



API 使用說明書 (.Net Remoting)

(對應 SkyMars Professional / Express 3.06 以上)

Version 3.06

2014/5/7

目錄

一、	SkyMars Professional 架構圖.....	7
二、	開發者撰寫程式流程.....	8
三、	API 連線及寫入密碼.....	12
四、	回傳錯誤碼說明.....	13
五、	FTP 回傳錯誤列表.....	15
六、	內部資訊類別函式.....	17
1.	SKY_version：取得 SkyMars 的版本及 USB Key 資訊.....	17
2.	SKY_conn_status：取得 CNC 連線狀態.....	17
3.	SKY_conn_ip_port：取得 CNC 連線 IP 及 Port 及其他相關資訊.....	18
4.	SKY_nc_filename：取得 CNC NC 檔名.....	19
5.	SKY_conn_count：取得 SkyMars 的已經連接的機台數量.....	20
七、	基本類別函式.....	21
1.	GET_information：基本相關資訊.....	21
2.	GET_information_heid：海德漢控制器_基本相關資訊.....	21
3.	GET_status：狀態資訊.....	23
4.	GET_position：座標資訊.....	24
5.	GET_gcode：G Code.....	25
6.	GET_othercode：其他 Code(H Code,D Code,T Code,M Code,B Code,F Code,S Code).....	25
7.	GET_feed_spindle：進給率/轉速.....	26
8.	GET_time：時間.....	27
9.	GET_time_heid：海德漢控制器_時間.....	28
10.	GET_time_cnc：取得控制器系統時間.....	30
11.	GET_part_count：工件數.....	30
12.	GET_part_total：.....	31
13.	GET_part_required：.....	32
14.	SET_relpos：設定相對座標數值.....	33
15.	SET_time_cnc：設定控制器的時間.....	34
八、	警報類別函式.....	35
1.	GET_alm_current：目前發生的警報.....	35
2.	GET_alm_current2：目前發生的警報.....	35
3.	GET_alm_current_heid：海德漢控制器_目前發生的警報.....	37
4.	GET_alm_history：警報履歷.....	38
5.	GET_alm_history2：警報履歷.....	38
6.	GET_alm_history_heid：警報履歷_海德漢控制器.....	40
7.	GET_msg_current：取得 Operation 訊息.....	41
8.	GET_msg_history：取得 Operation 履歷.....	42
9.	GET_plc_alarm：取得 PLC Alarm 履歷.....	43
九、	伺服主軸類別函式.....	44

1.	GET_servo_current：各軸負載電流值.....	44
2.	GET_servo_load：取得伺服負載%.....	44
3.	GET_servo_speed：取得伺服軸轉速.....	45
4.	GET_spindle_load：取得主軸負載%.....	46
5.	GET_spindle_speed：取得主軸轉速.....	47
6.	GET_servo_temperature：取得伺服馬達溫度.....	48
7.	GET_spindle_temperature：取得主軸溫度.....	48
十、	刀具管理類別函式 (控制器).....	50
1.	GET_offset_title：取得刀具 Offset 的標題列.....	50
2.	GET_offset_all：取得所有刀具 Offset.....	51
3.	GET_offset_scope：依指定範圍取得刀具 Offset.....	52
4.	GET_offset_single：取得單筆刀具 Offset.....	53
5.	SET_offset_all：寫入所有刀具 Offset.....	54
6.	SET_offset_single：寫入單筆刀具 Offset.....	55
7.	GET_offset_count：取得刀具 Offset 筆數.....	56
8.	GET_tool_title：取得刀具管理的標題列.....	57
9.	GET_tool_count：取得刀具 Offset 筆數.....	57
10.	GET_tool_data：取得所有刀具管理資料.....	58
11.	GET_tool_data_mem：取得刀具管理，並將資料讀取到 SkyMars 記憶體中....	59
12.	GET_tool_data_scope：取刀具管理資料，並依指定索引值範圍傳回.....	60
13.	SET_tool_data：寫入刀具管理.....	61
14.	GET_pocket_title：取得刀庫表的標題列.....	62
15.	GET_pocket_count：取得刀庫表筆數.....	63
16.	GET_pocket_data：取得刀庫表資料.....	64
17.	SET_pocket_data：寫入刀庫表.....	65
18.	GET_pocket_data_mem：取得刀庫表，並將資料讀取到 SkyMars 記憶體中....	66
19.	GET_pocket_data_scope：取刀庫表資料，並依指定索引值範圍傳回.....	66
十一、	工件座標類別函式.....	68
1.	GET_work_coord_title：取得工件座標的標題列.....	68
2.	GET_work_coord_all：取得所有工件座標資料.....	68
3.	GET_work_coord_scope：依指定範圍取得工件座標資料.....	69
4.	GET_work_coord_single：取得單筆工件座標資料.....	71
5.	SET_work_coord_all：寫入所有工件座標資料.....	72
6.	SET_work_coord_single：寫入單筆工件座標資料.....	73
7.	GET_work_coord_count：取得工件座標筆數.....	74
8.	GET_preset_title：取得工作台管理表的標題列.....	75
9.	GET_preset_count：取得工作台管理表筆數.....	76
10.	GET_preset_data：取得工作台管理表資料.....	77
11.	SET_preset_data：寫入工作台管理表.....	78
12.	GET_preset_data_mem：取得工作台管理表，並將資料讀取到 SkyMars 記憶體中	

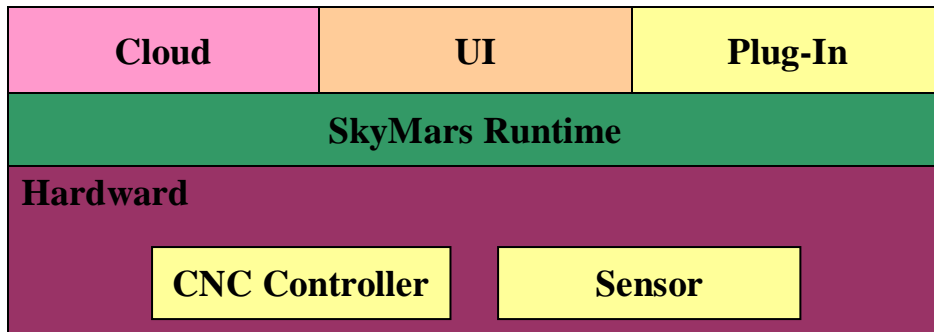
13.	GET_preset_data_scope：取得工作台管理表，並依指定索引值範圍傳回.....	79
十二、	Macro 變數類別函式(共通變數).....	82
1.	GET_macro_all：取得所有 Macro(共通變數)變數資料.....	82
2.	GET_macro_scope：依指定範圍取得 Macro(共通變數)變數資料.....	83
3.	SET_macro_all：寫入所有 Macro(共通變數)變數資料.....	84
4.	GET_macro_single：取得單筆 Macro(共通變數)變數資料.....	85
5.	SET_macro_single：寫入單筆 Macro(共通變數)變數資料.....	86
6.	GET_macro_variable：取得 Macro(共通變數)變數型態.....	87
7.	GET_pcode_single：取得單筆 P Code Macro 資料(FANUC).....	88
8.	SET_pcode_single：寫入單筆 P Code Macro 資料(FANUC).....	88
十三、	加工程式類別函式.....	90
1.	GET_nc_mem_list：取得機台內部加工程式清單(記憶體).....	90
2.	GET_nc_ftp_list：取得 FTP 卡加工程式清單(Fanuc DataServer/三菱 CF 卡).....	91
3.	GET_nc_mem_code：取得記憶體內的加工程式內容.....	92
4.	GET_nc_ftp_code：取得 FTP 加工程式內容(Fanuc DataServer/三菱 CF 卡).....	93
5.	UPLOAD_nc_mem：上傳加工程式至記憶體.....	94
6.	UPLOAD_nc_ftp：上傳加工程式至 FTP(Fanuc DataServer/三菱 CF 卡).....	95
7.	DEL_nc_mem：刪除加工程式(記憶體).....	96
8.	DEL_nc_ftp：刪除加工程式(FTP) (Fanuc DataServer/三菱 CF 卡).....	97
9.	GET_nc_pointer：取得程式目前的指標(行).....	98
10.	GET_nc_current_block：取得目前程式正在執行及未來要執行的單節.....	99
11.	GET_nc_freespace：取得剩餘空間.....	100
12.	SET_nc_main：設定或指定加工程式為主程式.....	101
13.	UPLOAD_nc_mdi：上傳加工程式碼到控制器(MDI 模式).....	101
十四、	系統參數類別函式.....	103
1.	GET_param_max：取得參數最大號碼(或最後一個號碼).....	103
2.	GET_param_data：取得參數資料.....	103
3.	SET_param_data：設定參數資料.....	106
十五、	PLC 類別函式.....	109
1.	GET_plc_ver：取得 PLC 版本資訊.....	109
2.	GET_plc_addr：取得 PLC 位址的起始及結束號碼.....	109
3.	GET_plc_addr2：取得 PLC 位址的起始及結束號碼.....	111
4.	GET_plc_status：取得機台 PLC 狀態資訊.....	112
5.	GET_plc_status2：取得機台 PLC 狀態資訊.....	113
6.	SET_plc_status：寫入機台 PLC.....	115
7.	SET_plc_status2：寫入機台 PLC.....	116
十六、	檔案管理類別函式.....	119
1.	GET_file_directory：取得控制上的檔案及目錄.....	119
2.	UPLOAD_file2：從 PC 端上傳檔案到裝置(如 CNC 控制器).....	120

3.	DOWNLOAD_file2：從裝置上(如 CNC 控制器)下載檔案到 PC 端.....	120
4.	DEL_file：從裝置上刪除檔案.....	121
十七、	National Instruments 9233 加速規模組類別	122
1.	START_ni9233_get_data：設定啟動模組進行資料擷取.....	122
2.	STOP_ni9233_get_data：停止擷取.....	123
3.	CHECK_ni9233_running：確認是否已啟動資料擷取	123
4.	GET_ni9233_spectrum_data：取得資料	124
5.	GET_ni9233_error_msg：取得模組錯誤訊息.....	125
十八、	RS232 模組類別	126
1.	GET_rs232_status：取得目前 RS232 的相關狀態及參數.....	126
2.	OPEN_rs232：開啟 RS232.....	127
3.	SET_rs232：將指定的資料寫入 RS232.....	128
4.	CHECK_rs232_data_received：確認是否已經接收到資料(IsOnCallbackEvent=true) 129	
5.	GET_rs232_data_received：取得 RS232 所傳回的資料.....	129
6.	CLOSE_rs232：關閉 RS232	130
7.	GET_rs232_error：取得 RS-232 例外錯誤訊息	131
十九、	PC Camera 模組類別(USB)	132
1.	CONNECT_cam：連線攝影機	132
2.	DISCONNECT_cam：關閉攝影機	132
3.	CHECK_cam_running：確認攝影機是否已連接及擷取.....	133
4.	GET_cam_image：取得攝影機影像	133
5.	GET_cam_error：取得攝影機例外錯誤訊息.....	134
二十、	SkyMars Professional 事件派送	136
1.	EVENT_conn_cnc：產生 SkyMars Professional 設定 CNC 資訊完成事件.....	137
2.	EVENT_disconn_cnc：產生 SkyMars Professional 結束 CNC 連線事件	138
3.	EVENT_conn_status：產生 CNC 狀態事件(OFF、RUN、IDLE、ALARM).....	139
4.	EVENT_nc_filename：產生 CNC 切換 NC 檔名事件.....	140
二十一、	SkyMars 稼動率資訊	142
1.	GET_utilization_all_today：取得今日機台稼動率資訊.....	142
2.	GET_utilization_single_time：取得單一機台總時間之稼動率資訊.....	142
3.	GET_utilization_single_total：取得單一機台總次數(工件數、加工循環次數)之稼動率 資訊.....	144
4.	GET_utilization_single_list：取得單一機台稼動率履歷資訊	145
5.	GET_utilization_single_all：取得單一機台稼動率資訊及履歷	146
二十二、	SkyMars 維護管理類別	149
1.	GET_maint_count：取得維護管理的總筆數	149
2.	GET_maint_title：取得維護管理的標題列文字	149
3.	GET_maint_all：取得維護管理所有資料.....	150
4.	GET_maint_all_time：取得維護管理項目內的所有時間(使用時間、壽命時間)	151

5.	GET_maint_single	: 取得單筆的維護管理的項目資料	151
6.	GET_maint_single_time	: 取得單筆維護管理的時間資料(使用時間、壽命時間)...	152
7.	ADD_maint_single	: 新增單筆維護管理資料.....	153
8.	SET_maint_single	: 修改單筆維護管理的資料	154
9.	DEL_maint_single	: 刪除單筆維護管理的資料.....	155
10.	SET_maint_single_usetime	: 設定單筆維護管理的使用時間	156
11.	SET_maint_single_zero	: 將單筆維護管理的使用時間歸零	156
二十三、	SkyMars 刀具資料庫管理類別		158
	以下表格所列函式皆已過時，我們建議您使用新的函式取代：		158
1.	GET_toolmanage_count	: 取得刀具資料庫管理的總筆數	158
2.	GET_toolmanage_title2	: 取得刀具資料庫管理的標題列文字	158
3.	GET_toolmanage_all2	: 取得刀具資料庫管理所有資料	159
4.	GET_toolmanage_all_execute	: 取得刀具資料庫管理的所有時間及工件數(加工時間、壽命時間、加工工件數及壽命工件數).....	160
5.	GET_toolmanage_single2	: 取得單筆的刀具資料庫管理資料.....	161
6.	GET_toolmanage_single_execute	: 取得單筆刀具資料庫管理的時間及工件數資料(加工時間、壽命時間、加工工件數及壽命工件數).....	162
7.	ADD_toolmanage_single2	: 新增單筆刀具資料庫管理資料.....	163
8.	SET_toolmanage_single2	: 修改單筆刀具資料庫管理的資料.....	164
9.	DEL_toolmanage_single	: 刪除單筆刀具資料庫管理的資料.....	166
10.	SET_toolmanage_single_cycletime	: 設定單筆刀具資料庫管理的加工時間	166
11.	SET_toolmanage_single_partcount	: 設定單筆刀具資料庫管理的加工工件數.	167
12.	SET_toolmanage_single_zero	: 將單筆刀具資料庫管理的加工時間歸零.....	168
13.	SET_toolmanage_single_zero_partcount	: 將單筆刀具資料庫管理的加工工件數歸零	168
二十四、	應用程式開發者軟體防盜版策略		170
二十五、	問題排除.....		171

一、SkyMars Professional 架構圖

SkyMars Professional 架構層級由 4 個部份組成，如下圖所示：



1. SkyMars Runtime

此層級負責與硬體端 CNC 各型控制器或其他感測器來連結。

2. User Interface (UI)

SkyMars Professional 提供標準的界面(UI)供使用者操作，使用者進行任何操作時，都會與 SkyMars Runtime 層級進行互動，此層級的界面含有資訊監視、程式上下載、參數資訊、伺服主軸監視、刀具管理及維護等。

3. Cloud

Cloud 模組會與雲端伺服器連結，由 SkyMars Runtime 層級視需要呼叫，如啟動稼動率服務後，SkyMars Runtime 會將 CNC 基本資訊、狀態，藉由 Cloud 模組傳送至雲端伺服器。

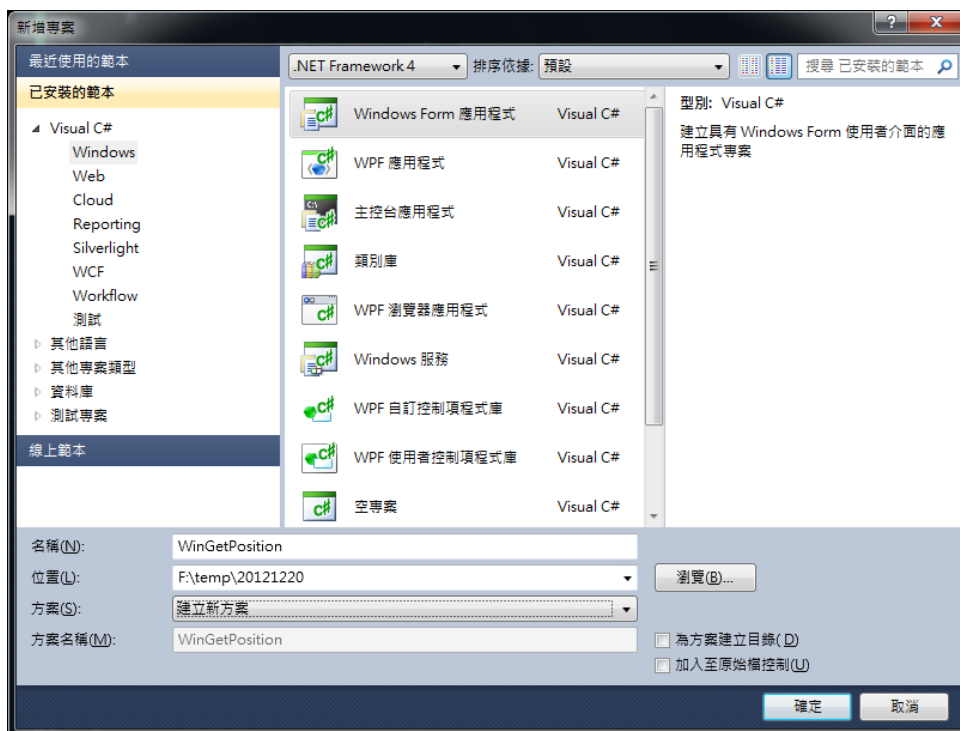
4. Plug-In Interface

Plug-In Interface 為 Interface DLL 檔，Interface 內定義函式名稱及結構變數，開發者必須加入參考，並建立命名空間，才可以使用 Interface 內所定義的函式及結構變數來呼叫使用。

二、開發者撰寫程式流程

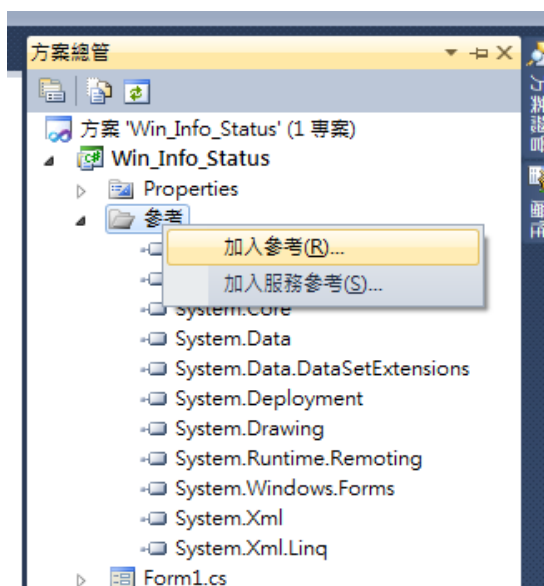
1. 新增專案

您可以使用微軟開發工具 Visual Studio 2005 / 2008 / 2010 / 2012 開發，在開發環境上將擁有完全的支援，這對我們撰寫程式碼有相當大的助益。我們新增一個專案，此專案選擇 Visual C#或 Visual Basic，並選擇相對應的開發項目及指定專案名稱後，按下「確定」開發環境將會為我們建立此專案。

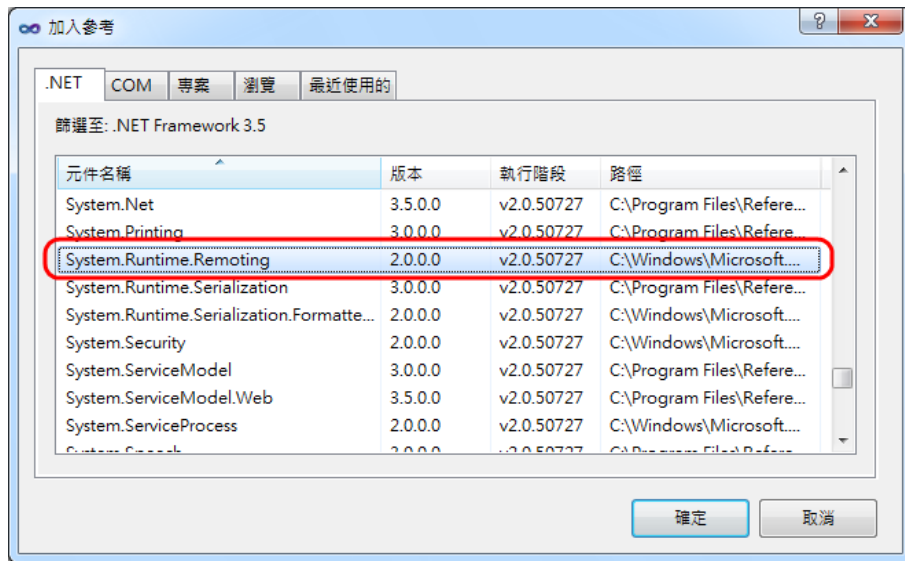


2. 加入參考 **System.Runtime.Remoting** 及 **InterfaceLib.dll**

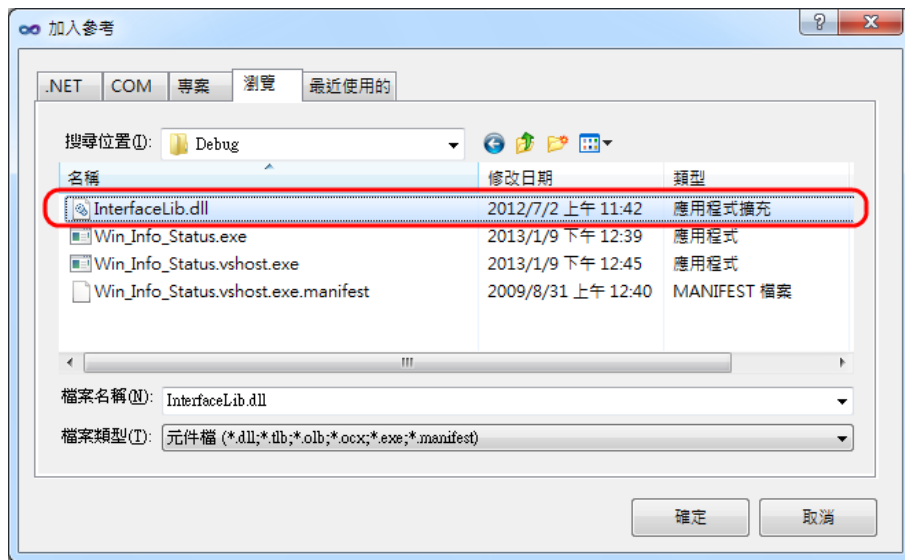
在撰寫程式碼前，我們必須要加入兩個參考，才能使用 SkyMars API 提供的函式；請從「方案總管」-「參考」再點選右鍵「加入參考」。



點選「.NET」頁次，在元件名稱上找到 **System.Runtime.Remoting**，並點選「確定」加入參考。



接下來我們必須要再加入「**InterfaceLib.dll**」；加入前，請先將該檔案複製到目前建置專案的執行檔底下(如：專案名稱bin\Debug)。依序在「方案總管」-「參考」再點選右鍵「加入參考」。點選「瀏覽」頁次，切換到 bin\Debug 的資料夾，將 **InterfaceLib.dll** 點選，並按下「確定」加入參考。



3. 建立命名空間

請在程式碼檢視，using的下面加入以下命名空間程式碼：

```
using System.Runtime.Remoting;  
using System.Runtime.Remoting.Channels;  
using System.Runtime.Remoting.Channels.Tcp;  
using InterfaceLib;
```

可參考如下圖：

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Windows.Forms;
9
10 using System.Runtime.Remoting;
11 using System.Runtime.Remoting.Channels;
12 using System.Runtime.Remoting.Channels.Tcp;
13 using InterfaceLib;
```

4. 建立 InterfaceLib.IMsg 型別的 iRemoting 變數

請將以下的程式碼，加入到適當的位置：

```
InterfaceLib.IMsg iRemoting = null;
```

- 可自行定義 iRemoting 變數

5. 註冊 .Net Remoting 通道並繫結

定義出來的 iRemoting 變數，無法直接使用，我們必須要繫結到 SkyMars 的 Channel，才能夠使用 API 函式，請將以下程式碼加入到適當位置。

```
if (ChannelServices.RegisteredChannels.Length == 0)
{
    ChannelServices.RegisterChannel(new TcpChannel());
}
iRemoting = (IMsg)Activator.GetObject(typeof(IMsg),
    "tcp://localhost:9501/RemoteObjectURI9501");
```

- "tcp://localhost:9501/RemoteObjectURI9501"：

9501 代表 SkyMars 連線的第一台機台，第二台之後會依序號接續下去，如第二台為 9502，第三台為 9503，以此類推。

- 採用 TCP 及二進位流傳輸連線，必須要注意防火牆是否阻擋。

6. 參考範例程式碼：

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
using InterfaceLib;
```

```
namespace WindowsFormsApplication1
{
    public partial class FormSample3 : Form
    {
        InterfaceLib.IMsg iRemoting = null;
        StructMsg.Pwd _Pwd;
        StructMsg.SkyConn_ip_port R1;

        public FormSample3()
        {
            InitializeComponent();
        }

        private void FormSample3_Load(object sender, EventArgs e)
        {
            if (ChannelServices.RegisteredChannels.Length == 0)
                ChannelServices.RegisterChannel(new TcpChannel(), false);

            iRemoting = (IMsg)Activator.GetObject(typeof(IMsg),
                "tcp://localhost:9501/RemoteObjectURI9501");
            _Pwd.ConnectionKey = "123";

            //取得廠牌及機台名稱
            short ret = iRemoting.SKY_conn_ip_port(_Pwd, ref R1);
            if (ret == 0)
            {
                txtManufacturer.Text = R1.Manufacturer[0];
                txtName.Text = R1.MachineName[0];
            }
        }
    }
}
```

7. 如何得知 SkyMars 已連線的機台數目

您可以先連線到 SkyMars 的第一台機台，如預設的 9501，並使用 API 函式 SKY_conn_ip_port，從資料結構裡面，陣列長度就是 SkyMars 已連線的機台數目。

三、API 連線及寫入密碼

目前 SkyMars Professional /Express 3.0 之後的版本，所有函式全部加入「API 連線密碼」及「API 寫入密碼」，主要目的在於提升安全性。在網際網路(Internet)上確保 SkyMars 不會隨意接受外部命令來更改或取得控制器資訊。所有函式都會要求輸入 Pwd 的資料結構，如下表所示：

結構名稱：Pwd

變數名稱	型別	[in/out]	說明
ConnectionKey	string	in	API 連線密碼
WritePwd	string	in	API 寫入密碼

「API 寫入密碼」意旨當外部命令透過函式要寫入控制器時，該函式會要求輸入「API 寫入密碼」，SkyMars 會進行比對，正確無誤後，才會寫入控制器。

設定方式：

1. 開啟 SkyMars 的安全性設定

請開啟 SkyMars Professional 或 SkyMars Express，[設定]→[API 安全性設定]。



2. 設定 API 連線密碼及 API 寫入密碼。
3. 函式傳回值說明
傳回值為 7：SkyMars API 被設定成寫入保護。
傳回值為 20：密碼不正確。

4. 程式碼範例：

```
StructMsg.SkyConn_status _SkyConn_status;
StructMsg.Pwd _Pwd;
private void button2_Click(object sender, EventArgs e)
{
    _Pwd.ConnectionKey = "123";
    short ret = iRemoting.SKY_conn_status(_Pwd, ref _SkyConn_status);
    if (ret == 0)
    {
        txtMachineNo.Text = _SkyConn_status.MachineNo[0].ToString();
        txtStatus.Text = _SkyConn_status.Status[0].ToString();
    }
}
```

四、回傳錯誤碼說明

傳回值	錯誤類別	說明
-31	Camera Exception	Camera 發生內部例外錯誤。
-30	Camera Current Running	Camera 目前已連接及影像擷取中。
-29	Camera Disconnect	Camera 未連接。
-28	RS-232 Close	RS-232 Com Port 已關閉。
-27	RS-232 Over limit	RS-232 所能連接的模組，已超過上限。
-26	RS-232 Return Type	未指定 RS-232 傳回的型別。
-25	RS-232 Open Failed	RS-232 COM Port 開啟失敗或已經開啟。
-24	RS-232 Exception	RS-232 發生內部例外錯誤，請使用相對應的函式取得錯誤。
-23	Sensor Exception	感測器模組內部或 API 發生錯誤，請使用相對應的函式取得錯誤。
-22	Sensor Current Running	目前感測器正在執行中。
-21	USB key error	USB Key 失效(硬體鎖失效)。
-20	Plug-In Time out	執行的函式已經 Time out，如果要加長，請直接修改 Register。
-19	SkyMars Busy	SkyMars 目前正在啟動中。
-18	Not supported	控制器不支援此函式。
-17	Protocol error (Ethernet version only)	網路卡設定不正確。
-16	Socket error (Ethernet version only)	連線嘗試失敗 / 控制器拒絕連線。
-15	DLL file error	使用的 DLL 檔，無法對應 CNC 的型號。或 DLL 檔已經遺失。
-8	Handle number error	請取得 handle 值。
-7	Version mismatch between the CNC/PMC and library	CNC/PMC 版本無法對應 library. 更換 library 或更新 CNC/PMC 控制軟體。
-6	Abnormal library state	執行的 library 發生例外錯誤。
-3	Random key timeout	隨機密鑰過時失效，請重新擷取 Random key。
-2	Reset or stop request	Reset 或 Stop 按鈕被按下。函式呼叫被中止。
-1	CNC Busy	目前 CNC 處於忙錄狀態，請稍後再測試。
0	Normal termination	正常執行，未發生錯誤。
1	Error(function is not executed, or not available)	尚未執行特定函式，必須事先執行。否則該函式無法使用。
2	Error(data block length error, error of number of data)	請檢查資料長度及資料數量。
3	Error(data number error)	請檢查資料號碼是否正確。

4	Error(data attribute error)	請檢查資料屬性是否正確。
5	Error(data error)	寫入的資料不正確。
6	Error(no option)	CNC 未選配(購買)該項功能。
7	Error(write protection)	寫入保護。
8	Error(memory overflow)	記憶體溢位。
9	Error(CNC parameter error)	參數設定錯誤。
10	Error(buffer empty/full)	緩衝區屬於 empty 或 full。
11	Error(path number error)	路徑不正確。
12	Error(CNC mode error)	CNC 模式(Mode)不正確。
13	Error(CNC execution rejection)	CNC 拒絕執行。
14	Error(Data server error)	Data Server 發生錯誤。
15	Error(alarm)	警報處理發生錯誤。
16	Error(stop)	CNC 狀態在 stop 或 emergency。
17	Error(State of data protection)	資料被 CNC 所被保護。
18	Error(Not found Machine ID)	找不到要連線的機台。
19	Error(No out)	請確認 NO。
20	Error>Password)	密碼錯誤。
21	Error(Compatible)	相容性問題。(無法相容舊版 API)

五、FTP 回傳錯誤列表

110	Restart marker reply.
120	Service ready in (n) minutes.
125	Data connection already open, transfer starting.
150	File status okay, about to open data connection.
200	Command okay.
202	Command not implemented
211	System status, or system help reply.
212	Directory status.
213	File status.
214	Help message.
215	NAME system type. (NAME is an official system name from the list in the Assigned Numbers document.)
220	Service ready for new user.
221	Service closing control connection. (Logged out if appropriate.)
225	Data connection open, no transfer in progress.
226	Closing data connection. Requested file action successful (file transfer, abort, etc.).
227	Entering Passive Mode
230	User logged in, proceed.
250	Requested file action okay, completed.
257	PATHNAME created.
331	User name okay, need password.
332	Need account for login.
350	Requested file action pending further information.
421	Service not available, closing control connection. (May be a reply to any command if the service knows it must shut down.)`
425	Can't open data connection.
426	Connection closed, transfer aborted.
450	Requested file action not taken. File unavailable (e.g., file busy).
451	Requested action aborted, local error in processing.
452	Requested action not taken. Insufficient storage space in system.
500	Syntax error, command unrecognized. This may include errors such as command line too long.
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands.
504	Command not implemented for that parameter.
530	User not logged in.
532	Need account for storing files.
550	Requested action not taken. File unavailable (e.g., file not found, no access).
552	Requested file action aborted, storage allocation exceeded

