EPCIO-4000/4005 Hardware User Manual

Version: 3.1.4

Date: 2009.04

http://www.epcio.com.tw

Table of Content

Chapter 1 Overview

1.1 Introduction

The Mechanical and Systems Research Laboratories (MSL) in Industrial Technology Research Institute (ITRI) has had the lead in experience of the industrial automation for many years. The EPCIO–4000/4005 PCI series four-axis motion control card is designed for industrial precision motor control. The EPCIO–4000/4005 uses the EPCIO ASIC, which is developed by MSL of ITRI.

The EPCIO ASIC uses four synchronous DDA pulse generators to send out pulses evenly, in order to achieve synchronous/asynchronous four-axis motion control (Note 1). The EPCIO ASIC can be used with the pulse-command servo motor or stepper motor control. Specifically, various high-level functions such as line, circle, arc etc. are supported in the auxiliary motion control library. Users can also read motor encoder values through the encoder input interface.

The EPCIO–4000/4005 uses closed loop control in the hardware, and adopts a P control algorithm that generates voltage output signals between –10V and 10V to drive velocity-command servo motors. The EPCIO–4000/4005 can be applied in multi-axis precision servo control (Note 2).

Each axis has three input connection and one output connection. They are home position, positive travel limit, negative travel limit and servo-on output; and a position ready output and an emergency stop input are available for the board.

A maximum 128 input connections and 128 output connections module can be connected to the board by a wire-saving transmission line to increase I/O connections **(EPCIO-4005 has 64 input connections and 64 output connnections).** In addition, the EPCIO-4000 can optionally add a 6-channel A/D converter **(EPCIO-4005 does not support)**.

Note 1: DDA please refer to Fig. 1-3 and Fig. 1-4.

Note 2: P algorithm please refers to Fig. 1-8.

1.2 Features

- 32 bit PCI interface
- 4 closed loop motion control or open loop motion control
- 4 D/A converters (16 bit) **(EPCIO-4005 does not support)**
- 5 encoder inputs (32 bit)
- 13 digital input points and 5 digital output points
- 6 A/D converters (12 bit) **(Optional for EPCIO-4000, EPCIO-4005 does not support)**
- 256 remote digital I/O communication interface **(EPCIO-4005 has 128 input/output connections)**
- Built-in 24-bit timer
- Built-in 16-bit watchdog timer
- Summary

	Open Loop Control	Closed Loop Control	ENC	DAC	ADC	LIO	RIO	Timer	Watchdog Timer
EPCIO-4000	4	4	5	4	6(Option)	13 In 5 Out	128 In 128 Out	Yes	Yes
EPCIO-4005	4	X	5	X	X	13 In 5Out	64 In 64 Out	Yes	Yes

Note:

ENC: Encoder inputs

DAC: Digital to analog converter

ADC: Analog to digital converter

LIO: Local digital input output

RIO: Remote digital I/O

1.3 Software Support

- EPCIO series Device Driver Library (EDDL)

  There are about 150 commands that can be used by the users. For detailed information, please refer to the “EPCIO series Device Driver Library User Manual” and “EPCIO series Device Driver Library Reference Manual”.

- Motion Control Command Library (MCCL)

  The precision motion control commands are provided in the MCCL library. The commands included are: 2D and 3D point-to-point, line, curve, circle, helix and other useful control functions. There are about 250 commands that can be used. For detailed information, please refer to the “EPCIO series Motion Control Command Library User Manual” and “EPCIO series Motion Control Command Library Reference Manual”.

1.4 System Connection Illustration

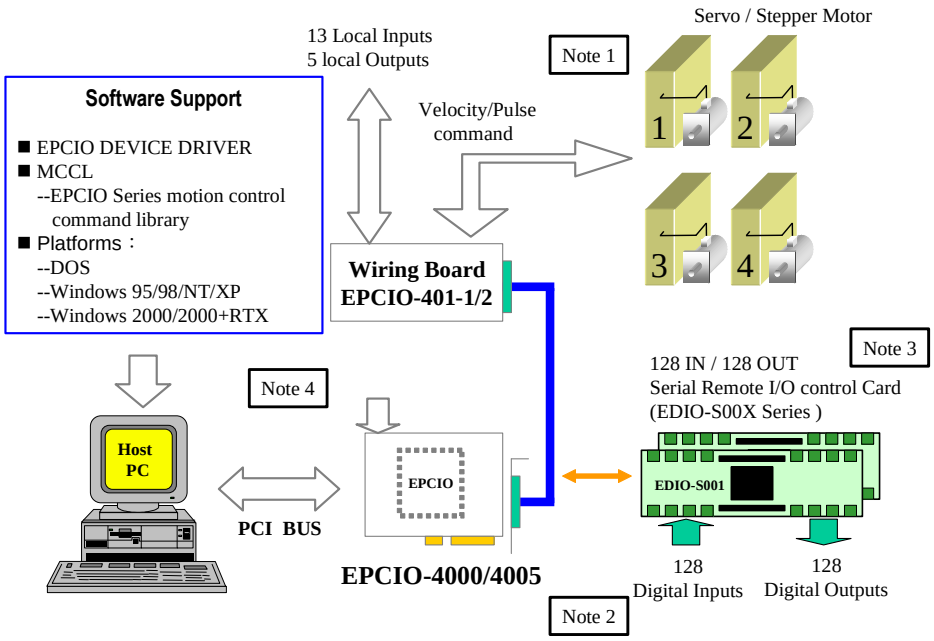


Fig. 1-1

Note 1: The EPCIO-4005 only supports pulse output command.

Note 2: The ADC for the EPCIO-4000 is optional, but the EPCIO-4005 does not support.

Note 3: The EPCIO-4005 only can connect one EDIO card. (64 IN / 64 OUT) the EDIO-X whereas X: S001, S002, S003.

Note 4: Current available wiring boards:
EPCIO-400-1➔Universal wiring board.
EPCIO-400-2➔PANASONIC MINAS series servo motor wiring board.

1.5 System Block Diagram

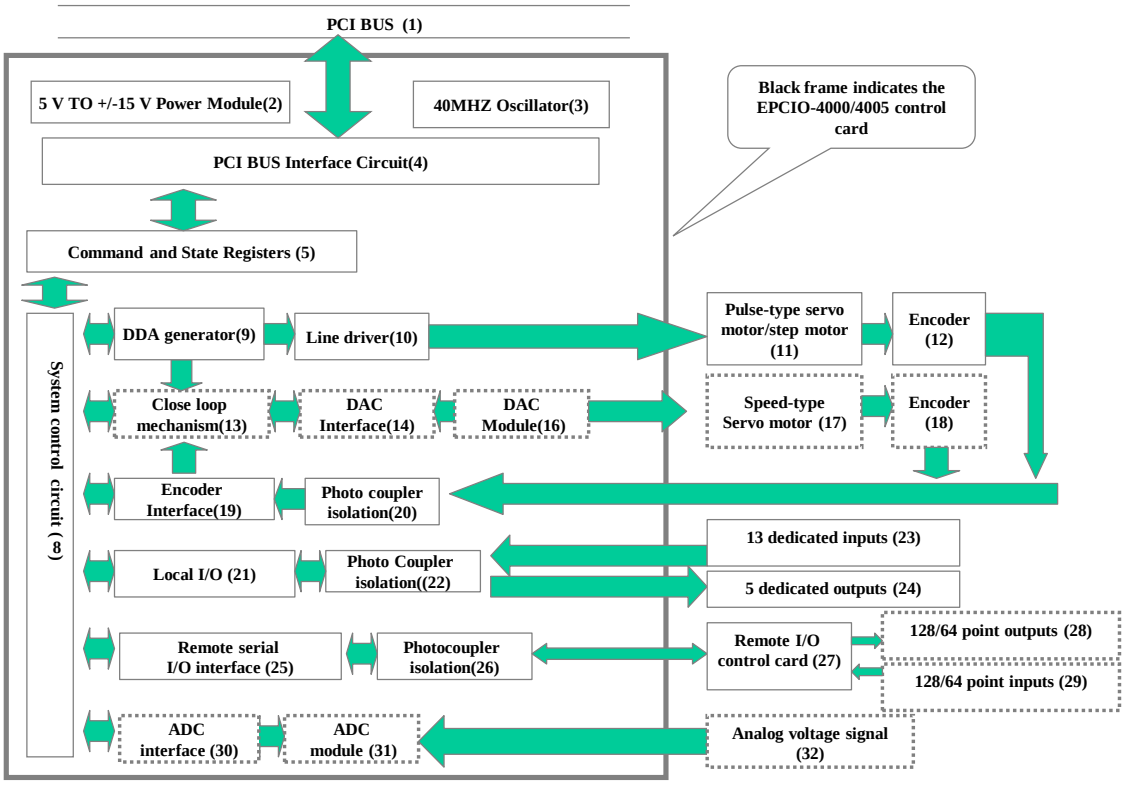


Fig. 1-2

Notes:

- **There are no closed loops modules and no circuit of the ADC in the hardware, so blocks (13), (14), (16), (17), (18), (30), (31), and (32) are not supported on the EPCIO-4005. In addition, the ADC for the EPCIO-4000 is optional.**
- **Blocks (28) and (29) on the EPCIO-4005 only have 64 outputs and 64 inputs.**

1.5.1 Four-axis Synchronous/Asynchronous Open Loop Control (Pulse Command)

In Fig. 1-2, the EPCIO device driver (or MCCL) sends commands to the EPCIO-4000/4005 card via PCI BUS (1). The command and state register (5) and system control circuit (8) will decode commands and enable the DDA generator (9) to send out pulses smoothly (A/B Phase, CW/CCW, or Pulse/Direction format). Then the pulses are differentially amplified by line driver (10) and sent to the pulse-command servo motor or stepper motor (11). The encoder interface (19) with the photo coupler isolator (20) is for receiving the motor encoder (12) signals.

Note 1: Usage Note:

1. **In EPCIO ASIC, there are four motion control cores. Each core can be selected for one of the two following control modes.**

   ---- Closed loop control (velocity-command output)

   ---- Open loop control (pulse command output)

   ➔User can select M sets of closed loop controls and N sets of open loop controls, and *M+N must satisfy M+N≦4* (EPCIO-4005 only supports the open loop control mode).

2. **The corresponding hardware resources of a control core on EPCIO-4000:**

   ---- 1 Control core [DDA generator (9) +closed loop module (13)]

   ---- 1 Line driver (10)

   ---- 1 DAC [DAC interface (14) + DAC Module (16)]

   ---- 1 Encoder interface [encoder interface (19) + photo coupler isolation (20)]

3. **If a control core is configured as an open loop control, the DDA generator (9) inside control core and a line driver (10) are needed. The remaining DAC and encoder interface of that core is then spared for CPU use alone. If that control core is configured as closed loop control, a DDA generator (9), a closed loop module (13), a DAC and an encoder interface are needed. Note that the line driver (10) corresponding to that core will be abandoned and cannot be used by the CPU in any way.**

4. **The total hardware resources about control cores are the following.**

   ---- 4 Control cores [DDA generator (9) +closed loop module (13)]

   ---- 4 Line drivers (10)

   ---- 4 DACS [DAC Interface (14) + DAC module (16)]

   ---- 5 Encoder interfaces

   [Encoder interface (19) + photo coupler isolation (20)]

5. For example, 2 closed loop controls and 2 open loop controls are used in the EPCIO-4000. The hardware resources are as following.

➔ 2 closed loop controls need 2 control cores, 2 DACS and 2 encoder interfaces.

➔ 2 open loop controls need 2 control cores and 2 line drivers.

Summary:-----4 Control cores => 4 used.

-----4 Line Drivers => 2 used+ 2 abandoned (4 for closed loop control).

-----2 DAC => 2 are used and the other 2 can be used by the CPU independently.

-----2 Encoder interfaces=>2 are used and the other 2 can be used by the CPU independently.

Note 2: DDA generator, refer to Fig. 1-3 and Fig. 1-4 below.

- **DDA: Digital Different Analyzer**
- **Function: The input signals of DDA generator receive position commands from the CPU (i.e. the required rotation in pulse units) and time required to execute the command (defined as DDA CYCLE TIME). After calculation, the DDA generator can send out the required pulse evenly within DDA cycle time.**
- **See Fig. 1-3. The rotation command is 1000 pulses in the positive direction (assuming the starting position for motor is at 0). The DDA CYCLE TIME is 0.5 sec. Through the DDA generator, 1000 pulses will be sent out to the motor drive smoothly and evenly within 0.5 sec. That is, the motor will rotate 1000 pulses at a constant speed.**

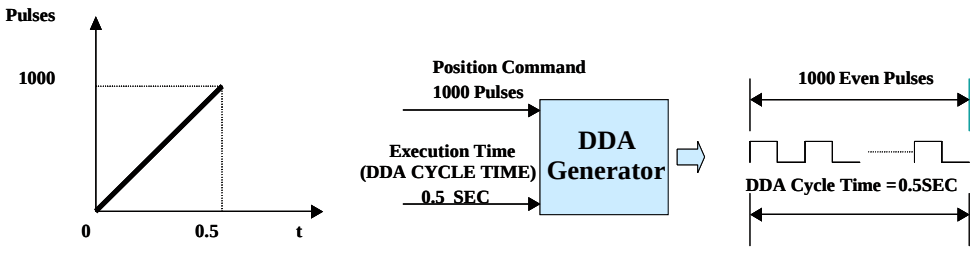


Fig. 1-3

- **See Fig. 1-4. The figure shows t (time) on the horizontal axis and motor position (unit: pulse)on the vertical axis, where Δt is the DDA CYCYLE TIME.**
- **Interval 1: The motor rotates from the 0 pulse position to 1000 pulses at a constant speed in the positive direction. The rotation speed is 1000 pulses/Δt.**

Interval 2: The motor stops at the position of 1000 pulses.

Interval 3: The motor rotates from 1000 pulses to 1500 pulses at a constant speed

in the positive direction. The rotation speed is 500 pulses/Δt.

Interval 4: The motor stops at the position of 1500 pulses.

Interval 5: The motor rotates back to the 0 position in the reverse direction. The speed is 1500 pulses/Δt.

- **The more pulses that are sent within one DDA TIME, the faster and the more distance the motor rotates.**
- **For the same rotation value, the smaller DDA TIME means the motor rotates faster.**

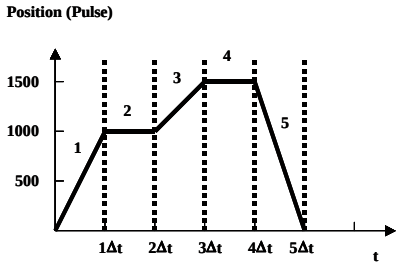


Fig. 1-4

Note 3: The pulses generated by the DDA generator have three formats: Pulse/Direction, CW/CCW and A/B Phase. (Depending on the users' motor drivers as shown below Fig.1-5).

