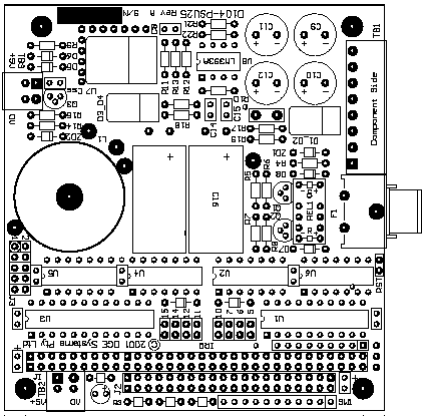
D104-PCS20

5V/ 4Amp Power Supply Module for PC/104 Systems

Rev B
26th March 2001



DGE Systems Pty Ltd
103 Broadmeadow Road
Broadmeadow, NSW
Australia

www.dge.com.au

Intentionally Blank

Contents

Tables

Figures

Appendix

Introduction

The D104-PCS20 provides a regulated 5 volts for PC/104 systems. It has the same form factor as PC/104 modules and is designed to bolt onto the PC/104 stack.

The module accepts a wide ranging DC input voltage allowing the D104-PCS20 to operate from control power voltages and from vehicular voltages. An auxiliary power input is provided for connection of a standby or backup power source.

A design feature of the D104-PCS20 is the addition of control and status circuitry which allows system power down under software control. Under interrupts or through register polling, application programs are able to detect a power off request or when the system is being powered by a backup supply. This would typically be used to allow the proper shutdown of the system.

A source of regulated 5Vdc is available at two terminal block sites for powering external equipment.

A "Power Good" LED indicator is adjacent to the PC/104 connectors.

Features of the D104-PCS20 include:

- **PC/104 Compliant**
- **5V at 4 Amps output**
- **10 - 30Vdc input range**
- **Software controlled power off**
- **Status by poll or interrupt**
- **5 digital outputs**

Basic Power Supply Unit

The D104-PCS20 is available as a basic 5V/4A power supply without the interface logic by specifying an order code of D104-PCS20-X.

Primary and Auxiliary DC Sources

Two power sources may be connected to the D104-PCS20 - a primary source and an auxiliary source.

The auxiliary source may be a short term supply to allow an orderly system shutdown or to retain system power for intermittent disconnection of the primary source. To ensure a clean and swift take up of power from this source, it is suggested a battery be used. The auxillary supply should be at a lower voltage than the primary supply. Some situations may need a hold-up capacitor if the auxiliary supply is slow to start. The capacitor may be fitted between pin 8 (V+) and pin 6 (0V) of terminal block TB1.

The sources are diode isolated which also provides reverse polarity protection to the power supply.

Note: The D104-PCS20 does not provide any charging services for the auxiliary source.

Power On/Off Switch for Controlled Shutdown

Users wishing to implement controlled shutdown will connect a switch to the main terminal block TB1, between terminals 7 and 8. This may be a manual switch or it may be an ignition switch in the case of vehicular mounted systems. On engagement of the switch, a latching relay is set and power is delivered to the system.

When the switch is disengaged an event is caused rather than power being switched off. An application program can detect this event and proceed with an orderly shutdown by closing files and programs and then issuing a shutdown signal which resets the latching relay, removing system power.

Either alternate action switch or momentary action switch can be used. For momentary switches the power off event occurs on the second release of the switch.

Systems not requiring a power switch will need to have a wire link in place of the switch. This puts the latching relay in a permanent set mode.

Terminal Block Connections

TB1 is an eight way two-part terminal block providing a convenient connect/disconnect of the DC sources and switch. A SPST contact is available at pin 1 of the terminal block which connects to the 0V return when power is on. This may be used for indicator lamps or for switching external devices.

Terminal Block TB1		
Pin	**Signal**	**Function**
1	Contact	Grounded when PSU is on.
2	0V	Common return
3	V_{IN}	Primary DC input
4	0V	Common return
5	V_{STBY}	Backup DC input
6	0V	Common return
7	Switch	Alternate action or momentary action switch connected across pin 7 and pin 8
8	V_{PWR}	

TB2 and TB3 provide take off points for powering external equipment. TB2 is limited on the amount of deliverable current due to size of the PCB tracking.

