FORTRA

beSTORM

Embedded Device Waveform EDSA Guide

202303231154

Table of Contents

Background **1**

Prerequisites **3**

Usage **4**

Process Control **6**

Timing **7**

Digital **9**

Analog **10**

Harness **11**

DUT Control **12**

Logging **13**

Network **14**

Testing **15**

User Interface **17**

Appendix **18**

Appendix A – Tool concise help printout 18

Appendix B – Examples of warnings and exceptions 20

Background

ISASecure EDSA requirements dictate that during testing, a DUT will output a specific electrical signal for each type of output. The timing accuracy of that signal's transition will be monitored for critical degradations, as a means of confirming the validity of "essential downward services".

There are 3 possible signals coming out from the DUT and/or its test-harness:

1. Electric digital, TTL level waveform with a period of 1s up, and 2s down.
2. Electric analog, 4-20mA signal (usually at "signal-level" voltages) waveform with a period of going up 10 x 1s at each range/10 level, then going down 20 x 1s at each range/20 level.
3. For signals that do not conform to the upper 2 groups, the DUT incorporates its own testharness that monitors those signals and switches the state of a digital out as a result.

The EDW monitor is NOT expected to monitor the voltage output accuracy of the digital (a) and analog (b) channels, and subsequently also not the "shape" of the analog waveform - but only their correct timing. Any transition with a timing jitter bigger than allowed, or a test-harness fault signal (c), will trigger an exception, register it in the log, and optionally reset the DUT.

However, the new EDSA-310-ERT.R30 calls for decreasing the max allowed polling lag ("measurement jitter") from 100ms (10%) to 10ms (1%). Luckily, the new value was already the default since the first version (see -pl option). As well as adding three new pieces of information:

A transition in an Analog signal is considered to occur once the sampled voltage reaches at least 90% of its target value. For example, suppose our signal range is 0..5v. A step up from 0.5v to 1.0v would be already satisfied if the sampled value at the time of transition would be at least 0.95v (0.5 + (0.90 * 0.5)). A step down from 5v to 4.75v would be already satisfied if the sampled value at the time of transition would be at most 4.775v (5.0 - (0.90 * 0.25)). To accommodate that, a new -at (Analog Threshold) parameter with a default of 0.9 was added.

- NOTE 3, Bullet #1: Transition timing jitters must be under the given maximum allowed (-mj) at least 95% of the samples, or at a higher percentage supplied by the vendor. To accommodate that, a new -mc (Minimum Confidence) parameter with a default/minimum of 0.95 was added. To reproduce the "old", strict behaviour, set the value to 1.

> **NOTE:** The percentage count restarts together with the tool restart, whether auto or manual.

- NOTE 3, Bullet #2: In parallel with Bullet #1 above, ALL transition timing jitters must be smaller than: max_allowed_polling_lag + (1.5 * max_allowed_jitter). To accommodate that, a new –cj (“Composite Jitter” factor) parameter that factors that expression was added with a default of 1.0.

Prerequisites

The monitor is implemented as a windows script that requires:

- LabJack U3-HV, HW version 1.30, firmware version 1.46, driver ver. 3.42. This is the DAQ unit which electrically interface the DUT and/or its test-harness. Please install its drivers and test it according to the manual before initial use. See http://labjack.com/catalog/u3-hv.

  > **NOTE:** LV model compatibility MAY be achieved using the “-ax” parameter to lower max V).

- Python 2.7.x Monitor script interpreter. See https://www.python.org/.
- LabJackPython (version >= 042414). See http://labjack.com/support/labjackpython.
- pywin32. For process priority control. See http://sourceforge.net/projects/pywin32/
- Windows 32 or 64 bit.

Usage

For a concise usage listing, type “edw-mon.py -h" at the command prompt (see Appendix-A). To start monitoring:

1. Connect the DUT signals to the LabJack ports.
2. Connect the LabJack to the TD.
3. Type “edw-mon.py” to start monitoring with the default values.
4. Optionally, run the beSTORM client.

When starting, the monitor will print and log a large bunch of information that helps guarantee reproducible runs as required, as well as giving vital debugging information. It will then “count-in” for 30 seconds allowing for any necessary DUT booting, and then begin actual monitoring.

