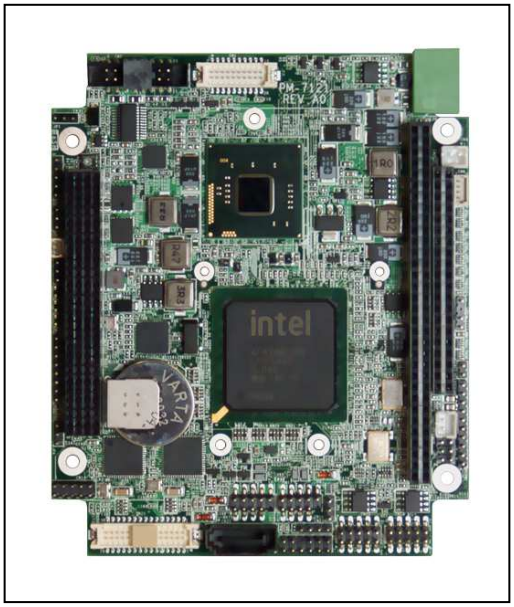
User Manual

PM-7121

PC/104-Plus module with onboard Atom processor D2550 or N2600, Intel ICH10R chipset, DDR3 up to 2GB, 2 x Giga LAN, Onboard VGA, LVDS, SATA, CF socket, 4 x USB, 2 x COM, GPIO, DC 5V input



Ver.	Release Date	Update
1.0	2012.06.28	Release
1.1	2012.08.16	
1.2	2014.04.16	Correct CN4 Pin-1 location

Copyright

Trademark

Limitation of liability

For more information or other AEWIN products, please visit our website http://www.aewin.com.tw.

For technical supports, please send your inquiry to sales@aewin.com.tw

Packing list

Before use this product, please make sure that the following materials have been shipped.

- 1 x PM-7121 PC/104-Plus board
- 1 x CPU cooling Fan (p/n: 49L-F000047-00)
- 1 x VGA cable (p/n: 46L-IVGA01-00)
- 1 x COM port cable (p/n: 46L-ICOM25-00)
- 1 x LAN cable (p/n: 46L-ILAN10-00)
- 1 x LAN module (p/n: R214A)
- 1 x USB cable (p/n: 46L-IUSB03-00)
- 1 x Power cable (p/n: 46L-IPOW82-01)
- 1 x CD Utility (p/n: TBD)

* If any of those items are missing or damaged, please contact with sales representative or distributor

Model Name	Description
PM-7121A-D21	PC/104-Plus with Atom D2550, onboard VGA, LVDS, 2 x Giga LAN, COM, USB, SATA, CF socket, DC 5V input.
PM-7121A-N16	PC/104-Plus with Atom N2600, onboard VGA, LVDS, 2 x Giga LAN, COM, USB, SATA, CF socket, DC 5V input.
PM-7121B-D21	PC/104-Plus with Atom D2550, onboard VGA, LVDS, 2 x Giga LAN, COM, USB, SATA, CF socket, DC 5V input. (Pin Down Type)
PM-7121B-N16	PC/104-Plus with Atom N2600, onboard VGA, LVDS, 2 x Giga LAN, COM, USB, SATA, CF socket, DC 5V input. (Pin Down Type)

Optional Accessory

Audio daughter board (p/n: IP-90340)

Audio cable connect daughter board & PM-7121 (p/n: 46L-IDE18-00)

Safety Information

- To prevent electrical shock hazard, disconnect the power cable from the electrical outlet before relocating the system.
- When adding or removing devices to or from the system, ensure that the power cables for the devices are unplugged before the signal cables are connected. If possible, disconnect all power cables from the existing system before you add a device.
- Before connecting or removing signal cables from the motherboard, ensure that all power cables are unplugged.
- Seek professional assistance before using an adapter or extension cord. These devices could interrupt the grounding circuit.
- Make sure that your power supply is set to the correct voltage in your area.
- If you are not sure about the voltage of the electrical outlet you are using, contact your local power company.
- If the power supply is broken, do not try to fix it by yourself. Contact a qualified service technician or your retailer.

Operation Safety

- Before installing the motherboard and adding devices on it, carefully read all the manuals that came with the package.
- Before using the product, make sure all cables are correctly connected and the power cables are not damaged. If you detect any damage, contact your dealer immediately.
- To avoid short circuits, keep paper clips, screws, and staples away from connectors, slots, sockets and circuitry.
- Avoid dust, humidity, and temperature extremes. Do not place the product in any area where it may become wet.
- Place the product on a stable surface.
- If you encounter technical problems with the product, contact a qualified service technician or your retailer.

Contents

1.1 Introduction

PM-7121 is a PC/104-Plus with onboard Intel® Atom™ processor and Intel® ICH10R chipset, Intel® Atom™ N2600 is a dual core processor which Clocked at 1.60 GHz, the power consumption of N2600 processor only 3.5 watt (Atom™ D2550 for 10W TDP) and it is good for application which need high-performance at low energy-consumption levels, such as medical devices, Digital Signage, Information Kiosk, Point-of-Sale, and Gaming machines and Industrial control systems.

