

Quick Start Manual

For ICPDAS DCON (I-7000/8000/87K) series modules Series [Version 1.0]

Warranty

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

Warning

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

Copyright

Copyright 1997-2003 by ICPDAS Inc and all rights is reserved.

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.

1 DCON_DLL (Optional)

Please contact your local field agent to obtain detailed information about the DCON_DLL software package.

The DCON_DLL is a Windows DLLs designed for the Windows 95/98/NT/2000/XP user, and can be accessed with by VC++, BC++, VB, Delphi, BC++ Builder. The features of DCON_DLL are as follows:

1. **Multi-task DLLs design**
2. **Provides general-purpose RS-232 application functions**
3. **Provides general-purpose DCON series module send/receive functions**
4. **Provides high performance DCON series module application functions**
5. **Multi-speed demos**
6. **Multi-data-format demos**
7. **Many demo programs show complete with your source code**

In this section, we will show you how to write your first program with DCON_DLL using Visual Basic 5.0. For the Delphi, BCB and VC++ user, please refer to the demo programs folder of Delphi, BCB and VC++ respectively.

After you install the DCON_DLL software package, the I7000.DLL and UART.DLL files will be copied into your system folder as follows:

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

C:\WinNT\System32 (Windows NT/2000/XP)

To use the I7000.DLL and UART.DLL functions, we must declare the function's prototype (interface) for your development tools. We provide the complete I7000.DLL and UART.DLL for VC++ declaration files, VB, Delphi and BCB. You can copy these declaration files into your project folder and include them in your project, allowing you to use the functions in the same way as the

those provided by your development tools.

The following procedure will allow you to write your first program:

1.1 Your first program with DCON_DLL

To begin writing your first program, follow the steps below

1. Create a folder in which to store your project, for example: E:\Test\Start1
2. Copy the declaration files into your new project folder, Refer to Figure 1-1-1.
3. Open Visual Basic 5.0, and select the “Standard EXE” icon, then click the “OK” button. Refer to Figure 1-1-2.

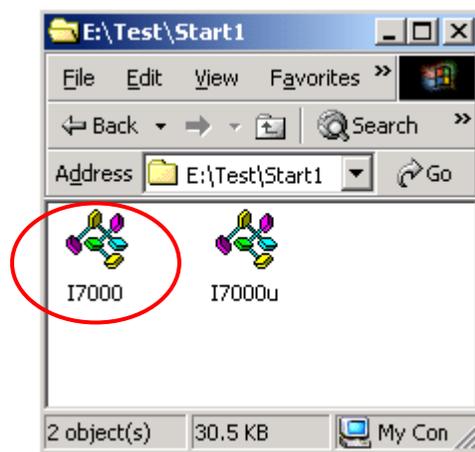


Figure 1-1-1. Visual Basic declaration files.

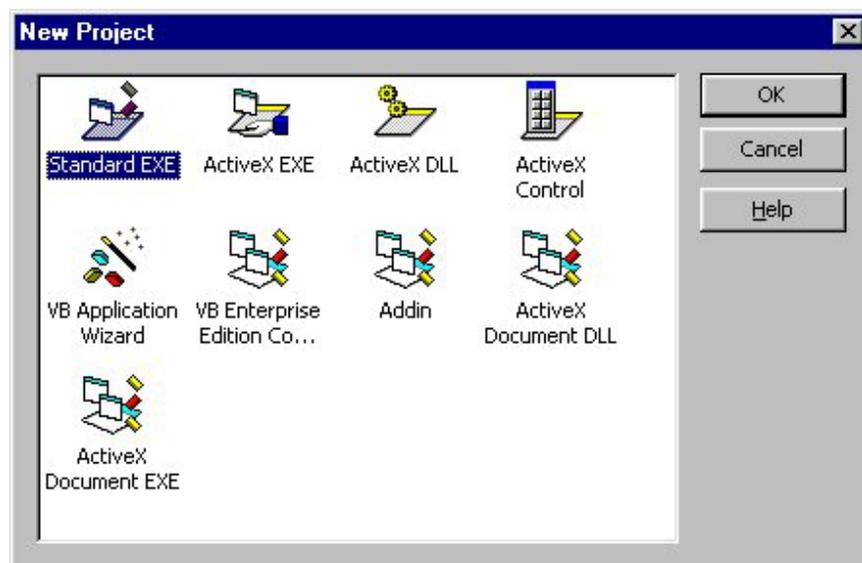


Figure 1-1-2. Select the “Standard EXE”

