

CPC-1631CVD2NA

6U CompactPCI Motherboard

Version: A2

Copyright Notice

Information offered in this manual is believed to be correct at the time of printing, and is subject to change without prior notice in order to improve reliability, design and function and does not represent a commitment on the part of the manufacturer. In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of improper installation and/or use, or inability to use the product or documentation.

This user manual is protected by copyright. No part of this manual may be reproduced, stored in any retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopied, recorded or otherwise, without the prior written permission from the manufacturer.

Trademarks

EVOC is a registered trademark of EVOC Intelligent Technology Co., Ltd. Other product names mentioned herein are used for identification purposes only and may be trademark and/or registered trademarks of their respective companies.

Safety Instructions

1. Please read the User Manual carefully before using your product.
2. For the board which is not ready to be installed, please put it in the anti-static packaging;
3. Before taking the board out from anti-static packaging, please put your hand on grounded metal object for a while (about 10 seconds) to discharge static;
4. Please wear static protective gloves when holding the board; and always hold the board by edges;
5. Before inserting, removing or re-configuring the motherboard or the expansion card, please firstly disconnect the AC power or unplug the AC power cable from the power source to prevent damage to the product and ensure your personal safety;
6. Before moving the boards or Box PC, firstly turn off all power resources and unplug the power cable from power source;
7. For Box PC products, when inserting or removing boards, please disconnect the AC power in advance;
8. Before connecting or disconnecting any device, make sure all power cables are unplugged in advance;
9. To avoid power on/off computer frequently, wait at least 30 seconds after turning off the computer before re-starting the computer;

C o n t e n t s

Chapter 1 Product Introduction	1
Overview	1
Environmental and Mechanical Dimensions	2
Micro-processor	2
Chipset.....	2
Memory	3
Display Function	3
IDE Function.....	3
Network Function.....	3
USB Function.....	3
I/O Port.....	3
Power Supply Feature	4
Chapter 2 Installation Guide.....	5
Dimensions.....	5
I/O Outline.....	7
Jumper Settings	9
Display Port.....	10
USB Port	10
Parallel Port and Serial Port	11
IDE Port.....	13
Network Port	14
MPC8245 COP Debugging Port.....	14
CPLD Debugging Port	15
Indicator LED.....	15
PMC Port.....	16
CompactFlash Port	18
CompactPCI Port.....	19
Appendix 1 Linux Operating System	23
Watchdog Programming Guide.....	23
I/O Mapping	29

IRQ Assignment Table	30
Appendix 2 VxWorks Operating System	31
Watchdog Programming Guide	31
I/O Mapping.....	33
IRQ Assignment Table	34

Chapter 1 Product Introduction

Overview

CPC-1631CVD2NA 6U CompactPCI motherboard is designed for high-end applications such as, information communication, network storage, network audio processing, network image processing, military. The motherboard is PICMG2.0-compliant.

The motherboard adopts Motorola 350M 8245CPU+VT82686 solution. MPC8245 is the new-released 82xx general CPU of Motorola, its advantages in size, consumption and cost performance, its strong processing ability and simple external connectors bring great convenience for the applications. The motherboard supports optional onboard 512MB SDRAM memory, 8MB Flash and 1MB Boot BIOS. It also adopts SMI712 display controller released by Silicon Motion, Inc. as graphics processor, which integrates 4MB, 8MB or 16MB graphics memory and features up to 1GB/S bandwidth, VGA display port supported. It also integrates 2-channel 10/100Mbps fast Ethernet port (which can be switched between motherboard and I/O board according to actual needs of customer); onboard 32-bit PCI bridge; integrated coprocessor realizes IPMC function and is compatible with IPMI V1.5 standard. It is designed with onboard 2-channel USB1.1 Host port and supports 2-channel expansion from rear I/O board, which is able to meet increasing requirements for USB ports; it also adopts 3-channel COM port and one parallel port.

Note: the motherboard is a system board, which is only available for system slot.

Environmental and Mechanical Dimensions

- Operating Environment
Temperature: 0 °C~55 °C;
Humidity: 5%~95%, non-condensing;
- Storage Environment:
Temperature: -25 °C~75 °C;
Humidity: 5%~95%, non-condensing;
- Dimensions: 233.35mm x 161.50mm

Micro-processor

MPC8245 is an ultra universal CPU released by Motorola, based on 32-bit Power PC architecture and features RISC and high-performance. It internally integrates a core similar with MPC603e and a peripheral control logic similar with MPC106 bridge, offering memory bus, PortX bus and standard PCI bus. MPC8245 supports two types of internal frequencies, 350MHz and 400MHz. External memory bus frequency is up to 100Mhz and PCI bus is up to 66MHz.

Chipset

VT82C686B is a high-performance south bridge offering one IDE port, one CF card and four USB 1.1 ports, supporting three serial ports, among which the COM1 on I/O board supports RS-232/RS-422/RS-485 working modes, as well as one LPT port.

Memory

Onboard 512MB SDRAM, providing stable memory data exchanging environment.

Display Function

It also adopts SMI712 display controller released by Silicon Motion, Inc. as graphics processor, which integrates 4MB, 8MB or 16MB graphics memory and features up to 1GB/S bandwidth, VGA display port supported.

IDE Function

It supports one Ultra ATA 100/66/33 standard IDE port, providing up to 100MB/s information transmitting speed and supporting one CF card.