Terminal Blocks TB2 and TB3		
Pin	**Signal**	**Function**
1	+5Vdc	DC for external supply
2	0V	Common return

Table 1: Terminal Block Connections

System Bus Interface

The D104-PCS20 has I/O mapped control and status registers occupying a single location in the I/O address space. The registers' address is set by a ten position SIP switch. This switch is conveniently placed near a PCB edge such that the address can be set even with the module mounted on the PC/104 stack.

Setting the Module's I/O Address

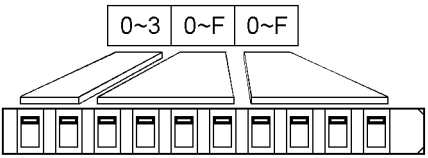


Figure 1: I/O Address SIP Switch

Figure 1 shows the SIP switch configuration for setting the module's I/O address which can range from 000H to 3FFH. Users should ensure the setting does not conflict with I/O addresses of other devices.

Setting the Module's IRQ

When interrupts are to be implemented, the user has the choice of a number of IRQ settings. Figure 2 shows the link positions. Users must ensure that if interrupts are to be used that there is no conflict with IRQ settings on other devices.

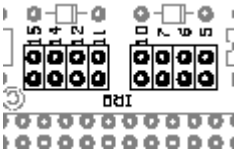


Figure 2: IRQ Link Setting

Control Register

The D104-PCS20 is able to signal two events: a power off event and when the system is running from an auxiliary power supply. These events can optionally cause an interrupt if the appropriate interrupt enable bit is asserted and an IRQ line is selected. A '1' written in these bits enables interrupts.

Bit 2 is the power off signal, a '1' written to bit 2 will reset the latching relay, which subsequently shuts down the 5V supply.

This register is a write-only register, there is no readback facility.

Control Register		
Bit	**Signal**	**Meaning**
0	BPIE	Backup Power Interrupt Enable
1	POIE	Power Off Interrupt Enable
2	POFF	Switch Power Off
3	DO0	Digital Output 0, pin 1 of J3
4	DO1	Digital Output 1, pin 2 of J3
5	DO2	Digital Output 2, pin 3 of J3
6	DO3	Digital Output 3, pin 4 of J3
7	DO4	Digital Output 4, pin 5 of J3

Table 2: Control Register Bits

Status Register

The status register has only two valid bits - bit 0 and bit 1. These bits will be asserted when a power off signal occurs or when the system is being powered from an auxiliary source. By polling the status register, applications programs can determine if a power off event or if the auxiliary power supply is in use. An I/O read of this register will clear these bits.

Status Register		
Bit	**Signal**	**Meaning**
0	BP	Backup Power In Use
1	PO	Power Off Signal Asserted
2	X	Don't Care
3		
4		
5		
6		
7		

Table 3: Status Register Bits

Digital Outputs

The otherwise unused bits within the control register are mapped to 10 pin header J3 and can be considered for use as general purpose digital output lines. Software must ensure bits 0, 1 and 2 remain undisturbed whenever the digital output states are set.

Control Register	J3 Pin Assignment			
	Signal	Pin	Pin	Signal
Bit 3	D0	1	2	Gnd
Bit 4	D1	3	4	
Bit 5	D2	5	6	
Bit 6	D3	7	8	
Bit 7	D4	9	10	

Table 4: J3 Pin Assignment

Interrupts

If programmed to do so, the D104-PCS20 is able to generate an interrupt on two events. An event will occur when the power switch is deactivated. This allows software to detect the operator's intention of switching off and proceed with an orderly system shutdown before signalling the D104-PCS20 to remove power.

Another event occurs if the primary supply's terminal voltage is less than the auxillary supply's terminal voltage Application programs are able to detect this event and take appropriate action. Note: Interrupts are not possible for this event if the power switch is a momentary action type switch as the interrupt line is asserted, masking out the signal . However, software can still detect this event by polling the status register.

Power Up Reset

On power up the control register is cleared, interrupts are disabled and digital outputs set low. The status register bits are also cleared.