If all goes well, you are not supposed to see any output. That’s in order to minimize any processing lags that may affect timing. However, the waveform monitor sends a “heartbeat” to the beSTORM client every 5 seconds, so if the beSTORM client is up you are supposed to see it updated (green). There are basically four kinds of things that can go wrong:

1. Internal OS/HW errors (file system, LabJack, drivers, etc…) – those will print (and log) the related error and terminate the monitor completely. If beSTORM is up, heartbeat will not be updated (red) and that will trigger a beSTORM client self-exception (Monitor Down).
2. Polling exceptions (caused by the system’s inability to catch up with the polling period/lag imposed on it, the priority level used and/or the general system load) – those will print (and log) a “POLL LAG EXCEPTION”. If beSTORM client is up, it will also trigger that exception in it. The monitor will then completely terminate (to allow re-adjustment/analysis of the timing issues), which (as a safety fallback) will stop heartbeat update and cause beSTORM to trigger a self exception (Monitor Down).
3. Timeout exceptions (caused by a digital or analog signal not transitioning for 3 seconds or as given with the “-to” parameter) – those will print (and log) and trigger a “TIMEOUT EXCEPTION” in the beSTORM client if it’s up. The DUT will get a reset signal for 2 seconds, and the monitor will recycle, starting from the count-in.
4. Excessive jitter exceptions (caused by a single or total jitter bigger than the allowed) – those will print (and log) and trigger an “EXCESSIVE JITTER EXCEPTION” in the beSTORM client if it’s up. The DUT will get a reset signal for 2 seconds, and the monitor will recycle, starting from the count-in. Excessive Jitters now trigger an actual exception only if the percentage of valid-bound timing transitions drops under

the given confidence value. Otherwise they only register a warning. NOTE: Digital and Analog transitions keep their own record of “valid” jitter percentages. This makes sense because their rate of transition differs and “slides” in a 2/3 ratio.

5. Composite jitter exceptions, triggered by ANY jitter which is bigger than the value resulting from multiplying the –cj value with the max_allowed_polling_lag + (1.5 * max_allowed_jitter). Note that regardless of the –cj value, this exception will trigger ONLY if the result of the expression is at least as large as the max jitter value. Appendix B presents some examples of these warnings and exceptions.

To skip count-in and/or finish at any time, press ‘Q’.

Please read the rest of this guide for proper use of the monitor.

Process Control

The monitor is started with a default process priority set as “real-time” which can be changed using the “-zz” parameter.

Normally both digital and analog waveforms will be monitored. To monitor only digital or analog, use the “-xx” parameter.

If not encountering internal errors, the monitor will restart upon every jitter or harness trigger, and optionally reset the DUT. To specify a “count-in” duration in seconds, use the “-ci” parameter. The DUT will also go off upon a failure, and then on after count-in finishes. To skip the count-in, just use 0 as the parameter value (the default is 30 seconds). Please note that internal errors will exit the loop.

Timing

The monitor maintains a process loop, with a polling period given with the “-pp” parameter, defaulted to 10ms., meaning that the process will attempt polling and processing every 10ms.

Since system scheduling on a non-RT system cannot assure a sharp 10ms, a maximum lag value can be given with the “-pl” parameter (defaulted to 10ms <EDSAv2> which complies with the new EDSA 310ERT.R30 </EDSAv2>, which defines the extra timing “slack” the polling system has. However, scheduling for next poll is based on the “-ht” hard-timing parameter:

- **1/True** - Next polling should occur at 10ms from the last scheduled poll, regardless of any lag. This is the default, and will answer any hard-RT constraints (-pl defines a TOTAL “slack”).
- **0/False** - Next polling should occur at 10ms from actual poll’s time. (Relative, soft-timing). Any iteration timing exceeding the allowed, will result in sending a “Poll Lag” exception and reporting it.

All three inputs channels (digital, analog and harness) are read at once. However, digital and analog inputs are concurrently handled in separate instances, which does not force them to synchronize beyond the maximum jitter allowed. A significant change of value or state is considered as a transition.

The expected transition time is a function of step-period, for e.g. by default, a digital transition from down state (2 steps) to up state is expected to occur 2s after the last one, while any other transition (digital and analog) is expected to occur just 1s after. The default 1s step-period used in both the digital and analog waves can be modified with the “-sp” parameter.

Jitter is calculated as the absolute value of the actual transition time subtracted by the expected time. The default 100ms max jitter allowed can be modified with the “-mj” parameter. Its behavior is based on the “-ht” hard-timing parameter:

- **1/True** – Both current jitter and cumulative jitter are tested against max jitter (default). Again, this means that the max jitter is regarded as the TOTAL timing slack in which the waveform can have a “spiel”.
- **0/False** – Only current jitter is tested. (Relative, soft-timing).

