMMET

Insättning av en ny rutintyp kräver ändring i följande program:

1. BD
2. OPCOM
3. BG

Som exempel används insättning av rutinen GROU som finns beskriven i kapitel 4.2.2.

ÄNDRING I BG

I vektorn CM2 införs rutinens namn (4 tecken). Lediga platser är markerade med tecknen & $ # % . Antalet program är begränsat till 20 stycken.

Exempel:

Rutinnamnet GROU placeras i femte vektorelementet.

Satsen:

```
    DATA CM2/4HGRAF,4HXWRI,4HMAXM,4HIWRI,
             4H&$#%,4HSTAF,4HNORM,11*4H&$#%,
             4HOUTP,4HINPU/
```

ändras till:

```
    DATA CM2/4HGRAF,4HXWRI,4HMAXM,4HIWRI,
             4HGROU,4HSTAF,4HNORM,11*4H&$#%,
             4HOUTP,4HINPU/
```

ÄNDRING I OPCOM

Vid aktivering av varje beräkningsrutin exekveras en uppstartrutin som har till uppgift att utföra följande:

1. Beräkna antalet ord som rutinen använder för lagring av data och reservera motsvarande utrymme i IPA med hjälp av rutinen RS. Observera att ett reellt tal använder två ord.

2. Beräkna adresser till rutinens areor ("vektorer") som skall vara tillgängliga för andra program. De beräknade adresserna placeras i LIST(IL+4), LIST(IL+5), och LIST(IL+6) där IL är en pekare på nodens första element och LIST vektorn som innehåller alla nodhuvuden. Högsta och lägsta adressen beräknas av RS och lagras i LIST(IL+7) respektive LIST(IL+3).

3. Inför erforderliga värden i parametervektorn IPA. Nödvändig information från operatören erhåller uppstartrutinen med hjälp av kommandon av typen fråga/svar.

Exempel:

Uppstartrutinen för GROU, GROUST, finns listad på sidan C45.

Ett anrop av uppstartrutinen införs i OPCOM med satsnummer: 99+rutinnamnets läge i CM2.

Exempel:

GROUST införes i OPCOM genom att satsen:

```
104   CONTINUE
```

ändras till:

```
104   Call GROUST(LIST,IL,IPA,IW2)
```

ÄNDRING I BG

I BG införes anropet av rutinen med satsnummer:

210+rutinnamnets läge i CM2.

Exempel:

GROU införes i BG genom att satsen:

```
   215     CONTINUE
```

ändras till:

```
   215     CALL GROU(IPA(K6),IPA(K2),IPA(K3),IPA(K5),IPA(K1)).
```

APPENDIX B

INSÄTTNING AV EN NY RUTINTYP I FÖRGRUNDSPROGRAMMET

Insättning av en ny rutintyp kräver ändring i följande program:

1. BG
2. FG
3. eventuellt i OPEN

Som exempel används insättning av rutinerna SINU, som finns beskriven i kapitel 4.1.7, och PRBG, som beskrivs i kapitel 4.1.4.

ÄNDRING I BD

1. Vektorn CM3 införs i rutinens namn (4 tecken). Lediga platser är markerade med tecknen & $ # %. Rutiner som kräver uppstartrutin införs på en ledig plats med större elementnummer än 10. Antalet program är begränsat till 20 stycken.

   Exempel:

   Rutinnamnet SINU placeras i 6:e vektorelementet och PRBG i 11:e.

   Satsen:

   ```
   DATA CM3/4HPIDR,4H&$#%,4HSTEP,4HADDS,
            4HRAMP,4H&$#%,4HPULS,3*4H&$#%,
            4H&$#%,4HNODI,7*4H&$#%,4HIGJN/
   ```

   ändras till:

   ```
   DATA CM3/4HPIDR,4H&$#%,4HSTEP,4HADDS,
            4HRAMP,4HSINU,4HPULS,3*4H&$#%,
            4HPRBG,4HNODI,7*4H&$#%,4HIGJN/
   ```

2. I vektorn IOP införes rutinens parameternamn på en plats som motsvarar rutinnamnets läge i CM3.

Parameternamnen placeras i följande inbördes ordning:

1. ändringsbara parametrar
2. statusparametrar
3. utsignal

Parametervärdena kommer att lagras på följande platser i vektorn IPF:

1:a parametern	IPF(K1)
2:a parametern	IPF(K1+1)
3:e parametern	IPF(K1+2)

osv.

Utsignalen kommer att lagras i vektorelementet IPF(K2-2).

Exempel:

Parameternamnen tillhörande SINU och PRBG införes i vektorn IOP och satsen DATA IOP... får följande utseende:

```
*DATA IOP/2HG=,2HTI,2HTD,2HGD,2HYO,2HRI,2HD=,2HY=,    ;PIDR
*2HAM,2HX=,2HY=,                                      ;STEP
*2HW1,2HW2,2HW3,2HY=,                                 ;ADDS
*2HXK,2HX=,2HY=,                                      ;RAMP
*2HAM,2HFR,2HDF,2HX=,2HY=,                            ;SINU
*2HAM,2HFR,2HPL,2HX=,2HY=,                            ;PULS
*2HAM,2HY=,                                           ;PRBG
*2HXM,2HSG,2HY=,                                      ;NODI
*65*2H  ,                                             ;Ej använda
*2HLI,2HCL/                                           ;Används av
                                                        systemet
```

Rutinnamnen för parametrarna är listade efter ";".

3. I vektorn IRD införes slutligen information om antalet parametrar och antalet ändringsbara parametrar på en plats som motsvarar rutinnamnets läge i vektorn CM3. Vektorelementets värde beräknas enligt formeln:

   IRD()=100*antalet parametrar (totala) + antalet ändringsbara

   Kräver rutinen en uppstartrutin sätts talet negativt.

   Exempel:

   Rutinerna SINU och PRBG har totalt 5 respektive 3 parametrar och 3 respektive 1 ändringsbara parametrar. Vidare

har rutinen PRBG en uppstartrutin. Detta resulterar i att:

satsen:

```
DATA IRD/805,0,301,403,301,0,503,3*0,
        0,-302,7*0,2424/
```

ändras till:

```
DATA IRD/805,0,301,403,301,503,503,3*0,
        -201,-302,7*0,2424/
```

ÄNDRINGAR I FG

I FG införes ett subrutinanrop av en ny rutin i satsen med satsnummer 100+rutinnamnets läge i vektorn CM3.

Exempel:

Anropet av rutinen SINU införes i FG genom att satsen:

```
106   CONTINUE
```

ändras till:

```
106   CALL SINU(IPF(K1+6),IPF(K1),IPF(K1+2),TS,IPF(K1+4),
      *IPF(K2-2))
```

Rutinen PRBG införes genom att satsen:

```
111   CONTINUE
```

ändras till:

```
111   CALL PRBG(IPF(K2),IPF(K3),IPF(K2-2),IPF(K5),IPF(K1))
```

ÄNDRING I OPEN

För rutiner med uppstartrutin krävs en ändring i OPEN, för övriga rutiner krävs ej någon ändring.

Uppstartrutinen är antingen en separat subrutin eller ett antal satser inskjutna i OPEN.

Ändringen består i att anrop av en subrutin eller motsvarande satser införs, med satsnummer 10*(rutinnamnets läge i CMD3).

Exempel:

Antag att en rutin med namnläget 13 i CMD3 behöver lagringsutrymme för två reelltalsvektorer med 3 resp 2 element, samt för en heltalsvektor med 5 element. Då införs följande satser i OPEN:

```
C       Addera till de övriga parametrarna, de antal ord som
C       vektorerna kräver.
  130   RS=RS+3*2+2*2+5
C       Tilldela arbetsvektorn IW3 de olika vektorernas
        begynnelsevärden.
C       Tilldela alla elementen i första vektorn värdet -1.0.
        DO 131  I=1,6,2
  131   CALL CV(IW3(I),-1.0)
C       Nollställ andra vektorn.
        DO 132  I=7,10,2
  132   CALL CV(IW3(I),0.0)
C       Tilldela alla elementen i tredje vektorn värdet 5.
        DO 133  I=11,15
  133   IW3(I)=5
C       Beräkna startadresserna till vektorerna två och tre.
        IP5=7
        IP6=11
C       Klart'
        GO TO 210
```

OPEN ser till att de beräknade adresserna får absolutvärdena och att de inkopieras i nodhuvudet, samt att IW3 inkopieras i IPF. De tre vektorerna kan refereras på följande sätt i förgrundsprogrammet:

```
Vektor 1    IPF(K2)
Vektor 2    IPF(K3)
Vektor 3    IPF(K4)
```

APPENDIX C

This appendix contains all program lists in alphabetical order.

