MODULI SOFTWARE PER IL CONTROLLO DELL'HARDWARE DELLA "CROCE DEL NORD"

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

Rapporto interno IRA 159/92

word file: moduli.doc

INDICE

INTRODUZIONE

In questa nota tecnica interna vengono descritti tutti i moduli software, scritti per un rapido controllo della funzionalita` dei blocchi hardware che costituiscono l'acquisizione dati del Radiotelescopio 'Croce del Nord'.

Come accennato in altre note tecniche, il sistema di acquisizione dati del radiotelescopio 'Croce del Nord' e` costituito da vari blocchi funzionali, come riportato nello schema di fig.1. Da questo si nota come tale sistema sia composto da un certo numero di sottosistemi opportunamente collegati e gestiti via calcolatore tramite programmi di acquisizione, controllo e procedure DOS, ampiamente descritti in altri rapporti interni. Vista la estrema facilita` di 'accesso' da parte del Personal IBM XT di gestione, ad ogni blocco che lo costituisce, e` stato scritto un set di moduli software per effettuarne il controllo off-line, qualora se ne presentasse la necessita`.

Oltre che rispondere ad una esigenza di descrivere il software sopraddetto, queste note vogliono anche essere una guida pratica all'utilizzo dei vari programmi in questione, in fase di controllo, messa a punto e manutenzione hardware del sistema. Viene data una breve ma esauriente descrizione di ogni modulo, facendola poi seguire dal relativo listato; tutti i programmi sono stati scritti in Fortran e linkati con la libreria 'Croce' contenente tutte le subroutines necessarie (vedere appendice). Si parte con una miscellanea di moduletti dedicati alla acquisizione (multiplexer & convertitore A/D), seguita da una relativa agli orologi di stazione

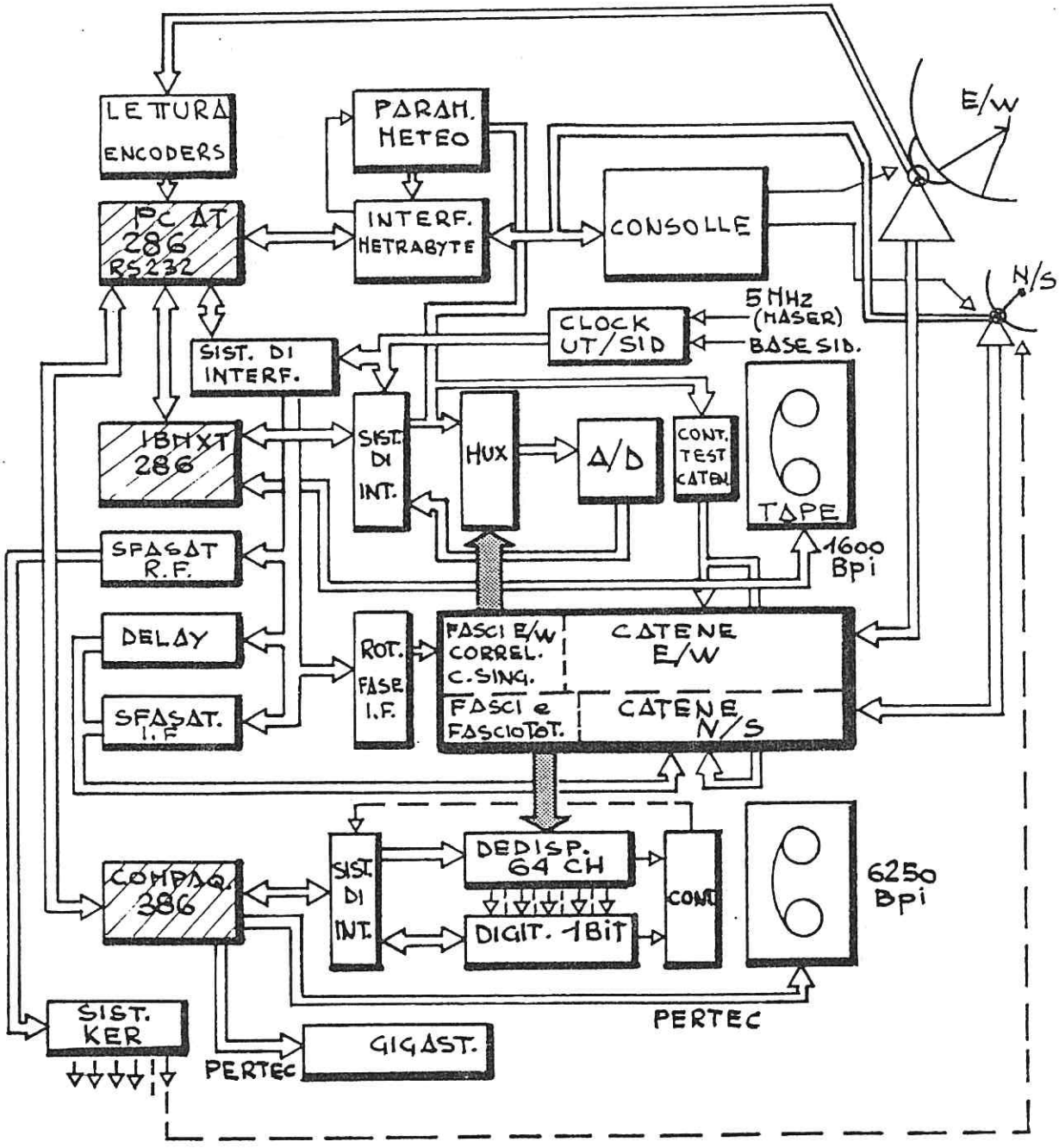
LETTURA ENCODERS
PARAM. METEO
IPC AT 286 RS232
INTERF. METRABYTE
CONSOLLE
E/W
N/S
5 MHz (MASER)
CLOCK UT/SID
BASE SID.
SIST. DI INTERF.
IBMXT 286
SIST. DI INT.
MUX
A/D
CONT. TEST CATEN
TAPE
1600 Bpi
SFASAT R.F.
DELAY
SFASAT. I.F.
ROT. FASE I.F.
FASCI E/W CORREL. C.SING.
CATENE E/W
FASCI e FASCIOTOT.
CATENE N/S
COMPAQ 386
SIST. DI INT.
DEDISP. 64 CH
DIGIT. 1BIT
CONT
6250 Bpi
PERTEC
SIST. KER
PERTEC
GIGAST.

FIG. 1

ELENCO PROGRAMMI

I programmi in oggetto son situati nel direttorio \service e sono i seguenti:

- MUX
- LEGGI
- LETAD
- LET
- BLOCK
- LETSUC
- LETCLOCK
- LETSCLOC
- SIDTIME
- RELE
- RESET
- ETOT
- ROTANS
- ROTMAN
- ROTTEMP
- ZENITH

-MUX-

Visualizza sui due video (di consolle e di servizio), l'output della coppia di correlatori cos/sen scelti, in unita` di codificatore.

In fase di controllo e/o manutenzione dei correlatori, puo` essere molto utile potere visualizzare su due video, quello del personal di gestione e quello di servizio posto in prossimita` dei correlatori e collegato via RS232, l'uscita della coppia di correlatori scelta. Una volta lanciato, il programma chiede in ingresso quale 'coppia' cos/sen si vuole prendere in considerazione, e ne visualizza l'output in unita` di codificatore. Questo permette di effettuare varie tarature, tipo regolazione dell'isolamento e dell'offset, agendo direttamente sul correlatore leggendone l'uscita sul monitor di servizio. Da ricordare che il numero da entrare per considerare la coppia complessa desiderata, e` quello relativo al cos. I numeri relativi alle coppie cos/sen costituenti un determinato correlatore complesso, si possono dedurre dalla lista dei canali di ingresso dell'acquisizione dati, riportati in appendice. Per uscire bisogna battere Ctrl-C. Nelle pagine che seguono, viene riportato il listato del programma.

```
$DECLARE
$NOTRUNCATE

c                - MUX -

c       il programma e` linkato con
c       -FORT488
c       -CONV
c       -WAIT
c       -OUTPORT
c
      integer*2 numcom,ibaud,ipar,istop,iwlen,iscom,ismod,ierror
      integer*2 nmax,nact,iech,idsr,ics,irts,itype
      character*8 stg
        integer*2 nc,icp,i,jp,j,keycde,nchr,delay,fondo,ns,ew
        integer*2 ab1,ab2,k,irow,icolm,ivpage,ic1,ic2,dat1,dat2
        integer*4 iseg
        real*4 da,cp,dato,ang(20),icc
        character*1 chr(80),chaar,ret
        character*80 out
      character*30 out1
        equivalence(out,chr(1))
      ret=char(13)
       numcom=2
      ibaud=6
      ipar=1
      istop=1
      iwlen=8
      nmax=30
      nact=30
      iech=1
      idsr=0
      ics=0
      irts=0
      itype=1
c      write(*,*)'enter ibaud,ipar,istop,iwlen'
c      read(*,*)ibaud,ipar,istop,iwlen
      call opncom(numcom,ibaud,ipar,istop,iwlen,iscom,ismod,ierror)
        iseg=45056
        do 3312 i=1,20
 3312       ang(i)=i
        call cls
  210  continue
c       ---- canale da controllare ----
        call cls
        write(*,*) 'numero canale ? (^C per uscire)'
        read (*,*) cp
        write(*,*) 'battere -C- per cambiare canale'
        cp=int (cp)
        icp1=cp
        ic1=icp1-1
        ic2=icp1
        icp2=icp1+1
```

```
 100    continue
 105    continue
c       ---- lettura ----
        call conv (ic1,dat1)
        call conv(ic2,dat2)
        call pausa(20)
 108    continue
        write (out,110) icp1,dat1,icp2,dat2
        write(out1,1119)icp1,dat1,icp2,dat2,ret
1119    format(i4,i7,i4,i7,a1)
        call inocom(numcom,out1,nmax,nact,iech,idsr,ics,irts,itype,
     +iscom,ierror)
 110    format (10x,i4,i7,i4,i7)
        j=0
        do 200 i=1280,1438,2
        j=j+1
        jp=ichar(chr(j))
        call poke(iseg,i,jp)
        call inkey (chaar,keycde,nchr)
        if(nchr.eq.1.and.chaar.eq.'E')stop
        if (nchr.eq.1.and.chaar.eq.'C') go to 210
        if(nchr.eq.1.and.chaar.eq.'e')stop
        if (nchr.eq.1.and.chaar.eq.'c') go to 210
 200    continue
        go to 105
        end
```

-LEGGI-

Legge e visualizza l'ampiezza del segnale di un canale, annotando quante volte e` uscito dalla finestra di allarme prefissata.

