```
      SUBROUTINE TOFDAT
C***********************************************************************
C
C     This is routine getting tof barrel counter's charge, time, z e
c     coordinate of hit on the counter and expected time of flight.
C
C                                         LIU QI 8/26/91
C
C
C     Modified the routine from VAX to HP-UNIX, working with routines in
C     new CERN program libary (HBK4).
C                                    G. RONG    September, 1995
C
C
C     TRK(ISUB11+33) is used is used to calculate the expected path
C     of a track hitting endcap tof.     Originally, TRK(ISUB11+32)
C     was used.
C                                   Xie  Yuehong  Mar 22, 1999

C     remove T0 obtained in barrel calibration from traw.
C                                   Xie Yuehong  June 11, 1999

*     tofcalb.inc is renamed as caltof.inc to include ETOF calibration
*                              Guangpei Chen     Sep.1, 1999
C**********************************************************************
*     Common blocks:
      INCLUDE 'rawdat.inc'
      INCLUDE 'evlst.inc'
      INCLUDE 'trklst.inc'
      INCLUDE 'trk5pt.inc'
      INCLUDE 'evsubs.inc'
      INCLUDE 'tofcon.inc'
      INCLUDE 'toflst.inc'
      INCLUDE 'toftof.inc'
      INCLUDE 'numcon.inc'
      INCLUDE 'tofcal.inc'
      INCLUDE 'contrl.inc'
      INCLUDE 'calpas.inc'
      INCLUDE 'caltof.inc'

*     Local variables:
      INTEGER SIGID
      INTEGER ZRHALF

      REAL*4 ETOTEBEAM
      LOGICAL*1 DCTRK, VCTRK
C
      DIMENSION IBACK(2),PATH(2),Texp(2)
      DIMENSION ICN(2),Qr1(2),Tr1_e(2),Qr2(2),Tr2_e(2),ZZ(2)
      DIMENSION IECN(2),QE(2),TE(2),RE(2)
      DATA TEZF,TERI,TERO/1.2107,0.35,1.05/
      DATA T1,T2/800.,4000./
      DATA ID/0/
      DATA ZRHALF /'0000FFFF'X/

C     Used for tottal energy cut:
      NCALPAS(1) = NCALPAS(1) + 1             ! Total number of events
      IF(NCHRG.NE.2)RETURN
      NCALPAS(2) = NCALPAS(2) + 1             ! N2

      PX=0.
      PY=0.
      PZ=0.
      ETOT=0.
```

```
      IDSC=0
      DO ITRACK=1,2
        ISUB1 = TRKSUB(1,ITRACK)
        IF(ISUB1.LE.0)RETURN

*       Guangpei Chen    Mar.2 1999
*       Add VC track:
        DCTRK = IAND(ITRK(ISUB1+5),1).EQ.1
        VCTRK = IAND(ITRK(ISUB1+5),'100'X).EQ.'100'X
        IF(.NOT.(DCTRK .OR. VCTRK))RETURN

        ISUB3 = TRKSUB(3,ITRACK)
        IF(ISUB3.LE.0)RETURN
        MFI=ITRK(ISUB3+14)
        IF(MFI.NE.2)RETURN
        P=TRK(ISUB3+7)
        PX=PX+TRK(ISUB3+4)
        PY=PY+TRK(ISUB3+5)
        PZ=PZ+TRK(ISUB3+6)
        ETOT=ETOT+P
        ISUB5 = TRKSUB(5,ITRACK)
        IF(ISUB5.GT.0)THEN
          IHSC=ITRK(ISUB5+SCPHIT)
          IF(IHSC.GE.25)IDSC=1
        END IF
      END DO
      NCALPAS(3) = NCALPAS(3) + 1             ! N good track

      PTOT=SQRT(PX**2+PY**2+PZ**2)
      IF(PTOT.GE.0.6)RETURN
      NCALPAS(4) = NCALPAS(4) + 1             ! N Ptot

*     Guangpei Chen   Mar.2, 1999
*     Scale ETOT to EBEAM:
      ETOTEBEAM = ETOT/EBEAM
      IF(ETOTEBEAM.LE.1.0.OR.ETOTEBEAM.GE.7.0)RETURN
      NCALPAS(5) = NCALPAS(5) + 1             ! N Etot

C     Unpack TOF raw data:
      CALL CLRMEM(NTOFCT(1),96)
      NP=IRAW(11)
      DO K=1,NP
        JTYPE = IAND(ISHFT(IRAW(11+K),-16),ZRHALF) ! RG 14/9/95
        IF(JTYPE .EQ. 1) GO TO 100            ! RG 14/9/95
      END DO
      RETURN
 100  NM=K+11
      NCALPAS(6) = NCALPAS(6) + 1             ! With TOF data

      IP1 = IAND(IRAW(11+K),ZRHALF)           ! RG 14/9/95
      NWD = IAND(IRAW(IP1),ZRHALF)/4 -1
      IF(NWD .LE. 0) RETURN
C     Loop over TOF data here:
      DO 3 N = 1, NWD
        LOC=IP1+N
        IDAT = IAND(ISHFT(IRAW(LOC),-16),ZRHALF)
        CALL UTFTAG(IDAT,ICNTR,IEND,SIGID)
        IF(SIGID .NE. 1 .AND. SIGID .NE. 2) GO TO 3
        IF(ICNTR .LE. 0 .OR. ICNTR .GE. 97 .OR. IEND.LE.0 .OR. IEND
     &    .GE.3) THEN
          WRITE(6,3001) ICNTR,IEND
 3001     FORMAT(' WRONG COUNT',2I6)
          GO TO 3
        ENDIF

C       + RG 14/9/95
```

```
C       ADC data (in the rightmost 16 bits of IRAW word).
