

System transfer speed test

Application Note

Author(s): Teledyne SP Devices
Document ID: 19-2245
Classification: Open
Revision: A
Print date: 2019-03-01

Contents

1 Introduction	2
1.1 Definitions and Abbreviations	2
2 What to expect	2
2.1 Performance	2
3 Code overview	4
3.1 Digitizer mode of operation	4
3.2 Code parts	4
3.2.1 Source files	4
3.2.2 Flow	4
3.3 Tools	4
4 Running the application	5
4.1 Configurable parameters	5
4.2 Test case setup report	7
4.2.1 Default setup	7
4.2.2 Configured setup	8
4.3 Reporting during run	9
4.3.1 Status flow	9
4.3.2 Periodic report	9
4.3.3 Indications	9
4.4 Interpretation of final results	11
4.4.1 Passed test	11
4.4.2 Failed test	12

Document History

Revision	Date	Section	Description	Author
PA1	2019-02-25	-	Initial revision	SA
A	2019-02-25	-	Released revision	SA

1 Introduction

This document describes an application intended for testing the transfer link between the host PC and the digitizer with respect to achievable and sustainable performance. It can also act as a code example for guidance on how to write an application for triggered streaming acquisitions with the digitizer. The code is written for the ADQ14 digitizer family.

1.1 Definitions and Abbreviations

Table 1 lists the definitions and abbreviations used in this document.

Table 1: Definitions and abbreviations used in this document.

Item	Description
Triggered Streaming	The mode of acquisition used in the application
PXIe	PXI Express form factor (chassis)
PCIe	PCI Express form factor (PC-based)
USB	USB form factor (USB3 or USB2)
xNgM	PCIe enumeration result (N lanes, generation M)
std out	Standard output (to the console)
std in	Standard input (from the console, typically the user)
MB	MegaByte (1,048,576 bytes)

2 What to expect

2.1 Performance

The performance is limited in different steps of the path from the digitizer to the host PC.

- **Interface specification** Limits the maximum capacity of link of the raw number of transferred bytes.
- **Host PC infrastructure limitations** Can the host PC slot used support the correct number of lanes and link lane speeds intended? Are there any bottlenecks when using the board further up the infrastructure where bandwidth may be shared with other units?
- **Host PC CPU and memory bandwidth limitations** The parsing of records relies on the CPU to interpret and perform copying of the data to the user space. The memory bandwidth of the host system must support the intended transfer performance.
- **Data overhead** There is a difference between effective transfer rate and the raw number of transferred bytes. For a record from the digitizer there are record headers, and the internal packet generation also adds some overhead.

- **Application overhead** The application can be written in a thousand different ways, providing different levels of performance.
- **Enumeration** The enumeration result depends on the host system capabilities. The ADQ14 (in form factors PXIe/PXIe) supports up to x8g2 and ADQ14-FWDT (in form factor PCIe) supports up to x8g3. ADQ7 supports up to x8g3. Check that the used slot supports the intended enumeration of the board.
- **Digitizer transfer settings** The host PC uses transfer buffers to get data from the digitizer. These are configure (number of buffers and size of buffers) through the API SetTransferBuffers. Higher transfer speeds will require higher number and higher size for these buffers. The buffers are allocated in *contiguous* kernel memory of the host.

Table 2: Maximum interface performance.