Network Function

Onboard two PCI 10/100Mbps Ethernet port. LILED and ACTLED are the two green and dual-color LEDs by sides of Ethernet port, which indicate active and transmitting status of LAN port. You can bring the network port out from I/O board by switching if required.

USB Function

4 x USB1.1 port, meeting increasing demands of USB applications.

I/O Port

- The motherboard offers one RJ45 serial communication port (COM1)

and a group of serial communication port pins (COM3), meanwhile the I/O board offers one group of serial communication port (COM1) and you can select RS-232/RS-422/RS-485 mode by jumper settings.

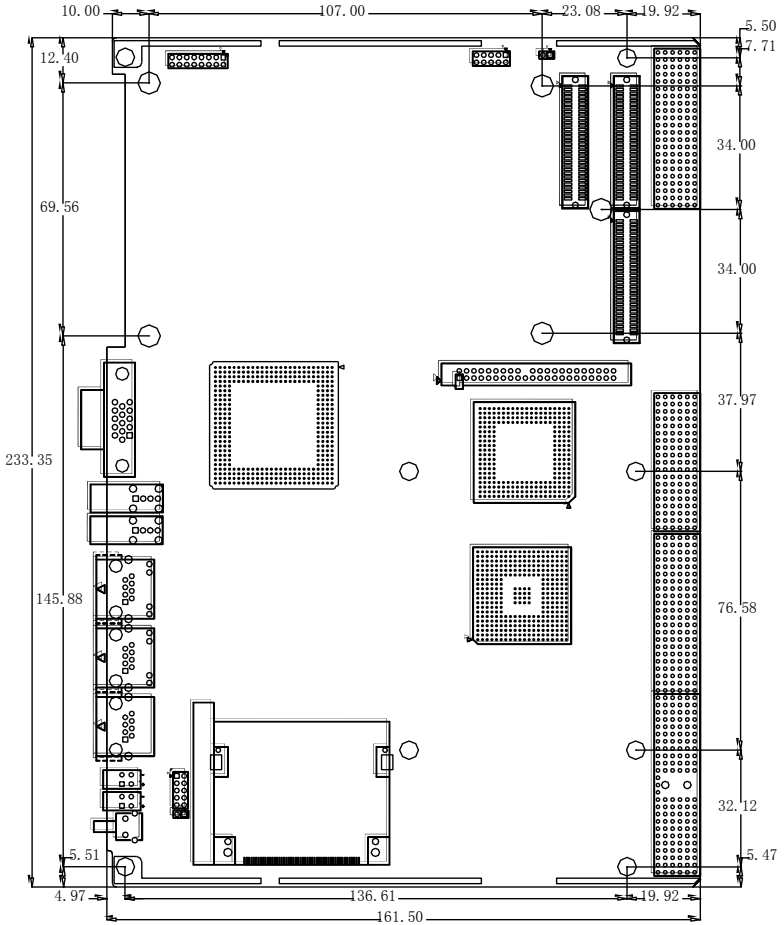
- The I/O board is configured with one LPT high-speed parallel port.

Power Supply Feature

Powered via backplane.

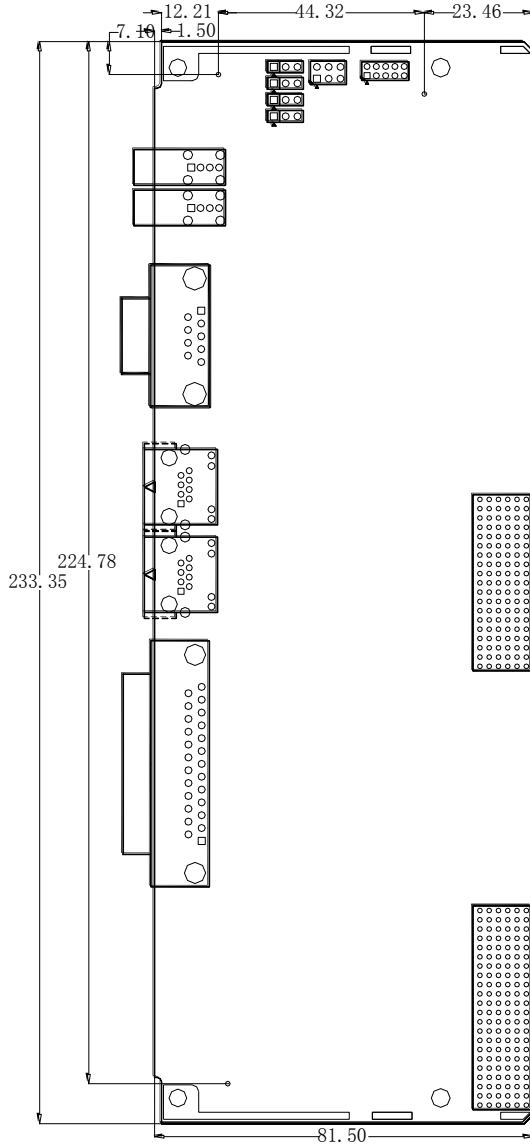
Chapter 2 Installation Guide

Dimensions



(Unit: mm)

