

I-8026W Module User Manual

V 2.0.1 Jan. 2025



Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright © 2021 by ICP DAS Co., Ltd. All rights are reserved.

Trademarks

Names are used for identification purposes only and may be registered trademarks of their respective companies.

Contact Us

If you have any problems, please feel free to contact us.
You can count on us for a quick response.
Email: service@icpdas.com

Table of Contents

Table of Contents	3
1. Introduction	4
1.1. Specifications	5
1.2. Pin Assignments	7
1.3. Wire Connections.....	8
1.4. Block Diagram	9
2. Getting Started	10
2.1. MiniOS7-based Controllers	11
2.2. Windows-based Controllers.....	13
2.3. Linux-based Controllers.....	14
3. SDKs and Demo Programs.....	15
4. API References.....	17
4.1. i8026W_Init / pac_i8026W_Init.....	19
4.2. i8026W_GetFirmwareVer / pac_i8026W_GetFirmwareVer	21
4.3. i8026W_GetLibVersion / pac_i8026W_GetLibVersion	23
4.4. i8026W_GetLibDate / pac_i8026W_GetLibDate	25
4.5. i8026W_ReadAIGainOffset / pac_i8026W_ReadAIGainOffset	26
4.6. i8026W_ReadAI / pac_i8026W_ReadAI.....	29
4.7. i8026W_ReadAIHex / pac_i8026W_ReadAIHex	31
4.8. i8026W_Set_AVGCnt / pac_i8026W_Set_AVGCnt	33
4.9. i8026W_ReadAO_GainOffset / pac_i8026W_ReadAO_GainOffset.....	35
4.10. i8026W_WriteAO / pac_i8026W_WriteAO.....	38
4.11. i8026W_ReadbackAO / pac_i8026W_ReadbackAO	40
4.12. i8026W_WriteAOHex / pac_i8026W_WriteAOHex	42
4.13. i8026W_ReadbackAOHex / pac_i8026W_ReadbackAOHex	44
4.14. i8026W_ReadDIO / pac_i8026W_ReadDIO	46
4.15. i8026W_WriteDO / pac_i8026W_WriteDO	49
4.16. i8026W_WriteDOBit / pac_i8026W_WriteDOBit	51
Appendix A. Error Code.....	53
Appendix B. Revision History	54

1. Introduction

I-8026W is a multifunction module that provides 6 analog input channels, 2 analog output channels, 2 digital input channels and 2 digital output channels. It also allows a programmable input range on all analog Input channels (± 10 V, ± 5 V, ± 2.5 V, ± 1.25 V, and ± 20 mA), while analog output channels are 12 bit at either ± 10 V, ± 5 V, 0 to 10 V, 0 to 5 V, or 0 to 20 mA. Each analog input channel can be configured for an individual range, and a high overvoltage protection of 240 Vrms is also provided. Voltage and current inputs/outputs are jumper selectable.

Supported PACs/ViewPACs

The table below shows platforms and PACs/ViewPACs Supported by I-8026W.

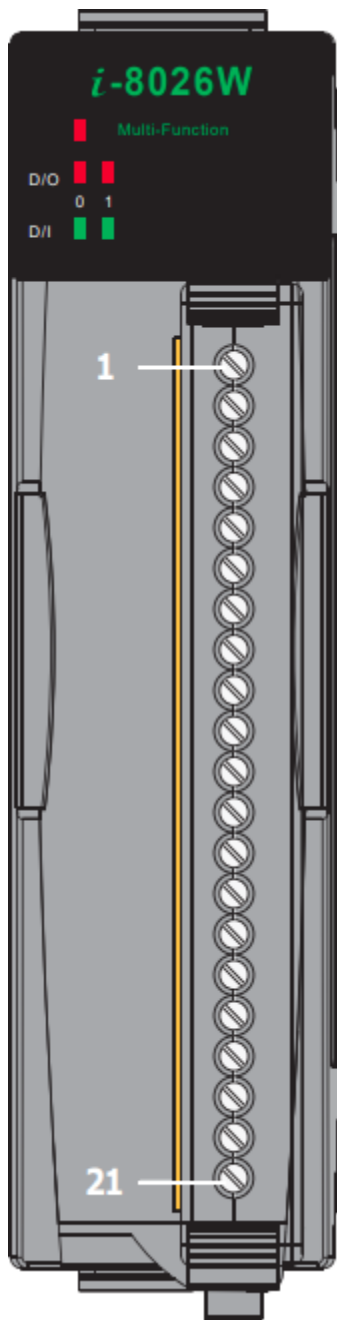
PACs/ViewPACs	Platforms	Models
WinPAC	WinCE	WP-8000-CE7
		WP-8000
XPAC	Windows	XP-8000
		XP-8000-Atom
		XP-8000-WES7
		XP-8000-CE6
		XP-8000-Atom-CE6
LinPAC	Linux	LP-8000 (Linux kernel 3.2/4.4)
		LX-8000 (Linux kernel 3.2/4.4)
IPAC	MiniOS7	iPAC-8000 (MiniOS7)
		I-8000 (MiniOS7)
ViewPAC	WinCE	WinCE ViewPAC with I/O Slot(s)
	MiniOS7	MiniOS7 ViewPAC with I/O Slot(s)

1.1. Specifications

Analog Input	
Channels	6
Range	± 10 VDC, ± 5 VDC, ± 2.5 VDC, ± 1.25 VDC, ± 20 mA (Jumper Select)
Resolution	12-bit
Accuracy	0.2% of FSR ± 1 LSB
Overvoltage Protection	240 Vrms
Input Impedance	>2 M Ω
Sampling Rate	Max. to 9k Samples/Sec.
Analog Output	
Channels	2
Range	± 10 V, ± 5 V, $0 \sim 10$ V, $0 \sim 5$ V, $0 \sim 20$ mA (Jumper Select)
Resolution	12-bit
Accuracy	$\pm 0.2\%$ of FSR
Voltage Output Capability	10 VDC @ 20 mA
Digital Input	
Channels	2
Type	Wet contact
Sink/Source	Sink, Source
On Voltage Level	$+10 \sim 30$ VDC
Off Voltage Level	$+4$ VDC max
Digital Output	
Channels	2
Sink/Source (NPN/PNP)	Sink
Type	Isolated Open Collector
Max. Load Current	100 mA/Channel
Load Voltage	$+5 \sim +30$ VDC

