

Winmate® API Specifications

**Project Name: PPC Digital I/O
SDK (IBDR)**

Revision: 2.4

May 29, 2015

Contents

1.	General Description	3
1.1	Introduction	3
1.2	Block Diagram.....	3
1.3	I/O Pin Define.....	3
2.	Driver.....	7
2.1	Install WMDIO Driver	7
3.	Programming environment.....	8
3.1	Project Setting.....	8
3.2	WMDIODLL.h File Reference:.....	8
3.3	GPIO Function Block.....	9
4.	API Definition	10
4.1	Function Procedure:.....	10
4.1.1	Function Block.....	10
4.2	Function Name:	10
4.2.1	WM_GpioOpen(void);	10
4.2.2	WM_GpioClose(void);.....	11
4.2.3	WM_SetDOData(UINT16 uiData);	11
4.2.5	WM_GetDIData(PUINT16 puiData);	12
4.2.6	WM_GetDOData(PUINT16 puiData);.....	12
4.3	Program Flow:	13

Revision History

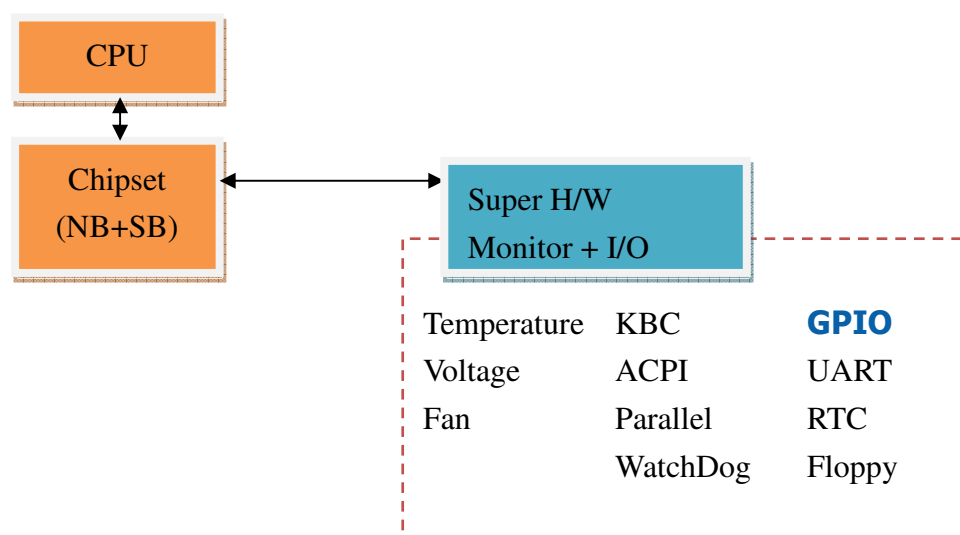
Revision	Date	Description
2.0	2014/03/31	1. Initial draft.
2.1	2014/04/22	1. Add WMDIO Driver 5.0.1.0 for IH70 use only.(for other main board use WMDIO Driver 5.0.0.0)
2.2	2014/08/26	1. Add pin define description for ID31
2.3	2015/04/13	1. Add WMDIO Driver v5.0.2.0 support ID70/IV70 2. Add WMDIO Driver v5.0.3.0 support ID32
2.4	2015/5/29	1. Add WMDIO Driver v5.1.0.0 support IBDR

1. General Description

1.1 Introduction

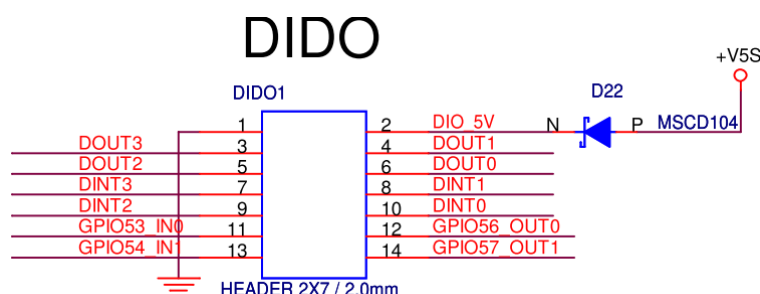
The document describes how to program the Digital I/O. There are 18 programmable digital I/O pins, each pin can be set for output data control or get input status.