4. Save your project into the folder that you previously created. Refer to Figures 1-1-3 to 1-1-5.

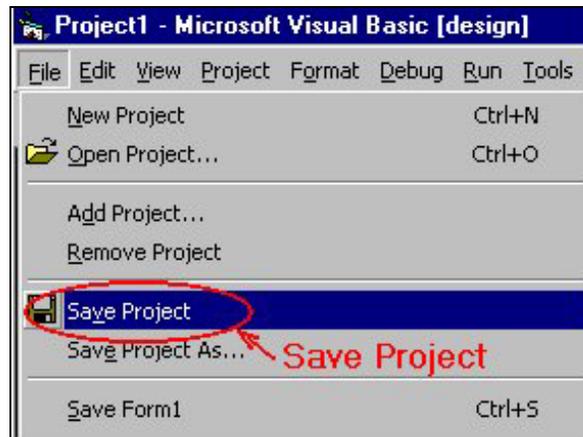


Figure 1-1-3. Save Project

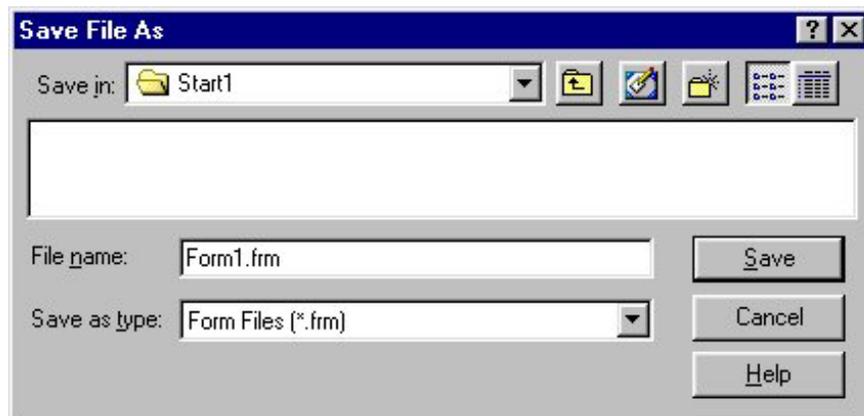


Figure 1-1-4. Save form1

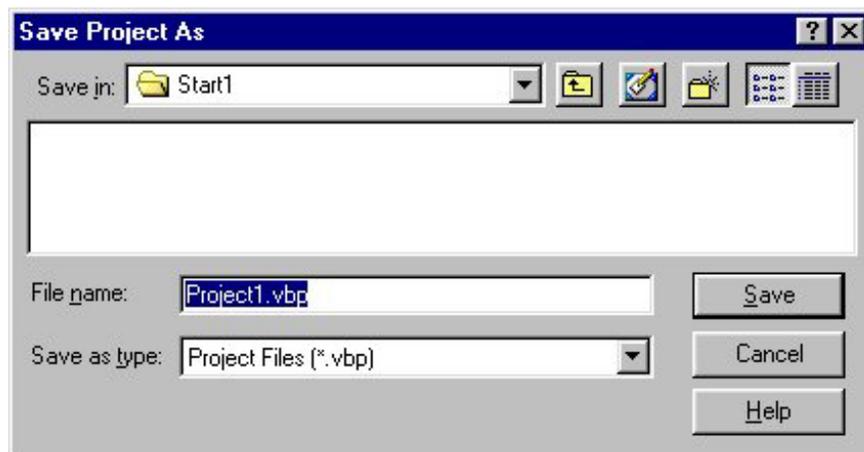


Figure 1-1-5. Save Project1.

5. Click the right mouse button in the Project manager window, and select “Add” then select ”Module” , Refer to Figure 1-1-6.
6. After the “Add Module” window appears, select the “Existing” tab and locate your project folder. Select the declaration files required and then click on the “open” button to add them to your project. Refer to Figure 1-1-7.

