By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

Intel, Core, and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

© 2015, Intel Corporation
Contents

1.0 Introduction ............................................................................................................... 5
  1.1 Scope of document ............................................................................................. 5
  1.2 Terminology ...................................................................................................... 5
  1.3 Reference Documents ..................................................................................... 6
  1.4 System Requirements ..................................................................................... 6

2.0 Software Developer Manual ..................................................................................... 7
  2.1 WEC7* and WEC2013* IO Control Code Structure ............................................. 7
      2.1.1 GPIO Driver ........................................................................................ 7
      2.1.2 GPIOPublic.h .................................................................................... 9

Figures

Figure 2-1. Code Example.............................................................................................. 7
Figure 2-2. GPIO Driver............................................................................................... 7
Figure 2-3. GPIOPublic.h ......................................................................................... 9

Tables

Table 1-1. Terminology ............................................................................................... 5
Table 1-2. Reference Documents ............................................................................... 6
Table 2-1. IOCTL_GPIO_READ .............................................................................. 8
Table 2-2. IOCTL_GPIO_WRITE ............................................................................ 8
Table 2-3. IOCTL_GPIO_DIRECTION ...................................................................... 8
Table 2-4. IOCTL_GPIO_MIX ............................................................................... 8
Table 2-5. IOCTL_GPIO_QUERY ............................................................................ 9
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2015</td>
<td>1.0</td>
<td>Initial release. Corresponds to Gold Software Version.</td>
</tr>
</tbody>
</table>

§
1.0 Introduction

1.1 Scope of document

This document provides information for software developers on control codes and header files for GPIO drivers on the Windows* Embedded Compact 7 (WEC7*) and 2013 (WEC2013*).

1.2 Terminology

Table 1-1. Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ATAPI</td>
<td>ATA Packet Interface</td>
</tr>
<tr>
<td>BSP</td>
<td>Board Support Package</td>
</tr>
<tr>
<td>CRB</td>
<td>Customer Reference Board</td>
</tr>
<tr>
<td>DMA</td>
<td>Direct Memory Access</td>
</tr>
<tr>
<td>GPIO</td>
<td>General Purpose Input/Output</td>
</tr>
<tr>
<td>HSIUAR</td>
<td>High Speed Universal Asynchronous Receiver/Transmitter</td>
</tr>
<tr>
<td>I²C</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>IO</td>
<td>Input Ouput</td>
</tr>
<tr>
<td>IOCTL</td>
<td>Input Output Control</td>
</tr>
<tr>
<td>KITL</td>
<td>Kernel Independent Transport Layer</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MSDN</td>
<td>Microsoft Developer Network</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
</tbody>
</table>
### Term Description

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCI</td>
<td>Peripheral Component Interconnect</td>
</tr>
<tr>
<td>SATA</td>
<td>Serial ATA</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
</tbody>
</table>

1.3 **Reference Documents**

Table 1-2. Reference Documents

<table>
<thead>
<tr>
<th>Document</th>
<th>Document No./Location</th>
</tr>
</thead>
</table>

1.4 **System Requirements**

The following are required to build Intel Windows® Embedded Compact 7 (WEC7)/2013 (WEC2013*) IO BSP on Walnut Canyon platform.

1. For WEC7: Install WEC7 Platform Builder with August 2014 QFE Update (7.1.2843). This creates a WINCE700 base directory on the default hard drive (for example, the following path should exist on C: drive: “C:\WINCE700”. If a WINCE700 base directory does not exist, the installation will fail.
2. For WEC2013*: Install Microsoft Windows® Embedded Compact 2013 Platform Builder with update August 2014 (WEC2013- 8.0.6204.0). This creates a WINCE800 base directory on the default hard drive (for example, the following path should exist on C: drive: “C:\WINCE800”. If a WINCE800 base directory does not exist, the installation will fail.
3. Haswell Walnut Canyon fabB Rev 02 platform and C0 (i5) processor
4. Intel BIOS Version: Walnut Canyon 2 BIOS 94.1
5. Intel WEC IO Board Support Package (BSP) version: Intel® Processor WEC IO BSP.msi

§


2.0  Software Developer Manual

2.1  WEC7* and WEC2013* IO Control Code Structure

This section describes the control code and data structures that are exposed to end user on GPIO drivers.

Intel developed a set of IOCTL control code and data structures. End user or OEM code will call the Microsoft WEC7*/2013* Framework API with the Intel developed IOCTL and data structure as parameters.

For example, Intel developed test application will call the WEC7*/2013* Framework API function DeviceIoControl() and pass in Intel created IOCTL code and data structure as a parameter.

