



User's

Manual of ISaGRAF®

Embedded Controllers

Chapter 18 : Motion

18.1: Install motion driver

Limitation:

1. I-8437/8837 **CAN NOT** do ethernet communication when using I-8091 to do motion control, while W-8337/8737 doesn't have this limitation.
2. Only one I-8091 board in I-8417/8817/8437/8837 & W-8337/8737 can do X-Y dependent motion, other I-8091s should be moving independent. Or all I-8091s are moving independent.

The I-8417/8817/8437/8837 & Wincon-8337/8737 can integrate with the I-8091 to do Motion control. The default ISaGRAF driver burned in the Flash memory of the I-8417/8817/8437/8837 controller is for general usage not for motion control. Please update it to the motion driver by yourself. While user don't need to upgrade the driver of Wincon-8337/8737 if its driver version is 3.08 or higher.

The motion driver of I-8417/8817/8437/8837 can be found in the CD-ROM.

Please refer to the "ReadMe.txt" in the folder of "motion?.??" (for ex. "Motion2.45")

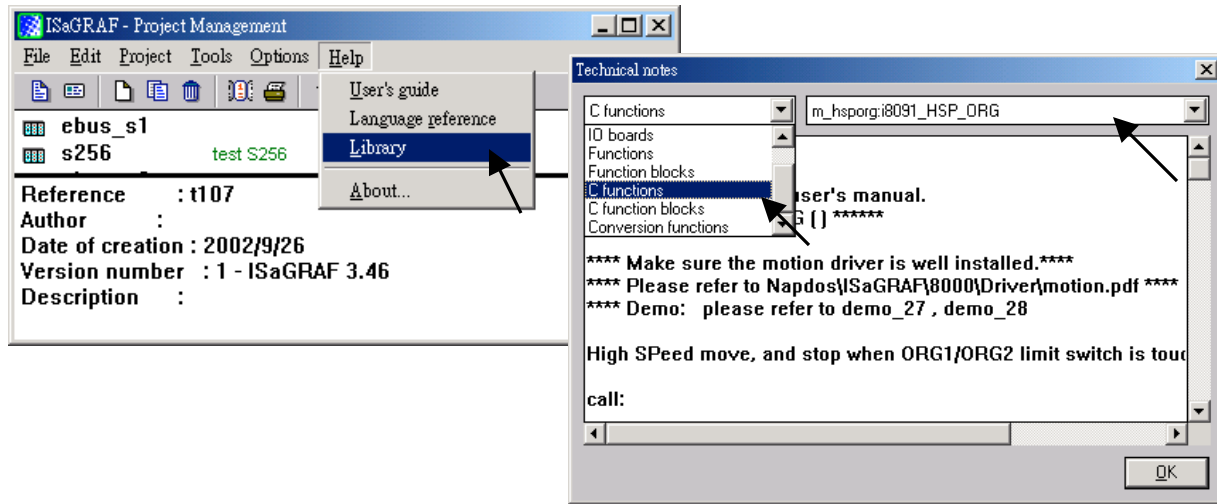
Restriction of the motion driver of I-8417/8817/8437/8837:

The motion driver for I-8417/8817/8437/8837 doesn't support the Ethernet communication, however W-8337/8737 doesn't have this limitation.

The ISaGRAF demo projects of motion for I-8417/8817/8437/8837 are "demo_27", "demo_28", & "demo_46". They are located in the 8000 CD-ROM: napdos\isagraf\8000\demo\.

The ISaGRAF demo projects of motion for W-8337/8737 are "wdemo_26", "wdemo_27", "wdemo_28" & "wdemo_29". They are located in the Wincon CD-ROM:
napdos\isagraf\wincon\demo\.

All functions that trigger I-8091 & I-8090 are named as "M_???", Please refer to the On-line help from the ISaGRAF "Help" – "Library" - "C functions" for names starting with "M_???".



Beside, please refer to "I-8091 & I-8090 User's Manual" .It can be found in the package box of the i-8091, or
CD-ROM: napdos\8000\motion\i8091>manual\

18.2: Introduction

18.2.1: System Block Diagram

The I-8091 stepping motor control card is a micro-computer controlled, 2-axis pulse generation card. It includes a 2Kbytes-FIFO to receive motion command from host, a micro-computer for profile generation and protection, 2-axis DDA chip to execute DDA function when interpolation command is used, 2500Vrms optical isolation inserted for industrial application.

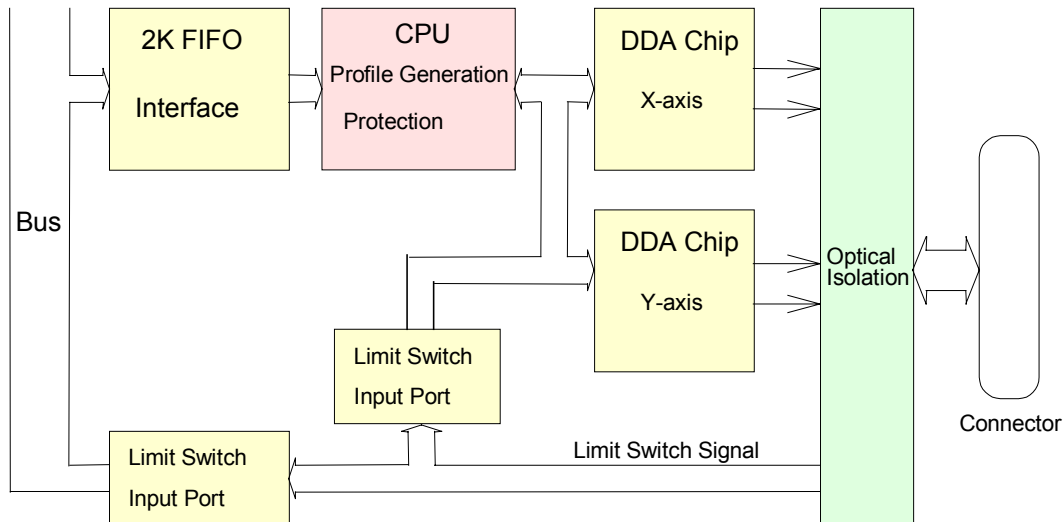


Fig.(1) block diagram of I-8091 card

18.2.2: DDA Technology

The DDA chip is the heart of I-8091 card, it will generate equal-space pulse train corresponding to specific pulse number during a DDA period. This mechanism is very useful to execute pulse generation and interpolation function. The DDA period can be determined by DDA cycle. Table(1) shows the relation among DDA cycle, DDA period and output pulse rate. When DDA cycle set to 1, the DDA period is equal to $(1+1) \times 1.024\text{ms} = 2.048\text{ms}$. The output pulse number can be set to 0~2047, therefore the maximum output pulse rate will be 1Mpps. The minimum output pulse rate is 3.83pps when set DDA cycle=254 (DDA period = $(254+1) \times 1.024\text{ms} = 261.12\text{ms}$).

