

Various Filters and Techniques for:

Serial Data Analysis

Part 2, Vicon Protocol

11 July 2008

Eric Hamilton



Contents

1	Filters for use with Vicon's Protocols	3
1.1	BAT files for Vicon Processing	3
1.1.1	FixVicon.bat for use with Aurora 99 files	3
1.1.2	Visampa.1 Test File from an Aurora 99	4
1.2	AVicon1	5
1.2.1	Listing of AVicon1.l	6
1.3	AVicon2	8
1.3.1	Listing of AVicon2.l	9
1.4	AVicon3	11
1.4.1	Listing of AVicon3.l	12
1.5	AVicon4	14
1.5.1	Listing of AVicon4.l	15
1.6	BAT files for Vicon Keyboard Processing	17
1.6.1	ProcV.bat for use with keyboard files	17
1.6.2	Visampv.1 Test File from a Vicon keyboard	18
1.7	Vicon01	19
1.7.1	Listing of Vicon01.l	20
1.8	Vicon01x	22
1.8.1	Listing of Vicon01x.l	23
1.9	Vicon02	25
1.9.1	Listing of Vicon02.l	26
1.10	Special Software for the Reno 2008 Trip	28
1.11	Batch files	28
1.11.1	HH.BAT	28
1.11.2	PLOTUM.BAT	28
1.11.3	reno03.txt Reno Test File	29
1.12	GetSpeed	30
1.12.1	Listing of GetSpeed.l	31
1.13	IdIt	32
1.13.1	Listing of IdIt.l	33
1.14	IdStatus	39
1.14.1	Listing of IdStatus.l	40
1.15	JustSped	42
1.15.1	Listing of JustSped.l	43
1.16	NoStatus	45
1.16.1	Listing of NoStatus.l	46
1.17	ProcDCE	47
1.17.1	Listing of ProcDCE.l	48
1.18	Other files	50
1.18.1	Listing of hplot.gp	50

1 Filters for use with Vicon's Protocols

The Vicon protocols are difficult to work with due to their heavy, and constant, use of polling. In the processing that follows, an important step is the removal of the polling and the resultant replies. Different equipment types have differing poll rates.

The test file shown in Section 1.1.2, page 4 was collected from a Vicon Aurora 99 multiplexer. The test file shown in Section 1.6.2, page 18 were captured from a Vicon keyboard. The keyboard and possibly the multiplexer, have different modes of communications. Unfortunately the active modes were not captured during the data captures.

1.1 BAT files for Vicon Processing

1.1.1 FixVicon.bat for use with Aurora 99 files

```

1 @echo off
2 rem $Header: d:/Binder4/sda2/RCS/FixVicon.bat,v 1.1 2008-03-06 08:05:22-08 Hamilton Exp Hamilton $
3 makesecs %1.txt %1.out
4 enbl %1.out %1.nbl
5 avicon1 %1.nbl %1.01
6 avicon2 %1.01 %1.02
7 avicon3 %1.02 %1.03
8 avicon4 %1.03 %1.04

```

²\$Header: d:/Binder4/sda2/RCS/Vico.inc,v 1.2 2008-06-10 09:20:58-07 Hamilton Exp Hamilton \$

1.1.2 Visampa.1 Test File from an Aurora 99

1 FTS capture buffer (3/29/2006 12:02:17 PM)
 2 Event 1 (3/29/2006 11:57:26.605388 AM) through
 3 Event 45,853 (3/29/2006 12:02:01.425299 PM)

4

5 Sd Event# Timestamp Hx

6 DCE 1 3/29/2006 11:57:26.605388 AM 80

7 DCE 2 3/29/2006 11:57:26.607468 AM 01

8 DTE 3 3/29/2006 11:57:26.611658 AM 00

9 DCE 4 3/29/2006 11:57:26.627052 AM 80

10 DCE 5 3/29/2006 11:57:26.629131 AM 01

11 DTE 6 3/29/2006 11:57:26.633320 AM 00

12 DCE 7 3/29/2006 11:57:26.649149 AM 80

13 DCE 8 3/29/2006 11:57:26.651228 AM 01

14 DTE 9 3/29/2006 11:57:26.655184 AM 00

15 DCE 10 3/29/2006 11:57:26.670604 AM 80

16 DCE 11 3/29/2006 11:57:26.672687 AM 01

17 DTE 12 3/29/2006 11:57:26.676849 AM 00

18 DCE 13 3/29/2006 11:57:26.693523 AM 80

19 DCE 14 3/29/2006 11:57:26.695602 AM 01

20 DTE 15 3/29/2006 11:57:26.699765 AM 00

21 DCE 16 3/29/2006 11:57:26.715188 AM 80

22 DCE 17 3/29/2006 11:57:26.717269 AM 01

23 DTE 18 3/29/2006 11:57:26.721421 AM 00

24 DCE 19 3/29/2006 11:57:26.736858 AM 80

25 DCE 20 3/29/2006 11:57:26.738935 AM 01

26 DTE 21 3/29/2006 11:57:26.743088 AM 00

27 DCE 22 3/29/2006 11:57:26.758310 AM 80

28 DCE 23 3/29/2006 11:57:26.760391 AM 01

29 DTE 24 3/29/2006 11:57:26.764543 AM 00

30 DCE 25 3/29/2006 11:57:26.779976 AM 80

31 DCE 26 3/29/2006 11:57:26.782058 AM 01

32 DTE 27 3/29/2006 11:57:26.786209 AM 00

33 DCE 28 3/29/2006 11:57:26.802893 AM 80

34 DCE 29 3/29/2006 11:57:26.804974 AM 01

35 DTE 30 3/29/2006 11:57:26.809112 AM 00

36 DCE 31 3/29/2006 11:57:26.824558 AM 80

37 DCE 32 3/29/2006 11:57:26.826641 AM 01

38 DTE 33 3/29/2006 11:57:26.830779 AM 00

39 DCE 34 3/29/2006 11:57:26.846225 AM 80

40 DCE 35 3/29/2006 11:57:26.848308 AM 01

41 DTE 36 3/29/2006 11:57:26.852446 AM 00

42 DCE 37 3/29/2006 11:57:26.867683 AM 80

43 DCE 38 3/29/2006 11:57:26.869766 AM 01

44 DTE 39 3/29/2006 11:57:26.873891 AM 00

45 DCE 40 3/29/2006 11:57:26.889349 AM 80

46 DCE 41 3/29/2006 11:57:26.891433 AM 01

47 DTE 42 3/29/2006 11:57:26.895558 AM 00

48 DCE 43 3/29/2006 11:57:26.911224 AM 80

49 DCE 44 3/29/2006 11:57:26.913308 AM 01

50 DTE 45 3/29/2006 11:57:26.917432 AM 00

51 DCE 46 3/29/2006 11:57:26.934353 AM 80

52 DCE 47 3/29/2006 11:57:26.936433 AM 01

53 DTE 48 3/29/2006 11:57:26.940545 AM 00

54 DCE 49 3/29/2006 11:57:26.956016 AM 80

55 DCE 50 3/29/2006 11:57:26.958100 AM 01

56 DTE 51 3/29/2006 11:57:26.962211 AM 00

57 DCE 52 3/29/2006 11:57:26.977682 AM 80

58 DCE 53 3/29/2006 11:57:26.979766 AM 01

59 DTE 54 3/29/2006 11:57:26.983878 AM 00

60 DCE 55 3/29/2006 11:57:26.999141 AM 80

61 DCE 56 3/29/2006 11:57:27.001224 AM 01

62 DTE 57 3/29/2006 11:57:27.005323 AM 00

63 DCE 58 3/29/2006 11:57:27.020808 AM 80

64 DCE 59 3/29/2006 11:57:27.022891 AM 01

65 DTE 60 3/29/2006 11:57:27.026990 AM 00

66 DCE 61 3/29/2006 11:57:27.043724 AM 80

67 DCE 62 3/29/2006 11:57:27.045808 AM 01

68 DTE 63 3/29/2006 11:57:27.049906 AM 00

69 DCE 64 3/29/2006 11:57:27.065391 AM 80

70 DCE 65 3/29/2006 11:57:27.067474 AM 01

71 DTE 66 3/29/2006 11:57:27.071560 AM 00

72 DCE 67 3/29/2006 11:57:27.086849 AM 80

73 DCE 68 3/29/2006 11:57:27.088933 AM 01

74 DTE 69 3/29/2006 11:57:27.093019 AM 00

75 DCE 70 3/29/2006 11:57:27.108516 AM 80

76 DCE 71 3/29/2006 11:57:27.110612 AM 01

77 DTE 72 3/29/2006 11:57:27.114685 AM 00

78 DCE 73 3/29/2006 11:57:27.130197 AM 80

79 DCE 74 3/29/2006 11:57:27.132281 AM 01

80 DTE 75 3/29/2006 11:57:27.136338 AM 00

81 DCE 76 3/29/2006 11:57:27.153529 AM 80

82 DCE 77 3/29/2006 11:57:27.155612 AM 01

83 DTE 78 3/29/2006 11:57:27.159672 AM 00

84 DCE 79 3/29/2006 11:57:27.175195 AM 80

85 DCE 80 3/29/2006 11:57:27.177279 AM 01

86 DTE 81 3/29/2006 11:57:27.181339 AM 00

87 DCE 82 3/29/2006 11:57:27.196862 AM 80

88 DCE 83 3/29/2006 11:57:27.198946 AM 01

89 DTE 84 3/29/2006 11:57:27.202992 AM 00

90 DCE 85 3/29/2006 11:57:27.218320 AM 80

91 DCE 86 3/29/2006 11:57:27.220404 AM 01

92 DTE 87 3/29/2006 11:57:27.224452 AM 00

93 DCE 88 3/29/2006 11:57:27.239987 AM 80

94 DCE 89 3/29/2006 11:57:27.242071 AM 01

95 DTE 90 3/29/2006 11:57:27.246118 AM 00

96 DCE 91 3/29/2006 11:57:27.261445 AM 80

97 DCE 92 3/29/2006 11:57:27.263529 AM 01

98 DTE 93 3/29/2006 11:57:27.267562 AM 00

99 DCE 94 3/29/2006 11:57:27.284572 AM 80

100 DCE 95 3/29/2006 11:57:27.286654 AM 01

101 DTE 96 3/29/2006 11:57:27.290687 AM 00

102 DCE 97 3/29/2006 11:57:27.306238 AM 80

103 DCE 98 3/29/2006 11:57:27.308321 AM 01

104 DTE 99 3/29/2006 11:57:27.312355 AM 00

105 DCE 100 3/29/2006 11:57:27.327699 AM 80

106 DCE 101 3/29/2006 11:57:27.329779 AM 01

107 DTE 102 3/29/2006 11:57:27.333802 AM 00

108 DCE 103 3/29/2006 11:57:27.349364 AM 80

109 DCE 104 3/29/2006 11:57:27.351445 AM 01

110 DTE 105 3/29/2006 11:57:27.355466 AM 00

111 DCE 106 3/29/2006 11:57:27.371031 AM 80

112 DCE 107 3/29/2006 11:57:27.373112 AM 01

113 DTE 108 3/29/2006 11:57:27.377134 AM 00

114 DCE 109 3/29/2006 11:57:27.394362 AM 80

115 DCE 110 3/29/2006 11:57:27.396445 AM 01

116 DTE 111 3/29/2006 11:57:27.400452 AM 00

117 DCE 112 3/29/2006 11:57:27.416237 AM 80

118 DCE 113 3/29/2006 11:57:27.418320 AM 01

119 DTE 114 3/29/2006 11:57:27.422328 AM 00

120 DCE 115 3/29/2006 11:57:27.437280 AM 80

121 DCE 116 3/29/2006 11:57:27.439362 AM 01

122 DTE 117 3/29/2006 11:57:27.443578 AM 00

123 DCE 118 3/29/2006 11:57:27.457696 AM 80

124 DCE 119 3/29/2006 11:57:27.459778 AM 01

125 DTE 120 3/29/2006 11:57:27.463773 AM 00

126 DCE 121 3/29/2006 11:57:27.478320 AM 80

127 DCE 122 3/29/2006 11:57:27.480403 AM 01

128 DTE 123 3/29/2006 11:57:27.484398 AM 00

129 DCE 124 3/29/2006 11:57:27.498946 AM 80

130 DCE 125 3/29/2006 11:57:27.501028 AM 01

131 DTE 126 3/29/2006 11:57:27.505023 AM 00

132 DCE 127 3/29/2006 11:57:27.519362 AM 80

133 DCE 128 3/29/2006 11:57:27.521445 AM 01

134 DTE 129 3/29/2006 11:57:27.525428 AM 00

135 DCE 130 3/29/2006 11:57:27.541032 AM 80

136 DCE 131 3/29/2006 11:57:27.543114 AM 01

137 DTE 132 3/29/2006 11:57:27.547302 AM 00

138 DCE 133 3/29/2006 11:57:27.564363 AM 80

1.2 AVicon1

AVICON1 is used to group pairs of bytes together when they occur on DCE lines only.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file FixVicon.bat (Section 1.1.1, page 3) shows use of this command.
6. **Listing of: hvia.nbl a typical input file.**

```

1 $ Id: MakeSecs.1,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
2 $Header: d:/Binder4/sda2/RCS/Hvia.nbl,v 1.2 2008-03-06 13:01:33-08 Hamilton Exp Hamilton $
3 FTS capture buffer (3/29/2006 12:02:17 PM)
4 Event 1 (3/29/2006 11:57:26.605388 AM) through
5 Event 45,853 (3/29/2006 12:02:01.425299 PM)
6     1,      1: DCE      1      0.000000  0.000000 80
7     1,      2: DCE      2      0.002080  0.002080 01
8     1,      1: DTE      3      0.006270  0.002080 00
9     2,      3: DCE      4      0.021664  0.002080 80
10    2,      4: DCE      5      0.023743  0.002080 01
11    2,      2: DTE      6      0.027932  0.004189 00
12    3,      5: DCE      7      0.043761  0.015829 80
13    3,      6: DCE      8      0.045840  0.002079 01
14    3,      3: DTE      9      0.049796  0.002079 00
15    4,      7: DCE     10      0.065216  0.002079 80
16    4,      8: DCE     11      0.067299  0.002079 01
17    4,      4: DTE     12      0.071461  0.004162 00
18    5,      9: DCE     13      0.088135  0.016674 80
19    5,     10: DCE     14      0.090214  0.002079 01
20    5,      5: DTE     15      0.094377  0.002079 00

```

7. **Listing of: hvia.01 a typical output file.**

```

1 $ Id: MakeSecs.1,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
2 $Header: d:/Binder4/sda2/RCS/Hvia.01,v 1.2 2008-03-06 13:01:33-08 Hamilton Exp Hamilton $
3 FTS capture buffer (3/29/2006 12:02:17 PM)
4 Event 1 (3/29/2006 11:57:26.605388 AM) through
5 Event 45,853 (3/29/2006 12:02:01.425299 PM)
6     1,      1: DCE      1      0.000000  0.000000 80 01
7     1,      1: DTE      3      0.006270  0.002080 00
8     2,      3: DCE      4      0.021664  0.002080 80 01
9     2,      2: DTE      6      0.027932  0.004189 00
10    3,      5: DCE      7      0.043761  0.015829 80 01
11    3,      3: DTE      9      0.049796  0.002079 00
12    4,      7: DCE     10      0.065216  0.002079 80 01
13    4,      4: DTE     12      0.071461  0.004162 00
14    5,      9: DCE     13      0.088135  0.016674 80 01
15    5,      5: DTE     15      0.094377  0.002079 00
16    6,     11: DCE     16      0.109800  0.002079 80 01
17    6,      6: DTE     18      0.116033  0.004152 00
18    7,     13: DCE     19      0.131470  0.015437 80 01
19    7,      7: DTE     21      0.137700  0.002077 00
20    8,     15: DCE     22      0.152922  0.002077 80 01

