

GM8126

BURN-IN TEST

User Guide

Rev.: 1.1

Issue Date: June 2011



REVISION HISTORY

GM8126 Burn-in Test User Guide

Date	Rev.	From	To
Dec. 2010	1.0	-	Original
Jun. 2011	1.1	-	<ul style="list-style-type: none">Added Sections 1.5 and 2.13Modified Sections 2.10 and 2.18Modified Figure 2-2, Figure 2-11, and Figure 2-12

Copyright © 2011 Grain Media, Inc.

All Rights Reserved.

Printed in Taiwan 2011

Grain Media and the Grain Media Logo are trademarks of Grain Media, Inc. in Taiwan and/or other countries. Other company, product and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation or other life support application where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Grain Media's product specification or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Grain Media or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will Grain Media be liable for damages arising directly or indirectly from any use of the information contained in this document.

Grain Media, Inc.
5F, No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Grain Media's home page can be found at:
<http://www.grain-media.com>

TABLE OF CONTENTS

Chapter 1	Introduction.....	1
	1.1 Version of the IP	2
	1.2 Host System Environment.....	2
	1.3 Target System Environment.....	3
	1.4 Archive Burn-in Program	3
	1.5 Size-down Function	3
Chapter 2	Burn-in Test.....	5
	2.1 Setting-up Environment.....	7
	2.2 DDR Test.....	9
	2.3 Timer Test	10
	2.4 DMA Test	12
	2.5 Watchdog Test	13
	2.6 Watchdog Reset Test.....	14
	2.7 I ² C Test.....	14
	2.8 SD Test.....	15
	2.9 Enable/Disable Cache.....	16
	2.10 Run U-Boot/Linux	17
	2.11 GM8126 QCtest 1 Auto and GM8126 QCtest 2 Auto	18
	2.12 FCS Test	18
	2.13 MCP Clock Setting	20
	2.14 USB OTG 2.0 Test	20
	2.14.1 Environment Setting	23
	2.14.2 Peripheral Mode Test Reference.....	25
	2.14.3 Host Mode Test	36
	2.14.4 Role Selection	50
	2.15 Burn U-BOOT Image Function.....	50
	2.16 Burn Linux Image Function	51

	2.17	Run Linux from SD Card	52
	2.18	Address Read/Write Test	53
	2.19	Security Test	54
	2.20	UART Test.....	55
	2.21	Burn Boot Code Function	55
	2.22	NAND Flash Test	57
	2.23	VGA Setting.....	58
	2.24	LCD Test and Burn Logo Image	59
	2.25	Keyscan Test Only for 8126.....	61
	2.26	IRDET Test Only for 8126.....	62
Chapter 3		Compile GM8126 Burn-in Program.....	63
	3.1	ROM Version	64
	3.2	RAM Version	65
	3.3	ICE Options/Configure Processor	66
Chapter 4		References.....	67

LIST OF FIGURES

Figure 2-1.	Factory EV Board	7
Figure 2-2.	Starting Screen of Burn-in Test Program	8
Figure 2-3.	Example of DDR Test	9
Figure 2-4.	Timer Test	11
Figure 2-5.	DMA Test	12
Figure 2-6.	Watchdog Test	13
Figure 2-7.	Watchdog Reset Test.....	14
Figure 2-8.	I ² C Test	14
Figure 2-9.	SD Test	16
Figure 2-10.	Cache Operation	16
Figure 2-11.	U-BOOT or Linux Operation.....	17
Figure 2-12.	FCS Test	19
Figure 2-13.	USB 2.0 PC Screen	21
Figure 2-14.	USB 2.0 Test Bench (1)	22
Figure 2-15.	USB 2.0 Test Bench (2)	22
Figure 2-16.	USB 2.0 Test Bench (3)	23
Figure 3-1.	ROM Version.....	65
Figure 3-2.	RAM Version	66

Chapter 1

Introduction

This chapter contains the following sections:

- 1.1 Version of the IP
- 1.2 Host System Environment
- 1.3 Target System Environment
- 1.4 Archive Burn-in Program
- 1.5 Size-down Function

This document introduces the test items and test procedures for the GM8126 burn-in program. The test program includes the test items that are used to verify the functions of most IPs. Users can issue commands to invoke individual test program and specify how many cycles the program should run. The following sections introduce the targets and the host environments. The testing items will be separately addressed in Chapter 2.

1.1 Version of the IP

This is the preliminary version of the IP.

1.2 Host System Environment

The host system environment is comprised of the following items:

- Hardware:
 - Intel x86 compatible PC
 - Cross-wired RS-232 cable
 - JTAG-ICE
- OS:
 - Windows XP/2000/98
- Tool chain:
 - ARM developer suite, v1.2

The ARM developer suite v1.2 and JTAG-ICE are required when debugging the burn-in test program. To compile the source code, please refer to Appendix A for details.

1.3 Target System Environment

The target system environment is comprised of the following items:

- Grain Media ARM-based multimedia platform, GM8126
- 128-MB DDR2 memory
- 8-MB expansion SPI Flash memory or 1-GB NAND Flash
- Grain Media SD Flash card interface

1.4 Archive Burn-in Program

- Unpack the file, \StartCell_Drivers\source\GM8126_NonOS.zip, to the specified directories
- The unpacked directories include:
 - burnin
 - Source code of the burn-in program
 - fLib
 - Library accessed by the burn-in program
 - \StartCell_Drivers\image\rom.bin
 - This file is the burn-in program that executes the image for the Flash. Users can also find this file, used in the ADS 1.2 environment, in the following directory:
burnin\burnin\burnin_ads_1.2_Data\Rom, when the source code of Flash is rebuilt.
Please refer to Appendix A for details.

1.5 Size-down Function

Users can reduce the image size after the release of the Grain Media EVB. Users can modify the GM8126_NonOS/burnin/code/00_main/00_main.c file as follows:

```
//#define small_size    //debug version, including the debug EVB and performance function
//#define mini_size     //release version, if users finish PP
```

If users enable `small_size`, certain memories and burn functions will be reserved. If users enable `mini_size`, only the burn functions will be reserved. If users want to use both the `small_size` and `mini_size` functions, please confirm that the Grain Media EVB functions are set.

Chapter 2

Burn-in Test

This chapter contains the following sections:

- 2.1 Setting-up Environment
- 2.2 DDR Test
- 2.3 Timer Test
- 2.4 DMA Test
- 2.5 Watchdog Test
- 2.6 Watchdog Reset Test
- 2.7 I²C Test
- 2.8 SD Test
- 2.9 Enable/Disable Cache
- 2.10 Run U-Boot/Linux
- 2.11 GM8126 QCtest 1 Auto and GM8126 QCtest 2 Auto
- 2.12 FCS Test
- 2.13 MCP Clock Setting
- 2.14 USB OTG 2.0 Test
- 2.15 Burn U-BOOT Image Function

- 2.16 Burn Linux Image Function
- 2.17 Run Linux from SD Card
- 2.18 Address Read/Write Test
- 2.19 Security Test
- 2.20 UART Test
- 2.21 Burn Boot Code Function
- 2.22 NAND Flash Test
- 2.23 VGA Setting
- 2.24 LCD Test and Burn Logo Image
- 2.25 Keyscan Test Only for 8126
- 2.26 IRDET Test Only for 8126

2.1 Setting-up Environment

Before starting the burn-in program, the working environment should be properly set. Users may follow the steps below to prepare the working environment for the burn-in test.

Step 1: Use the factory EV board, as shown in Figure 2-1.

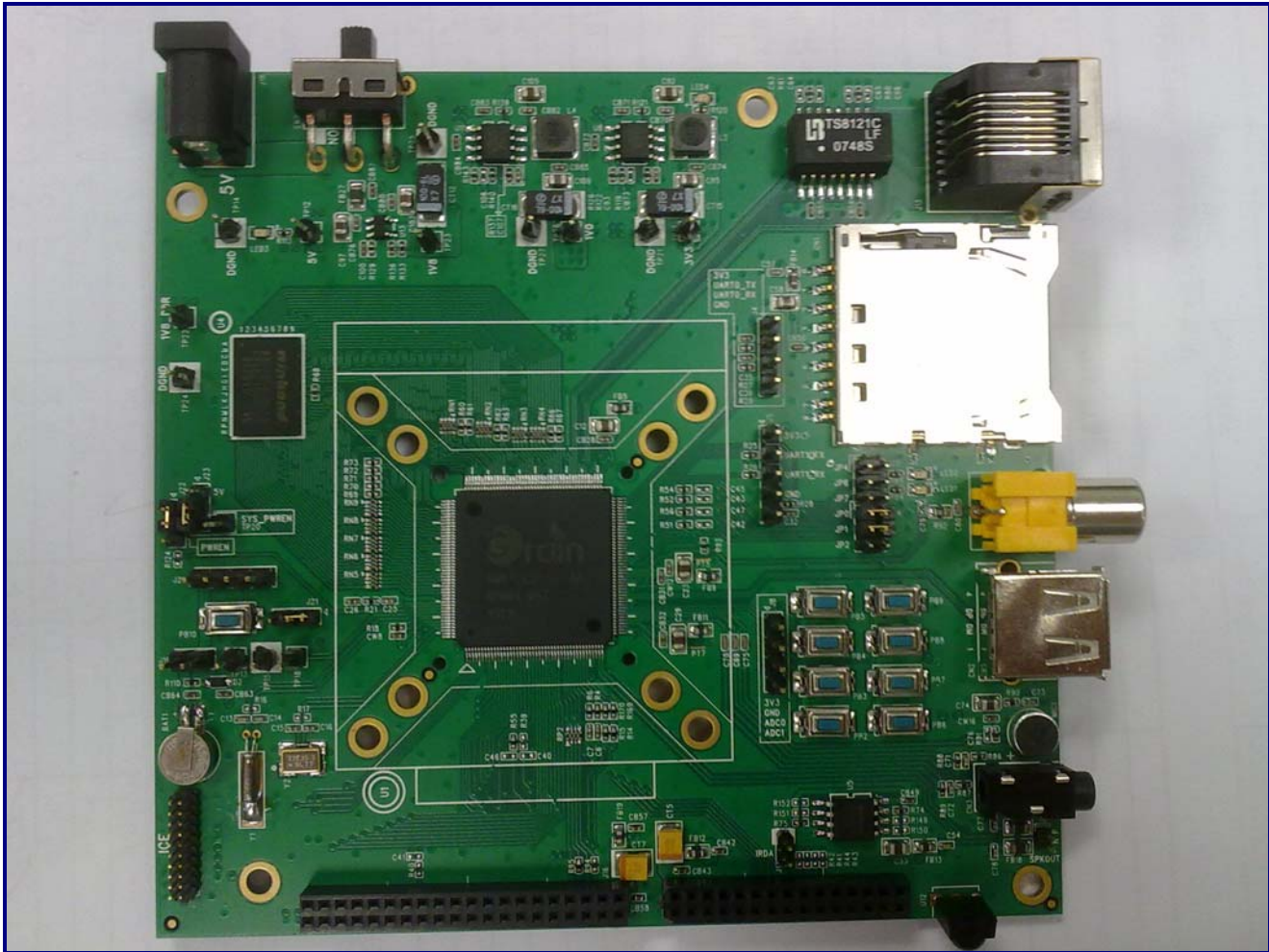


Figure 2-1. Factory EV Board

Step 2: Open HyperTerminal and configure settings to 38400 bps, no parity, 8-bit data length, and one-stop bit

Step 3: Power on the EV board; HyperTerminal will display the messages as shown in Figure 2-2.

```
-----  
-----  
GM Technology Corporation CPE Burnin Program v0.2.4  
Platform: GM81262110 Cache: Disabled  
PLL1: 800 MHz    PLL2: 540 MHz    DDR: 800 MHz  
CPU : 540 MHz    HCLK: 270 MHz    PCLK: 135 MHz  
UART: 1 MHz     H.264: 270 MHz    MPEG4: 200 MHz  
STXWAMODE: RAO=1, STMWA=x  
-----  
( 1) DDR Test      ( 2) Timer Test    ( 3) DMA Test  
( 6) Watchdog Test ( 7) Watchdog Rest Test (10) I2S Module1 Test  
(12) I2C Test     (13) LCD Test      (14) Query RTC  
(15) RTC Alarm Test (16) GPIO Test     (18) SAR ADC  
(22) Enable Cache (23) Disable Cache (24) SD Card Test  
(31) PWM Test     (35) OTG 210 Test  (36) Powerdown Test  
(42) Irdet Test   (45) Security Test (50) MCP Clock Setting  
(51) FCS Test     (52) VGA setting   (53) DVS mode switch  
(54) Burn Logo image (60) Change console's UART(61) I2C Read/Write  
(62) Change Cache WA_Mode (65) GM8126 QCtest 1 Auto (66) GM8126 QCtest 2 Auto  
(67) GM8126 QCtest 3 Auto (68) GM8126 QCtest 4 Auto (69) Load Pattern  
(70) Address Fill Data (71) Address Read (72) Address Write  
(73) Burn UBOOT image (74) Burn Burn-In image (75) Burn Linux image  
(76) Burn Loader image (77) Burn fresh spi flash (78) BootFromSD  
(79) BootFromUSB     (80) UBOOT, Linux  
Command>>
```

