IMP Series

Motion Control Command Library

Integrated Testing Environment

User Manual

Version: V.1.01

Date: 2013.01

http://www.epcio.com.tw

Table of Contents

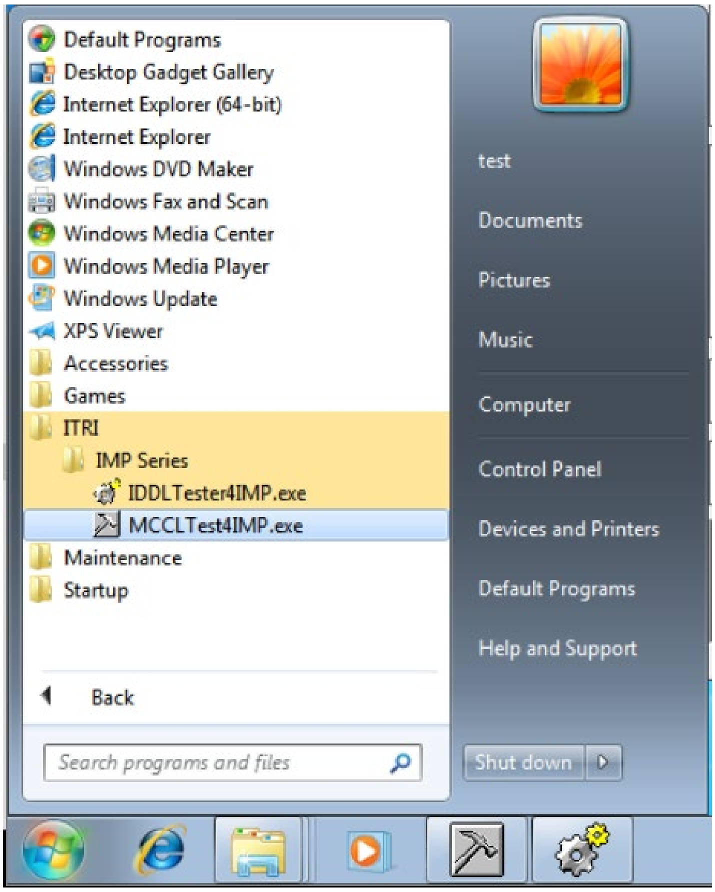
I. INTRODUCTION OF THE MOTION CONTROL COMMAND LIBRARY (MCCL) TESTING SOFTWARE

The motion control command library testing software can be used to test single IMP series motion control card using a single group (For a description of groups, please refer to "**IMP Series Motion Control Command Library User Manual**"); group parameter setting are as follows:

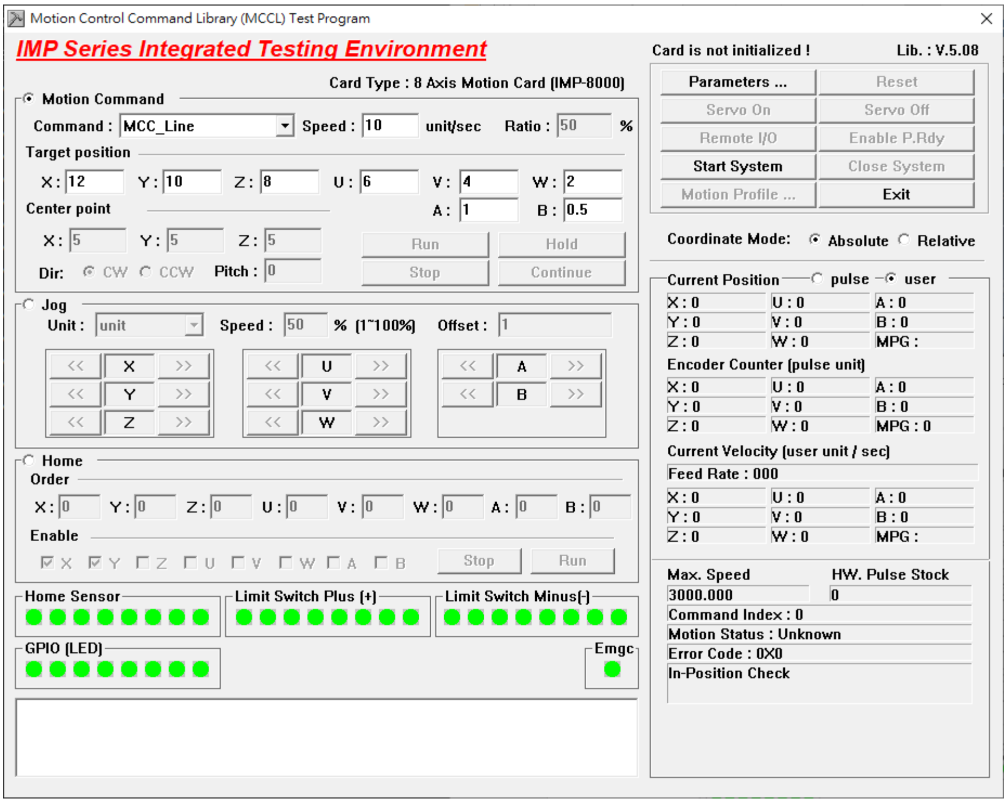
```
m_nGroupIndex = MCC_CreateGroup(
                        0,    // X-axis programming results output from Channel 0
                        1,    // Y-axis programming results output from Channel 1
                        2,    // Z-axis programming results output from Channel 2
                        3,    // U-axis programming results output from Channel 3
                        4,    // V-axis programming results output from Channel 4
                        5,    // W-axis programming results output from Channel 5
                        6,    // A-axis programming results output from Channel 6
                        7,    // B-axis programming results output from Channel 7
                        0);   // Control card number corresponding to this group
```

Therefore, if the group number is required to be input into the command used in the software, m_nGroupIndex shall be applied.

This testing software only uses basic function commands provided by the MCCL. For usage of other commands, please refer to descriptions in "**IMP Series Motion Control Command Library Reference Manual**". For more details regarding usage of commands, please refer to descriptions in "**IMP Series Motion Control Command Library Example Manual**". Following figures will be used to illustrate basic function commands used and simply explain the operation of testing software. Enter the main menu of motion control command library testing software (MCCLTest4IMP) through the following operation.