Component Layout

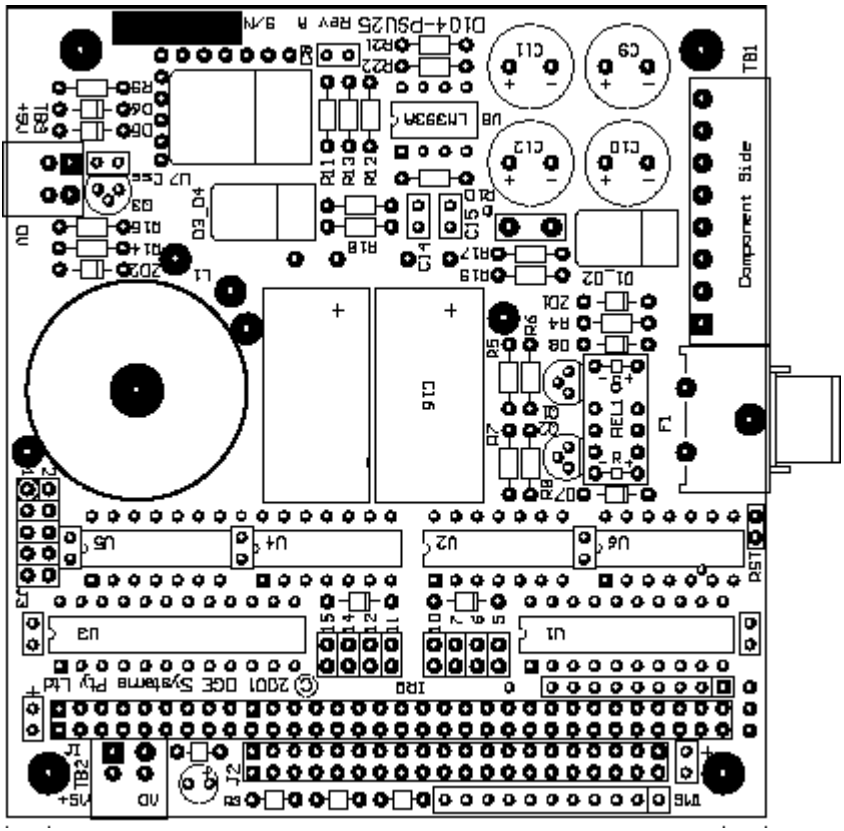


Figure 4: Component Layout

Connector Pin Assignments

Connector J1/P1 Pin Assignments					
Signal		**Pin**	**Pin**		**Signal**
/IOCHCHK		1a	1b	¤	0V (Gnd)
SD7	¤	2a	2b	¤	RESETDRV
SD6	¤	3a	3b	¤	+5V (VCC)
SD5	¤	4a	4b	¤	IRQ2/9
SD4	¤	5a	5b		-5V
SD3	¤	6a	6b		DRQ2
SD2	¤	7a	7b		-12V
SD1	¤	8a	8b		/ENDXFR
SD0	¤	9a	9b		+12V
IOCHRDY		10a	10b		(KEY)
AEN	¤	11a	11b		/SMEMW
SA19		12a	12b		/SMEMR
SA18		13a	13b	¤	/IOW
SA17		14a	14b	¤	/IOR
SA16		15a	15b		/DACK3
SA15		16a	16b		DRQ3
SA14		17a	17b		/DACK1
SA13		18a	18b		DRQ1
SA12		19a	19b		/REFRESH
SA11		20a	20b		CLK
SA10		21a	21b	¤	IRQ7
SA9	¤	22a	22b	¤	IRQ6
SA8	¤	23a	23b	¤	IRQ5
SA7	¤	24a	24b		IRQ4
SA6	¤	25a	25b		IRQ3
SA5	¤	26a	26b		/DACK2
SA4	¤	27a	27b		TC
SA3	¤	28a	28b		BALE
SA2	¤	29a	29b	¤	+5V
SA1	¤	30a	30b		OSC
SA0	¤	31a	31b	¤	0V (Gnd)
0V (Gnd)	¤	32a	32b	¤	0V (Gnd)

