Intel® EP80579 Software Drivers for Embedded Applications

Package Version 1.0.3

Release Notes

June 2010

Contents

Tables

Revision History

Date	Revision	Description
04 June 2010	014	Added "IXA00161211 - Security Vulnerability in Gigabit Ethernet Driver due to Ethernet frames that exceed MTU" on page 11
26 March 2010	013	Added the following: • "IXA00161206 - Two CAN nodes sending and receiving packets to one another hangs the system" on page 9 • "IXA00365821 - Disabling the CAN device in Device Manager causes increased CPU usage" on page 15
03 September 2009	012	• See Section 1.1, "New Features" on page 5 • Updated BIOS version: TRXTG064.ROM
23 June 2009	011	Added "IXA00343773 - Software lockup may occur in Embedded Gigabit Ethernet driver" on page 21
04 May 2009	010	Added Section 2, "Known Issues - Silicon" on page 6
28 April 2009	009	Modified BIOS version: TRXTG063.ROM
25 March 2009	008	Updates include: • Added support for FreeBSD* v7.1 • Modified BIOS version: TRXTG061.ROM
05 February 2009	007	Updated BIOS version: TRXTG062.ROM
18 December 2008	006	Updates include: • Change of operating system from Red Hat* to CentOS • Added support for FreeBSD* v6.3 • Updated BIOS version: TRXTG061.ROM
2 October 2008	005	Updated BIOS version: TRXTG060.ROM
23 September 2008	004	Updated BIOS version: TRXTG059.ROM
12 September 2008	003	Updated Package Versions: Embedded.X.1.0.70 (supports National Semiconductor* DP83848I PHY) and Embedded.L.1.0.75 (supports Broadcom* BCM5481 PHY)
22 August 2008	002	Updated BIOS version: TRXTG058.ROM
13 August 2008	001	Initial release of this document.

§ §

1 Description of Release

This document describes extensions to and deviations from the release functionality described in Intel® EP80579 Software Drivers for Embedded Applications Programmer's Guide and API Reference Manual.

For instructions on loading and running the release, see the Getting Started Guide for your operating system. See Section 5, "Related Documentation" on page 21 for details.

Note: Prior to installing the EP80579 software package, uninstall the previous installation of the software. Refer to the Getting Started Guide for instructions on uninstalling the software.

These release notes may also include known issues with third-party or reference platform components that affect the operation of the software.

Note: The "Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board" is referred to as "development board" throughout this document.

1.1 New Features

- CompactFlash Driver for CentOS 5.2 Linux*

    Note: The CompactFlash driver was tested with Transcend 2 GB 120x CompactFlash Cards that come with the development kit; if software vendor uses a different CompactFlash card, the CompactFlash driver might need additional changes by the software vendor to operate with that card.

Please refer to the Getting Started Guide for your operating system for additional information.

1.2 Supported Operating Systems

This software release has been validated with the following operating systems:

- CentOS 5.2 Linux*
- Microsoft* Windows* XP Embedded Service Pack 2
- FreeBSD* v7.1

1.3 Supported Component Versions

1.3.1 Version Numbering Scheme

The version numbering scheme used in this software release uses this naming convention: package.os.major.minor.maintenance-build

where:

- package can be one of the following:
    — Embedded
    — Security
    — Telephony
- os can be one of the following:
    — X = Windows XP Embedded*

— L = Linux*

— B = FreeBSD*

1.3.2 Package Versions

Table 1. **Package Versions**

Operating System	Package Version
Microsoft* Windows* XP Embedded Package	Embedded.X.1.0.3-127
Linux* Package	Embedded.L.1.0.3-144
FreeBSD* Package	Embedded.B.1.0.3-146

1.3.3 BIOS/Firmware Version

The term BIOS is used to refer to pre-boot firmware which could include legacy BIOS or Extensible Firmware Interface (EFI) compliant firmware.

BIOS Version: TRXTG064.ROM

2 Known Issues - Silicon

The Intel® EP80579 Integrated Processor Product Line Specification Update describes known silicon defects. Defects that are specific to Intel® EP80579 silicon are listed in the section called "Intel® EP80579 Integrated Processor Product Line Errata".

Table 2 lists defects identified with the Intel® EP80579 silicon that have software workarounds. If the workaround is implemented in the released software, it is indicated by **X** in the appropriate Operating System column in Table 2. For workaround details, see the Intel® EP80579 Integrated Processor Product Line Specification Update.

