

```

EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSSS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MM MM MM PP PP LL EE SS
EE XX XX AA AA MM MM MM PP PP LL EE SS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSSS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS

```

```
LL          AAAAAA  BBBB BBBB  IIIIII  000000  AAAAAA  CCCCCCCC  QQQQQQ
LL          AAAAAA  BBBB BBBB  IIIIII  000000  AAAAAA  CCCCCCCC  QQQQQQ
LL          AA      AA  BB      BB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BB      BB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BB      BB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BB      BB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LL          AA      AA  BBBB BBBB  II      00      00  AA      AA  CC      QQ      QQ
LLLLLLLLLL AA      AA  BBBB BBBB  IIIIII  000000  AA      AA  CCCCCCCC  QQQQ  QQ
LLLLLLLLLL AA      AA  BBBB BBBB  IIIIII  000000  AA      AA  CCCCCCCC  QQQQ  QQ
```

```
FFFFFFFFFF  000000  RRRRRRRR
FFFFFFFFFF  000000  RRRRRRRR
FF          00      00  RR      RR
FF          00      00  RR      RR
FF          00      00  RR      RR
FF          00      00  RR      RR
FFFFFFFFFF  00      00  RRRRRRRR
FFFFFFFFFF  00      00  RRRRRRRR
FF          00      00  RR      RR
FF          00      00  RR      RR
FF          00      00  RR      RR
FF          00      00  RR      RR
FF          00      00  RR      RR
FF          00      00  RR      RR
FF          00      00  RR      RR
FF          000000  RR      RR
FF          000000  RR      RR
```

LAE
: G
: 10
: I
: D
: S
: 11
: D
: 12
: I
: 13
: C
: 21
: D
: 22


```
!File: LABIOACQ.FOR
      Version 'V04-000'
```

```
*****
*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

```
Program LABIO_DATA_ACQ
```

```
! This is the program that acquires data for the LABIO system
! It uses the connect-to-interrupt feature of VMS to acquire
! via a user written I/O routine. The actual I/O routine is
! written in MACRO. The main program monitors the event flags
! and enables and disables data acquisition for each channel.
! It also notifies users via event flags when a buffer is full.
```

```
! Define the LABIO data base
```

```
      Include 'LABCHNDEF.FOR'
```

```
! Local Variables
```

```
      Logical*4 SECTION_FLAGS, SECTION_PROT
```

```
! System Services
```

```
      Logical*4 SYSS$ASCEFC, SYSS$MGBLSC, SYSS$ASSIGN, SYSS$QIO
      Logical*4 SYSS$CLREF
```

```
! External constants
```

```
      External SEC$M_GBL, SEC$M_WRT, SSS$_CREATED, SSS$_WASSET
      External SET_EF_ASI
```

```
! Misc.
```

```
      Logical*4 AD_CIN_UP, SUCCESS
```

```
! Create the Global Section for the data buffer
```



```
! This data buffer will be READ/WRITE for the owner, READ only for the GROUP.
```

```
! First see if the global section already exists, if it
! does just map to it. and set the restart flag.
```

```
! If not, Open the Data File. This can not be opened
! via FORTRAN since we need the VMS channel number.
```

```
SECTION(1) = %Loc( LABIO_BUFFER_S)      !Start address of section
SECTION(2) = %Loc( LABIO_BUFFER_E) - 1 !End address
```

```
! Page count for the section
SECTION_SIZE = ( SECTION(2) - SECTION(1) )/512 + 1
```

