

**PROCEDURE DOS PER L'USO DEL
RADIOTELESCOPIO "CROCE DEL NORD"**

**C. Bortolotti - A. Cattani
G. Grueff - A. Maccaferri
S. Montebugnoli**

Collaborazione grafica di V. Albertazzi

Rapporto interno IRA 161/92

word file: proced.doc

INDICE

- INTRODUZIONE
- CALIBRA.BAT
- P.BAT
- SORCAL.BAT
- OSSERVA.BAT
- APPENDICE

INTRODUZIONE

Nel presente rapporto interno viene descritta una serie di procedure DOS e relativi programmi per il computer XT-286, che rendono piu` agevole l'uso del Radiotelescopio "Croce del Nord" e del sistema di acquisizione dati. Oltre che rappresentare una dettagliata descrizione del software, questo rapporto interno vuole essere una guida operativa.

L'attuale configurazione della Croce, appare in fig.1. Tale schema evidenzia il fatto che il PC XT-286 di gestione Croce, puo` interagire con i blocchi che controllano il puntamento e l'acquisizione dati. Il PC XT-286 ha il completo controllo dei blocchi (marche interne, messa su resistenza dell'antenna, blocco inserzione 1 dB in catena ecc..) che permettono il test delle catene, e di tutti i blocchi che costituiscono l'acquisizione dati, inoltre puo` mandare la richiesta di puntamento, via RS232, al PC AT di puntamento che ha il compito, a sua volta, di puntare meccanicamente ed elettricamente la Croce. Questa disposizione rende possibile il funzionamento automatico del radiotelescopio. Per espletare questo compito sono stati scritti alcuni programmi di acquisizione (TOS, TOSKED), di calibrazione e test delle catene del telescopio (CORR, CAL, WIDECAL) e di interfaccia con il sistema di puntamento (POINT, PCPOINT). La possibilita` di raggruppare questi blocchi software all'interno di procedure DOS, rende automatiche le fasi di calibrazione dello strumento e le osservazioni. Nei paragrafi che seguono vengono riportate le descrizioni delle procedure ora in uso e dei programmi che le compongono, si precisa che dette procedure possono essere facilmente modificate in base alle esigenze del caso, e quindi non sono da considerarsi standard ma esempi di come operare.

- CALIBRA.BAT -

Il procedura -CALIBRA.BAT-, con il nome del calibratore come input, gestisce tutte le operazioni di puntamento, acquisizione dati e successive elaborazioni, per la calibrazione della Croce. Tutti i dati relativi al calibratore richiesto (1950), devono essere presenti nel source data base \sdb\XXX.cal (XXX = nome sorgente).

Lo schema a blocchi di CALIBRA.BAT e` visibile in fig.2 in cui ogni blocco e` costituito da un programma, il cui listato apparira` in appendice. Una volta lanciata la procedura, viene chiesto dal blocco STAMP di accendere la stampante e dal blocco NASTRO il numero del nastro, su cui verra` registrato il calibratore che viene memorizzato nel file \croce\nast.dat; questo blocco da` la possibilita` di posizionare manualmente il nastro dove si vuole, oppure di posizionarlo automaticamente alla fine dell'ultimo file scritto. Nella successiva fase viene chiesto dal blocco LETDB, il nome del calibratore da osservare; tramite un 'inquire' si controlla se e` presente nella lista sorgenti \sdb\XXX.cal (XXX = nome sorgente) in ognuno di questi file i dati sono disposti nel seguente formato:

3c123	(Nome sorgente di calibrazione max 5 chr)
04 33 55.40	(Ascensione retta al 1950.0)
29 34 14.	(Declinazione al 1950.0)
120.	(Flusso in unita` di flusso)

Se la risposta e` affermativa, i relativi dati vengono letti e parcheggiati nel file \croce\dec50.dat assieme alla data letta dall'orologio di stazione; in caso contrario il programma chiede i dati nello stesso ordine riportato sopra, ed oltre a scriverlo nello stesso file \croce\dec50.cal, lo scrive anche nella lista delle sorgenti \sdb\XXX.cal in modo da ampliarne sempre piu` il set contenuto. Il blocco PRE, (G. Grueff), ha il compito di leggere le coordinate al 1950 della sorgente e la data dal file \croce\dec50.dat, e di effettuare la precessione a coordinate apparenti alla data corrente (precessione, nutazione, aberrazione); le coordinate calcolate alla data corrente, vengono quindi parcheggiate nel file \croce\decnew.dat. Il file \croce\decnew.dat viene letto successivamente dal programma NEWSK che ha il compito di comporre la schedula contenente tutti i dati correnti, relativi al puntamento ed acquisizione del calibratore richiesto. Una volta letta l'ascensione retta, viene calcolata l'ora di start delle operazioni di puntamento (Alfa-27 min.), di start acquisizione (Alfa-15 min) e di stop acquisizione (Ora di Start Acq. +30 min.). La

schedula operativa presente ora in \croce\newsked.dat, ha il seguente formato:

start 04 15	(Ora di start operazioni di puntamento)
3c123	(Nome sorgente di calibrazione max 5 chr)
04 36 35.643	(Ascensione retta precessata)
29 39 24.56	(Declinazione precessata)
120.	(Flusso in unita` di flusso)
04 21	(Ora di start acquisizione)
04 51	(Ora di stop acquisizione)
stop	

A questo punto la procedura lancia il programma SIDTIME derivato da SIDER (G. Grueff), che legge data, ora UT e Siderale dagli orologi di stazione e calcola l'errore dell'orologio siderale riferito all'orologio UT; questo viene scritto nel file \croce\sider.err assieme alla data; successivamente il blocco CORR1, ha il compito di controllare il banco di correlatori (ampiezza, fase, isolamento, fondo, fondo in quadratura ed errore quadratico medio) e di memorizzare il set di risultati (in questo caso solo ampiezza, fasi ed isolamento) nei files \croce\calib\corr.day (disco rigido) e in b:corr.day (Log-Disk), day = numero giorni trascorsi dall'inizio dell'anno. Il programma di acquisizione sotto schedula TOSKED che segue (descritto in un altro rapporto interno), dopo avere letto il numero del nastro da \croce\nast.dat, i correlatori da \croce\calib\corr.day, errore siderale e data da \croce\sider.err ed i rimanenti dati per l'header dal newsked.dat, prende il controllo dell'acquisizione dati. Al termine della acquisizione viene lanciato il programma CAL che fornisce come output i dati per le correzioni di ampiezza e fase, da apportare ai canali singoli EW e NS per la relativa calibrazione di ampiezza e fase.

-CALIBRA. BAT-

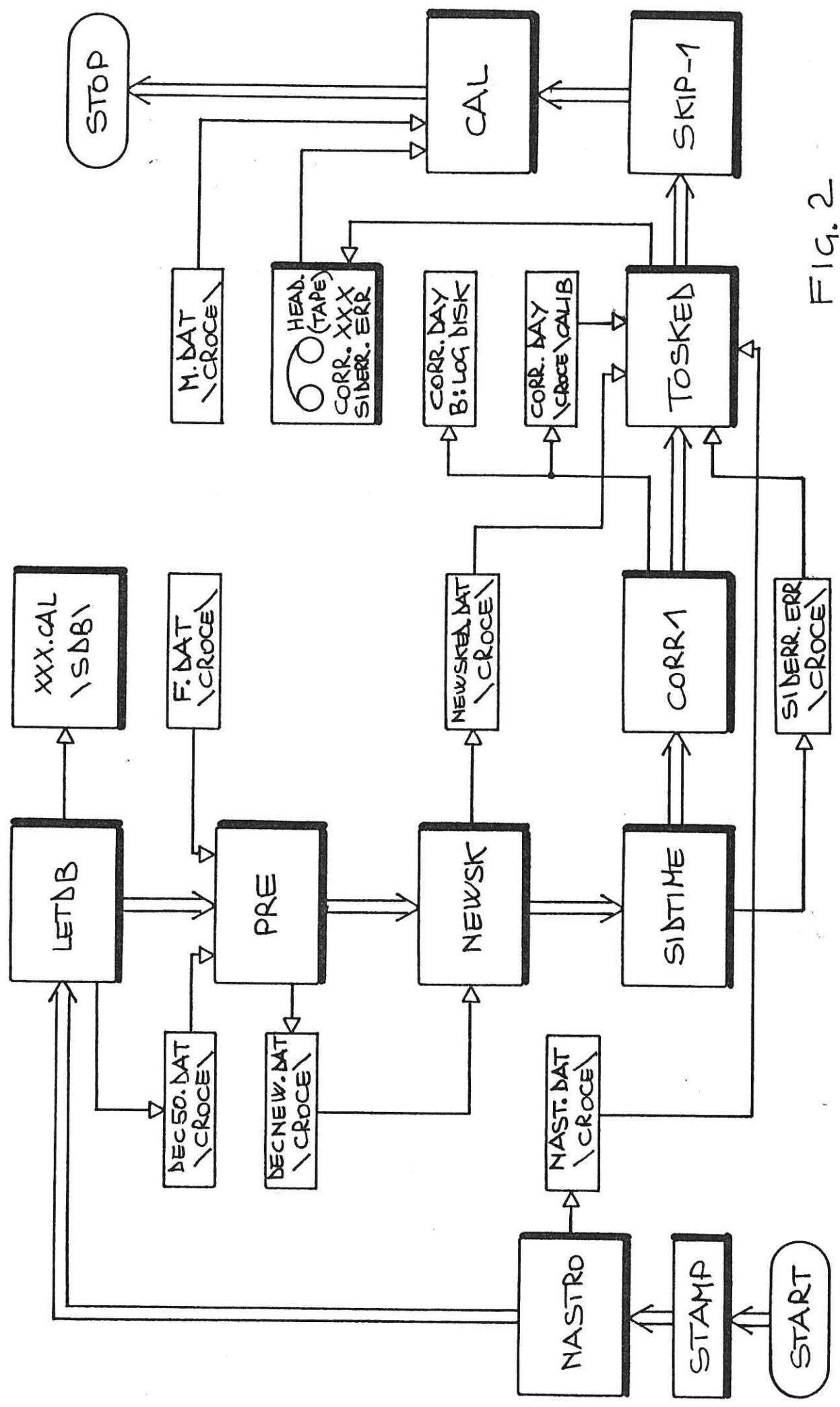


FIG. 2

```
:-----  
:CALIBRAZIONE AUTOMATICA DELLA CROCE  
:-----  
:La stampante deve essere ACCESA.  
stamp  
:numero nastro e Posizionamento  
nastro  
:input nome Sorgente. Viene messo in (\croce\sname.dat)  
:input dati Sorgente da Data-Base (1950) (\sdb\SSS.cal)  
:vengono lette Alfa e Delta e messe in (\croce\dec50.dat )  
letdb  
:calcolo della nuova posizione (alla data att.) e  
:scrittura in (\croce\decnow.dat) della nuova posizione  
pre <\croce\f.dat  
:scrittura della nuova schedula in (\croce\newsked.dat)  
newsk  
:controllo errore siderale (\croce\sider.err)  
sidtime  
:controllo correlatori (log disk corr.day)  
corr1 p  
:chiamata di tosked con la schedula (\croce\newsked.dat)  
tosked \croce\newsked.dat < \croce\nast.dat  
:posizionamento del nastro per il cal  
skip -1  
cal p <\croce\h.dat
```


-P.BAT-

Il programma -P.BAT- punta EW e/o NS (solo cielo Nord), su un calibratore di cui si e` dato il nome come input. I dati relativi devono essere presenti nella lista delle sorgenti \sdb\XXX.cal (XXX = nome sorgente).

Questa procedura puo essere molto utile e comoda in fase di prova e/o calibrazione delle due antenne, EW e NS. Digitando semplicemente 'P', viene chiesto quale ramo puntare e su quale sorgente. Lo schema a blocchi del programma P.BAT, appare in fig.2. All'inizio della procedura viene chiesto dal programma PENT, se si vuole puntare meccanicamente il ramo EW (E) e/o meccanicamente ed elettricamente il ramo NS (N); in funzione della risposta data, viene formato il relativo 'vettore' da passare a PCPOINT, e viene scritto nel file \croce\pword.dat. Il nome della sorgente su cui si vuole puntare il ramo scelto, viene chiesto dal modulo LETDB; questo e` lo stesso modulo visto nella precedente procedura, che ha il compito di estrarre dal data base delle sorgenti (\sdb\XXX.cal, con XXX = nome sorgente) i dati relativi con posizione al 1950, e parcheggiarli nel file \croce\dec50.dat. Da qui i dati vengono letti dal programma di precessione PRECES e memorizzati nel file \croce\decnew.dat. Il blocco responsabile del puntamento meccanico ed elettrico dei due rami, e` POINT che, trovando una F nel file \croce\f.dat legge la declinazione dal file \croce\decnew.dat e quali rami puntare dal file \croce\pword.dat (se invece trovasse una K chiederebbe la declinazione e quali rami puntare da tastiera).

Anche in questo caso il programma LETDB controlla se esiste nel data set delle sorgenti, la sorgente richiesta, in caso contrario chiede i dati con le stesse modalita` viste in precedenza, andando cosi` ad ampliare il contenuto dello stesso data set. Considerando che le sorgenti che servono per questi scopi non sono molte, in un lasso ragionevole di tempo si dovrebbe avere il data set completo di tutte le sorgenti 'utili' per le calibrazioni di fase ed ampiezza; in questo modo le operazioni di test e calibrazione saranno piu` rapide e, soprattutto, senza possibilita` di errori.

Nella pagina seguente viene riportato il listato del suddetto programma.

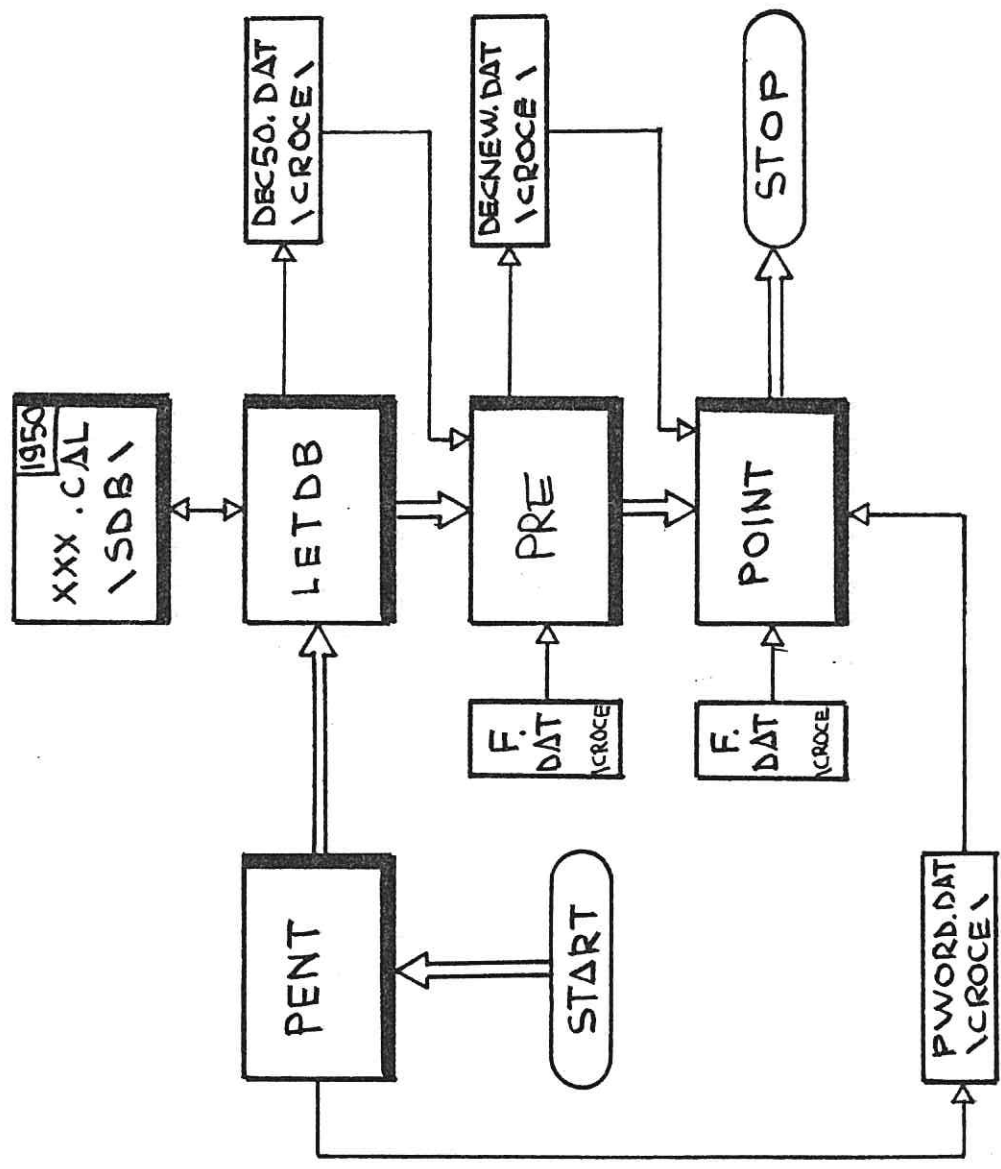


FIG. 3

```
:-----  
: PROGRAMMA DI PUNTAMENT ANTENNE -EW- E -NS-  
:-----  
: Scelta della antenna da puntare ('parola' di puntam. \croce\pword.dat)  
pent  
: Il Nome Sorgente deve essere nel Data-Base (\croce\nomcal.dat)  
: Lettura dei datai dal Data-Base (R.A. Dec. al 50 in \croce\dec50.dat)  
letdb \croce\sname.dat  
: Precessione (R.A. dec. alla data attuale, in \croce\decnew.dat)  
pre <\croce\f.dat  
: Puntamento antenna scelta (delta letta da \croce\decnew.dat)  
point <\croce\f.dat
```

-SORCAL.BAT-

Il programma -SORCAL.BAT- esegue in successione dopo aver fatto un rewind, i CAL di tutte le sorgenti presenti in un nastro.