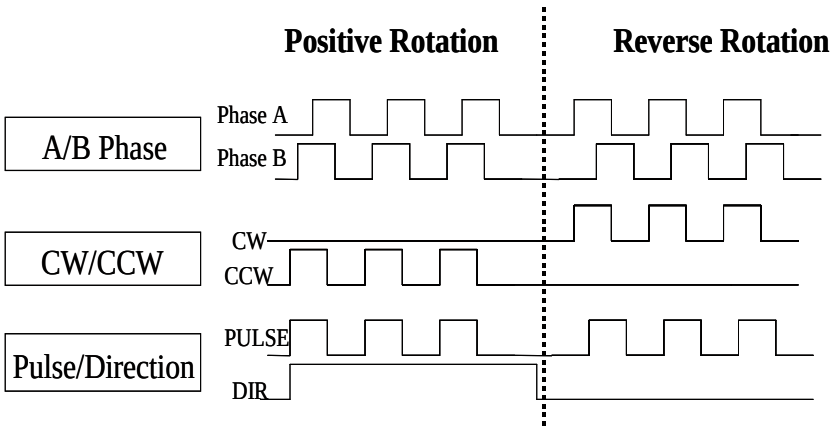


Fig. 1-5

Note 4: Transmission of differential signal (Fig. 1-6)

Transmitter	Transmission Signals		Receiver
X	X	X/	Y
0	0	1	0
1	1	0	1

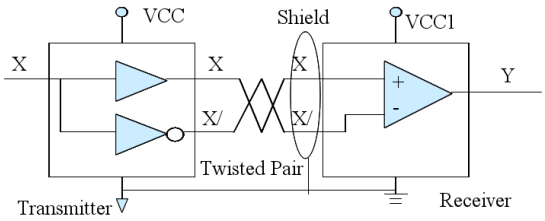


Fig. 1-6

- **Transmitter converts the input signal X into the transmission signals X and X/.**
- **Receiver compares X and X/ to obtain Y.**
- **Truth table is shown in the left corner of Fig. 1-6.**
- **Transmission using differential signals can eliminate common mode noise.**
- **Reference GND of transmitter and receiver must be connected together to prevent the leakage current damaging the transmitting terminal and receiving terminal due to potential difference.**
- **Twisted transmission wires with a foil shielding are better.**

Note 5: Pulse format transmitted from EPCIO-4000/4005 (differential signals) of Fig. 1-7.

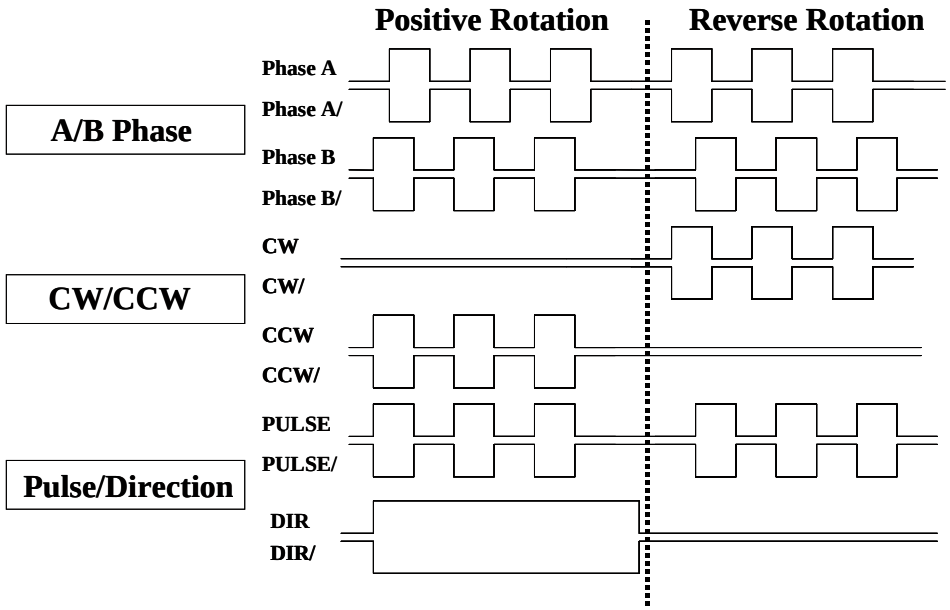


Fig. 1-7

Note 6: Encoder interface can be selected to A/B Phase, CW/CCW or Pulse/Direction format. In A/B PHASE format, the encoder pulse value can be multiplied by 0 (input forbidden) and 1, 2, or 4 times.

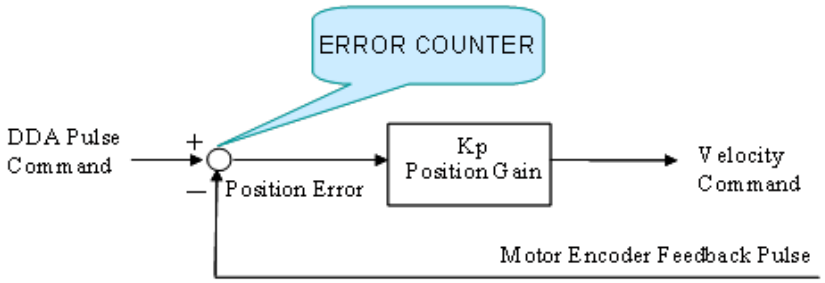
1.5.2 Four-axis Synchronous/ Asynchronous Closed Loop Control (Velocity-Voltage command)

The EPCIO-4005 does not support this function.

In Fig. 1-2, if a closed loop control is configured, the EPCIO ASIC will decode the motion position commands and trigger the DDA generator (9) to send out pulses to the closed loop module (13) for caculation. At the same time, the feedback signal of the motor encoder (18) (differential signal) use the closed loop module (13) for caculation. The closed loop module (13) uses P control algorithm to produce velocity command for driving motor via the DAC interface (14) and DAC module (16). The voltage commands are produced by multiplying gain on the difference of pulses sent by the DDA and the encoder responds to the pulses. Finally, a voltage command (–10V ~ +10V) is sent to the velocity-command servo motor (17) for velocity control.

Note 1: See Note 1 of 1.5.1.

Note 2: P-TYPE control algorithm in Fig. 1-8.



- Position Error=DDA pulse command motor encoder feedback pulse
- Position Error is recorded in the ERROR COUNTER

Fig. 1-8

Note 3: Velocity-command servo motor: It indicates that the interface of the motor driver is in the velocity command input format (using voltage to express velocity command). The voltage input range is -10V~+10V, representing maximum motor reverse speed to maximum motor forward speed. The rotation speed linearly depends on the input voltage.

1.5.3 Local Digital Input/Output

Refer to Fig.1-2. The PC driver gives a command to the ASIC on the EPCIO-4000/4005 through the PCI BUS (1). According to the user's command, the ASIC can read or write data to the local I/O (21). The output signals go through photo coupler isolation (22) and then are amplified by a Darlington circuit. The inputs are also entered to the local I/O (21) through a photo coupler isolation (22).

1.5.4 Remote Digital Input/Output

Refer to Fig.1-2. The remote digital I/O is designed with wire-saving technology patented by MSL. It uses a serial communication cable to remotely control the remote I/O control card (27) via the remote serial I/O interface (25). The maximum capacity is up to 128 outputs (28) and 128 inputs (29).

Note 1: There are two remote IO sockets on the EPCIO-4000 card. Each socket can be serially connected to a remote serial I/O control card (part no. EDIO-S001, EDIO-S002, EDIO-S003). Each EDIO-S00X has 64 inputs and 64 outputs.

The EPCIO-4005 can only connect to one EDIO-S00X with 64 inputs and 64 outputs.

1.5.5 6-Channel Analog-to-Digital Converter

Optional for the EPCIO–4000 ; the EPCIO-4005 does not support this function.

In Fig.1-2, the 6 analog voltage signals (32) (selecting -5V~5V for the bipolar mode or 0V ~10V for the unipolar mode) are connected to the ADC module (31) for conversion and then the EPCIO-4000 reads the ADC value (12 bit resolution) through the ADC interface (30).

1.5.6 4-Channel Digital-to-Analog Converter (4 DAC)

The EPCIO–4000 is optional; the EPCIO-4005 does not support this function.

The EPCIO-4000 supplies 4-channel voltage output interface each with an analog voltage output with an output range of ±10V. Each analog voltage output can be used with encoder input and pulse command output in a closed loop control module. The D/A converter can be also used independently when it is not in the closed loop control.

Refer to the system block diagram Fig.1-2, the EPCIO-4000 sends a command to the DAC interface (14) through the system control circuit (8), and the DAC module (16) translates the command into an analog voltage output. The motion card has been adjusted, so that the voltage offsets are close to 0V

before the card is shipped. Therefore the DAC does not need to modify the voltage offsets when the user independently uses a DAC converter.

Please refer to 1.5.2 4-axis Synchronous/Asynchronous closed loop motion control when the user uses the hardware closed loop control mode. When a motor driver is connected to the EPCIO, the reference voltage between D/A converters in the EPCIO and motor driver have tiny offsets, which will lead the motor to produce a slow drift. This is a normal function. Once the closed loop is enabled, the closed loop circuit will automatically correct the drift action. The motor will be locked at this time by the EPCIO internal error counter to read the offset value.

If offset value is not set at 0, then you can adjust the variable resister to modify the value of the voltage offset.

Chapter 2 Specifications

A description such as "➔ EPCIO_SetWaitState()", please refer to EPCIO_SetWaitState() function in the "**EPCIO Series Device Driver Library User Manual**".

2.1 System Architecture

- Dimensions: 174x107mm
- System clock: 40Mhz
- Bus interface: 32-bit PCI
- Interrupt

Note 1: There are many interrupt generators in the EPCIO-4000/4005 (to be described in detail in the following sections). Each interrupt generator has a latch. When interrupt occurs for its dedicated purpose, the interrupt signal will be sent to the CPU and the corresponding latch will be set to 1 to record what kind of interrupt has occurred. One can scan interrupt latches to recognize what interrupt has occurred and the latches' value will be cleared to 0 after reading the active latch.

Note 2: Each interrupt can be Enabled/Disabled (default: all disabled).

- Reset

  Reset DAC, ADC, position closed loop control mechanism (PCL), DDA generator, local I/O, remote I/O, and EPCIO ASIC peripheral individually or reset all completely.

  ➔ EPCIO_ResetModule()

2.2 Motion Control Specifications

There are four motion control cores in the EPCIO ASIC. Each core can be selected to closed loop control (Velocity output) or open loop control (Pulse output). The user can select M sets of closed loop control and N sets of open loop control, M+N must satisfy M+N≦4.

The EPCIO-4005 only supports open loop pulse output control.

2.2.1 Open Loop Control (Pulse Command)

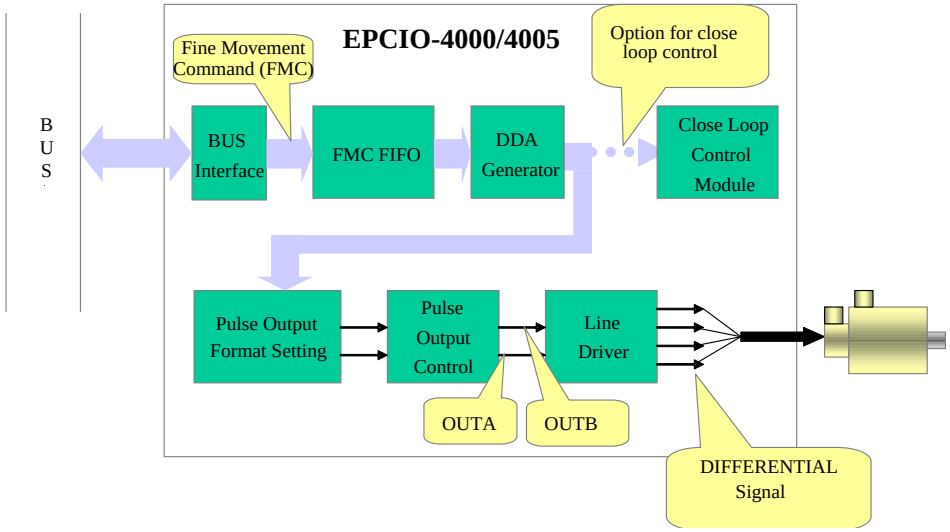


Fig. 2-1

- Fine motion command format: One direction bit with 10~15 bits movement value
  Unit: Pulses (i.e. square wave wave(s))
- Pulses output control module: DDA
- Maximum controllable axis: Four (Each axis has one DDA controller)
- DDA Specifications
  - Enable/Disable: Enabling or disabling the DDA of each axis (default: disable)
    - ➔ EPCIO_DDA_EnableOutputChannel()
    - ➔ EPCIO_DDA_DisableOutputChannel()
    - ➔ EPCIO_DDA_StartEngine()
    - ➔ EPCIO_DDA_StopEngine()
  - DDA Engine Length: 10~15 bits
    - ➔ EPCIO_DDA_SetBitLength()

- Maximum pulse rate: 1024 ~32767 Pulses in a DDA cycle

  Maximum pulse rate can be adjusted by setting the length of DDA Engine. If the DDA Engine Length=10 bits, then the maximum pulses per DDA cycle time is 1023 pulses ((2^{10})-1). When the DDA Engine Length=15 bits, the maximum pulses per DDA cycle is 32,767 pulses ($(2^{15)}$-1).

  Note: Refer to Fig.1-3 and Fig.1-4.

- DDA cycle time interrupt

  DDA cycle time interrupt can be generated at the end of each DDA cycle time. Under such conditions, the EPCIO-4000/4005 will generate an interrupt for every DDA cycle. The users should make sure whether your PC can handle such frequent interrupt requests.

  ➔ EPCIO_DDA_EnableCycleInt()

  ➔ EPCIO_DDA_DisableCycleInt()

- DDA Clock Divider: 12 bit

  ➔ EPCIO_DDA_SetClockDivider()

- DDA Cycle Time: 25us ~3350ms Programmable

  DDA Cycle Time=25 ns X (DDA Cock Divider value +1) X 2^(DDA Engine Length)

  ➔ EPCIO_DDA_SetClockDivider()

  ➔ EPCIO_DDA_SetBitLength()

  Or by calling ➔ EPCIO_DDA_SetTime()

  Note: Refer to Fig.1-3 and Fig.1-4.

- Pulse Width Extender (Pulse/Direction and CW/CCW formats): 12 bit

  Pulse Width = 25 ns × n, where n=1～4096，n defaults to 1.

