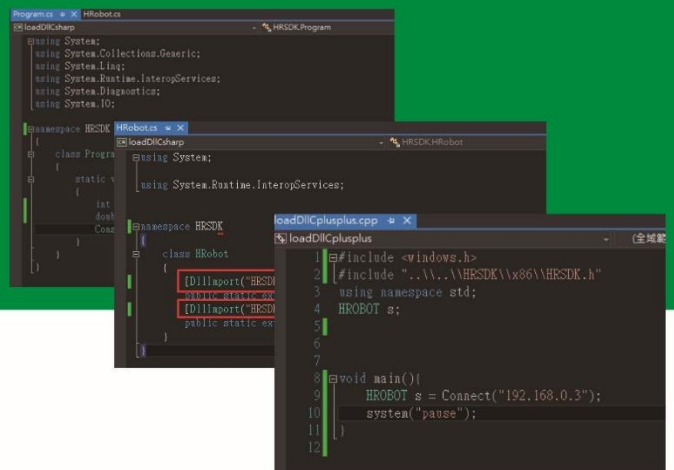


機器人軟體開發套件

使用手冊
User Manual



目錄

1. 產品介紹.....	8
1.1 功能.....	8
1.2 需求.....	8
2. 使用方法.....	9
2.1 C++	9
2.2 C#.....	14
2.3 VB.....	17
2.4 連線等級說明.....	18
2.5 操作模式說明.....	18
3. 命令清單.....	19
3.1. 連線命令.....	19
3.2. 暫存器命令.....	19
3.3. 系統變數命令.....	20
3.4. 輸入輸出命令.....	20
3.5. 座標系命令.....	22
3.6. 任務命令.....	22
3.7. 檔案管理命令.....	23
3.8. 控制器設定命令.....	23
3.9. 吋動.....	23
3.10. 運動命令.....	24
3.11. 手臂資訊命令.....	25
3.12. 通訊設定.....	25
4. 命令參考.....	26
4.1. 連線命令.....	26
4.1.1. 建立連線：	26
4.1.2. 關閉連線：	26
4.1.3. 設定連線等級：	27
4.1.4. 取得連線等級：	27
4.1.5. 取得 HRSDK 版本號：	28
4.1.6. Callback	28
4.2. 暫存器命令.....	32
4.2.1. 設定機器人計時器：	32
4.2.2. 取得機器人計時器:	32
4.2.3. 啟動機器人計時器:	33
4.2.4. 停止機器人計時器:	33

4.2.5.	取得機器人計時器狀態:	33
4.2.6.	設定機器人計時器名稱:	34
4.2.7.	取得機器人計時器名稱:	34
4.2.8.	設定機器人計數器:	34
4.2.9.	取得機器人計數器:	35
4.2.10.	設定機器人計數器名稱:	35
4.2.11.	取得機器人計數器名稱:	35
4.2.12.	設定位置暫存器座標系種類:	36
4.2.13.	取得位置暫存器座標系種類:	36
4.2.14.	設定位置暫存器座標:	36
4.2.15.	取得位置暫存器座標:	37
4.2.16.	設定位置暫存器工具、基底座標:	37
4.2.17.	取得位置暫存器工具、基底座標:	38
4.2.18.	設定位置暫存器資料:	39
4.2.19.	取得位置暫存器值:	39
4.2.20.	清除位置暫存器值:	40
4.2.21.	取得位置暫存器註解:	40
4.2.22.	設定位置暫存器註解:	40
4.3.	系統變數命令.....	41
4.3.1.	設定加速度比例:	41
4.3.2.	取得加速度比例:	41
4.3.3.	設定加速度時間:	42
4.3.4.	取得加速度時間:	42
4.3.5.	設定點對點運動速度:	42
4.3.6.	取得點對點運動速度:	42
4.3.7.	設定直線運動速度:	43
4.3.8.	取得直線運動速度:	43
4.3.9.	設定整體速度:	43
4.3.10.	取得整體速度:	44
4.3.11.	設定機器人編號:	44
4.3.12.	取得機器人編號:	44
4.3.13.	設定運動平滑半徑:	45
4.3.14.	取得錯誤代碼:	45
4.3.15.	設定數位設置:	46
4.3.16.	取得數位設置:	46
4.3.17.	設定語言:	47
4.3.18.	取得控制器現在時間:	47
4.3.19.	設定使用者自訂警報訊息:	48

4.3.20.	取得使用者自訂警報訊息：	48
4.4.	輸出輸入命令	49
4.4.1.	取得輸入狀態：	49
4.4.2.	設定輸入模擬：	49
4.4.3.	設定輸入狀態：	49
4.4.4.	取得輸入模擬：	50
4.4.5.	取得輸出狀態：	50
4.4.6.	設定輸出狀態：	50
4.4.7.	設定輸入註解：	51
4.4.8.	設定輸出註解：	51
4.4.9.	取得輸入註解：	51
4.4.10.	取得輸出註解：	51
4.4.11.	取得機器人輸入：	52
4.4.12.	取得機器人輸出：	52
4.4.13.	設定機器人輸出：	53
4.4.14.	取得電磁閥輸出：	53
4.4.15.	設定電磁閥輸出：	53
4.4.16.	取得功能輸入狀態：	54
4.4.17.	取得功能輸出狀態：	54
4.4.18.	取得模組輸入設置：	55
4.4.19.	取得模組輸出設置：	55
4.4.20.	設定模組輸入模擬：	56
4.4.21.	設定模組輸入：	56
4.4.22.	設定模組輸入開始編號：	56
4.4.23.	設定模組輸入結束編號：	57
4.4.24.	設定模組輸入註解：	57
4.4.25.	設定模組輸出：	57
4.4.26.	設定模組輸出開始編號：	57
4.4.27.	設定模組輸出結束編號：	57
4.4.28.	設定模組輸出註解：	58
4.4.29.	設定模組輸入型態：	58
4.4.30.	設定模組輸出型態：	58
4.4.31.	保存模組設定：	58
4.3.32.	運動中進行 DO 開關操作：	59
4.5.	座標命令	60
4.5.1.	設定基底號碼：	60
4.5.2.	取得基底號碼：	61
4.5.3.	定義基底座標：	61

4.5.4.	取得基底座標：	61
4.5.5.	設定工具號碼：	62
4.5.6.	取得工具號碼：	62
4.5.7.	定義工具座標：	62
4.5.8.	取得工具座標：	63
4.6.	任務命令	63
4.6.1.	設定 RSR：	63
4.6.2.	取得 RSR 程式名稱：	64
4.6.3.	移除 RSR：	64
4.6.4.	RSR/PNS 啟動外部觸發任務：	65
4.6.5.	啟動任務：	65
4.6.6.	暫停任務：	66
4.6.7.	繼續任務：	67
4.6.8.	停止任務：	67
4.6.9.	取得目前執行任務名稱：	69
4.7.	檔案管理命令	69
4.7.1.	下載 HIWIM Robot language 檔案	69
4.7.2.	上傳 HIWIN Robot language 檔案	70
4.7.3.	刪除 Robot 運動檔案	70
4.7.4.	刪除 Robot 運動檔案資料夾	70
4.7.5.	新增 Robot 運動檔案資料夾	71
4.7.6.	重新命名 Robot 運動檔案	71
4.7.7.	拖動 Robot 運動檔案	71
4.7.8.	取得檔案數量	72
4.7.9.	取得檔案名稱	72
4.8.	控制器命令	72
4.8.1.	取得 HRSS 當前模式：	73
4.8.2.	伺服設定：	73
4.8.3.	取得伺服狀態：	73
4.8.4.	設定操作模式：	74
4.8.5.	取得操作模式：	74
4.8.6.	錯誤清除：	75
4.8.7.	更新 HRSS：	75
4.9.	吋動	75
4.9.1.	吋動：	75
4.9.2.	吋動復歸：	76
4.9.3.	吋動停止：	76
4.10.	運動命令	76

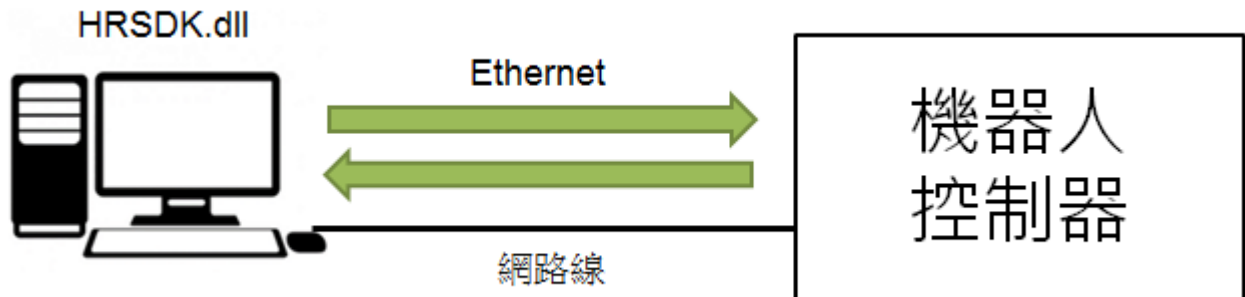
4.10.1.	絕對座標位置點對點運動：	77
4.10.2.	絕對關節角度點對點運動：	77
4.10.3.	相對座標位置點對點運動：	77
4.10.4.	相對關節角度點對點運動：	78
4.10.5.	位置暫存器之點對點運動：	78
4.10.6.	絕對座標位置直線運動：	79
4.10.7.	絕對關節角度直線運動：	79
4.10.8.	相對座標位置直線運動：	80
4.10.9.	相對關節角度直線運動：	80
4.10.10.	位置暫存器之直線運動：	81
4.10.11.	絕對座標位置圓弧運動：	82
4.10.12.	關節座標位置圓弧運動：	82
4.10.13.	位置暫存器之圓弧運動：	83
4.10.14.	運動暫停：	84
4.10.15.	運動繼續：	84
4.10.16.	運動停止：	84
4.10.17.	運動延遲：	85
4.10.18.	設定運動命令編號：	86
4.10.19.	取得運動命令編號：	86
4.10.20.	取得運動命令佇列中命令個數：	87
4.10.21.	取得目前運動狀態：	87
4.10.22.	移除運動命令佇列中單一命令：	88
4.10.23.	移除運動命令佇列中最新數個命令：	88
4.11.	手臂資訊命令	88
4.11.1.	取得目前編碼器數值：	88
4.11.2.	取得目前關節座標：	89
4.11.3.	取得目前絕對座標：	89
4.11.4.	取得目前轉速：	89
4.11.5.	取得裝置出廠時間：	90
4.11.6.	取得控制器開機時間：	90
4.11.7.	取得各軸馬達里程數：	90
4.11.8.	取得各軸馬達累積里程數：	91
4.11.9.	取得累積稼動率：	91
4.11.10.	取得稼動率百分比：	92
4.11.11.	取得馬達負載百分比：	92
4.11.12.	取得 HRSS 版本號：	92
4.11.13.	取得機器人型號：	93
4.11.14.	取得負載設定值：	93

4.11.15.	設定復歸位置：	93
4.11.16.	取得復歸位置：	94
4.11.17.	取得上次關機位置：	94
4.11.18.	啟動關節座標軟體極限：	94
4.11.19.	啟動卡式座標軟體極限：	95
4.11.20.	設定關節座標上下限：	95
4.11.21.	設定卡式座標上下限：	95
4.11.22.	取得關節座標軟體極限設置：	96
4.11.23.	取得卡式座標軟體極限設置：	96
4.12.	通訊設定	97
4.12.1.	設定顯示訊息狀態：	97
4.12.2.	取得顯示訊息狀態：	97
4.12.3.	網路連線：	97
4.12.4.	網路中斷連線：	97
4.12.5.	傳送網路訊息：	98
4.12.6.	接收網路訊息：	98
4.12.7.	設定網路配置：	98
4.12.8.	取得網路配置：	99
4.12.9.	更改網路 IP 位址：	100
4.12.10.	取得連線狀態：	100
5.	錯誤代碼	102
6.	附錄	103
6.1.	HRSDK 2.1.8 修改項目	103
6.2.	HRSDK 2.1.8 新增項目	104
6.3.	HRSDK 2.1.9 新增項目	105
6.4.	HRSDK 2.1.10 新增項目	106
6.5.	HRSDK 2.1.11 新增項目	107
6.6.	HRSDK 2.2.3 新增項目	108
6.7.	HRSDK 2.2.4 修改項目	109

HRSDK 版本	日期	適用軟體	註記
1.0.0	2017/02/06	HRSS V2.1.23 以上 HRSS V3.1.6 以上	初版發行
2.0.0	2017/07/13	HRSS V3.2.0 HRSS V3.2.1	新增控制函式
2.1.1	2017/09/11	HRSS V3.2.2	新增認證
2.1.2	2018/01/05	HRSS V3.2.5	新增產品介紹
2.1.4	2018/02/14	HRSS V3.2.5	修改範例程式
2.1.5	2018/03/09	HRSS V3.2.8	新增指令
2.1.6	2018/07/18	HRSS V3.2.11	新增章節 1.3，1.4 和 2.6
2.1.7	2018/09/27	HRSS V3.2.13	新增指令
2.1.8	2019/05/15	HRSS V3.2.15 以上	參考附錄 6.1, 6.2
2.1.9	2019/09/09	HRSS V3.3.1	新增指令，參考附錄 6.3
2.1.10	2019/09/09	HRSS V3.3.1	新增指令，參考附錄 6.4
2.1.11	2019/09/09	HRSS V3.3.1	新增指令，參考附錄 6.5
2.2.0	2020/01/01	HRSS V3.3.2	連線穩定，新增指令
2.2.2	2020/02/27	HRSS V3.3.3	新增資料回傳
2.2.3	2020/03/20	HRSS V3.3.3	新增指令，參考附錄 6.6
2.2.4	2020/04/20	HRSS V3.3.3	新增修改指令，參考附錄 6.7

