ISO-P32C32/P64/C64

User's Manual

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assume no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Tables of C1ontents

1. Introduction

The ISO-P32C32 consists of 32 channels of isolated D/I & 32 channels of isolated D/O. The ISO-P64 consists of 64 channels of isolated D/I. The ISO-C64 consists of 64 channels of isolated D/O. The D/I specifications of ISO-P32C32 & ISO-P64 are the same. The D/O specifications of ISO-P32C32 & ISO-C64 are the same.

1.1 Specifications

Isolated digital input

- Input voltage: 5V to 30V DC
- Input impedance: 3K
- Isolation voltage:
  Using internal power: 3000V
  Using external power: 3750V
- Response time: 30K Hz Max.

Isolated digital output

- Isolation voltage: 3750V
- Open collector output: 100 mA/30V per channel
- Response time: 4K Hz typical

I/O channels

	D/I channels	D/O channels
ISO-P32C32	32	32
ISO-P64	64	0
ISO-C64	0	64

Other specifications

- PC compatible ISA bus
- Four isolated I/O bank
- Operating Temperature: 0°C to 50°C
- Storage Temperature: -20°C to 70°C
- Humidity: 0 to 90% non-condensing
- Dimensions:
  ISO-P32C32: 163mm X 115mm
  ISO-P64: 163mm X 115mm
  ISO-C64: 163mm X 115mm
- Power Consumption:
  ISO-P32C32: +5V @ 600mA (typical)
  ISO-P64: +5V @ 400mA (typical)
  ISO-C64: +5V @ 500mA (typical)
- Input impedance:
  ISO-P32C32: 3K Ohm
  ISO-P64 : 3K Ohm

1.2 Order Description

- ISO-P32C32: ISA bus D/I of 32-bit, D/O of 32-bit.
- ISO-P64: ISA bus D/I of 64-bit.
- ISO-C64: ISA bus D/O of 64-bit.

1.3 Options

- DB-24P, DB-24PD: 24 channel isolated D/I board
- DB-24R, DB-24RD: 24 channel relay board
- DB-24PR, DB-24PRD: 24 channel power relay board
- DB-16P8R:16 channel isolated D/I and 8 channel relay output board
- DB-24POR: 24 channel Photo MOS output board
- DB-24SSR: 24 channel Solid State output board
- DB-24C: 24-channel open-collector output board
- ADP-37/PCI: extender, 50-pin OPTO-22 header to DB-37 for PCI Bus I/O boards
- ADP-50/PCI: extender, 50-pin OPTO-22 header to 50-pin header, for PCI Bus I/O boards

1.4 Product Check List

In addition to this manual, the package includes the following items:

- ISO-P32C32/P64/C64 card.
- One piece of company floppy diskette or CD-ROM.
- One piece of release note.

It's recommended to read the release note first. All important information will be given in. release notes as follows:

1. Where you can find the software driver & utility.
2. How to install software & utility.
3. Where is the diagnostic program.
4. FAQ.

Attention !

If any of this items is missing or damaged, contact the dealer from whom you purchased the product. Save the shipping materials and carton in case you want to ship or store the product in the future.

2. Hardware configuration

2.1 ISO-P32C32 Board Layout

The board layout of ISO-P32C32 is given as follows:

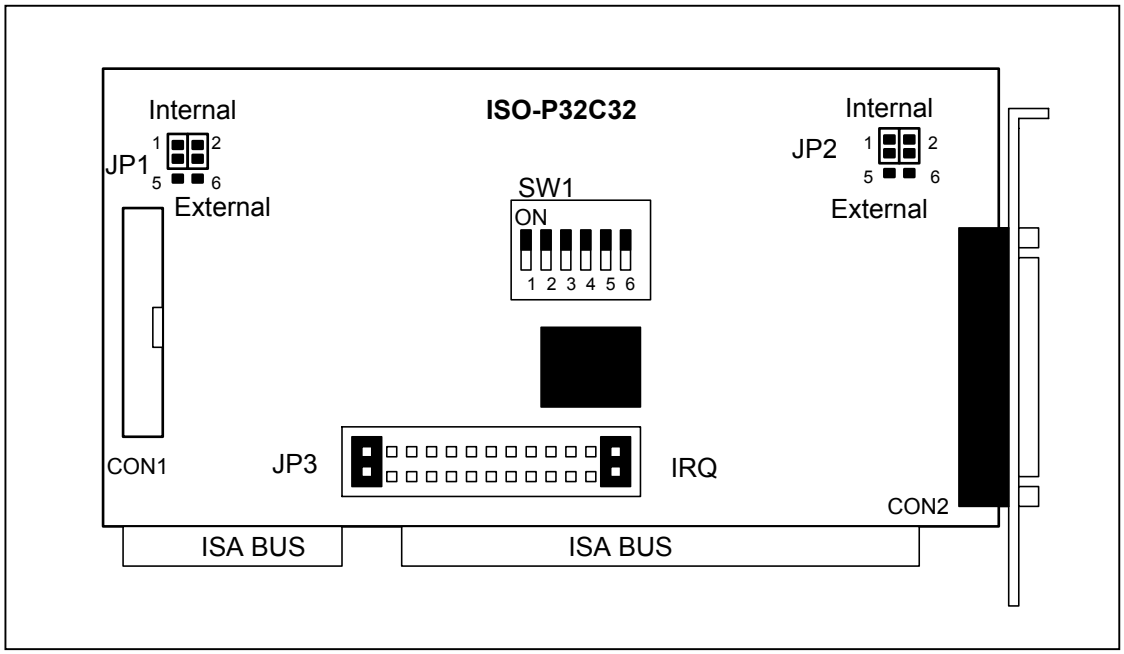


Figure 2-1. Board layout of ISO-P32C32

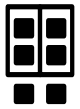
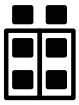
- CON1: 40-PIN connector for D/I/O channel 16-31
- CON2: 37-PIN D-sub connector for D/I/O channel 0~15

- JP1: CON1 select internal/external power
- JP2: CON2 select internal/external power
- JP3: Interrupt levels selection
  (ISO-P32C32/H provides IRQ9, 10, 11, 12, 14 and 15)
- SW1: I/O Address selection

- Isolation bank 1: DI_16 to DI_31, Power=CON1_18,Ground=CON1_19
- Isolation bank 2: DO_16 to DO_31, Power=CON1_37,Ground=CON1_1 & CON1_20
- Isolation bank 3: DI_0 to DI_15, Power=CON2_18,Ground=CON2_19
- Isolation bank 4: DO_0 to DO_15, Power=CON2_37,Ground=CON2_1 & CON2_20
- All these four banks are fully isolated from each other.

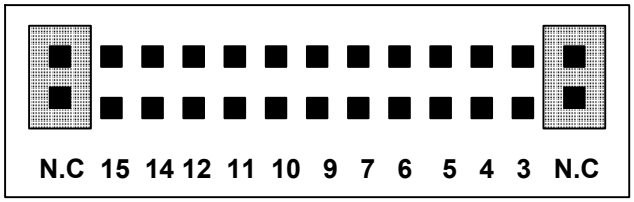
2.1.1 Jumper Setting

JP1/JP2 Jumper setting:

	**Internal**	**Default Setting**
	**External**	

- JP1: CON1 Internal power/External selection.
- JP2: CON2 Internal power/External selection.

JP3: IRQ selection



Default IRQ: N.C (Not use IRQ)

- **Select IRQ one of the 3/4/5/6/7 for D/I channel 16 of CON1.**
- **Select IRQ one of the 9/10/11/12/14/15 for D/I channel 0 of CON2.**

- **Usage:** Step 1: Select IRQ 3/4/5/6/7 for channel 16.
  Step 2: Select IRQ 9/10/11/12/14/15 for channel 0.
  Step 3: Input interrupt signals by way of channel 0 / 16 of CON2 / CON1.
  Step 4: Run P32Cint.EXE used interrupt function for ISO-P32C32 card.
  Step 5: Select I/O Base address of ISO-P32C32 card.
  Step 6: Input IRQ number (Refer to JP3 setting of P32C32 board).

2.1.2 IRQ Circuit diagram

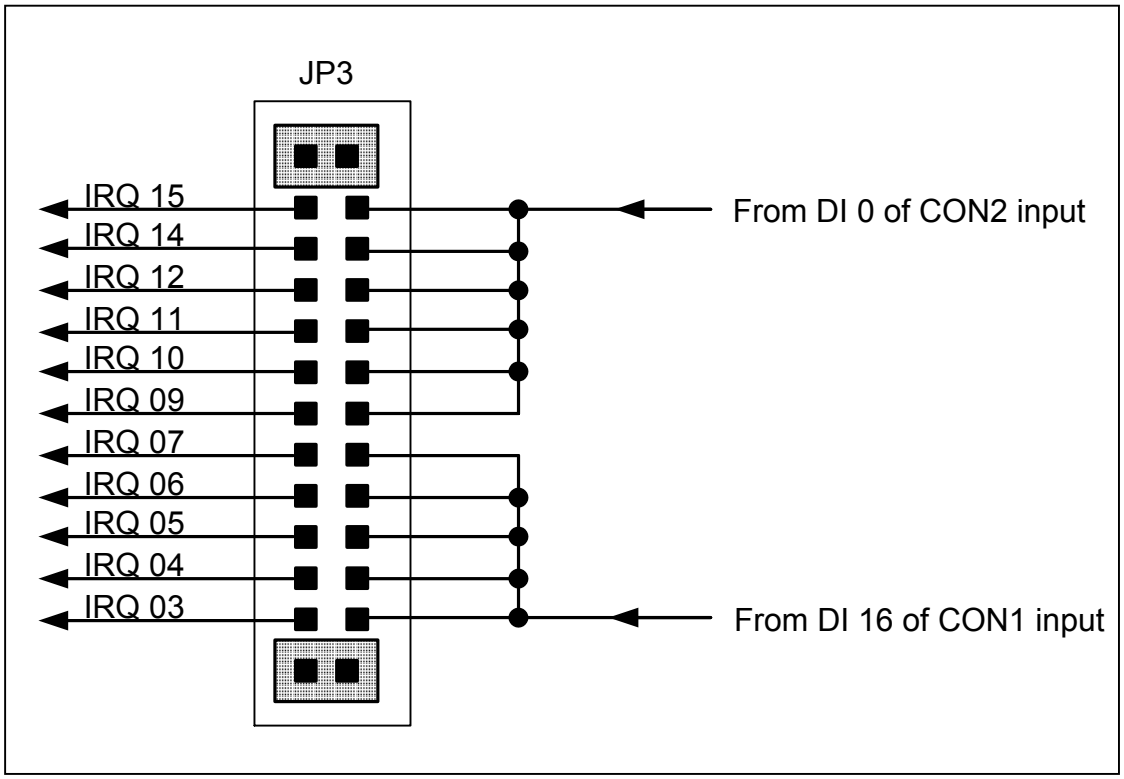


Figure 2-2. Select circuit diagram of IRQ for ISO-P32C32 board.

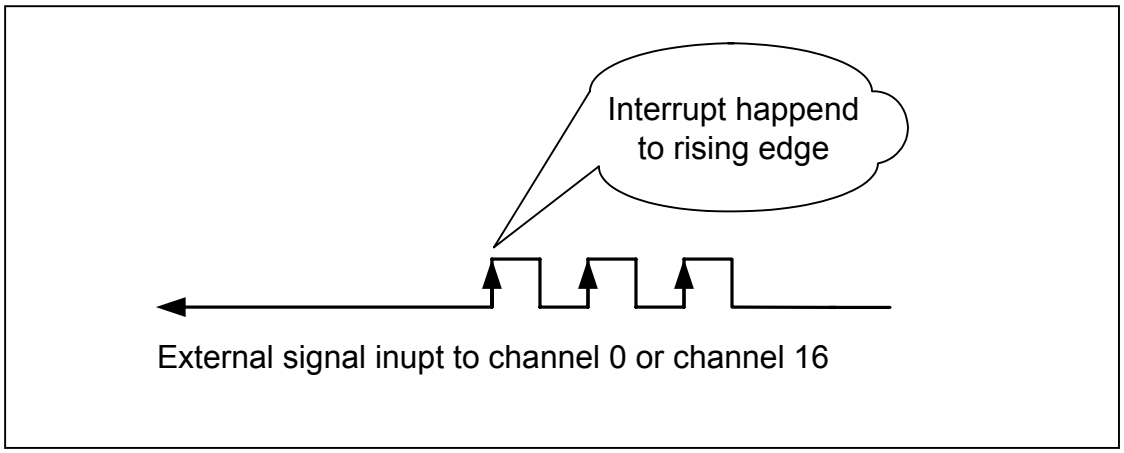
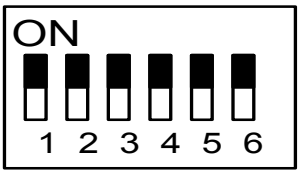


Figure 2-3. Interrupt happened to rising edge.

2.1.3 I/O Base Address Setting

The ISO-P32C32 occupies 8 consecutive in I/O address space. The base address is set by DIP switch SW1.

SW1: Base Address



Default Address: 0x200

Address Table:

Address	SW1_1	SW1_2	SW1_3	SW1_4	SW1_5	SW1_6
	A8	A7	A6	A5	A4	A3
*0x200	ON	ON	ON	ON	ON	ON
0x208	ON	ON	ON	ON	ON	OFF
0x210	ON	ON	ON	ON	OFF	ON
0x218	ON	ON	ON	ON	OFF	OFF
0x220	ON	ON	ON	OFF	ON	ON
0x228	ON	ON	ON	OFF	ON	OFF
:	:	:	:	:	:	:
0x2C0	ON	OFF	OFF	ON	ON	ON
0x2C8	ON	OFF	OFF	ON	ON	OFF
:	:	:	:	:	:	:
0x300	OFF	ON	ON	ON	ON	ON
0x308	OFF	ON	ON	ON	ON	OFF
:	:	:	:	:	:	:

- **Default Base Address: 0x200**

2.1.4 I/O Register Address

The ISO-P32C32 card occupies 8 consecutive PC I/O addresses. The following table lists the registers and their locations.

D/I: digital input data format.
D/O: digital output data format.

