Ethernet IO

Programming Manual

Visual Basic 6

Design Gateway Co., Ltd.

Rev 1.0

(PD0401-6-02C-01E)

*** Please read this manual carefully before using Ethernet IO ***

Reversion History

reversion	Date	Detail of change
1.0	26 August 2005	Initial Release

Table of Contents

1. How to Setup

1.1. Minimum Requirement

1 Personal Computer

Operating System	Windows XP, Windows 2000
Space requirement	10 Mbytes
CPU	300 MHz or higher Pentium PC or compatible PC
LAN interface card	10/100 Mbit.
Development program	Microsoft Visual Basic 6.0

2 Ethernet IO Board

Board version	1.0
Firmware Version	2.0
DLL version	1.2
Module	1.0

2. Contents in Software Directory

- CFG_APP_VB Configuration application and source code
- Library_VB Software library (DLL)

2.1. Description of Library Files

File	Detail
EtherIODll.dll	DLL file (use when executing)
EthernetIO.bas	Library file module (use when compile)

Ethernet IO development kit provides “Regular” DLL library. User can use this library with Microsoft Visual Basic (version 6 recommended). For now it does not support Delphi etc…

2.2. Communicate with Ethernet IO Board

Ethernet IO's firmware is service network (LAN) communication and serial (RS232 or RS485). In network communication service, Ethernet IO uses TCP/IP port **4025** for serial communication and map uses **4026** for "Command service". User can communicate directly to serial port on Ethernet IO without this DLL. In case of using IO command, user must use this DLL.

3. Example using DLL Library 0f Ethernet IO

This is topic description how to use DLL in user's application, just like source codes in configuration program do. However there are no deep details on coding. User should have some skill in Microsoft Visual Basic programming.

Now we are creating Example Application Program. this application is able to polling receive serial data by use timer and reading logic value GPIO PORT C (8 bit) of Ethernet IO board. Please note that all **bold-source codes** in this topic 3.2.1-3.2.7 are added codes to procedure each ID Components and important codes.

3.1. Using Library Function of "EthernetIO.Dll"

3.1.1) Open Visual Basic program and create Standard EXE Application project.

3.1.2) Copy "EtherIODll.dll" and "EthernetIO.bas" file to project directory. This file "EthernetIO.bas" contains all Library Functions of EtherIODll.dll.

3.1.3) Select to Menu project->Add Module

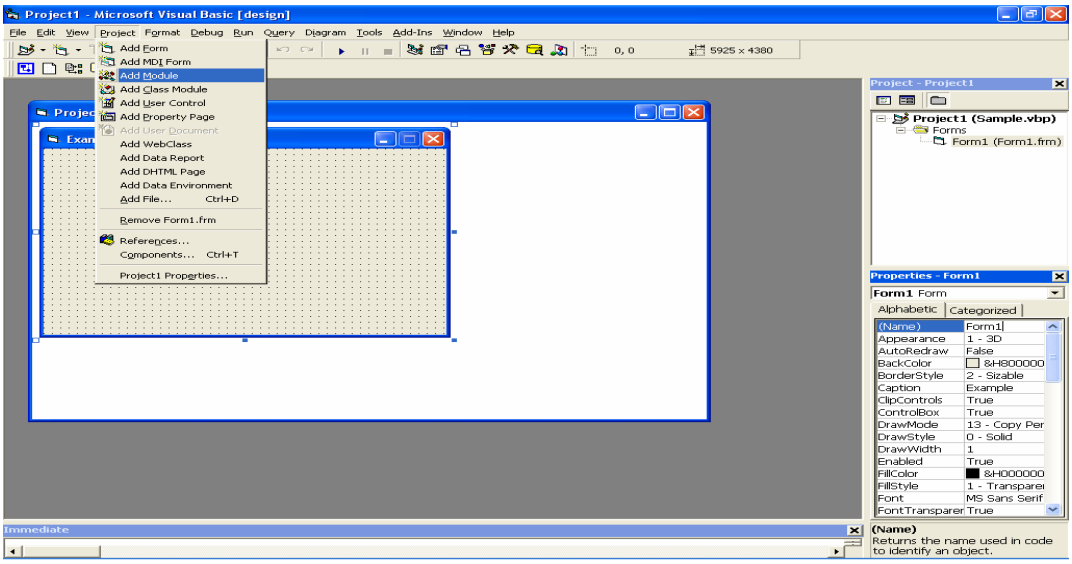

Figure 3-1 Show Adding Module Menu

3.1.4) Select to “Existing” tab and select file EthernetIO.bas from directory of project.

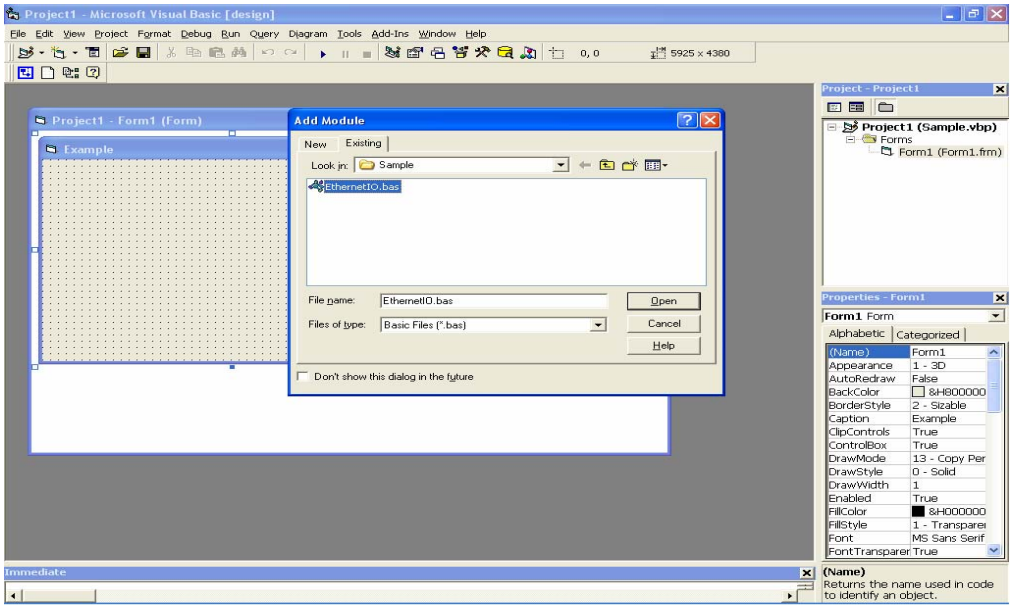

Figure 3-2 Adding File Module

After adding file module, module directory automatic created in project Explorer Windows of project. These directory modules contain module form of EthernetIO.bas. In module form contain all Prototype Functions of EtherIODll.dll.