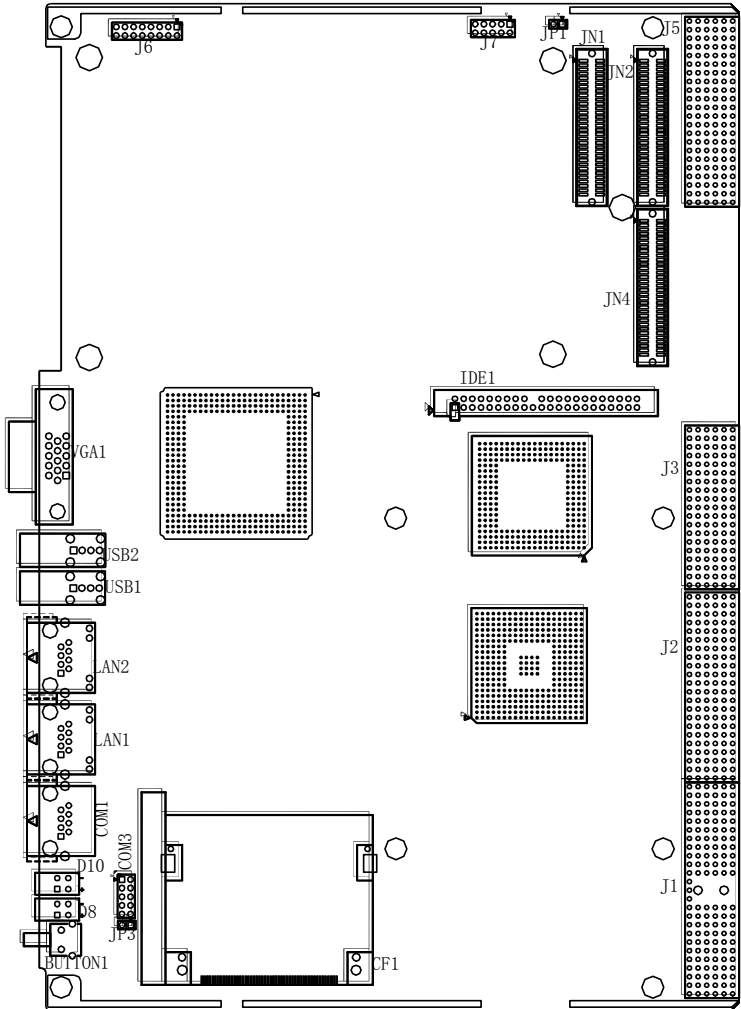
Dimensions of CPC-1631CVD2NA



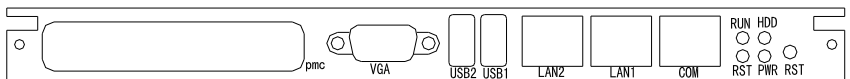
(Unit: mm)

Dimensions of CPC-RP602

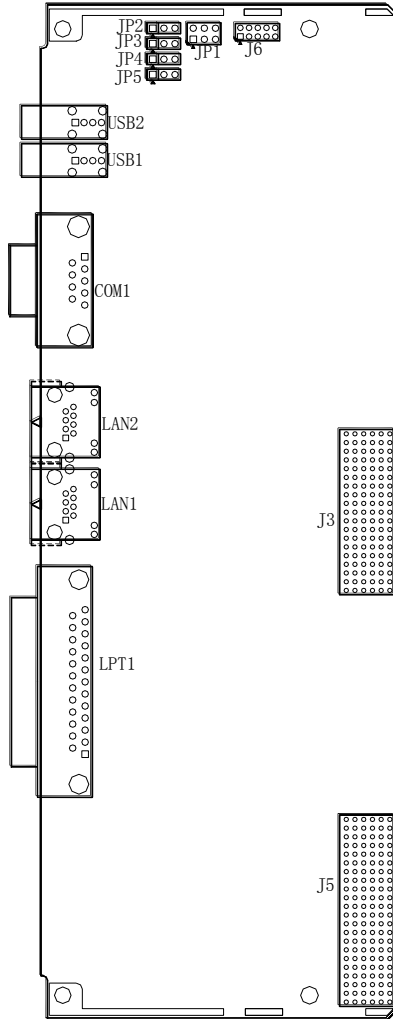
I/O Outline



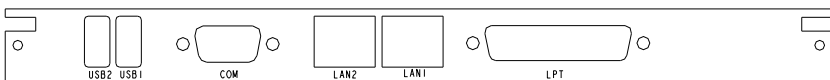
I/O outline of CPC-1631CVD2NA



I/O outline of CPC-1631CVD2NA front panel



I/O outline of CPC-RP602



I/O outline of CPC-RP602 front panel

Jumper Settings

Jumper setting of motherboard:

1. JP1: network switching pins



JP1

Setup	Function
[1-2] Open	Motherboard network port available (Default)
[1-2] Short	I/O board (CPC-RP602) network port available

2. JP3: reset pins

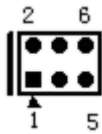


JP3

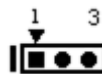
Setup	Function
[1-2] Open	(Normal, default)
[1-2] Short	Reset the motherboard

Jumper setting of I/O board:

1. JP1/JP2/JP3/JP4/JP5: COM1(CPC-RP602)RS232/RS422/RS485 mode selection



JP1

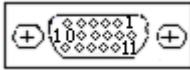


JP2/JP3/JP4/JP5

Jumper	Setup		
	RS232	RS485	RS422
JP1	1-2	3-4	5-6
JP2	1-2	2-3	2-3
JP3	1-2	2-3	2-3
JP4	1-2	2-3	2-3
JP5	1-2	2-3	2-3

Display Port

The motherboard offers one 15-pin type D VGA display socket (VGA1), connectable with all displays with VGA port.