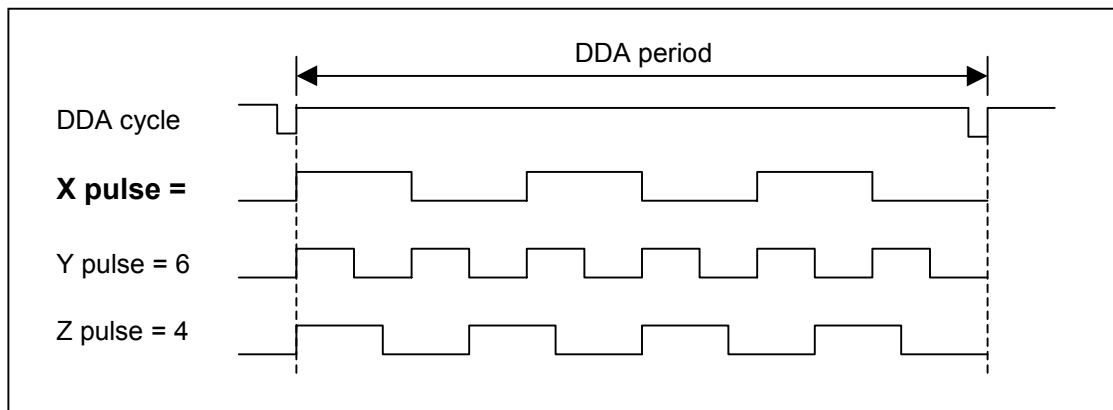
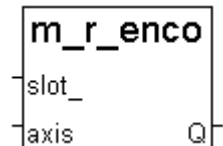


Fig.(2) DDA mechanism

I-8090 encoder commands:

M_r_enco **Reset I-8090's encoder value to 0**



Parameters:

slot_ integer the slot No. where the i8090 is plugged, 0 ~ 7
axis_ integer 1: x-axis, 2: y-axis, 3: z-axis

Return:

Q_ boolean always return TRUE.

Example: demo_27, demo_28, demo_46

The DDA cycle can be calculated by follow equation

$$PR_{max} = \frac{Np}{(DDA_{cycle} + 1) * 1.024ms}$$

$$4166.67 = \frac{250}{(DDA_{cycle} + 1) * 1.024ms}$$

DDA cycle = 58
 High Speed = 247 pulse (4166.67*58*0.001024)

The above results means that maximum speed is 500rpm when send command i8091_SET_VAR(0, 58, 2, 2, 247) to I-8091 card.

Example: Pulse type input Servo Motor

The spec. of servo motor is 8000 pulse/rev, max. speed 3000 rpm, speed resolution 2 rpm.

The required max. pulse rate

$$PR_{max} = 3000 \text{ rpm} * 8000 / 60 = 400,000 \text{ pps}$$

The maximum output pulse

$$Np = 3000 \text{ rpm} / 2 \text{ rpm} = 1500 \text{ pulse number}$$

The DDA cycle can be calculated by follow equation

$$PR_{max} = \frac{Np}{(DDA_{cycle} + 1) * 1.024ms}$$

$$400,000 = \frac{1500}{(DDA_{cycle} + 1) * 1.024ms}$$

DDA cycle = 3
 High Speed = 1638 pulse (400,000*4*0.001024)

The above results means that maximum speed is 3000rpm when send command i8091_SET_VAR(0, 3, 2, 2, 1638) to I-8091 card.

18.3: Hardware

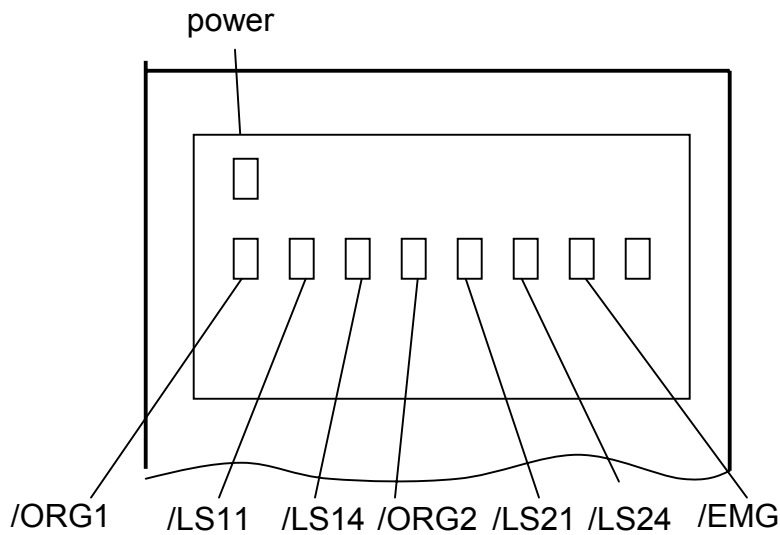
18.3.1: I-8000 hardware address

The hardware address of I-8000 main system is fixed as following table. There are 4 slots I-8000 and 8 slots I-8000.

	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
I-8000, 4 slot address	0x080	0x0A0	0x0C0	0x0E0	---	---	---	---
I-8000, 8 slot address	0x080	0x0A0	0x0C0	0x0E0	0x140	0x160	0x180	0x1A0

Fig.(3) I-8000 hardware address

18.3.2: LED Indicator



/ORG1: X-axis's original limit switch for machine home position.
 /LS11, /LS14 : X-axis's negative and positive limit switches.
 /ORG2: Y-axis's original limit switch for machine home position.
 /LS21, /LS24 : Y-axis's negative and positive limit switches.
 /EMG : system's emergency signal input.

Fig.(4) I-8091 LED indicator

18.3.3: Hardware Configuration

Limit switch configuration

Because the profile generation and protection is executed by the CPU on I-8091 card, the limit switches must configure as following diagram. The motion command just can work properly.

