

Program.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices;
using System.Diagnostics;
using System.IO;

```

loadDllCSharp.cs

```

namespace HROBOT
{
    class Program
    {
        static void Main()
        {
            using (HROBOT hRobot = new HROBOT())
            {
                hRobot.Connect("192.168.0.3");
                hRobot.Move(100, 100);
                hRobot.Move(200, 200);
                hRobot.Move(300, 300);
                hRobot.Move(400, 400);
                hRobot.Move(500, 500);
                hRobot.Move(600, 600);
                hRobot.Move(700, 700);
                hRobot.Move(800, 800);
                hRobot.Move(900, 900);
                hRobot.Move(1000, 1000);
                hRobot.Move(1100, 1100);
                hRobot.Move(1200, 1200);
                hRobot.Move(1300, 1300);
                hRobot.Move(1400, 1400);
                hRobot.Move(1500, 1500);
                hRobot.Move(1600, 1600);
                hRobot.Move(1700, 1700);
                hRobot.Move(1800, 1800);
                hRobot.Move(1900, 1900);
                hRobot.Move(2000, 2000);
                hRobot.Move(2100, 2100);
                hRobot.Move(2200, 2200);
                hRobot.Move(2300, 2300);
                hRobot.Move(2400, 2400);
                hRobot.Move(2500, 2500);
                hRobot.Move(2600, 2600);
                hRobot.Move(2700, 2700);
                hRobot.Move(2800, 2800);
                hRobot.Move(2900, 2900);
                hRobot.Move(3000, 3000);
                hRobot.Move(3100, 3100);
                hRobot.Move(3200, 3200);
                hRobot.Move(3300, 3300);
                hRobot.Move(3400, 3400);
                hRobot.Move(3500, 3500);
                hRobot.Move(3600, 3600);
                hRobot.Move(3700, 3700);
                hRobot.Move(3800, 3800);
                hRobot.Move(3900, 3900);
                hRobot.Move(4000, 4000);
                hRobot.Move(4100, 4100);
                hRobot.Move(4200, 4200);
                hRobot.Move(4300, 4300);
                hRobot.Move(4400, 4400);
                hRobot.Move(4500, 4500);
                hRobot.Move(4600, 4600);
                hRobot.Move(4700, 4700);
                hRobot.Move(4800, 4800);
                hRobot.Move(4900, 4900);
                hRobot.Move(5000, 5000);
                hRobot.Move(5100, 5100);
                hRobot.Move(5200, 5200);
                hRobot.Move(5300, 5300);
                hRobot.Move(5400, 5400);
                hRobot.Move(5500, 5500);
                hRobot.Move(5600, 5600);
                hRobot.Move(5700, 5700);
                hRobot.Move(5800, 5800);
                hRobot.Move(5900, 5900);
                hRobot.Move(6000, 6000);
                hRobot.Move(6100, 6100);
                hRobot.Move(6200, 6200);
                hRobot.Move(6300, 6300);
                hRobot.Move(6400, 6400);
                hRobot.Move(6500, 6500);
                hRobot.Move(6600, 6600);
                hRobot.Move(6700, 6700);
                hRobot.Move(6800, 6800);
                hRobot.Move(6900, 6900);
                hRobot.Move(7000, 7000);
                hRobot.Move(7100, 7100);
                hRobot.Move(7200, 7200);
                hRobot.Move(7300, 7300);
                hRobot.Move(7400, 7400);
                hRobot.Move(7500, 7500);
                hRobot.Move(7600, 7600);
                hRobot.Move(7700, 7700);
                hRobot.Move(7800, 7800);
                hRobot.Move(7900, 7900);
                hRobot.Move(8000, 8000);
                hRobot.Move(8100, 8100);
                hRobot.Move(8200, 8200);
                hRobot.Move(8300, 8300);
                hRobot.Move(8400, 8400);
                hRobot.Move(8500, 8500);
                hRobot.Move(8600, 8600);
                hRobot.Move(8700, 8700);
                hRobot.Move(8800, 8800);
                hRobot.Move(8900, 8900);
                hRobot.Move(9000, 9000);
                hRobot.Move(9100, 9100);
                hRobot.Move(9200, 9200);
                hRobot.Move(9300, 9300);
                hRobot.Move(9400, 9400);
                hRobot.Move(9500, 9500);
                hRobot.Move(9600, 9600);
                hRobot.Move(9700, 9700);
                hRobot.Move(9800, 9800);
                hRobot.Move(9900, 9900);
                hRobot.Move(10000, 10000);
            }
        }
    }
}

```

loadDllCplusplus.cpp

```

#include <windows.h>
#include "..\..\..\HROBOT\HROBOT.h"
using namespace std;
HROBOT s;

void main()
{
    s = Connect("192.168.0.3");
    system("pause");
}

```

機器人軟體開發套件

HRSDK

使用手冊
User Manual

Program.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices;
using System.Diagnostics;
using System.IO;

```

loadDllCSharp.cs

```

namespace HROBOT
{
    class Program
    {
        static void Main()
        {
            using (HROBOT hRobot = new HROBOT())
            {
                hRobot.Connect("192.168.0.3");
                hRobot.Move(100, 100);
                hRobot.Move(200, 200);
                hRobot.Move(300, 300);
                hRobot.Move(400, 400);
                hRobot.Move(500, 500);
                hRobot.Move(600, 600);
                hRobot.Move(700, 700);
                hRobot.Move(800, 800);
                hRobot.Move(900, 900);
                hRobot.Move(1000, 1000);
                hRobot.Move(1100, 1100);
                hRobot.Move(1200, 1200);
                hRobot.Move(1300, 1300);
                hRobot.Move(1400, 1400);
                hRobot.Move(1500, 1500);
                hRobot.Move(1600, 1600);
                hRobot.Move(1700, 1700);
                hRobot.Move(1800, 1800);
                hRobot.Move(1900, 1900);
                hRobot.Move(2000, 2000);
                hRobot.Move(2100, 2100);
                hRobot.Move(2200, 2200);
                hRobot.Move(2300, 2300);
                hRobot.Move(2400, 2400);
                hRobot.Move(2500, 2500);
                hRobot.Move(2600, 2600);
                hRobot.Move(2700, 2700);
                hRobot.Move(2800, 2800);
                hRobot.Move(2900, 2900);
                hRobot.Move(3000, 3000);
                hRobot.Move(3100, 3100);
                hRobot.Move(3200, 3200);
                hRobot.Move(3300, 3300);
                hRobot.Move(3400, 3400);
                hRobot.Move(3500, 3500);
                hRobot.Move(3600, 3600);
                hRobot.Move(3700, 3700);
                hRobot.Move(3800, 3800);
                hRobot.Move(3900, 3900);
                hRobot.Move(4000, 4000);
                hRobot.Move(4100, 4100);
                hRobot.Move(4200, 4200);
                hRobot.Move(4300, 4300);
                hRobot.Move(4400, 4400);
                hRobot.Move(4500, 4500);
                hRobot.Move(4600, 4600);
                hRobot.Move(4700, 4700);
                hRobot.Move(4800, 4800);
                hRobot.Move(4900, 4900);
                hRobot.Move(5000, 5000);
                hRobot.Move(5100, 5100);
                hRobot.Move(5200, 5200);
                hRobot.Move(5300, 5300);
                hRobot.Move(5400, 5400);
                hRobot.Move(5500, 5500);
                hRobot.Move(5600, 5600);
                hRobot.Move(5700, 5700);
                hRobot.Move(5800, 5800);
                hRobot.Move(5900, 5900);
                hRobot.Move(6000, 6000);
                hRobot.Move(6100, 6100);
                hRobot.Move(6200, 6200);
                hRobot.Move(6300, 6300);
                hRobot.Move(6400, 6400);
                hRobot.Move(6500, 6500);
                hRobot.Move(6600, 6600);
                hRobot.Move(6700, 6700);
                hRobot.Move(6800, 6800);
                hRobot.Move(6900, 6900);
                hRobot.Move(7000, 7000);
                hRobot.Move(7100, 7100);
                hRobot.Move(7200, 7200);
                hRobot.Move(7300, 7300);
                hRobot.Move(7400, 7400);
                hRobot.Move(7500, 7500);
                hRobot.Move(7600, 7600);
                hRobot.Move(7700, 7700);
                hRobot.Move(7800, 7800);
                hRobot.Move(7900, 7900);
                hRobot.Move(8000, 8000);
                hRobot.Move(8100, 8100);
                hRobot.Move(8200, 8200);
                hRobot.Move(8300, 8300);
                hRobot.Move(8400, 8400);
                hRobot.Move(8500, 8500);
                hRobot.Move(8600, 8600);
                hRobot.Move(8700, 8700);
                hRobot.Move(8800, 8800);
                hRobot.Move(8900, 8900);
                hRobot.Move(9000, 9000);
                hRobot.Move(9100, 9100);
                hRobot.Move(9200, 9200);
                hRobot.Move(9300, 9300);
                hRobot.Move(9400, 9400);
                hRobot.Move(9500, 9500);
                hRobot.Move(9600, 9600);
                hRobot.Move(9700, 9700);
                hRobot.Move(9800, 9800);
                hRobot.Move(9900, 9900);
                hRobot.Move(10000, 10000);
            }
        }
    }
}

```

loadDllCplusplus.cpp

```

#include <windows.h>
#include "..\..\..\HROBOT\HROBOT.h"
using namespace std;
HROBOT s;

void main()
{
    s = Connect("192.168.0.3");
    system("pause");
}

```



工業4.0 最佳夥伴

INDUSTRIE 4.0 Best Partner



多軸機器人

Multi-Axis Robot

取放作業/組裝/整列與包裝/半導體/
光電業/汽車工業/食品業

- 關節式機器手臂
- 並聯式機器手臂
- 史卡拉機器手臂
- 晶圓機器人
- 電動夾爪
- 整合型電動夾爪
- 旋轉接頭



單軸機器人

Single-Axis Robot

高精密產業/半導體/
醫療自動化/FPD面板搬運

- KK, SK
- KS, KA
- KU, KE, KC



Torque Motor 回轉工作台

Torque Motor Rotary Table

醫療/汽車工業/工具機/產業機械

- RAB系列
- RAS系列
- RCV系列
- RCH系列



滾珠螺桿

Ballscrew

精密研磨/精密轉造

- Super S 系列 (高Dm-N值/高速化)
- Super T 系列 (低噪音/低振動)
- 微小型研磨級
- E2 環保潤滑模組
- R1 螺帽旋轉式
- Cool Type 節能溫控螺桿
- RD 高DN節能重負荷
- 滾珠花鍵



線性滑軌

Linear Guideway

精密機械/電子半導體/生技醫療

- 滾珠式—
HG重負荷型, EG低組裝, WE寬幅型,
MG微型, CG扭矩型
- 靜音式—
QH重負荷型, QE低組裝型,
QW寬幅型, QR滾柱型
- 其他—
RG滾柱型, E2自潤型, PG定位型,
SE金屬端蓋型, RC強化型



特殊軸承

Bearing

工具機產業/機械手臂

- 交叉滾柱軸承
- 滾珠螺桿軸承
- 精密線性軸承
- 軸承座



諧波減速機

DATOKER® Strain Wave Gear

機器人/自動化設備/半導體設備/工具機

- 調心輸入軸組合式 DSC-PO
- 調心輸入軸組件式 DSC-CO
- 中空輸入軸組合式 DSH-PH
- 中空輸入軸密封式 DSH-AH



AC 伺服馬達&驅動器

AC Servo Motor & Drive

半導體設備/包裝機/SMT機台/
食品業機台/LCD設備

- 驅動器—D1, D2T/D2T-LM, E1
- 伺服馬達—50W~2000W



醫療設備

Medical Equipment

醫療院所/復健中心/療養中心

- 下肢肌力訓練機
- 內視鏡扶持機器手臂



線性馬達平台

Linear Motor Stage

自動化搬運/AOI光學檢測/

精密加工/電子半導體

- 鐵心式線性馬達
- 無鐵心式線性馬達
- 棒狀線性馬達
- 平面馬達
- 空氣軸承定位平台
- X-Y平台
- 龍門系統
- 單軸線性馬達定位平台



力矩馬達&直驅馬達

Torque Motor &

Direct Drive Motor

工具機

- 力矩馬達—TM-2/IM-2, TMRW系列
- 檢測設備/機器人
- 直驅馬達—DMS, DMY, DMN, DMT系列

目 錄

<u>1.</u>	<u>產品介紹</u>	<u>10</u>
1.1.	功能說明	10
1.2.	使用軟/硬體需求	10
1.3.	軟體版本對應說明	11
<u>2.</u>	<u>匯入資料庫前置作業</u>	<u>12</u>
2.1.	C++ 方式說明	12
2.2.	C#方式說明	18
2.3.	Visual Basic(VB)方式說明	22
2.4.	連線等級說明	23
2.5.	操作模式說明	23
<u>3.</u>	<u>命令格式說明與參考範例</u>	<u>24</u>
3.1.	連線命令類	24
3.1.1.	建立連線- open_connection	25
3.1.2.	關閉連線- disconnect	25
3.1.3.	設定連線等級- set_connection_level	26
3.1.4.	取得連線等級- get_connection_level	26
3.1.5.	取得 HRSDK 版本號- get_hrsdk_version	26
3.1.6.	取得 HRSDK 連線版本- get_hrsdk_sdkver	27
3.1.7.	回饋訊息說明- callback_function	28
3.2.	暫存器命令類	33
3.2.1.	設定機器人計時器- set_timer	34
3.2.2.	取得機器人計時器- get_timer	34
3.2.3.	啟動機器人計時器- set_timer_start	34
3.2.4.	停止機器人計時器- set_timer_stop	35
3.2.5.	取得機器人計時器狀態- get_timer_status	35
3.2.6.	設定機器人計時器名稱- set_timer_name	36
3.2.7.	取得機器人計時器名稱- get_timer_name	36
3.2.8.	設定機器人計數器- set_counter	37
3.2.9.	取得機器人計數器- get_counter	37
3.2.10.	設定機器人計數器名稱- set_counter_name	37

3.2.11. 取得機器人計數器名稱- get_counter_name	38
3.2.12. 設定位置暫存器座標系種類- set_pr_type.....	38
3.2.13. 取得位置暫存器座標系種類- get_pr_type	39
3.2.14. 設定位置暫存器座標- set_pr_coordinate	39
3.2.15. 取得位置暫存器座標- get_pr_coordinate	39
3.2.16. 設定位置暫存器工具、基底座標- set_pr_tool_base	39
3.2.17. 取得位置暫存器工具、基底座標- get_pr_tool_base.....	41
3.2.18. 設定位置暫存器資料- set_pr	41
3.2.19. 取得位置暫存器值- get_pr.....	41
3.2.20. 清除位置暫存器值- remove_pr	42
3.2.21. 取得位置暫存器註解- get_pr_comment	43
3.2.22. 設定位置暫存器註解- set_pr_comment	43
3.3. 系統變數命令類.....	45
3.3.1. 設定加速度比例- set_acc_dec_ratio.....	46
3.3.2. 取得加速度比例- get_acc_dec_ratio.....	46
3.3.3. 設定加速度時間- set_acc_time.....	47
3.3.4. 取得加速度時間- get_acc_time	47
3.3.5. 設定點對點運動速度- set_ptp_speed	47
3.3.6. 取得點對點運動速度- get_ptp_speed	48
3.3.7. 設定直線運動速度- set_lin_speed.....	49
3.3.8. 取得直線運動速度- get_lin_speed	49
3.3.9. 設定整體速度- set_override_ratio	49
3.3.10. 取得整體速度- get_override_ratio	50
3.3.11. 設定機器人編號- set_robot_id.....	50
3.3.12. 取得機器人編號- set_robot_id.....	50
3.3.13. 設定運動平滑半徑- set_smooth_length.....	51
3.3.14. 取得錯誤代碼- get_alarm_code	51
3.3.15. 設定數位設置- set_digital_setting.....	52
3.3.16. 取得數位設置- get_digital_setting	54
3.3.17. 設定語言 - set_language.....	55
3.3.18. 取得控制器現在時間- get_controller_time	55
3.3.19. 設定使用者自訂警報訊息- set_user_alarm_setting_message	56
3.3.20. 取得使用者自訂警報訊息- get_user_alarm_setting_message	56
3.3.21. 取得外部軸一般設置- get_ext_axis_setting	57
3.3.22. 設定外部軸一般設置- set_ext_axis_setting	57
3.3.23. 取得外部軸進階設置- get_ext_axis_setting_advanced.....	58
3.3.24. 設定外部軸進階設置- set_ext_axis_setting_advanced.....	59
3.3.25. 外部軸零點校正- ext_mastering	60

3.3.26. 取得外部軸目前位置- get_current_ext_pos.....	61
3.3.27. 取得外部軸同異步模式- get_current_ext_mode.....	61
3.3.28. 回復出廠校正位置- mastering	62
3.3.29. 重置原點位置- calibration	62
3.4. 輸出輸入命令類.....	65
3.4.1. 取得輸入狀態- get_digital_input.....	68
3.4.2. 設定輸入模擬數值- set_DI_simulation_Enable	68
3.4.3. 設定輸入狀態- set_DI_simulation	68
3.4.4. 取得輸入模擬數值- get_DI_simulation_Enable.....	69
3.4.5. 取得輸出狀態- get_digital_output	69
3.4.6. 設定輸出狀態- set_digital_output	69
3.4.7. 設定輸入註解- set_digital_input_comment	70
3.4.8. 設定輸出註解- set_digital_output_comment	70
3.4.9. 取得輸入註解- get_digital_input_comment	70
3.4.10. 取得輸出註解- get_digital_output_comment.....	71
3.4.11. 取得機器人輸入- get_robot_input.....	71
3.4.12. 取得機器人輸出- get_robot_output	72
3.4.13. 設定機器人輸出- set_robot_output	72
3.4.14. 取得電磁閥輸出- get_valve_output.....	72
3.4.15. 設定電磁閥輸出- set_valve_output	73
3.4.16. 取得功能輸入狀態- get_function_input	73
3.4.17. 取得功能輸出狀態- get_function_output.....	74
3.4.18. 取得模組輸入設置- get_module_input_config	75
3.4.19. 取得模組輸出設置- get_module_output_config	76
3.4.20. 設定模組輸入模擬數值- set_module_input_config.....	76
3.4.21. 設定模組輸入- set_module_input_value	77
3.4.22. 設定模組輸入開始編號- set_module_input_start	77
3.4.23. 設定模組輸入結束編號- set_module_input_end	77
3.4.24. 設定模組輸入註解- set_module_input_comment	78
3.4.25. 設定模組輸出- set_module_output_value	78
3.4.26. 設定模組輸出開始編號- set_module_output_start.....	78
3.4.27. 設定模組輸出結束編號- set_module_output_end	79
3.4.28. 設定模組輸出註解- set_module_output_comment	79
3.4.29. 設定模組輸入型態- set_module_input_type	79
3.4.30. 設定模組輸出型態- set_module_output_type.....	80
3.4.31. 保存模組設定- save_module_io_setting.....	80
3.4.32. 運動中進行 DO 開關操作- SyncOutput	81
3.4.33. 取得 DI 範圍數值- get_DI_range	82

3.4.34. 取得 DI 範圍模擬數值- get_DI_sim_range	83
3.4.35. 取得 DI 範圍註解- get_DI_comment_range.....	83
3.4.36. 取得 DO 範圍數值- get_DO_range.....	84
3.4.37. 取得 DO 範圍註解- get_DO_comment_range	85
3.4.38. 取得 FI 全部數值- get_FI_all.....	85
3.4.39. 取得 FO 全部數值-get_FO_all	85
3.4.40. 取得 Timer 全部狀態- get_timer_status_all.....	86
3.4.41. 取得 Timer 全部數值- get_timer_value_all.....	86
3.4.42. 取得 Timer 範圍註解- get_timer_comment_range	86
3.4.43. 取得 Counter 全部數值- get_counter_value_all	87
3.4.44. 取得 Counter 範圍註解- get_counter_comment_range.....	87
3.4.45. 取得 Fieldbus Register SRW 範圍數值- get_fieldbus_rs_srw_range	87
3.4.46. 取得 Fieldbus Register SRR 範圍數值- get_fieldbus_rs_srr_range	88
3.4.47. 取得 Fieldbus Register 範圍參數- get_fieldbus_rs_parameter_range.....	88
3.4.48. 取得 Fieldbus Register 範圍註解- get_fieldbus_rs_comment_range	89
3.4.49. 取得 System 輸入全部數值- get_system_input_all.....	89
3.4.50. 取得 System 輸出全部數值- get_system_output_all.....	89
3.4.51. 取得模組輸入全部設定- get_MI_config_all	90
3.4.52. 取得模組輸出全部設定- get_MO_config_all.....	90
3.4.53. 取得模組輸入範圍註解- get_MI_comment_range	91
3.4.54. 取得模組輸出範圍註解- get_MO_comment_range	91
3.4.55. 取得 SI 範圍數值- get_SI_range	92
3.4.56. 取得 SI 範圍模擬數值- get_SI_sim_range.....	92
3.4.57. 取得 SO 範圍數值- get_SO_range	92
3.4.58. 取得 SI 範圍註解- get_SI_comment_range.....	93
3.4.59. 取得 SO 範圍註解- get_SO_comment_range.....	93
3.4.60. 取得 PR 範圍註解- get_PR_comment_array.....	94
3.4.61. 取得 RI 全部數值- get_RI_all	94
3.4.62. 取得 RO 全部數值- get_RO_all	94
3.4.63. 取得 VO 全部數值- get_VO_all.....	95
3.4.64. 設定 DI 多個數值- set_DI_array	95
3.4.65. 設定 DI 多個模擬數值- set_DI_sim_array.....	96
3.4.66. 設定 DO 多個數值- set_DO_array	96
3.4.67. 設定 Timer 多個數值- set_timer_value_array	96
3.4.68. 設定 Counter 多個數值- set_counter_array.....	97
3.4.69. 設定 SI 多個數值- set_SI_array.....	97
3.4.70. 設定 SI 多個模擬數值- set_SI_sim_array.....	97
3.4.71. 設定 SO 多個數值- set_SO_array	98