Figure 2-2. Starting Screen of Burn-in Test Program

2.2 DDR Test

The size of DDR is 128 MB on the EV board. The DDR test program is used to verify the read/write patterns from address 0x1000000 to address 0x10000000. Different cache enable status will have different DDR memory access timings. The memory access time will be much faster when the cache is enabled. In a practical design, the cache will always be enabled after the system boots up. Users can input '22' or '23' to enable or disable the cache when testing the DDR memory. After switching on/off the cache to '1', the system will prompt users to input the number of operations. Figure 2-3 shows the one-time DDR test. Users can press "ESC" on the keyboard to quit this test item.

```
Command>>1
DDR DDR Test (Press ESC to escape)
Please input running time:
1
Please input refresh type (0)non-staggered (1)staggered:
0
module:0, rank start address:0x0, rank size:128 MB
Begin DDR1 test round 0 from 0x01000000 to 0x06efffff ...
Test DDR_2 from 0x08000000 to 0x10000000 ...
DDR Test Item 1...(Word), pattern is address value
Write DDR...
Verify DDR...
0x6ef0000 Write DDR...
Verify DDR...
rDDR Test Item 2...(Word), pattern is 0x5aa55aa5
Write DDR...
Verify DDR...
0x6ef0000 Write DDR...
Verify DDR...
rDDR Test Item 3...(Word), pattern is 0xa55aa55a
Write DDR...
Verify DDR...
```

Figure 2-3. Example of DDR Test

The DDR test items are listed as follows.

- Test item 1: Use the 32-bit access to verify the DDR memory with all test patterns. The patterns will be the addresses.
- Test item 2: Use the 32-bit access to verify the DDR memory with all test patterns. The pattern will be 0x5aa55aa5.
- Test item 3: Use the 32-bit access to verify the DDR memory with all test patterns. The pattern will be 0xa55aa55a.
- Test item 4: Use the 8-bit access to verify the DDR memory with all test patterns. The patterns will be the addresses.
- Test item 5: Use the 16-bit access to verify the DDR memory with one test pattern. This pattern will be an address.

The system will run from the Flash code. At the system booting stage, the boot loader should set the DDR timing and verify the DDR functions.

2.3 Timer Test

GM8126 contains three timers. The timer source can be PCLK or EXT CLK. The default timer source is PCLK. The test items are used to test the timer features, which include the “down counting”, “match1”, “match2”, “overflow”, “up counting”, and “EXT CLK” features. After typing ‘2’ and pressing “ENTER” on the terminal, the timer test will run, as shown in Figure 2-4. If the test is completed successfully, the screen will show “Pass!”.

```
Command>>2
Begin Timer Test...

first timer test
Timer1 Test...
5 Pass!
Timer2 Test...
5 Pass!
Timer3 test...
5 Pass!
Timer1 Overflow Test...Pass!
Timer2 Overflow Test...Pass!
Timer3 Overflow Test...Pass!
Timer1 Match Test...Pass!
Timer2 Match Test...Pass!
Timer3 Match Test...Pass!
Pass!
```

Figure 2-4. Timer Test

Generally, the operating system (OS) uses a timer for the time slice, task delay, and timer functions. Other timers may be used by the device driver or protocol stack to obtain precise timing. When an application is running on a specific OS, the timer may operate in the longer timer interrupt latency so that the driver may not be able to identify the cause of an interrupt. Therefore, it is recommended that users apply a timer to OS to schedule its MATCH1 register. If more precise timer is needed by the driver, users may use another timer to schedule the MATCH1 register.

2.4 DMA Test

GM8126 contains an AHB DMA. The DMA controller can perform the memory-to-memory, memory-to-peripheral, and peripheral-to-memory transfers. A list of the DMA test items is shown in Figure 2-5.

```
Command>>3
MMU/Cache disabled
DMA Test
  Two AHB masters test
    Channel#1, src: AHB Master 0(LitEnd), dest: AHB Master 0(LitEnd)...pass
    Channel#1, src: AHB Master 0(LitEnd), dest: AHB Master 1(LitEnd)...pass
    Channel#1, src: AHB Master 1(LitEnd), dest: AHB Master 0(LitEnd)...pass
    Channel#1, src: AHB Master 1(LitEnd), dest: AHB Master 1(LitEnd)...pass
    Channel#2, src: AHB Master 0(BigEnd), dest: AHB Master 0(BigEnd)...pass
    Channel#2, src: AHB Master 0(BigEnd), dest: AHB Master 1(BigEnd)...pass
    Channel#2, src: AHB Master 1(BigEnd), dest: AHB Master 0(BigEnd)...pass
    Channel#2, src: AHB Master 1(BigEnd), dest: AHB Master 1(BigEnd)...pass
Pass!
```

Figure 2-5. DMA Test

Because the test program can perform large memory-to-memory transfers in the main memory, it will always disable the cache before testing. In a practical design, the cache should remain enabled even if DMA is activated. The DMA transfer region should be set by the device driver as an I/O region.

2.5 Watchdog Test

This test item is used to test the kicking operations of a watchdog timer. Once the watchdog test is selected, the terminal will prompt users to verify the following clock sources:

- PCLK: This clock source is generated from the Power Memory Unit (PMU). Please refer to the GM8126 data sheet for more detailed information.
- External: This is the 32.768-kHz onboard clock. Please refer to the relevant GM8126 documents for more detailed information.

Figure 2-6 shows the watchdog test by using the PCLK clock source.

```
Command>>6
WatchDog Test..
Please select clock source:(1)PCLK (2)32.768KHz(external)
1
Pass!
```

Figure 2-6. Watchdog Test

The watchdog timer is used to reset the system to prevent the system from hanging up. If the watchdog timer is not reset periodically, it will automatically reset the system or trigger an interrupt to signal the device driver.

2.6 Watchdog Reset Test

This test is used to verify the watchdog reset operations. GM8126 will reboot the system after the watchdog reset test is selected. The test run is shown in Figure 2-7.

```
Command>>7
Watch Dog Reset Test Start
Watchdog Reset has occurred since the last time the CPU or Hardware cleared this bit (Y/N => 1/0):
1 01234567Will set the following freq...
PLL1: 480 MHz, PLL3: 648 MHz, CPU freq: 480 MHz, AHB freq: 240 MHz, DDR freq: 480 MHz
go...
```

Figure 2-7. Watchdog Reset Test

2.7 I²C Test

GM8126 contains a serial EEPROM, which is 24C02N with 2K in size. This test performs the single-byte access and page access. This test item reads and writes EEPROM from the I2C interface. The terminal will prompt a successful message if the test is successfully complete. The test is shown in Figure 2-8.

```
Command>>12
Begin I2C Test...
Write Byte and Verify Loop Test Success!
Write All Bytes and Verify Test Success!
Write Page and Verify Loop Test Success!
Write All Pages and Verify Test Success!
End I2C Test!
```

Figure 2-8. I²C Test

Users should use 61 items to perform the read/write operations to verify an I2C device so that the hardware works (Such as writing one byte to address 0xa8 for the I2C device or to register 0x05 with a byte value of 0x36, and then read to verify the correctness). The number of the input bytes represents the set bytes that users cannot write any more bytes before exiting from this function. The 61 items should be performed again to update the written bytes.

2.8 SD Test

Because GM8126 controls the SD cards, users can insert an SD card to the socket on the EV board to perform this test. After entering the SD test, the terminal will prompt the following two operating modes:

- PIO mode: Programming the SD controller by using the PIO operation
- DMA mode: Programming the SD controller by using the APB DMA operation

If all operations are successful, the message, "SD>>>", will appear on the terminal. Users can perform several file operations, such as "cp" and "dir". Users can type "?" to get more instructions.

The FAT12/FAT16 or FAT32 file system will not support long file names; the file number is limited.

MBR in SD sector 0

The SDHC card is supported.

```
Command>>24
MMU/Cache disabled
Select the mode: 1.PIO 2.AHBDMA
1

...SD Card on Drive C: !!!
Command initial

SD>>>
SD>>>
SD>>>dir
    <command>: dir
(FAT32) Volume Lable: , Volume Serial Number: 00000000
Directory of \, Cluster:2, Sector:15679
  BURNHD1  5320Bytes  -rw 09-17-2009 03:03:25 BURNHD1
  OUTPUT.TXT  6144Bytes  -rw 10-21-2009 09:48:11 OUTPUT.TXT
  GAMMA_~1.SH  6144Bytes  -rw 10-21-2009 10:35:12 GAMMA_~1.SH
  GAMMA_~2.SH  6144Bytes  -rw 10-21-2009 10:35:12 GAMMA_~2.SH
  CONST_~1.SH  6144Bytes  -rw 10-21-2009 13:34:15 CONST_~1.SH
  CONST_~2.SH  6144Bytes  -rw 10-21-2009 13:34:16 CONST_~2.SH
```

```
GAMMA_1.SH 6144Bytes -rw 10-21-2009 13:34:22 GAMMA_1.SH
CONST_~3.SH 6144Bytes -rw 10-21-2009 13:34:29 CONST_~3.SH
CONST_~4.SH 6144Bytes -rw 10-21-2009 13:35:00 CONST_~4.SH
LINUX~~1 11861824Bytes -rw 10-21-2009 19:35:22 LINUX~~1
LINUX~~2 9548772Bytes -rw 10-14-2009 22:18:10 LINUX~~2
LINUX 11861824Bytes -rw 10-21-2009 19:35:22 LINUX
FTSDC010.KO 156102Bytes -rw 10-22-2009 13:45:03 FTSDC010.KO
```

Figure 2-9. SD Test

The system may display an error message when the SD card is tested at a bus clock rate of 25 MHz. Some SD cards may become unstable when operating at the maximum bus speed. Lowering the speed of the SD bus may solve the issue; however, it may degrade the system performance.

2.9 Enable/Disable Cache

The commands shown in Figure 2-10 are used to enable or disable the cache in GM8126. After enabling the cache, the message, "MMU/Cache enabled", will show on the terminal.

```
Command>>22
MMU/Cache enabled

Command>>23
MMU/Cache disabled

Command>>
```

Figure 2-10. Cache Operation

2.10 Run U-Boot/Linux

GM8126 can jump to U-Boot/Linux for GM8126 after issuing the '80' command on the terminal. Users can press "ESC" and decide to run U-BOOT or Linux .The terminal will show as listed in Figure 2-11 if the U-BOOT code or Linux code can be written into Flash.

```
Command>>80

*****

Please input Space to run Linux
Please input ESC to run UBOOT
Please input . to run burn-in
Otherwise, system will run Linux after 5 sec
*****

Load image from SPI-NOR offset 0xa6000 to sdram 0x4000000
Jump 0x4000000

U-Boot 2008.10 (Feb 9 2011 - 17:19:13)

DRAM: 128 MB
Manufacturer ID : 0018
Device ID      : 009F
Device Code 2  : 0018
Flash: 0 kB
*** Warning - bad CRC, using default environment

flash no default environment
In: serial
Out: serial
Err: serial
Net: FTMAC110#0
Hit any key to stop autoboot: 0
=>
```

Figure 2-11. U-BOOT or Linux Operation

2.11 GM8126 QCtest 1 Auto and GM8126 QCtest 2 Auto

These tests are reserved only for the Grain Media internal usage.

2.12 FCS Test

This test is used to change the speeds of CPU, AHB, MPEG codec, and APB clock. Users should enter the cpuxxx mode, then PLL1 multiplier, and PLL1 divider. After entering the information, the system will change the clock speed according to the new configuration for the frequency. If the FCS test is successful, a new frequency setting will be displayed on the main menu. Figure 2-12 shows the FCS test procedure.

Please note that the DDR timing needs to be tuned for better performance. If users want to set to a lower frequency, the DDR timing must be tuned.

