

EPSON

**Epson RC+ 8.0 擴張功能
Library Builder 8.0**

翻譯版

© Seiko Epson Corporation 2025

Rev.2
TCM256S7322R

目錄

1. 前言	4
1.1 前言	5
1.2 商標	5
1.3 本手冊中的商標註釋	5
1.4 注意	5
1.5 製造商	5
1.6 聯絡資訊	5
1.7 閱讀本手冊之前	6
2. 概述	7
2.1 概述	8
3. 開發函式庫	11
3.1 函式庫的元件	12
3.2 建立函式庫用專案	12
3.3 建立函式庫	14
3.4 報告函式庫內所定義使用者錯誤	17
3.5 使用回呼函數	18
3.6 使用libcfg檔案的同步	18
4. 建立發佈用函式庫	20
4.1 建立發佈用函式庫	21
5. 使用函式庫	25
5.1 匯入函式庫	26
5.2 啟用函式庫	26
5.3 將函式庫登錄至專案	29
5.4 將函式庫從專案取消登錄	30
5.5 使用函式庫工具	31
5.6 顯示函式庫的手冊	31
5.7 設定函式庫的屬性	31
5.8 函式庫使用者錯誤定義	32
5.9 刪除函式庫	33

6. SPEL+命令參考	34
6.1 SPEL+命令清單	35
6.2 ArchReserve函數	36
6.3 ArmLib函數	37
6.4 ArmReserve函數	38
6.5 LibGetInfo	39
6.6 LocalReserve函數	40
6.7 PointReserve函數	41
6.8 SignalReserve函數	42
6.9 SyncLockReserve函數	43
6.10 TaskReserve函數	44
6.11 TimerReserve函數	45
6.12 TLReserve函數	46
6.13 ToolLib函數	47
6.14 UploadFileAfterStop函數	48

1. 前言

1.1 前言

感謝您購買我們的機器人產品。

本手冊包含正確使用Library Builder的必要資訊。

在安裝機器人系統之前，請詳閱本手冊及其他相關手冊。

請妥善保管本手冊以供隨時取用。

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。

1.2 商標

Microsoft、Windows、Windows、Visual Basic、Visual C++標誌皆為Microsoft Corporation在美國及其他國家的註冊商標或商標。

其他品牌及產品名稱皆為個別擁有者所有之商標或註冊商標。

1.3 本手冊中的商標註釋

Microsoft® Windows® 10作業系統

Microsoft® Windows® 11作業系統

於本手冊中，Windows 10、Windows 11分別表示上述作業系統。某些情況下，Windows意指Windows 10與Windows 11。

1.4 注意

如未獲授權，不得複製或重製本手冊中的任何部分。

本手冊內容如有變更，恕不另行通知。

若您在本手冊中發現任何錯誤或對相關內容有任何意見，請告知我們。

1.5 製造商

SEIKO EPSON CORPORATION

1.6 聯絡資訊

在下列手冊的「供應商」中，記載了詳細的聯絡資訊。

各地區的聯絡資訊有所不同，敬請注意。

「安全手冊 - 聯絡資訊」

亦可從以下網站瀏覽安全手冊。
URL : <https://download.epson.biz/robots/>



1.7 閱讀本手冊之前

本節說明在閱讀本手冊前您應先瞭解的資訊。

關於Epson RC+ 8.0安裝資料夾

Epson RC+ 8.0允許您將安裝資料夾路徑更改為任何位置。本手冊假定Epson RC+ 8.0安裝在C:\EpsonRC80中進行說明。

符號意義

在手冊中的安全注意事項皆以下列符號表示。請確實閱讀各符號的說明。



警告

此符號代表若不正確遵守相關指示，可能會有重傷或死亡的危險。



警告

此符號代表若不正確遵守相關指示，可能會有因觸電導致重傷或死亡的危險。



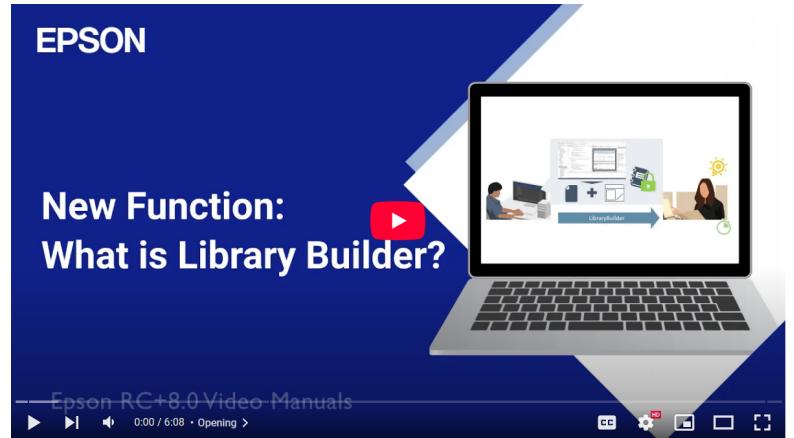
注意

此符號代表若不正確遵守相關指示，可能會有人員受傷或設備及設施受損的危險。

2. 概述

2.1 概述

影片中也有Library Builder說明。

Title	Link
1. New Function: What is Library Builder?	
2-1. Basic Operation of Library Builder Creator Edition	
2-2. Basic Operation of Library Builder User Edition	

Title	Link
3. Let's try it! Library Builder	 <p>EPSON Let's try it! Library Builder Epson RC+8.0 Video Manuals</p>

提示

- 觀看視頻時需網際網路連線。
- 播放語音。
- 顯示母語請使用YouTube的自動翻譯字幕功能。

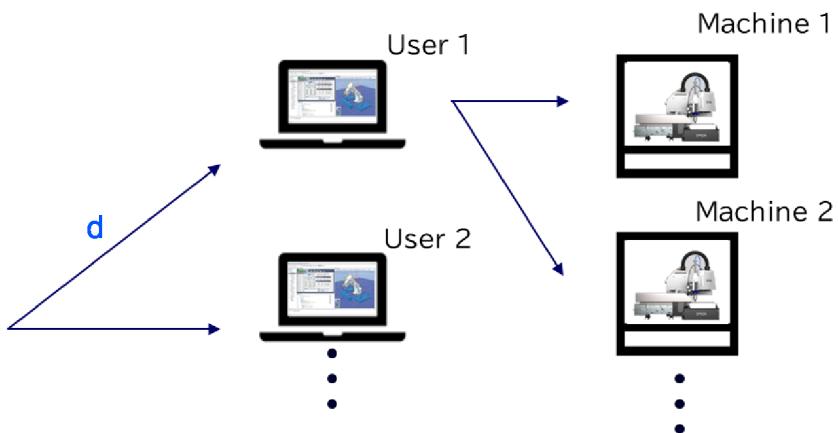
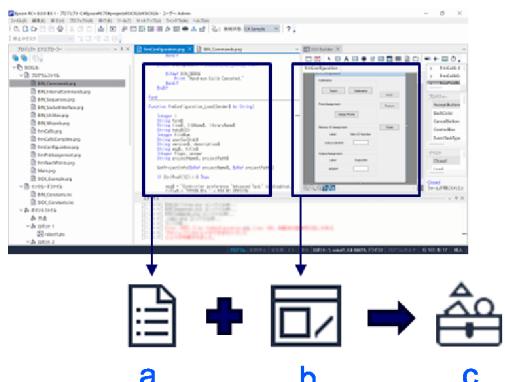
Library Builder是以下功能的總稱。

- 將SPEL+專案的設定資料和設定檔封裝成為函式庫的功能
- 可將建立的函式庫添加至其他SPEL+專案後使用的功能

封裝成函式庫後的SPEL+專案元件中，以下元件可從添加函式庫後的SPEL+專案使用。

- 全域變數
- 保留變數(Global Preserve)
- SPEL+函數
- 透過GUI Builder建立的表單
- 使用者錯誤
- 程式檔案
- 含入檔案
- 視覺順序

透過使用Library Builder，建立自己的函式庫用於輔助進行製作裝置和第三方設備的設定，並提供給使用者。



符號	說明
a	·程式 ·函數 ·變數
b	·設定畫面 ·顯示畫面
c	函式庫
d	提供

提示

建立、使用函式庫需要Epson RC+ 8.0 Ver8.1.0.0或更新版本。此外，建立函式庫需要Epson RC+ Premium Edition授權。

關於建立函式庫的實用SPEL+命令，在以下控制器韌體版本或更新版本中可使用。

- RC800系列：8.1.0.0
- RC90、RC700系列：7.5.5.0
- T、VT系列：7.5.55.0

如需SPEL+命令的詳細資訊，請參閱以下內容。

- [SPEL+命令參考](#)
- 「[SPEL+語言參考 - Appendix C: Epson RC+ 8.0的命令 - C-2: Epson RC+ 8.0各版本添加的命令清單](#)」

3. 開發函式庫

3.1 函式庫的元件

- 封裝成函式庫後的SPEL+專案元件分為公用元件和私人元件。
- 公用元件可從添加函式庫後的SPEL+專案直接使用。
- 私人元件無法從添加函式庫後的SPEL+專案直接使用。
- 公用元件和私人元件的區分依據是否有前綴。
- 公用程式檔案和含入檔案稱為公開檔案，可在添加函式庫後的SPEL+專案進行編輯。

功能	公用元件	私人元件
從使用函式庫的SPEL+專案使用	可	不可
函式庫的使用者進行編輯	可	不可

- 公用元件的用途如下。

元件	指定為公用時的用途
函數	使用目的為向函式庫使用者提供作為函數的功能。
全域變數	用於透過變數之值顯示函式庫的動作狀態、指定動作所需之值。
保留變數	在與全域變數相同用途中使用。若要將值保留在控制器內並在下次啟動時繼續使用，設為保留變數。
程式檔案	<ul style="list-style-type: none"> • 呈現函式庫的共用函數使用範例時使用。 • 希望函式庫使用者描述從函式庫內呼叫的共用函數（回呼函數）之處理時使用。
含入檔案	關於共用函數的引數和傳回值，使用時函式庫使用者可作為常數處理。透過設為常數，讓函式庫使用者更容易理解值的含義。
視覺順序	使用目的為向函式庫使用者提供視覺順序。