3.4.72. 設定 Fieldbus SRW 多個數值-set_fieldbus_srw_array.....	98
3.4.73. 設定 MO 多個數值-set_MO_array	98
3.4.74. 設定 VO 多個數值-set_VO_array.....	99
3.4.75. 設定 RO 多個數值-set_RO_array.....	99
3.5. 座標命令類	100
3.5.1. 設定基底號碼-set_base_number	100
3.5.2. 取得基底號碼-get_base_number.....	100
3.5.3. 定義基底座標-define_base.....	101
3.5.4. 取得基底座標-get_base_data	101
3.5.5. 設定工具號碼-set_tool_number	101
3.5.6. 取得工具號碼-get_tool_number	102
3.5.7. 定義工具座標-define_tool.....	102
3.5.8. 取得工具座標-get_tool_data.....	102
3.5.9. 工具座標多點校準-tool_calibration.....	104
3.5.10. 基底座標三點校準-base_calibration.....	105
3.6. 任務命令類	106
3.6.1. 設定 RSR-set_rsr	106
3.6.2. 取得 RSR 程式名稱-get_rsr_prog_name	107
3.6.3. 移除 RSR-remove_rsr.....	107
3.6.4. RSR/PNS 啟動外部觸發任務-ext_task_start	108
3.6.5. 啟動任務-task_start	108
3.6.6. 暫停任務-task_hold.....	110
3.6.7. 繼續任務-task_continue.....	110
3.6.8. 停止任務-task_abort	110
3.6.9. 取得目前執行任務名稱-get_execute_file_name	112
3.7. 檔案管理命令類.....	113
3.7.1. 下載 HIWIN Robot language 檔案-download_file	113
3.7.2. 上傳 HIWIN Robot language 檔案-send_file.....	114
3.7.3. 刪除 Robot 運動檔案-delete_file	114
3.7.4. 刪除 Robot 運動檔案資料夾-delete_folder.....	114
3.7.5. 新增 Robot 運動檔案資料夾-new_folder	115
3.7.6. 重新命名 Robot 運動檔案-file_rename	115
3.7.7. 拖動 Robot 運動檔案-file_drag	115
3.7.8. 取得檔案數量-get_prog_number	116
3.7.9. 取得檔案名稱-get_prog_name	116
3.8. 控制器命令類	117
3.8.1. 取得 HRSS 當前模式-get_hrss_mode.....	117

3.8.2. 伺服設定- set_motor_state	118
3.8.3. 取得伺服狀態- get_motor_state.....	118
3.8.4. 設定操作模式- set_operation_mode	118
3.8.5. 取得操作模式- get_operation_mode.....	119
3.8.6. 錯誤清除- clear_alarm	119
3.8.7. 更新 HRSS- update_hrss.....	120
3.8.8. 取得手臂資訊- get_robot_info.....	121
3.9. 吻動類	123
3.9.1. 吻動- jog	123
3.9.2. 吻動復歸- jog_home	124
3.9.3. 吻動停止- jog_stop	124
3.10. 運動命令類	125
3.10.1. 絶對座標位置點對點運動- ptp_pos	126
3.10.2. 絶對關節角度點對點運動- ptp_axis.....	126
3.10.3. 相對座標位置點對點運動- ptp_rel_pos.....	127
3.10.4. 相對關節角度點對點運動- ptp_rel_axis	127
3.10.5. 位置暫存器之點對點運動- ptp_pr	127
3.10.6. 絶對座標位置直線運動- lin_pos.....	128
3.10.7. 絶對關節角度直線運動- lin_axis	129
3.10.8. 相對座標位置直線運動- lin_rel_pos	129
3.10.9. 相對關節角度直線運動- lin_rel_axis	130
3.10.10. 位置暫存器之直線運動- lin_pr	130
3.10.11. 絶對座標位置圓弧運動- circ_pos.....	131
3.10.12. 關節座標位置圓弧運動- circ_axis.....	132
3.10.13. 位置暫存器之圓弧運動- circ_pr	133
3.10.14. 運動暫停- motion_hold	133
3.10.15. 運動繼續- motion_continue	134
3.10.16. 運動停止- motion_abort.....	134
3.10.17. 運動延遲- motion_delay.....	134
3.10.18. 設定運動命令編號- set_command_id	136
3.10.19. 取得運動命令編號- get_command_id	136
3.10.20. 取得運動命令佇列中命令個數- get_command_count.....	137
3.10.21. 取得目前運動狀態- get_motion_state.....	138
3.10.22. 移除運動命令佇列中單一命令- remove_command.....	138
3.10.23. 移除運動命令佇列中最新數個命令- remove_command_tail.....	138
3.10.24. 外部軸絕對關節角度點對點運動- ext_ptp_axis.....	139
3.10.25. 外部軸絕對座標位置點對點運動- ext_ptp_pos	139
3.10.26. 外部軸絕對關節角度直線運動- ext_lin_axis	140

3.10.27. 外部軸絕對座標位置直線運動- ext_lin_pos.....	141
3.10.28. 外部軸非同步關節角度點對點運動- ext_asyptp.....	142
3.10.29. 目標位置運動是否可到位- motion_reachable.....	143
3.10.30. 兩點運動是否可到位- motion_check_lin	143
3.11. 手臂資訊命令類.....	145
3.11.1. 取得目前編碼器數值- get_encoder_count.....	146
3.11.2. 取得目前關節座標- get_current_joint.....	146
3.11.3. 取得目前絕對座標- get_current_position	147
3.11.4. 取得目前轉速- get_current_rpm.....	147
3.11.5. 取得裝置出廠時間- get_device_born_date.....	147
3.11.6. 取得控制器開機時間- get_operation_time.....	148
3.11.7. 取得各軸馬達里程數- get_mileage	148
3.11.8. 取得各軸馬達累積里程數- get_total_mileage.....	149
3.11.9. 取得累積稼動率- get_utilization	149
3.11.10. 取得稼動率百分比- get_utilization_ratio	149
3.11.11. 取得馬達負載百分比- get_motor_torque	150
3.11.12. 取得 HRSS 版本號- get_hrss_version	150
3.11.13. 取得 HRSS 連線版本- get_hrss_sdkver.....	151
3.11.14. 取得機器人型號- get_robot_type.....	152
3.11.15. 設定復歸位置- set_home_point.....	152
3.11.16. 設定外部軸復歸位置- set_ext_home_point.....	152
3.11.17. 取得復歸位置- get_home_point	153
3.11.18. 取得外部軸復歸位置- get_ext_home_point	153
3.11.19. 取得上次關機位置- get_previous_pos	153
3.11.20. 取得上次外部軸關機位置- get_previous_extpos	153
3.11.21. 點位檢查- confirm_home_point.....	154
3.11.22. 啟動關節座標軟體極限- enable_joint_soft_limit	154
3.11.23. 啟動卡式座標軟體極限- enable_cart_soft_limit.....	155
3.11.24. 設定關節座標上下限- set_joint_soft_limit.....	155
3.11.25. 設定卡式座標上下限- set_cart_soft_limit.....	155
3.11.26. 取得關節座標軟體極限設置- get_joint_soft_limit_config	156
3.11.27. 取得卡式座標軟體極限設置- get_cart_soft_limit_config	156
3.11.28. 設定負載設置- set_payload_config	157
3.11.29. 取得負載設置- get_payload_config	157
3.11.30. 設定有效負載- set_payload_active	158
3.11.31. 取得有效負載- get_payload_active.....	158
3.11.32. 取得點位警報設置- get_home_warning_setting	159
3.11.33. 設定點位警報設置- set_home_warning_setting.....	160

3.11.34. 取得外部軸驅動器極限- get_ext_driver_limit	161
3.11.35. 設定外部軸驅動器極限- set_ext_driver_limit.....	162
3.11.36. 取得外部軸編碼器數值-get_ext_encoder.....	162
3.11.37. 取得手臂 DH 值- get_robot_dh	163
3.11.38. 取得齒輪比- get_gear_ratio.....	164
3.11.39. 取得手臂資訊- get_robot_data	165
3.12. 通訊設定類	166
3.12.1. 設定顯示訊息狀態- set_network_show_msg.....	166
3.12.2. 取得顯示訊息狀態- get_network_show_msg	166
3.12.3. 網路連線- network_connect.....	167
3.12.4. 網路中斷連線- network_disconnect	167
3.12.5. 傳送網路訊息- network_send_msg	167
3.12.6. 接收網路訊息- network_recieve_msg	167
3.12.7. 設定網路配置- set_network_config	168
3.12.8. 取得網路配置- get_network_config.....	168
3.12.9. 更改網路 IP 位址- network_change_ip	169
3.12.10. 取得連線狀態- network_get_state	170
4. 程式編輯回饋錯誤碼說明	172

[索引 174](#)

1. 產品介紹

1.1. 功能說明

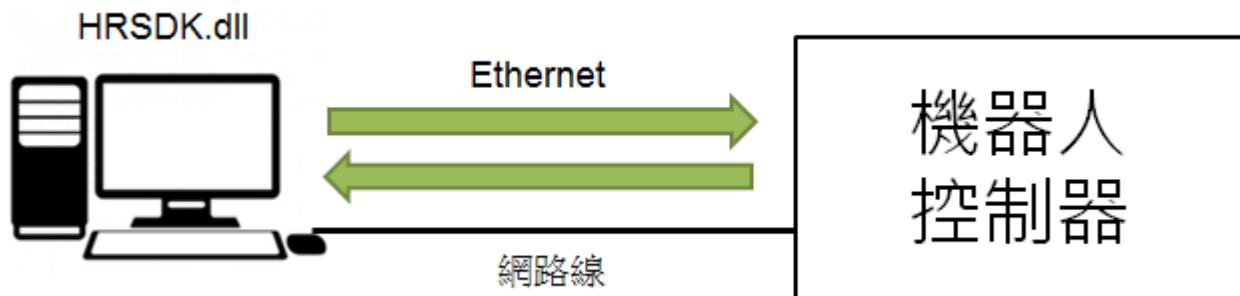


圖 1-1 簡易系統架構圖

HRSDK 為「HIWIN ROBOT Software Development Kit」之縮寫，中文名稱為上銀機器人軟體開發套件。利用上銀科技提供已製作完成的軟體資料庫工具，讓具備程式撰寫之使用者可作開發應用於自主的控制介面與程序來控制機器人，達成系統整合的目的。

- 可是適用機種為 六軸 GB、GC 系列控制器或 SCARA-LU 系列控制器，每個需對應不同的 HRSDK 資料庫與 HRSS 手臂版本。

1.2. 使用軟/硬體需求

- 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

1.3. 軟體版本對應說明

軟體版本對應表

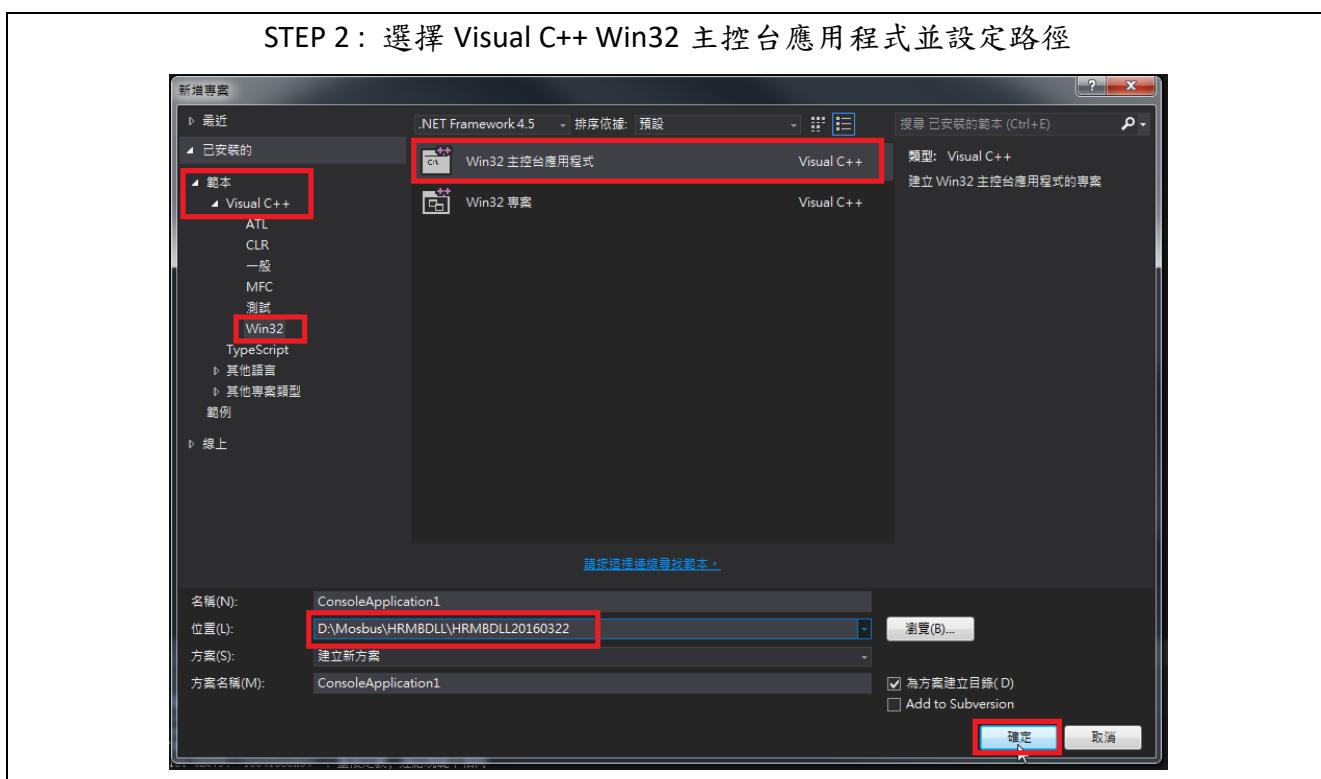
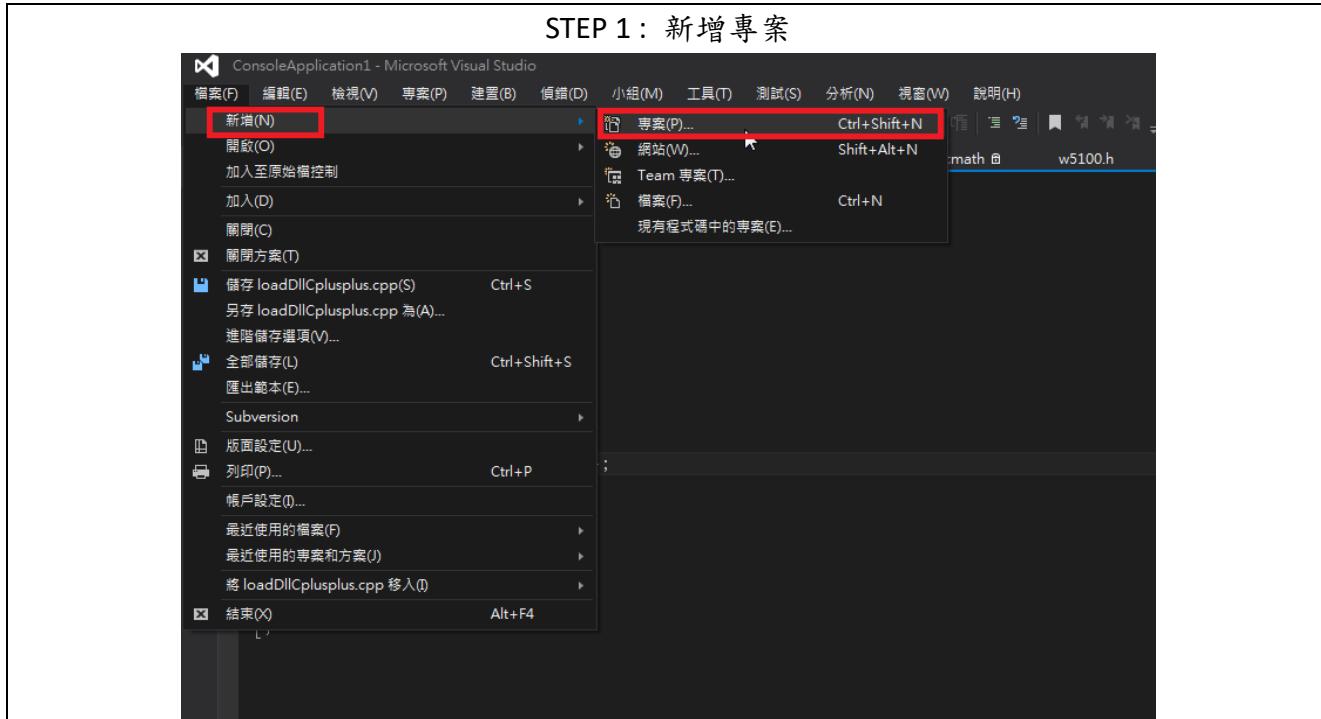
HRSDK 版本	適用 HRSS 軟 體版本	發布日期	HRSDK 版本	適用 HRSS 軟 體版本	發布日期
1.0.0	V2.1.23 以上 V3.1.6 以上	2017/02/06	2.2.9	V3.3.11	2020/10/23
2.0.0	V3.2.0 V3.2.1	2017/07/13	2.2.10	V3.3.12	2020/11/23
2.1.1	V3.2.2	2017/09/11	2.2.11	V3.3.13	2020/12/25
2.1.2	V3.2.5	2018/01/05	2.2.12	V3.3.14	2021/01/28
2.1.4	V3.2.5	2018/02/14	2.2.13	V3.3.15	2021/02/26
2.1.5	V3.2.8	2018/03/09	3.0.1	V3.3.16	2021/03/26
2.1.6	V3.2.11	2018/07/18	3.0.2	V3.3.18	2021/5/30
2.1.7	V3.2.13	2018/09/27	3.0.3	V3.3.19	2021/6/30
2.1.8	V3.2.15 以上	2019/05/15	3.0.4	V3.3.20	2021/09/01
2.1.9	V3.3.1	2019/09/09	3.0.5	V3.3.22	2022/2/01
2.1.10	V3.3.1	2019/09/09	3.0.6	V3.3.23	2022/4/1
2.1.11	V3.3.1	2019/09/09			
2.2.0	V3.3.2	2020/01/01			
2.2.2	V3.3.3	2020/02/27			
2.2.3	V3.3.3	2020/03/20			
2.2.4	V3.3.3	2020/04/20			
2.2.5	V3.3.7	2020/07/10			
2.2.6	V3.3.8	2020/07/22			
2.2.7	V3.3.9	2020/08/22			
2.2.8	V3.3.10	2020/09/22			

補充說明：

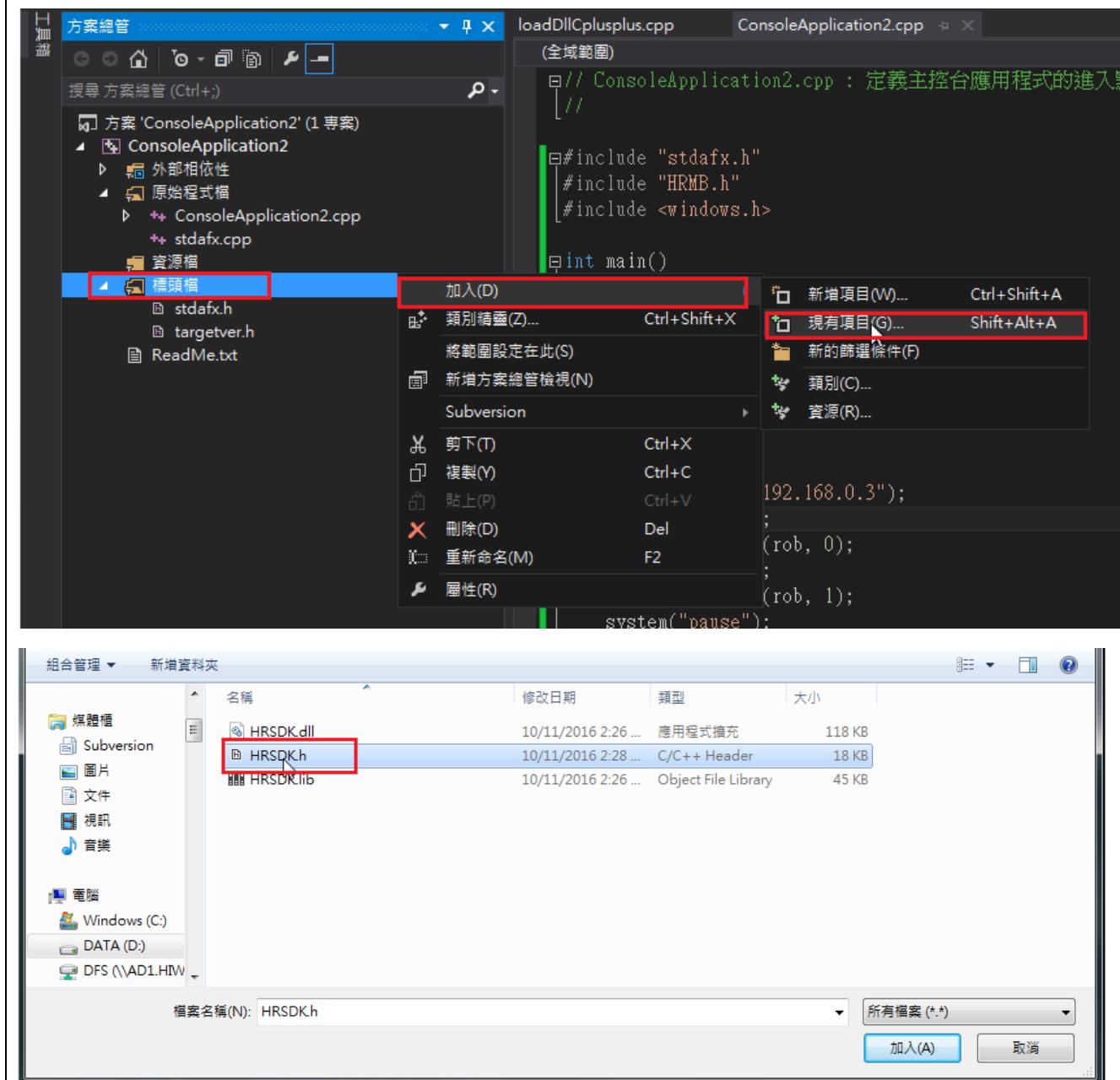
- HRSDK 軟體架構，因 DLL 設定需對應該版本問題，以上表顯示對應版本，
例如：HRSDK 2.2.11，需對應至 HRSS3.3.13 版本，以此類推。
- 在 HRSDK3.0.1 以上版本支援 HRSS 3.3.16 以上連線相容，也就是版本可以向下相容，
除非重大改版。
- 若手冊有更新會更新版本對應說明，若手冊無更新可先詢問原廠對應版本。