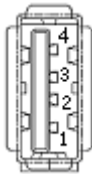


VGA1

Pin	Signal Name	Pin	Signal Name
1	Red	2	Green
3	Blue	4	NC
5	GND	6	GND
7	GND	8	GND
9	CRT_5V	10	GND
11	NC	12	DDCData
13	HSYNC	14	VSYNC
15	DDCCLK		

USB Port

Motherboard and I/O board provide two groups of USB sockets (USB1/USB2).



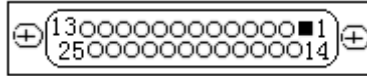
USB1/USB2

Pin	Signal Name
1	+5VUSB
2	USB Data-
3	USB Data+
4	GND

Parallel Port and Serial Port

(1) Parallel Port

The I/O board offers one 25-pin type D connector, which is available for peripherals with parallel port.

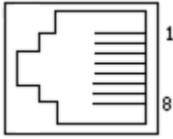


LPT1

Pin	Signal Name	Pin	Signal Name
1	STB#	14	AFD#
2	PD0	15	ERR#
3	PD1	16	INIT#
4	PD2	17	SLIN#
5	PD3	18	GND
6	PD4	19	GND
7	PD5	20	GND
8	PD6	21	GND
9	PD7	22	GND
10	ACK#	23	GND
11	BUSY	24	GND
12	PE	25	GND
13	SLCT		

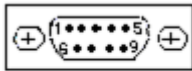
(2) Serial Port

The motherboard offers one RJ45 serial communication port (COM1) and a group of serial communication port pins (COM3), meanwhile the I/O board (CPC-RP602) offers one group of serial communication port (COM1) and you can select RS-232/RS-422/RS-485 mode by jumper settings, please refer to the jumper setting guide for (JP1/JP2/JP3/JP4/JP5) for more information.



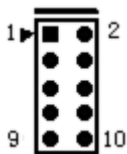
COM1 of
motherboard

Pin	Signal Name
1	RTS#
2	DTR#
3	TXD
4	GND
5	GND
6	RXD
7	DSR#
8	CTS#



IO板COM1

Pin	Signal Name		
	RS-232	RS-422	RS-485
1	DCD#	TXD-	Data-
2	RXD	TXD+	Data+
3	TXD	RXD+	X
4	DTR#	RXD-	X
5	GND	GND	GND
6	DSR#	X	X
7	RTS#	X	X
8	CTS#	X	X
9	RI#	X	X

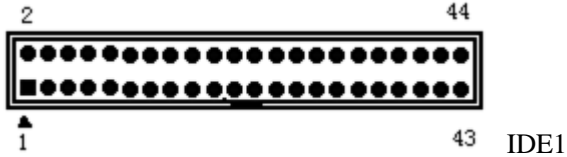


COM3

Pin	Signal Name	Pin	Signal Name
1	NC	2	RXD
3	TXD	4	NC
5	GND	6	NC
7	NC	8	NC
9	NC	10	NC

IDE Port

The motherboard offers one group of 44-pin IDE port (IDE1).

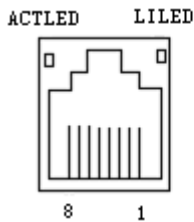


Pin	Signal Name	Pin	Signal Name
1	RESET#	2	GND
3	D7	4	D8
5	D6	6	D9
7	D5	8	D10
9	D4	10	D11
11	D3	12	D12
13	D2	14	D13
15	D1	16	D14
17	D0	18	D15
19	GND	20	Key
21	DREQ	22	GND
23	IOW#	24	GND
25	IOR#	26	GND
27	IORDY	28	GND
29	DACK#	30	GND
31	IRQ	32	NC
33	DA1	34	ATA66_DET
35	DA0	36	DA2
37	CS1#	38	CS3#
39	LED#	40	GND
41	+5V	42	+5V
43	GND	44	GND

Network Port

The motherboard and I/O board integrate a 10/100Mbps Ethernet port. LILED and ACTLED are the two green and yellow LEDs by sides of Ethernet port indicating active and transmitting status of LAN port. Please refer to the following description for LED status:

TD+, TD-: Positive/negative send	RD+, RD-: Positive/negative
ACTLED: Network activity status	LILED: Network link status light.

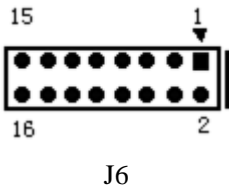


TX+ (Pin#1)
 TX- (Pin#2)
 RX+ (Pin#3)
 RX- (Pin#6)

LAN1, LAN2

ACTLED (Green light)	Network activity indicating status	LILED (Yellow light)	Network link indicating status
Flashing	Transmitting data	On	Linked
Off	No data transmitting	Off	Unlinked

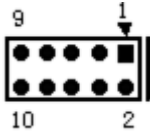
MPC8245 COP Debugging Port



Pin	Signal Name	Pin	Signal Name
1	COP_TDO	2	CLKFLIP_DIS
3	COP_TDI	4	COP_TRST#
5	+3.3V pull up	6	+3.3V pull up
7	COP_TCK	8	NC
9	COP_TMS	10	NC
11	COP_SRST#	12	GND

13	COP_HRESET#	14	NC
15	+3.3V pull up	16	GND

CPLD Debugging Port

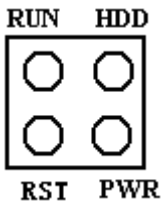


J7

Pin	Signal Name	Pin	Signal Name
1	MPLD_TCK	2	GNDS
3	MPLD_TDO	4	+3.3V pull up
5	MPLD_TMS	6	NC
7	NC	8	NC
9	MPLD_TDI	10	GND

Indicator LED

Indicator LEDs are brought out by D8 and D10.



