

# API 使用说明书 (.Net Remoting)

(对应 SkyMars Professional / Express 3.06 以上)

# Version 3.06 2014/9/28

# 目录

<u> </u>		SkyMars Professional 架构图	7
<u> </u>		开发者撰写程序流程	8
Ξ,		API 联机及写入密码	12
四、		回传错误码说明	13
五、		FTP 回传错误列表	15
六、		内部信息类别函式	17
	1.	SKY_version: 取得 SkyMars 的版本及 USB Key 信息	17
	2.	SKY_conn_status: 取得 CNC 联机状态	17
	3.	SKY_conn_ip_port: 取得 CNC 联机 IP 及 Port 及其他相关信息	18
	4.	SKY_nc_filename: 取得 CNC NC 檔名	19
	5.	SKY_conn_count: 取得 SkyMars 的已经连接的机台数量	20
と、		基本类别函式	21
	1.	GET_information: 基本相关信息	21
	2.	GET_information_heid: 海德汉控制器_基本相关信息	21
	3.	GET_status: 状态信息	23
	4.	GET_position:坐标信息	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	le)25
	7.	GET_feed_spindle: 进给率/转速	26
	8.	GET_time: 时间	27
	9.	GET_time_heid: 海德汉控制器_时间	28
	10.	GET_time_cnc: 取得控制器系统时间	30
	11.	GET_part_count: 工件数	30
	12.	GET_part_total:	31
	13.	GET_part_required:	32
	14.	SET_relpos: 设定相对坐标数值	33
	15.	SET_time_cnc: 设定控制器的时间	34
八、		警报类别函式	35
	1.	GET_alm_current: 目前发生的警报	35
	2.	GET_alm_current2: 目前发生的警报	35
	3.	GET_alm_current_heid: 海德汉控制器_目前发生的警报	37
	4.	GET_alm_history: 警报履历	38
	5.	GET_alm_history2: 警报履历	38
	6.	GET_alm_history_heid: 警报履历_海德汉控制器	40
	7.	GET_msg_current: 取得 Operation 讯息	41
	8.	GET_msg_history: 取得 Operation 履历	42
	9.	GET_plc_alarm: 取得 PLC Alarm 履历	43
九、		伺服主轴类别函式	44

	1.	GET_servo_current: 各轴负载电流值	44
	2.	GET_servo_load: 取得伺服负载%	44
	3.	GET_servo_speed: 取得伺服轴转速	45
	4.	GET_spindle_load: 取得主轴负载%	46
	5.	GET_spindle_speed: 取得主轴转速	47
	6.	GET_servo_temperature: 取得伺服马达温度	48
	7.	GET_spindle_temperature: 取得主轴温度	48
+	•	刀具管理类别函式 (控制器)	50
	1.	GET_offset_title: 取得刀具 Offset 的标题栏	50
	2.	GET_offset_all: 取得所有刀具 Offset	51
	3.	GET_offset_scope: 依指定范围取得刀具 Offset	52
	4.	GET_offset_single: 取得单笔刀具 Offset	53
	5.	SET_offset_all: 写入所有刀具 Offset	54
	6.	SET_offset_single: 写入单笔刀具 Offset	55
	7.	GET_offset_count: 取得刀具 Offset 笔数	56
	8.	GET_tool_title: 取得刀具管理的标题栏	57
	9.	GET_tool_count: 取得刀具 Offset 笔数	58
	10.	GET_tool_data: 取得所有刀具管理数据	58
	11.	GET_tool_data_scope: 取刀具管理数据,并依指定索引值范围传回	60
	12.	SET_tool_data: 写入刀具管理	61
	13.	GET_pocket_title:取得刀库表的标题栏	62
	14.	GET_pocket_count: 取得刀库表笔数	62
	15.	GET_pocket_data: 取得刀库表资料	63
	16.	SET_pocket_data: 写入刀库表	64
	17.	GET_pocket_data_scope: 取刀库表数据,并依指定索引值范围传回	65
+-	- <b>、</b>	工件坐标类别函式	67
	1.	GET_work_coord_title: 取得工件坐标的标题栏	67
	2.	GET_work_coord_all: 取得所有工件坐标数据	67
	3.	GET_work_coord_scope: 依指定范围取得工件坐标数据	68
	4.	GET_work_coord_single: 取得单笔工件坐标资料	70
	5.	SET_work_coord_all: 写入所有工件坐标数据	71
	6.	SET_work_coord_single: 写入单笔工件坐标数据	72
	7.	GET_work_coord_count: 取得工件坐标笔数	73
	8.	GET_preset_title: 取得工作台管理表的标题栏	74
	9.	GET_preset_count: 取得工作台管理表笔数	75
	10.	GET_preset_data: 取得工作台管理表数据	76
	11.	SET_preset_data: 写入工作台管理表	77
	12.	GET_preset_data_scope: 取得工作台管理表,并依指定索引值范围传回	78
+-	_``	Macro 变量类别函式(共通变量)	80
	1.	GET_macro_all: 取得所有 Macro(共通变量)变量数据	80

2.	GET_macro_scope: 依指定范围取得 Macro(共通变量)变量数据	81
3.	SET_macro_all: 写入所有 Macro(共通变量)变量数据	82
4.	GET_macro_single: 取得单笔 Macro(共通变量)变量数据	83
5.	SET_macro_single: 写入单笔 Macro(共通变量)变量数据	84
6.	GET_macro_variable: 取得 Macro(共通变量)变量型态	85
7.	GET_pcode_single: 取得单笔 P Code Macro 资料(FANUC)	86
8.	SET_pcode_single: 写入单笔 P Code Macro 资料(FANUC)	86
十三、	加工程序类别函式	88
1.	GET_nc_mem_list: 取得机台内部加工程序列表(内存)	88
2.	GET_nc_ftp_list: 取得 FTP 卡加工程序列表(Fanuc DataServer/三菱 CF 卡)	89
3.	GET_nc_mem_code: 取得内存内的加工程序内容	90
4.	GET_nc_ftp_code: 取得 FTP 加工程序内容(Fanuc DataServer/三菱 CF 卡)	91
5.	UPLOAD_nc_mem: 上传加工程序至内存	92
6.	UPLOAD_nc_ftp: 上传加工程序至 FTP(Fanuc DataServer/三菱 CF 卡)	93
7.	DEL_nc_mem: 删除加工程序(内存)	94
8.	DEL_nc_ftp: 删除加工程序(FTP) (Fanuc DataServer/三菱 CF 卡)	95
9.	GET_nc_pointer: 取得程序目前的指针(行)	96
10.	GET_nc_current_block: 取得目前程序正在执行及未来要执行的单节	97
11.	GET_nc_freespace: 取得剩余空间	98
12.	SET_nc_main: 设定或指定加工程序为主程序	99
13.	UPLOAD_nc_mdi: 上传加工程序代码到控制器(MDI 模式)	99
十四、	系统参数类别函式	101
1.	GET_param_max: 取得参数最大号码(或最后一个号码)	101
2.	GET_param_data: 取得参数资料	101
3.	SET_param_data: 设定参数数据	104
十五、	PLC 类别函式	107
1.	GET_plc_ver: 取得 PLC 版本信息	107
2.	GET_plc_addr: 取得 PLC 地址的起始及结束号码	107
3.	GET_plc_addr2: 取得 PLC 地址的起始及结束号码	109
4.	GET_plc_status: 取得机台 PLC 状态信息	110
5.	GET_plc_status2: 取得机台 PLC 状态信息	111
6.	SET_plc_status: 写入机台 PLC	113
7.	SET_plc_status2: 写入机台 PLC	114
十六、	档案管理类别函式	117
1.	GET_file_directory: 取得控制上的档案及目录	117
2.	UPLOAD_file2:从 PC 端上传档案到装置(如 CNC 控制器)	118
3.	DOWNLOAD_file2:从装置上(如 CNC 控制器)下载文件到 PC 端	118
4.	<b>DEL_file:</b> 从装置上删除档案	119
十七、	National Instruments 9233 加速规模组类别 (暂不开放)	120
1.	START_ni9233_get_data: 设定启动模块进行数据撷取	120

2.	STOP_ni9233_get_data: 停止撷取	121
3.	CHECK_ni9233_running: 确认是否已启动资料撷取	121
4.	GET_ni9233_spectrum_data: 取得资料	122
5.	GET_ni9233_error_msg: 取得模块错误讯息	123
十八、	RS232 模块类别	124
1.	GET_rs232_status: 取得目前 RS232 的相关状态及参数	124
2.	OPEN_rs232: 开启 RS232	125
3.	SET_rs232: 将指定的数据写入 RS232	126
4.	CHECK_rs232_data_received: 确认是否已经接收到资料(IsOnCallbackEvent=tru	e)
	127	
5.	GET_rs232_data_received: 取得 RS232 所传回的资料	127
6.	CLOSE_rs232: 关闭 RS232	128
7.	GET_rs232_error: 取得 RS-232 例外错误讯息	128
十九、	PC Camera 模块类别(USB)	130
1.	CONNECT_cam: 联机摄影机	130
2.	DISCONNECT_cam: 关闭摄影机	130
3.	CHECK_cam_running: 确认摄影机是否已连接及撷取	131
4.	GET_cam_image: 取得摄影机影像	131
5.	GET_cam_error: 取得摄影机例外错误讯息	132
$\underline{-+}$	SkyMars Professional 事件派送	134
1.	EVENT_conn_cnc: 产生 SkyMars Professional 设定 CNC 信息完成事件	135
2.	EVENT_disconn_cnc: 产生 SkyMars Professional 结束 CNC 联机事件	136
3.	EVENT_conn_status: 产生 CNC 状态事件(OFF、RUN、IDLE、ALARM)	137
4.	EVENT_nc_filename: 产生 CNC 切换 NC 檔名事件	138
<u> </u>	、 SkyMars 稼动率信息	140
1.	GET_utilization_all_today: 取得今日机台稼动率信息	140
2.	GET_utilization_single_time: 取得单一机台总时间之稼动率信息	140
3.	GET_utilization_single_total: 取得单一机台总次数(工件数、加工循环次数)之稼	动率
信	息	142
4.	GET_utilization_single_list: 取得单一机台稼动率履历信息	143
5.	GET_utilization_single_all: 取得单一机台稼动率信息及履历	144
二十二	、 SkyMars 维护管理类别	147
1.	GET_maint_count: 取得维护管理的总笔数	147
2.	GET_maint_title: 取得维护管理的标题栏文字	147
3.	GET_maint_all: 取得维护管理所有数据	148
4.	GET_maint_all_time: 取得维护管理项目内的所有时间(使用时间、寿命时间)	149
5.	GET_maint_single: 取得单笔的维护管理的项目数据	149
6.	GET_maint_single_time: 取得单笔维护管理的时间数据(使用时间、寿命时间).	150
7.	ADD_maint_single: 新增单笔维护管理数据	151
8.	SET_maint_single: 修改单笔维护管理的数据	152

	9.	DEL_maint_single: 删除单笔维护管理的数据153
	10.	SET_maint_single_usetime: 设定单笔维护管理的使用时间154
	11.	SET_maint_single_zero: 将单笔维护管理的使用时间归零154
<u> </u>	- <u>=</u> ,	SkyMars 刀具数据库管理类别156
	1.	GET_toolmanage_count: 取得刀具数据库管理的总笔数156
	2.	GET_toolmanage_title2: 取得刀具数据库管理的标题栏文字156
	3.	GET_toolmanage_all2: 取得刀具数据库管理所有数据157
	4.	GET_toolmanage_all_execute: 取得刀具数据库管理的所有时间及工件数(加工时间、
	寿命	时间、加工工件数及寿命工件数)158
	5.	GET_toolmanage_single2: 取得单笔的刀具数据库管理数据159
	6.	GET_toolmanage_single_execute: 取得单笔刀具数据库管理的时间及工件数数据(加
	工时	计间、寿命时间、加工工件数及寿命工件数)160
	7.	ADD_toolmanage_single2: 新增单笔刀具数据库管理数据161
	8.	SET_toolmanage_single2: 修改单笔刀具数据库管理的数据162
	9.	DEL_toolmanage_single: 删除单笔刀具数据库管理的数据164
	10.	SET_toolmanage_single_cycletime:设定单笔刀具数据库管理的加工时间164
	11.	SET_toolmanage_single_partcount:设定单笔刀具数据库管理的加工工件数.165
	12.	SET_toolmanage_single_zero:将单笔刀具数据库管理的加工时间归零166
	13.	SET_toolmanage_single_zero_partcount:将单笔刀具数据库管理的加工工件数归
	零	166
<u> </u>	一四、	应用程序开发者软件防盗版策略168
<u> </u>	五、	问题排除169

#### 一、SkyMars Professional 架构图

SkyMars Professional 架构层级由 4 个部份组成,如下图所示:

Cloud	UI		Plug	-In
	SkyMars I	Runtime		
Hardward				
CNC Co	ntroller	Se	nsor	

#### 1. SkyMars Runtime

此层级负责与硬件端 CNC 各型控制器或其他传感器来连结。

#### 2. User Interface (UI)

SkyMars Professional 提供标准的界面(UI)供使用者操作,使用者进行任何操作时,都会与 SkyMars Runtime 层级进行互动,此层级的界面含有信息监视、程序上下载、参数信息、伺服主轴监视、刀具管理及维护等。

#### 3. Cloud

Cloud 模块会与云端服务器链接,由 SkyMars Runtime 层级视需要呼叫,如启动家 动率服务后,SkyMars Runtime 会将 CNC 基本信息、状态,藉由 Cloud 模块传送至 云端服务器。

#### 4. Plug-In Interface

Plug-In Interface 为 Interface DLL 檔, Interface 内定义函式名称及结构变量, 开发者 必须加入参考, 并建立命名空间, 才可以使用 Interface 内所定义的函式及结构变量 来呼叫使用。

#### 二、开发者撰写程序流程

## 1. 新增专案

您可以使用微软开发工具 Visual Studio 2005 / 2008 / 2010 / 2012 开发,在开发环境上 将拥有完全的支持,这对我们撰写程序代码有相当大的帮助。我们新增一个项目, 此项目选择 Visual C#或 Visual Basic,并选择相对应的开发项目及指定项目名称后, 按下「确定」开发环境将会为我们建立此项目。

新増専案							?	x
最近使用的範本		.NET Fra	amework 4 🛛 排序依据	<b>彖:</b> 預設		▼ Ⅲ Ⅲ 搜尋 E	己安裝的範本	ρ
已安裝的範本			Windows Form 應用把式	Vieual C#		Visual C#		
▲ Visual C#			Windows Form 應用准式	Visual C#	建立	具有 Windows Form	使用者介面的	匀應
Windows Web		<b>S</b>	WPF 應用程式	Visual C#	用程3	다.콩.꽃		
Cloud Reporting		с\ <b>ЕС</b>	主控台應用程式	Visual C#				
Silverlight WCF		c#	類別庫	Visual C#	=			
Workflow 測試		<sup>∞</sup> c≇	WPF 瀏覽器應用程式	Visual C#				
▷ 其他語言 ▷ 其他専案類型		_ <b>c</b> #	Windows 服務	Visual C#				
▷ 資料庫 ▷ 測試専案		<b>€C</b>	WPF 自訂控制項程式庫	Visual C#				
線上範本		<b>₹C</b>	WPF 使用者控制項程式庫	Visual C#				
		C♯	空專案	Visual C#	Ŧ			
名稱( <u>N</u> ):	WinGetPosition							
位置(山):	F:\temp\201212	20		•	瀏覽(8	3)		
方案( <u>S</u> ):	建立新方案			•				
方案名稱( <u>M</u> ):	WinGetPosition				□ 為方案	健立目錄( <u>D</u> )		
					□ 加入至	EI尿姑僧控制(U)	_	
						確定	取消	

## 2. 加入参考 System.Runtime.Remoting 及 InterfaceLib.dll

在撰写程序代码前,我们必须要加入两个参考,才能使用 SkyMars API 提供的函式; 请从「方案总管」-「参考」再点选右键「加入参考」。



点选「.NET」页次,在组件名称上找到 System.Runtime.Remoting,并点选「确定」加入参考。

筛選至: .NET Framework 3.5			
	版本	執行階段	路徑
System.Net	3.5.0.0	v2.0.50727	C:\Program Files\Refere
System.Printing	3.0.0.0	v2.0.50727	C:\Program Files\Refere
System.Runtime.Remoting	2.0.0.0	v2.0.50727	C:\Windows\Microsoft
System.Runtime.Serialization	3.0.0.0	v2.0.50727	C:\Program Files\Refere
System.Runtime.Serialization.Formatte.	2.0.0.0	v2.0.50727	C:\Windows\Microsoft
System.Security	2.0.0.0	v2.0.50727	C:\Windows\Microsoft
System.ServiceModel	3.0.0.0	v2.0.50727	C:\Program Files\Refere
System.ServiceModel.Web	3.5.0.0	v2.0.50727	C:\Program Files\Refere
System.ServiceProcess	2.0.0.0	v2.0.50727	C:\Windows\Microsoft
Contract Contracts	2000		CAD

接下来我们必须要再加入「InterfaceLib.dl1」;加入前,请先将该档案复制到目前建置项目的执行档底下(如:专案名称\bin\Debug)。依序在「方案总管」-「参考」 再点选右键「加入参考」。点选「浏览」页次,切换到 bin\Debug 的文件夹,将 InterfaceLib.dl1 点选,并按下「确定」加入参考。

.NET COM 専業 創見 最近使用的 搜尋位置①: <u>}</u> Debug	▼ G Ø ▷ Ⅲ▼	
名稱	修改日期	類型
S InterfaceLib.dll	2012/7/2 上午 11:42	應用程式擴充
Win_Info_Status.exe	2013/1/9 下午 12:39	應用程式
Win_Info_Status.vshost.exe	2013/1/9 下午 12:45	應用程式
Win_Info_Status.vshost.exe.manifest	2009/8/31 上午 12:40	MANIFEST檔案
< [		•
檔案名稱(M): InterfaceLib.dll		-
檔案類型(I): 元件檔 (*.dll;*.tb;*.olb;*.ocx;*.exe;*.ma	nifest)	•

3. 建立命名空间

请在程序代码检视, using的下面加入以下命名空间程序代码: using System.Runtime.Remoting; using System.Runtime.Remoting.Channels; using System.Runtime.Remoting.Channels.Tcp; using InterfaceLib; 可参考如下图:

1	Jusing System;
2	<pre>using System.Collections.Generic;</pre>
3	<pre>using System.ComponentModel;</pre>
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	<pre>using System.Runtime.Remoting.Channels;</pre>
12	<pre>using System.Runtime.Remoting.Channels.Tcp;</pre>
13	using InterfaceLib;

4. 建立 InterfaceLib.IMsg 型别的 iRemoting 变数 请将以下的程序代码,加入到适当的位置:

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

- 可自行定义 iRemoting 变量
- 5. 注册.Net Remoting 通道并系结

```
定义出来的 iRemoting 变量,无法直接使用,我们必须要系结到 SkyMars 的 Channel, 才能够使用 API 函式,请将以下程序代码加入到适当位置。
```

```
if (ChannelServices.RegisteredChannels.Length == 0)
```

```
{
```

```
ChannelServices.RegisterChannel(new TcpChannel());
```

```
}
```

- "tcp://localhost:9501/RemoteObjectURI9501":
   9501 代表 SkyMars 联机的第一台机台,第二台之后会依序号接续下去,如 第二台为 9502,第三台为 9503,以此类推。
- 采用 TCP 及二进制流传输联机,必须要注意防火墙是否阻挡。

```
6. 参考范例程序代码:
```

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
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";
           //取得厂牌及机台名称
           short ret = iRemoting.SKY_conn_ip_port(_Pwd, ref R1);
           if (ret == 0)
           {
              txtManufacturer.Text = R1.Manufacturer[0];
              txtName.Text = R1.MachineName[0];
           }
       }
    }
}
```

 如何得知 SkyMars 已联机的机台数目 您可以先联机到 SkyMars 的第一台机台,如预设的 9501,并使用 API 函式 SKY\_conn\_ip\_port,从数据结构里面,数组长度就是 SkyMars 已联机的机台数目。

#### 三、API 联机及写入密码

目前 SkyMars Professional /Express 3.0 之后的版本,所有函式全部加入「API 联机密码」及「API 写入密码」,主要目的在于提升安全性。在因特网(Internet)上确保 SkyMars 不会随意接受外部命令来更改或取得控制器信息。所有函式都会要求输入 Pwd 的数据结构,如下表所示:

结构名称: Pwd			
变量名称	型别	[in/out]	说明
ConnectionKey	string	in	API 联机密码
WritePwd	string	in	API 写入密码

「API 写入密码」意旨当外部命令透过函式要写入控制器时,该函式会要求输入「API 写入密码」, SkyMars 会进行比对,正确无误后,才会写入控制器。

设定方式:

1. 开启 SkyMars 的安全性设定

请开启 SkyMars Professional 或 SkyMars Express, [设定]→[API 安全性设定]。

021	
設定API寫入密碼:	
123	

- 2. 设定 API 联机密码及 API 写入密码。
- 函式传回值说明 传回值为 7: SkyMars API 被设定成写入保护。 传回值为 20: 密码不正确。
- 4. 程序代码范例:

```
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、回传错误码说明

传回值	错误类别	说明
-31	Camera Exception	Camera 发生内部例外错误。
-30	Camera Current Running	Camera 目前已连接及影像撷取中。
-29	Camera Disconnect	Camera 未连接。
-28	RS-232 Close	RS-232 Com Port 己关闭。
-27	RS-232 Over limit	RS-232 所能连接的模块,已超过上限。
-26	RS-232 Return Type	未指定 RS-232 传回的型别。
-25	RS-232 Open Failed	RS-232 COM Port 开启失败或已经开启。
-24	RS-232 Exception	RS-232 发生内部例外错误,请使用相对应的函式取得错误。
-23	Sensor Exception	传感器模块内部或 API 发生错误,请使用相对应的函式取得错
-23	Sensor Exception	误。
-22	Sensor Current Running	目前传感器正在执行中。
-21	USB key error	USB Key 失效(硬件锁失效)。
-20	Plug-In Time out	执行的函式已经 Time out,如果要加长,请直接修改 Register。
-19	SkyMars Busy	SkyMars 目前正在激活中。
-18	Not supported	控制器不支持此函式。
-17	Protocol error (Ethernet version only)	网络卡设定不正确。
-16	Socket error (Ethernet version only)	联机尝试失败 / 控制器拒绝联机。
-15	DLL file error	使用的 DLL 档,无法对应 CNC 的型号。或 DLL 档已经遗失。
-8	Handle number error	请取得 handle 值。
7	Version mismatch between the	CNC/PMC 版本无法对应 library. 更换 library 或更新 CNC/PMC
-7	CNC/PMC and library	控制软件。
-6	Abnormal library state	执行的 library 发生例外错误。
-3	Random key timeout	随机密钥过时失效,请重新撷取 Random key。
-2	Reset or stop request	Reset 或 Stop 按钮被按下。函数调用被中止。
-1	CNC Busy	目前 CNC 处于忙录状态,请稍后再测试。
0	Normal termination	正常执行,未发生错误。
1	Error(function is not executed, or not	当去地行特定承式 必须再先地行 不则达承式无法使用
	available)	四小12(1) 付疋图八, 20 次爭兀12(1)。
2	Error(data block length error, error of	请检查数据长度及数据数量。
3	Error(data number error)	法处本教报日初日不工体
-		用他亘剱掂亏旳定盲止佣。

4	Error(data attribute error)	请检查数据属性是否正确。
5	Error(data error)	写入的数据不正确。
6	Error(no option)	
7	Error(write protection)	写入保护。
8	Error(memory overflow)	内存溢位。
9	Error(CNC parameter error)	参数设定错误。
10	Error(buffer empty/full)	缓冲区属于 empty 或 full。
11	Error(path number error)	路径不正确。
12	Error(CNC mode error)	CNC 模式(Mode)不正确。
13	Error(CNC execution rejection)	CNC 拒绝执行。
14	Error(Data server error)	Data Server 发生错误。
15	Error(alarm)	警报处理发生错误。
16	Error(stop)	CNC 状态在 stop 或 emergency。
17	Error(State of data protection)	数据被 CNC 所被保护。
18	Error(Not found Machine ID)	找不到要联机的机台。
19	Error(No out)	请确认 NO。
20	Error(Password)	密码错误。
21	Error(Compatible)	兼容性问题。(无法兼容旧版 API)

# 五、FTP 回传错误列表

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
421	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	
131	Requested action aborted, local error in processing.
452	Requested action aborted, local error in processing.         Requested action not taken. Insufficient storage space in system.
452 500	Requested action aborted, local error in processing.         Requested action not taken. Insufficient storage space in system.         Syntax error, command unrecognized. This may include errors such as command line too long.
452 500 501	Requested action aborted, local error in processing.         Requested action not taken. Insufficient storage space in system.         Syntax error, command unrecognized. This may include errors such as command line too long.         Syntax error in parameters or arguments.
452 500 501 502	Requested action aborted, local error in processing.         Requested action not taken. Insufficient storage space in system.         Syntax error, command unrecognized. This may include errors such as command line too long.         Syntax error in parameters or arguments.         Command not implemented.
452 500 501 502 503	Requested action aborted, local error in processing.Requested action not taken. Insufficient storage space in system.Syntax error, command unrecognized. This may include errors such as command line too long.Syntax error in parameters or arguments.Command not implemented.Bad sequence of commands.
452 500 501 502 503 504	Requested action aborted, local error in processing.         Requested action not taken. Insufficient storage space in system.         Syntax error, command unrecognized. This may include errors such as command line too long.         Syntax error in parameters or arguments.         Command not implemented.         Bad sequence of commands.         Command not implemented for that parameter.
452 500 501 502 503 504 530	Requested action aborted, local error in processing.Requested action not taken. Insufficient storage space in system.Syntax error, command unrecognized. This may include errors such as command line too long.Syntax error in parameters or arguments.Command not implemented.Bad sequence of commands.Command not implemented for that parameter.User not logged in.
452 500 501 502 503 504 530 532	Requested action aborted, local error in processing.Requested action not taken. Insufficient storage space in system.Syntax error, command unrecognized. This may include errors such as command line too long.Syntax error in parameters or arguments.Command not implemented.Bad sequence of commands.Command not implemented for that parameter.User not logged in.Need account for storing files.
452 500 501 502 503 504 530 532 550	Requested action aborted, local error in processing.Requested action not taken. Insufficient storage space in system.Syntax error, command unrecognized. This may include errors such as command line too long.Syntax error in parameters or arguments.Command not implemented.Bad sequence of commands.Command not implemented for that parameter.User not logged in.Need account for storing files.Requested action not taken. File unavailable (e.g., file not found, no access).

553 Requested action not taken. Illegal file name.

## 六、内部信息类别函式

# 1. SKY\_version: 取得 SkyMars 的版本及 USB Key 信息

API 名称	SKY_Version				
函式说明	取得 SkyMars 版本及 USB Key 信息				
C#方法	short SKY_versio	n(StructMsg	g.Pwd P, re	f StructMsg.SkyVersion R);	
VB.Net 方法	SKY_version(ByVa	l P As Stru	.ctMsg.Pwd	ByRef R As	
	StructMsg.SkyVer	sion) As Sh	nort		
数据结构	结构名称: Pwd,联	机密码及写入	密码设置 <b>(</b> 请	参考第三章)。	
	结构名称: SkyVers	ion			
	变量名称	型别	[in/out]	说明	
	Version	float	out	SkyMars 版本	
	SerialNumber	long	out	USB key 的序号	1
	LicenseCount	short	out	可联机机台数	
备注	SerialNumber 为硬	件 USB Key	ID,毎一支し	JSB Key 都会有不一样的 ID	
适用范围	不限控制器。				
C#程序范例	StructMsg.SkyVer	sion _SkyVe	ersion;		
	StructMsg.Pwd _P	wd;			
	private void but	ton1_Click(	object ser	nder, EventArgs e)	
	{				
	_Pwd.Connect:	ionKey = "1	23";		
	short ret = :	iRemoting.S	KY_version	(_Pwd, ref _SkyVersion);	
	if (ret == 0)	)			
	{				
	txtVersic	on.Text = _	SkyVersion	.Version.ToString();	
	txtKeyId.	Text = _Sky	Version.Se	erialNumber.ToString();	
	txtLic.Te	ext = _SkyVe	ersion.Lice	ense.ToString();	
	}				
	}				

2. SKY\_conn\_status: 取得 CNC 联机状态

API 名称	SKY_conn_status
函式说明	取得 CNC 联机状态
C#方法	<pre>short SKY_conn_status(StructMsg.Pwd P, ref StructMsg.SkyConn_status</pre>
	R);
VB.Net 方法	SKY_conn_status(ByVal P As StructMsg.Pwd, ByRef R As
	StructMsg.SkyConn_status) As Short

数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。					
	结构名称: SkyConn_status					
	变量名称	型别	[in/out]	说明		
	MachineNo	int[]	out	机器的索引值		
	Status	short[]	out	0:未开机 1:RUN 2:IDLE		
				3:Alarm		
				4:同时发生 RUN 及 ALARM		
备注						
适用范围	不限控制器。					
C#程序范例	StructMsg.SkyConn	_status _Sky	yConn_statu	ls;		
	StructMsg.Pwd _Pwo	d;				
	private void butto	on2_Click(ob	oject sende	er, EventArgs e)		
	{					
	_Pwd.Connectio	onKey = "123	;";			
	short ret = iR	emoting.SKY	_conn_stat	us(_Pwd, ref		
	_SkyConn_status);					
	if (ret == 0)					
	{					
	<pre>txtMachineNo.Text =</pre>					
	_SkyConn_status.MachineNo[0].ToString();					
	txtStatus.	Text = _Sky	Conn_statu	s.Status[0].ToString();		
	}					
	}					

# 3. SKY\_conn\_ip\_port: 取得 CNC 联机 IP 及 Port 及其他相关信息

API 名称	SKY_conn_ip_port						
函式说明	取得 CNC 联机 IP 及 Port 及其他相关信息						
C#方法	<pre>short SKY_conn_ip_port(StructMsg.Pwd P, ref</pre>						
	<pre>StructMsg.SkyConn_ip_port R);</pre>						
<b>VB.Net</b> 方法	SKY_conn_ip_port(ByVal P As StructMsg.Pwd, ByRef R As						
	StructMsg.SkyConn_ip_port) As Short						
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。						
	结构名称: SkyConn_ip_port						
	变量名称         型别         [in/out]         说明						
	MachineNo int[] out 机器的索引值						
	MachineName string[] out 机台名称						
	IP   string[]   out   机台 IP 地址						
	Port	short[]	out	机台 Port 号码			
	Manufacturer	<pre>string[]</pre>	out	厂牌			

	PlugInPort	int[]	out	Plug-In Port 号码			
备注							
适用范围	不限控制器。						
C#程序范例	StructMsg.SkyConn_	ip_port _Sky	Conn_ip_po	rt;			
	StructMsg.Pwd _Pwd	;					
	private void butto	n3_Click(obj	ect sender	, EventArgs e)			
	{						
	_Pwd.Connectior	nKey = "123";	;				
	short ret = iRe	<pre>short ret = iRemoting.SKY_conn_ip_port(_Pwd, ref</pre>					
	_SkyConn_ip_port);						
	if (ret == 0)						
	{						
	txtMachineN	o2.Text =					
	_SkyConn_ip_port.M	achineNo[0].	ToString()	;			
	txtMachineN	ame.Text = _	_SkyConn_ip	_port.MachineName[0];			
	txtIP.Text	= _SkyConn_i	.p_port.IP[	0];			
	txtPort.Tex	t = _SkyConn	_ip_port.P	<pre>Port[0].ToString();</pre>			
	txtManufact	urer.Text =	_SkyConn_i	.p_port.Manufacturer[0];			
	<pre>txtPlugInPort.Text =</pre>						
	_SkyConn_ip_port.P	_SkyConn_ip_port.PlugInPort[0].ToString();					
	}						
	}						

# 4. SKY\_nc\_filename: 取得 CNC NC 檔名

API 名称	SKY_nc_filename						
函式说明	取得 CNC NC 檔名						
C#方法	short SKY_nc_f	ilename(Str	uctMsg.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						
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。						
	结构名称: SkyNo	c_filename					
	变量名称	变量名称         型别         [in/out]         说明					
	RetType	short	in	指定			
				-1:All ,Other:Machine Index			
	MachineNo	int[]	out	机器的索引值			
	MainProg	string[]	out	主程序文件名			
	SubProg	string[]	out	子程序文件名			
备注		·					

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

5. \_SKY\_conn\_count: 取得 SkyMars 的已经连接的机台数量

API 名称	SKY_conn_count					
函式说明	取得 SkyMars 的已经连接的机台数量					
C#方法	short SKY_conn_co	ount(Struct	Msg.Pwd P,	<pre>ref StructMsg.total_count R);</pre>		
VB.Net 方法						
数据结构	结构名称: Pwd,联	机密码及写入	密码设置 <b>(</b> 请	参考第三章)。		
	结构名称: total_c	ount				
	变量名称	型别	[in/out]	说明		
	Count	short	out	联机机台数		
备注						
适用范围	不限控制器。					
C#程序范例	StructMsg.Pwd _P	wd = new St	ructMsg.Pw	ud();		
	_Pwd.ConnectionK	ey = "123";				
	StructMsg.total_	count R = n	new StructM	lsg.total_count();		
	short ret = iRem	<pre>short ret = iRemoting.SKY_conn_count(_Pwd, ref R);</pre>				
	if (ret == 0)					
	MessageBox.SI	<pre>MessageBox.Show(R.Count.ToString());</pre>				
	else					
	MessageBox.S	now(ret.ToS <sup>,</sup>	tring());			

### 七、基本类别函式

### 1. GET\_information: 基本相关信息

API 名称	GET_information							
函式说明	基本相关信息							
C#方法	short GET_info	rmation(St	ructMsg.Pwd	P, refStructMs	g.information	R);		
VB.Net 方法	GET_informatio	on(ByVal P	As StructMs	g.Pwd, ByRef R	As			
	StructMsg.info	ormation) A	s Short					
数据结构	结构名称: Pwd,	联机密码及算	写入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。				
	结构名称: infor	rmation				-		
	变量名称         型别         [in/out]         说明							
	Axes	short	out	可控制轴数				
	CncType	string	out	ex :"18", "31	", "0"			
	MaxAxes	short	out	最大轴数				
	Series	string	out	M/T type				
	Nc_Ver	string	out	NC 版本				
	AxisName	<pre>string[]</pre>	out	各轴坐标名称				
备注								
适用范围	0:支援 X:	未支援 ?:	未实作					
	FANUC	三菱	海德汉	西门子	新代			
	0	0	Х	?	0			
	宝元	工研院						
	0	0						
C#程序范例	StructMsg.info	ormation _i	nformation;					
	StructMsg.Pwd	_Pwd;						
	private void b	outton1_Cli	ck(object s	ender, EventAr	gs e)			
	{							
	_Pwd.Conne	ctionKey =	"123";					
	short ret	= iRemoting	g.GET_inform	nation(_Pwd, re	ef _informatio	on);		
	if (ret ==	0)						
	{							
	txtAxe	s.Text = _i	information.	Axes.ToString(	);			
	txtCnc	Type.Text =	= _informati	on.CncType;				
	txtMax	Axes.Text =	= _informati	on.MaxAxes.ToS	<pre>string();</pre>			
	txtSer	ies.Text =	_informatio	on.Series;				
	txtNc_	Ver.Text =	_informatio	on.Nc_Ver;				
	}							
	}							

2. GET\_information\_heid: 海德汉控制器\_基本相关信息

API 名称	GET_informati	GET_information_heid					
函式说明	基本相关信息						
C#方法	<pre>short GET_information_heid(StructMsg.Pwd P, ref</pre>						
	StructMsg.inf	ormation_he	eid R);				
<b>VB.Net</b> 方法	GET_informati	on_heid(By\	/al P As Str	uctMsg.Pwd, ByP	Ref R As		
	StructMsg.inf	ormation_he	eid) As Shor	٠t			
数据结构	结构名称: Pwd,	联机密码及	写入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。			
	结构名称: info	rmation_he	id			I	
	变量名称	型别	[in/out]	说明		1	
	Axes	short	out	可控制轴数		1	
	Model	string	out			1	
	Nc_Ver	string	out			1	
	FCL	string	out			1	
	Plc_Ver	string	out			1	
	AxisName	string[]	out	各轴坐标名称			
备注							
适用范围	0 : 支援 X :	未支援 ?:	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	X	X	0	X	Х		
	宝元	工研院					
	x	Х					
C#程序范例	StructMsg.info	ormation_hei	.d _informati	ion_heid;			
	StructMsg.Pwd	_Pwd;					
	private void b	outton1_Clic	k(object ser:	nder, EventArgs	e)		
	{						
	_Pwd.Conne	ctionKey =	"123";				
	information h	= ikemoting	.GEI_INTORMA	tion_neid(_Pwa,	гет		
	if (ret	(a)					
	{	0)					
	txtAxes	s.Text = ir	nformation he	eid.Axes.ToStrin	ug():		
	txtMode	el.Text = i	information I	heid.Model;			
	txtNc \	/er.Text =	_ information	_heid.Nc_Ver;			
	txtFCL	- Text = _inf.	- formation_he:	id.FCL;			
	txtPlc_	_Ver.Text =	_informatio	n_heid.Plc_Ver;			
	txtAxis	Name.Text =	= _informatio	on_heid.AxisName	e[0];		
	}						
	}						

