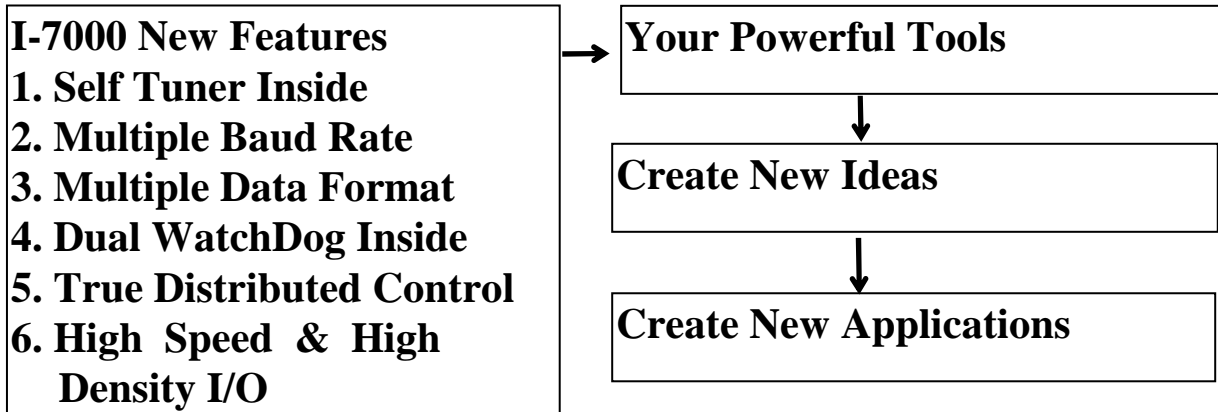


WDT-03

User's Manual



Warranty

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

Warning

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

Copyright

Copyright 2001 by ICP DAS. All rights are reserved.

Trademark

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

Table of Contents

1.	INTRODUCTION.....	4
1.1	PRODUCTS CHECK LIST	4
1.2	DEFAULT SETTING.....	5
1.3	INTRODUCTION.....	6
1.4	TRADITIONAL WDT CARDS.....	9
1.5	ADVANCE WDT SYSTEM	10
1.6	CONNECTORS	13
1.7	PIN ASSIGNMENT.....	14
1.8	SPECIFICATIONS.....	19
1.9	BLOCK DIAGRAM	20
1.10	QUICK START	21
2.	SOFTWARE UTILITY	29
2.1	CONTENT OF COMPANION CD	29
2.2	UTILITY SOFTWARE INSTALLATION	30
2.3	CONFIGURATION UTILITY	31
2.4	AUTO MONITORING UTILITY FOR WDT-03	36
2.5	HOW TO EXECUTE A PROGRAM WHEN WINDOWS STARTUP.....	43
3.	SOFTWARE DRIVER	44
3.1	DRIVER INSTALLATION & USER'S MANUAL	44
3.2	DOS DRIVER FOR HOST-PC	46
3.3	WINDOWS UTILITY FOR HOST-PC	47
3.4	WINDOWS DLLS FOR HOST-PC.....	49
3.5	WINDOWS DDE FOR HOST-PC.....	50
3.6	WINDOWS OCX FOR HOST-PC.....	51
3.7	WINDOWS OPC FOR HOST-PC	54
3.8	LINUX DRIVER FOR HOST-PC.....	56
3.9	LABVIEW DRIVER FOR HOST-PC.....	62
4.	COMMAND SET.....	63
4.1	%AANNTTCCFF.....	64
4.2	\$AA2	65
4.3	\$AA5	67
4.4	\$AA6	68
4.5	#AABBDD	69
4.6	\$AAF & \$AAM.....	70
4.7	~AA0	71

4.8	~AA1	72
4.9	~AA2 & ~AA3ETTTT	73
4.10	~AA4P & ~AA4S	74
4.11	~AA5P & ~AA5S	75
4.12	~AA7N & ~AA8	76
4.13	~AAE0NN & ~AAE1NNDD	77
4.14	~AAE2 & ~AAE3	79
4.15	~AAPNDD	80
4.16	~**	81
5.	APPLICATION NOTES	82
5.1	GO TO INITIAL SETTING	82
5.2	DUAL WATCHDOG OPERATION PRINCIPLE	83
5.3	HOST WATCHDOG APPLICATIONS NOTES	84
5.4	MODULE WATCHDOG APPLICATIONS NOTES	86
5.5	ANALOG DATA FORMAT	87
5.6	D/O OPERATION PRINCIPLE	88
5.7	COMMAND RESPONSE TIME	89
5.8	USERDEMO.EXE OPERATION PRINCIPLE	90
5.9	DB-3R APPLICATIONS NOTES	91

1. Introduction

1.1 Products Check List

In addition to this manual, the package includes the following items:

- one piece of WDT-03 card
- one cable, CA0910F, for linking to local host-PC
- one reset-wire for JP1
- two temperature sensors with wire for external T1 & T2
- one piece of companion CD for software
- one piece of release note

It is recommended to read the release note first. Some important information will be given in release note as follows:

1. where you can find the software driver & utility
2. how to install software & utility
3. where is the diagnostic program
4. FAQ

Attention!

If any of these items is missing or damaged, contact the local dealer sold you this product. Save the shipping materials and carton in case you want to ship or store the product in the future.

1.2 Default Setting

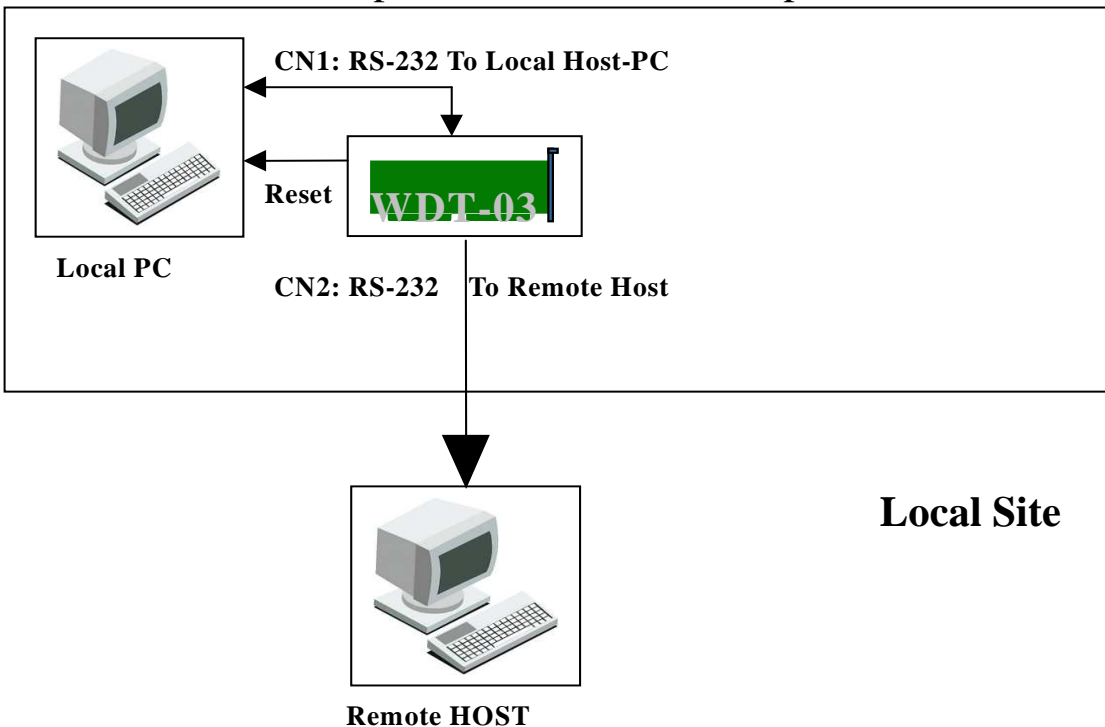
The default setting of WDT-03 is given as following:

- address=01, baud rate=9600, checksum disable
- data=1 start+8 data+1 stop(no parity)
- Mounting to the PCI-bus
- power of FAN = internal +12V
- host watchdog is disable & its timer=0
- Start values for all D/O are OFF
- Safe values for all D/O are OFF
- Duty cycles of all PWM are 100%
- Module name is → **WDT-03**

1.3 Introduction

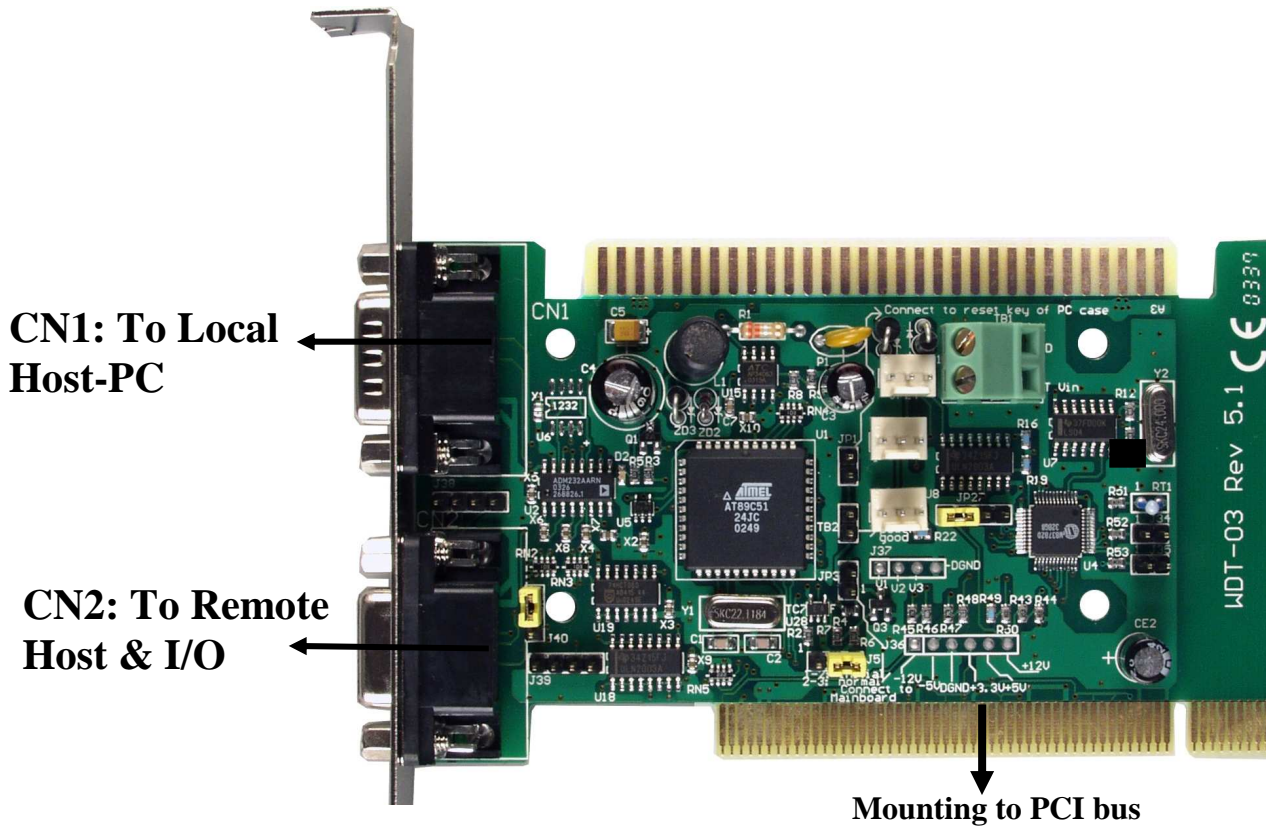
It's a fact that the PC hardware and software maybe fail. To prevent the failure, many different solutions are proposed. However, none of these solutions offer a 100% assurance. Since it's hard to prevent the failure, to detect the failure becomes more and more important. The WDT-03 is used to detect the failure of the software and hardware. It can be used to reduce the risk of PC failure. The WDT-03 is useful even for those systems with built-in watchdog circuit.

The WDT-03 equips two RS-232 ports, one is master port and another one is slave port. The master RS-232 port is designed to communicate with the local host-PC. The local host-PC can enable/disable/refresh/read-back the watchdog timer through the master RS-232 port. The slave RS-232 is used to send information to the remote Host-PC for monitoring. So the remote HOST will get information immediatly if the local host-PC has any software or hardware problem, even the local power is failure.

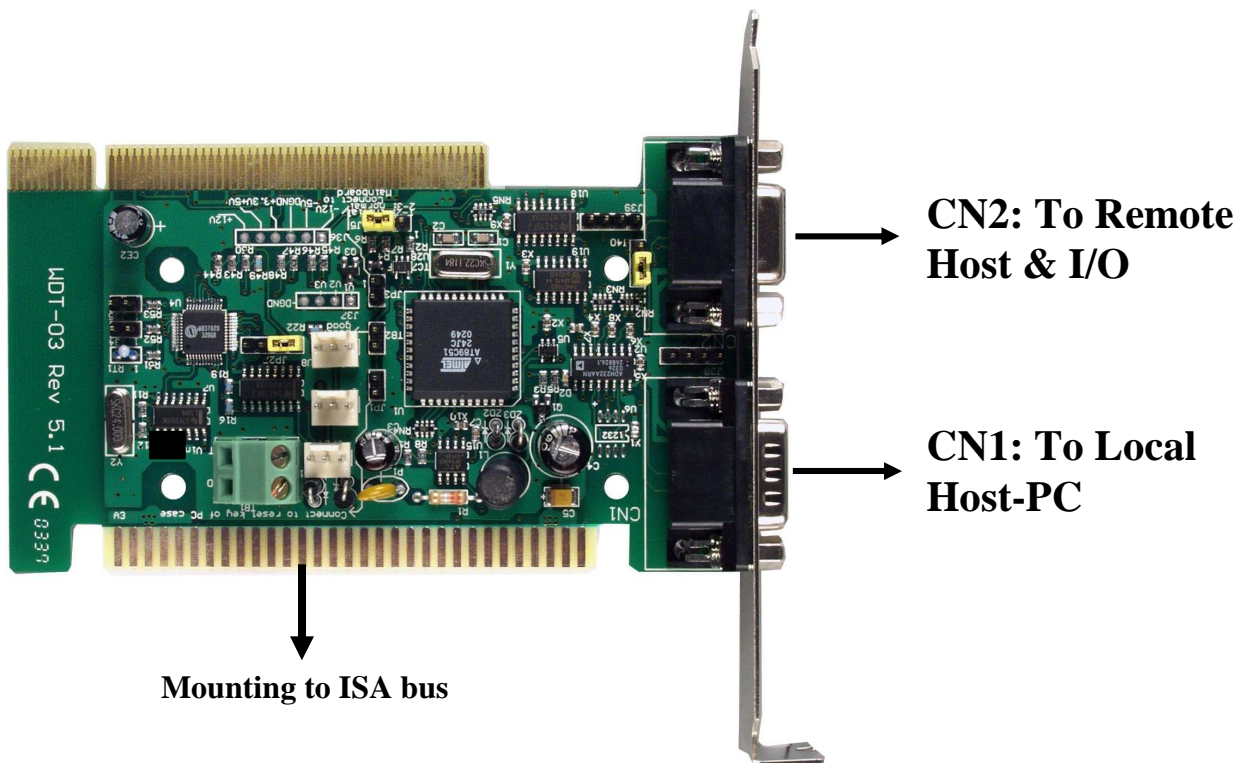


The WDT-03 can be mounted in ISA/PCI bus or any bus as follows:

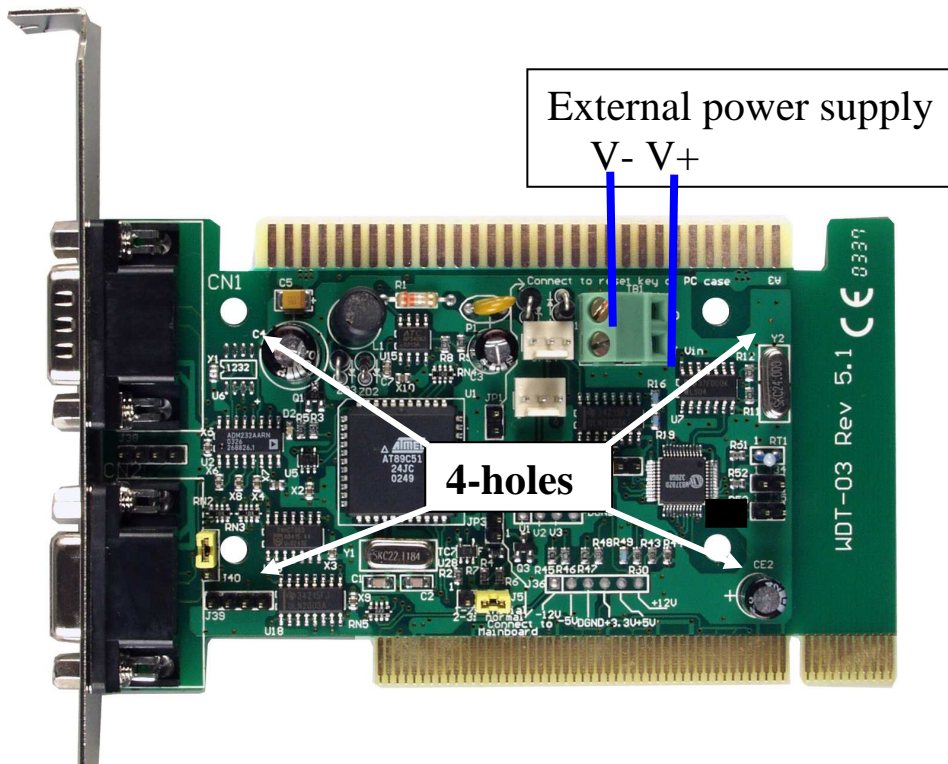
1. Mounting on PCI-bus → default setting



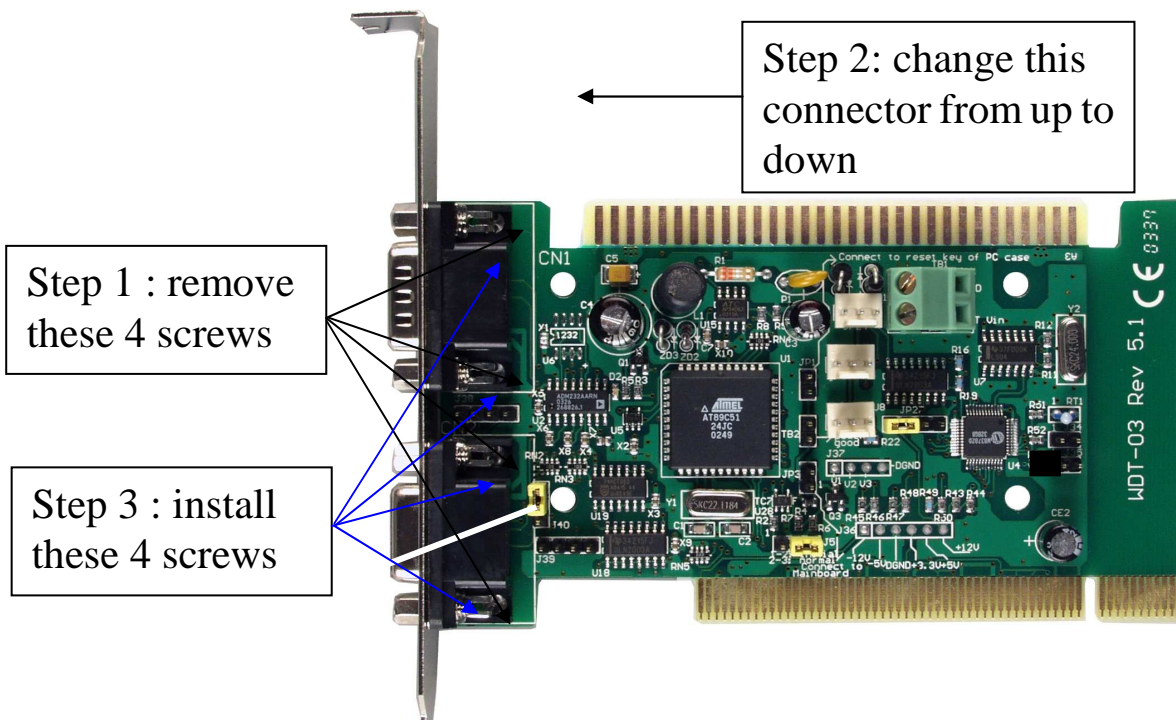
2. Mounting on ISA-bus



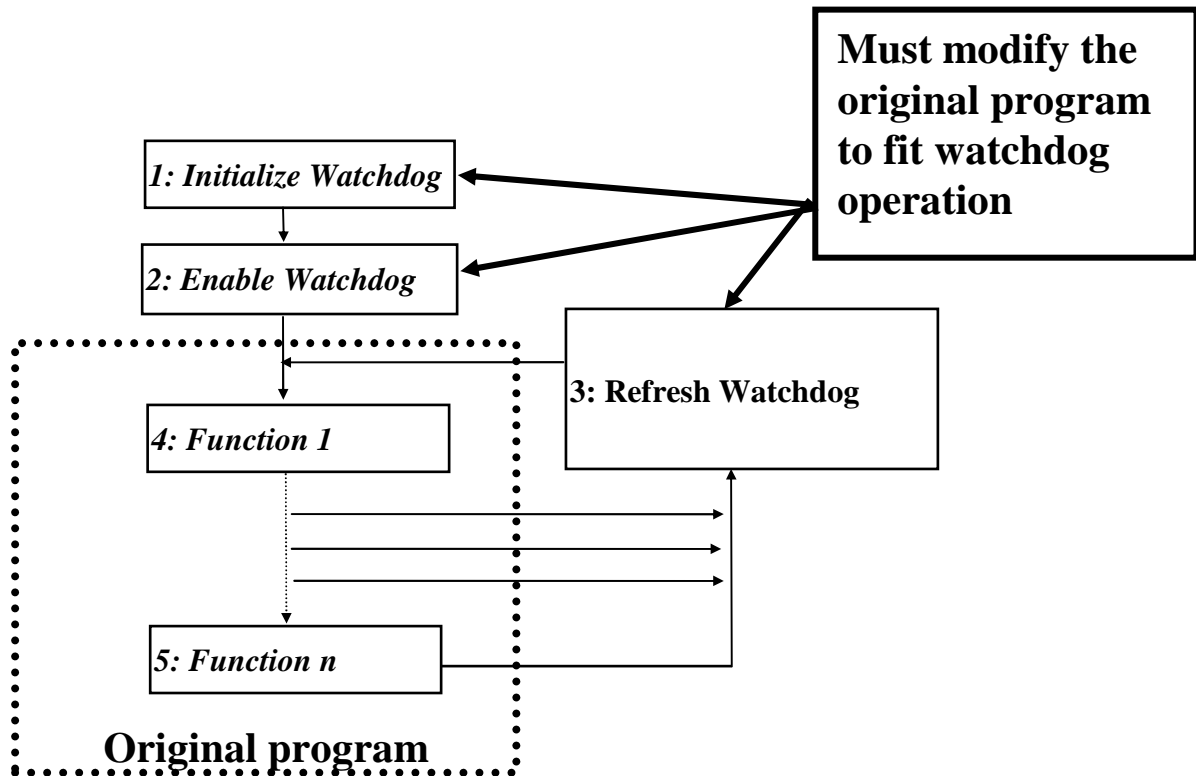
3. Mounting on any bus (4-hole position):



*** Change WDT-03 from PCI bus to ISA bus ***



1.4 Traditional WDT Cards



The traditional WDT cards equip a hardware watchdog for application program to start & refresh it in the normal operation. If the host-PC is failure for software or hardware problem, the refresh operation will be discontinued & the hardware watchdog can reset the host-PC.

