

# NModbus API 手冊

版本 1.2 , 2013.10

Written by Renee Lin

# 目錄

目錄 .....	2
<b>1. 關於手冊 .....</b>	<b>4</b>
<b>2. NModbus Master API.....</b>	<b>5</b>
2.1. CreateRtu .....	5
2.2. CreateAscii .....	6
2.3. CreateIp(TcpClient).....	7
2.4. CreateIp(UdpClient).....	11
2.5. Retries [屬性].....	12
2.6. ReadTimeout[屬性] .....	12
2.7. ReadCoils .....	13
2.8. ReadInputs.....	14
2.9. ReadHoldingRegisters.....	15
2.10. ReadInputRegisters.....	16
2.11. WriteSingleCoil .....	17
2.12. WriteSingleRegister .....	18
<b>3. NModbus Slave API.....</b>	<b>19</b>
3.1. CreateRtu .....	20
3.2. CreateAscii .....	21
3.3. CreateTcp.....	22
3.4. CreateUdp.....	23
3.5. CreateDefaultDataStore .....	24
3.6. ModbusSlaveRequestReceived[事件] .....	24
3.7. DataStoreWrittenTo[事件].....	25
3.8. Listen.....	28
3.9. CoilDiscretes [DO 資料陣列].....	28
3.10. InputDiscretes [DI 資料陣列] .....	29
3.11. HoldingRegisters [AO 資料陣列].....	29
3.12. InputRegisters [AI 資料陣列] .....	30

<b>4.</b>	<b>通用 API .....</b>	<b>31</b>
4.1.	Dispose.....	31
<b>附錄 A :</b>	<b>錯誤訊息.....</b>	<b>32</b>
<b>附錄 B :</b>	<b>AI, AO 型別轉換 .....</b>	<b>35</b>

# 1. 關於手冊

此手冊對於 NModbus 中所使用的 API 說明。

## 什麼是 NModbus ?

利用 NModbus 可實現 Modbus 的通訊協定，它是由一群自願者所開發以及維護，並且免費開放使用。

ICP DAS 基於官方發佈的 NModbus 原始碼進行功能確認與改善，程式開發員可以利用 ICP DAS 發佈的 DLL 進行 Windows 版 PC 或是 WinCE 版的 Modbus 程式開發。這個 DLL 具備以下功能

- a. Modbus/RTU Master/Slave
- b. Modbus/ASCII Master/Slave
- c. Modbus/TCP Master/Slave
- d. Modbus/UDP Master/Slave

下載 DLL 與範例程式。

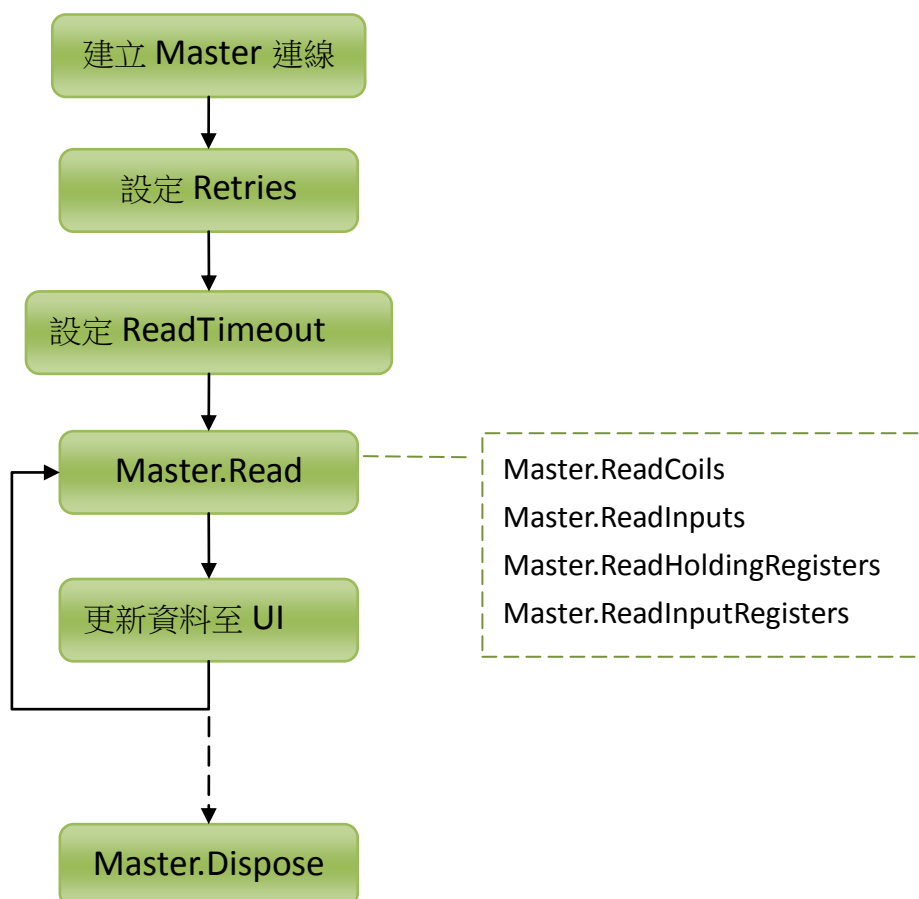
- a. WinForm 版本
  - [DLL 和文件](#) : nModbusPC.dll, log4net.dll
  - [範例程式](#) (For C#, VB.Net)
- b. WinCE 版本
  - [DLL 和文件](#) : nModbusCE.dll, CABCDLL.dll, FC19.dll
  - [範例程式](#) (For C#, VB.Net)

