

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
 HAYSTACK OBSERVATORY
 WESTFORD, MASSACHUSETTS 01886

30 June 2008

Telephone: 781-981-5407
 Fax: 781-981-0590

To: Mark 5 Development Group

From: Dan L. Smythe

Subject: Mark 5B File Utility Programs

We have some handy utility programs that read `.m5b` files written by the `dimino disk2file` command. These programs can be downloaded from <ftp://web.haystack.mit.edu/pub/mark5/B/util/>.

A summary of the function of these programs is

`bstate` – Analyze state counts in each channel in the file
`bpcal` – Calculate amplitude and phase of a phase cal tone in each channel in the file
`time_check` – Read start and stop times in the file, requires `header_reader`
`time_check.py` – A faster python version of `time_check`
`vlbi2` – Calculate cross- or autocorrelation spectra

With no arguments, each of these programs returns its usage syntax.

The `bstate` program is used to analyze the state counts in a Mark 5B file. The `bstate` program accepts a Mark 5B file name and the number of Mark 5B disk frames to analyze. The `bstate` program can only analyze Mark 5B files recorded with a bitmask of `0xFFFFFFFF`. For more information on Mark 5B bitmasks and recording modes see the *Mark 5B Command Set* documentation at www.haystack.edu/tech/vlbi/mark5/software.html. The `bstate` program reports the number of samples and percentage of time each channel is in each state. It also reports a correction factor to apply to the channel gain to produce ideal state counts. The usage of `bstate` is

```
bstate <input m5b file name> <number of disk frames>
```

The `bpcal` program is used to analyze the phase cal tones in a Mark 5B file. The `bpcal` program accepts a Mark 5B file name, the tone frequency, and optionally the number of Mark 5B disk frames to analyze. The `bpcal` program can only analyze Mark 5B files recorded with a bitmask of `0xFFFFFFFF`. The `bpcal` program reports the amplitude and phase of the specified phase cal tone in each channel in the file. The usage of `bpcal` is

```
bpcal <input m5b fname> <tone freq (KHz)> [<# frames>]
```

Where `[<# frames>]` is optional and defaults to 200.

The `time_check` programs are used to check the start and end times in a Mark 5B file. The `time_check` programs accept a Mark 5B file name. The `time_check` programs report the time code in the first and last Mark 5B headers found in the file. The usage of `time_check` is

```
time_check <input m5b file name>
```

To use the `time_check.py` python version of the program, you must have python installed on your system. To install python on a Debian (or Ubuntu) system, enter, as root

```
aptitude install python
```

The `vlbi2` program is used to correlate two Mark 5B files with the same starting record time and record mode. A postscript file `ddl.pos` is produced showing the correlation amplitude and phase as a function of frequency in all channels in the two files with all channels plotted on the same axes. Correlation coefficients (fringe amplitudes, which will be 1.00 for an autocorrelation) are indicated below each channel. The `vlbi2` program accepts two files, each 10 MB in size, and two flags. The usage is:

```
vlbi2 <file1> <file2> [-2bit <2bit_arg>] [-rev <rev_arg>]
```

Where the items in `[]` are optional. The `-2bit` flag indicates whether the files are 2 bits/sample recordings (`2bit_arg` is 1) or 1 bit/sample (`2bit_arg` is 0) [default]. The files must be the same number of bits per sample. The `-rev` flag determines whether the channels in `ddl.pos` are plotted in reverse order (`rev_arg` is 1) or normal order (`rev_arg` is 0) [default]. It is important to note that with the `-rev` flag, though the channels can be plotted in the reverse order, they are not flipped in frequency. If `file1` and `file2` are the same, an autocorrelation is produced; if they are different a cross correlation is produced.

To display the results of the correlation, enter

```
gs ddl.pos in an X terminal window.
```

If `Ctrl-Alt-F7` doesn't give you an X terminal window, try `startx &`. If `gs` is not installed on your system, and you have a Debian (or Ubuntu) operating system, you can enter, as root

```
aptitude install gs
```

to install `gs`. If you are unable to view the `ddl.pos` output file on your system, you can copy it to another system that has linux ghostscript or Windows gsview installed. If you copy it to a Windows system, I suggest changing the name to `ddl.ps`.