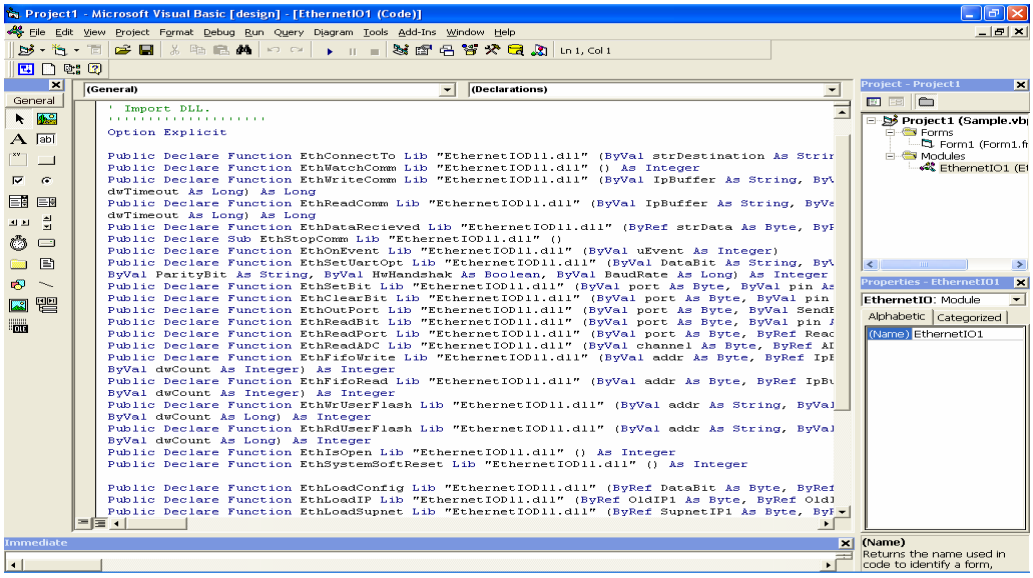

Figure 3-3 Show Detail Prototype Functions

3.2. Create Example application form

Now we create Example application form and setting value of ID Component. Example application form show as picture below and adding Timing Control.

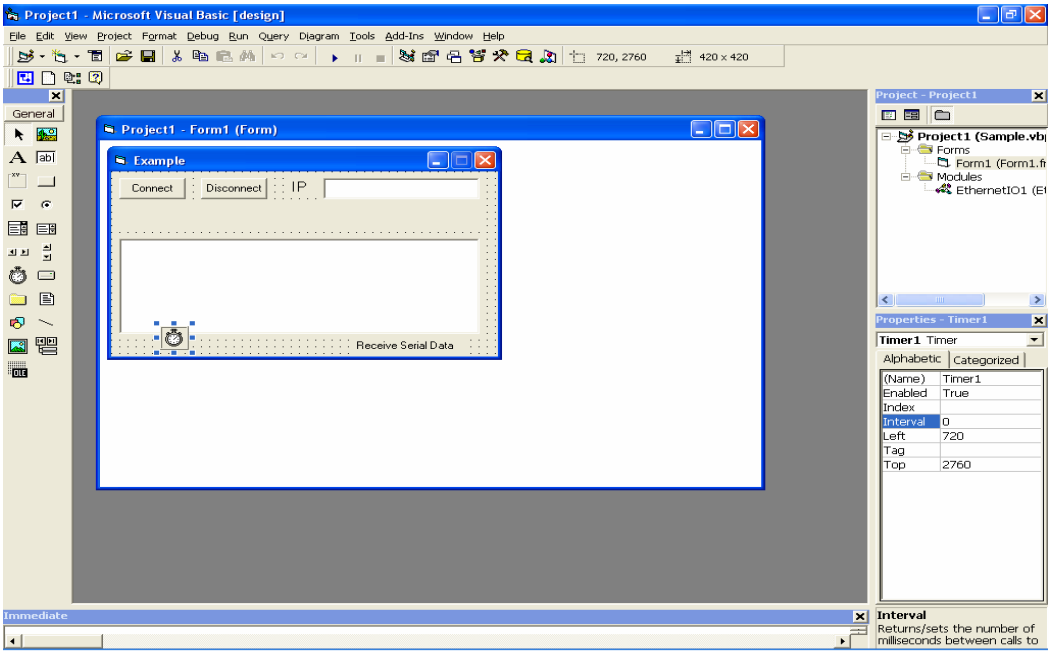

Figure 3-4 Example Application Form

Table 3-1 Show Detail of Setting ID Component

ID Component	Properties	value	EVEN	Description
Command button	NAME	cmdConnect	Click Button	Use to connect to Ethernet IO Board
	Caption	Connect		
Command button	NAME	cmdDisconnect	Click Button	Use to disconnect Ethernet IO Board
	Caption	Disconnect		
Command button	NAME	cmdRDPortC	Click Button	Use to read Logic Value Port C
	Caption	RD_PORTC		
Text Box	NAME	txtIP	None	input IP Address Ethernet IO Board
	Text	None		
Text Box	NAME	txtReceivedData	None	Display Receive Serial Data Message
	Text	None		
Text Box	NAME	txtRDPort	None	Display Logic Value Port C
	Text	None		
Label	NAME	lblStatus	None	Show Status of Communication
Label	NAME	Label1	None	Show Text IP
	Caption	IP		
Label	NAME	Label2	None	Show Text Receive Serial Data
	Caption	Receive Serial Data		
Timer Control	NAME	tmrCtl	Timer	Timer interrupt for do command
	Enable	True		
	Interval	5		

3.2.1) Declare variable and constant value.

```
Const STATUS_CONN_FAILED As String = "Connection failed"
Const STATUS_DISCONN_MSG As String = "Disconnected"
Const STATUS_READY_MSG   As String = "Read ....."
Const DEFAULT_IP As String = "192.168.11.241"
Const PORT_A As Byte = &H20
Const PORT_B As Byte = &H24
Const PORT_C As Byte = &H28
Const PORT_D As Byte = &H2C
Const PORT_E As Byte = &H30
Const PORT_F As Byte = &H34
Const PORT_G As Byte = &H36
Public strData As String
```

