I-8093W/I-9093

I/O Module User Manual

V2.0.0 July 2018

Written by Edward Ku

Edited by Anna Huang

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 © 2018 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

Preface

I-8093W/I-9093 are 3-axis encoder counter board. I-8093W/I-9093 encoder card has 32 bits counter and high counting rate 10Mpps. The application of I-8093W/I-9093 board is position/distance measurement, velocity measurement, feedback for motor control, hard wheel input and so on.

The information contained in this manual is divided into the following topics:

- Chapter 1, "Introduction" – This chapter provides information related to the hardware, such as the specifications, the jumper settings details and wiring information.
- Chapter 2, "Quick Start" – This chapter provides information on how to get started, an overview of the location of the demo programs, a "Getting Started Guide", and an outline of the calibration process.
- Chapter 3, "Compare Trig Out" – This chapter introduces the attributes related to the Compare Trig Out function, the programming procedures, and demo programs.
- Chapter 4, "API References" – This chapter describes the functions provided in the I-8014W library together with an explanation of the differences in the naming rules used for the MiniOS7 and Windows platforms.
- Chapter 5, "API References" – This chapter provides some troubleshooting solutions should you encounter any problems while operating the I-8014W.

1. Introduction

I-8093W is a 3-axis high speed encoder module. Its each axis can be independently configured as one of A/B Phase, Pulse/Direction and CW/CCW input mode.

I-9093 is also a 3-axis encoder which included A/B Phase, Pulse/Direction and CW/CCW input mode with compare trigger output function.
It can generate a periodic trigger signal when the motor reaches a specified position.
The specified position is called a breakpoint and is similar to a switch that is triggered after the motor passes a certain position.

The high-end specifications of I-8093W/I-9093 and complete software support make it ideal for wide range applications in position measurement of motion systems for industrial and laboratory environment.

Applicable Platform table

The following table shows which platform the module applies to.

Platform	OS	Module
XPAC	XP-8000(WES)	I-8093W
	XP-8000-Atom (WES)	I-8093W
	XP-8000-WES7 (WES7)	I-8093W
	XP-8000-CE6 (WinCE 6.0)	I-8093W
	XP-8000-Atom-CE6 (WinCE 6.0)	I-8093W
	XP-9000-WES7(WES7)	I-9093
WinPAC	WP-8000 (CE 5.0/7.0)	I-8093W
	WP-9000-CE7 (CE 7.0)	I-9093
LinPAC	LinPAC-8000(Linux kernel 3.2/4.4)	I-8093W
	LinPAC-9000(Linux kernel 3.2/4.4)	I-9093
IPAC	iPAC-8000 (MiniOS7)	I-8093W
	I-8000 (MiniOS7)	I-8093W

Features

Model	I-8093W	I-9093
Encoder Axis	3-axis	
Encoder Counter	32-bit	
Encoder counting mode	A/B Phase , Dir/Pulse , CW/CCW	
Input Level	5V/12V/24V	
Max. Speed	4M hz	6M hz
System LED Indicator	1 LED as Power Indicator 9 LED as Status Indicator	1 LED as Power Indicator 12 LED as Status Indicator
Dimension (L x W x H)	102 mm x 30 mm x 115 mm	144 mm x 30.3 mm x 134 mm

The difference between I-8093W and I-9093

Functions	I-8093W	I-9093
CW/CCW	✓	✓
Dir/Pulse	✓	✓
A/B Phase	✓	✓
Read Encoder	✓	✓
XOR Bit	✓	✓
Index Latched Status	✓	×
Read Frequency	✓	×
External Trig Mode	×	✓
Compared Trig Out	×	✓
Previous Trig Steps	×	✓
Low Pass Filter	×	✓

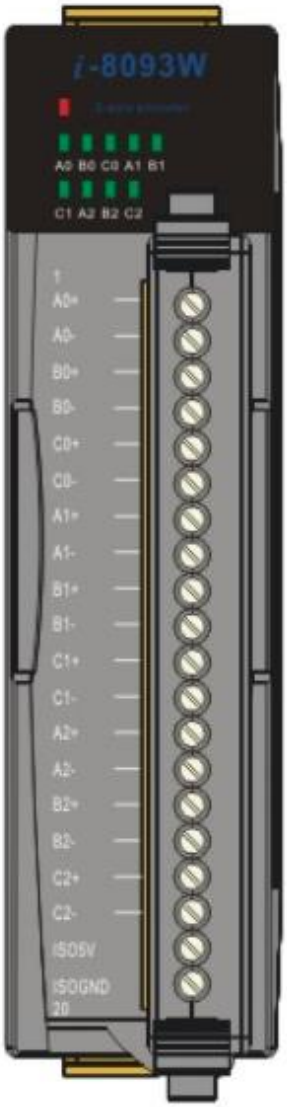
1.1. Specifications

Model		I-8093W	I-9093
Encoder Input			
Encoder Axis		3-axis	
Encoder Counter		32-bit	
Encoder Mode		A/B Phase , CW/CCW , Pulse/Dir	
Input Level	5V	Logic High: 4 V ~ 5 V Logic Low: 0 V ~ 2 V (Jumper Select)	Logic High:3.5 V ~ 5 V Logic Low:0.8V Max (Jumper Select)
	12V	Logic High: 5 V ~ 12 V Logic Low: 0 V ~ 2 V (external resistor 1 K ohm)	Logic High: 5 V ~ 12 V Logic Low: 0 V ~ 2 V (external resistor 1 K ohm)
	24V	Logic High: 7 V ~ 24 V. Logic Low: 0 V ~ 2 V (Jumper Select)	Logic High:10 V ~ 24 V Logic Low:0.8V Max (Jumper Select)
Max. Speed	Quadrant	1 MHz	2 MHz Max.
	CW/CCW	4 MHz	6 MHz
	Pulse/Dir	4 MHz	6 MHz
Programmable Digital Filter		--	1 ~ 250 µs
A/B/C signal isolation		2500 VDC	2500 VDC
External Input			
Channel		--	3
ON Voltage Level		--	+3.5 VDC ~ +5 VDC Or 10 VDC ~ 24 VDC(Jumper Select)
OFF Voltage Level		--	+0.8 VDC Max.
Trigger Output			
Channel		--	3
Trigger Pulse Width		--	10 uS~ 128 uS
Load Voltage		--	5 ~ 48 V
Max Load Current		--	100 mA
LED Indicators			
System LED Indicator		1 LED as Power Indicator 9 LED as Status Indicator	1 LED as Power Indicator 12 LED as Status Indicator
Isolation			
Intra-module Isolation, Field-to-Logic		2500 Vrms	3000 VDC

EMS Protection		
ESD(IEC 61000-4-2)	±4 kV Contact for Each Terminal	±4 kV Contact for Each Terminal
	--	±8 kV Air for Random Point
Power		
Power Consumption	2 W Max.	
Mechanical		
Dimension (L x W x H)	102 mm x 30 mm x 115 mm	144 mm x 30.3 mm x 134 mm
Environment		
Operating Temperature	-25 °C ~ +75°C	
Storage Temperature	-30 ~ 85 °C	-40 ~ +85°C
Humidity	5 ~ 95 % RH, Non-condensing	10 ~ 90% RH, non-condensing

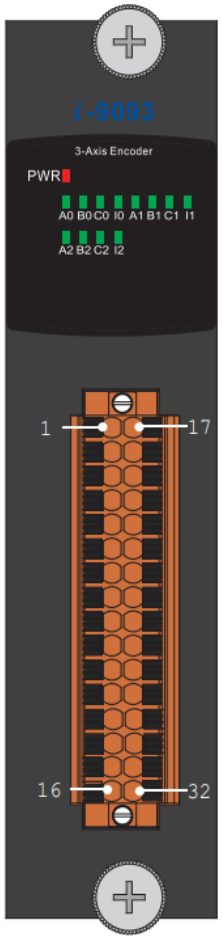
1.2. Pin Assignments

I-8093W



Terminal No.	Pin Assignment
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

I-9093

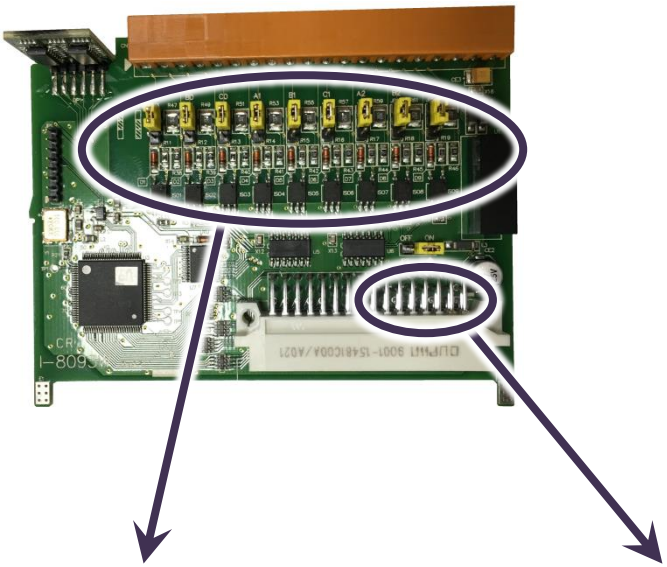


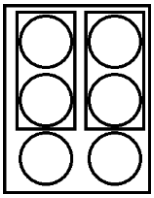
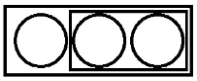
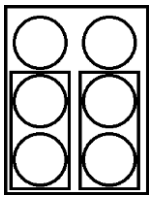
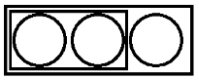
Pin Assignment	Terminal No.		Pin Assignment
A0+	01	17	A0-
B0+	02	18	B0-
C0+	03	19	C0-
I0+	04	20	I0-
Trig0	05	21	ISO.GND
A1+	06	22	A1-
B1+	07	23	B1-
C1+	08	24	C1-
I1+	09	25	I1-
Trig1	10	26	ISO.GND
A2+	11	27	A2-
B2+	12	28	B2-
C2+	13	29	C2-
I2+	14	30	I2-
Trig2	15	31	ISO.GND
ISO5V	16	32	ISO.GND

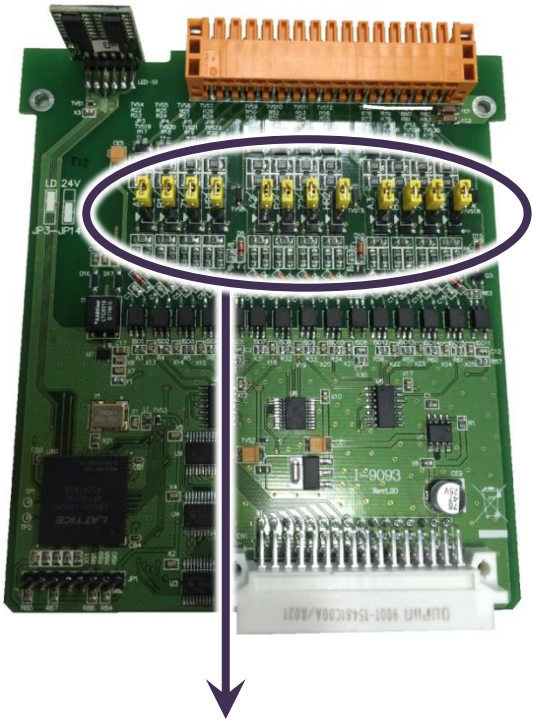
32-pin Connector

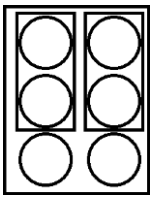
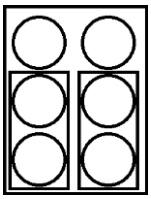
1.3. Jumper Settings

I-8093W



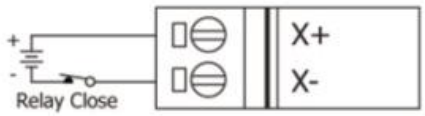
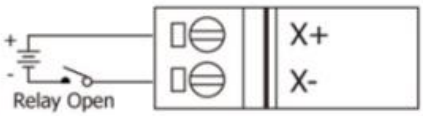
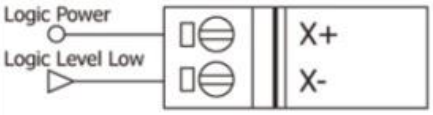
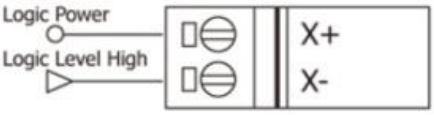
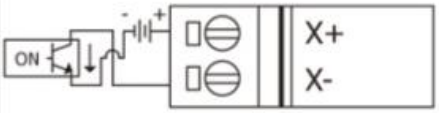
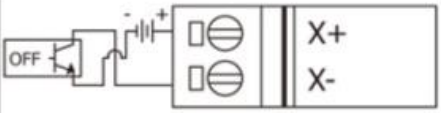
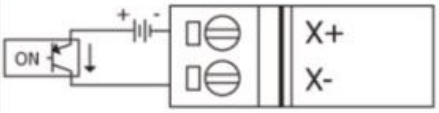
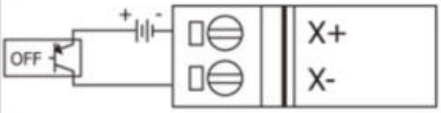
5V / 24V Input Level	Enable / Disable 5V power supply
5V	Enable
	
