White Paper
Ho Nee Shen
Senior Software Engineer
Intel Corporation
Chan Swee Tat
System Engineer
Intel Corporation

Internal LVDS Dynamic Backlight Brightness Control

A platform and software design using PWM

October, 2010

324567-001US

Executive Summary

The Intel® Atom™ E6xx processor has the ability for dynamically controlling the internal LVDS display backlight brightness using Pulse Width Modulation. This feature opens up many usage possibilities for embedded systems, such as dimming the internal LVDS display in IVI systems when entering a tunnel. Customers could also create an application that allows the user to control the internal LVDS brightness, for example using software buttons on a handheld touch screen device. This paper describes the method to implement dynamic internal LVDS backlight brightness control as a software user application to work with Intel® Embedded Media and Graphics Driver (Intel® EMGD). This paper covers the platform design for Intel Automatic Display Brightness Adjustment. The methods described are operating system- and programming language-independent. This paper does not cover details of the API for programming, however, it describes how to use the Legacy Backlight Brightness PCI Configuration Register to dynamically control the backlight intensity.

Benefits of these methods:

- Removes the need for mechanical potentiometers to control internal LVDS Backlight Brightness resulting in a much cleaner mechanical design.
- Allows developers to integrate internal LVDS backlight control into their embedded system software.
- Intel Automatic Display Brightness Adjustment gives hardware level control over the backlight brightness.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. www.intel.com/embedded/edc.

Contents

Background

The Intel® Atom™ E6xx processor supports internal LVDS display dynamic backlight brightness control using Pulse Width Modulation (PWM). For this document, internal LVDS displays are LCD display panels that use LVDS signals, which are connected to the E6xx's display adapter pin outs. Using the E6xx internal display adapter means that the platform does not need additional Integrated Circuit components to plug in a display panel.

Internal LVDS display backlight brightness is controlled by applying Pulse Width Modulation (PWM) with power to the backlight. PWM switches the backlight on and off very quickly. To dim the LVDS backlight, the duty cycle of the PWM is decreased accordingly. Duty cycle is defined as the proportion of "on" time to a period in the PWM. Toggling the power to the backlight to control its intensity is the de-facto method to control LVDS backlight intensity. The Intel® Atom™ E6xx processor **GPIOSUS [4:0]** pins that support PWM can be used to generate PWM for backlight control.

Throughout this document, the backlight intensity is denoted as a percentage, where, for example, 100% backlight intensity means that the duty cycle is 100% of the PWM period. This setting results is the internal LVDS backlight being at its brightest.

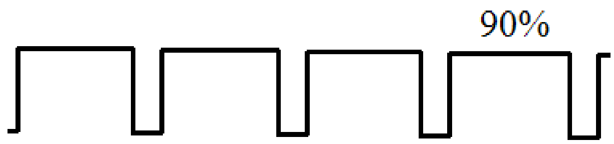
Software controls the backlight intensity by manipulating the internal LVDS display register, a configurable option in Configuration Editor (CED) for Intel® EMGD, and the Legacy Backlight Brightness register in the PCI Configuration Space for the display adapter (to be programmed by the user). The value of Legacy Backlight Control PCI configuration register determines the intensity of the backlight.

There are two types of software required for dynamic backlight brightness control. First is the driver (Intel® EMGD) that is provided by Intel and second is the user application that is created by the customer. This document uses the term "user application" to define the software that the user or customer needs to develop. A User application allows the customer or user to create the interface to control the backlight brightness according to their requirements. This term does not mean that the software is in user space, though it is normally the case but not necessary.

Pulse Width Modulation

With PWM the dimming is achieved by turning the LVDS backlight on and off at a high frequency – so fast the human eye cannot detect the strobe effect. The longer the "on" periods are relative to the "off" periods, the brighter the display will appear to the observer.

Figure 1: Example of PWM Signal



Duty Cycle is a percentage measure of time that the LVDS backlight is physically on. If, for example, the LVDS backlight cycles ON 90% of the time, 10% of the time it is OFF. Therefore the duty cycle is 90% the intensity of the light and will be approximately 90% of its undimmed level.