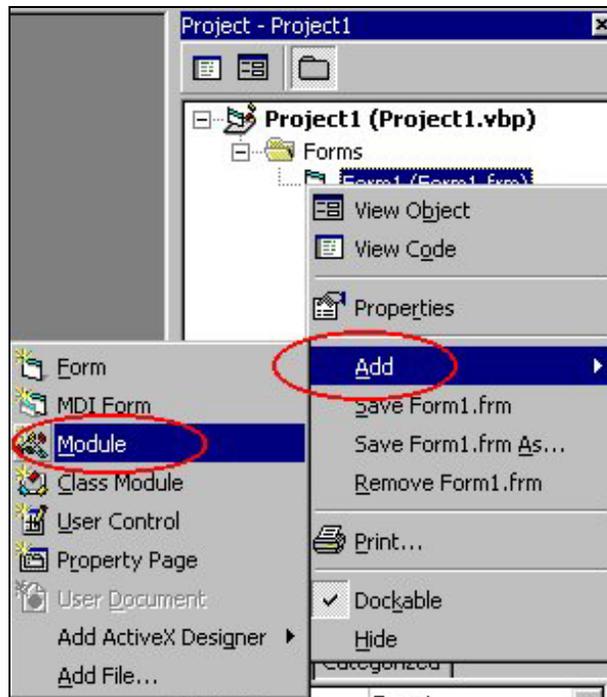


Figure 1-1-6. Add Module

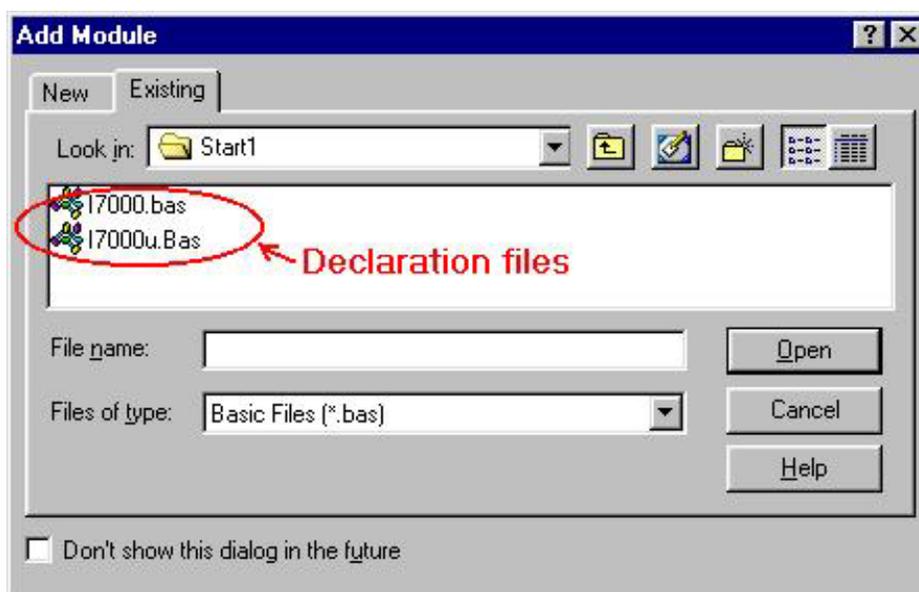


Figure 1-1-7. Add the declaration files

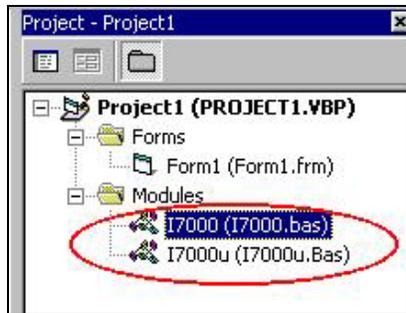


Figure 1-1-8. The declaration files have been added in the project.

7. Add a “Command1” button to your form1 then enter the following code in the a sub routine called “Command1_Click() “ :

```
Private Sub Command1_Click()
    Dim wt As Integer
    Port = 2
    Open_Com Port, 9600, 8, 0, 0
    SendTo7000 = "$01M"
    ReceiveFrom7000 = Space(100)
    Send_Receive_Cmd Port, SendTo7000, ReceiveFrom7000, 100, 0, wt
    Command1.Caption = ReceiveFrom7000
    Close_Com Port
End Sub
```

8. Run the program and click the “Command1” button. The caption of the button will change to “!01xxxx”. The “!” means “OK”. The “01” refers to the address at 0x01”. The “xxxx” is the module ID. Refer to Figure 1-1-9.



Figure 1-1-9. The module return “!01xxxx”.