24V	Disable
	



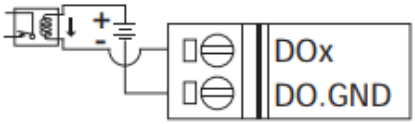
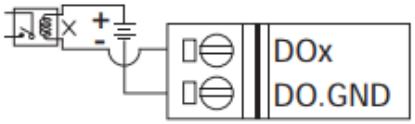
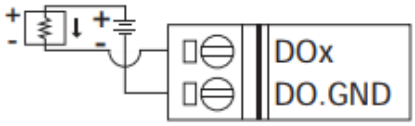
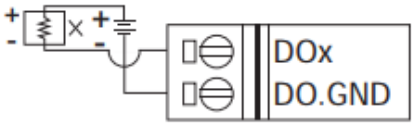
5V / 24V Input Level
5V

24V


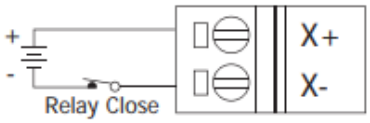
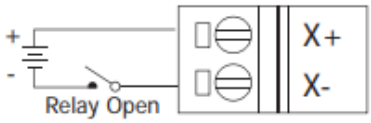
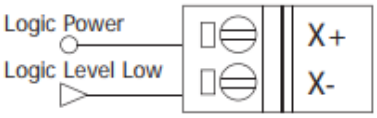
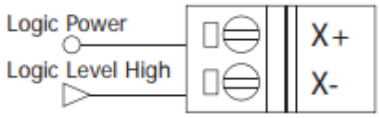
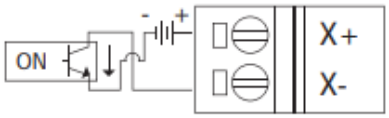
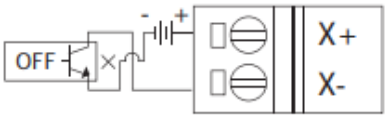
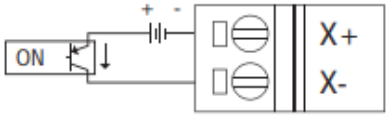
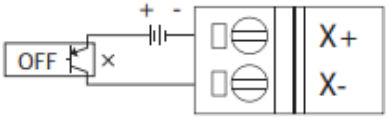
1.4. Wire Connections

I-8093W

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
		

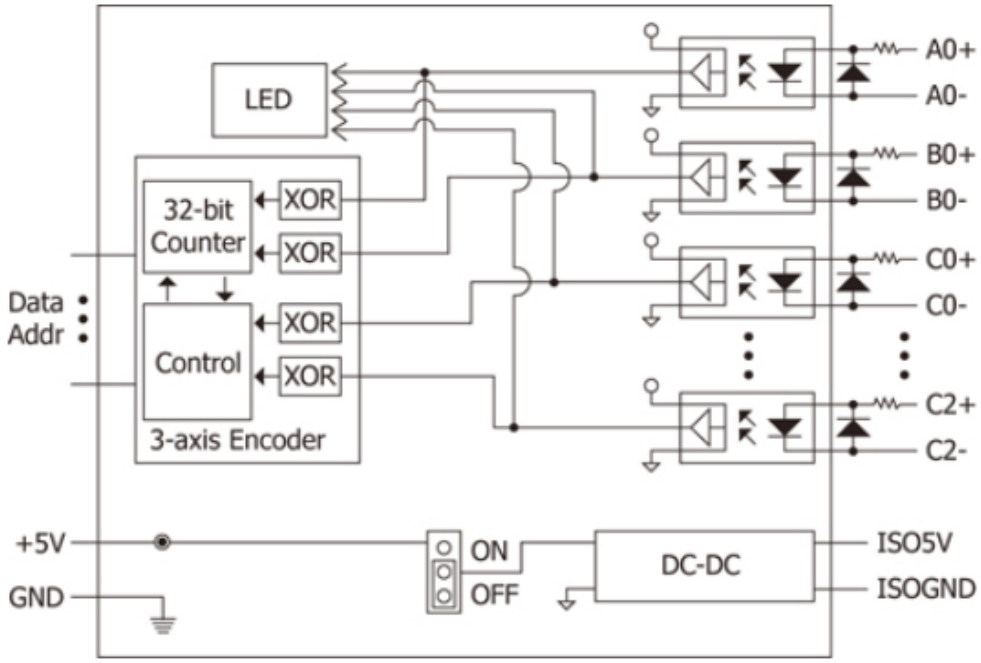
I-9093

Output Type	ON State Readback as 1	OFF State Readback as 0
Drive Relay	Relay ON	Relay OFF
		
Resistance Load		
		

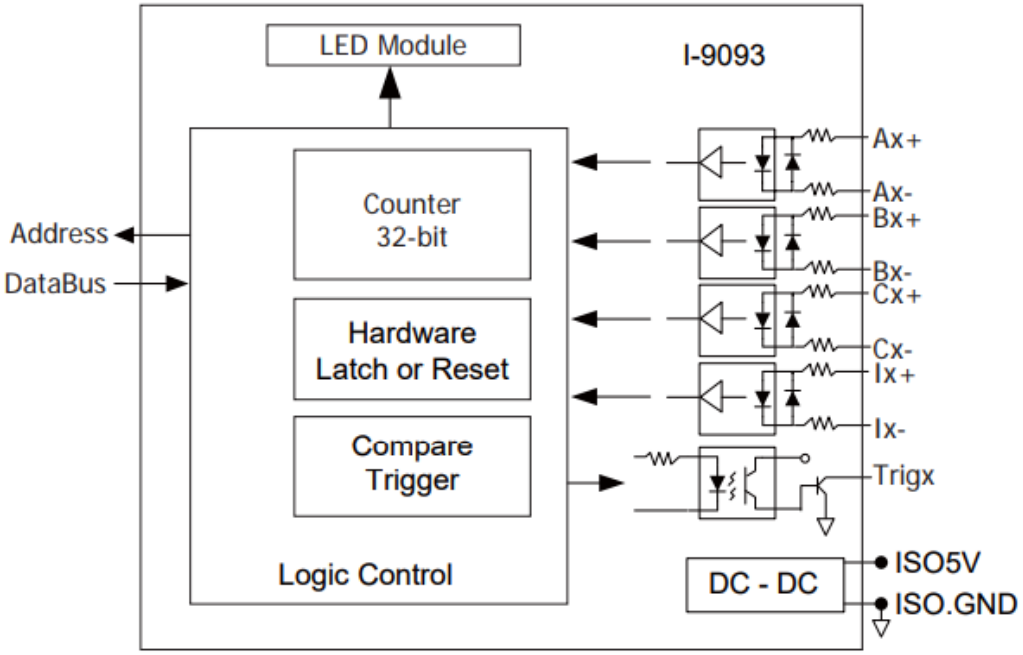
Input Type	ON State LED ON Readback as 1	OFF State LED OFF Readback as 0
Relay Contact	Relay ON	Relay OFF
		
TTL/CMOS Logic	Voltage > 4 V	Voltage < 0.8 V
		
NPN Output	Open Collector ON	Open Collector OFF
		
PNP Output	Open Collector ON	Open Collector OFF
		

1.5. Block Diagram

I-8093W



I-8014CW/I-9014C



2. Quick Start

This section provides a Getting-Started guide and details for using the I-8093W/I-9093 module on either the MiniOS7 or Windows platforms.

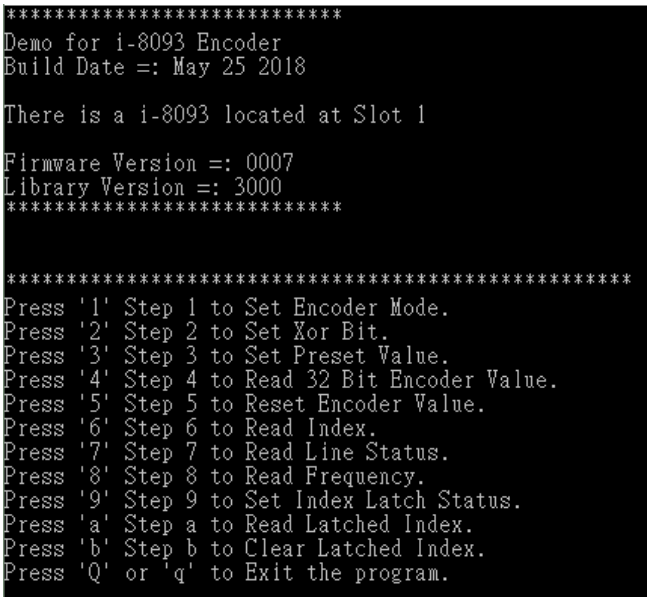
- **MiniOS7-based Controllers**
  I-8093W Getting started, see section 2.1.
- **Windows-based Controllers**
  I-8093W/I-9093 Getting started, see section 2.2.

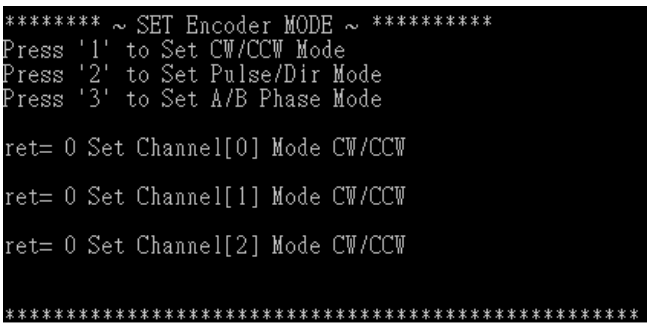
2.1. Getting start on MiniOS7-Based Controllers

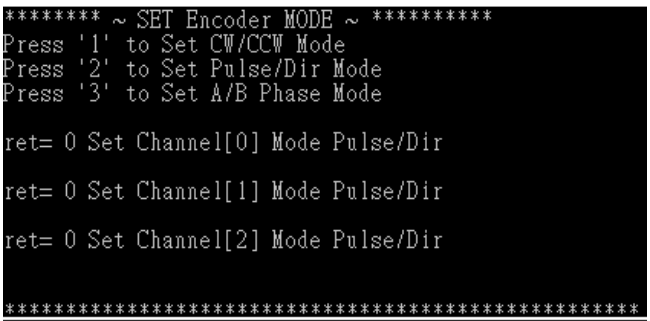
This part will show the functions of I-8093W with 8093DEMO.

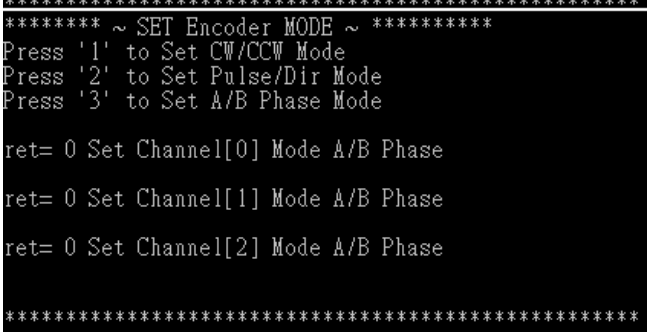
*To find the Demo path please refer to 1.7 Demo Programs.

