Subject: Re: NOAA Satellite Active Archive IDL Imagery Display Tool
Posted by Kelly Dean on Wed, 28 Apr 1999 07:00:00 GMT
View Forum Message <> Reply to Message

I put this routine together to read the NOAA-15 Level 1B 10-bit files from
SAA. I used it to help create this image:

http://www.cira.colostate.edu/special/csuidl/N15HRPT3A.htm

Warning - very computer intensive. Something I hope to improve on later.

If anyone makes any improvements, be should to let me know.

Kelly Dean
CSU/CIRA

```
======================ReadKLM1B.PRO======================
;+
; NAME:
FUNCTION ReadKLM1B, unit, ARSstr, HEADstr, LinePre, LineSuf, Image
;
; PURPOSE:
;      Read NOAA's 10-bit LAC/HRPT images
;
; Information about unpacking LAC/HRPT data can be found at:
;
;  http://www2.ncdc.noaa.gov:80/docs/podug/
;  http://www2.ncdc.noaa.gov:80/docs/klm/
;
; CALLING SEQUENCE:
;
;   OpenR, unit, filename, /GET_LUN, /SWAP_ENDIAN ! Assuming read on PC.
;
;   IF ( ReadKLM1B(unit, ARSstr, HEADstr, LinePre, LineSuf, L1BImage ) )
THEN BEGIN
;     PRINT, ' Read >', filename,'...'
;   ENDIF ELSE BEGIN
;     PRINT, ' Trouble with reading -', filename, '...'
;   ENDELSE
;
;   Close, unit
;   Free_Lun, unit
;
; INPUTS:
;       unit - file unit number
;
; OUTPUTS:
;       ARSstr - Structure - Archive Retrieval System Record
```

```
;       HEADstr - Structure - Image header information
;       LinePre - Structure - Line prefix information
;       LineSuf - Structure - Line suffix information
;       L1Bimage - imager - 3-D array with 10-bit image data from 5 channels
( channel, width, length )
;
;
; SIDE EFFECTS:
;       Works with 16-bit Level 1B files from the Satellite Active Archive
with imagery
;    from NOAA-15 ( NOAA-KLM series ) only.
;
;
; MODIFICATION HISTORY:
;
;    Developed by Kelly Dean - CIRA/Colorado State University
(dean@cira.colostate.edu)
;-
;
;
HEADstr = { KLMLevel1B,         $ ; Data Set Header record format for
NOAA-KLM AVHRR
          SiteID:'   ',       $ ; Data Set Creation Site ID
          Blank0:' ',         $ ; Blank
          VerNum:0,           $ ; Level 1B Format Version Number
          VerYear:0,          $ ; Level 1B Format Version Year ( ex: 1999
)
          VerJday:0,          $ ; Level 1B Format Version Day of Year (
ex: 077 )
          Reserved0:0,        $ ; Reserved
          Reserved1:0,        $ ; Reserved
          NumHeadRec:0,       $ ; Number of Header Records
   Fill0:INTarr(3),     $ ; Zero filled
   dataset:STRING(' ',FORMAT='(A42)'), $ ; 42 character dataset name
          ProcID:'       ',   $ ; Processing Block ID
          SatID:0,            $ ; Satellite ID ( NOAA-15: 4 )
          InstrID:0,          $ ; Instrument ID
          DataType:0,         $ ; Data Type Code
          TIPSrc:0,           $ ; TIP Source Code
          StartDay:0L,        $ ; Start of Data Set Day ( since 1 Jan
1950 )
          StartYear:0,        $ ; Start of Data Set Year ( ex: 1999 )
          StartJDay:0,        $ ; Start of Data Set Julian Day ( ex: 079
)
          StartTime:0L,       $ ; Start of Data Set Time UTC (
millisecond )
          EndDay:0L,          $ ; End of Data Set Day ( since 1 Jan 1950
)
```

```
        EndYear:0,          $ ; End of Data Set Year ( ex: 1999 )
        EndJDay:0,          $ ; End of Data Set Julian Day ( ex: 079 )
        EndTime:0L,         $ ; End of Data Set Time UTC ( millisecond
)
        CYear:0,           $ ; Year of Last CPIDS Update ( ex: 1999 )