For the 8126 series platforms, users are not recommended performing the FCS operation, because of the DDR timing issue.

```
Command>>51
MMU/Cache disabled
When you use this function, you must check if SDRAM timing is correct.
PLL1 = 30 MHz * mul / div
PLL2 = 30 MHz * mul / div
0: CPU mode 0: FCLK = PLL1 / 2, HCLK = PLL1 / 2, DDR = PLL1
1: CPU mode 1: FCLK = PLL1 / 1.5, HCLK = PLL1 / 3, DDR = PLL1
2: FCS0: FCLK = PLL2, HCLK = PLL2 / 2, DDR = PLL1
3: FCS1: FCLK = PLL1 / 1.5, HCLK = PLL1 / 3, DDR = PLL1(async)
4: FCS2: FCLK = PLL1 / 2, HCLK = PLL1 / 4, DDR = PLL1(sync 1:2)
5: expert mode

Enter mode (current val: 2, possible val: 0~5): 2
PLL1 mul (current val: 80, possible val: 1~511): 78
PLL1 div (current val: 3, possible val: 1~31): 3
PLL2 mul (current val: 54, possible val: 1~511): 54
PLL2 div (current val: 3, possible val: 1~31): 3
```

Will set the following freq...

PLL1: 780 MHz, PLL2: 540 MHz, CPU freq: 540 MHz, AHB freq: 270 MHz, DDR freq: 78

0 MHz

go...

Success!!!

Figure 2-12. FCS Test

The system will automatically set to a higher CPU speed after booting.

For the 8128 series platforms, the default speeds are:

- CPU = 594 MHz
- AHB = 297 MHz
- APB = 148 MHz
- JPEG codec = 225 MHz
- H.264 codec encode = 360 MHz
- H.264 codec decode = 360 MHz

For the 8126 series platforms, the default speeds are:

- CPU = 540 MHz
- AHB = 270 MHz
- APB = 135 MHz
- JPEG codec = 200 MHz
- H.264 codec encode = 270 MHz
- H.264 codec decode = 270 MHz

2.13 MCP Clock Setting

The following command is used to change the video codec clock. Users can change the H.264 or MPEG4 video codec clock.

```
Command>>50
MPEG4 clock source(0:Async CLK, 1:HCLK, current val: 0): 0
H.264 clock source(0:Async CLK, 1:HCLK, 2:Div_2.5, current val: 2): 0

-----
GM Technology Corporation CPE Burnin Program v0.2.2
Platform: GM81262001 Cache: Disabled
PLL1: 800 MHz   PLL2: 540 MHz   DDR: 800 MHz
CPU : 540 MHz   HCLK: 270 MHz   PCLK: 135 MHz
UART:  1 MHz   H.264: 200 MHz   MPEG4: 200 MHz
VGA : 1024x768
-----
```

2.14 USB OTG 2.0 Test

Users should following the procedure listed below to install the PC USB driver.

- Perform test item 35 to enter the OTG test environment
- Connect the EV board to PC by using a USB cable
- PC pops up a dialog to request the USB driver.
- Enter the driver path which contains "FaradayUSB.inf" and select "FaradayUSB.Sys(v1.2) Faraday FUSB200 IO Test EV-Board".
- If successful, users can observe "FaradayUSB.Sys (V1.2) Faraday FUSB200 IO Test EV-Board" from the Windows device manager.

USB Test

The following example uses the “bulk and Interrupt” test items and transfers data in the PIO mode. This basic item can help users verify whether the USB device is functioning correctly.

- Perform test item 35 to enter the USB test environment
- Enter <1> for “bulk and interrupt” then enter <1> to activate the PIO mode

After the installation is complete and the USB cable is unplugged, users will see the message shown in Figure 2-13.

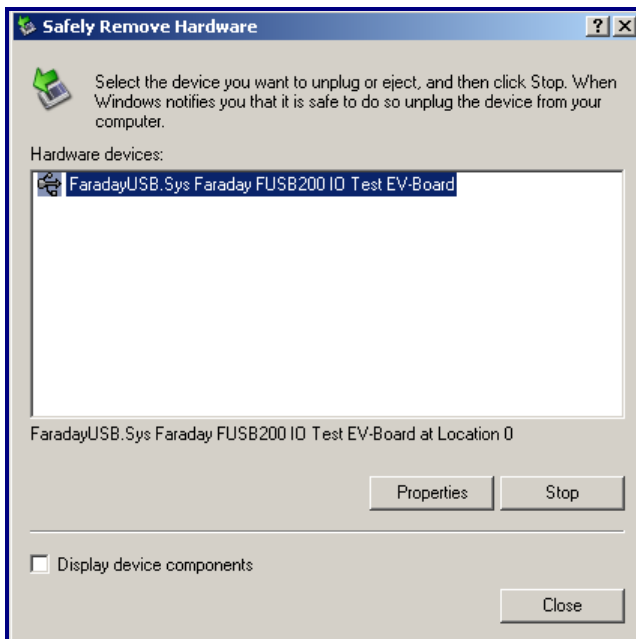


Figure 2-13. USB 2.0 PC Screen

Test Bench

By launching the “USB_Testbench.exe” test bench program in the “USB_tool” directory, users can find the Faraday USB driver, as shown in Figure 2-14. If not, users may encounter a compatibility problem. Please ask the Grain Media service team for help.

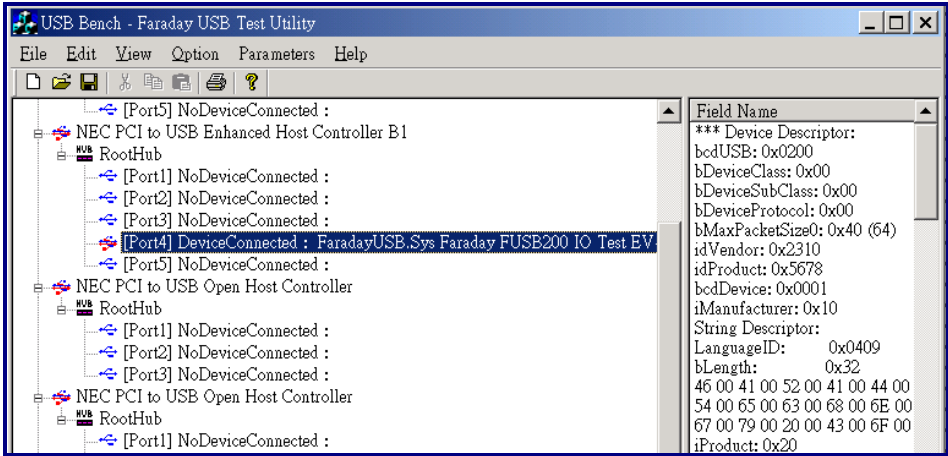


Figure 2-14. USB 2.0 Test Bench (1)

If the operation is successful, users should double-click the device item, "FaradayUSB.Sys Faraday".

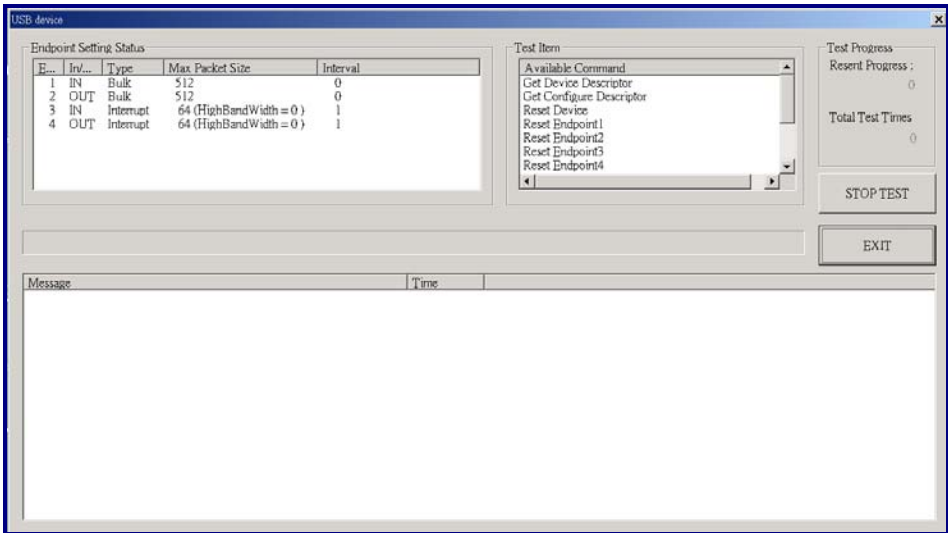
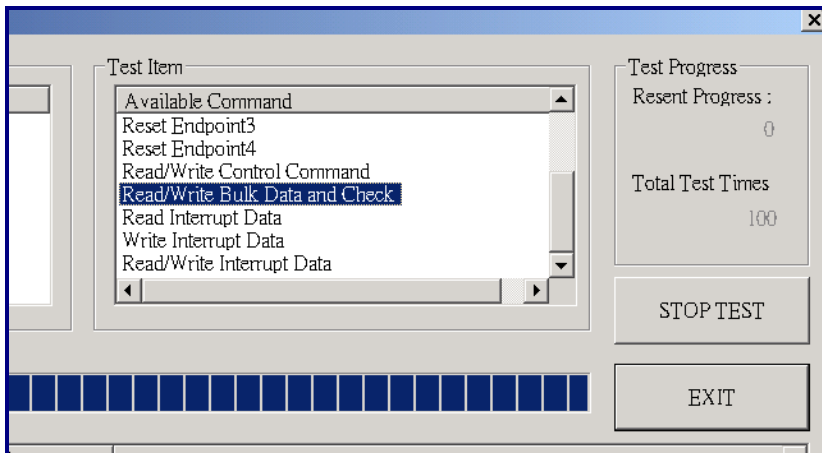


Figure 2-15. USB 2.0 Test Bench (2)

Select the "Read/Write Bulk Data and Check" item to test the USB device. If successful, the message, "Congratulations Test 100 times finish...", will display.



```

~~~~~ Bulk IN/OUT transfer Start ~~~~~ 16:39:02 03...
~~~~~ Congratulations Test 100 times finish ~~~~~ 16:39:04 03...
~~~~~ Bulk IN/OUT transfer Start ~~~~~ 16:39:10 03...
~~~~~ Congratulations Test 100 times finish ~~~~~ 16:39:12 03...

```

Figure 2-16. USB 2.0 Test Bench (3)

After the operation is complete, please press "ESC" to exit. For the compatibility problem of the test bench, please change the PC USB EHCI driver. If the problem still occurs, add a PC USB card, such as NEC or VIA.

2.14.1 Environment Setting

(1) OTG will detect the USB port to be the host or device role.

Host: If linked with the mini-A endpoint

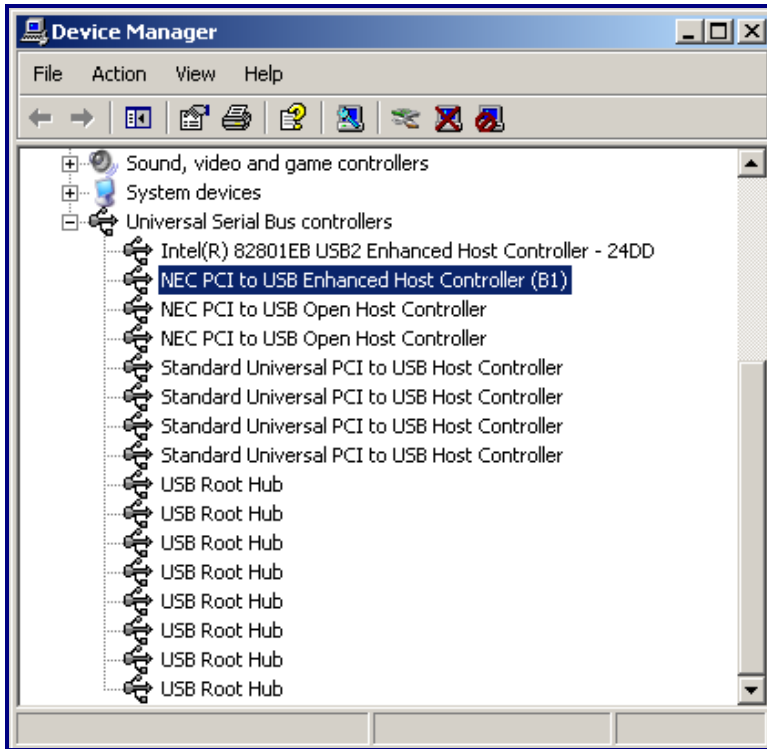
Device: If linked with the mini-B endpoint

(2) The program does not allocate the memory for the program. The absolute value is defined below to debug the code.

```
#define Host20_STRUCTURE_BASE_ADDRESS 0x3000000
```

If the program uses 0x3000000 ~ 0x3200000, users can allocate the memory to update the definitions at Lib_Host20.h.