After execute 8093DEMO you can see the functions as following picture:







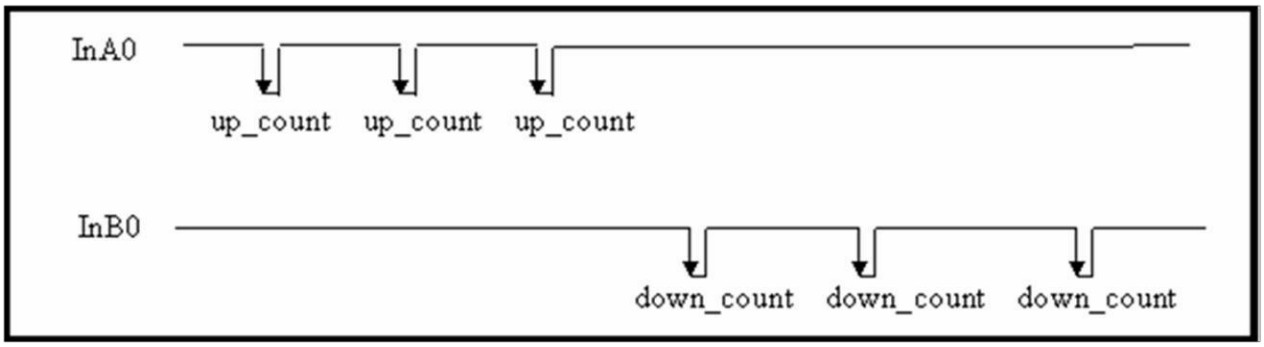


There are 4 options you can choose for each channel.

(1) (STOP , CW/CCW , Dir/Pulse , A/B Phase)

CW/CCWCounting

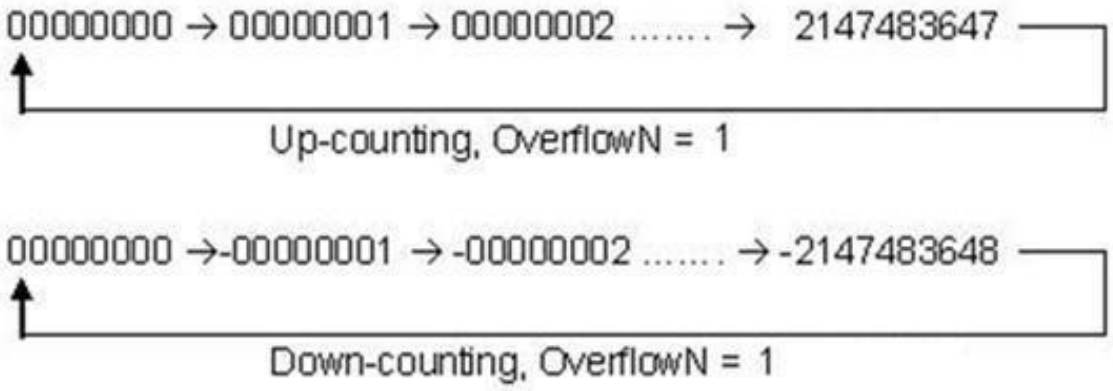
The counter operation for Up/Down mode is as follows:



When InA0 is used as an UP_clock and InB0 is used as a DOWN_clock, counter_0 will be increased by one for every falling edge of InA0 and decreased by one for every falling edge of InB0.

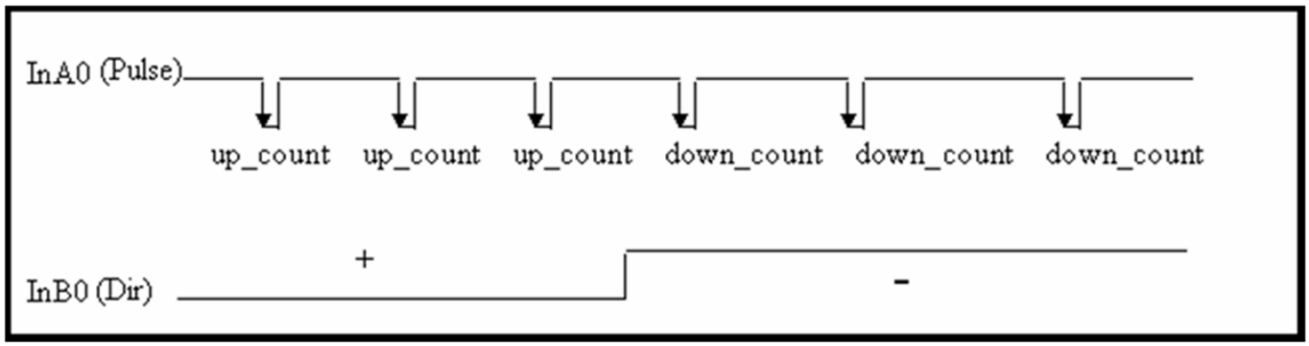
CountN	the current counter value for channel N, 32 bits wide, from -2147483648 to 2147483647
OverflowN	0 = no overflow 1 = overflow

This gives the following:



Dir/PulseCounting

The counter operation for Dir/Pulse mode is as follows:

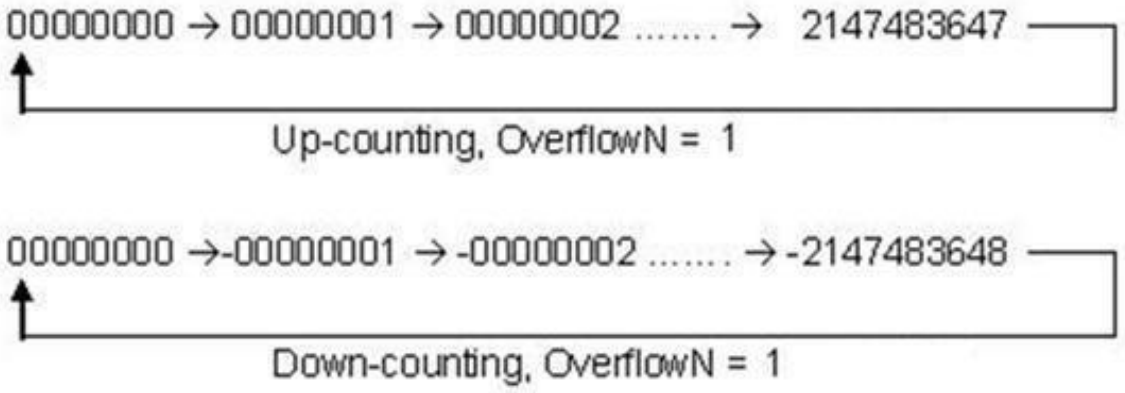


When InB0 is used as Dir:

- If InB0 is High, counter_0 will be increased by one for every falling edge of InA0.
- If InB0 is Low, counter_0 will be decreased by one for every falling edge of InA0.

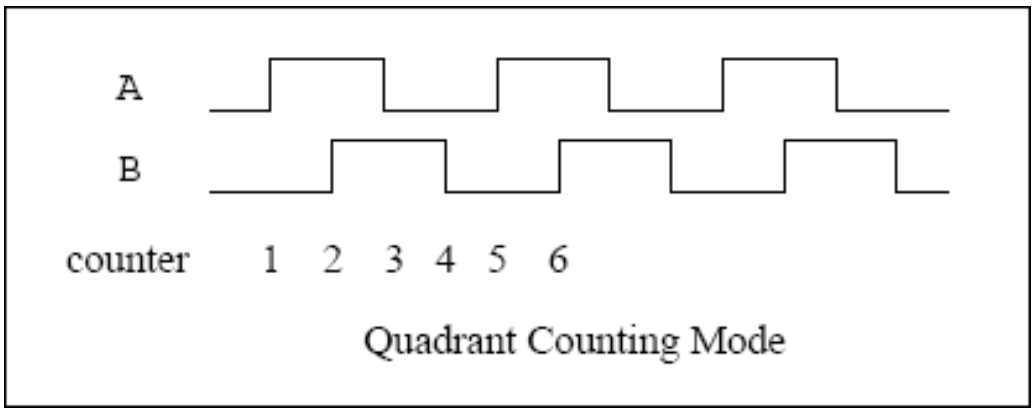
CountN	the current counter value for channel N, 32 bits wide, from -2147483648 to 2147483647
OverflowN	0 = no overflow 1 = overflow

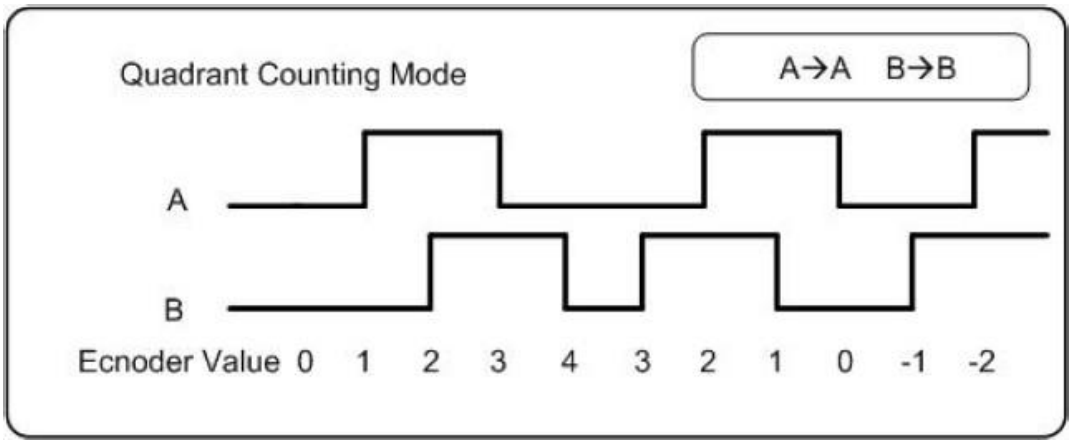
This gives the following:



A/B PfaseCounting

The counter operation for A/B Phase is as follows:





When InA0 is used as an A signal and InB0 is used as a B signal:

- Counter_0 will be increased by one when the InA0 phase leads by 90 degrees to InB0.
- Counter_0 will be decreased by one when the InA0 phase lags by 90 degrees to InB0.

(2) XOR Bit is used to change the status of Z_index.

The status of Z index is 1 and the XOR Bit is disable when shipment, enable XOR bit and the status of Z index will become 0.

*When the status of Z index = 1, the encoder value will be latched.

```
******** ~ SET XOR ~ **********
Press '0' to Set XOR bit = 0
Press '1' to Set XOR bit = 1

ret= 0 Set Channel[0] Xor 1

ret= 0 Set Channel[1] Xor 1

ret= 0 Set Channel[2] Xor 1

****************************************************
```

(3) Preset value is used to set the set the starting position of the count.

```
******** ~ SET PRESET VALUE  ~ **********
Input CH:0 Hex Data Range 0 ~ 0xffffffff
1
Set CH:0 Preset 1
Input CH:1 Hex Data Range 0 ~ 0xffffffff
10
Set CH:1 Preset 10
Input CH:2 Hex Data Range 0 ~ 0xffffffff
100
Set CH:2 Preset 100

****************************************************
```

```
******** ~ READ 32 BIT ENCODER VALUE ~ **********
Press 'Q' or 'q' to Go Back Main Menu.

S[1]C[0] 0000000001 00000001    S[1]C[1] 0000000016 00000010    S[1]C[2] 0000000256 00000100

S[1]C[0] 0000000001 00000001    S[1]C[1] 0000000016 00000010    S[1]C[2] 0000000256 00000100

S[1]C[0] 0000000001 00000001    S[1]C[1] 0000000016 00000010    S[1]C[2] 0000000256 00000100

S[1]C[0] 0000000001 00000001    S[1]C[1] 0000000016 00000010    S[1]C[2] 0000000256 00000100

****************************************************
```

(4) Read Frequency

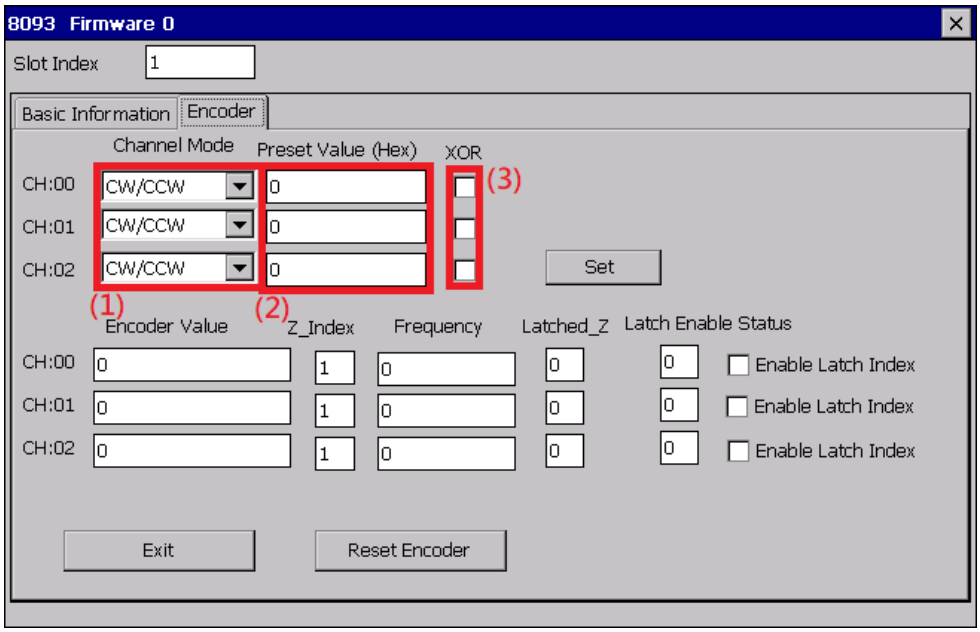
```
******** ~ READ Frequency ~ **********
S[1]C[0] Freq= 0.093    S[1]C[1] Freq= 0.093    S[1]C[2] Freq= 0.093
S[1]C[0] Freq= 0.093    S[1]C[1] Freq= 0.093    S[1]C[2] Freq= 0.093
S[1]C[0] Freq= 0.093    S[1]C[1] Freq= 0.093    S[1]C[2] Freq= 0.093
S[1]C[0] Freq= 0.093    S[1]C[1] Freq= 0.093    S[1]C[2] Freq= 0.093
S[1]C[0] Freq= 0.093    S[1]C[1] Freq= 0.093    S[1]C[2] Freq= 0.093
S[1]C[0] Freq= 0.093    S[1]C[1] Freq= 0.093    S[1]C[2] Freq= 0.093

****************************************************
```