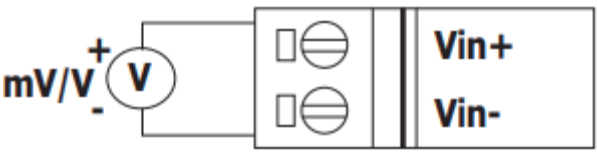
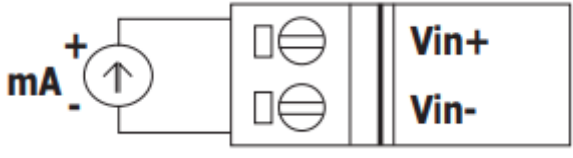
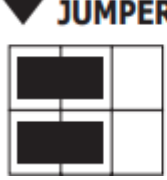
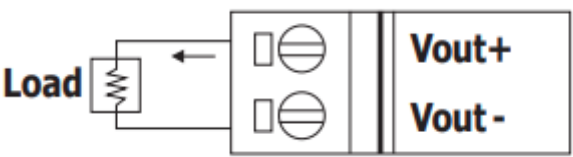
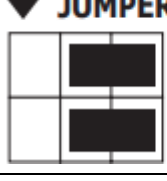
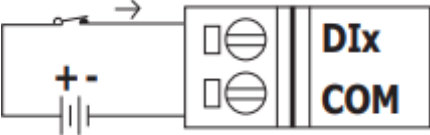
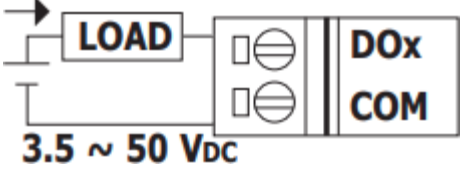
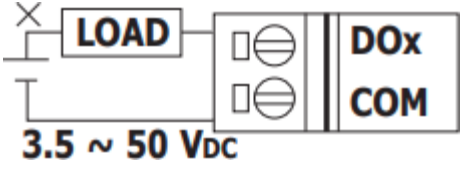
LED Indicators	
System LED Indicator	1 LED as Power/Communication Indicator
I/O LED Indicator	4 LEDs as Digital Input & Digital Output Indicators
Isolation	
Intra-module Isolation, Field to Logic	2500 VDC
EMS Protection	
ESD (IEC 61000-4-2)	±4 kV Contact for Each Terminal
Power	
Power Consumption	1.8 W Max.
Mechanical	
Dimensions (L x W x H)	30 mm x 102 mm x 115 mm
Environment	
Operating Temperature	-25 ~ +75 °C
Storage Temperature	-40 ~ +85 °C
Humidity	10 ~ 90% RH, Non-condensing