        CJDay:0,            $ ; Day of Year of Last CPIDS Update ( ex:
079)
  Fill1:INTarr(4),     $ ; Zero filled
  InstrStatus1:0L,     $ ; Instrument Status ( in bits )
  Fill2:0,            $ ; Zero filled
  StatusChange:0,      $ ; Record Number of Status Change
  InstrStatus2:0L,     $ ; Second Instrument Status
  NumRec:0,            $ ; Count of Data Recods in this Data Set
  NumAnc:0,            $ ; Count of Calib and Earth Loc Scan lines
  NumMiss:0,           $ ; Count of missing scan lines
  NumGaps:0,           $ ; Count of Data Gaps in this Data Set
  NumFrames:0,         $ ; Count of Data Frames w/o Frame Sync Word Errors
  TIPErr:0,           $ ; Count of PACS Detected TIP Parity Error
  AuxErr:0,           $ ; Sum of All Auxiliary Sync Errors Detected in the
Input Data
  TimeErr:0,          $ ; Time Sequence Error
  TimeErrCode:0,       $ ; Time Sequence Error Code ( in bits )
  ClockUpdate:0,       $ ; SOCC Clock Update Indicator
  EarthLocErr:0,       $ ; Earth Location Error Indicator
  EarthLocErrCode:0,   $ ; Earth Location Error Indicator Code ( in bits )
  PACSstatus:0,        $ ; PACS Status Bit Field ( in bits )
  PACSSource:0,        $ ; PACS Data Source ( 1:Glimore, 2:Wallops, 3:SOCC
)
  Fill3:0L,           $ ; Zero Fill
  Reserved2:'         ', $ ; Reserved
  Reserved3:'         ', $ ; Reserved
  Fill4:INTarr(5),     $ ; Zero filled
  RampField:0,         $ ; Ramp/Auto Calibration Indicators Bit Field ( in
bits )
  SolarChanYear:0,     $ ; Year of Most Recent Solar Channel Calibration (
ex: 1999)
  SolarChanJDay:0,     $ ; Day of Year of Most Recent Solar Channel
Calibration ( ex: 079)
  AlgorID1:0,          $ ; Primary Calibration Algorithm
  AlgorIDOpt1:0,       $ ; Primary Calibration Algorithm Options ( in bits
)
  AlgorID2:0,          $ ; Second Calibration Algorithm
  AlgorIDOpt2:0,       $ ; Second Calibration Algorithm Options ( in bits )

  IR1coef1:0,          $ ; IR1 Target Temperature Conversion Coefficient 1
( * 10^2)
  IR1coef2:0,          $ ; IR1 Target Temperature Conversion Coefficient 2
( * 10^5 )
```

```
   IR1coef3:0,        $ ; IR1 Target Temperature Conversion Coefficient 3
( * 10^8 )
   IR1coef4:0,        $ ; IR1 Target Temperature Conversion Coefficient 4
( * 10^8 )
   IR1coef5:0,        $ ; IR1 Target Temperature Conversion Coefficient 5
( * 10^8 )
   IR1coef6:0,        $ ; IR1 Target Temperature Conversion Coefficient 6
( * 10^8 )
   IR2coef1:0,        $ ; IR2 Target Temperature Conversion Coefficient 1
( * 10^2)
   IR2coef2:0,        $ ; IR2 Target Temperature Conversion Coefficient 2
( * 10^5 )
   IR2coef3:0,        $ ; IR2 Target Temperature Conversion Coefficient 3
( * 10^8 )
   IR2coef4:0,        $ ; IR2 Target Temperature Conversion Coefficient 4
( * 10^8 )
   IR2coef5:0,        $ ; IR2 Target Temperature Conversion Coefficient 5
( * 10^8 )
   IR2coef6:0,        $ ; IR2 Target Temperature Conversion Coefficient 6
( * 10^8 )
   IR3coef1:0,        $ ; IR3 Target Temperature Conversion Coefficient 1
( * 10^2)
   IR3coef2:0,        $ ; IR3 Target Temperature Conversion Coefficient 2
( * 10^5 )
   IR3coef3:0,        $ ; IR3 Target Temperature Conversion Coefficient 3
( * 10^8 )
   IR3coef4:0,        $ ; IR3 Target Temperature Conversion Coefficient 4
( * 10^8 )
   IR3coef5:0,        $ ; IR3 Target Temperature Conversion Coefficient 5
( * 10^8 )
   IR3coef6:0,        $ ; IR3 Target Temperature Conversion Coefficient 6
( * 10^8 )