The following figure shows the main menu of motion control command library testing software (MCCLTest4IMP).



Card is not initialized ! Lib. : V.5.08

Card Type : 8 Axis Motion Card (IMP-8000)

● Motion Command

Command : MCC_Line Speed : 10 unit/sec Ratio : 50 %

Target position

X : 12 Y : 10 Z : 8 U : 6 V : 4 W : 2

A : 1 B : 0.5

Center point

X : 5 Y : 5 Z : 5

Dir: ● CW ○ CCW Pitch : 0

○ Jog

Unit : unit Speed : 50 % (1~100%) Offset : 1

○ Home

Order

X : 0 Y : 0 Z : 0 U : 0 V : 0 W : 0 A : 0 B : 0

Enable

☑ X ☑ Y ☐ Z ☐ U ☐ V ☐ W ☐ A ☐ B

Coordinate Mode: ● Absolute ○ Relative

Current Position ○ pulse ● user

X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG :

Encoder Counter (pulse unit)

X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG : 0

Current Velocity (user unit / sec)

Feed Rate : 000

X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG :

Max. Speed	HW. Pulse Stock
3000.000	0

Command Index : 0

Motion Status : Unknown

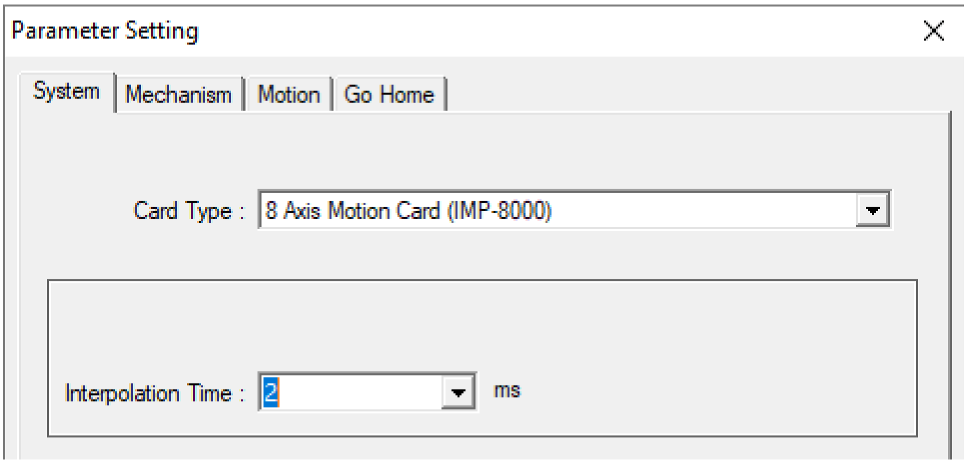
Error Code : 0X0

In-Position Check

II. TESTING SOFTWARE ACTIVATION

It is required to activate the MCCL before testing its functions; parameters can be set by following procedures:

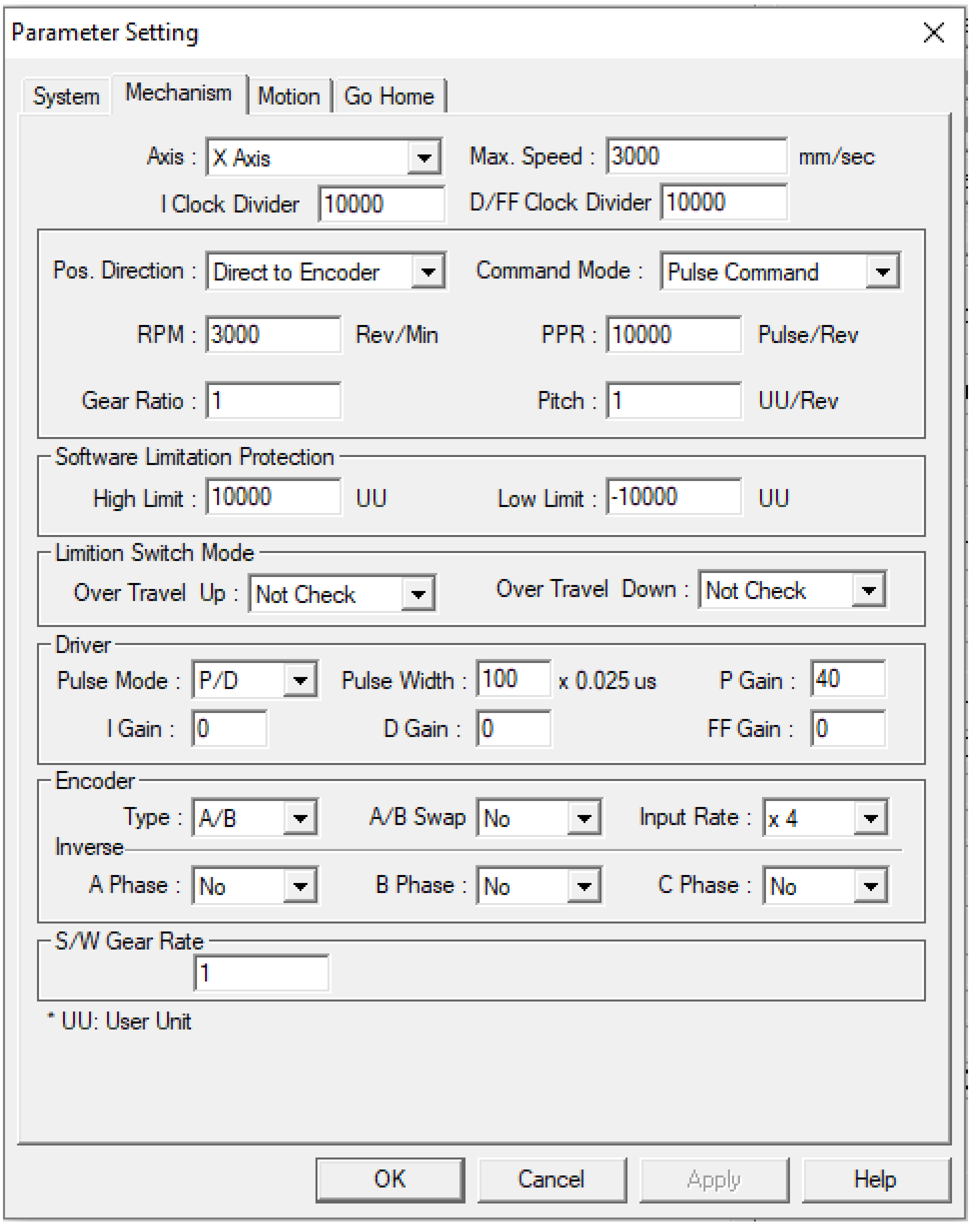
1. Clip "Parameters Seting" on the main menu; checking all settings in **"System" Parameters Setting Page** (see following figure) can verify if the setting of card type is correct. Meanwhile, the interpolation time (suggested value: 2ms) can be set here as well.



