PAUL BUEHLER

Decoding of the GIOVEB/SREM raw data

Document No	:	ESTEC contract No. 19336/05/NL/Ja/pg, WP 20
Issue	:	1.0
Date	:	January 10, 2007

Dr. Paul Bühler
Auhofstrasse 22/3
1130 Vienna
Austria

Contents

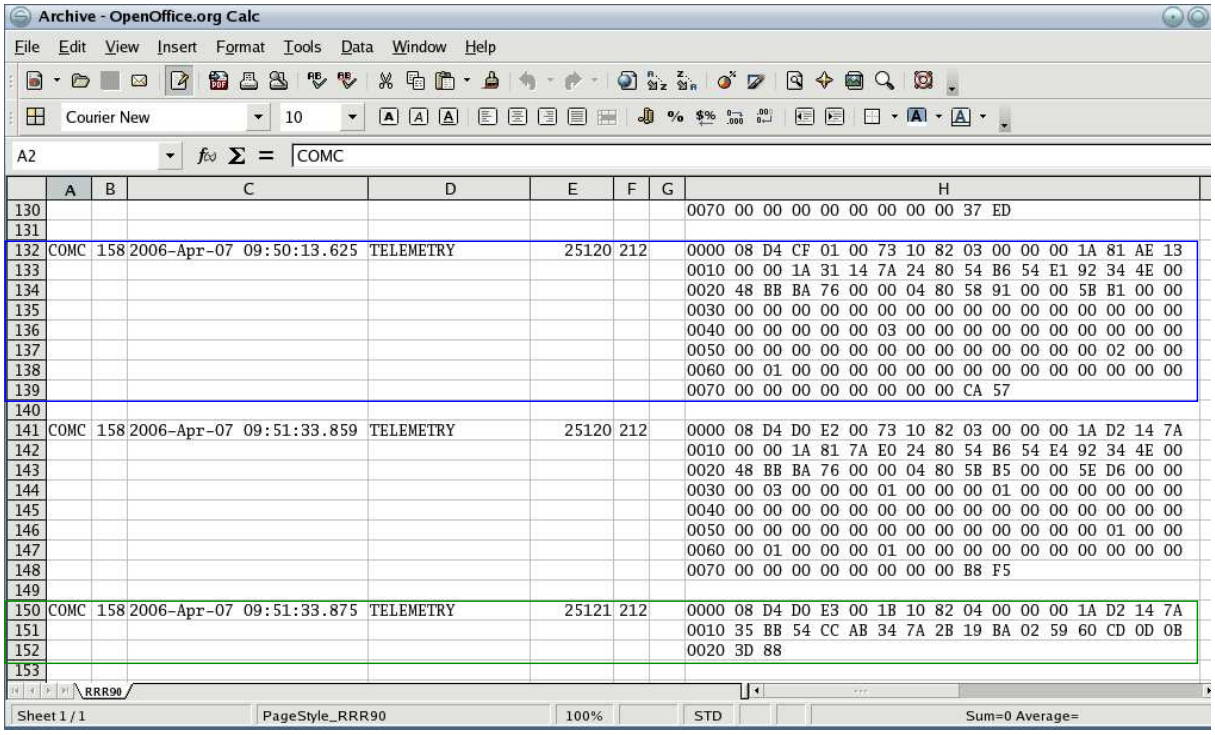
1 GIOVEB/SREM test data format

The test data from the GIOVEB/SREM was provided by Giuseppe Mandorlo (Giuseppe.Mandorlo@esa.int). The raw data is contained in a EXCEL xls-file. In addition an ASCII file was provided describing the test sequence and detailing the used telecommands.

According to the information contained in the ASCII file the test was carried out on April 7, 2006 and contained a sequence of 15 accumulations and one total dose reading. The nominal accumulation time was set to 80 *sec*.

Figure 1 shows an excerpt of the xls-file. The table consists of 8 columns (A - H). The data is sorted into data blocks. A block represents the data of an issued telecommand or data from the telemetry. The first line of each block contains header information (columns A - G). The raw data is contained in column H and can span several rows.

The important information contained in the header line is the absolute time information in column C and the number in column E, which indicates the type of data block. Of special interest are data block types 25120 (SREM science data block) and 25121 (total dose reading). Examples of such data blocks are framed in figure 1 by a blue (science data) and a green (total dose) box.