```

1.2.1 Listing of AVicon1.1

```

1  %{
2  // $Header: d:/Binder4/sda2/RCS/AVicon1.1,v 1.2 2008-03-06 09:36:32-08 Hamilton Exp Hamilton $
3  // AVicon1 is used to group pairs of bytes together when they occur
4  //      on DCE lines only.
5  #include <stdio.h>
6  #include <stdlib.h>
7  #undef yywrap
8  %}
9  %x ONELINE
10 %%
11 .*: DCE ".*          ECHO; BEGIN ONELINE;
12 <ONELINE>" "[0-9a-f]{2}$ ECHO; BEGIN 0;
13 <ONELINE>.\n          ;
14
15 .\n          ECHO;
16
17 %%
18 // yywrap is for End Of File processing, 1 = done
19 int yywrap(void)
20 {
21     return 1;
22 }
23
24 main(int argc, char *argv[])
25 {
26     fprintf(stderr,"$Id: AVicon1.1,v 1.2 2008-03-06 09:36:32-08 Hamilton Exp Hamilton $\n");
27
28     // Do we have to deal with stdin/stdout?
29     // 3 = one for the program name and one each for input and output names
30     if (argc == 3) {
31         // Nope
32         // Get over the program, etc., name
33         argc--;
34         argv++;
35         yyin = fopen(argv[0],"r");
36         if (yyin == NULL) {
37             fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
38                     argv[0]);
39             exit(EXIT_FAILURE);
40         }
41
42         // Now get over the input file name and get to the output file name
43         argc--;
44         argv++;
45         yyout = fopen(argv[0],"w");
46         if (yyout == NULL) {
47             fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
48                     argv[0]);
49             exit(EXIT_FAILURE);
50         }
51     }
52     else {
53         // Woops we are using std... type io
54         yyin = stdin;
55         yyout = stdout;
56     }
57
58     yylex();
59     exit(EXIT_SUCCESS);
60 }
61
62 // $Log: AVicon1.1,v $
63 // Revision 1.2  2008-03-06 09:36:32-08  Hamilton
64 // Normal end of day data saving
65 //
66 // Revision 1.1  2008-03-06 08:05:22-08  Hamilton
67 // Initial revision
68 //

```

```
69 // Revision 1.2 2006-03-30 09:39:39-08 Hamilton
70 // Normal end of day data saving
71 //
72 // Revision 1.1 2006-03-30 07:12:25-08 Hamilton
73 // Initial revision
74 //
75 // Revision 1.1 2005-10-18 15:15:05-07 Hamilton
76 // Initial revision
77 //
78 // Revision 1.6 1999-07-09 07:54:17-07 Hamilton
79 // Deleted references to NOGETOPT, as I am not using it with the new vesion
80 // of FLEX.
81 //
82 // Revision 1.5 1997-01-12 21:12:50-08 ham
83 // Added in define for NO_GETOPT. When set this will inhibit the inclusion
84 // of GNU's getopt routines, kinda.
85 //
86 // Revision 1.4 1996-11-14 19:36:28-0800 ham
87 // Add in end of file processing by adding the yywrap() function.
88 //
89
```

1.3 AVicon2

AVICON2 is used to delete all lines ending with a “_80_01\n” poll message. and their responses of “_20\n”.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file FixVicon.bat (Section 1.1.1, page 3) shows use of this command.
6. **Listing of: hvia.01 a typical input file.**

```

1 $ Id: MakeSecs.1,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
2 $Header: d:/Binder4/sda2/RCS/Hvia.01,v 1.2 2008-03-06 13:01:33-08 Hamilton Exp Hamilton $
3 FTS capture buffer (3/29/2006 12:02:17 PM)
4 Event 1 (3/29/2006 11:57:26.605388 AM) through
5 Event 45,853 (3/29/2006 12:02:01.425299 PM)
6     1,      1: DCE      1      0.000000  0.000000 80 01
7     1,      1: DTE      3      0.006270  0.002080 00
8     2,      3: DCE      4      0.021664  0.002080 80 01
9     2,      2: DTE      6      0.027932  0.004189 00
10    3,      5: DCE      7      0.043761  0.015829 80 01
11    3,      3: DTE      9      0.049796  0.002079 00
12    4,      7: DCE     10      0.065216  0.002079 80 01
13    4,      4: DTE     12      0.071461  0.004162 00
14    5,      9: DCE     13      0.088135  0.016674 80 01
15    5,      5: DTE     15      0.094377  0.002079 00
16    6,     11: DCE     16      0.109800  0.002079 80 01
17    6,      6: DTE     18      0.116033  0.004152 00
18    7,     13: DCE     19      0.131470  0.015437 80 01
19    7,      7: DTE     21      0.137700  0.002077 00
20    8,     15: DCE     22      0.152922  0.002077 80 01

```

7. **Listing of: hvia.02 a typical output file.**

```

1 $ Id: MakeSecs.1,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
2 $Header: d:/Binder4/sda2/RCS/Hvia.02,v 1.3 2008-03-06 14:11:08-08 Hamilton Exp Hamilton $
3 FTS capture buffer (3/29/2006 12:02:17 PM)
4 Event 1 (3/29/2006 11:57:26.605388 AM) through
5 Event 45,853 (3/29/2006 12:02:01.425299 PM)
6    339,     677: DCE    1015    7.398695  0.016410 80 51
7    339,     339: DTE    1017    7.405187  0.002082 41 00
8    340,     679: DCE    1019    7.409736  0.002630 40 00
9    340,     681: DCE    1021    7.413903  0.002083 00 00
10   340,     683: DCE    1023    7.418069  0.002073 00 08
11   340,     685: DCE    1025    7.422238  0.002086 00 00
12   340,     341: DTE    1027    7.427480  0.002081 00
13   341,     687: DCE    1028    7.444528  0.002081 80 51
14   341,     342: DTE    1030    7.455396  0.008785 41 00
15   342,     689: DCE    1032    7.459943  0.002463 41 00
16   342,     691: DCE    1034    7.464110  0.002084 00 00
17   342,     693: DCE    1036    7.468278  0.002075 00 08
18   342,     695: DCE    1038    7.472444  0.002083 00 00
19   342,     344: DTE    1040    7.477676  0.002082 00
20   343,     697: DCE    1041    7.494542  0.002082 80 51

```


1.3.1 Listing of AVicon2.1

```

1  %{
2  // $Header: d:/Binder4/sda2/RCS/AVicon2.1,v 1.2 2008-03-06 09:36:32-08 Hamilton Exp Hamilton $
3  // AVicon2 is used to delete all lines ending with a " 80 01\n" poll message.
4  // and their responses of " 20\n"
5  #include <stdio.h>
6  #include <stdlib.h>
7  #undef yywrap
8  %}
9  %x ONELINE LINETWO
10 %%
11 .*" 80 01"\n.*": DTE ".*" 00"\n ;
12 .*" 80 01"\n.*": DTE ".*" 01"\n ;
13 .*" 80 01"\n.*": DTE ".*" 02"\n ;
14
15 .*" 80 51"$          ECHO; BEGIN ONELINE;
16 <ONELINE>\n.*        ECHO; BEGIN LINETWO;
17
18 <LINETWO>" "[0-9a-f]{2}$ fprintf(yyout, " %c%c", yytext[1], yytext[2]); BEGIN 0;
19 <LINETWO>.           ;
20 <LINETWO>\n          ;
21
22
23
24 .|\n      ECHO;
25 %%
26 // yywrap is for End Of File processing, 1 = done
27 int yywrap(void)
28 {
29     return 1;
30 }
31
32 main(int argc, char *argv[])
33 {
34     fprintf(stderr, "$Id: AVicon2.1,v 1.2 2008-03-06 09:36:32-08 Hamilton Exp Hamilton $\n");
35
36     // Do we have to deal with stdin/stdout?
37     // 3 = one for the program name and one each for input and output names
38     if (argc == 3) {
39         // Nope
40         // Get over the program, etc., name
41         argc--;
42         argv++;
43         yyin = fopen(argv[0], "r");
44         if (yyin == NULL) {
45             fprintf(stderr, "\aCould not open \"%s\" for reading, quitting",
46                     argv[0]);
47             exit(EXIT_FAILURE);
48         }
49
50         // Now get over the input file name and get to the output file name
51         argc--;
52         argv++;
53         yyout = fopen(argv[0], "w");
54         if (yyout == NULL) {
55             fprintf(stderr, "\aCould not open \"%s\" for writing, quitting",
56                     argv[0]);
57             exit(EXIT_FAILURE);
58         }
59     }
60     else {
61         // Woops we are using std... type io
62         yyin = stdin;
63         yyout = stdout;
64     }
65
66     yylex();
67     exit(EXIT_SUCCESS);
68 }

```

```
69
70 // $Log: AVicon2.1,v $
71 // Revision 1.2  2008-03-06 09:36:32-08  Hamilton
72 // Normal end of day data saving
73 //
74 // Revision 1.1  2008-03-06 08:05:22-08  Hamilton
75 // Initial revision
76 //
77 // Revision 1.2  2006-03-30 09:39:39-08  Hamilton
78 // Normal end of day data saving
79 //
80 // Revision 1.1  2006-03-30 07:12:29-08  Hamilton
81 // Initial revision
82 //
83 // Revision 1.1  2005-10-18 15:15:25-07  Hamilton
84 // Initial revision
85 //
86 // Revision 1.6  1999-07-09 07:54:17-07  Hamilton
87 // Deleted references to NOGETOPT, as I am not using it with the new vesion
88 // of FLEX.
89 //
90 // Revision 1.5  1997-01-12 21:12:50-08  ham
91 // Added in define for NO_GETOPT.  When set this will inhibit the inclusion
92 // of GNU's getopt routines, kinda.
93 //
94 // Revision 1.4  1996-11-14 19:36:28-0800  ham
95 // Add in end of file processing by adding the yywrap() function.
96 //
97
```

1.4 AVicon3

AVICON3 is used to block up groups of command bytes that start up ending with “_xx_00\n”, and xx = 08, 09, 10, 11, 20, 21, 40, 41, 28, 29, 30, 31, 48, 49, 50, 51. xx must not be equal to 00, and consist of four total lines of data pairs. It is assumed that this is the output of AVICON2.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon’s Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file FixVicon.bat (Section 1.1.1, page 3) shows use of this command.
6. Listing of: hvia.02 a typical input file.

```

1 $ Id: MakeSecs.1,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
2 $Header: d:/Binder4/sda2/RCS/Hvia.02,v 1.3 2008-03-06 14:11:08-08 Hamilton Exp Hamilton $
3 FTS capture buffer (3/29/2006 12:02:17 PM)
4 Event 1 (3/29/2006 11:57:26.605388 AM) through
5 Event 45,853 (3/29/2006 12:02:01.425299 PM)
6   339,    677: DCE    1015    7.398695    0.016410 80 51
7   339,    339: DTE    1017    7.405187    0.002082 41 00
8   340,    679: DCE    1019    7.409736    0.002630 40 00
9   340,    681: DCE    1021    7.413903    0.002083 00 00
10  340,    683: DCE    1023    7.418069    0.002073 00 08
11  340,    685: DCE    1025    7.422238    0.002086 00 00
12  340,    341: DTE    1027    7.427480    0.002081 00
13  341,    687: DCE    1028    7.444528    0.002081 80 51
14  341,    342: DTE    1030    7.455396    0.008785 41 00
15  342,    689: DCE    1032    7.459943    0.002463 41 00
16  342,    691: DCE    1034    7.464110    0.002084 00 00
17  342,    693: DCE    1036    7.468278    0.002075 00 08
18  342,    695: DCE    1038    7.472444    0.002083 00 00
19  342,    344: DTE    1040    7.477676    0.002082 00
20  343,    697: DCE    1041    7.494542    0.002082 80 51

```

7. Listing of: hvia.03 a typical output file.

```

1 $ Id: MakeSecs.1,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
2 $Header: d:/Binder4/sda2/RCS/Hvia.03,v 1.3 2008-03-06 14:11:09-08 Hamilton Exp Hamilton $
3 FTS capture buffer (3/29/2006 12:02:17 PM)
4 Event 1 (3/29/2006 11:57:26.605388 AM) through
5 Event 45,853 (3/29/2006 12:02:01.425299 PM)
6   339,    677: DCE    1015    7.398695    0.016410 80 51
7   339,    339: DTE    1017    7.405187    0.002082 41 00
8   340,    679: DCE    1019    7.409736    0.002630 40 00 00 00 08 00 00
9   340,    341: DTE    1027    7.427480    0.002081 00
10  341,    687: DCE    1028    7.444528    0.002081 80 51
11  341,    342: DTE    1030    7.455396    0.008785 41 00
12  342,    689: DCE    1032    7.459943    0.002463 41 00 00 00 08 00 00
13  342,    344: DTE    1040    7.477676    0.002082 00
14  343,    697: DCE    1041    7.494542    0.002082 80 51
15  343,    345: DTE    1043    7.505592    0.008969 41 00
16  344,    699: DCE    1045    7.510165    0.002490 41 00 00 00 08 00 00
17  344,    347: DTE    1053    7.527717    0.002085 00
18  345,    707: DCE    1054    7.544335    0.002085 80 51
19  345,    348: DTE    1056    7.555788    0.009436 41 00
20  346,    709: DCE    1058    7.560373    0.002501 40 00 00 00 08 00 00

```

1.4.1 Listing of AVicon3.1

```

1  %{
2  // $Header: d:/Binder4/sda2/RCS/AVicon3.1,v 1.2 2008-03-06 09:36:32-08 Hamilton Exp Hamilton $
3  // AVicon3 is used to block up groups of command bytes that start up ending
4  //      with " xx 00\n", and xx = 08, 09, 10, 11, 20, 21, 40, 41
5  //      28, 29, 30, 31, 48, 49, 50, 51
6  //      xx must not be equal to 00
7  //      and consist of four total lines of data pairs. It is assumed that
8  //      this is the output of AVicon2.
9  //
10 #include <stdio.h>
11 #include <stdlib.h>
12 #undef yywrap
13 %}
14 %x GOTDCE LINETWO LINETHREE LINEFOUR
15 %%
16 .*: DTE ".*\n"          ECHO;
17 .*: DCE ".*" 80 51"$    ECHO;
18
19 .*: DCE ".*" 00 00"$    ECHO; BEGIN LINETWO;
20 .*: DCE ".*" 00 01"$    ECHO; BEGIN LINETWO;
21 .*: DCE ".*" 00 02"$    ECHO; BEGIN LINETWO;
22 .*: DCE ".*" 00 04"$    ECHO; BEGIN LINETWO;
23 .*: DCE ".*" 00 08"$    ECHO; BEGIN LINETWO;
24 .*: DCE ".*" 00 10"$    ECHO; BEGIN LINETWO;
25 .*: DCE ".*" 00 20"$    ECHO; BEGIN LINETWO;
26 .*: DCE ".*" 00 40"$    ECHO; BEGIN LINETWO;
27 .*: DCE ".*" 00 80"$    ECHO; BEGIN LINETWO;
28 .*: DCE ".*" 08 00"$    ECHO; BEGIN LINETWO;
29 .*: DCE ".*" 09 00"$    ECHO; BEGIN LINETWO;
30 .*: DCE ".*" 10 00"$    ECHO; BEGIN LINETWO;
31 .*: DCE ".*" 11 00"$    ECHO; BEGIN LINETWO;
32 .*: DCE ".*" 20 00"$    ECHO; BEGIN LINETWO;
33 .*: DCE ".*" 21 00"$    ECHO; BEGIN LINETWO;
34 .*: DCE ".*" 28 00"$    ECHO; BEGIN LINETWO;
35 .*: DCE ".*" 29 00"$    ECHO; BEGIN LINETWO;
36 .*: DCE ".*" 30 00"$    ECHO; BEGIN LINETWO;
37 .*: DCE ".*" 31 00"$    ECHO; BEGIN LINETWO;
38 .*: DCE ".*" 40 00"$    ECHO; BEGIN LINETWO;
39 .*: DCE ".*" 41 00"$    ECHO; BEGIN LINETWO;
40 .*: DCE ".*" 48 00"$    ECHO; BEGIN LINETWO;
41 .*: DCE ".*" 49 00"$    ECHO; BEGIN LINETWO;
42 .*: DCE ".*" 50 00"$    ECHO; BEGIN LINETWO;
43 .*: DCE ".*" 51 00"$    ECHO; BEGIN LINETWO;
44
45 <LINETWO>.{6}$          ECHO; BEGIN LINETHREE;
46 <LINETWO>.\n            ;
47 <LINETHREE>.{6}$        ECHO; BEGIN LINEFOUR;
48 <LINETHREE>.\n          ;
49 <LINEFOUR>.{6}$         ECHO; BEGIN 0;
50 <LINEFOUR>.\n           ;
51
52 .|\n                    ECHO;
53 %%
54 // yywrap is for End Of File processing, 1 = done
55 int yywrap(void)
56 {
57     return 1;
58 }
59
60 main(int argc, char *argv[])
61 {
62     fprintf(stderr, "$Id: AVicon3.1,v 1.2 2008-03-06 09:36:32-08 Hamilton Exp Hamilton $\n");
63
64     // Do we have to deal with stdin/stdout?
65     // 3 = one for the program name and one each for input and output names
66     if (argc == 3) {
67         // Nope
68         // Get over the program, etc., name