Il programma puo` essere utile nelle situazioni in cui avendo osservato una serie di sorgenti, si voglia avere fase ed ampiezza dei canali singoli, per ogni calibratore, ottenuti con CAL. Lo schema a blocchi della procedura appare in fig.4. Alla partenza viene eseguito il programma di controllo del banco di correlatori CORR1, necessario per fornire al programma CAL il relativo set di dati (ampiezza, fasi ed isolamento), viene riavvolto il nastro al BOT con REWIND, si calcola l'errore dell'orologio siderale con SIDTIME e si esegue CAL con i dati relativi alle sorgenti, presi da nastro (headers). Al termine dell'esecuzione del programma CAL, viene fatto uno skip in avanti con SKIP 1, per posizionarsi subito dopo il successivo file-mark ed essere cosi` pronti per il prossimo CAL. Si ripete la stessa procedura, fino all'esaurimento di tutte le sorgenti presenti sul nastro. Un aspetto poco 'elegante' di questa procedura, e` dato dal fatto che dopo l'elaborazione dell'ultima sorgente, occorra uscire con un ctrl-C, infatti non c'e un controllo di fine nastro registrato. Il listato della procedura, viene riportato alla pagina seguente.

- SORCAL. BAT -

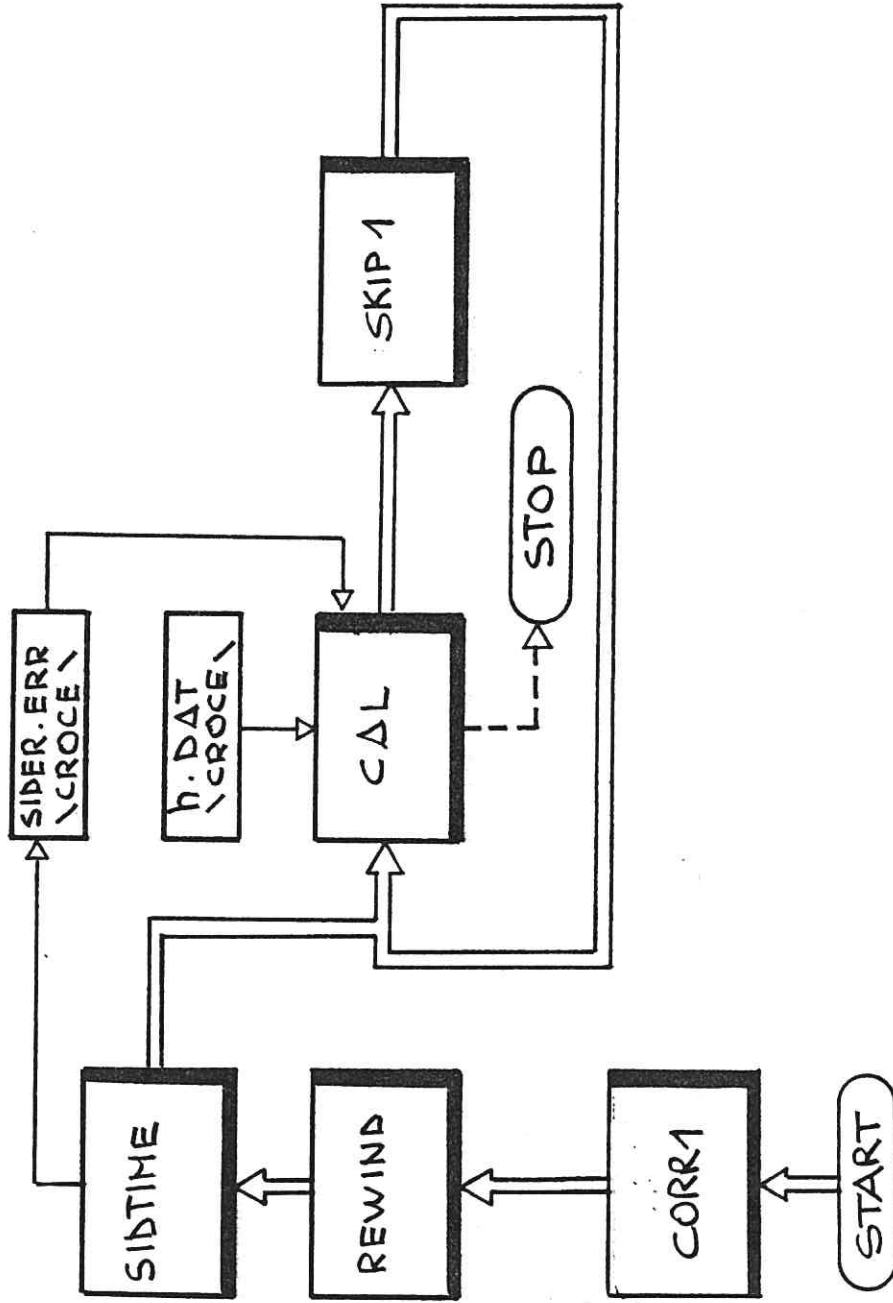


FIG. 4

```
:-----  
:PROCEDURA DI CALIBRAZIONE SU DIVERSE SORGENTI  
:-----  
corr1  
rewind  
sidtime  
:inizio  
cal <\croce\h.dat  
skip 1  
goto inizio
```

-OSSERVA.BAT-

Il programma -OSSERVA.BAT- automatizza tutte le operazioni inerenti le osservazioni nell'ambito della sky survey, anche condotta parallelamente alla millisecond pulsar sky survey.

Nell'ambito del proseguimento della sky survey nel continuo e della nuova millisecond pulsar sky survey, si è deciso di effettuare osservazioni parallele per una migliore ottimizzazione dell'utilizzo dello strumento e dei tempi. Le osservazioni vengono effettuate per una durata di circa 23 ore per ogni strip; si cambia il nastro, si calibra sulla sorgente di calibrazione (3C123 o 3C295 o altro) e su una pulsar di calibrazione (CRAB o altro). La procedura OSSERVA.BAT che segue, ha il compito di automatizzare tutta la sequenza di operazioni richieste, annullando in pratica le possibilità di errori di 'manovra' e facendo in modo che tutte le operazioni vengano eseguite seguendo una cronologia ben definita; questo per evitare qualsiasi tipo di interferenza operativa tra i due programmi. Alla partenza viene riavvolto il nastro al BOT con REWIND ed inviato un reset generale, a tutti i blocchi di acquisizione, comando e controllo, con RESET. Viene poi lanciato il programma PROVE che esegue una misura di livello in unità di 0.01 microwatt (tipo Prova E) dei vari canali singoli EW e NS, i cui risultati vengono scritti sul Log-Disk nel file pre.day. Segue il programma SIDTIME per il controllo dell'errore dell'orologio siderale, errore che viene memorizzato nel file \croce\sider.err; quest'ultimo tiene conto anche della costante di integrazione degli integratori del banco correlatori (1 sec). Il blocco successivo, ATTESA, offre la possibilità all'operatore di cambiare il nastro, guidandolo con opportuni messaggi su schermo. Il blocco NASTRO gestisce la numerazione automatica dei nastri, incrementando di 1 il numero del nastro appena smontato letto dal file \croce\nast.dat, e riscrivendolo nello stesso file. Esiste la possibilità di cambiare il numero del nastro corrente nel caso in cui sia necessario farlo; il nuovo numero viene scritto sempre in \croce\nast.dat pronto per gli usi successivi. Il blocco seguente, DEC, chiede in input la declinazione della nuova strip nel formato gg mm ss.cc ed un commento di massimo 40 caratteri. Le informazioni relative al calibratore 3C123 rimangono invariate per tutta la survey e sono contenute nel file \croce\3c123.sur nel seguente formato:

start 04 15	(Ora di start operazioni di puntamento)
1	(Numero file)
3c123	(Nome sorgente di calibrazione max 5 chr)
04 36 35.643	(Ascensione retta precessata una tantum)
29 39 24.56	(Declinazione precessata una tantum)
calib	(Nome programma)
120.	(Flusso in unita` di flusso)
04 21	(Ora di start acquisizione)
04 51	(Ora di stop acquisizione)
stop	

Le informazioni relative alla pulsar di calibrazione CRAB rimangono invariate per tutta la survey e sono contenute nel file \croce\crab.sur nel seguente formato:

Start 05 03	(Ora di start operazioni di puntamento)
2	(Numero file)
crab	(Nome pulsar di calibrazione max 5 chr)
05 34 04.490	(Ascensione retta precessata una tantum)
22 00 58.66	(Declinazione precessata una tantum)
calib pulsar	(Nome programma)
1150	(Flusso in unita` di flusso)
05 05	(Ora di start acquisizione)
05 45	(Ora di stop acquisizione)
stop	

Le informazioni relative alla strip corrente sono contenute nel file \croce\strip.sur nel seguente formato:

start 05 55	(Ora di start operazioni di puntamento)
3	(Numero file)
strip survey	
06 00 00.	(Ascensione retta)
27 00 00.	(Declinazione della strip in corso)
strip survey	
bel tempo	(Commenti max 40 chr)
06 00	(Ora di start acquisizione)
03 30	(Ora di stop acquisizione)
Stop	

Nel suddetto file le righe relative a 'declinazione della strip in corso' e 'commenti max 40 chr' vengono di giorno in giorno riaggornate. Il commento viene anche scritto nel Log-Disk, file B:comm.day, con la data corrente in modo tale da costituire un Log delle osservazioni, facilmente consultabile ed archiviabile. A questo punto della procedura il programma TOSKED (di cui si parla in un altro rapporto interno) prende il controllo del sistema, e gestisce tutte le operazioni di acquisizione dati relative all'osservazione in corso.

Al termine delle operazioni di acquisizione dei calibratori, il nastro viene riportato sul BOT con il programma REWIND. Nel blocco successivo, viene eseguito il controllo del banco di correlatori tramite CORR1; i risultati delle misure vengono memorizzati nei due files \croce\calib\corr.day e nel Log-Disk in corr.day. La elaborazione dei dati per l'ottenimento dei valori delle ampiezze e delle fasi dei singoli canali, avviene con il modulo CAL; questo prende il set dei dati relativi ai correlatori e l'errore siderale di cui necessita, dai rispettivi files B:corr.day e \croce\sider.err. I risultati del CAL, vengono scritti sul Log-Disk nel file lcal.day e sul disco rigido in log\cal.log. A questo punto si posiziona il nastro per la registrazione della strip corrente eseguendo con i comandi REWIND e SKIP 2. Il controllo dell'acquisizione dati, viene preso ancora da TOSKED che, come visto in precedenza, gestisce l'acquisizione della strip corrente. Alla fine della osservazione della strip, cioè circa 23 ore dopo; il nastro viene riportato al BOT con REWIND. Raggiunto il BOT si ferma pronto per essere smontato e sostituito con uno nuovo, ripartendo così con l'intera procedura che prevede l'esecuzione del primo blocco di RESET, e così via.

Si sottolinea il fatto che durante tutta la durata delle osservazioni delle due survey, il controllo del calcolatore di puntamento AT è tenuto dal calcolatore di gestione del sistema Pulsar. L'incremento o decremento giornaliero della delta della strip di 20 primi, viene calcolato automaticamente da quest'ultimo ed inviato al calcolatore di puntamento. Una volta entrata la delta della nuova strip da tastiera nel calcolatore XT 286 della Croce, al momento opportuno, viene mandata al calcolatore di puntamento che, essendo allocato all'altro, non fa altro che rispondere Master=F e ritorna la delta di puntamento reale. A questo punto della procedura si prevede in futuro di modificare tosked in modo tale da farla proseguire solo se la delta di puntamento reale risulta essere uguale a quella richiesta, altrimenti attenderà (WAIT) per 1 minuto e ritenterà la procedura fino a che non trova che l'altro gestore ha portato l'antenna alla giusta delta. In ogni caso se le schedule sono scritte con un certo criterio, questa situazione non si dovrebbe verificare per cui quando il sistema Croce chiederà di puntare sulla nuova delta, si dovrebbe già trovare l'antenna puntata.

Il listato del programma OSSERVA.BAT, viene riportato alla pagina seguente.


```
:-----  
:PROCEDURA DI OSSERVAZIONE PER SKY SURVEY  
:-----  
rewind  
:inizio  
echo on  
:reset di tutti i rele`  
reset  
:lettura della prova E  
:i dati vanno in 'pre.XXX' nel log-disk (disco B)  
prove  
:controllo dell'errore siderale. Viene messo in -\croce\sider.err-  
sidtime  
be beep  
attesa  
:inserimento NUMERO NASTRO  
nastro  
:inserimento DELTA della -strip- del giorno (messa in \croce\strip.sur)  
dec  
be beep  
:corr1 -day- e -dati- vanno in 'corr.XXX' nel log-disk (B:)  
corr1  
:tosked (input reindirizzato a \croce\nast.dat)  
tosked \croce\3c123.sur < \croce\nast.dat  
:pulsar di calibrazione  
tosked \croce\crab.sur < \croce\nast.dat  
rewind  
:reset di tutti i rele`  
reset  
:i dati di cal vanno nel log-disk (B:)  
cal < \croce\h.dat  
rewind  
skip 2  
:osservazione della STRIP  
tosked \croce\strip.sur < \croce\nast.dat  
rewind  
goto inizio
```

APPENDICE

In Appendice vengono riportati i listati dei moduli di programma usati nelle procedure.bat descritte. Alcuni di questi programmi sono stati derivati da programmi già esistenti ed adattati alle specifiche necessità dei vari '.bat'. È ad esempio, il caso di CORR1 che deriva da CORR. Il programma CORR1 parte automaticamente assumendo di default di controllare con 10 misure, tutto il banco di correlatori (1/96). Il programma CORR, invece, offre la possibilità di definire il blocco di correlatori da controllare e con quante misure; opzione questa, molto utile specialmente in fase di manutenzione degli stessi.

```
call cls  
write(*,*) 'ATTENZIONE: Assicurarsi che la STAMPANTE sia ON.'  
pause  
end
```