3.2.2) Adding function Check_IP for check IP Address of Ethernet IO board.

```
''''''''''''''''''''''''''''''''''''''''

' Summary : Check IP Addess value.

''''''''''''''''''''''''''''''''''''''''

Private Function Check_IP(ByVal strIPServer As String, ByRef nStatus As Integer)
    Dim strIPChk As String
    Dim strIP() As String
    Dim i As Integer, j As Integer
    Dim strIPAddr As String, strIPNo As String
    Dim nIPAddr As Integer
    Dim nUp As Integer
    Dim nLen As Integer, nAsc As Integer
    strIPChk = strIPServer
    strIP = Split(strIPChk, ".", -1, vbTextCompare)
    nUp = UBound(strIP)
    If (nUp <> 3) Then
```

```
        nStatus = -1
        GoTo End5
    Else
        For i = 0 To 3
            strIPAddr = strIP(i)
            nLen = Len(strIPAddr)
            If (nLen = 1) Then
                nAsc = Asc(strIPAddr)
                If (((nAsc >= 48) And (nAsc <= 57)) = False) Then
                    nStatus = -1
                    Exit For
                End If
            ElseIf (nLen = 2) Then
                strIPNo = Left(strIPAddr, 1)
                nAsc = Asc(strIPNo)
                If (((nAsc >= 48) And (nAsc <= 57)) = False) Then
                    nStatus = -1
                    Exit For
                End If
                strIPNo = Right(strIPAddr, 1)
                nAsc = Asc(strIPNo)
                If (((nAsc >= 48) And (nAsc <= 57)) = False) Then
                    nStatus = -1
                    Exit For
                End If
            ElseIf (nLen = 3) Then
                strIPNo = Left(strIPAddr, 1)
                nAsc = Asc(strIPNo)
                If (((nAsc >= 48) And (nAsc <= 57)) = False) Then
                    nStatus = -1
```

```
                Exit For
            End If
            strIPNo = Mid(strIPAddr, 2, 1)
            nAsc = Asc(strIPNo)
            If (((nAsc >= 48) And (nAsc <= 57)) = False) Then
                nStatus = -1
                Exit For
            End If
            strIPNo = Right(strIPAddr, 1)
            nAsc = Asc(strIPNo)
            If (((nAsc >= 48) And (nAsc <= 57)) = False) Then
                nStatus = -1
                Exit For
            End If
        Else
            nStatus = -1
            Exit For
        End If
        nIPAddr = Val(strIPAddr)
        If (((nIPAddr >= 0) And (nIPAddr <= 255)) = False) Then
            nStatus = -1
            Exit For
        End If
        nStatus = 1
    Next i%
End5: End If
End Function
```

3.2.3) Double click on Example application form to add “Form_Load()” procedure. This function will load IP Address, default value and setup button default. This

function load default value of Example application form and ID components when open form.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Summary : Load IP Addess Default value when start up program
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub Form_Load()
 txtIP = DEFAULT_IP
 lblStatus = STATUS_READY_MSG
 cmdDisConnect.Enabled = False
 cmdRDPortC.Enabled = False
End Sub
```

3.2.4) Double click on "Connect" button to add cmdConnect_Click() procedure. This procedure use to connection to Ethernet IO board.

```
 ''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Summary : Connecting to Ethernet IO Board
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub cmdConnect_Click()
    Dim IPServerStr As String
    Dim nConnect As Integer
    Dim nWatchComm As Integer
    Dim nStatus As Integer
    lblStatus = STATUS_CONNECTING_MSG
    IPServerStr = txtIP.Text
    Call Check_IP(IPServerStr, nStatus)
    If (nStatus <> 1) Then
        MsgBox "Invalid IP Address. Please insert new IP Address.", vbOKOnly,
"Error"
        GoTo End_Con
```

```
    End If

    nConnect = EthernetIO.EthConnectTo(IPServerStr)
    nWatchComm = EthernetIO.EthWatchComm
        If ((nConnect And nWatchComm) = 1) Then
        cmdConnect.Enabled = False
        cmdDisconnect.Enabled = True
        cmdRDPortC.Enabled = True
        lblStatus = ("Connection established with EthernetIO Board : ") + IPServerStr
      Else
End_Con:        lblStatus = STATUS_CONN_FAILED
    End If
End Sub
```

3.2.5) Double click on “Disconnect“ button to add cmdDisconnect_Click() procedure. This procedure use to disconnect to Ethernet IO board.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Summary : Close connection Ethernet IO Board
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub cmdDisconnect_Click()
    EthernetIO.EthStopComm
    Dim nResult As Integer
    nResult = EthernetIO.EthIsOpen()
    If (nResult = 1) Then
        cmdConnect.Enabled = True
        cmdDisconnect.Enabled = False
        cmdRDPortC.Enabled = False
        lblStatus = STATUS_DISCONN_MSGSTATUS_READY_MSG
    End If
```

```
End Sub
```

3.2.6) Double click on “RD_PORTC” button add to cmdRDPortC_Click() procedure.

This procedure use to read Logic value of Port C from Ethernet IO board.

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Summary : Read Logic value Port_C
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Sub cmdRDPortC _Click()
Dim btValue As Byte
Dim btPort  As Byte
Dim bResult As Integer
Dim nLen     As Integer
Dim strValue As String

        btPort = PORT_C
        bResult = EthernetIO.EthReadPort(btPort, btValue)

        If (bResult = 0) Then
            Debug.Print "ReadPort fail."
            GoTo End2
        End If
        strValue = Hex(btValue)
        strValue = LCase(strValue)
        nLen = Len(strValue)
        If (nLen = 1) Then
            txtRDPort.Text = "0x0" + strValue
        Else
            txtRDPort.Text = "0x" + strValue
        End If
```

```
End2:
End Sub
```

3.2.7) Double click on “Timer1” ID component add to tmrCtl_Timer() procedure. This procedure have been polling check incoming data from serial port of Ethernet IO board every 5mS.