Il programma Leggi legge e visualizza in continuazione l'output di un canale scelto (mV), offrendo la possibilita` di stabilirne a priori il sampling-time in centesimi di secondo e la soglia (mV), definendo la fascia attorno alla media, entro la quale il segnale di quel canale deve rimanere per essere accettabile. Tutte le volte che viene superato l'intervallo prefissato, incrementa il valore di un contatore e genera un allarme acustico, ogni lettura vengono visualizzati numero di canale, valore della lettura, valore medio delle prime 19 letture e valore del contatore. Per terminare bisogna battere Ctrl-C. Questo modulo puo` essere utile in molte situazioni in cui sia necessario tenere sotto controllo l'uscita di un determinato canale in presenza di instabilita` o interferenze. Nelle pagine che seguono, viene riportato il listato.

```
      dimension val(19)
      integer*2 ktime,kk,irow,icol,ivp,ier
      irow=3000
      icol=2
      ivp=0
      write(*,*)' '
      call cls
c     5   call putcur(irow,icol,ivp,ier)
      write(*,*)'Reading in mV                          '
      write(*,*)'                                       '
      write(*,*)'enter starting channel (from 0...)'
      read(*,*)kstart
      write(*,*)'enter sampling interval (1/100 sec.units)'
      read(*,*)ktime
      write(*,*)'enter max. difference (10 mV or less)'
      read(*,*)diff
      kk=kstart
      do 10 k=1,19
      call pausa(ktime)
10      call legge(kk,val(k))
      vmed=0.
      do k=1,19
      vmed=vmed+val(k)
      enddo
      vmed=vmed/19.
    5 continue
      call legge(kk,val(2))
      if(abs(vmed-val(2)).gt.diff)then
      ierr=ierr+1
      call beep2(irow,icol,ivp)
      endif
      write(*,'(2x,i3,8x,2f8.1,2x,i6)')kk,val(2),vmed,ierr
      call pausa(ktime)
      goto 5
      end
c     SUBROUTINE DI LETTURA DI UN CANALE
c     DELL'ACQUISIZIONE DATI DEL RADIOTELESCOPIO
c     'CROCE DEL NORD"
c     NOME DELLA SUBROUTINE:LEGGE
c
      subroutine legge(b1,b2)
c     b1=canale da leggere (INTEGER*2)
c     b2=valore letto (REAL*4)
      integer*2 amux,sconv,flag,w,r,let1,adr1,adr2,adr3,val,can
      integer*2 let2 ,z1,z2,b1,ida
      real b2
      can=b1
      adr1=768
      adr2=769
      adr3=770
      amux=17
      w=0
      r=1
      sconv=16
```

```
        let1=16
        let2=17
        flag=18
        can=b1
 10     call output (adr2,amux)
        call output (adr1,can)
        call output (adr3,w)
        call output (adr3,r)
c  delay per il mux (500 microsec)
        ang=23.
        icc=cos(ang)
c       ----------------------
        call output (adr2,sconv)
        val=1
        call output (adr1,val)
        call output (adr3,w)
        call output (adr3,r)
c       ----------------------
 20     call output (adr2,flag)
        val=input (adr1)
        if (val.ne.0) go to 20
c       ----------------------
 30     call output (adr2,let1)
        z1=input (adr1)
        call output (adr2,let2)
        z2=input (adr1)
        ida=z1+(z2*256)
        isign=ida
c        if(isign.ge.0)ida=ida-32768
c        if(isign.lt.0)ida=ida+32768
        ida=ida-32768
        b2=ida
        b2=b2*0.305175
c       write (*,*)can, red
        return
        end
```

-LETAD-

Legge una sola volta tutti i canali collegati alla acquisizione dati (1-155) e li visualizza su schermo

Il programma in questione e` quanto di piu` semplice ed immediato si possa concepire; infatti legge e stampa su video il risultato (mV) della conversione di tutti i canali collegati al telescopio. Da` la possibilita` di rendersi conto immediatamente se multiplexer e/o convertitore analogico-digitale hanno problemi. Allo scopo di faciltare la lettura dei valori che scorrono sul monitor, si puo' utilizzare il tasto Pause come stop/start. Segue il listato.

```
$DECLARE
c       programma che legge e stampa l'out dei canali
c       dell`acquisizione dati voluti

        integer*2 i,istart,istop,b2
        real*4 dat

 1       do 100 i=0,154
        call conv(i,b2)
        dat=b2*0.00030516
        write(*,*)'canale',i+1,dat
 100    continue
        pause
        go to 1
        end
```

Legge 21 canali e li visualizza su schermo

Programma che puo` essere usato per visualizzare in continuazione su schermo, 21 canali (quanti ne stanno in colonna in una 'videata') della acquisizione, il cui numero di partenza puo` essere scelto all'inizio del programma. Per terminare battere ctrl-C. Segue il listato.

```
$DECLARE
c       programma che legge e stampa l'out dei canali
c       dell`acquisizione dati voluti

        integer*2 i,istart,istop,irow,icolm,ivpage,ierr,b2
        real*4 dat
        irow=1
        icolm=1
        ivpage=0

 1      call cls
        write(*,*)'Lettura e stampa di un certo numero di canali (mV)'
        write(*,*)'impostare canale di Start e Stop'
        write(*,*)'  '
        write(*,*)'numero dei canali di start e stop (st stp)?'
        write(*,*)'max numero canali=21'
        write(*,*)'-^C per terminare-'
        read(*,*) istart,istop

        if((istop-istart).gt.21.or.istart.gt.istop)then
         write(*,*)'numero canali NON valido -CONTROLLARE-'
         pause
         go to 1
        endif

        call cls
        call putcur(irow,icolm,ivpage,ierr)
        write(*,*)' '
 10     call putcur(irow,icolm,ivpage,ierr)
        write(*,*)'-^C per terminare-'
        write(*,*)'------------------'
         do 100 i=istart-1,istop-1
        call conv(i,b2)
        dat=b2*0.000305176
        write(*,50)i+1,dat
 50     format(' canali ',i3,f9.3)
 100    continue
        write(*,*)'  '
        go to 10
        end
```

-BLOCK-

Controlla 16 canali dell'acquisizione, afferenti alla stessa scheda del multiplexer

I 155 canali dell'acquisizione dati sono collegati al convertitore A/D, tramite un multiplexer composto da 16 schede con 16 canali ciascuna, selezionate una per volta da una scheda 'principale' a sua volta a 16 canali (submultiplexer); i primi 16 canali sono cosi` contenuti nella prima scheda, i 'secondi' 16 canali nella seconda e cosi` via fino a 155 (max. 256). Puo` quindi essere utile per controllare i 16 switches CMOS del submultiplexer o di una determinata scheda del multiplexer, potere visualizzare le uscite dei relativi canali su schermo. Per fare cio` e` sufficiente, una volta lanciato il programma, entrare il numero di scheda che si vuole controllare (1/16). Segue il listato.

```
$DECLARE
c       programma che legge e stampa i 16 canali di una scheda MUX
c       dell`acquisizione dati voluti

        integer*2 i,istart,ibloc,irow,icolm,ivpage,ierr,b2,key
        real*4 dat
        character*1 chr
        irow=1
        icolm=1
        ivpage=0

 1      call cls
        write(*,*)'Lettura e Stampa di una Scheda MUX (16 ch -mV-)'
        write(*,*)'impostare numero del blocco-MUX (1-16)'
        write(*,*)'  '
        write(*,*)'numero del blocco?'
        write(*,*)'-ESC- per cambiare blocco'
        write(*,*)'-^C per terminare-'
        read(*,*) ibloc

        if(ibloc.le.0.or.ibloc.gt.16)then
         write(*,*)'HO DETTO 1-16 CAPITO!!! '
         pause
         go to 1
        endif

        call cls
        call putcur(irow,icolm,ivpage,ierr)
        write(*,*)' '
        istart=ibloc-1
 10     call putcur(irow,icolm,ivpage,ierr)
        write(*,*)'BLOCCO',ibloc
c       write(*,*)'numero della scheda? '
        write(*,*)'-ESC- per cambiare blocco'
        write(*,*)'-^C per terminare-'
        write(*,*)'------------------'
        dat=0
         do 100 i=(istart*16),(istart*16+15)
        do c=1,100
        call conv(i,b2)
        dat=dat+b2
        enddo
        dat=dat/100
        dat=b2*0.305175
        write(*,50)i+1,dat
 50     format(' canali ',i3,4x,f8.1)

 100    continue
        write(*,*)'  '
c       -- controllo del tasto ESC ---
        call inkey(chr,key,ierr)
        if(key.eq.1) go to 1
        go to 10
        end
```

-LETSUC-

Legge e stampa su video l'ora degli orologi UT e Siderale, di stazione

Questo programma e` da usarsi per un controllo immediato dell'hardware relativo agli orologi UT e Siderale di stazione e relative interfacce. Il congelamento delle cifre per le letture delle due 'ore', avviene sul PPS siderale e sul centesimo di secondo UT immediatamente successivo. Il risultato della lettura viene visualizzato su schermo, assieme alla data corrente, ricavata dalla lettura dell'orologio UT, per terminare bisogna battere Ctrl-C.

-LETCLOCK-

Legge e visualizza su schermo data ed ora UT.

Viene visualizzata su schermo ora e data corrente UT, per terminare bisogna battere Ctrl-C. Questo modulo, puo` risultare utile in fase di controllo dell'hardware dell'orologio UT.

-LETSCLOC-

Legge e visualizza su schermo data ed ora Siderale.

Viene visualizzata su schermo ora e data corrente Siderale, per terminare bisogna battere Ctrl-C. Questo modulo, puo` risultare utile in fase di controllo dell'hardware relativo all'orologio siderale.

-SIDTIME-

Controlla l'errore dell'orologio siderale rispetto l'orologio UT.