Typical parameters that need to be taken into consideration are the PWM's dimming (Duty) Ratio, Frequency, High Level Voltage and the Low Level Voltage. Different vendors might have different sets of values and the user will need to refer to respective vendor specs for the actual value.

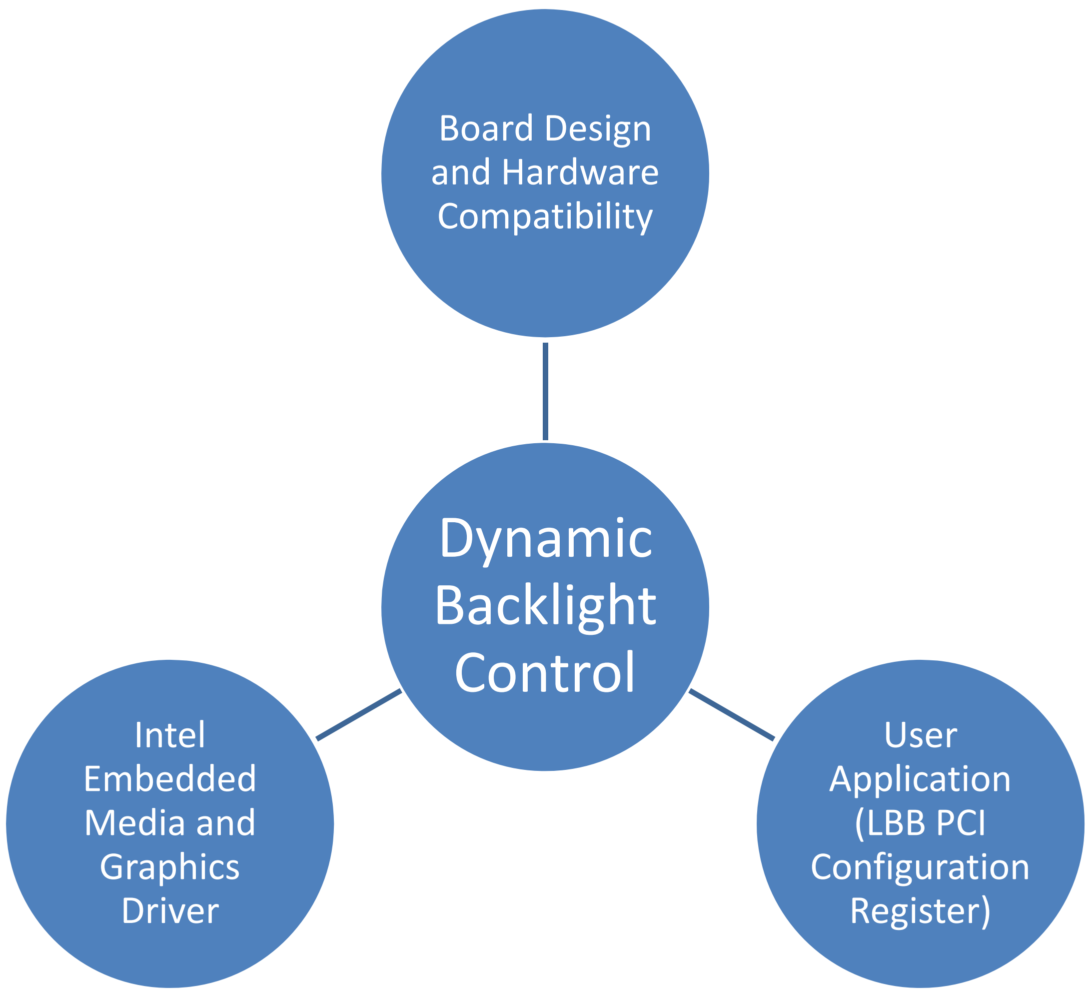
Dependency

Overview

Implementing dynamic backlight control of the internal LVDS requires three key components to build the user application platform:

- Board Design and Hardware Compatibility
- Intel EMGD
- User Application Software Design

Figure 2: Required components for Dynamic backlight control



Board Design and Hardware Compatibility

One of the Intel® Atom™ E6xx processor GPIOSUS [4:0] pins must be routed to the LVDS display's PWM-based backlight inverter module with a point-to-point topology. For the best visual experience, it is recommended that the PWM-based backlight inverter module be capable of an input granularity of 0.1%. That is, the inverter should be able to respond to a 0.1% change in duty cycle. For example, the change from 20% duty cycle to 20.1% duty cycle should be within the capabilities of the PWM-based inverter.

