I-8092F Getting Started Manual

(Version 2.2)

Hardware & Software & Application Using I-8092F PAC Motion Control Module



icpdas.com

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright 1997-2005 by ICPDAS Inc., LTD. All rights reserved worldwide.

Trademark

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

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Contents of I8092F

1 INTRODUCTION	6
1 1 Introduction	6
1.1 Introduction	0
1.2.1 Main Specification	
1.2.2 Interpolation Function	
1.2.3 Pulse Output	
1.2.4 Encoder Input	
1.2.5 Position counter	
1.2.6 Servo Motor Input Signal	
1.2.7 Limit Switch Input Signal	
1.2.8 Other Input Signals	
1.2.9 Emergency Stop Signal Input	
1.2.10 General Output Signal	
1.2.11 Integral Input Signal Filters	9
1.2.12 Software Limit	9
1.2.13 Manual Pulse Generator	9
1.2.14 LED for Module status	9
1.2.15 FRnet	9
1.3 Environment	
1.4 Ordering Information	10
2 HARDWARE INSTALLATION	11
2.1 Checking Package and Installation	
2.1.1 Checking package	11
2.1.2 Installation	11
2.2 DN-8237-GB Daughter Board	
2.2.1 Board Layout for DN-8237-GB	12
2.2.2 Signal Connections for DN-8237-GB	13
2.2.3 Jumper and Switch Settings	17
2.3 Input/Output Connections	
2.3.1 Pulse output signals	20
2.3.2 Connection for Limit switch Signal	22
2.3.3 General Purpose Input Signals(nINPOS,nALARM)	22
2.3.4 Encoder Signals	23
2.3.5 Emergency Stop Signal	23

2.3.6 Manual Pulse Generator Input Signal (EXP+,EXP-)	24
2.3.7 General Purpose Output signals(Servo On/Off)	24
2.4 Connection Example for Motor Driver	
3 SOFTWARE DEVELOPMENT OVERVIEW	26
3.1 Software development Overview	26
3.1.1 Register Module	27
3.2 Safety IO Setting	
3.2.1 Emergency Stop Signal Input	27
3.2.2 Configure the Servo ALARM Signals	27
3.2.3 Configure the Limit Switch Signals(±EL)	27
3.2.4 Configure the Software Limite(±SEL)	28
3.3 Error Checking	
3.4 Basic Configuration of Motion	
3.5 Manual Pulse Generator Testing	
3.6 Home Search	
3.6.1 Home Search Configuration	30
3.7 Basic Motion	
3.7.1 Speed Profie of the Motion Control	32
3.7.2 Basic Setting of Single Axis	33
3.7.3 Basic Motion of Single Axis	34
3.7.4 Basic Setting of Muti-Axes Interpolation	34
3.7.5 Basic Motion of Muti-Axes Interpolation	35
3.8 Advance Motion	
4 GETTING STARTED OF SOFTWARE	
4.1 WinCon eVC++ Guideline	
4.1.1 Confirm the Relative Files	37
4.1.2 Create a new eVC++ Application Project	37
4.1.3 Add the I8094.h into eVC++ Application Project	39
4.1.4 Add the Reference Path into eVC++ Application Project	40
4.1.5 Start the eVC++ Sample	41
4.1.6 Build the Project	44
4.1.7 Download and Run	45
4.2 Microsoft Visual Studio .NET 2003(VB.NET , C#) Guidelin	ne46
4.2.1 Confirm the Relative Files	46
4.2.2 Create a new VB.NET/C# Application Project	46

4.2.3 Add the DLL into Application Project	
4.2.4 Start the VB.NET/C# Sample	
4.2.5 Build the Project	
4.2.6 Download and Run	
4.3 I-8000 Turbo C Guideine	
4.3.1 Confirm the Relative Files	
4.3.2 Create a new TC ++ Application Project	
4.3.3 Start the TC Sample	
4.3.4 Build the Project	
4.3.6 Download and Run	
APPENDIX-A SETUP TOOLS & OTI	HERS65
A.1 Setup the Development Environment of I80	9465
A.1.1 eVC ++ 4.0	
A.1.2 Visual Studio .NET 2003(VB.NET , C#)	
A.1.3 Turbo C	
A 2 ISOOA Surface	66
A.2 10094 Sui lace	······································
A.3 Dimensions	
A.2 Dimensions A.3 Dimensions A.4 The Version Upgrades Note	
A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E	67 67 80ARDS
A.2 18094 Surface A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E B 1 DN-8237-DB Daughter Board	67 67 67 80ARDS
A.2 Dimensions A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board B 1 1 Board Layout for DN-8237-DB	67 67 80ARDS68 68
A.2 Dimensions A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB	67 67 67 80ARDS68 68 68 68
A.2 Dimensions A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings	67 67 67 80ARDS
A.2 DN-8237-MB Daughter Board B.1.3 Jumper and Switch Settings	67 67 67 67 80ARDS
A.2 Dimensions A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings B.2 DN-8237-MB Daughter Board B.2.1 Board Layout for DN-8237-MB	67 67 67 67 68 68 68 68 69 74 77 77
A.2 ISO94 Surface A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings B.2 DN-8237-MB Daughter Board B.2.1 Board Layout for DN-8237-MB. B.2.2 Signal Connections for DN-8237-MB.	67 67 80ARDS68 68 68 68 69 74 77 77
A.2 ISO94 Surface A.3 Dimensions. A.4 The Version Upgrades Note. APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board. B.1.1 Board Layout for DN-8237-DB. B.1.2 Signal Connections for DN-8237-DB. B.1.3 Jumper and Switch Settings. B.2 DN-8237-MB Daughter Board. B.2.1 Board Layout for DN-8237-MB. B.2.2 Signal Connections for DN-8237-MB. B.2.3 Jumper and Switch Settings.	67 67 67 67 68 68 68 68 69 74 74 77 77 78 84
A.2 ISO94 Surface A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings B.2 DN-8237-MB Daughter Board B.2.1 Board Layout for DN-8237-MB B.2.2 Signal Connections for DN-8237-MB B.2.3 Jumper and Switch Settings B.2.3 Jumper and Switch Settings	67 67 67 67 68 68 68 69 74 77 77 77 78 84
 A.2 18094 Surface. A.3 Dimensions. A.4 The Version Upgrades Note. APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board. B.1.1 Board Layout for DN-8237-DB. B.1.2 Signal Connections for DN-8237-DB. B.1.3 Jumper and Switch Settings. B.2 DN-8237-MB Daughter Board. B.2.1 Board Layout for DN-8237-MB. B.2.2 Signal Connections for DN-8237-MB. B.2.3 Jumper and Switch Settings. B.3 DN-8237-PB Daughter Board. B.3.1 Board Layout for DN-8237-PB. 	67 67 67 67 68 68 68 68 69 74 74 77 77 78 84 84 86
 A.2 18094 Stillace A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings B.2 DN-8237-MB Daughter Board B.2.1 Board Layout for DN-8237-MB B.2.2 Signal Connections for DN-8237-MB B.2.3 Jumper and Switch Settings B.3 DN-8237-PB Daughter Board B.3.1 Board Layout for DN-8237-PB B.3.2 Signal Connections for DN-8237-PB 	67 67 67 67 80ARDS68 68 68 69 74 74 77 77 78 84 84 86 86
 A.2 18094 Surface A.3 Dimensions A.4 The Version Upgrades Note. APPENDIX B OTHER TERMINAL E B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings. B.2 DN-8237-MB Daughter Board B.2.1 Board Layout for DN-8237-MB B.2.2 Signal Connections for DN-8237-MB B.2.3 Jumper and Switch Settings. B.3 DN-8237-PB Daughter Board B.3.1 Board Layout for DN-8237-PB B.3.2 Signal Connections for DN-8237-PB B.3.3 Jumper and Switch Settings. 	67 67 67 67 68 68 68 68 69 74 74 77 77 78 84 84 84 86 86 87 92
 A.2 18094 Surface. A.3 Dimensions. A.4 The Version Upgrades Note. APPENDIX B OTHER TERMINAL F B.1 DN-8237-DB Daughter Board. B.1.1 Board Layout for DN-8237-DB. B.1.2 Signal Connections for DN-8237-DB. B.1.3 Jumper and Switch Settings. B.2 DN-8237-MB Daughter Board. B.2.1 Board Layout for DN-8237-MB. B.2.2 Signal Connections for DN-8237-MB. B.2.3 Jumper and Switch Settings. B.3 DN-8237-PB Daughter Board. B.3.1 Board Layout for DN-8237-PB. B.3.2 Signal Connections for DN-8237-PB. B.3.3 Jumper and Switch Settings. 	67 67 80ARDS
 A.2 18094 Surface A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL F B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings B.2 DN-8237-MB Daughter Board B.2.1 Board Layout for DN-8237-MB B.2.2 Signal Connections for DN-8237-MB B.2.3 Jumper and Switch Settings B.3 DN-8237-PB Daughter Board B.3.1 Board Layout for DN-8237-PB B.3.2 Signal Connections for DN-8237-PB B.3.3 Jumper and Switch Settings B.3.4 DN-8237-YB Daughter Board B.4 DN-8237-YB Daughter Board B.4.1 Board Layout for DN-8237-YB 	67 67 80ARDS68 68 68 68 69 74 74 77 77 77 78 84 84 86 86 87 92 92 94
 A.2 18094 Surface A.3 Dimensions A.4 The Version Upgrades Note APPENDIX B OTHER TERMINAL F B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings B.2 DN-8237-MB Daughter Board B.2.1 Board Layout for DN-8237-MB B.2.2 Signal Connections for DN-8237-MB B.2.3 Jumper and Switch Settings B.3 DN-8237-PB Daughter Board B.3.1 Board Layout for DN-8237-PB B.3.2 Signal Connections for DN-8237-PB B.3.3 Jumper and Switch Settings B.4 DN-8237-YB Daughter Board B.4.1 Board Layout for DN-8237-YB B.4.2 Signal Connections for DN-8237-YB 	67 67 80ARDS68 68 68 69 74 74 77 77 78 84 84 86 86 86 87 92 92 94 94
 A.2 18094 Surface A.3 Dimensions A.4 The Version Upgrades Note. APPENDIX B OTHER TERMINAL F B.1 DN-8237-DB Daughter Board B.1.1 Board Layout for DN-8237-DB B.1.2 Signal Connections for DN-8237-DB B.1.3 Jumper and Switch Settings. B.2 DN-8237-MB Daughter Board B.2.1 Board Layout for DN-8237-MB B.2.2 Signal Connections for DN-8237-MB B.2.3 Jumper and Switch Settings. B.3 DN-8237-PB Daughter Board B.3.1 Board Layout for DN-8237-PB B.3.2 Signal Connections for DN-8237-PB B.3.3 Jumper and Switch Settings. B.4 DN-8237-YB Daughter Board B.4.1 Board Layout for DN-8237-YB B.4.2 Signal Connections for DN-8237-YB B.4.3 Jumper and Switch Settings. 	67 67 80ARDS68 68 68 69 74 74 77 77 78 84 84 86 86 86 87 92 92 94 94 95

1 INTRODUCTION

1.1 Introduction

The I-8092F are the 2-axes pulse-type stepping/servo motor motion control module that can be used on any of the ICPDAS I-8000, WinCon and LinCo series controllers, and is suitable for general-purpose motion application. These modules contain a high-performance motion ASIC. Apart from a wide speed range, these intelligent motion controllers have a variety of motion control functions built in, such as 2-axes linear interpolation, 2-axes circular interpolation, T/S-curve acceleration/ deceleration, automatic homing, and others. Besides, it is a module that has full functions of I-8092F plus one port of FRnet. The FRnet port allows this module to expand its fast remote I/O easily. This two-wired FRnet can automatically scan its 128 DI and 128 DO with a period of 0.72/2.88ms. In addition, most of the I-8092F motion control functions are performed with little load on the processor. While driving the motors, the motion status, and the other I/O status on the I-8000, WinCon, or LinCon controllers, can still be monitored. As a result of the low CPU loading requirements of I-8092F, one or more motion modules may be used on a single I-8000, WinCon, ot LinCon controllers. ICPDAS also has provided a wide range of functions and examples to reduce the need for programming by user, making it a highly cost-effective solution for machine makers.



I8092f with PAC controller (WinCon-8000 \ LinCon-8000 \ I-8000)

1.2 Hardware Specification

1.2.1 Main Specification

ASIC Chip	MCX312
Number of controllable	2-Axes, Pulse output (stepping & servo
	motor)
Up to 4M PPS pulse output	

1.2.2 Interpolation Function

2-axes & 3-axes linear interpolation

Interpolation range -8,388,607 ~ +8,388607
 Vectors speed of interpolation 1 PPS ~ 4M PPS
 Precision of interpolation ± 0.5 LSB
 Circular interpolation

Interpolation range −8,388,607 ~ +8,388607
 Vectors Speed of interpolation 1 PPS ~ 4M PPS

Bit interpolation

Vectors Speed of interpolation

1 PPS ~ 4M PPS(Dependent on CPIU data writing time)

Relative interpolation function

- Fixed vectors speed
- Continuous interpolation

1.2.3 Pulse Output

Output speed range	1 PPS ~ 4 MPPS
Output precision	± 0.1%
Jerk range of S-curve	954 ~ 62.5 x 10^6 PPS/S^2
	477 x 10^3 ~ 31.25 x 10^9 PPS/S^2
Acceleration/deceleration range	125 ~ 1 x 10^6 PPS/S
	62.5×10^3 ~ 500 x 10^6 PPS/S
Speed precision	1 PPS ~ 500PPS(Depend on the
	max.speed)
Output numbers	0 ~ 4,294,967,295 / unlimited
Velocity profiles mode:	

- Fixed
- Symmetrical & Asymmetrical Trapezoidal velocity profile

- Symmetrical & Asymmetrical S-curve velocity profile
- Acceleration & Deceleration mode
 - Auto
 - By user define
- Position & Speed change on the fly
- Fixed pulse output by Trapezoidal and S-curve velocity profile
- Pulse output option: CW/CCW, PULSE/DIR
- Programmable logic level (Rising Edge/ Falling Edge)

1.2.4 Encoder Input

- Encoder option: A/B phase, Up/Down
- Programmable A/B phase mode: 1, 1/2, and 1/4 A/B phase
- Programmable direction of counter

1.2.5 Position counter

- Command counter range -2
 - $-2,147,483,648 \sim +2,147,483,647$
- Encoder counter range
- -2,147,483,648 ~ +2,147,483,647
- Programmable ring counter
- Programmable direction of counter
- Programmable read & write counter

1.2.6 Servo Motor Input Signal

- Alarm
- Choose IN2: In Position or Servo Ready signal
- Choose input signal: Enable/Disable and logical level.