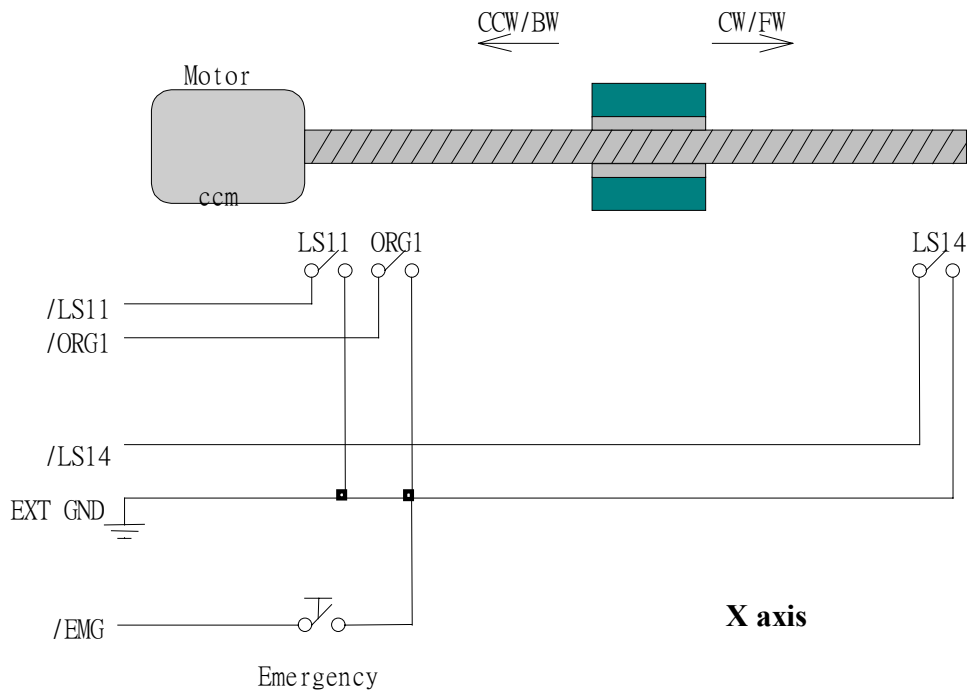


Fig.(5) Limit switch configuration of X axis

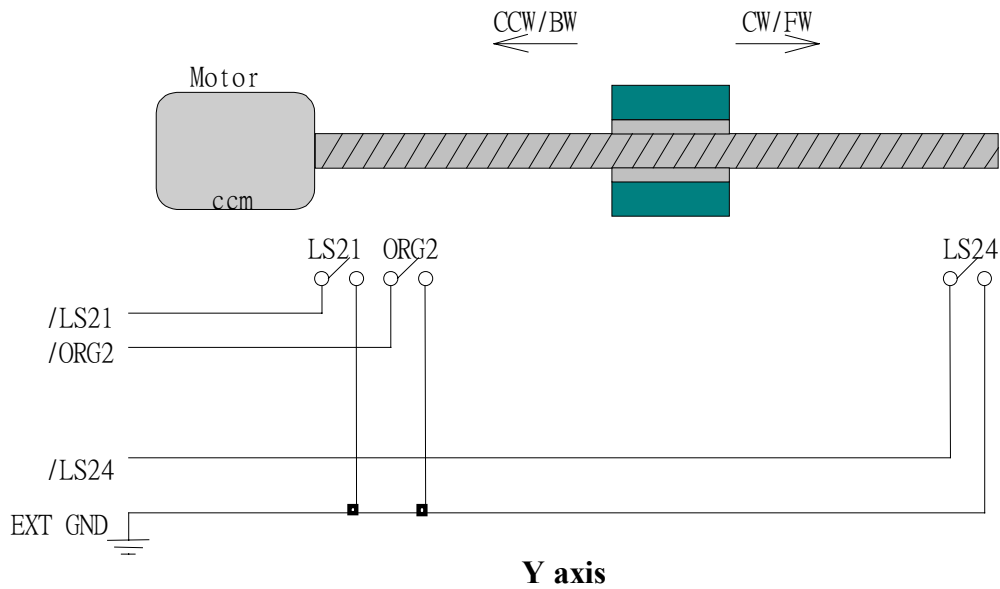


Fig.(6) Limit switch configuration of Y axis

Output pulse mode configuration

I-8091 card provide two kind output method.

- (a) CW/CCW mode
- (b) Pulse/Direction mode

The command **M_s_mode(card_NO_, modeX_, modeY_)** provide parameters 0: CW_CCW and 1: PULSE_DIR to define output pulse mode.

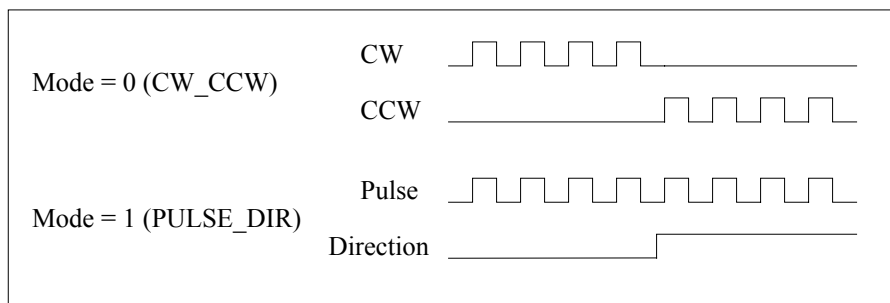


Fig.(7) Output pulse mode

Direction configuration

Sometimes, the output direction of X-axis, Y-axis is not in the desired direction due to the motor's connection or gear train. It is recommended to unify the output direction as shown in Figure(5)(6). The CW/FW direction is defined as toward outside from motor and the CCW/BW direction is defined as toward inside to motor. The **M_s_dir(card_NO_, defdirX_, defdirY_)** command provides parameters 0: NORMAL_DIR and 1: REVERSE_DIR to define the rotating direction of motor.

Turn Servo ON/OFF (Hold ON/OFF)

To turn servo motor into servo ON(OFF) state, or turn stepping motor into hold ON(OFF) state, the command **M_s_serv(card_NO_, sonX_, sonY_)** provide parameters 1:ON and 0:OFF to turn ON or OFF.

Automatic protection

The I-8091 card has a automatic protected system.

- (a) If X-axis command is executing and moving toward CW/FW direction, X-axis will immediately stop when LS14 is touched. To release this protection as long as X-axis move toward CCW/BW direction.
- (b) If X-axis command is executing and moving toward CCW/BW direction, X-axis will immediately stop when LS11 is touched. To release this protection as long as X-axis move toward CW/FW direction.
- (c) If Y-axis command is executing and moving toward CW/FW direction, Y-axis will immediately stop when LS24 is touched. To release this protection as long as Y-axis move toward CCW/BW direction.
- (d) If Y-axis command is executing and moving toward CCW/BW direction, Y-axis will immediately stop when LS21 is touched. To release this protection, as long as Y-axis move toward CW/FW direction.
- (e) If the signal of the emergency limit switch /EMG was found in CPU firmware, all motion will be terminated and stop.

Set limit switch as normal close condition

The limit switches /EMG, /LS11, /LS14, /LS21, /LS24, /ORG1, /ORG2 is initially normal open condition, that is, these signal is active when connect it to ground. In industrial application, it might be recommended normal close condition, that is, these signal is active when open from ground.

The **M_s_nc(card_NO_, sw_)** command can be set sw=0 (default), for normal open condition. When set sw=1, for normal close condition.

18.3.4: Pin assignment of connector CN2

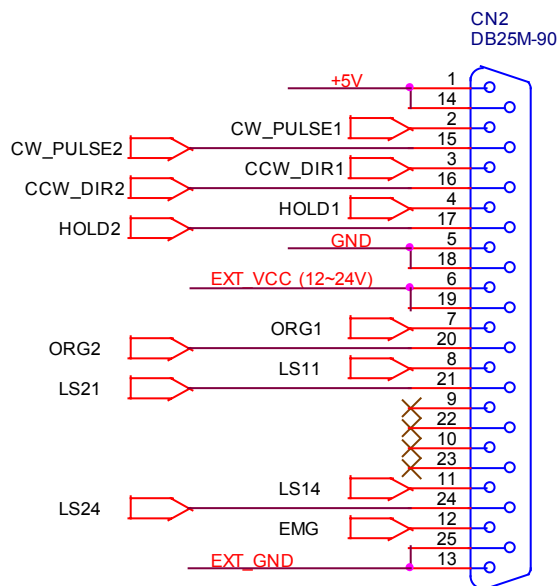


Fig.(8) CN2 connector of I-8091

Table of CN2 connector's pin assignment

pin name	pin number	Description
+5V	1	Internal +5V power, Max. output current: 50mA
CW_PULSE1	2	X-axis CW (Pulse) output pin
CCW_DIR1	3	X-axis CCW (Direction) output pin
HOLD1	4	X-axis HOLD (servo on) output pin
GND	5	Signal ground of pin 2,3,4
EXT_VCC	6	External power(12~24V) for limit switches
/ORG1	7	X-axis original (home) limit switch
/LS11	8	X-axis limit switch
	9,10	No used
/LS14	11	X-axis limit switch
/EMG	12	Emergency input
EXT_GND	13	External ground for limit switch
+5V	14	Internal +5V power, Max. output current: 50mA
CW_PULSE2	15	Y-axis CW (Pulse) output pin
CCW_DIR2	16	Y-axis CCW (Direction) output pin
HOLD2	17	Y-axis HOLD (servo on) output pin
GND	18	Signal ground of pin 15,16,17
EXT_VCC	19	External power(12~24V) for limit switches
/ORG2	20	Y-axis original (home) limit switch
/LS21	21	Y-axis limit switch
	22,23	No used
/LS24	24	Y-axis limit switch
EXT_GND	25	External ground for limit switch

The internal circuit of CW_PULSE, CCW_DIR, HOLD

When output these signal as 1, it can source 15mA(max.).
When output these signal as 0, it can sink 50mA(max.)