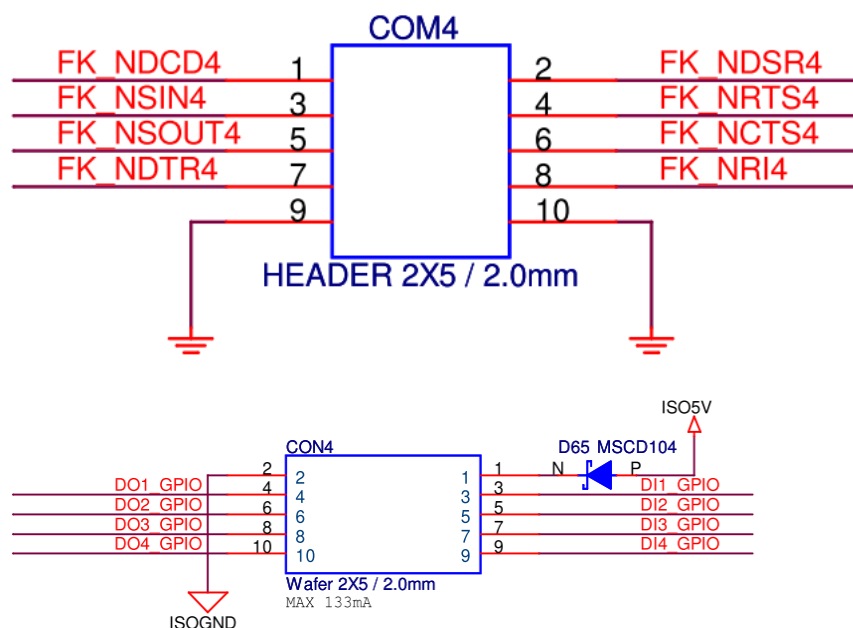
1.2 Block Diagram



1.3 I/O Pin Define

GPIO define :





Pin Definition:

Pin of DIDO	Pin name of DIDO	User define
1	Ground	
2	DIO_5V	
3	DOUT3	DI05
4	DOUT1	DI06
5	DOUT2	DO06
6	DOUT0	DI04
7	DINT3	DO04
8	DINT1	DO05
9	DINT2	DI08
10	DINT0	DI09
11	GPIO53_IN0	DO09
12	GPIO56_OUT0	DI07
13	GPIO54_IN1	DO07
14	GPIO57_OUT1	DO08

Pin of COM4	Pin name of COM4	User define
1	FK_NDCD4	DI03
2	FK_NDSR4	DI02
3	FK_NSIN4	DI01
4	FK_NRTS4	DO03
5	FK_NSOUT4	DO02

6	FK_NCTS4	DO01
---	----------	------

Pin of CON4	Pin name of CON4	User define
1	ISO5V	
2	Ground	
3	DI1_GPIO	
4	DO1_GPIO	
5	DI2_GPIO	
6	DO2_GPIO	
7	DI3_GPIO	
8	DO3_GPIO	
9	DI4_GPIO	
10	DO4_GPIO	

I/O Address:

BIT	Read/Write	Binary Address	Hex Address	PIN
0	Read/Write	0000 0000 0000 0001	0x0001	DI01 \ DO01
1	Read/Write	0000 0000 0000 0010	0x0002	DI02 \ DO02
2	Read/Write	0000 0000 0000 0100	0x0004	DI03 \ DO03
3	Read/Write	0000 0000 0000 1000	0x0008	DI04 \ DO04
4	Read/Write	0000 0000 0001 0000	0x0010	DI05 \ DO05
5	Read/Write	0000 0000 0010 0000	0x0020	DI06 \ DO06
6	Read/Write	0000 0000 0100 0000	0x0040	DI07 \ DO07
7	Read/Write	0000 0000 1000 0000	0x0080	DI08 \ DO08
8	Read/Write	0000 0001 0000 0000	0x0100	DI09 \ DO09
9	Read/Write	0000 0010 0000 0000	0x0200	DI1_GPIO \ DO1_GPIO
10	Read/Write	0000 0100 0000 0000	0x0400	DI2_GPIO \ DO2_GPIO
11	Read/Write	0000 1000 0000 0000	0x0800	DI3_GPIO \ DO3_GPIO
12	Read/Write	0001 0000 0000 0000	0x1000	DI4_GPIO \ DO4_GPIO