Any jitter exceeding the allowed, will result in sending an “Excessive Jitter” exception and reporting it. <EDSAv2>

Note that in compliance with EDSA-310-ERT.R30, after Note #3, Bullets #1 and #2:

- An exception will be triggered only if the given Minimum Confidence (see: -mc option) for allowed jitters can’t be maintained.
- An exception will ALSO be triggered if ANY transition timing jitter is greater than: (see:

  ```
  - cj option) composite_factor * (max_allowed_polling_lag +
  (1.5 * max_allowed_jitter)).
  ```

No transition occurring during a default of 3s for each of the digital and analog channels, will result in sending a “Time Out” exception and reporting it. The default can be modified with the “-to” parameter.

Digital

The expected digital transition timings are a function of the step-period, with:

- Number of “up” steps, defaulted to 1, modified with “-du”
- Number of “down” steps, defaulted to 2, modified with “-dd” The DUT’s digital waveform channel goes into LabJack’s FIO4.

Analog

The expected analog transition timings are a function of the step-period, with:

- Number of “up” increments, defaulted to 10, modified with “-au”
- Number of “down” increments, defaulted to 20, modified with “-ad”

Minimum and maximum analog voltages are given with the “-an” and “-ax” parameters, and are defaulted to 0 and 5 respectively. These values are not observed, however, the analog minimum resolution is calculated as the range divided to the higher between up and down increments, and defaulted to (5-0)/max(10,20) = 0.25.

Since analog values are subject to continuous variation, they can’t be just compared throughout the polls. Rather, they are rounded to integer “bins” based on the analog minimum resolution.

A transition is marked as soon as the current bin is different from the last bin.

In compliance with EDSA-310-ERT.R30, After Note #2:

A transition in an Analog signal is considered to occur once the sampled voltage reaches at least 90% of its target value. Technically, the current bin is the one with a voltage proximity of at least 1.0analogthreshold to an adjacent bin (see: -at option).

However, since valid transition times can take more than a poll’s period, including error which span can trigger “false-positive” transitions, the monitor’s analog channel will be “ready for change” only after polling a value associated with the same bin the specified number of times, marking a “stabilization” of the signal. Using smaller polling periods with higher stabilization steps effectively increases the polling resolution without “side-effects”. The number of steps required for stabilization is defaulted to 3, and can be modified using the “-as” parameter. For monitoring high-precision signals only, stabilization can be turned off using “-as 0”. Setting stabilization steps out of reasonable limits is allowed, and can be used to simulate various errors.

The DUT’s analog waveform channel goes into LabJack’s AIN1 (FIO1).

Harness

The digital state coming from a test harness triggering an error is set by “-th” and defaulted to 0 (due to the DIN’s pull-up). Polling this value will result in sending a “Test Harness” exception and reporting it.

The DUT’s harness signal channel goes into LabJack’s FIO6.

> **NOTE:** To skip test-harness monitoring, just set this value and/or connect FIO6 to GND or VS as required.

DUT Control

The DUT can be optionally connected to a relayed outlet which can be switched off when a fault occurs, by signaling it using significant electrical current from DAC0.

With every re-initialization, the monitor will bring DAC0 to 0v. When it recycles, it will bring DAC0 to 5v for two seconds, so any connected switched outlet can be reset. The DUT's recovery/reboot time can be matched using the "count-in" parameter.

Logging

All starts, stops, errors and exceptions are printed to standard output and if applicable, also

APPENDED to a file with a timestamp. The name of the file can be specified using the “-lf” parameter, defaulted to “edw-mon.log”. If not existing, this file will be created.

Each start records all machine environments and parameters for reproducibility requirements.

Network

Exception reporting is done by sending UDP messages to the beSTORM client.

The IP address (or hostname) to report and its UDP port are specified using “-na” and “-ne” parameters, and are defaulted to “localhost” and 6969 respectively. Unsuccessful messaging attempts are logged.

Similarly, heartbeats are sent every 5 seconds to a port specified by “-nh” and defaulted to 6970.

Testing

For verifying the applicability of given parameters in a given environment, it is possible to test the given parameters in two ways. First, a pure software test is possible using the “-st internal” parameter, which will generate and read signals from the internal clock rather than from the DUT/LabJack. This will test that the monitor software is working properly.

