ICP
DAS

DCON_FUN

User's Manual

(Version 1.0)

Dynamic Link Library (DLL) for DCON
(I-7000/8000/87K) Series Modules

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

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.

Table of Contents

1. Introduction

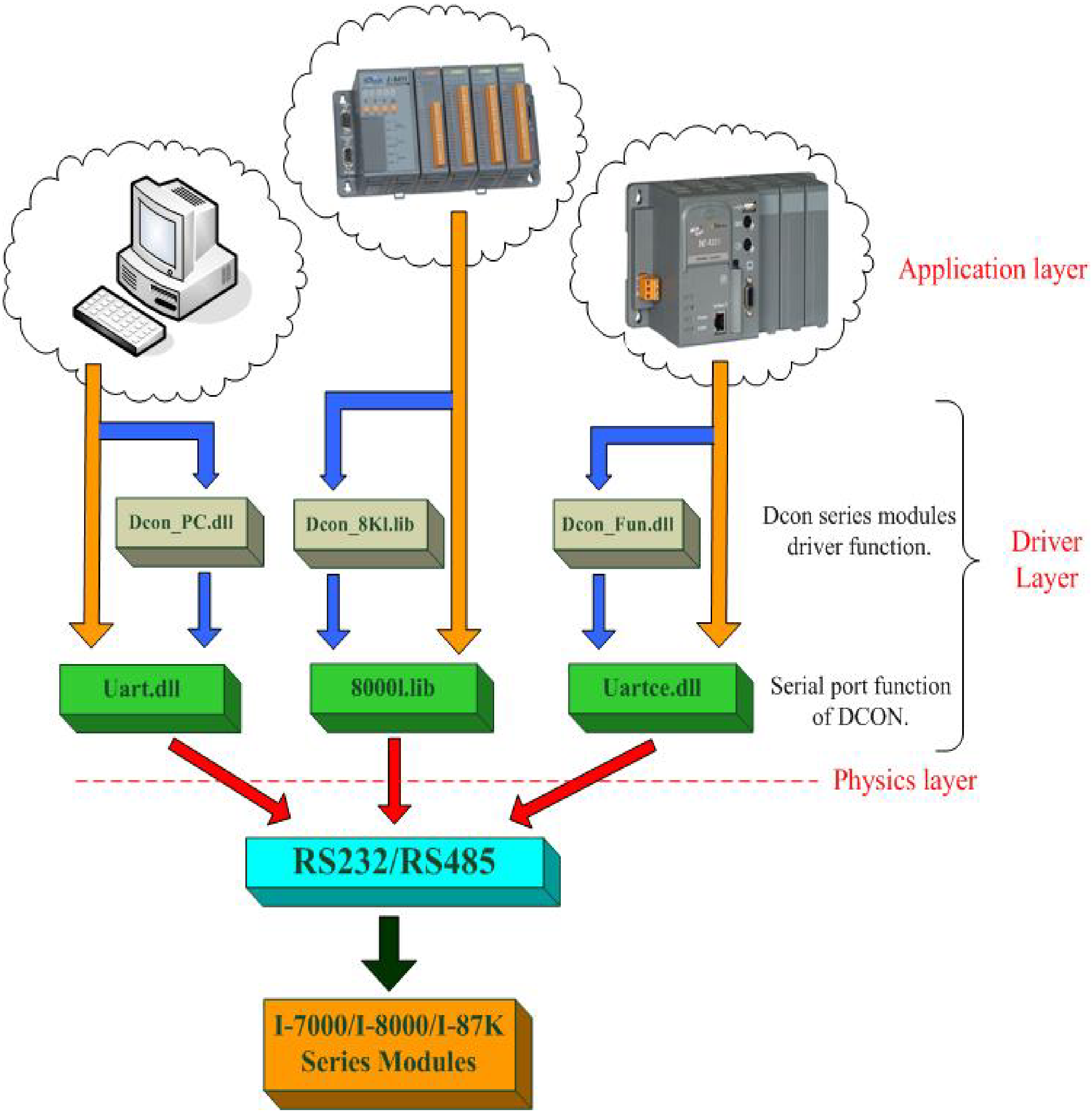
The general Dcon function is a LIB or DLL functions designed for Minios7, Windows 95/98/2000/XP, or Wince and can use the same function interface on those platform.

1.1. Feature

	I7000.DLL	DCON_FUN
Support protocol	DCON protocol	DCON protocol
Method of communication	RS-232, RS-485	RS-232, RS-485
Support Module	DCON series modules	DCON series modules
Need Module ID?	Yes	No
Function Number	100~	20~
Support OS	Windows, Wince	Minios7, Windows, Wince
Support Demo programs	VB5, VC5, Delphi, BCB3	VB6.0, EVC4.0, BC3.1

1.2. Architectures under every OS platform

	OS	Hardware	Link name	Develop Environment
Platform	MiniOS7	I-8000	Dcon_8Kl.lib 8000l.lib	BC3.01
	Windows	PC	DCON_PC.dll Uart.dll	VC++6.0
	Wince	Wincon	DCON_CE.dll Uartce.dll	EVC4.0



2. Dcon_Fun APPLICATION

2.1. USING C language compiler for I-8000

The demo programs are tested OK under MiniOS7 and C language compiler .

Those usable compilers are shown as follows:

- TC 2.01
- TC++ 1.01
- BC++ 3.1
- MSC 6.0
- MSVC++ (before Version 1.52)

From Borland website, use can download the free TC2.01 and TC++1.01 compilers.

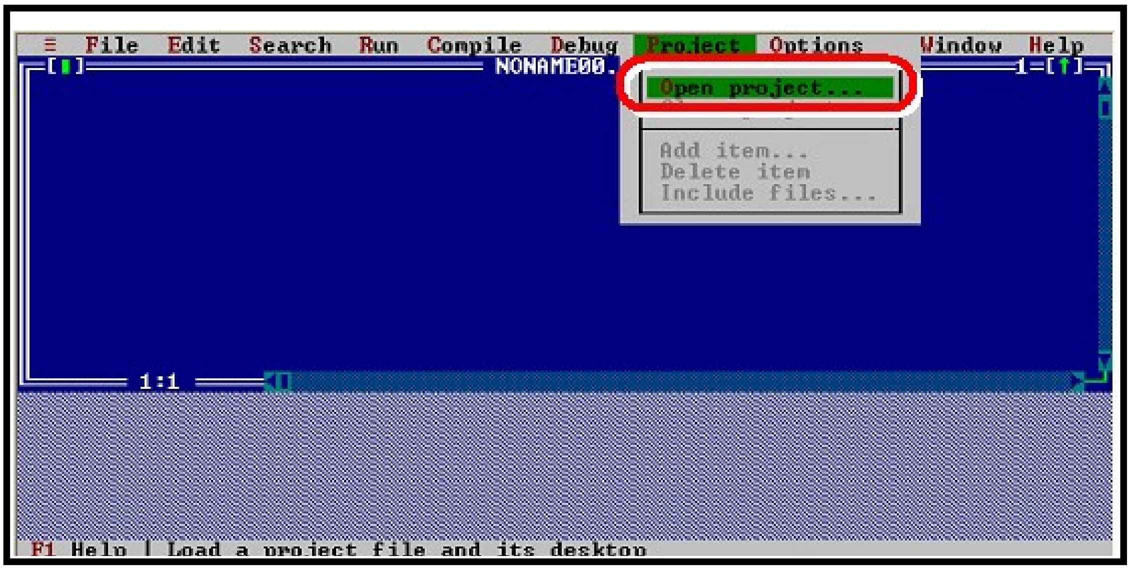
Website: http://community.borland.com/museum

The user of I-8000 series has to use these file to develop program as following:

- \Lib\8000e.lib → function to deal with RS-232 or RS-485
- \Lib\8000.h
- \Lib\Dcon_8K.lib → function for A/D, D/A, D/I, D/O, Counter
- \Lib\Dcon_fun.h

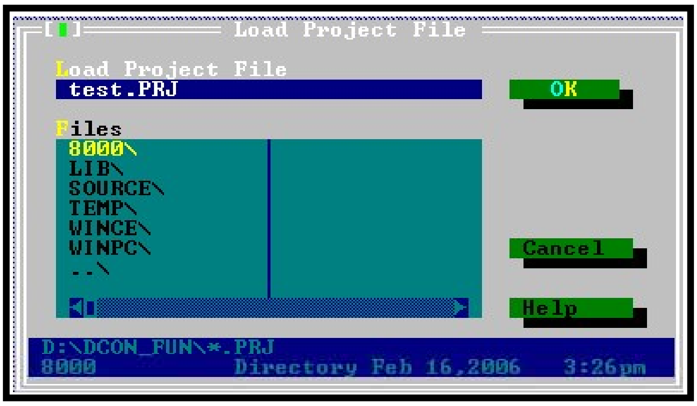
The key points for how to develop these demo programs are given as following:

Step 1: Create a new project file (*.prj).



F1 Help | Load a project file and its desktop

Step 2: Type the name of the project file and then click the **OK** button.

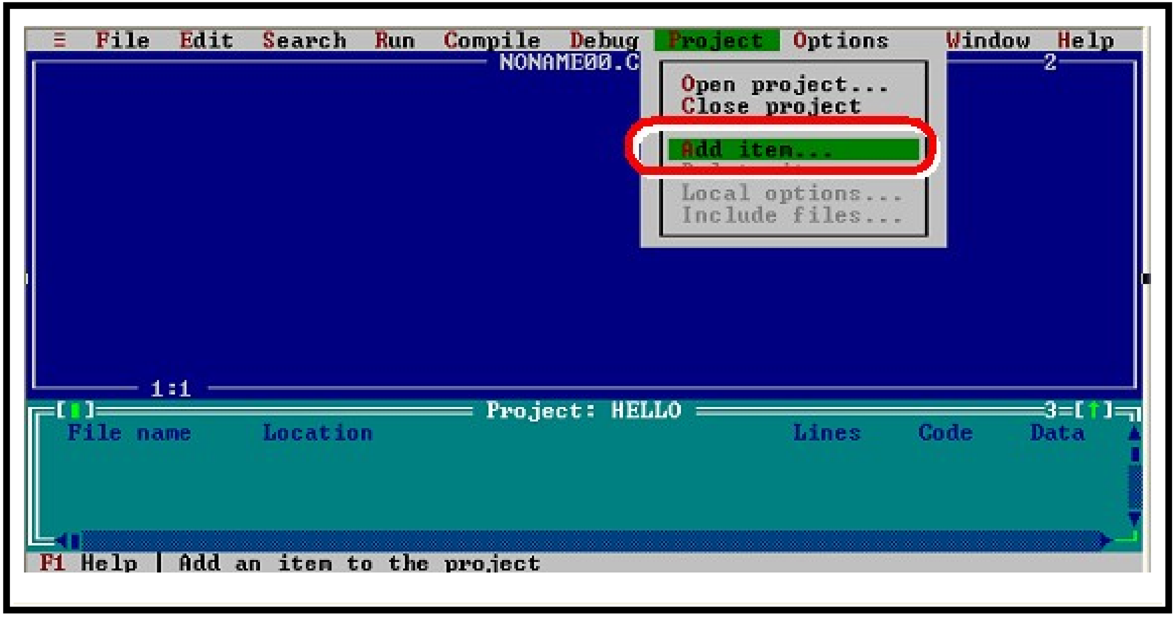


Load Project File
test.PRJ

Files
8000\
LIB\
SOURCE\
TEMP\
WINCE\
WINPC\
..\

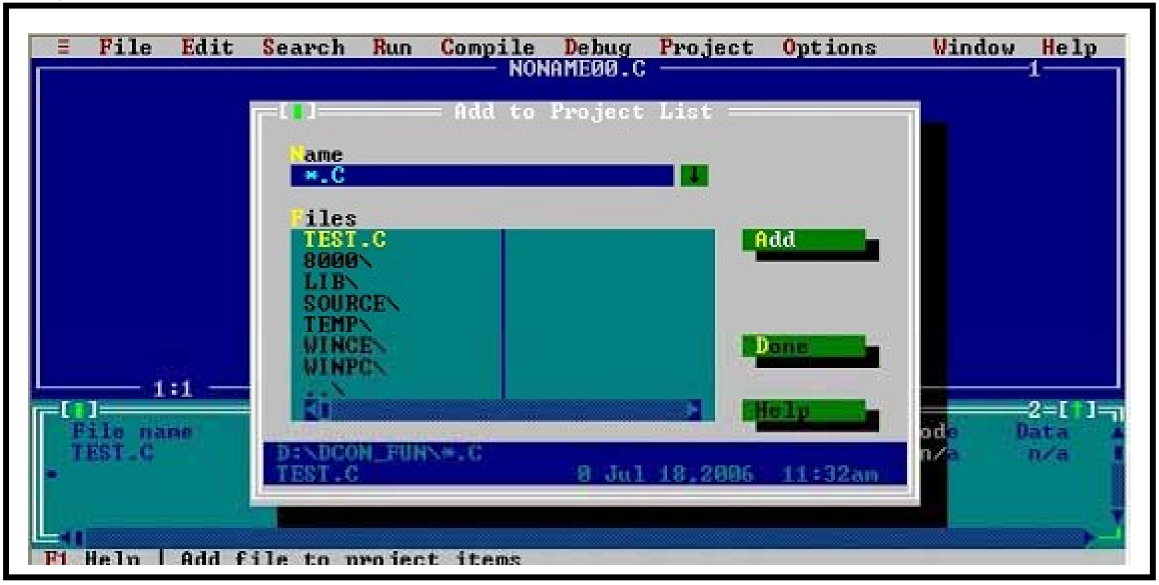
D:\DCON_FUN\*.PRJ
8000 Directory Feb 16,2006 3:26pm

Step 3: Add all necessary files to the project.



F1 Help | Add an item to the project

Step 4: Select the source file and then click the **Add** button.