   IR4coef1:0,        $ ; IR4 Target Temperature Conversion Coefficient 1
( * 10^2)
   IR4coef2:0,        $ ; IR4 Target Temperature Conversion Coefficient 2
( * 10^5 )
   IR4coef3:0,        $ ; IR4 Target Temperature Conversion Coefficient 3
( * 10^8 )
   IR4coef4:0,        $ ; IR4 Target Temperature Conversion Coefficient 4
( * 10^8 )
   IR4coef5:0,        $ ; IR4 Target Temperature Conversion Coefficient 5
( * 10^8 )
   IR4coef6:0,        $ ; IR4 Target Temperature Conversion Coefficient 6
( * 10^8 )
   Fill5:Lonarr(2),   $ ; Fill
   Ch1Irrad:0L,        $ ; Channel 1 Solar Filter Irradiance in Wavelength
( * 10^1 )
   Ch1Width:0L,        $ ; Channel 1 Equivalent Filter Width in Wavelength
```

```
( * 10^3 )
   Ch2Irrad:0L,        $ ; Channel 2 Solar Filter Irradiance in Wavelength
( * 10^1 )
   Ch2Width:0L,        $ ; Channel 2 Equivalent Filter Width in Wavelength
( * 10^3 )
   Ch3Irrad:0L,        $ ; Channel 3a Solar Filter Irradiance in Wavelength
( * 10^1 )
   Ch3Width:0L,        $ ; Channel 3a Equivalent Filter Width in Wavelength
( * 10^3 )
   Ch3WavNum:0L,       $ ; Channel 3b Central Wavenumber ( * 10^2 )
         Ch3Const1:0L,       $ ; Channel 3b Constant 1 ( * 10^5 )
         Ch3Const2:0L,       $ ; Channel 3b Constant 2 ( * 10^6 )
   Ch4WavNum:0L,       $ ; Channel 4 Central Wavenumber ( * 10^2 )
         Ch4Const1:0L,       $ ; Channel 4 Constant 1 ( * 10^5 )
         Ch4Const2:0L,       $ ; Channel 4 Constant 2 ( * 10^6 )
   Ch5WavNum:0L,       $ ; Channel 5 Central Wavenumber ( * 10^2 )
         Ch5Const1:0L,       $ ; Channel 5 Constant 1 ( * 10^5 )
         Ch5Const2:0L,       $ ; Channel 5 Constant 2 ( * 10^6 )
   Fill6:LONarr(3),    $ ; Fill
   Ellipsoid:'       ', $ ; Reference Ellipsoid Model ID
   EarthNadir:0,       $ ; Nadir Earth Location Tolerance in Km ( * 10^1 )
   EarthLoc:0,         $ ; Earth Location Bit Field ( in bits )
   Fill7:0,            $ ; Fill
   RollAttErr:0,       $ ; Constant Roll Attitude Error in Degrees ( * 10^3
)
   PitchAttErr:0,      $ ; Constant Pitch Attitude Error in Degrees ( *
10^3 )
   YawAttErr:0,        $ ; Constant Yaw Attitude Error in Degree ( * 10^3 )

   EpochYear:0,        $ ; Epoch Year for Orbit Vector
   EpochJDay:0,        $ ; Epoch Day of Year for Orbit Vector
   EpochTime:0L,       $ ; Epoch UTC time of day in milliseconds for Orbit
Vector
   SemiAxis:0L,        $ ; Semi-Major Axis (km) ( * 10^5 )
   Eccen:0L,           $ ; Eccentricity  ( * 10^8 )
   Incl:0L,            $ ; Inclination ( degree ) ( * 10^5 )
   ArgPer:0L,          $ ; Argument of Perigee ( degree ) ( *10^5 )
   AscNode:0L,         $ ; Right Ascension of the Ascending Node ( degree )
(  * 10^5 )
   Mean:0L,            $ ; Mean Anomaly ( degree ) ( * 10^5 )
   PosVecX:0L,         $ ; Position Vector X Component (km) ( * 10^5 )
   PosVecY:0L,         $ ; Position Vector Y Component (km) ( * 10^5 )
   PosVecZ:0L,         $ ; Position Vector Z Component (km) ( * 10^5 )
   VecXdot:0L,         $ ; Velocity X-dot Component in (km/sec) ( * 10^8 )
   VecYdot:0L,         $ ; Velocity Y-dot Component in (km/sec) ( * 10^8 )
   VecZdot:0L,         $ ; Velocity Z-dot Component in (km/sec) ( * 10^8 )
   EarthSunDist:0L,    $ ; Earth/Sun Distance ratio ( * 10^6 )
   Fill8:LONarr(4),    $ ; Fill
```

```
        PatchTempCoef:INTarr(6), $ ; Patch Temperature Conversion
Coefficient
        PatchTempExtCoef:INTarr(6), $ ; Patch Temperature Extended
Conversion Coefficient
        PatchPowCoef:INTarr(6),  $ ; Patch Power Conversion Coefficient