了解更多有關 Modbus→[http://www.icpdas.com/products/PAC/i-8000/modbus\\_c.htm](http://www.icpdas.com/products/PAC/i-8000/modbus_c.htm)

## 適用 NModbus 開發的 PAC

WinForm	XPAC(WES 2009)
	Win8,Win7,Vista,Xp(需有.NET framework 2(含)以上)
WinCE	ViewPAC(CE5)
	WinPAC(CE5)
	XPAC(CE6)

## 2.NModbus Master API



### 2.1.CreateRtu

建立 modbus master RTU 的連線。

#### 語法

```
C#  
ModbusSerialMaster CreateRtu(  
    SerialPort serialPort  
)
```

## 變數

### *serialPort*

*serialPort* 是由 `new SerialPort()` 建立，而序列埠必需先被 `serialPort.Open()` 開啟。  
若 *serialPort* 沒有指定值，則會由系統帶入預設的值。例如：連接埠名稱為 COM1，  
同位檢查為 None，資料位元為 8，停止位元為 1。

## 回傳值

回傳 `ModbusSerialMaster`。

## 範例

```
[C#]  
SerialPort serialPort = new SerialPort(); //Create a new SerialPort object.  
serialPort.PortName = "COM1";  
serialPort.BaudRate = 115200;  
serialPort.DataBits = 8;  
serialPort.Parity = Parity.None;  
serialPort.StopBits = StopBits.One;  
serialPort.Open();  
ModbusSerialMaster master = ModbusSerialMaster.CreateRtu(serialPort);
```

## 2.2. CreateAscii

建立 modbus master Ascii 的連線。

## 語法

```
C#  
ModbusSerialMaster CreateAscii(  
    SerialPort serialPort  
)
```

## 變數

### *serialPort*

serialPort 是由 new SerialPort()建立，而序列埠必需先被 serialPort.Open()開啟。

## 回傳值

回傳 ModbusSerialMaster 。

## 範例

```
[C#]  
SerialPort serialPort = new SerialPort(); //Create a new SerialPort object.  
serialPort.PortName = "COM1";  
serialPort.BaudRate = 115200;  
serialPort.DataBits = 8;  
serialPort.Parity = Parity.None;  
serialPort.StopBits = StopBits.One;  
serialPort.Open();  
ModbusSerialMaster master = ModbusSerialMaster.CreateAscii(serialPort);
```

## 2.3. CreateIp(TcpClient)

建立 modbus master IP 的連線。

## 語法

```
C#  
ModbusIpMaster CreateIp(  
    TcpClient tcpClient  
)
```

## 變數

*tcpClient*

tcpClient 是由 new TcpClient()建立，而 tcpClient 必需由 tcpClient.BeginConnect()產生連接。

## 回傳值

回傳 ModbusIpMaster 。

## 範例

**[C#]**

```
string ipAddress = "10.0.0.69";
int tcpPort = 502;
TcpClient tcpClient = new TcpClient(); //Create a new TcpClient object.
tcpClient.BeginConnect(ipAddress, tcpPort, null, null);
ModbusIpMaster master = ModbusIpMaster.CreateIp(tcpClient);
```

若是斷線時要重新連線，請參考下列程式碼。

**[C#]**

```
[DllImport("WININET", CharSet = CharSet.Auto)]
static extern bool InternetGetConnectedState(ref InternetConnectionState lpdwFlags, int
dwReserved);
enum InternetConnectionState : int
{
    INTERNET_CONNECTION_MODEM = 0x1,
    INTERNET_CONNECTION_LAN = 0x2,
    INTERNET_CONNECTION_PROXY = 0x4,
    INTERNET_RAS_INSTALLED = 0x10,
    INTERNET_CONNECTION_OFFLINE = 0x20,
    INTERNET_CONNECTION_CONFIGURED = 0x40
}
TcpClient tcpClient;
ModbusIpMaster master;
string ipAddress = "10.0.0.69";
int tcpPort = 502;
DateTime dtDisconnect = new DateTime();
DateTime dtNow = new DateTime();
bool NetworkIsOk = false;

private void btStart_Click(object sender, EventArgs e)
{
    // when button clicked, connect to Modbus TCP Server
    NetworkIsOk = Connect();
    timer1.Enabled = true;
}
private bool Connect()
```



```

{
    if (master != null)
        master.Dispose();
    if (tcpClient != null)
        tcpClient.Close();
    if (CheckInternet())
    {
        try
        {
            tcpClient = new TcpClient();
            IAsyncResult asyncResult = tcpClient.BeginConnect(ipAddress, tcpPort, null, null);
            asyncResult.AsyncWaitHandle.WaitOne(3000, true); //wait for 3 sec
            if (!asyncResult.IsCompleted)
            {
                tcpClient.Close();
                Console.WriteLine(DateTime.Now.ToString() + ":Cannot connect to server.");
                return false;
            }
            // create Modbus TCP Master by the tcpclient
            master = ModbusIpMaster.CreateIp(tcpClient);
            master.Transport.Retries = 0; //don't have to do retries
            master.Transport.ReadTimeout = 1500;
            Console.WriteLine(DateTime.Now.ToString() + ":Connect to server.");
            return true;
        }
        catch (Exception ex)
        {
            Console.WriteLine(DateTime.Now.ToString() + ":Connect process " + ex.StackTrace +
"==>" + ex.Message);
            return false;
        }
    }
    return false;
}

private bool CheckInternet()
{
    //http://msdn.microsoft.com/en-us/library/windows/desktop/aa384702(v=vs.85).aspx