Note: Refer to MSDN website for the detail of DeviceIoControl() API function:

Figure 2-1. Code Example

```c
result = (BOOL)DeviceIoControl(hI2CCtrl,
    (DWORD)IOCTL_I2C_EXECUTE_WRITE,
    (LPVOID)pTransBuf,
    sizeof(pTransBuf),
    NULL,
    NULL,
    &BytesHandled,
    NULL);
```

2.1.1  GPIO Driver

Figure 2-2. GPIO Driver

```c
typedef struct
{
    ULONG pin;
    union
    {
        ULONG data;
        GPIO_CONNECT_IO_PINS_MODE ConnectMode;
    } u;
} GPIO_PIN_PARAMETERS, *PGPIO_PIN_PARAMETERS;
```
## 2.1.1.1 IOCTL for GPIO

### Table 2-1. IOCTL_GPIO_READ

<table>
<thead>
<tr>
<th>Description</th>
<th>This function allows the user to read an input pin.</th>
</tr>
</thead>
</table>
| Defined Macro | IOCTL_GPIO_READ \  
| | CTL_CODE( FILE_DEVICE_UNKNOWN, 0x900, METHOD_BUFFERED, FILE_ANY_ACCESS ) |
| Returns | None |

### Table 2-2. IOCTL_GPIO_WRITE

<table>
<thead>
<tr>
<th>Description</th>
<th>This function allows the user to write high or low an output selected pin.</th>
</tr>
</thead>
</table>
| Defined Macro | IOCTL_GPIO_WRITE \  
| | CTL_CODE( FILE_DEVICE_UNKNOWN, 0x901, METHOD_BUFFERED, FILE_ANY_ACCESS ) |
| Returns | None |

### Table 2-3. IOCTL_GPIO_DIRECTION

<table>
<thead>
<tr>
<th>Description</th>
<th>This function allows the user to set as input or output pin.</th>
</tr>
</thead>
</table>
| Defined Macro | IOCTL_GPIO_DIRECTION \  
| | CTL_CODE( FILE_DEVICE_UNKNOWN, 0x902, METHOD_BUFFERED, FILE_ANY_ACCESS ) |
| Returns | None |

### Table 2-4. IOCTL_GPIO_MIX

<table>
<thead>
<tr>
<th>Description</th>
<th>This function allows the user to configure multiplexing for selected GPIO pins.</th>
</tr>
</thead>
</table>
| Defined Macro | IOCTL_GPIO_MIX \  
| | CTL_CODE( FILE_DEVICE_UNKNOWN, 0x903, METHOD_BUFFERED, FILE_ANY_ACCESS ) |
| Returns | None |
Table 2-5. IOCTL_GPIO_QUERY

<table>
<thead>
<tr>
<th>Description</th>
<th>This function allows the user to query current pin's setting.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defined Macro</td>
<td>IOCTL_GPIO_QUERY \</td>
</tr>
<tr>
<td></td>
<td>CTL_CODE( FILE_DEVICE_UNKNOWN, 0x904, METHOD_BUFFERED, FILE_ANY_ACCESS )</td>
</tr>
<tr>
<td>Returns</td>
<td>None</td>
</tr>
</tbody>
</table>

2.1.2 GPIOPublic.h

Figure 2-3. GPIOPublic.h

```c
#ifndef GPIOPUBLIC_H
#define GPIOPUBLIC_H

// Windows Header Files:
#include <windows.h>
#include <winioctl.h>
// C RunTime Header Files
#include <stdlib.h>

//**********************************************************************
*****
// IOCTL code definition
//**********************************************************************
*****
// The IOCTL function codes from 0x800 to 0xFFF are for customer use.
//
#define IOCTL_GPIO_READ \
    CTL_CODE( FILE_DEVICE_UNKNOWN, 0x900, METHOD_BUFFERED, FILE_ANY_ACCESS )

#define IOCTL_GPIO_WRITE \
    CTL_CODE( FILE_DEVICE_UNKNOWN, 0x901, METHOD_BUFFERED, FILE_ANY_ACCESS )

#define IOCTL_GPIO_DIRECTION \
    CTL_CODE( FILE_DEVICE_UNKNOWN, 0x902, METHOD_BUFFERED, FILE_ANY_ACCESS )

#define IOCTL_GPIO_MUX \
    CTL_CODE( FILE_DEVICE_UNKNOWN, 0x903, METHOD_BUFFERED, FILE_ANY_ACCESS )

#define IOCTL_GPIO_QUERY \
    CTL_CODE( FILE_DEVICE_UNKNOWN, 0x904, METHOD_BUFFERED, FILE_ANY_ACCESS )
```

GPIOPublic.h 2.1.2
typedef enum
{
    CONNECT_MODE_INVALID = 0,
    CONNECT_MODE_INPUT,
    CONNECT_MODE_OUTPUT,
    CONNECT_MODE_MAXIMUM = CONNECT_MODE_OUTPUT
} GPIO_CONNECT_IO_PINS_MODE;

typedef struct
{
    ULONG pin;
    union
    {
        ULONG data;
        GPIO_CONNECT_IO_PINS_MODE ConnectMode;
    } u;
} GPIO_PIN_PARAMETERS, *PGPIO_PIN_PARAMETERS;

#endif /* GPIOPUBLIC_H */