≡ File Edit Search Run Compile Debug Project Options Window Help
NONAME00.C
1
Add to Project List
Add
Done
Help
1:1
2
File name
Data

Name
*.C

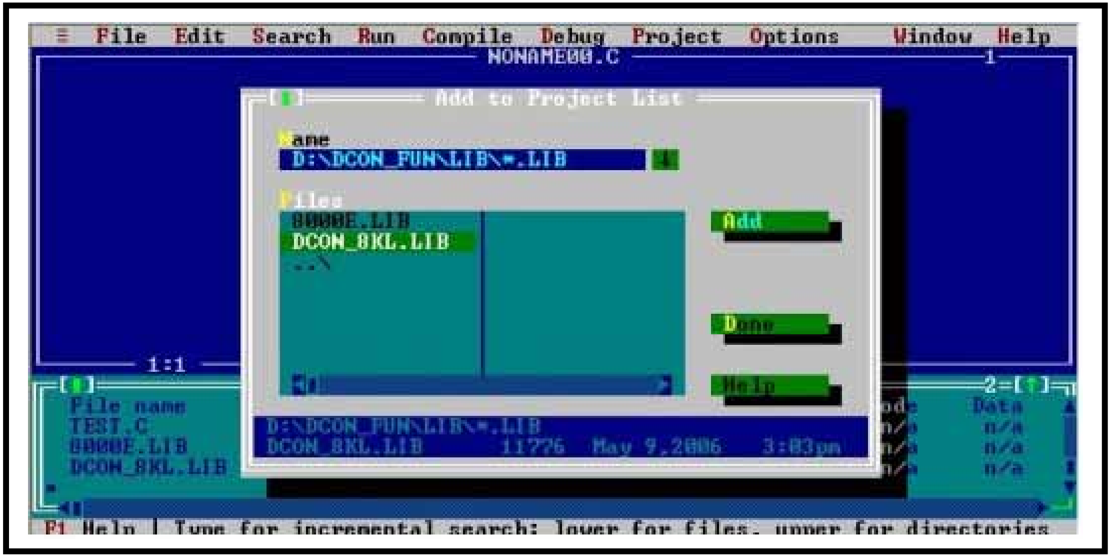
Files
TEST.C
8000\
LIB\
SOURCE\
TEMP\
WINCE\
WINPC\
..\

D:\DCON_FUN\*.C
TEST.C 0 Jul 18,2006 11:32am

File name
TEST.C

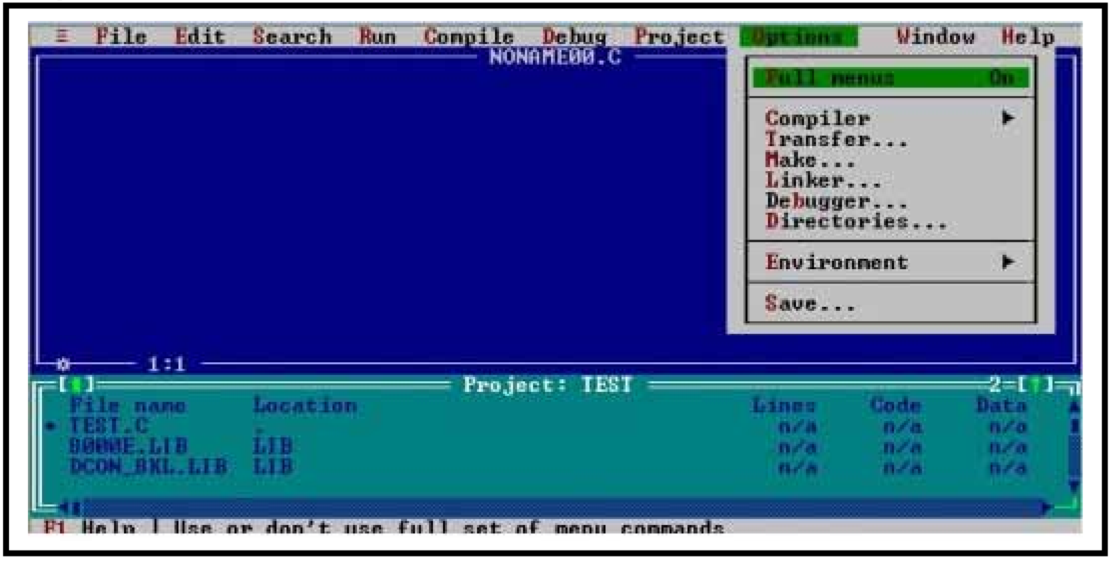
F1 Help | Add file to project items

Step 5: Select the function library and then click the **Add** button.



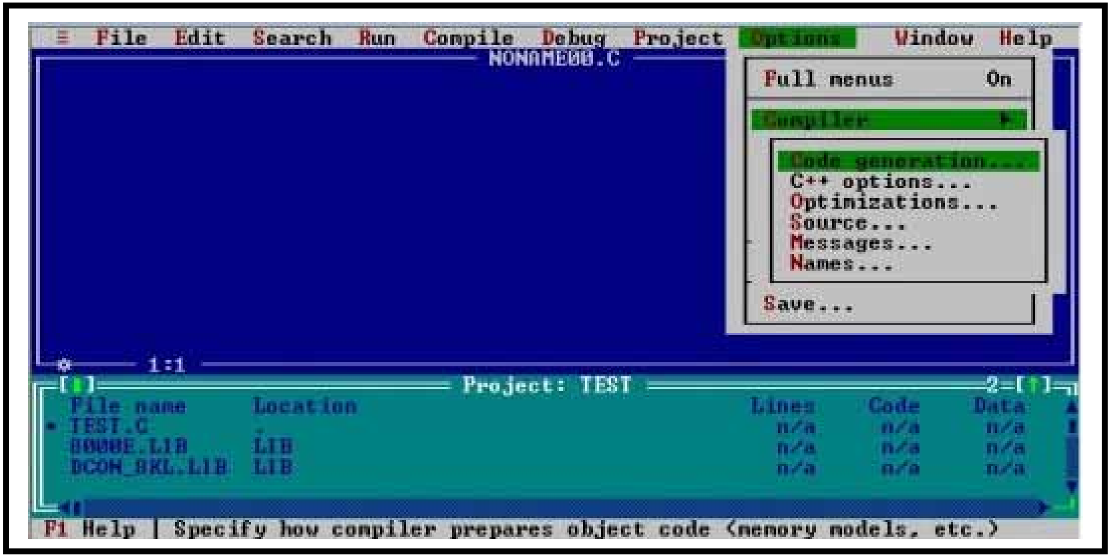
```
≡  File  Edit  Search  Run  Compile  Debug  Project  Options   Window  Help
                         NONAME00.C                                    1
        Add to Project List
Name
D:\DCON_FUN\LIB\*.LIB
Files
8000E.LIB
DCON_8KL.LIB                              Add
..\
                                          Done
                                          Help
D:\DCON_FUN\LIB\*.LIB
DCON_8KL.LIB          11776   May 9,2006    3:03pm
     1:1
File name                                              Data
TEST.C                                                 n/a
8000E.LIB                                              n/a
DCON_8KL.LIB                                           n/a
F1 Help | Type for incremental search: lower for files, upper for directories
```

Step 6: Show full menus.



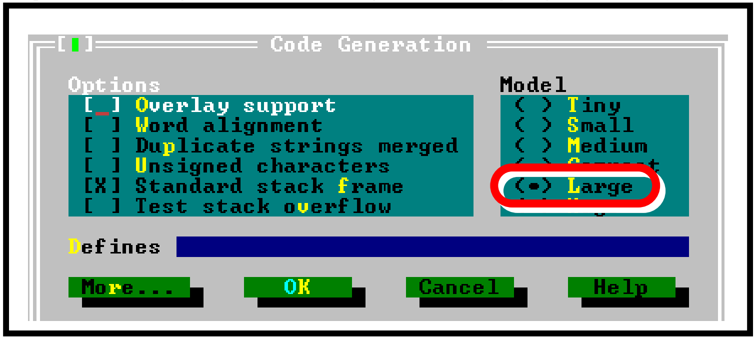
```
≡  File  Edit  Search  Run  Compile  Debug  Project  Options   Window  Help
                         NONAME00.C
                                              Full menus      On
                                              Compiler        ►
                                              Transfer...
                                              Make...
                                              Linker...
                                              Debugger...
                                              Directories...
                                              Environment     ►
                                              Save...
     1:1
                      Project: TEST                              2=[↑]
  File name      Location                    Lines   Code   Data
• TEST.C         .                           n/a     n/a    n/a
  8000E.LIB      LIB                         n/a     n/a    n/a
  DCON_8KL.LIB   LIB                         n/a     n/a    n/a
F1 Help | Use or don't use full set of menu commands
```

Step 7: Set Code generation options.



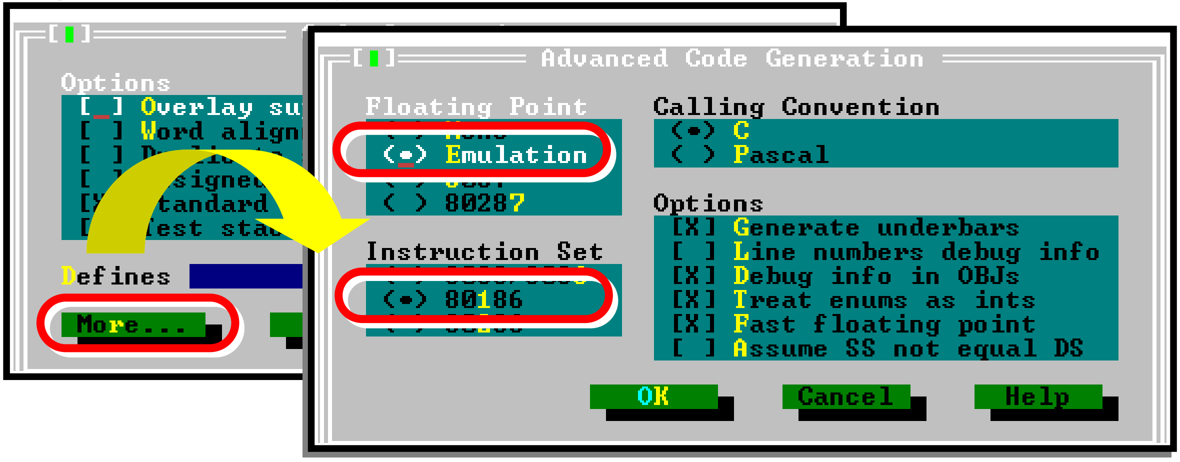
```
Project: TEST
File name     Location        Lines   Code   Data
TEST.C        .               n/a     n/a    n/a
8000E.LIB     LIB             n/a     n/a    n/a
DCON_8KL.LIB  LIB             n/a     n/a    n/a
F1 Help | Specify how compiler prepares object code (memory models, etc.)
```

Step 8: Change the Memory model (large for 8000E.lib).



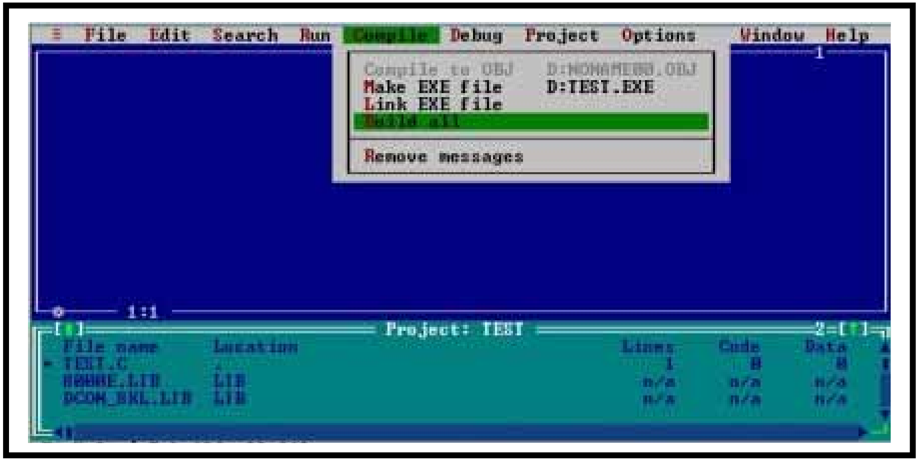
```
Options                               Model
[ ] Overlay support                   ( ) Tiny
[ ] Word alignment                    ( ) Small
[ ] Duplicate strings merged          ( ) Medium
[ ] Unsigned characters               ( ) Compact
[X] Standard stack frame              (•) Large
[ ] Test stack overflow               ( ) Huge
Defines
```

Step 9: Set the Floating Point to **Emulation** and the Instruction Set to **80186**.



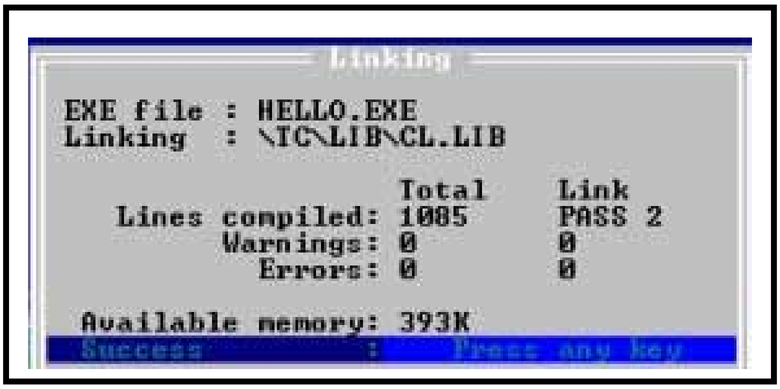
```
Floating Point              Calling Convention
( ) None                    (•) C
(•) Emulation               ( ) Pascal
( ) 8087
( ) 80287                   Options
                            [X] Generate underbars
Instruction Set             [ ] Line numbers debug info
( ) 8088/8086               [X] Debug info in OBJs
(•) 80186                   [X] Treat enums as ints
( ) 80286                   [X] Fast floating point
                            [ ] Assume SS not equal DS
```

Step 10: Make the project.