TABLE OF CONTENTS

```
C-----------------------------------------------------------------------
C     NAME: ADDS                                              NUMBER
C     ----                                                    ------
C
C     SUBTITLE: CALCULATING A SUM
C     ----------
C
C     LANGUAGE: FORTRAN IV
C     ----------
C
C     KEYWORDS: WEIGHTING, SUM
C     ----------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON      DATE:1977-07-05
C     -------------                               ----
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                               VERSION:
C     ---------                                               --------
C
C-----------------------------------------------------------------------
C
C     PURPOSE
C     =======
C
C     CALCULATES A WEIGHTING SUM OF THREE SIGNALS
C
C     USAGE
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     -------------
C
C     ARGUMENTS:
C     ----------
C
C     ADDS(X1,X2,X3,W1,W2,W3)
C
C     X1   -INPUT VALUE 1, (I)
C     X2   -INPUT VALUE 2, (I)
C     X3   -INPUT VALUE 3, (I)
C     W1   -WEIGHTING FACTOR 1, (I)
C     W2   -WEIGHTING FACTOR 2, (I)
C     W3   -WEIGHTING FACTOR 3, (I)
C     Y    -CALCULATED SUM, (O)
C
C
C
C
C
```

```
      C     CHARACTERISTICS
      C     ===============
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 30 WORDS
      C
      C-----------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE ADDS( X1, X2, X3, W1, W2, W3, Y )
      C
0002        Y=W1*X1+W2*X2+W3*X3
      C
0003        RETURN
0004        END
```

```
        .NLIST TTM
        .TITLE  ADIN
;
;       ANALOG IN, FORTRAN CALLER
;
;       AUTHOR: SVEN ERIK MATTSSON 1977-08-29
;       REVISED:
;       REVISION NUMBER: 0
;       NOTE:   THE FUNCTION ADIN(ICHAN) READS SIGNALS
;               IN A FORTRAN PROGRAM FROM AN AD-CONVERTER,
;               WHERE ICHAN IS THE MULTIPLEXER ADDRESS OF THE
;               MODULE. THE MODULE IS ASSUMED TO BE ZERO.
;               CHAN MUST BE AN INTEGER 0-15 DECIMAL.
;               HOWEVER, NO CHECKS ARE MADE ON ICHAN AND THE
;               FOUR LOWEST BITS OF ICHAN ARE USED AS THE
;               MULTIPLEXER ADDRESS. THE VALUE OF ADIN IS
;               IN THE INTERVAL [-1.0,1.0).
;       EXCECUTION TIME: 0.23-0.42 MILLISECONDS (TYPICAL 0.34)
;
        .MCALL .REGDEF
        .REGDEF
        .GLOBL ADIN
        .CSECT  AD
;
;       CONVERTER REGISTERS
;
        COBA    = 167772        ;CONVERTER OUTPUT BUFFER ADDRESS
        CIBA    = 167774        ;CONVERTER  INPUT BUFFER ADDRESS
;
;       ENTRY POINT
;
ADIN:   MOV     @2(R5),R0       ;GET ICHAN
        BIC     #360,R0         ;CALCULATE THE ADDRESS
        BIS     #177400,R0
        MTPS    #340            ;INTERRUPT OFF
        MOV     R0,@#COBA       ;ADRRESS THE AD-CONVERTER
1$:     MOV     @#CIBA,R1       ;IS THE AD-CONVERTER READY?
        BMI     1$              ;NO-TRY ONCE MORE
        MTPS    #0              ;INTERRUPT ON
        TST     R1              ;IS VALUE=0?
        BNE     2$              ;NO-GO ON
        CLR     R0              ;YES-ADIN:=0
        CLR     R1
        RTS     PC              ;CONVERSION DONE-RETURN
2$:     CLR     R2              ;R2:=0
        ASL     R1              ;GET RID OF THE FOUR FIRST BITS
        ASL     R1
        ASL     R1
        ASL     R1
        BPL     3$              ;IS VALUE POSITIVE?
        NEG     R1              ;NO-R1:=ABS(R1) AND
        INC     R2              ;REMEMBER THAT VALUE IS NEGATIVE
3$:     MOV     #201,R0         ;SET UP START VALUE FOR THE
                                ;EXPONENT IN OFFSET BINARY
        ASL     R1              ;GET RID OF THE SIGN BIT
                                ;IS VALUE=-1?
        BEQ     5$              ;YES-EXPONENT AND MANTISSA IS OK
4$:     DEC     R0              ;R0:=R0-1
        ASL     R1              ;R1:=R1/2-WAS R1)=1/2?
        BCC     4$              ;NO-GO AND FIND A 1 TO HIDE
;
;       AT THIS POINT R0 CONTAINS THE EXPONENT AND
;       R1 THE 16 MOST SIGNIFICANT BITS OF THE MANTISSA
;       (THE MOST SIGNIFICANT BIT IS HIDDEN) AND
```

```
;       R2 THE SIGN. PUT THESE PIECES TOGETHER AND
;       FORM ADIN.
;
5$:     ROR     R2
        RORB    R0
        ROR     R1
        SWAB    R1
        MOVB    R1,-(SP)
        MOVB    R0,1(SP)
        MOV     (SP)+,R0
        CLRB    R1                  ;CONVERSION DONE
        RTS     PC                  ;RETURN
        .END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: BD
C       ----
C
C       SUBTITLE: BLOCK DATA
C       ----------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ----------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 6-SEP-77
C       ----------------                                 ------
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, LUND
C
C-------------------------------------------------------------------------------
C
C       PURPOSE
C       =======
C
C       ESTABLISH AND DEFINE COMMON BLOCKS, ASSIGN VARIABLES
C       AND ARRAYS TO THOSE BLOCKS, AND ASSIGN INITIAL VALUES TO
C       THE COMPONENTS
C
C       USAGE
C       =====
C
C       COMMON BLOCKS:
C       --------------
C       BG      -CONTAINING LIST- AND PARAMETER ARRAYS FOR
C                BACKGROUND JOB
C       CMD     -CONTAINING 4-CHARACTERS COMMAND NAME AND 2-
C                CHARACTERS PARAMETER NAME
C       IM      -CONTAINING 1-CHARACTER CONSTANTS
C       RD      -CONTAINING THE NUMBER OF PARAMETERS OF THE
C                NODETYPES IN THE FOREGROUND JOB
C       FG      -CONTAINING ARRAYS FOR SENDING AND RECEIVING
C                MESSAGES AND INFORMATION BETWEEN THE JOBS
C
C       NOTE:
C       -----
C
C       SAY THAT A ROUTINE HAS:
C
C       1.     TOTALLY N1 PARAMETERS
C       2.     N2 NOT STATUS PARAMETERS
C
C       THEN
C                                IRD( )=100*N1+N2
C
```

```
      C       IF THE NODE TYPE HAS A START-UP ROUTINE
      C
      C                                  IRD( )=-100*N1+N2
      C
      C-------------------------------------------------------------------------------
      C
0001          BLOCK DATA
      C
0002          LOGICAL*1 ICM
      C
0003          COMMON /BG/ LIST(120)
0004          COMMON /CMD/CM1(20),CM2(20),CM3(20),IOP(100)
0005          COMMON /IM/ ICM(7)
0006          COMMON /RD/ IRD(20)
      C
0007          DATA LIST /120*-1/
      C
0008          DATA CM1/4HOPEN,4HACTI,4HSTOP,4HSTOR,4HSTAR,4HSTOC,4HSTAC,
             *4HCRNF,4HLSTF,4HLSTB,10*4H&$$%/
      C
0009          DATA CM2/4HGRAF,4HXWRI,4HMAXM,4HIWRI,4HGROU,4HSTAF,4HNORM
             *,11*4H&$$%,4HOUTP,4HINPU/
      C
0010          DATA CM3/4HPIDR,4H&$$%,4HSTEP,4HADDS,4HRAMP,4HSINU,4HPULS,
             *3*4H&$$%,4HPRBG,4HNODI,7*4H&$$%,4HIGJN/
      C
0011          DATA IOP/2HG=,2HTI,2HTD,2HGD,2HYO,2HRI,2HD=,2HY=,
             *2HAM,2HX=,2HY=,
             *2HW1,2HW2,2HW3,2HY=,
             *2HXK,2HX=,2HY=,
             *2HAM,2HFR,2HDF,2HX=,2HY=,
             *2HAM,2HFR,2HPL,2HX=,2HY=,
             *2HAM,2HY=,
             *2HXM,2HSG,2HY=,
             *65*2H   ,
             *2HLI,2HCL/
      C
0012          DATA ICM/1H,,1H.,1H/,1H$,1H:,1H=,1H /
      C
0013          DATA IRD/805,0,301,403,301,503,503,3*0,-201,-302,7*0,2424/
      C
0014          END
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 17:18:56                PAGE 001

      C-----------------------------------------------------------------------
      C     NAME:BELL                                          NUMBER:
      C     ~~~~~                                              ~~~~~~~
      C
      C     SUBTITLE: RING BELL ONCE.
      C     ~~~~~~~~~~~~
      C
      C     LANGUAGE: FORTRAN IV+RT11- FORTRAN
      C     ~~~~~~~~~~~~
      C
      C     KEYWORDS:RING
      C     ~~~~~~~~~~~~
      C
      C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON       DATE:1977-07-05
      C     ~~~~~~~~~~~~~~~~~~~~                           ~~~~~~
      C
      C     INSTITUTE:
      C     ~~~~~~~~~~
      C     DEPARTMENT OF AUTOMATIC CONTROL
      C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
      C
      C     ACCEPTED:                                          VERSION:
      C     ~~~~~~~~~                                          ~~~~~~~~
      C-----------------------------------------------------------------------
      C
      C     USAGE:
      C     ======
      C     PROGRAM TYPE: SUBROUTINE
      C     ~~~~~~~~~~~~~~~~~~~~~~~~
      C
      C     CHARACTERISTICS:
      C     ================
      C
      C     SUBROUTINES REQUIRED:
      C     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      C     FROM SYSTEM LIBRARY: ITTOUR
      C
      C     SIZE:
      C     ~~~~~
      C     PDP11/03: 17 WORDS
      C
      C-----------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE BELL
      C
      C     RING BELL ONCE.
      C
0002     10 IF( ITTOUR( '007 ).NE.0 ) GO TO 10
      C
0004        RETURN
0005        END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: BG
C       ----
C
C       SUBTITLE: MAIN PROGRAM IN BACKGROUND JOB
C       ----------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ----------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 7-SEP-77
C       -------------------                              ------
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       BG STARTS BACKGROUND JOB BY ASKING QUESTIONS AND USING THE
C       OPERATORS ANSWERS, BY ORGANICZING THE COMMUNIZATION WITH MASS
C       STORAGE AND BY ALLOCATING SPACE ON MASS STORAGE.
C       THE PROGRAM ORGANIZES THE TRANSFERING OF THE COLLECTED MEASUREMEN
C       DATA, THE TRANSFERING OF CALCULATED DATA AND THE RUNNING OF
C       CALCULATING PROGRAMS. FURTHER IT ORGANIZES THE TRANSFERING OF
C       AND INFORMATION TO AND FROM FOREGROUND JOB AND THE SUBROUTINE
C       OPCOM AND THE EXECUTION OF COMMANDS FROM OPCOM.
C
C       USAGE
C       =====
C
C       COMMON BLOCKS:
C       --------------
C
C       BG          -CONTAINING LIST- AND PARAMETER ARRAYS FOR
C                    BACKGROUND JOB
C
C       BUFF        -CONTAINING BUFFERS FOR TRANSFERING DATA BETWEEN
C                    FOREGROUND JOB, BACKGROUND JOB AND MASS STORAGE.
C
C       LINK        -CONTAINING LINKAGE INFORMATION FOR COMPLETITION
C                    ROUTINES.
C
C       PAV,FLA     -CONTAINING FLAGS AND POINTERS WITH INFORMATION
C                    ABOUT COLLECTING AND CALCULATING BLOCKS.
C
C       CMD         -CONTAINING 4-CHARACTERS COMMAND NAME AND 2-
C                    CHARACTERS PARAMETER NAME.
C
```

```
C       IM        -CONTAINING 1-CHARACTER CONSTANTS.
C
C       RD        -CONTAINING THE NUMBER OF PARAMETERS OF THE
C                  NODE TYPES IN THE FOREGROUND JOB.
C
C       FG        -CONTAINING ARRAYS FOR SENDING AND RECEIVING
C                  MESSAGES AND INFORMATION BETWEEN THE JOBS.
C
C       WRK       -WORK AREA
C
C
C       CHARACTERISTICS
C       ===============
C
C        SIZE:
C        -----
C        PDP11/03: 711 WORDS
C
C-------------------------------------------------------------------------------
C
C     ************************************
C     *                                  *
C     *          BACKGROUND PROGRAM      *
C     *                                  *
C     ************************************
C
0001          EXTERNAL DFBT,COP
0002          DIMENSION DF(2),IEK(12)
0003          LOGICAL*1 ICM
      C
0004          COMMON /BG/ LIST(120),IPA(512)
0005          COMMON /BUFF/ IB1(100),IB2(512),IB4(256)
0006          COMMON /LINK/ LI1(4),LI2(4)
0007          COMMON /PAV/ I1,I2,I3,I4,I5,I6,I7,I8,IC1,IR1,
             *NB1,IR2,NB2,IC2,I9,I10
0008          COMMON /FLA/ IFL1,ICN,NBN,ICO,IFL2
0009          COMMON /CMD/ CM1(20),CM2(20),CM3(20),IOP(100)
0010          COMMON /IM/ ICM(7)
0011          COMMON /RD/ IRD(20)
0012          COMMON /FG/ IV(100)
0013          COMMON /WRK/ IW1(50),IW2(50),IW3(50),IW4(50)
      C
0014          EQUIVALENCE (IEK(3),K1),(IEK(4),K2),(IEK(5),K3),(IEK(6),K4),
             *(IEK(7),K5),(IEK(8),K6),(IEK(9),K7),(IEK(10),K8),(IEK(11),K9)
      C
0015          DATA DF/6RDX1WOR,6RK   DAT/
      C
      C       ALLOCATE TWO RT-11 CHANNELS AND MARKS THEM
      C       IN USE FOR THE FORTRAN I/O SYSTEM
      C
0016          ICN=IGETC( )
0017          ICO=IGETC( )
      C
      C       ALLOCATE SPACE FOR NBN BLOCKS ON FILE NAMED
```

```
      C     DX1:WORK.DAT AND CREATES A TENTATIVE ENTRY
      C     FOR IT.
      C     NBN IS EQUAL THE LARGER OF EITHER ONE-HALF THE
      C     LARGEST EMPTY SEGMENT OR THE ENTIRE SECOND
      C     LARGEST EMPTY SEGMENT.
      C
0018        NBN=IENTER(ICN,DF,0)
0019        IF(NBN.LE.0) STOP'ENTER FAILURE'
0021        NBN=NBN-1
      C
      C     ADD AVAILABLE ELEMENTS TO THE QUEUE.
      C
0022        IF(IQSET(7).NE.0) STOP'NOT ENOUGH FREE SPACE FOR
           * QUEUE ELEMENTS'
      C
      C     READ AND SEND INFORMATION ABOUT MAIN SAMPLE PERIOD,
      C     NUMBER OF INPUT SIGNALS AND OUTPUT SIGNALS.
      C
0024        IB1(1)=13
0025        IB1(2)=0
0026      1 WRITE(7,501)
0027    501 FORMAT(1X,$,'MAIN SAMPLE PERIOD:')
0028        READ(5,5) IB1(3)
0029        IF(IB1(3).GE.2.AND.IB1(3).LE.32767) GO TO 2
      C
0031        WRITE(7,502)
0032    502 FORMAT(1X,'?ILL CMD?')
0033        GO TO 1
      C
0034      2 WRITE(7,503)
0035    503 FORMAT(1X,$,'NUMBER OF INPUT SIGNALS:')
0036        READ(5,5) IB1(4)
0037        IF(IB1(4).GE.1.AND.IB1(4).LE.8) GO TO 3
0039        WRITE(7,502)
0040        GO TO 2
      C
0041      3 WRITE(7,504)
0042    504 FORMAT(1X,$,'NUMBER OF OUTPUT SIGNALS:')
0043        READ(5,5) IB1(5)
0044        IF(IB1(5).GE.1.AND.IB1(5).LE.4) GO TO 4
0046        WRITE(7,502)
0047        GO TO 3
      C
0048      4 CALL ISDAT(IB1,5)
0049      5 FORMAT(I10)
      C
      C     CONVERT A 2-WORD INTERNAL TIME FORMAT TO INTEGER*4 FORMAT
      C     AND THEN CONVERT IT TO REAL*4 VALUE
      C
0050        CALL JJCVT(IB1)
0051        TSM=AJFLT(IT1)*0.02
      C
      C
0052        I1=0
```

```
0053          NI=IB1(4)*2+2
0054          NU=IB1(5)*2
0055          I2=NI+NU
0056          NU=NU+18
0057          I3=256/I2
0058          I4=0
0059          I5=0
0060          I6=1
0061          I7=1
0062          I10=0
0063          IR1=-1
0064          IR2=-1
0065          NB1=NBN
0066          NB2=NBN
0067          IC1=ICN
0068          IC2=ICN
0069          IFL1=0
        C
        C     ENTER THE SUBROUTINE DFBT WHEN A MESSAGE FROM
        C     FOREGROUND JOB IS RECEIVED.
        C
0070          CALL IRCVDF(IB1,56,LI1,DFBT)
        C
        C     SET FLAG NO DATA AVAILABLE.
        C
0071       10 I9=0
        C
        C     MESSAGE FROM CONSOLE TERMINAL?
        C
0072          CALL TCMD
        C
        C     SCHEDULE THE SUBROUTINE COP TO BE RUN AS
        C     AN ASYNCHRONOUS ROUTINE.
        C
0073          CALL ITIMER(0,0,0,0,LI2,ID,COP)
        C
        C     WAIT UNTIL DATA IS AVAILABLE.
        C
0074       15 IF(I9) 10,15,16
        C
        C     SUSPEND EXECUTION UNTIL ALL INPUT/OUTPUT ON CHANNEL IC2
        C     ARE COMPLETE.
        C
0075       16 CALL IWAIT(IC2)
0076          DO 100 I=I7,I8
        C
        C     MESSAGE FROM CONSOLE TERMINAL?
        C
0077          CALL TCMD
        C
        C     COPY ONE SAMPLE BLOCK FROM VECTOR IB4 TO VECTOR IFA
        C
0078          J1=(I-1)*I2
0079          DO 17 K=1,NI
```

```
0080          J1=J1+1
0081       17 IPA(K)=IB4(J1)
      C
0082          DO 18 K=19,NU
0083          J1=J1+1
0084       18 IPA(K)=IB4(J1)
      C
      C       RUN ACTIVATED NODS.
      C
0085          DO 100 K=1,120,12
0086          IF(LIST(K)) 100,200,300
0087      200 J=LIST(K+2)/1000
      C
0088          DO 210 J2=3,11
0089      210 IEK(J2)=LIST(K+J2)
0090          LIST(K)=LIST(K+1)
      C
0091          GO TO (211,212,213,214,215,216,217,218,219,220,
             *230,230,230,230,230,230,230,230,230,230),J
      C
0092      211 CALL GRAF(IPA(K6),IPA(K1),IPA(K2),IPA(K3),IPA(K4),
             *IPA(K5),IW1)
0093          GO TO 300
0094      212 CALL XWRI(IPA(K6),IPA(K7),IPA(K8),IPA(K9),IPA(K1),CM1,
             *IPA(K5),IPA,IW1)
0095          GO TO 300
0096      213 CALL MAXM(IPA(K6),IPA(K2),IPA(K4),IPA(K1),IPA(K3),IPA(K5))
0097          GO TO 300
0098      214 CALL IWRI(IPA(K6),IPA(K7),IPA(K8),IPA(K9),IPA(K1),CM1,
             *IPA(K5),IPA,IW1)
0099          GO TO 300
0100      215 CALL GROU(IPA(K6),IPA(K2),IPA(K3),IPA(K5),IPA(K1))
0101          GO TO 300
0102      216 CALL STAF(IPA(K6),IPA(K1),IPA(K2),IPA(K3),IPA(K4),
             *IPA(K5-2),IPA(K5))
0103          GO TO 300
0104      217 CALL NORM(IPA(K6),IPA(K7),IPA(K8),IPA(K1))
0105          GO TO 300
0106      218 CONTINUE
0107          GO TO 300
0108      219 CONTINUE
0109          GO TO 300
0110      220 CONTINUE
0111          GO TO 300
0112      230 CONTINUE
0113      300 LIST(K)=LIST(K)-1
0114      100 CONTINUE
      C
0115          I7=I8+1
0116          IF(I8.EQ.I3) I7=1
      C
      C       SET FLAG NO DATA AVAILABLE.
      C
0118          I9=0
```

```
0119          GO TO 10
      C
0120          STOP
0121          END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: COP
C       ----
C
C       SUBTITLE: COPIES MEASUREMENT DATA FROM MASS STORAGE OR
C       ---------  BUFFER TO CALCULATING PROGRAMS
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ---------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 7-SEP-77
C       ------------                                     ----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       COPIES MEASUREMENT DATA FROM MASS STORAGE OR BUFFER TO
C       CALCULATING PROGRAMS
C
C       DEPENDING ON THE DIFFERENCE IN TIME BETWEEN COLLECTING
C       AND CALCULATING THE MEASUREMENT DATA THE FOLLOWING
C       SITUATIONS CAN OCCUR:
C       1.CALCULATING ON A FILE IN MASS STORAGE IS FINISHED
C         BUT THERE EXISTS ANOTHER FILE,THEN START ON THE NEW FILE.
C         POSSIBLE LOST BLOCKS ARE COUNTED AND AN ERROR MESSAGE IS GIVEN.
C
C       2.THE FILE ON MASS STORAGE CONTAINS MORE DATA. A BLOCK IS  THEN
C         COPIED TO THE CALCULATING PROGRAMS.
C
C       3.THERE ARE FOR THE PRESENT NO MORE BLOCKS ON MASS STORAGE TO
C         CALCULATE ON, BUT THE BUFFER CONTAINS A FULL BLOCK. COPY IT
C         TO CALCULATING PROGRAMS.
C
C       4.THE CALCULATING PROGRAMS IS FOR THE PRESENT KEEPING UP WITH
C         THE COLLECTING OF DATA. COPY DIRECTLY FROM THE BUFFER TO
C         CALCULATING PROGRAMS.
C
C
C       USAGE
C       =====
C
C       ARGUMENTS:
C       ----------
C       COP( ID )
C       ID       -NOT USED
C
```

```
      C       COMMON BLOCKS:
      C       ~~~~~~~~~~~~~~
      C       BUFF     -CONTAINING BUFFERS FOR TRANSFERING DATA BETWEEN FORE-
      C                 GROUND JOB, BACKGROUND JOB AND MASS STORAGE.
      C       FLA,PAV -CONTAINING FLAGS AND POINTERS WITH INFORMATION ABOUT
      C                 COLLECTING AND CALCULATING BLOCKS.
      C
      C       CHARACTERISTICS
      C       ===============
      C
      C       SUBROUTINE REQUIRED:
      C       ~~~~~~~~~~~~~~~~~~~~
      C       FROM SYSTEM LIBRAY: IREAD
      C
      C       SIZE:
      C       ~~~~~
      C       PDP11/03:209 WORDS
      C
      C-------------------------------------------------------------------------------
      C
0001          SUBROUTINE COP(ID)
0002          COMMON /BUFF/IB1(100),IB2(256),IB3(256),IB4(256)
0003          COMMON /PAV/I1,I2,I3,I4,I5,I6,I7,I8,IC1,IR1,NB1,IR2,NB2,IC2,I9
             *,I10
0004          COMMON /FLA/IFL1,ICN,NBN
0005          IF(IR2.LT.NB2.OR.IC1.EQ.IC2) GO TO 5
      C
      C       START CALCULATING ON A NEW FILE.
      C
0007          MS=I10-IR2+NB2
      C
0008          IC2=IC1
0009          NB2=NB1
0010          IF(MS.LE.0) GO TO 7
0012          I6=I6+MS
0013          I7=1
0014          WRITE(7,8) MS
0015        8 FORMAT(1X,I5,' MISSING BLOCKS')
0016          MS=0
0017        7 IR2=-MS-1
0018          IFL1=-1
0019        5 IF(I6-I1) 10,20,30
      C
      C       READ MEASUREMENT DATA BLOCK FROM MASS STORAGE.
      C
0020       10 IF(IR2.LT.NB2) GO TO 11
0022          I9=-1
0023          RETURN
0024       11 IR2=IR2+1
0025          CALL IREAD(256,IB4,IR2,IC2)
0026          I6=I6+1
0027          I8=I3
0028          GO TO 80
      C
```

```