        IADC = IAND(IRAW(LOC),ZRHALF)          ! ??
        IF(SIGID .EQ. 1) THEN
          Q = FLOAT(IAND(IRAW(LOC),ZRHALF))    ! ??
C         - RG 14/9/95

          IF(ICNTR .LE. 48)THEN
            TOF(1,IEND,ICNTR) = Q
          ELSE
            TOF(1,1,ICNTR)=Q
            TOF(1,2,ICNTR)=Q
          ENDIF
        ELSEIF( SIGID .EQ. 2 )THEN

C         RAW CHARGE:
C         + RG  14/9/95
C         TDC data (in the rightmost 16 bits of IRAW word).
          T = FLOAT(IAND(IRAW(LOC),ZRHALF))    !  ??
C         - RG 14/9/95

          IF(T .GT.T1 .AND. T .LT.T2)THEN
            IF(ICNTR .LE. 48)THEN
              TOF(2,IEND,ICNTR) = T
              Q=TOF(1,IEND,ICNTR)
            ELSE
              TOF(2,1,ICNTR)=T
              TOF(2,2,ICNTR)=T
              Q=TOF(1,1,ICNTR)
            ENDIF
            IF(Q.GT.100..AND.Q.LT.6000.)THEN
              IF( NTOFCT(ICNTR) .EQ. 0 )THEN
                NTOFCT(ICNTR)=1
              ELSEIF( NTOFCT(ICNTR) .EQ. 1 )THEN
                NTOFCT(ICNTR)=2
              END IF
            ENDIF
          END IF
        END IF
 3    CONTINUE

 999  Format('IRUN  IREC  NWD  N  Q  T = ',7(2xI6),3x,F7.2,2xF7.2)

      ICN(1)=0
      ICN(2)=0
      IECN(1)=0
      IECN(2)=0
      IBACK(1)=0
      IBACK(2)=0
      IDBARREL=0
      IDENDCAP=0
      DO ITRACK=1,2
        ISUB1 = TRKSUB(1,ITRACK)
        ISUB3 = TRKSUB(3,ITRACK)
        ISUB11= TRKSUB(11,ITRACK)
        IF(ISUB11.LE.0)GO TO 200
        ZZZZ=TRK(ISUB11+4)
        IF(ABS(ZZZZ).GT.1.40)GO TO 200
        ZZ(ITRACK)=ZZZZ
        PATH(ITRACK)=TRK(ISUB11+32)
        IF(PATH(ITRACK).GE.900.0)GO TO 200
C       Expected time of flight:
        Texp(ITRACK)=PATH(ITRACK)/VELC
        IF(TRK(ISUB11+2).GE.900.0.OR.
     <    TRK(ISUB11+3).GE.900.0)GO TO 200
C       Sensible X and Y extrapolation
C       The PHI angle of the track on TOF counter is
```

```
        PHIT=XATAN(TRK(ISUB11+3),TRK(ISUB11+2))
        ACNTR = 48*PHIT/TWOPI + 1.0
        IF (ACNTR.GE.49) ACNTR = ACNTR -48.
        ICNTR = ACNTR
        IBACK(ITRACK)=ICNTR
        ICNTR1=ICNTR-1
        IF(ICNTR.EQ.1)ICNTR1=48
        ICNTR2=ICNTR+1
        IF(ICNTR.EQ.48)ICNTR2=1
        IF(NTOFCT(ICNTR).EQ.2)THEN
          IDBARREL=1
          ICN(ITRACK)=ICNTR
C         Charges of east and west ends:
          Qr1(ITRACK)=TOF(1,1,ICNTR)
          Qr2(ITRACK)=TOF(1,2,ICNTR)
C         Difference of raw time and expected time of flight:
          Tr1_e(ITRACK)=TOF(2,1,ICNTR)/70.-Texp(ITRACK)
          Tr2_e(ITRACK)=TOF(2,2,ICNTR)/70.-Texp(ITRACK)
        ELSEIF(NTOFCT(ICNTR1).EQ.2)THEN
          IDBARREL=1
          ICN(ITRACK)=ICNTR1
C         Charges of east and west ends:
          Qr1(ITRACK)=TOF(1,1,ICNTR1)
          Qr2(ITRACK)=TOF(1,2,ICNTR1)
C         Difference of raw time and expected time of flight:
          Tr1_e(ITRACK)=TOF(2,1,ICNTR1)/70.-Texp(ITRACK)
          Tr2_e(ITRACK)=TOF(2,2,ICNTR1)/70.-Texp(ITRACK)
        ELSEIF(NTOFCT(ICNTR2).EQ.2)THEN
          IDBARREL=1
          ICN(ITRACK)=ICNTR2
C         Charges of east and west ends:
          Qr1(ITRACK)=TOF(1,1,ICNTR2)
          Qr2(ITRACK)=TOF(1,2,ICNTR2)
C         Difference of raw time and expected time of flight:
          Tr1_e(ITRACK)=TOF(2,1,ICNTR2)/70.-Texp(ITRACK)
          Tr2_e(ITRACK)=TOF(2,2,ICNTR2)/70.-Texp(ITRACK)
        END IF

C       Following use D.C. to extrapolates tracks to the endcap tof cou
+nters:

 200    IF(TRK(ISUB11+14).GE.900.0.AND.
     <    TRK(ISUB11+15).GE.900.0)GO TO 300
C       Sensible X and Y extrapolation

C       Added by  Xie Yuehong, Mar 22,1999
        PATH(ITRACK)=TRK(ISUB11+33)
        IF(PATH(ITRACK).GE.900.0)GO TO 300
C       Expected time of flight:
        Texp(ITRACK)=PATH(ITRACK)/VELC

C       The PHI angle of the track on end cap tof counter is
        PHIT=XATAN(TRK(ISUB11+15),TRK(ISUB11+14))
        R=SQRT(TRK(ISUB11+14)**2+TRK(ISUB11+15)**2)
        RE(ITRACK)=R
        IF(R.LE.TERI.OR.R.GE.TERO)GO TO 300
        IF(ISUB11.EQ.0)THEN
          ZD=999.0
        ELSE
          ZD=TRK(ISUB11+16)
        END IF
        ACNTR1=0.
        IF(ZD.LT.999..AND.ZD.GT.0.0)THEN
C         Track is extrapolated to the east end cap counter
          ACNTR1 = 24*PHIT/TWOPI + 49.0
          IF(ACNTR1.GE.73) ACNTR1 = ACNTR1 -24.
```

```
        ELSEIF(ZD.LT.0.0)THEN
C         Track is extrapolated to the west end cap counter
          ACNTR1 = 24*PHIT/TWOPI + 73.0
          IF(ACNTR1.GE.97) ACNTR1 = ACNTR1 -24.
        END IF
        IF(ACNTR1.LT.1.)GO TO 300
        ICNTR=ACNTR1
        IF (NTOFCT(ICNTR).EQ.1) THEN
          IDENDCAP=1
          IECN(ITRACK)=ICNTR
          QE(ITRACK)=TOF(1,1,ICNTR)
          TE(ITRACK)=TOF(2,1,ICNTR)/70.-Texp(ITRACK)
        ELSEIF(ICNTR.LE.72)THEN
          ICNTR1=ICNTR+1
          ICNTR2=ICNTR-1
          IF(ICNTR.EQ.72)ICNTR1=49
          IF(ICNTR.EQ.49)ICNTR2=72
          IF(NTOFCT(ICNTR1).EQ.1)THEN
            IDENDCAP=1
            IECN(ITRACK)=ICNTR1
            QE(ITRACK)=TOF(1,1,ICNTR1)
            TE(ITRACK)=TOF(2,1,ICNTR1)/70.-Texp(ITRACK)
          ELSEIF(NTOFCT(ICNTR2).EQ.1)THEN
            IDENDCAP=1
            IECN(ITRACK)=ICNTR2
            QE(ITRACK)=TOF(1,1,ICNTR2)
            TE(ITRACK)=TOF(2,1,ICNTR2)/70.-Texp(ITRACK)
          END IF
        ELSEIF(ICNTR.LE.96)THEN
          ICNTR1=ICNTR+1
          ICNTR2=ICNTR-1
          IF(ICNTR.EQ.96)ICNTR1=73
          IF(ICNTR.EQ.73)ICNTR2=96
          IF(NTOFCT(ICNTR1).EQ.1)THEN
            IDENDCAP=1
            IECN(ITRACK)=ICNTR1
            QE(ITRACK)=TOF(1,1,ICNTR1)
            TE(ITRACK)=TOF(2,1,ICNTR1)/70.-Texp(ITRACK)
          ELSEIF(NTOFCT(ICNTR2).EQ.1)THEN
            IDENDCAP=1
            IECN(ITRACK)=ICNTR2
            QE(ITRACK)=TOF(1,1,ICNTR2)
            TE(ITRACK)=TOF(2,1,ICNTR2)/70.-Texp(ITRACK)
          END IF
        END IF
 300  END DO
      IF(ID.EQ.0)THEN
        IRUN0=IRUN
        ID=100
      END IF
      RUNS(1)=IRUN0
      RUNS(2)=IRUN-IRUN0+1
      KRUN=RUNS(2)
      IDC=ABS(IBACK(2)-IBACK(1))-24
      IF(ABS(IDC).GT.1)IDBARREL=0
      IF(IDBARREL.EQ.1.AND.IDSC.GE.1)THEN
        IEV=IEV+1

C       To fill arrays  which is used to determine barrel countr attenu
+ation
C       lenth : ALEN
        DO IBC=1,2
          IC=ICN(IBC)
          IF(IC.EQ.0)GO TO 1111
          IF(Qr1(IBC).GE.3600..OR.Qr2(IBC).GE.3600.)GO TO 2222
          IF(ABS(ZZ(IBC)).GE.1.0)GO TO 2222
```

```
          X=ALOG(Qr1(IBC)/Qr2(IBC))
          AL(1,1,IC)=AL(1,1,IC)+1.