```

```

InternetConnectionState flag = InternetConnectionState.INTERNET_CONNECTION_LAN;
return InternetGetConnectedState(ref flag, 0);
}
private void timer1_Tick(object sender, EventArgs e)
{
    //start timer1, timer1.Interval = 1000 ms
    try
    {
        if (NetworkIsOk)
        {
            #region Master to Slave
            //read AI, AO, DI, DO
            #endregion
        }
        else
        {
            //retry connecting
            dtNow = DateTime.Now;
            if ((dtNow - dtDisconnect) > TimeSpan.FromSeconds(10))
            {
                Console.WriteLine(DateTime.Now.ToString() + ":Start connecting");
                NetworkIsOk = Connect();
                if (!NetworkIsOk)
                {
                    Console.WriteLine(DateTime.Now.ToString() + ":Connecting fail. Wait for retry");
                    dtDisconnect = DateTime.Now;
                }
            }
            else
            {
                Console.WriteLine(DateTime.Now.ToString() + ":Wait for retry connecting");
            }
        }
    }
    catch(Exception ex)
    {
        if (ex.Source.Equals("System"))

```

```
{
    //set NetworkIsOk to false and retry connecting
    NetworkIsOk = false;
    Console.WriteLine(ex.Message);
    dtDisconnect = DateTime.Now;
}
}
```

## 2.4. CreateIp(UdpClient)

建立 modbus master IP 的連線。

### 語法

```
C#
ModbusIpMaster CreateIp(
    UdpClient udpClient
)
```

### 變數

*udpClient*

*udpClient* 是由 `new UdpClient()` 建立，而 *udpClient* 必需由 `udpClient.Connect()` 產生連接。

### 回傳值

回傳 `ModbusIpMaster`。

### 範例

```
[C#]
IPAddress ipAddress = "10.0.0.69";
int udpPort = 502;
UdpClient udpClient = new UdpClient(); //Create a new UdpClient object.
udpClient.Connect(ipAddress, udpPort);
ModbusIpMaster master = ModbusIpMaster.CreateIp(udpClient);
```

## 2.5.Retries [屬性]

[屬性]在遇到 IO 錯誤、逾時或損壞訊息等等的失敗情況後，重試發送訊息的次數。

### 語法

C#

```
int Retries { get; set; }
```

### 範例

[C#]

```
string ipAddress = "10.0.0.69"; //use TCP for example
int tcpPort = 502;
TcpClient tcpClient = new TcpClient();
tcpClient.BeginConnect(ipAddress, tcpPort, null, null);
ModbusIpMaster master = ModbusIpMaster.CreateIp(tcpClient);
master.Transport.Retries = 0;
```

### 注意

預設值為 Retries = 3。

在 NModbus 中不需要重發訊息，請設定 Retries = 0。

## 2.6.ReadTimeout[屬性]

[屬性] 取得或設定讀取作業未完成時，發生逾時之前的毫秒數。

### 語法

C#

```
int ReadTimeout { get; set; }
```

### 範例

[C#]

```
SerialPort serialPort = new SerialPort();//use RTU for example
```

```
serialPort.Open();
ModbusSerialMaster master = ModbusSerialMaster.CreateRtu(serialPort);
master.Transport.ReadTimeout = 300; //milliseconds
```

## 注意

ReadTimeout 建議設定值

- a. RTU: ReadTimeout = 300
- b. TCP: ReadTimeout = 1500

## 2.7. ReadCoils

讀取 DO 的狀態。

## 語法

```
C#
bool[] ReadCoils(
    byte slaveID,
    ushort startAddress,
    ushort numOfPoints
)
```

## 變數

*slaveID*

欲讀取裝置的 ID。

*startAddress*

開始讀取的位址。

*numOfPoints*

讀取的長度。

## 回傳值

回傳 bool[]。

## 範例

**[C#]**

```
byte slaveID = 1;  
ushort startAddress = 0;  
ushort numOfPoints = 10;  
bool[] coilstatus = master.ReadCoils(slaveID , startAddress , numOfPoints);
```

