

第五章：Modbus 通信最新发展

1 OPC Server 及 Client 系统

1.1 何谓 OPC

OPC 为 OLE for Process Control 的缩写。是工业界最先进的资料交换标准。回顾自动控制系统的发展，无论是 DCS、PLC、监控套装或控制器等系统，都会面临系统整合的瓶颈。各系统无法有效共享资料，对于相同的控制设备，每个厂牌的监控系统都要设计一套自有的通信 Driver。当 MIS 系统需要现场实时信息时，也需要一连串复杂的信息接口才能达到所要的功能。面对以上种种状况，由一群工业自动化领导厂商，诸如 Fisher-Rosemount、Rockwell Software、Opto32、Intellution、Intuitive Technology 及 Microsoft 等，于 1996 年共同成 OPC Foundation，希望建立一个业界标准接口规格。由目前信息使用的发展方向来看，大多数的 client 端使用的系统为 MS Windows，故使用 OLE 的架构有利与大多数的使用者整合，且利用现有的 COM、DCOM、ActiveX 等技术，可快速将系统建立完成，现有系统所要变动的项目最少等优点。故当 OPC 架构提出后，不到数年间已成为业界的标准，且成为自控系统都必须具备的通信功能。

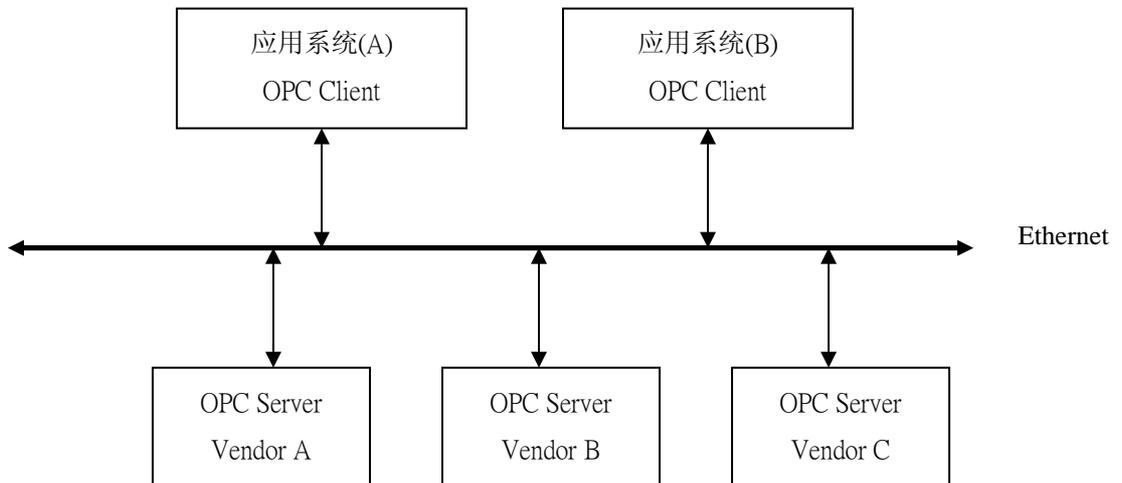
1.2 OPC 基本架构

对于工厂自动化信息架构可分为三个主要层面：

- Plant Floor：仪表信号、现场控制器、各种资料收集设备等。此层次有各种的现场运转信息，设定系数、各种 command 等信息，需要与上层系统作资料交换。
- Process Floor：此为 DCS、SCADA 等统合控制及操作系统，此控制结果要下传至现场设备，也要将执行结果上传至 MIS 系统。
- Business Floor：全公司或全厂的管理系统，需要现场实时信息做回馈，也要将生产排程计划等资料下载至下层系统。系统会使用 Database、Spreadsheet、MES 套装等工具或系统。

OPC 就是一套介于以上三层系统间的通信标准接口规格。只要依据此接口规格所开发的软件系统，就可以做到系统间或设备间的资料交换，而不会受到不同系统或设备的限制。

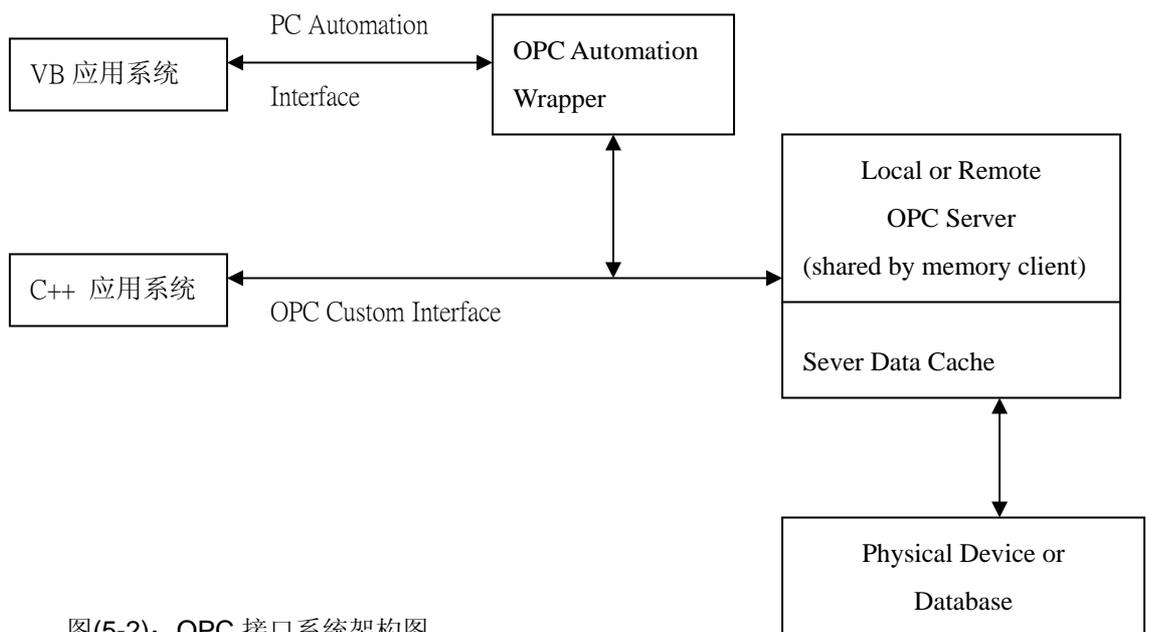
应用系统架构图：



图(5-1)：OPC 接口基本架构图

- OPC Server 可依需求设计而提供此方面的服务，例如连接不同的自控设备，或特殊的数据处理及计算功能。
- OPC Client 可同时连接数个 OPC Server 以存取所要的资料。应用系统可为具有 OPC Client 的套装监控系统，或自行开发的报表系统。
- 符合 MS COM、DCOM 架构，可以取入网络内任一个 OPC Server 的服务。
- 由 OPC Client 的角度来看，可达到下列的利益：
 - Client 的开发者不需考虑 Driver 的设计，此由 OPC Server 厂商提供此服务。
 - 不需重复投入 Driver 的开发，只须向市场上购买该 Driver 的 OPC Server 即可。
 - 即使不同操作系统或自控设备也可通信，只要符合 OPC 接口规格。
 - 当自控设备硬件改变时，只要更新 OPC Server 即可。符合使用者的最大弹性需求。
 - 以往两个监控系统，不可能同时对同一个控制设备做通信，如今透过同一窗口 OPC Server 即可做到。
 - 目前有太多的应用软件开发人员，使用 VB、Delphi、Power Builder 等等，以往对于存取现场实时信息束手无策，如今藉由 OPC Server 的联机达到完美的地步。
- OPC 如同一条无形的线连接自控硬设备与应用软件系统，也可将 OPC Server 视同对象，买来即可使用，非常便利。

OPC 接口架构图：



图(5-2): OPC 接口系统架构图

系统资料的交换可分为两大部分，一为与自控设备的连接，此依不同设备有不同的通信协议，需要设计不同的 OPC Server 与其相对应。另一为与 Server 与 Client 间的通信，此部份既为 OPC 的 COM 接口规格。如同大部分 COM 的设计方式，OPC 也是 client/server 架构。

其中 Client 要顾及大部分应用系统的开发者，简单易用、快速上线等原则，多数开发工具都具备 OPC Client 功能。另外 Server 方面需要应付多数个 Client 的资料要求，且要面对不同自控硬件而有不同的通信协议，故要以 C++ 开发工具才可掌控所有的系统设计功能，都以专业公司设计各种 OPC Server，以提供使用者选择。