```
''''''''''''''''''''''''''''''''''''''''''''''
' Summary : Receive data from serial port.
''''''''''''''''''''''''''''''''''''''''''''''
Private Sub tmrCtl_Timer()
    Dim btRecieve(256) As Byte
    Dim lCount As Long
    Dim nResult As Integer, i As Integer
    nResult = EthernetIO.EthDataRecieved(btRecieve(0), lCount)
        If (lCount <> 0) Then
        For i = 0 To lCount - 1
            strData = strData + Chr(btRecieve(i))
        Next i
        txtDataReceived.Text = strData
    End If
End Sub
```

4. Runing Example application

After Making Example application finished. We need to test it with hardware board. This application is simple for reading port c (8 bit) and polling receive data form Ethernet IO board. If user wants to use other function, user can refer to Prototype Function topic.

4.1. Set Up

1) Connecting cross cable or LAN cable between PC Host and Ethernet IO Board.
2) Connecting serial cable of Ethernet IO to com port of PC Host and power on Ethernet IO board.
3) Opening HyperTerminal select COM port of PC Host and select port setting as below.

Baud Rate	:	9600 bps
Data Bits	:	8
Parity Bits	:	None
Stop bits	:	1
Flow Control	:	None

***Note: You must set port setting coincident with UART Setting of Ethernet IO. ***

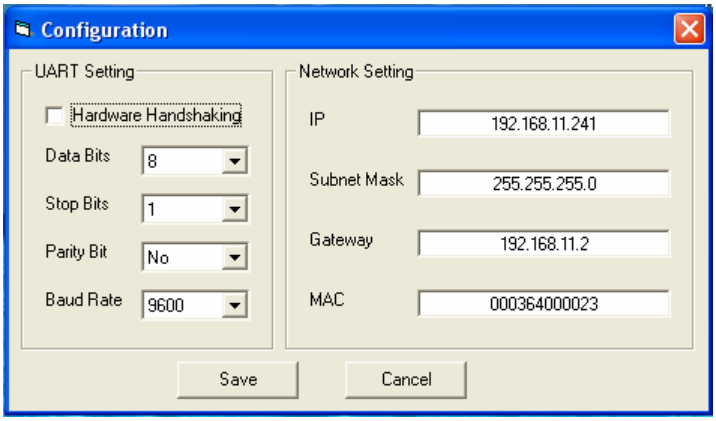

Figure 4-1 Show Configuration Dialog

4.2. Testing run Example application

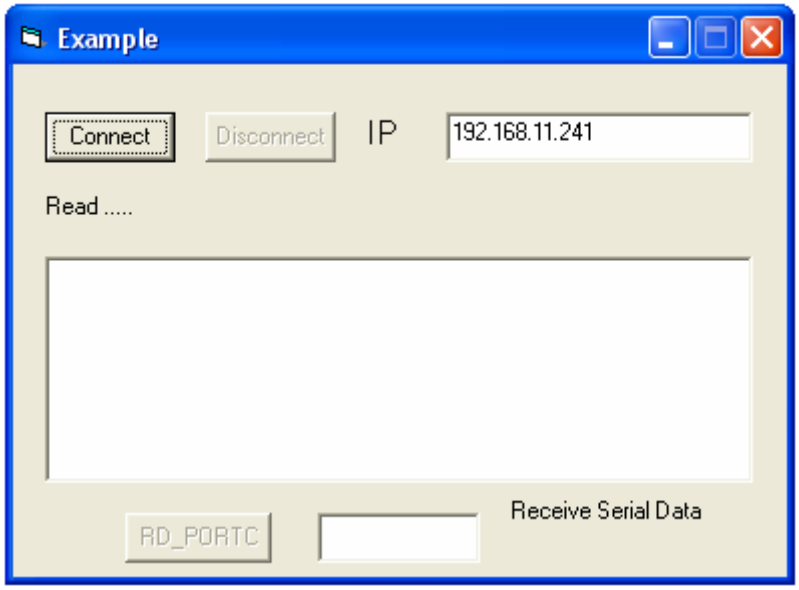

Figure 4-2 Show Example Application Dialog

1) Click “Connect” button to connect to Ethernet IO Board. If connection success, it show Connection established message “*Connection established with Ethernet IO Board: (IP of Ethernet IO) “on* status bar. And if connection unsuccessful, it shows message “connection failed”. Show in figure 4-2.

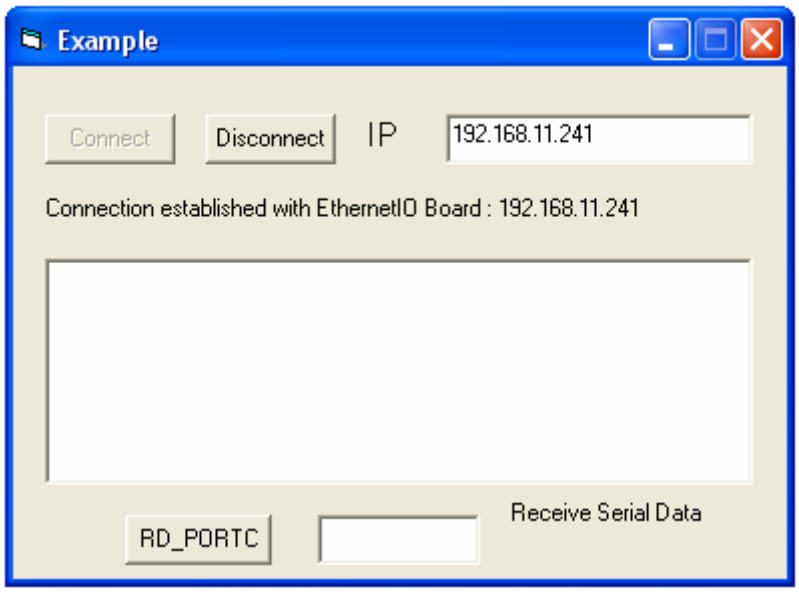

Figure 4-3 Show Connection Established

2) Open HyperTeminal program when you type character on HyperTerminal. It will be show character on textbox Receive Serial Data of Example application. Show in figure 4-3

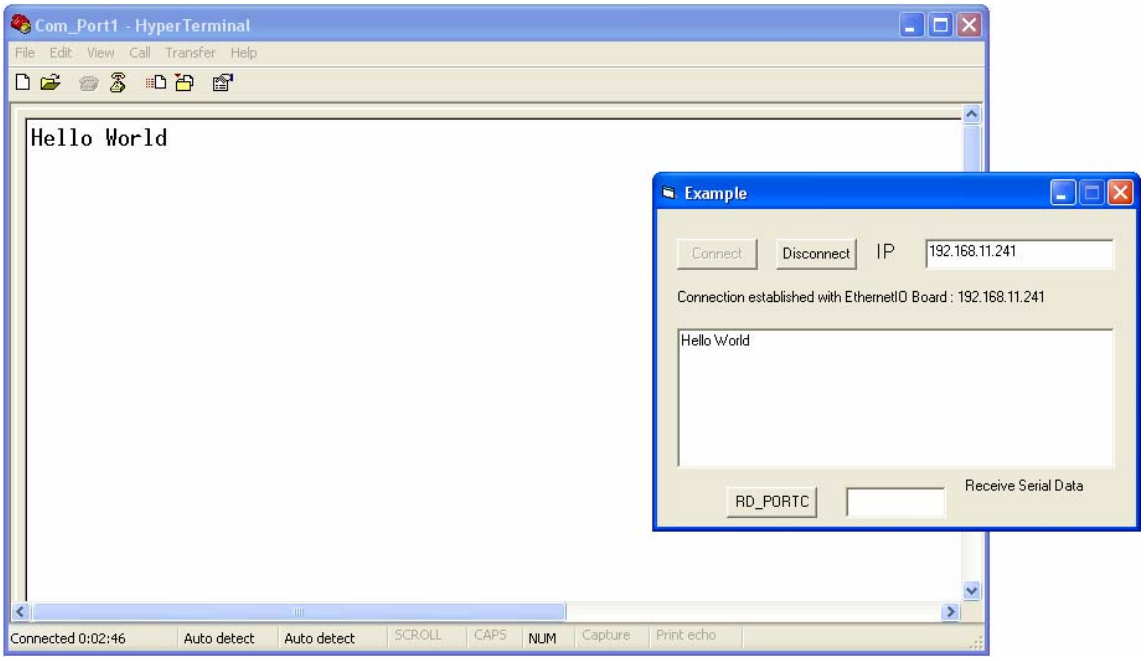

Figure 4-4 Show Receive Data from Serial Port

3) Click "RD_PORTC" button, it will be display logic value of port c on textbox near RD_PORTC button.

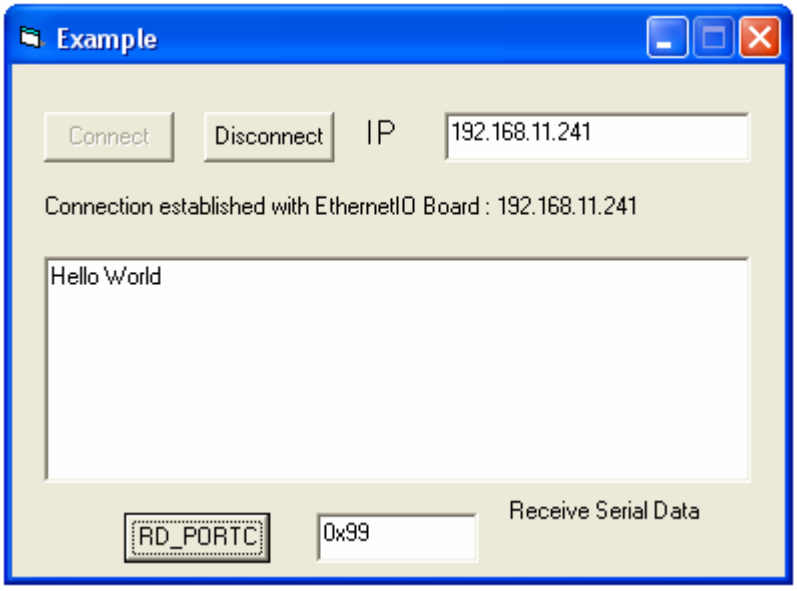

Figure 4-5 Show Display Logic Value of GPIO PORTC

4) If user click “Disconnect” button, this program will be disconnection with Ethernet IO board. It show message “Disconnected” on status bar.

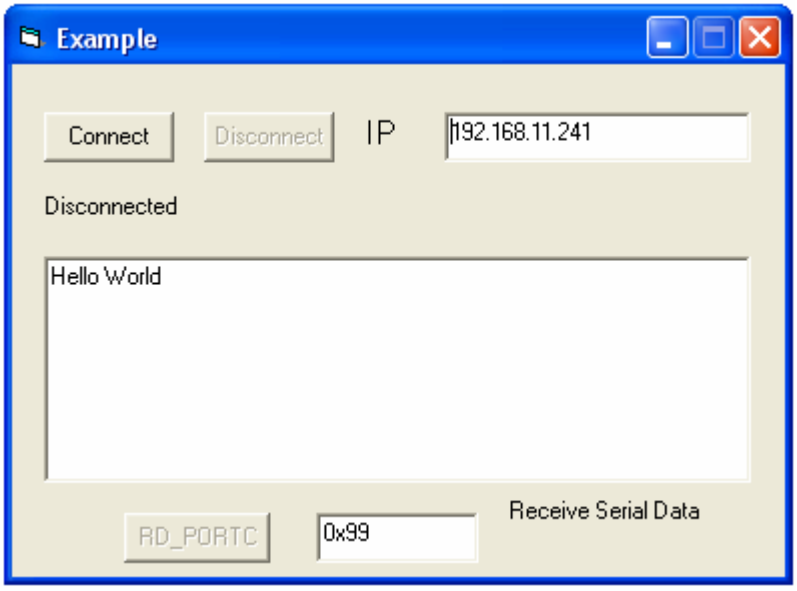

Figure 4-6 Show Disconnection

5. Prototype Function

This is topic description all function that provided by Ethernet IO DLL. Ethernet IO development provides function library in DLL module (EtherIODll). It is easy for user to use it without any knowledge about Winsock. User can include this DLL module to user’s Standard EXE application project and develop application base on example program. EtherIODll.DLL contains 2 classes as below.

- **SockAddrIn** contains utility function of Winsock programming (not necessary for communication with Ethernet IO board)
- **CEtherIOComm** is the main class for sending and receiving data from Ethernet IO. This class uses multithread working, one for send data and one for waiting incoming packet. User must override some functions in this class to handle incoming packet.

5.1. Communication Function

Ethernet IO board is a network embedded board. Another application can control this board via TCP/IP protocol. Library also contains function for communication with Ethernet IO board.

5.1.1) EthConnectTo

1). Format	EthConnectTo (ByVal strDestination As String) As Integer	
Function	Open communication with Ethernet IO board.	