2. 匯入資料庫前置作業

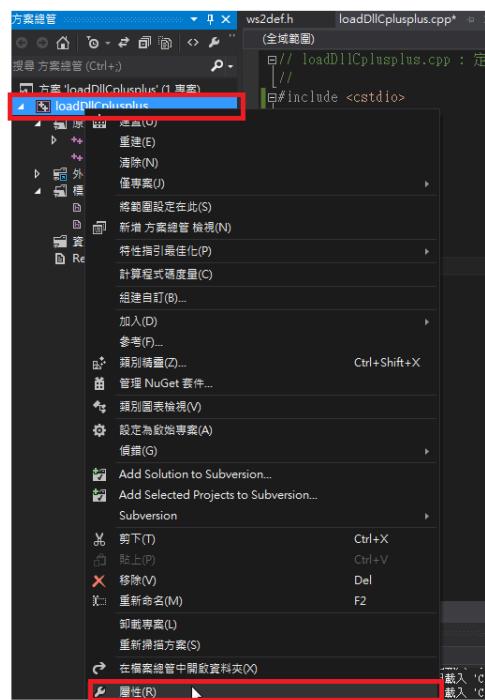
2.1. C++ 方式說明



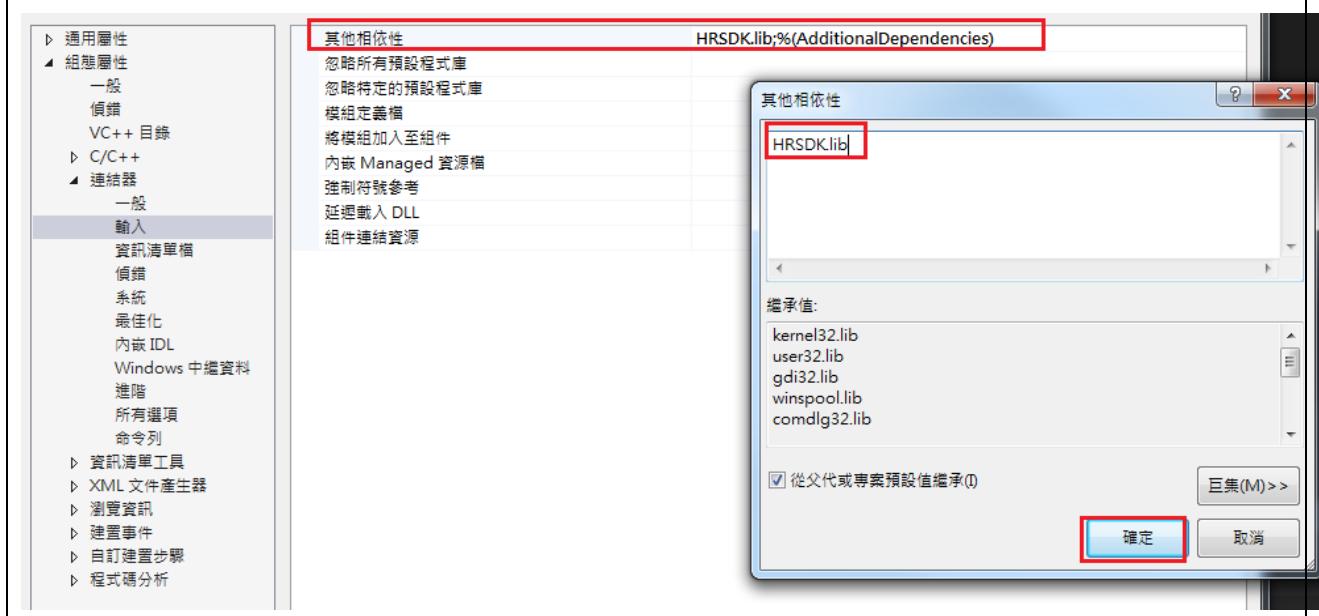
STEP 3：將 HRSDK.h 加入到標頭檔

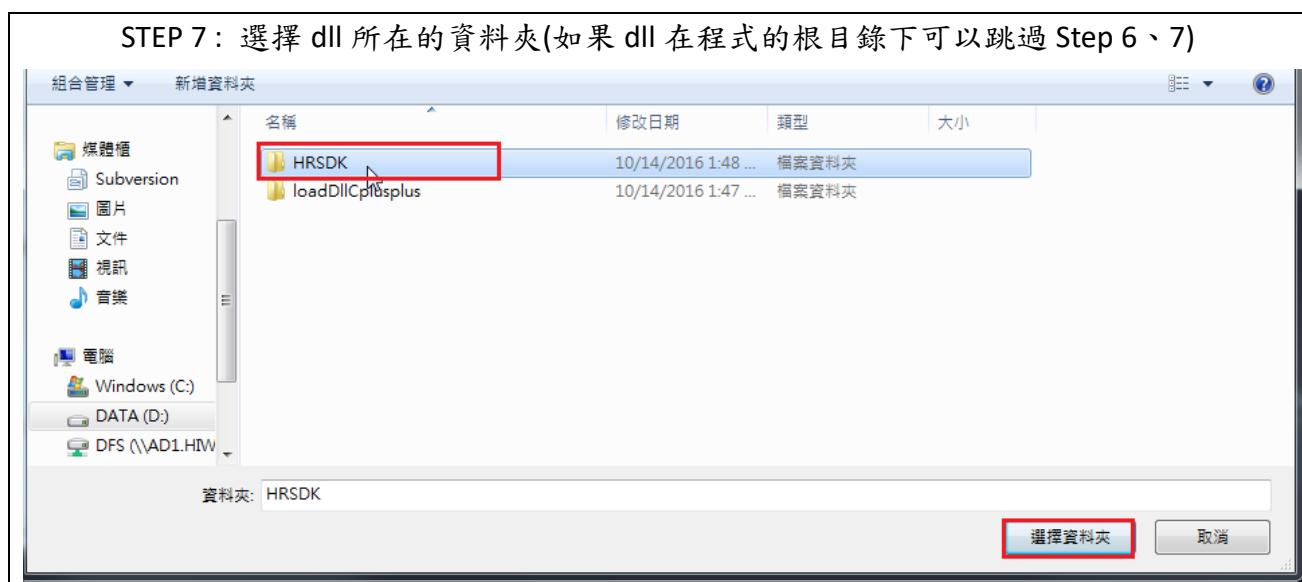


STEP 4：對專案右鍵選擇「屬性」

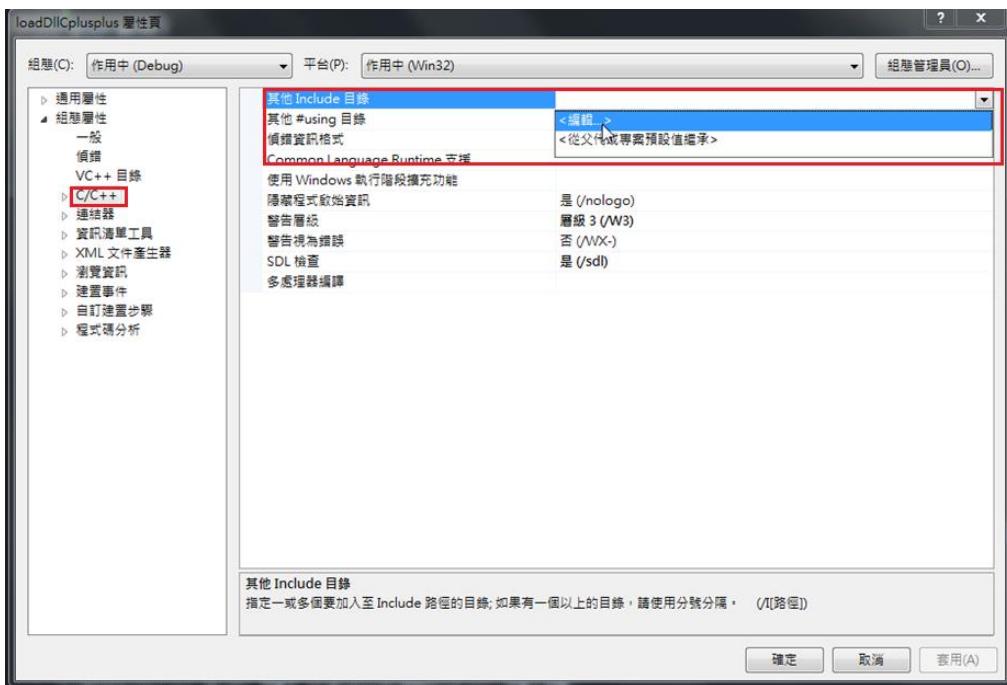


STEP 5：選擇連結器 -> 其他相依性 -> 輸入「HRSDK.lib」-> 確定

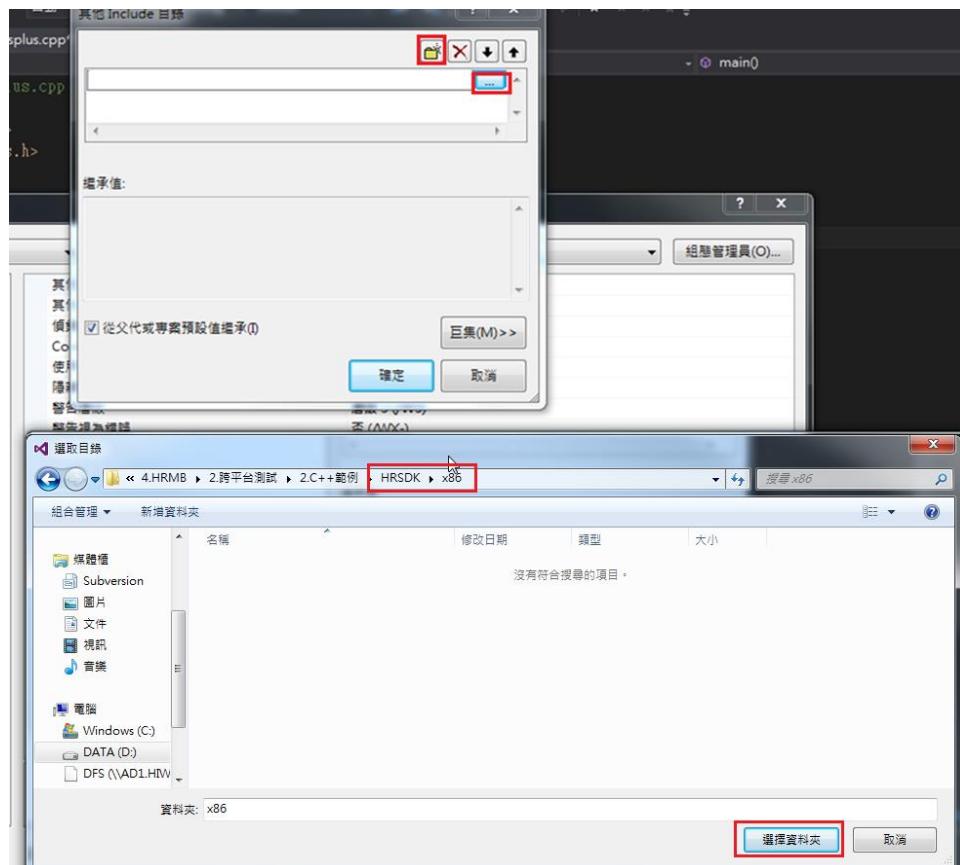




STEP 8 : C/C++ -> 其他 include 目錄 -> 編輯



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

The screenshot shows a code editor window with a dark theme. The title bar says 'loadDllCplusplus.cpp' and there is a '(全局範例)' button. The code in the editor is:

```
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#方式說明

遵照以下 5 個步驟，作 Visual Studio C# 程式開始撰寫的作業程序。

1. 從官網下載最新的 HRSDK 套件，需要有 HRSDK.dll 與 HRSDK.h 檔案。
 2. 開啟 Visual Studio C# 程式軟體，並創建一個檔案<專案名稱>.cs。
 3. 將 HRSDK.dll 複製放至專案路徑，並加入 HRobot.cs 檔案。
 4. 確定 namespace 與加入的檔案開頭名稱相同，若不同需更改。
 5. 開始使用 dll 資料庫(HRobot.XXX)。
- 路徑如下：`.../<專案名稱>/bin/Debug`
`(.../<專案名稱>/bin/Release)`



- 步驟 1：進入上銀官方網址：<http://www.hiwin.tw/>，點選欲下載之機器人型號，點選機器人軟體開發套件(HRSDK)軟體資料。

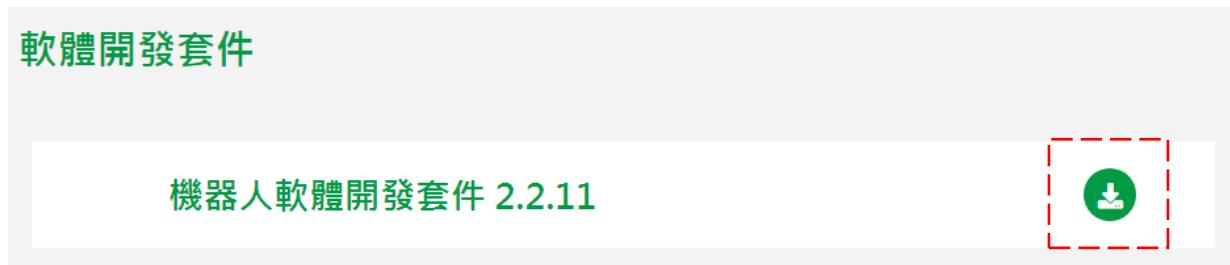


圖 2-1 下載軟體開發套件 HRSDK 資料圖

- 步驟 2：(1).開啟 Visual Studio C# 程式，開啟檔案(F) ->新增(N) -> 專案(P)。

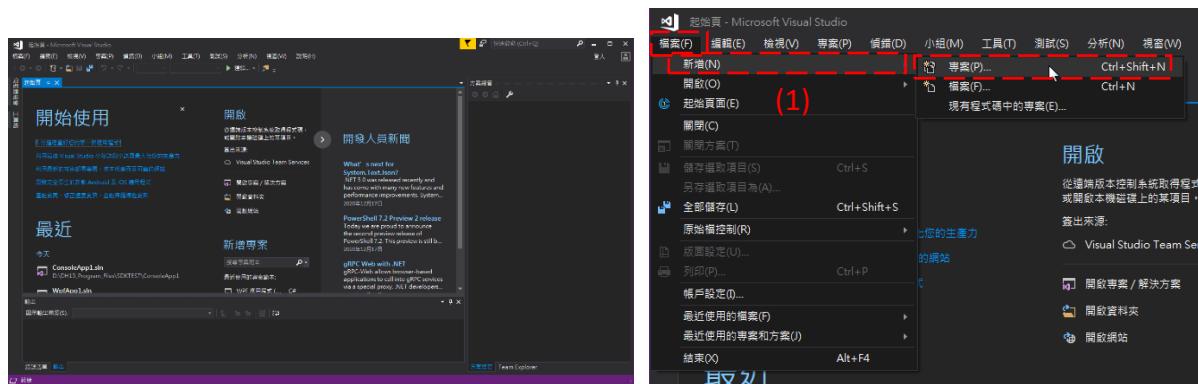


圖 2-2 開啟 Visual Studio C# 程式專案圖

- (2). 選擇主控台應用程式。(無介面只有文字顯示，若需介面選擇 Windows Forms App)。
- (3). 更改名稱與方案名稱->按「確定」創建檔案。

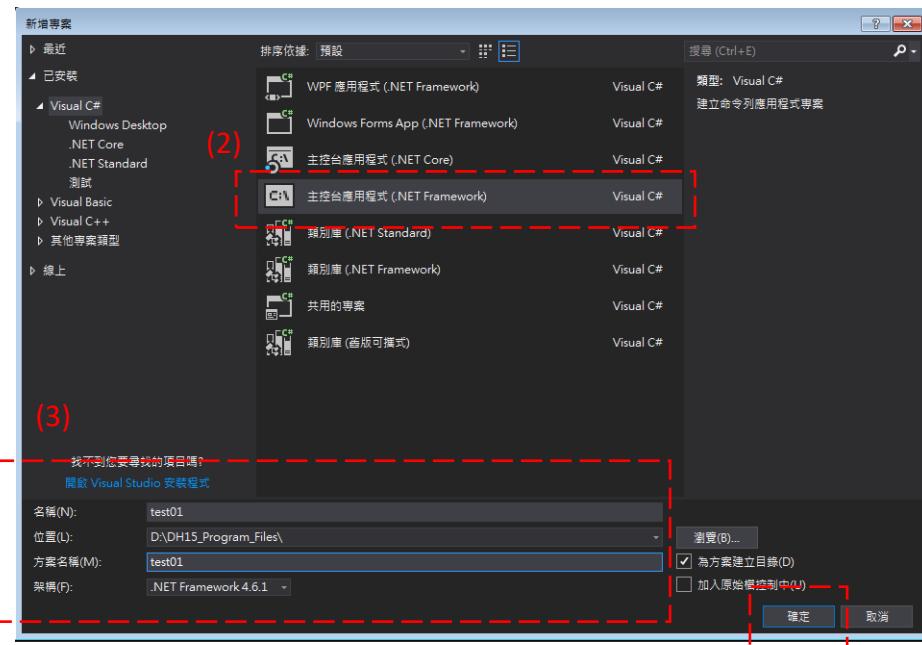


圖 2-3 主控台應用程式圖

■ 步驟 3：將下載的資料夾內複製 HRSDK.dll 和 HRobot.cs 檔案並放進路徑如下：

.../<專案名稱>/bin/Debug/

(.../<專案名稱>/bin/Release/)



圖 2-4 複製 HRSDK.dll & HRobot.cs 檔案至指定資料夾圖

■ 步驟 4：(1)專案(P) -> 加入現有項目(G) -> 選擇 HRobot.cs 檔案

(.../<專案名稱>/bin/Debug/)

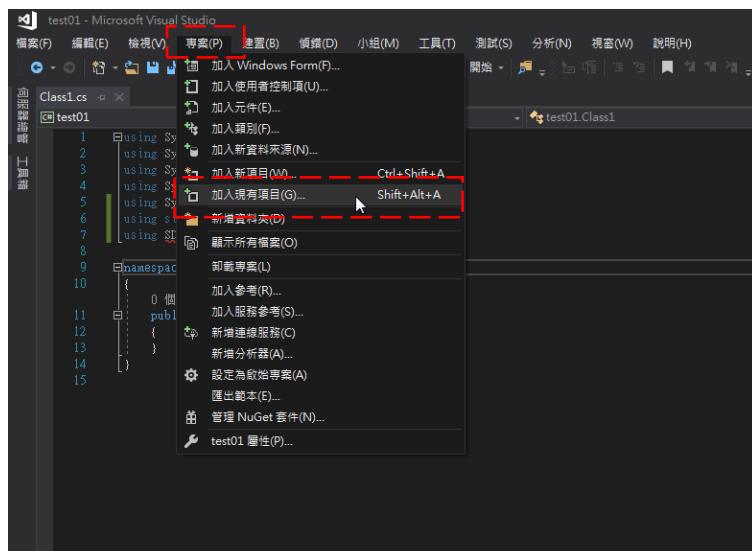


圖 2-5 加入 HRobot.cs 檔案圖

(2). 確認方案總管有 HRobot.cs 檔案。

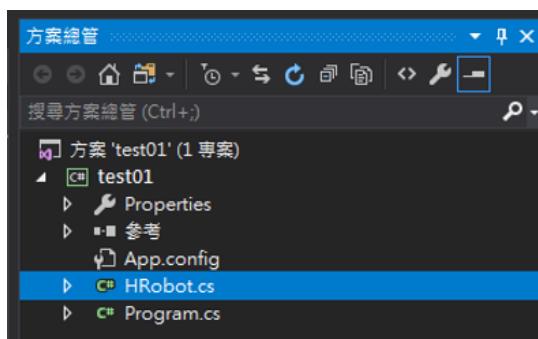


圖 2-6 確認方案總管有 HRobot.cs 檔案圖

(3). 打開 HRobot.cs 檔案，確認其開頭名稱為 SDKHrobot { }; (預設名稱)。

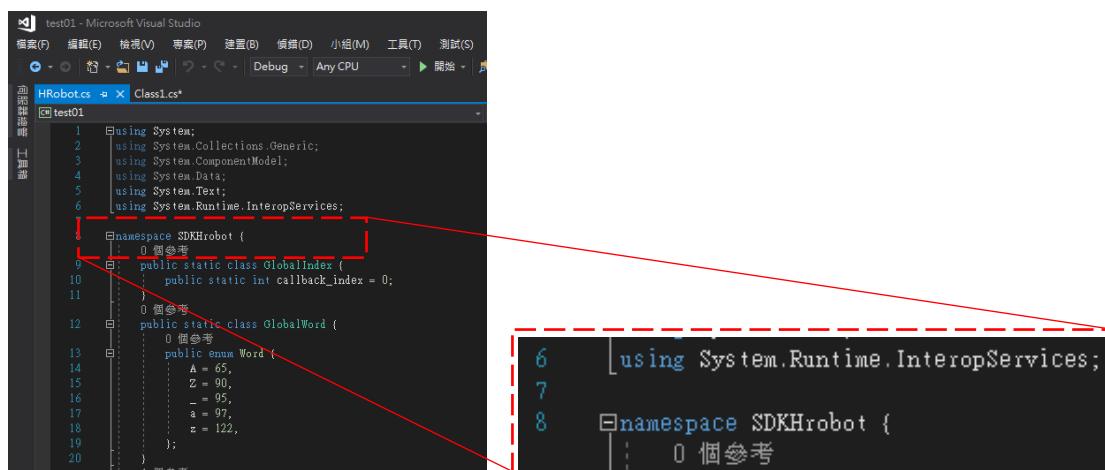


圖 2-7 確認其開頭名稱為 SDKHrobot 圖

- 步驟 5：(1). 程式編輯檔內 Program.cs 檔案，手動加入標頭檔 using SDKHrobot；
(2). 在主程式(Main)內，打上"HR"，測試是否有可以呼叫 HRobot 資料庫，若有代表已成功可以使用 HRSDK 資料庫。

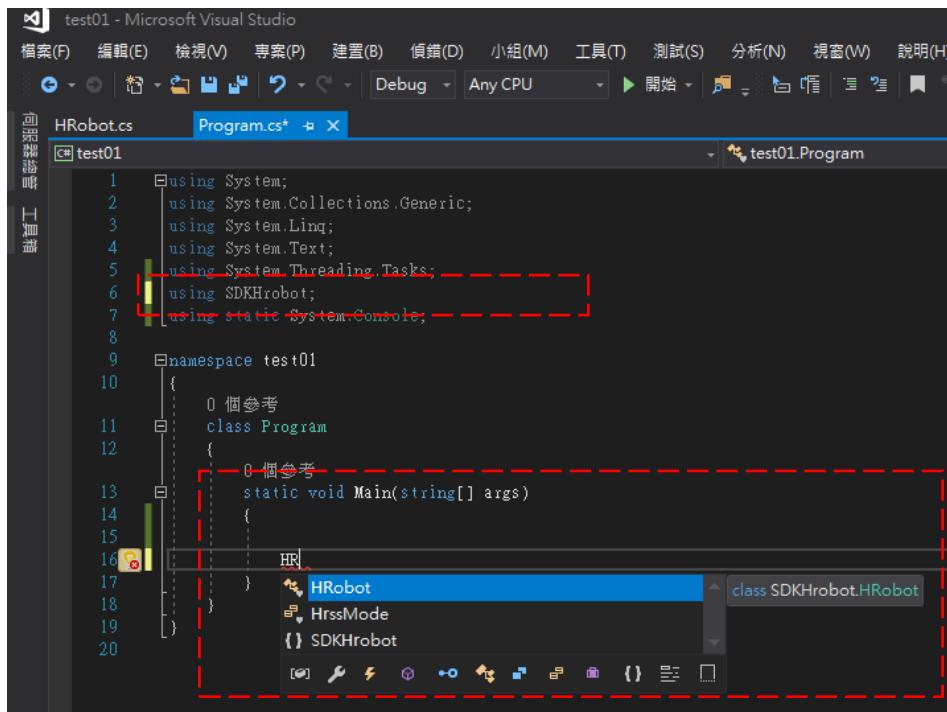


圖 2-8 Program.cs 檔案加入標頭檔圖

(3).鍵盤輸入一段範例程式，執行程式(F5)，測試是否能夠運作範例程式。