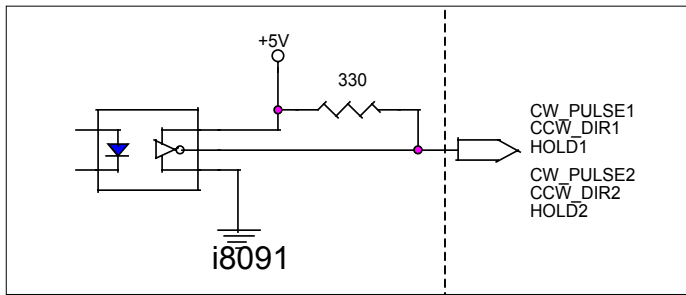


Fig.(9) internal circuit of pulse output pin

The internal circuit of limit switch input

Initially, the limit switch inputs of I-8091 board are normal open (N.O.), the I-8091 board will automatic protect when limit switch pin connect to EXT_GND. The user can use the command **M_s_nc(card_NO, 1)** to let those limit switch input as normal close condition at the beginning of the user's program.

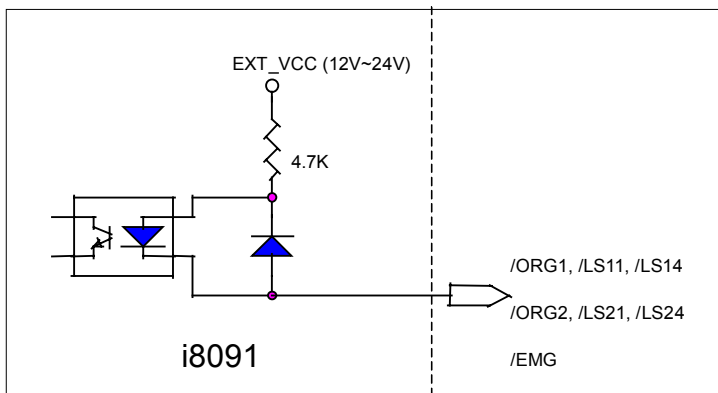


Fig.(10) internal circuit of limit switch input pin

Example of connection

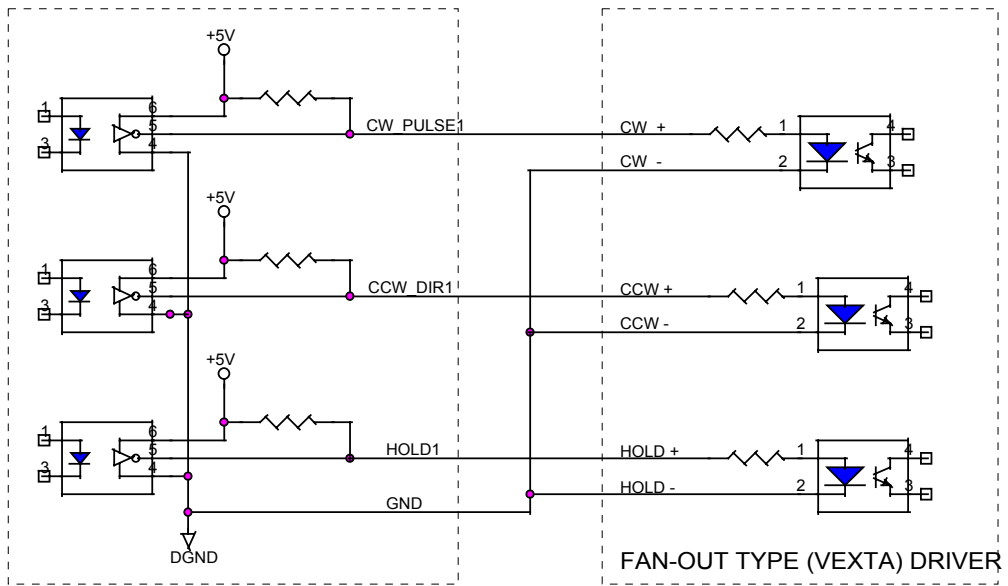


Fig.(11) fan-out type driver (VEXTA's motor driver)

