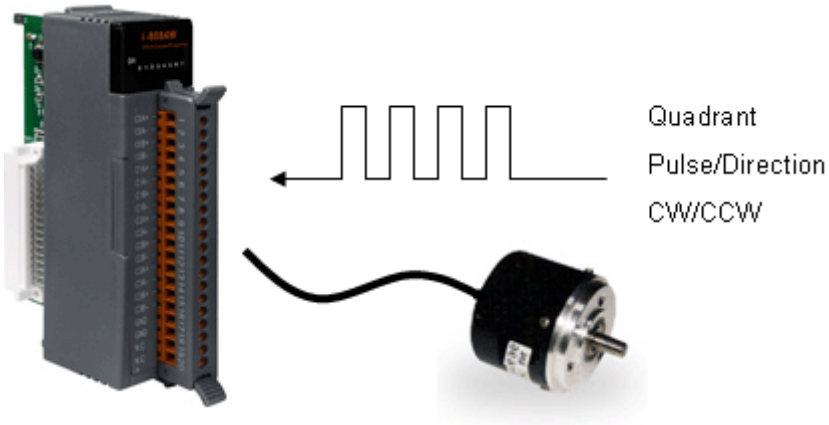
I-8093W

API Reference Manual

Version 1.0.1, August 2009

Service and usage information for

iPAC 8000 and WinPAC 8000



Written by Martin Hsu

Edited by Anna Huang

General Information

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, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only may be registered trademarks of their respective companies.

TABLE OF CONTENTS

1. Introduction.....5

1.1. Specification 6

1.2. Pin Assignment.....7

1.3. Block Diagram 8

1.4. Wiring Connection.....9

2. Hardware Operation Principle.....10

2.1. Counting Mode 11

2.2. Input Level 13

3. Usage and API for iPAC-8000 15

3.1. i8093W_Init.....16

3.2. i8093W_GetFirmwareVersion 18

3.3. i8093W_GetLibVersion.....19

3.4. i8093W_GetLibDate 20

3.5. i8093W_SetMode.....21

3.6. i8093W_GetMode 23

3.7. i8093W_SetXOR 25

3.8. i8093W_GetXOR.....26

3.9. i8093W_GetLineStatus 28

3.10. i8093W_GetIndex 30

3.11. i8093W_Read32BitEncoder 31

3.12. i8093W_ResetEncoder.....32

4. Usage and API for WinPAC-8000 33

4.1. pac_i8093W_Init 35
4.2. pac_i8093W_GetFirmwareVersion 37
4.3. pac_i8093W_GetLibVersion 38
4.4. pac_i8093W_GetLibDate 39
4.5. pac_i8093W_SetMode 40
4.6. pac_i8093W_GetMode 42
4.7. pac_i8093W_SetXOR 44
4.8. pac_i8093W_GetXOR 46
4.9. pac_i8093W_GetLineStatus 48
4.10. pac_i8093W_GetIndex 50
4.11. pac_i8093W_Read32BitEncoder 52
4.12. pac_i8093W_ResetEncoder 54

1. INTRODUCTION

I-8093W is a 3-axis encoder counter board on I-8000 platform. I-8093W encoder card has 32 bits counter and high counting rate 10Mpps. The application of I-8093W board is position/distance measurement, velocity measurement, feedback for motor control, hard wheel input and so on.

Features:

- ✓ 3-axis, 32-bit encoder counter
- ✓ Encoder counting mode: CW/CCW , Pulse/Direction, A/B Phase
- ✓ Maximum counting rate: 4 MHz
- ✓ Encoder Input: A, B, Z differential
- ✓ Input Level: 5V, 12V/24V with external resistor
- ✓ A/B/Z signal isolation voltage: 2500V optical isolation
- ✓ Built-in XOR logic for active high or active low encoder input

1.1. Specification

Parallel I/O Module

3-axis, 32 bits encoder counter.

Encoder Mode

1. CW/CCW mode
2. Pulse/Dir mode
3. Quadrant mode

Maximum Counting Rate

1. CW/CCW mode 4M pulse/sec
2. Pulse/Dir mode 4M pulse/sec
3. Quadrant mode 1M pulse/sec

Input Level

1. Input level 5 V

   Logic High: 4 V ~ 5 V

   Logic Low: 0 V ~ 2 V
2. Input 12 V with external resistor 1 K ohm

   Logic High: 5 V ~ 12 V

   Logic Low: 0 V ~ 2 V
3. Input 24V with external resistor 2K ohm

   Logic High: 7 V ~ 24 V

   Logic Low: 0 V ~ 2 V

Isolated voltage output

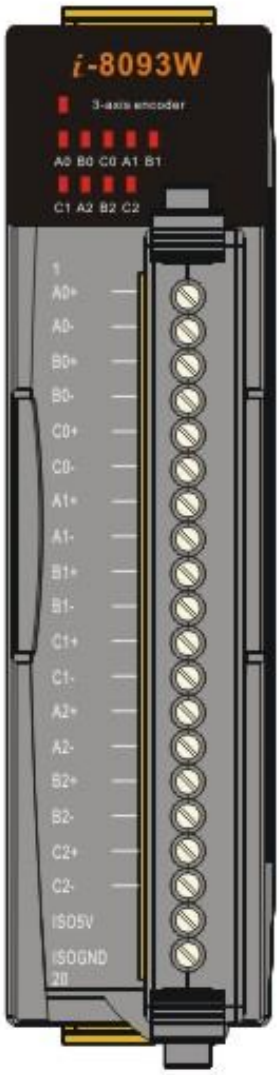
+5 V, 200 mA (max)