553	Requested action not taken. Illegal file name.
-----	--

六、內部資訊類別函式

1. SKY_version：取得 SkyMars 的版本及 USB Key 資訊

API 名稱	SKY_Version			
函式說明	取得 SkyMars 版本及 USB Key 資訊			
C#方法	short SKY_version(StructMsg.Pwd P, ref StructMsg.SkyVersion R);			
VB.Net 方法	SKY_version(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.SkyVersion) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：SkyVersion			
	變數名稱	型別	[in/out]	說明
	Version	float	out	SkyMars 版本
	SerialNumber	long	out	USB key 的序號
	LicenseCount	short	out	可連線機台數
備註	SerialNumber 為硬體 USB Key ID，每一支 USB Key 都會有不一樣的 ID			
適用範圍	不限控制器。			
C#程式範例	<pre>StructMsg.SkyVersion _SkyVersion; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.SKY_version(_Pwd, ref _SkyVersion); if (ret == 0) { txtVersion.Text = _SkyVersion.Version.ToString(); txtKeyId.Text = _SkyVersion.SerialNumber.ToString(); txtLic.Text = _SkyVersion.License.ToString(); } }</pre>			

2. SKY_conn_status：取得 CNC 連線狀態

API 名稱	SKY_conn_status			
函式說明	取得 CNC 連線狀態			
C#方法	short SKY_conn_status(StructMsg.Pwd P, ref StructMsg.SkyConn_status R);			
VB.Net 方法	SKY_conn_status(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.SkyConn_status) As Short			

資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：SkyConn_status			
	變數名稱	型別	[in/out]	說明
	MachineNo	int[]	out	機器的索引值
Status	short[]	out	0:未開機 1:RUN 2:IDLE 3:Alarm 4:同時發生 RUN 及 ALARM	
備註				
適用範圍	不限控制器。			
C#程式範例	<pre>StructMsg.SkyConn_status _SkyConn_status; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.SKY_conn_status(_Pwd, ref _SkyConn_status); if (ret == 0) { txtMachineNo.Text = _SkyConn_status.MachineNo[0].ToString(); txtStatus.Text = _SkyConn_status.Status[0].ToString(); } }</pre>			

3. SKY_conn_ip_port：取得 CNC 連線 IP 及 Port 及其他相關資訊

API 名稱	SKY_conn_ip_port			
函式說明	取得 CNC 連線 IP 及 Port 及其他相關資訊			
C#方法	short SKY_conn_ip_port(StructMsg.Pwd P, ref StructMsg.SkyConn_ip_port R);			
VB.Net 方法	SKY_conn_ip_port(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.SkyConn_ip_port) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：SkyConn_ip_port			
	變數名稱	型別	[in/out]	說明
	MachineNo	int[]	out	機器的索引值
	MachineName	string[]	out	機台名稱
	IP	string[]	out	機台 IP 位址
Port	short[]	out	機台 Port 號碼	
Manufacturer	string[]	out	廠牌	

	PlugInPort	int[]	out	Plug-In Port 號碼
備註				
適用範圍	不限控制器。			
C#程式範例	<pre>StructMsg.SkyConn_ip_port _SkyConn_ip_port; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.SKY_conn_ip_port(_Pwd, ref _SkyConn_ip_port); if (ret == 0) { txtMachineNo2.Text = _SkyConn_ip_port.MachineNo[0].ToString(); txtMachineName.Text = _SkyConn_ip_port.MachineName[0]; txtIP.Text = _SkyConn_ip_port.IP[0]; txtPort.Text = _SkyConn_ip_port.Port[0].ToString(); txtManufacturer.Text = _SkyConn_ip_port.Manufacturer[0]; txtPlugInPort.Text = _SkyConn_ip_port.PlugInPort[0].ToString(); } }</pre>			

4. SKY_nc_filename：取得 CNC NC 檔名

API 名稱	SKY_nc_filename																							
函式說明	取得 CNC NC 檔名																							
C#方法	short SKY_nc_filename(StructMsg.Pwd P, ref StructMsg.SkyNc_filename R);																							
VB.Net 方法	SKY_nc_filename(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.SkyNc_filename) As Short																							
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：SkyNc_filename <table border="1" data-bbox="475 1709 1444 2004"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>RetType</td> <td>short</td> <td>in</td> <td>指定 -1:All ,Other:Machine Index</td> </tr> <tr> <td>MachineNo</td> <td>int[]</td> <td>out</td> <td>機器的索引值</td> </tr> <tr> <td>MainProg</td> <td>string[]</td> <td>out</td> <td>主程式檔名</td> </tr> <tr> <td>SubProg</td> <td>string[]</td> <td>out</td> <td>副程式檔名</td> </tr> </tbody> </table>				變數名稱	型別	[in/out]	說明	RetType	short	in	指定 -1:All ,Other:Machine Index	MachineNo	int[]	out	機器的索引值	MainProg	string[]	out	主程式檔名	SubProg	string[]	out	副程式檔名
變數名稱	型別	[in/out]	說明																					
RetType	short	in	指定 -1:All ,Other:Machine Index																					
MachineNo	int[]	out	機器的索引值																					
MainProg	string[]	out	主程式檔名																					
SubProg	string[]	out	副程式檔名																					
備註																								

適用範圍	不限控制器。
C#程式範例	<pre>StructMsg.SkyNc_filename _SkyNc_filename; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.SKY_nc_filename(_Pwd, ref _SkyNc_filename); if (ret == 0) { txtMainProg.Text = _SkyNc_filename.MainProg[0]; txtSubProg.Text = _SkyNc_filename.SubProg[0]; } }</pre>

5. SKY_conn_count：取得 SkyMars 的已經連接的機台數量

API 名稱	SKY_conn_count		
函式說明	取得 SkyMars 的已經連接的機台數量		
C#方法	short SKY_conn_count(StructMsg.Pwd P, ref StructMsg.total_count R);		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：total_count		
	變數名稱	型別	[in/out] 說明
	Count	short	out 連線機台數
備註			
適用範圍	不限控制器。		
C#程式範例	<pre>StructMsg.Pwd _Pwd = new StructMsg.Pwd(); _Pwd.ConnectionKey = "123"; StructMsg.total_count R = new StructMsg.total_count(); short ret = iRemoting.SKY_conn_count(_Pwd, ref R); if (ret == 0) MessageBox.Show(R.Count.ToString()); else MessageBox.Show(ret.ToString());</pre>		

七、基本類別函式

1. GET_information：基本相關資訊

API 名稱	GET_information			
函式說明	基本相關資訊			
C#方法	short GET_information(StructMsg.Pwd P, ref StructMsg.information R);			
VB.Net 方法	GET_information(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.information) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：information			
	變數名稱	型別	[in/out]	說明
	Axes	short	out	可控制軸數
	CncType	string	out	ex : "18", "31", "0"
	MaxAxes	short	out	最大軸數
	Series	string	out	M/T type
	Nc_Ver	string	out	NC 版本
AxisName	string[]	out	各軸座標名稱	
備註				
適用範圍	O：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	O	O	X	?
	寶元	工研院		
	O	O		
C#程式範例	<pre>StructMsg.information _information; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_information(_Pwd, ref _information); if (ret == 0) { txtAxes.Text = _information.Axes.ToString(); txtCncType.Text = _information.CncType; txtMaxAxes.Text = _information.MaxAxes.ToString(); txtSeries.Text = _information.Series; txtNc_Ver.Text = _information.Nc_Ver; } }</pre>			

2. GET_information_heid：海德漢控制器_基本相關資訊

API 名稱	GET_information_heid																															
函式說明	基本相關資訊																															
C#方法	short GET_information_heid(StructMsg.Pwd P, ref StructMsg.information_heid R);																															
VB.Net 方法	GET_information_heid(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.information_heid) As Short																															
資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：information_heid</p> <table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>Axes</td> <td>short</td> <td>out</td> <td>可控制軸數</td> </tr> <tr> <td>Model</td> <td>string</td> <td>out</td> <td></td> </tr> <tr> <td>Nc_Ver</td> <td>string</td> <td>out</td> <td></td> </tr> <tr> <td>FCL</td> <td>string</td> <td>out</td> <td></td> </tr> <tr> <td>Plc_Ver</td> <td>string</td> <td>out</td> <td></td> </tr> <tr> <td>AxisName</td> <td>string[]</td> <td>out</td> <td>各軸座標名稱</td> </tr> </tbody> </table>				變數名稱	型別	[in/out]	說明	Axes	short	out	可控制軸數	Model	string	out		Nc_Ver	string	out		FCL	string	out		Plc_Ver	string	out		AxisName	string[]	out	各軸座標名稱
變數名稱	型別	[in/out]	說明																													
Axes	short	out	可控制軸數																													
Model	string	out																														
Nc_Ver	string	out																														
FCL	string	out																														
Plc_Ver	string	out																														
AxisName	string[]	out	各軸座標名稱																													
備註																																
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>X</td> <td>O</td> <td>X</td> <td>X</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>X</td> <td>X</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				FANUC	三菱	海德漢	西門子	新代	X	X	O	X	X	寶元	工研院				X	X											
FANUC	三菱	海德漢	西門子	新代																												
X	X	O	X	X																												
寶元	工研院																															
X	X																															
C#程式範例	<pre>StructMsg.information_heid _information_heid; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_information_heid(_Pwd, ref _information_heid); if (ret == 0) { txtAxes.Text = _information_heid.Axes.ToString(); txtModel.Text = _information_heid.Model; txtNc_Ver.Text = _information_heid.Nc_Ver; txtFCL.Text = _information_heid.FCL; txtPlc_Ver.Text = _information_heid.Plc_Ver; txtAxisName.Text = _information_heid.AxisName[0]; } }</pre>																															

3. GET_status：狀態資訊

API 名稱	GET_status				
函式說明	狀態資訊				
C#方法	short GET_status(StructMsg.Pwd P, ref StructMsg.status R);				
VB.Net 方法	GET_status(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.status) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：status				
	變數名稱	型別	[in/out]	說明	
	MainProg	string	out	主程式號碼	
	CurProg	string	out	目前執行程式的號碼	
	CurSeq	int	out	目前執行程式的行號	
	Mode	string	out	ex:"MDI", "MEM"...	
	Status	string	out	ex:"STOP", "START"...	
	Alarm	string	out	ALARM	
Emg	string	out	EMG		
Monition	string	out	DWL / MTN		
備註	Monition 目前只支援 FANUC				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.status _status; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_status(_Pwd, ref _status); if (ret == 0) { txtMainProg.Text = _status.MainProg; txtCurProg.Text = _status.CurProg; txtCurSeq.Text = _status.CurSeq.ToString(); txtMode.Text = _status.Mode; txtStatus.Text = _status.Status; txtAlarm.Text = _status.Alarm; txtEMG.Text = _status.Emg; } }</pre>				

	}
--	---

4. GET_position：座標資訊

API 名稱	GET_position																																				
函式說明	座標資訊																																				
C#方法	short GET_position(StructMsg.Pwd P, ref StructMsg.position R);																																				
VB.Net 方法	GET_position(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.position) As Short																																				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：position <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>AxisName</td> <td>string[]</td> <td>out</td> <td>各軸座標名稱</td> </tr> <tr> <td>DecPoint</td> <td>short</td> <td>out</td> <td>座標小數點位數</td> </tr> <tr> <td>Unit</td> <td>string[]</td> <td>out</td> <td>座標單位</td> </tr> <tr> <td>Mach</td> <td>double[]</td> <td>out</td> <td>機械座標</td> </tr> <tr> <td>Abs</td> <td>double[]</td> <td>out</td> <td>絕對座標</td> </tr> <tr> <td>Rel</td> <td>double[]</td> <td>out</td> <td>相對座標</td> </tr> <tr> <td>Dist</td> <td>double[]</td> <td>out</td> <td>剩餘距離</td> </tr> </tbody> </table>					變數名稱	型別	[in/out]	說明	AxisName	string[]	out	各軸座標名稱	DecPoint	short	out	座標小數點位數	Unit	string[]	out	座標單位	Mach	double[]	out	機械座標	Abs	double[]	out	絕對座標	Rel	double[]	out	相對座標	Dist	double[]	out	剩餘距離
變數名稱	型別	[in/out]	說明																																		
AxisName	string[]	out	各軸座標名稱																																		
DecPoint	short	out	座標小數點位數																																		
Unit	string[]	out	座標單位																																		
Mach	double[]	out	機械座標																																		
Abs	double[]	out	絕對座標																																		
Rel	double[]	out	相對座標																																		
Dist	double[]	out	剩餘距離																																		
備註	※ 海德漢控制器不支援相對座標。																																				
適用範圍	0：支援 X：未支援 ?：未實作 <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>					FANUC	三菱	海德漢	西門子	新代	0	0	0	0	0	寶元	工研院				0	0															
FANUC	三菱	海德漢	西門子	新代																																	
0	0	0	0	0																																	
寶元	工研院																																				
0	0																																				
C#程式範例	<pre>StructMsg.position _position; StructMsg.Pwd _Pwd; private void button11_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_position(_Pwd, ref _position); if (ret == 0) { txtAbs.Text = _position.Abs[0].ToString(); txtMach.Text = _position.Mach[0].ToString(); txtRel.Text = _position.Rel[0].ToString(); txtDist.Text = _position.Dist[0].ToString(); } }</pre>																																				

5. GET_gcode : G Code

API 名稱	GET_gcode				
函式說明	G Code				
C#方法	short GET_gcode(StructMsg.Pwd P, ref StructMsg.gcode R);				
VB.Net 方法	GET_gcode(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.gcode) As Short				
資料結構	結構名稱：gcode				
	變數名稱	型別	[in/out]	說明	
	Gdata	string[]	out	G Code	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.gcode _gcode; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_gcode(_Pwd, ref _gcode); if (ret == 0) { for (int i = 0; i <= _gcode.Gdata.Length - 1; i++) { txtGCode.Text += _gcode.Gdata[i] + " "; } } }</pre>				

6. GET_othercode : 其他 Code(H Code,D Code,T Code,M Code,B Code,F Code,S Code)

API 名稱	GET_othercode				
函式說明	G Code (H Code,D Code,T Code,M Code,B Code,F Code,S Code)				
C#方法	short GET_othercode(StructMsg.Pwd P, ref StructMsg.othercode R);				
VB.Net 方法	GET_othercode(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.othercode) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：othercode				

	變數名稱	型別	[in/out]	說明
	HCode	int	out	H Code
	DCode	int	out	D Code
	TCode	int	out	T Code
	MCode	int	out	M Code
	BCode	int	out	B Code, 只有三菱有支援
	FCode	int	out	F Code
	SCode	int	out	S Code
備註	※ 只有三菱有支援 B Code ※ 海德漢控制器只支援 M Code、F Code、T Code、S Code			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	0	?
	寶元	工研院		
	0	0		
C#程式範例	<pre> StructMsg.othercode _othercode; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_othercode(_Pwd, ref _othercode); if (ret == 0) { txtHCode.Text = _othercode.HCode.ToString(); txtDCode.Text = _othercode.DCode.ToString(); txtTCode.Text = _othercode.TCode.ToString(); txtMCode.Text = _othercode.MCode.ToString(); txtBCode.Text = _othercode.BCode.ToString(); txtFCode.Text = _othercode.FCode.ToString(); txtSCode.Text = _othercode.SCode.ToString(); } } </pre>			

7. GET_feed_spindle：進給率/轉速

API 名稱	GET_feed_spindle
函式說明	進給率/轉速
C#方法	short GET_feed_spindle(StructMsg.Pwd P, ref StructMsg.feed_spindle R);
VB.Net 方法	GET_feed_spindle(ByVal P As StructMsg.Pwd, ByRef R As

	StructMsg.feed_spindle) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：feed_spindle				
	變數名稱	型別	[in/out]	說明	
	OvFeed	double	out	Feed Rate Over Ride	
	OvSpindle	double	out	Spindle Over Ride	
ActFeed	double	out	實際進給率		
ActSpindle	int	out	實際主軸轉速		
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.feed_spindle _feed_spindle; StructMsg.Pwd _Pwd; private void button5_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_feed_spindle(_Pwd, ref _feed_spindle); if (ret == 0) { txtOvFeed.Text = _feed_spindle.OvFeed.ToString(); txtOvSpindle.Text = _feed_spindle.OvSpindle.ToString(); txtActFeed.Text = _feed_spindle.ActFeed.ToString(); txtActSpindle.Text = _feed_spindle.ActSpindle.ToString(); } }</pre>				

8. GET_time：時間

API 名稱	GET_time			
函式說明	取得 CNC 相關時間			
C#方法	short GET_time(StructMsg.Pwd P, ref StructMsg.time R);			
VB.Net 方法	GET_time(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.time) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：time			
	變數名稱	型別	[in/out]	說明
Power	int[]	out	開機時間	

	Cutting	int[]	out	切削時間
	Cycle	int[]	out	CYCLE 時間
	Operation	int[]	out	操作時間
備註	※ Cutting time:三菱不支援。 ※ time 資料結構裡面的所有變數，固定都有 3 個陣列，陣列中索引 0 為 hour，索引 1 為 minuite，索引 2 為 second。			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	X	?
	寶元	工研院		
	X	0		
C#程式範例	<pre>StructMsg.time _time; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_time(_Pwd, ref _time); if (ret == 0) { txtPower.Text = _time.Power[0].ToString() + " : " + _time.Power[1].ToString() + " : " + _time.Power[2].ToString(); txtCutting.Text = _time.Cutting[0].ToString() + " : " + _time.Cutting[1].ToString() + " : " + _time.Cutting[2].ToString(); txtCycle.Text = _time.Cycle[0].ToString() + " : " + _time.Cycle[1].ToString() + " : " + _time.Cycle[2].ToString(); txtOperation.Text = _time.Operation[0].ToString() + " : " + _time.Operation[1].ToString() + " : " + _time.Operation[2].ToString(); } }</pre>			

9. GET_time_heid：海德漢控制器_時間

API 名稱	GET_time_heid
函式說明	取得 CNC 相關時間
C#方法	short GET_time_heid(StructMsg.Pwd P, ref StructMsg.time_heid R);
VB.Net 方法	GET_time(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.time_heid) As Short
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：time_heid

	變數名稱	型別	[in/out]	說明	
	MachineRunning	int[]	out	This is the cumulative machining time since installation.	
	MachineUp	int[]	out	This is the cumulative time that the machine has been on (no emergency stop) since installation.	
	NcUp	int[]	out	This is the cumulative time that the NC has been turned on since installation of the machine.	
	SpindleRunning	int[]	out	This is the cumulative time the spindle has been running (M3 or M4) since installation of the machine.	
備註					
適用範圍	O : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	O	X	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.time_heid _time_heid; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_time_heid(_Pwd, ref _time_heid); if (ret == 0) { txtMachineRunning.Text = _time_heid.MachineRunning[0] + ":" + _time_heid.MachineRunning[1]; txtMachineUp.Text = _time_heid.MachineUp[0] + ":" + _time_heid.MachineUp[1]; txtNcUp.Text = _time_heid.NcUp[0] + ":" + _time_heid.NcUp[1]; txtSpindleRunning.Text = _time_heid.SpindleRunning[0] + ":" + _time_heid.SpindleRunning[1]; } }</pre>				

	}
--	---

10. GET_time_cnc：取得控制器系統時間

API 名稱	GET_time_cnc				
函式說明	取得控制器系統時間				
C#方法	short GET_time_cnc(StructMsg.Pwd P, ref StructMsg.time_cnc R);				
VB.Net 方法	GET_time_cnc(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.time_cnc) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：time_cnc				
	變數名稱	型別	[in/out]	說明	
	SystemTime	short[]	out	控制器系統日期時間，會傳回 6 個陣列如下： [0]:年,[1]:月,[2]:日,[3]:時,[4]:分,[5]:秒	
備註	※ 海德漢控制器目前尚不支援。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	X
	寶元	工研院			
	X	0			
C#程式範例	<pre>StructMsg.time_cnc _time_cnc; StructMsg.Pwd _Pwd; private void button10_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_time_cnc(_Pwd, ref _time_cnc); txtSysTime.Text = _time_cnc.SystemTime[0] + "/" + _time_cnc.SystemTime[1] + "/" + _time_cnc.SystemTime[2] + " " + _time_cnc.SystemTime[3] + ":" + _time_cnc.SystemTime[4] + ":" + _time_cnc.SystemTime[5]; }</pre>				

11. GET_part_count：工件數

API 名稱	GET_part_count
函式說明	工件數
C#方法	short GET_part_count(StructMsg.Pwd P, ref StructMsg.part_count R);
VB.Net 方法	GET_part_count(ByVal P As StructMsg.Pwd, ByRef R As

	StructMsg.part_count) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：part_count				
	變數名稱	型別	[in/out]	說明	
	part_count	int	out	工件數	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	0	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.part_count _part_count; StructMsg.Pwd _Pwd; private void button7_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_part_count(_Pwd, ref _part_count); if (ret == 0) { txtPartCount.Text = _part_count.PartCount.ToString(); } }</pre>				

12. GET_part_total：

API 名稱	GET_part_total				
函式說明	總工件數				
C#方法	short GET_part_total(StructMsg.Pwd P, ref StructMsg.part_total R);				
VB.Net 方法	GET_part_total(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.part_total) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：part_total				
	變數名稱	型別	[in/out]	說明	
	PartTotal	int	out		
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	X	X	X	0
	寶元	工研院			

	X	X			
C#程式範例	<pre>StructMsg.part_total _part_total; StructMsg.Pwd _Pwd; private void button8_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_part_total(_Pwd, ref _part_total); if (ret == 0) { txtPartTotal.Text = _part_total.PartTotal.ToString(); } }</pre>				

13. GET_part_required :

API 名稱	GET_part_required				
函式說明	需求工件數				
C#方法	short GET_part_required(StructMsg.Pwd P, ref StructMsg.part_required R);				
VB.Net 方法	GET_part_required(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.part_required) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：part_required				
	變數名稱	型別	[in/out]	說明	
	PartRequired	int	out		
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	X	X	X	0
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.part_required _part_required; StructMsg.Pwd _Pwd; private void button9_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_part_required(_Pwd, ref _part_required); if (ret == 0) {</pre>				

	<pre> txtPartRequired.Text = _part_required.PartRequired.ToString(); } } </pre>
--	---

14. SET_relpos：設定相對座標數值

API 名稱	SET_relpos			
函式說明	設定相對座標數值			
C#方法	short SET_relpos(StructMsg.Pwd P, StructMsg.relpos R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：relpos			
	變數名稱	型別	[in/out]	說明
	AxisName	string	in	座標名稱
	PosValue	double	in	寫入到控制器的座標數值
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	?	?	?
	寶元	工研院		
	?	?		
C#程式範例	<pre> StructMsg.Pwd _Pwd; StructMsg.relpos _relpos; private void button12_Click(object sender, EventArgs e) { _Pwd.WritePwd = "123"; _relpos.AxisName = "X"; _relpos.PosValue = 0; short ret = iRemoting.SET_relpos(_Pwd, _relpos); _relpos.AxisName = "Y"; _relpos.PosValue = 0; ret = iRemoting.SET_relpos(_Pwd, _relpos); _relpos.AxisName = "Z"; _relpos.PosValue = 0; ret = iRemoting.SET_relpos(_Pwd, _relpos); </pre>			

	}
--	---

15. SET_time_cnc：設定控制器的時間

API 名稱	SET_time_cnc			
函式說明	設定控制器的時間			
C#方法	short SET_time_cnc(StructMsg.Pwd P, StructMsg.time_cnc R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：time_cnc			
	變數名稱	型別	[in/out]	說明
	SystemTime	short[]	in	寫入到控制器系統日期時間，必須有 6 個陣列如下： [0]:年,[1]:月,[2]:日,[3]:時,[4]:分,[5]:秒
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	?	?	?
	寶元	工研院		
	?	?		
C#程式範例	<pre>StructMsg.time_cnc _time_cnc = new StructMsg.time_cnc(); _time_cnc.SystemTime = new short[]{(short)DateTime.Now.Year, (short)DateTime.Now.Month, (short)DateTime.Now.Day, (short)DateTime.Now.Hour, (short)DateTime.Now.Minute, (short)DateTime.Now.Second}; short ret = iRemoting.SET_time_cnc(_Pwd, _time_cnc);</pre>			

八、警報類別函式

1. GET_alm_current：目前發生的警報

API 名稱	GET_alm_current				
函式說明	目前發生的警報				
C#方法	short GET_alm_current(StructMsg.Pwd P, ref StructMsg.alm_current R);				
VB.Net 方法	GET_alm_current(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.alm_current) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：alm_current				
	變數名稱	型別	[in/out]	說明	
	IsAlarm	bool	out	判斷是否有警報	
	AlmCode	int	out	取得警報編號	
AlmMsg	string	out	取得目前發生的警報訊息		
備註	※ 警報發生時，AlmMsg 是屬於空白，可能資料庫尚未建立。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.alm_current _alm_current; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_alm_current(_Pwd, ref _alm_current); if (ret == 0 && _alm_current.IsAlarm) { txtAlmsg.Text = _alm_current.Almsg; } }</pre>				

2. GET_alm_current2：目前發生的警報

API 名稱	GET_alm_current2
函式說明	目前發生的警報
C#方法	short GET_alm_current2(StructMsg.Pwd P, ref StructMsg.alm_current2 R);
VB.Net 方法	GET_alm_current2(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.alm_current2) As Short

資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：alm_current2</p> <table border="1" data-bbox="480 248 1398 495"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>IsAlarm</td> <td>bool</td> <td>out</td> <td>判斷是否有警報</td> </tr> <tr> <td>AlmClass</td> <td>string[]</td> <td>out</td> <td>警報類別</td> </tr> <tr> <td>AlmCode</td> <td>string[]</td> <td>out</td> <td>警報編號</td> </tr> <tr> <td>AlmMsg</td> <td>string[]</td> <td>out</td> <td>目前發生的警報訊息</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	IsAlarm	bool	out	判斷是否有警報	AlmClass	string[]	out	警報類別	AlmCode	string[]	out	警報編號	AlmMsg	string[]	out	目前發生的警報訊息
變數名稱	型別	[in/out]	說明																		
IsAlarm	bool	out	判斷是否有警報																		
AlmClass	string[]	out	警報類別																		
AlmCode	string[]	out	警報編號																		
AlmMsg	string[]	out	目前發生的警報訊息																		
備註	<p>※ 變數 AlmClass(警報類別)適用在：海德漢及新代控制器。 當有「SkyMars 維護管理」或「SkyMars 刀具壽命」警報發生時，所屬類別名稱將會寫入到 AlmClass 變數內： 「SkyMars 維護管理警報」所屬類別名稱為 SkyMarsMaintenance 「SkyMars 刀具壽命警報」所屬類別名稱為 SkyMarsToolManagement</p> <p>※ 變數 AlmCode 適用在：FANUC、海德漢及新代控制器。</p> <p>※ 變數 AlmMsg 適用在：FANUC、三菱、海德漢、新代控制器、SkyMars 維護管理警報及 SkyMars 刀具壽命警報。</p>																				
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1" data-bbox="480 927 1398 1128"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FANUC	三菱	海德漢	西門子	新代	0	0	0	0	0	寶元	工研院				0	0			
FANUC	三菱	海德漢	西門子	新代																	
0	0	0	0	0																	
寶元	工研院																				
0	0																				
C#程式範例	<pre> StructMsg.alm_current2 _alm_current2; StructMsg.Pwd _Pwd; private void button8_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_alm_current2(_Pwd, ref _alm_current2); if (ret == 0) { txtC1.Text = _alm_current2.IsAlarm.ToString(); if (_alm_current2.IsAlarm) { txtC2.Text = _alm_current2.AlmClass[0]; txtC3.Text = _alm_current2.AlmCode[0]; txtC4.Text = _alm_current2.AlmMsg[0]; } } } </pre>																				

3. GET_alm_current_heid：海德漢控制器_目前發生的警報

API 名稱	GET_alm_current_heid				
函式說明	目前發生的警報				
C#方法	short GET_alm_current_heid(StructMsg.Pwd P, ref StructMsg.alm_current_heid R);				
VB.Net 方法	GET_alm_current_heid(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.alm_current_heid) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：alm_current_heid				
	變數名稱	型別	[in/out]	說明	
	IsAlarm	bool	out	判斷是否有警報	
	ErrNumber	int[]	out	Number error.	
	ErrGroup	string[]	out	Group error.	
	ErrClass	string[]	out	Class error.	
	ErrMsg	string[]	out	錯誤訊息。	
ErrDescription	string[]	out	錯誤描述		
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	X	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.alm_current_heid _alm_current_heid; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_alm_current_heid(_Pwd, ref _alm_current_heid); if (ret == 0 && _alm_current_heid.IsAlarm) { txtNumber.Text = _alm_current_heid.ErrNumber[0].ToString(); txtGroup.Text = _alm_current_heid.ErrGroup[0]; txtClass.Text = _alm_current_heid.ErrClass[0]; txtMsg.Text = _alm_current_heid.ErrMsg[0]; txtDescription.Text = _alm_current_heid.ErrDescription[0]; } }</pre>				

4. GET_alm_history：警報履歷

API 名稱	GET_alm_history																								
函式說明	警報履歷																								
C#方法	short GET_alm_history(StructMsg.Pwd P, ref StructMsg.alm_history R);																								
VB.Net 方法	GET_alm_history(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.alm_history) As Short																								
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：alm_history <table border="1" data-bbox="475 555 1396 757"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>AlmCode</td> <td>int[]</td> <td>out</td> <td>警報編號</td> </tr> <tr> <td>AlmDate</td> <td>string[]</td> <td>out</td> <td>警報日期</td> </tr> <tr> <td>AlmMsg</td> <td>string[]</td> <td>out</td> <td>警報訊息</td> </tr> </tbody> </table>					變數名稱	型別	[in/out]	說明	AlmCode	int[]	out	警報編號	AlmDate	string[]	out	警報日期	AlmMsg	string[]	out	警報訊息				
變數名稱	型別	[in/out]	說明																						
AlmCode	int[]	out	警報編號																						
AlmDate	string[]	out	警報日期																						
AlmMsg	string[]	out	警報訊息																						
備註	變數 AlmCode 目前不適用在三菱。																								
適用範圍	0：支援 X：未支援 ?：未實作 <table border="1" data-bbox="475 853 1396 1055"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>X</td> <td>?</td> <td>0</td> </tr> <tr> <td>寶元</td> <td>工研院</td> <td></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>					FANUC	三菱	海德漢	西門子	新代	0	0	X	?	0	寶元	工研院				0	0			
FANUC	三菱	海德漢	西門子	新代																					
0	0	X	?	0																					
寶元	工研院																								
0	0																								
C#程式範例	<pre>StructMsg.alm_history _alm_history; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_alm_history(_Pwd, ref _alm_history); if (ret == 0) { txtAlmCode.Text = _alm_history.AlmCode[0].ToString(); txtAlmDate.Text = _alm_history.AlmDate[0]; txtAlmMsg.Text = _alm_history.AlmMsg[0]; } }</pre>																								

5. GET_alm_history2：警報履歷

API 名稱	GET_alm_history2				
函式說明	警報履歷				
C#方法	short GET_alm_history2(StructMsg.Pwd P, ref StructMsg.alm_history2 R);				
VB.Net 方法	GET_alm_histor2y(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.alm_history2) As Short				