```

```

69         argc--;
70         argv++;
71         yyin = fopen(argv[0], "r");
72         if (yyin == NULL) {
73             fprintf(stderr, "\aCould not open \"%s\" for reading, quitting",
74                     argv[0]);
75             exit(EXIT_FAILURE);
76         }
77
78         // Now get over the input file name and get to the output file name
79         argc--;
80         argv++;
81         yyout = fopen(argv[0], "w");
82         if (yyout == NULL) {
83             fprintf(stderr, "\aCould not open \"%s\" for writing, quitting",
84                     argv[0]);
85             exit(EXIT_FAILURE);
86         }
87     }
88     else {
89         // Woops we are using std... type io
90         yyin = stdin;
91         yyout = stdout;
92     }
93
94     yylex();
95     exit(EXIT_SUCCESS);
96 }
97
98 // $Log: AVicon3.1,v $
99 // Revision 1.2  2008-03-06 09:36:32-08  Hamilton
100 // Normal end of day data saving
101 //
102 // Revision 1.1  2008-03-06 08:05:22-08  Hamilton
103 // Initial revision
104 //
105 // Revision 1.2  2006-03-30 09:39:39-08  Hamilton
106 // Normal end of day data saving
107 //
108 // Revision 1.1  2006-03-30 07:12:32-08  Hamilton
109 // Initial revision
110 //
111 // Revision 1.1  2005-10-18 15:15:10-07  Hamilton
112 // Initial revision
113 //
114 // Revision 1.6  1999-07-09 07:54:17-07  Hamilton
115 // Deleted references to NOGETOPT, as I am not using it with the new vesion
116 // of FLEX.
117 //
118 // Revision 1.5  1997-01-12 21:12:50-08  ham
119 // Added in define for NO_GETOPT. When set this will inhibit the inclusion
120 // of GNU's getopt routines, kinda.
121 //
122 // Revision 1.4  1996-11-14 19:36:28-0800  ham
123 // Add in end of file processing by adding the yywrap() function.
124 //
125

```

1.5 AVicon4

AVICON4 is used to left justify the interesting part of the data.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file FixVicon.bat (Section 1.1.1, page 3) shows use of this command.
6. **Listing of: hvia.03 a typical input file.**

```

1  $ Id: MakeSecs.1,v 1.11 2008-02-20 08:06:33-08 Hamilton Exp Hamilton $
2  $Header: d:/Binder4/sda2/RCS/Hvia.03,v 1.3 2008-03-06 14:11:09-08 Hamilton Exp Hamilton $
3  FTS capture buffer (3/29/2006 12:02:17 PM)
4  Event 1 (3/29/2006 11:57:26.605388 AM) through
5  Event 45,853 (3/29/2006 12:02:01.425299 PM)
6      339,      677: DCE      1015      7.398695      0.016410 80 51
7      339,      339: DTE      1017      7.405187      0.002082 41 00
8      340,      679: DCE      1019      7.409736      0.002630 40 00 00 00 08 00 00
9      340,      341: DTE      1027      7.427480      0.002081 00
10     341,      687: DCE      1028      7.444528      0.002081 80 51
11     341,      342: DTE      1030      7.455396      0.008785 41 00
12     342,      689: DCE      1032      7.459943      0.002463 41 00 00 00 08 00 00
13     342,      344: DTE      1040      7.477676      0.002082 00
14     343,      697: DCE      1041      7.494542      0.002082 80 51
15     343,      345: DTE      1043      7.505592      0.008969 41 00
16     344,      699: DCE      1045      7.510165      0.002490 41 00 00 00 08 00 00
17     344,      347: DTE      1053      7.527717      0.002085 00
18     345,      707: DCE      1054      7.544335      0.002085 80 51
19     345,      348: DTE      1056      7.555788      0.009436 41 00
20     346,      709: DCE      1058      7.560373      0.002501 40 00 00 00 08 00 00

```

7. **Listing of: hvia.04 a typical output file.**

```

1  ton Exp Hamilton $
2  $Header: d:/Binder4/sda2/RCS/Hvia.04,v 1.2 2008-03-06 14:11:09-08 Hamilton Exp Hamilton $
3  FTS capture buffer (3/29/2006 12:02:17 PM)
4  Event 1 (3/29/2006 11:57:26.605388 AM) through
5  Event 45,853 (3/29/2006 12:02:01.425299 PM)
6      80 51
7      41 00
8      40 00 00 00 00 08 00 00
9      00
10     80 51
11     41 00
12     41 00 00 00 00 08 00 00
13     00
14     80 51
15     41 00
16     41 00 00 00 00 08 00 00
17     00
18     80 51
19     41 00
20     40 00 00 00 00 08 00 00

```

1.5.1 Listing of AVicon4.1

```

1  %{
2  // $Header: d:/Binder4/sda2/RCS/AVicon4.1,v 1.1 2008-03-06 08:05:22-08 Hamilton Exp Hamilton $
3  // AVicon4 is used to left justify the interesting part of the data.
4  #include <stdio.h>
5  #include <stdlib.h>
6  #undef yywrap
7  %}
8  %%
9  .{52}      ;
10 .{2}$      ECHO;
11 .{5}$      ECHO;
12 .{23}$     ECHO;
13
14 %%
15 //.\n      ECHO;
16 // yywrap is for End Of File processing, 1 = done
17 int yywrap(void)
18 {
19     return 1;
20 }
21
22 main(int argc, char *argv[])
23 {
24     fprintf(stderr,"$Id: AVicon4.1,v 1.1 2008-03-06 08:05:22-08 Hamilton Exp Hamilton $\n");
25
26     // Do we have to deal with stdin/stdout?
27     // 3 = one for the program name and one each for input and output names
28     if (argc == 3) {
29         // Nope
30         // Get over the program, etc., name
31         argc--;
32         argv++;
33         yyin = fopen(argv[0],"r");
34         if (yyin == NULL) {
35             fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
36                     argv[0]);
37             exit(EXIT_FAILURE);
38         }
39
40         // Now get over the input file name and get to the output file name
41         argc--;
42         argv++;
43         yyout = fopen(argv[0],"w");
44         if (yyout == NULL) {
45             fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
46                     argv[0]);
47             exit(EXIT_FAILURE);
48         }
49     }
50     else {
51         // Woops we are using std... type io
52         yyin = stdin;
53         yyout = stdout;
54     }
55
56     yylex();
57     exit(EXIT_SUCCESS);
58 }
59
60 // $Log: AVicon4.1,v $
61 // Revision 1.1 2008-03-06 08:05:22-08 Hamilton
62 // Initial revision
63 //
64 // Revision 1.2 2006-03-30 09:39:40-08 Hamilton
65 // Normal end of day data saving
66 //
67 // Revision 1.1 2006-03-30 07:12:34-08 Hamilton
68 // Initial revision

```

```
69 //
70 // Revision 1.1 2005-10-20 08:14:17-07 Hamilton
71 // Initial revision
72 //
73 // Revision 1.6 1999-07-09 07:54:17-07 Hamilton
74 // Deleted references to NOGETOPT, as I am not using it with the new vesion
75 // of FLEX.
76 //
77 // Revision 1.5 1997-01-12 21:12:50-08 ham
78 // Added in define for NO_GETOPT. When set this will inhibit the inclusion
79 // of GNU's getopt routines, kinda.
80 //
81 // Revision 1.4 1996-11-14 19:36:28-0800 ham
82 // Add in end of file processing by adding the yywrap() function.
83 //
84
```


1.6 BAT files for Vicon Keyboard Processing

1.6.1 ProcV.bat for use with keyboard files

```
1 @echo off
2 rem $Header: d:/Binder4/sda2/RCS/ProcV.bat,v 1.2 2008-03-06 11:47:37-08 Hamilton Exp Hamilton $
3 MakeSecs %1.txt %1.01
4 Enbl %1.01 %1.02
5 Vicon01 %1.02 %1.03
6 Vicon01x %1.03 %1.04
7 Vicon02 %1.04 %1.05
```

1.6.2 Visampv.1 Test File from a Vicon key-board

1 FTS capture file: E:\Capture\newtest2.cfa (10/18/2005 8:13:27 AM)

2 Event 1 (10/18/2005 8:12:47.309236 AM) through

3 Event 2,636 (10/18/2005 8:13:08.772429 AM)

4

5 Sd Timestamp Hx

6 DCE 10/18/2005 8:12:47.309236 AM 80

7 DCE 10/18/2005 8:12:47.311319 AM 01

8 DTE 10/18/2005 8:12:47.315562 AM 02

9 DCE 10/18/2005 8:12:47.332151 AM 80

10 DCE 10/18/2005 8:12:47.334230 AM 01

11 DTE 10/18/2005 8:12:47.338266 AM 02

12 DCE 10/18/2005 8:12:47.353813 AM 80

13 DCE 10/18/2005 8:12:47.355897 AM 01

14 DTE 10/18/2005 8:12:47.359919 AM 02

15 DCE 10/18/2005 8:12:47.375480 AM 80

16 DCE 10/18/2005 8:12:47.377563 AM 01

17 DTE 10/18/2005 8:12:47.381586 AM 02

18 DCE 10/18/2005 8:12:47.396938 AM 80

19 DCE 10/18/2005 8:12:47.399022 AM 01

20 DTE 10/18/2005 8:12:47.403253 AM 02

21 DCE 10/18/2005 8:12:47.418605 AM 80

22 DCE 10/18/2005 8:12:47.420688 AM 01

23 DTE 10/18/2005 8:12:47.424906 AM 02

24 DCE 10/18/2005 8:12:47.442212 AM 80

25 DCE 10/18/2005 8:12:47.444297 AM 01

26 DTE 10/18/2005 8:12:47.448306 AM 02

27 DCE 10/18/2005 8:12:47.463255 AM 80

28 DCE 10/18/2005 8:12:47.465337 AM 01

29 DTE 10/18/2005 8:12:47.469334 AM 02

30 DCE 10/18/2005 8:12:47.483879 AM 80

31 DCE 10/18/2005 8:12:47.485968 AM 01

32 DTE 10/18/2005 8:12:47.489964 AM 02

33 DCE 10/18/2005 8:12:47.504296 AM 80

34 DCE 10/18/2005 8:12:47.506380 AM 01

35 DTE 10/18/2005 8:12:47.510376 AM 02

36 DCE 10/18/2005 8:12:47.524922 AM 80

37 DCE 10/18/2005 8:12:47.527004 AM 01

38 DTE 10/18/2005 8:12:47.530988 AM 02

39 DCE 10/18/2005 8:12:47.545337 AM 80

40 DCE 10/18/2005 8:12:47.547422 AM 01

41 DTE 10/18/2005 8:12:47.551406 AM 02

42 DCE 10/18/2005 8:12:47.567005 AM 80

43 DCE 10/18/2005 8:12:47.569088 AM 01

44 DTE 10/18/2005 8:12:47.573287 AM 02

45 DCE 10/18/2005 8:12:47.589087 AM 80

46 DCE 10/18/2005 8:12:47.591172 AM 01

47 DTE 10/18/2005 8:12:47.595355 AM 02

48 DCE 10/18/2005 8:12:47.610547 AM 80

49 DCE 10/18/2005 8:12:47.612629 AM 01

50 DTE 10/18/2005 8:12:47.616808 AM 02

51 DCE 10/18/2005 8:12:47.633468 AM 80

52 DCE 10/18/2005 8:12:47.635492 AM 01

53 DTE 10/18/2005 8:12:47.639504 AM 02

54 DCE 10/18/2005 8:12:47.655131 AM 80

55 DCE 10/18/2005 8:12:47.657211 AM 01

56 DTE 10/18/2005 8:12:47.661380 AM 02

57 DCE 10/18/2005 8:12:47.677212 AM 80

58 DCE 10/18/2005 8:12:47.679297 AM 01

59 DTE 10/18/2005 8:12:47.683461 AM 02

60 DCE 10/18/2005 8:12:47.698813 AM 80

61 DCE 10/18/2005 8:12:47.700898 AM 01

62 DTE 10/18/2005 8:12:47.705117 AM 02

63 DCE 10/18/2005 8:12:47.720338 AM 80

64 DCE 10/18/2005 8:12:47.722421 AM 01

65 DTE 10/18/2005 8:12:47.726573 AM 02

66 DCE 10/18/2005 8:12:47.743462 AM 80

67 DCE 10/18/2005 8:12:47.745546 AM 01

68 DTE 10/18/2005 8:12:47.749686 AM 02

69 DCE 10/18/2005 8:12:47.764906 AM 80

70 DCE 10/18/2005 8:12:47.767004 AM 01

71 DTE 10/18/2005 8:12:47.771145 AM 02

72 DCE 10/18/2005 8:12:47.786587 AM 80

73 DCE 10/18/2005 8:12:47.788671 AM 01

74 DTE 10/18/2005 8:12:47.792811 AM 02

75 DCE 10/18/2005 8:12:47.808254 AM 80

76 DCE 10/18/2005 8:12:47.810337 AM 01

77 DTE 10/18/2005 8:12:47.814464 AM 02

78 DCE 10/18/2005 8:12:47.829947 AM 80

79 DCE 10/18/2005 8:12:47.832036 AM 01

80 DTE 10/18/2005 8:12:47.836143 AM 02

81 DCE 10/18/2005 8:12:47.852861 AM 80

82 DCE 10/18/2005 8:12:47.854946 AM 01

83 DTE 10/18/2005 8:12:47.858850 AM 02

84 DCE 10/18/2005 8:12:47.874267 AM 80

85 DCE 10/18/2005 8:12:47.876403 AM 01

86 DTE 10/18/2005 8:12:47.880503 AM 02

87 DCE 10/18/2005 8:12:47.895985 AM 80

88 DCE 10/18/2005 8:12:47.898071 AM 01

89 DTE 10/18/2005 8:12:47.902171 AM 02

90 DCE 10/18/2005 8:12:47.917653 AM 80

91 DCE 10/18/2005 8:12:47.919735 AM 01

92 DTE 10/18/2005 8:12:47.923825 AM 02

93 DCE 10/18/2005 8:12:47.939736 AM 80

94 DCE 10/18/2005 8:12:47.941820 AM 01

95 DTE 10/18/2005 8:12:47.945908 AM 02

96 DCE 10/18/2005 8:12:47.962654 AM 80

97 DCE 10/18/2005 8:12:47.964736 AM 01

98 DTE 10/18/2005 8:12:47.968565 AM 02

99 DCE 10/18/2005 8:12:47.986404 AM 80

100 DCE 10/18/2005 8:12:47.988486 AM 01

101 DTE 10/18/2005 8:12:47.992568 AM 02

102 DCE 10/18/2005 8:12:48.008278 AM 80

103 DCE 10/18/2005 8:12:48.010361 AM 01

104 DTE 10/18/2005 8:12:48.014436 AM 02

105 DCE 10/18/2005 8:12:48.030286 AM 80

106 DCE 10/18/2005 8:12:48.032445 AM 01

107 DTE 10/18/2005 8:12:48.036506 AM 02

108 DCE 10/18/2005 8:12:48.053278 AM 80

109 DCE 10/18/2005 8:12:48.055360 AM 01

110 DTE 10/18/2005 8:12:48.059631 AM 02

111 DCE 10/18/2005 8:12:48.074735 AM 80

112 DCE 10/18/2005 8:12:48.076826 AM 01

113 DTE 10/18/2005 8:12:48.081090 AM 02

114 DCE 10/18/2005 8:12:48.096404 AM 80

115 DCE 10/18/2005 8:12:48.098486 AM 01

116 DTE 10/18/2005 8:12:48.102534 AM 02

117 DCE 10/18/2005 8:12:48.118044 AM 80

118 DCE 10/18/2005 8:12:48.120153 AM 01

119 DTE 10/18/2005 8:12:48.124202 AM 02

120 DCE 10/18/2005 8:12:48.139736 AM 80

121 DCE 10/18/2005 8:12:48.141820 AM 01

122 DTE 10/18/2005 8:12:48.145867 AM 02

123 DCE 10/18/2005 8:12:48.161182 AM 80

124 DCE 10/18/2005 8:12:48.163278 AM 01

125 DTE 10/18/2005 8:12:48.167522 AM 02

126 DCE 10/18/2005 8:12:48.184527 AM 80

127 DCE 10/18/2005 8:12:48.186611 AM 01

128 DTE 10/18/2005 8:12:48.190854 AM 02

129 DCE 10/18/2005 8:12:48.206194 AM 80

130 DCE 10/18/2005 8:12:48.208284 AM 01

131 DTE 10/18/2005 8:12:48.212509 AM 02

132 DCE 10/18/2005 8:12:48.227860 AM 80

133 DCE 10/18/2005 8:12:48.229869 AM 01

134 DTE 10/18/2005 8:12:48.234177 AM 02

135 DCE 10/18/2005 8:12:48.249464 AM 80

136 DCE 10/18/2005 8:12:48.251611 AM 01

137 DTE 10/18/2005 8:12:48.255841 AM 02

138 DCE 10/18/2005 8:12:48.271195 AM 80

1.7 Vicon01

VICON01 is used to group pairs of bytes together when they occur on DCE lines only.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MOLD_FMT.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file ProcV.bat (Section 1.6.1, page 17) shows use of this command.
6. **Listing of: hviv.02 a typical input file.**

```

1 $Header: d:/Binder4/sda2/RCS/Hviv.02,v 1.1 2008-03-06 11:48:42-08 Hamilton Exp Hamilton $
2 FTS capture file: E:\Capture\newtest2.cfa (10/18/2005 8:13:27 AM)
3 Event 1 (10/18/2005 8:12:47.309236 AM) through
4 Event 2,636 (10/18/2005 8:13:08.772429 AM)
5 1, 1: DCE 0.000000 0.000000 80
6 1, 2: DCE 0.002083 0.002083 01
7 1, 1: DTE 0.006326 0.002083 02
8 2, 3: DCE 0.022915 0.002083 80
9 2, 4: DCE 0.024994 0.002083 01
10 2, 2: DTE 0.029030 0.004036 02
11 3, 5: DCE 0.044577 0.015547 80
12 3, 6: DCE 0.046661 0.002084 01
13 3, 3: DTE 0.050683 0.002084 02
14 4, 7: DCE 0.066244 0.002084 80
15 4, 8: DCE 0.068327 0.002084 01
16 4, 4: DTE 0.072350 0.004023 02
17 5, 9: DCE 0.087702 0.015352 80
18 5, 10: DCE 0.089786 0.002084 01
19 5, 5: DTE 0.094017 0.002084 02
20 6, 11: DCE 0.109369 0.002084 80