## 2.8.ReadInputs

讀取 DI 的狀態。

### 語法

**C#**

```
bool[] ReadInputs(  
    byte slaveID,  
    ushort startAddress,  
    ushort numOfPoints  
)
```

### 變數

*slaveID*

欲讀取裝置的 ID。

*startAddress*

開始讀取的位址。

*numOfPoints*

讀取的長度。

### 回傳值

回傳 bool[]。

### 範例

**[C#]**

```
byte slaveID = 1;  
ushort startAddress =0;  
ushort numOfPoints = 10;
```

```
bool[] status = master.ReadInputs(slaveID , startAddress , numOfPoints);
```

## 2.9.ReadHoldingRegisters

讀取 AO 的值。

### 語法

```
C#  
ushort[] ReadHoldingRegisters(  
    byte slaveID,  
    ushort startAddress,  
    ushort numOfPoints  
)
```

### 變數

*slaveID*

欲讀取裝置的 ID。

*startAddress*

開始讀取的位址。

*numOfPoints*

讀取暫存區的長度。

### 回傳值

回傳 ushort[]。

### ExampIs

```
[C#]  
byte slaveID = 1;  
ushort startAddress =0;  
ushort numOfPoints = 10;  
ushort[] holding_register = master.ReadHoldingRegisters(slaveID, startAddress,  
numOfPoints);
```

## 2.10. ReadInputRegisters

讀取 AI 的值。

### 語法

C#

```
ushort[] ReadInputRegisters(  
    byte slaveID,  
    ushort startAddress,  
    ushort numOfPoints  
)
```

### 變數

*slaveID*

欲讀取裝置的 ID。

*startAddress*

開始讀取的位址。

*numOfPoints*

讀取暫存區的長度。

### 回傳值

回傳 ushort[]。

### 範例

[C#]

```
byte slaveID = 1;  
ushort startAddress = 0;  
ushort numOfPoints = 10;  
ushort[] register = master.ReadInputRegisters(slaveID, startAddress, numOfPoints);
```



## 2.11. WriteSingleCoil

寫入值到 DO 位址。

### 語法

```
C#  
void WriteSingleCoil(  
    byte slaveID,  
    ushort coilAddress,  
    bool value  
)
```

### 變數

#### *slaveID*

欲寫入裝置的 ID。

#### *coilAddress*

欲寫入的位址。

#### *value*

若該位址要被寫入，則寫入值為是(TRUE)；  
若該位址沒有被寫入，則值為否(FALSE)。

### 回傳值

無回傳值。

### 範例

```
[C#]  
byte slaveID = 1;  
ushort coilAddress = 1;  
bool value = true;  
master.WriteSingleCoil(slaveID , coilAddress ,value);
```

## 2.12. WriteSingleRegister

寫入值到 AO 位址。

### 語法

C#

```
void WriteSingleRegister(  
    byte slaveID,  
    ushort registerAddress,  
    ushort value  
)
```