Interface	Enumeration	Maximum transfer speed
PXIe/PCIe	x1g1	200 MB/s
PXIe/PCIe	x4g1	800 MB/s
PXIe/PCIe	x4g2	1600 MB/s
PXIe/PCIe	x8g1	1600 MB/s
PXIe/PCIe	x8g2	3200 MB/s
PXIe/PCIe	x8g3	6400 MB/s
MTCA	x4g2	1600 MB/s
MTCA	x4g3	3200 MB/s
USB	USB2	25 MB/s
USB	USB3	300 MB/s
10GbE	10GbE	800 MB/s

3 Code overview

3.1 Digitizer mode of operation

Table 3: Used digitizer mode

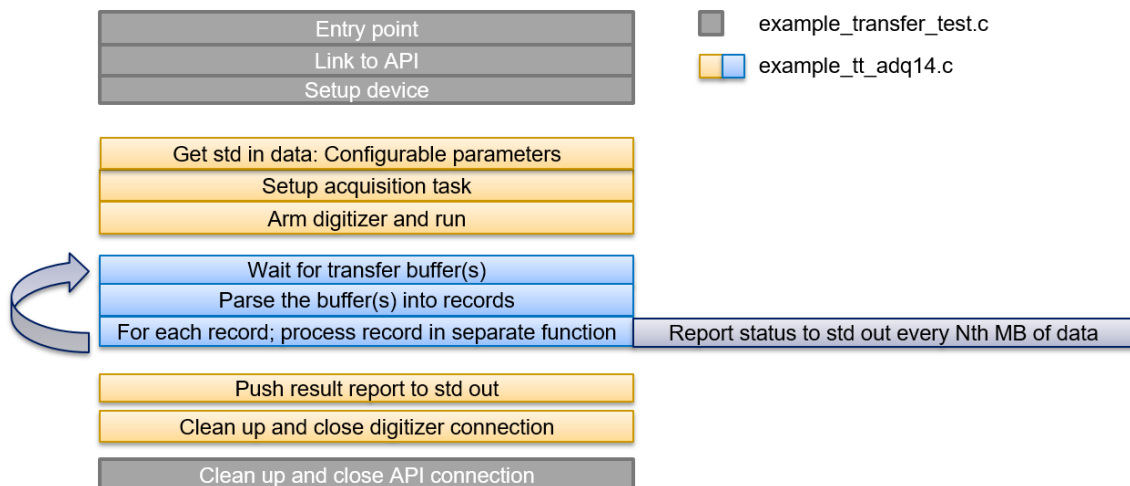
Category	Used mode
Acquisition mode	Triggered Streaming
Trigger mode	Internal trigger (periodic trigger)

3.2 Code parts

3.2.1 Source files

- example_transfer_test.c - Holding the main() entry point and the ADQAPI connection setup.
- example_tt_adq14.c - The code performing the test.
- ADQAPI.h - The header file for the installed API (contains all API commands)

3.2.2 Flow



3.3 Tools

The application is written in the programming language C and requires an appropriate C compiler to be used. In Windows it can be used with for instance Visual Studio from Microsoft and in Linux it is typically compiled with the gcc components.

Note

Make sure that you have selected the appropriate 32-bit/64-bit target selection for the OS you intend to run the application on.

4 Running the application

4.1 Configurable parameters

There are a number of parameters that can be chosen to provide the test you desire to perform. These are chosen run-time by the user in the console.

Tip

These parameters can also be inserted automatically with redirecting a file into standard input. There is a file `params.txt` in the example directory that contains the default run-time parameters. The example is run with taking these parameters from the file by issuing `"ADQAPI_transfer_test_example.exe < params.txt"` from the command window

- **Configuration Mode** This sets the way to configure the test. 0 will just run with default settings, 1 will enable test configuration and 2 will enable test configuration and the advanced settings.
- **Parsing Mode** This sets the mode of parsing. 1 is full parsed mode (DEFAULT), 2 is non-user parse mode and 3 is raw mode. Full parsed mode generates records to the user, the other two does not parse into records. Raw mode can be used together with offline parsing methods. Mode 3 will have the highest performance, mode 2 slightly less and mode 1 the least performance - due to the host PC load inferred by the different modes.
- **Desired run time** This sets the number of seconds for the test to run. 0 is a 30 second run (DEFAULT). Using -1 will make it an infinite run. However, infinite runs will not generate any final reports, as when interrupted it will break immediately.
- **Desired transfer speed to test.** Sets the test data generation speed in MB/s. 1024 MB/s (DEFAULT). The test will auto-configure from this parameter and the trigger rate to select a record size and amount of channels that will produce the correct amount of data.
- **Desired trigger rate to use.** Sets the trigger rate in Hz to use. 1 kHz (DEFAULT). The test will auto-configure from this parameter and the desired transfer test speed to select a record size and amount of channels that will produce the correct amount of data.

