

# **ENCODER3 3-Axis Encoder Card**

## **1 Hardware setup**

1.1 Address selection

1.2 Registers of ENCODER3 card

1.3 Connection

## **2 Software-Library**

2.1 Functions

2.2 DOS Driver (C, C++)

2.3 Windows 95 Driver

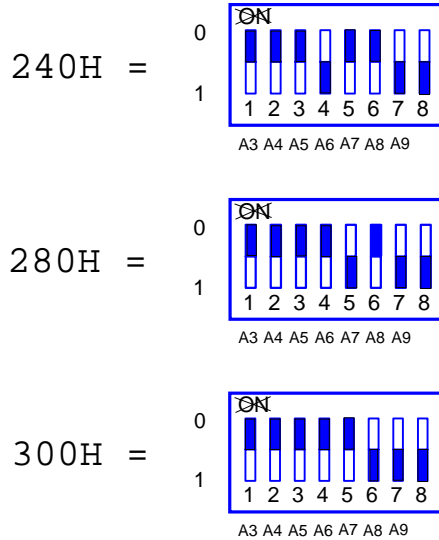
2.4 Windows NT Driver

## **3 Example**

# 1 Hardware setup

## 1.1 Address selection (encoder\_base)

The address is determined by A3~A9, there exist a dip switch on ENCODER3 card for address selection. The address can be select as following examples. The default selected address is 0x240h.



Figure(1) Address selection

## 1.2 Registers of ENCODER3 card

There are 7 registers which resided in selected address (encoder\_base) on ENCODER3 card. It includes 3 control register, 3 counter value register, 1 index register.

- (1) CTRL1 register (encoder\_base + 0) (write only)
- CTRL2 register (encoder\_base + 1) (write only)
- CTRL3 register (encoder\_base + 2) (write only)

CTRL1, CTRL2, CTRL3 is the control register for axis1, axis2 and axis3, respectively. The register format as following.

MSB 7	6	5	4	3	2	1	0 LSB
xx	xx	S1	S0	X2	/SEL	/INH	/RST

**/RST** : reset counter to zero

**/INH** : inhibit the activity of counter. This bit must be set 0 before reading the counter value to inhibit the activity of counter.

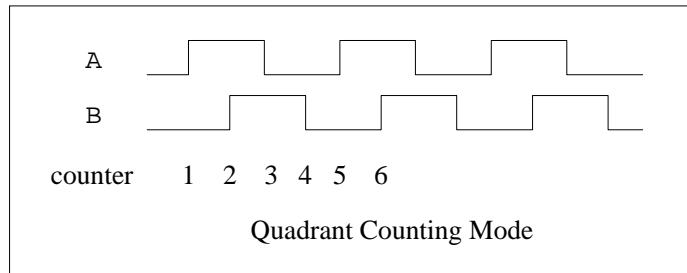
**/SEL** : to select low byte or high byte for reading the counter value.