3. GET\_status: 状态信息

API 名称	GET_status	GET_status					
函式说明	状态信息						
C#方法	short GET_sta	tus(StructMs	g.Pwd P, r	ef StructMsg.	status R);		
VB.Net 方法	GET_status(By	Val P As Stru	ictMsg.Pwd,	ByRef R As Str	uctMsg.status	) As	
	Short						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。			
	结构名称: stat	us				1	
	变量名称	型别	[in/out]	说明			
	MainProg	string	out	主程序号码			
	CurProg	string	out	目前执行程序的	」号码	-	
	CurSeq	int	out	目前执行程序的	门行号		
	Mode	string	out	ex:"MDI", "M	EM"		
	Status	string	out	ex:"STOP", "	START"	_	
	Alarm	string	out	ALARM			
	Emg	string	out	EMG			
	Monition	string	out	DWL / MTN			
备注	Monition 目前只	只支持 FANUC					
适用范围	0:支援 X:	未支援 ?::	未实作			•	
	FANUC	三菱	海德汉	西门子	新代		
	0	0	0	?	0		
	宝元	工研院					
	Ο	0					
C#程序范例	StructMsg.sta	tus _status;					
	StructMsg.Pwd	_Pwd;					
	private void	button2_Clic	k(object s	ender, EventA	rgs e)		
	{						
	_Pwd.Conne	ectionKey =	"123";				
	short ret	= iRemoting	.GET_statu	s(_Pwd, ref _s	status);		
	if (ret ==	= 0)					
	{						
	txtMai	nProg.Text =	= _status.M	lainProg;			
	txtCur	Prog.Text =	_status.Cu	ırProg;			
	txtCur	Seq.Text = _	_status.Cur	Seq.ToString(	);		
	txtMod	e.Text = _st	tatus.Mode;				
	txtSta	tus.Text = _	_status.Sta	itus;			
	txtAla	rm.Text = _s	status.Alar	'm;			
	txtEMG	.Text = _sta	atus.Emg;				
	}						

}

# 4. GET\_position: 坐标信息

API 名称	GET_position						
函式说明	坐标信息						
C#方法	short GET_posi	ition(Struct	Msg.Pwd P,	ref StructMsg	.position R);		
VB.Net 方法	GET_position(E	ByVal P As Sti	ructMsg.Pwd	d, ByRef R As Str	uctMsg.positi	ion)	
	As Short						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。			
	结构名称: posit	tion				_	
	变量名称	型别	[in/out]	说明			
	AxisName	string[]	out	各轴坐标名称			
	DecPoint	sohrt	out	坐标小数点位数			
	Unit	string[]	out	坐标单位			
	Mach	double[]	out	机械坐标			
	Abs	double[]	out	绝对坐标			
	Rel	double[]	out	相对坐标			
	Dist	double[]	out	剩余距离			
备注	※ 海德汉控制器	不支持相对坐	标。				
适用范围	0:支援 X:	未支援 ?::	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	0	0	0	0		
	宝元	工研院					
	0	0					
C#程序范例	StructMsg.posi	tion _positi	on;				
	StructMsg.Pwd	_Pwd;					
	private void b	utton11_Clic	k(object se	nder, EventArgs	e)		
	{						
	_Pwd.Connec	tionKey = "1	L23";				
	short ret =	= iRemoting.(	GET_position	n(_Pwd, ref _pos	sition);		
	if (ret ==	0)					
	{						
	txtAbs.	Text = _posi	tion.Abs[0]	<pre>l.ToString();</pre>			
	txtMach	.Text = _pos	ition.Mach[	[0].ToString();			
	txtRel.	Text = _posi	tion.Rel[0]	<pre>.ToString();</pre>			
	txtDist	.Text = _pos	ition.Dist[	[0].ToString();			
	}						
	}						

## 5. GET\_gcode: G Code

API 名称	GET_gcode					
函式说明	G Code					
C#方法	short GET_gcod	de(StructMsg	.Pwd P, re	f StructMsg.gcd	ode R);	
<b>VB.Net</b> 方法	GET_gcode(ByVa	al P As Stru	ctMsg.Pwd,	ByRef R As Str	uctMsg.gcode)	) As
	Short					
数据结构	结构名称: gcode	e				
	变量名称	型别	[in/out]	说明		
	Gdata	<pre>string[]</pre>	out	G Code		
备注						
适用范围	0 : 支援 X :	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	Ο	0				
C#程序范例	StructMsg.gcod	de _gcode;				
	StructMsg.Pwd	_Pwd;				
	private void b	outton3_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ctionKey = '	"123";			
	short ret	= iRemoting	.GET_gcode(	(_Pwd, ref _gco	ode);	
	if (ret ==	: 0)				
	{					
	for (i	nt i = 0; i	<= _gcode.	Gdata.Length -	1; i++)	
	{					
	txt	GCode.Text	+= _gcode.0	Gdata[i] + " ";	;	
	}					
	}					
	}					

## 6. GET\_othercode: 其他 Code(H Code,D Code,T Code,M Code,B Code,F Code,S Code)

API 名称	GET_othercode					
函式说明	G Code (H Code, D Code, T Code, M Code, B Code, F Code, S Code)					
C#方法	<pre>short GET_othercode(StructMsg.Pwd P, ref StructMsg.othercode R);</pre>					
<b>VB.Net</b> 方法	GET_othercode(ByVal P As StructMsg.Pwd, ByRef R As					
	StructMsg.othercode) As Short					
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: othercode					

	变量名称	型别	[in/out]	说明		
	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,只有三家	<b></b> 差有支持	
	FCode	int	out	F Code		
	SCode	int	out	S Code		
备注	※ 只有三菱有支	持 B Code				
	※ 海德汉控制器	只支持 M Code	e、F Code、	T Code、S Code		
适用范围	0:支援 X:	未支援 ?: 🗦	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	0	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.othe	ercode _othe	rcode;			
	StructMsg.Pwd	_Pwd;				
	private void b	outton4_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ctionKey = '	'123" <b>;</b>			
	short ret	= iRemoting	.GET_othero	code(_Pwd, ref	_othercode);	
	if (ret ==	0)				
	{					
	txtHCod	de.Text = _o	thercode.H	Code.ToString(	);	
	txtDCoo	de.Text = _o	thercode.D	Code.ToString(	);	
	txtTCoo	de.Text = _o	thercode.T	Code.ToString(	);	
	txtMCod	de.Text = _o	thercode.M	Code.ToString(	);	
	txtBCoo	de.Text = _c	thercode.B	Code.ToString(	);	
	txtFCod	de.Text = _o	thercode.F	Code.ToString(	);	
	txtSCoo	de.Text = _o	thercode.S	Code.ToString(	);	
	}					
	}					

# 7. GET\_feed\_spindle: 进给率/转速

API 名称	GET_feed_spindle
函式说明	进给率/转速
<b>C#</b> 方法	<pre>short GET_feed_spindle(StructMsg.Pwd P, ref StructMsg.feed_spindle</pre>
	R);
VB.Net 方法	GET_feed_spindle(ByVal P As StructMsg.Pwd, ByRef R As

	StructMsg.feed_spindle) As Short					
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。					
	结构名称: feed_spindle					
	变量名称	型别	[in/out]	说明		
	OvFeed	double	out	Feed Rate Ove	r Ride	
	OvSpindle	double	out	Spindle Over	Ride	
	ActFeed	double	out	实际进给率		
	ActSpindle	int	out	实际主轴转速		
备注						
适用范围	0:支援 X:	未支援 ?:氵	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	0	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.feed	d_spindle _f	eed_spindl	e;		
	StructMsg.Pwd	_Pwd;				
	private void b	outton5_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ctionKey = '	"123" <b>;</b>			
	short ret =	iRemoting.0	<pre>iET_feed_sp</pre>	indle(_Pwd, ref	f_feed_spindl	.e);
	if (ret ==	0)				
	{					
	txtOvF	eed.Text = _	_feed_spind	lle.OvFeed.ToSt	ring();	
	txt0vS	pindle.Text	= _feed_sp	indle.OvSpindl	e.ToString();	
	txtAct	Feed.Text =	_feed_spin	dle.ActFeed.To	<pre>String();</pre>	
	txtAct	Spindle.Text	: = _feed_s	pindle.ActSpin	dle.ToString(	);
	}					
	}					

8. GET\_time: 时间

API 名称	GET_time				
函式说明	取得 CNC 相关时间	盯			
C#方法	<pre>short GET_time(StructMsg.Pwd P, ref StructMsg.time R);</pre>				
VB.Net 方法	GET_time(ByVa]	P As StructM	1sg.Pwd, By	Ref R As StructMsg.time) As Sh	ort
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。	
	结构名称: time	结构名称: time			
	变量名称	型别	[in/out]	说明	
	Power	int[]	out	开机时间	

	Cutting	int[]	out	切削时间		
	Cycle	int[]	out	CYCLE 时间		
	Operation	int[]	out	操作时间		
备注	※ Cutting tim	e:三菱不支援	0			
	※ Operation t	ime:工研院不	支援。			
	※ time 数据结构	的里面的所有变	至,固定都不	与 3 个数组,数组 <sup>,</sup>	中索引 0 为 hour	r,
	索引 1 为 minuit	e, 索引 2 为 s	second。			
适用范围	0:支援 X:	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	х	0				
C#程序范例	StructMsg.time	e_time;		·		
	StructMsg.Pwd	_Pwd;				
	private void b	outton6_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ctionKey = '	"123";			
	short ret	= iRemoting	.GET_time(_	_Pwd, ref _time	);	
	if (ret ==	0)				
	{					
	txtPowe	er.Text = _t	ime.Power[	0].ToString()	+ " : " +	
	_time.Power[1]	.ToString()	+ " : " +	_time.Power[2]	<pre>l.ToString();</pre>	
	txtCut	ting.Text =	_time.Cutt	ing[0].ToStrin	g() + " : " +	
	_time.Cutting[	1].ToString	() + " : "	+ _time.Cuttin	g[2].ToString	;();
	txtCyc	le.Text = _t	ime.Cycle[	0].ToString()	+ " : " +	
	_time.Cycle[1]	.ToString()	+ " : " +	_time.Cycle[2]	.ToString();	
	txt0pe	ration.Text	= _time.Op	eration[0].ToSt	:ring() + " :	" +
	_time.Operatic	on[1].ToStri	ng() + " :	" +		
	_time.Operatio	on[2].ToStri	ng();			
	}					
	}					

9. GET\_time\_heid: 海德汉控制器\_时间

API 名称	GET_time_heid
函式说明	取得 CNC 相关时间
C#方法	<pre>short GET_time_heid(StructMsg.Pwd P, ref StructMsg.time_heid R);</pre>
VB.Net 方法	<pre>GET_time(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.time_heid)</pre>
	As Short

数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。					
	结构名称: time	e_heid				_
	变量名称	型别	[in/out]	说明		
	MachineRunni	.ng int[]	out	This is the	cumulative	
				machining t	ime since	
				installatio	n.	
	MachineUp	int[]	out	This is the o	cumulative time	
				that the ma	chine has been	
				on (no emer	gency stop)	
				since insta	llation.	
	NcUp	int[]	out	This is the o	cumulative time	
				that the NC	has been turned	
				on since in	stallation of	
				the machine	•	
	SpindleRunni	.ng int[]	out	This is the o	cumulative time	
				the spindle	has been	
				running (M3	or M4) since	
				installatio	n of the	
				machine.		
备注						
适用范围	0 : 支援 X :	未支援 ?::	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	x	Х	0	Х	x	
	宝元	工研院				
	x	Х				
C#程序范例	StructMsg.time	e_heid _time_	heid;			
	StructMsg.Pwd	_Pwd;				
	private void b	outton6_Click	(object sende	r, EventArgs	e)	
	{					
	_Pwd.Conne	ctionKey = "1	123";			
	short ret	= iRemoting.(	GET_time_heid	(_Pwd, ref _t:	ime_heid);	
	if (ret ==	0)				
	{					
	txtMac	hineRunning.T	ext = _time_h	eid.MachineRu	nning[0] + ":"	+
			_time_H	neid.MachineR	unning[1];	
	txtMac	hineUp.Text =	<pre>_time_heid.M</pre>	lachineUp[0] +	- ":" +	
			_time_heid.M	<pre>lachineUp[1];</pre>		
	txtNcU	p.Text = _tim	ne_heid.NcUp[0	] + ":" + _ti	.me_heid.NcUp[1]	];
	<pre>txtSpindleRunning.Text = _time_heid.SpindleRunning[0] + ":" +</pre>					

_time_heid.SpindleRunning[1];	
}	
}	

10. GET\_time\_cnc: 取得控制器系统时间

API 名称	GET_time_cnc					
函式说明	取得控制器系统即	寸间				
C#方法	short GET_time	e_cnc(Struct	Msg.Pwd P,	ref StructMsg.	.time_cnc R);	
<b>VB.Net</b> 方法	GET_time_cnc(E	ByVal P As Str	uctMsg.Pwd	, ByRef R As Str	uctMsg.time_c	:nc)
	As Short					
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
	结构名称: time_	_cnc				_
	变量名称	型别	[in/out]	说明		
	SystemTime	short[]	out	控制器系统日期时	时间,会传回6	
				个数组如下:		
				[0]:年,[1]:月,	[2]:日,[3]:	
				时,[4]:分,[5]:	秒	
备注	※ 海德汉控制器	目前尚不支持。				
适用范围	0:支援 X:	未支援 ?: 🗦	未实作			1
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	Х	
	宝元	工研院				
	X	0				
C#程序范例	StructMsg.time	_cnc _time_cr	nc;			
	StructMsg.Pwd	_Pwd;				
	private void b	utton10_Clic	<pre>&lt;(object se</pre>	nder, EventArgs	e)	
	{					
	_Pwd.Connec	tionKey = "1	23";			
	short ret =	<pre>iRemoting.G</pre>	ET_time_cnd	:(_Pwd, ref _tin	ne_cnc);	
	txtSysTime.	Text = _time	_cnc.System	nTime[0] + "/" +	•	
	_time_cnc	SystemTime[	1] + "/" +	_time_cnc.Syste	emTime[2] + "	" + 
	time_cnc	.SystemTime[	3] + " <b>:</b> " +	_time_cnc.Syste	emlime[4] + ":	+
	time_cnc	.Systemlime[	5];			
	}					

11. GET\_part\_count: 工件数

API 名称	GET_part_count
函式说明	工件数

C#方法	<pre>short GET_part_count(StructMsg.Pwd P, ref StructMsg.part_count R);</pre>					
VB.Net 方法	GET_part_coun	t(ByVal P As	StructMsg	.Pwd, ByRef R	As	
	StructMsg.par <sup>.</sup>	t_count) As	Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: part	_count	•			-
	变量名称	型别	[in/out]	说明		
	part_count	int	out	工件数		
备注						
适用范围	0 : 支援 X :	未支援 ?::	未实作			-
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	0	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.par <sup>.</sup>	t_count _par	t_count;			
	StructMsg.Pwd	_Pwd;				
	private void	button7_Clic	k(object s	ender, EventAr	gs e)	
	{					
	_Pwd.Conne	ectionKey =	"123";			
	short ret	= iRemoting	.GET_part_c	count(_Pwd, ref	part_count	);
	if (ret ==	= 0)				
	{					
	txtPar	tCount.Text	= _part_co	unt.PartCount.	ToString();	
	}					
	}					

### 12. GET\_part\_total:

API 名称	GET_part_tota	1				
函式说明	总工件数					
C#方法	<pre>short GET_part_total(StructMsg.Pwd P, ref StructMsg.part_total R);</pre>				R);	
<b>VB.Net</b> 方法	GET_part_tota	GET_part_total(ByVal P As StructMsg.Pwd, ByRef R As				
	StructMsg.par	t_total) As	Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
	结构名称: part	_total				
	变量名称	型别	[in/out]	说明		
	PartTotal	int	out			
备注						
适用范围	0:支援 X:	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	

	0	Х	х	Х	0				
	宝元	工研院							
	Х	Х							
C#程序范例	StructMsg.pa	<pre>:ructMsg.part_total _part_total;</pre>							
	StructMsg.Pw	StructMsg.Pwd _Pwd;							
	private void button8_Click(object sender, EventArgs e)								
	{								
	_Pwd.ConnectionKey = "123";								
	short ret	: = iRemoting	.GET_part_tot	cal(_Pwd, ref	_part_total	);			
	if (ret =	= 0)							
	{								
	txtPa	rtTotal.Text	= _part_tota	l.PartTotal.	ToString();				
	}								
	}								

## 13. GET\_part\_required:

API 名称	GET_part_requi	ired				
函式说明	需求工件数	需求工件数				
C#方法	short GET_part	t_required(S <sup>.</sup>	tructMsg.P	wd P, ref		
	StructMsg.part	t_required R	);			
VB.Net 方法	GET_part_requi	ired(ByVal P	As Struct	Msg.Pwd, ByRef	R As	
	StructMsg.part	t_required)	As Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
	结构名称: part	_required				
	变量名称	型别	[in/out]	说明		
	PartRequired	int	out			]
备注						-
适用范围	0 : 支援 X :	未支援 ?: 🕫	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	Х	Х	x	0	
	宝元	工研院				
	X	Х				
C#程序范例	StructMsg.part	t_required _	part_requi	red;		
	StructMsg.Pwd	_Pwd;				
	private void b	outton9_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ctionKey = '	'123" <b>;</b>			
	short ret	= iRemoting.	.GET_part_r	required(_Pwd,	ref	
	_part_required	d);				

if (ret == 0)
{
<pre>txtPartRequired.Text =</pre>
_part_required.PartRequired.ToString();
}
}

## 14. SET\_relpos: 设定相对坐标数值

API 名称	SET_relpos						
函式说明	设定相对坐标数值						
C#方法	short SET_rel	pos(StructMs	g.Pwd P, S <sup>.</sup>	tructMsg.relp	os R);		
VB.Net 方法							
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。			
	结构名称: relp	os					
	变量名称	型别	[in/out]	说明			
	AxisName	string	in	坐标名称			
	PosValue	double	in	写入到控制器的	句坐标数值	1	
备注						J	
适用范围	0 : 支援 X :	未支援 ?: 🗦	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	?	?	?	0		
	宝元	工研院					
	?	?					
C#程序范例	StructMsg.Pwd	_Pwd;					
	StructMsg.rel	pos _relpos;					
	private void	button12_Cli	ck(object	sender, Event	Args e)		
	{						
	_Pwd.Write	ePwd = "123";	;				
			_				
	_relpos.A	xisName = "X"	';				
	_relpos.Po	osvalue = 0;	CFT				
	short ret	= iRemoting.	SEI_reipos	s(_Pwd, _relpo	)\$);		
	_relpos.A	xisName = "Y'	';				
	_relpos.Po	osValue = 0;					
	ret = iRe	moting.SET_re	elpos(_Pwd,	_relpos);			
	_relpos.A:	xisName = "Z'	';				

_relpos.PosValue = 0;
<pre>ret = iRemoting.SET_relpos(_Pwd, _relpos);</pre>
}

#### 15. SET\_time\_cnc: 设定控制器的时间

	2.7.2.1.1.1						
API 名称	SET_time_cnc	SET_time_cnc					
函式说明	设定控制器的时间	设定控制器的时间					
C#方法	short SET_time	e_cnc(Struct	Msg.Pwd P,	StructMsg.tim	e_cnc R);		
VB.Net 方法							
数据结构	结构名称: Pwd,	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: time	_cnc					
	变量名称	型别	[in/out]	说明			
	SystemTime	short[]	in	写入到控制器系:	统日期时间,必		
				须有 6 个数组如	下:		
				[0]:年,[1]:月	,[2]:日,[3]:		
				时,[4]:分,[5]	:秒		
备注							
适用范围	0 : 支援 X :	未支援 ?:	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	?	?	?	?		
	宝元	工研院					
	?	?					
C#程序范例	StructMsg.time	e_cnc _time_	cnc = new	StructMsg.time	_cnc();		
	_time_cnc.Syst	temTime = ne	w short[]{	(short)DateTim	e.Now.Year,		
			(s)	hort)DateTime.N	Now.Month,		
			(s	hort)DateTime.N	Now.Day,		
			(s	hort)DateTime.N	Now.Hour,		
			(s	hort)DateTime.N	√ow.Minute,		
			(s)	hort)DateTime.N	√ow.Second};		
	short ret = i	Remoting.SET	_time_cnc(	_Pwd, _time_cn	c);		

## 八、警报类别函式

## 1. GET\_alm\_current: 目前发生的警报

API 名称	GET_alm_currer	nt				
函式说明	目前发生的警报	目前发生的警报				
C#方法	short GET_alm_	current(Stru	uctMsg.Pwd	P, refStructMs	sg.alm_current	R);
<b>VB.Net</b> 方法	GET_alm_currer	nt(ByVal P A	s StructMs	g.Pwd, ByRef I	R As	
	StructMsg.alm_	_current) As	Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: alm_o	current				_
	变量名称	型别	[in/out]	说明		
	IsAlarm	bool	out	判断是否有警报	J Z	
	AlmCode	int	out	取得警报编号		
	AlmMsg	string	out	取得目前发生的	的警报讯息	
备注	※ 警报发生时,	AlmMsg 是属于	空白,可能	数据库尚未建立。		
适用范围	0 : 支援 X :	未支援 ?: 🗦	未实作			_
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	Ο	0				
C#程序范例	StructMsg.alm_	_current _al	m_current;			
	StructMsg.Pwd	_Pwd;				
	private void b	outton1_Clic	k(object s	ender, EventA	rgs e)	
	{					
	_Pwd.Conne	ctionKey = '	"123";			
	short ret	= iRemoting.	.GET_alm_cu	urrent(_Pwd, r	ef _alm_curre	ιt);
	if (ret ==	0 && _alm_0	current.Is/	Alarm)		
	{					
	txtAlm	sg.Text = _a	alm_current	.Almsg;		
	}					
	}					

### 2. GET\_alm\_current2: 目前发生的警报

API 名称	GET_alm_current2
函式说明	目前发生的警报
C#方法	<pre>short GET_alm_current2(StructMsg.Pwd P, ref StructMsg.alm_current2</pre>
	R);
<b>VB.Net</b> 方法	GET_alm_current2(ByVal P As StructMsg.Pwd, ByRef R As
	StructMsg.alm_current2) As Short

数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。							
	结构名称: alm_current2							
	变量名称	型别	[in/out]	说明				
	IsAlarm	bool	out	判断是否有警报				
	AlmClass	string[]	out	警报类别				
	AlmCode	string[]	out	警报编号				
	AlmMsg	string[]	out	目前发生的警报i	汛息			
备注	※ 变量 AlmClass(警报类别)适用在:海德汉及新代控制器。							
	当有「SkyMars 维护管理」或「SkyMars 刀具寿命」警报发生时,所属类别名							
	称将会写入到 AlmClass 变量内:							
	「SkyMars 维护管理警报」所属类别名称为 SkyMarsMaintenance							
	「SkyMars 刀具寿命警报」所属类别名称为 SkyMarsToolManagement							
	※ 变数 AlmCode 适用在: FANUC、海德汉及新代控制器。							
	※ 变数 AlmMsg 适用在: FANUC、三菱、海德汉、新代控制器、SkyMars 维护管理							
	警报及 SkyMars 刀具寿命警报。							
适用范围	0 : 支援 X :	未支援 ?:	未实作			-		
	FANUC	三菱	海德汉	西门子	新代			
	0	0	0	0	0			
	宝元	工研院						
	0	0						
C#程序范例	StructMsg.alm_current2 _alm_current2;							
	StructMsg.Pwd _Pwd;							
	<pre>private void button8_Click(object sender, EventArgs e)</pre>							
	{							
	_Pwd.ConnectionKey = "123";							
	<pre>short ret = iRemoting.GET_alm_current2(_Pwd, ref _alm_current2);</pre>							
	if (ret == 0)							
	{							
	<pre>txtC1.Text = _alm_current2.IsAlarm.ToString();</pre>							
	if (_alm_current2.IsAlarm)							
	{							
	txt	C2.Text = _a	lm_current2	.AlmClass[0];				
	<pre>txtC3.Text = _alm_current2.AlmCode[0];</pre>							
	<pre>txtC4.Text = _alm_current2.AlmMsg[0];</pre>							
	}							
	}							
	}							
# 3. GET\_alm\_current\_heid: 海德汉控制器\_目前发生的警报

API 名称	GET_alm_current_heid								
函式说明	目前发生的警报								
C#方法	<pre>short GET_alm_current_heid(StructMsg.Pwd P, ref</pre>								
	StructMsg.alm	_current_hei	d R);						
VB.Net 方法	GET_alm_curre	nt_heid(ByVa	l P As Struc	tMsg.Pwd, By	Ref R As				
	StructMsg.alm	_current_hei	d) As Short						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(请	参考第三章 <b>)</b> 。					
	结构名称: alm_current_heid								
	变量名称     型别     [in/out]     说明								
	IsAlarm	bool	out	判断是否有警	报				
	ErrNumber	int[]	out	Nuumber err	or.				
	ErrGroup	string[]	out	Group error	•				
	ErrClass	string[]	out	Class error	•				
	ErrMsg	string[]	out	错误讯息。					
	ErrDescripti	on string[]	out	错误描述					
备注									
适用范围	0 : 支援 X :	未支援 ?:	未实作			_			
	FANUC	三菱	海德汉	西门子	新代				
	x	Х	0	Х	Х				
	宝元	工研院							
	x	Х							
C#程序范例	StructMsg.alm_	_current_heid	_alm_current	_heid;					
	StructMsg.Pwd	_Pwd;							
	private void b	outton6_Click	(object sende	r, EventArgs	e)				
	{								
	_Pwd.Conne	ctionKey = "1	.23";						
	short ret	= iRemoting.0	GET_alm_curre	nt_heid(_Pwd,	ref				
	_alm_current_h	neid);							
	if (ret ==	0 && _alm_cu	urrent_heid.Is	sAlarm)					
	{								
	txtNum	per.Text = _a	lm_current_he	id.ErrNumber[	[0].ToString()	;			
	txtGrou	up.Text = _al	m_current_hei	d.ErrGroup[0]	;				
	txtClas	ss.Text = _al	<pre>m_current_hei</pre>	d.ErrClass[0]	;				
	txtMsg	.Text = _alm_	current_heid.	ErrMsg[0];					
	txtDeso	cription.Text	= _alm_curre	ent_heid.ErrDe	escription[0];				
	}								
	}								

4. GET\_alm\_history: 警报履历

API 名称	GET_alm_histo	°У						
函式说明	警报履历							
<b>C#</b> 方法	short GET_alm_	<pre>short GET_alm_history(StructMsg.Pwd P, ref StructMsg.alm_history R);</pre>						
<b>VB.Net</b> 方法	GET_alm_histo	∽y(ByVal P A	s StructMs	g.F	Pwd, ByRef R	As		
	StructMsg.alm	_history) As	Short					
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请爹	参考第三章 <b>)</b> 。			
	结构名称: alm_	nistory						
	变量名称	型别	[in/out]	访	纪明			
	AlmCode	int[]	out	敬言	<b>肾报编号</b>			
	AlmDate	string[]	out	敬言	降报日期			
	AlmMsg	string[]	out	敬言	各报讯息			
备注	变量 AlmCode 目	前不适用在三蒙	菱。					
适用范围	0 : 支援 X :	未支援 ?:	未实作					
	FANUC	三菱	海德汉		西门子	新代		
	0	0	Х		?	0		
	宝元	工研院						
	0	0						
C#程序范例	StructMsg.alm	_history _al	m_history;					
	StructMsg.Pwd	_Pwd;						
	private void l	outton2_Clic	k(object s	end	der, EventArg	gs e)		
	{							
	_Pwd.Conne	ctionKey =	"123";					
	short ret	= iRemoting	.GET_alm_hi	İst	ory(_Pwd, re	f _alm_histor	·y);	
	if (ret ==	: 0)						
	{							
	txtAlm	Code.Text =	_alm_histo	ory	.AlmCode[0].	ToString();		
	txtAlm	Date.Text =	_alm_histo	ory	.AlmDate[0];			
	txtAlm	Msg.Text = _	_alm_histor	٠y.	A1mMsg[0];			
	}							
	}							

## 5. GET\_alm\_history2: 警报履历

API 名称	GET_alm_history2
函式说明	警报履历
C#方法	<pre>short GET_alm_history2(StructMsg.Pwd P, ref StructMsg.alm_history2</pre>
	R);
VB.Net 方法	GET_alm_histor2y(ByVal P As StructMsg.Pwd, ByRef R As
	StructMsg.alm_history2) As Short

数据结构	结构名称: Pwd,	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。							
	结构名称: alm_	history2							
	变量名称	型别	[in/out]	说明					
	AlmClass	string[]	out	警报类别					
	AlmCode	string[]	out	警报编号					
	AlmDate	string[]	out	警报日期					
	AlmMsg	<pre>string[]</pre>	out	警报讯息					
备注	※ 变数 AlmCla	※ 变数 AlmClass 适用在:海德汉及新代控制器。							
	当有「SkyMa	rs 维护管理」	或「SkyMars	刀具寿命」警报发	<b>文生时,所属类</b> 第	引名			
	称将会写入到	称将会写入到 AlmClass 变量内:							
	「SkyMars 绯	主护管理警报」	所属类别名称	为 SkyMarsMaint	enance				
	「 <b>SkyMars</b> フ	」具寿命警报」	所属类别名称	为 SkyMarsToolM	lanagement				
	※ 变数 AlmCod	e适用在: FANU	JC、海德汉及	新代控制器。					
	※ 变数 AlmDate	e适用在: FANU	JC、三菱、海	德汉及新代控制器	ž To				
	※ 变数 AlmMsg	适用在: FANUC	2、三菱、海德	您汉、新代控制器、	SkyMars 维护	管理			
	警报及 SkyMa	ars 刀具寿命警	报。						
适用范围	0 : 支援 X :	未支援 ?: 5	未实作			1			
	FANUC	三菱	海德汉	西门子	新代	_			
	0	0	0	?	0				
	宝元	工研院				_			
	0	0							
C#程序范例	StructMsg.alm_	history2 _al	n_history2;						
	StructMsg.Pwd	_Pwd;	<i>.</i>		,				
	private void b	outton7_Click	(object send	der, EventArgs	e)				
			22.1						
	_Pwd.Conne	ctionKey = "1	.23"; 		alm history	- <b>)</b> .			
	snort ret	= iRemoting.e	EI_aIm_nist	ory2(_Pwa, ret	_aim_nistory	2);			
	IT (ret ==	0)							
	i if ( a)	lm history? A	lmClass lon	a + b > 0					
	11 (_a1	III_1113101 y2.A		gen > 0)					
	txt	H1.Text = a	lm historv2	AlmClass[0]:					
	txt	H2.Text = a	lm_historv2.	AlmCode[0]:					
	txt	H3.Text = a	lm historv2.	.AlmDate[0]:					
	txt	H4.Text = a	 Lm history2.	.AlmMsg[0];					
	}	-	_ ,	01 17					
	}								
	}								

## 6. GET\_alm\_history\_heid: 警报履历\_海德汉控制器

API 名称	GET_alm_history_heid									
函式说明	警报履历_海德汉控制器									
C#方法	<pre>short GET_alm_history_heid(StructMsg.Pwd P, ref</pre>									
	<pre>StructMsg.alm_history_heid R);</pre>									
<b>VB.Net</b> 方法	GET_alm_history_heid(ByVal P As StructMsg.Pwd, ByRef R As									
	StructMsg.alm_h	StructMsg.alm_history_heid) As Short								
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。									
	结构名称: alm_history_heid									
	变量名称     型别     [in/out]     说明									
	ErrNumber	int[]	out	Nuumber err	or.					
	ErrGroup	<pre>string[]</pre>	out	Group error	•					
	ErrClass	<pre>string[]</pre>	out	Class error	•					
	ErrMsg	<pre>string[]</pre>	out	错误讯息。						
	ErrDescription	<pre>string[]</pre>	out	错误描述						
	ErrDate	<pre>string[]</pre>	out	警报日期						
备注										
适用范围	0:支援 X:未	支援 ?: 🗄	未实作			7				
	FANUC	三菱	海德汉	西门子	新代					
	x	Х	0	Х	Х					
	宝元	工研院								
	x	Х								
C#程序范例	StructMsg.alm_hi	story_heid	_alm_history	_heid;						
	StructMsg.Pwd _P	wd;								
	private void but	ton1_Click	(object sende	r, EventArgs	e)					
	{									
	_Pwd.Connect:	ionKey = "1	.23";							
	short ret = :	iRemoting.0	GET_alm_histor	ry_heid(_Pwd,	ref					
	_alm_history_hei	d);								
	it (ret == 0	)								
		الم الم الم								
		atheia =	"HData" Type	();	tom Staing"))					
	dtHeid Co	Jumps Add(	"EnpNumber"	June CetTune(	"System Int32	; "\\.				
	dtHaid Co	Jumns Add(	" Group" Typ	e GetTune/"Cu	stem String"	))) ))				
	dtHaid Co	lumns Δdd(	_Group , typ " Class" Typ	e GetTyne("Sy	(stem String)	), ),				
	dtHeid Co	lumns Add(	" Msg". Tvne	GetTyne("Svst	em.String")).	/ 3				
	dtHeid.Co	lumns.Add(	" Description	".						
	Type.GetType("Sv	stem.String	g"));	,						
		•								

### 7. GET\_msg\_current: 取得 Operation 讯息

API 名称	GET_msg_curren	GET_msg_current							
函式说明	取得 Operation 讯息								
C#方法	short GET_msg_	<pre>short GET_msg_current(StructMsg.Pwd P, ref StructMsg.msg_current R);</pre>							
VB.Net 方法	GET_msg_current(ByVal P As StructMsg.Pwd, ByRef R As								
	StructMsg.msg	_current) As	Short						
数据结构	结构名称: Pwd,	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。							
	结构名称: msg_	current							
	变量名称	型别	[in/out]	说明					
	IsMsg	bool	out	判断是否有 Oper	ration 讯息				
	MsgCode	short	out	讯息编号					
	MsgText	string	out	讯息内容					
备注									
适用范围	0 : 支援 X :	未支援 ?:	未实作						
	FANUC	三菱	海德汉	西门子	新代				
	0	Х	Х	?	Х				
	宝元	工研院							
	X	Х							
C#程序范例	StructMsg.msg	_current _ms	g_current;						
	StructMsg.Pwd	_Pwd;							
	private void b	outton3_Clic	k(object s	ender, EventAr	gs e)				
	{								

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

# 8. GET\_msg\_history: 取得 Operation 履历

API 名称	GET_msg_histo	°у				GET_msg_history						
函式说明	取得 Operation 履历											
C#方法	short GET_msg_	<pre>short GET_msg_history(StructMsg.Pwd P, ref StructMsg.msg_history R);</pre>										
VB.Net 方法	GET_msg_histor	ry(ByVal P A	s StructMs	g.F	Pwd, ByRef R	As						
	StructMsg.msg_	_history) As	Short									
数据结构	结构名称: Pwd,	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。										
	结构名称: msg_l	nistory	-									
	变量名称	型别	[in/out]	说	纪明							
	MsgCode	short[]	out	讦	记息编号							
	MsgDate	string[]	out	访	方息日期							
	MsgText	string[]	out	计	记息内容							
备注	此函式三菱不支持	爱。										
适用范围	0 : 支援 X :	未支援 ?:	未实作									
	FANUC	三菱	海德汉		西门子	新代						
	0	Х	Х		?	Х						
	宝元	工研院										
	Х	Х										
C#程序范例	StructMsg.msg_	_history _ms	g_history;									
	StructMsg.Pwd	_Pwd;										
	private void b	outton4_Clic	k(object s	end	der, EventArg	gs e)						
	{											
	_Pwd.Conne	ctionKey =	"123";									
	short ret	= iRemoting	.GET_msg_hi	ist	ory(_Pwd, re	f _msg_histor	·у);					
	if (ret ==	: 0)										
	{	- · ·										
	txtMsg	Lode2.Text =	<pre>_msg_hist</pre>	or	y.MsgCode[0]	.loString();						
	txtMsg	Date.lext =	_msg_histo	ory	.msgDate[0];							
	txtMsg	iext2.lext =	• _msg_nist	or	y.msgiext[0]	;						
	}											