  ➔ EPCIO_DDA_SetPulseWidth()

■ Fine Movement Motion command (FMC) FIFO: Pre-store the position commands. The DDA engine will fetch a command from FIFO at the end of every DDA cycle.

- Structure: 64 x 16-bit FIFO (First In First Out)
- Command format: One direction bit with 10~15 bits movement value

  ➔ EPCIO_DDA_SendPulse()

- Full flag: Indicates that FIFO is full

  ➔ EPCIO_DDA_CheckFIFOFull( )

- Empty flag: Indicates that FIFO is empty

  ➔ EPCIO_DDA_CheckFIFOEmpty( )

- Number of unexecuted command(s) left in the register.
  - ➔ EPCIO_DDA_GetStockCount( )
- Current command being executed.
  - ➔ EPCIO_DDA_GetCurrentCmd( )
- Minimum command threshold interrupt: When the number of command(s) left in FIFO is equal to the minimum command interrupt threshold, the system will generate an interrupt. This method will lower the frequency of interrupt(s) to decrease the loading of CPU as compared to the DDA cycle time interrupt mentioned previously. In general, the software programmers can use Minimum command threshold interrupt or DDA cycle time interrupt to put command into FIFO.
  - ➔ EPCIO_DDA_SetMinStockNo()
  - ➔ EPCIO_DDA_EnableStockInt()
  - ➔ EPCIO_DDA_DisableStockInt()

■ Pulse output format (default: Pulse/Direction)
- Pulse/Direction
- CW/CCW
- A/B (motion command ÷ 4)
- Inhibit (default: Disable output)
  - ➔ EPCIO_DDA_SetOutputFormat()

Note: Refer to Fig. 1-5 and Fig. 1-7.

■ Pulse output control (Refer to Fig. 2-1, assuming that output pulse are OutA and OutB)
- Out A and Out B can be reversed respectively (default: non-reverse).
  - ➔ EPCIO_DDA_EnableOutAInverse()
  - ➔ EPCIO_DDA_DisableOutAInverse()
  - ➔ EPCIO_DDA_EnableOutBInverse()
  - ➔ EPCIO_DDA_DisableOutBInverse()
- Interchanging of OutA and OutB signals (default: non-interchange).
  - ➔ EPCIO_DDA_EnableOutABSwap()
  - ➔ EPCIO_DDA_DisableOutABSwap()

■ Line driver: The output with 5V differential signals.

Note: For differential signals, please refer to Fig. 1-6.

2.2.2 Closed Loop Control (Velocity Command)

The EPCIO-4005 does not support this function.

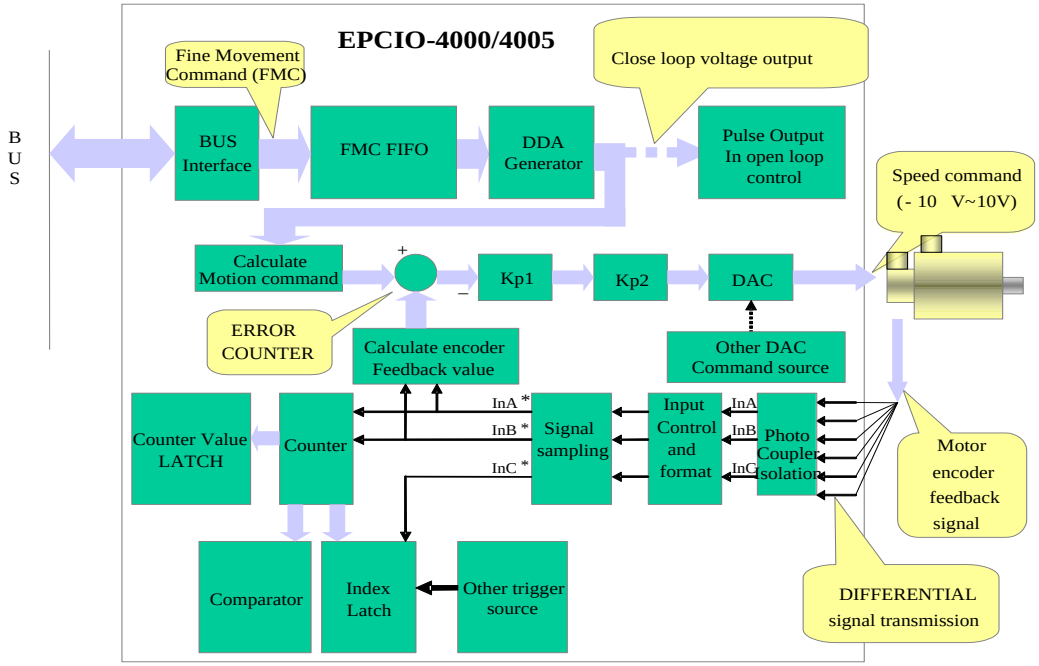


Fig. 2-2

- Fine motion command format: One direction bit with 10~15 bits movement value.
  Unit: Pulses (i.e. square wave(s))
- Maximum controllable axis: 4 (Each axis has one closed loop control module)
- Range of velocity command: +/- 10V via DAC output
- DDA specifications (For details please refer to 2.2.1)
- FMC FIFO (For details please refer to 2.2.1)
- Control algorithm: P type
  - ➔ EPCIO_PCL_StartControl()
  - ➔ EPCIO_PCL_StopControl()
  - ◆ ERROR COUNTER: Length is 16 bits, and it can be set to generate the interrupt after error counter overflows.
    - ➔ EPCIO_PCL_EnableErrorCounter()
    - ➔ EPCIO_PCL_DisableErrorCounter()
    - ➔ EPCIO_PCL_GetErrorCounter()
    - ➔ EPCIO_PCL_ClearErrorCounter()
    - ➔ EPCIO_PCL_EnableOverflowInt()
    - ➔ EPCIO_PCL_DisableOverflowInt()

- Kp1: Closed loop scaling gain
  - ➔ EPCIO_PCL_SetScaleGain()
- Kp2: Closed loop shift gain
  - ➔ EPCIO_PCL_SetScaleGain()

The relationship between the error counter and the DAC output voltage after P control law can be converted as in the following formula: Output voltage = (position error) × Kp1 × 2^{Kp2} × 10 ÷ (16 × 32767). For example, if position error = 1024, Kp1 = 100、Kp2 = 0, the output voltage will be: Vout =1024 × 100 × 1 × 10 ÷ (16×32767) = 1.95 volt.

2.3 Digital-to-Analog Converter Specifications

- Channel: 4
  - In closed loop control mode, the velocity-command of the DAC must come from the PCL module. A closed loop control function needs a DAC converter and an encoder counter. When a closed loop control mode is disabled, the corresponding DAC converter and the encoder counter can be used by the CPU for other purposes.

    In the EPCIO-4005, the DAC does not support.

    ➔ EPCIO_DAC_SetOutput()
- START or STOP (default: STOP). When the DAC stops, the DAC output value is kept on the previous value.
  - ➔ EPCIO_DAC_StartConv()
  - ➔ EPCIO_DAC_StopConv()
- Resolution: 16 bit
- Power on value of the EPCIO-4000/4005: 0V (Please adjust the variable resistor if not at zero)
- DAC output interface specifications

  Load: Must be over 2KΩ.

  Output voltage swing：±10V maximum
- DAC command source (default: PCL)

  ➔ EPCIO_DAC_SetCmdSource()

  1. PCL: The PCL generates a signal that is sent to the DAC to make the output voltage.
  2. Direct Write Buffer (DAC stand-alone operation mode)

  Operation: the CPU sends a DAC conversion value into the direct write buffer first. Then the DAC will convert this value into an analog voltage.

  ➔ EPCIO_DAC_SetOutput()

  3. Trigger Buffer (the DAC stand-alone operation mode).

  Operation: Pre-store a conversion value into the trigger buffer. When a TRIGGER signal occurs, the conversion value in the trigger buffer will be sent to the DAC and converted to a voltage.

  ➔ EPCIO_DAC_SetTrigOutput()

  Number of trigger signal sources: 23. For details please refer to the following three functions.

  ➔ EPCIO_DAC_SetTrigSource()

  ➔ EPCIO_DAC_EnableTrigMode()

  ➔ EPCIO_DAC_DisableTrigMode()

2.4 Encoder Input Specifications

- Channels: 5
- Encoder input isolation

  Isolation: Photo coupler

  Input signal frequency: 2MHz maximum
- Input signal controls (InA, InB, and InC)

  1. Polarity Inversion (default: non-inversed phase).
  - ➔ EPCIO_ENC_EnableInAInverse( )
  - ➔ EPCIO_ENC_DisableInAInverse( )
  - ➔ EPCIO_ENC_EnableInBInverse( )
  - ➔ EPCIO_ENC_ DisableInBInverse( )
  - ➔ EPCIO_ENC_EnableInCInverse( )
  - ➔ EPCIO_ENC_DisableInCInverse( )

  2. Swap for In A and In B (default: non-swap).
  - ➔ EPCIO_ENC_EnableInABSwap()
  - ➔ EPCIO_ENC_DisableInABSwap()
- Input format (setting for InA and InB)
  - ➔ EPCIO_ENC_SetInputType()

  1. A/B phase format

     Multiplication can be set to ×0, ×1, ×2 or ×4 (default to ×0)
  - ➔ EPCIO_ENC_SetInputRate()

  2. CW/CCW format

  3. Pulse/Direction format

  4. Input disabled.
- Input signal sampling and digital filter function
  - ➔ EPCIO_ENC_SetFilterClock()

  1. Format: Continuous Three (for InA，InB，InC)

  2. Sample rate: Programmable.

  3. Sample rate = 40 MHz ÷ (n+1), n=0~255 (default n=0).
- Counter

  1. Length: 32 bit

  2. Enable/disable (Note: Under closed loop control it must be set to enable.)
  - ➔ EPCIO_ENC_StartInput()
  - ➔ EPCIO_ENC_StopInput()

  3. Clear counter value to 0 (default: not cleared).
  - ➔ EPCIO_ENC_ClearCounter()

4. Random value after power on.
5. Read the counter value.

➔ EPCIO_ENC_GetValue()

◆ Counter value latch

1. Latch counter values when index signal or other trigger signals enter. The trigger mode is also selectable.

➔ EPCIO_ENC_SetTrigSource()
➔ EPCIO_ENC_SetTrigMode()
➔ EPCIO_ENC_GetLatchValue()

2. Can be set to Enable or Disable (default: Disable)
(Refer to the "EPCIO Series Motion Control Command library User Manual")

◆ Index latch

1. Index signal: To read the status of the index signal (high/low).

➔ EPCIO_ENC_GetIndexStatus()

2. Index interrupts: To generate an interrupt using an index signal.

➔ EPCIO_ENC_EnableIndexInt()
➔ EPCIO_ENC_DisableIndexInt()

◆ Comparator and comparator interrupt

Function: If the counter value is equivalent to the preset comparator value, the comparator flag is set to 1 and generates an interrupt.

➔ EPCIO_ENC_SetCompValue()
➔ EPCIO_ENC_EnableCompInt()
➔ EPCIO_ENC_DisableCompInt()

◆ Encoder inputs interrupt summary

1. Index can generate an interrupt directly (4 sets).
2. When "Compare equal" is valid, an interrupt is generated. (4 sets)
3. Each interrupt can be Enabled/Disabled (default: Disabled).

2.5 Local Digital Input/Output

- 13 dedicated input points:
  - Local input can be operated independently by the CPU and not related to other functions.
  - Operating voltage: DC 24V±10%.
  - When the input is 18V~30V (voltage difference across COM and input point), the internal EPCIO ASIC reading value is 0.
  - When the input is 0V~1V (voltage difference across COM and input point), the internal EPCIO ASIC reading value is 1.
  - Isolation: Photo coupler.
  - Types:

    1. Positive travel limit inputs: 4, labeled as OT1+, OT2+, OT3+, and OT4+ respectively. For Pin definition please refer to Chapter 3.

    ➔ EPCIO_LIO_GetOverTravelUp()

    2. Negative travel limit inputs: 4, labeled as OT1-, OT2-, OT3-, and OT4- respectively.

    ➔ EPCIO_LIO_GetOverTravelDown()

    3. Home sensor inputs: 4, labeled as HOM1, HOM2, HOM3, and HOM4, respectively.

    ➔ EPCIO_LIO_GetHomeSensor()

    4. Emergency stop input: 1, labeled as ESTP.

    When the emergency stop occurs (i.e. emergency stop input value is 1), the hardware will cut off the pulse outputs and set the DAC to 0V. The EPCIO-4000/4005 build-in LATCH will latch the state of the emergency stop.

    ➔ EPCIO_LIO_GetEmgcStopStatus()

Note 1: Emergency Stop Input Procedure: Please resolve the cause of emergency stop first (i.e. make the emergency stop input value to 0). Then use software to reset EPCIO ASIC peripheral. In this way, emergency stop latch state will be cleared and then, and the system can be restarted again.

➔ **EPCIO_ResetModule()**

Note 2: Shorting the E_STOP of JP5 will disable the emergency stop function. (i.e. the emergency stop input value is always 0.)

Note 3: JP5 is shorted when the EPCIO-4000/4005 is shipped out. JP5 can be removed for E_STOP functioning.

Note 4: If DAC output voltage is not 0V when under E_STOP, adjust the variable resistor to make the DAC to 0V. Please refer to 3.2.2.5.

- 5 dedicated output points
  - Operating voltage: DC 24V ±10%
  - Output driving stage: OPEN COLLECTOR. When the EPCIO ASIC internal output value is 0, OPEN COLLECTOR is in an ON state. (1 is in an OFF state)
  - Maximum load current /each point: 60mA (Do not connect 24V power directly. It is hazardous and will cause an overload.).
  - Isolation: Photo coupler
  - Types

    1. Servo on/off outputs: 4 points, and they are named SVON1, SVON2, SVON3, and SVON4 respectively.
       - ➔ EPCIO_LIO_ServoOff ()
       - ➔ EPCIO_LIO_ServoOn ()
    2. POSITION READY outputs: 1, which is used to inform a peripheral device whether the EPCIO-4000/4005 is READY or not.
       - ➔ EPCIO_LIO_EnablePrdy ()
       - ➔ EPCIO_LIO_DisablePrdy ()

- Safety control output (Pulse_DA_output_enable): When the system is under initialization, there may be a period of instability. To ensure that during this period the motor will not make an unpredictable action/motion, the EPCIO-4000/4005 is designed with an internal dedicated safety control output. When the power is on, the pulse output and the DAC output are cut off. The User should call EPCIO_LIO_EnablePulseDAC () after success of initialization in order for the pulse output to be enabled and the DAC output to be enabled.