Card Type : 8 Axis Motion Card (IMP-8000)

Interpolation Time : 2 ms

2. Checking all settings in "**Mechanism**" **parameters Setting Page** (see the following figure) can verify if settings of mechanism parameter are correct. For the meaning of each parameter in the following figure, please refer to the description in "**IMP Series Motion Control Command Library User Manual**".



Axis : X Axis

Max. Speed : 3000 mm/sec

I Clock Divider 10000

D/FF Clock Divider 10000

Pos. Direction : Direct to Encoder

Command Mode : Pulse Command

RPM : 3000 Rev/Min

PPR : 10000 Pulse/Rev

Gear Ratio : 1

Pitch : 1 UU/Rev

Software Limitation Protection

High Limit : 10000 UU

Low Limit : -10000 UU

Limition Switch Mode

Over Travel Up : Not Check

Over Travel Down : Not Check

Driver

Pulse Mode : P/D

Pulse Width : 100 x 0.025 us

P Gain : 40

I Gain : 0

D Gain : 0

FF Gain : 0

Encoder

Type : A/B

A/B Swap No

Input Rate : x 4

Inverse

A Phase : No

B Phase : No

C Phase : No

S/W Gear Rate

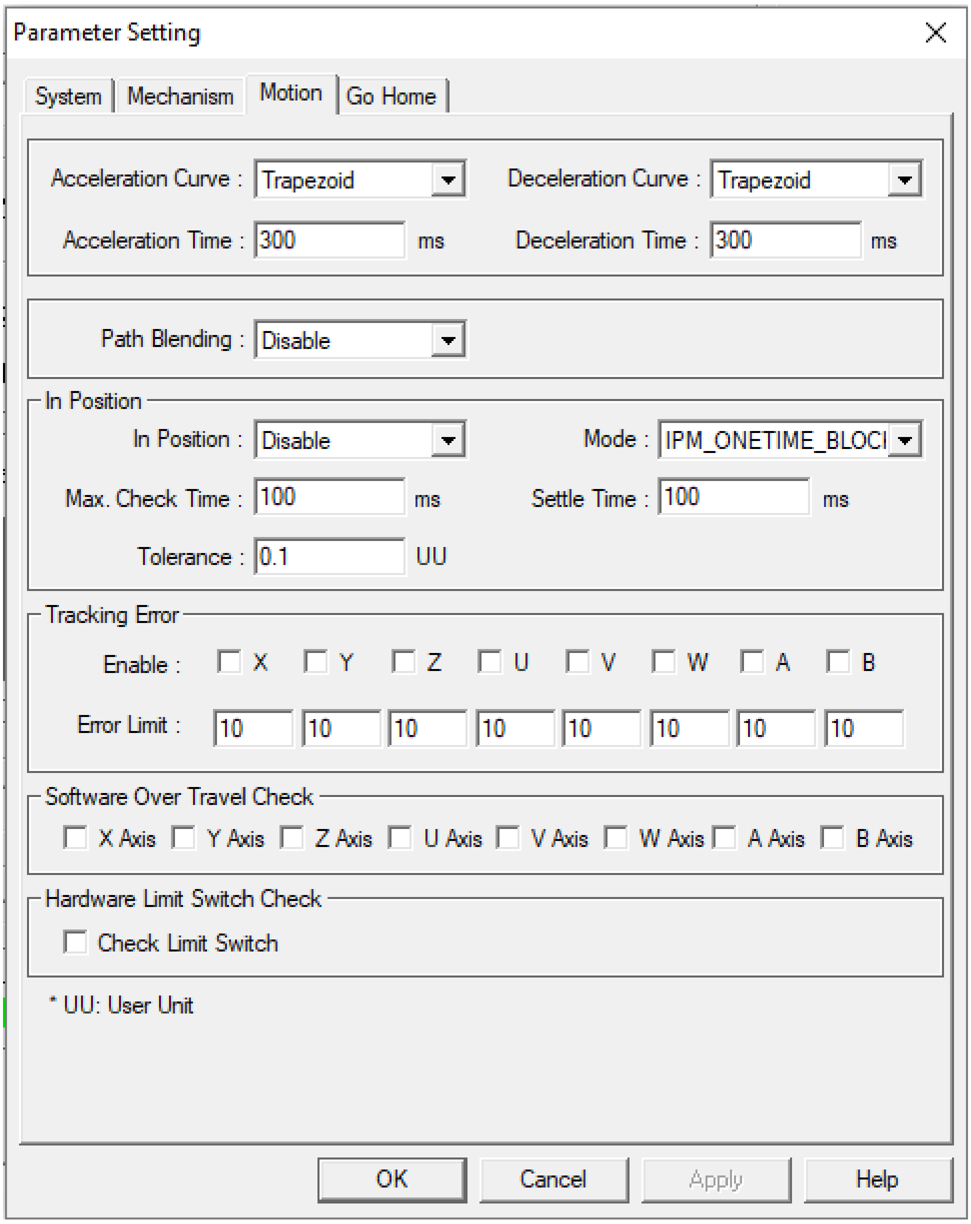
1

* UU: User Unit

3. Click the Start System button in main screen; this operation will set system parameters as well as call MCC_InitSystem().