2.2. Getting start on Windows Platform

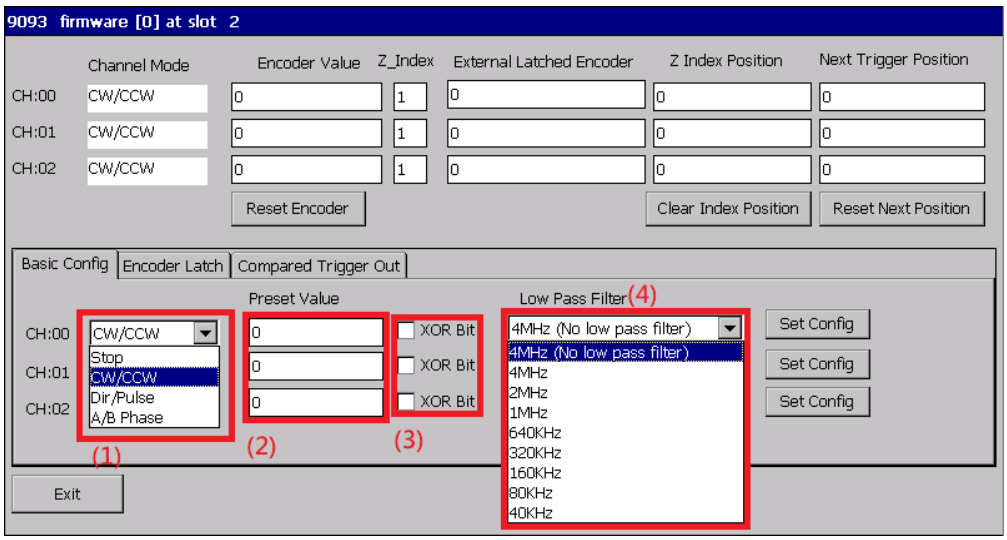
This part will show the functions of I-8093W/I-9093 with Dcon_utility.

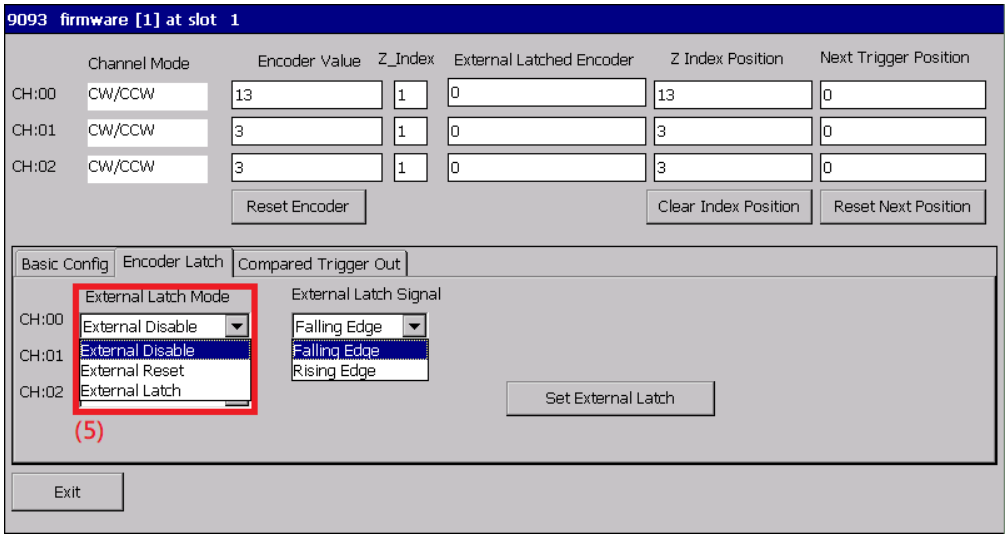
*Dcon_utility support I-9093 with version 2.0.0.9 or above.

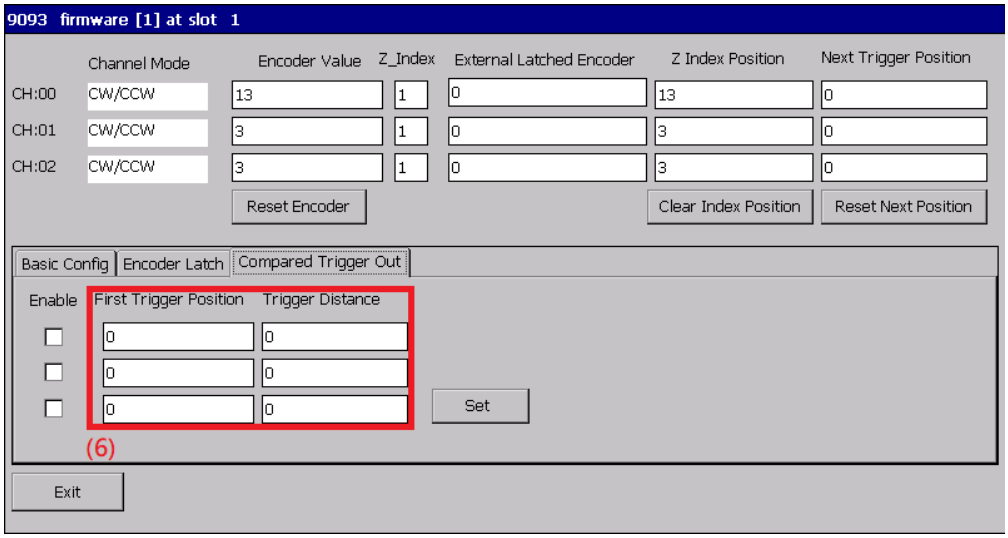
The following picture is the User Interface of I-8093W



The following picture is the User Interface of I-9093.





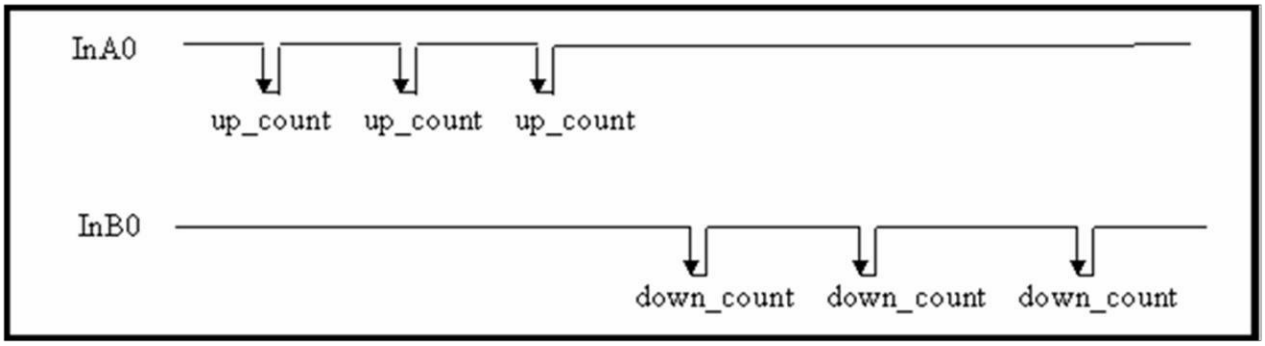


There are 4 options you can choose for each channel.

(1) (STOP , CW/CCW , Dir/Pulse , A/B Phase)

CW/CCWCounting

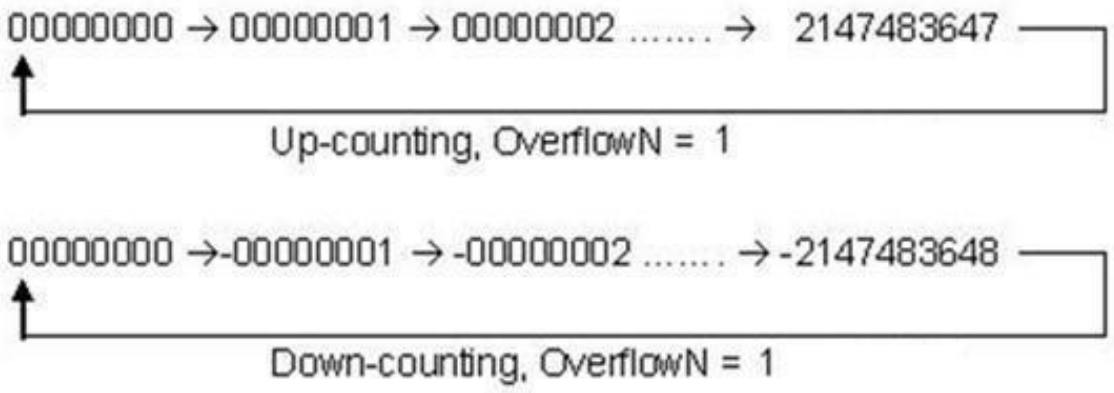
The counter operation for Up/Down mode is as follows:



When InA0 is used as an UP_clock and InB0 is used as a DOWN_clock, counter_0 will be increased by one for every falling edge of InA0 and decreased by one for every falling edge of InB0.

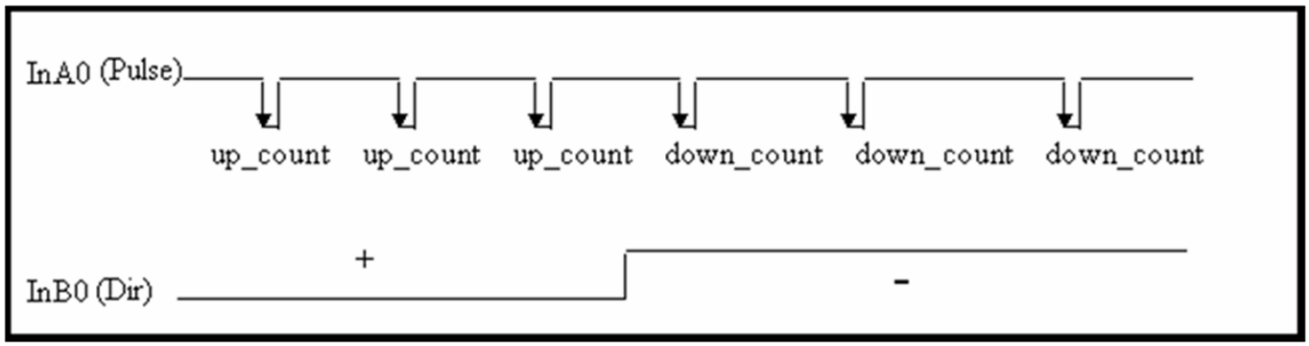
CountN	the current counter value for channel N, 32 bits wide, from -2147483648 to 2147483647
OverflowN	0 = no overflow 1 = overflow

This gives the following:



Dir/PulseCounting

The counter operation for Dir/Pulse mode is as follows:

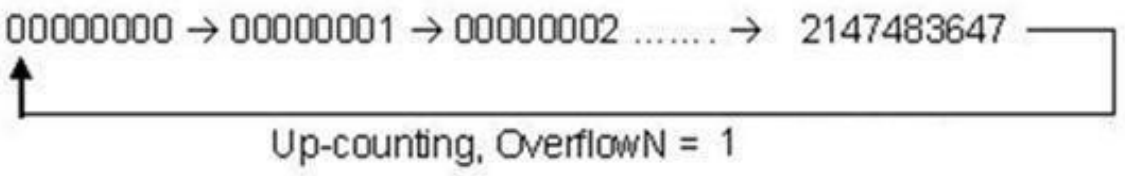


When InB0 is used as Dir:

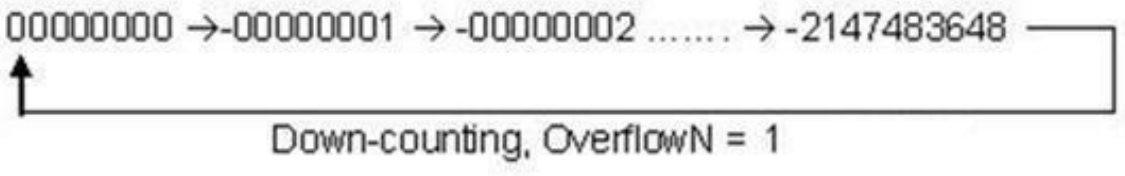
- If InB0 is High, counter_0 will be increased by one for every falling edge of InA0.
- If InB0 is Low, counter_0 will be decreased by one for every falling edge of InA0.

