EPSON
EXCEED YOUR VISION

EPSON RC+ 7.0 選配

RC+ API 7.0

Rev.21 TCM231S5558F

翻譯版

EPSON RC+ 7.0 選配 RC+ API 7.0 Rev.21

EPSON RC+ 7.0選配

RC+ API 7.0

Rev.21

前言

感謝您選購本公司的機器人產品。
本手冊包含正確使用機器人所需的資訊。
在安裝機器人系統之前，請務必詳閱本手冊及其他相關手冊。
請將本手冊放在方便隨時取用的地方。

所有機器人系統與其選配部件經嚴格的品質控管、測試與檢驗，以確保其符合我們的高效能標準，始能出貨給貴客戶。請注意，若未依本手冊說明的使用條件與產品規格使用本機器人系統，將無法發揮產品的基本性能。

本手冊說明我們可預測的可能危險及後果。務必遵守本手冊的安全注意事項，確保安全及正確地使用機器人系統。

商標

Microsoft、Windows及Windows標誌是Microsoft Corporation在美國及／或其他國家的註冊商標或商標。其他品牌及產品名稱均為其各自持有人的商標或註冊商標。

本手冊中的商標符號

Microsoft® Windows® 8 operating system
Microsoft® Windows® 10 operating system
Microsoft® Windows® 11 operating system
本手冊中的Windows 8、Windows 10和Windows 11分別指上述作業系統。 在某些情況下，Windows通常是指Windows 8、Windows 10和Windows 11。

聲明

未經授權，不得翻印或重製本手冊的任何內容。
本手冊內容如有變更，恕不另行通知。
如發現本手冊有任何錯誤或有對內容任何意見，歡迎與我們聯繫。

製造商

SEIKO EPSON CORPORATION

諮詢服務

如需詳細資訊，請參閱下列手冊第一頁的供應商。

機器人系統 安全手冊 請先行閱讀本手冊

1. 簡介

EPSON RC+ 7.0 選購件 RC+ API 可讓您使用支援.NET 技術的 Microsoft Visual Basic 或其他語言來執行您的機器應用程式。它可讓您建立精密的使用者介面、使用資料庫，以及運用專為搭配.NET 使用所設計的第三方產品。

也包含 LabVIEW 程式庫。

1.1 功能

RC+ API 套件支援以下功能：

- .NET 程式庫與 LabVIEW 程式庫。

- 支援 32 位元與 64 位元應用程式。

- 從多個控制器控制多個機器人、I/O 及任務的屬性與方法。

- 執行視覺及力覺感測*命令的方法。
  * 力覺感測及力覺感測器並不相同。
    API 手冊中所述之力覺感測的方法與屬性不適用於力覺感測器。若要使用力覺感測器的命令，請使用 Xpt 方法執行 SPEL 函數。
    API 不支援 EPSON RC+選購件 Force Guide。

- 支援透過多執行緒並行執行非同步命令。

- 您可透過.NET 應用程式使用多個 EPSON RC+ 7.0 視窗及對話方塊，包括：
  機器人管理器
  IO 監視器
  任務管理器
  模擬器
  控制器工具對話方塊

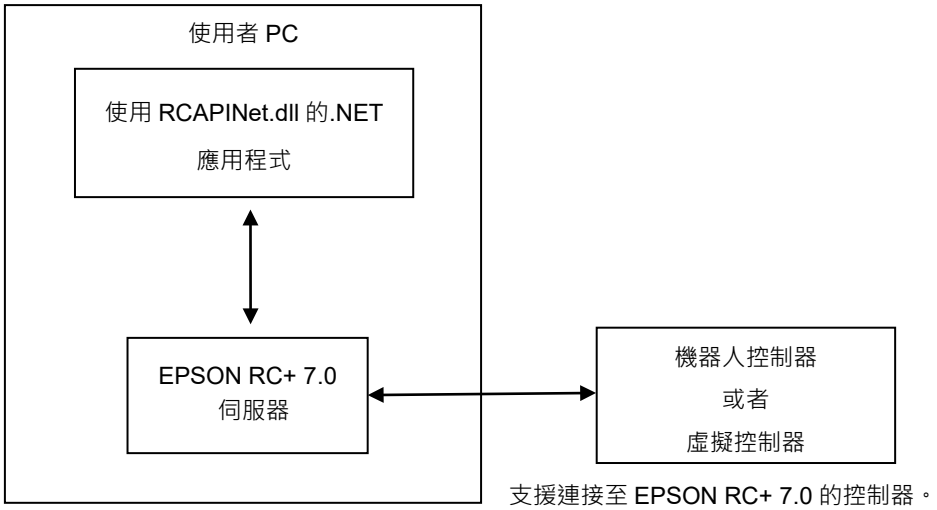
在開發期間，EPSON RC+ 7.0 可以和 Visual Basic 一起執行。
在生產設備中，EPSON RC+ 7.0 可隱藏在背景中執行。

NOTE

RC+ API 支援“.NET Framework Library”與“LabVIEW VI Library”。

不支援“.NET Core”與“.NET”5 或更高版本。

下圖顯示使用 RC+ API 之系統的基本結構。



.NET 程式庫的 RC+ API 基本結構

EPSON RC+ 7.0 是 RCAPINet 程式庫的跨處理序伺服器。

RCAPINet Spel 類別的每個執行個體會啟動一個 EPSON RC+ 7.0 執行個體。

2. 安裝

請依照本章的說明，確保正確安裝 RC+ API 軟體。

在開始之前，請確認已關閉所有 Windows 應用程式。

2.1 逐步說明

(1) 安裝以下之一：
Visual Studio 2012, 2013, 2015, 2017, 2019
(包括 Enterprise、Professional、Community 或 Express Edition)
LabVIEW 2009 或更新版本

(2) 安裝 EPSON RC+ 7.0。

(3) 如果使用 LabVIEW，請安裝 LabVIEW VI 程式庫。

(4) 確定已在您要使用的控制器中啟用 RC+ API 軟體密匙。有關如何在控制器中啟用選購件的詳細資訊，請參閱 EPSON RC+ 7.0 使用指南。

RC+ API 安裝程序到此結束。

2.2 安裝項目

下表顯示的目錄及檔案會在安裝過程中安裝至您的電腦。

目錄與檔案	描述
\EPSONRC70\API\VS20xx\VB\DEMOS	Visual Basic .NET 展示
\EPSONRC70\API\VS20xx\VCS\DEMOS	Visual C# .NET 展示
\EPSONRC70\API\VS20xx\VC\DEMOS	Visual C++ .NET 展示
\EPSONRC70\API\LabVIEW	LabVIEW VI 程式庫安裝程式
\EPSONRC70\PROJECTS\API_Demos	展示用的 EPSON RC+ 7.0 專案
\EPSONRC70\EXE\RCAPINet.dll	RCAPINet 類別程式庫(32 位元或 64 位元)

3. 開始使用

本章包含下列開發環境的開始使用資訊。

- Visual Basic .NET

- Visual C# .NET

- Visual C++ .NET

展示程式隨附於 RC+ API。建議您透過展示程式來更加熟悉本產品。

LabVIEW 使用者現在應參閱第 *16. 使用 LabVIEW VI 程式庫*，瞭解開始使用及使用程式庫的說明。

第一次在 Visual C++ 2017 上建置展示程式時，程式建置可能失敗。程式建置失敗時，請參閱下列章節：

3.4 Visual C++ 2017 上發生建置失敗(MSB8036)的應對措施

在 EPSON RC+7.0 版本或更高版本中使用.NET 應用程式時，.NET Framework 應設置爲 v4.5 或更高版本。

3.1 開始使用Visual Basic

若要在 Visual Basic .NET 專案中使用 RCAPINet，請宣告 Spel 類別執行個體，如以下範例所示。*g_spel* 現在已能在您的專案中使用。

1. 在 Visual Studio .NET 中，選擇檔案 | 專案。
2. 建立 Visual Basic 專案作為 Windows Form 應用程式。
3. 從專案功能表中，選擇添加參考。
4. 在 NET 元件標籤中，瀏覽至\EpsonRC70\Exe 目錄，並選擇 RCAPINet.dll 檔案。
5. 從專案功能表中，建立新的模組並添加以下程式碼。

```
Module Module1
  Public WithEvents g_spel As RCAPINet.Spel
  Public Sub InitApp()
    g_spel = New RCAPINet.Spel
    With g_spel
      .Initialize
      .Project = "c:\EpsonRC70\projects\API_Demos\Demo1 \demo1.sprj"
    End With
  End Sub

  Public Sub EventReceived( _
          ByVal sender As Object, _
          ByVal e As RCAPINet.SpelEventArgs) _
          Handles g_spel.EventReceived

    MsgBox("received event " & e.Event)
  End Sub
End Module
```

NOTE ☞ 當應用程式結束時，您必須針對每個 Spel 類別執行個體執行 Dispose。您可在主要表單的 FormClosed 事件中完成此操作。若沒有執行 Dispose，應用程式將無法正確關閉。

```
g_spel.Dispose()
```

3.2 開始使用Visual C#

1. 在 Visual Studio .NET 中，選擇檔案 | 專案。
2. 建立 Visual C#作為 Windows Form 應用程式。
3. 從專案功能表中，選擇添加參考。
4. 選擇瀏覽標籤，瀏覽至\EpsonRC70\Exe 目錄，並選擇 RCAPINet.dll 檔案。
5. 在 Form1 類別中宣告 Spel 類別變數，如下所示。
   ```
   private RCAPINet.Spel m_spel;
   ```
6. 在 Form_Load 事件中添加初始化程式碼，如下所示。

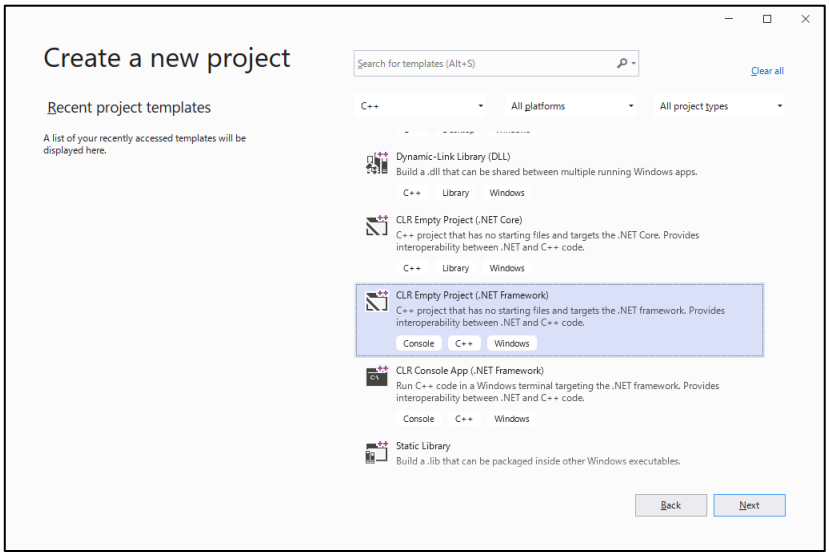
   ```
   private void Form1_Load(object sender, EventArgs e)
   {
      m_spel = new RCAPINet.Spel();
      m_spel.Initialize();

      m_spel.Project =

   "c:\\EpsonRC70\\projects\\API_Demos\\Demo1\\demo1.sprj";

      m_spel.EventReceived += new
         RCAPINet.Spel.EventReceivedEventHandler(m_spel_
   EventReceived);
   ```
7. 添加事件處理常式，如下所示。
   ```
   public void m_spel_EventReceived(object sender,
      RCAPINet.SpelEventArgs e)
   {
   }
   ```

NOTE ☞ 當應用程式結束時，您必須針對每個 Spel 類別執行個體執行 Dispose。您可在主要表單的 FormClosed 事件中完成此操作。若沒有執行 Dispose，應用程式將無法正確關閉。

```
m_spel.Dispose();
```

3.3 開始使用Visual C++

1. 在 Visual Studio .NET 中，選擇[Create a new project]。
2. 選擇 [Visual C++]-[CLR]-[CLR Empty Project (.NET Framework)]。

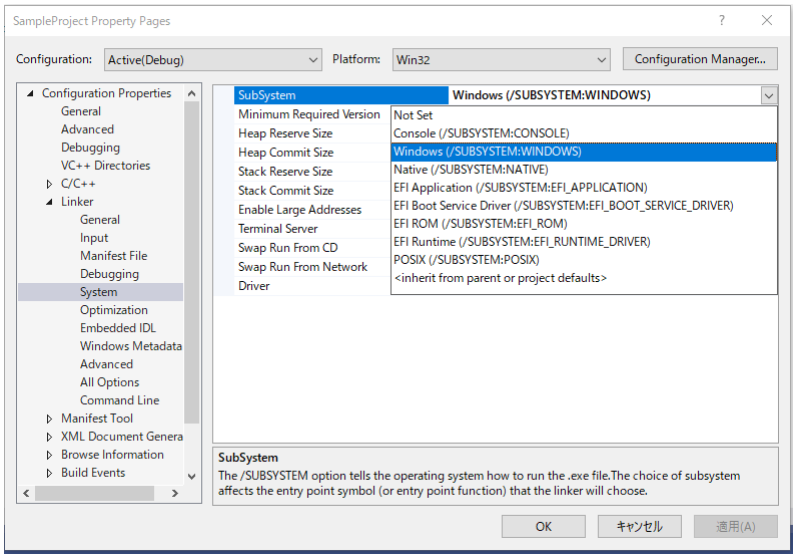


3. 選擇功能表-[Project]-[Add Reference]。
4. 按一下 <Browse>按鈕，瀏覽至“/EpsonRC70/Exe” 目錄，並選擇 “RCAPINet.dll” 檔案。
5. 選擇功能表-[Project]-[Add New Item]-[UI]-[Windows Form]。
6. 打開添加的表單的 cpp 檔案(例: Form1.cpp)，然後添加以下原始碼。

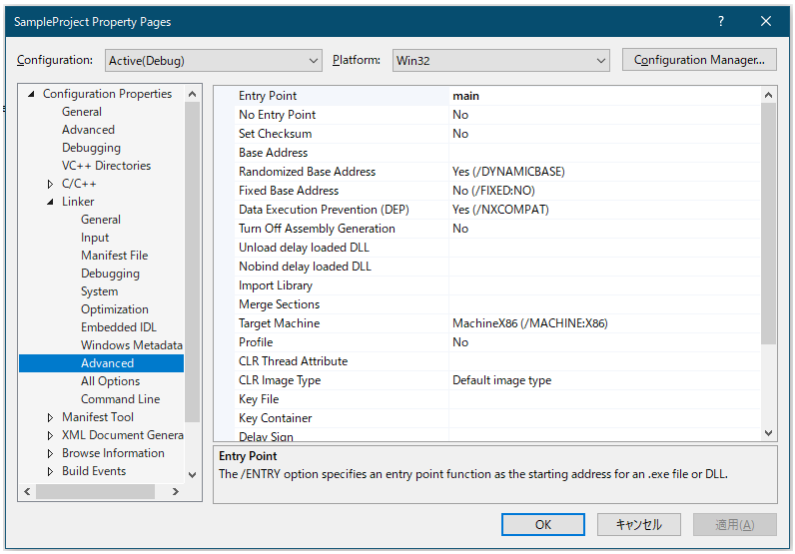
```
#include "Form1.h"
using namespace SampleProject; ←新建專案的名稱
void main() {
        Application::EnableVisualStyles();
        Application::SetCompatibleTextRenderingDefault(false);

        Form1 frm; ← 添加表單的名稱
        Application::Run(% frm);
}
```

7. 選擇功能表- [Project]-[Project Properties].

8. 在屬性頁選擇 [Configuration Properties]-[Linker]-[System]，然後從子系統中選擇“Windows (/SUBSYSTEM:WINDOWS)”。



9. 在屬性頁選擇[Configuration Properties]-[Linker]-[Advanced]，然後在“Entry Point”中鍵入步驟 6 添加的功能的名稱。
在此範例中，鍵入 “main”。



10. 按一下 <OK> 按鈕。

NOTE

設定完成後，建置解決方案，并確認沒有發生錯誤。然後我們建議您關閉解決方案，并重新打開它。

11. 在 Form1 類別中宣告 Spel 變數，如下所示。

```
private RCAPINet::Spel^ m_spel;
```

12. 在 Form_Load 事件中添加初始化程式碼，如下所示。

```
private:System::Void Form1_Load(
  System::Object^ sender, System::EventArgs^ e)
{
  m_spel = gcnew RCAPINet::Spel();
  m_spel->Initialize();
  m_spel->Project =
    “c:\\EpsonRC70\\projects\\ API_Demos\\Demo1\\demo1.sprj";
  m_spel->EventReceived += gcnew
    RCAPINet::Spel::EventReceivedEventHandler(
    this, &Form1::m_spel_EventReceived);
}
```

13. 添加事件處理常式，如下所示。

```
private:System::Void m_spel_EventReceived(
  System::Object^ sender, RCAPINet::SpelEventArgs^ e)
{
  MessageBox::Show(e->Message);
}
```

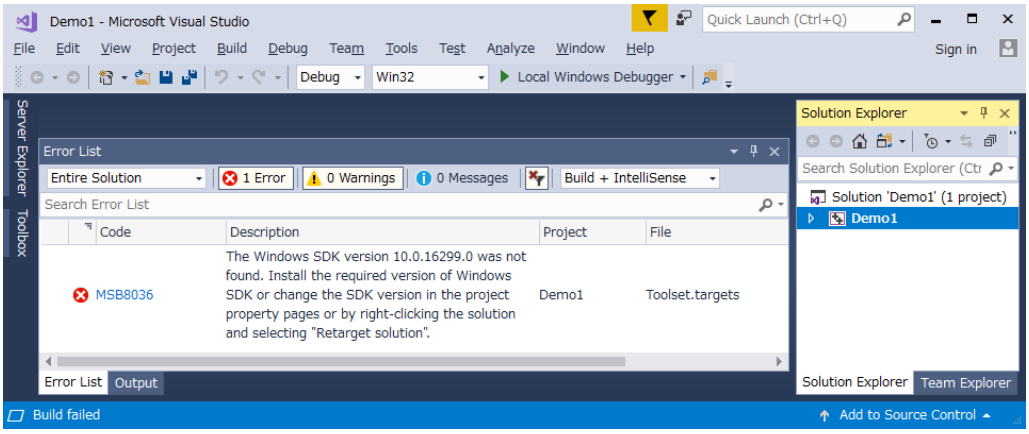
NOTE

當應用程式結束時，若 Spel 類別執行個體是在堆積上分配(使用 gcnew)，則您必須刪除每個 Spel 類別執行個體。您可在主要表單的 FormClosed 事件中完成此操作。若沒有刪除 Spel 類別執行個體，應用程式將無法正確關閉。

```
delete m_spel;
```

3.4 Visual C++ 2017上發生建置失敗(MSB8036)的應對措施

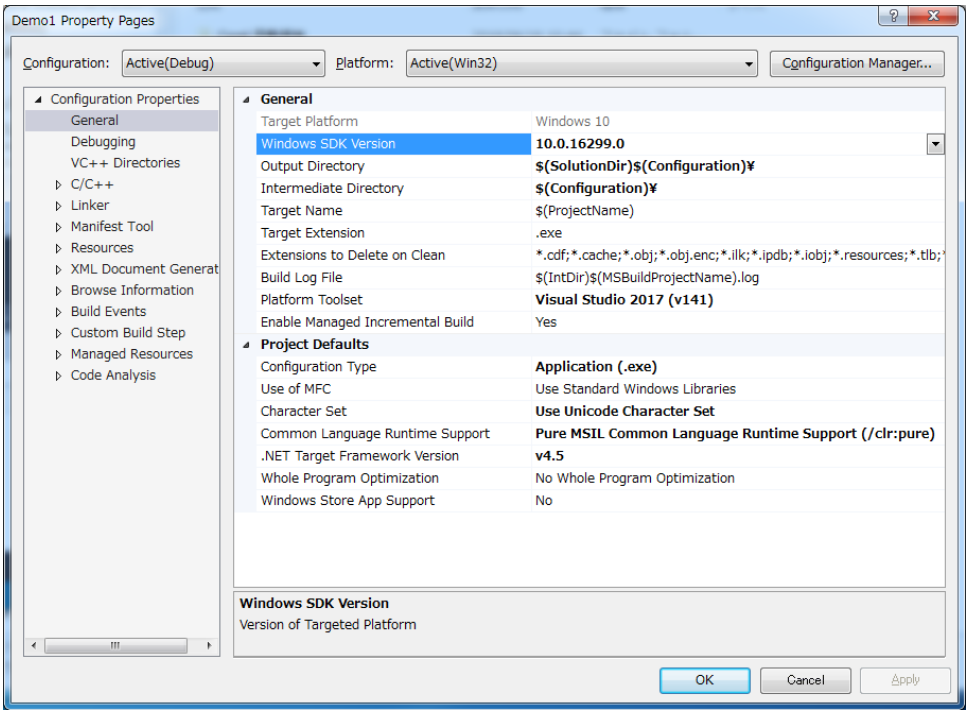
第一次在 Visual C++ 2017 上建置展示程式時，若因錯誤(MSB8036)而發生建置失敗，請執行下列程序：



(1) 選擇 Visual Studio C++ 2017-Solution Explorer-「Demo1」專案。

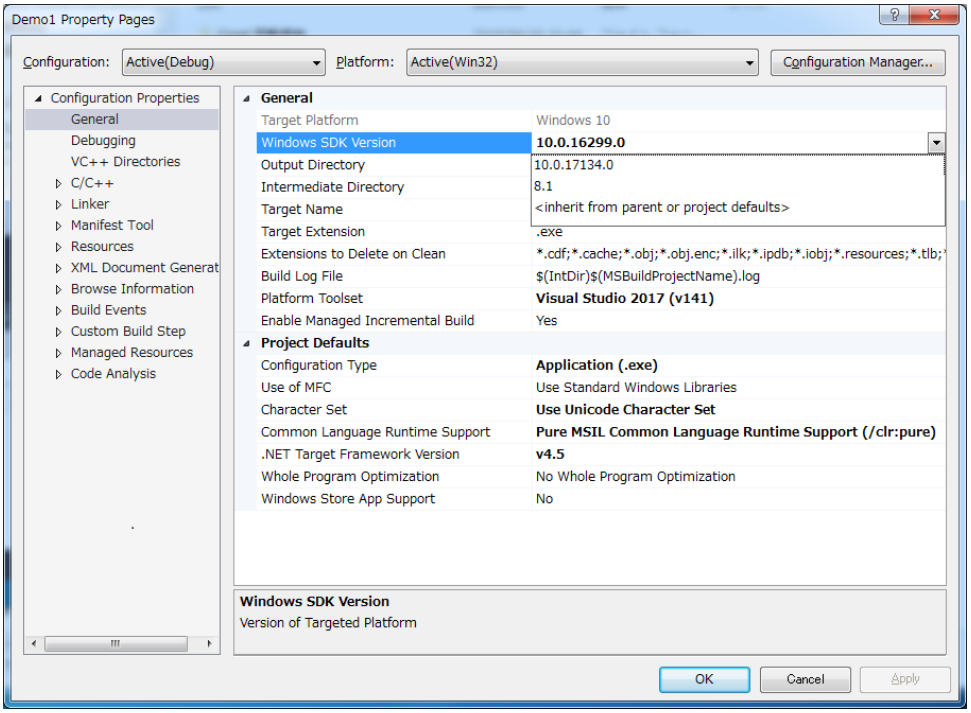
(2) 選擇功能表-[Project]-[Properties]。

(3) 在「Demo1 Proeprty Pages」上選擇[Configuration Properties]-[General]-[Windows SDK Version]。



(4) 按一下「10.0.16299.0」右側的下拉式按鈕。

(5) 選擇在開發環境中安裝的「Windows SDK Version」。



(6) 按一下<OK>按鈕。

(7) 重新建置展示程式。

4. 環境

4.1 開發環境

4.1.1 著手開發

一般而言，您會執行下列步驟開始進行開發：

1. 在.NET 專案的模組中宣告 Spel 類別變數。
2. 啟動 EPSON RC+ 7.0。
3. 開啟所需的 EPSON RC+ 7.0 專案，或建立一新建 EPSON RC+ 7.0 專案。
4. 建置 EPSON RC+ 7.0 專案。
5. 針對 SPEL 類別執行個體，添加初始化程式碼。
6. 執行並偵錯.NET 專案。

4.1.2 Spel類別執行個體初始化

建立 Spel 類別的新建執行個體後，必須予以初始化。執行初始化時，會載入並初始化基本 EPSON RC+ 7.0 模組。在第一個方法調用或屬性存取的情況下，初始化為隱含狀態。您可透過調用 Initialize 方法將類別初始化。

```
m_spel.Initialize()
```

4.1.3 Spel類別執行個體終止

當應用程式結束時，您必須針對每個 Spel 類別執行個體執行 Dispose。您可在主要表單的 FormClosed 事件中完成此操作。若沒有執行 Dispose，應用程式將無法正確關閉。

對於 Visual Basic 及 Visual C#，請使用 Dispose 方法：

```
m_spel.Dispose()
```

對於 Visual C++，若 Spel 類別執行個體是在堆積上建立(使用 gcnew)，則使用 delete：

```
delete m_spel;
```

4.1.4 開發週期

請依照下列基本步驟編輯及執行.NET 程式碼：

1. 停止.NET 專案。
2. 編輯.NET 專案。
3. 開啟 EPSON RC+ 7.0。
4. 在 EPSON RC+ 7.0 專案中做些變更。
5. 建置 EPSON RC+ 7.0 專案。
6. 關閉 RC+ 7.0。
7. 切換至 Visual Studio。
8. 執行.NET 專案。

4.2 於生產設備

4.2.1 在運行時刻開啟EPSON RC+ 7.0

決定是否要從您的應用程式開啟 EPSON RC+ 7.0 環境。此在偵錯時特別實用。將 OperationMode 屬性設至 Program，讓 EPSON RC+ 7.0 進入 Program 模式，並開啟 EPSON RC+ 7.0 GUI。

4.2.2 使用EPSON RC+ 7.0對話方塊及視窗

在運行時刻，您可從.NET 應用程式開啟及隱藏特定的 EPSON RC+ 7.0 視窗。您也可以執行特定的 EPSON RC+ 7.0 對話方塊。
如需詳細資訊，請參閱 *EPSON RC+ 7.0 視窗及對話方塊*章節。

4.2.3 在目標系統上安裝

您應使用 Visual Studio 安裝專案對.NET 專案進行程式安裝，然後依下列步驟設定.NET 應用程式的目標系統：

1. 安裝 EPSON RC+ 7.0。
2. 安裝 EPSON RC+ 7.0 專案。
3. 安裝.NET 應用程式。

5. 執行方法、程式、任務

5.1 執行方法

Spel 類別具有多種方法。如需可用方法的說明，請參閱 *14.3 Spel 類別方法*章節。執行方法時，關聯的內部函數會在 EPSON RC+伺服器程序中調用，接著與控制器進行通信以執行關聯的函數。方法有以下兩種：立即和非同步。使用立即方法時，內部函數會在控制器中執行並且會立即傳回回覆。立即命令包含所有 I/O 命令。使用非同步方法時，關聯的函數會在控制器中啟動，然後 Spel 類別執行個體會等待來自 EPSON RC+伺服器程序的事件指出函數已完成。非同步方法包含所有機器人動作命令。當等待命令完成時，Spel 類別執行個體會發送 Windows 事件，讓使用者 GUI 保持回應。例如：當調用 Go 方法時，機器人會移至指定點，使用者可能會想按一下按鈕來停止移動。您可將 DisableMsgDispatch 設為 True，即可在執行非同步方法時停用 Windows 事件發送。您也可以將 AsyncMode 設為 True，等待非同步方法在程式中完成。

5.1.1 使用多執行緒

您可在應用程式中以多執行緒執行 Spel 方法。下列章節說明各種案例。

一個 Spel 類別執行個體用於多執行緒

您可在多執行緒中使用相同的 Spel 類別執行個體來執行方法，但一次只能執行一個非同步命令。如果您嘗試在一個執行緒中執行非同步命令，但同時已經有另一個非同步命令在其他執行緒中執行，將會發生「命令循環中」錯誤。當在其他執行緒中執行非同步命令時，您可執行立即命令。

獨立 Spel 類別執行個體用於每個執行緒

對於每個控制器連線，您可以有一或多個 Spel 類別執行個體。每個控制器的第一個執行個體會初始化 EPSON RC+ 7.0 伺服器程序，並連接至指定的控制器。若要在其他執行緒使用一或多個其他執行個體和相同的控制器進行通信，您必須將 ServerInstance 屬性指定為相同的值。在使用其他 Spel 類別執行個體前，您需要為第一個執行個體調用 Initialize。

VB 例:

```
' 初始化執行緒 1 的 Spel 類別執行個體
m_spel_1 = New Spel
m_spel_1.ServerInstance = 1
m_spel_1.Initialize()
m_spel_1.Project =
"c:\EpsonRC70\Projects\MyProject\MyProject.sprj"
m_spel_1.Connect(1)

' 初始化執行緒 2 的 Spel 類別執行個體
' 此執行個體使用與 m_spel_1 相同的控制器
m_spel_2 = New Spel
m_spel_2.ServerInstance = 1

執行緒 1
' 將 m_spel_1 執行個體用於動作
m_spel_1.Robot = 1
Do
 m_spel_1.Go(1)
 m_spel_1.Go(2)
Loop Until m_stop

執行緒 2
' 將 m_spel_2 執行個體用於 I/O

Do
 m_spel_2.On(1)
 m_spel_2.Delay(500)
 m_spel_2.Off(1)
 m_spel_2.Delay(500)
Loop Until m_stop
```

C# 例:

```
// 初始化執行緒 1 的 Spel 類別執行個體
RCAPINet.Spel m_spel_1 = new RCAPINet.Spel();
m_spel_1.ServerInstance = 1;
m_spel_1.Initialize();
m_spel_1.Project =
@"c:\EpsonRC70\Projects\MyProject\MyProject.sprj";
m_spel_1.Connect(1);

// 初始化執行緒 2 的 Spel 類別執行個體
// 此執行個體使用與 m_spel_1 相同的控制器
RCAPINet.Spel m_spel_2 = new RCAPINet.Spel();
m_spel_2.ServerInstance = 1;
```

執行緒 1

```
// 將 m_spel_1 執行個體用於動作
m_spel_1.Robot = 1;
do{
 m_spel_1.Go(1);
 m_spel_1.Go(2);
}while(!m_stop);
```

執行緒 2

```
// 將 m_spel_2 執行個體用於 I/O
do{
 m_spel_2.On(1);
 m_spel_2.Delay(500);
 m_spel_2.Off(1);
 m_spel_2.Delay(500);
}while(!m_stop);
```

在控制器中使用 API 執行緒

根據預設，控制器僅支援一個 API 執行緒。在此範例中，即使控制多個機器人，在控制器中一次只執行一個非同步方法。對於使用一個機器人或以 SPEL+任務執行機器人動作的大多數應用，此設定便以足夠，但您可將系統設為最多在控制器中使用 10 個 API 任務，用以平行處理您的.NET 執行緒，例如從相同控制器控制多個機器人時。

若要在控制器中使用多個 API 任務，必須執行兩個基本步驟。

1. 在 EPSON RC+ GUI 中，連接至控制器，然後開啟[Setup]-[System Configuration]-[Controller]-[Preferences]。將「Reserved tasks for API」設為所需的 API 任務數量。請注意，您保留的 API 任務越多，SPEL+程式可使用的任務就越少。例如：如果您保留 5 個 API 任務，則 SPEL+可使用 27 個任務(32 – 5)。
2. 在應用程式中，設定 CommandTask 屬性以指定要執行方法的 API 任務。

在下方的簡易範例中，相同控制器的每個機器人都有一個執行緒。由於在每個執行緒中使用不同的 CommandTask，且兩個 Spel 執行個體的 ServerInstance 皆設為 1，因此機器人動作命令將會並行執行。

VB 例:

```
' 初始化執行緒 1 的 Spel 類別執行個體
m_spel_1 = New Spel
m_spel_1.ServerInstance = 1
m_spel_1.CommandTask = 1
m_spel_1.Initialize()
m_spel_1.Project =
"c:\EpsonRC70\Projects\MyProject\MyProject.sprj"
m_spel_1.Connect(1)

' 初始化執行緒 2 的 Spel 類別執行個體
' 此執行個體使用與 m_spel_1 相同的控制器
' 並在控制器中使用第二個 CommandTask。
m_spel_2 = New Spel
m_spel_2.ServerInstance = 1
m_spel_2.CommandTask = 2

執行緒 1
' 將 m_spel_1 執行個體用於 Robot 1 動作
m_spel_1.Robot = 1
Do
 m_spel_1.Go(1)
 m_spel_1.Go(2)
Loop Until m_stop

執行緒 2
' 將 m_spel_2 執行個體用於 Robot 2 動作
m_spel_2.Robot = 2
Do
 m_spel_2.Go(1)
 m_spel_2.Go(2)
Loop Until m_stop
```

C# 例:

```
// 初始化執行緒 1 的 Spel 類別執行個體
RCAPINet.Spel m_spel_1 = new RCAPINet.Spel();
m_spel_1.ServerInstance = 1;
m_spel_1.CommandTask = 1;
m_spel_1.Initialize();
m_spel_1.Project =
@"c:\EpsonRC70\Projects\MyProject\MyProject.sprj";
m_spel_1.Connect(1);

// 初始化執行緒 2 的 Spel 類別執行個體
// 此執行個體使用與 m_spel_1 相同的控制器
// 並在控制器中使用第二個 CommandTask。
RCAPINet.Spel m_spel_2 = new RCAPINet.Spel();
m_spel_2.ServerInstance = 1;
m_spel_2.CommandTask = 2;

執行緒 1
// 將 m_spel_1 執行個體用於 Robot 1 動作
m_spel_1.Robot = 1;
do{
 m_spel_1.Go(1);
 m_spel_1.Go(2);
}while(!m_stop);

執行緒 2
// 將 m_spel_2 執行個體用於 Robot 2 動作
m_spel_2.Robot = 2;
do{
 m_spel_2.Go(1);
 m_spel_2.Go(2);
}while(!m_stop);
```

5.2 執行SPEL+程式

SPEL+程式包含一或多個函數，並透過啟動其 main 函數來執行程式。您可透過使用 Spel 類別的 *Start* 方法，在目前控制器專案中執行 64 種內建 main 函數的任何一種。您啟動的 main 函數必須在您的 SPEL+程式碼中定義。當您啟動 main 函數時，所有全域變數和模組變數會恢復為預設值。

下表顯示 SPEL+專案中的程式編號及其對應函數名稱。

程式編號	SPEL+函數名稱
0	main
1	main1
2	main2
3	main3
…	…
63	main63

以下是啟動「main」函數的範例：

VB 例:

```
Sub btnStart_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnStart.Click

    m_spel.Start(0) ' 啟動 main 函數
    btnStart.Enabled = False
    btnStop.Enabled = True
End Sub
```

C# 例:

```
void btnStart_Click(object sender, EventArgs e)
{
    m_spel.Start(0); // 啟動 main 函數
    btnStart.Enabled = false;
    btnStop.Enabled  = true;
}
```

5.3 執行SPEL+程式

您可使用 Xqt 方法在 SPEL+程式中執行函數，如同一般任務一樣。當您執行任務時，全域變數不會恢復為您使用 Start 方法時的預設值。

若要暫停及繼續任務，請使用 Halt 及 Resume 方法。

若要結束任務，請使用 Quit 方法。

您也可以使用 StartBGTask 方法啟動控制器背景任務。

5.4 終止所有任務

正在執行任務時，若您想一次終止所有任務，您可使用 Spel 類別的 *Stop* 方法。Stop 方法具有選用參數，可讓您另外停止所有背景任務。

VB 例:

```
Sub btnStop_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnStop.Click
    m_spel.Stop()
    btnStop.Enabled = False
    btnStart.Enabled = True
End Sub
```

C# 例:

```
void btnStop_Click(object sender, EventArgs e)
{
    m_spel.Stop();
    btnStop.Enabled = false;
    btnStart.Enabled = true;
}
```

6. 事件

6.1 概述

Spel 類別支援兩種事件：系統事件和使用者事件。系統事件是系統狀態的通知。使用者定義事件會從任何 SPEL+任務傳送至.NET 應用程式。

6.2 系統事件

有幾種系統事件會傳送至.NET 應用程式。每個系統事件都代表狀態的改變。事件包括 Pause、Continue、Emergency Stop 等。如需所有系統事件的完整資料，請參閱 *14.4 Spel 類別事件 - EventReceived* 的描述。

請使用 Spel 類別 EnableEvents 方法控制要傳送的系統事件。

6.3 來自SPEL+的使用者事件

您可從 SPEL+程式讓事件在.NET 應用程式中發生。例如，您可以將連續循環通知給.NET 應用程序。這種方法比在控制器中從.NET 使用變數值輪詢要來得好。

若要從 SPEL+將事件引發至.NET，請在 SPEL+程式陳述式中使用 SPELCom_Event 命令。例如：

```
    SPELCom_Event 1000, cycNum, lotNum, cycTime
```

SPELCom_Event 命令類似於 Print 命令。您可指定一或多個要傳送至.NET 應用程式的資料。請參閱 *13. SPELCom_Event*，瞭解 SPELCom_Event 的詳細資訊。

在接收事件之前，您必須使用 WithEvents 子句宣告 Spel 類別變數。

```
Public WithEvents m_spel As RCAPINet.Spel
```

針對 Spel 類別執行個體攔截 EventReceived 常式中的事件。若要編輯此常式，請在宣告 Spel 類別的模組中從類別名稱清單選擇「m_spel」，並從程序清單選擇 EventReceived。

以下是發生事件時更新部分標籤的 EventReceived 常式中之程式碼範例。

VB 例:

```
Sub m_spel_EventReceived (ByVal sender As Object, _
ByVal e As RCAPINet.SpelEventArgs) _
Handles m_spel.EventReceived
    Dim tokens() As String
    Select Case e.Event
        Case 2000
           tokens = e.Message.Split(New [Char]() {" "c}, _
System.StringSplitOptions.RemoveEmptyEntries)
           lblCycCount.Text = tokens(0)
           lblLotNumber.Text = tokens(1)
           lblCycTime.Text = tokens(2)
    End Select
End Sub
```

C# 例:

```
void m_spel_EventReceived(object sender, SpelEventArgs e)
{
  string[] tokens = new string[3];
  switch(e.Event)
  {
    case 2000:
              tokens = e.Message.Split(' ');
              lblCycCount.Text = tokens(0);
              lblLotNumber.Text = tokens(1);
              lblCycTime.Text = tokens(2);
              break;
    default:
              break;
  }
}
```

7. 錯誤處理

7.1 Spel方法的錯誤

當您執行 Spel 類別方法時，會在發生錯誤時擲回例外。

發生錯誤時，Spel 類別執行個體會將其擲回調用常式。您應使用應用程式中的錯誤處理常式來攔截此錯誤。在某些情況下，您會只想顯示錯誤訊息。

VB 例:

```
Sub btnStart_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnStart.Click

  Try
    m_spel.Start(0)
  Catch ex As RCAPINet.SpelException
    MsgBox(ex.Message)
  End Try
End Sub
```

您可使用 SpelException 的 ErrorNumber 屬性，檢視有關例外的錯誤編號。

```
  Try
    m_spel.Start(0)
  Catch ex As RCAPINet.SpelException
    MsgBox("SPEL Error: " + ex.ErrorNumber.ToString())
  End Try
```

C# 例:

```
    void btnStart_Click(object sender, EventArgs e)
{
  try{
    m_spel.Start(0);
  }
  catch(SpelException ex){
    MessageBox.Show(ex.Message);
  }
}
```

您可使用 SpelException 的 ErrorNumber 屬性，檢視有關例外的錯誤編號。

```
try {
  m_spel.Start(0);
}
catch(SpelException ex) {
  MessageBox.Show(("SPEL Error: " +
ex.ErrorNumber.ToString());
}
```

8. 處理暫停和繼續

8.1 暫停狀態

發生暫停時，控制器與 SPEL+任務會處於暫停狀態。

在任務執行期間發生下列狀況時，控制器會處於暫停狀態：

- 執行了 Spel 類別 Pause 方法

- SPEL+任務執行了 Pause。

- 安全防護打開。

8.2 攔截Pause事件

Spel 類別會向您的.NET 應用程式通知有暫停狀況發生。

您可在 Spel 類別的 EventReceived 事件中攔截 Pause 事件。

VB 例:

```
Sub m_spel_EventReceived (ByVal sender As Object, ByVal e
As RCAPINet. SpelEventArgs) Handles m_spel.EventReceived
    Select Case e.Event
        Case RCAPINet.SpelEvents.Pause
            btnPause.Enabled = False
            btnContinue.Enabled = True
    End Select
End Sub
```

C# 例:

```
void m_spel_EventReceived(object sender, SpelEventArgs e)
{
  switch(e.Event)
  {
        case SpelEvents.Pause:
            btnPause.Enabled = false;
            btnContinue.Enabled = true;
            break;
        default:
            break;
  }
}
```

8.3 執行暫停

以下常式說明如何使用 *Pause* 方法從 Visual Basic 發出 PAUSE。

VB 例:

```
Sub btnPause_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnPause.Click

    m_spel.Pause()
    btnPause.Enabled = False
    btnContinue.Enabled = True
End Sub
```

C# 例:

```
void btnPause_Click(object sender, EventArgs e)
{
  m_spel.Pause();
  btnPause.Enabled = false;
  btnContinue.Enabled = true;
}
```

8.4 暫停後繼續

若要在發生暫停後繼續，請使用 *Continue* 方法。

VB 例:

```
Sub btnContinue_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnContinue.Click

    m_spel.Continue()
    btnContinue.Enabled = False
    btnPause.Enabled = True
End Sub
```

C# 例:

```
void btnContinue_Click(object sender, EventArgs e)
{
  m_spel.Continue();
  btnContinue.Enabled = false;
  btnPause.Enabled = true;
}
```

8.5 暫停後終止

若不想在暫停後繼續，您也可以執行 *Stop* 方法。

VB 例:

```
Sub btnStop_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnStop.Click

    m_spel.Stop()
    btnContinue.Enabled = False
    btnPause.Enabled = False
End Sub
```

C# 例:

```
  void btnStop_Click(object sender, EventArgs e)
  {
    m_spel.Stop();
    btnContinue.Enabled = true;
    btnPause.Enabled = false;
  }
```

9. 處理緊急停止

發生緊急停止時，您可能會想在程式中執行某些特定操作，例如顯示對話方塊或訊息方塊。

Spel 類別會針對緊急停止狀態發出兩種標準事件：EStopOn 和 EStopOff。

9.1 使用系統EStop事件

您可在 Visual Basic 應用程式的 EventReceived 處理常式中攔截系統 EStop 事件。

```
Imports RCAPINet.Spel

Private Sub m_spel_EventReceived(ByVal sender As Object,
ByVal e As SpelEventArgs) Handles m_spel.EventReceived
    Select Case e.Event
        Case RCAPINet.SpelEvens.EstopOn
            MsgBox "E-Stop detected"
            gEStop = True
            lblEStop.BackColor = Color.Red
            lblEStop.Text = "EStop ON"
        Case RCAPINet.SpelEvents.EstopOff
            gEStop = False
            lblEStop.BackColor = Color.Green
            lblEStop.Text = "EStop OFF"
    End Select
End Sub
```

在 C# 應用程序中，您可以將系統 Estop 時間捕獲到“EventReceived”中。

```
private void m_spel_EventReceived(object sender,
SpelEventArgs e)
{
  switch(e.Event)
  {
        case SpelEvents.EstopOn:
            MessageBox.Show(“E-Stop detected”);
            gEStop = true;
            lblEStop.BackColor = Color.Red;
            lblEStop.Text = "EStop ON";
        case SpelEvents.EstopOff:
            gEStop = false;
            lblEStop.BackColor = Color.Green;
            lblEStop.Text = "EStop OFF";

  }
}
```

10. EPSON RC+ 7.0視窗及對話方塊

您可以使用 Spel 類別的 ShowWindow 和 RunDialog 方法，從.NET 應用程式開啟特定的 EPSON RC+ 7.0 視窗及對話方塊。

10.1 視窗

視窗屬於非強制回應，也就是可在使用 Visual Basic GUI 的其他元素時保持開啟。您可從 Visual Basic 程式顯示及隱藏 EPSON RC+ 7.0 視窗。

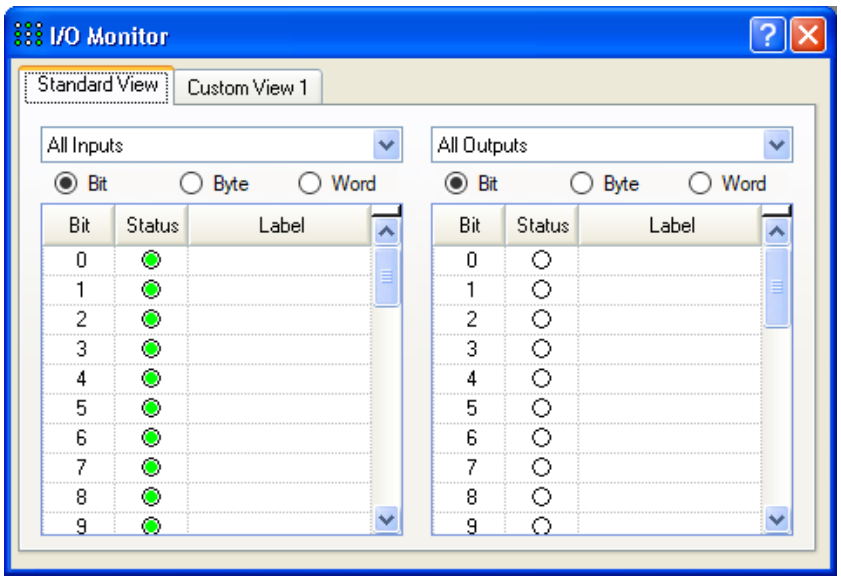
例如，若要開啟及關閉 I/O 監視器視窗：

```
m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor, Me)
m_spel.HideWindow(RCAPINet.SpelWindows.IOMonitor)
```

使用 C# GUI 的其他元素時，也可以顯示 EPSON RC+視窗。

```
m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor, this);
m_spel.HideWindow(RCAPINet.SpelWindows.IOMonitor);
```

視窗 ID	視窗
RCAPINet.SpelWindows.IOMonitor	IO Monitor
RCAPINet.SpelWindows.TaskManager	Task Manager
RCAPINet.SpelWindows.ForceMonitor	Force Monitor
RCAPINet.SpelWindows.Simulator	Simulator



I/O Monitor 視窗

10.2 對話方塊

對話方塊屬於強制回應：當對話方塊開啟時，您無法在關閉該對話方塊之前使用.NET GUI 的其他元素。

例如，若您開啟機器人管理器對話方塊：

```
m_spel.RunDialog(RCAPINet.SpelDialogs.RobotManager)
```

對話方塊一旦開啟，就必須由操作員關閉。您無法從程式中關閉對話方塊。此為安全上的考量。

下表顯示可以開啟的對話方塊。

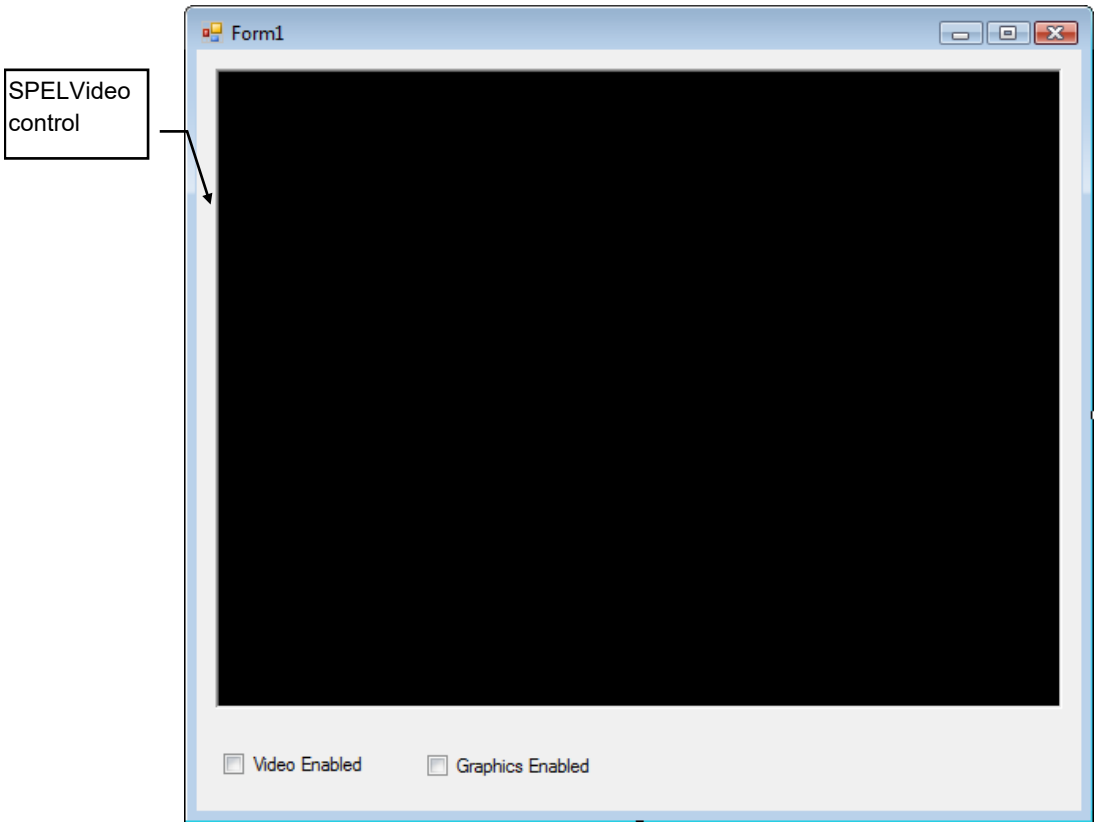
對話方塊 ID	對話方塊
RCAPINet.SpelDialogs.RobotManager	Robot Manager
RCAPINet.SpelDialogs.ControllerTools	Controller Tools
RCAPINet.SpelDialogs.VisionGuide	Vision Guide
RCAPINet.SpelDialogs.ForceGuide	Force Guide
RCAPINet.SpelDialogs.PartFeeding	Part Feeding

11. 顯示視訊

您可藉由使用 SPELVideo 控制項，在應用程式的表單上顯示視訊。執行視覺序列時，圖形也可顯示在視窗上。

請執行下列步驟來建立視訊顯示：

1. 將 SPELVideo 元件添加至專案。若要將控制項添加至 Visual Studio .NET 工具箱，請以滑鼠右鍵按一下工具箱，選取 Choose Items。選擇瀏覽標籤，瀏覽至\EpsonRC70\Exe 目錄，並選擇 RCAPINet.dll 檔案。SPELVideo 控制項圖示將會添加至工具箱。
2. 將 SPELVideo 控制項放置在您要顯示視訊的表單上。控制項尺寸最大可改變至全尺寸。
3. 將 VideoEnabled 屬性設為 True。
4. 若要顯示視覺圖形，請將 GraphicsEnabled 屬性設為 True。此外，您必須使用 Spel 類別 SpelVideoControl 屬性，將 SPELVideo 控制項附加至 Spel 類別執行個體。



放置在表單上的 SPELVideo 控制項

當 GraphicsEnabled 屬性為 True 且控制項附加至 Spel 類別執行個體時，每當在連接至 Spel 類別執行個體的控制器上執行 VRun 方法，都會顯示視覺圖形。

以下範例顯示如何從使用 Spel 類別執行個體並放置 SPELVideo 控制項的 Visual Basic 表單上啟用視訊與圖形：

```
Private Sub Form_Load(sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    m_spel = New Spel
    m_spel.Initialize()
    m_spel.Project =
"c:\EpsonRC70\projects\test\test.sprj"
    SpelVideo1.VideoEnabled = True
    SpelVideo1.GraphicsEnabled = True
    m_spel.SpelVideoControl = SPELVideo1
End Sub
```

如何在 SPELVideo 控制項所在的 C#表單上，啓用視訊與圖形

```
private void Form_Load(object sender, EventArgs e)
{
  RCAPINet.Spel m_spel = new RCAPINet.Spel();
  m_spel.Initialize();
  m_spel.Project = @"c:\EpsonRC70\projects\test\test.sprj";
  SpelVideo1.VideoEnabled = True;
  SpelVideo1.GraphicsEnabled = True;
  m_spel.SpelVideoControl = SPELVideo1;
}
```

使用多視訊顯示

自 EPSON RC+ 7.0 之 7.3.0 或更新版本開始，您可在應用程式中使用多視訊顯示。對於各顯示，您可選擇要顯示的攝影機視訊。

若要使用多個顯示，您必須為每個顯示設定 SpelVideoControl 屬性。

以下範例顯示包含兩個視訊顯示的初始化。

VB 例:

```
Private Sub Form_Load(sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    m_spel = New Spel
    m_spel.Initialize()
    m_spel.Project =
"c:\EpsonRC70\projects\test\test.sprj"
    SpelVideo1.VideoEnabled = True
    SpelVideo1.GraphicsEnabled = True
    SpelVideo1.Camera = 1
    SpelVideo2.VideoEnabled = True
    SpelVideo2.GraphicsEnabled = True
    SpelVideo2.Camera = 2
    m_spel.SpelVideoControl = SPELVideo1
    m_spel.SpelVideoControl = SPELVideo2
End Sub
```

C# 例:

```
private void Form_Load)object sender, EventArgs e)
{
  RCAPINet.Spel m_spel = new RCAPINet.Spel();
  m_spel.Initialize();
  m_spel.Project =
@"c:\\EpsonRC70\\project\\test\\test.sprj";
  SpelVideo1.VideoEnabled = true;
  SpelVideo1.GraphicsEnabled = true;
  SpelVideo1.Camera = 1;
  SpelVideo2.VideoEnabled = true;
  SpelVideo2.GraphicsEnabled = true;
  SpelViodeo2.Camera = 2;
  m_spel.SpelVdieoControl = SPELVideo1;
  m_spel.SpelVideoControl = SPELVideo2;
}
```

12. 使用AsyncMode

AsyncMode 可讓您在執行其他方法期間同時執行 Spel 方法。只有下列的 Spel 類別方法可以非同步執行：

Arc	Jump3
Arc3	Jump3CP
BGo	MCal
BMove	Move
CVMove	Pass
Go	PTran
Home	Pulse
JTran	TGo
Jump	TMove

若要非同步執行某方法，請將 AsyncMode 屬性設為 True，然後執行該方法。當 AsyncMode 屬性為 true 並執行一非同步方法時，該方法會隨即啟動，且控制項會立即返回.NET 應用程式，以作進一步處理。

若您在上一個方法正在執行時執行其他非同步方法，SPEL 將會等待第一個方法完成，再啟動下一個方法，然後返回.NET。

若要等待非同步方法完成，您可執行下列其中一項操作：

- 執行 WaitCommandComplete 方法。
- 將 AsyncMode 屬性設為 False。

如果因為錯誤(例如指定點不存在)而無法啟動非同步命令，將會立即出現例外。不過，如果在非同步執行命令期間發生錯誤，則會在執行下一個非同步命令、執行 WaitCommandComplete 或 AsyncMode 設為 False 時出現錯誤例外。如果在執行下一個命令時出現例外，您會不知道是哪個陳述式導致錯誤(上一個或目前陳述式)。如果您需要在執行其他命令前檢查非同步命令是否成功完成，則在執行下一個命令前調用 WaitCommandComplete。如果在上一個非同步命令期間發生錯誤，SpelException 例外將會與錯誤編號及訊息一起顯示。請參閱以下範例。

VB 例:

```
Try
   m_spel.AsyncMode = True
   m_spel.Go(1)
   ' 在動作期間於此處執行其他操作
   ' 如果 Go(2)執行時出現錯誤，Go(1)在執行期間出現異常，
   ' 則我們無法得知錯誤發生於 Go(1)或 Go(2)
   m_spel.Go(2)

   m_spel.Go(3)
   ' 在動作期間於此處執行其他操作
   ' 檢查 Go(3)是否成功
   m_spel.WaitCommandComplete()
   ' 如果 Go(3)發生錯誤，則會出現例外

   m_spel.Go(4)
Catch ex As SpelException
   ' 處理錯誤例外

End Try
```

C# 例:

```
  try {
    m_spel.AyncMode = true;
    m_spel.Go(1);
     // 在動作期間於此處執行其他操作
     // 如果 Go(2)執行時出現錯誤，Go(1)在執行期間出現異常，
     // 則我們無法得知錯誤發生於 Go(1)或 Go(2)
    m_spel.Go(2);

    m_spel.Go(3);
   // 在動作期間於此處執行其他操作
   // 檢查 Go(3)是否成功
    m_spel.WaitCommandComplete();
   // 如果 Go(3)發生錯誤，則會出現例外

    m_spel.Go(4);
  }
  catch(RCAPINet.SpelException ex) {
   // 處理錯誤例外
  }
```

13. SPELCom_Event

從 Spel 類別執行個體產生使用者事件。

語法

SPELCom_Event *eventNumber* [, *msgArg1, msgArg2, msgArg3,...*]

參數

eventNumber	數值介於 1000 - 32767 的整數運算式。
msgArg1, msgArg2, msgArg3...	選用。各訊息引數可以是數字、字串常值或變數名稱。

描述

此指令方便您將即時資訊傳送至控制器中正在執行的 Spel 任務之某應用程式。例如：您可透過傳送事件來更新工件計數、批號等。

SPELCom_Event 範例

在此範例中，SPEL+任務利用 RC+ API 將週期資料傳送至應用程式。

```
Function RunParts
    Integer cycNum
    String lot$
    Double cycTime

    cycNum = 0
    Do
        TmrReset(0)
        ...
        ...
        cycTime = Tmr(0)
        cycNum = cycNum + 1
        Spelcom_Event 3000, cycNum, lot$, cycTime
        Wait 0.01
    Loop
Fend
```

14. RCAPINet參考

14.1 Spel類別

描述

此類別可讓您執行命令並接收 EPSON RC+ 7.0 的事件。

檔案名稱

RCAPINet.dll(64 位元與 32 位元)

14.2 Spel類別屬性

AsyncMode 屬性，Spel 類別

描述

設定／傳回非同步執行模式。

語法

Property **AsyncMode** As Boolean

預設值

False

傳回值

非同步模式作用時，布林值為 True，未作用時則為 False。

另請參閱

使用 AsyncMode, WaitCommandComplete

AsyncMode 範例

VB 例:

```
With m_spel
   .AsyncMode = True
   .Jump("pick")
   .Delay(500)
   .On(1)
   .WaitCommandComplete()
End With
```

C# 例:

```
m_spel.AsyncMode = true;
m_spel.Jump("pick");
m_spel.Delay(500);
m_spel.On(1);
m_spel.WaitCommandComplete();
```

AvoidSingularity 屬性，Spel 類別

描述

設定／傳回奇點避開模式。

語法

Property **AvoidSingularity** As Boolean

預設值

False

傳回值

奇點避開作用時，布林值為 True，未作用時則為 False。

另請參閱

Go, Jump, Move

AvoidSingularity 範例

VB 例:

```
m_spel.AvoidSingularity = True
```

C# 例:

```
m_spel.AvoidSingularity = true;
```

CommandInCycle 屬性，Spel 類別

描述

傳回顯示某方法是否正在執行。

語法

ReadOnly Property **CommandInCycle** As Boolean

傳回值

方法正在執行時，布林值為 True，未執行時則為 False。

另請參閱

AsyncMode

CommandInCycle 範例

VB 例:

```
If m_spel.CommandInCycle Then
    MsgBox "A SPEL command is executing, operation
aborted"
End If
```

C# 例:

```
if (m_spel.CommandInCycle)
    MessageBox.Show("SPEL command is executing, operation
aborted");
```

CommandTask 屬性，Spel 類別

描述

為執行機器人命令指定在控制器使用的保留 API 任務。

語法

Property **CommandTask** As Integer

預設值

預設值為 0(不使用保留的 API 任務)。

備註

當您想在控制器中以其他執行緒執行 Spel 機器人命令時，請使用 CommandTask。一般而言，CommandTask 用於多機器人系統。使用 CommandTask 前，您必須先從 EPSON RC+功能表-[Setup]-[System Configuration]-[Controller]-[Preferences]保留要在控制器中使用的 API 任務。您最多可在控制器中保留 16 個 API 任務。

另請參閱

ServerInstance

CommandTask 範例

VB 例:

```
' 在 Robot1 執行緒中
m_spel.CommandTask = 1
m_spel.Robot = 1

' 在 Robot2 執行緒中
m_spel.CommandTask = 2
m_spel.Robot = 2
```

C# 例:

```
// 在 Robot1 執行緒中
m_spel.CommandTask = 1;
m_spel.Robot = 1;

// 在 Robot2 執行緒中
m_spel.CommandTask = 2;
m_spel.Robot = 2;
```