III. MOTION PROPERTY SETTING

This section will introduce the MCCL commands required by testing software when changing the settings in "**Motion**" **Property Setting Page**. The user can learn the usage of MCCL according to changes in motion trajectory. The following figure is "**Motion**" **Property Setting Page**.



Parameter Setting

Acceleration Curve : Trapezoid
Deceleration Curve : Trapezoid
Acceleration Time : 300 ms
Deceleration Time : 300 ms

Path Blending : Disable

In Position
In Position : Disable
Mode : IPM_ONETIME_BLOCI
Max. Check Time : 100 ms
Settle Time : 100 ms
Tolerance : 0.1 UU

Tracking Error
Enable : □ X □ Y □ Z □ U □ V □ W □ A □ B
Error Limit : 10 10 10 10 10 10 10 10

Software Over Travel Check
□ X Axis □ Y Axis □ Z Axis □ U Axis □ V Axis □ W Axis □ A Axis □ B Axis

Hardware Limit Switch Check
□ Check Limit Switch

* UU: User Unit

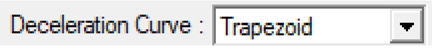
The command calls corresponding to each option are detailed below.

Acceleration Curve

Acceleration Curve : Trapezoid

Set the acceleration type of X, Y, Z, U, V, W, A and B axes when running linear, curve and circular motions. The type can be set as the trapezoidal curve or S curve. MCC_SetAccType('T') will be called when selecting trapezoid, meaning using the trapezoidal acceleration curve. MCC_SetAccType('S') will be called when selecting S, meaning using the S acceleration curve.

Deceleration Curve



Deceleration Curve : Trapezoid

Set the deceleration type of X, Y, Z, U, V, W, A and B axes when running linear, curve and circular motions. The type can be set as the trapezoidal curve or S curve. MCC_SetDecType('T') will be called when selecting trapezoid, meaning using the trapezoidal deceleration curve. MCC_SetAccType('S') will be called when selection S, meaning using the S deceleration curve.

Acceleration Time

Acceleration Time : 300 ms

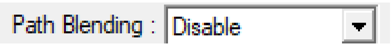
Set the acceleration time; unit: ms. The acceleration time must be set as greater than 0. Suppose the required acceleration time is dfTime, then MCC_SetAccTime (dfTime) can be called.

Deceleration Time

Deceleration Time : 300 ms

Set the deceleration time; unit: ms. The deceleration time must be set as greater than 0. Suppose the required deceleration time is dfTime, then MCC_SetDecTime (dfTime) can be called.

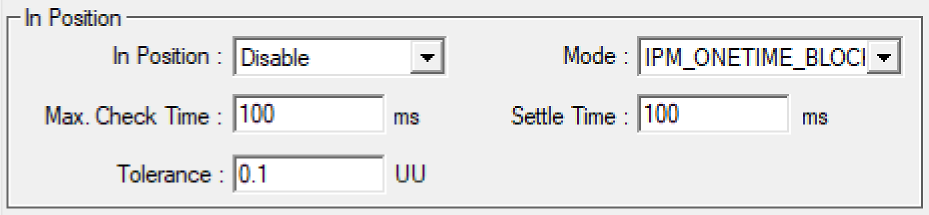
Path Blending



Path Blending : Disable

Enable path blending. Selecting “Disable” will disable path blending by calling MCC_DisableBlend()； selecting “Enable” will enable path blending by calling MCC_EnableBlend().

In Position



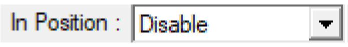
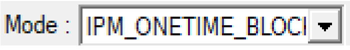
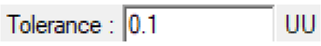
In Position

In Position : Disable ▾ Mode : IPM_ONETIME_BLOCI ▾

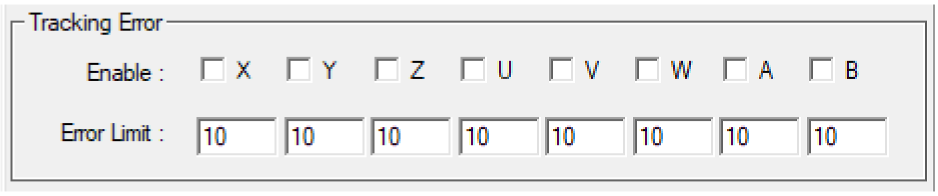
Max. Check Time : 100 ms Settle Time : 100 ms

Tolerance : 0.1 UU

This part is used to enable the in-position confirmation function and set its parameters (For the in-position confirmation function, please refer to **IMP Series Motion Control Command Library User Manual**).

 In Position : Disable ▾	MCC_EnableInPos / MCC_DisableInPos
 Mode : IPM_ONETIME_BLOCI ▾	MCC_SetInPosMode
Max. Check Time : 100 ms	MCC_SetInPosMaxCheckTime
Settle Time : 100 ms	MCC_SetInPosSettleTime
 Tolerance : 0.1 UU	MCC_SetInPosToleranceEx

Tracking Error



Tracking Error

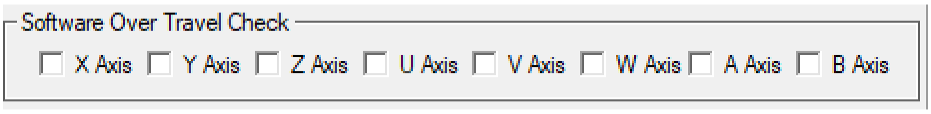
Enable : □ X □ Y □ Z □ U □ V □ W □ A □ B

Error Limit : 10 10 10 10 10 10 10 10

This part is used to enable the tracking error function and set its parameters (for the tracking error function, please refer to **IMP Series Motion Control Command Library User Manual**).

