Spectra Command/Response Timings

D Protocol

Part 2

8 July 2008

Eric Hamilton

Contents

A Results from Data Captured Response Times **3**
A.1 Results for run `s3pp1` . . . 3
A.2 Results for run `s3pp2` . . . 6
A.3 Results for run `s3steps` . . . 9
A.4 Results for run `s3steps1` . . . 12
A.5 Results for run `s4pp1` . . . 15
A.6 Results for run `s4steps` . . . 18
A.7 Results for run `epp1` . . . 21
A.8 Results for run `espp2` . . . 24
A.9 Results for run `s3pp3` . . . 27
A.10 Results for run `s4pp2` . . . 30

B GlassKeyboard Script files **33**
B.1 The `ptest` script file . . . 33
B.2 The `smr184a` script file . . . 39

List of Figures

1 Durations of Responses from Run `s3pp1` . . . 4
2 Response Timings for Commands from Run `s3pp1` . . . 5
3 Durations of Responses from Run `s3pp2` . . . 7
4 Response Timings for Commands from Run `s3pp2` . . . 8
5 Durations of Responses from Run `s3steps` . . . 10
6 Response Timings for Commands from Run `s3steps` . . . 11
7 Durations of Responses from Run `s3steps1` . . . 13
8 Response Timings for Commands from Run `s3steps1` . . . 14

[1] `$Header: d:/Binder2/Timings/RCS/PPTime.tex,v 1.7 2008-07-08 11:50:51-07 Hamilton Exp Hamilton $`
[2] `tocdepth = 2`

9 Durations of Responses from Run `s4pp1` . . . 16
10 Response Timings for Commands from Run `s4pp1` . . . 17
11 Durations of Responses from Run `s4steps` . . . 19
12 Response Timings for Commands from Run `s4steps` . . . 20
13 Durations of Responses from Run `epp1` . . . 22
14 Response Timings for Commands from Run `epp1` . . . 23
15 Durations of Responses from Run `espp2` . . . 25
16 Response Timings for Commands from Run `espp2` . . . 26
17 Durations of Responses from Run `s3pp3` . . . 28
18 Response Timings for Commands from Run `s3pp3` . . . 29
19 Durations of Responses from Run `s4pp2` . . . 31
20 Response Timings for Commands from Run `s4pp2` . . . 32

A Results from Data Captured Response Times

A.1 Results for run `s3pp1`

A.1.1 Test Details

This is a test run using a Spectra III. This test was run at 9600 baud.

The commands to the GlassKeyboard were generated by a script file shown in Appendix B.1, page 33.

There were this many reply times that were analyzed:

```
1       366 lines in S3PP1.TIM
```

A.1.2 Typical response times

The full set of response times are plotted in Figure 2, page 5.

```
   s3pp1.typ
 1       1,   0.001040
 2       2,   0.001049
 3       3,   0.001043
 4       4,   0.001048
 5       5,   0.001042
 6       6,   0.001042
 7       7,   0.001042
 8       8,   0.001042
 9       9,   0.001050
10      10,   0.001042
11      11,   0.001050
12      12,   0.001040
13      13,   0.001041
14      14,   0.000999
15      15,   0.001051
16      16,   0.001043
17      17,   0.001042
18      18,   0.001040
19      19,   0.001121
20      20,   0.001133
21      21,   0.001041
22      22,   0.001049
23      23,   0.001047
24      24,   0.001025
25      25,   0.001041
```

A.1.3 The minimum response times

All response times are plotted in Figure 1, page 4 by number of identical responses and their duration.

```
   s3pp1.min
 1     240,   0.000277
 2     144,   0.000761
 3     155,   0.000869
 4      72,   0.000873
 5     260,   0.000875
 6     107,   0.000881
 7      43,   0.000882
 8     189,   0.000886
 9     208,   0.000887
10     148,   0.000977
11     254,   0.000979
12      92,   0.000987
```

A.1.4 The maximum response times

All response times are plotted in Figure 1, page 4 by number of identical responses and their duration.

```
   s3pp1.max
 1     263,   1.372996
 2     263,   1.376971
 3     263,   1.379557
 4     263,   1.381323
 5     251,   1.383403
 6     263,   1.384132
 7     263,   1.386736
 8     227,   1.387148
 9     190,   1.389125
10     208,   1.398069
11     263,   1.399313
12     263,   1.434365
```

A.1.5 The most common response times

All response times are plotted in Figure 1, page 4 by number of identical responses and their duration.

```
   s3pp1.mde
 1       6,   0.001039
 2       6,   0.001050
 3       9,   0.001034
 4      10,   0.001036
 5      13,   0.001040
 6      20,   0.001047
 7      22,   0.001048
 8      23,   0.001035
 9      24,   0.001049
10      27,   0.001043
11      52,   0.001041
12      70,   0.001042
```

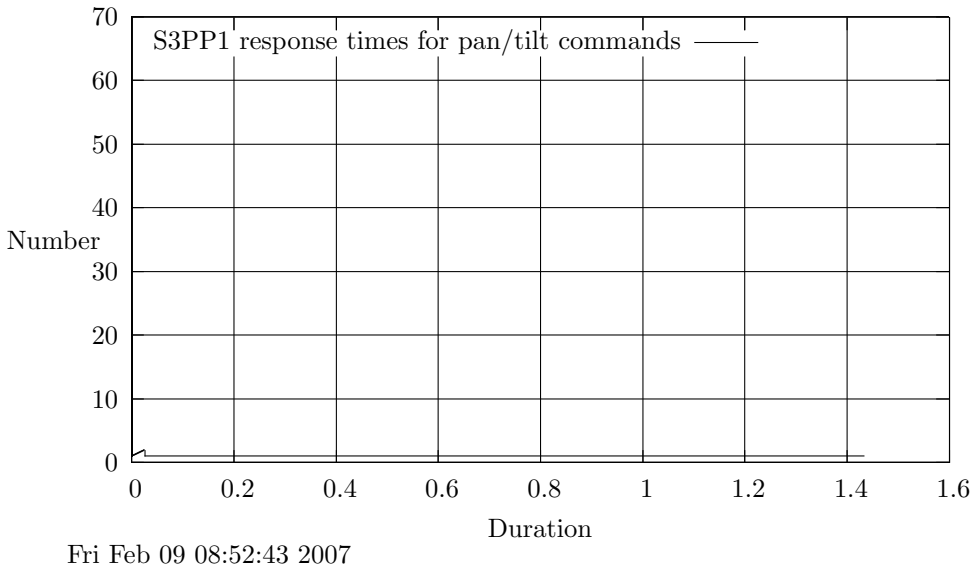

Figure 1: Durations of Responses from Run `s3pp1`

A.1.6 Test Run Data

```
1  Event 1 (2/8/2007 1:05:17.362093 PM) through
2  Event 5,470 (2/8/2007 1:08:41.929367 PM)

1  There were a total of     3591 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 192.705225 seconds from the start of data collection
4
5  There were a total of     1879 DTE bytes transferred
6  The first DTE byte came in at 0.008266 seconds from the start of data collection
7  The last DTE byte was at 204.567274 seconds from the start of data collection
```

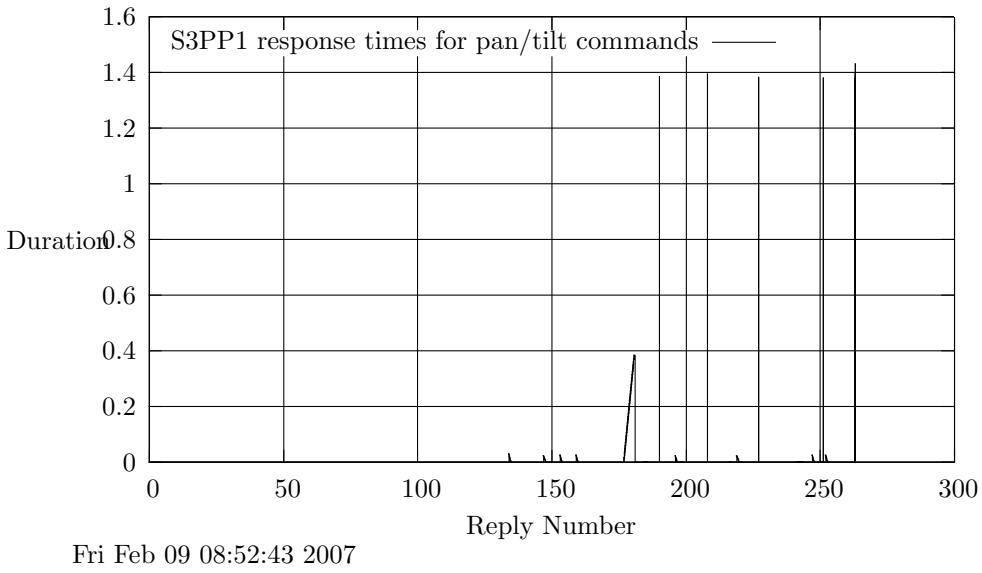

Figure 2: Response Timings for Commands from Run `s3pp1`

A.2 Results for run s3pp2

A.2.1 Test Details

This is a test run using a Spectra III. This test was run at 9600 baud.

The commands to the GlassKeyboard were generated by a script file shown in Appendix B.1, page 33.

There were this many reply times that were analyzed:

```
1        386 lines in S3PP2.TIM
```

A.2.2 Typical response times

The full set of response times are plotted in Figure 4, page 8.

	s3pp2.typ	
1	1,	0.001041
2	2,	0.001042
3	3,	0.001036
4	4,	0.001048
5	5,	0.001034
6	6,	0.001040
7	7,	0.001048
8	8,	0.001037
9	9,	0.001048
10	10,	0.001047
11	11,	0.001034
12	12,	0.001042
13	13,	0.001048
14	14,	0.001035
15	15,	0.001042
16	16,	0.001041
17	17,	0.001049
18	18,	0.001042
19	19,	0.001042
20	20,	0.001048
21	21,	0.001042
22	22,	0.001043
23	23,	0.001040
24	24,	0.001036
25	25,	0.001047

A.2.3 The minimum response times

All response times are plotted in Figure 3, page 7 by number of identical responses and their duration.

	s3pp2.min	
1	232,	0.000867
2	236,	0.000868
3	237,	0.000868
4	148,	0.000869
5	83,	0.000875
6	129,	0.000875
7	236,	0.000875
8	128,	0.000878
9	39,	0.000881
10	225,	0.000881
11	150,	0.000884
12	37,	0.000886

A.2.4 The maximum response times

All response times are plotted in Figure 3, page 7 by number of identical responses and their duration.

	s3pp2.max	
1	218,	1.375392
2	180,	1.377789
3	236,	1.377900
4	218,	1.377998
5	199,	1.379135
6	236,	1.379872
7	218,	1.380074
8	248,	1.385278
9	248,	1.385796
10	199,	1.399626
11	180,	1.410336
12	236,	1.426671

A.2.5 The most common response times

All response times are plotted in Figure 3, page 7 by number of identical responses and their duration.

	s3pp2.mde	
1	8,	0.001046
2	8,	0.001050
3	15,	0.001034
4	15,	0.001035
5	16,	0.001036
6	18,	0.001049
7	23,	0.001043
8	24,	0.001040
9	30,	0.001047
10	30,	0.001048
11	46,	0.001041
12	55,	0.001042