3.2 建立函式庫用專案

1. 建立函式庫需要Epson RC+ Premium Edition授權。請從供應商購買授權金鑰並啟用授權。
如需如何啟用的相關資訊，請參閱以下手冊。

「Epson RC+ 8.0 使用指南 - 系統操作 - Epson RC+ 8.0 的啟動 - 授權認證」
2. 新建函式庫用專案。
3. 為對函式庫的使用者公開的資訊添加前綴（前置詞）。前綴請指定包括結尾之底線在內的最多10個字元。
可公開的資訊如下。

資訊	公開條件
函數	函數應以前綴開始
全域變數	變數名應以前綴開始

資訊	公開條件
保留變數	變數名應以前綴開始
程式檔案	應滿足以下所有條件 ·檔案名應以前綴開始 ·檔案內記載的函數、常數、全域變數、保留變數均以前綴開始 ·檔案內記載的僅為函數、全域變數、保留變數、註解 ·不含未滿足公開條件的含入檔案
含入檔案	應滿足以下所有條件 ·檔案名應以前綴開始 ·檔案內記載的常數均以前綴開始 ·檔案內記載的僅為常數和註解·不含未滿足公開條件的含入檔案
視覺順序	視覺名應以前綴開始

例：前綴為“MyLib_”時

```
Global Integer MyLib_Counter           'public global variable
Global Preserve Integer MyLib_WorkPieces 'public global preserve variable

Function MyLib_SomeFunction            'public function
  'some kind of processing
Endf
```

MyLib_Callbacks.prg : 公開程式檔案

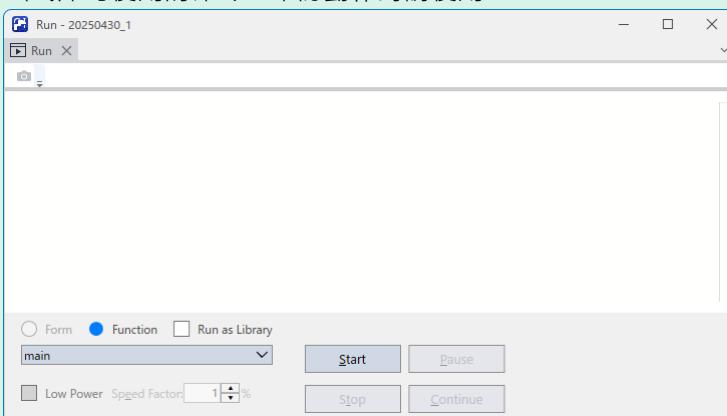
提示

使用多個函式庫時，請添加函式庫特有的前綴，以免函數名、變數名等重複。

1. 請開發包含上述公開資訊在內的專案。

提示

在Epson RC+ Premium Edition中，執行視窗上顯示[作為函式庫執行]核取方塊。啟用後，可從專案執行僅在函式庫可使用的命令。確認動作時請使用。

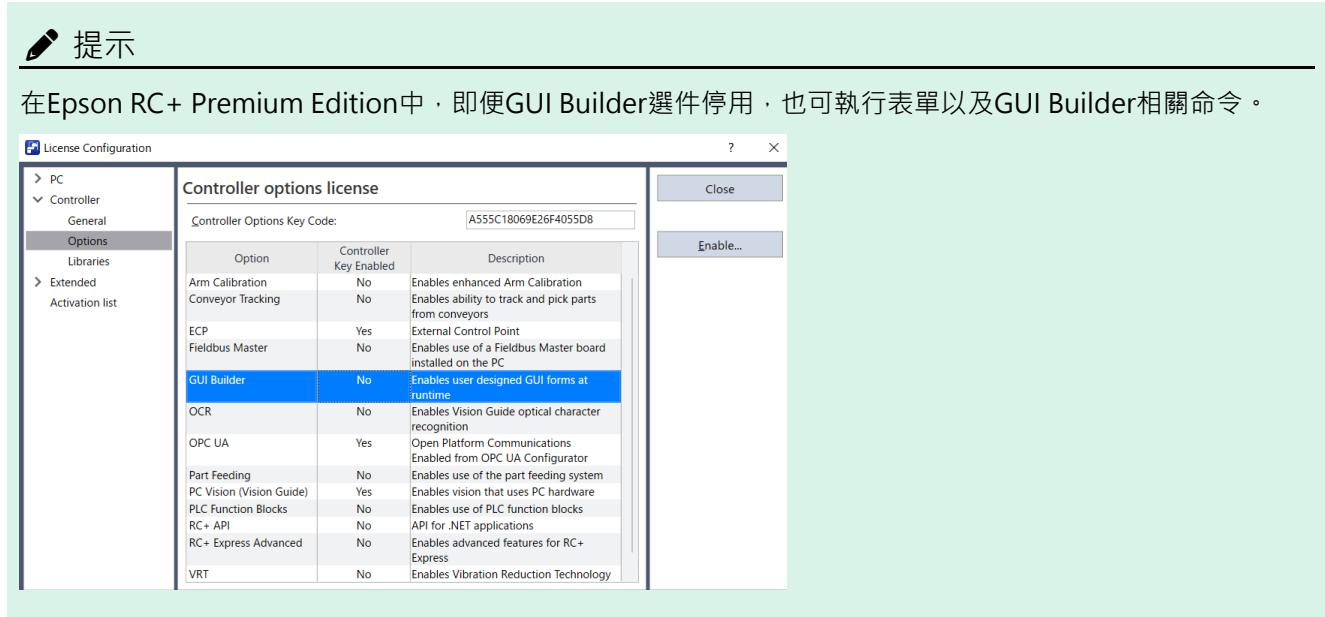


2. 若提供函式庫工具，透過[工具]-[GUI Builder]建立作為函式庫工具顯示的表單，並設為啟動表單。
如需函式庫工具的相關資訊，請參閱以下內容。

使用函式庫工具

如需GUI Builder使用方法的資訊，請參閱以下手冊。

「EPSON RC+ 8.0選件GUI Builder 8.0」



3. 如有需要，建立作為函式庫手冊顯示的PDF檔案、作為函式庫工具圖示顯示的影像檔案、裝置設定和手冊等添加檔案。

提示

- 在Epson RC+ 8.0 Ver8.1.0.0中，請勿將其他函式庫登錄至函式庫用專案。具體而言，無法在函式庫A內使用其他的函式庫B建立函式庫A。Library Builder啟動時發生錯誤。
- 可將函式庫內使用的設定值作為函式庫的使用者之專案資料[任意名稱].libcfg保留。
例如，若提供函式庫工具，可將使用者透過GUI設定的值保留在[任意名稱].libcfg中，下次啟動時從該檔案載入上次的設定值並重現上次的設定狀態。
請將[任意名稱]指定為不與其他函式庫重複的名稱。

3.3 建立函式庫

1. 打開函式庫用專案。
2. 在[擴張]選單中選擇[Library Builder]。
3. 成功建立專案後，[Library Builder]對話方塊打開。

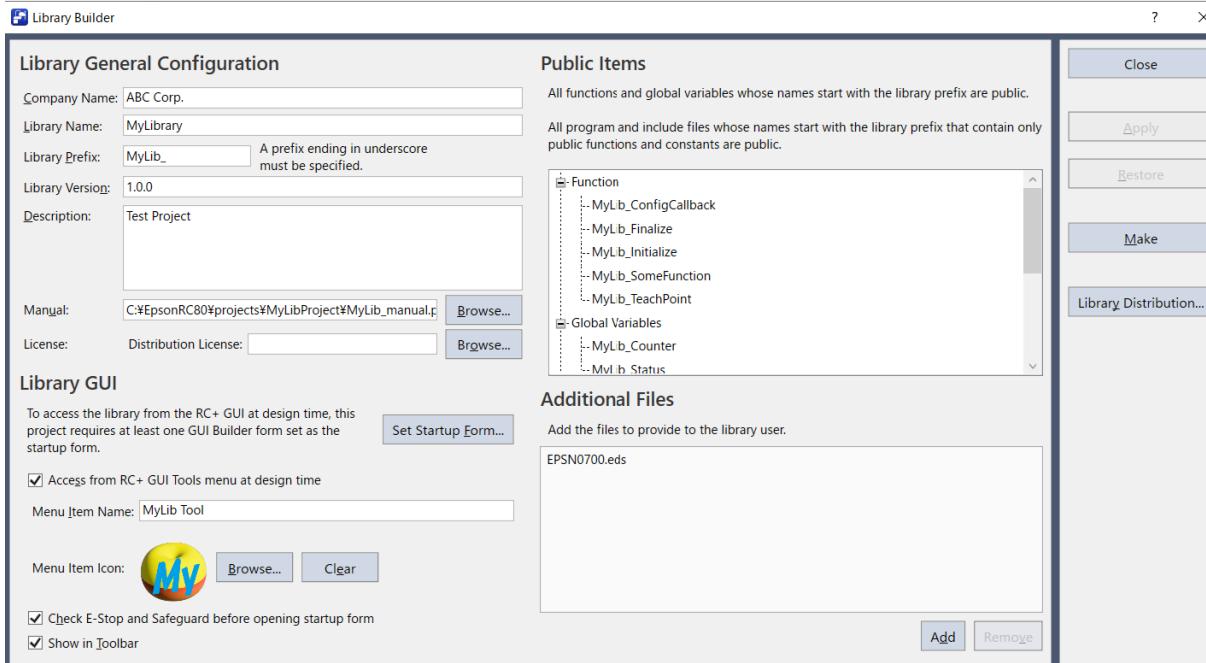
提示

根據TP4的操作模式如下所示動作。

·AUTO時，雖可使用Epson RC+，但在Epson RC+ 8.0 Ver8.1.0.0中無法建立函式庫（無法啟動Library

Builder)。

- TEACH、TEST時，無法使用Epson RC+，因此也無法建立函式庫。
- 無論TP4的操作模式如何，均可在專案中使用現有函式庫。
- 如需TP4操作模式的相關資訊，請參閱以下手冊。
「機器人控制器選件 教導器TP4手冊 - 操作模式 (TEACH、AUTO、TEST)」