Power consumption

1.0 W

Built-in XOR logic for active high or low

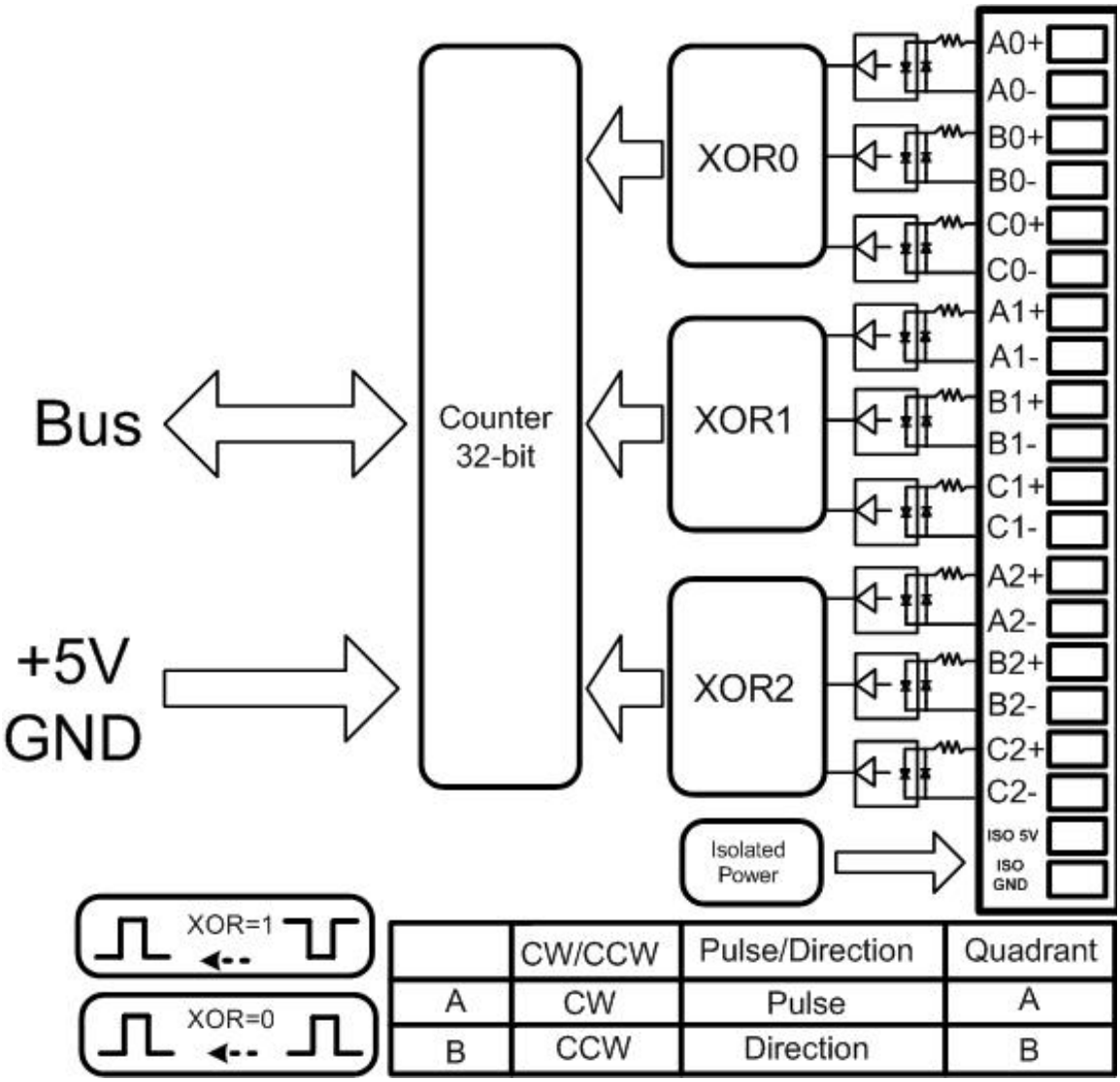
1.2. Pin Assignment



Terminal No.	Pin Assignment Name
01	A0+
02	A0-
03	B0+
04	B0-
05	C0+
06	C0-
07	A1+
08	A1-
09	B1+
10	B1-
11	C1+
12	C1-
13	A2+
14	A2-
15	B2+
16	B2-
17	C2+
18	C2-
19	ISO5V
20	ISOGND

- ✓ Pin 1 ~ 6: A0+/A0-/B0+/B0-/Z0+/Z0-, are designed for encoder0
- ✓ Pin 7 ~ 12: A1+/A1-/B1+/B1-/Z1+/Z1-, are designed for encoder1
- ✓ Pin 13~ 18: A2+/A2-/B2+/B2-/Z2+/Z2-, are designed for encoder2
- ✓ Pin 19 is isolated 5Vsupply, max: 50mA.
- ✓ Pin 20 is isolated ground.