}

# 9. GET\_plc\_alarm: 取得 PLC Alarm 履历

API 名称	GET_plc_alarm								
函式说明	取得 PLC Alarm 履历								
C#方法	short GET_plc	<pre>short GET_plc_alarm(StructMsg.Pwd P, ref StructMsg.plc_alarm R);</pre>							
<b>VB.Net</b> 方法	GET_plc_alarm	(ByVal P As	StructMsg.	Pwd	l, ByRef R As	5			
	StructMsg.plc	_alarm) As S	hort						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参	参考第三章 <b>)</b> 。				
	结构名称: plc_	alarm							
	变量名称	型别	[in/out]	说	明				
	AlmMsg	string[]	out	PL	-C 错误讯息				
备注									
适用范围	0:支援 X:	未支援 ?:	未实作						
	FANUC	三菱	海德汉		西门子	新代			
	0	Х	Х		?	Х			
	宝元	工研院							
	0	?							
C#程序范例	StructMsg.plc	_alarm _plc_	alarm;						
	StructMsg.Pwd	_Pwd;							
	private void	button5_Clic	k(object s	end	ler, EventArg	gs e)			
	{								
	_Pwd.Conne	ectionKey = '	"123";						
	short ret	= iRemoting	.GET_plc_al	ları	m(_Pwd, ref	_plc_alarm);			
	if (ret ==	= 0)							
	{								
	if (_p	lc_alarm.Alm	ıMsg != nul	1)					
	tx	tAlmMsg2.Tex	t = _plc_a	lar	m.AlmMsg[0];				
	}								
	}								

## 九、伺服主轴类别函式

### 1. GET\_servo\_current: 各轴负载电流值

API 名称	GET_servo_current								
函式说明	各轴负载电流值								
C#方法	<pre>short GET_servo_current(StructMsg.Pwd P, ref</pre>								
	StructMsg.serv	vo_current R	);						
<b>VB.Net</b> 方法	GET_servo_curr	rent(ByVal P	As Struct	Msg.Pwd, ByRef	R As				
	StructMsg.serv	vo_current)	As Short						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。					
	结构名称: serve	p_current							
	变量名称	型别	[in/out]	说明					
	AxisCurrent	int[]	out	目前电流值					
备注									
适用范围	0:支援 X:	未支援 ?: 🕫	未实作						
	FANUC	三菱	海德汉	西门子	新代				
	0	Х	Х	?	x				
	宝元	工研院							
	х	Х							
C#程序范例	StructMsg.serv	/o_current _	servo_curr	ent;					
	StructMsg.Pwd	_Pwd;							
	private void b	outton1_Clic	k(object s	ender, EventAr	gs e)				
	{								
	_Pwd.Conne	ctionKey = '	"123";						
	short ret	= iRemoting	.GET_servo_	_current(_Pwd,	ref				
	_servo_current	t);							
	if (ret ==	: 0)							
	{								
	txtAxi	sCurrent.Tex	:t =						
	_servo_current	t.AxisCurren	t[0].ToStr	ing();					
	}								
	}								

### 2. GET\_servo\_load: 取得伺服负载%

API 名称	GET_servo_load
函式说明	取得伺服负载%
C#方法	<pre>short GET_servo_load(StructMsg.Pwd P, ref StructMsg.servo_load R);</pre>
VB.Net 方法	GET_servo_load(ByVal P As StructMsg.Pwd, ByRef R As
	StructMsg.servo_load) As Short

数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。							
	结构名称: servo_load							
	变量名称	型别		[in/out	] 说明	说明		
	ServoLoadPerce	nt int[]		out	负载值			
备注								
适用范围	0:支援 X:未	支援 ?:	未实作	Ē				
	FANUC	三菱	涟	<b>承德汉</b>	西门子	新代		
	0	0		0	?	х		
	宝元	工研院						
	х	0						
C#程序范例	StructMsg.servo	_load _ser	vo_l	oad;			-	
	StructMsg.Pwd _	Pwd;						
	private void bu	tton2_Clic	k(ob	ject sen	der, EventAr	gs e)		
	{							
	_Pwd.Connect	ionKey =	"123'	';				
	short ret =	iRemoting	.GET_	_servo_lo	oad(_Pwd, ref	<pre>_servo_load</pre>	);	
	if (ret == 0	))						
	{							
	for (int	i = 0; i <:	= _se	rvo_load	.ServoLoadPe	rcent.Length	- 1;	
	i++)							
	txtS	ervoLoadPe	rcen	t.Text +	=			
	_servo_load.Ser	voLoadPerc	ent[	i].ToStr	ing() + " ";	;		
	}							
	}							

## 3. GET\_servo\_speed: 取得伺服轴转速

API 名称	GET_servo_spee	ed		
函式说明	取得伺服轴转速			
C#方法	<pre>short GET_servo_speed(StructMsg.Pwd P, ref StructMsg.servo_speed R);</pre>			
VB.Net 方法	GET_servo_speed(ByVal P As StructMsg.Pwd, ByRef R As			
	StructMsg.serv	vo_speed) As	Short	
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。
	结构名称: servo	_speed		
	变量名称	型别	[in/out]	说明
	ServoSpd	int[]	out	伺服轴转速
备注				

适用范围	0:支援 X:未支援 ?:未实作						
	FANUC	三菱	海德汉	西门子	新代		
	0	0	0	?	Х		
	宝元	工研院					
	Х	0					
C#程序范例	StructMsg.se	<pre>rvo_speed _se</pre>	ervo_speed;				
	StructMsg.Pwd _Pwd;						
	<pre>private void button3_Click(object sender, EventArgs e)</pre>						
	{						
	_Pwd.Conn	ectionKey =	"123";				
	short ret	= iRemoting	.GET_servo_sp	eed(_Pwd, re	f _servo_spee	ed);	
	if (ret =	= 0)					
	{						
	txtSe	rvoSpd.Text =	= _servo_spee	d.ServoSpd[0]	].ToString();		
	}						
	}						

## 4. GET\_spindle\_load: 取得主轴负载%

API 名称	GET_spindle_lo	oad					
函式说明	取得主轴负载%	取得主轴负载%					
C#方法	short GET_spir	ndle_load(St	ructMsg.Pw	d P, ref Struct	Msg.spindle_]	oad	
	R);	R);					
VB.Net 方法	GET_spindle_lo	oad(ByVal P	As StructM	sg.Pwd, ByRef H	R As		
	StructMsg.spir	ndle_load) A	s Short				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。			
	结构名称: spine	结构名称: spindle_load					
	变量名称	型别	[in/out]	说明			
	SpLoad	double	out	主轴负载			
备注	只能适用在一颗量	主轴。					
适用范围	0 : 支援 X :	未支援 ?: 🕫	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	0	0	?	Х		
	宝元	工研院					
	X	0					
C#程序范例	StructMsg.spir	ndle_load _s	pindle_loa	d;			
	StructMsg.Pwd	_Pwd;					
	private void b	outton4_Clic	k(object s	ender, EventArg	gs e)		
	{						
	_Pwd.Conne	ctionKey = '	"123";				

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

# 5. GET\_spindle\_speed: 取得主轴转速

API 名称	GET_spindle_s	peed				
函式说明	取得主轴转速	取得主轴转速				
<b>C#</b> 方法	short GET_spi	ndle_speed(S	tructMsg.P	wd P, ref		
	StructMsg.spi	ndle_speed R	:);			
VB.Net 方法	GET_spindle_s	peed(ByVal P	As Struct	Msg.Pwd, ByRef	R As	
	StructMsg.spi	ndle_speed)	As Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: spin	dle_speed				
	变量名称	型别	[in/out]	说明		
	SpSpeed	int	out	主轴转速		
备注	只能适用在一颗	主轴。				-
适用范围	0 : 支援 X :	未支援 ?:	未实作			_
	FANUC	三菱	海德汉	西门子	新代	
	0	0	0	?	Х	
	宝元	工研院				
	x	0				
C#程序范例	StructMsg.spi	ndle_speed _	_spindle_sp	eed;		
	StructMsg.Pwd	_Pwd;				
	private void	button7_Clic	k(object s	ender, EventAr	gs e)	
	{					
	_Pwd.Conne	ectionKey =	"123";			
	short ret	= iRemoting	.GET_spind	<pre>Le_speed(_Pwd,</pre>	ref	
	_spindle_spee	d);				
	if (ret ==	= 0)				
	{					
	txtSpi	ndleSpeed.Te	ext = _spin	dle_speed.SpSp	eed.ToString(	();
	}					

}

## 6. GET\_servo\_temperature: 取得伺服马达温度

API 名称	GET_servo_temperature					
函式说明	取得伺服马达温度	度				
<b>C#</b> 方法	short GET_serv	vo_temperatu	re(StructM	sg.Pwd P, ref		
	StructMsg.ser	vo_temperatu	re R);			
VB.Net 方法	GET_servo_tem	perature(ByV	al P As St	ructMsg.Pwd, By	Ref R As	
	StructMsg.ser	vo_temperatu	re) As Sho	rt		
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
	结构名称: serve	o_temperatur	e			
	变量名称	型别	[in/out]	说明		
	AxisTemp	double[]	out	伺服马达温度		
备注						
适用范围	0:支援 X:	未支援 ?: 🗦	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	?	0	?	Х	
	宝元	工研院				
	х	Х				
C#程序范例	StructMsg.ser	vo_temperatu	re _servo_	temperature;		
	StructMsg.Pwd	_Pwd;				
	private void l	outton5_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ctionKey = '	"123";			
	short ret	= iRemoting	.GET_servo_	_temperature(_P	wd, ref	
	_servo_tempera	ature);				
	if (ret ==	: 0)				
	{					
	for (int i	=0;i<=_se	ervo_temper	rature.AxisTemp	.Length - 1; i	[++)
	{					
	txt	tServoTemp.T	ext +=			
	_servo_tempera	ature.AxisTe	mp[i].ToSt	ring() + " "	;	
	}					
	}					
	}					

## 7. GET\_spindle\_temperature: 取得主轴温度

API 名称	GET_spindle_temperature
函式说明	取得主轴温度

C#方法	<pre>short GET_spindle_temperature(StructMsg.Pwd P, ref</pre>					
	<pre>StructMsg.spindle_temperature R);</pre>					
VB.Net 方法	GET_spindle_tem	perature(By	Val P As St	ructMsg.Pwd,	ByRef R As	
	StructMsg.spind	le_temperat	ure) As Shoi	^t		
数据结构	结构名称: Pwd, 联	〔机密码及写入	、密码设置(请参	参考第三章 <b>)</b> 。		
	结构名称: spindl	e_temperatu	re			
	变量名称	型别	[in/out]	说明		
	Spindle_1_Temp	double	out	第一颗主轴温	度	
	Spindle_2_Temp	double	out	第二颗主轴温	度	
备注	0i-D / 30 系列,	目前只能读取	单一主轴温度			
适用范围	0:支援 X:未	支援 ?:未	实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	?	0	?	Х	
	宝元	工研院				
	x	Х				
C#程序范例	StructMsg.spind	le_temperat	ure _spindle	e_temperature	; ;	
	StructMsg.Pwd _	Pwd;				
	private void bu	tton6_Click	(object send	der, EventArg	gs e)	
	{					
	_Pwd.Connect	ionKey = "1:	123";			
	short ret =	iRemoting.0	GET_spindle_	_temperature(	_Pwd, ref	
	_spindle_temper	ature);				
	if (ret == 0	))				
	{					
	txtSpind	leTemp.Text	: =			
	_spindle_temper	ature.Spind	le_1_Temp.To	oString();		
	}					
	}					

- 十、刀具管理类别函式 (控制器)
  - 以下表格所列函式皆已过时,不再使用:

GET\_tool\_data\_mem

GET\_pocket\_data\_mem

1. GET\_offset\_title: 取得刀具 Offset 的标题栏

API 名称	GET_offset_title			
函式说明	取得刀具 Offset 的标题栏			
C#方法	<pre>short GET_offset_title(StructMsg.Pwd P, ref StructMsg.offset_title</pre>			
	R);			
VB.Net 方法	GET_offset_tit	:le(ByVal P	As StructM	sg.Pwd, ByRef R As
	StructMsg.offs	et_title) A	s Short	
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。
	结构名称: offse	et_title		
	变量名称	型别	[in/out]	说明
	OffsetTitle	string[]	out	标题栏
备注	※以下为 FANUC 相	示题栏		
	M系列			
	System A:			
	{"DATA"}			
	System B:			
	{"GEOM","WEAR"	'}		
	System C:			
	{"LENGIH GEOM"	, LENGIH WE	AR", "RADIU	S GEOM", "RADIUS WEAR" }
	T 系列			
	没有∨轴			
	{"WEAR X"."WEA	R Z"."WEAR R"	."WEAR T".	"GEOM X" - "GEOM Z" - "GEOM R" - "GEOM
	T"}		<i>,</i> ,	
	5			
	有Y轴:			
	{"WEAR X","WEA	R Z","WEAR R"	,"WEAR T",	"WEAR Y", "GEOM X", "GEOM Z", "GEOM
	R","GEOM T","@	GEOM Y"}		
	※以下为三菱标匙	<u></u> 应栏		
	Tool compensat	ion type :	1	

	{"DATA"}							
	Tool compensa	Tool compensation type : 2						
	{"LENGTH GEOM","LENGTH WEAR","RADIUS GEOM","RADIUS WEAR"}							
	Tool compensa	ation type :	3					
适用范围	0 : 支援 X :	未支援 ?:	未实作					
	FANUC	三菱	海德汉	西门子	新代			
	0	0	Х	?	0			
	宝元	工研院						
	0	0						
C#程序范例	StructMsg.of	fset_title _o	ffset_title;					
	StructMsg.Pwg	d _Pwd;						
	private void	button1_Clic	k(object sen	der, EventArg	gs e)			
	{							
	_Pwd.Conn	ectionKey =	"123";					
	short ret	= iRemoting.	GET_offset_ti	tle(_Pwd, ref	<pre></pre>	le);		
	if (ret =	= 0)						
	{							
	for (i	inti=0;i<=	_offset_title	e.Offset⊺itle	e.Length - 1; i	i++)		
	rt	tbTitle.Text	+= _offset_t	itle.OffsetTi	itle[i] + "	";		
	}							
	}							

## 2. GET\_offset\_all: 取得所有刀具 Offset

API 名称	GET_offset_al	1				
函式说明	取得所有刀具 Of	fset				
C#方法	short GET_off	<pre>short GET_offset_all(StructMsg.Pwd P, ref StructMsg.offset_all R);</pre>				R);
<b>VB.Net</b> 方法	GET_offset_al	l(ByVal P As	StructMsg.	Pwd, ByRef R A	٩s	
	StructMsg.off	set_all) As	Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(请	青参考第三章 <b>)</b> 。		
	结构名称: offset_all					
	变量名称	型别	[in/out]	说明		
	OffsetData	double[][]	out	补正资料		
备注	OffsetData 第二	二个数组,请配	合标题名称(L	ength Geom)。		
	使用此函式可能	会有大量的读取	、时间 <b>(</b> 一般都)	发生在低速网络卡	<b>ミ)</b> ,请使用	
	GET_offset_sc	ope 函式来分段	式读取。			
适用范围	0 : 支援 X :	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	

	宝元	工研院				
	0	0				
C#程序范例	StructMsg.of	fset_all _off	set_all;			
	StructMsg.Pw	d _Pwd;				
	private void	button2_Clic	k(object sen	der, EventArg	gs e)	
	{					
	_Pwd.ConnectionKey = "123";					
	<pre>short ret = iRemoting.GET_offset_all(_Pwd, ref _offset_all);</pre>					
	if (ret == 0)					
	{					
	txtLG	eom.Text = _c	offset_all.Of	fsetData[0][	0].ToString()	);
	txtLh	ear.Text = _c	offset_all.Of	fsetData[0][	1].ToString()	);
	txtRG	eom.Text = _c	offset_all.Of	fsetData[0][	2].ToString()	);
	txtRh	ear.Text = _c	offset_all.Of	fsetData[0][	3].ToString()	);
	}					
	}					