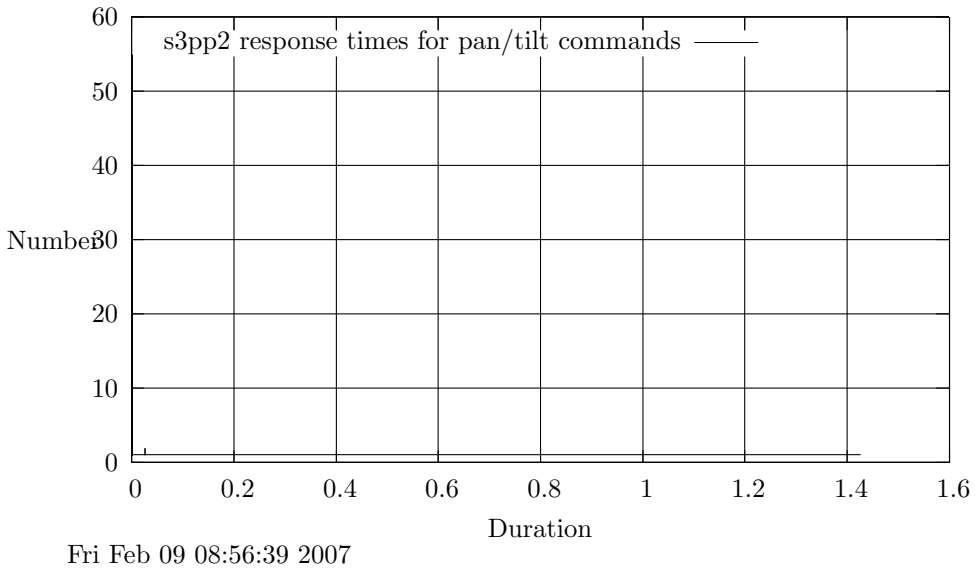

Figure 3: Durations of Responses from Run `s3pp2`

A.2.6 Test Run Data

```
1  Event 1 (2/8/2007 1:10:01.323705 PM) through
2  Event 5,576 (2/8/2007 1:14:13.851309 PM)


1  There were a total of     3591 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 249.089054 seconds from the start of data collection
4
5  There were a total of     1985 DTE bytes transferred
6  The first DTE byte came in at 0.007968 seconds from the start of data collection
7  The last DTE byte was at 252.527604 seconds from the start of data collection
```

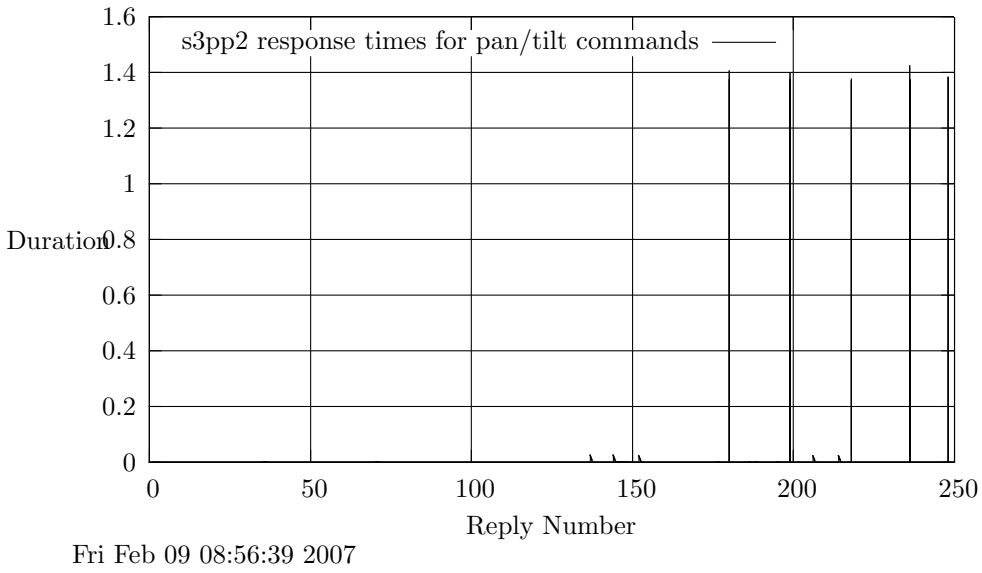

Figure 4: Response Timings for Commands from Run `s3pp2`

A.3 Results for run `s3steps`

A.3.1 Test Details

This is a test run using a Spectra III. This test was run at 9600 baud.

The commands to the GlassKeyboard were generated by a script file shown in Appendix B.2, page 39.

There were this many reply times that were analyzed:

```
1        729 lines in S3STEPS.TIM
```

A.3.2 Typical response times

The full set of response times are plotted in Figure 6, page 11.

```
    s3steps.typ
 1       1,   0.001052
 2       2,   0.001038
 3       3,   0.001049
 4       4,   0.001041
 5       5,   0.001041
 6       6,   0.001048
 7       7,   0.001043
 8       8,   0.001219
 9       9,   0.001042
10      10,   0.001043
11      11,   0.001040
12      12,   0.001048
13      13,   0.001043
14      14,   0.001048
15      15,   0.001047
16      16,   0.001057
17      17,   0.001039
18      18,   0.001035
19      19,   0.001047
20      20,   0.001042
21      21,   0.001042
22      22,   0.001042
23      23,   0.001042
24      24,   0.001040
25      25,   0.001035
```

A.3.3 The minimum response times

All response times are plotted in Figure 5, page 10 by number of identical responses and their duration.

```
    s3steps.min
 1      92,   0.000874
 2     486,   0.000874
 3     491,   0.000874
 4      87,   0.000875
 5     106,   0.000875
 6     119,   0.000875
 7     388,   0.000875
 8     457,   0.000875
 9     478,   0.000875
10      54,   0.000876
11     356,   0.000876
12     360,   0.000876
```

A.3.4 The maximum response times

All response times are plotted in Figure 5, page 10 by number of identical responses and their duration.

```
    s3steps.max
 1     253,   0.001202
 2     296,   0.001203
 3      37,   0.001208
 4     401,   0.001208
 5     613,   0.001209
 6     682,   0.001210
 7     661,   0.001213
 8     380,   0.001214
 9      99,   0.001215
10     111,   0.001215
11       8,   0.001219
12     616,   0.001226
```

A.3.5 The most common response times

All response times are plotted in Figure 5, page 10 by number of identical responses and their duration.

```
    s3steps.mde
 1      10,   0.001034
 2      11,   0.001046
 3      22,   0.001036
 4      26,   0.001050
 5      35,   0.001035
 6      40,   0.001040
 7      52,   0.001047
 8      55,   0.001043
 9      69,   0.001049
10      71,   0.001048
11      76,   0.001041
12     134,   0.001042
```

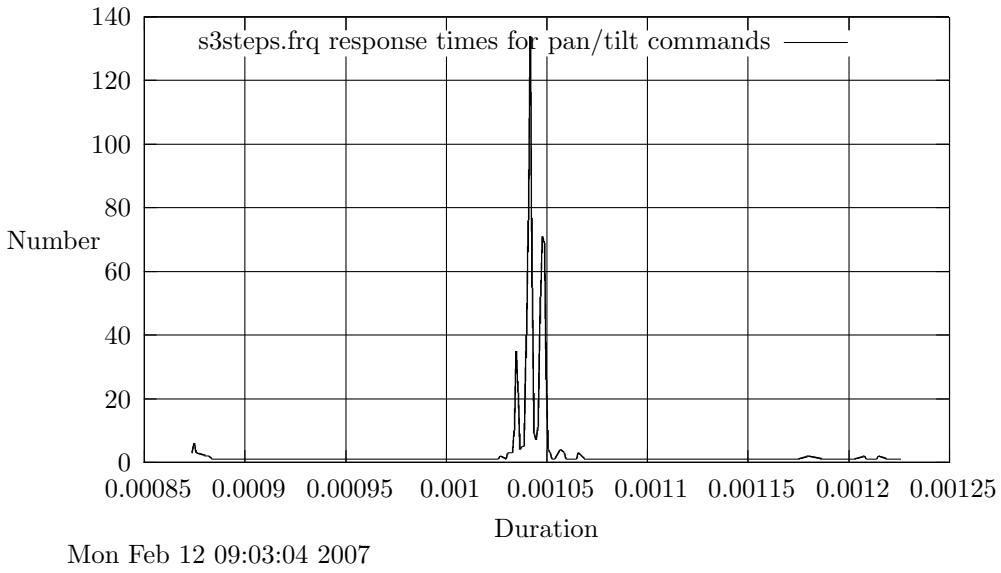

Figure 5: Durations of Responses from Run `s3steps`

A.3.6 Test Run Data

```
1  Event 1 (2/8/2007 11:38:34.871513 AM) through
2  Event 9,025 (2/8/2007 11:43:14.377122 AM)


1  There were a total of     5054 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 279.497405 seconds from the start of data collection
4
5  There were a total of     3971 DTE bytes transferred
6  The first DTE byte came in at 0.007873 seconds from the start of data collection
7  The last DTE byte was at 279.505609 seconds from the start of data collection
```

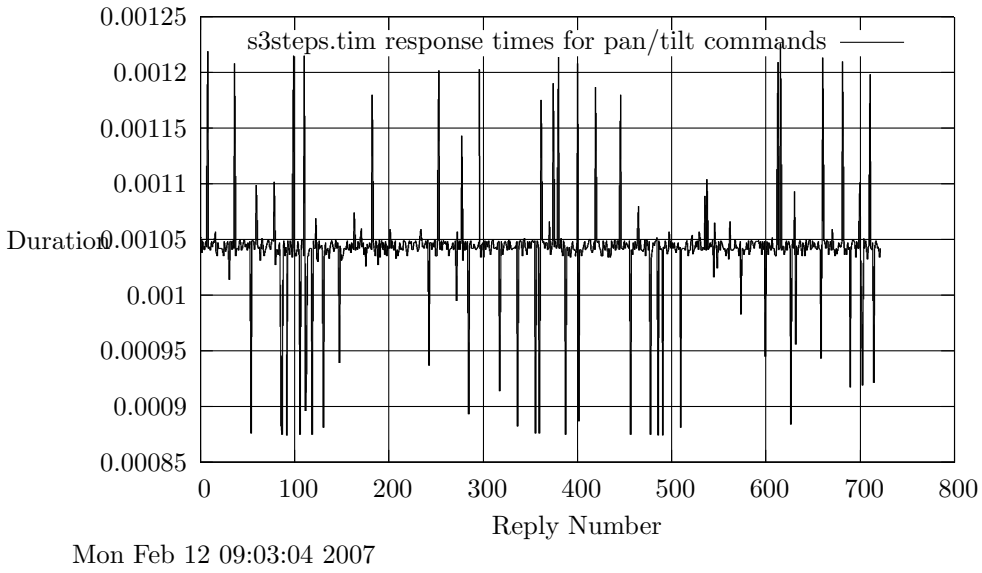

Figure 6: Response Timings for Commands from Run `s3steps`

A.4 Results for run s3steps1

A.4.1 Test Details

This is a test run using a Spectra III. This test was run at 9600 baud.

The commands to the GlassKeyboard were generated by a script file shown in Appendix B.2, page 39.