Parameter	ByVal strDestination As String	IP address in string definition
Return value	Integer True (1)	Open communication successful
	Integer False (0)	Failed to open communication. Check LAN cable or power supply on Ethernet IO board

5.1.2) EthWatchComm

2). Format	EthWatchComm ( ) As Integer	
Function	Create communication monitor thread. For receiving data from Ethernet IO	
Parameter	-	-
Return value	Integer True (1)	Created communication monitor thread successful
	Integer False (0)	Fail to create communication monitor thread.

5.1.3) EthWriteComm

3). Format	EthWriteComm (ByVal lpBuffer As String, ByVal dwCount As Long, dwTimeout As Long) As Long
Function	Send data to target Ethernet IO board.

Parameter	ByVal lpBuffer As String	Data to send in string format
	ByVal dwCount As Long	Size of data to send
	dwTimeout As Long	Specified time out for sending data. This parameter can be infinity time (INFINITE)
Return value	As Long	Size of data that can send to Ethernet IO board

5.1.4) EthDataRecieved

4). Format	EthDataRecieved (ByRef strData As Byte, ByRef lCount As Long) As Integer	
Function	This function will be call when there are incoming data from Ethernet IO board. User must overwrite this function to store receive data.	
Parameter	ByRef strData As Byte	Data from Ethernet IO board in pointer format
	ByRef lCount As Long	Size of receiving data
Return value	-	

5.1.5) EthStopComm

5). Format	EthStopComm ( )	
Function	Call this function to close communication with Ethernet IO board.	
Parameter	-	-
Return value	-	-

5.1.6) EthOnEvent

6). Format	EthOnEvent (ByVal uEvent As Integer)	
Function	This function will call when there are event on communication with Ethernet IO board. User must overwrite this function to accept event message	
Parameter	ByVal uEvent As Integer	Event message ID EVT_CONSUCCESS 0x0000 Connection established EVT_CONFAILURE 0x0001 Wait Connection failed EVT_CONDROP 0x0002 Connection dropped EVT_ZEROLENGTH 0x0003 Zero length message
Return value	-	

5.1.7) EthSetUartOpt

7). Format	EthSetUartOpt (ByVal DataBit As String, ByVal StopBit As String, ByVal ParityBit As String, ByVal HwHandshak As Boolean, ByVal BaudRate As Long) As Integer	
Function	Set UART option such as baud rate, data bit, etc in runtime mode. It useful for communication with another UART board. This function should be call before sending data to Ethernet IO board.	
Parameter	ByVal DataBit As String	Number of data bit in serial protocol can be 7 or 8

	ByVal StopBit As String	Number of stop bit in serial protocol can be 1 or 2
	ByVal ParityBit As String	Number of parity bit in serial protocol can be 0 (NO) 1 (ODD) 2 (EVEN)
	ByVal HwHandshak As Boolean	TRUE For enable hardware handshake pin(CTS, RTS pin) (no available in Ethernet IO board) FALSE Disable hardware handshake pin.
	ByVal BaudRate As Long	UART board rate in bit/seconds, can be set to 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200, 230400, 460800, 921600
Return value	Integer True (1)	Set UART option successful
	Integer False (0)	Failed to set UART option

5.2. I/O Port Function

5.2.1) EthSetBit

8). Format	EthSetBit (ByVal port As Byte, ByVal pin As Byte) As Integer	
Function	Set status of specified pin to logic high. I/O port to set in Ethernet IO board. User must	
Parameter	ByVal port As Byte	I/O port to set in Ethernet IO board. User must specified port in this definition

		PORT_A 0x0020 PORT_B 0x0024 PORT_C 0x0028 PORT_D 0x002C PORT_E 0x0030 PORT_F 0x0034 PORT_G 0x0038
	ByVal pin As Byte	Pin number to set. Can be 0 to 7
Return value	Integer True (1)	Set port command successful
	Integer False (0)	Failed to set port on Ethernet IO board.

