

NModbus API 手冊

版本 1.1, 2013.8

Written by Renee Lin

目錄

目錄	2
1. 關於手冊	4
2. NModbus API 函數	5
2.1. Master API	5
2.1.1. CreateRtu	5
2.1.2. Createlp	6
2.1.3. CreateAscii	7
2.1.4. WriteSingleCoil	7
2.1.5. ReadCoils	8
2.1.6. ReadInputs	9
2.1.7. WriteSingleRegister	10
2.1.8. ReadHoldingRegisters	11
2.1.9. ReadInputRegisters	12
2.1.10. ReadTimeout[屬性]	13
2.1.11. Retries [屬性]	13
2.2. Slave API	15
2.2.1. CreateRtu	15
2.2.2. CreateTcp	16
2.2.3. CreateAscii	17
2.2.4. Listen	18
2.2.5. ModbusSlaveRequestReceived[事件]	18
2.2.6. CreateDefaultDataStore	19
2.2.7. DataStoreWrittenTo[事件]	19

2.2.8.	CoilDiscretes [DO 資料陣列].....	20
2.2.9.	InputDiscretes [DI 資料陣列].....	20
2.2.10.	HoldingRegisters [AO 資料陣列].....	20
2.2.11.	InputRegisters [AI 資料陣列].....	21
2.3.	通用 API.....	22
2.3.1.	Dispose.....	22
附錄：	錯誤訊息.....	23

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](#)
- b. WinCE 版本
 - [DLL 和文件](#) : nModbusCE.dll, CABC.dll, FC19.dll
 - [範例程式](#)

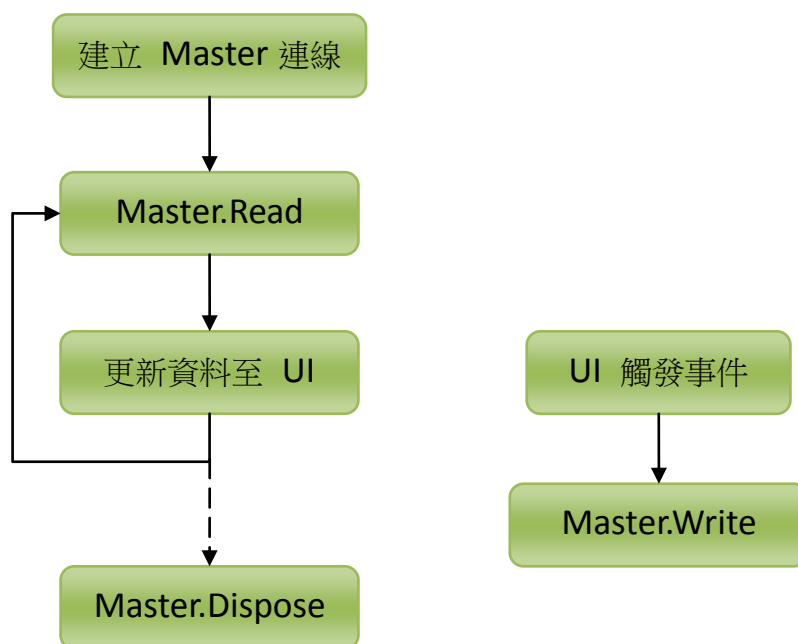
了解更多有關 Modbus→http://www.icpdas.com/products/PAC/i-8000/modbus_c.htm

適用 NModbus 開發的 PAC

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

2.NModbus API 函數

2.1.Master API



2.1.1.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.Open();  
ModbusSerialMaster master = ModbusSerialMaster.CreateRtu(serialPort);
```

2.1.2.CreateIp

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);
```

2.1.3. 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.Open();  
ModbusSerialMaster master = ModbusSerialMaster.CreateAscii(serialPort);
```

2.1.4. 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);
```

2.1.5.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);
```

2.1.6.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);
```

2.1.7. 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);
```

2.1.8. 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[]。

Examples

[C#]

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

2.1.9. 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);
```

