MCE File Formats

Revision History:

Rev	*Date*	*Description of change*
1.0	2006-04-03	MA/Initial Release txt file
2.0	2006-06-07	BB/Updated header information
3.0	2007-01-24	MA/Added Header, mcestatus block, and future formats
3.1	2007-02-04	Added data mode descriptions
3.2	2007-08-15	Added data_mode 6, sample files, doc number.
3.3	2007-08-28	Updated example status header file when using mcestatus
3.4	2007-09-18	BB: Updating section 2.2.1 with data packet header revisions Added logical data formatting for 4 RC cards in section 2.2.2 Added appendix A/B for legacy data.
3.5	2009-10-19	Added more detail on the internally populated status block. Added data mode 8
3.6	2009-10-19	Added the data frame timing error bit to the data frame status work in the status block.

1. Introduction

The data acquired from Multi-Channel Electronics (MCE) are saved on the Data-Acquisition PC (DA) as text or binary. The structure of data files is described in this document. Depending on the version of software running on data acquisition, the file structure may change. Depending on the version of the MCE firmware, the structure of the Data Packet Header may also change. Refer to [1] for more detailed and up-to-date information:

2. Data Structure

Data can be stored as text, text-2, or binary. The data files contain multiple frames of data, and each frame of data consists of a 43-word header followed by a data block and a checksum. Each data file is also accompanied by an auxiliary file referred to as *Status* file or .run file.

2.1 Auxiliary Status File (aka *.run* file)

The auxiliary file contains a <header> section with information about the status and different parameter settings in MCE and is generated by inquiring MCE for those parameters right before asking for data frames. This <header> is configurable and therefore when parsing data one should look for end-of-header marker </header> instead of relying on a fixed-size header. In general, the status header is configured to capture settings that do not change during a single data-acquisition of multiple data frames. Note that this auxiliary file may contain more sections and is best described in:

http://actexperiment.info/roundtable/pmwiki.php/ActWiki/RunFileFormat

The <header> section of this file is generated by a utility called `mcestatus` and is in pseudo-xml format with all values in decimal. Here is an example:

```
<HEADER>
<RB sys row_len> 00000064 00000064 00000064 00000064 00000064 00000064 00000064 00000064 00000064
<RB psc psc_status> 09659443 838860829 572653568 226823293 196348884 208536236 57476398 156501559 00000804
<RB cc ret_dat_s> 00000001 00000100
<RB cc use_dv> 00000000
….
<RB rc3 data_mode> 00000000
……
</HEADER>

<RC> 1 2 3 4
<DAS_VERSION> 011220070826
```

2.2 Data Frame

Each data frame contains of a 43-word header followed by a data block and then a 32-bit checksum. The Data Packet Header is described in detail in the document called monitoring_mce_status.doc. Refer to [1] for more detailed information.

2.2.1 Data Packet Header (Version 6)

The Data Packet Header contains 43 integer (32-bit) words. The contents of the header depend on the version of Clock Card (CC) firmware. The Clock Card firmware version number is stored in the auxiliary '.run' file associated with the data run in question. In the '.run' file, the decimal number following the entry "<RB cc fw_rev>" has to be converted to hexadecimal format to interpret the Clock Card firmware version as:

0xRRrrBBBB

Where:

- RR is the major revision number
- rr is the minor revision number
- BBBB is the build number

The most recent Data Packet Header version is listed below. Previous versions are listed in **Appendix B**. The header below is implemented in the following Clock Card firmware version, and later:

- "67108866" (.run file entry following '<RB cc fw_rev>'); "cc_ v04000002" (CVS tag) and greater