	A	B	C	D	E	F	G	H
130								0070 00 00 00 00 00 00 00 00 37 ED
131								
132	COMC	158	2006-Apr-07 09:50:13.625	TELEMETRY	25120	212		0000 08 D4 CF 01 00 73 10 82 03 00 00 00 1A 81 AE 13
133								0010 00 00 1A 31 14 7A 24 80 54 B6 54 E1 92 34 4E 00
134								0020 48 BB BA 76 00 00 04 80 58 91 00 00 5B B1 00 00
135								0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
136								0040 00 00 00 00 00 03 00 00 00 00 00 00 00 00 00 00
137								0050 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00
138								0060 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
139								0070 00 00 00 00 00 00 00 00 CA 57
140								
141	COMC	158	2006-Apr-07 09:51:33.859	TELEMETRY	25120	212		0000 08 D4 D0 E2 00 73 10 82 03 00 00 00 1A D2 14 7A
142								0010 00 00 1A 81 7A E0 24 80 54 B6 54 E4 92 34 4E 00
143								0020 48 BB BA 76 00 00 04 80 5B B5 00 00 5E D6 00 00
144								0030 00 03 00 00 00 01 00 00 00 01 00 00 00 00 00 00
145								0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
146								0050 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00
147								0060 00 01 00 00 00 01 00 00 00 00 00 00 00 00 00 00
148								0070 00 00 00 00 00 00 00 00 B8 F5
149								
150	COMC	158	2006-Apr-07 09:51:33.875	TELEMETRY	25121	212		0000 08 D4 D0 E3 00 1B 10 82 04 00 00 00 1A D2 14 7A
151								0010 35 BB 54 CC AB 34 7A 2B 19 BA 02 59 60 CD 0D 0B
152								0020 3D 88
153								

Figure 1: Excerpt of the GIOVEB/SREM raw data Excel file.

Each line in the raw data column contains a four character long line counter and 32 characters of hexadecimal data (16 byte). In case of the science data the raw data block has 122 byte (excluding line counters) and the total dose has 34 byte.

Table 1 shows the encoding of the SREM data items in the raw science data block and raw total dose data block.

Table 1: Encoding of the SREM data in the GIOVEB/SREM raw science and total dose data blocks

science data			total dose data		
Item	start byte	number of bytes	Item	start byte	number of bytes
startSCETtime	11	6	radval	17	2
stopSCETtime	17	6	radtemp	19	2
hkstat	23	2	Vref1	21	2
TD1D2	25	2	Vref2	23	2
TD3	27	2	Vref3	25	2
m6V	29	2	Vref4	27	2
p6V	31	2	Cref1	29	2
p5V	33	2	Cref2	31	2
p150V	35	2			
fid	37	2			
status	39	2			
startLT	41	4			
stopLT	45	4			
tc1	49	4			
s12	53	4			
s13	57	4			
s14	61	4			
s15	65	4			
tc2	69	4			
s25	73	4			
c1	77	4			
c2	81	4			
c3	85	4			
c4	89	4			
tc3	93	4			
s32	97	4			
s33	101	4			
s34	105	4			
pl1	109	4			
pl2	113	4			
pl3	117	4			

2 Data decoding

In order to process the GIOVEB/SREM raw data with PV-WAVE under Linux, the xls-file is converted into an ASCII file using the command

```
>> xls2csv -q 3 -c '/' filename.xls > filename.csv
```

The program xls2csv can be downloaded from http://www.45.free.net/ vitus/software/catdoc/.

In the converted ASCII file the excerpt of the xml-file shown in figure 1 looks then as shown in figure 2.

This ASCII file can be easily searched for telemetry data packets of type 25120 (SREM science data block) and 25121 (total does reading) and the belonging raw data blocks can be extracted and the SREM data items decoded according to table 1.

```
 00 00 00 00 00 00 37 ED"
-Apr-07 09:50:13.625"/"TELEMETRY"/"25120"/"212"//"0000 08 D4 CF 01 00 73 10 82 03 00 00 00 1A 81 AE 13"
 1A 31 14 7A 24 80 54 B6 54 E1 92 34 4E 00"
 BA 76 00 00 04 80 58 91 00 00 5B B1 00 00"
 00 00 00 00 00 00 00 00 00 00 00 00 00 00"
 00 00 00 03 00 00 00 00 00 00 00 00 00 00"
 00 00 00 00 00 00 00 00 00 00 00 02 00 00"
 00 00 00 00 00 00 00 00 00 00 00 00 00 00"
 00 00 00 00 00 00 CA 57"
-Apr-07 09:51:33.859"/"TELEMETRY"/"25120"/"212"//"0000 08 D4 D0 E2 00 73 10 82 03 00 00 00 1A D2 14 7A"
 1A 81 7A E0 24 80 54 B6 54 E4 92 34 4E 00"
 BA 76 00 00 04 80 5B B5 00 00 5E D6 00 00"
 00 00 00 01 00 00 00 01 00 00 00 00 00 00"
 00 00 00 00 00 00 00 00 00 00 00 00 00 00"
 00 00 00 00 00 00 00 00 00 00 00 01 00 00"
 00 00 00 01 00 00 00 00 00 00 00 00 00 00"
 00 00 00 00 00 00 B8 F5"
-Apr-07 09:51:33.875"/"TELEMETRY"/"25121"/"212"//"0000 08 D4 D0 E3 00 1B 10 82 04 00 00 00 1A D2 14 7A"
```

Figure 2: Same excerpt as shon in figure 1 but for the converted raw data file.