1.2.7 Limit Switch Input Signal

- Two-limit switch signal for each axis: +Limit, -Limit
- Programmable logic level
- Programmable action mode(slow-down stop or immediately stop)

1.2.8 Other Input Signals

■ IN3 : Digital Input of general purpose.

1.2.9 Emergency Stop Signal Input

There is a Emergency stop signal for Each module.

1.2.10 General Output Signal

- Capability of configurable nOUT0 of each axes as general purpose DO signals.
- Capability of configurable nOUT1 of each axes as Servo On/Off signal.

1.2.11 Integral Input Signal Filters

The motion module is equipped with an integral type filter in the input step of each input signal. User can be selected a filter time constant.

1.2.12 Software Limit

■ There are two software-limit for each axis: -SLimit & + SLimit (Setting range : -2,147,483,646 ~ +2,147,483,646)

1.2.13 Manual Pulse Generator

- Fixed Pulse Driving Mode (CW/CCW pulse mode)
- Continuous Pulse Driving Mode (CW/CCW pulse mode)
- Manual pulsar mode(A/B phase pulse mode)
- Disable Mode: Disable manual pulse function

1.2.14 LED for Module status

- Red LED \rightarrow Power light
- Orange LED → Servo Alarm
 - Ex:Misuibishi driver, No Alm: turn Orange LED on
- Green LED → during Running Motion

1.2.15 FRnet

Connect to the distributed DI/DO module

DI \rightarrow max up to 128

- $DO \rightarrow max up to 128$
- Read the status of distributed DI
- Control the status of distributed DO
- Support interrupt and frequence division function
- Reset function

1.3 Environment

- Operating Temp: -20 ~ + 75°C
- Storage Temp:
- Operating Humidity: Storage Humidity:
 - $5 \sim 90\%$, non-condensing

10 ~ 85% , non-condensing

-30 ~ +85°C

- 2500Vrms I/O optically isolated
- External Power supply(Input): 24V DC (connect to terminal board)

1.4 Ordering Information

- I-8000 V-8000 L-8000 PAC controllers
- 2-axes motion control module i8092F
- DN-8237 For general purpose usage
- CA-3710D 37-pin Dsub cable , length:1.0 m
- i8092F-G/S i-8092F+DN-8237+CA-3710D

2 HARDWARE INSTALLATION

2.1 Checking Package and Installation

2.1.1 Checking package

The i8092F are a 2-axes stepping/servo motor control module that can be used on any of the ICPDAS I-8000 and WinCon series controllers. The base system package is as below list:

I-8000
 W-8000 Embedded PAC control system series(Two systems choose one)

i8092F-G/S includes the following item

- i8092F 2-axes motion module
- DN-8237 Terminal board for i8092F
- CA-3710D 37-pin Dsub cable , length:1.0 m

2.1.2 Installation

Prepare controller

- 1. Choose a PAC controller of ICPDAS (I-8000 or W-8000series) and have empty slot.
- 2. Turn power off

Module Plug in controller and wiring

- 1. Plug in the i8092F into a empty slot of I-8000/W-8000.
- 2. Connect the i8092F with DN-8237 by a CA-3710D cable, as the below figure:



Figure. i8092F with PAC controller (WinCon-8000 \ LinCon-8000 \ I-8000)

2.2 DN-8237-GB Daughter Board

The DN-8237-GB is the daughter board for General Purpose Ampilifiers. It has 2-axis I/O signals.

2.2.1 Board Layout for DN-8237-GB



Fig. 3-1 Board layout for the DN-8237-GB

2.2.2 Signal Connections for DN-8237-GB

Maintaining signal connections is one of the most important factors in ensuring that your application system is sending and receiving data correctly.

Pin Assignment for CON1

The I/O connector on the DN-8237-GB is a 37-pin connector that enables you to connect to the PISO-PS200(or I-8092F) motion card. Fig. 3-2 shows the pin assignment for the 37-pin I/O connector on the DN-8237-GB (or on the motion card), and refer to Table 3-2 for description of each motion I/O signal.

		10	
FR A		19	
FRB		37	
X ECA	17. 17.	18	
YECA	<i>6</i>	36	
X ECB	-	17	
Y ECB		35	
V STOR2		16	
X_STOP2		34	
Y INDOS		15	
X_INPOS		33	
T_INPOS		14	
X_ALARM	-	32	
Y_ALARM		13	
X_LMTP		31	
Y_LMTP		12	
X_LMTM	1	30	
Y_LMTM		11	
X_STOP0		20	
Y_STOP0		29	
X_STOP1		10	
Y STOP1		28	
X_IN3		9	
Y IN3		21	
X EXPP		8	
Y EXPP		26	
X EXPM	22 22	7	
V EYDM		25	
	22	6	
X_OUTO		24	
	0 8	5	
		23	
Y_0011		4	
X_PP		22	
Y_PP		3	
X_PM		21	-0
Y_PM		2	
EMGN	-	20	$-\circ$
VCC		1	
GND		1	
	<u> </u>		

Fig. 3-2 I/O connector pin assignment for the CON1

Pin name	Pin number	Description	
FR_A	19	FRnet A-phase signal	
FR_B	37	FRnet B-phase signal	
X_ECA	18	Encoder A-phase signal for the X axis	
Y_ECA	36	Encoder A-phase signal for the Y axis	
X_ECB	17	Encoder B-Phase signal for the X axis	
Y_ECB	35	Encoder B-Phase signal for the Y axis	
X_STOP2	16	Stop 2 signal for the X axis	
Y_STOP2	34	Stop 2 signal for the Y axis	
X_INPOS	15	In-position signal for the X axis	
Y_INPOS	33	In-position signal for the Y axis	
X_ALARM	14	Alarm signal for the X axis	
Y_ALARM	32	Alarm signal for the Y axis	
X_LMTP	13	Limit switch input signal (+) for the X axis	
Y_LMTP	31	Limit switch input signal (+) for the Y axis	
X_LMTM	12	Limit switch input signal (-) for the X axis	
Y_LMTM	30	Limit switch input signal (-) for the Y axis	
X_STOP0	11	Stop 0 signal for the X axis	
Y_STOP0	29	Stop 0 signal for the Y axis	
X_STOP1	10	Stop 1 signal for the X axis	
Y_STOP1	28	Stop 1 signal for the Y axis	
X_IN3	9	Input 3 signal for the X axis	
Y_IN3	27	Input 3 signal for the Y axis	
X_EXPP	8	EXT pulsar input signal (+) for the X axis	
Y_EXPP	26	EXT pulsar input signal (+) for the Y axis	
X_EXPM	7	EXT pulsar input signal (-) for the X axis	
Y_EXPM	25	EXT pulsar input signal (-) for the Y axis	
X_OUT0	6	Output 0 signal for the X axis	
Y_OUT0	24	Output 0 signal for the Y axis	
X_OUT1	5	Output 1 signal for the X axis	
Y_OUT1	23	Output 1 signal for the Y axis	
XPP	4	Driving pulsar signal (+) for the X axis	
YPP	22	Driving pulsar signal (+) for the Y axis	
XPM	3	Driving pulsar signal (+) for the X axis	
YPM	21	Driving pulsar signal (+) for the Y axis	
EMGN	2	Emergency stop input signal	
VCC	20	Module power (+5V)	
GND	1	Ground	

Table 3-2 DN-8237-MB CON1 I/C	connector signal description
-------------------------------	------------------------------

■ CON2 & CON3 (I/O connector for each AXIS)

The connectors CON2 and CON3 are 20-pin connectors that enable you to connect to the I/O signals for general purpose motor drivers. Fig.3-3 shows the pin assignment for the 20-pin connector on the DN-8237-GB, and the Table 3-3 shows its I/O connector signal description.





Name	Number	Description
A+	1	Encoder A-Phase (+)
A-	2	Encoder A-Phase (-)
B+	3	Encoder B-Phase (+)
В-	4	Encoder B-Phase (-)
Z+	5	Encoder Z-Phase (+)
Z-	6	Encoder Z-Phase (-)
P+	7	Positive Direction Pulse Output(+)
P-	8	Positive Direction Pulse Output(-)
N+	9	Negative Direction Pulse Output(+)
N-	10	Negative Direction Pulse Output(-)
INP	11	Servo In Position
ALARM	12	Servo Alarm
SRV_ON	13	Servo On
LMT+	14	Limit Switch Input Signal (+)
LMT-	15	Limit Switch Input Signal (-)
IN3	16	Input Signal (IN3)
HOME	17	Home Sensor Input Signal
NHOME	18	Near Home Sensor Input Signal
EXP+	19	EXT Positive Direction Pulse (+)
EXP-	20	EXT Negative Direction Pulse (-)

Table 3-3 CON2 ~ CON3 Signal Connection

The connector CON4 is 16-pin connector that enables you to connect to the signals of your motor drivers. Fig.3-4 shows the pin assignment for the 16-pin connector on the DN-8237-GB, and the Table 3-4 shows its I/O connector signal

description.



Table 3-4 CON4 Signal Connection

	5
Name	Description
FR-A	FRnet port A
FR-B	FRnet port B
X-DCC	Deviation Counter Clear for X axis
Y-DCC	Deviation Counter Clear for Y axis
E-PLS	EXT pulse signal
EMG-A	EMG input signal for all axes
E-GND	EXT power ground
X-EMG	EMG input signal for X axis
Y-EMG	EMG input signal for Y axis
Z-EMG	EMG input signal for Z axis
U-EMG	EMG input signal for U axis
X-RDY	Ready input signal for X axis
Y-RDY	Ready input signal for Y axis
Z-RDY	Ready input signal for Z axis
U-RDY	Ready input signal for U axis

Fig. 3-4 Pin definition for CON4

TB1

The connector TB1 is 5-pin connector that enables you to connect to the signals of your motor drivers. Fig.3-4 shows the pin assignment for the 5-pin connector on the DN-8237-GB, and the Table 3-4 shows its I/O connector signal description.

E-PWR E-PWR E-GND E-GND FGND	1 2 3 4 5

Table 3-4 TB1 Signal Connection		
Pin name	Description	

Pin name	Description
E-PWR	EXT power supply +24V
E-GND	EXT power ground
FGND	Frame ground

Fig. 3-4 Pin definition for TB1

Note: Don't reverse connect signals with E_PWR and E_GND. Serious damage to your motion card and motion controller might be happened.

■ RJ1 (The I/O signals of the FRnet)

The connectors RJ1 is an 8-pin RJ45 connector that enable you to connect to the signals of FRnet. Fig.3-5 shows the pin assignment for the 8-pin connector on the DN-8237-GB, and the Table 3-5 shows its I/O connector signal description.



Table	9 3-5 RJ1
Pin name	Description
FRnetA	FRnet port A
FRnetB	FRnet port B
NC	No connection

Fig. 3-5 Pin definition for RJ1

Note: Don't connect NC (not connected) signals. Connecting these signals could cause permanent damage to your motion controller.

2.2.3 Jumper and Switch Settings

■ JP5

Jumper 5 controls the EMG-A signal of the CON4 connector. The following diagram is shown the selection condition of the jumper 5.



Fig. 3-6 Jumper 5 setting

■ SW 1

The emergency stop signal for each servo ampilfier can be selected from SW1. The number 1 and 2 on SW1 are denoted as axis X and Y, respectively. The number 3 and 4 on SW1 are reserved for future work. Fig. 3-7 is the default setting to connect the EMG singals to GND. The X-EMG and Y-EMG signal from CON4 not take effect. If the switch is disconnected as shown in Fig. 3-8, the emergency stop signals can be controlled from the X-EMG and Y-EMG signal in CON4.



Fig. 3-7 SW1 setting for normally GND (Default setting)



Fig. 3-8 SW1 setting for user controlled signals.

■ JP1/2 & JP3/4

Jumper 1, 2 controls the XPP, XPM signals of the CON2. The couple of jumpers are indicated the type of pulse output signal for X axis. However there are the same jumper settings for Y axis (Jumper 3, 4 for Y axis). The following diagram is shown the selection condition of the jumper 1, 2.



Fig. 3-9 Jumper 1, 2 setting

2.3 Input/Output Connections

The signal connections of all the I/O signals are described in this chapter. Please refer the contents of this chapter befor wiring the cable between the i8092F and the motor drivers.

2.3.1 Pulse output signals

There are 2-axes pulse output signals on I8092F, For every axis, two pairs of CW and CCW signals are used to send the pulse train. The CW and CCW signals can also be programmed as PULSE and DIR signals pait. Two types of the pulse output signal, Differential-Type and Open-Collector Type, can be selected from JP2/3 and JP4/5 and are described in section 2.2.2. The following wiring diagram is for the CW and CCW signals of the 2-axes.

• Output to Motor Drivers in Differential Circuit



Fig. 2.8 Differential-Type pulse output circuit

• Open Collector TTL Output



Fig. 2.9 The wiring is open collector output

Example: wiring of pulse signal

Two types of pulse output signal, Differential-Type and Open-Collector Type, can be selected from JP2/3 and JP4/5 for each axis. The following wiring diagram is an example to select pulse type of the output signal



Fig. 2.10 Output pulse example

Pulse/Direction Pulse Output Mode:

In Pulse/Direction pulse output mode, the PULSE signal is output only at Pulse pins (P+, P-). The driving direction is decided from the electric potential of Direction pins (N+, N-). The following diagram is example signal of Pulse/Direction pulse output mode.



• CW/CCW Pulse Output Mode:

In CW/CCW pulse output mode, the PULSE signal is output at both CW pins (P+, P-) and CCW pins(N+, N-). At the same time, the driving direction is determined directly. The following diagram is example signal of CW/CCW pulse output mode.

Ρ±	
N±	
	Positive Command
Ρ±	
N±	
	Negative Command

2.3.2 Connection for Limit switch Signal

Limit Switch Signal can prevent the over traveling appearance of the motion system. User can set the hardware limit switch signal to be normal open or normal close by the software instruction in I8092F software manual. The following figure indicates that the photo couplers are used to keep out the sensor noise of the Limit Switch.