          AL(1,2,IC)=AL(1,2,IC)+ZZ(IBC)
          AL(2,2,IC)=AL(2,2,IC)+ZZ(IBC)*ZZ(IBC)
          BL(1,IC)=BL(1,IC)+X
          BL(2,IC)=BL(2,IC)+X*ZZ(IBC)

C         To fill arrays  which is used to determine QZERO
          I=INT((ZZ(IBC)+1.42)/0.02)+1
          IF(I.GT.142)I=142
          IF(I.LT.1)I=1
          TQ(I,1,IC)=TQ(I,1,IC)+Qr1(IBC)
          TQ(I,2,IC)=TQ(I,2,IC)+Qr2(IBC)
 2222     IF(KRUN.GT.100.OR.IEV.GT.35000)RETURN
          NRUN=KRUN

C         To fill arrays with raw data
          TQZ(1,IBC,IEV)=Qr1(IBC)
C         Difference of counter east end raw time and expected time of
+flight:
          TQZ(2,IBC,IEV)=Tr1_e(IBC)
C         Counter west end raw charge:
          TQZ(3,IBC,IEV)=Qr2(IBC)
C         Difference of counter west end raw time and expected time of
+flight:
          TQZ(4,IBC,IEV)=Tr2_e(IBC)
C         Z-coordinate of hit on counter:
          TQZ(5,IBC,IEV)=ZZ(IBC)
C         TOF counter number :
          ICNT(IBC,IEV)=IC
          NH1(IC,KRUN)=NH1(IC,KRUN)+1
          NH2(IC)=NH2(IC)+1
 1111   END DO
        IR(IEV)=KRUN
      END IF
      NCALPAS(7) = NCALPAS(7) + 1             ! N BTOF

      IF(IDENDCAP.EQ.1)THEN
        DO IBC=1,2
          IC=IECN(IBC)
          IF(IC.GT.48.AND.IC.LE.96)THEN
            IF(IHITE(IC).LT.5000)THEN
              IHITE(IC)=IHITE(IC)+1
              K=IHITE(IC)
              TERAW(1,IC,K)=TE(IBC)-OFFSET(KRUN)
              TERAW(2,IC,K)=QE(IBC)
              TERAW(3,IC,K)=RE(IBC)
            END IF
          END IF
        END DO
      END IF

      END
```

```
      SUBROUTINE TOFCAL
C**********************************************************************
C
C     This is tof offline calibration main routine determing tof
C     correctio constants, time zero offset of different run, tof
C     counters attenuation lenth and qzero.
C
C                                                    LIU QI  8/26/91
C
*     Calibrate BTOF/ETOF separatelly
*     Guangpei Chen       Sep.2, 1999
C**********************************************************************

*     Common blocks:
      INCLUDE 'caltof.inc'

*     To calibrate BTOF:
      IF(TOFTYPE .EQ. 1)THEN
*       To determine tof barrel counters' attenuation length: ALEN:
        CALL TOFBAL

*       To determin barrel tof offline correction constants:
*       CALL TOFBP               ! ????

*       To book 1_dim histgram:
        CALL TOFBOK

*       To refill some common blocks with selected data:
        CALL TOFCUT

*       To determin barrel tof offline correction constants:
        CALL TOFBP

*       To determine time weighted average parameters:
        CALL TOFWTP

*       To determine tof barrel counters' time deviations:
        CALL TOFSIG

*---------------------------------------------------------------------------
*     To calibrate ETOF:
      ELSE IF(TOFTYPE .EQ. 2)THEN
*       To determin endcap tof offline correction constants:
        CALL TOFEP
      ENDIF

*     MAOZP--97/1/25/, to output constants:
      CALL CONSTW(1, IERR)

      END
```

```
      SUBROUTINE TOFBAL
C
C*******************************************************************
C
C     To determine tof barrel counters` optical attenuation length: and
+
C     ALEN mean normalized charge:QZERO.
C                                                  LIU QI 8/26/91
C
*     tofcalb.inc is renamed as caltof.inc to include ETOF calibration
*                              Guangpei Chen     Sep.1, 1999
C*******************************************************************
C
      INCLUDE 'tofcal.inc'
      INCLUDE 'caltof.inc'
C
      DO IC=1,48
        A11=AL(1,1,IC)
        A12=AL(1,2,IC)
        A22=AL(2,2,IC)
        B1=BL(1,IC)
        B2=BL(2,IC)
        IF(A11.GE.NENTRYMIN)THEN
          ALEN(IC)=2.*(A11*A22-A12*A12)/(A11*B2-B1*A12)
          QZERO(1,IC)=0.
          QZERO(2,IC)=0.
          DO I=1,142
            Z=-1.42+0.01*(2*I-1)
            QZERO(1,IC)=QZERO(1,IC)+TQ(I,1,IC)*EXP(-Z/ALEN(IC))*
     #                              1.157/SQRT(1.157**2+Z**2)
            QZERO(2,IC)=QZERO(2,IC)+TQ(I,2,IC)*EXP(Z/ALEN(IC))*
     #                              1.157/SQRT(1.157**2+Z**2)
          END DO
          QZERO(1,IC)=QZERO(1,IC)/A11
          QZERO(2,IC)=QZERO(2,IC)/A11
        ELSE
          ALEN(IC)=3.0          ! Changed from 2.0    Guangpei Chen  Ma
+r.2 1999
          QZERO(1,IC)=1000.
          QZERO(2,IC)=1000.
        END IF
      END DO
      RETURN
      END
```

```
      SUBROUTINE TOFBOK
C
C*********************************************************************
C
C     To book some histgrams
C                                         LIU QI 8/26/91
C
*     tofcalb.inc is renamed as caltof.inc to include ETOF calibration.
*     Book histograms for ETOF here too.