### 變數

*slaveID*

欲寫入裝置的 ID。

*registerAddress*

欲寫入的位址。

*value*

欲寫入的值。

### 回傳值

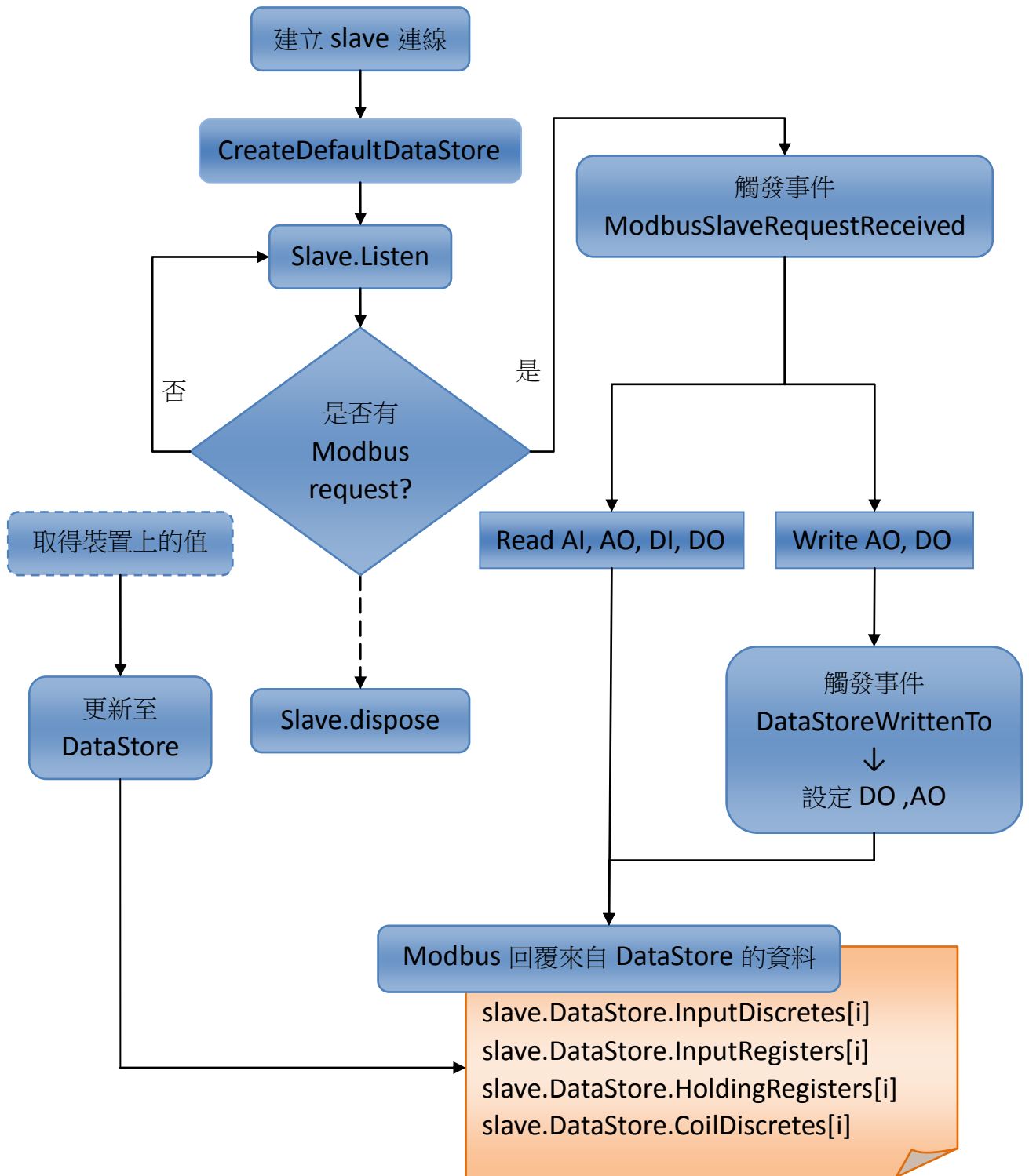
無回傳值。

### 範例

[C#]

```
byte slaveID = 1;  
ushort registerAddress = 1;  
ushort value = 1000;  
master.WriteSingleRegister(slaveID, registerAddress, value);
```

### 3.NModbus Slave API



### 3.1.CreateRtu

建立 Modbus slave Rtu 的連線。

#### 語法

```
C#
ModbusSerialSlave CreateRtu(
    byte slaveID,
    SerialPort serialPort
)
```

#### 變數

*slaveID*

欲建立連線裝置的 ID。

*serialPort*

序列埠必需被 serialPort.Open()開啟，而 serialPort 是由 new SerialPort()建立。

#### 回傳值

回傳 ModbusSerialSlave。

#### 範例

```
[C#]
byte slaveID = 1;
SerialPort serialPort = new SerialPort();
serialPort.PortName = "COM1";
serialPort.BaudRate = 115200;
serialPort.DataBits = 8;
serialPort.Parity = Parity.None;
serialPort.StopBits = StopBits.One;
serialPort.Open();
ModbusSlave slave = ModbusSerialSlave.CreateRtu(slaveID, serialPort);
```

## 3.2. CreateAscii

建立 Modbus slave Ascii 的連線。

### 語法

```
C#
ModbusSerialSlave CreateAscii(
    byte slaveID,
    SerialPort serialPort
)
```

### 變數

*slaveID*

欲建立連線裝置的 ID。

*serialPort*

序列埠必需被 `serialPort.Open()` 開啟，而 `serialPort` 是由 `new SerialPort()` 建立。

### 回傳值

回傳 `ModbusSerialSlave`。

### 範例

```
[C#]
byte slaveID = 1;
SerialPort serialPort = new SerialPort();
serialPort.PortName = "COM1";
serialPort.BaudRate = 115200;
serialPort.DataBits = 8;
serialPort.Parity = Parity.None;
serialPort.StopBits = StopBits.One;
serialPort.Open();
ModbusSlave slave = ModbusSerialSlave.CreateAscii(slaveID, serialPort);
```

### 3.3. CreateTcp

建立 Modbus slave TCP 的連線。TCP Slave 可接收連線最大數量為 50。

#### 語法

```
C#  
ModbusTcpSlave CreateTcp(  
    byte slaveID,  
    TcpListener tcpListener  
)
```

#### 變數

##### *slaveID*

欲建立連線裝置的 ID。

##### *tcpListener*

tcpListener 是由 new TcpListener () 建立，而 tcpListener 必需由 tcpListener.Start() 開始接聽。

#### 回傳值

回傳 ModbusTcpSlave。

#### 範例

```
[C#]  
int port = 502;  
IPHostEntry ipEntry = Dns.GetHostEntry(Dns.GetHostName());  
IPAddress[] addr = ipEntry.AddressList;  
TcpListener tcpListener = new TcpListener(addr[0], port);  
tcpListener.Start();  
  
ModbusSlave slave = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener);  
slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();  
slave.DataStore.DataStoreWrittenTo += new  
EventHandler<DataStoreEventArgs>(Modbus_DataStoreWriteTo);
```

```
slave.Listen();
```