```

7. **Listing of: hviv.03 a typical output file.**

```

1 $Header: d:/Binder4/sda2/RCS/Hviv.03,v 1.2 2008-03-06 14:11:09-08 Hamilton Exp Hamilton $
2 FTS capture file: E:\Capture\newtest2.cfa (10/18/2005 8:13:27 AM)
3 Event 1 (10/18/2005 8:12:47.309236 AM) through
4 Event 2,636 (10/18/2005 8:13:08.772429 AM)
5 1, 1: DCE 0.000000 0.000000 80 01
6 1, 1: DTE 0.006326 0.002083 02
7 2, 3: DCE 0.022915 0.002083 80 01
8 2, 2: DTE 0.029030 0.004036 02
9 3, 5: DCE 0.044577 0.015547 80 01
10 3, 3: DTE 0.050683 0.002084 02
11 4, 7: DCE 0.066244 0.002084 80 01
12 4, 4: DTE 0.072350 0.004023 02
13 5, 9: DCE 0.087702 0.015352 80 01
14 5, 5: DTE 0.094017 0.002084 02
15 6, 11: DCE 0.109369 0.002084 80 01
16 6, 6: DTE 0.115670 0.004218 02
17 7, 13: DCE 0.132976 0.017306 80 01
18 7, 7: DTE 0.139070 0.002085 02
19 8, 15: DCE 0.154019 0.002085 80 01
20 8, 8: DTE 0.160098 0.003997 02

```

1.7.1 Listing of Vicon01.1

```

1  %{
2  //123456789 123456789 123456789 123456789 123456789
3  // 106735, 289515: DCE 2920.747120 0.002082 80
4  // 106735, 289516: DCE 2920.749203 0.002082 01
5  // 106735, 115125: DTE 2920.753225 0.004022 02
6  // 106736, 289517: DCE 2920.770036 0.016811 80
7  // 106736, 289518: DCE 2920.772120 0.002084 01
8  // 106736, 115126: DTE 2920.776335 0.002084 02
9  // 106737, 289519: DCE 2920.791632 0.002084 80
10 // 106737, 289520: DCE 2920.793784 0.002084 01
11 // 106737, 115127: DTE 2920.798000 0.004216 02
12 // 106738, 289521: DCE 2920.813161 0.015161 80
13 // 106738, 289522: DCE 2920.815245 0.002084 01
14 // 106738, 115128: DTE 2920.819447 0.002084 02
15 // 106739, 289523: DCE 2920.831285 0.002084 80
16 // 106739, 289524: DCE 2920.833369 0.002084 01
17 // 106739, 115129: DTE 2920.837363 0.003994 02
18 // 106740, 289525: DCE 2920.846910 0.009547 86
19 // 106740, 289526: DCE 2920.848995 0.002085 00
20 // 106740, 289527: DCE 2920.892535 0.043540 86
21 // 106740, 289528: DCE 2920.894620 0.002085 01
22 // 106740, 289529: DCE 2920.937744 0.043124 80
23 // 106740, 289530: DCE 2920.939827 0.002083 01
24 // 106740, 115130: DTE 2920.943378 0.002083 02
25 // 106741, 289531: DCE 2920.955660 0.002083 88
26 // 106741, 289532: DCE 2920.957744 0.002083 0b
27 // 106741, 289533: DCE 2921.002118 0.002083 88
28 // 106741, 289534: DCE 2921.004202 0.002083 0c
29 // 106741, 289535: DCE 2921.045441 0.002083 80
30 // 106741, 289536: DCE 2921.047548 0.002083 01
31 // 106741, 115131: DTE 2921.051268 0.003720 02
32 //123456789 123456789 123456789 123456789 123456789
33 // $Header: d:/Binder4/sda2/RCS/Vicon01.1,v 1.2 2008-03-06 09:36:33-08 Hamilton Exp Hamilton $
34 // Vicon01 is used to group pairs of bytes together when they occur
35 // on DCE lines only.
36 #include <stdio.h>
37 #include <stdlib.h>
38 #undef yywrap
39 %}
40 %x ONELINE
41 %%
42 .{15}": DCE ".* ECHO; BEGIN ONELINE;
43 <ONELINE>.{46} ;
44 <ONELINE>.{3}\n ECHO; BEGIN 0;
45 <ONELINE>\n ;
46
47 .|\n ECHO;
48 %%
49 // yywrap is for End Of File processing, 1 = done
50 int yywrap(void)
51 {
52     return 1;
53 }
54
55 main(int argc, char *argv[])
56 {
57     fprintf(stderr, "$Id: Vicon01.1,v 1.2 2008-03-06 09:36:33-08 Hamilton Exp Hamilton $\n");
58
59     // Do we have to deal with stdin/stdout?
60     // 3 = one for the program name and one each for input and output names
61     if (argc == 3) {
62         // Nope
63         // Get over the program, etc., name
64         argc--;
65         argv++;
66         yyin = fopen(argv[0], "r");
67         if (yyin == NULL) {
68             fprintf(stderr, "\aCould not open \"%s\" for reading, quitting",

```

```

69             argv[0]);
70         exit(EXIT_FAILURE);
71     }
72
73     // Now get over the input file name and get to the output file name
74     argc--;
75     argv++;
76     yyout = fopen(argv[0],"w");
77     if (yyout == NULL) {
78         fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
79             argv[0]);
80         exit(EXIT_FAILURE);
81     }
82 }
83 else {
84     // Woops we are using std... type io
85     yyin = stdin;
86     yyout = stdout;
87 }
88
89 yylex();
90 exit(EXIT_SUCCESS);
91 }
92
93 // $Log: Vicon01.1,v $
94 // Revision 1.2  2008-03-06 09:36:33-08  Hamilton
95 // Normal end of day data saving
96 //
97 // Revision 1.1  2008-03-06 08:05:23-08  Hamilton
98 // Initial revision
99 //
100 // Revision 1.1  2005-10-18 15:15:05-07  Hamilton
101 // Initial revision
102 //
103 // Revision 1.6  1999-07-09 07:54:17-07  Hamilton
104 // Deleted references to NOGETOPT, as I am not using it with the new vesion
105 // of FLEX.
106 //
107 // Revision 1.5  1997-01-12 21:12:50-08  ham
108 // Added in define for NO_GETOPT.  When set this will inhibit the inclusion
109 // of GNU's getopt routines, kinda.
110 //
111 // Revision 1.4  1996-11-14 19:36:28-0800  ham
112 // Add in end of file processing by adding the yywrap() function.
113 //
114

```

1.8 Vicon01x

VICON01X is used to delete all lines ending with a “_80_01\n” poll message. and their responses of “_20\n”.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MOLD_FMT.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file ProcV.bat (Section 1.6.1, page 17) shows use of this command.
6. **Listing of: hviv.03 a typical input file.**

```

1 $Header: d:/Binder4/sda2/RCS/Hviv.03,v 1.2 2008-03-06 14:11:09-08 Hamilton Exp Hamilton $
2 FTS capture file: E:\Capture\newtest2.cfa (10/18/2005 8:13:27 AM)
3 Event 1 (10/18/2005 8:12:47.309236 AM) through
4 Event 2,636 (10/18/2005 8:13:08.772429 AM)
5 1, 1: DCE 0.000000 0.000000 80 01
6 1, 1: DTE 0.006326 0.002083 02
7 2, 3: DCE 0.022915 0.002083 80 01
8 2, 2: DTE 0.029030 0.004036 02
9 3, 5: DCE 0.044577 0.015547 80 01
10 3, 3: DTE 0.050683 0.002084 02
11 4, 7: DCE 0.066244 0.002084 80 01
12 4, 4: DTE 0.072350 0.004023 02
13 5, 9: DCE 0.087702 0.015352 80 01
14 5, 5: DTE 0.094017 0.002084 02
15 6, 11: DCE 0.109369 0.002084 80 01
16 6, 6: DTE 0.115670 0.004218 02
17 7, 13: DCE 0.132976 0.017306 80 01
18 7, 7: DTE 0.139070 0.002085 02
19 8, 15: DCE 0.154019 0.002085 80 01
20 8, 8: DTE 0.160098 0.003997 02

```

7. **Listing of: hviv.04 a typical output file.**

```

1 $Header: d:/Binder4/sda2/RCS/Hviv.04,v 1.2 2008-03-06 14:11:09-08 Hamilton Exp Hamilton $
2 FTS capture file: E:\Capture\newtest2.cfa (10/18/2005 8:13:27 AM)
3 Event 1 (10/18/2005 8:12:47.309236 AM) through
4 Event 2,636 (10/18/2005 8:13:08.772429 AM)
5 632, 1263: DCE 13.778306 0.002090 81 0c
6 640, 1281: DCE 13.997056 0.002055 81 0a
7 648, 1299: DCE 14.214972 0.002045 81 06
8 656, 1317: DCE 14.435819 0.002082 81 04
9 664, 1335: DCE 14.657909 0.002082 81 02
10 673, 1355: DCE 14.900622 0.011408 83 0a
11 680, 1371: DCE 15.096384 0.002081 80 0c
12 681, 1375: DCE 15.166608 0.012458 83 06
13 688, 1391: DCE 15.363123 0.002084 80 0b
14 689, 1395: DCE 15.428136 0.013071 83 04
15 690, 1399: DCE 15.495011 0.002156 85 0f
16 696, 1413: DCE 15.664185 0.002009 80 08
17 697, 1417: DCE 15.725217 0.011183 82 0f
18 698, 1421: DCE 15.786259 0.002105 85 0d
19 705, 1437: DCE 15.976854 0.011500 82 0c
20 706, 1441: DCE 16.038362 0.002149 85 07

```

1.8.1 Listing of Vicon01x.l

```

1  %{
2  // 106745, 289551: DCE 2921.297965 0.002080 80 01
3  // 106745, 115135: DTE 2921.304136 0.004089 02
4  // 106746, 289553: DCE 2921.321090 0.016954 80 01
5  // 106746, 115136: DTE 2921.327252 0.002082 02
6  // 106747, 289555: DCE 2921.342547 0.002082 80 01
7  // 106747, 115137: DTE 2921.348703 0.004071 02
8  // 106748, 289557: DCE 2921.358798 0.010095 86 08
9  // 106748, 289559: DCE 2921.402554 0.041673 86 09
10 //
11 // $Header: d:/Binder4/sda2/RCS/Vicon01x.l,v 1.2 2008-03-06 09:36:33-08 Hamilton Exp Hamilton $
12 // Vicon01x is used to delete all lines ending with a " 80 01\n" poll message.
13 // and their responses of " 20\n"
14 #include <stdio.h>
15 #include <stdlib.h>
16 #undef yywrap
17 %}
18 %%
19 .*" 80 01"\n ;
20 .*" 02"\n ;
21
22 .|\n ECHO;
23 %%
24 // yywrap is for End Of File processing, 1 = done
25 int yywrap(void)
26 {
27     return 1;
28 }
29
30 main(int argc, char *argv[])
31 {
32     fprintf(stderr, "$Id: Vicon01x.l,v 1.2 2008-03-06 09:36:33-08 Hamilton Exp Hamilton $\n");
33
34     // Do we have to deal with stdin/stdout?
35     // 3 = one for the program name and one each for input and output names
36     if (argc == 3) {
37         // Nope
38         // Get over the program, etc., name
39         argc--;
40         argv++;
41         yyin = fopen(argv[0], "r");
42         if (yyin == NULL) {
43             fprintf(stderr, "\aCould not open \"%s\" for reading, quitting",
44                     argv[0]);
45             exit(EXIT_FAILURE);
46         }
47
48         // Now get over the input file name and get to the output file name
49         argc--;
50         argv++;
51         yyout = fopen(argv[0], "w");
52         if (yyout == NULL) {
53             fprintf(stderr, "\aCould not open \"%s\" for writing, quitting",
54                     argv[0]);
55             exit(EXIT_FAILURE);
56         }
57     }
58     else {
59         // Woops we are using std... type io
60         yyin = stdin;
61         yyout = stdout;
62     }
63
64     yylex();
65     exit(EXIT_SUCCESS);
66 }
67
68 // $Log: Vicon01x.l,v $

```

```
69 // Revision 1.2 2008-03-06 09:36:33-08 Hamilton
70 // Normal end of day data saving
71 //
72 // Revision 1.1 2008-03-06 08:05:23-08 Hamilton
73 // Initial revision
74 //
75 // Revision 1.1 2005-10-18 15:15:25-07 Hamilton
76 // Initial revision
77 //
78 // Revision 1.6 1999-07-09 07:54:17-07 Hamilton
79 // Deleted references to NOGETOPT, as I am not using it with the new vesion
80 // of FLEX.
81 //
82 // Revision 1.5 1997-01-12 21:12:50-08 ham
83 // Added in define for NO_GETOPT. When set this will inhibit the inclusion
84 // of GNU's getopt routines, kinda.
85 //
86 // Revision 1.4 1996-11-14 19:36:28-0800 ham
87 // Add in end of file processing by adding the yywrap() function.
88 //
89
```


1.9 Vicon02

VICON02 is used to block up groups of command bytes that start up ending with “_xx_00\n”, and xx = 08, 09, 10, 11, 20, 21, 40, 41, 28, 29, 30, 31, 48, 49, 50, 51. xx must not be equal to 00 and consist of four total lines of data pairs. It is assumed that this is the output of VICON01X .