≡ File Edit Search Run Compile Debug Project Options Window Help
Compile to OBJ D:NONAME00.OBJ
Make EXE file D:TEST.EXE
Link EXE file
Build all
Remove messages
1:1

```
Project: TEST
File name      Location                    Lines   Code   Data
TEST.C         .                               1      0      0
8000E.LIB      LIB                           n/a    n/a    n/a
DCON_BKL.LIB   LIB                           n/a    n/a    n/a
```



```
                  Linking

EXE file  : HELLO.EXE
Linking   : \TC\LIB\CL.LIB

                    Total     Link
  Lines compiled:   1085      PASS 2
        Warnings:   0         0
          Errors:   0         0

 Available memory:  393K
 Success         :  Press any key
```

If user need the detail information, please refer to the following web site:

ftp://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/minios7/document/quickstart_develop_program.pdf

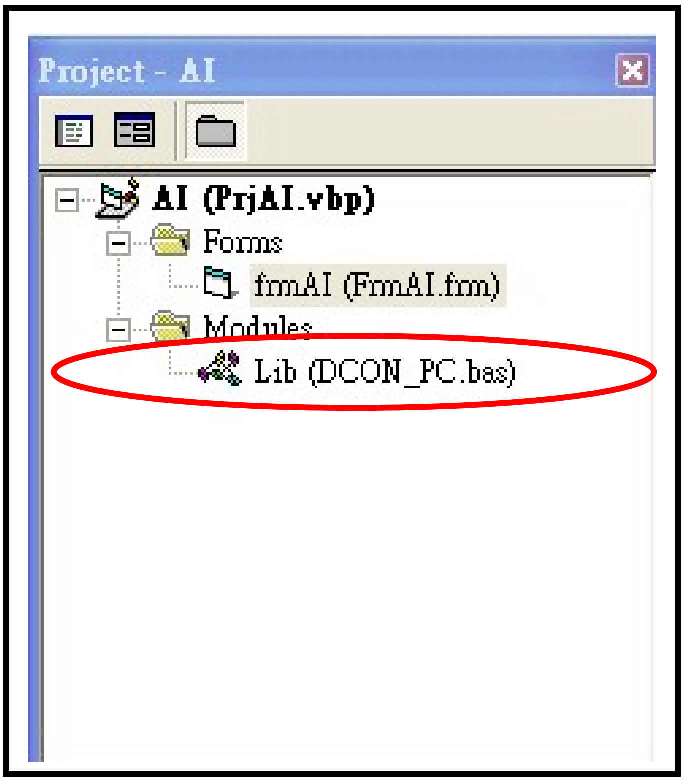
2.2. USING VISUAL BASIC for PC

The demo programs are tested OK in Windows 95/98/NT/2000/XP and VB6.0 version.

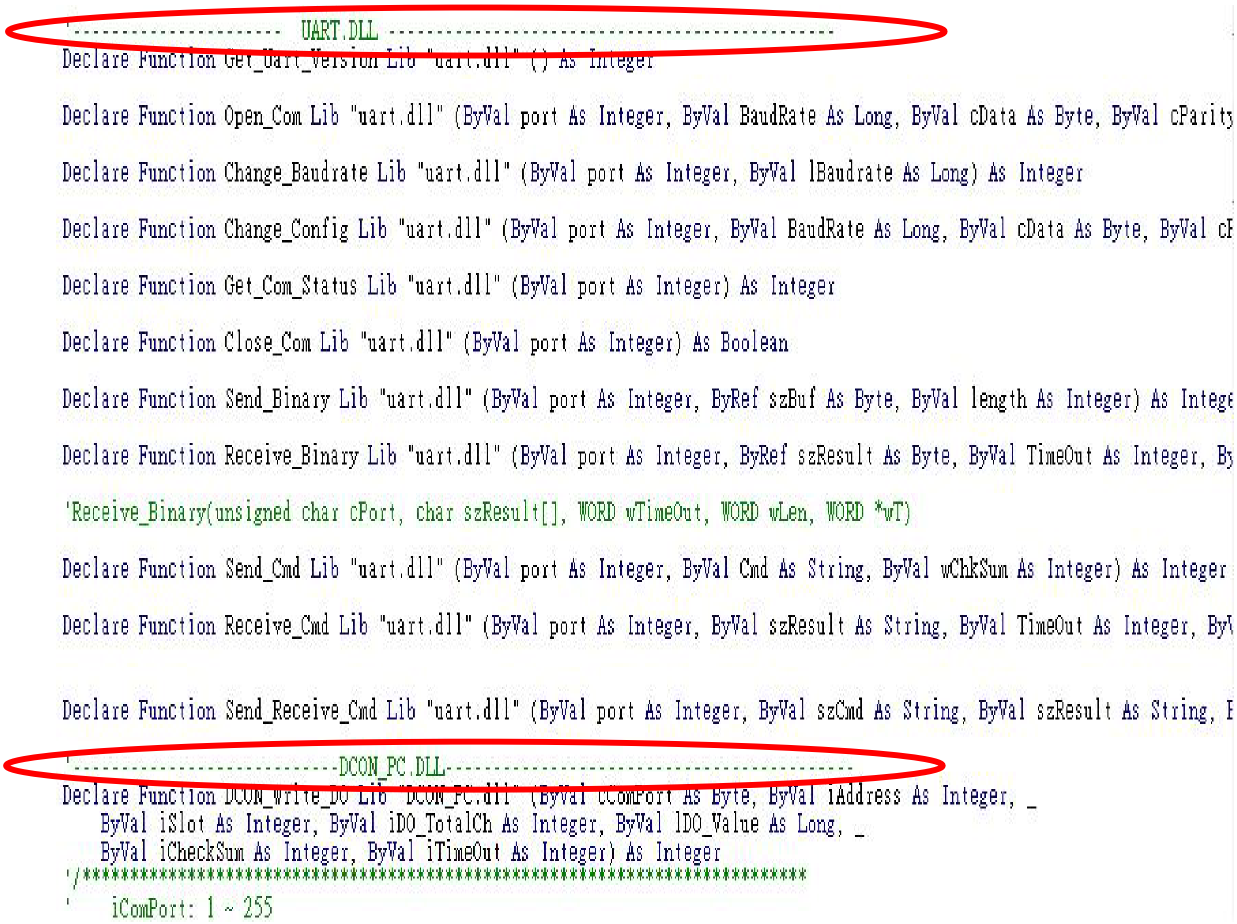
The user of PC has to implement these files as following:

- \DCON_DLL\Driver\UART.DLL → functions to deal with RS-232
- \DCON_DLL_NEW\DLL\DCON_PC.DLL → functions for A/D, D/A, D/I, D/O, Counter, Frequency
- \DCON_DLL_NEW\DLL\New\DCON_PC.as → declarations for UART & DCON_PC.DLL

In the project files, users must include declaration files DCON_PC.bas into VB modules environment, as shown in below figure.



After double clicking on the DCON_PC.bas to open the file, users can see the declarations of function for UART.DLL and DCON_PC.DLL and some defined constant declarations



```
'---------------------  UART.DLL ---------------------------------------------
Declare Function Get_Uart_Version Lib "uart.dll" () As Integer

Declare Function Open_Com Lib "uart.dll" (ByVal port As Integer, ByVal BaudRate As Long, ByVal cData As Byte, ByVal cParit

Declare Function Change_Baudrate Lib "uart.dll" (ByVal port As Integer, ByVal lBaudrate As Long) As Integer

Declare Function Change_Config Lib "uart.dll" (ByVal port As Integer, ByVal BaudRate As Long, ByVal cData As Byte, ByVal cP

Declare Function Get_Com_Status Lib "uart.dll" (ByVal port As Integer) As Integer

Declare Function Close_Com Lib "uart.dll" (ByVal port As Integer) As Boolean

Declare Function Send_Binary Lib "uart.dll" (ByVal port As Integer, ByRef szBuf As Byte, ByVal length As Integer) As Intege

Declare Function Receive_Binary Lib "uart.dll" (ByVal port As Integer, ByRef szResult As Byte, ByVal TimeOut As Integer, By

'Receive_Binary(unsigned char cPort, char szResult[], WORD wTimeOut, WORD wLen, WORD *wT)

Declare Function Send_Cmd Lib "uart.dll" (ByVal port As Integer, ByVal Cmd As String, ByVal wChkSum As Integer) As Integer

Declare Function Receive_Cmd Lib "uart.dll" (ByVal port As Integer, ByVal szResult As String, ByVal TimeOut As Integer, By


Declare Function Send_Receive_Cmd Lib "uart.dll" (ByVal port As Integer, ByVal szCmd As String, ByVal szResult As String,
'---------------------------DCON_PC.DLL----------------------------------------
Declare Function DCON_Write_DO Lib "DCON_PC.dll" (ByVal cComPort As Byte, ByVal iAddress As Integer, _
    ByVal iSlot As Integer, ByVal iDO_TotalCh As Integer, ByVal lDO_Value As Long, _
    ByVal iCheckSum As Integer, ByVal iTimeOut As Integer) As Integer
'/***************************************************************************
'    iComPort: 1 ~ 255
```

2.3. Using Embedded VISUAL C++ for Wincon

The demo program are tested OK in Wince and EVC 4.0 version. From Microsoft website, user can download the free EVC++ 4.0.
Website: http://msdn.microsoft.com/downloads/Default.aspx

How to create the new project of wince?
Step 1: Installing Embedded Visual C++ 4.0
Please refer to Microsoft website and look up related information.
Step 2: Installing EVC++4.0 Service Pack 4(SP4)
Please refer to Microsoft website and look up related information.
Step 3: Installing WinconSDK to your PC.
Download website:
http://www.icpdas.com/products/PAC/wincon-8000/Download/download_SDK.htm
CD:
W-8x3x:
CD\Napdos\WinCE\SDK
W-8x4x:
CD\SDK\

The user of wincon has to implement these files as following:

- ◆ \Program Files\Windows CE Tools\wce410\include\Armv4\Uartce.h
- ◆ \Program Files\Windows CE Tools\wce410\lib\Armv4\Uartce.lib

→ functions to deal with RS-232 or 485

- ◆ \Program Files\Windows CE Tools\wce410\include\Armv4\WinconSDK.h
- ◆ \Program Files\Windows CE Tools\wce410\lib\Armv4\WinconSDK.lib

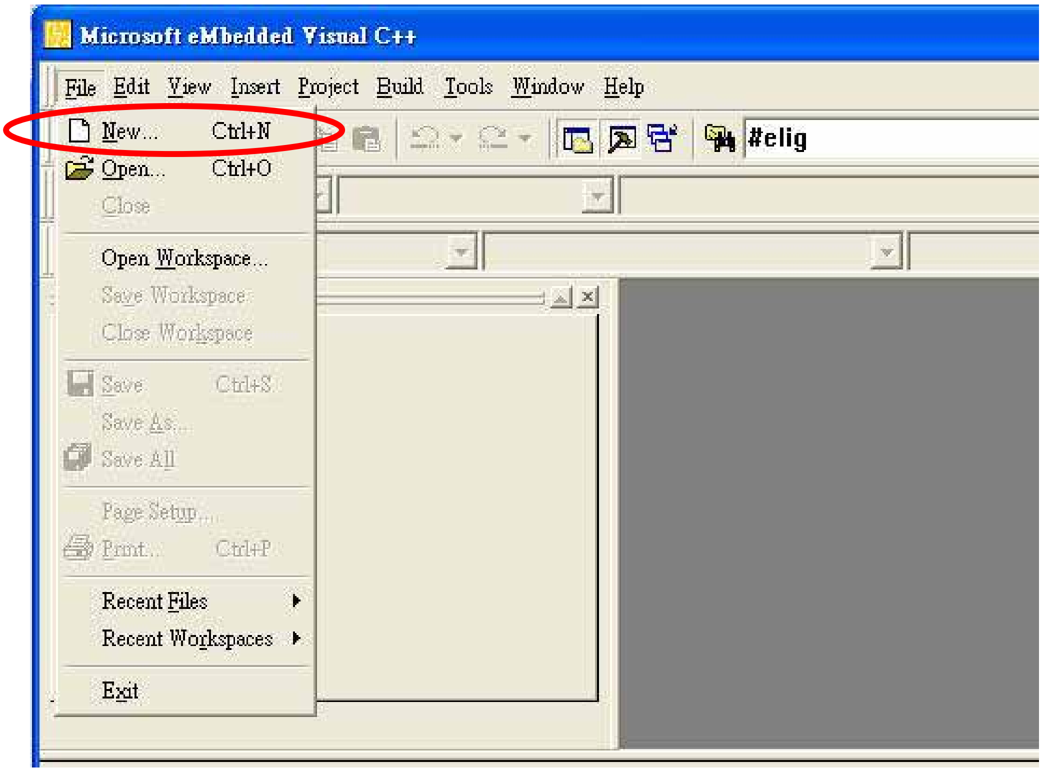
→ I-8000 series module function

- ◆ \Program Files\Windows CE Tools\wce410\include\Armv4\Dcon_Fun.h
- ◆ \Program Files\Windows CE Tools\wce410\lib\Armv4\Dcon_Fun.lib

→ functions for A/D, D/A, D/I, D/O, Counter, Frequency

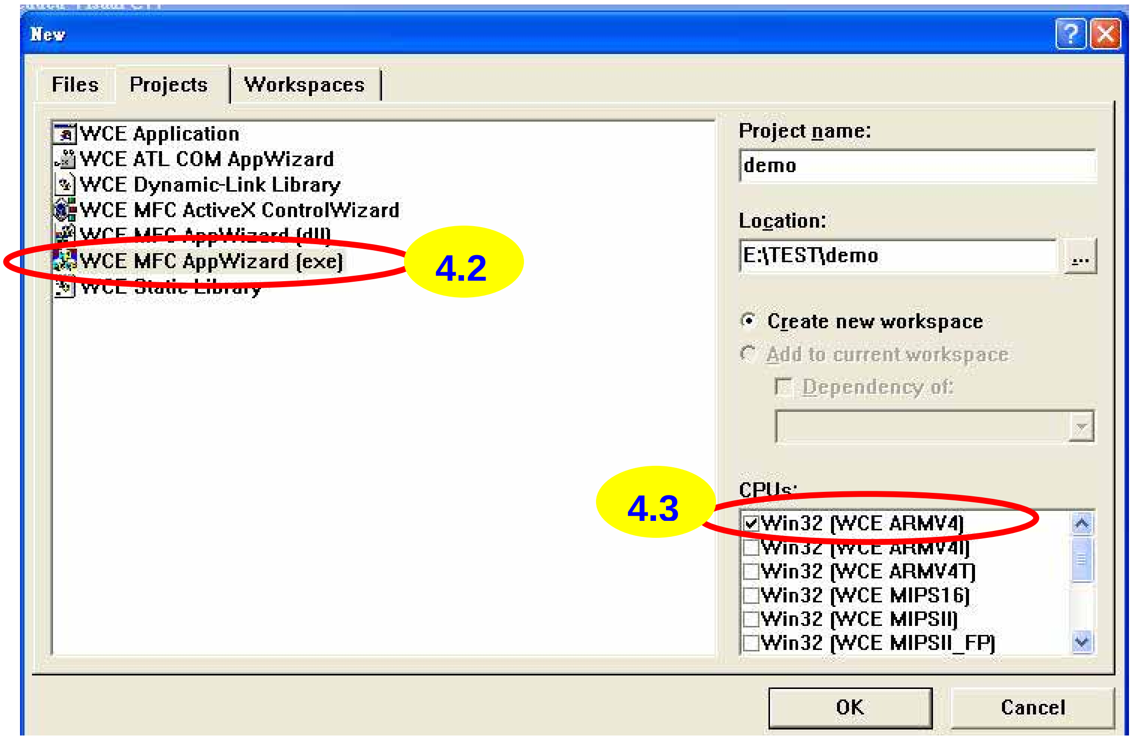
Step 4: Use EVC++4.0

Step 4.1: Create new project



Step 4.2: Select WCE MFC AppWizard[exe]