資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：alm_history2</p> <table border="1" data-bbox="480 248 1398 495"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>AlmClass</td> <td>string[]</td> <td>out</td> <td>警報類別</td> </tr> <tr> <td>AlmCode</td> <td>string[]</td> <td>out</td> <td>警報編號</td> </tr> <tr> <td>AlmDate</td> <td>string[]</td> <td>out</td> <td>警報日期</td> </tr> <tr> <td>AlmMsg</td> <td>string[]</td> <td>out</td> <td>警報訊息</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	AlmClass	string[]	out	警報類別	AlmCode	string[]	out	警報編號	AlmDate	string[]	out	警報日期	AlmMsg	string[]	out	警報訊息
變數名稱	型別	[in/out]	說明																		
AlmClass	string[]	out	警報類別																		
AlmCode	string[]	out	警報編號																		
AlmDate	string[]	out	警報日期																		
AlmMsg	string[]	out	警報訊息																		
備註	<p>※ 變數 AlmClass 適用在：海德漢及新代控制器。</p> <p>當有「SkyMars 維護管理」或「SkyMars 刀具壽命」警報發生時，所屬類別名稱將會寫入到 AlmClass 變數內：</p> <p>「SkyMars 維護管理警報」所屬類別名稱為 SkyMarsMaintenance</p> <p>「SkyMars 刀具壽命警報」所屬類別名稱為 SkyMarsToolManagement</p> <p>※ 變數 AlmCode 適用在：FANUC、海德漢及新代控制器。</p> <p>※ 變數 AlmDate 適用在：FANUC、三菱、海德漢及新代控制器。</p> <p>※ 變數 AlmMsg 適用在：FANUC、三菱、海德漢、新代控制器、SkyMars 維護管理警報及 SkyMars 刀具壽命警報。</p>																				
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1" data-bbox="480 976 1398 1173"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>?</td> <td>0</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FANUC	三菱	海德漢	西門子	新代	0	0	0	?	0	寶元	工研院				0	0			
FANUC	三菱	海德漢	西門子	新代																	
0	0	0	?	0																	
寶元	工研院																				
0	0																				
C#程式範例	<pre> StructMsg.alm_history2 _alm_history2; StructMsg.Pwd _Pwd; private void button7_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_alm_history2(_Pwd, ref _alm_history2); if (ret == 0) { if (_alm_history2.AlmClass.Length > 0) { txtH1.Text = _alm_history2.AlmClass[0]; txtH2.Text = _alm_history2.AlmCode[0]; txtH3.Text = _alm_history2.AlmDate[0]; txtH4.Text = _alm_history2.AlmMsg[0]; } } } </pre>																				

6. GET_alm_history_heid：警報履歷_海德漢控制器

API 名稱	GET_alm_history_heid				
函式說明	警報履歷_海德漢控制器				
C#方法	short GET_alm_history_heid(StructMsg.Pwd P, ref StructMsg.alm_history_heid R);				
VB.Net 方法	GET_alm_history_heid(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.alm_history_heid) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：alm_history_heid				
	變數名稱	型別	[in/out]	說明	
	ErrNumber	int[]	out	Number error.	
	ErrGroup	string[]	out	Group error.	
	ErrClass	string[]	out	Class error.	
	ErrMsg	string[]	out	錯誤訊息。	
	ErrDescription	string[]	out	錯誤描述	
ErrDate	string[]	out	警報日期		
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	X	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.alm_history_heid _alm_history_heid; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_alm_history_heid(_Pwd, ref _alm_history_heid); if (ret == 0) { DataTable dtHeid = new DataTable(); dtHeid.Columns.Add("HDate", Type.GetType("System.String")); dtHeid.Columns.Add("ErrNumber", Type.GetType("System.Int32")); dtHeid.Columns.Add("_Group", Type.GetType("System.String")); dtHeid.Columns.Add("_Class", Type.GetType("System.String")); dtHeid.Columns.Add("_Msg", Type.GetType("System.String")); dtHeid.Columns.Add("_Description", Type.GetType("System.String")); } }</pre>				


```

        for (int i = 0; i <= _alm_history_heid.ErrNumber.Length - 1; i++)
        {
            DataRow dr = dtHeid.NewRow();
            dr["HDate"] = _alm_history_heid.ErrDate[i];
            dr["ErrNumber"] = _alm_history_heid.ErrNumber[i];
            dr["_Group"] = _alm_history_heid.ErrGroup[i];
            dr["_Class"] = _alm_history_heid.ErrClass[i];
            dr["_Msg"] = _alm_history_heid.ErrMsg[i];
            dr["_Description"] = _alm_history_heid.ErrDescription[i];
            dtHeid.Rows.Add(dr);
        }

        dataGridView1.DataSource = dtHeid;
    }
}
    
```

7. GET_msg_current：取得 Operation 訊息

API 名稱	GET_msg_current			
函式說明	取得 Operation 訊息			
C#方法	short GET_msg_current(StructMsg.Pwd P, ref StructMsg.msg_current R);			
VB.Net 方法	GET_msg_current(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.msg_current) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：msg_current			
	變數名稱	型別	[in/out]	說明
	IsMsg	bool	out	判斷是否有 Operation 訊息
	MsgCode	short	out	訊息編號
	MsgText	string	out	訊息內容
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	X	X	?
	寶元	工研院		
	X	X		
C#程式範例	StructMsg.msg_current _msg_current; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) {			

	<pre> _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_msg_current(_Pwd, ref _msg_current); if (ret == 0 && _msg_current.IsMsg) { txtMsgCode.Text = _msg_current.MsgCode.ToString(); txtMsgText.Text = _msg_current.MsgText; } } </pre>
--	--

8. GET_msg_history：取得 Operation 履歷

API 名稱	GET_msg_history																							
函式說明	取得 Operation 履歷																							
C#方法	short GET_msg_history(StructMsg.Pwd P, ref StructMsg.msg_history R);																							
VB.Net 方法	GET_msg_history(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.msg_history) As Short																							
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：msg_history <table border="1" data-bbox="478 992 1396 1189"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>MsgCode</td> <td>short[]</td> <td>out</td> <td>訊息編號</td> </tr> <tr> <td>MsgDate</td> <td>string[]</td> <td>out</td> <td>訪息日期</td> </tr> <tr> <td>MsgText</td> <td>string[]</td> <td>out</td> <td>訊息內容</td> </tr> </tbody> </table>				變數名稱	型別	[in/out]	說明	MsgCode	short[]	out	訊息編號	MsgDate	string[]	out	訪息日期	MsgText	string[]	out	訊息內容				
變數名稱	型別	[in/out]	說明																					
MsgCode	short[]	out	訊息編號																					
MsgDate	string[]	out	訪息日期																					
MsgText	string[]	out	訊息內容																					
備註	此函式三菱不支援。																							
適用範圍	0：支援 X：未支援 ?：未實作 <table border="1" data-bbox="478 1288 1396 1485"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>X</td> <td>?</td> <td>X</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>X</td> <td>X</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				FANUC	三菱	海德漢	西門子	新代	0	X	X	?	X	寶元	工研院				X	X			
FANUC	三菱	海德漢	西門子	新代																				
0	X	X	?	X																				
寶元	工研院																							
X	X																							
C#程式範例	<pre> StructMsg.msg_history _msg_history; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_msg_history(_Pwd, ref _msg_history); if (ret == 0) { txtMsgCode2.Text = _msg_history.MsgCode[0].ToString(); txtMsgDate.Text = _msg_history.MsgDate[0]; txtMsgText2.Text = _msg_history.MsgText[0]; } } </pre>																							

	}
--	---

9. GET_plc_alarm：取得 PLC Alarm 履歷

API 名稱	GET_plc_alarm				
函式說明	取得 PLC Alarm 履歷				
C#方法	short GET_plc_alarm(StructMsg.Pwd P, ref StructMsg.plc_alarm R);				
VB.Net 方法	GET_plc_alarm(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.plc_alarm) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：plc_alarm				
	變數名稱	型別	[in/out]	說明	
	AlmMsg	string[]	out	PLC 錯誤訊息	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	X	X	?	X
	寶元	工研院			
	0	?			
C#程式範例	<pre>StructMsg.plc_alarm _plc_alarm; StructMsg.Pwd _Pwd; private void button5_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_plc_alarm(_Pwd, ref _plc_alarm); if (ret == 0) { if (_plc_alarm.AlmMsg != null) txtAlmMsg2.Text = _plc_alarm.AlmMsg[0]; } }</pre>				

九、伺服主軸類別函式

1. GET_servo_current：各軸負載電流值

API 名稱	GET_servo_current				
函式說明	各軸負載電流值				
C#方法	short GET_servo_current(StructMsg.Pwd P, ref StructMsg.servo_current R);				
VB.Net 方法	GET_servo_current(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.servo_current) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：servo_current				
	變數名稱	型別	[in/out]	說明	
	AxisCurrent	int[]	out	目前電流值	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	X	X	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.servo_current _servo_current; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_servo_current(_Pwd, ref _servo_current); if (ret == 0) { txtAxisCurrent.Text = _servo_current.AxisCurrent[0].ToString(); } }</pre>				

2. GET_servo_load：取得伺服負載%

API 名稱	GET_servo_load
函式說明	取得伺服負載%
C#方法	short GET_servo_load(StructMsg.Pwd P, ref StructMsg.servo_load R);
VB.Net 方法	GET_servo_load(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.servo_load) As Short

資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：servo_load				
	變數名稱	型別	[in/out]	說明	
	ServoLoadPercent	int[]	out	負載值	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	X
	寶元	工研院			
	X	0			
C#程式範例	<pre>StructMsg.servo_load _servo_load; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_servo_load(_Pwd, ref _servo_load); if (ret == 0) { for (int i = 0; i <= _servo_load.ServoLoadPercent.Length - 1; i++) txtServoLoadPercent.Text += _servo_load.ServoLoadPercent[i].ToString() + " "; } }</pre>				

3. GET_servo_speed：取得伺服軸轉速

API 名稱	GET_servo_speed				
函式說明	取得伺服軸轉速				
C#方法	short GET_servo_speed(StructMsg.Pwd P, ref StructMsg.servo_speed R);				
VB.Net 方法	GET_servo_speed(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.servo_speed) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：servo_speed				
	變數名稱	型別	[in/out]	說明	
	ServoSpd	int[]	out	伺服軸轉速	
備註					

適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	X
	寶元	工研院			
	X	0			
C#程式範例	<pre>StructMsg.servo_speed _servo_speed; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_servo_speed(_Pwd, ref _servo_speed); if (ret == 0) { txtServoSpd.Text = _servo_speed.ServoSpd[0].ToString(); } }</pre>				

4. GET_spindle_load：取得主軸負載%

API 名稱	GET_spindle_load				
函式說明	取得主軸負載%				
C#方法	short GET_spindle_load(StructMsg.Pwd P, ref StructMsg.spindle_load R);				
VB.Net 方法	GET_spindle_load(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.spindle_load) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：spindle_load				
	變數名稱	型別	[in/out]	說明	
SpLoad	double	out	主軸負載		
備註	只能適用在一顆主軸。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	X
	寶元	工研院			
	X	0			
C#程式範例	<pre>StructMsg.spindle_load _spindle_load; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123";</pre>				

```

short ret = iRemoting.GET_spindle_load(_Pwd, ref _spindle_load);
if (ret == 0)
{
    txtSpLoad.Text = _spindle_load.SpLoad.ToString();
}
else if (ret==6)
{
    MessageBox.Show("There is no corresponding CNC option.");
}
}
    
```

5. GET_spindle_speed：取得主軸轉速

API 名稱	GET_spindle_speed				
函式說明	取得主軸轉速				
C#方法	short GET_spindle_speed(StructMsg.Pwd P, ref StructMsg.spindle_speed R);				
VB.Net 方法	GET_spindle_speed(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.spindle_speed) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：spindle_speed				
	變數名稱	型別	[in/out]	說明	
	SpSpeed	int	out	主軸轉速	
備註	只能適用在一顆主軸。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	X
	寶元	工研院			
	X	0			
C#程式範例	<pre> StructMsg.spindle_speed _spindle_speed; StructMsg.Pwd _Pwd; private void button7_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_spindle_speed(_Pwd, ref _spindle_speed); if (ret == 0) { txtSpindleSpeed.Text = _spindle_speed.SpSpeed.ToString(); } } </pre>				

	}
--	---

6. GET_servo_temperature：取得伺服馬達溫度

API 名稱	GET_servo_temperature			
函式說明	取得伺服馬達溫度			
C#方法	short GET_servo_temperature(StructMsg.Pwd P, ref StructMsg.servo_temperature R);			
VB.Net 方法	GET_servo_temperature(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.servo_temperature) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：servo_temperature			
	變數名稱	型別	[in/out]	說明
	AxisTemp	double[]	out	伺服馬達溫度
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	?	0	?
	寶元	工研院		
	X	X		
C#程式範例	<pre>StructMsg.servo_temperature _servo_temperature; StructMsg.Pwd _Pwd; private void button5_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_servo_temperature(_Pwd, ref _servo_temperature); if (ret == 0) { for (int i = 0; i <= _servo_temperature.AxisTemp.Length - 1; i++) { txtServoTemp.Text += _servo_temperature.AxisTemp[i].ToString() + " "; } } }</pre>			

7. GET_spindle_temperature：取得主軸溫度

API 名稱	GET_spindle_temperature
函式說明	取得主軸溫度

C#方法	<code>short GET_spindle_temperature(StructMsg.Pwd P, ref StructMsg.spindle_temperature R);</code>			
VB.Net 方法	<code>GET_spindle_temperature(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.spindle_temperature) As Short</code>			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：spindle_temperature			
	變數名稱	型別	[in/out]	說明
	Spindle_1_Temp	double	out	第一顆主軸溫度
Spindle_2_Temp	double	out	第二顆主軸溫度	
備註	0i-D / 30 系列，目前只能讀取單一主軸溫度			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	?	0	?
	寶元	工研院		
	X	X		
C#程式範例	<pre>StructMsg.spindle_temperature _spindle_temperature; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_spindle_temperature(_Pwd, ref _spindle_temperature); if (ret == 0) { txtSpindleTemp.Text = _spindle_temperature.Spindle_1_Temp.ToString(); } }</pre>			

十、刀具管理類別函式 (控制器)

1. GET_offset_title：取得刀具 Offset 的標題列

API 名稱	GET_offset_title			
函式說明	取得刀具 Offset 的標題列			
C#方法	short GET_offset_title(StructMsg.Pwd P, ref StructMsg.offset_title R);			
VB.Net 方法	GET_offset_title(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.offset_title) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：offset_title			
	變數名稱	型別	[in/out]	說明
	OffsetTitle	string[]	out	標題列
備註	<p>※以下為 FANUC 標題列</p> <p>M 系列</p> <p>System A: {"DATA"}</p> <p>System B: {"GEOM", "WEAR"}</p> <p>System C: {"LENGTH GEOM", "LENGTH WEAR", "RADIUS GEOM", "RADIUS WEAR"}</p> <p>T 系列</p> <p>沒有Y軸： {"WEAR X", "WEAR Z", "WEAR R", "WEAR T", "GEOM X", "GEOM Z", "GEOM R", "GEOM T"}</p> <p>有Y軸： {"WEAR X", "WEAR Z", "WEAR R", "WEAR T", "WEAR Y", "GEOM X", "GEOM Z", "GEOM R", "GEOM T", "GEOM Y"}</p> <p>※以下為三菱標題列</p> <p>Tool compensation type : 1 {"DATA"}</p> <p>Tool compensation type : 2 {"LENGTH GEOM", "LENGTH WEAR", "RADIUS GEOM", "RADIUS WEAR"}</p> <p>Tool compensation type : 3</p>			

適用範圍	0 : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.offset_title _offset_title; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_offset_title(_Pwd, ref _offset_title); if (ret == 0) { for (int i = 0; i <= _offset_title.OffsetTitle.Length - 1; i++) rtbTitle.Text += _offset_title.OffsetTitle[i] + " "; } }</pre>				

2. GET_offset_all : 取得所有刀具 Offset

API 名稱	GET_offset_all											
函式說明	取得所有刀具 Offset											
C#方法	short GET_offset_all(StructMsg.Pwd P, ref StructMsg.offset_all R);											
VB.Net 方法	GET_offset_all(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.offset_all) As Short											
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：offset_all <table border="1" data-bbox="475 1429 1401 1529"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>OffsetData</td> <td>double[][]</td> <td>out</td> <td>補正資料</td> </tr> </tbody> </table>				變數名稱	型別	[in/out]	說明	OffsetData	double[][]	out	補正資料
變數名稱	型別	[in/out]	說明									
OffsetData	double[][]	out	補正資料									
備註	OffsetData 第二個陣列，請配合標題名稱(Length Geom...)。 使用此函式可能會有大量的讀取時間(一般都發生在低速網路卡)，請使用 GET_offset_scope 函式來分段式讀取。											
適用範圍	0 : 支援 X : 未支援 ? : 未實作											
	FANUC	三菱	海德漢	西門子	新代							
	0	0	X	?	0							
	寶元	工研院										
	0	0										
C#程式範例	<pre>StructMsg.offset_all _offset_all; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e)</pre>											

	<pre> { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_offset_all(_Pwd, ref _offset_all); if (ret == 0) { txtLGeom.Text = _offset_all.OffsetData[0][0].ToString(); txtLWear.Text = _offset_all.OffsetData[0][1].ToString(); txtRGeom.Text = _offset_all.OffsetData[0][2].ToString(); txtRWear.Text = _offset_all.OffsetData[0][3].ToString(); } } </pre>
--	---

3. GET_offset_scope：依指定範圍取得刀具 Offset

API 名稱	GET_offset_scope				
函式說明	依指定範圍取得刀具 Offset				
C#方法	short GET_offset_scope(StructMsg.Pwd P, ref StructMsg.offset_scope R);				
VB.Net 方法	GET_offset_scope(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.offset_scope) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：offset_scope				
	變數名稱	型別	[in/out]	說明	
	StartNumber	short	in	起始號碼(從 1 開始)	
EndNumber	short	in	終止號碼		
OffsetData	double[][]	out	補正資料		
備註	OffsetData 第二個陣列，請配合標題名稱(Length Geom...)。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
0	0				
C#程式範例	<pre> StructMsg.offset_scope _offset_scope; StructMsg.Pwd _Pwd; private void button7_Click(object sender, EventArgs e) { _offset_scope.StartNumber = 1; _offset_scope.EndNumber = 26; _Pwd.ConnectionKey = "123"; </pre>				

```

short ret = iRemoting.GET_offset_scope(_Pwd, ref _offset_scope);
if (ret == 0)
{
    DataTable dt = new DataTable();
    //LENGTH GEOM", "LENGTH WEAR", "RADIUS GEOM", "RADIUS WEAR
    dt.Columns.Add("LGeom", Type.GetType("System.Double"));
    dt.Columns.Add("LWrar", Type.GetType("System.Double"));
    dt.Columns.Add("RGeom", Type.GetType("System.Double"));
    dt.Columns.Add("RWear", Type.GetType("System.Double"));

    DataRow dr;
    for (int i = 0; i <= _offset_scope.OffsetData.Length - 1; i++)
    {
        dr = dt.NewRow();
        dr[0] = _offset_scope.OffsetData[i][0];
        dr[1] = _offset_scope.OffsetData[i][1];
        dr[2] = _offset_scope.OffsetData[i][2];
        dr[3] = _offset_scope.OffsetData[i][3];
        dt.Rows.Add(dr);
    }

    dataGridView1.DataSource = dt;
}
}

```

4. GET_offset_single : 取得單筆刀具 Offset

API 名稱	GET_offset_single		
函式說明	取得單筆刀具 Offset		
C#方法	short GET_offset_single(StructMsg.Pwd P, ref StructMsg.offset_single R);		
VB.Net 方法	GET_offset_single(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.offset_single) As Short		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：offset_single		
	變數名稱	型別	[in/out] 說明
	ofNumber	short	in 設定補正編號
OffsetData	double[]	out 補正資料	
備註			

適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.offset_single _offset_single; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _offset_single.ofNumber = 1; //Set offset number short ret = iRemoting.GET_offset_single(_Pwd, ref _offset_single); if (ret == 0) { txtLGeom2.Text = _offset_single.OffsetData[0].ToString(); txtLWear2.Text = _offset_single.OffsetData[1].ToString(); txtRGeom2.Text = _offset_single.OffsetData[2].ToString(); txtRWear2.Text = _offset_single.OffsetData[3].ToString(); } }</pre>				

5. SET_offset_all：寫入所有刀具 Offset

API 名稱	SET_offset_all				
函式說明	寫入所有刀具 Offset				
C#方法	short SET_offset_all(StructMsg.Pwd P, StructMsg.offset_all R);				
VB.Net 方法	SET_offset_all(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.offset_all) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：offset_all				
	變數名稱	型別	[in/out]	說明	
OffsetData	double[][]	in	設定補正資料		
備註	※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	StructMsg.offset_all _offset_all;				

```

StructMsg.Pwd _Pwd;
private void button5_Click(object sender, EventArgs e)
{
    _Pwd.ConnectionKey = "123";
    short ret = iRemoting.GET_offset_all(_Pwd, ref _offset_all);
    _offset_all.OffsetData[0][0] = double.Parse(txtLGeom3.Text);
    _offset_all.OffsetData[0][1] = double.Parse(txtLWear3.Text);
    _offset_all.OffsetData[0][2] = double.Parse(txtRGeom3.Text);
    _offset_all.OffsetData[0][3] = double.Parse(txtRWear3.Text);

    ret = iRemoting.SET_offset_all(_offset_all);
    if (ret == 0)
        MessageBox.Show("Write success!");
    else
        MessageBox.Show("Write failed!");
}
    
```

6. SET_offset_single：寫入單筆刀具 Offset

API 名稱	SET_offset_single			
函式說明	寫入單筆刀具 Offset			
C#方法	short SET_offset_single(StructMsg.Pwd P, StructMsg.offset_single R);			
VB.Net 方法	SET_offset_single(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.offset_single) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：offset_single			
	變數名稱	型別	[in/out]	說明
	ofNumber	short	in	設定補正編號
	OffsetData	double[]	in	設定補正資料
備註	※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	X	?
	寶元	工研院		
	0	0		
C#程式範例	StructMsg.offset_single _offset_single; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) {			

```

_Pwd.ConnectionKey = "123";
_offset_single.ofNumber = 1;
_offset_single.OffsetData = new double[4];
_offset_single.OffsetData[0] = double.Parse(txtLGeom4.Text);
_offset_single.OffsetData[1] = double.Parse(txtLWear4.Text);
_offset_single.OffsetData[2] = double.Parse(txtRGeom4.Text);
_offset_single.OffsetData[3] = double.Parse(txtRWear4.Text);

short ret = iRemoting.SET_offset_single(_Pwd, _offset_single);
if (ret == 0)
    MessageBox.Show("Write success!");
else
    MessageBox.Show("Write failed!");
}
    
```

7. GET_offset_count：取得刀具 Offset 筆數

API 名稱	GET_offset_count				
函式說明	取得刀具 Offset 筆數				
C#方法	short GET_offset_count(StructMsg.Pwd P, ref StructMsg.total_count R);				
VB.Net 方法	GET_offset_count(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.total_count) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：total_count				
	變數名稱	型別	[in/out]	說明	
	Count	short	out	補正筆數	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	StructMsg.total_count _total_count; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_offset_count(_Pwd, ref _total_count); if (ret == 0)				

	<pre> { txtOffsetCount.Text = _total_count.Count.ToString(); } } </pre>
--	---

8. GET_tool_title：取得刀具管理的標題列

API 名稱	GET_tool_title				
函式說明	取得刀具管理的標題列				
C#方法	short GET_tool_title(StructMsg.Pwd P, ref StructMsg.tool_title R);				
VB.Net 方法	GET_tool_title(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.tool_title) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：tool_title				
	變數名稱	型別	[in/out]	說明	
	ToolTitle	string[]	out	標題列	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre> StructMsg.tool_title _tool_title; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_tool_title(_Pwd, ref _tool_title); if (ret == 0) { for (int i = 0; i <= _tool_title.ToolTitle.Length - 1; i++) { rtbTitle.Text += _tool_title.ToolTitle[i] + "\r\n"; } } } </pre>				

9. GET_tool_count：取得刀具 Offset 筆數

API 名稱	GET_tool_count
函式說明	取得刀具管理筆數

C#方法	<code>short GET_tool_count(StructMsg.Pwd P, ref StructMsg.total_count R);</code>				
VB.Net 方法	<code>GET_tool_count(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.total_count) As Short</code>				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：total_count				
	變數名稱	型別	[in/out]	說明	
	Count	short	out	刀具筆數	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.total_count _total_count; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_tool_count(_Pwd, ref _total_count); if (ret == 0) { MessageBox.Show(_total_count.Count.ToString()); } }</pre>				

10. GET_tool_data：取得所有刀具管理資料

API 名稱	GET_tool_data			
函式說明	取得所有刀具管理資料			
C#方法	<code>short GET_tool_data(StructMsg.Pwd P, ref StructMsg.tool_data R);</code>			
VB.Net 方法	<code>GET_tool_data(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.tool_data) As Short</code>			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：tool_data			
	變數名稱	型別	[in/out]	說明
	Data	string[][]	out	刀具管理資料
備註	※ Data 第二個陣列，請配合標題名稱。			

適用範圍	0 : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre> StructMsg.tool_data _tool_data; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_tool_data(_Pwd, ref _tool_data); if (ret == 0) { DataTable dtTool = new DataTable(); for (int i = 0; i <= _tool_title.ToolTitle.Length - 1; i++) { dtTool.Columns.Add(_tool_title.ToolTitle[i], Type.GetType("System.String")); } for (int i = 0; i <= _tool_data.Data.Length - 1; i++) { DataRow dr = dtTool.NewRow(); for (int j = 0; j <= _tool_title.ToolTitle.Length - 1; j++) { dr[j] = _tool_data.Data[i][j]; } dtTool.Rows.Add(dr); } dgvTool.DataSource = dtTool; } } </pre>				

11. GET_tool_data_mem : 取得刀具管理，並將資料讀取到 SkyMars 記憶體中

API 名稱	GET_tool_data_mem
函式說明	將資料讀取到 SkyMars 記憶體中，並回傳資料陣列最大索引值
C#方法	short GET_tool_data_mem(StructMsg.Pwd P, ref StructMsg.tool_data_max_index R);

VB.Net 方法																					
資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：tool_data_max_index</p> <table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>MaxIndex</td> <td>int</td> <td>out</td> <td>資料陣列最大索引值</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	MaxIndex	int	out	資料陣列最大索引值												
變數名稱	型別	[in/out]	說明																		
MaxIndex	int	out	資料陣列最大索引值																		
備註	如果您需要分段讀取刀具管理資料，請先將執行 GET_tool_data_mem 函式，再使用 GET_tool_data_scope 函式，並指定索引值範圍將資料取回。																				
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>X</td> <td>0</td> <td>?</td> <td>X</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>X</td> <td>X</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FANUC	三菱	海德漢	西門子	新代	X	X	0	?	X	寶元	工研院				X	X			
FANUC	三菱	海德漢	西門子	新代																	
X	X	0	?	X																	
寶元	工研院																				
X	X																				
C#程式範例	<pre>StructMsg.tool_data_max_index _ToolMaxIndex; StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; ret = iRemoting.GET_tool_data_mem(_Pwd, ref _ToolMaxIndex);</pre>																				

12. GET_tool_data_scope：取刀具管理資料，並依指定索引值範圍傳回

API 名稱	GET_tool_data_scope																				
函式說明	刀具管理資料，並依指定索引值範圍傳回																				
C#方法	short GET_tool_data_scope(StructMsg.Pwd P, ref StructMsg.tool_data_scope R);																				
VB.Net 方法																					
資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：tool_data_scope</p> <table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>StartIndex</td> <td>int</td> <td>in</td> <td>起始索引值</td> </tr> <tr> <td>EndIndex</td> <td>int</td> <td>in</td> <td>結束索引值</td> </tr> <tr> <td>Data</td> <td>string[][]</td> <td>out</td> <td>刀具管理資料</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	StartIndex	int	in	起始索引值	EndIndex	int	in	結束索引值	Data	string[][]	out	刀具管理資料				
變數名稱	型別	[in/out]	說明																		
StartIndex	int	in	起始索引值																		
EndIndex	int	in	結束索引值																		
Data	string[][]	out	刀具管理資料																		
備註																					
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>X</td> <td>0</td> <td>?</td> <td>X</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>X</td> <td>X</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FANUC	三菱	海德漢	西門子	新代	X	X	0	?	X	寶元	工研院				X	X			
FANUC	三菱	海德漢	西門子	新代																	
X	X	0	?	X																	
寶元	工研院																				
X	X																				
C#程式範例	<pre>StructMsg.tool_data_scope _DataScope; StructMsg.Pwd _Pwd; for (int i = 0; i <= _ToolMaxIndex.MaxIndex; i += 20)</pre>																				

```

{
    //一次讀取20筆
    _DataScope.StartIndex = i;

    if ((i + 20) <= _ToolMaxIndex.MaxIndex)
        _DataScope.EndIndex = i + 20 - 1;
    else
        _DataScope.EndIndex = _ToolMaxIndex.MaxIndex;

    _Pwd.ConnectionKey = "123";
    ret = iRemoting.GET_tool_data_scope(_Pwd, ref _DataScope);
    if (ret == 0)
    {
        for (int j = 0; j <= _DataScope.Data.Length - 1; j++)
        {
            DataRow dr = dtTool.NewRow();
            for (int s = 0; s <= _tool_title.ToolTitle.Length - 1; s++)
            {
                dr[s] = _DataScope.Data[j][s];
            }
            dtTool.Rows.Add(dr);
        }
    }
}
    
```

13. SET_tool_data：寫入刀具管理

API 名稱	SET_tool_data		
函式說明	寫入刀具管理		
C#方法	short SET_tool_data(StructMsg.Pwd P, StructMsg.tool_data R);		
VB.Net 方法	SET_tool_data(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.tool_data) As Short		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：tool_data		
	變數名稱	型別	[in/out] 說明
	Data	string[][]	in 設定刀具管理資料
備註	※ 可針對某幾筆資料或全部資料，進行寫入更新。 ※ 針對某幾筆資料更新時，第[][0]個陣列可指定編號，如範例。 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。		

適用範圍	O : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	O	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _tool_data.Data = new string[1][]; _tool_data.Data[0] = new string[_tool_title.ToolTitle.Length]; DataTable dtTool = (DataTable)dgvTool.DataSource; for (int i = 0; i <= _tool_title.ToolTitle.Length - 1; i++) { _tool_data.Data[0][i] = dtTool.Rows[0][i].ToString(); } _Pwd.ConnectionKey = "123"; _Pwd.WritePwd = "123"; short ret = iRemoting.SET_tool_data(_Pwd, _tool_data); }</pre>				

14. GET_pocket_title : 取得刀庫表的標題列

API 名稱	GET_pocket_title				
函式說明	取得刀庫表的標題列				
C#方法	short GET_pocket_title(StructMsg.Pwd P, ref StructMsg.pocket_title R);				
VB.Net 方法	GET_pocket_title(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.pocket_title) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：pocket_title				
	變數名稱	型別	[in/out]	說明	
	PocketTitle	string[]	out	標題列	
備註					
適用範圍	O : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	O	?	X
	寶元	工研院			
	X	X			

C#程式範例	<pre>StructMsg.pocket_title _pocket_title; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_pocket_title(_Pwd, ref _pocket_title); if (ret == 0) { for (int i = 0; i <= _pocket_title.PocketTitle.Length - 1; i++) { rtbTitle.Text += _pocket_title.PocketTitle[i] + "\r\n"; } } }</pre>
--------	---

15. GET_pocket_count：取得刀庫表筆數

API 名稱	GET_pocket_count			
函式說明	取得刀庫表筆數			
C#方法	short GET_pocket_count(StructMsg.Pwd P, ref StructMsg.total_count R);			
VB.Net 方法	GET_pocket_count(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.total_count) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：total_count			
	變數名稱	型別	[in/out]	說明
	Count	short	out	刀具筆數
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	X	X	0	?
	寶元	工研院		
	X	X		
C#程式範例	<pre>StructMsg.total_count _total_count; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_pocket_count(_Pwd, ref _total_count); if (ret == 0)</pre>			

```

{
    MessageBox.Show(_total_count.Count.ToString());
}
}
    
```

16. GET_pocket_data：取得刀庫表資料

API 名稱	GET_pocket_data				
函式說明	取得刀庫表資料				
C#方法	short GET_pocket_data(StructMsg.Pwd P, ref StructMsg.pocket_data R);				
VB.Net 方法	GET_pocket_data(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.pocket_data) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：pocket_data				
	變數名稱	型別	[in/out]	說明	
Data	string[][]	out	刀具管理資料		
備註	※ Data 第二個陣列，請配合標題名稱。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre> StructMsg.pocket_data _pocket_data; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_pocket_data(_Pwd, ref _pocket_data); if (ret == 0) { DataTable dtPocket = new DataTable(); for (int i = 0; i <= _pocket_title.PocketTitle.Length - 1; i++) { dtPocket.Columns.Add(_pocket_title.PocketTitle[i], Type.GetType("System.String")); } for (int i = 0; i <= _pocket_data.Data.Length - 1; i++) { DataRow dr = dtPocket.NewRow(); </pre>				


```

        for (int j = 0; j <= _pocket_title.PocketTitle.Length - 1;
j++)
        {
            dr[j] = _pocket_data.Data[i][j];
        }
        dtPocket.Rows.Add(dr);
    }

    dgvTool.DataSource = dtPocket;
}
}