\$NOTRUNCATE
\$DECLARE

```
C      =====
C      ***** TAPE MANAGEMENT *****
C      =====
```

```
C      S.Montebugnoli
C      -1991-
```

```
C      -----
C      This program has to be linked with:
C      -----
```

```
C      -TAPEC          --> Tape control routine
C      -FORT488       --> IEEE 488 routine
C      -NOLIMIT.LIB   --> Nolimit libraries
C      -----
```

```
integer*2 address,level,key,num,ind,i,nast,inum,ier
integer*1 b1,b2,b3,b4
integer*2 irow,icolm,ivpage,ierror,ifreq,idurr,in
integer*4 aseq
real      nufi
character*5 rinp
character*2 r

integer*2 ntape,k
character*5 resp
```

```
C      -----
C      PC488 interface initializing
C      address=1
C      level=0
C      call initia (address,level)
C      -----
C      call cls
C      ---- make speaker beep ----
C      ifreq=1500
C      idurr=5
C      call beep2(ifreq,idurr,in)
C      write (*,*) '
C      write (*,15) '-----
C      &-'
C      write (*,15) 'NORTHERN CROSS RADIOTELESCOPE DATA ACQUISITION SYST
C      &EM'
C      write (*,15) '
C      write (*,15) '          AUTOMATIC TAPE POSITIONING PROGRAM
C      &
C      write (*,15) '-----
C      &-'
C      15  format (13x,a53)
C      -----
C      AUTOMATIC TAPE POSITIONING
```

```
c -----
500 write (*,*) '
write (*,*) '
503 write (*,*) 'NEW TAPE? (y/n) '
read (*, '(a1)') rinp
if (rinp.eq.'y'.or.rinp.eq.'Y') go to 1720
if(rinp.eq.'n'.or.rinp.eq.'N')goto 507
goto 503

c ---- OLD TAPE ----

507 call rew(b1,b2)
write(*,*)'Files to Skip (ENTER = End of Data)'
508 read(*, '(a5)') resp
if(resp.eq.' ') goto510

c --- skip files ---
read(resp,'(i3)') inum
do i=1,inum
call fsfwhighsp(b1,b2)
enddo
stop '**** done ****'

c ---- Automatic Tape Positioning ----
c ---- overall error ----
510 call sbytes (b1,b2)
call decode(b1,b2,b3,b4)
c ---- bot? ----
if (b4.eq.16.or.b4.eq.1) go to 1010
c ---- rev 1 block ----
call rv1blk (b1,b2)
c ---- bot or rev. requested at bot? ----
call decode (b1,b2,b3,b4)
if (b4.eq.16.or.b4.eq.1) go to 1010
c call clear screen
call cls
c ---- fm? ----
call decode(b1,b2,b3,b4)
if (b4.eq.8) go to 920
c ---- rev. file search ----
830 call fsrlowsp (b1,b2)
c ---- bot or rev.requested at bot? ----
call decode(b1,b2,b3,b4)
if (b4.eq.16.or.b4.eq.1) go to 1000
c ---- BLANK tape ? ----
if (b3.eq.40.or.b3.eq.52) go to 830
if (b4.eq.8) go to 920
go to 920
c ---- rew ---
840 call rew (b1,b2)
go to 1000
920 continue
c ---- rev 1 block ----
```

```
      call rv1blk (b1,b2)
c      ---- fm? ----
      call decode (b1,b2,b3,b4)
      if (b4.eq.8) go to 960
      go to 970
960    continue
c      ---- fw1blk ----
      call fw1blk (b1,b2)
      go to 1750
970    continue
1000   continue
c      ---- overall error ----
1010   continue
c      ---- fast fw search ----
      call fsfwhighsp(b1,b2)
      call decode (b1,b2,b3,b4)
c      ---- fm? ----
      if (b4.eq.8) go to 1220
c      ---- rev file search ----
1020   call fsrhighsp (b1,b2)
      call decode (b1,b2,b3,b4)

c      ----blank tape ? ----
      if (b3.eq.40) go to 1020

c      ---- fm? ----
      if (b4.eq.8) go to 1200
      go to 1020
1200   continue
c      ---- fw 1 block ----
      call fw1blk (b1,b2)
c      ---- clear screen and beep ----
      call cls
      ifreq=3000
      idurr=5
      call beep2(ifreq,idurr,in)
      write (*,*)' end of recorded tape (file mark), not found!!!!'
      write (*,*)' tape ready to write a new file at the end of last
& written file!!!!'
c      ---- delay loop ----
      do 1210 i=1,200
      ifreq=4500
      idurr=2
      call beep2(ifreq,idurr,in)
1210   continue
      pause
      go to 1750
1220   continue
c      ---- fw 1 block ----
      call fw1blk (b1,b2)
c      ---- fm? ----
      call decode (b1,b2,b3,b4)
      if (b4.eq.8) go to 1230
      go to 1010
```



```
1230  continue
c      ---- rev 1 block ----
      call rv1blk (b1,b2)
      go to 1750
1350  continue
1720  continue
c      ---- NEW TAPE ----
c      ---- rew ----
      ifreq=4000
      idurr=3
      call beep2 (ifreq,idurr,in)
      call rew (b1,b2)
1750  continue

      open(unit=15, file='\croce\nast.dat')
      irow=3
      icolm=1
      ivpage=0
      call cls
      write(*,*) '-- NTAPE LOG-FILE --'

c      rewind (15)

      read(15,16) ntape
      if(rinp.eq.'Y'.or.rinp.eq.'Y')ntape=ntape+1
      write(*,*) '      '
      call putcur(irow,icolm,ivpage,ier)
      write(*,11)ntape
11     format(' New Tape Number = ',i3)

      do 111 k=1,2
      ifreq=1000
      idurr=1
      do 110 i=0,6
      ifreq=ifreq+(i*170)
      idurr=1+(i*6)
      call beep2(ifreq,idurr,in)
110    continue
111    continue

      write(*,*) '      '
16     format(i3)
      write(*,*) 'OK? (y/n)'
      read(*,18) resp
18     format(a5)
      if(resp.eq.'n'.or.resp.eq.'N') go to 400
      go to 600

400    continue
      irow=12
      call putcur(irow,icolm,ivpage,ier)
      write(*,*) 'ENTER NEW TAPE NUMBER'
      write(*,*) '-----'
      write(*,*) '      '
      .
```

```
5      read(*,5)ntape
      format(i3)

      do 556 k=1,2
      ifreq=1000
      idurr=1
      do 555 i=0,6
      ifreq=ifreq+(i*170)
      idurr=1+(i*6)
      call beep2(ifreq,idurr,in)
555    continue
556    continue
600    rewind(15)
      write(15,10)ntape
10     format(i3)

      end
```

```

c      ----- reading source data from data-base Program-----
c              --- for CALIBRA.BAT -----

character*20 sname$
character*17 arg$,arg
character*40 calibr
character*15 info(4)
character*5 sorg
logical flag
integer*2 ifreq, idurr, ierr, k

integer*2 ut1, ut2, ut3, ut4, ut5, ut6, ut7, ut8, ut9, ncs, nchr

c      --reading date from clock --
call clock(ut1, ut2, ut3, ut4, ut5, ut6, ut7, ut8, ut9 )
iear=1900+ut3
mo=ut2
iday=ut1
iear1=1950
mol=1
iday1=1

11     write(*,*) '
        write(*,*) 'Enter Source Name (max 5 char.)'
        write(*,*) '-----'
        read(*, '(a5)') sorg

        sname$=' \sdb\ '//sorg//'.cal'

c      --- Is the entered source present in the Data-Base? ---
inquire(file=sname$, exist=flag)
if(.not.flag) then
write(*,*) sorg, ' Isn`t in the Sources List'
write(*,*) 'Enter R.A. (hh mm ss.ss) 1950.0'
read(*, '(a15)') info(2)
write(*,*) 'Enter Dec. (dd mm ss.ss)'
read (*, '(a15)') info (3)
write(*,*) 'Enter flux (ffff.ff)'
read(*, '(a15)') info(4)

c      -- writing of new source data, on \sdb\XXX.cal--
open (unit=10, file=sname$, status='new')
rewind(10)
write(10,101) sorg
write(10,102) info(2)
write(10,103) info(3)
write(10,104) info(4)
101    format(a5)
102    format(a13)
103    format(a13)
104    format(a10)
close(10)

```

```
goto 22
endif
```

```
write(*,*)sname$
```

```
open(unit=10,file=sname$,status='old')
do k=1,4
read(10,'(a15)') info(k)
write(*,*) info(k)
enddo
close(10)
```

```
22 open(unit=12,file='\croce\dec50.dat')
write(12,105)iear1,mo1,iday1
write(12,105)iear,mo,iday
105 format(1x,i4,2i3.2)
write(12,100) sorg,info(2),info(3),info(4)
100 format(a5,5x,a13,a13,a10)
stop
end
```

\$debug

```

IMPLICIT DOUBLE PRECISION(A-H,O-Z)
character*10 text1,text2
real*8 julian,pi
INTEGER AR1,AR2,DEC1,DEC2,AR1F,AR2F,DEC1F,DEC2F
pi=3.1415926536d0
rad=pi/180.
aberr=20.49
write(*,*)'*** To convert mean position at epoch 1 to ',
+'apparent position at epoch 2 ***'
write(*,*)
write(*,*)'Data input: from keyboard (k); from file (f)'
read(*,'(a1)')resp
if(resp.eq.'F'.or.resp.eq.'f')goto 50
828 WRITE(*,8)
8 FORMAT(1X,'initial epoch? (yyyy mm dd)')
READ(*,*)iear,mo,iday
if(iday.eq.0.or.mo.eq.0.or.mo.gt.12.or.iday.gt.31)then
write(*,*)iear,mo,iday
write(*,*)'*** Wrong format!! *****'
goto 828
endif
ep1=iear
ep1=ep1+mo/100.+iday/10000.d0
838 WRITE(*,10)
10 FORMAT(1X,'final epoch? (yyyy mm dd)')
READ(*,*)iear,mo,iday
if(iday.eq.0.or.mo.eq.0.or.mo.gt.12.or.iday.gt.31)then
write(*,*)'*** Wrong format!! *****'
goto 838
endif
ep2=iear+mo/100.d0+iday/10000.d0
epoch=ep2
t0=(julian(ep1)-2415020.313d0)/36524.2199d0
t=(julian(epoch)-2415020.313d0)/36524.2199d0
eps=23.4523-0.013*t
c write(*,*)eps
eps=eps*rad
omega=259.1833d0-1934.1420d0*t+0.002078d0*t*t
c write(*,*)omega
omega=omega*rad
call sunpos(epoch,aa,dd,slon,iflag)
c write(*,*)epoch,slon
slon=slon*rad
write(*,543)ep1,ep2
543 format(1x,'epoch1 is ',f9.4,3x,'epoch2 is ',f9.4)
1 WRITE(*,2)
2 FORMAT(/1X,'position? (hh mm ss.ss tgg tpp tss.s...) ^C to exit
+')
READ(*,*) AR1,AR2,AR3,dec1,dec2,dec3
goto 51
50 continue
open(unit=10,file='\croce\decnew.dat')
open(unit=11,file='\croce\dec50.dat')

```

```

      READ(11,*) iear,mo,iday
      ep1=iear
      ep1=ep1+mo/100.+iday/10000.d0
      READ(11,*) iear,mo,iday
      ep2=iear+mo/100.d0+iday/10000.d0
      epoch=ep2
      t0=(julian(ep1)-2415020.313d0)/36524.2199d0
      t=(julian(epoch)-2415020.313d0)/36524.2199d0
      eps=23.4523-0.013*t
      eps=eps*rad
      omega=259.1833d0-1934.1420d0*t+0.002078d0*t*t
      omega=omega*rad
      call sunpos(epoch,aa,dd,slon,iflag)
      slon=slon*rad
      write(*,543)ep1,ep2
333      read(11,222,end=190)text1,ar1,ar2,ar3,dec1,dec2,dec3,text2
222      format(a10,i2.2,1x,i2.2,1x,f6.2,1x,i2.2,1x,i2.2,1x,f6.2,a10)
51      CALL PRECES(t0,t,AR1,AR2,AR3,DEC1,DEC2,DEC3,AR1F,AR2F,AR3F,
1      DEC1F,DEC2F,DEC3F)
      write(*,*) ' General precession only '
      write(*,*) ar1f,ar2f,ar3f
      write(*,*) dec1f,dec2f,dec3f
      write(*,*)
      write(*,*) ' Adding corrections for aberration and nutation '
      call conver(ar1f,ar2f,ar3f,alfa)
      call conver(dec1f,dec2f,dec3f,delta)
      alfa=alfa*rad*15.
      delta=delta*rad
      dalfa2=-aberr*(dcos(alfa)*dcos(slon)*dcos(eps)+dsin(alfa)*
=dsin(slon))/dcos(delta)
      ddelt2=-aberr*(dcos(slon)*dcos(eps)*(dtan(eps)*dcos(delta)-
+dsin(alfa)*dsin(delta))+dcos(alfa)*dsin(delta)*dsin(slon))
c      write(*,*) dalfa2,ddelt2
c *** now dalfa2,ddelt2 are corrections for aberration in arcsec. *****
      dpsi=-(17.2327+0.01737*t)*dsin(omega)
      =-(1.2729+0.00013*t)*dsin(2.*slon)
      =+0.2088*dsin(2.*omega)
      deps=(9.2100+0.00091*t)*dcos(omega)
      =+(0.5522-0.00029*t)*dcos(2.*slon)
      dalfa1=dpsi*(dcos(eps)+dsin(eps)*dsin(alfa)*dtan(delta))
      =-dcos(alfa)*dtan(delta)*deps
      ddelt1=dpsi*dsin(eps)*dcos(alfa)+deps*dsin(alfa)
c      write(*,*) dpsi,deps,dalfa1,ddelt1
      alfa=alfa/rad
      delta=delta/rad
      alfa=alfa+dalfa1/3600.+dalfa2/3600.
      delta=delta+ddelt1/3600.+ddelt2/3600.
      alfa=alfa/15.
      call riconv(alfa,ar1f,ar2f,ar3f)
      call riconv(delta,dec1f,dec2f,dec3f)
      WRITE(*,12)AR1F,AR2F,AR3F,DEC1F,DEC2F,DEC3F
12      FORMAT(/1X,'New apparent position is '/
+2x,2(I2.2,1X),f7.3,3X,I3.2,1X,I3.2,1X,f6.2)
      if(resp.eq.'F'.or.resp.eq.'f')then

```

```

write(10,222)text1,ar1f,ar2f,ar3f,dec1f,dec2f,dec3f,text2
goto333
endif
goto 1
190 STOP
END
C-----
SUBROUTINE PRECES(tin,tfi,HH,MM,SS,GRADI,PRIMI,ASEC,
1 HHF,MMF,SSF,GRADIF,PRIMIF,ASECF)
C precessione dall'equinozio di partenza a quello di arrivo usando
C le formule esatte
C-----
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INTEGER HH,GRADI,PRIMI,HHF,GRADIF,PRIMIF
IF(tin.eq.tfi) GO TO 100
PI=3.1415926536d0
RAD=PI/180.0
tfin=tfi-tin
C converte alfa e delta in forma decimale
CALL CONVER(HH,MM,SS,ALFA)
CALL CONVER(GRADI,PRIMI,ASEC,DELTA)
ALFA0=ALFA*15.0*RAD
DELTA0=DELTA*RAD
C calcola le costanti di precessione in rad
zeta0=(2304.250d0+1.396d0*tin)*tfin+0.302d0*tfin*tfin+0.018*
=tfin*tfin*tfin
z=zeta0+0.791d0*tfin**2+0.001d0*tfin**3
z=z*rad/3600.d0
zeta0=zeta0*rad/3600.d0
teta=(2004.682d0-0.853d0*tin)*tfin-0.426d0*tfin**2-.042d0*tfin**3
teta=teta*rad/3600.d0
aa=dcos(delta0)*dsin(alfa0+zeta0)
bb=dcos(teta)*dcos(delta0)*dcos(alfa0+zeta0)-dsin(teta)*
=dsin(delta0)
cc=dsin(teta)*dcos(delta0)*dcos(alfa0+zeta0)+dcos(teta)*
=dsin(delta0)
alfaf=datan2(aa,bb)
if(alfaf.lt.0.)alfaf=alfaf+2.d0*pi
deltaf=dasin(cc)
alfaf=alfaf+z
alfaf=alfaf/(rad*15.d0)
deltaf=deltaf/rad
C calcola alfa finale in ore e delta finale in gradi
C write(*,*)zeta0,z,teta
C write(*,*)q,x
C ALFAF=(ALFA0+ZETA0+Z+X)*(1.0/(RAD*15.0))
C DELTAF=(DELTA0+2.0*DATAN(DTAN(TETA*0.5)*(DCOS(ALFA0+ZETA0)-
C 1 DSIN(ALFA0+ZETA0)*DTAN(X/2.0))))*(1.0/RAD)
CALL RICONV(ALFAF,HHF,MMF,SSF)
CALL RICONV(DELTAF,GRADIF,PRIMIF,ASECF)
GO TO 200
100 HHF=HH
MMF=MM
SSF=SS