So the software engineer has to add three parts of program into his original program as follows:

1. initialize the hardware watchdog
2. start the hardware watchdog
3. continually refresh the hardware watchdog

Sometimes, the application program is hard to modify as follows:

1. The application program is finished & hard to modify
2. It is a well-designed commercial package, so user can use it but can not modify it.
3. No time to modify the original program

So we need an **advance WDT system** to solve all these problems.

1.5 Advance WDT System

The features of an advance WDT system are given as following:

- **Don't have to modify the original program**
- **Can be used in ISA bus, PCI bus & any system with RS-232 interface**
- **Harsh environment detection & warning before system crash**
- **Cost Effective Solution**

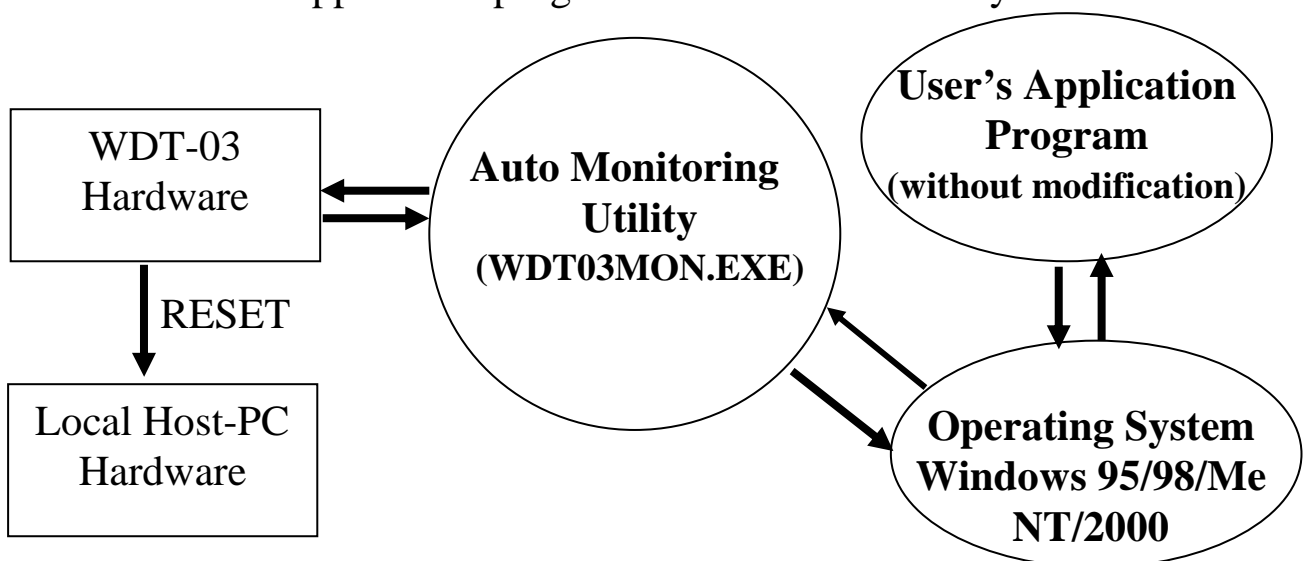
1.5.1 Don't Have to Modify User's Application Program

The traditional WDT system has to modify the original program. Sometimes, the application program is hard to modify. Refer to Sec. 1.4 for more information.

The working steps of using an **advance WDT system** are given as follows:

1. **Hardware installation & diagnostic, refer to Sec. 1.10.1**
2. **Reset mechanism installation & diagnostic, refer to Sec. 1.10.2**
3. **Install auto monitoring utility, refer to Sec. 2.5**
4. **Install & execute user's application program, refer to Sec. 1.10.3**

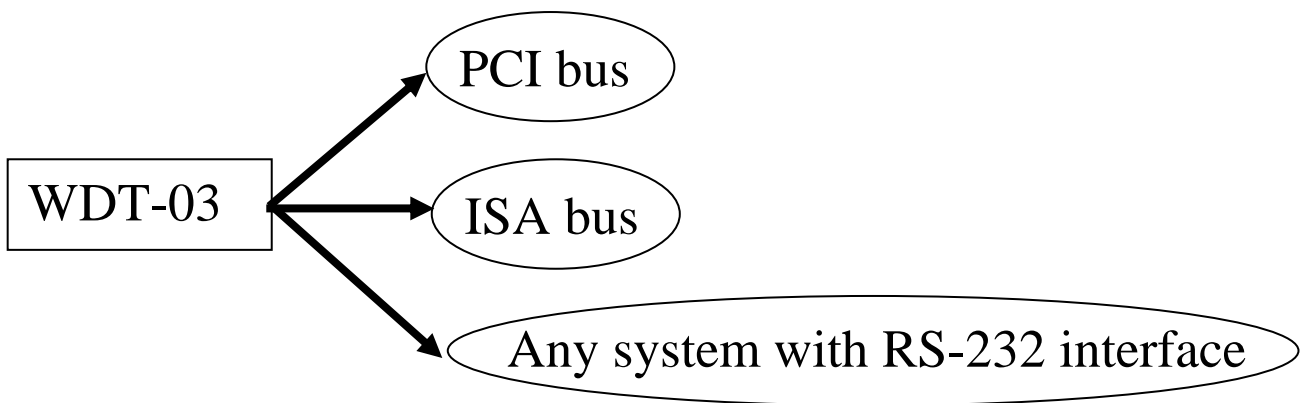
From the above steps, you can find that the user's application program can be executed without modification. The auto monitoring utility will monitors the operating system & user's application program automatically. If system is frozen by any software or hardware problems, the dual-watchdog mechanism will reset this frozen system immediately & restart the user's application program to work continuously.



1.5.2 Can be Used in ISA bus, PCI bus & any System

with RS-232 interface

An advance WDT system can be used in ISA bus, PCI bus or any bus with 4-holes mounting. Refer to Sec. 1.3 for more information.



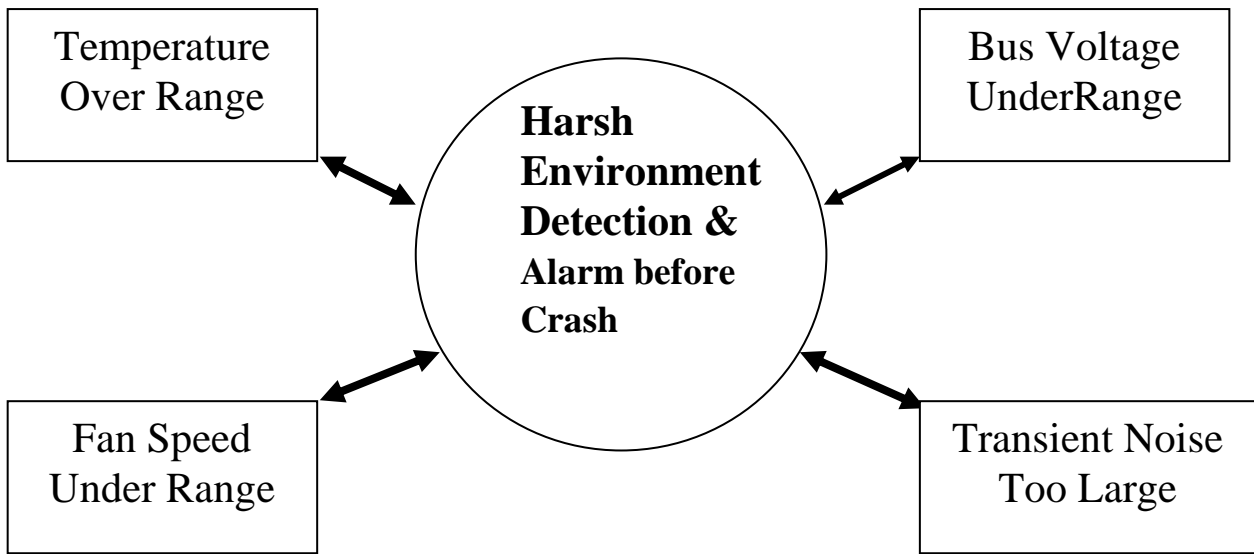
1.5.3 Harsh Environment Warning before System Crash

There are many couple noise or energy transient in industry application environment. The host-PC may fail if these noises are really too large.

The application environment will change without any notice. So a stable system in yesterday can not be guaranteed stable in tomorrow.

Refer to Sec. 5.4 for harsh environment detection & warning. If a harsh environment is detected, the system engineer can check what is happening immediately. So the harsh environment can be improved to a normal environment before the crash of host-PC.

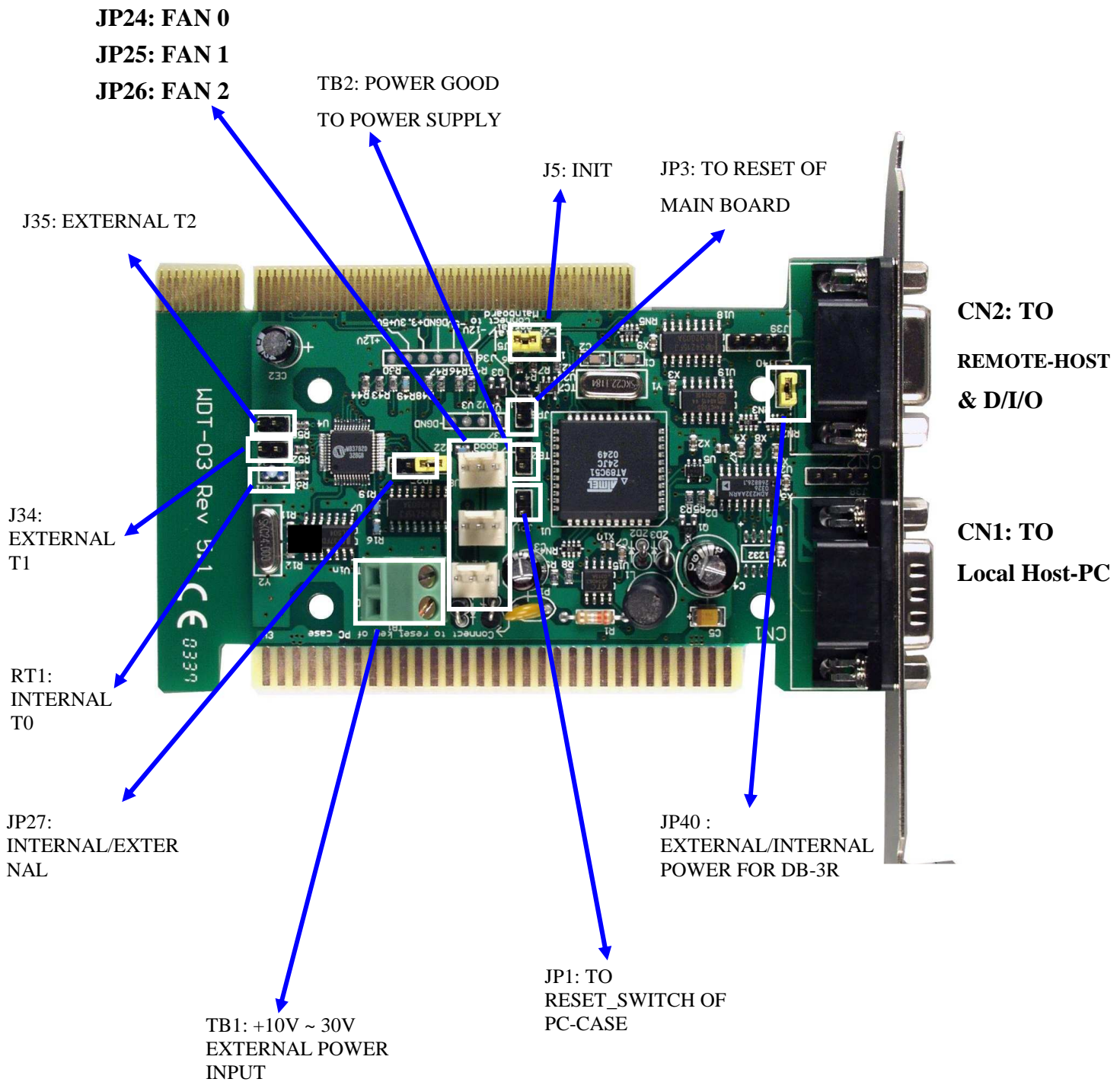




1.5.4 Cost-Effective Solution

The WDT-03 is the most cost-effective solution of available products in the world. Call the local dealer for price information.

1.6 Connectors



1.7 Pin Assignment

1.7.1 CN1 : to Local Host-PC

CN2 : to Remote-Host & D/I/O

CN1: DB9, Male	
Pin number	Signal name
1	N/C
2	TXD, RS-232 output
3	RXD, RS-232 input
4	N/C
5	GND
6	N/C
7	N/C
8	N/C
9	N/C

CN2: DB9, Female	
Pin Number	Signal name
1	D/I-3, 3.5V ~ 30V
2	TXD, RS-232 output
3	Power of D/O, 30V max.
4	D/O-2, 100mA, 30V
5	GND
6	D/I-2, 3.5V ~ 30V
7	D/I-1, 3.5V ~ 30V
8	D/O-1, 100mA, 30V
9	D/O-3, 100mA, 30V

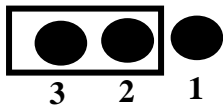
Note 1: Pin 2 of CN1 & Pin2 of CN2 are connected together to the uP of WDT-03. So the local-Host & remote-Host will receive the same message when uP of WDT-03 send out message.

2: When the watchdog is enable, the local-host must send "\$AA5" command to WDT-03 for harsh environment checking. The WDT-03 will send the status back to the local-HOST & remote-PC. If the local system is unstable, the remote-Host will receive the BAD status or even can not receive any message from WDT-03. Refer to Sec. 5.6 For more information.

3: The D/I of CN2 can be connected to photo sensors or limit switches. The GND of extern D/I should be connected to Pin 5 of CN2.

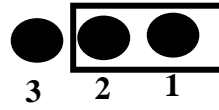
4: The D/O of CN2 are open collector output. They can be used to drive relay or LED for emergency control. The GND of extern D/I should be connected to Pin 5 of CN2. If these D/O are used to drive conductive load, such as relay, the power supply of conductive load should be connected to Pin 3 of CN2 for switching noises remove.

1.7.2 J5: Init



**Default: WDT-03 =
Normal State**

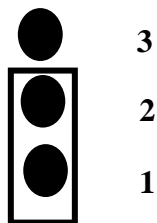
- Address=EEPROM
- Baudrate=EEPROM
- Checksum=EEPROM
- Data format=N81



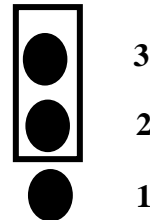
**WDT-03 =
Initial State**

- Address=0x00
- Baudrate=9600
- Checksum=disable
- Data format=N81

1.7.3 J40: Init

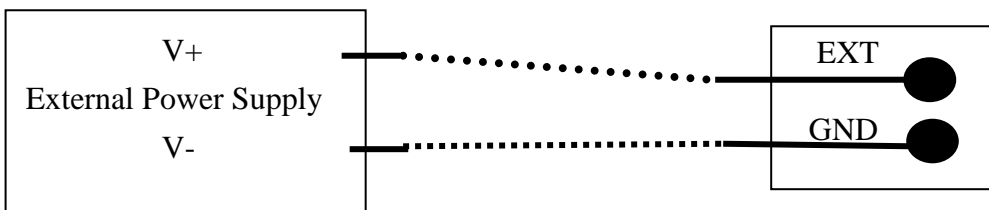


**Default: DB-3R need
external power(only for 5V)**



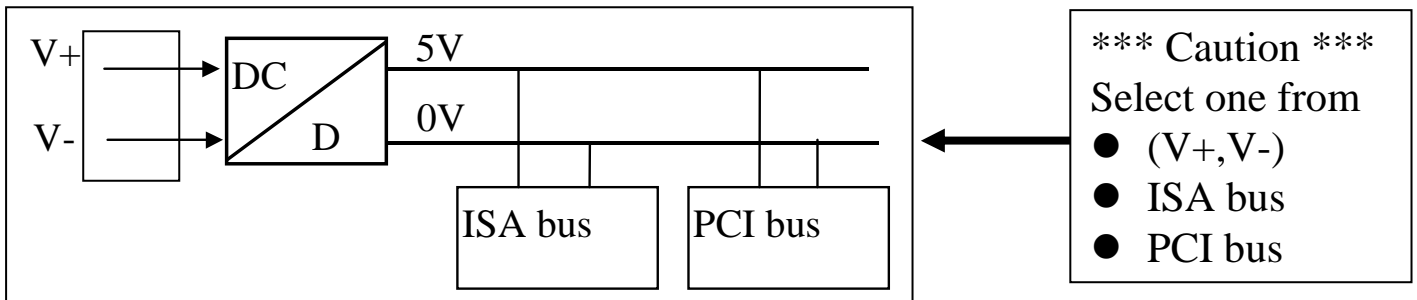
**DB-3R don't need
external power**

1.7.4 TB1: +10V ~ 30V External Power Input

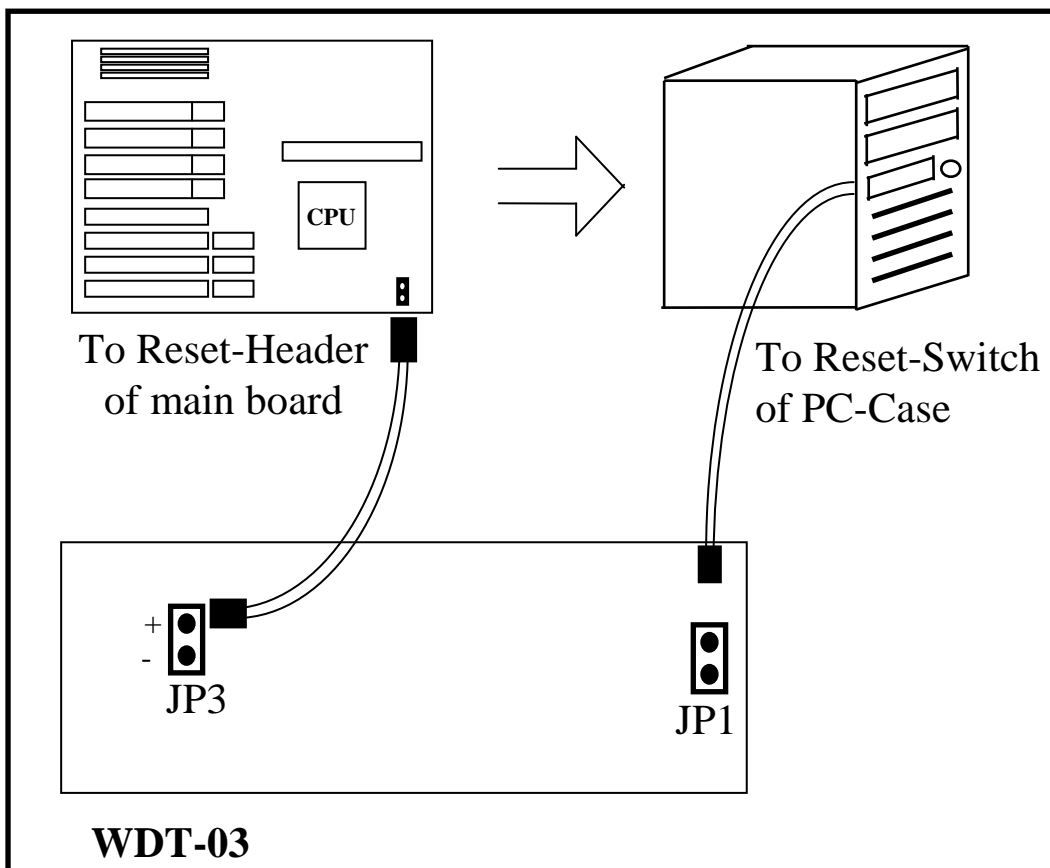


(V+, V-) can be +10V ~ +30V DC, unregulated

- Note 1: The (+5V, GND) can come from (V+,V-), PCI bus or ISA bus.
 2: (V+,V-), PCI bus or ISA bus are wired together. So only one of these 3 power sources can be applied at the same time.



**1.7.5 JP1: To Reset_Switch of PC-Case
 JP3: To Reset-header of main board**



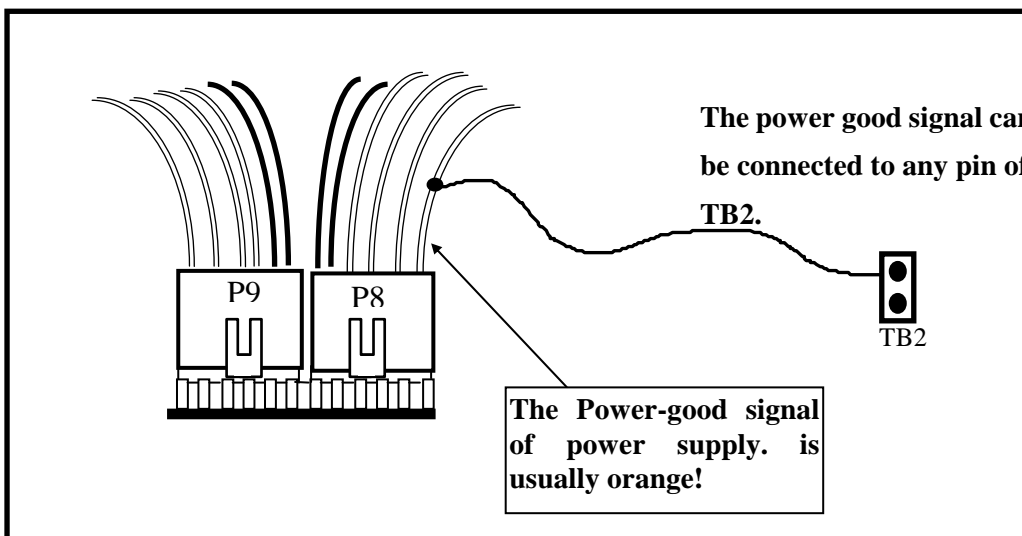
The installation steps are given as follows:

1. remove the original reset-cable from reset-header of main board & installed it into JP1
2. install the new reset-cable into JP3 & reset-header of main board

3. power-on & check pc
4. if pc can not boot-up, change the orientation of reset-cable installed in JP3 & power on
5. press reset-key in PC-Case to check the original reset function

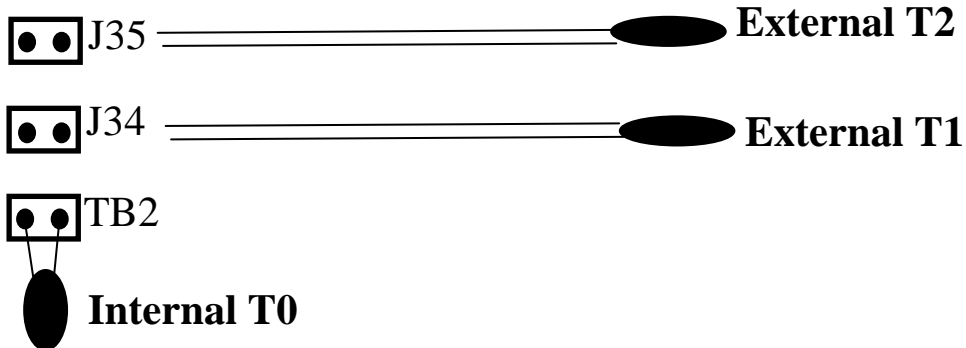
Note: 1. Both the RESET signal & Power-good signal can be used to reset local host-pc. Select one signal for your convenience.
 2. It is recommended to select reset signal.