2.1 Test results

The left panel in figure 3 shows the average count rate histogram of the 15 data accumulations contained in the test data file and the right panel shows the extracted accumulation times for all 15 accumulations. Although the statistics is very low, the histogram compares reasonably well with the response to the cosmics background on ground measured earlier e.g with the SREM now flying on Rosetta (see figure 9 in *P. Bühler*, report on *Rosetta/SREM TM data* [2004]). The extracted accumulation times agree with the nominal value of 80 seconds.

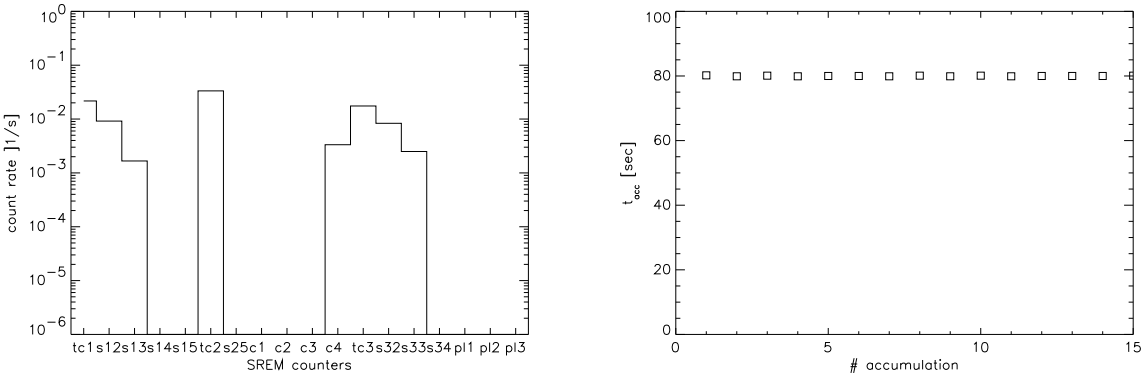


Figure 3: Data extracted from the test data file. The left panel shows the average count rate histogram of the 15 accumulations and the righ panel shows the extracted accumlation time for all accumulations.