      C     READ MEASUREMENT DATA BLOCK FROM BUFFER.
      C
0029     20 I8=I3
0030        J1=I2*(I7-1)+1
0031        J2=I2*I8
0032        I6=I6+1
0033        IR2=IR2+1
0034        IF(I5.GE.256) GO TO 40
0036        GO TO 60
      C
      C     READ AVAILABLE MEASUREMENT DATA FROM BUFFER.
      C
0037     30 I8=I4
      C
      C     ANY MESUREMENT DATA AVAILABLE?
      C
0038        IF(I7.LE.I8) GO TO 35
0040        I9=-1
0041        RETURN
      C
      C
0042     35 J1=I2*(I7-1)+1
0043        J2=I2*I8
0044        IF(I5.GT.256) GO TO 60
      C
      C     FETCH FROM BUFFER IB2
      C
0046     40 DO 50 I=J1,J2
0047     50 IB4(I)=IB2(I)
0048        GO TO 80
      C
      C     FETCH FROM BUFFER IB3
      C
0049     60 DO 70 I=J1,J2
0050     70 IB4(I)=IB3(I)
0051     80 I9=1
      C
0052        RETURN
0053        END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: CRNF
C       ----
C
C       SUBTITLE: CREATES A NEW FILE FOR DATA
C       ---------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ---------
C
C       IMPLEMENTOR:LEIF RADING, TYKE PAULSSON          DATE: 15-SEP-77
C       ------------                                    ----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       CREATES A NEW FILE FOR DATA ON MASS STORAGE, IF LEGAL.
C       MAX NUMBER OF ACTIVE DATAFILES IS TWO.
C       IT IS LEGAL IF:
C
C       1.ZERO OR ONE FILE IS ACTIVE
C       2.TWO FILES ARE ACTIVE, BUT ONE TREATED-THIS FILE
C         IS MADE PERMANENT AT THIS MOMENT
C
C       ERROR AND INFORMATION MESSAGES ARE GIVEN
C
C       USAGE
C       =====
C
C       ARGUMENTS:
C       ----------
C
C       CRNF(IC,ICO,NB,IFL1,IW)
C
C       IC,ICO   -CHANNEL NUMBERS, (I)
C       NB       -NUMBER OF ALLOCATED BLOCKS, (O)
C       IFL1     -FLAG, (I)
C       IW       -WORK AREA, SIZE(50)
C
C       CHARACTERISTICS
C       ===============
C
C       SIZE:
C       -----
C       PDP11/03: 200 WORDS
C
```

```
      C       SUBROUTINES REQUIRED:
      C       ---------- ---------
      C       FROM SYSTEM LIBRARY: CLOSEC,ICSI,IENTER
      C
      C
      C-------------------------------------------------------------------------------
      C
0001          SUBROUTINE CRNF(IC,ICO,NB,IFL1,IW)
      C
      C       IC1 NEW CHANNEL NUMBER.
      C       IC2 OLD CHANNEL NUMBER.
      C       NB  INTEGER NUMBER OF BLOCKS ALLOCATED FOR THE FILE.
      C
      C       IFL1          )0        TWO FILES ALREADY ACTIVE.
      C       IFL1          =0        OPEN A NEW FILE
      C       IFL1          (0        MAKE OLD FILE PERMANENT AND OPEN A NEW FILE
      C
0002          DIMENSION IW(1),EXT(2),IDEV(4)
0003          DATA EXT/6RDATDAT,6RDATDAT/
0004          DATA IDEV/3RDX ,3RDX0,3RDX1,3RDK /
0005          IF(IFL1) 10,20,95
      C
      C       MAKE FILE ASSOCIATED WITH CHANNEL NUMBER IC2 PERMANENT.
      C
0006       10 CALL CLOSEC(ICO)
0007          IFL1=0
0008          WRITE(7,500)
0009      500 FORMAT(1X,'OLD FILE PERMANENT'/)
      C
      C       CALL THE RT-11 COMMAND STRING INTERPRETER IN SPECIAL
      C       MODE TO PARSE A COMMAND STRING AND RETURN FILE
      C       DESCRIPTORS.
      C
0010       20 I=ICSI(IW,EXT,,,0)
0011          IF(I.EQ.0) GO TO 50
0013          WRITE(7,501)
0014      501 FORMAT(1X,'?ILL CMD?')
0015          RETURN
      C
      C       TEST IF THE SPECIFIED DEVICE EXIST.
      C
0016       50 DO 60 I=1,4
0017          IF(IDEV(I).EQ.IW(1)) GO TO 70
0019       60 CONTINUE
0020          WRITE(7,502)
0021      502 FORMAT(1X,'?ILL DEV?')
0022          RETURN
      C
      C       ALLOCATE SPACE ON THE SPECIFED DEVICE AND CREATE A
      C       TENTATIVE DIRECTORY ENTRY FOR THE FILE.
      C       ASSOCIATE CHANNEL NUMBER IC2 WITH THE NAMED FILE.
      C
0023       70 NB=IENTER(ICO,IW,IW(5))
0024          IF(NB) 80,80,90
```

```
0025      80 WRITE(7,503)
0026     503 FORMAT(1X,'NOT ENOUGH ROOM ON DEVICE FOR FILE SIZE REQUESTED')
0027         RETURN
0028      90 IFL1=1
0029         NB=NB-1
0030         IT=IC
0031         IC=ICO
0032         ICO=IT
0033         RETURN
0034      95 WRITE(7,504)
0035     504 FORMAT(1X,'TWO FILES ALREADY ACTIVE')
0036         RETURN
0037         END
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 17:24:48                PAGE 001

      C-------------------------------------------------------------------------------
      C
      C       NAME: CV
      C       ----
      C
      C       SUBTITLE: CONVERT
      C       ---------
      C
      C       LANGUAGE: FORTRAN + RT-11-FORTRAN
      C       ---------
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 15-SEP-77
      C       ------------                                     ------
      C
      C       INSTITUTE:
      C       ----------
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-------------------------------------------------------------------------------
      C
      C       PURPOSE AND METHOD
      C       ==================
      C       SETS Y=X
      C
      C       COMMENT:
      C       --------
      C       CAN BE USED TO STORE A REAL*4 NUMBER IN TWO CONTINUOUS INTEGER*2
      C       NUMBERS IN AN INTEGER-ARRAY AND VICE-VERSA.
      C
      C       USAGE
      C       =====
      C
      C       ARGUMENTS:
      C       ----------
      C
      C       CV(Y,X)
      C
      C       Y     -OUTPUT VALUE, (O)
      C       X     -INPUT VALUE, (I)
      C
      C       CHARACTERISTICS
      C       ===============
      C
      C       SIZE:
      C       -----
      C       PDP11/03: 12 WORDS
      C-------------------------------------------------------------------------------
0001          SUBROUTINE CV(Y,X)
      C
0002          Y=X
      C
0003          RETURN
0004          END
```

```
        .NLIST TTM
        .TITLE DAOUT
;
;       ANALOG OUT, FORTRAN CALLER
;
;       AUTHOR: SVEN ERIK MATTSSON 1977-09-01
;       REVISED:
;       REVISION NUMBER: 0
;       NOTE:   THE SUBROUTINE DAOUT(ICHAN,VALUE) WRITES
;               IN A FORTRAN PROGRAM ON A DA-CONVERTER,
;               WHERE ICHAN MUST BE AN INTEGER 0-13
;               DECIMAL AND VALUE A REAL NUMBER
;               IN THE INTERVAL [-1.0,1.0).
;               HOWEVER, NO CHECKS ARE MADE ON ICHAN
;               AND THE FOUR LOWEST BITS OF ICHAN ARE
;               USED AS ICHAN AND VALUE IS LIMITED TO
;               [-1.0,1.0).
;       EXCECUTION TIME: ABOUT 0.34 MILLISECONDS
;
        .MCALL .REGDEF
        .REGDEF
        .GLOBL DAOUT
        .CSECT DA
;
;       CONVERTER REGISTER
;
        COBA    = 167772        ;CONVERTER OUTPUT BUFFER ADDRESS
;
;       ENTRY POINT
;
DAOUT:  MOV     4(R5),R1        ;ADDRESS OF VALUE TO R1
        MOV     (R1),R0         ;GET THE HIGH ARGUMENT OF VALUE
        ASL     R0              ;THE EXPONENT TO THE HIGH BYTE OF R0
        ROR     R2              ;SAVE THE SIGN IN THE HIGHEST BIT OF R2
        CLRB    R0              ;R0:=EXPONENT
        SWAB    R0
        SUB     #200,R0         ;DECODE EXPONENT FROM OFFSET BINARY
        BLE     2$              ;IS ABS(VALUE))=1?
1$:     MOV     #77777,R1       ;YES-R1:=1
        TST     R2
        BPL     4$              ;IS VALUE(=-1?
        MOV     #100000,R1      ;YES-R1:=-1
        BR      4$
2$:     CMP     #-13,R0         ;IS ABS(VALUE) TOO SMALL?
        BLE     3$
        CLR     R1              ;YES-R1:=0
        BR      4$
3$:     MOVB    3(R1),-(SP)     ;GET THE 15 MOST SIGNIFICANT
        MOVB    (R1),1(SP)      ;BITS OF THE MANTISSA
        MOV     (SP)+,R1
        BIS     #100000,R1      ;DO NOT FORGET THE HIDDEN BIT
        CLC                     ;CLEAR CARRY AND
        ROR     R1              ;MAKE ROOM FOR THE SIGN BIT
        ASH     R0,R1           ;PLACE THE RADIX POINT
                                ;AFTER THE SIGN BIT
        ADD     #10,R1          ;ROUND OFF
        BMI     1$              ;7777X WILL OVERFLOW
        TST     R2              ;IS VALUE(0?
        BGE     4$
        NEG     R1              ;YES-R1:=-R1
;
;       AT THIS POINT R1 CONTAINS VALUE AS
;       TWO'S COMPLEMENT NUMBER.
;
```

```
4$:     MOV     @2(R5),R0       ;GET ICHAN
        ASHC    #-4,R0          ;WRITE ON DA-CONVERTER
        MOV     R1,@#COBA
        RTS     PC              ;RETURN
        .END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: DFBT
C       ----
C
C       SUBTITLE: HANDLE MEASUREMENT DATA AND INFORMATION TRANSFER FROM
C       ---------  FOREGROUND TO BACKGROUND JOB
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ---------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON            DATE:7-SEP-77
C       ------------                                       ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUT OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       CHECK THE MESSAGE RECEIVED FROM FOREGROUND JOB.
C       IF MEASUREMENT DATA, PUT IT INTO THE BUFFER. IF A
C       BLOCK IS FULL, SEND IT TO A FILE IN MASS STORAGE. IF
C       THE FILE IS FULL, SEE IF ANOTHER FILE EXISTS. IF SO,
C       TRANSFER TO THE NEW FILE, OTHERWISE AN ERROR MESSAGE
C       IS GIVEN.
C       IF OTHER TYPE OF MESSAGE THAN MEASUREMENT DATA, COPY IT
C       INTO IV.
C
C       USAGE
C       =====
C
C       COMMON BLOCKS:
C       --------------
C       BUFF     -CONTAINING BUFFERS FOR TRANSFERING DATA BETWEEN
C                 FOREGROUND JOB AND MASS STORAGE.
C       LINK     -CONTAINING LINKAGE INFORMATION FOR COMPLETITION
C                 ROUTINES.
C       FLA,FAV -CONTAINING FLAGS AND POINTERS WITH INFORMATION
C                 ABOUT COLLECTING AND CALCULATING BLOCKS.
C       FG       -CONTAINING ARRAYS FOR SENDING AND RECEIVING MESSAGES
C                 AND INFORMATION BETWEEN THE JOBS
C
C       CHARACTERISTICS
C       ===============
C
C       SIZE:
C       ----
C       PDP11/03:242 WORDS
C
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 17:27:50                PAGE 002

      C       SUBROUTINE REQUIRED:BELL
      C        ------------------------
      C       FROM SYSTEM LIBRAY:  IRCVDF,IWAIT,PRINT,IWRITE
      C
      C
      C-------------------------------------------------------------------------
      C
0001          SUBROUTINE DFBT
0002          EXTERNAL DFBT
0003          COMMON /BUFF/IB1(100),IB2(256),IB3(256)
0004          COMMON /PAV/I1,I2,I3,I4,I5,I6,I7,I8,IC1,IR1,NB1,IR2,NB2,IC2,I9
             *,I10
0005          COMMON /FLA/IFL1,ICN,NBN,ICO,IFL2
0006          COMMON /LINK/LI1(4)
0007          COMMON /FG/IV(100)
      C
      C
0008          IF(IB1(1).EQ.I2) GO TO 5
      C
      C       INFORMATION FROM FOREGROUND JOB.
      C
0010          J=IB1(1)+1
0011          DO 1 I=2,J
0012        1 IV(I+2)=IB1(I)
      C
      C       SET FLAG -INFORMATION RECEIVED.
      C
0013          IFL2=-1
      C
      C       QUEUE A RECEIVED DATA REQUEST AND CONTINUE EXECUTION OF
      C       ISSUING JOB UNTIL A NEW MESSAGE IS RECEIVED,THEN ENTER
      C       THE SUBROUTINE DFBT.
      C
0014          CALL IRCVDF(IB1,99,LI1,DFBT)
0015          RETURN
      C
      C       MEASUREMENT DATA FROM FOREGROUND JOB.
      C
0016        5 J1=I2+1
0017          DO 10 I=2,J1
0018          I5=I5+1
0019       10 IB2(I5)=IB1(I)
0020          I4=I4+1
      C
      C       QUEUE A RECEIVED DATA REQUEST AND CONTINUE EXECUTION OF
      C       ISSUING JOB UNTIL A NEW MESSAGE IS RECEIVED,THEN ENTER
      C       THE SUBROUTINE DFBT.
      C
0021          CALL IRCVDF(IB1,99,LI1,DFBT)
0022          IF(I4.NE.I3) RETURN
      C
      C       SEND BLOCK TO MASS STORAGE.
      C
0024          I4=0
```

```
0025          I1=I1+1
0026          IR1=IR1+1
      C
      C       SUSPEND EXECUTION UNTIL ALL INPUT/OUTPUT ON CHANNEL IC1 ARE
      C       COMPLETE.
      C
0027          CALL IWAIT(IC1)
0028          IF(I5.GT.256) GO TO 30
0030          I5=256
0031          I=1
0032          GO TO 35
0033       30 I5=0
0034          I=257
      C
      C       FILE FULL?
      C
0035       35 IF(IR1-NB1-1) 40,50,45
      C
      C       NEW FILE AVAILABLE?
      C
0036       50 IF(IC1.NE.ICN) GO TO 70
0038          CALL PRINT('')
0039          CALL PRINT('NO MORE SPACE IS AVAILABLE IN OPENED FILES')
      C
      C       RING BELL ONCE.
      C
0040          CALL BELL
0041          RETURN
      C
      C
0042       45 IF(IC1.EQ.ICN) RETURN
0044       70 I10=IR1-NB1-1
0045          IR1=0
0046          IC1=ICN
0047          NB1=NBN
      C
      C       SEND ONE BLOCK TO FILE CONNNECTED WITH CHANNEL
      C       NUMBER IC1.
      C
0048       40 IF(IWRITE(256,IB2(I),IR1,IC1).LT.0)  CALL PRINT(
            *'ERROR OCCURRED DURING TRANSFER TO MASS STORAGE')
      C
0050          RETURN
0051          END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: FG
C       ----
C
C       SUBTITLE: MAIN PROGRAM IN FOREGROUND JOB
C       ----------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ----------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 7-SEP-77
C       -----------------                                      ------
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ------------------
C
C       FG STARTS FOREGROUND JOB WITH INFORMATION RECEIVED FROM
C       BACKGROUND JOB.
C       THE PROGRAM ORGANIZES COLLECTING OF MEASUREMENT DATA, RUNNING
C       OF CONTROLLING PROGRAMS AND SENDING OF DATA TO BACKGROUND JOB.
C       FURTHER IT TAKES CARE OF INFORMATION AND COMMANDS FROM BACK-
C       GROUND JOB, EXECUTES THE COMMAND AND SENDS POSSIBLE REQUESTED
C       INFORMATION.
C
C       USAGE
C       =====
C
C       COMMON BLOCKS:
C       --------------
C
C       FL      -
C
C       CHRACTERISTICS
C       ==============
C
C       SIZE:
C       -----
C       PDP11/03: 1776 WORDS
C
C-------------------------------------------------------------------------------
C
C     ****************************************
C     *                                      *
C     *          FOREGROUND PROGRAM          *
C     *                                      *
C     ****************************************
```

```
      C
0001        EXTERNAL FLAG,FLAGR
0002        DIMENSION IV(60),IB(60),IT1(2),IT2(2)
           *,LINK1(4),LINK2(4),LF(240),IPF(400)
0003        COMMON /FL/I1
0004        DATA IPF/400*0/
      C
      C     SET ALL ZERO LEVELS EQUAL 0.0 AND SLOPES EQUAL 1.0
      C
0005        DO 5 I=29,75,4
0006      5 IPF(I)=16512
      C
      C     SET ALL NODES INACTIVE
      C
0007        DO 10 I=1,240,12
0008        LF(I)=-1
0009     10 LF(I+1)=0
      C
      C     ADD AVAILABLE ELEMENTS TO THE QUEUE
      C
0010        CALL IQSET(5)
      C
      C     RECEIVE DATA FROM BACKGROUND JOB
      C
0011        CALL IRCVDW(IB,59)
      C
      C     SETS INITIALY THAT ONE MESSAGE FROM BACKGROUND JOB ARE
      C     WAITING TO BE EXECUTED.
      C
0012        I1=0
      C
      C     INCREMENT SUSPENSION COUNTER TWICE.
      C
0013        CALL RESUME
0014        CALL RESUME
      C
      C     SET INITIALY NO REGULATION AND NO DATA COLLECTION.
      C
0015        I2=1
      C
      C     WHAT TO DO?
      C
0016     15 GO TO(20,21,22,23,24,25,26,29,31,33,35,36,38)IB(2)
      C     STOP REGULATION.
      C
0017     20 IF(I2.EQ.1.OR.I2.EQ.3) GO TO 50
0019        I2=I2-1
0020        GO TO 50
      C
      C     START REGULATION.
      C
0021     21 IF(I2.EQ.4.OR.I2.EQ.2) GO TO 50
0023        I2=I2+1
0024        GO TO 50
```

```
      C
      C     STOP COLLECTION.
      C
0025     22 IF(I2.EQ.1.OR.I2.EQ.2) GO TO 50
0027        I2=I2-2
0028        GO TO 50
      C
      C     START COLLECTION.
      C
0029     23 IF(I2.EQ.3.OR.I2.EQ.4) GO TO 50
0031        I2=I2+2
0032        GO TO 50
      C
      C     SEND PARAMETERS FROM VECTOR LF TO BACKGROUND JOB.
      C
0033     24 J1=IB(3)
      C
      C     IB(4) NOT ALLOWED TO BE EQUAL NS.
      C
0034        IF(IB(4).EQ.NS) IB(4)=IB(4)+1
0036        CALL ISDAT(LF(J1),IB(4))
0037        GO TO 50
      C
      C     SEND PARAMETERS FROM VECTOR IPF TO BACKGROUND JOB.
      C
0038     25 J1=IB(3)
      C
      C     IB(4) NOT ALLOWED TO BE EQUAL NS.
      C
0039        IF(IB(4).EQ.NS) IB(4)=IB(4)+1
0041        CALL ISDAT(IPF(J1),IB(4))
0042        GO TO 50
      C
      C     CHANGE PARAMETERS IN VECTOR LF.
      C
0043     26 J1=IB(3)
0044        J2=IB(4)
0045        I=4
0046        DO 27 J=J1,J2
0047        I=I+1
0048        IF(I.LT.13.OR.IB(I).GE.0) GO TO 27
0050        JJ=-IB(I)*12-7
0051        IB(I)=LF(JJ)-2
0052     27 LF(J)=IB(I)
0053        GO TO 50
      C
      C     CHANGE PARAMETERS IN VECTOR IPF.
      C
0054     29 J1=IB(3)
0055        J2=IB(4)
0056        I=4
0057        DO 30 J=J1,J2
0058        I=I+1
0059     30 IPF(J)=IB(I)
```

```
0060          GO TO 50
      C
      C       ACTIVATE NODE(S).
      C
0061       31 J1=IB(1)+1
0062          DO 32 I=3,J1
0063          J2=(IB(I)-1)*12+1
0064          LF(J2)=0
0065       32 LF(J2+1)=IABS(LF(J2+1))
0066          GO TO 50
      C
      C       DEACTIVATE NODE(S).
      C
0067       33 J1=IB(1)+1
0068          DO 34 I=3,J1
0069          J2=(IB(I)-1)*12+1
0070          LF(J2)=-1
0071       34 LF(J2+1)=0
0072          GO TO 50
      C
      C       AVAILABLE SPACE IN VECTOR IPF?
      C
0073       35 J1=IB(3)
0074          CALL RS1(IL,J1,LF,IV)
      C
      C       SEND THE RESULT OF THE SERCH TO THE BACKGROUND JOB.
      C
0075          IV(1)=IL
0076          CALL ISDAT(IV,1)
0077          GO TO 50
      C
      C       SEND INFORMATION ABOUT ACTIVE NODES TO THE BACKGROUND JOB.
      C
0078       36 J2=1
0079          DO 37 J1=2,240,12
0080          IF(LF(J1).EQ.0) GO TO 37
0082          J2=J2+1
0083          IV(J2)=LF(J1+1)
0084          J2=J2+1
0085          IV(J2)=J1/12+1
0086          J2=J2+1
0087          IV(J2)=LF(J1-1)
0088          J2=J2+1
0089          IV(J2)=LF(J1)
0090       37 CONTINUE
0091          IV(1)=J2
      C
      C       J2 NOT ALLOWED TO BE EQUAL NS.
      C
0092          IF(J2.EQ.NS) J2=J2+1
0094          CALL ISDAT(IV,J2)
0095          GO TO 50
      C
      C       RECEIVED INFORMATION ABOUT,NUMBER OF INPUT SIGNALS,
```

```
      C     OUTPUT SIGNALS AND MAIN SAMPLE PERIOD.
      C
0096     38 IT1(1)=IB(3)
0097        IT1(2)=IB(4)
0098        NI=IB(5)
0099        NU=IB(6)
0100        NS=(NI+NU+1)*2
      C
      C     CONVERT MAIN SAMPLE PERIOD FROM 2-WORD INTERNAL
      C     FORMAT TIME TO HOURS,MINUTES,SECONDS AND TICKS.
      C
0101        CALL CVTTIM(IT1,IH,IM,IS,IT)
      C
      C     CONVERT A 2-WORD INTERNAL TIME FORMAT TO INTEGER*4 FORMAT
      C     AND THEN CONVERT IT TO REAL*4 VALUE.
      C
0102        CALL JJCVT(IT1)
0103        TSM=AJFLT(IT1)*0.02
      C
      C
      C
0104     50 I1=I1-1
      C
      C     QUEUED A RECEIVE DATA REQUEST AND CONTINUE EXECUTION OF
      C     ISSUING JOB UNTIL A NEW MESSAGE IS RECEIVED,THEN ENTER
      C     THE SUBROUTINE FLAGR.
      C
0105        CALL IRCVDF(IB,59,LINK1,FLAGR)
      C
      C     DECREMENT SUSPENSION COUNTER TWICE.
      C
0106     60 CALL SUSPND
0107        CALL SUSPND
      C
      C     MESSAGE FROM BACKGROUND?
      C
0108        IF(I1.GE.0) GO TO 15
      C
      C     SCHEDULE THE SUBROUTINE FLAG TO BE RUN AFTER
      C     THE SAMPLE PERIOD HAS ELAPSED
      C
0110        CALL ITIMER(IH,IM,IS,IT,LINK2,ID,FLAG)
      C
      C     GET THE CURRENT TIME OF DAY
      C
0111        CALL GTIM(IT2)
0112        IPF(1)=IT2(1)
0113        IPF(2)=IT2(2)
      C
      C     READ INPUT MEASURMENT AND CONVERT TO PHYSICAL VALUE
      C
0114        CALL LOGIN(IPF,NI)
      C
0115        GO TO(350,90,330,90),I2
```

```
      C
      C     RUN ACTIVATED PROGRAMS.
      C
0116     90 DO 320 IP=1,240,12
0117        IF(LF(IP)) 320,100,310
0118    100 TS=TSM*FLOAT(LF(IP+1))
0119        K1=LF(IP+3)
0120        K2=LF(IP+4)
0121        K3=LF(IP+5)
0122        K4=LF(IP+6)
0123        K5=LF(IP+7)
0124        K6=LF(IP+8)
0125        K7=LF(IP+9)
0126        K8=LF(IP+10)
0127        K9=LF(IP+11)
0128        GO TO(101,102,103,104,105,106,107,108,109,110,
           *111,112,113,114,115,116,117,118,119,120) LF(IP+2)
      C
0129    101 CALL PIDR(IPF(K2-2),IPF(K6),IPF(K7),IPF(K1+8),IPF(K1+10),
           *IPF(K1+12),IPF(K1),IPF(K1+2),IPF(K1+4),IPF(K1+6),TS)
0130        GO TO 290
      C
0131    102 CONTINUE
0132        GO TO 290
      C
0133    103 CALL STEP(IPF(K1+2),IPF(K1),IPF(K2-2))
0134        GO TO 290
      C
0135    104 CALL ADDS(IPF(K6),IPF(K7),IPF(K8),IPF(K1),IPF(K1+2),