1.7.6 TB2: To Power_Good of Power Supply

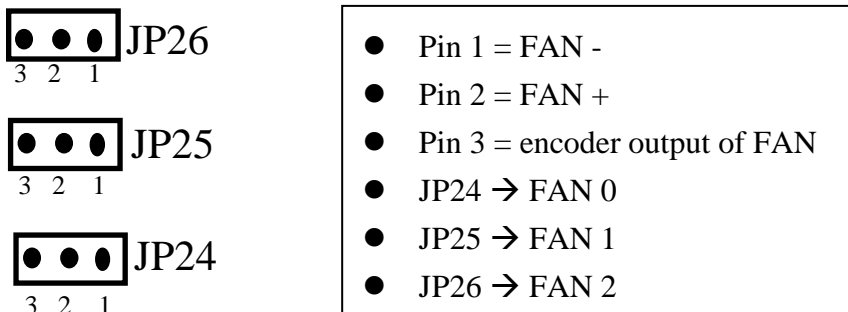


Note: 1. Both the RESET signal & Power-good signal can be used to reset the local host-PC. Select one signal for your convenience.
 2. It is recommended to select reset signal.

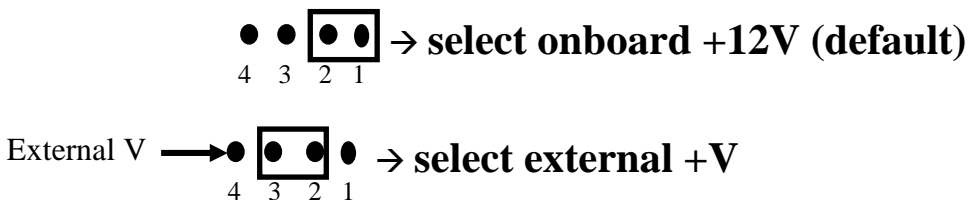
1.7.7 J34/J35: For External T1/T2 RT1: For Internal T0



1.7.8 JP24/J25/J26: For FAN-0/1/2



1.7.9 JP27: Internal/External Power of Fan



- | |
|--|
| <ul style="list-style-type: none"> ● Pin 1 → +12V of ISA/PCI ● Pin 2 → to VCC of FAN 0/1/2 ● Pin 3 → connect to Pin 4 ● Pin 4 → External V |
|--|

1.8 Specifications

- O. S./Bus/Mounting independent
- OEM/ODM acceptable

Watchdog timer

- Software enable/disable watchdog timer
- Form 0.03 second to 1966.05 seconds

Bus Voltage monitoring: 3.3V, 5V, 12V, -5V, -12V

Fan Speed monitoring: 3 channels

Temperature monitoring: 3 channels

EEPROM

- 63 bytes for user read/write
- read/write cycles:100,000 times

Interface

- RS-232 * 1 for local CPU
- RS232 * 1 for remote HOST(monitor the local CPU)

RS-232

- Baud rate: 2.4K/4.8K/9.6K/19.2K/38.4K/57.6K/115.2K BPS programmable
- Data format: N81
- Command format: the 7000 family compatible

Reset Mechanism

- Power good signal for PC system
- Reset signal to simulate external reset-key pressed

Mounting

- ISA/PCI bus for PC
- Four through-hole mounting for any system with RS-232 interface

Dimensions: 140mm x 70mm

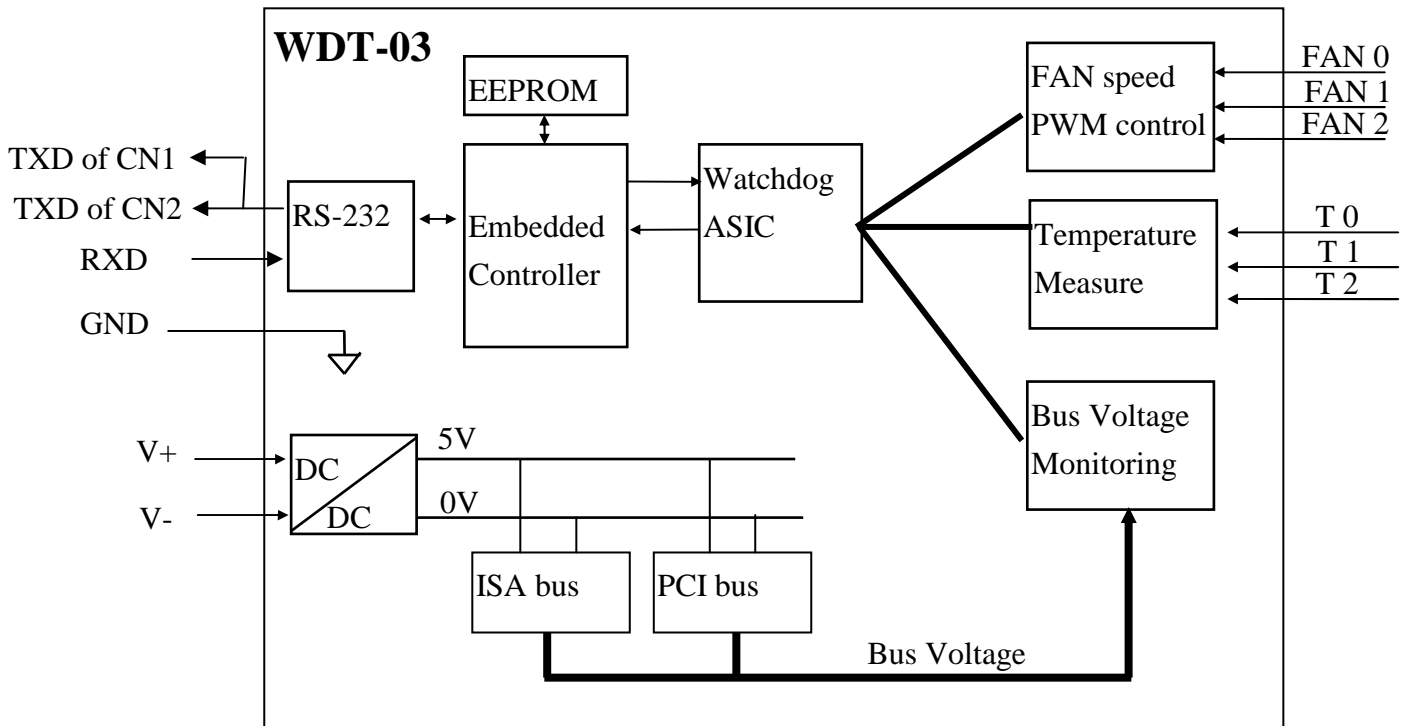
Power Consumption: 2W

Operating Temperature: 0°C ~ 60°C

Storage Temperature: -20°C ~ 75°C

Software: Support DOS, 95/98/Me/2000/NT driver

1.9 Block Diagram



1.10 Quick Start

1.10.1 Hardware Installation & Diagnostic

Step 1 : power off local host-PC first

Step 2 : Install WDT-03 into PCI or ISA bus of local host-PC. The default setting is PCI bus, refer to Sec. 1.3 for change from PCI bus mounting to ISA bus mounting

Sep 3 : connect the communication cable, CA0910F, between COM1 of host-PC & CN1 of WDT-03.

Step 4 : Power up the local host-PC

Step 5 : Refer to Sec. 2.2 for 7000 Utility Installation

Step 6 : Execute 7000 Utility to detect & diagnostic the WDT-03. Refer to Sec. 2.3 for more information.

Note: If the CA0910F is connected to COM2 of local host-PC, user should change the COM port of 7000 utility to COM2 & press “**Search**” again to detect WDT-03.

1.10.2 Reset Mechanism Installation & Diagnostic

Step 1 : power off local host-PC first

Step 2 : Refer to Sec. 1.7.4 for reset signal installation.

Step 3 : Power up the local host-PC

Step 4 : Press Reset-Switch in case of local host-PC to test the original reset mechanism.

Step 5 : Refer to Sec. 2.2 for WDT-03 Auto Monitoring Utility Installation

Step 6 : Refer to Sec. 2.4.1 to enable & configure the Auto Monitoring Utility.

Step 7 : Press Ctrl + Alt + Del to test the reset mechanism of WDT-03. Refer to Sec. 2.4.10 for more information.

1.10.3 Auto Execute User's Application Program

Step 1 : Refer to Sec. 2.4 to enable WDT-03 Auto Monitoring Utility

Step 2 : Drag and drop user's application program, for example

\CD\NAPDOS\WDT\WDT03\USERDEMO\userdemo.exe, to

“StartUp” at Start menu. Refer to Sec. 2.5 for more information

Step 3: power down the local-PC

Step 4 : Power up the local host-PC

Step 5 : Check user's application program, userdemo.exe, will automatic start to execute after the Windows is loading into system.

Step 6 : Check the WDT-03 icon is placed in system tray. Refer to Sec. 2.4.1 for more information.

Step 7 : Click the WDT-03 icon in system tray to execute it & check the FAN, temperature value & bus voltage. Refer to Sec. 2.4.3 / 2.4.4 / 2.4.5 for more information.

Step 8 : Press Ctrl + Alt + Del to test the reset mechanism of WDT-03. Refer to Sec. 2.4.10 for more information.

1.10.4 DOS Utility for Programming the WDT-03

Step 1 : Refer to Sec. 3.1 for NAP7000S installation

Step 2 : Refer to Sec. 3.2 for how to use NAP7000S

Step 3 : Refer to CD\NAPDOS\7000>manual\7520.pdf for the following information:

- **Find status of unknown module** (Sec. 5.1)
- **Change module address**(Sec. 5.2)
- **Change baud rate** (Sec. 5.3)
- **Checksum enable/disable**(Sec. 5.4)

The diagnostic utility, TEST.EXE, provide by NAP7000S is used to send/receive command to WDT-03 in the following sections.

1.10.5 WDT-03 Identification

1. Refer to Sec. 1.10.4 for more information & run **TEST.EXE**

2. press **2**

3. press **\$012[Enter]** → Receive=!01400600

4. press **2**

5. press **\$015[Enter]** → Receive=!011

6. press **2**

7. press **\$015[Enter]** → Receive=!010

8. press **2**

9. press **\$01M[Enter]** → Receive=!01WDT-03

10. press **2**

11. press **\$01F[Enter]** → Receive=!01A1.0

12. press **2**

13. press **~010[Enter]** → Receive=!0100

14. press **2**

15. press **~012[Enter]** → Receive=!0100000

- step 3: read the status of WDT-03 is baud rate 9600, checksum disable
- step 5/7: step 5 → first read since power on. Step 7 → not first read
- step 9: read module name=WDT-03
- step 11: WDT-03 firmware version number=A1.0

-
- step 13: host watchdog state is normal → no host watchdog failure
 - step 15: host watchdog is disable, watchdog time=0 second

1.10.6 System Status Reading

1. Refer to Sec. 1.10.4 for more information & run **TEST.EXE**
2. press **2**
3. press **~018[Enter]** →
Receive=!01D1.BB.AA.60.E0.24.1F.1C.FF.FF.84.FF.FF.FF[Enter]
4. press **2**
5. press **~0170[Enter]** → Receive=!D1
6. press **2**
7. press **~0171[Enter]** → Receive=!BB
8. press **2**
9. press **~0172[Enter]** → Receive=!AA
10. press **2**
11. press **~0173[Enter]** → Receive=!60
12. press **2**
13. press **~0174[Enter]** → Receive=!E0
14. press **2**
15. press **~0175[Enter]** → Receive=!24
16. press **2**
17. press **~017A[Enter]** → Receive=!84
18. press **2**
19. press **~017D[Enter]** → Receive=!FF

(Note: this WDT-03 is installed in PCI-bus)

- step 3: read all system status
- step 5: read +3.3V → 3.34V
- step 7: read +5V → 4.99V
- step 9: read +12V → 12.19V
- step 11: read -12V → -11.88V
- step 13: read -5V → PCI bus no -5V → it is a invalid data
- step 15: read T0=36°C

- step 17: read speed of FAN2=5114 RPM
- step 19: read PWM of FAN2=100%

1.10.7 PWM Control

1. Refer to Sec. 1.10.4 for more information & run **TEST.EXE**

2. press 2

3. press ~01P019[Enter] → Receive=!01

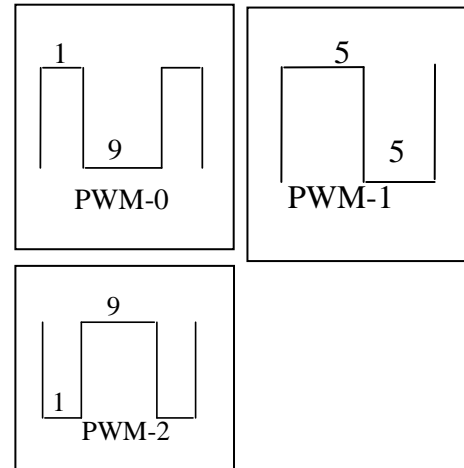
4. press 2

5. press ~01P17F[Enter] → Receive=!01

6. press 2

7. press ~01P2E5[Enter] → Receive=!01

- step 3: PWM-0 = 0x19/0xff=10%
- step 5: PWM-1 = 0x7f/0xff=50%
- step 7: PWM-2 = 0xe5/0xff=90%



1.10.8 Digital Input/Output Control

1. Refer to Sec. 1.10.4 for more information & run **TEST.EXE**

2. press 2

3. press #01000F[Enter] → Receive=>

4. press 2

5. press ~015P[Enter] → Receive=!01

6. press 2

7. press #010000[Enter] → Receive=>

8. press 2

9. press ~015S[Enter] → Receive=!01

10. press 2

11. press \$016[Enter] → Receive=!000F00

12. press 2

13. press ~014P[Enter] → Receive=!010F00

14. press 2

15. press ~014S[Enter] → Receive=!010000

- step 3: set all D/O=ON

-
- step 5: set all power on values = ON
 - step 7: set all D/O=OFF
 - step 9: set all safe values = OFF
 - step 11: read D/I/O, D/I = all ON, D/O = all OFF
 - step 13: read all power on values = ON
 - step 15: read all safe values = OFF

1.10.9 EEPROM Read/Write

1. Refer to Sec. 1.10.4 for more information & run **TEST.EXE**

2. press **2**

3. press **~01E001[Enter]** → Receive=!FF

4. press **2**

5. press **~01E002[Enter]** → Receive=!FF

6. press **2**

7. press **~01E3[Enter]** → Receive=!01

8. press **2**

9. press **~01E10155[Enter]** → Receive=!01

10. press **2**

11. press **\$01E102AA[Enter]** → Receive=!01

12. press **2**

13. press **~01E2[Enter]** → Receive=!01

14. press **2**

15. press **~01E001[Enter]** → Receive=!55

16. press **2**

17. press **~01E002[Enter]** → Receive=!AA

- step 3: read EEPROM[01]=0xFF

- step 5: read EEPROM[02]=0xFF

- step 7: enable EEPROM to write operation

- step 9: write 0x55 to EEPROM[01]

- step 11: write 0xAA to EEPROM[02]

- step 13: disable EEPROM from write operation

- step 15: read EEPROM[01]=0x55

- step 17: read EEPROM[02]=0xAA

1.10.10 Host Watchdog operation

1. Refer to Sec. 1.10.4 for more information & run **TEST.EXE**
 2. press **2**
 3. press **~010[Enter]** → Receive=!0100
 4. press **2**
 5. press **#0100FF[Enter]** → Receive=>
 6. press **2**
 7. press **~015S[Enter]** → Receive=!01
 8. press **2**
 9. press **#010000[Enter]** → Receive=>
 10. press **2**
 11. press **~01310064[Enter]** → Receive=!01
 12. press **2**
 13. press **~**[Enter]** → No response
 14. repeat step 12/13 to simulate normal operation until to test next step
 15. check all D/O are OFF now, go to next step
 16. wait 3 seconds to simulate local host-PC now is failure
 17. check all D/O are ON now
 18. press **2**
 19. press **~010[Enter]** → Receive=!0104
 20. press **2**
 21. press **#010000[Enter]** → Receive=!
 22. press **2**
 23. press **~011[Enter]** → Receive=!01
 24. press **2**
 25. press **#010000[Enter]** → Receive=>
- step 7: set safe states of D/O = ON
 - step 9: set all D/O to OFF for testing
 - step 11: set host watchdog timer=3 seconds & enable it
 - step 13: local host-PC send “~**” to tell WDT-03 “**I am OK**”
 - step 15: read EEPROM[01]=0x55
 - step 17: now WDT-03 find local host-PC is failure, so all D/O are going to their safe states now
 - step 21: WDT-03 does not accept D/O command now
 - step 23: clear host watchdog failure

- step 25: WDT-03 accept D/O command now

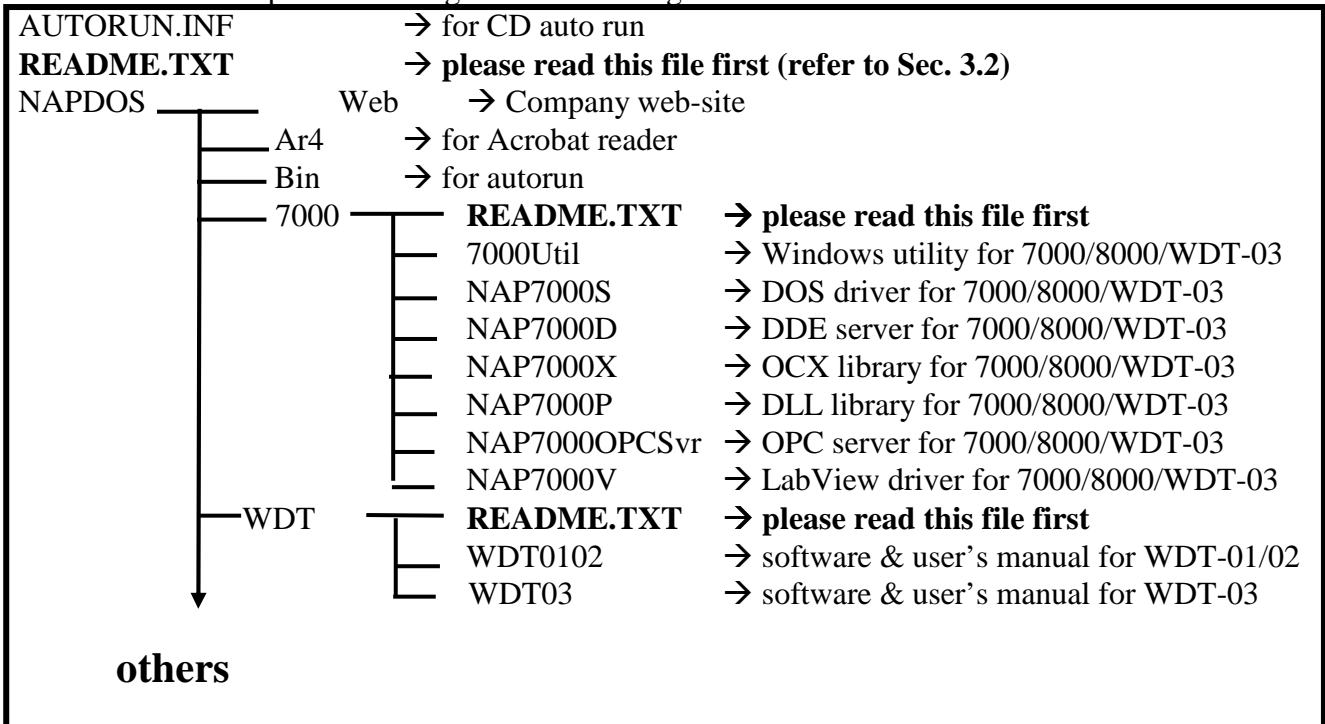
2. Software Utility

The software environment of WDT-03 can be divided into three parts as following:

- 1: Configuration Utility → refer to Sec. 2.3
- 2: Auto Monitoring Utility → refer to Sec. 2.4
- 3: Software driver → refer to chapter 3

2.1 Content of Companion CD

The content of companion CD is given as following:



Note 1: The software & manual are updated very frequently, so the content of companion CD is also updated very frequently. The best way is to read every README.TXT located in every directory. All updated information is given in these files.

Note 2: most of files in \NAPDOS\7000*. * are for both 7000 & 8000 family & WDT-03

Note 3: this manual is located in \NAPDOS\WDT\WDT03\WDT03.DOC

2.2 Utility Software Installation

It is recommended to read the **release note** first. The release note is given in the shipping.

(1) 7000 Utility Installation & User's Manual

Double Click on \CD\NAPDOS\7000\7000Util\SETUP\Setup.exe

Manual → on-line help after the 7000 utility is installed

(2) WDT-03 Auto Monitoring Utility Installation

Double Click on \CD\NAPDOS\WDT\WDT03\SETUP\Setup.exe

Manual → \CD\NAPDOS\WDT\WDT03\WDT03.DOC

The CD contents for WDT-03 is given as following:

\CD\NAPDOS\WDT\WDT03\readme.txt → read this file first

\CD\NAPDOS\WDT\WDT03\wdt03.DOC → user's manual of WDT-03

\CD\NAPDOS\WDT\WDT03\sheet03.DOC → data sheet of WDT-03

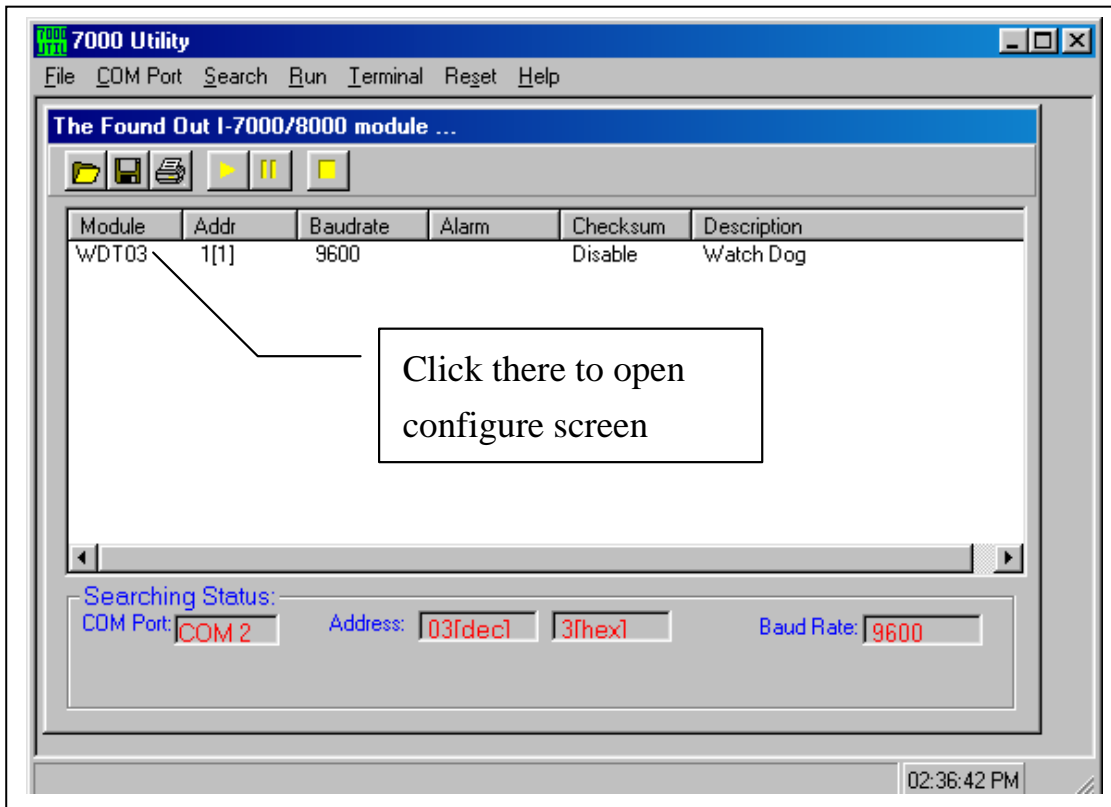
\CD\NAPDOS\WDT\WDT03\SETUP*. * → for auto monitoring utility

\CD\NAPDOS\WDT\WDT03\UserDemo*. * → example of user's application program

2.3 Configuration Utility

The default setting of WDT-03 is address = 01, baud rate = 9600 and checksum = disable. The 7000 Utility can be used to configure & test the WDT-03. Refer to Sec. 2.2 for installation & manual of 7000 utility.