Figure 4 shows the connection between GPIOSUS [4:0] and the corresponding pins on the LVDS panel.

Note: The LVDS backlight inverter comes attached to the LVDS panel or as a component in the LVDS cable assembly. The user must check the specifications of the LVDS panel to make sure that the backlight inverter is compatible with the E6xx processor.

The Intel® Atom™ E6xx processor Customer Reference Boards (CRB) use LVDS panels that have a built-in backlight inverter, although some LVDS panels may not have backlight inverters. In that case, the GPIOSUS [4:0] pins can be connected to the backlight power pin.

This paper is based on the Intel® Atom™ E6xx processor CRB design. Customers may have different hardware designs and they can consult their Intel® representative for further advice.

Figure 3: Intel® Atom E6xx processor to LVDS backlight inverter module connection topology

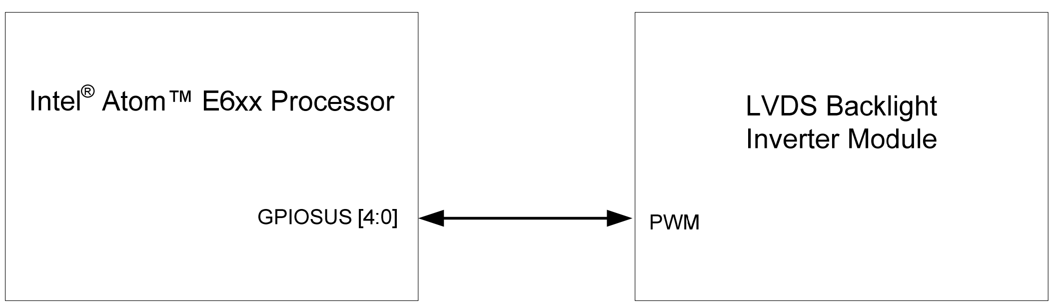
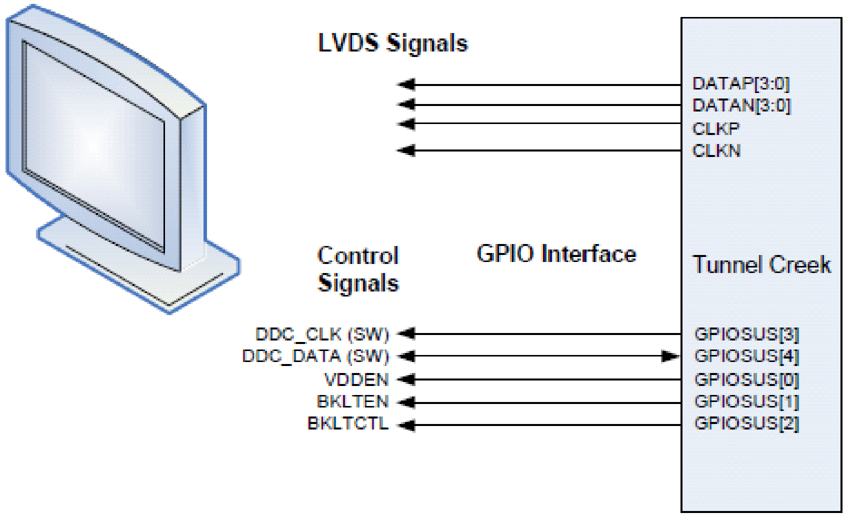


Figure 4: LVDS Control Signal Solution



DDC Pin Emulation

DDC_CLK and DDC_DATA are mainly used to read the Extended Display Identification Data (EDID) of the LVDS panel. The signals on these pins are emulated by software. Intel® EMGD handles the signaling of these pins for the Intel® Atom™ E6xx processor.