Note 1: Before activating the pulse_DA_output_enable, ensure the system is not in the emergency stop state; otherwise the activation of the safety control output will not work.

➔ **EPCIO_LIO_EnablePulseDAC ()**

Note 2: When closing off the pulse_DA_output_enable, no matter how other internal portions of the EPICO-4000/4005 are set, the pulse output and the DAC output are in the cutoff state.

➔ **EPCIO_LIO_DisablePulseDAC ()**

2.6 Remote Digital Input/Output

- There are two remote I/O sockets labeled as (RIO1 and RIO2) on the EPCIO-4000 card. (The EPCIO-4005 only has RIO1 socket.) Each socket can be connected to a remote serial I/O module (part number EDIO-S00X). Each remote serial I/O module has 64 inputs and 64 outputs.

Note: For specifications of EDIO-S001/2/3 please refer to the EDIO-S001/2/3 user manual.

  - The remote I/O module connected to RIO1 is denoted as RIO_SET0 and RIO_SLAVE0 in the software driver.
  - The remote I/O module connected to RIO2 is denoted as RIO_SET1 and RIO_SLAVE1 in the software driver.
    - ➔ EPCIO_RIO_GetInputValue()
    - ➔ EPCIO_RIO_SetOutputValue()

- The remote I/O can be operated by the CPU independently and is not related to other functions.
  - ➔ EPCIO_RIO_EnableSetControl()
  - ➔ EPCIO_RIO_DisableSetControl()
  - ➔ EPCIO_RIO_EnableSlaveControl()
  - ➔ EPCIO_RIO_DisableSlaveControl()
- Communication control: (Refer to the “EPCIO series Device Driver Library User Manual” for details.)
  - ➔ EPCIO_RIO_SetClockDivider()
  - ➔ EPCIO_RIO_GetTransStatus()
  - ➔ EPCIO_RIO_GetMasterStatus()
  - ➔ EPCIO_RIO_GetSlaveStatus()
  - ➔ EPCIO_RIO_SetTransError()
  - ➔ EPCIO_RIO_EnableTransInt()
  - ➔ EPCIO_RIO_DisableTransInt()
- Interrupt
  - The first four points of each EDIO-S00X module can be set to generate an interrupt.
    - ➔ EPCIO_RIO_EnableInputInt()
    - ➔ EPCIO_RIO_DisableInputInt()
    - ➔ EPCIO_RIO0_GetIntCondition()
    - ➔ EPCIO_RIO1_GetIntCondition()
  - Triggering mode (for the 4 interrupt mentioned above)
    - ➔ EPCIO_RIO_SetIntType()

- Communication failure interrupt (Refer to the “EPCIO series Device Driver Library User Manual” for details.)
  - ➔ EPCIO_RIO_SetTransError()
  - ➔ EPCIO_RIO_EnableTransInt()
  - ➔ EPCIO_RIO_DisableTransInt()

2.7 Analog-to-Digital Converter Spcifications

Optional for the EPCIO-4000；The EPCIO-4005 does not support.

- Inputs: 6
- Input range
  - BIPOLAR MODE: -5V~5V (short the BIP and COM of JP6).
  - UNIPOLAR MODE: 0~10V (short the UNI and COM of JP6).
    - ➔ EPCIO_ADC_SetConvType()
- Resolution: 12 bits
    - ➔ EPCIO_ADC_GetInput()
- SINGLE RUN mode
  - Chose one of the 6 channels, and make the ADC conversions.
  - Data Update Time (including data transmission time)=10us
    - ➔ EPCIO_ADC_SetConvMode()
    - ➔ EPCIO_ADC_SetSingleChannel()
    - ➔ EPCIO_ADC_StartConv()
    - ➔ EPCIO_ADC_GetInput()
    - ➔ EPCIO_ADC_StopConv()
- FREE RUN mode
  - 6 Channels can be enabled at the same time (must be assigned first). Conversion will only be switched among these channels and continually make conversions.
  - Data Update Time (including data transmission time) =10us x (Enable channels)
    - ➔ EPCIO_ADC_SetConvMode()
    - ➔ EPCIO_ADC_EnableConvChannel()
    - ➔ EPCIO_ADC_DisableConvChannel()
    - ➔ EPCIO_ADC_StartConv()
    - ➔ EPCIO_ADC_GetInput()
    - ➔ EPCIO_ADC_StopConv()
- Comparator and comparator interrupt
  - Set comparator values first. After comparison the result will generate an interrupt to inform the CPU. There are 6 comparator interrupts in total as the following three functions.
    - ➔ EPCIO_ADC_SetCompValue()
    - ➔ EPCIO_ADC_EnableCompInt()
    - ➔ EPCIO_ADC_DisableCompInt()

- MASK function

  At first, the ADC voltage comparator mask the last 0, 1, 2 or 3 bits to form the mask value, and then the preset value will compare with the mask value. In addition, the results of the comparison would inform the CPU through an interrupt. (Note: When reading the value of the REFRESH ADC voltage, the mask value will be updated.)

  ➔ EPCIO_ADC_SetCompMask()

- Method of comparison: The following three comparison methods can be selected to trigger an interrupt.

  ➔ EPCIO_ADC_SetCompType()

  1. Low-to-high

     The instant when the mask value is changed from less than the preset value to greater than or equal to the preset value.

  2. High-to-low

     The instant when mask value is changed from greater than or equal to preset value to smaller than the preset value.

  3.Both

     Both conditions above can trigger an interruption.

■ Interrupt

- Comparator interrupt: 6

  ➔ EPCIO_ADC_EnableCompInt()

  ➔ EPCIO_ADC_DisableCompInt()

- Conversion complete interrupt: 1

  When one of the ADC input has completed a conversion, it can be set to generate an interrupt.

  ➔ EPCIO_ADC_EnableConvInt()

  ➔ EPCIO_ADC_DisableConvInt()

- Tag channel conversion completion interrupt: 1

  Set one channel as “tagged” input. When the channel has completed conversion, it generates an interrupt.

  ➔ EPCIO_ADC_SetTagChannel()

  ➔ EPCIO_ADC_EnableTagInt()

  ➔ EPCIO_ADC_DisableTagInt()

2.8 Timer and Watchdog

- Timer
  - Enable/Disable
    - ➔ EPCIO_LIO_EnableTimer()
    - ➔ EPCIO_LIO_DisableTimer()
  - Timing unit: System clock (25ns).
  - Timer length: 24 bit

    Explanation: The time-out range of the timer can be set to 0~ ((2^{24})-1) timing units. It can be set to generate an interrupt at the end of the timer running down.
    - ➔ EPCIO_LIO_SetTimer()
    - ➔ EPCIO_LIO_EnableTimerInt()
    - ➔ EPCIO_LIO_DisableTimerInt()
- Watchdog timer
  - Enable/Disable
    - ➔ EPCIO_LIO_EnableWDogTimer()
    - ➔ EPCIO_LIO_DisableWDogTimer()
  - Timing unit: Timer length set by timer.
  - Watchdog timer length: 16 bit

    Explanation: The time-out range of watchdog timer can be set to 0~((2^{16})-1) timing units (i.e. 0~((2^{16})-1) times of timer length). When watchdog time ends the EPCIO-4000/4005 will automatically generate the RESET signal (length of the RESET signal is programmable). If the RESET is not desired, the watchdog time must be cleared to 0 with software before the watchdog timer runs out.
    - ➔ EPCIO_LIO_SetWDogTimer()
    - ➔ EPCIO_LIO_SetWDogReset()
    - ➔ EPCIO_LIO_RefreshWDogTimer()

Chapter 3 Hardware Installation

3.1 Basic Installation

A. Please execute setup.exe on the installation CD first.

B. Turn off the power before installation, including the computer and the motor(s).

C. Insert the EPCIO-4000/4005 into the PCI BUS and fix the card bracket.

D. Wire the peripheral circuits include motors, I/Os, ADCs, etc and finally connect the SCSI II 68 PIN cable into the socket on the bracket of the EPCIO-4000/4005.

E. Make sure the computer, the motors and other circuits are connected to the same ground in order to avoid the system being damaged due to voltage difference between the grounds.

F. Turn on the computer and install the software by following the installation guide on the installation CD.

G. Install the test program if needed (see the “EPCIO series Motion Control Command Library Integrated Testing Environment”). With the test program, you can start testing the EPCIO-4000/4005 and learn to use the card.

H. For details please refer to the “EPCIO series Motion Control Card Installation Manual”.

3.2 Board Layout and Definition of Connectors

3.2.1 Board Layout

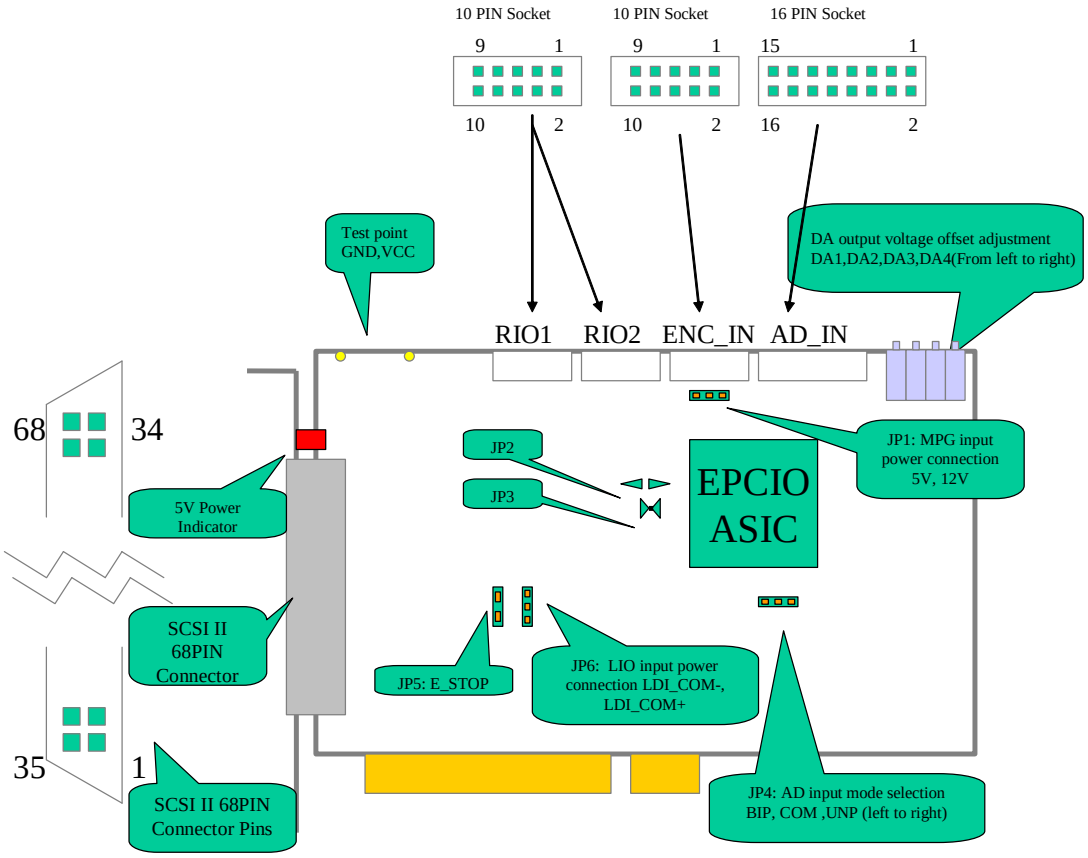


Fig. 3-1

Note 1: The EPCIO-4005 does not have RIO2, AD_IN connector and JP4.

Note 2: The EPCIO-4005 does not have DA output voltage adjustment variable resistor.

3.2.2 Definition of Each Connector on the Board

3.2.2.1 SCSI II 68 PIN connection definition---As below table

SCSI II-68 PIN CONNECTOR			
Pin Definitions	**Pin no**	**Pin no**	**Pin Definitions**
AGND	1	35	AGND
AGND	2	36	DAC1
DAC2	3	37	DAC3
DAC4	4	38	+5V
COM+	5	39	COM-
ESTP	6	40	PRDY
HOME1	7	41	HOM2
OT1+	8	42	OT2+
OT1-	9	43	OT2-
SVON1	10	44	SVON2
HOM3	11	45	HOM4
OT3+	12	46	OT4+
OT3-	13	47	OT4-
SVON3	14	48	SVON4
EA1+	15	49	EA2+
EA1-	16	50	EA2-
EB1+	17	51	EB2+
EB1-	18	52	EB2-
EC1+	19	53	EC2+
EC1-	20	54	EC2-
EA3+	21	55	EA4+
EA3-	22	56	EA4-
EB3+	23	57	EB4+
EB3-	24	58	EB4-
EC3+	25	59	EC4+
EC3-	26	60	EC4-
PA1+	27	61	PA2+
PA1-	28	62	PA2-
PB1+	29	63	PB2+
PB1-	30	64	PB2-
PA3+	31	65	PA4+
PA3-	32	66	PA4-
PB3+	33	67	PB4+
PB3-	34	68	PB4-

■ Description

◆ DDA pulse outputs

Signal	Description	Reference Potential	Remark
PAn+ & PAn-	Phase A: differential signals of the nth DDA output.	AGND	n=1~4
PBn+ & PBn-	Phase B: differential signals of the nth DDA output.	AGND	n=1~4

◆ ENCODER input signals

Signal	Description	Reference Potential	Remark
EAn+ & EAn-	A-phase differential signals of the nth Encoder Input	AGND	n=1~4
EBn+ & EBn-	B-phase differential signals of the nth Encoder Input	AGND	n=1~4
ECn+ & ECn-	C-phase differential signals of the nth Encoder Input (Index)	AGND	n=1~4

◆ Local digital I/O

Signal	Description	Reference Potential	Remark
OTn+	Positive travel limit input of the nth axis	COM	n=1~4
OTn-	Negative travel limit input of the nth axis	COM	n=1~4
HOMn	Home input of the nth axis	COM	n=1~4
SVONn	Servo On output of the nth axis	COM-	n=1~4
ESTP	Emergency stop input	COM	
PRDY	Position Ready Output	COM-	
COM+	Power terminal (+) of Local Digital Output		
COM-	Power terminal (-) of Local Digital Output		
COM	Common terminal of Local Digital Input		

◆ DAC outputs and others

Signal	Description	Reference Potential	Remark
DACn **(only for the EPCIO-4000)**	The nth analog voltage output	AGND	n=1~4
AGND	Analog GND (connected with Digital GND)		
+5V	+5V Output (Max: 500mA)	AGND	

3.2.2.2 Definition of RIO1 and RIO2 connector (Refer to fig. 3-2)

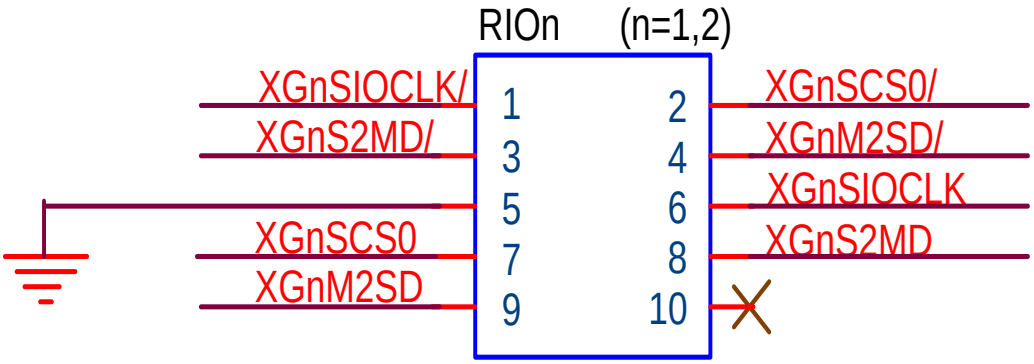


Fig. 3-2

■ Description:

Signal	Description	Reference Potential	Remark
xGnM2SD and xGnM2SD/	The nth Remote I/O. Serial data signals transmitted to Slave by Master. (Differential transmission)	AGND	n=1,2
xGnSIOCLK and xGnSIOCLK/	The nth Remote I/O. Synchronous signals transmitted to Slaves by Master (Differential transmission)	AGND	n=1,2
xGnSCS0 and xGnSCS0/	The nth Remote I/O. Selection signals transmitted to Slaves by Master (Differential transmission)	AGND	n=1,2
GnS2MD and GnS2MD/	The nth Remote I/O. Serial data signals transmitted to Master by activated Slave (Differential transmission).	AGND	n=1,2
DGND(PIN 5)	DIGITAL GND connection to AGND		

Note: The EPCIO-4005 only has RIO1 connector.