Direction:

Bit	Read/Write	Description
11 ~ 0	Read/Write	GPIO I/O Value 0: The respective GPIO PIN is programmed as an Output (Write) port 1: The respective GPIO PIN is programmed as an Input (Read) port

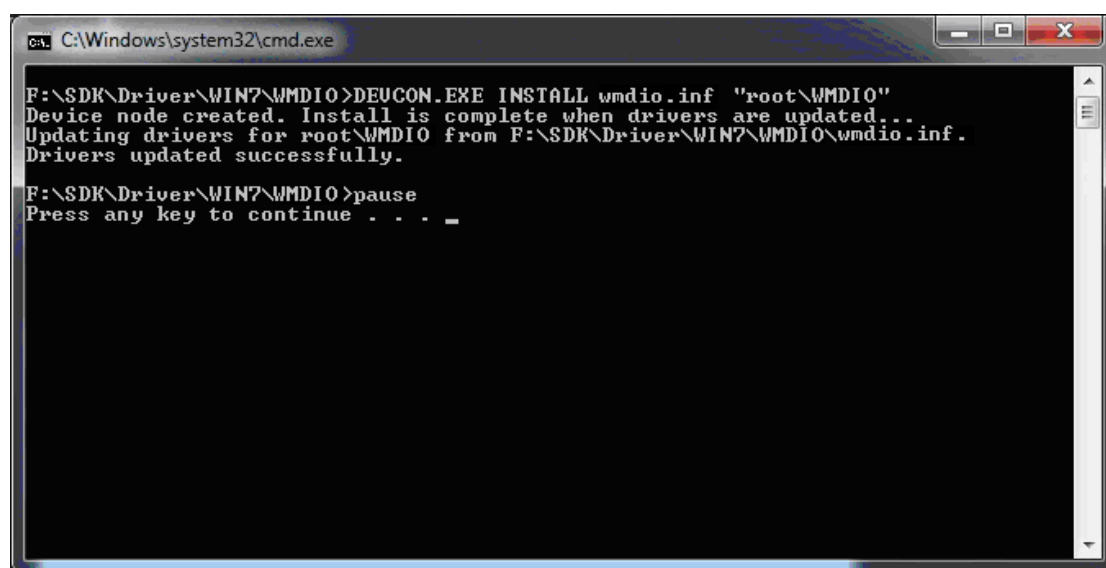
※Bit 8 ~ 11 only for GPIO define 12 pins use.

2.Driver

2.1 Install WMDIO Driver

The drivers are for Windows 7/8. Please double click install.bat in driver directories to start install.

When the driver is successfully installed, you can see "**Drivers Updated successfully**" message in the dos prompt.



```
C:\Windows\system32\cmd.exe
F:\SDK\Driver\WIN7\WMDIO>DEUCON.EXE INSTALL wmdio.inf "root\WMDIO"
Device node created. Install is complete when drivers are updated...
Updating drivers for root\WMDIO from F:\SDK\Driver\WIN7\WMDIO\wmdio.inf.
Drivers updated successfully.
F:\SDK\Driver\WIN7\WMDIO>pause
Press any key to continue . . . _
```

And "**WMDIO**" devices are also added in the **Device Manger** under "**System devices**"

Warning: It is important that only one **WMDIO** devices can appear in Device Manager.