There were this many reply times that were analyzed:

```
1        729 lines in S3STEPS1.TIM
```

A.4.2 Typical response times

The full set of response times are plotted in Figure 8, page 14.

```
    s3steps1.typ
 1       1,   0.001042
 2       2,   0.001048
 3       3,   0.001042
 4       4,   0.001142
 5       5,   0.001047
 6       6,   0.001126
 7       7,   0.001041
 8       8,   0.001046
 9       9,   0.001043
10      10,   0.001049
11      11,   0.001034
12      12,   0.001041
13      13,   0.001035
14      14,   0.001047
15      15,   0.001047
16      16,   0.001047
17      17,   0.001136
18      18,   0.001047
19      19,   0.001041
20      20,   0.001040
21      21,   0.001040
22      22,   0.001042
23      23,   0.001040
24      24,   0.001040
25      25,   0.001043
```

A.4.3 The minimum response times

All response times are plotted in Figure 7, page 13 by number of identical responses and their duration.

```
    s3steps1.min
 1      88,   0.000875
 2     296,   0.000875
 3     397,   0.000875
 4     532,   0.000876
 5     603,   0.000876
 6     679,   0.000878
 7     217,   0.000880
 8     500,   0.000880
 9     531,   0.000881
10     359,   0.000882
11     423,   0.000883
12     442,   0.000897
```

A.4.4 The maximum response times

All response times are plotted in Figure 7, page 13 by number of identical responses and their duration.

```
    s3steps1.max
 1     646,   0.001209
 2     711,   0.001209
 3     608,   0.001210
 4     705,   0.001211
 5      59,   0.001212
 6     567,   0.001214
 7     607,   0.001215
 8     381,   0.001216
 9     486,   0.001216
10     153,   0.001218
11     172,   0.001218
12     498,   0.001228
```

A.4.5 The most common response times

All response times are plotted in Figure 7, page 13 by number of identical responses and their duration.

```
    s3steps1.mde
 1      12,   0.001044
 2      14,   0.001046
 3      20,   0.001036
 4      23,   0.001034
 5      32,   0.001035
 6      35,   0.001040
 7      52,   0.001049
 8      56,   0.001043
 9      61,   0.001047
10      72,   0.001048
11      88,   0.001041
12     133,   0.001042
```

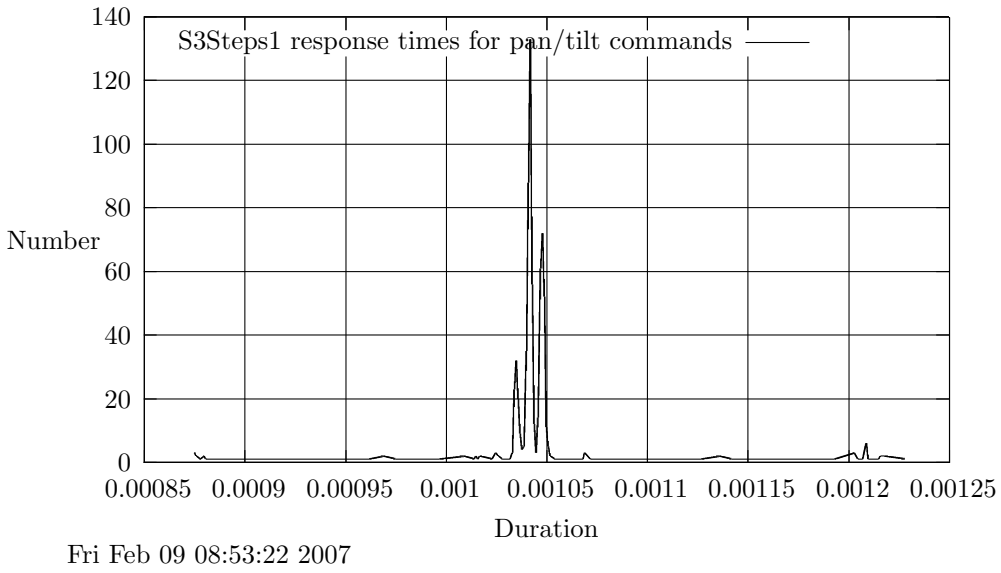

Figure 7: Durations of Responses from Run `s3steps1`

A.4.6 Test Run Data

```
1  Event 1 (2/8/2007 1:20:57.829566 PM) through
2  Event 9,025 (2/8/2007 1:25:37.385668 PM)


1  There were a total of     5054 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 279.548206 seconds from the start of data collection
4
5  There were a total of     3971 DTE bytes transferred
6  The first DTE byte came in at 0.008034 seconds from the start of data collection
7  The last DTE byte was at 279.556102 seconds from the start of data collection
```

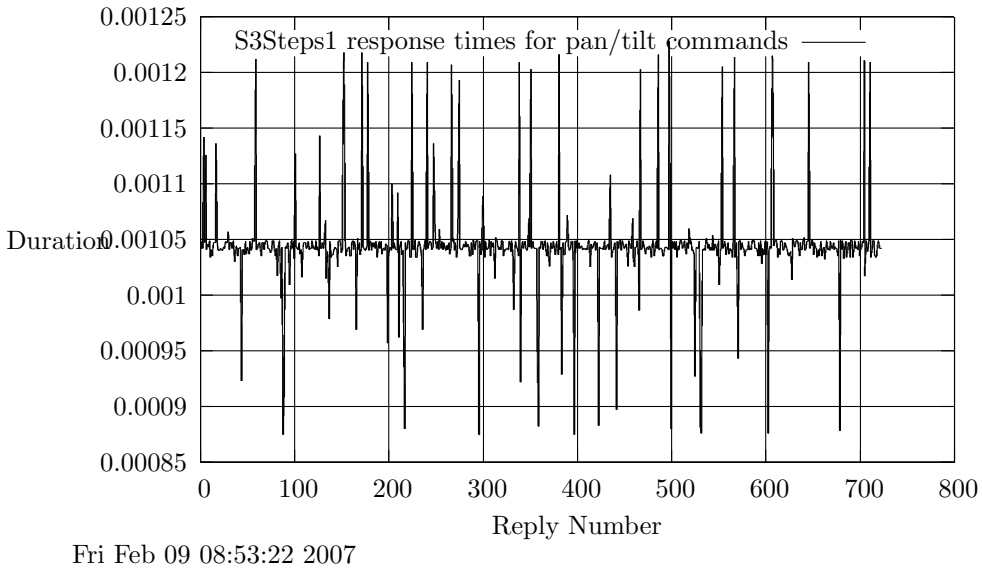

Figure 8: Response Timings for Commands from Run `s3steps1`

A.5 Results for run `s4pp1`

A.5.1 Test Details

This is a test run using a Spectra IV. This test was run at 9600 baud.

The commands to the GlassKeyboard were generated by a script file shown in Appendix B.1, page 33.

There were this many reply times that were analyzed:

```
1        368 lines in S4PP1.TIM
```

A.5.2 Typical response times

The full set of response times are plotted in Figure 10, page 17.

	s4pp1.typ	
1	1,	0.001049
2	2,	0.001042
3	3,	0.001209
4	4,	0.001048
5	5,	0.001036
6	6,	0.001033
7	7,	0.121708
8	11,	0.001043
9	12,	0.001037
10	13,	0.001058
11	14,	0.001047
12	15,	0.001034
13	16,	0.001043
14	17,	0.001050
15	18,	0.001047
16	19,	0.001033
17	20,	0.001043
18	21,	0.001035
19	21,	0.411519
20	21,	0.418501
21	21,	0.433499
22	21,	0.497349
23	21,	0.472561
24	21,	0.489022
25	21,	0.543492

A.5.3 The minimum response times

All response times are plotted in Figure 9, page 16 by number of identical responses and their duration.

	s4pp1.min	
1	136,	0.000865
2	72,	0.000869
3	46,	0.000874
4	53,	0.000874
5	89,	0.000875
6	97,	0.000875
7	89,	0.000880
8	141,	0.000892
9	114,	0.000898
10	125,	0.000900
11	97,	0.000902
12	23,	0.000916

A.5.4 The maximum response times

All response times are plotted in Figure 9, page 16 by number of identical responses and their duration.

	s4pp1.max	
1	85,	2.376834
2	141,	2.378401
3	141,	2.378852
4	134,	2.380062
5	125,	2.380586
6	114,	2.387712
7	141,	2.424853
8	141,	2.426202
9	141,	2.427730
10	141,	2.433805
11	134,	2.440161
12	114,	2.449951

A.5.5 The most common response times

All response times are plotted in Figure 9, page 16 by number of identical responses and their duration.

	s4pp1.mde	
1	5,	0.001044
2	5,	0.001051
3	6,	0.001050
4	7,	0.001034
5	7,	0.001048
6	9,	0.001036
7	9,	0.001047
8	9,	0.001049
9	20,	0.001040
10	31,	0.001043
11	48,	0.001041
12	85,	0.001042

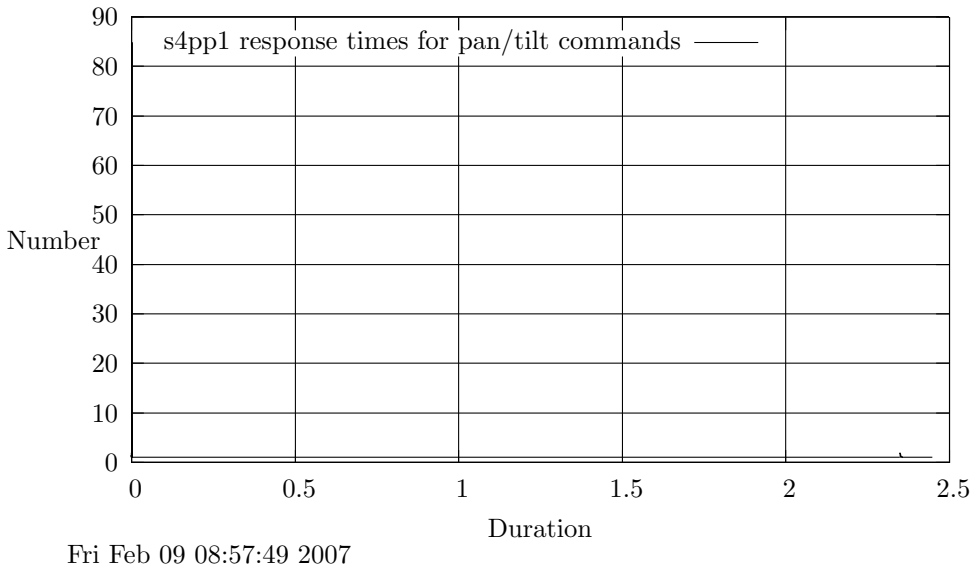

Figure 9: Durations of Responses from Run `s4pp1`

A.5.6 Test Run Data

```
1  Event 1 (2/8/2007 1:28:56.658336 PM) through
2  Event 5,512 (2/8/2007 1:33:58.155628 PM)


1  There were a total of     3591 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 259.849807 seconds from the start of data collection
4
5  There were a total of     1921 DTE bytes transferred
6  The first DTE byte came in at 0.020168 seconds from the start of data collection
7  The last DTE byte was at 301.497292 seconds from the start of data collection
```