1.2. Pin Assignments

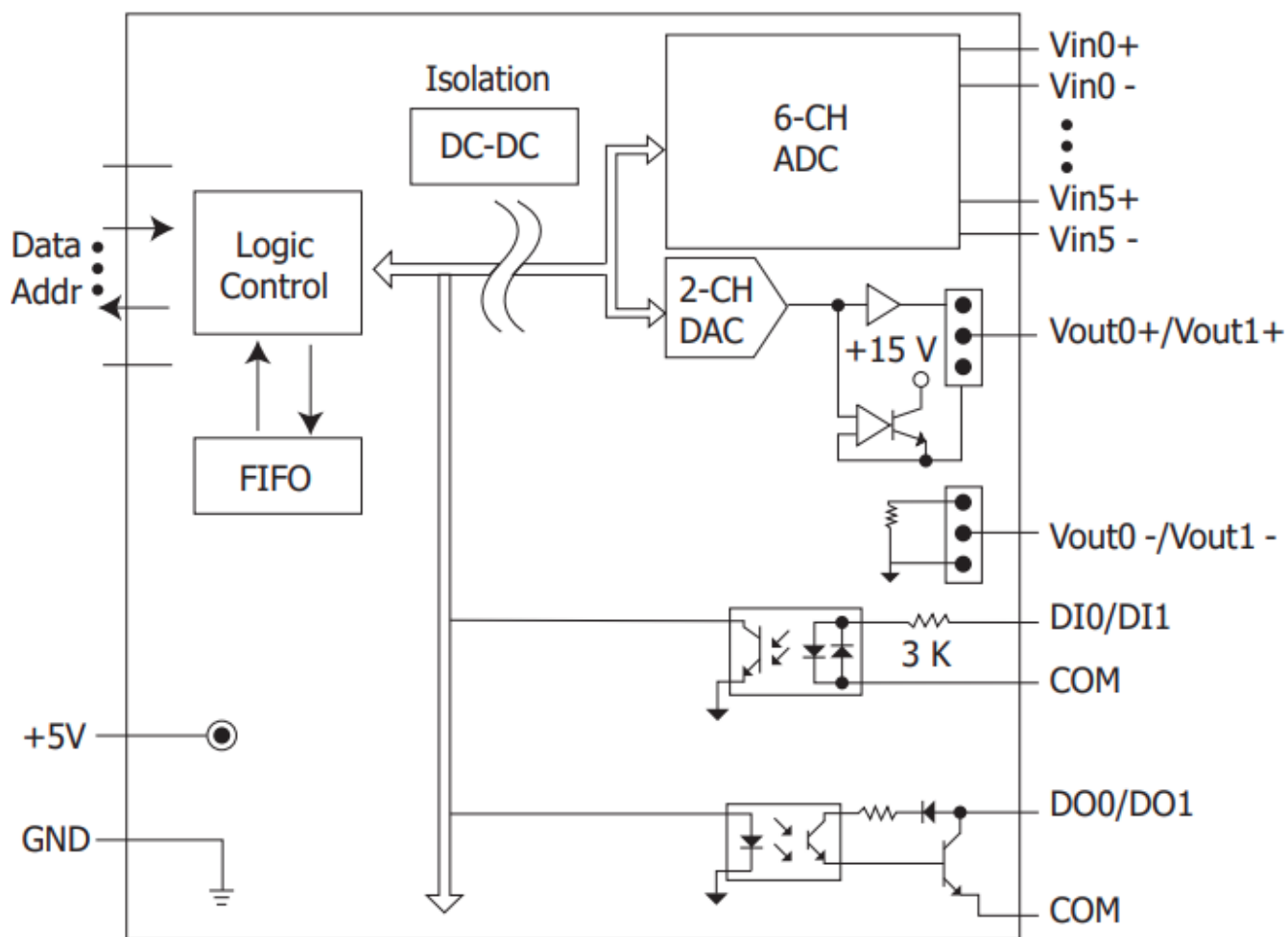


Terminal No.	Pin Assignment
01	Vin0+
02	Vin0 -
03	Vin1+
04	Vin1 -
05	Vin2+
06	Vin2 -
07	Vin3+
08	Vin3 -
09	Vin4+
10	Vin4 -
11	Vin5+
12	Vin5 -
13	Vout0+
14	Vout0 -
15	Vout1+
16	Vout2 -
17	DO0
18	DO1
19	DI0
20	DI1
21	COM

1.3. Wire Connections

AI	Voltage Input	  JUMPER Default
	Current Input	  JUMPER
AO	Voltage Output	  JUMPER Default
	Current Output	  JUMPER
DI	ON State Readback as 1	
	OFF State Readback as 0	
DO	ON State Readback as 1	
	OFF State Readback as 0	

1.4. Block Diagram



2. Getting Started

This chapter will be accompanied by demos to explain how to implement functions.

Demos can be downloaded in the following link:

<https://www.icpdas.com/en/download/show.php?num=2899&model=I-8026W-G>

FILE NAME
8026_WES_x64.zip
8026_WES_x86.zip
8026_UWP_x64.zip
8026_CE7_ARM.zip
8026_CE6_x86.zip
8026_CE5_ARM.zip
8026_MiniOS7.zip

2.1. MiniOS7-based Controllers

Basic function can be used to retrieve configuration information and verify the IO functions. The following steps show how to test I-8026W in MiniOS7 controller.

Step 1. Please refer to the “Wire Connections”,

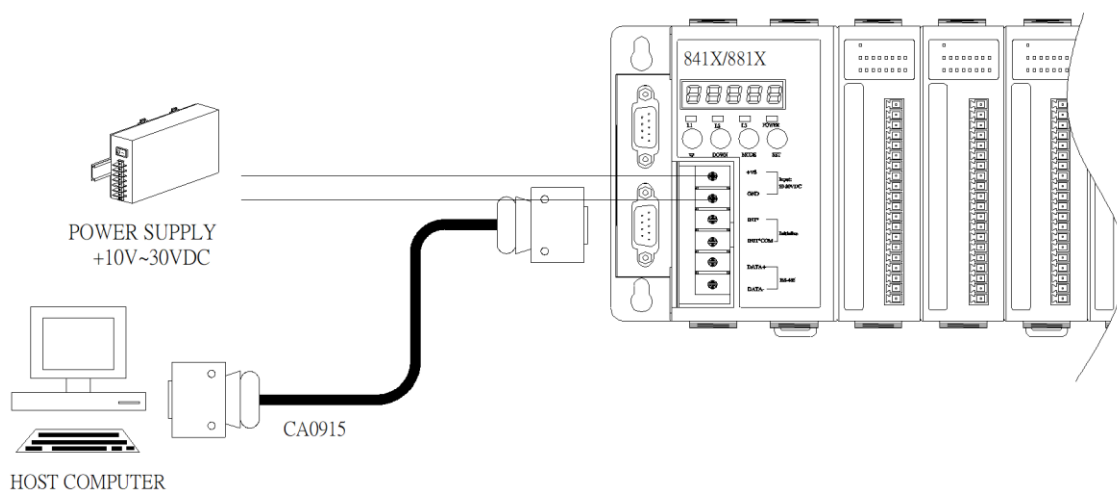
Connect AI channels of I-8026W to a stable source (such as battery);

Connect AO channels of I-8026W to voltage meter or current meter;

Connect DI channels of I-8026W to power supply;

Connect DO channels of I-8026W to voltage meter or LEDs.

Step 2. Plug I-8026W in the MiniOS7 controller, connect the power supply to the unit and connect the unit to the Host PC by RS-232 cable



Step 3. Run 7188xw.exe on the host PC and open the COM Port which is connecting to the MiniOS7 Controller

Tips & Warnings



7188xw.exe is an interface for PC, it can help users to communicate with MiniOS7 PAC, please refer to the MiniOS7 PAC user manual for more detail.

Step 4. Run the demo and then verify the basic functions.

```
C837_FD_UDP>run

This demo show I-8026W firmware and lib information.
There is an i8026 at slot 1

*****Firmware Version =: 0004
Library Version =: 1102
Build Date =: Jun 07 2021
*****
Choice Menu
      0: Test DIO
      1: Test AI
      2: Test AO
      3: to quit the program

1
Gain
  0=+/-10V
  1=+/-5V
  2=+10V
  3=+5V
  4=0~20mA
  Please choose (0~4):0
Select Format
  0=Engineering format
  1=Hex format

0
+/-10V ==> -1.601 -5.001 -5.003 -5.006 -5.003 -4.998
  Press '0' to quit the program
  others to menu
```

2.2. Windows-based Controllers

Basic function can be used to retrieve configuration information and verify the IO functions.

The following steps show how to test I-8026W in CE or WES controller.