LED	Status	Description
POWER	OFF	Power failure
	ON	Power normal
HDD	OFF	HDD free
	ON	HDD running
RST	OFF	Normal
	ON	System reset status
RUN	OFF	System idle
	ON	System operating

PMC Port

The PMC port on motherboard is used to connect PMC function card and expand function of system board, its signal is a set of PCI 32-bit bus with detailed signal definitions as below:

JN1:

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	TCK	2	-12V	3	GND	4	PIRQA#
5	PIRQB#	6	PIRQC#	7	BM1	8	+5V
9	PIRQD#	10	N/C	11	GND	12	+3.3V
13	CLK1	14	GND	15	GND	16	GNT0#
17	REQ0#	18	+5V	19	VIO	20	AD31
21	AD28	22	AD27	23	AD25	24	GND
25	GND	26	C/BE3#	27	AD22	28	AD21
29	AD19	30	+5V	31	VIO	32	AD17
33	FRAME#	34	GND	35	GND	36	IRDY#
37	DEVSEL#	38	+5V	39	GND	40	PLOCK#
41	NC	42	NC	43	PAR	44	GND
45	VIO	46	AD15	47	AD12	48	AD11
49	AD9	50	+5V	51	GND	52	C/BE0#
53	AD6	54	AD5	55	AD4	56	GND
57	VIO	58	AD3	59	AD2	60	AD1
61	AD0	62	+5V	63	GND	64	REQ64#

JN2:

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	+12V	2	TRST#(3)	3	TMS(2)	4	TDO(1)
5	TDI(2)	6	GND	7	GND	8	N/C
9	N/C	10	N/C	11	BM2(2)	12	+3.3V
13	RST#	14	BM3(3)	15	+3.3V	16	BM4(3)
17	PME#	18	GND	19	AD30	20	AD29
21	GND	22	AD26	23	AD24	24	+3.3V
25	IDSEL	26	AD23	27	+3.3V	28	AD20
29	AD18	30	GND	31	AD16	32	C/BE2#
33	GND	34	IDSL_B(1)	35	TRDY#	36	+3.3V
37	GND	38	STOP#	39	PERR#	40	GND
41	+3.3V	42	SERR#	43	C/BE1#	44	GND
45	AD14	46	AD13	47	M66EN	48	AD10
49	AD8	50	+3.3V	51	AD7	52	REQ_B(1)
53	+3.3V	54	GNT_B#(1)	55	N/C	56	GND
57	N/C	58	EREADEY(1)	59	GND	60	RSTOUT#(1)
61	ACK64#	62	+3.3V	63	GND	64	Monarch#(1)

JN4 is a customizable connector.

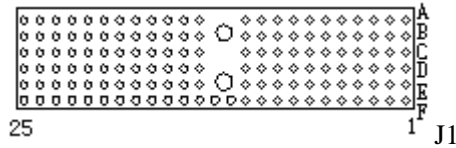
CompactFlash Port

Pin definitions of CompactFlash slot are as below (marked with CF1):

Pin	Signal Name	Pin	Signal Name
1	GND	26	CD1#
2	D3	27	D11
3	D4	28	D12
4	D5	29	D13
5	D6	30	D14
6	D7	31	D15
7	CS0#	32	CS1#
8	GND	33	VS1#
9	ATASEL#	34	IOR#
10	GND	35	IOW#
11	GND	36	WE#
12	GND	37	IRQ
13	VCC	38	VCC
14	GND	39	CSEL#
15	GND	40	VS2#
16	GND	41	RESET#
17	GND	42	IODY
18	A2	43	DREQ
19	A1	44	DACK#
20	A0	45	DASP#
21	D0	46	ATA66_DET
22	D1	47	D8
23	D2	48	D9
24	WP/IOCS16#	49	D10
25	CD2#	50	GND

CompactPCI Port

1. Pin definitions of J1



Pin	A	B	C	D	E	F
1	5V	-12V	TRST#	+12V	5V	GN
2	TCK	5V	TMS#	TDO	TDI	GN
3	INTA#	IRQB#	INTC#	5V	INTD#	GN
4	IPMB_P	HEALTHY#	V(I/O)	INTP	INTS	GN
5	NC	NC	PCI_RST#	GND	GNT0#	GN
6	REQ0#	PCI_PRESENT	3.3V	CLK0	AD31	GN
7	AD30	AD29	AD28	GND	AD27	GN
8	AD26	GND	V(I/O)	AD25	AD24	GN
9	C/BE3#	GND	AD23	GND	AD22	GN
10	AD21	GND	3.3V	AD20	AD19	GN
11	AD18	AD17	AD16	GND	C/BE2#	GN
12	KEY AREA					
13						
14						
15	3.3V	FRAME#	IRDY#	BD_SE	TRDY#	GND
16	DEVSEL	GND	V(I/O)	STOP#	LOCK#	GND
17	3.3V	IPMB_SCL	IPMB_SDA	GND	PERR#	GND
18	SERR#	GND	3.3V	PAR	C/BE1#	GND
19	3.3V	AD15	AD14	GND	AD13	GND
20	AD12	GND	V(I/O)	AD11	AD10	GND
21	3.3V	AD9	AD8	M66EN	C/BE0#	GND
22	AD7	GND	3.3V	AD6	AD5	GND
23	3.3V	AD4	AD3	5V	AD2	GND
24	AD1	5V	V(I/O)	AD0	ACK64	GND
25	5V	REQ64#	ENUM#	3.3V	5V	GND