Data Packet Header

Word #	Byte 3	Byte 2	Byte 1	Byte 0
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
0	Data Frame Status Bits			
1	Clock Card Data Frame Counter			
2	Address Dwell Time (row_len)			
3	Number of Rows Reported (num_rows-reported)			
4	Data Period (data_rate)			
5	Clock Card Address-Return-to-Zero-Counter			
6	Header Version #			
7	Ramp Value			
8	Ramp Card Address (ramp_card_addr)		Ramp Parameter ID (ramp_param_id)	
9	Number of Rows Servoed (num_rows_servoed)			
10	Sync Box Number			
11	Run ID # (run_id)			
12	User Writable (user_word)			
13	Errno			
14	FPGA Temperature, AC			
15	FPGA Temperature, BC1			
16	FPGA Temperature, BC2			
17	FPGA Temperature, BC3			
18	FPGA Temperature, RC1			
19	FPGA Temperature, RC2			
20	FPGA Temperature, RC3			
21	FPGA Temperature, RC4			
22	FPGA Temperature, CC			
23	Errno			
24	Card Temperature, AC			
25	Card Temperature, BC1			
26	Card Temperature, BC2			
27	Card Temperature, BC3			
28	Card Temperature, RC1			
29	Card Temperature, RC2			
30	Card Temperature, RC3			
31	Card Temperature, RC4			
32	Card Temperature, CC			
33	Errno			
34	Software Version, PSUC	Fan 1 Tachometer, PSUC	Fan 2 Tachometer, PSUC	Temperature 1, PSUC
35	Temperature 2, PSUC	Temperature 3, PSUC	ADC Offset, PSUC	
36	Supply Voltage 1, PSUC		Supply Voltage 2, PSUC	
37	Supply Voltage 3, PSUC		Supply Voltage 4, PSUC	
38	Supply Voltage 5, PSUC		Supply Current 1, PSUC	
39	Supply Current 2, PSUC		Supply Current 3, PSUC	
40	Supply Current 4, PSUC		Supply Current 5, PSUC	
41	Errno			
42	Box Temperature			

Table 1. Data Packet Header Format (Version 6)

Bit Number	Data Frame Status Bit Description
31	--
30	--
029	--
28	--
27	--
26	--
25	--
24	--
23	--
22	--
21	--
20	Data Timing Error (cc_v05000001+)
19	Number of Columns Reported [19:16] (cc_v0400000a+)
18	
17	
16	
15	Reserved
14	Reserved
13	Readout Card 4 responding (cc_v04000002+)
12	Readout Card 3 responding (cc_v04000002+)
11	Readout Card 2 responding (cc_v04000002+)
10	Readout Card 1 responding (cc_v04000002+)
9	Reserved
8	Reserved
7	Reserved
6	Reserved
5	TES Bias Square Wave Level (reported **ONLY** in cc_v03000004)
4	Active Clock (cc_v03000004 and cc_v04000000+)
3	Sync Box Error (cc_v04000002+)
2	Sync Box Free Run Mode (cc_v03000004, and cc_v04000000+)
1	Stop (reported in all Clock Card firmware versions)
0	Last Frame (reported in all Clock Card firmware versions)

Table 2. Data Frame Status Bit Description

Bit Number	Bit Description
31	Stale data
30	Internal reset has occurred
29	Card not present, AC
28	Backplane communications error, AC
27	Wishbone execution error, AC
26	Card not present, BC1
25	Backplane communications error, BC1
24	Wishbone execution error, BC1
23	Card not present, BC2
22	Backplane communications error, BC2
21	Wishbone execution error, BC2
20	Card not present, BC3
19	Backplane communications error, BC3
18	Wishbone execution error, BC3
17	Card not present, RC1
16	Backplane communications error, RC1
15	Wishbone execution error, RC1
14	Card not present, RC2
13	Backplane communications error, RC2
12	Wishbone execution error, RC2
11	Card not present, RC3
10	Backplane communications error, RC3
9	Wishbone execution error, RC3
8	Card not present, RC4
7	Backplane communications error, RC4
6	Wishbone execution error, RC4
5	Card not present, CC
4	Backplane communications error, CC
3	Wishbone execution error, CC
2	Card not present, PSUC
1	Backplane communications error, PSUC
0	Wishbone execution error, PSUC

Table 3. Errno Word Bits

Internally Populated Status Words (words 13 through 42):
The internal status words are populated with data when internal commands are enabled. This feature is available in firmware version "50331652" (cc_v03000004), and after firmware version "67108864" (cc_v04000000).

For information on what the meaning of each field in the header is, see the document titled monitoring_mce_status.doc.