```

```
GRADIF=GRADI
PRIMIF=PRIMI
ASECF=ASEC
200 RETURN
END
SUBROUTINE CONVER(I1,I2,X,XX)
DOUBLE PRECISION X,XX
XX=DFLOAT(I1)+DFLOAT(I2)/60.0+X/3600.0
RETURN
END
SUBROUTINE RICONV(YY,J1,J2,Y)
DOUBLE PRECISION YY,Y
IF(YY.LT.0.) GO TO 300
J1=IFIX(SNGL(YY))
J2=(60.0*DABS(YY-DFLOAT(J1)))
Y=(60.0*DABS(YY-DFLOAT(J1))-DFLOAT(J2))*60.0
GO TO 400
300 J1=IFIX(SNGL(YY))
J2=(-60.0)*DABS(YY-DFLOAT(J1))
Y=(60.0*DABS(YY-DFLOAT(J1))-DABS(DFLOAT(J2)))*(-60.0)
400 RETURN
END
```



```
$debug
```

```
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
real*8 julian,pi
character*30 cc$
character*5 snam$
character*28 str
character*15 stringa(10)
character*50 inp$
INTEGER AR1,AR2,DEC1,DEC2
```

```
c      --- reading position from dec50.dat file ----
      open(unit=11,file='\croce\decnew.dat')

      READ(11,222)stringa(3), AR1,AR2,AR3,dec1,dec2,dec3,stringa(7)
222    format(a10,i2.2,1x,i2.2,1x,f6.2,1x,i2.2,1x,i2.2,1x,f6.2,a10)

c      ---- writinf of new skedule on disk ----

      write(stringa(4),13) ar1,ar2,ar3
13    format(i2.2,1x,i2.2,1x,f6.2)

      write(stringa(5),16) dec1,dec2,dec3
16    format(i2.2,1x,i2.2,1x,f6.2)

      minsor=ar1*60+ar2
      minst=minsor-32
      if(minst.lt.0)minst=minst+1440
      mstart=minsor-15
      if(mstart.lt.0)mstart=mstart+1440
      mstop=mstart+30
      if(mstop.gt.1440)mstop=mstop-1440
      istart=mstart/60
      mstart=mstart-istart*60
      istop=mstop/60
      mstop=mstop-istop*60
      ist=minst/60
      minst=minst-ist*60
57    write(stringa(1),57) ist,minst
      format('start',2i3.2)
      stringa(2)='1'
      stringa(6)='calib'
      write(stringa(8),'(i2.2,1x,i2.2)') istart,mstart
      write(stringa(9),'(i2.2,1x,2i2.2)') istop,mstop
      stringa(10)='stop'

c      ---- writing of skeduled calibrator with act. position ----
      open(unit=12,file='\croce\newsked.dat')
      do k=1,10
      write(12,'(a13)')stringa(k)
c      write(*,'(a13)')stringa(k)
      enddo
      write(*,*)'Corrected source position is ready in the skedule'
      STOP
```

END

```
c      SIDERAL TIME CONTROL PROGRAM FOR  -OSSERV.BAT-
c          name=sidtime1

real*8 ut, sider,lon,date
real*8 st,str

character*1 ans
character*60 dat,sorg
integer*2 h,m
real*4 s,gr,pr,error,er,ff,e
integer*2 year,ifreq,idurr,ierr,adr1,adr2,adr3,adclk,sync
integer*1 ut1,ut2,ut3,ut4,ut5,ut6,ut7,ut8,ut9
integer*2 fs,st5,st6
integer*1 st1,st2,st3,st4
integer*1 dhh,hhu,dmu,mmu,dsu,ssu,dcs,ucs
logical pps
integer*1 input_buffer(60)
equivalence(input_buffer(1),sorg)

ifreq=1500
idurr=1
adr1=768
adr2=769
adr3=770
adclk=38

open (unit=10,file='\croce\sider.err')

c -----
c -- date and UT time reading --
c -- from UT clock --

2  call clock(ut1,ut2,ut3,ut4,ut5,ut6,ut7,ut8,ut9)
   write(*,3) ut4,ut5,ut6,ut7,ut8,ut9
3  format (1x,2i1,':',2i1,':', 2i1)
   year=1900+ut3

   if(ut2.le.9.and.ut1.gt.9) then
10  write(dat,10) year,ut2,ut1
   format(i4, '.0',i1,i2)
   go to 18
   endif

   if(ut2.le.9.and.ut1.le.9) then
11  write(dat,11) year,ut2,ut1
   format(i4, '.0',i1,'0',i1)
   go to 18
   endif

   if(ut2.gt.9.and.ut1.le.9) then
12  write(dat,12) year,ut2,ut1
   format(i4, '.',i2,'0',i1)
```

```

    go to 18
endif

    write(dat,15) year,ut2,ut1
15   format(i4, '.',i2,i2)
18   continue
    read(dat,20) date
20   format(f10.4)
    write(*,20) date

    io=(ut4*10.+ut5)
    min=(ut6*10.+ut7)
c    -- the controll will be at next second!---
    sec=(ut8*10.+ut9)+1

    ut=io*15.+min*15./60.+sec*15./3600.
    iflag=0

c    -----
c
c    --- Sideral time computing ---
c    -- for the previous UT time--

    call sidg(date,ut,sider,iflag)
    lon=-11.64528
100  sider=(sider-lon)/15.
    io=sider
    min=(sider-io)*60.
    sec=(sider-io-min/60.)*3600.
    sider=sider*15.

c    -----

c    -- sid time reading synchronised with UT pps --
c    -- UT pps sync.--

110  call output(adr2,adclk)
    fs=input(adr1)
    pps=btest(fs,0)
    if(pps) go to 110
200  call output(adr2,adclk)
    fs=input(adr1)
    pps=btest(fs,0)
    if(.not.pps) go to 200

    sync=7
    call sclock(sync,st1,st2,st3,st4,st5,st6)
c    write(*,*) st5,st6
    ifreq=2500
c    call beep2(ifreq,idurr,ierr)
    dhh=st3/16
    hhu=st3-dhh*16
    dmuh=st4/16
    mmuh=st4-dmuh*16
```

```

dsu=st5/16
ssu=st5-dsu*16
dcs=st6/16
c    --the sid. time reading occurs 1 sec/100 later the ut pps sync.--
ucs=(st6-dcs*16)-1

c    write(*,300) dhh,hhu,dmu,mmu,dsu,ssu,dcs,ucs
c 300  format (1x,2i1,':',2i1,':', 2i1,':',2i1)
c    write(*,301) io,min,sec
c 301  format (i3.2,':',i4.2,':', f7.2)

str=(dhh*10.+hhu)*3600.+(dmu*10.+mmu)*60.+(dsu*10.+ssu)+
&((dcs*10.+ucs)/100.)

st=(io*3600.+min*60.+sec)
if(st.ge.86400) then
    st=st-86400
endif
write(*,*) str,st
error=str-st
write(*,55) error
55  format (' sid. error (sec) =f6.2)
write(*,*) '-----'
c    --Adding 1 sec. (Integrators constant-time)--
error=error+1

c    -- reading of the previous value --
rewind(10)
read(10,510)er
510  format(f6.2,f10.4)

e=error-er
write(*,512) e
512  format(' Var. =',f6.2)

if(abs(error-1).ge.99.) then
write(*,*) 'ERROR TO LARGE TO STORE ON DISK. CHECK THE CLOCK!'
go to 600
endif

rewind 10
write(10,510)error,date
write(*,*) ' '
write(*,511)error
511  format(' Integrator time const. + Sid.Err.= f6.2)

c    -- error alarm control --
600  error=error-1
if(abs(error).ge.0.2) then
c    call cls
write(*,55) error
write(*,*) '-----'

write(*,*) 'BE CAREFUL!!! Sid errorr > 200 Msec.'

```

```
      do 556 k=1,2
      do 555 i=0,9
      ifreq=1000+(i*100)
      idurr=1+(i*6)
      call beep2(ifreq,idurr,ierr)
555   continue
556   continue
      endif
557  stop
      end
```

```
$notruncate
$declare
```

```
c      *****PROGRAMMA DI CALIBRAZIONE-CORRELATORI*****
c      *****DEL RADIOTELESCOPIO "CROCE DEL NORD"*****
c
c      S.Montebugnoli 12/1/90
c
c      --Il programma assume che all'ingresso dei correlatori
c      venga applicata una potenza di rumore di 600 microWatt
c      ed una marca interna correlata pari a 30 microWatt.
c
c      --Il programma richiede la lettura dell'ora e data
c      da effettuarsi tramite la routine CLOCK.
c
c      --Cambiare l'informazione sulla'anno in corso, nella
c      subroutine CLOCK.
c
c      --Il programma e'linkato con le routines:
c      CLOCK-FORT488-CONV-WAIT-OUTPORT-UCLOCK
c      NOLIMIT.LIB
c
c      --La misura del fondo viene fatta in un tempo di
c      inm secondi
c
c      =====SIMBOLI USATI=====
c      1) ICP=Primo Correlatore
c      2) ICN=Ultimo correlatore
c      3) INM=Numero di Misure
c      4) MA=Marca Interna.....(default=30 microWatt)
c      5) RF=Livello di Rumore.(default=600 microWatt)
c      6) DY=Delay fra le diverse misure
c      7) TE=Temperatura Stanza
c      =====
c
c      integer*2 icp,icn,inm,resp,ioccheck,a2
c      integer*1 t1,t2,t3,t4,t5,t6,t7,t8,t9,indice
c      integer*2 irow,icolm,ivpage,ierr,ifreq,idur,pc,nc,r,w
c      integer*2 ab1,ab2,delay,idat,day,cgu,dgu,ggu
c      integer*2 ma,rf,dy,di,ce,mm,ct,ze,i,j,jp,iseg,k,a1,key
c      integer*2 adr1,adr2,adr3,adr4,adclk,fs,ck,input,inf
c      integer*4 aseq
c      integer*2 k1,k2,k3,k4,k5,k6,k7,k8,k9,l
c      integer*2 n1,n2,n3,n4,n5,n6,iii
c      logical pps,exist
c      integer*2 iam(0:95),if(0:95),ifa(0:95)
c      integer*2 iis(0:95),ifq(0:95),ic1(0:95),num,status
c      real*4 ite,c,s,icos,ci,rd,fi,cp,cn,nm,ccp,ccn
c      real*8 f(0:96),c1(0:95),c2(0:95),c3(0:95),c4(0:95),am(0:95)
c      real*8 fa(0:95),fm(0:95),is(0:95),fq(0:95),a(0:95)
c      real*8 d1(0:95),s1(0:95),s2(0:95),s3(0:95),s4(0:95)
```

```

    ite=a2*0.00305175
1    icp=1
    icn=96
    call cls
    write (*,10)'PROGRAMMA DI CONTROLLO CORRELATORI'
    write (*,*)'      '
    j=15
    do5 i=1,159,2
5    call poke (iseg,i,j)
10   format(20x,a36)
    write (*,15)inm,icp,icn
15   format(' I Correlatori da controllare con ',i2,' Misure, vanno
&dal ',i2,' al ',i2,'      OK? (S/N)')
c    read(*,'(a5)') rinp
c    if (rinp.eq.'s'.or.rinp.eq.'S') go to 59
c    write (*,*)'
c
c    -----
c    ---- scrittura alla riga 7 ----
30   write (*,*)'Canale di Partenza (solo dispari)'
c    read (*,*) cp
    cp=1
c    ---- controllo se il numero e` dispari ----
    cp=abs(cp)
    cp=int(cp)
    ccp= cp-int(cp/2)*2
    if (ccp.eq.0) go to 30
    icp=cp
c    -----
c    50 write (*,*)'Canale di Arrivo (solo pari)'
c    read (*,*) cn
    cn=96
c    ---- scrittura della risp. alla riga 10 ----
    cn=abs(cn)
    cn=int(cn)
    ccn=cn-int(cn/2)*2
    if (ccn.ne.0) go to 50
    icn=cn
    if(cp.ge.cn) go to 1
    if (icn.gt.96) then
        write(*,*)'ATTENZIONE! Esistono solo 96 Canali!!!!'
        pause
        go to 1
    endif
c    -----
c    55 write (*,*)'Numero di misure per canale?? (>1)'
c    read (*,*) nm
    nm=10
    nm=abs(nm)
    nm=int(nm)
    if (nm.lt.2) go to 55
    inm=nm
c    ---- def.canali MUX. ----
59   pc=icp-1
    nc=icn-1
```



```

c      **scrittura data e ora con la subroutine clock**
      call cls
      call clock(t1,t2,t3,t4,t5,t6,t7,t8,t9)
      write (*,60)'PROGRAMMA DI -PROVA CORRELATORI-'
      write (*,60)'-----'
60     format (20x,a35)
c     ---- scrittura highlight della riga precedente ----
      j=15
      do 70 i=1,319,2
70     call poke (iseg,i,j)
      write (*,75)'TEMPERATURA STANZA=',ite
75     format (25x,a22,f4.1)
      write (*,80) t1,t2,t3,t4,t5,t6,t7,t8,t9
80     format(28x,i2,'/',i2,'/',i2,'/',' ',',2i1,':',2i1,':',2i1)
c     -----
c     ***** INIZIO PROCEDURA DI CONTROLLO *****
c     -----
c     ---- azzeramento variabili ----
      do 465 i=0,95
      c1(i)=0
      qd(i)=0
      s1(i)=0
465    continue
c     -----
c     ---- nm MISURE DEL FONDO A 0 GRADI ----
c     -----
c     ---- si fanno -inm- misure in inm sec.----
      do 480 k=1,inm
      delay=1
      call wait (delay)
      do 470 i=pc,nc
      a1=i
      call conv (a1,a2)
      c1(i)=a2+c1(i)
      qd(i)=a2**2+qd(i)
470    continue
480    continue
      do 490 i=pc,nc
      s1(i)=sqrt(abs(((inm*qd(i))-c1(i)**2)/(inm*(inm-1))))
      c1(i)=c1(i)/inm
490    continue
      write (*,*)'FATTE MISURE FONDO A 0 Gradi!'
c     -----
c     ---- nm MISURE CON MARCA A 0 GRADI ----
c     -----
      ab1=84
      ab2=64
      call outport (ab1,ab2)
c     ---- routine di wait:lettura di 5 sec----
      delay=5
      call wait (delay)
c     -----
498    continue

```

```
c      ---- azzeramento variabili ----
      do 500 i=0,95
      c2(i)=0
      qd(i)=0
      s2(i)=0
500    continue
c      ---- nm misure al secondo ----
      do 520 k=1,inm
      delay=1
      call wait(delay)
      do 510 i=pc,nc
      a1=i
      call conv (a1,a2)
      c2(i)=a2+c2(i)
c      qd(i)=a2**2+qd(i)
510    continue
520    continue
      do 530 i=pc,nc
c      s2(i)=sqrt(abs(((inm*qd(i))-c2(i)**2)/(inm*(inm-1))))
      c2(i)=c2(i)/inm
530    continue
      write (*,*) 'FATTE MISURE CON MARCA A 0 Gradi!'
c      -----
c      ---- nm MISURE CON MARCA A 90 GRADI ----
c      -----
      ab1=85
      ab2=64
      call outport (ab1,ab2)
c      ---- routine di wait:lettura di 5 sec----
      delay=5
      call wait(delay)
c      -----
536    continue
c      ---- azzeramento variabili ----
      do 540 i=0,95
      c4(i)=0
      qd(i)=0
      s4(i)=0
540    continue
c      ---- inm misure, una al sec.----
      do 570 k=1,inm
      delay=1
      call wait(delay)
      do 560 i=pc,nc
      a1=i
      call conv (a1,a2)
      c4(i)=a2+c4(i)
c      qd(i)=a2**2+qd(i)
560    continue
570    continue
      do 580 i=pc,nc
c      s4(i)=sqrt(abs(((inm*qd(i))-c4(i)**2)/(inm*(inm-1))))
      c4(i)=c4(i)/inm
580    continue
```

```

write (*,*)'FATTE MISURE CON MARCA A 90 Gradi'
c -----
c ---- nm MISURE DEL FONDO CON 90 Gradi INSERITI ----
c -----
ab1=81
ab2=64
call outport (ab1,ab2)
c ---- routine di wait:lettura di 5 sec----
delay=5
call wait (delay)
c -----
c ---- azzeramento variabili ----
589 do 600 i=0,95
c3(i)=0
qd(i)=0
s3(i)=0
600 continue
c ---- inm misure, una al sec. ----
do 620 k=1,inm
delay=1
call wait(delay)
do 610 i=pc,nc
a1=i
call conv (a1,a2)
c3(i)=a2+c3(i)
c qd(i)=a2**2+qd(i)
610 continue
620 continue
do 630 i=pc,nc
c s3(i)=sqrt(abs(((inm*qd(i))-c3(i)**2)/(inm*(inm-1))))
c3(i)=c3(i)/inm
630 continue
write (*,*)'FATTE MISURE DEL FONDO A 90 Gradi'
c ---- si tolgono i 90 gradi----
ab1=80
ab2=64
call outport (ab1,ab2)
c -----
c ---- CALCOLO DELLA FASE,AMPIEZZA,FONDO IN QUADR. ----
c -----
do 700 i=pc,nc
c=c2(i)-c1(i)
s=c4(i)-c3(i)
if (c.eq.0) go to 644
go to 646
644 c=ep
646 f(i)=atan2(s,c)*rd
f(i)=anint(f(i))
c ---- trasformazione in angoli "positivi" ----
if(f(i).gt.0) go to 680
f(i)=360+f(i)
c ---- ampiezza ----
c 680 am(i)= (sqrt(c**2+s**2))/ma
fq(i)=abs((c1(i)-c3(i))/c1(i))*100