DisableMsgDispatch 屬性，Spel 類別

描述

設定／傳回顯示 Windows 訊息是否應在 Spel 方法執行期間處理。

語法

DisableMsgDispatch

類型

Boolean

預設值

False

備註

此屬性通常不應使用。主要是用於當執行 Spel 方法時不想用鍵盤或滑鼠處理的特殊應用。

ErrorCode 屬性，Spel 類別

描述

傳回目前控制器錯誤代碼。

語法

ReadOnly Property **ErrorCode** As Integer

傳回值

包含錯誤代碼的整數值。

另請參閱

ErrorOn

ErrorCode 範例

VB 例:

```
If m_spel.ErrorOn Then
    lblErrorCode.Text = m_spel.ErrorCode.ToString()
Else
    lblErrorCode.Text = ""
End If
```

C# 例:

```
if (m_spel.ErrorOn)
    lblErrorCode.Text = m_spel.ErrorCode.ToString();
else
    lblErrorCode.Text = "";
```

ErrorOn 屬性，Spel 類別

描述

若控制器發生嚴重錯誤，會傳回 True。

語法

ReadOnly Property **ErrorOn** As Boolean

傳回值

若控制器處於錯誤狀態，會傳回 True，否則會傳回 False。

備註

當控制器處於錯誤狀態時，ErrorOn 屬性會傳回 True，且您可使用 ErrorCode 擷取錯誤代碼。

另請參閱

ErrorCode

ErrorOn 範例

VB 例:

```
If m_spel.ErrorOn Then
    m_spel.Reset
End If
```

C# 例:

```
if (m_spel.ErrorOn)
    m_spel.Reset();
```

EStopOn 屬性，Spel 類別

描述

傳回控制器緊急停止的狀態。

語法

ReadOnly Property **EStopOn** As Boolean

傳回值

若緊急停止生效，會傳回 True，否則會傳回 False。

EStopOn 範例

VB 例:

```
If m_spel.EStopOn Then
    lblEStop.Text = "Emergency stop is active"
Else
    lblEStop.Text = ""
EndIf
```

C# 例:

```
if (m_spel.EStopOn)
    lblEStop.Text = "Emergency stop is active";
else
    lblEStop.Text = "";
```

Force_Sensor 屬性，Spel 類別

描述

設定並傳回目前力覺感測器編號。

語法

Property **Force_Sensor** As Integer

預設值

1

傳回值

目前力覺感測器編號的整數值。

備註

在使用任何 force 方法之前，您必須使用此屬性設定目前的力覺感測器。

另請參閱

Force_Calibrate, Force_GetForces, Force_SetTrigger

Force_Sensor 範例

VB 例:

```
' 讀取感測器 2 的 Z 軸力
m_spel.Force_Sensor = 2
f = m_spel.Force_GetForce(3)
```

C# 例:

```
// 讀取感測器 2 的 Z 軸力
m_spel.Force_Sensor = 2;
f = m_spel.Force_GetForce(3);
```

MotorsOn 屬性，Spel 類別

描述

設定並傳回目前機器人馬達開啟或關閉的狀態。

語法

Property **MotorsOn** As Boolean

預設值

False

傳回值

馬達開啟時，布林值為 True，未開啟時則為 False。

另請參閱

PowerHigh, Reset, Robot

MotorsOn 範例

VB 例:

```
If Not m_spel.MotorsOn Then
   m_spel.MotorsOn = True
End If
```

C# 例:

```
if (!m_spel.MotorsOn)
   m_spel.MotorsOn = true;
```

NoProjectSync 屬性，Spel 類別

描述

設定／傳回是否要將 PC 中的目前專案與控制器專案進行同步。

語法

NoProjectSync

類型

Boolean

預設值

False

備註

當 NoProjectSync 設為 False(預設)時，Spel 類別會確保 PC 中的專案與控制器中的專案進行同步。

當 NoProjectSync 設為 True 時，Spel 類別不會檢查 PC 上的任何專案，也不會將 PC 專案與控制器進行同步。這可讓您執行控制器中的程式，而不會執行 PC 中的任何專案。

此屬性不是持續有效。如果您想將其設為 True，必須在建立 Spel 類別執行個體後進行設定。

另請參閱

Start

NoProjectSync 範例

VB 例:

```
m_spel.Initialize()
m_spel.NoProjectSync = True
```

C# 例:

```
m_spel.Initialize();
m_spel.NoProjectSync = true;
```

OperationMode 屬性，Spel 類別

描述

讀取或設定 EPSON RC+ 7.0 操作模式。

語法

Property **OperationMode** As SpelOperationMode

傳回值

SpelOperationMode 值

備註

當 **OperationMode** 設為 Program 時，Spel 類別目前執行個體的 EPSON RC+ 7.0 GUI 會開啟，而控制器操作模式會設為 Program。若使用者關閉 GUI，**OperationMode** 會設為 Auto。若從 Visual Basic 將 **OperationMode** 設為 Auto，則 GUI 會同時關閉。

OperationMode 範例

VB 例:

```
Sub btnSpelProgramMode_Click _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnHideIOMonitor.Click

  Try
    m_spel.OperationMode = _
                  RCAPINet.SpelOperationMode.Program
    ' 如果您想等待使用者關閉 RC+ GUI，
    ' 您可在此等待 OperationMode 變更為 Auto
    Do
      Application.DoEvents()
      System.Threading.Thread.Sleep(10)
    Loop Until m_spel.OperationMode = _
                   RCAPINet.SpelOperationMode.Auto
  Catch ex As RCAPINet.SpelException
    MsgBox(ex.Message)
  End Try
End If
```

C# 例:

```
void btnSpelProgramMode_Click(object sender, EventArgs e)
{
  try {
    m_spel.OperationMode =RCAPINet.SpelOperationMode.Auto;
    //如果想等待 RC+ GUI 關閉，可在此待機
    Do {
       Application.DoEvents();
       Systme.Threading.Thread.Sleep(10);
    } while(!m_spel.OperationMode =
RCAPINet.OperationMode.Auto);
  }
  Catch (SpelException ex){
    MessageBox.Show(ex.Message);
  }
}
```

ParentWindowHandle 屬性，Spel 類別

描述

設定／傳回對話方塊及視窗所用父視窗的控制代碼。

語法

Property **ParentWindowHandle** As Integer

傳回值

包含視窗控制代碼的整數值。

備註

從不包含.NET 表單的應用程式(例如 LabVIEW)，使用 ParentWindowHandle 來指定父視窗。

另請參閱

ShowWindow

ParentWindowHandle 範例

VB 例:

```
m_spel.ParentWindowHandle = Me.Handle
m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor)
```

C# 例:

```
m_spel.ParentWindowHandle  = (int)this.Handle;
m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor);
```

PauseOn 屬性，Spel 類別

描述

傳回控制器暫停狀態的狀態。

語法

ReadOnly Property **PauseOn** As Boolean

傳回值

若控制器處於暫停狀態，會傳回 True，否則會傳回 False。

另請參閱

Continue Pause

PauseOn 範例

VB 例:

```
If m_spel.PauseOn Then
    btnPause.Enabled = False
    btnContinue.Enabled = True
End If
```

C# 例:

```
if (m_spel.PauseOn){
    btnPause.Enabled = false;
    btnContinue.Enabled = true;
}
```

PowerHigh 屬性，Spel 類別

描述

設定並傳回目前機器人的運行功率狀態。

語法

Property **PowerHigh** As Boolean

預設值

False

傳回值

若目前機器人運行功率為 high，會傳回 True，否則會傳回 False。

另請參閱

MotorsOn

PowerHigh 範例

VB 例:

```
If Not m_spel.PowerHigh Then
    m_spel.PowerHigh = True
End If
```

C# 例:

```
if (!m_spel.PowerHigh)
    m_spel.PowerHigh = true;
```

Project 屬性，Spel 類別

描述

設定／傳回目前專案。

語法

Property **Project** As String

預設值

空字串。

傳回值

包含專案路徑及檔案的字串。

備註

設定 **Project** 時，您必須提供 EPSON RC+ 7.0 專案製作檔案的完整路徑與名稱。該製作檔案為包含.SPRJ 副檔名的專案名稱。

Project 範例

VB 例:

```
m_spel.Project = "c:\EpsonRC70\projects\myapp\myapp.sprj"
```

C# 例:

```
m_spel.Project = @"c:\EpsonRC70\projects\myapp\myapp.sprj";
```

ProjectBuildComplete 屬性，Spel 類別

描述

傳回目前專案建置的狀態。

語法

ReadOnly Property **ProjectBuildComplete** As Boolean

傳回值

若專案建置完成，會傳回 True，否則會傳回 False。

另請參閱

BuildProject

ProjectBuildComplete 範例

VB 例:

```
If m_spel.ProjectBuildComplete Then
    lblBuild.Text = "Project build is Complete"
Else
    lblBuild.Text = "Project build is not Complete"
End If
```

C# 例:

```
if (m_spel.ProjectBuildComplete)
    lblBuild.Text = “Project build is Complete”;
else
    lblBuild.Text = “Project build is not Complete”;
```

ProjectOverwriteWarningEnabled 屬性, Spel 類別

描述

設定／傳回是否要啟用專案覆蓋時的錯誤訊息顯示。

語法

Property **ProjectOverwriteWarningEnabled** As Boolean

預設值

True

傳回值

若啟用覆蓋時的錯誤訊息，會傳回 True，否則會傳回 False。

另請參閱

BuildProject

備註

預設情況下，若當前專案與控制器的專案不同，生成程式或向控制器發送訊息時，將顯示專案將被覆蓋的錯誤訊息。如果您不需要顯示訊息，請將 ProjectOverwriteWarningEnabled 設定為 False。當您的應用時需要切換控制器使用的專案時，可以選擇這個功能。

ProjectOverwriteWarningEnabled 範例

VB 例:

```
' Disable the project overwrite warning
m_spel.ProjectOverwriteWarningEnabled = False
m_spel.Project =
"c:\EpsonRC70\Projects\Project1\Project1.sprj"
```

C# 例:

```
// Disable the project overwrite warning
m_spel.ProjectOverwriteWarningEnabled = false;
m_spel.Project =
@"c:\EpsonRC70\Projects\Project1\Project1.sprj";
```

ResetAbortEnabled 屬性，Spel 類別

描述

設定／傳回是否要啟用 ResetAbort 方法。

語法

Property **ResetAbortEnabled** As Boolean

預設值

True

傳回值

若 ResetAbort 已啟用，會傳回 True，否則會傳回 False。

另請參閱

ResetAbort

ResetAbortEnabled 範例

VB 例:

```
' 啟用 ResetAbort
m_spel.ResetAbortEnabled = True
```

C# 例:

```
// 啟用 ResetAbort
m_spel.ResetAbortEnabled = true;
```

Robot 屬性，Spel 類別

描述

設定／傳回目前機器人編號。

語法

Property **Robot** As Integer

預設值

若有一或多個機器人存在，第一個 Spel 執行個體的預設值為 1，否則為 0。對於其他 Spel 執行個體，預設值為 0。

傳回值

包含目前機器人編號的整數值。

備註

在使用多個機器人的系統上，使用 **Robot** 屬性設定後續機器人相關命令，例如動作命令。

另請參閱

RobotModel, RobotType

Robot 範例

VB 例:

```
m_spel.Robot = 2
If Not m_spel.MotorsOn Then
  m_spel.MotorsOn = True
End If
```

C# 例:

```
m_spel.Robot = 2;
if (!m_spel.MotorsOn)
  m_spel.MotorsOn = true;
```

RobotModel 屬性，Spel 類別

描述

傳回目前機器人的型號名稱。

語法

ReadOnly Property **RobotModel** As String

傳回值

包含目前機器人型號名稱的字串。

另請參閱

Robot, RobotType

RobotModel 範例

VB 例:

```
lblRobotModel.Text = m_spel.RobotModel
```

C# 例:

```
lblRobotModel.Text = m_spel.RobotModel;
```

RobotType 屬性，Spel 類別

描述

傳回目前機器人的類型。

語法

ReadOnly Property **RobotType** As SpelRobotType

傳回值

SpelRobotType 值

另請參閱

Robot, RobotModel

RobotType 範例

VB 例:

```
Select Case m_spel.RobotType
  Case RCAPINet.SpelRobotType.Scara
    lblRobotType.Text = "Scara"
  Case RCAPINet.SpelRobotType.Cartesian
    lblRobotType.Text = "Cartesian"
End Select
```

C# 例:

```
switch (m_spel.RobotType)
{
    case SpelRobotType.Scara:
         lblRobotType.Text = “Scara”;
    break;
    case SpelRobotType.Cartesian:
         lblRobotType.Text = “Cartesian”;
    break;
    default:
         break;
}
```

SafetyOn 屬性，Spel 類別

描述

傳回控制器安全防護輸入的狀態。

語法

ReadOnly Property **SafetyOn** As Boolean

傳回值

若安全防護打開，會傳回 True，否則會傳回 False。

備註

請在應用程式啟動時，使用 SafetyOn 屬性取得安全防護狀態，然後使用 SafeguardOpen 和 SafeguardClose 事件來更新狀態。

SafetyOn 範例

VB 例:

```
If m_spel.SafetyOn Then
    lblSafeguard.Text = "Safe guard is active"
Else
    lblSafeguard.Text = ""
End If
```

C# 例:

```
if (m_spel.SafetyOn)
    lblSafeguard.Text = "Safe guard is active";
else
    lblSafeguard.Text = "";
```

ServerInstance 屬性，Spel 類別

描述

指定 EPSON RC+伺服器要使用的執行個體。

語法

Property **ServerInstance** As Integer

預設值

預設值為下一個可用的伺服器執行個體。

備註

API 與 RC+ 伺服器進程通信。 ServerInstance 指定要使用的伺服器。 每個伺服器實例對應一個控制器和一個專案。預設情況下，創建新 Spel 類實例時，ServerInstance 將自動設置為"1"。

當您想要對同一控制器 (如應用程式的多線程)使用多個 Spel 類實例，請為使用同一控制器的每個 Spel 類實例設置 ServerInstance 屬性。

ServerInstance 必須介於 1 和 10 之間，並且必須在初始化或執行其他方法之前進行設置。

另請參閱

CommandTask, Initialize

ServerInstance 範例

VB 例:

```
' Controller 1
spel1 = New Spel
spel1.ServerInstance = 1
spel1.Initialize()
spel1.Connect(1)

' Controller 2
spel2 = New Spel
spel2.ServerInstance = 2
spel2.Initialize()
spel2.Connect(2)
```

C# 例:

```
// Controller 1
RCAPINet.Spel spel1 = new RCAPINet.Spel();
spel1.ServerInstance = 1;
spel1.Initialize();
spel1.Connect(1);

// Controller 2
RCAPINet.Spel spel2 = new RCAPINet.Spel();
spel2.ServerInstance = 2;
spel2.Initialize();
spel2.Connect(2);
```

SPELVideoControl 屬性，Spel 類別

描述

用來將 SPELVideo 控制項連接至 Spel 類別執行個體，以顯示視訊與圖形。

語法

Property **SpelVideoControl** As SpelVideo

另請參閱

Graphics Enabled, VideoEnabled, Camera

SpelVideoControl 範例

VB 例:

```
m_spel.SpelVideoControl = SpelVideo1
```

C# 例:

```
m_spel.SpelVideoControl = SpelVideo1;
```

Version 屬性，Spel 類別

描述

傳回目前 EPSON RC+ 7.0 軟體版本。

語法

ReadOnly Property **Version** As String

傳回值

包含目前 EPSON RC+ 7.0 軟體版本的字串。

Version 範例

VB 例:

```
' 取得軟體版本
curVer = m_spel.Version
```

C# 例:

```
// 取得軟體版本
curVer = m_spel.Version;
```

WarningCode 屬性，Spel 類別

描述

傳回控制器警告代碼。

語法

ReadOnly Property **WarningCode** As Integer

傳回值

包含目前控制器警告代碼的整數值。

另請參閱

WarningOn

WarningCode 範例

VB 例:

```
If m_spel.WarningOn Then
    lblWarningCode.Text = m_spel.WarningCode.ToString()
Else
    lblWarningCode.Text = ""
End If
```

C# 例:

```
if (m_spel.WarningOn)
    lblWarningCode.Text = m_spel.WarningCode.ToString();
else
    lblWarningCode.Text = "";
```

WarningOn 屬性，Spel 類別

描述

傳回控制器警告狀態的狀態。

語法

ReadOnly Property **WarningOn** As Boolean

傳回值

若控制器處於警告狀態，會傳回 True，否則會傳回 False。

另請參閱

WarningCode

WarningOn 範例

VB 例:

```
If m_spel.WarningOn Then
    lblWarningStatus.Text = "ON"
Else
    lblWarningStatus.Text = "OFF"
End If
```

C# 例:

```
if (m_spel.WarningOn)
    lblWarningStatus.Text = "ON";
else
    lblWarningStatus.Text = "OFF";
```

14.3 Spel類別方法

Accel 方法，Spel 類別

描述

設定 PTP 動作命令 Go、Jump 及 Pulse 的加速與減速。

語法

Sub **Accel** (*PointToPointAccel* As Integer, *PointToPointDecel* As Integer, _
[*JumpDepartAccel* As Integer], [*JumpDepartDecel* As Integer], _
[*JumpApproAccel* As Integer], [*JumpApproDecel* As Integer])

參數

PointToPointAccel	介於 1-100 的整數運算式，代表最大加速速率的百分比。
PointToPointDecel	介於 1-100 的整數運算式，代表最大減速速率的百分比。
JumpDepartAccel	介於 1-100 的整數運算式，代表 Jump 命令 Z 軸向上動作的最大加速速率百分比。
JumpDepartDecel	介於 1-100 的整數運算式，代表 Jump 命令 Z 軸向上動作的最大減速速率百分比。
JumpApproAccel	介於 1-100 的整數運算式，代表 Jump 命令 Z 軸向下動作的最大加速速率百分比。
JumpApproDecel	介於 1-100 的整數運算式，代表 Jump 命令 Z 軸向下動作的最大減速速率百分比。

另請參閱

Accels, Speed

Accel 範例

VB 例:

```
m_spel.Accel(50, 50)
m_spel.Go ("pick")
```

C# 例:

```
m_spel.Accel(50, 50);
m_spel.Go ("pick");
```

AccelR 方法，Spel 類別

描述

設定工具旋轉動作的加速與減速。

語法

Sub **AccelR** (*Accel* As Single, [*Decel* As Single])

參數

Accel	從 0.1 至 5000 deg/sec^2 的單一運算式，用以定義在動作命令中使用 ROT 時的工具旋轉加速。若省略 Decel，此值會同時用於加速與減速速率。
Decel	選用。從 0.1 至 5000 deg/sec^2 的單一運算式，用以定義在動作命令中使用 ROT 時的工具旋轉減速。

另請參閱

Arc, Arc3, BMove, Jump3CP, Power, SpeedR, TMove

AccelR 範例

VB 例:

```
Sub MoveToPlace()
    m_spel.AccelR(100)
    m_spel.Move("place ROT")
End Sub
```

C# 例:

```
void MoveToPlace()
{
    m_spel.AccelR(100);
    m_spel.Move("place ROT");
}
```

AccelS 方法，Spel 類別

描述

設定直綫動作和 CP 動作(Jump3CP, Move, TMove)的加速與減速。

語法

Sub **AccelS** (*Accel* As Single*, Decel* As Single,
[*JumpDepartAccel* As Single], [*JumpDepartDecel* As Single], _
[*JumpApproAccel* As Single], [*JumpApproDecel* As Single])

參數

Accel	介於 1-5000 mm/sec^{2}的單一運算式，用以定義直線及連續路徑動作的加速與減速值。若省略 Decel，此值會同時用於加速與減速速率。
Decel	介於 1-5000 mm/sec^{2}的單一運算式，用以定義直線及連續路徑動作的減速值。單一參數同時用來表示加速與減速速率。
JumpDepartAccel	介於 1-5000 的單一運算式，代表 Jump3CP 命令 Z 軸向上動作的最大加速速率百分比。
JumpDepartDecel	介於 1-5000 的單一運算式，代表 Jump3CP 命令 Z 軸向上動作的最大減速速率百分比。
JumpApproAccel	介於 1-5000 的單一運算式，代表 Jump3CP 命令 Z 軸向下動作的最大加速速率百分比。
JumpApproDecel	介於 1-5000 的單一運算式，代表 Jump3CP 命令 Z 軸向下動作的最大減速速率百分比。

另請參閱

Accel, SpeedS, Jump3CP, Move, TMove

AccelS 範例

VB 例:

```
Sub MoveToPlace()
    m_spel.AccelS(500)
    m_spel.Move(pick)
    m_spel.AccelS(500, 300)
    m_spel.Move(place)
End Sub
```

C# 例:

```
void MoveToPlace()
{
    m_spel.AccelS(500);
    m_spel.Move(pick);
    m_spel.AccelS(500, 300);
    m_spel.Move(place);
}
```

Agl 方法，Spel 類別

描述

傳回所選旋轉軸的關節角度，或所選線性軸的位置。

語法

Function **Agl** (*JointNumber* As Integer) As Single

參數

JointNumber　介於 1-9 的整數運算式，代表關節編號。

另請參閱

Pls, CX - CT

Agl 範例

VB 例:

```
Dim j1Angle As Single
j1Angle = m_spel.Agl(1)
```

C# 例:

```
float j1Angle;
j1Angle = m_spel.Agl(1);
```

AIO_In 方法, Spel 類別

描述

從選配件類比 IO 的輸入通道讀取類比值。

語法

Function **AIO_In** (*Channel* As Integer) As Single

參數

Channel　　指定類比 IO 的通道編號。

傳回值

傳回由通道指定的類比 IO 通道的類比輸入值(以實數爲單位)。傳回值得範圍取決於類比 IO 板的輸入範圍設定。

另請參閱

AIO_InW, AIO_Out, AIO_OutW

AIO_In 範例

VB 例:

```
Dim val As Single
val = m_spel.AIO_In(2)
```

C# 例:

```
float val;
val = m_spel.AIO_In(2);
```

AIO_InW 方法, Spel 類別

描述

從選配件類比 IO 的輸入通道讀取類比值。

語法

Function **AIO_InW** (*Channel* As Integer) As Integer

參數

Channel　　指定類比 IO 的通道編號。

傳回值

傳回指定的類比 IO 通道的輸入狀態 (0~65535 的整數值)。

根據類比 IO 板的輸入範圍設定，每個輸入通道的輸入電壓(電流)和返回值之間的對應關係如下所示。

輸入數據		設定輸入範圍				
16 進數	10 進數	±10.24(V)	±5.12(V)	0-5.12(V)	0-10.24(V)	0-24(mA)
0xFFFF	65535	10.23969	5.11984	5.12000	10.24000	24.00000
0x8001	32769	0.00031	0.00016	2.56008	5.12016	12.00037
0x8000	32768	0.00000	0.00000	2.56000	5.12000	12.00000
0x0000	0	-10.24000	-5.12000	0.00000	0.00000	0.00000

另請參閱

AIO_In, AIO_Out, AIO_OutW

AIO_InW 範例

VB 例:

```
Dim val As Integer
val = m_spel.AIO_InW(2)
```

C# 例:

```
int val;
val = m_spel.AIO_InW(2);
```

AIO_Out 方法, Spel 類別

描述

從選配件類比 IO 的輸出通道，讀取或設定類比值。

語法

Function **AIO_Out** (*Channel* As Integer) As Single
Sub **AIO_Out** (*Channel* As Integer, *Value* As Single)

參數

Channel　指定類比 IO 的通道編號。

Value　用陳述式或數位，表示要輸出的電壓[V]或電流值[mA]的實數。

傳回值

傳回指定類比 IO 通道的電壓或電流輸出狀態(以實數爲單位)。電壓輸出的單位為[V]，電流輸出的單位為[mA]。

Function **AIO_Out** (*Channel* As Integer) As Single 時：

如果在指定通道輸出機器人的速度資訊時，也可以用本方法取得傳回值。

備註

輸出表示指定電壓[V]或電流[mA]的實數，到通道號碼指定的類比輸出埠。類比輸出埠的電壓輸出範圍和電壓、電流輸出選擇，由機板上的開關設定。如果指定的值超出類比 IO 板的輸出範圍，則輸出不超過範圍的邊界值(最大值或最小值)。

Sub **AIO_Out** (*Channel* As Integer, *Value* As Single) 時：

如果在指定通道輸出機器人的速度資訊時，本方法會發生輸出設定錯誤。請在停止輸出速度信息后，執行本方法。

另請參閱

AIO_In, AOI_InW, AIO_OutW

AIO_Out 範例

VB 例:

```
Dim val As Single
val = m_spel.AIO_Out(1)
```

C# 例:

```
float val;
val = m_spel.AIO_Out(1);
```

AIO_OutW 方法, Spel 類別

描述

從選配件類比 IO 的輸出通道，讀取或設定類比值。

語法

Function **AIO_OutW** (*Channel* As Integer) As Integer
Sub **AIO_OutW** (*Channel* As Integer, *OutputData* As Integer)

參數

Channel 指定類比 IO 的通道編號。

OutputData 用陳述式或數位，指定輸出數據(從 0 到 65535 的整數)。

傳回值

傳回指定類比 IO 通道的輸出狀態，用 0~65535 的整數表示。

根據類比 IO 板的輸出範圍設定，每個輸出通道的輸出電壓(電流)和返回值之間的對應關係如下所示。

輸出數據		設定輸出範圍					
16 進數	10 進數	±10(V)	±5(V)	0-5(V)	0-10(V)	4-20(mA)	0-20(mA)
0xFFFF	65535	9.99970	4.99985	5.00000	10.00000	20.00000	20.00000
0x8001	32769	0.00031	0.00015	2.50008	5.00015	12.00024	10.00031
0x8000	32768	0.00000	0.00000	2.50000	5.00000	12.00000	10.00000
0x0000	0	-10.00000	-5.00000	0.00000	0.00000	4.00000	0.00000

如果在指定通道輸出機器人的速度資訊時，也可以執行此方法。

另請參閱

AIO_In, AOI_InW, AIO_Out

AIO_OutW 範例

VB 例:

```
Dim val As Integer
val = m_spel.AIO_OutW(1)
```

C# 例:

```
int val;
val = m_spel.AIO_OutW(1);
```

Arc 方法，Spel 類別

描述

Arc 利用 XY 平面中的圓形插補將手臂移至指定點。

語法

Sub **Arc** (*MidPoint* As Integer, *EndPoint* As Integer)
Sub **Arc** (*MidPoint* As SpelPoint, *EndPoint* As SpelPoint)
Sub **Arc** (*MidPoint* As String, *EndPoint* As String)

參數

每個語法都有兩個參數，分別指定圓弧的中點與結束點。

MidPoint	使用整數、SpelPoint 或字串運算式來指定中點。
EndPoint	使用整數、SpelPoint 或字串運算式來指定結束點。使用字串運算式時，您可加入 ROT、CP、SYNC，Till 的搜尋運算式，以及平行處理陳述式。

另請參閱

AccelR, AccelS, SpeedR, SpeedS
Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move
BGo, BMove, TGo, TMove
CP, Till

Arc 範例

VB 例:

```
' 使用 SpelPoint 指定的點
Dim midPoint, endPoint As SpelPoint
midPoint = m_spel.GetPoint("P1")
endPoint = m_spel.GetPoint("P2")
m_spel.Arc(midPoint, endPoint)

' 使用運算式指定的點
m_spel.Arc("P1", "P2")
m_spel.Arc("P1", "P2 CP")

' 使用平行處理
m_spel.Arc("P1", "P2 !D50; On 1; D90; Off 1!")
```

C# 例:

```
// 使用 SpelPoint 指定的點
SpelPoint midPoint, endPoint;
midPoint = m_spel.GetPoint("P1");
endPoint = m_spel.GetPoint("P2");
m_spel.Arc(midPoint, endPoint);

// 使用運算式指定的點
m_spel.Arc("P1", "P2");
m_spel.Arc("P1", "P2 CP");

// 使用平行處理
m_spel.Arc("P1", "P2 !D50; On 1; D90; Off 1!")
```

Arc3 方法，Spel 類別

描述

Arc3 在 3D 中利用圓形插補將手臂移至指定點。

語法

Sub **Arc3** (*MidPoint* As Integer, *EndPoint* As Integer)
Sub **Arc3** (*MidPoint* As SpelPoint, *EndPoint* As SpelPoint)
Sub **Arc3**(*MidPoint* As String, *EndPoint* As String)

參數

每個語法都有兩個參數，分別指定圓弧的中點與結束點。

MidPoint	使用整數、SpelPoint 或字串運算式來指定中點。
EndPoint	使用整數、SpelPoint 或字串運算式來指定結束點。使用字串運算式時，您可加入 ROT、ECP、CP、SYNC，Till 的搜尋運算式，以及平行處理陳述式。

另請參閱

AccelR, AccelS, SpeedR, SpeedS
Arc, CVMove, Go, Jump, Jump3, Jump3CP, Move
BGo, BMove, TGo, TMove
CP, Till

Arc3 範例

VB 例:

```
' 使用 SpelPoint 指定的點
Dim midPoint, endPoint As SpelPoint
midPoint = m_spel.GetPoint("P1")
endPoint = m_spel.GetPoint("P2")
m_spel.Arc3(midPoint, endPoint)

' 使用運算式指定的點
m_spel.Arc3("P1", "P2")
m_spel.Arc3("P1", "P2 CP")

' 使用平行處理
m_spel.Arc3("P1", "P2 !D50; On 1; D90; Off 1!")
```

C# 例:

```
// 使用 SpelPoint 指定的點
SpelPoint midPoint, endPoint;
midPoint = m_spel.GetPoint("P1");
endPoint = m_spel.GetPoint("P2");
m_spel.Arc3(midPoint, endPoint);

// 使用運算式指定的點
m_spel.Arc3("P1", "P2");
m_spel.Arc3("P1", "P2 CP");

// 使用平行處理
m_spel.Arc3("P1", "P2 !D50; On 1; D90; Off 1!");
```

Arch 方法，Spel 類別

描述

定義要搭配 JUMP 指令使用的 ARCH 參數(開始水平動作之前要移動的 Z 高度)。

語法

Sub **Arch** (*ArchNumber* As Integer, *DepartDist* As Integer, *ApproDist* As Integer)

參數

ArchNumber	要定義的拱形編號。有效的拱形編號為(0-6)，共有 7 個項目編入拱形表。
DepartDist	在開始水平移動之前，Jump 指令開始時移動的起始距離(公釐)。
ApproDist	Jump 指令目標位置以上的結束距離(公釐)。

另請參閱

Jump, Jump3, Jump3CP

Arch 範例

VB 例:

```
Sub SetArchs()
    With m_spel
        .Arch(1, 30, 30)
        .Arch(2, 60, 60)
        .Jump("P1 C1")
        .Jump("P2 C2")
    End With
End Sub
```

C# 例:

```
void SetArchs()
{
    m_spel.Arch(1, 30, 30);
    m_spel.Arch(2, 60, 60);
    m_spel.Jump("P1 C1");
    m_spel.Jump("P2 C2");
}
```

Arm 方法，Spel 類別

描述

選擇目前機器人手臂。

語法

Sub **Arm** (*ArmNumber* As Integer)

參數

ArmNumber 介於 0-15 的整數運算式。使用者最多可選擇 16 種不同的手臂。手臂 0 是標準(預設)機器人手臂。手臂 1-15 是透過 ArmSet 指令定義的輔助手臂。

另請參閱

ArmSet, GetArm, Tool

Arm 範例

VB 例:

```
m_spel.Arm(1)
```

C# 例:

```
m_spel.Arm(1);
```

ArmClr 方法，Spel 類別

描述

清除(取消定義)目前機器人的手臂。

語法

Sub **ArmClr** (*ArmNumber* As Integer)

參數

ArmNumber 介於 1-15 的整數運算式。手臂 0 是標準(預設)機器人手臂，且無法清除。手臂 1-15 是透過 ArmSet 指令定義的輔助手臂。

另請參閱

ArmSet, GetArm, Tool

ArmClr 範例

VB 例:

```
m_spel.ArmClr(1)
```

C# 例:

```
m_spel.ArmClr(1);
```

ArmDef 方法，Spel 類別

描述

傳回機器人手臂是否定義。

語法

Function **ArmDef** (*ArmNumber* As Integer) As Boolean

參數

ArmNumber 介於 1-15 的整數運算式。手臂 0 是標準(預設)機器人手臂，且為永遠定義。手臂 1-15 是使用 ArmSet 方法定義的輔助手臂。

傳回值

若指定手臂已定義，會傳回 True，否則會傳回 False。

另請參閱

ArmSet, GetArm, Tool

ArmDef 範例

VB 例:

```
x = m_spel.ArmDef(1)
```

C# 例:

```
x = m_spel.ArmDef(1);
```

ArmSet 方法，Spel 類別

描述

定義輔助的機器人手臂。

語法

Sub **ArmSet** (*ArmNumber* As Integer, *Param1* As Single, *Param2* As Single, *Param3* As Single, *Param4* As Single, *Param5* As Single)

參數

ArmNumber	整數：有效範圍介於 1-15。
Param1	(SCARA 機器人適用)肘部中心線到新方向軸中心線的水平距離。(亦即新輔助手臂方向軸中心線所在的位置。) (Cartesian 機器人適用)原始 X 位置的 X 軸方向位置偏移(公釐)。
Param2	(SCARA 機器人適用)從標準肘部中心線與標準方向軸中心線所形成的直線，到新輔助手臂肘部中心線與新方向軸中心線所形成的直線，兩者之間的偏移(度)。(這兩條直線應該在肘部中心線相交，形成的角度即為 *Param2*。) (Cartesian 機器人適用)原始 Y 位置的 Y 軸方向位置偏移(公釐)。
Param3	(SCARA & Cartesian 機器人適用)新方向軸中心與舊方向軸中心之間的 Z 高度偏移差距。(此為距離。)
Param4	(SCARA 機器人適用)肩部中心線到新輔助軸肘部方向之肘部中心線的距離。 (Cartesian 機器人適用)此為虛擬參數(指定 0)
Param5	(SCARA & Cartesian 機器人適用)新方向軸與舊方向軸的角度偏移(度)。

另請參閱

Arm, Tool, TLSet

ArmSet 範例

VB 例:

```
Sub SetArms()
    With m_spel
        .ArmSet(1, 1.5, 0, 0, 0, 0)
        .ArmSet(2, 3.2, 0, 0, 0, 0)
    End With
End Sub
```

C# 例:

```
void SetArms()
{
    m_spel.ArmSet(1, 1.5, 0, 0, 0, 0);
    m_spel.ArmSet(2, 3.2, 0, 0, 0, 0);
}
```

Atan 方法，Spel 類別

描述

傳回數值運算式的圓弧正切函數。

語法

Function **Atan** (*number* As Double) As Double

參數

number　代表角度值之正切函數的數值運算式。

傳回值

指定值的圓弧正切函數

另請參閱

Atan2

Atan 範例

VB 例:

```
Dim angle As Double

angle = m_spel.Atan(.7)
```

C# 例:

```
double angle;

angle = m_spel.Atan(.7);
```

Atan2 方法，Spel 類別

描述

傳回虛線連接點(0,0)和(X, Y)的角度(以徑度表示)。

語法

Function **Atan2** (*Dx* As Double, *Dy* as Double) As Double

參數

Dx　　代表 X 座標的數值運算式。

Dy　　代表 Y 座標的數值運算式。

傳回值

包含角度的雙精度值。

另請參閱

Atan

Atan2 範例

VB 例:

```
Dim angle As Double

angle = m_spel.Atan2(-25, 50)
```

C# 例:

```
double angle;

angle = m_spel.Atan2(-25, 50);
```

ATCLR 方法，Spel 類別

描述

清除并初始化關節的有效轉矩。

語法

Sub **ATCLR** ()

另請參閱

ATRQ, PTCLR, PTRQ

ATCLR 範例

VB 例:

```
m_spel.ATCLR()
```

C# 例:

```
m_spel.ATCLR();
```

AtHome 方法，Spel 類別

描述

若目前機器人位於起始點位置，會傳回 True。

語法

Function **AtHome** () As Boolean

傳回值

若目前機器人位於起始點位置，會傳回 True，否則會傳回 False。

另請參閱

Home

AtHome 範例

VB 例:

```
If m_spel.AtHome() Then
    lblCurPos.Text = "Robot is at home position"
Else
    lblCurPos.Text = "Robot is not at home position"
End If
```

C# 例:

```
if(m_spel.AtHome())
    lblCurPos.Text = "Robot is at home position";
else
    lblCurPos.Text = "Robot is not at home position";
```

ATRQ 方法，Spel 類別

描述

傳回指定關節的有效轉矩。

語法

Function **ATRQ** (*JointNumber* As Integer) As Single

參數

JointNumber	軸編號的整數值 (範圍: 1~ 機器人的關節數)

傳回值

0~1 的實數值。

另請參閱

ATCLR, PTCLR, PTRQ

ATRQ 範例

VB 例:

```
Dim val As Single
Dim i As Integer
For i = 1 To 4
  val = m_spel.ATRQ(i)
Next i
```

C# 例:

```
float avgTorque;
for(int i = 1; i <= 4; i++)
    avgTorque = m_spel.ATRQ(i);
```

AvgSpeed 方法，Spel 類別

描述

傳回指定關節速度的平均絕對值。

語法

Function **AvgSpeed** (*JointNumber* As Integer) As Single

參數

JointNumber　軸編號的整數值 (範圍: 1~ 機器人的關節數)

傳回值

0~1 的實數值。

備註

本方法傳回指定關節速度的平均絕對值。本方法可以顯示馬達的負載狀態。結果由介於 0 到 1 之間的實數表示。最大平均速度是 1。

在執行本方法前，必須先執行 AvgSpeedClear 方法。

本方法具有時間限制。執行 AvgSpeedClear 方法后，請在 60 秒內執行本方法。超過 60 秒，就會出現錯誤 4008。

對於虛擬控制器和乾運轉，從命令速度而不是實際速度，計算速度的平均絕對值。

本方法不支援 PG 附加軸。

另請參閱

AvgSpeedClear, PeakSpeed, PeakSpeedClear

AvgSpeed 範例

VB 例:

```
Dim val As Single
Dim i As Integer
For i = 1 To 4
  val = m_spel.AvgSpeed(i)
Next i
```

C# 例:

```
float avgSpeed;
for(int i = 0; i <=4; i++)
    avgSpeed = m_spel.AvgSpeed(i);
```

AvgSpeedClear 方法，Spel 類別

描述

清除並初始化關節速度的平均絕對值。

語法

Sub **AvgSpeedClear** ()

備註

本方法清除指定關節速度的平均絕對值。

執行 AvgSpeed 方法前，必須先執行本方法。

本方法不支援 PG 附加軸。

另請參閱

AvgSpeed, PeakSpeed, PeakSpeedClear

AvgSpeedClear 範例

VB 例:

```
m_spel.AvgSpeedClear()
```

C# 例:

```
m_spel.AvgSpeedClear();
```

AxisLocked 方法，Spel 類別

描述

若指定軸受伺服系統控制，會傳回 True。

語法

Function **AxisLocked** (*AxisNumber* As Integer) As Boolean

參數

AxisNumber 代表軸編號的數值運算式。數值可以介於 1 – 9。

傳回值

若指定軸受伺服系統控制，會傳回 True。

另請參閱

SLock, SFree

AxisLocked 範例

VB 例:

```
If m_spel.AxisLocked(1) Then
    lblAxis1.Text = "Robot axis #1 is locked"
Else
    lblAxis1.Text = "Robot axis #1 is free"
End If
```

C# 例:

```
if (m_spel.AxisLocked(1))
    lblAxis1.Text = "Robot axis #1 is locked";
else
    lblAxis1.Text = "Robot axis #1 is free";
```

Base 方法，Spel 類別

描述

定義基座座標系統。

語法

Sub **Base** (*OriginPoint* As SpelPoint [, *XAxisPoint* As SpelPoint] [, *YAxisPoint* As SpelPoint]
[, *Alignment* As SpelBaseAlignment])

參數

OriginPoint	代表基座座標系統之原點的 SpelPoint。
XAxisPoint	位於基座座標系統 X 軸上任何位置的 SpelPoint。
YAxisPoint	位於基座座標系統 Y 軸上任何位置的 SpelPoint。
Alignment	當提供 *XAxisPoint* 和 *YAxisPoint* 參數時，請使用 Alignment 參數指定要與基座對齊的軸。

另請參閱

Local

Base 範例

VB 例:

```
Dim originPoint As New SpelPoint
originPoint.X = 50
originPoint.Y = 50
m_spel.Base(originPoint)
```

C# 例:

```
SpelPoint originPoint = new SpelPoint();
originPoint.X = 50;
originPoint.Y = 50;
m_spel.Base(originPoint);
```

描述

在所選本地座標系統中執行 PTP 相對動作。

語法

Sub **BGo** (*PointNumber* As Integer)
Sub **BGo** (*Point* As SpelPoint)
Sub **BGo** (*Point* As SpelPoint, *AttribExpr* As String)
Sub **BGo** (*PointExpr* As String)

參數

每個語法都有一個參數，用以指定在 BGo 動作期間手臂移動的結束點。此為 PTP 動作結束時的最終位置。

PointNumber	透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
Point	透過使用 SpelPoint 資料類型來指定結束點。
AttribExpr	透過使用字串運算式來指定結束點屬性。
PointExpr	透過使用字串運算式來指定結束點。

另請參閱

Accel, Speed
Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move
BMove, TGo, TMove
CP, Till

BGo 範例

VB 例:

```
' 使用點編號
m_spel.Tool(1)
m_spel.BGo(100)

' 使用 SpelPoint
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.BGo(pt)

' 使用屬性式
m_spel.BGo(pt, "ROT")

' 使用點運算式
m_spel.BGo("P0 /L /2")
m_spel.BGo("P1 :Z(-20)")

' 使用平行處理
m_spel.BGo("P1 !D50; On 1; D90; Off 1!")

' 使用點標籤
m_spel.BGo("pick")
```

C# 例:

```
// 使用點編號
m_spel.Tool(1);
m_spel.BGo(100);

// 使用 SpelPoint
SpelPoint pt;
pt = m_spel.GetPoint("P*");
pt.X = 125.5;
m_spel.BGo(pt);

// 使用屬性式
m_spel.BGo(pt, "ROT");

// 使用點運算式
m_spel.BGo("P0 /L /2");
m_spel.BGo("P1 :Z(-20)");

// 使用平行處理
m_spel.BGo("P1 !D50; On 1; D90; Off 1!");

// 使用點標籤
m_spel.BGo("pick");
```

BMove 方法，Spel 類別

描述

在所選本地座標系統中執行線性插補相對動作。

語法

Sub **BMove** (*PointNumber* As Integer)
Sub **BMove** (*Point* As SpelPoint)
Sub **BMove** (*Point* As SpelPoint, *AttribExpr* As String)
Sub **BMove** (*PointExpr* As String)

參數

每個語法都有一個參數，用以指定在 BMove 動作期間手臂移動的結束點。此為線性插補動作結束時的最終位置。

PointNumber	透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
Point	透過使用 SpelPoint 資料類型來指定結束點。
AttribExpr	透過使用字串運算式來指定結束點屬性。
PointExpr	透過使用字串運算式來指定結束點。

另請參閱

AccelR, AccelS, SpeedR, SpeedS
Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move
BGo, TGo, TMove
CP, Till

BMove 範例

VB 例:

```
' 使用點編號
m_spel.Tool(1)
m_spel.BMove(100)

' 使用 SpelPoint
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.BMove(pt)

' 使用點運算式
m_spel.BMove("P0 /L /2")

' 使用平行處理
m_spel.BMove("P1 !D50; On 1; D90; Off 1!")

' 使用點標籤
m_spel.BMove("pick")
```

C# 例:

```
// 使用點編號
m_spel.Tool(1);
m_spel.BMove(100);

// 使用 SpelPoint
SpelPoint pt;
pt = m_spel.GetPoint("P*");
pt.X = 125.5;
m_spel.BMove(pt);

// 使用點運算式
m_spel.BMove("P0 /L /2");

// 使用平行處理
m_spel.BMove("P1 !D50; On 1; D90; Off 1!");

//使用點標籤
m_spel.BMove("pick");
```

Box 方法，Spel 類別

描述

指定在盒體內定義的接近檢查區域。

語法

Sub **Box** (*AreaNumber* As Integer, *MinX* As Single, *MaxX* As Single, *MinY* As Single, *MaxY* As Single, *MinZ* As Single, *MaxZ* As Single)
Sub **Box** (*AreaNumber* As Integer, *MinX* As Single, *MaxX* As Single, *MinY* As Single, *MaxY* As Single, *MinZ* As Single, *MaxZ* As Single, *PolarityOn* As Boolean)

參數

AreaNumber	介於 1-15 的整數，代表 15 個方塊中要定義的方塊。
MinX	接近檢查區域的最小 X 座標位置。
MaxX	接近檢查區域的最大 X 座標位置。
MinY	接近檢查區域的最小 Y 座標位置。
MaxY	接近檢查區域的最大 Y 座標位置。
MinZ	接近檢查區域的最小 Z 座標位置。
MaxZ	接近檢查區域的最大 Z 座標位置。
PolarityOn	選用。設定在使用對應遠端輸出時的遠端輸出邏輯。若要在夾具末端位於盒體區域中時將 I/O 輸出設為開啟，請使用 True。若要在夾具末端位於盒體區域中時將 I/O 輸出設為關閉，請使用 False。

另請參閱

BoxClr, BoxDef, Plane

Box 範例

VB 例:

```
m_spel.Box(1, -5, 5, -10, 10, -20, 20)
```

C# 例:

```
m_spel.Box(1, -5, 5, -10, 10, -20, 20);
```

BoxClr 方法，Spel 類別

描述

清除 box(接近檢查區域)的定義。

語法

Sub **BoxClr** (*BoxNumber* As Integer)

參數

BoxNumber　　介於 1 至 15 的整數運算式，代表區域編號。

另請參閱

Box, BoxDef

BoxClr 範例

VB 例:

```
m_spel.BoxClr(1)
```

C# 例:

```
m_spel.BoxClr(1);
```

BoxDef 方法，Spel 類別

描述

傳回 Box 是否定義。

語法

Function **BoxDef** (*BoxNumber* As Integer) As Boolean

參數

BoxNumber　　介於 1 至 15 的整數運算式，代表區域編號。

傳回值

若指定 box 已定義，會傳回 True，否則會傳回 False。

另請參閱

Box, BoxClr

BoxDef 範例

VB 例:

```
x = m_spel.BoxDef(1)
```

C# 例:

```
x = m_spel.BoxDef(1);
```

Brake 方法，Spel 類別

描述

讀取或設定關節制的制動狀態。

語法

Sub **Brake** (*JointNumber* As Integer, *State* As Boolean)

Function **Brake** (*JointNumber* As Integer) As Boolean

參數

JointNumber	軸編號的整數值 (範圍: 1~ 機器人的關節數)
State	開啓制動器 : 使用 On。 解除制動器 : 使用 Off。

傳回值

0 = 制動器 Off

1 = 制動器 On

備註

本方法可以對垂直六軸型機器人(含 N 系列)的單個關節開啓或解除制動。此方法旨在僅供維護人員使用。

執行本方法，會初始化機器人的控制參數。

警 告	■ 解除制動器時要小心。確保關節被正確固定。如果關節沒有正確固定，則可能導致墜落，從而導致機器人故障或操作員受傷。

注意

在執行 Brake Off 命令之前，請確保將緊急停止開關放在手邊。

當控制器處於緊急停止狀態時，馬達制動器將被鎖定。執行 Brake Off 命令時，機械臂可能會由於自生重量而下降。

另請參閱

Reset, SFree, SLock

Brake 範例

VB 例:

```
Dim state As Boolean
state = m_spel.Brake(1)
```

C# 例:

```
bool state;
state = m_spel.Brake(1);
```

BTst 方法，Spel 類別

描述

以數值傳回 1 位元的狀態。

語法

Function **BTst** (*Number* As Integer, *BitNumber* As Integer) As Boolean

參數

Number	以運算式或數值指定位元測試的數字。
BitNumber	指定要測試的位元(介於 0 至 31 的整數)。

傳回值

若指定位元已設定，會傳回 True，否則會傳回 False。

另請參閱

On, Off

BTst 範例

VB 例:

```
x = m_spel.BTst(data, 2)
```

C# 例:

```
x = m_spel.BTst(data, 2);
```

BuildProject 方法，Spel 類別

描述

建置透過 Project 屬性所指定的 EPSON RC+ 7.0 專案。

語法

Sub **BuildProject** ()

另請參閱

Project, ProjectBuildComplete, ProjectOverwriteWarningEnabled

BuildProject 範例

VB 例:

```
With m_spel
    .Project = "c:\EpsonRC70\projects\myproj\myproj.sprj"
    If Not .ProjectBuildComplete() Then
        .BuildProject()
    End If
End With
```

C# 例:

```
m_spel.Project =
@"c:\EpsonRC70\projects\myproj\myproj.sprj";
if(!m_spel.ProjectBuildComplete())
    m_spel.BuildProject();
```

Call 方法，Spel 類別

描述

調用(執行)可選擇性傳回數值的 SPEL+函數。

語法

Function **Call** (*FuncName* As String [, *Parameters* As String]) As Object

參數

FuncName	已在目前專案中定義之函數的名稱。
Parameters	選用。 指定引數清單。使用以逗號(,)分隔的引數。

傳回值

SPEL+函數的傳回值。資料類型符合函數的資料類型。

備註

使用 Call 方法調用 SPEL+函數並擷取傳回值。將 Call 的結果指派至變數時，請確定使用正確的資料類型，否則會發生類型不相符錯誤。

您也可以從 Visual Basic 應用程式調用在 SPEL+程式碼中宣告的 DLL 函數。透過 Call 方法執行的函數無法由 Stop、Pause、Halt 或 Quit 方法停止／暫停。

如果需要停止或暫停，請使用 Xgt 方法。

注意

Call 方法執行的函數，不能被 Stop, Pause, Halt 與 Quit 方法停止或暫停。

要停止或暫停，請使用 Xqt 方法。

另請參閱

Xqt

Call 範例

```
' Visual Basic 程式碼
Dim errCode As Integer
errCode = m_spel.Call("GetPart", """Test"",2")

' C# 程式碼
int errCode;
errCode = m_spel.Call("GetPart", """Test"",2");

' SPEL+ 函數
Function GetPart(Info$ As String, Timeout As Integer) As
Integer
    Long errNum
OnErr GoTo GPErr
    Print Info$
    errNum = 0
    Jump P1
    On vacuum
    Wait Sw(vacOn) = On, Timeout
    If TW = True Then
        errNum = VAC_TIMEOUT
    EndIf

    GetPart = errNum
    Exit Function
GPErr:
    GetPart = Err
Fend
```

CalPls 方法，Spel 類別

描述

讀取或設定用於校準的位置姿態脈衝值。

語法

Function CalPls (JointNumber As Integer) As Integer

Sub CalPls (J1Pulses As Integer, J2Pulses As Integer, J3Pulses As Integer, J4Pulses As Integer, [J5Pulses As Integer], [J6Pulses As Integer], [J7Pulses As Integer], [J8Pulses As Integer], [J9Pulses As Integer])

參數

J1Pulses – J9Pulses	顯示第 1~9 關節的脈衝值得整數 *J5Pulses – J9Pulses* 可以省略

傳回值

如果省略脈衝值，則顯示當前設定的脈衝值。

備註

輸入並保存正確的脈衝值，用於校準位置。

本方法僅供維護時使用。當馬達的原點不匹配機械臂的機械原點時(例如更換馬達時)使用。匹配原點的操作稱爲校準。

在正常情況下，校準位置的脈衝值與本方法中設定的脈衝值匹配。但在維護(例如更換馬達)後，這兩個值不匹配，需要校準。

校準的一種方法是，將關節移動到特定位置並執行 Calib。執行 Calib 會將校準特定位置的脈衝值，更改為本方法中設定的脈衝值(校準位置的正確脈衝值)。

必須設置 Hofs 值才能執行校準。要自動計算 Hofs 值，請先將關節移動到特定的位置，然後執行 Calib。控制器會根據校準位置的脈衝值，自動計算 Hofs 的值。

注意

CalPls 值不能透過關閉電源變更

關閉控制器然後重新啟動時，CalPls 的值不會被初始化。更改 CalPls 值的唯一方法是執行 Calib 命令。

另請參閱

Hofs

CalPls 範例

VB 例:

```
Dim val As Single
Dim i As Integer
For i = 1 To 4
  val = m_spel.CalPls(i)
Next i
```

C# 例:

```
float val;
for(int i = 1; i <= 4; i++)
    val = m_spel.CalPls(i);
```

ClearPoints 方法，Spel 類別

描述

清除目前機器人記憶體中的點。

語法

Sub **ClearPoints** ()

另請參閱

LoadPoints, Robot, SavePoints, SetPoint

ClearPoints 範例

VB 例:

```
With m_spel
    .ClearPoints()
    .SetPoint(1, 100, 200, -20, 0, 0, 0)
    .Jump(1)
End With
```

C# 例:

```
m_spel.ClearPoints();
m_spel.SetPoint(1, 100, 200, -20, 0, 0, 0);
m_spel.Jump(1);
```

Connect 方法，Spel 類別

描述

連接 Spel 類別執行個體與控制器。

語法

Sub **Connect** (*ConnectionName* As String)

Sub **Connect** (*ConnectionName* As String, ConnectionPassword As String)

Sub **Connect** (*ConnectionNumber* As Integer)

Sub **Connect** (*ConnectionNumber* As Integer, ConnectionPassword As String)

參數

ConnectionName	表示連接名稱的字串。。
ConnectionNumber	連線號碼的整數運算式。 連接號碼是 EPSON RC+中[配置] - [電腦與控制器連接]對話方塊的號碼。 若要使用上次成功連接的目標，請指定 -1。
ConnectionNumber	連線密碼的字串。 如果控制器具有用與控制器連接的密碼，並且密碼未在 EPSON RC+中[配置] - [電腦與控制器連接]對話方塊中設置，則必須設置連接到控制器的密碼。

備註

當 Spel 類別執行個體需要與控制器進行通訊時，它會自動連線。若要明確地連線至控制器，請使用 Connect 方法。

另請參閱

Disconnect, Initialize

Connect 範例

VB 例:

```
Try
  m_spel.Connect(1)
Catch ex As RCAPINet.SpelException
  MsgBox(ex.Message)
End Try
```

C# 例:

```
try{
    m_spel.Connect(1);
}
catch(RCAPINet.SpelException ex){
    MessageBox.Show(ex.Message);
}
```

Continue 方法，Spel 類別

描述

發生暫停時，恢復控制器的所有任務。

語法

Sub **Continue** ()

備註

使用 **Continue** 可恢復因 Pause 方法或安全防護打開而暫停的所有任務。

當安全防護在任務執行期間打開時，機器人將會減速至停止狀態，且機器人馬達將會關閉。關閉安全防護後，您可使用 **Continue** 恢復週期。

另請參閱

Pause, Start, Stop

Continue 範例

VB 例:

```
Sub btnContinue_Click( _
     ByVal sender As System.Object, _
     ByVal e As System.EventArgs) Handles btnContinue.Click

    btnPause.Enabled = True
    btnContinue.Enabled = False
  Try
    m_spel.Continue()
  Catch ex As RCAPINet.SpelException
    MsgBox(ex.Message)
  End Try
End Sub
```

C# 例:

```
void btnContinue_Click(object sender, EventArgs e)
{
btnPause.Enabled = true;
   btnContinue.Enabled = true;

    try{
                m_spel.Continue();
      }
      catch(RCAPINet.SpelException ex){
                MessageBox.Show(ex.Message);
      }
}
```

Ctr 方法，Spel 類別

描述

傳回指定輸入計數器的計數器值。

語法

Function **Ctr** (*BitNumber* As Integer) As Integer

參數

BitNumber　輸入位元設為計數器的數值。
只能同時啟用 16 個計數器。

傳回值

傳回計數器值。(0 至 65535 的整數)

另請參閱

CtReset

Ctr 範例

VB 例:

```
lblCounter.Text = m_spel.Ctr(1).ToString()
```

C# 例:

```
lblCounter.Text = m_spel.Ctr(1).ToString();
```

CtReset 方法，Spel 類別

描述

重置指定輸入計數器的計數器值。同時將輸入定義為計數器輸入。

語法

Sub **CtReset** (*BitNumber* As Integer)

參數

BitNumber　輸入位元設為計數器的數值。
只能同時啟用 16 個計數器。

另請參閱

Ctr

CtReset 範例

VB 例:

```
m_spel.CtReset(2)
```

C# 例:

```
m_spel.CtReset(2);
```

Curve 方法，Spel 類別

描述

定義沿著曲徑移動手臂所需的資料與點。可以在路徑中定義許多點資料，以改善路徑的精確度。

語法

Sub **Curve** (*FileName* As String, *Closure* As Boolean, *Mode* As Integer, *NumOfAxis* As Integer, *PointList* As String)

參數

FileName 點資料保存檔案的路徑與名稱之字串運算式。指定的 *fileName* 將會在末尾加上 CRV 副檔名，因此使用者無需指定副檔名。當執行 **Curve** 指令時，將會建立 *fileName*。

Closure 指定是否要將路徑的最後一個點連接至第一個點的 Boolean 運算式。

Mode 指定是否要在 U 軸的正切方向自動插補手臂。

模式設定	正切修正
0	No
2	Yes

NumOfAxis 介於 2 - 4 的整數運算式，用以指定曲線動作期間受控軸的數量，如下所述：

2: 在沒有 Z 軸移動或 U 軸旋轉的情況下，於 XY 平面產生曲線。
3: 在沒有 U 軸旋轉的情況下，於 XYZ 平面產生曲線。(Theta 1、Theta2 及 Z)
4: 在 U 軸旋轉的情況下，於 XYZ 平面產生曲線。(控制 4 個軸)

PointList { point expression | P(*start*:*finish*) } [, *output command*] ...
此參數實際上是連續的點編號，以及用逗號分隔的選用輸出陳述式，或是以冒號分隔的遞增點範圍。
以逗號分隔的連續點通常如下所示：

```
Curve MyFile, O, 0, 4, P1, P2, P3, P4
```

或以冒號指定，如下所示：

```
Curve MyFile, O, 0, 4, P(1:4)
```

備註

使用 Curve 來定義以 CVMove 方法執行的樣條路徑。如需詳細資訊，請參閱 SPEL+ 的 Curve 命令。

另請參閱

Curve (SPEL+ Statement), CVMove Method

Curve 範例

VB 例:

```
m_spel.Curve("mycurveFile", True, 0, 4, "P(1:3), On 1,
P(4:7)")
m_spel.CVMove("mycurveFile")
```

C# 例:

```
m_spel.Curve("mycurveFile", True, 0, 4, "P(1:3), On 1,
P(4:7)");
m_spel.CVMove("mycurveFile");
```

CVMove 方法，Spel 類別

描述

執行由 **Curve** 指令定義的連續樣條路徑動作。

語法

Sub **CVMove** (*FileName* As String [, *OptionList* As String])

參數

FileName 要用於連續路徑動作資料之檔案的路徑與名稱的字串運算式。此檔案必須先使用 Curve 指令建立。

OptionList 選用。包含 Till 規格的字串運算式。

備註

使用 CVMove 來執行以 Curve 方法定義的路徑。如需詳細資訊，請參閱 SPEL+命令的 **CVMove**。

若需要以 CP 執行 CVMove，建議您從 SPEL+任務來執行 CVMove，而不要從應用程式執行。這是因為要讓 CP 動作正確執行，系統必須事先知道下一個動作目標的位置。由於一次只能執行一個 RC+ API 命令，因此系統無法事先知道下一個目標的位置。

另請參閱

Curve, CVMove(SPEL+ 命令)

CVMove 範例

VB 例:

```
m_spel.Curve("mycurveFile", True, 0, 4, "P(1:3), On 1,
P(4:7)")
m_spel.CVMove("mycurveFile", "CP Till Sw(1) = 1")
m_spel.CVMove("mycurveFile")
```

C# 例:

```
m_spel.Curve("mycurveFile", True, 0, 4, "P(1:3), On 1,
P(4:7)");
m_spel.CVMove("mycurveFile", "CP Till Sw(1) = 1");
m_spel.CVMove("mycurveFile");
```

CX、CY、CZ、CU、CV、CW、CR、CS、CT 方法，Spel 類別

描述

從一點擷取座標值

CV 和 CW 用於 6 軸機器人
CS 和 CT 用於附加軸
CR 用於 Joint 7 軸機器人

語法

Function **CX** (*PointExpr* As String) As Single
Function **CY** (*PointExpr* As String) As Single
Function **CZ** (*PointExpr* As String) As Single
Function **CU** (*PointExpr* As String) As Single
Function **CV** (*PointExpr* As String) As Single
Function **CW** (*PointExpr* As String) As Single
Function **CR** (*PointExpr* As String) As Single
Function **CS** (*PointExpr* As String) As Single
Function **CT** (*PointExpr* As String) As Single

參數

PointExpr 指定用以擷取指定座標之點的字串運算式。可以使用任何有效的點運算式。**P***也可以用於從目前位置擷取座標。

傳回值

指定的座標值。
CX、CY、CZ 的傳回值：實際值(公釐)
CU、CV、CW 的傳回值：實際值(度)
CR、CS、CT 的傳回值：實際值

另請參閱

GetPoint, SetPoint

CX、CY、CZ、CU、CV、CW、CR、CS、CT 範例

VB 例:

```
Dim x As Single, y As Single
x = m_spel.CX("P1")
y = m_spel.CY("P*")
```