```

17. SET_pocket_data：寫入刀庫表

API 名稱	SET_pocket_data			
函式說明	寫入刀庫表			
C#方法	short SET_pocket_data(StructMsg.Pwd P, StructMsg.pocket_data R);			
VB.Net 方法	SET_pocket_data(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.pocket_data) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：pocket_data			
	變數名稱	型別	[in/out]	說明
	Data	string[][]	in	設定刀具管理資料
備註	※ 可針對某幾筆資料或全部資料，進行寫入更新。 ※ 針對某幾筆資料更新時，第[][0]個陣列可指定編號，如範例。 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	X	X	0	?
	寶元	工研院		
	X	X		
C#程式範例	<pre> _pocket_data.Data = new string[1][]; _pocket_data.Data[0] = new string[_pocket_title.PocketTitle.Length]; DataTable dtTool = (DataTable)dgvTool.DataSource; for (int i = 0; i <= _pocket_title.PocketTitle.Length - 1; i++) { _pocket_data.Data[0][i] = dtTool.Rows[0][i].ToString(); } </pre>			

	<pre>StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; _Pwd.WritePwd = "123"; short ret = iRemoting.SET_pocket_data(_pocket_data);</pre>
--	--

18. GET_pocket_data_mem：取得刀庫表，並將資料讀取到 SkyMars 記憶體中

API 名稱	GET_pocket_data_mem			
函式說明	將資料讀取到 SkyMars 記憶體中，並回傳資料陣列最大索引值			
C#方法	short GET_pocket_data_mem(StructMsg.Pwd P, ref StructMsg.pocket_data_max_index R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：pocket_data_max_index			
	變數名稱	型別	[in/out]	說明
	MaxIndex	int	out	資料陣列最大索引值
備註	如果您需要分段讀取刀具管理資料，請先將執行 GET_pocket_data_mem 函式，再使用 GET_pocket_data_scope 函式，並指定索引值範圍將資料取回。			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	X	X	0	?
	寶元	工研院		
	X	X		
C#程式範例	<pre>StructMsg.pocket_data_max_index _PocketMaxIndex; StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; ret = iRemoting.GET_pocket_data_mem(_Pwd, ref _PocketMaxIndex);</pre>			

19. GET_pocket_data_scope：取刀庫表資料，並依指定索引值範圍傳回

API 名稱	GET_pocket_data_scope			
函式說明	刀庫表資料，並依指定索引值範圍傳回			
C#方法	short GET_pocket_data_scope(StructMsg.Pwd P, ref StructMsg.tool_data_scope R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：pocket_data_scope			
	變數名稱	型別	[in/out]	說明
	StartIndex	int	in	起始索引值
	EndIndex	int	in	結束索引值

	Data	string[][]	out	刀具管理資料	
備註					
適用範圍	0 : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre> StructMsg.pocket_data_scope _DataScope; StructMsg.Pwd _Pwd; for (int i = 0; i <= _PocketMaxIndex.MaxIndex; i += 20) { //一次讀取20筆 _DataScope.StartIndex = i; if ((i + 20) <= _PocketMaxIndex.MaxIndex) _DataScope.EndIndex = i + 20 - 1; else _DataScope.EndIndex = _PocketMaxIndex.MaxIndex; _Pwd.ConnectionKey = "123"; ret = iRemoting.GET_pocket_data_scope(_Pwd, ref _DataScope); if (ret == 0) { for (int j = 0; j <= _DataScope.Data.Length - 1; j++) { DataRow dr = dtTool.NewRow(); for (int s = 0; s <= _pocket_title.PocketTitle.Length - 1; s++) { dr[s] = _DataScope.Data[j][s]; } dtTool.Rows.Add(dr); } } } </pre>				

十一、 工件座標類別函式

1. GET_work_coord_title：取得工件座標的標題列

API 名稱	GET_work_coord_title			
函式說明	取得工件座標的標題列			
C#方法	short GET_work_coord_title(StructMsg.Pwd P, ref StructMsg.work_coord_title R);			
VB.Net 方法	GET_work_coord_title(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.work_coord_title) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：work_coord_title			
	變數名稱	型別	[in/out]	說明
	WorkCoordTitle	string[]	out	標題列
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	X	?
	寶元	工研院		
	0	0		
C#程式範例	<pre>StructMsg.work_coord_title _work_coord_title; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_work_coord_title(_Pwd, ref _work_coord_title); if (ret == 0) { for (int i = 0; i <= _work_coord_title.WorkCoordTitle.Length - 1; i++) rtbTitle.Text += _work_coord_title.WorkCoordTitle[i] + " "; } }</pre>			

2. GET_work_coord_all：取得所有工件座標資料

API 名稱	GET_work_coord_all			
函式說明	取得所有工件補正資料			
C#方法	short GET_work_coord_all(StructMsg.Pwd P, ref StructMsg.work_coord_all R);			
VB.Net 方法	GET_work_coord_all(ByVal P As StructMsg.Pwd, ByRef R As			

	StructMsg.work_coord_all) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：work_coord_all				
	變數名稱	型別	[in/out]	說明	
	CoordName	string[]	out	項目,ex:EXT,G54,G55...	
	WorkCoord	double[][]	out	工件座標資料	
備註	CoordName 項目如： EXT, G54, G55, G56, G57, G58, G59 , G54.1P1, G54.1P2, G54.1P3 等 使用此函式可能會有大量的讀取時間(一般都發生在低速網路卡)，請使用 GET_work_coord_scope 函式來分段讀取。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.work_coord_all _work_coord_all; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_work_coord_all(_Pwd, ref _work_coord_all); if (ret == 0) { //EXT Work Coord txtItem.Text = _work_coord_all.CoordName[0]; txtX.Text = _work_coord_all.WorkCoord[0][0].ToString(); txtY.Text = _work_coord_all.WorkCoord[0][1].ToString(); txtZ.Text = _work_coord_all.WorkCoord[0][2].ToString(); } }</pre>				

3. GET_work_coord_scope：依指定範圍取得工件座標資料

API 名稱	GET_work_coord_scope
函式說明	依指定範圍取得工件補正資料
C#方法	short GET_work_coord_scope(StructMsg.Pwd P, ref StructMsg.work_coord_scope R);
VB.Net 方法	GET_work_coord_scope(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.work_coord_scope) As Short

資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：work_coord_scope</p> <table border="1" data-bbox="480 248 1402 495"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>StartNumber</td> <td>short</td> <td>in</td> <td>起始編號(從 1 開始)</td> </tr> <tr> <td>EndNumber</td> <td>short</td> <td>in</td> <td>終止編號</td> </tr> <tr> <td>CoordName</td> <td>string[]</td> <td>out</td> <td>項目,ex:EXT,G54,G55...</td> </tr> <tr> <td>WorkCoord</td> <td>double[][]</td> <td>out</td> <td>工件座標資料</td> </tr> </tbody> </table>					變數名稱	型別	[in/out]	說明	StartNumber	short	in	起始編號(從 1 開始)	EndNumber	short	in	終止編號	CoordName	string[]	out	項目,ex:EXT,G54,G55...	WorkCoord	double[][]	out	工件座標資料																		
變數名稱	型別	[in/out]	說明																																								
StartNumber	short	in	起始編號(從 1 開始)																																								
EndNumber	short	in	終止編號																																								
CoordName	string[]	out	項目,ex:EXT,G54,G55...																																								
WorkCoord	double[][]	out	工件座標資料																																								
備註	<ul style="list-style-type: none"> ● FANUC 控制器及三菱控制器 CoordName 項目如： EXT, G54, G55, G56, G57, G58, G59 , G54.1P1, G54.1P2, G54.1P3 等 依指定編號對應至工件座標名稱如下列範圍： <table border="1" data-bbox="512 689 906 1133"> <thead> <tr> <th>工件名稱</th> <th>對應編號</th> </tr> </thead> <tbody> <tr><td>EXT</td><td>1</td></tr> <tr><td>G54</td><td>2</td></tr> <tr><td>G55</td><td>3</td></tr> <tr><td>G56</td><td>4</td></tr> <tr><td>G57</td><td>5</td></tr> <tr><td>G58</td><td>6</td></tr> <tr><td>G59</td><td>7</td></tr> <tr><td>G54.1 P1~Pn</td><td>8~n</td></tr> </tbody> </table> ● 新代控制器 CoordName 項目如： EXT,MPGShift, G54, G55, G56, G57, G58, G59 , G54P7, G54P8, G54P100 等 依指定編號對應至工件座標名稱如下列範圍： <table border="1" data-bbox="512 1373 906 1865"> <thead> <tr> <th>工件名稱</th> <th>對應編號</th> </tr> </thead> <tbody> <tr><td>EXT</td><td>0</td></tr> <tr><td>MPG Shift</td><td>1</td></tr> <tr><td>G54</td><td>2</td></tr> <tr><td>G55</td><td>3</td></tr> <tr><td>G56</td><td>4</td></tr> <tr><td>G57</td><td>5</td></tr> <tr><td>G58</td><td>6</td></tr> <tr><td>G59</td><td>7</td></tr> <tr><td>G54 P7~P100</td><td>8~101</td></tr> </tbody> </table> 					工件名稱	對應編號	EXT	1	G54	2	G55	3	G56	4	G57	5	G58	6	G59	7	G54.1 P1~Pn	8~n	工件名稱	對應編號	EXT	0	MPG Shift	1	G54	2	G55	3	G56	4	G57	5	G58	6	G59	7	G54 P7~P100	8~101
工件名稱	對應編號																																										
EXT	1																																										
G54	2																																										
G55	3																																										
G56	4																																										
G57	5																																										
G58	6																																										
G59	7																																										
G54.1 P1~Pn	8~n																																										
工件名稱	對應編號																																										
EXT	0																																										
MPG Shift	1																																										
G54	2																																										
G55	3																																										
G56	4																																										
G57	5																																										
G58	6																																										
G59	7																																										
G54 P7~P100	8~101																																										
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1" data-bbox="480 1912 1393 2054"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>X</td> <td>?</td> <td>0</td> </tr> <tr> <td>寶元</td> <td>工研院</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>					FANUC	三菱	海德漢	西門子	新代	0	0	X	?	0	寶元	工研院																										
FANUC	三菱	海德漢	西門子	新代																																							
0	0	X	?	0																																							
寶元	工研院																																										

	O	O			
C#程式範例	<pre> StructMsg.work_coord_scope _work_coord_scope; StructMsg.Pwd _Pwd; private void button7_Click(object sender, EventArgs e) { _work_coord_scope.StartNumber = 1; _work_coord_scope.EndNumber = 8; _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_work_coord_scope(_Pwd, ref _work_coord_scope); if (ret == 0) { //EXT Work Coord txtItem.Text = _work_coord_scope.CoordName[0]; txtX.Text = _work_coord_scope.WorkCoord[0][0].ToString(); txtY.Text = _work_coord_scope.WorkCoord[0][1].ToString(); txtZ.Text = _work_coord_scope.WorkCoord[0][2].ToString(); } } </pre>				

4. GET_work_coord_single：取得單筆工件座標資料

API 名稱	GET_work_coord_single				
函式說明	取得單筆工件座標資料				
C#方法	short GET_work_coord_single(StructMsg.Pwd P, ref StructMsg.work_coord_single R);				
VB.Net 方法	GET_work_coord_single(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.work_coord_single) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：work_coord_single				
	變數名稱	型別	[in/out]	說明	
	CoordName	string	in	項目,ex:EXT,G54,G55...	
	WorkCoord	double[]	out	工件座標資料	
備註	CoordName 項目如： EXT, G54, G55, G56, G57, G58, G59 , G54.1P1, G54.1P2, G54.1P3 等				
適用範圍	O：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	O	O	X	?	O
	寶元	工研院			

	O	O			
C#程式範例	<pre>StructMsg.work_coord_single _work_coord_single; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _work_coord_single.CoordName = "G55"; _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_work_coord_single(_Pwd, ref _work_coord_single); if (ret == 0) { //EXT Work Coord txtItem2.Text = _work_coord_single.CoordName; txtX2.Text = _work_coord_single.WorkCoord[0].ToString(); txtY2.Text = _work_coord_single.WorkCoord[1].ToString(); txtZ2.Text = _work_coord_single.WorkCoord[2].ToString(); } }</pre>				

5. SET_work_coord_all：寫入所有工件座標資料

API 名稱	SET_work_coord_all				
函式說明	寫入所有工件座標資料				
C#方法	short SET_work_coord_all(StructMsg.Pwd P, StructMsg.work_coord_all R);				
VB.Net 方法	SET_work_coord_all(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.work_coord_all) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：work_coord_all				
	變數名稱	型別	[in/out]	說明	
	CoordName	string[]	in	設定項目,ex:EXT,G54,G55...	
	WorkCoord	double[][]	in	設定工件座標資料	
備註	CoordName 設定項目如： EXT, G54, G55, G56, G57, G58, G59 , G54.1P1, G54.1P2, G54.1P3 等 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。				
適用範圍	O：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	O	O	X	?	O
	寶元	工研院			
	O	O			

C#程式範例	<pre> StructMsg.work_coord_all _work_coord_all; StructMsg.Pwd _Pwd; private void button5_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_work_coord_all(_Pwd, ref _work_coord_all); if (ret == 0) { //write G57 = [4] _work_coord_all.WorkCoord[4][0] = double.Parse(txtX3.Text); _work_coord_all.WorkCoord[4][1] = double.Parse(txtY3.Text); _work_coord_all.WorkCoord[4][2] = double.Parse(txtZ3.Text); } ret = iRemoting.SET_work_coord_all(_work_coord_all); if (ret == 0) MessageBox.Show("Write success!"); else MessageBox.Show("Write failed!"); } </pre>
--------	--

6. SET_work_coord_single：寫入單筆工件座標資料

API 名稱	SET_work_coord_single														
函式說明	寫入單筆工件座標資料														
C#方法	short SET_work_coord_single(StructMsg.Pwd P, StructMsg.work_coord_single R);														
VB.Net 方法	SET_work_coord_single(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.work_coord_single) As Short														
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：work_coord_single <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #e0e0e0;">變數名稱</th> <th style="background-color: #e0e0e0;">型別</th> <th style="background-color: #e0e0e0;">[in/out]</th> <th style="background-color: #e0e0e0;">說明</th> </tr> </thead> <tbody> <tr> <td>CoordName</td> <td>string</td> <td style="background-color: yellow;">in</td> <td>設定項目,ex:EXT,G54,G55...</td> </tr> <tr> <td>WorkCoord</td> <td>double[]</td> <td style="background-color: yellow;">in</td> <td>設定工件座標資料</td> </tr> </tbody> </table>			變數名稱	型別	[in/out]	說明	CoordName	string	in	設定項目,ex:EXT,G54,G55...	WorkCoord	double[]	in	設定工件座標資料
變數名稱	型別	[in/out]	說明												
CoordName	string	in	設定項目,ex:EXT,G54,G55...												
WorkCoord	double[]	in	設定工件座標資料												
備註	※ CoordName 設定項目如： EXT, G54, G55, G56, G57, G58, G59 , G54.1P1, G54.1P2, G54.1P3 等 ※ WorkCoord 的陣列長度，必須等於軸個數。 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。														

適用範圍	O : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	O	O	X	?	O
	寶元	工研院			
	O	O			
C#程式範例	<pre>StructMsg.work_coord_single _work_coord_single; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _work_coord_single.CoordName = "G57"; _work_coord_single.WorkCoord = new double[3]; _work_coord_single.WorkCoord[0] = double.Parse(txtX3.Text); _work_coord_single.WorkCoord[1] = double.Parse(txtY3.Text); _work_coord_single.WorkCoord[2] = double.Parse(txtZ3.Text); short ret = iRemoting.SET_work_coord_single(_Pwd, _work_coord_single); if (ret == 0) MessageBox.Show("Write success!"); else MessageBox.Show("Write failed!"); }</pre>				

7. GET_work_coord_count : 取得工件座標筆數

API 名稱	GET_work_coord_count			
函式說明	取得工件座標筆數			
C#方法	short GET_work_coord_count(StructMsg.Pwd P, ref StructMsg.total_count R);			
VB.Net 方法	GET_work_coord_count(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.total_count) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：total_count			
	變數名稱	型別	[in/out]	說明
Count	short	out	工件座標筆數	
備註				

適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.total_count _total_count; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_work_coord_count(_Pwd, ref _total_count); if (ret == 0) { txtCount.Text = _total_count.Count.ToString(); } }</pre>				

8. GET_preset_title：取得工作台管理表的標題列

API 名稱	GET_preset_title				
函式說明	取得工作台管理表的標題列				
C#方法	short GET_preset_title(StructMsg.Pwd P, ref StructMsg.preset_title R);				
VB.Net 方法	GET_preset_title(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.preset_title) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：preset_title				
	變數名稱	型別	[in/out]	說明	
	PresetName	string	in	工作台管理表名稱(PRESET.PR / PRESET2.PR / PRESET3.PR)	
	PresetTitle	string[]	out	標題列	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.preset_title _preset_title; StructMsg.Pwd _Pwd;</pre>				

```
private void button1_Click(object sender, EventArgs e)
{
    _Pwd.ConnectionKey = "123";
    _preset_title.PresetName = "PRESET.PR";
    short ret = iRemoting.GET_preset_title(_Pwd, ref _preset_title);
    if (ret == 0)
    {
        for (int i = 0; i <= _preset_title.PresetTitle.Length - 1; i++)
        {
            rtbTitle.Text += _preset_title.PresetTitle[i] + "\r\n";
        }
    }
}
```

9. GET_preset_count：取得工作台管理表筆數

API 名稱	GET_preset_count			
函式說明	取得工作台管理表筆數			
C#方法	short GET_preset_count(StructMsg.Pwd P, ref StructMsg.preset_count R);			
VB.Net 方法	GET_preset_count(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.preset_count) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：preset_count			
	變數名稱	型別	[in/out]	說明
	PresetName	string	in	工作台管理表名稱(PRESET.PR / PRESET2.PR / PRESET3.PR)
Count	short	out	工作台可設定總數	
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	X	X	0	?
	寶元	工研院		
X	X			
C#程式範例	StructMsg.preset_count _preset_count; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _preset_count.PresetName = "PRESET.PR"; }			

```

short ret = iRemoting.GET_preset_count(_Pwd, ref _preset_count);
if (ret == 0)
{
    MessageBox.Show(_preset_count.Count.ToString());
}
}
    
```

10. GET_preset_data：取得工作台管理表資料

API 名稱	GET_preset_data			
函式說明	取得工作台管理表資料			
C#方法	short GET_preset_data(StructMsg.Pwd P, ref StructMsg.preset_data R);			
VB.Net 方法	GET_preset_data(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.preset_data) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：preset_data			
	變數名稱	型別	[in/out]	說明
PresetName	string	in	工作台管理表名稱(PRESET.PR / PRESET2.PR / PRESET3.PR)	
Data	string[][]	out	工作台管理資料	
備註	※ Data 第二個陣列，請配合標題名稱。			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	X	X	0	?
	寶元	工研院		
	X	X		
C#程式範例	<pre> StructMsg.preset_data _preset_data; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _preset_data.PresetName = "PRESET.PR"; short ret = iRemoting.GET_preset_data(_Pwd, ref _preset_data); if (ret == 0) { DataTable dtTool = new DataTable(); for (int i = 0; i <= _preset_title.PresetTitle.Length - 1; i++) { dtTool.Columns.Add(_preset_title.PresetTitle[i], Type.GetType("System.String")); } } } </pre>			

```

    }

    for (int i = 0; i <= _preset_data.Data.Length - 1; i++)
    {
        DataRow dr = dtTool.NewRow();
        for (int j = 0; j <= _preset_title.PresetTitle.Length - 1; j++)
        {
            dr[j] = _preset_data.Data[i][j];
        }
        dtTool.Rows.Add(dr);
    }

    dgvTool.DataSource = dtTool;
}
}
}

```

11. SET_preset_data：寫入工作台管理表

API 名稱	SET_preset_data			
函式說明	寫入工作台管理表			
C#方法	short SET_preset_data(StructMsg.Pwd P, StructMsg.preset_data R);			
VB.Net 方法	SET_preset_data(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.preset_data) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：preset_data			
	變數名稱	型別	[in/out]	說明
	PresetName	string	in	工作台管理表名稱(PRESET.PR / PRESET2.PR / PRESET3.PR)
	Data	string[][]	in	設定工作台管理資料
備註	※ 可針對某幾筆資料或全部資料，進行寫入更新。 ※ 針對某幾筆資料更新時，第[][0]個陣列可指定編號，如範例。 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	X	X	0	?
	寶元	工研院		
	X	X		
C#程式範例	StructMsg.Pwd _Pwd; _preset_data.Data = new string[1][]; _preset_data.Data[0] = new string[_preset_title.PresetTitle.Length];			

	<pre> DataTable dtTool = (DataTable)dgvTool.DataSource; for (int i = 0; i <= _preset_title.PresetTitle.Length - 1; i++) { _preset_data.Data[0][i] = dtTool.Rows[0][i].ToString(); } _Pwd.ConnectionKey = "123"; short ret = iRemoting.SET_preset_data(_Pwd, _preset_data); </pre>
--	---

12. GET_preset_data_mem：取得工作台管理表，並將資料讀取到 SkyMars 記憶體中

API 名稱	GET_preset_data_mem				
函式說明	將資料讀取到 SkyMars 記憶體中，並回傳資料陣列最大索引值				
C#方法	short GET_preset_data_mem(StructMsg.Pwd P, ref StructMsg.preset_data_max_index R);				
VB.Net 方法					
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：preset_data_max_index				
	變數名稱	型別	[in/out]	說明	
	PresetName	string	in	PRESET.PR、 PRESET2.PR、 PRESET3.PR	
	MaxIndex	int	out	資料陣列最大索引值	
備註	如果您需要分段讀取工具台管理表，請先將執行 GET_preset_data_mem 函式，再使用 GET_preset_data_scope 函式，並指定索引值範圍將資料取回。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre> StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; _PresetMaxIndex.PresetName = "PRESET.PR"; ret = iRemoting.GET_preset_data_mem(_Pwd, ref _PresetMaxIndex); </pre>				

13. GET_preset_data_scope：取得工作台管理表，並依指定索引值範圍傳回

API 名稱	GET_preset_data_scope
函式說明	工作台管理表資料，並依指定索引值範圍傳回

C#方法	short GET_preset_data_scope(StructMsg.Pwd P, ref StructMsg.preset_data_scope R);				
VB.Net 方法					
資料結構	結構名稱：preset_data_scope				
	變數名稱	型別	[in/out]	說明	
	StartIndex	int	in	起始索引值	
	EndIndex	int	in	結束索引值	
	PresetName	string	in	PRESET.PR、 PRESET2.PR、 PRESET3.PR	
Data	string[][]	out	工作台管理資料		
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	X	X	0	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.preset_data_scope _DataScope; StructMsg.Pwd _Pwd; for (int i = 0; i <= _PresetMaxIndex.MaxIndex; i += 20) { _Pwd.ConnectionKey = "123"; //一次讀取20筆 _DataScope.StartIndex = i; if ((i + 20) <= _PresetMaxIndex.MaxIndex) _DataScope.EndIndex = i + 20 - 1; else _DataScope.EndIndex = _PresetMaxIndex.MaxIndex; _DataScope.PresetName = "PRESET.PR"; ret = iRemoting.GET_preset_data_scope(_Pwd, ref _DataScope); if (ret == 0) { for (int j = 0; j <= _DataScope.Data.Length - 1; j++) { DataRow dr = dtTool.NewRow(); for (int s = 0; s <= _preset_title.PresetTitle.Length - 1; s++) {</pre>				


```
        dr[s] = _DataScope.Data[j][s];
    }
    dtTool.Rows.Add(dr);
}
}
```

十二、 Macro 變數類別函式(共通變數)

1. GET_macro_all：取得所有 Macro(共通變數)變數資料

API 名稱	GET_macro_all				
函式說明	取得所有 Macro 變數資料				
C#方法	short GET_macro_all(StructMsg.Pwd P, ref StructMsg.macro_all R);				
VB.Net 方法	GET_macro_all(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.macro_all) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：macro_all				
	變數名稱	型別	[in/out]	說明	
	MacroNumber	int[]	out	Macro 編號	
MacroData	double[]	out	Macro 變數資料		
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.macro_all _macro_all; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_macro_all(_Pwd, ref _macro_all); if (ret == 0) { DataTable dtMacro = new DataTable(); dtMacro.Columns.Add("No", Type.GetType("System.Int16")); dtMacro.Columns.Add("Data", Type.GetType("System.Double")); DataRow dr; for (int i = 0; i <= _macro_all.MacroNumber.Length - 1; i++) { dr = dtMacro.NewRow(); dr[0] = _macro_all.MacroNumber[i]; dr[1] = _macro_all.MacroData[i]; dtMacro.Rows.Add(dr); } dataGridView1.DataSource = dtMacro; } }</pre>				

	<pre> } } </pre>
--	------------------

2. GET_macro_scope：依指定範圍取得 Macro(共通變數)變數資料

API 名稱	GET_macro_scope				
函式說明	依指定範圍取得 Macro 變數資料				
C#方法	short GET_macro_scope(StructMsg.Pwd P, ref StructMsg.macro_scope R);				
VB.Net 方法	GET_macro_scope(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.macro_scope) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：macro_scope				
	變數名稱	型別	[in/out]	說明	
	StartNumber	int	in	起始號碼(從 1 開始)	
	EndNumber	int	in	終止號碼	
	MacroNumber	int[]	out	Macro 編號	
	MacroData	double[]	out	Macro 變數資料	
備註					
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre> StructMsg.macro_scope _macro_scope; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _macro_scope.StartNumber = 513; _macro_scope.EndNumber = 515; short ret = iRemoting.GET_macro_scope(_Pwd, ref _macro_scope); if (ret == 0) { DataTable dtMacro = new DataTable(); dtMacro.Columns.Add("No", Type.GetType("System.Int16")); dtMacro.Columns.Add("Data", Type.GetType("System.Double")); DataRow dr; for (int i = 0; i <= _macro_scope.MacroNumber.Length - 1; i++) { </pre>				

```

        dr = dtMacro.NewRow();
        dr[0] = _macro_scope.MacroNumber[i];
        dr[1] = _macro_scope.MacroData[i];
        dtMacro.Rows.Add(dr);
    }
    dataGridView1.DataSource = dtMacro;
}
}

```

3. SET_macro_all：寫入所有 Macro(共通變數)變數資料

API 名稱	SET_macro_all				
函式說明	寫入所有 Macro 變數資料				
C#方法	short SET_macro_all(StructMsg.Pwd P, StructMsg.macro_all R);				
VB.Net 方法	SET_macro_all(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.macro_all) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：macro_all				
	變數名稱	型別	[in/out]	說明	
MacroNumber	int[]	in	設定 Macro 編號		
MacroData	double[]	in	設定 Macro 變數資料		
備註	※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre> StructMsg.macro_all _macro_all; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_macro_all(_Pwd, ref _macro_all); if (ret == 0) { //Search no for (int i = 0; i <= _macro_all.MacroNumber.Length - 1; i++) { if (_macro_all.MacroNumber[i] == short.Parse(txtNo.Text)) </pre>				

```

        {
            _macro_all.MacroData[i] =
double.Parse(txtData.Text);
            break;
        }
    }
}

ret = iRemoting.SET_macro_all(_macro_all);
if (ret == 0)
    MessageBox.Show("Write success!");
else
    MessageBox.Show("Write failed! ret:" + ret.ToString());
}
    
```

4. GET_macro_single：取得單筆 Macro(共通變數)變數資料

API 名稱	GET_macro_single			
函式說明	取得單筆 Macro 變數資料			
C#方法	short GET_macro_single(StructMsg.Pwd P, ref StructMsg.macro_single R);			
VB.Net 方法	GET_macro_single(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.macro_single) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：macro_single			
	變數名稱	型別	[in/out]	說明
	MacroNumber	int	in	Macro 編號
	MacroData	double	out	Macro 變數資料
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	X	?
	寶元	工研院		
	0	0		
C#程式範例	StructMsg.macro_single _macro_single; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _macro_single.MacroNumber = short.Parse(txtNo.Text); }			

	<pre> short ret = iRemoting.GET_macro_single(_Pwd, ref _macro_single); if (ret == 0) { txtData.Text = _macro_single.MacroData.ToString(); } } </pre>
--	--

5. SET_macro_single：寫入單筆 Macro(共通變數)變數資料

API 名稱	SET_macro_single																								
函式說明	寫入單筆 Macro 變數資料																								
C#方法	short SET_macro_single(StructMsg.Pwd P, StructMsg.macro_single R);																								
VB.Net 方法	SET_macro_single(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.macro_single) As Short																								
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：macro_single <table border="1" style="margin-top: 5px;"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>MacroNumber</td> <td>int</td> <td>in</td> <td>設定 Macro 編號</td> </tr> <tr> <td>MacroData</td> <td>double</td> <td>in</td> <td>設定 Macro 變數資料</td> </tr> </tbody> </table>					變數名稱	型別	[in/out]	說明	MacroNumber	int	in	設定 Macro 編號	MacroData	double	in	設定 Macro 變數資料								
變數名稱	型別	[in/out]	說明																						
MacroNumber	int	in	設定 Macro 編號																						
MacroData	double	in	設定 Macro 變數資料																						
備註	※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。																								
適用範圍	0：支援 X：未支援 ?：未實作 <table border="1" style="margin-top: 5px;"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>X</td> <td>?</td> <td>0</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>					FANUC	三菱	海德漢	西門子	新代	0	0	X	?	0	寶元	工研院				0	0			
FANUC	三菱	海德漢	西門子	新代																					
0	0	X	?	0																					
寶元	工研院																								
0	0																								
C#程式範例	<pre> StructMsg.macro_single _macro_single; StructMsg.Pwd _Pwd; private void button5_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _macro_single.MacroNumber = short.Parse(txtNo.Text); _macro_single.MacroData = double.Parse(txtData.Text); short ret = iRemoting.SET_macro_single(_Pwd, _macro_single); if (ret == 0) MessageBox.Show("Write success!"); else MessageBox.Show("Write failed! ret:" + ret.ToString()); } </pre>																								

6. GET_macro_variable：取得 Macro(共通變數)變數型態

API 名稱	GET_macro_count			
函式說明	取得 Macro 變數型態			
C#方法	short GET_macro_variable(StructMsg.Pwd P, ref StructMsg.macro_variable R);			
VB.Net 方法	GET_macro_variable(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.macro_variable) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：macro_variable			
	變數名稱	型別	[in/out]	說明
	Variable	int[][]	out	Macro 變數型態
備註	陣列宣告：Variable[type][0]:Start Number Variable[type][1]:End Number 如： Variable[0][0]:100 Variable[0][1]:199 Variable[1][0]:500 Variable[1][1]:599			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	X	?
	寶元	工研院		
	0	0		
C#程式範例	<pre>InterfaceLib.StructMsg.macro_variable _macro_variable; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_macro_variable(_Pwd, ref _macro_variable); if (ret == 0) { for (int i = 0; i <= _macro_variable.Variable.Length - 1; i++) { txtVariable.Text += "Type:" + (i + 1).ToString() + "\r\n"; txtVariable.Text += "Start Number:" + _macro_variable.Variable[i][0].ToString() + "\r\n"; txtVariable.Text += "End Number:" + _macro_variable.Variable[i][1].ToString() + "\r\n"; } } }</pre>			