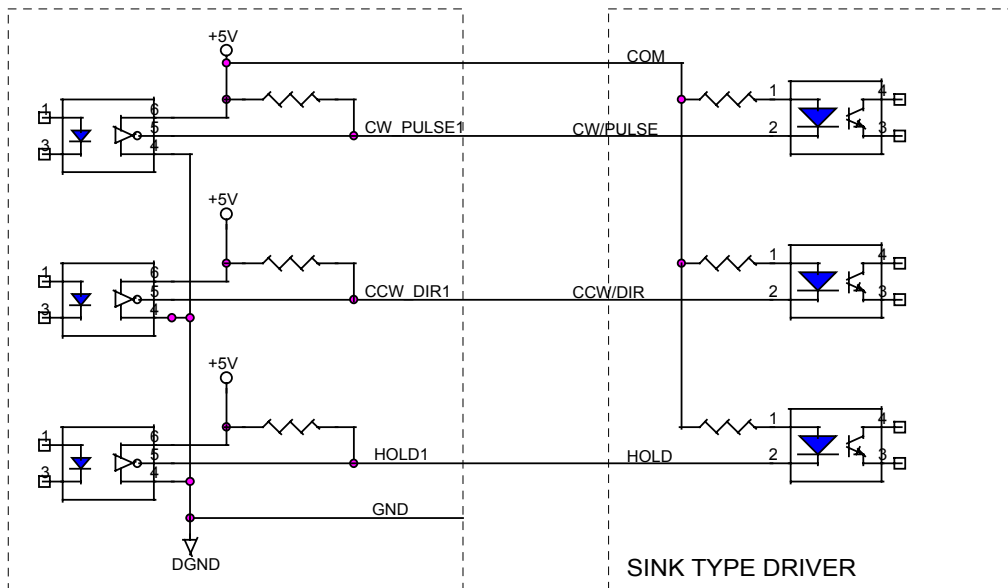


Fig.(12) Sink type driver

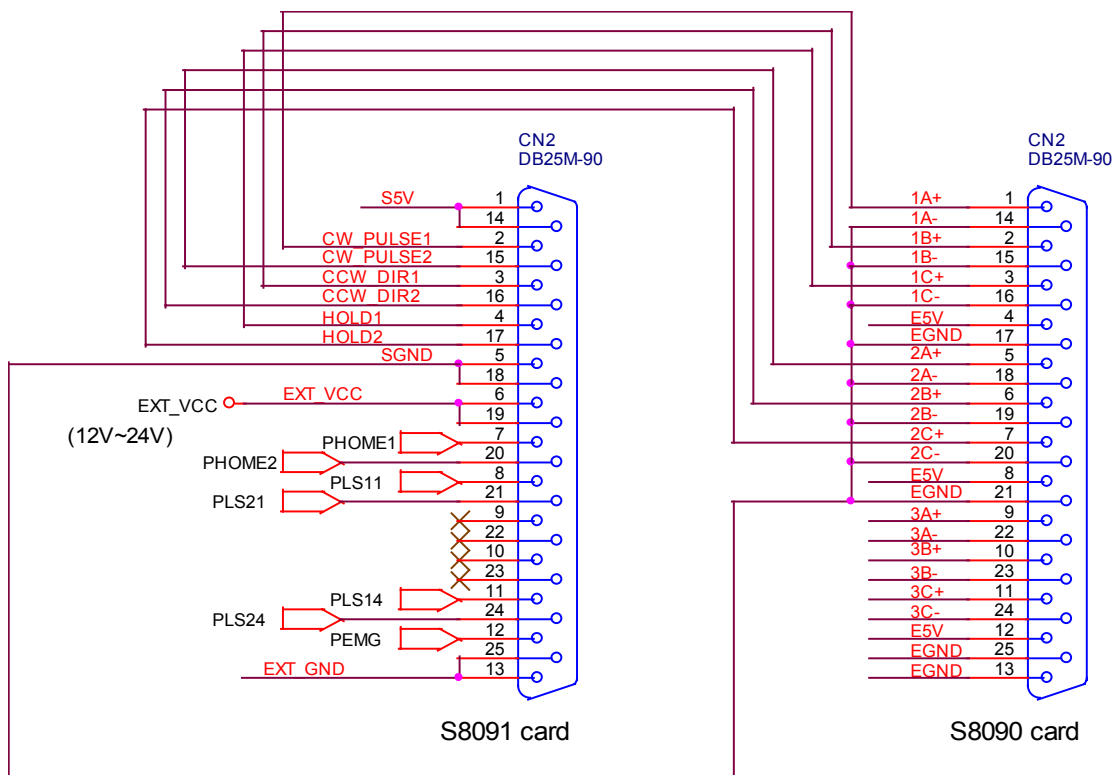
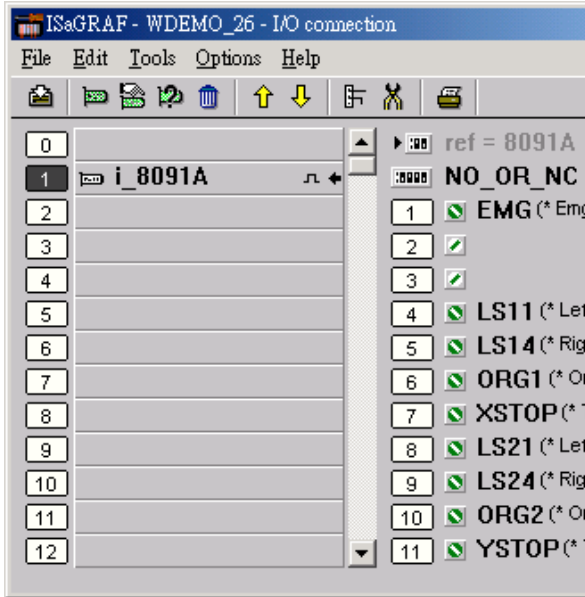


Fig.(13) The connection between I-8090 and I-8091 for function testing or pulse feedback by I-8090 encoder card.

18.4: Software

I/O connection:

The "I-8091A" connected on the I/O connection window contains 11 digital input channels.

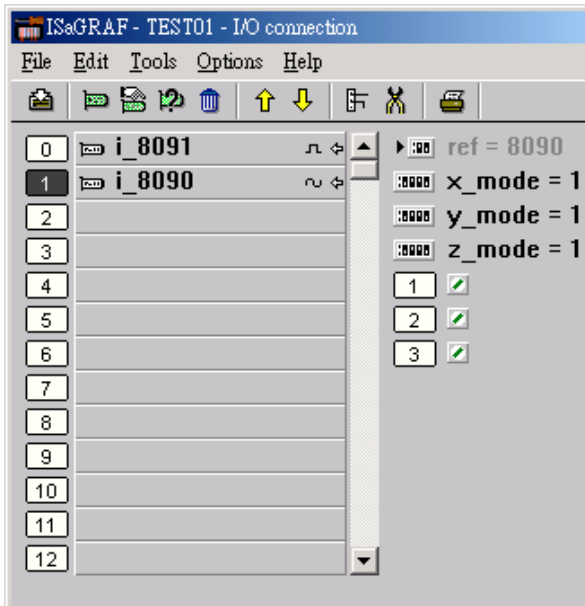


The screenshot shows the 'I/O connection' window for 'I-8091A'. It lists 12 channels, with channels 1 through 11 being active digital inputs. Channel 1 is 'EMG (* Eng)'. Channels 2 through 11 are limit and stop switches for X and Y axes.

Input Channel:

- CH1 : EMG, emergency stop
- CH2 : /FFEF, FIFO is empty or not, TRUE: empty
- CH3 : /FFFF, FIFO is full or not, TRUE: full
- CH4 : LS11, Left limit switch of X-axis
- CH5 : LS14, Right limit switch of X-axis
- CH6 : ORG1, Original position switch of X-axis
- CH7 : XSTOP, Stop or not of X-axis, TRUE: stop
- CH8 : LS21, Left limit switch of Y-axis
- CH9 : LS24, Right limit switch of Y-axis
- CH10 : ORG2, Original position switch of Y-axis
- CH11 : YSTOP, Stop or not of Y-axis, TRUE: stop

I-8090 contains 3 analog input channels.



The screenshot shows the 'I/O connection' window for 'I-8090'. It lists 12 channels, with channels 1 through 3 being active analog inputs. Channel 1 is 'x_mode', channel 2 is 'y_mode', and channel 3 is 'z_mode'. All are set to a value of 10.

Parameter:

- x_mode : integer counting mode of X-axis
- y_mode : integer counting mode of Y-axis
- z_mode : integer counting mode of Z-axis
- 00: quadrant counting mode
- 10: CW/CCW counting mode
- 20: pulse/direction counting mode

Input Channel:

- CH1 : encorder value of X-axis
- CH2 : encorder value of Y-axis
- CH3 : encorder value of Z-axis

CH1 to CH3 are signed 32-bit integer format

Setting commands:

M_regist Register one I-8091



In order to distinguish more than one I-8091 card in I-8417/8817/8437/8837 platform, the I-8091 cards should be registered before using it. This command will assign a card number = "card_NO_" to I-8091 card at that "address_". If there is no I-8091 at the given address, this command will return FALSE.

Note: If using "I_8091A" rather than "I_8091" on the I/O connection window, user don't need to call "m_regist" & "m_s_nc", they are ignored. The card_NO of "I-8091A" is equal to its slot No. I-8xx7: 0 ~ 7. W-8xx7: 1 ~ 7.