3 gioveb_rawdatadecoder.pro

The PV-WAVE program *gioveb_rawdatadecoder.pro* is used to decode the GIOVEB/SREM raw data files. The program takes as input the name of a to-ASCII-converted raw data file and gives as output a structure containing the extracted SREM science and total dose data.

```
; ------------------------------------------------------------------------------
pro gioveb_rawdatadecoder, fname,gbdstructs
;
;
;
; ..............................................................................
; initialisations
pattern='/'
ss=['"COMC"','"TELEMETRY",  $
  '"25120"','"25121"','"25122"','"25123"']
line=''
datablock=''

; data buffer
buffsize=1000L
mk_gioveb_rawdatastruct, gbdstruct
gbdstructs=replicate(gbdstruct,buffsize)

; open data file
openr,fid,fname,/get_lun
if (fid le 0) then return

; loop over data file entries
db=0      ; data block flag
nl=0      ; number of lines in data block
ndb=0L    ; number of data blocks

while (not eof(fid)) do begin

  ; read next line
  readf,fid,line

  ; split line into its constituents
  tokens=strsplit(line,pattern)

  ; check for string 'COMC'
  newblock=strpos(line,'COMC') gt 0
  if (newblock) then begin

    ; update data block buffer
    if (db gt 0) then begin

      if (ndb mod buffsize eq 0) then begin
        print,ndb
        gbdstructs=[gbdstructs,replicate(gbdstruct,buffsize)]
      endif

      ; data type independend items
      gbdstructs(ndb).datatype=db
```

```
  gbdstructs(ndb).timestamp=dtstamp
  gbdstructs(ndb).pcktdatafield.startSCETtime=  $
    hex2dec(strmid(datablock,20,12))/2.^16.

  ; finalize last data block
  if (db eq 1) then begin
    if (nl ne 8) then stop

    ; SCE stop time
    gbdstructs(ndb).pcktdatafield.stopSCETtime=  $
      hex2dec(strmid(datablock,32,12))/2.^16.

    ; HK info
    gbdstructs(ndb).pcktdatafield.sciedata.hkstat=  $
      dec2bin(hex2dec(strmid(datablock,44,4)),16)
    for ii=0,6 do  $
      gbdstructs(ndb).pcktdatafield.sciedata.(ii+1)=  $
        hex2dec(strmid(datablock,48+ii*4,4))
    gbdstructs(ndb).pcktdatafield.sciedata.status=  $
      dec2bin(hex2dec(strmid(datablock,76,4)),16)

    ; start/stop LT
    gbdstructs(ndb).pcktdatafield.sciedata.startLT=  $
      hex2dec(strmid(datablock,84,4)+strmid(datablock,80,4))
    gbdstructs(ndb).pcktdatafield.sciedata.stopLT=  $
      hex2dec(strmid(datablock,92,4)+strmid(datablock,88,4))

    ; counters
    for ii=0,17 do  $
      gbdstructs(ndb).pcktdatafield.sciedata.(ii+13)=  $
        hex2dec(strmid(datablock,100+8*ii,4)+strmid(datablock,96+8*ii,4))

  endif else if (db eq 2) then begin
    if (nl ne 3) then stop

    ; total dose readings
    for ii=0,7 do  $
      gbdstructs(ndb).pcktdatafield.radfetdata.(ii)=  $
        hex2dec(strmid(datablock,32+4*ii,4))

  endif else if (db eq 3) then begin
    if (nl ne 17) then stop
  endif else if (db eq 4) then begin
    if (nl ne 4) then stop
  endif

  ; increment ndb
  ndb=ndb+1

endif

; reset data block flag and line counter
db=0
nl=0

; analyze new line
; check for string 'TELEMETRY'
```

```
    telem=strpos(line,'TELEMETRY') gt 0
    if (telem) then begin
      if (tokens(4) eq ss(2)) then db=1  $
      else if (tokens(4) eq ss(3)) then db=2  $
      else if (tokens(4) eq ss(4)) then db=3  $
      else if (tokens(4) eq ss(5)) then db=4

      ; extract time stamp
      qq=strsplit(tokens(2),'"')
      qq=strsplit(qq(1),' ')
      dstr=strmid(qq(0),9,2)+strmid(qq(0),4,5)+strmid(qq(0),0,4)
      tstr=qq(1)
      dtstamp=str_to_dt(dstr,tstr,date_fmt=4,time_fmt=-1)

      ; extract first data block line
      qq=strcompress(strsplit(tokens(7),'"'),/remove_all)
      datablock=strmid(qq(1),4,1000L)
      nl=nl+1

    endif

  endif else begin

    ; a new entry in data block?
    if (db gt 0) then begin
      qq=strcompress(strsplit(tokens(7),'"'),/remove_all)
      datablock=datablock+strmid(qq(1),4,1000L)

      nl=nl+1
    endif

  endelse

endwhile

close,fid
free_lun,fid

; remove empty elements in gbdstructs
sel=where(gbdstructs.datatype gt 0,nsel)
if (nsel gt 0) then  $
  gbdstructs=gbdstructs(sel)  $
else  $
  gbdstructs=0

return
end

; ------------------------------------------------------------------------------
pro mk_gioveb_rawdatastruct, gbdstruct
;
;
;
; ..............................................................................
sciedata={sciedata,  $
  hkstat: '',   $
  TD1D2: 0L,  $
```

```
  TD3: 0L,  $
  m6V: 0L,  $
  p6V: 0L,  $
  p5V: 0L,  $
  p150V: 0L,  $
  fid: 0,  $
  status: '',  $
  startLT: 0.0D0,  $
  stopLT: 0.0D0,  $
  startUT: today(),  $
  stopUT: today(),  $
  tc1: 0.0D0,  $
  s12: 0.0D0,  $
  s13: 0.0D0,  $
  s14: 0.0D0,  $
  s15: 0.0D0,  $
  tc2: 0.0D0,  $
  s25: 0.0D0,  $
  c1: 0.0D0,  $
  c2: 0.0D0,  $
  c3: 0.0D0,  $
  c4: 0.0D0,  $
  tc3: 0.0D0,  $
  s32: 0.0D0,  $
  s33: 0.0D0,  $
  s34: 0.0D0,  $
  pl1: 0.0D0,  $
  pl2: 0.0D0,  $
  pl3: 0.0D0}
radfetdata={radfetdata,  $
  radval: 0L,  $
  radtemp: 0L,  $
  Vref1: 0L,  $
  Vref2: 0L,  $
  Vref3: 0L,  $
  Vref4: 0L,  $
  Cref1: 0L,  $
  Cref2: 0L}
pcktdatafield={pcktdatafield,  $
  startSCETtime: 0.0D0,  $
  stopSCETtime: 0.0D0,  $
  sciedata: sciedata,  $
  radfetdata: radfetdata}

gbdstruct={gbdstruct,   $
  datatype:  0L,  $
  timestamp:  today(),  $
  pcktdatafield: pcktdatafield}

return
end

; ---------------------------------------------------------------------------
```