Note

A higher trigger rate will use shorter records to provide the amount of data generation requested. As a consequence this will create more records and thus also more data overhead.

- **Number of kernel buffers.** [Advanced] Sets the number of kernel buffers to use. 32 is DEFAULT.
- **Size of kernel buffers.** [Advanced] Sets the size of kernel buffers to use for allocation (in MBytes). 4 MB is DEFAULT.

⚠ Warning

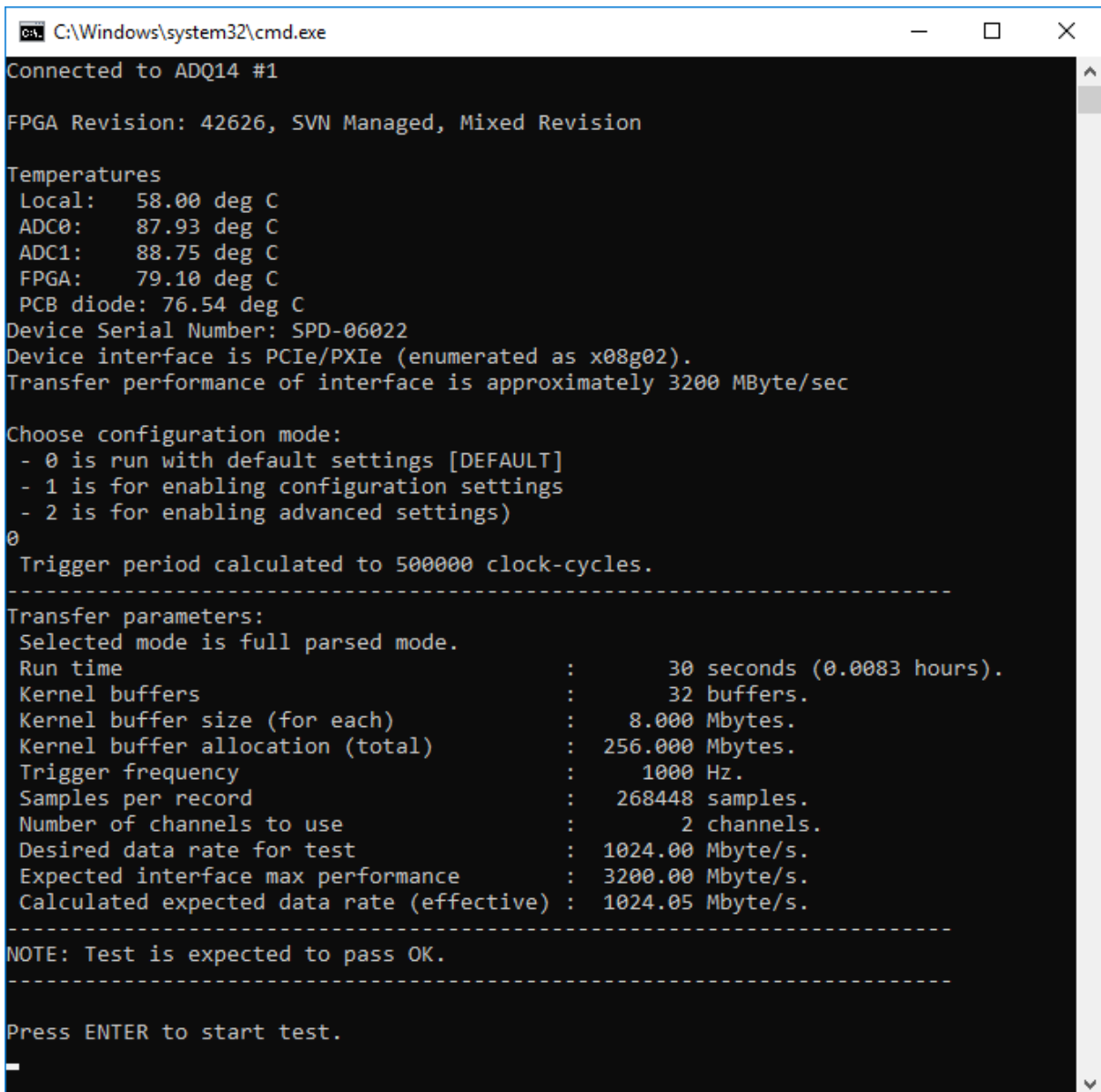
On Linux systems contiguous memory area may be very limited by the system configuration. On some Linux versions CMA settings may be changed to support higher allocations.