PM-7121 support dual display for VGA + LVDS, The Atom processor also build-in graphic (GMA 3650) and a video decoding engine for MPEG4 Part2, VC-1, WMV9, and H.264. The integrated 64Bit Single-Channel DDR3-800/1067 memory controller supports one SODIMM and a maximum of 2 GB RAM.

Onboard Intel® ICH 10R chipset provide extensive IO ports support, including 1 x SATAII interface with 300MB/s transfer rate, 4 x USB 2.0, HD Audio, onboard 2 x Gigabit Ethernet provided by Intel® 82583V Gigabit Ethernet. 1 x RS232/422/485 & 1 x RS232, 8-bit GPIO and 1 x CompactFlash socket support Type-2 CompactFlash card.

Aewin's PM-7121 support PC/104-Plus module and accept single DC 5V input for operating.

AEWIN offers reliable and solid products which are produced under Management System Standards: ISO9001-2000 Certificate. The certificate keeps us focused on our quality objectives of management and environmental production. Its willingness to customize standard products for meet unique customer needs makes AEWIN different. All ODM projects are welcome. Years of experiences enables AEWIN to fulfill the customer's vision, by delivering products to exact specifications. AEWIN R&D team is proud of its strong engineering background. R&D professionals account for 25% of the AEWIN workforce. We focus on developing new products for both emerging and established markets

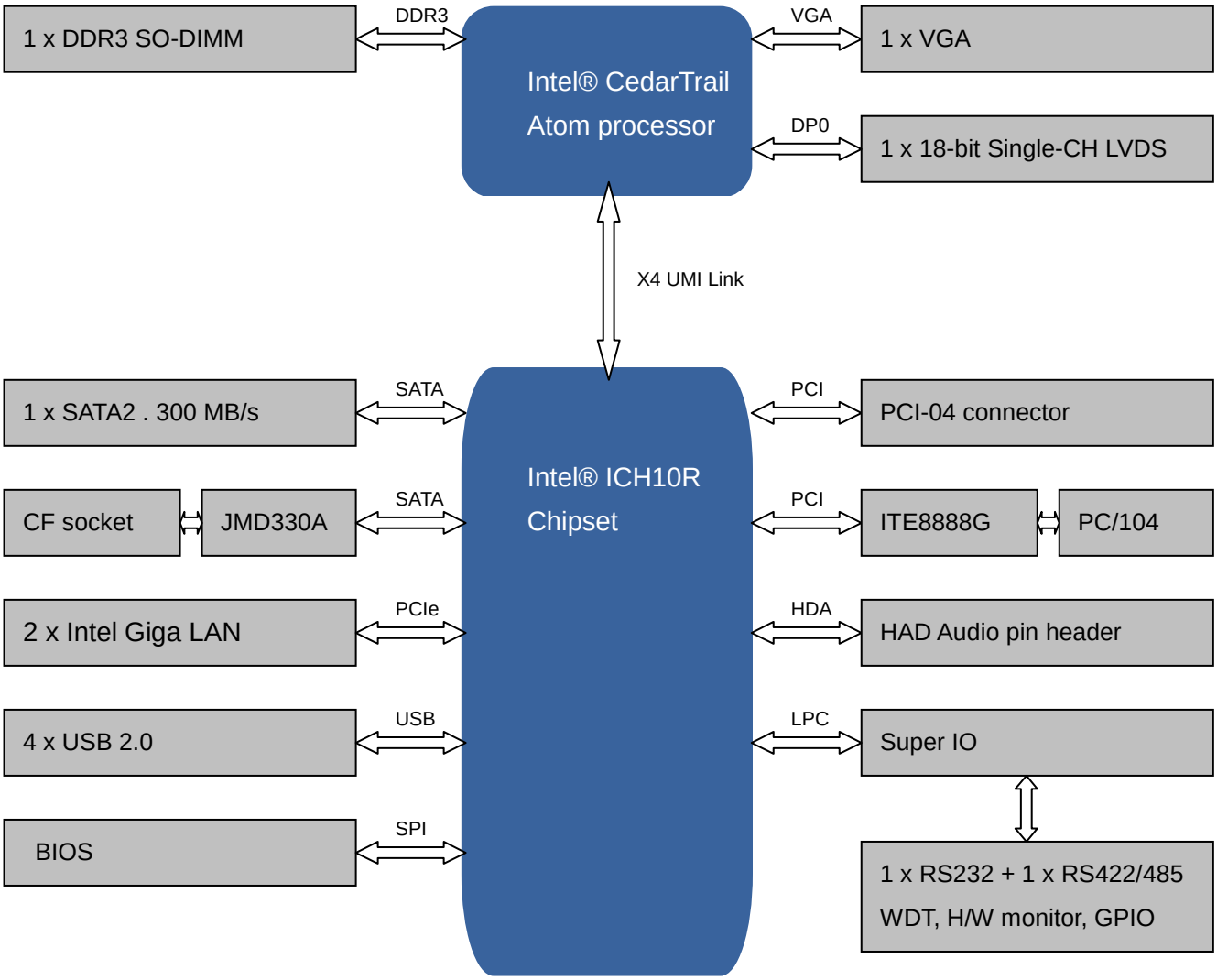
For more information about OEM/ODM, please contact us
Email: sales@aewin.com.tw TEL: +886-2-8692-6677