項目	描述
公司名	輸入公司名。 最多32個一位元組字元。
函式庫名	輸入函式庫名。 最多32個一位元組字元。
前綴	輸入前綴名。 最多10個一位元組字元，需以底線結束。
版本	輸入函式庫的版本。 最多32個一位元組字元。
註解	輸入函式庫的說明。 最多255個字元。
手冊	設定所建立函式庫的手冊 (PDF檔案)。
Distribution License	輸入EPSON發佈的Distribution License。 也可從[瀏覽]按鈕載入記載授權的PDF檔案。
設定啟動表單	顯示啟動表單設置畫面。將用作函式庫GUI (函式庫工具) 初始顯示畫面的表單設為啟動表單。

項目	描述
開發時從RC+的工具選單使用函式庫GUI	將透過GUI Builder建立的表單作為從工具選單啟動的函式庫工具內建在函式庫中。專案內存在設為啟動表單的表單時可核取。 預設為關閉。 如需函式庫工具的相關資訊，請參閱以下內容。 使用函式庫工具
選單顯示名	輸入作為函式庫工具顯示在工具選單中的名稱。 最多32個一位元組字元。
選單顯示圖示	設定作為函式庫工具顯示在工具選單中的圖示。
啟動表單啟動時檢查緊急停止、安全門的狀態	在啟動表單（函式庫工具）啟動時，檢查緊急停止、安全門的狀態。 預設為開啟。
顯示於工具列	在工具列上顯示啟動函式庫工具的圖示。
公開資訊	顯示已添加前綴的函數、全域變數、保留變數、程式檔案、含入檔案和視覺順序。
添加檔案	對函式庫中使用的裝置添加裝置的設定檔案（現場匯流排設定用檔案(eds)等）、手冊等，以便使用者設定、參照。 複製至C:\EpsonRC80\¥Libraries資料夾下該函式庫資料夾下的AdditionalFiles資料夾中。
關閉	關閉對話方塊。
套用	儲存變更。
復原	返回之前的設定。
建立	建立函式庫。
函式庫發佈	函式庫發佈精靈啟動。輸出向使用者發佈的函式庫和驗證金鑰。 如需詳細資訊，請參閱以下內容。 建立發佈用函式庫

4. 指定前綴。包含指定前綴的函數、全域變數、保留變數、程式檔案、含入檔案和視覺順序自動添加至公開資訊區。

提示

程式檔案、含入檔案即便在檔案名中添加了前綴，若檔案內包含未添加前綴的函數、常數，則不作為公開資訊顯示。

5. 指定公司名、函式庫名、版本、註解、手冊檔案、添加檔案。

6. 若提供函式庫工具，核取[開發時從RC+的工具選單使用函式庫GUI]後，指定在Epson RC+工具選單中顯示的選單顯示名和選單顯示圖示。

若核取[啟動表單啟動時檢查緊急停止、安全門的狀態]，控制器在緊急停止狀態、安全門打開的狀態下顯示警告訊息，在這些狀態解除前，函式庫工具不會啟動。

7. 點選[套用]按鈕儲存設定變更。

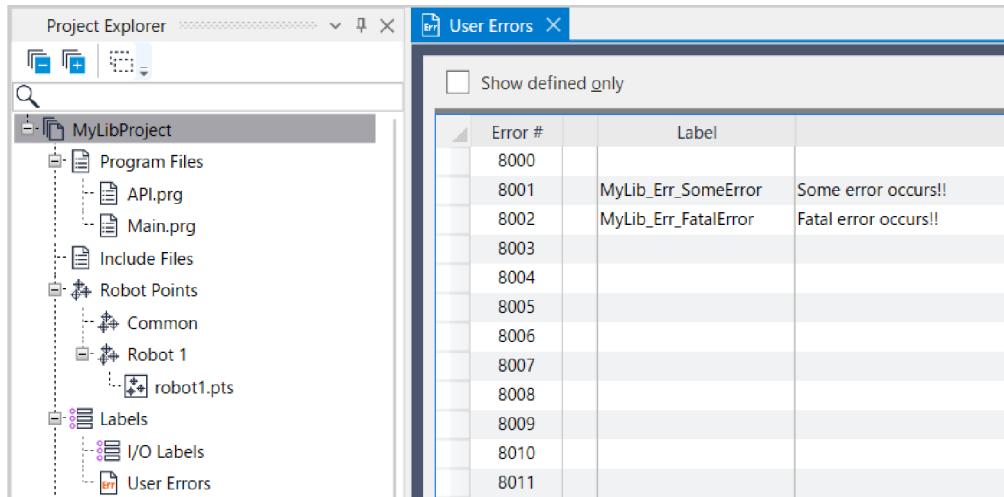
8. 點選[建立]按鈕建立函式庫。

在C:\EpsonRC80\¥Libraries資料夾下產生建立的函式庫檔案和zip檔案。

3.4 報告函式庫內所定義使用者錯誤

1. 定義使用者錯誤。

此時，不僅需定義訊息，也務必定義標籤。



2. 建立使用者錯誤報告用函數。

```
Function RaiseError(errLabel$ As String)
    Integer errNum

    errNum = UserErrorNumber(errLabel$)      ' Get user error number from label
    If errNum <> -1 Then
        Error errNum
    EndIf
EndFunction
```

如需UserErrorNumber、Error的相關資訊，請參閱以下手冊。

「SPEL+語言參考」

3. 實施錯誤處理。

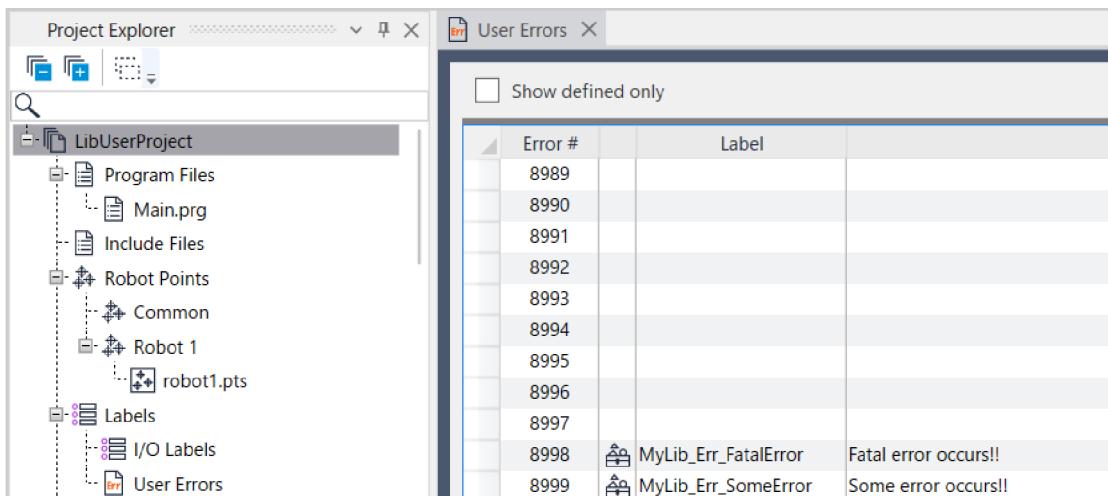
函式庫用專案的程式碼內錯誤處理呼叫上述使用者錯誤報告用函數。

```
Function SomeInternalLibraryFunction
    ' Error occurs so throw an error
    RaiseError("MyLib_Err_SomeError")
EndFunction
```

4. 依照下述步驟建立函式庫。

[建立函式庫](#)

5. 使用函式庫時，將上述步驟建立的函式庫登錄至專案後，從使用者錯誤結尾開始，依次登錄至空閒的錯誤編號。



與建立函式庫時的錯誤編號不同，在函式庫側依據使用者錯誤標籤報告錯誤，因此顯示正確訊息。

3.5 使用回呼函數

回呼函數是可在函式庫之共用函數處理中放入專案處理的功能。

函式庫的開發者在公共程式檔案中透過函式庫指定的函數名、引數和傳回值，描述回呼函數的雛形（骨架程式碼）和說明。

函式庫的使用者在添加函式庫至專案時，公共程式檔案會被添加，因此可查看檔案內的說明，並實施必要的處理。

例：回呼函數記載範例

```
Function Lib_Callback(num As Integer) As Boolean
    'XXX的處理後從函式庫呼叫。
    '參數：檢查num的值是否正確，若正確傳回True，否則傳回False。
End
```

具體有以下用途。

· 在函式庫處理中，透過使用者程式改變後續處理。

【具體範例】

- 函式庫監視輸入位元的值，若Off狀態持續一定時間以上，將呼叫Boolean型傳回值的回呼函數。傳回值為False時繼續處理，True時結束處理。
- 函式庫的使用者透過回呼函數實作如下，決定繼續/結束後續處理。
 - 另一個輸入位元為On時：透過MsgBox命令讓操作員選擇繼續/結束，並將結果作為回呼函數的傳回值。
 - 另一個輸入位元為Off時：傳回值設為True，無需操作員選擇即結束處理。

· 共用函數處理時間較長時的進度顯示和中斷。

【具體範例】

- 函式庫定期呼叫透過引數表示進度百分比、Boolean型傳回值的回呼函數。傳回值為True時，結束處理。
- 函式庫的使用者在GUI Builder的表單中，用GSet取得透過回呼函數得到的進度百分比並顯示。此外，若按下表單上的中止按鈕，在呼叫回呼函數時將回呼函數的傳回值設為True。