Il programma SIDTIME controlla e visualizza l'errore dell'orologio siderale rispetto all'orologio UT (molto stabile avento come base dei tempi il Hmaser), calcola e visualizza la variazione dell'errore dall'ultima volta in cui e` stato lanciato il programma, inoltre se l'errore e superiore a 0.2 secondi genera un allarme acustico. L'errore che include la costante di integrazione dei correlatori (1 sec), viene memorizzato nel file \croce\sider.err.

```
$NOTRUNCATE
$DECLARE

c       *********READING UT & SID CLOCKS **********

c                   prog. name: letsuc

        integer*2 amux,adr1,adr2,adr3,w,r,in
        integer*2 scad,input,inad1,inad2,iciclo,i,ind,is
        integer*2 ncan,val,wr,statad,adclk,a,b,ik
        integer*1 fs,mo(12),im,b1,b2,b3,b4
        integer*1 a1,a2,a3,a4,a5,an,gm,d(4)
        integer*2 sd,day,ifreq,idurr,ier,a6,key
        integer*2 cgu,dgu,ggu,dhu,hhu,dmu,mmu,dsu,ssu,dcs,ucs
        integer*2 cgs,dgs,ggs,dhs,hhs,dms,mms,dss,sss
        integer*2 rcg,rgg,rhh,rmm,rss,rcs,ltctime,sync,bis,biss
        logical pps
        real*4 icc,ang
        character*8 string,cr
        data mo/31,28,31,30,31,30,31,31,30,31,30,31/
        ifreq=1900
        idurr=1
        bis=int(an/4)
        biss=an-bis*4
        if(biss.eq.0) then
        data mo/31,29,31,30,31,30,31,31,30,31,30,31/
        endif

        call gdate(string)
        read(string,'(6x,i2)')an


        amux=17
        adr1=768
        adr2=769
        adr3=770
        w=0
        r=1
        wr=0
        scad=16
        inad1=16
        inad2=17
        statad=18
        adclk=54
        rcg=48
        rgg=49
        rhh=50
        rmm=51
        rss=52
        rcs=53
        ltctime=48
        fs=0
        call cls
```

```
        write(*,*)'PROGRAMMA CHE LEGGE ORA SIDERALE  E UT'
        write(*,*)'SINCRONIZZANDOSI SUL PPS SIDERALE       '
        write(*,*)'     - ESC -  per Uscire                 '
c       ----read condition----
        call output (adr3,r)
c       ----sync con il SID. PPS----
  100   call output(adr2,adclk)
        fs=input(adr1)
        pps=btest(fs,0)
        if(pps) go to 100
  200   call output(adr2,adclk)
        fs=input (adr1)
        pps=btest(fs,0)
        if (.not.pps) go to 200
c       ----sid. data/clock latching----
        call output (adr2,ltctime)
        call output (adr3,w)
c       ----ut time latching and reading----
        sync=7
        call uclock(sync,a1,a2,a3,a4,a5,a6)
        call output (adr3,r)
        call beep2(ifreq,idurr,ier)
c       ----reading sid.clock----
        call output (adr2,rhh)
        d(1)= input (adr1)
        call output (adr2,rmm)
        d(2)=input (adr1)
        call output (adr2,rss)
        d(3)= input (adr1)
        call output (adr2,rcs)
        d(4)= input (adr1)
c       ----sid.clock nibble swooping----
        d(1)= ishc(d(1),4)
        d(2)= ishc(d(2),4)
        d(3)= ishc(d(3),4)
        d(4)= ishc(d(4),4)


        dhu=d(1)/16
        hhu=d(1)-dhu*16
        dmu=d(2)/16
        mmu=d(2)-dmu*16
        dsu=d(3)/16
        ssu=d(3)-dsu*16
        dss=d(4)/16.
        sss=d(4)-dss*16.

c       ---- SID clock display ----
        call putcur(12,1,0,in)
        write (*,299) dhu,hhu,dmu,mmu,dsu,ssu,dss,sss
  299   format('+','SIDERALE',28x,2i1,':',2i1 ,':',2i1,'.',2i1)

c       ----UT clock display----
        cgu=a1/16
```

```
        ggu=a2/16
        dgu=a2-ggu*16
        day=(cgu*100)+(dgu*10)+ggu
        dhu=a3/16
        hhu=a3-dhu*16
        dmu=a4/16
        mmu=a4-dmu*16
        dsu=a5/16
        ssu=a5-dsu*16
        dcs=a6/16
        ucs=a6-dcs*16

        sd=337+mo(2)
        do 290 im=12,1,-1
        sd=sd-mo(im)
        if (day.gt.sd) go to 295
 290    continue
 295    gm=day-sd
        call putcur(13,1,0,in)
        write (*,300) gm,im,an,dhu,hhu,dmu,mmu,dsu,ssu,dcs,ucs
 300    format('+','UT',24x,i2,'/',i2,'/',i2,'  ',2i1,':',2i1 ,':',2i1,
    &'.',2i1)
        call inkey(cr,key,ier)
        if(key.eq.1) goto 500
        go to 100
 500    stop'programm end'
        end
```

```
$DECLARE
c       *********CLOCK CONTROL*********
c                Prog.Name:LETCLOCK

        integer*2 adr1,adr2,adr3,w,r,i,input
        integer*2 adclk,d(2),wr,ier,key,irow,icolm,ivpage
        integer*1 c(10),fs,mo(12),t1,t2,t3,t4,t5,t6,t7,t8,t9
        integer*2 sd,day,hhs,mms,sss,css,cgu,dgu,ggu,bis,biss
        integer*1  gm,im,an,dhu,hhu,dmu,mmu,dsu,ssu,dcu,ucu
        integer*2 rcg,rgg,rhh,rmm,rss,rcs,ltctime,j
        logical pps,of
        character*8 string,cr

        data mo/31,28,31,30,31,30,31,31,30,31,30,31/
        call gdate(string)
        read(string,'(6x,i2)') an
        bis=int(an/4)
        biss=an-bis*4
        if(biss.eq.0) then
        data mo/31,29,31,30,31,30,31,31,30,31,30,31/
        endif


        call cls
        adr1=768
        adr2=769
        adr3=770
        w=0
        r=1
        wr=0
        adclk=38
        rcg=32
        rgg=33
        rhh=34
        rmm=35
        rss=36
        rcs=37
        ltctime=32
        fs=0
        irow=15
        icolm=5
        ivpage=0
        call putcur(irow,icolm,ivpage,ier)
        write(*,*)'-ESC- per uscire'

 1      continue
c       ----read condition----
        call output (adr3,r)
c       ----clear memory data buffer----
        do 90 i=1,10
        c(i)=0
 90     continue
c       ----sync con il PPS----
```

```
 100    call output(adr2,adclk)
        fs=input(adr1)
        pps=btest(fs,0)
        if(pps) go to 100
 200    call output(adr2,adclk)
        fs=input (adr1)
        pps=btest(fs,0)
        if (.not.pps) go to 200
c       ----ut data/clock information----
        call output (adr2,ltctime)
        call output (adr3,w)
        call output (adr3,r)
c       ----reading data/clock----
        call output (adr2,rcg)
        d(1)= input (adr1)
        call output (adr2,rgg)
        d(2)=input (adr1)
        call output (adr2,rhh)
        c(3)= input (adr1)
        call output (adr2,rmm)
        c(4)=input (adr1)
        call output (adr2,rss)
        c(5)= input (adr1)
        call output (adr2,rcs)
        c(6)= input (adr1)
c       ----data/clock nibble swooping----
        c(3)= ishc(c(3),4)
        c(4)= ishc(c(4),4)
        c(5)= ishc(c(5),4)
        c(6)= ishc(c(6),4)
c       ----clock's display----
c       d(2)=ibclr(d(2),15)
        cgu=d(1)/16
        ggu=d(2)/16
        dgu=d(2)-ggu*16
        day=(cgu*100)+(dgu*10)+ggu
        if (day.gt.366) write (*,*)'clock`s fatal error!!'
        dhu=c(3)/16
        hhu=c(3)-dhu*16
        dmu=c(4)/16
        mmu=c(4)-dmu*16
        dsu=c(5)/16
        ssu=c(5)-dsu*16
        sd=337+mo(2)
        do 290 im=12,1,-1
        sd=sd-mo(im)
        if (day.gt.sd) go to 295
 290    continue
 295    gm=day-sd
        irow=15
        icolm=5
        ivpage=0
        call putcur(irow,icolm,ivpage,ier)
        write (*,300)day,gm,im,an,dhu,hhu,dmu,mmu,dsu,ssu
```

```
300     format(25x,i3,5x,i2,'/',i2,'/',i2,'/',' ',2i1,':',2i1 ,':',2i1)
        call inkey(cr,key,ier)
        if(key.eq.1) goto 500
        go to 1
500     continue
        end
```

```
$DECLARE
c       *********CLOCK CONTROL*********
c                 Prog.name:LETSCLOC

        integer*2 adr1,adr2,adr3,w,r,i,input
        integer*2 adclk,d(2),wr,key,ier,irow,icolm,ivpage
        integer*1 c(10),fs,mo(12),t1,t2,t3,t4,t5,t6,t7,t8,t9
        integer*2 sd,day,hhs,mms,sss,css,cgu,dgu,ggu
        integer*1  gm,im,an,dhu,hhu,dmu,mmu,dsu,ssu,dcu,ucu
        integer*2 rcg,rgg,rhh,rmm,rss,rcs,ltctime,j,bis,biss
        logical pps,of
        character*8 string,cr

        data mo/31,28,31,30,31,30,31,31,30,31,30,31/
        call gdate(string)
        read(string,'(6x,i2)') an
        bis=int(an/4)
        biss=an-bis*4
        if(biss.eq.0) then
        data mo/31,29,31,30,31,30,31,31,30,31,30,31/
        endif


        adr1=768
        adr2=769
        adr3=770
        w=0
        r=1
        wr=0
        adclk=54
        rcg=48
        rgg=49
        rhh=50
        rmm=51
        rss=52
        rcs=53
        ltctime=48
        fs=0

        call cls
        irow=4
        icolm=5
        ivpage=0
        call putcur(irow,icolm,ivpage,ier)
        write(*,*)'-ESC- per uscire'
1       continue
c       ----read condition----
        call output (adr3,r)
c       ----clear memory data buffer----
        do 90 i=1,10
        c(i)=0
 90     continue
c       ----sync con il PPS----
```

```
 100    call output(adr2,adclk)
        fs=input(adr1)
        pps=btest(fs,0)
        if(pps) go to 100
 200    call output(adr2,adclk)
        fs=input (adr1)
        pps=btest(fs,0)
        if (.not.pps) go to 200
c       ----ut data/clock information----
        call output (adr2,ltctime)
        call output (adr3,w)
        call output (adr3,r)
c       ----reading data/clock----
        call output (adr2,rcg)
        d(1)= input (adr1)
        call output (adr2,rgg)
        d(2)=input (adr1)
        call output (adr2,rhh)
        c(3)= input (adr1)
        call output (adr2,rmm)
        c(4)=input (adr1)
        call output (adr2,rss)
        c(5)= input (adr1)
        call output (adr2,rcs)
        c(6)= input (adr1)
c       ----data/clock nibble swooping----
        c(3)= ishc(c(3),4)
        c(4)= ishc(c(4),4)
        c(5)= ishc(c(5),4)
        c(6)= ishc(c(6),4)
c       ----clock's display----
c       d(2)=ibclr(d(2),15)
        cgu=d(1)/16
        ggu=d(2)/16
        dgu=d(2)-ggu*16
        day=(cgu*100)+(dgu*10)+ggu
        if (day.gt.366) write (*,*)'clock`s fatal error!!'
        dhu=c(3)/16
        hhu=c(3)-dhu*16
        dmu=c(4)/16
        mmu=c(4)-dmu*16
        dsu=c(5)/16
        ssu=c(5)-dsu*16
        sd=337+mo(2)
        do 290 im=12,1,-1
        sd=sd-mo(im)
        if (day.gt.sd) go to 295
 290    continue
 295    gm=day-sd
        irow=15
        icolm=5
        ivpage=0
        call putcur(irow,icolm,ivpage,ier)
        write (*,300) day,gm,im,an,dhu,hhu,dmu,mmu,dsu,ssu
```

```
300     format(25x,i3,4x,i2,'/',i2,'/',i2,'/',' ',2i1,':',2i1 ,':',
     +2i1)
        call inkey(cr,key,ier)
        if(key.eq.1) goto 500
        go to 1
500     continue
        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
        write(dat,10) year,ut2,ut1
 10     format(i4,'.0',i1,i2)
        go to 18
       endif

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

       if(ut2.gt.9.and.ut1.le.9) then
        write(dat,12) year,ut2,ut1
 12     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 syncronised 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
      dmu=st4/16
      mmu=st4-dmu*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
```

-RELE-

Prova dei 4 blocchi di rele`.