2.2.2 Data Block

Each MCE can acquire data from up-to 41x32 array of pixels. There are up-to 4 readout cards in each MCE and each readout card serves 8 columns of the array, i.e. RC1 reports colums 0 to 7. The size of the data block can vary depending on number of rows reported and number of readout cards that are queried. Hence, the data block has N signed 32-bit integer entries where:

$N = n_{row} \times n_{RC} \times 8$

- n_{row}: In the auxiliary file, the number of rows reported can be found out from <RB CC num_rows_to_read xx>.

  NOTE: For CC firmware revisions prior to 4.0.1, the number of rows reported and number of rows servoed were the same and therefore in the auxiliary file, you should check for <RB CC num_rows xx>

- n_{RC}: In the auxiliary file, the number of RCs queried can be found out from <RC 1 2 3 4>. This is either set to one or all.

Using default settings, the data-block size for a single readout-card is 41×8=328 words. Each 32-bit data item may contain feedback or error data or both depending on the *data_mode* parameter settings. The auxiliary file includes the *data_mode, servo_mode*, and other settings that correspond to each data block (Refer to [3] "MCE Command Description", SC2_ELE_S580_515]).

The *data_mode* can be set to return SQ1 feedback, error signal, low-pass filtered data, or *raw* 50MHz feedback data for each pixel, number of flux jumps, or a combination. Different data modes are described in Table 4.

For all *data_mode* settings except raw mode, the data block is organized such that a row is reported through all columns before moving to the next row. i.e.:

row0col0, row0col1 …. row0col31, row1col0row1col31, …..row40col31

(Please see Appendix A for legacy data acquired prior to DAS version: 011220070826, DAS version can be found in the auxiliary file associated with the data file).

Data_mode	**Pixel-data Description**
0	Signed 32b Co-added Error signal: *num_samp**(ADC sample – *adc_offset*)
1	32b SQ1 Feedback data: when *servo_mode* =3, this is the PID calculation result which has 12 more bits of accuracy than the actual 14b SQ1 feedback applied.
2	Low-pass-filtered SQ1 feedback data
3	Raw data, 50MHz ADC samples in signed 8b format where the 8b are the most significant bits of the 14b ADC sample.
4	Signed 18b SQ1 feedback* + signed 14b coadded error signal
5	Signed 24b SQ1 feedback* + signed 8b num_flux_jumps
6	Signed 18b filtered data** + signed 14b coadded error signal
7	Signed 22b filtered data** + signed 10b coadded error signal
8	Signed 24b filtered data*** + signed 8b num_flux_jumps

Table 4. data_mode settings

* Feedback data reported in this mode is scaled by 2^{12} compared to data_mode=1.
** Filtered data in this mode is scaled by 2^{10} compared to data_mode 2 (to normalize filter gain of 1216)

*** Filtered data in this mode is divided-by-2 compared to data_mode 2

When *data_mode* is set to 3 or *raw* mode, then the data block has to be interpreted differently. In raw mode, the 8 most-significant bits of consecutive ADC samples are stored in a homogenous 8192×8b memory block per column. This memory is filled by consecutive 50MHz samples of the first row (pixel) followed by the next row (pixel) and so on for 2 consecutive frames and then shipped out as 41×32 data blocks. Number of samples are determined by *row_len* parameter reported in the auxiliary file.

For simplicity, consider *row_len* =128 and *number of rows reported*= 41 then 128 frames of data has to be acquired to capture all rows:

r row, c column, s sample

43-word header
r0c0s0 r0c1s0..... r0c7s0
r0c0s1 r0c1s0....

...
r0c0s40 ...
checksum
43B header
r0c0s41 ...
....
checksum
43B header
r0c0s82....
....
checksum
43B header
r0c0s123....
...
r0c0s127....
r1c0s0....
.....
and so on.

2.2.3 Checksum

The last line of the frame data is a 32-bit checksum that is calculated as simple xor checksum of the overall frame data. Refer to [2] “SCUBA-2 Data Acquisition Software Overview – Part two Protocols”, SC2/SOF/S200/014, *UK Astronomy* for details of checksum calculation.

2.3 Binary, Text , Text-2 Format

In Text format, all entries are space delimited and there is a newline character after the 43-word header, after every row of data, after the checksum.

In Text-2 format, there is a newline character after every entry.

When binary format is chosen, the data structure is similar to the above with no newline and space characters.