1. 產品介紹

1.1 功能



使用者可以自行撰寫程式來控制機器人，達成系統整合的目的。

1.2 需求

■ 硬體需求：

- 上銀機器人
- 個人電腦
- 網路線

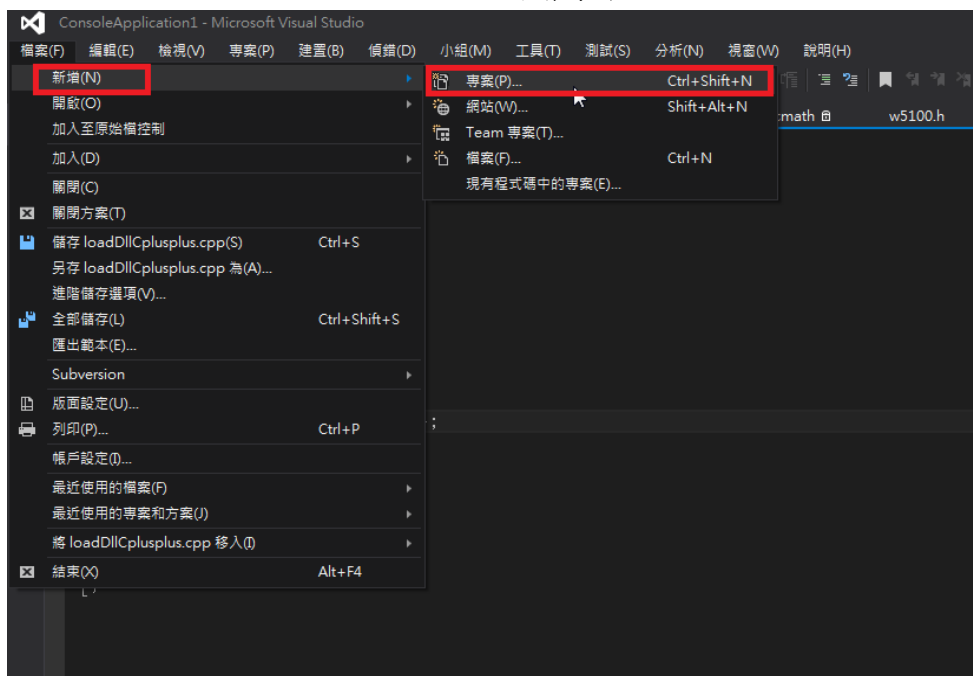
■ 軟體需求：

- 上銀機器人系統軟體：HRSS V3.2.5 以上。
- 程式整合開發環境，支援 C++, C#, VB

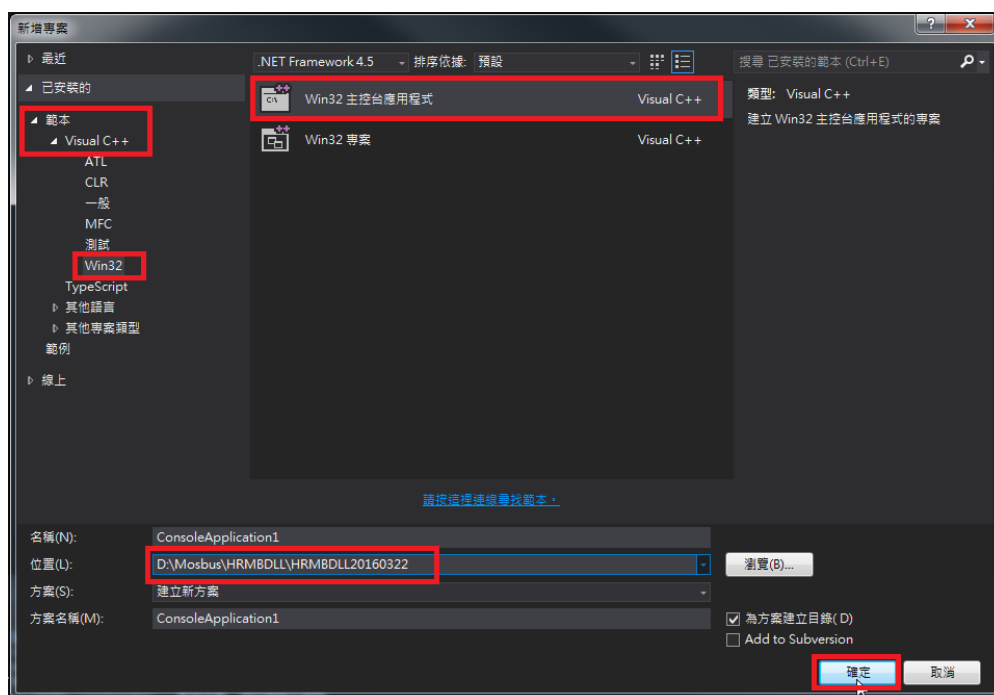
2. 使用方法

2.1 C++

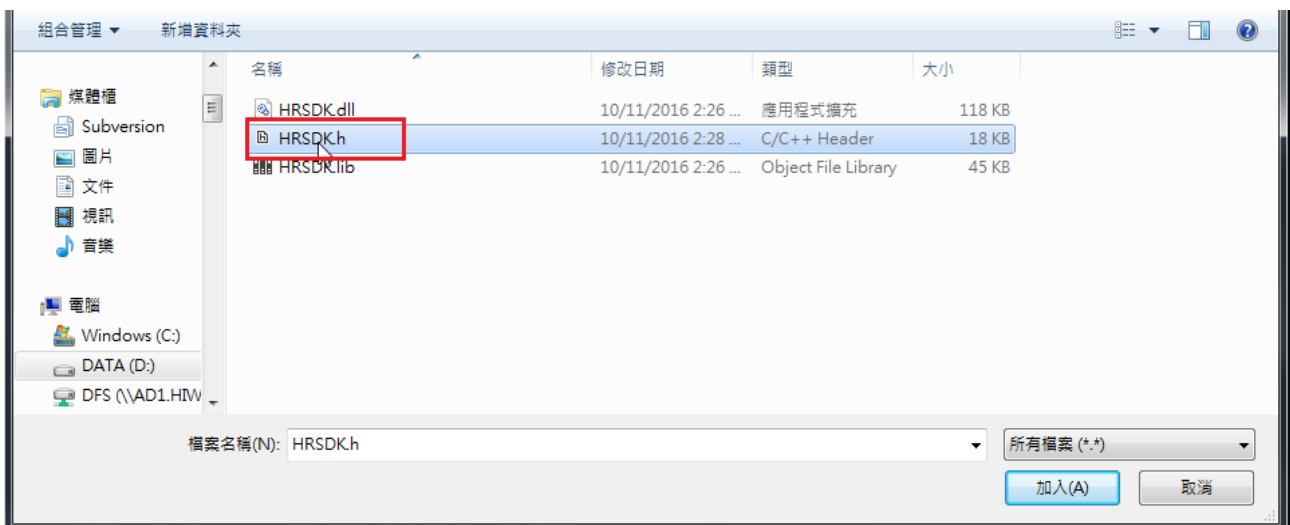
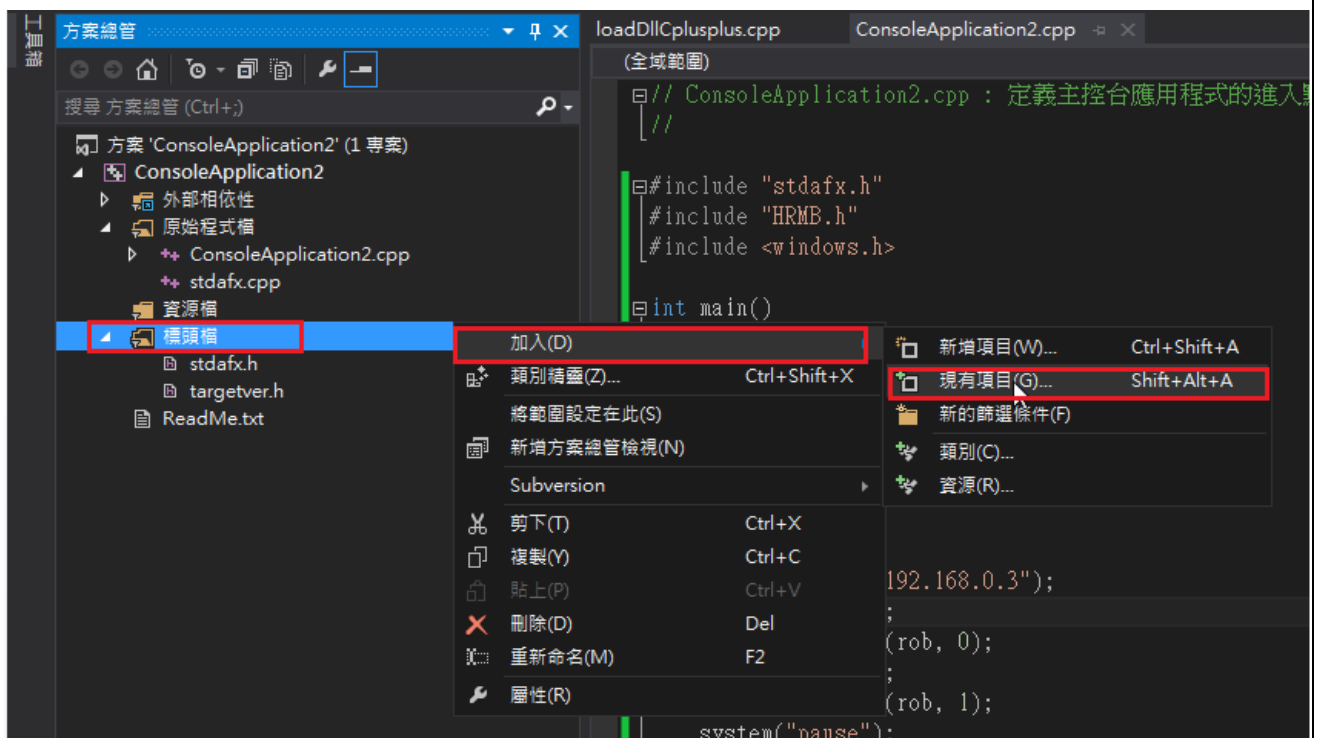
STEP 1：新增專案



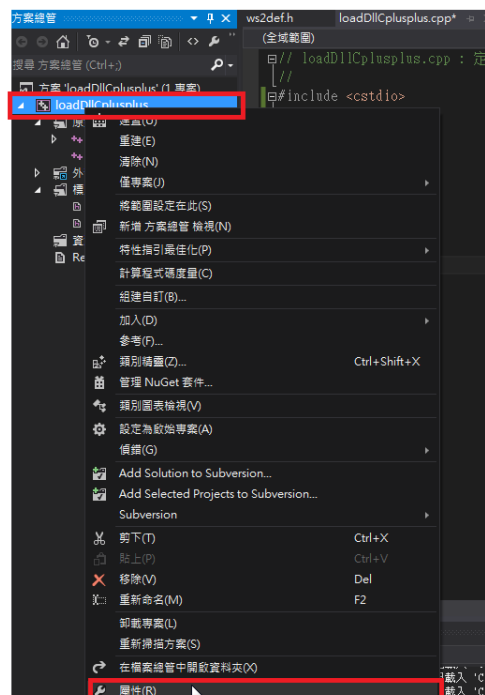
STEP 2：選擇 Visual C++ Win32 主控台應用程式並設定路徑



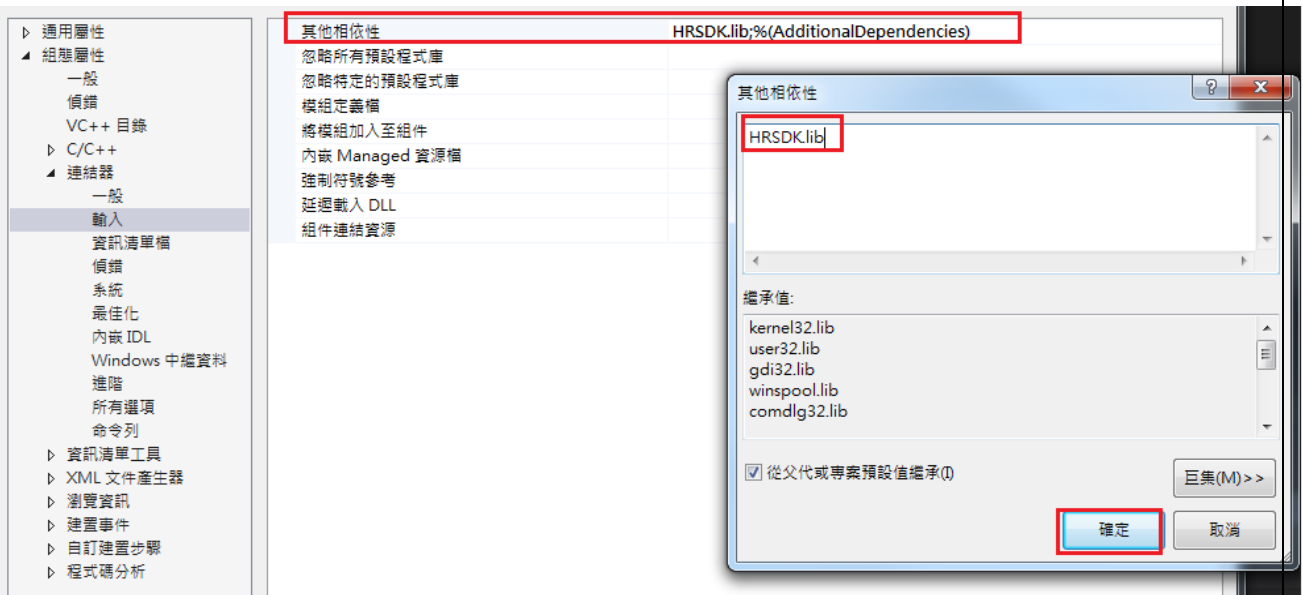
STEP 3：將 HRSDK.h 加入到標頭檔



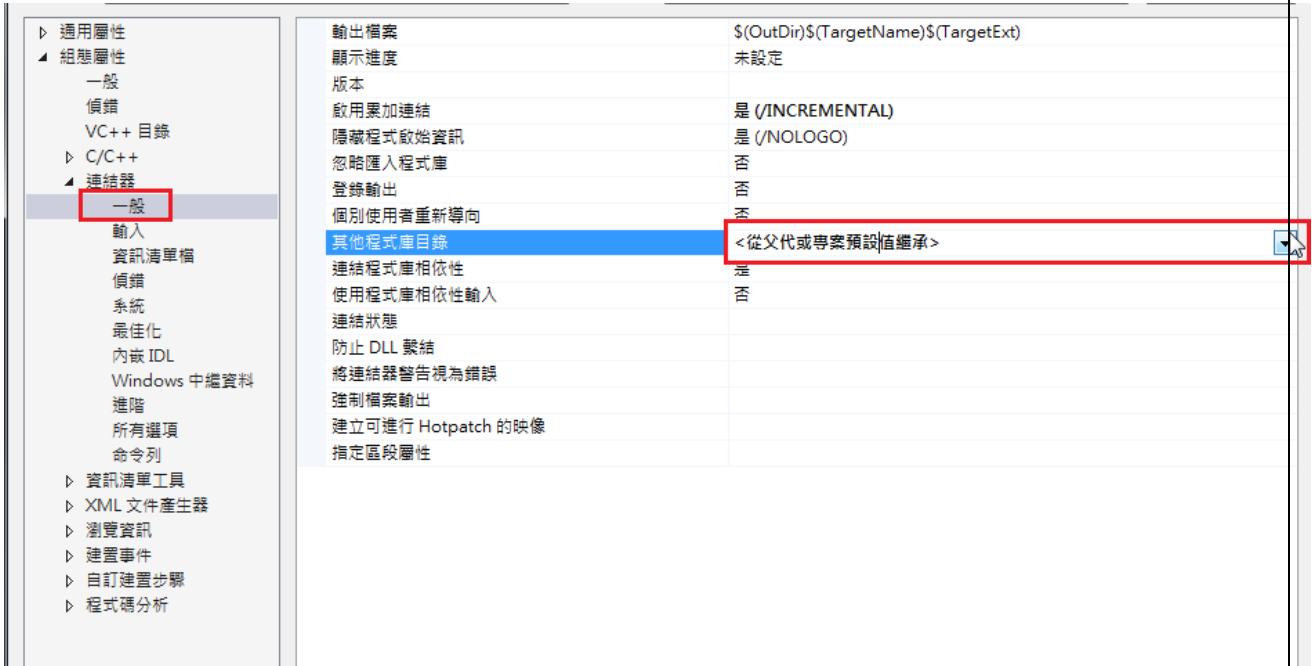
STEP 4：對專案右鍵選擇「屬性」



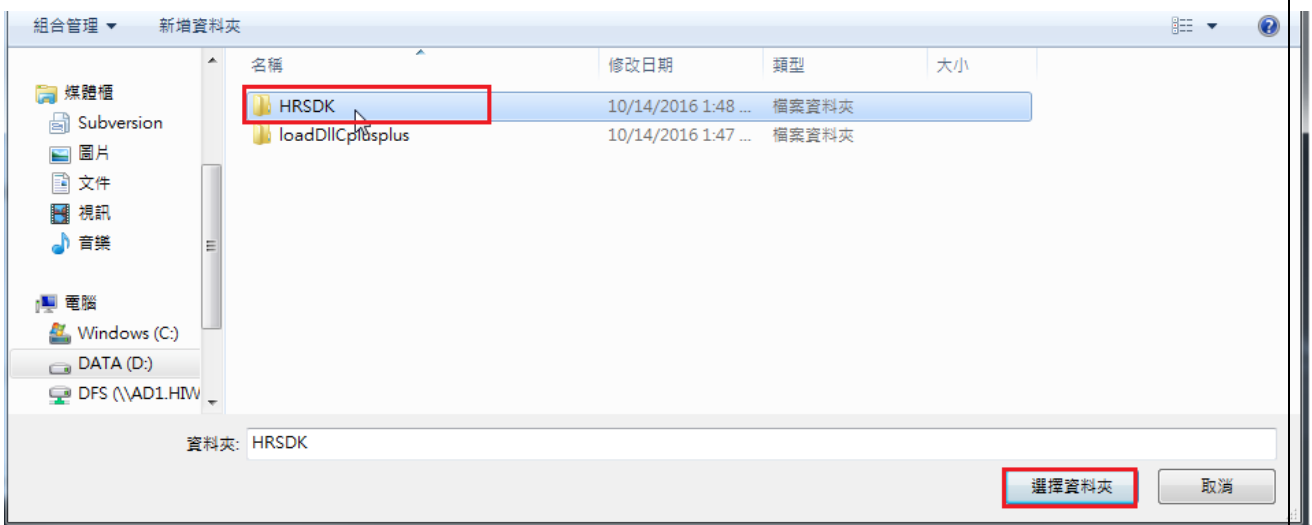
STEP 5：選擇連結器 -> 其他相依性 -> 輸入「HRSDK.lib」-> 確定



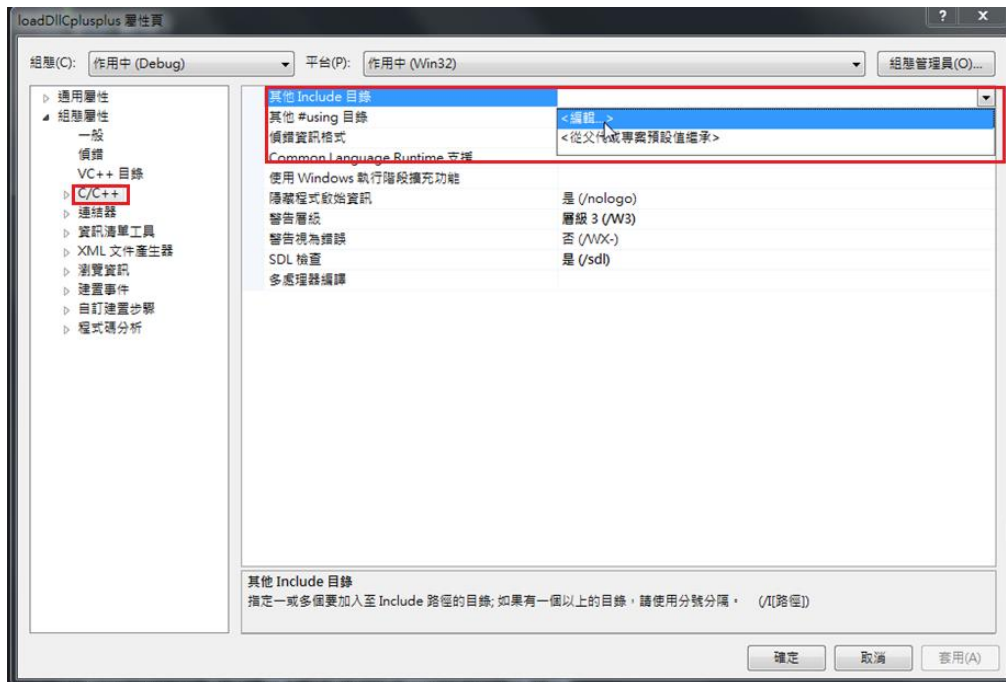
STEP 6：連結器 -> 一般 -> 其他程式庫目錄 -> 新增



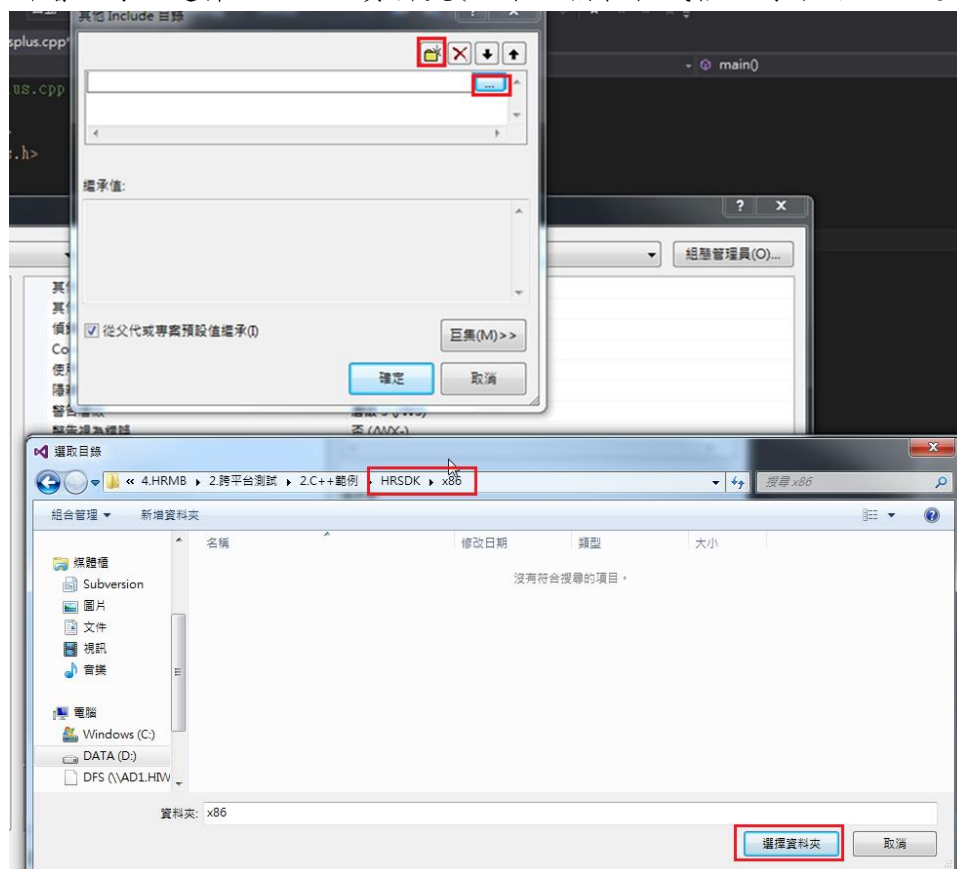
STEP 7：選擇 dll 所在的資料夾(如果 dll 在程式的根目錄下可以跳過 Step 6、7)