3.2.2.3 Definition of ENC_IN(MPG) Connector

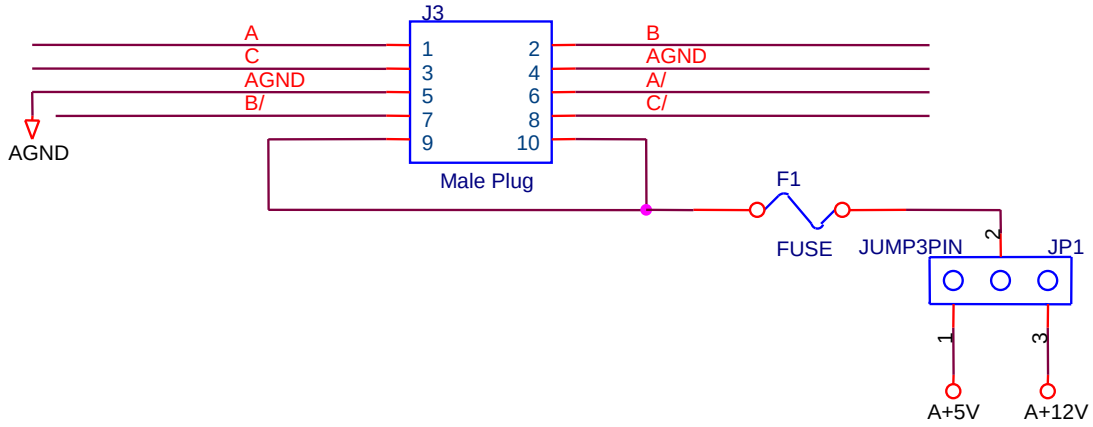


Fig. 3-3

■ Description:

Signal	Description	Reference ground	Remark
A and A/	Encoder A-Phase differential input signal	AGND	
B and B/	Encoder B-Phase differential input signal	AGND	
C and C/	Encoder C-Phase differential input signal	AGND	
A+5V / A+12V	Positive output of +5V or +12V power for encoder provided by JP1.	AGND	
AGND	Negative output of +5V or +12V power for encoder provided by JP1.	AGND	

3.2.2.4 Definition of ADC Connector

Optional for the EPCIO-4000; the EPCIO-4005 does not surppot.

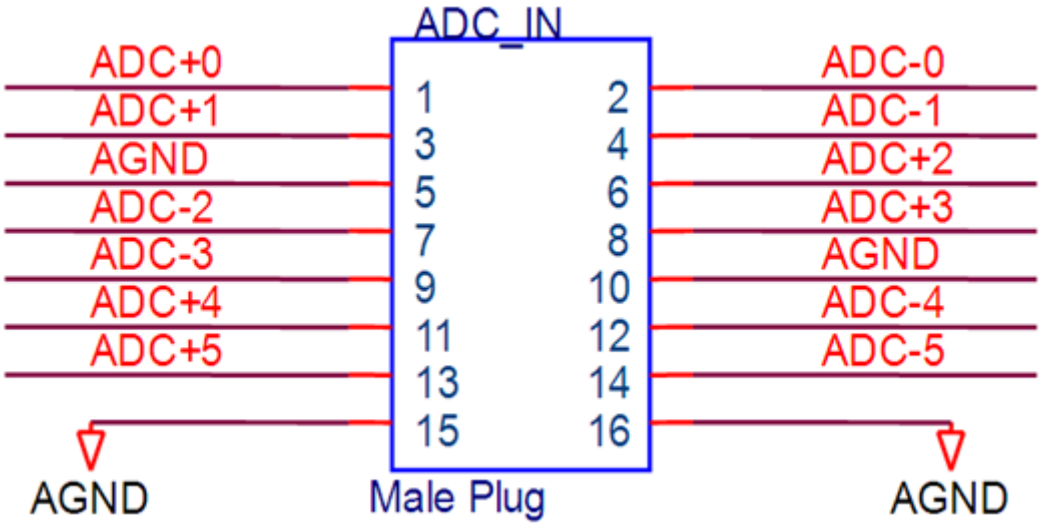


Fig. 3-4

■ Description:

Signal	Description	Reference Potential	Remark
ADC+n	Positive terminals of the nth ADC analog differential input signal	AGND	n=0~5
ADC-n	Negative terminals of the nth ADC analog differential input signal	AGND	n=0~5
AGND	ANALOG GND is the common GND for VCC_OUT, DAC OUTPUT and ADC. It is connected with DGND (DIGITAL GND) and DGND is connected with BUS ground of the computer		

3.2.2.5 Others—refer to board layout diagram shown in Fig. 3-1

A. Testing points GND and VCC

---- GND is DIGITAL GROUND (or PCI BUS GROUND)

---- VCC is the power of DIGITAL 5V (or PCI BUS 5V)

B. 5V power indicator

---- Light, indicating VCC (PCI BUS 5V)

C. Variable resistors VR1 (DA1), VR2 (DA2), VR3 (DA3), VR4 (DA4).

Only for the EPCIO-4000.

---- For adjustment of the DAC amplifier output OFFSET voltage.

---- When the DAC is used for closed loop control mode, the user can adjust the variable resistor of each axis so that the error counter value becomes minimum (error counter value = error signal = Target position – Feedback position)

---- When using the DAC alone, the user can adjust the variable resistor to obtain 0V output voltage.

D .JP4—AD input mode selection

Optional for the EPCIO-4000；the EPCIO-4005 does not support.

---- If the range of the input voltage for the ADC is between -5V~5V, please select the BIPOLAR MODE (Short COM and BIP of JP4).

---- If the range of the input voltage for the ADC is between 0V~10V, please select the UNIPOLAR MODE, (Short COM and UNP of JP4).

---- The default for JP4 is the BIPOLAR MODE

---- Refer to section 2.7.

E. JP5—E_STOP

---- Shorting the E_STOP of JP5 can disable the emergency stop function

---- The default for JP5 is shorted.

---- Refer to 2.5 for explanations of emergency stop input.

3.3 Wiring

3.3.1 Four-axis Synchronous/Asynchronous Closed Loop Control

Only for the EPCIO-4000.

The circuit diagram is for the EPCIO-4000 and the velocity-command servo motor driver.

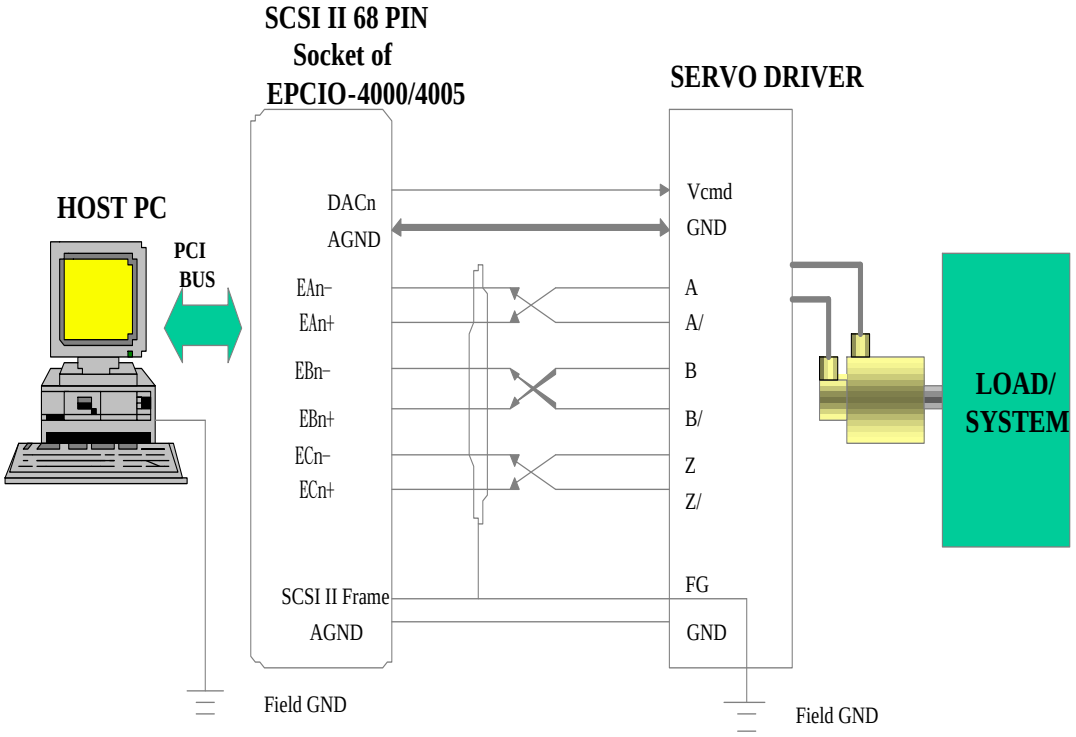


Fig. 3-5

- DACn is the velocity command output of the nth closed loop control module. It must be connected to the Vcmd (VELOCITY COMMAND) input of the nth SERVO DRIVER. The signal ground (AGND) of the EPCIO-4000 must be connected with the driver signal ground (GND).
- The motor encoder signal of the SERVO DRIVER (A/B/Z signals) must be connected back to the EPCIO-4000 in differential form (as shown in the diagram Fig. 3-5). It is recommended that the three sets of signals A and A/, B and B/, and Z and Z/ are suggested using twisted-pair cable to reduce the common mode noise. In addition, as illustrated in the diagram a shield net is used to reduce the interference of electromagnetic.
- Connect one side of the shield net to the external housing of SCSI II 68 PIN (connected to PC frame ground) and the other side to SERVO DRIVER

Field Ground (FG). Then make sure Frame ground of both PC and SERVO DRIVER are connected together to the same field GND.

- Important---A ground wire must be connected between the GND of SERVO DRIVER and AGND of the EPCIO-4000 (This is very important as it could result in element damage due to ground voltage difference).
- Functions of the wiring board (the EPCIO-400-1/2) in closed loop control are bypass signals.

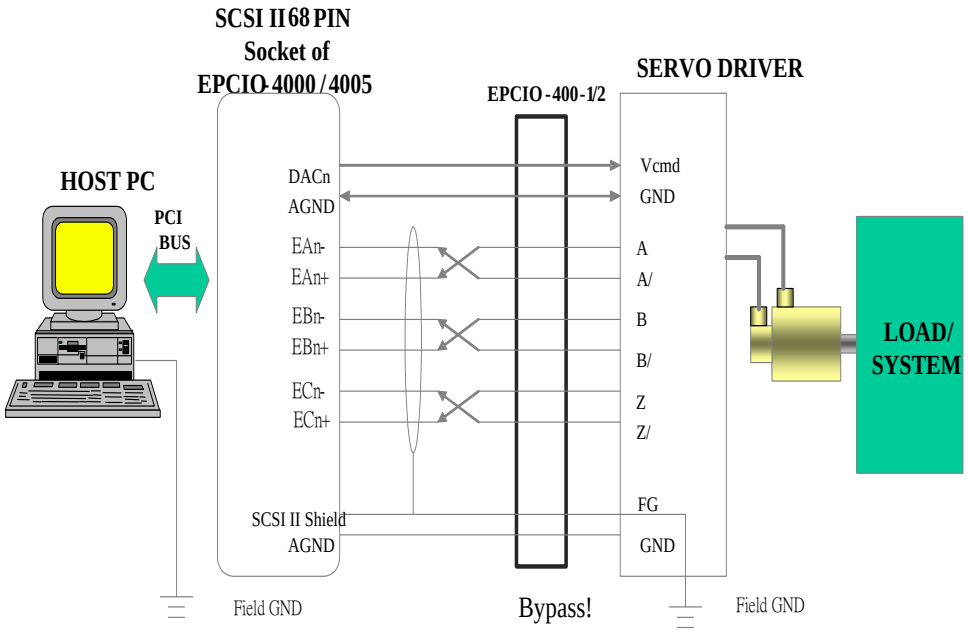


Fig. 3-6

3.3.2 Four-axis Synchronous/Asynchronous Pulse Output Control

As Fig. 3-7, the connection diagram of the pulse-command servo motor or the stepper motor for the EPCIO-4000/4005.