2.1.10. 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
```

2.1.11. 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

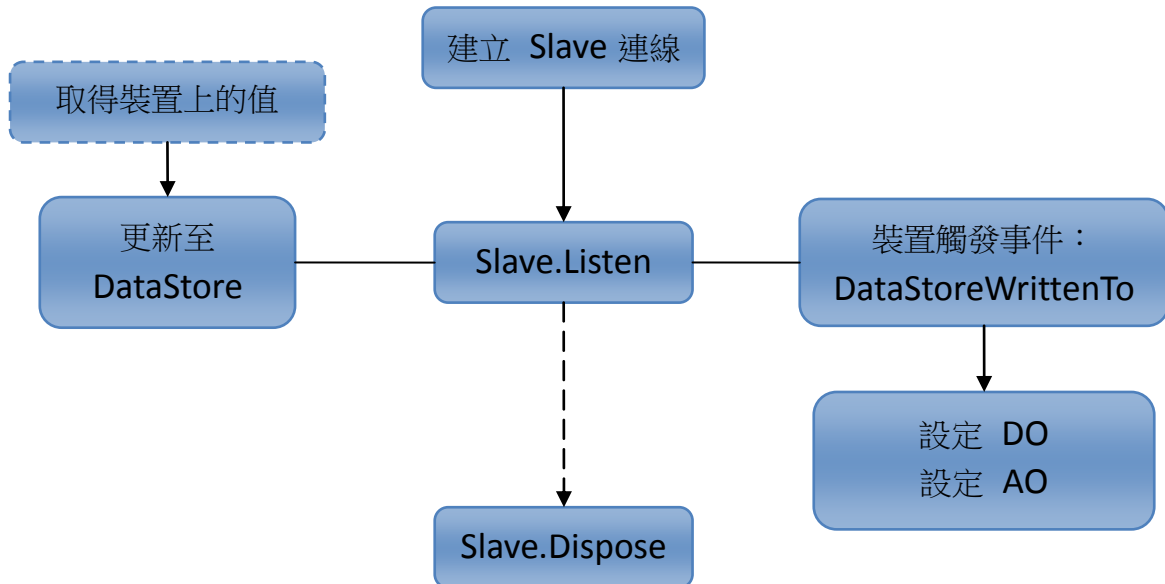
[C#]

```
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

Retries = 0 表示不需要重試。

2.2.Slave API



2.2.1.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.Open();
ModbusSlave slave = ModbusSerialSlave.CreateRtu(slaveID, serialPort);
```

2.2.2. CreateTcp

建立 Modbus slave TCP 的連線。

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);
```

2.2.3. 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.Open();
```

```
ModbusSlave slave = ModbusSerialSlave.CreateAscii(slaveID, serialPort);
```

2.2.4.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.Listen();
```

回傳值

無回傳值。

2.2.5.ModbusSlaveRequestReceived[事件]

當 slave 收到要求的時候觸發事件。

語法

```
C#
```

```
EventHandler<ModbusSlaveRequestEventArgs> ModbusSlaveRequestReceived
```

範例

```
[C#]
```

```
slave.ModbusSlaveRequestReceived += new  
EventHandler<ModbusSlaveRequestEventArgs>(Modbus_Request_Event);
```

2.2.6.CreateDefaultDataStore

將 AO,AI 預設為 0，DO,DI 預設為 false。

語法

```
C#  
DataStore CreateDefaultDataStore()
```

範例

```
[C#]  
Slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();
```

回傳值

回傳 DataStore。

2.2.7.DataStoreWrittenTo[事件]

當 slave 的 DataStore 被 master 之命令寫入時觸發事件。

語法

```
C#  
EventHandler<DataStoreEventArgs> DataStoreWrittenTo
```

範例

```
[C#]  
slave.DataStore.DataStoreWrittenTo += new  
EventHandler<DataStoreEventArgs>(Modbus_DataStoreWriteTo);
```

2.2.8.CoilDiscretes [DO 資料陣列]

DO 的資料陣列。

語法

C#

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

範例

[C#]

```
slave.DataStore.CoilDiscretes[0] = true;  
slave.DataStore.CoilDiscretes[1] = false;
```

2.2.9.InputDiscretes [DI 資料陣列]

DI 的資料陣列，可將裝置的 DI 數值儲存至此。

語法

C#

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

範例

[C#]

```
slave.DataStore.InputDiscretes[0] = true;  
slave.DataStore.InputDiscretes[1] = false;
```

2.2.10. HoldingRegisters [AO 資料陣列]

AO 的資料陣列。

語法

C#

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

範例

```
[C#]  
slave.DataStore.HoldingRegisters[0] = 222;  
slave.DataStore.HoldingRegisters[1] = 333;
```

2.2.11. InputRegisters [AI 資料陣列]

AI 的資料陣列，可將裝置的 AI 值儲存至此。

語法

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

範例

```
[C#]  
slave.DataStore.InputRegisters[0] = 222;  
slave.DataStore.InputRegisters[1] = 333;
```

2.3.通用 API

2.3.1.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 master tcp 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：錯誤訊息

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	無效的閘道	閘道配置錯誤或過載。
0B	目標裝置閘道沒有回應	目標裝置沒有回應。通常表示裝置目前不在線上。