- (3) The test bench only supports certain types of chipsets. USB ICE and the test board of the USB port cannot be linked with the same USB host on PC. The NEC USB driver cannot use the OS default driver; the enhanced host controller must be changed to the NEC PCI-to-USB enhanced host controller, B1. The driver is located in the “Win2K” directory.



2.14.2 Peripheral Mode Test Reference

If the OTG peripheral device is linked to PC, it will display the following message, "Line: mini B to link EV board PHY". If the OTG peripheral device is not linked to PC yet, users should run the ICE demo program, and the following message will be displayed.

```
*** OTG Ready *****
*** OTG Ready-1 *****
>>> Waiting for B_SESS_END & Phy Reset...(under 5sec)
*** OTG Ready-2 *****
***** Peripheral Mode *****
-- Normal--
 1.Peripheral Mode (Path Test)
 2.Peripheral Mode (Standard Peripheral CV Test)
 3.Peripheral Mode (Resume From Peripheral Test)
 4.Reflash
 5.Request 'Role Change'
 6.Enable/Disable 'Force Host to Full Speed'
*****
>>> (Device-B / Peripheral Mode) Input Command : 1

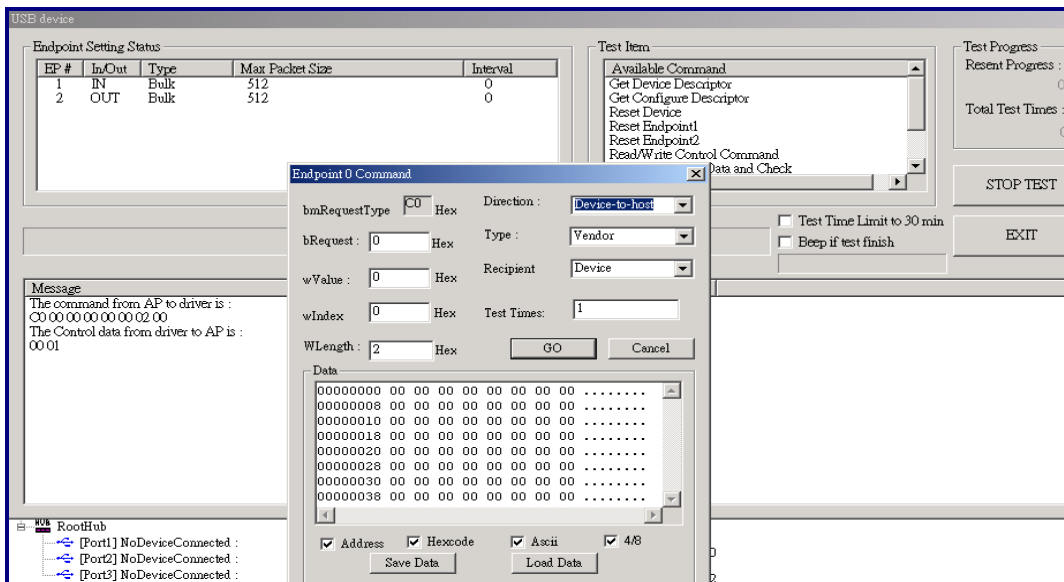
@@@ Device-B Enter Peripheral Mode...
```



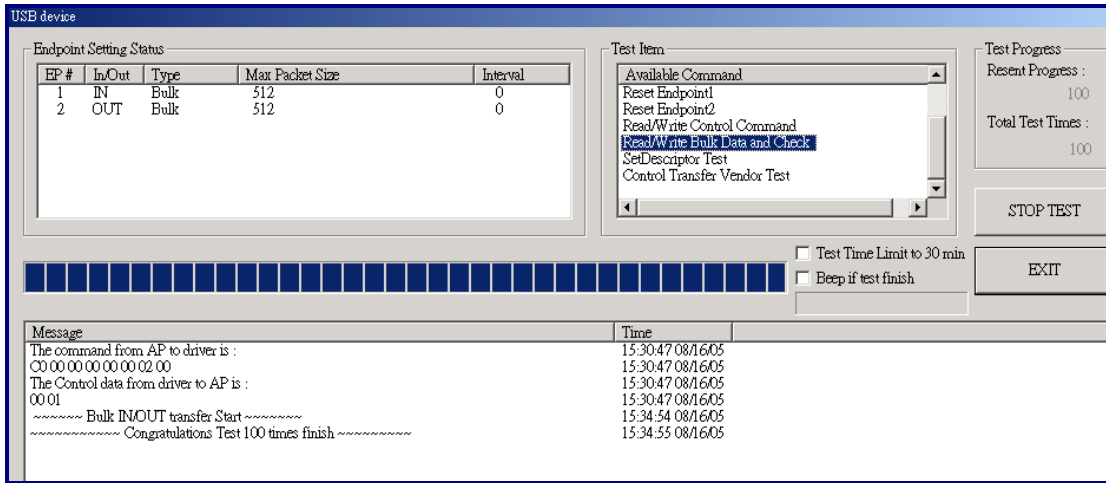
Users should link the OTG peripheral device to PC and run the PC USB test bench program; a device may be found as listed below.



Click this item twice and it will display the Endpoint Setting Status and the test item. Users can select the items, such as the read/write control commands, and click "GO". The following message window will appear.



If the read/write bulk data and check are selected and the test times and patterns are set, the following message will appear.



In the Pe_usb.h file, users must change “AP_Satus” to “Bulk_AP”, “Interrupt_AP”, “IsochronousIN_AP”, or “IsochronousOUT_AP”, when users verify the Bulk/Interrupt/IsochronousIN test item or the IsochronousOUT test item.

```

////// Define USB transfer test defines//////
#define Bulk_AP                0
#define Interrupt_AP           1
#define IsochronousIN_AP      2
#define IsochronousOUT_AP     3
#define AP_Satus                Interrupt_AP//IsochronousIN_AP

```

Bulk Path Test with PC/Faraday USB Test Bench

- Test firmware operation sequence:
 - Step 1: Attach the mini-B plug to PHY
 - Step 2: Run the OTG test firmware
 - Step 3: Select '1' to enter the peripheral mode (Use AP_Satus as Bulk)
 - Step 4: Attach to PC
 - Step 5: PC → Run the Faraday USB test bench
 - Step 6: PC/Faraday USB test bench → Select by double clicking the item, "Faraday USB Sys Faraday FUSB200 IO Test EV Board"
 - Step 7: PC/Faraday USB test bench → Select by double-clicking the item, "Read/Write Bulk Data and Check"
 - Step 8: PC/Faraday USB test bench → Input the read/write times and click "OK"
- Check the results:
 - PC/Faraday USB test bench will print out a message for verification.
- For example:
 - Step 3: Select '1' to enter the peripheral mode (Use Bulk configuration)

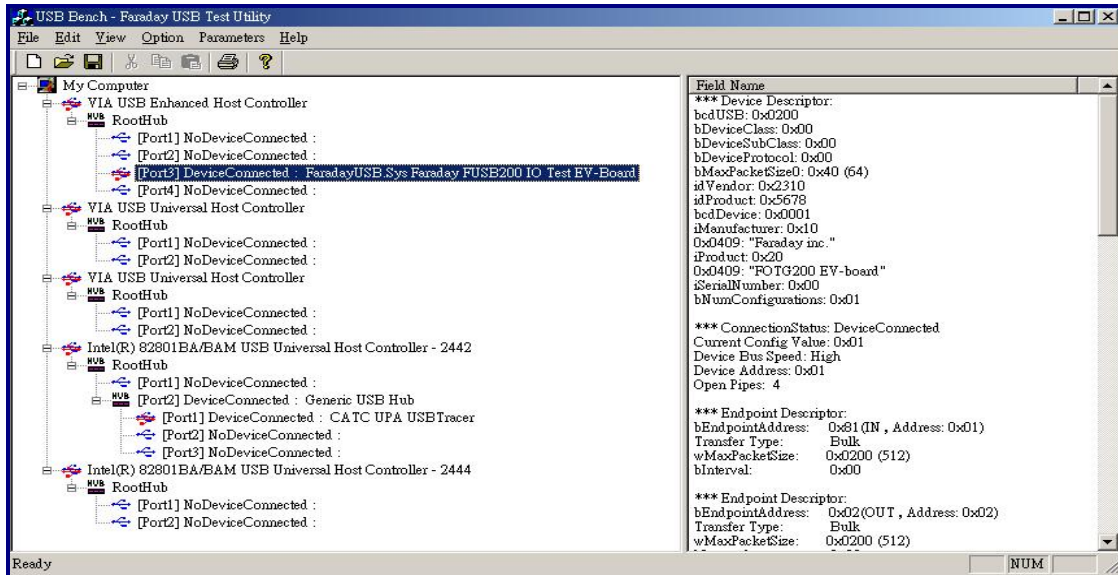
***** Peripheral Mode *****

-- Normal--

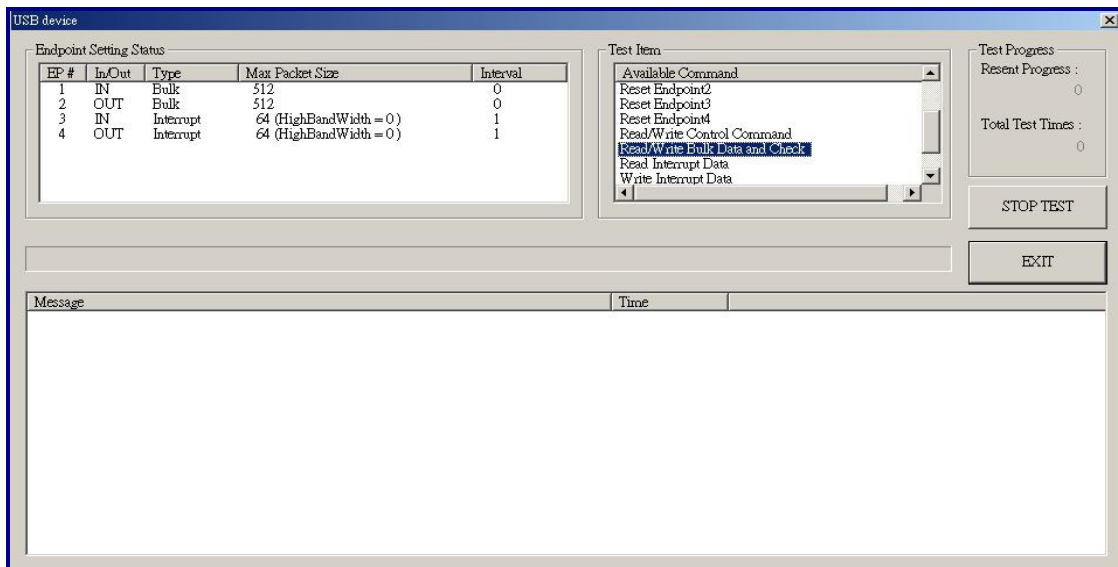
1. Peripheral mode (Path test)
2. Peripheral mode (Standard peripheral CV test)
3. Peripheral mode (Resume from peripheral test)
4. Reflash
5. Request "Role Change"
6. Enable/Disable "Force Host to Full Speed"

>>> (Device-B/Peripheral Mode) Input Command:

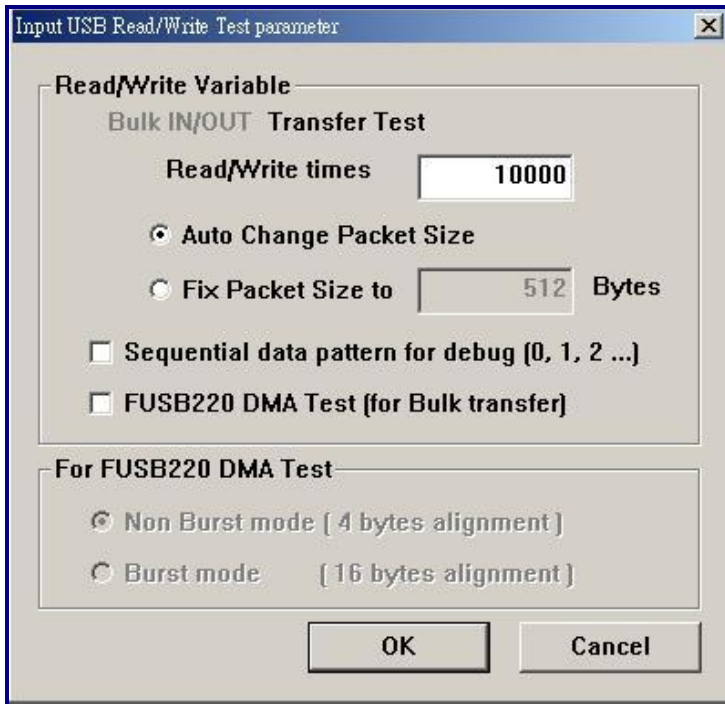
Step 6: PC/Faraday USB test bench → Select by double-clicking the item, “Faraday USB Sys Faraday FUSE200 IO Test EV Board”