ISO-P32C32 I/O Register Address

Address	Read	Write	ISO-P32C32
Base+0	D/I Channel 0~7	D/O Channel 0~7	CON 2
Base+1	D/I Channel 8~15	D/O Channel 8~15	CON 2
Base+2	D/I Channel 16-23	D/O Channel 16~23	CON 1
Base+3	D/I Channel 14~31	D/O Channel 14~31	CON 1

ISO-P32C32/H I/O Resister Address

Address	Read	Write	ISO-P32C32/H
Base+0	D/I Channel 0~7	D/O Channel 0~7	CON2
Base+1	D/I Channel 8~15	D/O Channel 8~15	CON2

Read/Write Base+0

Bit	7	6	5	4	3	2	1	0
Channel	7	6	5	4	3	2	1	0

Read/Write Base+1

Bit	7	6	5	4	3	2	1	0
Channel	15	14	13	12	11	10	9	8

Read/Write Base+2

Bit	7	6	5	4	3	2	1	0
Channel	23	22	21	20	19	18	17	16

Read/Write Base+3

Bit	7	6	5	4	3	2	1	0
Channel	31	30	29	28	27	26	25	24

2.2 ISO-P64 Board Layout

The board layout of ISO-P64 is given as follows:

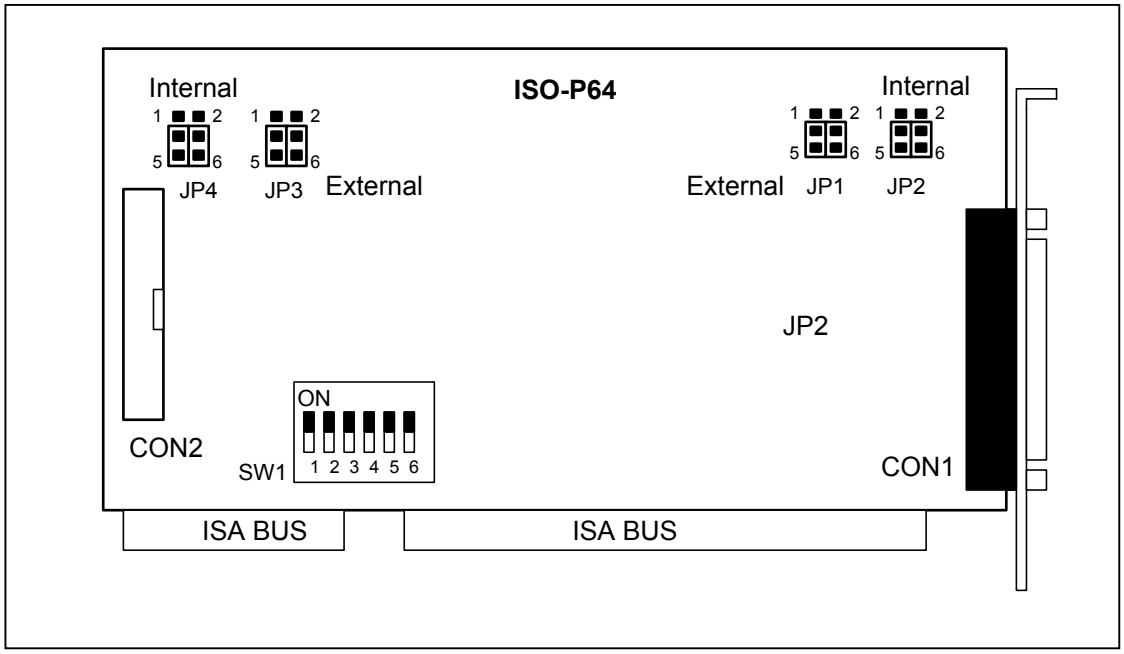


Figure 2-4. Board layout of ISO-P64

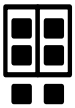
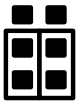
J1: select internal/external power for DI_0 to DI_15 (3000V isolation)
J2: select internal/external power for DI_16 to DI_31 (3000V isolation)
J3: select internal/external power for DI_47 to DI_32 (3000V isolation)
J4: select internal/external power for DI_63 to DI_48 (3000V isolation)

Isolation bank 1: DI_0 to DI_15, Power=CON1_18, Ground=CON1_1
Isolation bank 2: DI_16 to DI_31, Power=CON1_37, Ground=CON1_20
Isolation bank 3: DI_32 to DI_47, Power=CON2_18, Ground=CON2_1
Isolation bank 4: DI_48 to DI_63, Power=CON2_37, Ground=CON2_20
All these four banks are fully isolated from each other.

The DC/DC1 is used to provide the internal power supply for bank1 & bank2.
The DC/DC2 is used to provide the internal power supply for bank3 & bank4.

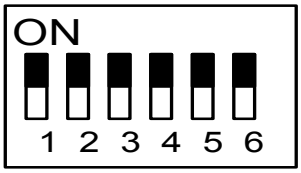
2.2.1 Jumper setting

JP1/JP2/JP3/JP4 jumper setting:

	**Internal**	
	**External**	**Default Setting**

2.2.2 I/O Base Address Setting

SW1: Base Address



Default Address: 0x200

Address Table:

Address	A8	A7	A6	A5	A4	A3
*0x200	ON	ON	ON	ON	ON	ON
0x208	ON	ON	ON	ON	ON	OFF
0x210	ON	ON	ON	ON	OFF	ON
0x218	ON	ON	ON	ON	OFF	OFF
0x220	ON	ON	ON	OFF	ON	ON
:	:	:	:	:	:	:
0x300	OFF	ON	ON	ON	ON	ON
0x308	OFF	ON	ON	ON	ON	OFF
0x310	OFF	ON	ON	ON	OFF	ON
:	:	:	:	:	:	:

- **Default Base Address: 0x200**

2.2.3 I/O Register Address

The ISO-P64 card occupies 8 consecutive PC I/O address. The following table lists the registers and their locations.

ISO-P64 I/O Register Address

Address	D/I Channel	ISO-P64
Base+0	Channel 0-7	Read Only
Base+1	Channel 8-15	Read Only
Base+2	Channel 16-23	Read Only
Base+3	Channel 24-31	Read Only
Base+4	Channel 32-39	Read Only
Base+5	Channel 40-47	Read Only
Base+6	Channel 48-55	Read Only
Base+7	Channel 56-63	Read Only

Read

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Base+0	7	6	5	4	3	2	1	0
Base+1	15	14	13	12	11	10	9	8
Base+2	23	22	21	20	19	18	17	16
Base+3	31	30	29	28	27	26	25	24
Base+4	39	38	37	36	35	34	33	32
Base+5	47	46	45	44	43	42	41	40
Base+6	55	54	53	52	51	50	49	48
Base+7	63	62	61	60	59	58	57	56

2.3 ISO-C64 Board Layout

The board layout of ISO-C64 is given as follows:

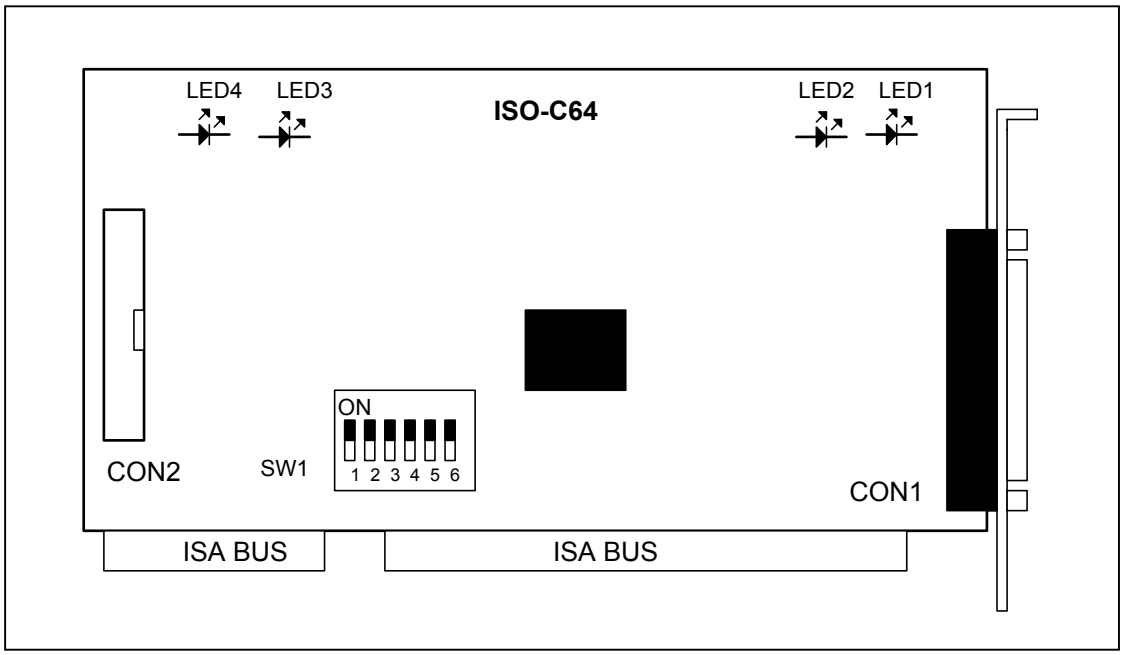


Figure 2-5. Board layout of ISO-C64

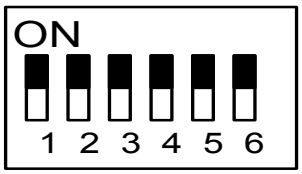
LED 1: power indicator for DO_0 to DO_15
LED 2: power indicator for DO_31 to DO_16
LED 3: power indicator for DO_47 to DO_32
LED 4: power indicator for DO_63 to DO_48

Isolation bank 1: DO_0 to DO_15, Power=CON1_18, Ground=CON1_1
Isolation bank 2: DO_16 to DO_31, Power=CON1_37, Ground=CON1_20
Isolation bank 3: DO_32 to DO_47, Power=CON2_18, Ground=CON2_1
Isolation bank 4: DO_48 to DO_63, Power=CON2_37, Ground=CON2_20

All these four banks are fully isolated from each other.

2.3.1 I/O Base Address Setting

SW1: Base Address



Default Address: 0x200

Address Table:

Address	A8	A7	A6	A5	A4	A3
*0X200	ON	ON	ON	ON	ON	ON
0X208	ON	ON	ON	ON	ON	OFF
0x210	ON	ON	ON	ON	OFF	ON
0x218	ON	ON	ON	ON	OFF	OFF
0x220	ON	ON	ON	OFF	ON	ON
:	:	:	:	:	:	:
0x300	OFF	ON	ON	ON	ON	ON
0x308	OFF	ON	ON	ON	ON	OFF
0x310	OFF	ON	ON	ON	OFF	ON
:	:	:	:	:	:	:

- **Default Address: 0x200**

2.3.2 I/O Register Address

The ISO-C64 card occupies 8 consecutive PC I/O address. The following table lists the registers and their locations.

ISO-C64 I/O Register Address

Address	D/O Channel	ISO-C64
Base+0	Channel 0-7	Write Only
Base+1	Channel 8-15	Write Only
Base+2	Channel 16-23	Write Only
Base+3	Channel 24-31	Write Only
Base+4	Channel 32-39	Write Only
Base+5	Channel 40-47	Write Only
Base+6	Channel 48-55	Write Only
Base+7	Channel 56-63	Write Only

Write

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Base+0	7	6	5	4	3	2	1	0
Base+1	15	14	13	12	11	10	9	8
Base+2	23	22	21	20	19	18	17	16
Base+3	31	30	29	28	27	26	25	24
Base+4	39	38	37	36	35	34	33	32
Base+5	47	46	45	44	43	42	41	40
Base+6	55	54	53	52	51	50	49	48
Base+7	63	62	61	60	59	58	57	56

2.4 Isolated D/I Architecture

The D/I architecture of ISO-P32C32 & ISO-P64 are the same. The users can select internal power or external power to supply photo-couple digital input power. The block diagram of D/I are given as follows:

Configure 1: Internal power supply (Default Setting)

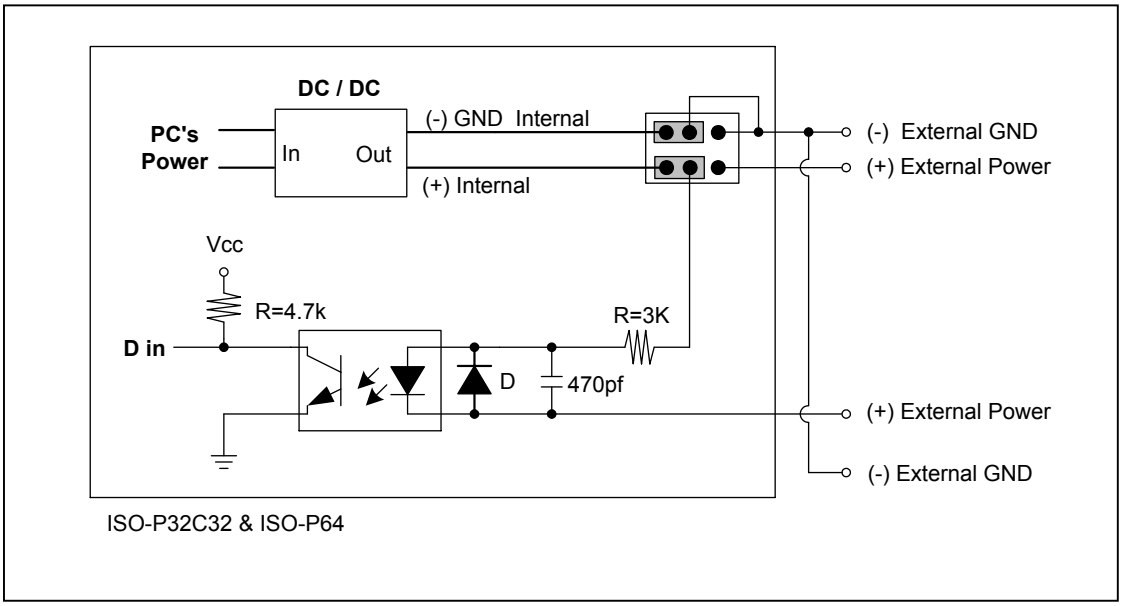


Figure 6. Isolated D/I Architecture with internal power supply

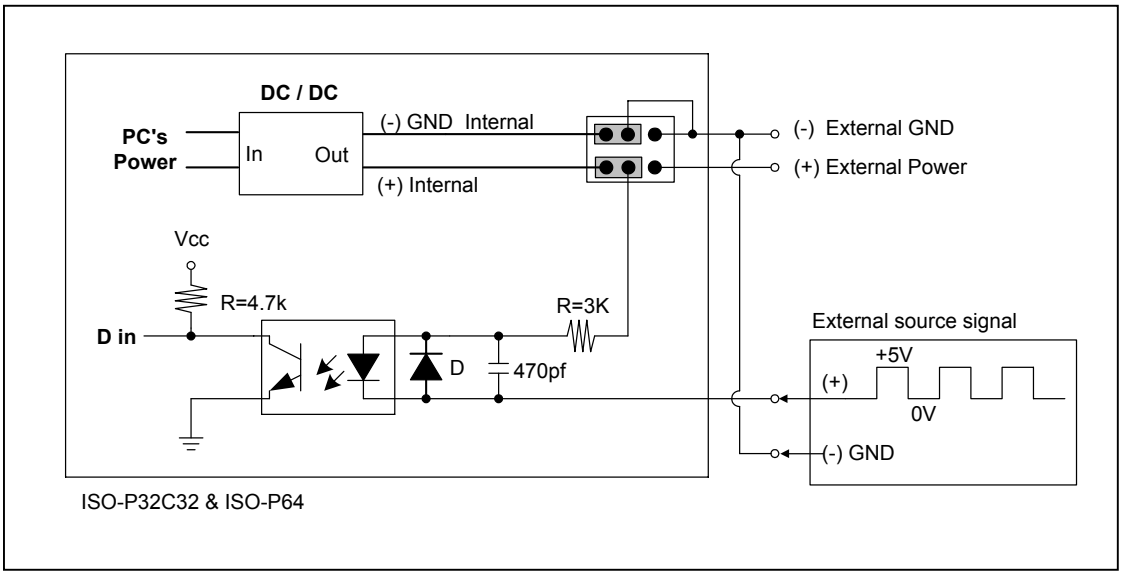


Figure 2-6. Typical Applications of D/I with internal power supply.