1.2 Specification of board

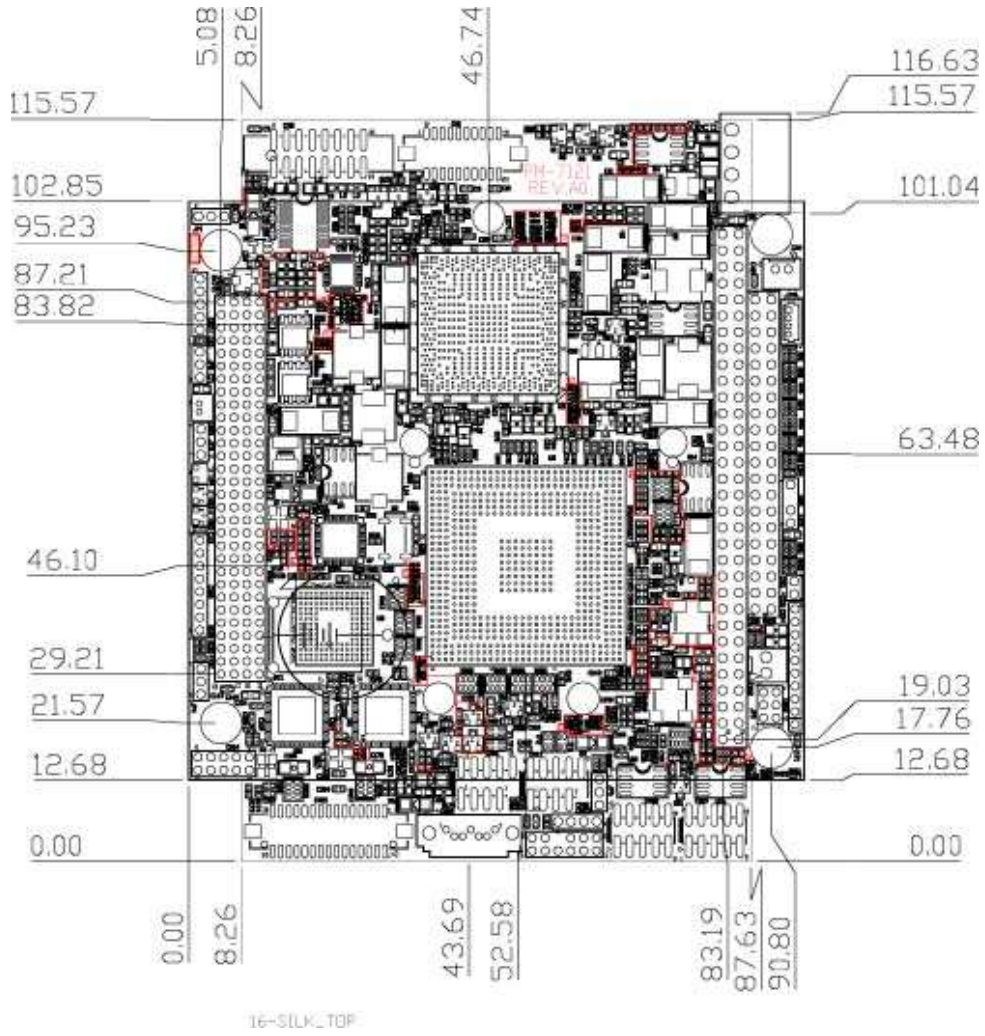
Form Factor	PC/104-Plus
CPU	Onboard Intel® Atom processor D2550 1.86 GHz Onboard Intel® Atom processor N2600 1.60 GHz
Chipset	Intel® ICH10R chipset
Memory	1 x DDR3 800/1066 MHz SO-DIMM up to 2GB
BIOS	AMI® SPI BIOS
Watchdog Timer	255 levels timer interval, (1 ~ 255 seconds), setup by software
Integrated graphic	Intel® Cedarview integrated graphic
VGA interface	1 x internal VGA connector
LVDS interface	1 x 18/24-bit single channel LVDS interface (D2550) 1 x 18-bit single channel LVDS interface (N2600)
SATA interface	1 x SATAII up to 300 MB/s
SSD interface	1 x CompactFlash socket
Ethernet	2 x Intel® i211AT or i210AT Gigabit ethernet
Expansion interface	1 x PC/104-Plus
COM	1 x RS422/485 , 1 x RS232
USB	4 x USB 2.0
Audio	1 x HDA audio pin-header (need to purchase Audio module to work with)
GPIO	8-bit programmable GPIO interface
Power Input	Single DC 5V input
Board Size	96 mm x 116 mm
Operating temperature	0°C ~ 60°C