3.6 使用libcfg檔案的同步

SPEL專案具有如同點資料那樣，在檔案中保留程式動作所需的設定和計算值的功能。

動作中儲存至控制器，也隨時同步至運行RC+的PC之專案中，因此可順利進行SPEL專案的開發、維護、向其他環境的移植等。

使用函式庫時，根據操作對象的設備和功能，可能會產生必需的資訊，因此對在任意格式下可使用的檔案提供相同功能。

想保留下述資訊時請考慮。

- 功能和設備參照的設定檔案
- 功能和設備保留·參照計算結果的檔案
- 保留·監視功能和設備動作狀態的記錄檔案

若在控制器內操作副檔名為「libcfg」的檔案，將在RC+連接時進行同步確認並放入PC側。無格式限制。

例：libcfg檔案使用範例：檔案匯出

```
Function WriteSettingsToFile(fileName$ As String, ... , paramB As Integer)
    Integer iFileID

    iFileID = FreeFile
    ChDisk FLASH                         '指定控制器的專案資料夾
    WOpen "Lib1.libcfg" As #iFileID     '在所需模式下打開〈

    Print #iFileID, "Name: ", name$   '寫入設定的類型和值

    '按參數數量重複操作。

    Print #iFileID, "ParamB: ", paramB '寫入設定的類型和值
    Close #iFileID
End
```

例：libcfg檔案使用範例：檔案讀取

```
Function ReadSettingsFromFile(fileName$ As String, ByRef name$ As String, ... ,
ByRef paramB As Integer)
    Integer iFileID
    Integer iPos
    String Buf$

    iFileID = FreeFile
    ChDisk FLASH                         '指定控制器的專案資料夾
    ROpen "Lib1.libcfg" As #iFileID     '在所需模式下打開

    Line Input #iFileID, Buf$
    iPos = InStr(Buf$, ": ")           '細分設定的類型和值後載入。
    name$ = Mid$(Buf$, iPos + 2)

    '按參數數量載入。

    Line Input #iFileID, Buf$
    iPos = InStr(Buf$, ": ")           '細分設定的類型和值後載入。
    paramB = Val(Mid$(Buf$, iPos + 2))

    Close #iFileID
End
```

若因所使用設備的規格等需使用特定副檔名，請使用「UploadFileAfterStop」命令。透過函式庫內的描述，在任務結束後從控制器內放入指定的檔案。但動作時需要連接RC+。

4. 建立發佈用函式庫

4.1 建立發佈用函式庫

說明如何建立用於向有意使用函式庫之使用者（以下稱函式庫使用者）發佈的函式庫。

發佈函式庫時，可指定哪些使用者能使用函式庫。根據發佈形式進行函式庫價格設定等情況下請使用。

符號	可使用函式庫的使用者	用途	必需的資訊
a	僅指定控制器的使用者	允許透過特定控制器使用。 函式庫使用者必須按想執行動作的控制器台數購買函式庫並啟用。 請函式庫開發者按台數取得相應的序號後，發佈驗證金鑰。	Distribution License, 函式庫使用者的控制器序號（從函式庫使用者的RC+也可輸出。請事先取得）
b	僅指定PC的使用者	允許透過在特定PC上運行的RC+之虛擬控制器使用。 與實際控制器不同。 函式庫使用者必須按想執行動作的PC台數購買函式庫並啟用。 請函式庫開發者按台數取得相應的PC硬體ID後，發佈驗證金鑰。	Distribution License, 函式庫使用者的PC硬體ID (從函式庫使用者的RC+可輸出。請事先取得)
c	取得驗證金鑰的所有使用者	函式庫使用者使用時不受控制器、PC限制。 函式庫使用者必須啟用。	Distribution License, 函式庫開發者規定的任意字串（僅用於設定。 啟用函式庫使用者時不需要）
d	取得函式庫的所有使用者	函式庫使用者使用時不受控制器、PC限制。 函式庫使用者不需要啟用即可使用。	Distribution License

- 打開要發佈的函式庫之專案。
- 在[擴張]選單中選擇[Library Builder]。
- 成功建立專案後，[Library Builder]對話方塊打開。

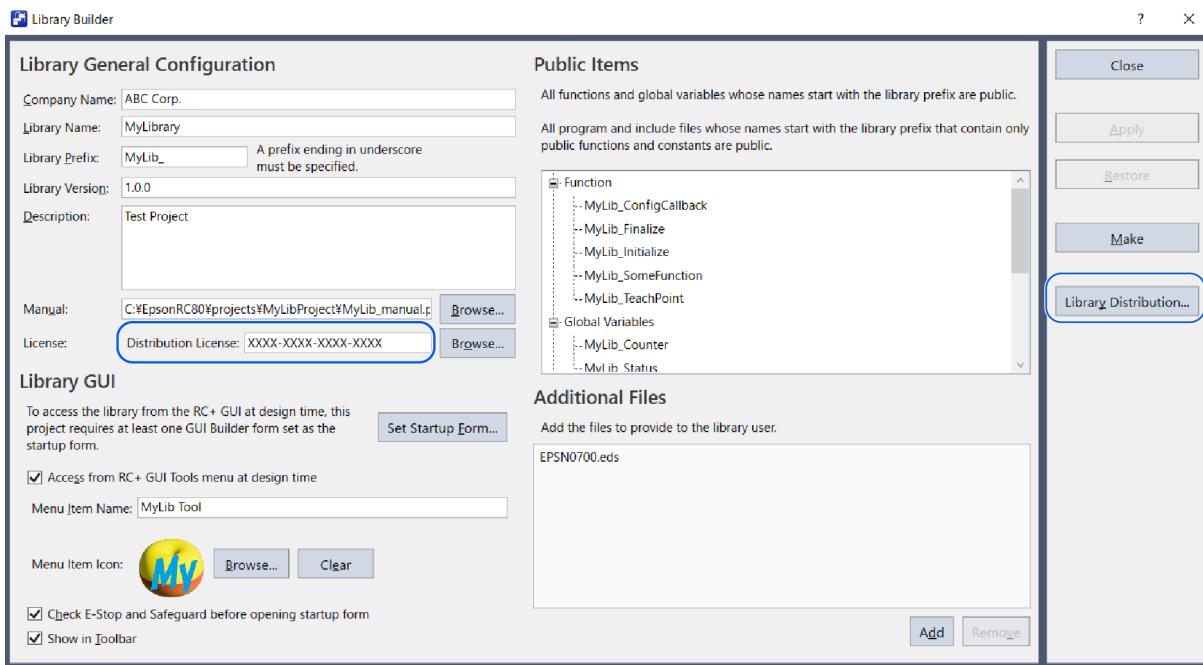


提示

在Epson RC+ 8.0 Ver8.1.0.0中無法透過TP4（操作模式：AUTO）使用Library Builder。
如需TP4（操作模式：AUTO）的相關資訊，請參閱以下手冊。

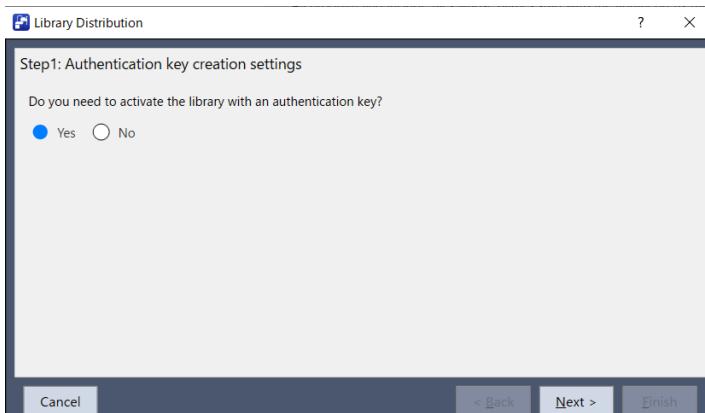
「機器人控制器選件 教導器TP4手冊 - 操作模式（TEACH、AUTO、TEST）」

- 輸入EPSON發佈的Distribution License。



5. 點選[函式庫發佈]按鈕。

6. 在顯示的[函式庫發佈]精靈中，選擇啟用發佈的函式庫是否需要驗證金鑰。



- [是]：可使用函式庫的使用者除d以外時選擇。繼續前往下一步。
- [否]：可使用函式庫的使用者為d時選擇。繼續前往步驟9。

7. 根據想指定的函式庫使用者，選擇啟用函式庫的裝置。



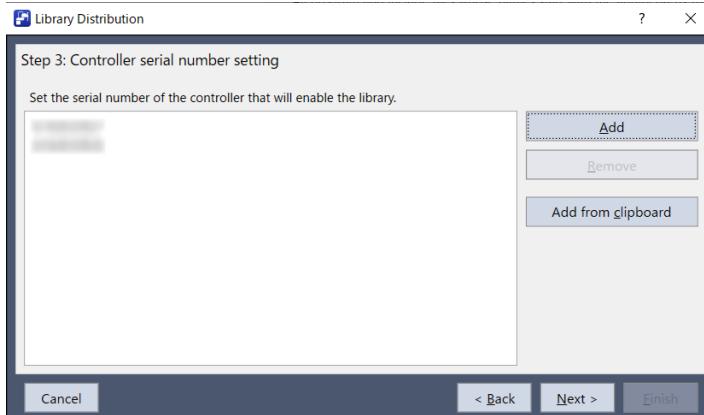
- [控制器]：可使用函式庫的使用者為a時選擇。
- [PC(虛擬控制器)]：可使用函式庫的使用者為b時選擇。