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon’s Protocols only.
3. This filter requires preprocessing by MOLD_FMT.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file ProcV.bat (Section 1.6.1, page 17) shows use of this command.
6. **Listing of: hviv.04 a typical input file.**

```

1 $Header: d:/Binder4/sda2/RCS/Hviv.04,v 1.2 2008-03-06 14:11:09-08 Hamilton Exp Hamilton $
2 FTS capture file: E:\Capture\newtest2.cfa (10/18/2005 8:13:27 AM)
3 Event 1 (10/18/2005 8:12:47.309236 AM) through
4 Event 2,636 (10/18/2005 8:13:08.772429 AM)
5 632, 1263: DCE 13.778306 0.002090 81 0c
6 640, 1281: DCE 13.997056 0.002055 81 0a
7 648, 1299: DCE 14.214972 0.002045 81 06
8 656, 1317: DCE 14.435819 0.002082 81 04
9 664, 1335: DCE 14.657909 0.002082 81 02
10 673, 1355: DCE 14.900622 0.011408 83 0a
11 680, 1371: DCE 15.096384 0.002081 80 0c
12 681, 1375: DCE 15.166608 0.012458 83 06
13 688, 1391: DCE 15.363123 0.002084 80 0b
14 689, 1395: DCE 15.428136 0.013071 83 04
15 690, 1399: DCE 15.495011 0.002156 85 0f
16 696, 1413: DCE 15.664185 0.002009 80 08
17 697, 1417: DCE 15.725217 0.011183 82 0f
18 698, 1421: DCE 15.786259 0.002105 85 0d
19 705, 1437: DCE 15.976854 0.011500 82 0c
20 706, 1441: DCE 16.038362 0.002149 85 07

```

7. **Listing of: hviv.05 a typical output file.**

```

1 $Header: d:/Binder4/sda2/RCS/Hviv.05,v 1.2 2008-03-06 14:11:09-08 Hamilton Exp Hamilton $
2 FTS capture file: E:\Capture\newtest2.cfa (10/18/2005 8:13:27 AM)
3 Event 1 (10/18/2005 8:12:47.309236 AM) through
4 Event 2,636 (10/18/2005 8:13:08.772429 AM)
5 632, 1263: DCE 13.778306 0.002090 81 0c
6 640, 1281: DCE 13.997056 0.002055 81 0a
7 648, 1299: DCE 14.214972 0.002045 81 06
8 656, 1317: DCE 14.435819 0.002082 81 04
9 664, 1335: DCE 14.657909 0.002082 81 02
10 673, 1355: DCE 14.900622 0.011408 83 0a
11 680, 1371: DCE 15.096384 0.002081 80 0c
12 681, 1375: DCE 15.166608 0.012458 83 06
13 688, 1391: DCE 15.363123 0.002084 80 0b
14 689, 1395: DCE 15.428136 0.013071 83 04
15 690, 1399: DCE 15.495011 0.002156 85 0f
16 696, 1413: DCE 15.664185 0.002009 80 08
17 697, 1417: DCE 15.725217 0.011183 82 0f
18 698, 1421: DCE 15.786259 0.002105 85 0d
19 705, 1437: DCE 15.976854 0.011500 82 0c
20 706, 1441: DCE 16.038362 0.002149 85 07

```

1.9.1 Listing of Vicon02.1

```

1  %{
2  //123456789 123456789 123456789 123456789 123456789 12
3  // 2958, 8957: DCE 117.872900 0.002479 20 00
4  // 2958, 8959: DCE 117.877065 0.002082 00 00
5  // 2958, 8961: DCE 117.881232 0.002082 00 30
6  // 2958, 8963: DCE 117.885401 0.002086 00 00
7  // 2959, 8965: DCE 117.907276 0.002082 80 51
8  // 2959, 3128: DTE 117.918339 0.008981 41
9  // 2960, 8967: DCE 117.922899 0.002479 21 00
10 // 2960, 8969: DCE 117.927064 0.002082 00 00
11 // 2960, 8971: DCE 117.931233 0.002084 00 38
12 // 2960, 8973: DCE 117.935399 0.002085 00 00
13 // 2961, 8975: DCE 117.957274 0.002083 80 51
14 // 2961, 3131: DTE 117.968534 0.009177 41
15 //123456789 123456789 123456789 123456789 123456789 12
16 //
17 // $Header: d:/Binder4/sda2/RCS/Vicon02.1,v 1.2 2008-03-06 09:36:33-08 Hamilton Exp Hamilton $
18 // Vicon03 is used to block up groups of command bytes that start up ending
19 // with " xx 00\n", and xx = 08, 09, 10, 11, 20, 21, 40, 41
20 // 28, 29, 30, 31, 48, 49, 50, 51
21 // xx must not be equal to 00
22 // and consist of four total lines of data pairs. It is assumed that
23 // this is the output of Vicon01x.
24 //
25 #include <stdio.h>
26 #include <stdlib.h>
27 #undef yywrap
28 %}
29 %x LINETWO LINETHREE LINEFOUR
30 %%
31 .*" 08 00" |
32 .*" 09 00" |
33 .*" 10 00" |
34 .*" 11 00" |
35 .*" 20 00" |
36 .*" 21 00" |
37 .*" 28 00" |
38 .*" 29 00" |
39 .*" 30 00" |
40 .*" 31 00" |
41 .*" 40 00" |
42 .*" 41 00" |
43 .*" 48 00" |
44 .*" 49 00" |
45 .*" 50 00" |
46 .*" 51 00" ECHO; BEGIN LINETWO;
47 <LINETWO>\n ;
48 <LINETWO>.{46} ;
49 <LINETWO>.{6} ECHO; BEGIN LINETHREE;
50
51 <LINETHREE>\n ;
52 <LINETHREE>.{46} ;
53 <LINETHREE>.{6} ECHO; BEGIN LINEFOUR;
54
55 <LINEFOUR>\n ;
56 <LINEFOUR>.{46} ;
57 <LINEFOUR>.{6}\n ECHO; BEGIN 0;
58
59 .|\n ECHO;
60 %%
61 // yywrap is for End Of File processing, 1 = done
62 int yywrap(void)
63 {
64     return 1;
65 }
66
67 main(int argc, char *argv[])
68 {

```

```

69     fprintf(stderr,"$Id: Vicon02.1,v 1.2 2008-03-06 09:36:33-08 Hamilton Exp Hamilton $\n");
70
71     // Do we have to deal with stdin/stdout?
72     // 3 = one for the program name and one each for input and output names
73     if (argc == 3) {
74         // Nope
75         // Get over the program, etc., name
76         argc--;
77         argv++;
78         yyin = fopen(argv[0],"r");
79         if (yyin == NULL) {
80             fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
81                     argv[0]);
82             exit(EXIT_FAILURE);
83         }
84
85         // Now get over the input file name and get to the output file name
86         argc--;
87         argv++;
88         yyout = fopen(argv[0],"w");
89         if (yyout == NULL) {
90             fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
91                     argv[0]);
92             exit(EXIT_FAILURE);
93         }
94     }
95     else {
96         // Woops we are using std... type io
97         yyin = stdin;
98         yyout = stdout;
99     }
100
101     yylex();
102     exit(EXIT_SUCCESS);
103 }
104
105 // $Log: Vicon02.1,v $
106 // Revision 1.2  2008-03-06 09:36:33-08  Hamilton
107 // Normal end of day data saving
108 //
109 // Revision 1.1  2008-03-06 08:05:23-08  Hamilton
110 // Initial revision
111 //
112 // Revision 1.1  2005-10-18 15:15:10-07  Hamilton
113 // Initial revision
114 //
115 // Revision 1.6  1999-07-09 07:54:17-07  Hamilton
116 // Deleted references to NOGETOPT, as I am not using it with the new vesion
117 // of FLEX.
118 //
119 // Revision 1.5  1997-01-12 21:12:50-08  ham
120 // Added in define for NO_GETOPT.  When set this will inhibit the inclusion
121 // of GNU's getopt routines, kinda.
122 //
123 // Revision 1.4  1996-11-14 19:36:28-0800  ham
124 // Add in end of file processing by adding the yywrap() function.
125 //
126

```

1.10 Special Software for the Reno 2008 Trip

The following filters were written for use in investigating the data captures made in Reno. In addition to these six Flex filters there are two batch files, a gnuplot control file and a control file for generating the document (this is it) its self.

Note that this full document is available in d:\TXB-Vicon\captures.

1.11 Batch files

1.11.1 HH.BAT

This is the primary script for processing a data capture file.

```

1 @echo off
2 rem $Header: d:/Binder4/sda2/RCS/Hh.bat,v 1.2 2008-06-10 09:21:01-07 Hamilton Exp Hamilton $
3 rem copy original.harrahs\%1.txt %1.cap
4 dir      %1.txt | tail -3 | head -1
5 makesecs %1.txt %1.out
6 enbl     %1.out %1.nbl
7 getdce   %1.nbl %1.dce
8 getdte   %1.nbl %1.dte
9 ProcDce  %1.dce %1.pc
10 copy    %1.pc+%1.dte %1.msg
11 qsort   %1.msg %1.dat /22:13
12 IdIt    %1.dat %1.pol
13 IdStatus %1.dat %1.sts
14 copy    %1.pol+%1.sts %1.all
15 qsort   %1.all %1.srt /22:13
16 NoStatus %1.srt %1.nos
17 GetSpeed %1.nos %1.spd
18 JustSpd %1.spd %1.plt

```

1.11.2 PLOTUM.BAT

This script runs GNUPLLOT with PLOT.GP to generate the various plot figures.

```

1 @echo off
2 rem $Header: d:/Binder4/sda2/RCS/Plotum.bat,v 1.1 2008-06-10 07:58:50-07 Hamilton Exp Hamilton $
3 head -2 H06.plt > h06a.plt
4 head -1502 h06.plt | tail -500 >> h06a.plt
5 head -2 H06.plt > h06b.plt
6 head -1312 h06.plt | tail -225 >> h06b.plt
7 gnuplot plot.gp

```

1.11.3 reno03.txt Reno Test File

1 FTS capture buffer (5/15/2008 7:55:13 PM)
 2 Event 1 (5/15/2008 7:52:50.801500 PM) through
 3 Event 17,156 (5/15/2008 7:54:49.955976 PM)
 4

Sd	Event#	Timestamp	Hx
DCE	1	5/15/2008 7:52:50.801500 PM	88
DCE	2	5/15/2008 7:52:50.802539 PM	0b
DTE	3	5/15/2008 7:52:50.805679 PM	02
DCE	4	5/15/2008 7:52:50.821292 PM	88
DCE	5	5/15/2008 7:52:50.822333 PM	0b
DTE	6	5/15/2008 7:52:50.825360 PM	02
DCE	7	5/15/2008 7:52:50.833791 PM	84
DCE	8	5/15/2008 7:52:50.834827 PM	06
DCE	9	5/15/2008 7:52:50.861817 PM	88
DCE	10	5/15/2008 7:52:50.862854 PM	0b
DTE	11	5/15/2008 7:52:50.866078 PM	02
DCE	12	5/15/2008 7:52:50.877961 PM	87
DCE	13	5/15/2008 7:52:50.879001 PM	0f
DCE	14	5/15/2008 7:52:50.911194 PM	88
DCE	15	5/15/2008 7:52:50.912231 PM	0b
DTE	16	5/15/2008 7:52:50.915439 PM	02
DCE	17	5/15/2008 7:52:50.926399 PM	89
DCE	18	5/15/2008 7:52:50.927438 PM	01
DCE	19	5/15/2008 7:52:50.959315 PM	89
DCE	20	5/15/2008 7:52:50.960355 PM	06
DCE	21	5/15/2008 7:52:50.991926 PM	88
DCE	22	5/15/2008 7:52:50.992965 PM	0b
DTE	23	5/15/2008 7:52:50.996253 PM	02
DCE	24	5/15/2008 7:52:51.007236 PM	8b
DCE	25	5/15/2008 7:52:51.008278 PM	03
DCE	26	5/15/2008 7:52:51.038695 PM	8b
DCE	27	5/15/2008 7:52:51.039736 PM	05
DCE	28	5/15/2008 7:52:51.071403 PM	88
DCE	29	5/15/2008 7:52:51.072445 PM	0b
DTE	30	5/15/2008 7:52:51.075609 PM	02
DCE	31	5/15/2008 7:52:51.086611 PM	8d
DCE	32	5/15/2008 7:52:51.087654 PM	07
DCE	33	5/15/2008 7:52:51.111611 PM	88
DCE	34	5/15/2008 7:52:51.112653 PM	0b
DTE	35	5/15/2008 7:52:51.115914 PM	02
DCE	36	5/15/2008 7:52:51.125576 PM	8e
DCE	37	5/15/2008 7:52:51.126611 PM	00
DCE	38	5/15/2008 7:52:51.162868 PM	88
DCE	39	5/15/2008 7:52:51.163909 PM	0b
DTE	40	5/15/2008 7:52:51.167047 PM	02
DCE	41	5/15/2008 7:52:51.182763 PM	88
DCE	42	5/15/2008 7:52:51.183805 PM	0b
DTE	43	5/15/2008 7:52:51.186728 PM	02
DCE	44	5/15/2008 7:52:51.200785 PM	88
DCE	45	5/15/2008 7:52:51.201826 PM	0b
DTE	46	5/15/2008 7:52:51.204957 PM	02
DCE	47	5/15/2008 7:52:51.218909 PM	88
DCE	48	5/15/2008 7:52:51.219951 PM	0b
DTE	49	5/15/2008 7:52:51.222972 PM	02
DCE	50	5/15/2008 7:52:51.239013 PM	88
DCE	51	5/15/2008 7:52:51.240055 PM	0b
DTE	52	5/15/2008 7:52:51.243175 PM	02
DCE	53	5/15/2008 7:52:51.254534 PM	89
DCE	54	5/15/2008 7:52:51.255576 PM	09
DCE	55	5/15/2008 7:52:51.281729 PM	88
DCE	56	5/15/2008 7:52:51.282770 PM	0b
DTE	57	5/15/2008 7:52:51.286084 PM	02
DCE	58	5/15/2008 7:52:51.296729 PM	8a
DCE	59	5/15/2008 7:52:51.297874 PM	0e
DCE	60	5/15/2008 7:52:51.329854 PM	8b
DCE	61	5/15/2008 7:52:51.330895 PM	06
DCE	62	5/15/2008 7:52:51.362458 PM	88
DCE	63	5/15/2008 7:52:51.363499 PM	0b
DTE	64	5/15/2008 7:52:51.366689 PM	02
DCE	65	5/15/2008 7:52:51.382458 PM	88

DCE	66	5/15/2008 7:52:51.383499 PM	0b
DTE	67	5/15/2008 7:52:51.386579 PM	02
DCE	68	5/15/2008 7:52:51.396943 PM	8e
DCE	69	5/15/2008 7:52:51.397985 PM	08
DCE	70	5/15/2008 7:52:51.429652 PM	8e
DCE	71	5/15/2008 7:52:51.430693 PM	09
DCE	72	5/15/2008 7:52:51.463090 PM	88
DCE	73	5/15/2008 7:52:51.464131 PM	0b
DTE	74	5/15/2008 7:52:51.467392 PM	02
DCE	75	5/15/2008 7:52:51.483089 PM	88
DCE	76	5/15/2008 7:52:51.484235 PM	0b
DTE	77	5/15/2008 7:52:51.487288 PM	02
DCE	78	5/15/2008 7:52:51.502985 PM	88
DCE	79	5/15/2008 7:52:51.504027 PM	0b
DTE	80	5/15/2008 7:52:51.506969 PM	02
DCE	81	5/15/2008 7:52:51.521526 PM	88
DCE	82	5/15/2008 7:52:51.522568 PM	0b
DTE	83	5/15/2008 7:52:51.525712 PM	02
DCE	84	5/15/2008 7:52:51.534554 PM	86
DCE	85	5/15/2008 7:52:51.535596 PM	0c
DCE	86	5/15/2008 7:52:51.569138 PM	86
DCE	87	5/15/2008 7:52:51.570179 PM	0f
DCE	88	5/15/2008 7:52:51.604138 PM	88
DCE	89	5/15/2008 7:52:51.605179 PM	0b
DTE	90	5/15/2008 7:52:51.608505 PM	02
DCE	91	5/15/2008 7:52:51.624242 PM	88
DCE	92	5/15/2008 7:52:51.625283 PM	0b
DTE	93	5/15/2008 7:52:51.628401 PM	02
DCE	94	5/15/2008 7:52:51.638095 PM	8a
DCE	95	5/15/2008 7:52:51.639137 PM	01
DCE	96	5/15/2008 7:52:51.660701 PM	8a
DCE	97	5/15/2008 7:52:51.661748 PM	0a
DCE	98	5/15/2008 7:52:51.694144 PM	88
DCE	99	5/15/2008 7:52:51.695185 PM	0b
DTE	100	5/15/2008 7:52:51.698382 PM	02
DCE	101	5/15/2008 7:52:51.708624 PM	8c
DCE	102	5/15/2008 7:52:51.709664 PM	0d
DCE	103	5/15/2008 7:52:51.734352 PM	88
DCE	104	5/15/2008 7:52:51.735394 PM	0b
DTE	105	5/15/2008 7:52:51.738688 PM	02
DCE	106	5/15/2008 7:52:51.749352 PM	8f
DCE	107	5/15/2008 7:52:51.750393 PM	00
DCE	108	5/15/2008 7:52:51.780814 PM	8f
DCE	109	5/15/2008 7:52:51.781852 PM	03
DCE	110	5/15/2008 7:52:51.813630 PM	88
DCE	111	5/15/2008 7:52:51.814671 PM	0b
DTE	112	5/15/2008 7:52:51.817835 PM	02
DCE	113	5/15/2008 7:52:51.833213 PM	88
DCE	114	5/15/2008 7:52:51.834254 PM	0b
DTE	115	5/15/2008 7:52:51.837307 PM	02
DCE	116	5/15/2008 7:52:51.851235 PM	88
DCE	117	5/15/2008 7:52:51.852275 PM	0b
DTE	118	5/15/2008 7:52:51.855322 PM	02
DCE	119	5/15/2008 7:52:51.871025 PM	88
DCE	120	5/15/2008 7:52:51.872067 PM	0b
DTE	121	5/15/2008 7:52:51.875010 PM	02
DCE	122	5/15/2008 7:52:51.890713 PM	88
DCE	123	5/15/2008 7:52:51.891754 PM	0b
DTE	124	5/15/2008 7:52:51.894691 PM	02
DCE	125	5/15/2008 7:52:51.910505 PM	88
DCE	126	5/15/2008 7:52:51.911546 PM	0b
DTE	127	5/15/2008 7:52:51.914580 PM	02
DCE	128	5/15/2008 7:52:51.930296 PM	88
DCE	129	5/15/2008 7:52:51.931338 PM	0b
DTE	130	5/15/2008 7:52:51.934267 PM	02
DCE	131	5/15/2008 7:52:51.950094 PM	88
DCE	132	5/15/2008 7:52:51.951135 PM	0b
DTE	133	5/15/2008 7:52:51.954156 PM	02