```
StringBuilder SDK_ver = new StringBuilder();
SDKHrobot.HRobot.get_hrsdk_version (SDK_ver);
Console.WriteLine(" SDK_ver= " + SDK_ver);
```

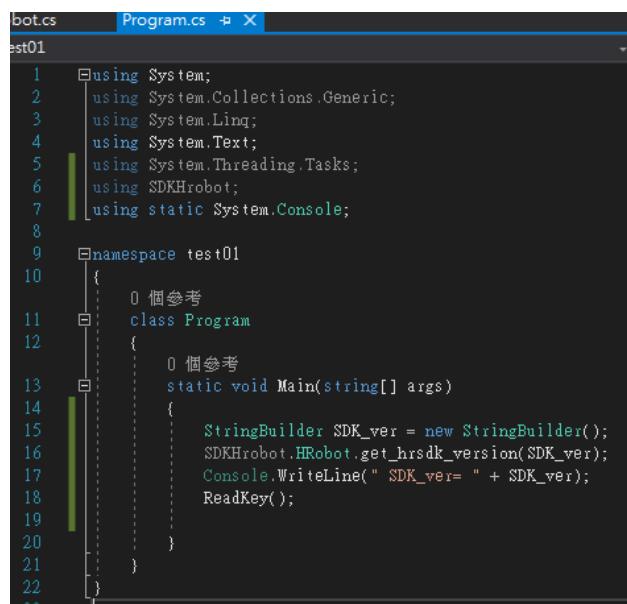
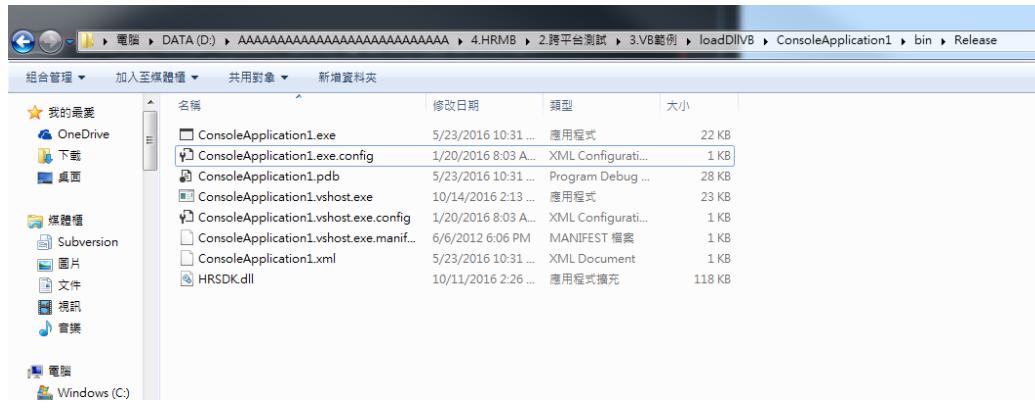


圖 2-9 輸入範例程式圖

2.3. Visual Basic(VB)方式說明

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 + X
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

```

2.4. 連線等級說明

連線等級分成兩種身分，觀察者與操作者。

- 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 4 個時連線。
- 操作者：能使用所有應用成式介面，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觀察者等級。

2.5. 操作模式說明

操作模式有兩個模式，手動模式與自動模式。

➤ 手動模式功能說明

- (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

手動模式注意事項說明：

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. 命令格式說明與參考範例

將所有命令根據不同功能作為 12 個分類：

分類項次	分類內容	對應章節
1	連線命令類	3.1
2	暫存器命令類	3.2
3	系統變數命令類	3.3
4	輸入輸出命令類	0
5	座標系命令類	3.5
6	任務命令類	3.6
7	檔案管理命令類	3.7
8	控制器設定命令類	3.8
9	吋動類	3.9
10	運動命令類	3.10
11	手臂資訊命令類	3.11
12	通訊設定類	3.12

以下各小節說明其分類指令與對應身分。

3.1. 連線命令類

函式名稱說明與對應身分參考下表：

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 HRSRK 版本號	○	○	○
get_hrss_sdkver	取得機器人 SDK 連線版本	○	○	○
get_hrsdk_sdkver	取得本地端 SDK 連線版本	○	○	○

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.1.1. 建立連線- open_connection

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 (0-3 為有效裝置 id) 失敗：-1 回傳機制創建失敗：-2 無法連至 Robot：-3 版本不相符：-4

- 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。

3.1.2. 關閉連線- disconnect

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
}
```

3.1.3. 設定連線等級- set_connection_level

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: 錯誤碼

C++:

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

3.1.4. 取得連線等級- get_connection_level

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);
```

3.1.5. 取得 HRSDK 版本號- get_hrsdk_version

void get_hrsdk_version (char* version)

參數	資料型態	說明
version	char*	HRSDK 版本號
備註	<ul style="list-style-type: none"> ➤ 此版本為釋出版本號碼。 ➤ 3.0.X 版本支援 HRSS 3.3.16 以上連線相容。 ➤ 請注意，新指令無法在舊版 SDK 使用。新功能請搭配新版 SDK 	

C++:

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

3.1.6. 取得 HRSDK 連線版本- get_hrsdk_sdkver

void get_hrsdk_sdkver (**int&** large_ver, **int&** small_ver, **int&** revision)

參數	資料型態	說明
large_ver	int	大版本連線號碼
small_ver	int	小版本連線號碼
revision	int	版本戳記
備註	<ul style="list-style-type: none"> ➤ 從 SDK 3.0.1 與 HRSS 3.3.16 開始，SDK 連線會依據此版本號碼與機器人連線配對，當與機器人的 SDK 版本不相符會有下列情況。 ➤ 大版本連線號碼不同則無法連線，需要更新 SDK 才可以使用。 ➤ 小版本連線號碼不同可以正常連線，會有新指令無法在舊版 SDK 使用的情況。 ➤ 版本戳記，無特殊功能。 	

C++:

```
int client_L=0, server_L=0;
int client_s, server_s=0;
int rev;
get_hrsdk_sdkver (client_L, client_s, rev);
get_hrss_sdkver (robot, server_L, server_s, rev);

if(client_L != server_L){
    printf("Large version does not match and can't be connected.");
}else if(client_s != server_s){
    printf("small version does not match.");
}
```

3.1.7. 回饋訊息說明- callback_function

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。 ⇒ 需執行 get_robot_info 才會開啟 DIO 與 Counter 的 Callback 觸發機制。可參考“11.CallbackNotify”範例程式。

命令 cmd	結果 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	4034	網路通訊訊息
0	4035	RS232 通訊訊息
0	4036	Modbus Monitor 訊息
0	4144	System Input 數值改變
0	4145	System Output 數值改變
0	4700	目前程式執行的行號
0	4702	當以下資訊改變，會回傳以下資訊
		hrss_mode HRSS 模式
		operation_mode 操作模式
		override_ratio 整體速度
		motor_state 駕服狀態
		exe_file_name 執行檔案名稱
		FO Function Output
		alarm_count alarm 數量
		keep_alive 保持連線
		motion_status 運動狀態

命令 cmd	結果 rlt	說明
		若是運動停止則會多加傳送以下資訊
	position coor joint encoder ext_pos ext_enc	座標是否改變 笛卡兒座標 軸座標 編碼值 外部軸位置 外部軸編碼值
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 變更 , off_on, index, mode, position
0	4711	Digital Input 數值改變
0	4712	Digital Output 數值改變
0	4713	Utilization 歷史紀錄資訊
0	4714	Utilization start , 當 task_start 開始執行
0	4715	Utilization end , 當 task_start 停止
0	4716	HRSS 傳給 SDK 的訊息
0	4717	PR 有哪些 Comment 有訊息
0	4718	外部軸感測器極限燈號顯示
13	0	連線已中斷
1410	4045	開始執行 Mastering zero position
1410	4046	完成 Mastering zero position
1411	4045	開始執行 Mastering calibration
1411	4046	完成 Mastering calibration
1450	4028	開始清除 alarm
1450	4029	結束清除 alarm
2162	4025	刪除 HRSS 備份檔案成功
2165	4044	下載 Caterpillar 失敗
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

命令 cmd	結果 rlt	說明
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
4151	0	手臂資訊
4202	0	TCP/IP 通訊連線成功
4202	9999	TCP/IP 通訊連線失敗
4290	4047	外部軸 Mastering 開始執行
4290	4048	外部軸 Mastering 結束
4290	4049	外部軸 Mastering 失敗，有外部軸正在移動中。
4290	4050	外部軸 Mastering 失敗。
4701	1000	API 指令解析失敗
4709	501	Module IO 儲存成功
63774	500	DIO 設置成功

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_WARNING
            } 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;
}
}
```

3.2. 暫存器命令類

函式名稱說明與對應身分參考下表：

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	O	O	×
get_timer	取得機器人計時器	O	O	O
set_timer_start	啟動機器人計時器	O	O	×
set_timer_stop	停止機器人計時器	O	O	×
get_timer_status	取得機器人計時器狀態	O	O	O
set_timer_name	設定機器人計時器名稱	O	O	×
get_timer_name	取得機器人計時器名稱	O	O	O
set_counter	設定機器人計數器	O	O	O
get_counter	取得機器人計數器	O	O	O
get_counter_name	取得機器人計數器名稱	O	O	O
set_counter_name	設定機器人計數器名稱	O	O	×
set_pr_type	設定位置暫存器座標型態	O	O	×
get_pr_type	取得位置暫存器座標型態	O	O	O
set_pr_coordinate	設定位置暫存器座標值	O	O	×
get_pr_coordinate	取得位置暫存器座標值	O	O	O
set_pr_tool_base	設定位置暫存器工具基底編號	O	O	×
get_pr_tool_base	取得位置暫存器工具基底編號	O	O	O
set_pr	設定位置暫存器值	O	O	×
get_pr	取得位置暫存器值	O	O	O
remove_pr	清空位置暫存器值	O	O	×
set_pr_comment	設定位置暫存器註解	O	O	×
get_pr_comment	取得位置暫存器註解	O	O	O

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.2.1. 設定機器人計時器- set_timer

`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: 錯誤碼

C++:

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

3.2.2. 取得機器人計時器- get_timer

`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: 錯誤碼

C++:

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

3.2.3. 啟動機器人計時器- set_timer_start

`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: 錯誤碼

C++:

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

3.2.4. 停止機器人計時器- set_timer_stop

`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: 錯誤碼

C++:

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

3.2.5. 取得機器人計時器狀態- get_timer_status

`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: 錯誤碼

C++:

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

3.2.6. 設定機器人計時器名稱- set_timer_name

`int set_timer_name(HROBOT robot, int index, wchar_t* name)`

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

C++:

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

3.2.7. 取得機器人計時器名稱- get_timer_name

`int get_timer_name(HROBOT robot, int index, wchar_t* name, int array_size)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Timer 的編號 (1-20)
name	wchar_t*	➤ 取得計時器名稱 ➤ 第一個值為註解長度
array_size	int	可設定想要取得的註解長度
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

```
int timer_index = 1;
wchar_t* name = L"This is Timer1.";
wchar_t re_name[20] = "";
int array_size = 10;
set_timer_name(robot, timer_index, name);
get_timer_name(robot, timer_index, re_name, array_size);
```

3.2.8. 設定機器人計數器- set_counter

`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: 錯誤碼

C++:

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

3.2.9. 取得機器人計數器- get_counter

`int get_counter(HROBOT robot, int index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Counter 的編號 (1-20)
回傳值	int	成功: 0 失敗: 錯誤碼

3.2.10. 設定機器人計數器名稱- set_counter_name

`int set_counter_name(HROBOT robot, int index, wchar_t* name)`

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

3.2.11. 取得機器人計數器名稱- get_counter_name

`int get_counter_name(HROBOT robot, int index, wchar_t* name, int array_size)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	Counter 的編號 (1-20)
name	wchar_t*	➤ 取得計數器名稱 ➤ 第一個值為註解長度
array_size	int	可設定想要取得的註解長度
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

```
int index = 5;
wchar_t* name = L"This is Counter5";
wchar_t str[20] = "";
int array_size = 10;
set_counter_name(robot, index, name);
get_counter_name(robot, index, str, array_size);
```

3.2.12. 設定位置暫存器座標系種類- set_pr_type

`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: 錯誤碼

3.2.13. 取得位置暫存器座標系種類- get_pr_type

```
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: 錯誤碼

3.2.14. 設定位置暫存器座標- set_pr_coordinate

```
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: 錯誤碼

3.2.15. 取得位置暫存器座標- get_pr_coordinate

```
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: 錯誤碼

3.2.16. 設定位置暫存器工具、基底座標- set_pr_tool_base

```
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

3.2.17. 取得位置暫存器工具、基底座標- get_pr_tool_base

```
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: 錯誤碼

3.2.18. 設定位置暫存器資料- set_pr

```
int set_pr(HROBOT robot ,int pr_num, int coor_type, double *coor, double *ext_pos,  
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 Value1~Value6
ext_pos	double[3]	欲設定的位置暫存器座標 Value7~Value9
tool	int	工具座標號碼
base	int	基底座標號碼
回傳值	int	成功: 0 失敗: 錯誤碼

3.2.19. 取得位置暫存器值- get_pr

```
int get_pr(HROBOT robot ,int pr_num, int* coor_type, double* coor, double *ext_pos,  
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 Value1~Value6
ext_pos	double[3]	欲設定的位置暫存器座標 Value7~Value9
tool	int	工具座標號碼
base	int	基底座標號碼
回傳值	int	成功:0 失敗: 錯誤碼

3.2.20. 清除位置暫存器值- remove_pr

`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: 錯誤碼
備註	comment 與 Value 同時清除	

3.2.21. 取得位置暫存器註解- get_pr_comment

```
int get_pr_comment(HROBOT robot,int pr_num , wchar_t* comment, int arr_size)
```

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	wchar_t*	➤ 取得位置暫存器註解的文字 ➤ 第一個值為註解長度
arr_size	int	可設定想要取得的註解長度
回傳值	int	成功:0 失敗: 錯誤碼

3.2.22. 設定位置暫存器註解- set_pr_comment

```
int set_pr_comment(HROBOT robot,int pr_num ,wchar_t* 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	wchar_t*	位置暫存器註解
回傳值	int	成功:0 失敗: 錯誤碼

C++:

```
(1)
double coor[6]={0,0,0,0,-90,0};
double ext_pos[3]={0,0,0}
set_pr(robot,2,1,coor,ext_pos,2,2); //set address register
                                         //address register number:2
                                         //coordinate type:joint
                                         //coordinate:{0,0,0,0,-90,0}
                                         //ext_pos:{0,0,0}
                                         //tool number:2
                                         //base number:2

int coor_type=-1;
coor[6]={0}
int tool = -1;
int base= -1;
int array_size = 10;
wchar_t* comment = L"This is a comment.";
```