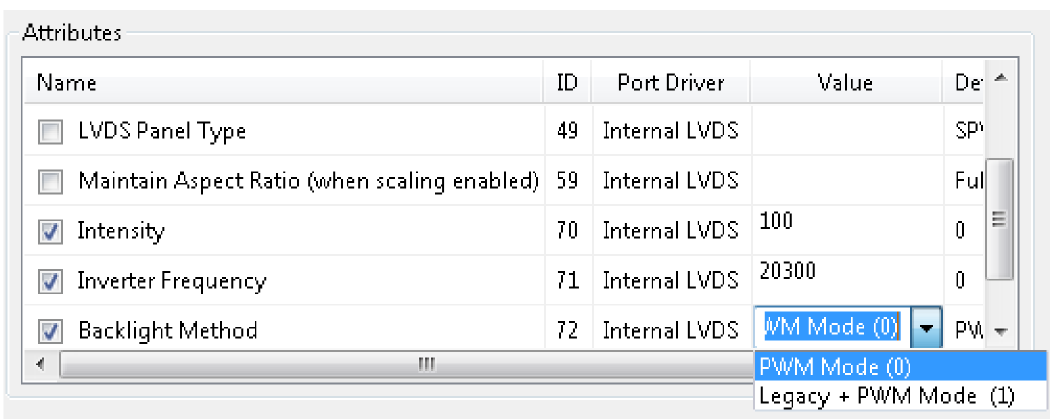
Intel® Embedded Media and Graphics Driver

Intel® EMGD programs the display adapter to enable backlight brightness control. By default, the driver programs the backlight intensity to 100%.

Intel® EMGD Configuration Editor (CED) provides three attributes that need to be programmed to start the display. The Intel® EMGD User's Guide, Port Driver Attributes appendix defines all the attributes and describes how to use CED to configure them. These attributes are:

- Intensity – maximum and default brightness expressed in percentage (attribute ID 70)
- Inverter Frequency – PWM frequency in Hz (attribute ID 71)
- Backlight Method – PWM Mode or Legacy + PWM Mode (attribute ID 72)

Figure 5: Intel® EMGD CED Internal LVDS Attribute settings page with PWM settings



Note that the "Intensity" attribute sets the maximum intensity of the backlight as well as the intensity of the backlight during bootup. The backlight duty cycle will be calculated based on this attribute. With this value set, the user cannot increase the intensity of the backlight more than this value. For example, if the user sets the "intensity" attribute to 90%, the user application will not be able to set more than 90% intensity. The user application sets the backlight intensity as a percentage of the "intensity" attribute.

The "inverter frequency" attribute is the desired frequency for the backlight control. This is normally determined by the inverter in the LVDS panel or the panel tolerance. This value is also known as the Backlight Modulation Frequency. This value may differ depending on the panel being used.

Note: Using the example of the Intel® Atom™ E6xx CRB and panel, we set this value to 20300.

For the LBB (Legacy Backlight Brightness) register to affect the backlight brightness, Intel EMGD needs to be configured with "Backlight Method = 1" (as shown in Figure 5). The legacy backlight method tells the display adapter to factor in the values from LBB register when generating the PWM signal for the internal LVDS backlight. Without setting this attribute, the display adapter will use the "Intensity" attribute value for the backlight and the user has no control over the backlight intensity.

If the"Backlight Method" is not set to '1', the LBB PCI Configuration register value would not take effect. This means that user could not change the backlight brightness from the LBB register. The display will be set to the backlight brightness intensity configured in the "Intensity" attribute in CED (as shown in Figure 4).

EMGD VBIOS

The Intel EMGD VBIOS can be configured with the PWM parameters and then merged into the system BIOS' OROM. During boot-up, the Video BIOS will program the backlight control registers in the display adapter (default to 100%) so that the backlight will be lit and the boot screen can be seen on the internal LVDS Panel.

The driver configuration mentioned in this section also applies to the Intel EMGD Video BIOS.