Table 2. **Intel® EP80579 Integrated Processor Product Line Errata**

No.[1]	Errata	Operating System		
		Linux*	FreeBSD*	Windows*
3	Gigabit Ethernet MAC Receive Timer interrupt problems	X	X	X
4	Gigabit Ethernet MAC Large Segment Offload (LSO) premature descriptor write back	X	N/A[2]	X
5	Gigabit Ethernet MAC XOFF from link partner can pause flow-control (XON/XOFF) transmission[3]	N/A	N/A	N/A
6	Gigabit Ethernet MAC transmit descriptor use of Report Status (RS) bit for non-data (Context & Null) descriptors	X	X	X
8	Gigabit Ethernet MAC legacy transmit descriptor write-back may occur before the packet data associated with the descriptor is fetched[4]		N/A[5]	
9	Gigabit Ethernet MAC may have EEPROM deadlock when using manual software EEPROM access	X	X	X

1. This is the same defect number assigned in the Intel® EP80579 Integrated Processor Product Line Specification Update.
2. LSO is not supported by the FreeBSD driver.
3. This issue depends on the link partner flow control settings. The Flow Control thresholds have been tested extensively with no observed issues.
4. The software workaround defined in the Specification Update is not currently implemented in the Linux or Windows Ethernet driver.
5. TCP Segmentation Offload is not supported by the FreeBSD driver.

3 Known Issues - Embedded

For supplementary information relating to the Known Issues, please refer to the following documents:

- Intel® EP80579 Software Drivers for Embedded Applications Programmer's Guide and API Reference Manual, Number: 320154

- Section 5, "Related Documentation" on page 21 lists other relevant documents

  Note: This release supports CentOS v5.2 Linux. If the Affected OS field in an errata table lists Red Hat Enterprise Linux 5.0, readers can assume the errata is also present under CentOS.

Table 3. Summary of Embedded Software Open Issues

3.1 IXA00058242 - Disabling/Enabling output pin causes board reset when using the Microsoft Windows XP Embedded Watchdog Timer driver

Title	Disabling/Enabling output pin causes board reset when using the Microsoft Windows XP Embedded Watchdog Timer driver
Reference #	IXA00058242
Description	If the Watchdog Timer enable pin is set to enable while the Watchdog Timer is actively counting down in watchdog mode during the first stage, the system will reset.
Implication	Unreliable results will occur when changing board setting while the Watchdog Timer driver has been engaged in a countdown cycle.

Title	Disabling/Enabling output pin causes board reset when using the Microsoft Windows XP Embedded Watchdog Timer driver (Continued)
Resolution	Do not enable or disable the Watchdog Timer enable pin while the Watchdog Timer is actively in a countdown cycle.
Affected OS	Windows XP Embedded
Driver/Module	Watchdog Timer Driver

3.2 IXA00058263 - SATA port 1 not showing populated when CD/DVD ROM attached

Title	SATA port 1 not showing populated when CD/DVD ROM attached
Reference #	IXA00058263
Description	In the BIOS Setup Menu under the IDE selection screen the 2nd SATA port (SATA 1) shows "Not Present" when a CD/DVD ROM is plugged into SATA port 1 and the IDE mode is AHCI. Hard Drives will appear correctly. However, the CD/DVD ROM appears to be fully functional and can be selected in the boot order.
Implication	In the AHCI mode for IDE, CD/DVD ROMs may not show up when plugged into the SATA ports. This is viewed in the Advanced->IDE Configuration Screen in the BIOS setup menu. Functionality of the device is not impacted and will still be selectable in the boot order. The CD/DVD ROM devices tested were Plextor PX-72Sa and LG GSA-H62L.
Resolution	No work around available.
Affected OS	FreeBSD 6.2 Red Hat Enterprise Linux 5.0 Windows XP Embedded
Driver/Module	Driver-General

3.3 IXA00058281 - SMBus Driver cannot be enabled/utilized on Microsoft Windows XP Embedded

Title	SMBus Driver cannot be enabled/utilized on Microsoft Windows XP Embedded
Reference #	IXA00058281
Description	It is not possible to enable or use the SMBus driver on Microsoft Windows XP Embedded unless specific components are included in the target image creation process. 'PCI Standard ISA bridge' and 'PCI Standard Host CPU bridge' components need to be included when creating target Microsoft Windows XP Embedded image in order to use the SMBus driver.
Implication	It is not possible to enable or use the SMBus driver on Microsoft Windows XP Embedded unless specific components are included in the target image creation process.
Resolution	Include 'PCI Standard ISA bridge' and 'PCI Standard Host CPU bridge' components when creating target Windows XP Embedded image. This will allow SMBus to be enabled and utilized under Microsoft Windows XP Embedded. Refer to Intel® EP80579 Software Drivers for Embedded Applications on Microsoft Windows* XP Embedded Getting Started Guide for additional information on target image creation.
Affected OS	Windows XP Embedded
Driver/Module	SMBus Controller Driver

3.4 IXA00160881 - Using the smbmsg utility to probe the SMBus may hang the Intel® EP80579 Development Board on FreeBSD

Title	Using the smbmsg utility to probe the SMBus may hang the Intel® EP80579 Development Board on FreeBSD
Reference #	IXA00160881
Description	If the FreeBSD smbmsg utility is used to probe the SMBus, the system may hang. Note that probing the SMBus is risky. Individual devices can perform unwanted actions upon receiving the probe request message. For example, if a particular SMBus device considers any write operation issued to it as a request to power off the system, the probing would trigger this action. For additional information please refer to the smbmsg man page (http://www.ipnom.com/FreeBSD-Man-Pages/smbmsg.8.html)
Implication	Using the smbmsg utility to probe the SMBus may hang the development board on FreeBSD.
Resolution	It is not advisable to use the smbmsg utility to probe SMBus.
Affected OS	FreeBSD 6.2
Driver/Module	SMBus Controller Driver