3. Programming environment

3.1 Project Setting

To control the device power state, brightness, led flash control and obtain SMBIOS information, dynamic file (dll), library (lib) and header (h) files are provided to develop the Application.

WMDIODLL.lib	Library
WMDIODLL.dll	Dynamic Library
WMDIODLL.h	Header File

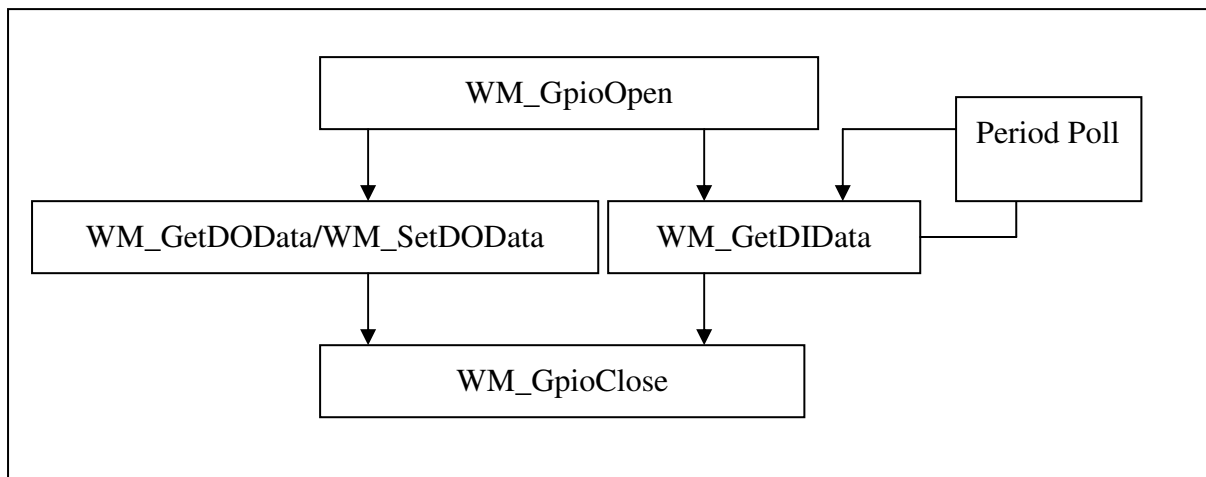
1. Include “WMDIODLL.h” in the project.
2. Add “WMDIODLL.lib” into project Link.
3. Put “WMDIODLL.dll” in the same path with application or into “windows “directory.
4. Check “WMDIODLL.dll” version is 2.0.0.0

3.2 WMDIODLL.h File Reference:

```
#ifdef WMDIODLL_EXPORTS
#define WMDIO_API extern "C"  __declspec(dllexport)
#else
#define WMDIO_API extern "C"  __declspec(dllimport)
#endif
```

```
WMDIO_API int WM_GpioOpen(void);
WMDIO_API int WM_GpioClose(void);
WMDIO_API int WM_SetDODData(UINT16 uiData);
WMDIO_API int WM_GetDIDData(PUINT16 puiData);
WMDIO_API int WM_GetDODData(PUINT16 puiData);
```

3.3 GPIO Function Block



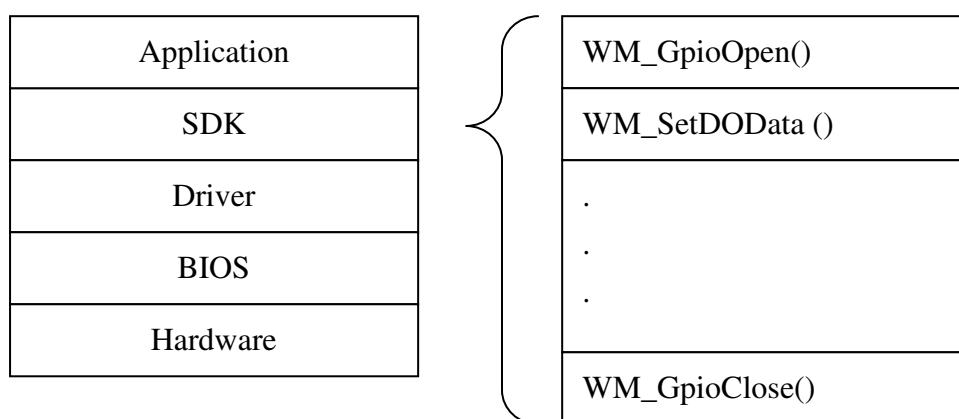
GPIO does not support interrupt.