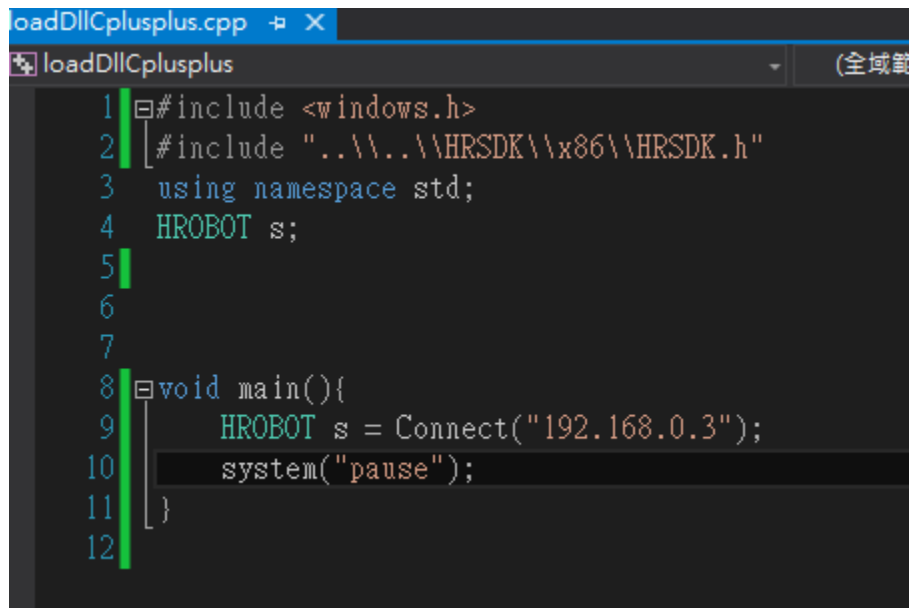
STEP 8 : C/C++ -> 其他 include 目錄 -> 編輯



STEP 9 : 新增目錄->選擇 HRSDK 資料夾(如果.h 檔在程式根目錄下可以跳過 Step 8、9)



STEP 10 : include HRSDK.h 標頭檔 -> 開始使用 dll



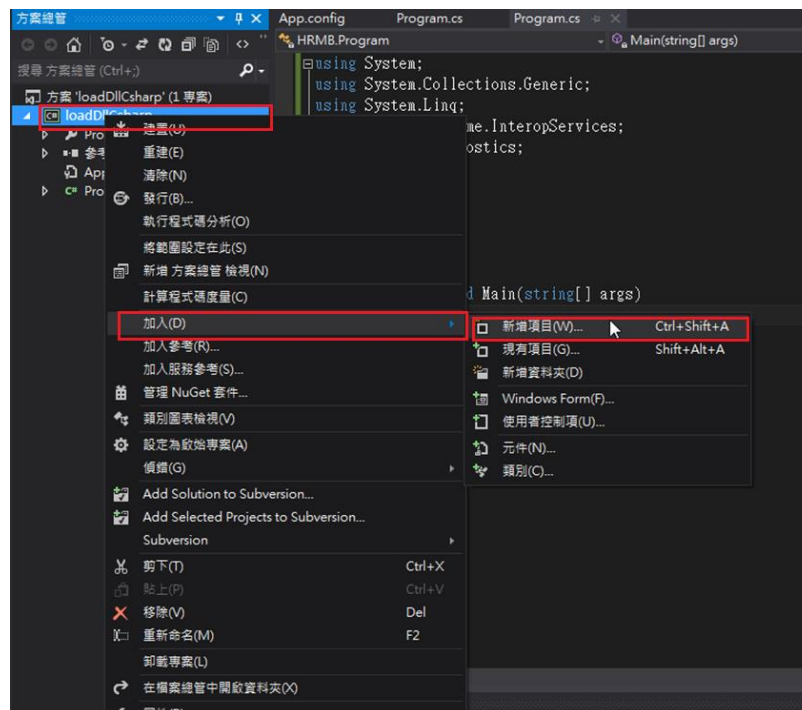
```

1  #include <windows.h>
2  #include "..\..\HRSDK\x86\HRSDK.h"
3  using namespace std;
4  HROBOT s;
5
6
7
8  void main(){
9      HROBOT s = Connect("192.168.0.3");
10     system("pause");
11 }
12

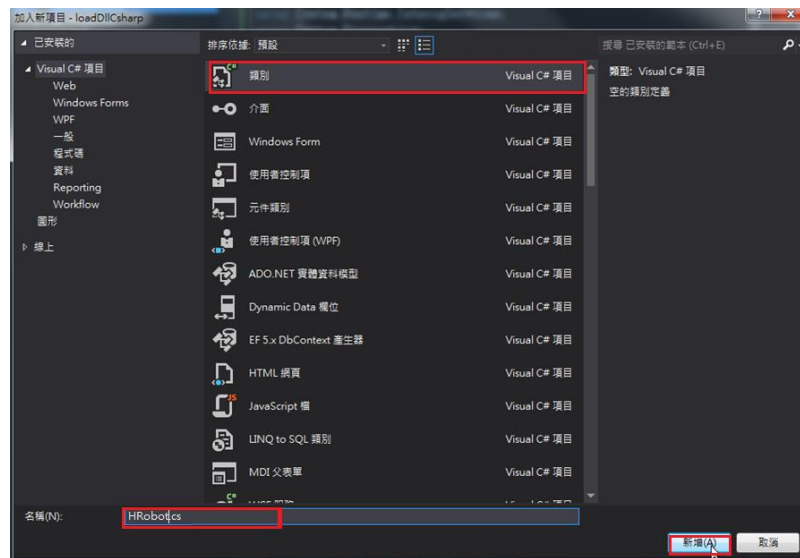
```

2.2 C#

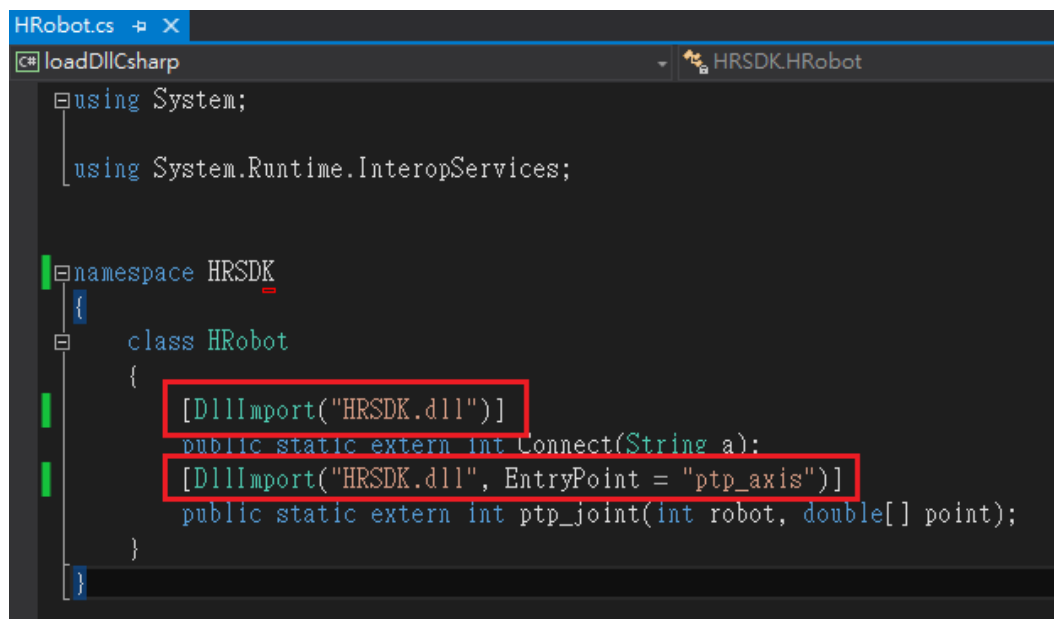
STEP 1: 對專案點右鍵 -> 加入 -> 新增項目



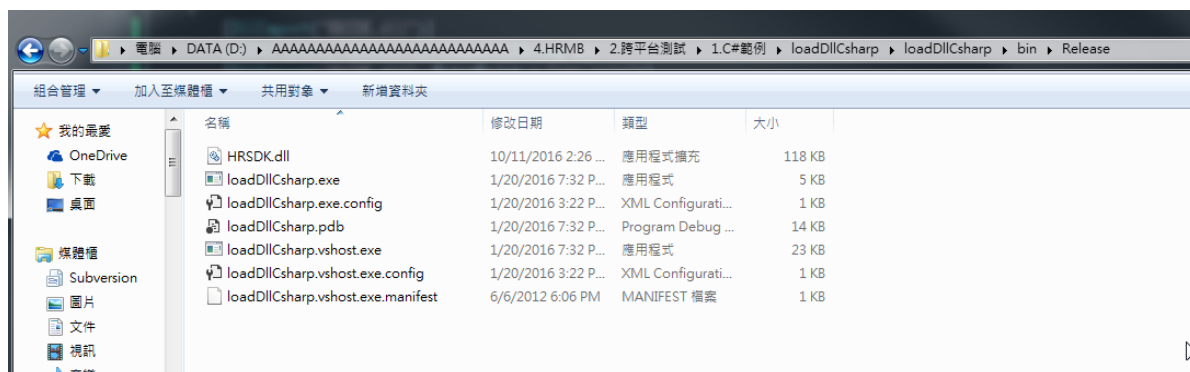
STEP 2: 取好名字 -> 新增



STEP 3: 輸入所需要的函式。可以對函式更改使用名稱。 EntryPoint 名稱要與 dll 內名稱一致。



STEP 4：將 HRSDK.dll 放進 專案/bin/Debug(Release) 路徑



STEP 5：開始使用 dll。

```

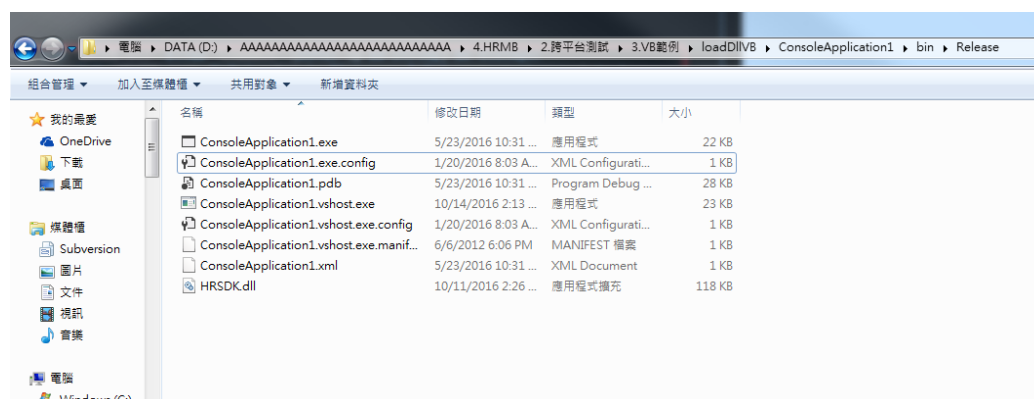
Program.cs  HRobot.cs
loadDllCsharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices;
using System.Diagnostics;
using System.IO;

namespace HRSDK
{
    class Program
    {
        static void Main(string[] args)
        {
            int rol=HRobot.Connect("192.168.0.3");
            double[] point={0,0,0,0,-90,0};
            Console.ReadLine();
        }
    }
}

```

2.3 VB

STEP 1：將 HRSDK.dll 放進 專案/bin/Debug(Release) 路徑



STEP 2：宣告好要使用的函數之後，即可以開始使用 HRSDK.dll

```
Module1.vb
Module1
Imports System.Text
Imports System.Runtime.InteropServices

Module Module1
    Public Declare Function StartRobot Lib "HRSDK.dll" Alias "Connect" (ByVal ip As String) As UInteger
    Public Declare Function set_motor_state Lib "HRSDK.dll" (ByVal robot As UInteger, ByVal state As Integer) As Integer
    Public Declare Function ptp_axis Lib "HRSDK.dll" (ByVal robot As UInteger, ByVal point As Double()) As Integer
    Public Declare Ansi Function set_mode Lib "HRSDK.dll" (ByVal robot As UInteger, ByVal p2 As Integer) As Integer

    Sub Main()
        Dim Name As String
        Dim stringBuilder As String
        stringBuilder = "127.0.0.1"

        Dim RobotID As UInteger
        RobotID = StartRobot(stringBuilder)
        Dim targetpoint As Double() = {0, 0, 0, 0, -90, 0}
        ptp_axis(RobotID, targetpoint)
        Console.WriteLine("Continue.....")
        Console.ReadLine()
    End Sub
End Module
```

2.4 連線等級說明

- 連線等級
- 觀察者：只能使用部分應用程式介面，能在 HRSS 有運行的情況且連線數小於最大容許數量 10 個時連線。
- 操作者：能使用所有應用程式介面，需在 HRSS 運行模式設定成 EXT 模式時連線。於 HRSS 中切換模式會導致連線的等級變為操作者等級。

2.5 操作模式說明

➤ 手動模式功能

- (1). 適用於測試運行，編程和示教
- (2). 線性運動速度會被限制在 250mm/s。
- (3). 點對點運動速度比例會依機型而限制
- (4). 每次開啟或關閉時會將整體速度設定成 10%，線性運動速度會設定成 250mm/s。
- (5). 可使用 set_operation_mode 指令切換為手動模式。
- (6). 允許吋動操作

➤ 手動模式說明

手動模式用於調試。調試包括設置，安裝，調整，修改或者故障排除時，手臂應開啟安全限速功能。在安全限速功能下，應注意以下事項：

- (1). 新的或修改的程序必須在開啟安全功能環境下測試。
- (2). 不允許工具，機械手臂或外部軸接觸或延伸到防護區的防護柵欄上。
- (3). 如果工件，工具或任何部件卡住或掉落，有任何機械故障或電器短路，則不允許啟動機器手臂。
- (4). 所有調試工作必須在保護/圍欄/保護區外進行。

➤ 自動模式說明

自動模式功能必須符合以下安全保護措施：

- (1). 所有保護措施必須經過測試且安裝。
- (2). 所有暫停的保護措施應恢復其功能。
- (3). 不允許人員留在操作區。
- (4). 一定要遵守標準操作規定。
- (5). 如果機器人因未知原因停止，則只有在繳活緊急停止按鈕後才允許訪問危險區域。
- (6). 使用 SDK 的 disconnect 可以讓自動模式繼續動作，否則異常斷線會停下當前運動。

3. 命令清單

3.1. 連線命令

函 式 名 稱	說 明	操 作 者		觀 察 者
		手動 模式	自動 模式	
open_connection	與裝置連線	○	○	○
disconnect	關閉與裝置的連線	○	○	○
set_connection_level	設定連線等級	○	○	○
get_connection_level	取得連線等級	○	○	○
get_hrsdk_version	取得 HRSDK 版本號	○	○	○

3.2. 暫存器命令

函 式 名 稱	說 明	操 作 者		觀 察 者
		手動 模式	自動 模式	
set_timer	設定機器人計時器	○	○	×
get_timer	取得機器人計時器	○	○	○
set_timer_start	啟動機器人計時器	○	○	×
set_timer_stop	停止機器人計時器	○	○	×
get_timer_status	取得機器人計時器狀態	○	○	○
set_timer_name	設定機器人計時器名稱	○	○	×
get_timer_name	取得機器人計時器名稱	○	○	○
set_counter	設定機器人計數器	○	○	○
get_counter	取得機器人計數器	○	○	○
get_counter_name	取得機器人計數器名稱	○	○	○
set_counter_name	設定機器人計數器名稱	○	○	×
set_pr_type	設定位置暫存器座標型態	○	○	×
get_pr_type	取得位置暫存器座標型態	○	○	○
set_pr_coordinate	設定位置暫存器座標值	○	○	×
get_pr_coordinate	取得位置暫存器座標值	○	○	○
set_pr_tool_base	設定位置暫存器工具基底編號	○	○	×
get_pr_tool_base	取得位置暫存器工具基底編號	○	○	○

set_pr	設定位置暫存器值	○	○	×
get_pr	取得位置暫存器值	○	○	○
remove_pr	清空位置暫存器值	○	○	×
set_pr_comment	設定位置暫存器註解	○	○	×
get_pr_comment	取得位置暫存器註解	○	○	○