Fig. 2.11 Limit switch signal circuit

2.3.3 General Purpose Input Signals(nINPOS,nALARM)

INPOS is a digital input signal to indicate the In-Position signal of the driver. User can enable or disable the signal from the software instruction in I8092F software manual.

ALARM is a digital input signal to indicate the servo alarm signal of the driver. The output pulse will be stop if I-8092F receives the ALARM signal. User can enable or disable the signal from the software instruction in I8092F software manual.



Fig. 2.12 General Digital Input circuit

2.3.4 Encoder Signals

The following diagram is for Differential-Type encoder signals. Connect the Phase A signal to A+ and A- pins and connect Phase B signal to B+ and B- pins. After the high speed photo coupler isolation, the isolated encoder signals are connected to motion IC.



Fig. 2.13 Encoder signal connection

2.3.5 Emergency Stop Signal

The following diagram is for Emergency STOP signal. If the emergency signal is occurred, the output pulse for all axes will be STOP and the error flag will be set as 1. After the photo coupler isolation, the isolated emergency signal is connected to motion IC.



Fig. 2.14 Emergency Stop Signal connection

2.3.6 Manual Pulse Generator Input Signal (EXP+, EXP-)

The signals, EXP+ and EXP-, are used for manual pulsar signals. The following diagram is an example connection for the external inputs. User can set the signals as fixed pulse CW/CCW mode, continuous pulse CW/CCW mode, or A/B phase manual pulsar mode by using the setting in section 3.5.



Fig. 2.15 EXP+/- connection diagram

2.3.7 General Purpose Output signals(Servo On/Off)

The following diagram is a digital output signal for driver Servo On/Off signal. The output signal enable or disable the driver.



Fig. 2.16 Servo On/Off signal connection diagram

2.4 Connection Example for Motor Driver

The following diagram is the connection example between MITSUBISH MR-J2S AC servo driver and the extension boardDN-8237.



Fig. 2.17 The connection between MR-J2S AC servo driver and DN-8237 extension board.

3 Software Development Overview

3.1 Software development Overview

Please refer to the demo_start sample



3.1.1 Register Module

User must register for each I8092F module before sending command otherwise user will get error. Please refer to *i8092MF_REGISTRATION()* function, the section 2.2 of I8092F user manual.

3.2 Safety IO Setting

There are many reasons to stop motion during driving. Some reasons are described in this subsection.

3.2.1 Emergency Stop Signal Input

Emergency Stop is especially for the purpose to stop all of the Motion operations immediately when danger occurs in order to avoid critical accident.

If you don't need to use the Emg. stop push button, configure the JP1 as pin2-3 short which descripts in the section 2.2.2 .If you need the EMG signal input , configure the JP1 as pin1-2 short, and the EMG_IN signal connect to the N.C. type EMG push button switch and install it at the suitable location.

3.2.2 Configure the Servo ALARM Signals

When the ALARM signals are occurred from servomotor drivers, users can be notified by these signals and determine what to do. The operating mode (Enable or Disable) and the proper trigger level of these signals can be set by user. Please refer to *i8092MF_SET_ALARM()* function, the section 2.13 of I8092F user manual.

3.2.3 Configure the Limit Switch Signals(±EL)

To insure the machine in safety, hardware limit switches are placed at the both ends of machine traveling range. If the machine touch the hardware limit switch sensors, I-8092F will stop immediately. The operating mode (Enable or Disable) and the proper trigger level of these signals can be set by user. Please refer to *i*8092*M*F_*SET_HLMT* () function, the section 2.6 of I8092F user manual.

3.2.4 Configure the Software Limite(±SEL)

To insure the machine in safety, hardware limit switches are placed at the both ends of machine traveling range. In addition, user can set the software limits to avoid the happening of the over range before the hardware limit takes effect. If the machine reach the software limits condition, I-8092F will stop immediately. The operating mode (Enable or Disable) and the proper trigger condition of these signals can be set by user. Please refer to i8092MF_SET_SLMT () and i8092MF_CLEAR_SLMT() function, the section 2.10 of I8092F user manual.

3.3 Error Checking

Check whether there is any error. If there are something wrongs, please use the *GET_ERROR_CODE()* function to get the error-code, then check the reason and remove it. Please refer to *GET_ERROR_CODE()* function, the section 3.6 of I8092F manual.

User also can use *i8092MF_GET_DI()* function to check the all of DI status. Please refer to *i8092MF_GET_DI()* function, the section 3.5 of I8092F user manual.

3.4 Basic Configuration of Motion

The basic Motion configuration is mainly aimed for general necsseary setting,

as below:

1 Pulse output mode setting: Pulse/Dir
 CW/CCW...

i8092MF_SET_PULSE_MODE() (Please refer to the section 2.4 of I8092F user manual)

2 Max. speed limitation setting for each axis

i8092MF_SET_MAX_V ()(Please refer to the section 2.5 of I8092F

user manual)

3 Encoder input setting

i8092MF_SET_ENCODER()(Please refer to the section 2.11 of I8092F user manual)

4 DI noise filter setting(If necessary)

i8092MF_SET_FILTER()(Please refer to the section 2.15 of I8092F

user manual)

5 Circular motion declaration(Ring counter)(If necessary)

icpdas.com

i8092MF_VRING_ENABLE()(Please refer to the section 2.16 of I8092F user manual)

3.5 Manual Pulse Generator Testing

User can use the manual pulse generator function directly to drive motion forward or backward. For further wiring and parameter tuning, user have to check the correction of the DI signals and the moving direction.

The manual pulse generator can be achieved from three driving methods described below:

1. A/B phase Manual Pulse Generator:

Use the A/B phase Manual Pulse Generator for forward/backward moving. *i8092MF_EXD_MP()*(Please refer to the section 2.18.1 of I8092F user manual)



2. Fixed-pulse driving Manual Pulse Generator:

User have to preset fixed driving pulses. After setting, user can push the forward or backward button to drive fixed pulses for each direction.

i8092MF_EXD_FP()(Please refer to the section 2.18.2 of I8092F usere

manual)

3. Continuous- pulse driving Manual Pulse Generator:

User can preset output-pulse frequency. After setting, user can push the forward or backward button to drive fixed velocity for each direction. If user release the button, the motion will be stop immediately.

```
i8092MF_EXD_CP ()( Please refer to section 2.18.3 of I8092F user manual).
```

4 Disable external pulse input:

Disable external pulse input by this command after operating anyone of three functions above.

i8092MF_EXD_DISABLE() (Please refer to section 2.18.4 of I8092F user manual)

3.6 Home Search

18092F provides the home function of automatic search. Operate that automatically after setting properly. The main steps is as bellow:

- Near-home sensor searching under high-speed motion.
- Near-home sensor searching under high-speed motion.
- Servomotor Z-phase searching under low-speed motion.
- Offset movement to the origin of the working area under high-speed motion.

User can select which steps are ignored when setting for the actual operation. It performs automatically that economize the CPU resource and program code reducing. Although there are four home search steps, but user can create more than 10 types of different home search mode by vary with the software functions. It is attributed to the configurable home search direction and perform it or not of each step.

3.6.1 Home Search Configuration

- Logic level setting for Near home sensor and Home sensor (If necessary) *i8092MF_SET_NHOME()* (Please refer to section 2.8 of I8092F user manual)
- 2 Home sensor logic level setting

i8092MF_SET_HOME_EDGE() (Please refer to section 2.9 of I8092F user manual)

3 Auto-Home

i8092MF_AUTO_HOME() (Please refer to section 5.2 of I8092F user manual)

4 Step by step Home function (Only Software) BYTE i8092MF_SEARCH_NHOME()

icpdas.com

BYTE i8092MF_SEARCH_HOME() BYTE i8092MF_SEARCH_ZPHASE() Please refer to section 5.3 of I8092F user manual.

3.7 Basic Motion

3.7.1 Speed Profie of the Motion Control





3 Symmetrical S-curve of motion velocity



4 Asymmetrical S-curve of motion velocity

3.7.2 Basic Setting of Single Axis

- 1 Setting the mode of Acceleration/deceleration: There are four speed modes
 - $0 \rightarrow$ Symmetrical T-Profile (SV \cdot V \cdot A \cdot AO)
 - 1 \rightarrow Symmetrical S-curve (SV \cdot V \cdot K \cdot AO)
 - 2 \rightarrow Asymmetrical T-profile (SV \vee V \vee A \vee D \vee AO)
 - 3 \rightarrow Asymmetrical S-curve (SV \cdot V \cdot K \cdot L \cdot AO)

i8092MF_NORMAL_SPEED()(Please refer to section 6.1.1 of I8092F user

manual)

2 Setting the start velocity: Set lowest speed

i8092MF_SET_SV ()(Please refer to section 6.1.2 of I8092F user manual)

3 Setting the Velocity: Set the desired speed

i8092MF_SET_V ()(Please refer to section 6.1.3 of I8092F user manual)

4 Setting the Acceleration/Deceleration speed: Set the Acceleration/Deceleration speed. *i8092MF_ SET_A ()*(Please refer to section 6.1.4 of I8092F user manual) *i8092MF_ SET_D ()*(Please refer to section 6.1.5 of I8092F user manual)

3.7.3 Basic Motion of Single Axis

- 1 Fixed-pulse driving output: Perform fixed-quantity of single axis pulse output. *i8092MF_FIXED_MOVE()*(Please refer to section 6.1.9 of I8092F user manual)
- 2 Continuous-pulse driving output: Perform continuous pulse output of single axis. *i8092MF_CONTIUNE_MOVE ()*(Please refer to section 6.1.10 of I8092F user manual)
- 3 Waiting for motion done: Waiting for the axis driving accomplished. *i8092MF_STOP_WAIT()*(Please refer to section 6.5.3 of I8092F user manual)

3.7.4 Basic Setting of Muti-Axes Interpolation

1 Setting axes of interpolation: Don't need to select axes to do the interpolation.

2 Setting the mode of Acceleration/Deceleration of vector: There are seven modes as below:

- $0 \rightarrow 2$ -axes(Linear & ARC & Circular) Fixed-vector velocity (VV)
- 1 \rightarrow 2-axes linear symmetrical T-profile (VSV \cdot VV \cdot VA \cdot VAO)
- 2 \rightarrow 2-axes linear symmetrical S-curve (VSV \cdot VV \cdot VK \cdot VAO)
- 3 \rightarrow 2-axes linear asymmetrical T-profile (VSV \cdot VV \cdot VA \cdot VD \cdot VAO)
- $4 \rightarrow N/A$
- 5 \rightarrow 2-axes (ARC & Circular) symmetrical T-profile (VSV \cdot VV \cdot VA \cdot VAO)

6 → 2-axes (ARC & Circular) asymmetrical T-profile (VSV \cdot VV \cdot VA \cdot VD \cdot VAO)

i8092MF_VECTOR_SPEED()(Please refer to section 6.2.2 of I8092F user

manaul)

2 Setting the start vector velocity: Set the lowest vector speed.

i8092MF_SET_VSV()(Please refer to section 6.2.3 of I8092F user manual)

3 Setting the vector velocity: Set the desired vector speed

i8092MF_SET_VV()(Please refer to section 6.2.4 of I8092F user manual)

4 Setting the velocity of Acceleration/Deceleration of vector: Set the speed of

Acceleration/Deceleration of vector.

i8092MF_SET_VA()(Please refer to section 6.2.5 of I8092F user manual) *i8092MF_SET_VD()*(Please refer to section 6.2.6 of I8092F user manual)

3.7.5 Basic Motion of Muti-Axes Interpolation

- 1 2-axes linear interpolation: Perform 2-axes linear interpolation. *i8092MF_LINE_2D()*(Please refer to section 6.2.10 of I8092F user manual)
- 2 2-axes ARC interpolation: Perform 2-axes ARC interpolation. *i8092MF_ARC_CW ()*(Please refer to section 6.2.12 of I8092F user manual) *i8092MF_ARC_CCW ()*(Please refer to section 6.2.12 of I8092F user manual)
- 3 2-axesCircular interpolation: Perform 2-axes Circular interpolation.

i8092MF_ CIRCLE_CW ()(Please refer to section 6.2.13 of I8092F user manual)

i8092MF_ CIRCLE_CCW ()(Please refer to section 6.2.13 of I8092F user manual)

3.8 Advance Motion

1 2-axes continuous interpolation of rectangle: Perform2-axes continuous interpolation of rectangle.

i8092MF_RECTANGLE()(Please refer to section 6.4.1 of I8092F user manual)

2 2-axes continuous interpolation of line:

Initial setting continuous interpolation of 2-axes line(Symmetrical T-profile). *i8092MF_LINE_2D_INITIAL()*(Please refer to section 6.4.2 of I8092F user

manual) Perform 2-axes continuous interpolation of line. *i8092MF_LINE_2D_CONTINUE()*(Please refer to section 6.4.2 of I8092F user manual)

3 Others continuous interpolation: Muti-point continuous interpolation, 3-axes Helix interpolation, 2-axes Ratio motion (Please refer to section 6.4.4~6.4.7 of I8092F user manual)
4 GETTING STARTED OF SOFTWARE

4.1 WinCon eVC++ Guideline

4.1.1 Confirm the Relative Files

Please confirm you have the following relevance files:

- 1. I8092.lib
- 2. I8092.dll
- 3. I8092.h

If you don't have, please look for CD or download the latest edition from ICPDAS's website <u>http://www.icpdas.com/download/download-list.htm</u>.

4.1.2 Create a new eVC++ Application Project

Please execute the Microsoft eVC++ 4.0. Then click "File" -> "New" to create a new application project. In the "Projects" property page, choose "WCE MFC AppWizard (exe)" option and specifythe project name "Demo_First", then key in the disk path in the "Location" field, then select the "Win 32[WCE ARMV4]" in CPU list. If necessary, please also select others options together. And then click "OK".

New	<u>? ×</u>
Files Projects Workspaces	
 WCE Application WCE ATL COM AppWizard WCE Dynamic-Link Library WCE MFC ActiveX ControlWizard WCE MFC AppWizard [dll] WCE MFC AppWizard [excl WCE Static Library 	Project <u>pame</u> : demo_First Location: E:\TEST\demo_First Create new workspace Add to current workspace Dependency of: CPUs: CPUs: CPUs: Win32 (WCE ARMV4) Win32 (WCE ARMV4) Win32 (WCE ARMV4) Win32 (WCE MIPS16) Win32 (WCE MIPS11)
	OK Cancel

Choose " Dialog based " and click "NEXT"

WCE MFC App Wizard (exe) - Step	p 1 of 4			<u>?×</u>
Application		What type of create? <u>Single d</u> <u>Dialog b</u> <u>Mat languag</u> in?	application would y ocument ased :nt/View architectur je would you like y	you like to re support? rour resources
	< <u>B</u> ack	<u>N</u> ext >	<u> </u>	Cancel

Click "Finish" and finish the new project establishment.