Step 7: PC/Faraday USB test bench → Select by double-clicking the item, “Read/Write Bulk Data and Check”



Step 8: PC/Faraday USB test bench → Input read/write times

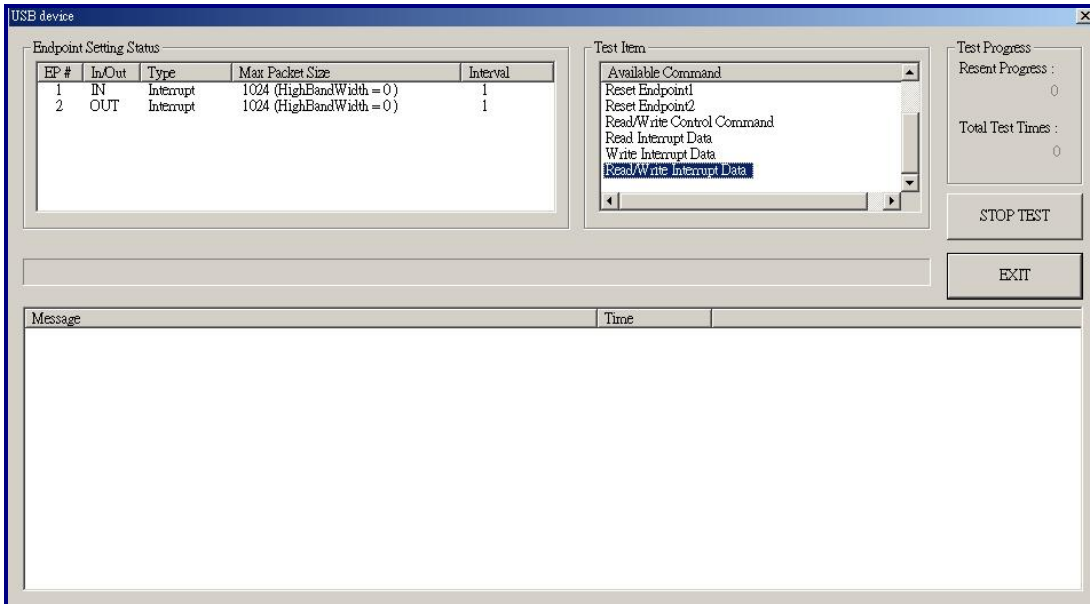


Interrupt Path Test with PC/Faraday USB Test Bench

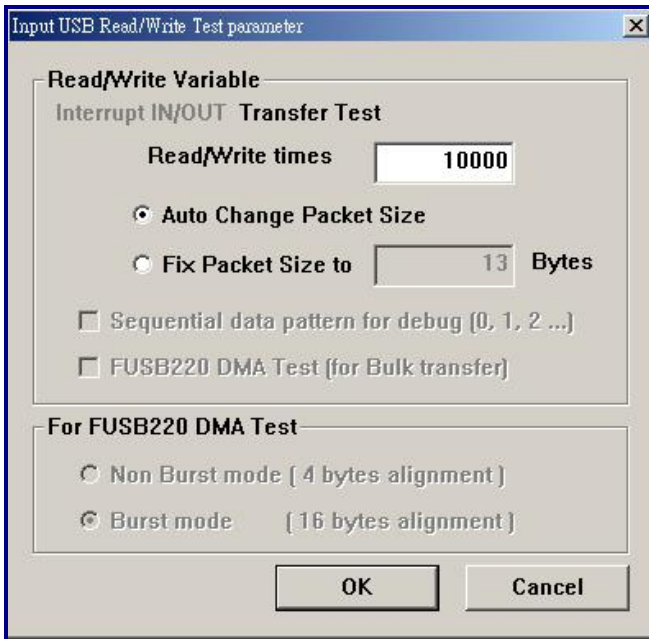
- Test firmware operation sequence:
 - Step 1: Attach the mini-B plug to PHY
 - Step 2: Run the OTG test firmware
 - Step 3: Select '1' to enter the peripheral mode (Use the Interrupt_AP configuration)
 - Step 4: Attach to the PC
 - Step 5: PC → Run the Faraday USB test bench
 - Step 6: PC/Faraday USB test bench → Select by double-clicking the item, "Faraday USB Sys Faraday FUSB200 IO Test EV Board"
 - Step 7: PC/Faraday USB test bench → Select by double-clicking the item, "Read/Write Interrupt Data and Check"
 - Step 8: PC/Faraday USB test bench → Input the read/write times
- Check the results:
 - PC/Faraday USB test bench will print out a message for verification.

- For example:

Step 7: PC/Faraday USB test bench → Select by double-clicking the item, “Read/Write Interrupt Data and Check”



Step 8: PC/Faraday USB test bench → Input the read/write times



ISO-In Path Test with PC-Faraday-USB-TestBench

Config Pe_usb.h file

```
#define OTG_AP_Satus      IsochronousIN_AP//Bulk_AP
```

Test firmware operation sequence:

Step 1: Attach the mini-B plug to PHY

Step 2: Run the OTG test firmware

Step 3: Select '1' to enter the peripheral mode (Use the ISO-In configuration)

Step 4: Attach to PC

Step 5: PC → Run Faraday USB test bench

Step 6: PC/Faraday USB test bench → Select by double-clicking the item, "Faraday USB Sys
Faraday FUSB200 IO Test EV Board"

Step 7: PC/Faraday USB test bench → Select by double-clicking the item, "Read Isochronous
Data"

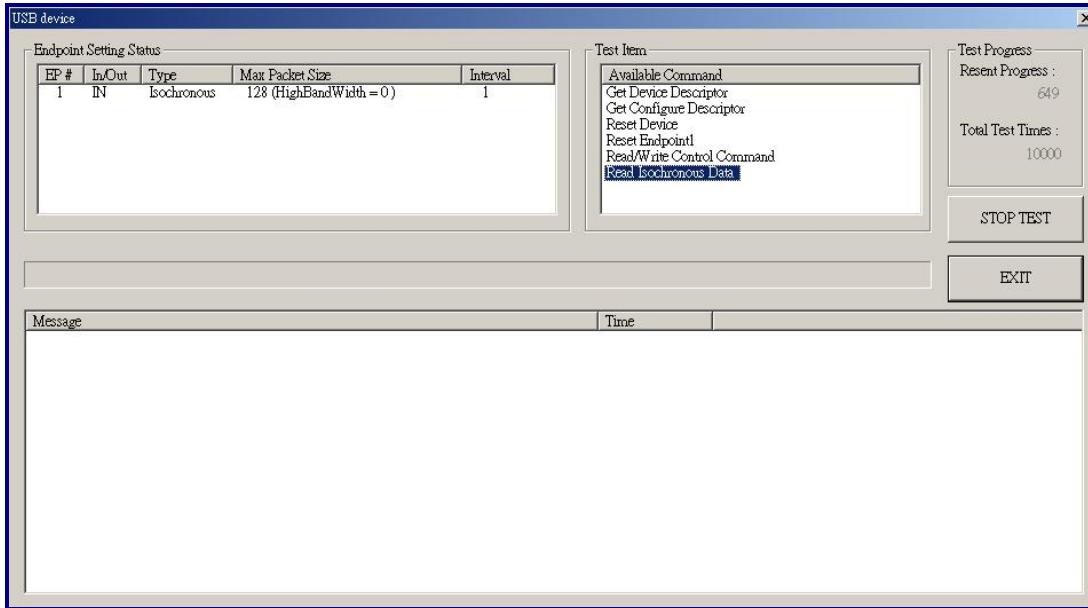
Step 8: PC/Faraday USB test bench → Input the read/write times

Check the results:

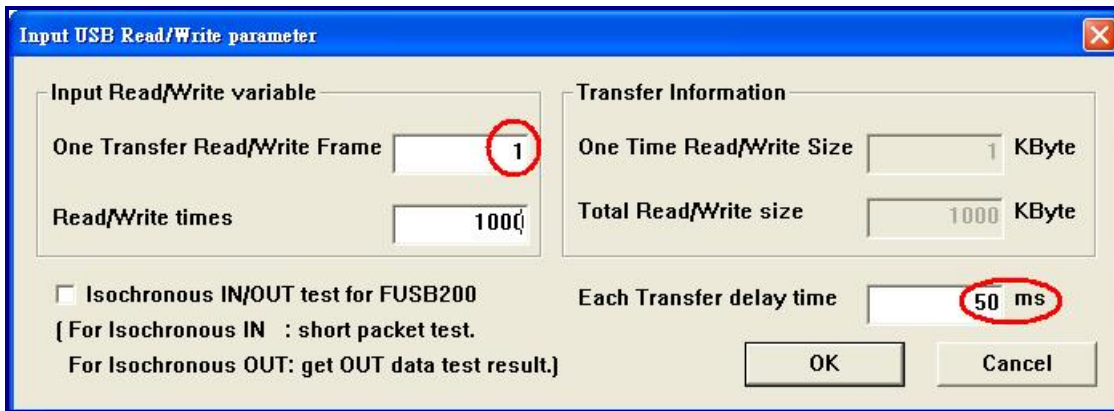
The PC/Faraday USB test bench will print out a message for verification.

For example:

Step 7: PC/Faraday USB test bench → Select by double-clicking the item, “Read Isochronous Data”



Step 8: PC/Faraday USB test bench → Input the read/write times



ISO-Out Path Test with PC/Faraday USB Test Bench

Config Pe_usb.h file

```
#define OTG_AP_Satus      IsochronousOUT_AP//Bulk_AP
```

Test firmware operation sequence:

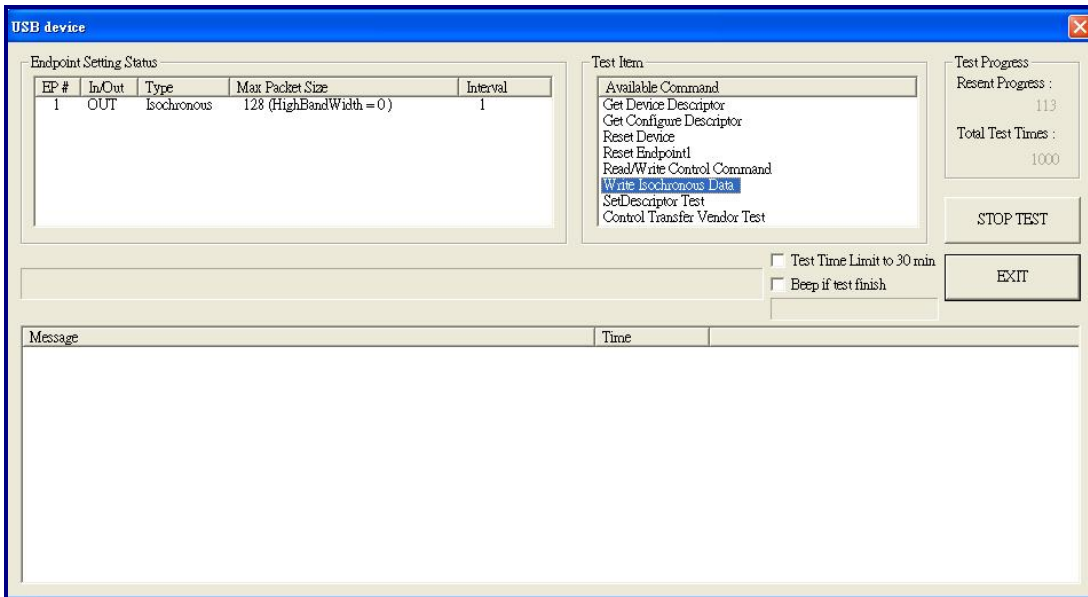
- Step 1: Attach the mini-B plug to PHY
- Step 2: Run the OTG test firmware
- Step 3: Select 1 to enter the peripheral mode (Use the ISO-Out configuration)
- Step 4: Attach to the PC
- Step 5: PC → Run the Faraday USB test bench
- Step 6: PC/Faraday USB test bench → Select by double-clicking the item, "FaradayUSB Sys Faraday FUSB200 IO Test EV Board"
- Step 7: PC/Faraday USB test bench → Select by double-clicking the item, "Write Isochronous Data"
- Step 8: PC/Faraday USB test bench → Input the read/write times

Check the results:

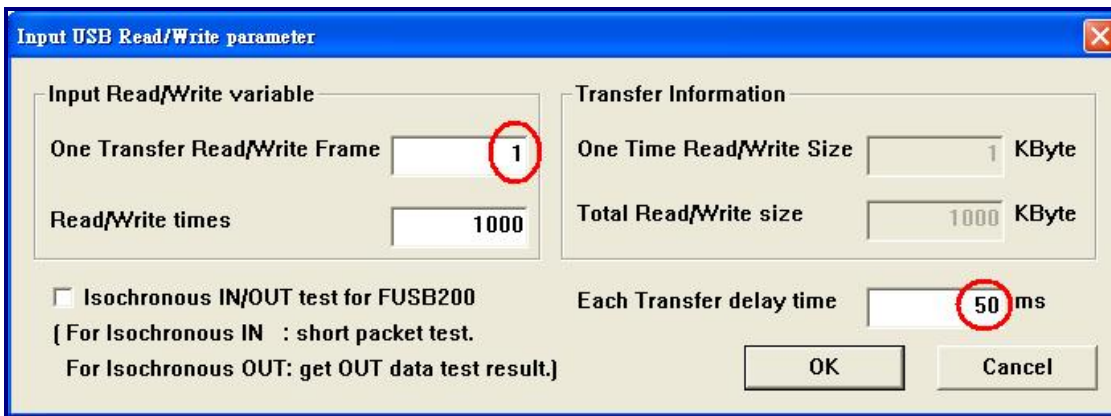
If there is any error, the peripheral will print out a message for verification.