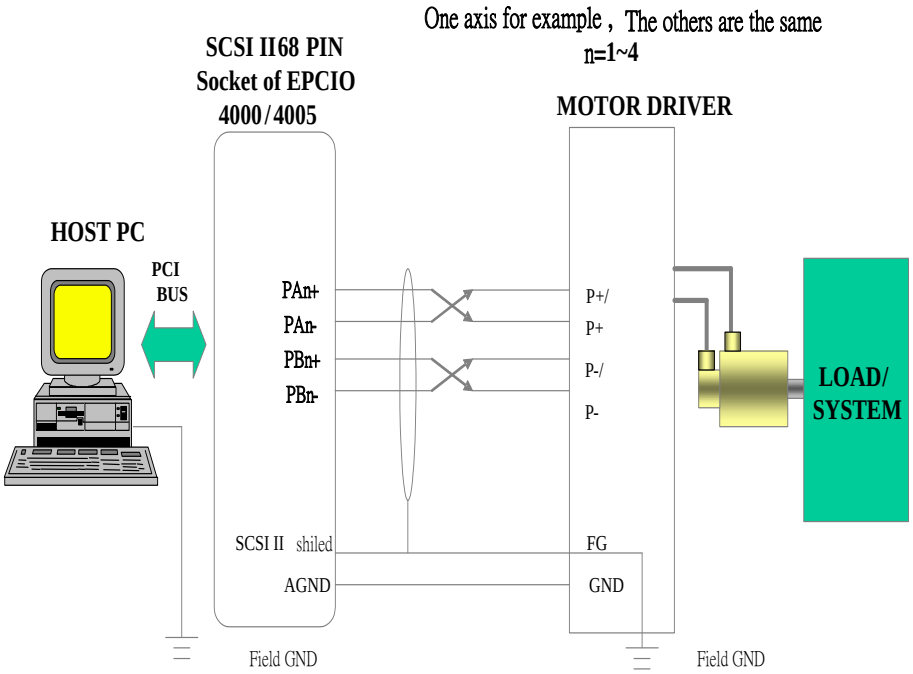


Fig. 3-7

- PAn+, PAn-, PBn+ and PBn- represent the pulse output signals. They should be connected to P+ and P+/, P- and P-/ of the nth MOTOR DRIVER as shown in the diagram. (Please refer to the manul of motor driver)
- The suggested above four lines use twisted-pair transmission wires to reduce the common-mode noise. As Fig.3-7 shows, use the shielded cable to isolate them from the peripheral device in order to reduce the interference.
- Connect one of the shielded cables to the external housing of the SCSI II 68 PIN (connected to PC frame ground) and another side to the SERVO DRIVER FG (Field Ground). Then make sure the Frame ground of both the PC and the SERVO DRIVER are connected together to the same GND.
- Importantt—A ground wire must be connected between the GND of the SERVO DRIVER and AGND of the EPCIO-4000 (This is very important as it could result in element damage due to ground voltage difference).
- Functions of the wiring board (the EPCIO-400-1/2) in pulse output control are bypass signals. The issues of Field GND, AGND, GND, twisted-pair cable, shielded cable please refer to 3.3.1.

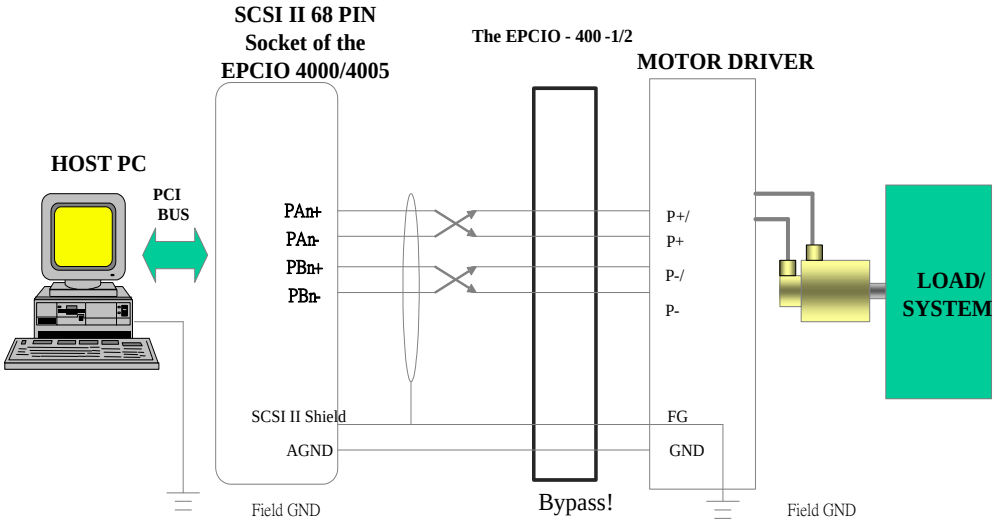

SCSI II 68 PIN
Socket of the
EPCIO 4000/4005
The EPCIO - 400 -1/2
MOTOR DRIVER
HOST PC
PCI
BUS
PAn+
PAn-
PBn+
PBn-
P+/
P+
P-/
P-
LOAD/
SYSTEM
SCSI II Shield
AGND
FG
GND
Field GND
Bypass!
Field GND

Fig. 3-8

3.3.3 Local I/O Wiring

3.3.3.1 Input wiring diagram

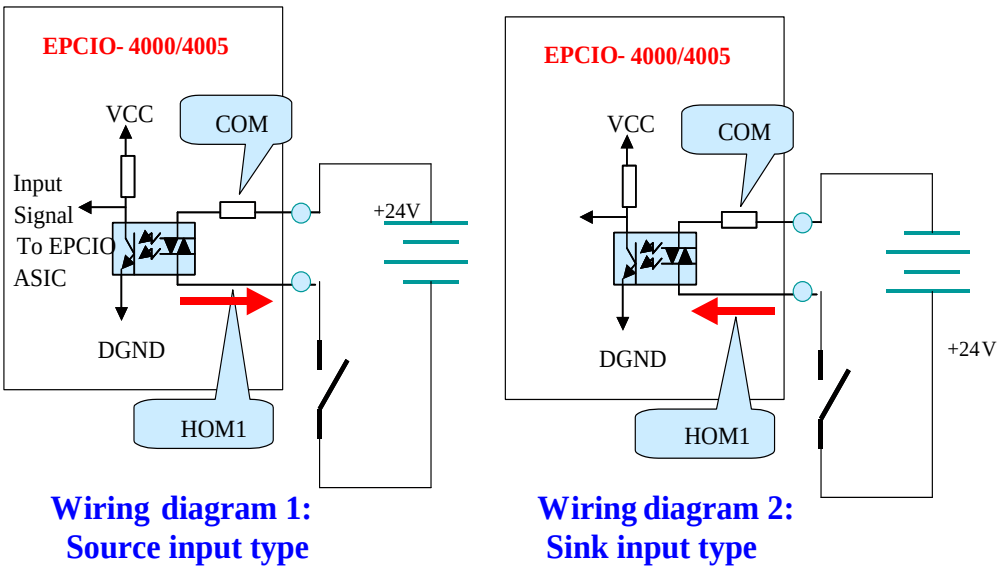


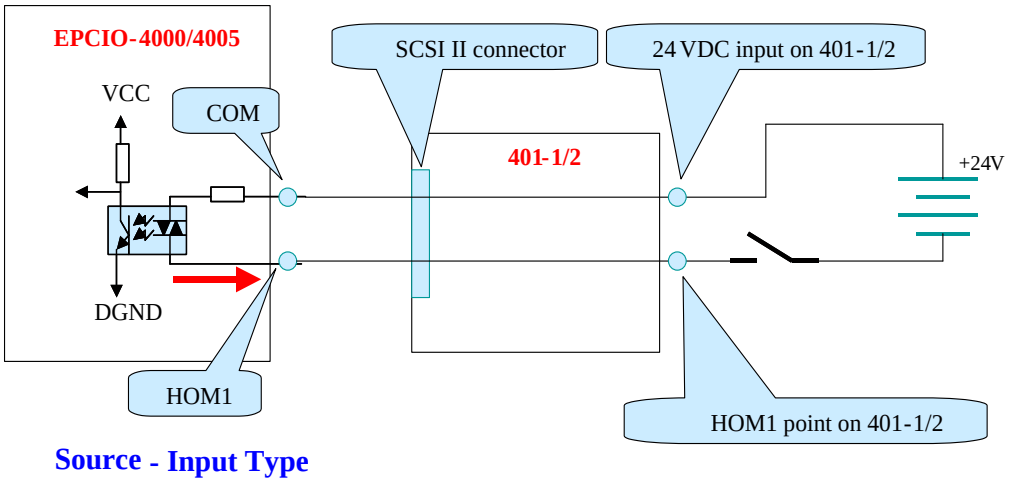
Fig. 3-9

- An example of HOME (HOM1) input type is shown in Fig. 3-9.
- Types of input wiring: Source-input type and sink-input type
- When the switch is closed, the internal EPCIO reading value is 0.
- When the switch is open, the internal EPCIO reading value is 1.
- The system must provide +24VDC power.

◆ **Caution**: Bouncing state

At the instant when the mechanical switch in Fig. 3-9 is turned from open to close, the switch has a bouncing phenomenon. At this time the reading value of the EPCIO will oscillate between 0 and 1. When at the end of bouncing, the switch conducts and the reading value becomes 0. On the other hand, at the instant when the mechanical switch is turned from closed to open, there is only a very little of the bouncing phenomenon.

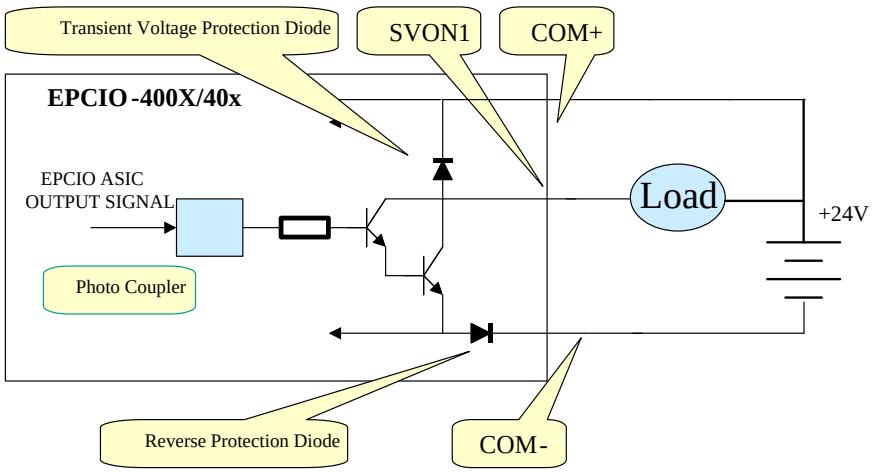
- With the wiring board (EPCIO-401-1/2), the EPCIO-4000/4005 inputs can only configured as the source-input type.



Source - Input Type

Fig. 3-10

3.3.3.2 Output Wiring Diagram



Sink-Output Type

Fig. 3-11

- An example of the output point SVON1 for the sink output is shown in Fig. 3-11.
- With the wiring board (EPCIO-401-1/2), the EPCIO-4000/4005 outputs can only be configured as the sink-output type.
- When the output signal is 0, the transistor (Darlington stage) is conducted and the load is power ON.
- HAZARD: Maximum load of each connection is 60 mA. Overload will cause transistor burn down. It prohibits that the EPCIO-400X from being directly connected to the 24V power supply output point in no-load conditions.
- The hardware already has a diode to avoid the voltage instantly pass. When load is RELAY, it does not require connecting to the other diode to receive the impulse noise.

- Functions of the wiring board (EPCIO-401-1/2) in output control are bypass signals.

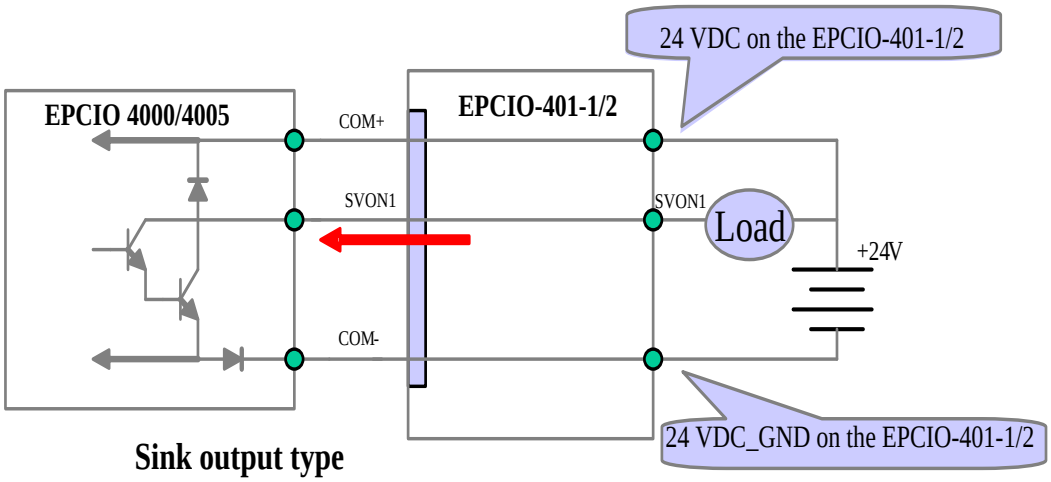


Fig. 3-12

3.3.3.3 EPCIO-4000/4005 local I/O wiring - An example for motion control

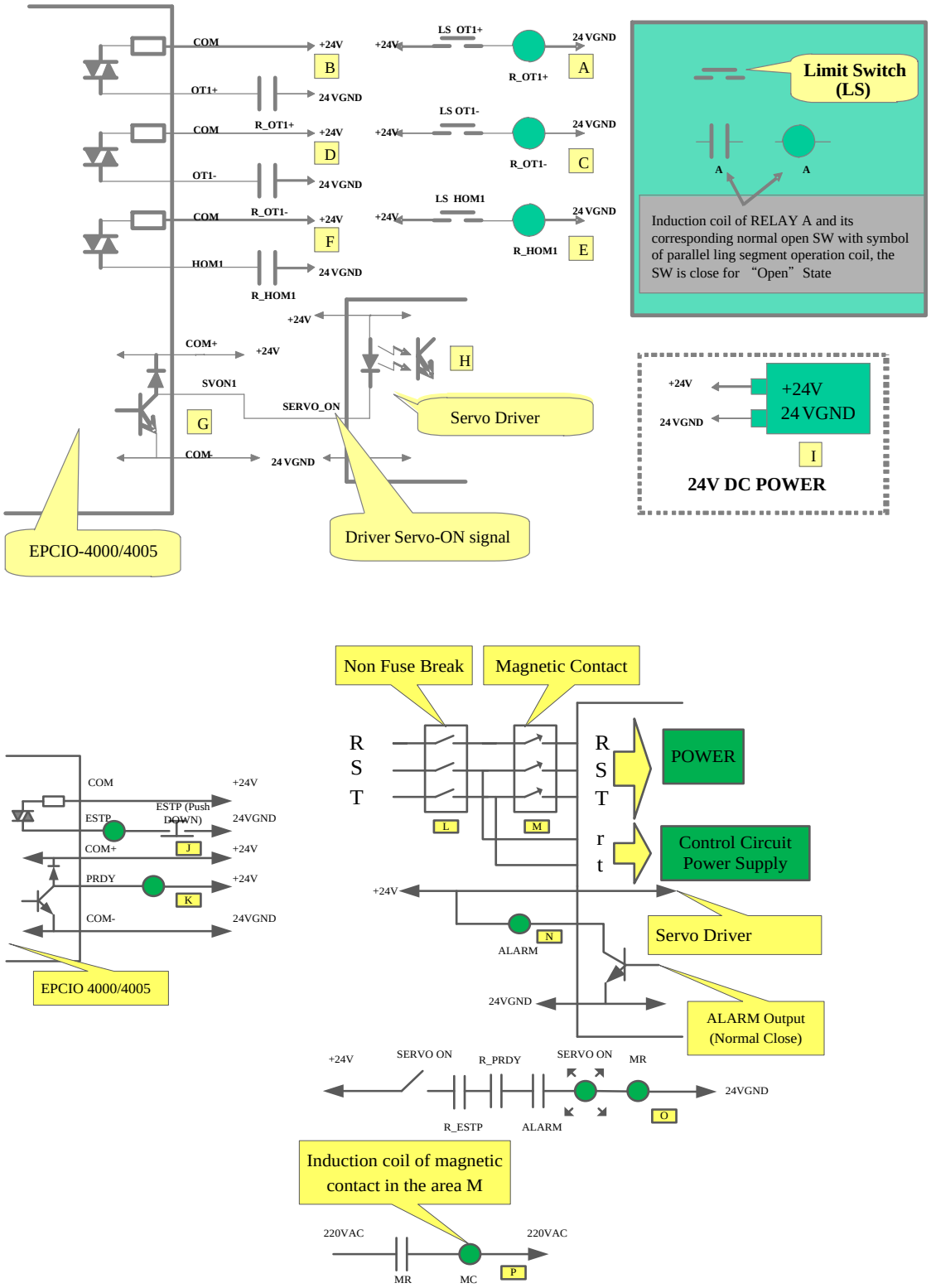


Fig. 3-13

Description:

The above circuit shown as Fig. 3-13 is an example of motion axis and its related I/O.