# 3. GET\_offset\_scope: 依指定范围取得刀具 Offset

API 名称	GET_offset_scope					
函式说明	依指定范围取得入	依指定范围取得刀具 Offset				
C#方法	short GET_offs	set_scope(St	ructMsg.Pwd	P, ref Struct	Msg.offset_sc	ope
	R);					
<b>VB.Net</b> 方法	GET_offset_sco	ope(ByVal P	As StructMs	g.Pwd, ByRef F	R As	
	StructMsg.off:	set_scope) A	s Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> 词	青参考第三章 <b>)</b> 。		
	结构名称: offs	et_scope				_
	变量名称	型别	[in/out]	说明		
	StartNumber	short	in	起始号码(从1开始)		
	EndNumber	short	in	终止号码		
	OffsetData	OffsetData double[][] out 补正资料				
备注	OffsetData 第二	二个数组,请配	合标题名称 <b>(</b> l	ength Geom)。		
适用范围	0 : 支援 X :	未支援 ?:ㅋ	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	0	0 0 0				
C#程序范例	StructMsg.offs	et_scope _off	<pre>set_scope;</pre>			
	StructMsg.Pwd	_Pwd;				

```
private void button7_Click(object sender, EventArgs e)
{
   _offset_scope.StartNumber = 1;
   _offset_scope.EndNumber = 26;
   _Pwd.ConnectionKey = "123";
   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++)</pre>
       {
           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: 取得单笔刀具 Offset

API 名称	GET_offset_single				
函式说明	取得单笔刀具 Offset				
C#方法	short GET_offset_single(StructMsg.Pwd P, ref				
	<pre>StructMsg.offset_single R);</pre>				
<b>VB.Net</b> 方法	GET_offset_single(ByVal P As StructMsg.Pwd, ByRef R As				
	StructMsg.offset_single) As Short				
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。				
	结构名称: offset_single				

	变量名称	型别	[in/out]	说明		
	ofNumber	short	in	设定补正编号		
	OffsetData	double[]	out	补正资料		
备注						
适用范围	0:支援 X:	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.offs	et_single _	offset_sin	gle;		
	StructMsg.Pwd _Pwd;					
	<pre>private void button4_Click(object sender, EventArgs e)</pre>					
	{	{				
	_Pwd.Conne	_Pwd.ConnectionKey = "123";				
	_offset_si	_offset_single.ofNumber = 1; //Set offset number				
	<pre>short ret = iRemoting.GET_offset_single(_Pwd, ref</pre>					
	_offset_single	_offset_single);				
	if (ret ==	0)				
	{					
	txtLGe	om2.Text = _	offset_sin	gle.OffsetData	[0].ToString(	);
	txtLWea	ar2.Text = _	offset_sin	gle.OffsetData	[1].ToString(	);
	txtRGe	om2.Text = _	offset_sin	gle.OffsetData	[2].ToString(	);
	txtRWea	ar2.Text = _	offset_sin	gle.OffsetData	[3].ToString(	);
	}					
	}					

## 5. SET\_offset\_all: 写入所有刀具 Offset

API 名称	SET_offset_all				
函式说明	写入所有刀具 Of	写入所有刀具 Offset			
C#方法	<pre>short SET_offset_all(StructMsg.Pwd P, StructMsg.offset_all R);</pre>				
<b>VB.Net</b> 方法	SET_offset_all(ByVal P As StructMsg.Pwd, ByRef R As				
	StructMsg.offset_all) As Short				
数据结构	结构名称: Pwd,	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: offse	et_all			
	变量名称	变量名称 型别 [in/out] 说明			
	OffsetData	OffsetData double[][] in 设定补正数据			
备注	※ 必须要设定 Al	PI 写入密码(Pw	vd.WritePwc	l),请参阅「四、API 联机及写入密码」。	

适用范围	0 : 支援 X : 未支援 ? : 未实作					
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.of	fset_all _off	set_all;			
	StructMsg.Pwg	d _Pwd;				
	private void	private void button5_Click(object sender, EventArgs e)				
	{					
	_Pwd.Conn	_Pwd.ConnectionKey = "123";				
	short ret	<pre>short ret = iRemoting.GET_offset_all(_Pwd, ref _offset_all);</pre>				
	_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 = iRe	<pre>ret = iRemoting.SET_offset_all(_offset_all);</pre>				
	if (ret =	if (ret == 0)				
	Messa	<pre>MessageBox.Show("Write success!");</pre>				
	else	else				
	Messa	geBox.Show("W	Vrite failed!	");		
	}					

# 6. SET\_offset\_single: 写入单笔刀具 Offset

API 名称	SET_offset_si	SET_offset_single				
函式说明	写入单笔刀具 01	fset				
C#方法	short SET_off	set_single(S	tructMsg.Pu	wd P, StructMs	g.offset_sing	le
	R);	R);				
<b>VB.Net</b> 方法	SET_offset_si	SET_offset_single(ByVal P As StructMsg.Pwd, ByRef R As				
	StructMsg.off	StructMsg.offset_single) As Short				
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: offset_single					
	变量名称	型别	[in/out]	说明		
	ofNumber	short	in	设定补正编号		
	OffsetData	double[]	in	设定补正数据		
备注	※ 必须要设定	※ 必须要设定 API 写入密码(Pwd.WritePwd),请参阅「四、API 联机及写入密码」。				
适用范围	0:支援 X:未支援 ?:未实作					
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	;	0	

	宝元	工研院				
	0	0				
C#程序范例	StructMsg.of	fset_single _	offset_single	e;		
	StructMsg.Pwo	d _Pwd;				
	private void	<pre>private void button6_Click(object sender, EventArgs e)</pre>				
	{	[				
	_Pwd.Conr	_Pwd.ConnectionKey = "123";				
	_offset_s	ingle.ofNumb	er = 1;			
	_offset_s	_offset_single.OffsetData = new double[4];				
	_offset_s	_offset_single.OffsetData[0] = double.Parse(txtLGeom4.Text);				
	_offset_s	_offset_single.OffsetData[1] = double.Parse(txtLWear4.Text);				
	_offset_s	_offset_single.OffsetData[2] = double.Parse(txtRGeom4.Text);				
	_offset_s	_offset_single.OffsetData[3] = double.Parse(txtRWear4.Text);				
	short ret	: = iRemoting	.SET_offset_s	ingle(_Pwd,	_offset_singl	le);
	if (ret =	= 0)				
	Messa	geBox.Show("W	Vrite success	!");		
	else					
	Messa	geBox.Show("W	Vrite failed!	");		
	}					

## 7. GET\_offset\_count: 取得刀具 Offset 笔数

API 名称	GET_offset_count					
函式说明	取得刀具 Offse	t笔数				
C#方法	short GET_off	set_count(St	ructMsg.Pw	d P, ref Struct	tMsg.total_co	unt
	R);					
VB.Net 方法	GET_offset_co	GET_offset_count(ByVal P As StructMsg.Pwd, ByRef R As				
	StructMsg.tot	al_count) As	Short			
数据结构	结构名称: Pwd,	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。				
	结构名称: tota	结构名称: total_count				
	变量名称 型别 [in/out] 说明					
	Count         short         out         补正笔数					
备注						
适用范围	0 : 支援 X :	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	0	0				

C#程序范例	<pre>StructMsg.total_count _total_count;</pre>
	StructMsg.Pwd _Pwd;
	<pre>private void button3_Click(object sender, EventArgs e)</pre>
	{
	_Pwd.ConnectionKey = "123";
	<pre>short ret = iRemoting.GET_offset_count(_Pwd, ref _total_count);</pre>
	if (ret == 0)
	{
	<pre>txtOffsetCount.Text = _total_count.Count.ToString();</pre>
	}
	}

# 8. GET\_tool\_title: 取得刀具管理的标题栏

API 名称	GET_tool_title						
函式说明	取得刀具管理的机	取得刀具管理的标题栏					
C#方法	short GET_too	l_title(Stru	ctMsg.Pwd I	P, r	ref StructMs	g.tool_title	R);
VB.Net 方法	GET_tool_titl	e(ByVal P As	StructMsg	.Pw	d, ByRef R A	ls	
	StructMsg.too	l_title) As	Short				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参	送考第三章 <b>)</b> 。		
	结构名称: tool	_title					_
	变量名称	型别	[in/out]	说	明		
	ToolTitle	<pre>string[]</pre>	out	标	题栏		
备注							
适用范围	0 : 支援 X :	0 : 支援 X : 未支援 ? : 未实作					
	FANUC	三菱	海德汉		西门子	新代	
	x	Х	0		?	Х	
	宝元	工研院					
	x	Х					
C#程序范例	StructMsg.tool	_title _tool	_title;				
	StructMsg.Pwd	_Pwd;					
	private void b	utton1_Click	(object sen	der	, EventArgs	e)	
	{						
	_Pwd.Conne	_Pwd.ConnectionKey = "123";					
	short ret :	<pre>short ret = iRemoting.GET_tool_title(_Pwd, ref _tool_title);</pre>					
	if (ret ==	if (ret == 0)					
	{						
	for (ir	nt i = 0; i <	= _tool_tit	:le.	ToolTitle.Le	ngth - 1; i++	)
	{						
	rtb	Title.Text +	= _tool_tit	le.	ToolTitle[i]	+ "\r\n";	

}
}
}

## 9. GET\_tool\_count: 取得刀具 Offset 笔数

API 名称	GET_tool_count	t					
函式说明	取得刀具管理笔数	取得刀具管理笔数					
C#方法	short GET_tool	<pre>short GET_tool_count(StructMsg.Pwd P, ref StructMsg.total_count R);</pre>					
VB.Net 方法	GET_tool_count(ByVal P As StructMsg.Pwd, ByRef R As						
	StructMsg.tota	al_count) As	Short				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参	参考第三章 <b>)</b> 。		
	结构名称: tota	l_count					
	变量名称	型别	[in/out]	说	.明		
	Count	short	out	刀	具笔数		
备注							
适用范围	0 : 支援 X :	未支援 ?:	未实作				
	FANUC	三菱	海德汉		西门子	新代	
	х	Х	0		?	Х	
	宝元	工研院					
	Х	Х					
C#程序范例	StructMsg.tota	1_count _tota	al_count;				
	StructMsg.Pwd	_Pwd;					
	private void b	utton2_Click	(object sen	der	, EventArgs	e)	
	{						
	_Pwd.Connec	ctionKey = "1	.23";				
	short ret =	= iRemoting.0	ET_tool_cou	unt	(_Pwd, ref _t	cotal_count);	
	if (ret ==	0)					
	{						
	Message	Box.Show(_to	tal_count.C	Coun	nt.ToString()	);	
	}						
	}						

## 10. GET\_tool\_data: 取得所有刀具管理数据

API 名称	GET_tool_data
函式说明	取得所有刀具管理数据
C#方法	<pre>short GET_tool_data(StructMsg.Pwd P, ref StructMsg.tool_data R);</pre>
VB.Net 方法	GET_tool_data(ByVal P As StructMsg.Pwd, ByRef R As

	StructMsg.too	l_data) As SI	nort			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> ì	青参考第三章)。		
	结构名称: tool	_data				
	变量名称	型别	[in/out]	说明		
	Data	<pre>string[][]</pre>	out	刀具管理数据		
备注	※ Data 第二个教	<b>数组,请配合标</b>	题名称。			
适用范围	0 : 支援 X :	未支援 ?: 考	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	x	Х	0	?	Х	
	宝元	工研院				
	x	Х				
C#程序范例	StructMsg.tool	_data _tool_d	lata;			
	StructMsg.Pwd	_Pwd;				
	private void b	utton3_Click(	object send	der, EventArgs	e)	
	{					
	_Pwd.Conne	ctionKey = "1	23";			
	short ret	= iRemoting.G	ET_tool_dat	a(_Pwd, ref _to	ool_data);	
	if (ret ==	0)				
	{					
		ole dtlool = r	new Datalab	le();		、
	for (ir	1t 1 = 0; 1 <=	= _tool_tit	le.loollitle.Le	ength - 1; 1++	)
	1 1	ool Columns A	dd( tool ti		÷1	
		001.0010018.4	Type (	GetType("System	⊥], String"))·	
	3		Typere		. Stilling //,	
	,					
	for (ir	nti=0;i<=	= _tool_dat	a.Data.Length -	· 1; i++)	
	{					
	Dat	aRow dr = dtT	ool.NewRow(	();		
	for	(int j = 0; ;	j <= _tool_	title.ToolTitle	.Length - 1; j	j++)
	{					
		dr[j] = _too	l_data.Data	[i][j];		
	}					
	dtT	ool.Rows.Add(	dr);			
	}					
	dgvToo]	.DataSource =	= dtTool;			
	}					
	}					

API 名称	GET_tool_data	_scope				
函式说明	刀具管理数据,并	刀具管理数据,并依指定索引值范围传回				
C#方法	short GET_too	L_data_scope	(StructMsg.	Pwd P, ref		
	StructMsg.too	<pre>StructMsg.tool_data_scope R);</pre>				
VB.Net 方法						
数据结构	结构名称: Pwd,	联机密码及写》	\密码设置 <b>(</b> 词	青参考第三章 <b>)</b> 。		
	结构名称: tool	_data_scope				
	变量名称	型别	[in/out]	说明		
	StartIndex	int	in	起始索引值		
	EndIndex	int	in	结束索引值		
	Data	<pre>string[][]</pre>	out	刀具管理数据		
备注						
适用范围	0 : 支援 X :	未支援 ?:オ	<b></b> 、实作			
	FANUC	三菱	海德汉	西门子	新代	
	x	Х	0	?	X	
	宝元	工研院				
	Х	Х				
C#程序范例	StructMsg.tool	_data_scope _	DataScope;			
	StructMsg.Pwd _Pwd;					
	<pre>for (int i = 0; i &lt;= _ToolMaxIndex.MaxIndex; i += 20)</pre>					
	{					
	//一次读取20笔					
	_DataScope.StartIndex = i;					
	if ((i + 20	)) <= _ToolMa:	xIndex.MaxI	ndex)		
	_DataSc	ope.EndIndex	= i + 20 -	1;		
	else		T 1M	T		
	_Datasc	ope.Endindex	= _1001Max.	index.Maxindex;		
	_Pwd.Conned	tionKey = "1	23";			
	ret = iRemo	oting.GET_too	l_data_scop	e(_Pwd, ref _Da	ataScope);	
	if (ret ==	0)				
	{					
	for (in	t j = 0; j <=	DataScop	e.Data.Length -	1; j++)	
	{					
	Dat	aRow dr = dtT	ool.NewRow(	();		
	for	(int s = 0; s	s <= _tool_t	title.ToolTitle	.Length - 1; s++)	

11. GET\_tool\_data\_scope: 取刀具管理数据,并依指定索引值范围传回

{
<pre>dr[s] = _DataScope.Data[j][s];</pre>
}
dtTool.Rows.Add(dr);
}
}
}

## 12. SET\_tool\_data: 写入刀具管理

API 名称	SET_tool_data	I				
函式说明	写入刀具管理	写入刀具管理				
C#方法	<pre>short SET_tool_data(StructMsg.Pwd P, StructMsg.tool_data R);</pre>					
VB.Net 方法	SET_tool_data(ByVal P As StructMsg.Pwd, ByVal R As					
	StructMsg.too	l_data) As S	hort			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> 请	青参考第三章 <b>)</b> 。		
	结构名称: tool	_data	,			_
	变量名称	型别	[in/out]	说明		
	Data	<pre>string[][]</pre>	in	设定刀具管理数	据	
备注	※ 可针对某几笔	笔数据或全部数据	居,进行写入9	更新。		
	※ 针对某几笔数	牧据更新时,第[	[][0]个数组可	丁指定编号, 如范	例。	
	※ 必须要设定	API 写入密码(P	wd.WritePwd	),请参阅「四、AP	I 联机及写入密码	码」。
适用范围	0:支援 X:	未支援 ?:	未实作			-
	FANUC	三菱	海德汉	西门子	新代	
	x	Х	0	?	Х	
	宝元	工研院				
	х	Х				
C#程序范例	StructMsg.Pwd	_Pwd;				
	private void b	outton4_Click	(object send	er, EventArgs	e)	
	{					
	_tool_data	.Data = new s	tring[1][];			
	_tool_data	.Data[0] = ne	w string[_to	ool_title.Tool1	<pre>[itle.Length];</pre>	,
	DataTable	dtTool = (Dat	aTable)dgvTo	pol.DataSource;	;	
	for (int i	= 0; i <= _t	ool_title.To	polTitle.Length	ı - 1; i++)	
	{					
	_tool_	data.Data[0][	i] = dtTool.	Rows[0][i].ToS	tring();	
	}					
	_Pwd.Conne	ctionKey = "1	.23";			

_Pwd.WritePwd = "123";
<pre>short ret = iRemoting.SET_tool_data(_Pwd, _tool_data);</pre>
}

## 13. GET\_pocket\_title: 取得刀库表的标题栏

API 名称	GET_pocket_ti	GET_pocket_title				
函式说明	取得刀库表的标	取得刀库表的标题栏				
<b>C#</b> 方法	short GET_poc	<pre>short GET_pocket_title(StructMsg.Pwd P, ref StructMsg.pocket_title</pre>				
	R);					
VB.Net 方法	GET_pocket_ti	tle(ByVal P	As StructM	sg.Pwd, ByRef	R As	
	StructMsg.poc	ket_title) A	s Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: pock	结构名称: pocket_title				
	变量名称	型别	[in/out]	说明		
	PocketTitle	string[]	out	标题栏		
备注						
适用范围	0 : 支援 X :	未支援 ?::	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	x	Х	0	?	X	
	宝元	工研院				
	x	Х				
C#程序范例	StructMsg.pock	<pre>xet_title _po</pre>	cket_title;			
	StructMsg.Pwd	_Pwd;				
	private void b	outton1_Click	(object sen	der, EventArgs	e)	
	{					
	_Pwd.Conne	ctionKey = "1	L23";			
	short ret	= iRemoting.(	GET_pocket_1	title(_Pwd, ref	_pocket_title	<u>;</u> );
	if (ret ==	0)				
	{					
	for (ir	nti=0;i<=	<pre>pocket_ti</pre>	tle.PocketTitl	e.Length - 1; i	Ĺ++)
	{					
	rtb	Title.Text +	= _pocket_t	itle.PocketTit]	.e[i] + "\r\n"	;
	}					
	}					
	}					

14. GET\_pocket\_count: 取得刀库表笔数

API 名称	GET_pocket_count
函式说明	取得刀库表笔数

C#方法	short GET_poc	ket_count(St	ructMsg.Pw	d P, ref Struc <sup>.</sup>	tMsg.total_co	unt
	R);	R);				
VB.Net 方法	GET_pocket_co	GET_pocket_count(ByVal P As StructMsg.Pwd, ByRef R As				
	StructMsg.tot	al_count) As	Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
	结构名称: tota	l_count				
	变量名称	型别	[in/out]	说明		
	Count	short	out	刀具笔数		
备注						
适用范围	0 : 支援 X :	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	x	Х	0	?	Х	
	宝元	工研院				
	X	Х				
C#程序范例	StructMsg.tota	l_count _tot	al_count;			
	StructMsg.Pwd	_Pwd;				
	private void b	utton2_Click	(object sen	der, EventArgs	e)	
	{					
	_Pwd.Connec	ctionKey = "1	.23";			
	short ret :	= iRemoting.0	GET_pocket_	count(_Pwd, ref	_total_count)	);
	if (ret ==	0)				
	{					
	Message	Box.Show(_to	tal_count.C	Count.ToString()	);	
	}					
	}					

## 15. GET\_pocket\_data: 取得刀库表资料

API 名称	GET_pocket_dat	a		
函式说明	取得刀库表资料			
C#方法	short GET_pock	et_data(Stru	ctMsg.Pwd P	<pre>P, ref StructMsg.pocket_data R);</pre>
<b>VB.Net</b> 方法	GET_pocket_data(ByVal P As StructMsg.Pwd, ByRef R As			
	StructMsg.pocket_data) As Short			
数据结构	结构名称: Pwd,	联机密码及写)	\密码设置 <b>(</b> i	青参考第三章)。
	结构名称: pocke	et_data		
	变量名称	型别	[in/out]	说明
	Data	string[][]	out	刀具管理数据
备注	※ Data 第二个数	y组,请配合标	题名称。	

适用范围	0:支援 X:	未支援 ?::	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	x	Х	0	?	Х		
	宝元	工研院					
	Х	Х					
C#程序范例	StructMsg.poc	ket_data _poc	ket_data;				
	StructMsg.Pwd	_Pwd;					
	private void	button3_Click	(object sende	r, EventArgs o	e)		
	{						
	_Pwd.Conne	ectionKey = "1	L23";				
	short ret	= iRemoting.(	GET_pocket_dat	a(_Pwd, ref _	_pocket_data);		
	if (ret =:	= 0)					
	{						
	<pre>DataTable dtPocket = new DataTable();</pre>						
	<pre>for (int i = 0; i &lt;= _pocket_title.PocketTitle.Length - 1; i++)</pre>						
	{						
	dt	Pocket.Column	s.Add(_pocket_	_title.Pocket <sup>-</sup>	Title[i],		
	Type.GetType(	"System.Strin	g"));				
	}						
	for (i	.nt i = 0; i <	= _pocket_dat	a.Data.Length	- 1; i++)		
	{						
	Da	taRow dr = dt	Pocket.NewRow	();			
	fo	r (int j = 0;	j <= _pocket_	_title.Pocket	Title.Length -	- 1;	
	j++)						
	{						
		dr[j] = _poo	:ket_data.Data	[i][j];			
	}						
	dt	Pocket.Rows.A	dd(dr);				
	}						
	dgvToc	J.DataSource	= dtPocket;				
	}						
	}						

16. SET\_pocket\_data: 写入刀库表

API 名称	SET_pocket_data
函式说明	写入刀库表
C#方法	<pre>short SET_pocket_data(StructMsg.Pwd P, StructMsg.pocket_data R);</pre>

VB.Net 方法	SET_pocket_data(ByVal P As StructMsg.Pwd, ByVal R As						
	StructMsg.pocl	ket_data) As	Short				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> 请	青参考第三章 <b>)</b> 。			
	结构名称: pocke	et_data					
	变量名称     型别     [in/out]     说明						
	Data	<pre>string[][]</pre>	in	设定刀具管理数	据		
备注	※ 可针对某几笔	数据或全部数据	居,进行写入§	更新。			
	※ 针对某几笔数	[据更新时,第]	][0]个数组可	J指定编号,如范	例。		
	※ 必须要设定 A	PI 写入密码(P	wd.WritePwd	),请参阅「四、AP	I 联机及写入密	码」。	
适用范围	0:支援 X:	未支援 ?: 🕫	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	х	Х	0	?	Х		
	宝元	工研院					
	Х	Х					
C#程序范例	_pocket_data.D	ata = new str	ring[1][];				
	_pocket_data.D	ata[0] = new	string[_poc	ket_title.Pock	etTitle.Lengt	h];	
	DataTable dtTo	ol = (DataTał	ole)dgvTool.	DataSource;			
	for (int i = 0	; i <= _pocke	et_title.Poc	ket⊺itle.Lengt	h - 1; i++)		
	{						
	_pocket_dat	ta.Data[0][i]	= dtTool.Ro	ows[0][i].ToSt	ring();		
	}						
	StructMsg.Pwd	_Pwd;					
	_Pwd.Connectio	nKey = "123";	;				
	_Pwd.WritePwd	= "123";					
	short ret = iR	emoting.SET_p	ocket_data(	_pocket_data);			

# 17. GET\_pocket\_data\_scope: 取刀库表数据,并依指定索引值范围传回

API 名称	GET_pocket_dat	ta_scope			
函式说明	刀库表数据,并依	刀库表数据,并依指定索引值范围传回			
C#方法	<pre>short GET_pocket_data_scope(StructMsg.Pwd P, ref</pre>				
	StructMsg.too]	<pre>StructMsg.tool_data_scope R);</pre>			
VB.Net 方法					
数据结构	结构名称: Pwd,	联机密码及写入	\密码设置 <b>(</b> i	青参考第三章)。	
	结构名称: pocke	et_data_scope	2		
	变量名称	型别	[in/out]	说明	
	StartIndex	int	in	起始索引值	
	EndIndex	int	in	结束索引值	

Data	string[][]	out	刀具管理数据		
0:支援 X:	未支援 ?:ㅋ	未实作			
FANUC	三菱	海德汉	西门子	新代	
х	Х	0	?	Х	
宝元	工研院				
Х	Х				
StructMsg.pock	et_data_scope	_DataScope	;		
StructMsg.Pwd	_Pwd;				
for (int $i = 0$	; i <= _Pocke	etMaxIndex.M	laxIndex; i += 1	20)	
{					
//一次读取20	9笔				
_DataScope.	StartIndex =	i;			
if ((i + 20	) <= _Pocket	MaxIndex.Ma	xIndex)		
_DataSc	ope.EndIndex	= i + 20 -	1;		
else					
_DataSc	ope.EndIndex	= _PocketMa	axIndex.MaxInde	x;	
_Pwd.Connec	tionKey = "1	23";			
ret = iRemo	oting.GET_poc	ket_data_sc	ope(_Pwd, ref _	DataScope);	
if (ret ==	0)				
{					
for (in	t j = 0; j <=	= _DataScope	e.Data.Length -	1; j++)	
{					
Data	aRow dr = dtT	ool.NewRow(	);		
for (	int s = 0; s <	= _pocket_ti	tle.PocketTitl	e.Length - 1; s	;++)
{					
,	dr[s] = _Dat	aScope.Data	[]][s];		
}		· .			
מדומ	JOT . KOMS . AQQ (	ur);			
ڑ ر					
}					
	Data         0:支援 X::         FANUC         X         宝元         X         宝元         X         StructMsg.pock         StructMsg.pock         StructMsg.Pwd         for (int i = 0)         {         //-次读取20         _DataScope         if ((i + 20)         _DataScope         if (ret ==         {         for (in         {         for (in         {         for (in         {         j         dtTox         }         }         }         }         j         j         j         j         j         j         j         j         j <th>Data       string[][]         0:支援 X:未支援 ?:克         FANUC       三菱         X       X         宝元       工研院         X       X         宝元       工研院         X       X         StructMsg.pocket_data_scope         StructMsg.Pwd _Pwd;         for (int i = 0; i &lt;= _Pocket        DataScope.StartIndex =         if ((i + 20) &lt;= _Pocket        DataScope.EndIndex         else        DataScope.EndIndex         else         _DataScope.EndIndex         else         _DataScope.EndIndex         else         _DataScope.EndIndex         _Pwd.ConnectionKey = "1         ret = iRemoting.GET_poc         if (ret == 0)         {         for (int j = 0; j &lt;=         {         DataRow dr = dtT         for (int s = 0; s &lt;         {         dtTool.Rows.Add(         }       }</th> <th>Data       string[][]       out         0 : 支援 X : 未支援 ? : 未实作         FANUC       三菱       海德汉         X       X       0         宝元       工研院       二         X       X       0         宝元       工研院       二         X       X       0         宝元       工研院       二         X       X       0         StructMsg.pocket_data_scope       DataScope         StructMsg.Pwd _Pwd;       for (int i = 0; i &lt;= _PocketMaxIndex.Ma         _DataScope.StartIndex = i;       if ((i + 20) &lt;= _PocketMaxIndex.Ma         _DataScope.EndIndex = i + 20 -       else         _DataScope.EndIndex = i + 20 -       else         _DataScope.EndIndex = _PocketMax       _Pwd.ConnectionKey = "123";         ret = iRemoting.GET_pocket_data_scope       if (ret == 0)         {       for (int j = 0; j &lt;= _DataScope.data_scope         {       DataRow dr = dtTool.NewRow(         for (int s = 0; s &lt;= _pocket_tii       {         {       dr[s] = _DataScope.Data       }         dtTool.Rows.Add(dr);       }       }</th> <th>Data       string[][]       out       刀具管理数据         0:支援 X:未支援 ?:未实作         FANUC       三菱       海德汉       西门子         X       X       0       ?         宝元       工研院       二      </th> <th>Data       string[][]       out       刀具管理数据         0: 支援 X: 未支援 ?: 未实作</th>	Data       string[][]         0:支援 X:未支援 ?:克         FANUC       三菱         X       X         宝元       工研院         X       X         宝元       工研院         X       X         StructMsg.pocket_data_scope         StructMsg.Pwd _Pwd;         for (int i = 0; i <= _Pocket        DataScope.StartIndex =         if ((i + 20) <= _Pocket        DataScope.EndIndex         else        DataScope.EndIndex         else         _DataScope.EndIndex         else         _DataScope.EndIndex         else         _DataScope.EndIndex         _Pwd.ConnectionKey = "1         ret = iRemoting.GET_poc         if (ret == 0)         {         for (int j = 0; j <=         {         DataRow dr = dtT         for (int s = 0; s <         {         dtTool.Rows.Add(         }       }	Data       string[][]       out         0 : 支援 X : 未支援 ? : 未实作         FANUC       三菱       海德汉         X       X       0         宝元       工研院       二         X       X       0         宝元       工研院       二         X       X       0         宝元       工研院       二         X       X       0         StructMsg.pocket_data_scope       DataScope         StructMsg.Pwd _Pwd;       for (int i = 0; i <= _PocketMaxIndex.Ma         _DataScope.StartIndex = i;       if ((i + 20) <= _PocketMaxIndex.Ma         _DataScope.EndIndex = i + 20 -       else         _DataScope.EndIndex = i + 20 -       else         _DataScope.EndIndex = _PocketMax       _Pwd.ConnectionKey = "123";         ret = iRemoting.GET_pocket_data_scope       if (ret == 0)         {       for (int j = 0; j <= _DataScope.data_scope         {       DataRow dr = dtTool.NewRow(         for (int s = 0; s <= _pocket_tii       {         {       dr[s] = _DataScope.Data       }         dtTool.Rows.Add(dr);       }       }	Data       string[][]       out       刀具管理数据         0:支援 X:未支援 ?:未实作         FANUC       三菱       海德汉       西门子         X       X       0       ?         宝元       工研院       二	Data       string[][]       out       刀具管理数据         0: 支援 X: 未支援 ?: 未实作

十一、 工件坐标类别函式

以下表格所列函式皆已过时,不再使用:

已过时的函式

GET\_preset\_data\_mem

#### 1. GET\_work\_coord\_title: 取得工件坐标的标题栏

API 名称	GET_work_coord_	GET_work_coord_title						
函式说明	取得工件坐标的标	取得工件坐标的标题栏						
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 Struc	tMsg.Pwd, ByF	Ref R As			
	StructMsg.work_	coord_title	) As Short					
数据结构	结构名称: Pwd, 联	关机密码及写入	.密码设置 <b>(</b> 请	参考第三章 <b>)</b> 。				
	结构名称: work_c	oord_title						
	变量名称	型别	[in/out]	说明				
	WorkCoordTitle	string[]	out	标题栏				
备注								
适用范围	0:支援 X:未	支援 ?:未	实作					
	FANUC	三菱	海德汉	西门子	新代			
	0	0	Х	?	0			
	宝元	工研院						
	0	0						
C#程序范例	StructMsg.work_	coord_title	_work_coor	d_title;				
	StructMsg.Pwd _	Pwd;						
	private void bu	tton1_Click	(object sen	der, EventArg	gs e)			
	{							
	_Pwd.Connec	tionKey = "1	.23";					
	short ret =	iRemoting.0	iET_work_cod	ord_title(_Pw	ıd, ref			
	_work_coord_tit	le);						
	if (ret == 0	9)						
	{							
	for (int i = 0; i	<= _work_coo	ord_title.Wo	orkCoordTitle	e.Length - 1;	i++)		
	rtbTitle	.Text += _wo	rk_coord_ti	tle.WorkCoor	dTitle[i]+"	";		
	}							
	}							

2. GET\_work\_coord\_all: 取得所有工件坐标数据

API 名称	GET_work_coord_all	
--------	--------------------	--

函式说明	取得所有工件补正数据					
<b>C#</b> 方法	<pre>short GET_work_coord_all(StructMsg.Pwd P, ref</pre>					
	StructMsg.worl	k_coord_all	R);			
VB.Net 方法	GET_work_coord	d_all(ByVal	P As Struct	Msg.Pwd, ByRet	f R As	
	StructMsg.worl	k_coord_all)	As Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> 词	青参考第三章 <b>)</b> 。		
	结构名称:work	_coord_all				
	变量名称	型别	[in/out]	说明		
	CoordName	string[]	out	项目,ex:EXT,G	54,G55	
	WorkCoord	double[][]	out	工件坐标数据		
备注	CoordName 项目	如:				
	EXT, G54, G55	, G56, G57, (	G58, G59 ,	G54.1P1, G54.1	P2, G54.1P3 等	ž
	使用此函式可能会	会有大量的读取	时间(一般都	发生在低速网络卡	<b>ミ)</b> ,请使用	
	GET_work_coord	d_scope 函式来	来分段读取。			
适用范围	0 : 支援 X :	未支援 ?: ラ	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.worl	k_coord_all _	_work_coord	_all;		
	StructMsg.Pwd	_Pwd;				
	private void l	button2_Clic	k(object se	nder, EventAr	gs e)	
	{					
	_Pwd.Conne	ectionKey = "	'123" <b>;</b>			
	short ret	= iRemoting.	GET_work_c	oord_all(_Pwd,	ref	
	_work_coord_a	11);				
	if (ret ==	= 0)				
	{					
	//EXT	Work Coord				
	txtIte	m.Text = _wo	rk_coord_a	<pre>L1.CoordName[0</pre>	];	
	txtX.T	ext = _work_	coord_all.N	NorkCoord[0][0	].ToString();	
	txtY.T	ext = _work_	coord_all.N	NorkCoord[0][1	].ToString();	
	txtZ.T	ext = _work_	coord_all.N	VorkCoord[0][2	].ToString();	
	}					
	}					

# 3. GET\_work\_coord\_scope: 依指定范围取得工件坐标数据

API 名称	GET_work_coord_scope
函式说明	依指定范围取得工件补正数据

C#方法	short GET_work	<pre>short GET_work_coord_scope(StructMsg.Pwd P, ref</pre>				
	StructMsg.work	<pre>StructMsg.work_coord_scope R);</pre>				
VB.Net 方法	GET_work_coord	GET_work_coord_scope(ByVal P As StructMsg.Pwd, ByRef R As				
	StructMsg.work	_coord_scop	e) As Short	:		
数据结构	结构名称: Pwd,	联机密码及写。	入密码设置 <b>(</b> i	青参考第三章 <b>)</b> 。		
	结构名称: work_	_coord_scope			-	
	变量名称	型别	[in/out]	说明		
	StartNumber	short	in	起始编号(从1开始)		
	EndNumber	short	in	终止编号		
	CoordName	string[]	out	项目,ex:EXT,G54,G55…		
	WorkCoord	double[][]	out	工件坐标数据		
备注	● FANUC 控制	器及三菱控制器	R F		-	
	CoordName 项	目如:				
	EXT, G54, G5	55, G56, G57	, G58, G59	,G54.1P1,G54.1P2,G54.1P3	\$等	
	依指定编号对应	应至工件坐标名	称如下列范围	<b>邦</b> :		
	工件名称	对应编	号			
	EXT	1				
	G54	2				
	G55	3				
	G56	4				
	G57	5				
	G58	6				
	G59	7				
	G54.1 P1~Pr	n 8~n				
	● 新代控制器					
	CoordName 項	目如:				
	EXT,MPGShift	t, G54, G55,	G56,G57,G	558, G59 , G54P7, G54P8, G54P	100	
	等					
	依指定編號對M	應至工件座標名	I稱如下列範圍			
	工件名称	对应编	号			
	EXT	0				
	MPG Shift	1				
	G54	2				
	G55	3				
	G56	4				
	G57	5				
	G58	6				
	G59	7				
	G54 P7~P100	8~101				

适用范围	0:支援 X:	未支援 ?::	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	0	Х	?	0		
	宝元	工研院					
	0	0					
C#程序范例	StructMsg.wor	k_coord_scope	_work_coord_	scope;			
	StructMsg.Pwd	_Pwd;					
	private void	button7_Click	(object sende	r, EventArgs	e)		
	{						
	_work_coor	rd_scope.Start	Number = 1;				
	_work_coord_scope.EndNumber = 8;						
	_Pwd.Conne	ectionKey = "1	L23";				
	short ret	= iRemoting.(	GET_work_coord	l_scope(_Pwd,	ref		
	_work_coord_s	cope);					
	if (ret ==	= 0)					
	{						
	//EXT	Work Coord					
	txtIte	m.Text = _wor	k_coord_scope	.CoordName[0]	;		
	txtX.T	ext = _work_c	oord_scope.Wo	rkCoord[0][0]	.ToString();		
	txtY.T	ext = _work_c	oord_scope.Wo	rkCoord[0][1]	.ToString();		
	txtZ.T	ext = _work_c	oord_scope.Wo	rkCoord[0][2]	.ToString();		
	}						
	}						

## 4. GET\_work\_coord\_single: 取得单笔工件坐标资料

API 名称	GET_work_coord	l_single			
函式说明	取得单笔工件坐标	示资料			
<b>C#</b> 方法	short GET_work	_coord_sing	le(StructM	sg.Pwd P, ref	
	StructMsg.work	_coord_sing	le R);		
VB.Net 方法	GET_work_coord	GET_work_coord_single(ByVal P As StructMsg.Pwd, ByRef R As			
	StructMsg.work	<pre>StructMsg.work_coord_single) As Short</pre>			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。	
	结构名称:work_	_coord_singl	e		
	变量名称	型别	[in/out]	说明	
	CoordName	string	in	项目,ex:EXT,G54,G55	
	WorkCoord	double[]	out	工件坐标数据	
备注	CoordName 项目	如:			
	EXT, G54, G55,	G56, G57,	G58, G59 ,	G54.1P1, G54.1P2, G54.1P3等	:

适用范围	0: 支援 X: 未支援 ?: 未实作					
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	]
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.wo	rk_coord_sing	le _work_coo	rd_single;		
	StructMsg.Pwo	d _Pwd;				
	private void	button4_Clic	k(object sen	der, EventArg	gs e)	
	{					
	_work_coc	ord_single.Co	ordName = "G	55";		
	_Pwd.Conr	ectionKey =	"123";			
	short ret	: = iRemoting	.GET_work_cod	ord_single(_P	wd, ref	
	_work_coord_:	single);				
	if (ret =	= 0)				
	{					
	//EXT	Work Coord				
	txtIt	em2.Text = _w	vork_coord_si	ngle.CoordNa	me;	
	txtX2	.Text = _work	coord_singl	e.WorkCoord[0	0].ToString()	);
	txtY2	.Text = _work	coord_singl	e.WorkCoord[:	1].ToString()	);
	txtZ2	.Text = _work	coord_singl	e.WorkCoord[2	2].ToString()	);
	}					
	}					

# 5. SET\_work\_coord\_all: 写入所有工件坐标数据

API 名称	SET_work_coord_all					
函式说明	写入所有工件坐标数据					
C#方法	<pre>short SET_work_coord_all(StructMsg.Pwd P, StructMsg.work_coord_all</pre>					
	R);					
<b>VB.Net</b> 方法	SET_work_coord_all(ByVal P As StructMsg.Pwd, ByVal R As					
	StructMsg.work_coord_all) As Short					
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: work_coord_all					
	变量名称         型别         [in/out]         说明					
	CoordName string[] in 设定项目,ex:EXT,G54,G55					
	WorkCoord	double[][]	in	设定工件坐标数据		
备注	CoordName 设定项目如:					
	EXT, G54, G55, G56, G57, G58, G59 , G54.1P1, G54.1P2, G54.1P3 等					
	※ 必须要设定 API 写入密码(Pwd.WritePwd),请参阅「四、API 联机及写入密码」。					

适用范围	0 : 支援 X : 未支援 ? : 未实作							
	FANUC	三菱	海德汉	西门子	新代			
	0	0	Х	?	0			
	宝元	工研院						
	0	0						
C#程序范例	StructMsg.wo	rk_coord_all	_work_coord_a	all;				
	StructMsg.Pwd _Pwd;							
	private void button5_Click(object sender, EventArgs e)							
	{							
	_Pwd.ConnectionKey = "123";							
	<pre>short ret = iRemoting.GET_work_coord_all(_Pwd, ref</pre>							
	_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);							
	}							
	<pre>ret = iRemoting.SET_work_coord_all(_work_coord_all);</pre>							
	if (ret == 0)							
	<pre>MessageBox.Show("Write success!");</pre>							
	else							
	<pre>MessageBox.Show("Write failed!");</pre>							
	}							

# 6. SET\_work\_coord\_single: 写入单笔工件坐标数据

API 名称	SET_work_coord_single						
函式说明	写入单笔工件坐标数据						
C#方法	<pre>short SET_work_coord_single(StructMsg.Pwd P,</pre>						
	<pre>StructMsg.work_coord_single R);</pre>						
<b>VB.Net</b> 方法	SET_work_coord_single(ByVal P As StructMsg.Pwd, ByVal R As						
	StructMsg.work_coord_single) As Short						
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。						
	结构名称: work_coord_single						
	变量名称	型别	[in/out]	说明			
	CoordName	string	in	设定项目,ex:EXT,G54,G55			
	WorkCoord	double[]	in	设定工件坐标数据			
备注	※ CoordName	※ CoordName 设定项目如:					
--------	---------------	--	---------------	---------------	--------------	-----	--
	EXT, G54, G5	XT, G54, G55, G56, G57, G58, G59 , G54.1P1, G54.1P2, G54.1P3 等					
	≫ WorkCoord	※ WorkCoord 的数组长度,必须等于轴个数。					
	※ 必须要设定	API 写入密码(P	Wd.WritePwd)	,请参阅「四、API	I联机及写入密	码」。	
适用范围	0:支援 X:	未支援 ?::	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	0	Х	?	0		
	宝元	工研院					
	0	0					
C#程序范例	StructMsg.wo	rk_coord_sing	le _work_coo	rd_single;			
	StructMsg.Pw	d _Pwd;					
	private void	button6_Clic	k(object sen	der, EventArg	gs e)		
	{						
	_Pwd.Conr	ectionKey =	"123";				
	_work_coc	ord_single.Co	ordName = "G	57";			
	_work_coc	ord_single.Wo	rkCoord = new	v double[3];			
	_work_coc	ord_single.Wo	rkCoord[0] =	double.Parse	(txtX3.Text)	;	
	_work_coc	ord_single.Wo	rkCoord[1] =	double.Parse	(txtY3.Text)	;	
	_work_coc	ord_single.Wo	rkCoord[2] =	double.Parse	(txtZ3.Text)	;	
	short ret	: = iRemoting	.SET_work_cod	ord_single(_P	wd,		
	_work_coord_:	single);					
	if (ret =	= 0)					
	Messa	geBox.Show("W	Vrite success	!");			
	else						
	Messa	geBox.Show("W	Vrite failed!	");			
	}						

7. GET\_work\_coord\_count: 取得工件坐标笔数

API 名称	GET_work_coord_count			
函式说明	取得工件坐标笔数			
<b>C#</b> 方法	<pre>short GET_work_coord_count(StructMsg.Pwd P, ref</pre>			
	<pre>StructMsg.total_count R);</pre>			
VB.Net 方法	GET_work_coord_count(ByVal P As StructMsg.Pwd, ByRef R As			
	StructMsg.total_count) As Short			
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。			
	结构名称: total_count			
	变量名称     型别     [in/out]     说明			

	Count	short	out	工件坐标笔数				
备注								
适用范围	0 : 支援 X :	O : 支援 X : 未支援 ? : 未实作						
	FANUC	三菱	海德汉	西门子	新代			
	0	0	Х	?	0			
	宝元	工研院						
	0	0						
C#程序范例	StructMsg.tot	al_count _to	tal_count;					
	StructMsg.Pwd	<pre>StructMsg.Pwd _Pwd;</pre>						
	private void	button3_Clic	k(object se	ender, EventArg	gs e)			
	{							
	_Pwd.Conn	ectionKey = '	'123" <b>;</b>					
	short ret	= iRemoting	.GET_work_c	oord_count(_Pw	d, ref			
	_total_count)	;						
	if (ret =	= 0)						
	{							
	txtCou	<pre>int.Text = _t</pre>	otal_count	.Count.ToStrin	g();			
	}							
	}							

# 8. GET\_preset\_title: 取得工作台管理表的标题栏

API 名称	GET_preset_title						
函式说明	取得工作台管理表的标题栏						
C#方法	<pre>short GET_preset_title(StructMsg.Pwd P, ref StructMsg.preset_title</pre>						
	R);	R);					
<b>VB.Net</b> 方法	GET_preset_title(ByVal P As StructMsg.Pwd, ByRef R As						
	StructMsg.pre	set_title) A	s Short				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。			
	结构名称: preset_title						
	变量名称	型别	[in/out]	] 说明			
	PresetName	string	in	工作台管理表名和	弥(PRESET.PR		
				/ PRESET2.PR /	PRESET3.PR)		
	PresetTitle	<pre>string[]</pre>	out	标题栏			
备注							
适用范围	0 : 支援 X :	未支援 ?:	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	х	Х	0	?	Х		
	宝元	工研院					
	X	Х					

C#程序范例	<pre>StructMsg.preset_title _preset_title;</pre>								
	StructMsg.Pwd _Pwd;								
	<pre>private void button1_Click(object sender, EventArgs e)</pre>								
	{								
	_Pwd.ConnectionKey = "123";								
	_preset_title.PresetName = "PRESET.PR";								
	<pre>short ret = iRemoting.GET_preset_title(_Pwd, ref _preset_title);</pre>								
	if (ret == 0)								
	{								
	<pre>for (int i = 0; i &lt;= _preset_title.PresetTitle.Length - 1; i++)</pre>								
	{								
	rtbTitle.Text += _preset_title.PresetTitle[i] + "\r\n";								
	}								
	}								
	}								

9. GET\_preset\_count: 取得工作台管理表笔数

API 名称	GET_preset_cou	unt					
函式说明	取得工作台管理表	取得工作台管理表笔数					
C#方法	short GET_pres	<pre>short GET_preset_count(StructMsg.Pwd P, ref StructMsg.preset_count</pre>					
	R);	R);					
VB.Net 方法	GET_preset_cou	unt(ByVal P	As StructM	sg.Pwd, ByRef	R As		
	StructMsg.pres	set_count) A	s Short				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。			
	结构名称: prese	et_count				_	
	变量名称	型别	[in/out]	说明			
	PresetName	string	in	工作台管理表名称(PRESET.PR			
				/ PRESET2.PR / PRESET3.PR)			
	Count	short	out	工作台可设定总数			
备注							
适用范围	0 : 支援 X :	未支援 ?: 🗦	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	x	х	0	?	Х		
	宝元	工研院					
	x	х					
C#程序范例	StructMsg.pres	et_count _pro	eset_count;				
	StructMsg.Pwd	_Pwd;					
	private void b	utton2_Click	(object sen	der, EventArgs	e)		
	{						

_Pwd.ConnectionKey = "123";
_preset_count.PresetName = "PRESET.PR";
<pre>short ret = iRemoting.GET_preset_count(_Pwd, ref _preset_count);</pre>
if (ret == 0)
{
<pre>MessageBox.Show(_preset_count.Count.ToString());</pre>
}
}

# 10. GET\_preset\_data: 取得工作台管理表数据

API 名称	GET_preset_da	ta				
函式说明	取得工作台管理著	表数据				
C#方法	short GET_pres	et_data(Stru	ictMsg.Pwd P	, ref StructMs	g.preset_data	R);
<b>VB.Net</b> 方法	GET_preset_data(ByVal P As StructMsg.Pwd, ByRef R As					
	StructMsg.pres	set_data) As	Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> 词	青参考第三章 <b>)</b> 。		
	结构名称: pres	et_data				_
	变量名称	变量名称         型别         [in/out]         说明				
	PresetName	string	in	工作台管理表名	称(PRESET.PR	
				/ PRESET2.PR	/ PRESET3.PR)	
	Data	<pre>string[][]</pre>	out	工作台管理数据		
备注	※ Data 第二个数	数组,请配合标	题名称。			
适用范围	0:支援 X:	未支援 ?:ㅋ	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	X	х	0	?	Х	
	宝元	工研院				
	X	Х				
C#程序范例	StructMsg.pres	et_data _pres	set_data;			
	StructMsg.Pwd	_Pwd;				
	private void b	utton3_Click(	object send	der, EventArgs	e)	
	{					
	_Pwd.Conned	ctionKey = "1	23";			
	_preset_dat	ta.PresetName	= "PRESET.	PR";		
	short ret =	= iRemoting.G	ET_preset_d	ata(_Pwd, ref	_preset_data);	
	if (ret ==	0)				
	{					
	DataTab	le dtTool = 1	new DataTab	le();		
	for (in	ti=0;i<=	_preset_ti	tle.PresetTitle	2.Length - 1; i	.++)
	{					

```
dtTool.Columns.Add(_preset_title.PresetTitle[i],
Type.GetType("System.String"));
    }
    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;
    }
}</pre>
```

### 11. SET\_preset\_data: 写入工作台管理表

API 名称	SET_preset_data					
函式说明	写入工作台管理表					
C#方法	<pre>short SET_preset_data(StructMsg.Pwd P, StructMsg.preset_data R);</pre>					
VB.Net 方法	SET_preset_dat	ta(ByVal P As	s StructMsg	.Pwd, ByVal R	As	
	StructMsg.pres	set_data) As	Short			
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: prese	et_data				
	变量名称	型别	[in/out]	说明		
	PresetName	string	in	工作台管理表名称(PRESET.PR		
				/ PRESET2.PR / PRESET3.PR)		
	Data	<pre>string[][]</pre>	in	设定工作台管理	数据	
备注	※ 可针对某几笔	数据或全部数排	居,进行写入)	更新。		
	※ 针对某几笔数	据更新时,第[	][0]个数组词	可指定编号,如范	例。	
	※ 必须要设定 A	PI 写入密码(Pu	wd.WritePwd	l),请参阅「四、AP	I 联机及写入密码	谒⅃。
适用范围	0 : 支援 X :	未支援 ?: 未	卡实作			
	FANUC	三菱	海德汉	西门子	新代	
	х	Х	0	?	Х	
	宝元	工研院				
	Х	Х				
C#程序范例	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;</pre>
<pre>for (int i = 0; i &lt;= _preset_title.PresetTitle.Length - 1; i++)</pre>
{
_preset_data.Data[0][i] = dtTool.Rows[0][i].ToString();
}
_Pwd.ConnectionKey = "123";
<pre>short ret = iRemoting.SET_preset_data(_Pwd, _preset_data);</pre>

# 12. GET\_preset\_data\_scope: 取得工作台管理表,并依指定索引值范围传回

API 名称	GET_preset_data_scope					
函式说明	工作台管理表数据	居,并依指定索	引值范围传回	1		
<b>C#</b> 方法	<pre>short GET_preset_data_scope(StructMsg.Pwd P, ref StructMsg.</pre>					
	<pre>preset_data_scope R);</pre>					
VB.Net 方法						
数据结构	结构名称: preset_data_scope					
	变量名称	型别	[in/out]	说明		
	StartIndex	int	in	起始索引值		
	EndIndex	int	in	结束索引值		
	PresetName	string	in	PRESET.PR.		
				PRESET2.PR		
				PRESET3.PR		
	Data	<pre>string[][]</pre>	out	工作台管理数据		
备注						
适用范围	0 : 支援 X :	未支援 ?: 未	卡实作			
	FANUC	三菱	海德汉	西门子	新代	
	x	Х	0	?	Х	
	宝元	工研院				
	X	Х				
C#程序范例	StructMsg.pres	et_data_scope	_DataScope	2;		
	StructMsg.Pwd	_Pwd;				
	for (int i = 0	; i <= _Prese	tMaxIndex.M	NaxIndex; i +=	20)	
	{					
	_Pwd.Connec	ctionKey = "1	23";			
	//一次读取2	0笔				
	_DataScope	.StartIndex =	i;			

```
if ((i + 20) <= _PresetMaxIndex.MaxIndex)</pre>
       _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++)</pre>
       {
           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 变量类别函式(共通变量)

## 1. GET\_macro\_all: 取得所有 Macro(共通变量)变量数据

API 名称	GET_macro_all					
函式说明	取得所有 Macro	变量数据				
C#方法	short GET_macr	ro_all(Struc	tMsg.Pwd P	, ref StructMsg	g.macro_all R	);
VB.Net 方法	GET_macro_all(	(ByVal P As	StructMsg.	Pwd, ByRef R As	5	
	StructMsg.macr	ro_all) As S	hort			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: macro	o_all				_
	变量名称	型别	[in/out]	说明		
	MacroNumber	int[]	out	Macro 编号		
	MacroData	double[]	out	Macro 变量数据		
备注						
适用范围	0 : 支援 X : :	未支援 ?: 🕫	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.macr	ro_all _macr	o_all;			
	StructMsg.Pwd	_Pwd;				
	private void b	outton2_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ctionKey = '	"123" <b>;</b>			
	short ret	= iRemoting	.GET_macro	_all(_Pwd, ref	_macro_all);	
	if (ret ==	0)				
	{					
	DataTal	ble dtMacro	= new Data	Table();		
	dtMacro	o.Columns.Ad	ld("No", Ty	pe.GetType("Sy	stem.Int16"))	;
	dtMacro	o.Columns.Ad	ld("Data",			
	Type.GetType('	'System.Doub	le"));			
	DataRo	w dr;				
	for (ir	nti=0;i<	= _macro_a	11.MacroNumber.	Length - 1; i	.++)
	{					
	dr	= atMacro.N	ewkow();	Numbers		
	ar	ט] = _macro	_all_Macro	Number[1];		
	ar معد		_all.macro	ναια[1];		
	ן ג	aci 0.80W5.A	uu(ur),			
	∫ dataGr	idView1 Data	Source = d	ltMacro.		
			3341 cc = 0			