```
wchar_t str[20] = "";
get_pr(robot,2 , &coor_type,coor,ext_pos,&tool,&base);
remove_pr(robot,2);
get_pr(robot,2 , &coor_type,coor,ext_pos,&tool,&base);
set_pr_comment(robot, 2, comment);
get_pr_comment(robot, 2, str, comment, array_size);
```

3.3. 系統變數命令類

函式名稱說明與對應身分參考下表：

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	×	O	×
get_acc_dec_ratio	取得加速度比例	O	O	O
set_acc_time	設定加速度時間	×	O	×
get_acc_time	取得加速度時間	O	O	O
set_ptp_speed	設定點對點移動速度	×	O	×
get_ptp_speed	取得點對點移動速度	O	O	O
set_lin_speed	設定直線移動速度	×	O	×
get_lin_speed	取得直線移動速度	O	O	O
set_override_ratio	設定進給速度倍率	O	O	×
get_override_ratio	取得進給速度倍率	O	O	O
set_robot_id	設定機器人識別名稱	O	O	×
get_robot_id	取得機器人識別名稱	O	O	O
set_smooth_length	設定運動平滑半徑	×	O	×
get_alarm_code	取得警報代碼	O	O	O
set_digital_setting	設定數位設置	O	O	×
get_digital_setting	取得數位設置	O	O	O
set_language	設定語言	O	O	O
get_controller_time	取得控制器現在時間	O	O	O
set_user_alarm_setting_message	設定使用者自訂警報訊息	O	O	×
get_user_alarm_setting_message	取得使用者自訂警報訊息	O	O	O
get_ext_axis_setting	取得外部軸一般設置	O	O	O
set_ext_axis_setting	設定外部軸一般設置	O	O	×
get_ext_axis_setting_advanced	取得外部軸進階設置	O	O	O
set_ext_axis_setting_advanced	設定外部軸進階設置	O	O	×
ext_mastering	外部軸零點校正	O	O	×
get_current_ext_pos	取得外部軸座標	O	O	O
get_current_ext_mode	取得外部軸同異步模式	O	O	O

函式名稱	說明	操作者		觀察者
		手動模式	自動模式	
mastering	回復出廠校正位置	○	○	×
calibration	重置原點位置	○	○	×
download_history_zip	下載歷史記錄	○	○	○

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.3.1. 設定加速度比例- `set_acc_dec_ratio`

`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: 錯誤碼

3.3.2. 取得加速度比例- `get_acc_dec_ratio`

`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: 錯誤碼

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

3.3.3. 設定加速度時間- set_acc_time

`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: 錯誤碼
備註	<ul style="list-style-type: none"> ● 設定速度指令會在核心運動進行排程，因此立刻取得的數值不一定會是剛才設定的數值，但是後續的運動指令會以設定的速度執行。 ● 若是改變自動/手動模式，速度設定被改變。 	



1. 使用 SET_ACC 指令可使機器手臂獲得更高的運行度，但是過低的設定值，可能會超出機器手臂的負載，導致發生錯誤警報。
2. 請根據實際使用條件，適當地調整參數，避免因過高的運行速度造成設備毀損。
3. 建議使用 set_acc_dec_ratio。

3.3.4. 取得加速度時間- get_acc_time

`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: 錯誤碼

3.3.5. 設定點對點運動速度- set_ptp_speed

`int set_ptp_speed(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: 錯誤碼

3.3.6. 取得點對點運動速度 - get_ptp_speed

`int get_ptp_speed(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: 錯誤碼

C++:

```
set_ptp_speed (robot,50);
vel = get_ptp_speed (robot);
```

3.3.7. 設定直線運動速度- set_lin_speed

`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: 錯誤碼
備註	<ul style="list-style-type: none"> ● 設定速度指令會在核心運動進行排程，因此立刻取得的數值不一定會是剛才設定的數值，但是後續的運動指令會以設定的速度執行。 ● 若是改變自動/手動模式，速度設定被改變。 	

3.3.8. 取得直線運動速度- get_lin_speed

`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: 錯誤碼

C++:

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

3.3.9. 設定整體速度- set_override_ratio

`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: 錯誤碼

3.3.10. 取得整體速度- get_override_ratio

```
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: 錯誤碼

C++:

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

3.3.11. 設定機器人編號- set_robot_id

```
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: 錯誤碼

C++:

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

3.3.12. 取得機器人編號- set_robot_id

```
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: 錯誤碼

C++:

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

3.3.13. 設定運動平滑半徑- set_smooth_length

`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: 錯誤碼

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

3.3.14. 取得錯誤代碼- get_alarm_code

`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 ➤ 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: 錯誤碼

C++:

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

2021/01/19_14:40:48_Err01-03-30
XY coor overlimit

// alarm_code: 0001033000000000
拆解成0001 0330 0000 0000
0001: 後兩碼對應HRSS第一分類
0330: 對應第二(03)與第三分類(30)
0000: 預備用
0000: 預備用

3.3.15. 設定數位設置- set_digital_setting

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

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int[18]	index 長度為 18 的陣列 ➤ index [0]: Clear Error DI: 0 SI: 1 ➤ index[1]: I/O 索引值 Disable: 0 DI: 1~48 SI: 1~256 ➤ index [2]: External Alarm DI: 0 SI: 1 ➤ index[3]: I/O 索引值 Disable: 0 DI: 1~48 SI: 1~256 ➤ index [4]: System Shutdown DI: 0 SI: 1 ➤ index[5]: I/O 索引值 Disable: 0

參數	資料型態	說明
		DI: 1~48 SI: 1~256 ➤ index [6]: Motor Warning DO: 0 SO: 1 ➤ index[7]: I/O 索引值 Disable: 0 DO: 1~48 SO: 1~256 ➤ index [8]: System StartUp DO: 0 SO: 1 ➤ index[9]: I/O 索引值 Disable: 0 DO: 1~48 SO: 1~256 ➤ index [10]: Mode Output DO: 0 SO: 1 ➤ index[11]: I/O 索引值 Disable: 0 DO: 1~48 SO: 1~256 ➤ index [12]: Reset Driver DO: 0 SO: 1 ➤ index[13]: I/O 索引值 Disable: 0 DO: 1~48 ➤ index [14]: Emergency Output DO: 0 ➤ index[15]: I/O 索引值 Disable: 0 DO: 1~48 ➤ index [16]: Reset Safety Relay DO: 0 ➤ index[17]: I/O 索引值

參數	資料型態	說明
		Disable: 0 DO: 1~48
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: 錯誤碼

3.3.16. 取得數位設置 - get_digital_setting

`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[19]	* 所設定的 DIO/SIO 來觸發 Clear Error 等指令。 ➤ index[0,2,4,6,8,10,12,1,16] 0: DIO 1:SIO ➤ index[1,3,5,7,9,11,13,15,17] 0:禁用此功能 DIO 範圍 1~48 SIO 範圍 1~256 ➤ index[18] 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: 錯誤碼

C++:

```

int DIO = 0;
int SIO = 1;
int index[18] = { DIO, 36, DIO, 37, SIO, 38, SIO, 39, DIO, 40, DIO, 41,
                  DIO, 42, DIO, 43, DIO, 44 };
int recv[19] = {0};
char* text = "External Alarm";
set_digital_setting (robot, index, text);
get_digital_setting (robot, recv, text);

```

3.3.17. 設定語言 - set_language

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 4:韓文
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

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

3.3.18. 取得控制器現在時間 - get_controller_time

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);
```

3.3.19. 設定使用者自訂警報訊息- set_user_alarm_setting_message

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: 錯誤碼

3.3.20. 取得使用者自訂警報訊息- get_user_alarm_setting_message

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);
```

3.3.21. 取得外部軸一般設置 - get_ext_axis_setting

```
int get_ext_axis_setting(HROBOT robot, int index, bool& enable, int& mode, double&
high_limit, double& low_limit)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	[0-2] 索引值
enable	bool&	0: Off 1: On
mode	int&	0: Sync 1: Async
high_limit	double&	上極限
low_limit	double&	下極限
回傳值	int	成功: 0 失敗: 錯誤碼

3.3.22. 設定外部軸一般設置 - set_ext_axis_setting

```
int set_ext_axis_setting (HROBOT robot, int index, bool enable, int mode, double high_limit,
double low_limit)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	[0-2] 索引值
enable	bool	0: Off 1: On
mode	int	0: Sync 1: Async
high_limit	double	上極限
low_limit	double	下極限
回傳值	int	成功: 0 參數範圍錯誤: -1 同步/非同步模式錯誤: -2 設定同步旋轉參數失敗: -3 設定同步線性參數失敗: -4 失敗: 錯誤碼
備註	<ul style="list-style-type: none"> ➤ 必須使用此 API, set_ext_axis_setting_advanced 才會保存設定。 ➤ 使用前需先設定過 set_ext_axis_setting_advanced，否則會因為 HRSS 初始預設值導致設定參數失敗。 	

3.3.23. 取得外部軸進階設置 - `get_ext_axis_setting_advanced`

```
int get_ext_axis_setting_advanced (HROBOT robot, int index, int& type, bool& math, bool&
continuous, int* int_value, double* double_value)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	[0-2] 索引值
type	int&	0: Linear 1: Rotary
math	bool&	0: Off 1: On
continuous	bool&	0: Off 1: On
int_value	int[3]	此陣列包含三個空間 ➤ int_value[0]: 馬達方向 0: Positive 1: Negative ➤ int_value[1]: 馬達最大轉速 [rpm] ➤ int_value[2]: 馬達解析度 [pls/rev]
double_value	double[3]	此陣列包含三個空間 ➤ double_value[0]: 加速度時間 [ms] 範圍: [20-1000] ➤ double_value[1]: 馬達裝配減速機的減速比 [mot-rev/axs-rev] ➤ double_value[2]: 導程 (Pitch) [mm/axs-rev]
回傳值	int	成功: 0 失敗: 錯誤碼

3.3.24. 設定外部軸進階設置 - set_ext_axis_setting_advanced

```
int set_ext_axis_setting_advanced (HROBOT robot, int index, int type, bool math, bool
continuous, int* int_value, double* double_value)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	[0-2] 索引值
type	int	0: Linear 1: Rotary
math	bool	0: Off 1: On
continuous	bool	0: Off 1: On
int_value	int[3]	此陣列包含三個空間 ➤ int_value[0]: 馬達方向 0: Positive 1: Negative ➤ int_value[1]: 馬達最大轉速 [rpm] ➤ int_value[2]: 馬達解析度 [pls/rev]
double_value	double[3]	此陣列包含三個空間 ➤ double_value[0]: 加速度時間 [ms] 範圍: [20-1000] ➤ double_value[1]: 馬達裝配減速機的減速比 [mot-rev/axs-rev] ➤ double_value[2]: 導程 (Pitch) [mm/axs-rev]
回傳值	int	成功: 0 失敗: 錯誤碼
備註	必須使用 set_ext_axis_setting 才會保存此設定	

3.3.25. 外部軸零點校正- ext_mastering

int ext_mastering (HROBOT robot , int index)

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	[0-2]索引值
回傳值	int	成功: 0 編碼器正在校正: 2 外部軸沒開啟: 3 不支援 Offline: 100 失敗: 錯誤碼

C++:

```

int index= 0;
bool enable = true;
int mode = 0;
double high_limit = 50;
double low_limit = -10;
int type = 0;
bool math = false;
bool continuous = false;
int value[3] = {5000, 500, 500};
double value[3] = {100, 100, 100};
set_ext_axis_setting_advanced (robot , int index, type, math, continuous,   int_value,
double_value)

set_ext_axis_setting (robot , index, enable, mode, high_limit, low_limit);

ext_mastering(robot, index);

```

3.3.26. 取得外部軸目前位置 - get_current_ext_pos

`int get_current_ext_pos (HROBOT robot, int index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
pos	double[3]	此陣列包含三個空間 pos[0]: 外部軸 1 點位資訊 pos[1]: 外部軸 2 點位資訊 pos[2]: 外部軸 3 點位資訊
回傳值	int	成功: 0 失敗: 錯誤碼

3.3.27. 取得外部軸同異步模式 - get_current_ext_mode

`int get_current_ext_mode (HROBOT robot, char* mode)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
mode	char*	SYNC: 同步模式 ASYNC: 異步模式 NULL: 外部軸沒開啟 Example: NULL_NULL_NULL 表示三個外部軸都沒開啟
回傳值	int	成功: 0 失敗: 錯誤碼

3.3.28. 回復出廠校正位置- mastering

`int mastering (HROBOT robot ,char* joint,char* type)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
joint	char*	j1: 第一軸 j2: 第二軸 j3: 第三軸 j4: 第四軸 j5: 第五軸 j6: 第六軸
type	char*	zero: 原點位置 positive_limit: 正極限位置 negative_limit: 負極限位置
回傳值	int	成功: 0 失敗: 錯誤碼
備註	<ul style="list-style-type: none"> ➤ 需等待 Callback cmd:1410 rlt:4046 回傳，才是指令執行結束。 ➤ 手臂姿態需接近校正位置，校正位置參考各機型本體手冊。 	

3.3.29. 重置原點位置- calibration

`int calibration (HROBOT robot ,char* joint,char* type)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
joint	char*	j1: 第一軸 j2: 第二軸 j3: 第三軸 j4: 第四軸 j5: 第五軸 j6: 第六軸
type	char*	zero: 原點位置 positive_limit: 正極限位置 negative_limit: 負極限位置
回傳值	int	成功: 0 失敗: 錯誤碼
備註	<ul style="list-style-type: none"> ➤ 需等待 Callback cmd:1411 rlt:4046 回傳，才是指令執行結束。 ➤ 執行機構校正位置設定，需搭配專用治具使用。 ➤ 特別注意：進行此操作模式會導致遺失原先機器人出廠所設定之 	

	<p>原點位置，若操作不慎可能造成絕對精度有所偏差！！並且偏差之前所已校正的所有點位，點位不同時有可能導致撞機現象，需再重新確認並修改校正所有點位，再進行手臂自動化運行。</p> <ul style="list-style-type: none"> ➤ 六軸機器手臂： 進行第六軸馬達校正時，需先確認第五軸馬達角度是否為 0° 且編碼器數值位於單圈，若不符合則需先進行第五軸馬達校正。 ➤ 四軸 SCARA 機器手臂： 進行第三軸馬達校正時，需先確認第四軸馬達角度是否為 0° 且編碼器數值位於單圈，若不符合則需先進行第四軸馬達校正。
--	---

C++:

```

void main(){
    mastering(robot, "j1", "zero");
    WaitForCallback();
    calibration(robot, "j1", "zero");
}

void __stdcall callBack(uint16_t cmd, uint16_t rlt, uint16_t* msg, int len) {
    switch (cmd)
    {
        case 1410:
            switch (rlt) {
                case 4045: // 開始執行Mastering Zero Position
                    break;
                case 4046: // 結束執行Mastering Zero Position
                    break;
            }
        case 1411:
            switch (rlt) {
                case 4045: // 開始執行Mastering Calibration
                    break;
                case 4046: // 結束執行Mastering Calibration
                    break;
            }
    }
}

```

3.3.30. 下載歷史記錄- download_history_zip

`int download_history_zip (HROBOT robot ,char* dest_path)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
dest_path	char*	存放 LOG 壓縮檔的目標位置
回傳值	int	成功:0 失敗: 錯誤碼

C++:

```
download_history_zip(robot, "D://history.zip")
```

3.4. 輸出輸入命令類

函式名稱說明與對應身分參考下表：

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	O	O	O
get_digital_output	取得輸出	O	O	O
set_digital_output	設定輸出	O	O	×
set_DI_simulation_Enable	設定輸入模擬數值	O	O	×
set_DI_simulation	設定輸入狀態	O	O	×
get_DI_simulation_Enable	取得輸入模擬數值	O	O	O
set_digital_input_comment	設定輸入註解	O	O	×
get_digital_input_comment	取得輸入註解	O	O	O
set_digital_output_comment	設定輸出註解	O	O	×
get_digital_output_comment	取得輸出註解	O	O	O
get_robot_input	取得機器人輸入	O	O	O
get_robot_output	取的機器人輸出	O	O	O
set_robot_output	設定機器人輸出	O	O	×
get_valve_output	取得電磁閥輸出	O	O	O
set_valve_output	設定電磁閥輸出	O	O	×
get_function_input	取得功能輸入	O	O	O
get_function_output	取得功能輸出	O	O	O
get_module_input_config	取得模組輸入設置	O	O	O
get_module_output_config	取得模組輸出設置	O	O	O
set_module_input_simulation	設定模組輸入模擬數值	O	O	×
set_module_input_value	設定模組輸入	O	O	×
set_module_input_start	設定模組輸入開始編號	O	O	×
set_module_input_end	設定模組輸入結束編號	O	O	×
set_module_input_comment	設定模組輸入註解	O	O	×

函式名稱	說明	操作者		觀察者
		手動模式	自動模式	
set_module_output_value	設定模組輸出	O	O	×
set_module_output_start	設定模組輸出開始編號	O	O	×
set_module_output_end	設定模組輸出結束編號	O	O	×
set_module_output_comment	設定模組輸出註解	O	O	×
set_module_input_type	設定模組輸入型態	O	O	×
set_module_output_type	設定模組輸出型態	O	O	×
save_module_io_setting	保存模組設定	O	O	×
SyncOutput	運動中進行 DO 開關操作	O	O	×
get_DI_range	取得 DI 範圍數值	O	O	O
get_DI_sim_range	取得 DI 範圍模擬數值	O	O	O
get_DI_comment_range	取得 DI 範圍註解	O	O	O
get_DO_range	取得 DO 範圍數值	O	O	O
get_DO_comment_range	取得 DO 範圍註解	O	O	O
get_FI_all	取得 FI 全部數值	O	O	O
get_FO_all	取得 FO 全部數值	O	O	O
get_timer_status_all	取得 Timer 全部狀態	O	O	O
get_timer_value_all	取得 Timer 全部數值	O	O	O
get_timer_comment_range	取得 Timer 範圍註解	O	O	O
get_counter_value_all	取得 Counter 全部數值	O	O	O
get_counter_comment_range	取得 Counter 範圍註解	O	O	O
get_fieldbus_rs_srw_range	取得 Fieldbus Register SRW 範圍數值	O	O	O
get_fieldbus_rs_srr_range	取得 Fieldbus Register SRR 範圍數值	O	O	O
get_fieldbus_rs_parameter_range	取得 Fieldbus Register 範圍參數	O	O	O
get_fieldbus_rs_comment_range	取得 Fieldbus Register 範圍註解	O	O	O
get_system_input_all	取得 System 輸入全部數值	O	O	O
get_system_output_all	取得 System 輸出全部數值	O	O	O
get_MI_config_all	取得模組輸入全部設定	O	O	O

函式名稱	說明	操作者		觀察者
		手動模式	自動模式	
get_MO_config_all	取得模組輸出全部設定	O	O	O
get_MI_comment_range	取得模組輸入範圍註解	O	O	O
get_MO_comment_range	取得模組輸出範圍註解	O	O	O
get_SI_range	取得 SI 範圍數值	O	O	O
get_SI_sim_range	取得 SI 範圍模擬數值	O	O	O
get_SO_range	取得 SO 範圍數值	O	O	O
get_SI_comment_range	取得 SI 範圍註解	O	O	O
get_SO_comment_range	取得 SO 範圍註解	O	O	O
get_PR_comment_array	取得 PR 範圍註解	O	O	O
get_RI_all	取得 RI 全部數值	O	O	O
get_RO_all	取得 RO 全部數值	O	O	O
get_VO_all	取得 VO 全部數值	O	O	O
set_DI_array	設定 DI 多個數值	O	O	X
set_DI_sim_array	設定 DI 多個模擬數值	O	O	X
set_DO_array	設定 DO 多個數值	O	O	X
set_timer_value_array	設定 Timer 多個數值	O	O	X
set_counter_array	設定 Counter 多個數值	O	O	X
set_fieldbus_srw_array	設定 Fieldbus SRW 多個數值	O	O	X
set_SI_array	設定 SI 多個數值	O	O	X
set_SI_sim_array	設定 SI 多個模擬數值	O	O	X
set_SO_array	設定 SO 多個數值	O	O	X
set_MO_array	設定 MO 多個數值	O	O	X
set_RO_array	設定 RO 多個數值	O	O	X
set_VO_array	設定 VO 多個數值	O	O	X

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.4.1. 取得輸入狀態- get_digital_input

`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);
```

3.4.2. 設定輸入模擬數值- set_DI_simulation_Enable

`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: 錯誤碼

3.4.3. 設定輸入狀態- set_DI_simulation

`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: 錯誤碼
備註	需先啟動輸入模擬	

3.4.4. 取得輸入模擬數值- get_DI_simulation_Enable

`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);
```

3.4.5. 取得輸出狀態- get_digital_output

`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);
```

3.4.6. 設定輸出狀態- set_digital_output

`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: 錯誤碼

C++:

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

3.4.7. 設定輸入註解- set_digital_input_comment

`int set_digital_input_comment (HROBOT robot, int index, wchar_t* 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	wchar_t*	設定註解的文字
回傳值	int	成功:0 失敗: 錯誤碼

3.4.8. 設定輸出註解- set_digital_output_comment

`int set_digital_output_comment (HROBOT robot, int index, wchar_t* 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	wchar_t*	設定註解的文字
回傳值	int	成功:0 失敗: 錯誤碼

3.4.9. 取得輸入註解- get_digital_input_comment

`int get_digital_input_comment (HROBOT robot, int index, wchar_t* comment, int arr_size)`

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	wchar_t*	➤ 取得註解的文字 ➤ 第一個值為註解長度
arr_size	int	可設定想要取得的註解長度
回傳值	int	成功:0 失敗: 錯誤碼

3.4.10. 取得輸出註解- get_digital_output_comment

`int get_digital_output_comment (HROBOT robot, int index, wchar_t* comment, int arr_size)`

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	wchar_t*	➤ 取得註解的文字 ➤ 第一個值為註解長度
array_size	int	可設定想要取得的註解長度
回傳值	int	成功:0 失敗:錯誤碼

C++:

```
int index = 1;
wchar_t* comment = L“Test comment”;
wchar_t str[20] = “”;
int array_size = 10;
set_digital_input_comment (robot, index, comment);
get_digital_input_comment(robot, index, str, array_size);
set_digital_output_comment (robot, index, comment);
get_digital_output_comment(robot, index, str, array_size);
```

3.4.11. 取得機器人輸入- get_robot_input

`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);
```

3.4.12. 取得機器人輸出- get_robot_output

```
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);
```

3.4.13. 設定機器人輸出- set_robot_output

```
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: 錯誤碼

C++:

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

3.4.14. 取得電磁閥輸出- get_valve_output

```
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);
```

3.4.15. 設定電磁閥輸出- set_valve_output

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: 錯誤碼

C++:

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

3.4.16. 取得功能輸入狀態- get_function_input

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);
```