- **Periodic report interval.** [Advanced] Sets the amount of data in Mbytes transferred for the periodic report to be output. 1024 MB is DEFAULT.

4.2 Test case setup report

The test auto-configures from the set parameters and reports back the chosen settings and some validation and warnings are provided for checking that the test case is reasonable.

4.2.1 Default setup



```
C:\Windows\system32\cmd.exe
Connected to ADQ14 #1
FPGA Revision: 42626, SVN Managed, Mixed Revision

Temperatures
Local: 58.00 deg C
ADC0: 87.93 deg C
ADC1: 88.75 deg C
FPGA: 79.10 deg C
PCB diode: 76.54 deg C
Device Serial Number: SPD-06022
Device interface is PCIe/PXIE (enumerated as x08g02).
Transfer performance of interface is approximately 3200 MByte/sec

Choose configuration mode:
- 0 is run with default settings [DEFAULT]
- 1 is for enabling configuration settings
- 2 is for enabling advanced settings)
0
Trigger period calculated to 500000 clock-cycles.
-----
Transfer parameters:
Selected mode is full parsed mode.
Run time : 30 seconds (0.0083 hours).
Kernel buffers : 32 buffers.
Kernel buffer size (for each) : 8.000 Mbytes.
Kernel buffer allocation (total) : 256.000 Mbytes.
Trigger frequency : 1000 Hz.
Samples per record : 268448 samples.
Number of channels to use : 2 channels.
Desired data rate for test : 1024.00 Mbyte/s.
Expected interface max performance : 3200.00 Mbyte/s.
Calculated expected data rate (effective) : 1024.05 Mbyte/s.
-----
NOTE: Test is expected to pass OK.
-----
Press ENTER to start test.
```

Here is what it looks like using the default configured test.

The application will make a rough self-assessment of the expected capability vs the configured test case and report whether it expects the test to pass OK or not. If everything looks OK, press ENTER to start the test.