}
}

2. GET\_macro\_scope: 依指定范围取得 Macro(共通变量)变量数据

	API 名称	GET_macro_scope					
C#方法short GET_macro_scope(StructMsg.Pwd P, ref StructMsg.macro_scope A);VB.Net 方法GET_macro_scope(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.macro_scope) As Short数据结构结构名称: Pwd. 取机密码及写入密码设置(请参考第三章)。 结构名称: macro_scope変量名称型別[in/out]変現[in/out]StartNumberintintin起始号码(从1开始)EndNumberEndNumberint[]outMacro 400ble[]MacroNumberint[]outMacro 400ble[]MacroNumberint[]outMacro 400ble[]MacroNumberint[]outMacro 400ble[]O:文授 X:* * * * * * * *FANUC三 / * # # # # # # # # # # # # # # # # # #	函式说明	依指定范围取得 Macro 变量数据					
VB.Net 方法         GET_macro_scope(ByVal P As StructMsg.Pwd, ByRef R As StructMsg.macro_scope) As Short           数据结构         结构名称: Pwd. 联机密码及写入密码设置(请参考第三章). 结构名称: macro_scope           変量名称         型別         [in/out]         说明           StartNumber         int         in         起始号码(从1开始)           EndNumber         int[]         out         Macro 编号           MacroNumber         int[]         out         Macro 编号           MacroNumber         int[]         out         Macro 编号           MacroData         double[]         out         Macro 2005           StartNumber         int[]         out         Macro 2005           ARit         0         int         int         int           MacroData         double[]         out         Macro 2005         int           StructMsg.macro_scope_macro_scope;         StructMsg.macro_scope_macro_scope;         int         int           C#程序范档         StructMsg.Pwd _Pwd;         private void button6_Click(object sender, EventArgs e)         {             [	C#方法	short GET_macr	o_scope(Stru	uctMsg.Pwd	P, refStructMs;	g.macro_scope	R);
StructMsg.macro_scope) As Short         数据结构       结构名称: Pwd, 联机密码及写入密码设置(请参考第三章).         结构名称: macro_scope         変量名称       型別       [in/out]         返明       StartNumber       int         in       起始号码(从1开始)         EndNumber       int[]       out         MacroNumber       int[]       out         MacroNumber       int[]       out         MacroData       double[]       out         StructMsg.macro_scope_recorescope;       StructMsg.macro_scope_macro_scope;         StructMsg.macro_scope_scope       structMsg.macro_scope.scope;         StructMsg.Pwd_Pwd;       private void button6_Cli	<b>VB.Net</b> 方法	GET_macro_scop	pe(ByVal P A	s StructMs	g.Pwd, ByRef R	As	
数据结构结构名称: Pwd. 联机密码及写入密码设置(请参考第三章).结构名称: macro_scope変量名称型别[in/out]返明StartNumberintin起始号码(从1开始)EndNumberint[]outMacro 编号MacroNumberint[]outMacro 変量数据各注02海德汉西门子新代00X?O0X?0宝元工研院四0文0X?C#程序范例StructMsg.macro_scope _macro_scope;StructMsg.Pwd_Pwd; private void button6_Click(object sender, EventArgs e) { 		StructMsg.macr	ro_scope) As	Short			
结构名称: macro_scope         変量名称       型別       [in/out]       说明         StartNumber       int       in       是如号码(从1开始)         EndNumber       int       in       终止号码         MacroNumber       int[]       out       Macro 3編号         MacroData       double[]       out       Macro 3編号         MacroData       double[]       out       Macro 2g量数据          0       0       X       ?          FANUC       三菱       海德汉       西门子       新代          0       0       X       ?       0          室元       工研院       □       □       □       □          0       0       X       ?       0       □          0       0       X       ?       0       □          StructMsg.macro_scope_macro_scope;       StructMsg.macro_scope_macro_scope;       StructMsg.macro_scope_macro_scope;       StructMsg.macro_scope.scope;       []	数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
変量名称       型別       [in/out]       说明         StartNumber       int       in       起始号码(从1开始)         EndNumber       int       in       终止号码         MacroNumber       int[]       out       Macro 線号         MacroData       double[]       out       Macro 変量数据          Account       double[]       out       Macro 変量数据          6       1       Macro 変量数据       Macro 変量数据          0       2       万       新代         0       0       X       ?       0         室電元       工研院       四口子       新代         0       0       X       ?       0         室元       工研院       0       0       0       0         C#程序范例       StructMsg.macro_scope_macro_scope;       StructMsg.macro_scope_macro_scope;       StructMsg.macro_scope_macro_scope;         StructMsg.macro_scope.StartNumber = 513;		结构名称: macro	o_scope		1		1
StartNumber       int       in       起始号码(从1开始)         EndNumber       int       in       终止号码         MacroNumber       int[]       out       Macro 3k号         MacroData       double[]       out       Macro 2kg         SA注       0:       支援 X: 未支援 ?: 未实作         FANUC       三菱       海德汉       西门子       新代         0       0       X       ?       0         室元       工研院       回       回       □         O       0       X       ?       0         室元       工研院       □       □       □         O       0       X       ?       0         StructMsg.macro_scope _macro_scope;       StructMsg.Pwd _Pwd;       private void button6_Click(object sender, EventArgs e)       { <ul> <li> <li></li></li></ul>		变量名称	型别	[in/out]	说明		
EndNumber       int       in       终止号码         MacroNumber       int[]       out       Macro 编号         MacroData       double[]       out       Macro 变量数据         备注         FANUC       三菱       海德汉       西门子       新代         0       0       X       ?       0         宝元       工研院            0       0       X       ?       0         宝元       工研院            0       0       X       ?       0         宝元       工研院            0       0            StructMsg.macro_scope _macro_scope;            StructMsg.Pwd _Pwd;       private void button6_Click(object sender, EventArgs e)           {		StartNumber	int	in	起始号码(从17	F始)	
MacroNumber       int[]       out       Macro 编号         MacroData       double[]       out       Macro 変量数据          A       A       A          0: 支援 X: 未支援 ?: 未实作       第億次       西门子       新代          0       0       X       ?       O         室元       工研院       0       0       Image: StructMsg.macro_scope       Image: StructMsg.macro_scope _ macro_scope;         C#程序范例       StructMsg.macro_scope _macro_scope;       StructMsg.Pwd _Pwd;       private void button6_Click(object sender, EventArgs e)       {		EndNumber	int	in	终止号码		
MacroData       double[]       out       Macro变量数据         备注       适用范围       0: 支援 X: 未支援 ?: 未实作         FANUC       三菱       海德汉       西门子       新代         0       0       X       ?       0         宝元       工研院       山       山       山         0       0       X       ?       0         宝元       工研院       山       山       山         0       0       x       ?       0         宝元       工研院       山       山       山         0       0       u       山       山         C#程序范例       StructMsg.macro_scope _macro_scope;       StructMsg.Pwd _Pwd;       private void button6_Click(object sender, EventArgs e)       {         {		MacroNumber	int[]	out	Macro 编号		
		MacroData	double[]	out	Macro 变量数据		
适用范围       0: 支援 X: 未支援 ?: 未实作                 FANUC 三菱 海德汉 西门子 新代           0       0       X       ?       0         宝元       工研院	备注						
FANUC       三菱       海德汉       西门子       新代         0       0       X       ?       0         宝元       工研院       -       -       -         0       0       -       -       -         C#程序范例       StructMsg.macro_scope _macro_scope;       StructMsg.Pwd _Pwd;       -         private void button6_Click(object sender, EventArgs e)       {       -       -         {	适用范围	0 : 支援 X : :	未支援 ?:	未实作			
0       0       X       ?       0         宝元       工研院		FANUC	三菱	海德汉	西门子	新代	
宝元       工研院       Image: Complexity of the state of the st		0	0	Х	?	0	
0       0         C#程序范例       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;		宝元	工研院				
C#程序范例 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;		О	0				
<pre>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;     } } </pre>	C#程序范例	StructMsg.macr	o_scope _mac	ro_scope;			
<pre>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;     } } </pre>		StructMsg.Pwd	_Pwd;				
<pre>{     _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;     } } </pre>		private void b	utton6_Click	(object sen	der, EventArgs	e)	
<pre>_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;</pre>		{					
<pre>_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;</pre>		_Pwd.Connec	tionKey = "1	.23";			
<pre>_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;</pre>							
<pre>_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;</pre>		_macro_scop	e.StartNumbe	er = 513;			
<pre>short ret = iRemoting.GET_macro_scope(_Pwd, ret _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;</pre>		_macro_scop	e.EndNumber	= 515;		,	
<pre>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;</pre>		short ret =	<pre>short ret = iRemoting.GET_macro_scope(_Pwd, ref _macro_scope);</pre>				
<pre>{     DataTable dtMacro = new DataTable();     dtMacro.Columns.Add("No", Type.GetType("System.Int16"));     dtMacro.Columns.Add("Data", Type.GetType("System.Double"));     DataRow dr; </pre>		1† (ret ==	if (ret == 0)				
<pre>dtalable dtMacro = new Datalable(); dtMacro.Columns.Add("No", Type.GetType("System.Int16")); dtMacro.Columns.Add("Data", Type.GetType("System.Double")); DataRow dr;</pre>		i DotoTob	{				
dtMacro.Columns.Add("No", Type.GetType("System.Intio")); dtMacro.Columns.Add("Data", Type.GetType("System.Double")); DataRow dr;			DataTable dtMacro = new DataTable();				
DataRow dr;		dtMacro	Columns Add	( NO , Type	vna Cattype( Syste	tom Doublo"	
			.corullins.Auu	(Data, Iy	perderiype( sys	stem.Double ))	ر
for (int $i = 0$ , $i < macro scope MacroNumber Length - 1, i \neq 1)$		for (in	, t i = 0· i /	= macro sc	one MacroNumber	length - 1· i	i++)
		{  {101}(111	τ - υ, τ Υ			·	

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

## 3. SET\_macro\_all: 写入所有 Macro(共通变量)变量数据

API 名称	SET_macro_all						
函式说明	写入所有 Macro 变量数据						
C#方法	short SET_mac	ro_all(Struc	tMsg.Pwd P	. ر	StructMsg.mac	<pre>cro_all R);</pre>	
VB.Net 方法	SET_macro_all	(ByVal P As	StructMsg.	Pw	d, ByVal R As	5	
	StructMsg.macro_all) As Short						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请	参考第三章 <b>)</b> 。		
	结构名称: macr	o_all					
	变量名称	型别	[in/out]	访	兑明		
	MacroNumber	int[]	in	访	设定 Macro 编号		
	MacroData	double[]	in	访	設定 Macro 变量	数据	
备注	※ 必须要设定 API 写入密码(Pwd.WritePwd),请参阅「四、API 联机及写入密码」。						
适用范围	0:支援 X:	未支援 ?:	未实作				
	FANUC	三菱	海德汉		西门子	新代	
	0	0	Х		?	0	
	宝元	工研院					
	0	0					
C#程序范例	StructMsg.mac	ro_all _macr	o_all;				
	StructMsg.Pwd	_Pwd;					
	private void	button3_Clic	k(object s	en	der, EventArg	gs e)	
	{						
	_Pwd.Conne	ectionKey = '	"123";				
	short ret	= iRemoting	.GET_macro	_a]	ll(_Pwd, ref	_macro_all);	
	if (ret == 0)						
	{	{					
	//Sear	ch no					
	for (1	nt 1 = 0; 1 <	= _macro_a	11	.MacroNumber.	Length - 1; 1	.++)
	۱ ۰.۰.	( macno all	MacaaNumb	on	[;]		
	IT	(_macro_all	• macronullD	er.	LT] ==		
	snort.Parse(t	rt.Parse(txtNo.Text))					

{
_macro_all.MacroData[i] =
<pre>double.Parse(txtData.Text);</pre>
break;
}
}
}
<pre>ret = iRemoting.SET_macro_all(_macro_all);</pre>
if (ret == 0)
<pre>MessageBox.Show("Write success!");</pre>
else
<pre>MessageBox.Show("Write failed! ret:" + ret.ToString());</pre>
}

4. GET\_macro\_single: 取得单笔 Macro(共通变量)变量数据

API 名称	GET_macro_single					
函式说明	取得单笔 Macro 变量数据					
C#方法	short GET_macr	<pre>`o_single(St</pre>	ructMsg.Pw	d P, ref Struct	Msg.macro_sin	ıgle
	R);					
VB.Net 方法	GET_macro_sing	gle(ByVal P	As StructM	sg.Pwd, ByRef R	R As	
	StructMsg.macr	ro_single) A	s Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
	结构名称: macro	o_single				
	变量名称	变量名称 型别 [in/out] 说明				
	MacroNumber	int	in	Macro 编号		
	MacroData	double	out	Macro 变量数据		
备注						
适用范围	<b>0:</b> 支援 X:	未支援 ?: 🗦	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.macr	ro_single _m	acro_singl	e;		
	StructMsg.Pwd _Pwd;					
	<pre>private void button4_Click(object sender, EventArgs e)</pre>					
	{					
	_Pwd.Conne	ctionKey = '	'123" <b>;</b>			
	_macro_sin	gle.MacroNur	nber = shor	rt.Parse(txtNo.	Text);	

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

# 5. SET\_macro\_single: 写入单笔 Macro(共通变量)变量数据

API 名称	SET_macro_single						
函式说明	写入单笔 Macro 变量数据						
C#方法	short SET_macr	ro_single(St	ructMsg.Pw	d F	?, StructMsg.	macro_single	R);
VB.Net 方法	SET_macro_sin	gle(ByVal P	As StructM	sg	.Pwd, ByVal F	R As	
	StructMsg.mac	ro_single) A	s Short				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请	参考第三章 <b>)</b> 。		
	结构名称: macro	o_single		-			1
	变量名称	型别	[in/out]	访	泡明		
	MacroNumber	int	in	讶	设定 Macro 编号		
	MacroData	double	in	访	と定 Macro 变量	数据	
备注	※ 必须要设定 A	PI 写入密码(P	wd.WritePw	d)	,请参阅「四、AP	I联机及写入密码	玛亅。
适用范围	0:支援 X:	0:支援 X:未支援 ?:未实作					
	FANUC	三菱	海德汉		西门子	新代	
	0	0	Х		?	0	
	宝元	工研院					
	0	0					
C#程序范例	StructMsg.mac	ro_single _m	acro_singl	e;			
	StructMsg.Pwd	_Pwd;					
	private void l	outton5_Clic	k(object s	en	der, EventArg	gs e)	
	{						
	_Pwd.Conne	ctionKey = '	"123";				
	_macro_single.MacroNumber = short.Parse(txtNo.Text);						
	<pre>_macro_single.MacroData = double.Parse(txtData.Text);</pre>						
	<pre>short ret = iRemoting.SET_macro_single(_Pwd, _macro_single);</pre>						
	if (ret == 0)						
	Messag	eBox.Show("W	Irite succe	ess	!");		
	else						
	Messag	eBox.Show("W	Vrite faile	ed !	ret:" + ret	.ToString());	
	}						

## 6. GET\_macro\_variable: 取得 Macro(共通变量)变量型态

API 名称	GET_macro_count					
函式说明	取得 Macro 变量型态					
C#方法	short GET_macr	<pre>short GET_macro_variable(StructMsg.Pwd P, ref</pre>				
	StructMsg.macr	ro_variable	R);			
<b>VB.Net</b> 方法	GET_macro_vari	iable(ByVal	P As Struct	tMsg.Pwd, ByRet	FRAs	
	StructMsg.macr	ro_variable)	As Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> i	清参考第三章 <b>)</b> 。		
	结构名称: macro	o_variable				-
	变量名称	型别	[in/out]	说明		
	Variable	int[][]	out	Macro 变量型态		
备注	数组宣告: Varia	able[type][0	]:Start Nur	nber		
	Varia	able[type][1	]:End Numbe	er		
	如:					
	Variable[0	][0]:100				
	Variable[0	Variable[0][1]:199				
	Variable[1][0]:500					
	Variable[1][1]:599					
适用范围	0 : 支援 X : :	未支援 ?: 🗦	未实作			1
	FANUC	三菱	海德汉	西门子	新代	
	0	0	X	?	0	
	宝元	工研院				
	0	0				
C#程序范例	InterfaceLib.S	StructMsg.ma	cro_variab	le _macro_varia	able;	
	StructMsg.Pwd	_Pwd;				
	<pre>private void button1_Click(object sender, EventArgs e)</pre>					
	{					
	_Pwd.Conne	ctionKey = '	'123";		c	
	<pre>short ret = iRemoting.GET_macro_variable(_Pwd, ref</pre>					
	_macro_variab	Le);				
		. 0)				
	۱ for (ir	y+ i = 0 · i / =	macro var	ishle Vanishle	length - 1 · f	ί <b>τ</b> τ \
		10 1 - 0, 1 (-			.Lengen - 1, 1	.++)
	י דעל לאל	Variable Te	xt += "Type	•" + (i + 1) ToS	$tring() + "\r$	n".
	txt	Variable.Te	xt += "Star	rt Number:" +		(,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
	macro variab	le.Variable[	i][0].ToStu	$ring() + "\r\n$	".	
	txt	Variable.Te	xt += "End	Number:" +	,	
		le.Variable[	i][1].ToSt	$ring() + "\r\n'$	. <sup>۱۱</sup>	

}
}
}

7. <u>GET\_pcode\_single</u>: 取得单笔 P Code Macro 资料(FANUC)

	-					
API 名称	GET_pcode_sing	GET_pcode_single				
函式说明	取得单笔 P Code Macro 资料(FANUC)					
C#方法	short GET_pcod	<pre>le_single(St</pre>	ructMsg.Pw	d P, ref Struct	Msg.macro_sir	ngle
	R);					
VB.Net 方法	GET_pcode_sing	gle(ByVal P	As StructM	sg.Pwd, ByRef F	₹ As	
	StructMsg.macr	ro_single) A	s Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: macro	o_single				
	变量名称	型别	[in/out]	说明		
	MacroNumber	int	in	P code 编号		
	MacroData	double	out	P code 变量数据	1	
备注	如果回传值为6即	<b>ग</b> :				
	1. 请检查此台控制器是否已选配 macro-executor 功能。					
	2. macro-executor 程序是否已经安装至控制器内。					
适用范围	0:支援 X:	未支援 ?: 🕫	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	Х	Х	х	Х	
	宝元	工研院				
	х	Х				
C#程序范例	StructMsg.macr	ro_single _m	acro_singl	e;		
	StructMsg.Pwd	_Pwd;				
	private void b	outton4_Clic	k(object s	ender, EventAr <sub>{</sub>	gs e)	
	{					
	_Single.Ma	croNumber =	int.Parse(	(txtNum.Text);		
	short ret	= iRemoting	.GET_pcode_	_single(_Pwd, r	ef _Single);	
	if (ret ==	0)				
	txtVal	ue.Text = _S	ingle.Macr	oData.ToString	();	
	else					
	Message	eBox.Show(re	t.ToString	;());		
	}					

8. SET\_pcode\_single: 写入单笔 P Code Macro 资料(FANUC)

API 名称	SET_pcode_single
函式说明	写入单笔 P Code Macro 资料(FANUC)

C#方法	<pre>short SET_pcode_single(StructMsg.Pwd P, StructMsg.macro_single R);</pre>					
<b>VB.Net</b> 方法	SET_pcode_sing	gle(ByVal P	As StructM	sg.Pwd, ByVal F	R As	
	StructMsg.mac	ro_single) A	s Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
	结构名称: macro	o_single				_
	变量名称	型别	[in/out]	说明		
	MacroNumber	int	in	设定 Macro 编号		
	MacroData	double	in	设定 Macro 变量	数据	
备注	※ 必须要设定 A	PI 写入密码(P	wd.WritePw	d),请参阅「四、AP	I联机及写入密	码」。
	如果回传值为6日	时:				
	1. 请检查此台控	控制器是否已选	配 macro-ex	ecutor 功能。		
	2. macro-exec	utor 程序是否	已经安装至控	它制器内。		
适用范围	0:支援 X:	未支援 ?:	未实作			-
	FANUC	三菱	海德汉	西门子	新代	
	0	Х	Х	x	Х	
	宝元	工研院				
	Х	Х				
C#程序范例	StructMsg.mac	ro_single _m	acro_singl	e;		
	StructMsg.Pwd	_Pwd;				
	private void l	outton5_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ectionKey =	"123";			
	_Single.Ma	croNumber =	int.Parse(	(txtNum.Text);		
	_Single.Ma	icroData = do	ouble.Parse	e(txtValue.Text	);	
	short ret	= iRemoting	.SET_pcode_	_single(_Pwd, _	Single);	
	}					

## 十三、 加工程序类别函式

## 1. GET\_nc\_mem\_list: 取得机台内部加工程序列表(内存)

API 名称	GET_nc_mem_list					
函式说明	取得机台内部加二	E程序列表(内存	子 <b>)</b>			
C#方法	short GET_nc_n	nem_list(Stru	uctMsg.Pwd	P, ref StructM	/sg.nc_list R)	;
VB.Net 方法	GET_nc_mem_lis	st(ByVal P As	s StructMsg	g.Pwd, ByRef R	As	
	StructMsg.nc_]	list) As Shor	rt			
数据结构	结构名称: Pwd,	联机密码及写入	\密码设置 <b>(</b> 词	青参考第三章 <b>)</b> 。		
	结构名称: nc_list					
	变量名称	型别	[in/out]	说明		
	FolderPath	string	in	文件夹路径		
	NcList	<pre>string[][]</pre>	out	加工程序列表		
备注	≪FANUC 0i-D /	′31i 控制器:n	c_list.Fol	derPath 默认值为	5	
	//CNC_MEM/USEF	R/PATH1/				
	※NcList 第二个	数组:				
	[0]:NC Nam	e				
	[1]:Size (	byte)				
	[2]:DateTi	me				
	[3]:批注(R	emark)				
	[4]:file /	directory				
	※此函式 FANUC 目前小支持 DateTime					
	※二委及新代日月	间个文持 Datel	1me、批注 1d:Rath 亦	<b>、</b> 曰.		
注田芬田	※HEIDENHAIN 招	2 前	IderPatn 变	.里		
迫用氾問			≤头作 海 徳 河	まれて	<b>並「</b> 42	
	PANOC	二交		2011日		
		丁亚陀	0	r	0	
	玉九	0				
<b>C</b> #程序范例	StructMcg nc 1	list no list				
し#小王/丁/1日/小	StructMsg Pwd	Pwd·	- ,			
	private void k	_' wa, nutton1 Click	(object se	onder EventArg	75 P)	
	{				,5 C/	
	Pwd.Conne	ctionKev = "	123":			
	short ret	= iRemoting.	GET nc mem	list( Pwd, re	f nc list);	
	if (ret ==	0)			/ )	
	{	,				
	DataTal	ble dtNcList	= new Data	aTable();		

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++)						
{						
<pre>dr = dtNcList.NewRow();</pre>						
dr[0] = _nc_list.NcList[i][0];						
<pre>dr[1] = int.Parse(_nc_list.NcList[i][1]);</pre>						
<pre>dr[2] = _nc_list.NcList[i][3];</pre>						
dtNcList.Rows.Add(dr);						
}						
<pre>dataGridView1.DataSource = dtNcList;</pre>						
}						
}						

2. GET\_nc\_ftp\_list: 取得 FTP 卡加工程序列表(Fanuc DataServer/三菱 CF 卡)

API 名称	GET_nc_ftp_lis	st			
函式说明	取得 FTP 加工程序	序列表(Fanuc∣	DataServer	/三菱 CF 卡)	
C#方法	short GET_nc_f	tp_list(Stru	uctMsg.Pwd	<pre>P, ref StructMsg.nc_list R);</pre>	
<b>VB.Net</b> 方法	GET_nc_ftp_lis	st(ByVal P As	s StructMsg	g.Pwd, ByRef R As	
	StructMsg.nc_]	.ist) As Shor	rt		
数据结构	结构名称: Pwd,	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: nc_li	lst			
	变量名称	型别	[in/out]	说明	
	FolderPath	string	in	Not use	
	NcList	string[][]	out	加工程序列表	
备注	※ FANUC 控制器	必须拥有 Data	<b>Server,</b> 并i	已经完成设定账号及密码。	
	※ 三菱控制器无	法使用在 Wind	ows 操作系统	Ž o	
	NcList 第二个数	组包含有:			
	[0]:加工程/	亨名称(Name)			
	[1]:大小(S:	ize)			
	[2]:日期				
	[3]:属性(d:	irectory:目录	<b>,file:</b> 档案	)	

	※ 若回传含有错误码时,请参照 FTP 回传错误列表。								
	※ 若错误码回传 14 代表系统尚未设定 FTP,请检查 SkyMars 机台设定。								
适用范围	0 : 支援 X :	0:支援 X:未支援 ?:未实作							
	FANUC	三菱	海德汉	西门子	新代				
	0	0	Х	?	Х				
	宝元	工研院							
	X	Х							
C#程序范例	StructMsg.Pw	d _Pwd;							
	private void	button2_Clic	k(object sen	der, EventArg	gs e)				
	{	{							
	_Pwd.Conr	_Pwd.ConnectionKey = "123";							
	short ret	: = iRemoting	.GET_nc_ftp_	list(_Pwd, re	f _nc_list);				
	if (ret =	= 0)							
	{								
	DataT	able dtNcList	z = new DataT	able();					
	dtNcL	ist.Columns./	Add("Name",						
	Type.GetType	("System.Stri	ng"));						
	dtNcList.Columns.Add("Size",								
	Type.GetType	("System.Int3	2"));						
	DataR	ow dr;							
	for (	int i = 0; i	<= _nc_list.	NcList.Lengt	h - 1; i++)				
	{								
	i	f (_nc_list.N	cList[i][3]	== "f")					
	{								
		dr = dtNcL	ist.NewRow()	;					
		dr[0] = _n	c_list.NcList	t[i][0];					
		dr[1] = in	t.Parse(_nc_	list.NcList[i	][1]);				
		dtNcList.R	ows.Add(dr);						
	} }								
	}								
	dataG	nidView? Dot:	Source - d+N	lclist.					
	L L L L L L L L L L L L L L L L L L L	I IUVIEWZ.Dala		162236,					
	J								
	1.1								

3. GET\_nc\_mem\_code: 取得内存内的加工程序内容

API 名称	GET_nc_mem_code
函式说明	取得内存内的加工程序内容

C#方法	short GET_nc_n	<pre>short GET_nc_mem_code(StructMsg.Pwd P, ref StructMsg.nc_code R);</pre>					
VB.Net 方法	GET_nc_mem_cod	GET_nc_mem_code(ByVal P As StructMsg.Pwd, ByRef R As					
	StructMsg.nc_c	code) As Sho	rt				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参	》考第三章 <b>)</b> 。		
	结构名称: nc_co	ode					
	变量名称	型别	[in/out]	说	明		
	FolderPath	string	in	文	件夹路径		
	NcName	string	in	名	称		
	NcCode	string	out	内	容		
备注	FANUC 0i-D / 3	31i 控制器:Fo	lderPath 默	认值	直为//CNC_MEM	/USER/PATH1/	
	HEIDENHAIN 控制	剧器需设定 Fol	derPath 变量	l L			
适用范围	<b>0 :</b> 支援 X : :	未支援 ?: 🕫	未实作				
	FANUC	三菱	海德汉		西门子	新代	
	0	0	0		?	0	
	宝元	工研院					
	О	0					
C#程序范例	StructMsg.nc_c	code _nc_cod	e;				
	StructMsg.Pwd	_Pwd;					
	private void b	outton3_Clic	k(object s	end	er, EventArg	gs e)	
	{						
	_Pwd.Conne	ctionKey = '	'123" <b>;</b>				
	_nc_code.N	cName =					
	dataGridView1.	CurrentRow.	Cells[0].Va	alu	e.ToString()	);	
	short ret	= iRemoting	.GET_nc_mem	n_co	ode(_Pwd, re	f _nc_code);	
	if (ret ==	0)					
	{						
	rtbMeml	NcCode.Text	= _nc_code	e.Nc	Code;		
	}						
	}						

# 4. GET\_nc\_ftp\_code: 取得 FTP 加工程序内容(Fanuc DataServer/三菱 CF 卡)

API 名称	GET_nc_ftp_code
函式说明	取得 FTP 加工程序内容(Fanuc DataServer/三菱 CF 卡)
C#方法	<pre>short GET_nc_ftp_code(StructMsg.Pwd P, ref StructMsg.nc_code R);</pre>
VB.Net 方法	GET_nc_ftp_code(ByVal P As StructMsg.Pwd, ByRef R As
	StructMsg.nc_code) As Short
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。
	结构名称: nc_code

	变量名称	型别	[in/out]	说明		
	FolderPath	string	in	Not use		
	NcName	string	in	名称		
	NcCode	string	out	内容		
备注	※ FANUC 控制器	必须拥有 FTP,	并已经完成	设定账号及密码。		
	※ 三菱控制器无	法使用在 Winc	lows 操作系约	T. •		
	※ 若回传含有错	误码时,请参知	照 FTP 回传错	误列表。		
	※ 若错误码回传	14 代表系统尚	的未设定 FTP,	请检查 SkyMars	机台设定。	
适用范围	<b>0 :</b> 支援 X : :	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	Х	
	宝元	工研院				
	х	Х				
C#程序范例	StructMsg.nc_c	ode _nc_cod	e;			
	StructMsg.Pwd	_Pwd;				
	private void b	outton4_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_nc_code.N	cName =				
	dataGridView2.	CurrentRow.	Cells[0].V	alue.ToString()	);	
	_Pwd.Conne	ctionKey = '	"123";			
	short ret	= iRemoting	.GET_nc_ftp	_code(_Pwd, re	f _nc_code);	
	if (ret ==	0)				
	{					
	rtbDsNo	Code.Text =	_nc_code.	NcCode;		
	}					
	}					

## 5. UPLOAD\_nc\_mem: 上传加工程序至内存

API 名称	UPLOAD_nc_mem				
函式说明	上传加工程序至内	内存			
<b>C#</b> 方法	short UPLOAD_r	nc_mem(Struc	tMsg.Pwd P	<pre>, StructMsg.nc_code R);</pre>	
VB.Net 方法	UPLOAD_nc_mem(	UPLOAD_nc_mem(ByVal P As StructMsg.Pwd, ByVal R As StructMsg.nc_code)			
	As Short				
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章)。	
	结构名称: nc_co	ode			
	变量名称	型别	[in/out]	说明	
	FolderPath	string	in	文件夹路径	
	NcName	string	in	名称	
	NcCode	string	in	内容	

备注	FANUC Øi-D / 31i 控制器:FolderPath 默认值为//CNC_MEM/USER/PATH1/						
	HEIDENHAIN 控制器需设定 FolderPath 变量						
	※ 必须要设定	API 写入密码(P	Wd.WritePwd)	,请参阅「四、API	I联机及写入密	码」。	
适用范围	0:支援 X:	未支援 ?::	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	0	0	?	0		
	宝元	工研院					
	0	0					
C#程序范例	StructMsg.nc	_code _nc_cod	le;			<u></u>	
	StructMsg.Pwo	d _Pwd;					
	<pre>private void button5_Click(object sender, EventArgs e)</pre>						
	{						
	_Pwd.ConnectionKey = "123";						
	_nc_code.	NcName = "00	002";				
	_nc_code.	NcCode = Sys	tem.IO.File.F	ReadAllText("	D:\\00002");		
	short ret	: = iRemoting	.UPLOAD_nc_me	em(_Pwd, _nc_	_code);		
	if (ret =	= 0)					
	Messa	geBox.Show("W	Nrite success	!");			
	else						
	Messa	geBox.Show("W	Vrite failed!	ret:" + ret	.ToString());	;	
	}						

# 6. UPLOAD\_nc\_ftp: 上传加工程序至 FTP(Fanuc DataServer/三菱 CF 卡)

API 名称	UPLOAD_nc_ftp					
函式说明	上传加工程序至日	上传加工程序至 FTP(Fanuc DataServer/三菱 CF 卡)				
C#方法	short UPLOAD_r	nc_ftp(Struc	tMsg.Pwd P	<pre>, StructMsg.nc_code R);</pre>		
<b>VB.Net</b> 方法	UPLOAD_nc_ftp	(ByVal P As St	ructMsg.Pw	vd, ByVal R As StructMsg.nc_code	e)	
	As Short					
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。		
	结构名称: nc_co	ode				
	变量名称	型别	[in/out]	说明		
	FolderPath	string	in	文件夹路径		
	NcName	string	in	名称		
	NcCode	string	in	内容		
备注	FANUC 控制器必须	页拥有 DataSei	rver,并已经	经完成设定账号及密码。		
	※ 三菱控制器无	法使用在 Wind	lows 操作系线	充。		
	※ 若回传含有错	误码时,请参照	照 FTP 回传错	<b></b> 请误列表。		
	※ 若错误码回传	14 代表系统尚	的未设定 FTP,	请检查 SkyMars 机台设定。		

	※ 必须要设定 API 写入密码(Pwd.WritePwd),请参阅「四、API 联机及写入密码」。						
适用范围	0: 支援 X: 未支援 ?: 未实作						
	FANUC	三菱	海德汉	西门子	新代		
	0	0	Х	?	Х		
	宝元	工研院					
	Х	Х					
C#程序范例	StructMsg.nc_code _nc_code;						
	StructMsg.Pwd _Pwd;						
	<pre>private void button6_Click(object sender, EventArgs e)</pre>						
	{						
	_Pwd.ConnectionKey = "123";						
	_nc_code.NcName = "00002";						
	_nc_code.	NcCode = Sys	tem.IO.File.F	ReadAllText("	D:\\00002");		
	short ret	: = iRemoting	.UPLOAD_nc_ft	p(_Pwd, _nc_	_code);		
	if (ret =	= 0)					
	Messa	geBox.Show("W	Vrite success	!");			
	else						
	Messa	geBox.Show("W	Vrite failed!	ret:" + ret	.ToString());	;	
	}						

## 7. DEL\_nc\_mem: 删除加工程序(内存)

API 名称	DEL_nc_mem	DEL_nc_mem					
函式说明	删除加工程序(内	删除加工程序(内存)					
<b>C#</b> 方法	short DEL_nc_	mem(StructMs	g.Pwd P, St	tructMsg.NcName	e R);		
VB.Net 方法	DEL_nc_mem(By	Val P As Stru	ctMsg.Pwd,	ByVal R As Stru	ctMsg.NcName	) As	
	Short						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> i	青参考第三章 <b>)</b> 。			
	结构名称: NcNa	me					
	变量名称	型别	[in/out]	说明			
	FolderPath	string	in	文件夹路径			
	Name	string	in	名称			
备注	FANUC 0i-D /	31i 控制器:Fo	lderPath 默	认值为//CNC_MEM	/USER/PATH1/		
	HEIDENHAIN 控制	间器需设定 Fol	derPath 变量	L			
	※ 必须要设定 A	.PI 写入密码(P	wd.WritePwo	d),请参阅「四、AP	I联机及写入密	码」。	
适用范围	0 : 支援 X :	未支援 ?:	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	0	0	?	0		
	宝元	工研院					
	0	0				1	

C#程序范例	StructMsg.NcName _NcName;							
	StructMsg.Pwd _Pwd;							
	<pre>private void button7_Click(object sender, EventArgs e)</pre>							
	{							
	_Pwd.ConnectionKey = "123";							
	_NcName.Name =							
	:aGridView1.CurrentRow.Cells[0].Value.ToString();							
	<pre>short ret = iRemoting.DEL_nc_mem(_Pwd, _NcName);</pre>							
	if (ret == 0)							
	<pre>MessageBox.Show("Write success!");</pre>							
	else							
	<pre>MessageBox.Show("Write failed! ret:" + ret.ToString());</pre>							
	}							

8. DEL\_nc\_ftp: 删除加工程序(FTP) (Fanuc DataServer/三菱 CF 卡)

API 名称	DEL_nc_ftp					
函式说明	删除加工程序(f	删除加工程序(ftp) (Fanuc DataServer/三菱 CF 卡)				
C#方法	short DEL_nc_	ftp(StructMs	g.Pwd P, S	tructMsg.NcNam	e R);	
<b>VB.Net</b> 方法	DEL_nc_ftp(By	Val P As Stru	ctMsg.Pwd,	ByVal R As Stru	uctMsg.NcName	) As
	Short					
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: NcNa	me				
	变量名称	型别	[in/out]	说明		
	FolderPath	string	in	Not use		
	Name	string	in	名称		
备注	FANUC 控制器必须	须拥有 DataSei	rver,并已经	至完成设定账号及额	密码。	
	※ 三菱控制器无	E法使用在 Wind	lows 操作系线	充。		
	※ 若回传含有错	错误码时,请参照	照 FTP 回传错	错误列表。		
	※ 若错误码回传	专14代表系统尚	前未设定 FTP,	请检查 SkyMars	机台设定。	
	※ 必须要设定 A	API 写入密码(P	wd.WritePw	d),请参阅「四、AP	I 联机及写入密码	玛亅。
适用范围	0 : 支援 X :	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	Х	?	Х	
	宝元	工研院				
	x	Х				
C#程序范例	StructMsg.NcN	ame _NcName;				
	StructMsg.Pwd	_Pwd;				
	private void	button8_Clic	k(object s	ender, EventAr	gs e)	
	{					

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

# 9. \_GET\_nc\_pointer: 取得程序目前的指针(行)

API 名称	GET_nc_pointer				
函式说明	取得程序目前的指针(行)				
C#方法	<pre>short GET_nc_pointer(StructMsg.Pwd P, StructMsg.NcPointer R);</pre>				
VB.Net 方法	GET_nc_pointer	r(ByVal R As	StructMsg	.NcPointer) As Short	
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。				
	本島夕称	刑되	[in/out]	沿明	
	又重石标 BlkPointer	主 <u>师</u> int		日前程序执行的指针	
	X 三茎控制哭不	支持此函式	040		
田 1	※ 如 EANUC NC	<b>スパル団ス。</b> 码・			
	00050(BT2):	• •			
	G01 Y0 X0 Z0	0 F1000.:			
	G01 X10.2 F	500.;			
	X-6.325;				
	M30;				
	%				
	正在执行 G01Y	0X0Z0F1000.	;		
	其指标为 B1kP	ointer=2			
	※ 如海德汉 NC 码	马:			
	BEGIN PGM tt	2 MM			
	BLK FORM 0.1	Z X+0 Y+0	) Z+0		
	BLK FORM 0.2	X+333 Y+3	333 Z+333		
	TOOL CALL 1	Z S2000			
	LBL 1				
	L X+0 Y+0	Z+100 R0 FM	АХ МЗ		
	L X+100 Y+	100 Z+0 FMA	Х		
	LBL Ø				
	CALL LBL 1 R	EP9999			

	END PGM tt2 MM						
	正在执行 L X+0 Y+0 Z+100 R0 FMAX M3						
	其指标为 B1k	Pointer=6					
适用范围	0:支援 X:	未支援 ?::	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	Х	0	?	0	]	
	宝元	工研院					
	0	0					
C#程序范例	StructMsg.Nc	Pointer _NcPc	ointer;				
	StructMsg.Pwd _Pwd;						
	private void button9_Click(object sender, EventArgs e)						
	{						
	_Pwd.Conn	ectionKey =	"123";				
	short ret	= iRemoting	.GET_nc_point	ter(_Pwd, ref	<pre>_NcPointer)</pre>	;	
	if (ret =	= 0)					
	{						
	txtPo	inter.Text =	_NcPointer.B	lkPointer.To	String();		
	}						
	}						

10. GET\_nc\_current\_block: 取得目前程序正在执行及未来要执行的单节

API 名称	GET_nc_curren	t_block					
函式说明	取得目前程序正在执行及未来要执行的单节						
C#方法	short GET_nc_	current_bloc	k(StructMs	g.Pwd P, ref			
	StructMsg.nc_	current_bloc	k R);				
VB.Net 方法							
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。			
	结构名称: nc_c	urrent_block					
	变量名称	变量名称         型别         [in/out]         说明					
	Block	string[]	out	单节数组			
备注	※ 数组中的第6	) 个索引, 即为	目前 CNC 正在	E执行的单节。			
	※ 只读取 256 /	〉字符长度。					
适用范围	0 : 支援 X :	未支援 ?:	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	0	0	?	0		
	宝元	工研院					
	0	Х					
C#程序范例	StructMsg.nc_	current_bloc	k _current	_block;			
	StructMsg.Pwd	_Pwd;					

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

11. GET\_nc\_freespace: 取得剩余空间

API 名称	GET_nc_freesp	ace				
函式说明	取得剩余空间					
C#方法	short GET_nc_	freespace(St	ructMsg.Pw	d P, ref Struct	Msg.nc_freesp	bace
	R);					
VB.Net 方法						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: nc_f	reespace		-		_
	变量名称	型别	[in/out]	说明		
	FreeSpace	int	out	剩余空间		
备注	※ FANUC 及三菱	泛计算单位采用	CHAR			
	※ 海德汉、新代	、宝元及工研	院计算单位采	用 byte		
适用范围	0 : 支援 X :	未支援 ?:	未实作			_
	FANUC	三菱	海德汉	西门子	新代	
	0	0	0	Ş	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.nc_f	reespace _nc	_freespace;			
	StructMsg.Pwd	_Pwd;				
	private void b	utton11_Clic	k(object se	nder, EventArgs	e)	
	{					
	_Pwd.Conne	ctionKey = "1	.23";			
	short ret :	= iRemoting.(	GET_nc_frees	space(_Pwd, ref	_nc_freespace	è);
	if (ret ==	0)				
	{					