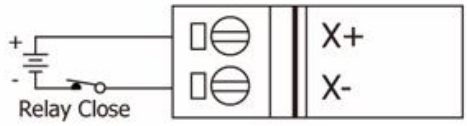
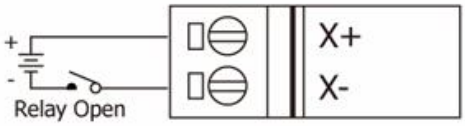
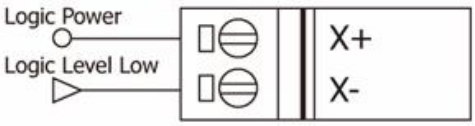
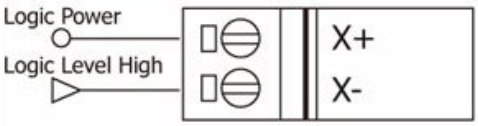
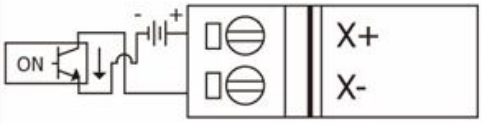
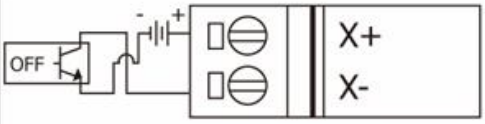
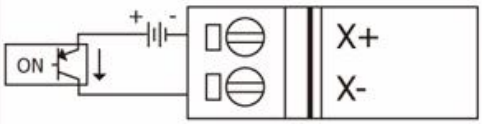
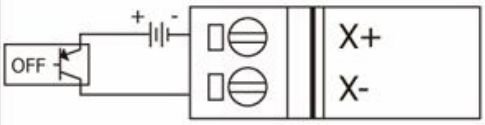
1.3. Block Diagram



	CW/CCW	Pulse/Direction	Quadrant
A	CW	Pulse	A
B	CCW	Direction	B

The input signal maybe active low or active high. The XOR0/XOR1/XOR2 are designed to invert the active low signal for internal logic requirement. If the value of XOR0/1/2 is the encoder value will have different errors. Refer to chapter 2 for more information.