3.5 IXA00161154 - Booting with EFI from Compact Flash on Intel® EP80579 Development Board fails with 1 GB DIMM installed

Title	Booting with EFI from Compact Flash on Intel® EP80579 Development Board fails with 1 GB DIMM installed
Reference #	IXA00161154
Description	Booting with EFI from CompactFlash on Intel® EP80579 Development Board fails with 1 GB DIMM installed. This problem is not seen when 512 MB DIMM is installed or when legacy boot is used with 1 GB DIMM installed.
Implication	It is not possible to perform EFI boot from Compact Flash on Intel® EP80579 Development Board with 1 GB DIMM installed.
Resolution	Booting from Compact Flash can be done in legacy mode with 1 GB DIMM installed or EFI Boot with 512 MB DIMM installed.
Affected OS	CentOS 5.2 Linux
Driver/Module	Intel® EP80579 Development Board BIOS

3.6 IXA00161206 - Two CAN nodes sending and receiving packets to one another hangs the system

Title	Two CAN nodes sending and receiving packets to one another hangs the system
Reference #	IXA00161206
Description	System hang could happen when a CAN node transmits data and does not receive a reply from the link partner. In this case, the transmit thread hangs waiting for the reply.
Implication	Observed side effects are as follows: 1. System becomes unresponsive and it cannot be shut down. 2. Driver's transmit thread hangs and attempts at disabling device from device manager fail.

Title	Two CAN nodes sending and receiving packets to one another hangs the system (Continued)

Resolution	Two code changes are required as described below.
	1. In the file can_main.h, add the following line:
	#define TX_TIMEOUT -20000000 /* Transmit time out = 2 sec */
	2. In the file can_main.c, replace the CANEvtIoWrite() function with the following:
	
	``
	VOID
	CANEvtIoWrite(
	   WDFQUEUE Queue,
	   WDFREQUEST Request,
	   size_t Length
	   )
	{
	   can_os_t *can_os;
	   icp_can_msg_t *msg = NULL;
	   int err;
	   NTSTATUS status = STATUS_UNSUCCESSFUL;
	   ULONG_PTR bytes_written = 0;
	   unsigned int req_len;
	    LARGE_INTEGER timeout;
	   can_os = WdfObjectGet_can_os_t(WdfIoQueueGetDevice(Queue));
	    if (can_os->is_suspending) {
	       goto Error;
	    }
	   status = WdfRequestRetrieveInputBuffer(Request, sizeof(icp_can_msg_t),
	      (PVOID) &msg, &req_len);
	   if (!NT_SUCCESS(status)) {
	      goto Error;
	   }
	   err = icp_can_msg_tx(can_os->can, msg);
	    if ((!err) && (can_os->block_mode)) {
	       timeout.QuadPart = TX_TIMEOUT;
	
	       status = KeWaitForSingleObject(&(can_os->tx_event), Executive,
	KernelMode, TRUE, &timeout);
	      KeResetEvent(&(can_os->tx_event));
	       if (!NT_SUCCESS(status)) {
	          /* Receive timed-out or error, try again */
	          status = STATUS_CANCELLED;
	      }
	   }
	   if (!err) {
	      bytes_written = sizeof(icp_can_msg_t);
	      status = STATUS_SUCCESS;
	   }
	Error:
	   WdfRequestCompleteWithInformation(Request, status, bytes_written);
	}
	``

Affected OS	Windows XP Embedded WindowsXP
Driver/Module	Controller Area Network Driver

3.7 IXA00161211 - Security Vulnerability in Gigabit Ethernet Driver due to Ethernet frames that exceed MTU

Title	Security Vulnerability in Gigabit Ethernet Driver due to Ethernet frames that exceed MTU
Reference #	IXA00161211
Description	A security vulnerability exists in the Gigabit Ethernet Driver. The driver allows remote DOS attack through careful selection of frame size in relation to interface MTU, which causes a denial of service (panic), via a crafted frame size.
Implication	Since the Intel® EP80579 Linux Gigabit Ethernet driver does not support receiving packets that span multiple Rx buffers, it checks the End of Packet bit of every frame, and discards it if it is not set. This creates a situation where the first part of a spanning packet is discarded, but the second part is not (since it is the end of packet and it passes the EOP bit test).
	If the second part of the frame is small (4 bytes or less), the driver subtracts 4 from it to remove its CRC, underflow the length, and winds up in skb_over_panic, when the driver tries to skb_put a huge number of bytes into the skb. This allows a remote DOS attack through careful selection of frame size in relation to interface MTU, which causes a denial of service (panic), via a crafted frame size.
	Additional information on this defect is available at:
	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-4536