9. The code is described as follows :

```
Dim wt As Integer
```

'Declare a variable wt, where the data type is an Integer.

```
Port = 2
```

'Specify the COM port that you are using.

```
Open_Com( Port, 9600, 8, 0, 0)
```

'Call the "Open_Com" function

'the parameters are the COM Port, Baubrate =9600,

'DataBits =8, ParityBit= 0(None parity bit), StopBits 0(One stop bit).

```
SendTo7000 = "$01M"
```

'Use the "\$01M" command to obtain the module name at address

'0x01.

```
ReceiveFrom7000 = Space(100)
```

'Declare a buffer to store the return value string.

```
Send_Receive_Cmd (Port, SendTo7000, ReceiveFrom7000, 100, 0, wt)
```

'Call the "Send_Receive_Cmd" function

'Parameters are Command String, Return String,

'Timeout value =100ms(0.1 seconds), Checksum= 0 (disable)

The wt variable stores the time that the function can be used.

```
Command1.Caption = ReceiveFrom7000
```

'Use the caption of "Command1" button to show the return value

```
Close_Com Port( )
```

'Close the COM port

'The Parameter is the COM port that you previously opened.

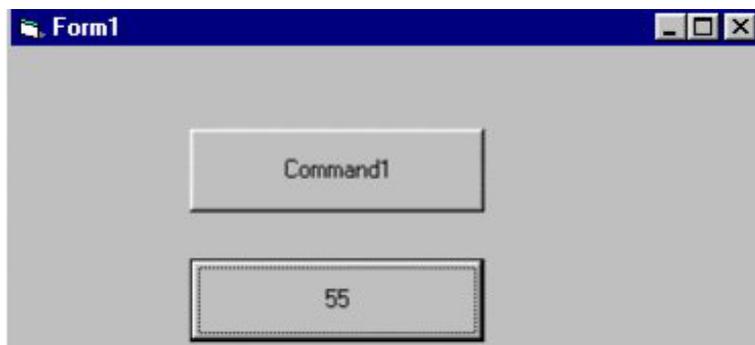


Figure 1-1-10. Use the "AnalogIn" function to obtain the analog input value

10. Add a second CommandButton to your form1, and enter the following code in a sub routine called "Command2_Click()" :

```
Private Sub Command2_Click()  
    Port = 2                                'Use COM2  
    Open_Com (Port, 9600, 8, 0, 0)  
    'Open COM Port with Baud Rate 9600, 8 DataBits,  
    'None Paritybits , One stop bit  
  
    SendTo7000 = Space(100)                'Declare a buffer to store the 'send  
    'string value  
    ReceiveFrom7000 = Space(100)          'Declare a buffer to return 'string '  
    'value  
    w7000(0) = Port                        'The COM port that you are 'using  
    w7000(1) = 1                            'The module Address at 0x01  
    w7000(2) = &H7013                       'Module ID 0x7013  
    w7000(3) = 0                            'Checksum 0 (disable)  
    w7000(4) = 100                          'Timeout value 0.1 seconds  
  
    AnalogIn( w7000(0), f7000(0), SendTo7000, ReceiveFrom7000)  
    'Call the "AnalogIn" function to obtain the value from the module  
    'The Parameters are w7000(0), f7000(0), SendTo7000,  
    'ReceiveFrom7000  
  
    Command2.Caption = f7000(0)  
    'Use the caption of the "Command2" button to show the analog value  
  
    Close_Com Port  
    'Close the COM port that you previously opened  
End Sub
```

11. Run the program and click the "Command2" button. The caption of the button will show the analog input value from the module. Refer to Figure 1-1-10.

For simplicity in this first program, we have not processed the return value of every function. In a normal project, you must obtain the return value and process it. For a detailed description of each function, please refer to the DCON_DLL user manual, which is provided in .PDF format in the DCON_DLL software package.

The software package also contains a variety of demo programs written for

VC++,VB,Delphi and BCB, including the complete source code.

2 Problems Report

Technical support is available at no charge. The best way to report problems is send electronic mail to

service@icpdas.com

When reporting problems, please include the following information:

1. Is the problem **reproducible**? If so, how?
2. What kind and version of **Platform** are you using? For example, Windows 3.1, Windows for Workgroups, Windows NT 4.0, etc.
3. What kinds of our **products** are you using? Please see the product's manual.
4. If a dialog box with an **error message** was displayed, please include the full text of the dialog box, including the text in the title bar.
5. If the problem involves **other programs** or **hardware devices**, what devices or version of the failing programs do you use?
6. Other **comments** relative to this problem or any **suggestions** will be welcomed.

After we had received your comments, we will take about two business days to test the problems that you have reported. And then We will reply it as soon as possible to you. Please check that we had received your comments? And please keep in contact with us.

E-mail: service@icpdas.com
icpdas@ms8.hinet.net

Web site: <http://www.icpdas.com.tw>