Note：All specifications and photos are subject to change without notice

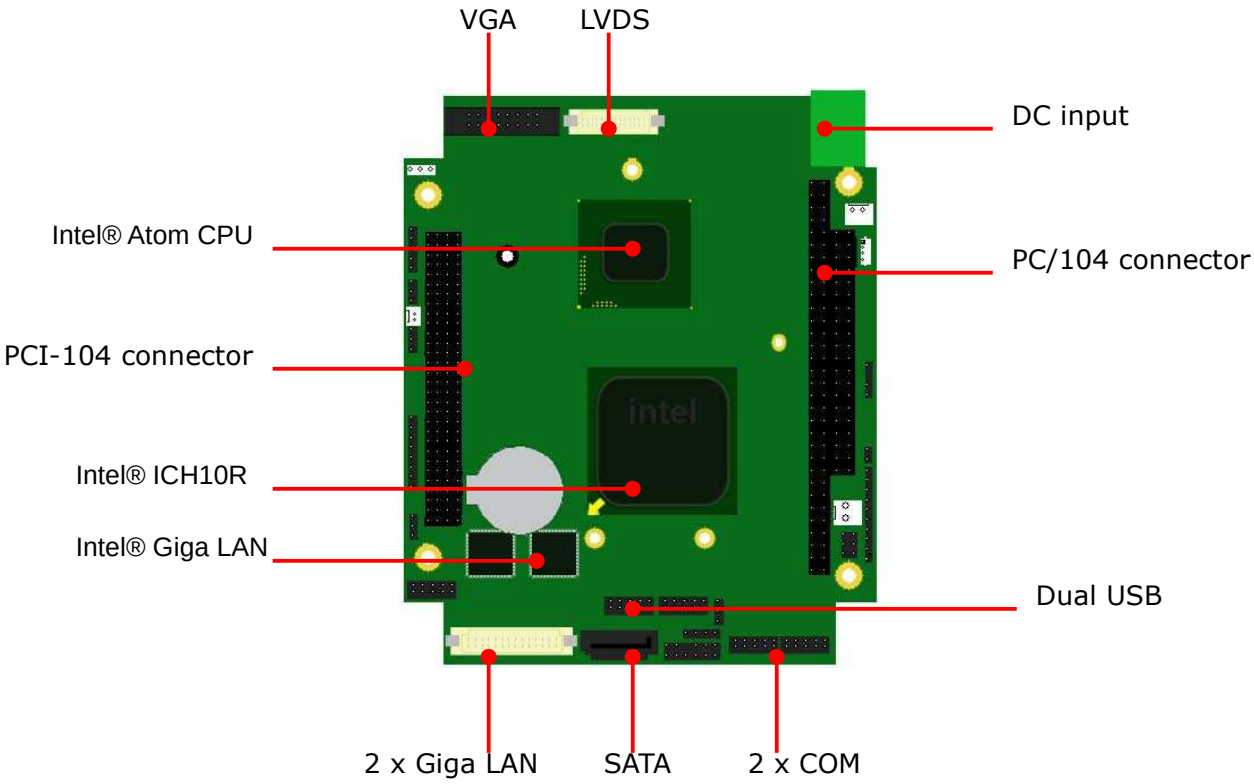
1.3 Block Diagram

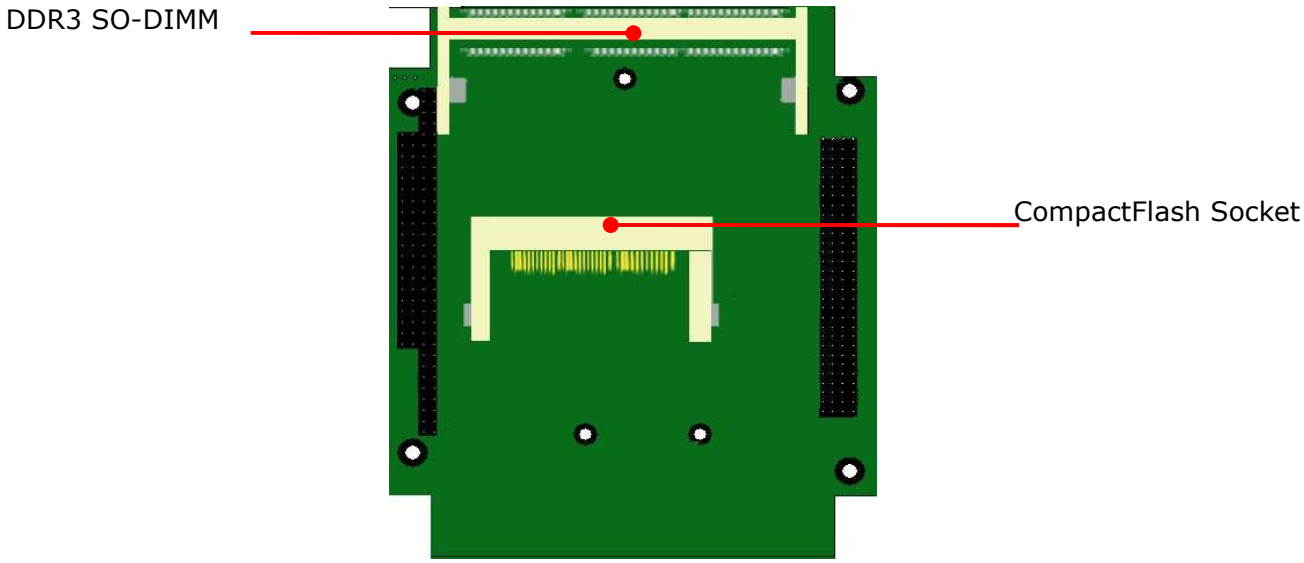


1.4 Board Layout Dimension

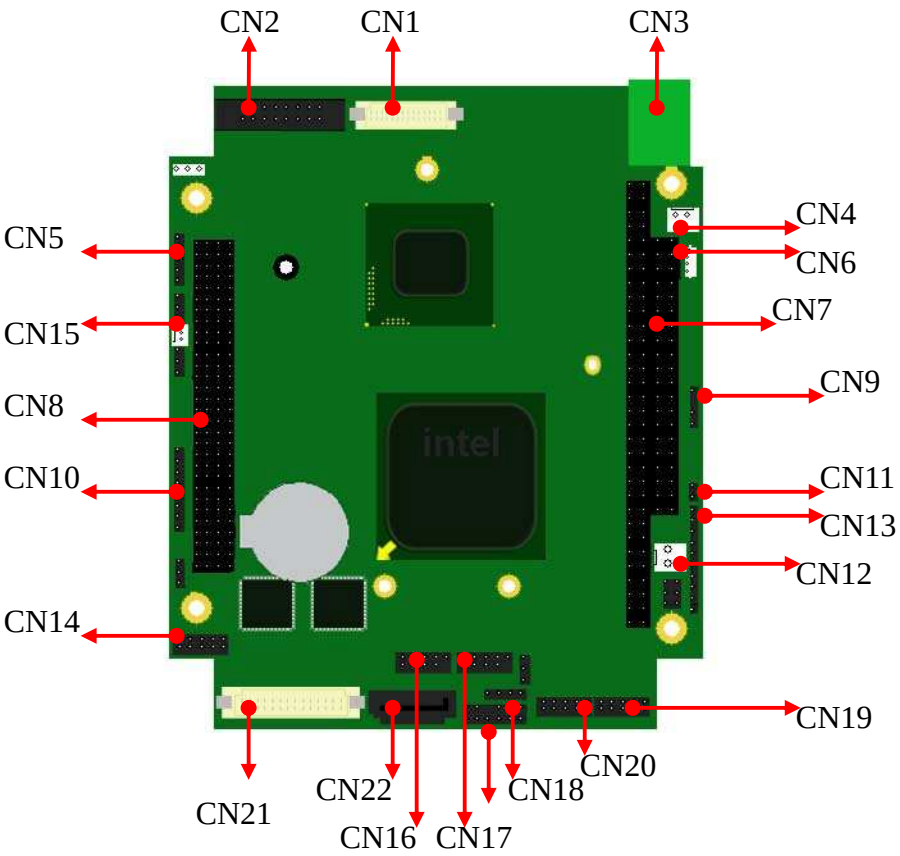


1.5 IO ports


VGA
LVDS
DC input
Intel® Atom CPU
PC/104 connector
PCI-104 connector
Intel® ICH10R
Intel® Giga LAN
Dual USB
2 x Giga LAN
SATA
2 x COM


DDR3 SO-DIMM
