

Interface of Direct IO DLL to Microsoft® Visual Basic 6.0

These are the programming guide lines required to interface the Module in Basic as follows:

1. **Directory Setup** : How to copy the files.?

The files required for a project to use the DirectIO functionalities are provided in the "SetupPacks\VBApp" folder. These files need to be copied to the project directory and some minor settings have to be done to include the DirectIO functionality in the project. The following are the files and corresponding target copy locations:

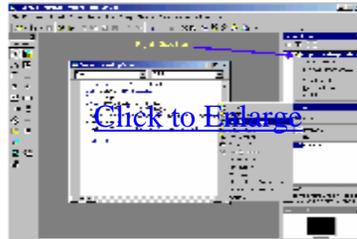
\SetupPacks\VBApp\PortAcc.bas	\{your project folder}\PortAcc.bas
\SetupPacks\VBApp\Directio.dll	\{your project folder}\Directio.dll
\SetupPacks\VBApp\Porttalk.sys	\{your project folder}\Porttalk.sys

NOTE: wherever the project's exe is placed the last two files Directio.dll and porttalk.sys must also be placed for the proper functioning of the program.

2. **Compiler Setup** : How to configure the compiler and build settings?

This is required in order that the particular DLL functionalities are inherited by the program during compile time. Here in case of DirectIO DLL the main module file portacc.bas is require to be instantiated. The "PortAcc.bas" file must be included in the project where the IO functions are being used. To compile the usability of the DLL these are the build settings :

Step 1.



Step2.



After all this settings changes are done the project is ready to compile and run . This setup is already done in the VB Sample App . You can compile and see it.

Functions Provided and Descriptions :

1. Declare Sub dap Lib "directio" ()
This function is the test function to display a message indicating that the DLL is working correctly and to check for port access to printer port.
Ex: Call dap
2. Declare Sub uninstalldriver Lib "directio" ()
This function is used to uninstall the Kernel Mode Driver used for IO in case of NT/2000/XP.
Ex: Call uninstalldriver
3. Declare Sub outp Lib "directio" Alias "outportb" (ByVal Addr As Integer, ByVal data As Byte)
This is the main function used to 'send data to a port'. It has two parameters :
Addr specifying the address of the port to which the data is to be sent. The parameter ranges are:
Addr: &H0000 [or 0x0000 or 0000H] to
&H0FFF [or 0x0FFF or 0FFFH] in Hexadecimal.
{12 bit Parameter}
data: &H000 [or 0x000 or 000H] to
&H0FF [or 0x0FF or 0FFH] in Hexadecimal.
{ 8 bit Parameter}
Ex: Call outp(&H0378, &H0AA)
4. Declare Function inp Lib "directio" Alias "inportb"
(ByVal Addr As Integer) As Byte
This is the main function used to 'read data from a port'. It has only one parameter as the Address of the port to be read. It returns the byte read from the port. The Parameter range is :
Addr: &H0000 [or 0x0000 or 0000H] to
&H0FFF [or 0x0FFF or 0FFFH] in Hexadecimal.
{12 bit Parameter}
The return value is :
byte: &H000 [or 0x000 or 000H] to
&H0FF [or 0x0FF or 0FFH] in Hexadecimal.
{ 8 bit Value}
Ex: test = inp(&H0378)
5. Declare Function andb Lib "directio" (ByVal a As Byte, ByVal b As Byte) As Byte
This function And two bytes (a & b) and return result.
Ex: c = andb(a, b)
Use: c = andb(&H35, &H12)
Then c = &H10
&H35 - 0011 0101
(AND) &H12 - 0001 0010
&H10 - 0001 0000

6. Declare Function orb Lib "directio" (ByVal a As Byte, ByVal b As Byte) As Byte
 This Function ORs two bytes (a | b) and return result.
 Ex: c = orb(a, b)
 Use: c = orb(&H35 , &H12)
 Then c = &H37
 &H35 - 0011 0101
 (OR) &H12 - 0001 0010
 &H37 - 0011 0111

7. Declare Function notb Lib "directio" (ByVal a As Byte) As Byte
 This function Not of a byte (~a) and return result1.
 Ex: c = notb(a)
 Use: c = notb(&H35)
 Then c = &HCA
 (NOT) &H35 - 0011 0101
 &HCA - 1100 1010

8. Declare Function shrb Lib "directio" (ByVal a As Byte, ByVal bit As Byte) As Byte
 This function Right Shift a byte by n no of bits (a>>bit) and return the result.
 Ex: c = shrb(a, 1)
 Use: c = shrb(&H35 , 3)
 Then c = &H06
 (SHR-3) &H35 - 0011 0101
 &H06 - 0000 0110