3.4.17. 取得功能輸出狀態- get_function_output

`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);
```

3.4.18. 取得模組輸入設置 - get_module_input_config

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

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸入編號[1-32]的索引值
sim	bool	模擬值 0:OFF 1:ON
value	bool	0:OFF 1:ON
type	int	0:DI 1:SI
start	int	0: Disable [1-48]: 數位輸入編號的開始索引值
end	int	0: Disable [1-48]: 數位輸入編號的結束索引值 (不得小於 start)
comment	wchar_t*	➤ 取得註解的文字 ➤ 第一個值為註解長度
arr_size	int	可設定想要取得的註解長度
回傳值	int	成功:0 失敗: 錯誤碼

3.4.19. 取得模組輸出設置- get_module_output_config

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

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸出編號[1-32]的索引值
sim	bool	模擬值 0:OFF 1:ON
value	bool	0:OFF 1:ON
type	int	0: DO 1: SO
start	int	0: Disable [1-48]: 數位輸出編號的開始索引值
end	int	0: Disable [1-48]: 數位輸出編號的結束索引值
comment	wchar_t*	➤ 取得註解的文字 ➤ 第一個值為註解長度
arr_size	int	可設定想要取得的註解長度
回傳值	int	成功:0 失敗: 錯誤碼

3.4.20. 設定模組輸入模擬數值- set_module_input_config

```
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[1-32]的索引值
enable	bool	模擬值 0:OFF 1:ON
回傳值	int	成功:0 失敗: 錯誤碼

3.4.21. 設定模組輸入- set_module_input_value

`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[1-32]的索引值
enable	bool	0:OFF 1:ON ※需 sim 為 ON 時，才能設為 ON。
回傳值	int	成功:0 失敗: 錯誤碼

3.4.22. 設定模組輸入開始編號- set_module_input_start

`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[1-32]的索引值
start_number	int	0: Disable [1-48]: 數位輸入編號的開始索引值
回傳值	int	成功:0 失敗: 錯誤碼

3.4.23. 設定模組輸入結束編號- set_module_input_end

`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[1-32]的索引值
end_number	int	0: Disable [1-48]: 數位輸入編號的結束索引值 (不得小於 start)
回傳值	int	成功:0 失敗: 錯誤碼

3.4.24. 設定模組輸入註解- set_module_input_comment

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

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸入編號[1-32]的索引值
comment	wchar_t*	設定指定模組輸入的註解
回傳值	int	成功:0 失敗: 錯誤碼

3.4.25. 設定模組輸出- set_module_output_value

`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[1-32]的索引值
enable	bool	0:OFF 1:ON
回傳值	int	成功:0 失敗: 錯誤碼

3.4.26. 設定模組輸出開始編號- set_module_output_start

`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[1-32]的索引值
start_number	int	0: Disable [1-48]: 數位輸出編號的開始索引值
回傳值	int	成功:0 失敗: 錯誤碼

3.4.27. 設定模組輸出結束編號- set_module_output_end

`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[1-32]的索引值
end_number	int	0: Disable [1-48]: 數位輸出編號的結束索引值 (不得小於 start)
回傳值	int	成功:0 失敗: 錯誤碼

3.4.28. 設定模組輸出註解- set_module_output_comment

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

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	模組輸出編號[1-32]的索引值
comment	wchar_t*	設定註解的文字
回傳值	int	成功:0 失敗: 錯誤碼

3.4.29. 設定模組輸入型態- set_module_input_type

`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[1-32]的索引值
type	int	0: DI 1: SI
回傳值	int	成功:0 失敗: 錯誤碼

3.4.30. 設定模組輸出型態- set_module_output_type

`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[1-32]的索引值
type	int	0: DO 1: SO
回傳值	int	成功:0 失敗: 錯誤碼

3.4.31. 保存模組設定- save_module_io_setting

`int save_module_io_setting (HROBOT s)`

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

C++:

```
bool sim,value;
int start,end, type;
wchar_t comment[100];
wchar_t* set_comment = L"module comment";
int array_size = 10;
get_module_input_config (robot,0,sim,value,type,start,end,comment, array_size);
get_module_output_config(robot,0,sim,value,type,start,end,comment, array_size);
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, set_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, set_comment);
set_module_output_type(robot, 0, type)
```

```
save_module_io_setting(robot);
```

3.4.32. 運動中進行 DO 開關操作- SyncOutput

```
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 );

```

3.4.33. 取得 DI 範圍數值- get_DI_range

`int get_DI_range (HROBOT robot, int from_idx, int end_idx, int* value)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得數值
end_idx	int	取得數值到指定 Index
value	int*	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

C++:

```

int from_idx = 1;
int end_idx = 48;
int values[48] = {0};
get_DI_range (robot, from_idx, end_idx, values);

// DI 1 = values[0]
// DI 2 = values[1]
// DI 3 = values[2]
....
```

3.4.34. 取得 DI 範圍模擬數值- get_DI_sim_range

`int get_DI_sim_range (HROBOT robot, int from_idx, int end_idx, int* value)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得數值
end_idx	int	取得數值到指定 Index
value	int*	取得數值的陣列
回傳值	int	成功: 0: 失敗: 錯誤碼

3.4.35. 取得 DI 範圍註解- get_DI_comment_range

`int get_DI_comment_range (HROBOT robot, int from_idx, int end_idx, wchar_t* comment, int& next_idx)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功: 0: 失敗: 錯誤碼

C++:

3.4.36. 取得 DO 範圍數值- get DO range

`int get DO range (HROBOT robot, int from_idx, int end_idx, int* value)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得數值
end_idx	int	取得數值到指定 Index
value	int*	取得數值的陣列
回傳值	int	成功: 0: 失敗: 錯誤碼

3.4.37. 取得 DO 範圍註解- get_DO_comment_range

```
int get_DO_comment_range (HROBOT robot, int from_idx, int end_idx, wchar_t* comment, int& next_idx)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.38. 取得 FI 全部數值-get_FI_all

```
int get_FI_all (HROBOT robot, int* values)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[8]	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.39. 取得 FO 全部數值-get_FO_all

```
int get_FO_all (HROBOT robot, int* values)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[8]	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.40. 取得 Timer 全部狀態- get_timer_status_all

`int get_timer_status_all (HROBOT robot, int* values)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[20]	取得狀態的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.41. 取得 Timer 全部數值- get_timer_value_all

`int get_timer_value_all (HROBOT robot, int* values)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[20]	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.42. 取得 Timer 範圍註解- get_timer_comment_range

`int get_timer_comment_range (HROBOT robot, int from_idx, int end_idx, wchar_t* comment, int& next_idx)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.43. 取得 Counter 全部數值-get_counter_value_all

`int get_counter_value_all (HROBOT robot, int* values)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[20]	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.44. 取得 Counter 範圍註解-get_counter_comment_range

`int get_counter_comment_range (HROBOT robot, int from_idx, int end_idx , wchar_t* comment, int& next_idx)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.45. 取得 Fieldbus Register SRW 範圍數值-get_fieldbus_rs_srw_range

`int get_fieldbus_rs_srw_range (HROBOT robot, int from_idx, int end_idx , int* values)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得數值
end_idx	int	取得數值到指定 Index
values	int*	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.46. 取得 Fieldbus Register SRR 範圍數值- get_fieldbus_rs_srr_range

```
int get_fieldbus_rs_srr_range (HROBOT robot, int from_idx, int end_idx , int* values)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得數值
end_idx	int	取得數值到指定 Index
values	int*	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.47. 取得 Fieldbus Register 範圍參數- get_fieldbus_rs_parameter_range

```
int get_fieldbus_rs_parameter_range (HROBOT robot, int from_idx, int end_idx ,  
wchar_t* para, int& next_idx)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得參數
end_idx	int	取得參數到指定 Index
comment	wchar_t*	陣列第一個數值表示此參數整體長度。 每個 Index 的參數以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的參數； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得參數。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.48. 取得 Fieldbus Register 範圍註解- get_fieldbus_rs_comment_range

```
int get_fieldbus_rs_comment_range (HROBOT robot, int from_idx, int end_idx ,  
wchar_t* comment, int& next_idx)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度 。 每個 Index 的註解以"\t"當分割符號 。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解 。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.49. 取得 System 輸入全部數值- get_system_input_all

```
int get_system_input_all (HROBOT robot, int* values,, wchar_t* comment)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int*	取得數值的陣列
comment	wchar_t*	陣列第一個數值表示此註解整體長度 。 每個 Index 的註解以"\t"當分割符號 。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.50. 取得 System 輸出全部數值- get_system_output_all

```
int get_system_output_all (HROBOT robot, int* values,, wchar_t* comment)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int*	取得數值的陣列
comment	wchar_t*	陣列第一個數值表示此註解整體長度 。 每個 Index 的註解以"\t"當分割符號 。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.51. 取得模組輸入全部設定- get_MI_config_all

```
int get_MI_config_all (HROBOT robot, int* sim, int* values, int* types, int* starts, int* end)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
sim	int[32]	取得模擬的陣列
values	int[32]	取得數值的陣列
types	int[32]	取得 IO 型態的陣列
starts	int[32]	取得開始的陣列
end	int[32]	取得結束的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.52. 取得模組輸出全部設定- get_MO_config_all

```
int get_MO_config_all (HROBOT robot, int* values, int* types, int* starts, int* end)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[32]	取得數值的陣列
types	int[32]	取得 IO 型態的陣列
starts	int[32]	取得開始的陣列
end	int[32]	取得結束的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.53. 取得模組輸入範圍註解- get_MI_comment_range

```
int get_MI_comment_range (HROBOT robot, int from_idx, int end_idx , wchar_t* comment, int& next_idx)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.54. 取得模組輸出範圍註解- get_MO_comment_range

```
int get_MO_comment_range (HROBOT robot, int from_idx, int end_idx , wchar_t* comment, int& next_idx)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.55. 取得 SI 範圍數值- get_SI_range

`int get_SI_range (HROBOT robot, int from_idx, int end_idx, int* value)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得數值
end_idx	int	取得數值到指定 Index
value	int*	取得數值的陣列
回傳值	int	成功: 0: 失敗: 錯誤碼

3.4.56. 取得 SI 範圍模擬數值- get_SI_sim_range

`int get_SI_sim_range (HROBOT robot, int from_idx, int end_idx, int* value)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得數值
end_idx	int	取得數值到指定 Index
value	int*	取得數值的陣列
回傳值	int	成功: 0: 失敗: 錯誤碼

3.4.57. 取得 SO 範圍數值- get_SO_range

`int get_SO_range (HROBOT robot, int from_idx, int end_idx, int* value)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得數值
end_idx	int	取得數值到指定 Index
value	int*	取得數值的陣列
回傳值	int	成功: 0: 失敗: 錯誤碼

3.4.58. 取得 SI 範圍註解- get_SI_comment_range

```
int get_SI_comment_range (HROBOT robot, int from_idx, int end_idx, wchar_t* comment, int& next_idx)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.59. 取得 SO 範圍註解- get_SO_comment_range

```
int get_SO_comment_range (HROBOT robot, int from_idx, int end_idx, wchar_t* comment, int& next_idx)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
from_idx	int	從指定 Index 開始取得註解
end_idx	int	取得註解到指定 Index
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.60. 取得 PR 範圍註解- get_PR_comment_array

```
int get_PR_comment_array (HROBOT robot, int* indexes, int from_idx, int len ,
wchar_t* comment, int& next_idx)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	此陣列帶入要取得註解的索引值。會依序從帶入的索引值抓取註解
from_idx	int	從指定陣列 Index 開始取得註解
len	int	需要取得的 indexes 總長度
comment	wchar_t*	陣列第一個數值表示此註解整體長度。 每個 Index 的註解以"\t"當分割符號。
next_idx	int&	當 next_idx 為 0 表示以完整取得指定範圍的註解； 若是 next_idx 非 0，表示需要從該數值(from_idx) 繼續取得註解。
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.61. 取得 RI 全部數值- get_RI_all

```
int get_RI_all (HROBOT robot, int* values)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[8]	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.62. 取得 RO 全部數值- get_RO_all

```
int get_RO_all (HROBOT robot, int* values)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[8]	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.63. 取得 VO 全部數值- get_VO_all

```
int get_VO_all (HROBOT robot, int* values)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
values	int[3]	取得數值的陣列
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.64. 設定 DI 多個數值- set_DI_array

```
int set_DI_array (HROBOT robot, int* indexes, int* values, int len)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

C++:

```
int len = 48;
int indexes[48] = {0};
int values[48] = {0};
for(int idx=1; idx<=48;i++){
    indexes[idx-1] = idx;
    if(idx%2==0){
        values[idx-1] = 1;
    }else{
        values[idx-1] = 0;
    }
}
set_DI_array (robot, indexes, values, len);

// DI 1 = values[0]
// DI 2 = values[1]
// DI 3 = values[2]
....
```

3.4.65. 設定 DI 多個模擬數值- set_DI_sim_array

`int set_DI_sim_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.66. 設定 DO 多個數值- set_DO_array

`int set_DO_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.67. 設定 Timer 多個數值- set_timer_value_array

`int set_timer_value_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.68. 設定 Counter 多個數值-set_counter_array

`int set_counter_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.69. 設定 SI 多個數值-set_SI_array

`int set_SI_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.70. 設定 SI 多個模擬數值-set_SI_sim_array

`int set_SI_sim_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.71. 設定 SO 多個數值- set_SO_array

`int set_SO_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.72. 設定 Fieldbus SRW 多個數值- set_fieldbus_srw_array

`int set_fieldbus_srw_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.73. 設定 MO 多個數值- set_MO_array

`int set_MO_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.74. 設定 VO 多個數值- set_VO_array

`int set_VO_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.4.75. 設定 RO 多個數值-set_RO_array

`int set_RO_array (HROBOT robot, int* indexes, int* values, int len)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
indexes	int*	放入所要設定的索引數值到此陣列。會依序以此陣列的索引與數值陣列設定數值。
values	int*	設定數值的陣列
len	int	所要設定的陣列長度
回傳值	int	成功:0: 失敗: 錯誤碼

3.5. 座標命令類

函式名稱說明與對應身分參考下表：

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	O	O	×
get_base_number	取得基底號碼	O	O	O
define_base	定義基底座標	O	×	×
get_base_data	取得基底座標	O	O	O
set_tool_number	設定工具號碼	O	O	×
get_tool_number	取得工具號碼	O	O	O
define_tool	定義工具座標	O	×	×
get_tool_data	取得工具座標	O	O	O
tool_calibration	工具座標多點校準	O	O	×
base_calibration	基座作標三點校準	O	O	×

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.5.1. 設定基底號碼- set_base_number

`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	基底座標標號
num	int	欲選擇的基底編號(0-31)
回傳值	int	成功: 0 失敗: 錯誤碼

3.5.2. 取得基底號碼- get_base_number

`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: 錯誤碼

C++:

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

3.5.3. 定義基底座標- define_base

`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	基底座標號碼(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: 錯誤碼

3.5.4. 取得基底座標- get_base_data

`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: 錯誤碼

3.5.5. 設定工具號碼- set_tool_number

`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: 錯誤碼

3.5.6. 取得工具號碼- get_tool_number

```
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: 錯誤碼

C++:

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

3.5.7. 定義工具座標- define_tool

```
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: 錯誤碼

3.5.8. 取得工具座標- get_tool_data

```
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);
```

- ⇒ 定義工具座標，並利用get_tool_data來取得座標資訊。
- ⇒ 定義基底座標，並利用get_base_data來取得座標資訊。

3.5.9. 工具座標多點校準- tool_calibration

說明：

將待測量工具的 TCP 從 4 個不同方向移向一個參照點。參照點可以任意選擇。機器人控制系統從不同的法蘭位置值中計算出工具中心點(TCP)。利用 4 點不同姿態的點位，透過軟體計算取得新工具座標 TCP 點(RS 系列採用 3 點校正)，即校正後取得新 TOOL 坐標系的 X、Y、Z、A、B、C 值，其中 RS 系列無 A、B 值，而 C 值(J4 旋轉角度)為第一個校正點姿態的 C 值。

```
int tool_calibration (HROBOT robot, int type, double* p0_coord, double* p1_coord,
double* p2_coord, double* p3_coord, double* result_coord)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
type	int	0: RA, RT 機型 1: RS 機型
p0_coord	double[6]	測量點位 1
p1_coord	double[6]	測量點位 2
p2_coord	double[6]	測量點位 3
p3_coord	double[6]	測量點位 4，RS 系列此點位不當作參考。
result_coord	double[6]	新工具座標系數值
回傳值	int	成功:0: 失敗: 錯誤碼
備註		將待測量工具的 TCP 從 3 個不同方向移向一個參照點。參照點可以任意選擇。機器手臂控制系統從不同的法蘭位置值中計算出 TCP。校正後取得 TOOL 坐標系的 X、Y、Z、A、B、C 值

C++範例檔：

C++:

```
double p0_coord [6] = {0, 431, 26, 163, 2, 89};
double p1_coord [6] = {0, 470, 22, 179, -27, 91};
double p2_coord [6] = {0, 476, 18, -145, 0, 90};
double p3_coord [6] = {0, 486, -6,-176, 20, 89};
double result_coord [6] = {0};

int selected_tool = 3;

tool_calibration(robot, 0, p0_coord, p1_coord, p2_coord, p3_coord, result_coord);
define_tool(robot, selected_tool, result_coord);
```

3.5.10. 基底座標三點校準- base_calibration

說明：在基準測量時，用戶配給工作面或工件一個笛卡爾座標系（基底座標系）。基底座標系的原點為用戶指定的一個點。利用三點的方法求得新的基座座標系。校正時需要三點資訊後，透過軟體計算後，將原本機器人座標系改換至新的用戶指定座標系。

1. 第一點 p0，用 TCP 移至新基底座標系的原點。
2. 第二點 p1，將 TCP 移至新基底座標系正向 X 軸上的一個點。
3. 第三點 p2，將 TCP 移至 XY 平面上一個帶有正 Y 值的點。

```
int base_calibration (HROBOT robot, int type, double* p0_coord, double* p1_coord,
double* p2_coord, double* result_coord)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
type	int	0: 不分機型
p0_coord	double[6]	新基底座標系的原點
p1_coord	double[6]	新基底座標系正向 X 軸的一個點位
p2_coord	double[6]	新基底座標系正向 Y 值的一個點位
result_coord	double[6]	新基底座標系數值
回傳值	int	成功: 0: 失敗: 錯誤碼

C++範例檔：

C++:

```
double p0_coord [6] = {0, 400, 0, 0, 0, 0};
double p1_coord [6] = {100, 400, 0, 0, 0, 0};
double p2_coord [6] = {0, 500, 0, 0, 0, 0};
double result_coord [6] = {0};
int selected_base = 3;

base_calibration(robot, 0, p0_coord, p1_coord, p2_coord, result_coord);
define_base(robot, selected_base, result_coord);
```

3.6. 任務命令類

函式名稱說明與對應身分參考下表：

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	○	○	○

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.6.1. 設定 RSR- set_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);
```

3.6.2. 取得 RSR 程式名稱- get_rsr_prog_name

`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: 錯誤碼

C++:

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

3.6.3. 移除 RSR- remove_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: 錯誤碼

C++:

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

3.6.4. RSR/PNS 啟動外部觸發任務- ext_task_start

`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。

3.6.5. 啟動任務- task_start

`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
```

3.6.6. 暫停任務- task_hold

`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: 錯誤碼

3.6.7. 繼續任務- task_continue

`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: 錯誤碼

3.6.8. 停止任務- task_abort

`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: 錯誤碼

備註：停止 hrb 程式，需要等待 Callback(cmd=4004, rlt=4018)回傳，才是 task_abort 完成動作。可參考範例程式 14.TaskAbort

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

3.6.9. 取得目前執行任務名稱- `get_execute_file_name`

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

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

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;
```

3.7. 檔案管理命令類

函式名稱說明與對應身分參考下表：

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	O	O	×
download_file	下載 HRB 至本地端	O	O	O
delete_file	刪除 Robot 運動檔案	O	O	×
delete_folder	刪除 Robot 運動檔案資料夾	O	O	×
new_folder	新增 Robot 運動檔案資料夾	O	O	×
file_rename	重新命名 Robot 運動檔案	O	O	×
file_drag	拖動 Robot 運動檔案	O	O	×
get_prog_number	取得檔案數量	O	O	O
get_prog_name	取得檔案名稱	O	O	O

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.7.1. 下載 HIWIM Robot language 檔案- download_file

`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: 錯誤碼

C++:

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

3.7.2. 上傳 HIWIN Robot language 檔案- send_file

```
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: 錯誤碼
備註	若是檔名一樣會直接覆蓋	

C++:

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

3.7.3. 刪除 Robot 運動檔案- delete_file

```
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: 錯誤碼

3.7.4. 刪除 Robot 運動檔案資料夾- delete_folder

```
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: 錯誤碼

3.7.5. 新增 Robot 運動檔案資料夾 - new_folder

`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: 錯誤碼

3.7.6. 重新命名 Robot 運動檔案 - file_rename

`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: 錯誤碼

3.7.7. 拖動 Robot 運動檔案 - file_drag

`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: 錯誤碼

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");
```

3.7.8. 取得檔案數量- get_prog_number

`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

3.7.9. 取得檔案名稱- get_prog_name

`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: 錯誤碼
備註	需先取得檔案數量 <code>get_prog_number</code> ，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);
}
```

3.8. 控制器命令類

函式名稱說明與對應身分參考下表：

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	○	×	×
get_robot_info	取得手臂資訊	○	○	○

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.8.1. 取得 HRSS 當前模式-`get_hrss_mode`

`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: 錯誤碼

C++:

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

3.8.2. 駕服設定- set_motor_state

`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:駕服關閉 1:駕服啟動 駕服啟動後需等待 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: 錯誤碼

3.8.3. 取得駕服狀態- get_motor_state

`int get_motor_state(HROBOT robot)`

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

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。

3.8.4. 設定操作模式- set_operation_mode

`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。

3.8.5. 取得操作模式- `get_operation_mode`

`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: 錯誤碼

3.8.6. 錯誤清除- `clear_alarm`

`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: 錯誤碼

C++:

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

3.8.7. 更新 HRSS- update_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: 錯誤碼 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);
```