Title	Security Vulnerability in Gigabit Ethernet Driver due to Ethernet frames that exceed MTU (Continued)

Resolution	The following updates should be applied to the driver to address the security vulnerability.
	Corrective Action #1:
	Embedded/src/GbE/iegbe_main.c:
	In the function iegbe_clean_rx_irq() find the conditional code fragment:
	`if(unlikely(!(rx_desc->status & E1000_RXD_STAT_EOP))) {`
	`    /* All receives must fit into a single buffer */`
	`    E1000_DBG("%s: Receive packet consumed multiple"`
	`            " buffers\n", netdev->name);`
	`    dev_kfree_skb_irq(skb);`
	`    goto next_desc;`
	`}`
	Replace it with:
	`/* !EOP means multiple descriptors were used to store a`
	` * single packet, if that is the case we need to toss it.`
	` * In fact, we need to toss every packet with the EOP bit`
	` * clear and the next frame that _does_ have the EOP bit set,`
	` * as it is by definition only a frame fragment */`
	`if (unlikely(!(rx_desc->status & E1000_RXD_STAT_EOP)))`
	`        adapter->discarding = TRUE;`
	`if (adapter->discarding) {`
	`    /* All receives must fit into a single buffer */`
	`    E1000_DBG("%s: Receive packet consumed multiple"`
	`                " buffers\n", netdev->name);`
	`    if (rx_desc->status & E1000_RXD_STAT_EOP)`
	`        adapter->discarding = FALSE;`
	`    dev_kfree_skb_irq(skb);`
	`    goto next_desc;`
	`}`
	Corrective Action #2:
	Embedded/src/GbE/iegbe.h:
	In the iegbe_adapter structure, add the discarding element to the end of the structure.
	Ensure that the element is not added within any of the conditional preprocessor statements:
	`struct iegbe_adapter {`
	`    struct timer_list tx_fifo_stall_timer;`
	`    struct timer_list watchdog_timer;`
	`    .`
	`    .`
	`    .`
	`    uint32_t pci_state[16];`
	`    int msg_enable;`
	`#ifdef CONFIG_PCI_MSI`
	`    boolean_t have_msi;`
	`#endif`
	`    boolean_t discarding;`
	`}`

Affected OS	Red Hat Enterprise Linux 5.0 CentOS 5.2 Linux
Driver/Module	Gigabit Ethernet Controller Driver

3.8 IXA00178488 - No option is available to enable VLAN in Microsoft Windows XP Embedded

Title	No option is available to enable VLAN in Microsoft Windows XP Embedded
Reference #	IXA00178488
Description	It is not currently possible to enable VLAN capability in Microsoft Windows XP Embedded.
Implication	No VLAN capability currently exists with EP80579 embedded software release for Microsoft Windows XP Embedded.
Resolution	No work around available.
Affected OS	Windows XP Embedded
Driver/Module	Gigabit Ethernet Controller Driver

3.9 IXA00179772 - IDE mode within the BIOS setup menu should be set to "AHCI" for optimal performance on Red Hat Enterprise Linux 5.0

Title	IDE mode within the BIOS setup menu should be set to "AHCI" for optimal performance on Red Hat Enterprise Linux 5.0
Reference #	IXA00179772
Description	The performance of the Embedded Gigabit driver is significantly lower when the IDE mode is set to "Legacy". AHCI inherently provides higher performance and setting the system to AHCI mode will result in superior performance results, provided the hard drive used in the system supports AHCI.
Implication	System performance is significantly lower when IDE mode is set to "Legacy" mode.
Resolution	AHCI inherently provides higher performance. Setting the system IDE mode to "AHCI" will result in superior performance results. Refer to the chapter titled Pre-boot (BIOS) Firmware within the Getting Started Guide for instructions to toggle the IDE mode to "AHCI" in the BIOS setup menu.
Affected OS	Red Hat Enterprise Linux 5.0
Driver/Module	Gigabit Ethernet Controller Driver

3.10 IXA00216017 - CompactFlash cards are not supported with embedded software drivers

Title	CompactFlash cards are not supported with embedded software drivers
Reference #	IXA00216017
Description	Currently CompactFlash on the Lower Expansion Bus of the Intel EP80579 Development Board are not supported with drivers in any of the embedded software releases.
Implication	No CompactFlash driver support is available for use of the CompactFlash.
Resolution	As of Embedded Release 1.0.3, CompactFlash is supported under Linux only. No work around available for FreeBSD or Windows XP Embedded.
Affected OS	FreeBSD 6.2 Windows XP Embedded
Driver/Module	Not Applicable

3.11   IXA00234579 - The iegbe.inf file references non-POR operating system and claims support of Intel PRO/1000 adapters