Step 4.3: Select Win32[WCE ARMV4]



Project name: demo

Location: E:\TEST\demo

● Create new workspace

○ Add to current workspace

□ Dependency of:

CPUs:

☑ Win32 (WCE ARMV4)

□ Win32 (WCE ARMV4I)

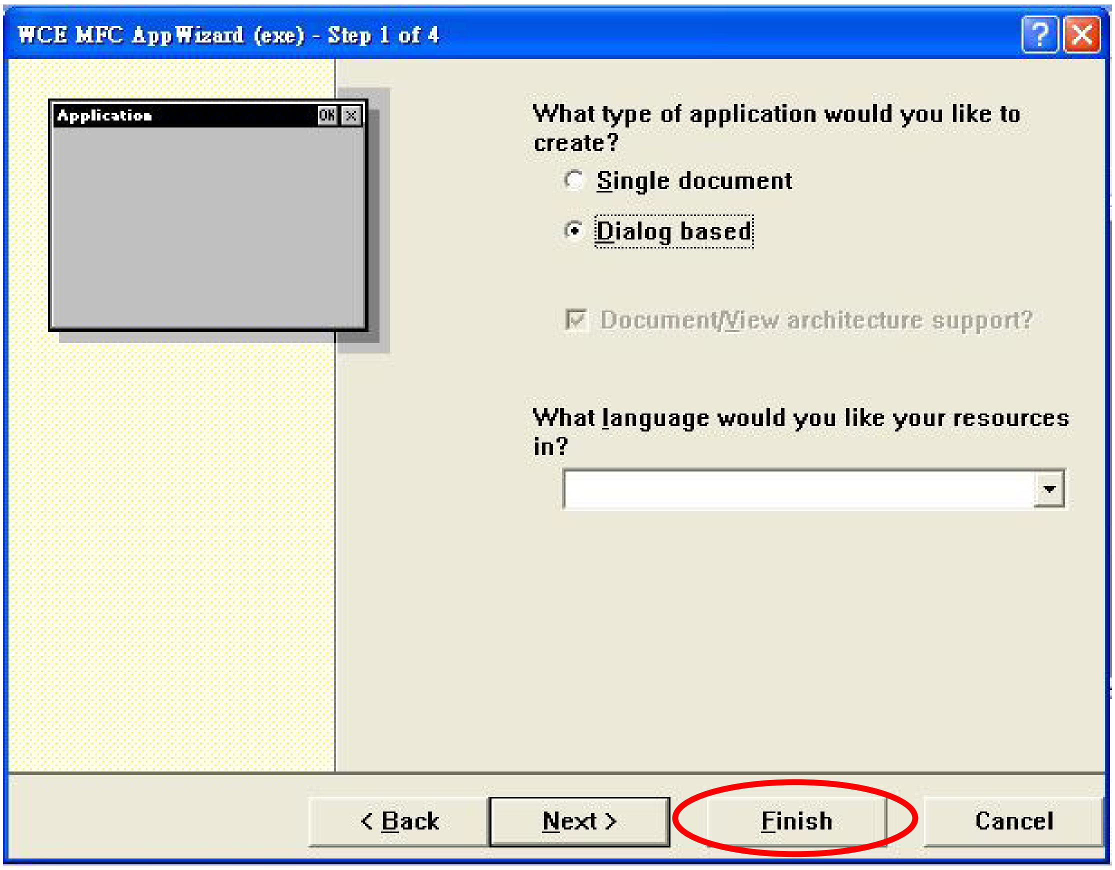
□ Win32 (WCE ARMV4T)

□ Win32 (WCE MIPS16)

□ Win32 (WCE MIPSII)

□ Win32 (WCE MIPSII_FP)

Step 4.4: Select your application and press “Finish ” buttom



WCE MFC AppWizard (exe) - Step 1 of 4

What type of application would you like to create?

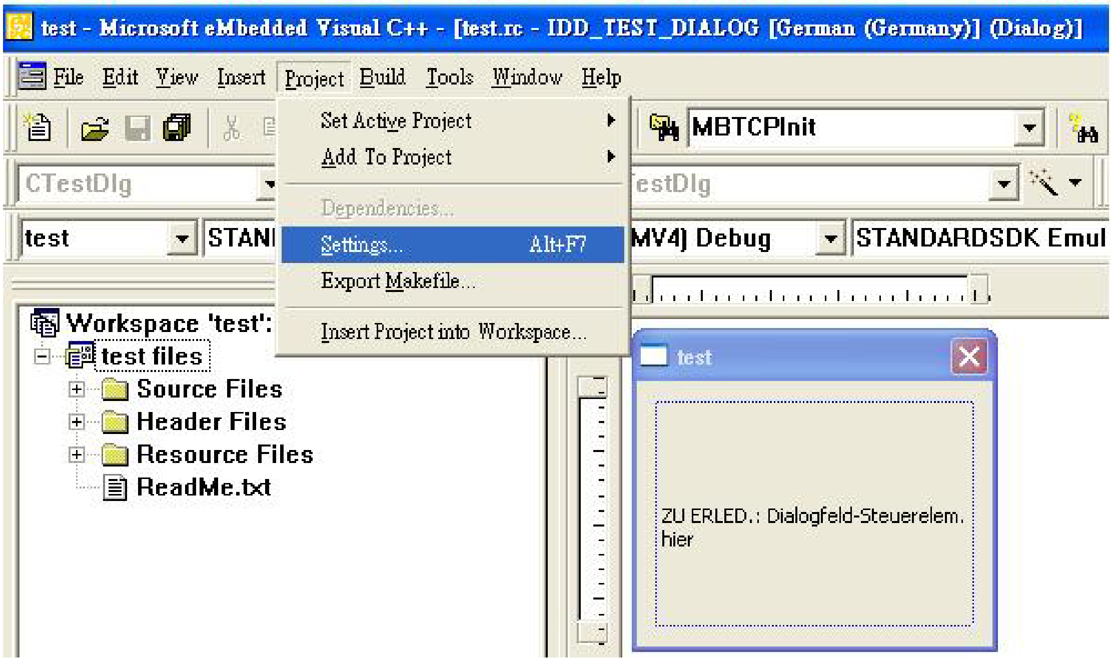
○ Single document

● Dialog based

☑ Document/View architecture support?

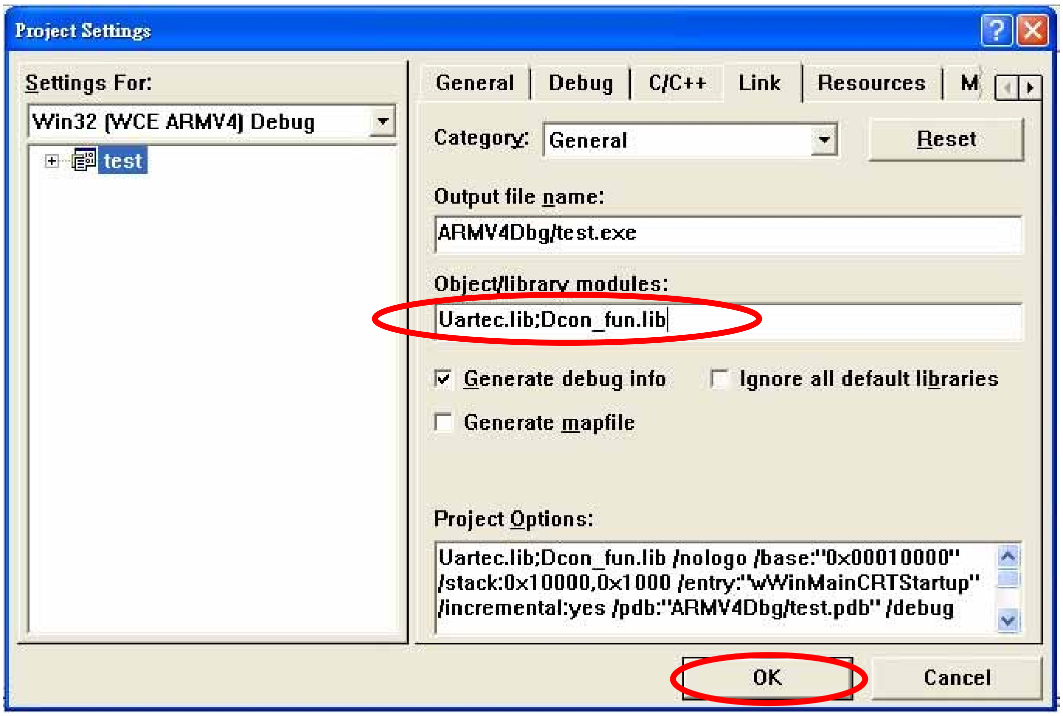
What language would you like your resources in?

Step 4.5: Select the project of Menu and choose the Setting?



test - Microsoft eMbedded Visual C++ - [test.rc - IDD_TEST_DIALOG [German (Germany)] (Dialog)]

Step 4.6: Select “Link” and key in “Uart.lib” and “Dcon_Fun.lib” in the Object/library modules field. Then press ”OK” button.



Project Settings

Settings For: Win32 (WCE ARMV4) Debug

Category: General

Output file name: ARMV4Dbg/test.exe

Object/library modules: Uartec.lib;Dcon_fun.lib

☑ Generate debug info

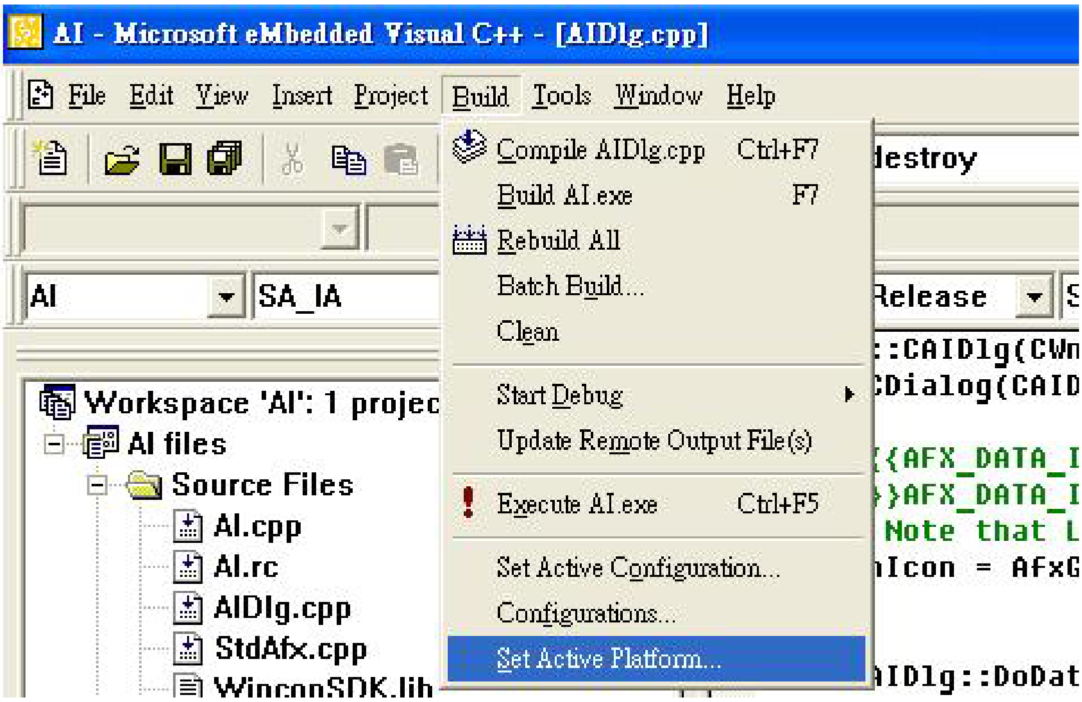
□ Ignore all default libraries

□ Generate mapfile

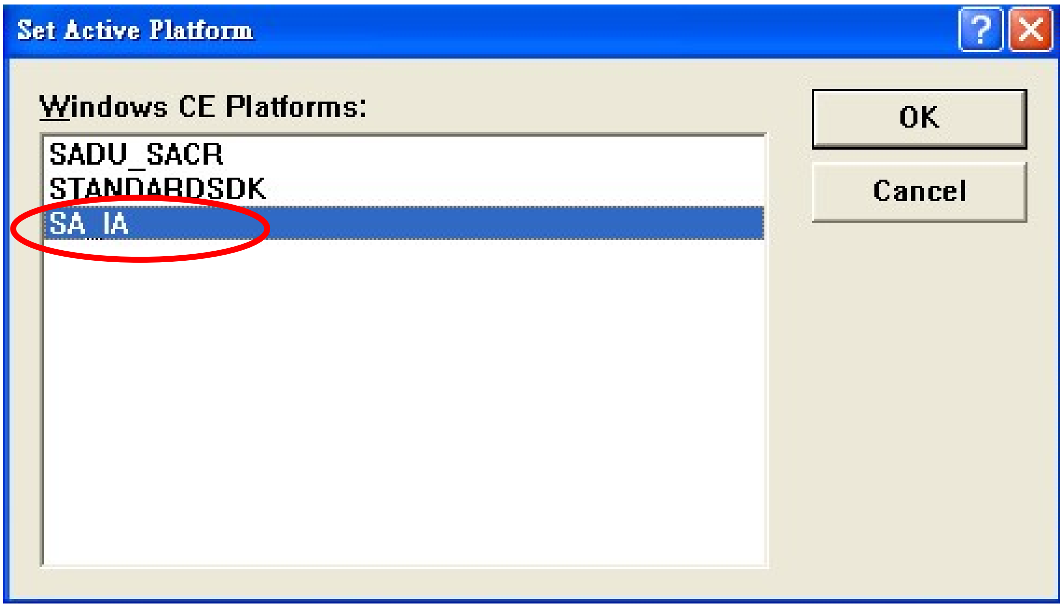
Project Options:

```
Uartec.lib;Dcon_fun.lib /nologo /base:"0x00010000"
/stack:0x10000,0x1000 /entry:"wWinMainCRTStartup"
/incremental:yes /pdb:"ARMV4Dbg/test.pdb" /debug
```

Step 4.7: Select the Build of Menu and choose the “Set Active Platform…”



Step 4.8: Select “SA_IA” in the Windows CE Platforms. Then press OK button.