        RadTempCoef:INTarr(6),   $ ; Radiator Temperature Conversion
Coefficient
        BBTempCoef1:INTarr(6),   $ ; Blackbody Temperature 1 Conversion
Coefficient
        BBTempCoef2:INTarr(6),   $ ; Blackbody Temperature 2 Conversion
Coefficient
        BBTempCoef3:INTarr(6),   $ ; Blackbody Temperature 3 Conversion
Coefficient
        BBTempCoef4:INTarr(6),   $ ; Blackbody Temperature 4 Conversion
Coefficient
        ElectCurrCoef:INTarr(6), $ ; Electronics Current Conversion
Coefficient
        MotorCurrCoef:INTarr(6), $ ; Motor Current Conversion
Coefficient
        EarthShieldCoef:INTarr(6), $ ; Earth Shield Position Conversion
Coefficient
        ElecTempCoef:INTarr(6),  $ ; Electronics Temperature Conversion
Coefficient
        CoolHouseCoef:INTarr(6), $ ; Cooler Housing Temperature
Conversion Coefiicient
        BaseTempCoef:INTarr(6),  $ ; Baseplate Temperature Conversion
Coefficient
        MotorHouseCoef:INTarr(6), $ ; Motor HousingTemperature
Conversion Coef
        ADTempCoef:INTarr(6),    $ ; A/D Converter Temperature
Conversion Coef
        Det4VCoef:INTarr(6),     $ ; Detector #4 Bias Voltage Conversion
Coefficient
        Det5VCoef:INTarr(6),     $ ; Detector #5 Bias Voltage Conversion
Coefficient
        BB3bView:INTarr(6),      $ ; Channel 3b Blackbody View
Conversion Coefficient
        BB4View:INTarr(6),       $ ; Channel 4 Blackbody View Conversion
Coefficient
        BB5View:INTarr(6),       $ ; Channel 5 Blackbody View Conversion
Coefficient
        RefVCoef:INTarr(6)       } ; Reference Voltage Conversion
Coefficient

ARSstr = { ArchiveRetrievalSystem,          $ ; Archive Retrieval System
Record
        Cost:'     ',               $ ; Cost Number
        SAA:'      ',               $ ; SAA Number
```

```
        CYear:'    ',                $ ; Order Creation Year ( ex:
1999 )
        CDOY:'   ',                  $ ; Order Creation Day of Year
( ex: 072 )
        PSite:' ',                   $ ; Processing Site Code (
A:SAA, S: NCDC/Suitland, N: NCDC/Asheville )
        PSoft:'       ',             $ ; Processing Software ID
        Name:STRING(' ',FORMAT='(A42)'),  $ ; Date Set Name
        Blank0:'  ',                 $ ; Blank
        SFlag:' ',                   $ ; Select Flag ( T: Total S:
Selective )
        BLat:'   ',                  $ ; Beginning Latitude
        ELat:'   ',                  $ ; Ending Latitude
        BLon:'    ',                 $ ; Beginning Longitude
        ELon:'    ',                 $ ; Ending Longitude
        SHour:'  ',                  $ ; Starting Hour (UTC)
        SMinute:'  ',                $ ; Starting Minute
        Duration:'   ',              $ ; Number of Minutes
        AppFlag:' ',                 $ ; Appended Data
        Channel:STRING(' ',FORMAT='(A20)'), $ ; Channel Select Flag
        Blank1:STRING(' ',FORMAT='(A29)'),  $ ; Blank
        ADFlag:' ',                  $ ; Ascend/Descend Flag ( A:
Ascending, D:Descending, B:Both )
        BegLat:'   ',                $ ; Begining Latitude
        EndLat:'   ',                $ ; Ending Latitude
        BegLon:'    ',               $ ; Begining Longitude
        EndLon:'    ',               $ ; Ending Longitude
        Format:STRING(' ',FORMAT='(A20)'),$ ; Data Format
        RecSize:'      ',            $ ; Size of record (octets)
        NumRec:'      ',             $ ; Number of Records,
including ARS and Data Set Header Records