Title	The iegbe.inf file references non-POR operating system and claims support of Intel PRO/1000 adapters
Reference #	IXA00234579
Description	The iegbe.inf file indicates support for operating systems other than Microsoft Windows XP Embedded. It also indicates support for many of the Intel(R) PRO/1000 Adapters. In fact, this .inf file only supports the Intel® EP80579 Gigabit Ethernet Controllers. The only operating system that the iegbe.inf and Intel® EP80579 Gigabit Ethernet Controller Driver support is Microsoft Windows XP Embedded. The driver is not interchangeable with any Intel LAN Access Division, Intel(R) PRO/1000 Gigabit Adapters. Intel LAN Access Division developed software was utilized to develop a gigabit adapter driver to function with the Intel® EP80579 Gigabit Ethernet Controllers only.
Implication	The customer may be misled into believing that the Intel® EP80579 Gigabit Ethernet Controller Driver supports multiple operating systems and a host of other Intel(R) PRO/1000 Adapters. This driver only supports Microsoft Windows XP Embedded and the Intel® EP80579 Gigabit Ethernet Controllers.
Resolution	No work around available.
Affected OS	Windows XP Embedded
Driver/Module	Gigabit Ethernet Controller Driver

3.12   IXA00241849 - Installation issues observed on Red Hat Enterprise Linux 5 using DVD SATA drives with IDE mode set to AHCI

Title	Installation issues observed on Red Hat Enterprise Linux 5 using DVD SATA drives with IDE mode set to AHCI
Reference #	IXA00241849
Description	When IDE mode is set to AHCI, installation of Red Hat Enterprise Linux using some DVD SATA drives will fail. During the installation process the message "Loading AHCI driver" is displayed, and the installation hangs. This has been observed on some DVD SATA drives.
Implication	When IDE mode is set to AHCI, installation of Red Hat Enterprise Linux using some DVD SATA drives will fail.
Resolution	If DVD SATA drives exhibit this behavior, perform installation with IDE mode set to Legacy. If AHCI is desired, perform installation using USB DVD, USB CD drive, or other DVD SATA drive.
Affected OS	Red Hat Enterprise Linux 5.0
Driver/Module	SATA

3.13   IXA00309267 - UART port may not come out of S3 hibernation

Title	UART port may not come out of S3 hibernation
Reference #	IXA00309267
Description	When the development board is suspended via the command 'echo -n mem > /sys/power/state' and brought back into service via pressing the power button, the UART may not come back into service. Crash trace is reported in /var/log/messages relating to IRQ 9 which is associated with ACPI as reported /proc/interrupts
Implication	Affects Embedded release packages only because power management feature is not available on accelerated software.

Title	UART port may not come out of S3 hibernation (Continued)
Resolution	To recover, reboot the system.
Affected OS	Red Hat Enterprise Linux 5.0
Driver/Module	Intel® EP80579 Development Board BIOS

3.14 IXA00320219 - Decreasing value of the metric 'MemFree' in /proc/meminfo may not imply a memory loss in Linux or the application

Title	Decreasing value of the metric 'MemFree' in /proc/meminfo may not imply a memory loss in Linux or the application
Reference #	IXA00320219
Description	When the performance sample code was run for several days, the metric 'MemFree' in /proc/meminfo indicated a non-trivial loss in free memory. When memory loss was ruled out in Intel drivers and sample code, the same metric was monitored on an idle Linux system over a similar period of time. Once again, the metric pointed to a memory loss. Further investigation on the web, indicated that the apparent loss in memory reported in /proc/meminfo is caused by the kernel not freeing memory released by applications. The kernel does not release memory to realize other optimizations. However, this unused memory can be released non-destructively by setting the tunable parameter as described in the Resolution section.
Implication	When memory loss is suspected or reported by memory tools while running applications, performance suites, or stress tests, the customer should use prescriptions in the Resolution section to rule out memory grabbed by kernel as described in this erratum.

Resolution	Writing to /proc/sys/vm/drop will cause the kernel to drop clean caches, dentries, and inodes from memory, causing that memory to become free.
	To free pagecache:
	  echo 1 > /proc/sys/vm/drop_caches
	To free dentries and inodes:
	  echo 2 > /proc/sys/vm/drop_caches
	To free pagecache, dentries and inodes:
	  echo 3 > /proc/sys/vm/drop_caches
	As this is a non-destructive operation, and dirty objects are not freeable, the user should run "sync" first in order to make sure all cached objects are freed.
	This tunable field was added in kernel version 2.6.16.

Affected OS	Red Hat Enterprise Linux 5.0
Driver/Module	Opensource Software

3.15 IXA00365821 - Disabling the CAN device in Device Manager causes increased CPU usage

Title	Disabling the CAN device in Device Manager causes increased CPU usage
Reference #	IXA00365821
Description	When the CAN Device is disabled in Device Manager, CPU usage is increased and not recovered.
Implication	Observed side effects are as follows: 1. When CAN drivers are disabled from device manager, CPU utilization increases and remains high, ranging between 4% and 50%. 2. System may become unresponsive.