*                              Guangpei Chen     Sep.1, 1999
C*********************************************************************
C
      INCLUDE 'tofcal.inc'
      INCLUDE 'caltof.inc'
      DATA AN/0./

      DO I=1,MIN(IEV,35000)
C       Run number:
        IRUN=IR(I)
        IF(IRUN.GT.0)THEN
          DO IBC=1,2
            IF(ICNT(IBC,I).GT.0)THEN
C             Cunter number:
              IC=ICNT(IBC,I)
              IF(NH2(IC).GE.NENTRYMIN)THEN
C               East end raw charge:
                Q1=TQZ(1,IBC,I)
C               Difference of east end raw time and expected time of fi
+lght:
                T1=TQZ(2,IBC,I)
C               West end raw charge:
                Q2=TQZ(3,IBC,I)
C               Difference of west end raw time and expected time of fi
+lght:
                T2=TQZ(4,IBC,I)
C               Z-coordinate of hit on the tof counter:
                Z =TQZ(5,IBC,I)
C               East end random time err determined by corrected tof co
+nsts:
                GT1=TOFDEV(1,IC,Q1,T1,Z,IRUN)
C               West end random time err determined by corrected tof co
+nsts:
                GT2=TOFDEV(2,IC,Q2,T2,Z,IRUN)
                GT3=0.5*(GT1+GT2)
                GT4=0.5*(GT1-GT2)
                ZT=TOFCALZ(Q1,T1,Q2,T2,IC)
                DZ=100.*(ZT-Z)
                SIG(1,IC)=SIG(1,IC)+GT1**2
                SIG(2,IC)=SIG(2,IC)+GT2**2
                SIG(3,IC)=SIG(3,IC)+GT3**2
                SIG(4,IC)=SIG(4,IC)+GT4**2
                SIG(5,IC)=SIG(5,IC)+DZ**2
                AN=AN+1.
                TSIGMA(1)=TSIGMA(1)+GT3**2
                TSIGMA(2)=TSIGMA(2)+DZ**2
              END IF
            END IF
          END DO
        END IF
      END DO

      DO IC=1,48
        IF(NH2(IC).GE.NENTRYMIN)THEN
          Am=NH2(IC)
          SIG(1,IC)=SQRT(SIG(1,IC)/Am)
          SIG(2,IC)=SQRT(SIG(2,IC)/Am)
```

```
          SIG(3,IC)=SQRT(SIG(3,IC)/Am)
          SIG(4,IC)=SQRT(SIG(4,IC)/Am)
          SIG(5,IC)=SQRT(SIG(5,IC)/Am)
        ELSE
          SIG(1,IC)=1.
          SIG(2,IC)=1.
          SIG(3,IC)=1.
          SIG(4,IC)=1.
          SIG(5,IC)=1.
        END IF
      END DO

      IF(AN.GT.NENTRYMIN)THEN
        TSIGMA(1)=SQRT(TSIGMA(1)/AN)
        TSIGMA(2)=SQRT(TSIGMA(2)/AN)
      ELSE
        TSIGMA(1)=1.
        TSIGMA(2)=10.
      END IF
C
      DO IC=1,48
        REST1=3.*SIG(1,IC)-INT(3.*SIG(1,IC))
        REST2=3.*SIG(2,IC)-INT(3.*SIG(2,IC))
        REST3=3.*SIG(3,IC)-INT(3.*SIG(3,IC))
        REST4=3.*SIG(4,IC)-INT(3.*SIG(4,IC))
        IT1=INT(REST1+0.5)
        IT2=INT(REST2+0.5)
        IT3=INT(REST3+0.5)
        IT4=INT(REST4+0.5)
        IF(IT1.EQ.1)THEN
          XT1=INT(3.*SIG(1,IC))+1.
        ELSE
          XT1=INT(3.*SIG(1,IC))+0.5
        END IF
        IF(IT2.EQ.1)THEN
          XT2=INT(3.*SIG(2,IC))+1.
        ELSE
          XT2=INT(3.*SIG(2,IC))+0.5
        END IF
        IF(IT3.EQ.1)THEN
          XT3=INT(3.*SIG(3,IC))+1.
        ELSE
          XT3=INT(3.*SIG(3,IC))+0.5
        END IF
        IF(IT4.EQ.1)THEN
          XT4=INT(3.*SIG(4,IC))+1.
        ELSE
          XT4=INT(3.*SIG(4,IC))+0.5
        END IF
        NT1=2*INT(XT1/0.05+0.5)
        NT2=2*INT(XT2/0.05+0.5)
        NT3=2*INT(XT3/0.05+0.5)
        NT4=2*INT(XT4/0.05+0.5)
        XTMIN1=-XT1
        XTMAX1=XT1
        XTMIN2=-XT2
        XTMAX2=XT2
        XTMIN3=-XT3
        XTMAX3=XT3
        XTMIN4=-XT4
        XTMAX4=XT4
        CALL HBOOK1(    IC,'W1T1+W2T2   $',NT3,XTMIN3,XTMAX3,0.)
        CALL HBOOK1(100+IC,'EAST END TE $',NT1,XTMIN1,XTMAX1,0.)
        CALL HBOOK1(200+IC,'WEST END TW $',NT2,XTMIN2,XTMAX2,0.)
        CALL HBOOK1(300+IC,'(TE+TW)/2   $',NT3,XTMIN3,XTMAX3,0.)