### 3.4. CreateUdp

建立 Modbus slave UDP 的連線。

#### 語法

```
C#  
ModbusUdpSlave CreateUdp(  
    byte slaveID,  
    UdpClient client  
)
```

#### 變數

*slaveID*

欲建立連線裝置的 ID。

*client*

client 由 new UdpClient()初始化且與指定的 port 連結。

#### 回傳值

回傳 ModbusUdpSlave。

#### 範例

```
[C#]  
int port = 502;  
UdpClient client = new UdpClient(port);  
ModbusSlave slave = Modbus.Device.ModbusUdpSlave.CreateUdp(slaveID, client);  
  
slave.ModbusSlaveRequestReceived += new  
EventHandler<ModbusSlaveRequestEventArgs>(Modbus_Request_Event);  
slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();  
slave.DataStore.DataStoreWrittenTo += new  
EventHandler<DataStoreEventArgs>(Modbus_DataStoreWriteTo);  
slave.Listen();
```

### 3.5.CreateDefaultDataStore

建立暫存器的記憶體空間，AO,AI 暫存器預設為 0，DO,DI 暫存器預設為 false。記憶體大小預設為 65535，可設定範圍為 1-65535。

#### 語法

```
C#
```

```
DataStore CreateDefaultDataStore()
```

若要自訂記憶體大小，可以使用下列語法。

```
DataStore CreateDefaultDataStore(  
    ushort coilsCount,  
    ushort inputsCount,  
    ushort holdingRegistersCount,  
    ushort inputRegistersCount  
)
```

#### 範例

```
[C#]
```

```
slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();
```

#### 回傳值

回傳 DataStore。

#### 注意

slave 需由 ModbusSlave 定義並且建立 slave 的連線，例如：建立 TCP slave 連線為 slave = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener)。

### 3.6.ModbusSlaveRequestReceived[事件]

當 slave 收到 master 命令時觸發事件，取得 Modbus 命令封包後，可拆解封包做特殊應用。

#### 語法



C#

EventHandler<ModbusSlaveRequestEventArgs> ModbusSlaveRequestReceived

## 範例

```
[C#]
slave.ModbusSlaveRequestReceived += new
EventHandler<ModbusSlaveRequestEventArgs>(Modbus_Request_Event);
//trigger Modbus_Request_Event
private void Modbus_Request_Event(object sender,
Modbus.Device.ModbusSlaveRequestEventArgs e)
{
    //disassemble packet from master
    byte fc = e.Message.FunctionCode;
    byte[] data = e.Message.MessageFrame;
    byte[] byteStartAddress = new byte[] { data[3], data[2] };
    byte[] byteNum = new byte[] { data[5], data[4] };
    Int16 StartAddress = BitConverter.ToInt16(byteStartAddress, 0);
    Int16 NumOfPoint = BitConverter.ToInt16(byteNum, 0);

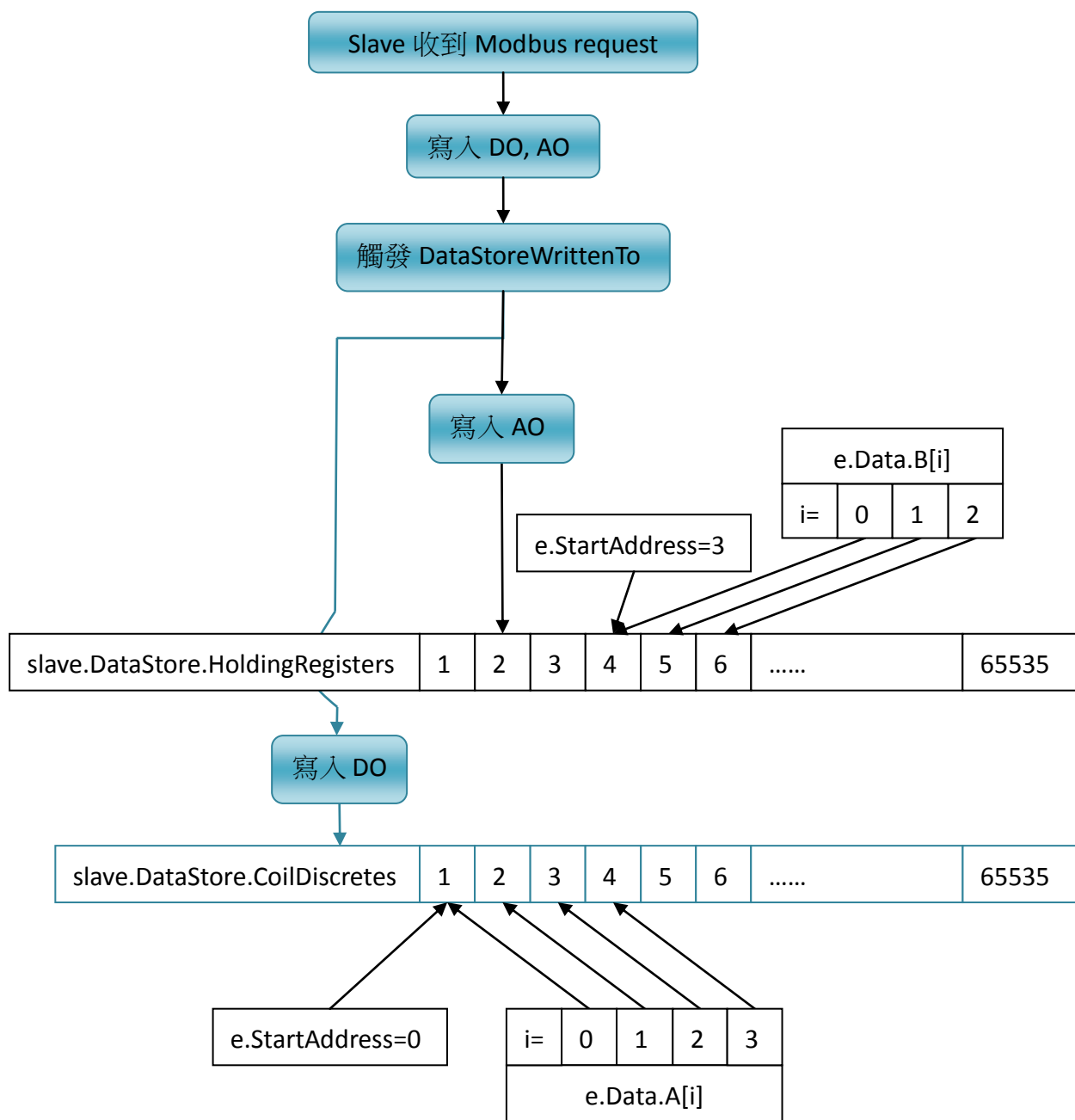
    Console.WriteLine(fc.ToString() + "," + StartAddress.ToString() + "," +
NumOfPoint.ToString());
}
```