Title	Disabling the CAN device in Device Manager causes increased CPU usage

Resolution	In the file can_main.c, replace the CANEvtDeviceD0Exit() function with the following:
	NTSTATUS
	 CANEvtDeviceD0Exit(
	  IN WDFDEVICE Device,
	  IN WDF_POWER_DEVICE_STATE TargetState
	  )
	{
	  unsigned int buffer_status;
	  KEVENT tmp_event;
	  can_os_t *can_os;
	  LARGE_INTEGER timeout;
	  unsigned int i;
	  unsigned int int_status;
	  unsigned short driver_ints;
	  PDEVICE_OBJECT pdo;
	
	  DbgPrint("CANEvtDeviceD0Exit … entered.\n");
	
	  can_os = WdfObjectGet_can_os_t(Device);
	  /* Indicate that we are suspending */
	  can_os->is_suspending = 1;
	  /* Set Read event so, suspend can continue */
	  KeSetEvent(&(can_os->rx_event), IO_NO_INCREMENT, FALSE);
	  if (can_os->ref_cnt) {
	    /* Save the Run Mode. */
	    icp_can_get_run_mode(can_os->can, &(can_os->run_mode));
	  }
	  /* Stop the CAN controller */
	  icp_can_set_run_mode(can_os->can, ICP_CAN_STOP);
	 /* Clear pending interrupts */
	 int_status = icp_can_int_pending(can_os->can);
	 if (int_status) {
	   icp_can_int_clr(can_os->can, int_status);
	 }
	 /* Mask interrupts to target the driver */
	 pdo = WdfDeviceWdmGetDeviceObject(can_os->hDevice);
	 driver_ints = 0x0;
	 PCIWriteConfigWord(pdo, 0xE8, &driver_ints);
	 if (can_os->ref_cnt) {
	   /* Save interrupt configuration and then disable them */
	   icp_can_get_int_enables(can_os->can, &(can_os->int_enables));
	   icp_can_set_int_enables(can_os->can, ICP_CAN_DISABLE);
	   /* Save Tx buffer enable state */
	   for (i=0; i<NUM_TX_BUFFS; i++) {
	    icp_can_get_tx_enable(can_os->can, i, &(can_os->tx_enable[i]));
	   }

Title	Disabling the CAN device in Device Manager causes increased CPU usage
| Resolution, continued | ```
    /* Disable all Transmit buffers */
    icp_can_tx_disable_all(can_os->can);
    /* Save Rx buffer enable state */
    for (i=0; i<NUM_RX_BUFFS; i++) {
        icp_can_get_rx_enable(can_os->can, i, &(can_os->rx_enable[i]));
        icp_can_get_rx_buffer_link(can_os->can, i, &(can_os->rx_link[i]));
          /* Save Rx Filters */
        can_os->rx_filter[i].num = i;
          icp_can_get_rx_filter(can_os->can, &(can_os->rx_filter[i]));
    }
     /* Disable all Receive buffers */
     icp_can_rx_disable_all(can_os->can);
    /* Save Context */
    icp_can_get_baud(can_os->can, &(can_os->timing));
    icp_can_get_listen_mode(can_os->can, &(can_os->listen_mode));
    icp_can_get_arbiter_mode(can_os->can, &(can_os->arbiter_mode));
     /* Free any waiting write threads */
     KeResetEvent(&(can_os->tx_event));
  }