Configure 2: External power supply

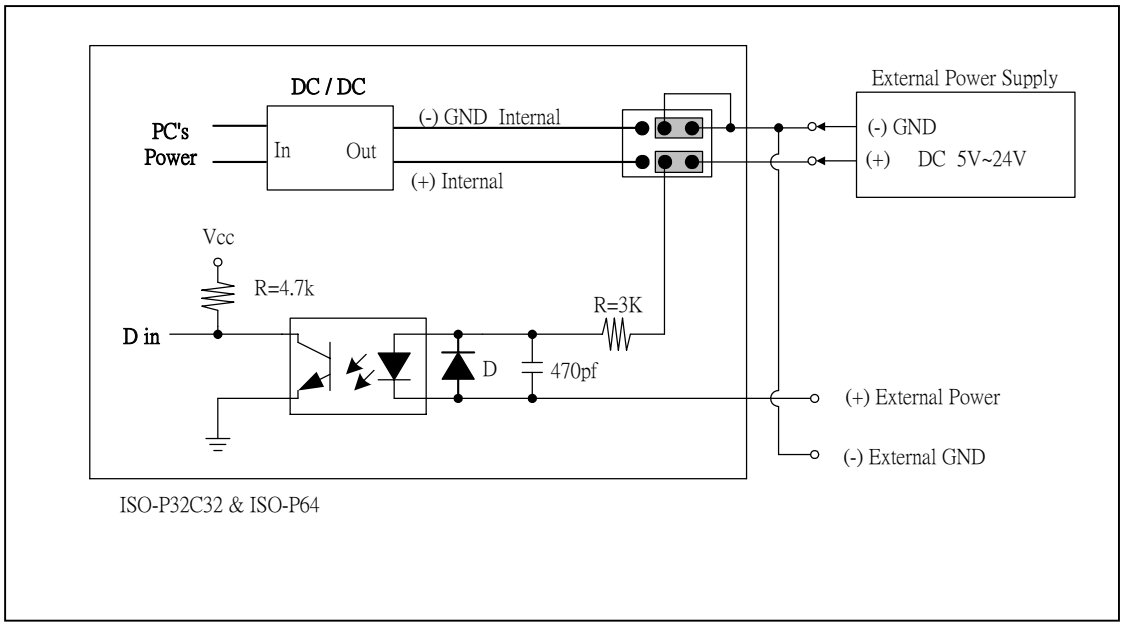


Figure 2-7. Isolated D/I Architecture with external power supply

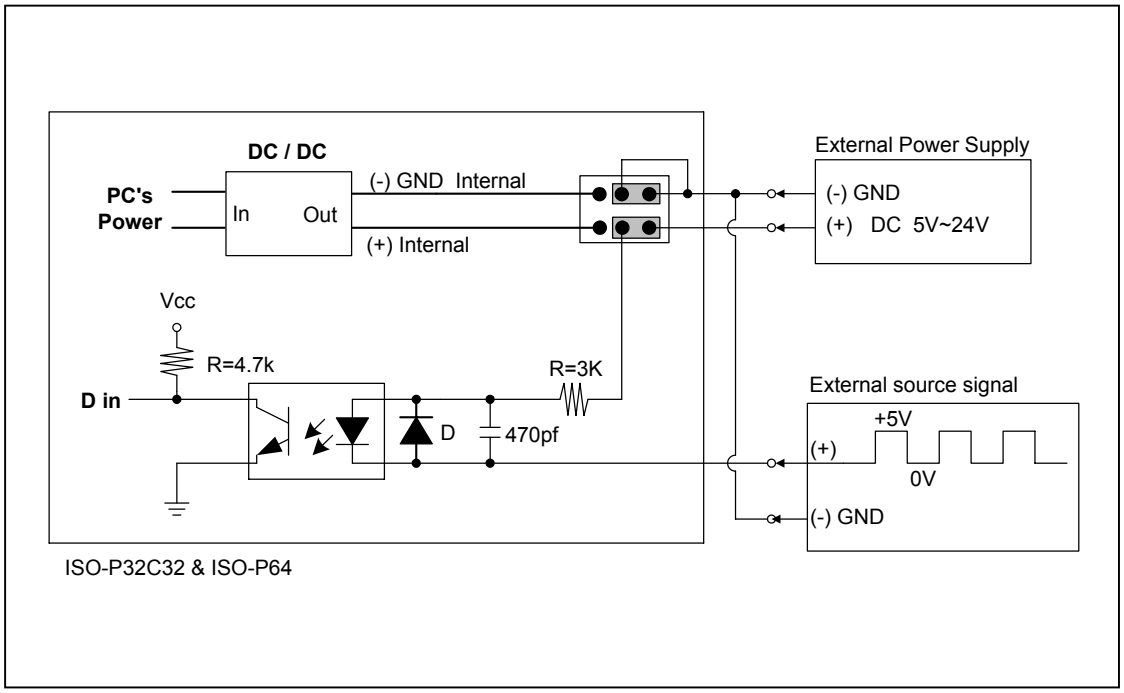
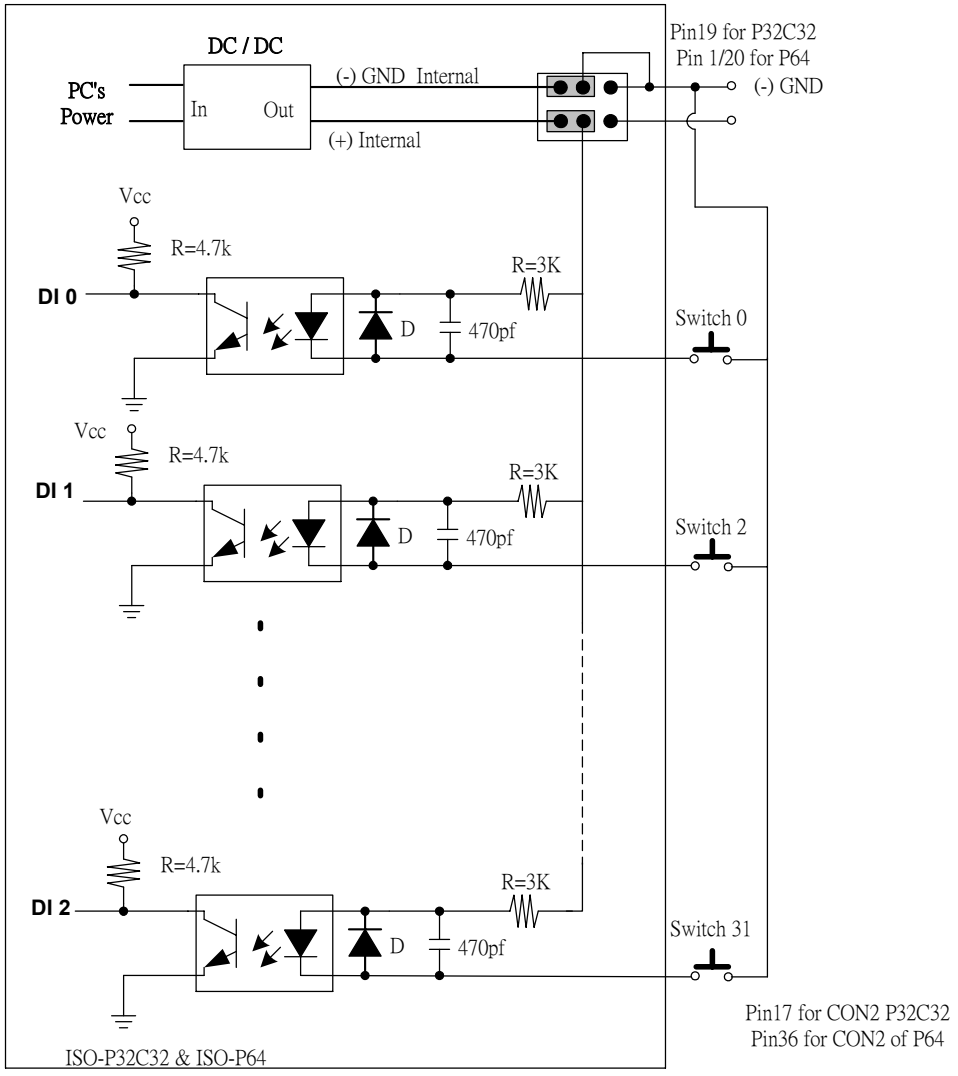
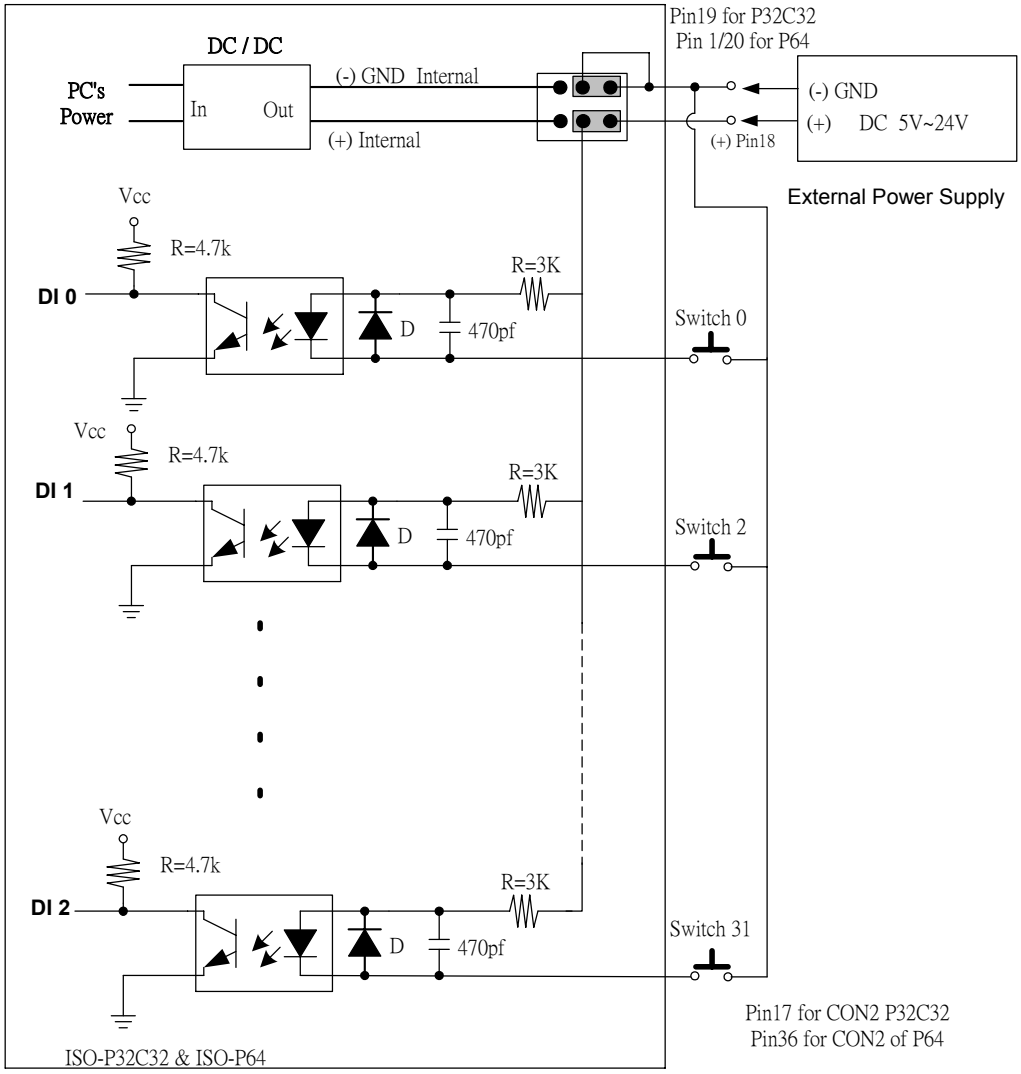


Figure 2-8. Typical Applications of D/I with external power supply.