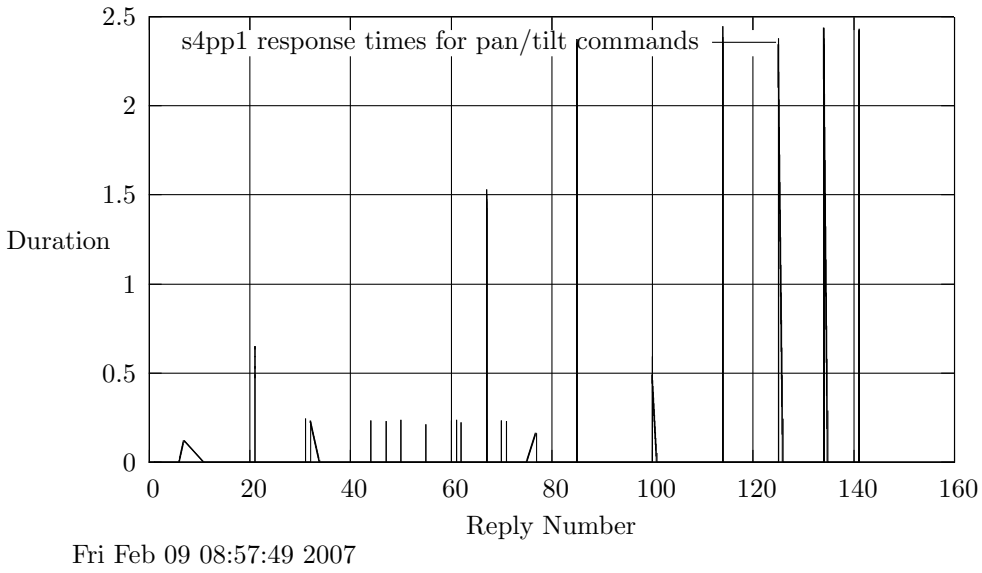

Figure 10: Response Timings for Commands from Run `s4pp1`

A.6 Results for run s4steps

A.6.1 Test Details

This is a test run using a Spectra IV. This test was run at 9600 baud.

The commands to the GlassKeyboard were generated by a script file shown in Appendix B.2, page 39.

There were this many reply times that were analyzed:

```
1        729 lines in S4STEPS.TIM
```

A.6.2 Typical response times

The full set of response times are plotted in Figure 12, page 20.

```
   s4steps.typ
 1       1,   0.001047
 2       2,   0.001048
 3       3,   0.001042
 4       4,   0.001035
 5       5,   0.001043
 6       6,   0.001048
 7       7,   0.001032
 8       8,   0.001036
 9       9,   0.001035
10      10,   0.001097
11      11,   0.001041
12      12,   0.001041
13      13,   0.001049
14      14,   0.001050
15      15,   0.001050
16      16,   0.001042
17      17,   0.001042
18      18,   0.001049
19      19,   0.001198
20      20,   0.001035
21      21,   0.001044
22      22,   0.001037
23      23,   0.001037
24      24,   0.001037
25      25,   0.001035
```

A.6.3 The minimum response times

All response times are plotted in Figure 11, page 19 by number of identical responses and their duration.

```
   s4steps.min
 1     611,   0.000011
 2     574,   0.000868
 3     642,   0.000868
 4     311,   0.000874
 5     465,   0.000875
 6     604,   0.000875
 7     157,   0.000876
 8     446,   0.000879
 9     464,   0.000880
10     118,   0.000881
11     401,   0.000881
12     600,   0.000881
```

A.6.4 The maximum response times

All response times are plotted in Figure 11, page 19 by number of identical responses and their duration.

```
   s4steps.max
 1     522,   0.001144
 2     560,   0.001180
 3      79,   0.001191
 4     515,   0.001193
 5      19,   0.001198
 6     169,   0.001205
 7     105,   0.001209
 8     185,   0.001209
 9     503,   0.001209
10     142,   0.001211
11     432,   0.001211
12     483,   0.001217
```

A.6.5 The most common response times

All response times are plotted in Figure 11, page 19 by number of identical responses and their duration.

```
   s4steps.mde
 1       13,   0.001037
 2       19,   0.001034
 3       19,   0.001046
 4       21,   0.001050
 5       30,   0.001040
 6       32,   0.001035
 7       48,   0.001047
 8       50,   0.001043
 9       68,   0.001048
10       74,   0.001041
11       81,   0.001049
12      137,   0.001042
```

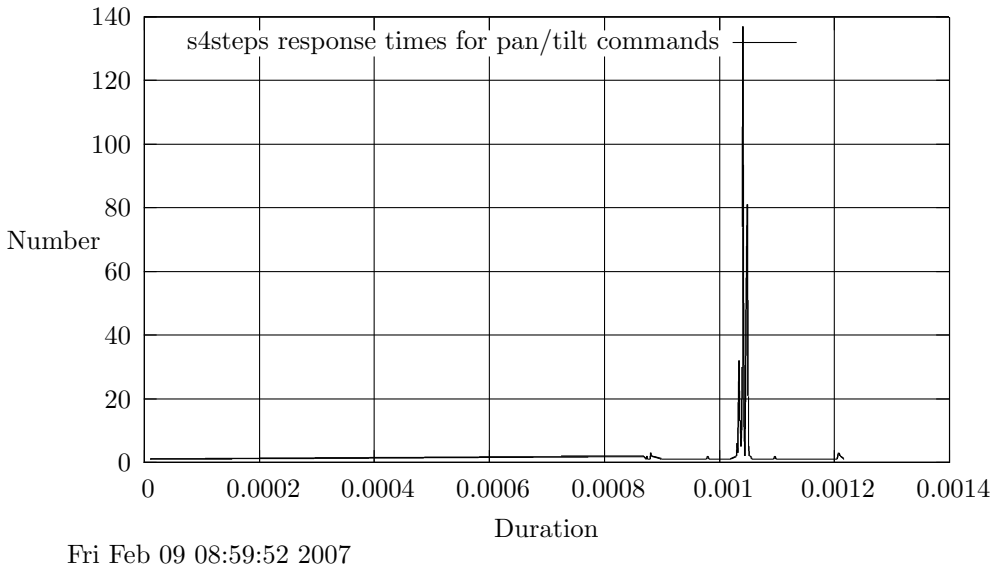

Figure 11: Durations of Responses from Run `s4steps`

A.6.6 Test Run Data

```
1  Event 1 (2/8/2007 11:24:09.969678 AM) through
2  Event 9,025 (2/8/2007 11:28:49.491856 AM)


1  There were a total of     5054 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 279.514622 seconds from the start of data collection
4
5  There were a total of     3971 DTE bytes transferred
6  The first DTE byte came in at 0.013071 seconds from the start of data collection
7  The last DTE byte was at 279.522178 seconds from the start of data collection
```

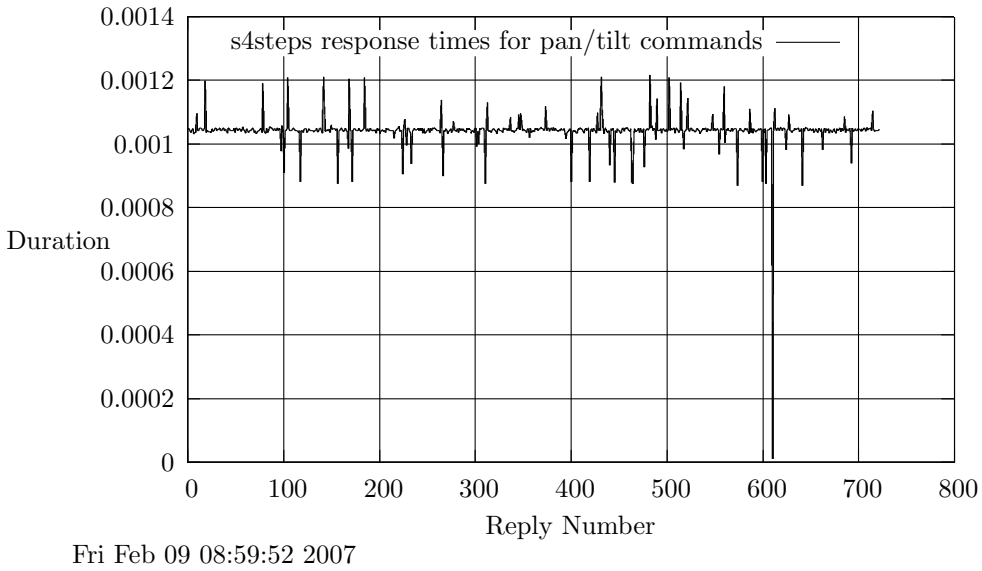

Figure 12: Response Timings for Commands from Run `s4steps`

A.7 Results for run `epp1`

A.7.1 Test Details

This is a test run using an Esprit TI. This test was run at 9600 baud.

The commands to the GlassKeyboard were generated by a script file shown in Appendix B.1, page 33.

There were this many reply times that were analyzed:

```
1        402 lines in EPP1.TIM
```

A.7.2 Typical response times

The full set of response times are plotted in Figure 14, page 23.

	epp1.typ	
1	1,	0.001049
2	2,	0.001042
3	3,	0.001145
4	4,	0.001154
5	5,	0.001043
6	6,	0.001041
7	7,	0.001036
8	8,	0.000881
9	9,	0.001050
10	10,	0.001041
11	11,	0.001041
12	12,	0.001200
13	13,	0.001043
14	14,	0.001043
15	15,	0.001048
16	16,	0.001042
17	17,	0.001035
18	18,	0.001041
19	19,	0.001041
20	20,	0.001036
21	21,	0.001050
22	22,	0.001041
23	23,	0.001041
24	24,	0.001042
25	25,	0.001042

A.7.3 The minimum response times

All response times are plotted in Figure 13, page 22 by number of identical responses and their duration.

	epp1.min	
1	195,	0.000010
2	98,	0.000011
3	103,	0.000874
4	226,	0.000875
5	141,	0.000876
6	287,	0.000876
7	8,	0.000881
8	257,	0.000881
9	364,	0.000881
10	264,	0.000883
11	347,	0.000883
12	349,	0.000889

A.7.4 The maximum response times

All response times are plotted in Figure 13, page 22 by number of identical responses and their duration.

	epp1.max	
1	62,	0.003021
2	62,	0.003961
3	63,	0.009791
4	213,	0.014062
5	62,	0.014375
6	63,	0.015941
7	61,	0.017188
8	62,	0.018647
9	324,	0.022083
10	213,	0.068962
11	52,	0.155634
12	62,	0.492615

A.7.5 The most common response times

All response times are plotted in Figure 13, page 22 by number of identical responses and their duration.

	epp1.mde	
1	7,	0.001034
2	12,	0.001046
3	12,	0.001050
4	13,	0.001036
5	16,	0.001035
6	21,	0.001040
7	24,	0.001043
8	25,	0.001049
9	32,	0.001048
10	33,	0.001047
11	46,	0.001041
12	67,	0.001042