3.3. 系統變數命令

函 式 名 稱	說 明	操 作 者		觀 察 者
		手動 模式	自動 模式	
set_acc_dec_ratio	設定加速度比例	×	○	×
get_acc_dec_ratio	取得加速度比例	○	○	○
set_acc_time	設定加速度時間	×	○	×
get_acc_time	取得加速度時間	○	○	○
set_ptp_speed	設定點對點移動速度	×	○	×
get_ptp_speed	取得點對點移動速度	○	○	○
set_lin_speed	設定取得直線移動速度	×	○	×
get_lin_speed	取得直線移動速度	○	○	○
set_override_ratio	設定進給速度倍率	○	○	×
get_override_ratio	取得進給速度倍率	○	○	○
set_robot_id	設定機器人識別名稱	○	○	×
get_robot_id	取得機器人識別名稱	○	○	○
set_smooth_length	設定運動平滑半徑	×	○	×
get_alarm_code	取得警報代碼	○	○	○
set_digital_setting	設定數位設置	○	○	×
get_digital_setting	取得數位設置	○	○	○
set_language	設定語言	○	○	○
get_controller_time	取得控制器現在時間	○	○	○
set_user_alarm_setting_message	設定使用者自訂警報訊息	○	○	×
get_user_alarm_setting_message	取得使用者自訂警報訊息	○	○	○

3.4. 輸入輸出命令

函 式 名 稱	說 明	操 作 者		觀 察 者
		手 動 模 式	自 動 模 式	
get_digital_input	取得輸入	○	○	○
get_digital_output	取得輸出	○	○	○
set_digital_output	設定輸出	○	○	×
set_DI_simulation_Enable	設定輸入模擬	○	○	×
set_DI_simulation	設定輸入狀態	○	○	×
get_DI_simulation_Enable	取得輸入模擬	○	○	○
set_digital_input_comment	設定輸入註解	○	○	×
get_digital_input_comment	取得輸入註解	○	○	○
set_digital_output_comment	設定輸出註解	○	○	×
get_digital_output_comment	取得輸出註解	○	○	○
get_robot_input	取得機器人輸入	○	○	○
get_robot_output	取的機器人輸出	○	○	○
set_robot_output	設定機器人輸出	○	○	×
get_valve_output	取得電磁閥輸出	○	○	○
set_valve_output	設定電磁閥輸出	○	○	×
get_function_input	取得功能輸入	○	○	○
get_function_output	取得功能輸出	○	○	○
get_module_input_config	取得模組輸入設置	○	○	○
get_module_output_config	取得模組輸出設置	○	○	○
set_module_input_simulation	設定模組輸入模擬	○	○	×
set_module_input_value	設定模組輸入	○	○	×
set_module_input_start	設定模組輸入開始編號	○	○	×
set_module_input_end	設定模組輸入結束編號	○	○	×
set_module_input_comment	設定模組輸入註解	○	○	×

set_module_output_value	設定模組輸出	○	○	×
set_module_output_start	設定模組輸出開始編號	○	○	×
set_module_output_end	設定模組輸出結束編號	○	○	×
set_module_output_comment	設定模組輸出註解	○	○	×
set_module_input_type	設定模組輸入型態	○	○	×
set_module_output_type	設定模組輸出型態	○	○	×
save_module_io_setting	保存模組設定	○	○	×
SyncOutput	運動中進行 DO 開關操作	○	○	×

3.5. 座標系命令

函 式 名 稱	說 明	操作者		觀察者
		手動 模式	自動 模式	
set_base_number	設定基底號碼	○	○	×
get_base_number	取得基底號碼	○	○	○
define_base	定義基底座標	○	×	×
get_base_data	取得基底座標	○	○	○
set_tool_number	設定工具號碼	○	○	×
get_tool_number	取得工具號碼	○	○	○
define_tool	定義工具座標	○	×	×
get_tool_data	取得工具座標	○	○	○

3.6. 任務命令

函 式 名 稱	說 明	操作者		觀察者
		手動 模式	自動 模式	
set_rsr	設定 RSR	○	×	×
get_rsr_prog_name	取得 RSR 程式名稱	○	○	○
remove_rsr	移除 RSR	○	×	×
ext_task_start	RSR/PNS 啟動外部觸發任	○	○	×

	務			
task_start	啟動任務	○	○	×
task_hold	暫停當前任務	○	○	×
task_continue	繼續當前任務	○	○	×
task_abort	停止當前任務	○	○	×
get_execute_file_name	取得當前執行中程式檔案名稱	○	○	○

3.7. 檔案管理命令

函 式 名 稱	說 明	操作者		觀察者
		手動 模式	自動 模式	
send_file	上傳 HRB 至控制器	○	×	×
download_file	下載 HRB 至本地端	○	○	×
delete_file	刪除 Robot 運動檔案	○	○	×
delete_folder	刪除 Robot 運動檔案資料夾	○	○	×
new_folder	新增 Robot 運動檔案資料夾	○	○	×
file_rename	重新命名 Robot 運動檔案	○	○	×
file_drag	拖動 Robot 運動檔案	○	○	×
get_prog_number	取得檔案數量	○	○	○
get_prog_name	取得檔案名稱	○	○	○

3.8. 控制器設定命令

函 式 名 稱	說 明	操作者		觀察者
		手動 模式	自動 模式	
get_hrss_mode	取得 HRSS 當前模式	○	○	○
set_motor_state	伺服設定	○	○	×
get_motor_state	取得伺服狀態	○	○	○
set_operation_mode	設定操作模式	○	○	×
get_operation_mode	取得操作模式	○	○	○
clear_alarm	錯誤清除	○	×	×
update_hrss	更新 HRSS 版本	○	×	×

3.9. 吋動

函 式 名 稱	說 明	操 作 者		觀 察 者
		手 動 模 式	自 動 模 式	
jog	吋動	○	×	×
jog_stop	吋動停止	○	×	×
jog_home	吋動復歸	○	×	×

3.10.運動命令

函 式 名 稱	說 明	操 作 者		觀 察 者
		手 動 模 式	自 動 模 式	
ptp_pos	絕對座標位置點對點運動	○	○	×
ptp_axis	絕對關節角度點對點運動	○	○	×
ptp_rel_pos	相對座標位置點對點運動	○	○	×
ptp_rel_axis	相對關節角度點對點運動	○	○	×
ptp_pr	位置暫存器之點對點運動	○	○	×
lin_pos	絕對座標位置直線運動	○	○	×
lin_axis	絕對關節角度直線運動	○	○	×
lin_rel_pos	相對座標位置直線運動	○	○	×
lin_rel_axis	相對關節角度直線運動	○	○	×
lin_pr	位置暫存器之直線運動	○	○	×
circ_axis	絕對關節角度圓弧運動	○	○	×
circ_pos	絕對座標位置圓弧運動	○	○	×
circ_pr	位置暫存器之圓弧運動	○	○	×
motion_hold	運動暫停	○	○	×
motion_continue	運動繼續	○	○	×
motion_abort	運動停止	○	○	×
motion_delay	運動延遲	○	○	×
set_command_id	設定運動命令編號	○	○	×
get_command_id	取得目前運動命令編號	○	○	○
get_command_count	取得目前運動命令佇列中命令個數	○	○	○
get_motion_state	取得目前運動狀態	○	○	○
remove_command	移除運動命令佇列中命令	○	○	×
remove_command_tail	移除運動命令佇列中數個命令	○	○	×

3.11.手臂資訊命令

函 式 名 稱	說 明	操 作 者		觀 察 者
		手動 模式	自動 模式	
get_encoder_count	取得目前編碼器數值	○	○	○
get_current_joint	取得目前關節座標	○	○	○
get_current_position	取得目前絕對座標	○	○	○
get_current_rpm	取得目前各軸轉速	○	○	○
get_device_born_date	取得裝置出廠時間	○	○	○
get_operation_time	取得控制器開機時間	○	○	○
get_mileage	取得各軸馬達里程數	○	○	○
get_total_mileage	取得各軸馬達累積里程數	○	○	○
get_utilization	取得累積稼動率	○	○	○
get_utilization_ratio	取得稼動率百分比	○	○	○
get_motor_torque	取得馬達負載百分比	○	○	○
get_hrss_version	取得 HRSS 版本號	○	○	○
get_robot_type	取得機器人型號	○	○	○
get_payload_value	取得負載設定值	○	○	○
set_home_point	設定復歸位置	○	○	×
get_home_point	取得復歸位置	○	○	○
get_previous_pos	取得上次關機位置	○	○	○
enable_joint_soft_limit	啟動關節座標軟體極限	○	○	×
enable_cart_soft_limit	啟動卡式座標軟體極限	○	○	×
set_joint_soft_limit	設定關節座標上下限	○	○	×
set_cart_soft_limit	設定卡式座標上下限	○	○	×
get_joint_soft_limit_config	取得關節座標軟體極限設置	○	○	○
get_cart_soft_limit_config	取得卡式座標軟體極限設置	○	○	○

3.12.通訊設定

函 式 名 稱	說 明	操 作 者		觀 察 者
		手動 模式	自動 模式	
set_network_show_msg	設定顯示訊息狀態	O	O	×
get_network_show_msg	取得顯示訊息狀態	O	O	O
network_connect	網路連線	O	O	×
network_disconnect	網路中斷連線	O	O	×
network_send_msg	傳送網路訊息	O	O	×
network_recieve_msg	接收網路訊息	O	O	O
set_network_config	設定網路配置	O	O	×
get_network_config	取得網路配置	O	O	O
network_change_ip	更改網路 IP 位址	O	O	×
network_get_state	取得連線狀態	O	O	O

4. 命令參考

4.1. 連線命令

4.1.1. 建立連線：

HROBOT open_connection(**const char*** address, **int** level, **callback_function** function)

參數	資料型態	說明
address	const char*	裝置的 IP 位址
level	int	連線的等級 0: 操作者 1: 專家
function	void __stdcall FunName(uint16_t uint16_t uint16_t* int);	Event 接收的函式，用以接收控制器回傳的訊息。
回傳值	HROBOT	成功：裝置 id (0-65535 為有效裝置 id) 失敗：-1

- 當控制器收到第一個操作者連線請求時，控制器會為手動模式。後續的連線都會為觀察者。

4.1.2. 關閉連線：

void disconnect (**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	欲中斷的裝置的 ID

C++:

```
void __stdcall FuncName(uint16_t uint16_t uint16_t* int);
void main(){
    HROBOT robot;
    robot=open_connection("192.168.0.3", 1 ,FuncName);
    //Do something
    disconnect (robot);
}
void __stdcall CallBackFun(uint16_t cmd, uint16_t rlt, uint16_t* msg, int len) {
    // process information from controller
}
```

4.1.3. 設定連線等級：

int set_connection_level(**HROBOT** robot, **int** mode)

參數	資料型態	說明
robot	HROBOT	裝置 ID
mode	int	連線模式 0: 觀察者 1: 操作者
回傳值	int	成功: 0 失敗: -1

C++:

```
int mode = 1;
set_connection_level(robot, mode);
```

4.1.4. 取得連線等級：

int get_connection_level(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置 ID
回傳值	int	0:觀察者 1:操作者

C++:

```
int level;
level = get_connection_level(robot);
```

4.1.5. 取得 HRSDK 版本號：

void get_hrsdk_version(**char*** version)

參數	資料型態	說明
version	char*	HRSDK 版本號

C++:

```
char* sdk_version = new char[256];
get_hrsdk_version(sdk_version);
std::cout << "HRSDK Version : " << sdk_version << std::endl;
delete[] sdk_version;
```

4.1.6. Callback

void callback_function(uint16_t cmd, uint16_t rlt, uint16_t* msg, **int** len)

參數	資料型態	說明
cmd	uint16_t	回傳指令的 Command number
rlt	uint16_t	回傳指令實作過程中的 Result
msg	uint16_t*	回傳指令所帶的字串
len	int	回傳字串長度
備註	需先執行 get_current_position 才會啟動機制	

cmd	rlt	說明
0	4030	發生 alarm 與錯誤訊息
0	4031	電池警告
0	4032	電池 alarm
0	4033	電池 normal
0	4144	System Input 數值改變
0	4145	System Output 數值改變
0	4702	當以下資訊改變，會回傳以下資訊

		hrss_mode operation_mode override_ratio motor_state exe_file_name FO alarm_count keep_alive motion_status payload speed position coor joint encoder	HRSS 模式 操作模式 整體速度 伺服狀態 執行檔案名稱 Function Output alarm 數量 保持連線 運動狀態 負載值 T1, T2 速度 座標改變 笛卡兒座標 軸座標 編碼值
0	4703	Timer 數值改變	
0	4704	Counter 數值改變	
0	4705	Module Input 數值改變	
0	4706	Module Output 數值改變	
0	4707	FieldBus Input 數值改變	
0	4708	FieldBus Output 數值改變	
0	4710	PR 改變，off_on, index, mode, position	
0	4711	Digital Input 數值改變	
0	4712	Digital Output 數值改變	
0	4714	Utilization start，當 task_start 開始執行	
0	4715	Utilization end，當 task_start 停止	
1450	4028	開始清除 alarm	
1450	4029	結束清除 alarm	
4000	0	執行 ext_task_start	
4000	4013	ext_task_start 已經在執行	
4001	2006	執行 task_start 前，已經有運動在執行	
4001	4011	task_start 檔案開啟錯誤	
4001	4012	task_start 檔名錯誤	
4001	4013	執行 task_start 前，已經有 task 在執行	
4001	4014	task_start 開始執行	
4004	4018	停止任務	
4009	0	下載檔案	

4009	201	下載檔案不存在
4010	0	上傳檔案
4011	4020	HRSS 上傳檔案錯誤
4011	4021	更新檔案傳送失敗
4011	4022	HRSS 更新檔案失敗
4011	4023	HRSS 硬體容量不足
4011	4026	HRSS 開始更新
4011	4027	HRSS 更新檔案傳送成功
4709	501	Module IO 儲存成功

C++:

```
void __stdcall callBack(uint16_t cmd, uint16_t rlt, uint16_t* msg, int len) {
    switch (cmd)
    {
    case 0:
        if (rlt == 4030) {
            // HRSS_ALARM_NOTIFY
        } else if (rlt == 4031) {
            // HRSS_BATTERY_WARRING
        } else if (rlt == 4032) {
            // HRSS_BATTERY_ALARM
        } else if (rlt == 4033) {
            // HRSS_BATTERY_NORMAL
        }
        break;
        /* Clear Alarm */
    case 1450:
        switch (rlt) {
        case 4028:
            // HRSS_START_CLEAR_ALARM
            break;
        case 4029:
            // HRSS_FINISH_CLEAR_ALARM
            break;
        default:
            break;
        }
    }
```

```
        break;
        /* Task Start */
case 4001:
    switch (rlt) {
    case 4011:
        // ERROR_OPEN_FILE
        break;
    case 4014:
        // HRSS_TASK_START_FINISH
        break;
    default:
        break;
    }
    break;
    /* Update HRSS */
case 4011:
    switch (rlt) {
    case 4020:
        // HRSS_UPDATE_FILE_ERROR
        break;
    case 4021:
        // HRSS_UPDATE+FILE_TRANSFER_ERROR
        break;
    case 4022:
        // HRSS_UPDATE_FILE_UNARCHIVER_ERROR
        break;
    case 4023:
        // HRSS_HARD_DISK_CAPACITY_IS_NOT_ENOUGH
        break;
    case 4026:
        // HRSS_START_UPDATE
        break;
    case 4027:
        // HRSS_UPDATE_FILE_TRANSFER_SUCCESS
        break;
    default:
        break;
    }
}
```



```

        break;
    default:
        break;
    }
}

```

4.2. 暫存器命令

4.2.1. 設定機器人計時器：

`int set_timer(HROBOT robot, int index, int value)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Timer 的編號 (1-20)
value	int	數值範圍(-2147483648~2147483647)
回傳值	int	成功: 0 失敗: -1

C++:

```
set_timer(robot,1,100) //set Timer number 1 with value 100
```

4.2.2. 取得機器人計時器:

`int get_timer(HROBOT robot, int index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Timer 的編號 (1-20)
回傳值	int	成功: Timer 的值 失敗: -1

C++:

```
set_timer(robot,1,100) //set Timer number 1 with value 100
get_timer(robot,1) //get value from Timer number 1
```

4.2.3. 啟動機器人計時器:

`int set_timer_start (HROBOT robot, int index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Timer 的編號 (1-20)
回傳值	int	成功: Timer 的值 失敗: -1

C++:

```
int timer_index;
set_timer_start(robot, timer_index);
```

4.2.4. 停止機器人計時器:

`int set_timer_stop(HROBOT robot, int index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Timer 的編號 (1-20)
回傳值	int	成功: Timer 的值 失敗: -1

C++:

```
int timer_index;
set_timer_stop(robot, timer_index);
```

4.2.5. 取得機器人計時器狀態:

`int get_timer_status(HROBOT robot, int index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Timer 的編號 (1-20)
回傳值	int	關閉: 0 開啟: 1 失敗: -1

C++:

```
int timer_index = 1;
set_timer_status(robot, timer_index);
```

4.2.6. 設定機器人計時器名稱:

int set_timer_name(**HROBOT** robot, **int** index, **char*** name)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Timer 的編號 (1-20)
name	char*	計時器名稱
回傳值	int	成功: 0 失敗: -1

C++:

```
int timer_index = 1;
char* name = "This is Timer1.";
set_timer_name(robot, timer_index, name);
```

4.2.7. 取得機器人計時器名稱:

int get_timer_name(**HROBOT** robot, **int** index, **char*** name)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Timer 的編號 (1-20)
name	char*	計時器名稱
回傳值	int	成功: 0 失敗: -1

C++:

```
int timer_index = 1;
char* name = "This is Timer1.";
set_timer_name(robot, timer_index, name);
get_timer_name(robot, timer_index, name);
```

4.2.8. 設定機器人計數器:

int set_counter(**HROBOT** robot, **int** index, **int** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Counter 的編號 (1-20)
value	int	數值範圍(-2147483648~2147483647)
回傳值	int	成功: 0 失敗: -1

C++:

```
set_counter(robot,1,100) //set Counter number 1 with value 100
```

4.2.9. 取得機器人計數器:

int get_counter(**HROBOT** robot, **int** index, **int** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Counter 的編號 (1-20)
value	int	數值範圍(-2147483648~2147483647)
回傳值	int	成功: 0 失敗: -1