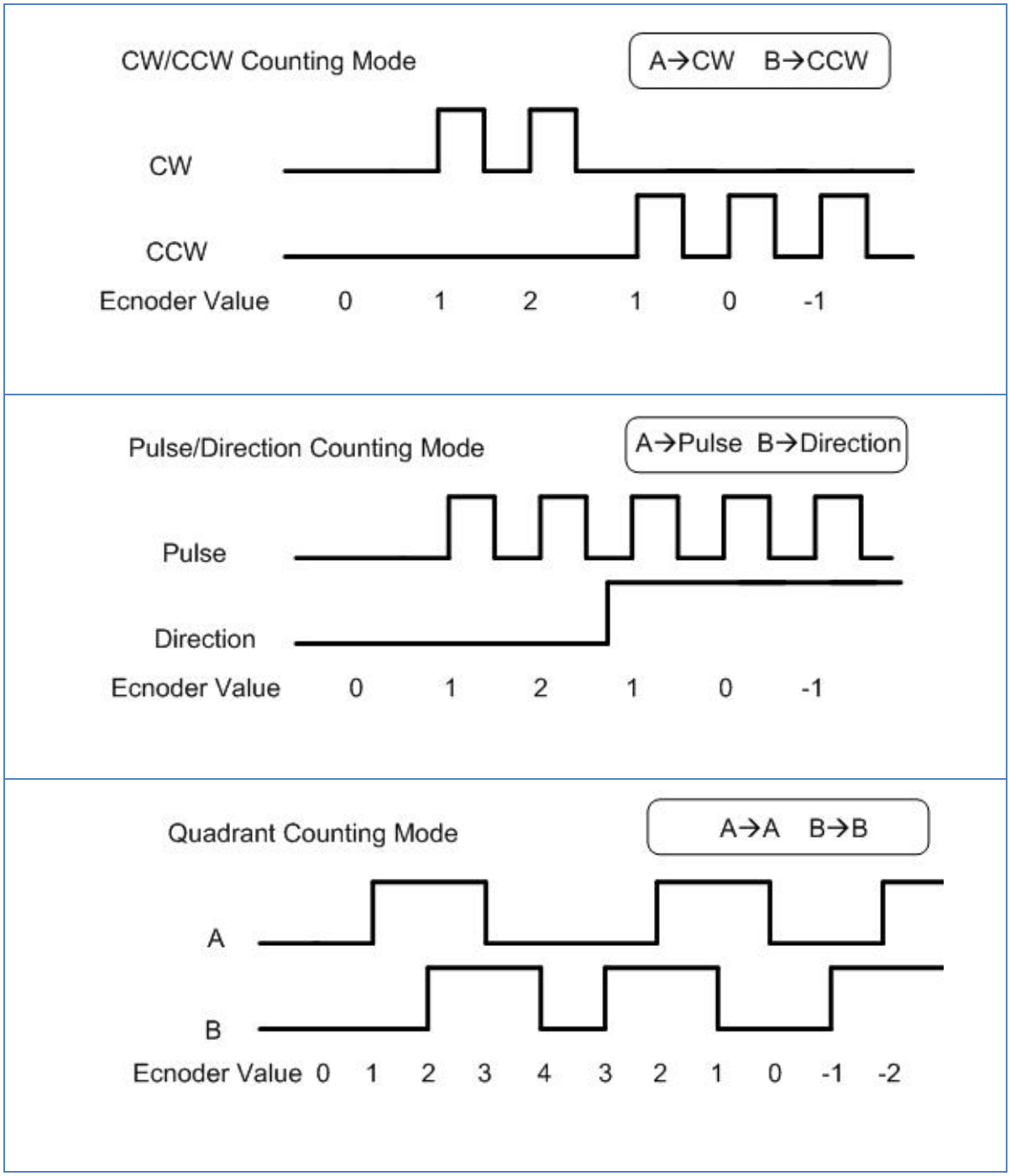
1.4. Wiring Connection

Input Type	ON State LED ON Readback as 0	OFF State LED OFF Readback as 1
Relay Contact	Relay ON	Relay Off
		
TTL/CMOS Logic	Voltage > 4V	Voltage < 0.8V
		
NPN Output	Open Collector On	Open Collector Off
		
PNP Output	Open Collector On	Open Collector Off
		

2. HARDWARE OPERATION PRINCIPLE

2.1. Counting Mode

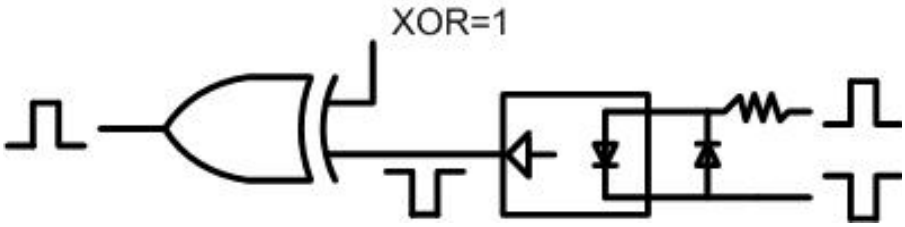
There are 3 counting modes, CC/PD/AB, given as follows:



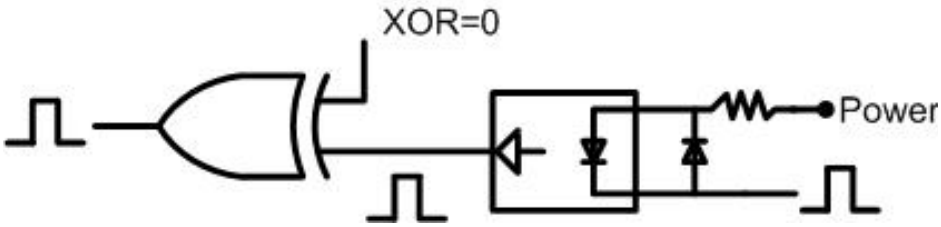
Note: -1=0xffffffff, -2=0xfffffffe

The internal counting logic is expected as active high. User can use XOR control bit to select the proper waveform as follows:

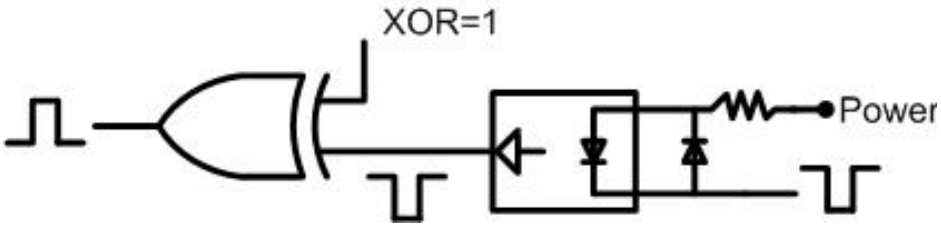
Case 1: differential input, set XOR=1



Case 2: active high single-ended input, set XOR=0



Case 3: active low single-ended input, set XOR=1

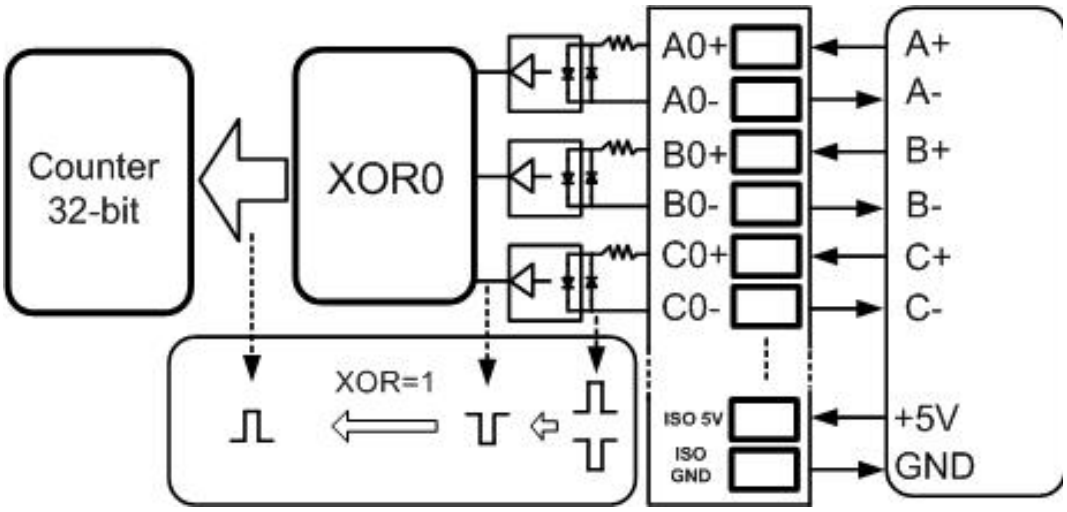


If the value of XOR control bit is error, the encoder value will have different errors given as follows:

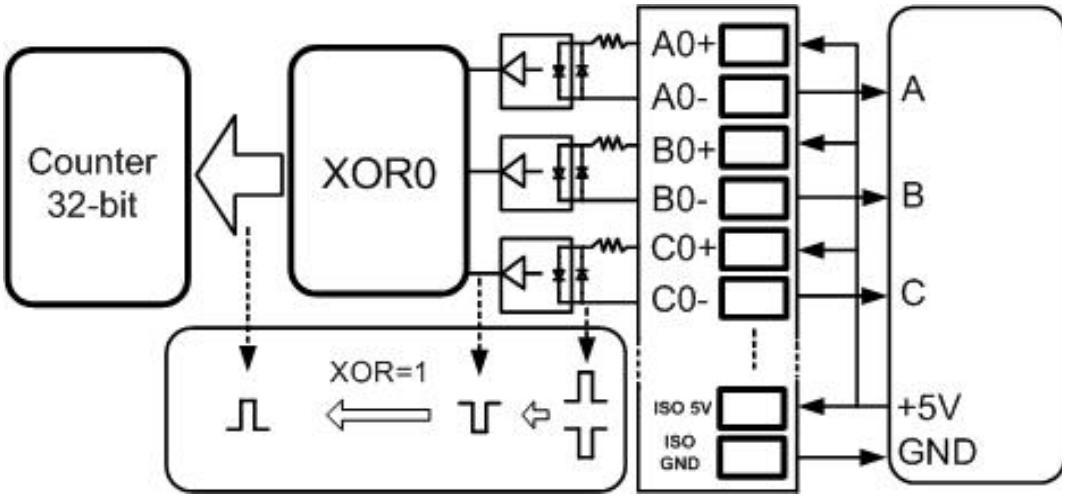
- ✓ The counting direction will be inverted
- ✓ The Z is inverted