        CALL HBOOK1(400+IC,'(TE-TW)/2   $',NT4,XTMIN4,XTMAX4,0.)
```

```
      END DO

      REST1=3.*TSIGMA(1)-INT(3.*TSIGMA(1))
      REST2=3.*TSIGMA(2)-INT(3.*TSIGMA(2))
      IT1=INT(REST1+0.5)
      IT2=INT(REST2+0.5)

      If(Ic .ge. 48) IC = 48    ! fixed a bug for the HP-UNIX --  RG

      IF(IT1.EQ.1)THEN
        XT1=INT(3.*SIG(1,IC))+1.
      ELSE
        XT1=INT(3.*SIG(1,IC))+0.5
      END IF

      IF(IT2.EQ.1)THEN
        XT2=INT(3.*SIG(2,IC))+10.
      ELSE
        XT2=INT(3.*SIG(2,IC))+5.0
      END IF

      NT1=2*INT(XT1/0.05+0.5)
      NT2=2*INT(XT2/1.0+0.5)
      XTMIN1=-XT1
      XTMAX1=XT1
      XTMIN2=-XT2
      XTMAX2=XT2
      CALL HBOOK1(501,'TIME WEIGHTED AVERAGE $',NT1,XTMIN1,XTMAX1,0.)
      CALL HBOOK1(502,'(TE+TW)/2 $',NT1,XTMIN1,XTMAX1,0.)
      CALL HBOOK1(503,'(TE-TW)/2 $',NT1,XTMIN1,XTMAX1,0.)
      CALL HBOOK1(504,'(T1+T2)/2 $',NT1,XTMIN1,XTMAX1,0.)
      CALL HBOOK1(505,'(T1-T2)/2 $',NT1,XTMIN1,XTMAX1,0.)
      CALL HBOOK1(506,'ZTOF-ZDC $',NT2,XTMIN2,XTMAX2,0.)

      END
```

```
      SUBROUTINE TOFBP
C
C*********************************************************************
+*
C
C     Calculate barrel countr tof offline constants
C
C                                                         LIU QI 8/26/91
C
*     tofcalb.inc is renamed as caltof.inc to include ETOF calibration
*                                 Guangpei Chen     Sep.1, 1999
C*********************************************************************
C
      INCLUDE 'tofcal.inc'
      INCLUDE 'caltof.inc'
      COMMON/FITA/AI(10,10,2,48),BI1(10,2,48),BI2(10,100,2,48),IEN,ICN
C
      REAL X1(10),X2(10),TA(100)
      INTEGER INFO1
      INTEGER IWA1(10)
      DOUBLE PRECISION TOL1
      DOUBLE PRECISION XPAR1(10)
      DOUBLE PRECISION AI,BI1,BI2
      DOUBLE PRECISION FVEC1(10),WA1(500)
      EXTERNAL TOFFCN1
      DATA TOL1/1.0D-12/

      CALL CLRMEM(OFFSET(1),100)
      CALL CLRMEM(TA(1),100)
      DO IC=1,48
        DO IE=1,2
          DO I=1,10
            BI1(I,IE,IC)=0.D0
            DO J=1,10
              AI(I,J,IE,IC)=0.D0
            END DO
            DO IRUN=1,100
              BI2(I,IRUN,IE,IC)=0.D0
            END DO
          END DO
        END DO
      END DO

C     Loop over countr #
      DO 1000 I=1,MIN(IEV,35000)
C       Run number:
        IRUN=IR(I)
        IF(IRUN.LE.0)GO TO 1000
        DO 2000 IBC=1,2
          IC=ICNT(IBC,I)
          IF(IC.LE.0)GO TO 2000
          IF(NH2(IC).LT.NENTRYMIN)GO TO 2000
C         East end raw charge:
          Q1=TQZ(1,IBC,I)
C         Difference of east end raw time and expected time of filght:
          T1=TQZ(2,IBC,I)
C         West end raw charge:
          Q2=TQZ(3,IBC,I)
C         Difference of west end raw time and expected time of filght:
          T2=TQZ(4,IBC,I)
C         Z-coordinate of hit on the tof counter:
          Z =TQZ(5,IBC,I)
          Z2=Z*Z
          Z3=Z2*Z
          X1(1)=1.0D0
          X1(2)=1.0D0/SQRT(Q1)
```

```
          X1(3)=X1(2)*Z
          X1(4)=Q1
          X1(5)=Q1**2
          X1(6)=Q1**3
          X1(7)=1.D0/(1.157**2+Z2)
          X1(8)=Z
          X1(9)=Z2
          X1(10)=Z3
          X2(1)=1.0D0
          X2(2)=1.0D0/SQRT(Q2)
          X2(3)=-X2(2)*Z
          X2(4)=Q2
          X2(5)=Q2**2
          X2(6)=Q2**3
          X2(7)=1.D0/(1.157**2+Z2)
          X2(8)=-Z
          X2(9)=Z2
          X2(10)=-Z3
C
          TA(IRUN)=TA(IRUN)+0.5*(T1+T2)
          DO J=1,10
            BI1(J,1,IC)=BI1(J,1,IC)+T1*X1(J)
            BI1(J,2,IC)=BI1(J,2,IC)+T2*X2(J)
            BI2(J,IRUN,1,IC)=BI2(J,IRUN,1,IC)+X1(J)
            BI2(J,IRUN,2,IC)=BI2(J,IRUN,2,IC)+X2(J)
            DO K=1,10
              AI(J,K,1,IC)=AI(J,K,1,IC)+X1(J)*X1(K)
              AI(J,K,2,IC)=AI(J,K,2,IC)+X2(J)*X2(K)
            END DO
          END DO
 2000   CONTINUE
 1000 CONTINUE

      DO 4000 IT=1,20
C       To calculate tof constants:
        DO 3000 IC=1,48
        IF(NH2(IC).LT.NENTRYMIN)GO TO 3000
          ICN=IC
          DO IE=1,2
            IEN=IE
C           Initialization
            XPAR1(1)=22.