C# 例:

```
float x, y;
x = m_spel.CX("P1");
y = m_spel.CY("P*");
```

Delay 方法，Spel 類別

描述

指定的延遲時間(毫秒)。

語法

Sub **Delay** (*Milliseconds* As Integer)

參數

Milliseconds　包含延遲毫秒數的整數值。

Delay 範例

VB 例:

```
m_spel.Delay(500)
```

C# 例:

```
m_spel.Delay(500);
```

DegToRad 方法，Spel 類別

描述

將度轉換成徑度。

語法

Function **DegToRad** (*degrees* As Double) As Double

參數

degrees　　轉換成徑度的度數。

傳回值

包含徑度的雙精度值。

另請參閱

RadToDeg

DegToRad 範例

VB 例:

```
Dim rad As Double

rad = m_spel.DegToRad(45)
```

C# 例:

```
double rad;

rad = m_spel.DegToRad(45);
```

Disconnect 方法，Spel 類別

描述

從目前連線中斷 Spel 類別執行個體的連線。

語法

Sub **Disconnect** ()

備註

使用 **Disconnect** 可中斷目前控制器的連線。

另請參閱

Connect, Initialize

Disconnect 範例

VB 例:

```
Try
  m_spel.Disconnect()
Catch ex As RCAPINet.SpelException
  MsgBox(ex.Message)
End Try
```

C# 例:

```
try{
    m_spel.Disconnect();
}
catch(RCAPINet.SpelException ex){
MessageBox.Show(ex.Message);
}
```

ECP 方法，Spel 類別

描述

選擇目前的 ECP 定義。

語法

Sub **ECP** (*ECPNumber* As Integer)

參數

ECPNumber	介於 0-15 的整數，代表 16 個 ECP 定義中要搭配下一個動作指令使用的定義。

另請參閱

ECPSet

ECP 範例

VB 例:

```
m_spel.ECP(1)
m_spel.Move("P1 ECP")
```

C# 例:

```
m_spel.ECP(1);
m_spel.Move("P1 ECP");
```

ECPClr 方法，Spel 類別

描述

清除(取消定義)目前機器人的外部控制點。

語法

Sub **ECPClr** (*ECPNumber* As Integer)

參數

ECPNumber	代表 15 個外部控制點中要清除(取消定義)之外部控制點的整數運算式。 (ECP 0 為預設值，無法清除。)

另請參閱

ECP, ECPDef

ECPClr 範例

VB 例:

```
m_spel.ECPClr(1)
```

C# 例:

```
m_spel.ECPClr(1);
```

ECPDef 方法，Spel 類別

描述

傳回 ECP 定義狀態。

語法

Function **ECPDef** (*ECPNumber* As Integer) As Boolean

參數

ECPNumber　　代表要傳回狀態之 ECP 的整數值。

傳回值

若指定 ECP 已定義，會傳回 True，否則會傳回 False。

另請參閱

ECP, ECPClr

ECPDef 範例

VB 例:

```
x = m_spel.ECPDef(1)
```

C# 例:

```
x = m_spel.ECPDef(1);
```

ECPSet 方法，Spel 類別

描述

定義 ECP(外部控制點)。

語法

Sub **ECPSet** (*ECPNumber* As Integer, *Point As SpelPoint*)
Sub **ECPSet** (*ECPNumber* As Integer, *XCoord* as Double, *YCoord* as Double, *ZCoord* as Double, *UCoord* as Double [, *VCoord* As Double] [, *WCoord* as Double)]

參數

ECPNumber	介於 1-15 的整數，代表 15 個外部控制點中要定義的外部控制點。
Point	包含點資料的 SpelPoint。
XCoord	外部控制點 X 座標。
YCoord	外部控制點 Y 座標。
ZCoord	外部控制點 Z 座標。
UCoord	外部控制點 U 座標。
VCoord	選用。外部控制點 V 座標。
WCoord	選用。外部控制點 W 座標。

另請參閱

ArmSet, ECP, GetECP, TLSet

ECPSet 範例

VB 例:

```
m_spel.ECPSet(1, 100.5, 99.3, 0, 0)
```

C# 例:

```
m_spel.ECPSet(1, 100.5, 99.3, 0, 0);
```

EnableEvent 方法，Spel 類別

描述

針對 EventReceived 事件啟用特定系統事件。

語法

Sub **EnableEvent** (EventNumber As RCAPINet.SpelEvents, Enabled as Boolean)

參數

Event	要啟用或停用的事件。
Enabled	設為 True 可啟用事件，設為 False 則停用事件。

另請參閱

EventReceived

EnableEvent 範例

VB 例:

```
With m_spel
  .EnableEvent(RCAPINet.SpelEvents.ProjectBuildStatus, True)
  .BuildProject()
End With
```

C# 例:

```
m_spel.EnableEvent(RCAPINet.SpelEvents.ProjectBuildStatus,
true);
m_spel.BuildProject();
```

ExecuteCommand 方法，Spel 類別

描述

將命令傳送至 EPSON RC+ 7.0 並等待完成。

語法

Sub **ExecuteCommand** (*Command* As String , [ByRef *Reply* As String])

參數

Command	包含 SPEL+命令的字串。
Reply	傳回選擇性的回覆。

備註

一般而言，不需要使用 **ExecuteCommand**。執行 Spel 方法可以執行大部分的操作。不過，有時還是需執行 SPEL+多重陳述式。多重陳述式是一行命令，包含以分號分隔的多個陳述式。使用 **ExecuteCommand** 可執行多重陳述式。

例如：

```
m_spel.ExecuteCommand("JUMP pick; ON tipvac")
```

最大命令行長度為 200 個字元。

另請參閱

Pause

ExecuteCommand 範例

VB 例:

```
m_spel.ExecuteCommand("JUMP P1!D50; ON 1!")
```

C# 例:

```
m_spel.ExecuteCommand("JUMP P1!D50; ON 1!");
```

FBusIO_GetBusStatus 方法，Spel 類別

描述

傳回指定的現場匯流排的狀態。

語法

Function FBusIO_GetBusStatus (*BusNumber* As Integer) As Integer

參數

BusNumber 代表現場匯流排系統的編號的整數值。
此數字始終為 16，是連接到控制器 PC 側的現場匯流排主機板的總線的 ID。
是一個代表要定義的外部控制點的運算式，或者 1~15 的整數運算式

傳回值

0 – OK
1 – 未連接
2 – 關機

備註

注意

僅當啓用了現場匯流排主機板選配件功能時，才可以用本方法。

另請參閱

FBusIO_GetDeviceStatus, FBusIO_SendMsg, IsOptionActive

FBusIO_GetBusStatus 範例

VB 例:

```
Dim val As Integer
val = m_spel.FBusIO_GetBusStatus(16)
```

C# 例:

```
int busStatus;
busStatus = m_spel.FbusIO_GetBusStatus(16);
```

FBusIO_GetDeviceStatus 方法，Spel 類別

描述

傳回指定的現場匯流排裝置的狀態。

語法

Function FBusIO_GetDeviceStatus (BusNumber As Integer, DeviceID As Integer) As Integer

參數

BusNumber 代表現場匯流排系統的編號的整數值。
此數字始終為 16，是連接到控制器 PC 側的現場匯流排主機板的總線的 ID。
是一個代表要定義的外部控制點的運算式，或者 1~15 的整數運算式

DeviceID 代表裝置的現場匯流排 ID 的整數值

傳回值

0 – OK

1 – 未連接

2 – 關機

3 – 同步錯誤: 裝置正在初始化，或裝備的傳輸速率不正確。

備註

注意

僅當啓用了現場匯流排主機板選配件功能時，才可以用本方法。

另請參閱

FBusIO_GetBusStatus, FBusIO_SendMsg, IsOptionActive

FBusIO_GetDeviceStatus 範例

VB 例:

```
Dim val As Integer
val = m_spel.FBusIO_GetDeviceStatus(16, 10)
```

C# 例:

```
int deviceStatus;
devideStatus = m_spel.FbusIO_GetDeviceStatus(16, 10);
```

FBusIO_SendMsg 方法，Spel 類別

描述

向現場匯流排 I/O 裝置發送訊息，並傳回回答。

語法

Sub FBusIO_SendMsg (BusNumber As Integer, DeviceID As Integer, MsgParam As Integer, SendData As Byte(), ByRef RecvData As Byte())

參數

BusNumber 代表現場匯流排系統的編號的整數值。
此數字始終為 16，是連接到控制器 PC 側的現場匯流排主機板的總線的 ID。
是一個代表要定義的外部控制點的運算式，或者 1~15 的整數運算式

DeviceID 代表裝置的現場匯流排 ID 的整數值

MsgParam 代表訊息參數的整數陳述式。
不支援 DeviceNet。

SendData 在位元組類型的陣列中，指定發送給裝置的資料。此陣列尺寸必須和發送的位元組尺寸相同。如果不發送訊息，請指定為"0"。

RecvData 在位元組類型的陣列中，指定從裝置接收的資料。此陣列會根據接收到的位元組數，自動轉換尺寸大小。

備註

注意

僅當啓用了現場匯流排主機板選配件功能時，才可以用本方法。

另請參閱

FBusIO_GetBusStatus, FBusIO_GetDeviceStatus, IsOptionActive

FBusIO_SendMsg 範例

VB 例:

```
'向 DeviceNet 發送訊息
Dim recvData() as Byte
Dim sendData(6) as Byte
Array.Clear(sendData, 0, sendData.Length)
sendData(0) = 14 '命令
sendData(1) = 1 '類別
sendData(3) = 1 '範例
sendData(5) = 7 '屬性
' DeviceNet的 MsgParam是 0
m_spel.FbusIO_SendMsg(16, 1, 0, sendData, recvData)

' 向 Profibus 發送訊息
Dim recvData() As Byte;
m_spel.FbusIO_SendMsg(16, 1, 56, Nothing, recvData);
```

C# 例:

```
// 向 DeviceNet 發送訊息
byte[] sendData, recvData;
byte[] sendData = new byte[6];
Array.Clear(sendData, 0, sendData.Length);
sendData[0] = 14; //命令
sendData[1] = 1; //類別
sendData[3] = 1; //範例
sendData[5] = 7; //屬性
// DeviceNet的 MsgParam 是 0
m_spel.FbusIO_SendMsg(16, 201, 0, sendData, out recvData);

// 向 Profibus 發送訊息
byte[] recvData;
m_spel.FbusIO_SendMsg(16, 1, 56, null, out recvData);
```

FGGet 方法，Spel 類別

描述

取得力覺引導序列，或者力覺引導物件的結果。

語法

Sub FGGet (Sequence As String, [Object As String], Property As SpelForceProps, ByRef Result As Boolean)

Sub FGGet (Sequence As String, [Object As String], Property As SpelForceProps, ByRef Result As Double)

Sub FGGet (Sequence As String, [Object As String], Property As SpelForceProps, ByRef Result As Integer)

Sub FGGet (Sequence As String, [Object As String], Property As SpelForceProps, ByRef Result As String)

參數

Sequence	力覺引導序列名稱，或代表力覺引導序列名稱的字串
Object	力覺引導物件的名稱，或代表力覺引導物件名稱的字串變數 當取得力覺引導序列的結果時可以省略。
Property	獲得值的結果的名稱
Result	代表傳回值得變數 數和陳述式因結果而異。

另請參閱

FGRun

FGGet 範例

VB 例:

```
Dim val As Integer
m_spel.MotorsOn = True

m_spel.FGRun("Sequence1")
m_spel.FGGet("Sequence1","", SpelForceProps.EndStatus, val)
```

C# 例:

```
int errCode;
m_spel.MotorsOn = true;

m_spel.FGRun("Sequence1");
m_spel.FGGet("Sequence1","", SpelForceProps.EndStatus, val);
```

FGRun 方法，Spel 類別

描述

執行力覺引導序列。

語法

Sub FGRun (Sequence As String)

參數

Sequence　力覺引導序列的名稱，或者代表力覺引導序列名稱的字串變數

備註

執行指定的力覺引導序列。力覺引導序列從執行了 FGRun 語句的位置開始。在執行之前，先用移動命令 Go 語句或 Move 語句移動到預期的起點位置。

FGRun 會在指定的力覺引導序列結束之後，進入下一個語句。
使用 SGGet 取得 FGRun 中執行的力覺引導序列的結果。
當在 CP 參數或 CP 語句中啟用了路徑運動，請等待其停止，然後執行力覺引導序列。

開始執行時，滿足以下條件之一，則錯誤。

- 程式中指定的機器人和 RobotNumber 屬性中指定的機器人相異
  在 Robot 語句中指定正確的機器人。
- 程式中指定的機器人類型和 RobotType 屬性中指定的機器人類型相異
  在 Robot 語句中指定正確的機器人。
- 程式中指定的工具編號和 RobotTool 屬性中指定的工具編號相異
  在 Tool 語句中指定正確的工具編號。
- 馬達關閉狀態
  在 Motor 語句中打開馬達。
- 力覺控制機能執行中
  在 FCEnd 語句中停止力覺控制。
- 傳送帶追蹤運行中
  在 Cnv_AbortTrack 語句中停止傳送帶追蹤。
- 轉矩控制模式啟用中
  在 TC 語句中禁用轉矩控制模式。

執行 FGRun 會自動重寫以下屬性，請不要和以下屬性一起使用。

FM 物件
AvgForceClear 屬性
PeakForceClear 屬性

程式執行中，無法執法執行本方法。

另請參閱

FGGet

FGRun 範例

VB 例:

```
Dim errCode As Integer
m_spel.MotorsOn = True

m_spel.FGRun("Sequence1")
errCode = m_spel.FGGet("Sequence1",
SpelForceProps.EndStatus, val)
```

C# 例:

```
int errCode;
m_spel.MotorsOn = true;

m_spel.FGRun("Sequence1");
errCode = m_spel.FGGet("Sequence1",
SpelForceProps.EndStatus, val);
```

Find 方法，Spel 類別

描述

設定要在動作指令期間儲存座標的條件。

語法

Sub **Find** (*Condition* As String)

參數

Condition　指定輸入狀態作為觸發。

另請參閱

Go, Jump

Find 範例

VB 例:

```
m_spel.Find("Sw(5) = On")
```

C# 例:

```
m_spel.Find("Sw(5) = On");
```

Fine 方法，Spel 類別

描述

指定並顯示目標點的定位準確度。

語法

Sub **Fine** (*J1MaxErr* As Integer, *J2MaxErr* As Integer, *J3MaxErr* As Integer,
J4MaxErr As Integer , *J5MaxErr* As Integer, *J6MaxErr* As Integer
[, *J7MaxErr* As Integer] [, *J8MaxErr* As Integer] [, *J9MaxErr* As Integer])

參數

J1MaxErr – *J9MaxErr* 介於 0-32767 的整數，代表各點允許的定位誤差。
關節 7、8 及 9 的值為選用。

另請參閱

Weight

Fine 範例

VB 例:

```
m_spel.Fine(1000, 1000, 1000, 1000, 0, 0)
```

C# 例:

```
m_spel.Fine(1000, 1000, 1000, 1000, 0, 0);
```

Force_Calibrate 方法，Spel 類別

描述

將目前力覺感測器的所有軸設為零偏移。

語法

Sub **Force_Calibrate( )**

備註

當應用程式啟動時，您應針對每個感測器調用 Force_Calibrate。此將會說明感測器上裝載之元件的重量。

注意

僅當安裝力感測選配(ATI 力感測器)時，此方法才可用。

使用愛普生力覺感測器時，可使用 FGGet 方法和 FGRun 方法。

另請參閱

Force_Sensor, Force_GetForces, Force_SetTrigger

Force_Calibrate 範例

VB 例:

```
m_spel.ForceSensor = 1
m_spel.Force_Calibrate()
```

C# 例:

```
m_spel.ForceSensor = 1;
m_spel.Force_Calibrate();
```

Force_ClearTrigger 方法，Spel 類別

描述

清除目前力覺感測器的所有觸發條件。

語法

Sub **Force_ClearTrigger( )**

備註

使用 Force_ClearTrigger 可清除目前力覺感測器觸發的所有條件。

注意

僅當安裝力感測選配(ATI 力感測器)時，此方法才可用。

使用愛普生力覺感測器時，可使用 FGGet 方法和 FGRun 方法。

另請參閱

Force_Sensor, Force_GetForces, Force_SetTrigger

Force_ClearTrigger 範例

VB 例:

```
m_spel.ForceSensor = 1
m_spel.Force_ClearTrigger()
```

C# 例:

```
m_spel.ForceSensor = 1;
m_spel.Force_ClearTrigger();
```

Force_GetForce 方法，Spel 類別

描述

傳回指定力覺感測器軸的力。

語法

Function **Force_GetForce(** *Axis* As SpelForceAxis**)** As Single

參數

Axis 要擷取的軸數值，如下所示：

SpelForceAxis	數值
XForce	1
YForce	2
ZForce	3
XTorque	4
YTorque	5
ZTorque	6

備註

使用 Force_GetForce 可讀取一個軸的目前力覺設定。單位是由力覺感測器設定來決定。

注意

僅當安裝力感測選配(ATI 力感測器)時，此方法才可用。

使用愛普生力覺感測器時，可使用 FGGet 方法和 FGRun 方法。

另請參閱

Force_Sensor, Force_GetForces, Force_SetTrigger

Force_GetForce 範例

VB 例:

```
m_spel.ForceSensor = 1
zForce = m_spel.Force_GetForce(SpelForceAxis.ZForce)
```

C# 例:

```
m_spel.ForceSensor = 1;
zForce = m_spel.Force_GetForce(SpelForceAxis.ZForce);
```

Force_GetForces 方法，Spel 類別

描述

傳回陣列中所有力覺感測器軸的力與力矩。

語法

Sub **Force_GetForces(** *Values()* As Single**)**

參數

Values 將會以 6 個元素傳回的單一陣列。

備註

使用 Force_GetForces 可立即讀取所有力與力矩值。

注意

僅當安裝力感測選配(ATI 力感測器)時，此方法才可用。

使用愛普生力覺感測器時，可使用 FGGet 方法和 FGRun 方法。

另請參閱

Force_Sensor, Force_GetForces, Force_SetTrigger

Force_GetForces 範例

VB 例:

```
Dim values() as Single = Nothing
m_spel.ForceSensor = 1
m_spel.Force_GetForces(values)
```

C# 例:

```
float[] values = new float[6];

m_spel.ForceSensor(1);
m_spel.Force_GetForces(values);
```

Force_SetTrigger 方法，Spel 類別

描述

設定 Till 命令的力覺觸發。

語法

Sub **Force_SetTrigger(** *Axis* As SpelForceAxis, *Threshold* As Single, *CompareType* As SpelForceCompareType **)**

參數

Axis 用於觸發的軸，如下所示：

SpelForceAxis	數值
XForce	1
YForce	2
ZForce	3
XTorque	4
YTorque	5
ZTorque	6

Threshold 代表閾值的單一運算式。

CompareType LessOrEqual 或 GreatorOrEqual。

注意

僅當安裝力感測選配(ATI 力感測器)時，此方法才可用。

使用愛普生力覺感測器時，可使用 FGGet 方法和 FGRun 方法。

備註

若要使用力覺感測器停止動作，您必須設定感測器的觸發，然後在動作陳述式中使用 Till Force。

您可以使用多個軸來設定觸發。針對每個軸調用 Force_SetTrigger。

若要清除所有觸發條件，請使用 Force_ClearTrigger。

另請參閱

Force_ClearTrigger, Force_Sensor, Till

Force_SetTrigger 範例

VB 例:

```
m_spel.ForceSensor = 1
m_spel.Force_SetTrigger(SpelForceAxis.ZForce, -2.0, _
    SpelForceCompareType.GreaterOrEqual)
m_spel.Till("Force")
m_spel.Move("P1 Till")
```

C#例:

```
m_spel.ForceSensor = 1;
m_spel.Force_SetTrigger(SpelForceAxis.ZForce, -2.0,
    SpelForceCompareType.GreaterOrEqual);
m_spel.Till("Force");
m_spel.Move("P1 Till");
```

GetAccel 方法，Spel 類別

描述

傳回指定的加速／減速值。

語法

Function **GetAccel** (*ParamNumber* As Integer) As Integer

參數

ParamNumber 為可包含下列數值的整數運算式：
1：加速規格值
2：減速規格值
3：Jump 的起始加速規格值
4：Jump 的起始減速規格值
5：Jump 的接近加速規格值
6：Jump 的接近減速規格值

傳回值

包含指定加速／減速值的整數。

另請參閱

Accel

GetAccel 範例

VB 例:

```
Dim x As Integer
x = m_spel.GetAccel(1)
```

C# 例:

```
int x;
x = m_spel.GetAccel(1);
```

GetArm 方法，Spel 類別

描述

傳回目前機器人的目前手臂編號。

語法

Function **GetArm** () As Integer

傳回值

包含目前手臂編號的整數。

另請參閱

Arm, ArmSet, Robot, Tool

GetArm 範例

VB 例:

```
saveArm = m_spel.GetArm()
m_spel.Arm(2)
```

C# 例:

```
saveArm = m_spel.GetArm();
m_spel.Arm(2);
```

GetConnectionInfo 方法，Spel 類別

描述

傳回有關控制器連線的資訊。

語法

Function **GetConnectionInfo**() As SpelConnectionInfo()

傳回值

SpelConnectionInfo 的陣列。

另請參閱

GetControllerInfo

備註

GetConnectionInfo 會傳回 SpelConnectionInfo 的陣列。連線資訊是在 EPSON RC+ 的[Setup]-[PC to Controller Communication]對話方塊中設定。

GetConnectionInfo 範例

VB 例:

```
Dim info() As SpelConnectionInfo

info = m_spel.GetConnectionInfo()
```

C# 例:

```
SpelConnectionInfo[] info = m_spel.GetConnectionInfo();
```

GetControllerInfo 方法，Spel 類別

描述

傳回有關目前控制器的資訊。

語法

Function **GetControllerInfo**() As SpelControllerInfo

傳回值

SpelControllerInfo 執行個體。

另請參閱

GetErrorMessage, GetRobotInfo, GetTaskInfo

備註

GetControllerInfo 會傳回 SpelControllerInfo 類別的新執行個體，其中包含控制器資訊屬性。

GetControllerInfo 範例

VB 例:

```
Dim info As SpelControllerInfo
Dim msg As String

info = m_spel.GetControllerInfo()
msg = "Project Name: " & info.ProjectName & vbCrLf _
    & "Project ID: " & info.ProjectID
MsgBox(msg)
```

C# 例:

```
SpelControllerInfo info;
string msg;

info = m_spel.GetControllerInfo();
msg = "Project Name:" + info.ProjectName + "\r\n
ProjectID :" +
"info.ProjectID";
MessageBox.Show(msg);
```

GetCurrentConnectionInfo 方法，Spel 類別

描述

傳回目前控制器的資訊。

語法

Function **GetCurrentConnectionInfo** () As SpelConnectionInfo

傳回值

SpelConnectionInfo

另請參閱

GetConnectionInfo, GetControllerInfo

GetCurrentConnectionInfo 範例

VB 例:

```
Dim info As SpelConnectionInfo

info = m_spel.GetCurrentConnectionInfo()
```

C# 例:

```
SpelConnectionInfo info;

info = m_spel.GetCurrentConnectionInfo();
```

GetCurrentUser 方法，Spel 類別

描述

傳回目前 EPSON RC+ 7.0 使用者。

語法

Function **GetCurrentUser** () As String

傳回值

包含目前使用者的字串變數。

另請參閱

Login

GetCurrentUser 範例

VB 例:

```
Dim currentUser As String

currentUser = m_spel.GetCurrentUser()
```

C# 例:

```
string currentUser;

currentUser = m_spel.GetCurrentUser();
```

GetECP 方法，Spel 類別

描述

傳回目前機器人的目前 ECP 編號。

語法

Function **GetECP** () As Integer

傳回值

包含目前 ECP 數量的整數。

另請參閱

ECP, ECPSet

GetECP 範例

VB 例:

```
saveECP = m_spel.GetECP()
m_spel.ECP(2)
```

C# 例:

```
saveECP = m_spel.GetECP();
m_spel.ECP(2);
```

GetErrorMessage 方法，Spel 類別

描述

傳回指定錯誤或警告代碼的錯誤訊息。

語法

Function **GetErrorMessage** (*ErrorCode* As Integer) As String

參數

ErrorCode　　傳回相關錯誤訊息的錯誤代碼。

傳回值

包含錯誤訊息的字串。

另請參閱

ErrorCode

GetErrorMessage 範例

VB 例:

```
Dim msg As String

If m_spel.ErrorOn Then
  msg = m_spel.GetErrorMessage(m_spel.ErrorCode)
  MsgBox(msg)
End If
```

C# 例:

```
string msg;

if (m_spel.ErrorOn) {
    msg = m_spel.GetErrorMessage(m_spel.ErrorCode);
    MessageBox.Show(msg);
}
```

GetIODef 方法，Spel 類別

描述

取得輸入、輸出或記憶體 I/O 位元、位元組或字元的定義資訊。

語法

Sub **GetIODef**(*Type* As SpelIOLabelTypes, *Index* As Integer, ByRef *Label* as String, ByRef *Description* As String)

參數

Type 指定 I/O 類型，如下所示：

InputBit = 1 InputByte = 2 InputWord = 3

OutputBit = 4 OutputByte = 5 OutputWord = 6

MemoryBit = 7 MemoryByte = 8 MemoryWord = 9

InputReal = 10 OutputReal = 11

Index 指定位元或埠號。

Label 傳回標籤。

Description 傳回描述。

傳回值

數值會在 Label 和 Description 參數中傳回。

備註

使用 GetIODef 可取得目前專案中所有 I/O 所使用的標籤與描述。

另請參閱

SetIODef

GetIODef 範例

VB 例:

```
Dim label As String
Dim desc As String
m_spel.GetIODef(SpelIOLabelTypes.InputBit, 0, label, desc)
```

C# 例:

```
string label, desc;

m_spel.GetIODef(SpelIOLabelTypes.InputBit, 0, out label, out
desc);
```

GetJRange 方法，Spel 類別

描述

針對指定關節，傳回允許動作區域的脈衝值(範圍設定值)。

語法

Function **GetJRange** (*JointNumber* As Integer, *Bound* As Integer) As Integer

參數

JointNumber	代表關節編號的整數值 (範圍: 1~ 機器人的關節數量)
Bound	用證書指定以下 2 個值之一。 1: 指定下限脈衝值 2: 指定上限脈衝值

傳回值

傳回指定關節的範圍設定值(整數值、單位: ，脈衝)。

另請參閱

JRange

GetJRange 範例

VB 例:

```
Dim val1 As Integer
Dim val2 As Integer
val1 = m_spel.GetJRange(1, 1)
val2 = m_spel.GetJRange(1, 2)
```

C# 例:

```
int minRange, maxRange;
minRange = m_spel.GetJRange(1, 1);
maxRange = m_spel.GetJRange(1, 2);
```

GetLimitTorque 方法，Spel 類別

描述

針對目前機器人的指定關節，傳回限制力矩。

語法

Function **GetLimitTorque** (*JointNumber* As Integer) As Integer

參數

JointNumber　所需之關節的整數運算式。

傳回值

介於 1 至 9 的整數值，代表指定關節的限制力矩設定。

另請參閱

GetRealTorque, GetRobotPos

GetLimitTorque 範例

VB 例:

```
Dim j1LimitTorque As Integer
j1LimitTorque = m_spel.GetLimitTorque(1)
```

C# 例:

```
int j1LimitTorque;
j1LimitTorque = m_spel.GetLimitTorque(1);
```

GetLimZ 方法，Spel 類別

描述

傳回目前 LimZ 設定。

語法

Function **GetLimZ** () As Single

傳回值

包含 LimZ 值的實際值。

另請參閱

LimZ, Jump

GetLimZ 範例

VB 例:

```
saveLimZ = m_spel.GetLimZ()
m_spel.LimZ(-22)
```

C# 例:

```
saveLimZ = m_spel.GetLimZ();
m_spel.LimZ(-22);
```

GetPoint 方法，Spel 類別

描述

擷取某一機器人點的座標資料。

語法

Function **GetPoint** (*PointNumber* As Integer) As SpelPoint
Function **GetPoint** (*PointName* As String) As SpelPoint

參數

PointNumber　目前機器人的控制器點記憶體中某個點的整數運算式。

PointName　字串運算式。此可以是點標籤、「Pxxx」、「P*」或「*」。

另請參閱

SetPoint

GetPoint 範例

VB 例:

```
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 25.0
m_spel.Go(pt)
```

C# 例:

```
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 25.0;
m_spel.Go(pt);
```

GetRealTorque 方法，Spel 類別

描述

針對目前機器人的指定關節，傳回力矩。

語法

Function **GetRealTorque** (*JointNumber* As Integer) As Double

參數

JointNumber　所需之關節的整數運算式。

傳回值

介於 0 至 1 的雙精度值，代表目前功率模式及指定關節的最大力矩部分。

另請參閱

GetLimitTorque, GetRobotPos

GetRealTorque 範例

VB 例:

```
Dim j1Torque As Double
j1Torque = m_spel.GetRealTorque(1)
```

C# 例:

```
double j1Torque;
j1Torque = m_spel.GetRealTorque(1);
```

GetRobotInfo 方法，Spel 類別

描述

傳回機器人的資訊。

語法

Function **GetRobotInfo** (*RobotNumber* As Integer) As SpelRobotInfo

參數

RobotNumber　指定機器人的編號

傳回值

SpelRobotInfo

另請參閱

GetControllerInfo, GetTaskInfo

GetRobotInfo 範例

VB 例:

```
Dim info As SpelRobotInfo
Dim msg As String

info = m_spel.GetRobotInfo(1)
msg = "Robot Model: " & info.RobotModel & vbCrLf _
    & "Robot Serial: " & info.RobotSerial
MsgBox(msg)
```

C# 例:

```
SpelRobotInfo info;
string msg;

info = m_spel.GetRobotInfo(1);
msg = "Robot Model: " + info.RobotModel +
      "\r\n Robot Serial: " + info.RobotSerial;
MessageBox.Show(msg);
```

GetRobotPos 方法，Spel 類別

描述

傳回目前機器人位置。

語法

Function **GetRobotPos(** *PosType* As SpelRobotPosType, *Arm* As Integer, *Tool* As Integer, *Local* As Integer**) As Single()**

參數

PosType	指定要傳回之位置資料的類型。
Arm	指定機器人手臂的整數運算式。
Tool	指定機器人工具的整數運算式。
Local	指定機器人本地的整數運算式。

傳回值

包含 9 個元素的單一資料類型陣列。傳回的資料係依指定的 *PosType* 而定。

World	X, Y, Z, U, V, W, R, S, T
Joint	J1, J2, J3, J4, J5, J6, J7, J8, J9
Pulse	Pls1, Pls2, Pls3, Pls4, Pls5, Pls6, Pls7, Pls8, Pls9

另請參閱

GetPoint

GetRobotPos 範例

VB 例:

```
Dim values() As Single
values = m_spel.GetRobotPos(SpelRobotPosType.World, 0, 0, 0)
```

C# 例:

```
float[] values;
values = m_spel.GetRobotPos(SpelRobotPosType.World, 0, 0,
0);
```

GetSpeed 方法，Spel 類別

描述

傳回目前機器人的三個速度設定的其中之一。

語法

Function **GetSpeed** (*ParamNumber* As Integer) As Integer

參數

ParamNumber 評估下列任一數值的整數運算式。
1: PTP 動作速度
2: Jump 起始速度
3: Jump 結束速度

傳回值

介於 1-100 的整數運算式。

另請參閱

Speed

GetSpeed 範例

VB 例:

```
Dim x As Integer
x = m_spel.GetSpeed(1)
```

C# 例:

```
int ptpSpeed;
ptpSpeed = m_spel.GetSpeed(1);
```

GetTaskInfo 方法，Spel 類別

描述

傳回任務資訊。

語法

Function **GetTaskInfo** (*TaskName* As String) As SpelTaskInfo
Function **GetTaskInfo** (*TaskNumber* As Integer) As SpelTaskInfo

參數

TaskName	指定任務名稱
TaskNumber	指定任務編號

傳回值

SpelTaskInfo

另請參閱

GetControllerInfo, GetRobotInfo

GetTaskInfo 範例

VB 例:

```
Dim info As SpelTaskInfo
Dim msg As String

info = m_spel.GetTaskInfo(1)
msg = "Task Name: " & info.TaskName & vbCrLf _
    & "Task State: " & info.TaskState
MsgBox(msg)
```

C# 例:

```
SpelTaskInfo info;
string msg;

info = m_spel.GetTaskInfo(1);
msg= "Task Name: " + info.TaskName +
     "\r\n Task State: " + info.TaskState;
MessageBox.Show(msg);
```

GetTool 方法，Spel 類別

描述

傳回目前機器人的目前工具編號。

語法

Function **GetTool** () As Integer

傳回值

包含目前工具編號的整數。

另請參閱

Arm, TLSet, Tool

GetTool 範例

VB 例:

```
saveTool = m_spel.GetTool()
m_spel.Tool(2)
```

C# 例:

```
saveTool = m_spel.GetTool();
m_spel.Tool(2);
```

GetVar 方法，Spel 類別

描述

傳回控制器中 SPEL+全域保留變數的值。

語法

Function **GetVar**(*VarName* As String) As Object

參數

VarName	SPEL+全域保留變數的名稱。 對於陣列，可以傳回整個陣列或只傳回一個元素。

傳回值

傳回由 SPEL+變數的類型來判定資料類型的數值。

備註

您可使用 GetVar 在控制器的目前專案中擷取任何全域保留變數的數值。在擷取數值之前，必須成功建置專案。

若要擷取整個陣列，請在 *VarName* 中提供陣列名稱。若要擷取一個陣列的某個元素，請在 *VarName* 中提供子腳本。

另請參閱

SetVar

GetVar 範例

在 SPEL+專案中會宣告變數：

```
Global Preserve Integer g_myIntVar
Global Preserve Real g_myRealArray(10)
Global Preserve String g_myStringVar$
Function main
   ...
Fend
```

在 Visual Basic 專案中：

由於 g_myIntVar 宣告為整數，因此，用來擷取 g_myInVar 值的 Visual Basic 變數必須宣告為整數。對於 g_myRealArray，Visual Basic 變數必須宣告為 Single 變數的陣列。

```
Dim myIntVar As Integer
Dim myRealArray() As Single
Dim myStringVar As String

myIntVar = m_spel.GetVar("g_myIntVar")
myRealArray = m_spel.GetVar("g_myRealArray")
myStringVar = m_spel.GetVar("g_myStringVar$")
```

在 C# 專案中：

由於 g_myIntVar 宣告為整數，因此，用來擷取 g_myInVar 值的 C#變數必須宣告為整數。對於 g_myRealArray，C#變數必須宣告為 Float 變數的陣列。

```
int myIntVar;
float[] myRealArray;
string myStringVar;

myIntVar = m_spel.GetVar("g_myIntVar");
myRealArray = m_spel.GetVar("g_myRealArray");
myStringVar = m_spel.GetVar("g_myStringVar$");
```

描述

以 PTP 的形式將手臂從目前位置移至指定點或 XY 位置。**GO** 指令可同時移動任何組合的機器人軸。

語法

Sub **Go** (*PointNumber* As Integer)
Sub **Go** (*Point* As SpelPoint)
Sub **Go** (*Point* As SpelPoint, *AttribExpr* As String)
Sub **Go** (*PointExpr* As String)

參數

每個語法都有一個參數，用以指定在 Go 動作期間手臂移動的結束點。此為 PTP 動作結束時的最終位置。

PointNumber	透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
Point	透過使用 SpelPoint 資料類型來指定結束點。
AttribExpr	透過使用字串運算式來指定結束點屬性。
PointExpr	透過使用字串運算式來指定結束點。

另請參閱

Accel, Speed
Arc, Arc3, CVMove, Jump, Jump3, Jump3CP, Move
BGo, BMove, TGo, TMove
Arch, CP, Sense, Till

Go 範例

VB 例:

```
' 使用點編號指定的點
m_spel.Tool(1)
m_spel.Go(100)

' 使用 SpelPoint 指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.Go(pt)

' 使用運算式指定的點
m_spel.Go("P0 /L /2")
m_spel.Go("P1 :Z(-20)")

' 使用平行處理
m_spel.Go("P1 !D50; On 1; D90; Off 1!")

' 使用標籤指定的點
m_spel.Go("pick")
```

C# 例:

```
// 使用點編號指定的點
m_spel.Tool(1);
m_spel.Go(100);

// 使用 SpelPoint 指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.Go(pt);

// 使用運算式指定的點
m_spel.Go("P0 /L /2");
m_spel.Go("P1 :Z(-20)");

//使用平行處理
m_spel.Go("P1 !D50; On 1; D90; Off 1!");

//使用標籤指定的點
m_spel.Go("pick");
```

Halt 方法，Spel 類別

描述

暫停執行指定的任務。

語法

Sub **Halt** (*TaskNumber* As Integer)
Sub **Halt** (*TaskName* As String)

參數

TaskNumber	要暫停之任務的任務編號。 任務編號的範圍介於 1 至 32。
TaskName	要暫停之包含任務名稱的字串運算式。

另請參閱

Resume, Xqt

Halt 範例

VB 例:

```
m_spel.Halt(3)
```

C# 例:

```
m_spel.Halt(3);
```

Here 方法，Spel 類別

描述

在目前位置示教一個點。

語法

Sub **Here** (*PointNumber* As Integer)
Sub **Here** (*PointName* As String)

參數

PointNumber 目前機器人的點記憶體中某個點的整數運算式。可以使用 0 開始的任何有效點編號。

PointName 點標籤的字串運算式。

另請參閱

SetPoint

Here 範例

VB 例:

```
m_spel.Here("P20")
```

C# 例:

```
m_spel.Here("P20");
```

HideWindow 方法，Spel 類別

描述

隱藏先前與 ShowWindow 一起顯示的 EPSON RC+ 7.0 視窗。

語法

Sub **HideWindow** (WindowID As SpelWindows)

參數

WindowID　　要隱藏之 EPSON RC+ 7.0 視窗的 ID。

另請參閱

RunDialog, ShowWindow

HideWindow 範例

VB 例:

```
Sub btnHideIOMonitor_Click _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnHideIOMonitor.Click

    m_spel.HideWindow(RCAPINet.SpelWindows.IOMonitor)
End Sub
```

C# 例:

```
void btnHideIOMonitor_Click(object sender, EventArgs e)
{
   m_spel.HideWindow(RCAPINet.SpelWindows.IOMonitor);
}
```

Home 方法，Spel 類別

描述

將機器人手臂移至使用 HomeSet 方法設定的使用者定義起始點位置。

語法

Sub **Home** ()

另請參閱

HomeSet, MCal

Home 範例

VB 例:

```
With m_spel
    .MotorsOn = True
    .Home()
End With
```

C# 例:

```
m_spel.MotorsOn = true;
m_spel.Home();
```

Hofs 方法，Spel 類別

描述

讀取或設置從每個關節的編碼器原點到軟體原點的校正脈衝值。

語法

Function Hofs (JointNumber As Integer) As Integer

Sub Hofs (J1Pulses As Integer, J2Pulses As Integer, J3Pulses As Integer, J4Pulses As Integer, [J5Pulses As Integer], [J6Pulses As Integer], [J7Pulses As Integer], [J8Pulses As Integer] , [J9Pulses As Integer])

參數

J1Pulses – J9Pulses	代表第 1~9 關節的脈衝值的整數 *J5Pulses – J9Pulses* 可以省略

傳回值

傳回指定關節的校正脈衝值。

備註

本方法可以顯示或設定原點校正脈衝值。本方法可以設定編碼器原點(Z 相)到機械原點的偏位值。

機器人的動作控制基於安裝在每個關節上的編碼器原點，但是編碼器原點不一定與機器人機械原點匹配。因此，可以使用本方法設定校正脈衝值，使編碼器遠點和軟體原點匹配。

注意

除非必須，否則切勿更改 Hofs 值

Hofs 值在出廠時已經精確設定。不必要的更改此值可能會導致定位錯誤或機器人意外行爲。除非必須，否則切勿更改 Hofs 值。

自動計算 Hofs 值

要自動計算 Hofs 值，請將機械臂移動到要校準的位置，然後執行 Calib。這樣控制器就會根據 CalPls 脈衝值和校準位置脈衝值，自動計算 Hofs 值。

保存和復原 Hofs

可以在功能表中選擇[System Configuration]-[Robot]-[Calibration]，然後用保存或復原功能，保存或復原 Hofs。

另請參閱

CalPls, Home, Hordr, MCal

Hofs 範例

VB 例:

```
Dim val As Integer
val = m_spel.Hofs(1)
```

C# 例:

```
int val;
val = m_spel.Hofs(1);
```

HomeSet 方法，Spel 類別

描述

指定 Home 方法所使用的位置。

語法

Sub **HomeSet** (*J1Pulses* As Integer, *J2Pulses* As Integer, *J3Pulses* As Integer,
J4Pulses As Integer , *J5Pulses* As Integer, *J6Pulses* As Integer
[, *J7Pulses* As Integer] [, *J8Pulses* As Integer] [, *J9Pulses* As Integer])

參數

J1Pulses – *J9Pulses* 每個關節的 Home 位置編碼器脈衝值。
關節 7、8 及 9 為選用。

另請參閱

Home, MCal

HomeSet 範例

VB 例:

```
' 設定目前位置的起始點位置
With m_spel
    .HomeSet(.Pls(1), .Pls(2), .Pls(3), .Pls(4), 0, 0)
End With
```

C# 例:

```
// 設定目前位置的起始點位置
m_spel.HomeSet(m_spel.Pls(1), m_spel.Pls(2), m_spel.Pls(3),
          m_spel.Pls(4), 0 ,0);
```

Hordr 方法，Spel 類別

描述

指定傳回 HOME 位置的軸順序。

語法

Sub **Hordr** (*Home1* As Integer, *Home2* As Integer, *Home3* As Integer, *Home4* As Integer, *Home5* As Integer, *Home6* As Integer [, *Home7* As Integer] [, *Home8* As Integer] [, *Home9* As Integer])

參數

Home 1 - 9	通知哪個軸應在起始點程序的各步驟期間復歸原點的位元模式。0 至所有軸之間的任何軸數量都可在第一個步驟期間復歸原點。在指定 R、S 或 T 時可以指定 *Home 7 – 9*。

另請參閱

Home, HomeSet, Mcordr

Hordr 範例

VB 例:

```
m_spel.Hordr(2, 13, 0, 0, 0, 0)
```

C# 例:

```
m_spel.Hordr(2, 13, 0, 0, 0, 0);
```

Hour 方法，Spel 類別

描述

傳回累計系統運作時間(小時)。

語法

Function **Hour** () As Single

傳回值

表示時間的整數運算式。

Hour 範例

VB 例:

```
Dim hoursRunning As Single
hoursRunning = m_spel.Hour()
```

C# 例:

```
float hoursRunning;
hoursRunning = m_spel.Hour();
```

ImportPoints 方法，Spel 類別

描述

將點文件導入至目前機器人的目前專案。

語法

Sub **ImportPoints** (*SourcePath* As String, *ProjectFileName* As String [, *RobotNumber* As Integer])

參數

SourcePath	包含要導入目前專案之特定路徑與檔案的字串運算式。副檔名必須為.pts。
ProjectFileName	包含目前機器人或指定機器人(若有提供 *RobotNumber*)之目前專案中要導入的特定檔案。副檔名必須為.pts。
RobotNumber	選用。要使用點文件之機器人的整數運算式。指定 0 可讓其變為共同點文件。

另請參閱

SavePoints

ImportPoints 範例

VB 例:

```
With m_spel
    .ImportPoints("c:\mypoints\model1.pts", "robot1.pts")
End With
```

C# 例:

```
m_spel.ImportPoints(@"c:\mypoints\model1.pts",
"robot1.pts");
```

In 方法，Spel 類別

描述

傳回指定輸入埠的狀態。每個連接埠包含 8 個輸入位元(一位元組)。

語法

Function **In** (*PortNumber* As Integer) As Integer
Function **In** (*Label* As String) As Integer

參數

PortNumber	代表任一個輸入埠的整數運算式。 每個連接埠包含 8 個輸入位元(一位元組)。
Label	包含輸入位元組標籤的字串運算式。

傳回值

介於 0 至 255 的整數，代表輸入埠的狀態。

另請參閱

InBCD, Out, OpBCD, Sw

In 範例

VB 例:

```
Dim port1Value As Integer
port1Value = m_spel.In(1)
```

C# 例:

```
int port1Value;
port1Value = m_spel.In(1);
```

InBCD 方法，Spel 類別

描述

傳回 8 個輸入的輸入狀態(使用 BCD 格式)。(二進位十進碼)

語法

Function **InBCD** (*PortNumber* As Integer) As Integer
Function **InBCD** (*Label* As String) As Integer

參數

PortNumber	代表任一個輸入埠的整數運算式。
Label	包含輸入位元組標籤的字串運算式。

傳回值

介於 0 至 9 的整數，代表輸入埠的狀態。

另請參閱

In, Out, OpBCD, Sw

InBCD 範例

VB 例:

```
Dim port1Value As Integer
port1Value = m_spel.InBCD(1)
```

C# 例:

```
int port1Value;
port1Value = m_spel.InBCD(1);
```

Inertia 方法，Spel 類別

描述

指定目前機器人的裝載慣性和離心率。

語法

Sub **Inertia** (*LoadInertia* As Single, *Eccentricity* As Single)

參數

LoadInertia	實數運算式，用以指定夾具末端關節(包含夾具末端和工件)中心周圍的總慣性力矩(kgm^2)。
Eccentricity	實數運算式，用以指定夾具末端關節(包含夾具末端和工件)中心周圍的離心率(公釐)。

另請參閱

Weight

Inertia 範例

VB 例:

```
m_spel.Inertia(0.02, 1.0)
```

C# 例:

```
m_spel.Inertia(0.02, 1.0);
```

Initialize 方法，Spel 類別

描述

初始化 Spel 類別執行個體。

語法

Sub **Initialize** ()

備註

一般而言，Spel 類別執行個體會在第一個方法執行時自動初始化。當 EPSON RC+ 7.0 載入至記憶體時，初始化可能需要幾秒鐘的時間。在某些情況下，您可能會想在啟動期間先於應用程式中調用 initialize。

根據 ServerInstance，將 RC+ 作為伺服器進程啟動。每個 ServerInstance 對應於一個控制器和一個專案。如果使用 ServerInstance 屬性，則必須在執行初始化之前設置它。

另請參閱

Connect, Disconnect, ServerInstance

Initialize 範例

VB 例:

```
m_spel.Initiialize()
```

C# 例:

```
m_spel.Initialize();
```

InReal 方法，Spel 類別

描述

將 2 個字元(32 位元)的輸入資料，讀取為 32 位元浮點資料 (符合 IEEE754 標準)。

語法

Function **InReal** (*PortNumber* As Integer) As Single

參數

PortNumber　　代表輸入埠的整數

傳回值

返回輸入埠的狀態，在 32 位元浮點資料(符合 IEEE754 標準)中返回。

另請參閱

In, InBCD, InW, Out, OutW, OutReal

InReal 範例

VB 例:

```
Dim val As Single
val = m_spel.InReal(32)
```

C# 例:

```
float val;
val = m_spel.InReal(32);
```

InsideBox 方法，Spel 類別

描述

傳回接近檢查區域的檢查狀態。

語法

Function **InsideBox** (*BoxNumber* As Integer) As Boolean

參數

BoxNumber　　介於 1 至 15 的整數運算式，代表要傳回狀態的接近檢查區域。

傳回值

若機器人夾具末端位於指定工作空間內，會傳回 True，否則會傳回 False。

另請參閱

Box, InsidePlane

InsideBox 範例

VB 例:

```
Dim isInside As Boolean
isInside = m_spel.InsideBox(1)
```

C# 例:

```
bool isInside;
isInside = m_spel.InsideBox(1);
```

InsidePlane 方法，Spel 類別

描述

傳回接近檢查平面的檢查狀態。

語法

Function **InsidePlane** (*PlaneNumber* As Integer) As Boolean

參數

PlaneNumber　介於 1 至 15 的整數運算式，代表要傳回狀態的接近檢查平面。

傳回值

若機器人夾具末端位於指定工作空間內，會傳回 True，否則會傳回 False。

另請參閱

InsideBox, Plane

InsidePlane 範例

VB 例:

```
Dim isInside As Boolean
isInside = m_spel.InsidePlane(1)
```

C# 例:

```
bool isInside;
isInside = m_spel.InsidePlane(1);
```

InW 方法，Spel 類別

描述

傳回指定輸入字元埠的狀態。每個字元埠包含 16 個輸入位元。

語法

Function **InW** (*PortNumber* As Integer) As Integer
Function **InW** (*Label* As String) As Integer

參數

PortNumber	代表輸入埠的整數。
Label	包含輸入字元標籤的字串運算式。

傳回值

介於 0 至 65535 的整數值，代表輸入埠

另請參閱

In, InBCD, Out, OpBCD, Sw

InW 範例

VB 例:

```
Dim data As Integer
data = m_spel.InW(0)
```

C# 例:

```
int data;
data = m_spel.InW(0);
```

IsOptionActive 方法，Spel 類別

描述

傳回軟體選配件的狀態。

語法

Function **IsOptionActive** (*option* As SpelOptions) As Boolean

參數

option 代表選配件編號的整數。

傳回值

False – 無效

True – 有效

另請參閱

GetControllerInfo

IsOptionActive 範例

VB 例:

```
Dim ret As Boolean
ret = m_spel.IsOptionActive(SpelOptions.FieldbusMaster)
```

C# 例:

```
bool ret;
ret = m_spel.IsOptionActive(SpelOptions.FieldbusMaster);
```

JRange 方法，Spel 類別

描述

定義指定機器人關節的允許工作範圍(以脈衝為單位)。

語法

Sub **JRange** (*JointNumber* As Integer, *LowerLimitPulses* As Integer, *UpperLimitPulses* As Integer)

參數

JointNumber	介於 1 - 9 的整數，代表要指定之 JRange 的關節。
LowerLimitPulses	代表指定關節下限範圍之編碼器脈衝計數位置的整數。
UpperLimitPulses	代表指定關節上限範圍之編碼器脈衝計數位置的整數。

另請參閱

XYLim

JRange 範例

VB 例:

```
m_spel.JRange(1, -30000, 30000)
```

C# 例:

```
m_spel.JRange(1, -30000, 30000);
```

JS 方法，Spel 類別

描述

Jump Sense 會偵測手臂是否在完成 JUMP 指令之前停止(使用 SENSE 輸入)，或手臂是否完成 JUMP 移動。

語法

Function **JS** () As Boolean

傳回值

若動作期間偵測到 SENSE 輸入，會傳回 True，否則會傳回 False。

另請參閱

Jump, Jump3, Jump3CP, Sense, Till

JS 範例

VB 例:

```
With m_spel
    .Sense("Sw(1) = On")
    .Jump("P1 Sense")
    stoppedOnSense = .JS()
End With
```

C# 例:

```
m_spel.Sense("Sw(1) = On");
m_spel.Jump("P1 Sense");
stoppedOnSense = m_spel.JS();
```

JTran 方法，Spel 類別

描述

執行相對關節移動。

語法

Sub **JTran** (*JointNumber* As Integer, *Distance* As Single)

參數

JointNumber	要移動的特定關節。
Distance	要移動的距離。旋轉關節的單位為度，線性關節的單位為公釐。

另請參閱

PTran, Pulse

JTran 範例

VB 例:

```
'以正方向將關節 1 移動 45 度。
m_spel.JTran(1, 45.0)
```

C# 例:

```
//以正方向將關節 1 移動 45 度。
m_spel.JTran(1, 45.0);
```

Jump 方法，Spel 類別

描述

利用 PTP 動作將手臂從目前位置移至指定點(先垂直向上移動，接著水平移動，最後向下垂直移至最終目標點)。

語法

Sub **Jump** (*PointNumber* As Integer)
Sub **Jump** (*Point* As SpelPoint)
Sub **Jump** (*Point* As SpelPoint, *AttribExpr* As String)
Sub **Jump** (*PointExpr* As String)

參數

每個語法都有一個參數，用以指定在 Jump 動作期間手臂移動的結束點。此為 PTP 動作結束時的最終位置。

PointNumber	透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
Point	透過使用 SpelPoint 資料類型來指定結束點。
AttribExpr	透過使用字串運算式來指定結束點屬性。
PointExpr	透過使用字串運算式來指定結束點。

另請參閱

Accel, Speed
Arc, Arc3, CVMove, Go, Jump3, Jump3CP, Move
BGo, BMove, TGo, TMove
Arch, CP, Sense, Till

Jump 範例

VB 例:

```
' 使用點編號指定的點
m_spel.Tool(1)
m_spel.Jump(100)

' 使用 SpelPoint 指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.Jump(pt)

' 使用運算式指定的點
m_spel.Jump("P0 /L /2")
m_spel.Jump("P1 :Z(-20)")
m_spel.Jump("P1 C0")
m_spel.Jump("P1 C0 LimZ -10")
m_spel.Jump("P1 C0 Sense Sw(0)=On")

' 使用平行處理
m_spel.Jump("P1 !D50; On 1; D90; Off 1!")

' 使用標籤指定的點
m_spel.Jump("pick")
```

C# 例:

```
// 使用點編號指定的點
m_spel.Tool(1);
m_spel.Jump(100);

// 使用 SpelPoint 指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.Jump(pt);

// 使用運算式指定的點
m_spel.Jump("P0 /L /2");
m_spel.Jump("P1 :Z(-20)");
m_spel.Jump("P1 C0");
m_spel.Jump("P1 C0 LimZ -10");
m_spel.Jump("P1 C0 Sense Sw(0)=On");

// 使用平行處理
m_spel.Jump("P1 !D50; On 1; D90; Off 1!");

// 使用標籤指定的點
m_spel.Jump("pick");
```

描述

使用兩個 CP 動作與一個 PTP 動作之組合的 3D 閘道動作。機器人會移到起始點、接近點和目標點。

語法

Sub **Jump3** (*DepartPoint* As Integer, *ApproPoint* As Integer, *DestPoint* As Integer)
Sub **Jump3** (*DepartPoint* As SpelPoint, *ApproPoint* As SpelPoint, *DestPoint* As SpelPoint)
Sub **Jump3** (*DepartPoint* As String, *ApproPoint* As String, *DestPoint* As String)

參數

DepartPoint	目前位置之上的起始點(使用點編號或字串點運算式)。
ApproPoint	目標位置之上的結束點(使用點編號或字串點運算式)。
DestPoint	動作的目標位置(使用點編號或字串點運算式)。

另請參閱

Accel, AccelR, AccelS, Speed, SpeedR, SpeedS
Arc, Arc3, CVMove, Go, Jump, Jump3CP, Move
BGo, BMove, TGo, TMove
Arch, CP, Sense, Till

Jump3 範例

VB 例:

```
' 使用點編號指定的點
m_spel.Tool(1)
m_spel.Jump3(1, 2, 3)

' 使用 SpelPoint 指定的點
Dim pd As SpelPoint
Dim pa As SpelPoint
Dim pt As SpelPoint
pd = m_spel.GetPoint("P*")
pd.Z = 125.5
pa = m_spel.GetPoint("P2")
pa.Z = 125.5
pt = m_spel.GetPoint("P2")
m_spel.Jump3(pd, pa, pt)

' 使用運算式指定的點
m_spel.Jump3("P1", "P2", "P3 C0")
m_spel.Jump3("P1", "P2", "P3 C0 Sense Sw(0)=On")
m_spel.Jump3("P0 -TLZ(10), P1 -TLZ(10), P1")

' 使用平行處理
m_spel.Jump3("P1", "P2", "P3 !D50; On 1; D90; Off 1!")

' 使用標籤指定的點
m_spel.Jump3("depart", "approach", "place")
```

C# 例:

```
// 使用點編號指定的點
m_spel.Tool(1);
m_spel.Jump3(1, 2, 3);

// 使用 SpelPoint 指定的點
SpelPoint pd, pa, pt;
pd = m_spel.GetPoint("P1");
pd.Z = 125.5;
pa = m_spel.GetPoint("P2");
pa.Z = 125.5;
pt = m_spel.GetPoint("P2");
m_spel.Jump3(pd, pa, pt);

// 使用運算式指定的點
m_spel.Jump3("P1", "P2", "P3 C0");
m_spel.Jump3("P1", "P2", "P3 C0 Sense Sw(0)=On");
m_spel.Jump3("P0 -TLZ(10), P1 -TLZ(10), P1");

// 使用平行處理
m_spel.Jump3("P1", "P2", "P3 !D50; On 1; D90; Off 1!");

// 使用標籤指定的點
m_spel.Jump3("depart", "approach", "place");
```

Jump3CP 方法，Spel 類別

描述

使用三個 CP 動作之組合的 3D 閘道動作。

語法

Sub **Jump3CP** (*DepartPoint* As Integer, *ApproPoint* As Integer, *DestPoint* As Integer)
Sub **Jump3CP** (*DepartPoint* As SpelPoint, *ApproPoint* As SpelPoint, *DestPoint* As SpelPoint)
Sub **Jump3CP** (*DepartPoint* As String, *ApproPoint* As String, *DestPoint* As String)

參數

DepartPoint	目前位置之上的起始點(使用點編號或字串點運算式)。
ApproPoint	目標位置之上的結束點(使用點編號或字串點運算式)。
DestPoint	動作的目標位置(使用點編號或字串點運算式)。

另請參閱

AccelR, AccelS, SpeedR, SpeedS
Arc, Arc3, CVMove, Go, Jump, Jump3, Move
BGo, BMove, TGo, TMove
Arch, CP, Sense, Till

Jump3CP 範例

VB 例:

```
' 使用點編號指定的點
m_spel.Tool(1)
m_spel.Jump3CP(1, 2, 3)

' 使用 SpelPoint 指定的點
Dim pd As SpelPoint
Dim pa As SpelPoint
Dim pt As SpelPoint
pd = m_spel.GetPoint("P*")
pd.Z = 125.5
pa = m_spel.GetPoint("P2")
pa.Z = 125.5
pt = m_spel.GetPoint("P2")
m_spel. Jump3CP(pd, pa, pt)

' 使用運算式指定的點
m_spel.Jump3CP("P1", "P2", "P3 C0")
m_spel.Jump3CP ("P1", "P2", "P3 C0 Sense Sw(0)=On")
m_spel.Jump3CP("P0 -TLZ(10), P1 -TLZ(10), P1")

' 使用平行處理
m_spel.Jump3CP("P1", "P2", "P3 !D50; On 1; D90; Off 1!")

' 使用標籤指定的點
m_spel.Jump3CP("depart", "approch", "place")
```

C# 例:

```
// 使用點編號指定的點
m_spel.Tool(1);
m_spel.Jump3CP(1, 2, 3);

// 使用 SpelPoint 指定的點
SpelPoint pd, pa, pt;
pd = m_spel.GetPoint("P0");
pd.Z = 125.5;
pa = m_spel.GetPoint("P2");
pa.Z = 125.5;
pt = m_spel.GetPoint("P2");
m_spel. Jump3CP(pd, pa, pt);

// 使用運算式指定的點
m_spel.Jump3CP("P1", "P2", "P3 C0");
m_spel.Jump3CP ("P1", "P2", "P3 C0 Sense Sw(0)=On");
m_spel.Jump3CP("P0 -TLZ(10), P1 -TLZ(10), P1");

// 使用平行處理
m_spel.Jump3CP("P1", "P2", "P3 !D50; On 1; D90; Off 1!");

// 使用標籤指定的點
m_spel.Jump3CP("depart", "approch", "place");
```

LimitTorque 方法，Spel 類別

描述

針對目前機器人，設定高功率模式的上限力矩。

語法

Sub **LimitTorque** (*AllJointsMax* As Integer)
Sub **LimitTorque** (*J1Max* As Integer, *J2Max* As Integer, *J3Max* As Integer, *J4Max* As Integer, *J5Max* As Integer, *J6Max* As Integer)

參數

AllJointsMax 在高功率模式中，所有關節之所需力矩上限的整數運算式。

J1Max – *J6Max* 在高功率模式中，每個關節之所需力矩上限的整數運算式。

傳回值

介於 1 至 9 的整數值，代表指定關節的限制力矩設定。

另請參閱

GetRealTorque, GetRobotPos

LimitTorque 範例

VB 例:

```
Dim j1LimitTorque As Integer
j1LimitTorque = m_spel.LimitTorque(1)
```

C# 例:

```
int j1LimitTorquel
j1LimitTorque = m_spel.LimitTorque(1);
```

LimZ 方法，Spel 類別

描述

設定 JUMP 命令之 Z 軸高度的預設值。

語法

Sub **LimZ** (*ZLimit* As Single)

參數

ZLimit	Z 軸可移動範圍內的座標值。

另請參閱

Jump

LimZ 範例

VB 例:

```
saveLimZ = m_spel.GetLimZ()
m_spel.LimZ(-22)
```

C# 例:

```
saveLimZ = m_spel.GetLimZ();
m_spel.LimZ(-22);
```

LoadPoints 方法，Spel 類別

描述

將 SPEL+點文件載入至目前機器人的控制器點記憶體。

語法

Sub **LoadPoints** (*FileName* As String [, *Merge* As Boolean])