User Application Software Design

Overview

This section describes high level software designed to control the LVDS brightness intensity. The user needs to make sure that the dependencies for backlight control are met and that Intel® EMGD is installed with the proper attributes.

The key to controlling the backlight brightness is being able to read and write into the Legacy Backlight Brightness register.

Legacy Backlight Brightness (LBB) Register

The LBB register is located in the Intel® Atom™ E6xx display adapter's PCI Configuration Space. The table below shows the location of the LBB register in PCI Configuration Space.

Table 1: Location of LBB in PCI Configuration Space

Bus	0
Device	2
Function	0
Offset	0xF4
Bits	0:7

Software (user application) needs to write the value of the intensity that it is required to be set in the LBB register within bits 0 and 7. This means that the display adapter allows for 255 levels of intensity. Setting the LBB register with the value '0x0' causes the backlight to be turned off and '0xFF' causes the backlight to be set to 100% intensity as configured in the Intel® EMGD.

Because this document uses percentages to represent intensity of the backlight, the user application needs to normalize the value between percentage and the value being set in LBB. The normalization is a linear formula as denoted below.

LBB_Value = (Percentage x 255) ÷ 100

Example: Implementation for Microsoft* Windows* XP

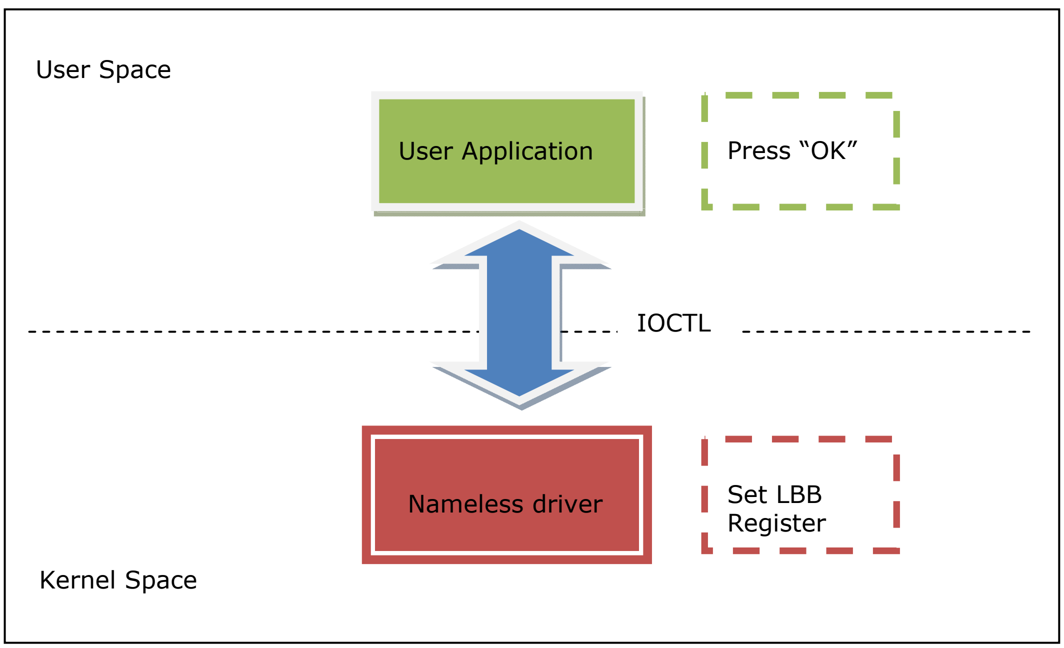
Overview

This section describes a high level software design for implementing a user application on Windows XP to control the backlight brightness. The recommended method to read and write from PCI Config Address space is published by Microsoft at the link below.

http://support.microsoft.com/kb/253232

However, in Windows XP, reading and writing to a PCI Config register can be done only in kernel mode. The API provided by Windows is part of the Windows DDK. As such, the user needs to write a "nameless" driver to access the PCI registers. The "nameless" driver sets up a dummy driver to access the PCI Config register and exposes IOCTLs for the user application to read and write the PCI Config register. The nameless driver is a simple Windows Driver Model (WDM) driver. The figure below shows the high level design.