2.4.1 Sample D/I Architecture for Internal Power



2.4.2 Sample D/I Architecture for External Power



2.5 Isolated D/O Architecture

The D/O architecture of ISO-P32C32 & ISO-C64 are the same. The block diagram of D/O are given as follows:

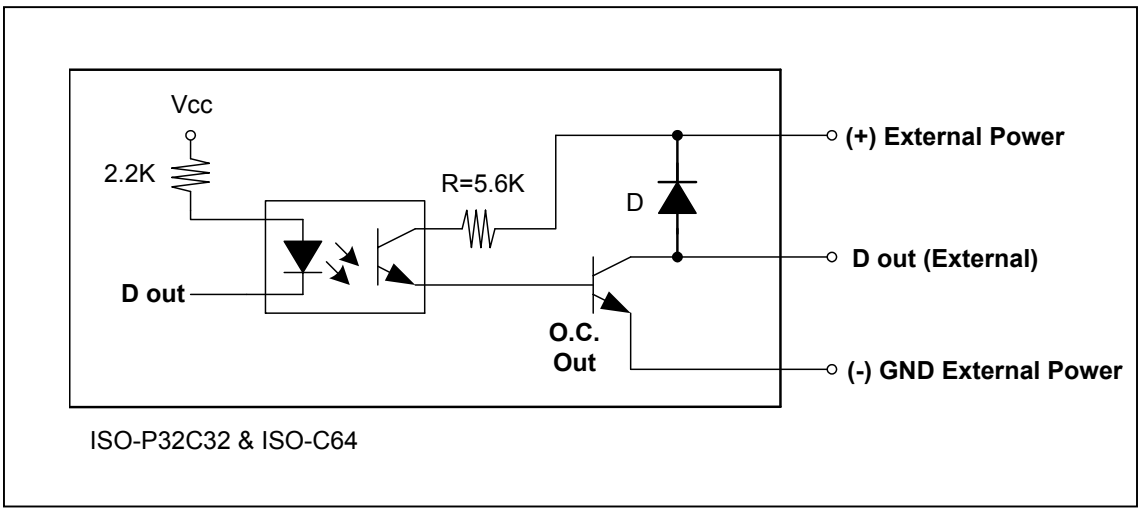


Figure 2-9. Isolated D/O Architecture

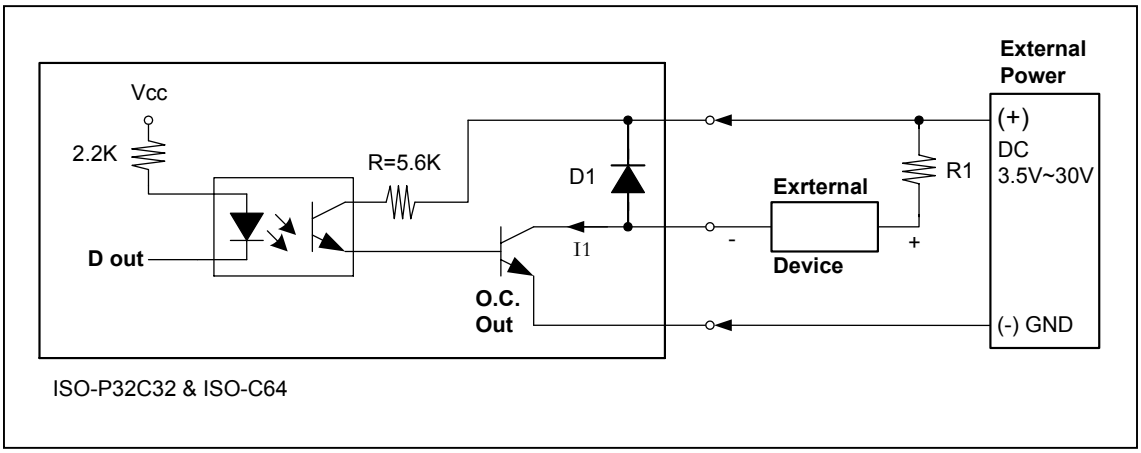
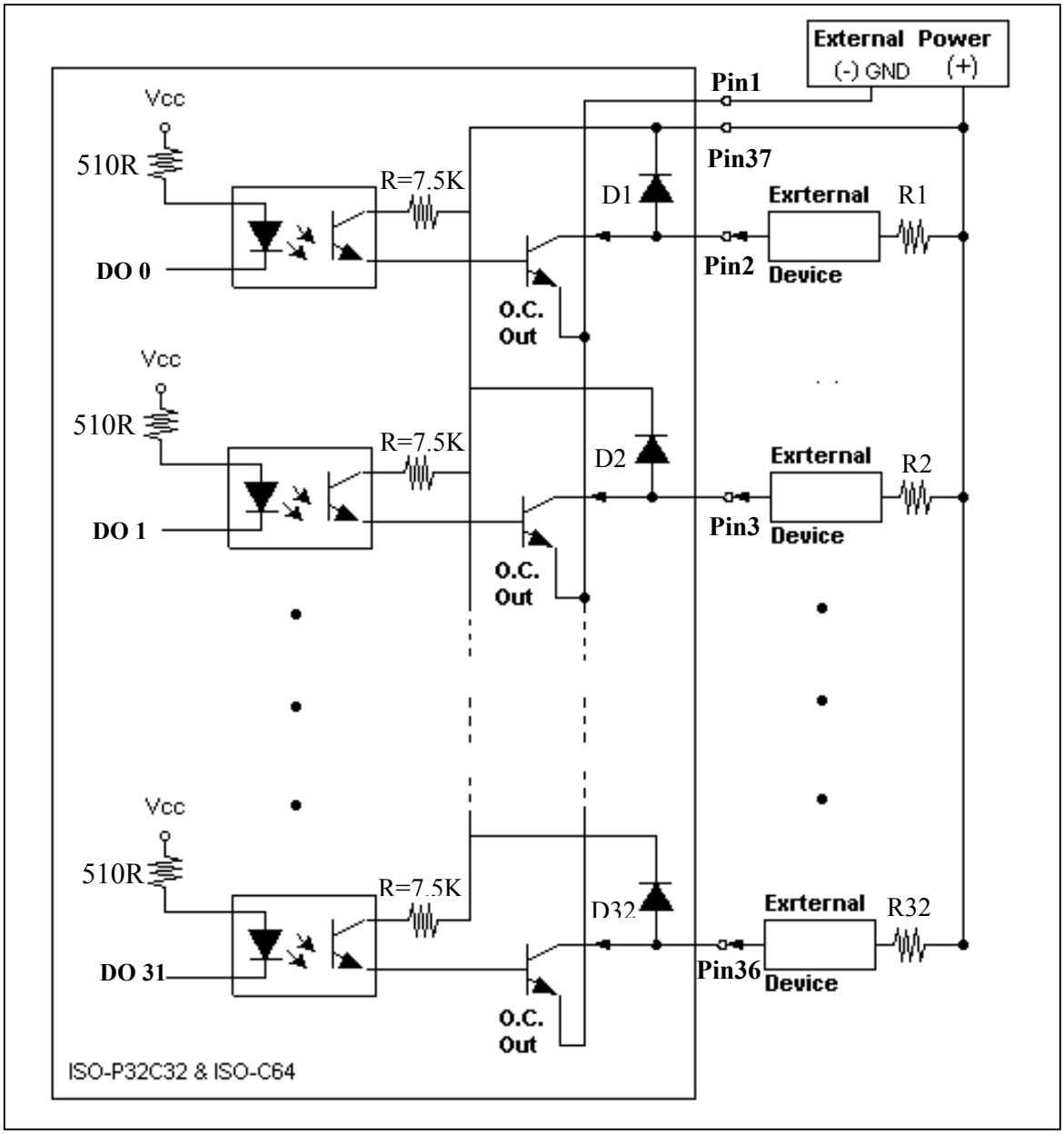


Figure 2-10. Typical Applications of D/O

NOTE:

1. The I1, I2, … & I32 must be < 100 mA
2. The R1, R2, ... & R32 are current-limit resistors. They must be designed to let I1, I2, ... & I32 <100 mA.
3. If the internal resistance of external device is large enough, the R can be omitted.
4. The D1, D2, ... & D31 are common-cathode diodes for switching inductive lords. They can be used as relay driver, hammer driver, lamp driver, display driver, line driver & logic buffers.

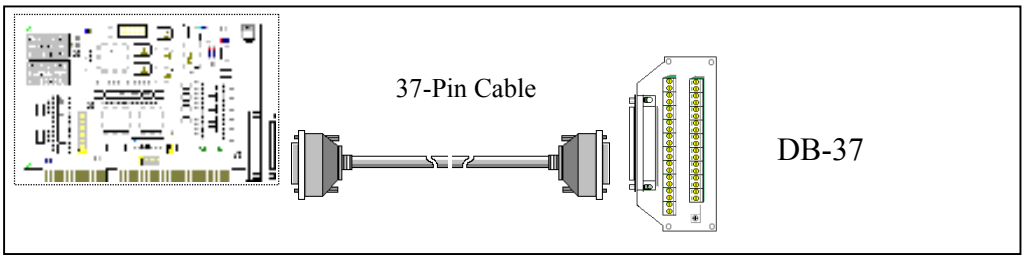
2.5.1 Sample D/O Architecture



2.6 Daughter Boards

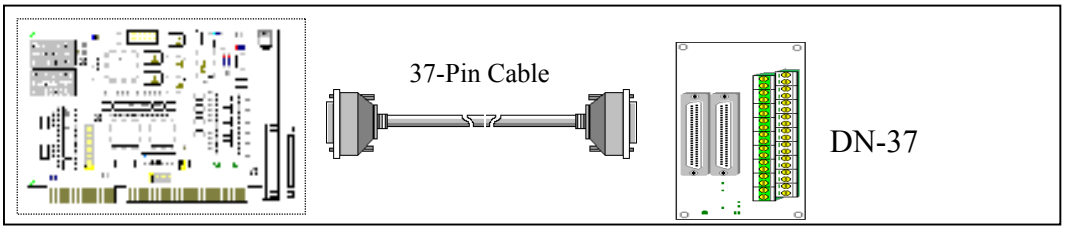
2.6.1 DB-37

The DB-37 is a general purpose daughter board for D-sub 37 pins. It is designed for easy wire connection.



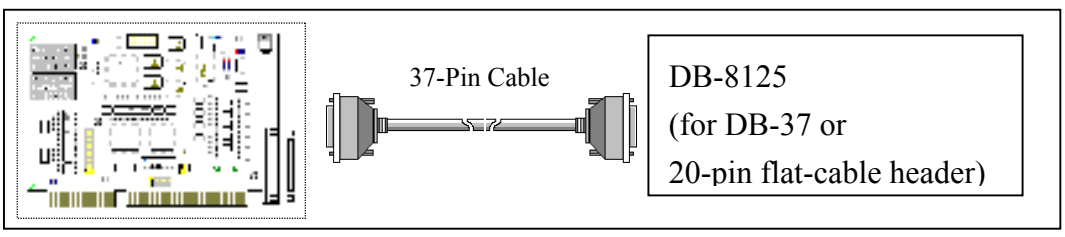
2.6.2 DN-37

The DN-37 is a general purpose daughter board for DB-37 with DIN-Rail Mounting. It is designed for easy wire connection.

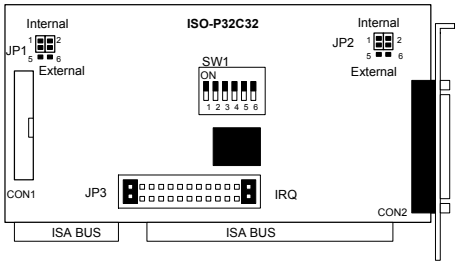


2.6.3 DB-8125

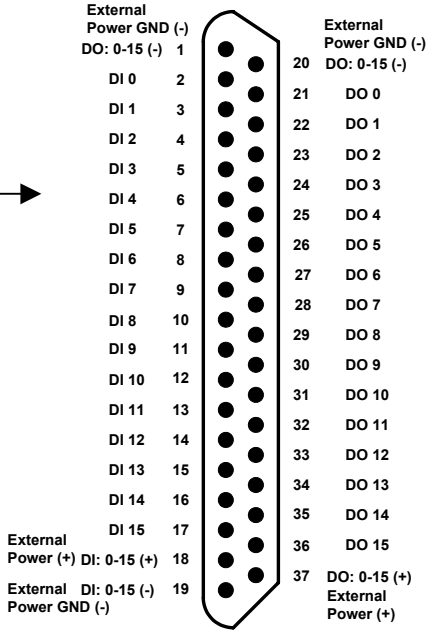
The DB-8125 is a general purpose screw terminal board. It is designed for easy wire connection. There are one DB-37 & two 20-pin flat-cable header in the DB-8125.