CompactFlash Socket

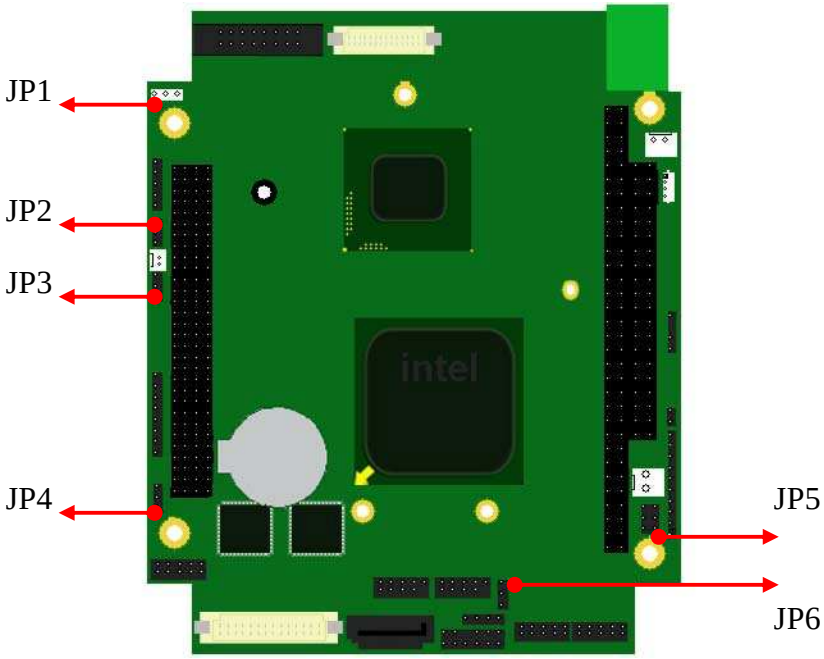
2.1 The location of onboard connectors



Label	Function
CN1	LVDS connector
CN2	VGA connector
CN3	+5V DC input
CN4	+12V DC input
CN5	LVDS backlight control
CN6	+5V_SB DC input and PS_ON
CN7	PC/104 connector
CN8	PCI-104 connector
CN9	-12V / -5V DC input
CN10	Front Panel
CN11	Power Button
CN12	FAN connector
CN13	GPIO pin header
CN14	LPC connector