Enable tracking error	MCC_EnableTrackError
Disable tracking error	MCC_DisableTrackError
Set tracking error tolerance	MCC_SetTrackErrorLimit

Software Over Travel Check

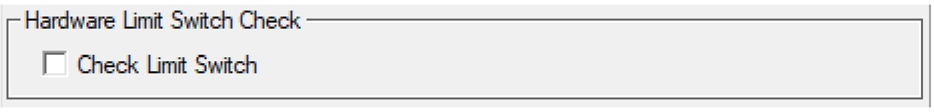


Software Over Travel Check

□ X Axis □ Y Axis □ Z Axis □ U Axis □ V Axis □ W Axis □ A Axis □ B Axis

”**Software Over Travel Check**” uses MCC_SetOverTravelCheck to enable the software over travel check of each axis and can limit the displacement within the work area.

Hardware Limit Switch Check

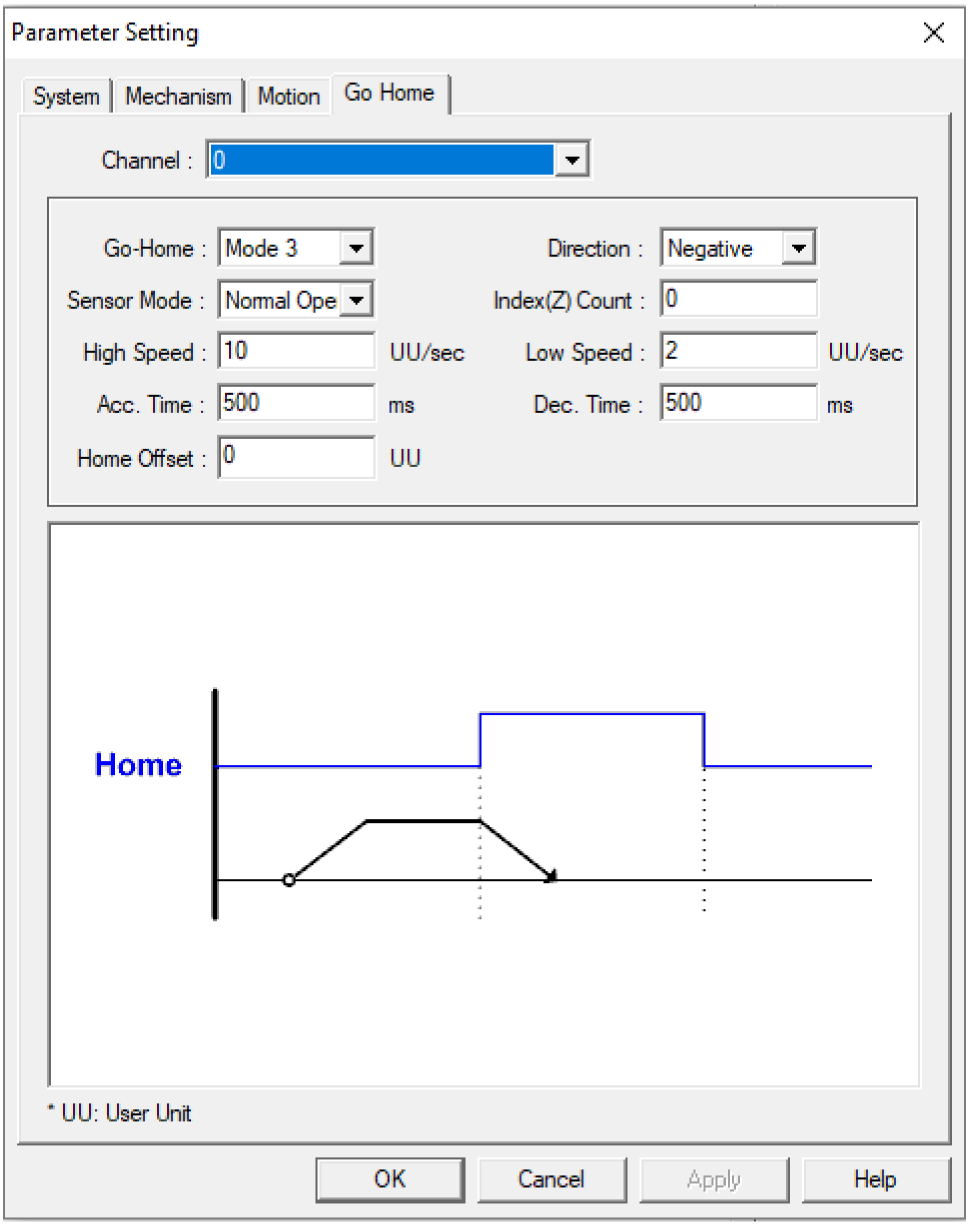


Hardware Limit Switch Check

□ Check Limit Switch

If the status is ☑ Check Limit Switch, then MCC_EnableLimitSwitchCheck will be called to enable the limit switch check; otherwise, MCC_DisableLimitSwitchCheck will be called to disable the limit switch check. The user can call MCC_GetLimitSwitchStatus to check if the limit switch is triggered currently. For using these functions, mechanism parameters *wOverTravelUpSensorMode* and *wOverTravelDownSensorMode* must be correctly set (must be Normal Open or Normal Close).