	<pre> } } } </pre>
--	--------------------------------

7. GET_pcode_single：取得單筆 P Code Macro 資料(FANUC)

API 名稱	GET_pcode_single																							
函式說明	取得單筆 P Code Macro 資料(FANUC)																							
C#方法	short GET_pcode_single(StructMsg.Pwd P, ref StructMsg.macro_single R);																							
VB.Net 方法	GET_pcode_single(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.macro_single) As Short																							
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：macro_single <table border="1" data-bbox="475 779 1396 929"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>MacroNumber</td> <td>int</td> <td>in</td> <td>P code 編號</td> </tr> <tr> <td>MacroData</td> <td>double</td> <td>out</td> <td>P code 變數資料</td> </tr> </tbody> </table>				變數名稱	型別	[in/out]	說明	MacroNumber	int	in	P code 編號	MacroData	double	out	P code 變數資料								
變數名稱	型別	[in/out]	說明																					
MacroNumber	int	in	P code 編號																					
MacroData	double	out	P code 變數資料																					
備註	如果回傳值為 6 時： <ol style="list-style-type: none"> 請檢查此台控制器是否已選配 macro-executor 功能。 macro-executor 程式是否已經安裝至控制器內。 																							
適用範圍	0：支援 X：未支援 ?：未實作 <table border="1" data-bbox="475 1122 1396 1323"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>X</td> <td>X</td> <td>X</td> </tr> <tr> <td>寶元</td> <td>工研院</td> <td></td> <td></td> <td></td> </tr> <tr> <td>X</td> <td>X</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				FANUC	三菱	海德漢	西門子	新代	0	X	X	X	X	寶元	工研院				X	X			
FANUC	三菱	海德漢	西門子	新代																				
0	X	X	X	X																				
寶元	工研院																							
X	X																							
C#程式範例	<pre> StructMsg.macro_single _macro_single; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Single.MacroNumber = int.Parse(txtNum.Text); short ret = iRemoting.GET_pcode_single(_Pwd, ref _Single); if (ret == 0) txtValue.Text = _Single.MacroData.ToString(); else MessageBox.Show(ret.ToString()); } </pre>																							

8. SET_pcode_single：寫入單筆 P Code Macro 資料(FANUC)

API 名稱	SET_pcode_single
函式說明	寫入單筆 P Code Macro 資料(FANUC)

C#方法	short SET_pcode_single(StructMsg.Pwd P, StructMsg.macro_single R);			
VB.Net 方法	SET_pcode_single(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.macro_single) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：macro_single			
	變數名稱	型別	[in/out]	說明
	MacroNumber	int	in	設定 Macro 編號
MacroData	double	in	設定 Macro 變數資料	
備註	<p>※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。</p> <p>如果回傳值為 6 時：</p> <ol style="list-style-type: none"> 請檢查此台控制器是否已選配 macro-executor 功能。 macro-executor 程式是否已經安裝至控制器內。 			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	X	X	X
	寶元	工研院		
	X	X		
C#程式範例	<pre>StructMsg.macro_single _macro_single; StructMsg.Pwd _Pwd; private void button5_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _Single.MacroNumber = int.Parse(txtNum.Text); _Single.MacroData = double.Parse(txtValue.Text); short ret = iRemoting.SET_pcode_single(_Pwd, _Single); }</pre>			

十三、 加工程式類別函式

1. GET_nc_mem_list：取得機台內部加工程式清單(記憶體)

API 名稱	GET_nc_mem_list																							
函式說明	取得機台內部加工程式清單(記憶體)																							
C#方法	short GET_nc_mem_list(StructMsg.Pwd P, ref StructMsg.nc_list R);																							
VB.Net 方法	GET_nc_mem_list(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.nc_list) As Short																							
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。																							
	結構名稱：nc_list																							
	變數名稱	型別	[in/out]	說明																				
	FolderPath	string	in	資料夾路徑																				
	NcList	string[][]	out	加工程式清單																				
備註	<p>※FANUC 0i-D / 31i 控制器:nc_list.FolderPath 預設值為 //CNC_MEM/USER/PATH1/</p> <p>※NcList 第二個陣列： [0]:NC Name [1]:Size (byte) [2]:DateTime [3]:註解(Remark) [4]:file / directory</p> <p>※此函式 FANUC 目前不支援 DateTime ※三菱及新代目前不支援 DateTime、註解 ※HEIDENHAIN 控制器需設定 FolderPath 變數</p>																							
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>?</td> <td>0</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				FANUC	三菱	海德漢	西門子	新代	0	0	0	?	0	寶元	工研院				0	0			
FANUC	三菱	海德漢	西門子	新代																				
0	0	0	?	0																				
寶元	工研院																							
0	0																							
C#程式範例	<pre>StructMsg.nc_list _nc_list; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_nc_mem_list(_Pwd, ref _nc_list); if (ret == 0) { DataTable dtNcList = new DataTable();</pre>																							

```

        dtNcList.Columns.Add("Name",
Type.GetType("System.String"));
        dtNcList.Columns.Add("Size",
Type.GetType("System.Int32"));
        dtNcList.Columns.Add("Remark",
Type.GetType("System.String"));

        DataRow dr;
        for (int i = 0; i <= _nc_list.NcList.Length - 1; i++)
        {
            dr = dtNcList.NewRow();
            dr[0] = _nc_list.NcList[i][0];
            dr[1] = int.Parse(_nc_list.NcList[i][1]);
            dr[2] = _nc_list.NcList[i][3];
            dtNcList.Rows.Add(dr);
        }

        dataGridView1.DataSource = dtNcList;
    }
}

```

2. GET_nc_ftp_list：取得 FTP 卡加工程式清單(Fanuc DataServer/三菱 CF 卡)

API 名稱	GET_nc_ftp_list		
函式說明	取得 FTP 加工程式清單(Fanuc DataServer/三菱 CF 卡)		
C#方法	short GET_nc_ftp_list(StructMsg.Pwd P, ref StructMsg.nc_list R);		
VB.Net 方法	GET_nc_ftp_list(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.nc_list) As Short		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：nc_list		
	變數名稱	型別	[in/out] 說明
	FolderPath	string	in Not use
	NcList	string[][]	out 加工程式清單
備註	※ FANUC 控制器必須擁有 DataServer，並已經完成設定帳號及密碼。 ※ 三菱控制器無法使用在 Windows 作業系統。 NcList 第二個陣列包含有： [0]:加工程式名稱(Name) [1]:大小(Size) [2]:日期 [3]:屬性(directory:目錄,file:檔案)		

	※ 若回傳含有錯誤碼時，請參照 FTP 回傳錯誤列表。 ※ 若錯誤碼回傳 14 代表系統尚未設定 FTP，請檢查 SkyMars 機台設定。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre> StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_nc_ftp_list(_Pwd, ref _nc_list); if (ret == 0) { DataTable dtNcList = new DataTable(); dtNcList.Columns.Add("Name", Type.GetType("System.String")); dtNcList.Columns.Add("Size", Type.GetType("System.Int32")); DataRow dr; for (int i = 0; i <= _nc_list.NcList.Length - 1; i++) { if (_nc_list.NcList[i][3] == "f") { dr = dtNcList.NewRow(); dr[0] = _nc_list.NcList[i][0]; dr[1] = int.Parse(_nc_list.NcList[i][1]); dtNcList.Rows.Add(dr); } } dataGridView2.DataSource = dtNcList; } } </pre>				

3. GET_nc_mem_code：取得記憶體內的加工程式內容

API 名稱	GET_nc_mem_code
函式說明	取得記憶體內的加工程式內容

C#方法	short GET_nc_mem_code(StructMsg.Pwd P, ref StructMsg.nc_code R);			
VB.Net 方法	GET_nc_mem_code(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.nc_code) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：nc_code			
	變數名稱	型別	[in/out]	說明
	FolderPath	string	in	資料夾路徑
	NcName	string	in	名稱
	NcCode	string	out	內容
備註	FANUC 0i-D / 31i 控制器:FolderPath 預設值為//CNC_MEM/USER/PATH1/ HEIDENHAIN 控制器需設定 FolderPath 變數			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	0	?
	寶元	工研院		
	0	0		
C#程式範例	<pre>StructMsg.nc_code _nc_code; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _nc_code.NcName = dataGridView1.CurrentRow.Cells[0].Value.ToString(); short ret = iRemoting.GET_nc_mem_code(_Pwd, ref _nc_code); if (ret == 0) { rtbMemNcCode.Text = _nc_code.NcCode; } }</pre>			

4. GET_nc_ftp_code：取得 FTP 加工程式內容(Fanuc DataServer/三菱 CF 卡)

API 名稱	GET_nc_ftp_code
函式說明	取得 FTP 加工程式內容(Fanuc DataServer/三菱 CF 卡)
C#方法	short GET_nc_ftp_code(StructMsg.Pwd P, ref StructMsg.nc_code R);
VB.Net 方法	GET_nc_ftp_code(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.nc_code) As Short
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：nc_code

	變數名稱	型別	[in/out]	說明	
	FolderPath	string	in	Not use	
	NcName	string	in	名稱	
	NcCode	string	out	內容	
備註	※ FANUC 控制器必須擁有 FTP，並已經完成設定帳號及密碼。 ※ 三菱控制器無法使用在 Windows 作業系統。 ※ 若回傳含有錯誤碼時，請參照 FTP 回傳錯誤列表。 ※ 若錯誤碼回傳 14 代表系統尚未設定 FTP，請檢查 SkyMars 機台設定。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.nc_code _nc_code; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _nc_code.NcName = dataGridView2.CurrentRow.Cells[0].Value.ToString(); _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_nc_ftp_code(_Pwd, ref _nc_code); if (ret == 0) { rtbDsNcCode.Text = _nc_code.NcCode; } }</pre>				

5. UPLOAD_nc_mem：上傳加工程式至記憶體

API 名稱	UPLOAD_nc_mem			
函式說明	上傳加工程式至記憶體			
C#方法	short UPLOAD_nc_mem(StructMsg.Pwd P, StructMsg.nc_code R);			
VB.Net 方法	UPLOAD_nc_mem(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.nc_code) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：nc_code			
	變數名稱	型別	[in/out]	說明
	FolderPath	string	in	資料夾路徑
	NcName	string	in	名稱
	NcCode	string	in	內容

備註	FANUC 0i-D / 31i 控制器:FolderPath 預設值為//CNC_MEM/USER/PATH1/ HEIDENHAIN 控制器需設定 FolderPath 變數 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd),請參閱「四、API 連線及寫入密碼」。				
適用範圍	0 : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.nc_code _nc_code; StructMsg.Pwd _Pwd; private void button5_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _nc_code.NcName = "00002"; _nc_code.NcCode = System.IO.File.ReadAllText("D:\\00002"); short ret = iRemoting.UPLOAD_nc_mem(_Pwd, _nc_code); if (ret == 0) MessageBox.Show("Write success!"); else MessageBox.Show("Write failed! ret:" + ret.ToString()); }</pre>				

6. UPLOAD_nc_ftp : 上傳加工程式至 FTP(Fanuc DataServer/三菱 CF 卡)

API 名稱	UPLOAD_nc_ftp			
函式說明	上傳加工程式至 FTP(Fanuc DataServer/三菱 CF 卡)			
C#方法	short UPLOAD_nc_ftp(StructMsg.Pwd P, StructMsg.nc_code R);			
VB.Net 方法	UPLOAD_nc_ftp(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.nc_code) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：nc_code			
	變數名稱	型別	[in/out]	說明
	FolderPath	string	in	資料夾路徑
	NcName	string	in	名稱
	NcCode	string	in	內容
備註	FANUC 控制器必須擁有 DataServer，並已經完成設定帳號及密碼。 ※ 三菱控制器無法使用在 Windows 作業系統。 ※ 若回傳含有錯誤碼時，請參照 FTP 回傳錯誤列表。 ※ 若錯誤碼回傳 14 代表系統尚未設定 FTP，請檢查 SkyMars 機台設定。			

	※ 必須要設定 API 寫入密碼(Pwd.WritePwd),請參閱「四、API 連線及寫入密碼」。				
適用範圍	0 : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	X
	寶元	工研院			
	X	X			
C#程式範例	<pre>StructMsg.nc_code _nc_code; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; _nc_code.NcName = "00002"; _nc_code.NcCode = System.IO.File.ReadAllText("D:\\00002"); short ret = iRemoting.UPLOAD_nc_ftp(_Pwd, _nc_code); if (ret == 0) MessageBox.Show("Write success!"); else MessageBox.Show("Write failed! ret:" + ret.ToString()); }</pre>				

7. DEL_nc_mem : 刪除加工程式(記憶體)

API 名稱	DEL_nc_mem				
函式說明	刪除加工程式(記憶體)				
C#方法	short DEL_nc_mem(StructMsg.Pwd P, StructMsg.NcName R);				
VB.Net 方法	DEL_nc_mem(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.NcName) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：NcName				
	變數名稱	型別	[in/out]	說明	
	FolderPath	string	in	資料夾路徑	
Name	string	in	名稱		
備註	FANUC 0i-D / 31i 控制器:FolderPath 預設值為//CNC_MEM/USER/PATH1/ HEIDENHAIN 控制器需設定 FolderPath 變數 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd),請參閱「四、API 連線及寫入密碼」。				
適用範圍	0 : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	0
	寶元	工研院			
	0	0			


```

C#程式範例
StructMsg.NcName _NcName;
StructMsg.Pwd _Pwd;
private void button7_Click(object sender, EventArgs e)
{
    _Pwd.ConnectionKey = "123";
    _NcName.Name =
dataGridView1.CurrentRow.Cells[0].Value.ToString();
short ret = iRemoting.DEL_nc_mem(_Pwd, _NcName);
if (ret == 0)
    MessageBox.Show("Write success!");
else
    MessageBox.Show("Write failed! ret:" + ret.ToString());
}
    
```

8. DEL_nc_ftp：刪除加工程式(FTP) (Fanuc DataServer/三菱 CF 卡)

API 名稱	DEL_nc_ftp				
函式說明	刪除加工程式(ftp) (Fanuc DataServer/三菱 CF 卡)				
C#方法	short DEL_nc_ftp(StructMsg.Pwd P, StructMsg.NcName R);				
VB.Net 方法	DEL_nc_ftp(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.NcName) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：NcName				
	變數名稱	型別	[in/out]	說明	
	FolderPath	string	in	Not use	
	Name	string	in	名稱	
備註	FANUC 控制器必須擁有 DataServer，並已經完成設定帳號及密碼。 ※ 三菱控制器無法使用在 Windows 作業系統。 ※ 若回傳含有錯誤碼時，請參照 FTP 回傳錯誤列表。 ※ 若錯誤碼回傳 14 代表系統尚未設定 FTP，請檢查 SkyMars 機台設定。 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	X
	寶元	工研院			
	X	X			
C#程式範例	StructMsg.NcName _NcName; StructMsg.Pwd _Pwd; private void button8_Click(object sender, EventArgs e) {				

```

_Pwd.ConnectionKey = "123";
_NcName.Name =
dataGridView2.CurrentRow.Cells[0].Value.ToString();
short ret = iRemoting.DEL_nc_ftp(_Pwd, _NcName);
if (ret == 0)
    MessageBox.Show("Write success!");
else
    MessageBox.Show("Write failed! ret:" + ret.ToString());
}
    
```

9. GET_nc_pointer：取得程式目前的指標(行)

API 名稱	GET_nc_pointer		
函式說明	取得程式目前的指標(行)		
C#方法	short GET_nc_pointer(StructMsg.Pwd P, StructMsg.NcPointer R);		
VB.Net 方法	GET_nc_pointer(ByVal R As StructMsg.NcPointer) As Short		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：NcPointer		
	變數名稱	型別	[in/out] 說明
	BlkPointer	int	out 目前程式執行的指標
備註	<p>※三菱控制器不支援此函式。</p> <p>※如 FANUC NC 碼：</p> <pre> O0050(BT2); G01 Y0 X0 Z0 F1000.; G01 X10.2 F500.; X-6.325; M30; % 正在執行 G01Y0X0Z0F1000.; 其指標為 BlkPointer=2 </pre> <p>※如海德漢 NC 碼：</p> <pre> BEGIN PGM tt2 MM BLK FORM 0.1 Z X+0 Y+0 Z+0 BLK FORM 0.2 X+333 Y+333 Z+333 TOOL CALL 1 Z S2000 LBL 1 L X+0 Y+0 Z+100 R0 FMAX M3 L X+100 Y+100 Z+0 FMAX LBL 0 CALL LBL 1 REP9999 </pre>		

	END PGM tt2 MM 正在執行 L X+0 Y+0 Z+100 R0 FMAX M3 其指標為 BlkPointer=6				
適用範圍	0 : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	X	0	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre> StructMsg.NcPointer _NcPointer; StructMsg.Pwd _Pwd; private void button9_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_nc_pointer(_Pwd, ref _NcPointer); if (ret == 0) { txtPointer.Text = _NcPointer.BlkPointer.ToString(); } } </pre>				

10. GET_nc_current_block : 取得目前程式正在執行及未來要執行的單節

API 名稱	GET_nc_current_block				
函式說明	取得目前程式正在執行及未來要執行的單節				
C#方法	short GET_nc_current_block(StructMsg.Pwd P, ref StructMsg.nc_current_block R);				
VB.Net 方法					
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：nc_current_block				
	變數名稱	型別	[in/out]	說明	
	Block	string[]	out	單節陣列	
備註	※ 陣列中的第 0 個索引，即為目前 CNC 正在執行的單節。 ※ 只讀取 256 個字元長度。				
適用範圍	0 : 支援 X : 未支援 ? : 未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	0
	寶元	工研院			
	0	X			
C#程式範例	<pre> StructMsg.nc_current_block _current_block; StructMsg.Pwd _Pwd; </pre>				

	<pre>private void button10_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_nc_current_block(_Pwd, ref _current_block); if (ret==0) { for (int i = 0; i <= _current_block.Block.Length - 1; i++) { richTextBox1.Text += _current_block.Block[i] + ";\r\n"; } } }</pre>
--	--

11. GET_nc_freespace：取得剩餘空間

API 名稱	GET_nc_freespace				
函式說明	取得剩餘空間				
C#方法	short GET_nc_freespace(StructMsg.Pwd P, ref StructMsg.nc_freespace R);				
VB.Net 方法					
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：nc_freespace				
	變數名稱	型別	[in/out]	說明	
	FreeSpace	int	out	剩餘空間	
備註	※ FANUC 及三菱計算單位採用 CHAR ※ 海德漢、新代、寶元及工研院計算單位採用 byte				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	0	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.nc_freespace _nc_freespace; StructMsg.Pwd _Pwd; private void button11_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_nc_freespace(_Pwd, ref _nc_freespace); if (ret == 0) {</pre>				

	<pre> MessageBox.Show("Free: " + _nc_freespace.FreeSpace); } } </pre>
--	---

12. SET_nc_main：設定或指定加工程式為主程式

API 名稱	SET_nc_main				
函式說明	設定或指定加工程式為主程式				
C#方法	short SET_nc_main(StructMsg.Pwd P, StructMsg.NcName R);				
VB.Net 方法					
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：NcName				
	變數名稱	型別	[in/out]	說明	
	FolderPath	string	in	資料夾路徑	
	Name	string	in	名稱	
備註	FANUC 0i-D / 31i 控制器:FolderPath 預設值為//CNC_MEM/USER/PATH1/ ※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	?	?	?	0
	寶元	工研院			
	?	?			
C#程式範例	<pre> _NcName.FolderPath = "//CNC_MEM/USER/PATH1/"; _NcName.Name = textBox1.Text; short ret = iRemoting.SET_nc_main(_Pwd, _NcName); if (ret != 0) MessageBox.Show(ret.ToString()); </pre>				

13. UPLOAD_nc_mdi：上傳加工程式碼到控制器(MDI 模式)

API 名稱	UPLOAD_nc_mdi			
函式說明	上傳加工程式碼到控制器(MDI 模式)			
C#方法	short UPLOAD_nc_mdi(StructMsg.Pwd P, StructMsg.nc_code R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：nc_code			
	變數名稱	型別	[in/out]	說明
	FolderPath	string	X	保留，不須輸入
	NcName	string	X	保留，不須輸入

	NcCode	string	in	加工程式碼																				
備註	<p>※ 必須要設定 API 寫入密碼(Pwd.WritePwd),請參閱「四、API 連線及寫入密碼」。</p> <p>※ FANUC 控制器在不同系列上,有不同的字數限制: Series 160/180/210, 160i/180i/210i, 0i-A/B/C :最長 470 個字 Series 300i, 0i-D :最長 500 個字</p> <p>※ 控制器必須切換到 MDI 才能寫入加工程式碼。</p>																							
適用範圍	<p>0 : 支援 X : 未支援 ? : 未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>?</td> <td>?</td> <td>?</td> <td>0</td> </tr> <tr> <td>寶元</td> <td>工研院</td> <td></td> <td></td> <td></td> </tr> <tr> <td>?</td> <td>?</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				FANUC	三菱	海德漢	西門子	新代	0	?	?	?	0	寶元	工研院				?	?			
FANUC	三菱	海德漢	西門子	新代																				
0	?	?	?	0																				
寶元	工研院																							
?	?																							
C#程式範例	<pre>string NcCode = "G00X0.Y0.Z0.\n" + "F1500.\n" + "G01X100.Y100.Z20.\n" + "G04X2."; _nc_code.NcCode = NcCode; short ret = iRemoting.UPLOAD_nc_mdi(_Pwd, _nc_code); if (ret != 0) MessageBox.Show(ret.ToString());</pre>																							

十四、 系統參數類別函式

1. GET_param_max：取得參數最大號碼(或最後一個號碼)

API 名稱	GET_param_max			
函式說明	取得參數最大號碼(或最後一個號碼)			
C#方法	short GET_param_max(StructMsg.Pwd P, ref StructMsg.param_max R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：param_max			
	變數名稱	型別	[in/out]	說明
	MaxNumber	int	out	參數最大號碼
備註				
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	?	?
	寶元	工研院		
	X	0		
C#程式範例	<pre>StructMsg.param_max _param_max; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_param_max(_Pwd, ref _param_max); if (ret == 0) { textBox1.Text = _param_max.MaxNumber.ToString(); } }</pre>			

2. GET_param_data：取得參數資料

API 名稱	GET_param_data			
函式說明	取得參數資料			
C#方法	short GET_param_data(StructMsg.Pwd P, StructMsg.param_search_number SearchNumber, ref StructMsg.param_data[] R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：param_search_number			
	變數名稱	型別	[in/out]	說明
	ParamStart	int	in	參數起始號碼

	ParamEnd	int	in	參數終止號碼
	結構名稱：param_data			
	變數名稱	型別	[in/out]	說明
	ParamNumber	int	out	參數號碼
	ParamType	short	out	參數資料格式 0:binary 1:int 2:double 3:string
	ParamMulti	bool	out	判斷是否有多維或多軸
	ParamDataS	string[]	out	binary / string 參數資料
	ParamDataI	int[]	out	參數資料
	ParamDataD	double[]	out	參數資料
備註	※ 所指定的參數號碼在控制器上不存在或取得失敗時，ParamType 會被設為-1			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	0	?	?
	寶元	工研院		
	X	0		
C#程式範例	<pre>StructMsg.param_search_number _InputNumber; StructMsg.param_data[] _param_data; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; DataTable dt = new DataTable(); dt.Columns.Add("ParamNum", Type.GetType("System.Int32")); dt.Columns.Add("ParamData", Type.GetType("System.String")); _InputNumber.ParamStart = int.Parse(txtStart.Text); _InputNumber.ParamEnd = int.Parse(txtEnd.Text); short ret = iRemoting.GET_param_data(_Pwd, _InputNumber, ref _param_data); if (ret == 0) { DataRow dr; for (int i = 0; i <= _param_data.Length - 1; i++) { if (_param_data[i].ParamType != -1)</pre>			


```
{
    dr = dt.NewRow();
    dr[0] = _param_data[i].ParamNumber;
    if (_param_data[i].ParamMulti == false) //沒有多軸
    {
        switch (_param_data[i].ParamType)
        {
            case 0:
                dr[1] = _param_data[i].ParamDataS[0];
                break;
            case 1:
                dr[1] = _param_data[i].ParamDataI[0].ToString();
                break;
            case 2:
                dr[1] = _param_data[i].ParamDataD[0].ToString();
                break;
            case 3:
                dr[1] = _param_data[i].ParamDataS[0];
                break;
        }
    }
    else
    {
        //有多軸
        switch (_param_data[i].ParamType)
        {
            case 0:
                for (int j = 0; j <= _param_data[i].ParamDataS.Length - 1; j++)
                {
                    if (j == 0)
                        dr[1] = _param_data[i].ParamDataS[j];
                    else
                        dr[1] += ", " + _param_data[i].ParamDataS[j];
                }
                break;
            case 1:
                for (int j = 0; j <= _param_data[i].ParamDataI.Length - 1; j++)
                {
                    if (j == 0)
                        dr[1] = _param_data[i].ParamDataI[j];
```

```

        else
            dr[1] += ", " + _param_data[i].ParamDataI[j];
        }
        break;
        case 2:
            for (int j = 0; j <= _param_data[i].ParamDataD.Length - 1; j++)
            {
                if (j == 0)
                    dr[1] = _param_data[i].ParamDataD[j];
                else
                    dr[1] += ", " + _param_data[i].ParamDataD[j];
            }
            break;
        case 3:
            for (int j = 0; j <= _param_data[i].ParamDataS.Length - 1; j++)
            {
                if (j == 0)
                    dr[1] = _param_data[i].ParamDataS[j];
                else
                    dr[1] += ", " + _param_data[i].ParamDataS[j];
            }
            break;
        }
    }

    dt.Rows.Add(dr);
}

dataGridView1.DataSource = dt;
}
}

```

3. SET_param_data：設定參數資料

API 名稱	SET_param_data
函式說明	設定參數資料
C#方法	short SET_param_data(StructMsg.Pwd P, StructMsg.param_data[] R);
VB.Net 方法	

資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：param_data</p> <table border="1" data-bbox="480 248 1398 689"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>ParamNumber</td> <td>int</td> <td>in</td> <td>參數號碼</td> </tr> <tr> <td>ParamType</td> <td>short</td> <td>in</td> <td>參數資料格式 0:binary 1:int 2:double 3:string</td> </tr> <tr> <td>ParamMulti</td> <td>bool</td> <td>in</td> <td>判斷是否有多維或多軸</td> </tr> <tr> <td>ParamDataS</td> <td>string[]</td> <td>in</td> <td>binary / string 參數資料</td> </tr> <tr> <td>ParamDataI</td> <td>int[]</td> <td>in</td> <td>參數資料</td> </tr> <tr> <td>ParamDataD</td> <td>double[]</td> <td>in</td> <td>參數資料</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	ParamNumber	int	in	參數號碼	ParamType	short	in	參數資料格式 0:binary 1:int 2:double 3:string	ParamMulti	bool	in	判斷是否有多維或多軸	ParamDataS	string[]	in	binary / string 參數資料	ParamDataI	int[]	in	參數資料	ParamDataD	double[]	in	參數資料
變數名稱	型別	[in/out]	說明																										
ParamNumber	int	in	參數號碼																										
ParamType	short	in	參數資料格式 0:binary 1:int 2:double 3:string																										
ParamMulti	bool	in	判斷是否有多維或多軸																										
ParamDataS	string[]	in	binary / string 參數資料																										
ParamDataI	int[]	in	參數資料																										
ParamDataD	double[]	in	參數資料																										
備註	<p>※ 目前只適用在 FANUC 及新代</p> <p>※ 新代控制器：(1)只接受 int 型別的格式寫入。(2)不支援多維及多軸格式。</p>																												
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1" data-bbox="480 837 1398 1032"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>?</td> <td>?</td> <td>?</td> <td>0</td> </tr> <tr> <td>寶元</td> <td>工研院</td> <td></td> <td></td> <td></td> </tr> <tr> <td>X</td> <td>?</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FANUC	三菱	海德漢	西門子	新代	0	?	?	?	0	寶元	工研院				X	?											
FANUC	三菱	海德漢	西門子	新代																									
0	?	?	?	0																									
寶元	工研院																												
X	?																												
C#程式範例	<pre>StructMsg.param_data[] _param_data = new StructMsg.param_data[3]; _param_data[0].ParamNumber = 162; _param_data[0].ParamType = 0; _param_data[0].ParamMulti = false; _param_data[0].ParamDataS = new string[] { "00000011" }; _param_data[1].ParamNumber = 171; _param_data[1].ParamType = 1; _param_data[1].ParamMulti = false; _param_data[1].ParamDataI = new int[] { 8 }; _param_data[2].ParamNumber = 1006; _param_data[2].ParamType = 0; _param_data[2].ParamMulti = true; _param_data[2].ParamDataS = new string[3]; _param_data[2].ParamDataS[0] = "01100000"; _param_data[2].ParamDataS[1] = "00000000"; _param_data[2].ParamDataS[2] = "00000000"; short ret = iRemoting.SET_param_data(_Pwd, _param_data);</pre>																												

```
MessageBox.Show(ret.ToString());
```

十五、 PLC 類別函式

1. GET_plc_ver：取得 PLC 版本資訊

API 名稱	GET_plc_ver				
函式說明	取得 PLC 版本資訊				
C#方法	short GET_plc_ver(StructMsg.Pwd P, ref StructMsg.plc_ver R);				
VB.Net 方法	GET_plc_ver(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.plc_ver) As Short				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。				
	結構名稱：plc_ver				
	變數名稱	型別	[in/out]	說明	
	Version	string	out	版本資訊	
	Date	string	out	日期	
	Designed	string	out	設計者	
備註	※ FANUC 控制器： Version = PROGRAM DRAWING NO. Date = DATE OF PROGRAMING Designed = PROGRAM DESIGNED BY				
適用範圍	0：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	0	0	X	?	0
	寶元	工研院			
	0	0			
C#程式範例	<pre>StructMsg.plc_ver _plc_ver; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_plc_ver(_Pwd, ref _plc_ver); if (ret == 0) { textBox1.Text = _plc_ver.Version; } }</pre>				

2. GET_plc_addr：取得 PLC 位址的起始及結束號碼

API 名稱	GET_plc_addr
函式說明	取得 PLC 位址的起始及結束號碼
C#方法	short GET_plc_addr(StructMsg.Pwd P, ref StructMsg.plc_addr R);

VB.Net 方法																					
資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：plc_addr</p> <table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>Addr</td> <td>string</td> <td>in</td> <td>ex:A,D,T,C</td> </tr> <tr> <td>PlcStart</td> <td>int[]</td> <td>out</td> <td>位址起始值</td> </tr> <tr> <td>PlcEnd</td> <td>int[]</td> <td>out</td> <td>位址終止值</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	Addr	string	in	ex:A,D,T,C	PlcStart	int[]	out	位址起始值	PlcEnd	int[]	out	位址終止值				
變數名稱	型別	[in/out]	說明																		
Addr	string	in	ex:A,D,T,C																		
PlcStart	int[]	out	位址起始值																		
PlcEnd	int[]	out	位址終止值																		
備註	<p>※ FANUC 控制器能接受的位址(Addr)有： G, F, Y, X, A, R, T, K, C, D, E</p> <p>※ 新代控制器能接受的位址(Addr)有： I, O, C, S, A, R</p>																				
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>?</td> <td>?</td> <td>?</td> <td>0</td> </tr> <tr> <td>寶元</td> <td>工研院</td> <td></td> <td></td> <td></td> </tr> <tr> <td>?</td> <td>?</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FANUC	三菱	海德漢	西門子	新代	0	?	?	?	0	寶元	工研院				?	?			
FANUC	三菱	海德漢	西門子	新代																	
0	?	?	?	0																	
寶元	工研院																				
?	?																				
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.plc_addr _plc_addr; private void comboBox1_SelectedIndexChanged(object sender, EventArgs e) { txtPlcStart.Text = ""; txtPlcEnd.Text = ""; _Pwd.ConnectionKey = "123"; _plc_addr.Addr = comboBox1.Text; short ret = iRemoting.GET_plc_addr(_Pwd, ref _plc_addr); if (ret == 0) { for (int i = 0; i <= _plc_addr.PlcStart.Length - 1; i++) { txtPlcStart.Text += _plc_addr.PlcStart[i].ToString() + "\t"; txtPlcEnd.Text += _plc_addr.PlcEnd[i].ToString() + "\t"; } } else { MessageBox.Show(ret.ToString()); } }</pre>																				