Al calcolatore IBM XT di gestione della acquisizione dati, sono collegati 30 rele` per il controllo on/off di periferiche. Per potere testare i suddetti rele`, si fa uso del programma RELE, che chiede su quale blocco di 8 rele` si vuole agire, ed il numero decimale corrispondente alla parola binaria di configurazione dei rele` che si desidera avere, considerando che il contatto e` ON quando il rispettivo bit e` 1 e viceversa. Per uscire battere Ctrl-C.
Nella pagina seguente si riporta il relativo listato.

```
      integer*2adr1,adr2,adr3,rel,w,r,a1,nb,key,in
      character*5 ch
      adr1=768
      adr2=769
      adr3=770
      w=0
      r=1
c      rel=64

 1    call cls
      write(*,*)'Programma di prova rele`'
      write(*,*)'    '
      write(*,*)'Entrare il numero di blocco (1/4)'
      write(*,*)'blocco 1 = rele` 1-8'
      write(*,*)'blocco 2 = rele` 9-16'
      write(*,*)'blocco 3 = rele` 17-24'
      write(*,*)'blocco 4 = rele` 25-30'

      read(*,5)nb
 5    format(i2)
      if(nb.eq.1) rel=64
      if(nb.eq.2) rel=65
      if(nb.eq.3) rel=66
      if(nb.eq.4) rel=67


 10   call cls
      write (*,11)'Entrare il numero decimale relativo alla parola bin
     &aria corrisp. alla combinazione dei rele` voluta del blocco',nb
 11   format(a115,i2)
      write(*,*)' 1-->rele` ON   0-->rele` OFF'
      write(*,*)' CTRL-C per Uscire'

      read (*,*) a1


      call output (adr2,rel)
      call output(adr1,a1)
      call output (adr3,w)
      call output(adr3,r)
      go to 10
100   continue
      end
```

-RESET-

Resetta l'hardware dell'acquisizione dati

Il calcolatore di gestione del'acquisizione dati ha la possibilita` di accesso, tramite una interfaccia 'rele`', ad una certa quantita` di blocchi di controllo del telescopio, tipo generatori di marche interne, attenuatori da 1 dB in media frequenza, marche esterne ecc. E` molto importante, quindi, avere un programma che sia in grado di resettare tutto, sia in fase di test che prima dell'inizio di un qualsiasi programma osservativo; potrebbe infatti verificarsi il caso in cui, per una errata 'manovra' di uscita da un programma che ne prevede l'uso, un generatore resti attivo e danneggi cosi` l'osservazione.

```
$DECLARE

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

-ETOT-

Legge il livello del segnale in media frequenza di ogni singolo canale

Nel corso di programmi tipo sky survey, puo` essere utile leggere il valore del livello in media frequenza dei vari canali, per avere una indicazione circa il buon funzionamento del telescopio. E` stato scritto il programma ETOT che tramite l'acquisizione dati, fornisce la misura dei 14 livelli relativi ai canali singoli. Per fare questo e` stato approntato un sistema di 14 rivelatori, amplificatori ed integratori (verra` descritto in altra sede), collegato ad altrettanti canali dell'acquisizione dati. I valori forniti sono in unita` di 0.01 uW. Il listato del programma e` nella pagina seguente.

```
$DECLARE
c       -----PROGRAMMA DI LETTURA DELLA PROVA 'E'-----

c               Nome=Prove
c
        integer*2 t1,t2,t3,t4,t5,t6,i,ik,icgu,idgu,iggu,iday,r
        integer*2 ival(14),a1,a2,a3,a4,irow,icolm,ivpage,ierror,d
        integer*2 cr,key,in,ix1,iy1,ix2,iy2,icol,ibflg,ixor,nbyte
        real*4 iti,ite,rh,val(14),v(14)
        character*5 ind(14),ibyte
        character*45 titolo1
        character*55 temp
        character*50 intest,chdata
        character*80 titolo2,titolo3,titolo4
        call cls
        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'

        icolm=28
        ivpage=0
        irow=8

        write(*,*)'   '
        write(*,99)'Digit meno significativo = microwatt/100'
        write(*,*)'                                          '
        write(*,98)'premere -ESC- per terminare '
 99     format(11x,a42)
 98     format(17x,a33)

c       --- ciclo di lettura canali (224-237)
        call putcur(irow,icolm,ivpage,ierror)
        write(*,1234)
 1234   format('          ',$)
 33     continue
        irow=8


        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)*0.305176/2
        irow=irow+1
        call putcur(irow,icolm,ivpage,ierror)
        write(*,55)ind(ik),val(ik)
 55     format(2x,a3,'=',f4.0,$)
c       ---- controllo tasto di uscita -ESC- ----
        call inkey(cr,key,in)
        if(key.eq.1) goto 200
c       ----------------------------------------
 100    continue
        goto 33
 200    irow=23
        icolm=23
        call putcur(irow,icolm,ivpage,ierror)
        stop'fine programma'
        end
```

PROGRAMMI PER IL RAMO NS

Vengono riportati i programmi relativi ai fasci NS, gestiti dal personal di puntamento. I fasci NS vengono formati agendo sui ritardi variabili NS e sugli 8 sfasatori programmabili a 5 bit, inseriti a livello di media frequenza. I programmi scritti dei quali seguira` il listato sono:

-ROTANS-

Programma per la rotazione della fase complessiva del ramo NS, con velocita` di rotazione selezionabile.

Questo programma e` stato scritto per formare il fascio NS e ruotare in continuazione le fasi (step di 11.25 gradi) con velocita` selezionabile, dei canali NS, in fase sperimentale di determinazione del flusso equivalente di sistema del 'fascio totale' EW+NS, non conoscendo la fase relativa EW/NS. E` stato riportato comunque in questa sede perche` potenzialmente utile in situazioni analoghe che si potrebbero ripresentare in futuro.

-ROTMAN-

Programma per la rotazione 'manuale' della fase NS

Questo programma oltre che formare il fascio NS, offre la possibilita` di ruotare 'manualmente' la fase del ramo NS, a step di 11.25 gradi inseribili (+/-) agendo sui tasti up/down. E` anche possibile inserire un offset iniziale di fase, da cui partire con la rotazione manuale a step.

-ROTTEMP-

Programma per la rotazione della fase del ramo NS, con possibilita` di 'temporizzare' la rotazione

Questo programma e`, in realta`, una variante del programma ROTANS. Una volta formato il fascio NS e selezionata la velocita` di rotazione della fase, la rotazione si ferma ad ogni fine giro, per circa 1 secondo.

-ZENITH-

Programma che imposta le fasi I.F. ed i ritardi del ramo NS allo zenith

Questo programma serve per impostare allo zenith (cioe` a meno di offsets fissi, a zero) le fasi ed i ritardi del ramo NS.

```
$NOTRUNCATE
$DECLARE
c       --------------- PROGRAMMA ROTANS -----------------

c       PROGRAMMA PER LA ROTAZIONE DELLE FASI DEI CANALI NS
c       per il controllo deL FLUSSO EQ.di sys. della somma EW+NS
        NON conoscendo la fase EW/NS

c      --  outsfns
c      --  outport
c      --  fort488

        real dl(8),ze,dc,df(8),uno,ksign,iif(8)
        integer*2 val,i,f(8),outp(8),res,inum,ind,ciclo
        real dg,pr,fas(8),stelio,fnum,ff

        ze=44.57
        uno=1
        res=0
        ff=0
        ciclo=0

c       reset di tutti gli sfasatori

        do 99 i=1,8
        call outsfns(i-1,res)
 99     continue

        call cls
        write(*,*)'        '
        write(*,*)' FORMAZIONE FASCI NORD-SUD '
 1      write (*,*) 'ENTER DECLINATION AND TIME(deg.,min,1-5000) inum'
        read (*,*) dg,pr,inum
        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))
        df(i)=dl(i)-val
        df(i)=df(i)*360.
        write(*,*)'fasi',df(i)
 30     continue
c51     format(7x,'    4N    3N    2N    1N    1S    2S    3S   4S')
        do 61 i=1,8
        if(df(i).lt.0) df(i)=df(i)+360.
        outp(i)=nint(df(i)/(360./32.))
        if(outp(i).eq.32) outp(i)=outp(i)-32
        call outsfns(i-1,outp(i))
 61     continue
        write (*,111) outp
        pause
 111    format(8i3)
        write (*,50) dl
```

```
      write (*,50) df
 50   format(8f9.2)
c     ---------------------------
c     ------ rotazione fase ------

c     rotazione fase a step di 11.25 gradi

 200  continue
      do 500 i=1,8
      outp(i)=outp(i)+1
      if(outp(i).gt.31) outp(i)=outp(i)-32
      call outsfns(i-1,outp(i))
 500  continue
c     write(*,510)outp
c     --- delay introdotto da inum ---
      do 505 i=1,inum
      fnum=i
 505  stelio=cos(fnum)
 510  format(8i3)
      go to 200
      end
```