¤ Signals used on the D104-PCS20

Connector J2/P2 Pin Assignments					
Signal		**Pin**	**Pin**		**Signal**
0V (Gnd)	¤	0c	0d	¤	0V (Gnd)
/SBHE		1c	1d		/MEMCS16
LA23		2c	2d		/IOCS16
LA22		3c	3d		IRQ10
LA21		4c	4d	¤	IRQ11
LA20		5c	5d	¤	IRQ12
LA19		6c	6d	¤	IRQ15
LA18		7c	7d	¤	IRQ14
LA17		8c	8d		/DACK0
/MEMR		9c	9d		DRQ0
/MEMW		10c	10d		/DACK5
SD8		11c	11d		DRQ5
SD9		12c	12d		/DACK6
SD10		13c	13d		DRQ6
SD11		14c	14d		/DACK7
SD12		15c	15d		DRQ7
SD13		16c	16d	¤	+5V (VCC)
SD14		17c	17d		/MASTER
SD15		18c	18d		0V (Gnd)
(KEY)		19c	19d		0V (Gnd)

Table 5: Connector J1/P1 and J2/P2 Pin Assignments

Specifications

D104-PCS20 Specifications	
Input:	10 - 36Vdc,
Output:	5Vdc at 4 Amps *.
Fuse:	2Amp blade fuse
Efficiency at Full Load: 12V 24V	 78% 82%
Line Regulation:	4.0mV ±0.04%, 10V - 36V, at 4A
Load Regulation:	1.0mV ±0.01%, 12Vin, 0.25A - 4A
Short Circuit Current:	6.5A
Registers:	1 read only (status), 1 write only (control)
Interrupts:	Selectable on IRQ5, 6, 7, 9 ,11, 12, 14, 15
Digital Outputs:	5 TTL level outputs, 32mA source, 64mA sink
Temperature: Operating: Storage:	 -20 to +70ºC. -55 to +100ºC.
Humidity:	5% to 95% non condensing.
Dimensions:	90 x 96mm (3.55 x 3.775").
Weight:	100g approx.

* Note:
The D104-PCS20 is capable of short burst current levels up to 5 amps

Typical Input Power Requirements

Supply Voltage	Load Current 1A	2A	3A
12V	0.57	1.10	2.50
18V	0.35	0.65	1.00
24V	0.28	0.50	0.80

Ordering Information

Part Number	Description
D104-PCS20	5V / 4A PC/104 Power Supply Module
D104-PCS20-X	Basic PSU without interface circuitry
D104-PCS20-TM	Technical Manual

Sample Program

This DOS based program shows how the D104-PCS20 operates under interrupts. At each backup-power event or power-off event an interrupt occurs.