           *IPF(K1+4),IPF(K2-2))
0136        GO TO 290
      C
0137    105 CALL RAMP(IPF(K1+2),IPF(K1),TS,IPF(K2-2))
0138        GO TO 290
      C
0139    106 CALL SINU(IPF(K1+6),IPF(K1),IPF(K1+2),TS,IPF(K1+4),IPF(K2-2))
0140        GO TO 290
      C
0141    107 CALL PULS(IPF(K1+6),IPF(K1),IPF(K1+2),TS,IPF(K1+4),
           *IPF(K2-2))
0142        GO TO 290
      C
0143    108 CONTINUE
0144        GO TO 290
      C
0145    109 CONTINUE
0146        GO TO 290
      C
0147    110 CONTINUE
0148        GO TO 290
      C
0149    111 CALL PRB(IPF(K2),IPF(K3),IPF(K2-2),IPF(K5),IPF(K1))
0150        GO TO 290
      C
```

```
0151     112 CALL NODI(IPF(K1),IPF(K1+2),IPF(K2-2),IPF(K3),IPF(K3+1))
0152         GO TO 290
0153     113 CONTINUE
0154         GO TO 290
         C
0155     114 CONTINUE
0156         GO TO 290
         C
0157     115 CONTINUE
0158         GO TO 290
         C
0159     116 CONTINUE
0160         GO TO 290
         C
0161     117 CONTINUE
0162         GO TO 290
         C
0163     118 CONTINUE
0164         GO TO 290
         C
0165     119 CONTINUE
0166         GO TO 290
         C
0167     120 CONTINUE
         C
0168     290 IF(K9.NE.0) IPF(K9)=IPF(K2-2)
         C
0170     300 LF(IP)=LF(IP+1)
0171     310 LF(IP)=LF(IP)-1
0172     320 CONTINUE
         C
         C     SEND OUTPUT SIGNALS TO D/A-CONVERTER.
         C
0173         CALL LOGOUT(IPF,NU)
0174     330 GO TO(350,350,340,340),I2
         C
         C     QUEUED A SEND DATA REQUEST AND CONTINUE EXECUTION OF ISSURING
         C     JOB UNTIL THE BACKGROUND JOB HAS RECEIVE THE MESSAGE.
         C
0175     340 J1=(NI-8)*2
0176         J2=NU*2+18
0177         DO 339 I=19,J2
0178     339 IPF(I+J1)=IPF(I)
0179         IF(ISDATF(IPF,NS,LINK2,FLAG).EQ.0) GO TO 360
0181         I2=I2-2
         C
         C     INCREMENT SUSPENSION COUNTER.
         C
0182     350 CALL RESUME
         C
         C     GIVE BACKGROUND JOB SOME TIME.
         C
0183     360 CALL ISLEEP(0,0,0,2)
0184         GO TO 60
```

```
0185          STOP
0186          END
```

```
      C-------------------------------------------------------------------------------
      C
      C       NAME: FLAG
      C       ----
      C
      C       SUBTITLE: SETS A FLAG
      C       ---------
      C
      C       LANGUAGE: FORTRAN + RT-11-FORTRAN
      C       ---------
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 12-SEP-77
      C       ------------                                     ------
      C
      C       INSTITUTE:
      C       ----------
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-------------------------------------------------------------------------------
      C
      C        PURPOSE AND METHOD
      C        ==================
      C
      C        THE ROUTINE IS USED AS A "COMPLETITION ROUTINE" IN FORGROUND
      C        JOB IN SUCH AWAY THAT THE EVENTS AS FOLLOWS START THE ROUTINE:
      C
      C        1. SAMPLE TIME IS GONE
      C        2. BACKGROUND JOB HAS RECEIVED THE MEASUREMENT DATA
      C
      C        THE ROUTINE INCREMENTS SUSPENSION COUNTER ONCE.
      C
      C        COMMENTS: SUSPENSION COUNTER>=0: THE JOB CONTINOUS TO
      C        ---------                        EXECUTE
      C
      C        CHARACTERISTICS
      C        ===============
      C
      C        SUBROUTINE REQUIRED:
      C        --------------------
      C        FROM SYSTEM LIBRARY: RESUME
      C
      C        SIZE:
      C        -----
      C        PDP11/03: 10 WORDS
      C
      C-------------------------------------------------------------------------------
      C
0001          SUBROUTINE FLAG
      C
0002          CALL RESUME
      C
0003          RETURN
0004          END
```

```
C-------------------------------------------------------------------------------
C     NAME: FLAGR                                              NUMBER:
C     ------                                                   ---------
C
C     SUBTITLE:SETS A MESSAGE-FLAG
C     ------------
C
C     LANGUAGE: FORTRAN IV
C     ------------
C
C     KEYWORDS:FLAG
C     ------------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON       DATE:1977-07-05
C     ----------------                             ------
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                VERSION:
C     ---------                                                --------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE AND METHOD
C     ==================
C
C     THE ROUTINE IS USED AS A COMPLETITION ROUTINE IN FORGROUND
C     JOB IN SUCH AWAY THAT THE EVENT:
C
C     A MESSAGE IS RECEIVED FROM BACKGROUND JOB
C
C     STARTS THE ROUTINE.
C
C     THE ROUTINE SETS A FLAG ( I1 ) AND INCREMENTS SUSPENSION COUNTER TWI
C
C     USAGE:
C     =====
C     PROGRAM TYPE: SUBROUTINE
C     ------------------
C
C     COMMON AREA:
C     ------------
C
C     FL  -CONTAIN I1
C
C     CHARACTERISTICS:
C     ================
C
C     SUBROUTINE REQUIRED:
C     --------------------------
C
C
```

```
      C     FROM SYSTEM LIBRARY: RESUME
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 15 WORDS
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE FLAGR
      C
0002        COMMON/FL/I1
      C
0003        I1=I1+1
      C
      C
      C     INCREMENT SUSPENSION COUNTER TWICE.
      C
0004        CALL RESUME
0005        CALL RESUME
      C
0006        RETURN
0007        END
```

```
      C-------------------------------------------------------------------------------
      C
      C       NAME: FRENOD
      C       ------
      C
      C       SUBTITLE: LOOKS FOR A FREE NODE
      C       ----------
      C
      C       LANGUAGE: FORTRAN IV + RT-11-FORTRAN
      C       ----------
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 12-SEP-77
      C       -----------------                                ------
      C
      C       INSTITUTE:
      C       ----------
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-------------------------------------------------------------------------------
      C
      C       PURPOSE AND METHOD
      C       ==================
      C
      C       LOOKS THROUGH THE LIST OF NODES, TO FIND A FREE ONE.
      C       IF NONE IS FOUND, AN ERROR MESSAGE IS GIVEN.
      C
      C       USAGE
      C       =====
      C
      C       ARGUMENTS:
      C       ----------
      C
      C       FRENOD(LIST,IL)
      C
      C       LIST     -ARRAY CONTAINING NODE HEADS, (I)
      C       IL       -POINTER TO THE FIRST ELEMENT IN A FREE NODE.
      C                 IF IL < 0 , THEN THERE ARE NO AVAILABLE NODE.
      C
      C       CHARACTERISTICS
      C       ===============
      C
      C       SIZE:
      C       -----
      C        PDP11/03: 51 WORDS
      C
      C-------------------------------------------------------------------------------
      C
0001          SUBROUTINE FRENOD(LIST,IL)
      C
0002          DIMENSION LIST(1)
      C
0003          DO 10  I=1,120,12
0004          IF(LIST(I).LT.0) GO TO 20
```

```
0006      10 CONTINUE
      C
0007         WRITE(7,500)
0008     500 FORMAT(1X,'NO AVAILABLE NODE')
0009         IL=-1
      C
0010         RETURN
0011      20 IL=I
      C
0012         RETURN
0013         END
```

```
C-------------------------------------------------------------------------------
C     NAME:GRAF                                                  NUMBER:
C     ----                                                       ------
C
C     SUBTITLE: GENERATES A CHARACTER PLOTT.
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C     KEYWORDS: PLOTT
C     ---------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                     DATE:1977-07-0
C     ------------                                               ----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                  VERSION:
C     --------                                                   -------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE AND METHOD
C     ==================
C
C     THE ROUINE GENERATES A CHARACTER PLOTT AND SEND IT TO
C     THE LINE PRINTER OR TO THE CONSOLE TERMINAL.
C
C     USAGE:
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ---------
C
C     GRAF( X,N,XMAX,XMIN,NC,LU,IRAD )
C
C          X        -VECTOR CONTAINING THE INPUT VALUES (I)
C          N        -NUMBER OF INPUT VALUES (I)
C          XMAX     -HIGHER LIMIT (I)
C          XMIN     -LOWER LIMIT (I)
C          NC       -NUMBER OF SAMPLE (I/O)
C          LU       -LOGICAL UNIT (I)
C          IRAD     -WORK ARRAY, SIZE(101)
C
C     NOTE: VALUES OUT OF RANGE ARE ONLY MAKED BY SPACE CHARACTERS.
C     ----
C
```

```
      C     CHARACTERISTICS:
      C     ================
      C
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 328 WORDS
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE GRAF(X,N,XMAX,XMIN,NC,LU,IRAD)
0002        DIMENSION X(1),IRAD(1)
0003        DATA IB,IP,IA,II/1H ,1H+,1H*,1HI/
      C
0004        NR=50
0005        NT=101
0006        IF(LU.EQ.6) GO TO 5
0008        NR=20
0009        NT=51
0010        IF(N.GT.1) NC=0
      C
      C
      C
0012      5 SF=(XMAX-XMIN)/FLOAT(NT-1)
0013        DO 50 I=1,N
      C
0014        NC=NC+1
      C
      C     NEW PAGE?
      C
0015        IF(MOD(NC-1,NR).NE.0) GO TO 20
      C
0017        WRITE(LU,9) XMIN,XMAX
0018      9 FORMAT(1H1,10X,'LOWER LIMIT=',E15.5,4X,'HIGHER LIMIT=',E15.5/)
0019        IF(LU.EQ.7) GO TO 15
0021        WRITE(LU,10)(J,J=10,100,10)
0022     10 FORMAT(5X,2H0%,10(5X,I4,1H%)/5X,1H+,10(9(1H-),1H+))
0023        GO TO 20
0024     15 WRITE(LU,16)(J,J=20,100,20)
0025     16 FORMAT(5X,2H0%,5(5X,I4,1H%)/5X,1H+,5(9(1H-),1H+))
      C
0026     20 DO 30 J=1,NT
0027     30 IRAD(J)=IB
      C
0028        IR=(X(I)-XMIN)/SF+1.5
0029        IRAD(1)=II
0030        IF(MOD(NC,5).EQ.0) IRAD(1)=IP
      C
      C     OUT OF RANGE?
      C
0032        IF(IR.GT.NT.OR.IR.LT.1) GO TO 35
```

```
      C
0034          IRAD(IR)=IA
0035       35 IF(MOD(NC,5).EQ.0) GO TO 40
0037          WRITE(LU,60)(IRAD(J),J=1,NT)
0038          GO TO 50
0039       40 WRITE(LU,70) NC,(IRAD(J),J=1,NT)
0040       50 CONTINUE
      C
      C
0041       60 FORMAT(5X,101A1)
0042       70 FORMAT(1X,I4,101A1)
0043          RETURN
0044          END
```

```
C-------------------------------------------------------------------------------
C     NAME:GRAFST                                                    NUMBER:
C     -----                                                          --------
C
C     SUBTITLE: START-UP ROUTINE TO THE ROUTINE GRAF
C     ----------
C
C     LANGUAGE: FORTRAN IV
C     ----------
C
C     KEYWORDS: START-UP
C     ----------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                         DATE:1977-07-0
C     -------------                                                  -----
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                      VERSION:
C     ---------                                                      --------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE AND METHOD
C     ==================
C
C     THE ROUTINE:
C
C     1. ALLOCATES SPACE IN THE PARAMETER AREA
C     2. SETS THE ADDRESS-PART IN THE NODE HEAD, ADDRESSES
C        TO THE PARAMETERAREA
C     3. ASKS FOR AND SETS SOME PARAMETER VALUES
C
C     USAGE:
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     --------------
C
C     ARGUMENTS:
C     ----------
C
C     GRAFST(LIST,IL,IPA,IW)
C
C     IL  -POINTER TO THE FIRST WORD IN THE ACTUAL NODE, (I)
C     IPA -VECTOR CONTAINING THE PARAMETERS IN BACKGROUND JOB, (I)
C     IW  -WORKAREA
C
C     CHARACTERISTICS:
C     ================
C
```

```
      C     SUBROUTINES REQUIRED:  CV,LU,RS
      C     ------------------------------
      C
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 215 WORDS
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE GRAFST(LIST,IL,IPA,IW)
0002        DIMENSION LIST(1),IPA(1),IW(1)
      C
0003        WRITE(7,500)
0004    500 FORMAT(1X,$,'NUMBER OF ELEMENT:')
0005        READ(5,10) NE
0006     10 FORMAT(I10)
      C
      C     ALLOCATE SPACE IN VECTOR IPA FOR 7 WORDS.
      C
0007        CALL RS(IL,7,LIST,IW)
      C
      C     ENOUGH FREE SPACE?
      C
0008        IF(IL.LT.0) RETURN
      C
      C     SET ADDRESSES.
      C
0010        K1=LIST(IL+3)
0011        LIST(IL+4)=K1+1
0012        LIST(IL+5)=K1+3
0013        LIST(IL+6)=K1+5
0014        K5=LIST(IL+7)
      C
0015        DO 15 I=K1,K5
0016     15 IPA(I)=0
      C
0017        IPA(K1)=NE
      C
0018        WRITE(7,501)
0019    501 FORMAT(1X,$,'HIGHER LIMIT:')
0020        READ(5,20) HL
0021     20 FORMAT(F15.0)
0022        CALL CV(IPA(K1+1),HL)
      C
0023        WRITE(7,502)
0024    502 FORMAT(1X,$,'LOWER LIMIT:')
0025        READ(5,20) XL
0026        CALL CV(IPA(K1+3),XL)
      C
      C
```

```
      C     WHICH LOGICAL UNIT?
      C
0027        CALL LU(IPA(K5))
      C
0028        RETURN
0029        END
```

```
C-------------------------------------------------------------------------------
C
C     NAME: GROU                                                NUMBER:
C     ----                                                      ------
C
C     SUBTITLE:  GROUPING OF DATA
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C     KEYWORDS: GROUPING
C     ---------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON       DATE:1977-07-05
C     ------------                                       ------
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                             VERSION:
C     --------                                              -------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE
C     =======
C
C     GROUPING OF DATA
C
C     USAGE:
C     =====
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ----------
C
C     GROU( Y,YL,CL,NC,ID )
C
C     Y    -INPUT VALUE, (I)
C     YL   -LOWER LIMIT, (I)
C     CL   -CLASS LENGHT, (I)
C     NC   -NUMBER OF CLASSES, (I)
C     ID   -VECTOR CONTAINING THE FREQUENCIES DISTRIBUTION, SIXE(NC+2), (
C
C     NOTE: THE LOWEST AND HIGHEST CLASSES IN ID CONTAIN MEASUREMENTS
C     ----  OUT OF RANGE, SO ID CONTAINS NC+2 ELEMENTS
C
C     CHARACTERISTICS:
C     ===============
C
```

```
      C     SIZE:
      C     -----
      C     PDP11/03: 58 WORDS
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE GROU(Y,YL,CL,NC,ID)
      C
0002        DIMENSION ID(1)
      C
0003        I=(Y-YL)/CL+2.0
0004        IF(I.LT.1) I=1
0006        IF(I.GT.NC+1) I=NC+2
0008        ID(I)=ID(I)+1
      C
0009        RETURN
0010        END
```

```
C-----------------------------------------------------------------------
C       NAME:GROUST                                          NUMBER:
C       ----                                                 -------
C
C       SUBTITLE: START-UP ROUTINE TO THE ROUTINE GROU
C       --------
C
C       LANGUAGE: FORTRAN IV
C       --------
C
C       KEYWORDS: START-UP
C       --------
C
C       IMPLEMENTOR: LEIF RADING,TYKE PAULSSON               DATE:1977-07-0
C       -----------                                          ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C       ACCEPTED:                                            VERSION:
C       --------                                             -------
C
C-----------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       THE ROUTINE:
C
C       1. ALLOCATES SPACE IN THE PARAMETER AREA
C       2. SETS THE ADDRESS-PART IN THE NODE HEAD, ADDRESSES
C          TO THE PARAMETER AREA
C       3. ASKS FOR AND SETS SOME PARAMETER VALUES
C
C       USAGE:
C       =====
C
C       PROGRAM TYPE: SUBROUTINE
C       ------------
C
C       ARGUMENTS:
C       ---------
C
C       GROUST(LIST,IL,IPA,IW)
C
C       IL  -POINTER TO THE FIRST WORD IN THE ACTUAL NODE, (I)
C       IPA -VECTOR CONTAINING THE PARAMETERS IN BACKGROUND JOB, (I)
C       IW  -WORKAREA
C
C       CHARACTERISTICS:
C       ===============
C
```

```
      C     SUBROUTINES REQUIRED:  CV,RS
      C     ---------------------------
      C
      C     SIZE:
      C     -----
      C     PDP11/03:212 WORDS
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE GROUST(LIST,IL,IPA,IW)
      C
0002        DIMENSION LIST(1),IPA(1),IW(1)
      C
0003        WRITE(7,500)
0004    500 FORMAT(1X,$,'NUMBER OF CLASSES:')
0005        READ(5,10) NC
0006     10 FORMAT(I10)
      C
      C     CALCULATE NUMBER OF WORDS TO BE ALLOCATED FOR THE ROUTINE.
      C
0007        NW=NC+7
      C
      C     ALLOCATE SPACE IN VECTOR IPA FOR NW WORDS.
      C
0008        CALL RS(IL,NW,LIST,IW)
      C
      C     ENOUGH FREE SPACE.
      C
0009        IF(IL.LT.0) RETURN
      C
      C     CALCULATE ADDRESSES AND STORE THEM IN THE NODE.
      C
0011        K1=LIST(IL+3)
0012        K2=K1+NC+2
0013        LIST(IL+4)=K2
0014        K3=K2+2
0015        LIST(IL+5)=K3
0016        K5=LIST(IL+7)
      C
0017        DO 15 I=K1,K5
0018     15 IPA(I)=0
      C
0019        IPA(K5)=NC
      C
0020        WRITE(7,501)
0021    501 FORMAT(1X,$,'LOWER LIMIT:')
0022        READ(5,20) YL
0023        CALL CV(IPA(K2),YL)
0024     20 FORMAT(F15.0)
      C
0025        WRITE(7,502)
```

```
0026     502 FORMAT( 1X, $, ' CLASS LENGTH: ' )
0027         READ( 5, 20 ) CL
0028         CALL CV( IPA( K3 ), CL )
      C
0029         RETURN
0030         END
```

```
C-----------------------------------------------------------------------
C     NAME: IWRI                                                NUMBER:
C     ----                                                      -------
C
C     SUBTITLE: WRITE INTEGER VALUES
C     ----------
C
C     LANGUAGE: FORTRAN IV
C     ----------
C
C     KEYWORDS: WRITE
C     ----------
C
C     IMPLEMENTOR: LEIF RADING, TYKE PAULSSON                   DATE: 1977-07-0
C     ----------                                                      ------
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C     ACCEPTED:                                 VERSION:
C     ---------                                 --------
C
C-----------------------------------------------------------------------
C
C     PURPOSE
C     =======
C
C     WRITES INTEGER VALUES, MAXIMUM 4, ON LINE PRINTER
C     OR CONSOLE
C
C     USAGE
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     --------------
C
C     ARGUMENTS:
C     ----------
C
C     IWRI( X1, X2, X3, X4, IN, CM1, LUN, IT, W1 )
C
C     X1   -INPUT VALUE 1, (I)
C     X2   -INPUT VALUE 2, (I)
C     X3   -INPUT VALUE 3,(I)
C     X4   -INPUT VALUE 4, (I)
C     IN   -ARRAY CONTAINING INFORMATION ABOUT NUMBER OF SIGNALS
C           AND NAMES, (I)
C     CM1 -ARRAY CONTAINING PARAMETER NAMES, (I)
C     LUN -LOGICAL UNIT, (I)
C     IT   -INTERNAL TIME, (I)
C     W1   -WORK AREA
C
```

```
      C
      C     CHARACTERISTICS
      C     ===============
      C
      C     SUBROUTINES REQUIRED:
      C     ---------------------
      C
      C     FROM SYSTEM LIBRARY: TIMASC
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 139
      C
      C
      C-----------------------------------------------------------------------
      C     PROGRAM
      C     =======
      C
0001        SUBROUTINE IWRI(X1,X2,X3,X4,IN,CM1,LUN,IT,W1)
0002        INTEGER X1,X2,X3,X4,W1
0003        DIMENSION IN(1),CM1(1),W1(1),IT(1)
      C
      C     CONVERT THE 2-WORD INTERNAL FORMAT TIME INTO AN
      C     ASCII STRING.
      C
0004        CALL TIMASC(IT,W1)
0005        WRITE(LUN,20)(W1(I),I=1,4)
0006     20 FORMAT(1H0,'TIME:',4A2/)
      C
0007        W1(1)=X1
0008        W1(2)=X2
0009        W1(3)=X3
0010        W1(4)=X4
0011        I1=IN(1)*3+1
0012        J=0
0013        DO 5 I=2,I1,3
0014        J=J+1
0015        I2=IN(I)
0016      5 WRITE(LUN,10) CM1(I2),IN(I+1),IN(I+2),W1(J)
0017     10 FORMAT(1X,A4,I1,'/',I1,':',I15)
      C
0018        RETURN
0019        END
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 17:39:51                PAGE 001

      C-------------------------------------------------------------------------------
      C
      C       NAME: LOGIN
      C       ~~~~
      C
      C       SUBTITLE: READS AND CONVERTS INPUTS
      C       ~~~~~~~~~~~~
      C
      C       LANGUAGE: FORTRAN + RT-11-FORTRAN
      C       ~~~~~~~~~~~~
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 12-SEP-77