CountN	the current counter value for channel N, 32 bits wide, from -2147483648 to 2147483647
OverflowN	0 = no overflow 1 = overflow

This gives the following:



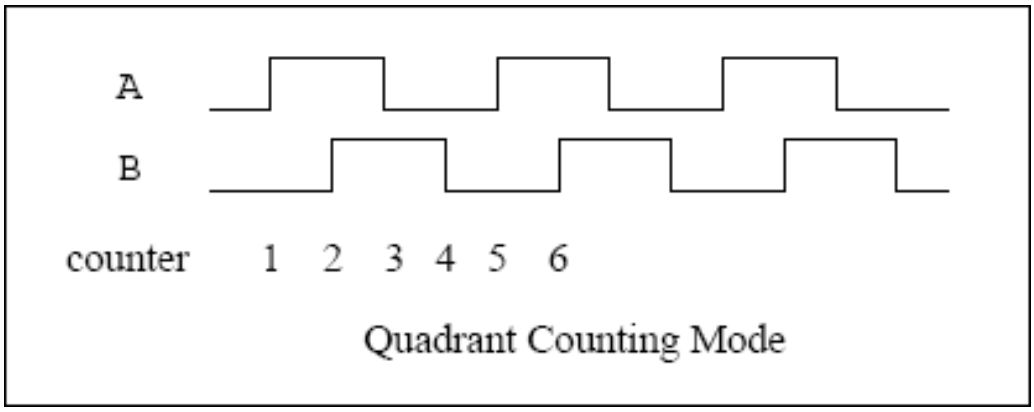
Up-counting, OverflowN = 1

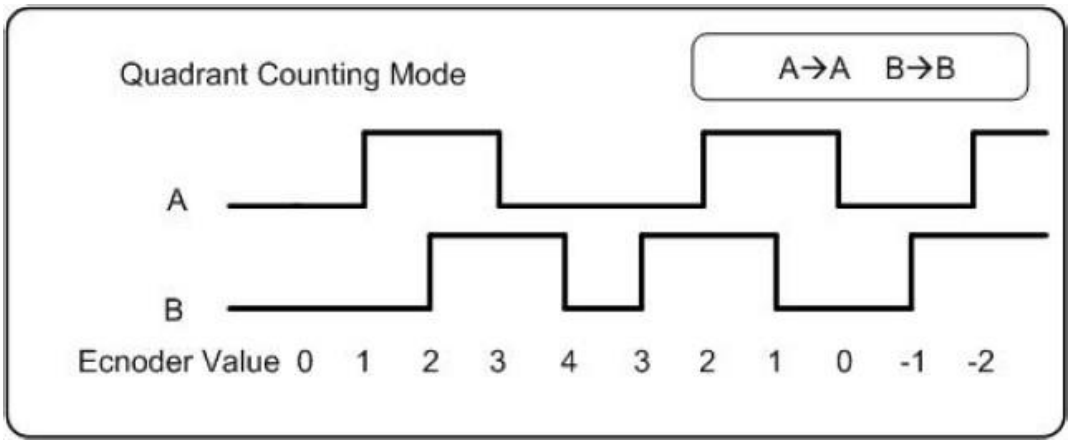


Down-counting, OverflowN = 1

A/B PfaseCounting

The counter operation for A/B Phase is as follows:

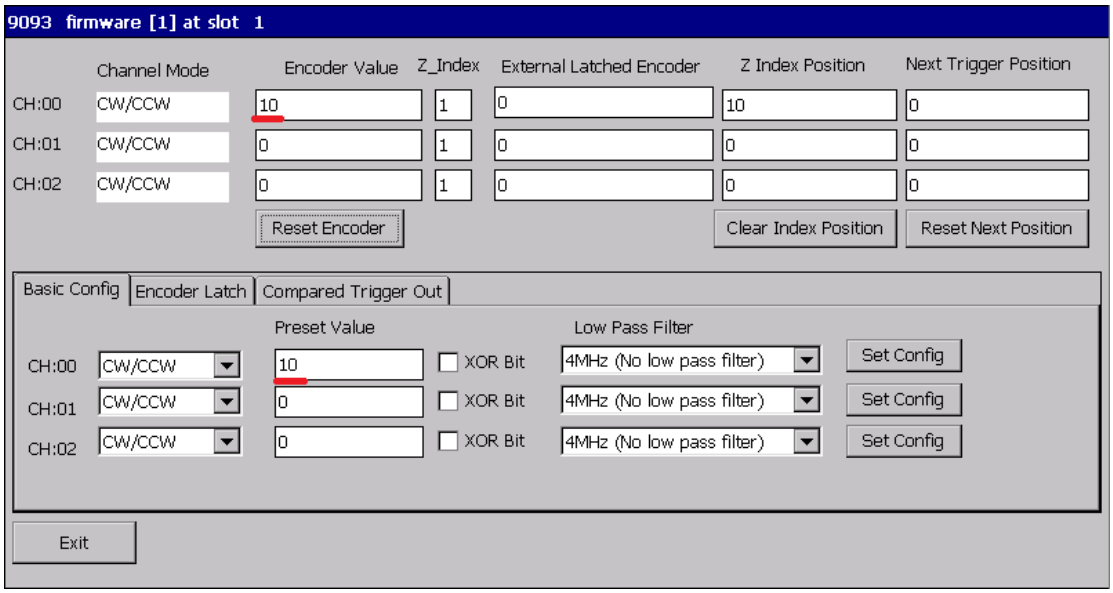




When InA0 is used as an A signal and InB0 is used as a B signal:

- Counter_0 will be increased by one when the InA0 phase leads by 90 degrees to InB0.
- Counter_0 will be decreased by one when the InA0 phase lags by 90 degrees to InB0.

(2) Preset value is used to set the set the starting position of the count.

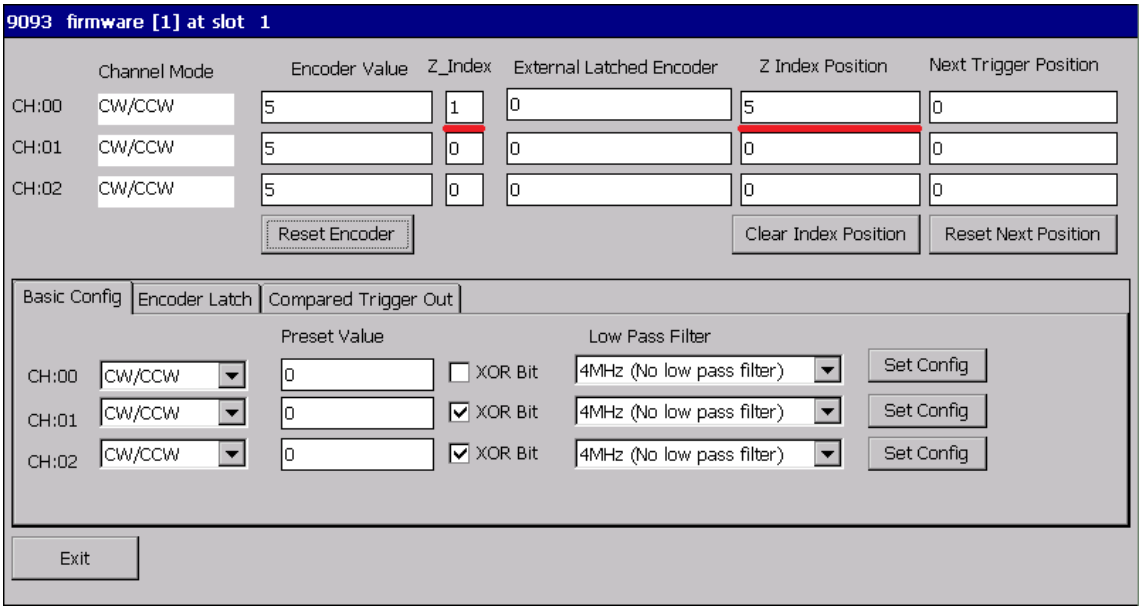


(3) XOR Bit is used to change the status of Z_index.

The status of Z index is 1 and the XOR Bit is disable when shipment,

enable XOR bit and the status of Z index will become 0.

*When the status of Z index = 1, the encoder value will be latched.



(4) Low Pass Filter.

Digital Low Pass Filter

The I-9093 includes digital noise filters that can be used to remove noise, and are implemented as follows:

- The Low Pass Filter can be set to either enabled or disabled. The width of the Low Pass Filter is programmable and can be set with 9 different types.
- The Low Pass Filter can be applied to all working modes.
- ICP DAS provides API functions that can be used within your custom program to change the settings of the Low Pass Filters.(To know the types ,please refer the 3.33 pac_i8093W_SetLowPassFilter Function.)

Refer to the following table for details of how to set the Low Pass Filter:

H = the HIGH width of the input signal T = the period of the filtering clock	
Case	**Filtering status**
H > 2T	The signal will be PASSED
T < H < 2T	The signal may be FILTERED or PASSED
H < T	The signal will be FILTERED

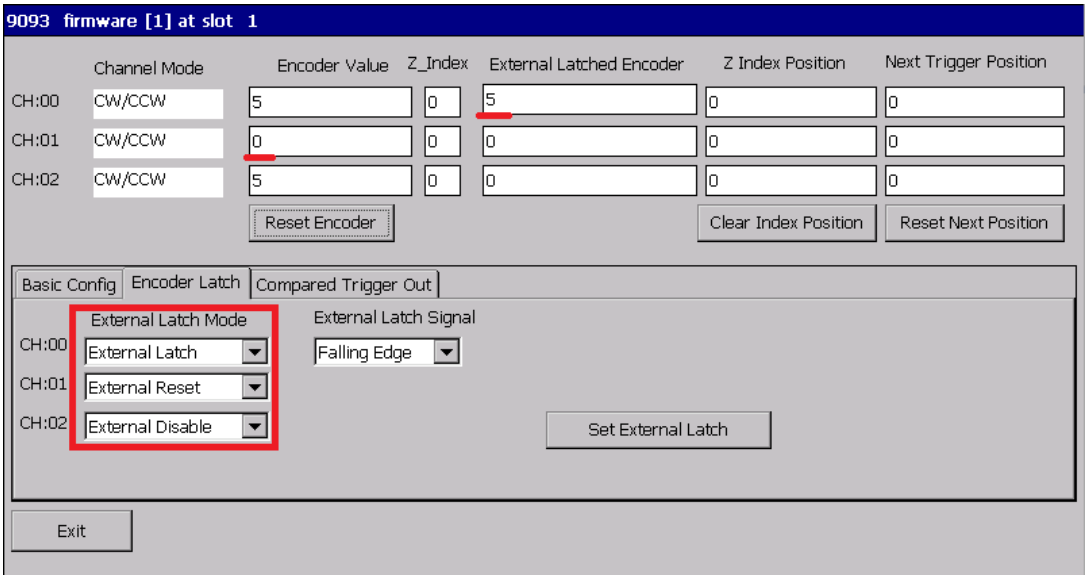
(5) External Trig Mode.

There are three different mode (Disable , Reset , Latch) , when input an external pulse at In+, I-9093 will do the corresponding actions.

Disable: do nothing.