If users need the detail information, please refer to the following content of web site or CD disk:

W-8x3x:

Web: ftp://ftp.icpdas.com.tw/pub/cd/winconcd/napdos/wince/user manual/wincon getting started 1.4.pdf

CD:\Napdos\Wince\User Manual\ WinCON Getting Started 1.4.pdf

W-8x4x:

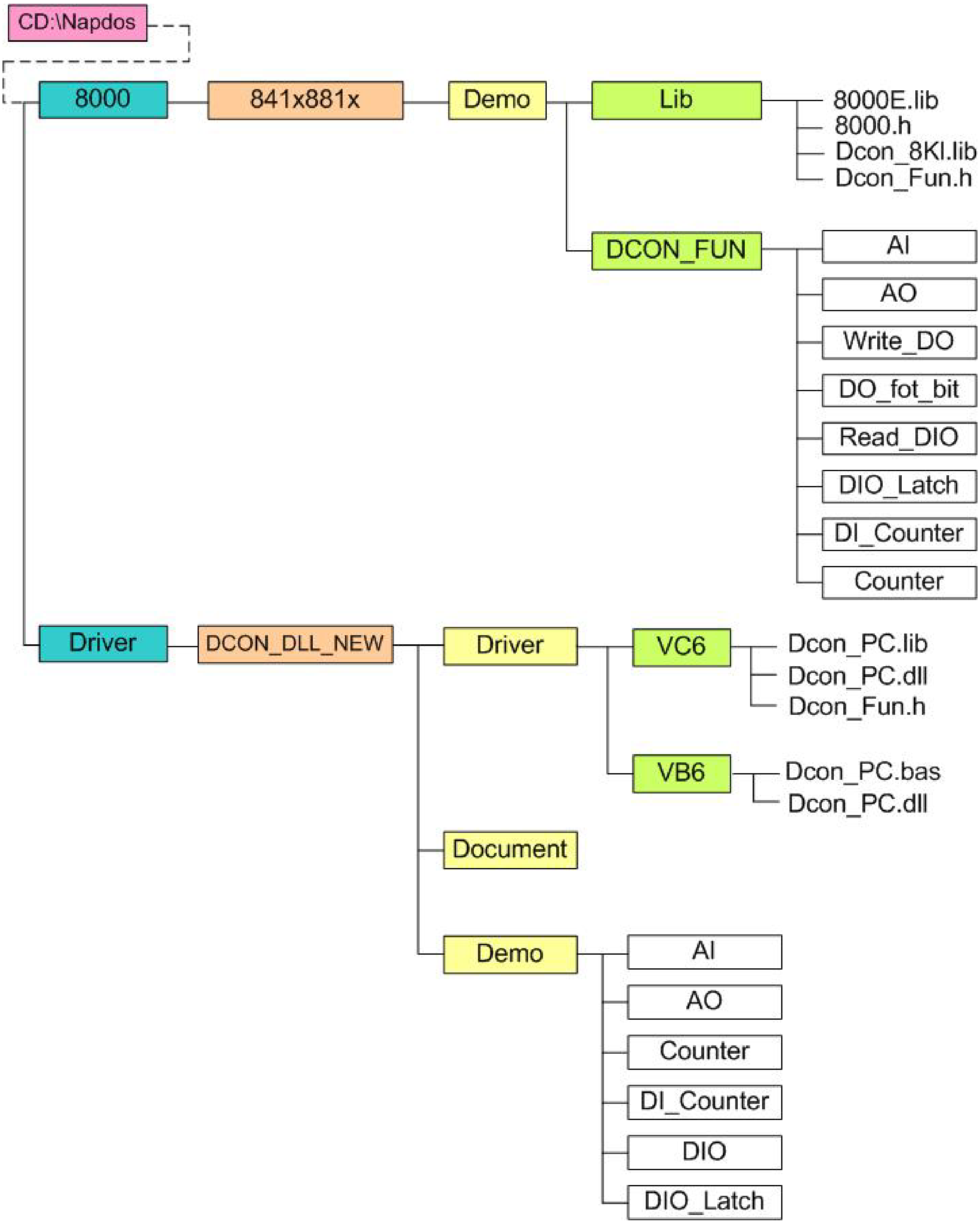
Web: ftp://ftp.icpdas.com.tw/pub/cd/w-8x4x/user manual/wincon getting started 1.4.pdf

CD:\User Manual\WinCON Getting Started 1.4.pdf

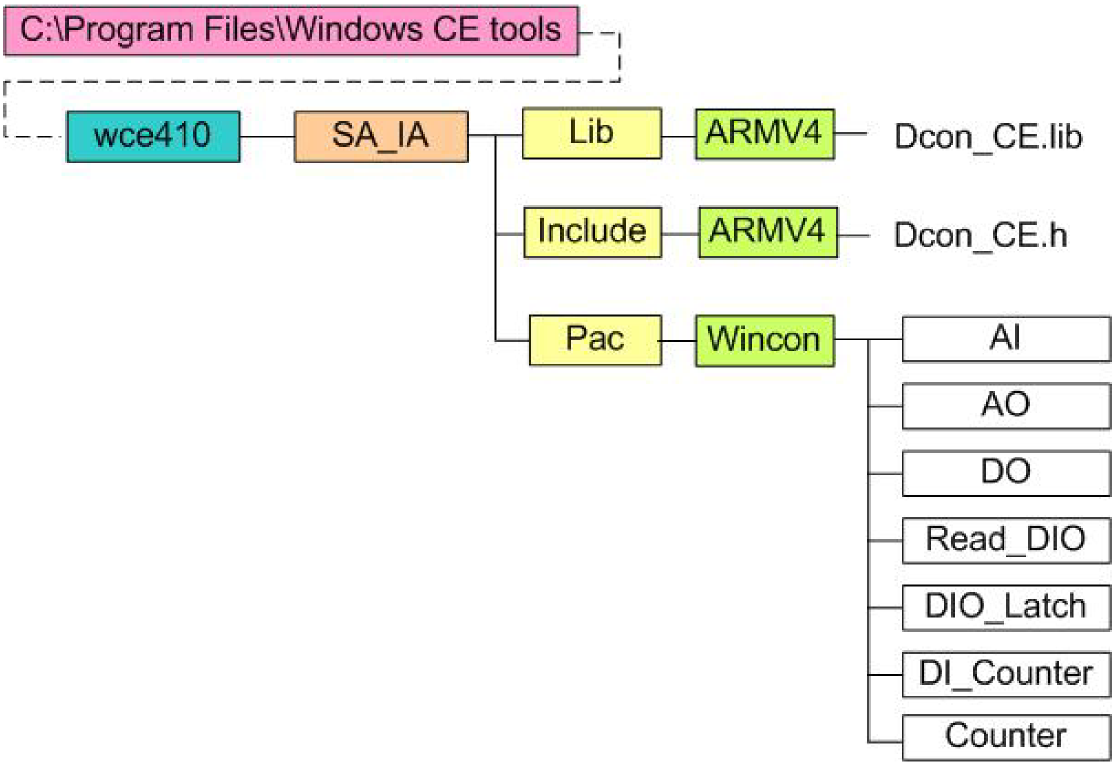
3. Demo List

Based on the demo programs, User can easily understand how to use the function and develop their own application in a quick way.

Sub of directory tree of 8000 or PC



Sub of directory tree of Wincon



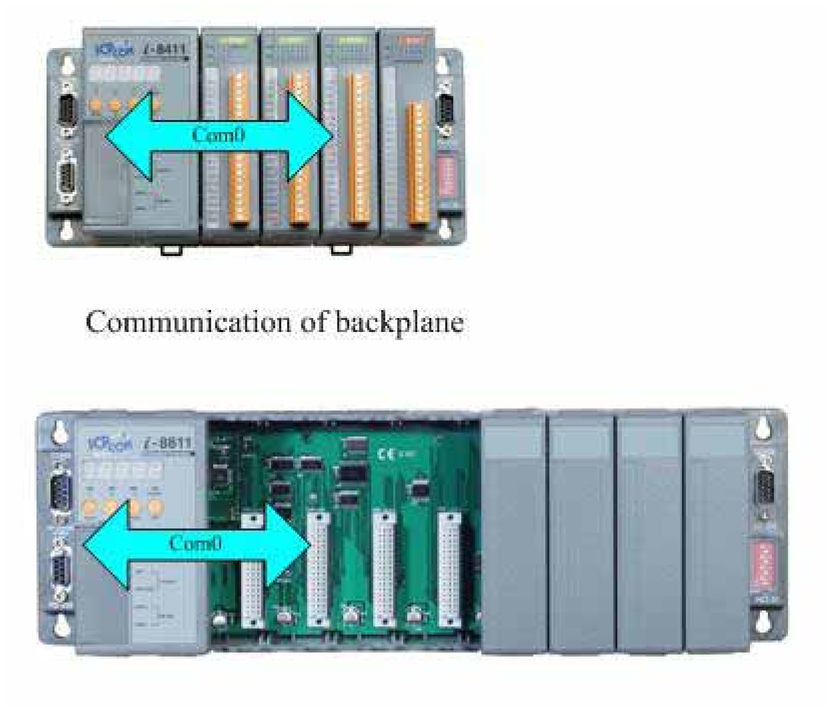
Note:

If users can’t find “wce410” in the “Windows CE tools” of Program Files, please install WinconSDK.

The location of WinconSDK:
CD:\Napdos\WinCE\SDK
http://www.icpdas.com/products/PAC/wincon-8000/Download/download_SDK.htm

3.1 Usage Mode of Demo for I-8000

Mode 1: The operation of Backplane

I-8000 parameter setting	Mode 1
Means of commumication (ComPort)	Backplane: Com0 (fixed)
Address	0 (fixed)
Baudrate	115200 (fixed)
Slot	0~3 or 0~7
Timeout	User define
CheckSum	0: Disable
Announcements	Only support 87K module.
 Communication of backplane	

Note:

Install the serial I/O modules(87K) into the I-8000 controller, the 87K module will go to its initial state as following:

Module address=00

Baud rate=115200

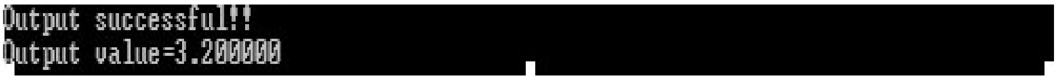
Checksum=disable

If user needs to look up the pin assignment, please refer to Appendix A.

Demo code:

```
int iRet;
float fRead_value;

InitLib();
InstallCom(0,115200,8,0,1);
//Comport: 0, Baudrate:115200, Databit: 8, Stopbit: 0, ParityBit: 1

iRet=DCON_Write_AO(0,0,2,0,2,3.2,0,100);
//Comport: 0, Address: 0, Slot: 2, SingleChannel: 0, TotalChannel: 2, AOVal: 3.2,
//Checksum: disable, Timeout: 100
if(iRet==NoError) {
      Print("Output successful!!\n\r");
      iRet=DCON_Read_AO(0,0,2,0,2,0,100,&fRead_value);
      //Comport: 0, Address: 0, Slot: 2, SingleChannel: 0, TotalChannel: 2,
      //Checksum: disable, Timeout: 100

      if(iRet==NoError) {
          Print("Output value=%f\n\r",fRead_value);
      }
      else
          Print("Reading Analog Output error!! Error Code %d\n\r",iRet);
}
else
     Print("Output error!! Error Code %d\n\r",iRet);
```

Test result:

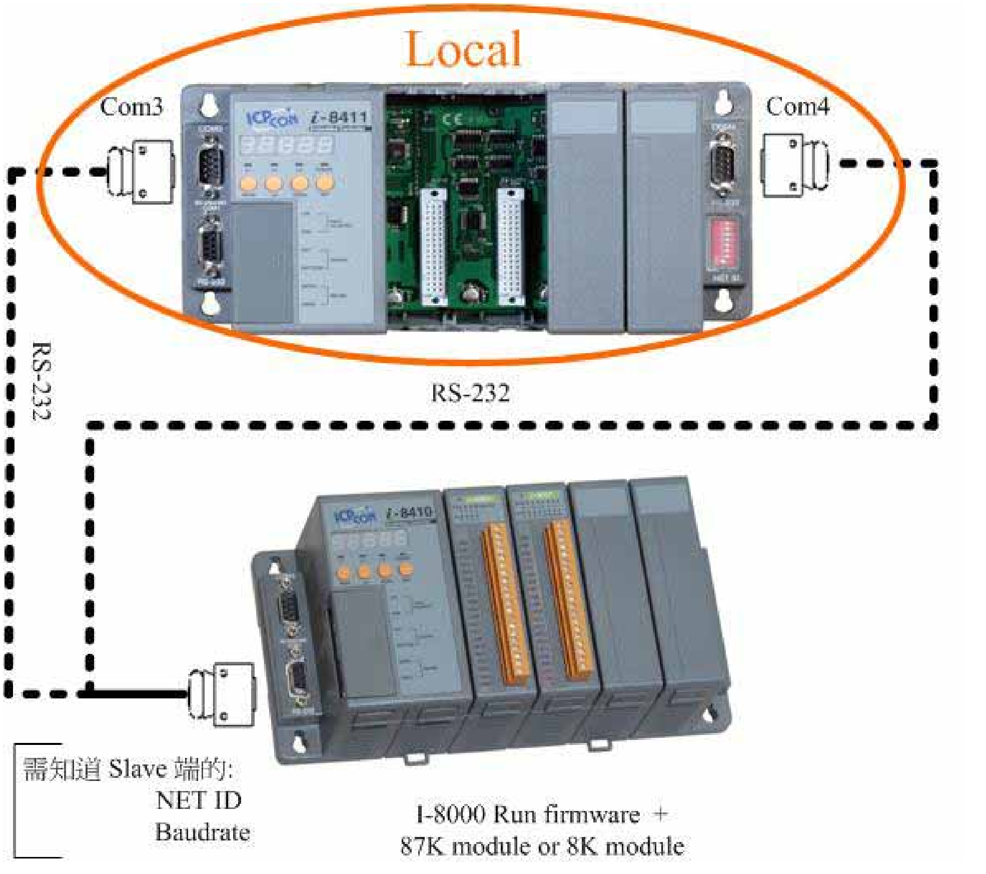


```
Output successful!!
Output value=3.200000
```

Mode 2: The I-8000 acts as master to control another I-8000 that run DCON firmware (slave type).