      C       ~~~~~~~~~~~~~~~~                                 ~~~~~~
      C
      C       INSTITUTE:
      C       ~~~~~~~~~~
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-------------------------------------------------------------------------------
      C
      C       PURPOSE AND METHOD
      C       ==================
      C
      C       THE SUBROUTINE ADIN READS THE MEASUREMENT-INSIGNALS FROM AN AD-
      C       CONVERTER AND THEN THE SIGNALS ARE CONVERTED BY ZEROLEVEL
      C       AND SLOPE FROM THE ENGENEER-NODE.
      C
      C       USAGE
      C       =====
      C
      C       ARGUMENTS:
      C       ~~~~~~~~~~
      C
      C       LOGIN(XIV,NI)
      C
      C       XIV      -SIGNALS AND PARAMETER ARRAY,(I/O)
      C       NI       -NUMBER OF INPUTSIGNALS,(I)
      C
      C       CHARACTERISTICS
      C       ===============
      C
      C       SUBROUTINE REQUIED: ADIN
      C       ~~~~~~~~~~~~~~~~~~~~~~~~
      C
      C       SIZE:
      C       ~~~~
      C       PDP11/03:  56 WORDS
      C
      C-------------------------------------------------------------------------------
      C
0001          SUBROUTINE LOGIN(XIV,NI)
      C
0002          DIMENSION XIV(1)
```

```
      C
0003          J=12
      C
0004          DO 10 I=1,NI
0005          J=J+2
0006       10 XIV( I+1 )=ADIN( I-1 )*XIV( J+1 )+XIV( J )
      C
0007          RETURN
0008          END
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 17:40:28                PAGE 001

      C-------------------------------------------------------------------------------
      C
      C       NAME: LOGOUT
      C       ------
      C
      C       SUBTITLE: CONVERTS OUTPUTS
      C       ----------
      C
      C       LANGUAGE: FORTRAN + RT-11-FORTRAN
      C       ----------
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 12-SEP-77
      C       ---------------                                  -----
      C
      C       INSTITUTE:
      C       ----------
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-------------------------------------------------------------------------------
      C
      C       PURPOSE AND METHOD
      C       ==================
      C
      C       THE OUTSIGNALS ARE CONVERTED BY ZEROLEVEL AND SLOPE FROM THE
      C       ENGINEER-NODE AND ARE SENT TO DA-CONVERTER BY THE SUBROUTINE
      C       DAOUT.
      C
      C       USAGE
      C       =====
      C
      C       LOGOUT( XIV, NO )
      C
      C       XIV      -SIGNALS AND PARAMETER ARRAY,( I/O )
      C       NO       -NUMBER OF OUTPUTSIGNALS,( I )
      C
      C       CHARACTERISTICS
      C       ===============
      C
      C       SIZE:
      C       -----
      C       PDP11/03  60 WORDS
      C
      C       SUBROUTINE REQUIRED: DAOUT
      C       ----------------------
      C
      C-------------------------------------------------------------------------------
      C
0001          SUBROUTINE LOGOUT( XIV, NO )
0002          DIMENSION XIV( 1 )
0003          J=28
0004          DO 10 I=1, NO
0005          J=J+2
0006          X=XIV( I+9 )*XIV( J+1 )+XIV( J )
```

```
0007      10 CALL DAOUT( I-1,X )
0008         RETURN
0009         END
```

```
      C-------------------------------------------------------------------------------
      C
      C       NAME: LSTB
      C       ----
      C
      C       SUBTITLE: LISTS ACTIVE NODES IN BACKGROUND JOB
      C       ---------
      C
      C       LANGUAGE: FORTRAN + RT-11-FORTRAN
      C       ---------
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 15-SEP-77
      C       ------------                                     ----
      C
      C       INSTITUTE:
      C       ----------
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-------------------------------------------------------------------------------
      C
      C        PURPOSE AND METHOD
      C        ==================
      C
      C        LOOKS THROUGH THE NODE LIST AND LISTS THE ACTIVE NODES
      C
      C        USAGE
      C        =====
      C
      C        ARGUMENTS:
      C        ----------
      C
      C        LSTB(LIST,CM2,LUN)
      C
      C        LIST     -NODE LIST IN BACKGROUND JOB, SIZE(120), (I)
      C        CM2      -ARRAY CONTAINING PROGRAM NAMES, SIZE(20), (I)
      C        LUN      -LOGICAL UNIT, (I)
      C
      C        CHARACTERISTICS
      C        ===============
      C
      C        SIZE:
      C        -----
      C        PDP11/03    109 WORDS
      C
      C-------------------------------------------------------------------------------
      C
0001          SUBROUTINE LSTB(LIST,CM2,LUN)
      C
0002          DIMENSION LIST(1),CM2(1)
      C
0003          WRITE(LUN,10)
0004       10 FORMAT(1H0,'NAME:',4X,'COUNTER',4X,'SAMPLE PERIOD')
      C
```

```
      C     LOOKS THROUGH THE NODELIST
      C
0005        DO 30 J=1,120,12
0006        IF(LIST(J).LT.0) GO TO 30
      C
      C     THE NODE IS ACTIVE
      C
0008        J1=LIST(J+2)/1000
0009        J2=LIST(J+2)-J1*1000
      C
      C     WRITES THE PROGRAM NAMES ON GIVEN DEVICE
      C
0010        WRITE(LUN,20) CM2(J1),J2,LIST(J),LIST(J+1)
0011     20 FORMAT(1X,A4,I1,I8,5X,I8)
0012     30 CONTINUE
      C
0013        RETURN
0014        END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: LSTF
C       ----
C
C       SUBTITLE: LISTS OCCUPIED NODES IN FOREGROUND JOB
C       ---------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ---------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 15-SEP-77
C       ------------                                      ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       SENDS FOR THE OCCUPIED NODES IN FOREGROUND JOB.
C       LISTS THEM ON GIVEN DEVICE
C
C       USAGE
C       =====
C
C       ARGUMENTS:
C       ----------
C
C       LSTF(IV,IFL2,CM3,LUN)
C
C       IV      -ARRAY FOR SENDING/TRANSFERING OF INFORMATION BETWEEN
C                THE JOBS, SIZE(100)
C       IFL2    -FLAG, (I/O)
C       CM3     -ARRAY CONTAINING PROGRAM NAMES, SIZE(20), (I)
C       LUN     -LOGICAL UNIT, (I)
C
C       CHARACTERISTICS
C       ===============
C
C       SIZE:
C       ----
C       PDP11/03   125 WORDS
C
C       SUBROITUNES REQUIRED:
C       --------------------
C
C       FROM SYSTEM LIBRAY: ISDAT
C-------------------------------------------------------------------------------
C
```

```
      C
0001          SUBROUTINE LSTF(IV,IFL2,CM3,LUN)
      C
0002          DIMENSION IV(1),CM3(1)
      C
0003          IV(1)=12
      C
      C       SENDS FOR OCCUPIED NODES
      C
0004          IFL2=0
0005          CALL ISDAT(IV,1)
0006       10 IF(IFL2) 20,10,20
      C
0007       20 WRITE(LUN,30)
0008       30 FORMAT(1H0,'NAME:',4X,'COUNTER',4X,'SAMPLE PERIOD')
0009          J1=IV(4)-1
0010          IF(J1.LE.0) RETURN
      C
      C       LISTS OCCUPIED NODES
      C
0012          DO 50 J=1,J1,4
0013          J2=IV(J+4)
0014          WRITE(LUN,40) CM3(J2),IV(J+5),IV(J+6),IV(J+7)
0015       40 FORMAT(1X,A4,I2,I8,4X,I8)
0016       50 CONTINUE
      C
0017          RETURN
0018          END
```

```
      C-----------------------------------------------------------------------
      C
      C       NAME: LU
      C       ----
      C
      C       SUBTITLE: SETS LOGICAL OUTPUT
      C       ---------
      C
      C       LANGUAGE: FORTRAN + RT-11-FORTRAN
      C       ---------
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 12-SEP-77
      C       ---------------                                  ------
      C
      C       INSTITUTE:
      C       ----------
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-----------------------------------------------------------------------
      C
      C       PURPOSE AND METHOD
      C       ==================
      C
      C       ASKS ABOUT LOGICAL OUTPUT, READS AND IDENTIFIES THE
      C       ANSWER.
      C
      C       USAGE
      C       =====
      C
      C       ARGUMENTS:
      C       ----------
      C
      C       LU(LUN)
      C
      C       LUN     -LOGICAL OUTPUT,(0)
      C
      C       CHARACTERISTICS
      C       ===============
      C
      C       SIZE:
      C       ----
      C       PDP11/03:  76 WORDS
      C
      C-----------------------------------------------------------------------
      C
0001          SUBROUTINE LU(LUN)
      C
0002          DIMENSION IDE(2)
      C
0003          DATA IDE/2HLP,2HTT/
      C
0004       10 WRITE(7,20)
0005       20 FORMAT(1X,$,'OUTPUT ON DEVICE:')
```

```
0006          READ(5,30) J
0007       30 FORMAT(A2)
      C
0008          DO 40 I=1,2
0009          IF(IDE(I).EQ.J) GO TO 60
0011       40 CONTINUE
      C
0012          WRITE(7,50)
0013       50 FORMAT(1X,'?ILL DEV?')
0014          GO TO 10
      C
0015       60 LUN=I+5
      C
0016          RETURN
0017          END
```

```
C-------------------------------------------------------------------------------
C     NAME: MAXM                                                    NUMBER:
C     ----                                                          ------
C
C     SUBTITLE: CALCULATIONS OF MAXIMUM AND MINIMUM VALUES
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C     KEYWORDS: MAXIMUM,MINIMUM
C     ---------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                DATE:1977-07-05
C     ------------                                          ----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                     VERSION:
C     --------                                                      -------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE
C     =======
C
C     THE ROUTINE REGISTRATES THE MAXIMUM AND MINIMUM VALUES
C     OF A PARAMETER AND THEIR SAMPLINGS NUMBERS.
C
C     USAGE
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ---------
C
C     MAXM( X,XMAX,XMIN,MAXS,MINS,NSAMP )
C
C     X             -INPUT VALUE, (I/O)
C     XMAX          -MAXIMUM VALUE, (O)
C     XMIN          -MINIMUM VALUE, (I/O)
C     MAXS          -SAMPLE NUMBER FOR MAXIMUM VALUE, (O)
C     MINS          -SAMPLE NUMBER FOR MINIMUM VALUE, (O)
C     NSAMP          -ACTUAL SAMPLE NUMBER, (I/O)
C
C
C     CHARACTERISTICS
C     ===============
C
```

```
      C     SIZE:
      C     -----
      C     PDP11/03:  47 WORDS
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM
      C     =======
      C
0001        SUBROUTINE MAXM(X,XMAX,XMIN,MAXS,MINS,NSAMP)
      C
0002        NSAMP=NSAMP+1
0003        IF(X.GE.XMIN.AND.X.LE.XMAX) RETURN
0005        IF(X.LT.XMIN) GO TO 10
0007        MAXS=NSAMP
0008        XMAX=X
      C
0009        RETURN
      C
0010     10 MINS=NSAMP
0011        XMIN=X
      C
0012        RETURN
0013        END
```

```
C-------------------------------------------------------------------------------
C     NAME:MAXMST                                                    NUMBER:
C     ----                                                           ------
C
C     SUBTITLE: START-UP ROUTINE TO THE ROUTINE MAXM
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     --------
C
C     KEYWORDS: START-UP
C     --------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                         DATE:1977-07-0
C     -----------                                                    ----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                      VERSION:
C     --------                                                       -------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE AND METHOD
C     ==================
C
C     THE ROUTINE:
C
C     1. ALLOCATES SPACE IN THE PARAMETER AREA
C     2. SETS THE ADDRESS-PART IN THE NODE HEAD, ADDRESSES
C        TO THE PARAMETERAREA
C     3. ASKS FOR AND SETS SOME PARAMETER VALUES
C
C     USAGE:
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ---------
C
C     MAXMST(LIST,IL,IPA,IW)
C
C     IL  -POINTER TO THE FIRST WORD IN THE ACTUAL NODE, (I)
C     IPA -VECTOR CONTAINING THE PARAMETERS IN BACKGROUND JOB, (I)
C     IW  -WORKAREA
C
C     CHARACTERISTICS:
C     ===============
C
```

```
      C     SUBROUTINES REQUIRED:  CV,RS
      C     ---------------------------
      C
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 125 WORDS
      C
      C
      C-----------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE MAXMST(LIST,IL,IPA,IW)
0002        DIMENSION LIST(1),IPA(1),IW(1)
      C
      C     ALLOCATE SPACE IN VECTOR IPA FOR 7 WORDS.
      C
0003        CALL RS(IL,7,LIST,IW)
      C
      C     ENOUGH FREE SPACE?
      C
0004        IF(IL.LT.0) RETURN
      C
      C     CALCULATE ADDRESSES AND STORE THEM IN THE NODE.
      C
0006        K1=LIST(IL+3)
0007        LIST(IL+4)=K1+1
0008        LIST(IL+5)=K1+3
0009        LIST(IL+6)=K1+4
0010        K5=LIST(IL+7)
      C
      C
0011        DO 10 I=K1,K5
0012     10 IPA(I)=0
      C
0013        CALL CV(IPA(K1+1),-1.0E32)
0014        CALL CV(IPA(K1+4),+1.0E32)
      C
0015        RETURN
0016        END
```

```
C-------------------------------------------------------------------------------
C     NAME:NODI                                                    NUMBER:
C     ----                                                         ------
C
C     SUBTITLE: GENERATE A PSEUDO-RANDOM NUMBER
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C     KEYWORDS: RANDOM,NORMAL
C     ---------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                       DATE:1977-07-0
C     ------------                                                 -----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                    VERSION:
C     --------                                                     -------
C
C-------------------------------------------------------------------------------
C
C     USAGE:
C     =====
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ---------
C
C     NODI(XM,SG,Y,I1,I2)
C
C     XM   -MEAN VALUE (I)
C     SG   -DEVIATION, (I)
C     Y    -OUTPUT FROM RANDOM NUMBER GENERATOR, (O)
C     I1   -INTEGER, (I/O)
C     I2   -INTEGER, (I/O)
C
C     NOTES:
C     -----
C
C     I1,I2 MAY NOT BE CHANGED.
C
C     METHOD:
C     ======
C
C     THE FUNCTION RAN GENERATES 12 RANDOM RECTANGULAR
C     DISTRIBUATED NUMBERS (0,1). THE SUM MINUS 6.0
C     IS TAKEN AS THE NORMAL (0,1) RANDOM NUMBER.
C     THE NORMAL (0,1) RANDOM NUMBER ARE THEN TRANSFORMED TO
```

```
C     NORMAL (XM,SG) RANDOM NUMBER.
C
C     REFERENCES
C     ----------
C
C    1)   B.JANSSON RANDOM NUMBER GENERATORS.
C
C
C     CHARACTERISTICS:
C     ================
C
C     FUNCTION REQIRED: RAN (FORTAN LIBRARY PDP11/03)
C     -----------------------
C
C     SIZE:
C     -----
C     PDP11/03:  41 WORDS
C
C
C-------------------------------------------------------------------------------
C     PROGRAM:
C     ========
C
0001        SUBROUTINE NODI(XM,SG,Y,I1,I2)
0002        Y=-6.0
0003        DO 10 I=1,12
0004     10 Y=Y+RAN(I1,I2)
0005        Y=Y*SG+XM
0006        RETURN
0007        END
```

```
C-------------------------------------------------------------------------------
C     NAME: NORM                                                  NUMBER:
C     ----                                                        ------
C
C     SUBTITLE:NORMALIZATION OF FREQUENCY FUNCTION
C     ------------
C
C     LANGUAGE: FORTRAN IV
C     ----------
C
C     KEYWORDS: NORMALIZATION
C     ----------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                      DATE:1977-07-0
C     ----------------                                            ------
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                   VERSION:
C     ---------                                                   --------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE
C     =======
C     NORMALIZATION OF FREQUENCY FUNCTION FOR
C     COMPARING WITH THE NORMAL DISTRIBUTION
C
C     USAGE:
C     =====
C     PROGRAM TYPE: SUBROUTINE
C     ------------------
C
C     ARGUMENTS:
C     ----------
C
C     NORM( IX,CL,NC,Y )
C
C     IX   -VECTOR CONTAINING THE CUMMULATIVE FREQUECX FUNCTION, SIZE(NC)
C     CL   -CLASS LENGHT, (I)
C     NC   -NUMBER OF CLASSES, (I)
C     Y    -VECTOR CONTAINIG THE CALCULATED PROBABILLTY FUNCTION, SIZE(NC
C
C
C     METHOD:
C     ======
C     Y(I)=IX(I)/(CL*S)
C     S=NUMBER OF DATA
C
C     CHARACTERISTICS:
C     ===============
```

```
      C
      C     SUBROUTINES REQUIRED: NONE
      C     ----------------------------
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 70 WORDS
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE NORM(IX,CL,NC,Y)
      C
0002        DIMENSION IX(1),Y(1)
0003        IS=0
      C
0004        NA=NC+2
0005        DO 10 I=1,NA
0006     10 IS=IS+IX(I)
      C
0007        S=IS
      C
0008        DO 20 I=1,NA
0009     20 Y(I)=FLOAT(IX(I))/(S*CL)
      C
0010        RETURN
0011        END
```

```
C-------------------------------------------------------------------------------
C     NAME:NORMST                                                NUMBER:
C     -----                                                      --------
C
C     SUBTITLE: START-UP ROUTINE TO THE ROUTINE NORM
C     ----------
C
C     LANGUAGE: FORTRAN IV
C     ----------
C
C     KEYWORDS: START-UP
C     ----------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                     DATE:1977-07-0
C     -------------                                              -----
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                  VERSION:
C     ---------                                                  ----------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE AND METHOD
C     ==================
C
C     THE ROUTINE:
C
C     1. ALLOCATES SPACE IN THE PARAMETER AREA
C     2. SETS THE ADDRESS-PART IN THE NODE HEAD, ADDRESSES
C        TO THE PARAMETER AREA
C     3. ASKS FOR AND SETS SOME PARAMETER VALUES
C
C     USAGE:
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     ------------------
C
C     ARGUMENTS:
C     ----------
C
C     NORMST(LIST,IL,IPA,IW)
C
C     IL  -POINTER TO THE FIRST WORD IN THE ACTUAL NODE, (I)
C     IPA -VECTOR CONTAINING THE PARAMETERS IN BACKGROUND JOB, (I)
C     IW  -WORK AREA
C
C     CHARACTERISTICS:
C     ================
C
```

```
      C     SUBROUTINES REQUIRED:  RS
      C     ------------------------
      C
      C     SIZE:
      C     -----
      C     PDP11/03:  48 WORDS
      C
      C
      C-----------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE NORMST(LIST,IL,IPA,IW)
      C
0002        DIMENSION LIST(1),IPA(1),IW(1)
      C
      C     FETCH ADDRESS TO NUMBER OF CLASSES
      C
0003        K8=LIST(IL+10)
0004        NC=IPA(K8)+4
0005        NW=NC*2
      C
      C     ALLOCATE SPACE IN VECTOR IPA FOR NW WORDS.
      C
0006        CALL RS(IL,NW,LIST,IW)
      C
0007        RETURN
0008        END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: OPCOM
C       ----
C
C       SUBTITLE: OPERATOR COMMUNICATION ROUTINE
C       ---------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ---------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 7-SEP-77
C       ---------------                                       ----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       OPCOM TAKES CARE OF THE OPERATOR COMMUNICATION.
C       WHEN THE PROGRAM PRINTS ">" IT IS READY TO RECEIVE A COMMAND.
C       AFTER READING AND IDENTIFYING THE COMMAND THE PROGRAM EITHER
C       CALLS A SUBROUTINE THAT TAKES CARE OF THE COMMAND OR EXECUTES
C       THE COMMAND ITSELF.
C
C       USAGE
C       =====
C
C       COMMON BLOCKS:
C       --------------
C
C       BG      -CONTAINING LIST- AND PARAMETER ARRAYS FOR BACKGROUND
C                JOB
C       FLA     -CONTAINING FLAGS AND POINTERS WITH INFORMATION
C                ABOUT COLLECTING AND CALCULATING BLOCKS.
C       CMD     -CONTAINING 4-CHARACTERS COMMAND NAME AND 2-CHARACTERS
C                PARAMETER NAME.
C       WRK     -WORK AREA
C
C       CHARACTERISRICS
C       ===============
C
C       SIZE:
C       -----
C       PDP11/03: 770 WORDS
C
C
C       SUBROUTINE REQUIRED:  CRNF,FRENOD,GRAFST,GROUST,LSTB,LSTF,MAXMST
C       --------------------  NORMST,OPEN,STAFST,TECK,TFD,WRIST
C
```

```
      C
      C      FROM SYSTEM LIBRRAY:  IPEEK,IPOKE,ISDAT
      C-------------------------------------------------------------------------
0001          SUBROUTINE OPCOM