1.3 OPC 主要功能

OPC 主要功能是工业界一种资料交换的通信接口标准。对于第一阶段的功能为三大类：Online Data Access、Alarm and Event Handling、Historical Data Access 等。其它功能如：Security、batch、Historical alarm and event data access 等属于下一次版本。

OPC 架构为 COM 的接口，希望能达到原则为：简单易用、适合多数厂商所需、提供高级语言的功能、有效率的资料交换等。所以 OPC 规格包含：一组 custom COM Interface 以及 OLE Automation Interface 以供 VB、Excel 等使用。并且利用 DCOM 技术以作为 Remote Client 的接口。

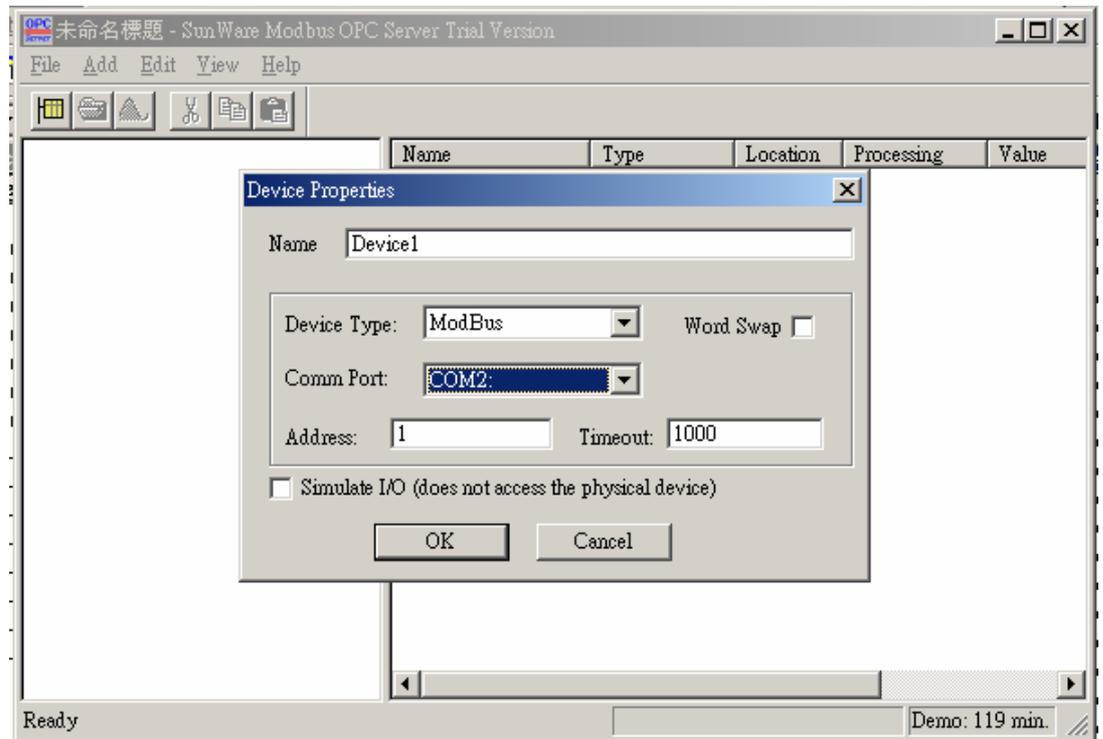
- Online Data Access: 此为最基本的功能，也是绝大多数应用程序的需求。此包含 Device、Group、Item 等三种层次的对象，以对应一个现场的信号点。Device 定义针对某一个特定的自控设备，具有该系统的通信协议。Group 为某一特定组合的资料，例如：Group 代表一个特定的操作画面或报表的数据群。最后 Item 就对应某一个信号点，此信号点依据控制设备的规格可以读取或写入，又信号种类可有不同的资料型式，例如：DI/O 以 Boolean 表示、AI/O 以整数或实数表示。OPC Client 可设定通信间隔的 Polling Rate 以因应资料的更新速率。每一个 OPC Item 具有 Value、Quality 及 Time Stamp 等三种数值。Value 为 VARIANT 资料型态，表示信号真正数值可为实数、整数、Boolean 或字符串等型式。Quality 表示该信号的品质，例如：通信断线时，品质为 Bad。Time Stamp 纪录该信号取入的时间。
- Alarm and Event Handling: 此方面的 Interface 是定义，当有特定的 Alarm 或 Event 发生时，会被通知的 OPC Client。同时也允许 OPC Client 对 Server 做设定，以决定何种状态需要被通知。发生不正常的状况称为 Alarm，例如：某一点 AI 发生超上限或超下限，或者某一个马达油位不足跳脱的 DI Alarm 等。Event 是发生一件被定义的事件，需要 OPC client 被告知，例如：操作员激活主马达，控制系数被修改，系统发生不正常状况等。IOPCEventServer Interface 提供下列 OPC Client 使用方

法:

- 定义 OPC Event Server 所要提供的 Alarm 及 Event 的型式。
- 将定义的 Alarm 及 Event 登录于 Server 上, 并具有过滤(Filter)功能以取入特定的 Event 等。
- Server 如何处理所定义的 Alarm 及 Event。
- Server 提供查询 Alarm 及 Event 的相关画面。
- Historical Data Access: 目前大多数的历史趋势资料, 都以自订的规格提供资料, 各系统间并无兼容性。故需要再定义共同的规范, 以做到随插随用的环境。为了适合各方面使用者的不同需求需要考虑到下列资料型式:
 - Simple Trend data servers: 只提供单纯的原始资料, 也就是由 Online Data Access server 传过来的资料, 包含 Time Value and Quality。
 - Complex data compression and analysis servers: 除了原始资料的储存外, 更提供资料加工, 例如: 各种统计分析功能、资料比较等, 也提供资料批注的功能, 更为便利使用。

1.4 Modbus OPC Server Demo System

本书提供 Modbus 及 Modbus/TCP 等两种 OPC Server 试用版 (Trial Version), 符合 OPC Data Access V1.0 and V2.0 规范, 可以执行 2 小时的联机, 当 Demo 时间到时, 就不做联机功能。以下说明 OPC Server 的操作方法, 并且将利用此 OPC Server 进一步说明, 如何以 VB 程序设计一个 OPC Client 功能。

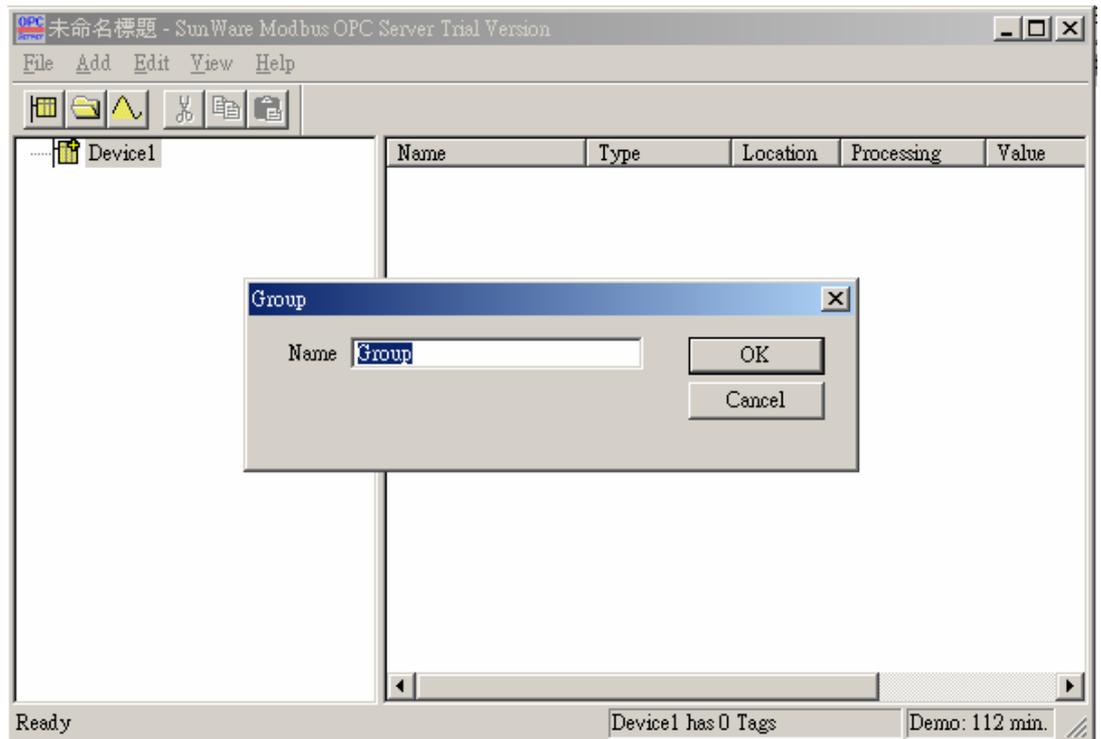


图(5-3): OPC Server Device Properties Dialog 画面

首先选择 Menu [Add][New Device], 显示出 Device Properties Dialog Box 如上图所示, 以建立一个新的 Device。