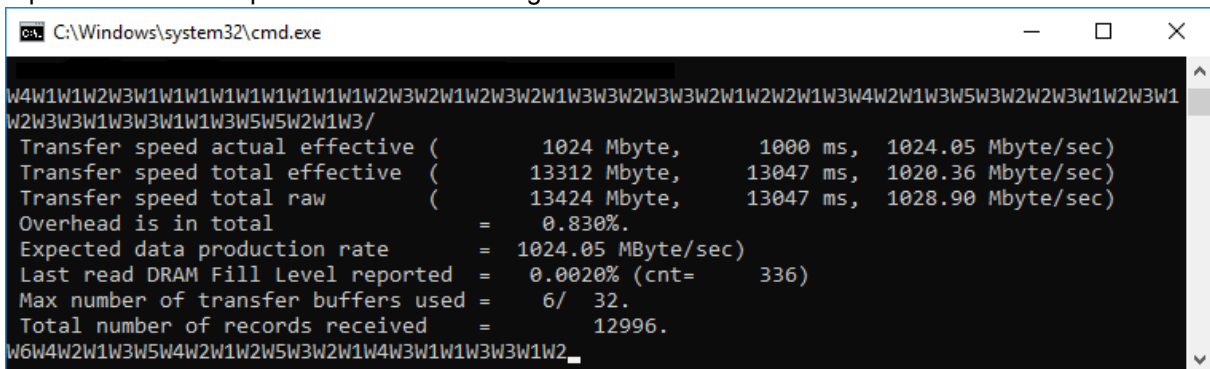
4.3 Reporting during run

4.3.1 Status flow

A 'W' character is written each time the application calls waitingfortransferbuffers. For each time there is more than zero buffers, that number is printed. If it exceeds 9 buffers, a 'N' character is printed. This repeats for the extent of the test, these printouts are only interrupted by any periodic reports or indications produced.

4.3.2 Periodic report

For every **PRINT_EVERY_N_MBYTES_DEFAULT** MBytes passed (default is 1024 MB) an intermediate report on results are produced. Can be configured to other value in advanced mode. It looks like this:



```

C:\Windows\system32\cmd.exe
W4W1W1W2W3W1W1W1W1W1W1W1W2W3W2W1W2W3W2W1W3W3W2W3W3W2W1W2W2W1W3W4W2W1W3W5W3W2W2W3W1W2W3W1
W2W3W3W1W3W3W1W1W3W5W5W2W1W3/
Transfer speed actual effective (      1024 Mbyte,      1000 ms,  1024.05 Mbyte/sec)
Transfer speed total effective (     13312 Mbyte,     13047 ms,  1020.36 Mbyte/sec)
Transfer speed total raw       (     13424 Mbyte,     13047 ms,  1028.90 Mbyte/sec)
Overhead is in total          =         0.830%.
Expected data production rate =    1024.05 MByte/sec)
Last read DRAM Fill Level reported =  0.0020% (cnt=   336)
Max number of transfer buffers used =   6/  32.
Total number of records received =  12996.
W6W4W2W1W3W5W4W2W1W2W5W3W2W1W4W3W1W1W3W3W1W2_

```

4.3.3 Indications

- **Streaming overflow detected**

[ERROR] Streaming overflow detected.

This means that there has been an overflow in the system and data has been lost. The test will terminate as unsuccessful and produce a final report.

- **Maximum kernel buffer fill level detected**

[WARNING] Maximum buffer fill level detected.

This means that kernel buffers are exhausted. Overflow is imminent to occur.

- **High kernel buffer fill level**

[WARNING] High buffer fill level detected.

This means that kernel buffers are on the limit of getting exhausted. All data is currently OK but this is an indication that long-time operation may not be robust with these settings.

- **High DRAM buffer usage**

[WARNING] High DRAM usage.

This means that FPGA DRAM buffers are used to more than 75% . This is an indication of that application is not emptying the transfer buffers fast enough and an overflow may be the result soon.