            XPAR1(2)=50.
            XPAR1(3)=0.
            XPAR1(4)=0.
            XPAR1(5)=0.
            XPAR1(6)=0.D0
            XPAR1(7)=0.D0
            XPAR1(8)=-6.0
            XPAR1(9)=0.
            XPAR1(10)=0.
            IEN=IE
            CALL LMDIF1(TOFFCN1,10,10,XPAR1,FVEC1,TOL1,INFO1,
     $        IWA1,WA1,500)

C           To fill tof offline correction constants array:
            DO I=1,10
              TOFPAR(I,IE,IC)=XPAR1(I)
            END DO
          END DO
 3000   CONTINUE
        IF(IT.EQ.20)GO TO 4000
C       To determine time zero offset of different run:
        DO IRUN=1,NRUN
          AN=0.
          T0=TA(IRUN)
```

```
          DO IC=1,48
            IF(NH2(IC).GE.NENTRYMIN)THEN
              AN=AN+NH1(IC,IRUN)
              DO I=1,10
                T0=T0-0.5*(TOFPAR(I,1,IC)*BI2(I,IRUN,1,IC)+
     #                     TOFPAR(I,2,IC)*BI2(I,IRUN,2,IC))
              END DO
            END IF
          END DO
          IF(AN.GE.2*NENTRYMIN)THEN
            OFFSET(IRUN)=T0/AN
            IF(IRUN.EQ.1)OFF=OFFSET(1)
            OFFSET(IRUN)=OFFSET(IRUN)-OFF
          END IF
        END DO
 4000 CONTINUE

      DO IC=1,48
        TOFVEL(IC)=0.162
        TOFVEL(IC+48)=0.162
        IF(NH2(IC).LT.NENTRYMIN)THEN
          TOFPAR(1,2,IC)=999.
          TOFPAR(1,2,IC)=999.
          DO I=2,10
            TOFPAR(I,1,IC)=0.
            TOFPAR(I,2,IC)=0.
          END DO
        END IF
      END DO

      END
```

```
      SUBROUTINE TOFWTP
C
C*********************************************************************
C
C     This routine calculates time weighted average parameters
C
C                                    LIU QI 8/26/91
C
*     tofcalb.inc is renamed as caltof.inc to include ETOF calibration
*                            Guangpei Chen     Sep.1, 1999
C*********************************************************************
C
      INCLUDE 'tofcal.inc'
      INCLUDE 'caltof.inc'
      COMMON /TOFWT/AT(4,4,48),BT(4,48),NC
C
      INTEGER INFO
      INTEGER IWA(4)
      DOUBLE PRECISION TOL
      DOUBLE PRECISION X(4)
      DOUBLE PRECISION AT,BT
      DOUBLE PRECISION XPAR(4),FVEC(4),WA(100)
      EXTERNAL TOFFCN3
      DATA TOL/1.0D-12/
C
      DO IC=1,48
        DO I=1,4
          BT(I,IC)=0.D0
          DO J=1,4
            AT(I,J,IC)=0.D0
          END DO
        END DO
      END DO
C
      DO I=1,MIN(IEV,35000)
C       Run number:
        IRUN=IR(I)
        IF(IRUN.GT.0)THEN
          DO IBC=1,2
            IC=ICNT(IBC,I)
            IF(IC.GT.0.AND.IC.LE.48)THEN
              IF(NH2(IC).GE.NENTRYMIN)THEN
C               East end raw charge:
                Q1=TQZ(1,IBC,I)
C               Difference of east end raw time and expected time of fi
     +lght:
                T1=TQZ(2,IBC,I)
C               West end raw charge:
                Q2=TQZ(3,IBC,I)
C               Difference of west end raw time and expected time of fi
     +lght:
                T2=TQZ(4,IBC,I)
C               Z-coordinate of hit on the tof counter:
                Z =TQZ(5,IBC,I)
C               East end random time err determined with corrected tof
     +consts:
                GT1=TOFDEV(1,IC,Q1,T1,Z,IRUN)
C               West end random time err determined with corrected tof
     +consts:
                GT2=TOFDEV(2,IC,Q2,T2,Z,IRUN)
                X(1)=GT1-GT2
                X(2)=X(1)*Z
                X(3)=X(2)*Z
                X(4)=1.D0
                DO M=1,4
                  BT(M,IC)=BT(M,IC)+GT2*X(M)
```

```
                  DO N=1,4
                    AT(M,N,IC)=AT(M,N,IC)+X(M)*X(N)
                  END DO
                END DO
              END IF
            END IF
          END DO
        END IF
      END DO
      DO IC=1,48
        NC=IC
        IF(NH2(IC).GE.NENTRYMIN)THEN
          XPAR(1)=0.5D0
          XPAR(2)=0.0D0
          XPAR(3)=0.0D0
          XPAR(4)=0.0D0
          CALL LMDIF1(TOFFCN3,4,4,XPAR,FVEC,TOL,INFO,IWA,WA,100)
          TOFWP(1,IC)=XPAR(1)
          TOFWP(2,IC)=XPAR(2)
          TOFWP(3,IC)=XPAR(3)
          TOFWP(4,IC)=XPAR(4)
        ELSE
          TOFWP(1,IC)=0.5
          TOFWP(2,IC)=0.