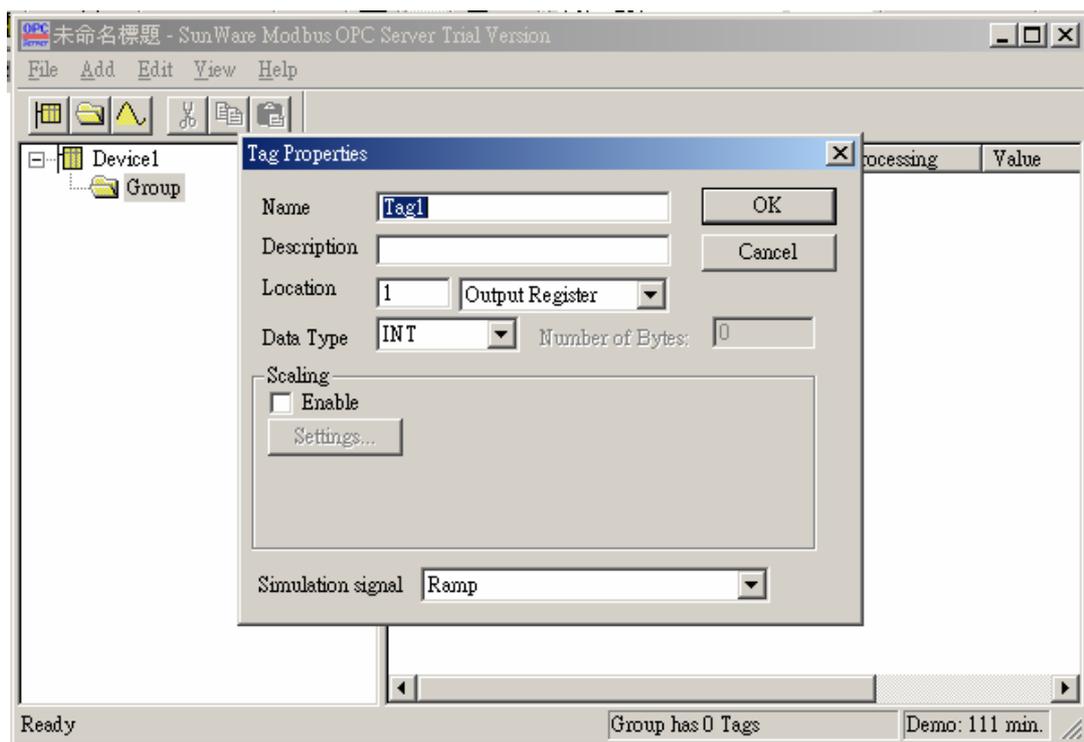
Dialog Box 的各字段说明如下:

- Name: Device Name 必须是唯一的
- Device Type: 联机的 Modicon PLC Type 或 Modbus Device
- Comm Port: 所连接的 Comm Port 总共有 8 个 Port
- Address: Modbus Address 也就是每个 Modbus Device 的 ID No.
- Timeout: 通信超时时间, 超过此时间表示通信断线
- Simulation: 选择此项目后, OPC Server 不会实际通信, 提供仿真数据, 此可以用来当未接实际 Modbus 设备时, 显示仿真信号值提供给其它 OPC Client 程序联机测试用。
- Word Swap: 当所接到的 Modbus 资料, 其 16 bit integer 依据 Protocol 规定为 High Byte 在前, Low Byte 在后的顺序, 但是某些 Modbus 设备所传送的资料格式相反, 此利用此选项修正之。



图(5-4): OPC Server Group Define Dialog 画面

然后选择 Menu [Add][New Group], 显示出 Group Dialog Box 如上图, 输入 Group Name。



图(5-5): OPC Server TAG Properties Define Dialog 画面

然后再选择 Menu [Add][New Tag]，显示出 Tag Properties Dialog Box 如上图，以建立一个新的 TAG。

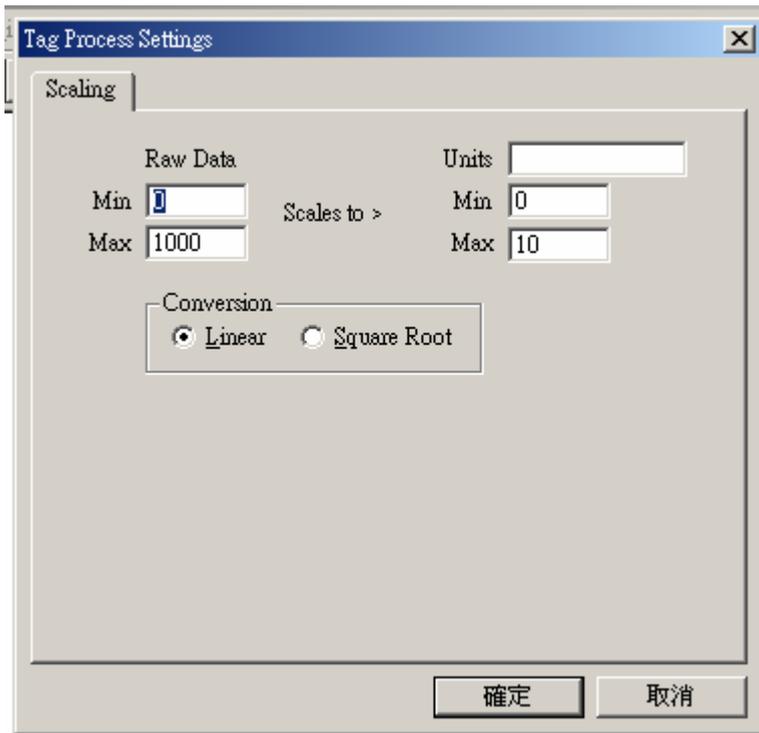
Dialog Box 的各字段说明如下：

- Name: TAG Name 于此 Group 内，此 Name 必须是唯一的。
- Description: 此 TAG 的说明，提供使用者更多的说明叙述。
- Location and Type: 该 TAG 的 Modbus 信号 Type，具有 Input Coil、Output Coil、Input Register、Output Register 等四种。Location 表示该 TAG 的地址。
- Data Type: 此 TAG 的资料型态。
- Number of bytes: 所具有的 Byte 数
- Scaling: 原始信号值是否需要再做工程数值 Range 的转换，详见下图
- Simulation signal: 选择仿真计算方法。有 Sin、Ramp、Random 等三种方法。

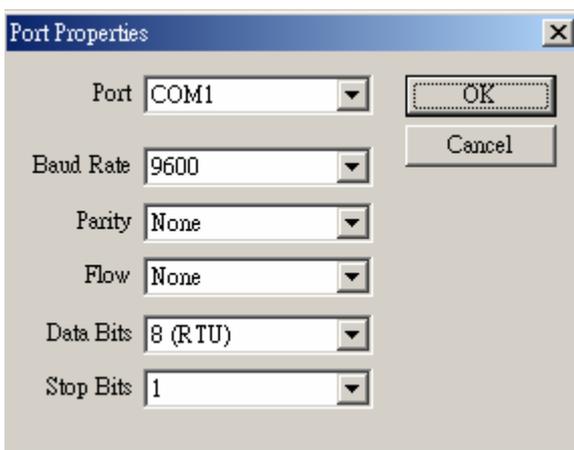
Scaling 画面说明：Range 转换的设定画面

- Raw Min: 原始信号值 Range Low
- Raw Max: 原始信号值 Range High
- Unit: 工程单位
- Min: 信号工程值 Range Low
- Max: 信号工程值 Range Low