2.2. Input Level

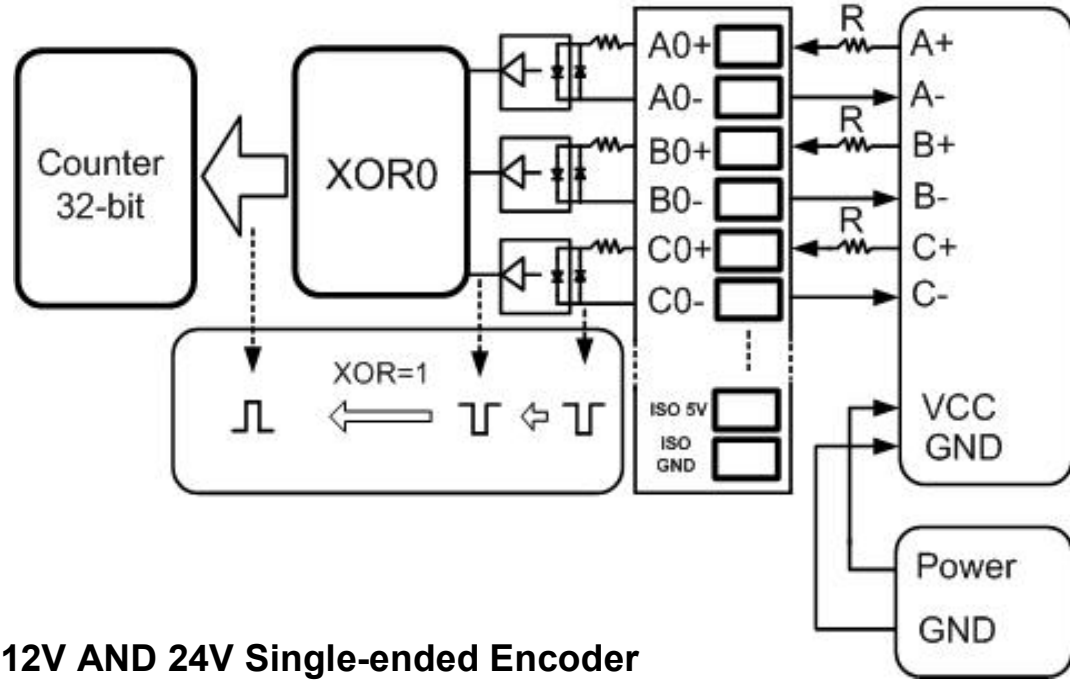
5V Differential Encoder



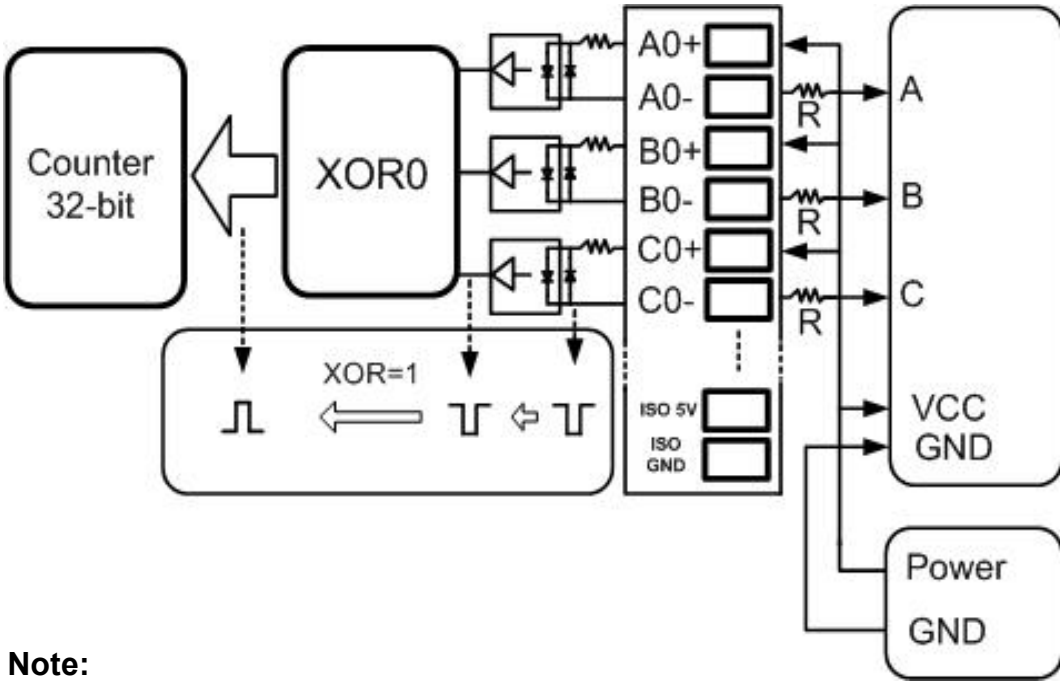
5V Single-ended Encoder



12V AND 24 V Differential Encoder



12V AND 24V Single-ended Encoder



Note:

1. Input 12V with external resistor(R) 1K ohm
2. Input 24V with external resistor(R) 2K ohm

3. USAGE AND API FOR IPAC-8000

iPAC Introduction and Software Development

8084W can be plugged on the IPAC-8000 . Please refer to the Web site

http://www.icpdas.com/products/PAC/i-8000/ip-8x41.htm

or view the iPAC-8000 User manual for getting more information ：

http://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/document/

Software Development using C language

Please refer to the Web site

http://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/c_language_guide_eng.html

I-8093W Demo and library for IPAC-8000

The latest library and demo as below:

Library：

http://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/lib/

Demo：

http://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/io_in_slot/8093/

3.1. i8093W_Init

This function can initial the I-8093W and can check the hardware ID, if return 0 for input slot index, it means the ID of this slot index is I-8093W,if return -1, it means there is no I-8093W for this slot index.

Syntax

```
int i8093W_Init(int slot);
```

Parameter

slot: 0 ~ 7

Return Values

Please refer to Error Code Table.

Examples

[C]

```
int slotIndex,err;
err=i8093W_Init(slotIndex);
if(err==0)
{
  printf("There is an I-8093W at slot %d\n",slotIndex);
}
else
{
  printf("There is no I-8093W at slot %d\n",slotIndex);
}
```

3.2. i8093W_GetFirmwareVersion

This function gets the firmware version of I-8093W hardware.

Syntax

```
short i8093W_GetFirmwareVersion(int slot);
```

Parameter

slot: 0 ~ 7

Return Values

The firmware version of I-8093W hardware

Examples

[C]

```
short firmware_version;
firmware_version = i8093W_GetFirmwareVersion (slot);
```

3.3. i8093W_GetLibVersion

This function gets the library version of i8093W.dll.

Syntax

```
short i8093W_GetLibVersion ();
```

Parameter

none

Return Values

The library version of i8093W.dll

Examples

[C]

```
short version;
version = i8093W_GetLibVersion ();
```

3.4. i8093W_GetLibDate

This function gets the library built date of i8093W.dll.

Syntax

```
void i8093W_GetLibDate(char *LibDate);
```

Parameter

LibDate the string buffer of library built date

Return Values

The library built date of i8093W.dll

Examples

[C]

```
char lib_date[32];
i8093W_GetLibDate(lib_date);
```

3.5. i8093W_SetMode

This function to set the operation mode of I-8093W.

Syntax

```
int i8093W_SetMode(int slot, int ch, int mode);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

mode:

1: CW/CCW counting mode

2: Pulse/Direction counting mode

3: AB Phase (Quadrant counting) mode

Return Values

Please refer to Error Code Table.

Examples

[C]

```
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  i8093W_SetMode(slot,ch,3); // AB Phase
}
```

3.6. i8093W_GetMode

This function to get the operation mode of I-8093W.

Syntax

```
int i8093W_GetMode(int slot, int ch, int* mode);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

mode:

1: CW/CCW counting mode

2: Pulse/Direction counting mode

3: AB Phase (Quadrant counting) mode

Return Values

Please refer to Error Code Table.