1. Please refer to A zone. When the first axis moves through the limit switch (LS OT1+), the RELAY (indicated as R_OT1+) will be activated. At this time, the NORMAL OPEN switch (R_OT1+) (in zone B) will close and enable the current to flow into the OT1+ point on SCSI II 100PIN connector. Now, the reading value of OT1+ on the EPCIO-4000/4005 will change from 1 to 0.
2. As above in Number 1, zone C and zone D are for the negative travel limit switch (LS OT1-).
3. As above in Number 1, zone E and zone F are the home limit switch (LS HOM1).
4. G and H zones: When the SVON1 signal of the first axis is changed from 1 to 0, the open collector output stage is connected, allowing current to flow through it and enable the driver to Servo-ON. (For the definition of Servo-ON, please refer to SERVO DRIVER manual).
5. Zone I is the 24V DC power for wiring. Note: if two or more 24V power sources are required, their grounds must be tied together. In addition, the suitable conducting wire gauge should be selected to avoid excessive voltage drop resulting in error action.
6. Zone J: Under normal conditions, the emergency stop switch should be in the CLOSED state. It forms a current loop in the ESTP circuit. At this time, the reading value of ESTP is 0 and also that the RELAY (R_ESTP) is activated. When E_STOP switch is pressed down, the input current loop cuts off. The reading value of E_STOP becomes 1 and the EPCIO-4000/4005 will disable the pulses output and make the outputs of DAC become 0V (slightly adjust the variable resistor if it is not 0).
7. Caution: The E_STOP function is disabled when JP5 (E_STOP) is shorted. (Default: JP5 short). For the emergency stop function to work properly, it is necessary to remove the jumper on JP5 (E_STOP).
8. Zone K: User can use PRDY signal to communicate to a peripheral circuit or system that the MCCL system is ready. After the EPCIO-4000/4005 is initialized PRDY should be changed from 1 to 0.
9. Zone L is the main power switch.
10. Zone M: Controllable electromagnetic contactor. The control activation coil is labeled MC.
11. Zone N: When SERVO DRIVER is abnormal, the transistor output stage will turn to off. The ALARM loop will be open.
12. Zone O: In zone J, the linked switch R_STOP is closed, the ESTP switch is not pressed down. In zone K, the linked switch R_PRDY is closed, the system outputs a POSITION_ READY signal. In zone N, if the driver operates normally, the linked switch ALARM will still close. If the above conditions are all on hold and the servo-on switch is closed, the current will flow into the MR coil. It makes the linked switch MR in zone P closed and then control coil MC of magnetic contact M is activated. Finally the magnetic contact M is closed and the power to the driver is linked.

3.3.4 Remote I/O Wiring

The EPCIO-4005 has only one RIO.

3.3.4.1 Connection diagram

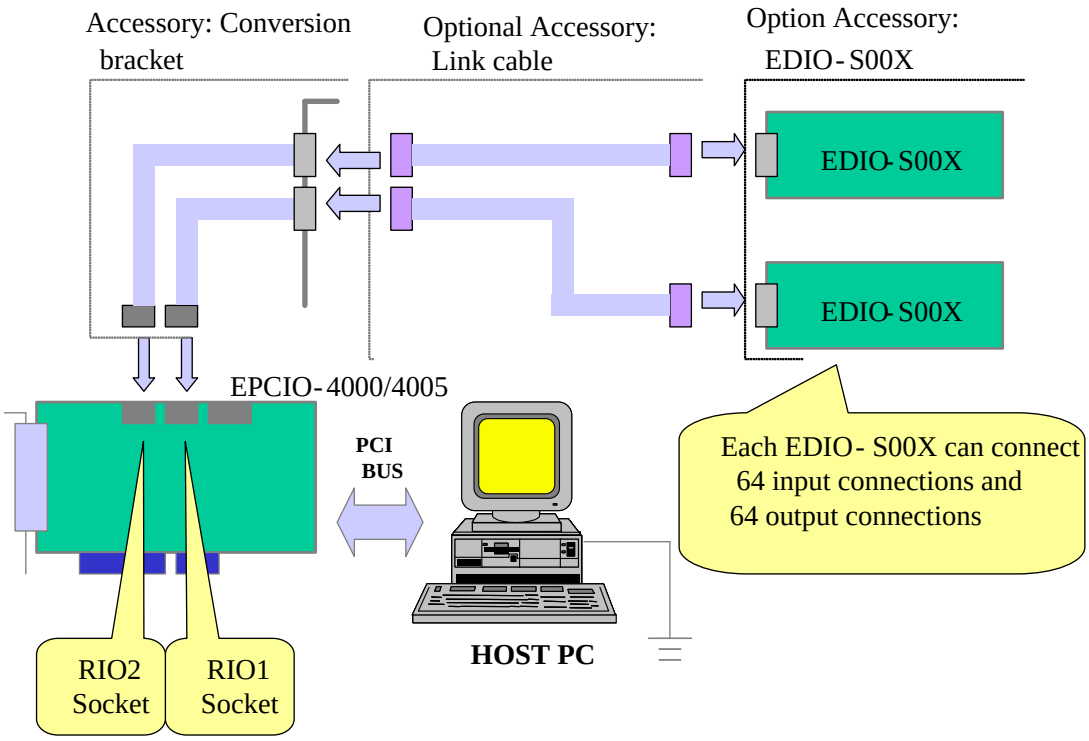
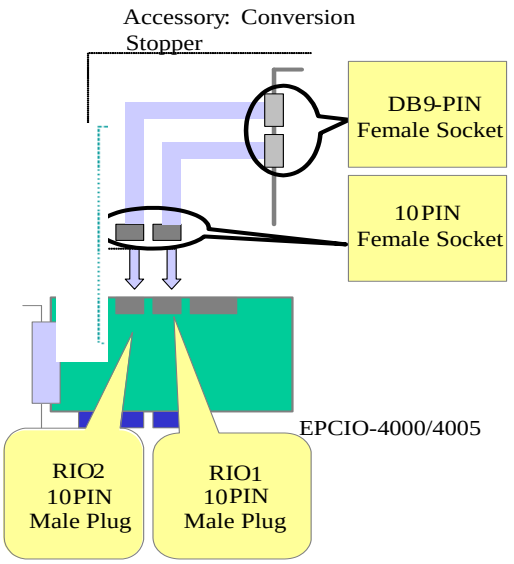
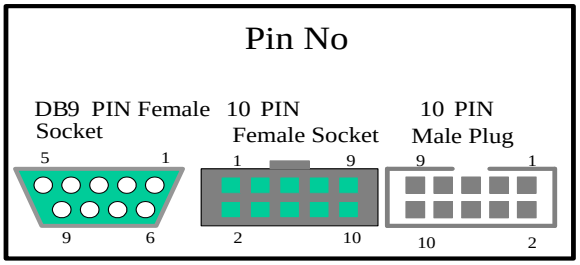


Fig. 3-14

3.3.4.2 Explanation on connector bracket





◆There are two rows of flat cable on the bracket. One end of the flat cable is 10PIN female plug to be inserted on the RIO1 and RIO2 sockets on EPCIO-4000. The other end is DB9PIN female socket for connecting to external EDIO-S00X module.
◆Internal connection of flat cable is as show below (refer to above pin definitions).

10PIN female plug DB9PIN female plug
1← →1
2← →2
3← →3
4← →4
5← →5
6← →6
7← →7
8← →8
9← →9
10 No connection

Fig. 3-15

3.3.4.3 Explanation for EDIO-S00X Remote Control I/O Card

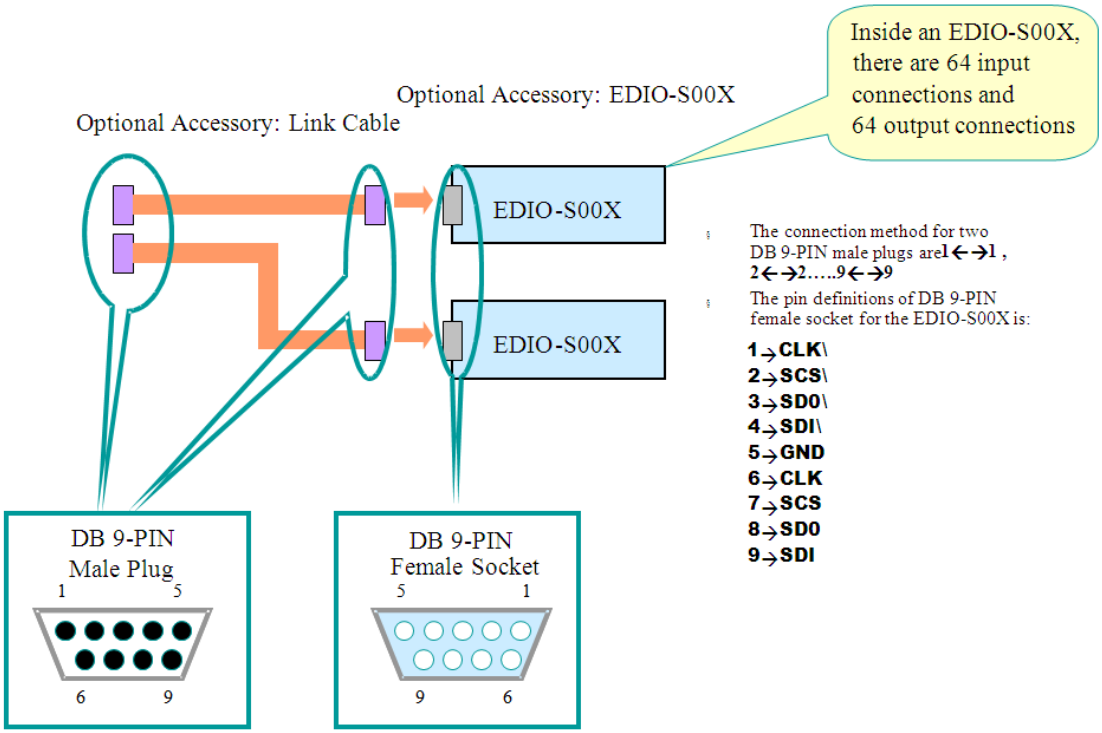


Fig. 3-16

3.3.5 ADC Wiring

The ADC is optional for EPCIO-4000; the EPCIO-4005 does not support.

3.3.5.1

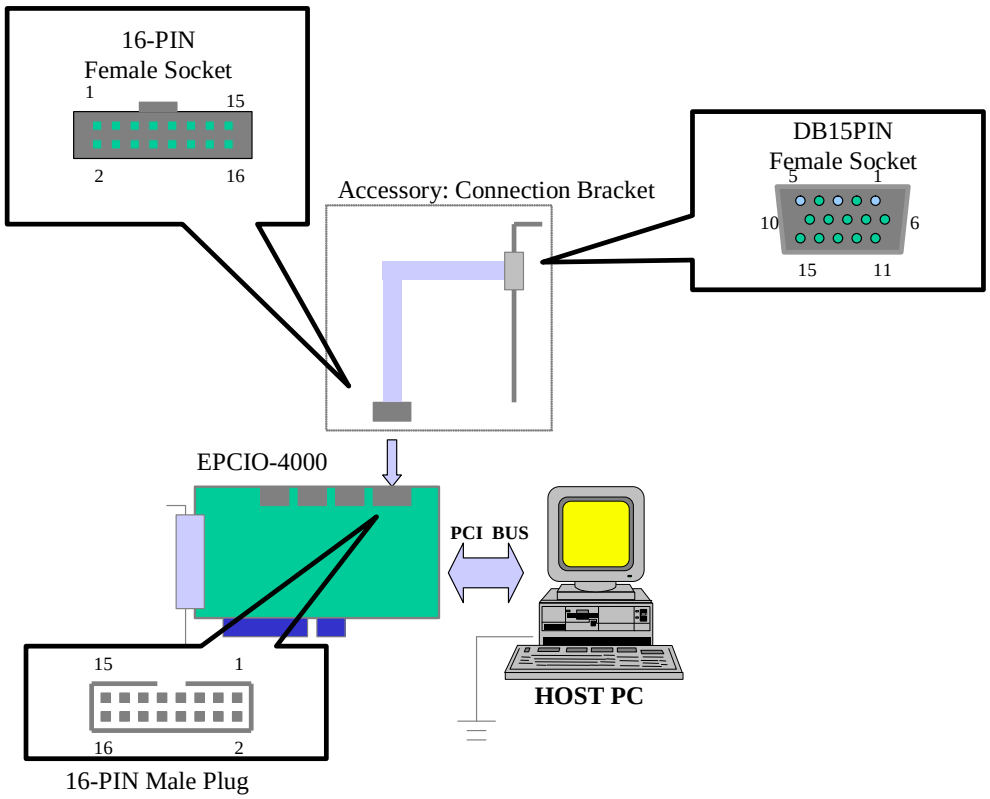


Fig. 3-17

- There is a flat cable on the bracket. One terminal is a 16-PIN Female Socket for the connection to the ADC socket of the EPCIO -4000. The other terminal is connected to the DB15-PIN Female Socket for the connection to the external signal desired to be measure.
- Internal connection of the flat cable

16-PIN Female Socket← →DB15-PIN Female Socket

1← →1

2← →2

3← →3

…

14← →14

15← →15

16← →No Connection

3.3.5.2 DB 15-PIN male plug pin definition

Pin 01: ADC+0	Pin 06: ADC+1	Pin 11: ADC+2
Pin 02: ADC-0	Pin 07: ADC-1	Pin 12: ADC-2
Pin 03: ADC+4	Pin 08: ADC+5	Pin 13: ADC+6
Pin 04: ADC-4	Pin 09: ADC-5	Pin 14: ADC-6
Pin 05: GND	Pin 10: GND	Pin 15: GND

3.3.5.3 Connection Description

A. Refer to the following table and select the suitable wiring mode:

DUT Voltage Range (Note1)	Description	SW setup	JP4	Wiring Mode
0~10VDC	DUTs have the same ground	Unipolar	Unipolar (Note 2)	Fig. 3-18 (Note 4)
0~10VDC	DUTs don't have the same ground	Unipolar	Unipolar	Fig. 3-19 (Note 4)
-5~5VDC	DUTs have the same ground	Bipolar	Bipolar(Note 3)	Fig. 3-18
-5~5VDC	DUTs don't have the same ground	Bipolar	Bipolar	Fig. 3-19

Note 1: DUT (Device Under Test) Voltage Range means that the voltage difference of DUT (see Fig. 3-18, Fig. 3-19) from positive terminal to negative terminal. For example, V_ADC6+ minuses V_ADC6-.