Step 1. Please refer to the “Wire Connections”,

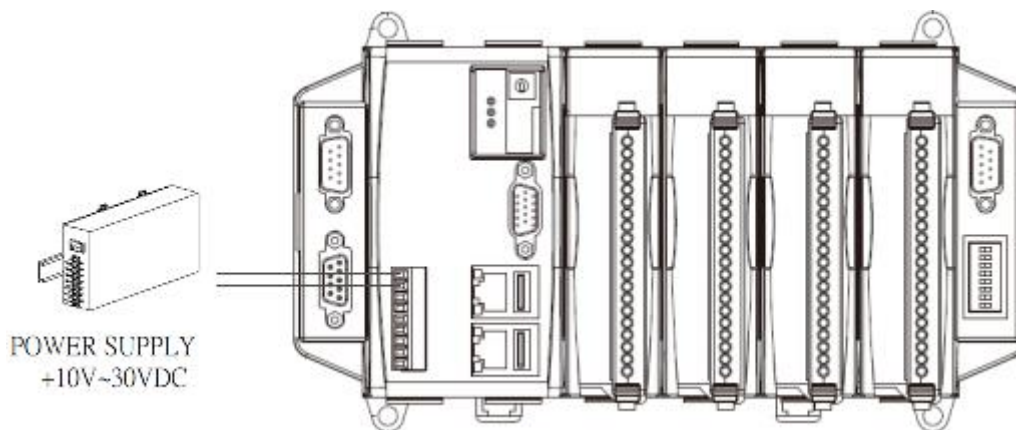
Connect AI channels of I-8026W to a stable source (such as battery);

Connect AO channels of I-8026W to voltage meter or current meter;

Connect DI channels of I-8026W to power supply;

Connect DO channels of I-8026W to voltage meter or LEDs.

Step 2. Plug I-8026W in the Windows-based controller and connect the power supply to the unit.



Step 3. Run the demo and then verify the basic functions.

Form1

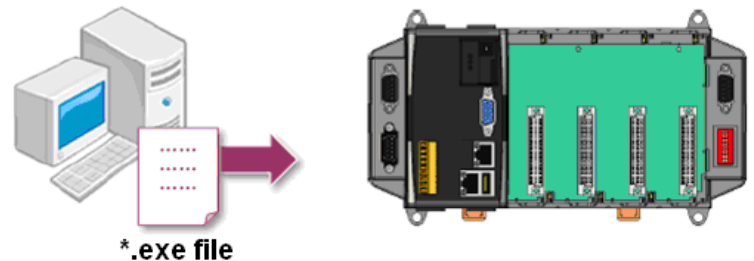
slot Index	Slot 7	Library	1102
Firmware	0004	Library Date	Jun 7 2021
Select AI Gain	[0]: +/-10V	Read AI	
CH:00	[5.065308]	CH:03	[5.064087]
CH:01	[5.059204]	CH:04	[5.061646]
CH:02	[5.061646]	CH:05	[2.637024]
Select AO Gain	[0]: +/-10V		
Write AO CH0	2	Write AO CH1	1
Write DO	0	0~3	
Read DIO	DO Readback	0	
	DI Readback	0	
Exit			

2.3. Linux-based Controllers

Basic function can be used to retrieve configuration information and verify IO output functions.

Basic information includes:

- The version number and the FPGA version information
- The gain and offset values for each input range and each channel
- The data read from each channel



Step 1. Users need to download LinPAC SDK, which includes GNU tool chain, Libraries, header, examples files, etc.

Step 2. Check the power cable, Ethernet cable, VGA monitor, the communication cable between controller and PC has been connected well, and then check the I-8026W has been plugged in the controller.

Step 3. Check the communication between controller and PC is fine, and download the demo program files to the controller.

The following is a list of the locations where both the demo programs and associated libraries (**libi8k.a**) can be found on either the ICP DAS web site, and user can find the related files in the below website:

Models	Links
LP-8x2x	http://www.icpdas.com/en/download/show.php?num=915
LP-8x4x	http://www.icpdas.com/en/download/show.php?num=982
LX-8000	http://www.icpdas.com/en/download/show.php?num=904
LP-9x2x	http://www.icpdas.com/en/download/show.php?num=915
LX-9000	http://www.icpdas.com/en/download/show.php?num=904

3. SDKs and Demo Programs

ICP DAS provides a range of demo programs for different platforms that can be used to verify the functions of the I-8026W. The source code contained in these programs can also be reused in your own custom programs if needed. The following is a list of the locations where both the demo programs and associated libraries can be found on either the ICP DAS web site.

Platform	Location
MiniOS7 I-8000(I-8x11)/iPAC-8000 Platform	
Library	https://www.icpdas.com/en/download/show.php?num=2899
Demo	Demo: I-8026W_MiniOS7
Windows CE5(WP-8x4x/8x3x,VP25Wx) Platform	
Library	https://www.icpdas.com/en/download/show.php?num=2899
Demo	Demo: 8026_CE5_ARM VC Demo: i8026W_Basic_Info_Cplusplus C# Demo: i8026W_Basic_Info_Csharp
Windows CE7(WP-8x2x ,VP-x23x) Platform	
Library	https://www.icpdas.com/en/download/show.php?num=2899
Demo	Demo: 8026_CE7_ARM VC Demo: i8026W_Basic_Info_Cplusplus C# Demo: i8026W_Basic_Info_Csharp
Windows CE6(XP-8000-CE6/XP-8000-Atom-CE6) Platform	
Library	https://www.icpdas.com/en/download/show.php?num=2899
Demo	Demo: 8026_CE6_x86 VC Demo: i8026W_Basic_Info_Cplusplus C# Demo: i8026W_Basic_Info_Csharp

Windows WES/IOT (XP-8000/XP-Atom/IPPC) Platform	
Library	https://www.icpdas.com/en/download/show.php?num=2899
Demo	Demo: 8026_WES_x86/ 8026_WES_x64 VC Demo: i8026W_Basic_Info_Cplusplus C# Demo: i8026W_Basic_Info_Csharp Demo: 8026_UWP_x64 (IoT PAC) C# Demo: i8026W_Basic_Info_Csharp
Linux LP-8x4x Platform	
SDK & 7k/8K/87K I/O module Demo (old) I/O module Demo	https://www.icpdas.com/en/download/show.php?num=982 WinDows PC : linpac_pxa270_sdk_for_windows.exe Linux PC: linpac_pxa270_sdk_for_linux.tar.bz2 https://www.icpdas.com/en/download/show.php?num=992
Linux LP-8x2x/9x2x Platform	
SDK&7k/8K/9K/ 87K/97K I/O module Demo	https://www.icpdas.com/en/download/show.php?num=915 WinDows PC : linpac_am335x_sdk_for_windows.exe Linux PC: linpac_am335x_sdk_for_linux.tar.bz2
Linux LX-8000/9000 Platform	
SDK&7k/8K/9K/8 7K/97K I/O module Demo	http://www.icpdas.com/en/download/show.php?num=904 Linux P : linpac_x86_sdk.tgz

4. API References

ICPDAS supplies a range of C/C++ API functions for I-8026W.