- [不指定]：可使用函式庫的使用者為c時選擇。

8. 依據所選取的裝置，輸入以下資訊。

- 若選擇[控制器]：

從[添加]按鈕載入從函式庫使用者取得的控制器序號記載csv檔案。
若點選[從剪貼簿添加]按鈕，將添加剪貼簿的字串。複製貼上序號時請使用。

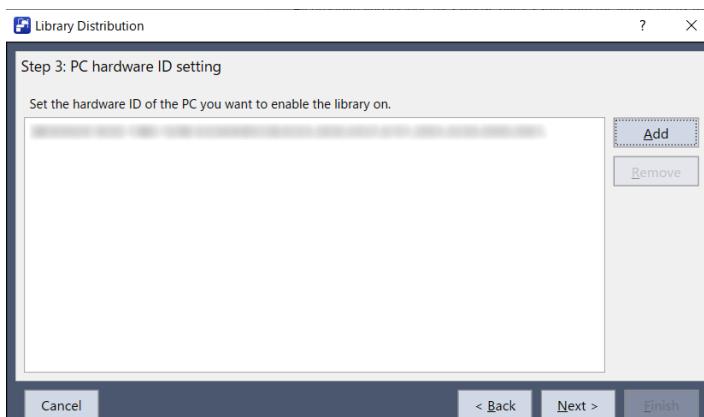


提示

可載入多個序號，一次產生多個控制器的驗證金鑰。

- 若選擇[PC(虛擬控制器)]：

從[添加]按鈕載入從函式庫使用者取得的PC硬體ID記載csv檔案。

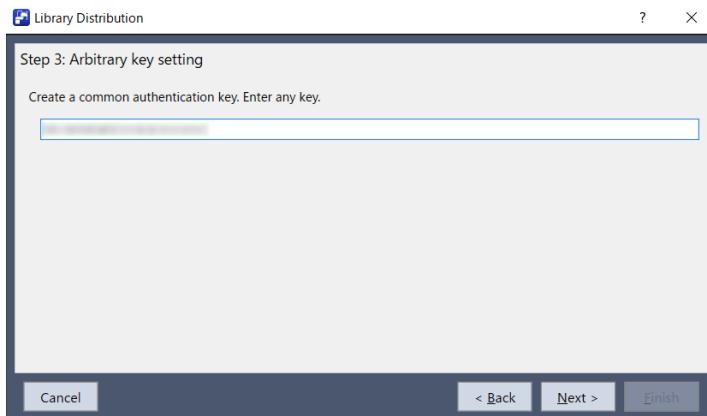


提示

可載入多個PC硬體ID，一次產生多個PC的驗證金鑰。

- 若選擇[不指定]：

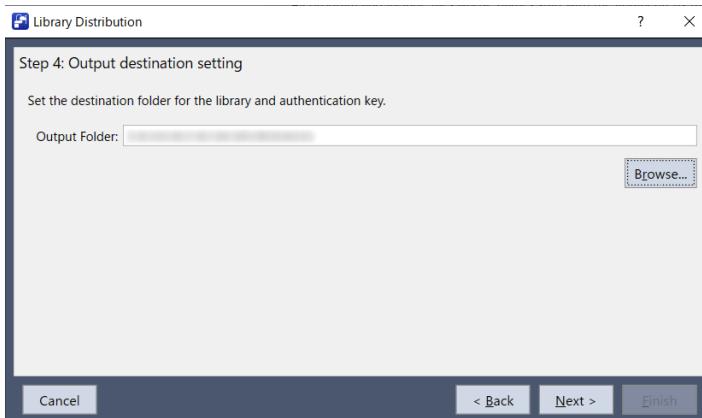
輸入任意金鑰。可使用最多32個字元的英數字元和底線、連字號。
即使不輸入金鑰，也可建立「c. 僅已登錄的使用者可使用」的發佈函式庫。



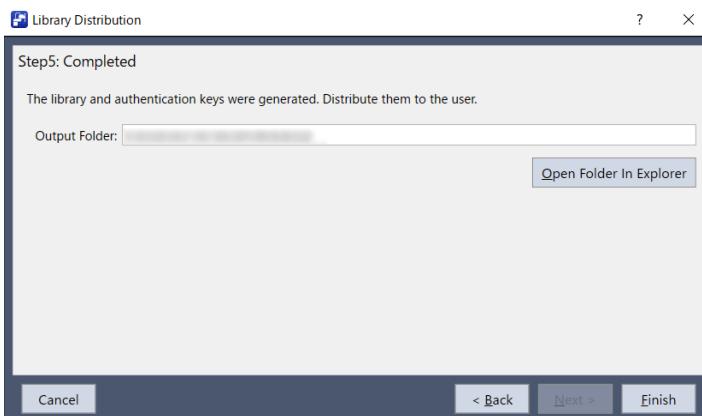
提示

產生的驗證金鑰不論在任何裝置，均可啟用函式庫。

9. 指定函式庫和驗證金鑰的輸出目的地。



10. 輸出函式庫和驗證金鑰（若步驟6中選擇[否]，則不輸出驗證金鑰）。 若點選[使用檔案總管打開資料夾]按鈕，將顯示輸出目的地資料夾。



5. 使用函式庫

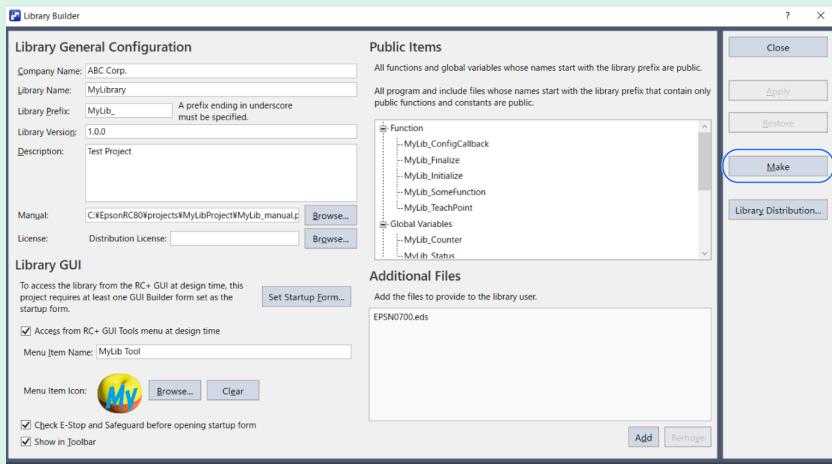
5.1 匯入函式庫

說明將獲得的函式庫放入Epson RC+的方法。

1. 函式庫以zip檔案格式發佈。
2. 在[擴張]選單中選擇[匯入函式庫]。
3. 從顯示的函式庫選擇對話方塊中選擇要匯入的函式庫。
4. 點選[開啟]。在C:\EpsonRC80\libraries資料夾下產生匯入的函式庫檔案和zip檔案。若已有同名函式庫，將顯示覆寫確認訊息。