## 注意

slave 需由 ModbusSlave 定義並且建立 slave 的連線，例如：建立 TCP slave 連線為 slave = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener)。

### 3.7.DataStoreWrittenTo[事件]

當 slave 收到 master 的命令寫入 DO 值或 AO 值到 DataStore 時觸發事件。位址為 1-65535。



## 語法

C#

EventHandler<DataStoreEventArgs> DataStoreWrittenTo

## 範例

[C#]

```

slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();
slave.DataStore.DataStoreWrittenTo += new
EventHandler<DataStoreEventArgs>(Modbus_DataStoreWriteTo);

//when receive write AO or DO command from master, it will trigger following function
private void Modbus_DataStoreWriteTo(object sender, Modbus.Data.DataStoreEventArgs e)
{
    switch (e.ModbusDataType)
    {
        case ModbusDataType.HoldingRegister:
            for (int i = 0; i < e.Data.B.Count; i++)
            {
                //Set AO
                //e.Data.B[i] already write to
slave.DataStore.HoldingRegisters[e.StartAddress + i + 1]
                //e.StartAddress starts from 0
                //You can set AO value to hardware here
            }
            break;
        case ModbusDataType.Coil:
            for (int i = 0; i < e.Data.A.Count; i++)
            {
                //set DO
                //e.Data.A[i] already write to
slave.DataStore.CoilDiscretes[e.StartAddress + i + 1]
                //e.StartAddress starts from 0
                //You can set DO value to hardware here
            }
            break;
    }
}
}

```

## 注意

slave 需由 ModbusSlave 定義並且建立 slave 的連線，例如：建立 TCP slave 連線為 slave = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener)。

## 3.8.Listen

Slave 開始監聽要求。

### 語法

C#

```
void Listen()
```

### 範例

```
[C#]
int port = 502; //use Tcp for example
IPHostEntry ipEntry = Dns.GetHostEntry(Dns.GetHostName());
IPAddress[] addr = ipEntry.AddressList;
TcpListener tcpListener = new TcpListener(addr[0], port);
tcpListener.Start();

ModbusSlave slave =ModbusTcpSlave.CreateTcp(slaveID, tcpListener);
slave.ModbusSlaveRequestReceived += new
EventHandler<ModbusSlaveRequestEventArgs>(Modbus_Request_Event);
slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();
lave.DataStore.DataStoreWrittenTo += new
EventHandler<DataStoreEventArgs>(Modbus_DataStoreWriteTo);
slave.Listen();
```

### 回傳值

無回傳值。

## 3.9.CoilDiscretes [DO 資料陣列]

DO 的資料陣列。位址為 1-65535。

### 語法

C#

```
ModbusDataCollection<bool> CoilDiscretes { get; private set; }
```

## 範例

```
[C#]  
slave.DataStore.CoilDiscretes[1] = true;  
slave.DataStore.CoilDiscretes[65535] = false;
```

### 3.10. InputDiscretes [DI 資料陣列]

DI 的資料陣列，可將裝置的 DI 數值儲存至此。位址為 1-65535。

## 語法

```
C#  
ModbusDataCollection<bool> InputDiscretes { get; private set; }
```

## 範例

```
[C#]  
slave.DataStore.InputDiscretes[1] = true;  
slave.DataStore.InputDiscretes[65535] = false;
```

### 3.11. HoldingRegisters [AO 資料陣列]

AO 的資料陣列。位址為 1-65535。

## 語法

```
C#  
ModbusDataCollection<ushort> HoldingRegisters { get; private set; }
```

## 範例

```
[C#]  
slave.DataStore.HoldingRegisters[1] = 222;  
slave.DataStore.HoldingRegisters[65535] = 333;
```

### 3.12. InputRegisters [AI 資料陣列]

AI 的資料陣列，可將裝置的 AI 值儲存至此。位址為 1-65535。

#### 語法

C#

```
ModbusDataCollection<ushort> InputRegisters { get; private set; }
```

#### 範例

[C#]

```
slave.DataStore.InputRegisters[1] = 222;  
slave.DataStore.InputRegisters[65535] = 333;
```