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

### 12. SET\_nc\_main: 设定或指定加工程序为主程序

API 名称	SET_nc_main						
函式说明	设定或指定加工利	设定或指定加工程序为主程序					
C#方法	short SET_nc_n	nain(StructM	sg.Pwd P,	StructMsg.NcNam	ne R);		
VB.Net 方法							
数据结构	结构名称: Pwd,	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: NcNar	ne					
	变量名称	型别	[in/out]	说明			
	FolderPath	string	in	文件夹路径			
	Name	string	in	名称			
备注	FANUC 0i-D / 3	31i 控制器:Fo	lderPath 默	认值为//CNC_MEM	/USER/PATH1/		
	※ 必须要设定 A	PI 写入密码(P	wd.WritePw	d),请参阅「四、AP	I联机及写入密	玛亅。	
适用范围	0 : 支援 X : :	未支援 ?: 🗧	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	?	?	?	0		
	宝元	工研院					
	?	?					
C#程序范例	_NcName.Folder	rPath = "//C	NC_MEM/USE	R/PATH1/";			
	_NcName.Name =	= textBox1.T	ext;				
	short ret = i	Remoting.SET	_nc_main(_	Pwd, _NcName);			
	if (ret != 0)						
	MessageBox	.Show(ret.To	oString());	;			

## 13. UPLOAD\_nc\_mdi: 上传加工程序代码到控制器(MDI 模式)

API 名称	UPLOAD_nc_mdi						
函式说明	上传加工程序代码	马到控制器(MD	I 模式)				
C#方法	short UPLOAD_r	<pre>short UPLOAD_nc_mdi(StructMsg.Pwd P, StructMsg.nc_code R);</pre>					
<b>VB.Net</b> 方法							
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。			
	结构名称: nc_co	结构名称: nc_code					
	变量名称	型别	[in/out]	说明			
	FolderPath	string	х	保留,不须输入			
	NcName	string	x	保留,不须输入			

	NcCode	string	<mark>in</mark> j	加工程序代码					
备注	※ 必须要设定 A	※ 必须要设定 API 写入密码(Pwd.WritePwd),请参阅「四、API 联机及写入密码」。							
	※ FANUC 控制器	※ FANUC 控制器在不同系列上,有不同的字数限制:							
	Series 160,	/180/210, 16	0i/180i/210i	, 0i-A/B/C :∄	<b>最长 470</b> 个字				
	Series 300	<b>i, 0i-D :</b> 最长	:500 个字						
	※ 控制器必须切	]换到 MDI 才能	写入加工程序伯	弋码。					
适用范围	0:支援 X:	未支援 ?:	未实作	-					
	FANUC	三菱	海德汉	西门子	新代				
	0	?	?	?	0				
	宝元	工研院							
	?	?							
C#程序范例	string NcCode	= "G00X0.Y0	.Z0.\n" +						
		"F1500.\n'	' +						
		"G01X100.\	/100.Z20.\n"	+					
		"G04X2.";							
	_nc_code.NcCo	de = NcCode;							
	short ret = i	Remoting.UPL	OAD_nc_mdi(	_Pwd, _nc_code	e);				
	if (ret != 0)								
	MessageBox	.Show(ret.To	oString());						

## 十四、 系统参数类别函式

1. GET\_param\_max: 取得参数最大号码(或最后一个号码)

API 名称	GET_param_max						
函式说明	取得参数最大号码(或最后一个号码)						
C#方法	short GET_para	am_max(Struc	tMsg.Pwd P	, ref StructMsg	g.param_max R	);	
VB.Net 方法							
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。			
	结构名称: para	n_max					
	变量名称	型别	[in/out]	说明			
	MaxNumber	int	out	参数最大号码			
备注							
适用范围	0 : 支援 X :	未支援 ?: 🗦	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	0	0	?	0		
	宝元	工研院					
	х	0					
C#程序范例	StructMsg.para	am_max _para	m_max;				
	StructMsg.Pwd	_Pwd;					
	private void b	outton1_Clic	k(object s	ender, EventArg	gs e)		
	{						
	_Pwd.Conne	ctionKey = '	'123" <b>;</b>				
	short ret	= iRemoting	.GET_param_	_max(_Pwd, ref	_param_max);		
	if (ret ==	: 0)					
	{						
	textBo	x1.Text = _p	aram_max.M	axNumber.ToStr	ing();		
	}						
	}						

2. GET\_param\_data: 取得参数资料

API 名称	GET_param_data	3					
函式说明	取得参数资料						
C#方法	short GET_para	<pre>short GET_param_data(StructMsg.Pwd P, StructMsg.param_search_number</pre>					
	SearchNumber,	SearchNumber, ref StructMsg.param_data[] R);					
VB.Net 方法							
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。			
	结构名称: param_search_number						
	变量名称	型别	[in/out]	说明			
	ParamStart	int	in	参数起始号码			

				台北位于日南		
	ParamEnd	int	in	参数终止号码		l
	结构名称: para	n_data				
	变量名称	型别	[in/out]	说明		
	ParamNumber	int	out	参数号码		
	ParamType	short	out	参数数据格式		
				0:binary 1:in	t 2:double	
				3:string		
	ParamMulti	bool	out	判断是否有多维重	或多轴	
	ParamDataS	string[]	out	binary / stri	ng 参数资料	
	ParamDataI	int[]	out	参数资料		
	ParamDataD	double[]	out	参数资料		
备注	※ 所指定的参数	数号码在控制器	上不存在或国	取得失败时,Para	mType 会被设为	-1
适用范围	0 : 支援 X :	未支援 ?:	未实作			1
	FANUC	三菱	海德汉	西门子	新代	
	0	0	0	?	0	
	宝元	工研院				
	X	0				
C#程序范例	StructMsg.para	am_search_nu	mber _Inpu	tNumber;		
	StructMsg.para	am_data[] _p	aram_data;			
	StructMsg.Pwd	_Pwd;				
	private void l	outton2_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_Pwd.Conne	ctionKey = '	"123";			
	DataTable	dt = new Dat	taTable();			
	dt.Columns	.Add("Parami	Num", Type	.GetType("Syste	m.Int32"));	
	dt.Columns	.Add("ParamI	Data", Type	e.GetType("Syst	em.String"));	;
		5		() (c) ( <del>,</del>		
		er.Paramsta	rt = 1nt.Pa	arse(txtStart.)	ext);	
	_inputNumb	- iPomoting	= int.Pars	data ( Dud Ta	j nutNumbon na	<b>۔</b>
		= IKellioting	. de l'_paraiii_	_uata(_Pwu, _IN	iputiumber, re	=1
	_param_uata),	. (0)				
		,				
	L DataRo	w dr:				
	for (i	nt i = 0: i	<= naram	data.length -	1: i++)	
	{	· · · · · ·				
	if	(_param_dat	a[i].Param	Type != -1)		

```
{
        dr = dt.NewRow();
        dr[0] = _param_data[i].ParamNumber;
      if (_param_data[i].ParamMulti == false)
                                                  //没有多轴
        {
            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
        {
            //有多轴
            switch (_param_data[i].ParamType)
            {
                case 0:
for (int j = 0; j <= _param_data[i].ParamDataS.Length - 1; j++)</pre>
                    {
                       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++)</pre>
                    {
                       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++)</pre>
                           {
                               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++)</pre>
                           {
                               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: 设定参数数据

API 名称	SET_param_data
函式说明	设定参数数据
C#方法	<pre>short SET_param_data(StructMsg.Pwd P, StructMsg.param_data[] R);</pre>
VB.Net 方法	

数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。						
	结构名称: para	m_data					
	变量名称	型别	[in/out]	说明			
	ParamNumber	int	in	参数号码			
	ParamType	short	in	参数数据格式			
				0:binary 1:in <sup>.</sup>	t 2:double		
				3:string			
	ParamMulti	bool	in	判断是否有多维重	或多轴		
	ParamDataS	string[]	in	binary / stri	ng 参数资料		
	ParamDataI	int[]	in	参数资料			
	ParamDataD	double[]	in	参数资料			
备注	※ 目前只适用る	车 FANUC 及新作	£				
	※ 新代控制器:	(1)只接受 ir	nt 型别的格式	式写入。 <b>(2)</b> 不支持	事多维及多轴格式。		
适用范围	0:支援 X:	未支援 ?:	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	0	?	?	?	0		
	宝元	工研院					
	X	?					
C#程序范例	StructMsg.para	am_data[] _p	aram_data	= new StructMsg	g.param_data[3];		
	_param_data[0]	].Paramnumbe	r = 162;				
		].Paramiype	= 0;				
	_param_data[0]	].Parammutti	= now stn	ing[] { "00000	×11" L.		
			- 1166 301	ing[] ( 000000	,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,		
	param data[1]	l ParamNumbe	r = 171•				
	param data[1]	].ParamTvpe	= 1:				
	param data[1]	].ParamMulti	= false;				
	param data[1]	].ParamDataI	= new int	[] { 8 };			
	_param_data[2]	].ParamNumbe	r = 1006;				
	_param_data[2]	].ParamType	= 0;				
	_param_data[2]	].ParamMulti	= true;				
	_param_data[2]	].ParamDataS	= new str	ing[3];			
	_param_data[2]	].ParamDataS	[0] = "011	00000";			
	_param_data[2]	].ParamDataS	[1] = "000	00000";			
	_param_data[2]	].ParamDataS	[2] = "000	00000";			
	short ret = il	Remoting.SET	_param_dat	a(_Pwd, _param_	_data);		

MessageBox.Show(ret.ToString());

## 十五、 PLC 类别函式

1. GET\_plc\_ver: 取得 PLC 版本信息

API 名称	GET_plc_ver						
函式说明	取得 PLC 版本信息						
C#方法	<pre>short GET_plc_ver(StructMsg.Pwd P, ref StructMsg.plc_ver R);</pre>						
<b>VB.Net</b> 方法	GET_plc_ver(By	vVal P As Str	uctMsg.Pwd	, ByRef R As St	ructMsg.plc_v	/er)	
	As Short						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。			
	结构名称: plc_v	/er					
	变量名称	型别	[in/out]	说明			
	Version	string	out	版本信息			
	Date	string	out	日期			
	Designed	string	out	设计者			
备注	※ FANUC 控制器	:					
	Version =	PROGRAM DRAW	VING NO.				
	Date = DAT	E OF PROGRAM	1ING				
	Designed =	PROGRAM DES	SIGNED BY				
适用范围	<b>0 :</b> 支援 X : :	未支援 ?: 🕫	未实作		1		
	FANUC	三菱	海德汉	西门子	新代		
	0	0	х	?	0		
	宝元	工研院					
	0	0					
C#程序范例	StructMsg.plc_	ver _plc_ve	r;				
	StructMsg.Pwd	_Pwd;					
	private void b	outton2_Clic	k(object s	ender, EventAr <sub>e</sub>	gs e)		
	{						
	_Pwd.Conne	ctionKey = '	'123" <b>;</b>				
	short ret	= iRemoting.	.GET_plc_ve	er(_Pwd, ref _p	lc_ver);		
	if (ret ==	0)					
	{						
	textBo	<1.Text = _p	lc_ver.Ver	sion;			
	}						
	}						

2. GET\_plc\_addr: 取得 PLC 地址的起始及结束号码

API 名称	GET_plc_addr
函式说明	取得 PLC 地址的起始及结束号码
<b>C#</b> 方法	<pre>short GET_plc_addr(StructMsg.Pwd P, ref StructMsg.plc_addr R);</pre>

VB.Net 方法						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。		
	结构名称: plc_	addr				_
	变量名称	型别	[in/out]	] 说明		
	Addr	string	in	ex:A,D,T,C		
	PlcStart	int[]	out	地址起始值		
	PlcEnd	int[]	out	地址终止值		
备注	※ FANUC 控制器	能接受的地址	(Addr)有:			
	G, F, Y, X,	A, R, T, K,	C, D, E			
	※ 新代控制器能	接受的地址(A	ddr)有:			
	I, O, C, S,	A, R				
适用范围	0:支援 X:	未支援 ?::	未实作			•
	FANUC	三菱	海德汉	西门子	新代	
	0	?	?	?	0	
	宝元	工研院				
	?	?				
C#程序范例	StructMsg.Pwd	_Pwd;				
	StructMsg.plc	_addr _plc_a	ddr;			
	private void	comboBox1_Se	lectedInde	xChanged(objec	t sender,	
	EventArgs e)					
	{					
	txtPlcStar	rt.Text = ""	; txtPlcEnd	1.Text = "";		
	_Pwd.Conne	ectionKey =	"123";			
	_pic_addr.	Addr = comb	OBOX1.lext;	dan (Dud nof	nla adda).	
	if (net	- a)	.GET_PIC_a	uur(_pwu, rer_	_pic_auur),	
		- 0)				
	for (i	nt i = 0: i	<= plc ad	ldr.PlcStart.Le	ngth - 1: i++	+)
	{		· _p=o_oo			/
	txt	tPlcStart.Te	xt += plc	addr.PlcStart	[i].ToString(	) +
				_	3( "\t	.";
	tx	tPlcEnd.Text	+= _plc_ad	dr.PlcEnd[i].T	oString() + "\	\t";
	}					
	}					
	else					
	{					
	Messag	eBox.Show(re	et.ToString	s());		
	}					
}

3. GET\_plc\_addr2: 取得 PLC 地址的起始及结束号码

API 名称	GET_plc_addr2						
函式说明	取得 PLC 地址的起	起始及结束号码	1				
C#方法	short GET_plc_	_addr2(Struc	tMsg.Pwd P	, ref StructMsg	g.plc_addr2 R	);	
VB.Net 方法							
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章 <b>)</b> 。			
	结构名称: plc_a	addr2					
	变量名称	型别	[in/out]	说明			
	Addr	string	in	Plc 地址代号 ex	:A,D,T,C		
	PlcStart	int[]	out	地址起始值			
	PlcEnd	int[]	out	地址终止值			
	РІсТуре	short[]	out	0:byte, 1:shor	rt, 2:int		
				3:bool, 4:doub	le, 5:string		
				6:sbyte			
备注	※ FANUC 控制器	能接受的地址	(Addr)有:				
	G, F, Y, X,	A, R, T, K,	C, D, E				
	※ 新代控制器能	接受的地址(A	ddr)有:				
	I, O, C, S, A, R						
	※ 海德汉控制器(iTNC530)能接受的地址(Addr)有:						
送田范围	M, B, W, D,	」, U, I, L,  土古垤 )・=	5, 1B, 1W <sub>.</sub> 	, ID, OB, OW, O	U		
地口花田		不又版 ! ・ /	海海辺	而门子	新代		
	0	交 >	0	<b>5</b>	0		
	宝元	· 工研院		•			
	?	?					
 C#程序范例	StructMsg.plc	addr2 plc	addr2 = ne	w StructMsg.plo	addr2();		
	_Pwd.Connectio	 onKey = "123	";				
	_plc_addr2.Add	dr = comHeid	Addr.Text;				
	short ret = i	Remoting.GET	_plc_addr2	(_Pwd, ref _plo	_addr2);		
	if (ret == 0)						
	{						
	txtHeidSta	rt.Text = _p	olc_addr2.F	lcStart[0].ToS	tring();		
	txtHeidEnd	I.Text = _plo	c_addr2.Plo	End[0].ToStrin	g();		
	txtHeidTyp	e.Text = _p	lc_addr2.Pl	LcType[0].ToStr	ing();		
	}						

4. \_GET\_plc\_status: 取得机台 PLC 状态信息

API 名称	GET_plc_status					
函式说明	取得机台 PLC 状态	态信息				
C#方法	short GET_plc_	_status(Stru	ctMsg.Pwd I	P, ref StructMs	g.plc_status	R);
VB.Net 方法	GET_plc_status	s(ByVal P As	StructMsg	.Pwd, ByRef R A	ls	
	StructMsg.plc_	_status) As	Short			
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章 <b>)</b> 。		
	结构名称: plc_s	status				
	变量名称	型别	[in/out]	说明		
	Addr	string	in	ex:A,D,T,C		
	PlcStart	int	in	地址起始值		
	PlcEnd	int	in	地址终止值		
	РІсТуре	short	out	0:byte, 1:shor	rt, 2:int	
	PlcDataB	byte[]	out	Plc 资料(byte)		
	PlcDataS	short[]	out	Plc 资料(short)	)	
	PlcDataI	int[]	out	Plc 资料(int)		
备注	※ FANUC 控制器	能接受的地址	(Addr)有:			
	G, F, Y, X,	A, R, T, K,	C, D, E			
	※ 三菱控制器(M	170/M700)能接	受的地址(Ad	ddr)有:		
	X, Y, M, F,	L, SM, D, R	, W, B, SB	, V, SW, SD		
	※ 新代控制器能	接受的地址(A	ddr)有:			
	I, O, C, S,	A, R				
	※宝元控制器能挂	接受的地址(Ad	dr)有:			
	A, R, C, I,	S, O				
	※工研院控制器能	虑接受的地址( <i>/</i>	Addr)有:			
	I, O, C, S,	A, B, 强制 I	点 <b>,</b> 强制 0)	点,强制C点		
	※ 取得值会被放	$\lambda$ <code>PlcData(B</code>	,S,I)的数组	中,如果该号码取	得失败,此数组	值会
	被设为-1					
适用范围	0 : 支援 X : :	未支援 ?: 🗦	未实作			1
	FANUC	三菱	海德汉	西门子	新代	
	0	0	?	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.plc_	_status _plc	_status;			
	StructMsg.Pwd	_Pwd;				
	private void b	outton1_Clic	k(object s	ender, EventArg	gs e)	
	{					
	_plc_statu	s.Addr = "T	". ;			
	_plc_statu	s.PlcStart :	= 2;			

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

5. GET\_plc\_status2: 取得机台 PLC 状态信息

API 名称	GET_plc_status2			
函式说明	取得机台 PLC 状态	信息		
<b>C#</b> 方法	short GET_plc_s	tatus2(Stru	ctMsg.Pwd F	<pre>p, ref StructMsg.plc_status2[]</pre>
	R);			
VB.Net 方法				
数据结构	结构名称: Pwd, 耳	关机密码及写入	、密码设置(请	f参考第三章)。
	结构名称: plc_st	atus2		
	变量名称	型别	[in/out]	说明
	Addr	string	in	Plc 地址代号 ex:A,D,T,C
	PlcNumber	int	in	Plc 编号
	PlcType	short	out	0:byte, 1:short, 2:int
				3:bool, 4:double, 5:string
				6:sbyte
	PlcDataByte	byte	out	Plc 资料(byte)
	PlcDataShort short out Plc资料(short)			
	PlcDataInt	int	out	Plc 资料(int)
	PlcDataBool	bool	out	Plc 资料(bool)
	PlcDataDouble	double	out	Plc 资料(double)
	PlcDataString	string	out	Plc 资料(string)
	PlcDataSByte	sbyte	out	Plc 资料(sbyte)
备注	※ FANUC 控制器能	送接受的地址(A	Addr)有:	
	G, F, Y, X, A	, R, T, K, (	C, D, E	
	※ 三菱控制器(M7	0/M700)能接到	受的地址(Add	dr)有:
	X, Y, M, F, L	, SM, D, R,	W, B, SB,	V, SW, SD

	※ 海德汉控制器(iTNC530)能接受的地址(Addr)有:					
	M, B, W, D, I, O, T, C, S, IB, IW, ID, OB, OW, OD					
	※ 新代控制器能接受的地址(Addr)有:					
	I, O, C, S,	A, R				
	※宝元控制器能	接受的地址(Ad	dr)有:			
	A, R, C, I,	S, 0				
	※工研院控制器	能接受的地址 <b>(</b>	Addr)有:			
	I, O, C, S,	A, B, 强制 I	点,强制0点,	强制 C 点		
适用范围	<b>0</b> :支援 X:	未支援 ?:	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	0	0	?	0	
	宝元	工研院				
	0	0				
<b>C#</b> 程序范例	Pwd.Connecti	onKey = "123	" . ;			
		status2[] _	plc_status2 :	= new		
	StructMsg.plc	_status2[1];				
	_plc_status2[	0].Addr = "W	";			
	_plc_status2[	0].PlcNumber	= 9804;			
	DataTable dt	= new DataTa	ble();			
	dt.Columns.Ad	d("PlcNumber	", Type.GetTy	/pe("System.]	Int32"));	
	dt.Columns.Ad	d("PlcType",	Type.GetType	e("System.Int	:16"));	
	<pre>dt.Columns.Add("PlcValue", Type.GetType("System.String"));</pre>					
	short ret = i	Remoting.GET	_plc_status2	(_Pwd, ref _p	olc_status2);	
	if (ret == 0)					
	{					
	for (int i	i = 0; i <=	_plc_status2.	Length - 1;	i++)	
	{					
	DataRo	w dr = dt.Ne	ewRow();			
	dr["Pl	cNumber"] =	_plc_status2	[i].PlcNumber	r;	
	dr["Pl	cType"] = _p	lc_status2[i	].PlcType;		
	switch	(_plc_statu	us2[i].PlcTyp	e)		
	{					
	ca	se 0:				
		dr["PlcValu	ue"] = _plc_s	tatus2[i].Pl	cDataByte;	
		break;				
	ca	se 1:				

```
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: 写入机台 PLC

API 名称	SET_plc_status	5		
函式说明	写入机台 PLC			
C#方法	short SET_plc_	_status(Stru	ctMsg.Pwd	P, StructMsg.plc_status R);
<b>VB.Net</b> 方法	SET_plc_status	s(ByVal P As	StructMsg	.Pwd, ByVal R As
	StructMsg.plc_	_status) As	Short	
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。
	结构名称: plc_s	status		
	变量名称	型别	[in/out]	说明
	Addr	string	in	ex:A,D,T,C
	PlcStart	int	in	地址起始值
	PlcEnd	int	in	地址终止值
	РІсТуре	short	in	0:byte, 1:short, 2:int
	PlcDataB	byte[]	in	Plc 资料(byte)
	PlcDataS	short[]	in	Plc 资料(short)
	PlcDataI	int[]	in	Plc 资料(int)
备注	※ FANUC 控制器	能接受的地址(	(Addr)有:	

	<ul> <li>G, F, Y, X, A, R, T, K, C, D, E</li> <li>※ 三菱控制器(M70/M700)能接受的地址(Addr)有:</li> <li>X, Y, M, F, L, SM, D, R, W, B, SB, V, SW, SD</li> <li>※ 新代控制器能接受的地址(Addr)有:</li> <li>I, O, C, S, A, R</li> <li>※宝元控制器能接受的地址(Addr)有:</li> </ul>					
	A, R, C, I,	, S, O				
			Addr)有:	現ました。古		
	, 0, C, S, ※ 必须要设定	, A, B, <sub>强</sub> 嗣 I API 写入密码(P	点,强耐U点, wd.WritePwd)	,强耐C点 ,请参阅「四、API	I 联机及写入密码」。	
适用范围	0 : 支援 X :	未支援 ?::	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	0	?	?	?	0	
	宝元	工研院				
	0	0				
C#程序范例	StructMsg.Pwd	_Pwd;				
	private void	button4_Click	(object sende	r, EventArgs	e)	
		ic Adda - "D"				
	_pic_stati	us.Auur = D ; us PlcStart =	15.			
	plc_state	us.PlcEnd = 20	<u>-</u> ,			
	plc state	us.PlcType = 0	); //-:	定要设定成 <b>0(</b> FA	ANUC)	
		us.PlcDataB =	new byte[5];	•	,	
	_plc_state	us.PlcDataB[0]	] = 1;			
	_plc_state	us.PlcDataB[1]	] = 2;			
	_plc_state	us.PlcDataB[2]	] = 3;			
	_plc_state	us.PlcDataB[3]	] = 4;			
	_plc_state	us.PlcDataB[4]	] = 5;			
	_Pwd.Conne	ectionKey = "1	123";			
	short ret	= iRemoting.	SET_plc_status	s(_Pwd, _plc_s	status);	
	if (ret ==	= 0)				
	Messag	geBox.Show("写	入成功!");			
	else					
	Messag	geBox.Show(ret	.ToString());			
	}					

7. SET\_plc\_status2: 写入机台 PLC

API 名称 SET\_plc\_status2

函式说明	写入机台 PLC						
C#方法	short SET_plc_s	tatus2(Stru	ctMsg.Pwd F	<pre>&gt;, StructMsg.pl</pre>	<pre>lc_status2[]</pre>	R);	
VB.Net 方法							
数据结构	结构名称: Pwd, 联	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: plc_st	atus2					
	变量名称	变量名称 型别 [in/out] 说明					
	Addr	string	in	Plc 地址代号 e	x:A,D,T,C		
	PlcNumber	int	in	Plc 编号			
	РІсТуре	short	in	0:byte, 1:shc	ort, 2:int		
				3:bool, 4:dou	ble,5:string		
				6:sbyte			
	PlcDataByte	byte	in	Plc 资料(byte)	)		
	PlcDataShort	short	in	P1c 资料(short	:)		
	PlcDataInt	int	in	Plc 资料(int)			
	PlcDataBool	bool	in	Plc 资料(bool)			
	PlcDataDouble	double	in	Plc 资料(doub]	Le)		
	PlcDataString	string	in	Plc 资料(strir	ng)		
	PlcDataSByte	sbyte	in	Plc 资料(sbyte	2)		
备注	※ FANUC 控制器能	送接受的地址( <i>/</i>	Addr)有:				
	G, F, Y, X, A	, R, T, K,	C, D, E				
	※ 三菱控制器(M7	0/M700)能接到	受的地址(Add	lr)有:			
	X, Y, M, F, L	, SM, D, R,	W, B, SB,	V, SW, SD			
	※ 海德汉控制器(:	iTNC530)能接	受的地址(Ac	ldr)有: (IB 可能	无法写入)		
	M, B, W, D, I	, 0, T, C,	S, IB, IW,	ID, OB, OW, OI	)		
	※新代控制器能接	受的地址(Add	r)有:				
	I, O, C, S, A	,R 〒// ₩ ₩ ₩ • • • •	<i>t</i> -				
	※宝兀控制器能接	受的地址(Add	<b>r)</b> 有:				
		, U 拉亚的山山 / •					
		医文的地址(A	uar)11: 5     迟生lo_占				
	」, U, C, S, A	, D, 浊 时 I ) 「 写 入	ス, 近町 U 点	,	- 联机 乃 写 λ 宓ቭ	冠	
	<ul> <li>、</li></ul>	· 支援 ? ・ 未	· 立作	/, ну мі дія і	城市汉马八山市	-110	
X2/116	FANUC		海德汉	西门子	新代		
	0	 ?	0	?	0		
	宝元	工研院			-		
	0	0					
C#程序范例	_Pwd.Connection	Key = "123"	;	1			
	StructMsg.plc_s	tatus2[] _p	lc_status2	= new			

<pre>StructMsg.plc_status2[1];</pre>
_plc_status2[0].Addr = "W";
_plc_status2[0].PlcNumber = 9804;
_plc_status2[0].PlcType = 1;
_plc_status2[0].PlcDataShort = 20;
<pre>short ret = iRemoting.SET_plc_status2(_Pwd, _plc_status2);</pre>
<pre>MessageBox.Show(ret.ToString());</pre>

# 十六、 档案管理类别函式

1. GET\_file\_directory: 取得控制上的档案及目录

API 名称	GET_file_directory						
函式说明	取得控制上的档案	家及目录					
C#方法	short GET_file	e_directory(S	StructMsg.F	wd P, ref Stru	ctMsg.file_1	ist	
	R);						
VB.Net 方法							
数据结构	结构名称: Pwd,	联机密码及写》	\密码设置 <b>(</b> 词	青参考第三章 <b>)</b> 。			
	结构名称: file_	_list					
	变量名称	型别	[in/out]	说明			
	FolderPath	string	in	文件夹路径			
	DirList	<pre>string[][]</pre>	out	目录数据			
备注	※ 目前仅适用在	海德汉控制器				_	
	※file_list 第二	二个数组:					
	[0]:档案或目:	录名称					
	[1]:大小 (by	te)					
	[2]:日期						
	[3]:权限,会	有3个字符,分	·别是 drw:				
	第1个字符(	代表这个项目是	「文件夹(目	录)或档案」:			
	当为[d]则	当为[d]则是文件夹(目录); 当为[-]则是档案。					
	第2个字符代表这个档案是否「可读」						
	当为[r]则是可读; 当为[-]则是不可读						
	第3个字符代表这个档案是否「可写」						
	当为[w]则是可写; 当为[-]则是不可写						
适用范围	0 : 支援 X : :	未支援 ?:オ	天实作		) and the	l	
	FANUC	三菱	海德汉	西门子	新代		
	?	?	0	?	?		
	宝兀	工研院					
	?	?					
C#桯序范例	StructMsg.file	e_list _file_	_list = new	/ StructMsg.fil	e_list();		
	IVList.Items.	Lear();					
	TILE_LIST.FO	lderPath = t>	Cile diner	;	. (*1. 1***)		
	short ret = $1$	emoting.GEI_	_TILE_dired	tory(_Pwa, ret	(	j	
	IT (Net == 0)						
	1 fon (int i	- 0. i /-	filo lict	Diplict Longth	1· i++)		
		- · · · ·		LISC.LEIIBUI	⊥, ⊥+ <b>⊤</b> )		
	ListVi	ewItem item :	= new				
	ListViewItem(	file list.Di	irList[i][0	));			
	{ ListVi	ewItem item	= new	č			
	ListViewItem(_	_file_list.Di	irList[i][0	);			

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

# 2. UPLOAD\_file2:从 PC 端上传档案到装置(如 CNC 控制器)

API 名称	UPLOAD_file2					
函式说明	从 PC 端上传档案	到装置(如 CNG	2 控制器)			
C#方法	short UPLOAD_1	file2(Struct	Msg.Pwd P,	StructMsg.file	<pre>2_path2 R);</pre>	
VB.Net 方法						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> 订	青参考第三章 <b>)</b> 。		
	结构名称: file_	_path2				
	变量名称	型别	[in/out]	说明		
	FileContent	byte[]	in	从 PC 端上传的档	当案内容	
	DestFile	string	in	该装置上的档案路径及名称		
备注	※ 目前仅适用在海德汉控制器					
适用范围	0:支援 X:	未支援 ?: 🗦	未实作			
	FANUC	三菱	海德汉	西门子	新代	
	?	?	0	?	?	
	宝元	工研院				
	?	?				
C#程序范例	StructMsg.file	e_path _file	_path = new	v StructMsg.fil	<pre>le_path();</pre>	
	_file_path.Sou	urceFile = o	penFileDial	log1.FileName;		
	_file_path.Des	stFile = txt	Path.Text +	+ "\\" +		
	openFileDialo	g1.SafeFileN	ame;			
	short ret = i	Remoting.UPL	OAD_file(_F	wd, _file_path	ı);	

# 3. DOWNLOAD\_file2:从装置上(如 CNC 控制器)下载文件到 PC 端

API 名称	DOWNLOAD_file2				
函式说明	从装置上(如 CNC 控制器)下载文件到 PC 端				
C#方法	<pre>short DOWNLOAD_file2(StructMsg.Pwd P, StructMsg.file_path2 R);</pre>				
VB.Net 方法					
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。				
	结构名称: file_path2				_
	变量名称	型别	[in/out]	说明	

财团法人精密机械研究发展中心 智能整合开发部

	SourceFile	string	in	该装置上的档案	路径及名称		
	FileContent	byte[]	out	下载到 PC 端的棒	当案内容		
备注	※ 目前仅适用在海德汉控制器						
适用范围	0 : 支援 X :	未支援 ?: 🕫	未实作				
	FANUC	三菱	海德汉	西门子	新代		
	?	?	0	?	?		
	宝元	工研院					
	?	?					
C#程序范例	<pre>StructMsg.file_path _file_path = new StructMsg.file_path();</pre>						
	_file_path.SourceFile = txtPath.Text + "\\" +						
	<pre>lvList.SelectedItems[0].Text;</pre>						
	_file_path.De	_file_path.DestFile = saveFileDialog1.FileName;					
	short ret = i	<pre>short ret = iRemoting.DOWNLOAD_file(_Pwd, _file_path);</pre>					

4. DEL\_file: 从装置上删除档案

API 名称	DEL_file					
函式说明	从装置上删除档案					
C#方法	short DEL_fil	<pre>short DEL_file(StructMsg.Pwd P, StructMsg.file_path R);</pre>				
VB.Net 方法						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b> 词	青参考第三章 <b>)</b> 。		
	结构名称: file	_path				
	变量名称	型别	[in/out]	说明		
	SourceFile	string	in	该装置上的档案	路径及名称	
备注	※ 目前仅适用在	※ 目前仅适用在海德汉控制器				
适用范围	0 : 支援 X :	O:支援 X:未支援 ?:未实作				
	FANUC	三菱	海德汉	西门子	新代	
	?	?	0	?	?	
	宝元	工研院				
	?	?				
C#程序范例	StructMsg.fil	<pre>StructMsg.file_path _ file_path = new StructMsg.file_path();</pre>				
	_file_path.So	_file_path.SourceFile = txtPath.Text + "\\" +				
	lvList.Select	edItems[0].T	ext;			
	short ret = i	Remoting.DEL	_file(_Pwd,	file_path);		

- 十七、 National Instruments 9233 加速规模组类别 (暂不开放)
  - 1. START\_ni9233\_get\_data: 设定启动模块进行数据撷取

API 名称	START_ni9233_get_data					
函式说明	设定启动模块进行数据撷取					
C#方法	<pre>short START_ni9233_get_data(StructMsg.Pwd P, StructMsg.NI9233_Param</pre>					
	R);					
VB.Net 方法	START_ni9233_get_	_data(ByVal !	P As Struc	tMsg.Pwd, ByRef R As		
	StructMsg.NI9233_	StructMsg.NI9233_Param) As Short				
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: NI9233_Param					
	变量名称	型别	[in/out]	说明		
	PhysicalChannel	string	in	实体 Channel 名称,如第一个		
				模块第一个 Channel 为		
				Dev1/ai0,同一个模块有两个		
				Channel 为		
				Dev1/ai0,Dev1/ai1		
	MinimumValue	double	in	NI9233 模块所提供的最小电压		
				值		
	MaximumValue	double	in	NI9233 模块所提供的最大电压		
				值		
	Rate	double	in	1秒内有撷取多少 Sample Rate		
	Gain	double[]	in	设定 Gain 值		
	Offset	double[]	in	设定 Offset 值		
备注	※该函式执行后,将会	会持续撷取,直	፤到 STOP_ni	.9233_get_data 函式执行后,才		
	会停止。					
	※错误传回-22, 代表	:传感器已在执行	行,可使用 G	ET_ni9233_spectrum_data 幽式		
	将致据取回。	化武品生产		1) * 住田 057 - * 0000 - 0000 - 000		
	※ 错 厌 传 凹 - 23, 1 \ 衣	传感 奋 內	API 反生钼 医	;,		
注田范国	因 <u></u> 因 日 任 田 左 National					
迫用犯回 ————————————————————————————————————	「 「 「 」 「 」 」 」 」 」 」 」 」 」 」 」 」 」		9233 //1/28///	出佚珇。		
C#程序氾例	StructMsg.N19233_	Param _NI92:	33_Param;			
	Structmsg.rwu _rw	a;	biact sand	on EventAnge a)		
			Dject sena	er, Eventargs ej		
	L Pwd Connecti	onkev - "12:	<b>۲</b> ۳.			
		CHECK nig2:	', 33 running/	(Pwd) == false)		
		·encer_nijzj		(_1 wu) == 1013C)		
	//参数设定					
	_NI9233_Pa	aram.Physica	lChannel =	1		

"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 };
<pre>iRemoting.START_ni9233_get_data(_Pwd, _NI9233_Param);</pre>
}
else
{
//正在执行
MessageBox.Show("目前正在执行");
}
}

# 2. STOP\_ni9233\_get\_data: 停止撷取

API 名称	STOP_ni9233_get_data
函式说明	停止资料撷取
C#方法	<pre>void STOP_ni9233_get_data(StructMsg.Pwd P);</pre>
<b>VB.Net</b> 方法	<pre>STOP_ni9233_get_data()</pre>
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。
备注	
适用范围	只适用在 National Instruments 9233 加速规模组。
C#程序范例	StructMsg.Pwd _Pwd;
	_Pwd.ConnectionKey = "123";
	<pre>iRemoting.STOP_ni9233_get_data();</pre>

# 3. CHECK\_ni9233\_running: 确认是否已启动资料撷取

API 名称	STOP_ni9233_get_data
函式说明	确认是否已启动资料撷取
C#方法	<pre>bool CHECK_ni9233_running(StructMsg.Pwd P);</pre>
VB.Net 方法	CHECK_ni9233_running() As Boolean
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。
备注	
适用范围	只适用在 National Instruments 9233 加速规模组。

C#程序范例	StructMsg.Pwd _Pwd;					
	<pre>private void button4_Click(object sender, EventArgs e)</pre>					
	{					
	_Pwd.ConnectionKey = "123";					
	<pre>if (iRemoting.CHECK_ni9233_running(_Pwd))</pre>					
	MessageBox.Show("目前正在执行中!");					
	else					
	MessageBox.Show("尚未执行!");					
	}					