Examples

```
time_check 1bit.m5b
Please be patient, I'm working on it.
Start: 623.19:12:52.0000
  End: 623.19:12:52.6237
```

```
time_check.py 2bit.m5b
Start Time: 566.12:10:10.0000
  Stop Time: 566.12:10:10.1559
```

The “date code” in the above examples is the 3 least-significant digits of the Modified Julian Date (MJD). The MJD is defined in Section B of the “Astronomical Almanac” and in 2008 is given by: 54465 + day of year (DOY). Therefore, in 2008 the Mark 5B “date code” is 465 + DOY.

```
bstate 2bit.m5b 100
```

ch	--	-	+	++	--	-	+	++	gfact
0	41711	84031	82960	41298	16.7	33.2	33.6	16.5	1.07
1	48485	76818	76640	48057	19.4	30.7	30.7	19.2	0.95
2	50976	74837	72690	51497	20.4	29.1	29.9	20.6	0.91
3	50186	74926	74590	50298	20.1	29.8	30.0	20.1	0.92
4	47720	78170	77152	46958	19.1	30.9	31.3	18.8	0.97
5	48753	77121	76023	48103	19.5	30.4	30.8	19.2	0.95
6	46371	78573	78224	46832	18.5	31.3	31.4	18.7	0.98
7	44481	81065	79702	44752	17.8	31.9	32.4	17.9	1.01
8	48164	77136	76669	48031	19.3	30.7	30.9	19.2	0.95
9	47718	77739	76876	47667	19.1	30.8	31.1	19.1	0.96
10	43296	82539	81352	42813	17.3	32.5	33.0	17.1	1.04
11	46701	79146	78697	45456	18.7	31.5	31.7	18.2	0.99
12	43526	82236	79995	44243	17.4	32.0	32.9	17.7	1.02
13	51520	73822	72917	51741	20.6	29.2	29.5	20.7	0.90
14	45475	80074	79228	45223	18.2	31.7	32.0	18.1	1.00
15	44273	80917	80483	44327	17.7	32.2	32.4	17.7	1.02

```
vlbi2 1bit.m5b 1bit.m5b ; mv dd1.pos 1bit.ps
```

produces the autocorrelation spectra shown in the PostScript file 1bit.ps.

```
vlbi2 2bit.m5b 2bit.m5b -2bit 1 ; mv dd1.pos 2bit.ps
```

produces the autocorrelation spectra shown in the PostScript file 2bit.ps.

Channel 13 has a strong harmonic of the 5-MHz reference at 4.010 MHz, which corresponds to a sky frequency that is a multiple of 5 MHz.

You can see this harmonic in the phase cal with

```
bpcal 2bit.m5b 4010
```

```
integration time 0.062 sec
```

ch	amp	phase(dg)
0	46	-44.5
1	44	-55.4
2	41	64.6
3	44	-76.7
4	43	49.9
5	44	61.8
6	45	64.4
7	44	140.6
8	0	-151.1
9	2	101.5
10	38	72.9
11	40	-69.8
12	42	47.0
13	358	147.9
14	32	-154.6
15	38	-139.0

Compare the above with normal phase cal at 3.01 MHz:

```
bpcal 2bit.m5b 3010
```

```
integration time 0.062 sec
```

ch	amp	phase(dg)
0	44	-175.5
1	43	173.4
2	42	-70.6
3	43	146.2
4	42	-78.4
5	44	-69.0
6	45	-67.5
7	43	9.9
8	2	-58.1
9	2	-2.3
10	36	-58.6
11	40	152.1
12	41	-82.7
13	37	-157.9
14	33	73.6
15	37	92.3

Channels 8 and 9 are lower side-band, with phase cal at n MHz – 10 KHz:

```
bpcal 2bit.m5b 2990
```

```
integration time 0.062 sec
```

ch	amp	phase(dg)
0	1	145.1
1	2	83.6
2	1	-32.0
3	3	50.1
4	0	-166.6
5	2	83.0
6	3	147.1
7	3	131.5
8	44	-104.1
9	43	62.1
10	2	-108.0
11	2	67.9
12	3	-99.0
13	2	-80.1
14	4	133.6
15	3	141.1

Acknowledgements:

bstate and bpcal were written by Roger Cappallo at Haystack.

time_check uses a header_reader perl script written by David Graham at MPI.

time_check.py was written by Pablo de Vicente at Yebes.

vlbi2 was written by Alan Rogers at Haystack .