2.4 Choosing a File Format

When using DAS, use a script called set_data to set the format to TEXT or BINARY.

3. Sample Files

3.1 Sample Text File

```
       0         1       128        41        48     24787         1         0         0         0         0         0
0        0        0
   0        0        0        0        0        0        0        0        0        0        0        0        0
0        0
 0        0        0        0        0        0        0        0        0        0        0        0        0
     108 24608841      234 -9912279 21807141       71      388    15542
     166  9682454      246 -25707162   409317      118      310    10345
     123  2146602      314 11108984  5685603       58      385     4535
   16031  4799873      220 43842638 16645542    16040        1      912
      81 -4604331      101  -785718 11911113      182      198    10586
     103 -22150773      138  -311433  -180170       80      124    15608
     383 -388150685      542 2107121626 1391352814      359      353     4562
     178 -293565775      441 2096340932 1893555741      162      330     5275
      55 -360641823      161 2061836157 1863851503        9       93     4522
     135 -370390437      368 2050727885 1855463062      181      449     4834
     120 -1261736      281 -1031809  1769475      157      312    15730
     126 -364639740      241 2077613860 1374198777      226      497     4397
     198 -375436691      559 2054184834 1823088201      107      169    15834
     511 -370750911      613 2046713606 1836703149      425      443     2831
     281 -359921109      403 2043469692 1857101349      125      577    11581
     166 -362739170      360 2040717194 1840815673      112      312    12179
     901 -358839776      878 2038505276 1836850771      789      955     5982
     132 -363050434      130 2036293466 1838521926      212      290     1410
     243 -363902275      332 2035589050 1842667048      189      592     5203
     140 -384873692      324 2030526269 1848041004      123      315     5964
     135 -383874383      449 2030804761 1841241499      170      303     4880
     116 -385398138      587 2028560221 1837473382      138      351     5226
     203  5390070      503 13354107 11337309       76      273    15658
     246  8388763      285 -44908153 -1982297      219      640    14264
     163 -55345249      408  -671098 32784943      204      180    12355
     124 -342455735       87 2101780446 1370266545    16367      279     4111
     239 -352925143      472 2068275120 1858969098      246      395     4969
     255 -367195814      391 2051022643 1855282697      137      152     4702
     169 23101588      373 -14057642 1861410299      178      162     8765
     172 -308196804      487 2091016109 1837604378      156      144     3594
   15786 42056680    15999  3979868 2045664570    15840    16095    15003
     166 17875434      330 25805008 -1663068569       69      243     9082
     238 -5864941       77 -1179342 53493478      145      208    14853
   16213 -54280165       31 13564869 36306570    16125       68    13530
     130 -50905101      415 -41010157 27410418       73      190    11523
     157 -328578592      478 2122178408 1379294143      173      306     3821
     189 -259356197      226 2104041382 1885773373      135      305     4914
     202 39846473      139 24641353 12534017      156      239    15069
     173 -362837276      178 2085609388 1366318018       45      397     4432
     165 -359101749      313 2062196647 1853283907      104      312     4711
   15874 -465123381    16197 1966644965 1845763782    15967    16174     4382
1390701744
```

3.2 Sample Text-2 File

3.3 Sample Binary File

Data_mode	Auxiliary file	Data file	plot
Raw (data_mode =3)	txt	bin	ps

Appendix A: Legacy Data:

Prior to DAS_VERSION 011220070826, in all *data_mode* settings except raw mode, the data block is organized such that a row is reported for all RC1 columns (column 0 to 7) before moving to next row for RC1; when last row for RC1 columns are reported, then row 0 for all RC2 columns (8 to 15) are reported and so on. For example, in Text mode, each line has 8 entries and the data block looks like:

row 0 - column 0 to column 7 space-delimited (RC1)
row 1 - column 0 to column 7 space-delimited (RC1)
....
row 40 - column 0 to column 7 space-delimited (RC1)
row 0 - column 8 to column 15 space-delimited (RC2)
.....
row 40 - column 8 to column 15 space-delimited (RC2)
row 0 - column 16 to column 23 space-delimited (RC3)
.....
....
row 40 - column 24 to column 31 space-delimited (RC4)