For example:

Step 7: PC/Faraday USB test bench → Select by double-clicking the item “Write Isochronous Data”



Step 8: PC/Faraday USB test bench → Input the read/write times



2.14.3 Host Mode Test

To test in the host mode, users should link the EV board with PHY by using a mini-A line. The USB device should be linked with another point. The non-OS driver does not support the USB smart U3 card.

Bus Reset Test

- Test firmware operation sequence:
 - Step 1: Attach the mini-A plug to PHY
 - Step 2: Run the OTG test firmware
 - Step 3: Attach the peripheral to mini-B plug (Or the Standard B plug)
 - Step 4: Enter the host mode (Select item 2)
- Check the results:
 - The host will enumerate the peripheral and print out the peripheral information.
 - Please check:
 - Speed mode
 - Device/Configuration descriptor information
- For example:

```
Command>>35
*** OTG Ready *****
*** OTG Ready-1 *****
*** Full Speed Phy Reset Fail Work around solution => Enable 0x80 Bit28
*** OTG Ready-2 *****
***** Device-A: Host Mode *****
-- Normal--
  1.Host Mode (OTG Path Test)
  3.Reflash
-- SRP/HNP --
  5.Waiting for Role Change(SRP+HNP)(VBUS)
  6.Waiting for Role Change(SRP+HNP)(VBUS)Auto-Test x times
  7.Waiting for Role Change(SRP+HNP)(Data Line)
  8.Disable the SRP Detect
  9.Drive VBUS
 10.Drop the VBUS
 11.Reset Host Controller
```

```

12.OPT-Device-A Test-Auto
13.Dump Memory
15.FPGA Half Speed (HCLK < 30 ==> Enable)((HCLK >= 30 ==> Disable)
-- Status
**VBUS:0x80000 **SRP_DET:0x0 ** SRP_DET_TYPE:0x0
**Host-Connect:0x0 ** ID/Role:0xe0010
*****
>>> (Device-A/Host Mode) Input Command : 1

@@@ Drive VBUS ok...
@@@ Enter DRD Host Mode...
@@@ Waiting for Device Connect...
>>>Device already connect...
Get PID=2310, VID=5678
>>> Device Speed: High Speed...
*****
***** Device Enumerate Information *****
*****
>>> VID=0x2310 / PID=0x5678 / USB Ver=0x20 / Ed0 MaxSize=64
>>> Manufacturer:GM inc.

>>> Product:FOTG200 EV-board?
>>> Serial:

>>>(Hex) 12 01 00 02 00 00 00 40 10 23 78 56 01 00 10 20 00 01
***** Configuration[0] = 1 *****
>>>(Hex) 09 02 31 00 01 01 30 e0 00
***** Interface[0] = 0 *****
>>>(Hex) 09 04 00 00 04 00 00 00 40
*** EndPoint[0] = 1 ***
>>> Bulk-In (MaxSize=512 ,Interval=0 ,HighBandWidth=1 )
>>>(Hex) 07 05 81 02 00 02 00
*** EndPoint[1] = 2 ***
>>> Bulk-Out (MaxSize=512 ,Interval=0 ,HighBandWidth=1 )
>>>(Hex) 07 05 02 02 00 02 00

```

```
*** EndPoint[2] = 3 ***
>>> Interrupt-In (MaxSize=64 ,Interval=1 ,HighBandWidth=1 )
>>>(Hex) 07 05 83 03 40 00 01
*** EndPoint[3] = 4 ***
>>> Interrupt-Out (MaxSize=64 ,Interval=1 ,HighBandWidth=1 )
>>>(Hex) 07 05 04 03 40 00 01
***** OTG Descriptor *****
>>>(Hex) 03 09 03
***** Host Mode AP *****
-- Device Speed: High Speed...
-- Normal--
1.Control => Get Device Descriptor
5.GM AP Test
7.Suspend/Resume Test 100 times
8.Suspend/Remote Wake Up Test
10.Hot Plug test 10 times
11.Cold Plug test 10 times
12.Static Emulation Test (127 times)
19.Quit
*****
*** Please Input the Item:5
```

Control Command Test

- Test firmware operation sequence:
 - Step 1: Attach the mini-A plug to PHY (OTG-A)
 - Step 2: Run the OTG Test Firmware in OTG-A
 - Step 3: OTG-A forces speed (Select item 9).
 - Step 4: Attach the Peripheral-GM-OTG to mini-B plug
 - Step 5: Run the OTG Test Firmware in OTG-B and select '1' to enter the peripheral mode
 - Step 6: OTG-A enters the host mode (Select item 2).
 - Step 7: OTG-A enters "GM AP Test" (Select item 5).
 - Step 8: OTG-A enters "Control → Control Command Test" (Select item 1).



- Check the results:
 - The host will issue the standard control commands and verify the results.
 - Check if “PASS” is printed out

For example:

```

***** Host Mode GM Path Test AP *****
1.Control => Control Command Test
3.Bulk In/Out Test
4.Int => OTGH_PT_Interrupt_In_Out
9.Quit
*****
*** Please Input the Item:1

***** Test Control Command Counter = 1 *****

>>> Get_Configuration(1) Test      ==> PASS
>>> Get_Interface(0) Test         ==> PASS
>>> Set_Interface to 0 Finish      ==> PASS
>>> Get_Interface(0) Test         ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
    (Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 2 *****

>>> Get_Configuration(1) Test      ==> PASS
>>> Get_Interface(0) Test         ==> PASS
>>> Set_Interface to 0 Finish      ==> PASS

```



```

>>> Get_Interface(0) Test          ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
    (Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 3 *****

>>> Get_Configuration(1) Test      ==> PASS
>>> Get_Interface(0) Test          ==> PASS
>>> Set_Interface to 0 Finish      ==> PASS
>>> Get_Interface(0) Test          ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
    (Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 4 *****

>>> Get_Configuration(1) Test      ==> PASS
>>> Get_Interface(0) Test          ==> PASS
>>> Set_Interface to 0 Finish      ==> PASS
>>> Get_Interface(0) Test          ==> PASS

```



```

>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
    (Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 5 *****

>>> Get_Configuration(1) Test ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Set_Interface to 0 Finish ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
    (Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
    (Bit0=1 => Power From Device)
    (Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 6 *****

>>> Get_Configuration(1) Test ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Set_Interface to 0 Finish ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS

```

```

(Bit0=1 => Power From Device)
(Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
(Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
(Bit0=1 => Power From Device)
(Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 7 *****

>>> Get_Configuration(1) Test ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Set_Interface to 0 Finish ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
(Bit0=1 => Power From Device)
(Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
(Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
(Bit0=1 => Power From Device)
(Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 8 *****

>>> Get_Configuration(1) Test ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Set_Interface to 0 Finish ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
(Bit0=1 => Power From Device)

```

```

(Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
(Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
(Bit0=1 => Power From Device)
(Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 9 *****

>>> Get_Configuration(1) Test ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Set_Interface to 0 Finish ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
(Bit0=1 => Power From Device)
(Bit1=0 => RemoteWakeUp Disable)
>>> Get_Status(0) (Endpoint status) Test ==> PASS
(Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
(Bit0=1 => Power From Device)
(Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS
***** Test Control Command Counter = 10 *****
*

>>> Get_Configuration(1) Test ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Set_Interface to 0 Finish ==> PASS
>>> Get_Interface(0) Test ==> PASS
>>> Get_Status(1)(Power/RemoteWakeUp) Test ==> PASS
(Bit0=1 => Power From Device)
(Bit1=0 => RemoteWakeUp Disable)

```

```

>>> Get_Status(0) (Endpoint status) Test ==> PASS
      (Bit0=0 => Not Halt)
>>> Set_Feature to 'RemoteWakeUp Enable' ==> PASS
>>> Get_Status (3)(Power/RemoteWakeUp) Test ==> PASS
      (Bit0=1 => Power From Device)
      (Bit1=0 => RemoteWakeUp Disable)
>>> Clear_Feature to 'RemoteWakeUp Disable' ==> PASS
>>> Get_Status(1) (Power/RemoteWakeUp) Test ==> PASS

```

Bulk Path Test

- Test firmware operation sequence:
 - Step 1: Attach the mini-A plug to PHY (OTG-A)
 - Step 2: Run the OTG test firmware in OTG-A
 - Step 3: OTG-A forces speed (Select item 9).
 - Step 4: Attach the peripheral GM OTG to the mini-B plug
 - Step 5: Run the OTG test firmware in OTG-B and select '1' to enter the peripheral mode (Use the Bulk configuration)
 - Step 6: OTG-A enters the host mode (Select item 2).
 - Step 7: OTG-A enters "GM AP Test" (Select item 5).
 - Step 8: OTG-A enters "Bulk In/Out Test" (Select item 3).
- Check the results:
 - The host will print out a "PASS" message.

For example:

```

***** Host Mode GM Path Test AP *****
1.Control => Control Command Test
3.Bulk In/Out Test
4.Int => OTGH_PT_Interrupt_In_Out
9.Quit
*****
*** Please Input the Item:3

```

```

>>> Please Input the Bulk-In/Out counter:10

>>> Please Input the 'Print Message' counter:1

>>> Bulk In/Out Max Packet Size = 512
>>> High Speed Bulk-IN/Out Test:PASS=(1) (Address=0x3102000) (Size=19880) (Total Size=0MB + 39848Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(2) (Address=0x3102001) (Size=19779) (Total Size=0MB + 79494Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(3) (Address=0x3102002) (Size=19678) (Total Size=0MB + 118938Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(4) (Address=0x3102003) (Size=19577) (Total Size=0MB + 158185Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(5) (Address=0x3102004) (Size=19476) (Total Size=0MB + 197220Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(6) (Address=0x3102005) (Size=19375) (Total Size=0MB + 236058Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(7) (Address=0x3102006) (Size=19274) (Total Size=0MB + 274694Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(8) (Address=0x3102007) (Size=19173) (Total Size=0MB + 313128Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(9) (Address=0x3102008) (Size=19072) (Total Size=0MB + 351360Bytes)...
>>> High Speed Bulk-IN/Out Test:PASS=(10) (Address=0x3102009) (Size=18971) (Total Size=0MB + 389390Bytes)...
>>> Finish Bulk In/Out Test PASS (10)...
>>> Again(1/2->y/n):2

```

Interrupt Path Test

Low-speed test

- Test firmware operation sequence:
 - Step 1: Attach the mini-A plug to PHY (OTG-A)
 - Step 2: Run the OTG test firmware in OTG-A
 - Step 3: OTG-A forces speed (Select item 9).
 - Step 4: Attach the keyboard/mouse to standard B plug
 - Step 5: OTG-A enters the host mode (Select item 2).
 - Step 6: OTG-A enters "5.Keyboard Mouse HID Test" (Select item 5).
 - Step 7: Move the keyboard input/Mouse

- Check the results:
 - The host will print out a message for verification.
- For example:
 - Mouse: 4-byte data
 - Keyboard: 8-byte data