```
$DECLARE
c       PROGRAMMA PER LA ROTAZIONE MANUALE DELLA FASE  NS
c       PER LA DETERMINAZIONE DELLA FASE EW/NS
c       il programma ROTMAN e` linkato con:

c      --  outsfns
c      --  outport
c      --  fort488
c      --  nolimit.lib

        real dl(8),ze,dc,df(8),uno,ksign,iif(8)
        integer*2 val,i,f(8),outp(8),res,inum,ind,outp1(8),key,in,of
        real dg,pr,fas(8),stelio,fnum,ff,fase,off
        character*5 rinp
        ze=44.57
        uno=1
        res=0
        ff=0
        fase=0
c       reset di tutti gli sfasatori

c       do 99 i=1,8
c       call outsfns(i-1,res)
c99     continue

        call cls
        write(*,*)'        '
        write(*,*)' FORMAZIONE FASCI NORD-SUD con poss. di offsettare'
 1      write (*,*) 'ENTRA DECLINATIONE e OFFSET (deg.,min   deg.)'
        write (*,*)'premere le freccie up/dwn per mod. la fase'
        write (*,*)'a step di +/- 11 gradi'
        read (*,*) dg,pr,off
        write(*,*)' <ESC> to Exit'
        write(*,*)
        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))
        df(i)=dl(i)-val
        df(i)=df(i)*360.
c       write(*,*)'fasi',df(i)
 30     continue
c51     format(7x,'    4N    3N    2N    1N    1S    2S    3S   4S')
        do 61 i=1,8
        if(df(i).lt.0) df(i)=df(i)+360.
        outp(i)=nint(df(i)/(360./32.))
        if(outp(i).ge.32) outp(i)=outp(i)-32
 61     continue
        write(*,112) outp
 112    format(' senza offset',8i3)
        do 1000 i=1,8

c       ---- introduzione offset  e gestione sfasatori----
```

```
        of=nint(off/(360./32.))
        outp(i)=outp(i)+of
        if(outp(i).ge.32) outp(i)=outp(i)-32
        if(outp(i).lt.0) outp(i)=outp(i)+32
        call outsfns(i-1,outp(i))
 1000   continue

        write(*,59)  outp
 59     format(' con offset  ',8i3)
        write(*,*)
 111    format(1h+, 'fase ',f8.0)

c       --- controllo della fase tramite freccie up-dw ---
c       ---ogni premuta di tasto up-dw vale +11/-11 gradi rispett. ---

 45     call inkey(rinp,key,in)
        if(in.eq.0) go to 45
c       write(*,*) key,in
        if(key.eq.1) stop
        do 100 i=1,8
        if(key.eq.72)  outp(i)=outp(i)+1
        if(key.eq.80)  outp(i)=outp(i)-1
        if(outp(i).ge.32) outp(i)=outp(i)-32
        if(outp(i).lt.0) outp(i)=outp(i)+32


        call outsfns(i-1,outp(i))
 100    continue

        if(key.eq.72)  fase=fase+11.25
        if(key.eq.80)  fase=fase-11.25

        write (*,111) fase
        go to 45
        end
```

```
$NOTRUNCATE
$DECLARE
c       PROGRAMMA PER IL CALCOLO DELLE FASI DEI CANALI NS
c       PER LA FORMAZIONE DEL FASCIO NS
c       il programma ROTTEMP e` linkato con:

c      --  outsfns
c      --  outport
c      --  fort488

        real dl(8),ze,dc,df(8),uno,ksign,iif(8)
        integer*2 val,i,f(8),outp(8),res,inum,ind,ciclo
        real dg,pr,fas(8),stelio,fnum,ff

        ze=44.57
        uno=1
        res=0
        ff=0
        ciclo=0

c       reset di tutti gli sfasatori

        do 99 i=1,8
        call outsfns(i-1,res)
 99     continue

        call cls
        write(*,*)'        '
        write(*,*)' FORMAZIONE FASCI NORD-SUD '
 1      write (*,*) 'ENTER DECLINATION (deg.,min) inum'
        read (*,*) dg,pr,inum
        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))
        df(i)=dl(i)-val
        df(i)=df(i)*360.
        write(*,*)'fasi',df(i)
 30     continue
c51     format(7x,'    4N    3N    2N    1N    1S    2S    3S   4S')
        do 61 i=1,8
        if(df(i).lt.0) df(i)=df(i)+360.
        outp(i)=nint(df(i)/(360./32.))
        if(outp(i).eq.32) outp(i)=outp(i)-32
        call outsfns(i-1,outp(i))
 61     continue
        write (*,111) outp
        pause
 111    format(8i3)
        write (*,50) dl
        write (*,50) df
 50     format(8f9.2)
c       ---------------------------
```

```
c       ------ rotazione fase ------

c       rotazione fase a step di 11.25 gradi

 200    ciclo=ciclo+1
        do 500 i=1,8
        outp(i)=outp(i)+1
        if(outp(i).gt.31) outp(i)=outp(i)-32
        call outsfns(i-1,outp(i))
 500    continue
c       write(*,510)outp
c       --- controllo del giro di 360 gradi ---
        ff=ff+1
        if(ff.gt.31) go to 600
        do 505 i=1,inum
        fnum=i
 505    stelio=cos(fnum)
 510    format(8i3)
        write(*,*)ciclo
        go to 200
c       --- controllo pausa rotazione fase ----
 600    ind=inum*15
        write(*,608)outp
        do 605 i=1,ind
        fnum=i
 605    stelio=cos(fnum)
        ff=0
        ciclo=0
 608    format(' fine giro: fase attuale ',8i3)
        go to 200
        end
```

```
$DECLARE

c       ========================
c       PROGRAMMA AZZERAMENTO
c       DEL RAMO N/S ALLO ZENITH
c       (RITADI)
c       ========================
c       Nome:ZENITH

c       S. Montebugnoli 12/11/1990

c       ---Questo programma e` linkato con le routine
c       ---- FORT488  --> routine (scheda IEEE488) IN/OUT

        integer*2 di(8),pi(8),zi(8),ab,ad,ver,abl,jdg,jpr,i
        integer*2 a,b,c,d,e,f,g,h,smot,amot,mux,sconv,flag,val
        integer*2 a1,a2,a3,a4,a5,a6,a7,a8,w,r,v,s,ind,let1,let2
        integer*2 al1,al2,al3,al4,al5,al6,al7,al8,z1,z2
        integer*2 in,ng,nr,fre,dur,input,j,inx,jp
        integer*2 adr1,adr2,adr3,adr4,s1,s2,s3,s4,n1,n2,n3,n4
        integer*2 irow,icolm,ivpage,ierror,key
        integer*1 zs(8)
        integer*4 iseg,rec
        character*80 out
        character*30 pun$
        character*1 chr(80),as$,cr
        character*11 punt$
        equivalence(out,chr(1))
        real ze,dc,dg,pr,k,ksign,b1,b2,b3,b4,b5,b6,b7,b8,ai,bi,ri
        real dl(8),df(8),p(8),uno,l(8),z(8),m(8),liv,del
        character*10 a$
        data adr1/768/,adr2/769/,adr3/770/,adr4/771/
        data smot/3/,amot/4/,mux/5/,sconv/6/,flag/13/,let1/1/,let2/0/
        data s4/8/,s3/2/,s2/1/,s1/0/
        data n1/9/,n2/10/,n3/11/,n4/12/
        data w/0/,r/1/
        data uno/1./
c       ========================
c       Reset di tutto il Sistema
c       ========================
        val=0
        call output (adr4,val)
c       ************************
c       ze=zenith in gradi,decimi
c       ************************
        ze=44.57
c       ------------------------
c       smot=senso motori
c       amot=abilitazione motori
c       mux,sconv,flag,let=gestione mux/convertitore A/D
c       ----blocco del sistema----
        call output (adr2,amot)
        abl=0
        call output (adr1,abl)
```

```
        call output (adr3,w)
        call output (adr3,r)
c       ================================
c         GESTIONE RITARDI VARIABILI
c       ================================
c       ---- scrittura su disco della delta 'puntata' -----
        open(unit=10,file='\croce\puntns.dat',access='direct',recl=50,
     &form='formatted')
        jdg=99
        jpr=99
        write(10,19,rec=1) jdg
        write(10,19,rec=2) jpr
 19     format(2x,i2)
c       ------------------------------------------------------------
        bi=ze/57.29578
        ri=ze/57.29578
        ai=(bi-ri)
c       ---- inizio procedura "ritardi" ----
c       **** 4 SUD ***
        k=1
        b1=8*(k-4.5)*sin(ai)
        a1=nint(b1)
        a1=a1+23
        if(a1.ge.32) go to 20
        fre=800
        dur=4
        call beep2(fre,dur,in)
        call output (adr2,s4)
        call output (adr1,a1)
        call output (adr3,w)
        call output (adr3,r)
        go to 30
 20     val=a1+16
        call output (adr2,s4)
        call output (adr1,val)
        call output (adr3,w)
        call output (adr3,r)
c       --------------------
c       **** 3 SUD ****
 30     k=2
        b2=8*(k-4.5)*sin(ai)
        a2=nint(b2)
        a2=a2+23
        fre=1000
        dur=4
        call beep2(fre,dur,in)
        if (a2.ge.32) go to 40
        call output (adr2,s3)
        call output (adr1,a2)
        call output (adr3,w)
        call output (adr3,r)
        go to 45
 40     val=a2+16
        call output(adr2,s3)
```

```
        call output(adr1,val)
        call output(adr3,w)
        call output(adr3,r)
c       ---------------------
c       **** 2 SUD ****
 45     k=3
        b3=8*(k-4.5)*sin(ai)
        a3=nint(b3)
        a3=a3+15
        fre=1200
        dur=4
        call beep2(fre,dur,in)
        call output (adr2,s2)
        call output (adr1,a3)
        call output (adr3,w)
        call output (adr3,r)
c       ---------------------
c       **** 1 SUD ****
        k=4
        b4=8*(k-4.5)*sin(ai)
        a4=nint(b4)
        a4=a4+7
        fre=1400
        dur=4
        call beep2(fre,dur,in)
        call output (adr2,s1)
        call output (adr1,a4)
        call output (adr3,w)
        call output (adr3,r)
c       ---------------------
c       **** 1 NORD ****
        k=5
        b5=8*(k-4.5)*sin(ai)
        a5=nint(b5)
        a5=a5+7
        fre=1600
        dur=4
        call beep2(fre,dur,in)
        call output (adr2,n1)
        call output (adr1,a5)
        call output (adr3,w)
        call output (adr3,r)
c       ---------------------
c       **** 2 NORD ****
        k=6
        b6=8*(k-4.5)*sin(ai)
        a6=nint(b6)
        a6=a6+15
        fre=1800
        dur=4
        call beep2(fre,dur,in)
        call output (adr2,n2)
        call output (adr1,a6)
        call output (adr3,w)
```