9. Declare Function shlb Lib "directio" (ByVal a As Byte, ByVal bit As Byte) As Byte
 This Function Left Shift a byte by n no of bits (a<<bit) and return the result.
 Ex: c = shlb(a, 1)
 Use: c = shlb(&H35 , 3)
 Then c = &HA8
 (SHL-3) &H35 - 0011 0101
 &HA8 - 1010 1000

10. Declare Function andw Lib "directio" (ByVal a As Integer, ByVal b As Integer) As Integer
 This function is used to And two words (a & b) and return result.
 Ex: c = andw(a, b)
 Use: c = andw(&HAF35 , &HD212)
 Then c = &H8210
 &HAF35 - 1010 1111 0011 0101
 (AND) &HD212 - 1101 0010 0001 0010
 &H8210 - 1000 0010 0001 0000

11. Declare Function orw Lib "directio" (ByVal a As Integer, ByVal b As Integer) As Integer
 This function is used to OR two words (a | b) and return result.
 Ex: c = orw(a, b)
 Use: c = orw(&HAF35 , &HD212)
 Then c = &HFF27

&HAF35 -	1010 1111 0011 0101
(OR) &HD212 -	1101 0010 0001 0010
&HFF27 -	1111 1111 0011 0111

12. Declare Function notw Lib "directio" (ByVal a As Integer) As Integer
 This function is used to Not of a word (~a) and return result.
 Ex: c = notw(a)
 Use: c = notw(&HAF35)
 Then c = &H50CA

(NOT) &HAF35 -	1010 1111 0011 0101
&H50CA -	0101 0000 1100 1010

13. Declare Function shrw Lib "directio" (ByVal a As Integer, ByVal bit As Integer) As Integer
 This function Right Shifts a word by n no of bits (a>>bit) and return result.
 Ex: c = shrw(a, 1)
 Use: c = shrw(&HAF35 , 3)
 Then c = &H15E6

(SHR-3) &HAF35 -	1010 1111 0011 0101
&H15E6 -	0001 0101 1110 0110

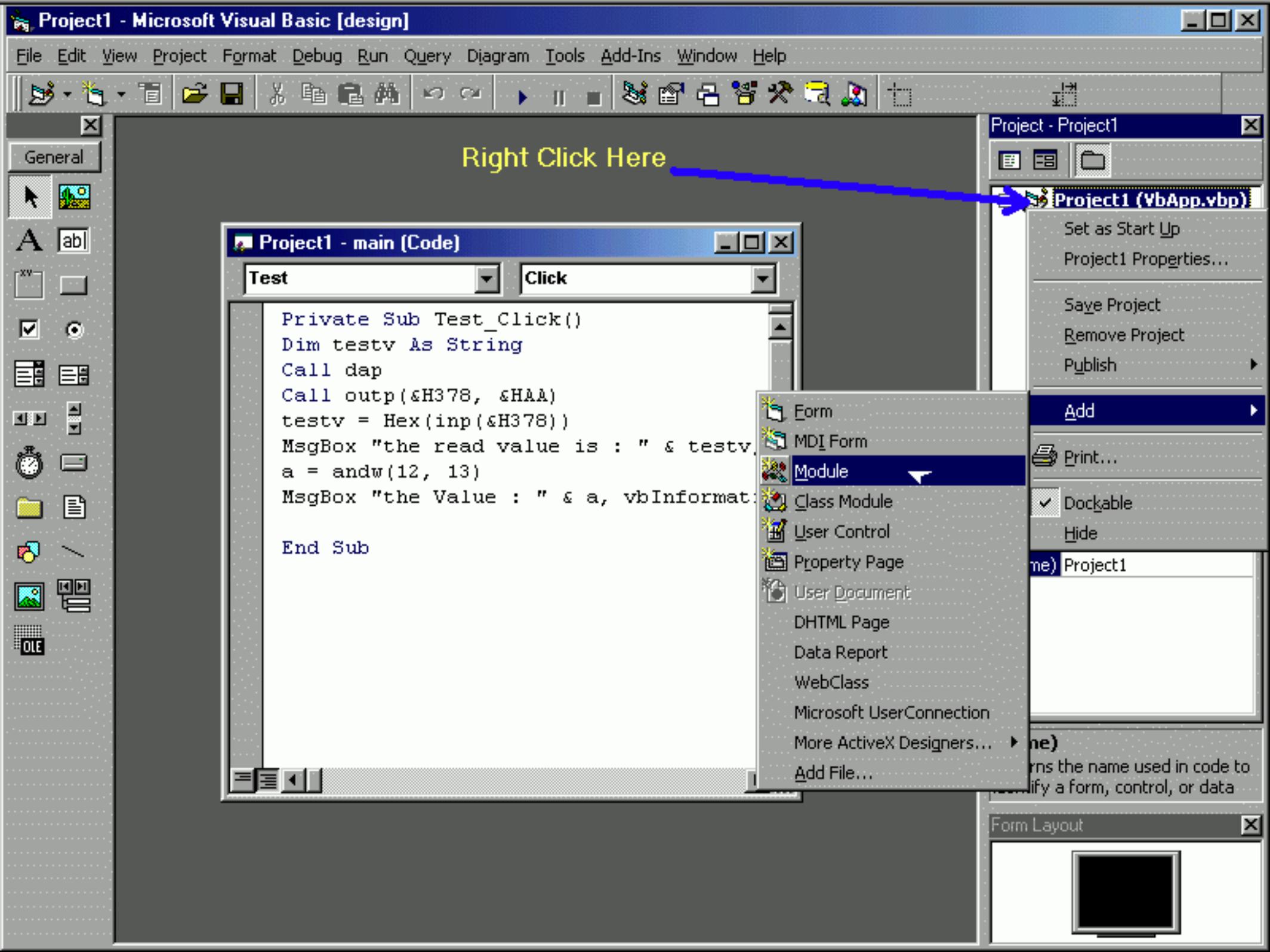
14. Declare Function shlw Lib "directio" (ByVal a As Integer, ByVal bit As Integer) As Integer
 This function Left Shifts a word by n no of bits (a<<bit) and return the result.
 Ex: c = shlw(a, 1)
 Use: c = shlw(&HAF35 , 3)
 Then c = &H79A8