4.2.10. 設定機器人計數器名稱:

int set_counter_name(**HROBOT** robot, **int** index, **char*** name)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Counter 的編號 (1-20)
name	char*	計數器名稱長度 200 以內
回傳值	int	成功: 0 失敗: -1

4.2.11. 取得機器人計數器名稱:

int get_counter_name(**HROBOT** robot, **int** index, **int** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Counter 的編號 (1-20)

name	char*	取得計數器名稱
回傳值	int	成功: 0 失敗: -1

C++:

```
int index = 5;
char* name = "This is Counter5";
set_counter_name(robot, index, name);
get_counter_name(robot, index, name);
```

4.2.12. 設定位置暫存器座標系種類：

int set_pr_type(HROBOT robot, int pr_num, int type)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
type	int	笛卡爾座標:0 關節座標:1
回傳值	int	成功: 0 失敗: 錯誤代碼

4.2.13. 取得位置暫存器座標系種類：

int get_pr_type(HROBOT robot, int pr_num)

參數	資料型態	說明
Robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
回傳值	int	笛卡爾座標:0 關節座標:1 失敗: -1

4.2.14. 設定位置暫存器座標：

int set_pr_coordinate(HROBOT robot, int pr_num, double* coor)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
coor	double[6]	欲設定的座標系陣列: 笛卡爾座標 {X,Y,Z,A,B,C}

		關節座標{A1,A2,A3,A4,A5,A6}
回傳值	int	成功: 0 失敗: 錯誤代碼

4.2.15. 取得位置暫存器座標：

int get_pr_coordinate(**HROBOT** robot, **int** pr_num, **double*** coor)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
coor	double[6]	回傳的座標振烈: 笛卡爾座標{X,Y,Z,A,B,C} 關節座標{A1,A2,A3,A4,A5,A6}
回傳值	int	成功: 0 失敗: -1

4.2.16. 設定位置暫存器工具、基底座標：

int set_pr_tool_base(**HROBOT** robot, **int** pr_num, **int** tool, **int** base)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
tool	int	工具號碼
base	int	基底號碼
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
(1)
double coor[6]={0,0,0,0,-90,0};
set_pr(robot,1,1,coor,2,2); //set address register
                                //address register number:1
                                //coordinate type:joint
                                //coordinate:{0,0,0,0,-90,0}
                                //tool number:2
                                //base number:2

(2)
set_pr_type(robot,1,1);
set_pr_coordinate(robot,1,coor);
set_pr_tool_base(robot,1,2,2);

(3)
int prType=get_pr_type(robot,1,1);    //get pr type
double coor[6]
get_pr_coordinate(robot,1,coor); //get coordinate from pr 1
int tool_base[2];
get_pr_tool_base(robot,1,tool_base); //get tool and base from pr 1
                                //tool:tool_base[0]
                                //base:tool_base[1]
```

- (1). 設定 pr 資訊
- (2). 效果同(1)
- (3). 取得位置暫存器的各項資訊
 - A. 座標型態
 - B. 關節座標
 - C. 工具編號
 - D. 基底編號

4.2.17. 取得位置暫存器工具、基底座標：

`int get_pr_tool_base(HROBOT robot, int pr_num, int* tool_base)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID

pr_num	Int	位置暫存器號碼(1-4000)
tool_base	int[2]	int[0]:工具號碼 int[1]:基底號碼
回傳值	int	成功: 0 失敗: -1

4.2.18. 設定位置暫存器資料：

`int set_pr(HROBOT robot, int pr_num, int coor_type, double* coor, int tool, int base)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
coor_type	int	座標系形態 笛卡爾座標:0 關節座標:1
coor	double[6]	欲設定的位置暫存器座標
tool	int	工具座標號碼
base	int	底座座標號碼
回傳值	int	成功: 0 失敗: -1

4.2.19. 取得位置暫存器值：

`int get_pr(HROBOT robot, int pr_num, int* coor_type, double* coor, int* tool, int* base)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
coor_type	int	座標系形態 笛卡爾座標:0 關節座標:1
coor	double[6]	欲設定的位置暫存器座標
tool	int	工具座標號碼
base	int	底座座標號碼
回傳值	int	成功: 0 失敗: -1

4.2.20. 清除位置暫存器值：

`int remove_pr(HROBOT robot, int pr_num)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
回傳值	int	成功: 0 失敗: -1

4.2.21. 取得位置暫存器註解：

`int get_pr_comment(HROBOT robot, int pr_num, char* comment)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
comment	char*	位置暫存器註解
回傳值	int	成功: 0 失敗: -1

4.2.22. 設定位置暫存器註解：

`int set_pr_comment(HROBOT robot, int pr_num, char* comment)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pr_num	int	位置暫存器號碼(1-4000)
comment	char*	位置暫存器註解
回傳值	int	成功: 0 失敗: -1

C++:

```
(1)
double coor[6]={0,0,0,0,-90,0};
set_pr(robot,2,1,coor,2,2); //set address register
                                //address register number:2
                                //coordinate type:joint
                                //coordinate:{0,0,0,0,-90,0}
                                //tool number:2
                                //base number:2

int coor_type=-1;
coor[6]={0}
int tool = -1;
int base= -1;
char* comment = "This is a comment.";
get_pr(robot,2 , &coor_type,coor,&tool,&base);
remove_pr(robot,2);
get_pr(robot,2 , &coor_type,coor,&tool,&base);
set_pr_comment(robot,2 ,comment);
get_pr_comment(robot,2 ,comment);
```

4.3. 系統變數命令

4.3.1. 設定加速度比例：

int set_acc_dec_ratio(**HROBOT** robot, **int** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
value	int	加速度比例 1-100(%)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.3.2. 取得加速度比例：

int get_acc_dec_ratio(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 加速度比例 1-100(%) 失敗: -1

C++:

```
set_operation_mode (robot, 1);
set_acc_dec_ratio(robot,20);
acc=get_acc_dec_ratio (robot);
```

➤ 自動模式才可以設定加/減速度的比例

4.3.3. 設定加速度時間：

int set_acc_time(**HROBOT** robot, **double** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
Value	double	加速度時間
回傳值	int	成功: 0 失敗: 錯誤代碼

4.3.4. 取得加速度時間：

double get_acc_time(**HROBOT** robot)

參數	資料型態	說明
Robot	HROBOT	裝置的 ID
回傳值	double	成功: 加速度時間 失敗: -1

4.3.5. 設定點對點運動速度：

int set_ptp_speed_ratio(**HROBOT** robot, **int** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
value	value	點對點速度比例 1-100(%)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.3.6. 取得點對點運動速度：

int get_ptp_speed_ratio(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID

回傳值	int	成功: 速度比例 1-100(%) 失敗: -1
-----	-----	-----------------------------

C++:

```
set_ptp_speed_ratio (robot,50);  
vel = get_ptp_speed_ratio (robot);
```

4.3.7. 設定直線運動速度：

int set_lin_speed(**HROBOT** robot, **double** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
value	value	直線運動速度(mm/s) 上限數值依機型而定
回傳值	int	成功: 0 失敗: 錯誤代碼

4.3.8. 取得直線運動速度：

double get_lin_speed(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	double	成功: 直線運動速度(mm/s) 失敗: -1

C++:

```
set_lin_speed(robot);  
vel =get_lin_speed(robot);
```

4.3.9. 設定整體速度：

int set_override_ratio(**HROBOT** robot, **double** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
value	value	整體速度比例:1-100(%)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.3.10. 取得整體速度：

int get_override_ratio(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 整體速度比例:1-100(%) 失敗: -1

C++:

```
set_override_ratio(robot,80);
override=get_override_ratio(robot);
```

4.3.11. 設定機器人編號：

int set_robot_id (**HROBOT** robot, char* robot_id)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
robot_id	char*	
回傳值	int	成功: 整體速度比例:1-100(%) 失敗: -1

C++:

```
char* robot_id = "robot1"
set_robot_id (robot, robot_id);
```

4.3.12. 取得機器人編號：

int set_robot_id (**HROBOT** robot, **char*** robot_id)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
robot_id	char*	
回傳值	int	成功: 整體速度比例:1-100(%) 失敗: -1

C++:

```
char* v = new char[256];
get_robot_id(robot, v);
delete[] v;
```

4.3.13. 設定運動平滑半徑：

int set_smooth_length(**HROBOT** robot, **int** radius)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
radius	int	運動平滑半徑，需大於 100
回傳值	int	成功: 0 失敗: 錯誤代碼

```
double radius = 200.0;
set_smooth_length (robot, radius);
```

4.3.14. 取得錯誤代碼:

int get_alarm_code(**HROBOT** robot, **int&** count **uint64_t*** alarm_code)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
count	int&	傳回警報數量
alarm_code	unsigned uint64_t[20]	警報代碼陣列 <ul style="list-style-type: none"> ➤ 最多存放 20 筆警報 ➤ 呼叫 clear_alarm 會將此陣列清空 ➤ 警報代碼對應請參考 HRSS 軟體手冊 ➤ 轉成 16 進制顯示以對應軟體手冊
回傳值	int	成功: 0 失敗: -1

C++:

```
uint64_t alarm_code[20]={0};
int* count=0;
get_alarm_code(robot,count,alarm_code);
```

4.3.15. 設定數位設置:

int set_digital_setting (**HROBOT** robot, **int*** index **char*** text)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
count	int[12]	* 所設定的 DIO/SIO 來觸發 Clear Error 等指令。 * index[0], index[2], index[4] 為 0 選擇 DI 觸發，為 1 選擇 SI 觸發。 * index[6], index[8], index[10] 為 0 選擇 DO 觸發，為 1 選擇 SO 觸發。 * index[1], index[3], index[5], index[7], index[9], index[11] 為 0 禁用此功能，DIO 範圍 1~48，SIO 範圍 1~256
text	char	設定 DI/SI 腳位觸發 external alarm 所顯示的文字。
回傳值	int	成功: 0 失敗: 2004

4.3.16. 取得數位設置:

int get_digital_setting (**HROBOT** robot, **int*** index **char*** text)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
count	int[13]	* 所設定的 DIO/SIO 來觸發 Clear Error 等指令。 * index[0], index[2], index[4], index[6], index[8], index[10] 0: DIO 1: SIO * Index[1], index[3], index[5], Index[7], index[9], index[11] 0: 禁用此功能

		DIO 範圍 1~48，SIO 範圍 1~256 index[12], text 文字的長度
text	char	取得 DI/SI 腳位觸發 external alarm 所顯示的文字。
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
int DIO = 0;
int SIO = 1;
int index[12] = { DIO, 36, DIO, 37, SIO, 38, SIO, 39, DIO, 40, DIO, 41 };
int recv[13] = {0};
char* text = "External Alarm";
set_digital_setting (robot, index, text);
get_digital_setting (robot, recv, text);
```

4.3.17. 設定語言：

int set_language (**HROBOT** robot, **int** language)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
language	int	0:英文 1:繁體中文 2:簡體中文 3:日文
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
int language = 0;
set_language (robot, language);
```

4.3.18. 取得控制器現在時間：

int get_controller_time (**HROBOT** robot, **int&** year, **int&** month, **int&** day, **int&** hour, **int&** minute, **int&** second)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
year	int&	年份
month	int&	月份
day	int&	日期

hour	int&	小時
minute	int&	分鐘
second	int&	秒
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
int* year = 0;
int* month = 0;
int* day = 0;
int* hour = 0;
int* minute = 0;
int* second = 0;
get_controller_time (robot, year, month, day, hour, minute, second);
```

4.3.19. 設定使用者自訂警報訊息：

int set_user_alarm_setting_message(**HROBOT** robot , **int** num, **char*** message)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
num	int	[0-9]編號
message	char*	設定要顯示的訊息
回傳值	int	成功: 0 失敗: 錯誤代碼

4.3.20. 取得使用者自訂警報訊息：

int get_user_alarm_setting_message(**HROBOT** robot , **int** num, **char*** message)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
num	int	[0-9]編號
message	char*	顯示的訊息
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
int num= 0;
char* msg = "user alarm text";
get_user_alarm_setting_message(robot, num, msg);
get_user_alarm_setting_message(robot, num, msg);
```

4.4. 輸出輸入命令

4.4.1. 取得輸入狀態:

int get_digital_input (**HROBOT** robot, **int** index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	輸入[1-48]的索引值
回傳值	int	0:OFF 1:ON

C++:

```
int state;
state=get_digital_input (robot,1);
```

4.4.2. 設定輸入模擬:

int set_DI_simulation_Enable (**HROBOT** robot, **int** index, **bool** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	輸入[1-48]的索引值
value	bool	true：啟動 false：關閉
回傳值	int	成功:0 失敗:-1

4.4.3. 設定輸入狀態:

int set_DI_simulation (**HROBOT** robot, **int** index, **bool** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	輸入[1-48]的索引值
value	bool	true：啟動 false：關閉
回傳值	int	成功:0 失敗:-1

備註	需先啟動輸入模擬
----	----------

4.4.4. 取得輸入模擬:

int get_DI_simulation_Enable (**HROBOT** robot, **int** index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	輸入[1-48]的索引值
回傳值	int	0:OFF 1:ON

C++:

```
int index = 1;
bool ON = true;
int recv = -1;
set_DI_simulation_Enable(robot, index, ON);
set_DI_simulation(robot, index, ON);
recv = get_DI_simulation_Enable(robot, index);
```

4.4.5. 取得輸出狀態:

int get_digital_output (**HROBOT** robot, **Int** Index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
Index	int	輸出[1-48]的索引值
回傳值	int	0:OFF 1:ON

C++:

```
int state;
state=get_digital_output (robot,1);
```

4.4.6. 設定輸出狀態:

int set_digital_output (**HROBOT** robot, **Int** Index, **bool** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
Index	int	輸出[1-48]的索引值
value	bool	true or false

回傳值	int	成功:0 失敗:-1
-----	-----	---------------

C++:

```
int index;
set_digital_output (robot, index, true);
```

4.4.7. 設定輸入註解:

int set_digital_input_comment (HROBOT robot, int index, char* comment)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	輸入[1-48]的索引值
comment	char*	設定註解的文字
回傳值	int	成功:0 失敗:-1

4.4.8. 設定輸出註解:

int set_digital_output_comment (HROBOT robot, int index, char* comment)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	輸出[1-48]的索引值
comment	char*	設定註解的文字
回傳值	int	成功:0 失敗:-1

4.4.9. 取得輸入註解:

int get_digital_input_comment (HROBOT robot, int index, char* comment)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	輸入[1-48]的索引值
comment	char*	取得註解的文字
回傳值	int	成功:註解長度 失敗:-1

4.4.10. 取得輸出註解:

int get_digital_output_comment (HROBOT robot, int index, char* comment)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	輸出[1-48]的索引值
comment	char*	取得註解的文字
回傳值	int	成功:註解長度 失敗:-1

C++:

```
int index = 1;
char* comment = "Test comment";
char str[20] = "";
set_digital_input_comment (robot, comment, index-1, 0)
get_digital_input_comment(robot, index, str)
set_digital_output_comment (robot, comment, index-1, 0)
get_digital_output_comment(robot, index, str)
```

4.4.11. 取得機器人輸入:

int get_robot_input (HROBOT robot, Int Index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
Index	int	輸入[1-8]的索引值
回傳值	int	0:OFF 1:ON

C++:

```
int state;
state=get_robot_input (robot,1);
```

4.4.12. 取得機器人輸出:

int get_robot_output (HROBOT robot, Int Index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
Index	int	輸出[1-8]的索引值
回傳值	int	0:OFF

		1:ON
--	--	------

C++:

```
int state;
state=get_robot_output (robot,1);
```

4.4.13. 設定機器人輸出:

int set_robot_output (**HROBOT** robot ,**Int** Index,**bool** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
Index	int	輸出[1-8]的索引值
value	bool	true 或 false
回傳值	int	成功:0 失敗:錯誤代碼

C++:

```
int state;
set_robot_output (robot,1,true);
```

4.4.14. 取得電磁閥輸出:

int get_valve_output (**HROBOT** robot ,**Int** Index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
Index	int	輸出[1-3]的索引值
回傳值	int	0:OFF 1:ON

C++:

```
int state;
state=get_valve_output (robot,1);
```

4.4.15. 設定電磁閥輸出:

int set_valve_output (**HROBOT** robot ,**Int** Index,**bool** value)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
Index	int	輸出[1-3]的索引值
value	bool	true 或 false
回傳值	int	成功:0 失敗:錯誤代碼

C++:

```
int state;
set_valve_output (robot,1,true);
```

4.4.16. 取得功能輸入狀態:

int get_function_input (**HROBOT** robot ,**Int** Index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
Index	int	功能輸入[1-8]的索引值 1: Start 2: Hold 3: Stop 4: Enable 5: RSR1 6: RSR2 7: RSR3 8: RSR4
回傳值	int	0: OFF 1: ON

C++:

```
int state;
state=get_function_input (robot,1);
```

4.4.17. 取得功能輸出狀態:

int get_function_output (**HROBOT** robot ,**Int** Index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID

Index	int	功能輸出[1-8]的索引值 1: Run 2: Held 3: Fault 4: Ready 5: ACK1 6: ACK2 7: ACK3 8: ACK4
回傳值	int	0:OFF 1:ON

C++:

```
int state;
state=get_function_output (robot,1);
```

4.4.18. 取得模組輸入設置:

`int get_module_input_config (HROBOT s, int index, bool& sim, bool& value, int& start, int& end, char* comment)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸入編號[0-31]的索引值
sim	bool	模擬值 0:OFF 1:ON
value	bool	0:OFF 1:ON
start	int	數位輸入編號[0-48]的開始索引值
end	int	數位輸入編號[0-48]的結束索引值 (不得小於 start)
comment	char	該模組輸入的註解

4.4.19. 取得模組輸出設置:

`int get_module_output_config (HROBOT s, int index, bool& sim, bool& value, int& start, int& end, char* comment)`

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
index	int	模組輸出編號[0-31]的索引值
sim	bool	模擬值 0:OFF 1:ON
value	bool	0:OFF 1:ON
start	int	數位輸出編號[0-48]的開始索引值
end	int	數位輸出編號[0-48]的結束索引值
comment	char	該模組輸出的註解

4.4.20. 設定模組輸入模擬:

int set_module_input_config (**HROBOT** s, **int** index, **bool** enable)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸入編號[0-31]的索引值
enable	bool	模擬值 0:OFF 1:ON

4.4.21. 設定模組輸入:

int set_module_input_value (**HROBOT** s, **int** index, **bool** enable)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸入編號[0-31]的索引值
enable	bool	0:OFF 1:ON ※需 sim 為 ON 時，才能設為 ON。

4.4.22. 設定模組輸入開始編號:

int set_module_input_start(**HROBOT** s, **int** index, **int** start_number)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸入編號[0-31]的索引值
start_number	int	數位輸入的編號[0-48]索引值

4.4.23. 設定模組輸入結束編號:

`int set_module_input_end(HROBOT s, int index, int end_number)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸入編號[0-31]的索引值
end_number	int	數位輸入的編號[0-48]索引值 (不得小於 start)

4.4.24. 設定模組輸入註解

`int set_module_input_comment(HROBOT s, int index, char* comment)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸入編號[0-31]的索引值
comment	char*	設定指定模組輸入的註解

4.4.25. 設定模組輸出:

`int set_module_output_value(HROBOT s, int index, bool enable)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸出編號[0-31]的索引值
enable	bool	0:OFF 1:ON

4.4.26. 設定模組輸出開始編號:

`int set_module_output_start(HROBOT s, int index, int start_number)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸出編號[0-31]的索引值
start_number	int	數位輸出的編號[0-48]索引值

4.4.27. 設定模組輸出結束編號:

`int set_module_output_end(HROBOT s, int index, int end_number)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID

index	int	模組輸出編號[0-31]的索引值
end_number	int	數位輸出的編號[0-48]索引值 (不得小於 start)

4.4.28. 設定模組輸出註解

`int set_module_output_comment(HROBOT s, int index, char* comment)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸出編號[0-31]的索引值
comment	char*	設定註解的文字

4.4.29. 設定模組輸入型態

`int set_module_input_type(HROBOT s, int index, int type)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸出編號[0-31]的索引值
type	int	0: DI 1: SI

4.4.30. 設定模組輸出型態

`int set_module_output_type(HROBOT s, int index, int type)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸出編號[0-31]的索引值
type	int	0: DO 1: SO

4.4.31. 保存模組設定

`int save_module_io_setting (HROBOT s)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID

C++:

```
bool sim,value;
int start,end;
char comment[100];
get_module_input_config (robot,0,sim,value,start,end,comment);
get_module_output_config(robot,0,sim,value,start,end,comment);
set_module_input_simulation(robot,0,true);
set_module_input_value(robot, 0,true);
set_module_input_start(robot, 0,2);
set_module_input_end(robot, 0,5);
set_module_input_type(robot, 0, type)
set_module_output_comment(robot, 0, comment);
set_module_output_value(robot,0, true);
set_module_output_start(robot, 0,10);
set_module_output_end(robot, 0, 10);
set_module_output_comment(robot, 0, comment);
set_module_output_type(robot, 0, type)
save_module_io_setting(robot);
```

4.3.32. 運動中進行 DO 開關操作

int SyncOutput (HROBOT robot, int O_type, int O_id, int on_off, int synMode, int delay, int distance)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
O_Type	int	0: DO 數位輸出 1: RO 機器人輸出
O_id	int	[1-48]數位輸出的索引值 [1-8]機器人輸出的索引值
on_off	int	0: 輸出 off 1: 輸出 on
synMode	int	0: Start 開始運動位置判斷延遲時間；delay 若為正數，經過 N 毫秒改變輸出；若為負數，開始運動就會改變輸出。 1: End 結束位置判斷延遲時間；delay 若為正數，到點位才會改變輸出；若為負數，結束位置前 N 毫秒改變輸出。 2: Path 路徑長度判斷；距離若為正，開始位置距

		離+延遲時間=改變輸出的時間點；距離若為負，結束位置前距離+延遲時間=改變輸出的時間點。
delay	int	±1000ms 延遲時間
distance	int	只在 Path 模式下有效 ±2000mm 延遲距離
回傳值	int	成功: 0 失敗: 參數錯誤
備註	最多同時支援 8 個 SyncOutput 指令。	

C++:

```
int type = 0; //DO
int index = 3;
int ON = 1;
int Start = 0;
int End = 1;
int Path = 2;
int delay = 500;
int distance = 50;
double p1[6] = { 0, -200, 0, 0, 0, 0 };
double p2[6] = { 0, 150, 0, 0, 0, 0 };
lin_rel_pos(robot, 0, 0, p1 );
// 從p1開始運動後經過500ms後，DO[3]轉為ON。
SyncOutput(robot, type, index, ON, Start, delay, distance);
// 從p1開始運動快到p2前1000ms的點位，DO[4]轉為ON。
SyncOutput(robot, type, 4, ON, End, -1000, distance);
// 從p1開始運動後經過50mm距離加上500ms後的點位，DO[5]轉為ON。
SyncOutput(robot, type, 5, ON, Path, delay, distance);
// 從p1開始運動後經過50mm距離扣掉1000ms後的點位，DO[6]轉為ON；此選項會比
// DO[5]還要早1000ms的時間改變輸出狀態。
SyncOutput(robot, type, 6, ON, Path, -1000, distance);
lin_rel_pos( device_id, 0, 0, p2 );
```

4.5. 座標命令

4.5.1. 設定基底號碼：

int set_base_number(**HROBOT** robot , **int** baseNum, **int** num)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
baseNum	int	底座標號
num	int	欲選擇的基底編號(0-31)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.5.2. 取得基底號碼：

int get_base_number(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 基底編號(0-31) 失敗: -1

C++:

```
set_base_number(robot,1);
int num=get_base_number(robot);
```

4.5.3. 定義底座座標：

int define_base(**HROBOT** robot ,**int** baseNum ,**double ***coor)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
baseNum	int	底座座標號碼(1-31) 基底編號 0 不允許定義 基底編號 1-30:可自定義座標
coor	double[6]	座標{X,Y,Z,A,B,C} 範圍(2147418.112,- 2147418.112)
回傳值	int	成功:0 失敗: 錯誤代碼

4.5.4. 取得底座座標：

int get_base_data(**HROBOT** robot ,**int** num,**double*** coor)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
num	int	欲取得的基底座標編號(0-31)
coor	double[6]	座標{X,Y,Z,A,B,C}
回傳值	int	成功: 0 失敗: 錯誤代碼

4.5.5. 設定工具號碼：

int set_tool_number(**HROBOT** robot , **int** num)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
num	int	欲選擇的工具編號(0-15)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.5.6. 取得工具號碼：

int get_tool_number(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 工具編號(0-15) 失敗: 錯誤代碼

C++:

```
set_tool_number(robot,10);           // set_tool_number
int vel=get_tool_number(robot);// get_tool_number
```

4.5.7. 定義工具座標：

int define_tool(**HROBOT** robot , **int** toolNum, **double*** coor)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
toolNum	int	欲定義的工具編號(1-15) 工具編號 0 不可定義座標 工具編號 1-15:可自定義座標

coor	double[6]	座標{X,Y,Z,A,B,C} 範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.5.8. 取得工具座標：

`int get_tool_data(HROBOT robot ,int num,double* coor)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
num	int	欲定義的工具編號(0-15)
coor	double[6]	座標{X,Y,Z,A,B,C} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
(1)
double coor={0,50,300,0,0,0};
define_tool(robot,2,coor);
double ToolCoor[6];
get_tool_data(robot,2,ToolCoor);
(2)
double coor={0,0,100,0,0,0};
define_base(robot,2,coor);
double BaseCoor[6];
get_base_data(robot,2,BaseCoor);
```

5. 定義工具座標，並利用get_tool_data來取得座標資訊。
6. 定義底座座標，並利用get_base_data來取得座標資訊。

4.6. 任務命令

4.6.1. 設定 RSR：

`int set_rsr(HROBOT robot ,char* file_name,int index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
file_name	char*	欲設定之程式檔案名稱
index	int	任務設定編號 RSR:(1-4)
回傳值	int	成功: 0 失敗: 錯誤代碼

➤ 注意!如果 SDK 使用操作者層級連線，控制器的 RSR 會失效。

C++:

```
char* filename = "file.hrb";
set_rsr(robot, filename, 1);
```

4.6.2. 取得 RSR 程式名稱：

int get_rsr_prog_name (**HROBOT** robot, **int** index, **char*** file_name)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	欲取得 RSR 之編號 RSR:(1-4)
file_name	char*	取得程式檔案名稱
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
char* filename;
get_rsr_prog_name(robot, 1, filename);
```

4.6.3. 移除 RSR：

int remove_rsr (**HROBOT** robot, **int** index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	任務設定編號 RSR:(1-4)
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
remove_rsr (robot, 1);
```

4.6.4. RSR/PNS 啟動外部觸發任務：

`int ext_task_start(HROBOT robot, int mode, int select)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	外部觸發模式 0: RSR 模式 1: PNS 模式
select	int	任務設定編號 RSR:(1-4) PNS:(1-2047)
回傳值	int	成功: 0 失敗: 錯誤代碼

- 使用此命令前需設定 RSR/PNS 任務編號
- 執行此命令時，若有其他運動命令則命令將會失敗。執行前需執行 motion_abort 指令清空運動佇列或等待運動命令執行完畢。

4.6.5. 啟動任務：

`int task_start(HROBOT robot, char* file_name)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
file_name	char*	任務名稱，指定的檔名需要有".hrb" 任務需存在於 HRSS 中
回傳值	int	成功: 0 失敗: 錯誤代碼

- 執行此命令時，若有其他運動命令則命令將會失敗。執行前需執行 motion_abort 指令清空運動佇列或等待運動命令執行完畢。

C++:

```
double pos1[6] = { 50, 0, 0, 0, -90, 0 };
double pos2[6] = { -50, 0, 0, 0, -90, 0 };
set_motor_state(s, 1); // servo on

// wait for servo on
while (!get_motor_state(s)) {
    Sleep(10);
}

// execute motion command
set_command_id(s, 20);
ptp_axis(s, pos1); // this motion id will be 20
set_command_id(s, 21);
ptp_axis(s, pos2); // this motion id will be 21

// task start will fail because there are motion command in motion queue
task_start(s, "program_Test.hrb"); // fail

// clean command in motion queue
motion_abort(s);

// wait for command count = 0
while (get_command_count(s)) {
    Sleep(100);
}

// task start will succeed
task_start(s, "program_Test.hrb"); // successful

// task start will fail because there is a task exist
task_start(s, "program_Test.hrb"); // fail
```

4.6.6. 暫停任務：

int task_hold(**HROBOT** robot)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: 錯誤代碼

4.6.7. 繼續任務：

int task_continue(**HROBOT** robot)

參數	資料型態	說明
Robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: 錯誤代碼

4.6.8. 停止任務：

int task_abort(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

(1)

```
ext_task_start(s, 1, 9);  
Sleep(3000);  
ext_task_start(s, 1, 94);  
task_hold(s);  
Sleep(1000);  
task_conti(s);  
Sleep(1000);  
task_abort(s);  
Sleep(100);
```

(2)

```
ext_task_start(s, 0, 4);  
Sleep(3000);  
task_abort(s);  
Sleep(100);
```

(3)

```
task_start(s, "task1.hrb");  
Sleep(3000);  
task_abort(s);  
Sleep(100);
```

(1). 啟動 PNS 模式編號 9 的任務

運行三秒後再啟動 PNS 任務 94 的任務

- 若任務 9 已經執行完則會依序執行任務 94
- 若任務 9 未執行完成則任務起動失敗，並回傳錯誤代碼 4013。任務 9 會繼續執行

暫停任務後一秒繼續任務

繼續執行一秒後停止任務

(2). 啟動 RSR 模式編號 4 的任務

運行三秒後停止任務

(3). 啟動任務名稱為「task1」的任務

運行三秒後停止任務

4.6.9. 取得目前執行任務名稱：

`int get_execute_file_name (HROBOT robot, char* filename)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
filename	char*	目前任務名稱
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
char* execute_file_name = new char[256];
get_execute_file_name(robo, execute_file_name);
std::cout << "Execute file name : " << execute_file_name << std::endl;
delete[] execute_file_name;
```

4.7. 檔案管理命令

4.7.1. 下載 HIWIN Robot language 檔案

`int download_file (HROBOT robot, char * from_path, char * to_path)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_path	char*	欲下載之 hrb 檔的路徑。
to_path	char*	hrb 檔下載後的路徑。
回傳值	int	成功: 0 失敗: -1

C++:

```
HROBOT s;  
char form_path[255] = "hrss.hrb";  
char to_path [255] = " HRL_files\\hrss.hrb";  
download_file(robot, form_path, to_path);
```

4.7.2. 上傳 HIWIN Robot language 檔案

int send_file(**HROBOT** robot , **char*** from_path, **char*** to_path)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_path	char*	欲上傳之 hrb 檔，須指定檔案的路徑。
to_path	char*	欲存放的 hrb 檔案路徑，此檔案會存放在 HRS 的 Program 資料夾之下。
回傳值	int	成功: 0 失敗: -1
備註	若是檔名一樣會直接覆蓋	

C++:

```
HROBOT robot;  
send_file(robot , "folder1\\a123.hrb" , "a123.hrb");
```

4.7.3. 刪除 Robot 運動檔案

int delete_file(**HROBOT** robot , **char*** file_path)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
file_path	char*	刪除指定路徑的 hrb 檔，此檔案會在 HRS 的 Program 資料夾之下。
回傳值	int	成功: 0 失敗: 9999

4.7.4. 刪除 Robot 運動檔案資料夾

int delete_folder (**HROBOT** robot , **char*** file_path)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
file_path	char*	刪除指定路徑的資料夾，此資料夾會在 HRS 的 Program 資料夾之下。

回傳值	int	成功: 0 失敗: 錯誤代碼
-----	-----	-------------------

4.7.5. 新增 Robot 運動檔案資料夾

int new_folder (HROBOT robot , char* name)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
name	char*	新增資料夾名稱，此資料夾會在 HRS 的 Program 資料夾之下。
回傳值	int	成功: 0 失敗: 9999

4.7.6. 重新命名 Robot 運動檔案

int file_rename (HROBOT robot , char* old_file_name, char* new_file_name)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
old_file_name	char*	欲重新命名之 hrb 檔，須指定檔案的路徑。
new_file_name	char*	更改為所指定的檔案名稱。
回傳值	int	成功: 0 失敗: 錯誤訊息

4.7.7. 拖動 Robot 運動檔案

int file_drag (HROBOT robot , char* old_file_name, char* new_file_name)

參數	資料型態	說明
Robot	HROBOT	裝置的 ID
old_file_name	char*	欲拖動之 hrb 檔，須指定檔案的路徑。
new_file_name	char*	更改為所指定的檔案路徑。
回傳值	int	成功: 0 失敗: 錯誤訊息

C++:

```
HROBOT robot;
new_folder(robot , "test_folder");
send_file(robot , "folder1/a123.hrb", "test_folder/a123.hrb");
file_drag(robot , "test_folder/a123.hrb", "b456.hrb");
file_rename (robot , "b456.hrb", "c789.hrb");
delete_file (robot , "c789.hrb");
delete_folder(robot , "test_folder");
```

4.7.8. 取得檔案數量

int get_prog_number(**HROBOT** robot)

參數	資料型態	說明
Robot	HROBOT	裝置的 ID
回傳值	int	回傳 Program 資料夾內，檔案與資料夾全部數量

4.7.9. 取得檔案名稱

int get_prog_name(**HROBOT** robot , **int** file_index, **char*** file_name)

參數	資料型態	說明
Robot	HROBOT	裝置的 ID
file_index	int	取得第 n 個檔案，包含資料夾
file_name	char*	<ul style="list-style-type: none"> ➤ 取得的檔案名稱 ➤ 按照檔案與資料夾名稱排序，資料夾也算是單獨一個名稱
回傳值	int	成功: 0 失敗: 2004
備註	需先取得檔案數量 get_prog_number，才可以執行此 API。	

C++:

```
char str[20] = "";
int num = HRobot.get_prog_number( robot );
for ( int i = 0; i < num; i++ ) {
    get_prog_name( robot, i, str );
    printf("name: %s", str);
}
```

4.8. 控制器命令

4.8.1. 取得 HRSS 當前模式：

`int get_hrss_mode(HROBOT robot)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	T1 模式: 0 AUTO 模式: 1 T2 模式: 2 EXT 模式: 3 失敗: -1

C++:

```
ret=get_hrss_mode (robot);
```

4.8.2. 伺服設定：

`int set_motor_state(HROBOT robot ,int state)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
state	int	0:伺服關閉 1:伺服啟動 伺服啟動後需等待 100ms 左右的時間才可下達執行其他運動命令
回傳值	int	成功:0 失敗:錯誤代碼

4.8.3. 取得伺服狀態：

`int get_motor_state(HROBOT robot)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	伺服關閉:0 伺服啟動:1 失敗: -1

C++:

```
if (get_motor_state(robot)==0){
    set_motor_state(robot,1);
}
```

(1). 判斷馬達是否激磁，若未激磁則將馬達激磁。

4.8.4. 設定操作模式：

int set_operation_mode (**HROBOT** robot, **int** mode)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	手動模式: 0 自動模式: 1
傳回值	int	成功: 0 失敗: 錯誤代碼

➤ 手動模式

- 運動命令速度限制在 250mm/s
- 可使用吋動指令
- 整體速度會設置成 10%
- 無法設定加速度比例
- 無法設定直線速度與點對點速度比例

➤ 自動模式

- 運動命令速度限制視機型而定
- 無法使用吋動指令
- 整體速度會設置成 10%
- 此模式下執行運動程式，欲維持離線下繼續運動，需按標準程序，使用 disconnect 命令斷線。

4.8.5. 取得操作模式：

int get_operation_mode (**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
傳回值	int	手動模式: 0 自動模式: 1

		失敗: -1
--	--	--------

4.8.6. 錯誤清除：

int clear_alarm (**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
傳回值	int	成功: 0 失敗: 錯誤代碼

C++:

```
//if your arm get error
clear_alarm(robot);
```

4.8.7. 更新 HRSS：

int update_hrss (**HROBOT** robot, **char*** path)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
path	char*	更新檔路徑
傳回值	int	成功: 0 失敗: -1 更新檔檔案錯誤:4020 檔案傳送失敗:4021 更新檔解壓縮失敗:4022 控制器空間不足:4023 指定路徑更新檔案不存在:4024

C++:

```
char path[255]=" C:/ HRSS 3.2.12.3940_update.exe";
update_hrss(robot,path);
```

4.9. 吋動

4.9.1. 吋動：

int jog(HROBOT robot, int space_type, int index, int dir)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
space	int	座標系種類 0:基底座標系 1:關節座標系 2:工具座標系
index	int	吋動對像 笛卡爾座標(X:0,Y:1,Z:2,A:3,B:4,C:5) 關節座標系(A1:0, A2:1, A3:2, A4:3, A5:4, A6:5)
dir	int	方向 1:正方向 -1:負方向
回傳值	int	成功: 0 失敗: 錯誤代碼

➤ 只有在手動模式下有效

4.9.2. 吋動復歸：

int jog_home(HROBOT robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: 錯誤代碼

➤ 只在安全線速模式下有效

4.9.3. 吋動停止：

int jog_stop(HROBOT robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: 錯誤代碼

只在安全線速模式下有效

4.10.運動命令

4.10.1. 絕對座標位置點對點運動：

`int ptp_pos(HROBOT robot, int mode, double* p)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 依兩線段速度平滑
p	double[6]	笛卡爾座標{X,Y,Z,A,B,C} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.10.2. 絕對關節角度點對點運動：

`int ptp_axis(HROBOT robot, int mode, double* p)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
p	double[6]	關節座標{A1,A2,A3,A4,A5,A6} 數值範圍(2147418.112,- 2147418.112)
mode	int	平滑模式 0: 關閉平滑功能 1: 依兩線段速度平滑
回傳值	int	成功: 0 失敗: 錯誤代碼

4.10.3. 相對座標位置點對點運動：

`int ptp_rel_pos(HROBOT robot, int mode, double* p)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 依兩線段速度平滑
p	double[6]	笛卡爾座標{X,Y,Z,A,B,C} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0

		失敗: 錯誤代碼
--	--	----------

4.10.4. 相對關節角度點對點運動：

int ptp_rel_axis(**HROBOT** robot, **int** mode, **double*** p)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 依兩線段速度平滑
p	double[6]	關節座標{A1,A2,A3,A4,A5,A6} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.10.5. 位置暫存器之點對點運動：

int ptp_pr(**HROBOT** robot, **int** mode, **int** p)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 依兩線段速度平滑
p	int	位置暫存器編號(1-100)
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
double AxishomePoint[6]={0,0,0,0,-90,0};
ptp_axis(s,0, AxishomePoint); // RA robot home point

double TargetCartPoint[6]={217.5 ,58.5 ,601.659 ,60.8 ,35.5 ,95.7}
ptp_pos(s,0, TargetCartPoint); // move to Point you want in Cart Space using PTP

double Xoffset[6]={-50,0,0,0,0,0};
ptp_rel_pos(s, 0, 0, Xoffset); // new point will be { 167.5,58.5,601.659,60.8,35.5,95.7}

set_pr(robot,100,1, AxishomePoint,0,0);
// set pr no.100 data with joint space{0,0,0,0,-90,0} in base 0 tool 0
ptp_pr(robot, 0,100); //move to position register no.100 using PTP
```

4.10.6. 絕對座標位置直線運動：

int lin_pos(**HROBOT** robot, **int** mode, **double** smooth_value, **double*** p)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 貝茲曲線平滑百分比 2: 貝茲曲線平滑半徑 3: 依兩線段速度平滑
smooth_value	double	mode 為 0: 無效 mode 為 1: 平滑百分比(1-100%) mode 為 2: 平滑半徑(mm) mode 為 3: 無效
p	double[6]	笛卡爾座標{X,Y,Z,A,B,C} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.10.7. 絕對關節角度直線運動：

int lin_axis(**HROBOT** robot, **int** mode, **double** smooth_value, **double*** p)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 貝茲曲線平滑百分比 2: 貝茲曲線平滑半徑 3: 依兩線段速度平滑
smooth_value	double	mode 為 0: 無效 mode 為 1: 平滑百分比(1-100%) mode 為 2: 平滑半徑(mm) mode 為 3: 無效
p	double[6]	關節座標{A1,A2,A3,A4,A5,A6} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功:0 失敗: 錯誤代碼

4.10.8. 相對座標位置直線運動：

int lin_rel_pos(**HROBOT** robot, **int** mode, **double** smooth_value, **double*** p)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 貝茲曲線平滑百分比 2: 貝茲曲線平滑半徑 3: 依兩線段速度平滑
smooth_value	double	mode 為 0: 無效 mode 為 1: 平滑百分比(1-100%) mode 為 2: 平滑半徑(mm) mode 為 3: 無效
p	double[6]	笛卡爾座標{X,Y,Z,A,B,C} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.10.9. 相對關節角度直線運動：

int lin_rel_axis(**HROBOT** robot , **int** mode, **double** smooth_value, **double*** p)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 貝茲曲線平滑百分比 2: 貝茲曲線平滑半徑 3: 依兩線段速度平滑
smooth_value	double	mode 為 0: 無效 mode 為 1: 平滑百分比(1-100%) mode 為 2: 平滑半徑(mm) mode 為 3: 無效
p	double[6]	關節座標{A1,A2,A3,A4,A5,A6} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.10.10. 位置暫存器之直線運動：

int lin_pr(**HROBOT** robot, **int** mode, **double** smooth_value, **int** p)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 貝茲曲線平滑百分比 2: 貝茲曲線平滑半徑 3: 依兩線段速度平滑
smooth_value	double	mode 為 0: 無效 mode 為 1: 平滑百分比(1-100%) mode 為 2: 平滑半徑(mm) mode 為 3: 無效
p	int	位置暫存器編號(1-100)
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
double AxishomePoint[6]={0,0,0,0,-90,0};
lin_axis(s, AxishomePoint); // RA robot home point

double TargetCartPoint[6]={217.5 ,58.5 ,601.659 ,60.8 ,35.5 ,95.7}
lin_pos(s, 0, 0, TargetCartPoint);

double Xoffset[6]={-50,0,0,0,0,0};
lin_rel_pos(s, 0, 0, Xoffset); // new point will be {167.5,58.5,601.659,60.8,35.5,95.7}

set_pr(robot,100,1, AxishomePoint,0,0);
// set pr no.100 data with joint space{0,0,0,0,-90,0} in base 0 tool 0
lin_pr(robot, 0, 0, 100); //move to position register no.100 using LIN
```

4.10.11. 絕對座標位置圓弧運動：

int circ_pos(**HROBOT** robot, **int** mode, **double*** p_aux,**double*** p_end)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 依兩線段速度平滑
p_aux	double[6]	圓弧運動之弧點 笛卡爾座標{X,Y,Z,A,B,C} 數值範圍(2147418.112,- 2147418.112)
p_end	double[6]	圓弧運動之終點 笛卡爾座標{X,Y,Z,A,B,C} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

4.10.12. 關節座標位置圓弧運動：

int circ_axis(**HROBOT** robot, **int** mode, **double*** p_aux,**double*** p_end)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式

		0: 關閉平滑功能 1: 依兩線段速度平滑
p_aux	double[6]	圓弧運動之弧點 關節座標{A1, A2, A3, A4, A5, A6} 數值範圍 (2147418.112,- 2147418.112)
p_end	double[6]	圓弧運動之終點 關節座標{A1, A2, A3, A4, A5, A6} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
//Declare p1,p2
double aux_p[6] = { 174.5, 368, 164.7, -180, 0, 90 };
double end_p[6] = { 51, 368, -69.7, 180, 0, 90 };
double homeAxis[6] = { 0, 0, 0, 0, -90, 0};
ptp_axis(s, 0, homeAxis); // ptp to home point
circ_pos(s, 0, aux_p, end_p);

double aux_p[6] = { -20, 0, -34, 0, -56, -20 };
double end_p[6] = { -13.5, 22.4, -28.4, 0, -96, -13.5 };
circ_axis(s, 0, aux_p, end_p);
```

4.10.13. 位置暫存器之圓弧運動：

int circ_pr(HROBOT robot, int mode, int p1, int p2)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	int	平滑模式 0: 關閉平滑功能 1: 依兩線段速度平滑
p1	int	位置暫存器編號(0-100)
p2	int	位置暫存器編號(0-100)
回傳值	int	成功: 0 失敗: 錯誤代碼

C++:

```
//Set position register 1,2
double aux_p1[6] = { 174.5, 368, 164.7, -180, 0, 90 };
double end_p1[6] = { 51, 368, -69.7, 180, 0, 90 };
double aux_p2[6] = { -20, 0, -34, 0, -56, -20 };
double end_p2[6] = { -13.5, 22.4, -28.4, 0, -96, -13.5 };
set_pr_type(s, 1, 0); // set_pr type to cart space
set_pr_type(s, 2, 0);
set_pr_type(s, 3, 1); // set_pr type to joint space
set_pr_type(s, 4, 1);
set_pr_coordinate(s, 1, aux_p1);
set_pr_coordinate(s, 2, end_p1);
set_pr_coordinate(s, 3, aux_p2);
set_pr_coordinate(s, 4, end_p2);
circ_pr(robot, 0, 1,2); // circle motion
```

4.10.14. 運動暫停：

int motion_hold(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: 錯誤代碼

➤ 當有任務執行時，指令會失敗。

4.10.15. 運動繼續：

int motion_continue(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: 錯誤代碼

➤ 當有任務執行時，指令會失敗。

4.10.16. 運動停止：

int motion_abort(**HROBOT** robot)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: 錯誤代碼

➤ 當有任務執行時，指令會失敗。

4.10.17. 運動延遲：

`int motion_delay(HROBOT robot, int delay)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
delay	int	延遲時間，單位毫秒
回傳值	int	成功: 0 失敗: 錯誤代碼

➤ 當有任務執行時，指令會失敗。

C++:

```
double pos1[6] = { 50, 0, 0, 0, -90, 0 };
double pos2[6] = { -50, 0, 0, 0, -90, 0 };
set_motor_state(s, 1); // servo on

// wait for servo on
while (!get_motor_state(s)) {
    Sleep(10);
}

// execute motion command
set_command_id(s, 20);
ptp_axis(s, pos1); // this motion id will be 20
set_command_id(s, 21);
ptp_axis(s, pos2); // this motion id will be 21
motion_delay(s, 1000);
motion_hold(s);
motion_continue(s);
motion_abort(s);

// task start will fail because there are motion command in motion queue
task_start(s, "program_Test.hrb"); // fail
```

```
// clean command in motion queue
motion_abort(s);

// wait for command count = 0
while (get_command_count(s)) {
    Sleep(100);
}

// task start will succeed
task_start(s, "program_Test.hrb"); // successful

// task start will fail because there is a task exist
task_start(s, "program_Test.hrb"); // fail

// motion planning command will fail
motion_delay(s, 1000); // fail
motion_hold(s); // fail
motion_continue(s); // fail
motion_abort(s); // fail
set_command_id(s, 20); // fail
// task stop
task_abort(s);
```

4.10.18. 設定運動命令編號：

int set_command_id(**HROBOT** robot, **int** id)

參數	資料型態	說明
Robot	HROBOT	裝置的 ID
Id	int	欲設定的命令編號
回傳值	int	成功: 0 失敗: 錯誤代碼

➤ 當有任務執行時，指令會失敗。

4.10.19. 取得運動命令編號：

int get_command_id(**HROBOT** robot)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
回傳值	int	成功: 運動命令編號 失敗: -1

C++:

```

set_command_id(robot,10);
ptp_pos(robot,p1);

set_command_id(robot,11);
ptp_pos(robot,p2);

set_command_id(robot,12);
ptp_pos(robot,p3);

set_command_id(robot,13);
ptp_pos(robot,p4);

set_command_id(robot,14);
ptp_pos(robot,p5);

while(get_motion_state(robot
)!=1){
    comId=get_command_id(robot);
}

```

4.10.20. 取得運動命令佇列中命令個數：

int get_command_count(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 運動命令個數 失敗: -1

4.10.21. 取得目前運動狀態：

int get_motion_state(**HROBOT** robot)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
回傳值	int	成功: 目前運動狀態 1:閒置狀態 2:運動狀態 3:暫停狀態 4:延遲狀態 5:命令等待狀態 失敗: -1

- 閒置狀態:沒有運動命令
- 命令等待狀態:有運動命令但未執行

4.10.22. 移除運動命令佇列中單一命令：

int remove_command(**HROBOT** robot, **int** num)

參數	資料型態	說明
Robot	HROBOT	裝置的 ID
num	int	要移除的命令
回傳值	int	成功: 0 失敗: -1

4.10.23. 移除運動命令佇列中最新數個命令：

int remove_command_tail(**HROBOT** robot, **int** num)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
num	int	移除的最新命令個數
回傳值	int	成功: 0 失敗: -1

4.11. 手臂資訊命令

4.11.1. 取得目前編碼器數值：

int get_encoder_count(**HROBOT** robot, **INT32*** value)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
value	double[6]	各軸編碼器數值
回傳值	int	成功: 0 失敗: -1

4.11.2. 取得目前關節座標：

`int get_current_joint(HROBOT robot, double* coor)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
coor	double[6]	存放關節座標的陣列
回傳值	int	成功: 0 失敗: -1

C++:

```
double[6] pos;
get_current_joint(robot,pos); //get current point in joint coordinate
```

4.11.3. 取得目前絕對座標：

`int get_current_position(HROBOT robot, double* coor)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
coor	double[6]	存放絕對座標的陣列
回傳值	int	成功: 0 失敗: -1

C++:

```
double[6] pos;
get_current_position(robot,pos); //get current point in Cartesian coordinate
```

4.11.4. 取得目前轉速：

`int get_current_rpm(HROBOT robot, double* coor)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
coor	double[6]	各關節馬達轉速
回傳值	int	成功: 0

		失敗: -1
--	--	--------

C++:

```
double[6] rpm;
rpm=get_current_rpm(robot);
```

4.11.5. 取得裝置出廠時間：

int get_device_born_date (**HROBOT** robot, **int*** YMD)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
YMD	int[3]	YMD[0]: 出廠年 YMD[1]: 出廠月 YMD[2]: 出廠日
回傳值	int	成功: 0 失敗: -1

4.11.6. 取得控制器開機時間：

int get_operation_time(**HROBOT** robot, **int*** YMDHm)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
YMDHm	int[5]	YMDHm[0]: 開機時間年 YMDHm[1]: 開機時間月 YMDHm[2]: 開機時間日 YMDHm[3]: 開機時間時 YMDHm[4]: 開機時間分
回傳值	int	成功: 0 失敗: -1

4.11.7. 取得各軸馬達里程數：

int get_mileage(**HROBOT** robot, **double*** mil)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mil	double[6]	mil[0]: 一軸馬達旋轉轉數 mil[1]: 二軸馬達旋轉轉數

		mil[2]: 三軸馬達旋轉轉數 mil[3]: 四軸馬達旋轉轉數 mil[4]: 五軸馬達旋轉轉數 mil[5]: 六軸馬達旋轉轉數
回傳值	int	成功: 0 失敗: -1

➤ 馬達旋轉轉數可以借由 HRSS 歸零

4.11.8. 取得各軸馬達累積里程數：

int get_total_mileage(**HROBOT** robot, **double*** mil)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mil	double[6]	mil[0]: 一軸馬達累積旋轉轉數 mil[1]: 二軸馬達累積旋轉轉數 mil[2]: 三軸馬達累積旋轉轉數 mil[3]: 四軸馬達累積旋轉轉數 mil[4]: 五軸馬達累積旋轉轉數 mil[5]: 六軸馬達累積旋轉轉數
回傳值	int	成功: 0 失敗: -1

➤ 馬達累積旋轉轉數無法借由 HRSS 歸零

4.11.9. 取得累積稼動率：

int get_utilization (**HROBOT** robot, **int*** ult)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
ult	double[6]	ult [0]: 稼動率年 ult [1]: 稼動率月

		ult [2]: 稼働率日 ult [3]: 稼働率時 ult [4]: 稼働率分 ult [5]: 稼働率秒
回傳值	Int	成功: 0 失敗: -1

4.11.10. 取得稼働率百分比：

int get_utilization_ratio(**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	Int	成功: 稼働率百分比 失敗: -1

4.11.11. 取得馬達負載百分比：

int get_motor_torque(**HROBOT** robot, **double*** cur)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
cur	double[6]	cur [0]: 一軸扭力百分比 cur [1]: 二軸扭力百分比 cur [2]: 三軸扭力百分比 cur [3]: 四軸扭力百分比 cur [4]: 五軸扭力百分比 cur [5]: 六軸扭力百分比
回傳值	Int	成功: 0 失敗: -1

4.11.12. 取得 HRSS 版本號：

int get_hrss_version (**HROBOT** robot, **char*** ver)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
ver	double[6]	HRSS 版本號
回傳值	Int	成功: 0

		失敗: -1
--	--	--------

C++:

```
char* HrssV = new char[256];
get_hrss_version(s, HrssV);
std::cout << "HRSS version:" << HrssV << std::endl;
delete[]HrssV;
```

4.11.13. 取得機器人型號：

int get_robot_type (**HROBOT** robot, **char*** robType)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
robType	double[6]	機器人型號
回傳值	int	成功: 0 失敗: -1

C++:

```
char* v = new char[256];
get_robot_type(device_id, v);
std::cout << "ROBOT TYPE:\t" << v << std::endl;
delete[]v;
```

4.11.14. 取得負載設定值：

int get_payload_value (**HROBOT** robot, **int&** value)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
value	int&	負載值
回傳值	int	成功: 0 失敗: -1