```
        call output (adr3,r)
c       --------------------
c       **** 3 NORD ****
        k=7
        b7=8*(k-4.5)*sin(ai)
        a7=nint(b7)
        a7=a7+23
        fre=2000
        dur=4
        call beep2(fre,dur,in)
        if (a7.ge.32) go to 60
        call output (adr2,n3)
        call output (adr1,a7)
        call output (adr3,w)
        call output (adr3,r)
        go to 70
 60     val=a7+16
        call output(adr2,n3)
        call output(adr1,val)
        call output(adr3,w)
        call output(adr3,r)
c       --------------------
c       **** 4 NORD ****
 70     k=8
        b8=8*(k-4.5)*sin(ai)
        a8=nint(b8)
        a8=a8+23
        fre=2200
        dur=4
        call beep2(fre,dur,in)
        if (a8.ge.32) go to 80
        call output (adr2,n4)
        call output (adr1,a8)
        call output (adr3,w)
        call output (adr3,r)
        go to 90
 80     val=a8+16
        call output (adr2,n4)
        call output (adr1,val)
        call output (adr3,w)
        call output (adr3,r)
c       --------------------
 90     call beep2 (fre,dur,in)
        call  cls
        stop ' **** Ritardi puntati allo Zenith ****'
        end
```

APPENDICE 1

Viene riportata la lista dei canali d'ingresso all'acquisizione dati; i canali dall'1 al 96 sono gli "interferometri sciolti"; i canali dal 97 al 110 sono interferometri NS-NS (non ancora calibrabili); i canali dal 111 al 120 ed il canale 126 sono alcuni fasci correlati (non attualmente utilizzati; i canali dal 121 al 125 sono 5 fasci B a banda stretta (circa 500 KHz).

Seguono poi i canali singoli "total power"; il canale 134 ed i canali dal 146 al 150, sono interferometri a somma EW-EW a banda larga, usati per calibrare (tramite il programma WIDECAL) il fascio B a larga banda (circa 6 MHz), usato per la survey di pulsar.

CANALE 1 ..... 1EST * 1NORD (cos)
CANALE 2 ..... 1EST * 1NORD (sen)
CANALE 3 ..... 2EST * 1NORD (cos)
CANALE 4 ..... 2EST * 1NORD (sen)
CANALE 5 ..... 3EST * 1NORD (cos)
CANALE 6 ..... 3EST * 1NORD (sen)
CANALE 7 ..... 4EST * 1NORD (cos)
CANALE 8 ..... 4EST * 1NORD (sen)
CANALE 9 ..... 5EST * 1NORD (cos)
CANALE 10 ..... 5EST * 1NORD (sen)
CANALE 11 ..... 6EST * 1NORD (cos)
CANALE 12 ..... 6EST * 1NORD (sen)

CANALE 13 ..... 1EST * 2NORD (cos)
CANALE 14 ..... 1EST * 2NORD (sen)
CANALE 15 ..... 2EST * 2NORD (cos)
CANALE 16 ..... 2EST * 2NORD (sen)
CANALE 17 ..... 3EST * 2NORD (cos)
CANALE 18 ..... 3EST * 2NORD (sen)
CANALE 19 ..... 4EST * 2NORD (cos)
CANALE 20 ..... 4EST * 2NORD (sen)
CANALE 21 ..... 5EST * 2NORD (cos)
CANALE 22 ..... 5EST * 2NORD (sen)
CANALE 23 ..... 6EST * 2NORD (cos)
CANALE 24 ..... 6EST * 2NORD (sen)

CANALE 25 ..... 1EST * 3NORD (cos)
CANALE 26 ..... 1EST * 3NORD (sen)
CANALE 27 ..... 2EST * 3NORD (cos)
CANALE 28 ..... 2EST * 3NORD (sen)
CANALE 29 ..... 3EST * 3NORD (cos)
CANALE 30 ..... 3EST * 3NORD (sen)
CANALE 31 ..... 4EST * 3NORD (cos)
CANALE 32 ..... 4EST * 3NORD (sen)
CANALE 33 ..... 5EST * 3NORD (cos)
CANALE 34 ..... 5EST * 3NORD (sen)
CANALE 35 ..... 6EST * 3NORD (cos)
CANALE 36 ..... 6EST * 3NORD (sen)

CANALE 37 ..... 1EST * 4NORD (cos)
CANALE 38 ..... 1EST * 4NORD (sen)
CANALE 39 ..... 2EST * 4NORD (cos)
CANALE 40 ..... 2EST * 4NORD (sen)

CANALE 41 . . . . . 3EST * 4NORD (cos)
CANALE 42 . . . . . 3EST * 4NORD (sen)
CANALE 43 . . . . . 4EST * 4NORD (cos)
CANALE 44 . . . . . 4EST * 4NORD (sen)
CANALE 45 . . . . . 5EST * 4NORD (cos)
CANALE 46 . . . . . 5EST * 4NORD (sen)
CANALE 47 . . . . . 6EST * 4NORD (cos)
CANALE 48 . . . . . 6EST * 4NORD (sen)

CANALE 49 . . . . . 1EST * 1SUD (cos)
CANALE 50 . . . . . 1EST * 1SUD (sen)
CANALE 51 . . . . . 2EST * 1SUD (cos)
CANALE 52 . . . . . 2EST * 1SUD (sen)
CANALE 53 . . . . . 3EST * 1SUD (cos)
CANALE 54 . . . . . 3EST * 1SUD (sen)
CANALE 55 . . . . . 4EST * 1SUD (cos)
CANALE 56 . . . . . 4EST * 1SUD (sen)
CANALE 57 . . . . . 5EST * 1SUD (cos)
CANALE 58 . . . . . 5EST * 1SUD (sen)
CANALE 59 . . . . . 6EST * 1SUD (cos)
CANALE 60 . . . . . 6EST * 1SUD (sen)

CANALE 61 . . . . . 1EST * 2SUD (cos)
CANALE 62 . . . . . 1EST * 2SUD (sen)
CANALE 63 . . . . . 2EST * 2SUD (cos)
CANALE 64 . . . . . 2EST * 2SUD (sen)
CANALE 65 . . . . . 3EST * 2SUD (cos)
CANALE 66 . . . . . 3EST * 2SUD (sen)
CANALE 67 . . . . . 4EST * 2SUD (cos)
CANALE 68 . . . . . 4EST * 2SUD (sen)
CANALE 69 . . . . . 5EST * 2SUD (cos)
CANALE 70 . . . . . 5EST * 2SUD (sen)
CANALE 71 . . . . . 6EST * 2SUD (cos)
CANALE 72 . . . . . 6EST * 2SUD (sen)

CANALE 73 . . . . . 1EST * 3SUD (cos)
CANALE 74 . . . . . 1EST * 3SUD (sen)
CANALE 75 . . . . . 2EST * 3SUD (cos)
CANALE 76 . . . . . 2EST * 3SUD (sen)
CANALE 77 . . . . . 3EST * 3SUD (cos)
CANALE 78 . . . . . 3EST * 3SUD (sen)
CANALE 79 . . . . . 4EST * 3SUD (cos)
CANALE 80 . . . . . 4EST * 3SUD (sen)
CANALE 81 . . . . . 5EST * 3SUD (cos)
CANALE 82 . . . . . 5EST * 3SUD (sen)
CANALE 83 . . . . . 6EST * 3SUD (cos)
CANALE 84 . . . . . 6EST * 3SUD (sen)

CANALE 85 . . . . . 1EST * 4SUD (cos)
CANALE 86 . . . . . 1EST * 4SUD (sen)
CANALE 87 . . . . . 2EST * 4SUD (cos)
CANALE 88 . . . . . 2EST * 4SUD (sen)
CANALE 89 . . . . . 3EST * 4SUD (cos)
CANALE 90 . . . . . 3EST * 4SUD (sen)
CANALE 91 . . . . . 4EST * 4SUD (cos)
CANALE 92 . . . . . 4EST * 4SUD (sen)
CANALE 93 . . . . . 5EST * 4SUD (cos)
CANALE 94 . . . . . 5EST * 4SUD (sen)

CANALE 95 . . . . . 6EST * 4SUD (cos)
CANALE 96 . . . . . 6EST * 4SUD (sen)

CANALE 97 . . . . . 2SUD * 3NORD (cos)
CANALE 98 . . . . . 2SUD * 3NORD (sen)
CANALE 99 . . . . . 1SUD * 2NORD (cos)
CANALE 100 . . . . . 1SUD * 2NORD (sen)
CANALE 101 . . . . . 1NORD * 2NORD (cos)
CANALE 102 . . . . . 1NORD * 2NORD (sen)
CANALE 103 . . . . . 4SUD * 4NORD (cos)
CANALE 104 . . . . . 4SUD * 4NORD (sen)
CANALE 105 . . . . . 3SUD * 1NORD (cos)
CANALE 106 . . . . . 3SUD * 1NORD (sen)
CANALE 107 . . . . . 3SUD * 1SUD (cos)
CANALE 108 . . . . . 3SUD * 1SUD (sen)
CANALE 109 . . . . . 2SUD * 4NORD (cos)
CANALE 110 . . . . . 2SUD * 4NORD (sen)

CANALE 111 . . . . . H * B (cos)
CANALE 112 . . . . . H * B (sen)
CANALE 113 . . . . . H * C (cos)
CANALE 114 . . . . . H * C (sen)
CANALE 115 . . . . . I * A (cos)
CANALE 116 . . . . . I * A (sen)
CANALE 117 . . . . . I * B (cos)
CANALE 118 . . . . . I * B (sen)
CANALE 119 . . . . . I * C (cos)
CANALE 120 . . . . . I * C (sen)
CANALE 121 . . . . . B1 (500 KHz)
CANALE 122 . . . . . B2 (500 KHz)
CANALE 123 . . . . . B3 (500 KHz)
CANALE 124 . . . . . B4 (500 KHz)
CANALE 125 . . . . . B5 (500 KHz)
CANALE 126 . . . . . Y * C (sen)

CANALE 127 . . . . . 1EST
CANALE 128 . . . . . 2EST
CANALE 129 . . . . . 3EST
CANALE 130 . . . . . 4EST
CANALE 131 . . . . . 5EST
CANALE 132 . . . . . 6EST

CANALE 133 . . . . . A
CANALE 134 . . . . . 1E+4E (wide)
CANALE 135 . . . . . B
CANALE 136 . . . . . B/1
CANALE 137 C
CANALE 138 . . . . . 1NORD
CANALE 139 . . . . . 2NORD
CANALE 140 . . . . . 3NORD
CANALE 141 . . . . . 4NORD
CANALE 142 . . . . . 1SUD
CANALE 143 . . . . . 2SUD
CANALE 144 . . . . . 3SUD
CANALE 145 . . . . . 4SUD

CANALE 146 . . . . . 2E+5E (wide)
CANALE 147 . . . . . 3E+6E (wide)

CANALE 148 . . . . . 2E+6E (wide)
CANALE 149 . . . . . 1E+4E (wide)
CANALE 150 . . . . . 1E+5E (wide)

CANALE 151 . . . . . Temperatura Int. (100mV/°C)
CANALE 152 . . . . . Temperatura Ext. (100mV/°C)
CANALE 153 . . . . . Umidita` Relativa Ext. (10mV/%)
CANALE 154 . . . . . .

. . . . . .

CANALE 224 . . . . . .
CANALE 225 . . . . . 1E (Livello di I.F. 200mV/µW)
CANALE 226 . . . . . 2E (Livello di I.F. 200mV/µW)
CANALE 227 . . . . . 3E (Livello di I.F. 200mV/µW)
CANALE 228 . . . . . 4E (Livello di I.F. 200mV/µW)
CANALE 229 . . . . . 5E (Livello di I.F. 200mV/µW)
CANALE 230 . . . . . 6E (Livello di I.F. 200mV/µW)
CANALE 231 . . . . . 1S (Livello di I.F. 200mV/µW)
CANALE 232 . . . . . 2S (Livello di I.F. 200mV/µW)
CANALE 233 . . . . . 3S (Livello di I.F. 200mV/µW)
CANALE 234 . . . . . 4S (Livello di I.F. 200mV/µW)
CANALE 235 . . . . . 1N (Livello di I.F. 200mV/µW)
CANALE 236 . . . . . 2N (Livello di I.F. 200mV/µW)
CANALE 237 . . . . . 3N (Livello di I.F. 200mV/µW)
CANALE 238 . . . . . 4N (Livello di I.F. 200mV/µW)

APPENDICE 2

Viene riportato l'elenco dei moduli software contenuti nella libreria CROCE.