0 : low byte

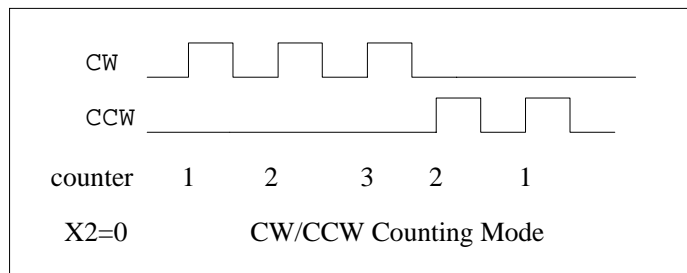
1 : high byte

**S1, S0** : to select counting mode

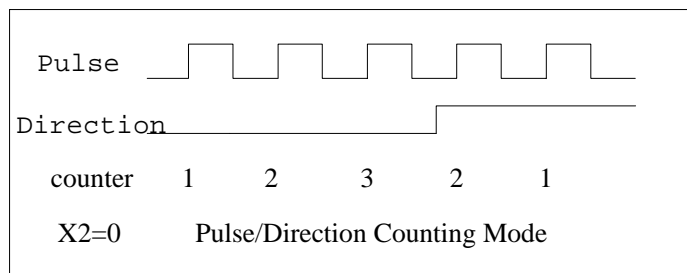
**00** : quadrant counting mode



**01** : CW/CCW counting mode



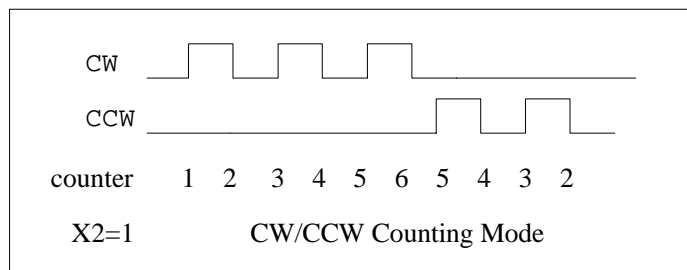
**10** : Pulse/Direction counting mode

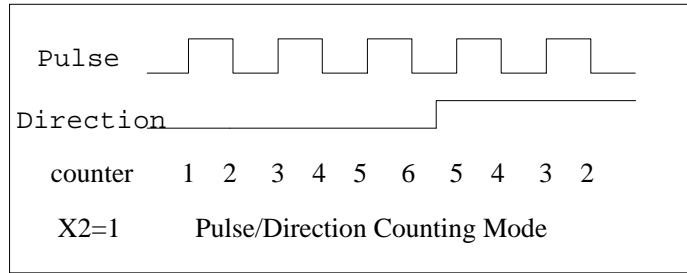


**X2** : set 2 times mode. This mode only for CW/CCW counting mode and Pulse/Direction mode.

0 : x1

1 : x2





- (2) VALUE1 register (encoder\_base + 0) (read only)  
 VALUE2 register (encoder\_base + 1) (read only)  
 VALUE3 register (encoder\_base + 2) (read only)

The counter value can be read from VALUE1, VALUE2, VALUE3 register for axis1, axis2, axis3, respectively. It can be read low byte when /SEL=0, and high byte when /SEL=1.

- (3) INDEX register (encoder\_base + 4) (read only)

MSB 7	6	5	4	3	2	1	0 LSB
xx	xx	xx	xx	xx	/C3	/C2	/C1

- /C1 : indicate the index of axis1.  
 /C2 : indicate the index of axis2.  
 /C3 : indicate the index of axis3.

### EAMPLES:

example:

```
//----- RESET COUNTER-----
encoder_ctrl1 = 0x07; //initial control register value
//S1S0=00, x2=0
encoder_ctrl1 &= 0xFE;
outportb(encoder_base + 0 , encoder_ctrl1);
encoder_ctrl1 |= 0x01;
outportb(encoder_base + 0 , encoder_ctrl1);
//----- READ COUNTER -----
encoder_ctrl1 &= 0xF9; //1111 1001 low byte & inhibit
outportb(encoder_base + 0 , encoder_ctrl1);
value = inportb(encoder_base + 0); // read low byte of axis1
//-----
encoder_ctrl1 |= 0x04; //0000 0100 high byte
outportb(encoder_base + 0 , encoder_ctrl1);
value += inportb(encoder_base + 0)*256; //read high byte
//-----
encoder_ctrl1 |= 0x06; //0000 0110 reading over
outportb(encoder_base + 0 , encoder_ctrl1);
```