Note 2: Unipolar mode (shorting the UNI and COM of JP4).

Note 3: Bipolar mode (shorting the BIP and COM of JP4).

Note 4: substrace AGND (pin 5, pin 10, pin 15) form the voltage of 1~4 pin，6~9 pin, 11~14 pin of 15 pins plug in Fig. 3-18, Fig. 3-19 must less than 15 VDC.

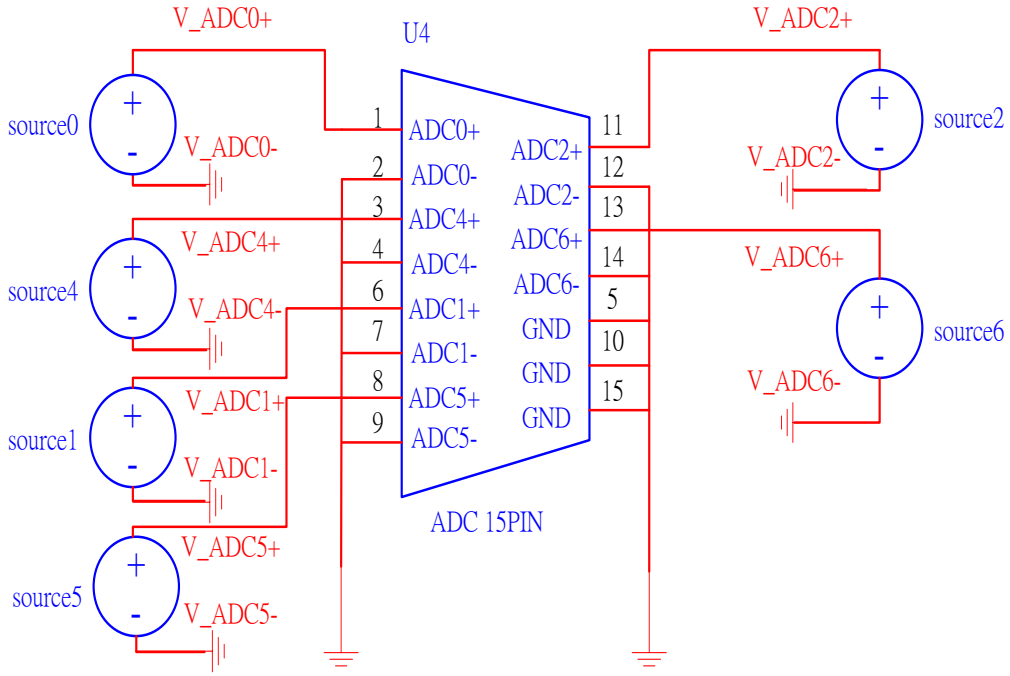


Fig. 3-18 Single-ended input

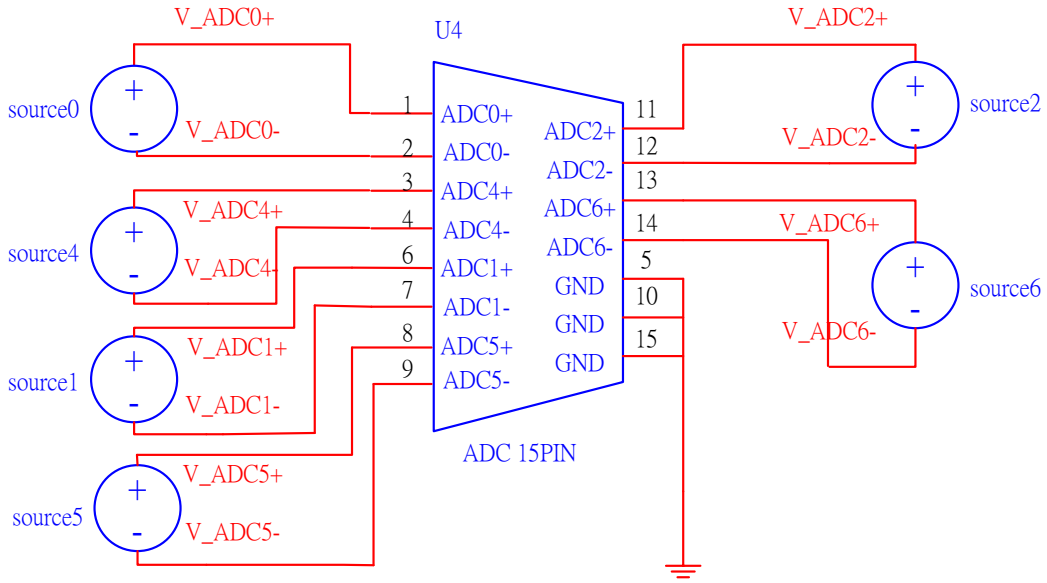


Fig. 3-19 Differential input

3.3.6 Wiring and Description of Encoder Input (MPG)

3.3.6.1 Please refer to below diagram:

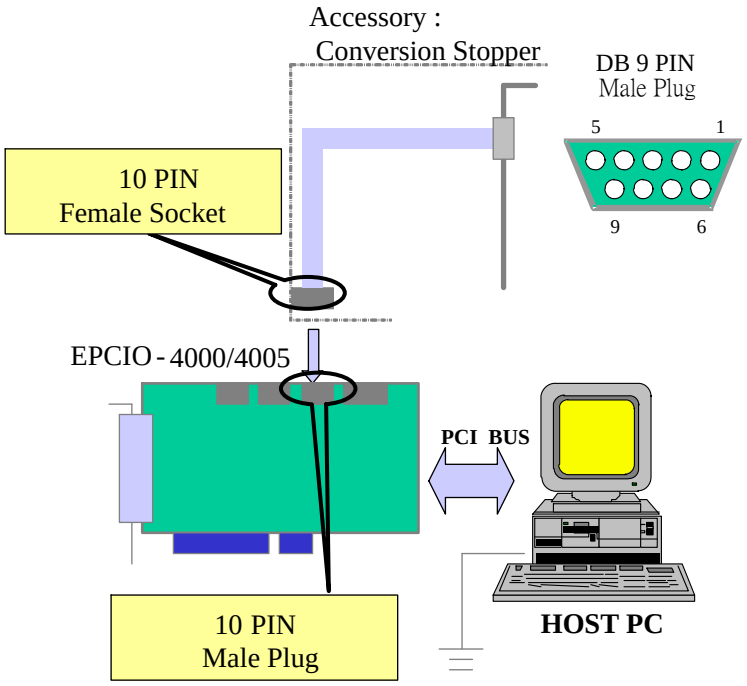


Fig. 3-20

There is a flat cable on bracket. One side of the flat cable is 10 PIN Female Socket for connection to the ENC_IN socket on the EPCIO-4000/4005. The other side is DB 9PIN Male Plug for connecting to external input encoder signal.

- The internal connection of flat cable is as shown below (refer to above pin definition).

10 PIN Female Plug←→DB 9PIN Male Plug

1←→1

2←→2

3←→3

. . . .

8←→8

9←→9

10←→NC

3.3.6.2 DB 9 pin Female Socket Pin Description

Pin 01: A	Pin 06: A/
Pin 02: B	Pin 07: B/
Pin 03: C	Pin 08: C/
Pin 04: GND	Pin 09: V-out (+5V or +12V)
Pin 05: GND	

Note：Pin 09: V-out can be selected +5V or +12V by JP1.

3.3.6.3 Wiring method

Refer to Fig 3-21 and Fig 3-22. MPG wiring method has two type “Differential” and “Single End”. ”Single End” has two wiring mode “Open Collector” and “Direct Drive”. If user used MPG with Single End method, user would check which wiring mode had been used.

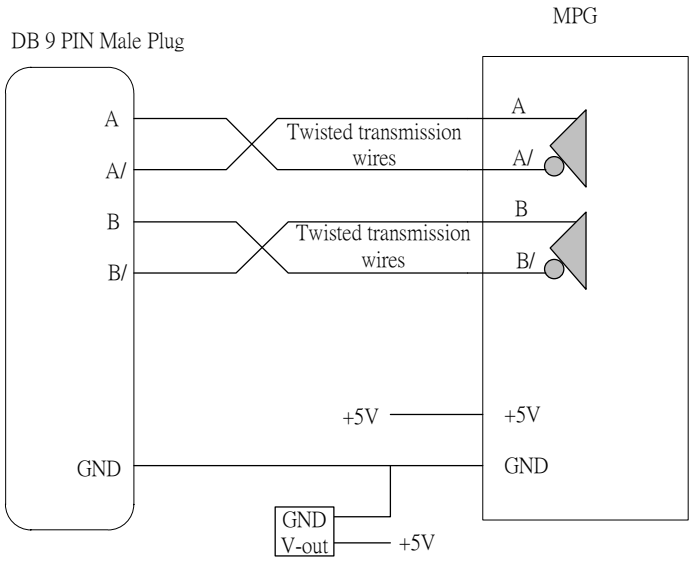
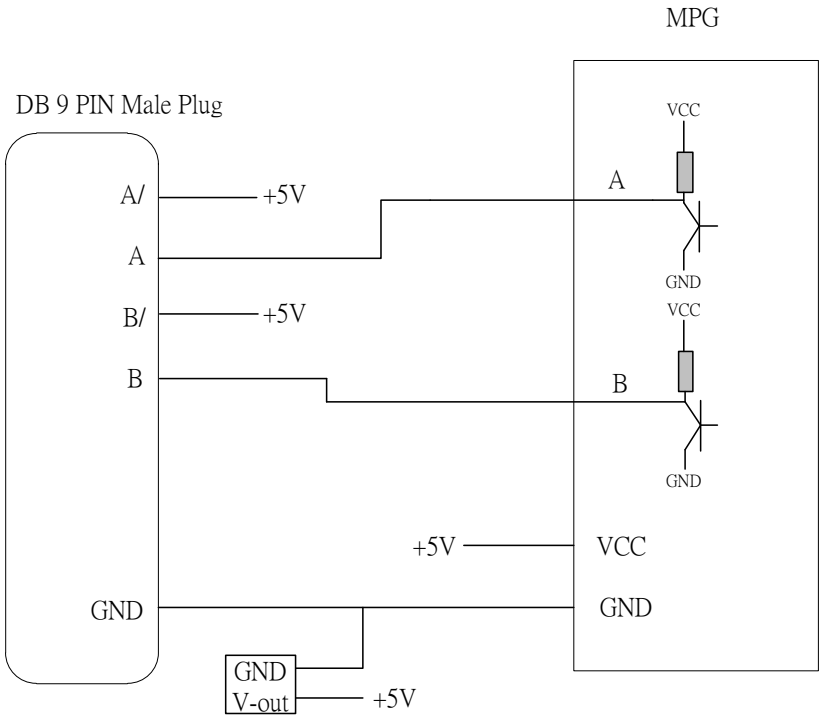
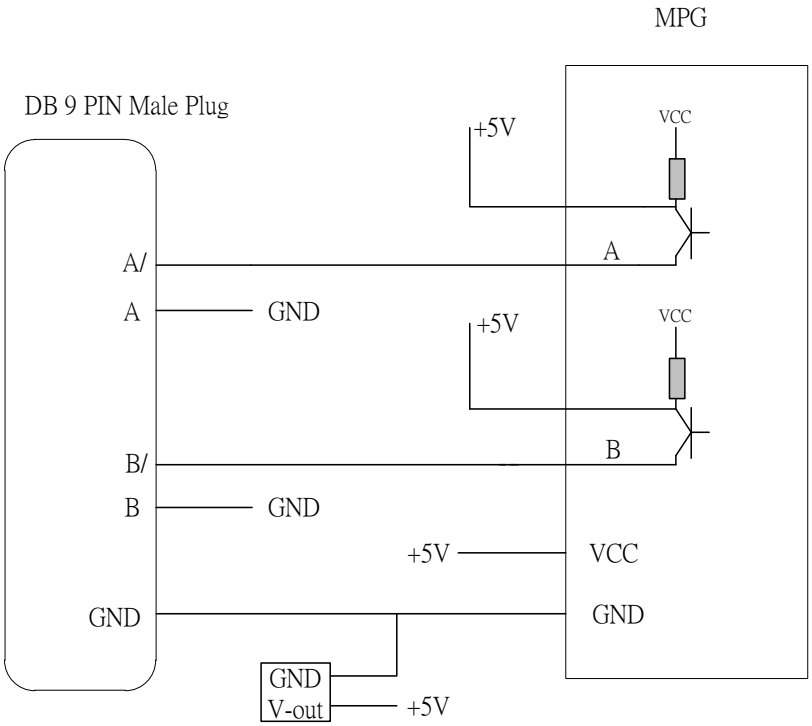


Fig. 3-21 Differential Type



Wiring mode1：Open Collector



Wiring mode2：Direct Drive

Fig. 3-22 Single End Type