Label	Function
CN15	Battery connector
CN16	USB ports 2/3
CN17	USB ports 0/1
CN18	COM2 (RS422 / 485) connector
CN19	COM1 (RS232) connector
CN20	COM2 (RS232) connector
CN21	LAN1 , LAN2 connector
CN22	SATA connector
CN23	HAD connector
DIMM	1 x SO-DIMM on solder-ide
CF1	1 x CompactFlash socket on solder-side

2.2 The location of onboard jumpers



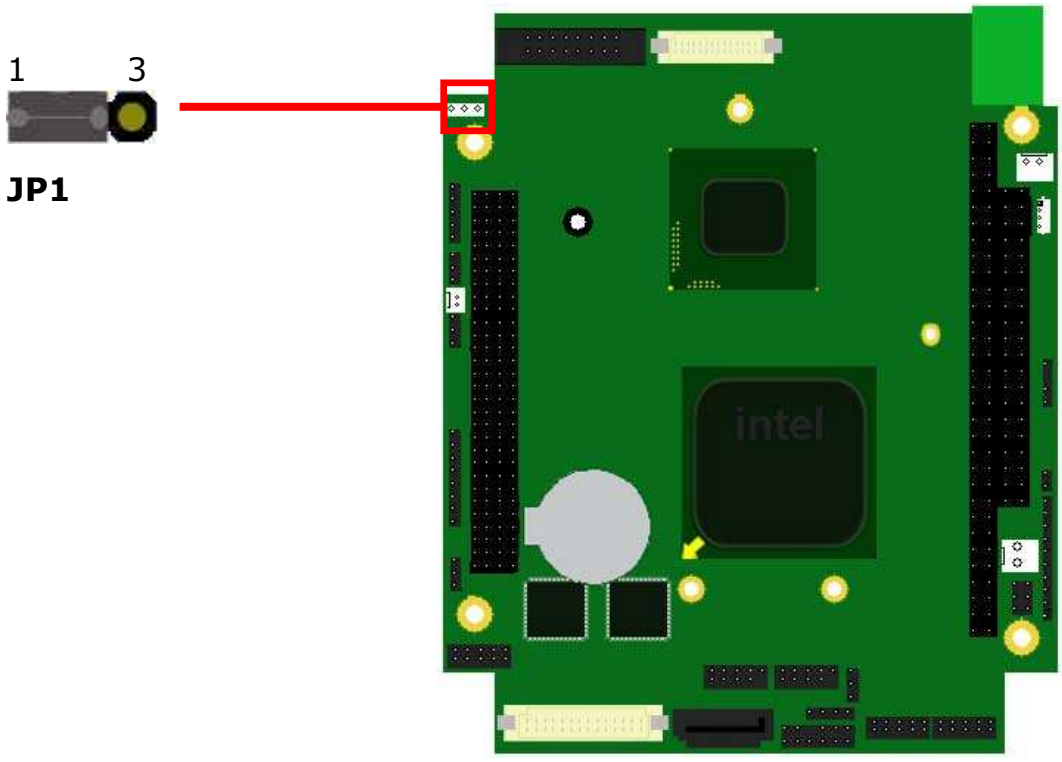
Label	Function
JP1	Panel Voltage (VCC) select
JP2	PCI-104 Vio select
JP3	WDT select
JP4	Clear CMOS
JP5	COM2 select
JP6	AT & ATX mode select