When developing a program, refer to either the header file, or the API functions described in the following sections for more detailed information.

ICPDAS also supplies a range of C# function that can be used to develop .NET programs, these functions are ported from the relevant C/C++ functions.

Download link : <https://www.icpdas.com/en/download/show.php?num=2899&model=I-8026W-G>

API naming table

The following table shows the API names on different platforms and the beginning of API.

Platform	Product included	API prefix characters	
		C / C++	C#
CE5.0	I-8026W	"pac_i8026W_" + function name	"pac8026WNet.pac8026W" + function name
CE6.0	I-8026W		
CE7.0	I-8026W		
WES	I-8026W	"i8026W_" + function name	Null
MiniOS7	I-8026W		
Linux	I-8026W		

The following table lists the functions provided in 8026W.lib and pac_8026W.lib.

API	Description
i8026W_Init pac_i8026W_Init	Used to initialize the module
i8026W_GetFirmwareVer pac_i8026W_GetFirmwareVer	Used to read the firmware (FPGA) version.
i8026W_GetLibVersion pac_i8026W_GetLibVersion	Used to read the version and build information for the currently installed Library
i8026W_GetLibDate pac_i8026W_GetLibDate	Used to read the version and build information for the currently installed Library
i8026W_ReadAIGainOffset pac_i8026W_ReadAIGainOffset	Used to read the calibrated Gain and Offset values.
i8026W_ReadAI pac_i8026W_ReadAI	Used to read the Analog Input value from a specific channel in float format
i8026W_ReadAIHex pac_i8026W_ReadAIHex	Used to read the Analog Input value from a specific channel in 16-bit hexadecimal format
i8026W_Set_AVGCnt pac_i8026W_Set_AVGCnt	Used to set the number of calculation averages.
i8026W_ReadAO_GainOffset pac_i8026W_ReadAO_GainOffset	Used to read the calibrated Gain and Offset values
i8026W_WriteAO pac_i8026W_WriteAO	Used to output the voltage of float format value in the specified channel.
i8026W_ReadbackAO pac_i8026W_ReadbackAO	Used to read back output AO value in float format.
i8026W_WriteAOHex pac_i8026W_WriteAOHex	Used to output the voltage of hexadecimal format value in the specified channel.
i8026W_ReadbackAOHex pac_i8026W_ReadbackAOHex	Used to read back output AO value in hexadecimal format.
i8026W_ReadDIO pac_i8026W_ReadDIO	Used to read DO and Di status.
i8026W_WriteDO pac_i8026W_WriteDO	Used to write DO status. For all channel.
i8026W_WriteDOBit pac_i8026W_WriteDOBit	Used to write DO status in specified channel

4.1. i8026W_Init / pac_i8026W_Init

This function is used to initialize the module, and must be called at least once in the beginning.

Syntax

For MiniOS7

```
short i8026W_Init(int slot);
```

For Windows

```
short pac_i8026W_Init(int slot);
```

For Linux

```
short i8026W_Init(int slot);
```

Parameters

slot:

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C++]

```
int slot;  
i8026W_Init(slot);
```

[C#]

```
int slot;  
pac8026WNet.pac8026W.Init(slot);
```

[C] (For LinPAC)

```
int main(){  
    int slot, ret;  
    ret=Open_Slot(slot);  
    if (ret > 0) {  
        printf("Open Slot%d failed, return value=%d \n", slot, ret);  
        return (-1);  
    }  
  
    i8026W_Init(slot);  
    Close_Slot(slot);  
    return 0;  
}
```

4.2. i8026W_GetFirmwareVer / pac_i8026W_GetFirmwareVer

This function is used to retrieve the version information for the FPGA firmware.

Syntax

For MiniOS7

```
short i8026W_GetFirmwareVer(int slot);
```

For Windows (CE and WES)

```
short pac_i8026W_GetFirmwareVer(int slot);
```

For Linux

```
short i8026W_GetFirmwareVer(int slot);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

Return Values

The firmware version number.

Others: Refer to Appendix A: “Error Code Definitions” for more details.

Examples

[C++]

```
int slot;  
short version;  
version = i8026W_GetFirmwareVer(slot);
```

[C#]

```
Int16 slot,version;  
version = pac8026WNet.pac8026W.FirmwareVersion(slot);
```

[C] (For LinPAC)

```
int main(){  
    int slot, ch,ret;  
    float data;  
    ret=Open_Slot(slot);  
    if (ret > 0) {  
        printf("Open Slot%d failed, return value=%d \n", slot, ret);  
        return (-1);  
    }  
  
    i8026W_Init(slot);  
    i8026W_GetFirmwareVer(slot);  
    Close_Slot(slot);  
    return 0;  
}
```

4.3. i8026W_GetLibVersion / pac_i8026W_GetLibVersion

This function is used to read the version and build information for the Library.

Syntax

For MiniOS7

```
short i8026W_GetLibVersion(void);
```

For Windows (CE and WES)

```
short pac_i8026W_GetLibVersion(void);
```

For Linux

```
short i8026W_GetLibVersion(void);
```

Parameter

None

Return Values

The library version number.

Others: Refer to Appendix A: “Error Code Definitions” for more details.

Examples

[C++]

```
short version;  
version = i8026W_GetLibVersion();
```

[C#]

```
Int16 version;  
version = pac8026WNet.pac8026W.LibVersion();
```

[C] (For LinPAC)

```
int main(){  
    int slot, ch, ret;  
    float data;  
    ret=Open_Slot(slot);  
    if (ret > 0) {  
        printf("Open Slot%d failed, return value=%d \n", slot, ret);  
        return (-1);  
    }  
  
    i8026W_Init(slot);  
    i8026W_GetLibVersion();  
    Close_Slot(slot);  
    return 0;  
}
```

4.4. i8026W_GetLibDate / pac_i8026W_GetLibDate

This function is used to get the build date library.

Syntax

For MiniOS7

```
void i8026W_GetLibDate(char libDate[]);
```

For Windows (CE and WES)

```
void pac_i8026W_GetLibDate(char libDate[]);
```

Parameter

libDate[]

A string indicating the build date of the Library.

Return Values

None

Examples

[C++]