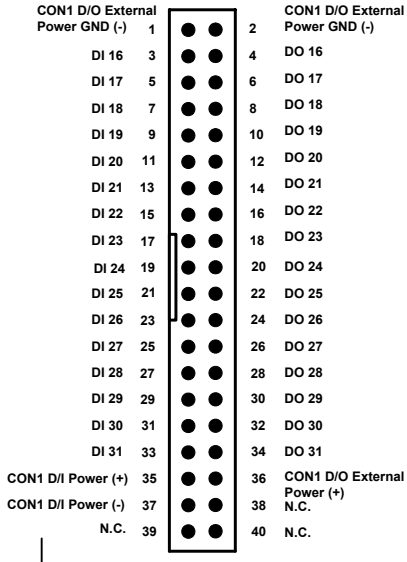
2.7 Pin Assignment of ISO-P32C32



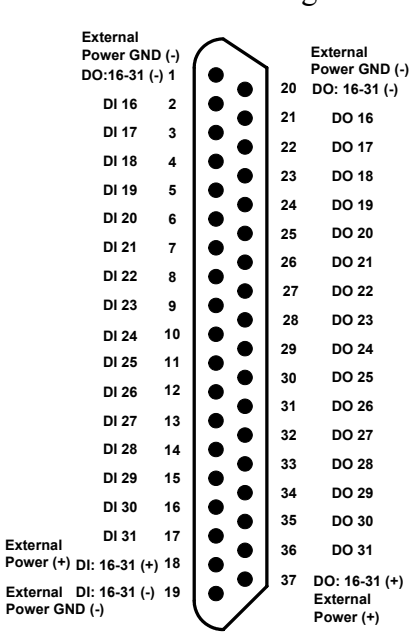
CON2: Pin assignment

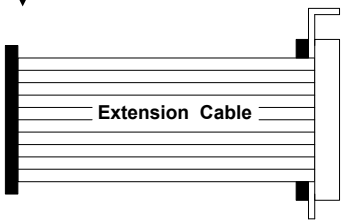


CON1: Pin assignment



CON1: Pin assignment

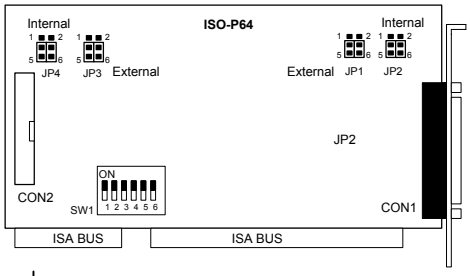




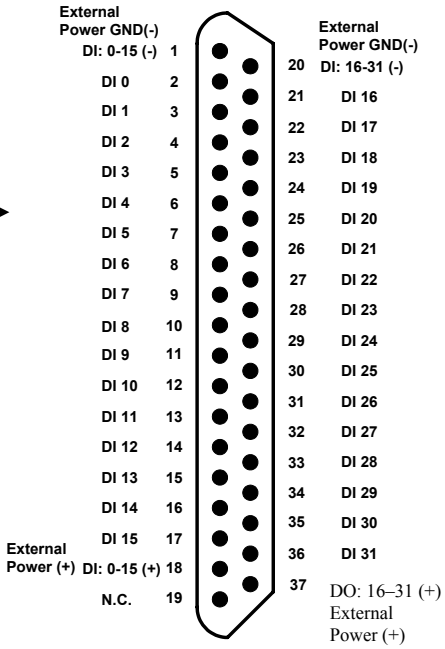
DB-40-Pin conversion DB-37-Pin

Pin assignment of CON1 via extension cable

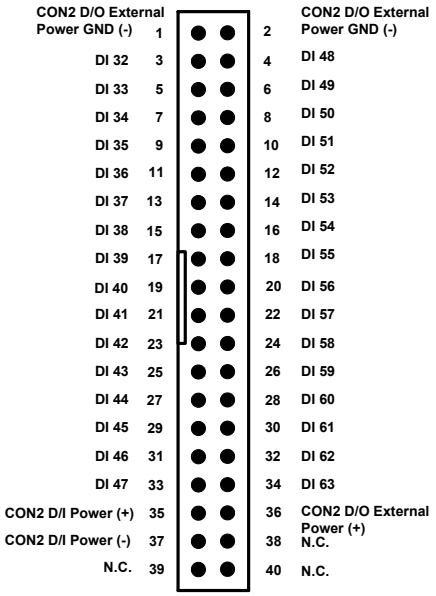
2.8 Pin Assignment of ISO-P64



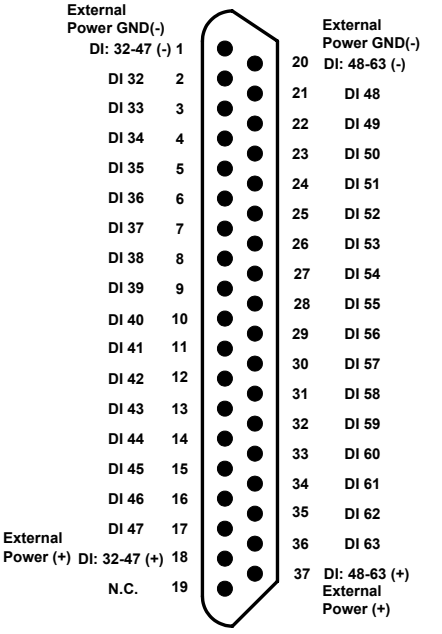
CON1 Pin assignment

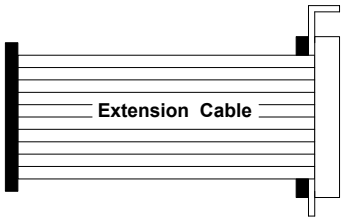


CON2 Pin assignment



CON2 Pin assignment

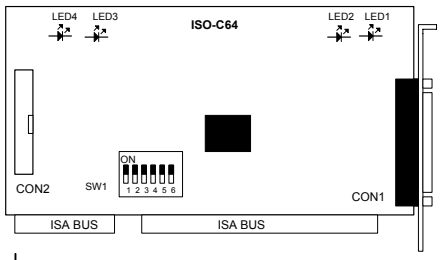




Pin assignment of CON2 via extension cable

DB-40-Pin conversion DB-37-Pin

2.9 Pin Assignment of ISO-C64

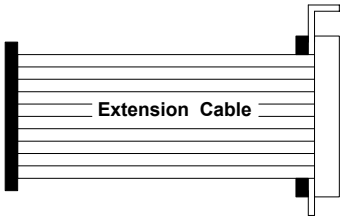


CON1 Pin assignment

External Power GND (-)		20	External Power GND (-)
DO: 0-15 (-)	1		DO: 16-31 (-)
DO 0	2	21	DO 16
DO 1	3	22	DO 17
DO 2	4	23	DO 18
DO 3	5	24	DO 19
DO 4	6	25	DO 20
DO 5	7	26	DO 21
DO 6	8	27	DO 22
DO 7	9	28	DO 23
DO 8	10	29	DO 24
DO 9	11	30	DO 25
DO 10	12	31	DO 26
DO 11	13	32	DO 27
DO 12	14	33	DO 28
DO 13	15	34	DO 29
DO 14	16	35	DO 30
DO 15	17	36	DO 31
External Power (+) DO: 0-15 (+)	18	37	DO: 16-31 (+)
N.C.	19		External Power (+)

CON2 Pin assignment

CON2 D/O External Power GND (-)	1	2	CON2 D/O External Power GND (-)
DO 32	3	4	DO 48
DO 33	5	6	DO 49
DO 34	7	8	DO 50
DO 35	9	10	DO 51
DO 36	11	12	DO 52
DO 37	13	14	DO 53
DO 38	15	16	DO 54
DO 39	17	18	DO 55
DO 40	19	20	DO 56
DO 41	21	22	DO 57
DO 42	23	24	DO 58
DO 43	25	26	DO 59
DO 44	27	28	DO 60
DO 45	29	30	DO 61
DO 46	31	32	DO 62
DO 47	33	34	DO 63
CON2 D/I Power (+)	35	36	CON2 D/O External Power (+)
CON2 D/I Power (-)	37	38	N.C.
N.C.	39	40	N.C.



DB-40-Pin conversion DB-37-Pin

CON2 Pin assignment

External Power GND (-)		20	External Power GND (-)
DO: 32-47(-)	1		DO: 48-63(-)
DO 32	2	21	DO 48
DO 33	3	22	DO 49
DO 34	4	23	DO 50
DO 35	5	24	DO 51
DO 36	6	25	DO 52
DO 37	7	26	DO 53
DO 38	8	27	DO 54
DO 39	9	28	DO 55
DO 40	10	29	DO 56
DO 41	11	30	DO 57
DO 42	12	31	DO 58
DO 43	13	32	DO 59
DO 44	14	33	DO 60
DO 45	15	34	DO 61
DO 46	16	35	DO 62
DO 47	17	36	DO 63
External Power (+) DO: 32-47 (+)	18	37	DO: 48-63 (+)
N.C.	19		External Power (+)

Pin assignment of CON2 via extension cable

3. The applications of Digital I/O

3.1 The example of ISO-P32C32

- **The circuit diagram of D/O of ISO-P32C32 is given as follows:**

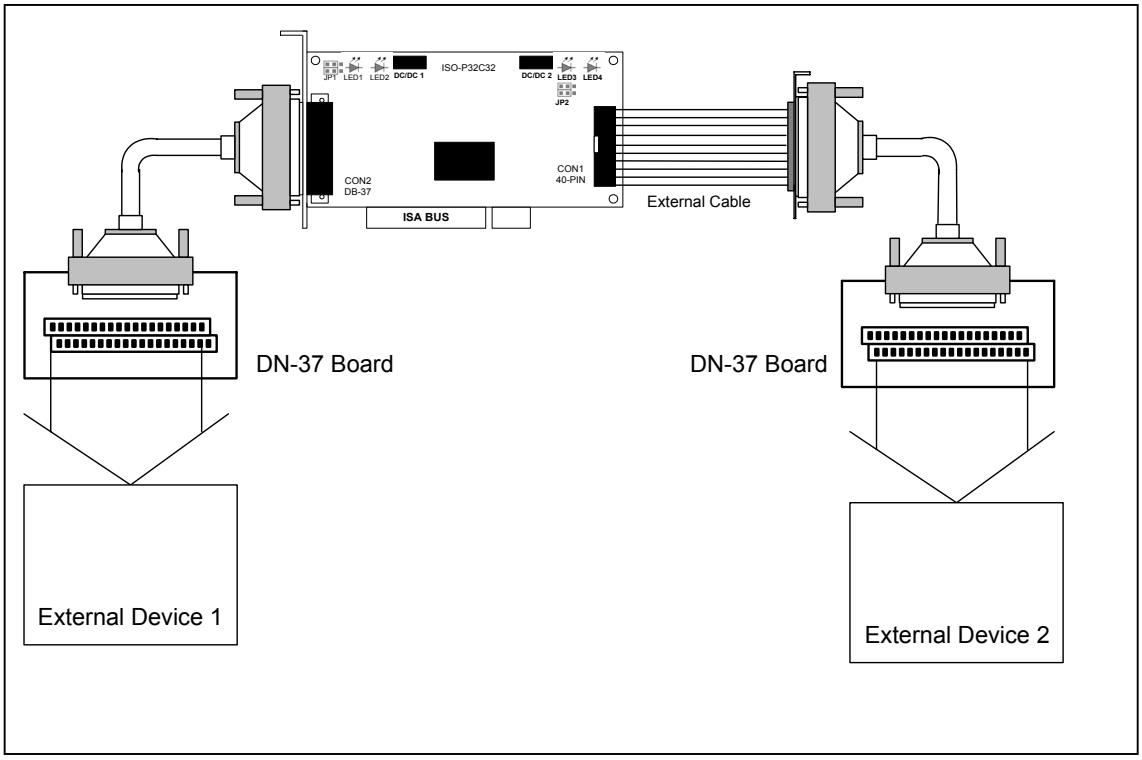


Figure 3-1. The example of digital inputs/outputs for ISO-P32C32

- The circuit diagram of external device 1: Refer to Figure 11.
- The circuit diagram of external device 2: Refer to Figure 12.

- **The test circuit diagram of D/I/O of ISO-P32C32 is given as follows:**

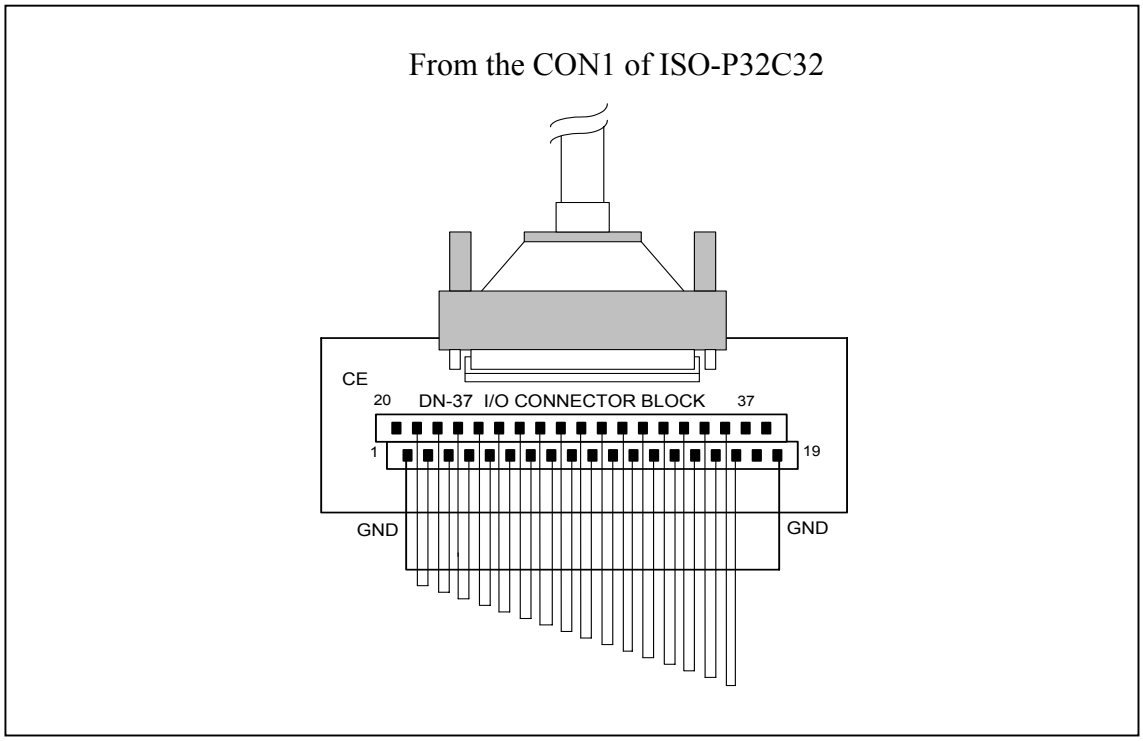


Figure 3-2. The D/I/O of CON1 of ISO-P32C32

- The D/I of CON2 of ISO-P32C32 for internal power.