2.3 The function list of onboard jumpers setting

- 2.3.1 : JP1 for LVDS Panel Vcc select

JP1	
Closed Pin	Result
1-2 *	+3.3V
2-3	+5V

* Default setting



- 2.3.2 : JP2 for PCI-104 Vio select

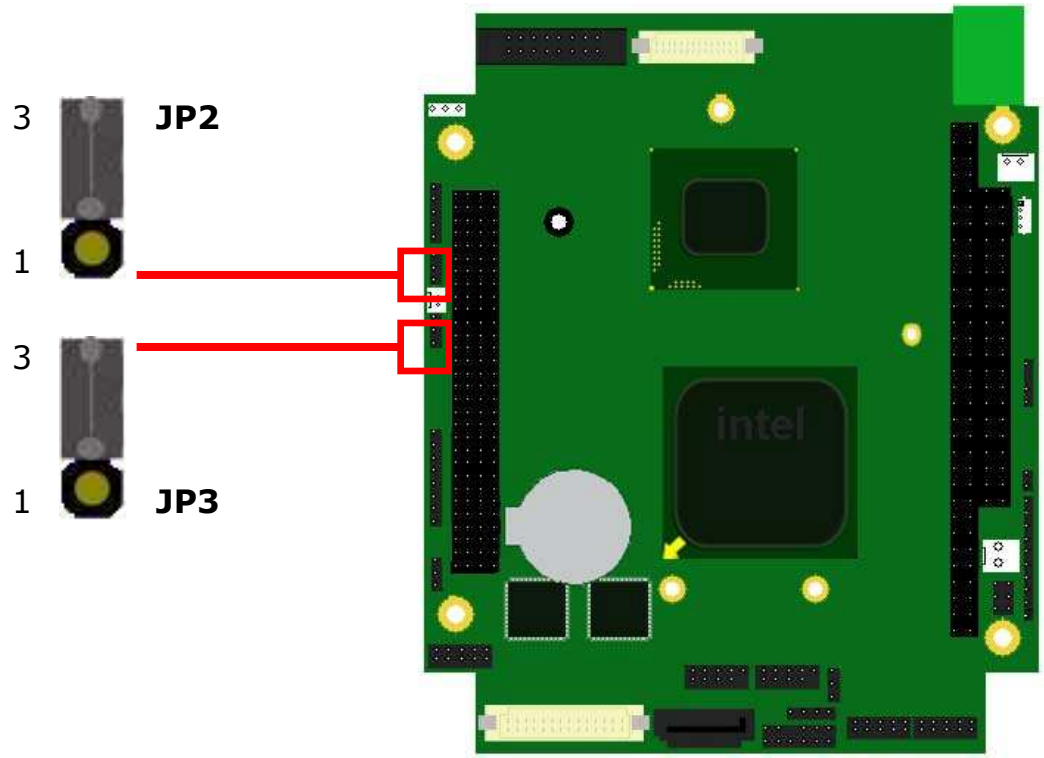
JP2	
Closed Pin	Result
1-2	+5V
2-3 *	+3.3V

* Default setting

- 2.3.3 : JP3 for WDT select

JP3	
Closed Pin	Result
1-2	+5V
2-3 *	+3.3V

* Default setting

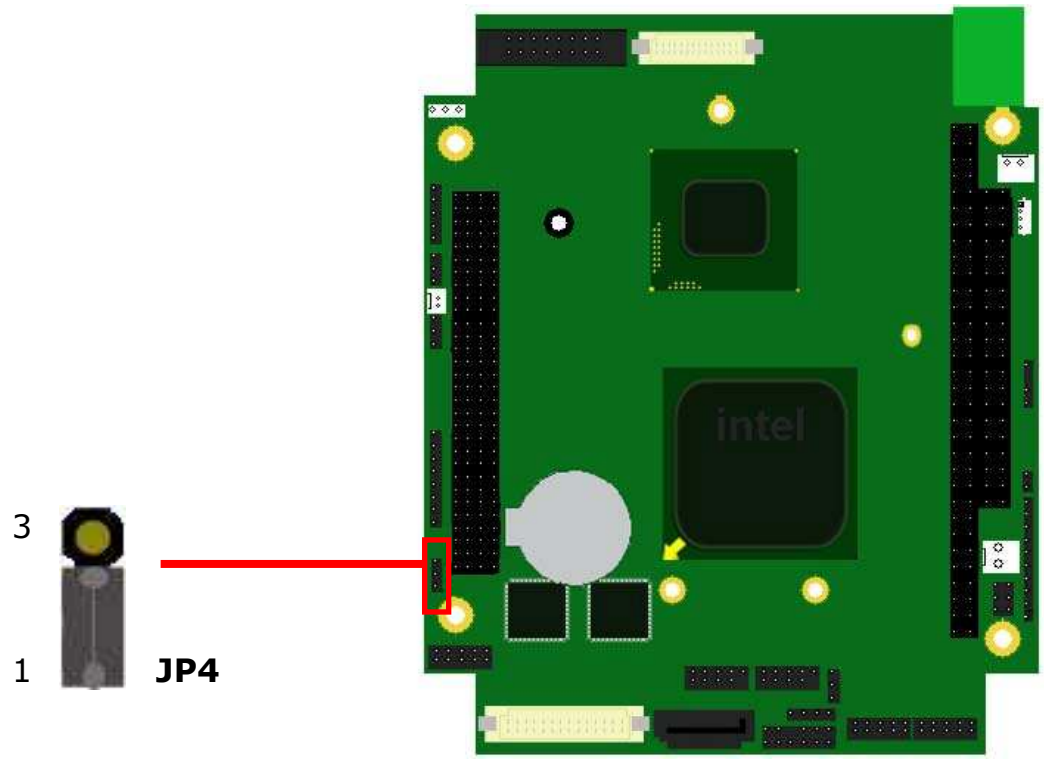


- 2.3.4 : JP4 for Clear CMOS

If you want to clean the CMOS data, set jumper to 2-3 just for few seconds, Then, Move the jumper back to 1-2 pin