```

```

      fq(i)=anint(fq(i))
700   continue
c    ---- differenza di fase ----
      do 710 i=pc,nc,2
      fa(i)=f(i+1)-f(i)
      if(abs(fa(i)).gt.180.and.f(i+1).le.f(i)) then
         fa(i)=-360+f(i)-f(i+1)
      endif
      if(abs(fa(i)).gt.180) then
         fa(i)=f(i+1)-(360-f(i))
      endif
      fa(i)=abs(fa(i))
710   continue
c    -----
c    ---- MISURA DELL'ISOLAMENTO ----
c    -----
c    ---- si toglie un Db in Media Freq. ----
      ab1=88
      ab2=64
      call outport (ab1,ab2)
c    ---- routine di wait:lettura di 5 sec----
      delay=5
      call wait (delay)
720   continue
c    ---- azzeramento variabili ----
      do 725 i=0,95
      fm(i)=0
725   continue
c    ---- inm misure del fondo maggiorato di 1 Db ----
      do 740 k=1,inm
      delay=1
      call wait (delay)
      do 730 i=pc,nc
      a1=i
      call conv (a1,a2)
      fm(i)=a2+fm(i)
730   continue
740   continue
c    ---- calcolo di fm(i) ----
      do 750 i=0,95
      fm(i)=fm(i)/inm
750   continue
c    -----
c    ---- REINSERZIONE DEL Db IN MEDIA FREQUENZA ----
c    -----
      ab1=80
      ab2=64
      call outport (ab1,ab2)
c    ---- calcolo dell'isolamento ----
      do 760 i=pc,nc
      is(i)=abs(fm(i)-c1(i))/(am(i)*rf*0.259)
c    nella precedente 0.259 e` di quanto si innalza il fondo
c    togliendo (amplificando) 1 Db in Media Frequenza
      if (is(i).eq.0) is(i)=ep

```

```

      is(i)=-4.343*log (is(i))
760  continue
c    ---- formattazione dei dati per la stampa ----
      do 790 i=pc,nc
      iam(i)=nint(am(i)*ma)
      if(i)=nint(f(i))
      ifa(i)=nint(fa(i))
      iis(i)=nint(is(i))
      ifq(i)=nint(fq(i))
      icl(i)=nint(c1(i))
      s1(i)=abs(s1(i))
790  continue
c    ---- ricevitori su antenna ----
      ab1=0
      ab2=64
      call outport (ab1,ab2)
c    ---- stampa dei risultati ----
c792  write (*,*) 'STAMPA --> SCHERMO (1) STAMPANTE (2) '
792  continue
      resp=0
c    ---- st. su stampante----
      call uclock(0,n1,n2,n3,n4,n5,n6)
      cgu=n1/16
      ggu=n2/16
      dgu=n2-ggu*16
      day=(cgu*100)+(dgu*10)+ggu
      write (chdata,877)day
877  format(i3.3)

      write (str1,865)ite
      write (str,875) t1,t2,t3,t4,t5,t6,t7,t8,t9
      if(buffer.eq.'p') goto 860
      goto 908

860  resp=1
      open(unit=1,file='prn')
      write (resp,870)'TEMPERATURA STANZA=',ite
c    write (str1,865)ite
870  format (25x,a22,f4.1)
865  format (24x,' TEMPERATURA STANZA=',f4.1)
c    write (str,875) t1,t2,t3,t4,t5,t6,t7,t8,t9

      write (resp,875) t1,t2,t3,t4,t5,t6,
: t7,t8,t9,day
875  format(25x,i2,'/',i2,'/',i2,'/',' ',',2i1,':',2i1,':',2i1,'
&i3)

      write(resp,*) '=====
:===== '

      write(resp,876)'N.', 'INTERF.', 'LIV.', 'FASE',
&'ANG', 'ISOL', 'DFQ', 'FONDO', 'DISP.'
      write(resp,*) '=====
&===== '

```

```

876   format(3x,a2,2x,a7,15x,a4,2x,a4,3x,a4,2x,a5,2x,a3,3x,a5,4x,a5)
c     ---- indice per riga vuota ogni 12 righe (stampa) ----
      indice=1
      do 900 i=pc,nc
        l=i+1
        k2=int(l/2)
        k3=l-k2
        k8=int(i/12)
        k4=k3-k8*6
        k5=l-k2*2
        if (k5.eq.1) go to 880
        go to 885
880   write (resp,*) '      '
        sc$='c'
885   if (k5.eq.0) sc$='s'
        k1=int(i/48)
        k9=k8+1-k1*4
        if (k1.eq.0) ns$='N'
        if (k1.eq.1) ns$='S'
        write(resp,890) i+1,k4,'E*',k9,ns$,sc$,iam(i),
& if(i),ifa(i),iis(i),ifq(i),icl(i),s1(i)
        if(indice.eq.12) then

          write(resp,*) '-----'
&-----'
          indice=0
          endif
          indice=indice+1
890   format(i4,' ',i2,a2,i2,a5,a5,' ',',',',i5,' ',',i4,' ',',i3
& ,',',i3,' ',',i3,' ',',i6,' ',',f5.0)

900   continue
908   continue

      dsdat='b:corr.'//chdata
      write(*,1234)dsdat
1234  format(1x,a30)
      open(unit=10,file=dsdat)
c     write(day$,904)day
c904  format(' Day=',i3.3)
      write(str2,905)'
905   format(a50)
      write (str3,910)'amp','fase','isol'
910   format('correl.',5x,a3,2x,a4,2x,a5)
      write (10,970) str1
      write (10,970) str
      write (10,970) str3
      write (10,970) str2
970   format (a60)

      do 1000 i=0,95
        iii=nint(iam(i)*10./ma)
        write (10,950) i+1,iii,if(i),iis(i)
950   format(5x,i2,4x,i3,2x,i4,2x,i4)

```

```
1000  continue
      close(10)

      dsdat='\croce\calib\corr.'//chdata
      write(*,1234)dsdat
      open(unit=11,file=dsdat)
      write (11,970) str1
      write (11,970) str
      write (11,970) str3
      write (11,970) str2

      do 1010 i=0,95
      iii=nint(iam(i)*10./ma)
1010  write (11,950) i+1,iii,if(i),iis(i)
      continue
      close(11)

      write(resp,'(1x,a1)')char(12)
c    ---- controllo della fase e stampa ----
      do 1222 i=pc,nc,2
      if (ifa(i).le.85.or.ifa(i).ge.95) then
1223  write (resp,1223)i+1,ifa(i)
      format(' ang. canale N. ',i2,' fuori! ',i3)
      endif
1222  continue
      write(resp,'(1x,a1)')char(12)
1002  continue
      stop ' '
      end
```

```
$DEBUG
```

```
$NOTRUNCATE
```

```
$DECLARE
```

```
c =====
c ***** SCHED1 OPERATIVE SYSTEM*****
c               for OSSERV.BAT Program
c =====
```

```
c               name=tosked
```

```
c               S.Montebugnoli-A.Cattani-G.Grueff
```

```
c               -1991-
```

```
c -----
c This program has to be linked with:
c -----
```

```
c -SUCONV          --> A/D channels conversion
c -TAPEC           --> Tape control routine
c -FORT488        --> IEEE 488 routine
c -CLOCK          --> UT clock reading
c -CLOCKSID       --> SID clock reading
c -NOLIMIT.LIB   --> Nolimit libraries
c -UCLOCK         --> Fast UT clock reading
c -PCPOINT        --> Send pointing infor.
c -PCEND          --> Close serial comunic.
c -----
```

```
integer*2 numcom,ibaud,ipar,iwlen,istop,irs,ics,ids,icd,rel3
* ,ilf,ierr,i1,i2,i3,k
integer*2 address,level,adr1,adr2,adr3,w,r,key,rel,rel1,rel2
integer*1 vet(2884),b1,b2,b3,b4,gg,pr
integer*2 it1,it2,it3,it4,it5,it6,sync
integer*1 t1,t2,t3,t4,t5,t6,t7,t8,t9,hstart,mstart,hstop,mstop
integer*4 points,nump,h1,h2
integer*2 nchr,ltctime,ntape,nfile,delay,num,movant(5)
integer*1 rcg,rgg,rhh,rmm,rss,rsc,adclk,fs,nast,ki
integer*4 rec,ski
integer*2 irow,icolm,ivpage,ifreq,idurr,in,ierror,vrel
integer*2 i,j,iseg,hst,mst,sst,res,day,cgu,ggg,dgu
integer*4 aseq
logical flagi
real*4 se,decl(5),decl2,decl3,dew,dns,temp,siderr
real*8 date
character*5 rinp,t1$,t2$,t3$,t4$,t5$,t6$,t7$,t8$,t9$
character*5 start$
character*80 stri(14),out,text(14)
character*30 idsk,epoch
character*35 dec$,transit
character*10 scom$
character*5 ntap$,nfil$,day$
character*35 sfitt$,snom$,sked$
character*1 chr(80),cr,flag
```

```
c -----
c character*1176 text1
```

```
c character*12 str(98)
```



```
equivalence(text1,str(1))
equivalence(vet(1125),text1)
```

c

```
-----
equivalence(aseg,iseg)
common/corre/text
equivalence(out,chr(1))
equivalence(vet(5),stri(1))
```

```
num=1
call getarg(num,sked$,ierr)
```

c

```
-----
numcom=1
ibaud=6
ipar=0
iwlen=7
istop=1
irs=0
ics=0
ids=0
icd=0
ilf=0
```

```
call iopcom(numcom,ibaud,ipar,iwlen,istop,irs,ics,ids
+,icd,ilf,ierr)
if(ierr.ne.0)then
  write(*,*)'Error in opening RS232 ',ierr
  stop
endif
```

c

```
-----
1  aseg=45056
   adr1=768
   adr2=769
   adr3=770
```

```
w=0
r=1
rcg=32
rgg=33
rhh=34
rmm=35
rss=36
rcs=37
ltctime=32
adclk=38
fs=0
res=0
rel=65
rel1=64
rel2=66
rel3=67
```

```
c      ---- System Reset ----
      vrel=0
      call outport(vrel,rel1)
      call outport(vrel,rel)
      call outport(vrel,rel2)
c      call outport(vrel,rel3)

c      ---- reading time informations from disk ----
      open(unit=15,file='\croce\sider.err')
      read(15,5,err=6)siderr,date
5      format(f6.2,f10.4)

c      -----

6      do 2 ki=1,5
      movant(ki)=1
2      continue

c      -----
c      PC488 interface initializing
      address=1
      level=0
      call initia (address,level)
c      ----- c
c      ---- clear memory buffer ----
      do 10 i=1,2884
      vet(i)=0
10     continue
c      ---- IEEE 488 first bytes informations for TARRAY ----
c      ---- '#' character ----
      vet(1)=35
c      ---- write one block command ----
      vet(2)=69
c      ---- MSByte's block length ----
      vet(3)=11
c      ---- LSByte's block length ----
      vet(4)=64
c      ---- input mode of operation ----
c      ---- clear screen ----
      nast=0
      call cls
      cr=' '
c      ---- make speaker beep ----
      ifreq=1500
      idurr=5
      call beep2(ifreq,idurr,in)
      write (*,*)'          '
      write (*,15)'NORTHERN CROSS RADIOTELESCOPE DATA ACQUISITION SYST
&EM'
      write (*,15)'-----'
```

```

&- '
  write(*,*)
  write(*,*)
15  format (13x,a53)
    do 20 i=161,639,2
      j=112
20  call poke (iseg,i,j)
    irow=10
    icolm=28
    ivpage=0
    goto 500
c
c *****
c -----End Of Program-----
c *****
30  call cls
    call wrfmk(b1,b2)
    call decode(b1,b2,b3,b4)
    call beep2(ifreq,idurr,in)
    stop'                               End of program'
c -----
500  write (*,*) '

    open(unit=15,file=sked$,status='old')
    write(*,*)' Enter tape number'
    read(*,1771)ntape
    12  continue
    1720 continue
    call cls
c *****
c *   writing the header -1750- *
c *****
    1750 continue
c ---- clear screen ----
    1765 continue
c -----
    ifreq=4000
    idurr=1
    call beep2(ifreq,idurr,in)

c -----
    stri(1)='-NORTHERN CROSS- RADIOTELESCOPE'
    write (*,*) stri(1)
    write(*,*)
    write(*,*)

    write(*,*)'===== '
    write(*,*)'WAIT FOR OPERATIONS STARTING TIME '
    write(*,*)'===== '

c ---- operation starting time control ----

329  read(15,'(a5,2i3)',end=3000) start$,hstart,mstart
    write(*,'(1x,a5,2i5.2,5x,12h ESC to exit)')start$,

```

```

* hstart,mstart
write(*,*) ' '
if (start$.ne.'start') stop' Wrong schedule'
328 call clocksid(t1,t2,t3,t4,t5,t6,t7,t8,t9)

hst=t4*10+t5
mst=t6*10+t7
sst=t8*10+t9

irow=9
icolm=4
ivpage=0
call putcur(irow,icolm,ivpage,ierror)

c -- -0- key control (to go on) --
call inkey(cr,key,in)
if(in.eq.1.and.key.eq.24) goto 1760

332 write(*,332)hst,mst,sst
format (' Current Time',3i5.2)
call inkey(cr,key,in)
if(key.eq.1)stop'Program aborted by user '
if(hst.ne.hstart.or.mst.ne.mstart) then
  ifreq=6000
  idurr=1
  call beep2(ifreq,idurr,in)
  h1=hstart*3600+(mstart*60)
  h2=hst*3600+(mst*60)+sst
  if (h1.le.h2) then
    h1=h1+86400
  endif
  points=h1-h2

hst=points/3600
mst=(points-(hst*3600))/60
sst=(points-(hst*3600)-(mst*60))

irow=10
call putcur(irow,icolm,ivpage,ierror)

337 write(*,337)hst,mst,sst
format (' Count-down ',3i5.2)
c write(*,341)points
c341 format(' Time to Start ',1i6)

c -----sked informations-----
write(*,*) ' '
write(*,*) ' CURRENT SKEDULE'
write (*,*) '-----'
write(*,*) ' ',sked$
write (*,*) '-----'
write(*,*) ' '
write(*,*) ' Hit -0- (Operator) key to go on '