Figure 6: High-level design for user application interaction with nameless driver



Note: The nameless driver should be implemented as a WDM Function Driver.

WDM driver samples (and source codes) are available in the Windows DDK installation under the "\src" folder.

Nameless Driver

Implementing the nameless driver requires five callback functions to be initialized during *DriverEntry* function. These are:

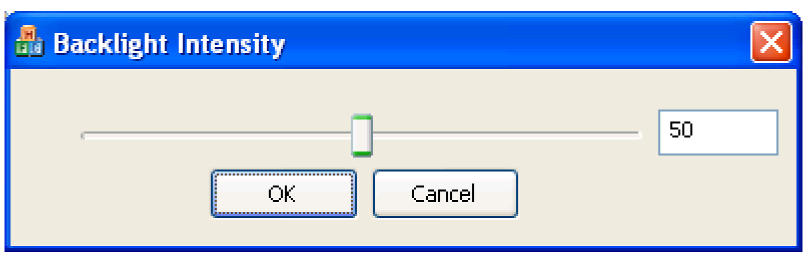
```
pDrvObj->DriverUnload                      = drv_unload;
pDrvObj->MajorFunction[IRP_MJ_CREATE]      = drv_create;
pDrvObj->MajorFunction[IRP_MJ_CLOSE]       = drv_close;
pDrvObj->MajorFunction[IRP_MJ_DEVICE_CONTROL]= drv_device_control;
pDrvObj->MajorFunction[IRP_MJ_CLEANUP]     = drv_cleanup;
```

The *drv_device_control* callback function would implement the I/O Control Codes for the nameless driver. The IOCTL will catch the message packet sent from the user application.

User Application

Using the design in the figure above, the user application can be implemented as an Microsoft Foundation Class application. The application could look like the figure below.

Figure 7: Example of a user application to control the backlight brightness



When the user clicks the "OK" button, the application calls the nameless driver IOCTL to set the LBB Register.

Note: The user application uses the Windows API *DeviceIoControl* to send a message through the IOCTL to the driver.

Platform Design

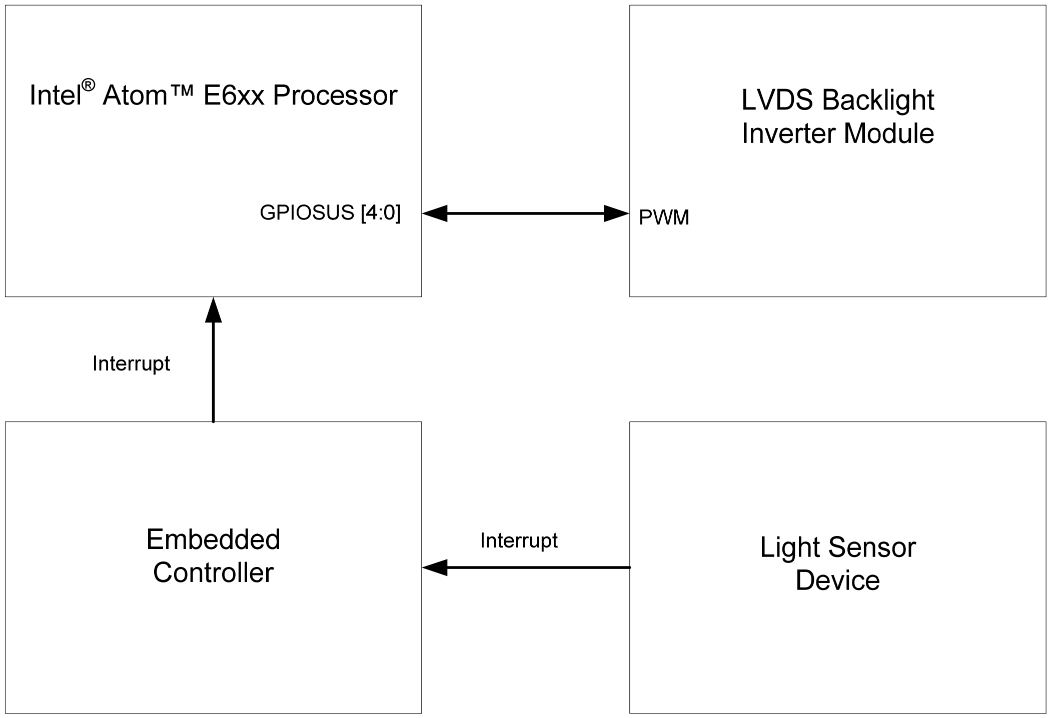
Intel® Automatic Display Brightness Adjustment