1.12 GetSpeed

GETSPEED.L: This filter extracts pan/tilt speeds from the interpreted capture data.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file Hh.bat (Section 1.11, page 28) shows use of this command.
6. Listing of: hh03.nos a typical input file.

```

1
2
3 2558, 2558: DTE 8260 57.477922 0.003307 41
4 2559, 5707: DCE 8266 57.524199 0.041667 88 5b 00 00 00 00 00 00 00:
5 2561, 2562: DTE 8282 57.618724 0.001144 41
6 2562, 5725: DCE 8288 57.663164 0.039897 88 5b 00 00 00 00 00 08 00 00: Pan Speed 1,
7 2563, 2565: DTE 8301 57.737447 0.003555 41
8 2564, 5741: DCE 8307 57.783210 0.041186 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
9 2568, 2571: DTE 8329 57.918025 0.001042 41
10 2569, 5763: DCE 8335 57.964427 0.041876 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
11 2575, 2579: DTE 8363 58.139328 0.001041 41
12 2579, 2584: DTE 8382 58.291587 0.003287 41
13 2583, 2589: DTE 8405 58.468318 0.003345 41
14 2588, 2595: DTE 8431 58.669726 0.001039 41
15 2590, 2598: DTE 8448 58.810006 0.003561 41
16 2595, 2604: DTE 8470 58.981215 0.000873 41
17 2598, 2608: DTE 8486 59.115044 0.001043 41
18 2601, 2612: DTE 8506 59.271568 0.001040 41
19 2602, 5899: DCE 8512 59.316158 0.040000 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
20 2618, 2630: DTE 8570 59.681575 0.001090 41

```

7. Listing of: hh03.spd a typical output file.

```

1 2562, 5725: DCE 8288 57.663164 0.039897 88 5b 00 00 00 00 00 08 00 00: Pan Speed 1,
2 2564, 5741: DCE 8307 57.783210 0.041186 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
3 2569, 5763: DCE 8335 57.964427 0.041876 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
4 2602, 5899: DCE 8512 59.316158 0.040000 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
5 2654, 6057: DCE 8728 60.838736 0.040424 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
6 2656, 6073: DCE 8747 60.958639 0.039695 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
7 2658, 6089: DCE 8766 61.078954 0.040167 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
8 2661, 6107: DCE 8788 61.219480 0.041773 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
9 2667, 6141: DCE 8830 61.519394 0.041732 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
10 2675, 6175: DCE 8874 61.830343 0.039894 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
11 2695, 6233: DCE 8954 62.389786 0.038374 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
12 2701, 6257: DCE 8985 62.601841 0.040424 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
13 2773, 6471: DCE 9281 64.786216 0.036667 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
14 2776, 6489: DCE 9303 64.926952 0.041564 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
15 2789, 6547: DCE 9377 65.466354 0.042188 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
16 2808, 6615: DCE 9468 66.149719 0.039897 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
17 2836, 6707: DCE 9593 67.070289 0.038026 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
18 2891, 6873: DCE 9821 68.756307 0.041258 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
19 2893, 6889: DCE 9840 68.874849 0.040312 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
20 2895, 6905: DCE 9859 68.996522 0.041667 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,

```

1.12.1 Listing of GetSpeed.l

```

1  %{
2  // $Header: d:/Binder4/sda2/RCS/GetSpeed.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $
3  // GetSpeed is used to extract all lines from a file with the word
4  //      " Pan Speed" in them. (Note the leading blank.)
5
6  #include <stdio.h>
7  #include <stdlib.h>
8  #undef yywrap
9  %}
10 %%
11 .*" Pan Speed".*\n ECHO;
12
13 .|\n      ;
14 %%
15 // yywrap is for End Of File processing, 1 = done
16 int yywrap(void)
17 {
18     return 1;
19 }
20
21 main(int argc, char *argv[])
22 {
23     fprintf(stderr,"$Id: GetSpeed.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $\n");
24
25     // Do we have to deal with stdin/stdout?
26     // 3 = one for the program name and one each for input and output names
27     if (argc == 3) {
28         // Nope
29         // Get over the program, etc., name
30         argc--;
31         argv++;
32         yyin = fopen(argv[0],"r");
33         if (yyin == NULL) {
34             fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
35                     argv[0]);
36             exit(EXIT_FAILURE);
37         }
38
39         // Now get over the input file name and get to the output file name
40         argc--;
41         argv++;
42         yyout = fopen(argv[0],"w");
43         if (yyout == NULL) {
44             fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
45                     argv[0]);
46             exit(EXIT_FAILURE);
47         }
48     }
49     else {
50         // Woops we are using std... type io
51         yyin = stdin;
52         yyout = stdout;
53     }
54
55     yylex();
56     exit(EXIT_SUCCESS);
57 }
58
59 // $Log: GetSpeed.l,v $
60 // Revision 1.1  2008-06-10 07:58:49-07  Hamilton
61 // Initial revision
62 //
63 // Revision 1.1  2008-05-19 08:45:00-07  Hamilton
64 // Initial revision
65 //
66 // Revision 1.10 2007-02-08 06:45:52-08  Hamilton
67 // Normal end of day data saving
68 //
69

```

1.13 IdIt

IdIt.L: This filter breaks down the capture data for common data types. There is no filter available for some of the data capture file types.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file Hh.bat (Section 1.11, page 28) shows use of this command.
6. **Listing of: hh03.dat a typical input file.**

```

1
2
3      1,      1: DCE      1      0.000000  0.000000 88 0b
4      2,      3: DCE      4      0.019792  0.001039 88 0b
5      2,      2: DTE      6      0.023860  0.003027 02
6      3,      5: DCE      7      0.032291  0.008431 84 06
7      3,      7: DCE      9      0.060317  0.026990 88 0b
8      3,      3: DTE     11      0.064578  0.001037 02
9      4,      9: DCE     12      0.076461  0.001037 87 0f
10     4,     11: DCE     14      0.109694  0.001037 88 0b
11     4,      4: DTE     16      0.113939  0.003208 02
12     5,     13: DCE     17      0.124899  0.010960 89 01
13     5,     15: DCE     19      0.157815  0.031877 89 06
14     5,     17: DCE     21      0.190426  0.031571 88 0b
15     5,      5: DTE     23      0.194753  0.001039 02
16     6,     19: DCE     24      0.205736  0.001039 8b 03
17     6,     21: DCE     26      0.237195  0.001039 8b 05
18     6,     23: DCE     28      0.269903  0.001039 88 0b
19     6,      6: DTE     30      0.274109  0.003164 02
20     7,     25: DCE     31      0.285111  0.011002 8d 07

```

7. **Listing of: hh03.pol a typical output file.**

```

1      1,      1: DCE      1      0.000000  0.000000 88 0b      : 139 Status Poll Request
2      2,      3: DCE      4      0.019792  0.001039 88 0b      : 139 Status Poll Request
3      3,      5: DCE      7      0.032291  0.008431 84 06      : 70 Status Poll Request
4      3,      7: DCE      9      0.060317  0.026990 88 0b      : 139 Status Poll Request
5      4,      9: DCE     12      0.076461  0.001037 87 0f      : 127 Status Poll Request
6      4,     11: DCE     14      0.109694  0.001037 88 0b      : 139 Status Poll Request
7      5,     13: DCE     17      0.124899  0.010960 89 01      : 145 Status Poll Request
8      5,     15: DCE     19      0.157815  0.031877 89 06      : 150 Status Poll Request
9      5,     17: DCE     21      0.190426  0.031571 88 0b      : 139 Status Poll Request
10     6,     19: DCE     24      0.205736  0.001039 8b 03      : 179 Status Poll Request
11     6,     21: DCE     26      0.237195  0.001039 8b 05      : 181 Status Poll Request
12     6,     23: DCE     28      0.269903  0.001039 88 0b      : 139 Status Poll Request
13     7,     25: DCE     31      0.285111  0.011002 8d 07      : 215 Status Poll Request
14     7,     27: DCE     33      0.310111  0.023957 88 0b      : 139 Status Poll Request
15     8,     29: DCE     36      0.324076  0.001042 8e 00      : 224 Status Poll Request
16     8,     31: DCE     38      0.361368  0.001042 88 0b      : 139 Status Poll Request
17     9,     33: DCE     41      0.381263  0.015716 88 0b      : 139 Status Poll Request
18    10,     35: DCE     44      0.399285  0.001042 88 0b      : 139 Status Poll Request
19    11,     37: DCE     47      0.417409  0.013952 88 0b      : 139 Status Poll Request
20    12,     39: DCE     50      0.437513  0.001042 88 0b      : 139 Status Poll Request

```


1.13.1 Listing of IdIt.l

```

1  %{
2  //
3  //
4  // 3901, 11367: DCE 311.720881 0.002031 80 51 20 00 00 00 00 50 00 00
5  // 3903, 11377: DCE 311.770673 0.002084 80 51 20 00 00 00 00 50 00 00
6  // 3905, 11387: DCE 311.820672 0.002091 80 51 20 00 00 00 00 50 00 00
7  //123456789 123456789 123456789 123456789 123456789 123456789
8  //
9  // $Header: d:/Binder4/sda2/RCS/IdIt.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $
10 // IdIt is used to get the device ID and breakdown standard commands. It
11 // only pays attention to DCE type data. I.e. All DTE data is thrown
12 // away.
13 //
14 // Decoding data comes from the Vicon publication "RS-422 Receiver
15 // Communications Protocol" Part Number 8015-5812-02-00 Rev 404 with
16 // copyright dates of 1994, 2003.
17 //
18 #include <stdio.h>
19 #include <string.h>
20 #include <stdlib.h>
21 #undef yywrap
22
23 // #define EXTRA // For debug information
24
25 #define FACTOR 8 // Scaling factor to use with speeds
26 int Speed;
27
28 char header[10000];
29 char CommandType[10000];
30
31 unsigned int MakeAdd(void); // Converts Vicon format addresses into decimal
32 unsigned int MakeHex(void); // Converts Vicon format speeds into decimal
33
34 void Command1(void);
35 void Command2(void);
36 void Command3(void);
37 void Command4(void);
38 void Command5(void);
39 void Command6(void);
40 void Command7(void);
41 void Command8(void);
42
43 %}
44 %s COMMAND1 COMMAND2 COMMAND3 COMMAND4 COMMAND5 COMMAND6 COMMAND7 COMMAND8
45 %%
46 ^.*"DCE".*8"[0-9a-f]" 0"[0-9a-f]$" fprintf(yyout,"%s : %3u Status Poll
Request\n",yytext,MakeAdd());
47
48 ^.*"DCE".*8"[0-9a-f]" "[1-7][0-9a-f]" strcpy(header,yytext); BEGIN COMMAND1;
49 <COMMAND1>" "[0-7][0-9a-f]" Command1(); BEGIN COMMAND2;
50 <COMMAND2>" "[0-7][0-9a-f]" Command2(); BEGIN COMMAND3;
51 <COMMAND3>" "[0-7][0-9a-f]" Command3(); BEGIN COMMAND4;
52 <COMMAND4>" "[0-7][0-9a-f]" Command4(); BEGIN COMMAND5;
53 <COMMAND5>" "[0-7][0-9a-f]" Command5(); BEGIN COMMAND6;
54 <COMMAND6>" "[0-7][0-9a-f]" Command6(); BEGIN COMMAND7;
55 <COMMAND7>" "[0-7][0-9a-f]" Command7(); BEGIN COMMAND8;
56 <COMMAND8>" "[0-7][0-9a-f]" Command8();
57 <COMMAND8>\n fprintf(yyout,"%s: %s\n",header,CommandType); BEGIN 0;
58
59 ^.*"DTE".*\n ;
60
61 .|\n ;
62 %%
63
64
65 //
66 // Processes bits for pan, tilt and auto pan/iris. (Table 2)
67 // Sometimes bit 0 is turned on. However it is an undefined bit.

```

```

68 //
69 void Command1(void)
70 {
71     int temp;
72     char string[10];
73
74     CommandType[0] = '\0';
75     strcat(header,yytext);
76     temp = MakeHex();
77 #ifdef EXTRA
78     sprintf(string,"%c%c ",yytext[1],yytext[2]);
79     strcat(CommandType,string);
80 #endif // EXTRA
81
82
83     if (temp & 0x40) strcat(CommandType," Pan Left, ");
84     if (temp & 0x20) strcat(CommandType,"Pan Right, ");
85     if (temp & 0x10) strcat(CommandType," Tilt Up, ");
86
87     if (temp & 0x08) strcat(CommandType,"Tilt Down, ");
88     if (temp & 0x04) strcat(CommandType,"Auto Pan, ");
89     if (temp & 0x02) strcat(CommandType,"Auto Iris, ");
90 //     if (temp & 0x01) strcat(CommandType,"undefined 1-0, ");
91
92 //     strcat(CommandType,"command1: ");
93 }
94
95
96 //
97 // Processes zoom, focus, iris and lens speed bits. (Table 2)
98 //
99 void Command2(void)
100 {
101     int temp;
102     char string[10];
103
104     strcat(header,yytext);
105     temp = MakeHex();
106 #ifdef EXTRA
107     sprintf(string,"%c%c ",yytext[1],yytext[2]);
108     strcat(CommandType,string);
109 #endif // EXTRA
110
111
112     if (temp & 0x40) strcat(CommandType,"Zoom Out, ");
113     if (temp & 0x20) strcat(CommandType,"Zoom In, ");
114     if (temp & 0x10) strcat(CommandType,"Focus Far, ");
115
116     if (temp & 0x08) strcat(CommandType,"Focus Near, ");
117     if (temp & 0x04) strcat(CommandType,"Iris Open, ");
118     if (temp & 0x02) strcat(CommandType,"Iris Close, ");
119     if (temp & 0x01) strcat(CommandType,"Lens Speed, ");
120
121 //     strcat(CommandType,"command2: ");
122 }
123
124
125 //
126 // Process Aux bits. (Table 2)
127 // Sometimes bit 0 is turned on. However it is an undefined bit.
128 //
129 void Command3(void)
130 {
131     int temp;
132     char string[10];
133
134     strcat(header,yytext);
135     temp = MakeHex();
136 #ifdef EXTRA
137     sprintf(string,"%c%c ",yytext[1],yytext[2]);
138     strcat(CommandType,string);
139 #endif // EXTRA