4.1.3 Add the I8094.h into eVC++ Application Project

Add the i8092.h into the WorkSpace of application project, as below: Click the right key of mouse on Header Files, then choose "Add Files to Folder...."

📴 demo_First - Microsoft eMbedded Visual C++
]] <u>File Edit Yiew Insert Project Build Iools Window H</u> elp
👔 😂 🖬 🕼 👗 🛍 🛱 🗠 - 🗠 - 🖪 🗖 😚 🎇 //============= - 🖓 🙀 CDemo_FirstApp 💽
demo 🔄 sa_la 🔹 Win32 (WCDARMV4) Debug 💽 SA_la Device 👽 🤣 🕮 🖮
Workspace 'demo_First': 1 proj Gemo_First files Gemo_First files

It will appear on a dialog of selecting file, find out the I8092.h and click OK.



4.1.4 Add the Reference Path into eVC++ Application Project

- A. Open the "Options" dialog in "Tools" menu.
- B. Select "Directories", then select the "SA_IA" in "Platform" item. Then select the "Win32 [WCE ARMV4]" in "CPUS" item and select the "include files" in "Show directories" item.
- C. Add in the path of including files. Double-click the rectangle in the buttom of Directories" List-Box. Please key in the specific path that your header files located. For instance, C:\DAQPRO\Wincon\inc, as below snapshot.
- D. Then select the "Library files" in "Show directories" item.

SA IA	▼ Win32 (WCE ARMV4)	▼ Include files
_ Directories:		 [1] X f
C:\Program	Files\Windows CE Tools\wce4	410\SA IA\Include\ARMV4
C:\Program	Files/Windows CE Tools/wce4	410\SA IA\atl\include 410\SA IA\atl\include
c:\DAQPRO\	wincon\inc	

E. Add in the path of library files. Double-click the rectangle in the buttom of Directories" List-Box. Please key in the specific path that your header files located. For instance, C:\DAQPRO\Wincon\lib, as below snapshot.

SA IA	Win32 (WCE ARMV4)	 Library files
Directories:		
C:\Program F	Files\Windows CE Tools\wce410\	SA IA\Iib\ARMV4
C:\Program I C:\Program I	-iles\Windows CE Tools\wce410\ -iles\Windows CE Tools\wce410\	SA IA\mfc\IIb\ARMV4 SA IA\atl\Iib\ARMV4
C:\DAQPRO\	Wincon\lib	
1		

4.1.5 Start the eVC++ Sample

Add a BUTTON on Dialog, as below snapshot:

📴 demo_First - Microsoft eMbedded Visual C++ - [demo_First.w - IDD_DEMO_FIRST_DIALOG [German (Germany)] (Dialog)]	<u> </u>
J File Edit Yiew Insert Project Build Layout Iools Window Help	_ 8 ×
👔 🖆 😹 🕼 🐍 🖻 🗟 🗠 + 🗠 + 🖪 🕅 🖼 🐄 //	• ID(
demo_First 🗸 SA_IA 🔹 Win32 (WCE ARMV4) Debug 💽 SA_IA Device 💽 🦃 🖄	🛱 🖽 🗷
Image: Second	
】	

Double-click on BUTTON and generate subprogram, then add "#include "i8092.h", "WinConSDK.h", and declare CI8092MF I8092MF & bool Driver_Open & BYTE cardNo=0 in start point, as below snapshot:



Because we have built a class "CI8092MF(For Macro function)", it is convenient to guide in designing program. User also can use the function of manual directly. Double-click on BUTTON that will generate a subprogram, then key in "I8092MF", then it will appear a windows guide to help user to select a relevance function.



Select "i8092MF.REGISTRATION" and key in (cardNo,3), that indicate the i8092F on third slot is registered to 0th module. The detailed procedure is as below:

//===='Step 1 Driver init

```
if (!Driver_Open)
{
     I8092MF.REGISTRATION(cardNo,3);
     Driver_Open = true;
}
//===='Step 2 CONFIG IO
```

18092MF.RESET_CARD (cardNo); I8092MF.SET_PULSE_MODE (cardNo, AXIS_XYZU, 2); //set the pulse output mode I8092MF.SET_ALARM (cardNo, AXIS_XYZU, 0, 0); //disable the SERVO ALARM Input I8092MF.SET_ENCODER (cardNo, AXIS_XYZU, 0, 0, 0); //set the encoder input type I8092MF.SET_MAX_V (cardNo, AXIS_XYZU, 16000); //set the max speed for XYZU 18092MF.EXD_DISABLE (cardNo, AXIS_XYZU); //set the external input Off I8092MF.SET_LP (cardNo, AXIS_XYZU, 0); //set the Logic position =0 //set the Encoger position =0 I8092MF.SET_EP (cardNo, AXIS_XYZU, 0); I8092MF.SET_A (cardNo, AXIS_XYZU, 1000); //set the Acc =1000

```
I8092MF.SERVO_ON (cardNo, AXIS_XYZU);
                                                   //set the Servo_ON to servo motors
 //====='Step 3 Check ERROR
 WORD KK=0;
 KK= I8092MF.GET_ERROR(cardNo);
 CString MSGG;
 if (KK != YES)
 {
      //No ERROR: Step 4 Move X axis
      BYTE axis=AXIS_X; //for AXIS_X it can be to AXIS_XYZU
      I8092MF.SET_MAX_V(cardNo, axis, 20000);
      I8092MF.NORMAL_SPEED(cardNo, axis, 0);
                                                   //set axis as Symmetrical T curve mode
                                                   //set v=10000 PPS
      I8092MF.SET_V(cardNo, axis, 20000);
      I8092MF.SET_A(cardNo, axis, 100000);
                                                   //set acc=100000 PPS/S
      I8092MF.SET_SV(cardNo, axis, 10);
                                                   //set start speed=1000 PPS
      I8092MF.SET_AO(cardNo, axis, 0);
                                                   //set offset pulse (at SV speed)= 0 PS
      I8092MF.FIXED_MOVE(cardNo, axis, 10000);
                                                   //run the fixed 10000 Pulse move.
      while (I8092MF.STOP_WAIT(cardNo, axis) == NO)
      {
           DoEvents();
           Sleep(1);
           //wait for axis to stop
      }
      long AA= I8092MF.GET_LP(cardNo,axis); //Get X Now position
 }
 else
 {
      //Please check the ERROR CODE
      //Get X ERROR CODE
      KK= I8092MF.GET_ERROR_CODE(cardNo, AXIS_X);
      //Get Y ERROR CODE
      KK= I8092MF.GET_ERROR_CODE(cardNo, AXIS_Y);
      //Get Z ERROR CODE
      KK= I8092MF.GET_ERROR_CODE(cardNo, AXIS_Z);
      //Get U ERROR CODE
      KK= I8092MF.GET_ERROR_CODE(cardNo, AXIS_U);
      //------
 }
```

Please refer to the example "demo First"

After you finished that, please choose the "Project"->"Setting" menu will appear the a dialgo as below, then select the "Link" item and key in "WinConSDK.lib i8092.lib"(as below snapshot) into the Object/library modules box and the click OK.

roject Settings	<u> </u>
<u>S</u> ettings For:	General Debug C/C++ Link Resources M
Win32 (WCE ARMV4) Debug ▼	Category: General Reset Output file name: ARMV4Dbg/demo_First.exe Object/library modules: WinConSDK.lib 18092.lib
	☑ <u>G</u> enerate debug info □ Ignore all default li <u>b</u> raries □ Generate <u>m</u> apfile
	Project Options:
	/incremental:yes /pdb:"ARMV4Dbg/demo_First.pdb"
	OK Cancel

4.1.6 Build the Project

Please select the "Build" -> "Build All" in the menu, then you will be finished this example program if there isn't any wrong.

📴 demo_First - Microsoft eMbedded V	isual C++ - [demo_FirstDlg.cpp]	
🖹 Eile Edit View Insert Project	Build Tools Window Help	X
	Compile demo_FirstDlg.cpp Ctrl+F7 Build demo_First.exe F7	======================================
demo_First 💌 SA_IA	Coball All	SA_IA Device 💽 🍪 🖽 🛣
demo_First resource Dialog	Betch Build Clgan Start Debug Update Remote Output File(s) Execute demo_First.exe Ctrl+F5 Set Active Configuration Configurations	<pre>herse for this dialog. The framework does this au plication's main window is not a dialog</pre>
≺ ■Class) ■ Reso) ■ F	SetIcon(m_hIco CenterWindow() // TODO: Add (g, fRGE); // Set Dig icon on, FALSE); // Set small icon SetDesktopWindow()); // center to the hpc scr extra initialization here
The Manual Server con Please make sure Manu Also make sure that y Failed downloading.	nection server has failed. al Server is running and ret ou have selected the correct in Files1	ry the download by Rebuild All or Update Remote platform.
Builds the project		Ln 115, Col 28 REC COL OVR READ

4.1.7 Download and Run

Please copy the "i8092Demo.exe" and "I8092.dll" into the same floder of WinCon (User can use the eVC++ Online Download/FTP/USB disk to do), then execute it.

4.2 Microsoft Visual Studio .NET 2003(VB.NET , C#) Guideline

Because the Microsoft Visual Studio .NET 2003 has similar environment, therefore we make an example with VB.NET.

4.2.1 Confirm the Relative Files

Please confirm you have the following relevance files: i8092.dll

i8092_NET.dll

If you don't have, please look for CD or download the latest edition from ICPDAS's website <u>http://www.icpdas.com/download/download-list.htm</u>

4.2.2 Create a new VB.NET/C# Application Project

Please execute the Microsoft Visual Studio .NET 2003. Then create a new application project of VB and select "Smart Device Application", as below snapshot:

New Project				×
Project Types:		<u>T</u> emplate	es:	
Visual Basic F Visual C# Pro Visual J# Pro Visual J# Pro Visual C++ Pr Setup and Dep Other Projects Visual Studio	rojects jects ects ojects ojects Solutions	Wind Applie Smart 1	VB Class Librar cation VB Device ASP. 104 T We	y Windows Contro
A project for creating :	an application for Pock	Applic ket PC and resourc	cation Application ce-constrained devices	1 Service M
Name:	Demo_First			
Location:	D:\temp			<u>B</u> rowse
New Solution Name:	Demo_First		Create director	y for Solution
Project will be created a	at D:\temp\Demo_First	\Demo_First.		
≵ L <u>e</u> ss		OK	Cancel	Help

Click "OK" after finishing all of the selecting, then go to next step.

Smart Device Application Wizard - Demo	o_First 🛛 🔀
Welcome to the Smart Device Application Wizard This wizard creates a project to develop an application to run on a	smart device.
What platform do you want to target?	
Pocket PC Windows CE	You currently have the following devices installed that will run an application targeting this platform.
What p <u>r</u> oject type do you want to create?	Windows CE .NET Device Windows CE .NET 模擬器 (Default) Pocket PC Device Pocket PC 2002 Emulator
Windows Application Class Library Console Application Empty Project	
	OK Cancel Help

Select the "WinDows CE" and "Windows Application", then click "OK".

4.2.3 Add the DLL into Application Project

Click the right key of mouse on "Solution Explorer" =>add Reference

Object Browser Start Page Form1.vb [Design]*	4 Þ 🗙	Solution Explorer - Demo_First 🛛 🕂 🗙
Form1	1	Properties 4 ×
	1	
Task List - O Build Error tasks shown (filtered)	џ×	
! Image: Click how to click how to be		_
Click here to add a new task		
	>	
Tara Tara Inter El Contra	1	

=>Select "Browse" button.

ld Reference			E
NET Projects			
Component Name	Version	Path	Browse
ICPDAS	1.0.1601.29	C:\Program Files\Microsoft Vi	S <u>e</u> lect
ICPDAS Dcon Remote I-7000/	1.0.4.0	C:\Program Files\Microsoft Vi	
ICPDAS UART for serial port	1.0.1.0	C:\Program Files\Microsoft Vi	=
ICPDAS Woon Inside I-8000/I	1.0.5.5	C:\Program Files\Microsoft Vi	
Microsoft.VisualBasic	7.0.5000.0	C:\Program Files\Microsoft Vi	
Microsoft.WindowsCE.Forms	1.0.5000.0	C:\Program Files\Microsoft Vi 👕	
Modbus	0.0.0.0	C:\Program Files\Microsoft Vi	
mscorlib	1.0.5000.0	C:\Program Files\Microsoft Vi	
ScanKernelNet	0.0.0.0	C:\Program Files\Microsoft V1	
System	1.0.5000.0	C:\Program Files\Microsoft V1	
System.Data	1.0.5000.0	C: Program Files Microsoft Vi	Image: A state of the state
I AVRIEM LIAIS LOMMON		F TENDORSM HILESUMICINISTI UI	
ected Components: Component Name	Туре	Source	Remove
094 NET All	File	C.\DAOPm\Wincon\ NET\i809	
		OK Canc	el Help

Select the i8902 _NET.DLL

Select Comp	onent		×
Look in:	DET	💌 🄄 🔹 🔍 🗙 🎦 🎹 + Too <u>l</u> s+	
History My Projects Desktop	DCONCE.dll 3092_NET.dll 318094.dll 318094_NET.dll 31CPDAS.dll 3ScanKernelNet.dll 3Serial.dll 3TModBus.dll 3Wincon.dll		
Favorites			
	File <u>n</u> ame:	▼ <u>O</u> pen	
	Files of type: Component I	Files (*.dll) Cancel	

Select the "Open" button, as above snapshot:

ødemo_First - M	icrosoft Visual Basic .NET [design] - Form1.vb [De	sign]*	
<u>File E</u> dit <u>V</u> iew <u>P</u> roject	<u>Build D</u> ebug D <u>a</u> ta F <u>o</u> rmat <u>T</u> ools <u>W</u> indow <u>H</u> elp		
🕼 • 🛅 • 🖨 🔛 🕼	🐰 🖻 💼 🗠 - 🖙 - 💭 - 🖳 🖡 🖡 Release 🔹 - 🍅 AlO		- 😡 🗗 🐋 🛠 😏
	표 등 밖 한 명 ~~ 봤 밖 밖 을 찾 왕 학 @ 图 🐂 🖷	Windows CE .NET Device	• 🕒 💭 🐥
Toolbox 🛛 🕂 🗙	Object Browser Form1.vb [Design]*	Solution Explorer - demo_F.	irst 🛛 🕹 🕹
装置控制項 🔺			
Pointer	💀 Form 1	Solution 'demo First' (1	project)
A Label		🖃 🕎 demo_First	
ab Button		🖃 📾 References	
abl TextBox		• 18092_NE	Г.
🛓 MainMenu		• System.Dat	a
CheckBox		• System.Dra	wing
RadioButton		• System. Wi	ndows.Forms
🔜 PictureBox	•••••••••••••••••••••••••••••••••••••••	• System AM	IL.
Panel		- Million AssemblyInfo.	∕b
🕤 DataGrid	•••••••••••••••••••••••••••••••••••••••	📰 Form1.vb	
📑 ListBox			
📰 ComboBox		Properties	πх
BB2 29 ListView	······	Form1 Syntam Windows F	ormo Form
ĕ≓ TreeView	Button1	Formi System windows.	
📩 TabControl	· · · · · · · · · · · · · · · · · · ·	2↓ □ □	
∎∎ HScrollBar			PMingLiU, 9pt 🔺
VScrollBar		FormBorderStyle	FixedSingle
🚫 Timer		Text	Form1
📳 DomainUpDown		- Benavior	(2022)
🔢 NumericUpDown		Enabled	True
📇 TrackBar	Ծարա 4	× □ Design	
Clipboard Ring 🗸 🗸		(Name)	Form1
General		DrawGrid	True
📲 Server 🛠 Toolb	🖉 Task List 📃 Output	⊞ GridSize	8, 8 💆
Ready			1

4.2.4 Start the VB.NET/C# Sample

Add a "BUTTON" on the Form1, then double-click the BUTTON, then it will appear a code of Form1.vb, then add the "imports i8092MF_NET" in top, as below snapshot:



Add the "i8092MF" into the Button1_Click, then it will appear a windows guide to help user to select a relevance function.