The interrupt handler simply increments counters and the main program display these count values.

```
/*
 *  pcs20.c
 *
 *  Test routine for the D104-PCS20 power supply
 *  module for PC/104 systems.
 *
 *  Two registers are used, which occupy one
 *  address in the I/O address map:-
 *
 *  Control Register
 *
 *      Bit 0:      Enable backup power interrupt
 *      Bit 1:      Enable power off interrupt
 *      Bit 2:      Switch power off
 *      Bits 3 - 7 Mapped to 10 pin header for digital out
 *      Access:   Write only, no readback facility
 *
 *  Status Register
 *
 *      Bit 0:      Backup power in use
 *      Bit 1:      Power off event occured
 *      Bits 2 - 7:Don't care
 *      Access:   Read only
 *
 *  Built and tested with MicroSoft's C/C++ V7.0.
 */

#include  <stdio.h>
#include  <conio.h>
#include  <stdlib.h>
#include  <string.h>
#include  <dos.h>
```

```
typedef struct
{
    unsigned int   addr;
    unsigned int   irq;
    union
    {
        unsigned char  byte;
        struct
        {
            unsigned    bpie        :1, // bp interrupt enable
                        poie        :1, // po int enable
                        shutdown    :1, // shutdown power
                        d0          :1, // digital outputs
                        d1          :1,
                        d2          :1,
                        d3          :1,
                        d4          :1;
        } bit;
    } control;
    union
    {
        unsigned char  byte;
        struct
        {
            unsigned    bp          :1,
                        po          :1,
                        unused      :6;
        } bit;
    } status;
    int             bpcnt;    // count of bp interrupts
    int             pocnt;    // count of po interrupts
    unsigned int    intnum;
    unsigned int    pic;      // 8259 prog. int controller
    void  __far     *isr;     // pointer to PCS20's isr
    void  __far     *oldisr;
    int             bell;
} PCS20;

// globals
PCS20               pcs20;    // global for interrupt handler

void                __interrupt __far PCS20Isr();
void                __far *IsrInstall( PCS20 *p    );
int                 IsrRemove( PCS20 *p );
unsigned int        ahex2i( char *str );
```

```
//.....................................................................

int main( int argc, char *argv[] )
{
    int      i, kbchar;

    pcs20.addr = 0x0330;      // factory default

    for ( i = 1; i < argc; i++ )
    {
        if ( argv[i][0] == 'A' || argv[i][0] == 'a' )
        {
            pcs20.addr = ahex2i( &argv[i][1] );
            if ( pcs20.addr < 0 || pcs20.addr > 0x3FF )
                pcs20.addr = 0x330;
        }

        if ( argv[i][0] == 'I' || argv[i][0] == 'i' )
        {
            pcs20.irq = atoi( &argv[i][1] );
            if ( pcs20.irq < 0 || pcs20.irq > 15 )
                pcs20.irq = 0;
        }
    }

    system( "Cls" );
    printf( "Test program for D104-PCS20" );
    if ( pcs20.irq == 0 )
    {
        printf( "\n\n    Invoke as: PCS20 axxx inn\n\n" );
        printf( "\twhere xxx is I/O address in hex ( 0h - 3FFh)\n" );
        printf( "\t       nn is  irq in decimal (1 - 15)\n" );
        return( -1 );
    }

    printf( " at Addr: %03Xh, Irq: %d\n\n", pcs20.addr, pcs20.irq );
    printf( "    Esc      Quit program without shut down\n\n" );

    printf( "    BP Interrupt       PO Interrupt\n" );
    printf( "       Count             Count\n" );
    printf( "    ----------------    ----------------\n" );
    printf( "       xxxxx             xxxxx\n" );

    // reset the interrupt latch on the PCS20
    inp( pcs20.addr );

    // set up the isr
    pcs20.isr = PCS20Isr;
    pcs20.oldisr = IsrInstall( &pcs20 );

    // enable interrupts
    pcs20.control.byte = 0;                          // clear all bits
    pcs20.control.bit.bpie = pcs20.control.bit.poie = 1;
    _outp( pcs20.addr, pcs20.control.byte );   // enable interrupts

    kbchar = 0;
    do
    {
        printf( "      %5d              %5d              %02X\r",
                    pcs20.bpcnt,
                    pcs20.pocnt,
                    ( inp( pcs20.addr) & 0x03 ) );

        if ( kbhit() )
            kbchar = getch();
    } while ( kbchar != 0x1B );

    // disable interrupts, remove the isr
    pcs20.control.bit.bpie = pcs20.control.bit.poie = 0;
    _outp( pcs20.addr, pcs20.control.byte );
    IsrRemove( &pcs20 );

    // to switchoff  the PSU
    // pcs20.control.bit.shutdown = 1;
    // _outp( pcs20.addr, pcs20.control.byte );

    printf( "\n\n    End Pcs20 Program\n" );

    return( 0 );
}
```

```
//.....................................................................
//
//   Install interrupt service routine (ISR) on IRQ.
//
//   Returns pointer to original ISR.
//

#define   PIC1  0x20       // 8259 prog int controller I/O address
#define   PIC2  0xA0

void __far    *IsrInstall( PCS20 *p )
{
    unsigned char       picval;
    void                (__interrupt __far *oldisr)();

    if ( p->irq > 7 )
    {
        p->intnum = p->irq + 0x70 - 8;
        p->pic = PIC2;
    }
    else
    {
        p->intnum = p->irq + 8;
        p->pic = PIC1;
    }

    oldisr = _dos_getvect( p->intnum );      // save current isr
    _disable();
    _dos_setvect( p->intnum, p->isr );       // our int handler
    _enable();

    // read Interrupt Mask Register
    picval = ( unsigned char )inp( p->pic + 1 );

    // clear bit mask (enable interrupts on the IRQ)
    outp( p->pic + 1, picval &
                ~( 1 << ( ( p->irq > 7 ) ? p->irq - 8 : p->irq ) ) );

    return ( oldisr );
}

//.....................................................................
//

int IsrRemove( PCS20 *p )
{
    unsigned char       picval;

    picval = ( unsigned char )inp( p->pic + 1);

    // set IRQ's mask bit (disable interrupts)
    outp( p->pic + 1, picval |
                ( 1 << ( (p->irq > 7 ) ? p->irq - 8 : p->irq ) ) );

    // restore original handler
    _disable();
    _dos_setvect( p->intnum, p->oldisr );
    _enable();

    return (0);
}
```

```
//.....................................................................
//
//   Interrupt routine.
//
//   This routine simply increments counters for the main
//    program to display.
//
//   Rather than act instantly to a single interrupt, user
//   applications may want to take action after a number
//   of interrupts have occured by checking these
//   counter values.
//

#pragma check_stack( off )

void __interrupt __far  PCS20Isr()
{
    _disable();

    // read status register,
    // also resets interrupt flip-flops on PCS20
    pcs20.status.byte = inp( pcs20.addr);

    if ( pcs20.status.bit.bp )
        ++pcs20.bpcnt;

    if ( pcs20.status.bit.po )
        ++pcs20.pocnt;

    if ( pcs20.irq > 7 )
    {
        // send EOI to pic 2 for IRQs 8 to 15
        __asm
        {
            mov  al, 020h
            out  0A0h, al
        }
    }

    // send EOI to pic 1 for IRQs 0 to 15
    __asm
    {
        mov  al, 020h
        out  020h, al
    }

    _enable():
}

#pragma check_stack( on )
```

```
//.....................................................................
//
//   Convert string with hex characters to an integer.
//

unsigned int ahex2i( char *str )
{
    unsigned int      num, mult;
    char              *cp;

    strupr( str );
    cp = str;
    while ( *cp )
        cp++;

    mult = 1;
    num = 0;
    do
    {
        —cp;
        if ( *cp >= 'A' && *cp <= 'F' )
            num += ( *cp - 'A' + 10 ) * mult;
        else if ( *cp >= '0' && *cp <= '9' )
            num += ( *cp - '0' ) * mult;
        else
            return( 0 );            // invalid hex character
        mult *= 16;
    } while ( cp > str ) ;

    return( num );
}

// end pcs20.c .....................................................
```