	}
--	---

3. GET_plc_addr2：取得 PLC 位址的起始及結束號碼

API 名稱	GET_plc_addr2			
函式說明	取得 PLC 位址的起始及結束號碼			
C#方法	short GET_plc_addr2(StructMsg.Pwd P, ref StructMsg.plc_addr2 R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：plc_addr2			
	變數名稱	型別	[in/out]	說明
	Addr	string	in	Plc 位址代號 ex:A,D,T,C
	PlcStart	int[]	out	位址起始值
	PlcEnd	int[]	out	位址終止值
	PlcType	short[]	out	0:byte, 1:short, 2:int 3:bool, 4:double, 5:string 6:sbyte
備註	※ FANUC 控制器能接受的位址(Addr)有： G, F, Y, X, A, R, T, K, C, D, E ※ 新代控制器能接受的位址(Addr)有： I, O, C, S, A, R ※ 海德漢控制器(iTNC530)能接受的位址(Addr)有： M, B, W, D, I, O, T, C, S, IB, IW, ID, OB, OW, OD			
適用範圍	O：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	0	?	0	?
	寶元	工研院		
	?	?		
C#程式範例	<pre>StructMsg.plc_addr2 _plc_addr2 = new StructMsg.plc_addr2(); _Pwd.ConnectionKey = "123"; _plc_addr2.Addr = comHeidAddr.Text; short ret = iRemoting.GET_plc_addr2(_Pwd, ref _plc_addr2); if (ret == 0) { txtHeidStart.Text = _plc_addr2.PlcStart[0].ToString(); txtHeidEnd.Text = _plc_addr2.PlcEnd[0].ToString(); txtHeidType.Text = _plc_addr2.PlcType[0].ToString(); } }</pre>			

4. GET_plc_status：取得機台 PLC 狀態資訊

API 名稱	GET_plc_status																																			
函式說明	取得機台 PLC 狀態資訊																																			
C#方法	short GET_plc_status(StructMsg.Pwd P, ref StructMsg.plc_status R);																																			
VB.Net 方法	GET_plc_status(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.plc_status) As Short																																			
資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：plc_status</p> <table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>Addr</td> <td>string</td> <td>in</td> <td>ex:A,D,T,C</td> </tr> <tr> <td>PlcStart</td> <td>int</td> <td>in</td> <td>位址起始值</td> </tr> <tr> <td>PlcEnd</td> <td>int</td> <td>in</td> <td>位址終止值</td> </tr> <tr> <td>PlcType</td> <td>short</td> <td>out</td> <td>0:byte, 1:short, 2:int</td> </tr> <tr> <td>PlcDataB</td> <td>byte[]</td> <td>out</td> <td>Plc 資料(byte)</td> </tr> <tr> <td>PlcDataS</td> <td>short[]</td> <td>out</td> <td>Plc 資料(short)</td> </tr> <tr> <td>PlcDataI</td> <td>int[]</td> <td>out</td> <td>Plc 資料(int)</td> </tr> </tbody> </table>				變數名稱	型別	[in/out]	說明	Addr	string	in	ex:A,D,T,C	PlcStart	int	in	位址起始值	PlcEnd	int	in	位址終止值	PlcType	short	out	0:byte, 1:short, 2:int	PlcDataB	byte[]	out	Plc 資料(byte)	PlcDataS	short[]	out	Plc 資料(short)	PlcDataI	int[]	out	Plc 資料(int)
變數名稱	型別	[in/out]	說明																																	
Addr	string	in	ex:A,D,T,C																																	
PlcStart	int	in	位址起始值																																	
PlcEnd	int	in	位址終止值																																	
PlcType	short	out	0:byte, 1:short, 2:int																																	
PlcDataB	byte[]	out	Plc 資料(byte)																																	
PlcDataS	short[]	out	Plc 資料(short)																																	
PlcDataI	int[]	out	Plc 資料(int)																																	
備註	<p>※ FANUC 控制器能接受的位址(Addr)有： G, F, Y, X, A, R, T, K, C, D, E</p> <p>※ 三菱控制器(M70/M700)能接受的位址(Addr)有： X, Y, M, F, L, SM, D, R, W, B, SB, V, SW, SD</p> <p>※ 新代控制器能接受的位址(Addr)有： I, O, C, S, A, R</p> <p>※ 寶元控制器能接受的位址(Addr)有： A, R, C, I, S, O</p> <p>※ 工研院控制器能接受的位址(Addr)有： I, O, C, S, A, B, 強制 I 點, 強制 O 點, 強制 C 點</p> <p>※ 取得值會被放入 PlcData(B,S,I)的陣列中，如果該號碼取得失敗，此陣列值會被設為-1</p>																																			
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>?</td> <td>?</td> <td>0</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				FANUC	三菱	海德漢	西門子	新代	0	0	?	?	0	寶元	工研院				0	0															
FANUC	三菱	海德漢	西門子	新代																																
0	0	?	?	0																																
寶元	工研院																																			
0	0																																			
C#程式範例	<pre>StructMsg.plc_status _plc_status; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _plc_status.Addr = "T"; _plc_status.PlcStart = 2; }</pre>																																			

	<pre> _plc_status.PlcEnd = 2; _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_plc_status(_Pwd, ref _plc_status); if (ret == 0) { if (_plc_status.PlcType==0) txtPlcData.Text = _plc_status.PlcDataB[0].ToString(); else if (_plc_status.PlcType==1) txtPlcData.Text = _plc_status.PlcDataS[0].ToString(); else txtPlcData.Text = _plc_status.PlcDataI[0].ToString(); } } </pre>
--	---

5. GET_plc_status2：取得機台 PLC 狀態資訊

API 名稱	GET_plc_status2			
函式說明	取得機台 PLC 狀態資訊			
C#方法	short GET_plc_status2(StructMsg.Pwd P, ref StructMsg.plc_status2[] R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：plc_status2			
	變數名稱	型別	[in/out]	說明
	Addr	string	in	Plc 位址代號 ex:A,D,T,C
	PlcNumber	int	in	Plc 編號
	PlcType	short	out	0:byte, 1:short, 2:int 3:bool, 4:double, 5:string 6:sbyte
	PlcDataByte	byte	out	Plc 資料(byte)
	PlcDataShort	short	out	Plc 資料(short)
	PlcDataInt	int	out	Plc 資料(int)
	PlcDataBool	bool	out	Plc 資料(bool)
	PlcDataDouble	double	out	Plc 資料(double)
	PlcDataString	string	out	Plc 資料(string)
	PlcDataSByte	sbyte	out	Plc 資料(sbyte)
備註	※ FANUC 控制器能接受的位址(Addr)有： G, F, Y, X, A, R, T, K, C, D, E ※ 三菱控制器(M70/M700)能接受的位址(Addr)有： X, Y, M, F, L, SM, D, R, W, B, SB, V, SW, SD			

	<p>※海德漢控制器(iTNC530)能接受的位址(Addr)有： M, B, W, D, I, O, T, C, S, IB, IW, ID, OB, OW, OD</p> <p>※新代控制器能接受的位址(Addr)有： I, O, C, S, A, R</p> <p>※寶元控制器能接受的位址(Addr)有： A, R, C, I, S, O</p> <p>※工研院控制器能接受的位址(Addr)有： I, O, C, S, A, B, 強制 I 點, 強制 O 點, 強制 C 點</p>																				
適用範圍	<p>O : 支援 X : 未支援 ? : 未實作</p> <table border="1" data-bbox="480 584 1394 779"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>O</td> <td>O</td> <td>O</td> <td>?</td> <td>O</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>O</td> <td>O</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FANUC	三菱	海德漢	西門子	新代	O	O	O	?	O	寶元	工研院				O	O			
FANUC	三菱	海德漢	西門子	新代																	
O	O	O	?	O																	
寶元	工研院																				
O	O																				
C#程式範例	<pre> _Pwd.ConnectionKey = "123"; StructMsg.plc_status2[] _plc_status2 = new StructMsg.plc_status2[1]; _plc_status2[0].Addr = "W"; _plc_status2[0].PlcNumber = 9804; DataTable dt = new DataTable(); dt.Columns.Add("PlcNumber", Type.GetType("System.Int32")); dt.Columns.Add("PlcType", Type.GetType("System.Int16")); dt.Columns.Add("PlcValue", Type.GetType("System.String")); short ret = iRemoting.GET_plc_status2(_Pwd, ref _plc_status2); if (ret == 0) { for (int i = 0; i <= _plc_status2.Length - 1; i++) { DataRow dr = dt.NewRow(); dr["PlcNumber"] = _plc_status2[i].PlcNumber; dr["PlcType"] = _plc_status2[i].PlcType; switch (_plc_status2[i].PlcType) { case 0: dr["PlcValue"] = _plc_status2[i].PlcDataByte; break; case 1: </pre>																				

```

        dr["PlcValue"] = _plc_status2[i].PlcDataShort;
        break;
    case 2:
        dr["PlcValue"] = _plc_status2[i].PlcDataInt;
        break;
    case 3:
        dr["PlcValue"] = _plc_status2[i].PlcDataBool;
        break;
    case 4:
        dr["PlcValue"] = _plc_status2[i].PlcDataDouble;
        break;
    case 5:
        dr["PlcValue"] = _plc_status2[i].PlcDataString;
        break;
    }

    dt.Rows.Add(dr);
}

dgvHeid.DataSource = dt;
}
    
```

6. SET_plc_status：寫入機台 PLC

API 名稱	SET_plc_status		
函式說明	寫入機台 PLC		
C#方法	short SET_plc_status(StructMsg.Pwd P, StructMsg.plc_status R);		
VB.Net 方法	SET_plc_status(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.plc_status) As Short		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：plc_status		
	變數名稱	型別	[in/out] 說明
	Addr	string	in ex:A,D,T,C
	PlcStart	int	in 位址起始值
	PlcEnd	int	in 位址終止值
	PlcType	short	in 0:byte, 1:short, 2:int
	PlcDataB	byte[]	in Plc 資料(byte)
	PlcDataS	short[]	in Plc 資料(short)
	PlcDataI	int[]	in Plc 資料(int)
備註	※ FANUC 控制器能接受的位址(Addr)有：		

	<p>G, F, Y, X, A, R, T, K, C, D, E</p> <p>※ 三菱控制器(M70/M700)能接受的位址(Addr)有： X, Y, M, F, L, SM, D, R, W, B, SB, V, SW, SD</p> <p>※ 新代控制器能接受的位址(Addr)有： I, O, C, S, A, R</p> <p>※ 寶元控制器能接受的位址(Addr)有： A, R, C, I, S, O</p> <p>※ 工研院控制器能接受的位址(Addr)有： I, O, C, S, A, B, 強制 I 點, 強制 O 點, 強制 C 點</p> <p>※ 必須要設定 API 寫入密碼(Pwd.WritePwd),請參閱「四、API 連線及寫入密碼」。</p>																				
適用範圍	<p>O : 支援 X : 未支援 ? : 未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>O</td> <td>?</td> <td>?</td> <td>?</td> <td>O</td> </tr> <tr> <th>寶元</th> <th>工研院</th> <td></td> <td></td> <td></td> </tr> <tr> <td>O</td> <td>O</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	FANUC	三菱	海德漢	西門子	新代	O	?	?	?	O	寶元	工研院				O	O			
FANUC	三菱	海德漢	西門子	新代																	
O	?	?	?	O																	
寶元	工研院																				
O	O																				
C#程式範例	<pre>StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _plc_status.Addr = "D"; _plc_status.PlcStart = 15; _plc_status.PlcEnd = 20; _plc_status.PlcType = 0; //一定要設定成0 (FANUC) _plc_status.PlcDataB = new byte[5]; _plc_status.PlcDataB[0] = 1; _plc_status.PlcDataB[1] = 2; _plc_status.PlcDataB[2] = 3; _plc_status.PlcDataB[3] = 4; _plc_status.PlcDataB[4] = 5; _Pwd.ConnectionKey = "123"; short ret = iRemoting.SET_plc_status(_Pwd, _plc_status); if (ret == 0) MessageBox.Show("寫入成功!"); else MessageBox.Show(ret.ToString()); }</pre>																				

7. SET_plc_status2 : 寫入機台 PLC

API 名稱	SET_plc_status2
--------	-----------------

函式說明	寫入機台 PLC				
C#方法	short SET_plc_status2(StructMsg.Pwd P, StructMsg.plc_status2[] R);				
VB.Net 方法					
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：plc_status2				
	變數名稱	型別	[in/out]	說明	
	Addr	string	in	Plc 位址代號 ex:A,D,T,C	
	PlcNumber	int	in	Plc 編號	
	PlcType	short	in	0:byte, 1:short, 2:int 3:bool, 4:double, 5:string 6:sbyte	
	PlcDataByte	byte	in	Plc 資料(byte)	
	PlcDataShort	short	in	Plc 資料(short)	
	PlcDataInt	int	in	Plc 資料(int)	
	PlcDataBool	bool	in	Plc 資料(bool)	
	PlcDataDouble	double	in	Plc 資料(double)	
	PlcDataString	string	in	Plc 資料(string)	
	PlcDataSByte	sbyte	in	Plc 資料(sbyte)	
備註	※ FANUC 控制器能接受的位址(Addr)有： G, F, Y, X, A, R, T, K, C, D, E ※ 三菱控制器(M70/M700)能接受的位址(Addr)有： X, Y, M, F, L, SM, D, R, W, B, SB, V, SW, SD ※ 海德漢控制器(iTNC530)能接受的位址(Addr)有：(IB 可能無法寫入) M, B, W, D, I, O, T, C, S, IB, IW, ID, OB, OW, OD ※ 新代控制器能接受的位址(Addr)有： I, O, C, S, A, R ※ 寶元控制器能接受的位址(Addr)有： A, R, C, I, S, O ※ 工研院控制器能接受的位址(Addr)有： I, O, C, S, A, B, 強制 I 點, 強制 O 點, 強制 C 點 ※ 必須要設定 API 寫入密碼(Pwd.WritePwd),請參閱「四、API 連線及寫入密碼」。				
適用範圍	O：支援 X：未支援 ?：未實作				
	FANUC	三菱	海德漢	西門子	新代
	O	?	O	?	O
	寶元	工研院			
	O	O			
C#程式範例	<pre>_Pwd.ConnectionKey = "123"; StructMsg.plc_status2[] _plc_status2 = new</pre>				

```
StructMsg.plc_status2[1];
_plc_status2[0].Addr = "W";
_plc_status2[0].PlcNumber = 9804;
_plc_status2[0].PlcType = 1;
_plc_status2[0].PlcDataShort = 20;

short ret = iRemoting.SET_plc_status2(_Pwd, _plc_status2);
MessageBox.Show(ret.ToString());
```

十六、 檔案管理類別函式

1. GET_file_directory：取得控制上的檔案及目錄

API 名稱	GET_file_directory																							
函式說明	取得控制上的檔案及目錄																							
C#方法	short GET_file_directory(StructMsg.Pwd P, ref StructMsg.file_list R);																							
VB.Net 方法																								
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：file_list																							
	變數名稱	型別	[in/out]	說明																				
	FolderPath	string	in	資料夾路徑																				
	DirList	string[][]	out	目錄資料																				
備註	<p>※ 目前僅適用在海德漢控制器</p> <p>※file_list 第二個陣列：</p> <p>[0]:檔案或目錄名稱</p> <p>[1]:大小 (byte)</p> <p>[2]:日期</p> <p>[3]:權限，會有 3 個字元，分別是 drw：</p> <p>第 1 個字元代表這個項目是「資料夾(目錄)或檔案」：</p> <p>當為[d]則是資料夾(目錄)；當為[-]則是檔案。</p> <p>第 2 個字元代表這個檔案是否「可讀」</p> <p>當為[r]則是可讀；當為[-]則是不可讀</p> <p>第 3 個字元代表這個檔案是否「可寫」</p> <p>當為[w]則是可寫；當為[-]則是不可寫</p>																							
適用範圍	<p>0：支援 X：未支援 ?：未實作</p> <table border="1"> <thead> <tr> <th>FANUC</th> <th>三菱</th> <th>海德漢</th> <th>西門子</th> <th>新代</th> </tr> </thead> <tbody> <tr> <td>?</td> <td>?</td> <td>0</td> <td>?</td> <td>?</td> </tr> <tr> <td>寶元</td> <td>工研院</td> <td></td> <td></td> <td></td> </tr> <tr> <td>?</td> <td>?</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>				FANUC	三菱	海德漢	西門子	新代	?	?	0	?	?	寶元	工研院				?	?			
FANUC	三菱	海德漢	西門子	新代																				
?	?	0	?	?																				
寶元	工研院																							
?	?																							
C#程式範例	<pre>StructMsg.file_list _file_list = new StructMsg.file_list(); lvList.Items.Clear(); _file_list.FolderPath = txtPath.Text; short ret = iRemoting.GET_file_directory(_Pwd, ref _file_list); if (ret == 0) { for (int i = 0; i <= _file_list.DirList.Length - 1; i++) { ListViewItem item = new ListViewItem(_file_list.DirList[i][0]);</pre>																							

	<pre> item.SubItems.Add(_file_list.DirList[i][1]); item.SubItems.Add(_file_list.DirList[i][2]); item.SubItems.Add(_file_list.DirList[i][3]); lvList.Items.Add(item); } } </pre>
--	--

2. UPLOAD_file2：從 PC 端上傳檔案到裝置(如 CNC 控制器)

API 名稱	UPLOAD_file2			
函式說明	從 PC 端上傳檔案到裝置(如 CNC 控制器)			
C#方法	short UPLOAD_file2(StructMsg.Pwd P, StructMsg.file_path2 R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：file_path2			
	變數名稱	型別	[in/out]	說明
	FileContent	byte[]	in	從 PC 端上傳的檔案內容
	DestFile	string	in	該裝置上的檔案路徑及名稱
備註	※ 目前僅適用在海德漢控制器			
適用範圍	0：支援 X：未支援 ?：未實作			
	FANUC	三菱	海德漢	西門子
	?	?	0	?
	寶元	工研院		
	?	?		
C#程式範例	<pre> StructMsg.file_path _file_path = new StructMsg.file_path(); _file_path.SourceFile = openFileDialog1.FileName; _file_path.DestFile = txtPath.Text + "\\\" + openFileDialog1.SafeFileName; short ret = iRemoting.UPLOAD_file(_Pwd, _file_path); </pre>			

3. DOWNLOAD_file2：從裝置上(如 CNC 控制器)下載檔案到 PC 端

API 名稱	DOWNLOAD_file2			
函式說明	從裝置上(如 CNC 控制器)下載檔案到 PC 端			
C#方法	short DOWNLOAD_file2(StructMsg.Pwd P, StructMsg.file_path2 R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：file_path2			
	變數名稱	型別	[in/out]	說明

	SourceFile	string	in	該裝置上的檔案路徑及名稱
	FileContent	byte[]	out	下載到 PC 端的檔案內容
備註	※ 目前僅適用在海德漢控制器			
適用範圍	0 : 支援 X : 未支援 ? : 未實作			
	FANUC	三菱	海德漢	西門子
	?	?	0	?
	寶元	工研院		
	?	?		
C#程式範例	<pre>StructMsg.file_path _file_path = new StructMsg.file_path(); _file_path.SourceFile = txtPath.Text + "\\\" + lvList.SelectedItems[0].Text; _file_path.DestFile = saveFileDialog1.FileName; short ret = iRemoting.DOWNLOAD_file(_Pwd, _file_path);</pre>			

4. DEL_file : 從裝置上刪除檔案

API 名稱	DEL_file			
函式說明	從裝置上刪除檔案			
C#方法	short DEL_file(StructMsg.Pwd P, StructMsg.file_path R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：file_path			
	變數名稱	型別	[in/out]	說明
	SourceFile	string	in	該裝置上的檔案路徑及名稱
備註	※ 目前僅適用在海德漢控制器			
適用範圍	0 : 支援 X : 未支援 ? : 未實作			
	FANUC	三菱	海德漢	西門子
	?	?	0	?
	寶元	工研院		
	?	?		
C#程式範例	<pre>StructMsg.file_path _file_path = new StructMsg.file_path(); _file_path.SourceFile = txtPath.Text + "\\\" + lvList.SelectedItems[0].Text; short ret = iRemoting.DEL_file(_Pwd, _file_path);</pre>			

十七、 National Instruments 9233 加速規模組類別 (暫不開放)

1. START_ni9233_get_data：設定啟動模組進行資料擷取

API 名稱	START_ni9233_get_data			
函式說明	設定啟動模組進行資料擷取			
C#方法	short START_ni9233_get_data(StructMsg.Pwd P, StructMsg.NI9233_Param R);			
VB.Net 方法	START_ni9233_get_data(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.NI9233_Param) As Short			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：NI9233_Param			
	變數名稱	型別	[in/out]	說明
	PhysicalChannel	string	in	實體 Channel 名稱，如第一個模組第一個 Channel 為 Dev1/ai0，同一個模組有兩個 Channel 為 Dev1/ai0,Dev1/ai1
	MinimumValue	double	in	NI9233 模組所提供的最小電壓值
	MaximumValue	double	in	NI9233 模組所提供的最大電壓值
	Rate	double	in	1 秒內有擷取多少 Sample Rate
	Gain	double[]	in	設定 Gain 值
Offset	double[]	in	設定 Offset 值	
備註	<p>※該函式執行後，將會持續擷取，直到 STOP_ni9233_get_data 函式執行後，才會停止。</p> <p>※錯誤傳回-22，代表感測器已在執行，可使用 GET_ni9233_spectrum_data 函式將資料取回。</p> <p>※錯誤傳回-23，代表感測器內部或 API 發生錯誤，請使用 GET_ni9233_error_msg 函式取得錯誤訊息。</p>			
適用範圍	只適用在 National Instruments 9233 加速規模組。			
C#程式範例	<pre>StructMsg.NI9233_Param _NI9233_Param; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; if (iRemoting.CHECK_ni9233_running(_Pwd) == false) { //參數設定 _NI9233_Param.PhysicalChannel =</pre>			

	<pre> "Dev1/ai0,Dev1/ai1,Dev1/ai2"; _NI9233_Param.MinimumValue = -5; _NI9233_Param.MaximumValue = 5; _NI9233_Param.Rate = 100; _NI9233_Param.SamplesPerChannel = 1000; _NI9233_Param.Gain = new double[] { 10.208, 9.94, 10.352 }; _NI9233_Param.Offset = new double[] { 0, 0, 0 }; iRemoting.START_ni9233_get_data(_Pwd, _NI9233_Param); } else { //正在執行 MessageBox.Show("目前正在執行"); } } </pre>
--	---

2. STOP_ni9233_get_data：停止擷取

API 名稱	STOP_ni9233_get_data
函式說明	停止資料擷取
C#方法	void STOP_ni9233_get_data(StructMsg.Pwd P);
VB.Net 方法	STOP_ni9233_get_data()
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。
備註	
適用範圍	只適用在 National Instruments 9233 加速規模組。
C#程式範例	<pre> StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; iRemoting.STOP_ni9233_get_data(); </pre>

3. CHECK_ni9233_running：確認是否已啟動資料擷取

API 名稱	STOP_ni9233_get_data
函式說明	確認是否已啟動資料擷取
C#方法	bool CHECK_ni9233_running(StructMsg.Pwd P);
VB.Net 方法	CHECK_ni9233_running() As Boolean
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。
備註	
適用範圍	只適用在 National Instruments 9233 加速規模組。

C#程式範例	<pre>StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; if (iRemoting.CHECK_ni9233_running(_Pwd)) MessageBox.Show("目前正在執行中!"); else MessageBox.Show("尚未執行!"); }</pre>
--------	---

4. GET_ni9233_spectrum_data：取得資料

API 名稱	GET_ni9233_spectrum_data		
函式說明	取得資料		
C#方法	bool GET_ni9233_spectrum_data(StructMsg.Pwd P, ref StructMsg.NI9233_SpectrumData R);		
VB.Net 方法	GET_ni9233_spectrum_data(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.NI9233_SpectrumData) As Boolean		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：NI9233_Param		
	變數名稱	型別	[in/out] 說明
	Time	DateTime[]	out 每一個取樣點的時間
	Samples	Double[,]	out 二維陣列取樣值
備註	Samples 二維陣列：[第 n 個 Sample, 第 n 個 Channel 的 Sample 值]		
適用範圍	只適用在 National Instruments 9233 加速規模組。		
C#程式範例	<pre>StructMsg.NI9233_SpectrumData SpectrumData; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; DataTable dt = new DataTable(); dt.Columns.Add("Time", Type.GetType("System.String")); dt.Columns.Add("Ch1", Type.GetType("System.Double")); dt.Columns.Add("Ch2", Type.GetType("System.Double")); dt.Columns.Add("Ch3", Type.GetType("System.Double")); dt.Columns.Add("Ch4", Type.GetType("System.Double")); bool IsSuccess = iRemoting.GET_ni9233_spectrum_data(_Pwd, ref SpectrumData);</pre>		

```

        if (IsSuccess == true)
        {
            DataRow dr;
            for (int i = 0; i <= SpectrumData.Time.Length - 1; i++)
            {
                dr = dt.NewRow();
                dr[0] = SpectrumData.Time[i].Millisecond.ToString();
                dr[1] = SpectrumData.Samples[i, 0];
                dr[2] = SpectrumData.Samples[i, 1];
                dr[3] = SpectrumData.Samples[i, 2];
                dt.Rows.Add(dr);
            }
        }

        dataGridView1.DataSource = dt;
    }
    
```

5. GET_ni9233_error_msg：取得模組錯誤訊息

API 名稱	GET_ni9233_error_msg		
函式說明	取得模組錯誤訊息		
C#方法	StructMsg.SkyException GET_ni9233_error_msg(StructMsg.Pwd P);		
VB.Net 方法	GET_ni9233_error_msg() As StructMsg.SkyException		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：SkyException		
	變數名稱	型別	[in/out] 說明
	IsException	bool	out 判斷是否發生例外錯誤
	ExceptionMsg	string	out 錯誤訊息
	ErrorCode	int	out 錯誤碼
備註			
適用範圍	只適用在 National Instruments 9233 加速規模組。		
C#程式範例	StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; StructMsg.SkyException R = iRemoting.GET_ni9233_error(_Pwd); MessageBox.Show(R.ExceptionMsg);		

十八、 RS232 模組類別

1. GET_rs232_status：取得目前 RS232 的相關狀態及參數

API 名稱	GET_rs232_status		
函式說明	取得目前 RS232 的相關狀態及參數		
C#方法	void GET_rs232_status(StructMsg.Pwd P, ref StructMsg.RS232_Param[] R);		
VB.Net 方法	GET_rs232_status(ByRef R() As StructMsg.RS232_Param)		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：RS232_Param[]		
	變數名稱	型別	[in/out] 說明
	IsOpen	bool	out 是否已經開啟 Com Port
	TaskName	string	out 任務名稱
	RS232_PortName	string	out Com Port 名稱,如:COM1
	RS232_Parity	Parity	out 同位檢查通訊協定
	RS232_DataBit	int	out 每一位元組之資料位元的標準長度
	RS232_StopBit	StopBits	out 每位元組之停止位元的標準數目
	RS232_BaudRate	int	out 設定序列傳輸速率
	RS232_Handshake	Handshake	out 資料序列埠傳輸的交握通訊協定
	SerialType	short	out 序列埠傳入及傳回的型別, 0:string , 1:byte[]
	IsOnCallbackEvent	bool	out 是否要掛載序列埠事件傳回, 如果為 true,必須要設定結束字元
EndText	string	out 字串的結束字元	
EndByte	byte	out 位元組的結束字元	
備註	※SkyMars 允許建立 10 個陣列(10 組不同序列埠 Com Port)，回傳固定長度為 10 的結構陣列。 ※可從各陣列的 IsOpen 判斷此序列埠是否正在使用及 TaskName 的變數。		
適用範圍	RS-232 模組或擷取卡。		
C#程式範例	<pre>StructMsg.RS232_Param[] _Param; StructMsg.Pwd _Pwd; private void button6_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; iRemoting.GET_rs232_status(_Pwd, ref _Param); }</pre>		

	<pre> DataTable dt = new DataTable(); dt.Columns.Add("IsOpen", Type.GetType("System.Boolean")); dt.Columns.Add("TaskName", Type.GetType("System.String")); for (int i = 0; i <= _Param.Length - 1; i++) { DataRow dr = dt.NewRow(); dr[0] = _Param[i].IsOpen; dr[1] = _Param[i].TaskName; dt.Rows.Add(dr); } dataGridView1.DataSource = dt; } </pre>
--	---

2. OPEN_rs232：開啟 RS232

API 名稱	OPEN_rs232																																										
函式說明	開啟 RS232																																										
C#方法	short OPEN_rs232(StructMsg.Pwd P, int index, StructMsg.RS232_Param R);																																										
VB.Net 方法	OPEN_rs232(ByVal index As Integer, ByVal R As StructMsg.RS232) As Short																																										
資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：RS232_Param</p> <table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>TaskName</td> <td>string</td> <td>in</td> <td>設定任務名稱</td> </tr> <tr> <td>RS232_PortName</td> <td>string</td> <td>in</td> <td>設定 Com Port 名稱,如:COM1</td> </tr> <tr> <td>RS232_Parity</td> <td>Parity</td> <td>in</td> <td>設定同位檢查通訊協定</td> </tr> <tr> <td>RS232_DataBit</td> <td>int</td> <td>in</td> <td>設定每一位元組之資料位元的標準長度</td> </tr> <tr> <td>RS232_StopBit</td> <td>StopBits</td> <td>in</td> <td>設定每位元組之停止位元的標準數目</td> </tr> <tr> <td>RS232_BaudRate</td> <td>int</td> <td>in</td> <td>設定序列傳輸速率</td> </tr> <tr> <td>RS232_Handshake</td> <td>Handshake</td> <td>in</td> <td>設定資料序列埠傳輸的交握通訊協定</td> </tr> <tr> <td>SerialType</td> <td>short</td> <td>in</td> <td>設定序列埠傳入及傳回的型別,0:string , 1:byte[]</td> </tr> <tr> <td>IsOnCallbackEvent</td> <td>bool</td> <td>in</td> <td>是否要掛載序列埠事件傳回,</td> </tr> </tbody> </table>			變數名稱	型別	[in/out]	說明	TaskName	string	in	設定任務名稱	RS232_PortName	string	in	設定 Com Port 名稱,如:COM1	RS232_Parity	Parity	in	設定同位檢查通訊協定	RS232_DataBit	int	in	設定每一位元組之資料位元的標準長度	RS232_StopBit	StopBits	in	設定每位元組之停止位元的標準數目	RS232_BaudRate	int	in	設定序列傳輸速率	RS232_Handshake	Handshake	in	設定資料序列埠傳輸的交握通訊協定	SerialType	short	in	設定序列埠傳入及傳回的型別,0:string , 1:byte[]	IsOnCallbackEvent	bool	in	是否要掛載序列埠事件傳回,
變數名稱	型別	[in/out]	說明																																								
TaskName	string	in	設定任務名稱																																								
RS232_PortName	string	in	設定 Com Port 名稱,如:COM1																																								
RS232_Parity	Parity	in	設定同位檢查通訊協定																																								
RS232_DataBit	int	in	設定每一位元組之資料位元的標準長度																																								
RS232_StopBit	StopBits	in	設定每位元組之停止位元的標準數目																																								
RS232_BaudRate	int	in	設定序列傳輸速率																																								
RS232_Handshake	Handshake	in	設定資料序列埠傳輸的交握通訊協定																																								
SerialType	short	in	設定序列埠傳入及傳回的型別,0:string , 1:byte[]																																								
IsOnCallbackEvent	bool	in	是否要掛載序列埠事件傳回,																																								