參數

FileName	目前專案中的有效點文件。
Merge	選用。設為將目前點整合至指定點檔案。

另請參閱

ImportPoints, SavePoints

LoadPoints 範例

VB 例:

```
With m_spel
    .LoadPoints("part1.pts")
End With
```

C# 例:

```
m_spel.LoadPoints("part1.pts");
```

Local 方法，Spel 類別

描述

定義本地座標系統。

語法

Sub **Local** (*LocalNumber* As Integer, *OriginPoint* As SpelPoint, [*XAxisPoint* As SpelPoint], [*YAxisPoint* As SpelPoint])
Sub Local (*LocalNumber* As Integer, *LocalPoint1* As Integer, *BasePoint1* As Integer, *LocalPoint2* As Integer, *BasePoint2* As Integer)

Sub Local (*LocalNumber* As Integer, *LocalPoint1* As String, *BasePoint1* As String, *LocalPoint2* As String, *BasePoint2* As String)

參數

LocalNumber	本地座標系統編號。總共可定義 15 個本地座標系統(1 至 15 的整數值)。
OriginPoint	本地座標系統之原點的 SpelPoint 變數。
XAxisPoint	選用。本地座標系統 X 軸上某一點的 SpelPoint 變數。
YAxisPoint	選用。本地座標系統 Y 軸上某一點的 SpelPoint 變數。
LocalPoint1, LocalPoint2	透過整數或字串指定，便是本地坐標系統的點數據。
BasePoint1,BasePoint2	透過整數或字串指定，便是 Base 坐標系統的點數據。

另請參閱

Base

Local 範例

VB 例:

```
Dim originPoint As New SpelPoint
originPoint.X = 100
originPoint.Y = 50
m_spel.Local(1, originPoint)
```

C# 例:

```
SpelPoint originPoint = new SpelPoint();
originPoint.X = 100;
originPoint.Y = 50;
m_spel.Local(1, originPoint);
```

LocalClr 方法，Spel 類別

描述

清除針對目前機器人定義的本地(Local)。

語法

Sub **LocalClr** (*LocalNumber* As Integer)

參數

LocalNumber　代表欲清除(取消定義)之本地(介於 1 至 15 的整數)的整數運算式。

另請參閱

Local, LocalDef

LocalClr 範例

VB 例:

```
m_spel.LocalClr(1)
```

C# 例:

```
m_spel.LocalClr(1);
```

LocalDef 方法，Spel 類別

描述

傳回本地定義狀態。

語法

Function **LocalDef** (*LocalNumber* As Integer) As Boolean

參數

LocalNumber　代表要傳回狀態之本地座標的整數運算式(1~15)。

傳回值

若指定本地已定義，會傳回 True，否則會傳回 False。

另請參閱

Local, LocalClr

LocalDef 範例

VB 例:

```
Dim localExists As Boolean
localExists = m_spel.LocalDef(1)
```

C# 例:

```
bool localExists;
localExists = m_spel.LocalDef(1);
```

Login 方法，Spel 類別

描述

以其他使用者身分登錄至 EPSON RC+ 7.0。

語法

Sub **Login** (*LoginID* As String, *Password* As String)

參數

LoginID	包含使用者登錄 ID 的字串運算式。
Password	包含使用者密碼的字串運算式。

備註

您可在應用程式中使用 EPSON RC+ 7.0 安全功能。例如，您可顯示一允許不同使用者登錄至系統的功能表。各類使用者都有專屬的安全權限。如需安全功能的詳細資訊，請參閱 *EPSON RC+ 7.0 使用指南*。

若已啟用安全功能且未執行 LogIn，則 Visual Basic 應用程式將會以訪客使用者身分登錄。若在 EPSON RC+ 7.0 中啟用自動登錄，應用程式將會以目前 Windows 使用者身分自動登錄(若已在 EPSON RC+ 7.0 中設定該使用者)。

另請參閱

GetCurrentUser

Login 範例

VB 例:

```
With m_spel
    .Project =
"c:\EpsonRC70\projects\myproject\myproject.sprj"
    .LogIn("operator", "oprpass")
End With
```

C# 例:

```
m_spel.Project =
@"c:\EpsonRC70\projects\myproject\myproject.sprj";
m_spel.LogIn("operator", "oprpass");
```

MCal 方法，Spel 類別

描述

針對搭載增量編碼器的機器人執行機器校準。

語法

Sub **MCal** ()

另請參閱

MCalComplete, MotorsOn

MCal 範例

VB 例:

```
If Not m_spel.MCalComplete() Then
    m_spel.MCal()
End If
```

C# 例:

```
if (!m_spel.MCalComplete())
    m_spel.MCal();
```

MCalComplete 方法，Spel 類別

描述

若 MCal 成功完成，會傳回 True。

語法

Function **MCalComplete** () As Boolean

傳回值

若 MCal 已經完成，會傳回 True，否則會傳回 False。

另請參閱

MCal

MCalComplete 範例

VB 例:

```
If m_spel.MCalComplete() Then
    lblStatus.Text = "MCal Complete"
Else
    lblStatus.Text = "MCal Not Complete"
End If
```

C# 例:

```
if (m_spel.MCalComplete())
    lblStatus.Text = "MCal Complete";
else
    lblStatus.Text = "MCal Not Complete";
```

Mcordr 方法，Spel 類別

描述

指定機器校準 MCal 的移動軸順序。

語法

Sub **MCordr** (*Step1* As Integer, *Step2* As Integer, *Step3* As Integer,
Step4 As Integer, *Step5* As Integer, *Step6*As Integer,
[*Step7* As Integer], [*Step8* As Integer], [*Step9* As Integer])

參數

Step 1 – 9 通知哪個軸應在 MCal 程序的各步驟期間復歸原點的位元模式。
0 至所有軸之間的軸或任何軸數量可在第一個步驟期間復歸原點。
Step 7 – 9 為選用，用於具有超過 7 個軸的機器人。

另請參閱

Home, HomeSet, Hordr, MCal

Mcordr 範例

VB 例:

```
m_spel.Mcordr(2, 13, 0, 0, 0, 0)
```

C# 例:

```
m_spel.Mcordr(2, 13, 0, 0, 0, 0);
```

MemIn 方法，Spel 類別

描述

傳回指定記憶體 I/O 位元組埠的狀態。每個連接埠包含 8 個記憶體 I/O 位元。

語法

Function **MemIn** (*PortNumber* As Integer) As Integer
Function **MemIn** (*Label* As String) As Integer

參數

PortNumber	代表任一個記憶體 I/O 埠的整數運算式。
Label	包含記憶體 I/O 位元組標籤的字串運算式。

傳回值

包含連接埠值的整數。

另請參閱

In, InBCD, MemOut, MemSw, Sw, Off, On, Oport

MemIn 範例

VB 例:

```
data = m_spel.MemIn(1)
```

C# 例:

```
data = m_spel.MemIn(1);
```

MemInW 方法，Spel 類別

描述

傳回指定記憶體 I/O 字元埠的狀態。每個字元埠包含 16 個記憶體 I/O 位元。

語法

Function **MemInW** (*PortNumber* As Integer) As Integer
Function **MemInW** (*Label* As String) As Integer

參數

PortNumber	代表記憶體 I/O 字元的整數運算式。
Label	包含記憶體 I/O 字元標籤的字串運算式。

傳回值

介於 0 至 65535 的整數運算式，代表輸入埠狀態。

另請參閱

In, InBCD, MemIn, MemSw, Sw, Off, On, Oport

MemInW 範例

VB 例:

```
data = m_spel.MemInW(1)
```

C# 例:

```
data = m_spel.MemInW(1);
```

MemOff 方法，Spel 類別

描述

關閉 S/W 記憶體 I/O 的指定位元。

語法

Sub **MemOff** (*BitNumber* As Integer)
Sub **MemOff** (*Label* As String)

參數

BitNumber	代表任一個記憶體 I/O 位元的整數運算式。
Label	包含記憶體 I/O 位元標籤的字串運算式。

另請參閱

In, InBCD, MemOut, MemSw, Sw, Off, On, Oport

MemOff 範例

VB 例:

```
m_spel.MemOff(500)
```

C# 例:

```
m_spel.MemOff(500);
```

MemOn 方法，Spel 類別

描述

開啟記憶體 I/O 的指定位元。

語法

Sub **MemOn** (*BitNumber* As Integer)
Sub **MemOn** (*Label* As String)

參數

BitNumber	代表任一個記憶體 I/O 位元的整數運算式。
Label	包含記憶體 I/O 位元標籤的字串運算式。

另請參閱

In, InBCD, MemOut, MemSw, Sw, Off, On, Oport

MemOn 範例

VB 例:

```
m_spel.MemOn(500)
```

C# 例:

```
m_spel.MemOn(500);
```

MemOut 方法，Spel 類別

描述

根據使用者指定的 8 位元值，同時設定 8 個記憶體 I/O 位元。

語法

Sub **MemOut** (*PortNumber* As Integer, *Value* As Integer)
Sub **MemOut** (*Label* As String, *Value* As Integer)

參數

PortNumber	代表任一個記憶體 I/O 位元組的整數運算式。
Label	包含記憶體 I/O 位元組標籤的字串運算式。
Value	包含指定位元組之輸出模式的整數運算式。有效值介於 0 - 255。

另請參閱

In, InBCD, MemIn, MemSw, Sw, Off, On, Oport

MemOut 範例

VB 例:

```
m_spel.MemOut(2, 25)
```

C# 例:

```
m_spel.MemOut(2, 25);
```

MemOutW 方法，Spel 類別

描述

根據使用者指定的 16 位元值，同時設定 16 個記憶體 I/O 位元。

語法

Sub **MemOutW** (*PortNumber* As Integer, *Value* As Integer)
Sub **MemOutW** (*Label* As String, *Value* As Integer)

參數

PortNumber	代表任一個記憶體 I/O 字元的整數運算式。
Label	包含記憶體 I/O 字元標籤的字串運算式。
Value	使用運算式或數值指定輸出資料(介於 0 至 65535 的整數)。

另請參閱

In, InBCD, MemIn, MemSw, Sw, Off, On, Oport

MemOutW 範例

VB 例:

```
m_spel.MemOutW(2, 25)
```

C# 例:

```
m_spel.MemOutW(2, 25);
```

MemSw 方法，Spel 類別

描述

傳回指定的記憶體 I/O 位元狀態。

語法

Function **MemSw** (*BitNumber* As Integer) As Boolean
Function **MemSw** (*Label* As String) As Boolean

參數

BitNumber	代表任一個記憶體 I/O 位元的整數運算式。
Label	包含記憶體 I/O 位元標籤的字串運算式。

傳回值

若指定記憶體 I/O 位元開啟，會傳回 True，否則會傳回 False。

另請參閱

In, InBCD, MemIn, Sw, Off, On, Oport

MemSw 範例

VB 例:

```
If m_spel.MemSw(10) Then
    m_spel.On(2)
End If
```

C# 例:

```
if (m_spel.MemSw(10))
   m_spel.On(2);
```

Move 方法，Spel 類別

描述

使用 CP 動作(直綫移動)，將手臂從目前位置移至指定點。

語法

Sub **Move** (*PointNumber* As Integer)
Sub **Move** (*Point* As SpelPoint)
Sub **Move** (*Point* As SpelPoint, *AttribExpr* As String)
Sub **Move** (*PointExpr* As String)

參數

每個語法都有一個參數，用以指定在 Move 動作期間手臂移動的結束點。此為線性插補動作結束時的最終位置。

PointNumber	透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
Point	透過使用 SpelPoint 資料類型來指定結束點。
AttribExpr	透過使用字串運算式來指定結束點屬性。
PointExpr	透過使用字串運算式來指定結束點。

另請參閱

AccelR, AccelS, SpeedR, SpeedS
Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP
BGo, BMove, TGo, TMove
Arch, CP, Till

Move 範例

VB 例:

```
' 使用點編號指定的點
m_spel.Tool(1)
m_spel.Move(100)

' 使用 SpelPoint 指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.Move(pt)

' 使用運算式指定的點
m_spel.Move("P0 /L /2 ROT")
m_spel.Move("P1 :Z(-20)")

' 使用平行處理
m_spel.Move("P1 !D50; On 1; D90; Off 1!")

' 使用標籤指定的點
m_spel.Move("pick")
```

C# 例:

```
// 使用點編號指定的點
m_spel.Tool(1);
m_spel.Move(100);

// 使用 SpelPoint 指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.Move(pt);

// 使用運算式指定的點
m_spel.Move("P0 /L /2 ROT");
m_spel.Move("P1 :Z(-20)");

// 使用平行處理
m_spel.Move("P1 !D50; On 1; D90; Off 1!");

// 使用標籤指定的點
m_spel.Move("pick");
```

Off 方法，Spel 類別

描述

關閉指定輸出。

語法

Sub **Off** (*BitNumber* As Integer)
Sub **Off** (*Label* As String)

參數

BitNumber 代表任一個標準或擴展輸出的整數運算式。此會通知 **Off** 指令要關閉哪個輸出。

Label 包含輸出位元標籤的字串運算式。

另請參閱

On, Oport, Out, OutW

Off 範例

VB 例:

```
m_spel.Off(1)
```

C# 例

```
m_spel.Off(1);
```

OLRate 方法，Spel 類別

描述

傳回指定關節的過載率。

語法

Function **OLRate** (*JointNumber* As Integer) As Single

參數

JointNumber	代表關節編號的整數值 (範圍: 1~ 機器人的關節數量)

傳回值

傳回指定關節的過載率。傳回值範圍是 0.0~2.0。

備註

OLRate 會檢查循環是否造成關節過載。使用高負載的循環應用，會由於溫度和電流導致伺服錯誤。OLRate 會檢查機器人是否容易出現伺服錯誤。

執行循環時，可以執行另一個任務來監視 OLRate。如果有關節的 OLRate 超過 1.0，則會發生伺服錯誤。

當過載時，最有可能發生伺服錯誤。可以在測試循環時，使用 OLRate 來確認速度和加減速度，以確保實際使用過程中不會發生伺服錯誤。

爲獲得有效值，請在機器人操作期間執行本方法。

正常負載狀態下，不能使用本方法。

OLRate 範例

VB 例:

```
Dim data As Single
data = m_spel.OLRate(1)
```

C# 例:

```
float data;
data = m_spel.OLRate(1);
```

On 方法，Spel 類別

描述

開啟指定輸出。

語法

Sub **On** (*BitNumber* As Integer)
Sub **On** (*Label* As String)

參數

BitNumber　代表任一個標準或擴展輸出的整數運算式。

Label　包含輸出位元標籤的字串運算式。

另請參閱

Off, Oport, Out, OutW

On 範例

VB 例:

```
m_spel.On(1)
```

C# 例:

```
m_spel.On(1);
```

OpBCD 方法，Spel 類別

描述

使用 BCD(二進位十進碼)格式，同時設定 8 個輸出位元。

語法

OpBCD (*PortNumber* As Integer, *Value* As Integer)
OpBCD (*Label* As String, *Value* As Integer)

參數

PortNumber	代表任一個連接埠的整數。 每個連接埠包含 8 個輸出位元(一位元組)。
Value	介於 0-99 的整數，代表指定連接埠的輸出模式。 第二個數字(稱為個位數)代表連接埠的下方 4 個輸出，第一個數字(稱為十位數)代表連接埠的上方 4 個輸出。

另請參閱

Off, Out, Sw

OpBCD 範例

VB 例:

```
m_spel.OpBCD(1, 25)
```

C# 例:

```
m_spel.OpBCD(1, 25);
```

Oport 方法，Spel 類別

描述

傳回指定輸出位元的狀態。

語法

Function **Oport** (*BitNumber* As Integer) As Boolean
Function **Oport** (*Label* As String) As Boolean

參數

BitNumber	代表任一個標準或擴展輸出的整數運算式。
Label	包含輸出位元組標籤的字串運算式。

傳回值

若指定輸出位元已開啟，會傳回 True，否則會傳回 False。

另請參閱

Off, On, OpBCD, Out, Sw

Oport 範例

VB 例:

```
If m_spel.Oport(1) Then
    m_spel.On(2)
End If
```

C# 例:

```
if(m_spel.Oport(1))
    m_spel.On(2);
```

Out 方法，Spel 類別

描述

同時讀取或設定 8 個輸出位元(一個位元組)。

語法

Sub **Out** (*PortNumber* As Integer, *Value* As Integer)
Sub **Out** (*Label* As String, *Value* As Integer)
Function **Out** (*PortNumber* As Integer) As Integer
Function **Out** (*Label* As String) As Integer

參數

PortNumber	代表任一個輸出埠的整數。
Label	包含輸出位元組標籤的字串運算式。
Value	介於 0-255 的整數，代表輸出埠的輸出模式。若以十六進位表示，範圍介於&H0 至&HFF。

傳回值

介於 0-255 的整數(包含連接埠值)。

另請參閱

InBCD, OpBCD, Oport, OutW, Sw

Out 範例

VB 例:

```
m_spel.Out(1, 240)
```

C# 例:

```
m_spel.Out(1, 240);
```

OutReal 方法，Spel 類別

描述

讀取或設定輸出埠狀態的 32 位元浮點資料 (符合 IEEE754 標準)。

語法

Function OutReal (*WordPortNumber* As Integer) As Single
Sub OutReal (*WordPortNumber* As Integer, *Value* As Single)

參數

WordPortNumber	代表輸出埠的整數
Value	代表輸出資料的實數值

傳回值

返回輸出埠的狀態，在 32 位元浮點資料(符合 IEEE754 標準)中返回。

另請參閱

In, InBCD, InReaql, InW, Out, OutW

OutReal 範例

VB 例:

```
Dim val As Single
val = m_spel.OutReal(32)
```

C# 例:

```
float val;
val = m_spel.OutReal(32);
```

OutW 方法，Spel 類別

描述

同時讀取或設定 16 個輸出位元(一個字元)。

語法

Sub **OutW** (*PortNumber* As Integer, *Value* As Integer)
Sub **OutW** (*Label* As String, *Value* As Integer)
Function **OutW** (*PortNumber* As Integer) As Integer
Function **OutW** (*Label* As String) As Integer

參數

PortNumber	代表任一個輸出埠的整數。
Label	包含輸出字元標籤的字串運算式。
Value	介於 0-65535 的整數，代表輸出埠的輸出模式。若以十六進位表示，範圍介於&H0 至&HFFFF。

傳回值

介於 0-65535 的整數(包含連接埠值)。

另請參閱

InBCD, OpBCD, Oport, Out, Sw

OutW 範例

VB 例:

```
m_spel.OutW(1, 240)
```

C# 例:

```
m_spel.OutW(1, 240);
```

PAgl 方法，Spel 類別

描述

傳回所選旋轉軸的關節角度，或所選線性軸的位置(來自指定點)。

語法

Function **PAgl** (*PointNumber* As Integer, *JointNumber* As Integer) As Single
Function **PAgl** (*Point* As SpelPoint, *JointNumber* As Integer) As Single
Function **PAgl** (*Label* As String, *JointNumber* As Integer) As Single

參數

PointNumber	代表目前機器人點記憶體中某個點之點編號的整數運算式。
Point	先前初始化的 SpelPoint。
Label	包含目前機器人點記憶體中某個點之點標籤的字串運算式。
JointNumber	代表所需之關節編號的整數運算式。 數值可以介於 1 ~ 9。

傳回值

包含指定關節之角度的單一數值(以度或公釐為單位)。

另請參閱

Agl, Pls, CX – CT

PAgl 範例

VB 例:

```
Dim t1Angle As Single
t1Angle = m_spel.PAgl(1, 1)
```

C# 例:

```
float t1Angle;
t1Angle = m_spel.PAgl(1, 1);
```

Pallet 方法，Spel 類別

描述

定義棧板。

語法

Sub **Pallet** (*PalletNumber* As Integer，*Point1* As String, *Point2* As String, *Point3* As String
[, *Point4* As String] , *rows* As Integer, *columns* As Integer)

參數

PalletNumber	以 0 至 15 的整數表示的棧板編號。
Point1	定義第一個棧板位置的點變數。
Point2	定義第二個棧板位置的點變數。
Point3	定義第三個棧板位置的點變數。
Point4	選用。定義第四個棧板位置的點變數。
Rows	棧板橫向的點數目。每個數值均為介於 1 至 32767 的整數。
Columns	棧板縱向的點數目。每個數值均為介於 1 至 32767 的整數。

另請參閱

Jump, Go, SetPoint

Pallet 範例

VB 例:

```
m_spel.Pallet(1, 1, 2, 3, 4, 3, 4)
```

C# 例:

```
m_spel.Pallet(1, 1, 2, 3, 4, 3, 4);
```

Pass 方法，Spel 類別

描述

指定通過指定點附近而不停止動作的 PTP 動作。

語法

Sub **Pass**(PointNumber As Integer)
Sub **Pass**(PassExpr As String)

參數

PointNumber 使用示教點從控制器中儲存之目前機器人的點記憶體中指定一個點。

PassExpr 使用字串運算式來指定點。

Point specification [, {On | Off | MemOn | MemOff} bit number [,point specification ...]] [LJM [Orientation flag]]

Point specification	指定點編號、P(運算式)或點標籤。如果點資料完整並依遞增或遞減順序列出，可使用一個冒號合併兩個點編號，例如 P(1:5)。
Bit number	使用整數或輸出標籤指定 I/O 輸出位或記憶體 I/O 位元進行開啟／關閉。
LJM	選用。使用 LJM 函數轉換起始座標、接近座標及目標座標。
Orientation flag	選用。指定 LJM 函數的方向旗標參數。

另請參閱

Accel, Go, Jump, Speed

Pass 範例

VB 例:

```
m_spel.Jump(1)
m_spel.Pass(2) '將手臂#2 移至更靠近 P2，並在到達 P2 之前執行以下命令
m_spel.On(2)
m_spel.Pass(3)
m_spel.Pass(4)
m_spel.Off(0)
m_spel.Pass(5)
```

C# 例:

```
m_spel.Jump(1);
m_spel.Pass(2); //將手臂#2 移至更靠近 P2，並在到達 P2 之前執行以下命令
m_spel.On(2);
m_spel.Pass(3);
m_spel.Pass(4);
m_spel.Off(0);
m_spel.Pass(5);
```

Pause 方法，Spel 類別

描述

使控制器中的所有正常 SPEL+任務暫停。若機器人正在移動，將會立即減速至停止。

語法

Sub **Pause** ()

另請參閱

Continue, EventReceived, Stop

Pause 範例

VB 例:

```
Sub btnPause_Click()_
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnPause.Click

    m_spel.Pause()
    btnPause.Enabled = False
    btnContinue.Enabled = True
End Sub
```

C# 例:

```
void btnPause_Click(object sender, EventArgs e)
{
    m_spel.Pause();
    btnPause.Enabled = false;
    btnContinue.Enabled = true;
}
```

PDef 方法，Spel 類別

描述

傳回指定點的定義狀態。

語法

Function **PDef** (*PointNumber* As Integer) As Boolean

參數

PointNumber　目前機器人點記憶體中某個點之點編號的整數運算式。

傳回值

若指定點已定義，會傳回 True，否則會傳回 False。

另請參閱

PDel

PDef 範例

VB 例:

```
Dim p1Defined As Boolean
p1Defined = m_spel.PDef(1)
```

C# 例:

```
bool p1Defined;
p1Defined = m_spel.PDef(1);
```

PDel 方法，Spel 類別

描述

刪除指定位置資料。

語法

Sub **PDel** (*FirstPointNumber* As Integer [, *LastPointNumber* As Integer])

參數

FirstPointNumber	指定範圍內欲刪除之第一個點的整數運算式。
LastPointNumber	選用。指定範圍內欲刪除之最後一個點的整數運算式。如果省略，則只會刪除在 *FirstPointNumber* 中指定的點。

另請參閱

PDef, LoadPoints, Clear, SavePoints

PDel 範例

VB 例:

```
m_spel.PDel(1, 10)
m_spel.SavePoints("model1.pts")
```

C# 例:

```
m_spel.PDel(1, 10);
m_spel.SavePoints("model1.pts");
```

PeakSpeed 方法，Spel 類別

描述

傳回指定關節的峰值速度。

語法

Function **PeakSpeed** (*JointNumber* As Integer) As Single

參數

JointNumber	代表關節編號的整數值 (範圍: 1~ 機器人的關節數量)

傳回值

傳回-1~1 範圍中的實數值。

備註

本方法傳回一個帶符號的值，該值是關節的最大速度的絕對值。最大速度為 1。峰值速度是-1~1 範圍中的實數值。

請先執行 PeakSpeedClear，然後再顯示本方法來顯示關節的峰值速度。

如果是虛擬控制器或者乾運轉，就使用命令速度而不是實際速度來計算。

本方法不支援 PG 附加軸。

另請參閱

AvgSpeed, AvgSpeedClear, PeakSpeedClear

PeakSpeed 範例

VB 例:

```
Dim val As Single
val = m_spel.PeakSpeed(1)
```

C# 例:

```
float val;
val = m_spel.PeakSpeed(1);
```

PeakSpeedClear 方法，Spel 類別

描述

清除並初值化關節的峰值速度。

語法

Sub **PeakSpeedClear** ()

備註

使用本方法可以清除指定關節的峰值速度值。

執行 PeakSpeed 方法之前，必須先執行本方法。

本方法不支援 PG 附加軸。

另請參閱

AvgSpeed, AvgSpeedClear, PeakSpeed

PeakSpeedClear 範例

VB 例:

```
m_spel.PeakSpeedClear()
```

C# 例:

```
m_spel.PeakSpeedClear();
```

PF_Abort 方法，Spel 類別

描述

强制中斷指定零件的 Part Feeding 程序動作。

語法

Sub **PF_Abort** (*PartID* As Integer)

參數

PartID	代表零件 ID 的整數值 (1~16)

備註

立即中斷指定零件的 Part Feeding 程序。

與 PF_Stop 方法相異，本方法會中斷正在執行的回呼函式。

如果 Part Feeding 程序還沒有開始，就不會發生任何動作。

PF_Abort 範例

VB 例:

```
m_spel.PF_Abort(1)
```

C# 例:

```
m_spel.PF_Abort(1);
```

PF_Backlight 方法，Spel 類別

描述

打開或關閉給料器的嵌入式背光燈。

語法

Sub **PF_Backlight** (*FeederNumber* As Integer, *State* As Boolean)

參數

FeederNumber	代表給料器編號的整數值
State	指定 On (True) 或 Off (False)

備註

如果系統自動運行視覺處理時，背光燈會自動 On/Off。

使用 PF_Vision 回呼函式，就可以使用本方法打開或關閉背光燈。

PF_Backlight 範例

VB 例:

```
m_spel.PF_Backlight(1, True)
```

C# 例:

```
m_spel.PF_Backlight(1, true);
```

PF_BacklightBrightness 方法，Spel 類別

描述

設定給料器嵌入式背光燈的亮度。

語法

Sub **PF_Backlightness** (*FeederNumber* As Integer, *Brightness* As Integer)

參數

FeederNumber	代表給料器編號的整數值。
SBrightness	用 0%~100%的數字指定亮度。

備註

一般情況下，可以使用[Part Feeding Configuration]對話方塊設定嵌入式背光燈的亮度。當要在運行中更改亮度時，就可以使用本方法。

PF_BacklightBrightness 範例

VB 例:

```
m_spel.PFBacklightness(1, 80)
```

C# 例:

```
m_spel.PF_BacklightBrightness(1, 80);
```

PF_Name 方法，Spel 類別

描述

從零件 ID 取得零件名稱。

語法

Function **PF_Name** (*PartID* As Integer) As String

參數

PartID	代表零件 ID 的整數值(1~16)

傳回值

傳回指定零件 ID 的名稱的字串。

備註

如果指定的零件 ID 無效，就傳回 “ ” (空白字符)。

PF_Name 範例

VB 例:

```
Dim part1Name As String
Part1Name = m_spel.PF_Name(1)
```

C# 例:

```
string part1Name;
part1Name = m_spel.PF_Name(1);
```

PF_Number 方法，Spel 類別

描述

從零件名稱取得零件 ID。

語法

Function **PF_Number** (*PartName* As String) As Integer

參數

PartName	代表零件名稱的字串

傳回值

傳回指定零件名稱的零件 ID (整数 1~16)。

備註

如果零件名稱不存在就傳回–1。

如果出現重複名稱的零件，就返回 ID 最小的零件。

PF_Number 範例

VB 例:

```
Dim part1ID As Integer
Part1ID = m_spel.PF_Number("Part1")
```

C# 例:

```
int part1ID;
part1ID = m_spel.PF_Number("Part1");
```

PF_Start 方法，Spel 類別

描述

開始指定零件的 Part Feeding 程序。

語法

Sub PF_Start (PartID1 As Integer, [PartID2 As Integer], [PartID3 As Integer], [PartID4 As Integer])

參數

PartID1	代表主零件 ID 的整數值 (1~16)。
PartID2	代表從零件 ID 的整數值 (1~16)。可以省略
PartID3	代表從零件 ID 的整數值 (1~16)。可以省略
PartID4	代表從零件 ID 的整數值 (1~16)。可以省略

備註

在開始本方法之前，請先進行以下操作。

- 選擇要使用的機器人
- 馬達 On
- 如果要生成日誌，請執行 PF_InitLog

執行本方法，會產生一個新的任務，并將控制回復給主叫程式。

發生這種情況時，Status 回呼函式會在以下條件中執行。
Part Feeding 程序不會開始。

條件	Status 回呼函式參數的 Status 值
零件 ID 無效	PF_STATUS_BAD_ID
零件參數設定無效 (沒有選擇 Enabled 選項)	PF_STATUS_BAD_PARAMETER
沒有完成校準	PF_STATUS_CAL_NOT_COMPLETE
發生錯誤	PF_STATUS_ERROR

本方法不能同時複數執行。如果執行了，就會繼續已經執行的處理。不會發生錯誤。

請在正常任務中執行 PF_Start 方法。如果在後台執行則會發生錯誤。

注意
如果在本方法中指定了不存在的零件 ID，就會發生 7600 錯誤。

PF_Start 範例

VB 例:

```
m_spel.PF_Start(1)
```

C# 例:

```
m_spel.PF_Start(1);
```

PF_Stop 方法，Spel 類別

描述

發佈 Part Feeding 程序終止的要求。
如果有正在執行的回呼函式，會等待執行完成。
然後執行 PF_CycleStop 回呼函式，停止程序。

語法

Sub **PF_Stop** (*PartID* As Integer)

參數

PartID	代表零件 ID 的整數值(1~16)

備註

停止 Part Feeding 程序。
與 PFAbort 方法相異，本方法會等待執行中的回呼函式完成。
回呼函式執行完成后，在執行 PF_CycleStop 回呼函式。
如果沒有開始任何 Part Feeding 程序，就不會發生任何動作。

PF_Stop 範例

VB 例:

```
m_spel.PF_Stop(1)
```

C# 例:

```
m_spel.PF_Stop(1);
```

PLabel 方法，Spel 類別

描述

取得或設定指定點編號中定義的點標籤。

語法

Function **PLabel** (*PointNumber* As Integer) As String
Sub **PLabel** (*PointNumber* As Integer, *PointName* As String)

參數

PointNumber	代表點編號的整數值
PointName	指定的點資料中使用的標籤的字串

傳回值

傳回指定點編號相對應的標籤。

另請參閱

PDef

PLabel 範例

VB 例:

```
Dim pt1Label As String
Pt1Label = m_spel.PLabel(1)
```

C# 例:

```
string pt1Label;
pt1Label = m_spel.PLabel(1);
```

Plane 方法，Spel 類別

描述

定義平面。

語法

Sub **Plane** (*PlaneNumber* As Integer, *Point* As SpelPoint)
Sub **Plane** (*PlaneNumber* As Integer, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single, *V* As Single, *W* As Single)

參數

PlaneNumber	介於 1-15 的整數，代表 15 個平面中要定義的平面。
Point	代表接近檢查平面之座標資料的點資料。
X	代表接近檢查平面之座標資料的點 X 座標。
Y	代表接近檢查平面之座標資料的點 Y 座標。
Z	代表接近檢查平面之座標資料的點 Z 座標。
U	代表接近檢查平面之座標資料的點 U 座標。
V	代表接近檢查平面之座標資料的點 V 座標。
W	代表接近檢查平面之座標資料的點 W 座標。

另請參閱

PlaneClr, PlaneDef

Plane 範例

VB 例:

```
m_spel.Plane(1, -5, 5, -10, 10, -20, 20)
```

C# 例:

```
m_spel.Plane(1, -5, 5, -10, 10, -20, 20);
```

PlaneClr 方法，Spel 類別

描述

清除(取消定義)平面。

語法

Sub **PlaneClr** (*PlaneNumber* As Integer)

參數

PlaneNumber　介於 1-15 的整數，代表 15 個平面中要清除的平面。

另請參閱

Plane, PlaneDef

PlaneClr 範例

VB 例:

```
m_spel.PlaneClr(1)
```

C# 例:

```
m_spel.PlaneClr(1);
```

描述

傳回平面是否定義。

語法

Function **PlaneDef** (*PlaneNumber* As Integer) As Boolean

參數

PlaneNumber　介於 1 至 15 的整數運算式，代表平面編號。

傳回值

若指定平面已定義，會傳回 True，否則會傳回 False。

另請參閱

Plane, PlaneClr

PlaneDef 範例

VB 例:

```
x = m_spel.PlaneDef(1)
```

C# 例:

```
x = m_spel.PlaneDef(1);
```

Pls 方法，Spel 類別

描述

傳回目前位置各軸的目前編碼器脈衝計數。

語法

Function **Pls** (*JointNumber* As Integer) As Integer

參數

JointNumber　要取得目前編碼器脈衝計數的特定軸。(1 至 9)

傳回值

包含指定關節之目前脈衝計數的整數。

另請參閱

Agl, Pulse

Pls 範例

VB 例:

```
j1Pulses = m_spel.Pls(1)
```

C# 例:

```
j1Pulses = m_spel.Pls(1);
```

PTCLR 方法，Spel 類別

描述

清除並初值化峰值轉矩。

語法

Sub **PTCLR** ()

備註

本方法可清除指定關節的峰值轉矩。

執行 PTRQ 方法之前，必須先執行本方法。

另請參閱

ATCLR, ATRQ, PTRQ

PTCLR 範例

VB 例:

```
m_spel.PTCLR ()
```

C# 例:

```
m_spel.PTCLR();
```

PTPBoost 方法，Spel 類別

描述

設定短距離 PTP(點至點)動作的提升參數。

語法

Sub **PTPBoost** (*BoostValue* As Integer [, *DepartBoost* As Integer] [, *ApproBoost* As Integer])

參數

BoostValue	介於 0 - 100 的整數運算式。
DepartBoost	選用。Jump 起始提升值。介於 0 - 100 的整數運算式。
ApproBoost	選用。Jump 結束提升值。介於 0 - 100 的整數運算式。

另請參閱

PTPBoostOK

PTPBoost 範例

VB 例:

```
m_spel.PTPBoost(50)
m_spel.PTPBoost(50, 30, 30)
```

C# 例:

```
m_spel.PTPBoost(50);
m_spel.PTPBoost(50, 30, 30);
```

PTPBoostOK 方法，Spel 類別

描述

傳回從目前位置到目標位置的 PTP(點至點)動作是否為短距離。

語法

Function **PTPBoostOK** (*PointNumber* As Integer) As Boolean
Function **PTPBoostOK** (*Point* As SpelPoint) As Boolean
Function **PTPBoostOK** (*PointExpr* As String) As Boolean

參數

每個語法都有一個參數，用以指定要檢查的目標點。

PointNumber 透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標點。

Point 透過使用 SpelPoint 資料類型來指定目標點。

PointExpr 透過使用字串運算式來指定目標點。

傳回值

若將會使用 PTPBoost，會傳回 True，否則會傳回 False。

另請參閱

PTPBoost

PTPBoostOK 範例

VB 例:

```
If m_spel.PTPBoostOK(1) Then
    m_spel.Go(1)
End If
```

C# 例:

```
if (m_spel.PTPBoostOK(1))
    m_spel.Go(1);
```

PTran 方法，Spel 類別

描述

執行相對關節移動(以脈衝為單位)。

語法

Sub **PTran** (*JointNumber* As Integer, *Pulses* As Integer)

參數

JointNumber	要移動的特定關節。
Pulses	要移動的脈衝數。

另請參閱

JTran, Pulse

PTran 範例

VB 例:

```
' 以正方向將關節 1 移動 5000 脈衝。
m_spel.PTran(1, 5000)
```

C# 例:

```
// 以正方向將關節 1 移動 5000 脈衝。
m_spel.PTran(1, 5000);
```

PTRQ 方法，Spel 類別

描述

傳回指定關節的峰值轉矩。

語法

Function PTRQ (*JointNumber* As Integer) As Single

參數

JointNumber	代表關節編號的整數值 (範圍: 1~ 機器人的關節數量)

傳回值

傳回 0~1 的實數值。

另請參閱

ATCLR, ATRQ, PTCLR

PTRQ 範例

VB 例:

```
Dim peakTorque As Single
peakTorque = m_spel.PTRQ(1)
```

C# 例:

```
float peakTorque;
peakTorque = m_spel.PTRQ(1);
```

Pulse 方法，Spel 類別

描述

透過 PTP 控制方式，將機器人手臂移至透過所有機器人關節的脈衝值所指定的點。

語法

Sub **Pulse** (*J1Pulses* As Integer, *J2Pulses* As Integer, *J3Pulses* As Integer,
J4Pulses As Integer [, *J5Pulses* As Integer] [, *J6Pulses* As Integer]
[, *J7Pulses* As Integer] [, *J8Pulses* As Integer] [, *J9Pulses* As Integer])

參數

J1Pulses – *J9Pulses* 包含關節 1 – 9 之脈衝值的整數運算式。
關節 5 – 9 為選用。

NOTE： 脈衝值必須介於各關節指定範圍內。

另請參閱

Go, Move, Jump

Pulse 範例

VB 例:

```
m_spel.Pulse(5000, 1000, 0, 0)
```

C# 例:

```
m_spel.Pulse(5000, 1000, 0, 0);
```

Quit 方法，Spel 類別

描述

終止執行指定的任務。

語法

Sub **Quit** (*TaskNumber* As Integer)
Sub **Quit** (*TaskName* As String)

參數

TaskNumber	要中斷之任務的任務編號。 任務編號的範圍介於 1 至 32。
TaskName	包含任務名稱的字串運算式。

另請參閱

Halt, Resume, Xqt

Quit 範例

VB 例:

```
m_spel.Quit(3)
```

C# 例:

```
m_spel.Quit(3);
```

RadToDeg 方法，Spel 類別

描述

將徑度轉換成度。

語法

Function **RadToDeg** (*Radians* As Double) As Double

參數

Radians　包含要轉換成度之徑度的雙精度運算式。

傳回值

包含以度為單位之轉換值的雙精度值。

另請參閱

DegToRad

RadToDeg 範例

VB 例:

```
Dim deg As Double

deg = m_spel.RadToDeg(1)
```

C# 例:

```
double deg;
deg = m_spel.RadToDeg(1);
```

RebootController 方法，Spel 類別

描述

對目前接連的控制器進行重新啟動。

語法

Sub **RebootController** (*ShowStatusDialog* As Boolean)

參數

ShowStatusDialog　設定在重新啟動完成前，是否顯示狀態對話方塊的畫面。

True=顯示畫面、False=不顯示畫面

備註

使用 ShowStatusDialog，顯示帶有進度條的對話方塊。可以使用對話方塊或者 Abort 方法，停止操作。

另請參閱

Abort

RebootController 範例

VB 例:

```
m_spel.RebootController(True)
```

C# 例:

```
m_spel.RebootController(true);
```

RebuildProject 方法，Spel 類別

描述

完整重建在 Project 屬性中指定的目前專案。

語法

Sub **RebuildProject** ()

另請參閱

BuildProject, EnableEvent, EventReceived, Project, ProjectBuildComplete

RebuildProject 範例

VB 例:

```
With m_spel
    .Project =
"c:\EpsonRC70\projects\myproject\myproject.sprj"
    .RebuildProject()
End With
```

C# 例:

```
m_spel.Project =
@"c:\EpsonRC70\projects\myproject\myproject.sprj";
m_spel.RebuildProject();
```

Recover 方法，Spel 類別

描述

Recover 會將機器人移回安全防護打開時的所在位置。

語法

Function **Recover** () As Boolean

備註

Recover 方法可以在安全防護關閉後用來開啟機器人馬達，並將機器人緩慢移回安全防護打開時的所在位置。成功完成 Recover 後，您可執行 Cont 方法來繼續該週期。

若成功完成 Recover，將會傳回 True。若在恢復動作期間發生暫停、終止或安全防護打開的情況，Recover 將會傳回 False。

傳回值

若恢復動作已經完成，會傳回 True，否則會傳回 False。

另請參閱

Continue, Pause

Recover 範例

VB 例:

此範例是先執行 recover，再執行 continue

```
Sub btnCont_Click( _
   ByVal sender As System.Object, _
   ByVal e As System.EventArgs) Handles btnCont.Click
  Dim sts As Boolean
  Dim answer As Integer

  sts = m_spel.Recover()
  If sts = False Then
    Exit Sub
  End If
  answer = MsgBox("Ready to continue?", vbYesNo)
  If answer = vbYes Then
    m_spel.Continue()
  EndIF
End sub
```

此範例顯示只要按下按鈕便可使用按鈕來執行 recover 的方法。若在恢復動作期間放開按鈕，則會發出 pause 並終止恢復動作。若按住按鈕直到完成恢復動作，會隨即顯示一訊息。

```
    Sub btnRecover_MouseDown( _
        ByVal sender As System.Object, _
        ByVal e As System.Windows.Forms.MouseEventArgs) _
        Handles btnRecover.MouseDown
      Dim sts As Boolean

      sts = m_spel.Recover()
      If sts = True Then
          MsgBox("Recover complete")
      EndIf
    End Sub

    Sub btnRecover_MouseUp( _
        ByVal sender As System.Object, _
        ByVal e As System.Windows.Forms.MouseEventArgs) _
        Handles btnRecover.MouseUp

      m_spel.Pause()
    End Sub
```

C#例:

此範例是先執行 recover，再執行 continue

```
void btnCont_Click(object sender, EventArgs e)
{
    bool sts;
    DialogResult answer;

    sts = m_spel.Recover();
    if (sts == true){
            answer = MessageBox.Show("Continue?", "",
            MessageBoxButtons.YesNo);
        If (answer == DialogResult.Yes)
            m_spel.Continue();
    }
}
```

此範例顯示只要按下按鈕便可使用按鈕來執行 recover 的方法。若在恢復動作期間放開按鈕，則會發出 pause 並終止恢復動作。若按住按鈕直到完成恢復動作，會隨即顯示一訊息。

```
  void btnCont_Click(object sender, EventArgs e)
  {
    bool sts;

    sts = m_spel.Recover();
    if (sts == true)
        MessageBox.Show("Recover complete");
  }

  void btnRecover_MouseUp(oject sender, EventArgs e)
  {
    m_spel.Pause();
  }
```

Reset 方法，Spel 類別

描述

將控制器重置為初始狀態。

語法

Sub **Reset** ()

另請參閱

ResetAbort

Reset 範例

VB 例:

```
m_spel.Reset()
```

C# 例:

```
m_spel.Reset();
```

ResetAbort 方法，Spel 類別

描述

重置以 Stop 方法設定的終止旗標。

語法

Sub **ResetAbort** ()

備註

當執行 Stop 方法且沒有其他 Spel 方法處於週期中時，下一個 Spel 方法將會產生使用者終止錯誤。程序到此結束；不論是在何時發出 Stop，目前執行 Spel 方法的常式都將接收到錯誤。使用 **ResetAbort** 可清除此狀況。

NOTE：ResetAbortEnabled 屬性必須設為 True，ResetAbort 功能才可正常運作。

另請參閱

Abort, Reset, ResetAbortEnabled

ResetAbort 範例

VB 例:

```
Sub btnMcal_Click() Handles btnMcal.Click
  m_spel.ResetAbort()
  m_spel.MCal()
End Sub
```

C# 例:

```
void btnMCal_Click(object sender, EventArgs e)
{
   m_spel.ResetAbort();
   m_spel.MCal();
}
```

Resume 方法，Spel 類別

描述

使遭到 Halt 方法暫停的任務重新開始。

語法

Sub **Resume** (*TaskNumber* As Integer)
Sub **Resume** (*TaskName* As String)

參數

TaskNumber　已中斷之任務的任務編號。任務編號的範圍介於 1 至 32。

TaskName　包含任務名稱的字串運算式。

另請參閱

Quit, Xqt

Resume 範例

VB 例:

```
m_spel.Resume(2)
```

C# 例:

```
m_spel.Resume(2);
```

RunDialog 方法，Spel 類別

描述

執行 EPSON RC+ 7.0 對話方塊。

語法

Sub **RunDialog** (*DialogID* As SpelDialogs [, *Parent* As Form])

參數

DialogID	要執行之 EPSON RC+ 7.0 對話方塊的 ID。
Parent	選用。係一將成為視窗的父項之.NET 表單。

另請參閱

ShowWindow

RunDialog 範例

VB 例:

```
Sub btnRobotManager_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnRobotManager.Click

    m_spel.RunDialog(SpelDialogs.RobotManager)
End Sub
```

C# 例:

```
void btnRobotManager_Click(object sender, EventArgs e)
{
    m_spel.RunDialog(SpelDialogs.RobotManager);
}
```

SavePoints 方法，Spel 類別

描述

保存目前機器人的點至檔案。

語法

Sub **SavePoints** (*FileName* As String)

參數

FileName　　在目前專案中保存之點的檔案名稱。

另請參閱

LoadPoints

SavePoints 範例

VB 例:

```
With m_spel
    .SavePoints("part1.pts")
End With
```

C# 例:

```
m_spel.SavePoints("part1.pts");
```

Sense 方法，Spel 類別

描述

指定輸入條件，符合條件時會停止目標位置上方的機器人，以完成進行中的 Jump。

語法

Sub **Sense** (*Condition* As String) As Boolean

參數

Condition 指定 I/O 條件。
如需詳細資訊，請參閱 *SPEL+語言參考手冊* 中的 *Sense 陳述式*。

另請參閱

Jump, JS

Sense 範例

VB 例:

```
With m_spel
    .Sense("Sw(1) = On")
    .Jump("P1 SENSE")
    stoppedOnSense = .JS()
End With
```

C# 例:

```
m_spel.Sense("Sw(1) = On");
m_spel.Jump("P1 SENSE");
stoppedOnSense = m_spel.JS();
```

SetIODef 方法，Spel 類別

描述

設定輸入、輸出或記憶體 I/O 位元、位元組或字元的 I/O 標籤與描述。

語法

Sub **SetIODef** (*Type* As SpelLabelTypes, *Index* As Integer, *Label* As String, *Description* As String)

參數

Type 指定 I/O 類型，如下所示：

InputBit = 1 InputByte = 2 InputWord = 3

OutputBit = 4 OutputByte = 5 OutputWord = 6

MemoryBit = 7 MemoryByte = 8 MemoryWord = 9

InputReal = 10 OutputReal = 11

Index 指定位元或埠號。

Label 指定新建標籤。

Description 指定新建描述。

備註

使用 SetIODef 可定義任何 I/O 點的標籤與描述。

另請參閱

GetIODef

SetIODef 範例

VB 例:

```
Dim label, desc As String
label = "StartCycle"
desc = "Starts the robot cycle"
m_spel.SetIODef(SpelLabelTypes.InputBit, 0, label, desc)
```

C# 例:

```
string label, desc;
label = "StartCycle";
desc = "Starts the robot cycle";
m_spel.SetIODef(SpelLabelTypes.InputBit, 0, label, desc);
```

SetPoint 方法，Spel 類別

描述

設定目前機器人某個點的座標資料。

語法

Sub **SetPoint**(*PointNumber* As Integer, *Point* As SpelPoint)
Sub **SetPoint**(*PointLabel* As String, *Point* As SpelPoint)

Sub **SetPoint**(*PointNumber* As Integer, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single)
Sub **SetPoint**(*PointLabel* As String, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single)

Sub **SetPoint**(*PointNumber* As Integer, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single,
Local As Integer, *Hand* As SpelHand)
Sub **SetPoint**(*PointLabel* As String, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single,
Local As Integer, *Hand* As SpelHand)

Sub **SetPoint**(*PointNumber* As Integer, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single,
V As Single, *W* As Single)
Sub **SetPoint**(*PointLabel* As String, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single,
V As Single, *W* As Single)

Sub **SetPoint**(*PointNumber* As Integer, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single,
V As Single, *W* As Single, *Local* As Integer, *Hand* As SpelHand, *Elbow* As SpelElbow, *Wrist* As SpelWrist, *J4Flag* As Integer, *J6Flag* As Integer)
Sub **SetPoint**(*PointLabel* As String, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single,
V As Single, *W* As Single, *Local* As Integer, *Hand* As SpelHand, *Elbow* As SpelElbow, *Wrist* As SpelWrist, *J4Flag* As Integer, *J6Flag* As Integer)

Sub **SetPoint**(*PointNumber* As Integer, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single,
V As Single, *W* As Single, *S* As Single, *T* As Single)
Sub **SetPoint**(*PointLabel* As String, *X* As Single, *Y* As Single, *Z* As Single, *U* As Single,
V As Single, *W* As Single, *S* As Single, *T* As Single)

Sub **SetPoint**(*PointNumber* As Integer, *PointExpr* As String)
Sub **SetPoint**(*PointLabel* As String, *PointExpr* As String)

參數

PointNumber	指定目前機器人點記憶體中某個點之點編號的整數運算式。
X	指定點的 X 座標。
Y	指定點的 Y 座標。
Z	指定點的 Z 座標。
U	指定點的 U 座標。
V	指定點的 V 座標。
W	指定點的 W 座標。
S	指定點的 S 座標。
T	指定點的 T 座標。
Local	指定點的本地編號。沒有本地時使用 0。
Hand	指定點的手部方向。
Elbow	指定點的肘部方向。
Wrist	指定點的腕部方向。
PointExpr	透過使用字串運算式來指定點。

NOTE

請勿將整數值輸入至 X、Y、Z、U、V、W、S 及 T 參數。使用單一變數或直接輸入單一類型值。

另請參閱

GetPoint, LoadPoints, SavePoints

SetPoint 範例

VB 例:

```
Dim pt As SpelPoint
' 取得 P1 的座標
pt = m_spel.GetPoint(1)
' 變更坐標
pt.U = pt.U - 10.5
m_spel.SetPoint(1, pt)
```

C# 例:

```
SpelPoint pt;
// 取得 P1 的座標
pt = m_spel.GetPoint(1);
// 變更坐標
pt.U = pt.U - 10.5;
m_spel.SetPoint(1, pt);
```

SetVar 方法，Spel 類別

描述

設定控制器中 SPEL+全域保留變數的值。

語法

Sub **SetVar** (*VarName* As String, *Value* As Object)

參數

VarName	SPEL+全域保留變數的名稱。
Value	新建值。

備註

您可使用 SetVar 來設定單一變數及陣列變數的值。請參閱以下範例。

另請參閱

GetVar

SetVar 範例

VB 例:

```
m_spel.SetVar("g_myIntVar", 123)

Dim i, myArray(10) As Integer
For i = 1 To 10
  myArray(i) = i
Next i
m_spel.SetVar("g_myIntArray", myArray)

m_spel.SetVar("g_myIntArray(1)", myArray(1))
```

C# 例:

```
m_spel.SetVar("g_myIntVar", 123);

int[] myArray = new int[10];
for(int i = 1; i < 10; i++)
    myArray[i] = i;

m_spel.SetVar("g_myIntArray", myArray);

m_spel.SetVar("g_myIntArray[1]", myArray[1]);
```

SFree 方法，Spel 類別

描述

從伺服系統控制，釋放指定的機器人軸。

語法

Sub **SFree** ()
Sub **SFree** (ParamArray *Axes*() As Integer)

參數

Axes 包含要釋放之各機器人軸的某一元素之整數參數陣列。
您可指定介於 1 – 9 的軸編號。

另請參閱

SLock

SFree 範例

VB 例:

```
' SFree 軸 1 和 2
m_spel.SFree(1, 2)
```

C# 例:

```
// SFree 軸 1 和 2
m_spel.SFree(1, 2);
```

ShowWindow 方法，Spel 類別

描述

顯示 EPSON RC+ 7.0 視窗。

語法

Sub **ShowWindow** (*WindowID* As SpelWindows [, *Parent* As Form])

參數

WindowID	要顯示的 EPSON RC+ 7.0 視窗的 ID。
Parent	選用。係一將成為視窗的父項之.NET 表單。

備註

您可使用 Parent 參數指定視窗的.NET 父項。若無法使用.NET 父表單，您必須省略 Parent 參數並使用 ParentWindowHandle 屬性來設定父項的控制代碼。

另請參閱

HideWindow, ParentWindowHandle, RunDialog, ServerOutOfProcess

ShowWindow 範例

VB 例:

```
Sub btnShowIOMonitor_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnShowIOMonitor.Click

    m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor, Me)
End Sub
```

C# 例:

```
void btnShowIOMonitor_Click(object sender, EventArgs e)
{
    m_spel.ShowWindow(RCAPINet.SpelWindows.IOMonitor, this);
}
```

SimGet 方法，Spel 類別

描述

取得模擬器的各種物件屬性的設定值。

語法

Sub **SimGet** (*Object* As String, *Property* As SpelSimProps, ByRef *Value* As Boolean)

Sub **SimGet** (*RobotName* As String, *HandName* As String, *Property* As SpelSimProps, ByRef *Value* As Boolean)

Sub **SimGet** (*Object* As String, *Property* As SpelSimProps, ByRef *Value* As Double)

Sub **SimGet** (*RobotName* As String, *HandName* As String, *Property* As SpelSimProps, ByRef *Value* As Double)

Sub **SimGet** (*Object* As String, *Property* As SpelSimProps, ByRef *Value* As Integer)

Sub **SimGet** (*RobotName* As String, *HandName* As String, *Property* As SpelSimProps, ByRef *Value* As Integer)

Sub **SimGet** (*Object* As String, *Property* As SpelSimProps, ByRef *Value* As Boolean)

Sub **SimGet** (*RobotName* As String, *HandName* As String, *Property* As SpelSimProps, ByRef *Value* As String)

參數

Object	代表取得了屬性值的物件名稱的字串變數
RobotName	代表安裝了“Hand”中指定夾具的機器人名稱的字串變數
HandName	代表取得屬性值的夾具名稱的字串變數
Property	取得值的屬性的名稱。
Value	代表傳回值得變數

備註

本方法用於取得模擬器中的各種物件的屬性設定值。

有關各屬性的詳細資訊，請參閱以下手冊。
EPSON RC+ 7.0 SPEL+ 語言參考 SimGet

另請參閱

SimSet

SimGet 範例

VB 例:

```
Dim posX As Double
m_spel.SimGet("SBox_1", SpelSimProps.PositionX, posX)
```

C# 例:

```
double posX;
SimGet("SBox_1", SpelSimProps.PositionX, out posX);
```

SimResetCollision 方法，Spel 類別

描述

重設碰撞偵測。

語法

Sub **SimResetCollision** ()

備註

執行本方法后，如果機器人和物件之間沒有發生碰撞，則重設碰撞狀態並更新模擬器的 3D 顯示。如果機器人和物件之間發生碰撞，則不會解除碰撞狀態，模擬器的 3D 顯示也不會更新。

有關詳細資訊，請參閱以下手冊。
EPSON RC+ 7.0 SPEL+ 語言參考 SimSet

另請參閱

SimSet

SimResetCollision 範例

VB 例:

```
m_spel.SimResetCollision ()
```

C# 例:

```
m_spel.SimResetCollision ();
```

SimSet 方法，Spel 類別

描述

設定模擬器中各種物件的屬性。還可以進行機器人動作和物件操作，以及模擬器設定等操作。

語法

Sub **SimSet** (*Object* As String, *Property* As SpelSimProps, *Value* As Boolean)

Sub **SimSet** (*RobotName* As String, *HandName* As String, *Property* As SpelSimProps, *Value* As Boolean)

Sub **SimSet** (*Object* As String, *Property* As SpelSimProps, *Value* As Integer)

Sub **SimSet** (*RobotName* As String, *HandName* As String, *Property* As SpelSimProps, *Value* As Integer)

Sub **SimSet** (*Object* As String, *Property* As SpelSimProps, *Value* As Double)

Sub **SimSet** (*RobotName* As String, *HandName* As String, *Property* As SpelSimProps, *Value* As Double)

Sub **SimSet** (*Object* As String, *Property* As SpelSimProps, *Value* As String)

Sub **SimSet** (*RobotName* As String, *HandName* As String, *Property* As SpelSimProps, *Value* As String)

參數

Object	代表取得屬性值的物件名稱的字串變數
RobotName	代表安裝了“Hand”中指定夾具的機器人名稱的字串變數
HandName	代表取得屬性值的夾具名稱的字串變數
Property	設定了新值的屬性的名稱
Value	新值的運算式

備註

本方法用於設定和操作模擬器中的各種物件的屬性、機器人動作個和更改模擬器設定。

NOTE

屬性不能指定 SpelSimProps.Type。

有關各屬性的詳細資訊，請參閱以下手冊。
EPSON RC+ 7.0 SPEL + 語言參考 SimSet

另請參閱

SimGet

SimSet 範例

VB 例:

```
m_spel.SimSet ("SBox_1", SpelSimProps.PositionX, 100.0)
```

C# 例:

```
m_spel.SimSet("SBox_1", SpelSimProps.PositionX, 100.0);
```

SimSetParent 方法，Spel 類別

描述

設定物件的操作。

語法

Sub **SimSetParent** (*Object* As String)

Sub **SimSetParent** (*Object* As String, *ParentObject* As String)

參數

Object	代表設定了父層物件的物件名稱的字串變數
ParentObject	代表父層物件的字串變數

備註

對於“Object”中指定的物件，會將“ParentObject”中指定物件設置為其父層物件。可以省略“ParentObject”。這種情況下“Object”中指定的物件則為父層物件。例如，如果“Object”中指定的物件是某個物件的子物件，則會解除對其子物件的設定。

此外，如果“Object”中指定的物件是零件或者機械臂安裝設備，則無法指定父層物件。

有關 SetParent 中可以指定的物件，請參閱以下手冊。

EPSON RC+ 7.0 SPEL+ 語言參考 SimSet

如果使用攝像機物件，只有設定為固定攝像機時，才可以使用 SetParent。

另請參閱

SimSet

SimSetParent 範例

VB 例:

```
m_spel.SimSetParent ("SBox_1")
```

C# 例:

```
m_spel.SimSetParent ("SBox_1");
```

SimSetPick 方法，Spel 類別

描述

使用指定的機器人抓取物件。

語法

Sub **SimSetPick** (*RobotName* As String, *Object* As String)
Sub **SimSetPick** (*RobotName* As String, *Object* As String, *ToolNumber* As Integer)

參數

RobotName	代表 Pick 的機器人名稱的字串變數
Object	代表被 Pick 的物件名稱的字串變量
ToolNumber	代表 Pick 時使用的 Tool 編號的運算式

備註

“Robot”中指定的機器人會抓取“Object”中指定的物件。被抓取的物件則會被登錄為機器人的一部分。然後在 Tool 中指定任意工具編號，就可以用指定的工具進行抓取。如果不指定“Tool”，則會使用 Tool0 進行抓取。

無法抓取已注冊為零件的物件或設爲機械手安裝設備的物件。此外，也無法抓取攝像機。

有關詳細資訊，請參閱以下手冊。
EPSON RC+ 7.0 SPEL+ 語言參考 SimSet

另請參閱

SimGet, SimSet, SimSetPlace

SimSetPick 範例

VB 例:

```
m_spel.SimSetPick ("Robot1", "SBox_1", 1)
```

C# 例:

```
m_spel.SimSetPick ("Robot1", "SBox_1", 1);
```

SimSetPlace 方法，Spel 類別

描述

使用指定的機器人放置物件。

語法

Sub **SimSetPlace** (*RobotName* As String, *Object* As String)

參數

RobotName	代表 Place 機器人名稱的字串變數
Object	代表被 Place 的物件名稱的字串變數

備註

“Robot”中指定的機器人會放置“Object”中指定的物件。被防止的物件將被解除該機器人零件的注冊。

不可放置已被解除零件注冊的物件。

有關詳細資訊，請參閱以下手冊。

EPSON RC+ 7.0 SPEL+ 語言參考 SimSet

另請參閱

SimGet, SimSet, SimSetPick

SimSetPlace 範例

VB 例:

```
m_spel.SimSetPlace ("Robot1", "SBox_1")
```

C# 例:

```
m_spel.SimSetPlace ("Robot1", "SBox_1");
```

Shutdown 方法，Spel 類別

描述

關閉或重啟 Windows。

語法

Sub **Shutdown** (*Mode* As SpelShutdownMode)

參數

Mode	0 = 關閉 Windows。 1 = 重啟 Windows。

另請參閱

Reset

Shutdown 範例

VB 例:

```
' 重啟 Windows
m_spel.Shutdown(1)
```

C# 例:

```
// 重啟 Windows
m_spel.Shutdown(1);
```

SLock 方法，Spel 類別

描述

將指定軸傳回伺服系統控制。

語法

Sub **SLock** ()
Sub **SLock** (ParamArray *Axes*() As Integer)