```

```
c      -- -0- key control (to go on) --
      call inkey(cr,key,in)
      if(in.eq.1.and.key.eq.24) goto 1760

      go to 328
      endif

1760   continue
      call cls

      write(*,*)'===== '
      write(*,*)'POINTING OPERATIONS IN PROGRESS '
      write(*,*)'===== '
      write(*,*)' '

c      ---- tape number ----
      sfitt$='TAPE NUMBER '

1771   format(i3)
1770   write (*,*) sfitt$,ntape
1790   write (ntap$,1791)ntape
1791   format(2x,i3.3)
      stri(2)=sfitt$//ntap$

c      ---- file number ----
      sfitt$='FILE NUMBER '
      read(15,*,end=30)nfile
1800   write (*,*) sfitt$,nfile
1820   write(nfil$,1791)nfile
      stri(3)=sfitt$//nfil$

c      ---- source name ----
      sfitt$='SOURCE NAME '
      read(15,'(a35)') snom$
      stri(6)=sfitt$//snom$
      write (*,*) stri(6)

c      ---- getting alfa ----
      sfitt$='ALFA '
      read (15,'(a35)') snom$
      transit=snom$
      stri(7)=sfitt$//snom$
      write (*,*) stri(7)

c      ---- getting delta ----
      sfitt$='DELTA '
      read(15,'(a35)') snom$
      read (snom$,388)gg,pr,se
388   format (i2,i3,f5.1)
398   do 327 ki=1,4
      decl(ki)=gg+pr/60.+se/3600.
      if (ki.eq.2) then
      . stri(8)=sfitt$//snom$
        write (*,*) stri(8)
      endif
```

```

327   continue
      decl(5)=44.57
c   *** decl(5)=ZENITH *****
c   ---- program's name ----
      sfitt$='PROGRAM NAME
      read (15,'(a35)') snom$
      stri(9)=sfitt$//snom$
      write (*,*) stri(9)
c   *****

c   ---- send informations ----
c   ---- to pointing PC ----
c   pointing informations sent are:
c
c   decl(1)=NS Kerosene pointing
c   decl(2)=EW mechanical pointing
c   decl(3)=NS mechanical pointing
c   decl(4)=NS delays
c   decl(5)=NS phase shifters (ZENITH)

      decl2=decl(2)
      decl3=decl(3)
      write(*,*) decl2
c   write(*,*) decl3

c   =====
1795  call pcpoint(decl,flag,movant)
c   =====
      write(*,*) decl(2)

c   --- stop?? ---
      write(*,*) ' <ESC> to Quit'

      call inkey(cr,key,in)
      if(in.eq.0) go to 7771
      if(key.ne.1)then
1793  call inkey(cr,key,in)
          if(in.ne.0)goto 1793
          goto 7771
      endif
      if(key.eq.1)then
          call pcend(flag)
          call pause(1)
          call clscom(numcom,ierror)
          write(*,*) ' Program stopped by user; flag =',flag
          stop
      endif

7771  dew=abs(decl(2)-decl2)
      dns=abs(decl(3)-decl3)
      write(*,*) dew

```

```
c      ---- flag 1/9 ----
c      -se c'e` errore 6 o 7 (no reply e com.err.) resetta e rientra-

      if(flag.eq.'6'.or.flag.eq.'7') then
        write(*,*) 'flag=',flag
        call ppend(flag)
        call pause(1)
        call clscom(numcom,ierror)
        call pause(1)
        call iopcom(numcom,ibaud,ipar,iwlen,istop,irs,ics,ids
+,icd,ilf,ierr)
        Write(*,*) 'Error (6/7) Occured! Waiting for 50 sec.'
        delay=50
        call wait(delay)
        goto 1795
      endif

      if (flag.eq.'1'.or.
:      flag.eq.'2'.or.
:      flag.eq.'3'.or.
:      flag.eq.'4'.or.
:      flag.eq.'5'.or.
:      flag.eq.'8'.or.
:      flag.eq.'9')then
        write (*,*) 'Irrecoverable Error ', flag
        delay=50
        Write(*,*) 'Error Occured! Waiting for 50 sec.'
        call wait(delay)
        goto 1795
      endif

c      -----
c
c      PARTE DA CONTROLLARE

      if (flag.eq.'B')then
        write(*,*) 'flag = B'

        goto 2223
        write(*,*) '
        write(*,*) 'NOT POSSIBLE TO HAVE RIGTH POSITION'
        write(*,*) '-----'
        write(*,*) '
        write(*,*) 'waiting for 3 min....'
        delay=180
        call wait(delay)
        goto 1795
      endif

c      -----
c
c      ---- flag=A ----
2223  if(flag.eq.'A'.or.flag.eq.'0') then
        write(*,*) 'flag = ',flag
```

```
if(dew.lt.0.4.and.dns.lt.0.80) goto 995
```

```
write(*,*) 'ANTENNA UNDER PULSAR COMPUTER CONTROL'
write(*,*) '
write(*,*) 'NOT POSSIBLE TO HAVE RIGTH POSITION'
write(*,*) '-----'
write(*,*) '
write(*,*) 'waiting for 3 min.....'
```

```
delay=180
call wait(delay)
```

```
goto 1795
endif
```

```
c *****
```

```
995 continue
```

```
c ---- comments ----
```

```
sfitt$='COMMENTS '
read (15,'(a10)') scom$
write(dec$,330)decl,flag
```

```
c write(*,*)decl,flag
```

```
330 format(5f6.2,1x,a1)
stri(10)=sfitt$//scom$//dec$
write (*,*) stri(10)
```

```
c ----- observation start -----
```

```
read (15,*) hstart,mstart
read (15,*)hstop,mstop
write (stri(11),9020)hstart,mstart
write(*,9020)hstart,mstart
```

```
9020 format (' Sid. START TIME          ',' ',',2i3.2)
```

```
write (stri(12),9030)hstop,mstop
```

```
write(*,9030)hstop,mstop
```

```
9030 format (' Sid. STOP TIME          ',' ',',2i3.2)
```

```
c ---- correlators Data ----
```

```
day=0
```

```
day$=' '
```

```
c --- 'day' number reading ---
```

```
sync=0
```

```
call uclock (sync,it1,it2,it3,it4,it5,it6)
```

```
cgu=it1/16
```

```
ggu=it2/16
```

```
dgu=it2-ggu*16
```

```
day=(cgu*100)+(dgu*10)+ggu
```

```
write(day$,992) day
```

```
992 format(i3.3)
```

```
write(*,*)day,' ',day$
```

```
idsk='\croce\calib\corr.'//day$
```

```
stri(13)='Corr.xxx NOT present!'
```

```
inquire(file=idsk,exist=flagi)
```

```
if(.not.flagi) then
```

```
day=day-1
```



```
write(day$,992) day
idsk='\croce\calib\corr.'//day$
inquire(file=idsk,exist=flagi)
if(.not.flagi) go to 437
endif
stri(13)='Present Corr.'//day$
write(*,*) idsk
open(unit=12,file=idsk,status='old ')
rewind(12)
c
---- corr. data reading ----
read(12,431) temp,epoch
431 format(44x,f4.1/25x,a25//)
write(stri(14),434)temp,epoch
434 format('Correlators tested at T=',f4.1,' Deg. ',a20)
write(*,*) temp,' ',epoch,' ',idsk
c439 format(f4.1,' ',a21,' ',a30)
do 432 k=1,96
read(12,433) i1,i2,i3
433 format(11x,i3,3x,i3,4x,i3)
write(str(k),435)i1,i2,i3
435 format(3(1x,i3))
432 continue
write(str(97),'(f6.2)')siderr
write(str(98),'(f10.4)')date

c
-----
437 continue
c
---- points to acquire ----
9021 format(36x,2i3)
nump=0
h1=hstop*3600+(mstop*60)
h2=hstart*3600+(mstart*60)
if (h1.le.h2) then
  h1=h1+86400
endif
points=1+(h1-h2)/9
c
-----
write(*,*)' Waiting for START TIME'
write(*,*)' Hit -O- key to go on '

write(*,*)' <ESC> to quit'
c
---- start time control ----
9040 call clocksid (t1,t2,t3,t4,t5,t6,t7,t8,t9)
hst=t4*10+t5
mst=t6*10+t7
c
-- -O- key control (to go on) --
call inkey(cr,key,in)
if(in.eq.1.and.key.eq.24) goto 9500

if (hst.eq.hstart.and.mst.eq.mstart). go to 9500
call inkey(cr,key,in)
if(in.eq.0) go to 9040
if(key.eq.1)then
```

```
c      call wrfmk(b1,b2)
c      call decode(b1,b2,b3,b4)
      stop 'Program aborted'
endif

c      -- -0- key control (to go on) --
      call inkey(cr,key,in)
      if(in.eq.1.and.key.eq.24) goto 9500

      goto 9040
9500   continue

c      *****
c      ----- write header on tape -----
c      *****
c      ---- read date/clock informations ----
      call clock (t1,t2,t3,t4,t5,t6,t7,t8,t9)

c      --- 'day' number reading ---
      sync=7
      call uclock(sync,it1,it2,it3,it4,it5,it6)
      cgu=it1/16
      ggu=it2/16
      dgu=it2-ggu*16
      day=(cgu*100)+(dgu*10)+ggu

      ifreq=4000
      idurr=3
      call beep2 (ifreq,idurr,in)
      write (stri(4),1822)t1,t2,t3,day
1822   format('DATE',30x,3i3.2,3x,i3.2)
      ifreq=400
      idurr=3
      call beep2 (ifreq,idurr,in)
      write (stri(5),1823)t4,t5,t6,t7,t8,t9
1823   format('UT TIME',28x,2i1,':',2i1,':',2i1 )
c      stri(5)=sfitt$/t4$/t5$/t6$/t7$/t8$/t9$
c      write (*,1830)'DATE',t1,t2,t3,day
1830   format (a5,31x,i2,'/', i2,'/', i2, 3x,i3.2)
c      write (*,1840)'UT TIME',t4,t5,t6,t7,t8,t9
1840   format (a8,28x,2i1,':',2i1,':',2i1)
      b1=0
      b2=0
      call write (vet,b1,b2)
      call decode (b1,b2,b3,b4)
      call cls

c      *****
c      * starting of subconversion routine *
c      *****
c
c      read date/clock's information
      call clock (t1,t2,t3,t4,t5,t6,t7,t8,t9)
      t1$=char(t1)
```

```

t2$=char(t2)
t3$=char(t3)
t4$=char(t4)
t5$=char(t5)
t6$=char(t6)
t7$=char(t7)
t8$=char(t8)
t9$=char(t9)

sfitt$='starting      '
stri(3)=sfitt$//t1$//t2$//t3
sfitt$='ut time      '
stri(4)=sfitt$//t4$//t5$//t6$//t7$//t8$//t9$
write(*,*)'          -NORTHERN CROSS RADIOTELESCOPE DATA ACQU
:ISITION SYSTEM-'

```

```

write (*,*) '          -TELESCOPE ON LINE-'
write (*,*) '

```

```

c      ---- ON-LINE lamp -ON- ----
      vrel=1
      call outport(vrel,rel)

c      ---- set inverse mode ----
      j=112
      do 1900 i=161,639,2
1900   call poke (iseg,i,j)
c      -----
      write (*,1996)'start',t1,t2,t3
      write (*,1997)'ut time',t4,t5,t6,t7,t8,t9
1996   format(29x,a6,3x,i2,'/',i2,'/',i2)
1997   format(29x,a8,1x,2i1,':',2i1,':',2i1)
      write (*,1999) '-----'
      write (*,1999) ' HIT ALT-F TO STOP ACQUISITION'
      write (*,1999) '-----'

c      ---- set highlight mode ----
      j=15
      do 1998 i=1121,1599,2
1998   call poke (iseg,i,j)
c      -----
1999   format(21x,a33)
      write (*,*)'          tape N.',ntape
      write (*,*)'          file N.',nfile

c      -----sked informations-----
      write(*,*)' '
      write(*,*)' '
      write(*,*)' '
      write (*,2999) '-----'
      write(*,2018) sked$
      write(*,2899) stri(6)
      write(*,2019) hstart,mstart

```

```

write(*,3009) transit
write(*,2017)hstop,mstop
write (*,2999) '-----'
2018 format(20x,'SKEDULE ',18x,a18)
2019 format(20x,'OBSERVATION STARTED AT SID.TIME:',4x,i2.2,' ',i2.2)
3009 format(20x,'EXPECTED SOURCE TRANSIT IS AT: ',a13)
2017 format(20x,'OBSERVATION WILL STOP AT SID.TIME:',2x,i2.2,' ',
#i2.2)
2899 format(20x,a45)
2999 format(18x,a45)

```

```

c -----
c acquisition routine (9 acquisitions)
c -----
c ---- Record's Counter Reset ----
c rec=0
2000 ifreq=7000
idurr=1
call beep2(ifreq,idurr,in)
c ---- status control ----
call sbytes (b1,b2)
c -----
call conversion
rec=rec+1
nump=nump+1
write (*,2015)'record N. ',rec,' of',points
2015 format (28x,a10,i4,a3,i4)
if (nump.eq.points) go to 2030
2020 continue
c ---- "ALT-F" key control ----
call inkey (rinp,key,in)
nchr=1
if (key.eq.33.and.in.eq.2) go to 2030
go to 2000
c *****
c ---- End of File 1 fm is written ----
c *****
rinp=' '
2030 continue
call wrfmk (b1,b2)
call decode (b1,b2,b3,b4)
ifreq=4000
idurr=1
call beep2 (ifreq,idurr,in)
write (*,*)' '
2035 continue
c *****
c ---- end of data acquisition ----
c *****
ifreq=3000
idurr=1
call beep2 (ifreq,idurr,in)

```

```

c      ---- End of schedule control ----
      read (15,'(a5)')start$
      if (start$.ne.'stop') go to 1760
      call ppend(flag)
      call clscom(numcom,ierror)

      if(flag.ne.'0') then
c      ' write(*,*)'Failure to end properly ',flag
      ---- ON-LINE lamp -OFF- ----
      vrel=0
      call outport(vrel,rel)
      stop' '
      endif

      go to 329

3000  irow=23
      icolm=1
      ivpage=0
      call putcur(irow,icolm,ivpage,ierror)

c      -- 2th FM is written (END OF SKEDULE)
      call wrfmk (b1,b2)
      call decode (b1,b2,b3,b4)
      ifreq=4000
      idurr=1
      call beep2 (ifreq,idurr,in)

3001  write(*,3001)'END OF SCHEDULE ',sked$
      format(28x,a17,a35)

c      ---- ON-LINE lamp -OFF- ----
      vrel=0
      call outport(vrel,rel)
c      --- 1 rev. block ---
c      ---(to leave tape between last 2 file marks)---
      ifreq=700
      idurr=6
      call beep2 (ifreq,idurr,in)
      call rv1blk(b1,b2)
      end

      subroutine skip(nskip)

c      =====
c      SKIP NUM FILES FORWARD ON THE TAPE UNIT
c      =====
      integer*1 b1,b2,b3,b4
      integer*2 ifreq,idurr,in,resp

```

```
integer*4 nskip
integer*2 i
-----
c
c
resp=nskip
ifreq=2500
idurr=1
call beep2(ifreq,idurr,in)
if(resp.le.0)return
do 5610 i=1,resp
call fsfwhighsp (b1,b2)
call decode (b1,b2,b3,b4)
ifreq=2500
idurr=1
call beep2(ifreq,idurr,in)
5610 continue
return
end
```

\$DECLARE

```

C      =====
C
C      SKIP qN FILES ON THE TAPE UNIT
C
C      =====
C
C      This program has to be linked with:
C      ---- FORT488    --> IEEE488 Routines
C      ---- TAPEC      --> Tape Routines
C      ---- RDHEAD     --> Header Reading Routine
C      ---- NOLIMIT    --> Libraries
C
C      -----
C      integer*1 b1,b2,b3,b4,resp
C      integer*2 ifreq,idurr,in,na,istat
C      character*20 command
C      integer*2 ii,i,address,level
C      integer*4 nargs
C
C      -----
C      PC488 interface initializing
C      address=1
C      level=0
C      call initia (address,level)
C
C      -----
C
C      ifreq=2500
C      idurr=1
C      write(*,*)' '
C      call beep2(ifreq,idurr,in)
C      if(nargs().eq.2)then
C          na=1
C          call getarg(na,command,istat)
C          write(*,*)command
C          read(command,'(i4)')resp
C          go to 333
C      endif
C      write (*,*)'How many Files? (qn, + is forward, - is backward)'
C      read (*,*) resp
333      if(resp.lt.0)then
C          resp=-(resp-1)
C          goto 5800
C      endif
C      if(resp.eq.0)goto 6000
C      do 5610 i=1,resp
C          call fsfwhighsp (b1,b2)
C          call decode (b1,b2,b3,b4)
C          if(b1.ne.65)stop
C          ifreq=2500
C          idurr=1
C          write(*,'(1h+,i5)')i
C          call beep2(ifreq,idurr,in)
5610      continue
C          go to 5000