				如果為 true,必須要設定結束字元
	EndText	string	in	設定字串的結束字元
	EndByte	byte	in	設定位元組的結束字元
備註	<p>※index:指向 SkyMars 第幾個陣列(第幾組)的序列埠。</p> <p>※SerialType 設為 0 時,即採用 string 方式輸入序列埠,採用 string 接收。</p> <p>※IsOnCallbackEvent 設為 true 時,必須要再設定 EndText 或 EndByte 變數。</p> <p>※IsOnCallbackEvent 設為 false,可忽略 EndText 及 EndByte 變數。</p> <p>※假若所指定 index 的序列埠已經開啟,將無法設定,並送出錯誤碼-25,須先關閉序列埠始可設定。</p>			
適用範圍	RS-232 模組或擷取卡。			
C#程式範例	<pre>StructMsg.RS232_Param _RS232_Param; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _RS232_Param.RS232_BaudRate = 2400; _RS232_Param.RS232_DataBit = 8; _RS232_Param.RS232_Handshake = Handshake.None; _RS232_Param.RS232_Parity = Parity.None; _RS232_Param.RS232_PortName = "COM6"; _RS232_Param.RS232_StopBit = StopBits.One; _RS232_Param.EndByte = 0x7c; _RS232_Param.IsOnCallbackEvent = true; _RS232_Param.SerialType = 1; _RS232_Param.TaskName = "溫度卡"; _Pwd.ConnectionKey = "123"; short ret = iRemoting.OPEN_rs232(_Pwd, 0, _RS232_Param); MessageBox.Show(ret.ToString()); }</pre>			

3. SET_rs232：將指定的資料寫入 RS232

API 名稱	SET_rs232
函式說明	將指定的資料寫入 RS232
C#方法	short SET_rs232(StructMsg.Pwd P, int index, StructMsg.RS232_Data R);
VB.Net 方法	SET_rs232(ByVal P As StructMsg.Pwd, ByVal index As Integer,ByVal R As StructMsg.RS232_Data) As Short
資料結構	結構名稱：Pwd, 連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：RS232_Data

	變數名稱	型別	[in/out]	說明
	Text	string	in	設定字串
	ArrayByte	byte[]	in	設定 byte 陣列
備註	<p>※ index: 指向 SkyMars 第幾個陣列(第幾組)的序列埠。</p> <p>※ 選擇 Text 或 ArrayByte 的變數，請依照之前使用的 OPEN_rs232 函式內的 SerialType 來選擇，如 SerialType=0，請設定 Text 變數，ArrayByte 可忽略。如果是 SerialType=1，請設定 ArrayByte 變數，Text 可忽略。</p> <p>※ 必須要設定 API 寫入密碼(Pwd.WritePwd)，請參閱「四、API 連線及寫入密碼」。</p>			
適用範圍	RS-232 模組或擷取卡。			
C#程式範例	<pre>StructMsg.RS232_Data _RS232_Data; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { byte[] wrCmd = { 0x05, 0x05, 0x05, 0x37 }; _RS232_Data.ArrayByte = wrCmd; _Pwd.ConnectionKey = "123"; short ret = iRemoting.SET_rs232(_Pwd, 0, _RS232_Data); MessageBox.Show(ret.ToString()); }</pre>			

4. CHECK_rs232_data_received：確認是否已經接收到資料(IsOnCallbackEvent=true)

API 名稱	CHECK_rs232_data_received
函式說明	確認是否已經接收到資料(IsOnCallbackEvent=true)
C#方法	bool CHECK_rs232_data_received(StructMsg.Pwd P, int index);
VB.Net 方法	CHECK_rs232_data_received(index As Integer) As Boolean
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。
備註	※ index: 指向 SkyMars 第幾個陣列(第幾組)的序列埠。
適用範圍	RS-232 模組或擷取卡。
C#程式範例	<pre>StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; bool IsRet = iRemoting.CHECK_rs232_data_received(_Pwd, 0); MessageBox.Show(IsRet.ToString());</pre>

5. GET_rs232_data_received：取得 RS232 所傳回的資料

API 名稱	GET_rs232_data_received
函式說明	取得 RS232 所傳回的資料

C#方法	short GET_rs232_data_received(StructMsg.Pwd P, int index, ref StructMsg.RS232_Data R);		
VB.Net 方法	GET_rs232_data_received(ByVal index As Integer,ByRef R As StructMsg.RS232_Data) As Short		
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：RS232_Data		
	變數名稱	型別	[in/out] 說明
	Text	string	out 傳回字串
	ArrayByte	byte[]	out 傳回 byte 陣列
備註	<p>※index:指向 SkyMars 第幾個陣列(第幾組)的序列埠。</p> <p>※選擇 Text 或 ArrayByte 的變數，請依照之前使用的 OPEN_rs232 函式內的 SerialType 來選擇，如 SerialType=0，由 Text 變數取得傳回值。</p> <p>如果是 SerialType=1，請由 ArrayByte 變數取得傳回值。</p>		
適用範圍	RS-232 模組或擷取卡。		
C#程式範例	<pre>StructMsg.RS232_Data _RS232_Data; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_rs232_data_received(_Pwd, 0, ref _RS232_Data); byte[] readByte = _RS232_Data.ArrayByte; }</pre>		

6. CLOSE_rs232：關閉 RS232

API 名稱	CLOSE_rs232
函式說明	取得 RS232 所傳回的資料
C#方法	void CLOSE_rs232(StructMsg.Pwd P, int index);
VB.Net 方法	CLOSE_rs232(index As Integer)
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。
備註	※index:指向 SkyMars 第幾個陣列(第幾組)的序列埠。
適用範圍	RS-232 模組或擷取卡。
C#程式範例	<pre>StructMsg.RS232_Data _RS232_Data; StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; iRemoting.CLOSE_rs232(_Pwd, 0);</pre>

7. GET_rs232_error：取得 RS-232 例外錯誤訊息

API 名稱	GET_rs232_error			
函式說明	取得 RS-232 例外錯誤訊息			
C#方法	StructMsg.SkyException GET_rs232_error_msg(StructMsg.Pwd P, int index);			
VB.Net 方法	GET_rs232_error_msg(index As Integer) As StructMsg.SkyException			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：SkyException			
	變數名稱	型別	[in/out]	說明
	IsException	bool	out	判斷是否發生例外錯誤
	ExceptionMsg	string	out	錯誤訊息
	ErrorCode	int	out	錯誤碼
備註				
適用範圍	RS-232 模組或擷取卡。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; StructMsg.SkyException R = iRemoting.GET_rs232_error(_Pwd, 0); MessageBox.Show(R.ExceptionMsg + "\r\n" + R.ErrorCode.ToString());</pre>			

十九、 PC Camera 模組類別(USB)

1. CONNECT_cam：連線攝影機

API 名稱	CONNECT_cam		
函式說明	連線攝影機		
C#方法	short CONNECT_cam(StructMsg.Pwd P, StructMsg.Camera_Param R);		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Camera_Param		
	變數名稱	型別	[in/out] 說明
	VideoDevice	int	in Camera 的 Index(從 0 開始)
	VideoWidth	int	in 擷取影像寬度
	VideoHeight	int	in 擷取影像高度
備註	※影像寬度(VideoWidth)若依預設值，請設定-1 ※影像高度若依預設值(VideoHeight)，請設定-1 ※當連線成功後，CPU 使用率將會升高。 ※一部電腦只限連接一台攝影機。		
適用範圍	USB PC Camera 或 USB Web Camera		
C#程式範例	<pre> _Camera_Param.VideoDevice = 0; _Camera_Param.VideoWidth = -1; _Camera_Param.VideoHeight = -1; StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; short ret = iRemoting.CONNECT_cam(_Pwd, _Camera_Param); if (ret == 0) MessageBox.Show("連接成功"); else MessageBox.Show("連接失敗"); </pre>		

2. DISCONNECT_cam：關閉攝影機

API 名稱	DISCONNECT_cam
函式說明	關閉攝影機
C#方法	void DISCONNECT_cam(StructMsg.Pwd P);
VB.Net 方法	
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。
備註	

適用範圍	USB PC Camera 或 USB Web Camera
C#程式範例	<pre>StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; iRemoting.DISCONNECT_cam(_Pwd);</pre>

3. CHECK_cam_running：確認攝影機是否已連接及擷取

API 名稱	CHECK_cam_running
函式說明	確認攝影機是否已連接及擷取
C#方法	bool CHECK_cam_running(StructMsg.Pwd P);
VB.Net 方法	
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。
備註	
適用範圍	USB PC Camera 或 USB Web Camera
C#程式範例	<pre>StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; bool IsRunning = iRemoting.CHECK_cam_running(_Pwd); if (IsRunning) MessageBox.Show("影像擷取中"); else MessageBox.Show("尚未連接 Camera");</pre>

4. GET_cam_image：取得攝影機影像

API 名稱	GET_cam_image								
函式說明	取得攝影機影像								
C#方法	short GET_cam_image(StructMsg.Pwd P, ref StructMsg.Camera_Image R);								
VB.Net 方法									
資料結構	<p>結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。</p> <p>結構名稱：Camera_Image</p> <table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>CamImage</td> <td>byte[]</td> <td>out</td> <td>送出影像 byte 陣列</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	CamImage	byte[]	out	送出影像 byte 陣列
變數名稱	型別	[in/out]	說明						
CamImage	byte[]	out	送出影像 byte 陣列						
備註	※ 變數 CamImage 轉換後屬於 Bitmap 型別。								
適用範圍	USB PC Camera 或 USB Web Camera								
C#程式範例	<pre>StructMsg.Camera_Image _Camera_Image; StructMsg.Pwd _Pwd; private void timer1_Tick(object sender, EventArgs e) { _Pwd.ConnectionKey = "123";</pre>								

```

short ret = iRemoting.GET_cam_image(_Pwd, ref _Camera_Image);
if (ret == 0)
{
    if (_Camera_Image.CamImage == null ||
_Camera_Image.CamImage.Length == 0) { return; }

    //Image oImage = null;
    Bitmap oBitmap = null;
    using (MemoryStream oMemoryStream = new
MemoryStream(_Camera_Image.CamImage))
    {
        //設定資料流位置
        oMemoryStream.Position = 0;
        oBitmap = new Bitmap(oMemoryStream);
        //oBitmap =
System.Drawing.Bitmap.FromStream(oMemoryStream);
        pictureBox1.Image = oBitmap;
    }
}
}

```

5. GET_cam_error：取得攝影機例外錯誤訊息

API 名稱	GET_cam_error			
函式說明	取得攝影機例外錯誤訊息			
C#方法	StructMsg.SkyException GET_cam_error(StructMsg.Pwd P);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：SkyException			
	變數名稱	型別	[in/out]	說明
	IsException	bool	out	判斷是否發生例外錯誤
	ExceptionMsg	string	out	錯誤訊息
	ErrorCode	int	out	錯誤碼
備註				
適用範圍	USB PC Camera 或 USB Web Camera			
C#程式範例	StructMsg.Pwd _Pwd; _Pwd.ConnectionKey = "123"; StructMsg.SkyException R = iRemoting.GET_cam_error(_Pwd); MessageBox.Show(R.ExceptionMsg + "\r\n" + R.ErrorCode.ToString());			

二十、 SkyMars Professional 事件派送

SkyMars Professional 本身內建事件派送的架構，就是讓應用程式端可向 SkyMars Professional 註冊事件，當 SkyMars Professional 有事件發生時，再通知應用程式。

另一方面，應用程式端的事件派送採用不同的執行緒進行，您必須建立委派(delegate)來呼叫，以防止跨執行緒問題。

在建立事件派送前，您必須先建立 EventMsg 類別，並建立執行個體：

建構函式	public EventMsg(string IP, int Port);
參數	IP SkyMars 的 IP。應用程式及 SkyMars 都在同一台電腦，請輸入「localhost」。 Port SkyMars 的 Port。預設「9500」。

- 請注意，建議將應用程式及 SkyMars 放置於同一台電腦。如果分開放有可能會造成事件派送傳遞失敗。
- 完整程式碼可參考如下：

```
private void Form1_Load(object sender, EventArgs e)
{
    try
    {
        InterfaceLib.EventMsg eMsg = new InterfaceLib.EventMsg("localhost", 9500);
        eMsg.EVENT_conn_cnc += new InterfaceLib.dele_conn_cnc(eMsg_EVENT_conn_cnc);
        eMsg.EVENT_conn_status += new
            InterfaceLib.dele_conn_status(eMsg_EVENT_conn_status);
        eMsg.EVENT_disconn_cnc += new
            InterfaceLib.dele_disconn_cnc(eMsg_EVENT_disconn_cnc);
        eMsg.EVENT_nc_filename += new
            InterfaceLib.dele_nc_filename(eMsg_EVENT_nc_filename);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

delegate void ShowDataDelegate(string info);
private void ShowData(string info)
{
    textBox1.Text = info;
}

void eMsg_EVENT_conn_cnc(InterfaceLib.StructMsg.SkyConn_ip_port2 R)
{
    //產生 SkyMars Professional 設定 CNC 資訊完成事件
    ShowDataDelegate _ShowDataDelegate = new ShowDataDelegate(ShowData);
    this.BeginInvoke(_ShowDataDelegate, new object[] { R.MachineNo + "," +
        R.MachineName + "," + R.Manufacturer });
}

void eMsg_EVENT_disconn_cnc(bool IsExitSkyMars)
{

```



```

//產生 SkyMars Professional 結束 CNC 連線事件
if (IsExitSkyMars == false)
{
    ShowDataDelegate _ShowDataDelegate = new ShowDataDelegate(ShowData);
    this.BeginInvoke(_ShowDataDelegate, new object [] { "發生結束 CNC 事件" });
}
else
{
    this.Close();
}
}

void eMsg_EVENT_conn_status(InterfaceLib.StructMsg.SkyConn_status2 R)
{
    //產生 CNC 狀態事件(OFF、RUN、IDLE、ALARM)
    ShowDataDelegate _ShowDataDelegate = new ShowDataDelegate(ShowData);
    this.BeginInvoke(_ShowDataDelegate, new object [] { R.MachineNo + ", " + R.Status });
}

void eMsg_EVENT_nc_filename(InterfaceLib.StructMsg.SkyNc_filename2 R)
{
    //產生 CNC 切換 NC 檔名事件
    ShowDataDelegate _ShowDataDelegate = new ShowDataDelegate(ShowData);
    this.BeginInvoke(_ShowDataDelegate, new object [] { R.MachineNo + ", " + R.MainProg
        + ", " + R.SubProg });
}
    
```

1. EVENT_conn_cnc：產生 SkyMars Professional 設定 CNC 資訊完成事件

API 名稱	EVENT_conn_cnc			
函式說明	產生 SkyMars Professional 設定 CNC 資訊完成事件			
C#方法	event EventHandler_conn_cnc EVENT_conn_cnc;			
VB.Net 方法				
資料結構	結構名稱：SkyConn_ip_port2			
	變數名稱	型別	[in/out]	說明
	MachineNo	int	out	機器的索引值
	MachineName	string	out	機台名稱
	IP	string	out	機台 IP 位址
	Port	short	out	機台 Port 號碼
	Manufacturer	string	out	廠牌
	PlugInPort	int	out	Plug-In Port 號碼
備註				
適用範圍	所有控制器			
C#程式範例	<pre> 1.Create event Wrapper.Local_EVENT_conn_cnc += new EventHandler_conn_cnc(Wrapper_EVENT_conn_cnc); iEvent.EVENT_conn_cnc += new EventHandler_conn_cnc(Wrapper.BC_conn_cnc); </pre>			

	<pre> 2.Create method delegate void ShowDataDelegate(string info); void Wrapper_EVENT_conn_cnc(StructMsg.SkyConn_ip_port2 R) { ShowDataDelegate _DisplayInfoDelegate = new ShowDataDelegate(ShowData); this.BeginInvoke(_DisplayInfoDelegate, new object[] { "Connection CNC" }); } private void ShowData(string info) { richTextBox1.Text += info + "\r\n"; } </pre>
--	--

2. EVENT_disconn_cnc：產生 SkyMars Professional 結束 CNC 連線事件

API 名稱	EVENT_disconn_cnc											
函式說明	產生 SkyMars Professional 結束 CNC 連線事件											
C#方法	event EventHandler_disconn_cnc EVENT_disconn_cnc;											
VB.Net 方法												
資料結構	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">變數名稱</th> <th style="width: 15%;">型別</th> <th style="width: 15%;">[in/out]</th> <th style="width: 40%;">說明</th> </tr> </thead> <tbody> <tr> <td>IsExitSkyMars</td> <td>bool</td> <td>out</td> <td>是否結束 SkyMars 平台</td> </tr> </tbody> </table>				變數名稱	型別	[in/out]	說明	IsExitSkyMars	bool	out	是否結束 SkyMars 平台
變數名稱	型別	[in/out]	說明									
IsExitSkyMars	bool	out	是否結束 SkyMars 平台									
備註	當結束 SkyMars Professional 平台時，事件會派送至應用程式，IsExitSkyMars 變數會傳出 true，可依此變數來結束應用程式。											
適用範圍	所有控制器											
C#程式範例	<pre> 1.Create event Wrapper.Local_EVENT_disconn_cnc += new EventHandler_disconn_cnc(Wrapper_EVENT_disconn_cnc); iEvent.EVENT_disconn_cnc += new EventHandler_disconn_cnc(Wrapper.BC_disconn_cnc); 2.Create method delegate void ShowDataDelegate(string info); void Wrapper_EVENT_disconn_cnc(bool IsExitSkyMars) { ShowDataDelegate _DisplayInfoDelegate = new ShowDataDelegate(ShowData); this.BeginInvoke(_DisplayInfoDelegate, new object[] </pre>											

```

        { "Disconnection CNC" });
    }

    private void ShowData(string info)
    {
        richTextBox1.Text += info + "\r\n";
    }
    
```

3. EVENT_conn_status：產生 CNC 狀態事件(OFF、RUN、IDLE、ALARM)

API 名稱	EVENT_conn_status			
函式說明	產生 CNC 狀態事件(OFF、RUN、IDLE、ALARM)			
C#方法	event EventHandler_conn_status EVENT_conn_status;			
VB.Net 方法				
資料結構	結構名稱：SkyConn_status2			
	變數名稱	型別	[in/out]	說明
	MachineNo	int	out	機器的索引值
	Status	short	out	0:未開機 1:RUN 2:IDLE 3:Alarm
備註				
適用範圍	所有控制器			
C#程式範例	<pre> 1.Create event Wrapper.Local_EVENT_conn_status += new EventHandler_conn_status(Wrapper_EVENT_conn_status); iEvent.EVENT_conn_status += new EventHandler_conn_status(Wrapper_BC_conn_status); 2.Create method delegate void ShowDataDelegate(string info); void Wrapper_EVENT_conn_status(StructMsg.SkyConn_status2 R) { ShowDataDelegate _DisplayInfoDelegate = new ShowDataDelegate(ShowData); string StatusName = ""; switch (R.Status) { case 0: StatusName = "未開機"; break; </pre>			

	<pre> case 1: StatusName = "RUN"; break; case 2: StatusName = "IDLE"; break; case 3: StatusName = "Alarm"; break; } this.BeginInvoke(_DisplayInfoDelegate, new object[] { StatusName }); } private void ShowData(string info) { richTextBox1.Text += info + "\r\n"; } </pre>
--	--

4. EVENT_nc_filename：產生 CNC 切換 NC 檔名事件

API 名稱	EVENT_nc_filename			
函式說明	產生 CNC 切換 NC 檔名事件			
C#方法	event EventHandler_conn_status EVENT_conn_status;			
VB.Net 方法				
資料結構	結構名稱：SkyNc_filename			
	變數名稱	型別	[in/out]	說明
	MachineNo	int	out	機器的索引值
	MainProg	string	out	主程式檔名
	SubProg	string	out	副程式檔名
備註				
適用範圍	所有控制器			
C#程式範例	<pre> 1.Create event Wrapper.Local_EVENT_nc_filename += new EventHandler_nc_filename(Wrapper_EVENT_nc_filename); iEvent.EVENT_nc_filename += new EventHandler_nc_filename(Wrapper.BC_nc_filename); 2.Create method </pre>			

```
delegate void ShowDataDelegate(string info);  
void Wrapper_EVENT_nc_filename(StructMsg.SkyNc_filename2 R)  
{  
    ShowDataDelegate _DisplayInfoDelegate = new  
        ShowDataDelegate(ShowData);  
    this.BeginInvoke(_DisplayInfoDelegate, new object[]  
        { R.MainProg + ", " + R.SubProg });  
}  
  
private void ShowData(string info)  
{  
    richTextBox1.Text += info + "\r\n";  
}
```

二十一、 SkyMars 稼動率資訊

1. GET_utilization_all_today：取得今日機台稼動率資訊

API 名稱	GET_utilization_all_today		
函式說明	取得今日機台稼動率資訊		
C#方法	short GET_utilization_all_today(StructMsg.Pwd P, ref StructMsg.Utilization_Today R)		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Utilization_Today		
	變數名稱	型別	[in/out] 說明
	PartTotal	int[]	out 今日工件總數
	CycleTotal	int[]	out 今日加工循環總數
	BusyTotal	double[]	out 今日加工總時間(單位:小時)
	IdleTotal	double[]	out 今日閒置總時間(單位:小時)
	AlarmTotal	double[]	out 今日異常總時間(單位:小時)
	OffTotal	double[]	out 今日關機總時間(單位:小時)
備註	※ 請連接 Port 9501 傳送及擷取稼動率資訊。 ※ BusyTotal 含有 Busy Time 及 BUSY&ALARM Time。		
適用範圍	所有控制器。		
C#程式範例	<pre>StructMsg.Utilization_Today _Util_Today; StructMsg.Pwd _Pwd; private void button1_Click(object sender, EventArgs e) { _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_utilization_all_today(_Pwd, ref _Util_Today); if (ret == 0) { txtMachineName.Text = _Util_Today.MachineName[0]; txtPartTotal.Text = _Util_Today.PartTotal[0].ToString(); txtCycleTotal.Text = _Util_Today.CycleTotal[0].ToString(); txtBusyTotal.Text = _Util_Today.BusyTotal[0].ToString(); txtIdleTotal.Text = _Util_Today.IdleTotal[0].ToString(); txtAlarmTotal.Text = _Util_Today.AlarmTotal[0].ToString(); txtOffTotal.Text = _Util_Today.OffTotal[0].ToString(); } }</pre>		

2. GET_utilization_single_time：取得單一機台總時間之稼動率資訊

API 名稱	GET_utilization_single_time			
函式說明	取得單一機台總時間之稼動率資訊			
C#方法	short GET_utilization_single_time(StructMsg.Pwd P, ref StructMsg.Utilization R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Utilization			
	變數名稱	型別	[in/out]	說明
	MachineName	string	in	機器名稱
	StDate	DateTime	in	起始日期時間
	EndDate	DateTime	in	結束日期時間
	BusyTotal	double	out	加工總時間(單位:小時)
	IdleTotal	double	out	閒置總時間(單位:小時)
	AlarmTotal	double	out	異常總時間(單位:小時)
OffTotal	double	out	關機總時間(單位:小時)	
備註	<p>※ 請連接 Port 9501 傳送及擷取稼動率資訊。</p> <p>※ BusyTotal 含有 Busy Time 及 BUSY&ALARM Time。</p> <p>※ 如果要查詢同一天資料,StDate 及 EndDate 建議使用完整的日期時間,如: StDate: 2014/5/7 00:00:00.000 EndDate: 2014/5/7 23:59:59.999</p>			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Utilization _Util; StructMsg.Pwd _Pwd; private void button2_Click(object sender, EventArgs e) { _Util.MachineName = "Demo"; _Util.StDate = DateTime.Now; _Util.EndDate = DateTime.Now; _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_utilization_single_time(_Pwd, ref _Util); if (ret == 0) { txtBusyTotal2.Text = _Util.BusyTotal.ToString(); txtIdleTotal2.Text = _Util.IdleTotal.ToString(); txtAlarmTotal2.Text = _Util.AlarmTotal.ToString(); txtOffTotal2.Text = _Util.OffTotal.ToString(); } }</pre>			

3. GET_utilization_single_total：取得單一機台總次數(工件數、加工循環次數)之稼動率資訊

API 名稱	GET_utilization_single_total			
函式說明	取得單一機台總次數(工件數、加工循環次數)之稼動率資訊			
C#方法	short GET_utilization_single_total(StructMsg.Pwd P, ref StructMsg.Utilization R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Utilization			
	變數名稱	型別	[in/out]	說明
	MachineName	string	in	機器名稱
	StartDate	DateTime	in	起始日期時間
	EndDate	DateTime	in	結束日期時間
	PartTotal	int	out	工件總數
	CycleTotal	int	out	加工循環總數
備註	※ 請連接 Port 9501 傳送及擷取稼動率資訊。 ※ 如果要查詢同一天資料,StartDate 及 EndDate 建議使用完整的日期時間,如: StartDate: 2014/5/7 00:00:00.000 EndDate: 2014/5/7 23:59:59.999			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Utilization _Util; StructMsg.Pwd _Pwd; private void button3_Click(object sender, EventArgs e) { _Util.MachineName = "Demo"; _Util.StartDate = DateTime.Now; _Util.EndDate = DateTime.Now; _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_utilization_single_total(_Pwd, ref _Util); if (ret == 0) { txtPartTotal2.Text = _Util.PartTotal.ToString(); txtCycleTotal2.Text = _Util.CycleTotal.ToString(); } }</pre>			

4. GET_utilization_single_list：取得單一機台稼動率履歷資訊

API 名稱	GET_utilization_single_list			
函式說明	取得單一機台稼動率履歷資訊			
C#方法	short GET_utilization_single_list(StructMsg.Pwd P, ref StructMsg.Utilization_List R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Utilization_List			
	變數名稱	型別	[in/out]	說明
	MachineName	string	in	機器名稱
	StDate	DateTime	in	起始日期時間
	EndDate	DateTime	in	結束日期時間
ListData	string[][]	out	履歷資訊	
備註	<p>※ 請連接 Port 9501 傳送及擷取稼動率資訊。</p> <p>※ ListData 屬於雙陣列(不規則陣列)，所包含的陣列索引識別如下：</p> <p style="padding-left: 20px;">ListData[][0]:狀態 (OFF、IDLE、ALARM、BUSY、BUSY&ALARM)</p> <p style="padding-left: 20px;">ListData[][1]:起始時間</p> <p style="padding-left: 20px;">ListData[][2]:結束時間</p> <p style="padding-left: 20px;">ListData[][3]:總時間(結束時間 - 起始時間)</p> <p style="padding-left: 20px;">ListData[][4]:工件數</p> <p style="padding-left: 20px;">ListData[][5]:加工程式名稱</p> <p style="padding-left: 20px;">ListData[][6]:警報訊息</p> <p>※ 如果要查詢同一天資料,StDate 及 EndDate 建議使用完整的日期時間,如：</p> <p style="padding-left: 20px;">StDate: 2014/5/7 00:00:00.000</p> <p style="padding-left: 20px;">EndDate: 2014/5/7 23:59:59.999</p>			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Utilization_List _RList; StructMsg.Pwd _Pwd; private void button4_Click(object sender, EventArgs e) { _RList.MachineName = "Demo"; _RList.StDate = DateTime.Now.AddDays(-10); _RList.EndDate = DateTime.Now; _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_utilization_single_list(_Pwd, ref _RList); if (ret == 0) {</pre>			

	<pre> DataTable dt = new DataTable(); dt.Columns.Add("STATUS", Type.GetType("System.String")); dt.Columns.Add("StTIME", Type.GetType("System.String")); dt.Columns.Add("EndTIME", Type.GetType("System.String")); dt.Columns.Add("TotalTIME", Type.GetType("System.String")); dt.Columns.Add("PARTCOUNT", Type.GetType("System.String")); dt.Columns.Add("NCNAME", Type.GetType("System.String")); for (int i = 0; i <= _RList.ListData.Length - 1; i++) { DataRow dr = dt.NewRow(); dr[0] = _RList.ListData[i][0]; dr[1] = _RList.ListData[i][1]; dr[2] = _RList.ListData[i][2]; dr[3] = _RList.ListData[i][3]; dr[4] = _RList.ListData[i][4]; dr[5] = _RList.ListData[i][5]; dt.Rows.Add(dr); } dataGridView1.DataSource = dt; } } </pre>
--	--

5. GET_utilization_single_all：取得單一機台稼動率資訊及履歷

API 名稱	GET_utilization_single_all		
函式說明	取得單一機台稼動率資訊及履歷		
C#方法	short GET_utilization_single_all(StructMsg.Pwd P, ref StructMsg.Utilization R , ref StructMsg.Utilization_List RList)		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Utilization		
	變數名稱	型別	[in/out] 說明
	MachineName	string	in 機器名稱
	StDate	DateTime	in 起始日期時間
	EndDate	DateTime	in 結束日期時間
	BusyTotal	double	out 加工總時間(單位:小時)
	IdleTotal	double	out 閒置總時間(單位:小時)
	AlarmTotal	double	out 異常總時間(單位:小時)

	<table border="1"> <tr> <td>OffTotal</td> <td>double</td> <td>out</td> <td>關機總時間(單位:小時)</td> </tr> <tr> <td>PartTotal</td> <td>int</td> <td>out</td> <td>工件總數</td> </tr> <tr> <td>CycleTotal</td> <td>int</td> <td>out</td> <td>加工循環總數</td> </tr> </table>	OffTotal	double	out	關機總時間(單位:小時)	PartTotal	int	out	工件總數	CycleTotal	int	out	加工循環總數
OffTotal	double	out	關機總時間(單位:小時)										
PartTotal	int	out	工件總數										
CycleTotal	int	out	加工循環總數										
	<p>結構名稱: Utilization_List</p> <table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>ListData</td> <td>string[][]</td> <td>out</td> <td>履歷資訊</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	ListData	string[][]	out	履歷資訊				
變數名稱	型別	[in/out]	說明										
ListData	string[][]	out	履歷資訊										
備註	<p>※ 請連接 Port 9501 傳送及擷取稼動率資訊。</p> <p>※ ListData 屬於雙陣列(不規則陣列), 所包含的陣列索引識別如下:</p> <p>ListData[][0]:狀態 (OFF、IDLE、ALARM、BUSY、BUSY&ALARM)</p> <p>ListData[][1]:起始時間</p> <p>ListData[][2]:結束時間</p> <p>ListData[][3]:總時間(結束時間 - 起始時間),單位:秒</p> <p>ListData[][4]:工件數</p> <p>ListData[][5]:加工程式名稱</p> <p>ListData[][6]:警報訊息</p> <p>※ 如果要查詢同一天資料,StartDate 及 EndDate 建議使用完整的日期時間,如:</p> <p>StartDate: 2014/5/7 00:00:00.000</p> <p>EndDate: 2014/5/7 23:59:59.999</p>												
適用範圍	所有控制器。												
C#程式範例	<pre>StructMsg.Utilization _Util; StructMsg.Utilization_List _RList; StructMsg.Pwd _Pwd; private void button5_Click(object sender, EventArgs e) { _Util.MachineName = "Demo"; _Util.StartDate = DateTime.Now.AddDays(-10); _Util.EndDate = DateTime.Now; _Pwd.ConnectionKey = "123"; short ret = iRemoting.GET_utilization_single_all(_Pwd, ref _Util, ref _RList); if (ret == 0) { txtBusyTotal2.Text = _Util.BusyTotal.ToString(); txtIdleTotal2.Text = _Util.IdleTotal.ToString(); txtAlarmTotal2.Text = _Util.AlarmTotal.ToString(); txtOffTotal2.Text = _Util.OffTotal.ToString(); txtPartTotal2.Text = _Util.PartTotal.ToString(); } }</pre>												

```
txtCycleTotal2.Text = _Util.CycleTotal.ToString();

//處理履歷資料
DataTable dt = new DataTable();
dt.Columns.Add("STATUS", Type.GetType("System.String"));
dt.Columns.Add("StTIME", Type.GetType("System.String"));
dt.Columns.Add("EndTIME", Type.GetType("System.String"));
dt.Columns.Add("TotalTIME", Type.GetType("System.String"));
dt.Columns.Add("PARTCOUNT", Type.GetType("System.String"));
dt.Columns.Add("NCNAME", Type.GetType("System.String"));

for (int i = 0; i <= _RList.ListData.Length - 1; i++)
{
    DataRow dr = dt.NewRow();
    dr[0] = _RList.ListData[i][0];
    dr[1] = _RList.ListData[i][1];
    dr[2] = _RList.ListData[i][2];
    dr[3] = _RList.ListData[i][3];
    dr[4] = _RList.ListData[i][4];
    dr[5] = _RList.ListData[i][5];
    dt.Rows.Add(dr);
}

dataGridView1.DataSource = dt;
}
}
```