        Padded:BYTarr(319)           } ; Padding
;
;  Create header structure of data record
;
Info = { ScanLineInfo,   $ ; Scan Line Information
       Num:0,          $ ; Scan Line Number
       Year:0,         $ ; Scan Line Year ( ex: 1999 )
       JDay:0,         $ ; Scan Line Day of Year ( ex: 079 )
       ClockDelta:0,   $ ; Satellite CLock Drift Delta ( milliseconds )
       Time:0L,        $ ; Scan Line UTC Time of Day in ( millisecond )
       Bit:0,          $ ; Scan Line bit field ( Channel 3 select bit 1 )
       Fill0:INTarr(5) } ; Zero Fill

indicators = { QualityIndicators , $ ; Quality Indicators
            BitField:0L,       $ ; Bit Field ( ( in bits )
            Scan:0L,           $ ; Scan Line Quality Flags ( in bit )
            Calib:INTarr(3),    $ ; Calibration Quality Flags
```

```
            BitErr:0,          $ ; Count of Bit Error in Frame Sync
            Fill1:LONarr(2)    } ; Zero Fill

VSCal = { VisibleCalibration, $ ; Visible Calibration
        Slope1:0L,         $ ; Slope 1 ( * 10^7 )
        Intercept1:0L,     $ ; Intercept 1 ( * 10^6 )
        Slope2:0L,         $ ; Slope 2 ( * 10^7 )
        Intercept2:0L,     $ ; Intercept 2 ( * 10^6 )
        Intersection:0L    } ; Intersection

IRCal = { InfraredCalibration, $ ; Infrared Calibration
        Coef1:0L,          $ ; Coefficient 1 ( * 10^6 )
        Coef2:0L,          $ ; Coefficient 2 ( * 10^6 )
        Coef3:0L           } ; Coefficient 3 ( * 10^6 )

Calib = { CalibrationCoefficient, $ ; Calibration Coefficients
        OperCh1:VSCal,        $ ; Visible Operational Calibration
        TestCh1:VSCal,        $ ; Visible Test Calibation
        PrelCh1:VSCal,        $ ; Visible Prelaunch Calibration
        OperCh2:VSCal,        $ ; Visible Operational Calibration
        TestCh2:VSCal,        $ ; Visible Test Calibation
        PrelCh2:VSCal,        $ ; Visible Prelaunch Calibration
        OperCh3a:VSCal,       $ ; Visible Operational Calibration
        TestCh3a:VSCal,       $ ; Visible Test Calibation
        PrelCh3a:VSCal,       $ ; Visible Prelaunch Calibration
        OperCh3b:IRCal,       $ ; Infrared Operational Calibration
        TestCh3b:IRCal,       $ ; Infrared Test Calibation
        OperCh4:IRCal,        $ ; Infrared Operational Calibration
        TestCh4:IRCal,        $ ; Infrared Test Calibation
        OperCh5:IRCal,        $ ; Infrared Operational Calibration
        TestCh5:IRCal,        $ ; Infrared Test Calibation
        Fill3:LONarr(3)       } ; Zero Fill


LatLon = REPLICATE({EarthLoc, latitude:0L,longitude:0L},51)

Angle  = REPLICATE({ Angular, $  ; Angular Relationship
                Solar:0, $  ; Solar Zenith Angle ( * 10^2 )
                Sat:0,   $  ; Satellite Zenith Angle ( * 10^2 )
                Az:0   },$  ; Relative Azimuth Angle ( * 10^2)
                51      )  ; Replicate this structure 51 times

nav = { Navigation,         $ ; Navigation
      StatusBit:0L,       $ ; NStatus Bit Field ( in Bit )
      TimeTIPAng:0L,      $ ; Time Associated with TIP Euler Angle (
seconds )
      TIPAngle:INTarr(3),  $ ; TIP Euler Angles, Roll, Pitch Yaw ( degree
) ( * 10^3 )
```

```
        AltRef:0,           $ ; Spacecraft Altitude above Reference
Ellipsoid (km) ( * 10^1)
        Angle:Angle,        $ ; Angular Relationships - Structure
        Fill4:INTarr(3),    $ ; Zero Fill
        EarthLoc:LatLon,    $ ; Earth Location - Structure
        Fill5:LONarr(2)     } ; Zero Fill
;