Location of DCON firmware:

CD:\NAPDOS\DCON\ or ftp://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/dcon/

I-8000 parameter setting	**Mode 2**
Means of Commumication (ComPort)	User can use com2, com3, com4 to connect Slave. (Slave: I-8000 run firmware + 8K and 87K module)
Address	NET ID: 0~255
Baudrate	Speed of Slave(1200~115200)
Slot	0~3 or 0~7
Timeout	User define
CheckSum	0: Disable 1: Enable
Announcements	1. Only support 8K or 87K module. 2. User need to know the address and baudrate of slave.
	

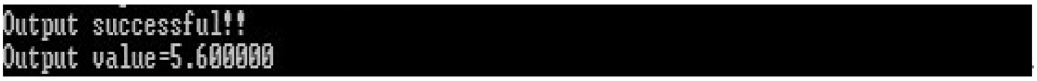
Note:

If user needs to look up the pin assignment, please refer to Appendix A.

Demo code:

```
int iRet;
float fRead_value;

InitLib();
InstallCom(3,115200,8,0,1);
//Comport: 3, Baudrate:115200, Databit: 8, Stopbit: 0, ParityBit: 1

iRet=DCON_Write_AO(3,2,1,0,2,3.2,0,100);
//Comport: 3, Address: 2, Slot: 1, SingleChannel: 0, TotalChannel: 2, AOVal: 5.6,
//Checksum: disable, Timeout: 100
if(iRet==NoError) {
      Print("Output successful!!\n\r");
      iRet=DCON_Read_AO(3,2,1,0,2,0,100,&fRead_value);
      //Comport: 3, Address: 2, Slot: 1, SingleChannel: 0, TotalChannel: 2,
      //Checksum: disable, Timeout: 100

      if(iRet==NoError) {
          Print("Output value=%f\n\r",fRead_value);
      }
      else
          Print("Reading Analog Output error!! Error Code %d\n\r",iRet);
}
else
     Print("Output error!! Error Code %d\n\r",iRet);
```

Test result:

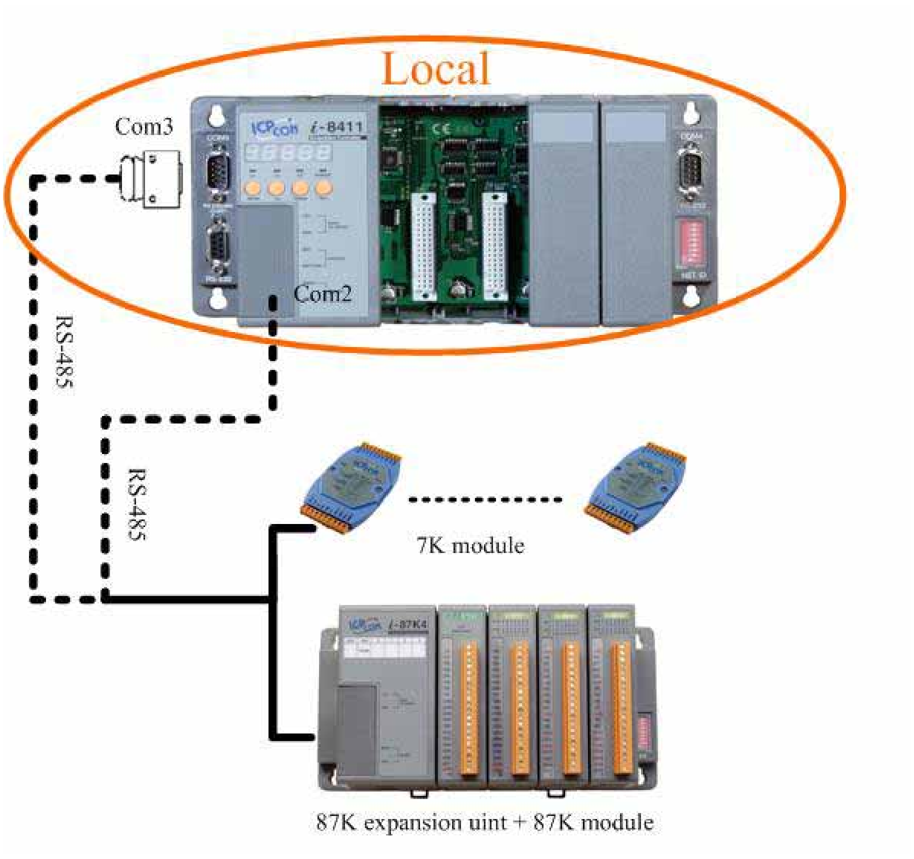


```
Output successful!!
Output value=5.600000
```

Note:

If use Com2 as communication interface please add AddCom2Fun().

Mode 3: I-8000 connect to 7K or 87K module.

I-8000 parameter setting	**Mode 3**
Means of Commumication (ComPort)	Com3(Rs-485) and Com2
Address	Address of 7K or 87K module
Baudrate	Baudrate of 7K or 87K module
Slot	-1 (NONE)
Timeout	User define
CheckSum	0: Disable 1: Enable
Announcements	Support 7K or 87K module
	

Note:

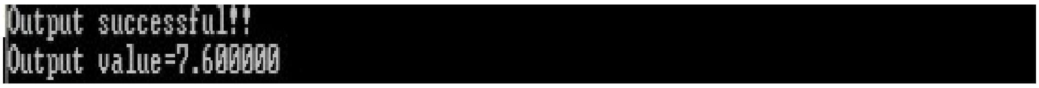
If user connects to the serial modules directly, user needs to set the “slot” to “–1”. The “–1” represents NONE.

If user needs to look up the pin assignment, please refer to Appendix A.

Demo code:

```
int iRet;
float fRead_value;

InitLib();
InstallCom(3,115200,8,0,1);
//Comport: 3, Baudrate:115200, Databit: 8, Stopbit: 0, ParityBit: 1

iRet=DCON_Write_AO(3,2,-1,0,2,7.6,0,100);
//Comport: 3, Address: 2, Slot: -1, SingleChannel: 0, TotalChannel: 2, AOVal: 7.6
//Checksum: disable, Timeout: 100
if(iRet==NoError) {
        Print("Output successful!!\n\r");
        iRet=DCON_Read_AO(3,2,-1,0,2,0,100,&fRead_value);
        //Comport: 3, Address: 2, Slot: -1, SingleChannel: 0, TotalChannel: 2,
        //Checksum: disable, Timeout: 100

        if(iRet==NoError) {
              Print("Output value=%f\n\r",fRead_value);
        }
        else
              Print("Reading Analog Output error!! Error Code %d\n\r",iRet);
}
else
      Print("Output error!! Error Code %d\n\r",iRet);
```

Test result:

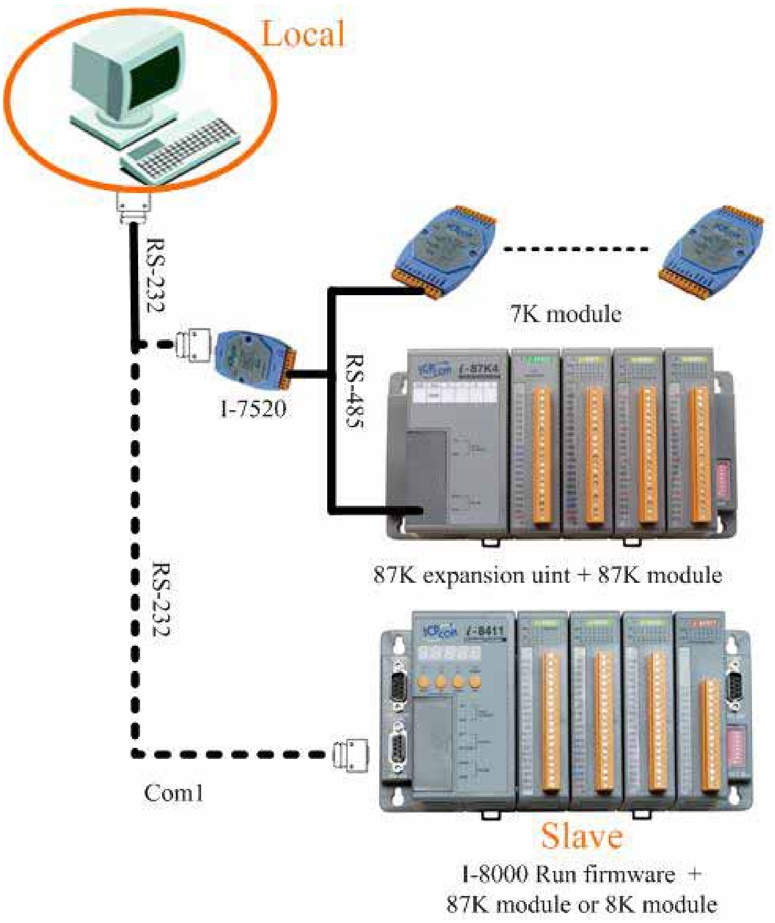


```
Output successful!!
Output value=7.600000
```

Note:

If use Com2 as communication interface please add AddCom2Fun() before installCom2.

3.2 Usage Mode of Demo for PC

PC parameter setting	
Means of Commumication (ComPort)	Com Port: 0~255
Address	Mode 1: Address of 7K or 87K module Mode 2: Address of slave (NET ID)
Baudrate	Mode 1: Baudrate of 7K or 87K module Mode 2: Speed of Slave
Slot	Mode 1: -1 (NONE) Mode 2: 0~4 or 0~8
Timeout	User define
CheckSum	0: Disable 1: Enable
Announcements	Mode 1: Support 7K or 87K module Mode 2: Only support 8K or 87K module. User need to know the address and baudrate of slave.
	

Mode1: PC connect to7K or 87K module.

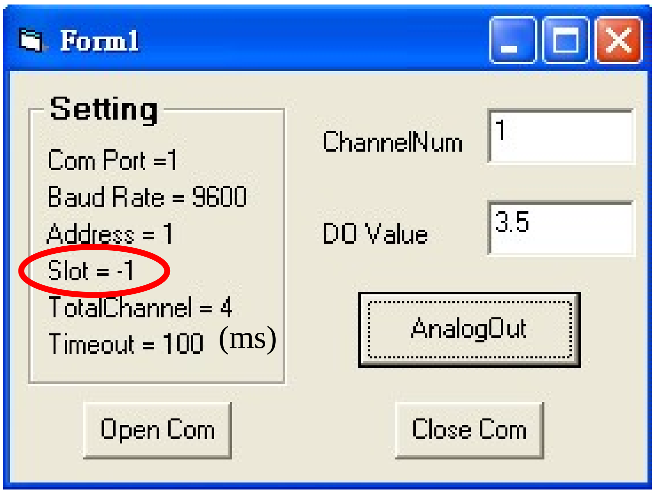
User can refer to the following demo to set parameter.

```
Step1   Private Sub CmdOpen_com_Click()
            iRet = Open_Com(1, 9600, 8, 0, 0)
        End Sub
```

```
            Private Sub CmdAnalogOut_Click()
                Dim fDO_Value As Single
                Dim iChannelNum As Integer

                iChannelNum = CInt(TxtChannelNum.Text)
                fDO_Value = Val(TxtDO_Value.Text)

Step2           iRet = DCON_Write_AO(1, 1, -1, iChannelNum, 4, fDO_Value, 0, 100)
                ' Comport: 1, Address: 1, Slot: -1, TotalChannel: 4, Checksum: disable
                ' Timeout: 100
                If iRet Then
                        MsgBox "Error Code" & Str(iRet), vbCritical, "Error"
                End If
            End Sub
```

```
Step3   Private Sub CmdClose_Com_Click()
            Close_Com (1)
        End Sub
```

Test result:



Setting

Com Port =1

Baud Rate = 9600

Address = 1

Slot = -1

TotalChannel = 4

Timeout = 100 (ms)

ChannelNum: 1

DO Value: 3.5

Mode2: PC connect to Slave(I-8000 run firmware).

User can refer to the following demo to set parameter.

Step1

```
Private Sub CmdOpen_com_Click()
    iRet = Open_Com(1, 9600, 8, 0, 0)
End Sub
```

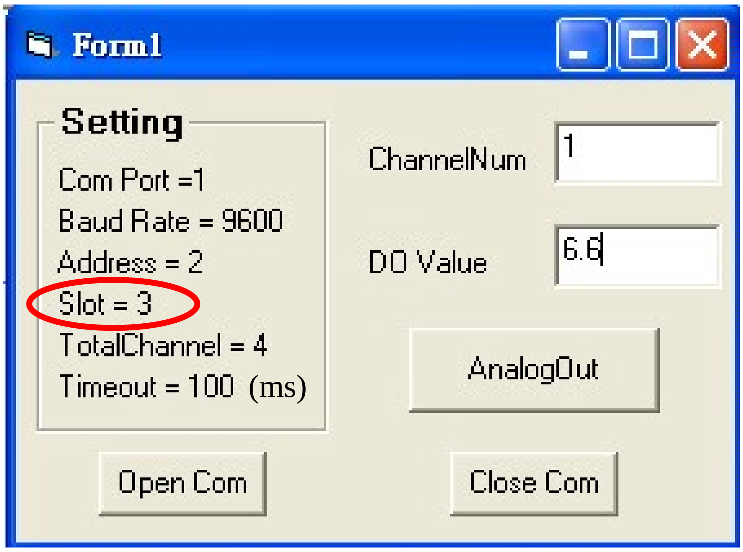
Step2

```
Private Sub CmdAnalogOut_Click()
    Dim fDO_Value As Single
    Dim iChannelNum As Integer

    iChannelNum = CInt(TxtChannelNum.Text)
    fDO_Value = Val(TxtDO_Value.Text)

    iRet = DCON_Write_AO(1, 2, 3, iChannelNum, 4, fDO_Value, 0, 100)
    ‘Comport: 1, Address: 2, Slot: 3, TotalChannel: 4, Checksum: disable
    ‘Timeout: 100
    If iRet Then
            MsgBox "Error Code" & Str(iRet), vbCritical, "Error"
    End If
End Sub
```

Step3

```
Private Sub CmdClose_Com_Click()
    Close_Com (1)
End Sub
```

Test result:



Setting

Com Port =1

Baud Rate = 9600

Address = 2

Slot = 3

TotalChannel = 4

Timeout = 100 (ms)