二十二、 SkyMars 維護管理類別

1. GET_maint_count：取得維護管理的總筆數

API 名稱	GET_maint_count			
函式說明	取得維護管理的總筆數			
C#方法	short GET_maint_count(StructMsg.Pwd P, ref StructMsg.total_count R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：total_count			
	變數名稱	型別	[in/out]	說明
	Count	short	out	總筆數
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.total_count _total_count; private void button1_Click(object sender, EventArgs e) { short ret = iRemoting.GET_maint_count(_Pwd, ref _total_count); MessageBox.Show(_total_count.Count.ToString()); }</pre>			

2. GET_maint_title：取得維護管理的標題列文字

API 名稱	GET_maint_title			
函式說明	取得維護管理的標題列文字			
C#方法	short GET_maint_title(StructMsg.Pwd P, ref StructMsg.Maint_Title R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Maint_Title			
	變數名稱	型別	[in/out]	說明
	Title	string[]	out	標題文字
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Maint_Title _Maint_Title; private void button1_Click(object sender, EventArgs e) { short ret = iRemoting.GET_maint_title(_Pwd, ref _Maint_Title); for (int i = 0; i <= _Maint_Title.Title.Length - 1; i++)</pre>			

	<pre> MessageBox.Show(_Maint_Title.Title[i]); } </pre>
--	--

3. GET_maint_all：取得維護管理所有資料

API 名稱	GET_maint_all																																						
函式說明	取得維護管理所有資料																																						
C#方法	short GET_maint_all(StructMsg.Pwd P, ref StructMsg.Maint_All R)																																						
VB.Net 方法																																							
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Maint_All <table border="1" data-bbox="475 676 1441 1122"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>ItemName</td> <td>string[]</td> <td>out</td> <td>項目名稱</td> </tr> <tr> <td>UseTime</td> <td>double[]</td> <td>out</td> <td>使用時間</td> </tr> <tr> <td>LifeTime</td> <td>double[]</td> <td>out</td> <td>壽命時間</td> </tr> <tr> <td>PlcNode</td> <td>string[]</td> <td>out</td> <td>PLC 節點</td> </tr> <tr> <td>Spec</td> <td>string[]</td> <td>out</td> <td>規格</td> </tr> <tr> <td>Supplier</td> <td>string[]</td> <td>out</td> <td>供應商</td> </tr> <tr> <td>Contact</td> <td>string[]</td> <td>out</td> <td>聯絡人</td> </tr> <tr> <td>Telephone</td> <td>string[]</td> <td>out</td> <td>電話</td> </tr> </tbody> </table>			變數名稱	型別	[in/out]	說明	ItemName	string[]	out	項目名稱	UseTime	double[]	out	使用時間	LifeTime	double[]	out	壽命時間	PlcNode	string[]	out	PLC 節點	Spec	string[]	out	規格	Supplier	string[]	out	供應商	Contact	string[]	out	聯絡人	Telephone	string[]	out	電話
變數名稱	型別	[in/out]	說明																																				
ItemName	string[]	out	項目名稱																																				
UseTime	double[]	out	使用時間																																				
LifeTime	double[]	out	壽命時間																																				
PlcNode	string[]	out	PLC 節點																																				
Spec	string[]	out	規格																																				
Supplier	string[]	out	供應商																																				
Contact	string[]	out	聯絡人																																				
Telephone	string[]	out	電話																																				
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。																																						
適用範圍	所有控制器。																																						
C#程式範例	<pre> StructMsg.Pwd _Pwd; StructMsg.Maint_All _Maint_All; private void LoadMaintData() { short ret = iRemoting.GET_maint_all(_Pwd, ref _Maint_All); for (int i = 0; i <= _Maint_All.ItemName.Length - 1; i++) { DataRow dr = dtMain.NewRow(); dr["ColStatus"] = Math.Round(_Maint_All.UseTime[i] / _Maint_All.LifeTime[i] * 100, 3) + "%"; dr["ColItem"] = _Maint_All.ItemName[i]; dr["ColUseTime"] = Math.Round(_Maint_All.UseTime[i], 3); dr["ColLiftTime"] = Math.Round(_Maint_All.LifeTime[i], 3); dr["ColPlc"] = _Maint_All.PlcNode[i]; dr["ColSpec"] = _Maint_All.Spec[i]; dr["ColMan"] = _Maint_All.Supplier[i]; dr["ColContact"] = _Maint_All.Contact[i]; } } </pre>																																						

	<pre> dr["ColTel"] = _Maint_All.Telephone[i]; dtMain.Rows.Add(dr); } } </pre>
--	---

4. GET_maint_all_time：取得維護管理項目內的所有時間(使用時間、壽命時間)

API 名稱	GET_maint_all_time			
函式說明	取得維護管理項目內的所有時間(使用時間、壽命時間)			
C#方法	short GET_maint_all_time(StructMsg.Pwd P, ref StructMsg.Maint_All_Time R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Maint_All_Time			
	變數名稱	型別	[in/out]	說明
	ItemName	string[]	out	項目名稱
	UseTime	double[]	out	使用時間
	LifeTime	double[]	out	壽命時間
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre> StructMsg.Pwd _Pwd; StructMsg.Maint_All_Time _Maint_All_Time; private void button1_Click(object sender, EventArgs e) { short ret = iRemoting.GET_maint_all_time(_Pwd, ref _Maint_All_Time); if (ret == 0) { for (int i = 0; i <= dtMain.Rows.Count - 1; i++) { dtMain.Rows[i]["ColUseTime"] = _Maint_All_Time.UseTime[i]; dtMain.Rows[i]["ColliftTime"] = _Maint_All_Time.LifeTime[i]; } } } </pre>			

5. GET_maint_single：取得單筆的維護管理的項目資料

API 名稱	GET_maint_single
--------	-------------------------

函式說明	取得單筆的維護管理的項目資料			
C#方法	short GET_maint_single(StructMsg.Pwd P, ref StructMsg.Maint_Single R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Maint_Single			
	變數名稱	型別	[in/out]	說明
	ItemName	string	in	項目名稱
	UseTime	double	out	使用時間
	LifeTime	double	out	壽命時間
	PlcNode	string	out	PLC 節點
	Spec	string	out	規格
	Supplier	string	out	供應商
Contact	string	out	聯絡人	
Telephone	string	out	電話	
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Maint_Single _Maint_Single; private void button1_Click(object sender, EventArgs e) { _Maint_Single.ItemName = "潤滑油"; short ret = iRemoting.GET_maint_single(_Pwd, ref _Maint_Single); if (ret == 0) { //... } }</pre>			

6. GET_maint_single_time：取得單筆維護管理的時間資料(使用時間、壽命時間)

API 名稱	GET_maint_single_time		
函式說明	取得單筆維護管理的時間資料(使用時間、壽命時間)		
C#方法	short GET_maint_single_time(StructMsg.Pwd P, ref StructMsg.Maint_Single_Time R)		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：Maint_Single_Time		
變數名稱	型別	[in/out]	說明

	ItemName	string	in	項目名稱
	UseTime	double	out	使用時間
	LifeTime	double	out	壽命時間
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Maint_Single_Time _Maint_Single_Time; private void button1_Click(object sender, EventArgs e) { _Maint_Single.ItemName = "潤滑油"; short ret = iRemoting.GET_maint_single_time(_Pwd, ref _Maint_Single_Time); if (ret == 0) { //... } }</pre>			

7. ADD_maint_single：新增單筆維護管理資料

API 名稱	ADD_maint_single			
函式說明	新增單筆維護管理資料			
C#方法	short ADD_maint_single(StructMsg.Pwd P, StructMsg.Maint_Single R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Maint_Single			
	變數名稱	型別	[in/out]	說明
	ItemName	string	in	項目名稱
	UseTime	double	in	使用時間
	LifeTime	double	in	壽命時間
	PlcNode	string	in	PLC 節點
	Spec	string	in	規格
	Supplier	string	in	供應商
	Contact	string	in	聯絡人
	Telephone	string	in	電話
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Maint_Single _Maint_Single;</pre>			

```

private void button1_Click(object sender, EventArgs e)
{
    _Maint_Single.ItemName = dt.Rows[i]["ColItem"].ToString();
    _Maint_Single.UseTime = (double)dt.Rows[i]["ColUseTime"];
    _Maint_Single.LifeTime = (double)dt.Rows[i]["ColLiftTime"];
    _Maint_Single.PlcNode = dt.Rows[i]["ColPlc"].ToString();
    _Maint_Single.Spec = dt.Rows[i]["ColSpec"].ToString();
    _Maint_Single.Supplier = dt.Rows[i]["ColMan"].ToString();
    _Maint_Single.Contact = dt.Rows[i]["ColContact"].ToString();
    _Maint_Single.Telephone = dt.Rows[i]["ColTel"].ToString();

    ret = iRemoting.ADD_maint_single(_Pwd, _Maint_Single);
    if (ret != 0)
    {
        MessageBox.Show("寫入失敗!");
        break;
    }
}

```

8. SET_maint_single：修改單筆維護管理的資料

API 名稱	SET_maint_single		
函式說明	修改單筆維護管理的資料		
C#方法	short SET_maint_single(StructMsg.Pwd P, StructMsg.Maint_Single R)		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Maint_Single		
	變數名稱	型別	[in/out] 說明
	Original_ItemName	string	in 原始尚未修改的項目名稱
	ItemName	string	in 項目名稱
	UseTime	double	in 使用時間
	LifeTime	double	in 壽命時間
	PlcNode	string	in PLC 節點
	Spec	string	in 規格
	Supplier	string	in 供應商
	Contact	string	in 聯絡人
	Telephone	string	in 電話
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。		
適用範圍	所有控制器。		

C#程式範例	<pre> StructMsg.Pwd _Pwd; StructMsg.Maint_Single _Maint_Single; private void button1_Click(object sender, EventArgs e) { _Maint_Single.Original_ItemName = dt.Rows[i]["ColItem", DataRowVersion.Original].ToString(); _Maint_Single.ItemName = dt.Rows[i]["ColItem"].ToString(); _Maint_Single.LifeTime = (double)dt.Rows[i]["ColLiftTime"]; _Maint_Single.PlcNode = dt.Rows[i]["ColPlc"].ToString(); _Maint_Single.Spec = dt.Rows[i]["ColSpec"].ToString(); _Maint_Single.Supplier = dt.Rows[i]["ColMan"].ToString(); _Maint_Single.Contact = dt.Rows[i]["ColContact"].ToString(); _Maint_Single.Telephone = dt.Rows[i]["ColTel"].ToString(); ret = iRemoting.SET_maint_single(_Pwd, _Maint_Single); if (ret != 0) { MessageBox.Show("寫入失敗!"); break; } } </pre>
--------	--

9. DEL_maint_single：刪除單筆維護管理的資料

API 名稱	DEL_maint_single			
函式說明	刪除單筆維護管理的資料			
C#方法	short DEL_maint_single(StructMsg.Pwd P, StructMsg.Maint_ItemName R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Maint_ItemName			
	變數名稱	型別	[in/out]	說明
	Name	string	in	項目名稱
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre> StructMsg.Pwd _Pwd; StructMsg.Maint_ItemName _Maint_ItemName; private void button1_Click(object sender, EventArgs e) { _Maint_ItemName.Name = dt.Rows[i]["ColItem", </pre>			

	<pre> DataRowVersion.Original].ToString()); ret = iRemoting.DEL_maint_single(_Pwd, _Maint_ItemName); if (ret != 0) { MessageBox.Show("刪除失敗!"); break; } } </pre>
--	---

10. SET_maint_single_usetime：設定單筆維護管理的使用時間

API 名稱	SET_maint_single_usetime			
函式說明	設定單筆維護管理的使用時間			
C#方法	short SET_maint_single_usetime(StructMsg.Pwd P, StructMsg.Maint_Single_UseTime R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Maint_Single_UseTime			
	變數名稱	型別	[in/out]	說明
	ItemName	string	in	項目名稱
	UseTime	double	in	使用時間
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre> StructMsg.Pwd _Pwd; StructMsg.Maint_Single_UseTime _Maint_Single_UseTime; private void button1_Click(object sender, EventArgs e) { _Maint_Single_UseTime.ItemName = "潤滑油"; _Maint_Single_UseTime.UseTime = 10; short ret = iRemoting.SET_maint_single_usetime(_Pwd, _Maint_Single_UseTime); } </pre>			

11. SET_maint_single_zero：將單筆維護管理的使用時間歸零

API 名稱	SET_maint_single_zero			
函式說明	將單筆維護管理的使用時間歸零			
C#方法	short SET_maint_single_zero(StructMsg.Pwd P, StructMsg.Maint_ItemName R)			
VB.Net 方法				

資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Maint_ItemName			
	變數名稱	型別	[in/out]	說明
	Name	string	in	項目名稱
備註	※ 請連接 Port 9501 傳送及擷取維護管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Maint_ItemName _Maint_ItemName; private void button1_Click(object sender, EventArgs e) { _Maint_ItemName.Name = dgvMain.Rows[e.RowIndex].Cells["ColItem"].Value.ToString(); short ret = iRemoting.SET_maint_single_zero(_Pwd, _Maint_ItemName); if (ret != 0) MessageBox.Show("歸零失敗!"); }</pre>			

二十三、 SkyMars 刀具資料庫管理類別

以下表格所列函式皆已過時，我們建議您使用新的函式取代：

已過時的函式	新的函式
GET_toolmanage_title	GET_toolmanage_title2
GET_toolmanage_all	GET_toolmanage_all2
GET_toolmanage_all_time	GET_toolmanage_all_execute
GET_toolmanage_single	GET_toolmanage_single2
GET_toolmanage_single_time	GET_toolmanage_single_execute
ADD_toolmanage_single	ADD_toolmanage_single2
SET_toolmanage_single	SET_toolmanage_single2

1. GET_toolmanage_count：取得刀具資料庫管理的總筆數

API 名稱	GET_toolmanage_count			
函式說明	取得刀具資料庫管理的總筆數			
C#方法	short GET_toolmanage_count(StructMsg.Pwd P, ref StructMsg.total_count R)			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：total_count			
	變數名稱	型別	[in/out]	說明
	Count	short	out	總筆數
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.total_count _total_count; private void button1_Click(object sender, EventArgs e) { short ret = iRemoting.GET_toolmanage_count(_Pwd, ref _total_count); MessageBox.Show(_total_count.Count.ToString()); }</pre>			

2. GET_toolmanage_title2：取得刀具資料庫管理的標題列文字

API 名稱	GET_toolmanage_title2			
函式說明	取得刀具資料庫管理的標題列文字			
C#方法	short GET_toolmanage_title2(StructMsg.Pwd P, ref StructMsg.Toolmanage_Title R)			
VB.Net 方法				

資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。							
	結構名稱：Toolmanage_Title							
	<table border="1"> <thead> <tr> <th>變數名稱</th> <th>型別</th> <th>[in/out]</th> <th>說明</th> </tr> </thead> <tbody> <tr> <td>Title</td> <td>string[]</td> <td>out</td> <td>標題文字</td> </tr> </tbody> </table>	變數名稱	型別	[in/out]	說明	Title	string[]	out
變數名稱	型別	[in/out]	說明					
Title	string[]	out	標題文字					
備註	<p>※ 請連接 Port 9501 傳送及擷取刀具管理資訊。</p> <p>Title 變數陣列如下：</p> <pre>[0] = "ToolNo"; [1] = "ToolType"; [2] = "CycleTime"; [3] = "LifeTime"; [4] = "PartCount"; [5] = "LifePartCount"; [6] = "FeedRate"; [7] = "SpindleSpeed"; [8] = "WorkMaterial"; [9] = "ToolMaterial"; [10] = "Diameter"; [11] = "ToolLength"; [12] = "Supplier"; [13] = "Contact"; [14] = "Telephone";</pre>							
適用範圍	所有控制器。							
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Title _Toolmanage_Title; private void button1_Click(object sender, EventArgs e) { short ret = iRemoting.GET_toolmanage_title2(_Pwd, ref _Toolmanage_Title); for (int i = 0; i <= _Toolmanage_Title.Title.Length - 1; i++) MessageBox.Show(_Toolmanage_Title.Title[i]); }</pre>							

3. GET_toolmanage_all2：取得刀具資料庫管理所有資料

API 名稱	GET_toolmanage_all2
函式說明	取得刀具資料庫管理所有資料
C#方法	short GET_toolmanage_all2(StructMsg.Pwd P, ref StructMsg.Toolmanage_Single2[] R);
VB.Net 方法	

資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Toolmanage_Single2			
	變數名稱	型別	[in/out]	說明
	ToolNo	int	out	刀號
	ToolType	string	out	刀型式
	CycleTime	double	out	加工時間
	LifeTime	double	out	使用壽命
	PartCount	int	out	工件數(已使用)
	LifePartCount	int	out	工件數(壽命)
	FeedRate	int	out	Feed Rate
	SpindleSpeed	int	out	主軸轉速
	WorkMaterial	string	out	工件材質
	ToolMaterial	string	out	刀具材質
	Diameter	double	out	刀徑
ToolLength	double	out	刀長	
Supplier	string	out	供應商	
Contact	string	out	聯絡人	
Telephone	string	out	電話	
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Single2[] _AllData; private void button1_Click(object sender, EventArgs e) { ret = iRemoting.GET_toolmanage_all2(_Pwd, ref _AllData); if (ret == 0) { for (int i = 0; i <= _AllData.Length - 1; i++) { ... } } }</pre>			

4. GET_toolmanage_all_execute：取得刀具資料庫管理的所有時間及工件數(加工時間、壽命時間、加工工件數及壽命工件數)

API 名稱	GET_toolmanage_all_execute
函式說明	取得刀具資料庫管理的所有時間(加工時間、壽命時間、加工工件數及壽命工件數)

C#方法	short GET_toolmanage_all_execute(StructMsg.Pwd P, ref StructMsg.Toolmanage_Single_Execute[] R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Toolmanage_Single_Execute			
	變數名稱	型別	[in/out]	說明
	ToolNo	int[]	out	刀號
	CycleTime	double[]	out	加工時間
	LifeTime	double[]	out	壽命時間
	PartCount	int	out	加工工件數(已使用)
	LifePartCount	int	out	工件數(壽命)
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Single_Execute[] _AllExecute; private void button1_Click(object sender, EventArgs e) { short ret = iRemoting.GET_toolmanage_all_execute(_Pwd, ref _AllExecute); if (ret == 0) { ... } }</pre>			

5. GET_toolmanage_single2：取得單筆的刀具資料庫管理資料

API 名稱	GET_toolmanage_single2			
函式說明	取得單筆的刀具資料庫管理資料			
C#方法	short GET_toolmanage_single2(StructMsg.Pwd P, ref StructMsg.Toolmanage_Single2 R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Toolmanage_Single2			
	變數名稱	型別	[in/out]	說明
	ToolNo	int	int	刀號
	ToolType	string	out	刀型式
	CycleTime	double	out	加工時間
	LifeTime	double	out	使用壽命

	PartCount	int	out	加工工件數(已使用)
	LifePartCount	int	out	工件數(壽命)
	FeedRate	int	out	Feed Rate
	SpindleSpeed	int	out	主軸轉速
	WorkMaterial	string	out	工件材質
	ToolMaterial	string	out	刀具材質
	Diameter	double	out	刀徑
	ToolLength	double	out	刀長
	Supplier	string	out	供應商
	Contact	string	out	聯絡人
	Telephone	string	out	電話
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Single2 _Single; private void button1_Click(object sender, EventArgs e) { _Single.ToolNo = int.Parse(textBox1.Text); short ret = iRemoting.GET_toolmanage_single2(_Pwd, ref _Single); if (ret == 0) { ... } }</pre>			

6. GET_toolmanage_single_execute：取得單筆刀具資料庫管理的時間及工件數資料(加工時間、壽命時間、加工工件數及壽命工件數)

API 名稱	GET_toolmanage_single_execute			
函式說明	取得單筆刀具資料庫管理的時間資料(加工時間、壽命時間、加工工件數及壽命工件數)			
C#方法	short GET_toolmanage_single_execute(StructMsg.Pwd P, ref StructMsg.Toolmanage_Single_Execute R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。			
	結構名稱：Toolmanage_Single_Execute			
	變數名稱	型別	[in/out]	說明
	ToolNo	int	in	刀號
	CycleTime	double	out	加工時間

	LifeTime	double	out	壽命時間
	PartCount	int	out	加工工件數(已使用)
	LifePartCount	int	out	工件數(壽命)
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Single_Execute _Execute; private void button1_Click(object sender, EventArgs e) { _Execute.ToolNo = int.Parse(textBox1.Text); short ret = iRemoting.GET_toolmanage_single_execute(_Pwd, ref _Execute); if (ret == 0) { ... } }</pre>			

7. ADD_toolmanage_single2：新增單筆刀具資料庫管理資料

API 名稱	ADD_toolmanage_single2			
函式說明	新增單筆刀具資料庫管理資料			
C#方法	short ADD_toolmanage_single2(StructMsg.Pwd P, StructMsg.Toolmanage_Single2 R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Toolmanage_Single2			
	變數名稱	型別	[in/out]	說明
	ToolNo	int	in	刀號
	ToolType	string	in	刀型式
	CycleTime	double	in	加工時間
	LifeTime	double	in	使用壽命
	PartCount	int	in	加工工件數(已使用)
	LifePartCount	int	in	工件數(壽命)
	FeedRate	int	in	Feed Rate
	SpindleSpeed	int	in	主軸轉速
	WorkMaterial	string	in	工件材質
	ToolMaterial	string	in	刀具材質
	Diameter	double	in	刀徑

	ToolLength	double	in	刀長
	Supplier	string	in	供應商
	Contact	string	in	聯絡人
	Telephone	string	in	電話
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。			
適用範圍	所有控制器。			
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Single2 _Single; private void button1_Click(object sender, EventArgs e) { _Single.ToolNo = 2; _Single.ToolType = "TEST"; _Single.CycleTime = 0; _Single.LifeTime = 100; _Single.PartCount = 0; _Single.LifePartCount = 10; _Single.FeedRate = 1000; _Single.SpindleSpeed = 1000; _Single.WorkMaterial = "TEST"; _Single.ToolMaterial = "TEST"; _Single.Diameter = 10; _Single.ToolLength = 10; _Single.Supplier = "TEST"; _Single.Contact = "XXX"; _Single.Telephone = "04-23599009"; short ret = iRemoting.ADD_toolmanage_single2(_Pwd, _Single); MessageBox.Show(ret.ToString()); }</pre>			

8. SET_toolmanage_single2：修改單筆刀具資料庫管理的資料

API 名稱	SET_toolmanage_single2			
函式說明	修改單筆刀具資料庫管理的資料			
C#方法	short SET_toolmanage_single2(StructMsg.Pwd P, StructMsg.Toolmanage_Single2 R);			
VB.Net 方法				
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Toolmanage_Single2			
	變數名稱	型別	[in/out]	說明

	Original_ToolNo	int	in	原始尚未修改的刀號
	ToolNo	int	in	刀號
	ToolType	string	in	刀型式
	LifeTime	double	in	使用壽命
	LifePartCount	int	in	工件數(壽命)
	FeedRate	int	in	Feed Rate
	SpindleSpeed	int	in	主軸轉速
	WorkMaterial	string	in	工件材質
	ToolMaterial	string	in	刀具材質
	Diameter	double	in	刀徑
	ToolLength	double	in	刀長
	Supplier	string	in	供應商
	Contact	string	in	聯絡人
	Telephone	string	in	電話
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。 ※ 加工時間及工件數無法修改。			
適用範圍	所有控制器。			
C#程式範例	<pre> StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Single2 _Single; private void button1_Click(object sender, EventArgs e) { _Single.Original_ToolNo = 2; _Single.ToolNo = 2; _Single.ToolType = "TEST"; _Single.LifeTime = 200; _Single.LifePartCount = 20; _Single.FeedRate = 1000; _Single.SpindleSpeed = 1000; _Single.WorkMaterial = "TEST"; _Single.ToolMaterial = "TEST"; _Single.Diameter = 10; _Single.ToolLength = 10; _Single.Supplier = "TEST"; _Single.Contact = "XXX"; _Single.Telephone = "04-23599009"; short ret = iRemoting.SET_toolmanage_single2(_Pwd, _Single); MessageBox.Show(ret.ToString()); } </pre>			

9. DEL_toolmanage_single：刪除單筆刀具資料庫管理的資料

API 名稱	DEL_toolmanage_single		
函式說明	刪除單筆刀具資料庫管理的資料		
C#方法	short DEL_toolmanage_single(StructMsg.Pwd P, StructMsg.Toolmanage_ToolNo R)		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：Toolmanage_ToolNo		
	變數名稱	型別	[in/out] 說明
No	int	in	刀號
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。		
適用範圍	所有控制器。		
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_ToolNo _Toolmanage_ToolNo; private void button1_Click(object sender, EventArgs e) { _Toolmanage_ToolNo.No = 1; ret = iRemoting.DEL_toolmanage_single(_Pwd, _Toolmanage_ToolNo); if (ret != 0) { MessageBox.Show("刪除失敗!"); } }</pre>		

10. SET_toolmanage_single_cycletime：設定單筆刀具資料庫管理的加工時間

API 名稱	SET_toolmanage_single_cycletime		
函式說明	設定單筆刀具資料庫管理的加工時間		
C#方法	short SET_toolmanage_single_cycletime(StructMsg.Pwd P, StructMsg.Toolmanage_Single_CycleTime R)		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：Toolmanage_Single_CycleTime		
	變數名稱	型別	[in/out] 說明
ToolNo	int	in	刀號
CycleTime	Double	in	加工時間
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。		

適用範圍	所有控制器。
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Single_CycleTime _Toolmanage_Single_CycleTime; private void button1_Click(object sender, EventArgs e) { _Toolmanage_Single_CycleTime.ToolNo = 1; short ret = iRemoting.SET_toolmanage_single_cycletime(_Pwd, _Toolmanage_Single_CycleTime); if (ret != 0) { MessageBox.Show("設定失敗!"); } }</pre>

11. SET_toolmanage_single_partcount：設定單筆刀具資料庫管理的加工工件數

API 名稱	SET_toolmanage_single_partcount		
函式說明	設定單筆刀具資料庫管理的工件數		
C#方法	short SET_toolmanage_single_partcount(StructMsg.Pwd P, StructMsg.Toolmanage_Single_PartCount R);		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。 結構名稱：Toolmanage_Single_PartCount		
	變數名稱	型別	[in/out] 說明
	ToolNo	int	in 刀號
	PartCount	Int	in 加工工件數
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。		
適用範圍	所有控制器。		
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_Single_PartCount _PartCount; private void button1_Click(object sender, EventArgs e) { _PartCount.ToolNo = 2; _PartCount.PartCount = 30; short ret = iRemoting.SET_toolmanage_single_partcount(_Pwd, _PartCount); MessageBox.Show(ret.ToString()); }</pre>		

12. SET_toolmanage_single_zero：將單筆刀具資料庫管理的加工時間歸零

API 名稱	SET_toolmanage_single_zero		
函式說明	將單筆刀具資料庫管理的加工時間歸零		
C#方法	short SET_toolmanage_single_zero(StructMsg.Pwd P, StructMsg.Toolmanage_ToolNo R)		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：Toolmanage_ToolNo		
	變數名稱	型別	[in/out] 說明
No	int	in	刀號
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。		
適用範圍	所有控制器。		
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_ToolNo _Toolmanage_ToolNo; private void button1_Click(object sender, EventArgs e) { _Toolmanage_ToolNo.No = 1; short ret = iRemoting.SET_toolmanage_single_zero(_Pwd, _Toolmanage_ToolNo); if (ret != 0) MessageBox.Show("歸零失敗!"); }</pre>		

13. SET_toolmanage_single_zero_partcount：將單筆刀具資料庫管理的加工工件數歸零

API 名稱	SET_toolmanage_single_zero_partcount		
函式說明	將單筆刀具資料庫管理的加工工件數歸零		
C#方法	short SET_toolmanage_single_zero_partcount(StructMsg.Pwd P, StructMsg.Toolmanage_ToolNo R);		
VB.Net 方法			
資料結構	結構名稱：Pwd，連線密碼及寫入密碼設置(請參考第三章)。		
	結構名稱：Toolmanage_ToolNo		
	變數名稱	型別	[in/out] 說明
No	int	in	刀號
備註	※ 請連接 Port 9501 傳送及擷取刀具管理資訊。		
適用範圍	所有控制器。		
C#程式範例	<pre>StructMsg.Pwd _Pwd; StructMsg.Toolmanage_ToolNo _ToolNo; private void button1_Click(object sender, EventArgs e)</pre>		

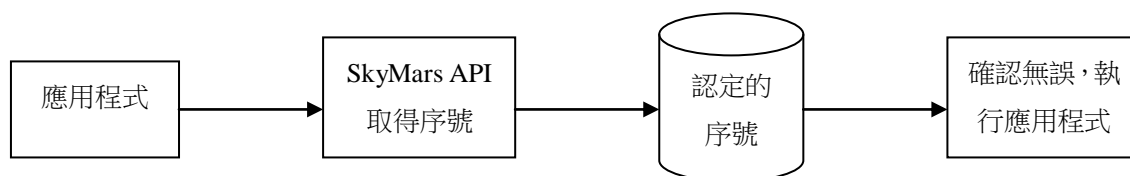

```
{
    _ToolNo.No = 2;
    short ret = iRemoting.SET_toolmanage_single_zero(_Pwd,
                                                    _ToolNo);
    MessageBox.Show(ret.ToString());
}
```

二十四、 應用程式開發者軟體防盜版策略

SkyMars Professional 提供兩種的軟體防盜版策略，允許程式設計師視需要選取最適當應用程式保護技術。不同層級的軟體防盜版策略的優點與花費的成本並不相同，所以我們必須了解各種不同的軟體防盜版策略的成本及其好處，以便為所開發的應用程式選擇最適合的保護機制。

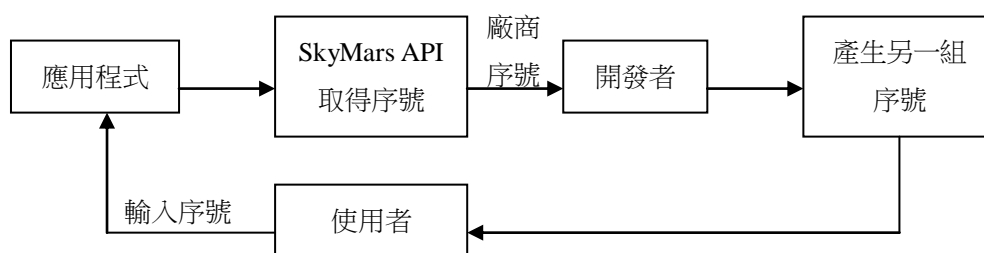
SkyMars Professional 執行時皆必須使用 USB Key，即為 USB 硬體鎖，此 USB 硬體鎖內含有兩個資訊，一個為序號(ID)及授權的連線機台數，可以使用 SKY_version 函式來取得，並且該把硬體鎖是無法複製及寫入。每一把的硬體鎖序號，都會不一致，開發者可以利用這組序號進行驗證及鎖定。以下介紹兩種軟體防盜版策略：

1. 依序號(USB Key ID)驗證



由於每一把 USB 硬體鎖都有唯一的序號，開發者可以事先得知，並寫入應用程式碼內或資料庫(需編碼)。應用程式執行後，可以藉由 SkyMars API 的 SKY_version 函式來取得序號，並確認此序號為合法，應用程式才可以執行。

2. 將序號(USB Key ID)回傳給開發者產生另一組序號驗證



第一次執行應用程式時，可以藉由 SkyMars API 的 SKY_version 函式來取得序號，並將序號及廠商資訊回傳給開發者，可以使用 e-mail 或開發者架設註冊伺服器等，當開發者收到資訊後，開發者可進行認定該廠商是為合法使用者，如果確認為合法使用者，開發者可以產生另一組序號(自定編碼規則或採用加密法則)，並傳送給使用者輸入序號，應用程式依據序號反推(自定編碼規則或採用加密法則)回 USB Key ID，並與 USB 硬體鎖的 ID 進行確認，若符合應用程式才可以執行。

二十五、 問題排除

1. 問題：應用程式呼叫使用 SkyMars API 時，造成程式發生錯誤，此錯誤訊息為「無法解析至正確方法的引動過程。」

解決：請更換最新的 InterfaceLib.dll 即可。