```

```
c      ---- call file search rev. high speed ----
5800  continue
      ifreq=2500
      idurr=1
      call beep2(ifreq,idurr,in)
      do 5810 i=1,resp
      call fsrhighsp (b1,b2)
      call decode (b1,b2,b3,b4)
      if(b4.eq.16.or.b4.eq.1)goto 5000
      ifreq=2500
      idurr=1
      ii=-i
      write(*,'(1h+,i4)')ii
      call beep2(ifreq,idurr,in)
5810  continue
      call fsfwhighsp(b1,b2)
      ii=-(resp-1)
      write(*,'(1h+,i4)')ii
      call beep2(ifreq,idurr,in)
      goto 5000
6000  call fsrhighsp(b1,b2)
      call decode(b1,b2,b3,b4)
      if(b4.eq.16.or.b4.eq.1)goto 5000
      call fsfwhighsp(b1,b2)
5000  stop '***** done ***** '
      end
```



```

character*1 idat1(2884)
integer*1 status(4),ch1,ff,printa
logical flag
character*25 epoch
character*2 nns(8),new(6),lun
integer*2 n1,n2,n3,n4,n5,n6,day,cgu,dgu,ggu,ncr,ncs
character*70 str1,str2,str3,str4,str5,str6,str7,str8,str9
character*40 sname
character*30 idsk
character*6 snam
character*3 ext
character*80 text(36),str,s$
real*8 array(10)
character*30 fill
dimension ampl(96),phase(96),coeff(96),ph(96),am(96),fns(8)
integer*2 idat(1440),ida,address,level,ipun(8),len,ncc
dimension data(96,701),a(6,8),p(6,8),aew(6),pew(6),ans(8),pns(8)
equivalence(array(1),str)
equivalence(idat1(5),idat(1))
equivalence(idat1(5),text(1))
equivalence(idat1(1),status(1))
c *** first index=channel, second index=time *****
c
data nns /'1N','2N','3N','4N','1S','2S','3S','4S'/
data new /'1E','2E','3E','4E','5E','6E'/
lun=1
call getarg(lun,printa,ncc)
write(*,*)ncc
if(printa.eq.'P')printa='p'
c printa='p'
pi=3.141592
ff=12
twopi=pi*2.
open(unit=24,status='scratch')
c *** init.IEEE 488
address=1
level=0
call initia(address,level)
open(unit=9,file='\croce\calib\map.cal')
c *** map.cal = file con i dati da usare per calibrare altre registrazion
**
c *** output
if(printa.ne.'p')open(unit=14,file='\log\cal.log',access='append')
if(printa.eq.'p')open(unit=14,file='prn')
write(14,*)' ***** CAL ** version 2 1 1991 **
*****'
write(14,*)' '
c *** write epoch etc *****
call read (idat1)
if(status(1).ne.64.and.status(1).ne.65) goto 100
do 451 k=1,12
451 write(14,452)text(k)

```

```
452 format(5x,a70)
   write(14,*)' '
c   write(14,*)' '
   write(14,*)' '
c
c
   write(*,*)'Enter source name (max 20 char.)'
   write(*,*)' if name=h, then take all data from tape header'
   read(*,521)sname
   if(sname.eq.'h'.or.sname.eq.'H')then
   str=text(4)
   write(24,159)(array(k),k=3,10)
   rewind(24)
   read(24,*)ida,mon,iear,day
   write(ext,'(i3.3)')day
   rewind 24
   str=text(7)
   write(24,159)(array(k),k=3,10)
   rewind(24)
   read(24,*)io,min,sec
   rewind(24)
159 format(7a8)
   str=text(8)
   write(24,159)(array(k),k=4,10)
   rewind (24)
   read(24,*)dg,pr
   rewind (24)
   str=text(10)
   write(24,159)(array(k),k=4,10)
   rewind(24)
   read(24,*)flux
   rewind(24)
c
c   --- reading time informations from disk ---
   open(unit=15,file='\croce\sider.err',status='old')
   read(15,*)siderr
c ***
   str=text(6)
   write(24,159)(array(k),k=4,10)
   rewind(24)
   read(24,160)sname
   len=40
   call justr(sname,len,ncc)
160 format(a40)
   goto 403
   endif
521 format(a20)
   write(*,*)'enter day number of observing date'
   read (*,*)day
   write(ext,'(i3.3)')day
   write(*,*)'Enter source R.A.,Dec,sid.error,'
   &,' flux (hh mm ss.ss dd pp ee.e fff.ff)'
   read(*,*)io,min,sec,dg,pr,siderr,flux
c
```

```
403      write(*,1233)day,io,min,sec,dg,pr,siderr,flux
c *****
      idsk='b:corr.'//ext
      open(unit=12,file=idsk,status='old')
c *** corr.ext = calibrazione dei correlatori (progr. corr) *****
c *** input
c *** lettura dati sui correlatori
      read(12,54)temp,epoch
      54 format(44x,f4.1/25x,a25//)
      do 37 k=1,96
      read(12,53)i1,i2,i3
      53 format(9x,3(2x,i4))
      ampl(k)=i1
      phase(k)=i2*pi/180.+pi
      37 coeff(k)=i3
c
c *** phase(96) = fasi dei correlatori in rad. *****
1233 format(i4,1x,2i3,f6.2,2f5.0,f6.2,f8.2)
c
c *** computing phase EW-NS ***
      d1=dg+pr/60.
      phdel=10.72*cos((44.57-d1-12.95)*pi/180.)
      phdel=phdel/(299.793/408.)
      phdel=- (phdel-int(phdel))*pi*2.
c
      call fasins(pns,dg,pr)
c
c *** pns(8) = fasi di puntamento in rad. *****
c
      write(14,453)sname,flux,io,min,sec,dg,pr,siderr
      write(9,453)sname,flux,io,min,sec,dg,pr,siderr
453 format(1x,'Calibrator is ',a40,/' flux ',f6.2,' R.A.',2i3.2,f6.2,
+' Dec',2f5.0,' Sid. time err.',f6.2)
      write(14,454)epoch,temp,ext
      write(9,454)epoch,temp,ext
454 format(1x,'Correlators calibrated on ',a25,'T='
+',f4.1,' °C',2x,'File CORR.',a3)
      write(14,*)' '
      write(14,*)' '
      call read (idat1)
      if(status(1).ne.64.and.status(1).ne.65) goto 100
      write(fill,51) (idat(k),k=1,6)
      read(fill,52)iday,ihd,miu,iou,iceu,iseu,mis,ios,ices,ises
      51 format(6z4)
      52 format(10i2)
      iday=iday+ihd*100
      seu=iceu
      seu=seu/100.+iseu
      ses=ices
      ses=ses/100.+ises
c      write(*,*)iday
c      write(*,*)iou,miu,ios,mis
c ***** compute reference data point *****
c
```

```
tstart=ios*3600.+mis*60.+ses
transit=io*3600.+min*60.+sec+siderr
if(tstart.gt.transit)tstart=tstart-86400.
tt=transit-tstart+1.
iref=tt
ref=tt-iref
iref=iref-351
write(*,*)iref,ref
c
c **** fill data array ****
c
nrec=0
1 nrec=nrec+1
call read (idat1)
if(status(1).ne.64.and.status(1).ne.65) goto 100
ipoint=(nrec)*9+1
if(ipoint.lt.iref)goto 1
id=ipoint-iref+1
c write(*,*)id
2 do 20 j=0,8
id=id+1
if(id.gt.701)goto 200
do 15 k=1,96
ida=idat(k+6+j*160)
isign=ida
if(isign.ge.0)ida=ida-32768
if(isign.lt.0)ida=ida+32768
data(k,id)=-ida
15 continue
20 continue
call read (idat1)
if(status(1).ne.64.and.status(1).ne.65) goto 1001
goto 2
200 continue
itot=701
goto 1002
1001 itot=id-1
1002 continue
c
c *** calibrate data ***
c
do 31 k=1,96
sum=0.
do 39 i=1,itot
39 sum=sum+data(k,i)
sum=sum/itot
do 31 i=1,701
31 data(k,i)=(data(k,i)-sum)/ampl(k)
do 131 k=2,96,2
delta=pi/2.-phase(k)+phase(k-1)
cosd=cos(delta)
sind=sin(delta)
do 131 i=1,701
data(k,i)=(data(k,i)-data(k-1,i)*sind)/cosd
```

```

131 continue
write(*,*)'start'
c *** compute observed amplitudes and phases ***
do 32 k=1,96,2
am(k)=0.
ph0=atan2(data(k+1,351),data(k,351))
ph1=atan2(data(k+1,352),data(k,352))
dph=(ph1-ph0)/(2.*pi)
if(dph.lt.0.)dph=dph+1.
dph=amod(dph,1.)*2.0*pi
ph(k)=ph0+dph*ref
c
c *** Note: phase is interpolated between points 101 and 102 *****
c *** 'ref' is the fractional second of expected transit *****
c
do 33 i=349,353
am(k)=am(k)+(data(k,i)**2+data(k+1,i)**2)**0.5
pha=atan2(data(k+1,i),data(k,i))
18 pd=pha-ph(k)
c if(abs(pd).lt.1.)goto 17
c if(pd.ge.1.)pha=pha-2.*3.141592
c if(pd.le.-1.)pha=pha+2.*3.141592
c goto 18
c 17 if(i.ne.101)ph(k)=ph(k)+pha
33 continue
am(k)=am(k)/(5.*flux)
32 continue
c
c *** ph(96) (solo le dispari) sono le fasi complessive degli interferomet
i *
c
write(*,*)'stop'
do 30 k=1,96,2
in1=k/12+1
if(in1.le.4)in1=5-in1
ph(k)=ph(k)-pns(in1)-phdel
c
c *** ora ph sono le fasi degli interf. depurate delle fasi di puntamento
c *** e EW-NS (radianti) *****
c
30 write(9,58)k,am(k),ph(k)
do 35 k=1,96
am(k)=am(k)*5.
ph(k)=ph(k)+phase(k)
ph(k)=amod(ph(k),twopi)
if(ph(k).lt.0.)ph(k)=ph(k)+twopi
35 ph(k)=ph(k)*180./3.141592
write(14,*)' ' 1E 2E 3E 4E 5E',
+' 6E'
do 36 k=1,96,12
kk=k/12+1
36 write(14,57)nns(kk),am(k),am(k+2),am(k+4),am(k+6),am(k+8),
+am(k+10)
write(14,*)' '

```

```

        write(14,*)'          1E      2E      3E      4E      5E',
+'          6E'
        do 339 k=1,96,12
        kk=k/12+1
339 write(14,557)nns(kk),ph(k),ph(k+2),ph(k+4),ph(k+6),ph(k+8),
+ph(k+10)
        do 301 k=1,8
        do 301 i=1,6
        kk=(k-1)*12+1+(i-1)*2
        a(i,k)=am(kk)
        p(i,k)=ph(kk)
301 continue
        do 302 k=1,8
        ans(k)=0.
        do 303 i=1,6
303 ans(k)=ans(k)+a(i,k)
302 ans(k)=ans(k)/6.
        do 306 i=1,6
        aew(i)=0.
        do 307 k=1,8
307 aew(i)=aew(i)+a(i,k)
306 aew(i)=aew(i)/8.
        call resolve(p,fns,pew)
        do 305 k=1,8
        fns(k)=amod(fns(k),360.)
        if(fns(k).lt.-180.)fns(k)=fns(k)+360.
305 if(fns(k).gt.180.)fns(k)=fns(k)-360.
        do 308 i=1,6
        pew(i)=amod(pew(i),360.)
        if(pew(i).lt.-180.)pew(i)=pew(i)+360.
308 if(pew(i).gt.180.)pew(i)=pew(i)-360.
        write(14,*)' '
        write(14,*)' '
        write(14,*)' '
        write(14,500)nns
        write(14,501)ans
        write(14,502)fns,'deg.'
        do 601 k=2,8
301 fns(k)=(fns(k)-fns(1))*734.8/360.
        fns(1)=0.
        write(14,502)fns,' mm.'
        write(14,*)' '
        write(14,*)' '
        write(14,500)new
        write(14,501)aew
        write(14,504)pew,'deg.'
        do 600 i=2,6
300 pew(i)=(pew(i)-pew(1))*734.8/360.
        pew(1)=0.
        write(14,504)pew,' mm.'

c      ncs=40
c      call pack1(sname,ncs,ncr)
c      write(*,*) ncr