5.2.2) EthClearBit

9). Format	EthClearBit (ByVal port As Byte, ByVal pin As Byte) As Integer	
Function	Set status of specified pin to logic low.	
Parameter	ByVal port As Byte	I/O port in Ethernet IO board
	ByVal pin As Byte	Pin number to set. Can be 0 to 7
Return value	Integer True (1)	Clear port to set command successful
	Integer False (0)	Failed to clear port on Ethernet IO board.

5.2.3) EthOutPort

10). Format	EthOutPort (ByVal port As Byte, ByVal SendByte As Byte) As Integer	
Function	Send one byte to specified port on Ethernet IO board.	
Parameter	ByVal port As Byte	I/O port to set in Ethernet IO board

	ByVal SendByte As Byte	One byte data to send.
Return value	Integer True (1)	Write port command successful
	Integer False (0)	Failed to write port on Ethernet IO board.

5.2.4) EthReadBit

11). Format	EthReadBit (ByVal port As Byte, ByVal pin As Byte, ByRef bit As Byte) As Integer	
Function	Read the status of specified pin on Ethernet IO board	
Parameter	ByVal port As Byte	I/O port to set in Ethernet IO board
	ByVal pin As Byte	Pin number to set. Can be 0 to 7
	ByRef bit As Byte	Pointer of read byte data. Status of specified pin on Ethernet IO board, can be 0 (low) or 1 (high)
Return value	Integer True (1)	Read pin command successful
	Integer False (0)	Failed to read pin on Ethernet IO board.

5.2.5) EthReadPort

12). Format	EthReadPort (ByVal port As Byte, ByRef ReadByte As Byte) As Integer	
Function	Read status of all pin in specified port on Ethernet IO board.	
Parameter	ByVal port As Byte	I/O port in Ethernet IO board
	ByRef ReadByte As Byte	Pointer of read byte data. an be 0x00 to 0xFF
Return value	Integer True (1)	Read port command successful

	Integer False (0)	Failed to Read port on Ethernet IO board.

5.2.6) EthReadADC

13). Format	EthReadADC (ByVal channel As Byte, ByRef ADCValue As Integer) As Integer	
Function	Read 10 bit ADC (Analog to Digital Converter) value on specified channel. Ethernet IO provide **6 ADC channel** on port G.	
Parameter	ByVal channel As Byte	Channel number to set. **Can be 0 to 5**
	ByRef ADCValue As Integer	Pointer of ADC value.
Return value	Integer True (1)	Read ADC value command successful
	Integer False (0)	Failed to read ADC value on Ethernet IO board.

5.3. Parallel I/O function

5.3.1) EthFifoWrite

14). Format	EthFifoWrite (ByVal addr As Byte, ByRef IpBuffer As Integer, ByVal dwCount As Integer) As Integer
Function	Writing data from buffer to Parallel I/O on Ethernet IO. Size must not exceed to 1024 bytes. Note: Port C is Data 8 to Data 15 Port D is Data 0 to Data 7 Port B 4 is Write signal Port B 5 is Read signal Port B 0 is Address 0 Port B 1 is Address 1

Parameter	ByVal addr As Byte	Address for parallel mode. Can be 0 to 3
	ByRef IpBuffer As Integer	Buffer that store byte of data to send
	ByVal dwCount As Integer	Size of data to write.
Return value	Integer True (1)	Writing to parallel mode successful
	Integer False (0)	Writing to parallel mode failed

Table 5-1 Write Data Timing Diagram

Symbol	Parameter	Min	Max	Unit
tp_{wr}	Period of Write Data	120	-	nS.
ts_{wr}	Setup Time of Write Data	60	-	nS.
th_{wr}	Hold Time of Write Data	90	-	nS.

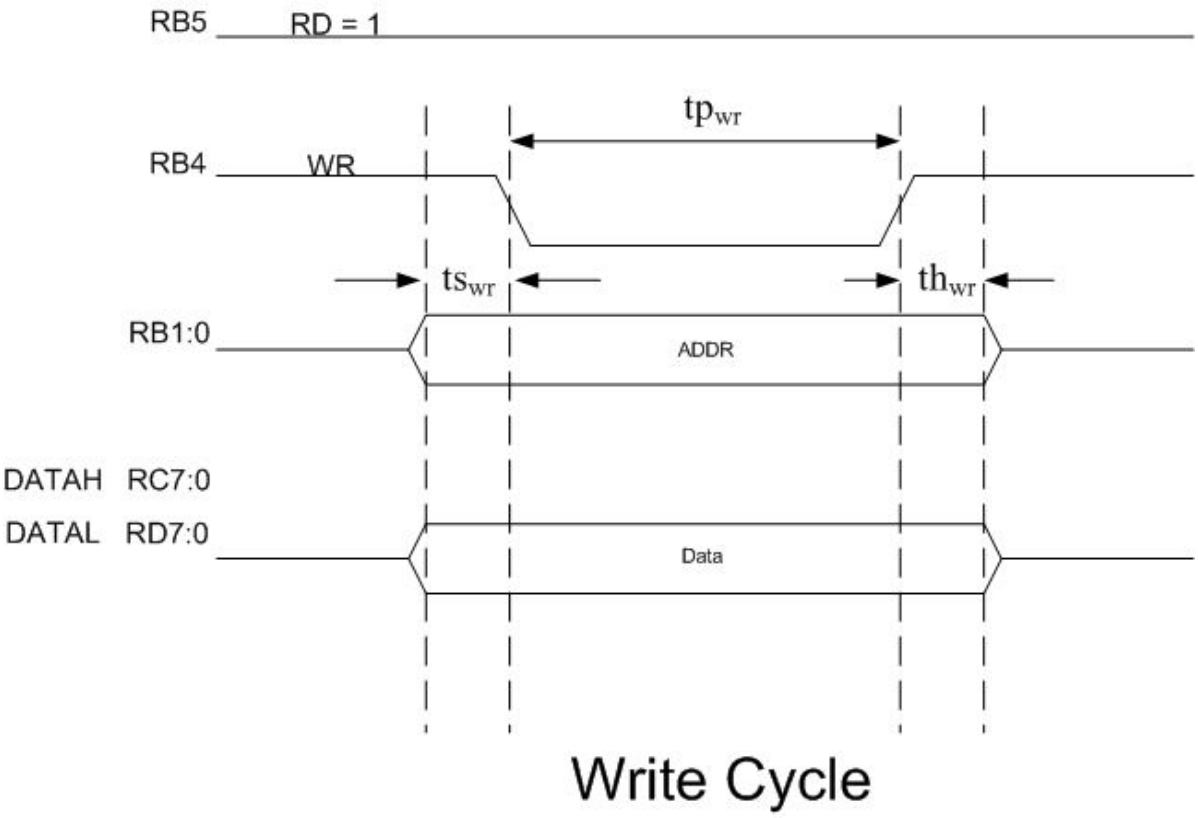

Figure 5-1 Write Timing Diagram

5.3.2) EthFifoRead

15). Format	EthFifoRead (ByVal addr As Byte, ByRef lpBuffer As Integer, ByVal dwCount As Integer) As Integer	
Function	Reading from parallel I/O on Ethernet IO board. Size must not exceed to 1024 byte too.	
Parameter	ByVal addr As Byte	Address for parallel mode. Can be 0 to 3
	const LPBYTE lpBuffer	Buffer that store byte of data to read
	ByVal dwCount As Integer	Size of data to read.
Return value	Integer True (1)	Writing to parallel mode successful