2. Pin definitions of J2

Pin	A	B	C	D	E	F
22	GA4	GA3	GA2	GA1	GA0	GN
21	CLK	GND	NC	NC	NC	GN
20	CLK	GND	NC	GND	NC	GN
19	GN	GND	SMB_S	SMB_SC	SMB_ALE	GN
18	NC	NC	NC	GND	NC	GN
17	NC	GND	PRST#	REQ6#	GNT6#	GN
16	NC	NC	DEG#	GND	NC	GN
15	NC	GND	FAL#	REQ5#	GNT5#	GN
14	NC	NC	NC	GND	NC	GN
13	NC	GND	NC	NC	NC	GN
12	NC	NC	NC	GND	NC	GN
11	NC	GND	NC	NC	NC	GN
10	NC	NC	NC	GND	NC	GN
9	NC	GND	NC	NC	NC	GN
8	NC	NC	NC	GND	NC	GN
7	NC	GND	NC	NC	NC	GN
6	NC	NC	NC	GND	NC	GN
5	NC	GND	NC	NC	NC	GN
4	NC	NC	NC	GND	NC	GN
3	CLK	GND	GNT3#	REQ4#	GNT4#	GN
2	CLK	CLK	SYSEN	GNT2#	REQ3#	GN
1	CLK	GND	REQ1#	GNT1#	REQ2#	GN



J2

3. Pin definitions of J3

Pin arrangement of J3 is the same with that of J2

Pin	A	B	C	D	E	F
19	NC	NC	NC	NC	NC	GND
18	NC	NC	NC	NC	NC	GND
17	NC	NC	NC	NC	NC	GND
16	NC	NC	NC	NC	NC	GND
15	NC	NC	NC	NC	NC	GND
14	VCC3_3	VCC3_3	VCC3_3	VCC5	VCC5	GND
13	PMC_IO5	VCC3_3	PMC_IO3	PMC_IO2	PMC_IO1	GND
12	PMC_IO10	PMC_IO14	PMC_IO8	PMC_IO7	PMC_IO6	GND
11	PMC_IO15	PMC_IO19	PMC_IO13	PMC_IO12	PMC_IO11	GND
10	PMC_IO20	PMC_IO24	PMC_IO18	PMC_IO17	PMC_IO16	GND
9	PMC_IO25	PMC_IO29	PMC_IO23	PMC_IO22	PMC_IO21	GND
8	PMC_IO30	PMC_IO34	PMC_IO28	PMC_IO27	PMC_IO26	GND
7	PMC_IO35	PMC_IO39	PMC_IO33	PMC_IO32	PMC_IO31	GND
6	PMC_IO40	PMC_IO44	PMC_IO38	PMC_IO37	PMC_IO36	GND
5	PMC_IO45	PMC_IO49	PMC_IO43	PMC_IO42	PMC_IO41	GND
4	PMC_IO50	PMC_IO54	PMC_IO48	PMC_IO47	PMC_IO46	GND
3	PMC_IO55	PMC_IO59	PMC_IO53	PMC_IO52	PMC_IO51	GND
2	PMC_IO60	PMC_IO64	PMC_IO58	PMC_IO57	PMC_IO56	GND
1	VIO_EARLY	PMC_IO9	PMC_IO63	PMC_IO62	PMC_IO61	GND

4. Pin definitions of J5

CPC-1631CVD2NA offers one J5 socket, whose pin arrangement is the same with that of J2:

Pin	Signal Name	Signal Name	Signal Name	Signal Name	Signal Name	Signal Name
	A	B	C	D	E	F
22	GND	GND	GND	GND	GND	GND
21	LPT_SLIN-	LPT_INIT-	LPT_STB-	LPT_AFD-	GND	LPT_SLIN-
20	LPT_SLCT	LPT_PE	LPT_ACK-	LPT_D4	GND	LPT_SLCT
19	LPT_D0	LPT_D1	LPT_D3	LPT_D7	GND	LPT_D0
18	LPT_D5	LPT_D6	NC	NC	GND	LPT_D5
17	GND	GND	GND	GND	GND	GND
16	ACZ_RST	ACZ_BITCLK	ACZ_SDOOUT	ACZ_SDIN	GND	ACZ_RST
15	VCC-12	VCC12	VCC5	NC	GND	VCC-12
14	GND	GND	GND	GND	GND	GND
13	COM2_DCD	COM2_RXD	COM2_DTR	NC	GND	COM2_DCD
12	COM2_DSR	COM2_CTS	COM2_RTS	NC	GND	COM2_DSR
11	GND	GND	GND	GND	GND	GND
10	VCC3_3	NC	NC	NC	GND	VCC3_3
9	IO_LAN0_A CT	IO_LAN0_LI NK	IO_LAN1_LI NK	NC	GND	IO_LAN0_AC T
8	IO_TXD+_0	IO_TXD-_0	IO_RXIN-_0	NC	GND	IO_TXD+_0
7	IO_TXD+_1	IO_TXD-_1	IO_RXIN-_1	NC	GND	IO_TXD+_1
6	NC	NC	NC	NC	GND	NC
5	NC	NC	NC	NC	GND	NC
4	NC	NC	NC	NC	GND	NC
3	NC	NC	NC	NC	GND	NC
2	NC	NC	NC	NC	GND	NC
1	USB3_D+	USB3_D-	USB4_D+	USB4_D-	GND	USB3_D+

Appendix 1 Linux Operating System

Watchdog Programming Guide

CPC-1631CVD2NA offers one hardware watchdog. Through programming, the watchdog timeout event can reset system or generate a maskable interrupt.