Second, using the “-st loopback” parameter, it’s possible to test the monitor by generating signals from the clock as mentioned before, but further sending them out from other LabJack’s terminals while reading them in the same way that signals from the DUT will come (for example, emulating the DUT’s waveform outputs). This will test that both the monitor software and the LabJack hardware are working properly, and is recommended as a part of the setup.

It is safe to assume that realistic processing will be “slower” than an internal test which uses no external hardware, while “faster” than a loopback test, which overheads the LabJack by outputting and rereading test signals.

To use the loopback test, please connect:

- DAC1 to AIN1 (FIO1)
- FIO5 to FIO4
- FIO7 to FIO6

All signal generation parameters are specified in the same way as “real” signals. However, analog signals generated by the LabJack’s DAC are approximately limited to 0..5v.

Generating a test-harness trigger is done by using the “-sh” parameter with a value specifying the probability of having an exception due to a test-harness state trigger. E.g., a poll-period of 10ms and a value of 1000 will probably yield such an exception within 10 seconds. The default is 0, which means that no test-harness exceptions will be generated during testing.

Other errors are possible to simulate using various tight or unreasonable parameters. Keep in mind that it is also possible to isolate digital and analog waveform inputs for testing purposes.

An additional “textfile” software test mode was added. Using –st textfile, causes the EDW tool to sample from a text file as its signal source. Specifying a non-default source file name is done using the –sf option. The source file is a text file contains a line corresponding to each future sample, with the format:

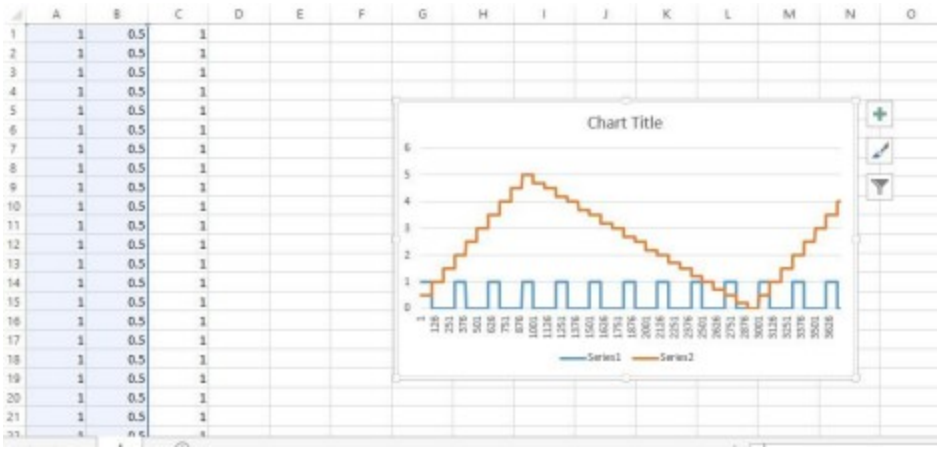
```
Digital_Value<TAB>Analog_Value<TAB>Harness_Value<NEWLINE>
```

All the three values are mandatory, even if you don't use all of them.

All lines are trimmed, and those beginning with '#' can be used as remarks.

To facilitate text file generation, it is possible to use the –sm option, which generates a "reverse" input file from the given timing and signal information. It is then easy to edit the file with a notepad or spreadsheet app to easily visualize the signals and introduce engineered "quirks" for testing purposes.

figure 1 - *input data generated by using the –sm option, then edited with a spreadsheet app. After introducing any desired changes, it can be used as an input to a –st textfile sampler.*



User Interface

To skip count-in, and to finish immediately at any time, please press ‘Q’.

> **NOTE:** All time durations and voltages are expressed in fractional seconds, rounded tmicros.

> **IMPORTANT:** Please consult LabJack’s user guide for guidelines and limits of electrical connections.

Please refer to [ISASecure EDSA-310] Common requirements for communication robustness testing of IP-based protocol implementations and the ISASecure EDSA documentation for information relating to EDSA testing methodology, found at www.isasecure.org. The protocol features and versions under test are defined in the RFCs and other standard documents listed in the normative references sections of the above EDSA specifications.