3.8.8. 取得手臂資訊- get_robot_info

```
int get_robot_info (HROBOT robot, int page_sel, int tool_num, int base_num, char* info, bool is_ready)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
page_sel	int	<p>選擇頁面取得該頁面的 Callback 資料。</p> <p>0: 手臂坐標位置資訊 1: 數位輸入 2: 數位輸出 3: FieldBus I/O, 4: Function I/O, 5: 計時器(Timer), 6: 計數器(Counter) 7: 通訊 TCP/IP 8: 通訊 RS232, 9: 原點設定 10: 電爪 11: 飛抓視覺設置 12: 飛抓視覺物件 13: 系統 I/O 14: 模組輸入 15: 模組輸出 16: 暫存器 17: 稼動率 18: 校正 0xa0: 監測所有頁面資料回傳。</p>
tool_num	int	在 robot info 取得選擇的 tool 資料
base_num	int	在 robot info 取得選擇的 base 資料
info	char*	<p>資訊包含下列資料，每個資料以"," 做區隔。</p> <p>0: HRSS 模式 1: 操作模式 2: 運動狀態 3: 加速度時間 4: 警報數量 5: Function 輸出 6: Tool 編號 7: Base 編號</p>

參數	資料型態	說明
		8: Tool 座標，共 6 個值 14: Base 座標，共 6 個值
bool	is_ready	0: 取得最新的手臂資訊 1: 使用上次自動收到的手臂資訊
傳回值	int	成功: 0 失敗: 錯誤碼
備註	➤ 若要取得 Callback Notify 需先下達 get_current_position。	

C++:

```
HRobot.get_current_position( device_id, pos ); // 必須有這一行
HRobot.get_robot_info( device_id, 0xa0, 0, 0, msg, false ); // 必須有這一行
```

3.9. 吋動類

函式名稱說明與對應身分參考下表：

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	○	×	×

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.9.1. 吋動- jog

`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 3:外部軸座標系
index	int	吋動對象 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) 工具座標系(Tx:0, Ty:1, Tz:2, RTx:3, RTy:4, RTz:5) 外部軸座標系(E1:0, E2:1, E3:2)
dir	int	方向 1:正方向 -1:負方向
回傳值	int	成功:0 失敗:錯誤碼

➤ 只有在手動模式下有效

➤ 吋動單位說明：

- (1). 笛卡爾座標 X、Y、Z 軸為毫米(mm) , A、B、C 軸為度(degree)。
- (2). 關節座標軸為度(degree) , 關節座標軸中手臂本體標示 Joint1(J1)軟體表示為 A1、Joint2(J2)軟體表示為 A2 、Joint3(J3)軟體表示為 A3...以此類推至 Joint6。
- (3). 工具座標系 Tx、Ty、Tz 軸為毫米(mm) , RTx、RTy、RTz 軸為度(degree)。
- (4). 外部軸座標系直線單位為毫米(mm) , 旋轉單位為度(degree)。

3.9.2. 叱動復歸- jog_home

`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: 錯誤碼

➤ 只在安全線速模式下有效

3.9.3. 叱動停止- jog_stop

`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: 錯誤碼

只在安全線速模式下有效

3.10. 運動命令類

函式名稱說明與對應身分參考下表：

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	O	O	×
ptp_axis	絕對關節角度點對點運動	O	O	×
ptp_rel_pos	相對座標位置點對點運動	O	O	×
ptp_rel_axis	相對關節角度點對點運動	O	O	×
ptp_pr	位置暫存器之點對點運動	O	O	×
lin_pos	絕對座標位置直線運動	O	O	×
lin_axis	絕對關節角度直線運動	O	O	×
lin_rel_pos	相對座標位置直線運動	O	O	×
lin_rel_axis	相對關節角度直線運動	O	O	×
lin_pr	位置暫存器之直線運動	O	O	×
circ_axis	絕對關節角度圓弧運動	O	O	×
circ_pos	絕對座標位置圓弧運動	O	O	×
circ_pr	位置暫存器之圓弧運動	O	O	×
motion_hold	運動暫停	O	O	×
motion_continue	運動繼續	O	O	×
motion_abort	運動停止	O	O	×
motion_delay	運動延遲	O	O	×
set_command_id	設定運動命令編號	O	O	×
get_command_id	取得目前運動命令編號	O	O	O
get_command_count	取得目前運動命令佇列中命令個數	O	O	O
get_motion_state	取得目前運動狀態	O	O	O
remove_command	移除運動命令佇列中命令	O	O	×
remove_command_tail	移除運動命令佇列中數個命令	O	O	×
ext_ptp_axis	外部軸絕對關節角度點對點運動	O	O	×
ext_ptp_pos	外部軸絕對座標位置點對點運動	O	O	×
ext_lin_axis	外部軸絕對關節角度直線運動	O	O	×
ext_lin_pos	外部軸絕對座標位置直線運動	O	O	×
ext_asyptp	外部軸非同步關節角度點對點運動	O	O	×
motion_reachable	目標位置運動是否可到位	O	O	O
motion_check_lin	兩點運動是否可到位	O	O	O

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.10.1. 絕對座標位置點對點運動- ptp_pos

`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: 錯誤碼

3.10.2. 絕對關節角度點對點運動- ptp_axis

`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: 錯誤碼

備註：關節座標軸中手臂本體標示 Joint1(J1)軟體表示為 A1、Joint2(J2)軟體表示為 A2、Joint3(J3)軟體表示為 A3...以此類推至 Joint6。

3.10.3. 相對座標位置點對點運動- ptp_rel_pos

`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: 錯誤碼

3.10.4. 相對關節角度點對點運動- ptp_rel_axis

`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: 錯誤碼

3.10.5. 位置暫存器之點對點運動- ptp_pr

`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, 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

```

3.10.6. 絕對座標位置直線運動- lin_pos

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: 錯誤碼

3.10.7. 絕對關節角度直線運動- lin_axis

`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: 錯誤碼

備註：關節座標軸中手臂本體標示 Joint1(J1)軟體表示為 A1、Joint2(J2)軟體表示為 A2、Joint3(J3)軟體表示為 A3...以此類推至 Joint6。

3.10.8. 相對座標位置直線運動- lin_rel_pos

`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: 錯誤碼

3.10.9. 相對關節角度直線運動- lin_rel_axis

`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: 錯誤碼

備註：關節座標軸中手臂本體標示 Joint1(J1)軟體表示為 A1、Joint2(J2)軟體表示為 A2、Joint3(J3)軟體表示為 A3...以此類推至 Joint6。

3.10.10. 位置暫存器之直線運動- lin_pr

`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: 錯誤碼

C++:

```

double AxishomePoint[6]={0,0,0,0,-90,0};
lin_axis(s, 0, 10, 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

```

3.10.11. 絕對座標位置圓弧運動- circ_pos

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: 錯誤碼

3.10.12. 關節座標位置圓弧運動- circ_axis

```
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 關節爾座標{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 關節爾座標{A1, A2, A3, A4, A5, A6} 數值範圍(2147418.112,- 2147418.112)
回傳值	int	成功:0 失敗: 錯誤碼

備註：關節座標軸中手臂本體標示 Joint1(J1)軟體表示為 A1、Joint2(J2)軟體表示為 A2 、Joint3(J3)軟體表示為 A3...以此類推至 Joint6 。

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);
```

3.10.13. 位置暫存器之圓弧運動- circ_pr

`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
```

3.10.14. 運動暫停- motion_hold

`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: 錯誤碼

- 當有任務執行時，指令會失敗。

3.10.15. 運動繼續- motion_continue

`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: 錯誤碼

- 當有任務執行時，指令會失敗。

3.10.16. 運動停止- motion_abort

`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: 錯誤碼

- 停止運動指令(PTP, LIN, CIRC)
- 當有任務執行時，指令會失敗。

3.10.17. 運動延遲- motion_delay

`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);
```

3.10.18. 設定運動命令編號- set_command_id

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: 錯誤碼

- 當有任務執行時，指令會失敗。

3.10.19. 取得運動命令編號- get_command_id

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: 錯誤碼

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);
}

```

3.10.20. 取得運動命令佇列中命令個數- get_command_count

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: 錯誤碼

3.10.21. 取得目前運動狀態- get_motion_state

`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 0：解激磁狀態 1：等待、閒置狀態 2：運行狀態 3：暫停狀態 4：延遲狀態 5：移動狀態 失敗： 錯誤碼

➤ 閒置狀態：沒有運動命令

3.10.22. 移除運動命令佇列中單一命令- remove_command

`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：錯誤碼

3.10.23. 移除運動命令佇列中最新數個命令- remove_command_tail

`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：錯誤碼

3.10.24. 外部軸絕對關節角度點對點運動- ext_ptp_axis

`int ext_ptp_axis (HROBOT robot, int mode, double* axis)`

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
axis	double[9]	關節座標{A1, A2, A3, A4, A5, A6, E1, E2, E3} 數值範圍(2147418.112, - 2147418.112) 四軸 SCARA 手臂，A5, A6 數值設為 0， 外部軸模式需設為同步(Sync)才會移動，若是不啟用或是異步模式則外部軸不會運動。
回傳值	int	成功: 0 失敗: 錯誤碼

3.10.25. 外部軸絕對座標位置點對點運動- ext_ptp_pos

`int ext_ptp_pos (HROBOT robot, int mode, double* axis)`

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
axis	double[9]	笛卡爾座標{X, Y, Z, A, B, C, E1, E2, E3} 數值範圍(2147418.112, - 2147418.112) 四軸 SCARA 手臂，A, B 數值設為 0， 外部軸模式需設為同步(Sync)才會移動，若是不啟用或是異步模式則外部軸不會運動。
回傳值	int	成功: 0 失敗: 錯誤碼

3.10.26. 外部軸絕對關節角度直線運動- ext_lin_axis

int ext_lin_axis (HROBOT robot, int mode, double smooth_value, double* axis)

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
axis	double[9]	關節座標{A1, A2, A3, A4, A5, A6, E1, E2, E3} 數值範圍(2147418.112, - 2147418.112) 四軸 SCARA 手臂，A5, A6 數值設為 0， 外部軸模式需設為同步(Sync)才會移動，若是不啟用或是異步模式則外部軸不會運動。
回傳值	int	成功: 0 失敗: 錯誤碼

備註：關節座標軸中手臂本體標示 Joint1(J1)軟體表示為 A1、Joint2(J2)軟體表示為 A2、Joint3(J3)軟體表示為 A3...以此類推至 Joint6。

3.10.27. 外部軸絕對座標位置直線運動- ext_lin_pos

`int ext_lin_pos (HROBOT robot, int mode, double smooth_value, double* pos)`

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
pos	double[9]	笛卡爾座標{X, Y, Z, A, B, C, E1, E2, E3} 數值範圍(2147418.112,- 2147418.112) 四軸 SCARA 手臂，A, B 數值設為 0， 外部軸模式需設為同步(Sync)才會移動，若是不啟用或是異步模式則外部軸不會運動。
回傳值	int	成功: 0 失敗: 錯誤碼

3.10.28. 外部軸非同步關節角度點對點運動- ext_asyptp

int ext_asyptp (HROBOT robot, int mode, double* axis)

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
axis	double[3]	關節座標{ E1, E2, E3} 數值範圍(2147418.112, -2147418.112) <ul style="list-style-type: none"> ➤ 外部軸模式需設為非同步(Async)才會移動，若是同步模式則會發出[需要設為非同步]的警報。 ➤ 若不運動該外部軸，可以帶入 0xffff，防止同步模式發警報。 ➤ EX: 只開 E2 軸 double pos[3] = {0xffff, 10, 0xffff}
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

```

double ext_axis[9] = {10, 20, 0, 0, -90, 0, 10.5, 20.5, 30.5};
ext_ptp_axis(robot, 0, ext_axis);
ext_lin_axis(robot, 0, 0, ext_axis);

double ext_pos[9]={ 217.5 ,58.5 ,601.659 ,60.8 ,35.5 ,95.7, -10.8, -20.6, -30.7};
ext_ptp_pos(robot, 0, ext_pos);
ext_lin_pos(robot, 0, 0, ext_pos);

double ext_asyaxis[3]={ -10.3, 20.4, 0xffff};
// E1, E2: Async; E3: Sync
ext_asyptp(robot, 0, ext_asyaxis);

```

3.10.29. 目標位置運動是否可到位- motion_reachable

```
int motion_reachable (HROBOT robot , double* dest_cart, bool& is_reachable)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
dest_coord	double[6]	目的地 Cartesian 座標。 以目前的點位與 Tool, Base 進行可否到達的計算。
is_reachable	bool&	可以到達: True 無法到達: False
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

```
bool is_reachable = false;
double dest_cart[6] = {200,300,200,0,0,0};
motion_reachable (robot, dest_cart, is_reachable)
if(is_reachable){
// .....
}
```

3.10.30. 兩點運動是否可到位- motion_check_lin

```
int motion_check_lin (HROBOT robot , double* point1, double* point2, bool& is_reachable)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
point1	double[6]	第一個點位的 Cartesian 座標
point2	double[6]	第二個點位的 Cartesian 座標
is_reachable	bool&	檢查是否可由點位一使用 LIN 運動到點位二， 此指令只能在 RA 或 RT 機型使用。 可以到達: True 無法到達: False
回傳值	int	成功: 0 非 RA 或 RT 機型: -1 失敗: 錯誤碼

C++:

```
bool is_reachable = false;  
double now_pos[6] = {0};  
double point2[6] = {200,300,200,0,0,0};  
  
get_current_position(robot, now_pos);  
motion_check_lin (robot, now_pos, point2, is_reachable)  
if(is_reachable){  
// .....}
```

3.11. 手臂資訊命令類

函式名稱說明與對應身分參考下表：

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	O	O	O
get_current_joint	取得目前關節座標	O	O	O
get_current_position	取得目前絕對座標	O	O	O
get_current_rpm	取得目前各軸轉速	O	O	O
get_device_born_date	取得裝置出廠時間	O	O	O
get_operation_time	取得控制器開機時間	O	O	O
get_mileage	取得各軸馬達里程數	O	O	O
get_total_mileage	取得各軸馬達累積里程數	O	O	O
get_utilization	取得累積稼動率	O	O	O
get_utilization_ratio	取得稼動率百分比	O	O	O
get_motor_torque	取得馬達負載百分比	O	O	O
get_hrss_version	取得HRSS版本號	O	O	O
get_robot_type	取得機器人型號	O	O	O
set_home_point	設定復歸位置	O	O	×
set_ext_home_point	設定外部軸復歸位置	O	O	×
get_home_point	取得復歸位置	O	O	O
get_ext_home_point	取得外部軸復歸位置	O	O	O
get_previous_pos	取得上次關機位置	O	O	O
get_previous_extpos	取得上次外部軸關機位置	O	O	O
confirm_home_point	點位檢查	O	×	×
enable_joint_soft_limit	啟動關節座標軟體極限	O	O	×
enable_cart_soft_limit	啟動卡式座標軟體極限	O	O	×
set_joint_soft_limit	設定關節座標上下限	O	O	×
set_cart_soft_limit	設定卡式座標上下限	O	O	×
get_joint_soft_limit_config	取得關節座標軟體極限設置	O	O	O
get_cart_soft_limit_config	取得卡式座標軟體極限設置	O	O	O
set_payload_config	設定負載設置	O	O	×
get_payload_config	取得負載設置	O	O	O
set_payload_active	設定有效負載	O	O	×
get_payload_active	取得有效負載	O	O	O
get_home_warning_setting	取得點位警報設置	O	O	O

函式名稱	說明	操作者		觀察者
		手動模式	自動模式	
set_home_warning_setting	設定點位警報設置	O	O	×
get_ext_driver_limit	取得外部軸驅動器極限	O	O	O
set_ext_driver_limit	設定外部軸驅動器極限	O	O	×
get_ext_encoder	取得外部軸編碼器數值	O	O	O
get_robot_dh	取得手臂 DH 值	O	O	×
get_gear_ratio	取得齒輪比	O	O	×
get_robot_data	取得手臂資訊	O	O	×

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.11.1. 取得目前編碼器數值 - get_encoder_count

`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: 錯誤碼

3.11.2. 取得目前關節座標 - get_current_joint

`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，單位：度(degree)。
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

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

3.11.3. 取得目前絕對座標- get_current_position

```
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，單位：毫米(mm)。
回傳值	int	成功:0 失敗: 錯誤碼

C++:

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

3.11.4. 取得目前轉速- get_current_rpm

```
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: 錯誤碼

C++:

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

3.11.5. 取得裝置出廠時間- get_device_born_date

```
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: 錯誤碼

3.11.6. 取得控制器開機時間 - get_operation_time

`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: 錯誤碼

3.11.7. 取得各軸馬達里程數 - get_mileage

`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: 錯誤碼

➤ 馬達旋轉轉數可以借由 HRSS 歸零

3.11.8. 取得各軸馬達累積里程數- get_total_mileage

`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: 錯誤碼

➤ 馬達累積旋轉轉數無法借由 HRSS 歸零

3.11.9. 取得累積稼動率- get_utilization

`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]: 稼動率日 ult [3]: 稼動率時 ult [4]: 稼動率分 ult [5]: 稼動率秒
回傳值	Int	成功: 0 失敗: 錯誤碼

3.11.10. 取得稼動率百分比- get_utilization_ratio

`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: 稼動率百分比 失敗: 錯誤碼

3.11.11. 取得馬達負載百分比- get_motor_torque

```
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: 錯誤碼

3.11.12. 取得 HRSS 版本號- get_hrss_version

```
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: 錯誤碼

C++:

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

3.11.13. 取得 HRSS 連線版本- get_hrss_sdkver

void get_hrss_sdkver (HROBOT robot, int& large_ver, int& small_ver, int& revision)

參數	資料型態	說明
robot	int	連線編號
large_ver	int	大版本連線號碼
small_ver	int	小版本連線號碼
revision	int	版本戳記
備註	<ul style="list-style-type: none"> ➤ 從 SDK3.0.1 與 HRSS 3.3.16 開始，SDK 連線會依據此版本號碼與機器人連線配對，當與機器人的 SDK 版本不相符會有下列情況。 ➤ 大版本連線號碼不同則無法連線，需要更新 SDK 才可以使用。 ➤ 小版本連線號碼不同可以正常連線，會有新指令無法在舊版 SDK 使用的情況。 ➤ 版本戳記，無特殊功能。 	

C++:

```
int client_L=0, server_L=0;
int client_s, server_s=0;
int rev;
get_hrsdk_sdkver (client_L, client_s, rev);
get_hrss_sdkver (robot, server_L, server_s, rev);

if(client_L != server_L){
    printf("Large version does not match and can't be connected.");
}else if(client_s != server_s){
    printf("small version does not match.");
}
```

3.11.14. 取得機器人型號- get_robot_type

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

參數	資料型態	說明
robot	HROBOT	裝置的 ID
robType	char*	機器人型號
回傳值	Int	成功: 0 失敗: 錯誤碼

C++:

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

3.11.15. 設定復歸位置- set_home_point

```
int set_home_point (HROBOT robot, double* joint)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
joint	double[6]	各關節座標 A1~A6
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.16. 設定外部軸復歸位置- set_ext_home_point

```
int set_ext_home_point (HROBOT robot, double* ext_pos)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
ext_pos	double[3]	各外部軸座標 E1~E3
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.17. 取得復歸位置 - get_home_point

```
int get_home_point (HROBOT robot, double* joint)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
joint	double[6]	各關節座標 A1~A6
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.18. 取得外部軸復歸位置 - get_ext_home_point

```
int get_ext_home_point (HROBOT robot, double* ext_pos)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
ext_pos	double[3]	各外部軸座標 E1~E3
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.19. 取得上次關機位置 - get_previous_pos

```
int get_previous_pos (HROBOT robot, double* joint)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
joint	double[6]	各關節座標 A1~A6
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.20. 取得上次外部軸關機位置 - get_previous_extpos

```
int get_previous_extpos (HROBOT robot, double* ext_pos)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
ext_pos	double[3]	各外部軸座標 E1~E3
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

```
double joint[6] = {20, 20, 0, 0, 0, 0};
double recv[6] = {0, 0, 0, 0, 0, 0};
double ext_pos = {10, 20, 30};
double recv_ext = {0, 0, 0};

set_home_point(robot, joint);
set_ext_home_point(robot, ext_pos);
get_home_point(robot, recv);
get_ext_home_point(robot, ext_pos);
get_previous_pos(robot, recv);
get_previous_extpos(robot, recv_ext);
```

3.11.21. 點位檢查- confirm_home_point

`int confirm_home_point(HROBOT s)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
回傳值	int	成功: 0 檢查失敗: 2004 失敗: 錯誤碼
備註	開機前與關機位置不相同(碰撞到 Robot)，出現 Alarm 01-04-30 "Start pos delination error"，需要進行點位檢查。 1. 移動到 Home 點 2. 使用此 API，進行點位檢查	