To get the status of GPIO, polling is necessary.

4.API Definition

4.1 Function Procedure:

4.1.1 Function Block



4.2 Function Name:

4.2.1 WM_GpioOpen(void);

The WM_GpioOpen() Function open the device.

```
WMDIO_API int WM_GpioOpen(void);
```

Parameters

None

Return Value

The function returns TRUE if it is successful open the device, and FALSE otherwise.

Requirements

Header: Declared in WMDIODLL.h

4.2.2 WM_GpioClose(void);

The WM_GpioClose() Function Close the Device.

```
WMDIO_API int WM_GpioClose(void);
```

Parameters

None

Return Value

The function returns TRUE if it is successful close the device, and FALSE otherwise.

Requirements

Header: Declared in WMDIODLL.h

4.2.3 WM_SetDOData(UINT16 uiData);

The WM_SetDOData() Function set the direction of each GPIO pins.

```
WMDIO_API int WM_SetDOData(UINT16 uiData);
```

Parameters

UINT16 uiData

Bit 0 – 11:

0: The respective GPIO PIN is programmed as Low (0V).

1: The respective GPIO PIN is programmed as High (5V)

Example: 0x0014

Set DO03(Bit 2), DO05(Bit 4) to High, others are Low.

Pin	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit	12	11	10	9	8	7	6	5	4	3	2	1	0
H/L	L	L	L	L	L	L	L	L	H	L	H	L	L
uiData	0	0				1				4			

```
WM_SetDOData(0x0014);
```

Return Value

The function returns TRUE if it is successful set the direction of device

Requirements

Header: Declared in WMDIODLL.h

4.2.5 WM_GetDIData(PUINT16 puiData);

The WM_GetDIData() Function get the High/Low of each GPIO pins.

```
WMDIO_API int WM_GetDIData(PUINT16 puiData);
```

Parameters

OUT PUINT16 puiData:

Result :

True : It is successful get the data.

False: otherwise.

Return Value

The function returns the data of each GPIO Pins.

0: The respective GPIO PIN is programmed as Low (0V).

1: The respective GPIO PIN is programmed as High (5V)

Example: 0xFFFC

Set DI01(Bit 0), DI02(Bit 1) to Low, others are High.

Pin	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit	12	11	10	9	8	7	6	5	4	3	2	1	0
H/L	H	H	H	H	H	H	H	H	H	H	H	L	L
puiData	F	F				F				C			

Requirements

Header: Declared in WMDIODLL.h

4.2.6 WM_GetDOData(PUINT16 puiData);

The WM_GetDOData() Function get the High/Low of each GPIO pins.

```
WMDIO_API int WM_GetDOData(PUINT16 puiData);
```

Parameters

OUT PUINT16 puiData:

Result :

True : It is successful get the data.

False: otherwise.

Return Value

The function returns the data of each GPIO Pins.

0: The respective GPIO PIN is programmed as Low (0V).

1: The respective GPIO PIN is programmed as High (5V)

Example: 0xFFCF

Set DO05(Bit 4), DO06(Bit 5) to Low, others are High.

Pin	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit	12	11	10	9	8	7	6	5	4	3	2	1	0
H/L	H	H	H	H	H	H	H	L	L	H	H	H	H
puiData	F	F				C				F			

Requirements

Header: Declared in WMDIODLL.h

4.3 Program Flow:

1. Open : WM_GpioOpen()
2. DO Data : WM_SetDOData () / WM_GetDOData()
3. DI Data : WM_GetDIData ()
4. Close: WM_GpioClose()