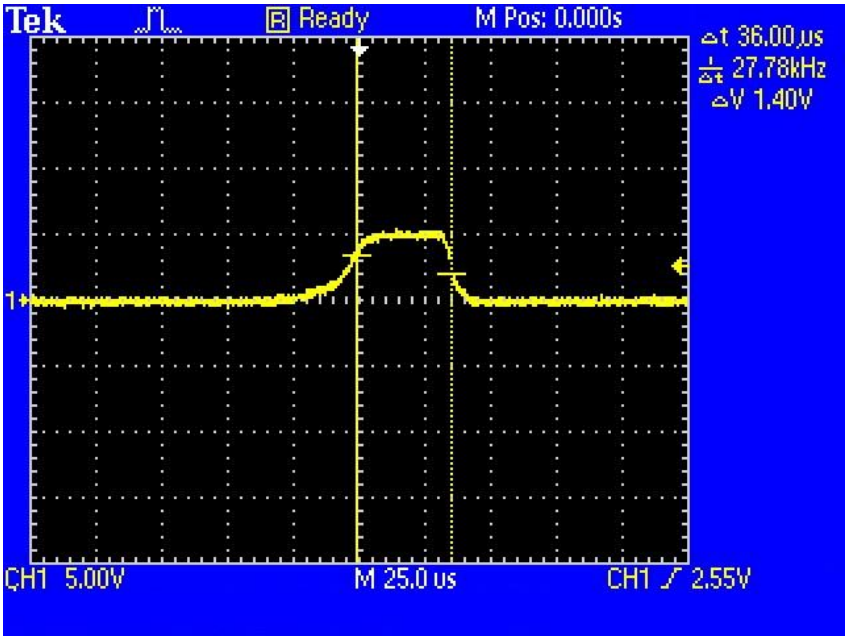
Reset: reset Encoder value

Latch: latch the encoder value

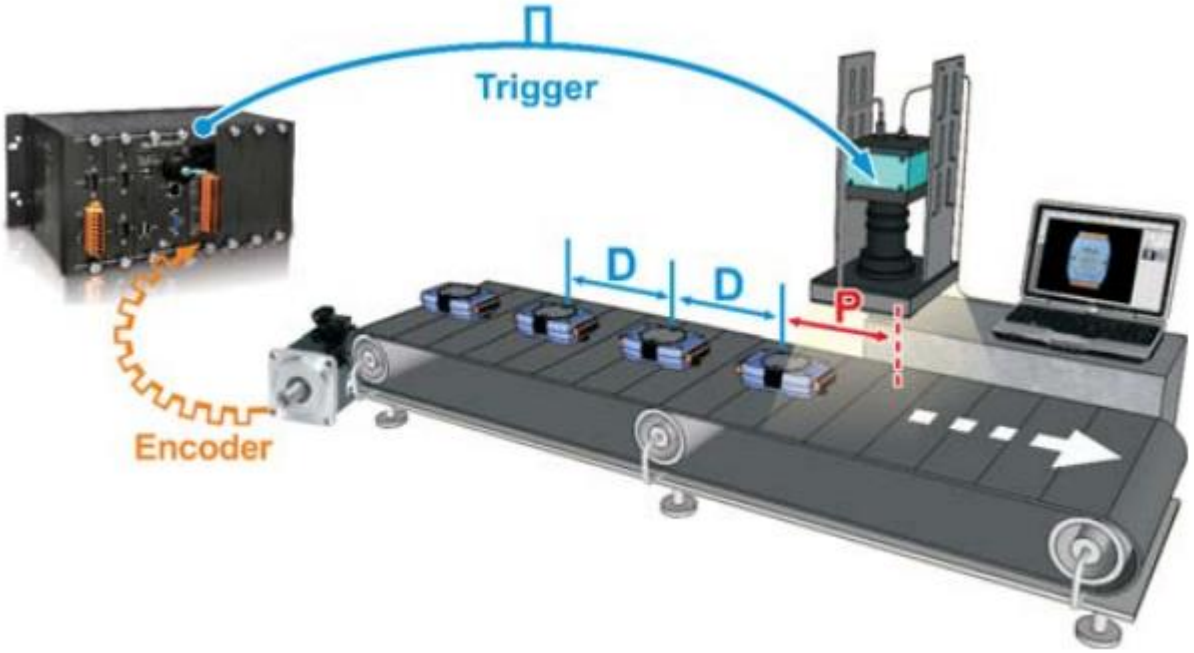


(6) Compared Trig Out

To use compared trig out user has to set first trig position, distance and enable this function. When encoder value is the same as next trig position, trig+ will output a pulse.



3. Compare Trig Out



To use the compare trigger output function, you have to set an initial point (First Trig Position) and a trigger period of the following points (Distance).

The trigger signal is an I/O line that can be used to fire another device.

For example, when a motor reaches a certain position, the trigger signal can be used to fire the shutter of a camera to capture an image for the defect detection.

If the action of device takes time to prepare, user can use pac_i8093W_SetPreTriggerSteps to adjust the delay that caused by the machine.

For example, if first trig position is 5000, and set pac_i8093W_SetPreTriggerSteps is 100, I-9093 will output a pulse when encoder value is 4900.

All operations of position compare and trigger pulse output are automatically done by the hardware circuit.

There is no software calculation effort when the system is operating.

I-9093 makes the system design simpler, and significantly increases the system performance.

4. Demo Programs

ICP DAS provides a range of demo programs for different platforms that can be used to verify the functions of the I-8093W/I-9093. 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 or the enclosed CD.

Platform	Location
I-8093W on I-8000 Platform	
Library	CD: \NAPDOS\8000\841x881x\demo\Lib ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/841x881x/demo/lib/
Demo	CD:NAPDOS\8000\841x881x\demo\IO_in_Slot ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/841x881x/demo/io_in_slot/
I-8093W on iPAC-8000 Platform	
Library	CD: \NAPDOS\iPAC8000\Demo\Basic\iP-84x1_iP-88x1\Lib ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/ip-84x1_ip-88x1/lib/
Demo	CD:\ NAPDOS\iPAC8000\Demo\Basic\iP-84x1_iP-88x1\IO_in_Slot ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/ipac8000/demo/basic/ip-84x1_ip-88x1/io_in_slot/
I-8093W on Windows CE5 Platform	
Library	CD: \napdos\wp-8x4x_ce50\SDK\IO_Modules ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/io_modules/
Demo	VC2005 Demo: CD: \napdos\wp-8x4x_ce50\Demo\WinPAC\VC2005\IO\Local ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/vc2005/io/local/ C# Demo: CD:\ napdos\wp-8x4x_ce50\Demo\WinPAC\C#\IO\Local ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/demo/winpac/c%23/io/local/

Platform	Location
I-8093W on XP-8000-CE6 Platform	
Library	CD:\SDK\Special_IO ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/special_io/
Demo	VC2005 Demo: CD:\demo\XPAC\VC2005\IO\Local ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/vc2005/io/local/ C# Demo: CD:\demo\XPAC\C#\IO\Local ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/demo/xpac/c%23/io/local/
I-8093W on XP-8000-Atom-CE6 Platform	
Library	CD:\SDK\Special_IO ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/sdk/special_io/
Demo	VC 2005 Demo: CD:\demo\XPAC\VC2005\IO\Local ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/vc2005/io/local/ C# Demo: CD:\demo\XPAC\C#\IO\Local ftp://ftp.icpdas.com/pub/cd/xpac-atom-ce6/demo/xpac/c%23/io/local/
I-8093W on XP-8000 Platform	
Library	CD:\SDK\IO ftp://ftp.icpdas.com/pub/cd/xp-8000/sdk/io/
Demo	VC Demo: CD:\Demo\pacsdk\vc\IO\Local ftp://ftp.icpdas.com/pub/cd/xp-8000/demo/pacsdk/vc/io/local/ C# Demo: CD:\Demo\pacsdk\csharp.net\IO\Local\windows_forms ftp://ftp.icpdas.com/pub/cd/xp-8000/demo/pacsdk/csharp.net/io/local/windows_forms/

Platform	Location
I-8093W on XP-Atom Platform	
Library	CD:\SDK\IO ftp://ftp.icpdas.com/pub/cd/xpac-atom/sdk/io/
Demo	VC Demo: CD:\Demo\pacsdk\vc\IO\Local ftp://ftp.icpdas.com/pub/cd/xpac-atom/demo/pacsdk/vc/io/local/ C# Demo: CD:\Demo\pacsdk\csharp.net\IO\Local\windows_forms ftp://ftp.icpdas.com/pub/cd/xpac-atom/demo/pacsdk/csharp.net/io/local/windows_forms/
I-9093 on WP-9000 Platform	
Library	CD:\WinPAC_AM335x\wp-9000\SDK\IO_Modules ftp://ftp.icpdas.com/pub/cd/winpac_am335x/wp-9000/sdk/io_modules/
Demo	VC2008 Demo: CD:\WinPAC_AM335x\wp-9000\demo\PAC\Vc2008\IO\Local ftp://ftp.icpdas.com/pub/cd/winpac_am335x/wp-9000/demo/pac/vc2008/io/local/ C# Demo: CD:\WinPAC_AM335x\wp-9000\demo\PAC\C#\IO\Local ftp://ftp.icpdas.com/pub/cd/winpac_am335x/wp-9000/demo/pac/c%23/io/local/
I-9093 on IPPC-WES7 Platform	
Library	CD:\ippc-wes7\sdk\IO ftp://ftp.icpdas.com/pub/cd/ippc-wes7/sdk/io/
Demo	VC Demo: CD:\ippc-wes7\demo\pacsdk\vc\io\local\io-9k ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/vc/io/local/io-9k/ C# Demo: CD:\ippc-wes7\demo\pacsdk\csharp.net\io\local\io-9k ftp://ftp.icpdas.com/pub/cd/ippc-wes7/demo/pacsdk/csharp.net/io/local/io-9k/

5. API References

ICPDAS supplies a range of C/C++ API functions for the I-8093W/I-9093 module. When developing a custom program, refer to either the 8093W.h 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 custom .NET programs. These functions are ported from the relevant C/C++ functions. For more information related to the .NET functions, refer to the pac_i8093W.h file.

More details of where to find the relevant libraries and files, refer to Chapter 1.7 Location of the Demo and Library Programs.

API naming table

The following table describes the platforms and in which the product series included and the different part of function name.

Platform	Product included	API prefix characters
Windows CE5 Windows CE6	WP-8000 series XP-8000-CE6 series XP-8000-Atom-CE6 series	"pac_i8093W_ "+ function name
Windows CE7	WP-8000 series WP-9000-CE7 series	"pac_i8093W_ "+ function name
WES	XP-8000 series XP-8000-Atom series	"pac_i8093W_ "+ function name
WES7	XP-8000-WES7 series XP-9000 series	"pac_i8093W_ "+ function name
MiniOS7	I-8000 series iPAC-8000 series	"i8093W_" + function name
Linux	LinPAC-8000 series LinPAC-9000 series	"i8093W_" + function name

The following is an overview of the functions provided in the 8093W.lib/pac_i8093W.lib for use with the MiniOS7 / Windows (CE and WES) platform.

API for both I-8093W and I-9093

Function for MiniOS7	Function for Windows	Description
i8093W_Init	pac_i8093W_Init	This function can initial the I-8093W/I-9093 and can check the hardware ID.
i8093W_GetFirmwareVersion	pac_i8093W_GetFirmwareVersion	This function use to get the firmware version of I-8093W/I-9093 hardware.
i8093W_GetLibVersion	pac_i8093W_GetLibVersion	This function use to get the library version of 8093W.lib/i8093W.dll.
i8093W_GetLibDate	pac_i8093W_GetLibDate	This function use to get the library built date of 8093W.lib/i8093W.dll.
i8093W_SetMode	pac_i8093W_SetMode	This function use to set the operation mode of I-8093W/I-9093.
i8093W_GetMode	pac_i8093W_GetMode	This function use to get the operation mode of I-8093W/I-9093.
i8093W_SetXOR	pac_i8093W_SetXOR	This function use to set the xor of I-8093W/I-9093 for each channel.
i8093W_GetXOR	pac_i8093W_GetXOR	This function use to get the xor of I-8093W/I-9093 for each channel.
i8093W_GetLineStatus	pac_i8093W_GetLineStatus	This function use to get A,B and Z status of I-8093W/I-9093.
i8093W_GetIndex	pac_i8093W_GetIndex	This function use to get Z index status of I-8093W/I-9093.
i8093W_Read32BitEncoder	pac_i8093W_Read32BitEncoder	This function use to get 32-Bit Encoder value of I-8093W/I-9093.
i8093W_ResetEncoder	pac_i8093W_ResetEncoder	This function use to reset 32-Bit Encoder value to zero.
i8093W_SetPresetValue	pac_i8093W_SetPresetValue	This function use to set 32-Bit preset value of I-8093W/I-9093.
i8093W_GetPresetValue	pac_i8093W_GetPresetValue	This function use to get 32-Bit preset value of I-8093W/I-9093.
i8093W_ClearLatchedIndex	pac_i8093W_ClearLatchedIndex	This function use to clear the index latched status.