```

```

140
141
142     if (temp & 0x40) strcat(CommandType,"AUX 1, ");
143     if (temp & 0x20) strcat(CommandType,"AUX 2, ");
144     if (temp & 0x10) strcat(CommandType,"AUX 3, ");
145
146     if (temp & 0x08) strcat(CommandType,"AUX 4, ");
147     if (temp & 0x04) strcat(CommandType,"AUX 5, ");
148     if (temp & 0x02) strcat(CommandType,"AUX 6, ");
149 //     if (temp & 0x01) strcat(CommandType,"undefined 3-0, ");
150
151 //     strcat(CommandType,"command3: ");
152 }
153
154
155 //
156 // Process preset data. (Table 2)
157 //
158 void Command4(void)
159 {
160     int temp;
161     char tempchar[10];
162     char string[10];
163
164     strcat(header,yytext);
165     temp = MakeHex();
166 #ifdef EXTRA
167     sprintf(string,"%c%c ",yytext[1],yytext[2]);
168     strcat(CommandType,string);
169 #endif // EXTRA
170
171
172     if (temp & 0x40) strcat(CommandType,"Preset Enter, ");
173     if (temp & 0x20) strcat(CommandType,"Preset Recall, ");
174 //     if (temp & 0x10) strcat(CommandType,"undefined 4-4, ");
175
176     if ((temp & 0x0F) != 0)
177     {
178         sprintf(tempchar," %d, ",(temp & 0x0F));
179         strcat(CommandType,tempchar);
180     }
181
182 //     strcat(CommandType,"command4: ");
183 }
184
185
186 //
187 // Process upper bits of pan speed. (Table 3)
188 //
189 void Command5(void)
190 {
191     int temp;
192     char string[10];
193
194     strcat(header,yytext);
195     temp = MakeHex();
196     Speed = temp & 0x0F;
197
198 #ifdef EXTRA
199     sprintf(string,"%c%c ",yytext[1],yytext[2]);
200     strcat(CommandType,string);
201 #endif // EXTRA
202 //     strcat(CommandType,"command5: ");
203
204 }
205
206
207 //
208 // Process the rest of the pan speed command. (Table 3)
209 //
210 void Command6(void)
211 {

```

```

212 int temp;
213 char string[100];
214
215     strcat(header,yytext);
216     temp = MakeHex();
217 #ifdef EXTRA
218     sprintf(string,"%c%c ",yytext[1],yytext[2]);
219     strcat(CommandType,string);
220 #endif // EXTRA
221
222     Speed *= 128;
223     temp &= 0x7F;
224     if ((Speed != 0) || (temp != 0))
225     {
226         strcat(CommandType,"Pan Speed ");
227         Speed += temp;
228         temp = Speed / FACTOR;
229         sprintf(string," %4d, ",temp);
230         strcat(CommandType,string);
231     }
232     //     strcat(CommandType,"command6: ");
233 }
234
235 //
236 // Process the upper bits of the tilt speed. (Table 3)
237 //
238 void Command7(void)
239 {
240     int temp;
241     char string[10];
242
243     strcat(header,yytext);
244     temp = MakeHex();
245     Speed = temp & 0x0F;
246
247 #ifdef EXTRA
248     sprintf(string,"%c%c ",yytext[1],yytext[2]);
249     strcat(CommandType,string);
250 #endif // EXTRA
251
252 //     strcat(CommandType,"command7: ");
253 }
254
255 //
256 // Process the rest of the tilt speed value. (Table 3)
257 //
258 void Command8(void)
259 {
260     int temp;
261     char string[100];
262
263     strcat(header,yytext);
264     temp = MakeHex();
265 #ifdef EXTRA
266     sprintf(string,"%c%c ",yytext[1],yytext[2]);
267     strcat(CommandType,string);
268 #endif // EXTRA
269
270     Speed *= 128;
271     temp &= 0x7F;
272     if ((Speed != 0) || (temp != 0))
273     {
274         strcat(CommandType,"Tilt Speed ");
275         Speed += temp;
276         temp = Speed / FACTOR;
277         sprintf(string," %4d, ",temp);
278         strcat(CommandType,string);
279     }
280 //     strcat(CommandType,"command8: ");
281 }

```

```

284
285
286 //
287 // Converts from Vicon address format to decimal. (Table 1)
288 //
289 // Vicon address format is in the lower bits of two bytes. The first
290 // byte's lower four bits have the upper half of the speed and the second
291 // half's lower four bits have the rest of the speed.
292 //
293 // Input is: "<bunch of junk followed by>xx xx" in ASCII
294 //           with each x = to one hex digit, i.e. a nibble.
295 //
296 unsigned int MakeAdd(void)
297 {
298     int temp[2];
299     int i;
300
301     // Find the end of the string
302     i = strlen(yytext);
303
304     // Convert from ASCII 0 --> 9 (0x30 --> 0x39) to decimal 0 --> 9
305     //           and from ASCII a --> f (0x61 --> 0x65) to decimal 10 --> 15
306     // Note lower case letters for larger hex values
307     temp[0] = yytext[i-4] & 0x0F;
308     if ((yytext[i-4] & 0x70) == 0x60) temp[0] += 9;
309     temp[0] = temp[0] << 4;
310
311     temp[1] = yytext[i-1] & 0x0F;
312     if ((yytext[i-1] & 0x70) == 0x60) temp[1] += 9;
313     temp[0] += temp[1];
314
315     return temp[0];
316 }
317
318 //
319 //
320 // Converts from two ASCII hex digits to decimal.
321 //
322 // Input is: "xx" in ASCII with each x = to one hex digit, i.e. a nibble.
323 //
324 unsigned int MakeHex(void)
325 {
326     int temp[2];
327
328     // Convert from ASCII 0 --> 9 (0x30 --> 0x39) to decimal 0 --> 9
329     //           and from ASCII a --> f (0x61 --> 0x65) to decimal 10 --> 15
330     // Note lower case letters for larger hex values
331     temp[0] = yytext[1] & 0x0F;
332     if ((yytext[1] & 0x70) == 0x60) temp[0] += 9;
333     temp[0] = temp[0] << 4;
334
335     temp[1] = yytext[2] & 0x0F;
336     if ((yytext[2] & 0x70) == 0x60) temp[1] += 9;
337     temp[0] += temp[1];
338
339     return temp[0];
340 }
341
342
343
344 // yywrap is for End Of File processing, 1 = done
345 int yywrap(void)
346 {
347     return 1;
348 }
349
350 main(int argc, char *argv[])
351 {
352     fprintf(stderr, "$Id: IdIt.1,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $\n");
353
354     // Do we have to deal with stdin/stdout?
355     // 3 = one for the program name and one each for input and output names

```

```

356     if (argc == 3) {
357         // Nope
358         // Get over the program, etc., name
359         argc--;
360         argv++;
361         yyin = fopen(argv[0], "r");
362         if (yyin == NULL) {
363             fprintf(stderr, "\aCould not open \"%s\" for reading, quitting",
364                     argv[0]);
365             exit(EXIT_FAILURE);
366         }
367
368         // Now get over the input file name and get to the output file name
369         argc--;
370         argv++;
371         yyout = fopen(argv[0], "w");
372         if (yyout == NULL) {
373             fprintf(stderr, "\aCould not open \"%s\" for writing, quitting",
374                     argv[0]);
375             exit(EXIT_FAILURE);
376         }
377     }
378     else {
379         // Woops we are using std... type io
380         yyin = stdin;
381         yyout = stdout;
382     }
383
384     yylex();
385     exit(EXIT_SUCCESS);
386 }
387
388 // $Log: IdIt.l,v $
389 // Revision 1.1  2008-06-10 07:58:49-07  Hamilton
390 // Initial revision
391 //
392 // Revision 1.6  2008-05-21 09:06:35-07  Hamilton
393 // Normal end of day data saving
394 //
395 // Revision 1.5  2008-05-20 13:11:13-07  Hamilton
396 // Normal end of day data saving
397 //
398 // Revision 1.4  2008-05-19 12:15:53-07  Hamilton
399 // Normal end of day data saving
400 //
401 // Revision 1.3  2008-05-19 08:27:52-07  Hamilton
402 // Normal end of day data saving
403 //
404 // Revision 1.2  2008-05-14 09:39:45-07  Hamilton
405 // Normal end of day data saving
406 //
407 // Revision 1.1  2008-05-14 06:55:16-07  Hamilton
408 // Initial revision
409 //
410 // Revision 1.2  2008-05-13 11:07:19-07  Hamilton
411 // Normal end of day data saving
412 //
413 // Revision 1.1  2008-05-13 10:32:43-07  Hamilton
414 // Initial revision
415 //
416 // Revision 1.1  2008-05-13 08:05:15-07  Hamilton
417 // Initial revision
418 //
419 // Revision 1.10 2007-02-08 06:45:52-08  Hamilton
420 // Normal end of day data saving
421 //
422

```

1.14 IdStatus

IdSTATUS.L: This filter identifies status commands.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file Hh.bat (Section 1.11, page 28) shows use of this command.
6. Listing of: hh03.pol a typical input file.

1	1,	1: DCE	1	0.000000	0.000000 88 0b	: 139 Status Poll Request
2	2,	3: DCE	4	0.019792	0.001039 88 0b	: 139 Status Poll Request
3	3,	5: DCE	7	0.032291	0.008431 84 06	: 70 Status Poll Request
4	3,	7: DCE	9	0.060317	0.026990 88 0b	: 139 Status Poll Request
5	4,	9: DCE	12	0.076461	0.001037 87 0f	: 127 Status Poll Request
6	4,	11: DCE	14	0.109694	0.001037 88 0b	: 139 Status Poll Request
7	5,	13: DCE	17	0.124899	0.010960 89 01	: 145 Status Poll Request
8	5,	15: DCE	19	0.157815	0.031877 89 06	: 150 Status Poll Request
9	5,	17: DCE	21	0.190426	0.031571 88 0b	: 139 Status Poll Request
10	6,	19: DCE	24	0.205736	0.001039 8b 03	: 179 Status Poll Request
11	6,	21: DCE	26	0.237195	0.001039 8b 05	: 181 Status Poll Request
12	6,	23: DCE	28	0.269903	0.001039 88 0b	: 139 Status Poll Request
13	7,	25: DCE	31	0.285111	0.011002 8d 07	: 215 Status Poll Request
14	7,	27: DCE	33	0.310111	0.023957 88 0b	: 139 Status Poll Request
15	8,	29: DCE	36	0.324076	0.001042 8e 00	: 224 Status Poll Request
16	8,	31: DCE	38	0.361368	0.001042 88 0b	: 139 Status Poll Request
17	9,	33: DCE	41	0.381263	0.015716 88 0b	: 139 Status Poll Request
18	10,	35: DCE	44	0.399285	0.001042 88 0b	: 139 Status Poll Request
19	11,	37: DCE	47	0.417409	0.013952 88 0b	: 139 Status Poll Request
20	12,	39: DCE	50	0.437513	0.001042 88 0b	: 139 Status Poll Request

7. Listing of: hh03.sts a typical output file.

1						
2						
3	2,	2: DTE	6	0.023860	0.003027 02:	Auto Iris Status
4	3,	3: DTE	11	0.064578	0.001037 02:	Auto Iris Status
5	4,	4: DTE	16	0.113939	0.003208 02:	Auto Iris Status
6	5,	5: DTE	23	0.194753	0.001039 02:	Auto Iris Status
7	6,	6: DTE	30	0.274109	0.003164 02:	Auto Iris Status
8	7,	7: DTE	35	0.314414	0.001042 02:	Auto Iris Status
9	8,	8: DTE	40	0.365547	0.003138 02:	Auto Iris Status
10	9,	9: DTE	43	0.385228	0.001042 02:	Auto Iris Status
11	10,	10: DTE	46	0.403457	0.003131 02:	Auto Iris Status
12	11,	11: DTE	49	0.421472	0.001042 02:	Auto Iris Status
13	12,	12: DTE	52	0.441675	0.003120 02:	Auto Iris Status
14	13,	13: DTE	57	0.484584	0.001041 02:	Auto Iris Status
15	14,	14: DTE	64	0.565189	0.003190 02:	Auto Iris Status
16	15,	15: DTE	67	0.585079	0.001041 02:	Auto Iris Status
17	16,	16: DTE	74	0.665892	0.003261 02:	Auto Iris Status
18	17,	17: DTE	77	0.685788	0.001146 02:	Auto Iris Status
19	18,	18: DTE	80	0.705469	0.002942 02:	Auto Iris Status
20	19,	19: DTE	83	0.724212	0.001042 02:	Auto Iris Status

1.14.1 Listing of IdStatus.l

```

1  %{
2  //
3  // $Header: d:/Binder4/sda2/RCS/IdStatus.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $
4  // IdStatus is used to get the device status from camera responses.
5  //
6  #include <stdio.h>
7  #include <stdlib.h>
8  #undef yywrap
9
10 unsigned int MakeAdd(void);
11
12 %}
13 %%
14 ^.*"DTE".*" 41 "[0-9a-f][0-9a-f]$ fprintf(yyout,"%s:      Status Response with ID",yytext);
15 ^.*"DTE".*" 02"$      fprintf(yyout,"%s:      Auto Iris Status",yytext);
16 ^.*"DCE".*\n          ;
17
18 \n      ECHO;
19 %%
20
21 unsigned int MakeAdd(void)
22 {
23     int temp[2];
24     int i;
25
26
27     i = strlen(yytext);
28
29     temp[0] = yytext[i-4] & 0x0F;
30     if ((yytext[i-4] & 0x70) == 0x60) temp[0] += 9;
31     temp[0] = temp[0] << 4;
32
33     temp[1] = yytext[i-1] & 0x0F;
34     if ((yytext[i-1] & 0x70) == 0x60) temp[1] += 9;
35
36     temp[0] += temp[1];
37     return temp[0];
38 }
39
40
41 // yywrap is for End Of File processing, 1 = done
42 int yywrap(void)
43 {
44     return 1;
45 }
46
47 main(int argc, char *argv[])
48 {
49     fprintf(stderr,"$Id: IdStatus.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $\n");
50
51     // Do we have to deal with stdin/stdout?
52     // 3 = one for the program name and one each for input and output names
53     if (argc == 3) {
54         // Nope
55         // Get over the program, etc., name
56         argc--;
57         argv++;
58         yyin = fopen(argv[0],"r");
59         if (yyin == NULL) {
60             fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
61                     argv[0]);
62             exit(EXIT_FAILURE);
63         }
64
65         // Now get over the input file name and get to the output file name
66         argc--;
67         argv++;
68         yyout = fopen(argv[0],"w");

```



```
69         if (yyout == NULL) {
70             fprintf(stderr, "\aCould not open \"%s\" for writing, quitting",
71                     argv[0]);
72             exit(EXIT_FAILURE);
73         }
74     }
75     else {
76         // Woops we are using std... type io
77         yyin = stdin;
78         yyout = stdout;
79     }
80
81     yylex();
82     exit(EXIT_SUCCESS);
83 }
84
85 // $Log: IdStatus.l,v $
86 // Revision 1.1  2008-06-10 07:58:49-07  Hamilton
87 // Initial revision
88 //
89 // Revision 1.4  2008-05-22 09:43:08-07  Hamilton
90 // Normal end of day data saving
91 //
92 // Revision 1.3  2008-05-14 06:54:56-07  Hamilton
93 // Normal end of day data saving
94 //
95 // Revision 1.2  2008-05-13 11:07:20-07  Hamilton
96 // Normal end of day data saving
97 //
98 // Revision 1.1  2008-05-13 10:32:50-07  Hamilton
99 // Initial revision
100 //
101 // Revision 1.1  2008-05-13 08:05:15-07  Hamilton
102 // Initial revision
103 //
104 // Revision 1.10 2007-02-08 06:45:52-08  Hamilton
105 // Normal end of day data saving
106 //
107
```

1.15 JustSped

JUSTSPED.L: This filter extracts just the pan speeds from an interpreted capture file.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file Hh.bat (Section 1.11, page 28) shows use of this command.
6. **Listing of: hh03.spd a typical input file.**

```

1 2562, 5725: DCE 8288 57.663164 0.039897 88 5b 00 00 00 00 08 00 00: Pan Speed 1,
2 2564, 5741: DCE 8307 57.783210 0.041186 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
3 2569, 5763: DCE 8335 57.964427 0.041876 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
4 2602, 5899: DCE 8512 59.316158 0.040000 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
5 2654, 6057: DCE 8728 60.838736 0.040424 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
6 2656, 6073: DCE 8747 60.958639 0.039695 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
7 2658, 6089: DCE 8766 61.078954 0.040167 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
8 2661, 6107: DCE 8788 61.219480 0.041773 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
9 2667, 6141: DCE 8830 61.519394 0.041732 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
10 2675, 6175: DCE 8874 61.830343 0.039894 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
11 2695, 6233: DCE 8954 62.389786 0.038374 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
12 2701, 6257: DCE 8985 62.601841 0.040424 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
13 2773, 6471: DCE 9281 64.786216 0.036667 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
14 2776, 6489: DCE 9303 64.926952 0.041564 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
15 2789, 6547: DCE 9377 65.466354 0.042188 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
16 2808, 6615: DCE 9468 66.149719 0.039897 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
17 2836, 6707: DCE 9593 67.070289 0.038026 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
18 2891, 6873: DCE 9821 68.756307 0.041258 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
19 2893, 6889: DCE 9840 68.874849 0.040312 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
20 2895, 6905: DCE 9859 68.996522 0.041667 88 5b 21 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
```