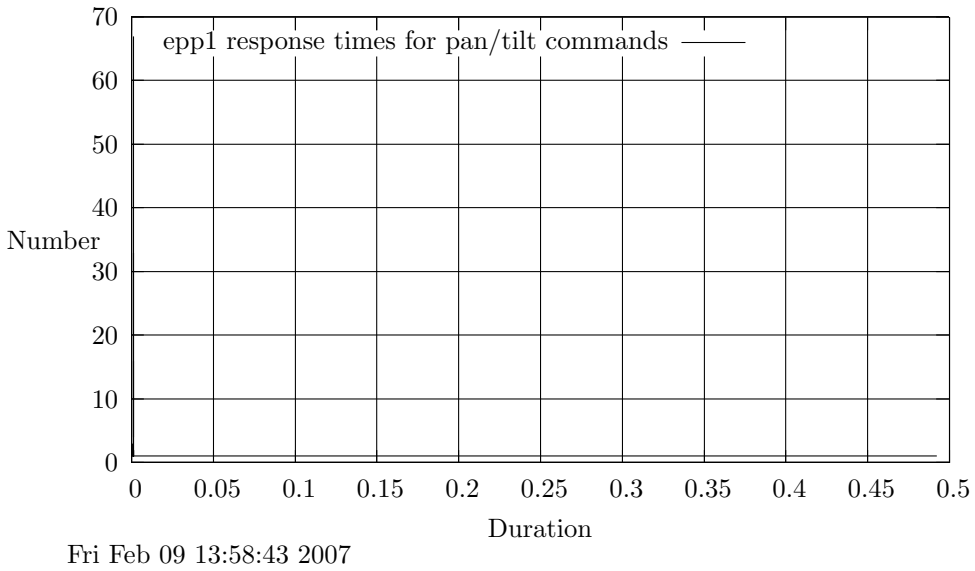

Figure 13: Durations of Responses from Run `epp1`

A.7.6 Test Run Data

```
1  Event 1 (2/9/2007 1:50:05.179955 PM) through
2  Event 5,733 (2/9/2007 1:53:50.367092 PM)

1  There were a total of     3591 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 225.158635 seconds from the start of data collection
4
5  There were a total of     2142 DTE bytes transferred
6  The first DTE byte came in at 0.010398 seconds from the start of data collection
7  The last DTE byte was at 225.187137 seconds from the start of data collection
```

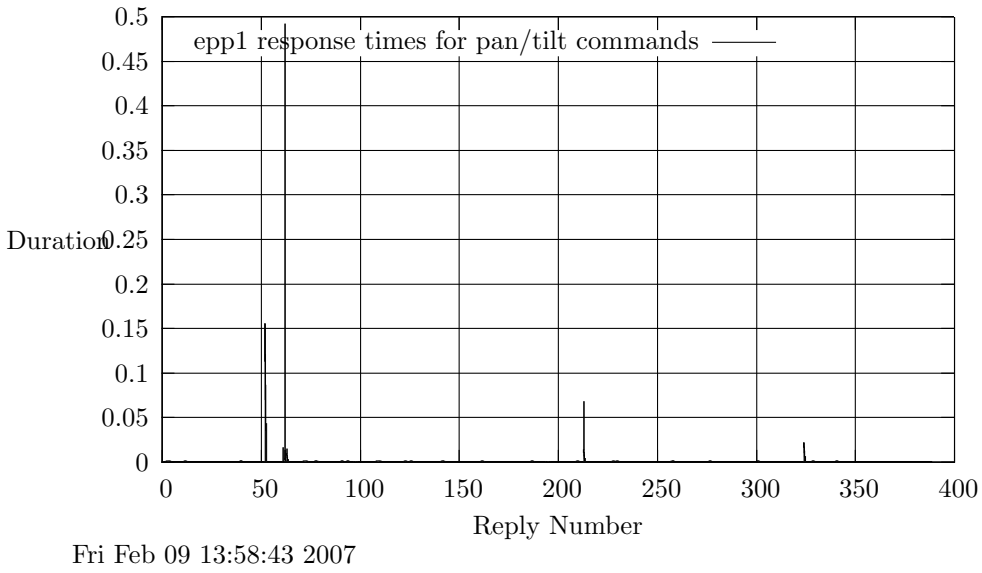

Figure 14: Response Timings for Commands from Run `epp1`

A.8 Results for run espp2

A.8.1 Test Details

This is a test run using an Esprit TI. This test was run at 9600 baud.

There were this many reply times that were analyzed:

```
1       3231 lines in ESPP2.TIM
```

A.8.2 Typical response times

The full set of response times are plotted in Figure 16, page 26.

```
    espp2.typ
 1       1,   0.001035
 2       2,   0.001035
 3       3,   0.001042
 4       4,   0.001044
 5       5,   0.001041
 6       6,   0.001041
 7       7,   0.001041
 8       8,   0.001041
 9       9,   0.001049
10      10,   0.001048
11      11,   0.001049
12      12,   0.001042
13      13,   0.001042
14      14,   0.001000
15      15,   0.001035
16      16,   0.001049
17      17,   0.001050
18      18,   0.001049
19      19,   0.001048
20      20,   0.001035
21      21,   0.001034
22      22,   0.001042
23      23,   0.001048
24      24,   0.001049
25      25,   0.001042
```

A.8.3 The minimum response times

All response times are plotted in Figure 15, page 25 by number of identical responses and their duration.

```
    espp2.min
 1    1402,   0.000011
 2    3128,   0.000515
 3    2566,   0.000840
 4     413,   0.000858
 5     453,   0.000864
 6     547,   0.000867
 7    2471,   0.000867
 8    3132,   0.000867
 9    1379,   0.000868
10    2831,   0.000868
11      45,   0.000869
12     374,   0.000869
```

A.8.4 The maximum response times

All response times are plotted in Figure 15, page 25 by number of identical responses and their duration.

```
    espp2.max
 1    1468,   0.016355
 2    2762,   0.017186
 3    2756,   0.018959
 4      61,   0.020041
 5      62,   0.022918
 6    1465,   0.049167
 7    2488,   0.063966
 8    2756,   0.092503
 9    2690,   0.114064
10    2220,   0.114110
11    2667,   0.121207
12    2756,   0.413968
```

A.8.5 The most common response times

All response times are plotted in Figure 15, page 25 by number of identical responses and their duration.

```
    espp2.mde
 1       74,   0.001046
 2       92,   0.001034
 3      102,   0.001036
 4      109,   0.001050
 5      141,   0.001040
 6      178,   0.001035
 7      230,   0.001043
 8      248,   0.001047
 9      261,   0.001049
10      280,   0.001048
11      362,   0.001041
12      555,   0.001042
```

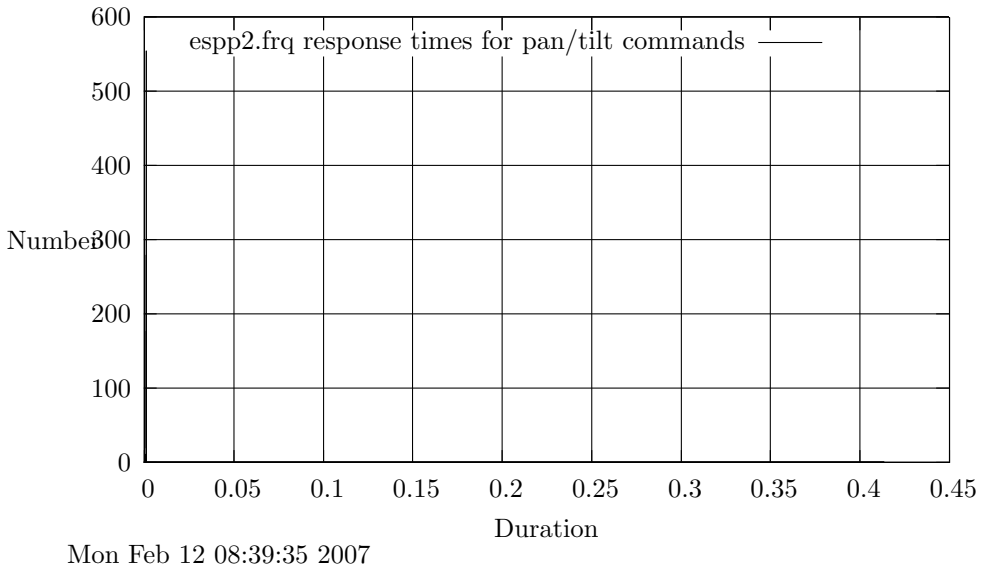

Figure 15: Durations of Responses from Run `espp2`

A.8.6 Test Run Data

```
1  Event 1 (2/12/2007 6:24:17.510999 AM) through
2  Event 42,372 (2/12/2007 6:52:05.907001 AM)

1  There were a total of    25186 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 1668.383182 seconds from the start of data collection
4
5  There were a total of    17186 DTE bytes transferred
6  The first DTE byte came in at 3.398897 seconds from the start of data collection
7  The last DTE byte was at 1668.396002 seconds from the start of data collection
```

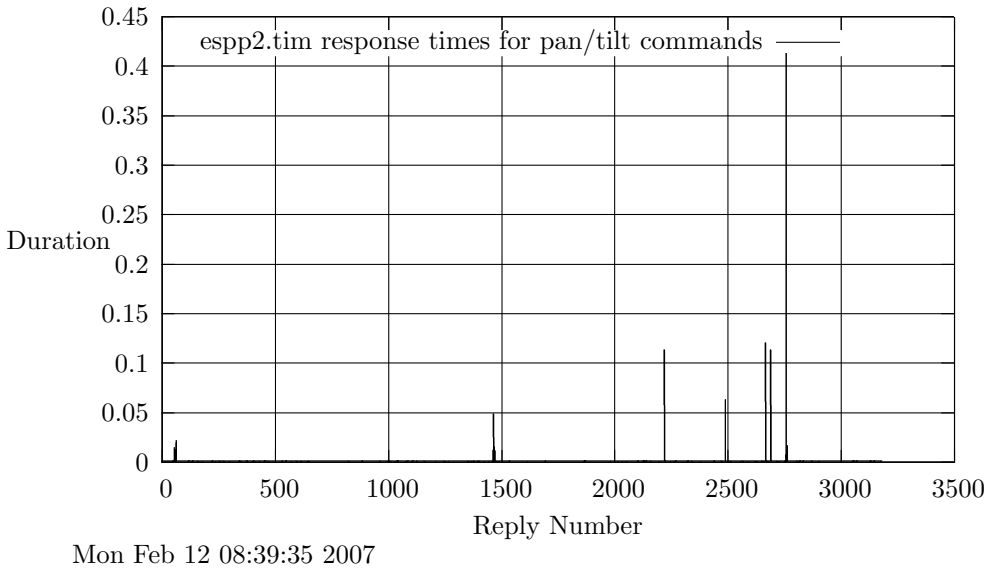

Figure 16: Response Timings for Commands from Run `espp2`

A.9 Results for run s3pp3

A.9.1 Test Details

This is a test run using a Spectra III. This test was run at 9600 baud.