3.11.22. 啟動關節座標軟體極限- enable_joint_soft_limit

`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: 錯誤碼

3.11.23. 啟動卡式座標軟體極限- enable_cart_soft_limit

`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: 錯誤碼

3.11.24. 設定關節座標上下限- set_joint_soft_limit

`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[6]	下限各關節座標
high_limit	double[6]	上限各關節座標
回傳值	int	成功: 0 失敗: 錯誤碼
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。

3.11.25. 設定卡式座標上下限- set_cart_soft_limit

`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[3]	下限各卡式座標
high_limit	double[3]	上限各卡式座標
回傳值	int	成功: 0 失敗: 錯誤碼
備註		<ul style="list-style-type: none"> ➤ 卡式座標極限可設定 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。

3.11.26. 取得關節座標軟體極限設置 - get_joint_soft_limit_config

```
int get_joint_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[6]	下限各關節座標
high_limit	double[6]	上限各關節座標
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.27. 取得卡式座標軟體極限設置 - get_cart_soft_limit_config

```
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[3]	下限各卡式座標
high_limit	double[3]	上限各卡式座標
回傳值	int	成功: 0 失敗: 錯誤碼

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[3] = { -100, 300, -100, 0, 0, 0 };
double cart_high_limit[3] = { 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);
```

3.11.28. 設定負載設置 - set_payload_config

`int set_payload_config(HROBOT s, int index, double* value, char* comment)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	編號範圍: [0-19]
value	double[7]	➤ 長度為 7 的陣列 0: 質量(kg), 不可為 0 中心(mm) 1: Xc 2: Yc 3: Zc 轉動慣量(kg*mm^2) 4: Ixx 5: Iyy 6: Izz
comment	char*	設定負載註解
回傳值	int	成功: 0 超出 RS405 允許值: 2 失敗: 錯誤碼

3.11.29. 取得負載設置 - get_payload_config

`int get_payload_config (HROBOT s, int index, double* value, char* comment)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	編號範圍: [0-19]
value	double[7]	➤ 長度為 7 的陣列 0: 質量(kg) 1: Xc 2: Yc 3: Zc 4: Ixx 5: Iyy 6: Izz
comment	char*	取得負載註解
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.30. 設定有效負載- set_payload_active

`int set_payload_active(HROBOT s, int index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	編號範圍: [0-19]
回傳值	int	成功: 0 失敗: 錯誤碼
備註	把此編號的負載設為有效，只能有一個啟用的負載	

3.11.31. 取得有效負載- get_payload_active

`int get_payload_active(HROBOT s, int& index)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int&	編號範圍: [0-19]
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

```
int value = -1;
int index = 5;
int& re_index = -1;
char* comment = "payload comment";
double value[7] = {1,2,3,4,5,6,7 };

set_payload_config(robot, index, value, comment);
get_payload_config(robot, index, value, comment);
set_payload_active(robot, index);
get_payload_active(robot, re_index);
```

3.11.32. 取得點位警報設置 - get_home_warning_setting

```
int get_home_warning_setting (HROBOT robot, double* allow_error, double*
near_home)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
allow_error	double[9]	允許關機前與開機後角度差距大於設定值才會發出 [Start pos delination error]，錯誤碼 01-04-30 警報。 此陣列包含 9 個空間 A1~A6 與 E1~E3 allow_error[0]: A1 allow_error[1]: A2 allow_error[2]: A3 allow_error[3]: A4 allow_error[4]: A5 allow_error[5]: A6 allow_error[6]: E1 allow_error[7]: E2 allow_error[8]: E3
near_home	double[9]	當發出點位警報 [Start pos delination error] 需要 JOG 回原點，若是手臂位置與 Home 點差距在此設定的角度值內，即可用 Confirm Home Point 清除警報。 此陣列包含 9 個空間 A1~A6 與 E1~E3
回傳值	int	成功: 0 失敗: 錯誤碼

備註：關節座標軸中手臂本體標示 Joint1(J1)軟體表示為 A1、Joint2(J2)軟體表示為 A2、Joint3(J3)軟體表示為 A3...以此類推至 Joint6。

3.11.33. 設定點位警報設置 - set_home_warning_setting

```
int set_home_warning_setting (HROBOT robot, double* allow_error, double*
near_home)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
allow_error	double[9]	允許關機前與開機後角度差距大於設定值才會發出 [Start pos delination error]，錯誤碼 01-04-30 警報。 此陣列包含 9 個空間 A1~A6 與 E1~E3 allow_error[0]: A1 allow_error[1]: A2 allow_error[2]: A3 allow_error[3]: A4 allow_error[4]: A5 allow_error[5]: A6 allow_error[6]: E1 allow_error[7]: E2 allow_error[8]: E3
near_home	double[9]	當發出點位警報 [Start pos delination error] 需要 JOG 回原點，若是手臂位置與 Home 點差距在此設定的角度值內，即可用 Confirm Home Point 清除警報。 此陣列包含 9 個空間 A1~A6 與 E1~E3
回傳值	int	成功: 0 失敗: 錯誤碼

C++:

```
double allow_error [9] = {1, 1, 1, 1, 1, 1, 3, 3, 3};  

double near_home [9] = {1, 1, 1, 1, 1, 1, 3, 3, 3};  
  

set_home_warning_setting (robot, allow_error, near_home);  

get_home_warning_setting (robot, allow_error, near_home);
```

3.11.34. 取得外部軸驅動器極限- get_ext_driver_limit

```
int get_ext_driver_limit (HROBOT robot, int index, bool& enable, bool& inverse, int& negative_num, int& positive_num, bool& N_light, bool& P_light)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	E1 - E3 編號範圍: [0-2]
enable	bool&	0: 關閉功能 1: 開啟功能
inverse	bool&	此選項會使正負燈號亮燈方式相反， 0: 關閉功能，觸發極限亮燈 1: 開啟功能，未觸發亮燈，觸發極限滅燈
negative_num	int&	取得負編碼值監控驅動器的腳位 範圍 [0-31]
positive_num	int&	取得正編碼值監控驅動器的腳位 範圍 [0-31]
N_light	bool&	觸發負驅動器極限亮燈，若[inverse]開啟顯示方 式則相反。 0: 滅燈 1: 亮燈
P_light	bool&	觸發正驅動器極限亮燈，若[inverse]開啟顯示方 式則相反。 0: 滅燈 1: 亮燈
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.35. 設定外部軸驅動器極限- set_ext_driver_limit

```
int set_ext_driver_limit (HROBOT robot, int index, bool enable, bool inverse, int negative_num, int positive_num)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
index	int	E1 - E3 編號範圍: [0-2]
enable	bool	0: 關閉功能 1: 開啟功能
inverse	bool	此選項會使正負燈號亮燈方式相反， 0: 關閉功能，觸發極限亮燈 1: 開啟功能，未觸發亮燈，觸發極限滅燈
negative_num	int	選擇負編碼值監控驅動器的腳位 範圍 [0-31]
positive_num	int	選擇正編碼值監控驅動器的腳位 範圍 [0-31]
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.36. 取得外部軸編碼器數值-get_ext_encoder

```
int get_ext_encoder (HROBOT robot, int32_t* EncCount)
```

參數	資料型態	說明
robot	HROBOT	裝置的 ID
EncCount	int32_t[3]	此陣列大小為 3 個空間，取得 E1 到 E3 的編碼器 數值
回傳值	int	成功: 0 失敗: 錯誤碼

3.11.37. 取得手臂 DH 值 - get_robot_dh

說明：透過此指令可以取得手臂的 D-H 表資訊。

int get_robot_dh(HROBOT s, int type, double dh_value[][6])

參數	資料型態	說明
robot	HROBOT	裝置的 ID
type	int	0: RA (關節式機器手臂，6 軸) 1: RS (史卡拉機器手臂，SCARA) 2: RD (並聯式機器手臂，Delta)
dh_value	double[][6]	RA: 所需空間[6][6] RS: 所需空間[4][6] RD: 所需空間[1][6]
回傳值	int	成功: 0 失敗: 錯誤碼

C++範例檔：

C++:

```
double ra_dh[6][6] = {0};
double rs_dh[4][6] = {0};
double rd_dh[1][6] = {0};

get_robot_dh(robot, 0, ra_dh);
get_robot_dh(robot, 1, rs_dh);
get_robot_dh(robot, 2, rd_dh);
```

3.11.38. 取得齒輪比- get_gear_ratio

說明：透過此指令可以取得手臂的齒輪比資訊。

`int get_gear_ratio(HROBOT s, double gear_ratio[6])`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
gear_ratio	double[6]	齒輪比(%)
回傳值	int	成功: 0 失敗: 錯誤碼

取得手臂資訊

C++範例檔：

C++:

```
double gear_ratio [6] = {0};  
  
get_gear_ratio(robot, gear_ratio);
```

3.11.39. 取得手臂資訊- get_robot_data

說明：透過此指令可以取得手臂的資訊，包括系統資訊、Port 資訊、各軸設定參數資訊。

`int get_robot_data (HROBOT robot, int* sys_info, int* port_info, double* axis_info)`

參數	資料型態	說明
robot	HROBOT	裝置的 ID
sys_info	int[3]	系統資訊。 [0] : Max Speed (mm/s) [1] : Mode [2] : Axis Count
port_info	int[6]	Port 資訊。 [0] : Robot IO [1] : External [2] : Reserved [3] : Driver [4] : Touch Panel [5] : Teach Pendant
axis_info	double[axis_count][7]	各軸資訊。 axis_count : 手臂軸數目。 [0] : Pos To Encoder Direction [1] : PPR [2] : RPM [3] : Pitch [4] : Gear Ratio [5] : High Limit(線性為 mm；旋轉為度) [6] : Low Limit(線性為 mm；旋轉為度)
回傳值	int	成功: 0: 失敗: 錯誤碼

C++範例檔：

C++:

```
int sys_info[3] = {0};
int port_info[6] = {0};
double axis_info[6][7] = {0};

get_robot_data(robot, sys_info, port_info, axis_info);
```

3.12. 通訊設定類

函式名稱說明與對應身分參考下表：

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

以下小節說明各函式名稱指令格式、參數設定與範例說明。

3.12.1. 設定顯示訊息狀態- set_network_show_msg

`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: 錯誤碼
備註	未把此函式設為 ON，會無法顯示收到的訊息。	

3.12.2. 取得顯示訊息狀態- get_network_show_msg

`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: 錯誤碼

3.12.3. 網路連線- network_connect

`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: 錯誤碼

3.12.4. 網路中斷連線- network_disconnect

`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: 錯誤碼

3.12.5. 傳送網路訊息- network_send_msg

`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: 錯誤碼

3.12.6. 接收網路訊息- network_recieve_msg

`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: 錯誤碼
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，並且在 get_robot_info 的頁面選擇設定為 7，才可以收到 CallBack。 ➤ 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。 	

3.12.7. 設定網路配置 - set_network_config

```
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	➤ 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	➤ 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: 錯誤碼

3.12.8. 取得網路配置 - get_network_config

```
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&	取得顯示格式 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: 錯誤碼

3.12.9. 更改網路 IP 位址- **network_change_ip**

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: 錯誤碼

3.12.10. 取得連線狀態 - network_get_state

`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);
}
```

4. 程式編輯回饋錯誤碼說明

使用者可透過變數之回傳值，可了解程式指令的執行狀態，藉此可作問題判定，並對應該數值作問題判定，概略泛用型錯誤碼說明，才可以找到其問題方向，參考下表說明。

概要	說明	代碼
失敗	命令執行失敗，請確認連線狀態是否正常	-11
	回傳機制創建失敗、無法連至 Robot、版本不相符	-1~-4
正常	命令正常完成	0000
無校新密碼	改變密碼所設定的新密碼不能為空或是特殊字	0049
密碼錯誤	輸入錯誤密碼	0050
未授權	授權失敗，請洽客服人員	0100
斷線	此連線已中斷	0110
無檔案在執行	目前機器手臂沒有正在執行的檔案	0150
檔案開啟失敗	檔案開啟失敗或是檔案名稱錯誤，請確認檔案路徑是否正確	0200
檔案不存在	檔案不存在，請確認檔案路徑是否正確	0201
檔案發送錯誤	網路通訊發生異常	0202
檔案通訊錯誤	傳輸檔案過大或是檔案傳輸通道發生異常	0206
接收檔案封包異常	檔案傳輸通道發生異常	0207
檔案寫入錯誤	檔案傳輸有問題時，會發生寫入錯誤	0208
檔案下載逾時	傳輸檔案過大或是檔案傳輸通道發生異常	0209
檔名錯誤	請確認檔案名稱是否有特殊字元	0210
檔名過長	檔案名稱過長	0211
命令逾時	傳輸命令逾時，連線異常或是加長命令傳輸間隔	0212
沒有警報發生	沒有警報發生，無法清除警報	0300
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
字串過長	輸入的字串長度過長	2007
模式錯誤	目前狀態不在EXT模式	2008
命令不一致	傳送的命令與接收命令編號不一致，請再發送一次	2009
狀態非閒置	當前狀態還在設定參數中或運動中	2010
控制者已經存在	控制者已經存在	2100
命令傳輸錯誤	網路通訊發生問題，命令接收/傳輸錯誤	3000

概要	說明	代碼
概要	說明	代碼
命令接收錯誤	網路通訊發生問題	3001
IO被使用	所設定的IO已經被使用	3100
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
外部軸沒開啟	所下達的指令因為外部軸沒開啟而無法執行。	4005
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
更新檔案異常	更新檔案失敗	4020
	更新檔案傳送失敗	4021
	更新檔案接收失敗	4022
	HRSS硬碟容量不足，請刪除不必要的檔案	4023
	HRSS更新檔案不存在，請確定路徑是否正確	4024
功能異常	本功能異常	9999

■ 範例說明：

以連線(open_connection)來作範例：

```
int device_id = HRobot.open_connection("127.0.0.1", 1, callback);
Console.WriteLine("回傳值 device_id 結果：" + device_id);
```

device_id 若回傳 0 表示正常，若以下回傳值-1~-4 則代表有問題：

回傳值結果	說 明
0	命令正常完成，連線成功
-1	命令執行失敗，請確認連線狀態是否正常，連線失敗
-2	回傳機制創建失敗
-3	無法連至 Robot
-4	HRSSDK與HRSS版本不相符

索引

指令名稱	頁碼
base_calibration	104
calibration	62
callback_function	28
circ_axis	131
circ_pos	130
circ_pr	132
clear_alarm	118
confirm_home_point	152
define_base	100
define_tool	101
delete_file	113
delete_folder	113
disconnect	25
download_file	112
enable_cart_soft_limit	153
enable_joint_soft_limit	152
ext_asyptp	141
ext_lin_axis	139
ext_lin_pos	140
ext_mastering	60
ext_ptp_axis	138
ext_ptp_pos	138
ext_task_start	107
file_drag	114
file_rename	114
get_acc_time	47
get_alarm_code	51
get_base_data	100
get_base_number	99
get_cart_soft_limit_config	154
get_command_count	136
get_command_id	135
get_connection_level	26

指令名稱	頁碼
get_controller_time	55
get_counter	37
get_counter_comment_range	86
get_counter_name	38
get_counter_value_all	86
get_current_ext_mode	61
get_current_ext_pos	61
get_current_joint	144
get_current_position	145
get_current_rpm	145
get_device_born_date	145
get_DI_comment_range	82
get_DI_range	81
get_DI_sim_range	82
get_DI_simulation_Enable	68
get_digital_input	67
get_digital_input_comment	69
get_digital_output	68
get_digital_output_comment	70
get_digital_setting	54
get_DO_comment_range	84
get_DO_range	83
get_encoder_count	144
get_execute_file_name	111
get_ext_axis_setting	57
get_ext_axis_setting_advanced	58
get_ext_driver_limit	159
get_ext_encoder	160
get_ext_home_point	151
get_FI_all	84
get_fieldbus_rs_comment_range	88
get_fieldbus_rs_parameter_range	87
get_fieldbus_rs_srr_range	87

指令名稱	頁碼
get_fieldbus_rs_srw_range.....	86
get_FO_all	84
get_function_input	72
get_function_output.....	73
get_gear_ratio.....	162
get_home_point	151
get_home_warning_setting.....	157
get_hrsdk_sdkver.....	27
get_hrsdk_version.....	26
get_hrss_mode	116
get_hrss_sdkver	149
get_hrss_version.....	148
get_joint_soft_limit_config	154
get_lin_speed.....	49
get_MI_comment_range	90
get_MI_config_all	89
get_mileage.....	146
get_MO_comment_range.....	90
get_MO_config_all.....	89
get_module_input_config.....	74
get_module_output_config	75
get_motion_state.....	137
get_motor_state	117
get_motor_torque	148
get_network_config.....	166
get_network_show_msg	164
get_operation_mode	118
get_operation_time	146
get_override_ratio	50
get_payload_active.....	156
get_payload_config.....	155
get_pr.....	41
get_pr_comment	43
get_PR_comment_array	93
get_pr_coordinate	39
get_pr_tool_base.....	41
get_pr_type.....	39

指令名稱	頁碼
get_previous_extpos	151
get_previous_pos	151
get_prog_name	115
get_prog_number.....	115
get_ptp_speed	48
get_RI_all	93
get_RO_all.....	93
get_robot_data	163
get_robot_dh	161
get_robot_info.....	120
get_robot_input.....	70
get_robot_output	71
get_robot_type	150
get_rsr_prog_name	106
get_SI_comment_range.....	92
get_SI_range	91
get_SI_sim_range	91
get_SO_comment_range	92
get_SO_range.....	91
get_system_input_all.....	88
get_timer_comment_range	85
get_timer_name	36
get_timer_status	35
get_timer_status_all	85
get_timer_value_all.....	85
get_tool_data.....	101
get_tool_number	101
get_total_mileage.....	147
get_user_alarm_setting_message.....	56
get_utilization	147
get_utilization_ratio	147
get_valve_output	71
get_VO_all.....	94
jog	122
jog_home	123
jog_stop	123
lin_axis	128

指令名稱	頁碼
lin_pos.....	127
lin_pr.....	129
lin_rel_axis	129
lin_rel_pos.....	128
mastering	62
motion_abort.....	133
motion_check_lin.....	142
motion_continue	133
motion_delay.....	133
motion_hold.....	132
motion_reachable.....	142
network_change_ip	167
network_connect.....	165
network_disconnect	165
network_get_state.....	168
network_recieve_msg	165
network_send_msg.....	165
new_folder.....	114
open_connection	25
ptp_axis.....	125
ptp_pos	125
ptp_pr	126
ptp_rel_axis.....	126
ptp_rel_pos.....	126
remove_command	137
remove_command_tail.....	137
remove_pr.....	42
remove_rsr.....	106
save_module_io_setting.....	79
send_file.....	113
set_acc_dec_ratio	46
set_acc_time	47
set_base_number	99
set_cart_soft_limit.....	153
set_command_id	135
set_connection_level.....	26
set_counter.....	37

指令名稱	頁碼
set_counter_array.....	96
set_DI_array.....	94
set_DI_sim_array	95
set_DI_simulation	67
set_DI_simulation_Enable	67
set_digital_input_comment	69
set_digital_output	68
set_digital_setting	52
set_DO_array	95
set_ext_axis_setting	57
set_ext_axis_setting_advanced.....	59
set_ext_driver_limit	160
set_ext_home_point	150
set_fieldbus_srw_array	97
set_home_point	150
set_home_warning_setting.....	158
set_joint_soft_limit	153
set_language.....	55
set_lin_speed	49
set_MO_array	97
set_module_input_comment	77
set_module_input_config.....	75
set_module_input_end	76
set_module_input_start	76
set_module_input_type	78
set_module_input_value.....	76
set_module_output_comment	78
set_module_output_end	78
set_module_output_start.....	77
set_module_output_type	79
set_module_output_value	77
set_motor_state	117
set_network_config	166
set_network_show_msg	164
set_operation_mode	117
set_override_ratio	49
set_payload_active	156

指令名稱	頁碼
set_payload_config	155
set_pr_comment	43
set_pr_tool_base	39
set_pr_type	38
set_ptp_speed	47
set_RO_array	98
set_robot_id	50
set_robot_output	71
set_rsr	105
set_SI_array	96
set_SI_sim_array	96
set_smooth_length	51
set_SO_array	97
set_timer	34
set_timer_name	36

指令名稱	頁碼
set_timer_start	34
set_timer_stop	35
set_timer_value_array	95
set_tool_number	100
set_user_alarm_setting_message	56
set_valve_output	72
set_VO_array	98
SyncOutput	80
task_abort	109
task_continue	109
task_hold	109
task_start	107
tool_calibration	103
update_hrss	119

機器人軟體開發套件使用手冊

出版日期：2022年07月

-
1. HIWIN 為上銀科技的註冊商標，請勿購買來路不明之仿冒品以維護您的權益。
 2. 本型錄所載規格、照片有時會與實際產品有所差異，包括因為改良而導致外觀或規格等發生變化的情況。
 3. HIWIN 產品專利清單查詢網址：http://www.hiwin.tw/Products/Products_patents.aspx
 4. 凡受”貿易法”等法規限制之相關技術與產品，HIWIN 將不會違規擅自出售。若要出口 HIWIN 受法律規範限制出口的產品，應根據相關法律向主管機關申請出口許可，並不得供作生產或發展核子、生化、飛彈等軍事武器之用。
-



全球銷售暨服務據點

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.us

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

法國 史特拉斯堡

HIWIN FRANCE
STRASBOURG, FRANCE
www.hiwin.fr

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 (銷售)
robotservice@hiwin.tw (客服)