7. **Listing of: hh03.plt a typical output file.**

```

1 # $Header: d:/Binder4/sda2/RCS/Hh03.plt,v 1.1 2008-06-10 09:25:19-07 Hamilton Exp Hamilton $
2 # From h03.spd
3 1
4 1
5 1
6 1
7 1
8 1
9 1
10 1
11 1
12 1
13 1
14 1
15 1
16 1
17 1
18 1
19 1
20 1
```

1.15.1 Listing of JustSped.l

```

1  %{
2  // 1969, 10891: DCE 91.038559 0.041470 88 5b 21 00 00 00 0f 78 00 00: Pan Right, Pan Speed 255,
3  // 1971, 10907: DCE 91.157937 0.041151 88 5b 30 00 00 00 06 40 01 48: Pan Right, Tilt Up, Pan Speed 104,
Tilt Speed 25,
4  //
5  // $Header: d:/Binder4/sda2/RCS/JustSped.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $
6  // JustSped is used to extract only pan speeds from a .SPD file.
7  //
8  #include <stdio.h>
9  #include <stdlib.h>
10 #undef yywrap
11 %}
12 %%
13 ^.*"Pan Speed" ;
14 "Tilt Speed".*$ ;
15
16 ", " ;
17 .|\n ECHO;
18 %%
19 // yywrap is for End Of File processing, 1 = done
20 int yywrap(void)
21 {
22     return 1;
23 }
24
25 main(int argc, char *argv[])
26 {
27     char fromfile[1000];
28
29     fprintf(stderr,"$Id: JustSped.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $\n");
30
31     // Do we have to deal with stdin/stdout?
32     // 3 = one for the program name and one each for input and output names
33     if (argc == 3) {
34         // Nope
35         // Get over the program, etc., name
36         argc--;
37         argv++;
38         strcpy(fromfile,argv[0]);
39         yyin = fopen(argv[0],"r");
40         if (yyin == NULL) {
41             fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
42                     argv[0]);
43             exit(EXIT_FAILURE);
44         }
45
46         // Now get over the input file name and get to the output file name
47         argc--;
48         argv++;
49         yyout = fopen(argv[0],"w");
50         if (yyout == NULL) {
51             fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
52                     argv[0]);
53             exit(EXIT_FAILURE);
54         }
55     }
56     else {
57         // Woops we are using std... type io
58         yyin = stdin;
59         yyout = stdout;
60     }
61
62     fprintf(yyout,"# $");
63     fprintf(yyout,"Header$\n");
64     fprintf(yyout,"# From %s\n",fromfile);
65     yylex();
66     exit(EXIT_SUCCESS);
67 }

```

```
68
69 // $Log: JustSped.1,v $
70 // Revision 1.1  2008-06-10 07:58:49-07  Hamilton
71 // Initial revision
72 //
73 // Revision 1.2  2008-05-21 09:38:26-07  Hamilton
74 // Normal end of day data saving
75 //
76 // Revision 1.1  2008-05-21 08:42:43-07  Hamilton
77 // Initial revision
78 //
79 // Revision 1.10  2007-02-08 06:45:52-08  Hamilton
80 // Normal end of day data saving
81 //
82
```

1.16 NoStatus

NoSTATUS.L: This filter dumps all status commands.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon's Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file Hh.bat (Section 1.11, page 28) shows use of this command.
6. Listing of: hh03.srt a typical input file.

```

1
2
3      1,      1: DCE      1      0.000000  0.000000 88 0b      : 139 Status Poll Request
4      2,      3: DCE      4      0.019792  0.001039 88 0b      : 139 Status Poll Request
5      2,      2: DTE      6      0.023860  0.003027 02:      Auto Iris Status
6      3,      5: DCE      7      0.032291  0.008431 84 06      : 70 Status Poll Request
7      3,      7: DCE      9      0.060317  0.026990 88 0b      : 139 Status Poll Request
8      3,      3: DTE     11      0.064578  0.001037 02:      Auto Iris Status
9      4,      9: DCE     12      0.076461  0.001037 87 0f      : 127 Status Poll Request
10     4,     11: DCE     14      0.109694  0.001037 88 0b      : 139 Status Poll Request
11     4,      4: DTE     16      0.113939  0.003208 02:      Auto Iris Status
12     5,     13: DCE     17      0.124899  0.010960 89 01      : 145 Status Poll Request
13     5,     15: DCE     19      0.157815  0.031877 89 06      : 150 Status Poll Request
14     5,     17: DCE     21      0.190426  0.031571 88 0b      : 139 Status Poll Request
15     5,      5: DTE     23      0.194753  0.001039 02:      Auto Iris Status
16     6,     19: DCE     24      0.205736  0.001039 8b 03      : 179 Status Poll Request
17     6,     21: DCE     26      0.237195  0.001039 8b 05      : 181 Status Poll Request
18     6,     23: DCE     28      0.269903  0.001039 88 0b      : 139 Status Poll Request
19     6,      6: DTE     30      0.274109  0.003164 02:      Auto Iris Status
20     7,     25: DCE     31      0.285111  0.011002 8d 07      : 215 Status Poll Request

```

7. Listing of: hh03.nos a typical output file.

```

1
2
3 2558, 2558: DTE 8260 57.477922 0.003307 41
4 2559, 5707: DCE 8266 57.524199 0.041667 88 5b 00 00 00 00 00 00 00:
5 2561, 2562: DTE 8282 57.618724 0.001144 41
6 2562, 5725: DCE 8288 57.663164 0.039897 88 5b 00 00 00 00 00 08 00 00: Pan Speed 1,
7 2563, 2565: DTE 8301 57.737447 0.003555 41
8 2564, 5741: DCE 8307 57.783210 0.041186 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
9 2568, 2571: DTE 8329 57.918025 0.001042 41
10 2569, 5763: DCE 8335 57.964427 0.041876 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
11 2575, 2579: DTE 8363 58.139328 0.001041 41
12 2579, 2584: DTE 8382 58.291587 0.003287 41
13 2583, 2589: DTE 8405 58.468318 0.003345 41
14 2588, 2595: DTE 8431 58.669726 0.001039 41
15 2590, 2598: DTE 8448 58.810006 0.003561 41
16 2595, 2604: DTE 8470 58.981215 0.000873 41
17 2598, 2608: DTE 8486 59.115044 0.001043 41
18 2601, 2612: DTE 8506 59.271568 0.001040 41
19 2602, 5899: DCE 8512 59.316158 0.040000 88 5b 20 00 00 00 00 08 00 00: Pan Right, Pan Speed 1,
20 2618, 2630: DTE 8570 59.681575 0.001090 41

```

1.16.1 Listing of NoStatus.l

```

1  %{
2  // $Header: d:/Binder4/sda2/RCS/NoStatus.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $
3  // LMAIN is used as a "universal" starter skeleton for FLEX programs
4  #include <stdio.h>
5  #include <stdlib.h>
6  #undef yywrap
7  %}
8  %%
9  ^.*"Status".*\n ;
10
11 .|\n      ECHO;
12 %%
13 // yywrap is for End Of File processing, 1 = done
14 int yywrap(void)
15 {
16     return 1;
17 }
18
19 main(int argc, char *argv[])
20 {
21     fprintf(stderr,"$Id: NoStatus.l,v 1.1 2008-06-10 07:58:49-07 Hamilton Exp Hamilton $\n");
22
23     // Do we have to deal with stdin/stdout?
24     // 3 = one for the program name and one each for input and output names
25     if (argc == 3) {
26         // Nope
27         // Get over the program, etc., name
28         argc--;
29         argv++;
30         yyin = fopen(argv[0],"r");
31         if (yyin == NULL) {
32             fprintf(stderr,"\aCould not open \"%s\" for reading, quitting",
33                 argv[0]);
34             exit(EXIT_FAILURE);
35         }
36
37         // Now get over the input file name and get to the output file name
38         argc--;
39         argv++;
40         yyout = fopen(argv[0],"w");
41         if (yyout == NULL) {
42             fprintf(stderr,"\aCould not open \"%s\" for writing, quitting",
43                 argv[0]);
44             exit(EXIT_FAILURE);
45         }
46     }
47     else {
48         // Woops we are using std... type io
49         yyin = stdin;
50         yyout = stdout;
51     }
52
53     yylex();
54     exit(EXIT_SUCCESS);
55 }
56
57 // $Log: NoStatus.l,v $
58 // Revision 1.1  2008-06-10 07:58:49-07  Hamilton
59 // Initial revision
60 //
61 // Revision 1.1  2008-05-14 08:51:35-07  Hamilton
62 // Initial revision
63 //
64 // Revision 1.10 2007-02-08 06:45:52-08  Hamilton
65 // Normal end of day data saving
66 //
67

```

1.17 ProcDCE

PROCDC.E.L: This filter builds up commands starting with the “first byte”. First bytes always have the most significant bit set.

1. This filter is written in Flex.
2. This filter is suitable for use with Vicon’s Protocols only.
3. This filter requires preprocessing by MAKESECS.
4. This filter is sensitive to having DCE (for data coming from the Keyboard) and DTE (for data coming from the Dome) having data coming from the Headend/Keyboard or the Dome/Camera.
5. Batch file Hh.bat (Section 1.11, page 28) shows use of this command.
6. **Listing of: hh03.dce a typical input file.**

```

1      1,      1: DCE      1      0.000000  0.000000 88
2      1,      2: DCE      2      0.001039  0.001039 0b
3      2,      3: DCE      4      0.019792  0.001039 88
4      2,      4: DCE      5      0.020833  0.001039 0b
5      3,      5: DCE      7      0.032291  0.008431 84
6      3,      6: DCE      8      0.033327  0.001036 06
7      3,      7: DCE      9      0.060317  0.026990 88
8      3,      8: DCE     10      0.061354  0.001037 0b
9      4,      9: DCE     12      0.076461  0.001037 87
10     4,     10: DCE     13      0.077501  0.001037 0f
11     4,     11: DCE     14      0.109694  0.001037 88
12     4,     12: DCE     15      0.110731  0.001037 0b
13     5,     13: DCE     17      0.124899  0.010960 89
14     5,     14: DCE     18      0.125938  0.001039 01
15     5,     15: DCE     19      0.157815  0.031877 89
16     5,     16: DCE     20      0.158855  0.001040 06
17     5,     17: DCE     21      0.190426  0.031571 88
18     5,     18: DCE     22      0.191465  0.001039 0b
19     6,     19: DCE     24      0.205736  0.001039 8b
20     6,     20: DCE     25      0.206778  0.001039 03

```

7. **Listing of: hh03.pc a typical output file.**

```

1
2      1,      1: DCE      1      0.000000  0.000000 88 0b
3      2,      3: DCE      4      0.019792  0.001039 88 0b
4      3,      5: DCE      7      0.032291  0.008431 84 06
5      3,      7: DCE      9      0.060317  0.026990 88 0b
6      4,      9: DCE     12      0.076461  0.001037 87 0f
7      4,     11: DCE     14      0.109694  0.001037 88 0b
8      5,     13: DCE     17      0.124899  0.010960 89 01
9      5,     15: DCE     19      0.157815  0.031877 89 06
10     5,     17: DCE     21      0.190426  0.031571 88 0b
11     6,     19: DCE     24      0.205736  0.001039 8b 03
12     6,     21: DCE     26      0.237195  0.001039 8b 05
13     6,     23: DCE     28      0.269903  0.001039 88 0b
14     7,     25: DCE     31      0.285111  0.011002 8d 07
15     7,     27: DCE     33      0.310111  0.023957 88 0b
16     8,     29: DCE     36      0.324076  0.001042 8e 00
17     8,     31: DCE     38      0.361368  0.001042 88 0b
18     9,     33: DCE     41      0.381263  0.015716 88 0b
19     10,    35: DCE     44      0.399285  0.001042 88 0b
20     11,    37: DCE     47      0.417409  0.013952 88 0b

```

1.17.1 Listing of ProcDCE.l

```

1  %{
2  //      1,      1: DCE      0.000000  0.000000 80
3  //      1,      2: DCE      0.002083  0.002083 01
4  //      2,      3: DCE      0.022915  0.002083 80
5  //      2,      4: DCE      0.024994  0.002083 01
6  //123456789 123456789 123456789 123456789 123456789
7  //
8  // $Header: d:/Binder4/sda2/RCS/ProcDCE.l,v 1.1 2008-06-10 07:58:50-07 Hamilton Exp Hamilton $
9  // ProcDce is used to process Vicon data captures that have been first
10 //      processed by MakeSecs and then by GetDce. The result will be
11 //      first bytes of commands being followed by the rest of the
12 //      command on each line.
13 //
14 #include <stdio.h>
15 #include <stdlib.h>
16 #undef yywrap
17 int i;
18 %{
19 %%
20 .*"8"[0-9a-f]$      fprintf(yyout,"\\n"); ECHO;
21 .*[0-7][0-9a-f]$   i = strlen(yytext); fprintf(yyout," %c%c",yytext[i-2],yytext[i-1]);
22
23 \\n                ;
24
25 %%
26 // yywrap is for End Of File processing, 1 = done
27 int yywrap(void)
28 {
29     return 1;
30 }
31
32 main(int argc, char *argv[])
33 {
34     fprintf(stderr,"$Id: ProcDCE.l,v 1.1 2008-06-10 07:58:50-07 Hamilton Exp Hamilton $\\n");
35
36     // Do we have to deal with stdin/stdout?
37     // 3 = one for the program name and one each for input and output names
38     if (argc == 3) {
39         // Nope
40         // Get over the program, etc., name
41         argc--;
42         argv++;
43         yyin = fopen(argv[0],"r");
44         if (yyin == NULL) {
45             fprintf(stderr,"\\aCould not open \"%s\\\" for reading, quitting",
46                 argv[0]);
47             exit(EXIT_FAILURE);
48         }
49
50         // Now get over the input file name and get to the output file name
51         argc--;
52         argv++;
53         yyout = fopen(argv[0],"w");
54         if (yyout == NULL) {
55             fprintf(stderr,"\\aCould not open \"%s\\\" for writing, quitting",
56                 argv[0]);
57             exit(EXIT_FAILURE);
58         }
59     }
60     else {
61         // Woops we are using std... type io
62         yyin = stdin;
63         yyout = stdout;
64     }
65
66     yylex();
67     exit(EXIT_SUCCESS);
68 }

```



```
69
70 // $Log: ProcDCE.1,v $
71 // Revision 1.1  2008-06-10 07:58:50-07  Hamilton
72 // Initial revision
73 //
74 // Revision 1.3  2008-05-22 09:43:09-07  Hamilton
75 // Normal end of day data saving
76 //
77 // Revision 1.2  2008-05-13 08:05:08-07  Hamilton
78 // Normal end of day data saving
79 //
80 // Revision 1.1  2008-05-13 06:48:42-07  Hamilton
81 // Initial revision
82 //
83 // Revision 1.10 2007-02-08 06:45:52-08  Hamilton
84 // Normal end of day data saving
85 //
86
```

1.18 Other files

1. PLOT.GP: This is the control file for GNU PLOT when it is generating plot data.

1.18.1 Listing of hplot.gp

```
1 # $Header: d:/Binder4/sda2/RCS/HPlot.gp,v 1.1 2008-06-10 09:25:20-07 Hamilton Exp Hamilton $
2 set grid
3 set term latex
4
5 set output "H03.inc"
6 plot "H03.plt" with lines
7
8 set output "H04.inc"
9 plot "H04.plt" with lines
10
11 set output "H05.inc"
12 plot "H05.plt" with lines
13
14 set output "H06.inc"
15 plot "H06.plt" with lines
16
17 set output "H06A.inc"
18 plot "H06A.plt" with lines title "H06A lines 500 -- 1500 from H06"
19
20 set output "H06B.inc"
```

Index

Aurora 99, 3, 4

AVicon1, 5

AVicon2, 8, 11

AVicon3, 11

AVicon4, 14

DCE, 5, 8, 11, 14, 19, 22, 25, 30, 32, 39, 42, 45, 47

DTE, 5, 8, 11, 14, 19, 22, 25, 30, 32, 39, 42, 45, 47

GetSpeed.l, 30

gnuplot, 28, 50

Hh.bat, 28

IdIt.l, 32

IdStatus.l, 39

JustSped.l, 42

MakeSecs, 5, 8, 11, 14, 30, 32, 39, 42, 45, 47

MoldFmt, 19, 22, 25

NoStatus.l, 45

Plot.gp, 28, 50

Plotum.bat, 28

ProcDCE.l, 47

Vicon, 3, 5, 8, 11, 14, 17–19, 22, 25, 30, 32, 39, 42,
45, 47

Vicon01, 19

Vicon01x, 22, 25

Vicon02, 25