- Conversion: 转换用公式。有 Linear、Square Root 等两种。

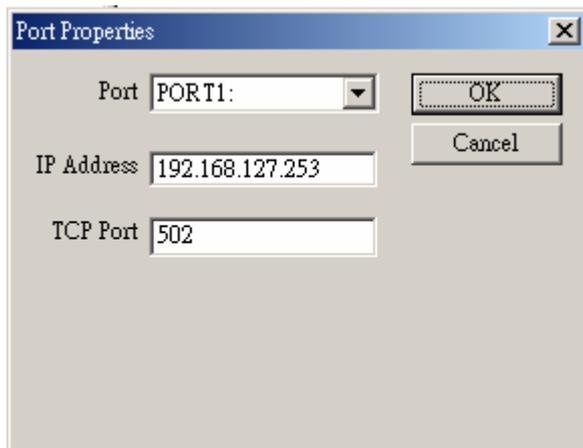


图(5-6): 信号工程单位转换定义画面



图(5-7): COM Port Define 画面

选择 Menu [Edit][Port]显示左边画面
 Modbus 的 COM Port 系数设定
 Port: 通信埠, 共有 COM1 – COM8
 Baud Rate: 通信速度
 Parity: None、Odd、Even
 Flow: RTS、CTS 流程控制
 Data Bits: 资料位
 Stop Bits: 停止位



选择 Menu [Edit][IP Address & Port]显示
左边画面

Modbus/TCP 的 TCP/IP 设定

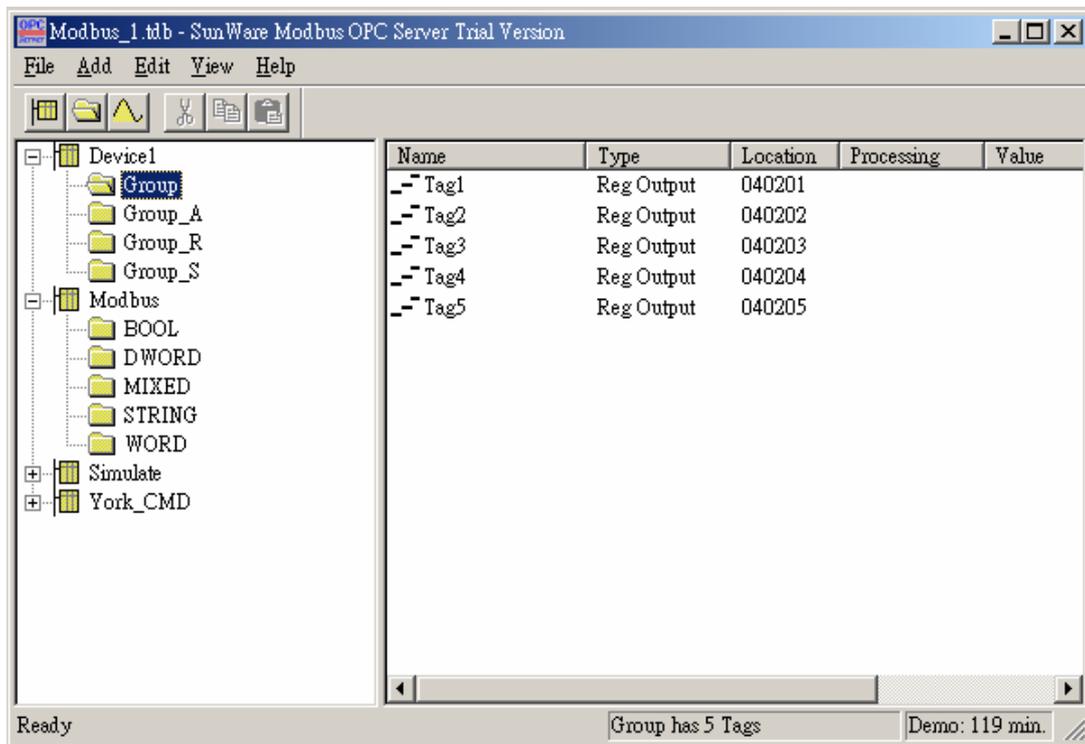
Port: TCP/IP 通信埠, 供有 Port1 – Port8

IP Address: 所要连接 Remote Modbus

Device IP Address

TCP Port: Modbus 默认值为 502

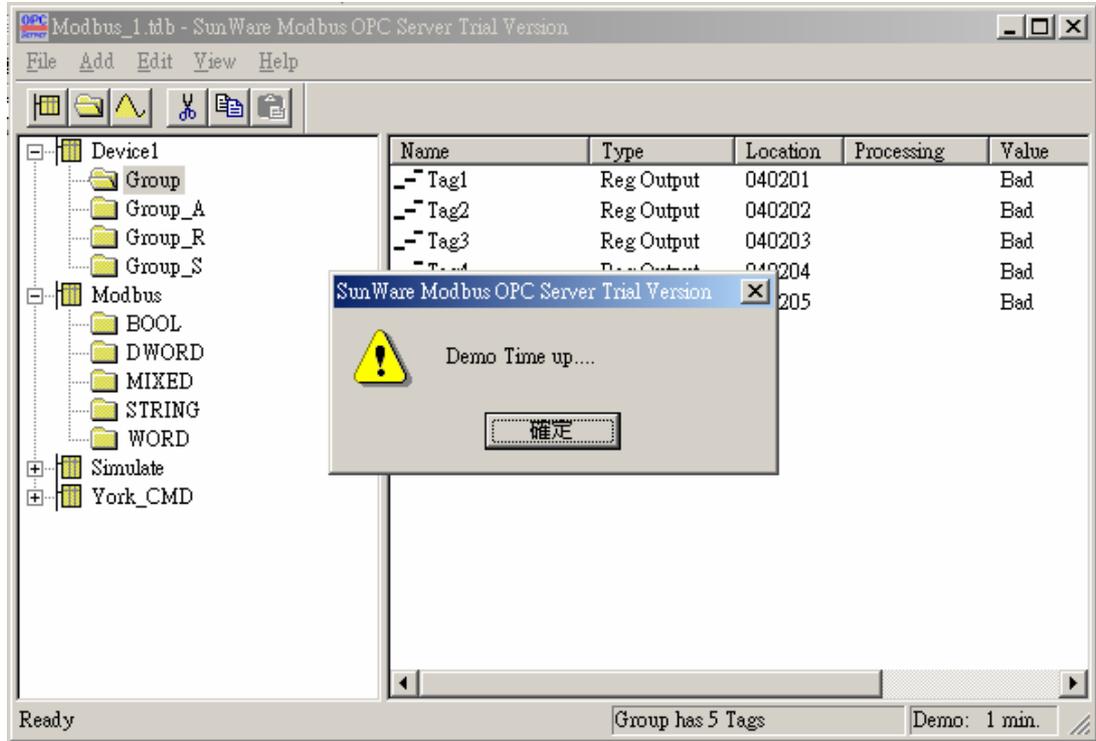
图(5-8): IP Address & Port Define 画面



图(5-9): OPC Server 操作画面

依据以上的操作方式，增加 Device、Group、Tag 等。如同上图例显示。

画面右下角的 Status Bar 显示，尚余的 Demo 时间（最大为 120 分钟）。当 Demo 时间到，会显示如下画面，同时不再通信，TAG 的 Quality Flag 会变成 Bad。



图(5-10): OPC Server Demo Time up 操作画面

然后选择 Menu [View][Monitor]就可以对外通信取入信号值。此功能用于确认所设定的内容是否正确，以及立刻通信以确认信号值是否正确。

设定完成后，选择 Menu [File][Save]将设定内容，存成一个 Configuration File 其附档名为 tdb。然后停止 OPC Server。

正式使用时，也就是当其它监控系统使用 OPC Client 功能 connect 此 OPC Server 后。OPC Server 会被激活于 Background 状态，主画面不会显示，此时 OPC Server 依据 OPC Client 所联机的 TAG 开始通信。注意点：如果 OPC Server 的 Configuration File 设定有 100 个 TAG，但是某一个 OPC Client 联机其中 10 个，此时 OPC Server 只就此 10 个 TAG 通信而不会 100 个全部通信。当有第二个 OPC Client 再联机其它 20 个 TAG 时，此时 OPC Server 再增加 20 个 TAG 的实际通信。如此系统运作弹性因应，以保持最佳反应状态。当所有的 OPC Client 都不联机后，OPC Server 自动停止程序。