```
char libDate [32];  
i8026W_GetLibDate(libDate);
```

[C#]

```
String libdate;  
libdate =pac8026WNet.pac8026W.LibDate();
```

4.5. i8026W_ReadAIGainOffset / pac_i8026W_ReadAIGainOffset

This function is used to read the calibrated AI gain and offset value.

Syntax

For MiniOS7

```
short i8026W_ReadAIGainOffset(  
    int slot,  
    int ch,  
    int gain,  
    unsigned short *gainValue,  
    short *offsetValue  
);
```

For Windows (CE and WES)

```
short pac_i8026W_ReadAIGainOffset(  
    int slot,  
    int ch,  
    int gain,  
    unsigned short *gainValue,  
    short *offsetValue  
);
```

For Linux

```
short i8026W_ReadAIGainOffset(  
    int slot,  
    int ch,  
    int gain,  
    unsigned short* refGain,  
    short * refOffset  
);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 5).

gain [in]

specifies the input type (0 ~ 4),

0: +/-10 V, 1: +/-5 V, 2: +/-2.5 V,

3: +/-1.25 V, 4: +/-20 mA

****refGain [Out]***

the reference gain value for the Analog Input type.

****refOffset [Out]***

the reference offset value for the Analog Input type.

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot;
short ch;
short gain;
unsigned short GainValue;
short OffsetValue;
i8026W_ReadAIGainOffset(slot, ch, gain,&GainValue, &OffsetValue);
```

[C#]

```
Int slot;
Int16 ch;
short gain;
UInt16 GainValue;
Int16 OffsetValue;
pac8026WNet.pac8026W.ReadAO_GainOffset(slot,ch,gain,ref GainValue,ref OffsetValue);
```

[C] (For LinPAC)

```
int main(){
    unsigned short gVal=0;
    short oVal=0;
    int slot, ch, gain,ret;
    ret=Open_Slot(slot);
    if (ret > 0) {
        printf("Open Slot%d failed, return value=%d \n", slot, ret);
        return (-1);
    }

    i8026W_Init(slot);
    i8026W_ReadAIGainOffset (slot,ch,gain,&gVal, &oVal);
    Close_Slot(slot);
    return 0;
}
```

4.6. i8026W_ReadAI / pac_i8026W_ReadAI

This function is used to read the calibrated floating-point value from a specified Analog Input channel.

Syntax

For MiniOS7

```
short float i8026W_ReadAI(int slot,int ch,short gain,float* fVal);
```

For Windows (CE and WES)

```
short pac_i8026W_ReadAI(int slot,int chIndex,short gain,float* fVal);
```

For Linux

```
short float i8026W_ReadAI(int slot, int ch, short gain, float* fVal);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 5).

gain [in]

specifies the input type (0 ~ 4),

0: +/-10 V, 1: +/-5 V, 2: +/-2.5 V,

3: +/-1.25 V, 4: +/-20 mA

***fVal [Out]**

the input data in float format.

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot, ch, gain;
float fVal=0;
i8026W_ReadAI( slot, ch, gain, &fVal);
```

[C#]

```
int slot ,ch, gain;
float fval;
pac8026WNet.pac8026W.ReadAI(slot, ch, gain ,ref fval);
```

[C] (For LinPAC)

```
int main(){
    int slot, ch, gain ,ret;
    float fval;
    ret=Open_Slot(slot);
    if (ret > 0) {
        printf("Open Slot%d failed, return value=%d \n", slot, ret);
        return (-1);
    }

    i8026W_Init(slot);
    i8026W_ReadAI(slot,ch,gain,&fval);
    Close_Slot(slot);
    return 0;
}
```

4.7. i8026W_ReadAIHex / pac_i8026W_ReadAIHex

This function is used to read the calibrated hexadecimal format value from a specified Analog Input channel.

Syntax

For MiniOS7

```
short i8026W_ReadAIHex(int slot,int chIndex,short gain,short* hVal);
```

For Windows (CE and WES)

```
short pac_i8026W_ReadAIHex(int slot,int chIndex,short Gain,short* hVal);
```

For Linux

```
short i8026W_ReadAIHex(int slot,int ch,short gain,short* hVal);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 5).

gain [in]

specifies the input type (0 ~ 4),

0: +/-10 V, 1: +/-5 V, 2: +/-2.5 V,

3: +/-1.25 V, 4: +/-20 mA

***hVal [Out]**

the input data in hexadecimal format.

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot, ch, gain;
short hVal=0;
i8026W_ReadAIHex( slot, ch, gain, &hVal);
```

[C#]

```
int slot ,ch, gain;
short hval;
pac8026WNet.pac8026W.ReadAIHex(slot, ch, gain ,ref hval);
```

[C] (For LinPAC)

```
int main(){
    int slot, ch, gain ,ret;
    short hval;
    ret=Open_Slot(slot);
    if (ret > 0) {
        printf("Open Slot%d failed, return value=%d \n", slot, ret);
        return (-1);
    }

    i8026W_Init(slot);
    i8026W_ReadAIHex(slot,ch,gain,&hval);
    Close_Slot(slot);
    return 0;
}
```

4.8. i8026W_Set_AVGCnt / pac_i8026W_Set_AVGCnt

This function is used to set the number of calculation averages

Syntax

For MiniOS7

```
void i8026W_Set_AVGCnt(int slot, unsigned short cnt);
```

For Windows (CE and WES)

```
void pac_i8026W_Set_AVGCnt(int slot, unsigned short cnt);
```

For Linux

N/A

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

cnt [in]

Specifies the channel number(1 ~ 65535).

Return Values

None

Examples

[C++]

```
int slot, cnt;  
i8026W_Set_AVGCnt(slot,cnt);
```

[C#]

```
int slot ,cnt;  
pac8026WNet.pac8026W.Set_AVGCnt (slot, cnt);
```

4.9. i8026W_ReadAO_GainOffset / pac_i8026W_ReadAO_GainOffset

This function is used to read the calibrated AO gain and offset value.

Syntax

For MiniOS7

```
void i8026W_ReadAO_GainOffset(  
    int slot,  
    int chIndex,  
    short gain,  
    unsigned short *gainValue,  
    short *offsetValue  
);
```

For Windows (CE and WES)

```
void pac_i8026W_ReadAO_GainOffset(  
    int slot,  
    int chIndex,  
    short gain,  
    unsigned short *gainValue,  
    short *offsetValue  
);
```

For Linux

```
void i8026W_ReadAOGainOffset(  
    int slot,  
    int ch,  
    int gain,  
    unsigned short* gainValue,  
    short* offsetValue  
);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 2)

gain [in]

specifies the input type (0 ~ 4),

0: +/-10 V, 1: +/-5 V, 2: 0 ~ 10 V,

3: 0 ~ 5 V, 4: 0 ~ 20 mA

****gainValue [Out]***

the reference gain value for the Analog Input type.