High-speed test

- Test firmware operation sequence:
 - Step1: Attach the mini-A plug to PHY (OTG-A)
 - Step2: Run the OTG test firmware in OTG-A
 - Step3: OTG-A forces speed (Select item 9).
 - In the full-speed mode, the host will select the full speed; while in the high-speed mode, the host will select "clear all".
 - Step4: Attach the peripheral GM OTG to the mini-B plug
 - Step5: Run the OTG test firmware in OTG-B and select '1' to enter the peripheral mode (Peripheral uses the interrupt configuration.)
 - Step6: OTG-A enters the host mode (Select item 2).
 - Step7: OTG-A enters "GM AP Test" (Select item 5).
 - Step8: OTG-A enters "Interrupt In/Out Test" (Select item 3).
- Check the results:
 - The host will print out a "PASS" message.

```

***** Host Mode GM Path Test AP *****
1.Control => Control Command Test
3.Bulk In/Out Test
4.Int => OTGH_PT_Interrupt_In_Out
9.Quit
*****
*** Please Input the Item:4

>>> Please Input the Interrupt Counter:10

```

```
>>> wMaxSize = 64
>>> wInputCounter = 10
>>> wPrintCounter = 10
>>> wIncrementSize = 1
>>> High Speed...
>>> INT In Test (wForceInterval=1 / bS_Mask=1 / bMult=1):PASS Counter(10)
>>> Current Size = 10 Total In Length=0 M Bytes (55 Bytes)
>>> INT Out Test (wForceInterval=1/bS_Mask=1/bMult=1):PASS Counter(10)
>>> Current Size = 10 Total Out Length=0 M Bytes (55 Bytes)
>>> Interupt Test Finish ... PASS (10)
```

ISO Path Test

Config Pe_usb.h file

```
#define OTG_AP_Satus          IsochronousIN_AP//Bulk_AP
```

- Test firmware operation sequence:
 - Step 1: Attach the mini-A plug to PHY (OTG-A)
 - Step 2: Run the OTG test firmware in OTG-A
 - Step 3: OTG-A forces speed (Select item 4).
 - Step 4: Attach the peripheral GM OTG to mini-B plug
 - Step 5: Run the OTG test firmware in OTG-B and select '1' to enter the peripheral mode (Peripheral uses the ISO-In/ISO-Out configuration.)
 - Step 6: OTG-A enters the host mode (Select item 2).
 - Step 7: OTG-A enters "GM AP Test" (Select item 5).
 - Step 8: OTG-A enters "ISO-In/ISO-Out Test" (Select item 3).
- Check the results:
 - The host will print out a "PASS" message.

When the ISO-In configuration is selected, it will display as follows.

```
***** Host Mode GM Path Test AP *****

1.Control => Control Command Test
3.ISO-In Test
9.Quit

*****

*** Please Input the Item:
3

>>> Free Run 1000 times:

>>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 0) Total Size=0MB + 1024Bytes PASS...=====> Counter=1
>>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 101) Total Size=0MB + 2048Bytes PASS...=====> Counter=2
>>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 202) Total Size=0MB + 3072Bytes PASS...=====> Counter=3
>>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 303) Total Size=0MB + 4096Bytes PASS...=====> Counter=4
>>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 404) Total Size=0MB + 5120Bytes PASS...=====> Counter=5
>>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 505) Total Size=0MB + 6144Bytes PASS...=====> Counter=6
>>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 606) Total Size=0MB + 7168Bytes PASS...=====> Counter=7
>>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 707) Total Size=0MB + 8192Bytes PASS...=====> Counter=8
... >>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 1313) Total Size=0MB + 1021952Bytes PASS...=====>
Counter=998 >>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 1414) Total Size=0MB + 1022976Bytes
PASS...=====> Counter=999 >>> High Speed ISO IN Test:(Current Size=1024 / Current Offset = 1515) Total Size=0MB +
1024000Bytes PASS...=====> Counter=1000 >>> ISO IN Test:PASS (1000)
```



Config Pe_usb.h file

```
#define OTG_AP_Satus          IsochronousOUT_AP//Bulk_AP
```

When the ISO-Out configuration is selected, it will display as follows.

```
***** Host Mode GM Path Test AP *****

1.Control => Control Command Test
3.ISO-Out Test
9.Quit

*****

*** Please Input the Item:3

>>> Free Run 1000 times:

>>> High Speed ISO-Out data : (Size =1024 / wOffset =0)Total Size=0MB + 1024Bytes =====> Counter=1
>>> High Speed ISO-Out data : (Size =1024 / wOffset =101)Total Size=0MB + 2048Bytes =====> Counter=2
>>> High Speed ISO-Out data : (Size =1024 / wOffset =202)Total Size=0MB + 3072Bytes =====> Counter=3
>>> High Speed ISO-Out data : (Size =1024 / wOffset =303)Total Size=0MB + 4096Bytes =====> Counter=4
>>> High Speed ISO-Out data : (Size =1024 / wOffset =404)Total Size=0MB + 5120Bytes =====> Counter=5
>>> High Speed ISO-Out data : (Size =1024 / wOffset =505)Total Size=0MB + 6144Bytes =====> Counter=6
>>> High Speed ISO-Out data : (Size =1024 / wOffset =606)Total Size=0MB + 7168Bytes =====> Counter=7
>>> High Speed ISO-Out data : (Size =1024 / wOffset =707)Total Size=0MB + 8192Bytes =====> Counter=8
...
>>> High Speed ISO-Out data : (Size =1024 / wOffset =1313)Total Size=0MB + 1021952Bytes =====>
Counter=998
>>> High Speed ISO-Out data : (Size =1024 / wOffset =1414)Total Size=0MB + 1022976Bytes =====>
Counter=999
>>> High Speed ISO-Out data : (Size =1024 / wOffset =1515)Total Size=0MB + 1024000Bytes =====>
Counter=1000
>>> ISO-Out data finish : 1000
```



2.14.4 Role Selection

For the 8126 series platforms, the IDDIG signal is set by using the PMU register. The jumper setting on board will no longer be used to set the IDDIG signal. Users should decide the role that the test will be performed.

```
Command>>35
>>> Please select role to work :
1. host mode
2. device mode
```

2.15 Burn U-BOOT Image Function

This section describes how to write the U-BOOT code into Flash. The system will copy the file name, U-BOOT.bin, from the SD card and write to Flash at the address, 0x10200000, for an 16-bit wide bus.

Test Result Reference

When the function item 73 is selected, it will display as follows.

```
Command>>73
File Name must be <U-BOOT.bin>
MMU/Cache enabled
Flash Manu.ID = 0x01 Device ID = 0x7e (Spansion)
CMD8: Card does not support SPEC 2.0 or voltage is not supported!
Card Size = 3958MB,max data block length = 0x200

...SD Card on Drive D: !!!
Command initial
Loading Image contents
Copy to address 0x2000000
Wait.....
Total copy 81850 bytes.
Erasing Flash Content
Erase from address = 0x10200000
```



```
Erasing Flash Content
Erase from address = 0x10240000
Erasing block 99/99
Flash Programming & Verifying from addr = 0x10240000
3218420 of 3218420 words programmed
Taking 0:0 to program/verify

Command>>
```

2.17 Run Linux from SD Card

This section describes how to run the Linux code from a SD card. The system will run the Linux image from the SD card with the file name, "linux", and boot it without writing into the Flash.

Test Result Reference

When the function item 76 is selected, it will display as follows.

```
Command>>76
File Name must be <linux>
CMD8: Card does not support SPEC 2.0 or voltage is not supported!
Card Size = 489MB,max data block length = 0x200
Drive 'C' Have No Find Partation Table

...SD Card on Drive C: !!!
Command initial
Copy to address 0x2000000
Wait.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
```



```
.....  
Total copy 14210988 bytes.  
## Starting AP at 0x02000000 ...  
Uncompressing Linux.....  
.....
```

2.18 Address Read/Write Test

This function test reads/writes the memory or the register data. If users have no ICE and want to read/write the memory or the register data, this function can be used for this purpose.

Test Result Reference

When the function item 72 is selected, it will display as follows.

```
Command>>72  
Address>>0x100  
  
Current Data = 0x00000071  
Data>>0x12345678
```

When the function item 71 is selected, it will display as follows.

```
Command>>71  
Address>>0x100  
  
Length, default 0x40>>0x10  
  
Addr 0x100 == 0x12345678 0xe3a00499 0xe59f1c04 0xe5801038  
Addr 0x110 == 0xe59f1c00 0xe580103c 0xe59f1bfc 0xe59f2bfc  
Addr 0x120 == 0xe5812000 0xe59f2bf8 0xe5812004 0xe3a00499  
Addr 0x130 == 0xe59f1bf0 0xe3811001 0xe5801034 0xe59f2be8
```

2.19 Security Test

This test is used to verify the encrypt and decrypt functions of AES/DES. Users can select a specific test items for testing and review the results.

Test Result Reference

When the function item 45 is selected, it will display as follows.

```
Command>>45
-----
GM Technology Corporation CPE Burn-in Program
-----
( 1) Simple AES Security Test
( 2) Simple DES & Triple-DES Security Test
( 5) Security DMA function Test

( 7) Security Self Encrypt/Decrypt Test
( 8) Security Not Change Key Special Test
(99) Return Main Menu

Command>>8
=== Algorithm_AES_128/256 ECB mode DMA encrypt/decrypt
Compare Cipher Pass
Compare Cipher Pass
=== Algorithm_AES_128/256 CBC mode DMA encrypt/decrypt
Compare Cipher Pass
Compare Cipher Pass

Command>>99
```

2.20 UART Test

- This test is used to verify the functions of UART 1/2/3/4. Users can select a specified UART for testing and review the results.

Test Result Reference

When the function item 60 is selected, it will display as follows.

```
Command>>60
Which UART do the console change?(1 or 2)?2
Console will be routed to UART2!!
Please press Enter to show the menu!!
```

2.21 Burn Boot Code Function

The system writes the burn-in code from the SD card with the file name, "rom.bin".

Test Result Reference

When the function item 74 is selected, it will display as follows.

```
Command>>74
File Name must be <rom.bin>
MMU/Cache enabled
Flash Manu.ID = 0x01 Device ID = 0x7e (Spansion)
CMD8: Card does not support SPEC 2.0 or voltage is not supported!
Card Size = 3958MB,max data block length = 0x200

...SD Card on Drive C: !!!
Command initial
Loading Image contents
Copy to address 2000000
Wait.....
.....
```

```
Total copy 599924 bytes.  
Erasing Flash Content  
Erase from address = 0x10000000  
Erasing block 5/5  
Flash Programming & Verifying from addr = 0x10000000  
149982 of 149982 words programmed  
Taking 0:0 to program/verify  
  
Command>>
```

Software Boot Sequence

Users can update the code step by the step as shown below. The hardware settings must be ready before running the function.

Step 1: Run OPENice and select the "Run Batch File" option, load "GM8126_openice.scp" (This file can be found in SDK StartCell_Drivers\image\)

Step 2: Open the RAM version image, burnin*.axf

Step 3: Check the top menu, "Configure Processor," and disable the "vector Catch" item

Step 4: Select "File/Load Memory From File" then select the "u-boot.bin" file (This file can be found in SDK Embedded_Linux\image\), address: 0x1800000

Step 5: Right-click the mouse to set the PC to 0x1800000 for the "Disassembly" region

Step 6: Select "Go"

Step 7: In HyperTerminal, enter the following command:

```
CPE > erase 0x10000000 0x1023ffff
```

Step 8: In HyperTerminal, enter the following command:

```
CPE > cp.b 0x1800000 0x10200000 0x14000
```

Step 9: Wait for Step 8 to complete. In OPENice, select "Stop"

Step 10: Select the "File/Load Memory From File" directory and select the "rom.bin" file (This file can be found in SDK StartCell_Drivers\image\), address: 0x2000000

Step 11: Select "Go"

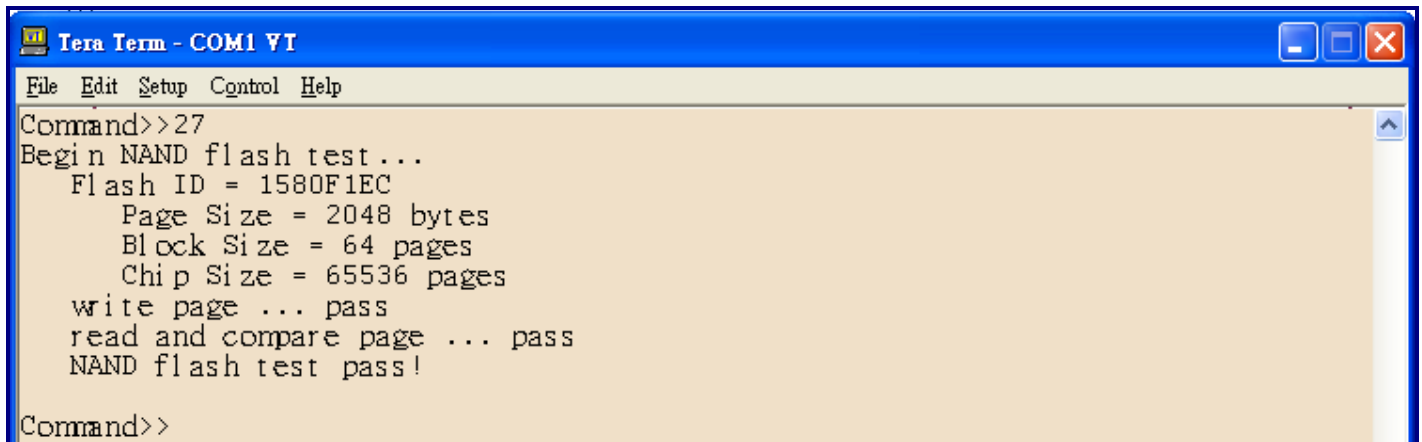
Step 12: In HyperTerminal, enter the following command:

```
CPE > cp.b 0x2000000 0x10000000 0xA0000
```

Step 13: Wait for Step 12 to complete; then shut down OPENice, remove ICE, and reset EVB

2.22 NAND Flash Test

GM8126 supports the SLC-type NAND Flash. This test item erases the Flash, writes the data pattern, and verifies the content on that Flash. The test run is shown as below.



```
Tera Term - COM1 VT
File Edit Setup Control Help
Command>>27
Begin NAND flash test...
Flash ID = 1580F1EC
Page Size = 2048 bytes
Block Size = 64 pages
Chip Size = 65536 pages
write page ... pass
read and compare page ... pass
NAND flash test pass!
Command>>
```

The test items of a NAND Flash are as following:

- Check whether the NAND Flash ID is known
- Erase each block and program the first page and last page of each good block
- Read back the programmed pages and checking whether the data are correct

Note: The test will erase all the programs installed in Flash.