注意外界要使用某一个 OPC TAG 必须使用全名，例如：Device1.Group.Tag1。各各名字中间要以逗号「.」相区隔。

主要 Menu 功能介绍：

[File][New]: 建立一个新的 Configuration File。

[File][Open]: 打开一个 Configuration File。

[File][Save]: 存入一个 Configuration File。

[File][Save As]: 存入一个 Configuration File, 使用其它 File Name。

[File][Export CSV]: 将 Configuration File 档转成 CSV 格式存盘, 因为 Excel 可读取 CSV 档, 所以非常方便将所设定的 TAG Name, Loaction 等转给其它监控系统使用。

[File][Reexport CSV]: 将同一个 Configuration File 档再转成 CSV 格式存盘一次。

[File][Import CSV]: 将一个与上述 CSV 档相同格式的内容, 转成 Configuration File。同样道理, 此种方式可由其它监控系统的 TAG 转成 CSV 档, 然后由 Excel 修改内容及格式, 再利用此功能转换进来成为 OPC Configuration File。

[File][Reimport CSV]: 将同一个 CSV 档再转成 Configuration File 一次。

[Add][New Device]: 建立一个新的 Device。

[Add][New Group]: 建立一个新的 Group。

[Add][New Tag]: 建立一个新的 Tag。

[Add][Multiply]: 对于一个 Copy Tag 可以建立多数个复制 TAG。

[Edit][Cut]: 取消一个 TAG, 可供 Paste 用。

[Edit][Copy]: 复制一个, 可供 Paste 用。

[Edit][Paste]: 贴上一个 TAG。

[Edit][Delete]: 删除一个。

[Edit][Ports...]: 定义 COM Port 或 IP Address 等通信系数。

[Edit][Properties]: 进入 Device、Group、Tag 等 Properties Dialog Box。

[View][Monitor]: 针对某一个选定的 Group 内 Tag 对外做通信。

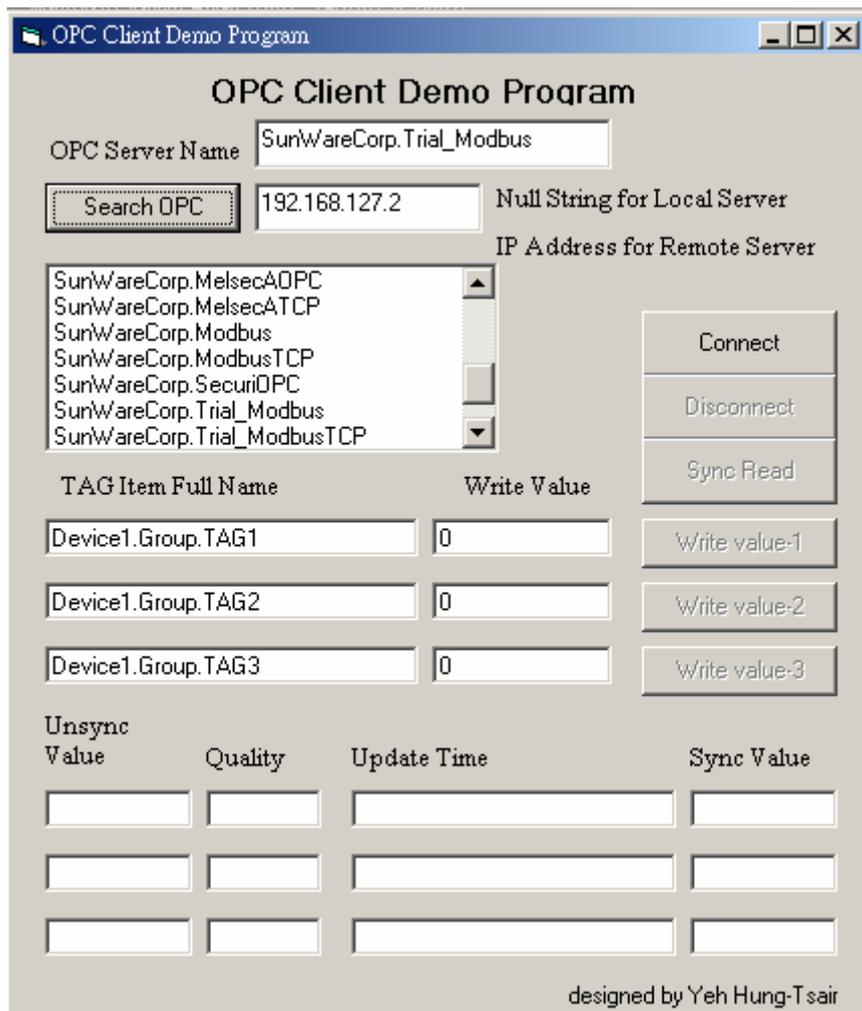
[View][Status Bar]: 显示或不显示 Status Bar。

1.5 VB 的 OPC Client Demo 程序功能介绍

于系统整合上, 有许多情形必须取得工厂的运转信号点, 但是几乎所有的信号点都是存在于 PLC、DCS、控制器等, 而且每种通信协议都不一样, 以前使用 VB 直接与控制系统相连接, 设计通信 Driver, 并不是一个好办法, 且效率差。本书提供 OPCClient_Demo VB 程序, 可以连接 OPC Data Access V1.0 and V2.0 Server。读者以此为范例, 可以延伸设计出实用的程序。

本程序使用 VB 6.0 设计, 具有下列主要功能。

- 寻找 Local or Remote Server 的所有已经安装完成的 OPC Server。
- 输入 TAG Item 全名后, 以同步 (Synchronize) 或异步 (Unsynchronize) 两种方式读取 TAG 值。
- 显示同步读取时的 TAG 值, 最新时间及信号值品质 (Quality) 等。
- 可输出 TAG 值至 OPC Server 然后输出至控制设备。



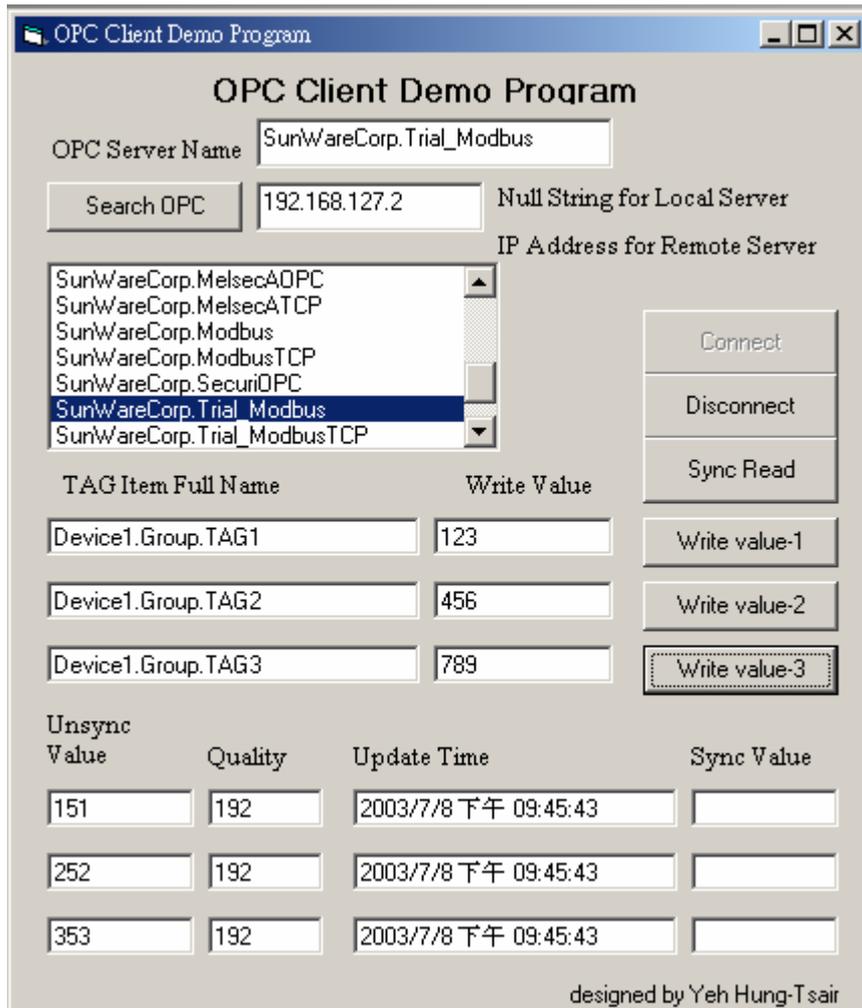
图(5-11): OPCClient_Demo 初期画面

要使用 OPCClient_Demo 前, 请先将两个 Trial Version 的 OPC Server 安装完成。OPCClient_Demo 操作步骤如下说明:

- 上图为程序激活后的画面, 首先输入 Remote Server 的 IP Address 或者将 IP Address 值清除表示要连接 Local Server。然后按「Search OPC」Button, 此时会显示该 Server 所有安装完成的 OPC Server Name。例如: 本书的两个 OPC Server Name 为:「SunWareCorp.Trial_Modbus」及「SunWareCorp.Trial_ModbusTCP」。
- 由 List Box 点选要联机的 OPC Server Name, 会出现在最上面第一字段上。
- 于 TAG Item Full Name 字段输入 OPC Tag 全名。然后按「Connect」Button 连接之。程序出现讯息 Box, 如果联机成功后, 其它 Button 变成可以选取。本程序 Demo 三个 TAG 的联机, 如果有任何一个 TAG Name 不对则会有错误讯息产生。
- 当联机成功, 如果 OPC Tag 值有任何变化, 会马上显示于 Unsync

Value、Quality、Update Time 等三个字段。也就是 OPC Server 主动通知所联机的 OPC Client 称为异步读取。

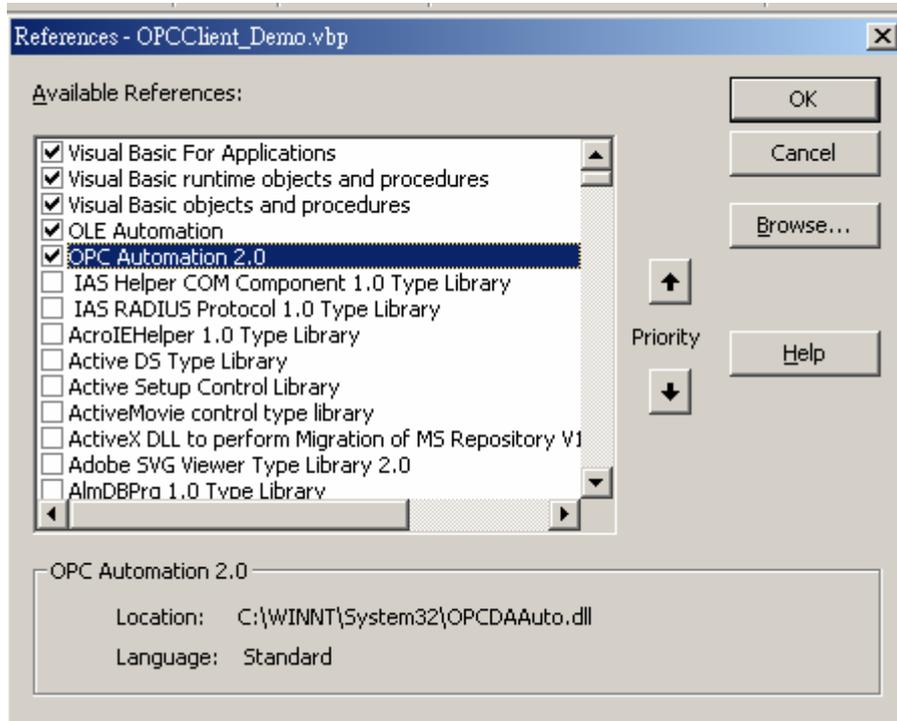
- 按「Sync Read」Button，会由 OPC Client 主动马上读取 OPC Tag 值，然后显示于 Sync Value 字段上。此种方式适用于 OPC Client 立刻要取得目前最新值得情形。
- 当要输出某一个 Tag 值时，于 Write Value 对应字段输入 Tag 值，然后按「Write Value-1」、「Write Value-2」、「Write Value-3」等 Button。此 Tag 值会传送至 OPC Server，然后经过 OPC Server 的实际通信输出至控制系统上。
- 输出时必须注意此 Tag 不是 Read Only 性质。此种 Tag 依据不同控制系统而有不同的定义方式。例如：Modbus 而言，只有 Output Coil、Output Register 等两种才可以写入 Tag 值。
- 当 connect 成功时，Demo 程序会建立 Server connect、Group、Item 等三种 OPC 系统用联机程序 Handle 等。故要脱离联机时按「Disconnect」Button，或者 Demo 程序结束时，就要将这些 Handle 全部 Remove。此时 OPC Server 接到此命令，就将与此 OPC Client 相连的相关资料全部删除。如果是最后一个 OPC Client 做 Disconnect 要求，OPC Server 就会自动停止程序。



图(5-12): OPCClient_Demo 操作结果画面

1.6 如何以 VB 设计 OPC Client 程序

建立一个新的 VB Project，首先选择 Menu [Project][References]进入组件 Dialog Box，选取 OPC Automation 2.0。如果看不到此选项，请于\winnt\system32 目录内找寻 OPCDAAuto.dll 增加之。如下画面所示：



图(5-13): 增加 OPC Automation 2.0 选取画面

使用OPC的接口API都于OPC Automation 2.0 组件内，其使用说明手册可以参考OPC Foundation网站 <http://www.opcfoundation.org> 下载OPC Data Access 2.0 规格的文件opcda20_auto.doc。

本书所提供的 OPCClient_Demo 程序，为介绍设计 OPC 的连接程序最主要的用法，以提供读者一个快速了解连接 OPC 使用方式。当然整体 OPC Client API 还有许多细部的用法，才能将一个实务应用程序设计得更完美，兹将 Demo 程序的主要程序内容说明如下：

- OPC 变量宣称：OPC 的变量都是一个对象（Object）。OPC 的规格提到与 OPC Server 的接口机制为三阶层架构，第一层为 OPC Server，第二层建立一个 Group 集合用来管理许多连接的 Group，第三层于每一个联机实际的 Group，建立此 Group 内的许多 Item 对应于 OPC Tag。注意要点：既然前面提到 OPC 变量为对象 object，所以具有对象的 properties、method、event 等三大要素，程序的设计就是使用这三大要素构成应用程序，至于这些三大要素的用法就参考 opcda20_auto.doc 内容设计。

Option Explicit

Option Base 1

'定义 OPC Interface 对象 object

Dim IconServer As OPCServer ' OPC server object 第一层 Server

Dim RemoteServer As OPCServer ' OPC server object 用于找寻 OPC Server list

Dim IconGroups As OPCGroups ' collection of groups 第二层 Group 集合

Private WithEvents IconGroup As OPCGroup ' individual group 各别联机的 Group

Dim IconItems As OPCItems 'collection of items 第三层于联机 Group 建立许多 Item

Dim OneItem_A As OPCItem 'Single item 再为每个 Write TAG 建立各别 Item

Dim OneItem_B As OPCItem 'Single item

Dim OneItem_C As OPCItem 'Single item

'item object 以下为处理 Item read/write 所需的变量

'Public OpcItem As IOPCItem

Dim ClientHandles(3) As Long

Dim ServerHandles() As Long

Dim Errors() As Long

Dim pQuality As Variant

Dim pTimestamp As Variant

Dim ItemIDs(3) As String

Dim AccessPaths(3) As String

Dim Active(3) As Boolean

Dim bServerStarted As Boolean

Dim bReadOn As Boolean

Dim ItemCount As Integer

Dim ReadValue() As Variant

- OPC Server List: 当使用者输入 Remote Server IP Address 或者将 IP Address 清成空白表示为 Local Server, 然后按 [Search OPC] Button 激活下列程序。因为 RemoteServer 已经宣称为 OPC 的 Server Object, 所以使用 GetOPCServers(Node) method 取入 OPC Server List。所得到的内容存于 AlLOPCServers 内, 此变量宣称为 Variant。最后以 for loop 将 AlLOPCServers 内容转存入 List Box 内。

Private Sub Command7_Click()

```

' get remote OPC

Dim AllOPCServers As Variant
Dim Node As Variant
Dim i As Integer
Set RemoteServer = New OPCServer
Node = Text2.Text
If Node = "" Then
    AllOPCServers = RemoteServer.GetOPCServers ' it is local server
Else
    AllOPCServers = RemoteServer.GetOPCServers(Node) ' it is remote server
End If

For i = LBound(AllOPCServers) To UBound(AllOPCServers)
    List1.AddItem AllOPCServers(i)
Next i

End Sub

