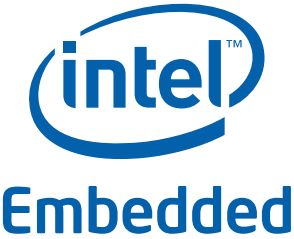


Intel® Platform Controller Hub EG20T

UART Controller Driver for Windows* Programmer's Guide

March 2011

Order Number: 324261-002US

Contents

Figures

Tables

Revision History

Date	Revision	Description
March 2011	002	Updated Section 2.0, “Operating System (OS) Support” on page 6
September 2010	001	Initial release

1.0 Introduction

This document provides the programming details of the Intel® Platform Controller Hub EG20T UART Controller driver for Windows*. This includes the information about the interfaces exposed by the driver and how to use those interfaces.

This document also covers the DMA driver API programming details which are used by the UART driver for data transmission at higher baud rates.

2.0 Operating System (OS) Support

The UART driver is supported by the following operating systems:

No	OS	Notes
1	Microsoft Windows XP*	Service Pack 3
2	Windows Embedded Standard*	2009
3	Windows Embedded POSReady*	2009
4	Microsoft Windows 7*	
5	Windows Embedded Standard7	

3.0 Dependencies

This driver is dependent upon appropriate OS driver installation and the Intel® Platform Controller Hub EG20T DMA Controller Driver and Packet HUB Driver.

The UART serial driver uses the DMA driver for data transfer at higher baud rates. The Intel® Platform Controller Hub EG20T DMA driver must be installed before enabling the DMA function of the Intel® Platform Controller Hub EG20T UART serial driver.

To enable higher baud rates (over 115 kbps), the UART clock must be configured with the Intel® Platform Controller Hub EG20T Packet HUB Driver. Please refer to Section 4.2 for information on how to configure the UART clock.

Please refer to the *Intel® Platform Controller Hub EG20T Direct Memory Access (DMA) Controller Driver for Windows* Programmer's Guide* for more information on DMA APIs.

4.0 Serial Driver API Details

This section describes details of interfaces exposed by the UART driver. Win32 Standard Serial communication APIs can be used for communicating with the UART serial driver.

4.1 Features

- Full-duplex buffering
- Full status reporting
- Reduces CPU interrupts caused by 256-bytes (UART0) or 64-bytes (UART1-3) transmit and receive FIFOs
- Independent control of the following: transmission interrupt, reception interrupt, line status interrupt, and FIFOs
- Programmable serial interface:
  - 5-, 6-, 7-, or 8-bit character
  - Generation and verification of odd parity, even parity, or no parity
  - 1, 1.5, and 2 stop bits
- Programmable baud rate generator:
  - UART0: from 300 bps to 4 Mbps
  - UART1-3: from 300 bps to 1 Mbps
- Equipped with DMA interface

4.2 Driver Configuration

The driver can be configured with the following entries in the iohserial.inf file or with the corresponding UART driver registry.

Table 1. Driver Configuration

No	Entry Name	Description
1	ClockRate	Configure the UART clock. Possible values are 1843200(1.8Mhz), 48000000(48Mhz) and 64000000(64Mhz). The UART clock value is dependent on the clock value set by the Packet Hub driver. See Table 2 for the configuration values.
2	Enable DMA	Set to 1 to enable the UART over DMA. Set 0 for normal UART operation

The following table lists the configuration information used by the packet hub driver and the UART driver to provide support for different UART clock rates. In order to set a particular clock, set the registry value with the corresponding values listed in the following table using the regedit tool.

Table 2. Driver Configuration

UART Clock	' ClockRate ' (UART Driver)	' UARTClock ' (Packet Hub driver)	Remarks
1.8432 MHz	0x1C2000	0x000C	Up to 115 Kbps
48 MHz	0x2DC6C00	0x010C	Up to 3 Mbps
64 MHz	0x3D09000	0x6510	Up to 4 Mbps

The above configurations can be set after the UART, DMA and Packet HUB driver installation by modifying the value in the below registry entries:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\PCI\VEN_8086&DEV_8811&SUBSYS_00000000&REV_00\5&335da31&0&5100B8\Device Parameters\ClockRate
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\PCI\VEN_8086&DEV_8811&SUBSYS_00000000&REV_00\5&335da31&0&5100B8\Device Parameters\"Enable DMA"
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\PCI\VEN_8086&DEV_8801&SUBSYS_00000000&REV_00\5&335da31&0&0000B8\Device Parameters\UARTClock

For example:

Setting up a 4 Mbps UART clock:

```
Windows Registry Editor Version 5.1
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\PCI\VEN_8086&DEV_8811&SUBSYS_00000000&REV_00\5&335da31&0&5100B8\Device Parameters]
"PortName"="COM3"
"PollingPeriod"=dword:00000000
"ClockRate"=dword:03d09000
"EnablePowerManagement"=dword:00000001
"Modem Enable"=dword:00000001
"Enable DMA"=dword:00000001
"TL16C550C Auto Flow Control"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\PCI\VEN_8086&DEV_8801&SUBSYS_00000000&REV_00\5&335da31&0&0000B8\Device Parameters]
"SomeConfig"=dword:00000004
"UARTClock"=dword:00006510
"CANClock"=dword:00000012
```

Go to start-> run-> regedit, traverse to the path stated above and edit the entries' values accordingly.

4.3 Interface Details

Refer to the MSDN page for information on the Serial Communication API:

http://msdn.microsoft.com/en-us/library/ms810467.aspx

4.4 Error Handling

Since the IOCTL command is implemented using the Windows* API, the return value of the call is dependent on and defined by the OS. On Windows*, the return value is a non-zero value. If the error is detected within or outside the driver, an appropriate system defined value is returned by the driver.