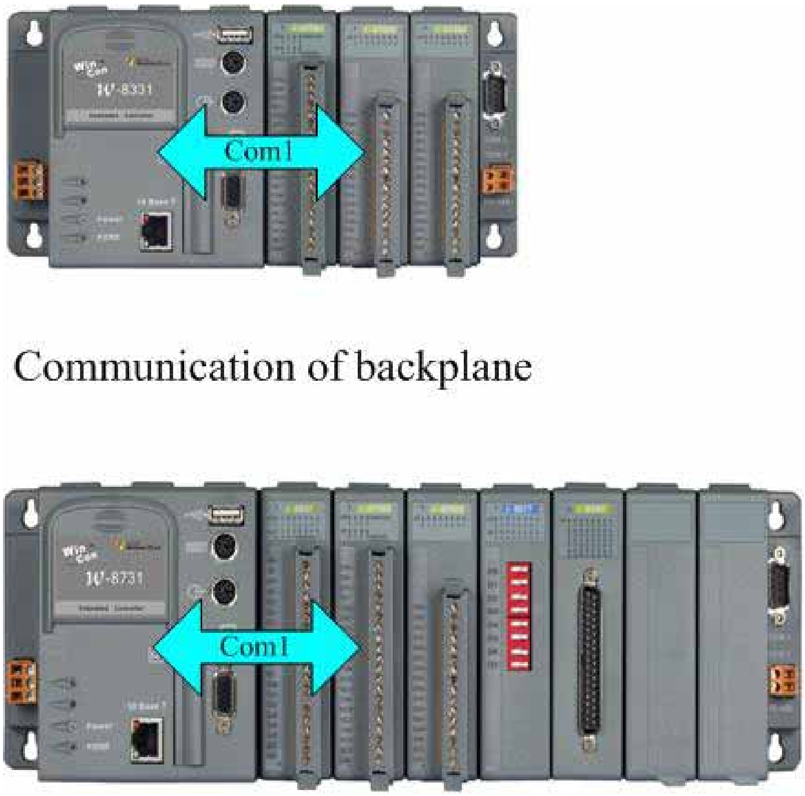
ChannelNum: 1

DO Value: 6.6

3.3 Usage Mode of Demo for Wincon-8000

Mode 1: Operation of Backplane

Wincon-8000 parameter setting	Mode 1
Means of Commumication (ComPort)	Backplane: Com1 (fixed)
Address	0 (fixed)
Baudrate	115200 (fixed)
Slot	1~3 or 1~7
Timeout	User define
CheckSum	0: Disable 1: Enable
Announcements	Only support 87K module.



Communication of backplane

Note:

Install the serial I/O modules(87K) into the W-8000 controller, the 87K module will go to its initial state as following:

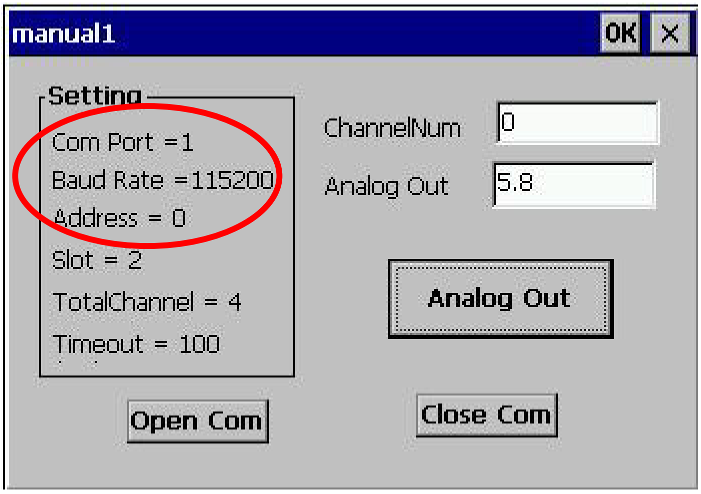
Module address=00
Baud rate=115200
Checksum=disable

If user needs to look up the pin assignment, please refer to Appendix B.

Demo code:

```
void CManual1Dlg::OnOpen_Com()
{
        // TODO: Add your control notification handler code here
        Open_Com(1,115200,8,0,1);
}

void CManual1Dlg::OnClose_Com()
{
        // TODO: Add your control notification handler code here
        Close_Com(1);
}

void CManual1Dlg::OnAnalog_Out()
{
        // TODO: Add your control notification handler code here
        int iRet;
        int iChannelNum;
        float fAnalog_out;
        CString sTemp;
        char cTemp[20];

        UpdateData();

        //ChannelNum
        m_ChannelNum.GetWindowText(sTemp);
        wcstombs(cTemp,sTemp,6);
        iChannelNum=atoi(cTemp);

        //AO_Value
        m_AnalogOut.GetWindowText(sTemp);
        wcstombs(cTemp,sTemp,6);
        sscanf(cTemp,"%f",&fAnalog_out);

        iRet=DCON_Write_AO(1,0,1,iChannelNum,4,fAnalog_out,0,100);
                //Com port: 1, Address: 0, Slot: 2, TotalChannel: 4, Checksum: disable,
                //Timeout: 100(ms)
}
```

Test result:



Com Port =1

Baud Rate =115200

Address = 0

Slot = 2

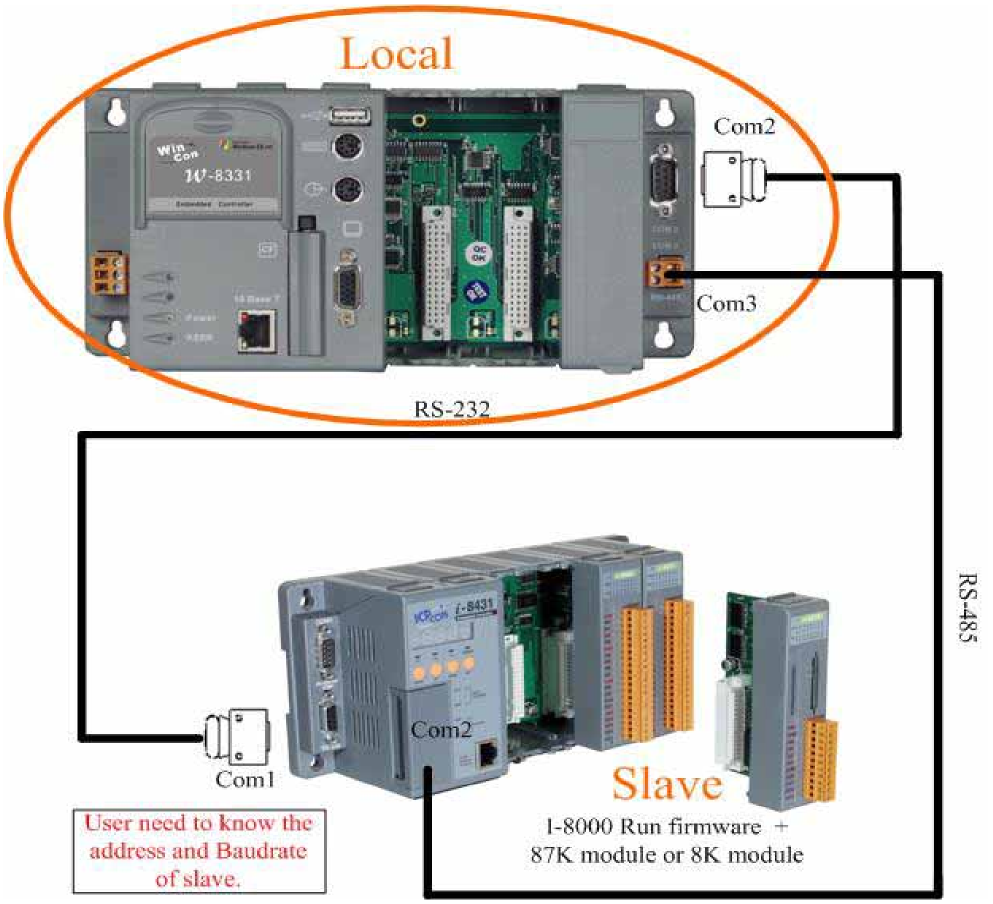
TotalChannel = 4

Timeout = 100

ChannelNum 0

Analog Out 5.8

Mode 2: Wincon connet to I-8000 run firmware

Wincon-8000 parameter setting	**Mode 2**
Means of Commumication (ComPort)	User can use com2, com3 to connect Slave. (Slave: Wincon-8000 run firmware + 8K and 87K module).
Address	(NET ID: 0~255)
Baudrate	Speed of Slave(1200~115200)
Slot	0~3 or 0~7
Timeout	User define
CheckSum	0: Disable 1: Enable
Announcements	1. Only support 8K or 87K module. 2. User need to know the address and baudrate of slave.
	

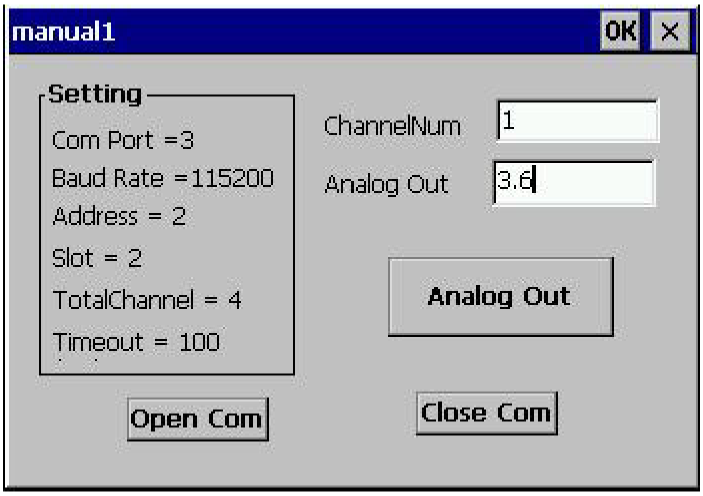
Note:

If user needs to look up the pin assignment, please refer to Appendix B.

Demo code:

```
void CManual1Dlg::OnOpen_Com()
{
        // TODO: Add your control notification handler code here
        Open_Com(3,115200,8,0,1); //ComPort: 3, Baudrate:115200, DataBit:8,
                                  //ParityBit: 0, StopBit: 1
}

void CManual1Dlg::OnClose_Com()
{
        // TODO: Add your control notification handler code here
        Close_Com(3);
}

void CManual1Dlg::OnAnalog_Out()
{
        // TODO: Add your control notification handler code here
        int iRet;
        int iChannelNum;
        float fAnalog_out;
        CString sTemp;
        char cTemp[20];

        UpdateData();

        //ChannelNum
        m_ChannelNum.GetWindowText(sTemp);
        wcstombs(cTemp,sTemp,6);
        iChannelNum=atoi(cTemp);

        //AO_Value
        m_AnalogOut.GetWindowText(sTemp);
        wcstombs(cTemp,sTemp,6);
        sscanf(cTemp,"%f",&fAnalog_out);

        iRet=DCON_Write_AO(3,2,2,iChannelNum,4,fAnalog_out,0,100);
                //Com port: 3, Address: 2, Slot: 2, TotalChannel: 4, Checksum: disable,
                //Timeout: 100(ms)
}
```

Test result:



Setting

Com Port =3

Baud Rate =115200

Address = 2

Slot = 2

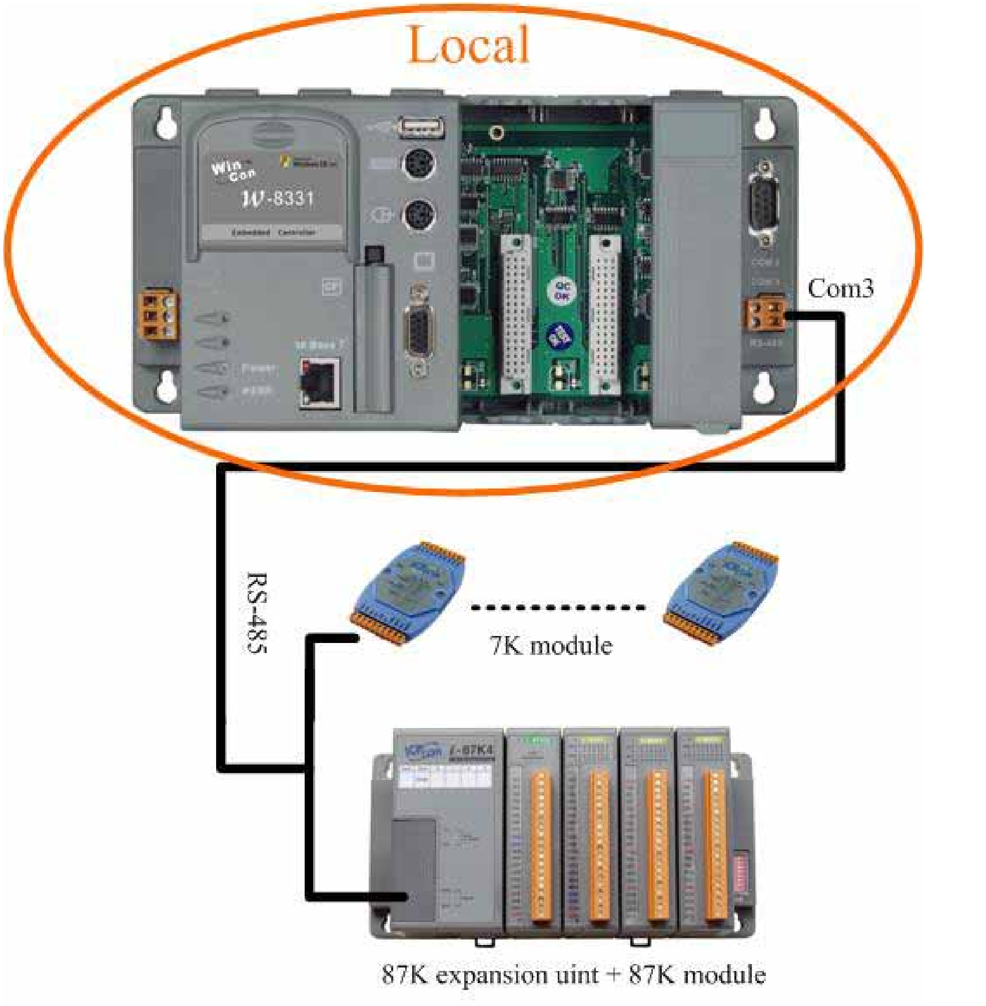
TotalChannel = 4

Timeout = 100

ChannelNum 1

Analog Out 3.6

Mode 3: Wincon connect to 7K or 87K module

Wincon-8000 parameter setting	**Mode 3**
Means of Commmication (ComPort)	Com3
Address	Address of 7K or 87K module
Baudrate	Baudrate of 7K or 87K module
Slot	-1 (NONE)
Timeout	User define
CheckSum	0: Disable 1: Enable
Announcements	Support 7K or 87K module
 Local W-8331 Com3 RS-485 7K module 87K expansion uint + 87K module	

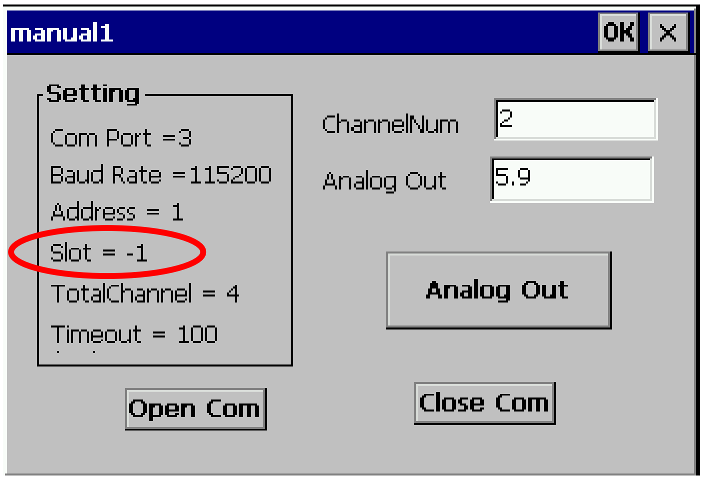
Note:

If user needs to look up the pin assignment, please refer to Appendix B.