API for I-8093W

Function for MiniOS7	Function for Windows	Description
i8093W_ReadFreq	pac_i8093W_ReadFreq (For I-8093W only)	This function use to read frequency value of encoder input signal.
i8093W_SetIndexLatchStatus	pac_i8093W_SetIndexLatchStatus (For I-8093W only)	This function use to enable/disable the index latch function of I-8093W.
i8093W_GetIndexLatchStatus	pac_i8093W_GetIndexLatchStatus (For I-8093W only)	This function use to get the index latch function status of I-8093W.

API for I-9093

Function for MiniOS7	Function for Windows	Description
N/A	pac_i8093W_SetExTrigMode (For I-9093 only)	This function use to set External latch mode of I-9093.
N/A	pac_i8093W_GetExTrigMode (For I-9093 only)	This function use to get External latch mode of I-9093.
N/A	pac_i8093W_SetExSignal (For I-9093 only)	This function use to set External latch signal of I-9093.
N/A	pac_i8093W_GetExSignal (For I-9093 only)	This function use to get External latch signal of I-9093.
N/A	pac_i8093W_SetPreTriggerSteps (For I-9093 only)	This function use to set steps to adjust the delay that caused by the machine.
N/A	pac_i8093W_GetPreTriggerSteps (For I-9093 only)	This function use to get the steps.
N/A	pac_i8093W_SetFirstTrigPosition (For I-9093 only)	This function use to set first trig position of I-9093.
N/A	pac_i8093W_GetFirstTrigPosition (For I-9093 only)	This function use to get first trig position of I-9093.
N/A	pac_i8093W_SetTrigDistance (For I-9093 only)	This function use to set distance of I-9093.
N/A	pac_i8093W_GetTrigDistance (For I-9093 only)	This function use to get distance of I-9093.
N/A	pac_i8093W_ReadNextPosition (For I-9093 only)	This function use to get next trig position of I-9093.
N/A	pac_i8093W_ClearNextPosition (For I-9093 only)	This function use to clear next trig position of I-9093.

N/A	pac_i8093W_ConfigTriggerOut (For I-9093 only)	This function use to enable/disable the compared trig out function of I-9093.
N/A	pac_i8093W_GetTriggerOutConfig (For I-9093 only)	This function use to get the compared trig out function status of I-9093.
N/A	pac_i8093W_SetLowPassFilter (For I-9093 only)	This function use to set the low pass filter of I-9093.
N/A	pac_i8093W_GetLowPassFilter (For I-9093 only)	This function use to get the Low Pass Filter of I-9093.
N/A	pac_i8093W_ReadIndexLatchedPosition (For I-9093 only)	This function use to get the index latched position of I-9093.
N/A	pac_i8093W_ReadExTrigLatchedPosition (For I-9093 only)	This function use to get the external trig latched position of I-9093.

5.1.i8093W_Init

This function can initial the I-8093W/I-9093 and can check the hardware ID.

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
int pac_i8093W_Init(
    int slot
);
```

Parameter

slot:

specifies the slot number (0 - 7).

Return Values

0 = the module in the slot is an I-8093W/I-9093.

-1 = there is no I-8093W/I-9093 module in this slot.

For other return values, please refer the Error Code.

Note

Before executing any functions on the I-8093W/I-9093, the i8093W_Init function needs to be called once for I-8093W/I-9093. If there are two or more I-8093W/I-9093 modules, you need call the i8093W_Init function for each I-8093W/I-9093 module individually by passing the slot number that the I-8093W/I-9093 module is plugged into.

Example

[C/C++]

```
Int slot;
i8014W_Init(slot);
```

[C#]

```
Int slot;
pac8093WNet.pac8093W.Init(slot);
```

5.2.i8093W_GetFirmwareVersion

This function use to get the firmware version of I-8093W hardware.

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
short pac_i8093W_GetFirmwareVersion(int slot);
```

Parameter

slot:

specifies the slot number (0 - 7).

Return Values

The version number of the primary FPGA firmware for the I-8093W/I-9093 module.

Example

[C/C++]

```
short ver, slot;
ver= i8093W_GetFirmwareVersion(slot);
```

[C#]

```
Int slot, ver;
ver=pac8093WNet.pac8093W.FirmwareVersion(slot);
```

5.3.i8093W_GetLibVersion

This function use to get the library version of 8093W.lib/i8093W.dll.

Syntax

For MiniOS7

```
short i8093W_GetLibVersion(void);
```

For Windows (CE and WES)

```
short pac_i8093W_GetLibVersion(void);
```

Parameter

None

Return Values

The version number of the 8093W.lib/i8093W.dll.

Example

[C/C++]

```
short ver, slot;
ver = i8093W_GetLibVersion(void);
```

[C#]

```
short ver, slot;
ver = pac8093WNet.pac8093W.LibVersion();
```

5.4.i8093W_GetLibDate

This function use to get the library built date of 8093W.lib/i8093W.dll.

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
void pac_i8093W_GetLibDate(char *LibDate);
```

Parameter

**libDate:*

[output] the release date of the 8093W.lib/i8093W.dll.

Return Values

None

Example

[C++]

```
short date;
i8093W_GetLibDate(&date);
```

[C#]

```
short date;
date =pac8093WNet.pac8093W.LibDate();
```

5.5.i8093W_SetMode

This function use to set the operation mode of I-8093W/I-9093.

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
int pac_i8093W_SetMode(
    int slot,
    int ch,
    int Mode
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Mode:

set encoder counting mode

1 ==> CW/CCW counting mode

2 ==> Pulse/Direction counting mode

3 ==> Quadrant counting mode

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch , mode;
i8093W_SetMode(lot , ch , mode);
```

[C#]

```
Int slot , ch , mode;
pac8093WNet.pac8093W.SetChannelMode(slot , ch , mode);
```

5.6.i8093W_GetMode

This function use to get the operation mode of I-8093W/I-9093.

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
int pac_i8093W_GetMode(
    int slot,
    int ch,
    int* Mode
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**Mode:*

[output] get encoder counting mode

1 ==> CW/CCW counting mode

2 ==> Pulse/Direction counting mode

3 ==> Quadrant counting mode

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch , mode;
i8093W_GetMode (lot , ch , *mode);
```

[C#]

```
Int slot , ch , mode;
pac8093WNet.pac8093W.ReadChannelMode(slot , ch , ref mode);
```

5.7.i8093W_SetXOR

This function use to set the xor of I-8093W/I-9093 for each channel.

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
int pac_i8093W_SetXOR(
    int slot,
    int ch,
    int Xor
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Xor:

to change the status of Z_index

0 : not activated

1 : activated

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch , xor;
i8093W_SetXOR(lot , ch , xor);
```

[C#]

```
Int slot , ch , xor;
pac8093WNet.pac8093W.SetXor(slot , ch , xor);
```

5.8.i8093W_GetXOR

This function use to get the xor of I-8093W/I-9093 for each channel.

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
int pac_i8093W_GetXOR(
    int slot,
    int ch,
    int* Xor
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**Xor:*

[output] to change the status of Z_index

0 : not activated

1 : activated

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch , xor;
pac_i8093W_GetXOR (lot , ch , &xor);
```

[C#]

```
Int slot , ch , xor;
pac8093WNet.pac8093W.ReadXor(slot , ch , ref xor);
```

5.9.i8093W_GetLineStatus

This function use to get A,B and Z status of I-8093W/I-9093.

Syntax

For MiniOS7

```
int i8093W_GetLineStatus(
    int slot,
    int ch,
    int* A_Status,
    int* B_Status,
    int* C_Status
);
```

For Windows (CE and WES)

```
int pac_i8093W_GetLineStatus(
    int slot,
    int ch,
    int* A_Status,
    int* B_Status,
    int* C_Status
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

[Output] A_Status:*

[output]

0: not activated

1: activated

**B_Status:*

[output]

0: not activated

1: activated

**C_Status:*

[output]

0: not activated

1: activated

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch , A_Status , B_Status , C_Status;
i8093W_GetLineStatus(slot , ch , & A_Status , & B_Status , & C_Status);
```

[C#]

```
Int slot , ch , A_Status , B_Status , C_Status;
pac8093WNet.pac8093W.ReadLineStatus(slot, ch, ref A_Status, ref B_Status, ref C_Status)
```

5.10. i8093W_GetIndex

This function use to get Z index status of I-8093W/I-9093.

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
int pac_i8093W_GetIndex(
    int slot,
    int ch,
    int* index
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

** index:*

[output]

0: not activated

1: activated

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch , index;
pac_i8093W_GetIndex(slot , ch , &index);
```

[C#]

```
Int slot , ch , index;
pac8093WNet.pac8093W.ReadIndex(slot , ch , ref index);
```

5.11. i8093W_Read32BitEncoder

This function use to get 32-Bit Encoder value of I-8093W/I-9093.

Syntax

For MiniOS7

```
int i8093W_Read32BitEncoder( int slot , int ch , long* EnCode32);
```

For Windows (CE and WES)

```
int pac_i8093W_Read32BitEncoder(int slot , int ch , long* EnCode32);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

** EnCode32:*

[output] get encoder value.

Return Values

return Index enable/disable and Index latched status of I-8093W

0: Disable Index Latch (0), Index Not Latched (0)

1: Enable Index Latch (1), Index Not Latched (0)

2: Disable Index Latch (0), Index Latched (1)

3: Enable Index Latch (1), Index Latched (1)

For I-9093 please refer the Error Code.

[C/C++]

```
Int slot , ch ;
long EnCode;
i8093W_Read32BitEncoder(slot , ch , &EnCode);
```

[C#]

```
Int slot , ch ;
long EnCode;
pac8093WNet.pac8093W.ReadEncoder(slot, ch, ref EnCode);
```

5.12. i8093W_ResetEncoder

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

Syntax

For MiniOS7

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

For Windows (CE and WES)

```
int pac_i8093W_ResetEncoder(
    int slot,
    int ch
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
i8093W_Read32BitEncoder(slot , ch , &EnCode);
```

[C#]

```
Int slot , ch ;
pac8093WNet.pac8093W.ReadEncoder(slot, ch, ref EnCode);
```

5.13. i8093W_SetPresetValue

This function use to set 32-Bit preset value of I-8093W/I-9093.

Syntax

For MiniOS7

```
int i8093W_SetPresetValue(
    int slot,
    int ch,
    long presetVal
);
```

For Windows (CE and WES)

```
int    pac_i8093W_SetPresetValue(
    int slot,
    int ch,
    long presetVal
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

presetVal:

Set preset value.

Return Values

Please refer the Error Code.

Note

Preset value is supported at lib version 0x1007 or above

Example

[C/C++]

```
Int slot , ch ;
long presetVal;
i8093W_SetPresetValue(int slot, int ch, long presetVal);
```

[C#]

```
Int slot , ch ;
long presetVal;
pac8093WNet.pac8093W.SetPreset(slot, ch, presetVal);
```

5.14. i8093W_GetPresetValue

This function use to get 32-Bit preset value of I-8093W/I-9093.

Syntax

For MiniOS7

```
int i8093W_GetPresetValue(
    int slot,
    int ch,
    long* presetVal
);
```

For Windows (CE and WES)

```
int pac_i8093W_GetPresetValue(
    int slot,
    int ch,
    long* presetVal
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

presetVal:

[output] get preset value.

Return Values

Please refer the Error Code.

Note

Preset value is supported at lib version 0x1007 or above

Example

[C/C++]

```
Int slot , ch ;
long presetVal;
i8093W_GetPresetValue(slot , ch , &presetVal);
```

[C#]

```
Int slot , ch ;
long presetVal;
pac8093WNet.pac8093W.GetPreset(slot, ch, ref presetVal);
```

5.15. i8093W_ReadFreq

This function use to read frequency value of encoder input signal.

(For I-8093W only)

Syntax

For MiniOS7

```
int i8093W_ReadFreq(
    int slot,
    int ch,
    float* freq
);
```

For Windows (CE and WES)

```
int pac_i8093W_ReadFreq(
    int slot,
    int ch,
    float* freq
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

** freq:*

[output] get frequency value.

Return Values

Please refer the Error Code.

Note

1: read frequency function supported by firmware version 3 or later version.

2: if there is no encoder input, the frequency will be 0.093 not 0

Example

[C/C++]

```
Int slot , ch ;
float freq;
i8093W_ReadFreq(slot , ch , &freq);
```

[C#]

```
Int slot , ch ;
float freq;
pac8093WNet.pac8093W.ReadFreq(slot, ch, ref freq);
```

5.16. i8093W_SetIndexLatchStatus

This function use to enable/disable the index latch function of I-8093W.