提示

在相同Epson RC+中透過Library Builder的[建立]按鈕建立的函式庫無需匯入。



5.2 啟用函式庫

對於需要授權的函式庫，需要啟用函式庫。此處說明匯入的函式庫之啟用方法。

若要啟用函式庫，需要函式庫的驗證金鑰。

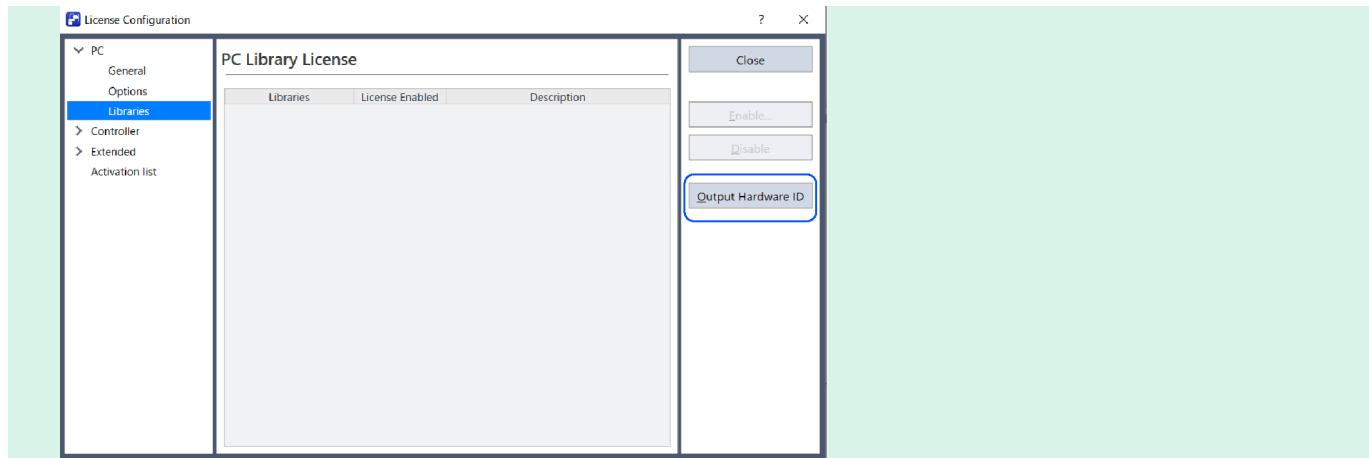
若不啟用，則無法使用該函式庫。對於無需授權的函式庫，不啟用即可使用。

提示

驗證金鑰由函式庫創作者發佈。如有需要，請將以下其中一項或兩項資訊傳遞給函式庫創作者。

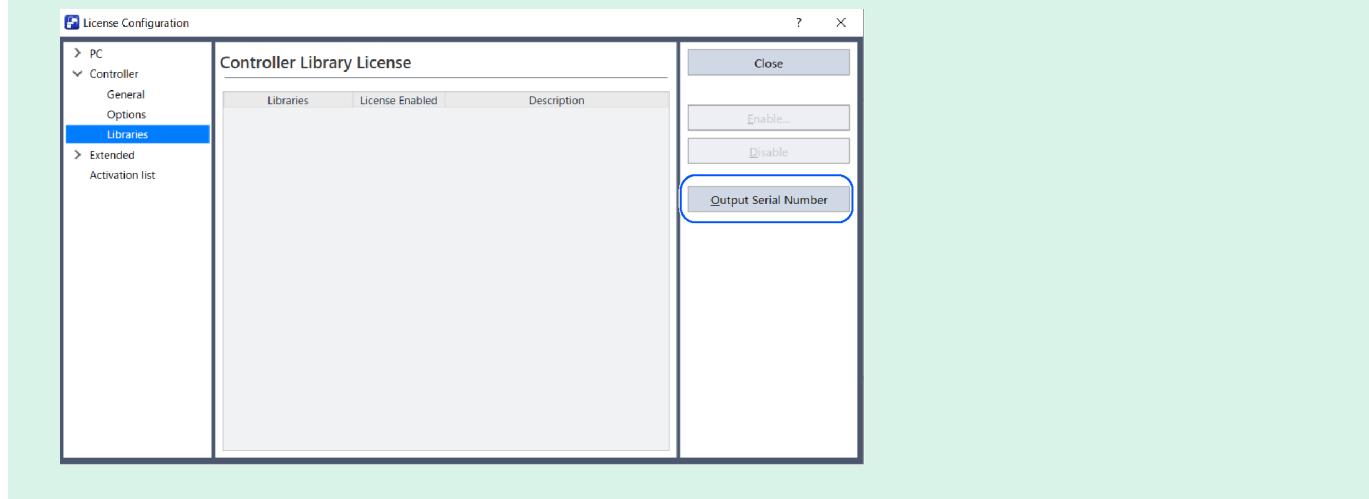
■ PC硬體ID

可在[設定] - [授權設定]畫面中，透過使用[PC] - [函式庫]顯示的[輸出固體ID]按鈕保存為csv檔案。



■ 控制器序號

可在[設定] - [授權設定]畫面中，透過使用[控制器] - [函式庫]顯示的[輸出序號]按鈕，將連線的控制器序號保存為CSV檔案。



提示

若使用未啟用的函式庫，建立函式庫或函式庫工具啟動時發生錯誤。

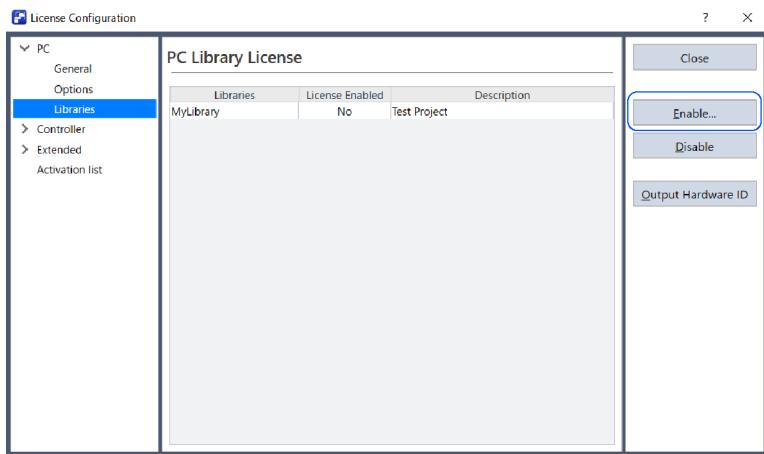
1. 點選[設定] - [授權設定]。
2. 在[授權設定]畫面的樹狀檢視中，選擇[PC] - [函式庫]或[控制器] - [函式庫]。

提示

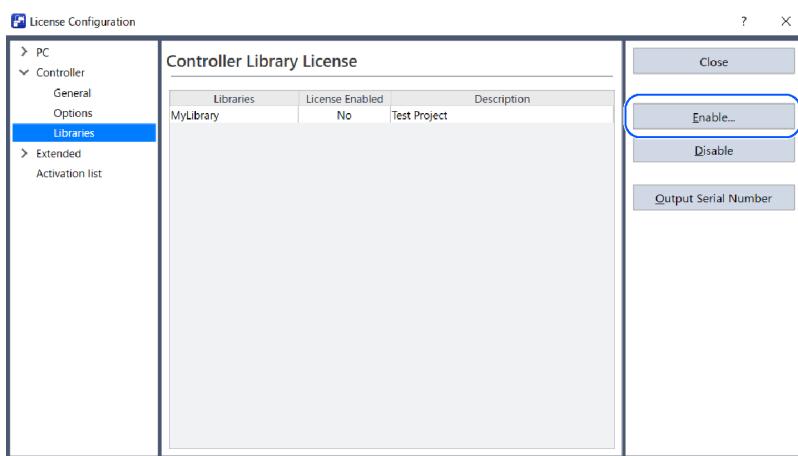
[控制器] - [函式庫]節點僅當連線實際控制器時顯示。

3. 確認顯示目的函式庫後，點選[啟用]按鈕。

■ PC函式庫授權時

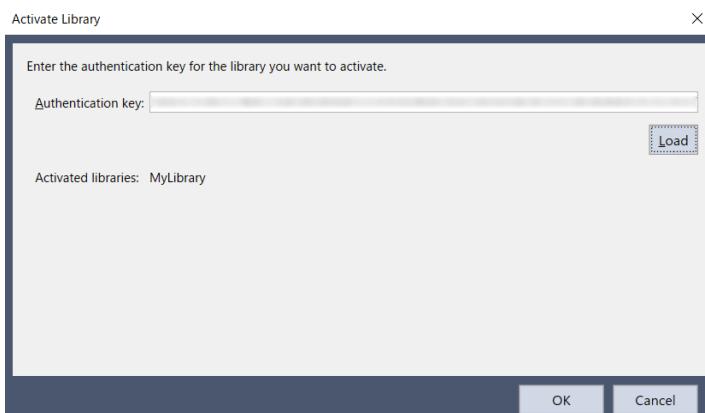


■ 控制器函式庫授權時



C:\EpsonRC80\Libraries資料夾中的函式庫清單顯示。

- 在顯示的[啟用函式庫]對話方塊中，輸入驗證金鑰或從[載入]按鈕載入csv檔案。若驗證金鑰正確，將顯示透過驗證金鑰啟用的函式庫。

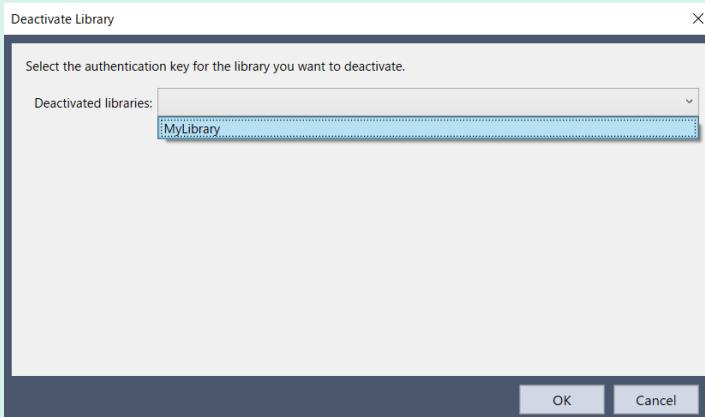


- 點選[OK]按鈕後，確認目的函式庫已啟用。

Libraries	License Enabled	Description
MyLibrary	Yes	Test Project

提示

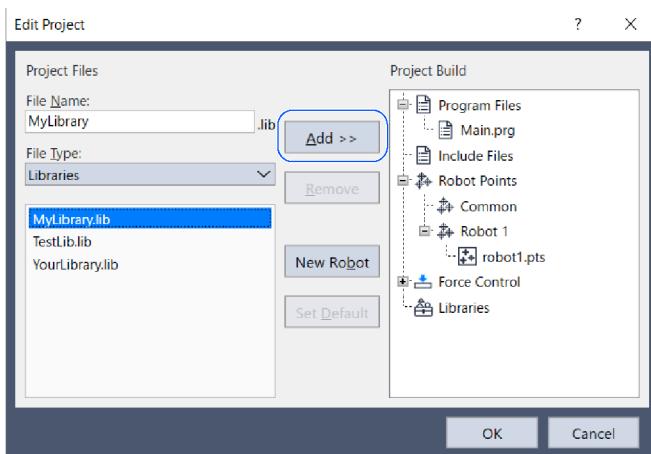
若要停用，點選[停用]按鈕後，從[停用函式庫]對話方塊的下拉式清單中選擇要停用的函式庫，然後點選[OK]按鈕。



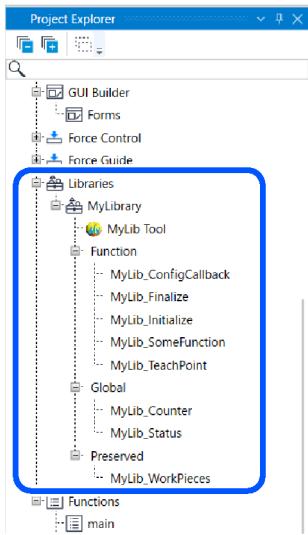
5.3 將函式庫登錄至專案

說明將函式庫登錄至專案的方法。

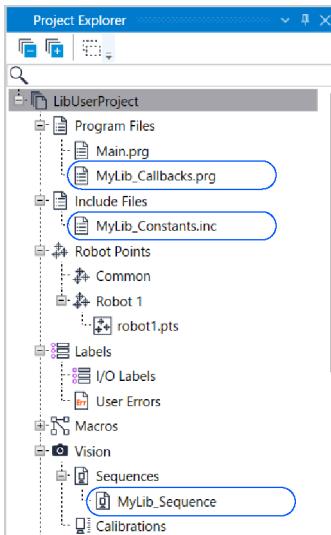
1. 建立使用函式庫的專案。
2. 選擇[專案]-[編輯專案]，或者在專案總管中的函式庫點選右鍵並選擇添加函式庫，打開專案編輯對話方塊。