Parameters:

card_NO_	integer	valid is 0 ~ 19.
address_	integer	the plugged slot address of the i8091 card

slot 0: 16#80
slot 1: 16#A0
slot 2: 16#C0
slot 3: 16#E0
slot 4: 16#140
slot 5: 16#160
slot 6: 16#180
slot 7: 16#1A0

Return:

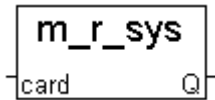
Q_ boolean TRUE: Ok , FALSE: Fail

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

```
(* declaration: INIT as boolean <internal> and has initial value of TRUE *)
(* TMP as boolean <internal> *)
(* cardNO as integer <internal> and has initial value of 1 *)
(* Do some init setting at 1st scan cycle *)
if INIT then
    INIT := FALSE;
    TMP := M_regist(cardNO,16#80); (* plug i8091 in slot 0 *)
    TMP := M_r_sys(cardNO); (* reset i8091's setting *)
    TMP := M_s_var(cardNO,4,2,5,100);
    TMP := M_s_dir(cardNO,0,0); (* Normal direction *)
    TMP := M_s_mode(cardNO,1,1); (* pulse_dir mode *)
    TMP := M_s_serv(cardNO,1,1); (* X & Y server ON *)
    TMP := M_s_nc(cardNO,0); (* Normal open *)
end_if;
```

M_r_sys Reset all setting

To reset I-8091 card, this command will terminate the running command in I-8091 card. User can use this command as software emergency stop. This command also will clear all of setting, so, all I-8091 card's parameter should be set again.



Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19

Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

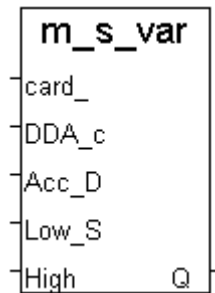
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_s_var Set motion system parameters

To set DDA cycle, accelerating/decelerating speed, low speed and high speed value.

Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
DDA_cycle_	integer	DDA cycle , valid is 1 ~ 254
Acc_Dec_	integer	Acc/Dec speed , valid is 1 ~ 200
Low_Speed_	integer	low speed , valid is 1 ~ 200 , Low_Speed_ >= Acc_Dec_
High_Speed_	integer	high speed , Low_Speed_ <= High_Speed_ <= 2047

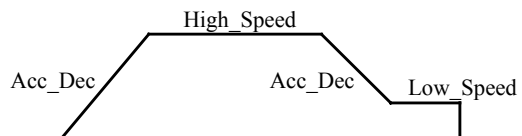


Return:

Q_ boolean always return TRUE.

Note:

The lower "DDA_cycle_" is given, the smaller delay time between /ORG1 ON and /X_STOP ON (or /ORG2 ON and /Y_STOP ON) when using M_hsporg & M_lsporg command. For ex, DDA_cycle_ set to 4, the delay time is about 5 to 13 ms.



Restriction:

$1 \leq DDA_cycle \leq 254$
 $1 \leq Acc_Dec \leq 200$
 $1 \leq Low_Speed \leq 200$
 $Low_Speed \leq High_Speed \leq 2047$
 $Low_Speed \geq Acc_Dec$

Default value

DDA_cycle = 10
Acc_Dec = 1
Low_Speed = 10
High_Speed = 100

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

TMP := M_s_var(1, 5, 2, 10, 150);

(* DDA_cycle = 5 --> DDA period = (5+1)*1.024ms = 6.144ms

Acc_Dec = 2 --> Acc/Dec speed = 2/(6.144ms)^2 = 52981 p/s^2

Low_Speed = 10 --> low speed = 10/6.144ms = 1628pps

High_Speed = 150 --> high speed = 150/6.144ms = 24414pps *)

M_s_dir Define output direction of axes

Sometimes, the output direction of X-axis, Y-axis is undesired direction due to the motor's connection or gear train. In order to unify the output direction as shown in Fig.(5) and Fig.(6). Where CW/FW direction is defined as toward outside from motor, CCW/BW direction is defined as toward inside from motor. This command provide parameters to define the rotating direction of motor.



Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
defdirX_	integer	X axis direction definition , valid is 0 ~ 1
defdirY_	integer	Y axis direction definition , valid is 0 ~ 1

0: normal direction, 1: reverse direction

Return:

Q_ boolean always return TRUE.

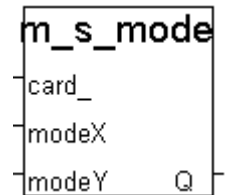
Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
 W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_s_mode Set output mode

Parameters:

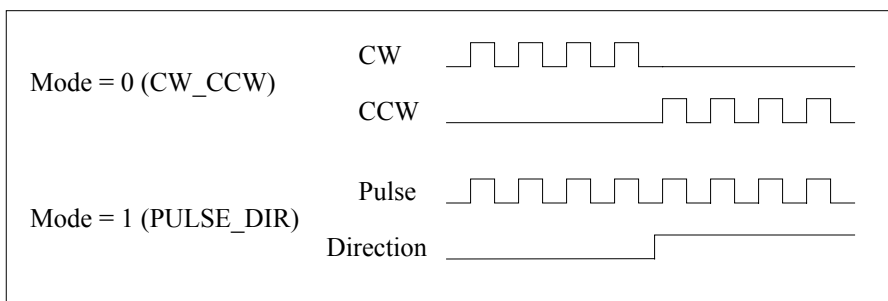
card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
modeX_	integer	X axis mode, valid is 0 ~ 1
modeY_	integer	Y axis mode, valid is 0 ~ 1

0: CW_CCW, 1: PULSE_DIR



Return:

Q_ boolean always return TRUE.

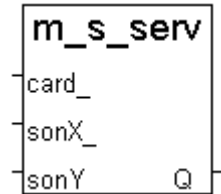


Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
 W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_s_serv Set servo ON/OFF

Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
sonX_integer X axis servo/hold on switch , valid is 0 ~ 1
sonY_integer Y axis servo/hold on switch , valid is 0 ~ 1
0: OFF, 1: ON



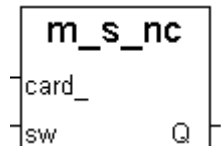
Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_s_nc Set N.O. / N.C.

To set all of the following limit switches as N.C.(normal close) or N.O.(normal open). If set as N.O., those limit switches are active low. If set as N.C., those limit switches are active high. The auto-protection will automatically change the judgement whatever it is N.O. or N.C..



Limit switches: ORG1, LS11, LS14, ORG2, LS21, LS24, EMG.

Note: If using "I_8091A" rather than "I_8091" on the I/O connection window, user don't need to call "m_regist" & "m_s_nc", they are ignored. The card_NO of "I-8091A" is equal to its slot No. I-8xx7: 0 ~ 7. W-8xx7: 1 ~ 7.

Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
sw_ integer 0: N.O. (default) , 1: N.C.

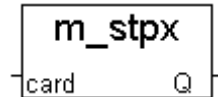
Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

Stop commands:

M_stpx **Stop X axis**



Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19

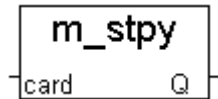
Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_stpy **Stop Y axis**



Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19

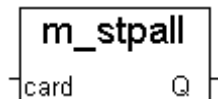
Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_stpall **Stop X & Y axes**



This command will stop X & Y axes and clear all of commands pending in the FIFO.

Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19

Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

Simple motion commands:

M_lsporg Low speed move to ORG

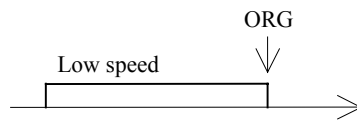
Low speed move , and stop when **ORG1/ORG2** limit switch is touched.

Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
DIR_ integer 0: CW , 1: CCW
AXIS_ integer 1: X axis , 2: Y axis

Return:

Q_ boolean always return TRUE.



M_hsporg High speed move to ORG

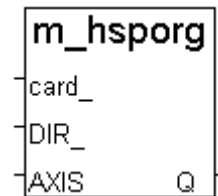
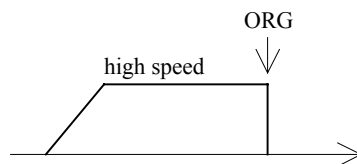
High speed move , and stop when **ORG1/ORG2** limit switch is touched.

Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
DIR_ integer 0: CW , 1: CCW
AXIS_ integer 1: X axis , 2: Y axis

Return:

Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

Note:

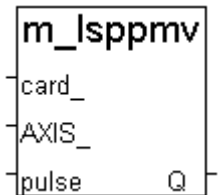
The lower "DDA_cycle_" is given, the smaller delay time between /ORG1 ON and /X_STOP ON (or /ORG2 ON and /Y_STOP ON) when using M_hsporg & M_lsporg command. For ex, DDA_cycle_ set to 4, the delay time is about 5 to 13 ms.

M_lsppmv Low speed pulse move

Low speed move a specified "pulse"

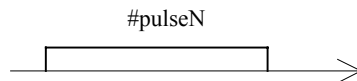
Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
AXIS_ integer 1: X axis , 2: Y axis
Pulse_ integer number of pulse to move. if > 0, move toward CW/FW dir.
if < 0, move toward CCW/BW dir.



Return:

Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_hsppmv High speed pulse move

High speed move a specified "pulse"

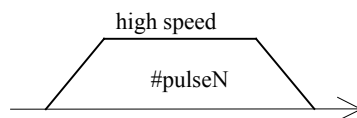
Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
AXIS_ integer 1: X axis , 2: Y axis
Pulse_ integer number of pulse to move. if > 0, move toward CW/FW dir.
if < 0, move toward CCW/BW dir.



Return:

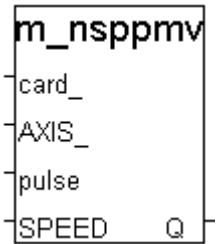
Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_nsppmv Normal speed pulse move

Normal speed move a specified "pulse"

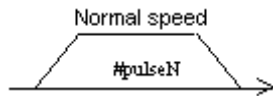


Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
AXIS_ integer 1: X axis , 2: Y axis
Pulse_ integer number of pulse to move. if > 0, move toward CW/FW dir.
if < 0, move toward CCW/BW dir.
SPEED_ integer Speed, low speed <= SPEED_ <= high speed

Return:

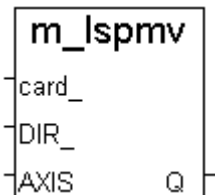
Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_lspmv Low speed move

Low speed move toward the direction specified. It can be stop by **M_stpx** or **M_stpy** or **M_stpall** command

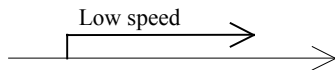


Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
DIR_ integer direction. 0: CW , 1: CCW
AXIS_ integer 1: X axis , 2: Y axis

Return:

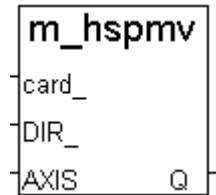
Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_hspmv High speed move

High speed move toward the direction specified. It can be stop by **M_stpx** or **M_stpy** or **M_stpall** command

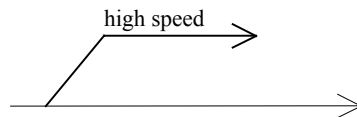


Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
DIR_ integer direction. 0: CW , 1: CCW
AXIS_ integer 1: X axis , 2: Y axis

Return:

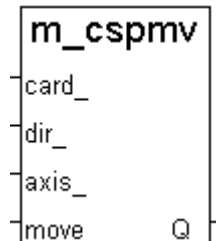
Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_cspmv Change speed move

This command will accelerate/decelerate the selected axis's motor to the "move_speed". This command can be continuously send to I-8091 to dynamically change speed. The rotating motor can be stop by the command **M_stpx**, **M_stpy**, **M_stpall**, or **M_slwstp**

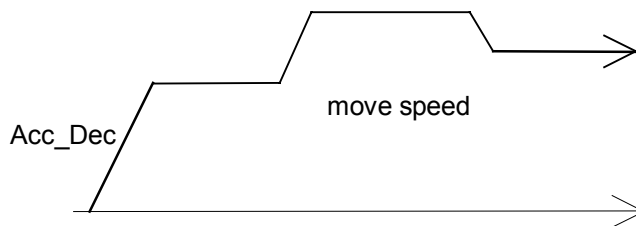


Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
dir_ integer direction. 0: CW , 1: CCW
axis_ integer 1: X axis , 2: Y axis
move_speed_ integer 0 < move_speed_ <= 2040

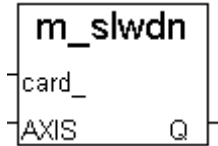
Return:

Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_slwdn Slow down to low speed



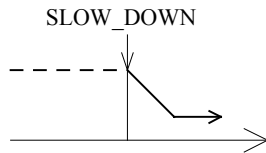
To decelerate to slow speed until **M_stpx** or **M_stpy** or **M_stpall** is executed.

Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
AXIS_integer 1: X axis , 2: Y axis

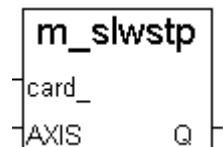
Return:

Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_slwstp Slow down to stop



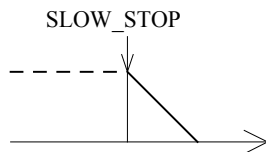
To decelerate to stop.

Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
AXIS_integer 1: X axis , 2: Y axis

Return:

Q_ boolean always return TRUE.

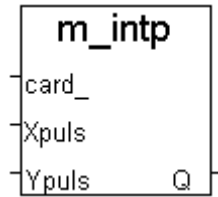


Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

Interpolation commands:

M_intp Move a short distance on X-Y plane

This command will move a short distance (interpolation short line) on X-Y plane. This command provided a method for user to generate an arbitrary curve on X-Y plane.