```
  ABERR.............astro
* BEEP2.............beep2
* CBKOFF............icomsub3
* CHDIR.............CHDIR
  CHKHRC............CHKHRC
* CIRCA.............CIRCA2
  CLOCKSID..........clocksid
* CLOSIT............CLOSIT
  CLROUT............icomsub3
* CLSATT............CLSATT
* CMPSTG............CMPSTG2
* COMLEN............icomsub3
  CONV..............conv
  CORRE.............correct
* CURDIR............CURDIR
* CURTYP............CURTYP
  DECODE............tapec
* DIRINF............dirinf
  DMA2..............fort488
* DSKINF............DSKINF
  ENTER.............fort488
  EXTRD.............tapec
  FASENS............fasens
  FCBTMPSEG.........OPENUP
* FILDIR............FILDIR
* FILSIZ............FILSIZ
* FLEDEL............FLEDEL3
* FRANDB............FRANDB
* FRNEGB............FRNEGB
* FRNOTB............FRNOTB
* FRORB.............FRORB
* FRROLB............frrolb
* FRRORB............frrorb
* FRSHLB............frshlb
* FRSHRB............frshrb
* FRXORB............FRXORB
  FSFWHIGHSP........tapec
  FSRHIGHSP.........tapec
  FW1BLK............tapec
* GETBUF............GETBUF2
* GETDOS............GETDOS
* GETVID............GETVID
* GSETCL............GSETCL
* GTVRFY............GTVRFY
* HRCCLS............herc10
* INCOM.............icomsub3
* INKEY.............INKEY
  INPUT.............fort488
* INSTR.............INSTR2
* IRANDM............IRANDM
  IUNITOFF..........OPENUP
  JULIAN............astro
* JUSTR.............JUSTR2
* LINEA.............LINEA3
* BCARAY............bcaray
* BEEPIT............BEEPIT
* CBRKON............icomsub3
* CHKDRV............CHKDRV
  CHK_EGA...........CHK_EGA
  CLOCK.............clock
  CLOSFZ............CLOSFZ
  CLRIN.............icomsub3
* CLS...............CLS
* CLSCOM............icomsub3
* CMSTAT............CMSTAT
* CONCAT............CONCAT2
  CONVERSION........suconv
  CORRECT...........correct
* CURDSK............CURDSK
  DATE..............astro
* DIRDIR............DIRDIR
  DMA...............fort488
* DPAGE.............DPAGE
* EGAPRS............EGAPRS
* EQPMNT............EQPMNT
  EXTWR.............tapec
  FCBTMPOFF.........OPENUP
* FEXIST............FEXIST3
* FILINF............FILINF
* FILSTG............FILSTG
* FLEDLZ............FLEDLZ
* FRANDW............FRANDW
* FRNEGW............FRNEGW
  FRNOTW............FRNOTW
* FRORW.............FRORW
* FRROLW............frrolw
* FRRORW............frrorw
* FRSHLW............frshlw
* FRSHRW............frshrw
* FRXORW............FRXORW
  FSLOWSP...........tapec
  FSRLOWSP..........tapec
* GDATE.............GDATE
* GETCUR............GETCUR
* GETSCR............getscr
  GPIBSE............fort488
* GTPATH............gtpath
  HMS...............astro
* HRCOPN............herc10
  INITIA............fort488
* INOCOM............INOCOM
* INSERT............insert
* IOPCOM............icomsub3
* IRNDM2............IRNDM2
  IUNITSEG..........OPENUP
* JUSTL.............JUSTL2
  LINE..............line
* LNEBUF............LNEBUF
```

```
* MEFARC.....MEFARC
* MKDIR.....MKDIR
  MOON.....astro
  MXRECSEG.....OPENUP
  NC_TEL_DEALLOCATE..nctel1
  NFILMX.....OPENUP
  NUTATION.....astro
  OPEN_COM1.....nctel1
* OPNNEW.....OPNNEW
* OUTCOM.....icomsub3
  OUTPORT.....outport
  OUTSFNS.....outsfns
* PACKR.....PACKR2
  PAUSA.....pausa
  PC488S.....fort488
  PCPOINT.....nctel1
* PEEK.....fort488
* PORTIN.....PORTIN2
* PPOLL.....fort488
* PRTSCR.....PRTSCR
* PUTSCR.....putscr
* RCHAR.....rchar
* RDLINE.....RDLINE
* RDRND.....RDRND
  READ.....tapec
  RECVA.....fort488
  REW.....tapec
* RNFLEZ.....RNFLEZ
  RV1BLK.....tapec
  SCLOCK.....sclock
* SCROLL.....SCROLL
* SEGAO1.....segao1
* SELDSK.....SELDSK
* SETBRD.....setbrd
  SPOLL.....fort488
  SRCFCBSEG.....OPENUP
  SRQ3.....fort488
  SUNPOS.....astro
* TEGAIB.....TEGAIB
* TIMEOD.....TIMEOD2
  TMPDTASEG.....OPENUP
  UCLOCK.....uclock
* UPCASE.....UPCASE2
* WCHAR.....wchar2
* WDOT.....wdot
  WRITE.....tapec
* WRTRND.....WRTRND
  XQTDOS.....xqtdos
* XQTPGM.....xqtpgm
* ZFBLNK.....ZFBLNK2
* MIDCHR.....MIDCHR2
* MKPATH.....MKPATH
  MXRECOFF.....OPENUP
  NC_TEL_ALLOCATE...nctel1
  NFIL.....nfil
  NOPEN.....OPENUP
* OPENUP.....OPENUP
* OPNCOM.....OPNCOM
* OPNOLD.....OPNOLD
* OUTLEN.....icomsub3
  OUTPUT.....fort488
* PACKL.....PACKL2
* PAINTA.....PAINTA2
  PAUSE.....nctel1
  PCEND.....nctel1
  PCSTAT.....pcstat
* POKE.....fort488
* PORTOT.....PORTOT2
* PRNASM.....PRNASM
* PUTCUR.....PUTCUR
  RARRAY.....fort488
  RDHEAD.....rdhead
* RDOT.....RDOT
* RDRND2.....rdrnd2
  RECEIV.....fort488
* RENFLE.....RENFLE3
* RMDIR.....RMDIR
* ROTATE.....rotate
  SBYTES.....tapec
* SCREEN.....screen
* SEGAAL.....SEGAAL
* SEGAP1.....SEGAP1
  SEND.....fort488
  SIDG.....astro
  SRCFCBOFF.....OPENUP
  SRQ2.....fort488
* STVRFY.....STVRFY
  TARRAY.....fort488
* TIMASM.....TIMASM
  TMPDTAOFF.....OPENUP
  TRANSM.....fort488
* UCMSTG.....UCMSTG
  WAIT.....wait
* WCHARS.....wchars
  WRFMK.....tapec
* WRTRN2.....wrtrn2
  XMITA.....fort488
  XQTDS2.....XQTDS2
  XQTPM2.....XQTPM2