3. 在檔案類型中選擇函式庫檔案。
4. 顯示C:\EpsonRC80\LIBRARIES資料夾內所有函式庫的清單。
5. 選擇函式庫後，點選[登錄]按鈕。
6. 點選[OK]按鈕。
7. 在專案總管上顯示可使用的公開函數、公開變數。



8. 函式庫和函式庫的公開檔案添加至專案。公開檔案的名稱從函式庫定義的前綴開始。有無公開檔案因函式庫而異。



9. 要添加多個函式庫時，重複此操作。

對於1個專案，最多可添加5個函式庫。

10. 將要執行的函式庫函數記錄至程式。

11. 如有需要，請編輯函式庫的公開檔案。

提示

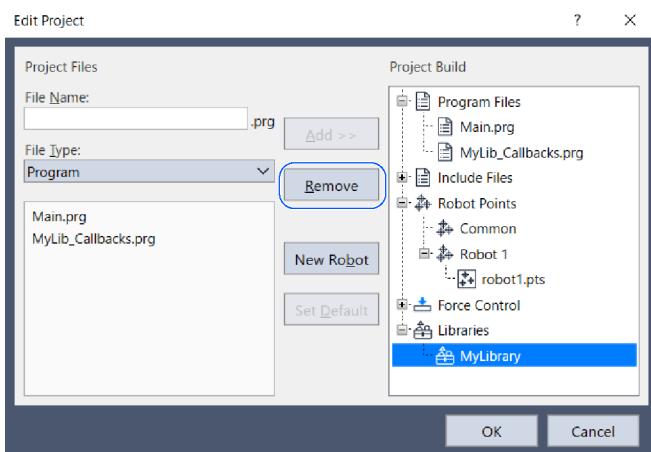
請勿登錄具有相同前綴的多個函式庫。函數、變數可能會重複並發生錯誤。

5.4 將函式庫從專案取消登錄

說明將已登錄的函式庫從專案取消的方法。

1. 在專案總管中要取消的函式庫點選右鍵並選擇[從專案中排除]後，函式庫被取消登錄。
或者選擇[專案]-[編輯專案]，打開專案編輯對話方塊。
 - i. 在專案設定樹內選擇要取消的函式庫後，點選[取消]按鈕。

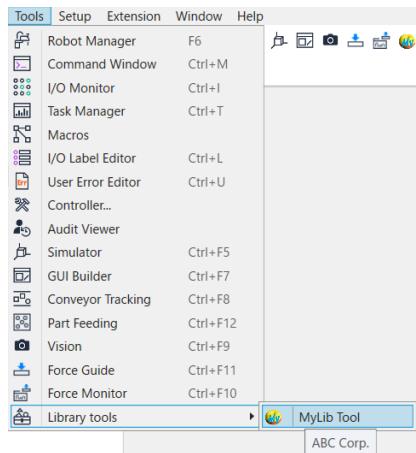
ii. 點選[OK]按鈕後，專案編輯對話方塊關閉，函式庫被取消登錄。



2. 如有需要，請將與函式庫一起添加的函式庫公開檔案也取消登錄。不會與函式庫同時取消登錄。

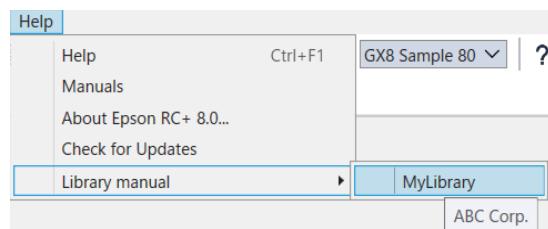
5.5 使用函式庫工具

函式庫有時支援透過GUI設定、執行。此時，[工具]-[函式庫工具]下將添加選單，請由此執行。如需函式庫工具的相關資訊，請參閱各函式庫的手冊或聯絡各函式庫的開發者。



5.6 顯示函式庫的手冊

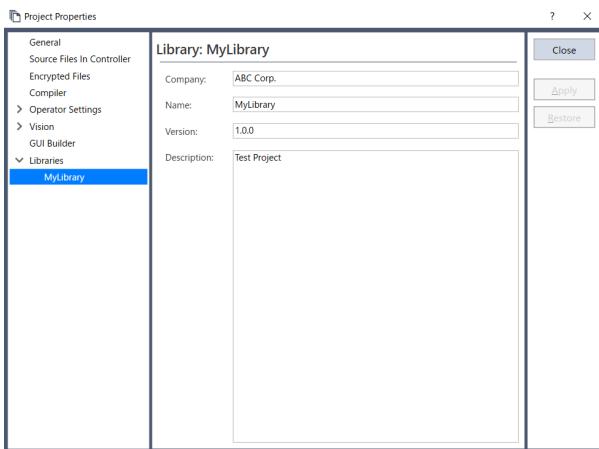
若函式庫中提供了手冊，可從[說明]-[函式庫手冊]參閱。



5.7 設定函式庫的屬性

可確認專案中所登錄函式庫的屬性。

1. 從[專案]-[屬性]顯示專案屬性對話方塊。
2. 若函式庫已登錄，則函式庫將被添加至樹狀檢視，將其展開後即顯示已登錄的函式庫。
3. 選擇各函式庫名後，將顯示函式庫的建立者、函式庫名、版本和說明。



5.8 函式庫使用者錯誤定義

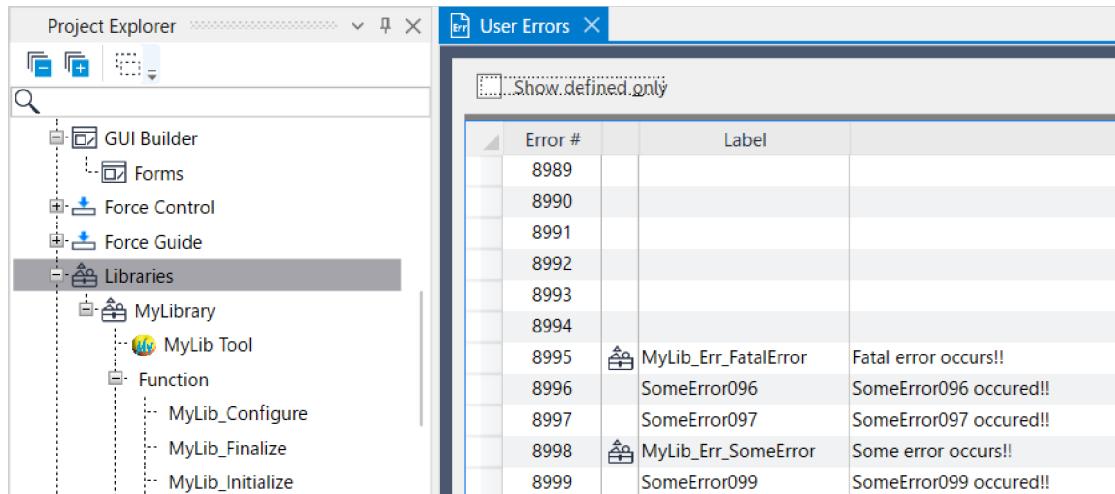
已登錄的函式庫使用者錯誤從錯誤編號8999開始，按遞減順序使用未定義的編號。函式庫使用者錯誤定義無法更改，請注意。

函式庫添加前後的範例：

添加前

Error #	Label	
8989		
8990		
8991		
8992		
8993		
8994		
8995		
8996	SomeError096	SomeError096 occurred!!
8997	SomeError097	SomeError097 occurred!!
8998		
8999	SomeError099	SomeError099 occurred!!

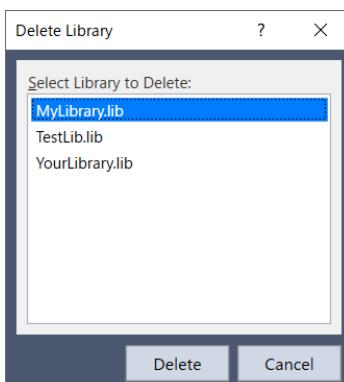
添加後



5.9 刪除函式庫

說明將不需要的函式庫從Epson RC+刪除的方法。

1. 在[擴張]選單中選擇[刪除函式庫]。
2. 從顯示對話方塊的清單中選擇要刪除的函式庫。
清單為目前Epson RC+保留的函式庫。
3. 點選[刪除]按鈕。C:¥EpsonRC80¥Libraries資料夾下相應的函式庫檔案被刪除。



6. SPEL+命令參考

6.1 SPEL+命令清單

無法從除函式庫之外的SPEL+專案、公共程式檔案使用。

僅可在函式庫中使用的SPEL+命令如下。

命令/函數	描述/用途
ArchReserve	動態取得函式庫內可獨佔使用的Arch編號。
ArmLib	選擇動態取得的手臂編號。
ArmReserve	動態取得函式庫內可獨佔使用的手臂編號。
LibGetInfo	取得SPEL+專案中包含的函式庫資訊。
LocalReserve	動態取得函式庫內可獨佔使用的區域編號。
PointReserve	動態取得函式庫內可獨佔使用的點編號。
SignalReserve	動態取得函式庫內可獨佔使用的訊號編號。
SyncLockReserve	動態取得函式庫內可獨佔使用的SyncLock編號。
TaskReserve	動態取得函式庫內可獨佔使用的任務編號。
TimerReserve	動態取得函式庫內可獨佔使用的計時器編號。
TLReserve	動態取得函式庫內可獨佔使用的工具編號。
ToolLib	選擇動態取得的工具編號。
UploadFileAfterStop	所有任務停止後，從控制器將指定檔案複製到PC的專案資料夾。