Examples

[C]

```
int slot,ch,mode;
slot = 0;
for(ch=0;ch<3;ch++)
{
  i8093W_SetMode(slot,ch,&mode);
}
```

3.7. i8093W_SetXOR

This function to set the xor of I-8093W for each channel.

Syntax

```
int i8093W_SetXOR (int slot, int ch, int xor);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

xor: 0/1

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  i8093W_ SetXOR (slot,ch,0);
}
```

3.8. i8093W_GetXOR

This function to get the xor of I-8093W for each channel.

Syntax

```
int i8093W_GetXOR (int slot, int ch, int* mode);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

mode:

1: CW/CCW counting mode

2: Pulse/Direction counting mode

3: AB Phase (Quadrant counting) mode

Return Values

Please refer to Error Code Table.

Examples

[C]

```
int slot,ch,mode;
slot = 0;
for(ch=0;ch<3;ch++)
{
  i8093W_SetMode(slot,ch,&mode);
}
```

3.9. i8093W_GetLineStatus

This function to get A,B and Z status of I-8093W.

Syntax

```
int i8093W_GetLineStatus (int slot, int ch, int* A,int*B,int*Z);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

A: 0/1

B: 0/1

Z: 0/1

Return Values

Please refer to Error Code Table.

Examples

[C]

```
int slot,ch,A,B,Z;
slot = 0;
for(ch=0;ch<3;ch++)
{
  i8093W_GetLineStatus (slot,ch,&A,&B,&Z);
}
```

3.10. i8093W_GetIndex

This function to get Z index status of I-8093W.

Syntax

```
int i8093W_GetIndex (int slot, int ch, int*Z);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

Z: 0/1

Return Values

Please refer to Error Code Table.

Examples

[C]

```
int slot,ch,Z;
slot = 0;
for(ch=0;ch<3;ch++)
{
  i8093W_GetIndex (slot,ch,&Z);
}
```

3.11. i8093W_Read32BitEncoder

This function to get 32-Bit Encoder value of I-8093W.

Syntax

```
int i8093W_Read32BitEncoder(int slot, int ch, long*encVal);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

encVal: 32-Bit Encoder value of I-8093W (0 ~ 0xffffffff)

Return Values

Please refer to Error Code Table.

Examples

[C]

```
int slot,ch;
long encoder=0;
slot = 0;
for(ch=0;ch<3;ch++)
{
  i8093W_ Read32BitEncoder(slot,ch,&encoder);
}
```

3.12. i8093W_ResetEncoder

This function to reset 32-Bit Encoder value to zero.

Syntax

```
int i8093W_ResetEncoder (int slot, int ch);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

Return Values

Please refer to Error Code Table.

Examples

[C]

```
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  i8093W_ResetEncoder (slot,ch);
}
```

4. API FOR WINPAC-8000

WinPAC Introduction and Software Development

WinPAC-8000 Introduction and user rmanual

I-8093W can be plugged on the WinPAC-8000 . Please refer to Web site

http://www.icpdas.com/products/PAC/winpac/introduction.htm

or view the WinPAC User manual for getting more information

http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/document/

Software Development using eMbedded Visual C++ or .NET

Both eMbedded Visual C++ and Visual Studio .NET can develop the programon on the WinPAC-8000. Please refer to the Web site:

http://www.icpdas.com/products/PAC/winpac/download/winpac_8000/download_documents.htm

and select the necessary document

I-8093W Demo and library for WinPAC-8000

Please refer to this page to download WinPAC-8000I demo

http://www.icpdas.com/products/PAC/winpac/download/winpac_8000/download_demo.htm

The latest 8084W library and demo as below:

Library:

.Net

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/io_modules/dotnet/

eVC:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/io_modules/evc/

Demo:

.NET:
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/dotnet/c%23.net/pac_io/local/

eVC:

ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/evc/pac_io/local/

4.1. pac_i8093_Init

This function can initial the I-8093W and can check the hardware ID, if return 0 for input slot index, it means the ID of this slot index is I-8093W,if return -1, it means there is no I-8093W for this slot index.

Syntax

```
int pac_i8093_Init(int slot);
```

Parameter

Slot: 0 ~ 7

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slotIndex,err;
err=pac_i8093_Init(slotIndex);
if(err==0)
{
  printf("There is an I-8093W at slot %d\n",slotIndex);
}
Else
{
  printf("There is no I-8093W at slot %d\n",slotIndex);
}
```

[C#]

```
using pac8093WNet;
int slotIndex,err;
err= pac8093W.Init(slotIndex);
if(err==0)
{
  printf("There is an I-8093W at slot %d\n",slotIndex);
}
else
{
  printf("There is no I-8093W at slot %d\n",slotIndex);
}
```

4.2. pac_i8093_GetFirmwareVersion

This function gets the firmware version of I-8093W hardware.

Syntax

```
short pac_i8093_GetFirmwareVersion(int slot);
```

Parameter

slot

Return Values

The firmware version of I-8093W hardware

Examples

[C++]

```
short firmware_version;
firmware_version = pac_i8093_GetFirmwareVersion (slot);
```

[C#]

```
using pac8093WNet;
short version;
firmware_version = pac8093W.GetFirmwareVersion (slot);
```

4.3. pac_i8093_GetLibVersion

This function gets the library version of pac_i8093.dll.