After start the 7000 Utility, it will begin to search all 7000/8000 modules and WDT-03 installed in the COM port. The searching result will be shown in the search result as follows:



Refer to Sec. 3.3 for more information.

Note 1: If 7000 utility can not find the WDT-03, please change the “**COM Port**” filed of 7000 utility to the other COM port & click “**Search**” field again.

Note 2: The final setting of 7000 utility will be saved in PC & change to correct configuration next times.

2.3.1 Configure the WDT-03

- Click “WDT-03” in 7000 Utility, the configure screen is given as follows:



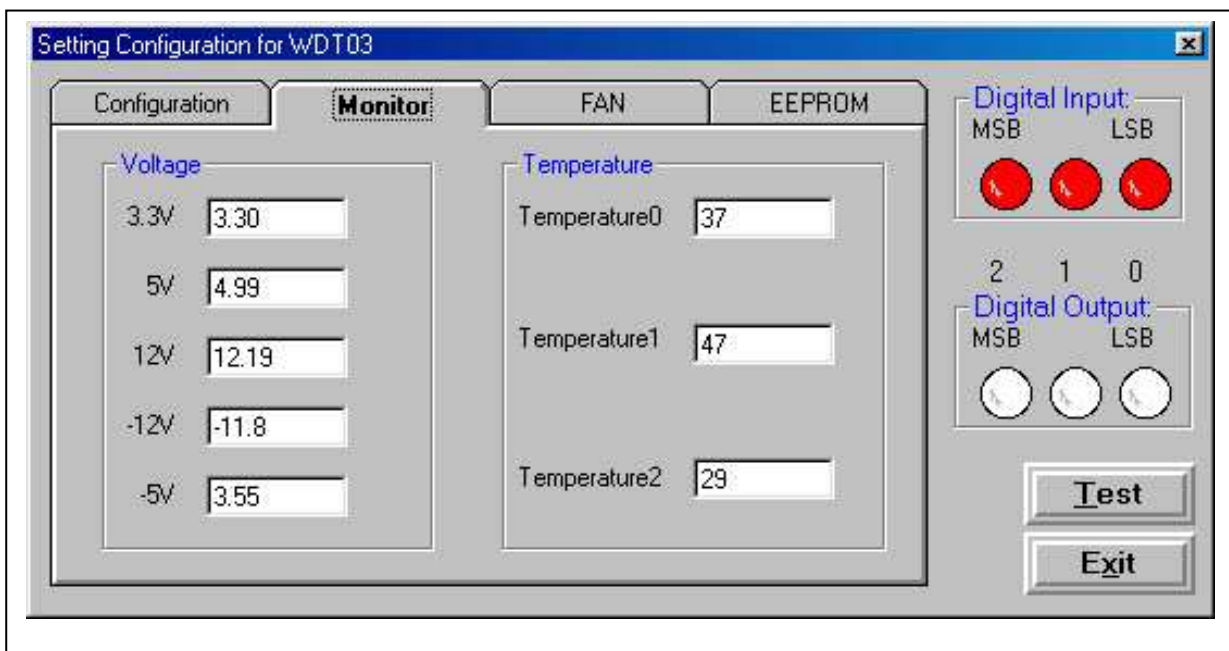
- The current configuration setting will be shown in this page
- User can change the current setting to the desire setting now
- Note: If user want to change the **baud rate** or **checksum state**, the J5 of WDT-03 should be change to initial state first. Refer to Sec. 1.7.3 for more information.
- “**Set**” button → save & enable host watchdog timer
- “**Stop**” button → disable the host watchdog timer
- “**Test**” button → stop send “~*~” command to WDT-03. If the host watchdog is enable now, the WDT-03 will reset this local host after the watchdog timer is up.
- “**Exit**” button → disable the host watchdog timer & exit to system
- After change all setting, press the “**Setting**” button to complete the setting process.
- click digital output icon → will toggle the current state of digital output (the red icon indication ON and the white icon indication OFF)

2.3.2 System Status Monitoring

The WDT-03 can be used to monitor the system status given as follows:

- 1) 3.3V, 5V, 12V, -5V, -12V bus voltage monitoring
- 2) 3 channels fan speed monitor
- 3) 3 channels fan speed PWM control
- 4) 3 channels temperature monitor
- 5) Watchdog timer control

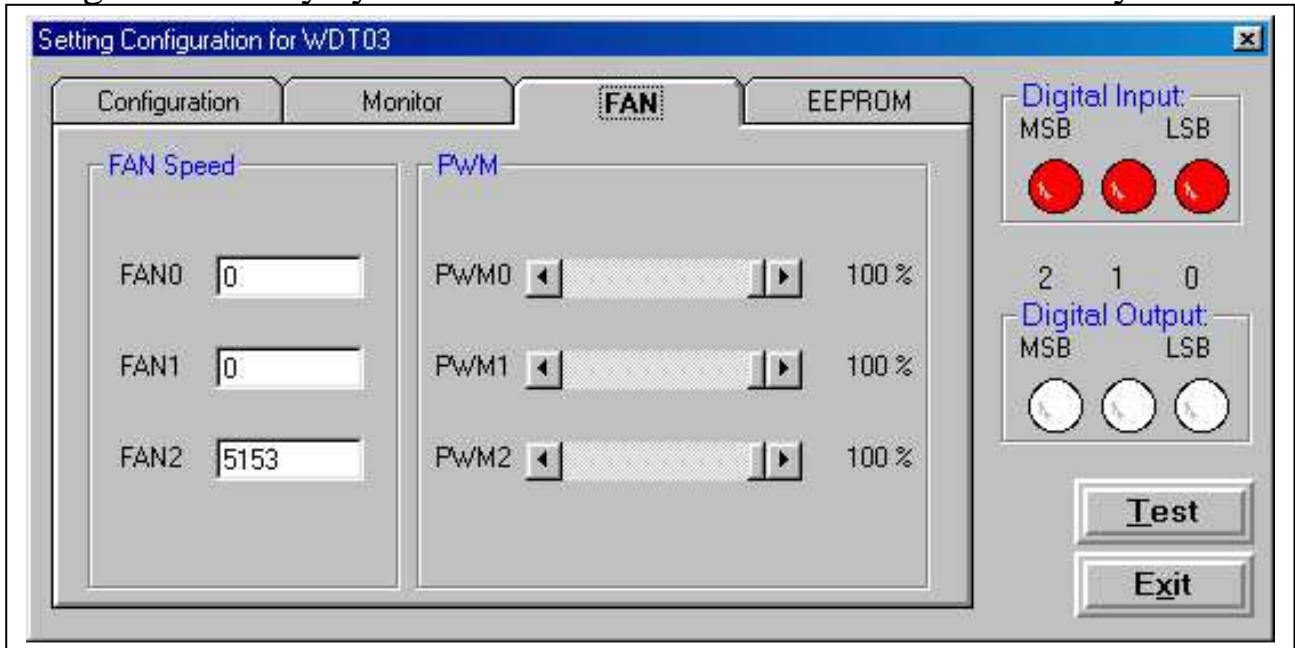
The bus voltage values & temperature values will be shown in “**Monitor**” page as follows:



- “**Test**” button → stop send “~*~” command to WDT-03. If the host watchdog is enable now, the WDT-03 will reset this local host after the watchdog timer is up.
- “**Exit**” button → disable the host watchdog timer & exit to system
- click digital output icon → will toggle the current state of digital output (the red icon indication ON and the white icon indication OFF)

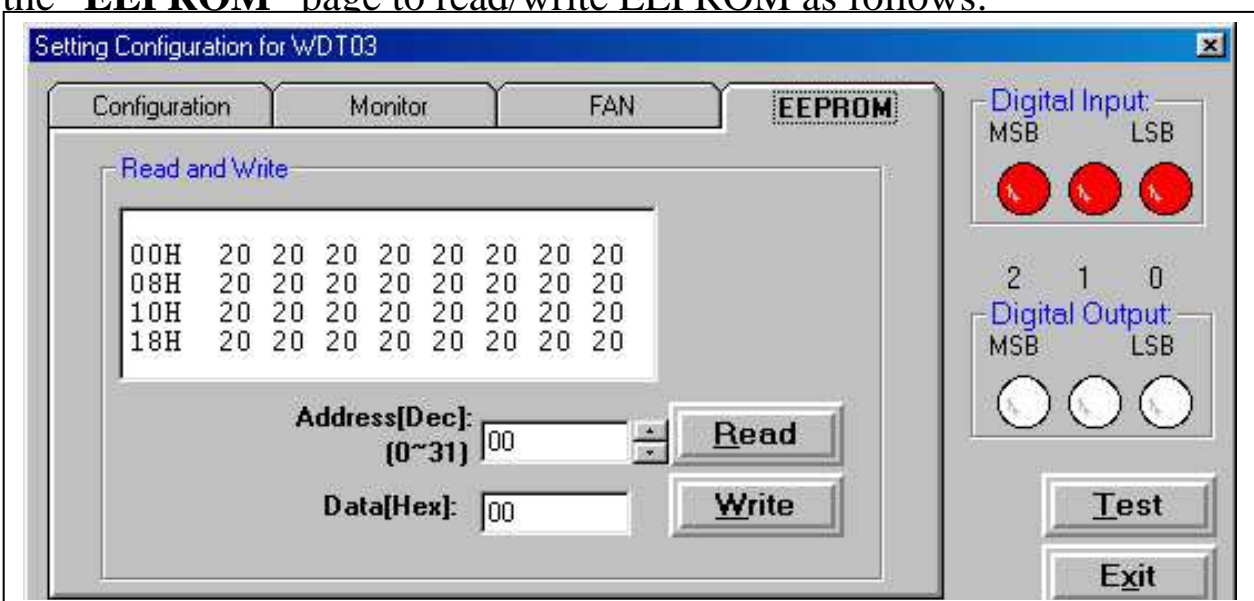
2.3.3 Fan Speed Monitoring and PWM Control

The WDT-03 provides 3-channels of fan speed sense and 3-channels of PWM control to monitor fan speed & to control the duty cycle of fan. The fan speed will be shown in “FAN” page. Scroll the “PWM” scroll bar to change PWM duty cycle and the result will be shown immediately.



2.3.4 Read and write EEPROM

The WDT-03 provides 32 bytes EEPROM for user to store data. Click the “EEPROM” page to read/write EEPROM as follows:



Press “Read” button to read data from EEPROM or press “Write” button to

write data to EEPROM.

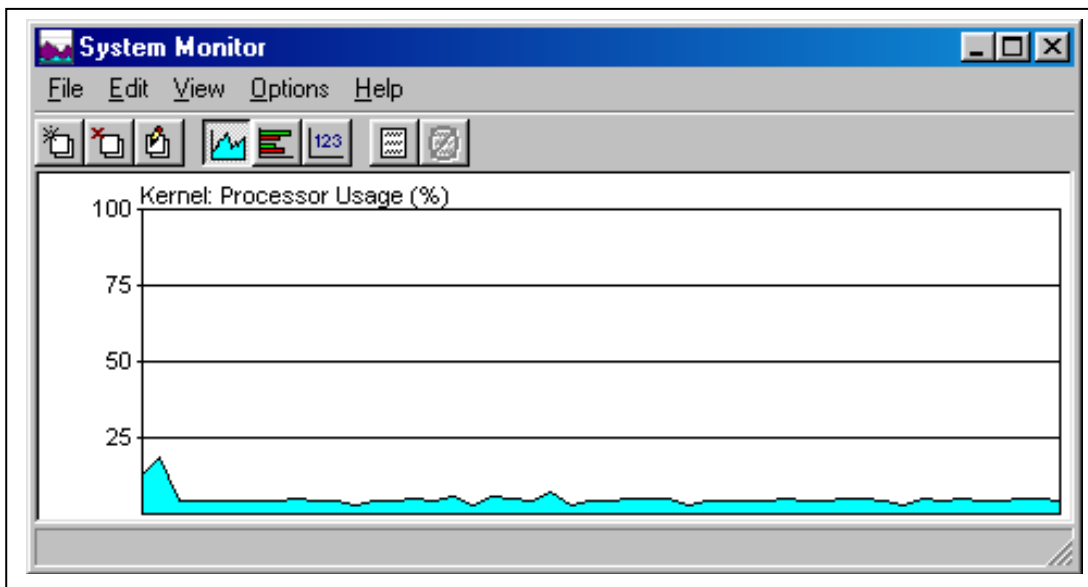
2.4 Auto Monitoring Utility for WDT-03

The WDT-03 is designed for a wide range of application. For example a video record center or a computer workstation or other application which needs a WDT system to monitor system status & alarm/reset the system when it is crashed for abnormal conditions.

The WDT-03 provides an auto monitoring utility software, its features are given as follows:

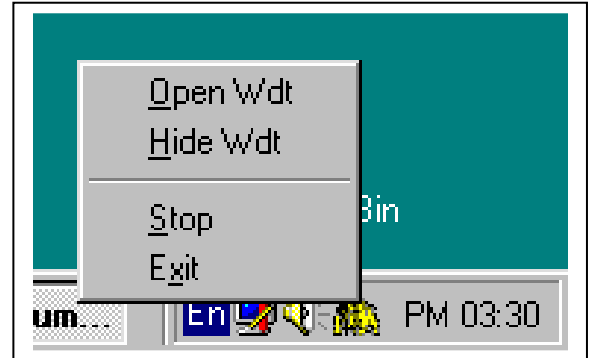
- can be executed under windows 95/98/Me/2000 or NT
- can auto monitor the system status such as bus voltage, environment temperature & fan speed
- can alarm by digital output if the system status is out of bound
- can reset & restart application program when system is crashed for abnormal conditions

The auto monitoring utility will execute automatically when windows start and resident in system tray. In NT system, this utility could logon into administrator count automatic when restart windows NT. The auto monitoring utility consumes very small system resource as following:



2.4.1 Start Auto Monitoring Utility

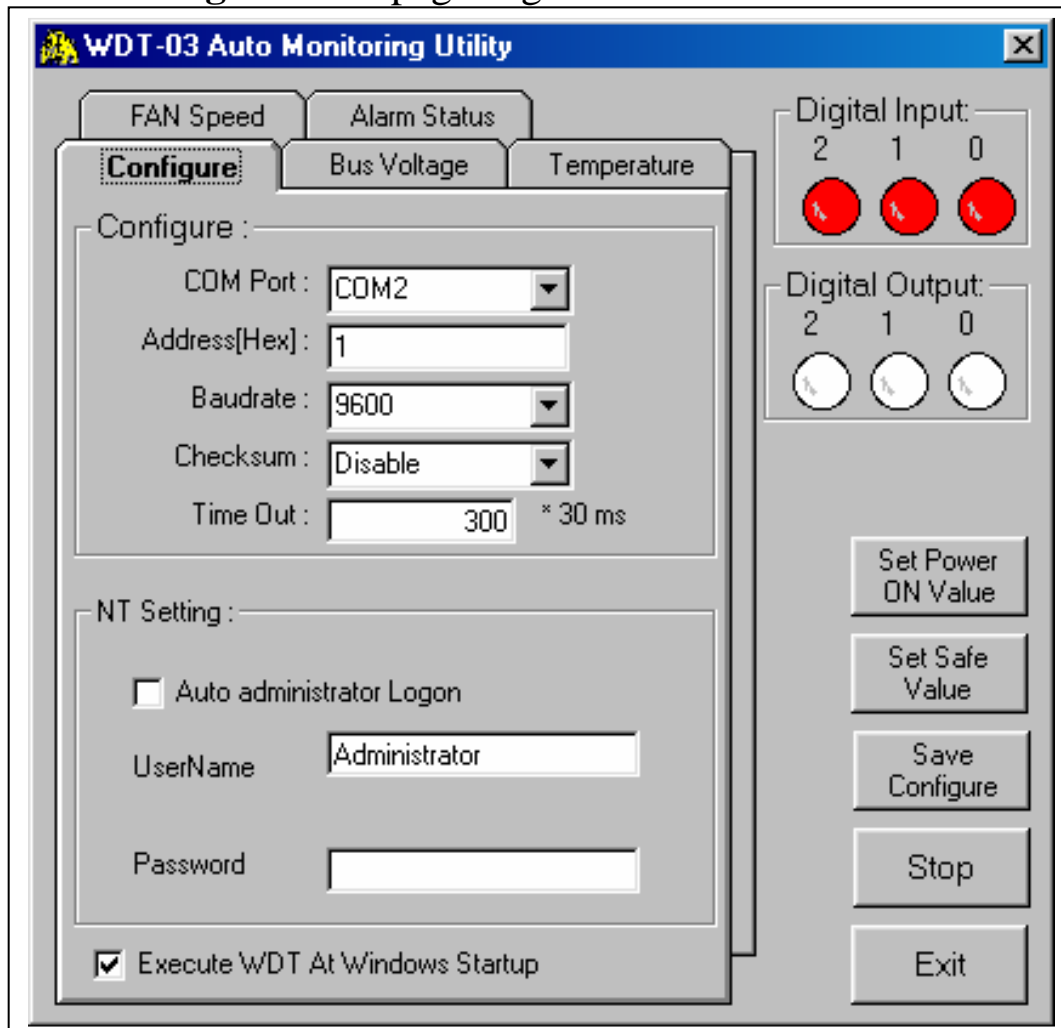
After the auto monitoring utility is installed, it will automatic execute when windows start to execute. The icon of auto monitoring utility will be shown in system tray. Click the WDT-03 icon in system tray and select "**OPEN WDT**" to open main windows of auto monitoring utility.



When the auto monitoring utility is first executed, please press "**OK**" to configure the auto monitoring utility in the following section.

2.4.2 Configure the Auto Monitoring Utility

The "**Configuration**" page is given as follows:

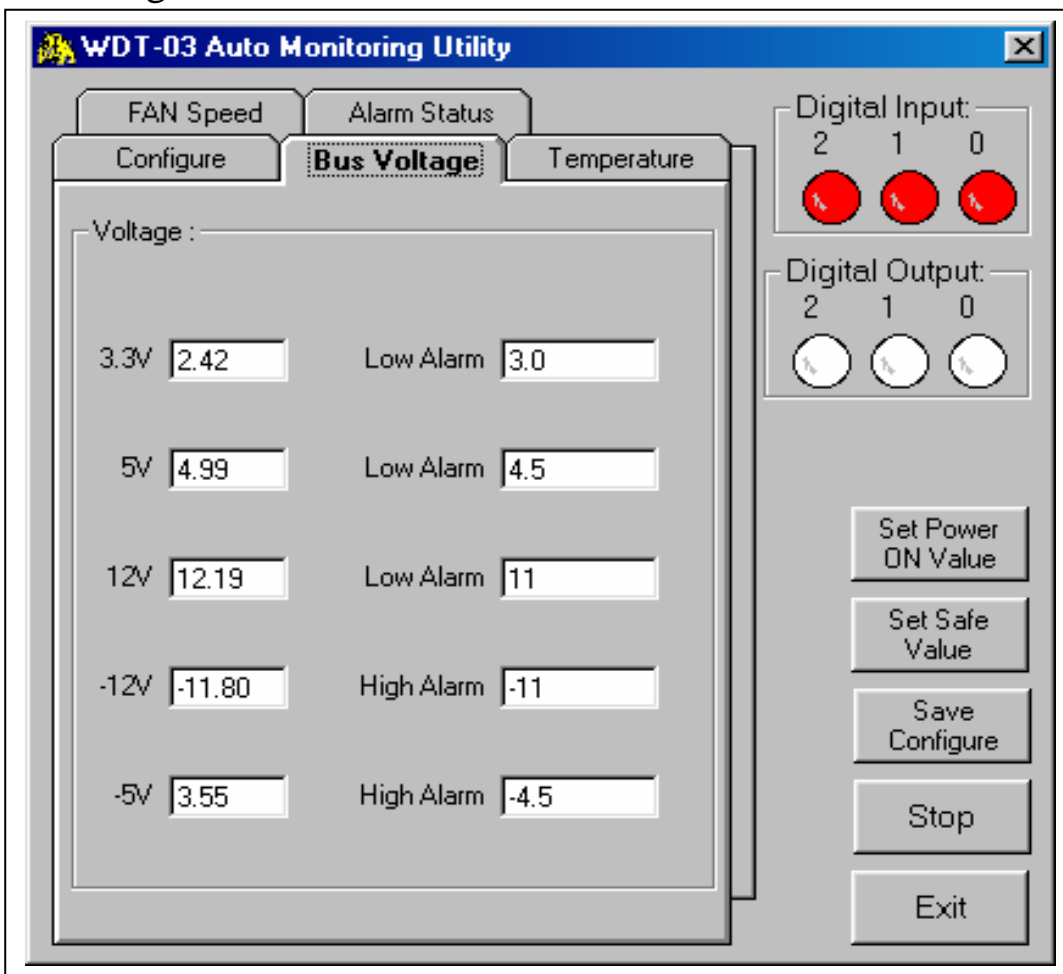


- **COM Port:** The serial port which is connected to WDT-03
- **Address:** module address of WDT-03, from 0xff to 0xFF
- **Baudrate:** baud rate of WDT-03, from 300 to 115200 BPS
- **Checksum:** checksum state, can be enable or disable
- **Time Out:** watchdog timer timeout value setting, from 3 to 196605 ms
- **Auto administrator Logon:** Automatic logon into Windows NT system with administrator count at next time
- **Execute WDT At Windows Startup:** create a short in windows startup folder and execute WDT when windows start to execute

After change all setting, press "**Save Configure**" button to save this special configuration.