There were this many reply times that were analyzed:

```
1       3477 lines in S3PP3.TIM
```

A.9.2 Typical response times

The full set of response times are plotted in Figure 18, page 29.

```
    s3pp3.typ
 1       1,   0.001043
 2       2,   0.001042
 3       3,   0.001043
 4       4,   0.001034
 5       5,   0.001041
 6       6,   0.001046
 7       7,   0.001047
 8       8,   0.001048
 9       9,   0.001065
10      10,   0.001032
11      11,   0.001035
12      12,   0.001036
13      13,   0.001040
14      14,   0.001088
15      15,   0.001043
16      16,   0.001041
17      17,   0.001049
18      18,   0.001034
19      19,   0.001047
20      20,   0.001043
21      21,   0.001041
22      22,   0.001048
23      23,   0.001042
24      24,   0.001041
25      25,   0.001047
```

A.9.3 The minimum response times

All response times are plotted in Figure 17, page 28 by number of identical responses and their duration.

```
    s3pp3.min
 1     776,   0.000008
 2     709,   0.000009
 3     125,   0.000010
 4    1440,   0.000011
 5     696,   0.000276
 6     959,   0.000276
 7     443,   0.000277
 8     175,   0.000342
 9     421,   0.000433
10    1471,   0.000653
11    1791,   0.000715
12     444,   0.000766
```

A.9.4 The maximum response times

All response times are plotted in Figure 17, page 28 by number of identical responses and their duration.

```
    s3pp3.max
 1     214,   1.429052
 2     521,   1.430304
 3    1287,   1.430520
 4    1782,   1.431866
 5    2324,   1.432793
 6    2055,   1.433636
 7    1539,   1.433950
 8    2019,   1.435608
 9    1005,   1.436373
10     250,   1.436955
11     986,   1.439769
12    2311,   1.453395
```

A.9.5 The most common response times

All response times are plotted in Figure 17, page 28 by number of identical responses and their duration.

```
    s3pp3.mde
 1       53,   0.001046
 2       80,   0.001050
 3       90,   0.001036
 4      115,   0.001034
 5      136,   0.001040
 6      163,   0.001047
 7      190,   0.001035
 8      199,   0.001049
 9      210,   0.001048
10      246,   0.001043
11      436,   0.001041
12      624,   0.001042
```

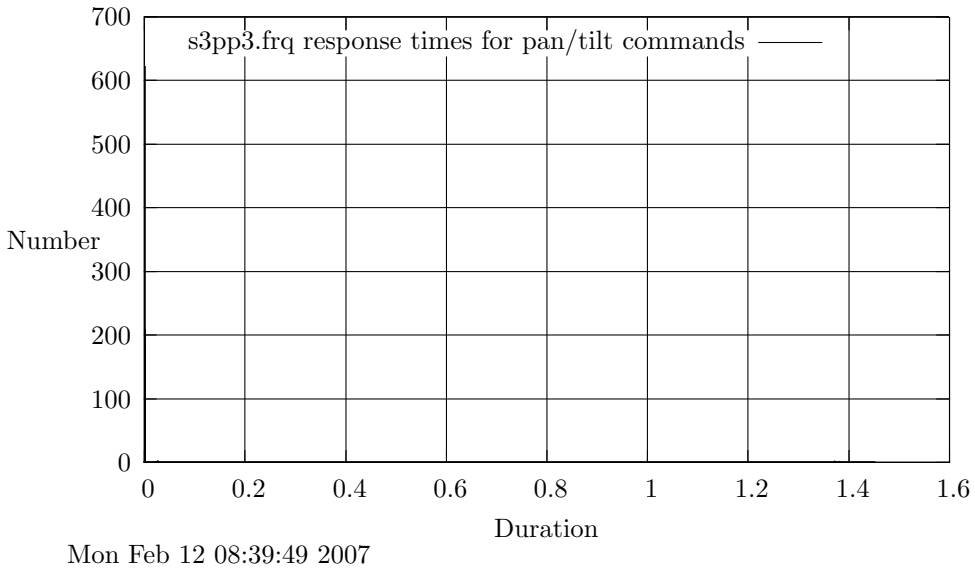

Figure 17: Durations of Responses from Run `s3pp3`

A.9.6 Test Run Data

```
1  Event 1 (2/12/2007 7:34:00.458876 AM) through
2  Event 50,409 (2/12/2007 8:14:34.157333 AM)

1  There were a total of    32382 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 2429.480801 seconds from the start of data collection
4
5  There were a total of    18027 DTE bytes transferred
6  The first DTE byte came in at 0.007894 seconds from the start of data collection
7  The last DTE byte was at 2433.698457 seconds from the start of data collection
```

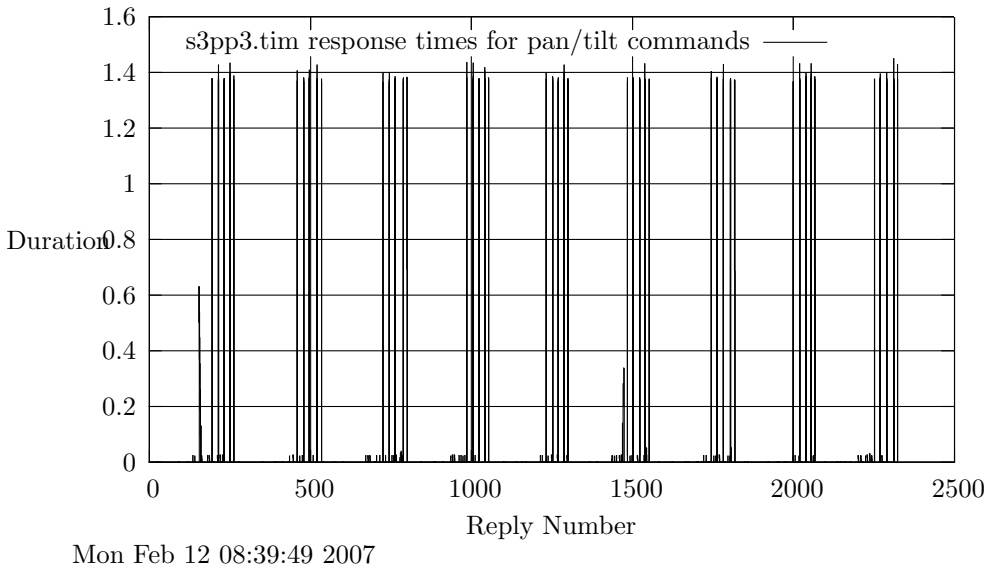

Figure 18: Response Timings for Commands from Run `s3pp3`

A.10 Results for run s4pp2

A.10.1 Test Details

This is a test run using a Spectra IV. This test was run at 9600 baud.

There were this many reply times that were analyzed:

```
1       3115 lines in S4PP2.TIM
```

A.10.2 Typical response times

The full set of response times are plotted in Figure 20, page 32.

```
    s4pp2.typ
 1       1,   0.001048
 2       2,   0.001041
 3       3,   0.001049
 4       4,   0.001042
 5       5,   0.001048
 6       6,   0.001049
 7      10,   0.001033
 8      11,   0.001042
 9      12,   0.001043
10      13,   0.001042
11      14,   0.001043
12      15,   0.001047
13      16,   0.001042
14      17,   0.000991
15      18,   0.001042
16      19,   0.001041
17      20,   0.001058
18      21,   0.001043
19      22,   0.001034
20      22,   0.418603
21      22,   0.418604
22      22,   0.442041
23      22,   0.488701
24      22,   0.472660
25      22,   0.488806
```

A.10.3 The minimum response times

All response times are plotted in Figure 19, page 31 by number of identical responses and their duration.

```
    s4pp2.min
 1     970,   0.000008
 2     686,   0.000009
 3     950,   0.000009
 4     450,   0.000031
 5     685,   0.000098
 6     242,   0.000282
 7     263,   0.000458
 8     545,   0.000571
 9     323,   0.000609
10     817,   0.000687
11     821,   0.000694
12     217,   0.000697
```

A.10.4 The maximum response times

All response times are plotted in Figure 19, page 31 by number of identical responses and their duration.

```
    s4pp2.max
 1    1142,   2.456618
 2     151,   2.457974
 3    1153,   2.458804
 4     960,   2.460054
 5     447,   2.463702
 6     739,   2.473396
 7     566,   2.510263
 8    1107,   2.609313
 9     607,   2.661291
10     147,   2.901572
11    1153,   3.058115
12     554,   3.097490
```

A.10.5 The most common response times

All response times are plotted in Figure 19, page 31 by number of identical responses and their duration.

```
    s4pp2.mde
 1      28,   0.001050
 2      36,   0.001036
 3      38,   0.001044
 4      42,   0.001034
 5      62,   0.001035
 6      71,   0.001047
 7      83,   0.001048
 8     108,   0.001049
 9     139,   0.001040
10     275,   0.001043
11     508,   0.001041
12     746,   0.001042
```

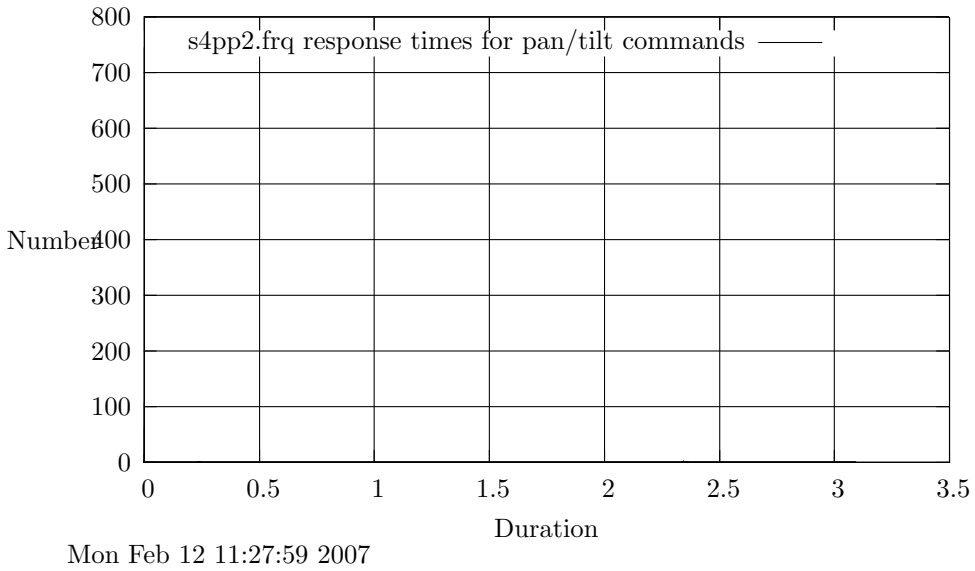

Figure 19: Durations of Responses from Run `s4pp2`

A.10.6 Test Run Data

```
1  Event 1 (2/12/2007 6:53:40.031599 AM) through
2  Event 45,210 (2/12/2007 7:32:03.770372 AM)

1  There were a total of    28784 DCE bytes transferred
2  The first DCE byte came in at 0.000000 seconds from the start of data collection
3  The last DCE byte was at 2258.588392 seconds from the start of data collection
4
5  There were a total of    16426 DTE bytes transferred
6  The first DTE byte came in at 0.007428 seconds from the start of data collection
7  The last DTE byte was at 2303.738773 seconds from the start of data collection
```