****offsetValue [Out]***

the reference offset value for the Analog Input type.

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot;
short ch;
short gain;
unsigned short GainValue;
short OffsetValue;
i8026W_ReadAO_GainOffset(slot,ch,gain,&GainValue,&OffsetValue);
```

[C#]

```
Int slot;
Int16 ch;
short gain;
UInt16 GainValue;
Int16 OffsetValue;
pac8026WNet.pac8026W.ReadAO_GainOffset(slot,ch,gain,ref GainValue,ref OffsetValue);
```

[C] (For LinPAC)

```
int main(){
    unsigned short gVal=0;
    short oVal=0;
    int slot, ch, gain,ret;
    ret=Open_Slot(slot);
    if (ret > 0) {
        printf("Open Slot%d failed, return value=%d \n", slot, ret);
        return (-1);
    }

    i8026W_Init(slot);
    i8026W_ReadAOGainOffset(slot,ch,gain,&gVal, &oVal);
    Close_Slot(slot);
    return 0;
}
```

4.10. i8026W_WriteAO / pac_i8026W_WriteAO

This function is used to write the output value to a Analog Output channel in floating point format.

Syntax

For MiniOS7

```
short i8026W_WriteAO(int slot,int ch,short gain,float aoData);
```

For Windows (CE and WES)

```
short pac_i8026W_WriteAO (int slot,int ch,short gain,float aoData);
```

For Linux

```
short i8026W_WriteAO(int slot,int ch,short gain,float fData);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 2)

gain [in]

specifies the input type (0 ~ 4),

0: +/-10 V, 1: +/-5 V, 2: 0 ~ 10 V,
3: 0 ~ 5 V, 4: 0 ~ 20 mA

fData [in]

the Analog Output data in floating point format

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot, ch, gain;
float fVal;
i8026W_WriteAO( slot, ch, gain, fVal);
```

[C#]

```
int slot ,ch, gain;
float fval;
pac8026WNet.pac8026W.WriteAO(slot,ch,gain,fval);
```

[C] (For LinPAC)

```
int main(){
    int slot, ch, gain ,ret;
    float fval;
    ret=Open_Slot(slot);
    if (ret > 0) {
        printf("Open Slot%d failed, return value=%d \n", slot, ret);
        return (-1);
    }

    i8026W_Init(slot);
    i8026W_WriteAO(slot,ch,gain,fval);
    Close_Slot(slot);
    return 0;
}
```

4.11. i8026W_ReadbackAO / pac_i8026W_ReadbackAO

This function is used to read the output value of an Analog Output channel in floating-point format.

Syntax

For MiniOS7

```
short i8026W_ReadbackAO(int slot,int ch,float* fVal);
```

For Windows (CE and WES)

```
short pac_i8026W_ReadbackAO (int slot ,int ch, float* fVal);
```

For Linux

```
short float i8026W_ReadbackAO(int slot, int ch, float* fVal);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 2)

****fVal [Out]***

The value read from memory that is written to the module

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot, ch;  
float data;  
i8026W_ReadbackAO(slot, ch, &data);
```

[C#]

```
int slot ,ch;  
float data;  
pac8026WNet.pac8026W.AOReadback (slot, ch,ref data);
```

[C] (For LinPAC)

```
int main(){  
    int slot, ch, ret;  
    float data;  
    ret=Open_Slot(slot);  
    if (ret > 0) {  
        printf("Open Slot%d failed, return value=%d \n", slot, ret);  
        return (-1);  
    }  
  
    i8026W_Init(slot);  
    i8026W_ReadbackAO(slot, ch, data);  
    Close_Slot(slot);  
    return 0;  
}
```

4.12. i8026W_WriteAOHex / pac_i8026W_WriteAOHex

This function is used to write the output value of an Analog Output channel in hexadecimal format.

Syntax

For MiniOS7

```
short i8026W_WriteAOHex (int slot ,int ch,short gain,short hData);
```

For Windows (CE and WES)

```
short pac_i8026W_WriteAOHex (int slot ,int ch,short gain,short hData);
```

For Linux

```
short i8026W_WriteAOHex(int slot, int ch, short gain, short hData);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 2)

gain [in]

specifies the input type (0 ~ 4),

0: +/-10 V, 1: +/-5 V, 2: 0 ~ 10 V,
3: 0 ~ 5 V, 4: 0 ~ 20 mA

hData [in]

the Analog Output data in hexadecimal format

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot, ch, gain;
short hval;
i8026W_WriteAOHex( slot, ch, gain,hVal);
```

[C#]

```
int slot ,ch, gain;
short hval;
pac8026WNet.pac8026W. WriteAOHex (slot,ch,gain,hval);
```

[C] (For LinPAC)

```
int main(){
    int slot, ch, gain ,ret;
    short hval;
    ret=Open_Slot(slot);
    if (ret > 0) {
        printf("Open Slot%d failed, return value=%d \n", slot, ret);
        return (-1);
    }

    i8026W_Init(slot);
    i8026W_WriteAOHex(slot,ch,gain,hval);
    Close_Slot(slot);
    return 0;
}
```

4.13. i8026W_ReadbackAOHex / pac_i8026W_ReadbackAOHex

This function is used to read the output value of an Analog Output channel in hexadecimal format.

Syntax

For MiniOS7

```
short i8026W_ReadbackAOHex(int slot,int ch,short* hVal);
```

For Windows (CE and WES)