2.23 VGA Setting

The LCD controller needs accurate working frequency to output a correct image. Several VGA output resolution settings are provided in default to compute the appropriate PLL3 clock. According to the application, users should select the desired resolution before Linux has been booting up. The default VGA output resolution is 1024x768.

```
Command>>52
Select VGA resolution:
0. 1024x768
1. 1280x800
2. 1280x960
3. 1280x1024
4. 1360x768
5. 720P
6. 1080I
7. 800x600
```

There is a definition to declare the default setting in the burnin\code\00_main\00_main.c file.

```
#define VGA_CONFIG 0 // Default VGA setting = 1024x768
```

2.24 LCD Test and Burn Logo Image

Select item 54 to burn the logo image on Flash and select item 13 to display the logo image on screen.

Before entering the function item 54, users should prepare the following tasks:

1. Create an image file with raw data and define its name as "426x147.yuv"
2. Store this file on an SD card

```
Command>>54
File Name must be <426x147.yuv>
MMU/Cache enabled
Flash Manu.ID = 0x01 Device ID = 0x7e (Spansion)
Card Size = 972MB,max data block length = 0x200

...SD Card on Drive C: !!!
Command initial
Loading Image contents
Copy to address 0x18a0000
Wait.....
Total copy 125244 bytes.
Erasing Flash Content
Erase from address = 0x100a0000
Erasing block 1/1
Flash Programming & Verifying from addr = 0x100a0000
31312 of 31312 words programmed finished
```

```
Command>>13
0. TV Composite NTSC
1. TV Composite PAL
2. VGA
3. HDMI(BT1120)
9. ESC

What kind of Display ?
2
```

The following definitions are defined in the burnin\code\00_main\00_main.c file and describe the base address that stores an image and the size of that image.

```
#define CPE_LOGO_BASE 0x100A0000
#define LOGO_SIZE 0x20000
```

Two LCD controllers are designed for the 8126 series platforms. This controller can display the logo image on two monitors simultaneously.

```
Command>>13
0. TV Composite NTSC
1. TV Composite PAL
2. VGA
3. HDMI(BT1120)
4. VGA(LCD)+TV_NTSC(LCD2)
9. ESC
```

What kind of Display ?

4

2.25 Keyscan Test Only for 8126

This test item can only be performed on the 8126 series platforms. Before selecting item 41, it is necessary to connect the keyboard to the EV board.

```
Command>>41
Begin Keyscan Test...
  Press anykey from keyboard! or Enter 'q' from console to exit test.
Keyscan... key[0x2000,0xf] is detected.
Keyscan... key[0x2000,0xf] is detected.
Keyscan... key[0x2000,0xf] is detected.
Keyscan... key[0x2000,0xb] is detected.
Keyscan... key[0x10000,0xc] is detected.
Keyscan... key[0x10000,0xf] is detected.
Keyscan... key[0x4000,0xc] is detected.
Keyscan... key[0x4000,0xf] is detected.
Keyscan... key[0x12000,0xb] is detected.
Keyscan... key[0x12000,0xc] is detected.
```

2.26 IRDET Test Only for 8126

This test item can only be performed on the 8126 series platforms. Please select item 42 to test the IRDET function. Depending on the stand EVB layout, users can choose the GPIO number used and the transmit protocol.

```
Command>>42
Begin Irdet Test...
Select input port(0:Return)=> 1:On-board(GPIO-21) 2:External board(GPIO-10)
1
Select protocol(0:Return)=> 1:NEC 2:RCA 3:Sharp 4:Philips RC-5 5:Nokia NRC17
1
Press button from remote controller! Enter 'q' from console to exit test.
Irdet... command[0x0] is detected. (Repeat)
Irdet... command[0x57f80000] is detected. (Repeat)
Irdet... command[0x8070057f] is detected. (Repeat)
Irdet... command[0x403802bf] is detected. (Repeat)
Irdet... command[0x201c015f] is detected. (Repeat)
Irdet... command[0xe916ff00] is detected.
Irdet... command[0xea15ff00] is detected.
Irdet... command[0xf80750af] is detected. (Repeat)
Irdet... command[0x2bfc03e0] is detected. (Repeat)
Irdet... command[0x2bfc03e0] is detected. (Repeat)
```

Chapter 3

Compile GM8126 Burn-in Program

This chapter contains the following sections:

- 3.1 ROM Version
- 3.2 RAM Version
- 3.3 ICE Options/Configure Processor

The GM8126 burn-in program contains two package versions: The ROM version and the RAM version. The ROM version is executed from Flash (Boot code), while the RAM version is for debugging with the ICE tool. The following sections introduce how to execute the burn-in programs in both the ROM and RAM versions. Please refer to the documents related to the ARM Developer Suite, Ver. 1.2 for more detailed information.

Users have to copy the burn-in source code to the working directory, and install the ARM Developer Suite Ver. 1.2 into PC to build the ROM and RAM versions.

The project files in ADS 1.2 is described in the table below.

Project File	ADS 1.2
burnin package	burnin_ads_1.2.mcp
flib package	flib_ads_1.2.mcp

3.1 ROM Version

Please follow the steps below for the ROM version of the GM8126 burn-in program:

- Step 1: Execute "Metrowerks CodeWarrior" and open the mcp file resided in "<your directory>\burnin\burnin\burnin_ads_1.2.mcp" in the ADS 1.2 environment.
- Step 2: Open "fLib_ads_1.2.mcp" by double-clicking the mcp file (Residing at <your directory >\fLib\Build\)
- Step 3: Set to the "Rom" mode on the burnin_ads_1.2.mcp and fLib_ads_1.2.mcp windows (As shown in Figure 3-1)
- Step 4: Press the "F5" key on the keyboard to compile flib_ads_1.2.mcp (Click the fLib.mcp window to ensure that the fLib project has been selected)
- Step 5: Click the burnin_ads_1.2.mcp window to select the burn-in project. Press the "F5" key to compile burnin_ads_1.2.mcp

The binary file will be created and saved in the "<your directory>\burnin\burnin\burnin_data \rom\rom.bin" directory. Users can use the binary file to burn the Flash for GM8126 when the source code has been modified.

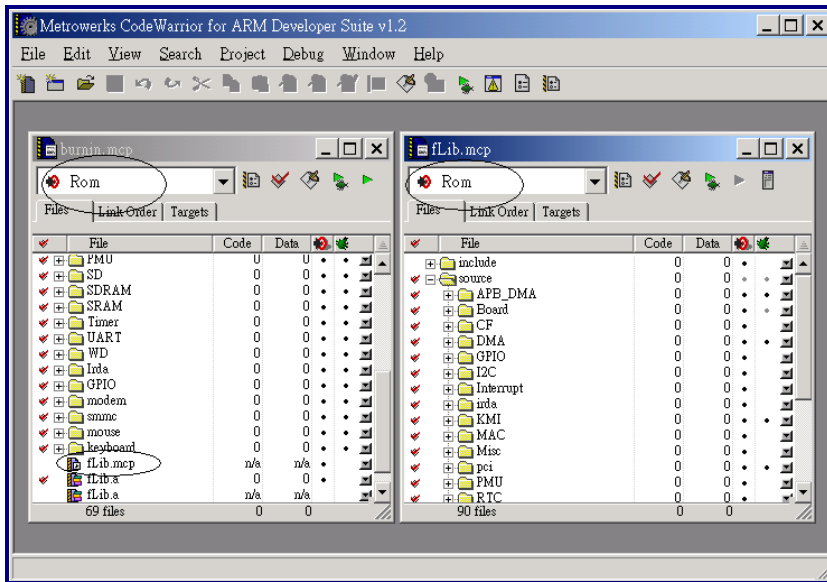


Figure 3-1. ROM Version

3.2 RAM Version

Please follow the steps below for the RAM version of the GM8126 burn-in program.

- Step 1: Execute the “Metrowerks CodeWarrior” program and open the “burnin_ads_1.2.mcp” file residing in “<your directory>\burnin\ burnin\burnin_ads_1.2.mcp”
- Step 2: Double-click the mcp file to open the “fLib_ads_1.2.mcp” file (As shown in Figure 3-2)
- Step 3: Set to the “Ram” mode on the “burnin_ads_1.2.mcp” and “fLib_ads_1.2.mcp” windows (As shown in Figure 3-2)
- Step 4: Press the “F5” key on the keyboard to compile the “fLib_ads_1.2.mcp” file (Click the “fLib_ads_1.2.mcp” window to ensure that the fLib project is selected)
- Step 5: Click the burnin_ads_1.2.mcp window to select the burn-in project. Press “F5” key to compile burnin_ads_1.2.mcp

The AXF file will be created and saved in the “<your directory>\burnin\burnin\ burnin_ads_1.2_data\ram\burnin.axf” directory.

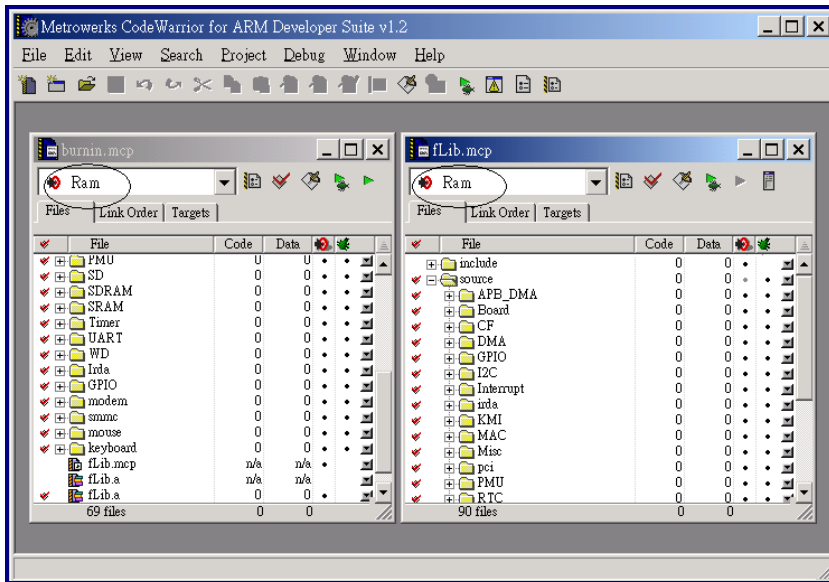


Figure 3-2. RAM Version

After successfully creating the RAM-version burn-in program, users can debug burnin.axf by using the AXD debugger. Please refer to the online documents of Metrowerks CodeWarrior for more detailed information.

Users may simultaneously press the “Ctrl” key and “-” key in the numeric keypad to remove the object codes. Once the project is rebuilt, ADS IDE will recompile all the source files.

3.3 ICE Options/Configure Processor

To use the AXD debugger, it should be configured with the following processor options:

- Semihosting on: UART outputs to ICE
- Semihosting off: UART outputs to the RS232 port
- Recommended setting: Semihosting off, vector cache clears all

Chapter 4

References

Users can refer to the following documents as reference:

- GM8126 User Guide
- GM8126 Data Sheet