2.4.3 Bus Voltage

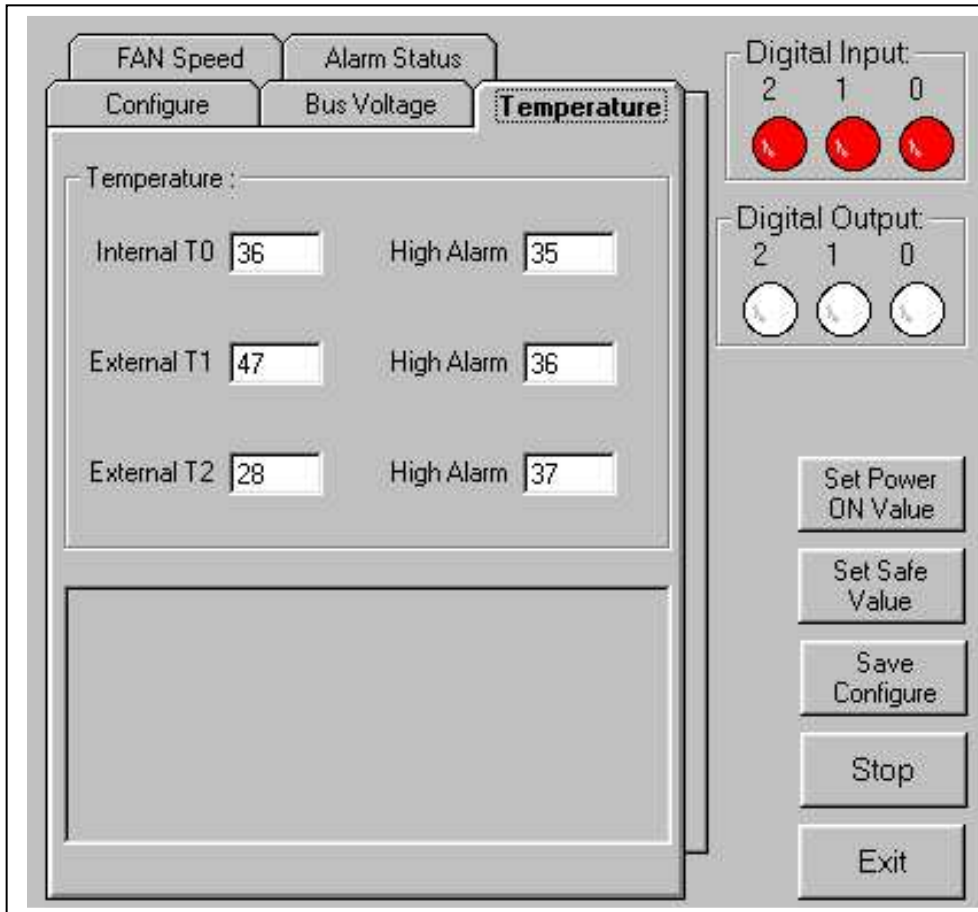
The bus voltages will be shown in the "**Bus Voltage**" page as following:



The field will display "**Error**" if the reading data is invalid.

2.4.4 Environment Temperature

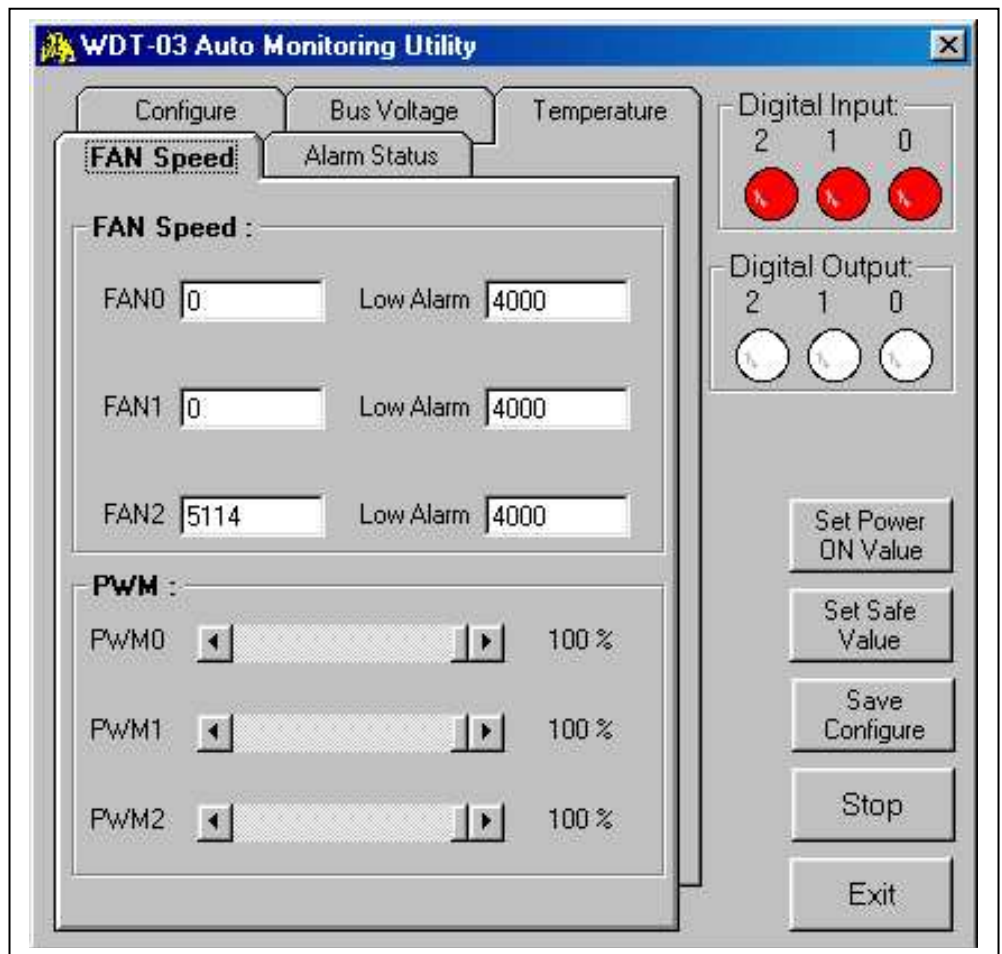
The environment temperatures will be shown in “Temperature” page as following:



- The field will display "**Error**" if the reading data is invalid.
 - "**High Alarm**": When the temperature is large then its "**High Alarm**" setting & its alarm status in Sec. 2.4.7 is enable, the D/O channel-0 will turn ON to drive external device for alarm message.
 - After change all setting, press the "**Save Configure**" button to complete the setting process.
 - "**Stop**" button → disable the host watchdog timer
 - "**Exit**" button → disable the host watchdog timer & exit to system
- (the following settings are designed for D/O control)**
- "**Set Power On Value**" button → set current value of D/O as its power on value.
 - "**Set Safe value**" button → set current value of D/O as its safe value.
 - click digital output icon → will toggle the current state of digital output (the red icon indication ON and the white icon indication OFF)

2.4.5 Fan Speed and PWM Cycle Setting

The WDT-03 provides 3-channel of fan speed monitor and 3-channel of PWM control. User can monitor fan speed or control fan PWM output duty cycle in the “**FAN Speed**” page. Scroll the PWM scroll bar can change the PWM duty cycle immediately.



- The field will display "**Error**" if the reading data is invalid.
- “**LOW Alarm**”: When the FAN speed is lower than its “**Low Alarm**” setting & its alarm status in Sec. 2.4.7 is enable, the D/O channel-0 will turn ON to drive external device for alarm message.

2.4.6 Digital Input/Output

The status of digital input will be shown in screen immediately. The red icon indicates High and the white icon indicates Low. For digital input, the open input will get High. Connect the digital input to GND will get Low.

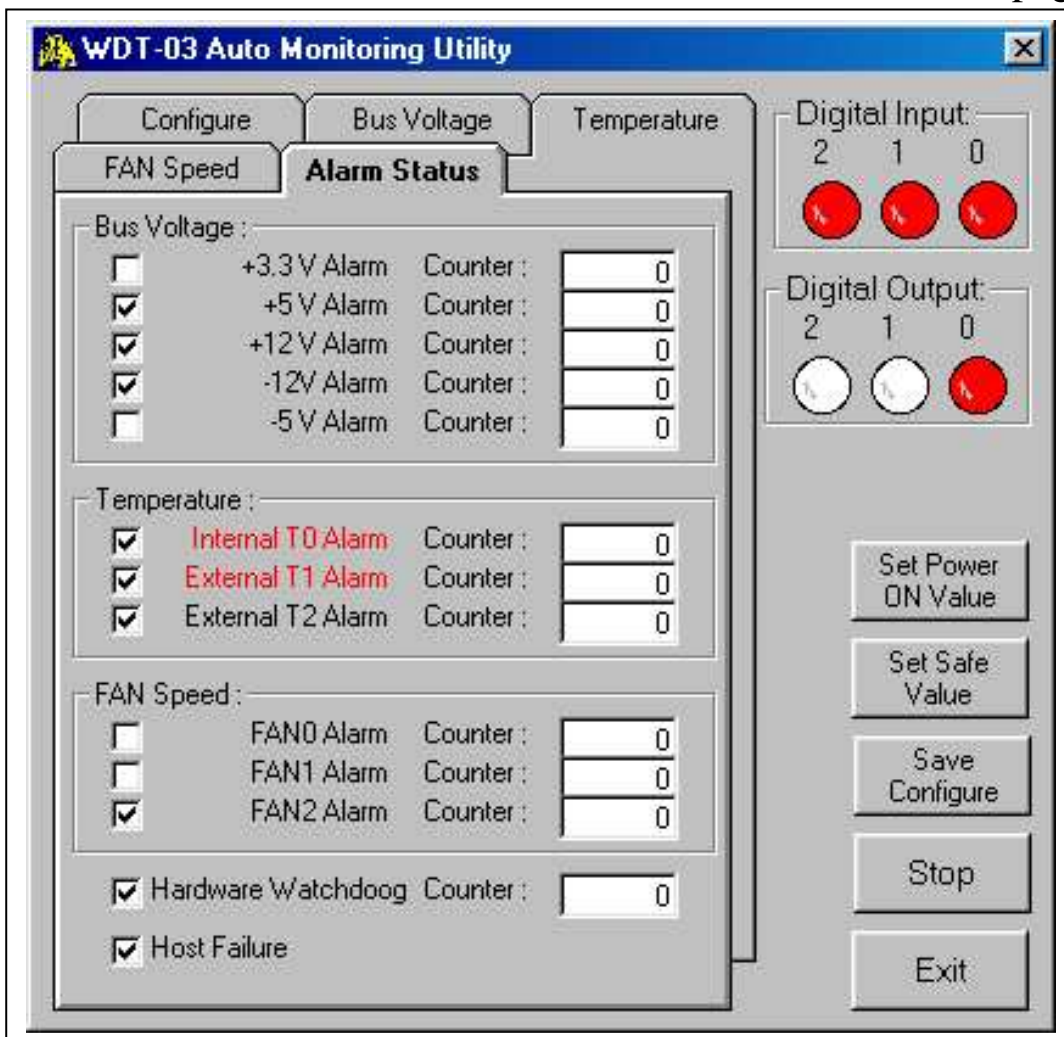
The D/O channel-0 is reserved for alarm. Refer to Sec. 2.4.7 for more information. The channel-1 & channel-2 of D/O can be controlled by user. Click digital output icon will toggle the current state of digital output (the red icon indication ON and the white icon indication OFF).

2.4.7 Alarm Status

WDT-03 auto monitoring utility can turn D/O channel-0 ON to drive external device for alarm message. The alarm sources are given as follows:

- Bus voltage under low limit or over high limit
- Temperature over high limit
- Fan speed under low limit
- Hardware watchdog failure (for WDT-03)
- Host watchdog (for local host-PC)

All alarm sources can be enable/disable in “**Alarm Status**” page as follows:



- The default alarm status is disable default
- When alarm event was happen the label will flash and the counter will increase by every 1 minute.
- The alarm will be latch and need press the label to clear alarm status.
- If one of alarm sources is active, the D/O channel-0 will be ON.
- “**Save Configure**” button → save all configuration data

2.4.8 Set Power ON Value

The "Set Power ON Value" button can be used to set the current D/O value as its power on value. The D/O will go to its power on value in PC's power on stage.

2.4.9 Set Safe Value

The "Set Safe Value" button can be used to set the current D/O value as its safe value. If the host watchdog is failure, all D/O will go to its safe value immediately.

Note: the auto monitoring utility use channel-0 of D/O for alarm. So the safe value of channel-0 should set to **High** for proper operation.

2.4.10 Save Configure

The "Save Configure" button can be used to save all configuration data introduced from Sec. 2.4.1 to Sec. 2.4.7. The auto monitoring utility will automatically load this configuration data when it starts to execute.

2.4.11 Press Ctrl + Alt + Del to test WDT-03

Press **Ctrl + Alt + Del** at the same time in Windows will force the system to stop all process. It can be used to simulate the host-PC's failure. After the watchdog timer is up, the WDT-03 will reset this host-PC.

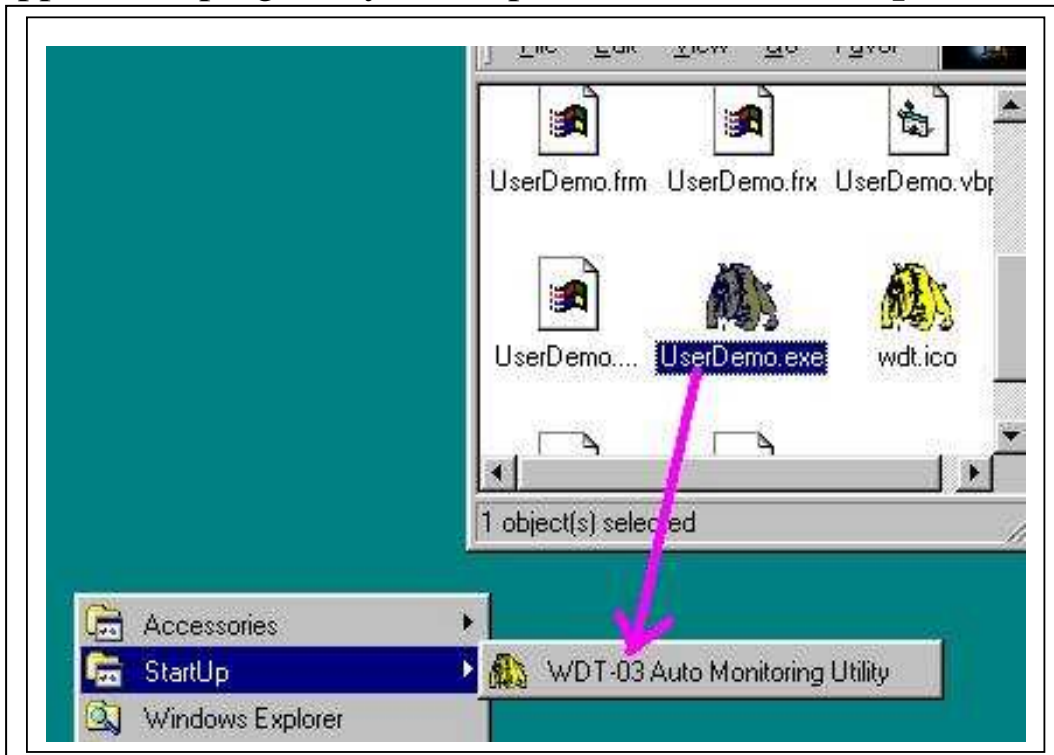
2.4.12 Stop and Exit Auto Monitoring Utility

The "STOP" button can be used to disable the watchdog timer. The "EXIT" button will disable the watchdog & exit the auto monitoring utility. If you **shutdown** the windows without "STOP" or "EXIT" the WDT-03, the host-PC will be reset again by watchdog hardware.

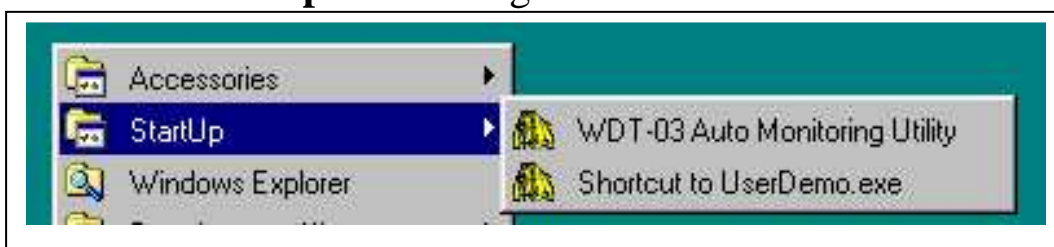
2.5 How to Execute a Program When Windows Startup

When windows startup, the program that place in “**StartUp**” folder will be automatic executed. If any user program desire to be executed when windows startup, it may need to create a shortcut in this folder. After the auto monitoring utility is installed, it will be placed in the “**StartUp**” folder automatically.

The **UserDemo.exe** is a sample program used to simulate user’s application program, you can place it to the “**StartUp**” folder as following:



Then the “**StartUp**” folder is given as follows:



Now the **auto monitoring utility** & **userdemo.exe** will be executed automatically when windows startup.

3. Software Driver

In general, user **does not have to** modify his application program. But the WDT-03 is very powerful & auto monitoring utility is designed for general application. So in some special application, user will not use auto monitoring utility & write his MMI & control algorithm based on his special requirement.

The command set of WDT-03 is compatible to our famous 7000/8000 series. So all software drivers provided for 7000/8000 family can be used for WDT-03.

3.1 Driver Installation & User's Manual

(1) DOS Utility Installation

Copy all files on \CD\NAPDOS\7000\NAP7000S*.*

Manual → \CD\NAPDOS\7000\NAP7000S\nap7000s.pdf

(2) 7000 Utility Installation & User's Manual

Double Click on \CD\NAPDOS\7000\7000Util\SETUP\Setup.exe

Manual → on-line help after the 7000 utility is installed

(3) NAP7000P Installation & User's Manual

Double Click on \CD\NAPDOS\7000\NAP7000P\SETUP\Setup.exe

Manual → read the README.TXT in the same folder

(4) NAP7000D Installation & User's Manual

Double Click on \CD\NAPDOS\7000\NAP7000D\SETUP\Setup.exe

Manual → read the README.TXT in the same folder

(5) NAP7000X Installation & User's Manual

Double Click on \CD\NAPDOS\7000\NAP7000X\SETUP\Setup.exe

Manual → read the README.TXT in the same folder

(6) NAP7000OPCSvr Installation & User's Manual

Double Click on \CD\NAPDOS\7000\NAP7000OPCSvr\SETUP\Setup.exe

Manual → read the README.TXT in the same folder

3.2 DOS Driver for Host-PC

The NAP7000S is designed for 7000 & 8000 series under DOS environment. It provide utility, demo program, performance evaluation and data acquisition subroutines for 7000 & 8000 fmaily. It can be executed under DOS & Windows 3.1/95/98/Me/2000/NT. All program sources of NAP7000S are given in the companion CD.

The driver sources of diagnostic utility are given as following:

UART.H → header file
UART.C → library file
TEST.C → application file
TEST.PRJ → project file

The compiler and link for TC are given as below:

1. Set DOS PATH to TC
2. Using TC integrated environment to set *PROJECT* to TEST.PRJ
3. Press **F9** to MAKE the project file (TEST.PRJ)
4. Execution file = *TEST.EXE*

The starting screen of DOS diagnostic utility is given as following:

```
***** Diagnostic Program *****
* STATUS : COM=1,Baud_Rate= 9600,Checksum=DISABLE *
*-----*
*      0 : init (for all module) *
*      1 : search (1200 to 115200) (for all module) *
*      2 : send command (for all module) *
*      3 : demo ($012,$01M,$01F) (for all module) *
*      4 : Host Watchdog test (for 7021) *
*      5 : RS-485 network testing (for all module) *
*      Q : quit *
***** Press Keyword *****
```

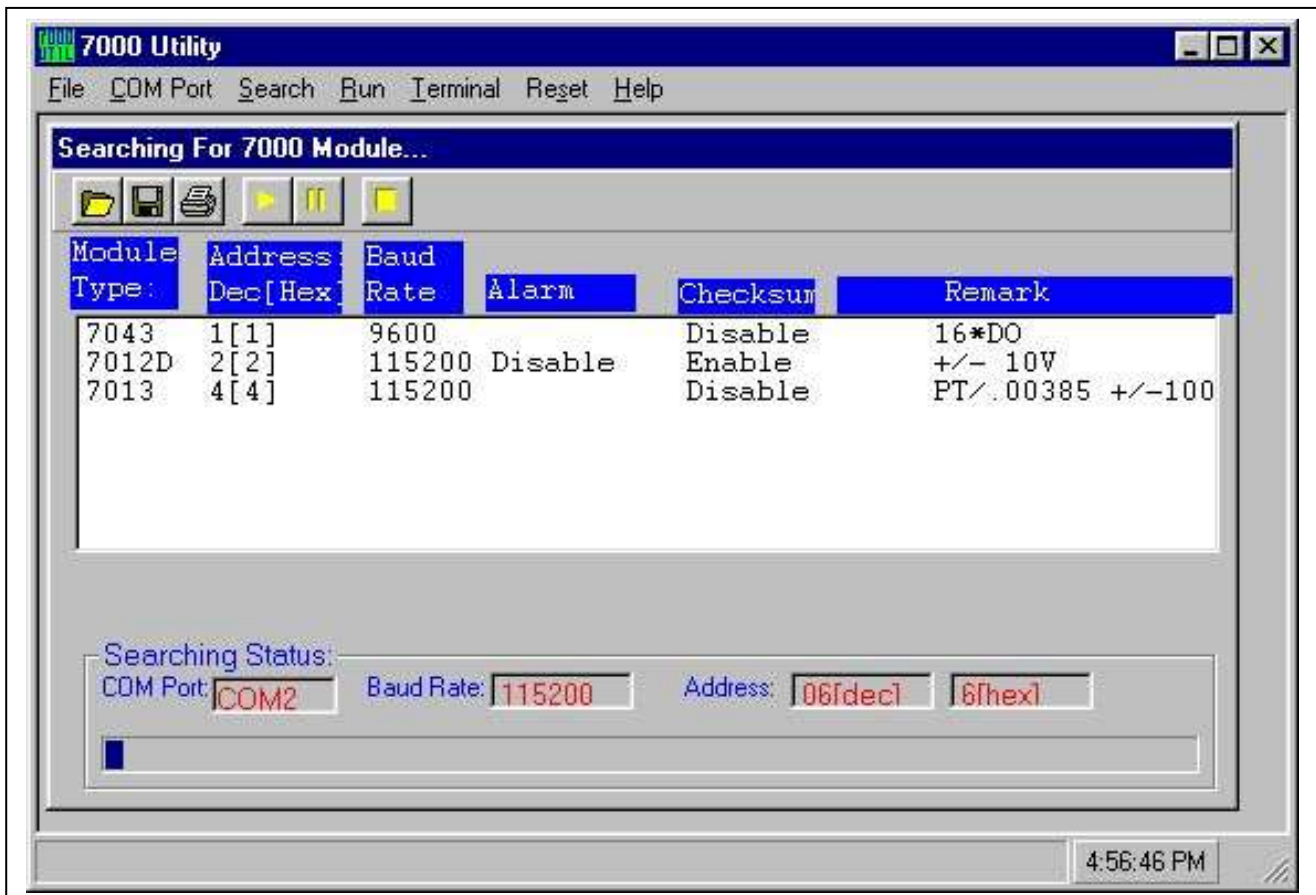
3.3 Windows Utility for Host-PC

The 7000 utility is designed for 7000 & 8000 series under Windows environment. Refer to Sec. 3.5 for 7000 utility installation. It provide diagnostic, module setting & data logger utility for 7000 & 8000 fmaily. **It provides on-line help about operation & documentation of 7000 utility. You can refer to \napdos\7000>manual\GetStart.pdf for easy starting of 7000 utility.**