```
short pac_i8026W_ReadbackAOHex(int slot,int ch,short* hVal);
```

For Linux

```
short i8026W_ReadbackAOHex(int slot,int ch,short* hVal);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 2)

****hVal [Out]***

The value read from memory that is written to the module

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot, ch;  
short data;  
i8026W_ReadbackAOHex(slot, ch, &data);
```

[C#]

```
int slot ,ch;  
Int16 data;  
pac8026WNet.pac8026W.AOHexReadback(slot, ch,ref data);
```

[C] (For LinPAC)

```
int main(){  
    int slot, ch, ret;  
    short data;  
    ret=Open_Slot(slot);  
    if (ret > 0) {  
        printf("Open Slot%d failed, return value=%d \n", slot, ret);  
        return (-1);  
    }  
  
    i8026W_Init(slot);  
    i8026W_ReadbackAOHex(slot, ch, data);  
    Close_Slot(slot);  
    return 0;  
}
```

4.14. i8026W_ReadDIO / pac_i8026W_ReadDIO

This function is used to read Digital Input and Digital Output status.

Syntax

For MiniOS7

```
short i8026W_ReadDIO(  
    int slot,  
    short* diVal,  
    short* doVal,  
    unsigned char diBitArr[],  
    unsigned char doBitArr[]  
);
```

For Windows (CE and WES)

```
short pac_i8026W_ReadDIO(  
    int slot,  
    short* diVal,  
    short* doVal,  
    unsigned char diBitArr[],  
    unsigned char doBitArr[]  
);
```

For Linux

```
short i8026W_ReadDIO(  
    int slot,  
    short* diVal,  
    short* doVal,  
    unsigned char diBitArr[],  
    unsigned char doBitArr[]  
);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

****diVal [Out]***

The Digital Input status

****doVal [Out]***

The Digital Output status

diBitArr[] [Out]

The bit status of the corresponding Digital Input channel.

doBitArr[] [Out]

The bit status of the corresponding Digital Output channel.

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot;
short diVal, doVal;
unsigned char diBitArr[2], doBitArr[2]
i8026W_ReadDIO(slot, &diVal, &doVal, diBitArr, doBitArr);
```

[C#]

```
int slot;
short diVal, doVal;
byte[] diBit = new byte[2];
byte[] doBit = new byte[2];
pac8026WNet.pac8026W.ReadDIO(slot, ref diVal, ref doVal, diBit, doBit);
```

[C] (For LinPAC)

```
int main(){
    int slot=1, diVal=0, doVal=0;
    unsigned char diBitArr[2], doBitArr[2];
    ret=Open_Slot(slot);
    if (ret > 0) {
        printf("Open Slot%d failed, return value=%d \n", slot, ret);
        return (-1);
    }

    i8026W_Init(slot);
    i8026W_ReadDIO(slot, &diVal, &doVal, diBitArr, doBitArr);
    Close_Slot(slot);
    return 0;
}
```

4.15. i8026W_WriteDO / pac_i8026W_WriteDO

This function is used to write Digital Output status.

Syntax

For MiniOS7

```
void i8026W_WriteDO(int slot, short doVal);
```

For Windows (CE and WES)

```
void pac_i8026W_WriteDO(int slot, short doVal);
```

For Linux

```
short i8026W_WriteDO(int slot, short doVal);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

doVal [in]

The Digital Output value (0 ~ 3).

Return Values

Refer to Appendix A: “Error Code” for more details.

Examples

[C++]

```
int slot;  
short doVal;  
i8026W_WriteDO(slot, doVal);
```

[C#]

```
int slot;  
short doVal;  
pac8026WNet.pac8026W.WriteDO(slot, doVal);
```

[C] (For LinPAC)

```
int main(){  
    int slot=1, doVal=0;  
    ret=Open_Slot(slot);  
    if (ret > 0) {  
        printf("Open Slot%d failed, return value=%d \n", slot, ret);  
        return (-1);  
    }  
  
    i8026W_Init(slot);  
    i8026W_WriteDO(slot, doVal);  
    Close_Slot(slot);  
    return 0;  
}
```

4.16. i8026W_WriteDOBit / pac_i8026W_WriteDOBit

This function is used to write one Digital Output channel.

Syntax

For MiniOS7

```
void i8026W_WriteDOBit(int slot,int ch, short bitVal);
```

For Windows (CE and WES)

```
void pac_i8026W_WriteDOBit(int slot,int ch, short bitVal);
```

For Linux

```
short i8026W_WriteDOBit(int slot, int ch, int bitVal);
```

Parameter

slot [in]

Specific slot number (0 - 7), except range of slot is number 1 ~ 8 for LinPAC.

ch [in]

Specifies the channel number (0 ~ 2)

bitVal [in]

The status of the Digital Output channel.

Return Values

Refer to Appendix A: "Error Code" for more details.

Examples

[C++]

```
int slot,ch,status;  
i8026W_WriteDOBit(slot, ch, status);
```

[C#]

```
int slot,ch;  
short dobit;  
pac8026WNet.pac8026W.WriteDOBit(slot,ch,dobit);
```

[C] (For LinPAC)

```
int main(){  
    int slot=1, ch=0;  
    short dobit;  
    ret=Open_Slot(slot);  
    if (ret > 0) {  
        printf("Open Slot%d failed, return value=%d \n", slot, ret);  
        return (-1);  
    }  
  
    i8026W_Init(slot);  
    i8026W_WriteDOBit(slot, ch, dobit);  
    Close_Slot(slot);  
    return 0;  
}
```

Appendix A. Error Code

Error Code	Definition	Description
0	NoError	This indicates that there have been no errors
-1	ID_ERROR	There was a problem with the module ID
-2	SLOT_ERROR	There was a slot index error.
-3	CHANNEL_ERROR	There was a channel index error.
-4	GAIN_ERROR	There was a gain index error.
-6	NOT_SUPPORT_ERROR	Reading invalid value

Appendix B. Revision History

This chapter provides revision history information to this document.

The table below shows the revision history.

Revision	Date	Description
1.0.0	JUL. 2013	Initial issue
2.0.0	JUL. 2021	- Modify Quick start. - Modify the download path of libraries and demos. - Added optioning modules on Linux platform.
2021	Jan. 2025	Added Demo Programs