Appendix

Appendix A – Tool concise help printout

```
usage: edw-mon[-h][-v][-zz {32,32768,128,256}][-xx {0,1,2,3}][-ci
COUNTIN][-pp POLLPERIOD][-pl POLLLAG][-sp STEPPERIOD][-mj MAXJITTER]
[-mc MINCONFIDENCE][-cj COMPOSITEJITTER][-ht {1,0}][-to TIMEOUT][-du
DIGITALUPS][-dd DIGITALDOWNS][-an ANALOGMIN][-ax ANALOGMAX][-au
ANALOGUPS][-ad ANALOGDOWNS][-at ANALOGTHRESHOLD][-as
ANALOGSTABILIZE][-th{0,1}][-na NETADDRESS][-ne NETEXCEPTION][-nh
NETHEARTBEAT][-lf LOGFNAME][-st {internal,loopback,textfile}][-sf
SWTESTFNAME][-sm SWTESTFMAKER][-sh SWTESTHARP]
```

Optional arguments:

- -h, --help show this help message and exit
- -v, --version show program's version number and exit - zz {32,32768,128,256}, --zz {32,32768,128,256}

Monitor's process priority, options are:

- 32 - Normal
- 32768 - Above Normal
- 128 - High
- 256 - Realtime (default)
- `-xx {0,1,2,3}, --xx {0,1,2,3}`

Waveforms to monitor:

- 0 - None (harness only)
- 1 - Digital only
- 2 - Analog only
- 3 - Digital and Analog
  Default = 3

```
-ci COUNTIN, --countin COUNTIN Count-in seconds to start. 0 to skip,
default=30.
```

```
-pp POLLPERIOD, --pollperiod POLLPERIOD Polling period in seconds.
default=0.01.
```

```
-pl POLLLAG, --polllag POLLLAG Max allowed polling lag in seconds.
default=0.01.

-sp STEPPERIOD, --stepperiod STEPPERIOD Correct step period in
seconds. default=1.0.

-mj MAXJITTER, --maxjitter MAXJITTER Total max jitter allowed in
seconds. default=0.1.

-mc MINCONFIDENCE, --minconfidence MINCONFIDENCE The minimum
expected confidence of a jitter being smaller than maxjitter.

default=0.95.

-cj COMPOSITEJITTER, --compositejitter COMPOSITEJITTER A composite
max jitter factor of: polllag + (1.5 * maxjitter) default=1.0

-ht {1,0}, --hardtiming {1,0} 1 = Use hard timing for polling and
jitter.    0 = Use soft timing for polling and jitter.

default=1.

-to TIMEOUT, --timeout TIMEOUT Timeout in seconds for no occured
transitions. default=3.0.

-du DIGITALUPS, --digitalups DIGITALUPS Number of digital up steps.
default=1.

-dd DIGITALDOWNS, --digitaldowns DIGITALDOWNS Number of digital down
steps. default=2.

-an ANALOGMIN, --analogmin ANALOGMIN Analog min v. default=0.

-ax ANALOGMAX, --analogmax ANALOGMAX Analog max v. default=5.

-au ANALOGUPS, --analogups ANALOGUPS Number of analog up steps.
default=10.

-ad ANALOGDOWNS, --analogdowns ANALOGDOWNS Number of analog down
steps. default=20.        -at ANALOGTHRESHOLD,    --analogthreshold
ANALOGTHRESHOLD assertion ratio of target dV. default=0.9.

-as ANALOGSTABILIZE, --analogstabilize ANALOGSTABILIZE Number of
analog stabilization polls. default=3.

-th {0,1}, --testharness {0,1} Test harness trigger state.
default=0.

-na NETADDRESS, --netaddress NETADDRESS IP address of report host.
default=localhost.

-ne NETEXCEPTION, --netexception NETEXCEPTION Host UDP exception
port. default=6969.
```

```
-nh NETHEARTBEAT, --netheartbeat NETHEARTBEAT Host UDP heartbeat
port. default=6970.

-lf LOGFNAME, --logfname LOGFNAME Name of logfile. Appended if
exists. default=edw-mon.log.

-st {internal,loopback,textfile}, --swtesttype
{internal,loopback,textfile} Software Test Type: internal = Generate
signals from system HPET. loopback = Inject HPET signals into HW
loopback. textfile = Read signals from a text file. default = None
(use hardware input)

-sf SWTESTFNAME, --swtestfname SWTESTFNAME Name of file to sample
signals from. default=edw-mon.stf.

-sm SWTESTFMAKER, --swtestfmaker SWTESTFMAKER Don't sample. Create
an input textfile using given params.
```