Telemetry = { Telemetry,    $ ; HRPT Minor Frame Telemetry
              FrameSync:INTarr(6),  $ ; Frame Sync
              ID:INTarr(2),         $ ; ID ( in bit )
              TimeCode:INTarr(4),   $ ; Time Code ( in bit )
              Telemetry:INTarr(10), $ ; Telemetry ( in bit )
              BackScan:INTarr(30),  $ ; Back Scan
              SpaceData:INTarr(50), $ ; Space Data
              SyncDelta:0,          $ ; Sync Delta ( in bit )
              Fill6:0             } ; Zero Fill
;
LACpre = { LACprefix,           $ ; Format of LAC/HRPT data record NOAA-KLM

           Scan:info,           $ ; Scan Line Information - structure
           Quality:Indicators,  $ ; Quality Indicators - structure
           Calib:Calib,         $ ; Calibration Coefficients - structure
           Nav:Nav,             $ ; Navigation - structure
           Telemetry:Telemetry  } ; HRPT Minor Frame Telemetry
;
;  Create tail structure of data record
;
LACsuf = { LACsuffix,           $ ;
           Fill1:LONarr(2),     $ ; Zero fill
           BadBit1:0,           $ ; Invalid Word Bit Flags ( in bits )
           BData:0,             $ ; AVHRR Digital B Data
           Fill2:LONarr(3),     $ ; Zero fill
           BadBit2:0L,          $ ; Invalid Word Bit Flags ( in bits )
           InstrTemp:BYTarr(22), $ ; Instrument Temperatures
           Fill3:INTarr(3),     $ ; Zero fill
           CLAVR:0L,            $ ; Clouds from AVHRR
           Reserved:0L,         $ ; Reserved
           CCM:INTarr(256),     $ ; Clear/Cloudy/Mixed
           Fill4:LONarr(94)     } ; Zero fill
;
; Read Record 1
; Archive Retrieval System Record
;   Users can verify which selection criteria were used by checking the ARS
Header record.
;
ReadU, unit, ARSstr
;
; Read Record 2
```

```
;  The first record contains the Data Set Header. The values need to be
swapped. The data files are generated
;  on an IBM.
;
ReadU, unit, HEADstr
;
;Determine the number of channels
;
IF ( ARSstr.channel EQ 'YYYYYYYYYYYYYYYYYYYY' ) THEN BEGIN
  nchan = 5
ENDIF ELSE BEGIN
  PRINT, ' What do I do here >', ARSstr.channel
ENDELSE
;
;  Determine number of elements ( always 2048 ) and lines
;
nx = 2048
ny = LONG(HEADstr.NumRec)
;
;  Create structure to be used to read the file's data section
;
  DataRecords =
 REPLICATE({lnhead:LACpre,video:LONarr(3414),lntail:LACsuf},n y)
;
; Skip the the first three records and jump to starting point of data
;
POINT_LUN, unit, LONG(ARSstr.RecSize) + 512L
NumRec = LONG(ARSstr.NumRec)
ReadU, unit, DataRecords
;
; Reverse last dimension of line documentation
;
LinePre   = DataRecords((ny-1)-INDGEN(ny)).LnHead
LineSuf   = DataRecords((ny-1)-INDGEN(ny)).LnTail
;
; Unpack 10-bit values
;
dim1   = 3414
pack   = INTarr(3*dim1*ny)
index1 = LINDGEN(dim1-1) * 3
index2 = LINDGEN(dim1-1) * 3 + 1
index3 = LINDGEN(dim1) * 3 + 2
Print, ' Unpacking 10 bit image data...'
ii = 0
FOR ii = 0, NY - 1 DO BEGIN
  video = REFORM(DataRecords(ii).video, dim1)
  pack(index3) = FIX(video AND '3FF'X)
  pack(index2) = FIX(ISHFT(video(0:3412),-10) AND '3FF'X)
```

```
  pack(index1) = FIX(ISHFT(video(0:3412),-20) AND '3FF'X)
  index1 = index1 + ( 3 * dim1 ) - 2
  index2 = index2 + ( 3 * dim1 ) - 2
  index3 = index3 + ( 3 * dim1 ) - 2
ENDFOR
Print, ' Trim array to new size...',nchan * nx * ny
trim      = LONG(nchan * nx * ny) - 1L
TrimPack  = pack(0:trim)
image     = REFORM(TrimPack,nchan,nx,ny)
;
;
; Return with status.
;
status = 1

RETURN, status
END ; ReadKLM1B
```