Other Products in the D104 Series of PC/104 Modules

D104-MO32

The D104-MO32 provides 32 optically isolated, single pole, normally open (form A), MOSFET relay outputs. The relay contacts are capable of switching both AC and DC loads. The loads can be commoned on either the positive side or negative side of the power source. A feature of the D104-MO32 is the ability for the MOSFET relays to act as an analog switch allowing low level analog signals and medium power audio levels to be controlled.

The module is available in two models, a low-voltage/high-current and a high-voltage/low-current version which makes the D104-MO32 suitable for a wide range of applications.

D104-RO32C

The D104-RO32C provides 32 electromechanical relay outputs. The relay contacts are single pole, changeover (form C). Typical applications for the D104-RO32C include light power switching and signal multiplexing/demultiplexing.

D104-RO16A

16 medium power, electromechanical relay outputs. The relay contacts are single pole, single throw (SPST, form A) which are capable of switching 5 Amps at 250Vac. The module can optionally be fitted with SPST MOSFET devices for solid-state switching of 600mA/250V ac/dc loads.

The module is fitted with heavy-duty, two-part detacheable terminal blocks for quick connect/disconnect of field wiring.

D104-DIO64

The D104-DIO64 provides 64 lines of digital I/O for systems based on the PC/104 format. The digital I/O is configured as eight ports of eight lines with each port individually jumper programmable for input only or for input/output. It has industry-standard connectors for attaching solid-state modules and racks.

Buffer Modules

A series of external buffer modules providing opto-isolation of inputs, MOSFET outputs and relay outputs. The modules connect to the D104-DIO64 I/O and other form-factor I/O cards having the industry-standard pinouts.

These modules have the PC/104 footprint and can be bolted onto the PC/104 stack.

D104-PSU15

15 watt power supply on a PC/104 footprint providing 5 volts at 3 Amps.