# 4. 通用 API

## 4.1. Dispose

對已定義的應用程序執行釋放或重設相關聯沒有應用的資源。

### 語法

```
C#  
void Dispose()
```

### 變數

無。

### 回傳值

無回傳值。

### 範例

```
[C#]  
string ipAddress = "10.0.0.69"; //use TCP master for example  
int tcpPort = 502;  
TcpClient tcpClient = new TcpClient(); //Create a new TcpClient object.  
tcpClient.BeginConnect(ipAddress, tcpPort, null, null);  
ModbusIpMaster master = ModbusIpMaster.CreateIp(tcpClient);  
master.Dispose();
```

# 附錄 A：錯誤訊息

## 對應表

以下為 NModbus 錯誤代碼對應表。

代碼	名稱	說明
01	不支援的功能	收到不被允許的指令碼。
02	不合法的位址	收到不正當的位址。
03	不合法的數值	收到不正確的數值。
04	Slave 裝置失效	要回復要求給 master 時，發生無法復原的錯誤。
05	確認(命令執行中)	當 master/slave 需要一段時間處理收到的要求時，會發出此代碼以避免發生逾時錯誤。
06	Slave 裝置忙碌	當 master/slave 正在處理長時間的要求時，對方必須等到處理完畢後再傳送訊息。
08	記憶體同位錯誤	當 master/slave 要讀取記錄檔時，偵測到記憶體同位錯誤。
0A	無效的 Gateway	Gateway 配置錯誤或過載。
0B	目標裝置 Gateway 沒有回應	目標裝置沒有回應。通常表示裝置目前不在線上。

## 範例

[C#]

```
//use TCP master for example
```

```
try
```

```
{
```

```
    #region Master to Slave
```

```
    //read AI, AO, DI, DO
```

```
    #endregion
```

```
}
```

```
catch (Exception exception)
```

```
{
```

```
    //Connection exception
```

```
    //No response from server.
```

```
    //The server maybe close the connection, or response timeout.
```



```

if (exception.Source.Equals("System"))
{
    NetworkIsOk = false;
    Console.WriteLine(exception.Message);
    this.Text = "Off line " + DateTime.Now.ToString();
    dtDisconnect = DateTime.Now;
}
//The server return error code.You can get the function code and exception code.
if (exception.Source.Equals("\nModbusPC"))
{
    string str = exception.Message;
    int FunctionCode;
    string ExceptionCode;

    str = str.Remove(0, str.IndexOf("\r\n") + 17);
    FunctionCode = Convert.ToInt16(str.Remove(str.IndexOf("\r\n")));
    Console.WriteLine("Function Code: " + FunctionCode.ToString("X"));

    str = str.Remove(0, str.IndexOf("\r\n") + 17);
    ExceptionCode = str.Remove(str.IndexOf("-"));
    switch (ExceptionCode.Trim())
    {
        case "1":
            Console.WriteLine("Exception Code: " + ExceptionCode.Trim() + "----> Illegal
function!");
            break;
        case "2":
            Console.WriteLine("Exception Code: " + ExceptionCode.Trim() + "----> Illegal data
address!");
            break;
        case "3":
            Console.WriteLine("Exception Code: " + ExceptionCode.Trim() + "----> Illegal data
value!");
            break;
        case "4":
            Console.WriteLine("Exception Code: " + ExceptionCode.Trim() + "----> Slave device
failure!");

```

```
        break;
    }
}
```

## 附錄 B：AI, AO 型別轉換

AI,AO 的輸出及輸入值皆為 ushort 型別。下列為 ushort 及 float 之間轉換的範例。  
若您需要更多型別轉換的範例，請至以下連結下載範例參考。

[http://ftp.icpdas.com/pub/cd/8000cd/napdos/modbus/nmodbus/demo/c%23.net/convertdatatype\\_c%23/](http://ftp.icpdas.com/pub/cd/8000cd/napdos/modbus/nmodbus/demo/c%23.net/convertdatatype_c%23/)

### Ushort(Int16) to float(Int32)

[C#]

```
//Convert ushort array to Float
ushort[] data = new ushort[2] { 59769, 17142};
float[] floatData = new float[data.Length / 2 ];
Buffer.BlockCopy(data, 0, floatData, 0, data.Length * 2);
for (int index = 0; index < floatData.Length; index ++)
{
    //print out the value
    Console.WriteLine(floatData[index / 2].ToString("0.0000")); //123.4560
}
Console.ReadLine();
```

### Float(Int32) to ushort(Int16)

[C#]

```
//Convert Float to short
ushort[] uintData = new ushort[2];
floatData = new float[1] { 223.4560f };
Buffer.BlockCopy(floatData, 0, uintData, 0, 4);
for (int index = 0; index < uintData.Length; index++)
{
    //uintData[0] = 29884; uintData[1] = 17247
    Console.WriteLine(string.Format("uintData[{0}] = {1}", index, uintData[index]));
}
Console.ReadLine();
```