IV. GO HOME MOTION PROPERTY SETTINGS



Parameter Setting

Channel : 0

Go-Home : Mode 3

Direction : Negative

Sensor Mode : Normal Ope

Index(Z) Count : 0

High Speed : 10 UU/sec

Low Speed : 2 UU/sec

Acc. Time : 500 ms

Dec. Time : 500 ms

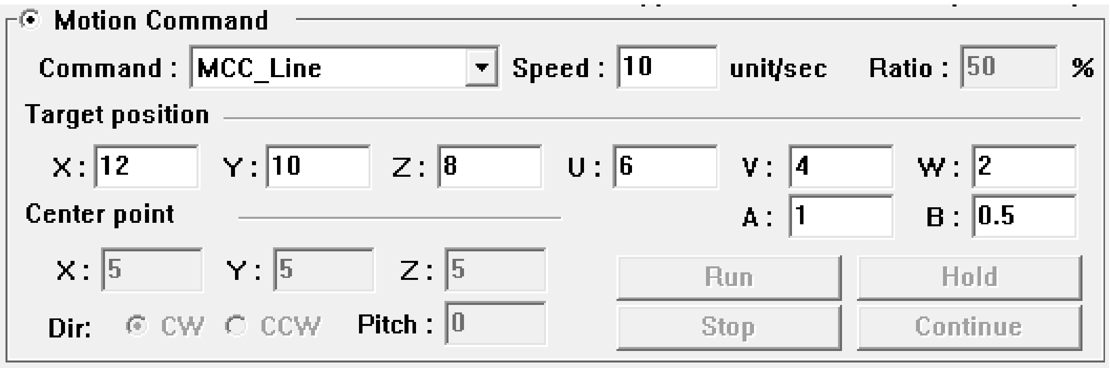
Home Offset : 0 UU

* UU: User Unit

The above figure shows property settings related to Go Home motion. These settings will directly reflect the Go Home parameters. For details, please refer to "**IMP Series Motion Control Command Library User Manual**".

V. MOTION COMMAND EXECUTION

The following figure shows “**Motion**” **Command Parameter Setting** section. Content related to general motion operation is introduced below.



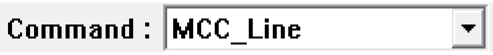
● Motion Command

Command : MCC_Line
Speed : 10 unit/sec
Ratio : 50 %

Target position
X : 12 Y : 10 Z : 8 U : 6 V : 4 W : 2
A : 1 B : 0.5

Center point
X : 5 Y : 5 Z : 5
Dir: ● CW ○ CCW
Pitch : 0

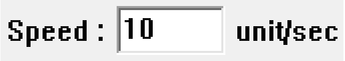
Motion Command Option:



Command : MCC_Line

The motion command type can be selected here. The option content and command name are the same, including point-to-point, linear, circular, curve and helical motions.

Speed Setting:



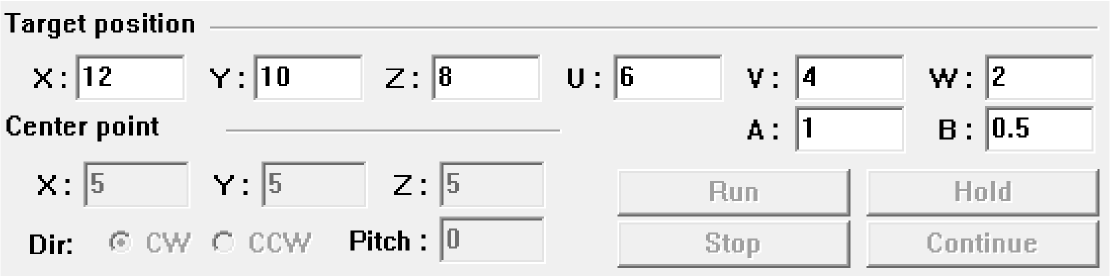
Speed : 10 unit/sec

is used to set the feed speed; unit: unit/sec. This input value will act as the call parameter of MCC_SetFeedSpeed and should not be smaller than or equal 0.

Ratio : 50 %

sets the point-to-point speed ratio. The input value is ranging from 1~100 and will act as the call parameter of MCC_SetPtPSpeed.

Parameters:



Target position
X : 12 Y : 10 Z : 8 U : 6 V : 4 W : 2
A : 1 B : 0.5

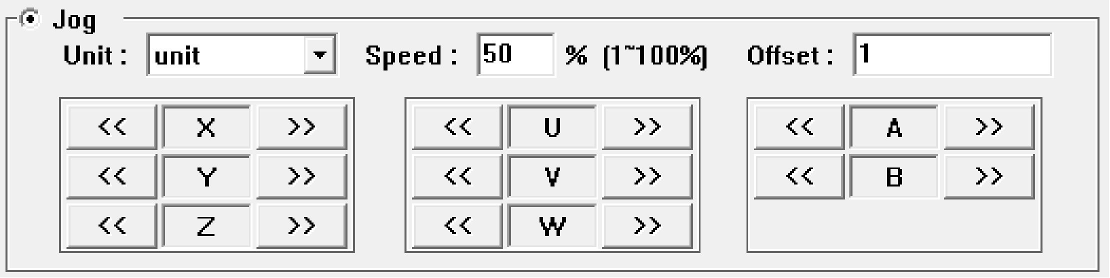
Center point
X : 5 Y : 5 Z : 5
Dir: ● CW ○ CCW
Pitch : 0

“**Target position**” and “**Center point**” in the above figure are the required parameters for calling the commands listed above. For details, please refer to "**IMP Series Motion Control Command Library Reference Manual**".

After all parameters are correctly set, click the Run button to send the motion command to motion command queue. Clicking the Hold button can call MCC_HoldMotion to temporarily stop motion; clicking the Continue button can call MCC_ContiMotion to resume the paused motion command. Clicking the Stop button can call MCC_AbortMotionEx to abort the current motion and delete the commands stored in motion command queue.

VI. JOG MOTION

The following figure shows "**JOG**" **Motion Parameter Setting** section. Options related to the JOG motion are introduced below.



● Jog

Unit : unit

Speed : 50 % (1~100%)

Offset : 1

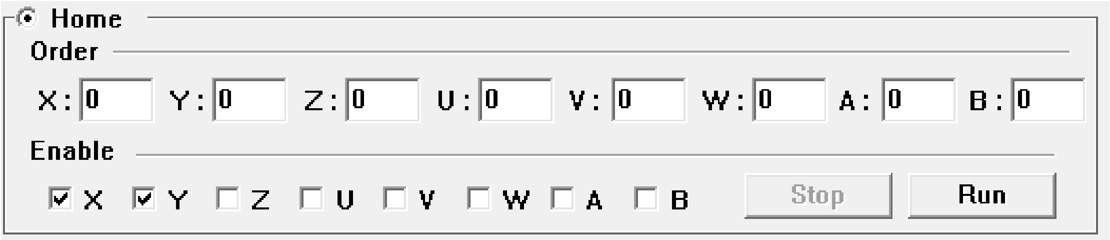
Displacement Unit:

If selecting "**Unit**" as the displacement unit, then the JOG motion will use User Unit as the displacement unit. Moreover, when using the JOG motion control button (such as clicking), the specified axis will be driven according to the designated incremental displacement value (the input value in) and the feed speed ratio (feed speed is the input value in multiplied by (RPM / 60) × Pitch / GearRatio of each axis). At this point, MCC_JogSpace() will be called.