參數

Axes 包含要鎖定之各機器人軸的一個元素的整數參數陣列。
您可指定介於 1 – 9 的軸編號。

另請參閱

SFree

SLock 範例

VB 例:

```
' 將軸 1 和 2 傳回伺服系統控制。
m_spel.SLock(1, 2)
```

C# 例:

```
// 將軸 1 和 2 傳回伺服系統控制。
m_spel.SLock(1, 2);
```

Speed 方法，Spel 類別

描述

指定要搭配 PTP 指令 Go、Jump 及 Pulse 使用的手臂速度。

語法

Sub **Speed** (*PointToPointSpeed* As Integer [, *JumpDepartSpeed* As Integer]
[, *JumpApproSpeed* As Integer])

參數

PointToPointSpeed	指定要搭配 PTP 指令 Go、Jump 及 Pulse 使用的手臂速度。
JumpDepartSpeed	介於 1-100 的整數，代表 Jump 指令的 Z 軸向上動作速度。
JumpApproSpeed	介於 1-100 的整數，代表 Jump 指令的 Z 軸向下動作速度。

另請參閱

Accel, Jump, Go

Speed 範例

VB 例:

```
m_spel.Speed(50)
```

C# 例:

```
m_spel.Speed(50);
```

SpeedR 方法，Spel 類別

描述

指定使用 ROT 時的工具旋轉速度。

語法

Sub **SpeedR** (*RotationSpeed* As Single)

參數

RotationSpeed　指定工具旋轉速度(度／秒)。

另請參閱

Arc, Arc3, BMove, Jump3CP, Power, TMove

SpeedR 範例

VB 例:

```
m_spel.SpeedR(100)
```

C# 例:

```
m_spel.SpeedR(100);
```

SpeedS 方法，Spel 類別

描述

指定要搭配連續路徑指令 Jump3CP、Move、Arc 及 CVMove 使用的手臂速度。

語法

Sub **SpeedS** (*LinearSpeed* As Single [, *JumpDepartSpeed* As Single] [, *JumpApproSpeed* As Single])

參數

LinearSpeed	指定要搭配連續路徑指令 Jump3CP、Move、Arc 及 CVMove 使用的手臂速度。
JumpDepartSpeed	代表 Jump3CP 指令的 Z 軸向上動作速度的單一運算式。
JumpApproSpeed	代表 Jump3CP 指令的 Z 軸向下動作速度的單一運算式。

另請參閱

AccelS, Jump3CP, Move, TMove

SpeedS 範例

VB 例:

```
m_spel.SpeedS(500)
```

C# 例:

```
m_spel.SpeedS(500);
```

Start 方法，Spel 類別

描述

啟動一個 SPEL+程式。

語法

Sub **Start** (*ProgramNumber* As Integer)

參數

ProgramNumber 要啟動的程式編號，對應至 SPEL+中的 64 個 main 函數，如下表所示。
範圍介於 0 至 63。

程式編號	SPEL+函數名稱
0	main
1	main1
2	main2
3	main3
4	main4
5	main5
…	…
63	main63

備註

執行 **Start** 時，控制將會立即回到調用程式。您無法啟動正在執行的程式。請注意，Start 會清除控制器中的全域變數且會載入預設的機器人點。

另請參閱

Continue, Pause, Stop, Xqt

Start 範例

VB 例:

```
Sub btnStart_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnStart.Click

    m_spel.Start(0)
End Sub
```

C# 例:

```
void btnStart_Click(object sender, EventArgs e)
{
    m_spel.Start(0);
}
```

StartBGTask 方法，Spel 類別

描述

啟動一個 SPEL+任務作為背景任務。

語法

Sub **StartBGTask** (*FuncName* As String)

參數

FuncName　　要執行之函數的名稱。

備註

使用 StartBGTask 可在控制器中啟動 Spel+背景任務。背景任務必須在控制器中啟用。

請注意，BGMain 會在控制器切換至自動模式時自動啟動，因此通常不需要使用 StartBGTask。StartBGTask 是提供用於需要停止所有任務、接著重啟背景任務的情況。

另請參閱

Call, Start, Stop, Xqt

StartBGTask 範例

VB 例:

```
' 停止所有任務，包含背景任務
m_spel.Stop(SpelStopType.StopAllTasks)
…
m_spel.RebuildProject()

' 啟動主要背景任務
m_spel.StartBGTask("BGMain")
```

C# 例:

```
// 停止所有任務，包含背景任務
m_spel.Stop(SpelStopType.StopAllTasks);
…
m_spel.RebuildProject();

// 啟動主要背景任務
m_spel.StartBGTask("BGMain");
```

Stat 方法，Spel 類別

描述

傳回控制器狀態。

語法

Function **Stat** (*Address* As Integer) As Integer

參數

Address 指定代表控制器狀態的地址。
(0 至 2 的整數)

傳回值

傳回代表控制器狀態的 4 位元組值。(請參閱下表。)

地址	位元		位元開啟時的控制器狀態
0	0-15	&H1-&H8000	任務 1 至 16 正在執行(Xqt)或處於停止狀態
	16	&H10000	任務正在執行
	17	&H20000	暫停狀態
	18	&H40000	錯誤狀態
	19	&H80000	TEACH 模式
	20	&H100000	緊急停止狀態
	21	&H200000	低運行功率模式
	22	&H400000	安全防護打開
	23	&H800000	啟動開關開啟
	24	&H1000000	未定義
	25	&H2000000	未定義
	26	&H4000000	測試模式
	27	&H8000000	T2 模式狀態
	28-31		未定義
1	0	&H1	符合 Jump...Sense 陳述式中的條件時，目標位置上方的停止日誌。(此日誌會在執行其他 Jump 陳述式時消除)。
	1	&H2	滿足 Go/Jump/Move...Till 陳述式中的條件時，中間移動位置的停止日誌。(此日誌會在執行其他 Go/Jump/Move...Till 陳述式時消除)
	2	&H4	未定義
	3	&H8	偵測到 Trap 陳述式時停止進行中動作的日誌。
	4	&H10	馬達開啟狀態
	5	&H20	起始點位置
	6	&H40	低運行功率模式
	7	&H80	未定義
	8	&H100	關節#4 已接合。
	9	&H200	關節#3 已接合。
	10	&H400	關節#2 已接合。
	11	&H800	關節#1 已接合。
	12	&H1000	關節#6 已接合。
	13	&H2000	關節#5 已接合。
	14	&H4000	T 軸已接合。
	15	&H8000	S 軸已接合。
	16	&H10000	關節#7 已接合。
	17-31		未定義
2	0-15	&H1-&H8000	任務 17 至 32 正在執行(Xqt)或處於停止狀態

另請參閱

EStopOn, PauseOn, SafetyOn

Stat 範例

VB 例:

```
Dim ctr_stat As Integer
ctr_stat = m_spel.Stat(0)
```

C# 例:

```
int ctr_stat;
ctr_stat = m_spel.Stat(0);
```

Stop 方法，Spel 類別

描述

停止控制器中執行的所有一般 SPEL+任務，以及選擇性停止所有背景任務。

語法

Sub **Stop** ()
Sub **Stop** (SpelStopType *StopType*)

參數

StopType　選用。指定僅停止正常任務(StopNormalTasks)或是停止所有任務(StopAllTasks)。
如果省略，則會指定 StopNormalTasks。

NOTE：如果在 ResetAbortEnabled 設為 True 時執行 Stop 方法，則執行 Start 或 Reset 方法時會出現 10101 錯誤。
若要解除錯誤，請在執行 Stop 方法後執行 ResetAbort 方法。

另請參閱

Continue, Pause, ResetAbort, ResetAbortEnabled, Start, SpelStopType

Stop 範例

VB 例:

```
Sub btnStop_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnStop.Click

    m_spel.Stop()
End Sub
```

C# 例:

```
void btnStop_Click(object sender, EventArgs e)
{
    m_spel.Stop();
}
```

Sw 方法，Spel 類別

描述

傳回選取的輸入位元狀態。

語法

Function **Sw** (*BitNumber* As Integer) As Boolean
Function **Sw** (*Label* As String) As Boolean

參數

BitNumber　代表任一個標準或擴展輸入的整數運算式。

Label　包含輸入位元標籤的字串運算式。

傳回值

若指定輸入位元已開啟，會傳回 True，否則會傳回 False。

另請參閱

In, InBCD, MemSw, Off, On, Oport

Sw 範例

VB 例:

```
If m_spel.Sw(1) Then
    m_spel.On(2)
End If
```

C# 例:

```
if (m_spel.Sw(1))
    m_spel.On(2);
```

TargetOK 方法，Spel 類別

描述

傳回指示目前位置到目標位置的 PTP(點至點)動作是否可行的狀態。

語法

Function **TargetOK** (*PointNumber* As Integer) As Boolean
Function **TargetOK** (*Point* As SpelPoint) As Boolean
Function **TargetOK** (*PointExpr* As String) As Boolean

參數

每個語法都有一個參數，用以指定要檢查的目標點。

PointNumber	透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標點。
Point	透過使用 SpelPoint 資料類型來指定目標點。
PointExpr	透過使用字串運算式來指定目標點。

傳回值

若目標可以從目前位置移動，會傳回 True，否則會傳回 False。

另請參閱

Go, Jump, Move, TGo, TMove

TargetOK 範例

VB 例:

```
If m_spel.TargetOK("P1 /F") Then
    m_spel.Go("P1 /F")
End If
```

C# 例:

```
if (m_spel.TargetOK("P1 /F"))
    m_spel.Go("P1 /F");
```

TasksExecuting 方法，Spel 類別

描述

若正在執行任何 SPEL+任務，會傳回 True。

語法

Function **TasksExecuting** () As Boolean

傳回值

若正在執行任何 SPEL+任務，會傳回 True，否則會傳回 False。

另請參閱

TaskState, Xqt

TasksExecuting 範例

VB 例:

```
tasksRunning = m_spel.TasksExecuting()
```

C# 例:

```
tasksRunning = m_spel.TasksExecuting();
```

TaskState 方法，Spel 類別

描述

傳回任務的狀態。

語法

Function **TaskState** (*TaskNumber* As Integer) As SpelTaskState
Function **TaskState** (*TaskName* As String) As SpelTaskState

參數

TaskNumber 傳回執行狀態的任務編號。

TaskName 包含任務名稱的字串運算式。

傳回值

SpelTaskState 值。

另請參閱

TasksExecuting, Xqt

TaskState 範例

VB 例:

```
Dim taskState As SpelTaskState
taskState = m_spel.TaskState(2)
```

C# 例:

```
SpelTaskState taskState;
taskState = m_spel.TaskState(2);
```

TeachPoint 方法，Spel 類別

描述

執行用以允許操作員步進示教一個點的對話方塊。

語法

Function **TeachPoint** (*PointFile* As String, *PointNumber* As Integer, *Prompt* As String) As Boolean
Function **TeachPoint** (PointFile As String, PointName As String, Prompt As String) As Boolean
Function **TeachPoint** (PointFile As String, PointNumber As Integer, Prompt As String, Parent As Form) As Boolean
Function **TeachPoint** (PointFile As String, PointName As String, Prompt As String, Parent As Form) As Boolean

參數

PointFile	包含點檔案名稱的字串。
PointNumber	要示教的點編號。
PointName	顯示點標簽的字串。
Prompt	包含顯示於示教對話方塊底部之說明文字的字串。
Parent	.NET 表單成爲視窗的父級 (可選項)

傳回值

若操作員按一下示教按鈕，會傳回 True；若操作員按一下取消，會傳回 False。

備註

使用 TeachPoints 可讓操作員在控制器中示教一個機器人點。執行 TeachPoints 時，點文件會從控制器載入。按一下示教按鈕時，點會在控制器中示教，且點文件會儲存至控制器。

TeachPoint 範例

VB 例:

```
Sub btnTeachPick_Click( _
       ByVal sender As System.Object, _
       ByVal e As System.EventArgs) _
       Handles btnTeachPick.Click

    Dim sts As Boolean
    Dim prompt As String

    prompt = "Jog to Pick position and click Teach"
    sts = m_spel.TeachPoint("points.pts", 1, prompt)

End Sub
```

C# 例:

```
void btnTeachPick_Click(object sender, EventArgs e)
{
    bool sts;
    string prompt;

    prompt = "Jog to Pick position and click Teach";
    sts = m_spel.TeachPoint("points.pts", 1, prompt);

}
```

Till 方法，Spel 類別

描述

指定事件條件，符合條件時會減速並停止中間位置的機器人，以完成進行中的動作命令(Jump、Go、Move 等)。

語法

Sub **Till** (*Condition* As String) As Boolean

參數

Condition	指定 I/O 條件。如需詳細資訊，請參閱 *SPEL+語言參考手冊* 中的 *Till 陳述式*。

另請參閱

Go, Jump, JS, Sense, TillOn

Till 範例

VB 例:

```
With m_spel
    .Till("Sw(1) = On")
    .Go("P1 TILL")
End With
```

C# 例:

```
m_spel.Till("Sw(1) = On");
m_spel.Go("P1 TILL");
```

TillOn 方法，Spel 類別

描述

若 till 條件在最後 Go/Jump/Move 陳述式期間發生停止狀況，會傳回 True。

語法

Function **TillOn** () As Boolean

傳回值

若機器人因 Till 條件而停止，會傳回 True，否則會傳回 False。

備註

使用 TillOn 可檢查是否已在最後動作命令期間用 Till 開啟 Till 條件。

TillOn 相當於((Stat(1) And 2) <> 0)

另請參閱

Jump, Till

TillOn 範例

VB 例:

```
If m_spel.TillOn() Then
    m_spel.Jump(2)
End If
```

C# 例:

```
if (m_spel.TillOn())
    m_spel.Jump(2);
```

TGo 方法，Spel 類別

描述

在目前工具座標系統中執行 PTP 相對動作。

語法

Sub **TGo** (*PointNumber* As Integer)
Sub **TGo** (*Point* As SpelPoint)
Sub **TGo** (*Point* As SpelPoint, *AttribExpr* As String)
Sub **TGo** (*PointExpr* As String)

參數

每個語法都有一個參數，用以指定在 TGo 動作期間手臂移動的結束點。此為 PTP 動作結束時的最終位置。

PointNumber	透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
Point	透過使用 SpelPoint 資料類型來指定結束點。
AttribExpr	透過使用字串運算式來指定結束點屬性。
PointExpr	透過使用字串運算式來指定結束點。

另請參閱

Accel, Speed
Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move
BGo, BMove, TMove
CP, Till

TGo 範例

VB 例:

```
' 使用點編號指定的點
m_spel.Tool(1)
m_spel.TGo(100)

' 使用 SpelPoint 指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.TGo(pt)

' 使用運算式指定的點
m_spel.TGo("P0 /L /2")
m_spel.TGo("P1 :Z(-20)")

' 使用平行處理
m_spel.TGo("P1 !D50; On 1; D90; Off 1!")

' 使用標籤指定的點
m_spel.TGo("pick")
```

C# 例:

```
// 使用點編號指定的點
m_spel.Tool(1);
m_spel.TGo(100);

// 使用 SpelPoint 指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.TGo(pt);

// 使用運算式指定的點
m_spel.TGo("P0 /L /2");
m_spel.TGo("P1 :Z(-20)");

// 使用平行處理
m_spel.TGo("P1 !D50; On 1; D90; Off 1!");

// 使用標籤指定的點
m_spel.TGo("pick");
```

TLClr 方法，Spel 類別

描述

清除(取消定義)工具座標系統。

語法

Sub **TLClr** (*ToolNumber* As Integer)

參數

ToolNumber	代表要清除(取消定義)之工具的整數運算式。 (工具 0 為預設工具，無法清除。)

另請參閱

Tool, TLDef

TLClr 範例

VB 例:

```
m_spel. TLClr(1)
```

C# 例:

```
m_spel. TLClr(1);
```

TLDef 方法，Spel 類別

描述

傳回工具定義狀態。

語法

Function **TLDef** (*ToolNumber* As Integer) As Boolean

參數

ToolNumber　代表要傳回狀態之工具的整數運算式。

傳回值

若指定工具已定義，會傳回 True，否則會傳回 False。

另請參閱

Tool, TLClr

TLDef 範例

VB 例:

```
m_spel.TLDef(1)
```

C# 例:

```
m_spel.TLDef(1);
```

TLSet 方法，Spel 類別

描述

定義工具座標系統。

語法

Sub **TLset** (*ToolNumber* As Integer , *Point* As SpelPoint)
Sub **TLset** (*ToolNumber* As Integer, *XCoord* As Single, *YCoord* As Single, *ZCoord* As Single,
UCoord As Single, *VCoord* As Single, *WCoord* As Single)

參數

ToolNumber	介於 1-15 的整數運算式，代表 15 個工具中要定義的工具。(Tool 0 為預設工具，無法改變。)
Point	包含點資料的 SpelPoint。
XCoord	工具座標系統原點 X 座標。
YCoord	工具座標系統原點 Y 座標。
ZCoord	工具座標系統原點 Z 座標。
UCoord	繞 Z 軸的工具座標系統旋轉。
VCoord	繞 Y 軸的工具座標系統旋轉。
WCoord	繞 X 軸的工具座標系統旋轉。

另請參閱

Arm, Armset, GetTool, Tool

TLSet 範例

VB 例:

```
m_spel.TLSet(1, .5, 4.3, 0, 0, 0, 0)
```

C# 例:

```
m_spel.TLSet(1, .5, 4.3, 0, 0, 0, 0);
```

TMove 方法，Spel 類別

描述

在目前工具座標系統中執行線性插補相對動作。

語法

Sub **TMove** (*PointNumber* As Integer)
Sub **TMove** (*Point* As SpelPoint)
Sub **TMove** (*Point* As SpelPoint, *AttribExpr* As String)
Sub **TMove** (*PointExpr* As String)

參數

每個語法都有一個參數，用以指定在 TMove 動作期間手臂移動的結束點。此為線性插補動作結束時的最終位置。

PointNumber	透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定結束點。
Point	透過使用 SpelPoint 資料類型來指定結束點。
AttribExpr	透過使用字串運算式來指定結束點屬性。
PointExpr	透過使用字串運算式來指定結束點。

另請參閱

AccelR, AccelS, SpeedR, SpeedS
Arc, Arc3, CVMove, Go, Jump, Jump3, Jump3CP, Move
BGo, BMove, TGo
CP, Till

TMove 範例

VB 例:

```
' 使用點編號指定的點
m_spel.Tool(1)
m_spel.TMove(100)

' 使用 SpelPoint 指定的點
Dim pt As SpelPoint
pt = m_spel.GetPoint("P*")
pt.X = 125.5
m_spel.TMove(pt)

' 使用運算式指定的點
m_spel.TMove ("P0")
m_spel.TMove ("XY(0, 0, -20, 0)")

' 使用平行處理
m_spel.TMove("P1 !D50; On 1; D90; Off 1!")

' 使用標籤指定的點
m_spel.TMove("pick")
```

C# 例:

```
// 使用點編號指定的點
m_spel.Tool(1);
m_spel.TMove(100);

// 使用 SpelPoint 指定的點
SpelPoint pt;
pt = m_spel.GetPoint("P0");
pt.X = 125.5;
m_spel.TMove(pt);

// 使用運算式指定的點
m_spel.TMove ("P0");
m_spel.TMove ("XY(0, 0, -20, 0)");

// 使用平行處理
m_spel.TMove("P1 !D50; On 1; D90; Off 1!");

// 使用標籤指定的點
m_spel.TMove("pick");
```

Tool 方法，Spel 類別

描述

選擇目前機器人工具。

語法

Sub **Tool** (*ToolNumber* As Integer)

參數

ToolNumber	介於 0-15 的整數，代表 16 個工具定義中要搭配後續動作指令使用的工具定義。

另請參閱

TLSet, Arm, TGo, TMove

Tool 範例

VB 例:

```
m_spel.Tool(1)
m_spel.TGo(100)
```

C# 例:

```
m_spel.Tool(1);
m_spel.TGo(100);
```

TrapStop 方法，Spel 類別

描述

若目前機器人在上一個動作命令期間因 trap 而停止，會傳回 True。

語法

Function **TrapStop** () As Boolean

傳回值

若機器人因 trap 而停止，會傳回 True，否則會傳回 False。

另請參閱

EStopOn, ErrorOn

TrapStop 範例

VB 例:

```
If m_spel.TrapStop() Then
    MsgBox "Robot stopped by Trap"
End If
```

C# 例:

```
if (m_spel.TrapStop())
    MessageBox.Show("Robot stopped by trap");
```

TW 方法，Spel 類別

描述

傳回 WAIT 條件及 WAIT 定時器間隔的狀態。

語法

Function **TW** () As Boolean

傳回值

若發生超時，會傳回 True，否則會傳回 False。

另請參閱

WaitMem, WaitSw

TW 範例

VB 例:

```
Const PartPresent = 1
m_spel.WaitSw(PartPresent, True, 5)
If m_spel.TW() Then
    MsgBox "Part present time out occurred"
End If
```

C# 例:

```
const int PartPresent = 1;
m_spel.WaitSw(PartPresent, True, 5);
if (m_spel.TW())
    MessageBox.Show("Part present time out occurred");
```

UserHasRight 方法，Spel 類別

描述

傳回目前登錄的使用者是否具有指定的權限。

語法

Function **UserHasRight** (SpelUserRights *Right*) As Boolean

參數

Right　　要針對目前登錄使用者進行檢查的權限。

傳回值

若使用者擁有指定權限，會傳回 True，否則會傳回 False。

另請參閱

Login, GetCurrentUser

UserHasRight 範例

VB 例:

```
Dim hasRight As Boolean
hasRight = m_spel.UserHasRight(SpelUserRights.EditPoints)
```

C# 例:

```
bool hasRight;
hasRight = m_spel.UserHasRight(SpelUserRights.EditPoints);
```

VCal 方法，Spel 類別

描述

此命令可讓您執行視覺校準週期。

語法

Sub **VCal** (*CalibName* As String)
Sub **VCal** (*CalibName* As String, ByRef *Status* As Integer)
Sub **VCal** (*CalibName* As String, *Parent* As Form)
Sub **VCal** (*CalibName* As String, *Parent* As Form, ByRef *Status* As Integer)

參數

CalibName	評估目前專案中校準機制之名稱的字串運算式。
Status	選用。接收校準狀態的整數變數。 0：失敗，1：成功
Parent	選用。.NET 父表單。

備註

執行 **VCal** 方法時，機器人將會移動。您應確認操作員是否已在執行 VCal 前完成準備。

VCal 只會執行校準週期，不能讓您示教點。使用 VCalPoints 則可示教點。此外，您必須先在 EPSON RC+ 7.0 中設定校準。如需詳細資訊，請參閱 Vision Guide 手冊。

使用 Status 參數檢查校準是否成功。如果 ShowConfirmation 校準屬性設為 True，則會顯示確認對話方塊。當操作員按一下<OK>按鈕時，狀態傳回 1：成功。

另請參閱

VCalPoints

VCal 範例

VB 例:

```
Dim status As Integer
m_spel.VCal("CAMCAL1", status)
```

C# 例:

```
int status;
m_spel.VCal("CAMCAL1", status);
```

VCalPoints 方法，Spel 類別

描述

此命令可讓您示教視覺校準點。

語法

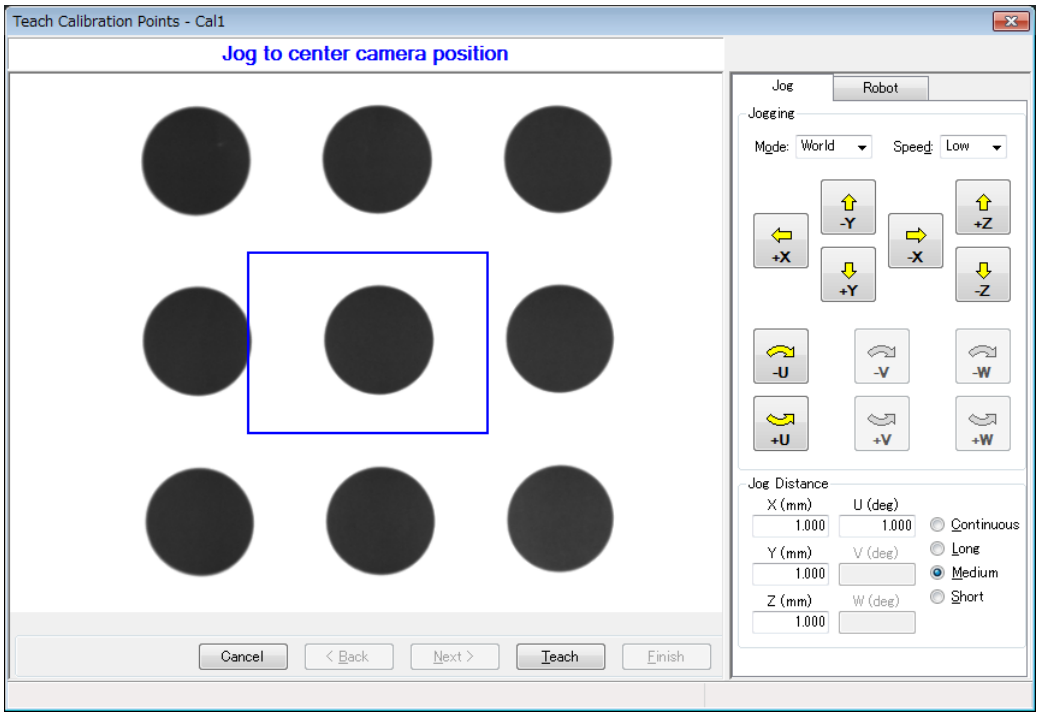
Sub **VCalPoints** (*CalibName* As String)
Sub **VCalPoints** (*CalibName* As String, ByRef *Status* As Integer)
Sub **VCalPoints** (*CalibName* As String, *Parent* As Form)
Sub **VCalPoints** (*CalibName* As String, ByRef *Status* As Integer, *Parent* As Form)

參數

CalibName	評估目前專案中校準機制之名稱的字串運算式。
Status	選用。接收點示教狀態的整數變數。 0：失敗，1：成功
Parent	選用。.NET 父表單。

備註

執行 **VCalPoints** 命令時，示教點校準點對話方塊會開啟。按一下<Finish>按鈕時，校準資料會自動保存。
您必須先從 EPSON RC+ 7.0 建立校準設定。



使用 Status 參數檢查點示教是否成功。當所有點已示教並按一下<Finish>按鈕時，Status 傳回 1：成功。

另請參閱

VCal

VCalPoints 範例

VB 例:

```
Dim status As Integer
m_spel.VCalPoints("CAMCAL1", status)
```

C# 例:

```
int status;
m_spel.VCalPoints("CAMCAL1", out status);
```

VCls 方法，Spel 類別

描述

清除視覺圖形。

語法

Sub **VCls** ()

備註

使用 VCls 方法可清除視覺畫面。

另請參閱

VRun

VCls 範例

VB 例:

```
m_spel.VCls()
```

C# 例:

```
m_spel.VCls();
```

VCreateCalibration 方法，Spel 類別

描述

在目前專案中建立新建視覺校準。

語法

Sub **VCreateCalibration** (*CameraNumber* As Integer, *CalibName* As String)
Sub **VCreateCalibration** (*CameraNumber* As Integer, *CalibName* As String, *CopyCalibName* As String)

參數

CameraNumber 包含要校準之攝影機編號的整數運算式。

CalibName 包含要建立之視覺校準名稱的字串運算式。

CopyCalibName 選用。包含要複製之視覺校準名稱的字串運算式。

另請參閱

VCreateObject, VCreateSequence, VDeleteCalibration

VCreateCalibration 範例

VB 例:

```
m_spel.VCreateCalibration(1, "mycal")
```

C# 例:

```
m_spel.VCreateCalibration(1, "mycal");
```

VCreateObject 方法，Spel 類別

描述

在目前專案中建立視覺物件。

語法

Sub **VCreateObject** (*Sequence* As String, *ObjectName* As String, *ObjectType* As SpelVisionObjectTypes)

參數

Sequence 包含目前專案中視覺序列之名稱的字串運算式。

ObjectName 包含要在 *Sequence* 序列中建立之物件名稱的字串運算式。

ObjectType 指定視覺物件類型的 SpelVisionObjectTypes。
(以下所示的常數也可使用)

物件類型	常數	數值
Correlation	Correlation	1
Blob	Blob	2
Edge	Edge	3
Polar	Polar	4
Line	Line	5
Point	Point	6
Frame	Frame	7
ImageOp	ImageOp	8
Ocr	Ocr	9
CodeReader	CodeReader	10
Geometric	Geometric	11
Color Match	ColorMatch	14
Line Finder	LineFinder	15
Arc Finder	ArcFinder	16
Defect Finder	DefectFinder	17
Line Inspector	LineInspector	18
Arc Inspector	ArcInspector	19
Box Finder	BoxFinder	20
Corner Finder	CornerFinder	21
Contour	Contour	22
Text	Text	23
Decision	Decision	26
Coordinates	Coordinates	27

另請參閱

VCreateSequence, VDeleteObject, VDeleteSequence

VCreateObject 範例

VB 例:

```
m_spel.VCreateObject("myseq", "myblob",
SpelVisionObjectTypes.Blob)
```

C# 例:

```
m_spel.VCreateObject("myseq", "myblob",
SpelVisionObjectTypes.Blob);
```

VCreateSequence 方法，Spel 類別

描述

在目前專案中建立新建視覺序列。

語法

Sub **VCreateSequence** (*CameraNumber* As Integer, *SequenceName* As String)
Sub **VCreateSequence** (*CameraNumber* As Integer, *SequenceName* As String, *CopySequenceName* As String)

參數

CameraNumber	包含要使用之攝影機編號的整數運算式。
SequenceName	包含要建立之視覺序列名稱的字串運算式。
CopySequenceName	選用。包含要複製之視覺序列名稱的字串運算式。

另請參閱

VCreateObject, VDeleteObject, VDeleteSequence

VCreateSequence 範例

VB 例:

```
m_spel.VCreateSequence(1, "myseq")
```

C# 例:

```
m_spel.VCreateSequence(1, "myseq");
```

VDefArm 方法，Spel 類別

描述

使用視覺系統可偵測到的特徵點，計算移動攝影機的手臂設定值。

NOTE：
機器人會根據目標偵測結果自動運作。小心機器人與周邊設備發生干擾。此外，這可用於避開各軸延伸姿態附近的奇點，防止手臂設定期間發生錯誤。

語法

Sub **VDefArm** (*ArmNumber* As Integer, *ArmDefType* As SpelArmDefType, *ArmDefMode* As SpelArmDefMode, *Sequence* As String, *Rotation* As Double, *TargetTolerance* As Double)

Sub **VDefArm** (*ArmNumber* As Integer, *ArmDefType* As SpelArmDefType, *ArmDefMode* As SpelArmDefMode, *Sequence* As String, *Rotation* As Double, *TargetTolerance* As Double, *Parent* As Form)

Sub **VDefArm** (*ArmNumber* As Integer, *ArmDefType* As SpelArmDefType, *ArmDefMode* As SpelArmDefMode, *Sequence* As String, *Rotation* As Double, *TargetTolerance* As Double，*RobotSpeed* As Integer, *RobotAccel* As Integer, *ShowWarning* As SpelVDefShowWarning

Sub **VDefArm** (*ArmNumber* As Integer, *ArmDefType* As SpelArmDefType, *ArmDefMode* As SpelArmDefMode, *Sequence* As String, *Rotation* As Double, *TargetTolerance* As Double，*RobotSpeed* As Integer, *RobotAccel* As Integer, *ShowWarning* As SpelVDefShowWarning*, Parent* As Form)

參數

ArmNumber	整數運算式，包含用以執行手臂設定的手臂編號(1 至 15)。
ArmDefType	包含手臂類型的整數運算式。 J2Camera：計算移動 J2 攝影機影像的中心。
ArmDefMode	包含手臂設定模式的整數運算式。 Rough：執行 rough 手臂設定的模式。 機器人將會以 1 公釐的設定準確度為目標進行移動。 機器人動作會比較小。 Fine：執行 fine 手臂設定的模式。 機器人將隨手臂方向改變大幅移動，提供更高準確度的手臂設定。
Sequence	包含目前專案中視覺序列名稱的字串運算式。
Rotation	包含 rough 手臂設定之旋轉角度(度)的實數運算式。 數值範圍：0 到 45
TargetTolerance	包含像素距離的實數運算式，將視覺偵測結果視為符合目標位置。 數值範圍： 0 至 3 像素
Parent	選用。視窗的父.NET 表單。
RobotSpeed	選用。包含機器人速度(%)的整數變數。 數值範圍：0 至 100 如果省略，請設為「5」。

RobotAccel 選用。包含機器人加速(%)的整數變數。
數值範圍：0 至 99
如果省略，請設為「5」。

ShowWarning 選用。決定是否在 *ArmSetMode* 為 Fine 時顯示警告的整數變數。
Always ：永遠顯示
DependsOnSpeed ：*RobotSpeed* 或 *RobotAccel* 大於 5 時顯示。
None ：不顯示
如果省略，請設為「DependsOnSpeed」。

另請參閱

VDefGetMotionRange, VDefLocal, VDefSetMotionRange, VDefTool, VGoCenter

VDefArm 範例

VB 例:

```
m_spel.VDefArm(1, SpelArmDefType.J2Camera,
SpelArmDefMode.Rough, "myseq", 5, 1)
```

C# 例:

```
m_spel.VDefArm(1, SpelArmDefType.J2Camera,
SpelArmDefMode.Rough, "myseq", 5, 1);
```

VDefGetMotionRange 方法，Spel 類別

描述

取得 VDefTool、VDefArm、VDefLocal 及 VGoCenter 所限制之動作範圍的值。

語法

Sub **VDefGetMotionRange**(ByRef MaxMoveDist As Double, ByRef MaxPoseDiffAngle As Double, ByRef LjmMode As Integer)

參數

MaxMoveDist	代表最大移動距離的實數變數。如果指定 0，範圍不受限制。(0 至 500。預設：200) VDeopfTool、VDefArm、VDefLocal 及 VGoCenter 用於限制範圍。
MaxPoseDiffAngle	代表工具方向(UVW)最大位移角度(度)的實數變數。 如果指定 0，角度不受限制。 這只會影響 VDefLocal。(0 至 180。預設：45 度)
LjmMode	代表 LJM 模式的整數變數。

另請參閱

VDefTool, VDefArm, VDefLocal, VGoCenter, VDefSetMotionRange

VDefGetMotionRange 範例

VB 例:

```
Dim maxMoveDist As Double
Dim maxPoseDiffAngle As Double
Dim ljmMode As Integer
m_spel.VDefGetMotionRange(maxMoveDist, maxPoseDiffAngle,
ljmMode)
```

C# 例:

```
double maxMoveDist, maxPoseDiffAngle;
int ljmMode;
m_spel.VDefGetMotionRange(out maxMoveDist, out
maxPoseDiffAngle, out ljmMode);
```

VDefLocal 方法，Spel 類別

描述

透過移動攝影機偵測放在工作平面上的校準板，並定義與工作平面平行的本地座標。
這也會透過固定攝影機偵測工具末端的使用者工件，並定義與固定攝影機感測器平行的本地平面。

NOTE：
機器人會根據目標偵測結果自動運作。小心機器人與周邊設備發生干擾。此外，這可用於避開各軸延伸姿態附近的奇點，防止本地座標設定期間發生錯誤。

語法

Sub **VDefLocal**(LocalNumber As Integer, LocalDefType As SpelLocalDefType, CalPlateType As SpelCalPlateType, Sequence As String, TargetTolerance As Double, CameraTool As Integer, RefPoint As SpelPoint)

Sub **VDefLocal**(LocalNumber As Integer, LocalDefType As SpelLocalDefType, CalPlateType As SpelCalPlateType, Sequence As String, TargetTolerance As Double, CameraTool As Integer, RefPoint As SpelPoint, Parent As Form)

Sub **VDefLocal**(*LocalNumber* As Integer, *LocalDefType* As SpelLocalDefType, *CalPlateType* As SpelCalPlateType, *Sequence* As String, *TargetTolerance* As Double, *CameraTool* As Integer, *RefPoint* As SpelPoint, *RobotSpeed* As Integer, *RobotAccel* As Integer)

Sub **VDefLocal**(*LocalNumber* As Integer, *LocalDefType* As SpelLocalDefType, *CalPlateType* As SpelCalPlateType, *Sequence* As String, *TargetTolerance* As Double, *CameraTool* As Integer, *RefPoint* As SpelPoint, *RobotSpeed* As Integer, *RobotAccel* As Integer, *Parent* As Form)

參數

LocalNumber　代表用以設定本地座標之工具編號的整數。 (1-15)

LocalDefType　代表本地類型的整數。

J5Camera：使用移動 J5 攝影機指定與校準板平行的本地座標。

J6Camera：使用移動 J6 攝影機指定與校準板平行的本地座標。

FixedUpwardCamera：使用上方固定攝影機指定與影像感測器平行的本地座標。

FixedDownwardCamera：使用下方固定攝影機指定向下影像感測器平行的本地座標。

CalPlateType　代表校準板類型的整數。

Large　：大校準板

Medium　：中校準板

Small　：小校準板

XSmall　：超小校準板

Sequence　代表目前專案中視覺序列名稱的字串運算式。

使用移動攝影機時，此為拍攝校準板相片的視覺序列。
使用固定攝影機時，此為偵測工具末端特徵點(例如使用者工件)的視覺序列。

TargetTolerance　代表刻度符合判定閾值的實際值。

CameraTool	固定攝影機：指定保持偵測目標工具偏移的工具編號。若要執行自動校準，請指定-1。 移動 J6 攝影機：如果已執行自動校準，請指定移動攝影機的工具編號。 若要執行自動校準，請指定-1。 移動 J5 攝影機：此選購件的設定會略過。
RefPoint	平行於工作平面之本地平面通過的點編號。 這個點用於指定本地平面高度。
Parent	選用。視窗的父.NET 表單。
RobotSpeed	選用。包含機器人速度(%)的整數變數。 數值範圍：0 至 100 如果省略，請設為「5」。
RobotAccel	選用。包含機器人加速(%)的整數變數。 數值範圍：0 至 99 如果省略，請設為「5」。

另請參閱

VDefArm, VDefGetMotionRange,VDefSetMotionRange, VDefTool, VGoCenter

VDefLocal 範例

VB 例:

```
Dim p2 = m_spel.GetPoint("P2")
m_spel.VDefLocal(1, SpelLocalDefType.J6Camera,
SpelCalPlateType.Large, "myseq", 1.0, 1, p2)
```

C# 例:

```
SpelPoint p2;
p2 = m_spel.GetPoint("P2");
m_spel.VDefLocal(1, SpelLocalDefType.J6Camera,
SpelCalPlateType.Large, "myseq", 1.0, 1, p2);
```

VDefSetMotionRange 方法，Spel 類別

描述

由 VDefTool、VDefArm、VDefLocal 及 VGoCenter 限制動作範圍。

語法

Sub **VDefSetMotionRange**(MaxMoveDist As Double, MaxPoseDiffAngle As Double, LjmMode As Integer)

參數

MaxMoveDist	代表最大移動距離的實數值。 如果指定 0，範圍不受限制。(0 至 500。預設：200) VDefTool、VDefArm、VDefLocal 及 VGoCenter 用於限制範圍。
MaxPoseDiffAngle	代表工具方向(UVW)最大位移角度(度)的實數值。 如果指定 0，角度不受限制。 這只會影響 VDefLocal。(0 至 180。預設：45 度)
LjmMode	代表 LJM 模式的整數。

另請參閱

VDefTool, VDefArm, VDefLocal, VGoCenter, VDefGetMotionRange

VDefSetMotionRange 範例

VB 例:

```
m_spel.VDefSetMotionRange(100, 30, 1)
```

C# 例:

```
m_spel.VDefSetMotionRange(100, 30, 1);
```

VDefTool 方法，Spel 類別

描述

使用視覺偵測，計算 TPC 和移動攝影機位置的工具偏移值。

NOTE：
當工具為 FixedCameraWithCal 以外的類型時，機器人會根據目標偵測結果自動運作。小心機器人與周邊設備發生干擾。此外，這可用於避開各軸延伸姿態附近的奇點，防止工具設定期間發生錯誤。

語法

Sub **VDefTool**(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, Object As String)
Sub **VDefTool**(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, Object As String, Parent As Form)
Sub **VDefTool**(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double)
Sub **VDefTool**(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, Parent As Form)
Sub **VDefTool**(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer)
Sub **VDefTool**(ToolNumber As Integer, ToolDefType As SpelToolDefType, Sequence As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer, Parent As Form)

參數

ToolNumber	代表用以執行工具設定(1-15)之工具編號的整數
ToolDefType	代表工具類型的整數。 FixedCamera：透過使用未校準之固定攝影機所設定的工具。 J4Camera：計算移動 J4 攝影機的影像中心。 J6Camera：計算移動 J6 攝影機的影像中心。 FixedCameraWithCal：透過使用已校準之固定攝影機所設定的工具。
Sequence	代表目前專案中視覺序列之名稱的字串運算式。
Object	代表指定序列中視覺物件的字串運算式。當 *ToolDefType* 設為 *FixedCameraWithCal* 時，需要此參數。當 *ToolDefType* 不是 FixedCameraWithCal 時，物件應為空字串。
FinalAngle	代表旋轉工具或攝影機工具角度(度)的實際值。 數值範圍：0、5 至 180、–5 至–180 如果省略，請設為「90」。
InitAngle	代表暫時性工具設定中旋轉工具或攝影機工具角度(度)的實際值。 此值必須小於 *FinalAngle*。 數值範圍：-10 至 10 如果省略，請設為「5」。
TargetTolerance	代表像素距離的實際值，將視覺偵測結果視為符合目標位置。 數值範圍：0 至 3 像素 如果省略，請設為「1」。

Parent	選用。視窗的父.NET 表單。
RobotSpeed	選用。包含機器人速度(%)的整數變數。 數值範圍：0 至 100 如果省略，請設為「5」。
RobotAccel	選用。包含機器人加速(%)的整數變數。 數值範圍：0 至 99 如果省略，請設為「5」。

另請參閱

VDefArm, VDefGetMotionRange, VDefLocal, VDefSetMotionRange, VGoCenter

VDefTool 範例

VB 例:

```
m_spel.VDefTool(1, SpelToolDefType.J6Camera, "myseq", 45, 5,
3.0)
m_spel.VDefTool(1, SpelToolDefType.FixedCameraWithCal,
"myseq", "myobj")
```

C# 例:

```
m_spel.VDefTool(1, SpelToolDefType.J6Camera, "myseq", 45, 5,
3.0);
m_spel.VDefTool(1, SpelToolDefType.FixedCameraWithCal,
"myseq", "myobj");
```

VDefToolXYZ 方法, Spel 類別

描述

VDefToolXYZ 使用視覺檢測來計算工具偏移值(XYZ)。

注意：

根據目標檢測結果，機器人會自動工作。注意機器人和週邊設備之間的干擾。此外，為了避免工具偏移導致的錯誤，請避免使用每個關節延伸的奇點附近的姿勢。

語法

Sub **VDefToolXYZ**(ToolNumber As Integer, LocalNumber As Integer, PointNumber1 As Integer, PointNumber2 As Integer, Sequence1 As String, Sequence2 As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer)

Sub **VDefToolXYZ**(ToolNumber As Integer, LocalNumber As Integer, PointNumber1 As Integer, PointNumber2 As Integer, Sequence1 As String,Sequence2 As String, FinalAngle As Double, InitAngle As Double, TargetTolerance As Double, RobotSpeed As Integer, RobotAccel As Integer, Parent As Form)

參數

ToolNumber	代表執行工具設置值工具編號的整數 (1~15)
LocalNumber	代表歐得拍移動機器人的本地坐標編號的整數。工具會移動到指定的本地坐標的 XY 平面。
PointNumber1	代表 Point Number 第一姿勢的整數
PointNumber2	代表 Point Number 第二姿勢的整數
Sequence1	代表當前專案中第一姿勢的視覺序列名稱的字串運算式
Sequence2	代表當前專案中第二姿勢的視覺序列名稱的字串運算式
FinalAngle	代表工具與攝影機工具旋轉角度(度)的實數 數值範圍：5 ~ 180, □5 ~ □180
InitAngle	代表臨時工具設定時的旋轉角度(度)的實數 該值必須小於 FinalAngle 數值範圍：0.01 ~ 10, □0.01 ~ □10
TargetTolerance	代表視覺檢測結果與物件位置一致的閾值像素的實數 數值範圍： 0.1 ~ 3.0 pixel
RobotSpeed	代表機器人速度 (%)的整數變數 數值範圍：1 ~ 100
RobotAccel	代表機器人加速度 (%)的整數變數 數值範圍： 1 ~ 99
Parent	.成為視窗父項的 NET 表單 (可以省略)

另請參閱

VDefTool, VDefToolXYZUVW

VDefToolXYZ 範例

VB 例:

```
m_spel.VDefToolXYZ(1, 0, 1, 2, "seq01", "seq02", 5, 30, 1, 5,
5)
m_spel.VDefToolXYZ(2, 0, 3, 4, "seq03", "seq04", 5, 30, 1, 5,
5)
m_spel.VDefToolXYZ(3, 0, 5, 6, "seq05", "seq06", 5, 30, 1, 5,
5)
m_spel.VDefToolXYZUVW(1, 2, 3, SpelToolDefType3D.Bar)
```

C# 例:

```
m_spel.VDefToolXYZ(1, 0, 1, 2, "seq01", "seq02", 5, 30, 1, 5,
5);
m_spel.VDefToolXYZ(2, 0, 3, 4, "seq03", "seq04", 5, 30, 1, 5,
5);
m_spel.VDefToolXYZ(3, 0, 5, 6, "seq05", "seq06", 5, 30, 1, 5,
5);
m_spel.VDefToolXYZUVW(1, 2, 3, SpelToolDefType3D.Bar);
```

VDefToolXYZUVW 方法, Spel 類別

描述

用 3 个工具定义來計算工具 UVW 偏移值。

注意：
計算出的 U, V, W 工具偏移值在指定的工具 1 中設置。工具 1 的 X, Y, Z 偏移不變。

語法

Sub **VDefToolXYZUVW**(ToolNumber1 As Integer, ToolNumber2 As Integer, ToolNumber3 As Integer, ToolDefType3D As SpelToolDefType3D)

參數

ToolNumber1	代表第 1 個工具定義的工具編號的整數變量 (1 ~ 15) 條形類型指定工具尖端的工具編號。 平面類型指定工具中心的工具編號。
ToolNumber2	代表第 2 個工具定義的工具編號的整數變量(1 ~ 15) 條形類型指定工具中央處的工具編號。 平面類型指定工具中心以外，與 ToolNumber3 不同的工具編號。
ToolNumber3	代表第 3 個工具定義的工具編號的整數變量(1 ~ 15) 條形類型指定工具末端的工具編號。 平面類型指定工具中心以外，與 ToolNumber2 不同的工具編號。
ToolDefType3D	代表工具定義的工具類型的整數變量 Bar: 條形類型 Plane: 平面類型

另請參閱

VDefToolXYZ

VDefToolXYZUVW 範例

VB 例:

```
m_spel.VDefToolXYZ(1, 0, 1, 2, "seq01", "seq02", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZ(2, 0, 3, 4, "seq03", "seq04", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZ(3, 0, 5, 6, "seq05", "seq06", 5, 30, 1, 5, 5)
m_spel.VDefToolXYZUVW(1, 2, 3, SpelToolDefType3D.Bar)
```

C# 例:

```
m_spel.VDefToolXYZ(1, 0, 1, 2, "seq01", "seq02", 5, 30, 1, 5, 5);
m_spel.VDefToolXYZ(2, 0, 3, 4, "seq03", "seq04", 5, 30, 1, 5, 5);
m_spel.VDefToolXYZ(3, 0, 5, 6, "seq05", "seq06", 5, 30, 1, 5, 5);
m_spel.VDefToolXYZUVW(1, 2, 3, SpelToolDefType3D.Bar);
```

VDeleteCalibration 方法，Spel 類別

描述

在目前專案中刪除視覺校準。

語法

Sub **VDeleteCalibration** (*CalibName* As String)

參數

CalibName　包含目前專案中視覺校準之名稱的字串運算式。

另請參閱

VCreateCalibration, VDeleteObject, VDeleteSequence

VDeleteCalibration 範例

VB 例:

```
m_spel.VDeleteCalibration("mycal")
```

C# 例:

```
m_spel.VDeleteCalibration("mycal");
```

VDeleteObject 方法，Spel 類別

描述

在目前專案中刪除視覺物件。

語法

Sub **VDeleteObject** (*Sequence* As String, *ObjectName* As String)

參數

Sequence 包含目前專案中視覺序列之名稱的字串運算式。

ObjectName 包含目前專案中視覺物件之名稱的字串運算式。

另請參閱

VCreateObject, VCreateSequence, VDeleteSequence

VDeleteObject 範例

VB 例:

```
m_spel.VDeleteObject("myseq", "myobj")
```

C# 例:

```
m_spel.VDeleteObject("myseq", "myobj");
```

VDeleteSequence 方法，Spel 類別

描述

在目前專案中刪除視覺序列。

語法

Sub **VDeleteSequence** (*Sequence* As String)

參數

Sequence　包含目前專案中視覺序列之名稱的字串運算式。

另請參閱

VCreateObject, VCreateSequence, VDeleteObject

VDeleteSequence 範例

VB 例:

```
m_spel.VDeleteSequence("myseq")
```

C# 例:

```
m_spel.VDeleteSequence("myseq");
```

VEditWindow 方法, Spel 類別

描述

編輯搜尋視窗中的不要緊的圖元。更多詳細資訊，請參閱 Vision Guide Properties and Results Reference 手冊的“EditWindow 屬性”。

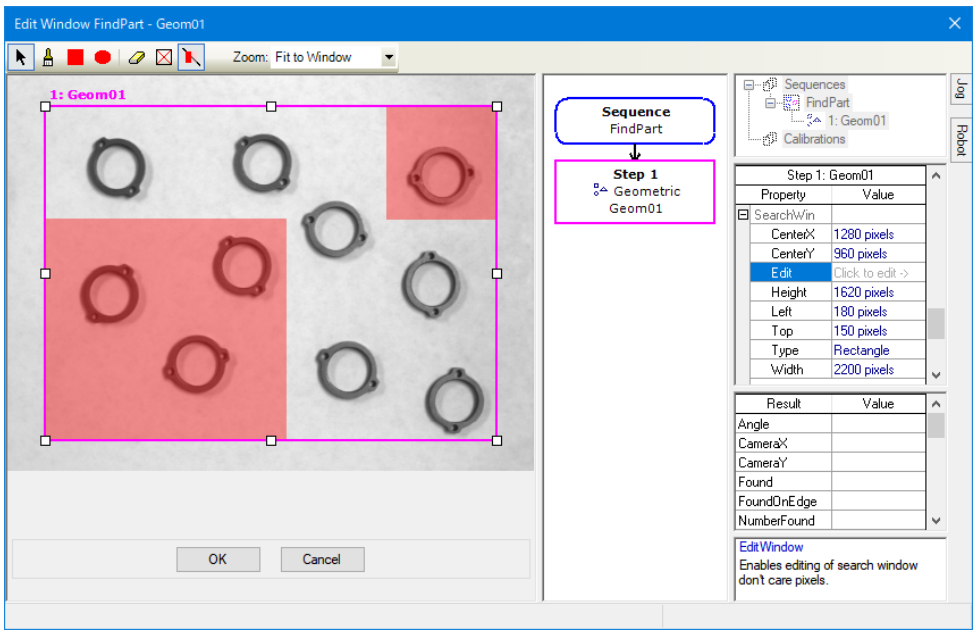
語法

Sub **VEditWindow** (Sequence As String, Object As String)

參數

Sequence　代表當前專案的視覺序列名稱的字串運算式

ObjectName　代表當前專案的視覺物件名稱的字串運算式



另請參閱

VSave

VEditWindow 範例

VB 例:

```
m_spel.VEditWindow("myseq", "myobj")
```

C# 例:

```
m_spel.VEditWindow("myseq", "myobj");
```

VGet 方法，Spel 類別

描述

取得視覺序列或物件屬性或結果的值。

語法

Sub **VGet** (*Sequence* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Integer)
Sub **VGet** (*Sequence* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Boolean)
Sub **VGet** (*Sequence* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Double)
Sub **VGet** (*Sequence* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Single)
Sub **VGet** (*Sequence* As String, *PropCode* As SpelVisionProps, ByRef *Value* As String)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Integer)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Boolean)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Color)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Double)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, ByRef *Value* As Single)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, ByRef *Value* As String)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Result* As Integer, ByRef *Value* As Integer)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Result* As Integer, ByRef *Value* As Boolean)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Result* As Integer, ByRef *Value* As Double)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Result* As Integer, ByRef *Value* As Single)
Sub **VGet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Result* As Integer, ByRef *Value* As String)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。若屬性用於序列，則此字串必須為空白。
PropCode	指定屬性代碼的 SpelVisionProps 值。
Result	代表結果編號的整數運算式。
Value	包含屬性或結果值的變數。變數的類型必須符合屬性或結果類型。

另請參閱

VSet, VRun

VGet 範例

VB 例:

```
Dim i As Integer
Redim score(10) As Integer

m_spel.VRun("testSeq")
For i = 1 to 10
    m_spel.VGet("testSeq", "corr" & Format$(i, "00"), _
        SpelVisionProps.Score, score(i))
Next i
```

C# 例:

```
int[] score = new int[11];
for(int i = 1; i <= 10; i++)
{
   m_spel.VGet("testSeq", string.Format("Corr 0{0}, i),
           SpelVisionProps.Score, score(i));
}
```

VGetCameraXYU 方法，Spel 類別

描述

擷取任何物件的攝影機 X、Y 及 U 實體座標。

語法

Sub **VGetCameraXYU** (*Sequence* As String, *Object* As String, *Result* As Integer, ByRef *Found* As Boolean, ByRef *X* As Single, ByRef *Y* As Single, ByRef *U* As Single)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Result	代表結果編號的整數運算式。
Found	包含是否找到物件的 Boolean 變數。
X	包含 x 座標的實際變數(以公釐為單位)。
Y	包含 y 座標的實際變數(以公釐為單位)。
U	包含角度的實際變數(以度為單位)。

另請參閱

VGetPixelXYU, VGetRobotXYU

VGetCameraXYU 範例

VB 例:

```
Dim found As Boolean
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String

seq = "testSeq"
blob = "blob01"
m_spel.VRun(seq)
m_spel.VGetCameraXYU(seq, blob, 1, found, x, y, u)
```

C# 例:

```
bool found;
float x, y, u;
string seq, blob;

seq = "testSeq";
blob = "blob01";
m_spel.VRun(seq);
m_spel.VGetCameraXYU(seq, blob, 1, out found, out x, out y,
out u);
```

VGetEdgeCameraXYU 方法，Spel 類別

描述

針對 Line Finder、Arc Finder 搜尋的每個邊緣，擷取攝影機 X、Y 及 U 實體座標。

語法

Sub **VGetEdgeCameraXYU** (*Sequence* As String, *Object* As String, *EdgeResultIndex* As Integer, ByRef *Found* As Boolean, ByRef *X* As Single, ByRef *Y* As Single, ByRef *U* As Single)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
EdgeResultIndex	代表邊緣結果索引的整數運算式。
Found	包含是否找到物件的 Boolean 變數。
X	包含 x 座標的實際變數(以公釐為單位)。
Y	包含 y 座標的實際變數(以公釐為單位)。
U	包含角度的實際變數(以度為單位)。

另請參閱

VGetEdgePixelXYU, VGetEdgeRobotXYU, VGetPixelXYU, VGetRobotXYU

VGetEdgeCameraXYU 範例

VB 例:

```
Dim found(10) As Boolean
Dim x(10) As Single, y(10) As Single, u(10) As Single
Dim seq As String, lineFinder As String

seq = "testSeq"
lineFinder = "LineFind01"
m_spel.VRun(seq)
' LineFinder 的 NumberOfEdges 為 10
For i = 1 To 10
  m_spel.VGetEdgeCameraXYU(seq, lineFinder, i, found(i),
x(i),
     y(i), u(i))
Next i
```

C# 例:

```
bool[] found = new bool[11];
float[] x = new float[11];
float[] y = new float[11];
float[] u = new float[11];
string seq, lineFinder;
seq = "testSeq";
lineFinder = "LineFind01";
m_spel.VRun(seq);
// LineFinder 的 NumberOfEdges 為 10
for(int i = 1; i <= 10; i++)
m_spel.VGetEdgeCameraXYU(seq, lineFinder, i, out found[i],
      out x[i], out y[i], out u[i]);
```

VGetEdgePixelXYU 方法，Spel 類別

描述

針對 Line Finder、Arc Finder 搜尋的每個邊緣，擷取 X、Y 及 U 像素座標。

語法

Sub **VGetEdgePixelXYU** (*Sequence* As String, *Object* As String, *EdgeResultIndex* As Integer, ByRef *Found* As Boolean, ByRef *X* As Single, ByRef *Y* As Single, ByRef *U* As Single)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
EdgeResultIndex	代表邊緣結果索引的整數運算式。
Found	包含是否找到物件的 Boolean 變數。
X	包含 x 座標的實際變數(以公釐為單位)。
Y	包含 y 座標的實際變數(以公釐為單位)。
U	包含角度的實際變數(以度為單位)。

另請參閱

VGetEdgeCameraXYU, VGetEdgeRobotXYU, VGetPixelXYU, VGetRobotXYU

VGetEdgePixelXYU 範例

VB 例:

```
Dim found(10) As Boolean
Dim x(10) As Single, y(10) As Single, u(10) As Single
Dim seq As String, lineFinder As String

seq = "testSeq"
lineFinder = "LineFind01"
m_spel.VRun(seq)
' LineFinder 的 NumberOfEdges 為 10
For i = 1 To 10
  m_spel.VGetEdgePixelXYU(seq, lineFinder, i, found(i),
x(i),
     y(i), u(i))
Next i
```

C# 例:

```
bool[] found = new bool[11];
float[] x = new float[11];
float[] y = new float[11];
float[] u = new float[11];
string seq, lineFinder;
seq = "testSeq";
lineFinder = "LineFind01";
m_spel.VRun(seq);
// LineFinder 的 NumberOfEdges 為 10
for(int i = 1; i <= 10; i++)
    m_spel.VGetEdgePixelXYU(seq, lineFinder, i, out found[i],
          out x[i], out y[i], out u[i]);
```

VGetEdgeRobotXYU 方法，Spel 類別

描述

針對 Line Finder、Arc Finder 搜尋的每個邊緣，擷取機器人 X、Y 及 U 實體座標。

語法

Sub **VGetEdgeRobotXYU** (*Sequence* As String, *Object* As String, *EdgeResultIndex* As Integer, ByRef *Found* As Boolean, ByRef *X* As Single, ByRef *Y* As Single, ByRef *U* As Single)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
EdgeResultIndex	代表邊緣結果索引的整數運算式。
Found	包含是否找到物件的 Boolean 變數。
X	包含 x 座標的實際變數(以公釐為單位)。
Y	包含 y 座標的實際變數(以公釐為單位)。
U	包含角度的實際變數(以度為單位)。

另請參閱

VGetEdgeCameraXYU, VGetEdgePixelXYU, VGetPixelXYU, VGetRobotXYU

VGetEdgeRobotXYU 範例

VB 例:

```
Dim found(10) As Boolean
Dim x(10) As Single, y(10) As Single, u(10) As Single
Dim seq As String, lineFinder As String

seq = "testSeq"
lineFinder = "LineFind01"
m_spel.VRun(seq)
' LineFinder 的 NumberOfEdges 為 10
For i = 1 To 10
  m_spel.VGetEdgeRobotXYU(seq, lineFinder, i, found(i),
x(i),
     y(i), u(i))
Next i
```

C# 例:

```
bool[] found = new bool[11];
float[] x = new float[11];
float[] y = new float[11];
float[] u = new float[11];
string seq, lineFinder;
seq = "testSeq";
lineFinder = "LineFind01";
m_spel.VRun(seq);
// LineFinder 的 NumberOfEdges 為 10
for(int i = 1; i <= 10; i++)
    m_spel.VGetEdgeRobotXYU(seq, lineFinder, i, out found[i],
      out x[i], out y[i], out u[i]);
```

VGetExtrema 方法，Spel 類別

描述

擷取 blob 物件的極值座標。

語法

Sub **VGetExtrema** (*Sequence* As String, *Object* As String, *Result* As Integer，ByRef *MinX* As Single, ByRef *MaxX* As Single, ByRef *MinY* As Single, ByRef *MaxY* As Single)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Result	代表結果編號的整數運算式。
MinX	包含最小 x 座標的實際變數(以像素為單位)。
MaxX	包含最大 x 座標的實際變數(以像素為單位)。
MinY	包含最小 y 座標的實際變數(以像素為單位)。
MaxY	包含最大 y 座標的實際變數(以像素為單位)。

另請參閱

VGet

VGetExtrema 範例

VB 例:

```
Dim xmin As Single, xmax As Single
Dim ymin As Single, ymax As Single
Dim seq As String, blob As String

seq = "testSeq"
blob = "blob01"
m_spel.VRun(seq)
m_spel.VGet(seq, blob, "found", found)
If found <> 0 Then
    m_spel.VGetExtrema(seq, blob, xmin, xmax, ymin, ymax)
End If
```