Configuration address of watchdog is decided by logic. bit7 of 0xffc00002 is the control bit of watchdog, if bit 7 is set to 0, watchdog will be enabled. 0xffc00001 is the watchdog register, it needs cyclical input of 0x55 and 0xaa to produce clear-watchdog pulse.

Programming source codes for watchdog under Linux operating system are as below.

```
#include <linux/types.h>
#include <linux/version.h>
#include <linux/module.h>
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/compiler.h>
#include <linux/string.h>
#include <linux/ctype.h>
#include <linux/ioport.h>
#include <linux/platform_device.h>
#include <linux/delay.h>
#include <linux/err.h>
#include <linux/fs.h>
#include <linux/slab.h>
```

```
#include <linux/completion.h>
#include <linux/crc32.h>

//#include <linux/clk.h>
#include <asm/uaccess.h>
#include <asm-generic/errno-base.h>
#include <asm/io.h>
#include <asm/irq.h>
#include <asm/types.h>
#include <linux/bitops.h>
#include <linux/interrupt.h>
//#include <linux/pm.h>
MODULE_AUTHOR ("kingzc <sz.zcliu@evoc.cn>");
MODULE_DESCRIPTION (" driver for HW-WatchDog ");
MODULE_LICENSE("GPL");
#define DRIVER_VERSION "0.1"
#define DRIVER_AUTHOR "kingzc <sz.zcliu@evoc.cn>"
#define DRIVER_DESC "HW-WatchDog driver"
#define MY_NAME "HW-WatchDog"
int debug=1;
#define dbg(format, arg...) \
    do \
    { \
        if(debug) \
            printk (KERN_DEBUG "%s: " format "\n", \
                MY_NAME , ## arg); \
    }
```

```
    } while(0)
#define err(format, arg...) printk(KERN_ERR "%s: " format "\n",
MY_NAME , ## arg)
#define info(format, arg...) printk(KERN_INFO "%s: " format "\n",
MY_NAME , ## arg)
#define warn(format, arg...) printk(KERN_WARNING "%s: " format
"\n", MY_NAME , ## arg)

#define DOG_CTRL (void __iomem *)0xffc00002 //bit 7 ,=1:close,
=0:open
#define DOG_REG (void __iomem *)0xffc00001 //need feed DOG
0x55 and 0xaa

static struct semaphore thread_exit;           /* guard ensure thread
has existed before calling it quits */
static int thread_finished = 1;

static WDT_Open(void)
{
    unsigned char data;
    data=readb(DOG_CTRL);
    data=data&0x7f;
    writeb(data,DOG_CTRL);
}

static WDT_Close(void)
{
    unsigned char data;
    data=readb(DOG_CTRL);
```

```
    data=data|0x80;
    writeb(data,DOG_CTRL);
}

static int feed_Dog_thread()
{
    lock_kernel();
    daemonize("Feed HW_WatchDog");
    unlock_kernel();
    while(1)
    {
        if (thread_finished || signal_pending(current))
            break;
        writeb(0x55,DOG_REG);
        msleep(10);
        writeb(0xaa,DOG_REG);
        msleep(10);
        writeb(0x55,DOG_REG);
        msleep(10);
        writeb(0xaa,DOG_REG);
        msleep(10);
        writeb(0x55,DOG_REG);
        msleep(10);
        writeb(0xaa,DOG_REG);
        msleep(10);
        writeb(0x55,DOG_REG);
        msleep(1000);
        // info("I'm now feeding dog :)\n");
    }
}
```

```
        dbg("feed_Dog_thread signals exit");
        up(&thread_exit);
        return 0;
    }
static int WDT_start_thread(void)
{
    int pid;
    /* initialize our semaphores */
    init_MUTEX_LOCKED(&thread_exit);
    thread_finished = 0;
        pid = kernel_thread(feed_Dog_thread, NULL, 0);
    if (pid < 0) {
        err("Can't start up our thread");
        return -1;
    }
    dbg("Our thread pid = %d", pid);
    return 0;
}

static void WDT_stop_thread(void)
{
    thread_finished = 1;
    dbg("thread finish command given");
    down(&thread_exit);
}
static int __init HW_WatchDog_init(void)
{
    int ret;
    info(DRIVER_DESC " version: " DRIVER_VERSION);
```

```
    WDT_Open();
    ret=WDT_start_thread();
    if(ret)
    {
        err("can't start feed_Dog_thread  :");
        WDT_Close();
        return ret;
    }
    return 0;
}
static void __exit HW_WatchDog_exit(void)
{
    info("exit frome HW-WatchDog \n");
    WDT_Close();
    WDT_stop_thread();
}
module_init(HW_WatchDog_init);
module_exit(HW_WatchDog_exit);rt();
```

I/O Mapping

The space of system I/O address is 64K, each peripheral occupies a section of I/O space. I/O ports of all peripherals are expanded from south bridge (VT82C686B), base address of MPC8245 I/O space is 0xfe000000. The I/O addresses of PCI devices (such as PCI network card) are not listed, for their addresses are configured by software.