### 1.3 Connection

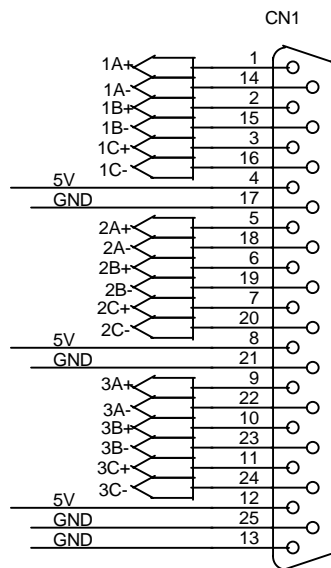


Figure (2) CN1 connector

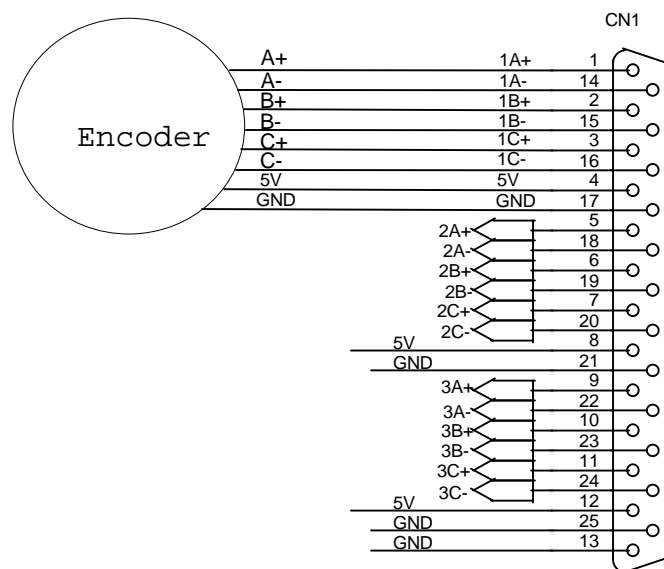


Figure (3) Connection between encoder and ENCODER3 card

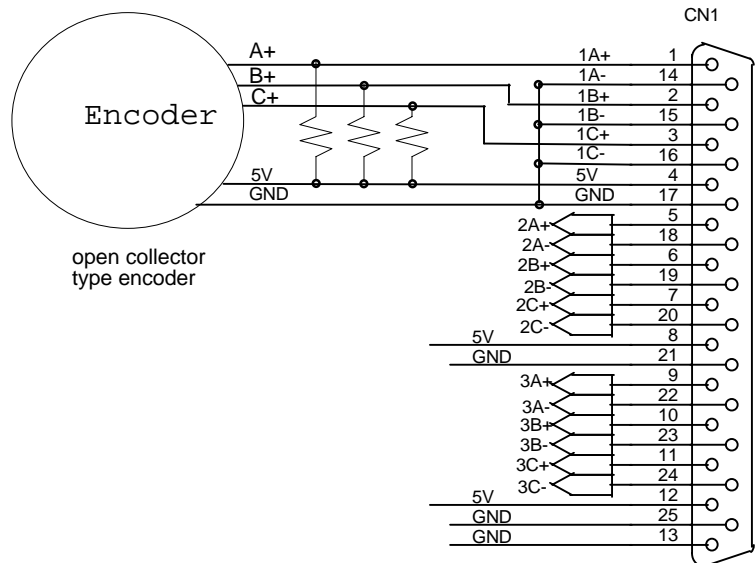


Figure (4) Connection between open collector type encoder and ENCODER3 card

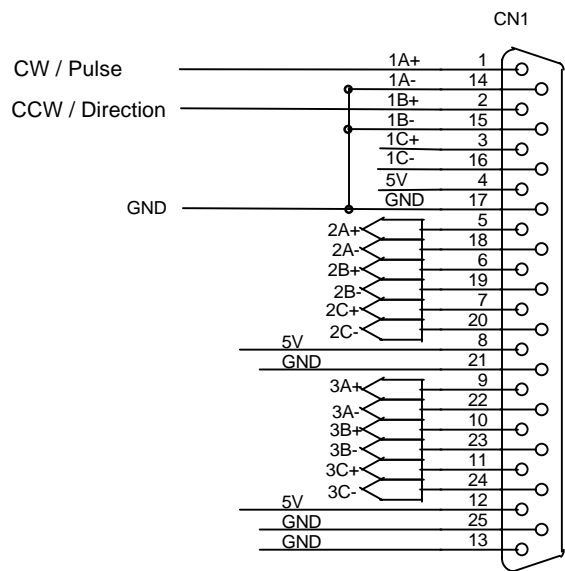


Figure (5) The connection for CW/CCW, Pulse/Direction counting mode

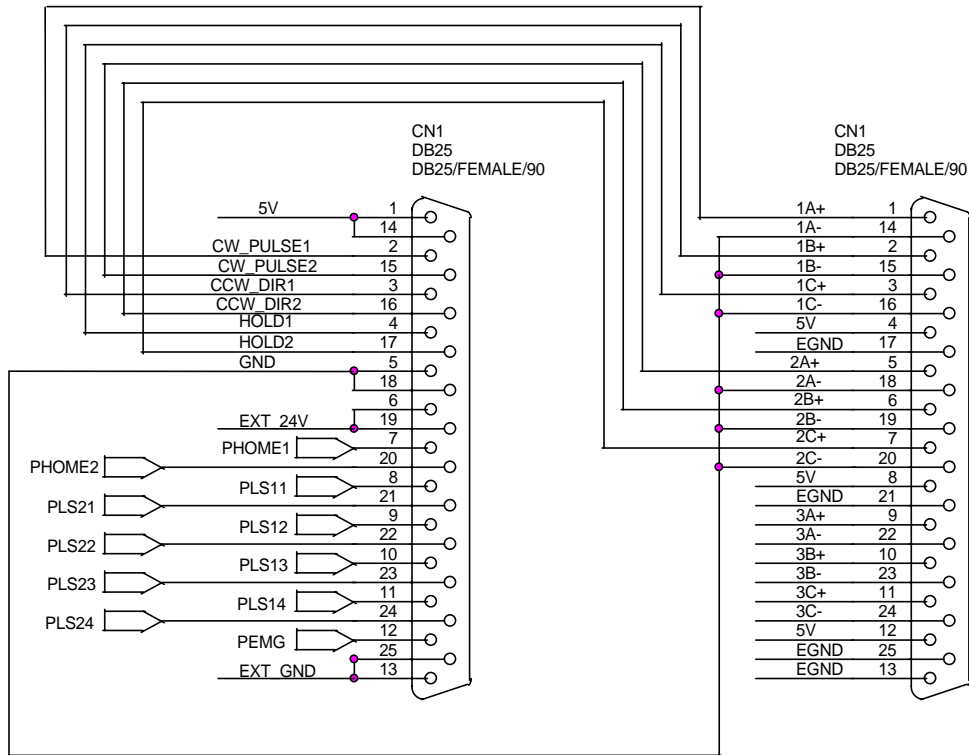


Figure (6) The connection between ENCODER3 card and MSTEP2 card.

## 2 Software - Library

directories

```
encoder _____ lib _____ c
                                     |_____ nt
                                     |_____ win32
```

### 2.1 Functions

#### Constants

##### axis

```
#define X_axis      1
```

```
#define Y_axis      2
```

```
#define Z_axis      3
```

##### times mode

```
#define ENC_X1      0x00
```

```
#define ENC_X2      0x08
```

##### counting mode

```
#define ENC_QUADRANT 0x00
```

```
#define ENC_CW_CCW  0x10
```

```
#define ENC_PULSE_DIR 0x20
```

(1) **ENC3\_INITIAL()** only used for Windows95 ,NT  
To load VxD driver.

(2) **ENC3\_END()** only used for Windows95 ,NT  
To release VxD driver.

(3) **ENC3\_SELECT(unsigned int address)**

To choose ENCODER3 card address. If there are many cards on your system, this command provide a flexibility to distinguish those cards. So, it must perform this command before to execute other command in multi-card system.

(4) **ENC3\_\_INIT\_CARD(unsigned char x\_mode,  
unsigned char y\_mode,  
unsigned char z\_mode)**

To set the control registers of each axis. The control registers has been explained in chapter 1.2. You can set the counting mode and times mode as your system required. There are some constants had been defined in header file. It is more comprehensive than numerical.