C# 例:

```
float xMin, xMax, yMin, yMax;
bool found;
string seq, blob;

seq = "testSeq";
blob = "blob01";
m_spel.VRun(seq);
m_spel.VGet(seq, blob, "found", found);

if(found == true)
    m_spel.VGetExtrema(seq, blob,out xMin,out xMax,out yMin,
        out yMax);
```

VGetModelWin 方法，Spel 類別

描述

擷取物件的模型視窗座標。

語法

Sub **VGetModelWin** (*Sequence* As String, *Object* As String, ByRef *Left* As Integer, ByRef *Top* As Integer, ByRef *Width* As Integer, ByRef *Height* As Integer)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Left	包含左側座標的整數變數(以像素為單位)。
Top	包含頂端座標的整數變數(以像素為單位)。
Width	包含寬度的整數變數(以像素為單位)。
Height	包含高度的整數變數(以像素為單位)。

另請參閱

VSetModelWin, VGetSearchWin, VSetSearchWin

VGetModelWin 範例

VB 例:

```
Dim left As Integer, top As Integer
Dim width As Integer, height As Integer

With m_spel
    .VGetModelWin("testSeq", "corr01", left, top, _
       width, height)
    .VSetModelWin("testSeq", "corr01", left + 20, top, _
       width, height)
    .VTeach("testSeq", "corr01")
End With
```

C# 例:

```
int left, top, width, height;

m_spel.VGetModelWin("testSeq", "corr01", out left, out top,
       out width, out height);
m_spel.VSetModelWin("testSeq", "corr01", left + 20, top,
       width, height);
m_spel.VTeach("testSeq", "corr01");
```

VGetPixelToCamera 方法, Spel 類別

描述

擷取指定圖元坐標的攝影機坐標。

語法

Sub **VGetPixelToCamera** (Calibration As String, PixelX As Single, PixelY As Single, Angle As Single, ByRef CameraX As Single, ByRef CameraY As Single, ByRef CameraU As Single)

參數

Calibration	代表目前專案中校準名稱的字串運算式
PixelX	代表以像素為單位的 x 坐標的實數變量
PixelY	代表以像素為單位的 y 坐標的實數變量
Angle	代表以度為單位的角度的實數變量
CameraX	代表以釐米為單位的 x 坐標的實數變量
CameraY	代表以釐米為單位的 y 坐標的實數變量
CameraU	代表以度為單位的角度的實數變量

另請參閱

VGetPixelXYU, VGetCameraXYU, VGetPixelToRobot

VGetPixelToCamera 範例

VB 例:

```
Dim x As Single, y As Single, u As Single
Dim cal As String, seq As String

cal = "testCal"
seq = "testSeq"
m_spel.VRun(seq)
m_spel.VGetPixelToCamera(cal, 400, 300, 0, x, y, u)
```

C# 例:

```
float x, y, u
string cal, seq;

cal = "testCal";
seq = "testSeq";
m_spel.VRun(seq);
m_spel.VGetPixelToCamera(cal, 400, 300, 0, out x, out y, out
u);
```

VGetPixelToRobot 方法, Spel 類別

描述

擷取指定圖元坐標的機器人坐標。

語法

Sub **VGetPixelToRobot** (Calibration As String, PixelX As Single, PixelY As Single, Angle As Single, ByRef RobotX As Single, ByRef RobotY As Single, ByRef RobotU As Single)

參數

Calibration	代表目前專案中校準名稱的字串運算式
PixelX	代表以像素為單位的 x 坐標的實數變量
PixelY	代表以像素為單位的 y 坐標的實數變量
Angle	代表以度為單位的角度的實數變量
CameraX	代表以釐米為單位的 x 坐標的實數變量
CameraY	代表以釐米為單位的 y 坐標的實數變量
CameraU	代表以度為單位的角度的實數變量

另請參閱

VGetPixelXYU, VGetRobotXYU, VGetPixelToCamera

VGetPixelToRobot 範例

VB 例:

```
Dim x As Single, y As Single, u As Single
Dim cal As String, seq As String

cal = "testCal"
seq = "testSeq"
m_spel.VRun(seq)
m_spel.VGetPixelToRobot(cal, 400, 300, 0, x, y, u)
```

C# 例:

```
float x, y, u
string cal, seq;

cal = "testCal";
seq = "testSeq";
m_spel.VRun(seq);
m_spel.VGetPixelToRobot(cal, 400, 300, 0, out x, out y, out
u);
```

VGetPixelXYU 方法，Spel 類別

描述

擷取任何物件的像素 X、Y 及 U 座標。

語法

Sub **VGetPixelXYU** (*Sequence* As String, *Object* As String, *Result* As Integer，ByRef *Found* As Boolean, ByRef *X* As Single, ByRef *Y* As Single, ByRef *U* As Single)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Result	代表結果編號的整數運算式。
Found	包含是否找到物件的 Boolean 變數。
X	包含 x 座標的實際變數(以像素為單位)。
Y	包含 y 座標的實際變數(以像素為單位)。
U	包含角度的實際變數(以度為單位)。

另請參閱

VGetCameraXYU, VGetRobotXYU

VGetPixelXYU 範例

VB 例:

```
Dim found As Integer
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String

seq = "testSeq"
blob = "blob01"
m_spel.VRun(seq)
m_spel.VGetPixelXYU(seq, blob, 1, found, x, y, u)
```

C# 例:

```
int found;
float x, y, u;
string seq, blob;

seq = "testSeq";
blob = "blob01";
m_spel.VRun(seq);
m_spel.VGetPixelXYU(seq, blob,1, out found, out x, out y,
out u);
```

VGetRobotPlacePos 方法，Spel 類別

描述

擷取機器人放置位置。

語法

Sub **VGetRobotPlacePos** (*Sequence* As String, *Object* As String, *Result* As Integer，
ByRef *Found* As Boolean, ByRef *PlacePoint*As SpelPoint)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Result	代表結果編號的整數運算式。
Found	包含已找到布林狀態的整數變數。若發現 false，表示 *x*、*y* 及 *u* 尚未定義。
PlacePoint	包含放置位置的 SpelPoint 變數

另請參閱

VGetRobotPlaceTargetPos, VSetRobotPlaceTargetPos

VGetRobotPlacePos 範例

VB 例:

```
Dim found As Integer
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String
Dim placePoint As SpelPoint

seq = "testSeq"
blob = "blob01"
' 移動高於上方攝影機的工件
m_spel.Jump("camPos")
m_spel.VRun(seq)
m_spel.VGetRobotPlacePos(seq, blob, 1, found, placePoint)
' 使用 SCARA 以使用+TLZ 接近
m_spel.Jump(placePoint, "+TLZ(10)")
m_spel.Go(placePoint)
```

C# 例:

```
bool found;
float x, y, u;
string seq, blob;
SpelPoint placePoint = new SpelPoint();
seq = "testSeq";
blob = "blob01";

// 移動高於上方攝影機的工件
m_spel.Jump("camPos");
m_spel.VRun(seq);
m_spel.VGetRobotPlacePos(seq,blob,1,out found, out
placePoint);
// 使用 SCARA 以使用+TLZ 接近
m_spel.Jump(placePoint, "+TLZ(10)");
m_spel.Go(placePoint);
```

VGetRobotPlaceTargetPos 方法，Spel 類別

描述

擷取工件放置位置。

語法

Sub **VGetRobotPlaceTargetPos** (*Sequence* As String, *Object* As String, ByRef *Point* As SpelPoint)

參數

Sequence 包含目前專案中視覺序列之名稱的字串運算式。

Object 包含 *Sequence* 序列中物件之名稱的字串運算式。

Point 包含放置位置的 SpelPoint 變數

另請參閱

VGetRobotPlacePos, VSetRobotPlaceTargetPos

VGetRobotPlaceTargetPos 範例

VB 例:

```
Dim seq As String, blob As String
Dim targetPoint As SpelPoint

seq = "testSeq"
blob = "blob01"
m_spel.VGetRobotPlaceTargetPos(seq, blob, targetPoint)

' 調整放置位置
targetPoint.X = targetPoint.X + 10
m_spel.VSetRobotPlaceTargetPos(seq, blob, targetPoint)
```

C# 例:

```
string seq, blob;
SpelPoint targetPoint = new SpelPoint();

seq = "testSeq";
blob = "blob01";
m_spel.VGetRobotPlaceTargetPos(seq, blob, out targetPoint);

// 調整放置位置
targetPoint.X = targetPoint.X + 10;
m_spel.VSetRobotPlaceTargetPos(seq, blob, targetPoint);
```

VGetRobotXYU 方法，Spel 類別

描述

擷取任何物件的機器人預設 X、Y 及 U 座標。

語法

Sub **VGetRobotXYU** (*Sequence* As String, *Object* As String, *Result* As Integer，ByRef *Found* As Boolean, ByRef *X* As Single, ByRef *Y* As Single, ByRef *U* As Single)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Result	代表結果編號的整數運算式。
Found	包含已找到布林狀態的整數變數。若發現 false，表示 *x*、*y* 及 *u* 尚未定義。
X	包含 x 座標的實際變數(以公釐為單位)。
Y	包含 y 座標的實際變數(以公釐為單位)。
U	包含角度的實際變數(以度為單位)。

另請參閱

VGetCameraXYU, VGetPixelXYU

VGetRobotXYU 範例

VB 例:

```
Dim found As Integer
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String

seq = "testSeq"
blob = "blob01"
m_spel.VRun(seq)
m_spel.VGetRobotXYU(seq, blob, 1, found, x, y, u)
```

C# 例:

```
bool found;
float x, y, u;
string seq, blob;

seq = "testSeq";
blob = "blob01";
m_spel.VRun(seq);
m_spel.VGetRobotXYU(seq, blob, 1, out found, out x, out y,
out u);
```

VGetRobotToolXYU 方法，Spel 類別

描述

擷取工具定義的機器人預設 X、Y 及 U 數值。

語法

Sub **VGetRobotToolXYU** (*Sequence* As String, *Object* As String, *Result* As Integer，
ByRef *Found* As Boolean, ByRef *X* As Single, ByRef *Y* As Single, ByRef *U* As Single)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Result	代表結果編號的整數運算式。
Found	包含已找到布林狀態的整數變數。若發現 false，表示 *x*、*y* 及 *u* 尚未定義。
X	包含 x 座標的實際變數(以公釐為單位)。
Y	包含 y 座標的實際變數(以公釐為單位)。
U	包含角度的實際變數(以度為單位)。

備註

使用 VGetRobotToolXYU 可針對透過上方攝影機檢視的工件輕鬆定義工具。 這可讓您拾取工件、在上方攝影機 FOV 中進行搜尋、定義工件的工具，然後放置工件。

另請參閱

VGetCameraXYU, VGetPixelXYU, VGetRobotXYU

VGetRobotToolXYU 範例

VB 例:

```
Dim found As Integer
Dim x As Single, y As Single, u As Single
Dim seq As String, blob As String

seq = "testSeq"
blob = "blob01"
' 移動高於上方攝影機的工件
m_spel.Jump("camPos")
m_spel.VRun(seq)
m_spel.VGetRobotToolXYU(seq, blob, 1, found, x, y, u)
m_spel.TLSet(1, x, y, u)
```

C# 例:

```
bool fnd;
float x, y, u;
string seq, blob;

seq = "testSeq";
blob = "blob01";
// 移動高於上方攝影機的工件
m_spel.Jump("camPos");
m_spel.VRun(seq);
m_spel.VGetRobotToolXYU(seq, blob, 1, out fnd, out x, out y,
out u);
m_spel.TLSet(1, x, y, u);
```

VGetSearchWIn 方法，Spel 類別

描述

擷取搜尋視窗座標。

語法

Sub **VGetSearchWin** (*Sequence* As String, *Object* As String, ByRef *Left* As Integer, ByRef *Top* As Integer, ByRef *Width* As Integer, ByRef *Height* As Integer)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Left	包含左側座標的整數變數(以像素為單位)。
Top	包含頂端座標的整數變數(以像素為單位)。
Width	包含寬度的整數變數(以像素為單位)。
Height	包含高度的整數變數(以像素為單位)。

另請參閱

VGetModelWin, VSetModelWin, VSetSearchWin

VGetSearchWin 範例

VB 例:

```
Dim left As Integer, top As Integer
Dim width As Integer, height As Integer

With m_spel
    .VGetSearchWin("testSeq", "corr01", left, top, _
       width, height)
    .VSetSearchWin("testSeq", "corr01", newLeft, top, _
       width, height)
    .VRun("testSeq")
End With
```

C# 例:

```
int left, top, width, height;

m_spel.VGetSearchWin("testSeq", "corr01", out left, out top,
       out width, out height);
m_spel.VSetSearchWin("testSeq", "corr01", newLeft, top,
       width, height);
m_spel.VRun("testSeq");
```

VGoCenter 方法，Spel 類別

描述

使用可由視覺系統偵測的特徵點時，將機器人移至特徵點位於攝影機影像中央的位置。

語法

Sub **VGoCenter**(Sequence As String, LocalNumber As Integer, TargetTolerance As Double)
Sub **VGoCenter**(Sequence As String, LocalNumber As Integer, TargetTolerance As Double, Parent As Form)
Sub **VGoCenter**(*Sequence* As String, *LocalNumber* As Integer, *TargetTolerance* As Double，*RobotSpeed* As Integer, *RobotAccel* As Integer)
Sub **VGoCenter**(*Sequence* As String, *LocalNumber* As Integer, *TargetTolerance* As Double，*RobotSpeed* As Integer, *RobotAccel* As Integer, *Parent* As Form)

參數

Sequence	代表目前專案中視覺序列名稱的字串運算式。
LocalNumber	代表機器人移動位置之本地座標編號的整數。 如果指定-1，機器人會在工具旋轉的 XY 平面中移動。
TargetTolerance	代表像素距離的實際值，將視覺偵測結果視為符合目標位置。 數值範圍：0 至 3 像素
Form	視窗的父.NET 表單(選用)。
RobotSpeed	選用。包含機器人速度(%)的整數變數。 數值範圍：0 至 100 如果省略，請設為「5」。
RobotAccel	選用。包含機器人加速(%)的整數變數。 數值範圍：0 至 99 如果省略，請設為「5」。

另請參閱

VDefArm, VDefGetMotionRange, VDefLocal, VDefSetMotionRange, VDefTool

VGoCenter 範例

VB 例:

```
m_spel.VGoCenter("myseq", 1, 1.0)
```

C# 例:

```
m_spel.VGoCenter("myseq", 1, 1.0);
```

VLoad 方法，Spel 類別

描述

從目前專案載入視覺屬性。

語法

Sub **VLoad** ()

備註

要在程式啟動期間將視覺屬性設定、模型及字型恢復為原始設定時，請使用 VLoad 方法。

另請參閱

VSave

VLoad 範例

VB 例:

```
m_spel.VLoad()
```

C# 例:

```
m_spel.VLoad();
```

VLoadModel 方法，Spel 類別

描述

從驅動盤檔案載入視覺模型。

語法

Sub **VLoadModel** (*Sequence* As String, *Object* As String, *Path* As String)

參數

Sequence	包含目前專案中序列之名稱的字串。
Object	包含物件名稱的字串。物件必須是 Correlation、Geometric 或 Polar。
Path	要載入模型之檔案的完整路徑名稱，不包括副檔名。

備註

若檔案的模型資料為錯誤的類型，將會發生錯誤。例如，如果嘗試將 polar 模型載入 correlation，將會發生錯誤。
如果提供副檔名，則會予以忽略。有兩個檔案與 fileName 相關聯。

對於 correlation 和 geometric 模型，ModelOrgX 和 ModelOrgY 值會與模型視窗寬度和高度一起恢復。
對於 polar 模型，則會恢復 Radius、Thickness 及 AngleOffset。

另請參閱

VSaveModel

VLoadModel 範例

VB 例:

```
m_spel.VLoadModel("seq01", "corr01", "d:\models\part1")
```

C# 例:

```
m_spel.VLoadModel("seq01", "corr01", @"d:\models\part1");
```

VRun 方法，Spel 類別

描述

在目前專案中執行視覺序列。

語法

Sub **VRun** (*Sequence* As String)

參數

Sequence　包含目前專案中序列之名稱的字串。

備註

VRun 支援任何攝影機校準或無校準類型的序列。

若要顯示圖形，您必須使用 SPELVideo 控制項，並將 Spel 類別執行個體的 SpelVideoControl 屬性設為 SPELVideo 控制項。

執行 VRun 後，請使用 VGet 擷取結果。

另請參閱

VGet, VSet

VRun 範例

VB 例:

```
Function FindPart(x As Single, y As Single, angle As Single)
As Boolean
  Dim found As Boolean
  Dim x, y, angle As Single
  With m_spel
    .VRun("seq01")
    .VGet("seq01", "corr01", "found", found)
    If found Then
      .VGet("seq01", "corr01", "cameraX", x)
      .VGet("seq01", "corr01", "cameraY", y)
      .VGet("seq01", "corr01", "angle", angle)
      FindPart = True
    End If
  End With
End Function
```

C# 例:

```
bool FindPart(float x, float y, float angle)
{
  bool found;
  m_spel.VRun("seq01");
  m_spel.VGet("seq01", "corr01", "found", found);
  if (found) {
        m_spel.VGet("seq01", "corr01", "cameraX", out x);
       m_spel.VGet("seq01", "corr01", "cameraY", out y);
        m_spel.VGet("seq01", "corr01", "angle", out angle);
    }
    return found;
}
```

VSave 方法，Spel 類別

描述

在目前專案中保存所有視覺資料。

語法

Sub **VSave** ()

備註

使用 **VSave** 可對視覺屬性作任何永久改變。

另請參閱

VSet

VSave 範例

```
With m_spel
```

VB 例:

```
    .VSet("seq01", "blob01", "SearchWinLeft", 100)
    .VSet("seq01", "corr01", "Accept", userAccept)
    .VSave()
End With
```

C# 例:

```
m_spel.VSet("seq01", "blob01", "SearchWinLeft", 100);
m_spel.VSet("seq01", "corr01", "Accept", userAccept);
m_spel.VSave();
```

VSaveImage 方法，Spel 類別

描述

將視覺視訊視窗保存至 PC 驅動盤檔案。

語法

Sub **VSaveImage** (*Sequence* As String, *Path* As String)
Sub **VSaveImage** (*Sequence* As String, *Path* As String, *WithGraphics* As Boolean)

參數

Sequence 包含目前專案中序列之名稱的字串。

Path 要保存影像之檔案的完整路徑名稱，包括副檔名。

WithGraphics 設定是否要在影像檔中保存序列結果圖形的 Boolean 運算式。

備註

使用 VSaveImage 可將視訊顯示器上的影像保存至驅動盤。副檔名必須是 MIM(Vision Guide 的預設格式)、BMP、TIF 或 JPG。

另請參閱

LoadImage(SPELVideo 控制項)

VSaveImage 範例

VB 例:

```
Dim found As Boolean
m_spel.VRun("Seq")
m_spel.VGet("Seq", SpelVisionProps.AllFound, found)
If Not found Then
    m_spel.VSaveImage("Seq", "d:\reject.mim")
End If
```

C# 例:

```
bool found;
m_spel.VRun("Seq");
m_spel.VGet("Seq", SpelVisionProps.AllFound, out found);

if (!found)
    m_spel.VSaveImage("Seq", @"d:\reject.mim");
```

VSaveModel 方法，Spel 類別

描述

將視覺物件搜尋模型保存至 PC 驅動盤檔案。

語法

Sub **VSaveModel** (*Sequence* As String, *Object* As String, *Path* As String)

參數

Sequence	包含目前專案中序列之名稱的字串。
Object	包含物件名稱的字串。物件必須是 Correlation、Geometric 或 Polar。
Path	要保存模型之檔案的完整路徑名稱，不包括副檔名。

備註

執行 **VSaveModel** 時，EPSON RC+ 7.0 會建立兩個檔案(*Path* + 副檔名)：
Path.**VOB**，*Path*.**MDL**
對於 correlation 和 geometric 模型，ModelOrgX 和 ModelOrgY 值會與模型視窗一起保存。
對於 Polar 模型，則會保存 Radius、Thickness 及 AngleOffset。

另請參閱

VLoadModel

VSaveModel 範例

VB 例:

```
m_spel.VSaveModel("seq01", "corr01", "d:\models\part1")
```

C# 例:

```
m_spel.VSaveModel("seq01", "corr01", @"d:\models\part1");
```

VSet 方法，Spel 類別

描述

設定視覺序列或物件屬性的值。

語法

Sub **VSet** (*Sequence* As String, *PropCode* As SpelVisionProps, *Value* As Integer)
Sub **VSet** (*Sequence* As String, *PropCode* As SpelVisionProps, *Value* As Boolean)
Sub **VSet** (*Sequence* As String, *PropCode* As SpelVisionProps, *Value* As Double)
Sub **VSet** (*Sequence* As String, *PropCode* As SpelVisionProps, *Value* As Single)
Sub **VSet** (*Sequence* As String, *PropCode* As SpelVisionProps, *Value* As String)
Sub **VSet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Value* As Integer)
Sub **VSet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Value* As Boolean)
Sub **VSet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Value* As Color)
Sub **VSet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Value* As Double)
Sub **VSet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Value* As Single)
Sub **VSet** (*Sequence* As String, *Object* As String, *PropCode* As SpelVisionProps, *Value* As String)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。若屬性用於序列，則此字串必須為空白。
PropCode	指定屬性代碼的 SpelVisionProps 值。
Value	包含新建值的運算式。運算式類型必須符合屬性類型。

另請參閱

VGet, VRun

VSet 範例

VB 例:

```
m_spel.VSet("seq01", "corr01", SpelVisionProps.Accept, 250)
```

C# 例:

```
m_spel.VSet("seq01", "corr01", SpelVisionProps.Accept, 250);
```

VSetModelWin 方法，Spel 類別

描述

設定模型視窗座標。

語法

Sub **VSetModelWin** (*Sequence* As String, *Object* As String, *Left* As Integer，*Top* As Integer,
Width As Integer, *Height* As Integer)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Left	代表左側座標的整數運算式(以像素為單位)。
Top	代表頂端座標的整數運算式(以像素為單位)。
Width	代表寬度的整數運算式(以像素為單位)。
Height	代表高度的整數運算式(以像素為單位)。

另請參閱

VGetModelWin, VGetSearchWin, VSetSearchWin

VSetModelWin 範例

VB 例:

```
Dim left As Integer, top As Integer
Dim width As Integer, height As Integer

With m_spel
  .VGetSearchWin("testSeq", "corr01", left, top, _
    width, height)
  .VSetSearchWin("testSeq", "corr01", left + 50, _
    top - 10, width, height)
  .VRun("testSeq")
End With
```

C# 例:

```
int left, top, width, height;

m_spel.VGetSearchWin("testSeq", "corr01", out left, out top,
    out width, out height);
m_spel. .VSetSearchWin("testSeq", "corr01", left + 50,
    top - 10, width, height);
m_spel.VRun("testSeq");
```

VSetRobotPlaceTargetPos 方法，Spel 類別

描述

設定工件放置位置。

語法

Sub **VSetRobotPlaceTargetPos** (*Sequence* As String, *Object* As String, *Point* As SpelPoint)

參數

Sequence 包含目前專案中視覺序列之名稱的字串運算式。

Object 包含 *Sequence* 序列中物件之名稱的字串運算式。

Point 包含放置位置的 SpelPoint 變數

另請參閱

VB 例:

VGetRobotPlacePos, VGetRobotPlaceTargetPos

VSetRobotPlaceTargetPos Example

```
Dim seq As String, blob As String
Dim targetPoint As SpelPoint

seq = "testSeq"
blob = "blob01"
m_spel.VGetRobotPlaceTargetPos(seq, blob, targetPoint)

' 調整放置位置
targetPoint.X = targetPoint.X + 10
m_spel.VSetRobotPlaceTargetPos(seq, blob, targetPoint)
```

C# 例:

```
string seq, blob;
SpelPoint targetPoint = new SpelPoint();

seq = "testSeq";
blob = "blob01";
m_spel.VGetRobotPlaceTargetPos(seq, blob, out targetPoint);

// 調整放置位置
targetPoint.X = targetPoint.X + 10;
m_spel.VSetRobotPlaceTargetPos(seq, blob, targetPoint);
```

VSetSearchWin 方法，Spel 類別

描述

設定搜尋視窗座標。

語法

Sub **VSetSearchWin** (*Sequence* As String, *Object* As String, *Left* As Integer，*Top* As Integer, *Width* As Integer, *Height* As Integer)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含 *Sequence* 序列中物件之名稱的字串運算式。
Left	代表左側座標的整數運算式(以像素為單位)。
Top	代表頂端座標的整數運算式(以像素為單位)。
Width	代表寬度的整數運算式(以像素為單位)。
Height	代表高度的整數運算式(以像素為單位)。

另請參閱

VGetModelWin, VSetModel, VGetSearchWin

VSetSearchWin 範例

VB 例:

```
Dim left As Integer, top As Integer
Dim width As Integer, height As Integer

With m_spel
  .VGetSearchWin("testSeq", "corr01", left, top, _
    width, height)
  .VSetSearchWin("testSeq", "corr01", newLeft, top, _
    width, height)
  .VRun("testSeq")
End With
```

C# 例:

```
int left, top, width, height;

m_spel.VGetSearchWin("testSeq", "corr01", out left, out top,
    out width, out height);
m_spel. .VSetSearchWin("testSeq", "corr01", left + 50,
    top, width, height);
m_spel.VRun("testSeq");
```

VShowModel 方法，Spel 類別

描述

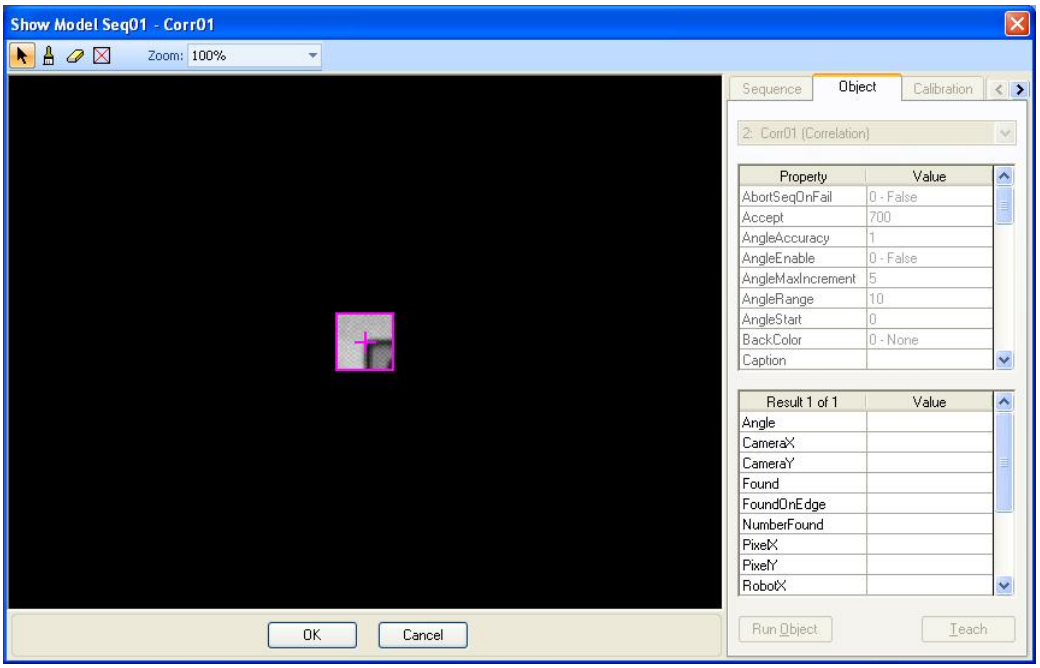
顯示物件模型。如需詳細資訊，請參閱 Vision Guide 屬性參考中的 ShowModel 屬性。

語法

Sub **VShowModel** (*Sequence* As String, *Object* As String)

參數

Sequence	包含目前專案中視覺序列之名稱的字串運算式。
Object	包含目前專案中視覺物件之名稱的字串運算式。



另請參閱

VShowSequence, VTrain

VShowModel 範例

VB 例:

```
m_spel.VShowModel("myseq", "myobj")
```

C# 例:

```
m_spel.VShowModel("myseq", "myobj");
```

VShowSequence 方法，Spel 類別

描述

顯示序列中的所有物件。

語法

Sub **VShowSequence** (*Sequence* As String)

參數

Sequence	包含要建立之視覺序列名稱的字串運算式。

備註

使用 VShowSequence 可在不執行序列的情況下顯示序列中的物件。使用中物件的顏色(洋紅色)會用於所有物件，以便輕易查看。其中一個用途是，當機器人攝影機移至某工件的特定部分上時，會利用多個序列進行掃描。機器人處於定位後，可調用 VShowSequence 以顯示序列。

另請參閱

VShowModel

VShowSequence 範例

VB 例:

```
m_spel.VShowSequence("myseq")
```

C# 例:

```
m_spel.VShowSequence("myseq");
```

VStatsReset 方法，Spel 類別

描述

重置目前專案中指定序列的視覺統計。

語法

Sub **VStatsReset** (*Sequence* As String)

參數

Sequence　　包含目前專案中視覺序列之名稱的字串運算式。

備註

VStatsReset 只會針對目前 EPSON RC+ 7.0 會話，重置記憶體中指定序列的統計。若要永久改變，您應執行 VStatsSave。否則，若您重啟 EPSON RC+ 7.0，統計就會從驅動盤恢復。

另請參閱

VStatsResetAll, VStatsShow, VStatsSave

VStatsReset 範例

VB 例:

```
Sub btnResetStats_Click()
    m_spel.VStatsReset("seq01")
End Sub
```

C# 例:

```
void btnResetStats_Click(object sender, EventArgs e)
{
    m_spel.VStatsReset("seq01");
}
```

VStatsResetAll 方法，Spel 類別

描述

重置所有序列的視覺統計。

語法

Sub **VStatsResetAll**

備註

VStatsResetAll 只會針對目前 EPSON RC+ 7.0 會話，重置記憶體中的統計。若要永久改變，您應執行 VStatsSave。否則，若您重啟 EPSON RC+ 7.0，統計就會從驅動盤恢復。

另請參閱

VStatsReset, VStatsShow, VStatsSave

VStatsResetAll 範例

VB 例:

```
Sub btnResetStats_Click()
    m_spel.VStatsResetAll()
End Sub
```

C# 例:

```
void btnResetStats_Click(object sender, EventArgs e)
{
    m_spel.VStatsResetAll();
}
```

VStatsSave 方法，Spel 類別

描述

保存目前專案中所有序列的視覺統計。

語法

Sub **VStatsSave ()**

備註

若要保存對視覺統計所作的改變，必須在關閉 EPSON RC+ 7.0 之前執行 **VStatsSave**。

另請參閱

VStatsReset, VStatsResetAll, VStatsShow

VStatsSave 範例

VB 例:

```
Sub btnResetStats_Click()
    m_spel.VStatsSave()
End Sub
```

C# 例:

```
void btnResetStats_Click(object sender, EventArgs e)
{
    m_spel.VStatsSave();
}
```

VStatsShow 方法，Spel 類別

描述

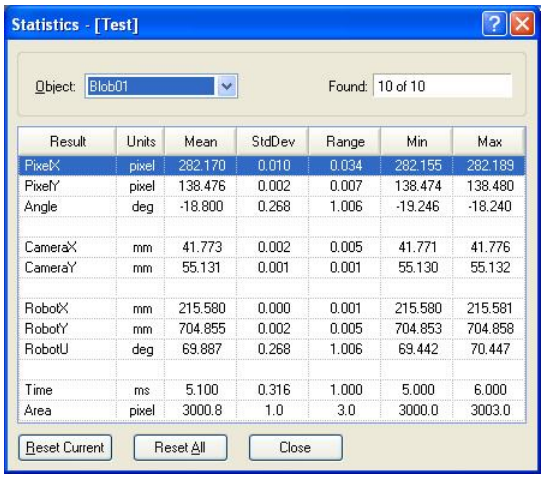
顯示目前專案中指定序列的視覺統計對話方塊。

語法

Sub **VStatsShow** (*Sequence* As String)

參數

Sequence　　包含目前專案中視覺序列之名稱的字串運算式。



另請參閱

VStatsReset, VStatsResetAll, VStatsSave

VStatsShow 範例

VB 例:

```
Sub btnShowStats_Click()
    m_spel.VStatsShow("seq01")
End Sub
```

C# 例:

```
void btnShowStats_Click(object sender, EventArgs e)
{
    m_spel.VStatsShow("seq01");
}
```

VTeach 方法，Spel 類別

描述

示教 correlation、geometric 或 polar 模型。

語法

Sub **VTeach** (*Sequence* As String, *Object* As String, ByRef *Status* as Integer)
Sub **VTeach** (*Sequence* As String, *Object* As String, *AddSample* as Boolean, *KeepDontCares* As Boolean, ByRef *Status* as Integer)

參數

Sequence	包含目前專案中之視覺序列的名稱。
Object	*Sequence* 中之物件的名稱。您可示教 Correlation、Geometric 或 Polar 物件。
AddSample	新增範例時為 True，新增新模型時為 False。
KeepDontCares	使用舊偵測遮罩時為 True，處置時為 False。
Status	傳回狀態。成功會傳回 1，否則會傳回 0。

備註

在調用 **VTeach** 之前，您必須確定模型視窗位於正確位置。
對於 polar 物件，搜尋視窗和 thickness 必須正確設定。使用 VSet 搜尋視窗位置及 thickness。
對於 correlation 和 geometric 物件，搜尋視窗及模型視窗必須正確設定。使用 SearchWin 和 ModelWin 的 VSet 來設定搜尋及模型視窗位置。或者，您也可以使用 VTrain 命令，讓操作員以互動方式改變視窗。
示教模型之後，您可以使用 VSaveModel 方法將其保存至 PC 驅動盤檔案。

另請參閱

VTrain, VSaveModel

VTeach 範例

VB 例:

```
Dim status As Integer

' 先讓操作員改變視窗位置
m_spel.VTrain("seq01", "corr01", status)

' 現在示教模型
m_spel.VTeach("seq01", "corr01", status)
```

C# 例:

```
int status;

// 先讓操作員改變視窗位置
m_spel.VTrain("seq01", "corr01", status);

// 現在示教模型
m_spel.VTeach("seq01", "corr01", out status);
```

VTrain 方法，Spel 類別

描述

此命令可讓您示教整個序列中的物件，或示教個別物件。

語法

Function VTrain (Sequence As String [, Object As String] [, Flags as Integer] [, *Parent* as Form]) As Boolean

參數

Sequence	包含目前專案中之視覺序列的名稱。
Object	*Sequence* 中之物件的名稱。您可示教任何物件類型。若 *Object* 是空白字串，則可示教整個序列。
Flags	選用。設定 VTrain 對話方塊 1 -顯示示教按鈕 2 -不顯示模型視窗。
Parent	選用。係一將成為視窗的父項之.NET 表單。

傳回值

若操作員按一下 OK 按鈕，VTrain 會傳回 True，否則會傳回 False。

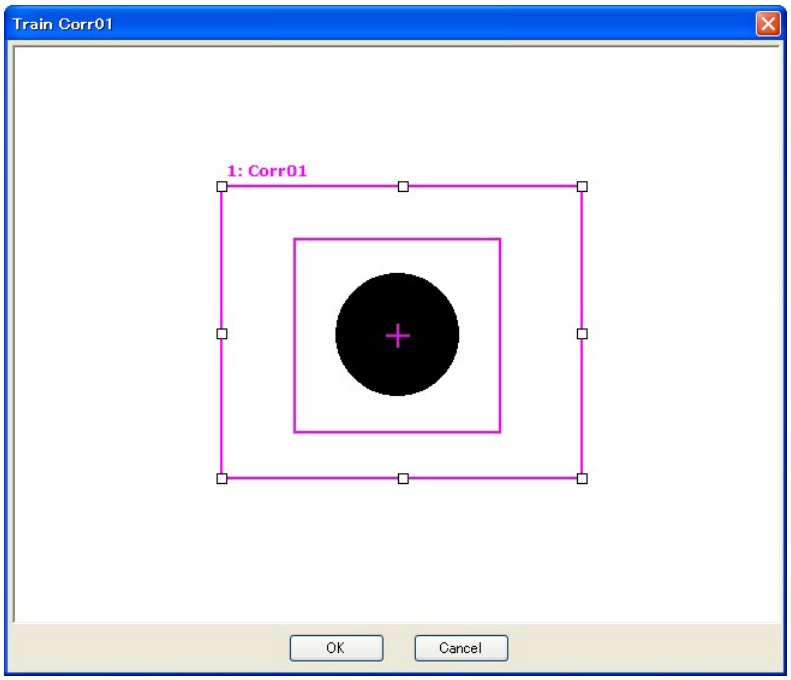
備註

執行 **VTrain** 時，會開啟顯示即時視訊的對話方塊，並顯示指定的物件。操作員可調整搜尋視窗大小或移動該視窗，並示教模型視窗(correlation 和 geometric 物件適用)。操作員完成操作時，可按一下 OK 以保存改變，或按一下取消以忽略改變。若按一下 OK，新建資訊會自動保存至目前專案。

若 *flags* 位元 1 已設定，將會顯示示教按鈕。對於 Correlation、Geometric 及 Polar 物件，若按一下示教按鈕，將會示教模型。您可在執行 VTrain 之後擷取 ModelOK 屬性，以確認模型是否完成示教。對於 Blob 物件，該按鈕將會開啟條形圖，且操作員可調整高低閾值，然後查看改變的結果。

若 *flags* 位元 2 已設定，將不會顯示模型視窗。操作員僅能改變搜尋視窗。

對於 correlation 和 geometric 物件，如果不顯示示教按鈕，您可在調用 **VTrain** 之後調用 VTeach 以示教模型。



另請參閱

VTeach, VSaveModel

VTrain 範例

VB 例:

```
Dim status As Integer

' 先讓操作員改變視窗位置
trainOK =m_spel.VTrain("seq01", "corr01")

' 現在示教模型
If trainOK Then
m_spel.VTeach("seq01", "corr01", status)
EndIf
```

C# 例:

```
int status;
bool trainOK;

// 先讓操作員改變視窗位置
trainOK = m_spel.VTrain("seq01", "corr01");

// 現在示教模型
if (trainOK)
    m_spel.VTeach("seq01", "corr01", out status);
```

WaitCommandComplete 方法，Spel 類別

描述

此命令會等待以 AsyncMode = True 開頭的命令完成。

語法

Sub **WaitCommandComplete ()**

另請參閱

AsyncMode

WaitCommandComplete 範例

VB 例:

```
With m_spel
  .AsyncMode = True
  .Jump("pick")
  .Delay(500)
  .On(1)
  .WaitCommandComplete()
End With
```

C# 例:

```
m_spel.AsyncMode = true;
m_spel.Jump("pick");
m_spel.Delay(500);
m_spel.On(1);
m_spel.WaitCommandComplete();
```

WaitMem 方法，Spel 類別

描述

等待記憶體位元狀態改變。

語法

Sub **WaitMem** (*BitNumber* As Integer, *Condition* As Boolean, *Timeout* As Single)
Sub **WaitMem** (*Label* As String, *Condition* As Boolean, *Timeout* As Single)

參數

BitNumber	代表記憶體位元編號的整數運算式。
Label	代表記憶體位元標籤的字串。
Condition	代表記憶體位元狀態的 Boolean 運算式。
Timeout	代表最長等待時間的單一運算式(以秒為單位)。

備註

請務必使用 TW 方法檢查是否發生超時。請參閱以下範例。

另請參閱

WaitSw

WaitMem 範例

VB 例:

```
' 等待記憶體位元 1 變成 1 (True)
' 最長時間為 5 秒
m_spel.WaitMem(1, True, 5)
' WaitMem 是否超時？
If m_spel.TW() Then
    MsgBox "memory bit time out occurred"
End If
```

C# 例:

```
// 等待記憶體位元 1 變成 1 (True)
// 最長時間為 5 秒
m_spel.WaitMem(1, True, 5);
// WaitMem 是否超時？
if (m_spel.TW())
    MessageBox.Show("memory bit time out occurred");
```

WaitSw 方法，Spel 類別

描述

等待輸入位元狀態改變。

語法

Sub **WaitSw** (*BitNumber* As Integer, *Condition* As Boolean, *Timeout* As Single)
Sub **WaitSw** (*Label* As String, *Condition* As Boolean, *Timeoutl* As Single)

參數

BitNumber	代表輸入位元編號的整數運算式。
Label	代表輸入位元標籤的字串。
Condition	代表輸入位元狀態的 Boolean 運算式。
Timeout	代表最長等待時間的單一運算式(以秒為單位)。

備註

請務必使用 TW 方法檢查是否發生超時。請參閱以下範例。

另請參閱

WaitMem

WaitSw 範例

VB 例:

```
Const PartPresent = 1
m_spel.WaitSw(PartPresent, True, 5)
If m_spel.TW() Then
    MsgBox "Part present time out occurred"
End If
```

C# 例:

```
const int PartPresent = 1;
m_spel.WaitSw(PartPresent, True, 5);
if (m_spel.TW())
    MessageBox.Show("Part Present time out occurred");
```

WaitTaskDone 方法，Spel 類別

描述

等待任務完成並傳回狀態。

語法

Function **WaitTaskDone** (*TaskNumber* As Integer) As SpelTaskState
Function **WaitTaskDone** (*TaskName* As String) As SpelTaskState

參數

TaskNumber　傳回執行狀態的任務編號。

TaskName　包含任務名稱的字串運算式。

傳回值

SpelTaskState 值。

另請參閱

SpelTaskState, TasksExecuting, TaskState, Xqt

WaitTaskDone 範例

VB 例:

```
Dim taskState As SpelTaskState
m_spel.Xqt 2, "mytask"
'
' 其他處理
'
taskState = m_spel.WaitTaskDone(2)
```

C# 例:

```
SpelTaskState taskState;
m_spel.Xqt(2, "mytask");
//
// 其他處理
//
taskState = m_spel.WaitTaskDone(2);
```

Weight 方法，Spel 類別

描述

指定目前機器人的重量參數。

語法

Sub **Weight** (*PayloadWeight* As Single, *ArmLength* As Single)
Sub **Weight** (*PayloadWeight* As Single, *Axis* As SpelAxis, [Axis])

參數

PayloadWeight	待運送之夾具末端的重量(公斤)。
ArmLength	從第二手臂旋轉中心至夾具末端重心的距離(公釐)。
Axis	指定指派負載重量的附加軸(S 或 T)。

NOTE

請勿將整數值輸入至 PayLoadWeight 和 ArmLength 參數。使用單一變數或直接輸入單一類型值。

另請參閱

Inertia, JRange, Tool

Weight 範例

VB 例:

```
m_spel.Weight(2.0F, 2.5F)
```

C# 例:

```
m_spel.Weight(2.0F, 2.5F);
```

Xqt 方法，Spel 類別

描述

啟動一個 SPEL+任務。

語法

Sub **Xqt** (*FuncName* As String [, *TaskType* As SpelTaskType])
Sub **Xqt** (*TaskNumber* As Integer, *FuncName* As String [, *TaskType* As SpelTaskType])

參數

TaskNumber	要執行之任務的任務編號。任務編號的範圍介於 1 至 32。
FuncName	要執行之函數的名稱。您也可以為函數提供引數。引數必須加上括號，並以逗號分隔。如需詳細資訊，請參閱 SPEL+ Xqt 陳述式。另請參閱範例。
TaskType	選用。將任務類型指定為 Normal、NoPause 或 NoEmgAbort。

備註

執行 **Xqt** 時，控制將會立即回到調用程式。請使用 Call 方法等待任務完成，或者，亦可使用含任務狀態事件的 EventReceived 等待任務結束。

另請參閱

Call, EnableEvent, EventReceived

Xqt 範例

VB 例:

```
m_spel.Xqt(2, "conveyor")

' 提供引數給 RunPart 函數
m_spel.Xqt(3, "RunPart(3)")

Dim funcToExec As String
funcToExec = "RunPart(" & partNum & ")"
m_spel.Xqt(3, funcCall)
```

C# 例:

```
m_spel.Xqt(2, "conveyor");

// 提供引數給 RunPart 函數
m_spel.Xqt(3, "RunPart(3)");

string funcToExec;
funcToExec = string.Format("RunPart({0})", partNum);
m_spel.Xqt(3, funcToExec);
```

XYLim 方法，Spel 類別

描述

設定機器人的允許動作範圍限制。

語法

Sub **XYLim** (*XLowerLimit* As Single, *XUpperLimit* As Single, *YLowerLimit* As Single, *YUpperLimit* As Single [, *ZLowerLimit* As Single] [, *ZUpperLimit* As Single])

參數

XLowerLimit	機器人可移動的最小 X 座標位置。(機器人無法移至小於 minX 的 X 座標位置。)
XUpperLimit	機器人可移動的最大 X 座標位置。(機器人無法移至大於 maxX 的 X 座標位置。)
YLowerLimit	機器人可移動的最小 Y 座標位置。(機器人無法移至小於 minY 的 Y 座標位置。)
YUpperLimit	機器人可移動的最大 Y 座標位置。(機器人無法移至大於 maxY 的 Y 座標位置。)
ZLowerLimit	選用。機器人可移動的最小 Z 座標位置。(機器人無法移至小於 minZ 的 Z 座標位置。)
ZUpperLimit	選用。機器人可移動的最大 Z 座標位置。(機器人無法移至大於 maxZ 的 Z 座標位置。)

備註

XYLim 係用來定義動作範圍限制。許多機器人系統都可讓使用者定義關節限制，而 SPEL+語言則可允許同時定義關節限制與動作範圍限制。此可讓使用者有效地為其應用建立工作空間。(請記住，關節範圍限制也可以使用 SPEL 定義。)

使用 XYLim 值建立的動作範圍只會套用至動作命令目標位置，不會套用到起始位置至目標位置的動作路徑。因此，手臂可能會在動作期間超出 XYLim 範圍以外。(亦即 XYLim 範圍不會影響脈衝。)

若要關閉動作範圍限制，請將範圍限制參數指定為 0。

另請參閱

JRange

XYLim 範例

VB 例:

```
m_spel.XYLim(0, 0, 0, 0)
```

C# 例:

```
m_spel.XYLim(0, 0, 0, 0);
```

XYLimClr 方法，Spel 類別

描述

清除(取消定義)XYLim 定義。

語法

Sub **XYLimClr** ()

另請參閱

XYLim, XYLimDef

XYLimClr 範例

VB 例:

```
m_spel.XLLimClr()
```

C# 例:

```
m_spel.XYLimClr();
```

XYLimDef 方法，Spel 類別

描述

傳回 XYLim 是否定義。

語法

Function **XYLimDef** () As Boolean

傳回值

若 XYLim 已定義，會傳回 True，否則會傳回 False。

另請參閱

XYLim, XYLimClr

XYLimDef 範例

VB 例:

```
Dim xyLimDefined As Boolean
xyLimDefined = m_spel.XYLimDef()
```

C# 例:

```
bool xyLimDefined;
xyLimDefined = m_spel.XYLimDef();
```

14.4 Spel類別事件

EventReceived 事件，Spel 類別

描述

於 EPSON RC+ 7.0 傳送系統事件或在 SPEL+中執行的程式利用 SPELCom_Event 陳述式傳送事件時發生。

語法

EventReceived (ByVal *sender* As Object, ByVal *e* As RCAPINet.SpelEventArgs)

參數

e.Event 代表特用使用者定義事件的編號。

e.Message 包含事件訊息的字串。

備註

EPSON RC+ 7.0 會發出多種系統事件。下表提供詳細說明。

系統事件

依據預設，部分事件為停用狀態。若要使用這些事件，您必須先使用 EnableEvent 方法將他們啟用。

事件編號	事件訊息	常數	描述
1	"PAUSE"	SpelEvents.Pause	於任務暫停時發生。預設為啟用。
2	"SAFE GUARD OPEN"	SpelEvents.SafeGuardOpen	於安全防護打開時發生。預設為啟用。
3	"SAFE GUARD CLOSE"	SpelEvents.SafeGuardClose	於安全防護關閉時發生。預設為啟用。
4	Project build status text	SpelEvents.ProjectBuildStatus	各建置狀態訊息會在 BuildProject 方法期間傳送。CRLF 會視需要添加。這些訊息與在 EPSON RC+ 7.0 GUI 的 Project Build Status 視窗上顯示的訊息相同。此事件必須以 EnableEvent 方法啟用。預設為停用。
5	"Error xxx!: mmm in task at line yyy"	SpelEvents.Error	於因未處理的錯誤或產生系統錯誤而導致任務終止時發生。預設為啟用。
6	Text from print statement	SpelEvents.Print	於一 Print 陳述式從 SPEL+ 任務執行時發生。預設為停用。
7	"ESTOP ON"	SpelEvents.EStopOn	於緊急停止情況變為 ON 時發生。預設為啟用。
8	"ESTOP OFF"	SpelEvents.EStopOff	於緊急停止情況變為 OFF 時發生。預設為啟用。
9	"CONTINUE"	SpelEvents.Continue	於執行 Cont 後發生。預設為啟用。

事件編號	事件訊息	常數	描述
10	<Robot #>,"MOTOR ON"	SpelEvents.MotorOn	於所指示機器人的馬達變為 ON 時發生。 預設為停用。
11	<Robot #>,"MOTOR OFF"	SpelEvents.MotorOff	於所指示機器人的馬達變為 OFF 時發生。 預設為停用。
12	<Robot #>,"POWER HIGH"	SpelEvents.PowerHigh	於所指示機器人的運行功率變為 HIGH 時發生。 預設為停用。
13	<Robot #>,"POWER LOW"	SpelEvents.PowerLow	於所指示機器人的運行功率變為 LOW 時發生。 預設為停用。
14	"TEACH MODE"	SpelEvents.TeachMode	於示教模式啟動時發生。 預設為啟用。
15	"AUTO MODE"	SpelEvents.AutoMode	於自動模式啟動時發生。 預設為啟用。
16	"<TaskID>,<Status>, <FuncName>" Status: "RUN", "HALT", "PAUSE", “FINISHED”, “ABORTED”	SpelEvents.TaskState	於任務狀態改變時發生。 預設為無效。
17	"SHUTDOWN"	SpelEvents.Shutdown	於 RC+正在關閉時發生。 預設為無效。
18	"ALL TASKS STOPPED"	SpelEvents.AllTasksStopped	於所有任務已停止時發生。 預設為無效。
19	"DISCONNECTED"	SpelEvents.Disconnected	當控制器通信與 PC 中斷時發生。 啟用時，RC+不會顯示表示連線中斷的訊息方塊。 預設為無效。
20	“MOTION STARTED”	SpelEvents.MotionStarted	控制命令已經開始。 預設為無效。
21	“MOTION COMPLETE”	SpelEvents.MotionComplete	控制命令已經結束。 預設為無效。

使用者事件

您可使用 **SPELCom_Event** 命令，將事件從 SPEL+程式傳送至 Visual Basic 應用程式。

```
Spelcom_Event 3000, cycNum, lotNum, cycTime
```

此陳述式執行時，EventReceived 常式將會連同事件編號及訊息一起調用。請參閱 *EPSON RC+ 7.0 線上說明* 或 *13. SPELCom_Event*，瞭解 SPELCom_Event 的詳細資訊。

使用範例

VB 例:

```
Sub m_spel_EventReceived ( _
         ByVal sender As Object, _
         ByVal e As RCAPINet.SpelEventArgs) _
         Handles m_spel.EventReceived
    Redim tokens(0) As String
    Select Case e.Event
        Case 3000
          tokens = e.Message.Split(New [Char]() {" "c}, _


System.StringSplitOptions.RemoveEmptyEntries)
            lblCycCount.Text = tokens(0)
            lblLotNumber.Text = tokens(1)
            lblCycTime.Text = tokens(2)
    End Select
End Sub
```

C# 例:

```
public void m_spel_EventReceived(object sender,
SpelEventArgs e)
{
    string[] tokens = new string[3];
    switch ((int) e.Event){
        case 3000:
            tokens = e.Message.Split(' ');
            lblCycCount.Text = tokens(0);
            lblLotNumber.Text = tokens(1);
            lblCycTime.Text = tokens(2);
            break;
        default:
            break;
    }
}
```

處理事件

當 **EventReceived** 從 Spel 類別執行個體調用時，EPSON RC+ 7.0 處理伺服器將需等待事件處理常式結束。因此，您切勿嘗試在 **EventReceived** 常式期間執行任何 RC+ API 命令或執行長執行時間的處理作業。若要根據發生的事件執行命令，請在 **EventReceived** 中設定旗標，並從事件處理函數範圍外的應用程式 main 迴圈處理旗標。

例如，在 Load 程序的 Visual Basic 主要表單中，您可從 SPEL+建立接收事件的事件迴圈。在 spel_EventReceived 常式中，設定全域旗標以指示已接收的事件。接著，您可從 Load 程序所建立的事件迴圈中執行實際的事件處理。

顯示事件訊息

將 TextBox 控制項添加至表單。
每次收到事件時，您可以在 TextBox 控制項的 Text 屬性中顯示事件訊息。

VB 例:

```
Private Sub m_spel_EventReceived(ByVal sender As Object, _
        ByVal e As SpelEventArgs) Handles
m_spel.EventReceived
    txtEvents.AppendText(e.Event & ": " & e.Message &
vbCrLf)
End Sub
```

C# 例:

```
private void m_spel_EventReceived(object sender,
SpelEventArgs e)
{
   txtEvents.AppendText(e.Event + ": " + e.Message);
}
```

另請參閱

EnableEvent(Spel 類別)

14.5 SPELVideo控制項

描述

此控制項可讓您顯示視覺系統的視訊。如需此控制項使用方式的資訊，請參閱第 11 章的*顯示視訊*。

檔案名稱

RCAPINet.dll

14.6 SPELVideo控制項屬性

除了標準.NET 元件屬性外(例如 Left、Top、Width 及 Height)，此控制項亦支援下列屬性。如需標準屬性的說明文件，請參閱 Visual Basic 線上說明。

- Camera
- GraphicsEnabled
- VideoEnabled

Camera 屬性，SPELVideo 控制項

描述

設定／取得要顯示視訊的攝影機編號。當您想在步進操作、即時視訊監控等情況下顯示視訊時，這會相當實用。若您目前使用控制項來顯示視覺序列的圖形，則序列執行時，將會使用該序列的攝影機編號來取代此屬性值。

語法

Property **Camera** As Integer

預設值

0 – 顯示任何攝影機

傳回值

包含目前攝影機編號的整數值

另請參閱

VideoEnabled, GraphicsEnabled

範例

VB 例:

```
SpelVideo1.Camera = 1
```

C# 例:

```
SpelVideo1.Camera = 1;
```

GraphicsEnabled 屬性，SPELVideo 控制項

描述

設定／傳回是否要在執行序列後顯示視覺圖形。為能看見圖形，您必須使用 SPELVideo 控制項屬性將控制項附加至 Spel 類別執行個體。此屬性可「即時」設定，能在執行序列時開啟／關閉圖形。

語法

Property **GraphicsEnabled** As Boolean

預設值

False

傳回值

若顯示視覺圖形，會傳回 True，否則會傳回 False。

另請參閱

Camera, VideoEnabled

範例

VB 例:

```
SpelVideo1.GraphicsEnabled = True
```

C# 例:

```
SpelVideo1.GraphicsEnabled = true;
```

VideoEnabled 屬性，SPELVideo 控制項

描述

決定是否要顯示視訊。

語法

Property **VideoEnabled** As Boolean

預設值

False

傳回值

若顯示視訊，會傳回 True，否則會傳回 False。

另請參閱

Camera, GraphicsEnabled

範例

VB 例:

```
SpelVideo1.VideoEnabled = True
```

C# 例:

```
SpelVideo1.VideoEnabled = true;
```

14.7 SPELVideo控制項方法

LoadImage 方法，SPELVideo 控制項

描述

從檔案載入要顯示的影像。

語法

Sub **LoadImage** (*Path* As String)

參數

Path 要載入影像之檔案的完整路徑名稱，包括副檔名。

備註

使用 LoadImage 可載入並顯示先前保存的影像。副檔名必須為 BMP、TIF 或 JPG。

另請參閱

VSaveImage(Spel 類別)

LoadImage 範例

VB 例:

```
m_spelVideo.LoadImage("c:\RejectImages\reject001.bmp")
```

C# 例:

```
m_spelVideo.LoadImage(@"c:\RejectImages\reject001.bmp");
```

14.8 SPELVideo控制項事件

此控制項的所有事件都是標準.NET 事件。 如需詳細資訊，請參閱 Visual Basic 線上說明。

14.9 SpelConnectionInfo類別

成員名稱	類型	描述
ConnectionIPAddress	String	在 EPSON RC+中設定的 IP 位址。 * 若以 USB 或虛擬控制器連接，ConnectionIPAddress 為空白。
ConnectionNumber	Integer	在 EPSON RC+中設定的連線數量。
ConnectionName	String	在 EPSON RC+中設定的連線名稱。
ConnectionType	SpelConnectionType	在 EPSON RC+中設定的連線類型。

以下是範例。

VB 例:

```
Dim connectionInfo() As RCAPINet.SpelConnectionInfo
connectionInfo = m_spel.GetConnectionInfo()
```

C# 例:

```
SpelConnectionInfo[] info = m_spel.GetConnectionInfo();
```

14.10 SpelControllerInfo類別

成員名稱	類型	描述
ProjectName	String	控制器中專案的名稱。
ProjectID	String	控制器中專案的唯一專案 ID。
Options	List<SpelOptionInfo>	控制器中選配件的清單。

以下是範例。

VB 例:

```
Dim info As RCAPINet.SpelControllerInfo
info = m_spel.GetControllerInfo()
Label1.Text = info.ProjectID + " " + info.ProjectName
```

C# 例:

```
SpelControllerInfo info;
info = m_spel.GetControllerInfo();
Label1.text = info.ProjectID + " " + info.ProjectName;
```

14.11 SpelException類別

SpelException 類別係衍生自 ApplicationException 類別。它會添加一 ErrorNumber 屬性及一些建構函式。

以下範例顯示如何擷取錯誤編號及錯誤訊息。

VB 例:

```
Try
  m_spel.Go(1)
Catch (ex As RCAPINet.SpelException)
  MsgBox(ex.ErrorNumber & " " & ex.Message)
End Try
```

C# 例:

```
try{
  m_spel.Go(1);
}
catch(RCAPINet.SpelException ex){
  MessageBox.Show(string.Format("{0}: {1}", ex.ErrorNumber,
ex.Message));
}
```

SpelException 屬性

ErrorNumber As Integer

SpelException 方法

Sub New ()

預設的建構函式。

Sub New (Message As String)

指定錯誤訊息的選用建構函式。

Sub New (ErrorNumber As Integer, Message As String)

指定錯誤編號及相關訊息的選用建構函式。

Sub New (Message As String, Inner As Exception)

指定錯誤訊息及內部例外的選用建構函式。

Sub New (ErrorNumber As Integer, Message As String, Inner As Exception)

指定錯誤編號、錯誤訊息及內部例外的選用建構函式。

14.12 SpelOptionInfo類別

成員名稱	類型	描述
Name	String	機器人名稱
Status	SpelOptionStatus	機器人編號

以下是範例。

VB 例:

```
Dim info As SpelControllerInfo
info = m_spel.GetControllerInfo()
Label1.Text = info.Options(1).Name + " " +
info.Options(1).Status
```

C# 例:

```
SpelControllerInfo info;
info = m_spel.GetControllerInfo();
Label1.Text = info.Options[1].Name + " " +
info.Options[1].Status;
```

14.13 SpelPoint類別

SpelPoint 類別可用於多種動作方法，以及 Spel 類別的 GetPoint 與 SetPoint 方法。

Visual Basic 範例如下：

1:

```
Dim pt As New RCAPINet.SpelPoint(25.5, 100.3, -21, 0)
m_spel.Go(pt)
```

2:

```
Dim pt As New RCAPINet.SpelPoint
pt.X = 25.5
pt.Y = 100.3
pt.Z = -21
m_spel.Go(pt)
```

3:

```
Dim pt As New RCAPINet.SpelPoint
pt = m_spel.GetPoint("P*")
pt.Y = 222
m_spel.Go(pt)
```

Visual C#範例如下：

1:
```
SpelPoint pt = new SpelPoint(25.5, 100.3,  -21, 0);
m_spel.Go(pt);
```
2:
```
SpelPoint pt = new SpelPoint();
pt.X = 25.5;
pt.Y = 100.3;
pt.Z = -21;
m_spel.Go(pt);
```
3:
```
SpelPoint pt = new SpelPoint();
pt = m_spel.GetPoint("P0");
pt.Y = 222;
m_spel.Go(pt);
```

14.13.1 SpelPoint屬性

VB 例:	**C# 例:**
X As Single	float x
Y As Single	float y
Z As Single	float z
U As Single	float u
V As Single	float v
W As Single	float w
R As Single	float r
S As Single	float s
T As Single	float t
Hand As SpelHand	SpelHand Hand
Elbow As SpelElbow	SpelElbow Elbow
Wrist As SpelWrist	SpelWrist Wrist
Local As Integer	int Local
J1Flag As Integer	int J1Flag
J2Flag As Integer	int J2Flag
J4Flag As Integer	int J4Flag
J6Flag As Integer	int J6Flag
J1Angle As Single	float J1Angle
J4Angle As Single	float J4Angle

14.13.2 SpelPoint方法

Sub Clear ()

清除所有點資料。

Sub New ()

預設的建構函式。建立空的點(所有資料會清除)。

Sub New (X As Single, Y As Single, Z As Single, U As Single [, V As Single] [, W As Single])

指定座標之新建點的選用建構函式。

Function ToString ([Format As String]) As String

允許指定 Format 的 ToString 覆寫。此會傳回在 SPEL+中所定義的點。

格式可為：

Empty	傳回整個點(含所有座標及屬性)。
"XY"	傳回"XY(...)"
"XYST"	傳回"XY(...) :ST(...)"