  return STATUS_SUCCESS;
}
``` |
| Affected OS | Windows XP Embedded
WindowsXP |
| Driver/Module | Controller Area Network Driver |

4 Resolved Issues - Embedded

Issues that have been resolved in previous versions of the software are included here.

Table 4.   **Summary of Embedded Software Resolved Issues**

4.1 IXA00058114 - Embedded Gigabit Ethernet driver Load/Unload memory leak on Red Hat Enterprise Linux v5.0

Title	Embedded Gigabit Ethernet driver Load/Unload memory leak on Red Hat Enterprise Linux v5.0
Reference #	IXA00058114
Description	A memory leak has been found while loading/unloading the Gigabit Ethernet driver over 15000 times in a 20 hour period without a reboot of the development board.
Implication	Although any memory leak is a concern, this is an extreme situation that should not impact any known usage model with the embedded software Linux release.
Resolution	This issue was determined to be an Operating System feature. See IXA00320219 for details
Affected OS	Red Hat Enterprise Linux 5.0
Driver/Module	Gigabit Ethernet Controller Driver

4.2 IXA00058233 - Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board will hang on reset command in FreeBSD

Title	Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board will hang on reset command in FreeBSD
Reference #	IXA00058233
Description	Commands to reboot the system under FreeBSD result in a system hang on the Development Board. This has been observed with 'reboot' and 'shutdown -r now' commands. The operating system completes the shutdown process, but hangs in the reset mechanism. There have been several reports of this in the public forums and is apparently caused by a defect in the operating system.
Implication	The board must be manually reset.
Resolution	This issue was resolved in FreeBSD 6.3
Affected OS	FreeBSD 6.2
Driver/Module	Not Applicable

4.3 IXA00058236 - Gigabit Ethernet devices do not appear after reboot on FreeBSD

Title	Gigabit Ethernet devices do not appear after reboot on FreeBSD
Reference #	IXA00058236
Description	When there is a dependency between drivers, FreeBSD detects the dependency when loading the drivers manually using kldload command and loads the drivers accordingly. However, when the task is automated to load dependent drivers after each boot, the OS seems to load the drivers in ascending order of PCI bus:device:function:number, as it enumerates devices. The GbE adapters and GCU are on the same PCI bus, but GCU has a function numerically larger than the adapters, hence its driver gets loaded only after the GbE driver loads (which fails to initialize the hardware because it is dependent on GCU).
Implication	After rebooting FreeBSD, Gigabit Ethernet devices do not appear.
Resolution	This issue was resolved in Embedded.B.1.0.1
Affected OS	FreeBSD 6.2
Driver/Module	Gigabit Ethernet Controller Driver

4.4 IXA00160925 - Manual setting of full-duplex mode of Embedded Gigabit Ethernet driver defaults to half-duplex on Microsoft Windows XP Embedded & Red Hat Enterprise Linux 5.0

Title	**Manual setting of full-duplex mode of Embedded Gigabit Ethernet driver defaults to half-duplex on Microsoft Windows XP Embedded & Red Hat Enterprise Linux 5.0**
Reference #	IXA00160925
Description	When Embedded Gigabit Ethernet driver is manually configured to run at 10Mbs or 100Mbs full-duplex, the operation mode defaults to half-duplex. This behavior has been observed under Microsoft Windows XP Embedded and Red Hat Enterprise Linux 5.0. Autonegotiation allows driver to run at 10Mbs and 100Mbs full-duplex.
Implication	It is not possible to manually set the duplex mode to full-duplex with the Embedded Gigabit Ethernet driver. Attempts to do this will result in duplex mode being set to half-duplex. The system will still function, but will do so in half-duplex mode.
Resolution	The errata text description and implication incorrectly states that it is not possible to manually set duplex mode and speed with the Embedded Gigabit Ethernet driver. Earlier testing included manually setting duplex mode and speed on the Embedded Gigabit Ethernet driver and then connecting to a link partner with autonegotiation enabled. When a link partner is connected to a device that is not using autonegotiation, the autonegotiation process fails. The autonegotiation end of the connection is still able to correctly detect the speed of the other end, but cannot detect the duplex mode. Industry standard requires use of half-duplex mode in these conditions. Because the link partner was placed in half-duplex mode, the Embedded Gigabit Ethernet driver was incorrectly identified as the source of the issue and errata was created to document the behavior. There is no issue with manual setting of duplex mode and speed with the Embedded Gigabit Ethernet driver.
Affected OS	Red Hat Enterprise Linux 5.0 Windows XP Embedded
Driver/Module	Gigabit Ethernet Controller Driver

4.5 IXA00160932 - Watchdog Timer driver counters are reset after a Power Management S3 Suspend action on Microsoft Windows XP Embedded

Title	**Watchdog Timer driver counters are reset after a Power Management S3 Suspend action on Microsoft Windows XP Embedded**
Reference #	IXA00160932
Description	Timer driver counters are reset after a Power Management S3 Suspend action. This issue is present on Microsoft Windows XP Embedded.
Implication	Watchdog Timer is typically used for critical functionality that will not allow for systems to go into S3. In addition, the Watchdog Timer driver will be functional as designed when the system is fully on. The only item of potential concern is that the Watchdog Timer driver counters will be reset when a system goes from S3 to on. At that time Watchdog Timer driver will work as designed.