If selecting "pulse" as the displacement unit, then the JOG motion will use pulse as the displacement unit. At this point, if the system is under motion stop status and the JOG motion control button is used, the specified axis will be driven according to the designated pulse displacement and direction. The setting of pulse displacement should not be overly large (cannot exceed 2048 pulses). At this point, MCC_JogPulse() will be called

VII. GO HOME MOTION

The following figure shows “**Go Home Parameters Setting**” section. Options related to the Home motion operation are introduced below.



● Home

Order

X: 0 Y: 0 Z: 0 U: 0 V: 0 W: 0 A: 0 B: 0

Enable

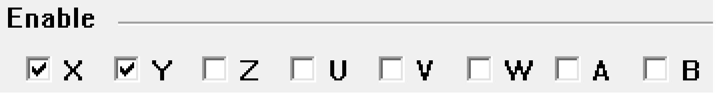
☑ X ☑ Y □ Z □ U □ V □ W □ A □ B

Command declarations regarding the Go Home motion are as follows:

```
MCC_Home(   int   nOrder0, int   nOrder1, int   nOrder2,
            int   nOrder3, int   nOrder4, int   nOrder5,
            int   nOrder6, int   nOrder7, WORD wCardIndex);
```

```
MCC_GetGoHomeStatus();
```

Where MCC_Home() can take the machine to home position. MCC_GetGoHomeStatus() can be used with this command to check if the Go Home motion has completed. nOrder0~ nOrder7 respectively represents the reset sequence of X, Y, Z, U, V, W, A and B axes. The setting value of reset sequence ranges between 0 ~ 7. These parameters can be obtained from “**Go Home Parameters Setting**” section.

The feed speed unit of each axis is User Unit. The reset sequence of motion axes not executing Go Home motion needs to be set as 0xff(255). In  figure, the reset sequence unselected motion axes will be set as 0xff.

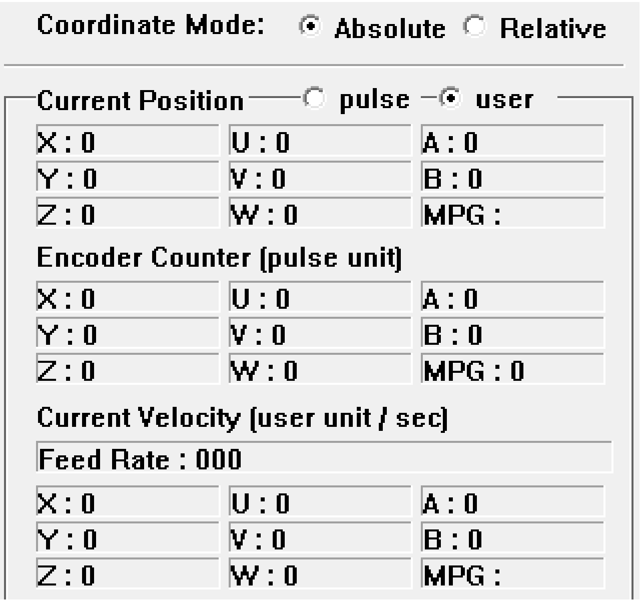
Enable

☑ X ☑ Y □ Z □ U □ V □ W □ A □ B

After all parameters are correctly set, click the Run button to call MCC_Home and execute the Go Home motion. During execution, click the Stop button to call MCC_AbortGoHome and stop the Go Home motion.

VIII. MOTION STATUS AND INFORMATION DISPLAY

The following figure shows “**Motion Status Information Display**” section. Methods for obtaining each information are introduced below.



Coordinate Mode: ● Absolute ○ Relative

Current Position ○ pulse ● user

X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG :

Encoder Counter (pulse unit)

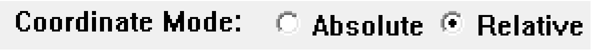
X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG : 0

Current Velocity (user unit / sec)

Feed Rate : 000

X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG :

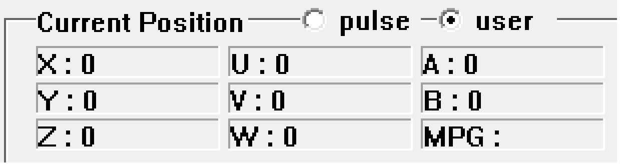
Coordinate Mode



Coordinate Mode: ○ Absolute ● Relative

Set coordinate values of each axis to be shown in absolute coordinate mode or relative coordernate mode (incremental). When selecting “Absolute”, MCC_SetAbsolute() will be called ; when selecting “Relative”, MCC_SetIncrease() will be called.

Cartesian Coordinates for the Current Position Command of Each Axis:

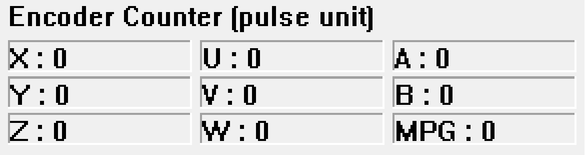


Current Position ○ pulse ● user

X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG :

MCC_GetCurPos can be used to acquire the Cartesian coordinates for the current position of each axis

Encoder Count of the Current Position of Each Axis:

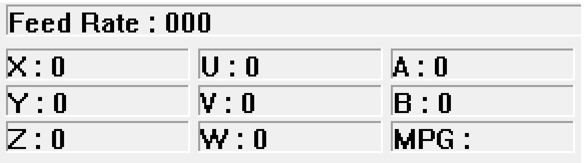


Encoder Counter (pulse unit)

X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG : 0

If an encoder is installed in the system, the user can use MCC_GetENCValue() to acquire the encoder count for current position of each axis