14.14 SpelRobotInfo類別

成員名稱	類型	描述
RobotModel	String	機器人型號
RobotName	String	機器人名稱
RobotNumber	Integer	機器人編號
RobotSerial	String	機器人序列號
RobotType	SpelRobotType	機器人類型

以下是範例。

VB 例:

```
Dim info As SpelRobotInfo
info = m_spel.GetRobotInfo()
Label1.Text = info.RobotNumber + " " + info.RobotModel
```

C# 例:

```
SpelRobotInfo info;
info = m_spel.GetRobotInfo();
Label1.Text = info.RobotNumber + " " + info.RobotModel;
```

14.15 SpelTaskInfo類別

成員名稱	類型	描述
CPU	Integer	SPEL 任務的 CPU 負載率
ErrorCode	Integer	錯誤代碼
StartTime	DateTime	任務開始日期
TaskName	String	任務名稱
TaskNumber	Integer	任務編號
TaskState	SpelTaskState	任務狀態

以下是範例。

VB 例:

```
Dim info As SpelTaskInfo
info.TaskState = m_spel.TaskState(1)
Label1.Text = info.TaskNumber + " " + info.TaskState
```

C# 例:

```
SpelTaskInfo info;
info.TaskState = m_spel.TaskState(1);
Label1.Text = info.TaskNumber + " " + info.TaskState;
```

14.16 列舉

14.16.1 SpelArmDefMode列舉

成員名稱	數值	描述
Rough	1	使用一個姿態定義手臂。
Fine	1	使用兩個姿態定義手臂。

14.16.2 SpelArmDefType列舉

成員名稱	數值	描述
J2Camera	1	定義 J2 安裝攝影機的手臂。

14.16.3 SpelAxis列舉

成員名稱	數值	描述
X	1	X 軸
Y	2	Y 軸
Z	3	Z 軸
U	4	U 軸
V	5	V 軸
W	6	W 軸
R	7	R 軸
S	8	S 軸
T	9	T 軸

14.16.4 SpelBaseAlignment列舉

成員名稱	數值	描述
XAxis	0	對齊 X 軸。
YAxis	1	對齊 Y 軸。

14.16.5 SpelCalPlateType列舉

成員名稱	數值	描述
None	0	無校準板。
Large	1	大校準板。
Medium	2	中校準板。
Small	3	小校準板。
XSmall	4	超小校準板。

14.16.6 SpelConnectionType列舉

成員名稱	數值	描述
USB	1	USB 連線。
Ethernet	2	乙太網連線。
Virtual	3	與虛擬控制器的連線。

14.16.7 SpelDialogs列舉

成員名稱	數值	描述
RobotManager	1	工具 | 機器人管理器對話方塊的 ID
ControllerTools	2	工具 | 控制器對話方塊的 ID
VisionGuide	3	工具 | Vision Guide 對話方塊的 ID
ForceGuide	4	Force Guide 對話方塊的 ID
PartFeeding	5	Part Feeding 對話方塊的 ID

14.16.8 SpelElbow列舉

成員名稱	數值	描述
Above	1	肘部方向為上方。
Below	2	肘部方向為下方。

14.16.9 SpelEvents列舉

成員名稱	數值	描述
Pause	1	pause 事件的 ID。
SafeguardOpen	2	安全防護打開事件的 ID。
SafeguardClose	3	安全防護關閉事件的 ID。
ProjectBuildStatus	4	專案建置狀態事件的 ID。
Error	5	錯誤事件的 ID。
Print	6	列印事件的 ID。
EstopOn	7	緊急停止 on 事件的 ID。
EstopOff	8	緊急停止 off 事件的 ID。
Continue	9	continue 事件的 ID。
MotorOn	10	馬達 on 事件的 ID。
MotorOff	11	馬達 off 事件的 ID。
PowerHigh	12	運行功率 high 事件的 ID。
PowerLow	13	運行功率 low 事件的 ID。
TeachMode	14	示教模式事件的 ID。
AutoMode	15	自動模式事件的 ID。
TaskState	16	任務狀態事件的 ID。
Shutdown	17	shutdown 事件的 ID。
AllTasksStopped	18	所有任務停止事件的 ID。
Disconnected	19	斷開事件的 ID
MotionStarted	20	控制命令開始事件的 ID
MotionComplete	21	控制命令停止事件的 ID

14.16.10 SpelForceAxis列舉

成員名稱	數值	描述
XForce	1	指定 X 力軸。
YForce	2	指定 Y 力軸。
ZForce	3	指定 Z 力軸。
XTorque	4	指定 X 力矩軸。
YTorque	5	指定 Y 力矩軸。
ZTorque	6	指定 Z 力矩軸。

14.16.11 SpelForceCompareType列舉

成員名稱	數值	描述
LessOrEqual	0	當力小於或等於指定閾值時，會觸發 Till。
GreaterOrEqual	1	當力大於或等於指定閾值時，會觸發 Till。

14.16.12 SpelForceProps列舉

成員名稱	數值	描述
EndStatus	28340	力覺引導序列或力覺引導物件的結束狀態
ForceCondOK	28440	力覺引導物件的力的結束條件的達成狀態
IOCondOK	28590	力覺引導物件的 I/O 結束條件的達成狀態
PosCondOK	28860	力覺引導物件的位置結束條件的達成狀態
Time	29070	力覺引導序列或力覺引導物件的執行時間
TimeOut	29080	力覺引導物件超時的達成狀態
LastExecObject	30100	最後被執行的力覺引導物件的名稱
MeasuredHeight	30500	高度檢查序列測量出的高度
FailedStatus	30510	力覺引導序列的失敗原因
AvgForcesFx	100211	力覺引導物件執行時，Fx 力的平均值
AvgForcesFy	100212	力覺引導物件執行時，Fy 力的平均值
AvgForcesFz	100213	力覺引導物件執行時，Fz 力的平均值
AvgForcesTx	100214	力覺引導物件執行時，Tx 轉矩的平均值
AvgForcesTy	100215	力覺引導物件執行時，Ty 轉矩的平均值
AvgForcesTz	100216	力覺引導物件執行時，Tz 轉矩的平均值
EndForcesFx	102411	力覺引導序列或力覺引導物件結束時 Fx 的力
EndForcesFy	102412	力覺引導序列或力覺引導物件結束時 Fy 的力
EndForcesFz	102413	力覺引導序列或力覺引導物件結束時 Fz 的力
EndForcesTx	102414	力覺引導序列或力覺引導物件結束時 Tx 的轉矩
EndForcesTy	102415	力覺引導序列或力覺引導物件結束時 Ty 的轉矩
EndForcesTz	102416	力覺引導序列或力覺引導物件結束時 Tz 的轉矩
EndPosX	102421	力覺引導物件結束時的位置 (X 坐標)
EndPosY	102422	力覺引導物件結束時的位置 (Y 坐標)
EndPosZ	102423	力覺引導物件結束時的位置 (Z 坐標)
EndPosU	102424	力覺引導物件結束時的位置 (U 坐標)
EndPosV	102425	力覺引導物件結束時的位置 (V 坐標)
EndPosW	102426	力覺引導物件結束時的位置 (W 坐標)
PeakForcesFx	105711	力覺引導序列或力覺引導物件執行時，Fx 力的峰值
PeakForcesFy	105712	力覺引導序列或力覺引導物件執行時，Fy 力的峰值
PeakForcesFz	105713	力覺引導序列或力覺引導物件執行時，Fz 力的峰值

PeakForcesTx	105714	力覺引導序列或力覺引導物件執行時，Tx 轉矩的峰值
PeakForcesTy	105715	力覺引導序列或力覺引導物件執行時，Ty 轉矩的峰值
PeakForcesTz	105716	力覺引導序列或力覺引導物件執行時，Tz 轉矩的峰值
TriggeredForcesFx	109411	力覺引導物件的力達成結束條件時 Fx 的力
TriggeredForcesFy	109412	力覺引導物件的力達成結束條件時 Fy 的力
TriggeredForcesFz	109413	力覺引導物件的力達成結束條件時 Fz 的力
TriggeredForcesTx	109414	力覺引導物件的力達成結束條件時 Tx 的轉矩
TriggeredForcesTy	109415	力覺引導物件的力達成結束條件時 Ty 的轉矩
TriggeredForcesTz	109416	力覺引導物件的力達成結束條件時 Tz 的轉矩
TriggeredPosX	109421	力覺引導物件的力達成結束條件時的位置(X 坐標)
TriggeredPosY	109422	力覺引導物件的力達成結束條件時的位置(Y 坐標)
TriggeredPosZ	109423	力覺引導物件的力達成結束條件時的位置(Z 坐標)
TriggeredPosU	109424	力覺引導物件的力達成結束條件時的位置(U 坐標)
TriggeredPosV	109425	力覺引導物件的力達成結束條件時的位置(V 坐標)
TriggeredPosW	109426	力覺引導物件的力達成結束條件時的位置(Z 坐標)

14.16.13 SpelHand列舉

成員名稱	數值	描述
Righty	1	右手姿態。
Lefty	2	左手姿態。

14.16.14 SpelIOLabelTypes列舉

成員名稱	數值	描述
InputBit	1	指定輸入位元。
InputByte	2	指定輸入位元組。
InputWord	3	指定輸入字元。
OutputBit	4	指定輸出位元。
OutputByte	5	指定輸出位元組。
OutputWord	6	指定輸出字元。
MemoryBit	7	指定記憶體位元。
MemoryByte	8	指定記憶體位元組。
MemoryWord	9	指定記憶體字元。
InputReal	10	指定實數輸入。
OutputReal	11	指定實數輸出。

14.16.15 SpelLocalDefType 列舉

成員名稱	數值	描述
J5Camera	1	定義 J5 設定攝像機的本地
J6Camera	2	定義 J6 設定攝像機的本地
FixedUpwardCamera	3	使用固定向上攝像機，定義本地
FixedDownwardCamera	4	使用固定向下攝像機，定義本地

14.16.16 SpelOperationMode列舉

成員名稱	數值	描述
Auto	1	EPSON RC+ 7.0 處於 auto 模式。
Program	2	EPSON RC+ 7.0 處於 program 模式。

14.16.17 SpelOptions 列舉

成員名稱	數值	描述
ECP	1	ECP 選配件
API	2	RC+ API 選配件
PCVision	3	PC 視覺選配件
ConveyorTracking	5	輸送帶追蹤選配件
GUIBuilder	6	GUI Builder 選配件
OCR	7	OCR 選配件
FieldbusMaster	8	現場匯流排主站選配件
LegacyForceSensing	9	Force Sensing 選配件
PartFeeding	11	Part Feeding 選配件
ThirdPartyForceSensors	13	第三方力覺感應器選配件

14.16.18 SpelOptionStatus 列舉

成員名稱	數值	描述
Inactive	0	無效
Active	1	有效

14.16.19 SpelRobotPosType列舉

成員名稱	數值	描述
World	0	指定世界座標。
Joint	1	指定關節座標。
Pulse	2	指定脈衝。

14.16.20 SpelRobotType列舉

成員名稱	數值	描述
Joint	1	機器人類型為 joint。
Cartesian	2	機器人類型為 Cartesian。
Scara	3	機器人類型為 SCARA。
Cylindrical	4	機器人類型為 Cylindrical。
SixAxis	5	機器人類型為 6 軸。
RS	6	機器人類型為 SCARA RS 系列。
N	7	機器人類型為 N 系列。

14.16.21 SpelShutdownMode列舉

成員名稱	數值	描述
ShutdownWindows	0	Windows 將會關閉。
RebootWindows	1	Windows 將會重啟。

14.16.22 SpelSimObjectType 列舉

成員名稱	數值	描述
Unknown	-1	未設定物件類型
Layout	0	佈局物件
Part	1	零件物件
MountedDevice	3	機械臂安裝設備

14.16.23 SpelSimProps 列舉

成員名稱	數值	描述
PositionX	100	X 坐標位置
PositionY	200	Y 坐標位置
PositionZ	300	Z 坐標位置
RotationX	400	X 軸旋轉角度
RotationY	500	Y 軸旋轉角度
RotationZ	600	Z 軸旋轉角度
CollisionCheck	700	啟用/禁止碰撞偵測
CollisionCheckSelf	800	啟用/禁止自碰撞偵測
Visible	900	顯示/不顯示狀態
Type	1000	物件的類型
HalfSizeX	1500	Box 物件 X 方向的長度
HalfSizeY	1600	Box 物件 Y 方向的長度
HalfSizeZ	1700	Box 物件 Z 方向的長度
HalfSizeHeight	1800	Plane 物件的高度
HalfSizeWidth	1900	Plane 物件的寬度
PlaneType	2000	Plane 物件的類型
Radius	2100	Sphere 或 Cylinder 物件的半徑
Height	2200	Cylinder 物件的高度
Name	2300	物件名稱
Color	2400	物件顯示顏色

14.16.24 SpelStopType列舉

成員名稱	數值	描述
StopNormalTasks	0	僅停止正常任務(非背景任務)。
StopAllTasks	1	停止所有任務，包含背景任務。

14.16.25 SpelTaskState列舉

成員名稱	數值	描述
Quit	0	任務處於 quit 狀態。
Run	1	任務處於 run 狀態。
Aborted	2	任務已終止。
Finished	3	任務已完成。
Breakpoint	4	任務位於中斷點。
Halt	5	任務處於 halt 狀態。
Pause	6	任務處於 pause 狀態。
Step	7	任務處於 step 狀態。
Walk	8	任務處於 walk 狀態。
Error	9	任務處於 error 狀態。
Waiting	10	任務處於 wait 狀態。

14.16.26 SpelTaskType列舉

成員名稱	數值	描述
Normal	0	標準任務。
NoPause	1	任務沒有受 pause 影響。
NoEmgAbort	2	任務沒有受緊急停止影響。

14.16.27 SpelToolDefType列舉

成員名稱	數值	描述
J4Camera	1	定義 J4 安裝攝影機的工具。
J6Camera	2	定義 J6 安裝攝影機的工具。
FixedCamera	3	透過使用未校準的固定攝影機定義工具。
FixedCameraWithCal	4	透過使用已校準的上方攝影機定義工具。

14.16.28 SpelToolDefType3D 列舉

成員名稱	數值	描述
Bar	1	為條形類型定義 3D 工具。
Plane	2	為平面類型定義 3D 工具。

14.16.29 SpelUserRights列舉

成員名稱	數值	描述
All	-1	使用者擁有所有權限。
None	0	使用者沒有權限。
EditSecurity	1	使用者可設定安全功能。
SysConfig	2	使用者可改變系統設定。
EditPrograms	4	使用者可編輯程式。
EditPoints	8	使用者可編輯點。
EditVision	16	使用者可改變視覺屬性。
JogAndTeach	32	使用者可步進示教。
CommandWindow	64	使用者可使用命令視窗。
EditRobotParameters	128	使用者可編輯機器人參數。
ConfigureOptions	256	使用者可設定選購件。
ViewAudit	512	使用者可查看安全審計日誌。
EditProject	1024	使用者可編輯專案設定。
DeleteAudit	2048	使用者可刪除安全審計日誌項目。
TeachPoints	4096	使用者可示教點。
ChangeOutputs	8192	使用者可改變輸出狀態。
ChangeMemIO	16384	使用者可改變記憶體 I/O 狀態。
EditGUIBuilder	32768	使用者可在 GUI Builder 中進行改變。
EditForce	65536	使用者可在 Force Guide 與 Force Control 中進行改變。
EditPartFeeding	131072	使用者可在工件供給中進行改變。

14.16.30 SpelVDefShowWarning列舉

成員名稱	數值	描述
None	-1	不顯示警告。
Always	0	永遠顯示警告。
DependsOnSpeed	1	RobotSpeed 或 RobotAccel 大於 5 時顯示。

14.16.31 SpelVisionImageSize列舉

成員名稱	數值	描述
Size320x240	1	320 x 240 影像大小。
Size640x480	2	640 x 480 影像大小。
Size800x600	3	800 x 600 影像大小。
Size1024x768	4	1024 x 768 影像大小。
Size1280x1024	5	1280 x 1024 影像大小。
Size1600x1200	6	1600 x 1200 影像大小。
Size2048x1536	7	2048 x 1536 影像大小。
Size2560x1920	8	2560 x 1920 影像大小。
Size3664x2748	9	3664 x 2748 影像大小。
Size5472x3648	10	5472 x 3648 影像大小。
Size4024x3036	11	4024 x 3036 影像大小。

14.16.32 SpelVisionObjectTypes列舉

成員名稱	數值	描述
Correlation	1	Correlation 物件。
Blob	2	Blob 物件。
Edge	3	Edge 物件。
Polar	4	Polar 物件。
Line	5	Line 物件。
Point	6	Point 物件。
Frame	7	Frame 物件。
ImageOp	8	ImageOp 物件。
OCR	9	OCR 物件。
CodeReader	10	CodeReader 物件。
Geometric	11	Geometric 物件。
ColorMatch	14	Color Match 物件。
LineFinder	15	Line Finder 物件。
ArcFinder	16	Arc Finder 物件。
DefectFinder	17	Defect Finder 物件。
LineInspector	18	Line Inspector 物件。
ArcInspector	19	Arc Inspector 物件。
BoxFinder	20	Box Finder 物件。
CornerFinder	21	Corner Finder 物件。
Contour	22	Contour 物件。
Text	23	Text 物件。
Decision	26	Decision 物件
Coordinates	27	Coordinates 物件

14.16.33 SpelVisionProps列舉

此列舉用於所有視覺屬性與結果。如需詳細資訊，請參閱 Vision Guide Reference 手冊。

14.16.34 SpelWrist列舉

成員名稱	數值	描述
NoFlip	1	腕部方向為不翻轉。
Flip	2	腕部方向為翻轉。

14.16.35 SpelWindows列舉

成員名稱	數值	描述
IOMonitor	1	I/O 監視器視窗的 ID。
TaskManager	2	任務管理器視窗的 ID。
ForceMonitor	3	力監視器視窗的 ID。
Simulator	4	模擬器視窗的 ID。

14.17 Spel錯誤編號及訊息

如需錯誤編號及錯誤訊息的詳細資訊，請參閱狀態碼和錯誤碼手冊。

15. 32位元與64位元應用程式

EPSON RC+ 7.0 之 7.1.0 及更新版本中提供的 RCAPINet 程式庫自動支援 32 位元與 64 位元應用程式。

對於 EPSON RC+ 7.0 之 7.1.0 之前的版本，則有提供獨立的程式庫以支援 32 位元與 64 位元。為確保相容性，第 7.5.0 版仍有提供這些過時的程式庫(SpelNetLib70.dll 和 SpelNetLib70_x64.dll)。如需過時程式庫的詳細資訊，請參閱所使用舊版 EPSON RC+ 7.0 的 RC+ API 手冊。

16. 使用LabVIEW VI程式庫

16.1 概述

在 EPSON RC+ 7.0 v7.1.0 之前的版本中，API .NET 程式庫可直接用於 LabVIEW。EPSON RC+ 7.0 v7.1.0 採用了新的 LabVIEW VI 程式庫。新程式庫具有以下功能：

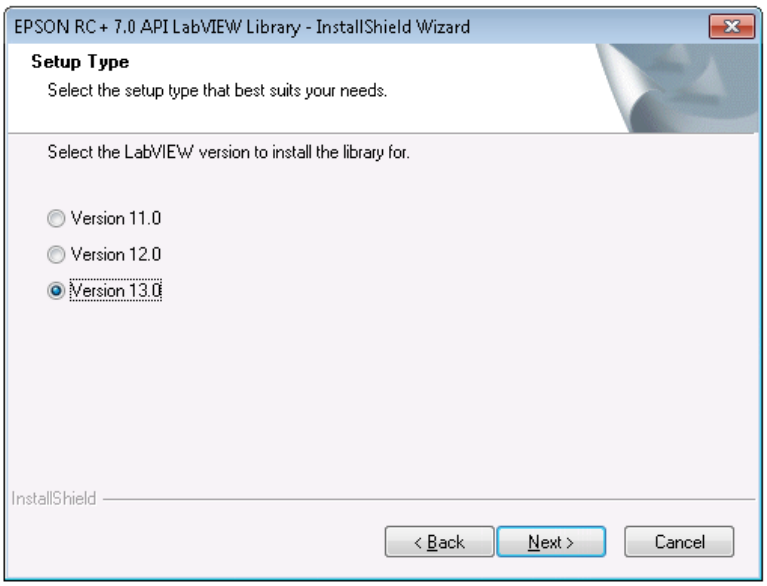
- 使用 VI(虛擬儀器)的 EPSON RC+ 7.0 高階層介面。
- 使用者無須處理 EPSON RC+ 7.0 的.NET 介面 – 此為自動處理。
- 每個 Spel 命令都包含在個別的 VI 中。
- VI 是在多個工具面板中組織。
- 同時支援 32 位元與 64 位元 LabVIEW 應用程式。
- 支援 LabVIEW 2009 及更新版本。

若要使用 LabVIEW VI 程式庫，您必須針對所連接的每個控制器購買 EPSON RC+ 7.0 API 軟體授權。

16.2 安裝

若要使用 EPSON RC+ 7.0 LabVIEW VI 程式庫，您必須使用 PC 上的 \EpsonRC70\API\LabVIEW 資料夾中的安裝程式進行安裝。

1. 安裝 LabVIEW 2009 或更新版本。
2. 瀏覽至電腦上的\EpsonRC70\API\LabVIEW 資料夾，並執行 EpsonRC70_vxxx_LabVIEW.exe 安裝程式，其中 xxx 表示 EPSON RC+ 7.0 的版本編號。例如：EpsonRC70_v710_LabVIEW.exe。
3. 安裝程式啟動時，將會顯示在 PC 上偵測到的 LabVIEW 安裝版本。預設為選擇最新版本。選擇要使用 EPSON RC+ 7.0 LabVIEW VI 程式庫的版本。



4. 按一下 Next，然後按一下 Install。所選 LabVIEW 版本的 VI、控制項及面板將會安裝。

16.3 工具和控制面板

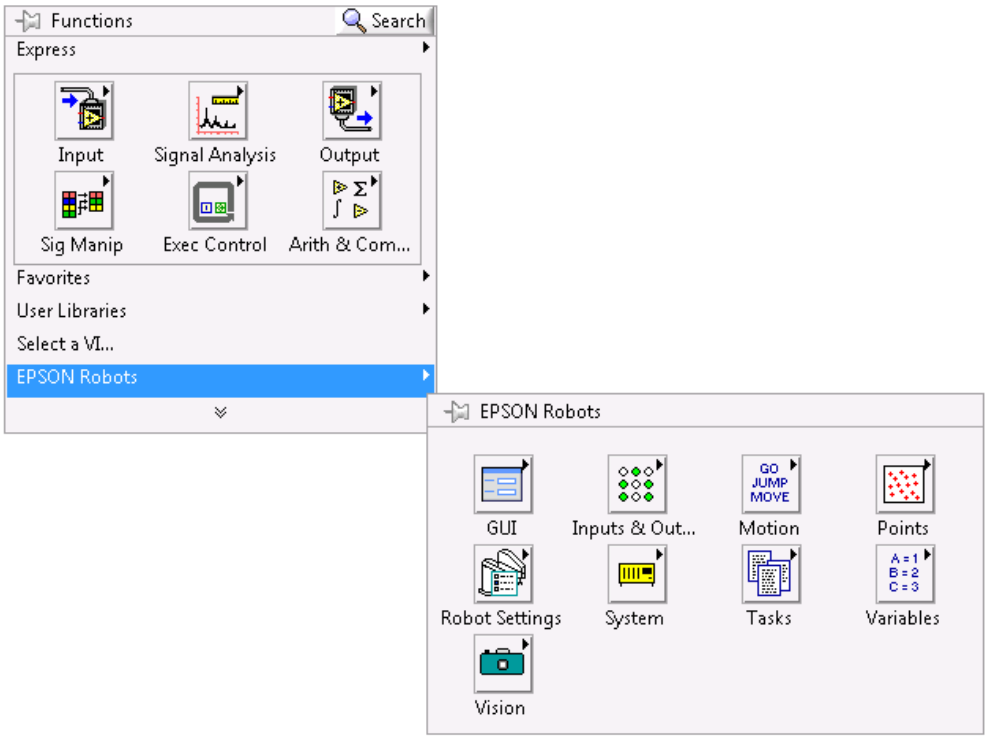
安裝 EPSON RC+ 7.0 LabVIEW VI 程式庫後，您可從[EPSON Robots]工具面板和[EPSON Robots]控制面板存取可用的 VI 及控制項。

工具面板

工具面板是從方塊圖存取。Epson Robots 工具面板內有多個子面板，如下表所示：

面板	描述
System	用於初始化及關閉 API。
Robot Settings	變更機器人參數。
Points	載入、儲存、變更機器人點。
Motion	執行機器人動作。
Inputs & Outputs	控制及監控控制器輸入與輸出。
Tasks	管理機器人控制器中的任務。
Variables	讀取及寫入控制器中的變數。
Vision	執行 Vision 命令。
GUI	顯示 GUI 功能。

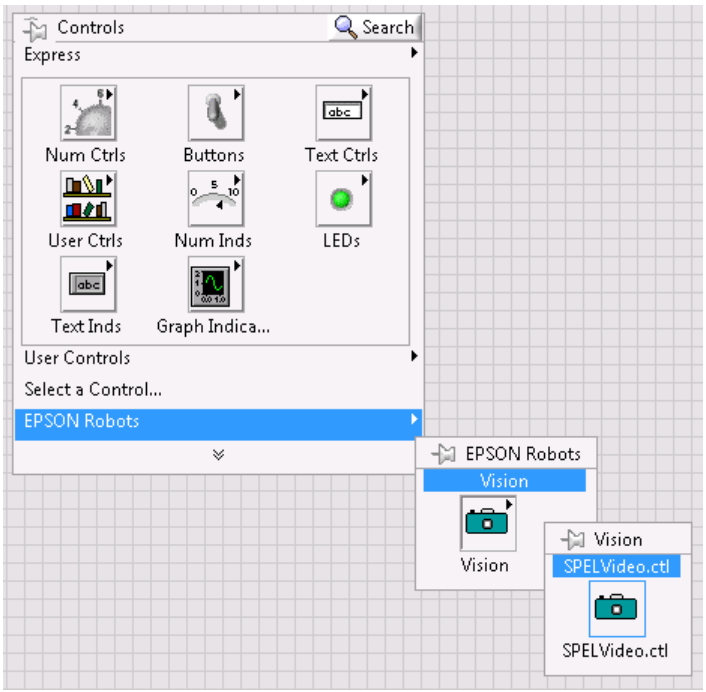
若要存取[Epson Robots]工具面板，請開啟 VI 中的方塊圖、用滑鼠右鍵按一下空白區域，然後選擇[Epson Robots]查看上述子面板。



控制面板

控制面板是從前面板存取。

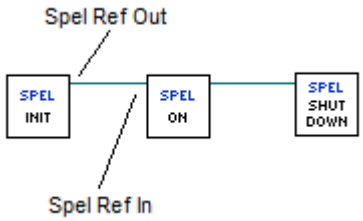
面板	描述
Vision	包含用於顯示視訊的 SPELVideo 控制項。



16.4 開始使用

若要使用 LabVIEW VI 程式庫，您的應用程式必須先針對要使用的每個控制器調用 Spel Initialize VI。Initialize VI 會啟動 EPSON RC+ 7.0 伺服器程序，將會連接至機器人控制器並處理後續的 Spel 命令 VI。Initialize VI 具有一個 Spel Ref Out 輸出。這必須連接至下一個 Spel VI Spel Ref In 輸入。對於每個後續 VI，上一個 Spel VI 的 Spel Ref Out 輸出必須連接至下一個 Spel VI 的 Spel Ref In 輸入。

例如：以下流程圖顯示每個 Spel 節點之間的 Spel Ref Out 及 Spel Ref In 連接。

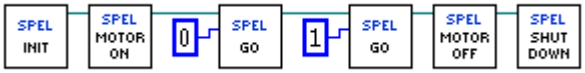


當應用程式關閉時，您必須調用 Spel Shutdown VI。這將會中斷與機器人控制器的連接，並關閉關聯的 EPSON RC+ 7.0 伺服器程序。

依照下列步驟開始使用。首先，您將會從 LabVIEW 預設 Spel+專案的 EPSON RC+ 7.0 GUI 中建立兩個安全機器人點。接著，您將會在 LabVIEW 中建置一個小應用程式，用以在兩點之間移動機器人。

1. 確定 EPSON RC+ 7.0 和 EPSON RC+ 7.0 LabVIEW VI 程式庫已安裝在 PC 上。如需安裝 LabVIEW VI 程式庫的詳細資訊，請參閱第 16.2 節。
2. 啟動 EPSON RC+ 7.0。
3. 從 Project 功能表選擇 Open，然後瀏覽至 LabVIEW 資料夾並選擇 LabVIEW_Default 專案。按一下 Open。
4. 從 Tools 功能表中，選擇 Robot Manager。按一下 Motor On 按鈕。
5. 在 Robot Manager 上，選擇 Jog & Teach 頁面。將機器人步進至某個安全位置。
6. 按一下 Teach 示教點 0。
7. 將機器人步進至其他安全位置。
8. 從 Point 清單選擇 P1，然後按一下 Teach 示教點 1。
9. 按一下主要工具條上的 Save 按鈕儲存這些點。
10. 關閉 EPSONRC+ 7.0。
11. 啟動 LabVIEW 並建立新的 VI。
12. 開啟新 VI 的方塊圖。
13. 從 Epson RC+ API | System 工具面板，將 Init VI 拖曳至方塊圖。您所連接的每個控制器都需要 Initialize VI。
14. 從 Epson RC+ API | Robot Settings 工具面板，將 MotorOn VI 拖曳至方塊圖。將 Inititalize VI 的 Spel Ref Out 輸出連接至 MotorOn VI 的 Spel Ref In 輸入。
15. 從 Epson RC+ API | Motion 工具面板，將 Go VI 拖曳至方塊圖。將 MotorOn VI 的 Spel Ref Out 輸出連接至 Go VI 的 Spel Ref In 輸入。新增 Point Number 輸入的常數並將數值設為 0。

16. 從 Epson RC+ API | Motion 工具面板，將其他 Go VI 拖曳至方塊圖。將上一個 Go VI 的 Spel Ref Out 輸出連接至第二個 Go VI 的 Spel Ref In 輸入。新增 Point Number 輸入的常數並將數值設為 1。
17. 從 Epson RC+ API | Robot Settings 工具面板，將 MotorOff VI 拖曳至方塊圖。將 Go VI 的 Spel Ref Out 輸出連接至 MotorOff VI 的 Spel Ref In 輸入。
18. 從 Epson RC+ API | System 工具面板，將 Shutdown VI 拖曳至方塊圖。每個 Init VI 都必須使用 Shutdown VI。
    方塊圖應看起來如下：



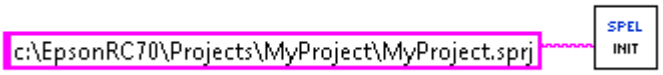
19. 執行應用程式。機器人馬達應該開啟，機器人應移至點 0，再移至點 1，然後機器人馬達將會關閉。

16.5 使用Spel+專案

您可選擇以 LabVIEW 應用程式使用 Spel+專案。不過，如果您要儲存點資料或想使用點標籤及／或 I/O 標籤、任務或視覺序列，則您需要使用 Spel+專案。

根據預設，專案為\EpsonRC70\Projects\LabVIEW 資料夾中的 LabVIEW_Default。

如有需要，您可使用 EPSON RC+ 7.0 建立專屬的專案，然後使用 Initialize VI *Project* 輸入參數指定您要使用的專案，如下所示：



若要使用 EPSON RC+ 7.0 專案，請啟動 EPSON RC+ 7.0 應用程式。使用 Project 功能表建立、開啟及編輯專案。如需詳細資訊，請參閱 EPSON RC+ 7.0 使用指南。

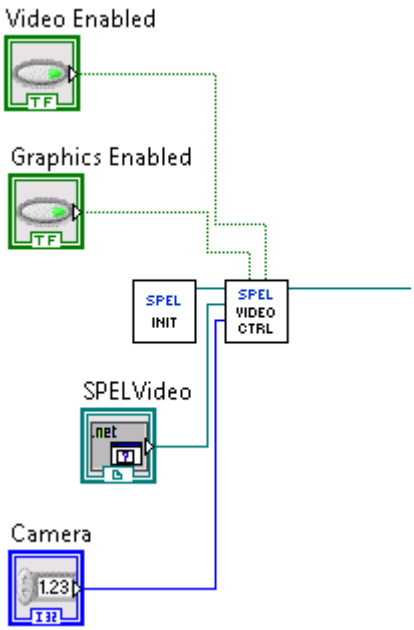
16.6 顯示視訊

您可使用 SPELVideo 控制項及 VideoControl VI 顯示 Vision Guide 序列的視訊。

若要顯示視訊：

1. 將 SPELVideo 控制項添加至前面板。
2. 將 VideoControl.vi 添加至對應的方塊圖。
3. 在 VideoControl VI 上，將 SPELVideo 控制項的輸出連接至 SPELVideo Ref In 輸入。
4. 連接 VideoControl VI 的 Spel Ref In 和 Spel Ref Out 參數。
5. 在 VideoControl VI 上，添加 *Camera*、*Graphics Enabled* 及 *Video Enabled* 參數的常數或控制項。Video Enabled 必須設為 true 才能顯示視訊。

下方流程圖顯示 SPELVideo 控制項與 SPEL VideoControl VI 的連接。



當 *Video Enabled* 為 true，且 VRun 從 VRun VI 或在控制器任務中執行時，您將會看見產生的視訊(視 Camera 設定而定)。

根據預設，*Camera* 輸入參數為零，這會允許顯示任何攝影機的視訊。如果將 *Camera* 設為零以外的數值，將會針對使用指定攝影機的序列顯示視訊。

當 *Graphics Enabled* 為 true 且 VRun 執行時，序列結果圖形會顯示在視訊影像之上。

您在應用程式中一次只能使用一個 SPEL Video 控制項。

16.7 VI參考

本節包含 EPSON RC+ 7.0 LabVIEW VI 程式庫中所有 VI 的資訊。

以下為每個 VI 的資訊：

工具面板	包含 VI 的工具面板。
描述	函數的簡短描述。
輸入	輸入參數
輸出	輸出參數
備註	其他詳細資訊。

Accel VI

工具面板

Epson Robots | Robot Settings

描述

設定目前機器人的 PTP 加速和減速。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Accel	PTP 加速的整數值。
Decel	PTP 減速的整數值。
Depart Accel	選用。Jump 起始加速的整數值。
Depart Decel	選用。Jump 起始減速的整數值。
Appro Accel	選用。Jump 接近加速的整數值。
Appro Decel	選用。Jump 接近減速的整數值。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

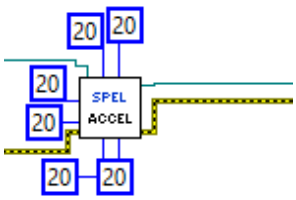
備註

使用 Accel 設定目前機器人的 PTP 加速和減速值。所有值可介於 1 至 100%。如果指定 Depart Accel，則必須同時指定剩餘輸入。

另請參閱

AccelS, Speed, SpeedS

Accel 範例



AccelS VI

工具面板

Epson Robots | Robot Settings

描述

設定目前機器人的線性加速和減速。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Accel	線性加速的雙精度值。
Decel	線性減速的雙精度值。
Depart Accel	選用。Jump 起始加速的雙精度值。
Depart Decel	選用。Jump 起始減速的雙精度值。
Appro Accel	選用。Jump 接近加速的雙精度值。
Appro Decel	選用。Jump 接近減速的雙精度值。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

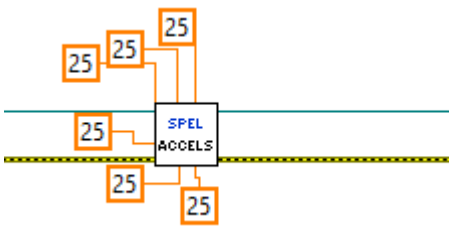
備註

使用 AccelS 設定目前機器人的線性加速和減速值。所有值以 mm/sec^2 為單位。如果指定 Depart Accel，則必須同時指定剩餘輸入。

另請參閱

Accel, Speed, SpeedS

AccelS 範例



Arc VI

工具面板

Epson Robots | Motion

描述

Arc 利用 XY 平面中的圓形插補將手臂移至指定點。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Mid Point Number	使用整數指定中點。
Mid Point Expr	透過使用字串運算式來指定中點。如果使用此輸入，您必須同時使用字串運算式指定結束點。
End Point Number	使用整數指定結束點。
End Point Expr	透過使用字串運算式來指定結束點。您可加入 ROT、CP、SYNC，Till 的搜尋運算式，以及平行處理陳述式。

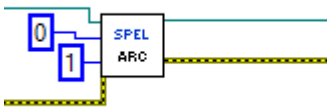
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Accel, Arc3, BMove, Go, Jump, Jump3, Move, Speed, TGo, TMove

Arc 範例



Arc3 VI

工具面板

Epson Robots | Motion

描述

Arc3 在 3D 中利用圓形插補將手臂移至指定點。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Mid Point Number	使用整數指定中點。
Mid Point Expr	透過使用字串運算式來指定中點。如果使用此輸入，您必須同時使用字串運算式指定結束點。
End Point Number	使用整數指定結束點。
End Point Expr	透過使用字串運算式來指定結束點。使用字串運算式時，您可加入 CP、SYNC，Till 的搜尋運算式，以及平行處理陳述式。

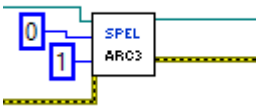
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

AccelS, Arc, BMove, Go, Jump, Jump3, Move, SpeedS, TGo, TMove

Arc3 範例



Arch VI

工具面板

Epson Robots | Robot Settings

描述

定義要搭配 JUMP 指令使用的 ARCH 參數(開始水平動作之前要移動的 Z 高度)。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Arch Number	在開始水平移動之前，Jump 指令開始時移動的起始距離(公釐)。
Depart Dist	在開始水平移動之前，Jump 指令開始時移動的起始距離(公釐)。
Appro Dist	Jump 指令目標位置以上的結束距離(公釐)。

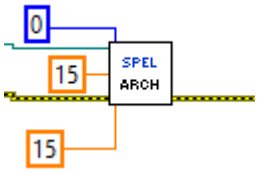
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Jump, Jump3

Arch 範例



Arm VI

工具面板

Epson Robots | Robot Settings

描述

選擇目前機器人手臂。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Arm Number	介於 0-15 的整數。使用者最多可選擇 16 種不同的手臂。手臂 0 是標準(預設)機器人手臂。手臂 1-15 是透過 ArmSet 指令定義的輔助手臂。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Armset, GetArm, Tool

Arm 範例

Armset VI

工具面板

Epson Robots | Robot Settings

描述

Defines an auxiliary robot arm.

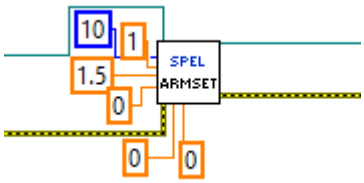
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
ArmNumber	整數：有效範圍介於 1-15。
Param1	(SCARA 機器人適用)肘部中心線到新方向軸中心線的水平距離。(亦即新輔助手臂方向軸中心線所在的位置。) (Cartesian 機器人適用)原始 X 位置的 X 軸方向位置偏移(公釐)。
Param2	(SCARA 機器人適用)從標準肘部中心線與標準方向軸中心線所形成的直線，到新輔助手臂肘部中心線與新方向軸中心線所形成的直線，兩者之間的偏移(度)。(這兩條直線應該在肘部中心線相交，形成的角度即為 *Param2*。) (Cartesian 機器人適用)原始 Y 位置的 Y 軸方向位置偏移(公釐)。
Param3	(SCARA & Cartesian 機器人適用)新方向軸中心與舊方向軸中心之間的 Z 高度偏移差距。(此為距離。)
Param4	(SCARA 機器人適用)肩部中心線到新輔助軸肘部方向之肘部中心線的距離。 (Cartesian 機器人適用)此為虛擬參數(指定 0)
Param5	(SCARA & Cartesian 機器人適用)新方向軸與舊方向軸的角度偏移(度)。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

ArmSet 範例



AtHome VI

工具面板

Epson Robots | Motion

描述

若目前機器人位於起始點位置，會傳回 True。

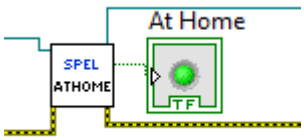
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
At Home	表示目前機器人是否位於起始點位置的布林值。

AtHome 範例



AvoidSing VI

工具面板

Epson Robots | Motion

描述

啟動／停用 Move、Arc 及 Arc3 動作方法的奇點避開功能。

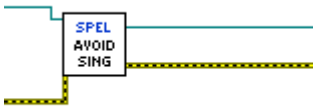
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Enable	True 可啟用奇點避開，False 則可停用。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

AvoidSing 範例



BGo VI

工具面板

Epson Robots | Motion

描述

在所選本地座標系統中執行 PTP 相對動作。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

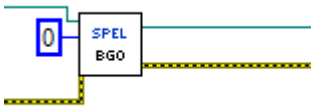
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BMove, Go, Jump, Jump3, Move, Speed, TGo, TMove

BGo 範例



BMove VI

工具面板

Epson Robots | Motion

描述

在所選本地座標系統中執行線性插補相對動作。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

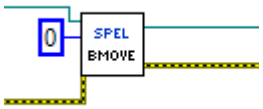
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

AccelS, Arc, Arc3, BGo, Go, Jump, Jump3, Move, SpeedS, TGo, TMove

BMove 範例



Box VI

工具面板

Epson Robots | Robot Settings

描述

指定在盒體內定義的接近檢查區域。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
AreaNumber	介於 1-15 的整數，代表 15 個方塊中要定義的方塊。
Min X	接近檢查區域的最小 X 座標位置。
Max X	接近檢查區域的最大 X 座標位置。
Min Y	接近檢查區域的最小 Y 座標位置。
Max Y	接近檢查區域的最大 Y 座標位置。
Min Z	接近檢查區域的最小 Z 座標位置。
Max Z	接近檢查區域的最大 Z 座標位置。
Polarity On	設定在使用對應遠端輸出時的遠端輸出邏輯。若要在夾具末端位於盒體區域中時將 I/O 輸出設為開啟，請使用 True。若要在夾具末端位於盒體區域中時將 I/O 輸出設為關閉，請使用 False。

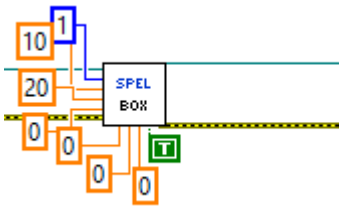
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

XYLim

Box 範例



Continue VI

工具面板

Epson Robots | Tasks

描述

使控制器中暫停的所有任務繼續。

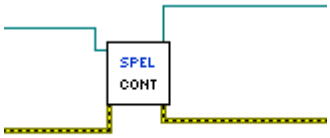
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

Continue 範例



Delay VI

工具面板

Epson Robots | System

描述

指定的毫秒數的延遲處理。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Milliseconds	延遲的毫秒數。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

Delay 範例

ECP VI

工具面板

Epson Robots | Robot Settings

描述

選擇目前的 ECP 定義。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
ECPNumber	介於 0-15 的整數，代表 16 個 ECP 定義中要搭配下一個動作指令使用的定義。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

ECPSet, GetECP

ECP 範例

ECPset VI

工具面板

Epson Robots | Robot Settings

描述

定義 ECP(外部控制點)。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
ECPNumber	介於 1-15 的整數，代表 15 個外部控制點中要定義的外部控制點。
X	外部控制點 X 座標。
Y	外部控制點 Y 座標。
Z	外部控制點 Z 座標。
U	外部控制點 U 座標。
V	外部控制點 V 座標。
W	外部控制點 W 座標。

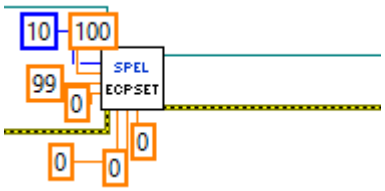
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

ECP, GetECP

ECPSet 範例



EStopOn VI

工具面板

Epson Robots | System

描述

返回緊急停止狀態。

輸入

Spel Ref In 來自上一個 Spel Ref Out 的 Spel 參考

Error In 來自上一個 Spel 節點的錯誤條件

輸出

Spel Ref Out 下一個要使用的 VI 的 Spel 參考輸出

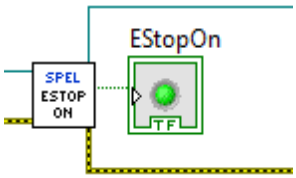
Error Out 後續 Spel 節點的錯誤條件輸出

EStopOn 緊急停止情況為 True，否則為 False

另請參閱

SafetyOn

EStopOn 範例



Find VI

工具面板

Epson Robots | Motion

描述

指定動作期間的座標儲存條件。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Condition	包含函數和運算符的字串運算式。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

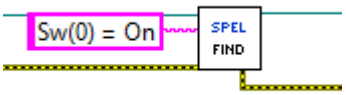
備註

使用 Find 指定在動作期間儲存位置的時間。符合條件時，目前位置會儲存至 FindPos。

另請參閱

FindPos

Find 範例



Fine VI

工具面板

Epson Robots | Robot Settings

描述

指定並顯示目標點的定位準確度。

輸入

Spel Ref In 來自上一個 Spel Ref Out 的 Spel 參考。

Error In 來自上一個 Spel 節點的錯誤條件。

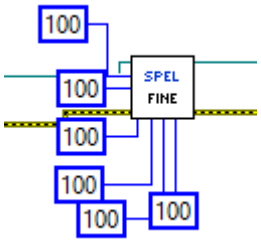
J1MaxErr – *J9MaxErr* 介於 0-32767 的整數，代表各點允許的定位誤差。 關節 7、8 及 9 的值為選用。

輸出

Spel Ref Out 下一個要使用的 VI 的 Spel 參考輸出。

Error Out 後續 Spel 節點的錯誤條件輸出。

Fine 範例



GetArm VI

工具面板

Epson Robots | Robot Settings

描述

傳回目前機器人的目前手臂編號。

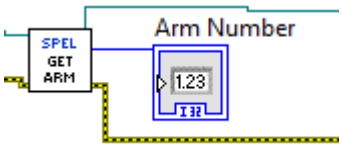
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Arm Number	目前手臂編號。

GetArm 範例



GetAvoidSing VI

工具面板

Epson Robots | Motion

描述

AvoidSingularity 戻り値を返す

輸入

Spel Ref In 來自上一個 Spel Ref Out 的 Spel 參考。

Error In 來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out 下一個要使用的 VI 的 Spel 參考輸出。

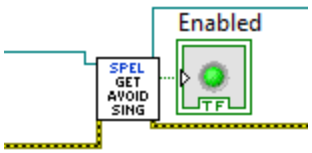
Error Out 後續 Spel 節點的錯誤條件輸出。

EStopOn 緊急停止情況為 True，否則為 False

另請參閱

AvoidSing

GetAvoidSing 範例



GetECP VI

工具面板

Epson Robots | Robot Settings

描述

傳回目前機器人的目前 ECP 編號。

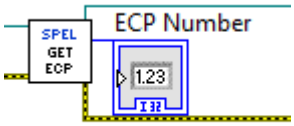
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
ECP Number	目前 ECP 編號。

GetECP 範例



GetLimZ VI

工具面板

Epson Robots | Motion

描述

傳回目前的 LimZ 設定

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

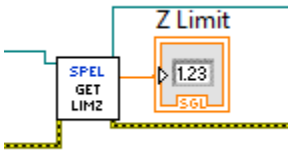
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Z Limit	目前的 LimZ 設定。

另請參閱

LimZ

GetLimZ 範例



GetMotor VI

工具面板

Epson Robots | Robot Settings

描述

傳回目前機器人的馬達開啟狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

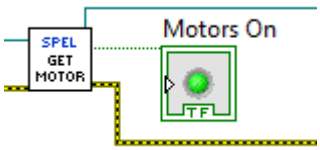
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Motors On	True 表示馬達開啟，false 表示未開啟。

另請參閱

GetPower, MotorOn, MotorOff

GetMotor 範例



GetOprMode VI

工具面板

Epson Robots | System

描述

讀取 EPSON RC+ 7.0 操作模式。

輸入

Spel Ref In 來自上一個 Spel Ref Out 的 Spel 參考。

Error In 來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out 下一個要使用的 VI 的 Spel 參考輸出。

Error Out 後續 Spel 節點的錯誤條件輸出。

Operation Mode 相關 EPSON RC+ 7.0 伺服器程序的操作模式。

模式	ID	描述
Auto	1	EPSON RC+ 7.0 處於 auto 模式。
Program	2	EPSON RC+ 7.0 處於 program 模式。

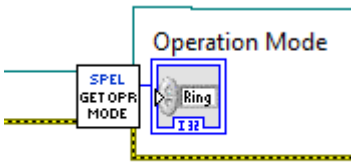
備註

當 *Operation Mode* 設為 Program 時，相關伺服器程序的 EPSON RC+ 7.0 GUI 會開啟，而控制器操作模式會設為 Program。若使用者關閉 RC+ GUI，*Operation Mode* 會設為 Auto。如果 *Operation Mode* 設為 Auto，則 RC+ GUI 會同時關閉。

另請參閱

OprMode

GetOprMode 範例



GetPoint VI

工具面板

Epson Robots | Points

描述

擷取某一機器人點的座標資料。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

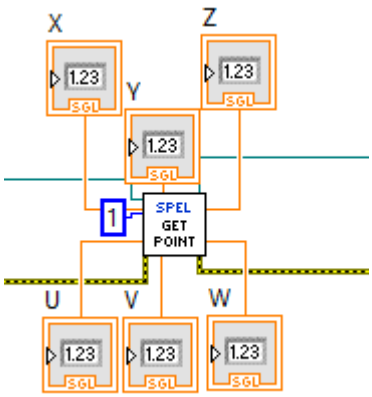
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
X – W	指定點的 X、Y、Z、U、V、W 座標。

另請參閱

LoadPoints, Robot, SavePoints, SetPoint

GetPoint 範例



GetPower VI

工具面板

Epson Robots | Robot Settings

描述

傳回目前機器人的高運行功率。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

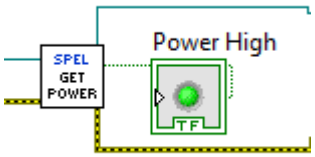
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Power High	True 表示功率高，否則會傳回 false。

另請參閱

PowerHigh, PowerLow

GetPower 範例



GetRobot VI

工具面板

Epson Robots | Robot Settings

描述

傳回目前機器人編號。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

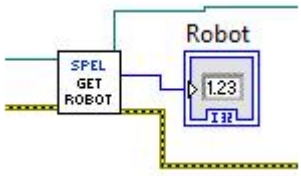
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Robot Number	目前機器人編號。

另請參閱

Robot

GetRobot 範例



GetTool VI

工具面板

Epson Robots | Robot Settings

描述

傳回目前機器人的目前工具編號。

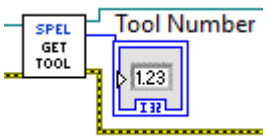
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Tool Number	目前工具編號。

GetTool 範例



GetVar VI

工具面板

Epson Robots | Variables

描述

傳回控制器中 SPEL+全域保留變數的值。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Var Name	SPEL+全域保留變數的名稱。對於陣列，可以傳回整個陣列或只傳回一個元素。

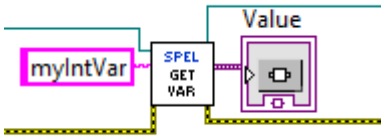
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	包含數值的變數。

另請參閱

SetVar

GetVar 範例



Go VI

工具面板

Epson Robots | Motion

描述

以 PTP 的形式將手臂從目前位置移至指定點或 XY 位置。**GO** 指令可同時移動任何組合的機器人軸。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

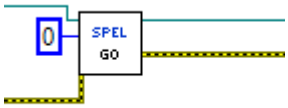
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Jump, Jump3, Move, Speed, TGo, TMove

Go 範例



Halt VI

工具面板

Epson Robots | Tasks

描述

暫停執行指定的任務。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Task Number	選用。要暫停之任務的任務編號。任務編號的範圍介於 1 至 32。如果指定 *Task Name*，則會略過 *Task Number*。
Task Name	選用。指定要暫停之任務的名稱。如果未指定 *Task Name*，將會使用 *Task Number* 輸入。

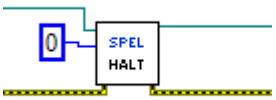
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Quit, Resume

Halt 範例



Here VI

工具面板

Epson Robots | Points

描述

在目前位置示教一個點。

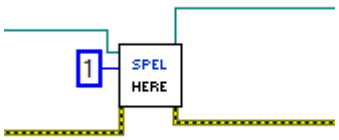
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Name	選用。指定點名稱。如果未指定 *Point Name*，將會使用 *Point Number* 輸入。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

Here 範例



HideWindow VI

工具面板

Epson Robots | GUI

描述

隱藏先前與 ShowWindow 一起顯示的 EPSON RC+ 7.0 視窗。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Window ID	要顯示的 EPSON RC+ 7.0 視窗的 ID。

視窗名稱	ID	描述
IOMonitor	1	I/O 監視器視窗的 ID
TaskManager	2	任務管理器視窗的 ID
ForceMonitor	3	力監視器視窗的 ID
Simulator	4	模擬器視窗的 ID

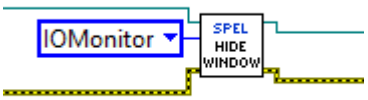
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

RunDialog, ShowWindow

HideWindow 範例



In VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回指定輸入埠的狀態。每個連接埠包含 8 個輸入位元(一位元組)。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含 8 個輸入位元(一位元組)。如果未指定 *Label*，則會使用 *Port Number*。
Label	選用。包含輸入位元組標籤的字串。如果指定 *Label*，則會略過 *Port Number*。

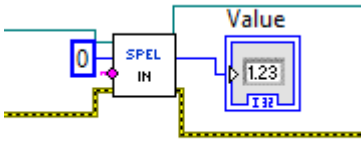
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	介於 0 至 255 的整數運算式，代表輸入埠的狀態。

另請參閱

InBCD, InW, Sw

In 範例



InBCD VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回 8 個輸入的輸入狀態(使用 BCD 格式)。(二進位十進碼)

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含 8 個輸入位元(一位元組)。如果未指定 *Label*，則會使用 *Port Number*。
Label	選用。包含輸入位元組標籤的字串。如果指定 *Label*，則會略過 *Port Number*。

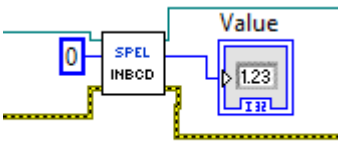
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	介於 0 至 9 的整數運算式，代表輸入埠的狀態。

另請參閱

In, InW, Sw

InBCD 範例



InsideBox VI

工具面板

Epson Robots | Motion

描述

傳回目前機器人夾具末端是否位於指定盒體區域內。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Area Number	介於 1-15 的整數，代表 15 個方塊中要勾選的方塊。

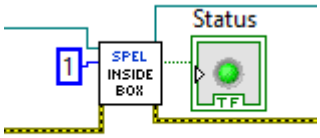
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Status	如果機器人夾具末端位於盒體內，則布林值為 True。

另請參閱

Box, InsidePlane, Plane

InsideBox 範例



InsidePlane VI

工具面板

Epson Robots | Motion

描述

傳回目前機器人夾具末端是否位於指定平面內。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
AreaNumber	介於 1-15 的整數，代表 15 個方塊中要勾選的方塊。

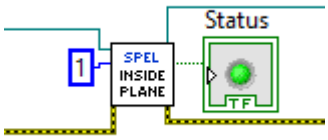
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Status	如果機器人夾具末端位於平面內，則布林值為 True。

另請參閱

Box, InsideBox, Plane

InsidePlane 範例



InW VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回指定輸入字元埠的狀態。每個字元埠包含 16 個輸入位元。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含 8 個輸入位元 (一位元組)。如果未指定 *Label*，則會使用 *Port Number*。
Label	選用。包含輸入位元組標籤的字串。如果指定 *Label*，則會略過 *Port Number*。

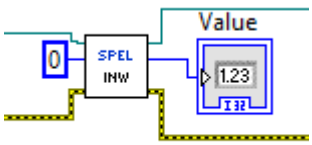
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	介於 0 至 65535 的整數值，代表輸入埠。

另請參閱

In, InBCD, Sw

InW 範例



Inertia VI

工具面板

Epson Robots | Robot Settings

描述

指定目前機器人的裝載慣性和離心率。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
LoadInertia	雙精度值，用以指定夾具末端關節(包含夾具末端和工件)中心周圍的總慣性力矩(kgm2)。
Eccentricity	雙精度值，用以指定夾具末端關節(包含夾具末端和工件)中心周圍的離心率(公釐)。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Weight

Inertia 範例

Initialize VI

工具面板

Epson Robots | System

描述

初始化 LabVIEW VI 程式庫所使用的 Spel 執行個體。

輸入

Server Product Type	選用。指定要連接的 EPSON RC+產品。
Connection Number	選用。指定要使用的控制器連線。
Project	選用。指定要使用的 EPSON RC+專案。
ServerInstance	選用。指定使用 EPSON RC+伺服器的哪個實例。
ConnectionPassword	選用。連線密碼的字串。 如果控制器具有用與控制器連接的密碼，並且密碼未在 EPSON RC+中[配置] - [電腦與控制器連接]對話方塊中設置，則必須設置連接到控制器的密碼

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

備註

針對要使用之程式庫的每個執行個體，必須調用 Initialize VI。

Server Product Type 用於指定要使用的 EPSON RC+產品。預設值為 RC70 (EPSON RC+ 7.0)。

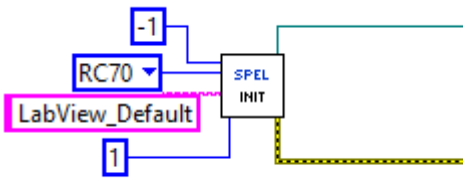
當未指定 *Connection Number* 時，將會使用 EPSON RC+ 7.0 中上次使用的連線。

當未指定 *Project* 時，將會使用預設的 LabVIEW EPSON RC+ 7.0 專案。專案必須在以 *Server Product Type* 指定的 EPSON RC+產品中使用。

另請參閱

Shutdown

Initialize 範例



JRange VI

工具面板

Epson Robots | Robot Settings

描述

定義指定機器人關節的允許工作範圍(以脈衝為單位)。

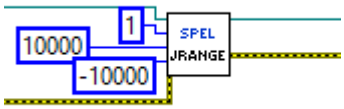
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
JointNumber	介於 1 - 9 的整數，代表要指定之 JRange 的關節。
LowerLimitPulses	代表指定關節下限範圍之編碼器脈衝計數位置的整數。
UpperLimitPulses	代表指定關節上限範圍之編碼器脈衝計數位置的整數。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

JRange 範例



JS VI

工具面板

Epson Robots | Motion

描述

Jump Sense 會偵測手臂是否在完成 JUMP 指令之前停止(使用 SENSE 輸入)，或手臂是否完成 JUMP 移動。JS 會傳回 Jump Sense 狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

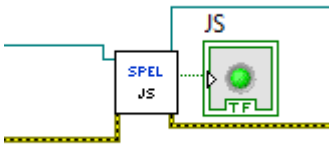
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
JS	若動作期間偵測到 SENSE 輸入，會傳回 True。否則會傳回 False。

另請參閱

Jump, Sense

JS 範例



JTran VI

工具面板

Epson Robots | Motion

描述

執行相對關節移動。

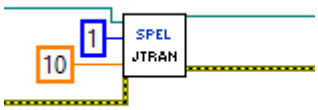
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
JointNumber	要移動的特定關節。
Distance	要移動的距離。旋轉關節的單位為度，線性關節的單位為公釐。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

JTran 範例



Jump VI

工具面板

Epson Robots | Motion

描述

利用 PTP 動作將手臂從目前位置移至指定點(先垂直向上移動，接著水平移動，最後向下垂直移至最終目標點)。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

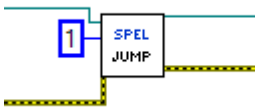
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Go, Jump3, Move, Speed, TGo, TMove