      C
0002          LOGICAL*1 ICM
      C
0003          COMMON /BG/LIST(120),IPA(512)
0004          COMMON /FLA/IFL1,ICN,NBN,ICO,IFL2
0005          COMMON /CMD/ CM1(20),CM2(20),CM3(20)
0006          COMMON /FG/ IV(100)
0007          COMMON /WRK/ IW1(50),IW2(50),IW3(50),IW4(50)
      C
      C       ZEROING BIT 6 IN JOB STATUS WORD WITHOUT ZEROING ANY OTHER BITS.
      C
0008          CALL IPOKE("44,"177677.AND.IPEEK("44))
      C
0009          WRITE(7,500)
0010      500 FORMAT(1X,$,'>')
      C
      C       READ THE COMMAND LINE.
      C
0011          READ(5,10)  (IW1(I),I=1,25)
0012       10 FORMAT(25A2)
      C
      C       IDENTIFY THE CHARACTERSTRING.
      C
0013          CALL TECK(IW1,IW4,1,60,4,50)
      C
      C       TEST COMMAND LINE.
      C
0014          IF(IW4(1).GT.0.AND.IW4(2).GT.0.AND.IW4(2).LE.20) GO TO 20
0016       15 WRITE(7,501)
0017      501 FORMAT(1X,'?ILL CMD?')
0018          GO TO 600
      C
0019       20 GO TO (30,40,40,301,301,301,301,305,306,307,
                *307,307,307,307,307,307,307,307,307,307),IW4(2)
      C
      C       FOREGROUND NODE?
      C
0020       30 IF(IW4(5)-40) 15,15,35
      C
      C       SET A NODE HEAD AND PARAMETERS (FOREGROUND NODE)
      C
0021       35 CALL OPEN
0022          GO TO 600
      C
0023       40 IF(IW4(5).GE.41) GO TO 200
0025          IF(IW4(5).LE.20.OR.IW4(5).GE.39)GO TO 15
      C
      C       ACTIVATE A BACKGROUND NODE.
      C
0027          NN=IW4(6)
```

```
      C
      C     LEGAL NODE NUMBER?
      C
0028        IF(NN.GT.0.AND.NN.LT.10) GO TO 45
0030        WRITE(7,502)
0031    502 FORMAT(1X,'?ILL NODE NUMBER?')
0032        GO TO 600
      C
0033     45 ID=IW4(5)-20
0034        ITY=1000*ID+NN
      C
0035        DO 50 I=3,120,12
0036        IF(LIST(I).EQ.ITY) GO TO 55
0038     50 CONTINUE
      C
0039        IF(IW4(2).EQ.2) GO TO 70
0041        WRITE(7,503)
0042    503 FORMAT(1X,'NOT FOUND')
0043        GO TO 600
      C
0044     55 IF(IW4(2).EQ.2) GO TO 65
      C
      C     DEACTIVATE ONE NODE.
      C
0046     60 IF(LIST(I-2).LT.0) WRITE(7,504) CM2(ID),NN
0048    504 FORMAT(1X,A4,I1,1X,'NOT ACTIVE')
0049        LIST(I-2)=-1
0050        GO TO 600
      C
0051     65 IF(LIST(I-2).LT.0) GO TO 70
0053        WRITE(7,505) CM2(ID),NN
0054    505 FORMAT(1X,A4,I1,1X,'ALREADY ACTIVE')
0055        GO TO 600
      C
      C     ACTIVATE ONE NODE.
      C
0056     70 CALL FRENOD(LIST,IL)
0057        IF(IL.LT.0) GO TO 600
      C
      C     READ SAMPLE PERIOD.
      C
0059        WRITE(7,506)
0060    506 FORMAT(1X,$,'SAMPLE PERIOD:')
0061        READ(5,80) LIST(IL+1)
0062     80 FORMAT(I10)
      C
0063        LIST(IL+2)=ITY
      C
      C     SET INFORMATION ABOUT WERE TO RECEIVE INPUT SIGNALS FROM.
      C
0064        CALL TFD(CM1,LIST,IL,IW1,IW2)
      C
0065        GO TO (100,101,102,103,104,105,106,107,108,109,
     *110,110,110,110,110,110,110,110,110,110) ID
```

```
      C
0066    100 CALL GRAFST(LIST,IL,IPA,IW1)
0067        GO TO 600
      C
0068    101 CALL WRIST(LIST,IL,IPA,IW1,IW2)
0069        GO TO 600
      C
0070    102 CALL MAXMST(LIST,IL,IPA,IW1,IW2)
0071        GO TO 600
      C
0072    103 CALL WRIST(LIST,IL,IPA,IW1,IW2)
0073        GO TO 600
      C
0074    104 CALL GROUST(LIST,IL,IPA,IW1)
0075        GO TO 600
0076    105 CALL STAFST(LIST,IL,IPA,IW1)
0077        GO TO 600
0078    106 CALL NORMST(LIST,IL,IPA,IW1)
0079        GO TO 600
0080    107 CONTINUE
0081        GO TO 600
0082    108 CONTINUE
0083        GO TO 600
0084    109 CONTINUE
0085        GO TO 600
0086    110 CONTINUE
0087        GO TO 600
      C
      C
      C     ACTIVATE FOREGROUND NODES.
      C
0088    200 IV(1)=12
0089        IFL2=0
      C
      C     RECEIVE INFORMATION ABOUT ACTIVE NODES AND WAIT FOR COMPLETION.
      C
0090        CALL ISDAT(IV,1)
0091    210 IF(IFL2) 220,210,220
      C
      C     ACTIVATE OR DEACTIVATE ?
      C
0092    220 IW1(1)=9
0093        IF(IW4(2).EQ.3) IW1(1)=10
      C
0095        DO 250 I=2,IW4(1)
0096        IF(IW4(3*I).LE.0.OR.IW4(3*I).GE.21) GO TO 270
0098        DO 230 I1=1,IV(4)
0099    230 IF(IV(I1*4+2).EQ.IW4(I*3).AND.IV(I1*4+1).EQ.IW4(I*3-1)
           *-40) GO TO 250
0101        WRITE(7,245)
0102    245 FORMAT(1X,'THE NODE IS NOT ACTIVATED OR OCCUPIED')
0103        GO TO 600
0104    250 IW1(I)=IW4(I*3)
0105        DO 260 I=1,IW4(1)
```

```
0106     260 IV(I)=IW1(I)
0107         CALL ISDAT(IV,IW4(1))
0108         GO TO 600
0109     270 WRITE(7,502)
0110         GO TO 600
       C
       C     STOP REGULATION,START REGULATION,STOP COLLECTION,START COLLECTION.
       C
0111     301 IW1(1)=IW4(2)-3
0112         CALL ISDAT(IW1,1)
0113         GO TO 600
       C
       C     OPEN NEW FILE.
       C
0114     305 CALL CRNF(ICN,ICO,NBN,IFL1,IW1)
0115         GO TO 600
       C
       C     LIST ACTIVE NODES. (FOREGROUND)
       C
0116     306 I=7
0117         IF(IW4(4).EQ.6) I=IW4(4)
0119         CALL LSTF(IV,IFL2,CM3,I)
0120         GO TO 600
       C
       C     LIST ACTIVE NODES. (BACKGROUND)
       C
0121     307 I=7
0122         IF(IW4(4).EQ.6) I=IW4(4)
0124         CALL LSTB(LIST,CM2,I)
       C
       C
0125     600 WRITE(7,507)
0126     507 FORMAT(1X,'#####')
0127         RETURN
0128         END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: OPEN
C       ----
C
C       SUBTITLE: SETS A NODE HEAD AND PARAMETERS
C       ---------
C
C       LANGUAGE: FORTRAN IV + RT-11-FORTRAN
C       ---------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 2-SEP-77
C       ------------                                     ----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       IF THE NODE  "N1" IS IN FOREGROUND'S NODE LIST IT IS
C       FETCHED, ELSE THE START ROUTINE TO THE NODE TYPE,
C       IF ANY, IS RUN.THE NODE HEAD ARE SET BY THE SUBROUTINE
C       SNOD. IF THE NODE IS ACTIVE, THE PARAMETERS IS FETCHED
C       ELSE A REQUEST IS MADE TO FOREGROUND JOB TO SEE IF ENOUGH
C       ROOM IS AVAILABLE IN FOREGROUND'S PARAMETERLIST FOR THE
C       NEW NODE. FINALLY THE PARAMETERS ARE SET OR CHANGED BY
C       THE SUBROUTINE SPAR.
C       NOTE: THE ENGINEERNODE HAS NO NODE HEAD.
C
C       USAGE
C       =====
C
C       ARGUMENTS:   NONE
C       ----------
C
C       COMMON BLOCKS
C       -------------
C       FLA      -CONTAINING FLAGS AND POINTERS WITH INFORMATION
C                 ABOUT COLLECTING AND CALCULATING BLOCKS
C       CMD      -CONTAINING 4-CHARACTERS COMMAND NAME AND 2-
C                 CHARACTERS PARAMETER NAME
C       IM       -CONTAINING 1-CHARACTER CONSTANTS
C       RD       -CONTAINING INFORMATION ABOUT TYPE OF START ROUTINE,
C                 (IF ANY) AND NUMBER OF PARAMETERS FOR EACH TYPE OF NODE
C       FG       -CONTAINING ARRAYS FOR SENDING AND RECEIVING MESSAGES
C                 AND INFORMATION BETWEEN THE JOBS
C       WRK      -WORK AREA
C
C
```

```
      C     CHARACTERISTICS
      C     ===============
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 514 WORDS
      C
      C     SUBROUTINES REQUIRED: TECK,SNOD,SPAR
      C     ------------------------------------
      C     FROM SYSTEM LIBRARY: ISDAT
      C
      C-------------------------------------------------------------------------------
      C
0001        SUBROUTINE OPEN
      C
      C
0002        COMMON /FLA/IFL1,ICN,NBN,ICO,IFL2
0003        COMMON /CMD/CM1(20),CM2(20),CM3(20),IOP(100)
0004        COMMON /RD/ IRD(20)
0005        COMMON /FG/IV(100)
0006        COMMON /WRK/IW1(50),IW2(50),IW3(50),IW4(50)
      C
      C     NODENUMBER
      C
0007        N1=IW4(6)
      C
      C     TYPE OF NODE
      C
0008        ID=IW4(5)-40
      C
      C       IF IRD( )  <0      :THE ROUTINE HAS A START ROUTINE
      C       ABS(IRD( ))        =NUMBER OF PARAMETERS
      C       ABS(IRD( ))/100    =NUMBER OF NOT STATUS PARAMETERS
      C
      C
0009        NS=IABS(IRD(ID))
0010        NS1=NS/100
0011        NS2=MOD(NS,100)
0012        NS=2*NS2
      C
      C     ENGINEER-NOD?
      C
0013        IF(ID.EQ.20) GO TO 300
      C
      C     LEGAL NODENUMBER?
      C
0015        IF(N1.GE.1.AND.N1.LE.20) GO TO 10
      C
0017        WRITE(7,5)
0018      5 FORMAT(1X,'?ILL NODNUMBER?')
0019        RETURN
      C
      C     SEND FOR A NODE AND WAIT FOR COMPLETITION
      C
```

```
0020      10 IV(1)=5
0021         IV(2)=(N1-1)*12+1
0022         IV(3)=12
0023         IFL2=0
0024         CALL ISDAT(IV,3)
0025      20 IF(IFL2) 30,20,30
      C
      C      THE NODE IS IN IV
      C      IS THE NODE ACTIVE?
      C
0026      30 IF(IV(5).EQ.0) GO TO 50
      C
      C      IS THE NODE OF THE SAME TYPE AS THE REQUESTED?
      C
0028         IF(IV(6).EQ.ID) GO TO 310
      C
0030         WRITE(7,45)
0031      45 FORMAT(1X,'THE NODE IS ALREADY ACTIVE AS ANOTHER TYPE')
0032         RETURN
      C
0033      50 NS=2*NS1
      C
      C      HAS THE NODE A START ROUTINE?
      C
0034         IF(IRD(ID).GT.0) GO TO 210
0036         IP5=0
0037         IP6=0
      C
0038         GO TO(110,120,120,120,120) ID-10
      C
      C      START UP THE PRB-GENERATOR.
      C
0039     110 WRITE(7,508)
0040     508 FORMAT(1X,$,'NUMBER OF BITS:')
0041         READ(5,509) NA
0042     509 FORMAT(I15)
0043         IF(NA.GE.3.AND.NA.LE.15) GO TO 115
0045         WRITE(7,510)
0046     510 FORMAT(1X,'NOT IN RANGE')
0047         GO TO 110
0048     115 NS=NS+NA*2+1
0049         CALL PRBST(IW3,IW3(NA+1),NA)
0050         IP5=NA
0051         IW3(NA*2+1)=NA
0052         GO TO 210
      C
0053     120 NS=NS+2
0054         IW3(1)=0
0055         IW3(2)=0
0056         GO TO 210
      C
      C
      C
      C      ENOUGH ROOM IN FORGROUND'S PARAMETER VECTOR?
```

```
      C     ASK AND WAIT FOR COMPLETITION
      C
0057    210 IV(1)=11
0058        IV(2)=NS
0059        IFL2=0
0060        CALL ISDAT(IV,2)
0061    220 IF(IFL2) 230,220,230
      C
      C     ENOUGH ROOM?
      C
0062    230 IF(IV(4).GE.0) GO TO 270
      C
0064        WRITE(7,235)
0065    235 FORMAT(1X,'NOT ENOUGH ROOM IN PARAMETER VECTOR')
0066        RETURN
      C
      C     SET A NODE IN IV
      C
0067    270 IP2=IV(4)
0068        IV(8)=NS1*2+IP2
0069        IV(9)=IP5+IV(8)
0070        IV(10)=IP6+IV(8)
0071        IP3=IP2+NS-1
0072        CALL SNOD(ID,N1,IP2,IP3)
      C
      C
      C     SEND THE NODE TO FOREGROUND
      C
0073        CALL ISDAT(IV,12)
      C
0074        DO 290 I=4,48
0075    290 IV(I)=0
      C
      C     ADD THE PARAMETERS FROM THE START ROUTINE
      C
0076        J1=NS1*2+3
0077        J=NS-J1+3
      C
0078        DO 350 I=1,J
0079    350 IV(J1+I)=IW3(I)
0080        GO TO 340
      C
      C     SEND FOR THE PARAMETERS AND WAIT FOR COMPLETITION
      C
0081    300 IP2=27
0082        IP3=74
0083        GO TO 320
      C
0084    310 IP2=IV(7)
0085        IP3=IV(8)
0086        WRITE(7,315)
0087    315 FORMAT(1X,'THE NODE IS ALREADY ACTIVE')
0088    320 IV(1)=6
0089        IV(2)=IP2
```

```
0090          IV(3)=IP3-IP2+1
0091          IFL2=0
0092          CALL ISDAT(IV,3)
0093      330 IF(IFL2) 340,330,340
          C
          C     THE PARAMETERS ARE IN IV
          C     SET OR CHANGE THE PARAMETERS IN IV
          C
0094      340 CALL SPAR(ID,IP2,NS)
          C
          C
          C
          C     SEND THE PARAMETERS TO FOREGROUND JOB
          C
0095          CALL ISDAT(IV,NS+3)
          C
0096          RETURN
0097          END
```

```
C-------------------------------------------------------------------------------
C
C     NAME: PIDR                                                NUMBER:
C     ----                                                      -------
C
C     SUBTITLE: PERFORMS PID CONTROL
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C     KEYWORDS: PID,SAMPLED
C     ---------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON      DATE:1977-07-05
C     ------------                                ----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                 VERSION:
C     --------                                                  -------
C
C-------------------------------------------------------------------------------
C
C     USAGE
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ---------
C
C     PIDR( U,Y,UR,YO,RI,D,G,TI,TD,GD,TS )
C
C     Y    -OUTPUT SIGNAL FROM PROCESS, (I)
C     UR   -INPUT REFERENCE VALUE, (I)
C     YO   -OLD VALUE OF PROCESS OUTPUT,  (I/O)
C     RI   -INTEGRATING TERM (INTERNAL VARIABLE), (I/O)
C     D    -DERIVATIVE TERM (INTERNAL VARIABLE), (I/O)
C     U    -RESULT, INPUT VALUE TO PROCESS, (O)
C     G    -GAIN, (I)
C     TI   -INTEGRATING TIME CONSTANT, (I)
C     TD   -DERIVATIVE CONSTANT, (I)
C     GD -DERIVATIVE FILTER CONSTANT, (I)
C     TS   -SAMPLING PERIOD, (I)
C
C     NOTE: RI AND D ARE INITIALLY SET TO ZERO
C     ----
C
C
```

```
C     METHOD
C     ======
C
C     INTEGRATING TERM RI, DERIVATIVE TERM D AND
C     OLD PROCESS OUTPUT ARE UPDATED. NO INTEGRATING
C     IF TI.LE.0 AND NO DERIVATIVE IF TD.LE.0.
C
C     CHARACTERISTICS:
C     ================
C
C     SIZE:
C     -----
C     PDP11/03:108 WORDS
C
C
C-----------------------------------------------------------------------
C     PROGRAM
C     =======
C
0001        SUBROUTINE PIDR(U,Y,UR,YO,RI,D,G,TI,TD,GD,TS)
C
0002        E=UR-Y
C
C     PROPRTIONAL
C
0003        U=G*E
C
C     DERIVATIVE
C
0004        IF(TD.GT.0.0) GO TO 10
0006        D=0.0
0007        GO TO 20
0008     10 AL=G*TD/TS
0009        BE=TD/(TD+GD*TS)
0010        D=BE*D+AL*(1.0-BE)*(YO-Y)
0011        U=U+D
C
C     INTEGRATING
C
0012     20 IF(TI.GT.0.0) GO TO 30
0014        RI=0.0
0015        GO TO 40
0016     30 RI=RI+G*TS/TI*E
0017        U=U+RI
C
C     UPDATE OLD PROCESS OUTPUT
C
0018     40 YO=Y
0019        RETURN
0020        END
```

```
C-------------------------------------------------------------------------------
C     NAME: PRB                                                 NUMBER:
C     ----                                                      ------
C
C     SUBTITLE:
C     ---------
C     GENERATE A NEW STATE IN A PRBS-GENERATOR
C
C     LANGUAGE: FORTRAN IV
C     --------------------
C
C     KEYWORDS:
C     ---------
C
C     IMPLEMENTOR: STURE LINDAHL                                DATE: 1970-02-10
C     -------------------------                                       ----------
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                 VERSION:
C     ---------                                                 --------
C
C-------------------------------------------------------------------------------
C
C     USAGE:
C     ======
C     PROGRAM TYPE: SUBROUTINE
C     ------------------------
C
C     ARGUMENTS:
C     ----------
C
C     PRB(LA,LX,Y,NA,AMP)
C
C     LA    -VECTOR,CONTAINING THE FEEDBACK-POLYNOMIAL,SIZE(NA),(I/O)
C     LX    -VECTOR,CONTAINING THE ACTUAL STATE,SIZE(NA),(I/O)
C     Y     -OUTPUT FROM PRBS-GENERATOR (O)
C     NA    -NUMBER OF BITS IN THE SHIFTREGISTER (I)
C     AMP   -SPECIFIED AMPLITUD OF OUTPUT-SIGNAL (I)
C
C     NOTES:
C     ------
C    1)   LA CAN BE ASSIGNED VALUES IN A
C     START ROUTINE CALLED PRBST
C
C     REFERENCES
C     ----------
C    1)   W.W. PETERSON, ERROR-CORRECTING CODES
C    2)   B. ROSENGREN AND I. NORDH, KONSTRUKTION AV PRBS-GENERATOR
C    3)   M. RUDEMO, ON PSEUDO-RANDOM NOISE GENERATED
C     BY SHIFT REGISTERS
```

```
      C
      C     CHARACTERISTICS:
      C     ================
      C
      C     SUBROUTINES REQUIRED: NONE
      C     ~~~~~~~~~~~~~~~~~~~~~~~~~~
      C
      C     SIZE:
      C     ~~~~~
      C     PDP11/03:  95 WORDS
      C
      C     REVISIONS:
      C     ~~~~~~~~~~
      C     LEIF RADING,TYKE PAULSSON  ,1977-07-04
      C
      C
      C-----------------------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE PRB(LA,LX,Y,NA,AMP)
      C
0002        DIMENSION LA(1),LX(1)
      C
0003        ISL=0
      C
0004        DO 10 I=1,NA
0005     10 ISL=ISL+LA(I)*LX(I)
      C
0006        DO 20 I=2,NA
0007        J=NA-I+1
0008     20 LX(J+1)=LX(J)
      C
0009        LX(1)=MOD(ISL,2)
0010        Y=FLOAT(2*LX(1)-1)*AMP
      C
0011        RETURN
0012        END
```

```
C-----------------------------------------------------------------------
C     NAME: PRBST                                          NUMBER:
C     ----                                                 ------
C
C     SUBTITLE:
C     ---------
C     SUBROUTINE TO START UP THE PRB-SUBROUTINE
C
C     LANGUAGE: FORTRAN IV
C     --------------------
C
C     KEYWORDS:
C     ---------
C
C     IMPLEMENTOR: STURE LINDAHL                           DATE: 1970-02-10
C     --------------------------                           ----------
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                            VERSION:
C     ---------                                            --------
C
C-----------------------------------------------------------------------
C
C     USAGE:
C     ======
C     PROGRAM TYPE: SUBROUTINE
C     ------------------------
C
C     ARGUMENTS:
C     ----------
C
C     PRBST(LA,LX,NA)
C
C     LA    -VECTOR,CONTAINING THE FEEDBACK-POLYNOMIAL,SIZE(NA) (O)
C     LX    -VECTOR,CONTAINING THE ACTUAL STATE,SIZE(NA) (O)
C     NA    -NUMBER OF BITS IN THE SHIFTREGISTER (I)
C
C     NOTES:
C     ------
C    1)   NA MUST BE IN THE RANGE 3.LE.NA.LE.15
C
C     REFERENCES
C     ----------
C    1)   W.W. PETERSON, ERROR-CORRECTING CODES
C    2)   B. ROSENGREN AND I. NORDH, KONSTRUKTION AV PRBS-GENERATOR
C    3)   M. RUDEMO, ON PSEUDO-RANDOM NOISE GENERATED
C     BY SHIFT REGISTERS
C
C     CHARACTERISTICS:
C     ================
```

```
      C
      C     SUBROUTINES REQUIRED: NONE
      C     --------------------------
      C
      C     SIZE:
      C     -----
      C     PDP11/03   158 WORDS
      C
      C     REVISIONS:
      C     ----------
      C     LEIF RADING,TYKE PAULSSON  ,1977-07-04
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE PRBST(LA,LX,NA)
      C
0002        DIMENSION LA(1),LX(1)
      C
0003        DO 1 I=1,NA
0004        LA(I)=0
0005      1 LX(I)=0
      C
0006        LA(1)=1
0007        LX(1)=1
0008        LA(NA)=1
      C
0009        IVAL=NA-4
0010        IF(IVAL) 15,15,2
0011      2 GO TO (5,6,7,8,9,10,11,12,13,14,15),IVAL
      C
      C
0012      5 LA(1)=0
0013        LA(2)=1
0014      6 RETURN
0015      7 LA(1)=0
0016        LA(3)=1
0017        RETURN
0018      8 LA(1)=0
0019        LA(2)=1
0020        LA(3)=1
0021        LA(4)=1
0022        RETURN
0023      9 LA(1)=0
0024        LA(4)=1
0025        RETURN
0026     10 LA(1)=0
0027        LA(3)=1
0028        RETURN
0029     11 LA(1)=0
0030        LA(2)=1
0031        RETURN
```

```
0032      12 LA( 4 )=1
0033         LA( 6 )=1
0034         RETURN
0035      13 LA( 3 )=1
0036         LA( 4 )=1
0037         RETURN
0038      14 LA( 6 )=1
0039         LA( 10 )=1
0040      15 RETURN
0041         END
```

```
C-----------------------------------------------------------------------
C
C     NAME: PULS                                          NUMBER:
C     ----                                                ------
C
C     SUBTITLE:GENERATES A PULS SIGNAL
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ----------
C
C     KEYWORDS: PULS
C     ----------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON      DATE:1977-07-05
C     ------------                                ------
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                           VERSION:
C     ---------                                           -------
C
C-----------------------------------------------------------------------
C
C     PURPOSE: GENERATES A PULS SIGNAL
C     ========
C
C     USAGE:
C     ======
C     PROGRAM TYPE: SUBROUTINE
C     -------------
C
C     ARGUMENTS:
C     ----------
C
C     PULS (X,AM,FR,TS,PL,Y)
C
C     X     -TIME RELATIVE PERIOD-START, INTERNAL VARIABLE, (I)
C     AM    -AMPLITUD, (I)
C     TS    -SAMPLE PERIOD, (I)
C     FR    -FREQUENCY(RAD/.), (I)
C     PL    -PULSE LENGHT, (I)
C     Y     -GENERATED SIGNAL, (O)
C
C     NOTES: X IS INITIALLY SET TO ZERO
C     -------
C
C     METHOD:
C     =======
C
C     Y=AMP DURING THE TIME OF THE PULSE LENGHT
```

```
      C
      C     CHARACTERISTICS:
      C     ================
      C
      C     EXECUTION TIME:
      C     ---------------
      C     PDP 11: 1.5 MS
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 45 WORDS