fort488          Offset: 00000010H  Code and data size: 61cH
  DMA              DMA2               ENTER            GPIBSE
  INITIA           INPUT              OUTPUT           PC488S
```

PEEK	POKE	PPOLL	RARRAY
RECEIV	RECVA	SEND	SPOLL
SRQ2	SRQ3	TARRAY	TRANSM
XMITA			

wait Offset: 000008c0H Code and data size: 149H
  WAIT

nfil Offset: 00000bc0H Code and data size: b8fH
  NFIL

BEEPIT Offset: 00000d80H Code and data size: 43H
  BEEPIT

CHDIR Offset: 00000e30H Code and data size: 2eH
  CHDIR

CHKDRV Offset: 00000ec0H Code and data size: 2c4H
  CHKDRV

CHKHRC Offset: 00001030H Code and data size: 2aH
  CHKHRC

CHK_EGA Offset: 000010c0H Code and data size: 29H
  CHK_EGA

CIRCA2 Offset: 00001150H Code and data size: 3d9H
  CIRCA

CLOSFZ Offset: 000019c0H Code and data size: 20H
  CLOSFZ

CLOSIT Offset: 00001a40H Code and data size: 157H
  CLOSIT

CLS Offset: 00001d30H Code and data size: 2fH
  CLS

CLSATT Offset: 00001dc0H Code and data size: 30H
  CLSATT

CMPSTG2 Offset: 00001e50H Code and data size: 74H
  CMPSTG

CMSTAT Offset: 00001f80H Code and data size: 5dH
  CMSTAT

CONCAT2 Offset: 00002040H Code and data size: 71H
  CONCAT

CURDIR Offset: 00002170H Code and data size: 39H
  CURDIR

CURDSK Offset: 00002210H Code and data size: 1aH

CURDSK

CURTYP Offset: 00002290H Code and data size: 31H
CURTYP

DIRDIR Offset: 00002320H Code and data size: 18eH
DIRDIR

DPAGE Offset: 000025d0H Code and data size: 87H
DPAGE

DSKINF Offset: 00002720H Code and data size: 40H
DSKINF

EGAPRS Offset: 000027c0H Code and data size: 8bH
EGAPRS

EQPMNT Offset: 00002910H Code and data size: a5H
EQPMNT

FEXIST3 Offset: 00002a10H Code and data size: 109H
FEXIST

FILDIR Offset: 00002bc0H Code and data size: 190H
FILDIR

FILINF Offset: 00002e30H Code and data size: f1H
FILINF

FILSIZ Offset: 00002fb0H Code and data size: 12fH
FILSIZ

FILSTG Offset: 00003180H Code and data size: 84H
FILSTG

FLEDEL3 Offset: 000032c0H Code and data size: 109H
FLEDEL

FLEDLZ Offset: 00003470H Code and data size: 1dH
FLEDLZ

FRANDB Offset: 000034f0H Code and data size: 2cH
FRANDB

FRANDW Offset: 00003580H Code and data size: 2cH
FRANDW

FRNEGB Offset: 00003610H Code and data size: 21H
FRNEGB

FRNEGW Offset: 00003690H Code and data size: 21H
FRNEGW

FRNOTB Offset: 00003710H Code and data size: 21H

FRNOTB

FRNOTW Offset: 00003790H Code and data size: 21H
FRNOTW

FRORB Offset: 00003810H Code and data size: 2cH
FRORB

FRORW Offset: 000038a0H Code and data size: 2cH
FRORW

FRXORB Offset: 00003930H Code and data size: 2cH
FRXORB

FRXORW Offset: 000039c0H Code and data size: 2cH
FRXORW

GDATE Offset: 00003a50H Code and data size: 58H
GDATE

GETBUF2 Offset: 00003b20H Code and data size: 1dcH
GETBUF

GETCUR Offset: 00003e50H Code and data size: 9bH
GETCUR

GETDOS Offset: 00003fc0H Code and data size: 2fH
GETDOS

GETVID Offset: 00004050H Code and data size: b2H
GETVID

GSETCL Offset: 00004200H Code and data size: 7eH
GSETCL

GTVRFY Offset: 00004300H Code and data size: 1aH
GTVRFY

INKEY Offset: 00004380H Code and data size: 58H
INKEY

INOCOM Offset: 00004450H Code and data size: 21bH
INOCOM

INSTR2 Offset: 00004800H Code and data size: bdH
INSTR

IRANDM Offset: 00004990H Code and data size: 4eH
IRANDM

IRNDM2 Offset: 00004a50H Code and data size: 93H
IRNDM2

JUSTL2 Offset: 00004b70H Code and data size: 99H

JUSTL

JUSTR2 Offset: 00004cb0H Code and data size: 9fH
JUSTR

LINEA3 Offset: 00004e00H Code and data size: 4b9H
LINEA

LNEBUF Offset: 00005840H Code and data size: 231H
LNEBUF

MEFARC Offset: 00005c00H Code and data size: 70dH
MEFARC

MIDCHR2 Offset: 00006a40H Code and data size: 71H
MIDCHR

MKDIR Offset: 00006b70H Code and data size: 2eH
MKDIR

MKPATH Offset: 00006c00H Code and data size: 92H
MKPATH

OPENUP Offset: 00006d10H Code and data size: 66aH
FCBTMPOFF FCBTMPSEG IUNITOFF IUNITSEG
MXRECOFF

OPNCOM Offset: 00007300H Code and data size: 108H
OPNCOM

OPNNEW Offset: 000074b0H Code and data size: 25H
OPNNEW

OPNOLD Offset: 00007530H Code and data size: 24H
OPNOLD

PACKL2 Offset: 000075b0H Code and data size: 7bH
PACKL

PACKR2 Offset: 000076c0H Code and data size: 88H
PACKR

PAINTA2 Offset: 000077f0H Code and data size: bbcH
PAINTA

PEEK Offset: 000085d0H Code and data size: 33H

POKE Offset: 00008660H Code and data size: 31H

PORTIN2 Offset: 000086f0H Code and data size: 37H
PORTIN

PORTOT2 Offset: 00008780H Code and data size: 34H
PORTOT

PRNASM Offset: 00008810H Code and data size: 33H
PRNASM

PRTSCR Offset: 000088a0H Code and data size: 24H
PRTSCR

PUTCUR Offset: 00008920H Code and data size: e9H
PUTCUR

RDLINE Offset: 00008b60H Code and data size: 5dH
RDLINE

RDOT Offset: 00008c60H Code and data size: 13aH
RDOT

RDRND Offset: 00008f30H Code and data size: 338H
RDRND

RENFLE3 Offset: 00009610H Code and data size: 172H
RENFLE

RMDIR Offset: 00009870H Code and data size: 2eH
RMDIR

RNFLEZ Offset: 00009900H Code and data size: 20H
RNFLEZ

SCROLL Offset: 00009980H Code and data size: 137H
SCROLL

SEGAAL Offset: 00009be0H Code and data size: 56H
SEGAAL

SEGAP1 Offset: 00009c90H Code and data size: 75H
SEGAP1

SELDSK Offset: 00009d80H Code and data size: 25H
SELDSK

STVRFY Offset: 00009e00H Code and data size: 21H
STVRFY

TEGAIB Offset: 00009e80H Code and data size: 44H
TEGAIB

TIMASM Offset: 00009f30H Code and data size: 2bH
TIMASM

TIMEOD2 Offset: 00009fc0H Code and data size: 58H
TIMEOD

UCMSTG Offset: 0000a090H Code and data size: 93H
UCMSTG

UPCASE2 Offset: 0000a1e0H Code and data size: 39H
UPCASE

WRTRND Offset: 0000a280H Code and data size: 3ebH
WRTRND

XQTDS2 Offset: 0000aa80H Code and data size: 33aH
XQTDS2

XQTPM2 Offset: 0000ae10H Code and data size: 2f4H
XQTPM2

ZFBLNK2 Offset: 0000b120H Code and data size: 35H
ZFBLNK

gtpath Offset: 0000b1b0H Code and data size: 5H
GTPATH

setbrd Offset: 0000b270H Code and data size: 55H
SETBRD

wdot Offset: 0000b340H Code and data size: 186H
WDOT

segao1 Offset: 0000b6c0H Code and data size: 54H
SEGAO1

bcaray Offset: 0000b790H Code and data size: 2c6H
BCARAY

screen Offset: 0000bd10H Code and data size: 1c5H
SCREEN

frshlw Offset: 0000c090H Code and data size: 1dH
FRSHLW

frshlb Offset: 0000c110H Code and data size: 1dH
FRSHLB

frshrw Offset: 0000c190H Code and data size: 1dH
FRSHRW

frshrb Offset: 0000c210H Code and data size: 1dH
FRSHRB

frrolw Offset: 0000c290H Code and data size: 1dH
FRROLW

frrolb Offset: 0000c310H Code and data size: 1dH
FRROLB

frrorw Offset: 0000c390H Code and data size: 1dH
FRRORW

frrorb Offset: 0000c410H Code and data size: 1dH
FRRORB

insert Offset: 0000c490H Code and data size: e0H
INSERT

rchar Offset: 0000c6a0H Code and data size: 11fH
RCHAR

herc10 Offset: 0000c8e0H Code and data size: 1a33H
HRCCLS HRCOPN

wrtrn2 Offset: 0000eab0H Code and data size: 230H
WRTRN2

rdrnd2 Offset: 0000ef40H Code and data size: 230H
RDRND2

wchar2 Offset: 0000f3d0H Code and data size: 141H
WCHAR

wchars Offset: 0000f6a0H Code and data size: 1aaH
WCHARS

beep2 Offset: 0000fa30H Code and data size: 6dH
BEEP2

rotate Offset: 0000fb00H Code and data size: 316H
ROTATE

putscr Offset: 00010190H Code and data size: 2e4H
PUTSCR

getscr Offset: 00010800H Code and data size: 2a4H
GETSCR

xqtpgm Offset: 00010dc0H Code and data size: 2ddH
XQTPGM

xqtdos Offset: 00011090H Code and data size: 30bH
XQTDOS

dirinf Offset: 000113b0H Code and data size: f4H
DIRINF

icomsub3 Offset: 00011540H Code and data size: f70H
CBKOFF CBRKON CLRIN CLROUT
CLSCOM COMLEN INCOM IOPCOM
OUTCOM OUTLEN

outport Offset: 000123b0H Code and data size: 6eH

```
  OUTPORT

outsfns             Offset: 00012550H  Code and data size: 8cH
  OUTSFNS

pausa               Offset: 00012720H  Code and data size: 9fH
  PAUSA

pcstat              Offset: 00012940H  Code and data size: 300H
  PCSTAT

astro               Offset: 00012ec0H  Code and data size: 30ddH
  ABERR               DATE               HMS                JULIAN
  MOON                NUTATION           SIDG               SUNPOS

sclock              Offset: 00018fc0H  Code and data size: 256H
  SCLOCK

correct             Offset: 000193e0H  Code and data size: 59fH
  CORRE               CORRECT

fasens              Offset: 00019820H  Code and data size: 2ebH
  FASENS

line                Offset: 00019e10H  Code and data size: f22H
  LINE

rdhead              Offset: 0001b440H  Code and data size: b98H
  RDHEAD

nctel1              Offset: 0001b610H  Code and data size: a8bH
  NC_TEL_ALLOCATE   NC_TEL_DEALLOCATE                    OPEN_COM1
  PAUSE             PCEND              PCPOINT

uclock              Offset: 0001c7d0H  Code and data size: 25eH
  UCLOCK

conv                Offset: 0001cc00H  Code and data size: 20aH
  CONV

clock               Offset: 0001d0b0H  Code and data size: 400H
  CLOCK

clocksid            Offset: 0001d810H  Code and data size: 3f8H
  CLOCKSID

suconv              Offset: 0001df70H  Code and data size: 12d4H
  CONVERSION

tapec               Offset: 0001ed00H  Code and data size: 1388H
  DECODE              EXTRD              EXTWR              FSFWHIGHSP
  FSLOWSP             FSRHIGHSP          FSRLOWSP           FW1BLK
  READ                REW                RV1BLK             SBYTES
  WRFMK               WRITE
```

BIBLIOGRAFIA

1) Rapporto interno IRA 105/88 "Acquisizione dati per il radiotelescopio 'Croce del Nord'"
S. Montebugnoli, A. Cattani

2) Rapporto interno IRA 106/88 "Orologio digitale per il sistema di acquisizione dati del radiotelescopio 'Croce del Nord'"
S. Montebugnoli, A. Cattani

3) Rapporto interno IRA 107/88 "Sistema modulare di interfacciamento con personal computer:
-interfacce per orologi digitali
-interfaccia con rele` di controllo"
S. Montebugnoli, A. Cattani

4) Rapporto interno IRA 130/90 "Scheda di interfaccia per il computer IBM XT/AT di gestione del radiotelescopio 'Croce del Nord'"
A. Cattani, A. Maccaferri, S. Montebugnoli

NOTE