Jump 範例



Jump3 VI

工具面板

Epson Robots | Motion

描述

使用兩個 CP 動作與一個 PTP 動作之組合的 3D 閘道動作。 機器人會移到起始點、接近點和目標點。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Depart Point Number	使用整數指定起始點。
Depart Point Expr	使用字串運算式指定起始點。如果使用此輸入，您必須同時使用字串運算式指定接近和目標點。
Appro Point Number	使用整數指定接近點。
Appro Point Expr	使用字串運算式指定接近點。如果使用此輸入，您必須同時使用字串運算式指定起始和目標點。
Dest Point Number	使用整數指定目標點。
Dest Point Expr	使用字串運算式指定目標點。如果使用此輸入，您必須同時使用字串運算式指定起始和接近點。

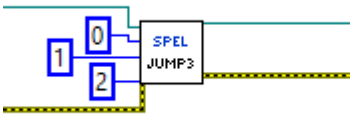
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Go, Jump, Move, Speed, TGo, TMove

Jump3 範例



Jump3CP VI

工具面板

Epson Robots | Motion

描述

使用三個 CP 動作之組合的 3D 閘道動作。機器人會移到起始點、接近點和目標點。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Depart Point Number	使用整數指定起始點。
Depart Point Expr	使用字串運算式指定起始點。如果使用此輸入，您必須同時使用字串運算式指定接近和目標點。
Appro Point Number	使用整數指定接近點。
Appro Point Expr	使用字串運算式指定接近點。如果使用此輸入，您必須同時使用字串運算式指定起始和目標點。
Dest Point Number	使用整數指定目標點。
Dest Point Expr	使用字串運算式指定目標點。如果使用此輸入，您必須同時使用字串運算式指定起始和接近點。

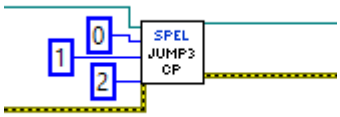
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Go, Jump, Jump3, Move, Speed, TGo, TMove

Jump3CP 範例



LimZ VI

工具面板

Epson Robots | Motion

描述

設定 JUMP 命令之 Z 軸高度的預設值。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Z Limit	Z 軸可移動範圍內的座標值。

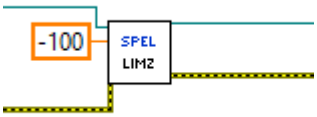
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Jump, Jump3

LimZ 範例



LoadPoints VI

工具面板

Epson Robots | Points

描述

將 SPEL+點文件載入至目前機器人的控制器點記憶體。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
File Name	目前 Spel 專案中或先前使用 SavePoints VI 儲存的有效點檔案。

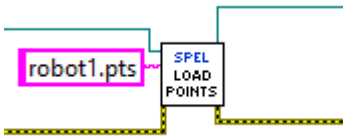
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

GetPoint, Robot, SavePoints, SetPoint

LoadPoints 範例



MemIn VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回指定記憶體 I/O 位元組埠的狀態。每個連接埠包含 8 個記憶體 I/O 位元。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含 8 個輸入位元 (一位元組)。如果未指定 *Label*，則會使用 *Port Number*。
Label	選用。包含輸入位元組標籤的字串。如果指定 *Label*，則會略過 *Port Number*。

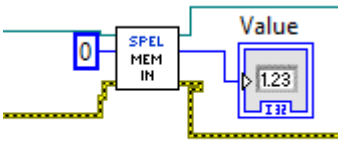
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	介於 0 至 255 的整數，代表埠的狀態。

另請參閱

MemInW, MemOut, MemOutW

MemIn 範例



MemInW VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回指定記憶體 I/O 字元埠的狀態。每個字元埠包含 16 個記憶體 I/O 位元。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個字元埠包含 16 個輸入位元。如果未指定 *Label*，則會使用 *Port Number*。
Label	選用。包含輸入位元組標籤的字串。如果指定 *Label*，則會略過 *Port Number*。

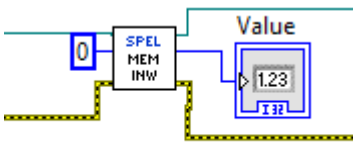
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	介於 0 至 255 的整數，代表埠的狀態。

另請參閱

MemIn, MemOut, MemOutW

MemInW 範例



MemOut VI

工具面板

Epson Robots | Inputs & Outputs

描述

根據使用者指定的 8 位元值，同時設定 8 個記憶體 I/O 位元。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Port Number	選用。代表任一個輸入埠的整數。每個連接埠包含 8 個輸入位元 (一位元組)。如果未指定 *Label*，則會使用 *Port Number*。
Label	選用。包含輸入位元組標籤的字串。如果指定 *Label*，則會略過 *Port Number*。
Value	包含指定位元組之輸出模式的整數。有效值介於 0 - 255。

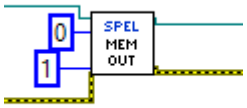
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MemIn, MemInW, MemOutW

MemOut 範例



MemOff VI

工具面板

Epson Robots | Inputs & Outputs

描述

關閉記憶體 I/O 的指定位元。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Bit Number	選用。代表任一個記憶體 I/O 位元的整數。如果未指定 *Label*，則會使用 *Bit Number*。
Label	選用。包含輸入位元標籤的字串。如果指定 *Label*，則會略過 *Bit Number*。

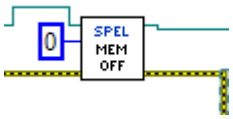
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MemOn, MemOut, MemOutW

MemOff 範例



MemOn VI

工具面板

Epson Robots | Inputs & Outputs

描述

開啟記憶體 I/O 的指定位元。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Bit Number	選用。代表任一個記憶體 I/O 位元的整數。如果未指定 *Label*，則會使用 *Bit Number*。
Label	選用。包含輸入位元標籤的字串。如果指定 *Label*，則會略過 *Bit Number*。

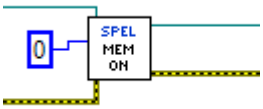
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MemOff, MemOut, MemOutW

MemOn 範例



MemOut VI

工具面板

Epson Robots | Inputs & Outputs

描述

根據使用者指定的 8 位元值，同時設定 8 個記憶體 I/O 位元。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Port Number	選用。代表任一個記憶體 I/O 埠的整數。每個連接埠包含 8 個記憶體 I/O 位元(一位元組)。如果未指定 *Label*，則會使用 *Port Number*。
Label	選用。包含記憶體 I/O 位元組標籤的字串。如果指定 *Label*，則會略過 *Port Number*。
Value	包含指定位元組之輸出模式的整數。有效值介於 0 - 255。

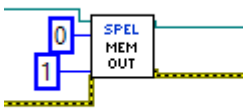
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MemOn, MemOff, MemOutW

MemOut 範例



MemOutW VI

工具面板

Epson Robots | Inputs & Outputs

描述

根據使用者指定的 16 位元值，同時設定 16 個記憶體 I/O 位元。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Port Number	選用。代表任一個記憶體 I/O 埠的整數。每個字元埠包含 16 個輸入位元。如果未指定 *Label*，則會使用 *Port Number*。
Label	選用。包含記憶體 I/O 位元組標籤的字串。如果指定 *Label*，則會略過 *Port Number*。
Value	包含指定字元之輸出模式的整數。有效值介於 0 - 65535。

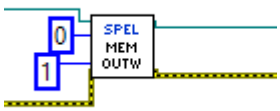
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MemOn, MemOff, MemOut

MemOutW 範例



MemSw VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回指定記憶體 I/O 位元的狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Bit Number	選用。代表任一個記憶體 I/O 位元的整數。如果未指定 *Label*，則會使用 *Bit Number*。
Label	選用。包含記憶體 I/O 位元標籤的字串。如果指定 *Label*，則會略過 *Bit Number*。

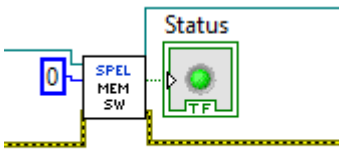
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	記憶體 I/O 位元開啟時為 True 的布林值。

另請參閱

MemIn, MemInW

MemSW 範例



MotorOff VI

工具面板

Epson Robots | Robot Settings

描述

關閉目前機器人的馬達。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MotorOn, PowerHigh, PowerLow, Robot

MotorOff 範例

MotorOn VI

工具面板

Epson Robots | Robot Settings

描述

開啟目前機器人的馬達。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

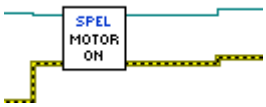
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MotorOff, PowerHigh, PowerLow, Robot

MotorOn 範例



Move VI

工具面板

Epson Robots | Motion

描述

使用線性插補法(即以直線移動)將手臂從目前位置移至指定點。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

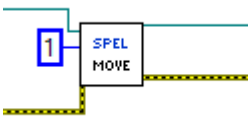
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

AccelS, Arc, Arc3, BGo, BMove, Go, Jump, Jump3, SpeedS, TGo, TMove

Move 範例



Off VI

工具面板

Epson Robots | Inputs & Outputs

描述

關閉指定輸出位元。

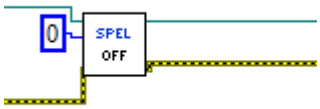
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Bit Number	選用。代表任一個輸出位元的整數。如果未指定 *Label*，則會使用 *Bit Number*。
Label	選用。包含輸入位元標籤的字串。如果指定 *Label*，則會略過 *Bit Number*。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

Off 範例



On VI

工具面板

Epson Robots | Inputs & Outputs

描述

開啟指定輸出位元。

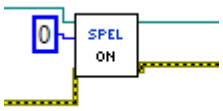
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Bit Number	選用。代表任一個輸出位元的整數。如果未指定 *Label*，則會使用 *Bit Number*。
Label	選用。包含輸入位元標籤的字串。如果指定 *Label*，則會略過 *Bit Number*。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

On 範例



OPort VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回指定輸出位元的狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Bit Number	選用。代表任一個輸出位元的整數。如果未指定 *Label*，則會使用 *Bit Number*。
Label	選用。包含輸出位元標籤的字串。如果指定 *Label*，則會略過 *Bit Number*。

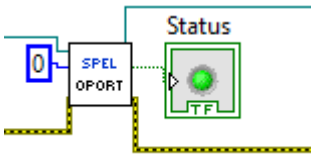
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	輸出位元開啟時為 True 的布林值。

另請參閱

In, InW, On, Off, Out, Sw

Oport 範例



OprMode VI

工具面板

Epson Robots | System

描述

設定 EPSON RC+ 7.0 操作模式。

輸入

Spel Ref In 來自上一個 Spel Ref Out 的 Spel 參考。

Error In 來自上一個 Spel 節點的錯誤條件。

Operation Mode 相關 EPSON RC+ 7.0 伺服器程序的操作模式。

模式	ID	描述
Auto	1	EPSON RC+ 7.0 處於 auto 模式。
Program	2	EPSON RC+ 7.0 處於 program 模式。

輸出

Spel Ref Out 下一個要使用的 VI 的 Spel 參考輸出。

Error Out 後續 Spel 節點的錯誤條件輸出。

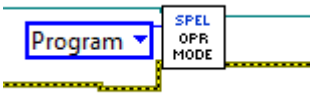
備註

當 *Operation Mode* 設為 Program 時，相關伺服器程序的 EPSON RC+ 7.0 GUI 會開啟，而控制器操作模式會設為 Program。若使用者關閉 RC+ GUI，*Operation Mode* 會設為 Auto。如果 *Operation Mode* 設為 Auto，則 RC+ GUI 會同時關閉。

另請參閱

GetOprMode

OprMode 範例



Pause VI

工具面板

Epson Robots | Tasks

描述

使控制器中的所有正常 SPEL+任務暫停。若機器人正在移動，將會立即減速至停止。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

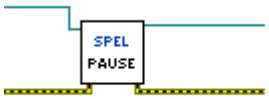
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Continue, Stop

Pause 範例



Plane VI

工具面板

Epson Robots | Robot Settings

描述

定義平面。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Plane Number	介於 1-15 的整數運算式，代表 15 個平面中要定義的平面。
X	平面座標系統原點 X 座標。
Y	平面座標系統原點 Y 座標。
Z	平面座標系統原點 Z 座標。
U	繞 Z 軸的平面座標系統旋轉。
V	繞 Y 軸的平面座標系統旋轉。
W	繞 X 軸的平面座標系統旋轉。

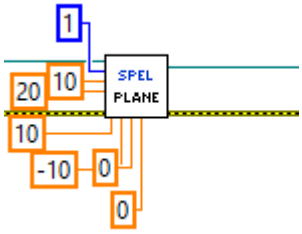
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Box, InsideBox, InsidePlane

Plane 範例



PowerHigh VI

工具面板

Epson Robots | Robot Settings

描述

將目前機器人的馬達運行功率設為高。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

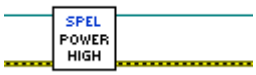
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MotorOff, MotorOn, PowerLow, Robot

PowerHigh 範例



PowerLow VI

工具面板

Epson Robots | Robot Settings

描述

將目前機器人的馬達運行功率設為低。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

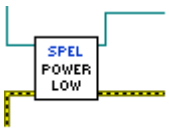
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

MotorOff, MotorOn, PowerHigh, Robot

PowerLow 範例



Quit VI

工具面板

Epson Robots | Tasks

描述

終止執行指定的任務。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Task Number	選用。要終止之任務的任務編號。任務編號的範圍介於 1 至 32。如果指定 *Task Name*，則會略過 *Task Number*。
Task Name	選用。指定要終止之任務的名稱。如果未指定 *Task Name*，將會使用 *Task Number* 輸入。

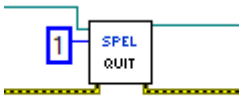
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Halt, Resume

Quit 範例



Reset VI

工具面板

Epson Robots | System

描述

將控制器重置為初始狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

Reset 範例

Resume VI

工具面板

Epson Robots | Tasks

描述

使遭到 Halt VI 暫停的任務重新開始。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Task Number	選用。要恢復之任務的任務編號。任務編號的範圍介於 1 至 32。如果指定 *Task Name*，則會略過 *Task Number*。
Task Name	選用。指定要恢復之任務的名稱。如果未指定 *Task Name*，將會使用 *Task Number* 輸入。

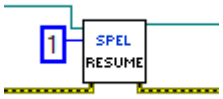
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Halt, Quit

Resume 範例



Robot VI

工具面板

Epson Robots | Robot Settings

描述

選擇目前機器人。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Robot Number	介於 1-16 的整數。

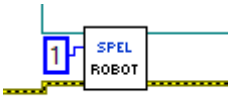
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

GetRobot, MotorOff, MotorOn, PowerHigh, PowerLow

Robot 範例



RunDialog VI

工具面板

Epson Robots | GUI

描述

執行 EPSON RC+ 7.0 對話方塊。

輸入

Spel Ref In 來自上一個 Spel Ref Out 的 Spel 參考。

Error In 來自上一個 Spel 節點的錯誤條件。

Dialog ID 要執行之 EPSON RC+ 7.0 對話方塊的 ID。

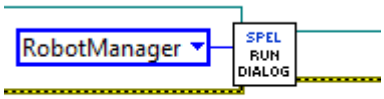
對話方塊名稱	ID	描述
RobotManager	1	工具 | 機器人管理器對話方塊的 ID
ControllerTools	2	工具 | 控制器對話方塊的 ID
VisionGuide	3	工具 | Vision Guide 對話方塊的 ID

輸出

Spel Ref Out 下一個要使用的 VI 的 Spel 參考輸出。

Error Out 後續 Spel 節點的錯誤條件輸出。

RunDialog 範例



SafetyOn VI

工具面板

Epson Robots | System

描述

傳回安全門的狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

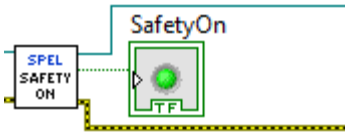
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
SafetyOn	安全門打開時為 True，否則為 False

另請參閱

EStopOn

SafetyOn 範例



SavePoints VI

工具面板

Epson Robots | Points

描述

保存目前機器人的點至檔案。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
File Name	在目前 Spel 專案中的點檔案名稱，或要儲存至控制器的新檔案名稱。

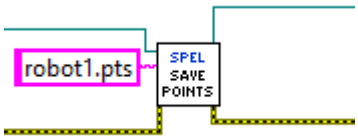
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

GetPoint, LoadPoints, Robot, SetPoint

SavePoints 範例



Sense VI

工具面板

Epson Robots | Motion

描述

指定輸入條件，符合條件時會停止目標位置上方的機器人，以完成進行中的 Jump。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Condition	使用字串運算式來指定 I/O 條件。如需詳細資訊，請參閱 SPEL+語言參考手冊中的 Sense 陳述式。

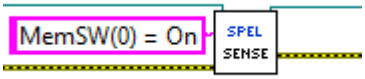
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

JS, Jump

Sense 範例



SetPoint VI

工具面板

Epson Robots | Points

描述

設定目前機器人某個點的座標資料。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Name*，則會略過 *Point Number*。
Point Name	選用。透過使用點名稱的字串運算式指定點。如果未指定 *Point Name*，將會使用 *Point Number* 輸入。
X － W	指定點的 X、Y、Z、U、V、W 座標。

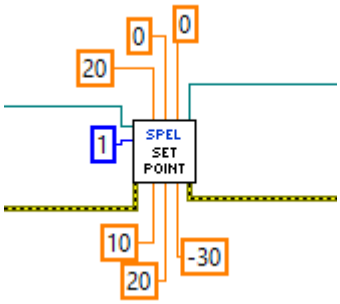
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

GetPoint, LoadPoints, Robot, SavePoints

SetPoint 範例



SetVar VI

工具面板

Epson Robots | Variables

描述

設定控制器中 SPEL+全域保留變數的值。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Var Name	SPEL+全域保留變數的名稱。
Value	包含數值的變數。

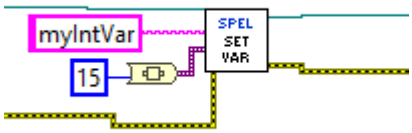
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

GetVar

SetVar 範例



SFree VI

工具面板

Epson Robots | Robot Settings

描述

從伺服系統控制，釋放指定的機器人軸。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Axes	選用。指定要釋放之軸的整數陣列。如果略過，所有軸會釋放。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

備註

如果略過 Axes，則所有軸會釋放。

另請參閱

MotorOff, MotorOn, SLock

SFree 範例

ShowWindow VI

工具面板

Epson Robots | GUI

描述

顯示 EPSON RC+ 7.0 視窗。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Window ID	要顯示的 EPSON RC+ 7.0 視窗的 ID。

視窗名稱	ID	描述
IOMonitor	1	I/O 監視器視窗的 ID
TaskManager	2	任務管理器視窗的 ID
ForceMonitor	3	力監視器視窗的 ID
Simulator	4	模擬器視窗的 ID

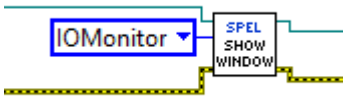
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

HideWindow, RunDialog

ShowWindow 範例



Shutdown VI

工具面板

Epson Robots | System

描述

關閉在 Initialize VI 調用時啟動的 EPSON RC+ 7.0 伺服器程序。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Error Out	後續 Spel 節點的錯誤條件輸出。

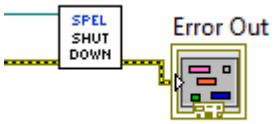
備註

針對程式庫的每個執行個體，必須調用 Shutdown VI。將會關閉相關的 EPSON RC+ 7.0 伺服器程序。

另請參閱

Initialize

Shutdown 範例



SLock VI

工具面板

Epson Robots | Robot Settings

描述

將指定的機器人軸傳回伺服系統控制。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Axes	選用。指定要鎖定之軸的整數陣列。如果略過，所有軸會鎖定。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

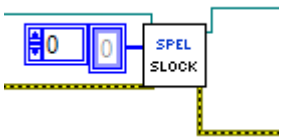
備註

如果略過 Axes，則所有軸會鎖定。

另請參閱

MotorOff, MotorOn, SFree

SLock 範例



Speed VI

工具面板

Epson Robots | Robot Settings

描述

指定要搭配 PTP 指令 Go、Jump 及 Pulse 使用的手臂速度。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
PointToPoint Speed	指定要搭配 PTP 指令 Go、Jump 及 Pulse 使用的手臂速度。
Depart Speed	介於 1-100 的整數，代表 Jump 指令的 Z 軸向上動作速度。
Appro Speed	介於 1-100 的整數，代表 Jump 指令的 Z 軸向下動作速度。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

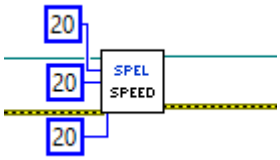
備註

使用 Speed 設定目前機器人的 PTP 速度。所有值可介於 1 至 100%。如果指定 *Depart Speed*，則必須同時指定 *Appro Speed*。

另請參閱

Accel, AccelS, SpeedS

Speed 範例



SpeedS VI

工具面板

Epson Robots | Robot Settings

描述

指定要搭配連續路徑指令 Jump3CP、Move、Arc 及 CVMove 使用的手臂速度。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Linear Speed	指定要搭配連續路徑指令 Jump3CP、Move、Arc 及 CVMove 使用的手臂速度。
Depart Speed	介於 1-5000 的雙精度值，代表 Jump3CP 指令的 Z 軸向上動作速度。
Appro Speed	介於 1-5000 的雙精度值，代表 Jump3CP 指令的 Z 軸向下動作速度。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

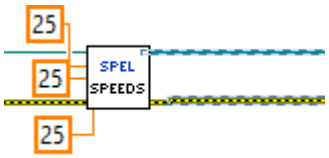
備註

使用 Speed 設定目前機器人的線性速度(mm / sec)。如果指定 *Depart Speed*，則必須同時指定 *Appro Speed*。

另請參閱

Accel, AccelS, Speed

SpeedS 範例



Start VI

工具面板

Epson Robots | Tasks

描述

啟動要在控制器中執行的程式。

輸入

Spel Ref In 來自上一個 Spel Ref Out 的 Spel 參考。

Error In 來自上一個 Spel 節點的錯誤條件。

ProgramNumber 要啟動的程式編號，對應至 SPEL+中的 8 個 main 函數，如下表所示。範圍介於 0 至 7。

程式編號	SPEL+函數名稱
0	main
1	main1
2	main2
3	main3
4	main4
5	main5
6	main6
7	main7

輸出

Spel Ref Out 下一個要使用的 VI 的 Spel 參考輸出。

Error Out 後續 Spel 節點的錯誤條件輸出。

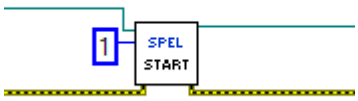
備註

執行 **Start** 時，控制將會立即回到調用 VI。您無法啟動正在執行的程式。請注意，Start 會清除控制器中的全域變數且會載入預設的機器人點。

另請參閱

Continue, Pause, Stop, Xqt

Start 範例



Stop VI

工具面板

Epson Robots | Tasks

描述

停止控制器中執行的所有一般 SPEL+任務，以及選擇性停止所有背景任務。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Stop Type	選用。指定 StopNormalTasks(預設)或 StopAllTasks(也會停止背景任務)。

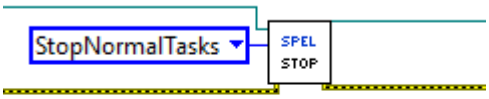
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Continue, Pause, Start, Xqt

Stop 範例



Sw VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回指定輸入位元的狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Bit Number	選用。代表任一個輸入位元的整數。如果未指定 *Label*，則會使用 *Bit Number*。
Label	選用。包含輸入位元標籤的字串。如果指定 *Label*，則會略過 *Bit Number*。

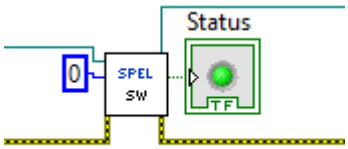
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	輸出位元開啟時為 True 的布林值。

另請參閱

In, InW, On, Off, OPort, Out

SW 範例



TargetOK VI

工具面板

Epson Robots | Motion

描述

傳回指示目前位置到目標位置的 PTP(點至點)動作是否可行的狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

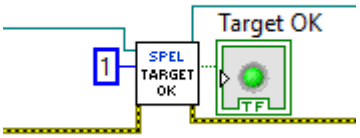
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Target OK	機器人可移至目標位置。
At Home	表示目前機器人是否位於起始點位置的布林值。

另請參閱

BGo, Go, Jump, TGo

TargetOK 範例



TGo VI

工具面板

Epson Robots | Motion

描述

在目前工具座標系統中執行 PTP 相對動作。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

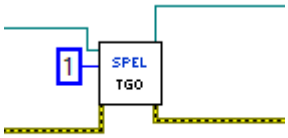
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Go, Jump, Jump3, Move, Speed, TMove

TGo 範例



Till VI

工具面板

Epson Robots | Motion

描述

指定事件條件，符合條件時會減速並停止中間位置的機器人，以完成進行中的動作命令(Jump、Go、Move 等)。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Condition	使用字串運算式來指定 I/O 條件。如需詳細資訊，請參閱 SPEL+語言參考手冊中的 Sense 陳述式。

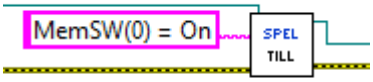
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Jump, Jump3, Move, Speed, TGo, TillOn, TMove

Till 範例



TillOn VI

工具面板

Epson Robots | Motion

描述

若 till 條件在最後 Go/Jump/Move 陳述式期間發生停止狀況，會傳回 True。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

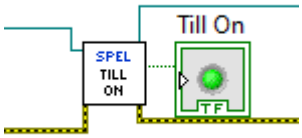
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Till On	若動作期間偵測到 till 條件，會傳回 True。否則會傳回 False。

另請參閱

Accel, Arc, Arc3, BGo, BMove, Jump, Jump3, Move, Speed, TGo, Till, TMove

TillOn 範例



TLSet VI

工具面板

Epson Robots | Robot Settings

描述

定義 ECP(外部控制點)。

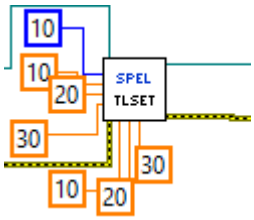
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
ToolNumber	介於 1-15 的整數運算式，代表 15 個工具中要定義的工具。(Tool 0 為預設工具，無法改變。)
X	工具座標系統原點 X 座標。
Y	工具座標系統原點 Y 座標。
Z	工具座標系統原點 Z 座標。
U	繞 Z 軸的工具座標系統旋轉。
V	繞 Y 軸的工具座標系統旋轉。
W	繞 X 軸的工具座標系統旋轉。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

TLSet 範例



TMove VI

工具面板

Epson Robots | Motion

描述

在所選工具座標系統中執行線性插補相對動作。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Point Number	選用。透過使用先前示教點的點編號(在目前機器人的控制器點記憶體中)來指定目標結束點。如果指定 *Point Expression*，則會略過 *Point Number*。
Point Expression	選用。透過使用字串運算式來指定目標結束點。如果未指定 *Point Expression*，將會使用 *Point Number* 輸入。

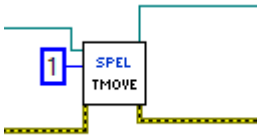
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

AccelS, Arc, Arc3, BGo, BMove, Go, Jump, Jump3, Move, SpeedS, TGo

TMove 範例



Tool VI

工具面板

Epson Robots | Robot Settings

描述

選擇目前機器人工具。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Tool Number	介於 0-15 的整數，代表 16 個工具定義中要搭配後續動作指令使用的工具定義。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Arm, Armset, GetTool, TLSet

Tool 範例

TW VI

工具面板

Epson Robots | Inputs & Outputs

描述

傳回 Wait 命令的條件是否成立命令。

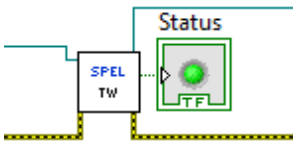
輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Status	傳回 Wait 命令的條件是否成立。

TW 範例



VGetBool VI

工具面板

Epson Robots | Vision

描述

擷取傳回布林值的視覺屬性或結果。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。擷取序列的屬性或結果時省略。
Property Code	屬性或結果程式碼。
Result Index	選用。結果的索引。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	布林值。

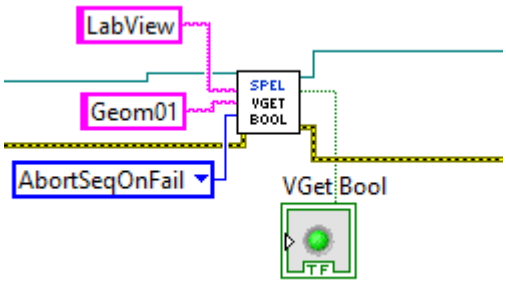
備註

如需 Vision Guide 屬性與結果的詳細資訊，請參閱 Vision Guide 7.0 屬性與結果參考手冊。

另請參閱

VRun, VGetDbl, VGetInt, VGetStr, VSetBool, VSetDbl, VSetInt, VSetStr

VGetBool 範例



VGetDbl VI

工具面板

Epson Robots | Vision

描述

擷取傳回雙精度值的視覺屬性或結果。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。擷取序列的屬性或結果時省略。
Property Code	屬性或結果程式碼。
Result Index	選用。結果的索引。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	雙精度值。

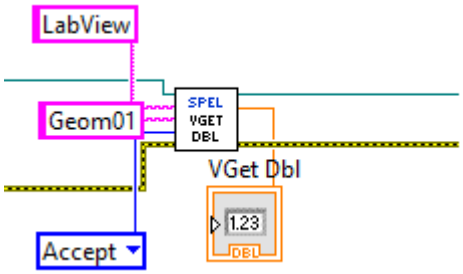
備註

如需 Vision Guide 屬性與結果的詳細資訊，請參閱 Vision Guide 7.0 屬性與結果參考手冊。

另請參閱

VRun, VGetBool, VGetInt, VGetStr, VSetBool, VSetDbl, VSetInt, VSetStr

VGetDbl 範例



VGetInt VI

工具面板

Epson Robots | Vision

描述

擷取傳回的整數值的視覺屬性或結果。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。擷取序列的屬性或結果時省略。
Property Code	屬性或結果程式碼。
Result Index	選用。結果的索引。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	整數值。

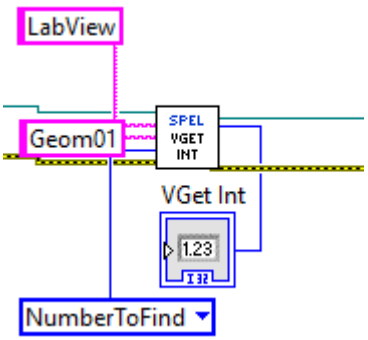
備註

如需 Vision Guide 屬性與結果的詳細資訊，請參閱 Vision Guide 7.0 屬性與結果參考手冊。

另請參閱

VRun, VGetBool, VGetDbl, VGetStr, VSetBool, VSetDbl, VSetInt, VSetStr

VGetInt 範例



VGetStr VI

工具面板

Epson Robots | Vision

描述

擷取傳回字串值的視覺屬性或結果。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。擷取序列的屬性或結果時省略。
Property Code	屬性或結果程式碼。
Result Index	選用。結果的索引。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Value	字串值。

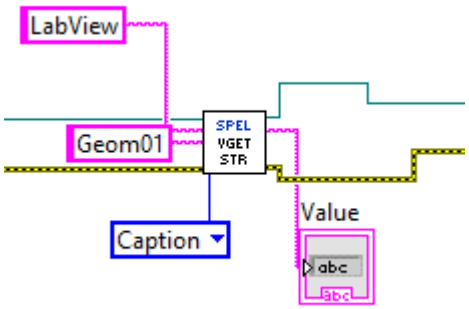
備註

如需 Vision Guide 屬性與結果的詳細資訊，請參閱 Vision Guide 7.0 屬性與結果參考手冊。

另請參閱

VRun, VGetBool, VGetDbl, VGetInt, VSetBool, VSetDbl, VSetInt, VSetStr

VGetStr 範例



VideoControl VI

工具面板

Epson Robots | Vision

描述

為 SPEL Video 控制項進行設定，以顯示視覺系統的影像。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
VideoRef In	來自 SPEL Video 控制項的參考。
Camera	設定要顯示的攝影機視訊。預設值為 0，會顯示任何攝影機。
Graphics Enabled	設定是否要顯示圖形。
Video Enabled	設定是否要顯示視訊。

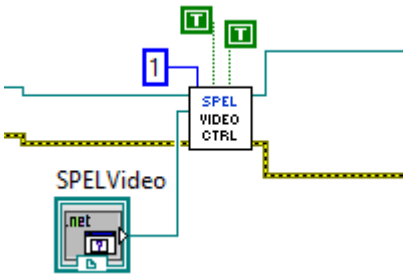
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Displaying Video, VGet, VRun

VideoControl 範例



VRun VI

工具面板

Epson Robots | Vision

描述

在目前專案中執行視覺序列。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

備註

如需執行視覺序列的資訊，請參閱 Vision Guide 7.0 軟體手冊。

另請參閱

VGetBool, VGetDbl, VGetInt, VGetStr, VSetBool, VSetDbl, VSetInt, VSetStr

VRun 範例

VSetBool VI

工具面板

Epson Robots | Vision

描述

設定資料類型為布林值的視覺屬性值。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。設定序列的屬性時省略。
Property Code	屬性程式碼。
Value	屬性的新布林值。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

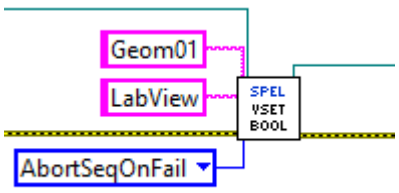
備註

如需 Vision Guide 屬性與結果的詳細資訊，請參閱 Vision Guide 7.0 屬性與結果參考手冊。

另請參閱

VGetBool, VGetDbl, VGetInt, VGetStr, VRun, VSetDbl, VSetInt, VSetStr

VSetBool 範例



VSetDbl VI

工具面板

Epson Robots | Vision

描述

設定資料類型為實數或雙精度值的視覺屬性值。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。設定序列的屬性時省略。
Property Code	屬性程式碼。
Value	屬性的新雙精度值。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

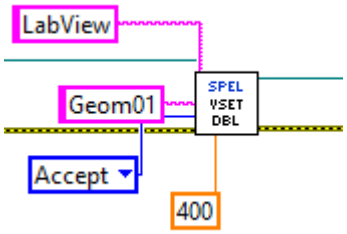
備註

如需 Vision Guide 屬性與結果的詳細資訊，請參閱 Vision Guide 7.0 屬性與結果參考手冊。

另請參閱

VGetBool, VGetDbl, VGetInt, VGetStr, VRun, VSetBool, VSetInt, VSetStr

VSetDbl 範例



VSetInt VI

工具面板

Epson Robots | Vision

描述

設定資料類型為整數的視覺屬性值。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。設定序列的屬性時省略。
Property Code	屬性程式碼。
Value	屬性的新整數值。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

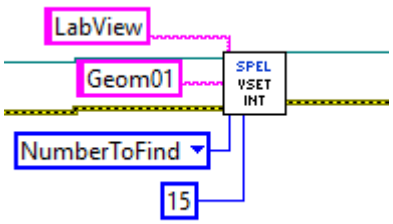
備註

如需 Vision Guide 屬性與結果的詳細資訊，請參閱 Vision Guide 7.0 屬性與結果參考手冊。

另請參閱

VGetBool, VGetDbl, VGetInt, VGetStr, VRun, VSetBool, VSetDbl, VSetStr

VSetInt 範例



VSetStr VI

工具面板

Epson Robots | Vision

描述

設定資料類型為字串的視覺屬性值。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Sequence	包含目前專案中之視覺序列的名稱。
Object	選用。指定序列中視覺物件的名稱。設定序列的屬性時省略。
Property Code	屬性程式碼。
Value	屬性的新字串值。

輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

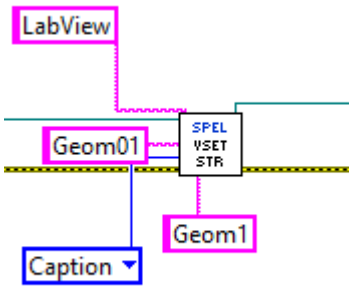
備註

如需 Vision Guide 屬性與結果的詳細資訊，請參閱 Vision Guide 7.0 屬性與結果參考手冊。

另請參閱

VGetBool, VGetDbl, VGetInt, VGetStr, VRun, VSetBool, VSetDbl, VSetInt

VSetStr 範例



WaitTaskDone VI

工具面板

Epson Robots | Tasks

描述

等待任務完成並傳回狀態。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Task Number	選用。要暫停之任務的任務編號。任務編號的範圍介於 1 至 32。如果指定 *Task Name*，則會略過 *Task Number*。
Task Name	選用。指定要暫停之任務的名稱。如果未指定 *Task Name*，將會使用 *Task Number* 輸入。

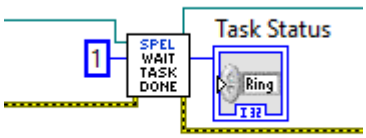
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。
Task State	代表任務的最終狀態(Quit、Aborted、Finished)。

另請參閱

Xqt

WaitTaskDone 範例



Weight VI

工具面板

Epson Robots | Robot Settings

描述

指定目前機器人的重量參數。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Payload Weight	待運送之夾具末端的重量(公斤)。
Arm Length	從第二手臂旋轉中心至夾具末端重心的距離(公釐)。

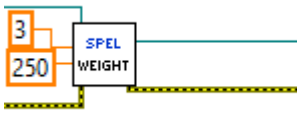
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Inertia

Weight 範例



Xqt VI

工具面板

Epson Robots | Tasks

描述

啟動一個 SPEL+任務。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Task Number	選用。要執行之任務的任務編號。任務編號的範圍介於 1 至 32。如果略過 *Task Number*，將會自動指派任務編號。
Func Name	指定要執行之函數的名稱。

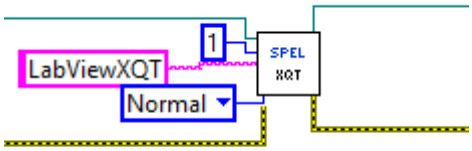
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Halt, Quit, Resume, WaitForTaskDone

XQT 範例



XYLim VI

工具面板

Epson Robots | Robot Settings

描述

設定機器人的允許動作範圍限制。

輸入

Spel Ref In	來自上一個 Spel Ref Out 的 Spel 參考。
Error In	來自上一個 Spel 節點的錯誤條件。
Min X	機器人可移動的最小 X 座標位置。 (機器人無法移至小於 min X 的 X 座標位置。)
Max X	機器人可移動的最大 X 座標位置。 (機器人無法移至大於 max X 的 X 座標位置。)
Min Y	機器人可移動的最小 Y 座標位置。 (機器人無法移至小於 min Y 的 Y 座標位置。)
Max Y	機器人可移動的最大 Y 座標位置。 (機器人無法移至大於 max Y 的 Y 座標位置。)
Min Z	機器人可移動的最小 Z 座標位置。 (機器人無法移至小於 min Z 的 Z 座標位置。)
Max Z	機器人可移動的最大 Z 座標位置。 (機器人無法移至大於 max Z 的 Z 座標位置。)

備註

XYLim 係用來定義動作範圍限制。許多機器人系統都可讓使用者定義關節限制，而 SPEL+語言則可允許同時定義關節限制與動作範圍限制。此可讓使用者有效地為其應用建立工作空間。(請記住，關節範圍限制也可以使用 SPEL 定義。)

使用 XYLim 值建立的動作範圍只會套用至動作命令目標位置，不會套用到起始位置至目標位置的動作路徑。因此，手臂可能會在動作期間超出 XYLim 範圍以外。(亦即 XYLim 範圍不會影響脈衝。)

若要關閉動作範圍限制，請將範圍限制參數指定為 0。

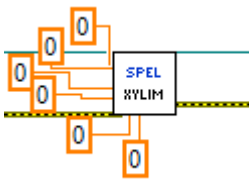
輸出

Spel Ref Out	下一個要使用的 VI 的 Spel 參考輸出。
Error Out	後續 Spel 節點的錯誤條件輸出。

另請參閱

Box

XYLim 範例



17. 以RCNetLib使用LabVIEW

17.1 概述

在「使用 LabVIEW VI 程式庫」章節所述的 LabVIEW VI 程式庫，是使用了 RCAPINet.dll 的高階層介面。某些使用者可能想直接連接 RCAPINet.dll，而不使用高階層程式庫。本章包含 LabVIEW 與 RCAPINet.dll 的搭配使用資訊。將針對下列主題進行說明。

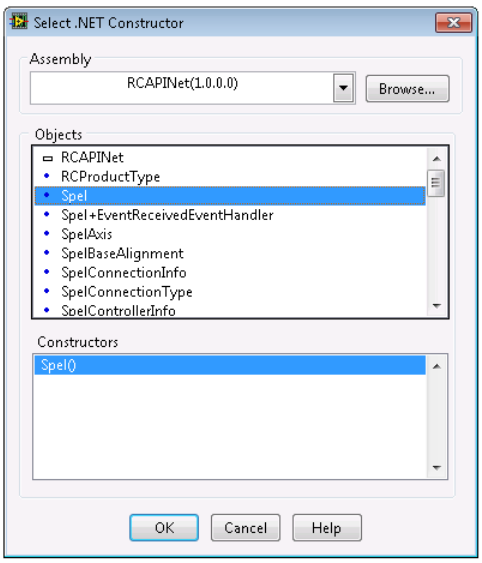
- 初始化
- 在應用程式中使用 Spel 屬性及方法
- 關閉
- 使用對話方塊與視窗

17.2 初始化

17.2.1 添加Spel類別的建構函式節點

從 Spel 類別調用方法或使用屬性之前，您必須使用建構函式節點建立 Spel 類別的執行個體。您應在應用程式中使用一個 Spel 類別執行個體。

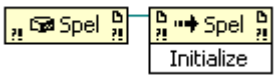
在將會包含 Spel 類別執行個體之 VI 的方塊圖中，從[RC+ API] – [.NET pallete]中添加建構函式節點。此時會顯示[Select .NET Constructor]對話方塊。在[Assembly]列表中選擇「RCAPINet」，再從[Objects]列表中選擇「Spel」，如下所示。



按一下<OK>，在方塊圖中建立 Spel 的建構函式節點。

17.2.2 初始化Spel類別執行個體

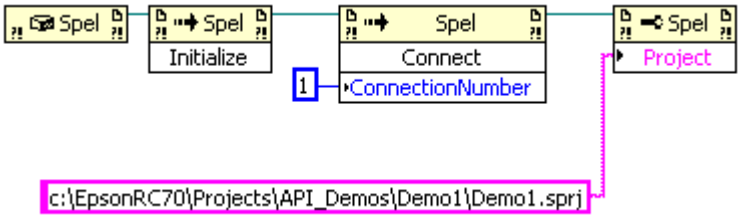
添加 Spel 類別 Initialize 方法的 Invoke 節點。執行 Initialize 時，將會在背景中設定並啟動 RC+作為伺服器。



17.2.3 連接至控制器並設定專案

添加 Spel 類別 Connect 方法的 Invoke 節點。針對您要使用的控制器連線，設定 ConnectionNumber 參數。若要查看連線數量，請啟動 EPSON RC+ 7.0，然後選擇[設定]-[電腦與控制器通信]。

添加 Spel 類別 Project 屬性的 Property 節點。將 Project 參數設定到所需的專案檔。



17.3 使用Spel屬性與方法

對您的應用程式添加更多節點，以使用 Spel 屬性與方法。您必須將先前節點的參考輸出連接到目前節點的參考輸入。此可讓每個屬性或方法使用您在先前步驟所建立及初始化的 Spel 類別執行個體。如需可使用之屬性與方法的詳細資訊，請參閱 RCAPINet 參考章節。

17.4 關閉

使用 Spel 類別執行個體後，您必須叫用 Dispose 方法。此將會關閉與 Spel 類別執行個體相關的 EPSON RC+ 7.0 伺服器。一般而言，您應該在應用程式結束時調用 Dispose。

若應用程式在沒有調用 Dispose 的情況下終止，則因 LabVIEW(用戶端程序)會繼續執行，故 RC+程序也會繼續執行。若重新啟動應用程式，則正在執行的 RC+程序會重啟。如您嘗試執行 RC+ GUI，系統將會詢問您是否要執行其他 RC+執行個體。在此情況下，您可從 Windows 任務管理器中先終止 RC+ 程序(erc70.exe)，然後執行 EPSON RC+ 7.0 GUI。

17.5 使用對話方塊與視窗

.NET 父表單搭配.NET 應用程式使用時，通常會作為從 Spel 類別執行個體顯示之對話方塊與視窗的父項。但 LabVIEW 並不會使用.NET 表單，因此，若要從 LabVIEW 顯示視窗與對話方塊，請使用 ParentWindowHandle 屬性。請將其設為 VI 的視窗控制代碼。您可調用 Windows API FindWindow 方法來取得視窗控制代碼。

使用 ParentWindowHandle 時，您必須直接調用 Spel.ShowWindow(無父參數)。

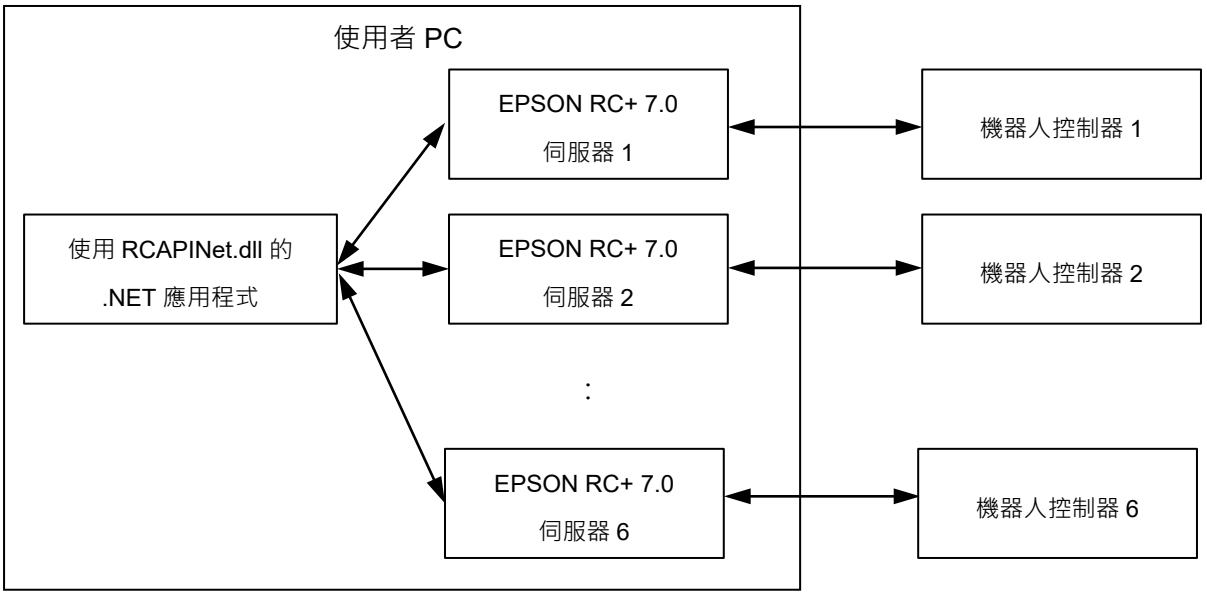
18. 如何從一台PC控制多個控制器

18.1 概述

使用 RC+ API 時，您可從一台 PC 控制最多六個機器人控制器。

若要控制多個控制器，必須針對每個控制器具現化 RCAPINet Spel 類別。

下圖顯示使用 RC+ API 控制多個控制器的基本系統設定圖。



應用程式透過為每個控制器所準備的伺服器(RCAPINet Spel 類別)控制多個控制器。

18.1.1 系統需求

建議 PC 必須符合下列需求。

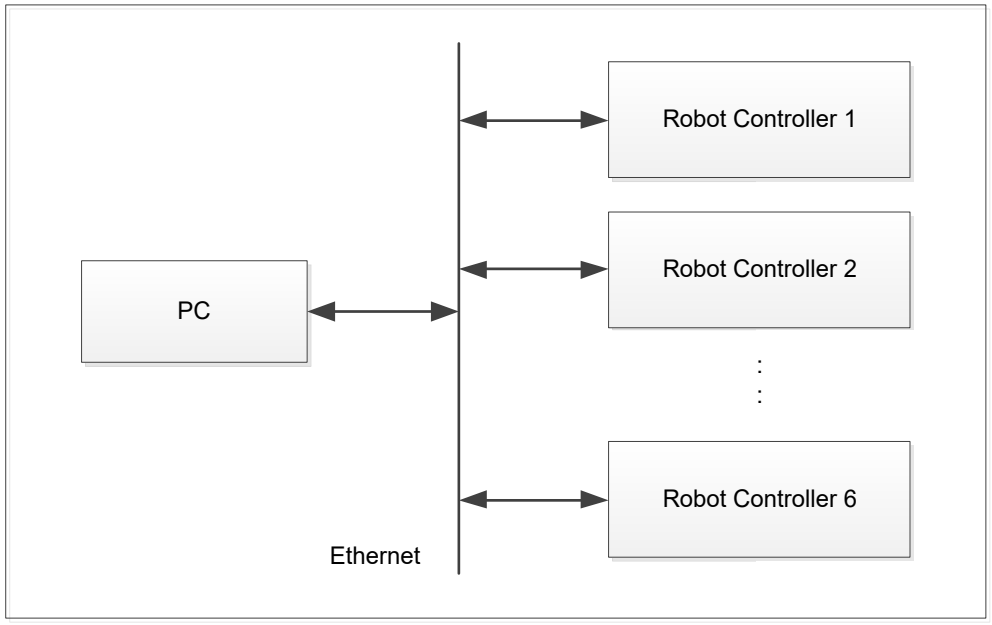
作業系統	Windows 8.1 Pro 64 位元版本 Windows 10 Pro 64 位元版本 Windows 11 Pro 64 位元版本
CPU	Core i5 或以上
記憶體	2 GB 或以上

注 意	■ 使用低於推薦性能的PC時，請客戶充分確認動作后使用。 如果使用的 PC 性能低於上述性能，則可能無法穩定地控制。

18.1.2 電腦與控制器連接

第一個控制器的連線類型可以為 USB 或乙太網。剩餘控制器的連線類型必須為乙太網。

下圖顯示 PC 和多個控制器的基本設定圖。



機器人控制器支援連接到 EPSON RC+ 7.0 的控制器。

NOTE 您可同時連接連接到 EPSON RC+ 7.0 的控制器。

NOTE 您可選擇一個虛擬控制器。

NOTE 可以透過 USB 通信連接一個控制器。
使用 USB 通信時，僅透過 USB 通信連接一個控制器，並透過乙太網連接其他控制器。

注 意	■ 如果PC有安裝防毒軟體，則執行防毒掃描時，與控制器的通信可能異常中斷。若要執行防毒掃描，請先中斷與控制器的通信。

18.2 控制多個控制器的限制

多個控制器的控制限制詳見下列章節的說明。

18.2.1 控制器選購件的限制

由每個控制器所控制的下列控制器選購件具有相關限制。

- PC 視覺
- 現場匯流排主控端
- 力覺
  (力覺及力覺感測器並不相同。)

當上述三個選購件的其中一個已連接到作用中的控制器時，這些選購件無法用於其他(第二個或後者)控制器。

18.2.2 模擬器的限制

模擬器視窗顯示

EPSON RC+ 7.0 模擬器視窗可從.NET 應用程式使用。
如需詳細資訊，請參閱本手冊的 *10.1 視窗*。

當連接多個控制器時，若針對每個控制器開啟模擬器視窗，則與不顯示模擬器視窗相比，循環時間可能增加 100 至 200 msec。

此外，如果在模擬器視窗開啟的情況下執行程式，CPU 使用率會提高將近 100 %，可能對 PC 造成巨大負載。

除非是進行程式偵錯，否則建議在模擬器視窗關閉的情況下使用系統。

碰撞偵測

若要使用模擬器避免與周邊設備發生碰撞，請在模擬器物件周圍設定 15 公釐以上的邊緣，以避免碰撞偵測。

模擬器的碰撞偵測無法保證準確度。應用於實際系統時，務必設定邊緣並確保正常運作。

如需各種限制的詳細資訊，請參閱 *EPSON RC+使用指南* 中的 *8.4 模擬器規格與限制*。

18.3 連接多個控制器的範例程式

下節說明使用.NET 應用程式將 PC 連接到控制器 1 和控制器 2 的範例程式。

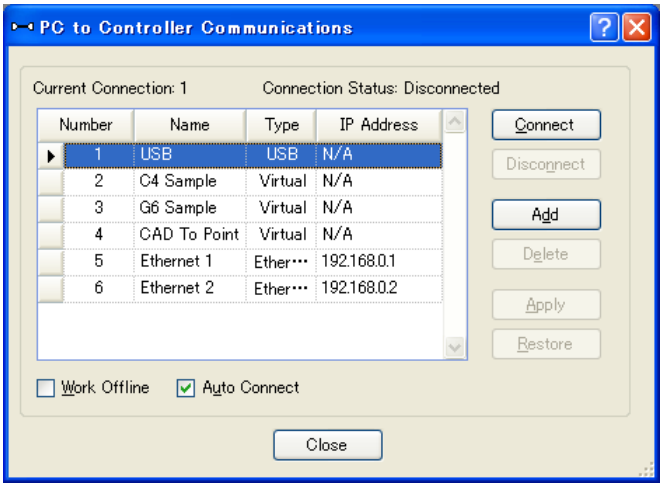
NOTE 如需可用屬性與方法的詳細資訊，請參閱本手冊的 *14. RCAPINet 參考*。

18.3.1 控制連線設定

同時連接至多個控制器時，使用 Spel 類別的 Connect 方法指定連線。

```
m_spel.Connect(1)
```

Connect 方法中的參數代表連線編號。此數字與下方對話方塊中「Number」中所示的數字相同(EPSON RC+ 7.0 功能表-[Setup]-[PC to Controller Communications])。如果數值使用-1，表示使用最近的連線。



NOTE 從 RC+7.5.0 開始，可以使用 Connect 方法的連接名稱(上圖中的名詞)進行連接。

18.3.2 專案設定

若要連接多個控制器，請使用 Spel 類別的 Project 屬性指定專案。每個控制器都必須使用獨立的專案。

```
m_spel.Project = “c:\EpsonRC70\projects\API_Demos\Demo1\Demo1.sprj”
```

18.3.3 使用Visual Basic的範例程式

(1) 在 Visual Studio .NET 中選擇功能表-[File]-[New]-[Project]。

(2) 建立 Visual Basic 專案。

(3) 選擇[Project]-[Add Reference]。

(4) 選擇[Browse]標籤、參照「\EpsonRC70\Exe」目錄，然後選擇「RCAPINet.dll」檔案。

(5) 添加兩個按鈕(btnController1、btnController2)至 Form1 類別。

(6) 添加各按鈕的快速事件，並建立執行緒以控制每個機器人控制器。

```
Private trd1 As System.Threading.Thread     '機器人控制器1
Private trd2 As System.Threading.Thread     '機器人控制器2

Private Sub btnController1_Click(ByVal sender As
System.Object, _
            ByVal e As System.EventArgs) Handles
            btnController1.Click
        '啟動機器人控制器1的執行緒
        trd1 = New System.Threading.Thread( _
          New System.Threading.ThreadStart(AddressOf
          StartController1))
        trd1.Start()
End Sub
Private Sub StartController1()
    '控制機器人控制器1
    Try
       Dim frm1 As New frmDemo1
       frm1.ShowDialog()
       frm1.Dispose()
    Catch ex As Exception
       MsgBox(ex.Message)
    End Try
End Sub
Private Sub btnController2_Click(ByVal sender As
System.Object, _
ByVal e As System.EventArgs) Handles btnController2.Click
     '啟動機器人控制器2的執行緒
     trd2 = New System.Threading.Thread( _
                New System.Threading.ThreadStart(AddressOf
                                        StartController2))
     trd2.Start()
End Sub
Private Sub StartController2()
    '控制機器人控制器2
    Try
       Dim frm2 As New frmDemo2
       frm2.ShowDialog()
       frm2.Dispose()
    Catch ex As Exception
       MsgBox(ex.Message)
    End Try
End Sub
```

(7) 添加控制器 1 的表單(frmDemo1)。

```
Private WithEvents m_spel1 As New Spel
Private Sub frmDemo1_Load(ByVal sender As System.Object,
_
```

```
                ByVal e As System.EventArgs) Handles MyBase.Load

        Try
            m_spel1.ServerInstance=1
            m_spel1.Initialize()
            m_spel1.Connect(5)
            m_spel1.Project = "
c:\\EpsonRC70\\Projects\\Demo1\\Demo1.sprj "
        Catch ex As SpelException
            MsgBox(ex.Message)
        End Try
End Sub
Private Sub m_spel1_EventReceived(ByVal sender As Object,
ByVal e As
                SpelEventArgs) _Handles m_spel1.EventReceived
    ' 機器人控制器1
End Sub
Private Sub frmDemo1_FormClosed(ByVal sender As
System.Object, _
              ByVal e As
System.Windows.Forms.FormClosedEventArgs) _
              Handles MyBase.FormClosed
    m_spel1.Dispose()
End Sub
```

(8) 添加控制器 2 的表單(frmDemo2)。

```
Private WithEvents m_spel2 As New Spel
Private Sub frmDemo2_Load(ByVal sender As System.Object,
_
            ByVal e As System.EventArgs) Handles MyBase.Load

    Try
        m_spel2.ServerInstance=2
        m_spel2.Initialize()
        m_spel2.Connect(6)
        m_spel2.Project = "
c:\\EpsonRC70\\Projects\\Demo2\\Demo2.sprj "
    Catch ex As SpelException
        MsgBox(ex.Message)
    End Try
End Sub

Private Sub m_spel2_EventReceived(ByVal sender As Object,
ByVal e As
                SpelEventArgs) _Handles m_spel2.EventReceived
    ' 機器人控制器2
End Sub
Private Sub frmDemo2_FormClosed(ByVal sender As
System.Object, _
                ByVal e As
System.Windows.Forms.FormClosedEventArgs) _
                Handles MyBase.FormClosed
    m_spel2.Dispose()
End Sub
```

18.3.4 使用Visual C#的範例程式

(1) 在 Visual Studio .NET 中選擇功能表-[File]-[New]-[Project]。

(2) 建立 Visual C#專案。

(3) 選擇功能表-[Project]-[Add Reference]。

(4) 選擇[Browse]標籤、參照「\EpsonRC70\Exe」目錄，然後選擇「RCAPINet.dll」檔案。

(5) 添加兩個按鈕(btnController1、btnController2)至 Form1 類別。

(6) 添加各按鈕的快速事件，並建立執行緒以控制每個機器人控制器。

```
private System.Threading.Thread trd1;  // 機器人控制器1
private System.Threading.Thread trd2;  // 機器人控制器2

private void btnController1_Click(object sender,
EventArgs e)
{
    // 啟動機器人控制器1的執行緒
    trd1 = new System.Threading.Thread(new _

System.Threading.ThreadStart(StartController1));
    trd1.Start();
}
private void StartController1()
{
    // 控制機器人控制器1
    try
    {
       frmDemo1 frm1 = new frmDemo1();
       frm1.ShowDialog();
       frm1.Dispose();
    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
private void btnController2_Click(object sender,
EventArgs e)
{
    // 啟動機器人控制器2的執行緒
    trd2 = new System.Threading.Thread(new _
          System.Threading.ThreadStart(StartController2));
    trd2.Start();
}
private void StartController2()
{
    // 控制機器人控制器2
    try
    {
       frmDemo2 frm2 = new frmDemo2();
       frm2.ShowDialog();
       frm2.Dispose();
    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
```

```
    }
}
```

(7) 添加控制器 1 的表單(frmDemo1)。

```
private Spel m_spel1;
private void frmDemo1_Load(object sender, EventArgs e)
{
    m_spel1 = new Spel();
    try
    {
       m_spel1.ServerInstance = 1;
       m_spel1.Initialize();
       m_spel1.Connect(5);
       m_spel1.Project =
"c:\\EpsonRC70\\Projects\\Demo1\\Demo1.sprj";
       m_spel1.EventReceived += new _
      Spel.EventReceivedEventHandler(m_spel1_EventReceive
                                                        d);
    }
    catch (SpelException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
public void m_spel1_EventReceived(object sender,
SpelEventArgs e)
{
    // 機器人控制器1
}
private void frmDemo1_FormClosed(object sender,
FormClosedEventArgs e)
{
    m_spel1.Dispose();
}
```

(8) 添加控制器 2 的表單(frmDemo2)。

```
private Spel m_spel2;
private void frmDemo2_Load(object sender, EventArgs e)
{
    m_spel2 = new Spel();
    try
    {
       m_spel2.ServerInstance = 2;
       m_spel2.Initialize();
       m_spel2.Connect(6);
       m_spel2.Project =
"c:\\EpsonRC70\\Projects\\Demo2\\Demo2.sprj";
       m_spel2.EventReceived += new _
   Spel.EventReceivedEventHandler(m_spel2_EventReceived);
    }
    catch (SpelException ex)
    {
        MessageBox.Show(ex.Message);
    }
}
public void m_spel2_EventReceived(object sender,
SpelEventArgs e)
{
    // 機器人控制器2
}
```

```
private void frmDemo2_FormClosed(object sender,
FormClosedEventArgs e)
{
    m_spel2.Dispose();
}
```