Address	Function
000h – 01Fh	DMA controller
020h – 03Fh	Interrupt controller
040h – 05Fh	Timer/counter
060h – 06Fh	Keyboard controller
070h – 077h	RTC/CMOS/NMI-Disable
081h – 08Fh	DMA page register
0A0h – 0BFh	Slave interrupt controller
0C0h – 0DFh	Slave DMA controller
170h – 177h	Secondary IDE channel data
1F0h – 1F7h	Primary IDE channel data
2F8h – 2FFh	COM2
376h – 377h	Secondary IDE channel ctl
3F6h – 3F7h	Primary IDE channel ctl
3F8h – 3FFh	COM1
378-37F or 278-27F	Parallel Port

IRQ Assignment Table

Interrupt of the motherboard contains two parts, one is the external interrupt (0~4) of MPC8245 and another is interrupt internally provided by I8259 of south bridge (VT82C686B), interrupts of south bridge are assigned at 0~15 and CPU internal interrupts are assigned at 16~20.

CLASS	Function
IRQ 3	COM2
IRQ 4	COM1
IRQ 7	Parallel Port
IRQ 10	USB
IRQ 14	Master IDE channel
IRQ 15	Slave IDE channel
IRQ 16	Network port 2
IRQ 17	Network port 1
IRQ 18	South bridge (VT82C686B)
IRQ 20	CPCI

Appendix 2 VxWorks Operating System

Watchdog Programming Guide

CPC-1631CVD2NA offers one hardware watchdog. Through programming, the watchdog timeout event can reset system or generate a maskable interrupt.

Configuration address of watchdog is decided by logic. bit7 of 0xffc00002 is the control bit of watchdog, if bit 7 is set to 0, watchdog will be enabled. 0xffc00001 is the watchdog register, it needs cyclical input of 0x55 and 0xaa to produce clear-watchdog pulse.

Programming sample of watchdog under VxWorks operating system is described as below.

1. Start up hardware watchdog function

```
/*Start up hardware watchdog*/  
void watchDogStart()  
{  
    *(volatile unsigned char *)0xffc00002 &= ~0x80 ;  
}
```

2. Clear-watchdog function

```
/*Clear-watchdog function*/  
void watchDogClear()  
{  
    while(1)  
    {
```

```
*(volatile unsigned char *)0xffc00001 = 0x55;
taskDelay(1);/*10ms delay*/
*(volatile unsigned char *)0xffc00001 = 0xaa;
taskDelay(1);
*(volatile unsigned char *)0xffc00001 = 0x55;
taskDelay(1);
*(volatile unsigned char *)0xffc00001 = 0xaa;
taskDelay(1);
*(volatile unsigned char *)0xffc00001 = 0x55;
taskDelay(1);
*(volatile unsigned char *)0xffc00001 = 0xaa;
taskDelay(1);
*(volatile unsigned char *)0xffc00001 = 0x55;

taskDelay(100);/*Clear watchdog per second*/

}
}
```

3. Start up watchdog and a clear-watchdog task

```
/*Start up clear-watchdog task*/
(void)taskSpawn("watchDogTask",5,0,4000,(FUNCPTR)watchDogClear
,0,0,0,0,0,0,0,0,0,0);

/*Startup hardware watchdog*/
watchDogStart();
```

I/O Mapping

The space of system I/O address is 64K, each peripheral occupies a section of I/O space. I/O ports of all peripherals are expanded from south bridge (VT82C686B), base address of MPC8245 I/O space is 0xfe000000. The I/O addresses of PCI devices (such as PCI network card) are not listed, for their addresses are configured by software.

Address	Function
000h – 01Fh	DMA controller
020h – 03Fh	Interrupt controller
040h – 05Fh	Timer/counter
060h – 06Fh	Keyboard controller
070h – 077h	RTC/CMOS/NMI-Disable
081h – 08Fh	DMA page register
0A0h – 0BFh	Slave interrupt controller
0C0h – 0DFh	Slave DMA controller
170h – 177h	Slave IDE channel
1F0h – 1F7h	Master IDE channel
2F8h – 2FFh	COM2
376h – 377h	Slave IDE channel
3F6h – 3F7h	Master IDE channel
3F8h – 3FFh	COM1

IRQ Assignment Table

Interrupt of the motherboard contains two parts, one is the external interrupt (IRQ0-IRQ4) of MPC8245 and another is interrupt internally provided by I8259 of south bridge (VT82C686B), interrupts of south bridge will be added with a offset of 16. The I/O addresses of PCI devices (such as PCI network card) are not listed, for their addresses are configured by software.

CLASS	Function
IRQ 0	Network port 2
IRQ 1	Network port 1
IRQ 2	South bridge (VT82C686B)
IRQ 3	Reservation
IRQ 4	CPCI
IRQ 16	System timer
IRQ 17	Reservation
IRQ 18	COM2
IRQ 19	COM1
IRQ 30	Master IDE channel
IRQ 31	Slave IDE channel