Resolution	This issue was resolved in Embedded.X.1.0.2.
Affected OS	Windows XP Embedded
Driver/Module	Watchdog Timer Driver

4.6   IXA00160970 - Make install targets do not work when executed within component directory on FreeBSD

Title	Make install targets do not work when executed within component directory on FreeBSD
Reference #	IXA00160970
Description	The Makefiles for 1588, CAN, EDMA, GCU, GPIO, and WDT each contain a "install" target. There is an issue with the "kldload $(KMOD).ko line. Executing "make install" when in component directory (such as 1588) the command fails with an error message similar to following: "can't load timesync.ko: No such file or directory" This error message is returned, because in FreeBSD the current directory is not part of the path for security reasons, so the kldload command does not know where the .ko image comes from.
Implication	Individual embedded drivers will not be installed.
Resolution	This issue was resolved in Embedded.B.1.0.1
Affected OS	FreeBSD 6.2
Driver/Module	All

4.7   IXA00161027 - Autonegotiate fails to switch modes in Gigabit Ethernet driver

Title	Autonegotiate fails to switch modes in Gigabit Ethernet driver
Reference #	IXA00161027
Description	When an ethernet interface on the EP80579 is set to autonegotiate and the link partner changes speed, the interface on the EP80579 does not autonegotiate.
Implication	Ifconfig reports the previous speed and there is no network connectivity on that interface.
Resolution	This issue was resolved in Embedded.B.1.0.2
Affected OS	FreeBSD 6.2 FreeBSD 6.3
Driver/Module	Gigabit Ethernet Controller Driver

4.8   IXA00206755 - Driver displays install warnings with Red Hat Enterprise Linux 5.0 distribution - 2.6.18 kernel

Title	Driver displays install warnings with Red Hat Enterprise Linux 5.0 distribution - 2.6.18 kernel
Reference #	IXA00206755
Description	When building the Intel® EP80579 Software Drivers for Embedded Applications using Red Hat Enterprise Linux 5.0, warnings are displayed as follows: Warning: vmlinux - Section mismatch: reference to .exit.text: from .smp_alternatives between '__smp_alt_begin' (at offset...) and '__smp_locks_end'
Implication	These warning messages are from the Red Hat Enterprise Linux 5.0 distribution. They are not from the embedded software drivers. These warnings will not interfere with the building and function of the embedded software drivers
Resolution	Operating System warning messages are not displayed because Red Hat Enterprise Linux is not supported.
Affected OS	Red Hat Enterprise Linux 5.0
Driver/Module	All

4.9 IXA00343773 - Software lockup may occur in Embedded Gigabit Ethernet driver

Title	Software lockup may occur in Embedded Gigabit Ethernet driver
Reference #	IXA00343773
Description	The Embedded Gigabit Ethernet driver uses spin_lock_bh() to protect the MDIO registers. This can cause a soft-lockup issue. The lockup would look similar to: BUG: soft lockup - CPU#0 stuck for 10s! [netstat:4281] Pid: 4281, comm: netstat EIP: 0060:[<c0609590>] CPU: 0 EIP is at _spin_lock+0x7/0xf EFLAGS: 00000286 Tainted: PF (2.6.18-GA102_OCF #1) EAX: f48c2f00 EBX: 00000002 ECX: c072bf8e EDX: f152be40 ESI: 00000011 EDI: c072bf8e EBP: e604a7c0 DS: 007b ES: 007b CR0: 8005003b CR2: b7fa6000 CR3: 2582a000 CR4: 000006d0 [<f48c10b6>] gcu_get_adapter+0x16/0x20 [gcu] [<f48c14e3>] gcu_read_eth_phy+0x23/0x130 [gcu] [<f494c6c5>] iegbe_oem_phy_is_link_up+0x55/0x90 [iegbe] [<f49432a9>] iegbe_watchdog+0x419/0x5c0 [iegbe] [<f4942e90>] iegbe_watchdog+0x0/0x5c0 [iegbe] [<c042dc29>] run_timer_softirq+0xfb/0x151 [<c042a77a>] __do_softirq+0x5a/0xbb [<c0407461>] do_softirq+0x52/0x9d [<c0407406>] do_IRQ+0xa5/0xae [<c040592e>] common_interrupt+0x1a/0x20 [<c04e5edd>] delay_tsc+0x9/0x13 [<c04e5f10>] __delay+0x6/0x7 [<f48c1519>] gcu_read_eth_phy+0x59/0x130 [gcu] [<f4946f63>] iegbe_read_phy_reg+0x163/0x180 [iegbe] [<f493f325>] iegbe_update_stats+0x855/0x880 [iegbe] [<f4941a56>] iegbe_get_stats+0x16/0x20 [iegbe] [<c05adeed>] dev_seq_show+0x22/0x8e [<c048bdae>] seq_read+0x191/0x273 [<c048bc1d>] seq_read+0x0/0x273 [<c0471174>] vfs_read+0x9f/0x141 [<c04715c2>] sys_read+0x3c/0x63 [<c0404eff>] syscall_call+0x7/0xb
Implication	The Gigabit Ethernet driver's use of spin_lock_bh() to protect the MDIO registers can cause a soft-lockup issue.
Resolution	This issue was resolved with Embedded.L.1.0.3
Affected OS	CentOS 5.2 Linux Red Hat Enterprise Linux 5.0
Driver/Module	Gigabit Ethernet Controller Driver

5 Related Documentation

5.1 Where to Find Current Software and Documentation

The software release and associated collateral can be found on the Hardware Design resource center.

1. In a web browser, go to http://www.intel.com/go/soc

2. For Software and pre-boot firmware: Click on "Tools & Software" tab.

3. For Documentation: Click on "Technical Documents" tab.

5.2 Embedded Documents

The following documents provide more information about the software provided in this release:

Document Name	Number
Intel® EP80579 Software Drivers for Embedded Applications Programmer's Guide and API Reference Manual	320154
Intel® EP80579 Software Drivers for Embedded Applications on Linux* Getting Started Guide	320151
Intel® EP80579 Software Drivers for Embedded Applications on FreeBSD* Getting Started Guide	320152
Intel® EP80579 Software Drivers for Embedded Applications on Microsoft* Windows* XP Embedded Getting Started Guide	320153
Software for Intel® EP80579 Integrated Processor Product Line PHY Porting Guide	320203
Ethernet PHY Selection Criteria for the Intel® EP80579 Integrated Processor Product Line Application Note	320254
Intel® EP80579 Integrated Processor Product Line Datasheet	320066
Intel® EP80579 Integrated Processor Product Line Specification Update	320176

§ §