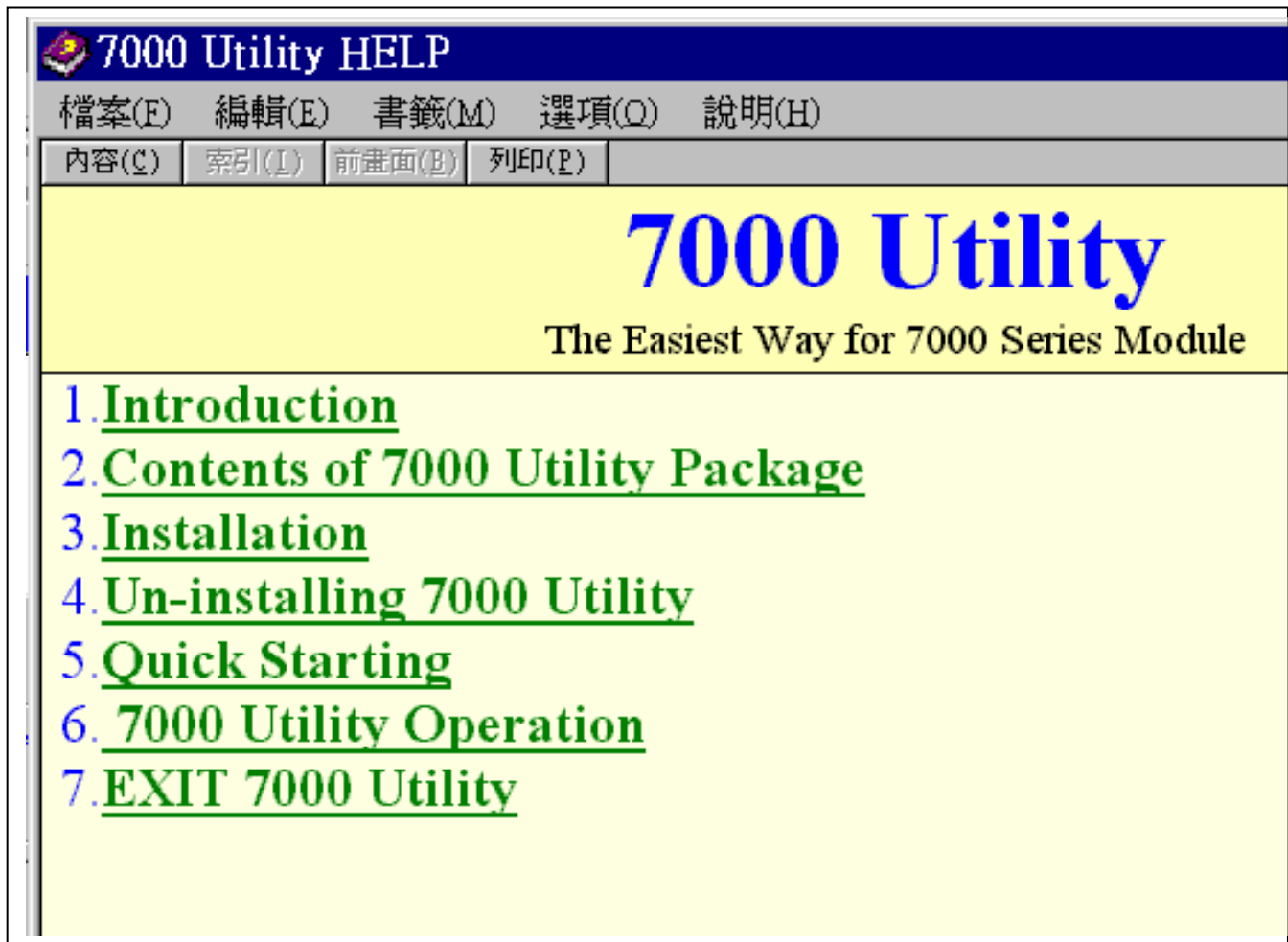
The 7000 utility provide 7 functions given as following:

- File : configuration save/load
- COM Port: configuration of PC COM port setting
- Search : search all 7000/8000 modules installed in the RS-485 network
- Run : diagnostic utility & data logger
- Terminal : debug terminal
- Reset : reset module
- Help : on-line help

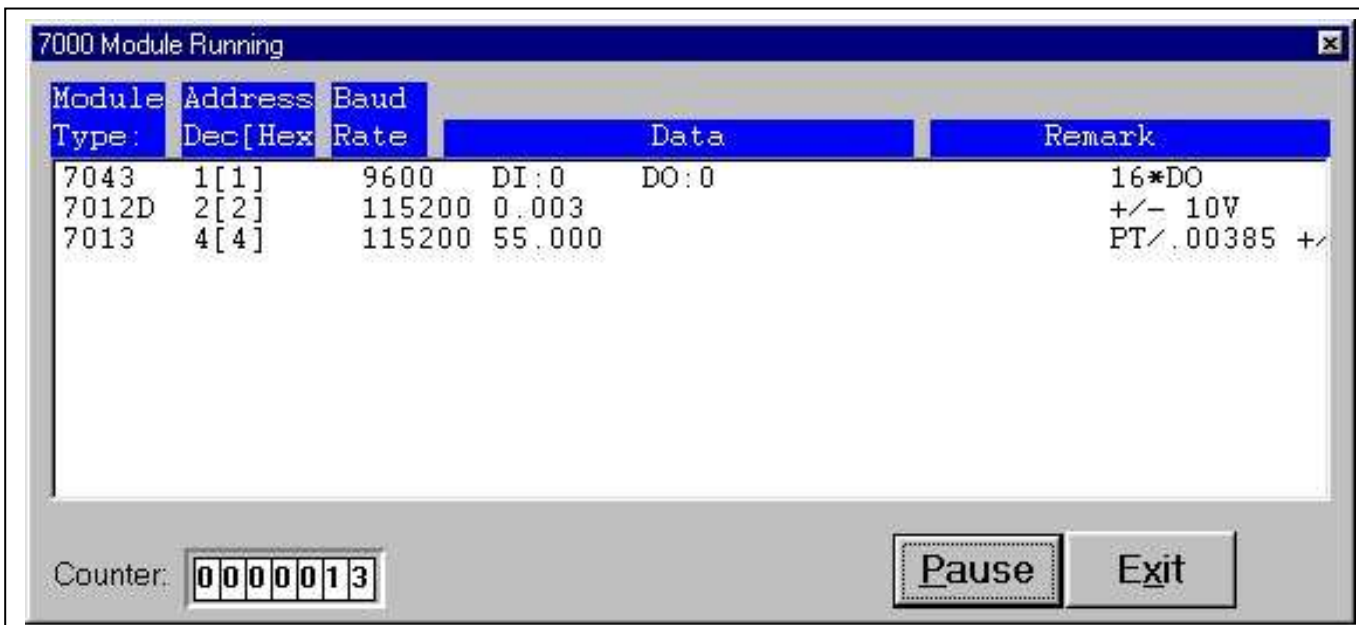
The main window of 7000 utility is given as following:



The on-line help of 7000 utility is given as following:



The module running window of 7000 utility is given as following:



3.4 Windows DLLs for Host-PC

The NAP700P is Windows DLLs designed for 7000 & 8000 series under Windows environment. Refer to Sec. 3.6 for NAP7000P installation. The NAP7000P provide windows compatible DLLs which can be called by. VC++, BC++, VB, Delphi, BC++ Builder or other package. The features of NAP7000P are given as following:

1. Provide general-purpose 7000/8000 series module send/receive functions.
2. Provide high performance 7000/8000 series module application functions.
3. Multi-speed demos.
4. Multi-data-format demos.
5. Lots of demo programs with completely source code.

After the user install the NAP7000P, the files I7000.DLL and UART.DLL will be copied into the user's system folder as follows :

C:\Windows\system (Windows 95/98)

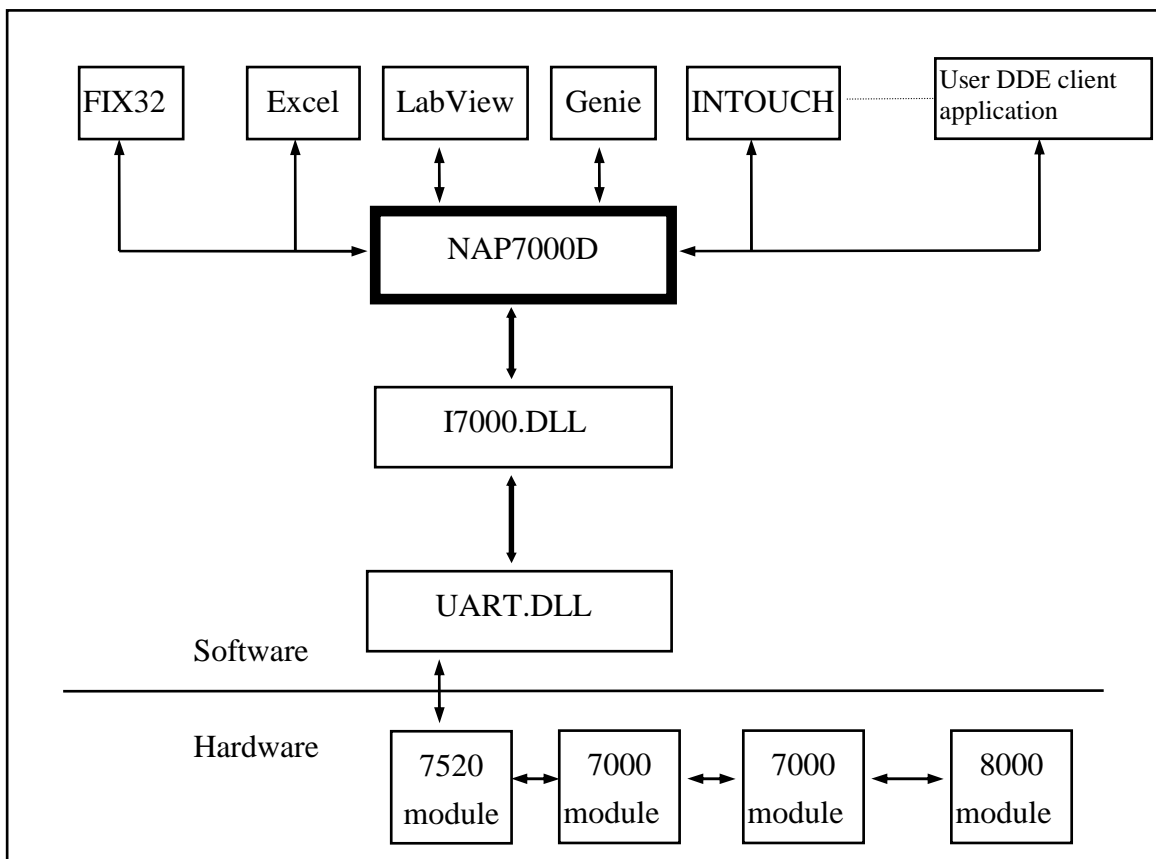
C:\WinNT\System32 (Windows NT)

To use these functions of I7000.DLL and UART.DLL, the user must declare these function's prototype(interface) for development tools that the user using. NAP7000P provides the completely declaration files of I7000.DLL and UART.DLL for VC++, VB, Delphi and BCB. The user can copy these declaration files into the project folder and include them into the user's project. And then the user can use these functions as the function that provided by the development tools that the user using.

3.5 Windows DDE for Host-PC

The NAP700D is Windows hot-link DDE(Dynamic Data Exchange) server designed for 7000 & 8000 series. Refer to Sec. 3.6 for NAP7000D installation. The DDE is a communication protocol that enables to exchange data between windows application. The NAP7000D pass the data acquired from 7000/8000 series module to other application by means of the hot-link. Or a user can control the 7000/8000 module using the NAP7000D after establishing these hot-link conversation. The NAP7000D is a mouse driven system that provides the ability of developing 7000/8000 application by just clicking mouse.

The block diagram of NAP7000D is given as follows:



After the nap7000D is installed, the sub-directory is given as follows:

- [Base Directory] <-- the directory you selected to setup
- |- \icpcon.cfg <-- configuration file for NAP7000D.
- |- \NAP7000D.EXE <-- the execute file.
- |- \NAP7000D.DOC <-- this user's **manual** for NAP7000D

There is a user manual of HTML format stored in
 \napdos\7000\nap7000d\html\index.html.

3.6 Windows OCX for Host-PC

The ActiveX Control (OCX) is designed for developing program easily and user friendly for **7000/8000 series module**. The user can plug the ActiveX Control (OCX) object easily and use the properties, methods and events of the object to programming easily. Before using the ActiveX Control (OCX), please install the software package firstly. Please refer to the following content to complete the installation process.

The ActiveX Control (OCX) support the Windows 95/98/NT 4.0 and Windows 2000 or above. It doesn't support the Windows 3.1 or Windows NT 3.51(the format of type-lib doesn't the same between NT 3.51 and NT 4.0).

The NAP700X is OCX library for 7000 & 8000 series. Refer to Sec. 3.6 for NAP7000X installation.

Directory Tree

After installed, these files : NAP7000X.OCX, NAP7000X.tlb, I7000.DLL

and UART.DLL will be copied into

C:\WINDOWS\SYSTEM(for Windows 95/98 platform)

C:\WINNT\SYSTEM32(for Windows NT/2000 platform)

And the NAP7000X.OCX will create the registry value for itself automatically.

The user also can register the NAP7000X.OCX manually by execute the "RegSvr32 NAP7000X.OCX" on the command mode.

For work correctly, the user's system will need the following DLL files:

Olepro32.dll, Oleaut32.dll, Mfc42.dll, Msvcrt.dll

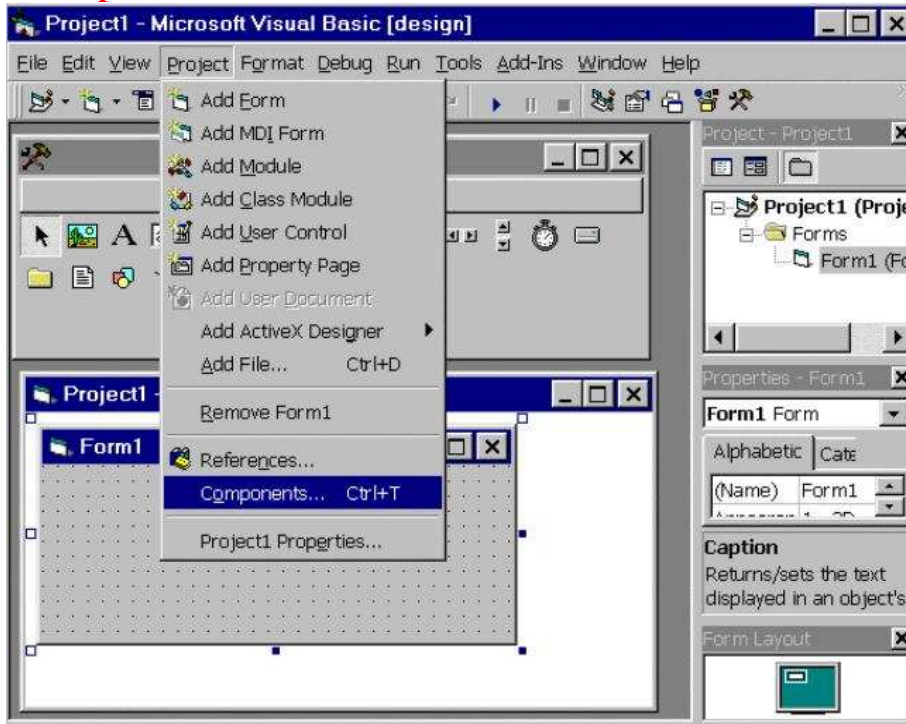
These files will be found in the folder:

C:\Windows\system ==> Windows 95/98/Me

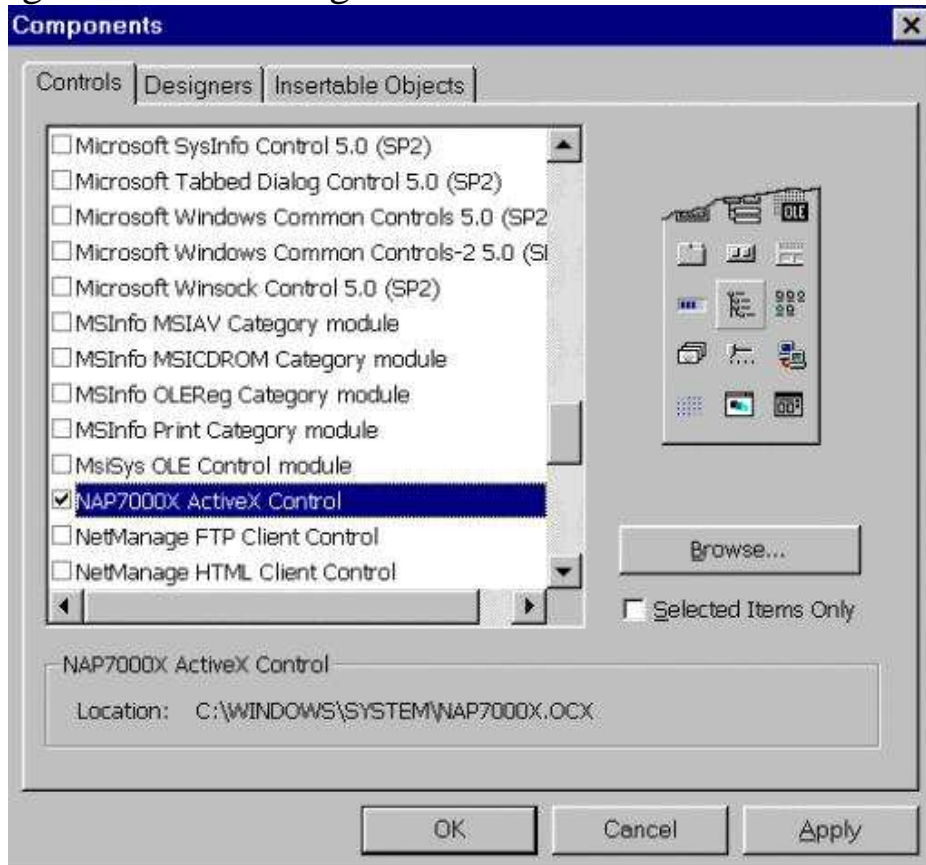
C:\WinNT\system32 ==> Windows NT

Using NAP7000X with Visual Basic

Step 1: Adding the ActiveX Control (OCX) into the VB toolbox. Clicking the **Project/Components** in Visual Basic 5.0 menu.



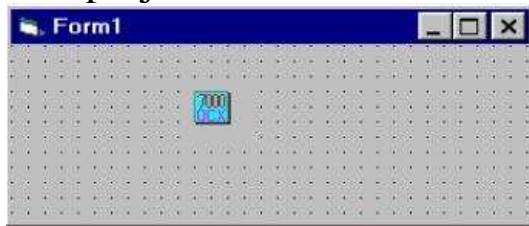
Step 2: Checking the item **NAP7000X ActiveX Control** in the Controls dialog box then clicking **OK** button.



Step 3: After step 2, the icon of **NAP7000X (7000 OCX)** will display in the ToolBox.

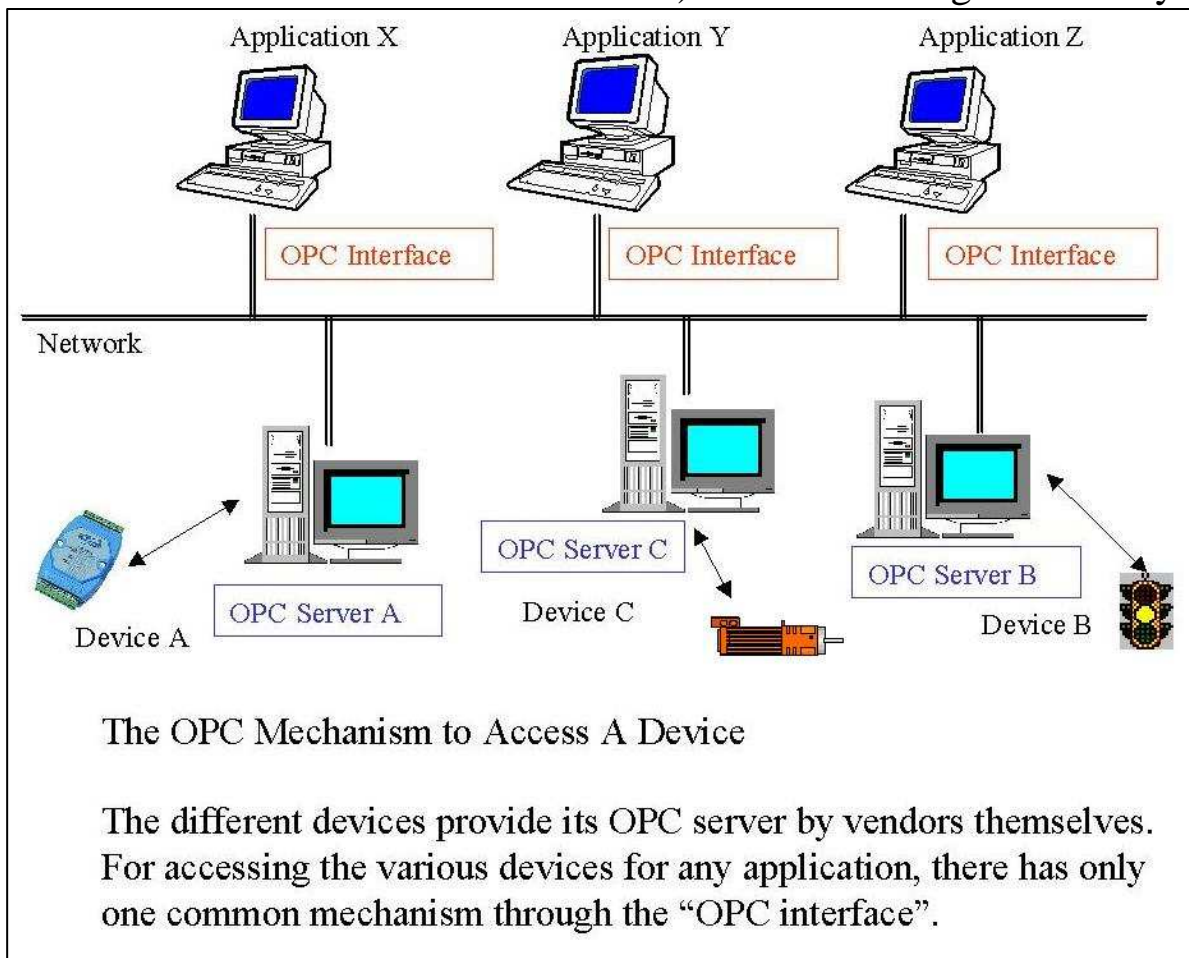


Step 4: Create a new standard project, then add an **NAP7000X (7000 OCX)** object into this project.



3.7 Windows OPC for Host-PC

The NAP7000OPCSvr supports 7000 & 8000 series. Refer to Sec. 3.6 for NAP7000OPCSvr installation. The NAP7000 OPC DA Server uses an Explorer style user interface to display a hierarchical tree of modules and groups and their associated tags. Groups are like subdirectories that logically group tags. A module may have subgroups that all have tags within them, but all of the tags really belong to their module when they are scanned to perform I/O. (The "OPC" is stand for "OLE for Process Control" and the "DA" is stand for "Data Access".) The block diagram of a system which



supports OPC interface is given as following:

All the configuration settings can be save into configuration file by click the menu item "[File/ Save As...](#)". The last configuration file will be load automatically when the OPC server is launched every time.

The Setup program:

You could find the setup.exe program on the CD disc.

If you got the program from our web-site, please

1. Unzip the opcdac.exe (Self-Extraction file) to a temp folder.
(For example: C:\DACSetup\)
2. Unzip the Disk1.exe (Self-Extraction file) to a temp folder.
(For example: C:\OPCSetup\)
3. Copy the other related *.cab (Data2.cab, Data3.cab) files into the temp folder. (For example: C:\OPCSetup\)

Installation Steps

Both the client site and server site needs to install the following packages.

Windows 95

step 1. Install DCOM95.EXE.