# 4. GET\_ni9233\_spectrum\_data: 取得资料

API 名称	GET_ni9233_spectrum_data				
函式说明	取得资料				
C#方法	<pre>bool GET_ni9233_spectrum_data(StructMsg.Pwd P, ref</pre>				
	StructMsg.NI9233_SpectrumData R);				
VB.Net 方法	GET_ni9233_spectr	GET_ni9233_spectrum_data(ByVal P As StructMsg.Pwd, ByRef R As			
	StructMsg.NI9233_	StructMsg.NI9233_SpectrumData) As Boolean			
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。				
	结构名称: NI9233_F	Param			
	变量名称	型别	[in/out]	说明	
	Time	DateTime[]	out	每一个取样点的时间	
	Samples	Double[,]	out	二维数组取样值	
备注	Samples 二维数组:	[第n个Sampl	e,第 n 个 Ch	annel 的 Sample 值]	
适用范围	只适用在 National	Instruments	9233 加速规	模组。	
C#程序范例	StructMsg.NI9233_	_SpectrumData	SpectrumD	ata;	
	StructMsg.Pwd _Pwd;				
	<pre>private void button2_Click(object sender, EventArgs e)</pre>				
	{				
	_Pwd.ConnectionKey = "123";				
	DataTable dt	= new DataTa	ble();		
	dt.Columns.Ad	d("Time", Ty	pe.GetType	("System.String"));	
	dt.Columns.Ad	d("Ch1", Typ	e.GetType(	"System.Double"));	
	dt.Columns.Ad	d("Ch2", Typ	e.GetType(	"System.Double"));	
	dt.Columns.Ad	d("Ch3", Typ	e.GetType(	"System.Double"));	
	dt.Columns.Ad	d("Ch4", Typ	e.GetType(	"System.Double"));	
	bool IsSucces	s = iRemotin	g.GET_ni92	33_spectrum_data(_Pwd, ref	
	<pre>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;
}</pre>
```

### 5. GET\_ni9233\_error\_msg: 取得模块错误讯息

API 名称	GET_ni9233_error_msg				
函式说明	取得模块错误讯息				
C#方法	<pre>StructMsg.SkyException GET_ni9233_error_msg(StructMsg.Pwd P);</pre>				
<b>VB.Net</b> 方法	<pre>GET_ni9233_error_msg() As StructMsg.SkyException</pre>				
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。				
	结构名称: SkyException				
	变量名称	型别	[in/out]	说明	
	IsException	bool	out	判断是否发生例外错误	
	ExceptionMsg	string	out	错误讯息	
	ErrorCode	int	out	错误码	
备注					
适用范围	只适用在 National :	Instruments	9233 加速规	模组。	
C#程序范例	StructMsg.Pwd _Pwd;				
	_Pwd.ConnectionKey = "123";				
	<pre>StructMsg.SkyException R = iRemoting.GET_ni9233_error(_Pwd);</pre>				
	MessageBox.Show(R	<pre>MessageBox.Show(R.ExceptionMsg);</pre>			

# 十八、 RS232 模块类别

1. GET\_rs232\_status: 取得目前 RS232 的相关状态及参数

API 名称	GET_rs232_status				
函式说明	取得目前 RS232 的相关状态及参数				
C#方法	<pre>void GET_rs232_status(StructMsg.Pwd P, ref StructMsg.RS232_Param[]</pre>				
	R);				
VB.Net 方法	GET_rs232_status(By	Ref R() As S	StructMsg.	RS232_Param)	
数据结构	结构名称: Pwd,联机密	码及写入密码	设置 <b>(</b> 请参考	第三章)。	
	结构名称: RS232_Para	ım[]			
	变量名称	型别	[in/out]	说明	
	Is0pen	bool	out	是否已经开启 Com Port	
	TaskName	string	out	任务名称	
	RS232_PortName	string	out	Com Port 名称,如:COM1	
	RS232_Parity	Parity	out	同位检查通讯协议	
	RS232_DataBit	int	out	每一字节之数据位的标准长 度	
	RS232_StopBit	StopBits	out	每字节之停止位的标准数目	
	RS232_BaudRate	int	out	设定串行传输速率	
	RS232_Handshake	Handshake	out	数据串行端口传输的交握通	
				讯协议	
	SerialType	short	out	串行端口传入及传回的型别,	
				0:string , 1:byte[]	
	IsOnCallbackEvent	bool	out	是否要挂载串行端口事件传	
				回,如果为true,必须要设定	
				终止符	
	EndText	string	out	字符串的终止符	
	EndByte	byte	out	字节的终止符	
备注	※ SkyMars 允许建立 1	0个数组 <b>(10</b> 绯	且不同串行端	口 Com Port),回传固定长度	
	为10的结构数组。				
	※ 可从各数组的 IsOpe	n判断此串行站	耑口是否正在	使用及 TaskName 的变量。	
适用范围	RS-232 模块或撷取卡。				
C#程序范例	StructMsg.RS232_Par	am[] _Param	;		
	StructMsg.Pwd _Pwd;				
	private void button	6_Click(obj	ect sender	, EventArgs e)	
	{				
	_Pwd.Connection	<pre>Key = "123";</pre>			
	iRemoting.GET_r	s232_status(	_Pwd, ref	_Param);	

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++)
{
<pre>DataRow dr = dt.NewRow();</pre>
dr[0] = _Param[i].IsOpen;
dr[1] = _Param[i].TaskName;
dt.Rows.Add(dr);
}
<pre>dataGridView1.DataSource = dt;</pre>
}

2. OPEN\_rs232: 开启 RS232

API 名称	OPEN_rs232					
函式说明	开启 RS232					
C#方法	short OPEN_rs232(StructMsg.Pwd P, int index, StructMsg.RS232_Param					
	R);					
<b>VB.Net</b> 方法	OPEN_rs232(ByVal in	dex As Integ	ger, ByVal	R As StructMsg.RS232) As		
	Short					
数据结构	结构名称: Pwd,联机密	码及写入密码	设置 <b>(</b> 请参考	第三章 <b>)</b> 。		
	结构名称: RS232_Para	m				
	变量名称	型别	[in/out]	说明		
	TaskName	string	in	设定任务名称		
	RS232_PortName	string	in	设定Com Port 名称,如:COM1		
	RS232_Parity	Parity	in	设定同位检查通讯协议		
	RS232_DataBit	int	in	设定每一字节之数据位的标		
				准长度		
	RS232_StopBit	StopBits	in	设定每字节之停止位的标准		
				数目		
	RS232_BaudRate	int	in	设定串行传输速率		
	RS232_Handshake	Handshake	in	设定数据串行端口传输的交		
				握通讯协议		
	SerialType         short         in         设定串行端口传入及传回的					
				型别,0:string,1:byte[]		
	IsOnCallbackEvent	bool	in	是否要挂载串行端口事件传		
				回,如果为true,必须要设定		

				终止符	
	EndText	string	in	设定字符串的终止符	
	EndByte	byte	in	设定字节的终止符	
备注	※ index:指向 SkyMars 第几个数组(第几组)的串行端口。			行端口。	
	※ SerialType 设为 0	时,即采用 str	ring 方式输入	入串行端口,采用 string 接收。	
	※ IsOnCallbackEven	t 设为 true 时	,必须要再试	殳定 EndText 或 EndByte 变量。	
	※ IsOnCallbackEven	t 设为 false,	可忽略 End	「ext 及 EndByte 变量。	
	※ 假若所指定 index 的	向串行端口已经	开启,将无法	长设定,并送出错误码-25,须先	
	关闭串行端口始可设定。				
适用范围	RS-232 模块或撷取卡。				
C#程序范例	StructMsg.RS232_Par	am _RS232_Pa	aram;		
	<pre>StructMsg.Pwd _Pwd;</pre>				
	private void button	1_Click(obje	ect 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.RS	232_StopBit	= StopBits	.One;	
	_RS232_Param.En	dByte = 0x7c	;		
	_RS232_Param.Is	OnCallbackEv	vent = true	·;	
	_RS232_Param.Se	rialType = 1	;		
	_RS232_Param.Ta	skName = "温	度卡 <b>";</b>		
	_Pwd.Connection	<pre>Key = "123";</pre>			
	short ret = iRe	moting.OPEN_	_rs232(_Pwd	, 0, _RS232_Param);	
	MessageBox.Show	(ret.ToStrin	lg());		
	}				

# 3. SET\_rs232: 将指定的数据写入 RS232

API 名称	SET_rs232			
函式说明	将指定的数据写入 RS232			
C#方法	<pre>short SET_rs232(StructMsg.Pwd P, int index, StructMsg.RS232_Data R);</pre>			
VB.Net 方法	SET_rs232(ByVal P As StructMsg.Pwd, ByVal index As Integer,ByVal R			
	As StructMsg.RS232_Data) As Short			
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: RS232_Data			
	变量名称	型别	[in/out]	说明

	Text	string	in	设定字符串
	ArrayByte	byte[]	in	设定 byte 数组
备注	※ index:指向 SkyMars 第几个数组(第几组)的串行端口。			
	※ 选择 Text 或 Arrayl	Byte 的变量,	请依照之前传	使用的 OPEN_rs232 函式内的
	SerialType 来选择,如	SerialType	=0,请设定 T	ext 变量,ArrayByte 可忽略。
	如果是 SerialType=1,	请设定 Array	'Byte 变量,	Text 可忽略。
	※ 必须要设定 API 写入	、密码(Pwd.Wr:	itePwd),请参	参阅「四、API 联机及写入密码」。
适用范围	RS-232 模块或撷取卡。			
C#程序范例	StructMsg.RS232_Data _RS232_Data;			
	StructMsg.Pwd _Pwd;			
	<pre>private void button2_Click(object sender, EventArgs e)</pre>			
	{			
	byte[] wrCmd = { 0x05, 0x05, 0x05, 0x37 };			
	_RS232_Data.ArrayByte = wrCmd;			
	_Pwd.Connection	Key = "123";		
	short ret = iRe	<pre>moting.SET_r</pre>	s232(_Pwd,	0, _RS232_Data);
	MessageBox.Show	(ret.ToStrir	g());	
	}			

### 4. CHECK\_rs232\_data\_received: 确认是否已经接收到资料(IsOnCallbackEvent=true)

API 名称	CHECK_rs232_data_received
函式说明	确认是否已经接收到资料(IsOnCallbackEvent=true)
C#方法	<pre>bool CHECK_rs232_data_received(StructMsg.Pwd P, int index);</pre>
<b>VB.Net</b> 方法	CHECK_rs232_data_received(index As Integer) As Boolean
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。
备注	※ index:指向 SkyMars 第几个数组(第几组)的串行端口。
适用范围	RS-232 模块或撷取卡。
C#程序范例	StructMsg.Pwd _Pwd;
	_Pwd.ConnectionKey = "123";
	<pre>bool IsRet = iRemoting.CHECK_rs232_data_received(_Pwd, 0);</pre>
	<pre>MessageBox.Show(IsRet.ToString());</pre>

5. GET\_rs232\_data\_received: 取得 RS232 所传回的资料

API 名称	GET_rs232_data_received
函式说明	取得 RS232 所传回的资料
C#方法	<pre>short GET_rs232_data_received(StructMsg.Pwd P, int index, ref</pre>
	StructMsg.RS232_Data R);

VB.Net 方法	GET_rs232_data_received(ByVal index As Integer,ByRef R As			
	StructMsg.RS232_Data) As Short			
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: RS232_Data	I		
	变量名称	型别	[in/out]	说明
	Text	string	out	传回字符串
	ArrayByte	byte[]	out	传回 byte 数组
备注	※ index:指向 SkyMar	s 第几个数组(	第几组 <b>)</b> 的串	行端口。
	※ 选择 Text 或 Arrayl	Byte 的变量,	请依照之前依	使用的
	SerialType 来选择,如	SerialType	=0,由Text	变量取得传回值。
	如果是 SerialType=1,请由 ArrayByte 变量取得传回值。			
适用范围	RS-232 模块或撷取卡。			
C#程序范例	StructMsg.RS232_Data _RS232_Data;			
	StructMsg.Pwd _Pwd;			
	private void button4_Click(object sender, EventArgs e)			
	{			
	_Pwd.ConnectionKey = "123";			
	short ret = iRe	moting.GET_r	s232_data_	_received(_Pwd, 0, ref
	_RS232_Data);			
	byte[] readByte	= _RS232_Da	ata.ArrayBy	/te;
	}			

### 6. CLOSE\_rs232: 关闭 RS232

_	
API 名称	CLOSE_rs232
函式说明	取得 RS232 所传回的资料
C#方法	<pre>void CLOSE_rs232(StructMsg.Pwd P, int index);</pre>
<b>VB.Net</b> 方法	CLOSE_rs232(index As Integer)
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。
备注	※ index:指向 SkyMars 第几个数组(第几组)的串行端口。
适用范围	RS-232 模块或撷取卡。
C#程序范例	StructMsg.RS232_Data _RS232_Data;
	StructMsg.Pwd _Pwd;
	_Pwd.ConnectionKey = "123";
	<pre>iRemoting.CLOSE_rs232(_Pwd, 0);</pre>

7. GET\_rs232\_error: 取得 RS-232 例外错误讯息

API 名称	GET_rs232_error

函式说明	取得 RS-232 例外错误讯息				
C#方法	<pre>StructMsg.SkyException GET_rs232_error_msg(StructMsg.Pwd P, int</pre>				
	index);	index);			
VB.Net 方法	<pre>GET_rs232_error_msg(index As Integer) As StructMsg.SkyException</pre>				
数据结构	结构名称: Pwd, 联机	l密码及写入密	码设置 <b>(</b> 请参	考第三章)。	
	结构名称: SkyExcep	tion			
	变量名称	型别	[in/out]	说明	
	IsException	bool	out	判断是否发生例外错误	
	ExceptionMsg	string	out	错误讯息	
	ErrorCode	int	out	错误码	
备注					
适用范围	RS-232 模块或撷取卡。				
C#程序范例	StructMsg.Pwd _Pwd;				
	_Pwd.ConnectionKe	_Pwd.ConnectionKey = "123";			
	StructMsg.SkyExce	<pre>StructMsg.SkyException R = iRemoting.GET_rs232_error(_Pwd, 0);</pre>			
	MessageBox.Show(R	.ExceptionM	sg +"\r\n"	+ R.ErrorCode.ToString());	

# 十九、 PC Camera 模块类别(USB)

1. CONNECT\_cam: 联机摄影机

API 名称	CONNECT_cam				
函式说明	联机摄影机				
<b>C#</b> 方法	<pre>short CONNECT_cam(StructMsg.Pwd P, StructMsg.Camera_Param R);</pre>				
VB.Net 方法					
数据结构	结构名称: Pwd,	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: Camer	ra_Param			_
	变量名称	型别	[in/out]	说明	
	VideoDevice	int	in	Camera 的 Index(从 0 开始)	
	VideoWidth	int	in	撷取影像宽度	
	VideoHeight	int	in	撷取影像高度	
备注	※影像宽度(Vide	eoWidth)若依	默认值,请设	と定-1	
	※影像高度若依點	犬认值 <b>(Video</b> H	eight),请	设定-1	
	※当联机成功后, CPU 使用率将会升高。				
	※一部计算机只限连接一台摄影机。				
适用范围	USB PC Camera 或 USB Web Camera				
C#程序范例	_Camera_Param.VideoDevice = 0;				
	_Camera_Param.VideoWidth = -1;				
	_Camera_Param.	VideoHeight	= -1;		
	StructMsg.Pwd	_Pwd;			
	_Pwd.Connectio	onKey = "123	";		
	short ret = iF	Remoting.CON	NECT_cam(_	Pwd, _Camera_Param);	
	if (ret == 0)				
	MessageBox	.Show("连接店	成功" <b>);</b>		
	else				
	MessageBox	.Show("连接约	失败");		

## 2. DISCONNECT\_cam: 关闭摄影机

API 名称	DISCONNECT_cam
函式说明	关闭摄影机
C#方法	<pre>void DISCONNECT_cam(StructMsg.Pwd P);</pre>
VB.Net 方法	
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。
备注	

适用范围	USB PC Camera 或 USB Web Camera		
C#程序范例	StructMsg.Pwd _Pwd;		
	_Pwd.ConnectionKey = "123";		
	<pre>iRemoting.DISCONNECT_cam(_Pwd);</pre>		

### 3. CHECK\_cam\_running: 确认摄影机是否已连接及撷取

API 名称	CHECK_cam_running
函式说明	确认摄影机是否已连接及撷取
C#方法	<pre>bool CHECK_cam_running(StructMsg.Pwd P);</pre>
<b>VB.Net</b> 方法	
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。
备注	
适用范围	USB PC Camera 或USB Web Camera
C#程序范例	StructMsg.Pwd _Pwd;
	_Pwd.ConnectionKey = "123";
	<pre>bool IsRunning = iRemoting.CHECK_cam_running(_Pwd);</pre>
	if (IsRunning)
	MessageBox.Show("影像撷取中");
	else
	MessageBox.Show("尚未连接Camera");

# 4. GET\_cam\_image: 取得摄影机影像

API 名称	GET_cam_image					
函式说明	取得摄影机影像					
C#方法	<pre>short GET_cam_image(StructMsg.Pwd P, ref StructMsg.Camera_Image R);</pre>					
<b>VB.Net</b> 方法						
数据结构	结构名称: Pwd,联机	l密码及写入密	码设置 <b>(</b> 请参	考第三章)。		
	结构名称: Camera_I	结构名称: Camera_Image				
	变量名称	变量名称     型别     [in/out]     说明				
	CamImage	byte[]	out	送出影像 byte 数组		
备注	※ 变数 CamImage 转	换后属于 Bitm	nap型别。			
适用范围	USB PC Camera或U	SB Web Camer	a			
C#程序范例	StructMsg.Camera_	<pre>StructMsg.Camera_Image _Camera_Image;</pre>				
	StructMsg.Pwd _Pwd;					
	<pre>private void timer1_Tick(object sender, EventArgs e)</pre>					
	{					
	_Pwd.Connectio	onKey = "123	;			

```
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))
       {
          //设定数据流位置
          oMemoryStream.Position = 0;
          oBitmap = new Bitmap(oMemoryStream);
          //oBitmap =
System.Drawing.Bitmap.FromStream(oMemoryStream);
          pictureBox1.Image = oBitmap;
       }
   }
}
```

#### 5. GET\_cam\_error: 取得摄影机例外错误讯息

API 名称	GET_cam_error					
函式说明	取得摄影机例外错误讯息					
C#方法	<pre>StructMsg.SkyException GET_cam_error(StructMsg.Pwd P);</pre>					
VB.Net 方法						
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: SkyExcep	结构名称: SkyException				
	变量名称         型别         [in/out]         说明					
	IsException         bool         out         判断是否发生例外错误					
	ExceptionMsg	string	out	错误讯息		
	ErrorCode	int	out	错误码		
备注						
适用范围	USB PC Camera 或USB Web Camera					
C#程序范例	StructMsg.Pwd _Pwd;					
	_Pwd.ConnectionKey = "123";					
	StructMsg.SkyExce	ption R = i	Remoting.GI	ET_cam_error(_Pwd);		
	MessageBox.Show(R	.ExceptionMs	sg + "\r\n"	+ R.ErrorCode.ToString());		

二十、 SkyMars Professional 事件派送

SkyMars Professional 本身内建事件派送的架构,就是让应用程序端可向 SkyMars Professional 注册事件,当 SkyMars Professional 有事件发生时,再通知应用程序。

另一方面,应用程序端的事件派送采用不同的线程进行,您必须建立委派(delegate) 来呼叫,以防止跨线程问题。

在建立事件派送前,您必须先建立 EventMsg 类别,并建立实例:

建构函式	<pre>public EventMsg(string IP, int Port);</pre>
参数	IP
	SkyMars 的 IP。应用程序及 SkyMars 都在同一台计算机,请输入「localhost」。
	Port
	SkyMars 的 Port。预设「9500」。

请注意,建议将应用程序及 SkyMars 放置于同一台计算机。如果分开放有可能会造成事件派送传递失败。

```
完整程序代码可参考如下:
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)
{
   //产生 SkyMars Professional 设定 CNC 信息完成事件
   ShowDataDelegate ShowDataDelegate = new ShowDataDelegate(ShowData);
   }
void eMsg_EVENT_disconn_cnc(bool IsExitSkyMars)
{
```

```
财团法人精密机械研究发展中心 智能整合开发部
        //产生 SkyMars Professional 结束 CNC 联机事件
        if (IsExitSkyMars == false)
        {
            ShowDataDelegate _ShowDataDelegate = new ShowDataDelegate(ShowData);
            this.BeginInvoke(_ShowDataDelegate, new object[] { "发生结束 CNC 事件" });
        }
        else
        {
           this.Close();
        }
    }
    void eMsg_EVENT_conn_status(InterfaceLib.StructMsg.SkyConn_status2 R)
    {
        //产生 CNC 状态事件(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)
    {
        //产生 CNC 切换 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: 产生 SkyMars Professional 设定 CNC 信息完成事件

API 名称	EVENT_conn_cnc							
函式说明	产生 SkyMars Professional 设定 CNC 信息完成事件							
C#方法	event EventHandler	_conn_cnc EV	ENT_conn_c	nc;				
VB.Net 方法								
数据结构	结构名称: SkyConn_i	p_port2						
	变量名称	型别	[in/out]	说明				
	MachineNo	int	out	机器的索引值				
	MachineName	MachineName string out 机台名称						
	IP   string   out   机台 IP 地址							
	Port         short         out         机台 Port 号码							
	Manufacturer	string	out	厂牌				
	PlugInPort	int	out	Plug-In Port 号码				
备注								
适用范围	所有控制器							
C#程序范例	1.Create event							
	Wrapper.Local_EVENT_conn_cnc += new							
	<pre>EventHandler_conn_cnc(Wrapper_EVENT_conn_cnc);</pre>							
	<pre>iEvent.EVENT_conn_</pre>	cnc += new						
	Eve	entHandler_co	onn_cnc(Wra	<pre>pper.BC_conn_cnc);</pre>				

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

2. \_EVENT\_disconn\_cnc: 产生 SkyMars Professional 结束 CNC 联机事件

API 名称	EVENT_disconn_cnc				
函式说明	产生 SkyMars Professional 结束 CNC 联机事件				
C#方法	<pre>event EventHandler_disconn_cnc EVENT_disconn_cnc;</pre>				
VB.Net 方法					
数据结构					
	变量名称	型别	[in/out]	说明	
	IsExitSkyMars	bool	out	是否结束 SkyMars 平台	
备注	当结束 SkyMars Profe	essional 平台印	时,事件会派	送至应用程序,IsExitSkyMars	
	变量会传出 true,可依	战此变量来结束	应用程序。		
适用范围	所有控制器				
C#程序范例	1.Create event				
	Wrapper.Local_EVENT_disconn_cnc += new				
	<pre>EventHandler_disconn_cnc(Wrapper_EVENT_disconn_cnc);</pre>				
	iEvent.EVENT_disconn_cnc += new				
	<pre>EventHandler_disconn_cnc(Wrapper.BC_disconn_cnc);</pre>				
	2.Create method				
	<pre>delegate void ShowDataDelegate(string info);</pre>				
	<pre>void Wrapper_EVENT_disconn_cnc(bool IsExitSkyMars)</pre>				
	{				
	ShowDataDelegat	ce _DisplayIn	nfoDelegate	e = new	
			ShowData	aDelegate(ShowData);	
	this.BeginInvol	ke(_DisplayIn	nfoDelegate	e, new object[]	

<pre>{ "Disconnection CNC" });</pre>
}
private void ShowData(string info)
{
richTextBox1.Text += info + "\r\n";
}

# 3. EVENT\_conn\_status: 产生 CNC 状态事件(OFF、RUN、IDLE、ALARM)

API 名称	EVENT_conn_status					
函式说明	产生 CNC 状态事件(OFF、RUN、IDLE、ALARM)					
C#方法	<pre>event EventHandler_conn_status EVENT_conn_status;</pre>					
VB.Net 方法						
数据结构	结构名称: SkyConn_status2					
	变量名称	型别	[in/out]	说明		
	MachineNo	int	out	机器的索引值		
	Status	short	out	0:未开机 1:RUN 2:IDLE		
				3:Alarm		
备注						
适用范围	所有控制器					
C#程序范例	1.Create event					
	Wrapper.Local_EVENT_conn_status += new					
	<pre>EventHandler_conn_status(Wrapper_EVENT_conn_status);</pre>					
	iEvent.EVENT_conn_status += new					
	<pre>EventHandler_conn_status(Wrapper.BC_conn_status);</pre>					
	2.Create method					
	<pre>delegate void ShowDataDelegate(string info);</pre>					
	void Wrapper_EVENT_conn_status(StructMsg.SkyConn_status2 R)					
	{					
	SnowDataDelegate _DisplayIntoDelegate = new					
	ShowDataDelegate(ShowData);					
	string StatusName = "":					
	switch (R.Status)					
	{					
	case 0:					
	Status	Name = "未开	F机";			
	break;					

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

# 4. EVENT\_nc\_filename: 产生 CNC 切换 NC 檔名事件

API 名称	EVENT_nc_filename					
函式说明	产生 CNC 切换 NC 檔名事件					
C#方法	<pre>event EventHandler_conn_status EVENT_conn_status;</pre>					
<b>VB.Net</b> 方法						
数据结构	结构名称: SkyNc	_filename				
	变量名称	型别	[in/out]	说明		
	MachineNo	MachineNo     int     out     机器的索引值				
	MainProg	string	out	主程序文件名		
	SubProg         string         out         子程序文件名					
备注						
适用范围	所有控制器					
C#程序范例	1.Create event					
	Wrapper.Local_EVENT_nc_filename += new					
	<pre>EventHandler_nc_filename(Wrapper_EVENT_nc_filename);</pre>					
	iEvent.EVENT_nc_filename += new					
	Eve	entHandler_n	c_filename	(Wrapper.BC_nc_filename);		
	2.Create metho	od				

# 二十一、 SkyMars 稼动率信息

1. GET\_utilization\_all\_today: 取得今日机台稼动率信息

API 名称	GET_utilization_all_today					
函式说明	取得今日机台稼动率信息					
C#方法	<pre>short GET_utilization_all_today(StructMsg.Pwd P, ref</pre>					
	StructMsg.Utilization_Today R)					
<b>VB.Net</b> 方法						
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: Utilization_Today					
	变量名称型别[in/out]说明					
	PartTotal	int[]	out	今日工件总数		
	CycleTotal	int[]	out	今日加工循环总数		
	BusyTotal	double[]	out	今日加工总时间(单位:小时)		
	IdleTotal	double[]	out	今日闲置总时间(单位:小时)		
	AlarmTotal	double[]	out	今日异常总时间(单位:小时)		
	OffTotal	double[]	out	今日关机总时间(单位:小时)		
备注	※ 请连接 Port	9501 传送及搞	顿取稼动率信.	自。		
	※ BusyTotal 含	育 Busy Time	e 及 BUSY&AL	ARM Time。		
适用范围	所有控制器。					
C#程序范例	StructMsg.Util	ization_Toda	y _Util_Tod	lay;		
	StructMsg.Pwd _Pwd;					
	private void b	utton1_Click	(object sen	nder, EventArgs e)		
	{					
	_Pwd.ConnectionKey = "123";					
	<pre>short ret = iRemoting.GET_utilization_all_today(_Pwd, ref</pre>					
	_Util_Today);					
	if (ret == 0)					
	{					
	<pre>txtMachineName.Text = _Util_Today.MachineName[0];</pre>					
	<pre>txtPartTotal.Text = _Util_Today.PartTotal[0].ToString();</pre>					
	<pre>txtCycleTotal.Text = _Util_Today.CycleTotal[0].ToString();</pre>					
	txtBusy	Total.Text =	_Util_Ioda	ay.Busylotai[0].Tostring();		
	<pre>txtIdleTotal.Text = _Util_Today.IdleTotal[0].ToString();</pre>					
	<pre>txtAlarmiotal.lext = _Util_loday.AlarmTotal[0].ToString(); txtOST_ttal_Tast = _Util_Tast = _SST_t_log_Tast = _SST_t</pre>					
	<pre>ixtorriotal.iext = _util_ioday.urriotal[0].iostring(); }</pre>					
	ر ۲					
	ſ					

2. GET\_utilization\_single\_time: 取得单一机台总时间之稼动率信息

API 名称	GET_utilization_single_time				
函式说明	取得单一机台总时间之稼动率信息				
C#方法	<pre>short GET_utilization_single_time(StructMsg.Pwd P, ref</pre>				
	StructMsg.Utilization R)				
<b>VB.Net</b> 方法					
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。	
	结构名称: Utili	zation			
	变量名称 型别 [in/out] 说明				
	MachineName	string	in	机器名称	
	StDate	DateTime	in	起始日期时间	
	EndDate	DateTime	in	结束日期时间	
	BusyTotal	double	out	加工总时间(单位:小时)	
	IdleTotal	double	out	闲置总时间(单位:小时)	
	AlarmTotal	double	out	异常总时间(单位:小时)	
	OffTotal	double	out	关机总时间(单位:小时)	
备注	※ 请连接 Port	9501 传送及撷	取稼动率信!	也。	
	※ BusyTotal 含	有 Busy Time	及 BUSY&AL	ARM Time。	
	※ 如果要查询同	一天数据,StD	ate 及 EndDa	ate 建议使用完整的日期时间,如:	
	StDate: 2014/5/7 00:00:00.000				
	EndDate: 2014/5/7 23:59:59.999				
适用范围	所有控制器。				
C#程序范例	StructMsg.Utilization _Util;				
	StructMsg.Pwd _Pwd;				
	<pre>private void button2_Click(object sender, EventArgs e)</pre>				
	{				
	_Util.MachineName = "Demo";				
	_Util.StDat	e = DateTime	.Now;		
	_Util.EndDate = DateTime.Now;				
	_Pwd.ConnectionKey = "123";				
	<pre>short ret = iRemoting.GET_utilization_single_time(_Pwd, ref_Util);</pre>				
	if (ret == 0)				
	{				
	<pre>txtBusyTotal2.Text = _Util.BusyTotal.ToString();</pre>				
	<pre>txtIdleTotal2.Text = _Util.IdleTotal.ToString();</pre>				
	<pre>txtAlarmTotal2.Text = _Util.AlarmTotal.ToString();</pre>				
	<pre>txtOffTotal2.Text = _Util.OffTotal.ToString();</pre>				
	}				
	}				

3. GET\_utilization\_single\_total: 取得单一机台总次数(工件数、加工循环次数)之稼动率 信息

API 名称	GET_utilization_single_total				
函式说明	取得单一机台总次数(工件数、加工循环次数)之稼动率信息				
C#方法	<pre>short GET_utilization_single_total(StructMsg.Pwd P, ref</pre>				
	StructMsg.Utilization R)				
<b>VB.Net</b> 方法					
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章)。	
	结构名称: Utilization				
	变量名称	型别	[in/out]	说明	
	MachineName	string	in	机器名称	
	StDate	DateTime	in	起始日期时间	
	EndDate	DateTime	in	结束日期时间	
	PartTotal	int	out	工件总数	
	CycleTotal	int	out	加工循环总数	
备注	※ 请连接 Port	9501 传送及撷	取稼动率信	自己。	
	※ 如果要查询同	一天数据,StD	ate 及 EndDa	ate 建议使用完整的日期时间,如:	
	StDate: 201	4/5/7 00:00	:00.000		
	EndDate: 20	14/5/7 23:59	9:59.999		
适用范围	所有控制器。				
C#程序范例	StructMsg.Util:	ization _Util	l;		
	StructMsg.Pwd _	_Pwd;			
	<pre>private void button3_Click(object sender, EventArgs e)</pre>				
	{				
	_Util.MachineName = "Demo";				
	_Util.StDat	e = DateTime	.Now;		
	_Util.EndDate = DateTime.Now;				
	_Pwd.ConnectionKey = "123";				
	<pre>short ret = iRemoting.GET_utilization_single_total(_Pwd, ref</pre>				
	Util);				
	if (ret == 0)				
	{				
	<pre>txtPartTotal2.Text = _Util.PartTotal.ToString();</pre>				
	txtCycl	eTotal2.Text	= _Util.Cy	<pre>/cleTotal.ToString();</pre>	
	}				
	}				

4.	GET_utilization_	_single_list:	取得单一次	机台稼动率履历信息
----	------------------	---------------	-------	-----------

API 名称	GET_utilization_single_list						
函式说明	取得单一机台稼动率履历信息						
C#方法	<pre>short GET_utilization_single_list(StructMsg.Pwd P, ref</pre>						
	StructMsg.Utili	StructMsg.Utilization_List R)					
<b>VB.Net</b> 方法							
数据结构	结构名称: Pwd,	联机密码及写入	\密码设置 <b>(</b> i	青参考第三章 <b>)</b> 。			
	结构名称: Utili	ization_List					
	变量名称	型别	[in/out]	说明			
	MachineName	MachineName string in 机器名称					
	StDate	DateTime	in	起始日期时间			
	EndDate	DateTime	in	结束日期时间			
	ListData	<pre>string[][]</pre>	out	履历信息			
备注	※ 请连接 Port	9501 传送及撷	取稼动率信息	<b>코</b> .。			
	※ ListData 属于	于双数组 <b>(</b> 不规则	则数组),所位	包含的数组索引识别如下:			
	ListData[	][0]:状态 (OF	F、IDLE、A	LARM、BUSY、BUSY&ALARM)			
	ListData[	][1]:起始时间					
	ListData[	][2]:结束时间					
	ListData[	][3]:总时间(约	吉東时间 – カ	起始时间),单位:小时			
	ListData[][4]:工件数						
	ListData[][5]:加工程序名称						
	ListData[][6]:警报讯息						
	※ 如果要查询同一大数据,StDate 及 EndDate 建议使用完整的日期时间,如:						
	StDate: 2014/5// 00:00:000						
	EndDate: 2014/5// 23:59:59.999						
C#程序范例	<pre>StructMsg.Utilization_List _RList;</pre>						
	StructMsg.Pwd _Pwd;						
	<pre>private void button4_Click(object sender, EventArgs e)</pre>						
	{						
	_RList.MachineName = "Demo";						
	_RList.StDate = DateTime.Now.AddDays(-10);						
	_RList.EndDate = DateTime.Now;						
	_Pwd.Connec	_Pwd.ConnectionKey = "123";					
	<pre>short ret = iRemoting.GET_utilization_single_list(_Pwd, ref</pre>						
	_RList);	_RList);					
	if (ret == 0)						
	{						

```
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++)</pre>
       {
           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;
   }
}
```