Detailed code as below:

'===='Step 1 Driver init

icpdas.com

If Not Driver_Open Then

i8092MF.i8092MF_REGISTRATION(cardNo, 1)

Driver_Open = True

End If

'===='Step 2 CONFIG IO

i8092MF.i8092MF_RESET_CARD(cardNo)

i8092MF.i8092MF_SET_PULSE_MODE(cardNo, AXIS_XYZU, 2) 'set the pulse output mode i8092MF.i8092MF_SET_ALARM(cardNo, AXIS_XYZU, 0, 0) 'disable the SERVO ALARM Input i8092MF.i8092MF_SET_ENCODER(cardNo, AXIS_XYZU, 0, 0, 0) 'set the encoder input type i8092MF.i8092MF_SET_MAX_V(cardNo, AXIS_XYZU, Convert.ToUInt32(16000)) 'set the max speed for XYZU i8092MF.i8092MF_EXD_DISABLE(cardNo, AXIS_XYZU) 'set the external input Off i8092MF.i8092MF_SET_LP(cardNo, AXIS_XYZU, 0) 'set the Logic position =0 i8092MF.i8092MF_SET_EP(cardNo, AXIS_XYZU, 0) 'set the Encoger position =0 i8092MF.i8092MF_SET_A(cardNo, AXIS_XYZU, Convert.ToUInt32(1000)) 'set the Acc =1000 i8092MF.i8092MF_SERVO_ON(cardNo, AXIS_XYZU) 'set the Servo_ON to servo motors '======'Step 3 Check ERROR

Dim KK As Long = 0

KK = i8092MF.i8092MF_GET_ERROR(cardNo)

Dim MSGG As String

If (KK <> YES) Then

'No ERROR: Step 4 Move X axis

Dim axis As UInt16 = AXIS_X 'for AXIS_X it can be to AXIS_XYZU

i8092MF.i8092MF_SET_MAX_V(cardNo, axis, Convert.ToUInt32(20000))

i8092MF.i8092MF_NORMAL_SPEED(cardNo, axis, Convert.ToUInt16(0)) 'set axis as Symmetrical T curve

mode

i8092MF.i8092MF_SET_V(cardNo, axis, Convert.ToUInt32(20000)) 'set v=10000 PPS i8092MF.i8092MF_SET_A(cardNo, axis, Convert.ToUInt32(100000)) 'set acc=100000 PPS/S i8092MF.i8092MF_SET_SV(cardNo, axis, Convert.ToUInt32(10)) 'set start speed=1000 PPS i8092MF.i8092MF_SET_AO(cardNo, axis, 0) 'set offset pulse (at SV speed)= 0 PS i8092MF.i8092MF_FIXED_MOVE(cardNo, axis, 10000) 'run the fixed 10000 Pulse move. Do While (i8092MF_i8092MF_STOP_WAIT(cardNo, axis) = NO) i8092MF.system.DoEvents() System.Threading.Thread.Sleep(1) 'wait for axis to stop

Loop

Dim AA As Long = i8092MF.i8092MF_GET_LP(cardNo, axis) 'Get X Now position

Else

'Please check the ERROR CODE

icpdas.com

'Get X ERROR CODE

KK = Convert.ToInt32(i8092MF.i8092MF_GET_ERROR_CODE(cardNo, AXIS_X)) 'Get Y ERROR CODE KK = Convert.ToInt32(i8092MF.i8092MF_GET_ERROR_CODE(cardNo, AXIS_Y)) 'Get Z ERROR CODE KK = Convert.ToInt32(i8092MF.i8092MF_GET_ERROR_CODE(cardNo, AXIS_Z)) 'Get U ERROR CODE KK = Convert.ToInt32(i8092MF.i8092MF_GET_ERROR_CODE(cardNo, AXIS_U)) '_____

End If

Please refer to a example " demo_First"

4.2.5 Build the Project

Please select the "Build" -> "Build Solution" in pull-down menu, then you will be finished this example program if there isn't any wrong.

🦚 Demo_First - M	icro	osoft Visual Basic .l	NET [design] - Form1.vb			
File Edit Yiew Project Image: Constraint of the state of the sta		Id Debug Tools Window Build Solution Ctrl+Shift+B Rebuild Solution Deploy Solution Build Demo_First Rebuild Demo_First Deploy Demo_First Build Cab File Batch Build Configuration Manager	Help Debug Volto Design] Form1.vb Forms.Form 1 As System.Windows.Forms.Button			
		'// Axis define Public AXIS_X As UInt16 Public AXIS_Y As UInt16 Public AXIS_Z As UInt16 Public AXIS_U As UInt16 Public AXIS_XY As UInt1 Public AXIS_XZ As UInt1	5 = Convert.ToUInt16(&H1) 5 = Convert.ToUInt16(&H2) 5 = Convert.ToUInt16(&H4) 5 = Convert.ToUInt16(&H8) 6 = Convert.ToUInt16(&H3) 6 = Convert.ToUInt16(&H5)			

4.2.6 Download and Run

Please copy the "Demo_First.exe", "I8092.dll" and "I8092_NET.dll" into the same floder of WinCon (User can use the VS.NET Online Download/FTP/USB disk to do), then execute it.

4.3 I-8000 Turbo C Guideine

4.3.1 Confirm the Relative Files

Please confirm you have the following relevance files:

18092.lib

l8092.h

18000.lib

l8000.h

If you don't have, please look for CD or download the latest edition from ICPDAS's website <u>http://www.icpdas.com/download/download-list.htm</u>

4.3.2 Create a new TC ++ Application Project

1. Execute the TC.EXE in the demo100 folder, then create a new Project(demo100.prj).



2. Add the contents of project : demo100.cpp and ..\lib\8000l.lib , I8092.lib

	File n 8000L DEM010 18092.	ame 11 B 0.CPI LIB	, L	00ati -NLU -NI80	ion 3 39 2	= Pr	oject:	DEMO100	L	ines n∕a 227 n∕a	Code n∕a 2014 n∕a		=1=[1] Data n/a 677 n/a	
F1	Help	Ins	Add	Del	Delete	^0	Options	Space	Includes	: 4-1	Edit	F10	Menu	_

- 3. Setting the relevance option
 - Compiler -> Code Generation item as below :



Compiler -> Advance Code Generation item as below :



Debugger setting as below, close the Source debugging.



4.3.3 Start the TC Sample

```
1. Add the declared contents into the demo100.cpp:

#include <dos.h>

#include <math.h>

#include "8000.h"

#include "18092.h"

BYTE cardNo;

long x_value, y_value, z_value, u_value;
```

2. Add the relevance code into the main program(Please refer to demo100.cpp):

```
void main ()
{
    //Set (slot0~slot7) = cardNO (1~8) •
    BYTE slot;
    int Found = 0;
    for (slot = 0; slot < 8; slot++)</pre>
    {
        cardNo = slot + 1;
        if (i8092MF_REGISTRATION(cardNo, slot) == YES)
        {
             //Found Axis Card •
             i8092MF_RESET_CARD(cardNo);
             Found++;
        }
    }
    if (Found == 0)
    {
    //Not Found •
        Print("I-8092 card not found ! \r\n");
    return;
    }
    cardNo = 1;
    i8092MF_INIT_CARD(cardNo);
    i8092MF_SET_PULSE_MODE(cardNo, AXIS_XYZU, 2);
```

```
i8094_IN3_LEVEL(cardNo,AXIS_XYZU, 1);
i8092MF_SET_ALARM(cardNo, AXIS_XYZU, 1, 1);
i8092MF_SET_ENCODER(cardNo, AXIS_XYZU, 0, 0, 0);
i8092MF_SET_MAX_V(cardNo, AXIS_XYZU, 16000);
```

```
BYTE ret1 = 0;
   BYTE chkey;
   DWORD sv; //PPS
   DWORD v; //PPS
   DWORD a: //PPS/s
   i8092MF_SERVO_ON(cardNo, AXIS_XYZU);
   do
   {
       Print(" (0:Exit, 1:HELIX_3D_1, 2:HELIX_3D_2, 3:RATIO, 4:FRnet output, 5:FRnet input)
\r\n");
       Print(" (6:Reset Encoder, 7:Stop, 8:Clear Error) \r\n");
       Print(" (X:Jog X, Y:Jog Y, Z:Jog Z, U:Jog U, S:Stop Jog) \r\n");
       Print("\n");
       x_value = i8092MF_GET_LP(cardNo, AXIS_X);
       y_value = i8092MF_GET_LP(cardNo, AXIS_Y);
       z value = i8092MF GET LP(cardNo, AXIS Z);
       u_value = i8092MF_GET_LP(cardNo, AXIS_U);
       Print("LOGIC POSITION: x=\%10ld, y=\%10ld, z=\%10ld, u=\%10ld r, x value, y value,
z value, u value);
       x_value = i8092MF_GET_EP(cardNo, AXIS_X);
       y value = i8092MF GET EP(cardNo, AXIS Y);
       z_value = i8092MF_GET_EP(cardNo, AXIS_Z);
       u_value = i8092MF_GET_EP(cardNo, AXIS_U);
       Print("REAL POSITION: x=%10ld, y= %10ld, z= %10ld, u=%10ld \r\n", x_value, y_value,
```

z_value, u_value);

while (!Kbhit()); chkey=Getch(); Print("%s\r\n",&chkey); switch (chkey) {

case '0': i8092MF_RESET_CARD(cardNo); Print("EXIT! \r\n"); return; //----case '1': v=50000;//PPS • i8092MF_SET_MAX_V(cardNo, AXIS_XYZU,160000L); ret1=i8092MF_HELIX_3D(cardNo, AXIS_Y, AXIS_Z, AXIS_X, 1, v, 0, 1000, 5, -2000); Delay(1000); Print("HELIX_3D_1 ! \r\n"); Print("ret1= %d \r\n",ret1); break; //----case '2': v=100000;//PPS • i8092MF_SET_MAX_V(cardNo, AXIS_XYZU,1600000L); ret1=i8092MF_HELIX_3D(cardNo, AXIS_Y, AXIS_Z, AXIS_U, 1, v, 0, 25000, 10, 3600); Delay(2000); Print("HELIX_3D_2 ! \r\n"); Print("ret1= %d \r\n",ret1); break; //----case '3': sv=300;//PPS · v=30000;//PPS • a=500000;//PPS/s • int loop1; int loop2; float ratio; i8092MF_SET_MAX_V(cardNo, AXIS_XYZU,160000L); Print("RATIO_2D ratio ? \r\n"); Scanf("%f", &ratio); Print("ratio= %f \r\n",ratio); i8092MF_RATIO_INITIAL(cardNo,AXIS_U, AXIS_X, sv, v, a, ratio); for (loop2 = 0; loop2 < 5; loop2++)</pre>

```
{
    for (loop1 = 0; loop1 < 5; loop1++)
    {
        i8092MF_RATIO_2D(cardNo, 0, 3600, 0);
        i8092MF_RATIO_2D(cardNo, 0, 3600, 1);
    }
    i8092MF_RATIO_2D(cardNo, 0, 7200, 0);
    i8092MF_RATIO_2D(cardNo, 0, 3600, 1);
}
i8092MF_RATIO_2D(cardNo, 1, 7200, 1);
Delay(3000);
Print("RATIO_2D OK ! \r\n");
break;
</pre>
```

```
case '4':
```

//--

```
WORD wSA;
WORD data;
Print("FRnet wSA ? \r\n");
Scanf("%d", &wSA);
Print("FRnet 16 bits data ? \r\n");
Scanf("%d", &data);
i8092MF_FRNET_SA(cardNo, wSA, data);
break;
```

case '5':

//-----

WORD wRA; Print("FRnet wRA ? \r\n"); Scanf("%d", &wRA); long data1 = i8092MF_FRNET_RA(cardNo, wRA); Print("FRnet 16 bits data = %10ld \r\n", data1); break;

//-----

case '6':