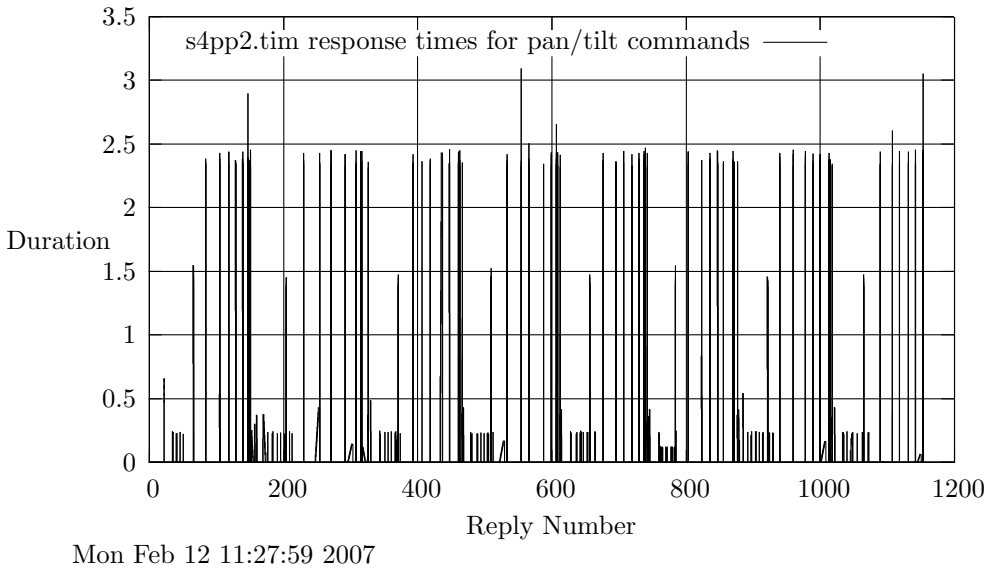

Figure 20: Response Timings for Commands from Run `s4pp2`

B GlassKeyboard Script files

B.1 The `ptest` script file

The unit was forced to cal_0

```
:go_to_zero_pan()
:delay(3000)
:query_pan_pos()
:delay(10)
```

Then presets from 1 ⟶ 32 were cleared

```
:clear_preset(1)
:delay(10)
:clear_preset(2)
:delay(10)
```

Next presets were set, usually, every 12 degrees apart and its location was read out.

```
:go_to_zero_pan()
:delay(3000)
:query_pan_pos()
:delay(10)
:set_pan_pos(12)
:delay(500)
:query_pan_pos()
:delay(10)
:SET_PRESET( 1)
:delay(10)
:set_pan_pos(24)
:delay(500)
:query_pan_pos()
:delay(10)
```

Next each preset was "gone to" and its location was read out.

```
:go_to_zero_pan()
:delay(3000)
:query_pan_pos()
:delay(10)
:go_to_preset( 1)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset( 2)
:delay(500)
:query_pan_pos()
:delay(10)
```

B.1.1 The full listing of `ptest`

```
1  :go_to_zero_pan()
2  :delay(3000)
3  :query_pan_pos()
4  :delay(10)
5
6  :clear_preset(1)
7  :delay(10)
8  :clear_preset(2)
9  :delay(10)
10 :clear_preset(3)
11 :delay(10)
12 :clear_preset(4)
13 :delay(10)
14 :clear_preset(5)
15 :delay(10)
16 :clear_preset(6)
17 :delay(10)
18 :clear_preset(7)
19 :delay(10)
20 :clear_preset(8)
21 :delay(10)
22 :clear_preset(9)
23 :delay(10)
24 :clear_preset(10)
25 :delay(10)
26 :clear_preset(11)
27 :delay(10)
28 :clear_preset(12)
29 :delay(10)
30 :clear_preset(13)
31 :delay(10)
32 :clear_preset(14)
33 :delay(10)
34 :clear_preset(15)
35 :delay(10)
36 :clear_preset(16)
37 :delay(10)
38 :clear_preset(17)
39 :delay(10)
40 :clear_preset(18)
41 :delay(10)
42 :clear_preset(19)
43 :delay(10)
44 :clear_preset(20)
45 :delay(10)
46 :clear_preset(21)
47 :delay(10)
48 :clear_preset(22)
49 :delay(10)
50 :clear_preset(23)
51 :delay(10)
52 :clear_preset(24)
53 :delay(10)
54 :clear_preset(25)
55 :delay(10)
56 :clear_preset(26)
57 :delay(10)
58 :clear_preset(27)
59 :delay(10)
60 :clear_preset(28)
61 :delay(10)
62 :clear_preset(29)
63 :delay(10)
64 :clear_preset(30)
65 :delay(10)
66 :clear_preset(31)
67 :delay(10)
68 :clear_preset(32)
69 :delay(10)
70
71 :go_to_zero_pan()
72 :delay(3000)
73 :query_pan_pos()
74 :delay(10)
75 :set_pan_pos(12)
76 :delay(500)
77 :query_pan_pos()
78 :delay(10)
79 :SET_PRESET( 1)
80 :delay(10)
81 :set_pan_pos(24)
82 :delay(500)
83 :query_pan_pos()
84 :delay(10)
85 :SET_PRESET( 2)
86 :delay(10)
87 :set_pan_pos(36)
88 :delay(500)
89 :query_pan_pos()
90 :delay(10)
91 :SET_PRESET( 3)
92 :delay(10)
93 :set_pan_pos(48)
94 :delay(500)
95 :query_pan_pos()
96 :delay(10)
97 :SET_PRESET( 4)
```

```
 98 :delay(10)
 99 :set_pan_pos(60)
100 :delay(500)
101 :query_pan_pos()
102 :delay(10)
103 :SET_PRESET( 5)
104 :delay(10)
105 :set_pan_pos(72)
106 :delay(500)
107 :query_pan_pos()
108 :delay(10)
109 :SET_PRESET( 6)
110 :delay(10)
111 :set_pan_pos(84)
112 :delay(500)
113 :query_pan_pos()
114 :delay(10)
115 :SET_PRESET( 7)
116 :delay(10)
117 :set_pan_pos(96)
118 :delay(500)
119 :query_pan_pos()
120 :delay(10)
121 :SET_PRESET( 8)
122 :delay(10)
123 :set_pan_pos(108)
124 :delay(500)
125 :query_pan_pos()
126 :delay(10)
127 :SET_PRESET( 9)
128 :delay(10)
129 :set_pan_pos(120)
130 :delay(500)
131 :query_pan_pos()
132 :delay(10)
133 :SET_PRESET(10)
134 :delay(10)
135 :set_pan_pos(124)
136 :delay(500)
137 :query_pan_pos()
138 :delay(10)
139 :SET_PRESET(11)
140 :delay(10)
141 :set_pan_pos(136)
142 :delay(500)
143 :query_pan_pos()
144 :delay(10)
145 :SET_PRESET(12)
146 :delay(10)
147 :set_pan_pos(148)
148 :delay(500)
149 :query_pan_pos()
150 :delay(10)
151 :SET_PRESET(13)
152 :delay(10)
153 :set_pan_pos(160)
154 :delay(500)
155 :query_pan_pos()
156 :delay(10)
157 :SET_PRESET(14)
158 :delay(10)
159 :set_pan_pos(172)
160 :delay(500)
161 :query_pan_pos()
162 :delay(10)
163 :SET_PRESET(15)
164 :delay(10)
165 :set_pan_pos(184)
166 :delay(500)
167 :query_pan_pos()
168 :delay(10)
169 :SET_PRESET(16)
170 :delay(10)
171 :set_pan_pos(196)
172 :delay(500)
173 :query_pan_pos()
174 :delay(10)
175 :SET_PRESET(17)
176 :delay(10)
177 :set_pan_pos(208)
178 :delay(500)
179 :query_pan_pos()
180 :delay(10)
181 :SET_PRESET(18)
182 :delay(10)
183 :set_pan_pos(220)
184 :delay(500)
185 :query_pan_pos()
186 :delay(10)
187 :SET_PRESET(19)
188 :delay(10)
189 :set_pan_pos(224)
190 :delay(500)
191 :query_pan_pos()
192 :delay(10)
193 :SET_PRESET(20)
194 :delay(10)
195 :set_pan_pos(236)
196 :delay(500)
197 :query_pan_pos()
198 :delay(10)
199 :SET_PRESET(21)
200 :delay(10)
201 :set_pan_pos(248)
202 :delay(500)
203 :query_pan_pos()
204 :delay(10)
205 :SET_PRESET(22)
206 :delay(10)
207 :set_pan_pos(260)
208 :delay(500)
209 :query_pan_pos()
210 :delay(10)
211 :SET_PRESET(23)
212 :delay(10)
213 :set_pan_pos(272)
214 :delay(500)
215 :query_pan_pos()
216 :delay(10)
217 :SET_PRESET(24)
218 :delay(10)
219 :set_pan_pos(284)
220 :delay(500)
221 :query_pan_pos()
222 :delay(10)
223 :SET_PRESET(25)
224 :delay(10)
225 :set_pan_pos(296)
226 :delay(500)
227 :query_pan_pos()
228 :delay(10)
229 :SET_PRESET(26)
230 :delay(10)
231 :set_pan_pos(308)
232 :delay(500)
233 :query_pan_pos()
234 :delay(10)
235 :SET_PRESET(27)
236 :delay(10)
237 :set_pan_pos(320)
238 :delay(500)
239 :query_pan_pos()
240 :delay(10)
241 :SET_PRESET(28)
242 :delay(10)
243 :set_pan_pos(336)
244 :delay(500)
245 :query_pan_pos()
246 :delay(10)
247 :SET_PRESET(29)
248 :delay(10)
249 :set_pan_pos(348)
250 :delay(500)
251 :query_pan_pos()
252 :delay(10)
253 :SET_PRESET(30)
254 :delay(10)
255 :set_pan_pos(360)
256 :delay(500)
257 :query_pan_pos()
258 :delay(10)
259 :SET_PRESET(31)
260 :delay(10)
261
262 :go_to_zero_pan()
263 :delay(3000)
264 :query_pan_pos()
265 :delay(10)
266 :go_to_preset( 1)
267 :delay(500)
268 :query_pan_pos()
269 :delay(10)
270 :go_to_preset( 2)
271 :delay(500)
272 :query_pan_pos()
273 :delay(10)
274 :go_to_preset( 3)
275 :delay(500)
276 :query_pan_pos()
277 :delay(10)
278 :go_to_preset( 4)
279 :delay(500)
280 :query_pan_pos()
281 :delay(10)
282 :go_to_preset( 5)
283 :delay(500)
284 :query_pan_pos()
285 :delay(10)
286 :go_to_preset( 6)
287 :delay(500)
```

```
:query_pan_pos()
:delay(10)
:go_to_preset( 7)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset( 8)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset( 9)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(10)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(11)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(12)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(13)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(14)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(15)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(16)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(17)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(18)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(19)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(20)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(21)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(22)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(23)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(24)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(25)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(26)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(27)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(28)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(29)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(30)
:delay(500)
:query_pan_pos()
:delay(10)
:go_to_preset(31)
:delay(500)
:query_pan_pos()
:delay(10)


:go_to_zero_pan()
:delay(3000)
:query_pan_pos()
:delay(10)
:go_to_preset( 1)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset( 2)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset( 3)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset( 4)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset( 5)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset( 6)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset( 7)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset( 8)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset( 9)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(10)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(11)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(12)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(13)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(14)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(15)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(16)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(17)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(18)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(19)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(20)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(21)
:delay(400)
```

```
:query_pan_pos()
:delay(10)
:go_to_preset(22)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(23)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(24)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(25)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(26)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(27)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(28)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(29)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(30)
:delay(400)
:query_pan_pos()
:delay(10)
:go_to_preset(31)
:delay(400)
:query_pan_pos()
:delay(10)


:go_to_zero_pan()
:delay(3000)
:query_pan_pos()
:delay(10)
:go_to_preset( 1)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset( 2)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset( 3)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset( 4)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset( 5)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset( 6)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset( 7)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset( 8)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset( 9)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(10)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(11)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(12)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(13)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(14)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(15)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(16)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(17)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(18)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(19)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(20)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(21)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(22)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(23)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(24)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(25)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(26)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(27)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(28)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(29)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(30)
:delay(300)
:query_pan_pos()
:delay(10)
:go_to_preset(31)
:delay(300)
:query_pan_pos()
:delay(10)


:go_to_zero_pan()
:delay(3000)
:query_pan_pos()
:delay(10)
:go_to_preset( 1)
:delay(200)
:query_pan_pos()
:delay(10)
:go_to_preset( 2)
:delay(200)
:query_pan_pos()
:delay(10)
:go_to_preset( 3)
:delay(200)
:query_pan_pos()
:delay(10)
```

```
668  :go_to_preset( 4)
669  :delay(200)
670  :query_pan_pos()
671  :delay(10)
672  :go_to_preset( 5)
673  :delay(200)
674  :query_pan_pos()
675  :delay(10)
676  :go_to_preset( 6)
677  :delay(200)
678  :query_pan_pos()
679  :delay(10)
680  :go_to_preset( 7)
681  :delay(200)
682  :query_pan_pos()
683  :delay(10)
684  :go_to_preset( 8)
685  :delay(200)
686  :query_pan_pos()
687  :delay(10)
688  :go_to_preset( 9)
689  :delay(200)
690  :query_pan_pos()
691  :delay(10)
692  :go_to_preset(10)
693  :delay(200)
694  :query_pan_pos()
695  :delay(10)
696  :go_to_preset(11)
697  :delay(200)
698  :query_pan_pos()
699  :delay(10)
700  :go_to_preset(12)
701  :delay(200)
702  :query_pan_pos()
703  :delay(10)
704  :go_to_preset(13)
705  :delay(200)
706  :query_pan_pos()
707  :delay(10)
708  :go_to_preset(14)
709  :delay(200)
710  :query_pan_pos()
711  :delay(10)
712  :go_to_preset(15)
713  :delay(200)
714  :query_pan_pos()
715  :delay(10)
716  :go_to_preset(16)
717  :delay(200)
718  :query_pan_pos()
719  :delay(10)
720  :go_to_preset(17)
721  :delay(200)
722  :query_pan_pos()
723  :delay(10)
724  :go_to_preset(18)
725  :delay(200)
726  :query_pan_pos()
727  :delay(10)
728  :go_to_preset(19)
729  :delay(200)
730  :query_pan_pos()
731  :delay(10)
732  :go_to_preset(20)
733  :delay(200)
734  :query_pan_pos()
735  :delay(10)
736  :go_to_preset(21)
737  :delay(200)
738  :query_pan_pos()
739  :delay(10)
740  :go_to_preset(22)
741  :delay(200)
742  :query_pan_pos()
743  :delay(10)
744  :go_to_preset(23)
745  :delay(200)
746  :query_pan_pos()
747  :delay(10)
748  :go_to_preset(24)
749  :delay(200)
750  :query_pan_pos()
751  :delay(10)
752  :go_to_preset(25)
753  :delay(200)
754  :query_pan_pos()
755  :delay(10)
756  :go_to_preset(26)
757  :delay(200)
758  :query_pan_pos()
759  :delay(10)
760  :go_to_preset(27)
761  :delay(200)
762  :query_pan_pos()
763  :delay(10)
764  :go_to_preset(28)
765  :delay(200)
766  :query_pan_pos()
767  :delay(10)
768  :go_to_preset(29)
769  :delay(200)
770  :query_pan_pos()
771  :delay(10)
772  :go_to_preset(30)
773  :delay(200)
774  :query_pan_pos()
775  :delay(10)
776  :go_to_preset(31)
777  :delay(200)
778  :query_pan_pos()
779  :delay(10)
780
781
782  :go_to_zero_pan()
783  :delay(3000)
784  :query_pan_pos()
785  :delay(10)
786  :go_to_preset( 1)
787  :delay(100)
788  :query_pan_pos()
789  :delay(10)
790  :go_to_preset( 2)
791  :delay(100)
792  :query_pan_pos()
793  :delay(10)
794  :go_to_preset( 3)
795  :delay(100)
796  :query_pan_pos()
797  :delay(10)
798  :go_to_preset( 4)
799  :delay(100)
800  :query_pan_pos()
801  :delay(10)
802  :go_to_preset( 5)
803  :delay(100)
804  :query_pan_pos()
805  :delay(10)
806  :go_to_preset( 6)
807  :delay(100)
808  :query_pan_pos()
809  :delay(10)
810  :go_to_preset( 7)
811  :delay(100)
812  :query_pan_pos()
813  :delay(10)
814  :go_to_preset( 8)
815  :delay(100)
816  :query_pan_pos()
817  :delay(10)
818  :go_to_preset( 9)
819  :delay(100)
820  :query_pan_pos()
821  :delay(10)
822  :go_to_preset(10)
823  :delay(100)
824  :query_pan_pos()
825  :delay(10)
826  :go_to_preset(11)
827  :delay(100)
828  :query_pan_pos()
829  :delay(10)
830  :go_to_preset(12)
831  :delay(100)
832  :query_pan_pos()
833  :delay(10)
834  :go_to_preset(13)
835  :delay(100)
836  :query_pan_pos()
837  :delay(10)
838  :go_to_preset(14)
839  :delay(100)
840  :query_pan_pos()
841  :delay(10)
842  :go_to_preset(15)
843  :delay(100)
844  :query_pan_pos()
845  :delay(10)
846  :go_to_preset(16)
847  :delay(100)
848  :query_pan_pos()
849  :delay(10)
850  :go_to_preset(17)
851  :delay(100)
852  :query_pan_pos()
853  :delay(10)
854  :go_to_preset(18)
855  :delay(100)
856  :query_pan_pos()
857  :delay(10)
```

```
:go_to_preset(19)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(20)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(21)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(22)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(23)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(24)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(25)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(26)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(27)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(28)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(29)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(30)
:delay(100)
:query_pan_pos()
:delay(10)
:go_to_preset(31)
:delay(100)
:query_pan_pos()
:delay(10)


:go_to_zero_pan()
:delay(3000)
:query_pan_pos()
:delay(10)
:go_to_preset( 1)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset( 2)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset( 3)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset( 4)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset( 5)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset( 6)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset( 7)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset( 8)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset( 9)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(10)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(11)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(12)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(13)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(14)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(15)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(16)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(17)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(18)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(19)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(20)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(21)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(22)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(23)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(24)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(25)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(26)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(27)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(28)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(29)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(30)
:delay(50)
:query_pan_pos()
:delay(10)
:go_to_preset(31)
:delay(50)
:query_pan_pos()
:delay(10)