> **NOTE:** NO default. Existing file will be automatically overwritten!

```
-sh SWTESTHARP, --swtestharp SWTESTHARP Software Test Harness
Probability. <num> to specify poll's probability to generate a
harness exception. 0 for continuous valid harness input (default).
```

All time values are given in fractional seconds. LabJack's I/O:

```
FIO4=DigitalIn, FIO5=DigitalOut(LB); AIN1=AnalogIn, DAC1=AnalogOut
(LB); FIO6=HarnessIn, FIO7=HarnessOut(LB); DAC0=RelayOut
```

Appendix B – Examples of warnings and exceptions

Each example is taken from a fresh run with the same parameters (except for the Poll Lag example).

All these messages are written to the log-file and sent in UDP. The log-file contains the run information as well as other messages and failures including “error sending exception”, so inspecting the log-file is essential for verifying the validity of the tests assertions.

1. Excessive Jitter (detection)

   The upcoming analog transition from 2.50v to 2.25v should occur next at 20.0s, but was delayed until after 20.1s. The monitor measures the jitter (110ms) and asserts that it’s larger than the maximum given (100ms). However, since it’s only the first fault out of 20 “good” transitions, it still passes the confidence threshold, and therefore just gives the fault info WITHOUT triggering any exception.

```
Fri Jan 29 10:10:05 2016 edw-mon v2.01_2901160438: Excessive
Jitter detected (1/20): Analog jitter of 0.110000s totalling
0.110000s, when transitioning from 2.500000v to 2.250000v took
1.110000s which is farther than max 1.000000s +- 0.100000s
```

2. Excessive Jitter Exception

   This scenario begins exactly like the previous one.

   ```
   Fri Jan 29 13:12:55 2016 edw-mon v2.02_2901161302: Excessive
   Jitter detected (1/20): Analog jitter of 0.110000s totalling
   0.110000s, when transitioning from 2.500000v to 2.250000v took
   1.110000s which is farther than max 1.000000s +- 0.100000s
   ```

   Later, rather than transitioning from 2.25v to 2.00v at 21.0s, it does so much earlier. This time the “good to-bad” ratio (2/21, approx. 90.5%) is under the specified (95%) confidence, which causes the monitor to trigger an “Excessive Jitter Exception”.

   ```
   Fri Jan 29 13:12:56 2016 edw-mon v2.02_2901161302: EXCEPTION!
   Max Jitter Confidence is now: 90.476190% which is under the
   minimum: 95.000000% after Excessive Jitter detected (2/21):
   Analog jitter of -0.210000s totalling 0.100000s, when
   transitioning from 2.250000v to 2.000000v took 0.790000s which
   is farther than max 1.000000s +- 0.100000s
   ```

3. Composite Jitter Exception

   The upcoming digital transition from 0 to 1 should occur next at 30.0s, but actually comes much earlier. The monitor measures the jitter (190ms) and asserts that it’s larger than the allowed (factored) composite jitter (160ms). It gives the fault info and triggers a “Composite Jitter Exception”.

   ```
   Fri Jan 29 09:27:48 2016 edw-mon v2.01_2901160438: EXCEPTION!
   Factored Composite Jitter (0.160000) is under an Excessive
   Jitter detected (1/20): Digital jitter of -0.190000s totalling
   -0.190000s, when transitioning from 0.000000v to 1.000000v took
   1.810000s which is farther than max 2.000000s +- 0.100000s
   ```

4. Test Harness Exception

   The “harness” is a Boolean state read from a digital pin connected to a vendor-supplied circuitry that encapsulates the required validation mechanisms for the “special” outputs. It was set to be “normally high”. While monitoring, the test harness state was flipped (to 0), meaning that “something got internally wrong on the special inputs”, causing the monitor to trigger the “Test Harness Exception”.

   ```
   Fri Jan 29 13:44:32 2016 edw-mon v2.03_2901161336: EXCEPTION!
   Test harness state = 0.0
   ```

5. Polling Lag Exception

   This time, the polling lag (“measurement jitter”) was deliberately reduced from the default 10ms to 10us using the –pl option. At some point, the polling took an extra 12us, which was over the allowed lag of 10us, causing the monitor to trigger the “Poll Lag Exception”.

   ```
   Fri Jan 29 14:10:58 2016 edw-mon v2.03_2901161336: POLL LAG
   EXCEPTION: Monitor Polling took 0.010012s which lagged over max
   0.010010s
   ```