          TOFWP(3,IC)=0.
          TOFWP(4,IC)=0.
        END IF
      END DO
      RETURN
      END
```

```
      SUBROUTINE TOFSIG
C
C*******************************************************************
C
C     This routine calculates time deviations
C                                              LIU QI 8/26/91
C
*     tofcalb.inc is renamed as caltof.inc to include ETOF calibration
*                               Guangpei Chen     Sep.1, 1999
C*******************************************************************
+*
C
      INCLUDE 'tofcal.inc'
      INCLUDE 'caltof.inc'
C
      DATA SAVE/999./,TOFSD/192*0/
C
      DO I=1,MIN(IEV,35000)
C       Run number:
        IRUN=IR(I)
        DO IBC=1,2
          IF(IBC.EQ.1)SAVE=999.
          IF(ICNT(IBC,I).GT.0)THEN
C           Cunter number:
            IC=ICNT(IBC,I)
            IF(NH2(IC).GE.NENTRYMIN)THEN
C             East end raw charge:
              Q1=TQZ(1,IBC,I)
C             Difference of east end raw time and expected time of filg
+ht:
              T1=TQZ(2,IBC,I)
C             West end raw charge:
              Q2=TQZ(3,IBC,I)
C             Difference of west end raw time and expected time of filg
+ht:
              T2=TQZ(4,IBC,I)
C             Z-coordinate of hit on the tof counter:
              Z =TQZ(5,IBC,I)
C             East end random time err determined with corrected tof co
+nsts:
              GT1=TOFDEV(1,IC,Q1,T1,Z,IRUN)
C             West end random time err determined with corrected tof co
+nsts:
              GT2=TOFDEV(2,IC,Q2,T2,Z,IRUN)
              WE=TOFWP(1,IC)+TOFWP(2,IC)*Z+TOFWP(3,IC)*Z**2
              WW=1.-WE
              TERR=WE*GT1+WW*GT2+TOFWP(4,IC)
              ZT=TOFCALZ(Q1,T1,Q2,T2,IC)
              DZ=100.*(ZT-Z)
              GT3=0.5*(GT1+GT2) !   moved here
              GT4=0.5*(GT1-GT2) !   moved here
              CALL HFILL(501,TERR,0.0,1.)
              CALL HFILL(100+IC,GT1,0.0,1.)
              CALL HFILL(200+IC,GT2,0.0,1.)
              CALL HFILL(300+IC,GT3,0.0,1.)
              CALL HFILL(400+IC,GT4,0.0,1.)
              CALL HFILL(IC,TERR,0.0,1.)
C             xieyh          GT3=0.5*(GT1+GT2)
C             xieyh          GT4=0.5*(GT1-GT2)
              IF(IBC.EQ.2.AND.SAVE.LT.900.0)THEN
                GT5=0.5*(GT3+SAVE)
                GT6=0.5*(GT3-SAVE)
                CALL HFILL(502,GT3,0.0,1.)
                CALL HFILL(502,SAVE,0.0,1.)
                CALL HFILL(503,GT4,0.0,1.)
                CALL HFILL(503,SGT4,0.0,1.)
```

```
                  CALL HFILL(504,GT5,0.0,1.)
                  CALL HFILL(505,GT6,0.0,1.)
                ELSE
                  SAVE=GT3
                  SGT4=GT4
                END IF
                CALL HFILL(506,DZ,0.0,1.)
              END IF
            END IF
          END DO
        END DO

        DO IC=1,48
          IF(NH2(IC).GE.NENTRYMIN)THEN
            DO KN=1,5
              ID=100*KN+IC
              IF(KN.EQ.5)ID=IC
              CALL HFITGA(ID,CP,AV,SD,CH2,112,DEV)
              SIG(KN,IC)=SD
              IF(KN.LE.2)TOFSD(2,KN,IC)=SD**2
            END DO
          ELSE
            SIG(1,IC)=9.999
            SIG(2,IC)=9.999
            SIG(3,IC)=9.999
            SIG(4,IC)=9.999
            SIG(5,IC)=9.999
            TOFSD(2,1,IC)=9.9992**2
            TOFSD(2,2,IC)=9.9992**2
          END IF
        END DO
        DO KN=1,6
          ID=500+KN
          CALL HFITGA(ID,CP,AV,SD,CH2,112,DEV)
          TSIGMA(KN)=SD
        END DO
C       Deviation of time caused by beam length:
        TSIGMA(7)=TSIGMA(4)**2-TSIGMA(5)**2
        TSIGMA(8)=TSIGMA(7)+TSIGMA(3)**2-TSIGMA(2)**2
        IF(TSIGMA(7).GT.0.)THEN
          TSIGMA(7)=SQRT(TSIGMA(7))
        ELSE
          TSIGMA(7)=0.
        END IF
        IF(TSIGMA(8).GT.0.)THEN
          TSIGMA(8)=SQRT(TSIGMA(8))
        ELSE
          TSIGMA(8)=0.
        END IF

        END
```