(It can be download from the Microsoft's web-site.)

step 2. Install DCM95CFG.EXE.

(It can be download from the Microsoft's web-site.)

step 3. Restart Computer.

step 4. Install the "OPC Data Access Component".

(Zipped in the opcdac.exe as self-extraction file.)

step 6. Install the "NAP7000 OPC DA Server".

Windows 98/NT/2000

step 1. Install the "OPC Data Access Component" package firstly.

(Zipped in the opcdac.exe as self-extraction file.)

step 2. Install the "NAP7000 OPC DA Server" package.

Quick Start

Please follows these steps...

1. Using the 7000 Utility to setting modules.
2. Install the NAP7000OPC on user's system.
3. Run the OPC Server "[NAP7000OPCSvr.Exe](#)".
4. Refer to "[NAP7000OPCSvr.PDF : Section 1.1 Search Modules..](#)" to search modules on the RS-485 network.
5. Save the configuration by using "[File/Save](#)" menu item.
6. Close the OPC Server "[NAP7000OPCSvr.Exe](#)".
7. Run the user's client program and connect to OPC Server.
(Please refers to user's manual for the client software that provided by vendor.)

3.8 Linux Driver for Host-PC

Since serial port is the standard device supported by Linux, user can use serial port via the standard device file system, for example, device files `/dev/ttyS0` and `/dev/ttyS1` are generally mapped to first (COM1) and second (COM2) serial ports. If you are an experienced user who is familiar with the serial device manipulation, write codes to manipulate serial port by yourself is recommended, you can fully control the behavior and get the maximum ability of the serial port.

Serial port manipulation is maybe too sophisticated to user new to Linux programming. The `i7k` is a small source code package that simplifies the serial port manipulation, provides easy programming interface to the serial port, allows user's program to communicate the 7000 or 8000 modules by only regards the basic serial communication characters – baud-rate, parity-check, data-bits, and stop-bit. The `i7k` also provides a simple console tool – the `i7kon`, a text utility for searching connected modules and sending manual command.

Another software comes with the `i7k` is the `iWarp`, a demonstration program that reads message from a connected TCP socket, and warps the message to a local serial port, and vice versa. In other word, you can set up the `iWarp`, and communicate with your 7000 or 8000 modules remotely and world widely.

For user who is familiar with the I7000 programming interface on Microsoft Windows but Linux, `i7k` wraps functions of the `I7000.DLL` for Windows into package. If you like to use it, just link it with your application. It is also ideal for migrating your Windows application to Linux. The `i7k` package is a zipped source code tarball that can be found from ICP DAS website:

Download ► I-7000 Series Module (ICP CON) ► 7000 Series Toolkits ► I7000 for LINUX.

Installation

Put the `i7k` package, e.g. `i7k_0.2.0.tar.gz`, into your work directory which you have the read/write privilege, and unzip it by command
`tar -zxvf i7k_0.2.0.tar.gz`
or `gunzip -c i7k_0.2.0.tar.gz | tar -xvf`

In this case, directory `i7k_0.2.0` will be created under the work directory. For convenience of access to directory, it is a good idea to put a

symbol link to `i7k_0.2.0` by command

```
ln -s i7k_0.2.0 i7k
```

Above command will create a symbol “i7k” links to “i7k_0.2.0.”

Once you have linked the symbol “i7k” to “i7k_0.2.0,” change your work directory into `i7k` and build package by command

```
cd i7k
make clean
make
```

Check the result of `make` for error (you can ignore warnings.) If you get trouble and don’t know how to shoot it, please contact us by email to service@icpdas.com and report the error message in mail.

Read/Write Privilege for Serial Device

Once the `make` command is completely successful, you are ready for test and programming your application. But before you run your program, you must enable the access to serial ports. Since modules are connected on serial port, you must have the read/write privilege to the system’s serial device for communicating with modules. Linux generally maps serial ports

COM1 to `/dev/ttyS0`

COM2 to `/dev/ttyS1`

COM3 to `/dev/ttyS2`

COM4 to `/dev/ttyS3`

If you use a non-standard serial device, contact your device provider for setup information. Check your serial device, e.g. `/dev/ttyS0`, by command

```
ls -l /dev/ttyS0
```

```
crw-rw---- 1 root tty 4, 64 Jul 18 1994 /dev/ttyS0
```

Above message means, if you are not user “root” and not a member of group “tty,” then you haven’t the read/write privilege to device `ttyS0`. To gain read/write privilege of `ttyS0` to all user, you should “su root” that make you became root temporarily then command

```
chmod a+rw /dev/ttyS0
```

And check again

```
ls -l /dev/ttyS0
```

```
crw-rw-rw- 1 root tty 4, 64 Jul 18 1994 /dev/ttyS0
```

Now you obtain the access to `ttyS0`. However, opened read/write privilege maybe damage your system security. If you are not familiar with privilege to the file system of Unix, please read document of your Linux box for detail information.

i7kon

The i7kon is a text mode console tool that allows you to search modules and send manual commands via the opened serial port. The i7kon can be run from a text mode system console or an X terminal. You can run the i7kon under a terminal connected to a remote host by the Telnet for sending command to the modules connected on the host.

The i7kon uses the library NCURSES (5.0) to manipulate screen. If your system hasn't the NCURSES installed, you can get it from <http://www.gnu.org/software/ncurses/ncurses.html>. Again, you must have the read/write privilege to serial device to run i7kon.

Picture is a screenshot of the i7kon run under the Eterm X terminal.



The command "Open Port" opens a serial port with parameters before other operation. The command "Send Command" allows you to send a manual command to the opened serial port. The command "Find Learning Kit" searches I7000 modules by address 01 for 7012, 02 for 7021, and 03 for 7060. The command "Search Devices" searches modules by the given ranges of address and baud-rate. The command "Close Port" closes current opened serial port. The command "Exit" closes serial port if it is opened, and exits.

The frame DEV displays the names of modules found by search-command. The frame SCOPE shows the data sent to and received from the port. Use Up-down arrow-keys and press Enter to choose and execute command from frame MENU. The frame MESSAGE displays the commands' status, tips,

and response.

iWarp

The iWarp is a demonstration program that allows you to control your modules across the Internet. The iWarp has two parts – the server side **iw** and the client side **iwc**. The server side runs “iw” on the remote host which has 7000/8000 modules connected. The Client side runs “iwc” on the local host which intends to control the modules.

The iWarp is included in the i7k package, but the root-make does not build the iWarp by default. To build the iw (server,) just type “make” under the directory of iw. Before make iw, you should edit file “main.c,” change the definition of SIO_PORT for serial port assignment. Then you must edit file /etc/services to add the service “iwarp” since iw looks it for port number, for example, register iwarp as port number 1500 with protocol TCP

```
iwarp      1500/tcp  # Internet Warp Service
```

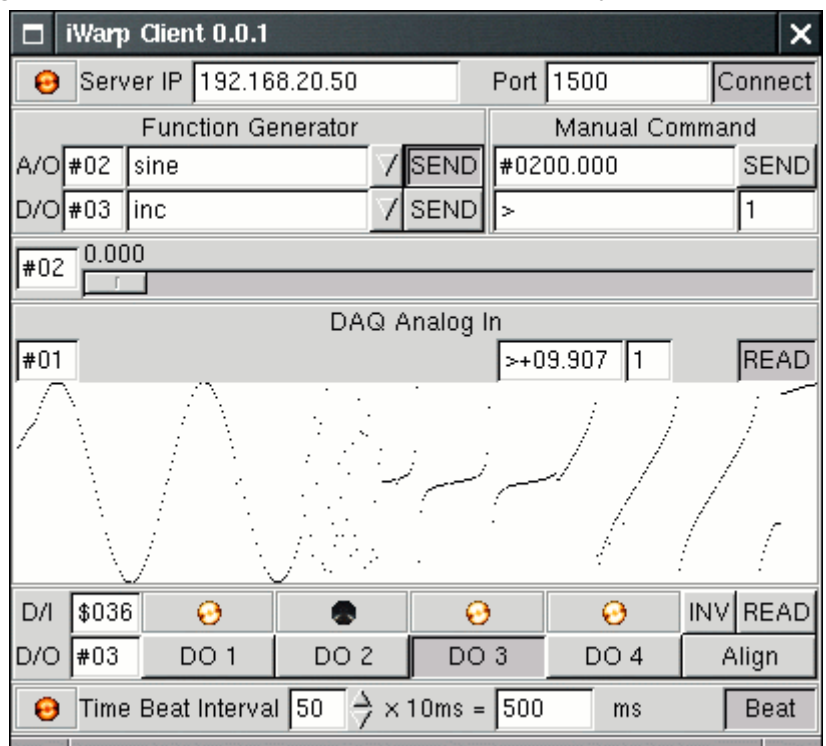
You must have read/write privilege to serial device to run iw.

The iwc (client) is an X GUI application that is written with GTK (1.2) library. If your system hasn't GTK installed, you can get it from <http://gtk.org/>. Type following commands under the iwc directory to build

```
make distclean  
sh configure      or  
sh ./configure  
make
```

Once make is completely successful, the iwc is ready to run from directory src.

Picture is the screenshot of the iwc run to connect remote host 192.168.20.50 at port 1500. The lamp left side of Server IP is turned on for what means the connection between client and server sides has been successfully set up.



Programming

The i7k is distributed with source code. You can modify it to suit your usage or just link it into your application. Once you unzipped the i7k package, there are seven source code modules under the sub-directory i7k; they are timer, sio, i7k, msw, i7000, i8000, and i87000.

The module timer contains functions for time-measurement that is used by module i7k. The module sio is the basic open/close manipulation of serial port. The module i7k is the basic send/read function for 7000/8000. Include file i7k.h and link modules timer.o, sio.o, and i7k.o into your application is enough for sending command to and reading response from your 7000 and 8000 series modules. Please read file README under directory i7k for more information, and read files SIO.TXT and I7K.TXT for function reference.

Following is the function briefs for module sio.c

sio_close: close a opened serial port

sio_open: open a serial port and set to default non-canonical mode

sio_set_baud: set baud-rate of a opened serial port

sio_set_canon: set a opened serial port to canonical mode

sio_set_noncan: set a opened serial port to non-canonical mode

sio_set_timer: set time-out for a opened serial port that works on non-canonical mode

And the function briefs for i7k.c

i7k_read: read response

i7k_read_cs: read response with checksum

i7k_readt: read response with timeout

i7k_readt_cs: read response with checksum and timeout

i7k_send: send command

i7k_send_cs: send command with checksum

i7k_send_read: send command then read response

i7k_send_read_cs: send command then read response with checksum

i7k_send_readt: send command then read response with timeout

i7k_send_readt_cs: send command then read response with checksum and timeout

Include sio.h and link sio.o or libi7k.a to use functions of sio.c. Include i7k.h and link i7k.o or libi7k.a to use functions of i7k.c.

Programming with I7000 Wrapped Functions

The modules i7000, i8000, and i87000 are wrapped from the I7000.DLL. If you like to use function that is like to the function provided by I7000.DLL, you should compile and link your program with file msw.h and library libi7k.a.

Please take notice that the difference of time-measurement between the Linux and the Windows when you are using the wrapped functions. The unit of serial port timeout is tenth of one second (1/10 sec) on Linux but millisecond (1/1000 sec) on Windows. The unit of the returned value of the time expended by the called function is microsecond (1/1000000 sec) on Linux but millisecond on Windows. For example

Function	Argument	Unit on Linux (Sec)	Unit on Windows (Sec)
Send_Receive_Cmd()	wTimeOut	1/10	1/1000
Send_Receive_Cmd()	wT	1/1000000	1/1000

Samples

The directory “samples” contains sample codes show you that how to use the basic i7k/sio functions and I7000 wrapped functions with library libi7k.a. Please read file README and source codes for further details.

3.9 LabVIEW Driver for Host-PC

LabVIEW is a industrial's graphical programming system developed by National Instruments. With LabVIEW, user can quickly designed the user interface and control program as a block diagram. The NAP7000V provides many driver VIs & demo VIs which can be used by LabVIEW 4.0 & above. These driver VIs will call DLLs supported by NAP7000P. The demo VIs will call driver VIs to control 7000/8000 family. So the user manuals of NAP7000V are all same as NAP7000P. Refer to Sec. 3.6 for NAP7000P installation.

After the NAP7000V is installed, the UART.DLL and I7000.DLL will be copied into

C:\WINDOWS\SYSTEM (for WINDOWS 95/98/Me)

or

C:\WINNT\SYSTEM32 (for WINDOWS NT).

The item "DAQPro NAP7000v" will be added into the "Add/Remove Programs" (under "Control Panel")

And the sub-directory is given as follows:

The default [Base Directory] is "\DAQPro\NAP7000v".

4. Command Set

Command Set Table

Command	Response	Description	Reference
%AANNTTCCFF	!AA	Set module configuration	Sec. 4.1
\$AA2	!AATTCFF	Read configuration	Sec. 4.2
\$AA5	!AAS	Reset status read back	Sec. 4.3
\$AA6	!(DATA)	Read digital in	Sec. 4.4
#AABBDD	>	Digital output control	Sec. 4.5
\$AAF	!AA(data)	Read the firmware version number	Sec. 4.6
\$AAM	!AA(data)	Read the module name	Sec. 4.6
~AA0	!AASS	Read system status	Sec. 4.7
~AA1	!AA	Reset system status	Sec. 4.8
~AA2	!AASTTTT	Read Host Watchdog Timer Value	Sec. 4.9
~AA3ETTTT	!AA	Enable Host Watchdog Timer	Sec. 4.9
~AA4P/~AA4S	!AAVV00	Read power-on/safe value	Sec. 4.10
~AA5P/~AA5S	!AA	Set power-on/safe value	Sec. 4.11
~AA7N	>(data)	Read monitoring data from channel N	Sec. 4.12
~AA8	>(data)	Read all monitoring data	Sec. 4.12
~AAE0NN	!(DATA)	Read EEPROM Data	Sec. 4.13
~AAE1NNDD	!AA	Write EEPROM Data	Sec. 4.13
~AAE2	!AA	EEPROM Write Protect	Sec. 4.14
~AAE3	!AA	EEPROM Write Enable	Sec. 4.14
~AAPNDD	!AA	Set PWM Duty Cycle	Sec. 4.15
~**	No Response	Host OK	Sec. 4.16

4.1 %AANNTTCCFF

- **Description:** Set the configuration of module.

- **Syntax:** %AANNTTCCFF[chk](cr)

% is a delimiter character

AA=2-character HEX module address, from 00 to FF

NN=new AA

TT=module type, 0x40 for WDT-03

CC=baud rate code,

CC	Baud rate
03	1200 BPS
04	2400 BPS
05	4800 BPS
06	9600 BPS
07	19200 BPS
08	38400 BPS
09	57600 BPS
0A	115200 BPS

FF=checksum status, Bit_6=1=enable, Bit_6=0=disable

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Response:** valid command → !AA[chk](cr)
 invalid command → ?AA[chk](cr)
 no response → syntax error or communication error or address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: %0102400600(cr)

response : !02(cr)

- | |
|---|
| <ul style="list-style-type: none"> ● changed to a new address 02 ● bard rate=9600 BPS ● checksum=disable |
|---|

command: %0001400640(cr)

response : !01(cr)

- | |
|---|
| <ul style="list-style-type: none"> ● Connect J5 of WDT-03 to initial state first. ● Enable checksum |
|---|

4.2 \$AA2

- **Description:** Read the configuration of module.
- **Syntax:** \$AA2[chk](cr)
\$ is a delimiter character
AA=2-character HEX module address, from 00 to FF
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Response:** valid command → !AATTCCFF[chk](cr),
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or
address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

TT, CC, FF: refer to Sec. 4.1

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: \$012(cr)	address 01, 9600 BPS, checksum disable
response : !01400600(cr)	

command: \$022(cr)	address 02, 19200 BPS, checksum disable
response : !02400700(cr)	

NOTE: If user use %AANNTTCCFF command to change configuration of WDT-03, the new configuration code will be stored into EEPROM immediately. The configuration code includes module address, module type, baud rate code, checksum enable/disable code, calibration code, power-on value and safe value. **The EEPROM data of WDT-03 can be read infinite times and can be written about 100,000 times max.** Therefore the user should not change configuration code often for testing.

The \$AA2 command is used to read EEPROM data only, therefore the user

can send this command to WDT-03 module infinitely.

4.3 \$AA5

- **Description:** Reset status read back. This is the only command to detect the module hardware watchdog failure. If the module is down, the module hardware watchdog circuit will reset this module. This reset will cause all D/O of WDT-03 going to their power-on values. These power-on values may be different from those output-values before WDT-03 reset. So the local host-PC has to send output command again to WDT-03 for keeping the same output states before and after reset.

- **Syntax:** \$AA5[chk](cr)

\$ is a delimiter character

AA=2-character HEX module address, from 00 to FF

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Response:** valid command → !(data)[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

S= 0, it has not been reset since last reset status read

1, it has been reset since last reset status read

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: \$015(cr)

response : !011(cr)

It is first time power-on

command: \$015(cr)

response : !010(cr)

It is normal

command: \$015(cr)

response : !010(cr)

It is normal

command: \$015(cr)

response : !011(cr)

This WDT-03 is reset by hardware watchdog. Therefore all D/O of WDT-03 are going to their power-on values now.
Refer to Sec. 5.4 for more information

4.4 \$AA6

- **Description:** Read digital input data and digital out data read back. The module will read the current state of digital input/output channel and return to host computer.

- **Syntax:** \$AA6[chk](cr)
 \$ is a delimiter character
 AA=2-character HEX module address, from 00 to FF
 [chk]=2-character checksum, if checksum disable → no [chk]
 (cr)=0x0D

- **Response:** valid command → !(data)[chk](cr)
 invalid command → ?AA[chk](cr)
 no response → syntax error or communication error or address error

! is a delimiter character indicating a valid command
 ? is a delimiter character indicating a invalid command
 AA=2-character HEX module address

Data=6-character HEX data value as follows:

Byte	5	4	3	2	1	0
data	Digital Output		Digital Input		0	0

[chk]=2-character checksum, if checksum disable → no [chk]
 (cr)=0x0D

- **Example:**

command: \$016(cr)
 response : !050700(cr)

Address-01
 DO=05, channel_0&2 = ON
 DI=07, all 3 channels = ON

command: \$026(cr)
 response : !020000(cr)

Address-02
 DO=02, channel_1 = ON
 DI=00, all 3 channels = OFF

4.5 #AABBDD

- **Description:** Set the value of digital output

- **Syntax:** #AABBDD[chk](cr)

is a delimiter character

AA=2-character HEX module address, from 00 to FF

The value of BB is given as follows:

BB	
00	Set all channels
1?	Set single channel, ?=channel number

DD=2-character HEX output data, from 00 to 07

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Response:** valid command → >[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: #010007(cr)

Turn all DO ON .

response : >(cr)

command: #011201(cr)

Turn DO channel_2 ON.

response : >(cr)

command: #011200(cr)

Turn DO channel_2 OFF.

response : >(cr)

4.6 \$AAF & \$AAM

- **Description:** \$AAF → Read the version number of WDT-03 firmware.
\$AAM → Read the module name of WDT-03
- **Syntax:** \$AAF[chk](cr) or \$AAM[chk](cr)
\$ is a delimiter character
AA=2-character HEX module address, from 00 to FF
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Response:** valid command → !AA(data)[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or
address error
! is a delimiter character indicating a valid command
? is a delimiter character indicating a invalid command
AA=2-character HEX module address
data=4-character for firmware version number
=6-character for module name
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D

- **Example:**

command: \$01F(cr) response : !01A1.6(cr)	<div style="border: 1px solid black; padding: 5px;">module 01 version A1.6</div>
command: \$02F(cr) response : !02A3.0(cr)	<div style="border: 1px solid black; padding: 5px;">module 02 version A3.0</div>
command: \$01M(cr) response : !01WDT-03(cr)	<div style="border: 1px solid black; padding: 5px;">Module name of 01 is WDT-03</div>

4.7 ~AA0

- **Description:** Read the system status. The system status will be latch until ~AA1 command is received. If the system status=0x04 → local host-PC is failure now → all output command received will be ignored.
- **Syntax:** ~AA0[chk](cr)
~ is a delimiter character
AA=2-character HEX module address, from 00 to FF
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Response:** valid command → !AASS[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

SS=2-character HEX status value as following:

Bit_0/1 = reserved

Bit_2 = 0 → OK

1 → host watchdog active → local host-PC is failure now

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: ~010(cr)

response : !0100(cr)

System status is OK

command: ~010(cr)

response : !0104(cr)

command: #0100FF(cr)

response : !(cr)

command: ~011(cr)

response : !01(cr)

command: #0100FF(cr)

response : >(cr)

System status =0x04 → host watchdog active → local host-PC is failure now → D/O command is ignored → Send ~AA1 to clear status → D/O command is accepted now

4.8 ~AA1

- **Description:** The system status will be latch until ~AA1 command is received. If the system status=0x04 → local host-PC is failure now → all output command received will be ignored. Therefore user should read the system status first to make sure that the system status is 0. If the system status is not 0, only ~AA1 command can clear the system status

- **Syntax:** ~AA1[chk](cr)

~ is a delimiter character

AA=2-character HEX module address, from 00 to FF

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Response:** valid command → !AA[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: ~010(cr)

response : !0104(cr)

system status=0x04 → local host-PC is failure now

command: #0100FF(cr)

response : !(cr)

Output command is ignored

command: ~011(cr)

response : !01(cr)

Clear module status

command: ~010(cr)

response : !0100(cr)

Module status = 0x00

command: #0100FF(cr)

response : >(cr)

Output command is accepted

4.9 ~AA2 & ~AA3ETTTT

- **Description:** Read the status and timer value of host watchdog. The host watchdog timer is designed for host watchdog. When the host watchdog is enable, the host must send ~** command to WDT-03 before the timer is up. When the ~** command is received, the host watchdog timer is reset and restart. Use ~AA3ETTTT to enable/disable/setting the host watchdog timer.

- **Syntax:** ~AA2[chk](cr) → read watchdog timer status
~AA3ETTTT[chk](cr) → set watchdog timer status

~ is a delimiter character

AA=2-character HEX module address, from 00 to FF

E=0 is disable and 1 is enable

TTTT=4-character HEX value, from 00 to FFFF, unit=0.03 second

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Response:** valid command → !AASTTTT[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

S=0: host watchdog is disable

S=1: host watchdog is enable

TTTT=4-character HEX value, from 00 to FFFF, unit=0.03 second

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: ~012(cr)

response : !0100000(cr)

Host watchdog timer of module 01 is disable

command: ~022(cr)

response : !021000A(cr)

host watchdog timer of module 02 is enable and =0.03*10 =0.3 second.
--

command: ~0131000A(cr)

response : !01(cr)

host watchdog timer of module 01 is enable and equal to 0.03*10 =0.3 second.
--

4.10 ~AA4P & ~AA4S

● **Description:** Read power-on / safe value of D/O.

1. When WDT-03 is first power-on, all output channels will go to their power-on values.
2. When WDT-03 is failure, the hardware watchdog will reset the WDT-03 and all output channels will also go to their power-on values. These power-on values may be different to old values before reset. Therefore the host-PC must send out a new command to control all output to the desire states.
3. When the host watchdog is enable and the host-PC is failure, all output will go to their safe values and the system status will be changed to 0x04.
4. If the system status is 0x04, all output command will be ignored before the system status is clear by ~AA1 command. Therefore the user must send ~AA1 command first, then send out a new output command to control all output to the desire states.