	Integer False (0)	Writing to parallel mode failed

Table 5-2 Read Data Timing Diagram

Symbol	Parameter	Min	Max	Unit
tp_{rd}	Period of Read Data	120	-	nS.
ts_{rd}	Setup Time of Read Data	60	-	nS.
th_{rd}	Hold Time of Read Data	90	-	nS.
tsd_{rd}	Setup Time of Data Read	60	-	nS.
thd_{rd}	Hold Time of Data Read Data	60	-	nS.

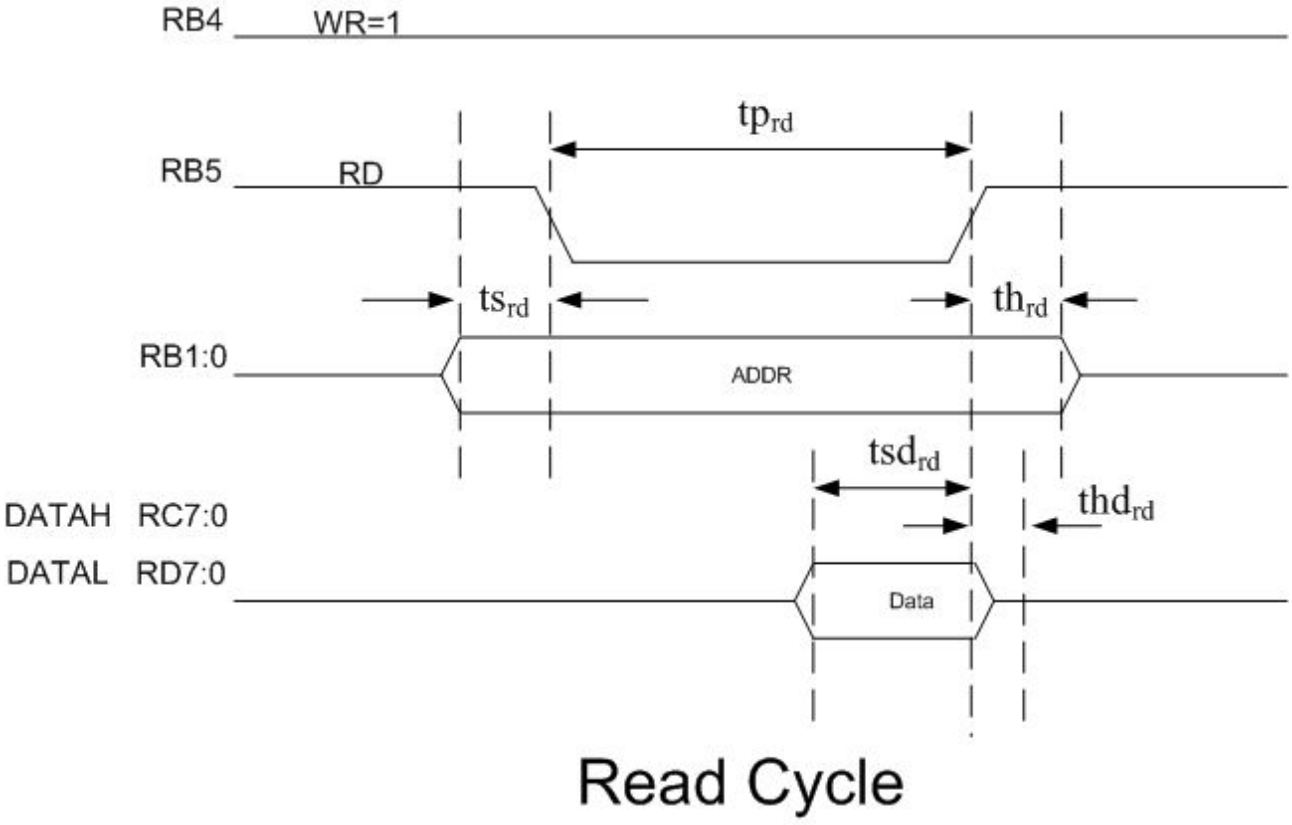

Figure 5-2 Read Data Timing

5.3.3) EthWrUserFlash

16). Format	EthWrUserFlash (ByVal addr As Byte, ByVal lpBuffer As Integer, ByVal dwCount As Integer) As Integer	
Function	Writing data at Ethernet IO's user area flash memory. Size of buffer must not exceed to 500 byte.	
Parameter	ByVal addr As Byte	Address of flash memory to write.
	ByRef lpBuffer As Integer	Buffer that stored data to write
	ByVal dwCount As Integer	Size of data to write
Return value	Integer True (1)	Writing flash memory successful
	Integer False (0)	Writing flash memory failed

5.3.4) EthRdUserFlash

17). Format	EthRdUserFlash (ByVal addr As String, ByVal IpBuffer As String, ByVal dwCount As Long) As Integer	
Function	Read data from user area flash memory. Size of buffer must not exceed to 500 byte.	
Parameter	ByVal addr As String	Address of flash memory to read
	ByVal IpBuffer As String	Buffer that stored data to read
	ByVal dwCount As Long	Size of data to read
Return value	Integer True (1)	Reading flash memory successful
	Integer False (0)	Failed to reading flash memory

5.4. Special Function

5.4.1) EthSystemSoftReset

18). Format	EthSystemSoftReset () As Integer	
Function	Reset target Ethernet IO board. This function will cause Ethernet IO IC to be “POWER ON RESET” state.	
Parameter	-	-
Return value	Integer True (1)	Ethernet IO board resetting successful
	Integer False (0)	Failed to reset Ethernet IO board.

5.4.2) EthSaveConfig

19). Format	EthSaveConfig (ByVal DataBit As Byte, ByVal StopBit As Byte, ByVal ParityBit As Byte,

	ByVal HwHandshak As Boolean, ByVal BaudRate As Long) As Integer	
Function	Save Ethernet IO configuration such as UART setting and network setting to flash memory in Ethernet IO. After call this function, it will reset Ethernet IO board immediately. Please refer to example application for using this function	
Parameter	ByVal DataBit As Byte	Number of data bit in serial protocol can be 7 or 8
	ByVal StopBit As Byte	Number of stop bit in serial protocol can be 1 or 2
	ByVal ParityBit As Byte	Number of parity bit in serial protocol can be 0 (NO) 1 (ODD) 2 (EVEN)
	ByVal HwHandshak As Boolean	TRUE For enable hardware handshake pin(CTS, RTS pin) (no available in Ethernet IO board) FALSE Disable hardware handshake pin.
	ByVal BaudRate As Long	UART board rate in bit/seconds, can be set to – 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200, 460800, 921600
Return value	Integer True (1)	Save configuration successful
	Integer False (0)	Failed to save configuration.

5.4.3) EthSaveIP

20). Format Return value Parameter Function	EthSaveIP (ByRef OldIP1 As Byte, ByRef OldIP2 As Byte, ByRef OldIP3 As Byte, ByRef OldIP4 As Byte) As Integer	
Function	Save IP Address configuration	
Parameter	ByRef OldIP1 As Byte	Digit 1
	ByRef OldIP2 As Byte	Digit 2
	ByRef OldIP3 As Byte	Digit 3
	ByRef OldIP4 As Byte	Digit 4
Return value	Integer True (1)	Save IP Address configuration successful
	Integer False (0)	Failed to save IP Address configuration.

5.4.4) EthLoadConfig

21). Format	EthLoadConfig (ByRef DataBit As Byte, ByRef StopBit As Byte, ByRef ParityBit As Byte, ByRef HwHandshak As Boolean, ByRef BaudRate As Long) As Integer	
Function	Load Ethernet IO configuration such as UART setting and network setting. Please refer to example application for using this function	
