
AVR32701: AVR32AP7 USB Performance

Features

- Linux USB bulk transfer performance
 - ATSTK1000 (32-bit SDRAM bus width)
 - ATNGW100 (16-bit SDRAM bus width)
 - GadgetFS driver and gadgetfs-test application
 - USB performance measurement tool

1 Introduction

This application note documents the USB bulk transfer performance of the ATSTK[®]1000 and ATNGW100 boards running a 2.6.22-amtel.3 Linux[®] kernel. The tests are based on the Linux USB gadget driver framework.



32-bit **AVR[®]**
Microcontrollers

Application Note

Rev. 32069A-AVR32-01/08





2 Benchmarking tools

2.1 Host side

On the host side a Linux application communicates with the USB driver over the libusb library. It scans all enumerated USB devices and searches for a matching vendor/product ID. If such devices are found it configures them, writes and reads a specified amount of data to and from the device and measures the time needed for the transfer. These measurements are used to document the USB bulk transfer performance. This tool is available with this application note.

2.2 Device side

To avoid kernel programming, a user space interface is available, GadgetFS. This will transfer all requests not handled by the kernel to user space and thus allowing a developer easy access to the USB interface.

To provide this user space API a special gadget driver is needed. This is the GadgetFS driver. After loading this gadget into the kernel it provides a file system for the developer that can be used to communicate with the USB system.

1. Load the GadgetFS driver
> modprobe gadgetfs
2. Make a directory for the mountpoint
> mkdir /dev/gadget
3. Mount the gadget file system to the mount point
> mount -t gadgetfs none /dev/gadget

After these steps the file `atmel_usba_udc` should appear in the directory `/dev/gadget/`. This file is the entry point for a user application.

The user-space test program is based on the `gadgetfs-test` application from <http://www.linux-usb.org>. It configures two bulk endpoints (IN and OUT) on the device, sinks all incoming data and provides data for the host. If compiled with the library `libaio` the application is able to access the gadget driver asynchronous and thus speed up the communication. These tools are also available in the buildroot board support package (<http://www.atmel.no/buildroot/>).

To measure the pure throughput some buffer checking has been disabled in this application.

3 Benchmark data

3.1 Test setup

The performance tests were done on a host with a Linux operating system with following hardware features:

- RAM: 2GiByte
- CPU: Intel(R) Core(TM)2 [T5600@1.83GHz](#)
- USB Controller: Intel Corporation 82801G (ICH7 Family) USB2 EHCI Controller

To start a measurement run the test application with:

```
> usbperf
```

The USB bulk transfer performance tests on the boards were done with following setup:

- GadgetFS driver
- gadgetfs-test application
- usbperf application
- Transferred amount of data 32MiB
- Maximum bulk packet size 512 bytes

To test the performance gain of asynchronous against synchronous I/O all test were done in both configurations.

Start the gadgetfs-test application without any argument to configure the device with the two bulk endpoints and synchronous access to the driver.

> gadgetfs-test

To run tests with asynchronous access start the application with the parameters -i and -o. Make sure you have compiled the program with libaio support.

> gadgetfs-test -i20 -o5

-i: Length of the asynchronous IN queue length

-o: Length of the asynchronous OUT queue length

The IN and OUT queue lengths can be used to optimize the transfers.

3.2 Test results

Table 3-1 and Table 3-2 document the received test results on the STK1000 and NGW100 development boards.

Table 3-1. STK1000 USB performance

| Test case | Read transfer [MiB/s] | Write transfer [MiB/s] |
|--|-----------------------|------------------------|
| Synchronous access to the GadgetFS driver | 14 | 14 |
| Asynchronous access to the GadgetFS driver | 22 | 15 |

Table 3-2. NGW100 USB performance

| Test case | Read transfer [MiB/s] | Write transfer [MiB/s] |
|--|-----------------------|------------------------|
| Synchronous access to the GadgetFS driver | 14 | 14 |
| Asynchronous access to the GadgetFS driver | 19 | 15 |

3.3 Summary

The test results on the two different boards do not differ very much except for the asynchronous read access case. Surprisingly it seems that the different bus widths to the external memory (16bit on NGW100, 32bit on STK1000) of the boards are not





directly visible in most cases. The use of asynchronous I/O seems to speed up the performance especially for read transfers.

Better performance can be reached with a real kernel driver. For instance the gadget zero driver from linux-usb.org reaches up to 25MiB/s on read transfers.

4 References

GadgetFS driver, gadgetfs-test application: <http://www.linux-usb.org>

libusb: <http://www.libusb.sourceforge.net>

libaio: <http://www.lse.sourceforge.net/io/aio.html>

buildroot: <http://www.atmel.no/buildroot>



Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr32@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2008 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, AVR® and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.