Demo code

```
void CManual1Dlg::OnOpen_Com()
{
        // TODO: Add your control notification handler code here
        Open_Com(3,115200,8,0,1); //ComPort: 3, Baudrate:115200, DataBit:8,
                                            //ParityBit: 0, StopBit: 1
}

void CManual1Dlg::OnClose_Com()
{
        // TODO: Add your control notification handler code here
        Close_Com(3);
}

void CManual1Dlg::OnAnalog_Out()
{
        // TODO: Add your control notification handler code here
        int iRet;
        int iChannelNum;
        float fAnalog_out;
        CString sTemp;
        char cTemp[20];

        UpdateData();

        //ChannelNum
        m_ChannelNum.GetWindowText(sTemp);
        wcstombs(cTemp,sTemp,6);
        iChannelNum=atoi(cTemp);

        //AO_Value
        m_AnalogOut.GetWindowText(sTemp);
        wcstombs(cTemp,sTemp,6);
        sscanf(cTemp,"%f",&fAnalog_out);

        iRet=DCON_Write_AO(3,1,-1,iChannelNum,4,fAnalog_out,0,100);
                //Com port: 3, Address: 1, Slot: -1, TotalChannel: 4, Checksum: disable,
                //Timeout: 100(ms)
}
```

Test result:



manual1

Setting

Com Port =3

Baud Rate =115200

Address = 1

Slot = -1

TotalChannel = 4

Timeout = 100

ChannelNum: 2

Analog Out: 5.9

4. Function List

4.1. DCON_Write_DO

Description:

Output the value of the digital output module.

Syntax:

```
DCON_Write_DO(unsigned char cComPort,
              short iAddress,
              short iSlot,
              short iDO_TotalCh,
              unsigned long lDO_Value,
              short iCheckSum,
              short iTimeOut);
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0x00 to 0xFF
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iDO_TotalCh:	The total channel of DO module.
lDO_Value:	Digital output data
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms

4.2. DCON_Write_DO_Bit

Description:

Set the digital output value of the specific digital output channel No. of the digital output module. The output value is only for “0” or “1”.

Syntax:

```
DCON_Write_DO_Bit(unsigned char cComPort,
                  short iAddress,
                  short iSlot,
                  short iChannel,
                  short iDO_TotalCh,
                  short iBitValue,
                  short iCheckSum,
                  short iTimeOut)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The digital output channel No.
iDO_TotalCh:	The total channel of DO module.
iBitVaule:	0: off 1: on
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms

4.3. DCON_Read_DIO

Description:

Obtain the DI, DO, DIO value.

Syntax:

```
DCON_Read_DIO(unsigned char cComPort, short iAddress,
              short iSlot, short iDI_TotalCh,
              short iDO_TotalCh, short iCheckSum,
              short iTimeOut, unsigned long* lDI_Value,
              unsigned long* lDO_Value,
              char* cDI_BitValue, char* cDO_BitValue)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iDI_TotalCh:	The total channel of DI module.
iDO_TotalCh:	The total channel of DO module.
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms
iDI_Value:	Read Digital input data
iDO_Value:	Read Digital output data
cDI_BitValue:	Read Digital input data; 0 or 1
cDO_BitValue:	Read Digital output data; 0 or 1

4.4. DCON_READ_DIO_Latch

Description:

Obtain the latch value of DI, DO, DIO .

Syntax:

```
DCON_READ_DIO_Latch(unsigned char cComPort,
                    short iAddress,
                    short iSlot,
                    short iDI_TotalCh,
                    short iDO_TotalCh,
                    short iLatchType,
                    short iCheckSum,
                    short iTimeOut,
                    unsigned long *lDI_Latch_Value,
                    unsigned long *lDO_Latch_Value,
                    char *cDI_Latch_BitValue,
                    char *cDO_Latch_BitValue)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iDI_TotalCh:	The total channel of DI module.
iDO_TotalCh:	The total channel of DO module.
iLatchType:	0: low latch mode 1: high latch mode
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms
lDI_Latch_Value:	Read Digital input latch
lDO_Latch_Value:	Read Digital output latch
cDI_Latch_BitValue:	Read DI state of single channel
cDO_Latch_BitValue:	Read DO state of single channel

4.5. DCON_Clear_DIO_Latch

Description:

The function can clear the latch status of DI, DO, DIO module when latch function has been enabled.

Syntax:

```
DCON_Clear_DIO_Latch(unsigned char cComPort,
                     short iAddress,
                     short iSlot,
                     short iCheckSum,
                     short iTimeOut)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms

4.6. DCON_Read_DI_Counter

Description:

Obtain the counter event value of the channel number of Digital input module.

Syntax:

```
DCON_Read_DI_Counter(unsigned char cComPort,
                     short iAddress,
                     short iSlot,
                     short iChannel,
                     short iDI_TotalCh,
                     short iCheckSum,
                     short iTimeOut,
                     unsigned long *lCounter_Value)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The digital input Channel No.
iDI_TotalCh:	The total channel of DI module.
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms
lCounter_Value:	Counter value

4.7. DCON_Clear_DI_Counter

Description:

Clear the counter value of the channel number of Digital input module.

Syntax:

```
DCON_Clear_DI_Counter(unsigned char cComPort,
                      short iAddress,
                      short iSlot,
                      short iChannel,
                      short iDI_TotalCh,
                      short iCheckSum,
                      short iTimeOut)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The digital input Channel No.
iDI_TotalCh:	The total channel of DI module.
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms

4.8. DCON_Write_AO

Description:

Output the analog value from analog output module.

Syntax:

```
DCON_Write_AO(unsigned char cComPort,
              short iAddress,
              short iSlot,
              short iChannel,
              short iAO_TotalCh,
              float fValue,
              short iCheckSum,
              short iTimeOut)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The analog output Channel No.
iAO_TotalCh:	The total channel of AO module.
fValue:	Write analog output value
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms

4.9. DCON_Read_AO

Description:

Read the analog output value of analog output modules.

Syntax:

```
DCON_Read_AO(unsigned char cComPort,
             short iAddress,
             short iSlot,
             short iChannel,
             short iAO_TotalCh,
             short iCheckSum,
             short iTimeOut,
             float *fValue)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The analog output Channel No.
iAO_TotalCh:	The total channel of AO module.
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms
fValue:	Read analog output value

4.10. DCON_Read_AI

Description:

Obtain the analog input value in float format.

Syntax:

```
DCON_Read_AI(unsigned char cComPort,
             short iAddress,
             short iSlot,
             short iChannel,
             short iAI_TotalCh,
             short iCheckSum,
             short iTimeOut,
             short iDataFormat,
             float *fValue,
             short *iValue)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The analog input Channel No.
iAI_TotalCh:	The total channel of AI module.
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms
fValue:	Read analog input value for float format.
iValue:	Read analog input value for hex format.

4.11. DCON_Read_Counter

Description:

Obtain the value of the selected counter/frequence for the counter/freuence module.

Syntax:

```
DCON_Clear_Counter(unsigned char cComPort,
                    short iAddress,
                    short iSlot,
                    short iChannel,
                    short iCheckSum,
                    short iTimeOut)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The Counter/frequence Channel No.
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms
lCounter_Value:	Read counter / frequnece value.

4.12. DCON_Clear_Counter

Description:

Obtain the value of the selected counter/frequence for the counter/freuence module.

Syntax:

```
DCON_Clear_Counter(unsigned char cComPort,
                   short iAddress,
                   short iSlot,
                   short iChannel,
                   short iCheckSum,
                   short iTimeOut)
```

Return Value:

NoError:	OK
Others:	Error code

Input Parameter:

cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The Counter/frequence Channel No.
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms

4.13. DCON_Read_Overflow

Description:

Read specified channel's Counter overflow value.

Syntax:

```
DCON_Read_Overflow(unsigned char cComPort,
                   short iAddress,
                   short iSlot,
                   short iChannel,
                   short iCheckSum,
                   short iTimeOut,
                   short iDataFormat,
                   unsigned int *iOverflow)
```

Return Value:

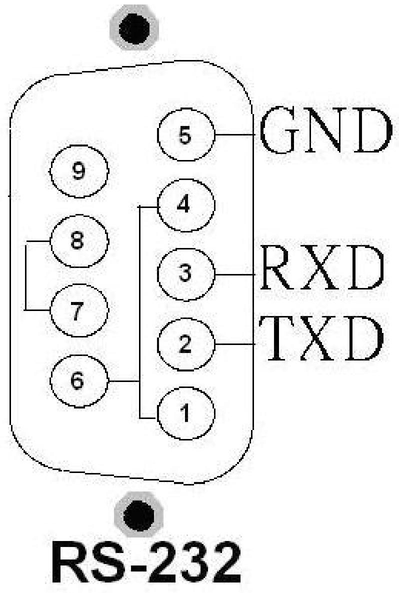
NoError:	OK
Others:	Error code

Input Parameter:

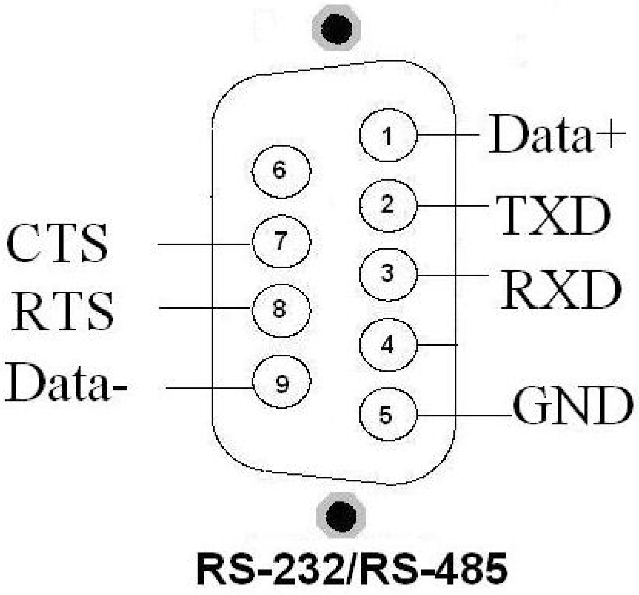
cComPort:	COM port number, 0 to 255
iAddress:	Module address, from 0 to 255
iSlot:	Slot number, 0 to 7 or –1 (for module of RS-485)
iChannel:	The Counter/frequence Channel No.
iCheckSum:	0: Disable or 1: Enable
iTimeout:	Time out setting, normal=100, unit: ms
iDataFormat:	0: engineer 1: s's comp
iOverflow:	Overflow value

Appendix A

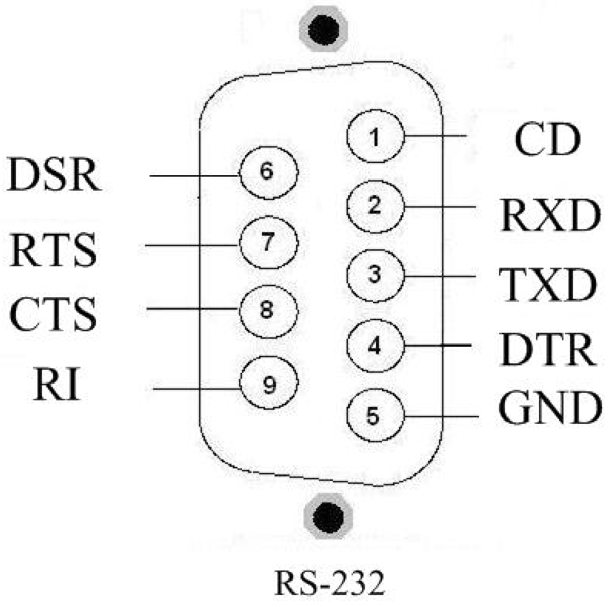
The pin assignment of I-8000:



The COM1 Pin assignment



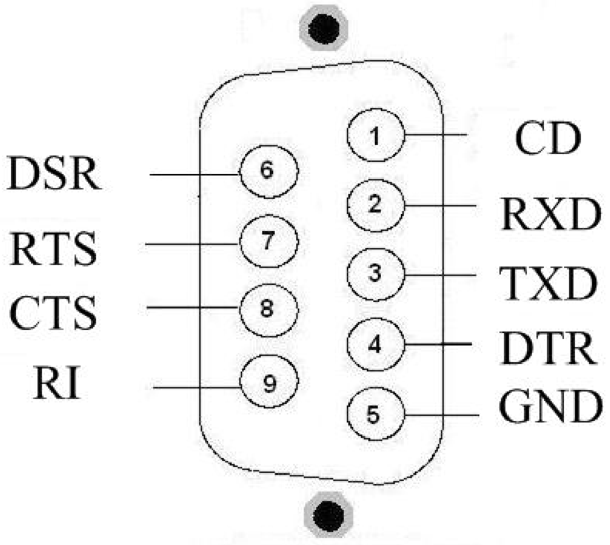
The COM3 Pin assignment



The COM4 Pin assignment

Appendix B

The pin assignment of wincon:



RS-232

The COM2 Pin assignment