6.2 ArchReserve函數

動態取得目前函式庫內可使用的Arch編號。

格式

ArchReserve

參數

無

傳回值

預約好的Arch編號

描述

透過預約Arch編號確保可使用，而不被其他函式庫使用。

在7至13中由大到小，依次預約尚未預約的編號。

定義至已預約Arch編號的資訊為臨時值，所有任務結束時預約取消，值也消失。透過將Arch編號指定至ArmClr也可取消預約。

另請參閱

Arch, ArchClr

ArchReserve函數使用範例

```
Integer i  
i = ArchReserve  
  
Arch i,20,20  
Jump3 P2, P3-TLZ(100), P3 C(i) '使用已預約的Arch編號動作。  
  
ArchClr i
```

6.3 ArmLib函數

選擇動態取得的手臂編號。

格式

ArmLib 手臂編號

參數

手臂編號 以整數值或運算式指定。有效值範圍為16至31。可選擇此函式庫已預約的手臂。

描述

將已預約手臂指定為執行機器人命令的手臂。
可選擇呼叫此命令之函式庫已預約的手臂編號。
要顯示所選擇的手臂編號，請執行Arm。
透過所選手臂編號的動作也與Arm類似。
呼叫此命令的使用者函數結束後，返回執行函數前的手臂編號。

另請參閱

Arm, ArmClr, ArmDef, ArmSet, [ArmReserve函數](#)

ArmLib函數使用範例

```
Integer i  
  
i = ArmReserve  
  
ArmSet i, X, Y, Z, U  
ArmLib i  
Move P1      '使用已預約的手臂編號動作，而非目前所選擇的編號。
```

6.4 ArmReserve函數

動態取得目前函式庫內可使用的手臂編號。

格式

ArmReserve

參數

無

傳回

預約好的手臂編號

描述

透過預約確保可使用，而不被其他函式庫使用。在16至31中由大到小，依次預約尚未預約的手臂編號。
定義至已預約手臂編號的資訊為臨時值，所有任務結束時，值也與預約一同消失。或者可透過將手臂編號指定至
ArmClr以取消預約。

另請參閱

Arm, [ArmLib函數](#), ArmClr, ArmDef, ArmSet

ArmReserve函數使用範例

```
Integer i  
  
i = ArmReserve  
  
ArmSet i, X, Y, Z, U  
ArmLib i  
Move P1 '使用已預約的手臂編號動作。  
ArmClr i
```

6.5 LibGetInfo

取得SPEL+專案中包含的函式庫資訊。

格式

LibGetInfo 函式庫名, ByRef 函式庫ID, ByRef 版本, ByRef 註解, ByRef 函式庫路徑

參數

- 函式庫名：透過字串指定函式庫名稱（最多32個字元）。
- 函式庫ID：指定取得函式庫ID的字串變數。函式庫ID的值在每次建立函式庫時改變。
- 版本：指定取得相應函式庫版本的字串變數。
- 註解：指定取得相應函式庫註解的字串變數。
- 函式庫路徑：指定取得儲存函式庫之路徑的字串變數。以「[Data資料夾]\Libraries\函式庫名」表示。

另請參閱

GetProjectInfo

LibGetInfo使用範例

```
String strID$  
String strVer$  
String strDescript$  
String strPath$  
LibGetInfo "test", ByRef strID$, ByRef strVer$, ByRef strDescript$, ByRef strPath$  
Print "ID = " + strID$  
Print "Version = " + strVer$  
Print "Description = " + strDescript$  
Print "Path = " + strPath$
```

6.6 LocalReserve函數

動態取得目前函式庫內可使用的區域編號。

格式

LocalReserve

參數

無

傳回

預約好的區域編號

描述

透過預約確保可使用，而不被其他函式庫使用。在16至31中由大到小，依次預約尚未預約的區域編號。

定義至已預約區域編號的資訊為臨時值，所有任務結束時，值也與預約一同消失。或者可透過將區域編號指定至 LocalClr以取消預約。

已預約的區域編號中定義的點作為區域編號0保存。

另請參閱

Local, LocalClr, LocalDef, P#, Arc, Go, Move, Jump

LocalReserve函數使用範例

```
Integer i  
  
i = LocalReserve  
  
Local i, X, Y, Z, U  
P1 = Here /(i)  
Move P1 '使用指定的區域編號動作。
```

6.7 PointReserve函數

動態取得目前函式庫內可使用的點編號。

格式

PointReserve

參數

無

傳回值

預約好的點編號

描述

透過預約確保可使用，而不被其他函式庫使用。

定義至已預約點編號的資訊為臨時值，所有任務結束時，值也與預約一同消失。透過將點編號指定至PDel也可取消預約。

另請參閱

P#, PDel

PointReserve函數使用範例

```
Integer i  
i = PointReserve  
P(i) = Here  
Move P(i) +Z50 '使用指定的點編號動作。  
PDel 1
```

6.8 SignalReserve函數

預約可指定至SigWait的訊號編號。

格式

SignalReserve

參數

無

傳回

預約好的訊號編號

描述

透過預約確保可使用，而不被其他函式庫使用。為臨時值，所有任務結束時，值也與預約一同消失。保留預約專用的訊號編號，因此可降低發生函式庫的使用者所使用之訊號編號被修正的風險。

另請參閱

WaitSig

SignalReserve函數使用範例

```
Function N4_1_Call_Hidden
    Integer i1, i2, n1, n2, L1, t1

    i1 = TaskReserve
    i2 = TaskReserve

    n1 = SignalReserve
    n2 = SignalReserve

    Xqt i1, Hidden_N4_1(n1)
    Xqt i2, Hidden_N4_1(n2)

    Signal n1
    Signal n2
Fend

Function Hidden_N4_1(j As Integer)

    WaitSig j

    Print "Start Signal!"

Fend
```

6.9 SyncLockReserve函數

預約指定至SyncLock的訊號編號。

格式

SyncLockReserve

參數

無

傳回

預約好的訊號編號

描述

透過指定至SyncLock和SyncUnLock，可獨佔存取檔案等。

透過預約訊號編號確保可使用，而不被其他函式庫使用。為臨時值，所有任務結束時，值也與預約一同消失。保留預約專用的訊號編號，因此可降低發生函式庫的使用者所使用之訊號編號被修正的風險。

另請參閱

SyncLock

SyncLockReserve函數使用範例

```
Function N4_1_Call_Hidden
    Integer i1, i2, n1, n2, L1, t1

    i1 = TaskReserve
    L1 = SyncLockReserve

    Xqt i1, Hidden_N4_1(L1)

Fend

Function Hidden_N4_1(j As Integer, i As Integer)
    SyncLock i

    Print "4-1 Layer Lib!"

    SyncUnlock i
Fend
```

6.10 TaskReserve函數

預約可在執行任務時指定的任務編號。

格式

TaskReserve

參數

無

傳回

預約好的任務編號

描述

透過預約確保可使用，而不被其他函式庫使用。為臨時值，所有任務結束時，值也與預約一同消失。
由大到小依次預約尚未預約的任務編號。

另請參閱

WaitSig

TaskReserve函數使用範例

```
Function N4_1_Call_Hidden
    Integer i1, i2, n1, n2, L1, t1

    i1 = TaskReserve
    i2 = TaskReserve

    n1 = SignalReserve
    n2 = SignalReserve

    Xqt i1, Hidden_N4_1(n1)
    Xqt i2, Hidden_N4_1(n2)

    Signal n1
    Signal n2
Fend

Function Hidden_N4_1(j As Integer)

    WaitSig j

    Print "Start Signal!"

Fend
```

6.11 TimerReserve函數

預約指定至Timer的計時器編號。

格式

TimerReserve

參數

無

傳回

預約好的計時器編號

描述

透過指定至Tmr函數，可獨佔測量週期時間。

透過預約計時器編號確保可使用，而不被其他函式庫使用。為臨時值，所有任務結束時，值也與預約一同消失。保留預約專用的計時器編號，因此可降低發生函式庫的使用者所使用之計時器編號被修正的風險。

另請參閱

Tmr, TmReset

TimerReserve函數使用範例

```
Function N4_1_Call_Hidden  
    t1 = TimerReserve  
  
    TmReset t1          '重設計時器0  
  
    Xqt ...  
  
    Print Tmr(t1) / 10      '計算週期時間後顯示  
Fend
```

6.12 TLReserve函數

動態取得目前函式庫內可使用的工具編號。

格式

TLReserve

參數

無

傳回

預約好的工具編號

描述

透過預約確保可使用，而不被其他函式庫使用。為臨時值，所有任務結束時，值也與預約一同消失。或者可透過將工具編號指定至TLCI以取消預約。

在16至31中由大到小，依次預約尚未預約的工具編號。

另請參閱

TLCI, TLDef, TLSet, Tool, [ToolLib函數](#)

TLReserve函數使用範例

```
Integer i  
i = TLReserve  
  
TLSet i, X, Y, Z, U  
ToolLib i  
Move P1      '使用指定的工具編號動作。  
TLCI i
```

6.13 ToolLib函數

選擇動態取得的工具編號。

格式

ToolLib 工具編號

參數

工具編號

以整數值或運算式指定。有效值範圍為16至31。可選擇此函式庫已預約的工具。

描述

將已預約工具指定為執行機器人命令的工具。

可選擇呼叫此命令之函式庫已預約的工具編號。

要顯示所選擇的工具編號，請執行Tool。

透過所選工具編號的動作也與Tool類似。

呼叫此命令的使用者函數結束後，返回執行函數前的工具編號。

另請參閱

TLClr, TLDef, TLSet, [TLReserve函數](#), Tool

ToolLib函數使用範例

```
Integer i  
  
i = TLReserve  
  
TLSet i, X, Y, Z, U  
ToolLib i  
Move P1           ' 使用已預約的工具編號動作。
```

6.14 UploadFileAfterStop函數

所有任務停止後，從控制器將指定檔案複製到PC的專案資料夾。

格式

UploadFileAfterStop 檔案名

參數

- 檔案名：指定專案內包含的檔案名。

另請參閱

GetProjectInfo

UploadFileAfterStop使用範例

```
UploadFileAfterStop("test.csv")
```