Intel® Automatic Display Brightness allows the system to change the backlight brightness in response to changing environmental lighting. This hardware design method does not use the Legacy Backlight Brightness PCI register, therefore no software implementation is required. This method is done exclusively in the hardware. Intel Automatic Display Brightness Adjustment is suitable for low level embedded systems where the backlight brightness does not need user interaction in an In-Vehicle Infotainment (IVI) system, for example, as the vehicle enters a dark tunnel. An ambient light sensor device was added to provide an interrupt to the Intel® Atom™ E6xx series processors to notify it of any changes in the environment brightness.

The suggested method of implementation is to connect the ambient light sensor device to an embedded controller. The embedded controller is then programmed to detect changes in the ambient light and generate an ACPI-visible interrupt (EC_SCI). The Intel® Atom™ E6xx processor responds to the interrupt by changing the PWM duty cycle to dim or brighten the backlight.

Note: The embedded controller can be implemented in a Field-programmable gate array (FPGA).

Figure 8: Automatic Display Brightness Adjustment connection topology



Results

Using the methods described in this paper, the user should be able to create a simple user application that can control the LVDS backlight brightness dynamically. Alternatively, the user can also create a backlight brightness control on a hardware level using the Intel Automatic Display Brightness Adjustment.

Although the example given in this paper is for the software to be developed on a Windows XP environment, the process can be easily extended to other operating system such as Windows CE and Linux. The method for writing the LBB register may be different for various operating systems.

This paper gives the user the ability to control the backlight brightness. This functionality opens up several implementations for embedded systems such as:

- Ability to save power by controlling the backlight.
- “Soft Buttons” on touch screen kiosks to control the display brightness. Useful if the system is used both indoors and outdoors.
- Allow the system to implement a “display fade” effect on the panel when turning off the display.
- Backlight brightness based on ambient light.

Alternative Method

Dynamic backlight brightness control can also be achieved by modifying the Backlight PWM Control Register (offset 0x61250 and 0x61254) in the Display Adapter (Bus 0, Device 2, Function 0), Memory Mapped IO address (MMIO). This method is NOT RECOMMENDED to be used with Intel® EMGD driver as it will conflict with the functionality of the driver, e.g., the driver may overwrite the values programmed by the user. Therefore this method is not covered in this paper.

Conclusion

This paper described at a high level the software implementation for dynamic backlight brightness control for the Internal LVDS panels on the Intel® Atom™ E6xx processor series. An implementation case study in Windows XP was included. However, customers could extend the details in this document to other operating systems such as Linux and Windows CE.

A sample platform design for Intel Automatic Display Brightness Adjustment was also described in this paper.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. http://intel.com/embedded/edc.

Authors

Ho Nee Shen is a Senior Software Engineer with Intel Embedded and Communication Group

Chan Swee Tat is a System Engineer with Intel Embedded and Communication Group

Acronyms

API – Application Programming Interface

BIOS – Basic Input / Output System

CED – Configuration Editor

DDC – Display Data Channel

DDK – Driver Development Kit

EDID – Extended Display Identification Data

EMGD – Embedded Media and Graphics Driver

FPGA – Field-programmable Gate Array

IOCTL – I/O Control Codes

LBB – Legacy Backlight Brightness

LCD – Liquid Crystal Display

LVDS – Low Voltage Differential Signal

MFC – Microsoft Foundation Class

PCI – Peripheral Component Interconnect

PWM – Pulse Width Modulation

WDM – Windows Driver Model

§