Parameter	ByRef DataBit As Byte	Number of data bit in serial protocol can be 7 or 8
	ByRef StopBit As Byte	Number of stop bit in serial protocol can be 1 or 2

	ByRef ParityBit As Byte	Number of parity bit in serial protocol can be 0 (NO) 1 (ODD) 2 (EVEN)
	ByRef HwHandshak As Boolean	TRUE For enable hardware handshake pin(CTS, RTS pin) (no available in Ethernet IO board) FALSE Disable hardware handshake pin.
	ByRef BaudRate As Long	UART board rate in bit/seconds
Return Vale	Integer True (1)	Load configuration successful
	Integer False (0)	Failed to load configuration

5.4.5) EthLoadConfig

22). Format	EthLoadConfig (ByRef DataBit As Byte, ByRef StopBit As Byte, ByRef ParityBit As Byte, ByRef HwHandshak As Boolean, ByRef BaudRate As Long) As Integer	
Function	Load Ethernet IO configuration such as UART setting and network setting. Please refer to example application for using this function	
Parameter	ByRef DataBit As Byte	Number of data bit in serial protocol can be 7 or 8
	ByRef StopBit As Byte	Number of stop bit in serial protocol can be 1 or 2

	ByRef ParityBit As Byte	Number of parity bit in serial protocol can be 0 (NO) 1 (ODD) 2 (EVEN)
	ByRef HwHandshak As Boolean	TRUE For enable hardware handshake pin(CTS, RTS pin) (no available in Ethernet IO board) FALSE Disable hardware handshake pin.
	ByRef BaudRate As Long	UART board rate in bit/seconds
Return Vale	Integer True (1)	Load configuration successful
	Integer False (0)	Failed to load configuration

5.4.6) EthLoadIP

23). Format	EthLoadIP(ByRef OldIP1 As Byte, ByRef OldIP2 As Byte, ByRef OldIP3 As Byte, ByRef OldIP4 As Byte) As Integer	
Function	Load IP Address configuration	
Parameter	ByRef OldIP1 As Byte	Digit 1
	ByRef OldIP2 As Byte	Digit 2
	ByRef OldIP3 As Byte	Digit 3
	ByRef OldIP4 As Byte	Digit 4
Return Vale	Integer True (1)	Load IP Address configuration success
	Integer False (0)	Failed to load IP Address configuration

5.4.7) EthLoadSupnet

24). Format	EthLoadSupnet (ByRef SupnetIP1 As Byte, ByRef SupnetIP2 As Byte, ByRef SupnetIP3 As Byte , ByRef SupnetIP4 As Byte) As Integer	
Function	Load Subnet IP Address configuration	
Parameter	ByRef SupnetIP1 As Byte	Digit 1
	ByRef SupnetIP2 As Byte	Digit 2
	ByRef SupnetIP3 As Byte	Digit 3
	ByRef SupnetIP4 As Byte	Digit 4
Return Vale	Integer True (1)	Load Subnet mask configuration successful
	Integer False (0)	Failed to load Subnet configuration

5.4.8) EthLoadGateway

25). Format	EthLoadGateway (ByRef GatewayIP1 As Byte, ByRef GatewayIP2 As Byte, ByRef GatewayIP3 As Byte, ByRef GatewayIP4 As Byte) As Integer	
Parameter	ByRef GatewayIP1 As Byte	Digit 1
	ByRef GatewayIP2 As Byte	Digit 2
	ByRef GatewayIP3 As Byte	Digit 3
	ByRef GatewayIP4 As Byte	Digit 4
Return Vale	Integer True (1)	Load Gateway configuration successful
	Integer False (0)	Failed to load Gateway configuration

5.4.9) EthLoadMacAdd

26). Format	EthLoadMacAdd (ByRef MacAdd1 As Byte, ByRef MacAdd2 As Byte, ByRef MacAdd3 As Byte, ByRef MacAdd4 As Byte, ByRef MacAdd5 As Byte, ByRef MacAdd6 As Byte) As Integer	
Function	Load Mac Address configuration	
Parameter	ByRef MacAdd1 As Byte	Digit 1
	ByRef MacAdd2 As Byte	Digit 2
	ByRef MacAdd3 As Byte	Digit 3
	ByRef MacAdd4 As Byte	Digit 4
	ByRef MacAdd5 As Byte	Digit 5
	ByRef MacAdd6 As Byte	Digit 6
Return Vale	Integer True (1)	Load Gateway configuration successful
	Integer False (0)	Failed to load Gateway configuration

6. Reference

File Name	Version
http://www.design-gateway.com	-

Note

54 BB Building, 13th Floor, Room No.1302, Sukhumvit 21 Rd. (Asoke)

Klongtoey-Nua, Wattana, Bangkok 10110 Thailand

Tel. (662)2612277, Fax. (662)261-2290

www.design-gateway.com