```

B.2 The smr184a script file

The unit was forced to cal_0

```
:go_to_zero_pan()
:delay(3000)
```

The location was read out.

```
:query_pan_pos()
:delay(10)
```

Then the location was forced to every degree step, and the results read out, from 1 to 359 degrees.

```
:set_pan_pos( 1)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 2)
:delay(500)
:query_pan_pos()
:delay(10)
```

B.2.1 The full listing of smr184a

```
:go_to_zero_pan()
:delay(3000)
:query_pan_pos()
:delay(10)
:set_pan_pos( 1)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 2)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 3)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 4)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 5)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 6)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 7)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 8)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos( 9)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(10)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(11)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(12)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(13)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(14)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(15)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(16)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(17)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(18)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(19)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(20)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(21)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(22)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(23)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(24)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(25)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(26)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(27)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(28)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(29)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(30)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(31)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(32)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(33)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(34)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(35)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(36)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(37)
```

```
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(38)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(39)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(40)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(41)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(42)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(43)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(44)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(45)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(46)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(47)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(48)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(49)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(50)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(51)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(52)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(53)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(54)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(55)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(56)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(57)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(58)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(59)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(60)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(61)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(62)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(63)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(64)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(65)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(66)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(67)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(68)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(69)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(70)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(71)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(72)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(73)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(74)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(75)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(76)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(77)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(78)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(79)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(80)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(81)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(82)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(83)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(84)
:delay(500)
:query_pan_pos()
```

```
:delay(10)
:set_pan_pos(85)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(86)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(87)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(88)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(89)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(90)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(91)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(92)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(93)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(94)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(95)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(96)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(97)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(98)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(99)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(100)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(101)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(102)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(103)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(104)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(105)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(106)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(107)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(108)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(109)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(110)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(111)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(112)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(113)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(114)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(115)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(116)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(117)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(118)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(119)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(120)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(121)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(122)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(123)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(124)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(125)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(126)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(127)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(128)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(129)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(130)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(131)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(132)
```

```
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(133)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(134)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(135)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(136)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(137)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(138)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(139)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(140)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(141)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(142)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(143)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(144)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(145)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(146)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(147)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(148)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(149)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(150)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(151)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(152)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(153)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(154)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(155)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(156)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(157)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(158)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(159)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(160)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(161)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(162)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(163)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(164)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(165)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(166)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(167)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(168)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(169)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(170)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(171)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(172)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(173)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(174)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(175)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(176)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(177)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(178)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(179)
:delay(500)
:query_pan_pos()
```

```
:delay(10)
:set_pan_pos(180)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(181)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(182)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(183)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(184)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(185)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(186)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(187)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(188)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(189)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(190)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(191)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(192)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(193)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(194)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(195)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(196)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(197)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(198)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(199)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(200)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(201)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(202)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(203)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(204)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(205)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(206)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(207)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(208)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(209)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(210)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(211)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(212)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(213)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(214)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(215)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(216)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(217)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(218)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(219)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(220)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(221)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(222)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(223)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(224)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(225)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(226)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(227)
```

```
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(228)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(229)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(230)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(231)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(232)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(233)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(234)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(235)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(236)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(237)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(238)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(239)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(240)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(241)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(242)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(243)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(244)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(245)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(246)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(247)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(248)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(249)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(250)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(251)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(252)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(253)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(254)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(255)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(256)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(257)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(258)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(259)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(260)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(261)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(262)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(263)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(264)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(265)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(266)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(267)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(268)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(269)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(270)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(271)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(272)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(273)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(274)
:delay(500)
:query_pan_pos()
```

```
:delay(10)
:set_pan_pos(275)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(276)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(277)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(278)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(279)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(280)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(281)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(282)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(283)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(284)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(285)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(286)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(287)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(288)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(289)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(290)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(291)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(292)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(293)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(294)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(295)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(296)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(297)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(298)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(299)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(300)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(301)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(302)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(303)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(304)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(305)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(306)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(307)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(308)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(309)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(310)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(311)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(312)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(313)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(314)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(315)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(316)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(317)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(318)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(319)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(320)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(321)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(322)
```

```
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(323)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(324)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(325)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(326)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(327)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(328)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(329)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(330)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(331)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(332)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(333)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(334)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(335)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(336)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(337)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(338)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(339)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(340)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(341)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(342)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(343)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(344)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(345)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(346)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(347)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(348)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(349)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(350)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(351)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(352)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(353)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(354)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(355)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(356)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(357)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(358)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(359)
:delay(500)
:query_pan_pos()
:delay(10)
:set_pan_pos(360)
:delay(500)
:query_pan_pos()
:delay(10)
```

Index

epp1, 21
espp2, 24
Esprit TI, 21, 24

GlassKeyboard, 3, 6, 9, 12, 15, 18, 21, 33

s3pp1, 3
s3pp2, 6
s3pp3, 27
s3steps, 9
s3steps1, 12
s4pp1, 15
s4pp2, 30
s4steps, 18
Spectra III, 3, 6, 9, 12, 27
Spectra IV, 15, 18, 30