```

- **Connect Server:** 首先以 IconServer 对象的 connect method 连接 OPC Server。联机成功后，取出 IconServer 的一种 properties 项目 OPCGroups，此为 Group 的集合。然后 Set IconGroup = IconGroups.Add("Group One") 建立联机 Group。本程序只建立一个“Group One”。再由联机的 IconGroup 取出 Item 集合对象，然后建立许多 Item。本程序建立三个 Item，给予每个 ItemID 要使用 OPC Tag 的全名，例如：device1.group.tag1。最后如果 Item 为可以输出的 TAG 则再建立专门输出用的 Item 对象。

```

Private Sub Command1_Click()
    Dim Node As String

' Create opc server object
Set IconServer = New OPCServer
' Establish connection to an OPC server
Node = Text2.Text
If Node = "" Then
    IconServer.Connect Form1.Text1 'connect local OPC Server
Else
    IconServer.Connect Form1.Text1, Node 'connect remote OPC server

```

```

End If

If TypeName(IconServer) = TypeName(Nothing) Then
    MsgBox "Connect Error" ' 无此 OPC Server
    Return
Else
    MsgBox "Connect OK"

End If

' 首先由此 Server 对象取入 group 集合对象
Set IconGroups = IconServer.OPCGroups

' 然后于此 group 集合对象增加一个要数据通信的 group 对象
' 要给该增加 Group 一个唯一的 Name
Set IconGroup = IconGroups.Add("Group One")
IconGroup.IsSubscribed = True

' 此 group 对象取入 Items 集合对象
Set IconItems = IconGroup.OPCItems
ItemCount = 3

' 于该 Items 集合对象, 设定所要通信的每个信号点 item
' 并且对每个 Item 要给予一个 ClientHandle 此为往后对区别各信号用
ItemIDs(1) = Form1.ItemID(1).Text
ClientHandles(1) = 1
ItemIDs(2) = Form1.ItemID(2).Text
ClientHandles(2) = 2
ItemIDs(3) = Form1.ItemID(3).Text
ClientHandles(3) = 3

'增加 Item 的设定于 "Group one"
IconItems.AddItem ItemCount, ItemIDs, ClientHandles, ServerHandles, Errors,
pQuality, pTimestamp

' 如果信号点为 AO or DO 则再设定单独的 one item 对象以 write data 用
Set OneItem_A = IconItems.GetOPCItem(ServerHandles(1))

```

```
Set OneItem_B = IconItems.GetOPCItem(ServerHandles(2))
Set OneItem_C = IconItems.GetOPCItem(ServerHandles(3))
```

```
' 检查 error code
If Errors(1) <> 0 Then
    MsgBox "Add Item Error"
    Return 'no server
Else
    MsgBox "Add Item OK"
End If
```

```
Command1.Enabled = False ' connect button
Command2.Enabled = True ' disconnect button
Command3.Enabled = True ' Sync Read button
Command4.Enabled = True ' Write value-1 button
Command5.Enabled = True ' Write value-2 button
Command6.Enabled = True ' Write value-3 button
```

```
End Sub
```

- Read Item: 读取 Item value 有两种方法。一种为 VB 程序主动读取称为同步方式, 使用 IconGroup.SyncRead method 既可。另一种为 OPC Server 检查 TAG value 有变化时主动通知称为异步方式。此方式必须建立一个 Event Subroutine 才可以使得 OPC Server 主动通知时, 进入此 Subroutine 处理。于程序最前端变量宣称时, 建立 Private WithEvents IconGroup As OPCGroup, 表示此 Group 会使用异步 Read。然后建立一个实际 Suboutine 其名字必须使用 Private Sub IconGroup_DataChange(……), 详见下面步骤:

```
Private Sub Command3_Click()
```

```
' 同步方式 Read data
```

```
IconGroup.SyncRead OPCCache, ItemCount, ServerHandles, ReadValue, Errors
If Errors(1) <> 0 Then
    MsgBox "Read Data Error"
    Return 'no server
```

```

Else

    MsgBox "Read Data OK"
    Form1.Value(14) = ReadValue(1)
    Form1.Value(15) = ReadValue(2)
    Form1.Value(16) = ReadValue(3)
End If

End Sub

' Data Change Event handler fired by group subscription callback
' 此由 OPC Server 对于该 OPC Client 所设定的信号点, 如果有数值改变
' 主动 call 此子程序, 将新的信号值传入
Private Sub IconGroup_DataChange(ByVal TransactionID As Long, ByVal NumItems
As Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long,
TimeStamps() As Date)

' MsgBox "CallBack Data OK"

Dim i, m, n As Integer

For i = 1 To NumItems
    If ClientHandles(i) > 0 Then
        Form1.Value(ClientHandles(i)) = ItemValues(i)
        m = 3 + ClientHandles(i)
        Form1.Value(m) = Qualities(i)
        n = 6 + ClientHandles(i)
        Form1.Value(n) = TimeStamps(i)
    End If
Next i

End Sub

```

- Write Item: 使用 connect 时, 已经建立的 Item 对象的 Write method 就可以将值输出。此值会传到 OPC Server 然后经过通信 Driver 将值写入现场控制系统内。注意要点: 此 Item 必须为 Read/Write Enabled OPC TAG 否则 OPC Server 将不会处理。至于 Read/Write TAG 的规定则必须

参照该 OPC Server 使用手册的 TAG 定义规格。

```
Private Sub Command4_Click()  
' Write value-1  
Dim Value As Variant  
  
Value = SetValue(0)  
OneItem_A.Write Value ' AO output  
  
End Sub
```

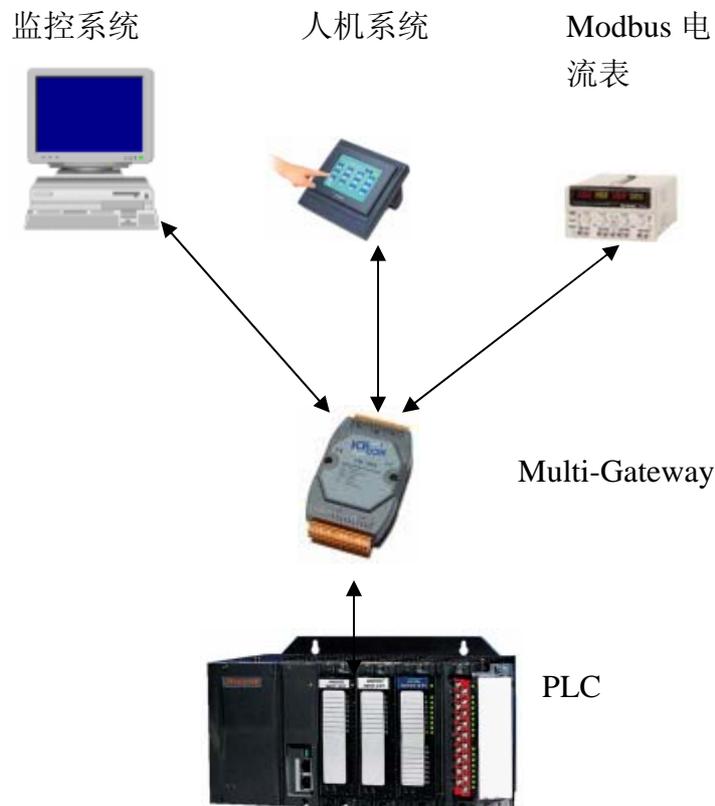
- Disconnect Server: 解除联机，也是分成三阶层来看，首先 Remove 所联机的 Item，再来 Group 集合，最后 Server Disconnect 等。注意要点：必须按照此三步骤顺序解除之，如果有任一步骤不确实，则 OPC Server 认为此 OPC Client 还在使用，因此产生一些系统问题。例如：前一个未解除，OPC Server 相关内存还保留，下次再次连接又建立新的资料造成内存浪费。当所有 OPC Client 解除后，OPC Server 程序应该停止，但是因为保留未被解除的旧连接资料，故 OPC Server 永远于激活状态。此情形于 OPC Client 开发除错阶段更是常常发生，设计者必须特别注意。

```
Private Sub Command2_Click()  
  
Dim Error() As Long  
  
IconItems.Remove 3, ServerHandles, Error 'remove item first  
IconGroups.RemoveAll ' and then remove Group  
IconServer.Disconnect ' disconnect server final  
  
Command1.Enabled = True ' connect button  
Command2.Enabled = False ' disconnect button  
Command3.Enabled = False ' Sync Read button  
Command4.Enabled = False ' Write value-1 button  
Command5.Enabled = False ' Write value-2 button  
Command6.Enabled = False ' Write value-3 button  
  
End Sub
```