- **Significant DRAM buffer usage**

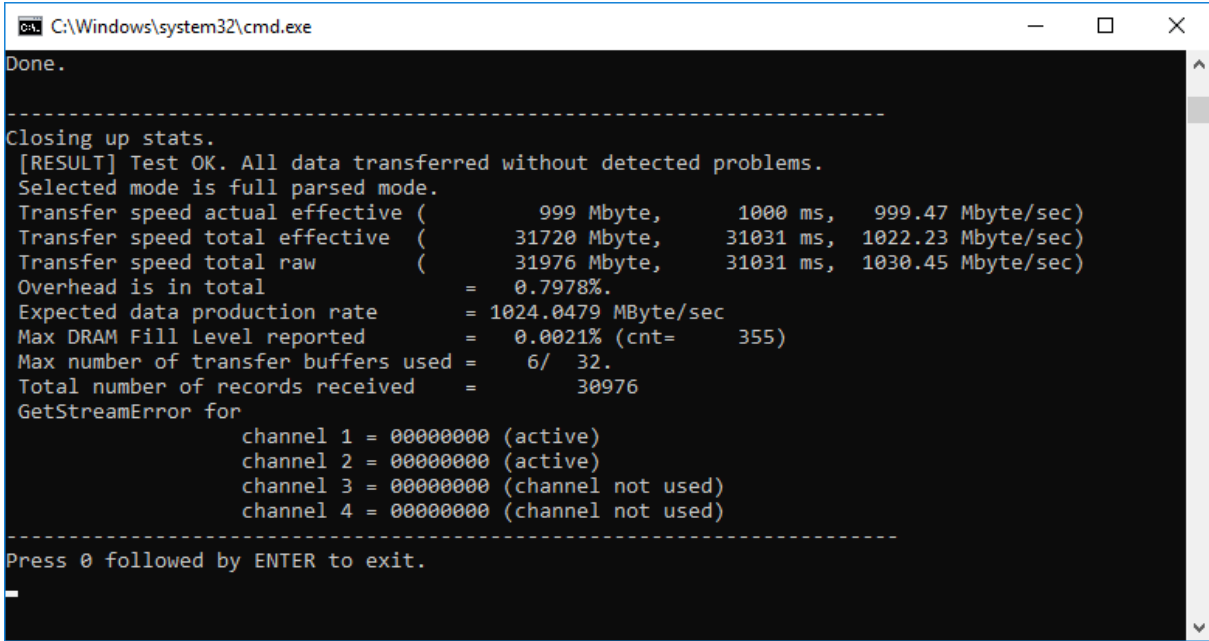
[WARNING] Significant DRAM usage.

This means that FPGA DRAM buffers are used to more than 25%. This is an early indication of possible robustness issues - host not emptying buffers fast enough, but if they stay at this level there is no problem.

4.4 Interpretation of final results

Before the final report is issued the application will read out the maximum DRAM fill level during the test.