5.0 Programming Guide

This section contains the following information:

- Programming details of serial communication using the Win32 API
- Exported functions used inside the UART driver to communicate with the Intel® PCH EG20T DMA controller driver to perform data transfer in higher baud rates

5.1 Programming Using Serial Communication API

Detailed information on the programming of the standard UART controller is provided at the following MSDN web site address. Refer to this link to understand the details on Serial Communication in Win32:

http://msdn.microsoft.com/en-us/library/ms810467.aspx

5.2 DMA Interface Usage Details

This section provides information on the Intel® PCH EG20T DMA driver exported functions used by the UART driver to perform data transfer at higher baud rates. Table 3 provides details of the functions exported by the Intel® PCH EG20T DMA driver.

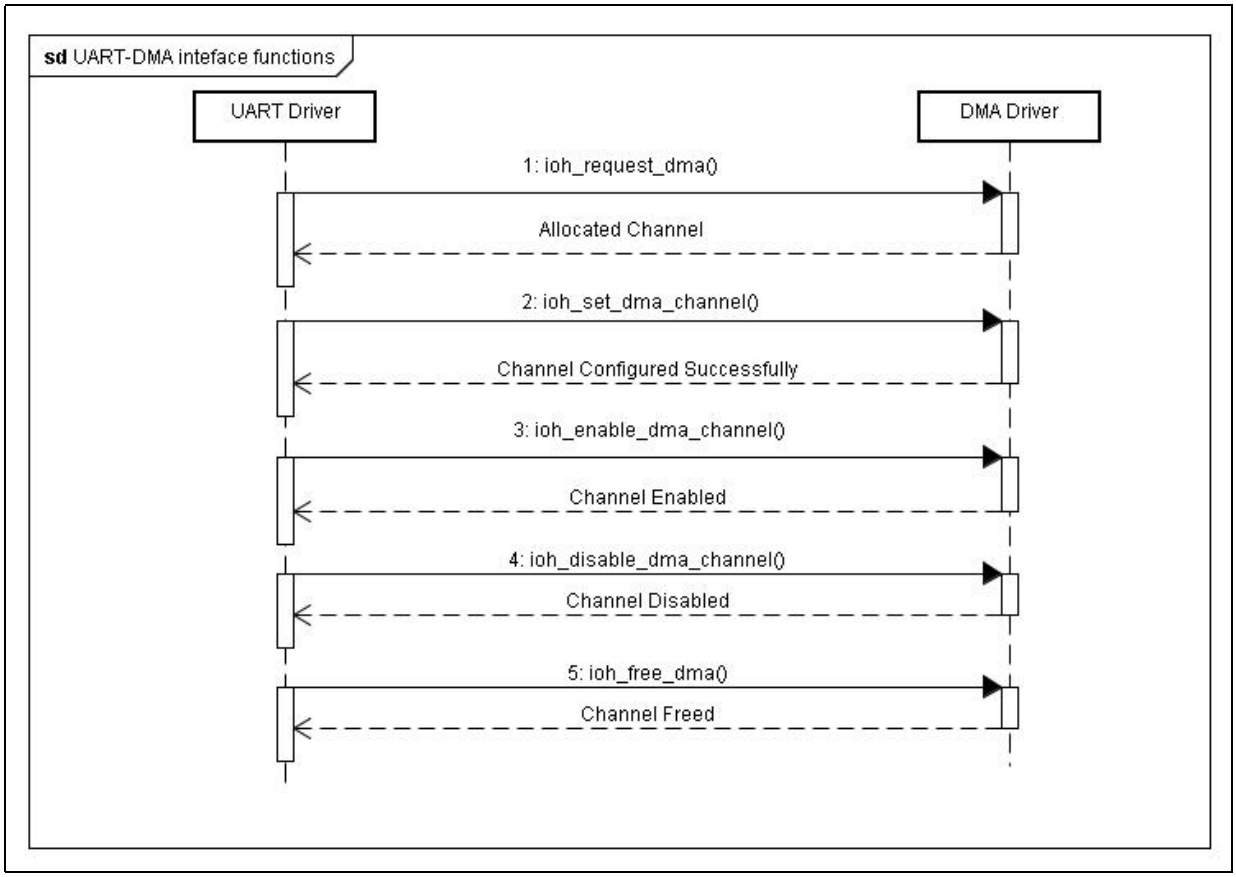
Note: No interfaces are exposed for applications in User Mode.

Table 3. Functions Exported by the Intel® Platform Controller Hub EG20T DMA Driver

No	Function Name	Function Description
1	ioh_request_dma	Requests (allocates) the DMA channel to perform DMA operation.
2	ioh_free_dma	Frees the DMA channel allocated by calling the ioh_request_dma.
3	ioh_set_dma_channel	Configures the DMA channel for the DMA operation.
4	ioh_enable_dma_channel	Enables the DMA transfer for the specified channel.
5	ioh_disable_dma_channel	Disables the DMA transfer for the specified channel.
6	ioh_direct_start_dma	Forces full start of the DMA operation. This function is only used for testing purposes.
7	ioh_add_dma_desc	Adds DMA descriptor information to the existing list of DMA descriptors. This function is only used in scatter gather mode of DMA operation.

To perform DMA transfer using the exported functions, the functions must be called in a particular sequence. The steps involved in calling those functions are explained in Figure 1.

Figure 1. Exported Function Sequence



Refer to the *Intel® Platform Controller Hub EG20T Direct Memory Access (DMA) Controller Driver for Windows Programmer's Guide* for detailed information on programming the DMA driver.