//---

i8092MF_SET_LP(cardNo, AXIS_XYZU, 0); i8092MF_SET_EP(cardNo, AXIS_XYZU, 0); Print("RESET Encoder ! \r\n"); break;

	case '7':		
		i8092MF_STOP_SLOWLY(cardNo, AXIS_XYZU);	
		Print("STOP! \r\n");	
		break;	
	//		
	case '8':		
		i8092MF_CLEAR_ERROR(cardNo);	
		Print("CLEAR ERROR ! \r\n");	
		break;	
	//		
	case 88:		
	case 120:		
		BYTE m_Axis=AXIS_X;	
		i8092MF_SET_MAX_V(cardNo, m_Axis, 32000);	
		i8092MF_NORMAL_SPEED(cardNo, m_Axis, 0);	//set axis as
Symmetrical T	curve mode		
		i8092MF_SET_A(cardNo, m_Axis, 50000);	//set Acc =50000
PPS/S			
		i8092MF_SET_V(cardNo, m_Axis, 50000);	
		i8092MF_EXD_MP(cardNo, AXIS_X, 100);	
		i8092MF_EXD_DISABLE(cardNo, AXIS_Y);	
		i8092MF_EXD_DISABLE(cardNo, AXIS_Z);	
		i8092MF_EXD_DISABLE(cardNo, AXIS_U);	
		break;	
	//		
	case 89:		
	case 121:		
		m_Axis=AXIS_Y;	
		i8092MF_SET_MAX_V(cardNo, m_Axis, 32000);	
		i8092MF_NORMAL_SPEED(cardNo, m_Axis, 0);	//set axis as
Symmetrical T	curve mode		
		i8092MF_SET_A(cardNo, m_Axis, 50000);	//set Acc =50000
PPS/S			
		i8092MF_SET_V(cardNo, m_Axis, 100000);	
		i8092MF_EXD_MP(cardNo, AXIS_Y, 100);	
		i8092MF_EXD_DISABLE(cardNo, AXIS_X);	
		i8092MF_EXD_DISABLE(cardNo, AXIS_Z);	
		i8092MF_EXD_DISABLE(cardNo, AXIS_U);	

break; //----case 90: case 122: m_Axis=AXIS_Z; i8092MF SET MAX V(cardNo, m Axis, 32000); i8092MF_NORMAL_SPEED(cardNo, m_Axis, 0); //set axis as Symmetrical T curve mode i8092MF_SET_A(cardNo, m_Axis, 50000); //set Acc =50000 PPS/S i8092MF SET V(cardNo, m Axis, 10000); i8092MF_EXD_MP(cardNo, AXIS_Z, 100); i8092MF_EXD_DISABLE(cardNo, AXIS_X); i8092MF_EXD_DISABLE(cardNo, AXIS_Y); i8092MF_EXD_DISABLE(cardNo, AXIS_U); break; //---------case 85: case 117: m Axis=AXIS U; i8092MF_SET_MAX_V(cardNo, m_Axis, 32000); i8092MF_NORMAL_SPEED(cardNo, m_Axis, 0); //set axis as Symmetrical T curve mode i8092MF_SET_A(cardNo, m_Axis, 50000); //set Acc =50000 PPS/S i8092MF_SET_V(cardNo, m_Axis, 10000); i8092MF_EXD_MP(cardNo, AXIS_U, 5); i8092MF EXD DISABLE(cardNo, AXIS X); i8092MF_EXD_DISABLE(cardNo, AXIS_Y); i8092MF_EXD_DISABLE(cardNo, AXIS_Z); break; //----case 83: case 115: i8092MF_EXD_DISABLE(cardNo, AXIS_X); i8092MF_EXD_DISABLE(cardNo, AXIS_Y); i8092MF_EXD_DISABLE(cardNo, AXIS_Z); i8092MF_EXD_DISABLE(cardNo, AXIS_U);

```
break;
//-----
default:
break;
}
} while (1);
```

}

4.3.4 Build the Project

Click F9 to compile program, LINK or demo100.EXE .

EXE file :\\TCPP\DEM0100.EXE Linking : \TCPP\LIB\CL.LIB						
Total Link Lines compiled: O PASS 2 Warnings: O O Errors: O O						
Available memory:	1928K					
Success :	Press	any key				

4.3.6 Download and Run

- 1. Please execute the "7188.EXE" on computer (The "7188.EXE" is a executed file of DOS, it can be used in DOS or DOS BOX of Win9X/WINNT/WIN2K).
- Please depend on actual wiring "COM PORT" that assign to "COM1(ALT_1)" or "COM2(ALT_2)" and set the transmission speed to "115200,N,8,1".
- 3. Turn on the power of I-8000. It will have two situation :
 - $_{\odot}~$ It will appear a version of MiniOs7 message if the " INIT*" connected to " INIT*COM", then appear I-8000> $_{\odot}~$
 - $_{\odot}$ The I-8000 will run the "AUTOEXEC.BAT" if the "INIT*" unconnected, then appear I-8000> $_{\odot}$
- 4. User can start to make a command of I-8000 after appearing the "I-8000>", as below drawing:



5. Press the F2 button on the keyboard, then key in "demo100.exe", then press the F10 button to download and execute demo100.exe, as following drawing:



Please refer to the 7188 getting started manual.

APPENDIX-A Setup Tools & Others

A.1 Setup the Development Environment of I8094

A.1.1 eVC ++ 4.0

- Microsoft eVC++ 4.0: at least ServicPack2 (Have already got at present ServicPack4)
- WinCon8000_EVC4_SP1: WinCon in eVC++ Development Environment (SA_IA)
- 3. WinConSDK:WinCon Software Tool(inc,lib,dll,demo...)

A.1.2 Visual Studio .NET 2003(VB.NET , C#)

- Above Microsoft Visual Studio.NET 2003 professional, including a SmartDeviceApplication item
- 2. Debug Tool: Windows CE .NET Utilities v1.1 for Visual Studio .NET 2003
- 3. WinConSDK:WinCon software Tool(inc,lib,dll,demo...)

A.1.3 Turbo C

1. Above boland Turbo C 2.0

A.2 I8094 Surface





I-8092F motion module



WinCon W-8331, I-8092F and DN-8237

A.3 Dimensions



A.4 The Version Upgrades Note

New i8092.DLL New i8092.h New i8092_NET.DLL

APPENDIX B Other Terminal Boards

B.1 DN-8237-DB Daughter Board

The DN-8237DB is the daughter board for Delta ASDA-A Series Ampilifier. It has 2-axis I/O signals.

B.1.1 Board Layout for DN-8237-DB



Fig. 1-1 Board layout for the DN-8237-DB

B.1.2 Signal Connections for DN-8237-DB

Maintaining signal connections is one of the most important factors in ensuring that your application system is sending and receiving data correctly.

Pin Assignment for CON1

The I/O connector on the DN-8237-DB is a 37-pin connector that enables you to connect to the PISO-PS200(or I-8092F) motion card. Fig. 1-2 shows the pin assignment for the 37-pin I/O connector on the DN-8237-DB (or on the motion card), and refer to Table 1-2 for description of each motion I/O signal.

		12.00	\bigcap
FR A		19	
FR B		37	
X ECA	2. 2.	18	
YECA	5	36	
X ECB		17	
V ECB		35	
V STOP2		16	
X_STOP2		34	
1_310F2		15	
X_INPOS		33	
T_INPOS		14	
X_ALARM		32	
Y_ALARM	02	13	
X_LMTP		31	$-\circ$
Y_LMTP		12	-
X_LMTM		30	
Y_LMTM		11	
X_STOP0		20	<u>+</u> 0
Y_STOP0		29	
X_STOP1	-	10	
Y STOP1		28	
X IN3		9	
Y IN3		21	
X EXPP		8	
Y EXPP		26	
X EXPM		7	
Y EXPM		25	
	2 2	6	
		24	
		5	
		23	
		4	
X_PP		22	
Y_PP		3	
X_PM		21	
Y_PM		2	
EMGN		20	
vcc		1	
GND			
	<u> </u>		

Fig. 1-2 I/O connector pin assignment for the CON1

Pin name	Pin number	Description			
FR_A	19	FRnet A-phase signal			
FR_B	37	FRnet B-phase signal			
X_ECA	18	Encoder A-phase signal for the X axis			
Y_ECA	36	Encoder A-phase signal for the Y axis			
X_ECB	17	Encoder B-Phase signal for the X axis			
Y_ECB	35	Encoder B-Phase signal for the Y axis			
X_STOP2	16	Stop 2 signal for the X axis			
Y_STOP2	34	Stop 2 signal for the Y axis			
X_INPOS	15	In-position signal for the X axis			
Y_INPOS	33	In-position signal for the Y axis			
X_ALARM	14	Alarm signal for the X axis			
Y_ALARM	32	Alarm signal for the Y axis			
X_LMTP	13	Limit switch input signal (+) for the X axis			
Y_LMTP	31	Limit switch input signal (+) for the Y axis			
X_LMTM	12	Limit switch input signal (-) for the X axis			
Y_LMTM	30	Limit switch input signal (-) for the Y axis			
X_STOP0	11	Stop 0 signal for the X axis			
Y_STOP0	29	Stop 0 signal for the Y axis			
X_STOP1	10	Stop 1 signal for the X axis			
Y_STOP1	28	Stop 1 signal for the Y axis			
X_IN3	9	Input 3 signal for the X axis			
Y_IN3	27	Input 3 signal for the Y axis			
X_EXPP	8	EXT pulsar input signal (+) for the X axis			
Y_EXPP	26	EXT pulsar input signal (+) for the Y axis			
X_EXPM	7	EXT pulsar input signal (-) for the X axis			
Y_EXPM	25	EXT pulsar input signal (-) for the Y axis			
X_OUT0	6	Output 0 signal for the X axis			
Y_OUT0	24	Output 0 signal for the Y axis			
X_OUT1	5	Output 1 signal for the X axis			
Y_OUT1	23	Output 1 signal for the Y axis			
XPP	4	Driving pulsar signal (+) for the X axis			
YPP	22	Driving pulsar signal (+) for the Y axis			
XPM	3	Driving pulsar signal (+) for the X axis			
YPM	21	Driving pulsar signal (+) for the Y axis			
EMGN	2	Emergency stop input signal			
VCC	20	Module power (+5V)			
GND	1	Ground			

Table 1-2 DN-8237-DB CON1 I/O connector signal description

The connector TB1 is 7-pin connector that enables you to connect to the signals of your motor drivers. Fig.1-3 shows the pin assignment for the 7-pin connector on the DN-8237-DB, and the Table 1-3 shows its I/O connector signal description.



Fig. 1-3 Pin definition for TB1

Name	Description
FR-A	FRnet port A
FR-B	FRnet port B
XOUT0	General Output 0 for X axis
YOUT0	General Output 0 for Y axis
E-PLS	EXT pulse signal
EMG-A	EMG input signal for all axes
E-GND	EXT power ground

Table 1-3 TB1 Signal Connection

TB2

The connector TB2 is 5-pin connector that enables you to connect to the signals of your motor drivers. Fig.1-4 shows the pin assignment for the 5-pin connector on the DN-8237-DB, and the Table 1-4 shows its I/O connector signal description.



Fig. 1-4 Pin definition for TB2

Table 1-4 TB2 Signal Connection

Pin name	Description	
E-PWR	EXT power supply +24V	
E-GND	EXT power ground	
FGND	Frame ground	

▶ Note: Don't reverse connect signals with E_PWR and E_GND. Serious damage to your motion card and motion controller might be happened.

■ CN-X & CN-Y (CN1 connector for each AXIS in Driver)

The connectors CN-X and CN-Y are 50-pin connectors that enable you to connect to the CN1 connector of Delta ASDA-A series motor drivers. Fig.1-5 shows the pin assignment for the 50-pin connector on the DN-8468-DB, and the Table 1-5 shows its I/O connector signal description.

	Name	Number	Description
1 26	A+	21	Encoder A-Phase (+)
$NC = \frac{1}{2} = 0.0 = \frac{20}{27} = NC$	A-	22	Encoder A-Phase (-)
P- 3 0 0 28 NC	B+	48	Encoder B-Phase (+)
P+ 4 0 0 29 SVON	В-	49	Encoder B-Phase (-)
	Z+	23	Encoder Z-Phase (+)
E-PWR 7 00 32 NC	Z-	24	Encoder Z-Phase (-)
E-GND 8 0 0 34 E-GND	P+	4	Positive Direction Pulse Output(+)
NC 10 0 35 BDY	P-	3	Positive Direction Pulse Output(-)
NC 11 0 0 36 E-GND	N+	6	Negative Direction Pulse
NC 12 00 37 ALARM	N-	5	Negative Direction Pulse Output(-)
E-GND 14 0 0 39 INP	INP	39	Servo In Position
E-GND 15 0 0 40 NC	RDY	35	Servo Ready
NC $\frac{10}{17}$ 0 0 $\frac{41}{42}$ E-GND	SVON	29	Servo On
E-GND 18 0 0 43 NC	A-CLR	31	Alarm Clear
NC $\frac{19}{20}$ O O $\frac{44}{45}$ NC	ALARM	37	Servo Alarm
NC 20 00 45 NC	E-PWR	7	EXT power +24V
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	E-GND	8, 9, 13, 15,17, 25, 33,34, 36, 38,41	EXT power ground
Fig. 1-5 Pin definition for CNX and CNY	NC	1,2,10,11, 12,14,16, 18,19,20, 26,27,28, 30,32,40, 42,43,44, 45,46,47, 50	No connection

Table 1-5 CN1 Signal Connection

Note 1: Don't connect NC (not connected) signals. Connecting these signals could cause permanent damage to your motion controller.
The connectors CN1 and CN2 are 11-pin connectors that enable you to connect to the signals of your motor drivers. Fig.1-6 shows the pin assignment for the 20-pin connector on the DN-8237-DB, and the Table 1-6 shows its I/O connector signal description.



Fig. 1-6 Pin definition for CN1 & CN2

Pin name	Description
E-PWR	EXT power supply +24V
EMG	EMG input signal
LMT+	Limit Switch Input Signal (+)
LMT-	Limit Switch Input Signal (-)
INPUT3	Input Signal (IN3)
NRHOME	Near Home Sensor Input Signal
HOME	Home Sensor Input Signal
RESET	Reset input signal
EXP+	EXT Positive Direction Pulse (+)
EXP-	EXT Negative Direction Pulse (-)
E-GND	EXT power ground

Table 1-6 CN1 & CN2 Signal Connection

■ RJ1 (The I/O signals of the FRnet)

The connectors RJ1 is an 8-pin RJ45 connector that enable you to connect to the signals of FRnet. Fig.1-7 shows the pin assignment for the 8-pin connector on the DN-8237-DB, and the Table 1-7 shows its I/O connector signal description.



Table 1-7 RJ1		
Pin name Descriptio		
FRnetA	FRnet port A	
FRnetB	FRnet port B	
NC	No connection	

Fig. 1-7 Pin definition for RJ1

Note: Don't connect NC (not connected) signals. Connecting these signals could cause permanent damage to your motion controller.