C++:

```
int* value = -1;
get_payload_value(robot, value);
```

4.11.15. 設定復歸位置：

int set_home_point (HROBOT robot, double* joint)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
joint	double*	各關節座標
回傳值	int	成功: 0 失敗: -1

4.11.16. 取得復歸位置：

int get_home_point (HROBOT robot, double* joint)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
joint	double*	各關節座標
回傳值	int	成功: 0 失敗: -1

4.11.17. 取得上次關機位置：

int get_previous_pos (HROBOT robot, double* joint)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
joint	double*	各關節座標
回傳值	int	成功: 0 失敗: -1

C++:

```
double joint[6] = {20, 20, 0, 0, 0, 0};
double recv[6] = {0, 0, 0, 0, 0, 0};
set_home_point (robot, joint);
get_home_point(robot, recv);
get_previous_pos(robot, recv);
```

4.11.18. 啟動關節座標軟體極限：

int enable_joint_soft_limit (HROBOT robot, bool enable)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
enable	bool	0: OFF 1: ON

回傳值	int	成功: 0 失敗: 2004
-----	-----	-------------------

4.11.19. 啟動卡式座標軟體極限：

int enable_cart_soft_limit (HROBOT robot, bool enable)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
enable	bool	0: OFF 1: ON
回傳值	int	成功: 0 失敗: 2004

4.11.20. 設定關節座標上下限：

int set_joint_soft_limit (HROBOT robot, double* low_limit, double* high_limit)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
low_limit	double*	下限各關節座標
high_limit	double*	上限各關節座標
回傳值	int	成功: 0 失敗: 2004
備註	<ul style="list-style-type: none"> ➤ 六軸機器人在各軸極限上可設定 A1~A6 軸的上下極限，SCARA 機器人在各軸極限上可設定 A1~A4 軸的上下極限。 ➤ 機器人在下達超過設定極限的運動命令，或運動過程中超過設定極限的位置時，即時發出警報並停止運動。 	

4.11.21. 設定卡式座標上下限：

int set_cart_soft_limit (HROBOT robot, double* low_limit, double* high_limit)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
low_limit	double*	下限各卡式座標
high_limit	double*	上限各卡式座標
回傳值	int	成功: 0 失敗: 2004
備註	<ul style="list-style-type: none"> ➤ 卡氏座標極限可設定 X、Y、Z 的上下極限 ➤ 機器人在下達超過設定極限的運動命令，或運動過程中超過設定極限的位置時，即時發出警報並停止運動。 	

	➤ 此極限範圍是基底座標為 0 (Base 0)情況下依據機器人的工具中心位置(TCP)進行設定。
--	---

4.11.22. 取得關節座標軟體極限設置：

int get_joint_soft_limit_config (**HROBOT** robot, **double*** low_limit, **double*** high_limit)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
enable	bool&	0: OFF 1: ON
low_limit	double*	下限各關節座標
high_limit	double*	上限各關節座標
回傳值	int	成功: 0 失敗: -1

4.11.23. 取得卡式座標軟體極限設置：

int get_cart_soft_limit_config (**HROBOT** robot, **bool&** enable, **double*** low_limit, **double*** high_limit)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
enable	bool&	0: OFF 1: ON
low_limit	double*	下限各卡式座標
high_limit	double*	上限各卡式座標
回傳值	int	成功: 0 失敗: -1

C++:

```
double joint_low_limit[6] = { -20, -20, -35, -20, 0, 0 };
double joint_high_limit[6] = { 20, 20, 0, 0, 0, 0 };
double cart_low_limit[6] = { -100, 300, -100, 0, 0, 0 };
double cart_high_limit[6] = { 100, 450, -25, 0, 0, 0 };
bool re_bool = false;

set_joint_soft_limit(device_id, joint_low_limit, joint_high_limit);
set_cart_soft_limit(device_id, cart_low_limit, cart_high_limit);
enable_joint_soft_limit(device_id, true);
enable_cart_soft_limit(device_id, true);
```

```
get_joint_soft_limit_config (device_id, re_bool, joint_low_limit, joint_high_limit);
get_cart_soft_limit_config (device_id, re_bool, cart_low_limit, cart_high_limit);
```

4.12. 通訊設定

4.12.1. 設定顯示訊息狀態：

int set_network_show_msg (**HROBOT** robot, **int** enable)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
enable	int	0: OFF 1: ON
回傳值	int	成功: 0 失敗: -1
備註	未把此函式設為 ON，會無法顯示收到的訊息。	

4.12.2. 取得顯示訊息狀態：

int get_network_show_msg (**HROBOT** robot, **int** enable)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
enable	int&	0: OFF 1: ON
回傳值	int	成功: 0 失敗: -1

4.12.3. 網路連線：

int network_connect (**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: -1

4.12.4. 網路中斷連線：

int network_disconnect (**HROBOT** robot)

參數	資料型態	說明
----	------	----

robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 失敗: -1

4.12.5. 傳送網路訊息：

int network_send_msg (**HROBOT** robot, **char*** msg)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
enable	char*	要發送的訊息，最後字串結尾需加上”%%” EX: str = “{test msg%%}”
回傳值	int	成功: 0 失敗: -1

4.12.6. 接收網路訊息：

int network_recieve_msg (**HROBOT** robot, **char*** msg)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
enable	char*	要接收的訊息
回傳值	int	成功: 0 失敗: -1
備註	<ul style="list-style-type: none"> ➤ 此函式無效，需使用 Callback notify 的方式來接收網路訊息。 ➤ 使用 Notify 需呼叫 get_current_position 才會開啟 Callback Notify 的功能。 ➤ 需持續呼叫 network_get_state 函式才會收到訊息。 ➤ 結尾會有”&&&” ➤ 詳細範例請參考 SampleCode 12.Network。 	

4.12.7. 設定網路配置：

int set_network_config (**HROBOT** robot, **int** connect_type, **char*** ip_addr, **int** port, **int** bracket_type, **int** separator_type, **bool** is_format)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
connect_type	int	設定連線狀態，Server 或是 Client 0: Server 1: Client
ip_addr	char*	連線目的 IP 地址

port	int	連線的閘道
bracket_type	int	<ul style="list-style-type: none"> ➤ 選擇括號包圍發送的資料。 ➤ 需使用與 HRSS 相同的括號來包圍資料，否則會無法顯示收到的資料。 0: {} 1: [] 2: () 3: <> 4: “?”在前
separator_type	int	<ul style="list-style-type: none"> ➤ 選擇符號來分隔資料，當使用機器手臂程式 HRB 指令 CREAD 接收網路訊息時，會以此符號分割資料內容。 0: , 1: _
is_format	bool	0: OFF 傳送資料以 ASCII 圖形顯示(英數字) 1: ON 傳送資料以 ASCII 16 進制顯示
回傳值	int	成功: 0 失敗: -1

4.12.8. 取得網路配置：

`int get_network_config (HROBOT robot, int& connect_type, char* ip_addr, int& port, int& bracket_type, int& separator_type, bool& is_format)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
connect_type	int&	取得設定連線狀態，Server 或是 Client 0: Server 1: Client
ip_addr	char*	取得連線目的 IP 地址
port	int&	取得連線的閘道
bracket_type	int&	取得選擇括號型態 0: {} 1: [] 2: () 3: <> 4: ? 在前
separator_type	int&	取得分隔符號 0: ,

		1: _
is_format	bool&	取得顯式格式 0: OFF 傳送資料以 ASCII 圖形顯示(英數字) 1: ON 傳送資料以 ASCII 16 進制顯示
回傳值	int	成功: 0 失敗: -1

4.12.9. 更改網路 IP 位址：

int network_change_ip (**HROBOT** robot, **int** lan_index, **int** ip_type, **char*** ip_addr)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
lan_index	int	設定第 N 個網路介面卡，數量與使用者電腦所擁有的網路介面卡相等。 0: LAN1 1: LAN2
ip_type	int	0: 動態 IP，由電腦自動取得 IP 位址。 1: 靜態 IP，使用者可以自行設定電腦 IP 位址。
ip_addr	char*	當使用靜態 IP，會根據此位址來設定
回傳值	int	成功: 0 失敗: -1

4.12.10. 取得連線狀態：

int network_get_state (**HROBOT** robot)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	連線: 1 未連線: 0
備註	<ul style="list-style-type: none"> ➤ 當起動 Notify 機制時，需持續呼叫此函式，Notify 才會收到訊息；接收的頻率為呼叫此函式才會收資料。 ➤ 要啟動 Notify 需先使用 get_current_position 才會開啟。 	

C++:

```
(1)----- set config
int server_type = 0; // socket server
char* ip = "127.0.0.1";
int port = 5123;
int bracket = 0; // {}
```

```

int separator = 0; // ,
bool is_format = false;
int show_msg = -1;
set_network_config(robot_id, client_type, ip, port, bracket, separator, is_format);
network_connect(robot_id);
get_network_show_msg(robot_id, show_msg);
if (!show_msg) {
    set_network_show_msg(robot_id, true);
}
char* msg = "test msg%% %";
network_send_msg (robot_id, msg)
network_disconnect (robot_id)

char set_ip = "192.168.0.25";
// network_change_ip(0, 0, set_ip);

(2) ----- get config
int re_type = 0; // socket server
char* re_ip = new char[20];
int re_port = 5123;
int re_bracket = 0; // { }
int re_separator = 0; // ,
bool re_is_format = false;
int show_msg = -1;
int connect_state = -1;
get_network_config(device_id, re_type, re_ip, re_port, re_bracket, re_separator, re_is_format);
for(int i=0;i<3; i++){
    connect_state = network_get_state(robot_id)
    Sleep(200);
}

```

5. 錯誤代碼

概要	說明	代碼
正常	命令正常完成	0000
未授權	授權失敗，請洽客服人員	0100
檔案錯誤	指定檔案路徑沒有檔案或是為無效檔案	200
檔案不存在	指定檔案路徑沒有檔案	201
無法執行	無法執行請求的命令	2000
參數錯誤	命令參數錯誤	2004
命令執行異常	命令執行發生異常	2005
無法接受命令	根據系統狀態不接受命令的執行	2006
字串長度超過	輸入的字串長度超過上限	2007
模式禁止	當下模式不接受命令的執行	4000
伺服禁止	此命令無法在非激磁狀態下執行	4001
運動暫存器禁止	運動暫存器數量到達1000筆，無法執行請求的命令，無法執行請求的命令	4003
檔案執行異常	RSR/PNS 任務設定異常	4010
	RSR/PNS 任務執行失敗	4011
	任務名稱錯誤或遺失	4012
	已有任務執行中	4013
功能異常	本功能異常	9999

6. 附錄

6.1. HRSDK 2.1.8 修改項目

修改項目	版本	版本	修改內容
	2.1.7	2.1.8	
1	Connect	open_connection	命名修改
2	Close	close_connection	命名修改
3	get_HRSDK_version	get_hrsdk_version	命名修改
4	get_DI	get_digital_input	命名修改
5	get_DO	get_digital_output	命名修改
6	set_DO	set_digital_output	命名修改
7	get_FI	get_function_input	命名修改
8	get_FO	get_function_output	命名修改
9	get_RI	get_robot_input	命名修改
10	get_RO	get_robot_output	命名修改
11	set_RO	set_robot_output	命名修改
12	get_VO	get_valve_output	命名修改
13	set_VO	set_valve_output	命名修改
14	get_HRSS_version	get_hrss_version	命名修改

6.2. HRSDK 2.1.8 新增項目

新增項目	命令名稱	功能說明
1	set_connection_level	設定連線等級
2	set_timer_start	啟動機器人計時器
3	set_timer_stop	停止機器人計時器
4	set_robot_id	設定機器人識別名稱
5	get_robot_id	取得機器人識別名稱
6	get_function_input	取得功能輸入
7	get_function_output	取得功能輸出
8	set_rsr	設定 RSR
9	remove_rsr	移除 RSR
10	get_rsr_prog_name	取得 RSR 程式名稱
11	get_execute_file_name	取得當前執行中程式檔案名稱
12	send_file	上傳 HRB 至控制器
13	download_file	下載 HRB 至本地端
14	get_hrss_mode	取得 HRSS 當前模式
15	update_hrss	更新 HRSS 版本
16	get_pr	取得位置暫存器數據
17	remove_pr	清除位置暫存器

6.3. HRSDK 2.1.9 新增項目

新增項目	命令名稱	功能說明
1	set_acc_time	設定加速度時間
2	get_acc_time	取得加速度時間
3	jog_home	吋動復歸
4	remove_command	移除運動命令佇列中單一命令
5	remove_command_tail	移除運動命令佇列中最新數個命令

6.4. HRSDK 2.1.10 新增項目

新增項目	命令名稱	功能說明
1	get_module_input_config	取得模組輸入設置
2	get_module_output_config	取得模組輸出設置
3	set_module_input_simulation	設定模組輸入模擬
4	set_module_input_value	設定模組輸入
5	set_module_input_start	設定模組輸入開始編號
6	set_module_input_end	設定模組輸入結束編號
7	set_module_input_comment	設定模組輸入註解
8	set_module_output_value	設定模組輸出
9	set_module_output_start	設定模組輸出開始編號
10	set_module_output_end	設定模組輸出結束編號
11	set_module_output_comment	設定模組輸出註解
12	set_pr_comment	設定位置暫存器註解
13	get_pr_comment	取得位置暫存器註解
14	get_timer_status	取得機器人計時器狀態
15	set_timer_name	設定機器人計時器名稱
16	get_timer_name	取得機器人計時器名稱

6.5. HRSDK 2.1.11 新增項目

新增項目	命令名稱	功能說明
1.	disconnect	close_connection 命名修改

6.6. HRSDK 2.2.3 新增項目

新增項目	命令名稱	功能說明
1	set_DI_simulation_Enable	設定輸入模擬
2	set_DI_simulation	設定輸入狀態
3	get_DI_simulation_Enable	取得輸入狀態
4	set_digital_input_comment	設定輸入註解
5	get_digital_input_comment	取得輸入註解
6	set_digital_output_comment	設定輸出註解
7	get_digital_output_comment	取得輸出註解
8	set_home_point	設定復歸位置
9	get_home_point	取得復歸位置
10	get_previous_pos	取得上次關機位置
11	get_user_alarm_setting_message	取得使用者自訂警報訊息
12	set_user_alarm_setting_message	設定使用者自訂警報訊息
13	get_prog_number	取得檔案數量
14	get_prog_name	取得檔案名稱
15	set_module_input_type	設定模組輸入型態
16	set_module_output_type	取得模組輸出型態
17	get_controller_time	取得控制器現在時間
18	get_payload_value	取得負載設定值
19	save_module_io_setting	保存模組設定
20	SyncOutput	運動中進行 DO 開關操作
21	get_counter_name	取得計數器名稱
22	set_counter_name	設定計數器名稱
23	set_language	設定語言
24	get_digital_setting	取得數位設置
25	set_digital_setting	設定數位設置
26	delete_file	刪除 Robot 運動檔案
27	delete_folder	刪除 Robot 運動檔案資料夾
28	new_folder	新增 Robot 運動檔案資料夾
29	file_rename	重新命名 Robot 運動檔案
30	file_drag	拖動 Robot 運動檔案
31	send_file	參數修改，刪除根目錄與操作選項
32	task_start	參數修改，所帶檔案名稱必須有副檔名".hrb"

6.7. HRSDK 2.2.4 修改項目

新增項目	命令名稱	功能說明
1.	set_digital_input_comment	[修改]去除 col 參數，並把 index 參數移至前面，index 範圍改為[1-48]
2.	set_digital_output_comment	[修改]去除 col 參數，並把 index 參數移至前面，index 範圍改為[1-48]
3.	enable_joint_soft_limit	[新增]啟動關節座標軟體極限
4.	enable_cart_soft_limit	[新增]啟動卡式座標軟體極限
5.	set_joint_soft_limit	[新增]設定關節座標上下限
6.	set_cart_soft_limit	[新增]設定卡式座標上下限
7.	get_joint_soft_limit_config	[新增]取得關節座標軟體極限設置
8.	get_cart_soft_limit_config	[新增]取得卡式座標軟體極限設置
9.	set_network_show_msg	[新增]設定顯示訊息狀態
10.	get_network_show_msg	[新增]取得顯示訊息狀態
11.	network_connect	[新增]網路連線
12.	network_disconnect	[新增]網路中斷連線
13.	network_send_msg	[新增]傳送網路訊息
14.	network_recieve_msg	[新增]接收網路訊息
15.	set_network_config	[新增]設定網路配置
16.	get_network_config	[新增]取得網路配置
17.	network_change_ip	[新增]更改網路 IP 位址
18.	network_get_state	[新增]取得連線狀態

全球子公司 / 研發中心

德國 歐芬堡

HIWIN GmbH
OFFENBURG, GERMANY
www.hiwin.de
www.hiwin.eu

日本 神戶・東京・名古屋・長野・ 東北・靜岡・北陸・廣島・ 福岡・熊本

HIWIN JAPAN
KOBE・TOKYO・NAGOYA・NAGANO・
TOHOKU・SHIZUOKA・HOKURIKU・
HIROSHIMA・FUKUOKA・KUMAMOTO, JAPAN
www.hiwin.co.jp

美國 芝加哥

HIWIN USA
CHICAGO, U.S.A.
www.hiwin.com

義大利 米蘭

HIWIN Srl
BRUGHERIO, ITALY
www.hiwin.it

瑞士 優納

HIWIN Schweiz GmbH
JONA, SWITZERLAND
www.hiwin.ch

捷克 布爾諾

HIWIN s.r.o.
BRNO, CZECH REPUBLIC
www.hiwin.cz

新加坡

HIWIN SINGAPORE
SINGAPORE
www.hiwin.sg

韓國 水原・昌原

HIWIN KOREA
SUWON・CHANGWON, KOREA
www.hiwin.kr

中國 蘇州

HIWIN CHINA
SUZHOU, CHINA
www.hiwin.cn

以色列 海法

Mega-Fabs Motion Systems, Ltd.
HAIFA, ISRAEL
www.mega-fabs.com

上銀科技股份有限公司

HIWIN TECHNOLOGIES CORP.

40852 台中市精密機械園區精科路7號

Tel: 04-23594510

Fax: 04-23594420

www.hiwin.tw

business@hiwin.tw