5. GET\_utilization\_single\_all: 取得单一机台稼动率信息及履历

	-						
API 名称	GET_utilization_single_all						
函式说明	取得单一机台稼动率信息及履历						
C#方法	<pre>short GET_utilization_single_all(StructMsg.Pwd P, ref</pre>						
	StructMsg.Utilization R , ref StructMsg.Utilization_List RList)						
VB.Net 方法							
数据结构							
	结构名称:Utilization						
	变量名称 型别 [		[in/out]	说明			
	MachineName	string	in	机器名称			
	StDate	DateTime	in	起始日期时间			
	EndDate	DateTime	in	结束日期时间			
	BusyTotal	double	out	加工总时间(单位:小时)			
	IdleTotal	double	out	闲置总时间(单位:小时)			
	AlarmTotal	double	out	异常总时间(单位:小时)			
	OffTotal	double	out	关机总时间(单位:小时)			
----------	--	-----------------------	-------------	----------------------------------	--	--	--
	PartTotal	int	out	工件总数			
	CycleTotal	int	out	加工循环总数			
	结构名称: Utilization_List						
	变量名称	型别	[in/out]	说明			
	ListData	<pre>string[][]</pre>	out	履历信息			
备注	※ 请连接 Port	9501 传送及撷	取稼动率信息	<b>君</b> 。			
	※ ListData 属于	于双数组(不规	则数组),所	包含的数组索引识别如下:			
	ListData[	][0]:状态(0	FF、IDLE、A	ALARM、BUSY、BUSY&ALARM)			
	ListData[	][1]:起始时间	]				
	ListData[	][2]:结束时间	]				
	ListData[	][3]:总时间(	结束时间 - 氵	起始时间),单位:小时			
	ListData[	][4]:工件数					
	ListData[	][5]:加工程序	名称				
	ListData[]	][6]:警报讯息					
	※ 如果要查询问	一大数据,StD	ate	ate 建议使用完整的日期时间,如:			
	StDate: 201	.4/5/7 00:00	:00.000				
	EndDate: 2014/5/7 23:59:59.999						
适用氾围 	│						
C#桯序范例	StructMsg.Utilization _Util;						
	<pre>StructMsg.Utilization_List _RList;</pre>						
	structmsg.Pwd	_PWQ; uttopE_Click	(object con	den EventAnge e)			
	private void button5_Click(object sender, EventArgs e)						
	1 Util MachineName - "Deme":						
		- DateTime	Now AddDay	vs(-10).			
	_Util EndDa	te = Dateline		ys(-10),			
	_Pwd.Connec	tionKey = "1	.23";				
		· iRemoting.0	ET_utilizat	tion_single_all(_Pwd, ref _Util,			
	<pre>ref _RList);</pre>	-					
	if (ret ==	0)					
	{						
	txtBusy	Total2.Text	= _Util.Bus	yTotal.ToString();			
	txtIdle	Total2.Text	= _Util.Idl	.eTotal.ToString();			
	txtAlar	mTotal2.Text	= _Util.Al	.armTotal.ToString();			
	txtOffT	otal2.Text =	_Util.OffT	<pre>Total.ToString();</pre>			
	txtPart	Total2.Text	= _Util.Par	<pre>tTotal.ToString();</pre>			

```
txtCycleTotal2.Text = _Util.CycleTotal.ToString();
       //处理履历数据
       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++)</pre>
       {
          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 维护管理类别

1. GET\_maint\_count: 取得维护管理的总笔数

API 名称	GET_maint_count					
函式说明	取得维护管理的总笔数					
C#方法	short GET_main	t_count(Stru	uctMsg.Pwd	P, ref StructMsg.total_count R)		
VB.Net 方法						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。		
	结构名称: tota]	_count				
	变量名称	型别	[in/out]	说明		
	Count	short	out	总笔数		
备注	※ 请连接 Port	9501 传送及撷	取维护管理(	言息。		
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd _Pwd;					
	StructMsg.tota	<pre>StructMsg.total_count _total_count;</pre>				
	<pre>private void button1_Click(object sender, EventArgs e)</pre>					
	{					
	1	<pre>short ret = iRemoting.GET maint count( Pwd, ref total count):</pre>				
	۱ short ret	= iRemoting.	GET_maint_	_count(_Pwd, ref _total_count);		
	۱ short ret MessageBox	= iRemoting. .Show(_total	.GET_maint_ L_count.Cou	_count(_Pwd, ref _total_count); unt.ToString());		

## 2. GET\_maint\_title: 取得维护管理的标题栏文字

API 名称	GET_maint_title					
函式说明	取得维护管理的标题栏文字					
C#方法	short GET_main	t_title(Stru	uctMsg.Pwd	<pre>P, ref StructMsg.Maint_Title R)</pre>		
VB.Net 方法						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。		
	结构名称: Maint	_Title				
	变量名称	型别	[in/out]	说明		
	Title	string[]	out	标题文字		
备注	※ 请连接 Port	9501 传送及撷	取维护管理伯	言息。		
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd	_Pwd;				
	StructMsg.Mair	<pre>StructMsg.Maint_Title _Maint_Title;</pre>				
	<pre>private void button1_Click(object sender, EventArgs e)</pre>					
	{					
	short ret	= iRemoting.	GET_maint_	_title(_Pwd, ref _Maint_Title);		
	for (int i	= 0; i <= _	_Maint_Tit	le.Title.Length - 1; i++)		

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

## 3. GET\_maint\_all: 取得维护管理所有数据

API 名称	GET_maint_all					
函式说明	取得维护管理所有数据					
C#方法	<pre>short GET_maint_all(StructMsg.Pwd P, ref StructMsg.Maint_All R)</pre>					
VB.Net 方法						
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	  结构名称: Maint_All					
	变量名称	型别	[in/out]	说明		
	ItemName	string[]	out	项目名称		
	UseTime	double[]	out	使用时间		
	LifeTime	double[]	out	寿命时间		
	PlcNode	string[]	out	PLC 节点		
	Spec     string[]     out     规格       Supplier     string[]     out     供货商					
	Contact	string[]	out	联络人		
	Telephone	string[]	out	电话		
备注	※ 请连接 Port	9501 传送及损	<b>顺取维护管理</b> (	信息。		
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd	_Pwd;				
	StructMsg.Maint_All _Maint_All;					
	private void LoadMaintData()					
	{					
	<pre>short ret = iRemoting.GET_maint_all(_Pwd, ref _Maint_All);</pre>					
	for (int i	= 0; i <=	_Maint_All	.ItemName.Length - 1; i++)		
	{					
	DataRo	w dr = dtMai	in.NewRow()	);		
	dr["Co	lStatus"] =	Math.Round	l(_Maint_All.UseTime[i] /		
	_Maint_All.Lit	feTime[i] *	100, 3) +	"%";		
	dr["Co	lItem"] = _N	Maint_All.I	[temName[i];		
	dr["Co	lUseTime"] =	= Math.Rour	nd(_Maint_All.UseTime[i], 3);		
	dr["Co	lLiftTime"]	= Math.Rou	<pre>und(_Maint_All.LifeTime[i], 3);</pre>		
	dr["Co	1Plc"] = _Ma	aint_All.Pl	<pre>LcNode[i];</pre>		
	dr["Co	1Spec"] = _N	Maint_All.S	<pre>Spec[i];</pre>		
	dr["Co	1Man"] = _Ma	aint_All.Su	upplier[i];		
	dr["Co	lContact"] =	= _Maint_Al	ll.Contact[i];		

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

### 4. GET\_maint\_all\_time: 取得维护管理项目内的所有时间(使用时间、寿命时间)

API 名称	GET_maint_all_time						
函式说明	取得维护管理项目内的所有时间(使用时间、寿命时间)						
C#方法	<pre>short GET_maint_all_time(StructMsg.Pwd P, ref</pre>						
	StructMsg.Mair	nt_All_Time	R)				
<b>VB.Net</b> 方法							
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章)。			
	结构名称: Maint	_All_Time					
	变量名称	型别	[in/out]	说明			
	ItemName	string[]	out	项目名称			
	UseTime	double[]	out	使用时间			
	LifeTime	double[]	out	寿命时间			
备注	※ 请连接 Port	9501 传送及撷	取维护管理	<b>信息</b> 。			
适用范围	所有控制器。						
C#程序范例	StructMsg.Pwd _Pwd;						
	StructMsg.Mair	nt_All_Time	_Maint_All	_Time;			
	<pre>private void button1_Click(object sender, EventArgs e)</pre>						
	{						
	<pre>short ret = iRemoting.GET_maint_all_time(_Pwd, ref</pre>						
	_Maint_All_Time);						
	if (ret == 0)						
	{	{					
	for (i	for (int i = 0; i <= dtMain.Rows.Count - 1; i++)					
	{						
	dtM	Nain.Rows[i]	["ColUseTi	me"] =			
		_M	aint_All_T	<pre>ime.UseTime[i];</pre>			
	dtM	lain.Rows[i]	["ColLiftT	ime"] =			
		_M	aint_All_T	<pre>ime.LifeTime[i];</pre>			
	}						
	}						
	}						

5. GET\_maint\_single: 取得单笔的维护管理的项目数据

API 名称	GET_maint_single
--------	------------------

函式说明	取得单笔的维护管理的项目数据					
C#方法	<pre>short GET_maint_single(StructMsg.Pwd P, ref StructMsg.Maint_Single</pre>					
	R)					
VB.Net 方法						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置 <b>(</b>	请参考第三章)。		
	结构名称: Maint	t_Single				
	变量名称	型别	[in/out]	说明		
	ItemName	string	in	项目名称		
	UseTime	double	out	使用时间		
	LifeTime	double	out	寿命时间		
	PlcNode	string	out	PLC 节点		
	Spec	string	out	规格		
	Supplier	string	out	供货商		
	Contact	string	out	联络人		
	Telephone	string	out	电话		
备注	※ 请连接 Port 9501 传送及撷取维护管理信息。					
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd _Pwd;					
	<pre>StructMsg.Maint_Single _Maint_Single;</pre>					
	private void b	outton1_Clic	k(object s	ender, EventArgs e)		
	{					
	_Maint_Sin	gle.ItemName	e = "润滑油	";		
	short ret =	iRemoting.0	GET_maint_s	<pre>single(_Pwd, ref _Maint_Single);</pre>		
	if (ret ==	0)				
	{					
	//					
	}					
	}					

# 6. GET\_maint\_single\_time: 取得单笔维护管理的时间数据(使用时间、寿命时间)

API 名称	GET_maint_single_time				
函式说明	取得单笔维护管理的时间数据(使用时间、寿命时间)				
C#方法	<pre>short GET_maint_single_time(StructMsg.Pwd P, ref</pre>				
	<pre>StructMsg.Maint_Single_Time R)</pre>				
<b>VB.Net</b> 方法					
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。				
	」结构名称: Maint_Single_Time				
	变量名称	型别	[in/out]	说明	

	ItemName	string	in	项目名称		
	UseTime	double	out	使用时间		
	LifeTime	double	out	寿命时间		
备注	※ 请连接 Port	9501 传送及损	取维护管理伯	言息。		
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd _Pwd;					
	<pre>StructMsg.Maint_Single_Time _Maint_Single_Time;</pre>					
	<pre>private void button1_Click(object sender, EventArgs e)</pre>					
	{					
	_Maint_Single.ItemName = "润滑油";					
	<pre>short ret = iRemoting.GET_maint_single_time(_Pwd, ref</pre>					
	_Maint_Single_Time);					
	if (ret == 0)					
	{					
	//					
	}					
	}					

# 7. ADD\_maint\_single: 新增单笔维护管理数据

API 名称	ADD_maint_single					
函式说明	新增单笔维护管理数据					
C#方法	short ADD_mair	nt_single(St	ructMsg.Pw	d P, StructMsg.Maint_Single R)		
<b>VB.Net</b> 方法						
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。		
	结构名称: Maint	_Single				
	变量名称	型别	[in/out]	说明		
	ItemName	string	in	项目名称		
	UseTime	double	in	使用时间		
	LifeTime	double	in	寿命时间		
	PlcNode	string	in	PLC 节点		
	Spec string in 规格					
	Supplier string in 供货商					
	Contact	string	in	联络人		
	Telephone	Telephone string in 电话				
备注	※ 请连接 Port 9501 传送及撷取维护管理信息。					
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd	_Pwd;				
	<pre>StructMsg.Maint_Single _Maint_Single;</pre>					

<pre>private void button1_Click(object sender, EventArgs e)</pre>
{
_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();
<pre>ret = iRemoting.ADD_maint_single(_Pwd, _Maint_Single);</pre>
if (ret != 0)
{
MessageBox.Show("写入失败!");
break;
}
}

### 8. SET\_maint\_single: 修改单笔维护管理的数据

API 名称	SET_maint_single						
函式说明	修改单笔维护管理的数据	居					
C#方法	short SET_maint_sin	gle(StructM	1sg.Pwd P,	<pre>StructMsg.Maint_Single R)</pre>			
VB.Net 方法							
数据结构	结构名称: Pwd,联机密	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: Maint_Sing	le					
	变量名称	型别	[in/out]	说明			
	Original_ItemName	Original_ItemName string in 原始尚未修改的项目名称					
	ItemName	ItemName string in 项目名称					
	UseTime	UseTime double in 使用时间					
	LifeTime double in 寿命时间						
	PlcNode string in PLC节点						
	Spec	Spec string in 规格					
	Supplier	Supplier string in 供货商					
	Contact	string	in	联络人			
	Telephone	string	in	电话			
备注	※ 请连接 Port 9501 传送及撷取维护管理信息。						
适用范围	所有控制器。						

C#程序范例	StructMsg.Pwd _Pwd;						
	<pre>StructMsg.Maint_Single _Maint_Single;</pre>						
	<pre>private void button1_Click(object sender, EventArgs e)</pre>						
	{						
	_Maint_Single.Original_ItemName = dt.Rows[i]["ColItem",						
	<pre>DataRowVersion.Original].ToString();</pre>						
	_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();						
	<pre>ret = iRemoting.SET_maint_single(_Pwd, _Maint_Single);</pre>						
	if (ret != 0)						
	{						
	MessageBox.Show("写入失败!");						
	break;						
	}						
	}						

9. DEL\_maint\_single: 删除单笔维护管理的数据

API 名称	DEL_maint_single					
函式说明	删除单笔维护管理的数据					
C#方法	short DEL_maint_sin	gle(StructM	sg.Pwd P, S	tructMsg.Maint_ItemName R)		
VB.Net 方法						
数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。					
	结构名称: Maint_ItemName					
	变量名称	变量名称         型别         [in/out]         说明				
	Name	string	in	项目名称		
备注	※ 请连接 Port 9501 作	专送及撷取维护	户管理信息。			
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd _Pwd;	StructMsg.Pwd _Pwd;				
	<pre>StructMsg.Maint_ItemName _Maint_ItemName;</pre>					
	private void button	1_Click(obj	ect sender	, EventArgs e)		
	{					
	_Maint_ItemName	.Name = dt.	Rows[i]["C	olItem",		

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

### 10. SET\_maint\_single\_usetime: 设定单笔维护管理的使用时间

API 名称	SET_maint_single_usetime						
函式说明	设定单笔维护管理的使用时间						
C#方法	<pre>short SET_maint_sin</pre>	gle_usetime	e(StructMsg	.Pwd P,			
	StructMsg.Maint_Sin	gle_UseTime	e R)				
VB.Net 方法							
数据结构	结构名称: Pwd,联机密	<b>昭</b> 及写入密码	设置(请参考	着第三章 <b>)</b> 。			
	结构名称: Maint_Sing	gle_UseTime					
	变量名称	型别	[in/out]	说明			
	ItemName	ItemName string in 项目名称					
	UseTime	double	in	使用时间			
备注	※ 请连接 Port 9501 f	专送及撷取维护	户管理信息。				
适用范围	所有控制器。						
C#程序范例	StructMsg.Pwd _Pwd;						
	<pre>StructMsg.Maint_Single_UseTime _Maint_Single_UseTime;</pre>						
	<pre>private void button1_Click(object sender, EventArgs e)</pre>						
	{						
	_Maint_Single_UseTime.ItemName = "润滑油";						
	_Maint_Single_U	seTime.UseT	ime = 10;				
	short ret = iRe	moting.SET_	maint_sing	le_usetime(_Pwd,			
			-	<pre>Maint_Single_UseTime);</pre>			
	}						

### 11. SET\_maint\_single\_zero: 将单笔维护管理的使用时间归零

API 名称	SET_maint_single_zero
函式说明	将单笔维护管理的使用时间归零
<b>C#</b> 方法	<pre>short SET_maint_single_zero(StructMsg.Pwd P,</pre>
	StructMsg.Maint_ItemName R)
VB.Net 方法	

数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。						
	结构名称: Maint_ItemName						
	变量名称	变量名称         型别         [in/out]         说明					
	Name	string	in	项目名称			
备注	※ 请连接 Port 9501 作	专送及撷取维持	户管理信息。				
适用范围	所有控制器。						
C#程序范例	StructMsg.Pwd _Pwd;						
	<pre>StructMsg.Maint_ItemName _Maint_ItemName;</pre>						
	<pre>private void button1_Click(object sender, EventArgs e)</pre>						
	{						
	_Maint_ItemName.Name =						
	<pre>dgvMain.Rows[e.RowIndex].Cells["ColItem"].Value.ToString();</pre>						
	<pre>short ret = iRemoting.SET_maint_single_zero(_Pwd,</pre>						
	Maint_ItemName);						
	if (ret != 0)						
	MessageBox.S	show("归零失	败!");				
	}						

### 二十三、 SkyMars 刀具数据库管理类别

以下表格所列函式皆已过时,我们建议您使用新的函式取代:

已过时的函式	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: 取得刀具数据库管理的总笔数

API 名称	GET_toolmanage	_count			
函式说明	取得刀具数据库管	管理的总笔数			
<b>C#</b> 方法	short GET_tool	.manage_coun	t(StructMs	g.Pwd P, ref	
	StructMsg.tota	al_count R)			
<b>VB.Net</b> 方法					
数据结构	结构名称: Pwd,	联机密码及写	入密码设置(	请参考第三章)。	
	结构名称: total	_count			
	变量名称	型别	[in/out]	说明	
	Count	short	out	总笔数	
备注	※ 请连接 Port	9501 传送及撷	取刀具管理伯	言息。	
适用范围	所有控制器。				
C#程序范例	StructMsg.Pwd _	StructMsg.Pwd _Pwd;			
	StructMsg.tota	<pre>StructMsg.total_count _total_count;</pre>			
	private void b	<pre>private void button1_Click(object sender, EventArgs e)</pre>			
	{	{			
	short ret	<pre>short ret = iRemoting.GET_toolmanage_count(_Pwd, ref</pre>			
	_total_count);				
	MessageBox	.Show(_tota	l_count.Cou	unt.ToString());	
	}				

### 2. GET\_toolmanage\_title2: 取得刀具数据库管理的标题栏文字

API 名称	GET_toolmanage_title2
函式说明	取得刀具数据库管理的标题栏文字
C#方法	<pre>short GET_toolmanage_title2(StructMsg.Pwd P, ref</pre>
	StructMsg.Toolmanage_Title R)
<b>VB.Net</b> 方法	

数据结构	结构名称: Pwd, 联机密码及写入密码设置(请参考第三章)。						
	结构名称: Toolmanage_Title						
	变量名称         型别         [in/out]         说明						
	Title	string[]	out	标题文字			
备注	※ 请连接 Port	9501 传送及撷	取刀具管理	言息。			
	Title 变量数组如	1下:					
	[0] = "ToolNo";						
	<pre>[1] = "ToolType";</pre>						
	<pre>[2] = "CycleTime";</pre>						
	[3] = "LifeTin	ie";					
	[4] = "PartCou	int";					
	[5] = "LifePar	tCount";					
	[6] = "FeedRat	:e";					
	[7] = "Spindle	<pre>[7] = "SpindleSpeed";</pre>					
	[8] = "WorkMaterial";						
	<pre>[9] = "ToolMaterial";</pre>						
	[10] = "Diamet	er";					
	[11] = "ToolLe	ength";					
	<pre>[12] = "Supplier";</pre>						
	[13] = "Contact";						
	[14] = "Telephone";						
适用范围	所有控制器。						
C#程序范例	StructMsg.Pwd _Pwd;						
	<pre>StructMsg.Toolmanage_Title _Toolmanage_Title;</pre>						
	private void button1_Click(object sender, EventArgs e)						
	{						
	short ret	= iRemoting	.GET_toolma	anage_title2(_Pwd, ref			
				_Toolmanage_Title);			
	for (int i	= 0; i <= _	_Toolmanage	e_Title.Title.Length - 1; i++)			
	Message	Box.Show(_T	oolmanage_	Title.Title[i]);			
	}						

# 3. GET\_toolmanage\_all2: 取得刀具数据库管理所有数据

GET_toolmanage_all2
取得刀具数据库管理所有数据
<pre>short GET_toolmanage_all2(StructMsg.Pwd P, ref</pre>
<pre>StructMsg.Toolmanage_Single2[] R);</pre>

数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。				
	结构名称: Toolmanage_Single2				
	变量名称	型别	[in/out]	说明	
	ToolNo	int	out	刀号	
	ToolType	string	out	刀型式	
	CycleTime	double	out	加工时间	
	LifeTime	double	out	使用寿命	
	PartCount	int	out	工件数(已使用)	
	LifePartCount	int	out	工件数(寿命)	
	FeedRate	int	out	Feed Rate	
	SpindleSpeed	int	out	主轴转速	
	WorkMaterial	string	out	工件材质	
	ToolMaterial	string	out	刀具材质	
	Diameter	double	out	刀径	
	ToolLength	double	out	刀长	
	Supplier	string	out	供货商	
	Contact	string	out	联络人	
	Telephone	string	out	电话	
备注	※ 请连接 Port 950	1 传送及撷取	刀具管理信息	0	
适用范围	所有控制器。				
C#程序范例	StructMsg.Pwd _Pw	d;			
	StructMsg.Toolmanage_Single2[] _AllData;				
	private void butt	on1_Click(o	bject send	er, EventArgs e)	
	{				
	<pre>ret = iRemoting.GET_toolmanage_all2(_Pwd, ref _AllData);</pre>				
	if (ret == 0)				
	{				
	for (int i	= 0; i <= _	AllData.Le	ngth - 1; i++)	
	{				
	}				
	}				
	}				

4. GET\_toolmanage\_all\_execute: 取得刀具数据库管理的所有时间及工件数(加工时间、 寿命时间、加工工件数及寿命工件数)

API 名称	GET_toolmanage_all_execute
函式说明	取得刀具数据库管理的所有时间(加工时间、寿命时间、加工工件数及寿命工件数)

C#方法	<pre>short GET_toolmanage_all_execute(StructMsg.Pwd P, ref</pre>			
	StructMsg.Toolm	anage_Singl	e_Execute[	] R);
VB.Net 方法				
数据结构	结构名称: Pwd, 用	关机密码及写入	密码设置(请	f参考第三章 <b>)</b> 。
	结构名称: Toolma	<pre>nage_Single</pre>	_Execute	
	变量名称	型别	[in/out]	说明
	ToolNo	int[]	out	刀号
	CycleTime	double[]	out	加工时间
	LifeTime	double[]	out	寿命时间
	PartCount	int	out	加工工件数(已使用)
	LifePartCount	int	out	工件数(寿命)
备注	※ 请连接 Port 9	501 传送及撷耳	又刀具管理信	息。
适用范围	所有控制器。			
C#程序范例	StructMsg.Pwd _	Pwd;		
	<pre>StructMsg.Toolmanage_Single_Execute[] _AllExecute;</pre>			
	<pre>private void button1_Click(object sender, EventArgs e)</pre>			
	{			
	<pre>short ret = iRemoting.GET_toolmanage_all_execute(_Pwd, ref</pre>			
	_AllExecute);			
	if (ret == 0)			
	{			
	•••			
	}			
	}			

## 5. GET\_toolmanage\_single2: 取得单笔的刀具数据库管理数据

API 名称	GET_toolmanage_single2				
函式说明	取得单笔的刀具数	据库管理数据			
C#方法	short GET_toolm	anage_singl	e2(StructM	sg.Pwd P, ref	
	StructMsg.Toolm	anage_Singl	e2 R);		
VB.Net 方法					
数据结构					
2010-011	结构复杂 Teelma	annan Cinala	、山下100旦(小 つ	19 J/1-+/°	
	结构在你: IOOImanage_Singrez				
	变量名称	型别	[in/out]	说明	
	ToolNo	int	int	刀号	
	ToolType         string         out         刀型式				
	CycleTime double out 加工时间				
	LifeTime	double	out	使用寿命	

	PartCount	int	out	加工工件数(已使用)		
	LifePartCount	int	out	工件数(寿命)		
	FeedRate	int	out	Feed Rate		
	SpindleSpeed	int	out	主轴转速		
	WorkMaterial	string	out	工件材质		
	ToolMaterial	string	out	刀具材质		
	Diameter	double	out	刀径		
	ToolLength	double	out	刀长		
	Supplier	string	out	供货商		
	Contact	string	out	联络人		
	Telephone	string	out	电话		
备注	※ 请连接 Port 9501 传送及撷取刀具管理信息。					
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd _	Pwd;				
	<pre>StructMsg.Toolmanage_Single2 _Single;</pre>					
	private void bu	tton1_Click	(object se	nder, EventArgs e)		
	{					
	_Single.ToolNo = int.Parse(textBox1.Text);					
	short ret = i	Remoting.GE	T_toolmana	<pre>ge_single2(_Pwd, ref _Single);</pre>		
	if (ret == 0)					
	{					
	}					
	}					

6. GET\_toolmanage\_single\_execute: 取得单笔刀具数据库管理的时间及工件数数据(加工时间、寿命时间、加工工件数及寿命工件数)

	,				
API 名称	GET_toolmanage_single_execute				
函式说明	取得单笔刀具数据库管理的时间数据(加工时间、寿命时间、加工工件数及寿命工件				
	(数)				
C#方法	short GET_toolm	anage_singl	e_execute(	StructMsg.Pwd P, ref	
	StructMsg.Toolm	<pre>StructMsg.Toolmanage_Single_Execute R);</pre>			
VB.Net 方法					
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。				
	结构名称: Toolmanage_Single_Execute				
	变量名称	型别	[in/out]	说明	
	ToolNo	int	in	刀号	
	CycleTime	double	out	加工时间	

	LifeTime	double	out	寿命时间	
	PartCount	int	out	加工工件数(已使用)	
	LifePartCount	int	out	工件数(寿命)	
备注	※ 请连接 Port 9	501 传送及撷耳	取刀具管理信	息。	
适用范围	所有控制器。				
C#程序范例	StructMsg.Pwd _	Pwd;			
	StructMsg.Toolm	anage_Singl	e_Execute _	_Execute;	
	private void bu	tton1_Click	(object se	nder, EventArgs e)	
	{				
	<pre>_Execute.ToolNo = int.Parse(textBox1.Text);</pre>				
	<pre>short ret = iRemoting.GET_toolmanage_single_execute(_Pwd, ref</pre>				
	_Execute);				
	if (ret == 0)				
	{				
	}				
	}				

# 7. ADD\_toolmanage\_single2: 新增单笔刀具数据库管理数据

API 名称	ADD_toolmanage_single2				
函式说明	新增单笔刀具数据库管理数据				
C#方法	short ADD_toolm	anage_singl	e2(StructM	sg.Pwd P,	
	StructMsg.Toolm	anage_Singl	e2 R);		
<b>VB.Net</b> 方法					
数据结构	结构名称: Pwd, 耳	关机密码及写入	、密码设置(请	青参考第三章 <b>)</b> 。	
	结构名称: Toolma	nage_Single	2		
	变量名称	型别	[in/out]	说明	
	ToolNo	int	in	刀号	
	ToolType	string	in	刀型式	
	CycleTime	double	in	加工时间	
	LifeTime	double	in	使用寿命	
	PartCount	int	in	加工工件数(已使用)	
	LifePartCount	int	in	工件数(寿命)	
	FeedRate	int	in	Feed Rate	
	SpindleSpeed	int	in	主轴转速	
	WorkMaterial string in 工件材质				
	ToolMaterial	string	in	刀具材质	
	Diameter	double	in	刀径	

	ToolLength	double	in	刀长			
	Supplier	string	in	供货商			
	Contact	string	in	联络人			
	Telephone	string	in	电话			
备注	※ 请连接 Port 9!	501 传送及撷耳	取刀具管理信	息。			
适用范围	所有控制器。						
C#程序范例	StructMsg.Pwd _	Pwd;					
	StructMsg.Toolm	anage_Singl	e2 _Single	;			
	private void bu	tton1_Click	(object se	nder, EventArgs e)			
	{						
	_Single.Too	lNo = 2;					
	_Single.Too	_Single.ToolType = "TEST";					
	_Single.Cycl	_Single.CycleTime = 0;					
	_Single.Life	_Single.LifeTime = 100;					
	_Single.PartCount = 0;						
	_Single.Life	_Single.LifePartCount = 10;					
	_Single.FeedRate = 1000;						
	_Single.SpindleSpeed = 1000;						
	_Single.WorkMaterial = "TEST";						
	_Single.Too	lMaterial =	"TEST";				
	_Single.Dia	meter = 10;					
	_Single.ToolLength = 10;						
	_Single.Supplier = "TEST";						
	_Single.Contact = "XXX";						
	_Single.Telephone = "04-23599009";						
	<pre>short ret = iRemoting.ADD_toolmanage_single2(_Pwd, _Single); MessageBox.Show(ret.ToString());</pre>						
	-						

# 8. \_SET\_toolmanage\_single2: 修改单笔刀具数据库管理的数据

API 名称	SET_toolmanage_single2				
函式说明	修改单笔刀具数据库管理的数据				
C#方法	<pre>short SET_toolmanage_single2(StructMsg.Pwd P,</pre>				
	<pre>StructMsg.Toolmanage_Single2 R);</pre>				
VB.Net 方法					
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。				
	结构名称: Toolmanage_Single2				
	变量名称	型别	[in/out]	说明	

	Original_ToolNo	int	in	原始尚未修改的刀号		
	ToolNo	int	in	刀号		
	ТооlТуре	string	in	刀型式		
	LifeTime	double	in	使用寿命		
	LifePartCount	int	in	工件数(寿命)		
	FeedRate	int	in	Feed Rate		
	SpindleSpeed	int	in	主轴转速		
	WorkMaterial	string	in	工件材质		
	ToolMaterial	string	in	刀具材质		
	Diameter	double	in	刀径		
	ToolLength	double	in	刀长		
	Supplier	string	in	供货商		
	Contact	string	in	联络人		
	Telephone	string	in	电话		
备注	※ 请连接 Port 950	1 传送及撷取	刀具管理信息	•		
	※ 加工时间及工件数	故无法修改。				
适用范围	所有控制器。					
C#程序范例	StructMsg.Pwd _Pw	d;				
	StructMsg.Toolman	age_Single2	_Single;			
	private void button1_Click(object sender, EventArgs e)					
	{					
	_Single.Original_ToolNo = 2;					
	_Single.ToolNo	o = 2;				
	_Single.ToolT	ype = "TEST	";			
	_Single.LifeT:	ime = 200;				
	_Single.LifePa	artCount =	20;			
	_Single.FeedRa	ate = 1000;				
	_Single.Spind	leSpeed = 1	000;			
	_Single.WorkMa	aterial = "	TEST";			
	_Single.ToolMa	aterial = "	TEST";			
	_Single.Diame	ter = 10;				
	_Single.ToolLo	ength = 10;				
	_Single.Suppl:	ier = "TEST	";			
	_Single.Contac	ct = "XXX";				
	_Single.Telep	hone = "04-	23599009";			
	short ret = il	Remoting.SE	T_toolmana	ge_single2(_Pwd, _Single);		
	MessageBox.Sho	ow(ret.ToSt	ring());			
	1					

API 名称	DEL_toolmanage_single			
函式说明	删除单笔刀具数据库管理的数据			
<b>C#</b> 方法	<pre>short DEL_toolmanage_single(StructMsg.Pwd P,</pre>			
	StructMsg.Toolmanage_ToolNo R)			
VB.Net 方法				
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: Toolmanage	_ToolNo		
	变量名称	型别	[in/out]	说明
	No	int	in	刀号
备注	※ 请连接 Port 9501 传送及撷取刀具管理信息。			
适用范围	所有控制器。			
C#程序范例	StructMsg.Pwd _Pwd;			
	<pre>StructMsg.Toolmanage_ToolNo _Toolmanage_ToolNo;</pre>			
	<pre>private void button1_Click(object sender, EventArgs e)</pre>			
	{			
	_Toolmanage_ToolNo.No = 1;			
	<pre>ret = iRemoting.DEL_toolmanage_single(_Pwd,</pre>			
	_Toolmanage_ToolNo);			
	if (ret != 0)			
	{			
	MessageBox.Show("删除失败!");			
	}			
	}			

9. DEL\_toolmanage\_single: 删除单笔刀具数据库管理的数据

10. SET\_toolmanage\_single\_cycletime: 设定单笔刀具数据库管理的加工时间

API 名称	SET_toolmanage_single_cycletime			
函式说明	设定单笔刀具数据库管理的加工时间			
C#方法	<pre>short SET_toolmanage_single_cycletime(StructMsg.Pwd P,</pre>			
	<pre>StructMsg.Toolmanage_Single_CycleTime R)</pre>			
VB.Net 方法				
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: Toolmanage_Single_CycleTime			
	变量名称	型别	[in/out]	说明
	ToolNo	int	in	刀号
	CycleTime	Double	in	加工时间
备注	※ 请连接 Port 9501 传送及撷取刀具管理信息。			

适用范围	所有控制器。			
C#程序范例	StructMsg.Pwd _Pwd;			
	<pre>StructMsg.Toolmanage_Single_CycleTime</pre>			
	_Toolmanage_Single_CycleTime;			
	<pre>private void button1_Click(object sender, EventArgs e)</pre>			
	{			
	_Toolmanage_Single_CycleTime.ToolNo = 1;			
	<pre>short ret = iRemoting.SET_toolmanage_single_cycletime(_Pwd,</pre>			
	_Toolmanage_Single_CycleTime);			
	if (ret != 0)			
	{			
	MessageBox.Show("设定失败!");			
	}			
	}			

11. SET\_toolmanage\_single\_partcount: 设定单笔刀具数据库管理的加工工件数

API 名称	SET_toolmanage_single_partcount			
函式说明	设定单笔刀具数据库管理的工件数			
C#方法	<pre>short SET_toolmanage_single_partcount(StructMsg.Pwd P,</pre>			
	StructMsg.Toolmanag	e_Single_Pa	artCount R)	;
VB.Net 方法				
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: Toolmanage	_Single_Par	rtCount	
	变量名称	型别	[in/out]	说明
	ToolNo	int	in	刀号
	PartCount	Int	in	加工工件数
备注	※ 请连接 Port 9501 传送及撷取刀具管理信息。			
适用范围	所有控制器。			
C#程序范例	StructMsg.Pwd _Pwd;			
	<pre>StructMsg.Toolmanage_Single_PartCount _PartCount;</pre>			
	<pre>private void button1_Click(object sender, EventArgs e)</pre>			
	{			
	_PartCount.ToolNo = 2;			
	_PartCount.PartCount = 30;			
	<pre>short ret = iRemoting.SET_toolmanage_single_partcount(_Pwd,</pre>			
	_PartCount);			
	<pre>MessageBox.Show(ret.ToString());</pre>			
	}			

12. SET\_toolmanage\_single\_zero: 将单笔刀具数据库管理的加工时间归零

API 名称	SET_toolmanage_single_zero			
函式说明	将单笔刀具数据库管理的加工时间归零			
C#方法	<pre>short SET_toolmanage_single_zero(StructMsg.Pwd P,</pre>			
	StructMsg.Toolmanag	;e_ToolNo R)	1	
VB.Net 方法				
数据结构	结构名称: Pwd,联机密	码及写入密码	设置(请参考	育第三章)。
	结构名称: Toolmanage	_ToolNo		
	变量名称	型别	[in/out]	说明
	No	int	in	刀号
备注	※ 请连接 Port 9501 作	专送及撷取刀具	具管理信息。	
适用范围	所有控制器。			
C#程序范例	StructMsg.Pwd _Pwd;			
	<pre>StructMsg.Toolmanage_ToolNo _Toolmanage_ToolNo;</pre>			
	<pre>private void button1_Click(object sender, EventArgs e)</pre>			
	{			
	_Toolmanage_ToolNo.No = 1;			
	<pre>short ret = iRemoting.SET_toolmanage_single_zero(_Pwd,</pre>			
	_Toolmanage_ToolNo);			
	if (ret != 0)			
	MessageBox.Show("归零失败!");			
	}			

13. SET\_toolmanage\_single\_zero\_partcount: 将单笔刀具数据库管理的加工工件数归零

API 名称	SET_toolmanage_single_zero_partcount			
函式说明	将单笔刀具数据库管理的加工工件数归零			
C#方法	<pre>short SET_toolmanage_single_zero_partcount(StructMsg.Pwd P,</pre>			
	<pre>StructMsg.Toolmanage_ToolNo R);</pre>			
VB.Net 方法				
数据结构	结构名称: Pwd,联机密码及写入密码设置(请参考第三章)。			
	结构名称: Toolmanage_ToolNo			
	变量名称	型别	[in/out]	说明
	No	int	in	刀号
备注	※ 请连接 Port 9501 传送及撷取刀具管理信息。			
适用范围	所有控制器。			
C#程序范例	StructMsg.Pwd _Pwd;			
	<pre>StructMsg.Toolmanage_ToolNo _ToolNo;</pre>			
	<pre>private void button1_Click(object sender, EventArgs e)</pre>			

二十四、 应用程序开发者软件防盗版策略

SkyMars Professional 提供两种的软件防盗版策略,允许程序设计师视需要选取最适当应用程序保护技术。不同层级的软件防盗版策略的优点与花费的成本并不相同,所以我们必须了解各种不同的软件防盗版策略的成本及其好处,以便为所开发的应用程序选择最适合的保护机制。

SkyMars Professional 执行时皆必须使用 USB Key,即为 USB 硬件锁,此 USB 硬件 锁内含有两个信息,一个为序号(ID)及授权的联机机台数,可以使用 SKY\_version 函式 来取得,并且该把硬件锁是无法复制及写入。每一把的硬件锁序号,都会不一致,开发 者可以利用这组序号进行验证及锁定。以下介绍两种软件防盗版策略:

1. 依序号(USB Key ID)验证



由于每一把 USB 硬件锁都有唯一的序号,开发者可以事先得知,并写入应用程序代码内或数据库(需编码)。应用程序执行后,可以藉由 SkyMars API 的 SKY\_version 函式来取得序号,并确认此序号为合法,应用程序才可以执行。

2. 将序号(USB Key ID)回传给开发者产生另一组序号验证



第一次执行应用程序时,可以藉由 SkyMars API 的 SKY\_version 函式来取得序 号,并将序号及厂商信息回传给开发者,可以使用 e-mail 或开发者架设注册服务器 等,当开发者收到信息后,开发者可进行认定该厂商是为合法使用者,如果确认为 合法使用者,开发者可以产生另一组序号(自定编码规则或采用加密法则),并传送 给使用者输入序号,应用程序依据序号反推(自定编码规则或采用加密法则)回 USB Key ID,并与 USB 硬件锁的 ID 进行确认,若符合应用程序才可以执行。

- 二十五、 问题排除
  - 1. 问题:应用程序呼叫使用 SkyMars API 时,造成程序发生错误,此错误讯息为「无 法解析至正确方法的调用。」

解决:请更换最新的 InterfaceLib.dll 即可。