B.1.3 Jumper and Switch Settings

■ JP5

Jumper 5 controls the EMG-A signal of the TB1 connector. The following diagram is shown the selection condition of the jumper 5.



Fig. 1-8 Jumper 5 setting

■ SW 1

The emergency stop signal for each servo ampilfier can be selected from SW1. The number 1 and 2 on SW1 are denoted as axis X and Y, respectively. The number 3 and 4 on SW1 are reserved for future work. Fig. 1-9 is the default setting to connect the EMG singals to GND. The EMG signals from CN1 and CN2 will not take effect. If the switch is disconnected as shown in Fig. 1-10, the emergency stop signals can be controlled from EMG signals in CN1 and CN2.



Fig. 1-9 SW1 setting for normally GND (Default setting)



Fig. 1-10 SW1 setting for user controlled signals.

Jumper 1 ~ Jumper 2 can select the reset function in CN1 and CN2 for each axis. The following diagram is shown the selection condition of the JP1.



Fig. 1-15 JP 1 and 2 setting

B.2 DN-8237-MB Daughter Board

The DN-8237MB is the daughter board for Mitsubishi J2 Series Ampilifier. It has 2-axis I/O signals.

B.2.1 Board Layout for DN-8237-MB



Fig. 3-1 Board layout for the DN-8237-MB

B.2.2 Signal Connections for DN-8237-MB

Maintaining signal connections is one of the most important factors in ensuring that your application system is sending and receiving data correctly.

Pin Assignment for CON1

The I/O connector on the DN-8237-MB is a 37-pin connector that enables you to connect to the PISO-PS200(or I-8092F) motion card. Fig. 3-2 shows the pin assignment for the 37-pin I/O connector on the DN-8237-MB (or on the motion card), and refer to Table 3-2 for description of each motion I/O signal.

FR A	-	19	
FRB		37	
X ECA	2. 	18	
YECA	<i>a</i> .	36	
X ECB		17	
V ECB		35	
V STORA		16	
X_STOP2		34	
1_510P2	20 	15	
X_INPOS		33	
Y_INPOS	1	14	
X_ALARM		32	$-\circ$
Y_ALARM	10-	13	-
X_LMTP		31	-
Y_LMTP		12	
X_LMTM	<u></u>	20	-
Y LMTM		30	
X STOP0		11	
Y STOPO		29	
X STOP1		10	
Y STOP1		28	
X IN3		9	
V IN3		27	
		8	
A_EAPP		26	
Y_EXPP		7	
X_EXPM		25	
Y_EXPM		6	
X_OUT0		24	
Y_OUT0		5	
X_OUT1		23	-
Y_OUT1	3	4	-
X_PP	-	22	$-\circ$
Y_PP		22	-
X_PM		21	-
Y PM		21	
EMGN		2	
VCC		20	
GND		1	
GILD			0
	_		

Fig. 3-2 I/O connector pin assignment for the CON1

Pin name	Pin number	Description
FR_A	19	FRnet A-phase signal
FR_B	37	FRnet B-phase signal
X_ECA	18	Encoder A-phase signal for the X axis
Y_ECA	36	Encoder A-phase signal for the Y axis
X_ECB	17	Encoder B-Phase signal for the X axis
Y_ECB	35	Encoder B-Phase signal for the Y axis
X_STOP2	16	Stop 2 signal for the X axis
Y_STOP2	34	Stop 2 signal for the Y axis
X_INPOS	15	In-position signal for the X axis
Y_INPOS	33	In-position signal for the Y axis
X_ALARM	14	Alarm signal for the X axis
Y_ALARM	32	Alarm signal for the Y axis
X_LMTP	13	Limit switch input signal (+) for the X axis
Y_LMTP	31	Limit switch input signal (+) for the Y axis
X_LMTM	12	Limit switch input signal (-) for the X axis
Y_LMTM	30	Limit switch input signal (-) for the Y axis
X_STOP0	11	Stop 0 signal for the X axis
Y_STOP0	29	Stop 0 signal for the Y axis
X_STOP1	10	Stop 1 signal for the X axis
Y_STOP1	28	Stop 1 signal for the Y axis
X_IN3	9	Input 3 signal for the X axis
Y_IN3	27	Input 3 signal for the Y axis
X_EXPP	8	EXT pulsar input signal (+) for the X axis
Y_EXPP	26	EXT pulsar input signal (+) for the Y axis
X_EXPM	7	EXT pulsar input signal (-) for the X axis
Y_EXPM	25	EXT pulsar input signal (-) for the Y axis
X_OUT0	6	Output 0 signal for the X axis
Y_OUT0	24	Output 0 signal for the Y axis
X_OUT1	5	Output 1 signal for the X axis
Y_OUT1	23	Output 1 signal for the Y axis
XPP	4	Driving pulsar signal (+) for the X axis
YPP	22	Driving pulsar signal (+) for the Y axis
XPM	3	Driving pulsar signal (+) for the X axis
YPM	21	Driving pulsar signal (+) for the Y axis
EMGN	2	Emergency stop input signal
VCC	20	Module power (+5V)
GND	1	Ground

Table 3-2 DN-8237-MB CON1 I/O connector signal description

The connector TB1 is 7-pin connector that enables you to connect to the signals of your motor drivers. Fig.3-3 shows the pin assignment for the 7-pin connector on the DN-8237-MB, and the Table 3-3 shows its I/O connector signal description.



Fig. 3-3 Pin definition for TB1

Name	Description	
FR-A	FRnet port A	
FR-B	FRnet port B	
XOUT0	General Output 0 for X axis	
YOUT0	General Output 0 for Y axis	
E-PLS	EXT pulse signal	
EMG-A	EMG input signal for all axes	
E-GND	EXT power ground	

Table 3-3 TB1 Signal Connection

■ TB2

The connector TB2 is 5-pin connector that enables you to connect to the signals of your motor drivers. Fig.3-4 shows the pin assignment for the 5-pin connector on the DN-8237-MB, and the Table 3-4 shows its I/O connector signal description.



Fig. 3-4 Pin definition for TB2

Table 3-4 TB2 Signal Connection

Pin name	Description	
E-PWR	EXT power supply +24V	
E-GND	EXT power ground	
FGND	Frame ground	

▶ Note: Don't reverse connect signals with E_PWR and E_GND. Serious damage to your motion card and motion controller might be happened.

■ CN-XA & CN-YA (CNA connector for each AXIS)

The connectors CN-XA and CN-YA are 20-pin connectors that enable you to connect to the CNA connector of Mitsubishi motor drivers. Fig.3-5 shows the pin assignment for the 20-pin connector on the DN-8237-MB, and the Table 3-5 shows its I/O connector signal description.





		_
Name	Number	Description
A+	6	Encoder A-Phase (+)
A-	16	Encoder A-Phase (-)
B+	7	Encoder B-Phase (+)
В-	17	Encoder B-Phase (-)
Z+	5	Encoder Z-Phase (+)
Z-	15	Encoder Z-Phase (-)
P+	3	Positive Direction Pulse Output(+)
P-	13	Positive Direction Pulse Output(-)
N+	2	Negative Direction Pulse Output(+)
N-	12	Negative Direction Pulse Output(-)
INP	18	Servo In Position
RDY	19	Servo Ready
E-PWR	9	EXT power +24V
E-GND	1, 10, 20	EXT power ground
NC	4,8,11,14	No connection

Note 1: Don't connect NC (not connected) signals. Connecting these signals could cause permanent damage to your motion controller.

Table 3-5 CNA Signal Connection

CN-XB & CN-YB (CNB connector for each AXIS)

The connectors CN-XB and CN-YB are 20-pin connectors that enable you to connect to the CNB connector of your motor drivers. Fig.3-6 shows the pin assignment for the 20-pin connector on the DN-8237-MB, and the Table 3-6 shows its I/O connector signal description.



■ CN1 & CN2 (The I/O signals of the X and Y AXIS)

The connectors CN1 and CN2 are 11-pin connectors that enable you to connect to the signals of your motor drivers. Fig.3-7 shows the pin assignment for the 20-pin connector on the DN-8237-MB, and the Table 3-7 shows its I/O connector signal description.



Fig. 3-7 Pin definition for CN1 & CN2

Table 3-7 CN1 & CN2 Signal Connection

Pin name	Description
E-PWR	EXT power supply +24V
EMG	EMG input signal
LMT+	Limit Switch Input Signal (+)
LMT-	Limit Switch Input Signal (-)
INPUT3	Input Signal (IN3)
NRHOME	Near Home Sensor Input Signal
HOME	Home Sensor Input Signal
RESET	Reset input signal
EXP+	EXT Positive Direction Pulse (+)
EXP-	EXT Negative Direction Pulse (-)
E-GND	EXT power ground

■ RJ1 (The I/O signals of the FRnet)

The connectors RJ1 is an 8-pin RJ45 connector that enable you to connect to the signals of FRnet. Fig.3-8 shows the pin assignment for the 8-pin connector on the DN-8237-MB, and the Table 3-8 shows its I/O connector signal description.



Table 3-8 RJ1		
Pin name	Description	
FRnetA	FRnet port A	
FRnetB	FRnet port B	
NC	No connection	

Fig. 3-8 Pin definition for RJ1

Note: Don't connect NC (not connected) signals. Connecting these signals could cause permanent damage to your motion controller.

B.2.3 Jumper and Switch Settings

■ JP5

Jumper 5 controls the EMG-A signal of the TB1 connector. The following diagram is shown the selection condition of the jumper 5.



Fig. 3-9 Jumper 5 setting

■ SW 1

The emergency stop signal for each servo ampilfier can be selected from SW1. The number 1 and 2 on SW1 are denoted as axis X and Y, respectively. The number 3 and 4 on SW1 are reserved for future work. Fig. 3-10 is the default setting to connect the EMG singals to GND. The EMG signals from CN1 and CN2 will not take effect. If the switch is disconnected as shown in Fig. 3-11, the emergency stop signals can be controlled from EMG signals in CN1 and CN2.



Fig. 3-10 SW1 setting for normally GND (Default setting)



Fig. 3-11 SW1 setting for user controlled signals.

B.3 DN-8237-PB Daughter Board

The DN-8237PB is the daughter board for Panasonic A4 Series Ampilifier. It has 2-axis I/O signals.

B.3.1 Board Layout for DN-8237-PB



Fig. 3-1 Board layout for the DN-8237-PB

B.3.2 Signal Connections for DN-8237-PB

Maintaining signal connections is one of the most important factors in ensuring that your application system is sending and receiving data correctly.

Pin Assignment for CON1

The I/O connector on the DN-8237-PB is a 37-pin connector that enables you to connect to the PISO-PS200(or I-8092F) motion card. Fig. 3-2 shows the pin assignment for the 37-pin I/O connector on the DN-8237-PB (or on the motion card), and refer to Table 3-2 for description of each motion I/O signal.

FR A	-	19	
FRB		37	
X ECA	2. 	18	
YECA	<i>a</i> .	36	
X ECB		17	
V ECB		35	
V STORA		16	
X_STOP2		34	
1_510P2	20 	15	
X_INPOS		33	
Y_INPOS	1	14	
X_ALARM		32	$-\circ$
Y_ALARM	10-	13	-
X_LMTP		31	-
Y_LMTP		12	
X_LMTM	<u></u>	20	-
Y LMTM		30	
X STOP0		11	
Y STOPO		29	
X STOP1		10	
Y STOP1		28	
X IN3		9	
V IN3		27	
		8	
A_EAPP		26	
Y_EXPP		7	
X_EXPM		25	
Y_EXPM		6	
X_OUT0		24	
Y_OUT0		5	
X_OUT1		23	-
Y_OUT1	3	4	-
X_PP		22	-
Y_PP		22	-
X_PM		21	-
Y PM		21	
EMGN		2	
VCC		20	
GND		1	
GILD			0
	_		

Fig. 3-2 I/O connector pin assignment for the CON1

Pin name	Pin number	Description
FR_A	19	FRnet A-phase signal
FR_B	37	FRnet B-phase signal
X_ECA	18	Encoder A-phase signal for the X axis
Y_ECA	36	Encoder A-phase signal for the Y axis
X_ECB	17	Encoder B-Phase signal for the X axis
Y_ECB	35	Encoder B-Phase signal for the Y axis
X_STOP2	16	Stop 2 signal for the X axis
Y_STOP2	34	Stop 2 signal for the Y axis
X_INPOS	15	In-position signal for the X axis
Y_INPOS	33	In-position signal for the Y axis
X_ALARM	14	Alarm signal for the X axis
Y_ALARM	32	Alarm signal for the Y axis
X_LMTP	13	Limit switch input signal (+) for the X axis
Y_LMTP	31	Limit switch input signal (+) for the Y axis
X_LMTM	12	Limit switch input signal (-) for the X axis
Y_LMTM	30	Limit switch input signal (-) for the Y axis
X_STOP0	11	Stop 0 signal for the X axis
Y_STOP0	29	Stop 0 signal for the Y axis
X_STOP1	10	Stop 1 signal for the X axis
Y_STOP1	28	Stop 1 signal for the Y axis
X_IN3	9	Input 3 signal for the X axis
Y_IN3	27	Input 3 signal for the Y axis
X_EXPP	8	EXT pulsar input signal (+) for the X axis
Y_EXPP	26	EXT pulsar input signal (+) for the Y axis
X_EXPM	7	EXT pulsar input signal (-) for the X axis
Y_EXPM	25	EXT pulsar input signal (-) for the Y axis
X_OUT0	6	Output 0 signal for the X axis
Y_OUT0	24	Output 0 signal for the Y axis
X_OUT1	5	Output 1 signal for the X axis
Y_OUT1	23	Output 1 signal for the Y axis
XPP	4	Driving pulsar signal (+) for the X axis
YPP	22	Driving pulsar signal (+) for the Y axis
XPM	3	Driving pulsar signal (+) for the X axis
YPM	21	Driving pulsar signal (+) for the Y axis
EMGN	2	Emergency stop input signal
VCC	20	Module power (+5V)
GND	1	Ground

Table 3-2 DN-8237-PB CON1 I/O connector signal description

The connector TB1 is 7-pin connector that enables you to connect to the signals of your motor drivers. Fig.3-3 shows the pin assignment for the 7-pin connector on the DN-8237-PB, and the Table 3-3 shows its I/O connector signal description.