Current Actual Feed Speed (Feed Rate) and Speed of Each Axis:

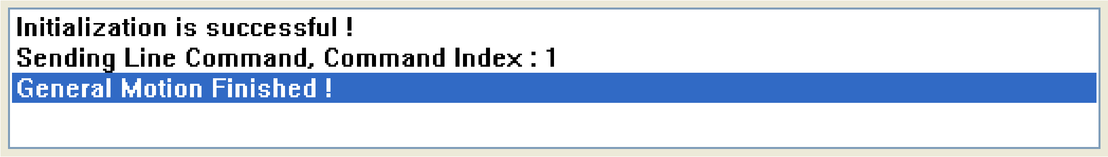


Feed Rate : 000

X : 0	U : 0	A : 0
Y : 0	V : 0	B : 0
Z : 0	W : 0	MPG :

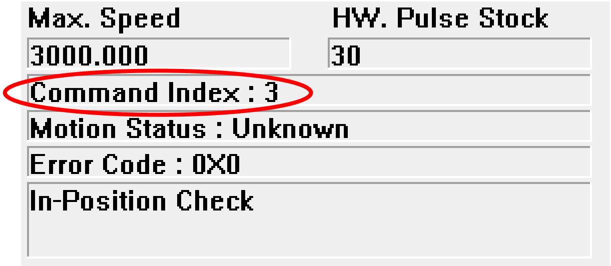
Calling MCC_GetCurFeedSpeed and MCC_GetSpeed can acquire the current feed Speed of general motion (excluding point-to-point motion) and the current speed of each axis

Information Window:



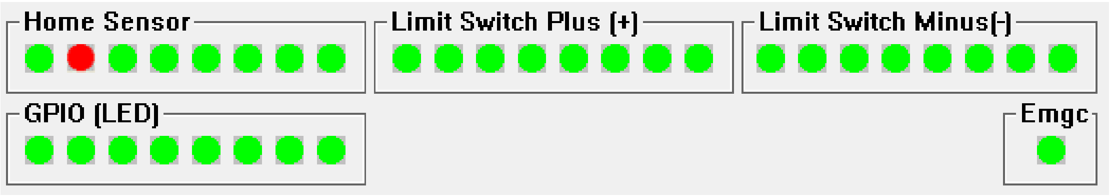
Initialization is successful !
Sending Line Command, Command Index : 1
General Motion Finished !

The "**Information Window**" shows the current motion status and index of motion commands sent to the motion command queue when the "**Run**" button in "**Motion Command Parameter Setting**" section is clicked. The index can be obtained from the command return value (such as the command return value of calling MCC_Line). Information related to motion commands currently being executed, including motion command indexes, can be acquired by using MCC_GetCurCommand. The index of motion command being executed is shown as follows.



Max. Speed 3000.000
HW. Pulse Stock 30
Command Index : 3
Motion Status : Unknown
Error Code : 0X0
In-Position Check

The following figure shows **“Home Sensor & Limit Switch Sensor & GPIO (LED) & Emergency Stop Status**” display area used to display the statuses of these inputs .



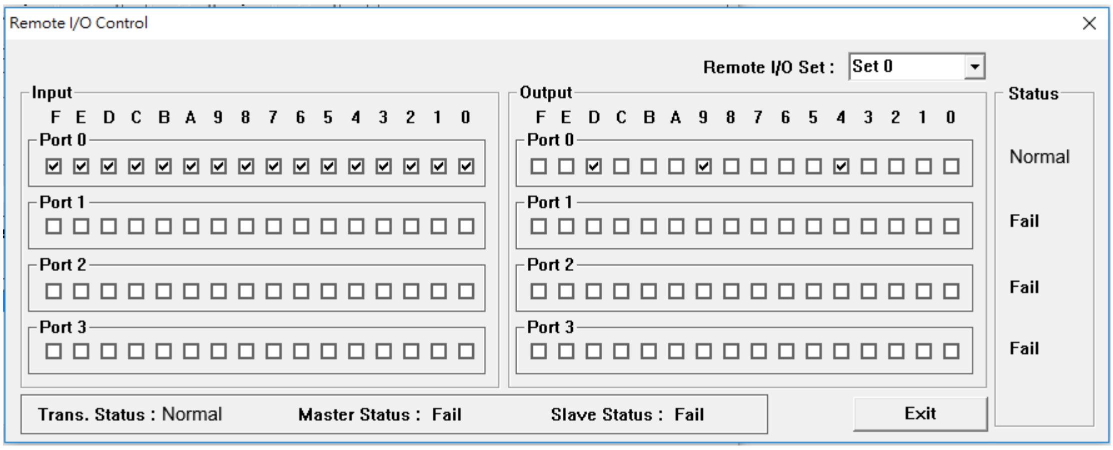
MCC_GetLimitSwitchStatus, MCC_GetGoHomeStatus, MCC_LIO_GetLedLightStatus and MCC_GetEmgcStopStatus can be used to acquire the statuses of the home sensor, limit switch sensor, GPIO (LED) and emergency stop.

IX. REMOTE I/O TEST

If the system is installed with an Ayschronous Remote I/O control card, once the system is successfully initialized, click the Remote I/O button to read the remote I/O control window. Note: After the system is successfully initialized by using MCC_InitSystem, it is still required to call following commands to normally use functions of remote I/O. The commands include:

MCC_EnableRIOSetControl();
MCC_EnableRIOSlaveControl()

Following is the remote I/O control window



Remote I/O Control

Remote I/O Set : Set 0

	Input F E D C B A 9 8 7 6 5 4 3 2 1 0	Output F E D C B A 9 8 7 6 5 4 3 2 1 0	Status
Port 0	☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑	□ □ ☑ □ □ □ ☑ □ □ □ □ ☑ □ □ □ □	Normal
Port 1	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	Fail
Port 2	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	Fail
Port 3	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	Fail

Trans. Status : Normal Master Status : Fail Slave Status : Fail

MCC_GetRIOInputValue and MCC_SetRIOOutputValue can be used to acquire and set the remote I/O signal status respectively.