Syntax

```
short pac_i8093_GetLibVersion ();
```

Parameter

none

Return Values

The library version of pac_i8093.dll

Examples

[C++]

```
short version;
version = pac_i8093_GetLibVersion ();
```

[C#]

```
using pac8093WNet;
short version;
version = pac8093W.GetLibVersion ( );
```

4.4. pac_i8093_GetLibDate

This function gets the library built date of pac_i8093.dll.

Syntax

```
void pac_i8093_GetLibDate(char *LibDate);
```

Parameter

LibDate the string buffer of library built date

Return Values

The library built date of pac_i8093.dll

Examples

[C++]

```
char lib_date[32];
pac_i8093_GetLibDate(lib_date);
```

[C#]

```
using pac8093WNet;
string lib_date;
lib_date = pac8093W. GetLibDate ( );
```

4.5. pac_i8093_SetMode

This function to set the operation mode of I-8093W.

Syntax

```
int pac_i8093_SetMode(int slot, int ch, int mode);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

mode:

1: CW/CCW counting mode

2: Pulse/Direction counting mode

3: AB Phase (Quadrant counting) mode

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac_i8093_SetMode(slot,ch,3); // AB Phase
}
```

[C#]

```
using pac8093WNet;
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac8093W.SetMode(slot,ch,3); // AB Phase
}
```

4.6. pac_i8093_GetMode

This function to get the operation mode of I-8093W.

Syntax

```
int pac_i8093_GetMode(int slot, int ch, int* mode);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

mode:

1: CW/CCW counting mode

2: Pulse/Direction counting mode

3: AB Phase (Quadrant counting) mode

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch,mode;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac_i8093_SetMode(slot,ch,&mode);
}
```

[C#]

```
using pac8093WNet;
int slot,ch,mode=0;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac8093W.GetMode(slot,ch,ref mode);
}
```

4.7. pac_i8093_SetXOR

This function to set the xor of I-8093W for each channel.

Syntax

```
int pac_i8093_SetXOR (int slot, int ch, int xor);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

xor: 0/1

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac_i8093_ SetXOR (slot,ch,0);
}
```

[C#]

```
using pac8093WNet;
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac8093W. SetXOR (slot,ch,0);
}
```

4.8. pac_i8093_GetXOR

This function to get the xor of I-8093W for each channel.

Syntax

```
int pac_i8093_GetXOR (int slot, int ch, int* mode);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

mode:

1: CW/CCW counting mode

2: Pulse/Direction counting mode

3: AB Phase (Quadrant counting) mode

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch,mode;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac_i8093_SetMode(slot,ch,&mode);
}
```

[C#]

```
using pac8093WNet;
int slot,ch,xor=0;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac8093W.GetXOR(slot,ch,ref xor);
}
```

4.9. pac_i8093_GetLineStatus

This function to get A,B and Z status of I-8093W.

Syntax

```
int pac_i8093_GetLineStatus (int slot, int ch, int* A,int*B,int*Z);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

A: 0/1

B: 0/1

Z: 0/1

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch,A,B,Z;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac_i8093_GetLineStatus (slot,ch,&A,&B,&Z);
}
```

[C#]

```
using pac8093WNet;
int slot,ch,A=0,B=0,Z=0;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac8093W.GetLineStatus (slot,ch,ref A,ref B, ref Z);
}
```

4.10. pac_i8093_GetIndex

This function to get Z index status of I-8093W.

Syntax

```
int pac_i8093_GetIndex (int slot, int ch, int*Z);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

Z: 0/1

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch,Z;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac_i8093_GetIndex (slot,ch,&Z);
}
```

[C#]

```
using pac8093WNet;
int slot,ch, Z=0;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac8093W.GetIndex(slot,ch, ref Z);
}
```

4.11. pac_i8093_Read32BitEncoder

This function to get 32-Bit Encoder value of I-8093W.

Syntax

```
int pac_i8093_Read32BitEncoder(int slot, int ch, long*encVal);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

encVal: 32-Bit Encoder value of I-8093W (0 ~ 0xffffffff)

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch;
long encoder=0;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac_i8093_ Read32BitEncoder(slot,ch,&encoder);
}
```

[C#]

```
using pac8093WNet;
int slot,ch, Z=0;
long encoder=0
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac8093W. ReadEncoder(slot,ch, ref encoder);
}
```

4.12. pac_i8093_ResetEncoder

This function to reset 32-Bit Encoder value to zero.

Syntax

```
int pac_i8093_ResetEncoder (int slot, int ch);
```

Parameter

slot: 0 ~ 7

ch: 0 ~ 2

Return Values

Please refer to Error Code Table.

Examples

[C++]

```
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac_i8093_ResetEncoder (slot,ch);
}
```

[C#]

```
using pac8093WNet;
int slot,ch;
slot = 0;
for(ch=0;ch<3;ch++)
{
  pac8093W.ResetEncoder (slot,ch);
}
```

APPENDIX A. ERROR CODE

0	OK
-1	ID_ERROR
-2	SLOT_OUT_RANGE
-3	CHANNEL_OUT_RANGE
-4	MODE_ERROR