(SHL-3) &HAF35 -	1010 1111 0011 0101
&H79A8 -	0111 1001 1010 1000

15. Declare Function bsb Lib "directio" (ByVal a As Byte, ByVal bit As Byte) As Byte
 This Function Sets a Particular bit in a byte and return result.
 Ex: c = bsb(a , 0)
 Use: c = bsb(&H35 , 1)
 Then c = &H37

(Set Bit 1{1 left from LSB})	&H35 -	0011 0101	bits - 7 6 5 4 3 2 1 0
	&H37 -	0011 0111	

20. Declare Function bsw Lib "directio" (ByVal a As Integer, ByVal bit As Integer) As Integer
This function Sets a Particular bit in a word and returns result, same as the bsb function. except the parameter is 16 bit.
Ex: `c = bsw(a,0)`
21. Declare Function bcw Lib "directio" (ByVal a As Integer, ByVal bit As Integer) As Integer
This function Clears a Particular bit in a word and returns result, same as the bcb function except the parameter is 16 bit.
Ex: `c = bcw(a,0)`
22. Declare Function cpbw Lib "directio" (ByVal dest As Integer, ByVal src As Integer, ByVal bit As Integer) As Integer
This function Copy a Particular bit from Source word to Destination word and return destination word, same as the cpbb function except the parameters are 16 bit.
Ex: `dest = cpbw(dest, src, 1)`
23. Declare Function getbw Lib "directio" (ByVal a As Integer, ByVal bit As Integer) As Integer
This function Gets a Particular bit from a word in the LSB of the retrned result, same as the getbb function except the parameter is 16 bit.
Ex: `bitchk = getbw(a,0)`
24. Declare Function compbw Lib "directio" (ByVal a As Integer, ByVal bit As Integer) As Integer
This function Complements a Particular bit in a word and returns the result same as the compbb function except the parameters are 16 bit.
Ex: `compa = compbw(a,0)`



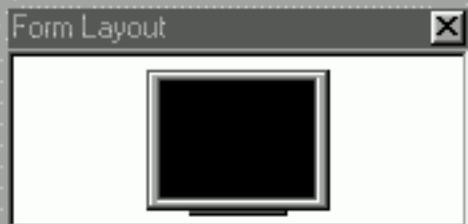
Right Click Here

```
Project1 - main (Code)  
Test Click  
Private Sub Test_Click()  
Dim testv As String  
Call dap  
Call outp(&H378, &HAA)  
testv = Hex(inp(&H378))  
MsgBox "the read value is : " & testv  
a = andw(12, 13)  
MsgBox "the Value : " & a, vbInformation  
  
End Sub
```

- Form
- MDI Form
- Module
- Class Module
- User Control
- Property Page
- User Document
- DHTML Page
- Data Report
- WebClass
- Microsoft UserConnection
- More ActiveX Designers...
- Add File...

Project - Project1

- Project1 (VbApp.vbp)
- Set as Start Up
- Project1 Properties...
- Save Project
- Remove Project
- Publish
- Add
- Print...
- Dockable
- Hide
- Project1



Add Module



New Existing

Select this TAB

Look in: VbApp



PortAcc.bas

Select the File

Click to Continue

File name: PortAcc.bas

Open

Files of type: Basic Files (*.bas)

Cancel

Help

Don't show this dialog in the future