(5) **unsigned int ENC3\_GET\_ENCODER(unsigned char axis)**

This function will return the selected-axis counter value.

(6) **ENC3\_RESET\_ENCODER(unsigned char axis)**

To reset the selected-axis counter value.

(7) **unsigned char ENC3\_GET\_INDEX()**

To get the index value from INDEX register.

## 2.2 DOS Driver (C, C++)

Item	File
Header file	enc3.h
Library file	enc3.lib
Example file	enc3.prj enc3.exe main.cpp

### enc3.h

```

//*****
//   ENCODER header file
//
//                                           Chi-Mou Chao 9/3/98
//*****
//----- define parameter -----
#define X_axis      1
#define Y_axis      2
#define Z_axis      3
#define ENC_X1      0x00
#define ENC_X2      0x08
#define ENC_QUADRANT 0x00
#define ENC_CW_CCW  0x10
#define ENC_PULSE_DIR 0x20
//----- subroutines -----
extern void      ENC3_SELECT(unsigned int address);
extern void      ENC3_INIT_CARD(unsigned char x_mode,
                                unsigned char y_mode,
                                unsigned char z_mode);
extern unsigned int ENC3_GET_ENCODER(unsigned char axis);
extern void      ENC3_RESET_ENCODER(unsigned char axis);
extern unsigned char ENC3_GET_INDEX();

```

### The Object

This card provide a powerful capability, one card one object. This is very useful for many card on your system. Therefore, all of ENCODER3 card's

function had been constructed as an object "TEncoder". The header file shows their relation.

Item	File
Header file	oenc3.h
Library file	oenc3.lib
Example file	oenc3.prj oenc3.exe omain.cpp

### **oenc3.h**

```

//*****
//  ENCODER header file
//
//                                                    Chi-Mou Chao 9/3/98
//*****
//----- define parameter -----
#define X_axis      1
#define Y_axis      2
#define Z_axis      3
#define ENC_X1      0x00
#define ENC_X2      0x08
#define ENC_QUADRANT 0x00
#define ENC_CW_CCW  0x10
#define ENC_PULSE_DIR 0x20
//-----
class TEncoder
{
public:
    void      ENC3_SELECT(unsigned int address);
    void      ENC3_INIT_CARD(unsigned char x_mode,
                             unsigned char y_mode,
                             unsigned char z_mode);
    unsigned int  ENC3_GET_ENCODER(unsigned char axis);
    void          ENC3_RESET_ENCODER(unsigned char axis);
    unsigned char ENC3_GET_INDEX();
private:
    unsigned int  encoder_base;
    unsigned char encoder_ctrl1;
    unsigned char encoder_ctrl2;
    unsigned char encoder_ctrl3;
};

```

## 2.3 Windows 95 Driver

Item	File
Header file	enc32.h
ImportLibrary file	enc32.lib bcenc32.lib (only for Borland C++ series)
Dynamic Link Library	enc32.dll
VxD file	vportd.vxd
Example file	project1.bpr(Borland C++ Builder) project1.cpp bbsetp.cpp

The files enc32.dll, vportd.vxd should be copied to the directory c:\windows\system.

## 2.4 Windows NT Driver

Item	File
Header file	enc32.h
Import Library file	enc32.lib bcenc32.lib (only for Borland C++ series)
Dynamic Link Library	enc32.dll
Driver	napwnt.ini napwnt.sys regini.exe
Example file	project1.bpr(Borland C++ Builder) project1.cpp bbsetp.cpp

### 3 Example

directories

```
encoder ___example ___ c
|___ nt
|___ win32
```

The project1.exe (source file included) is a example for ENCODER3 card and MSTEP2 card. It has windows95 and NT edition. The lower-right encoder sub-window shows the address(decimal), counter value and index value. When click the Update Parameters button, it will shows a dialog for selecting the counting mode and times mode.