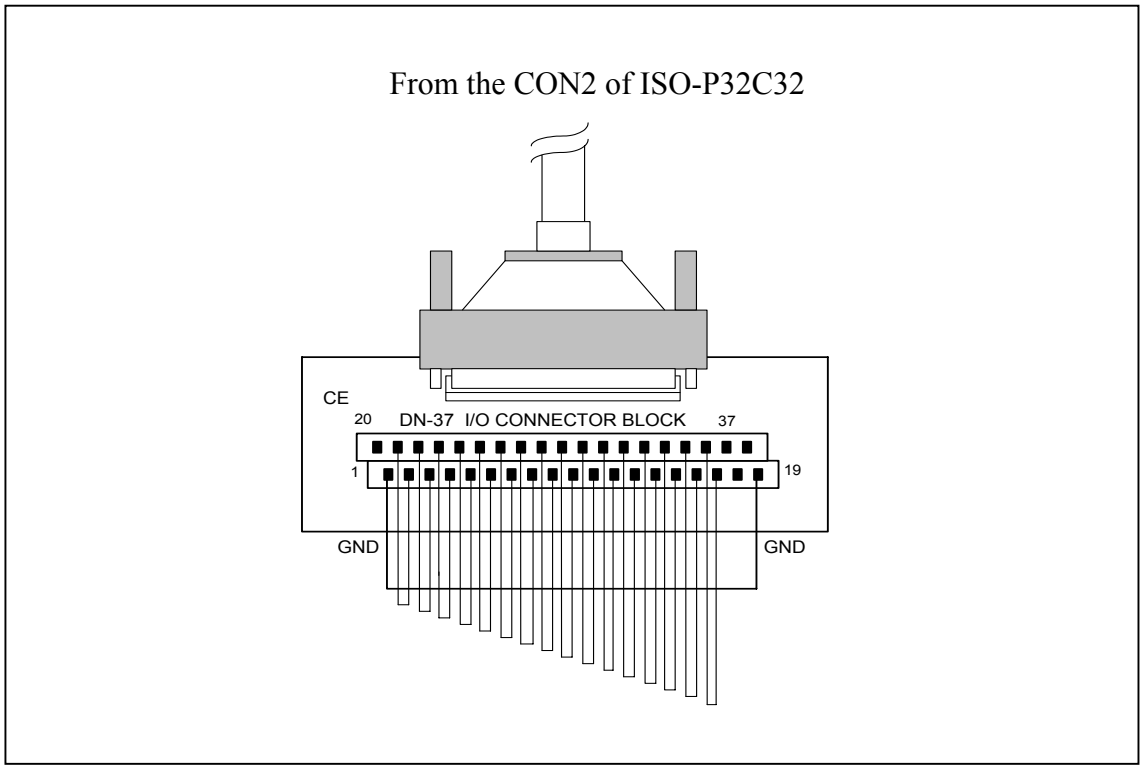
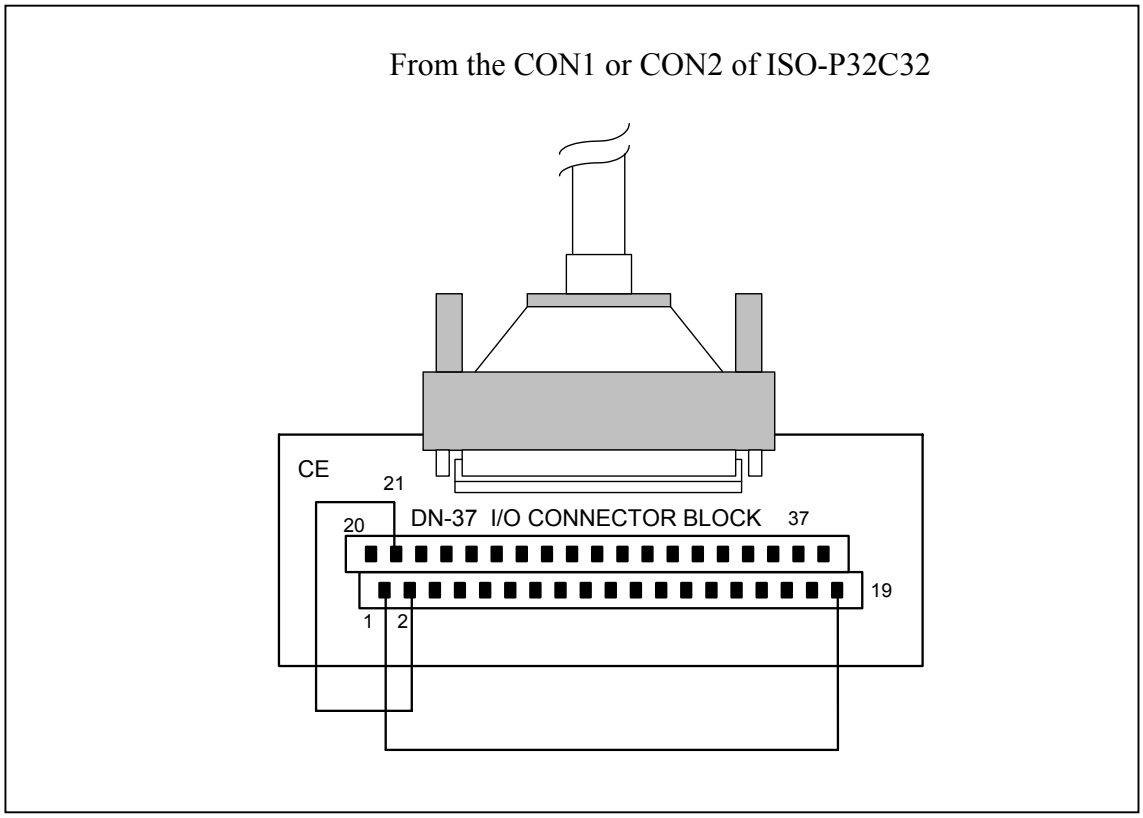


Figure 3-3. The D I/O of CON2 of ISO-P32C32

- The D/I of CON2 of ISO-P32C32 for internal power.

- **The interrupt input of circuit diagram (1) of ISO-P32C32 is given as follows:**



- Figure 3-4. The interrupt input for ISO-P32C32

- **The interrupt input of circuit diagram(2) of ISO-P32C32 is given as follows:**

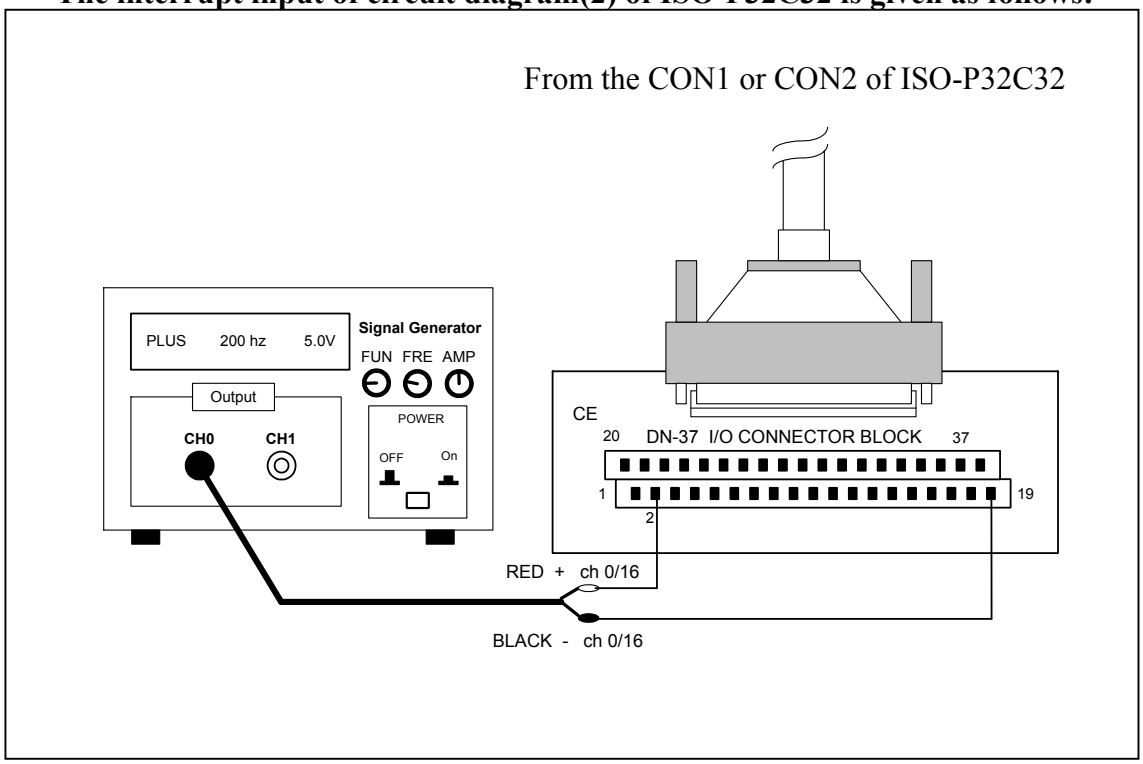


Figure 3-5. The interrupt input for ISO-P32C32

3.2 The example of ISO-P64

- **The circuit diagram of D/I for ISO-P64 is given as follows:**

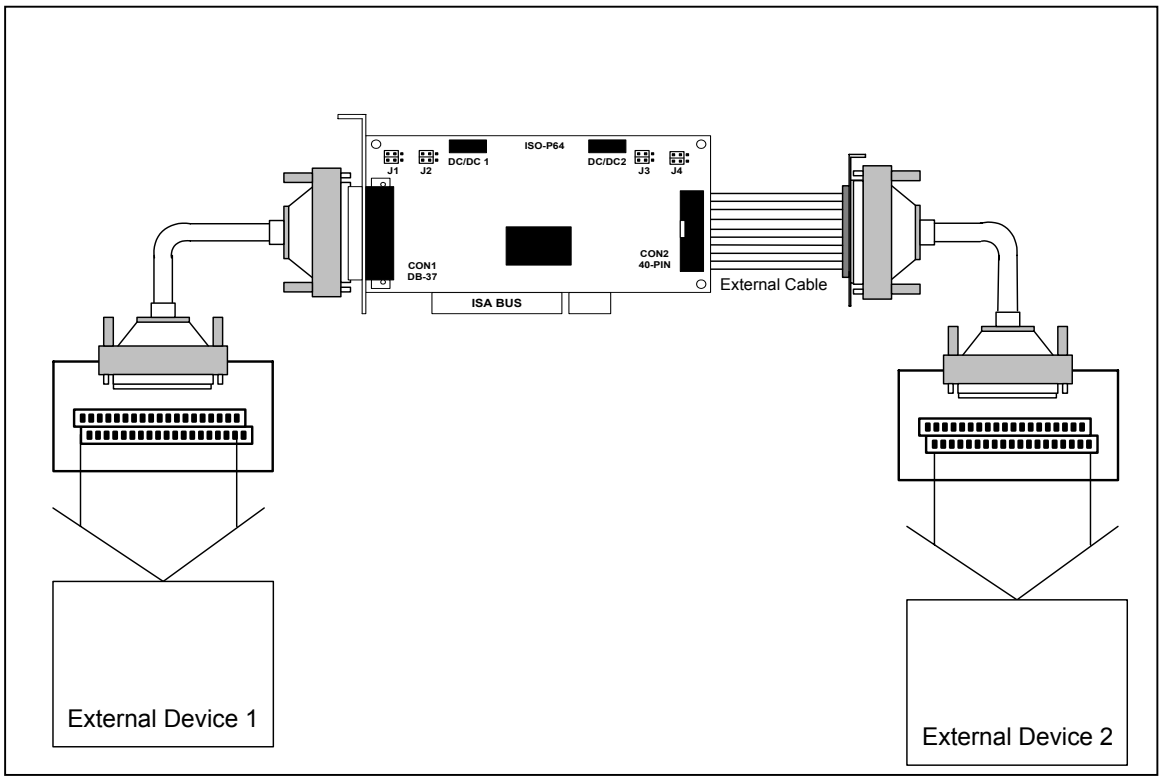
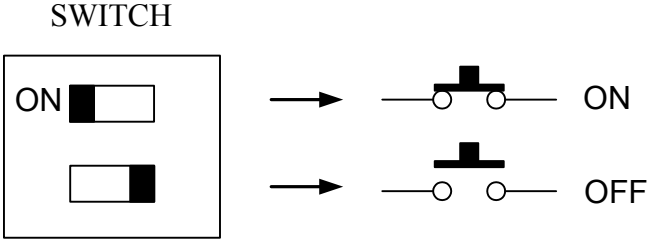


Figure 3-6. The example of digital inputs for ISO-P64

- The circuit diagram of external device 1: Refer to Figure 14.
- The circuit diagram of external device 2: Refer to Figure 15.



The circuit diagram of external device 1 is given as follows:

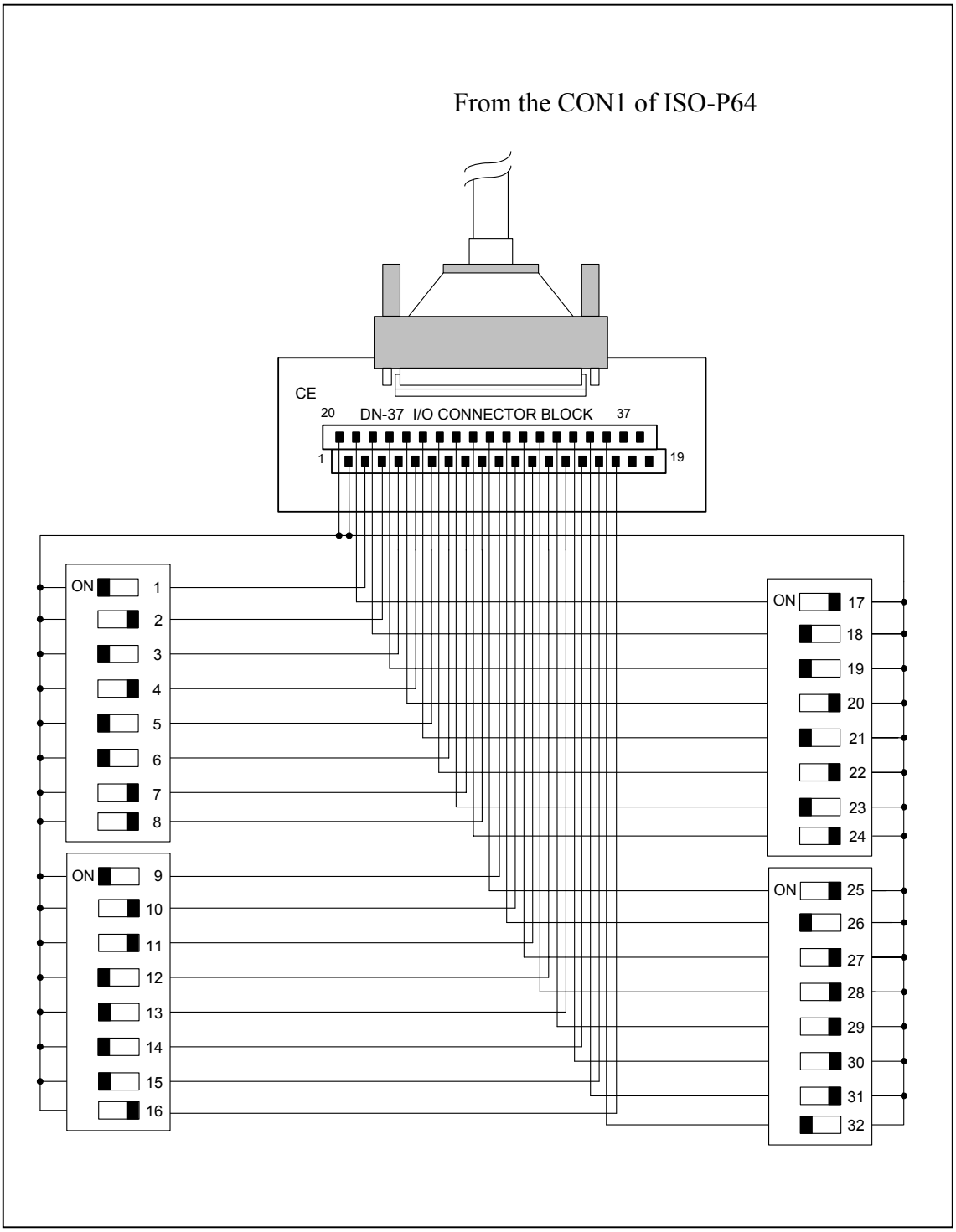


Figure 3-7. The circuit diagram of external device 2 for the digital inputs of ISO-P64

- The D/I of CON1 of ISO-P64 select internal power.