● **Syntax:** ~AA4P[chk](cr) → read power-on value

~AA4S[chk](cr) → read safe value

~ is a delimiter character

AA=2-character HEX module address, from 00 to FF

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

● **Response:** valid command → !AAVV00[chk](cr)

invalid command → ?AA[chk](cr)

no response → syntax error or communication error or
address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

VV=2-character HEX module address

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

● **Example:**

command: ~014P(cr)

Power-on value is 0x00

response : !010000(cr)

command: ~024S(cr)

Safe value is 0x07

response : !020700(cr)

4.11 ~AA5P & ~AA5S

- **Description:** Set current state of digital output as power-on/safe value. Refer to Sec.4.10 for more information.

- **Syntax:** ~AA5P[chk](cr) → set power-on value
 ~AA5S[chk](cr) → set safe value

~ is a delimiter character

AA=2-character HEX module address, from 00 to FF

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Response:** valid command → !AA[chk](cr)
 invalid command → ?AA[chk](cr)
 no response → syntax error or communication error or
 address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: #010000(cr)

response : >(cr)

Power-on value is 0x00

command: ~015P(cr)

response : !01(cr)

command: #02000F(cr)

response : >(cr)

Safe value is 0x0F

command: ~025S(cr)

response : !02(cr)

4.12 ~AA7N & ~AA8

- **Description:** Read the monitoring data.
- **Syntax:** ~AA7N[chk](cr) → Read channel-N
 ~AA8[chk](cr) → Read all channels
~ is a delimiter character
AA=2-character HEX module address, from 00 to FF
N =1-character HEX channel number, from 0 to 0x0D
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Response:** valid command → !(data)[chk](cr)
 invalid command → ?AA[chk](cr)
 no response → syntax error or communication error or
 address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

data=2-character HEX value, from 00 to FF, 2's

complement data format

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: ~0170(cr)

response : !D1(cr)

command: ~0171(cr)

response : !BB(cr)

command: ~0172(cr)

response : !AA(cr)

command: ~0173(cr)

response : !60(cr)

channel_00 = +3.3V
channel_01 = +5V
channel_02 = +12V
channel_03 = -12V
channel_04 = -5V
channel_05 = T0
channel_06 = T1

channel_07 = T2
channel_08 = FAN 0
channel_09 = FAN 1
channel_0A = FAN 2
channel_0B = PWM 0
channel_0C = PWM 1
channel_0D = PWM 2

command: ~018(cr)

response : !01D1.BB.AA.60.E0.24.1F.1C.FF.FF.84.FF.FF.FF(cr)

Refer to Sec. 5.5 & Sec. 1.10.6 for more information.

4.13 ~AAE0NN & ~AAE1NNDD

- **Description:** ~AAE0NN → Read from EEPROM
~AAE1NNDD → Write to EEPROM
- **Syntax:** ~AAE0NN[chk](cr) or ~AAE1NNDD[chk](cr)
~ is a delimiter character
AA=2-character HEX module address, from 00 to FF
NN=2-character HEX eeprom address, from 00 to 19H
DD=2-character HEX PWM value, from 00 to FF
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Response:** valid command → !(data)[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or
address error
! is a delimiter character indicating a valid command
? is a delimiter character indicating a invalid command
AA=2-character HEX module address
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Example:**

command: ~01E019(cr)
response : !78(cr)

Read the contents in EEPROM at address 0x19=0x78 .

command: ~01E10067(cr)
response : !01(cr)

Write 0x67 to EEPROM address 0x00 .
--

NOTE: The EEPROM is divided into two area, **system area & user area**. All EEPROM commands are active on **user area** only. The **system area** is

reserved by system & can not be read/write.

4.14 ~AAE2 & ~AAE3

- **Description:** ~AAE2 → Enable write data to EEPROM.
~AAE3 → Disable write data to EEPROM.
- **Syntax:** ~AAE2[chk](cr) or ~AAE3[chk](cr)
~ is a delimiter character
AA=2-character HEX module address, from 00 to FF
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Response:** valid command → !AA[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or address error
! is a delimiter character indicating a valid command
? is a delimiter character indicating a invalid command
AA=2-character HEX module address
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Example:**

command: ~01E2(cr)	Enable write to EEPROM.
response : !01(cr)	

command: ~01E3(cr)	Disable write to EEPROM.
response : !01(cr)	

NOTE 1: The EEPROM is divided into two area, **system area & user area**. All EEPROM commands are active on **user area** only. The **system area** is reserved by system & can not be read/write.

2: the EEPROM is disable for write operation since WDT-03 is first power on. The user has to use ~AAE2 command before any write to EEPROM operation.

4.15 ~AAPNDD

- **Description:** Set PWM duty cycle of FAN. Refer to Sec. 1.10.7 for more information.

- **Syntax:** ~AAPNDD[chk](cr)

~ is a delimiter character

AA=2-character HEX module address, from 00 to FF

N=PWM channel, from 0 to 2.

DD=PWM duty cycle, FF=100%

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Response:** valid command → !AA[chk](cr)
invalid command → ?AA[chk](cr)
no response → syntax error or communication error or address error

! is a delimiter character indicating a valid command

? is a delimiter character indicating a invalid command

AA=2-character HEX module address

[chk]=2-character checksum, if checksum disable → no [chk]

(cr)=0x0D

- **Example:**

command: ~01P119(cr)

response: !01(cr)

PWM1 of module 01 is enable and equal to 0x19/0xff = 10 % .
--

4.16 ~**

- **Description:** Host send this command to tell WDT-03 “Host is OK”. Refer to Sec. 5.3 for more information.
- **Syntax:** ~**[chk](cr)
~ is a delimiter character
[chk]=2-character checksum, if checksum disable → no [chk]
(cr)=0x0D
- **Response:** no response
- **Example:**
command: ~**(cr)
response : No Response

5. Application Notes

5.1 Go to Initial Setting

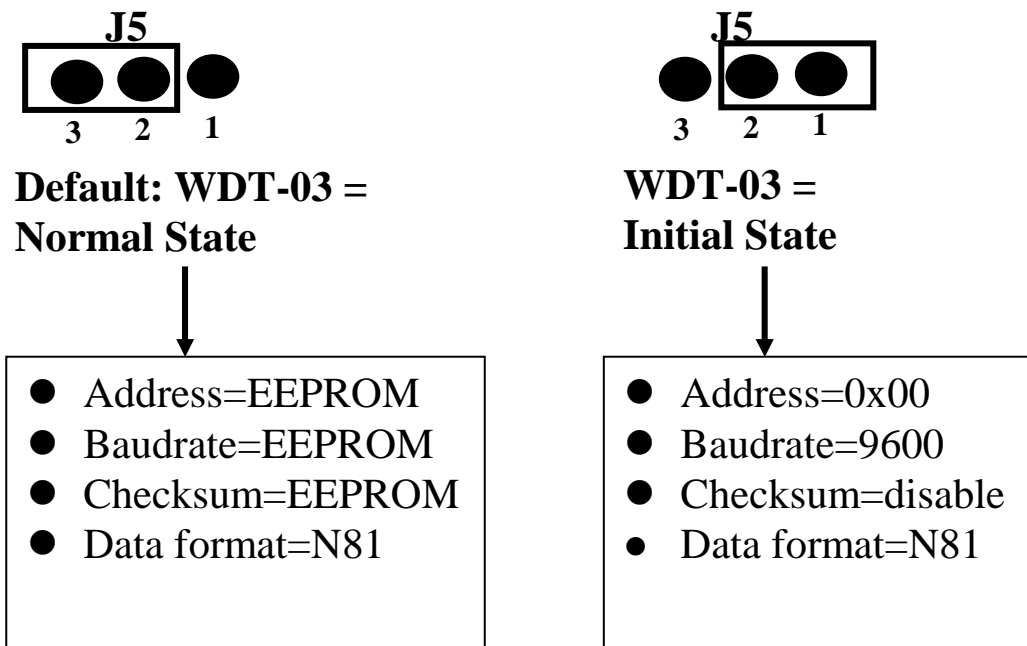
WDT-03 equips an EEPROM to store configuration information. Therefore the user is difficult to find out the status of WDT-03. If install one jumper in the pin-1 & pin-2 of J5 during the power-up stage, the WDT-03 will go to its **initial setting** without change the EEPROM data. The initial setting is given as following:

address = 00
baud rate = 9600
checksum = DISABLE
data format = 1 start + 8 data bits + 1 stop bit

Then user can ask WDT-03 what is its original setting by **\$AA2** command as following:

- Step 1 : power off and install jumper to pin-1 & pin-2 of J5
- Step 2 : power on
- Step 3 : send command string **\$002[0x0D]** to WDT-03
- Step 4 : WDT-03 will echo its original status stored in EEPROM
- Step 5 : power off and install jumper to pin-2 & pin-3 of J5
- Step 6 : power on, the WDT-03 will auto configured itself based on the EEPROM data

Refer to Sec. 1.10.4 for more information.



5.2 Dual WatchDog Operation Principle

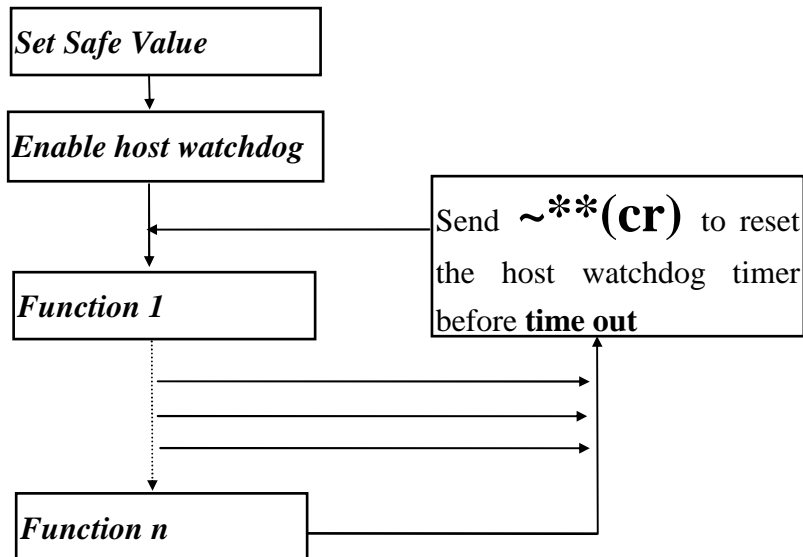
Dual watchdog = host watchdog + module watchdog
The host watchdog is a software watchdog.
The module watchdog is a hardware watchdog.

WDT-03 is designed for industry applications, therefore it can work in the harsh environment. There are many couple noise or energy transient in such environment. The WDT-03 may be failure if these noise is really too large. The module watchdog can reset WDT-03 if it is failure for too large signal. Sometimes even the host-PC may be failure for hardware or software problems. The host watchdog can monitor the status of host-PC. If the host-PC is failure, all the output of WDT-03 will go to their predefined safe states for safety protection.

If the RS-485 network is open, all the host command can not send to WDT-03. This is very dangerous in real world application. The output of WDT-03 will go to their predefined safe state for safety consideration if the host watchdog is enable. This dual watchdog mechanism will increase the system reliability very much.

5.3 Host WatchDog Applications Notes

The software host watchdog is designed to monitor the local host-PC. If the local host-PC is fail, the output of the WDT-03 will automatically go to their safe states to avoid unpredictable damaged. The flow chart for the local host-PC is given as following.



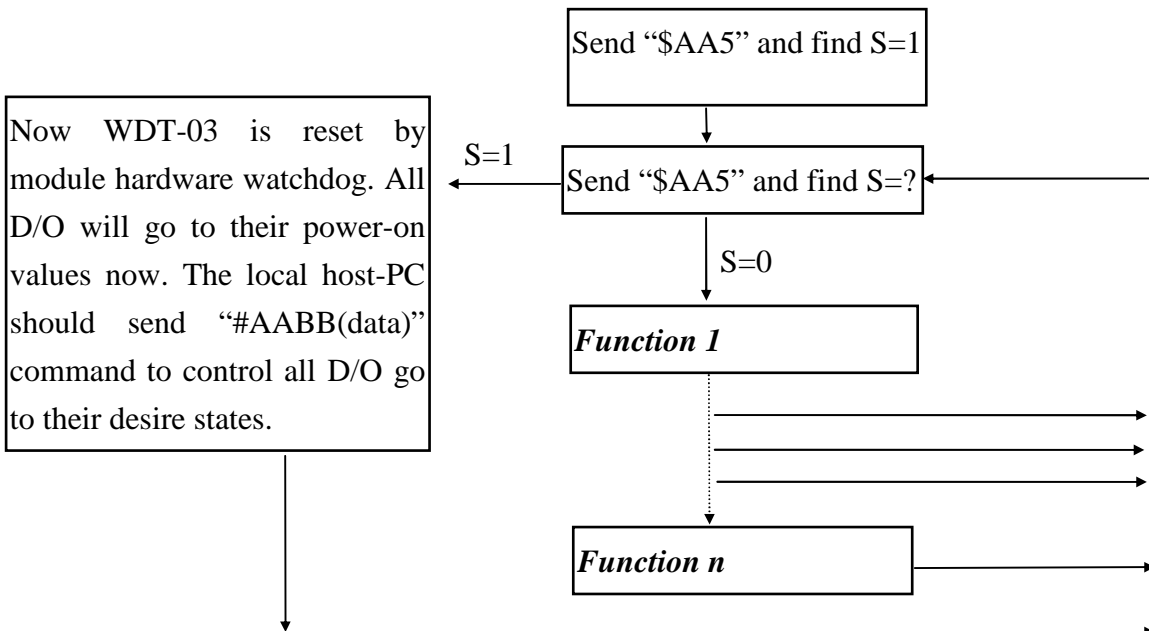
1. power on and run **test.exe**
2. press **2**
3. press **\$01M[Enter]** → Receive=!01WDT-03
4. press **2**
5. press **#01000F[Enter]** → Receive=>
6. press **2**
7. press **\$016[Enter]** → Receive=!0F0F00
8. press **2**
9. press **~014S[Enter]** → Receive=!0F0F00
10. press **2**
11. press **#010000[Enter]** → Receive=>
12. press **2**
13. press **~01310064[Enter]** → Receive=!01
14. wait 3 second, you will find that the led of WDT-03 is OFF and all D/O are turn ON now. This action simulates that the local host-PC is failed now, so all D/O go to their safe states.
15. press **2**
16. press **~010[Enter]** → Receive=!0104
17. press **2**

18. press **\$016[Enter]** → Receive=!010F0F00
19. press **2**
20. press **~013000[Enter]** → Receive=!01
21. press **2**
22. press **~011[Enter]** → Receive=!01
23. press **2**
24. press **#010000[Enter]** → Receive=>
 - step 3: It is WDT-03
 - step 5: set all the D/O to ON states
 - step 7: read back the state of all D/I/O: all D/O ON, all input HIGH
 - step 9: set current D/O states as safe-value
 - step 11: set all the D/O to OFF states
 - step 14: enable the host watchdog and timer = 64H*0.03s = 3 sec.
 - Step 16: find the host-watchdog is failure now
 - step 18: read back the D/I/O state, all relay ON, all input HIGH.
 - step 20: disable the host watchdog.
 - step 22: clear the system status to 00
 - step 24: set all D/O to OFF states again

5.4 Module WatchDog Applications Notes

The “\$AA5” command is designed to detect the module hardware watchdog failure. If WDT-03 is failed, the module hardware watchdog circuit will reset this WDT-03 immediately. After this reset, all output of WDT-03 will go to their power-on values. These power-on values may be different from those output-values before reset. Now the user has to send “#AABB(data)” command to WDT-03 for keeping the same output state before and after reset.

The flow chart for module hardware watchdog failure detection of local host-PC is given as following.



Note: When first power-on, the application program should use “\$AA5” command to find S=1, then use “\$AA5” again to check S=0. If S is changed to 1, it means that this WDT-03 is just been reseted by hardware watchdog. And all D/O will go to their power-on values now. Therefore the application program has to send “#AABB(data)” command again to control all D/O to new desired states

5.5 Analog Data Format

command: ~AA8(cr)

response : !AAD1.BB.AA.60.E0.24.1F.1C.FF.FF.84.FF.FF.FF(cr)

channel_00

channel_07

channel_0D

channel_00 = +3.3V
channel_01 = +5V
channel_02 = +12V
channel_03 = -12V
channel_04 = -5V
channel_05 = T0
channel_06 = T1

channel_07 = T2
channel_08 = FAN 0
channel_09 = FAN 1
channel_0A = FAN 2
channel_0B = PWM 0
channel_0C = PWM 1
channel_0D = PWM 2

All values
are in HEX
format

- $3.3V \rightarrow (0.0161 * \text{channel_00})$ Volt
- $+5V \rightarrow (0.0268 * \text{channel_01})$ Volt
- $+12V \rightarrow (0.0720 * \text{channel_02})$ Volt
- $-12V \rightarrow (0.1199 * \text{channel_03} - 23.266)$ Volt
- $-5V \rightarrow (0.0466 * \text{channel_04} - 6.853)$ Volt
- T1 \rightarrow channel_05 °C
- T2 \rightarrow channel_06 °C
- T3 \rightarrow channel_07 °C
- FAN1 $\rightarrow (675000 / \text{channel_08})$ RPM
- FAN2 $\rightarrow (675000 / \text{channel_09})$ RPM
- FAN3 $\rightarrow (675000 / \text{channel_0A})$ RPM
- PWM1 \rightarrow channel_0B
- PWM2 \rightarrow channel_0C
- PWM3 \rightarrow channel_0D

Refer to Sec. 1.10.6 for more information

5.6 D/O Operation Principle

- (1) The D/O output of WDT-03 will go to their power-on values when the power is first applied.
- (2) The D/O output will change to desire states if the “#AABB(data)” command is received. Then all these D/O will keep in the same states until next “#AABB(data)” command.
- (3) If WDT-03 is reseted by hardware watchdog, all D/O will go to their power-on values immediately. These power-on values may be different from the original values before reset. Now the local host-PC can use “\$AA5” command to detect this condition & use “#AABB(data)” command to set all D/O to their new desire states.
- (4) If the host watchdog is failure, all D/O will go to their safe-values immediately and the system status is set to 04. If the local host-PC send “#AABB(data)” command to WDT-03 now, the WDT-03 will ignore this command and return “!” as warning information. The local host-PC must use “~AA1” command to clear the system status to 00, then the WDT-03 will accept the “#AABB(data)” command again.

5.7 Command Response Time

The command response time of WDT-03 is given as following:

1. Local Host-PC send command
2. WDT-03 receive this command nearly at the same time
3. WDT-03 **wait a character time**
4. WDT-03 get the current value and send this value back to host-PC

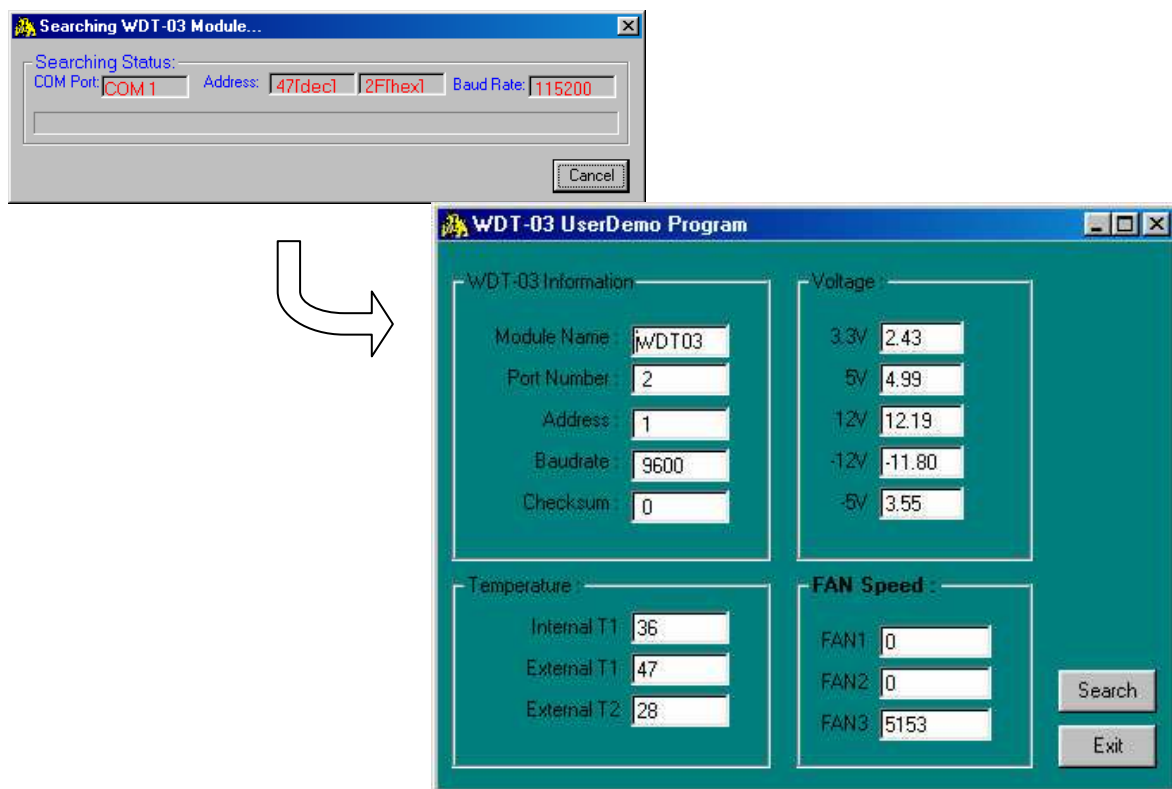
Assume the communication condition is given as following:

- baud rate=115.2K
- command= \$AA5(cr) → 5 character
- wait 1 character
- response= !AAS(cr) → 5 character
- total character= 5+1+5=11 character
- 1 character= 10 bits → $115.2K/10=11.52K$
- 11 character → $11.52K/11=1.0K$ max.
→ 1000 command/response per second max.

5.8 Userdemo.exe Operation Principle

The userdemo.exe is used as a demo of user's application program. Its operation principles are given as follows:

- When program is first executed, it will search for one WDT-03 as follows:
 1. COM port: search from COM 1 to COM 255
 2. Communication speed: search from 2400 to 115200 BPS
 3. Checksum: search for disable or enable
- If the WDT-03 is absent, all data fields will be empty. User should click the “**Exit**” field to stop this program.
- Does not enable/disable the watchdog & dual-watchdog of WDT-03
- Show all monitoring status provided by WDT-03 in screen
- Stop this program if user click the “**STOP**” field.



Because the userdemo.exe does not enable/disable the watchdog & dual watchdog of WDT-03, it can be used to simulate user's original application program without modification.

All source codes of userdemo.exe are given for reference.

5.9 DB-3R Applications Notes

The DB-3R is a 3-channel of digital input terminal and 3-channel of relay output daughter board for WDT-03.

The DB-3R features as follows

- 3-channel of digital input terminal
- 3-channel of relay output
- Connects directly to WDT-03
(Only the versions that come out after version 5.2, including version 5.2 support DB-3R)
- The Relay output of specifications: Channels : 3 (Form "A", Normal Open)
Max. Switching voltage : 220Vdc, 250Vac Max. Switching current : 2A
Max. Carrying current : 2A Surge Strength : 1,500V Electrical endurance : Typical 500,000 operation at 30Vdc / 1A Typical 100,000 operation at 125Vac / 0.5A Relay on time (typical) : 6ms Relay off time (typical) : 6ms
- Pin to Pin screw terminal for D-Sub (From the CN2 of WDT-03)
- Pitch : 5.08 m/m
- Dimension : 103mm x 86mm