      C
      C
      C-----------------------------------------------------------------------
      C     PROGRAM:
      C     =======
      C
      C
0001        SUBROUTINE PULS(X,AM,FR,TS,PL,Y)
      C
      C
0002        TP =1.0/FR
0003        X=X+TS
0004        Y=AM
0005        IF(X.GT.TP) X=X-TP
0007        IF(X.LT.PL) RETURN
0009        Y=0.0
0010        RETURN
0011        END
```

```
C-------------------------------------------------------------------------------
C     NAME: RAMP                                                  NUMBER:
C     ----                                                        ------
C
C     SUBTITLE:GENERATES A RAMP SIGNAL
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C     KEYWORDS:RAMP
C     ---------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                      DATE:1977-07-0
C     ------------                                                ----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                   VERSION:
C     --------                                                    -------
C
C-------------------------------------------------------------------------------
C
C     USAGE:
C     ======
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ---------
C
C     RAMP( X,XK,TS,Y )
C
C     X    -STATE, (I/O)
C     XK   -SLOPE, (I)
C     TS   -SAMPLE PERIOD, (I)
C     Y    -GENERATED SIGNAL, (O)
C
C     METHOD:
C     ======
C     Y=X+XK*TS
C
C     CHARACTERISTICS:
C     ===============
C
C     SUBROUTINES REQUIRED: NONE
C     --------------------
C
C     SIZE:
C     ----
C     PDP11/03:   20 WORDS
```

```
      C
      C
      C-----------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE RAMP( X, XK, TS, Y )
0002        X=X+XK*TS
0003        Y=X
0004        RETURN
0005        END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: RS
C       ------
C
C       SUBTITLE: SEARCH-ROUTINE
C       ------------------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ------------------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 12-SEP-77
C       ------------------                                ------
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       FINDS (IF ANY) THE SMALLEST CONTINUOUS AREA IN THE
C       PARAMETER AREA THAT IS >= THE WANTED
C
C       USAGE
C       =====
C
C       ARGUMENTS:
C       ----------
C
C       RS(IL,NS1,LIST,IW)
C
C       IL      -POINTER TO THE FIRST WORD IN THE PARAMETERAREA,(I)
C       NS1     -NUMBER OF WORDS ASKED FOR,(I)
C       LIST    -ARRAY CONTAINING NODE HEADS, (I)
C       IW      -WORK AREA
C
C       CHARACTERISTICS
C       ===============
C
C       SIZE:
C       -----
C       PDP11/03:214 WORDS
C
C-------------------------------------------------------------------------------
C
C      PROGRAM
C      =======
C
0001        SUBROUTINE RS(IL,NC,LIST,IW)
C
```

```
0002          DIMENSION LIST(1),IW(1)
      C
0003          IW(1)=26
0004          K=1
      C
      C       SEARCHES THROUGH THE NODES AND REGISTRATES THE OCCUPIED PARTS
      C       OF THE PARAMETER AREA.
      C
0005          DO 10 I=1,109,12
0006          IF(LIST(I).LT.0.OR.I.EQ.IL) GO TO 10
0008          K=K+1
0009          IW(K)=LIST(I+3)
0010          K=K+1
0011          IW(K)=LIST(I+7)
0012       10 CONTINUE
      C
0013          K=K+1
0014          IW(K)=513
0015          I3=0
0016          I4=10000
      C
0017          DO 25 I=1,K,2
0018          I2=10000
      C
0019          DO 20 J=2,K,2
      C
      C       FINDS A FREE CONTINOUS  AREA IN THE PARAMETER AREA
      C
0020          I1=IDIM(IW(J),IW(I))
0021          IF(I1.EQ.0.OR.I1.GT.I2) GO TO 20
0023          I2=I1
0024       20 CONTINUE
      C
0025          IF(I2.LE.NC.OR.I4.LE.I2) GO TO 25
0027          I3=IW(I)
0028          I4=I2
0029       25 CONTINUE
      C
0030          IF(I3.EQ.0) GO TO 30
0032          LIST(IL+3)=I3+1
0033          LIST(IL+7)=I3+NC
      C
0034          RETURN
      C
0035       30 WRITE(7,500)
0036      500 FORMAT(1X,'NOT ENOUGH ROOM')
0037          LIST(IL)=-1
0038          IL=-1
      C
0039          RETURN
0040          END
```

```
FORTRAN IV     V01C-03F+  WED 09-NOV-77 17:50:05                    PAGE 001

      C-------------------------------------------------------------------------------
      C
      C       NAME: RS1
      C       ~~~~~
      C
      C       SUBTITLE: SEARCH-ROUTINE
      C       ~~~~~~~~~~
      C
      C       LANGUAGE: FORTRAN + RT-11-FORTRAN
      C       ~~~~~~~~~
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 12-SEP-77
      C       ~~~~~~~~~~~~~~~~                                 ~~~~~~
      C
      C       INSTITUTE:
      C       ~~~~~~~~~
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-------------------------------------------------------------------------------
      C
      C       PURPOSE AND METHOD
      C       ==================
      C
      C       FINDS (IF ANY) THE SMALLEST CONTINUOUS AREA IN THE
      C       PARAMETER AREA THAT IS >= THE WANTED
      C
      C       USAGE
      C       =====
      C
      C       ARGUMENTS:
      C       ~~~~~~~~~
      C
      C       RS1(IL,NS1,LIST,IW)
      C
      C       IL       -POINTER TO THE FIRST WORD IN THE PARAMETER AREA,(I)
      C       NS1      -NUMBER OF WORDS ASKED FOR,(I)
      C       LIST     -ARRAY CONTAINING NODE HEADS, (I)
      C       IW       -WORK AREA
      C
      C       CHARACTERISTICS
      C       ===============
      C
      C       SIZE:
      C       ~~~~
      C       PDP11/03:  154 WORDS
      C
      C-------------------------------------------------------------------------------
      C
0001          SUBROUTINE RS1(IL,NS1,LIST,IW)
      C
0002          DIMENSION LIST(1),IW(1)
      C
0003          IW(1)=74
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 17:50:05                PAGE 002

0004          K=1
      C
      C         SEARCHES THROUGH THE NODES AND REGISTRATES THE OCCUPIED PARTS OF
      C         THE PARAMETER AREA
      C
0005          DO 10 I=2,240,12
0006          IF(LIST(I).EQ.0) GO TO 10
0008          K=K+1
0009          IW(K)=LIST(I+2)
0010          K=K+1
0011          IW(K)=LIST(I+6)
0012       10 CONTINUE
      C
0013          K=K+1
0014          IW(K)=401
0015          IL=0
0016          I4=10000
      C
0017          DO 25 I=1,K,2
0018          I2=10000
      C
0019          DO 20 J=2,K,2
      C
      C       FINDS A FREE CONTINUOUS  AREA IN THE PARAMETER AREA
      C
0020          I1=IDIM(IW(J),IW(I))
0021          IF(I1.EQ.0.OR.I1.GT.I2) GO TO 20
0023          I2=I1
0024       20 CONTINUE
      C
0025          IF(I2.LE.NS1.OR.I4.LE.I2) GO TO 25
0027          IL=IW(I)+1
0028          I4=I2
0029       25 CONTINUE
0030          RETURN
0031          END
```

```
C-------------------------------------------------------------------------------
C
C     NAME: SINU                                                NUMBER
C     ------                                                    ------
C
C     SUBTITLE:GENERATES A SINUSOIDAL SIGNAL
C     -------------
C
C     LANGUAGE: FORTRAN IV
C     ----------
C
C     KEYWORDS:SINUS SIGNAL
C     ----------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON      DATE:1977-07-05
C     ---------------                             ------
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                             VERSION:
C     ---------                                             --------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE: GENERATES A SINUS SIGNAL
C     =======
C
C     USAGE:
C     ======
C
C     PROGRAM TYPE: SUBROUTINE
C     -------------------
C
C     ARGUMENTS:
C     ----------
C
C     SINU( X, AM, FR, TS, DF, Y )
C
C     X     -ANGLE( RAD ), ( INTERNAL VARIABLE ), ( I )
C     AM    -AMPLITUD, ( I )
C     FR    -FREQUENCY ( RAD/S ), ( I )
C     TS    -SAMPLE PERIOD, ( I )
C     DF    -PHASE LAG, ( I )
C     Y     -GENERATED SIGNAL, ( O )
C
C     NOTE: X IS INITIALLY SET TO ZERO
C     ------
C
C     METHOD:
C     ======
C
```

```
      C     Y=AM*SIN(FR*TS+DF)
      C
      C     CHARACTERISTICS:
      C     ================
      C
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 50 WORDS
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE SINU(X,AM,FR,TS,DF,Y)
      C
0002        DATA P/6.283184/
0003        X=X+FR*TS*P
0004        IF(X.GT.P) X=X-P
0006        Y=AM*SIN(X+DF)
0007        RETURN
0008        END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: SNOD
C       ----
C
C       SUBTITLE: SETS A NODE HEAD
C       ---------
C
C       LANGUAGE: FORTRAN + PDP11/03 SYSTEM LIBRARY
C       ---------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON              DATE: 2-SEP-7
C       ------------                                         ----
C
C       INSTITUTE:
C       ---------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       A NODE HEAD IS SET IN IV BY ASKING FOR
C       SAMPLE PERIOD, INPUT SIGNALS, AND OUTPUT
C       SIGNAL. THE ANSWERS ARE TESTED.
C
C       USAGE
C       =====
C
C       ARGUMENTS:
C       ----------
C       ID      - NODE TYPE,(I)
C       N       - NODE NUMBER,(I)
C       IP1     - POINTER TO THE FIRST WORD OF THE NODE'S
C                 PARAMETER AREA IN THE VECTOR IPF IN FG,(I)
C       IP2     - POINTER TO THE LAST WORD OF THE NODE'S
C                 PARAMETER AREA,(I)
C
C       COMMON BLOCKS:
C       --------------
C       FG      - ARRAY FOR SENDING AND RECEIVING MESSAGES
C                 AND INFORMATION BETWEEN THE JOBS
C       WRK     - WORK AREA
C
C       SIZE:
C       ----
C       PDP11/03: 373 WORDS
C
C       SUBROUTINES REQUIRED:TECK
C       ---------------------
C
C-------------------------------------------------------------------------------
```

```
      C
0001          SUBROUTINE SNOD(ID,N,IP1,IP2)
      C
      C
0002          COMMON /FG/IV(100)
0003          COMMON /WRK/IW1(50),IW2(50),IW3(50),IW4(50)
      C
      C       USAGE-CODE IN FOREGROUND JOB
      C
0004          IV(1)=7
      C
      C       LOCATION IN FOREGROUND'S LF-ARRAY
      C
0005          IV(2)=(N-1)*12+1
0006          IV(3)=IV(2)+11
0007          IV(4)=-1
      C
      C       SET THE SAMPLE PERIOD
      C
0008          WRITE(7,5)
0009        5 FORMAT(1X,$,'SAMPLE PERIOD:')
0010          READ(5,10) IV(5)
0011       10 FORMAT(I5)
0012          IV(5)=-IV(5)
      C
      C       SET NODE-TYPE
      C
0013          IV(6)=ID
      C
      C       SET POINTERS FOR PARAMATERS LOCATION IN
      C       FOREGROUND'S IPF-ARRAY
      C
0014          IV(7)=IP1
0015          IV(11)=IP2
      C
0016          DO 20 I=12,15
0017       20 IV(I)=0
      C
      C       FROM HERE TO 110, THE INPUT SIGNALS WILL BE SET
      C
0018       30 WRITE(7,35)
0019       35 FORMAT(1X,$,'INPUT SIGNAL:')
0020          READ(5,40) (IW2(I),I=1,25)
0021       40 FORMAT(25A2)
0022          CALL TECK(IW2,IW4,40,59,4,40)
      C
      C       WHICH INPUTSIGNAL?
      C
0023          IF(IW4(1)) 50,120,60
      C
0024       50 WRITE(7,55)
0025       55 FORMAT(1X,'ILL CMD')
      C
0026          GO TO 30
```

```
      C
0027     60 DO 110 I=1,IW4(1)
0028        N1=IW4(I*3)
0029        N2=IW4(I*3-1)
0030        IF(N2.NE.40) GO TO 90
      C
0032        IF(N1.GE.1.AND.N1.LE.8) GO TO 80
      C
0034        WRITE(7,75)
0035     75 FORMAT(1X,'ILL SIGNAL NUMBER')
      C
0036        GO TO 30
      C
0037     80 IV(I+11)=1+2*N1
0038        GO TO 110
      C
0039     90 IF(N1.GE.1.AND.N1.LE.20) GO TO 100
      C
0041        WRITE(7,95)
0042     95 FORMAT(1X,'ILL NODE NUMBER')
      C
0043        GO TO 30
      C
0044    100 IV(I+11)=-N1
0045    110 CONTINUE
      C
      C     FROM HERE TO 140, THE OUTPUT SIGNAL WILL BE SET
      C
0046    120 WRITE(7,125)
0047    125 FORMAT(1X,$,'OUTPUT SIGNALS:')
0048        READ(5,40) (IW2(I),I=1,25)
0049        CALL TECK(IW2,IW4,39,39,4,40)
      C
      C     WHICH OUTPUT SIGNAL?
      C
0050        IF(IW4(1)) 130,160,140
      C
0051    130 WRITE(7,135)
0052    135 FORMAT(1X,'ILL CMD')
      C
0053        GO TO 120
      C
0054    140 IF(IW4(3).GE.1.AND.IW4(3).LE.4) GO TO 150
      C
0056        WRITE(7,75)
      C
0057        GO TO 120
      C
0058    150 IV(15)=17+2*IW4(3)
      C
0059    160 RETURN
0060        END
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 17:53:46                PAGE 001

C-----------------------------------------------------------------------
C
C     NAME: SPAR
C     ~~~~
C
C     SUBTITLE: SET PARAMETERS
C     ~~~~~~~~~~
C
C     LANGUAGE: FORTRAN + RT-11-FORTRAN
C     ~~~~~~~~~~
C     IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 7-SEP-77
C     ~~~~~~~~~~~~~~                                   ~~~~~~
C
C     INSTITUTE:
C     ~~~~~~~~~~
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-----------------------------------------------------------------------
C
C     PURPOSE AND METHOD
C     ==================
C
C     THE PARAMETERS OF A NODE ,PLACED IN IV, CAN BE SET OR
C     CHANGED, LISTED AND CLOSED (SEND TO FOREGROUND JOB).
C     WHEN THE PROGRAM PRINTS "*", THE OPERATOR ANSWERS WITH
C     EITHER CLOSE(CL), LIST(LI) OR A PARAMETER NAME AND ITS VALUE.
C
C     USAGE
C     =====
C
C     ARGUMENTS:
C     ~~~~~~~~~~
C     SPAR(ID,IP2,NS)
C     ID      -NODETYPE,(I)
C     IP2     -POINTER TO THE FIRST WORD OF THE NOD'S PARAMETERAREA
C              IN THE VECTOR IPF IN FOREGROUND JOB,(I)
C     NS      -NUMBER OF PARAMETERS ALLOWED TO BE CHANGED
C
C     COMMON BLOCKS:
C     ~~~~~~~~~~~~~~
C     CMD     -CONTAINING 4-CHARACTERS COMMAND NAME AND 2-CHARACTERS
C              PARAMETER NAME
C     IM      -CONTAINING 1-CHARACTER CONSTANTS
C     RD      -CONTAINING INFORMATION ABOUT TYPE OF START ROUTINE
C              (IF ANY) AND NUMBER OF PARAMETERS FOR EACH TYPE OF NODE
C     FG      -CONTAINING ARRAYS FOR SENDING AND RECEIVING MESSAGES
C              AND INFORMATION BETWEEN THE JOBS
C     WRK     -WORK AREA
C
C     CHARACTERISTICS
C     ===============
C
C     SIZE:
```

```
      C     -----
      C     PDP11/03:  504 WORDS
      C
      C     SUBROUTINE REQUIRED:TECK
      C     ------------------------
      C     FROM SYSTEM LIBRARY: ITTINR
      C
      C-------------------------------------------------------------------------------
      C
0001        SUBROUTINE SPAR(ID,IP2,NS)
      C
0002        LOGICAL*1 IW2,ICM
      C
0003        COMMON /CMD/CM(60),IOP(100)
0004        COMMON /FG/IV(100)
0005        COMMON/WRK/IW1(50),IW2(100),IW3(50),IW4(50)
0006        COMMON /IM/ICM(7)
0007        COMMON /RD/IRD(10)
      C
      C     USAGE-CODE IN FOREGROUND JOB
      C
0008        IV(1)=8
      C
      C     LOCATION IN FOREGROUND'S IPF-ARRAY
      C
0009        IV(2)=IP2
0010        IV(3)=IP2+NS-1
      C
      C     LOCATE THE PARAMETER NAMES IN THE PARAMETER NAME-ARRAY
      C
0011        IP=1
0012        IF(ID.NE.20) GO TO 5
0014        IP=99
0015        GO TO 20
      C
0016      5 IJ=ID-1
0017        IF(IJ.EQ.0) GO TO 20
0019        DO 10 I=1,IJ
0020     10     IP=IP+IABS(IRD(I)/100)
      C
0021     20 IP1=IP+IABS(IRD(ID))/100-1
0022     30 WRITE(7,35)
0023     35 FORMAT(1X,$,'*')
      C
      C     READ A PARAMETER NAME
      C
0024        IW2(1)=ITTINR()
0025        IW2(2)=ITTINR()
      C
      C     IDENTIFY THE CHARACTERSTRING
      C
0026        CALL TECK(IW2,IW4,IP,100,2,2)
      C
      C     PARAMETER NAME ,LIST(LI), OR CLOSE(CL)?
```

```
      C
0027          IF(IW4(1).GE.0) GO TO 80
      C
      C       ENGINEER-NODE?
      C
0029          IF(ID.EQ.20) GO TO 92
      C
0031       70 WRITE(7,75)
0032       75 FORMAT(1X,'ILL CMD')
0033          READ(5,76)
0034       76 FORMAT()
0035          GO TO 30
      C
0036       80 IF(IW4(2)-99) 90,120,170
      C
0037       90 I=2*(IW4(2)-IP)+4
0038          IF(IW2(2).NE.ICM(6)) IW2(3)=ITTINR()
0040          READ(5,91) X
0041       91 FORMAT(F10.0)
      C
      C       ILLEGAL PARAMETER NAME?
      C
0042          IF(IW4(2).LT.IP.OR.IW4(2).GT.IP1.OR.ID.EQ.20) GO TO 70
0044          IF(IW4(2).GE.IP+NS/2) GO TO 70
0046          CALL CV(IV(I),X)
0047          GO TO 30
      C
      C       ENGINEER-NODE
      C       READ AND IDENTIFY THE CHARACTER-STRING
      C
0048       92 READ(5,93)(IW2(L),L=3,8) ,X
0049       93 FORMAT(6A1,F10.0)
0050          CALL TECK(IW2,IW4,39,40,4,8)
      C
      C       ILLEGAL SIGNAL NAME?
      C
0051          IF(IW4(1).LT.1) GO TO 70
      C
      C       LEGAL SIGNAL NUMBER?
      C
0053          IF(IW4(2).EQ.40.AND.IW4(3).LE.8.AND.IW4(3).GE.1.OR.
             *IW4(2).EQ.39.AND.IW4(3).LE.4.AND.IW4(3).GE.1) GO TO 100
      C
0055          WRITE(7,95)
0056       95 FORMAT(1X,'?ILL SIGNALNUMBER?')
      C
      C       ZEROLEVEL OR SLOPE?
      C
0057      100 IF(IW4(4).EQ.1.OR.IW4(4).EQ.2) GO TO 110
      C
0059          WRITE(7,105)
0060      105 FORMAT(1X,'?ZEROLEVEL OR SLOPE?')
0061          GO TO 30
      C
```

```
0062     110 II=0
0063         IF(IW4(2).EQ.39) II=32
0065         I=II+4*IW4(3)+2*(IW4(4)-1)
0066         CALL CV(IV(I),X)
0067         GO TO 30
       C
       C     LIST
       C
0068     120 READ(5,130)LU1
0069     130 FORMAT(1X,I1)
0070         LU=7
0071         IF(LU1.EQ.6) LU=6
0073         IF(ID.EQ.20) GO TO 150
       C
0075         I3=4
0076         DO 140 I=IP,IP1
0077         CALL CV(X,IV(I3))
       C
0078         WRITE(LU,135) IOP(I),X
0079     135 FORMAT(1X,A2,1H=,E12.5)
       C
0080     140 I3=I3+2
0081         GO TO 30
       C
0082     150 DO 160 I=4,50,4
0083     160 CALL WW(IV(I),2)
       C
0084         GO TO 30
       C
       C     READY!
       C
0085     170 READ(5,130)
       C
0086         RETURN
0087         END
```

```
C-------------------------------------------------------------------------------
C     NAME: STAF                                                    NUMBER:
C     ----                                                          ------
C
C     SUBTITLE: RECURSIVE ESTIMATION OF MEAN VALUE
C     ---------
C                ,VARIANCE AND STANDARD DEVIATION.
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C     KEYWORDS: MEAN VALUE, VARIANCE, STANDARD DEVIATION, WEIGHTING FACT
C     ---------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                        DATE:1977-07-0
C     ------------                                                  -----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                                     VERSION:
C     --------                                                      -------
C
C-------------------------------------------------------------------------------
C
C     USAGE:
C     =====
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ---------
C
C     STAF( X,XM,VAR,DEV,WF,WFC,ND )
C
C     X    -INPUT VALUE, (I)
C     XM   -ESTIMATED MEAN VALUE,  (I/O)
C     VAR  -ESTIMATED VARIANCE, (I/O)
C     DEV  -ESTIMATED STANDARD DEVIATION, (O)
C     WF   -WEIGHTING FACTOR, (I/O)
C     WFC  -WEIGHTING CHANGE FACTOR, (I)
C     ND   -NUMBER OF DATA, (I/O)
C
C     METHOD:
C     ======
C
C     WEIGHTING FACTOR
C
C     WF=WFC*(WF-1.0)+1.0
C
C     MEAN VALUE
C
```

```
      C     XM=X+(XM-X)*(ND-1)*WF/ND
      C
      C     VARIANCE
      C
      C     VAR=(XM-X)**2+(ND-2)*(VAR-(XM-X)**2)*WF/(ND-1.0)
      C
      C     DEVIATION
      C
      C     DEV=SQRT(VAR)
      C
      C     CHARACTERISTICS:
      C     ================
      C
      C     SUBROUTINES REQUIRED: NONE
      C     --------------------------
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 96 WORDS
      C
      C
      C-----------------------------------------------------------------------
      C     PROGRAM:
      C     ========
0001        SUBROUTINE STAF(X,XM,VAR,DEV,WF,WFC,ND)
      C
      C     UPDATE WEIGHTING FACTOR.
      C
0002        WF=WFC*(WF-1.0)+1.0
      C
0003        ND=ND+1
0004        XND=ND
0005        XD=XM-X
      C
      C     ESTIMATE MEAN VALUE.
      C
0006        XM=X+XD*WF*(XND-1.0)/XND
      C
      C     ESTIMATE VARIANCE.
      C
0007        IF(ND.LE.1) RETURN
0009        VAR=XD**2+(XND-2.0)*(VAR-XD**2)*WF/(XND-1.0)
      C
      C     ESTIMATE DEVIATION.
      C
0010        DEV=SQRT(VAR)
      C
0011        RETURN
0012        END
```

```
C-------------------------------------------------------------------------------
C     NAME:STAFST                                                    NUMBER:
C     -----                                                          --------
C
C     SUBTITLE: START-UP ROUTINE TO THE ROUTINE STAF
C     ----------
C
C     LANGUAGE: FORTRAN IV
C     ----------
C
C     KEYWORDS: START-UP
C     ----------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                         DATE:1977-07-0
C     -------------------                                            -----
C
C     INSTITUTE:
C     ----------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                VERSION:
C     ---------                                --------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE AND METHOD
C     ==================
C
C     THE ROUTINE:
C
C     1. ALLOCATES SPACE IN THE PARAMETER AREA
C     2. SETS THE ADDRESS-PART IN THE NODE HEAD, ADDRESSES
C        TO THE PARAMETER AREA
C     3. ASKS FOR AND SETS SOME PARAMETER VALUES
C
C     USAGE:
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     -------------
C
C     ARGUMENTS:
C     ----------
C
C     STAFST(LIST,IL,IPA,IW)
C
C     IL  -POINTER TO THE FIRST WORD IN THE ACTUAL NODE, (I)
C     IPA -VECTOR CONTAINING THE PARAMETERS IN BACKGROUND JOB, (I)
C     IW  -WORK AREA
C
C     CHARACTERISTICS:
C     ================
C
```

```
      C     SUBROUTINES REQUIRED:  CV,RS
      C     ---------------------------
      C
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 175 WORDS
      C
      C
      C-----------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE STAFST(LIST,IL,IPA,IW)
     CL
0002        DIMENSION LIST(1),IPA(1),IW(1)
      C
      C     ALLOCATE SPACE IN VECTOR IPA FOR 11 WORDS.
      C
0003        CALL RS(IL,11,LIST,IW)
      C
      C     CALCULATE ADDRESSES AND STORE THEM IN THE NODE.
      C
0004        K1=LIST(IL+3)
0005        LIST(IL+4)=K1+2
0006        LIST(IL+5)=K1+4
0007        LIST(IL+6)=K1+6
0008        K5=LIST(IL+7)
      C
0009        DO 15 I=K1,K5
0010     15 IPA(I)=0
      C
0011        WRITE(7,500)
0012    500 FORMAT(1X,$,'WEIGHTING FACTOR:')
0013        READ(5,20) XLA
0014     20 FORMAT(F15.0)
0015        CALL CV(IPA(K1+6),XLA)
      C
0016        WRITE(7,501)
0017    501 FORMAT(1X,$,'WEIGHTING CHANGE FACTOR:')
0018        READ(5,20) XLA0
0019        CALL CV(IPA(K5-2),XLA0)
      C
0020        RETURN
0021        END
```

```
C-------------------------------------------------------------------------------
C
C     NAME: STEP                                              NUMBER:
C     ----                                                    ------
C
C     SUBTITLE: GENERATES A STEP SIGNAL
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C     KEYWORDS: STEP
C     ---------
C
C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON      DATE:1977-07-05
C     ------------                                ----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
C
C     ACCEPTED:                                           VERSION:
C     --------                                            -------
C
C-------------------------------------------------------------------------------
C
C     PURPOSE: GENERATE A STEP SIGNAL
C     =======
C
C     USAGE:
C     =====
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C     ARGUMENTS:
C     ---------
C
C     STEP(X,ST,Y)
C
C     X     -STATE, (I)
C     ST    -ST, (I)
C     Y     -GENERATED SIGNAL, (O)
C
C     METHOD:
C     ======
C
C     Y=X+ST
C
C     CHARACTERISTICS:
C     ===============
C
C
C     SIZE:
```

```
      C     ------
      C     PDP11/03: 16 WORDS
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE STEP(X,ST,Y)
      C
0002        X=X+ST
0003        Y=X
0004        RETURN
0005        END
```

```
C-----------------------------------------------------------------------
C     NAME: TCMD                                              NUMBER:
C     ----                                                    ------
C
C     SUBTITLE: TESTS IF RETURNED IS PUNSCHED
C     ---------
C
C     LANGUAGE: FORTRAN IV
C     ---------
C
C
C     IMPLEMENTOR: LEIF RADING, TYKE PAULSSON                 DATE: 1977-07-0
C     -----------                                             ----
C
C     INSTITUTE:
C     ---------
C     DEPARTMENT OF AUTOMATIC CONTROL
C     LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C     ACCEPTED:                                               VERSION:
C     --------                                                -------
C
C-----------------------------------------------------------------------
C
C     PURPOSE AND METHOD
C     ==================
C
C     THE ROUTINE TESTS IF THE USER HAS PUNSCHED RETURN.
C     IF SO, THE ROUTINE CALLS ON THE OPERATOR
C     COMMUNICATION ROUTINE OPCOM.
C
C     USAGE
C     =====
C
C     PROGRAM TYPE: SUBROUTINE
C     ------------
C
C
C     CHARACTERISTICS
C     ===============
C
C     SUBROUTINES REQUIRED:  OPCOM
C     --------------------
C     FROM SYSTEM LIBRARY: IPOKE
C
C     SIZE:
C     ----
C     PDP11/03: 40 WORDS
C
C
C-----------------------------------------------------------------------
C
C
C
```

```
      C     PROGRAM:
      C     =======
      C
0001        SUBROUTINE TCMD
      C
      C     SET BIT 6 IN JOB STATUS WORD WITHOUT ZEROING ANY OTHER BITS.
      C
0002        CALL IPOKE('44,'100.OR.IPEEK('44))
      C
      C    "RETURN CHARACTER" IN BUFFER?
      C
0003        IF(ITTINR().NE.13) RETURN
      C
      C     CALL OPERATOR COMMUNICATION ROUTINE.
      C
0005        CALL OPCOM
      C
0006        RETURN
0007        END
```

```
C------------------------------------------------------------------------------
C
C       NAME: TECK
C       ----
C
C       SUBTITLE: IDENTIFY A CHARACTER-STRING
C       ----------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ----------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 7-SEP-77
C       -------------------                              -----
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ------------------
C
C       A STRING IS DIVIDED INTO SUBPARTS, SEPARATED BY IT'S COMMAS
C       OR COLONS. AFTER THE IDENTIFICATION EACH SUBPART CORRESPONDS
C       TO THREE INTEGER NUMBERS IN THE RESULT VECTOR. THE FIRST
C       INTEGER IS THE PLACE OF THE COMMAND NAME IN THE COMMAND NAME-
C       ARRAY, THE SECOND IS THE FIRST INTEGER NUMBER IN THE SUBPART
C       (IF THERE IS ANY), THE THIRD IS THE INTEGER NUMBER AFTER A SLASH
C       (IF THERE IS ANY). THE FIRST ELEMENT OF THE RESULT VECTOR IS
C       -1 IF THE COMMANDS NAME ARE NOT ALL IDENTIFIED, ELSE EQUAL TO THE
C       INTEGER NUMBER OF SUBPARTS.
C
C       USAGE
C       =====
C
C       ARGUMENTS:
C       ----------
C
C       TECK(IV3,IV2,L1,L2,LO,N)
C
C       IV3         -VECTOR CONTAINING THE CHARACTER STRING, (I/O)
C       IV2         -VECTOR CONTAINING THE RESULT OF THE IDENTIFICATION (O
C       L1          -POINTER TO THE FIRST COMMAND NAME (I)
C       L2          -POINTER TO THE LAST COMMAND NAME (I)
C                         4   4-CHARACTER  NAME
C       LO          =
C                         2   2-CHARACTER  NAME
C       N          -INPUT STRING LENGHT (I)
C
C
C       CHARACTERISTICS:
C       ================
```

```
C
C       SIZE:
C       -----
C       PDP11/03: 301 WORDS
C
C
0001          SUBROUTINE TECK(IV3,IV2,L1,L2,LO,N)
C
0002          LOGICAL*1 ICM,IW4(4),IV3
C
0003          COMMON /CMD/CM1(60),IOP(100)
0004          COMMON /IM/ICM(7)
C
0005          DIMENSION IV2(1),IV3(1)
C
0006          EQUIVALENCE (X,IX,IW4(1))
C
0007          DO 10 I=1,20
0008       10 IV2(I)=0
C
0009          I2=0
C
C       SKIP ALL BLANK CHARACTERS
C
0010          DO 20 I=1,N
0011          IF(IV3(I).EQ.ICM(7))  GO TO 20
0013          I2=I2+1
0014          IV3(I2)=IV3(I)
0015       20 CONTINUE
C
C       MARK THE END OF THE STRING
C
0016          IV3(I2+1)=ICM(1)
0017          IV3(I2+2)=ICM(4)
0018          I1=0
C
C       EACH LOOP IDENTIFIES ONE SUBPART
C
0019          DO 140 I6=2,18,3
0020          ISL=0
0021       25 ID=0
0022          IR=0
0023          IS=0
0024       30 I1=I1+1
C
C       DIGIT?
C
0025       35 IF(IV3(I1).GE.48.AND.IV3(I1).LE.57) GO TO 50
C
C       IF SINGLE-CHARACTER CONSTANT, IDENTIFY WHICH AND JUMP
C
0027          DO 40 I=1,6
0028       40 IF(IV3(I1).EQ.ICM(I)) GO TO (130,70,80,150,130,30) I
C
```

```
      C     COMMAND
      C
0030        DO 100 I=1,LO
0031        IW4(I)=IV3(I1)
0032   100  I1=I1+1
0033        IF(LO.EQ.4) GO TO 110
      C
      C     IDENTIFY A 2-CHARACTERS COMMAND
      C
0035        DO 105 I=L1,L2
0036   105  IF(IX.EQ.IOP(I)) GO TO 120
0038        GO TO 115
      C
      C     IDENTIFY A 4-CHARACTERS COMMAND
      C
0039   110  DO 112 I=L1,L2
0040   112  IF(X.EQ.CM1(I)) GO TO 120
0042   115  IV2(1)=-1
      C
      C     ILLEGAL COMMAND NAME
      C
0043        RETURN
0044   120  IV2(I6)=I
0045        GO TO 35
      C
      C     DIGIT
      C
0046    50  ID=ID+IR
0047        IS=IS*10+IV3(I1)-48
0048        GO TO 30
      C
      C     POINT
      C
0049    70  IR=1
0050        GO TO 30
      C
      C     SLASH
      C
0051    80  ISL=1
0052        IV2(I6+1)=IS
0053        GO TO 25
      C
      C     COMMA,COLON
      C
0054   130  IV2(I6+2)=ID
0055        IV2(I6+ISL+1)=IS
0056   140  CONTINUE
0057   150  IV2(1)=(I6-2)/3
0058        IF(I2.EQ.0)IV2(1)=0
0060        RETURN
0061        END
```

```
      C-----------------------------------------------------------------------
      C
      C       NAME: TFD
      C       ~~~~
      C
      C       SUBTITLE: SETS INPUT SIGNALS
      C       ~~~~~~~~~~~~
      C
      C       LANGUAGE: FORTRAN + RT-11-FORTRAN
      C       ~~~~~~~~~~~~~~~~~
      C
      C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE:12-SEP-77
      C       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      C
      C       INSTITUTE:
      C       ~~~~~~~~~~
      C       DEPARTMENT OF AUTOMATIC CONTROL
      C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
      C
      C-----------------------------------------------------------------------
      C
      C       PURPOSE AND METHOD
      C       ==================
      C
      C       ASKS FOR INPUT SIGNALS AND SETS POINTERS.
      C
      C       USAGE
      C       =====
      C
      C       TFD(CM1,LIST,IL,IW1,IW2)
      C
      C       CM1      -ARRAY CONTAINING THE COMMAND NAMES, (I)
      C       LIST     -ARRAY CONTAINING THE NODE HEADS, (I)
      C       IL       -NODE NUMBER,(I)
      C       IW1,IW2 -WORK AREA
      C
      C       CHARACTERISTICS
      C       ===============
      C
      C       SIZE:
      C       ~~~~~
      C       PDP11/03: 310 WORDS
      C
      C       SUBROURINE REQUIRED: TECK
      C       ~~~~~~~~~~~~~~~~~~~~~~~~~
      C
      C-----------------------------------------------------------------------
      C
0001          SUBROUTINE TFD(CM1,LIST,IL,IW1,IW2)
      C
0002          LOGICAL*1 IW1
      C
0003          DIMENSION CM1(1),LIST(1),IW1(1),IW2(1)
      C
```

```
0004        5 WRITE(7,500)
0005      500 FORMAT(1X,$,'INPUT SIGNAL:')
0006          READ(5,10)(IW1(I),I=1,40)
0007       10 FORMAT(40A1)
      C
      C       IDENTIFY THE ANSWER
      C
0008          CALL TECK(IW1,IW2,21,40,4,40)
0009          J=IW2(1)
      C
      C       MAXIMUM 4 INPUT SIGNALS
      C
0010          IF(J.GE.0.AND.J.LE.4) GO TO 20
0012       15 WRITE(7,501)
0013      501 FORMAT(1X,'?ILL CMD?')
0014          GO TO 5
      C
0015       20 IF(J.EQ.0) RETURN
      C
      C       TESTS ONE INPUT SIGNAL AT EACH EXECUTION OF THE LOOP,
      C       GIVES ERROR MESSAGE.
      C
0017          DO 80 I=1,J
      C
      C       NODE TYPE,NODE NUMBER AND NUMBER OF A NODE 'S OUTPUT SIGNAL
      C
0018          J1=IW2(I*3-1)
0019          J2=IW2(I*3)
0020          J3=IW2(I*3+1)
0021          IF(J3.LT.0.OR.J3.GT.4) GO TO 15
      C
      C       MEASUREMENT INPUT SIGNAL
      C
0023       40 IF(J1.LT.39) GO TO 50
0025          J4=1
0026          IF(J1.EQ.40) J4=0
      C
      C       SETS A POINTER TO THE LOCATION OF THE INPUT SIGNAL IN THE
      C         PARAMETER ARRAY
      C
0028          LIST(IL+I+7)=J4*16+J2*2+1
0029          LIST(IL)=0
0030          GO TO 80
      C
      C       INPUT SIGNAL FROM OTHER NODE
      C
0031       50 DO 60 K=1,120,12
      C
      C       IS THE NODE, FROM WHICH THE INPUT SIGNAL IS FETCHED, ACTIVE?
      C
0032          IF(LIST(K).LT.0) GO TO 60
0034          IF(LIST(K+2).EQ.1000*(J1-20)+J2) GO TO 70
0036       60 CONTINUE
      C
```

```
      C     SETS A POINTER TO THE LOCATION OF THE INPUT SIGNAL IN THE
      C       PARAMETER ARRAY
      C
      C
0037          WRITE(7,502) CM1(J1),J2
0038      502 FORMAT(1X,A4,I1,': NOT ACTIVE')
0039          GO TO 5
0040       70 LIST(IL+I+7)=LIST(K+J3+3)
0041          LIST(IL)=0
0042          IF(LIST(IL+1).EQ.LIST(K+1)) LIST(IL)=LIST(K)
0044       80 CONTINUE
      C
0045          RETURN
0046          END
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 18:01:26                PAGE 001

      C-----------------------------------------------------------------------
      C     NAME: XWRI                                              NUMBER:
      C     ------                                                  --------
      C
      C     SUBTITLE: WRITE
      C     ----------
      C
      C     LANGUAGE: FORTRAN IV
      C     ----------
      C
      C     KEYWORDS: WRITE
      C     ----------
      C
      C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                  DATE:1977-07-0
      C     -----------------                                       -----
      C
      C     INSTITUTE:
      C     ----------
      C     DEPARTMENT OF AUTOMATIC CONTROL
      C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
      C
      C     ACCEPTED:                                VERSION:
      C     ---------                                --------
      C
      C-----------------------------------------------------------------------
      C
      C     PURPOSE
      C     =======
      C
      C     WRITES REAL VALUES, MAXIMUM 4, ON LINE PRINTER
      C     OR CONSOLE
      C
      C     USAGE
      C     =====
      C
      C     PROGRAM TYPE: SUBROUTINE
      C     ----------------
      C
      C     ARGUMENTS:
      C     ----------
      C
      C     XWRI(X1,X2,X3,X4,IN,CM1,LUN,IT,W1)
      C
      C     X1   -INPUT VALUE 1, (I)
      C     X2   -INPUT VALUE 2, (I)
      C     X3   -INPUT VALUE 3,(I)
      C     X4   -INPUT VALUE 4, (I)
      C     IN   -ARRAY CONTAINING INFORMATION ABOUT NUMBER OF SIGNALS
      C           AND NAMES, (I)
      C     CM1 -ARRAY CONTAINING PARAMETER NAMES, (I)
      C     LUN          -LOGICAL UNIT, (I)
      C     IT   -INTERNAL TIME, (I)
      C     W1   -WORK AREA
      C
```

```
      C
      C     CHARACTERISTICS
      C     ===============
      C
      C     SUBROUTINES REQUIRED:
      C     ---------------------
      C
      C     FROM SYSTEM LIBRARY:  TIMASC
      C
      C     SIZE:
      C     -----
      C     PDP11/03:  140 WORDS
      C
      C
      C-------------------------------------------------------------------------------
      C     PROGRAM
      C     =======
      C
0001        SUBROUTINE XWRI(X1,X2,X3,X4,IN,CM1,LUN,IT,W1)
0002        DIMENSION IN(1),CM1(1),W1(1),IT(1)
      C
      C     CONVERT THE 2-WORD INTERNAL FORMAT TIME INTO AN ASCII STRING.
      C
0003        CALL TIMASC(IT,W1)
      C
0004        WRITE(LUN,20)(W1(I),I=1,2)
0005     20 FORMAT(1H0,'TIME:',2A4/)
      C
0006        W1(1)=X1
0007        W1(2)=X2
0008        W1(3)=X3
0009        W1(4)=X4
0010        I1=IN(1)*3+1
0011        J=0
0012        DO 5 I=2,I1,3
0013        J=J+1
0014        I2=IN(I)
0015      5 WRITE(LUN,10) CM1(I2),IN(I+1),IN(I+2),W1(J)
0016     10 FORMAT(1X,A4,I1,'/',I1,':',E15.5)
      C
0017        RETURN
0018        END
```

```
FORTRAN IV      V01C-03F+  WED 09-NOV-77 18:02:22                PAGE 001

      C-------------------------------------------------------------------------------
      C     NAME:WRIST                                                NUMBER:
      C     ~~~~~                                                     ~~~~~~~~~~
      C
      C     SUBTITLE: START-UP ROUTINE TO  ROUTINE XWRI AND IWRI
      C     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      C
      C     LANGUAGE: FORTRAN IV
      C     ~~~~~~~~~~~~~~~~~~~~
      C
      C     KEYWORDS: START-UP
      C     ~~~~~~~~~~~~~~~~~~
      C
      C     IMPLEMENTOR: LEIF RADING,TYKE PAULSSON                    DATE:1977-07-0
      C     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~                    ~~~~~~~~~~~~~~
      C
      C     INSTITUTE:
      C     ~~~~~~~~~~
      C     DEPARTMENT OF AUTOMATIC CONTROL
      C     LUND INSTITUTE OF TECHNOLOGY,SWEDEN
      C
      C     ACCEPTED:                                                 VERSION:
      C     ~~~~~~~~~                                                 ~~~~~~~~~~
      C
      C-------------------------------------------------------------------------------
      C
      C     PURPOSE AND METHOD
      C     ==================
      C
      C     THE ROUTINE:
      C
      C     1. ALLOCATES SPACE IN THE PARAMETER AREA
      C     2. SETS THE ADDRESS-PART IN THE NODE HEAD, ADDRESSES
      C        TO THE PARAMETER AREA
      C     3. ASKS FOR AND SETS SOME PARAMETER VALUES
      C
      C     USAGE:
      C     =====
      C
      C     PROGRAM TYPE: SUBROUTINE
      C     ~~~~~~~~~~~~~~~~~~~~~~~~
      C
      C     ARGUMENTS:
      C     ~~~~~~~~~~
      C
      C     WRIST(LIST,IL,IPA,IW)
      C
      C     IL  -POINTER TO THE FIRST WORD IN THE ACTUAL NODE, (I)
      C     IPA -VECTOR CONTAINING THE PARAMETERS IN BACKGROUND JOB, (I)
      C     IW  -WORK AREA
      C
      C     CHARACTERISTICS:
      C     ================
      C
```

```
      C     SUBROUTINES REQUIRED:  LU,RS
      C     ---------------------------
      C
      C
      C     SIZE:
      C     -----
      C     PDP11/03: 90 WORDS
      C
      C
      C-----------------------------------------------------------------------
      C     PROGRAM:
      C     ========
      C
0001        SUBROUTINE WRIST(LIST,IL,IPA,IW1,IW2)
0002        DIMENSION LIST(1),IPA(1),IW1(1),IW2(1)
      C
      C     CALCULATE NUMBER OF WORDS TO BE ALLOCATED FOR THE ROUTINE.
      C
0003        NW=IW2(1)*3+2
      C
      C     ALLOCATE SPACE IN VECTOR IPA FOR NW WORDS.
      C
0004        CALL RS(IL,NW,LIST,IW1)
      C
      C     ENOUGHT FREE SPACE?
      C
0005        IF(IL.LT.0) RETURN
      C
      C     GET ADDRESSES.
      C
0007        K1=LIST(IL+3)
0008        K5=LIST(IL+7)
      C
0009        J=0
0010        DO 10 I=K1,K5
0011        J=J+1
0012     10 IPA(I)=IW2(J)
      C
      C     WHICH LOGICAL UNIT?
      C
0013        CALL LU(IPA(K5))
      C
0014        RETURN
0015        END
```

```
C-------------------------------------------------------------------------------
C
C       NAME: WW
C       ------
C
C       SUBTITLE: WRITE
C       ---------------
C
C       LANGUAGE: FORTRAN + RT-11-FORTRAN
C       ---------------------------------
C
C       IMPLEMENTOR: LEIF RADING, TYKE PAULSSON          DATE: 15-SEP-77
C       ---------------------------------------          -------
C
C       INSTITUTE:
C       ----------
C       DEPARTMENT OF AUTOMATIC CONTROL
C       LUND INSTITUTE OF TECHNOLOGY, SWEDEN
C
C-------------------------------------------------------------------------------
C
C       PURPOSE AND METHOD
C       ==================
C
C       WRITE REAL NUMBERS
C
C       COMMENT:
C       --------
C       CAN BE USED TO WRITE AN INTEGER-ARRAY STORING REAL*4 NUMBERS.
C       A REAL NUMBER IS STORED IN TWO CONTINOUS INTEGER*2 NUMBERS.
C
C       USAGE
C       =====
C
C       ARGUMENTS:
C       ----------
C
C       WW(X,N)
C
C       X         -ARRAY CONTAINING THE VALUES TO BE WRITTEN, (I)
C       N         -NUMBER OF REAL NUMBERS TO BE WRITTEN, (I)
C
C       CHARACTERISTICS
C       ===============
C
C       SIZE:
C       -----
C       PDP11/03: 33 WORDS
C
C-------------------------------------------------------------------------------
C
0001          SUBROUTINE WW(X,N)
C
0002          DIMENSION X(1)
```

```
      C
0003        WRITE(7,10)(X(I),I=1,N)
0004     10 FORMAT(1X,5E14.6)
      C
0005        RETURN
0006        END
```