Fig. 3-3 Pin definition for TB1

Name	Description
FR-A	FRnet port A
FR-B	FRnet port B
XOUT0	General Output 0 for X axis
YOUT0	General Output 0 for Y axis
E-PLS	EXT pulse signal
EMG-A	EMG input signal for all axes
E-GND	EXT power ground

Table 3-3 TB1 Signal Connection

TB2

The connector TB2 is 5-pin connector that enables you to connect to the signals of your motor drivers. Fig.3-4 shows the pin assignment for the 5-pin connector on the DN-8237-PB, and the Table 3-4 shows its I/O connector signal description.



Fig. 3-4 Pin definition for TB2

Table 3-4 TB2 Signal Connection

Pin name	Description
E-PWR	EXT power supply +24V
E-GND	EXT power ground
FGND	Frame ground

▶ Note: Don't reverse connect signals with E_PWR and E_GND. Serious damage to your motion card and motion controller might be happened.

CN-X &CN-Y(CN X5 connector for each Axis in Driver)

The connectors CN-X and CN-Y are 50-pin connectors that enable you to connect to the CN X5 connector of Panasonic motor drivers. Fig.3-5 shows the pin assignment for the 50-pin connector on the DN-8468-PB, and the Table 3-5 shows its I/O connector signal description.



Name	Number	Description
A+	21	Encoder A-Phase (+)
A-	22	Encoder A-Phase (-)
B+	48	Encoder B-Phase (+)
В-	49	Encoder B-Phase (-)
Z+	23	Encoder Z-Phase (+)
Z-	24	Encoder Z-Phase (-)
P+	4	Positive Direction Pulse Output(+)
P-	3	Positive Direction Pulse Output(-)
N+	6	Negative Direction Pulse
N-	5	Negative Direction Pulse Output(-)
INP	39	Servo In Position
RDY	35	Servo Ready
SVON	29	Servo On
A-CLR	31	Alarm Clear
ALARM	37	Servo Alarm
E-PWR	7	EXT power +24V
E-GND	8, 9, 13, 15,17, 25, 33,34, 36, 38,41	EXT power ground
NC	1,2,10,11, 12,14,16, 18,19,20, 26,27,28, 30,32,40, 42,43,44, 45,46,47, 50	No connection

Table 3-5 CN X5 Signal Connection

Note 1: Don't connect NC (not connected) signals. Connecting these signals could cause permanent damage to your motion controller.

■ CN1& CN2 (The I/O signals of the X and Y axis)

The connectors CN1 and CN2 are 11-pin connectors that enable you to connect to the signals of

your motor drivers. Fig.3-6 shows the pin assignment for the 20-pin connector on the

DN-8237-PB, and the Table 3-6 shows its I/O connector signal description.



Fig. 3-6 Pin definition for CN1 & CN2

Table 3-6 CN1 & CN2 Signal Connection		
Pin name	Description	
E-PWR	EXT power supply +24V	
EMG	EMG input signal	
LMT+	Limit Switch Input Signal (+)	
LMT-	Limit Switch Input Signal (-)	
INPUT3	Input Signal (IN3)	
NRHOME	Near Home Sensor Input Signal	
HOME	Home Sensor Input Signal	
RESET	Reset input signal	
EXP+	EXT Positive Direction Pulse (+)	
EXP-	EXT Negative Direction Pulse (-)	
E-GND	EXT power ground	

■ RJ1 (The I/O signals of the FRnet)

The connectors RJ1 is an 8-pin RJ45 connector that enable you to connect to the signals of FRnet. Fig.3-7 shows the pin assignment for the 8-pin connector on the DN-8237-PB, and the Table 3-7 shows its I/O connector signal description.



Table 3-7 RJ1		
Pin name	Description	
FRnetA	FRnet port A	
FRnetB	FRnet port B	
NC	No connection	

Fig. 3-7 Pin definition for RJ1

Note: Don't connect NC (not connected) signals. Connecting these signals could cause permanent damage to your motion controller.

B.3.3 Jumper and Switch Settings

■ JP5

Jumper 5 controls the EMG-A signal of the TB1 connector. The following diagram is shown the selection condition of the jumper 5.



Fig. 3-8 Jumper 5 setting

The emergency stop signal for each servo ampilfier can be selected from SW1. The number 1 and 2 on SW1 are denoted as axis X and Y, respectively. The number 3 and 4 on SW1 are reserved for future work. Fig. 3-9 is the default setting to connect the EMG singals to GND. The EMG signals from CN1 and CN2 will not take effect. If the switch is disconnected as shown in Fig. 3-10, the emergency stop signals can be controlled from EMG signals in CN1 and CN2.



Fig. 3-9 SW1 setting for normally GND (Default setting)



Fig. 3-10 SW1 setting for user

B.4 DN-8237-YB Daughter Board

The DN-8237YB is the daughter board for Yaskawa Series Ampilifier. It has 2-axis I/O signals.

B.4.1 Board Layout for DN-8237-YB



Fig. 1-1 Board layout for the DN-8237-YB

B.4.2 Signal Connections for DN-8237-YB

Maintaining signal connections is one of the most important factors in ensuring that your application system is sending and receiving data correctly.

Pin Assignment for CON1

The I/O connector on the DN-8237-YB is a 37-pin connector that enables you to connect to the PISO-PS200(or I-8092F) motion card. Fig. 1-2 shows the pin assignment for the 37-pin I/O connector on the DN-8237-YB (or on the motion card), and refer to Table 1-2 for description of each motion I/O signal.

FR A		19	_	-0-
FRB		37	_	
X ECA	2	18	_	-0
YECA	<i>a</i> .	36		
X ECB		17		<u> </u>
V ECB	-	35		\sim
V STORA		16		\sim
X_STOP2		34		\sim
1_310F2		15		
X_INPOS		33		-0
T_INPOS		14		
X_ALARM	-	32		-0
Y_ALARM	10-	13	-	
X_LMTP	-	31	-	-0
Y_LMTP		12	-+	O
X_LMTM	4 <u>1</u>	30	-+	-0
Y_LMTM	-	11	-+	
X_STOP0		20	-+	-0
Y_STOP0		29	-+	
X_STOP1		10	-+	-0
Y STOP1		28	-+	
X_IN3		9	-+	-0
Y IN3		21	-	
X EXPP		8	-	-0
YEXPP		26	\rightarrow	
X EXPM	-	7	_	-0
Y EXPM		25		
X OUTO	22 22	6		-0
		24		0
	- 17 	5		\sim
		23		\sim
		4		
X_PP		22		-0
Y_PP		3		
X_PM		21		-0
Y_PM		2	-	O
EMGN	-	20	-	-0
vcc		1	-+	O
GND			-	-0_)
	<u> </u>		-	-

Fig. 1-2 I/O connector pin assignment for the CON1

Pin name	Pin number	Description
FR_A	19	FRnet A-phase signal
FR_B	37	FRnet B-phase signal
X_ECA	18	Encoder A-phase signal for the X axis
Y_ECA	36	Encoder A-phase signal for the Y axis
X_ECB	17	Encoder B-Phase signal for the X axis
Y_ECB	35	Encoder B-Phase signal for the Y axis
X_STOP2	16	Stop 2 signal for the X axis
Y_STOP2	34	Stop 2 signal for the Y axis
X_INPOS	15	In-position signal for the X axis
Y_INPOS	33	In-position signal for the Y axis
X_ALARM	14	Alarm signal for the X axis
Y_ALARM	32	Alarm signal for the Y axis
X_LMTP	13	Limit switch input signal (+) for the X axis
Y_LMTP	31	Limit switch input signal (+) for the Y axis
X_LMTM	12	Limit switch input signal (-) for the X axis
Y_LMTM	30	Limit switch input signal (-) for the Y axis
X_STOP0	11	Stop 0 signal for the X axis
Y_STOP0	29	Stop 0 signal for the Y axis
X_STOP1	10	Stop 1 signal for the X axis
Y_STOP1	28	Stop 1 signal for the Y axis
X_IN3	9	Input 3 signal for the X axis
Y_IN3	27	Input 3 signal for the Y axis
X_EXPP	8	EXT pulsar input signal (+) for the X axis
Y_EXPP	26	EXT pulsar input signal (+) for the Y axis
X_EXPM	7	EXT pulsar input signal (-) for the X axis
Y_EXPM	25	EXT pulsar input signal (-) for the Y axis
X_OUT0	6	Output 0 signal for the X axis
Y_OUT0	24	Output 0 signal for the Y axis
X_OUT1	5	Output 1 signal for the X axis
Y_OUT1	23	Output 1 signal for the Y axis
XPP	4	Driving pulsar signal (+) for the X axis
YPP	22	Driving pulsar signal (+) for the Y axis
XPM	3	Driving pulsar signal (+) for the X axis
YPM	21	Driving pulsar signal (+) for the Y axis
EMGN	2	Emergency stop input signal
VCC	20	Module power (+5V)
GND	1	Ground

Table 1-2 DN-8237-YB CON1 I/O connector signal description

The connector TB1 is 7-pin connector that enables you to connect to the signals of your motor drivers. Fig.1-3 shows the pin assignment for the 7-pin connector on the DN-8237-YB, and the Table 1-3 shows its I/O connector signal description.



Fig. 1-3 Pin definition for TB1

<u>_</u>		
Name	Description	
FR-A	FRnet port A	
FR-B	FRnet port B	
XOUT0	General Output 0 for X axis	
YOUT0	General Output 0 for Y axis	
E-PLS	EXT pulse signal	
EMG-A	EMG input signal for all axes	
E-GND	EXT power ground	

Table 1-3 TB1 Signal Connection

TB2

The connector TB2 is 5-pin connector that enables you to connect to the signals of your motor drivers. Fig.1-4 shows the pin assignment for the 5-pin connector on the DN-8237-YB, and the Table 1-4 shows its I/O connector signal description.

	TB2
E-PWR E-PWR E-GND E-GND FGND	1 2 3 4 5

Fig. 1-4 Pin definition for TB2

Table 1	-4 TB2	Signal	Connection
---------	--------	--------	------------

Pin name	Description
E-PWR	EXT power supply +24V
E-GND	EXT power ground
FGND	Frame ground

▶ Note: Don't reverse connect signals with E_PWR and E_GND. Serious damage to your motion card and motion controller might be happened.

(CN1 connector for each AXIS in Driver) CN-X & CN-Y

The connectors CN-X and CN-Y are 50-pin connectors that enable you to connect to the CN1 connector of Yaskawa motor drivers. Fig.1-5 shows the pin assignment for the 50-pin connector on the DN-8468-YB, and the Table 1-5 shows its I/O connector signal description.

E-GND 2 GND 3 NC 3 NC 5 NC 5 NC 5 NC 6 NC 5 NC 6 NC 6 NC 5 NC 6 NC 6 NC 6 NC 6 NC 6 NC 6 NC 6 NC 6 NC 7 0 NC 6 0 NC 5 0 0 NC 5 0 0 0 0 0 0 0 0 0 0 0 0 0	26 E-GND 27 NC 28 NC 29 RDY 30 31 ALARM 33 ALARM 33 ALARM 33 A+ 33 ALARM 333 NC 334 A+ 35 B- 36 NC 37 NC 38 NC 000000000000000000000000000000000000
---	---

Fig. 3-5 Pin definition for CNX, CNY, CNZ, CNU

	Table 3-6	CN1 Signal Connection
Name	Number	Description
A+	33	Encoder A-Phase (+)
A-	34	Encoder A-Phase (-)
B+	35	Encoder B-Phase (+)
В-	36	Encoder B-Phase (-)
Z+	19	Encoder Z-Phase (+)
Z-	20	Encoder Z-Phase (-)
P+	7	Positive Direction Pulse Output(+)
P-	8	Positive Direction Pulse Output(-)
N+	11	Negative Direction Pulse
N-	12	Negative Direction Pulse Output(-)
INP	25	Servo In Position
RDY	29	Servo Ready
SVON	40	Servo On
RESET	44	Parameter Reset
ALARM	31	Servo Alarm
E-PWR	47	EXT power +24∨
E-GND	1,2,6,10, 26, 30,32, 42,43	EXT power ground
NC	3,4,5,9, 13,14,15, 16,17,18, 21,22,23, 24,27,28, 37,38,39, 41,45,46, 48,49,50,	No connection

The connectors CN1 and CN2 are 11-pin connectors that enable you to connect to the signals of your motor drivers. Fig.1-6 shows the pin assignment for the 20-pin connector on the DN-8237-YB, and the Table 1-6 shows its I/O connector signal description.



Fig. 1-6 Pin definition for CN1 & CN2

Pin name	Description
E-PWR	EXT power supply +24V
EMG	EMG input signal
LMT+	Limit Switch Input Signal (+)
LMT-	Limit Switch Input Signal (-)
INPUT3	Input Signal (IN3)
NRHOME	Near Home Sensor Input Signal
HOME	Home Sensor Input Signal
RESET	Reset input signal
EXP+	EXT Positive Direction Pulse (+)
EXP-	EXT Negative Direction Pulse (-)
E-GND	EXT power ground

Table 1-6 CN1 & CN2 Signal Connection

■ RJ1 (The I/O signals of the FRnet)

The connectors RJ1 is an 8-pin RJ45 connector that enable you to connect to the signals of FRnet. Fig.1-7 shows the pin assignment for the 8-pin connector on the DN-8237-YB, and the Table 1-7 shows its I/O connector signal description.



Table 1-7 RJ1		
Pin name	Description	
FRnetA	FRnet port A	
FRnetB	FRnet port B	
NC	No connection	

Fig. 1-7 Pin definition for RJ1

Note: Don't connect NC (not connected) signals. Connecting these signals could cause permanent damage to your motion controller.

B.4.3 Jumper and Switch Settings

■ JP5

Jumper 5 controls the EMG-A signal of the TB1 connector. The following diagram is shown the selection condition of the jumper 5.



Fig. 1-8 Jumper 5 setting

■ SW 1

The emergency stop signal for each servo ampilfier can be selected from SW1. The number 1 and 2 on SW1 are denoted as axis X and Y, respectively. The number 3 and 4 on SW1 are reserved for future work. Fig. 1-9 is the default setting to connect the EMG singals to GND. The EMG signals from CN1 and CN2 will not take effect. If the switch is disconnected as shown in Fig. 1-10, the emergency stop signals can be controlled from EMG signals in CN1 and CN2.



Fig. 1-9 SW1 setting for normally GND (Default setting)



Fig. 1-10 SW1 setting for user controlled signals.