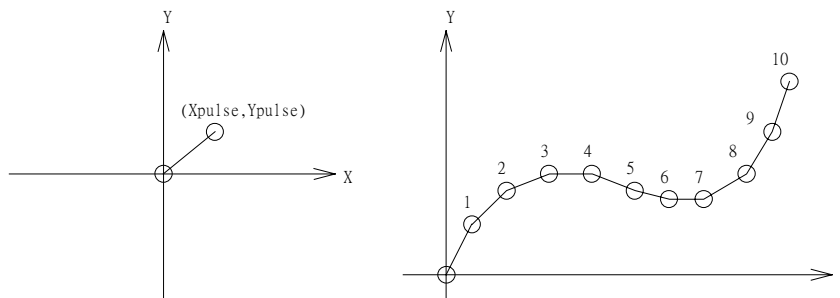


Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
Xpulse_	integer	-2047 <= Xpulse_ <= 2047
Ypulse_	integer	-2047 <= Ypulse_ <= 2047

Return:

Q_	boolean	always return TRUE.
----	---------	---------------------



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

NOTE:

For a lot of **M_intp** call set at the same time, please check if the FIFO is not full. Call it if FIFO is not full. FIFO indicator is a Digital Input resides at CH3 of i-8091.

i-8091 D/I channel on ISaGRAF I/O connection window:

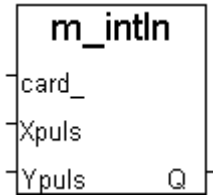
- CH1 : EMG, emergency stop
- CH2 : /FFEF, FIFO is empty or not, TRUE: empty
- CH3 : /FFFF, FIFO is full or not, TRUE: full**

- CH4 : LS11, Left limit swtch of X-axis
- CH5 : LS14, Right limit swtch of X-axis
- CH6 : ORG1, Original position swtch of X-axis
- CH7 : XSTOP, Stop or not of X-axis, TRUE: stop

- CH8 : LS21, Left limit swtch of Y-axis
- CH9 : LS24, Right limit swtch of Y-axis
- CH10 : ORG2, Original position swtch of Y-axis
- CH11 : YSTOP, Stop or not of Y-axis, TRUE: stop

M_intln Move a long distance on X-Y plane

This command will move a long distance (interpolation line) on X-Y plane. The CPU on I-8091 card will generate a trapezoidal speed profile of X-axis and Y-axis, and execute interpolation by way of DDA chip.

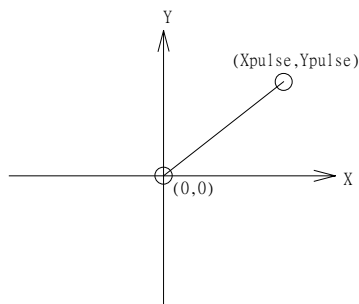


Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
Xpulse_	integer	-524287 <= Xpulse_ <= 524287
Ypulse_	integer	-524287 <= Xpulse_ <= 524287

Return:

Q_	boolean	always return TRUE.
----	---------	---------------------



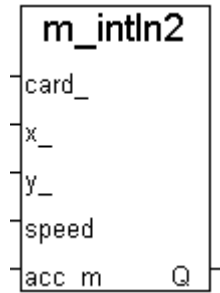
Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_intln2 Move a long distance on X-Y plane

This command will move a long interpolation line on X-Y plane. It will automatically generate a trapezoidal speed profile of X-axis and Y-axis by state-machine-type calculation method.

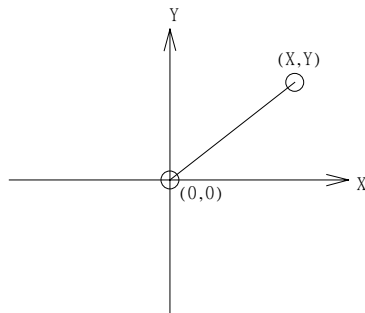
Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
x_, y_	integer	end point relate to present position
speed_	integer	0 ~ 2040
acc_mode_	integer	0: enable acceleration/deceleration profile 1: disable acceleration/deceleration profile



Return:

Q_	boolean	always return TRUE.
----	---------	---------------------



NOTE:

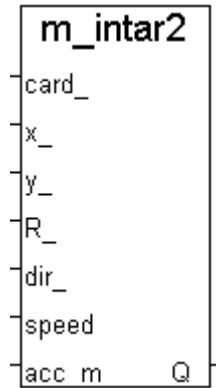
1. Only one of **M_intln2**, **M_intcl2** & **M_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M_intstp** to test command of **M_intln2**, **M_intcl2** & **M_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M_intln2** , **M_intcl2** , **M_intar2** command. Two or more I-8091 cards in the same controller to use **M_intln2** , **M_intcl2** , **M_intar2** at the same time is not possible.

M_intar2 Move a arc on X-Y plane

This command will generate an interpolation arc on X-Y plane. It will automatically generate a trapezoidal speed profile of X-axis and Y-axis by state-machine-type calculation method.

Parameters:

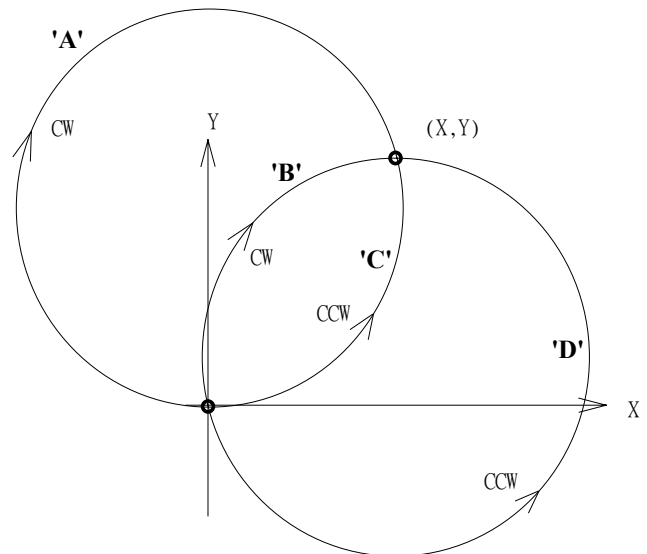
card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
 x_, y_ integer end point of arc relate to present position
 R_ integer radius of arc, if > 0, the arc < 180 degree, if < 0, the arc > 180 degree
 R_ must > (square root of (X_*X_+Y_*Y_)) / 2
 dir_ integer moving direction. 0: CW , 1: CCW
 speed_ integer 0 ~ 2040
 acc_mode_ integer 0: enable acceleration/deceleration profile
 1: disable acceleration/deceleration profile



Return:

Q_ boolean always return TRUE.

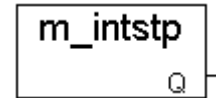
R	dir	path of curve
R>0	CW	'B'
R>0	CCW	'C'
R<0	CW	'A'
R<0	CCW	'D'



NOTE:

1. Only one of **M_intln2**, **M_intcl2** & **M_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M_intstp** to test command of **M_intln2**, **M_intcl2** & **M_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M_intln2** , **M_intcl2** , **M_intar2** command. Two or more I-8091 cards in the same controller to use **M_intln2** , **M_intcl2** , **M_intar2** at the same time is not possible.

M_intstp Test X-Y plane moving command



To test the below 3 commands completed or not.

M_intln2 , M_intcL2 , M_intar2

It will return FALSE for interpolation command completed while return TRUE for busy - not completed yet.

Return:

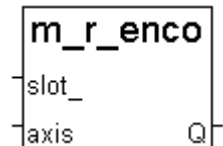
Q_ boolean TRUE: busy , FALSE: completed

NOTE:

1. Only one of **M_intln2**, **M_intcL2** & **M_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M_intstp** to test command of **M_intln2**, **M_intcL2** & **M_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M_intln2** , **M_intcL2** , **M_intar2** command. Two or more I-8091 cards in the same controller to use **M_intln2** , **M_intcL2** , **M_intar2** at the same time is not possible.

I-8090 encoder commands:

M_r_enco **Reset I-8090's encoder value to 0**



Parameters:

slot_ integer the slot No. where the i8090 is plugged, 0 ~ 7
axis_ integer 1: x-axis, 2: y-axis, 3: z-axis

Return:

Q_ boolean always return TRUE.

Example: demo_27, demo_28, demo_46