2 多功能通信 Gateway

于自动控制系统，PLC 是一个主要的控制设备，除了负责全系统的所有控制处理外，还必须与外部监控系统等做资料交换。目前 PLC 对外资料交换的机制，绝大部份为串行埠式的通信方式，此种方式目前已发展成 RS232、RS422、RS485 等三种硬件架构，传送速度也可达到 115Kbps 的高速。但是应用上还是有一个限制，就是连接埠对外只能接上一个监控系统，也就是一个 PLC 希望同时与两个以上监控系统，或者一个连接端使用 RS232 另一个连接端使用 RS485 就发生冲突。还有许多的仪表或特殊的控制设备都提供 Modbus(RTU, ASCII) 通信协议，但是每种厂牌的 PLC 通信协议都不同，如果要连接这些设备只有于 PLC 上安装该厂牌 PLC 所提供的 Modbus 转换模块，此种模块普遍价格很高。由 PC 系统的角度来看，以前用来解决此问题只能于 PLC 上再安装另一个通信模块。但是随着科技的进步，以 PC 为架构所开发出来的 PC-Base Controller 具有多个通信端口，且可以设计程序，因应各种不同通信协议的转换，又只有巴掌大的 Size 于现场安装非常方便，则是另一选择。作者使用泓格的 7524 及 7188E 系列产品设计多种款式的通信转换器，只要简单的做基本通信系数设定，即可随插随用，不用更动现有的控制系统，非常方便。以下简单介绍此种 Multi-Gateway 的架构及功能，提供各位业界先进参考。

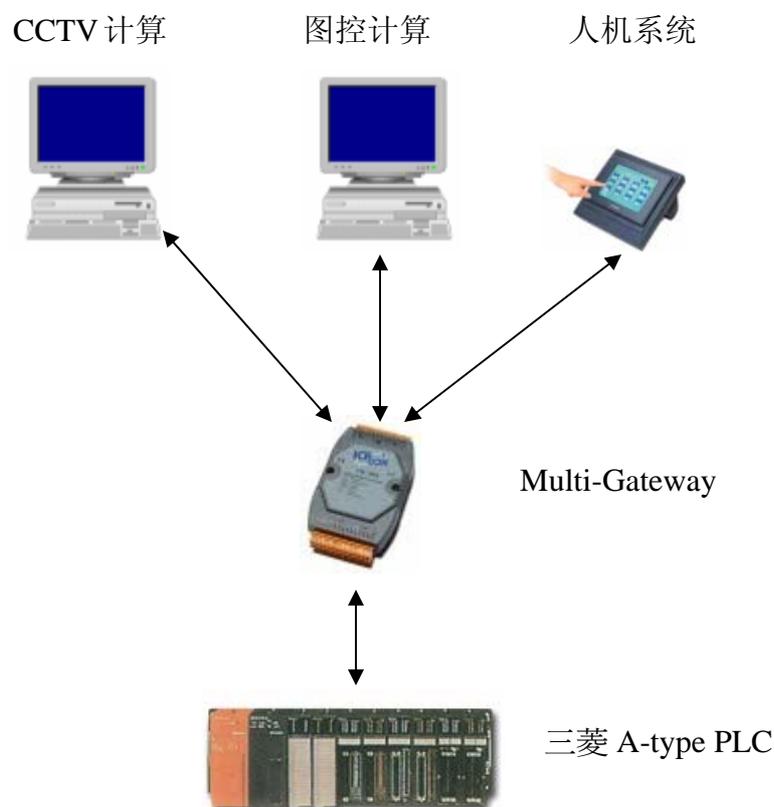
2.1 Multi-Gateway 架构(1): 一对多含 Modbus Protocol 转换



图(5-14): Multi-Gateway 架构图(1)

以一个通信埠连接 PLC，然后其它三个通信埠各连接不同的设备，例如：一个端口连接图控系统以该厂牌 PLC 的通信协议通信，另一端口连接触控屏幕的人机接口也是以 PLC 的通信协议通信，然后第三个埠连接电流表，以 Modbus 通信协议通信。最后所有通信都经过 Multi-Gateway 的处理，与 PLC 通信然后将资料回传所对应连接的系统上。如此架构好象将原本只有一个通信模块的 PLC，分成可同时接上三个系统与其通信，且可分别以不同的 baud rate 或 Modbus(RTU, ASCII)通信。此外 Multi-Gateway 提供第五个通信埠，可以与 PC 相连，然后以监视程序观看系统运作情形，且作必要的操作。

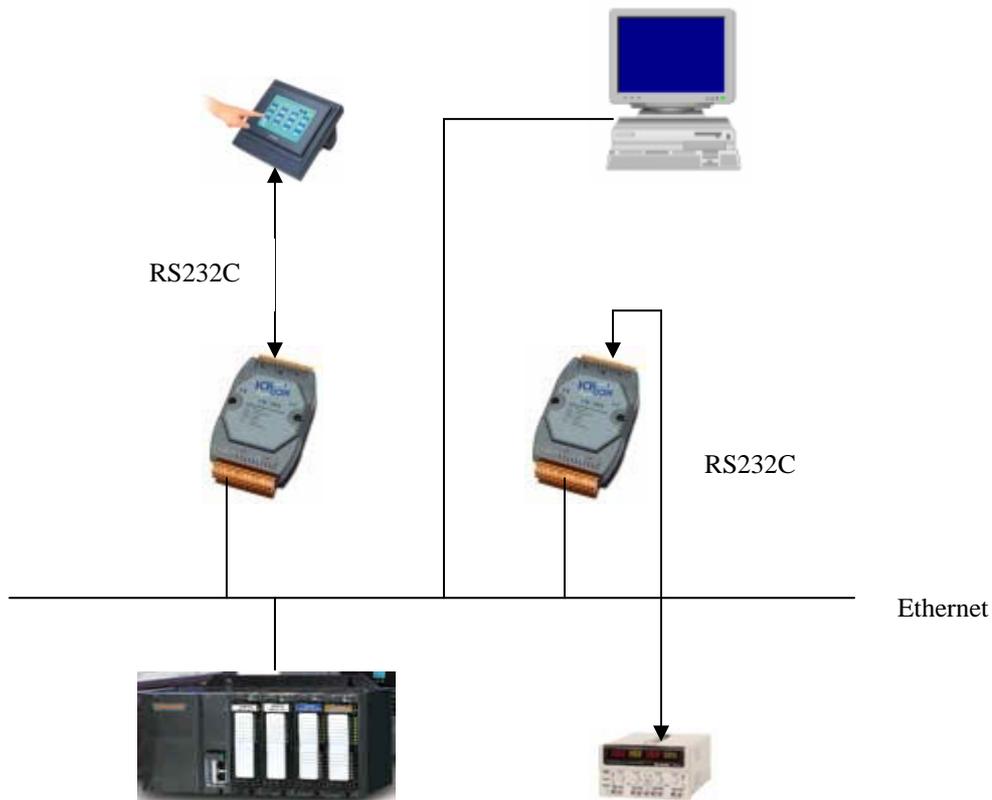
2.2 Multi-Gateway 架构(2): 一对多接多种监控系统



图(5-15): Multi-Gateway 架构图(2)

以一个通信埠连接 PLC，然后其它三个通信埠各连接不同的设备，例如：2 个端口连接图控系统以该厂牌 PLC 的通信协议通信，另一端口连接触控屏幕的人机接口也是以 PLC 的通信协议通信。所有通信都经过 Multi-Gateway 的处理，与三菱 PLC 通信然后将资料回传所对应连接的系统上。如此架构好象将原本只有一个通信模块的 PLC，分成可同时接上三个系统与其通信，且可分别以不同的 baud rates 通信。此外 Multi-Gateway 提供第五个通信埠，可以与 PC 相连，然后以监视程序观看系统运作情形，且作必要的操作。

2.3 Multi-Gateway 架构(3): Modbus 232/TCP 互转系统



图(5-16): Multi-Gateway 架构图(3)

主要功能:

- Modbus 为自控业界最主流的通信协议，目前通信技术已经发展至 Ethernet。Modbus 也提供 ModbusTCP 的通信协议，但是许多自控设备还是提供 Modbus Serial 方式，通信协议的转换 Modbus Gateway 成为必要的设备。
- Modbus Gateway 对通信数据处理，采取完全 transparent mode 保证转换资料的完整性。
- 可变成一对多方式，增加使用的便利性。
- 通信系数设定后，正式使用不需操作，只要 Power ON 即可。