```
! FLAGS for Section are GLOABAL,SHARED,NON_ZEROED,READ/WRITE,TEMP,GLOBAL
```

```
SECTION_FLAGS = %Loc( SEC$M_GBL ) + %Loc( SEC$M_WRT )
```

```
! Try just mapping to the global section
```

```
SUCCESS = SYS$MGBLSC( SECTION,,,%Val(SECTION_FLAGS),'LABIOCOMMON',,)
```

```
If( SUCCESS ) Then
```

```
RESTART = .TRUE.      !Succes, this is a restart
```

```
Else
```

```
SUCCESS = GBL_SECTION_UFO( SECTION_SIZE, 'LABIO_SEC_FILE',
SECTION_CHANNEL )
```

```
1 If( .not. SUCCESS )
```

```
1 Call FATAL_ERROR(SUCCESS,'Opening Global Section File')
```

```
! PROTECTION is OWNER = READ/WRITE, GROUP = READ, SYSTEM/WORLD = none
```

```
SECTION_PROT = 'F E 0 F'X !Protection for section
```

```
! Create and Map the Section
```

```
SUCCESS = SYS$CRMPSC( SECTION,,,%Val(SECTION_FLAGS),'LABIOCOMMON',
```

```
1 ,,%Val(SECTION_CHANNEL),%Val(SECTION_SIZE),,
```

```
1 ,,%Val(SECTION_PROT),%Val(SECTION_SIZE)
```

```
If( .not. SUCCESS )
```

```
1 Call FATAL_ERROR(SUCCESS,'Creating Global Section')
```

```
RESTART = .FALSE.      !We are not restarting
```

```
End If
```

```
! If this is not a restart, clear the data structures
```

```
If( .not. RESTART ) Then
```

```
Do 32 I = 1, MAX_AD_CHANNEL      !Clear AD_BLOCK
```

```
Do 30 J = 1, 16
```

```
AD_BLOCK(J,I) = 0
```

```
Do 31 K = 1, BUFFER COUNT      !Clear Data buffers
```

```
Do 31 J = 1, MAX_BUF_SIZE
```

```
DATA_BUFFER(J,K,I) = 0
```

```
Continue
```

```
Do 33 I = 1, MAX_PID
```

```
Do 33 J = 1, 2
```

```
CONNECT_BLOCK(I,J) = 0      !Clear Process connect block
```

```
End If
```

30

31

32

33


```
! Create event flag cluster EF_NOTIFY and associate with event flags 64-95
! These are used to notify the Data Acquisition process.
```

```
    SUCCESS = SYSSASCEFC( %VAL(EF_NOTIFY_1),EF_NOTIFY_CLSTR,,)
    If ( .not. SUCCESS)
    1 Call FATAL_ERROR( SUCCESS, 'CREATING EVENT FLAG CLUSTER')
```

```
! Create event flag cluster EF_STATUS and associate with event flags 96-127
! These are used to notify and report the status of the user buffers
```

```
    SUCCESS = SYSSASCEFC( %VAL(EF_STATUS_1),EF_STATUS_CLSTR,,)
    If ( .not. SUCCESS)
    1 Call FATAL_ERROR( SUCCESS, 'CREATING EVENT FLAG CLUSTER')
```

```
! Make sure that we can't be swapped
```

```
    Call SYSSSETSWM(%Val(1))
```

```
! Set-up the Connect-to-Interrupt
```

```
! First assign a VMS channel for the device
! Then call the connect-to-interrupt setup routine.
```

```
    SUCCESS = SYSSASSIGN( 'LABIO_AD',CIN_CHANNEL,, )
    If ( .not. SUCCESS )
    1 Call FATAL_ERROR( SUCCESS, 'assigning A/D device' )
```

```
    SUCCESS = AD_CIN_SETUP( CIN_CHANNEL,SET_EF_AST )
    If( .not. SUCCESS )
    1 Call FATAL_ERROR( SUCCESS, 'connecting-to-interrupt')
```

```
! End Of Initialization, Notify other processes by setting EF_DATA_ACQ
```

```
    Call SYSSSETEF( %Val( EF_DATA_ACQ ) )
```

```
!
! Wait for an event flag in the EF_NOTIFY cluster
! Then read the EF_NOTIFY CLUSTER and EF_STATUS_CLUSTER
10  Call SYSSWFLOr( %Val(EF_NOTIFY_1) , %Val('FFFF'X) )
!
! Look for the flag(s) set in EF_NOTIFY
! If the corresponding activity flag is set, activate the channel,
! otherwise deactivate it. Also check the buffer status flag, if clear
! clear the buffer index.
!
  Do 20 I = 1,16
  If( SYSSCLREF( %Val(EF_NOTIFY_OFF + I)) .eq. %Loc(SS$_WASSET)) Then
    If( AD_BLOCK(1,I) .ne. 0 ) Then
      If( SYSSREDEF( %Val(EF_ACTIVITY_OFF + I),EF_STATE )
        1 .eq. %Loc(SS$_WASSET) ) Then
          AD_BLOCK(1,I) = ACTIVE
        Else
          AD_BLOCK(1,I) = INACTIVE
        End if
      If( SYSSREDEF( %Val(EF_STATUS_OFF + I),EF_STATE )
        1 .eq. %Loc(SS$_WASCLR) ) AD_BLOCK(7,I) = 0
      End If
    End If
  Continue
  Go To 10
End
```

20


```
Subroutine SET_EF_AST( EVENT_FLAGS )
```

```
! This is a AST routine which is invoked by the  
! Interrupt service routine. This routine sets  
! the event flags indicated by the ISR.
```

```
Include 'LABCHNDEF.FOR'  
Integer EVENT_FLAGS
```

```
! The Event flags are set in cluster EF_STATUS_CLSTR
```

```
10 Do 10 I = 1,16  
   If( (EVENT_FLAGS .and. BIT(I)) .ne. 0 )  
     ? Call SYS$SETEF( %Val(EF_STATUS_OFF + I) )  
   Continue  
Return
```

```
End  
![End of File]
```


XALINK
MAR

LABIOLINK
COM

DRMASTER
FOR

LPATEST
FOR

LABTOPEAK
FOR

LABIOSTR
COM

XMESSAGE
MAR

XATEST
FOR

LABTOCOM
FOR

LABDEMO
COM

LABMBXDEF
FOR

LABTOSAMP
FOR

MAILCOMPRESS
COM

LABCHNDEF
FOR

CONNECT
COM

LABTOCON
FOR

LABDEMO
FOR

PEAK
FOR

DRCOPYBLD
COM

XIDRIVER
MAR

LABTOSPEC
FOR

DRSLAVE
FOR

LABTOACQ
FOR

LABTOCOMP
COM

LABIOSTAT
FOR

TESTLABIO
FOR