Appendix B: Legacy Data Packet Header Versions

Data Packet Header Version 0

Implemented in the following Clock Card firmware versions:

- “33554449” (.run file entry following ‘<RB cc fw_rev>’); “cc_v02000011” (CVS tag)
- “50331648” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v03000000” (CVS tag)

Data Packet Header Content:

Header Word Number	Word Description
0	Frame status bits
1	Row length
2	Number of rows multiplexed and read out
3	Data period (aka ‘data rate’)
4	Sync number (generated by clock card)
5	Frame sequence number (frame counter)
6	Active clock
7	Sync Box error
8	Sync Box free run mode
9	Sync Box data-valid number
10..39	Internally Populated Status Block
40	--
41	--
42	--

Data Packet Header Version 1

Implemented in the following Clock Card firmware versions:

- “33554450” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v02000012” (CVS tag)
- “33554451” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v02000013” (CVS tag)
- “50331649” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v03000001” (CVS tag)

Data Packet Header Content:

Header Word Number	Word Description
0	Frame status bits
1	Row length
2	Number of rows multiplexed and read out
3	Data period (aka ‘data rate’)
4	Sync number (generated by clock card)
5	Frame sequence number (frame counter)
6	Active clock
7	Sync Box error
8	Sync Box free run mode
9	Sync Box data-valid number
10	TES bias level
11..40	Internally Populated Status Block
41	--
42	--

Data Packet Header Version 2

Implemented in the following Clock Card firmware versions:

- “50331650” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v03000002” (CVS tag)
- “50331651” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v03000003” (CVS tag)
- “50331653” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v03000005” (CVS tag)
- “50331654” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v03000006” (CVS tag)
- “50331655” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v03000007” (CVS tag)
- “50331656” (.run file entry following ‘<RB cc fw_rev>’); “cc_ v03000008” (CVS tag)

Data Packet Header Content:

Header Word Number	Word Description
0	Frame status bits
1	Frame sequence number (frame counter)
2	Row length
3	Number of rows multiplexed and read out
4	Data period (aka 'data rate')
5	Sync number (generated by clock card)
6	Frame sequence number (frame counter)
7	Active clock
8	Sync Box error
9	Sync Box free run mode
10	Sync Box data-valid number
11	TES bias level
12..41	Internally Populated Status Block
42	--

Data Packet Header Version 3

Implemented in the following Clock Card firmware versions:

- "50331652" (.run file entry following '<RB cc fw_rev>'); "cc_ v03000004" (CVS tag)

Data Packet Header Content:

Header Word Number	Word Description
0	Frame status bits
1	Frame sequence number (frame counter)
2	Sync number (generated by clock card)
3	Sync Box data-valid number
4..33	Internally Populated Status Block
34	--
35	--
36	--
37	--
38	--
39	--
40	--
41	--
42	--

Data Packet Header Version 4

Implemented in the following Clock Card firmware versions:

- "67108864" (.run file entry following '<RB cc fw_rev>'); "cc_ v04000000" (CVS tag)

Data Packet Header Content:

Header Word Number	Word Description
0	Frame status bits
1	Frame sequence number (frame counter)
2	Sync number (generated by clock card)
3	Sync Box data-valid number
4..37	Internally Populated Status Block
38	Card address
39	Ramp value
40	Row length
41	Number of rows multiplexed and read out
42	Data period (aka 'data rate')

Data Packet Header Version 5

Implemented in the following Clock Card firmware versions:

- "67108865" (.run file entry following '<RB cc fw_rev>'); "cc_ v04000001" (CVS tag)

Data Packet Header Content:

Header Word Number	Word Description
0	Frame status bits
1	Frame sequence number (frame counter)
2	Row length
3	Number of rows multiplexed
4	Data period (aka 'data rate')
5	Sync number (generated by clock card)
6	Card address
7	Ramp value
8	Number of rows read
9	--
10	Sync Box data-valid number
11..40	Internally Populated Status Block
41	--
42	--

Table of Contents

4. Glossary

5. References:

[1] Fibre Protocol between RTL PC and MCE (xls)

[2] "SCUBA-2 Data Acquisition Software Overview – Part two Protocols", SC2/SOF/S200/014, *UK Astronomy*

[3] "MCE Command Description", SC2_ELE_S580_515