```

```
c
c   write(snam,1501) sname
c 1501 format(a5)
c
cc   write(*,1501) snam
c
c   s$='\\log\cal.'//ext
c   inquire(file=s$,exist=flag)
cc   if(.not.flag) open(unit=13,file=s$)
c   if(flag) open(unit=13,file=s$,access='append')
c   write(str1,1500) day, sname
c 1500 format(' Day=',i3,5x,a8)
c
c   write(str6,507)
c 507 format(5x,'1N/4N 1S/4S (amp.e fase) 1E/6E (amp. e fase)')
c
c
cc   write(str7,508)
cc508 format(5x,'1E/6E (amp.e fase)')
c
c
c   write(str2,701) ans
c   write(str3,702) fns
c   write(str4,701) aew
c   write(str5,704) pew
c
c   write(13,*) str1
c   write(13,*) str6
c   write(13,*) str2
c   write(13,*) str3
c   write(13,*) str4
c   write(13,*) str5
701 format(1x,8f8.2,1x,a4)
702 format(1x,8f8.1,1x,a4)
704 format(1x,6f8.1,1x,a4)

ch1=12
write(14,2543) itot,ch1

2543 format(//5x,i4,2x,'data points used',a1)
500 format(4x,8(6x,a2))
501 format(5x,8f8.3,2x,a4)
502 format(5x,8f8.1,2x,a4)
504 format(5x,6f8.1,2x,a4)
557 format(5x,a3,6f8.1)
57 format(5x,a3,6f8.3)
58 format(5x,i3,2f8.4)
100 write(*,398) status
398 format(2x,'Status is ',4i3)
if(status(1).ne.64.and.status(1).ne.65) write(*,*)
+' **** ERROR IN READING TAPE ****'
stop
end
subroutine fasins (df,dg,pr)
```

```
      real dl(8),ze,dc,df(8),uno,ksign
      integer*2 val,i,iif(8),f(8)
      real dg,pr
      ze=44.57
      uno=1
      dc=dg
      dc=dc+pr/60.
      do 30 i=1,8
      dl(i)=(i-4.5)*80*sin((ze-dc)*3.141592/180.)
      dl(i)=dl(i)*408/299.793
      val=int(dl(i))
      ksign=sign(uno,dl(i))
      df(i)=ksign*(abs(dl(i))-iabs(val))
      df(i)=df(i)*2.*3.141592
30      continue
100     return
      end
      subroutine resolve(a,ar,ac)
      dimension a(6,8),b(6),c(8),aa(6,8),ar(8),ac(6)
      do 10 k=1,6
10      b(k)=a(k,1)
      do 11 i=1,8
11      c(i)=a(1,i)
      do 20 k=1,6
      do 20 i=1,8
20      aa(k,i)=a(k,i)-b(k)
      do 23 i=1,8
23      ar(i)=0.
      do 22 i=2,8
      do 22 k=1,6
111     continue
      if((aa(k,i)-aa(1,i)).gt.180.)then
          aa(k,i)=aa(k,i)-360.
          goto 111
      endif
112     continue
      if((aa(k,i)-aa(1,i)).lt.-180.)then
          aa(k,i)=aa(k,i)+360.
          goto 112
      endif
      ar(i)=ar(i)+aa(k,i)
22     continue
      do 24 i=1,8
24     ar(i)=ar(i)/6.
      do 30 k=1,6
      do 30 i=1,8
30     aa(k,i)=a(k,i)-c(i)
      do 33 k=1,6
33     ac(k)=0.
      do 32 i=1,8
      do 32 k=2,6
311    continue
      if((aa(k,i)-aa(k,1)).gt.180.)then
          aa(k,i)=aa(k,i)-360.
```



```
        goto 311
    endif
312 continue
    if((aa(k,i)-aa(k,1)).lt.-180.)then
        aa(k,i)=aa(k,i)+360.
        goto 312
    endif
    ac(k)=ac(k)+aa(k,i)
32 continue
    do 34 k=1,6
34 ac(k)=ac(k)/8.
    return
end
```

```
c      --- programma per PEW e PNS ---
c
character*3 resp
character*30 punt
integer*2 ifreq,idurr,ierr,k

ifreq=1500
idurr=5

1      open(unit=10,file='\croce\pword.dat')

write(*,*) '-----'
write(*,*) 'Pointing:  e-w (E)   n-s (N)'
write(*,*) '-----'
read(*,'(a3)') resp
if(resp.eq.'E'.or.resp.eq.'e') goto 100
if(resp.eq.'N'.or.resp.eq.'n') goto 90

call beep2 (ifreq,idurr,ierr)
write(*,*)'Invalid  Enter'
goto 1

90     punt='1 0 1 1 1'
write(10,'(a10)') punt
goto 200

100    punt='0 1 0 0 0'
write(10,'(a10)') punt

200    continue
end
```

```
$NOTRUNCATE
```

```
$DECLARE
```

```
c -----
integer*2 numcom,ibaud,ipar,iwlen,istop,irs,ics,ids,icd
* ,ilf,ierr
integer*2 movant(5),k,dd,pp
integer*2 nchr,ltctime,ntape,nfile,delay,num
integer*4 rec
integer*4 aseg,nskip
real*4 se,decl(5),decl2,decl3,dew,dns,ss,dec
character*3 resp
character*15 stri(10)
character*35 dec$
character*10 scom$
character*5 ntap$,nfil$
character*35 sfitt$,snom$,sked$
character*1 chr(80),cr,flag
```

```
num=1
call getarg(num,sked$,ierr)
```

```
c numcom=1
   ibaud=8
   ibaud=3
c 3=300 bauds, 8=9600 bauds
   ipar=0
   iwlen=7
   istop=1
   irs=0
   ics=0
   ids=0
   icd=0
   ilf=0
```

```
call iopcom(numcom,ibaud,ipar,iwlen,istop,irs,ics,ids
+,icd,ilf,ierr)
if(ierr.ne.0)then
  write(*,*)'Error in opening RS232 ',ierr
  stop
endif
```

```
c -----
write(*,*)'Dec. from Keyboard (K), from File (F)'
read(*,'(a3)') resp
if(resp.eq.'F'.or.resp.eq.'f') go to 20
```

```
c ---- dec. from Keyboard ----
write(*,*)'Enter Dec. (dd pp )'
read(*,*) dd,pp
dec=dd+pp/60.
do k=1,5
  decl(k)=dec
```

```
        enddo
        write(*,*)'Enter ndec for five command'
        write(*,*)'ker e-w n-s dly phs (1=do, 0=do not)'
        read(*,*) movant
        goto 100

c      ---- dec. from File ----
20     open(unit=10,file='\croce\decnew.dat')
21     open(unit=11,file='\croce\pword.dat')

c      do k=1,4
c      read(10,'(a12)') stri(k)
c      enddo

29     read(10,29) dd,pp,ss
        format(23x,i2.2,1x,i2.2,1x,f6.2)

c      read(stri(3),35) dd,pp,ss

c35    format(i2.2,1x,i2.2,1x,f6.3)
        write(*,112) dd,pp,ss
112    format(1x,i2.2,1x,i2.2,1x,f6.3)

        dec=dd +pp/60.+ss/3600.
        do k=1,5
        decl(k)=dec
        enddo
36     read(11,36) movant
        format(i1,1x,i1,1x,i1,1x,i1,1x,i1)

100    continue
c      ---- send informations ----
c      ---- to pointing PC ----
c      pointing informations sent are:
c
c      decl(1)=NS Kerosene pointing
c      decl(2)=EW mechanical pointing
c      decl(3)=NS mechanical pointing
c      decl(4)=NS delays
c      decl(5)=NS phase shifters (ZENITH)

        decl2=decl(2)
        decl3=decl(3)
        write(*,*) decl2
        write(*,*) decl3

c      =====
c      398 call pcpoint(decl,flag,movant)
c      =====

        dew=abs(decl(2)-decl2)
```

```
dns=abs(decl(3)-decl3)
```

```
c      ---- flag 1/9 ----
      if (flag.eq.'1'.or.
:         flag.eq.'2'.or.
:         flag.eq.'3'.or.
:         flag.eq.'4'.or.
:         flag.eq.'5'.or.
:         flag.eq.'6'.or.
:         flag.eq.'7')then
          write (*,*) 'Irrecoverable Error ', flag

      if (flag.eq.'B') then
          write(*,*) 'ANTENNA UNDER PULSAR COMPUTER CONTROL'
          write(*,*) '
          write(*,*) 'NOT POSSIBLE TO HAVE RIGTH POSITION'
          write(*,*) '-----'
          write(*,*) '
          write(*,*) 'waiting for 1 min. before try again'
          endif

          delay=60
          call wait(delay)

c      ---- " -Ret- key control ----
c401   call inkey (rinp,key,in)
c      nchr=1
c      if (key.ne.28) go to 401
          go to 398
          endif

      if (flag.eq.'A') then
          write(*,*) 'Pointing Computer not allocable'
          call pcend(flag)
          if(flag.ne.'0') then
              write(*,*) 'Failure to end properly ',flag
          endif
          endif

c      ---- flag=A ----
      if(flag.eq.'A'.or.flag.eq.'0') then
          if(dew.le.0.4.and.dns.le.0.20) goto 995

          write(*,*) 'ANTENNA UNDER PULSAR COMPUTER CONTROL'
          write(*,*) '
          write(*,*) 'NOT POSSIBLE TO HAVE RIGTH POSITION'
          write(*,*) '-----'
          write(*,*) '
          write(*,*) 'waiting for 1 min. before try again'

          delay=60
          call wait(delay)
```

```
c      ---- " -Ret- key control ----
c399   call inkey (rinp,key,in)
c      nchr=1
c      if (key.ne.28) goto 399
c      call pcend(flag)
c      endif
c      *****
995   stop
      end
```

\$DECLARE

```
C      =====  
C  
C      REWIND THE TAPE UNIT  
C  
C      =====
```

```
integer*1 b1,b2,b3,b4  
integer*2 address,level  
integer*2 irow,icolm,ivpage,ifreq,idurr,in
```

```
C      -----  
C      PC488 interface initializing  
C      address=1  
C      level=0  
C      call initia (address,level)  
C      -----
```

```
ifreq=5000  
idurr=3  
call beep2(ifreq,idurr,in)  
irow=15  
icolm=20  
ivpage=0  
write (*,*)'**** TAPE UNIT IS REWINDING.....WAIT ! ****'  
call rew (b1,b2)  
call decode(b1,b2,b3,b4)  
stop ' '  
end
```

```
$DECLARE
```

```

c      =====
c
c      SKIP qN FILES ON THE TAPE UNIT
c
c      =====

c      This program has to be linked with:
c      --- FORT488    --> IEEE488 Routines
c      --- TAPEC      --> Tape Routines
c      --- RDHEAD     --> Header Reading Routine
c      --- NOLIMIT    --> Libraries
c
c      -----
c      integer*1 b1,b2,b3,b4,resp
c      integer*2 ifreq,idurr,in,na,istat
c      character*20 command
c      integer*2 ii,i,address,level
c      integer*4 nargs
c
c      -----
c      PC488 interface initializing
c      address=1
c      level=0
c      call initia (address,level)
c
c      -----
c
c      ifreq=2500
c      idurr=1
c      write(*,*)' '
c      call beep2(ifreq,idurr,in)
c      if(nargs().eq.2)then
c          na=1
c          call getarg(na,command,istat)
c          write(*,*)command
c          read(command,'(i4)')resp
c          go to 333
c      endif
c      write (*,*)'How many Files? (qn, + is forward, - is backward)'
c      read (*,*) resp
333   if(resp.lt.0)then
c          resp=-resp
c          goto 5800
c      endif
c      if(resp.eq.0)goto 5000
c      do 5610 i=1,resp
c          call fsfwhighsp (b1,b2)
c          call decode (b1,b2,b3,b4)
c          if(b1.ne.65)stop
c          ifreq=2500
c          idurr=1
c          write(*,'(1h+,i5)')i
c          call beep2(ifreq,idurr,in)
5610  continue
c          go to 5000

```



```
c      ---- call file search rev. high speed ----
5800  continue
      ifreq=2500
      idurr=1
      call beep2(ifreq,idurr,in)
      do 5810 i=1,resp
      call fsrhighsp (b1,b2)
      call decode (b1,b2,b3,b4)
      if(b1.ne.65)stop
      ifreq=2500
      idurr=1
      ii=-i
      write(*,'(1h+,i4)')ii
      call beep2(ifreq,idurr,in)
5810  continue
5000  stop '***** done ***** '
      end
```

\$DECLARE

```
integer*2 val,adr
val=0
adr=771
call output(adr,val)
end
```

```
$DECLARE
```

```

c      -----PROGRAMMA DI LETTURA DELLA PROVA 'E'-----
c
c      Nome=Prove
c
integer*2 t1,t2,t3,t4,t5,t6,i,ik,icgu,idgu,iggu,iday
integer*2 ival(14),a1,a2,a3,a4,r,ifreq,idurr,ierror
real*4 iti,ite,val(14),v(14),rh
character*5 ind(14)
character*24 titolo0
character*45 titolo1
character*55 temp
character*50 intest,chdata
character*80 titolo2,titolo3,titolo4

ind(1)='1E'
ind(2)='2E'
ind(3)='3E'
ind(4)='4E'
ind(5)='5E'
ind(6)='6E'
ind(7)='1S'
ind(8)='2S'
ind(9)='3S'
ind(10)='4S'
ind(11)='1N'
ind(12)='2N'
ind(13)='3N'
ind(14)='4N'
ifreq=1200
idurr=8

c      ---- misura della temperatura Int. e Ext. ----
a1=150
a2=151
a4=152

c      -temp.int-
call conv(a1,a3)
iti=a3*0.00305175

c      -temp.ext-
call conv(a2,a3)
ite=a3*0.00305175

c      -umid. real.-
c      call conv(a4,a3)
c      rh=a3*0.0305175

1      write(temp,1) iti,ite
      format(' Temp.Int=',f4.1,' C',2x,'Temp.Ext.=',f5.1,' C',2x,
:'Rh=',f5.1,'%')

      write(*,*) 'Lettura Prova -E- dei 14 canali'

```

```

write(*,2) 'LSD=microWatt/100'
write(titolo0,2) 'LSD=microWatt/100'
2 format(1x,a22)
write(*,1) iti,ite,rh
write(*,*)' '
c --- ciclo di lettura canali (224-237)
do 100 i=224,237
ik=i-223
v(ik)=0

do 111 r=1,200
call conv(i,ival(ik))
v(ik)=v(ik)+ival(ik)
111 continue
val(ik)=(v(ik)/200)*0.305175/2

if(val(ik).le.200) then
do r=1,9
call beep2(ufreq,idurr,ierror)
enddo
endif

if(val(ik).ge.450) then
do r=1,9
call beep2(ufreq,idurr,ierror)
enddo
endif

write(*,55) ind(ik),val(ik)
55 format(2x,a3,'=',f4.0)

if(ik.eq.6) then
write(*,*)' '
endif
if(ik.eq.10) then
write(*,*)' '
endif

100 continue

c ---- lettura della data ----
call uclock(0,t1,t2,t3,t4,t5,t6)
icgu=t1/16
iggu=t2/16
idgu=t2-iggu*16
iday=(icgu*100)+(idgu*10)+iggu
10 write(chdata,10)iday
format(i3.3)
intest='b:pre.'//chdata

open(unit=10,file=intest)
write(titolol,12) iday
12 format(' Prova E:', ' day=',i3.3)
13 format(a40)
```

```
        write(titolo2,14) val(1),val(2),val(3),val(4),val(5),
:val(6)
14      format(' 1E\6E   ',f4.0,1x,f4.0,1x,f4.0,1x,f4.0,1x,f4.0,1x,f4.0)

        write(titolo3,16) val(7),val(8),val(9),val(10)
16      format(' 1S\4S   ',f4.0,1x,f4.0,1x,f4.0,1x,f4.0)

        write(titolo4,18) val(11),val(12),val(13),val(14)
18      format(' 1N\4N   ',f4.0,1x,f4.0,1x,f4.0,1x,f4.0)

c      ----scrittura su disco----
        write(10,22)titolo1
        write(10,22)temp
        write(10,22)'      '
        write(10,2) titolo0
        write(10,20)titolo2
        write(10,20)titolo3
        write(10,20)titolo4
22      format(a55)
20      format(a80)

c      --- stampa ----
        open(unit=1,file='prn')
        write(titolo1,112) iday
112     format(' Prova -E- :', ' day=',i3.3)

        write(titolo2,114) val(1),val(2),val(3),val(4),val(5),
:val(6)
114     format(' 1E\6E   ',f4.0,1x,f4.0,1x,f4.0,1x,f4.0,1x,f4.0,1x,f4.0)

        write(titolo3,116) val(7),val(8),val(9),val(10)
116     format(' 1S\4S   ',f4.0,1x,f4.0,1x,f4.0,1x,f4.0)

        write(titolo4,118) val(11),val(12),val(13),val(14)
118     format(' 1N\4N   ',f4.0,1x,f4.0,1x,f4.0,1x,f4.0)

c      write(1,122)titolo1
        write(1,122)temp
        write(1,122)'      '
        write(1,2) titolo0
        write(1,22)'      '
        write(1,120)titolo2
        write(1,120)titolo3
        write(1,120)titolo4
        write(1,125)
122     format(a55)
120     format(a80)
124     format(1h1)
125     format(/1x)
        end
```



```
c      programma di attesa per cambio nastro nel
c      osserv.bat
      integer*2  ifreq,idurr,ier,i

      ifreq=800
      idurr=1

      call cls
      do 100 i=0,9
      ifreq=ifreq+(i*80)
      idurr=idurr+(i*2)
      call beep2(ifreq,idurr,ier)
100    continue

      write(*,*) 'MESSAGE TO OPERATOR:'
      write(*,*) '-----'
      write(*,*) ' '
      ifreq=800
      idurr=5
      call beep2(ifreq,idurr,ier)
      write(*,*) '-----'
      write(*,*) ' '
      write(*,*) '  CHANGE TAPE  '
      write(*,*) ' '
      write(*,*) '-----'
      write(*,*) ' '
      write(*,*) ' '
      write(*,*) 'Hit a key when ready '
      write(*,*) ' '
      pause
      call cls

      call beep2(ifreq,idurr,ier)
      write(*,*) 'PAY ATTENTION !'
      write(*,*) 'BE SURE OF THE FOLLOWING STATEMENTS: '
      write(*,*) ' '
      write(*,*) '-CALI.SUR- HAS TO BE IN C:\Croce '
      write(*,*) '-----'
      write(*,*) ' '
      write(*,*) '-SKED.SUR- HAS TO BE IN C:\Croce '
      write(*,*) '-----'
      write(*,*) ' '
      write(*,*) 'LOG-DISK HAS TO BE IN DRIVE A: '
      write(*,*) '-----'
      write(*,*) ' '
      write(*,*) ' '
      write(*,*) 'Hit a key when ready '
      write(*,*) ' '
      pause
      call beep2(ifreq,idurr,ier)
      end
```

```
c      --- programma di input della delta e commenti ---
c      --- per OSSERVA.BAT---
character*40 str,chdata,intest
character*40 com,stringa(10),ntape
integer*2 gg,pp,icgu,idgu,iggu,iday
integer*1 t1,t2,t3,t4,t5,t6
real*4 ss
write(*,*)'Enter DECLINATION ( gg mm ss.ss ) '
10    read(*,10)str
      format(a40)
      open(unit=10,file='\croce\strip.sur',status='old')
      rewind(10)
      do k=1,10
      read(10,'(a40)')stringa(k)
      enddo
      stringa(5)=str
      write(*,*)'Enter COMMENTS (40 char. max) '
      read(*,'(a40)') com
      write(stringa(7),'(a40)') com
      rewind (10)
      do k=1,10
      write(10,10)stringa(k)
      enddo

c      ---- comments and are written on the B: LOG DISK ----
c      ---- file: comm.XXX      XXX-->day ----
c      ---- reading of day ----
call uclock(0,t1,t2,t3,t4,t5,t6)
icgu=t1/16
iggu=t2/16
idgu=t2-iggu*16
iday=(icgu*100)+(idgu*10)+iggu

311   write(chdata,311)iday
      format(i3.3)
      intest='b:comm.'//chdata
      open(unit=13,file='\croce\nast.dat')
      open(unit=15,file=intest)
      read(13,'(a10)') ntape
      write(15,312) chdata,ntape
312   format(' Day=',a6,1x,'Tape N=',a8)
      write(15,500) stringa(7)
500   format(a40)
c      -----
      stop
      end
```