The circuit diagram of external device 2 is given as follows:

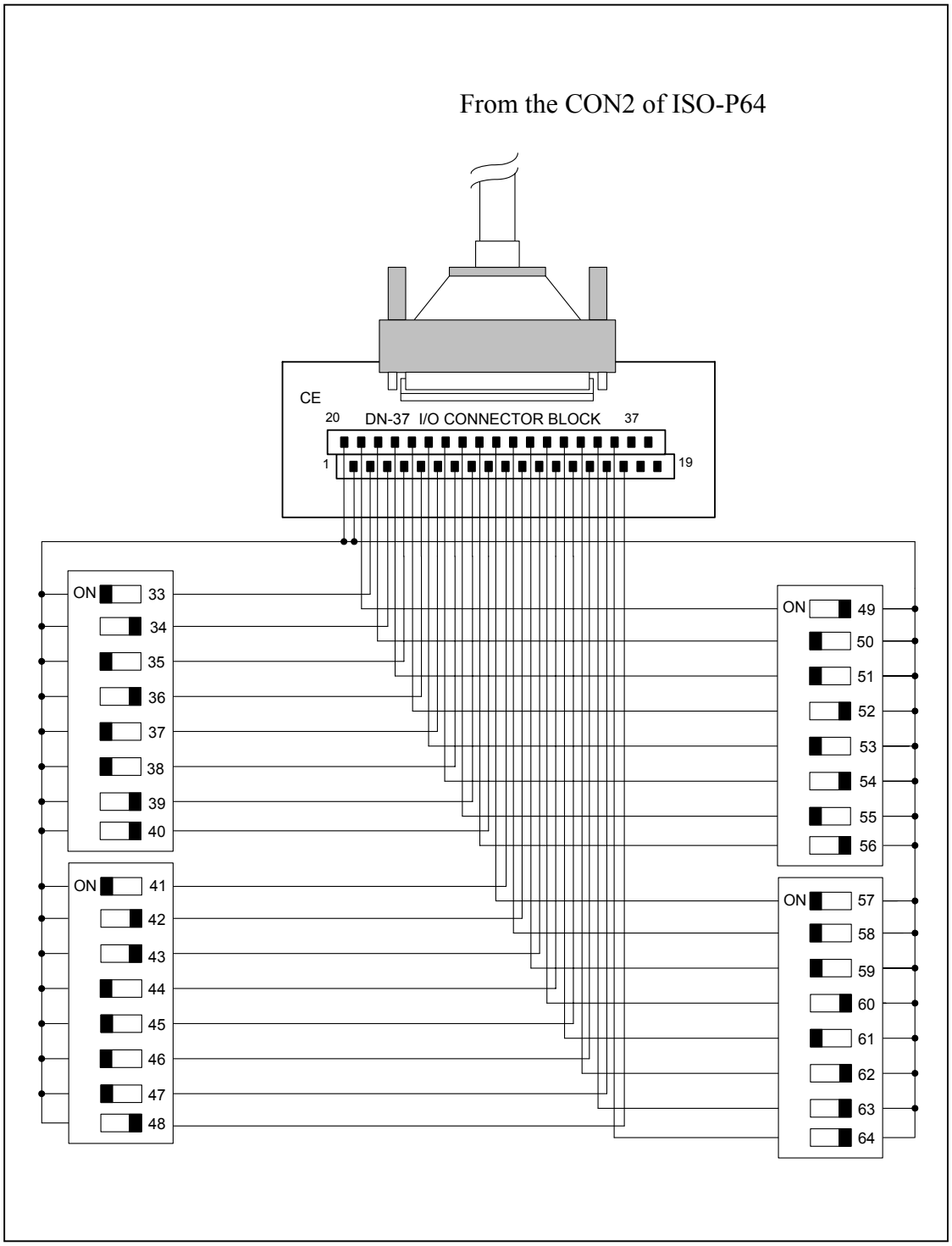


Figure 3-8. The circuit diagram of external device 2 for the digital inputs of ISO-P64

- The D/I of CON2 of ISO-P64 select internal power.

3.3 The example of ISO-C64

- **The circuit diagram of D/O for ISO-P32C32 is given as follows:**

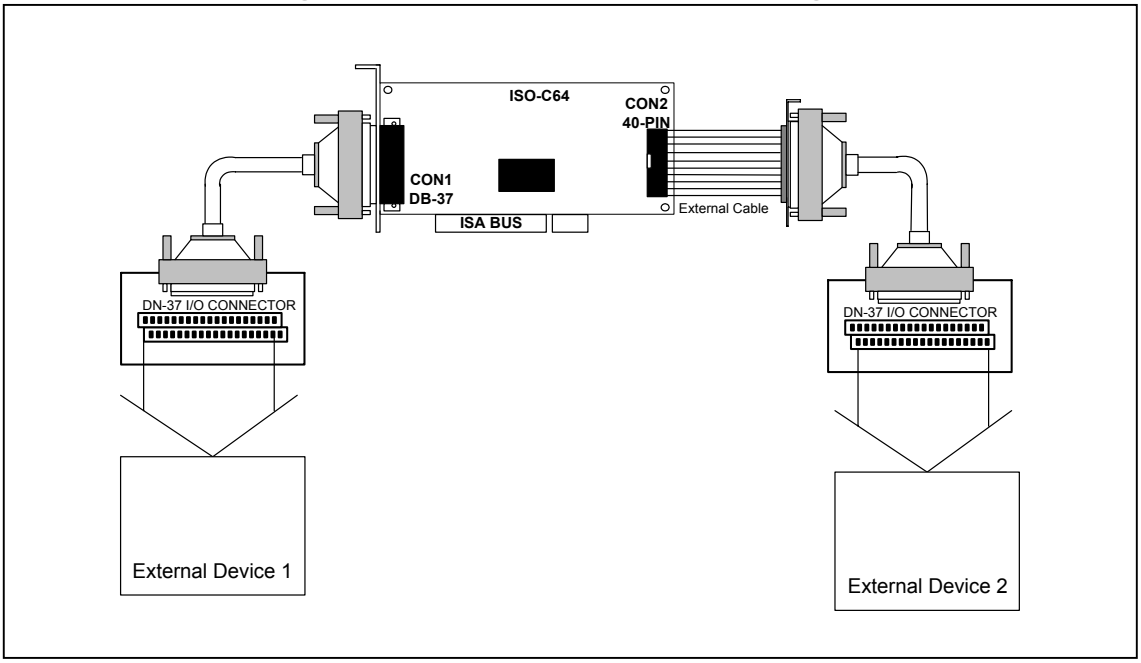


Figure 3-9. The example of digital outputs for ISO-C64

- The circuit diagram of external device 1: Refer to Figure 17.
- The circuit diagram of external device 2: Refer to Figure 18.

The circuit diagram of external device 1 is given as follows:

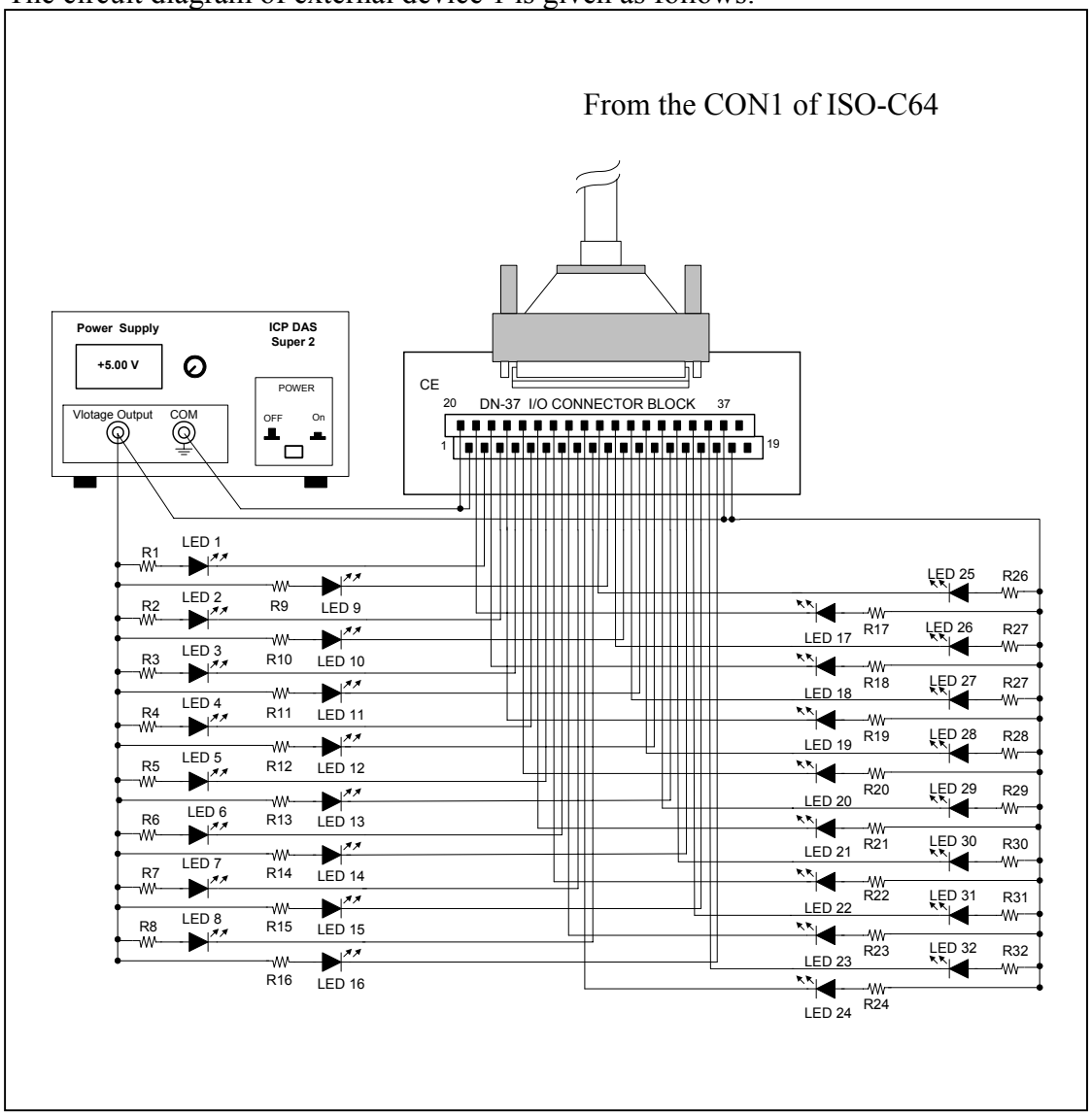


Figure 3-10. The circuit diagram of external device 1 for the digital outputs of ISO-C64

- The resistor of R1~R32 are 330 ohm.
- The LED 1~LED 32 are light emitting diode.
- The pin-1/20 are the GND signal of DO_0~DO_15 / DO_16~DO_31.
- The pin-18/37 are the voltage(+) signal of DO_0~DO_15 / DO_16~DO_31 (input DC +5V~+24V).

The circuit diagram of external device 2 is given as follows:

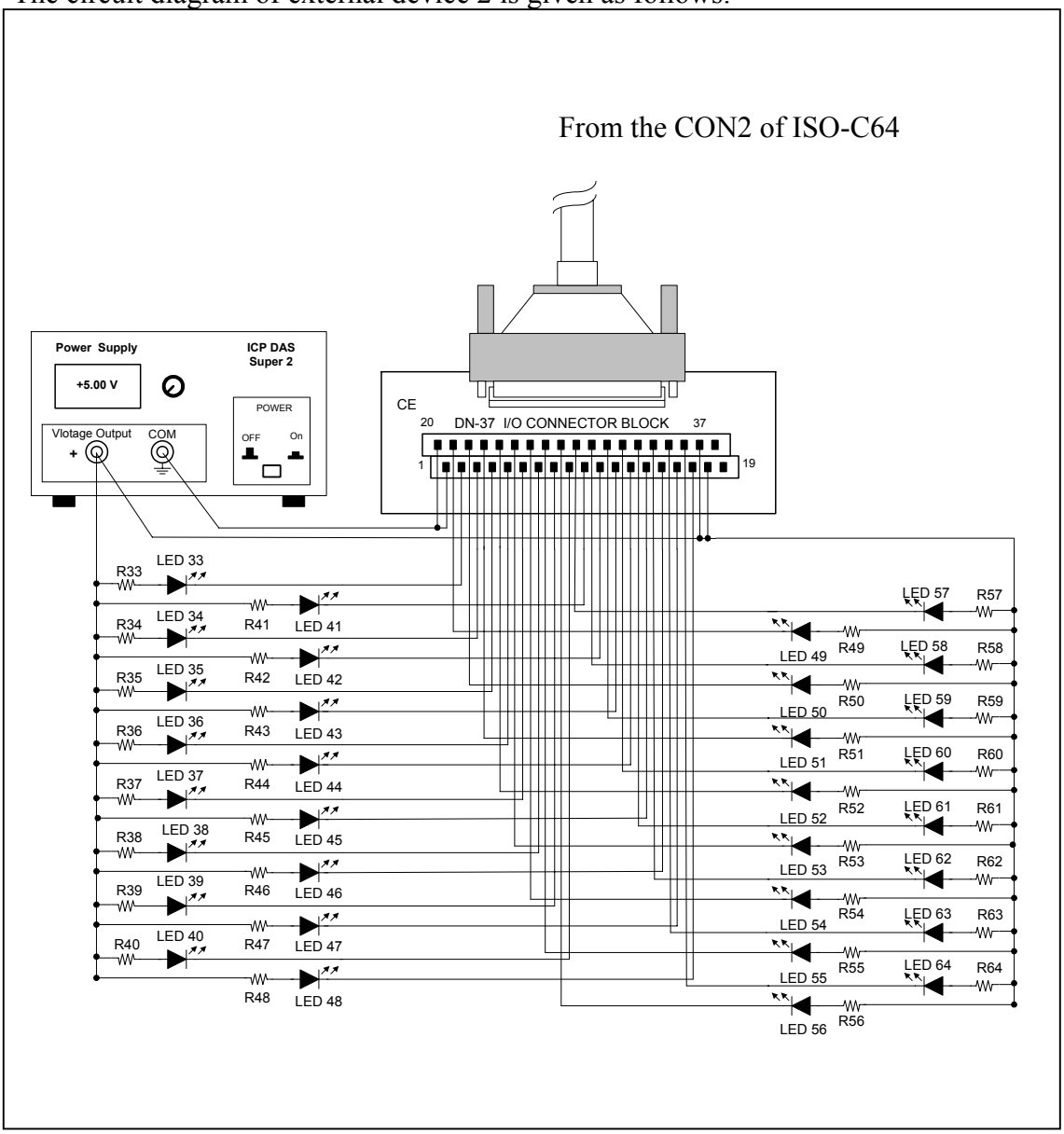


Figure 3-11. The circuit diagram of external device 2 for the digital outputs of ISO-C64

- The resistor of R33~R64 are 330 ohm.
- The LED 33~LED 64 are light emitting diode.
- The pin-1/20 are the GND signal of DO_32~DO_47 / DO_48~DO_63.
- The pin-18/37 are the voltage(+) signal of DO_32~DO_47 / DO_32~DO_63 (input DC +5V~+24V).