4.4.1 Passed test



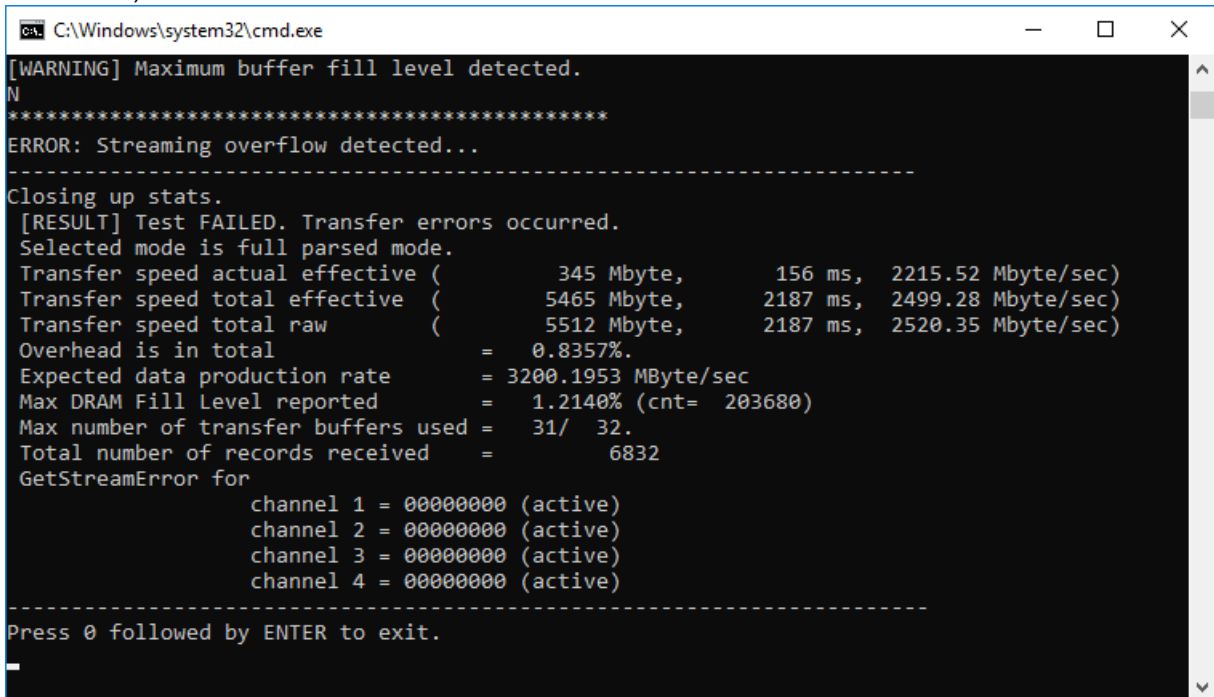
```
C:\Windows\system32\cmd.exe
Done.
-----
Closing up stats.
[RESULT] Test OK. All data transferred without detected problems.
Selected mode is full parsed mode.
Transfer speed actual effective (      999 Mbyte,      1000 ms,    999.47 Mbyte/sec)
Transfer speed total effective (     31720 Mbyte,     31031 ms,   1022.23 Mbyte/sec)
Transfer speed total raw       (     31976 Mbyte,     31031 ms,   1030.45 Mbyte/sec)
Overhead is in total          =    0.7978%.
Expected data production rate = 1024.0479 MByte/sec
Max DRAM Fill Level reported  =    0.0021% (cnt=    355)
Max number of transfer buffers used =    6/ 32.
Total number of records received =    30976
GetStreamError for
    channel 1 = 00000000 (active)
    channel 2 = 00000000 (active)
    channel 3 = 00000000 (channel not used)
    channel 4 = 00000000 (channel not used)
-----
Press 0 followed by ENTER to exit.
_
```

The passed result is shown by the line

```
[RESULT] Test OK. All data transferred without detected problems.
```

4.4.2 Failed test

In this case we select, by intent, a configuration that will fail (too high transfer rate for the interface and host controller).



```
C:\Windows\system32\cmd.exe
[WARNING] Maximum buffer fill level detected.
N
*****
ERROR: Streaming overflow detected...
-----
Closing up stats.
[RESULT] Test FAILED. Transfer errors occurred.
Selected mode is full parsed mode.
Transfer speed actual effective (      345 Mbyte,      156 ms, 2215.52 Mbyte/sec)
Transfer speed total effective (     5465 Mbyte,     2187 ms, 2499.28 Mbyte/sec)
Transfer speed total raw       (     5512 Mbyte,     2187 ms, 2520.35 Mbyte/sec)
Overhead is in total          = 0.8357%.
Expected data production rate = 3200.1953 MByte/sec
Max DRAM Fill Level reported  = 1.2140% (cnt= 203680)
Max number of transfer buffers used = 31/ 32.
Total number of records received = 6832
GetStreamError for
      channel 1 = 00000000 (active)
      channel 2 = 00000000 (active)
      channel 3 = 00000000 (active)
      channel 4 = 00000000 (active)
-----
Press 0 followed by ENTER to exit.
```

The failed result is shown by the line

[RESULT] Test FAILED. Transfer errors occurred.

Worldwide Sales and Technical Support

www.teledyne-spdevices.com

Teledyne SP Devices Corporate Headquarters

Teknikringen 6

SE-583 30 Linköping

Sweden

Phone: +46 (0)13 645 0600

Fax: +46 (0)13 991 3044

Email: spd_info@teledyne.com

Copyright © 2019 Teledyne Signal Processing Devices Sweden AB

All rights reserved, including those to reproduce this publication or parts thereof in any form without permission in writing from Teledyne SP Devices.