(For I-8093W only)

Syntax

For MiniOS7

```
int i8093W_SetIndexLatchStatus(
    int slot,
    int ch,
    int ifEnableLatch
);
```

For Windows (CE and WES)

```
int pac_i8093W_SetIndexLatchStatus(
    int slot,
    int ch,
    int ifEnableLatch
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

ifEnableLatch:

to enable or disable the index latch function of i-8093.

Return Values

Please refer the Error Code.

Note

1. normally we can use i8093W_GetIndex to read the Z_index status, but sometimes it is not easy to let the encoder to match the right position, so 8093W add this function to latch the Z_index status
2. if enable the latch index function and the Z_index latched, the encoder value will also be latched at the position where Z_index latched point, so we can adjust the encoder position to the latched point.
3. disable the index latch function, i8093W_Read32BitEncoder can read the normal encoder value as usual.

Example

[C/C++]

```
Int slot , ch ;
int ifEnableLatch;
i8093W_SetIndexLatchStatus(slot, ch , ifEnableLatch);
```

[C#]

```
Int slot , ch ;
int ifEnableLatch;
pac8093WNet.pac8093W.SetIndexLatchStatus(slot, ch, ifEnableLatch);
```

5.17. i8093W_GetIndexLatchStatus

This function use to get the index latch function status of I-8093W.

(For i-8093W only)

Syntax

For MiniOS7

```
int i8093W_GetIndexLatchStatus(
    int slot,
    int ch,
    int* latchedStatus,
    int* ifEnableLatch
);
```

For Windows (CE and WES)

```
int pac_i8093W_GetIndexLatchStatus(
    int slot,
    int ch,
    int* latchedStatus,
    int* ifEnableLatch
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

** latchedStatus:*

[output] check z index latched or not.

**ifEnableLatch:*

[output] read enable latched of i-8093.

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
int latchedStatus ,ifEnableLatch;
8093W_GetIndexLatchStatus(slot, ch, &latchedStatus, &ifEnableLatch);
```

[C#]

```
Int slot , ch ;
int latchedStatus ,ifEnableLatch;
pac8093WNet.pac8093W.GetIndexLatchStatus(slot, ch, ref latchedStatus, ref ifEnableLatch);
```

5.18. i8093W_ClearLatchedIndex

This function use to clear the index latched status.

Syntax

For MiniOS7

```
int i8093W_ClearLatchedIndex(
    int slot,
    int ch
);
```

For Windows (CE and WES)

```
int pac_i8093W_ClearLatchedIndex(
    int slot,
    int ch
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
i8093W_ClearLatchedIndex(slot, ch);
```

[C#]

```
Int slot , ch ;
pac8093WNet.pac8093W.ClearLatchedIndex(slot, ch);
```

5.19. pac_i8093W_SetExTrigMode

This function use to set External latch mode of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_SetExTrigMode(
    int slot,
    int ch,
    unsigned char Mode
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Mode:

0 : disable external latched mode

1 : reset encoder

2 : external latch

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
unsigned char mode;
pac_i8093W_SetExTrigMode(slot, ch, mode);
```

[C#]

```
Int slot , ch ;
Byte mode;
pac8093WNet.pac8093W.SetExTrigMode(slot, ch, mode);
```

5.20. pac_i8093W_GetExTrigMode

This function use to get External latch mode of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_GetExTrigMode(
    int slot,
    int ch,
    unsigned char *Mode
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**Mode:*

[output] get status of external latched mode

0 : disable external latched mode

1 : reset encoder

2 : external latch

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
unsigned char mode;
pac_i8093W_GetExTrigMode (slot, ch, &mode);
```

[C#]

```
int slot , ch ;
int16 mode;
pac8093WNet.pac8093W.GetExTrigMode(slot, ch, ref mode);
```

5.21. pac_i8093W_SetExSignal

This function use to set External latch signal of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_SetExSignal(
    int slot,
    unsigned char Edge
);
```

Parameter

slot:

specifies the slot number (0 - 7).

Edge:

0 : Falling edge

1 : rising edge

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot ;
unsigned char edge;
pac_i8093W_SetExSignal (slot, edge);
```

[C#]

```
Int slot ;
Uint16 edge;
pac8093WNet.pac8093W.SetExSignal(slot, edge);
```

5.22. pac_i8093W_GetExSignal

This function use to get External latch signal of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_GetExSignal(
    int slot,
    unsigned char *Edge
);
```

Parameter

slot:

specifies the slot number (0 - 7).

**Edge:*

[output]

0 : Falling edge

1 : rising edge

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot ;
unsigned char edge;
pac_i8093W_GetExSignal (slot, &edge);
```

[C#]

```
Int slot ;
Uint16 edge;
pac8093WNet.pac8093W.GetExSignal(slot, ref edge);
```

5.23. pac_i8093W_SetPreTriggerSteps

This function use to set steps to adjust the delay that caused by the machine.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_SetPreTriggerSteps(
    int slot,
    int ch,
    unsigned char Steps
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Steps:

Set the steps 0~255.

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
unsigned char step;
pac_i8093W_SetPreTriggerSteps (slot, ch, step);
```

[C#]

```
Int slot , ch ;
Uint16 step;
pac8093WNet.pac8093W.SetPreTriggerSteps(slot, ch, step);
```

5.24. pac_i8093W_GetPreTriggerSteps

This function use to get the steps.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_GetPreTriggerSteps(
    int slot,
    int ch,
    unsigned char *Steps
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**Steps:*

[output] get steps 0~255.

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
unsigned char step;
pac_i8093W_GetPreTriggerSteps (slot, ch, &step);
```

[C#]

```
Int slot , ch ;
int16 step;
pac8093WNet.pac8093W.GetPreTriggerSteps(slot , ch , ref step);
```

5.25. pac_i8093W_SetFirstTrigPosition

This function use to set first trig position of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_SetFirstTrigPosition(
    int slot,
    int ch,
    unsigned long Position
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Position:

Set first trig position

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
unsigned long position;
pac_i8093W_SetFirstTrigPosition(slot , ch , position);
```

[C#]

```
Int slot , ch ;
Uint32 position;
pac8093WNet.pac8093W.SetFirstTrigPosition(slot, ch, position);
```

5.26. pac_i8093W_GetFirstTrigPosition

This function use to get first trig position of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_GetFirstTrigPosition(
    int slot,
    int ch,
    unsigned long* Position
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**Position:*

[output] get first trig position

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
unsigned long position;
pac_i8093W_GetFirstTrigPosition (slot , ch , &position);
```

[C#]

```
Int slot , ch ;
Uint32 position;
pac8093WNet.pac8093W.GetFirstTrigPosition(slot, ch, ref position);
```

5.27. pac_i8093W_SetTrigDistance

This function use to set distance of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_SetTrigDistance(
    int slot,
    int ch,
    unsigned long Distance
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Distance:

set Distance

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
unsigned long distance;
pac_i8093W_SetTrigDistance (slot , ch , distance);
```

[C#]

```
Int slot , ch ;
Uint32 distance;
pac8093WNet.pac8093W.SetTrigDistance(slot, ch, distance);
```

5.28. pac_i8093W_GetTrigDistance

This function use to get distance of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_GetTrigDistance(
    int slot,
    int ch,
    unsigned long* Distance
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**Distance:*

[output] get Distance

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
Uint32 distance;
pac_i8093W_GetTrigDistance (slot, ch& distance);
```

[C#]

```
Int slot , ch ;
Uint32 distance;
pac8093WNet.pac8093W.GetTrigDistance(slot, ch, ref distance);
```

5.29. pac_i8093W_ReadNextPosition

This function use to get next trig position of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_ReadNextPosition(
    int slot,
    int ch,
    unsigned long* Data32
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

** Data32:*

[output] get next position

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
Uint32 Data32;
pac_i8093W_ReadNextPosition (slot, ch& Data32);
```

[C#]

```
Int slot , ch ;
Uint32 Data32;
pac8093WNet.pac8093W.ReadNextPosition(slot, ch , ref Data32);
```

5.30. pac_i8093W_ClearNextPosition

This function use to clear next trig position of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_ClearNextPosition(
    int slot,
    int ch
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
pac_i8093W_ClearNextPosition (slot, ch);
```

[C#]

```
Int slot , ch ;
pac8093WNet.pac8093W.ClearNextPosition(slot, ch);
```

5.31. pac_i8093W_ConfigTriggerOut

This function use to enable/disable the compared trig out function of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_ConfigTriggerOut(
    int slot,
    int ch,
    unsigned char enStatus
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

enStatus:

set compare trig out status

0 : disable

1 : enable

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ,status;
pac_i8093W_ConfigTriggerOut (slot, ch, status);
```

[C#]

```
Int slot , ch ,status;
pac8093WNet.pac8093W.ConfigTriggerOut(slot, ch, status);
```

5.32. pac_i8093W_GetTriggerOutConfig

This function use to get the compared trig out function status of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_GetTriggerOutConfig(
    int slot,
    int ch,
    unsigned char *enStatus
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**enStatus:*

[output] get compare trig out status

0 : disable

1 : enable

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ,status;
pac_i8093W_GetTriggerOutConfig (slot, ch, &status);
```

[C#]

```
Int slot , ch ,status;
pac8093WNet.pac8093W.GetTriggerOutConfig(slot, ch, ref status);
```

5.33. pac_i8093W_SetLowPassFilter

This function use to set the low pass filter of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int pac_i8093W_SetLowPassFilter(
    int slot,
    int ch,
    int filter
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

filter:

	Max Input frequency	
Value	CW/CCW,DIR/Pulse	A/B Phase
0	4MHz (No low pass filter)	6MHz(No low pass filter)
1	4MHz	1MHz
2	2MHz	500KHz
3	1MHz	250KHz
4	640KHz	160 KHz
5	320KHz	80KHz
6	160KHz	40KHz
7	80KHz	20KHz
8	40KHz	10KHz

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch , filter;
i9093_SetLowPassFilter(slot, ch, filter);
```

[C#]

```
Int slot , ch , filter;
pac8093WNet.pac8093W.SetLowPassFilter(slot, ch, filter);
```

5.34. pac_i8093W_GetLowPassFilter

This function use to get the Low Pass Filter of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int i9093_GetLowPassFilter(
    int slot,
    int ch,
    int* filter
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**filter:*

[output]

	Max Input frequency	
Value	CW/CCW,DIR/Pulse	A/B Phase
0	4MHz (No low pass filter)	6MHz(No low pass filter)
1	4MHz	1MHz
2	2MHz	500KHz
3	1MHz	250KHz
4	640KHz	160 KHz
5	320KHz	80KHz
6	160KHz	40KHz
7	80KHz	20KHz
8	40KHz	10KHz

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch , filter;
i9093_GetLowPassFilter (slot, ch, &filter);
```

[C#]

```
Int slot , ch , filter;
pac8093WNet.pac8093W.GetLowPassFilter(slot, ch, ref filter);
```

5.35. pac_i8093W_ReadIndexLatchedPosition

This function use to get the index latched position of I-9093.

With this function user can find the origin of the motor easier.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int i9093_ReadIndexLatchedPosition(
    int slot,
    int ch,
    long* Data32
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**Data32:*

[output] get the index latched position.

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
long Data32;
int i9093_ReadIndexLatchedPosition(slot, ch,*Data32);
```

[C#]

```
Int slot , ch ;
Int32 Data32;
pac8093WNet.pac8093W.ReadIndexLatchedPosition(slot, ch, ref Data32);
```

5.36. pac_i8093W_ReadExTrigLatchedPosition

This function use to get the external trig latched position of I-9093.

(for I-9093 only)

Syntax

For Windows (CE and WES)

```
int i9093_ReadExTrigLatchedPosition(
    int slot,
    int ch,
    long* Data32
);
```

Parameter

slot:

specifies the slot number (0 - 7).

ch:

specifies the channel (0 - 2).

**Data32:*

[output] get the external trig latched position.

Return Values

Please refer the Error Code.

Example

[C/C++]

```
Int slot , ch ;
long Data32;
i9093_ReadExTrigLatchedPosition(slot, ch,&Data32);
```

[C#]

```
Int slot , ch ;
Int32 Data32;
pac8093WNet.pac8093W.ReadExTrigLatchedPosition(slot, ch, ref Data32);
```

Appendix A. Error Code

Error Code	Definition	Description
0	OK	This indicates that there have been no errors
-1	ID_ERROR	There was a problem with the module ID
-2	SLOT_OUT_RANGE	There was a Slot index error (0 - 7)
-3	CHANNEL_OUT_RANGE	There was a Channel index error (0 - 15)
-4	MODE_ERROR	There was a Mode error(CW/CCW, DIR/Pulse, A/B Phase)
-5	FIRMWARENOTSUPPORT	The Firmware is not support.

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	January 2018	Initial issue
2.0.0	July 2018	Added content for the I-9093 modules Added 2.Quick start Added 3.Compared Trig Out Added API naming table Added Compare table Added Applicable Platform table Modify library , demo path Modify API