4. Demo Program

There are many demo programs given in the company floppy disk or CD-ROM. After the software installation, the driver will be installed into disk as following:

4.1.1 DEMO1 D/O for ISO-P32C32

```
/* ------------------------------------------------------------- */
/* Demo 1: Digital Output of ISO_P32C32                          */
/* Step 1: The circuit diagram of hardware: refer to Sec 3.1     */
/* Step 1: run demo1.EXE under DOS                               */
/* ------------------------------------------------------------- */

#include <dos.h>
#define  wBase     0x200

void main()
{
char c;
unsigned char i;

clrscr();

/* initial value  */
outp(wBase+0,0x00);
outp(wBase+1,0x00);
outp(wBase+2,0x00);
outp(wBase+3,0x00);

while(1)
{
printf("\n\n --------- Digital output of ISO-P32C32  ------------");
for (i=1;i<=0x80;i=i<<1)
    {
      outp(wBase+0,i);        /* DO_07 to DO_00 of CON2 */
      outp(wBase+1,i);        /* DO_15 to DO_08 of CON2 */
      outp(wBase+2,i);        /* DO_23 to DO_16 of CON1 */
      outp(wBase+3,i);        /* DO_31 to DO_24 of CON1 */
      sleep(1);
      printf("\nD 31-0 Output Value = %02x,%02x,%02x,%02x",i,i,i,i);

 if(i==0x80)  { i=0x01; break; }

    if (kbhit()!=0)
       {
       c=getch();
       if ((c=='q') || (c=='Q') ||  c==27 )
       return;
       }
    delay(1);
    }  /* end of while   */
  }   /* end of for loop*/
}
```

4.1.2 DEMO2 D/I for ISO-P32C32

```
/*---------------------------------------------------------------------*/
/* Demo 2: Digital input of ISO-P32C32                                 */
/* Step 1: The circuit diagram of hardware: refer to Sec 3.1           */
/* Step 1: run demo2.EXE  under DOS                                    */
/* --------------------------------------------------------------------*/

#define wBase    0x200
#include <dos.h>

void main()
{
unsigned char j1,j2,j3,j4;
char c;

clrscr();

/* step 1: digital input from DI_0 to DI_31                */
while(1)
{

for ( ; ; )
    {
   printf("\n\n --------- Digital input of ISO-P32C32 ------------");
      j1=inp(wBase+0)^0xff;   /* DI_07 to DI_00 of CON2 */
      j2=inp(wBase+1)^0xff;   /* DI_15 to DI_08 of CON2 */
      j3=inp(wBase+2)^0xff;   /* DI_23 to DI_16 of CON1 */
      j4=inp(wBase+3)^0xff;   /* DI_31 to DI_24 OF CON1 */

 printf("\nD 31-0 Input  Value = %02x,%02x,%02x,%02x",j4,j3,j2,j1);
       sleep(1);

    if (kbhit()!=0)
       {
       c=getch();
       if ((c=='q') || (c=='Q') ||  c==27 )
       return;
       }
    delay(1);
    }  /* end of while */
  } /* end of for */

}
```

4.1.3 DEMO3 D/I/O for ISO-P32C32

```
/* ---------------------------------------------------------------*/
/* Demo 3: Digital I/O test by itself for ISO-P32C32              */
/* Step 1: The circuit diagram of hardware: refer to Sec 3.1      */
/* Step 2: run demo3.EXE  under DOS                               */
/* ---------------------------------------------------------------- */

#define  wBase     0x200
#include <dos.h>

void main()
{

unsigned char i,j1,j2,j3,j4;
char c;

clrscr();

/* step 1: read from DI_0 to DI_31 of CON2 and CON1 */
while(1)
{
printf("\n --------- ISO-P32C32 test by itself ------------");
for (i=1;i<=0x80;i=i<<1)
    {

      outp(wBase+0,i);        /* DO_07 to DO_00 of CON2 */
      outp(wBase+1,i);        /* DO_15 to DO_08 of CON2 */
      outp(wBase+2,i);        /* DO_23 to DO_16 of CON1 */
      outp(wBase+3,i);        /* DO_31 to DO_24 of CON1 */
      sleep(1);
      j1=inp(wBase+0)^0xff;   /* DI_07 to DI_00 of CON2 */
      j2=inp(wBase+1)^0xff;   /* DI_15 to DI_08 of CON2 */
      j3=inp(wBase+2)^0xff;   /* DI_23 to DI_16 of CON1 */
      j4=inp(wBase+3)^0xff;   /* DI_31 to DI_24 of CON1 */


 printf("\nD 31-0 Output Value = %02x,%02x,%02x,%02x",i,i,i,i);
 printf("\nD 31-0 Input  Value = %02x,%02x,%02x,%02x\n",j4,j3,j2,j1);

 if( i != j1 )
   {
    printf("\nDO_7-DO_0 error in here!\n");
    putch(0x07);  putch(0x07);  putch(0x07);
   }
    if( i != j2 )
        {
         printf("\nDO_15-DO_8  error in here!\n");
         putch(0x07);  putch(0x07);  putch(0x07);
        }
     if( i != j3 )
        {
         printf("\nDO_24-DO_16 error in here!\n");
         putch(0x07);  putch(0x07);  putch(0x07);
        }
        if( i != j4 )
        {
         printf("\nDO_31-DO_25 error in here!\n");
         putch(0x07);  putch(0x07);  putch(0x07);
        }

  if(i==j1 & i==j2 & i==j3 & i==j4)
  {
```

```
   printf("The Digital I/O test of PISO-P32C32 by itself OK!\n");
  }

 if(i==0x80)  { i=0x01; break; }

    if (kbhit()!=0)
       {
       c=getch();
       if ((c=='q') || (c=='Q') ||  c==27 )
       return;
       }
    delay(3);
    }  /* end of while   */
  }
}
```

4.1.4 P32Cint.exe Interrupt input for ISO-P32C32

```
/*-----------------------------------------------------------------*/
/* P32Cint.exe: Win32 Console-Mode program used Interrupt function */
/* P32Cint.cpp used Visual C++ compiler                            */
/* Step 1: The circuit diagram of hardware: refer to Sec 3.1       */
/* Step 2: Run P32Cint.exe under Win95/98                          */
/*-----------------------------------------------------------------*/
#include <windows.h>
#include <iostream.h>
#include <stdio.h>
#include "dio.h"

WORD    wBase = 0x200;
WORD    nIRQ  = 5;
WORD    wApcCount=0;

/****************** Function Registered To Vxd *******************/
DWORD  WINAPI   RegisterApcFunction(PVOID param)
{
    //--------- Do you action in this function ------------
    wApcCount++;
    return 0;
}

int MyErr(char* s)
{
    cout << s << endl ;
    cout << "Press <Enter> key to exit." << endl;
    getchar();
    return 1;
}

int main(void)
{
cout<< "Demo program for Interrupt of ISO-P32C32"<< endl;
cout<<"****************************************" << endl;
cout<<"Wire connection....." << endl;
cout<<"Setting JP1, JP2 to Internal Power" << endl;
cout<<"Connect DO0   with DI0 => CON2.Pin21 with CON2.Pin2"<< endl;
cout<<"Connect DO16  with DI16 => CON1.Pin21 with CON1.Pin2"<< endl;
cout<<"Connect DI GND with DO GND =>CON2.Pin1 with CON2.Pin19"<<endl;
cout<<"Connect DI GND with DO GND =>CON1.Pin1 with CON1.Pin19"<<endl;
cout<< "****************************************" << endl << endl ;

cout << "Please input the Base-Address of I/O port(i.e 0x200): ";
cin  >> wBase ;
cout << endl << "Please input the IRQ number(i.e 5) : ";
cin  >> nIRQ ;

    char buffer[20];
   _itoa( wBase, buffer, 16 );
cout<<endl<< "The Base-Address of I/O port is: 0x" << buffer << endl;
cout << "The IRQ number is: " << nIRQ << endl;

cout<<endl<< "Press <Enter> key to test the interrupt, "     << endl;
cout<<"or press <Ctrl> + <c> key to exit program."           << endl;
    getchar();


    wApcCount = 0 ;
    if( DIO_DriverInit() != 0 )
        return MyErr( "Driver Initialize Error!!" );
```

```
    else
        cout << "Driver Initialize OK!!" << endl;

    if( DIO_RegisterApc(RegisterApcFunction) != 0 )
        return MyErr( "Register APC  Error!!" );
    else
        cout << "Register APC OK!!" << endl;

    if ( DIO_InstallIrq(wBase,static_cast<WORD>( nIRQ ) ) != 0 )
        return MyErr( "Interrupt Install Error!!" );
    else
        cout << "Interrupt Install OK!!" << endl;

    DIO_ResetIntCount();

    for (int i=0;i<10;i++)
    {

// ****************************************************************
// The user could replace the internal trigger by external trigger.
// ****************************************************************
        cout << "    Trigger interrupt" << endl;
// Make the DO16 to high then low to generate the interrupt signal
// for IRQ3,4,5,6,7 Connect the DO16 with DI16(Interrupt Source) to
// enable Interrupt
        DIO_OutputByte( (wBase + 2), 1);
        Sleep( 100 );
        DIO_OutputByte( (wBase + 2), 0);
        Sleep( 100 );

// Make the Output-Channel-0 to high then low to generate the
// interrupt signal for IRQ9,10,11,12,14,15 Connect the DO0 with
// DI0(Interrupt Source) to enable Interrupt
        DIO_OutputByte( wBase, 1 );
        Sleep( 100 );
        DIO_OutputByte( wBase, 0 );
        Sleep( 100 );

cout << "Waiting for an asynchronous procedure call(APC) is queued to
         the thread." << endl;
        SleepEx(INFINITE ,TRUE);
        cout << "    Got the APC!!" << endl << endl;
        //***** Do some thing in here ****
        //   :
    }

    WORD   dwIntCount;
    //*** How many times the interrupt occur ***
    if( DIO_GetIntCount(&dwIntCount) != 0 )
        return MyErr( " Get Interrupt Counter Error !!!" );
    else
        cout << "Interrupt Counter: " << dwIntCount << endl;

    DIO_DriverClose();
    cout << "Driver Close" << endl;

    return (1 - MyErr( "" ));
 }
```

4.1.5 DEMO1 D/I for ISO-P64

```
/* ---------------------------------------------------------------- */
/* Demo 1: Digital Input of ISO-P64                                 */
/* Step 1: The circuit diagram of hardware: refer to Sec 3.2        */
/* Step 2: run demo1.EXE                                            */
/* ---------------------------------------------------------------- */

#define  wBase    0x200
#include <dos.h>

void main()
{
char c;
unsigned char i,r1,r2,r3,r4,r5,r6,r7,r8;

clrscr();

/* Digital input from DI_0 to DI_63 */
while(1)
{

for ( ; ;)
    {
     printf("\n--------- Digital input of ISO-P64  ------------");
     r1 =inp(wBase+0);       /* DI_07 to DI_0  */
     r2 =inp(wBase+1);       /* DI_15 to DI_08 */
     r3 =inp(wBase+2);       /* DI_23 to DI_16 */
     r4 =inp(wBase+3);       /* DI_31 to DI_24 */

     r5 =inp(wBase+4);       /* DI_39 to DI_32 */
     r6 =inp(wBase+5);       /* DI_47 to DI_40 */
     r7 =inp(wBase+6);       /* DI_55 to DI_48 */
     r8 =inp(wBase+7);       /* DI_63 to DI_56 */

      printf("\nThe CON1 of ISO-P64 ...");
      printf("\nD31- 0 Input Value =
%02x,%02x,%02x,%02x\n",r4,r3,r2,r1);

      printf("\nThe CON2 of ISO-P64 ...");
      printf("\nD63-32 Input Value =
%02x,%02x,%02x,%02x\n",r8,r7,r6,r5);
      sleep(1);

    if (kbhit()!=0)
       {
       c=getch();
       if ((c=='q') || (c=='Q') ||  c==27 )
       return;
       }
    delay(1);
    }  /* end of while    */
  }   /* end of for loop */
}
```

4.1.6 DEMO1 D/O for ISO-C64

```
/* ---------------------------------------------------------------- */
/* Demo 1: Digital Output of ISO-C64                                */
/* Step 1: The circuit diagram of hardware: refer to Sec 3.3        */
/* Step 2: run demo1.EXE                                            */
/* ---------------------------------------------------------------- */
#define  wBase    0x200
#include <dos.h>

void main()
{
char c;
unsigned char i;

clrscr();

/* Digital output from DO_0 to DO_63 */
while(1)
{
printf("\n\n --------- Digital output of ISO-C64  ------------");
for (i=1;i<=0x80;i=i<<1)
    {
      outp(wBase+0,i);        /* DO_07 to DO_00 */
      outp(wBase+1,i);        /* DO_15 to DO_08 */
      outp(wBase+2,i);        /* DO_23 to DO_16 */
      outp(wBase+3,i);        /* DO_31 to DO_24 */
      outp(wBase+4,i);        /* DO_39 to DO_32 */
      outp(wBase+5,i);        /* DO_47 to DO_40 */
      outp(wBase+6,i);        /* DO_55 to DO_48 */
      outp(wBase+7,i);        /* DO_63 to DO_56 */

      printf("\nThe CON1 of ISO-C64 ...");
      printf("\nD31- 0 Output Value = %02x,%02x,%02x,%02x\n",i,i,i,i);

      printf("\nThe CON2 of ISO-C64 ...");
      printf("\nD63-32 Output Value = %02x,%02x,%02x,%02x\n",i,i,i,i);
      sleep(1);

 if(i==0x80)  { i=0x01; break; }

    if (kbhit()!=0)
       {
       c=getch();
       if ((c=='q') || (c=='Q') ||  c==27 )
       return;
       }
    delay(1);
    }  /* end of while    */
  }   /* end of for loop */
}
```