JP4	
Closed Pin	Result
1-2 *	Normal
2-3	Clear CMOS

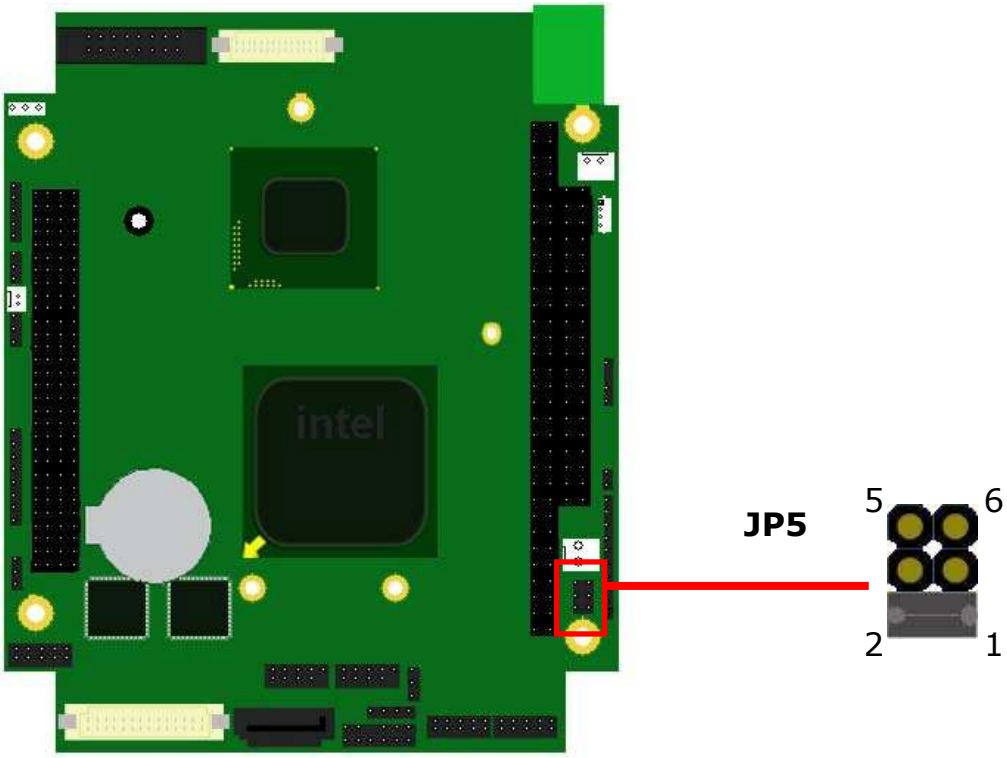
* Default setting



- 2.3.5 : JP5 for COM2 select

JP5	
Closed Pin	Result
1-2 *	RS232
2-3	RS422
5-6	RS485

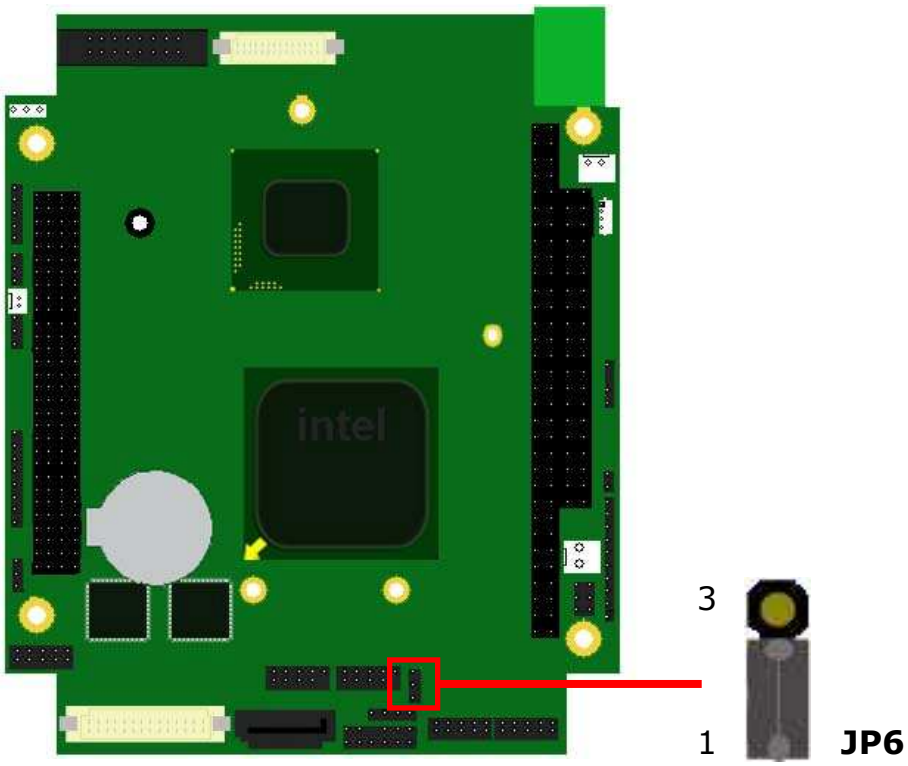
* Default setting



- 2.3.6 : JP6 for AT/ATX mode select

JP6	
Closed Pin	Result
1-2 *	ATX mode
2-3	AT mode

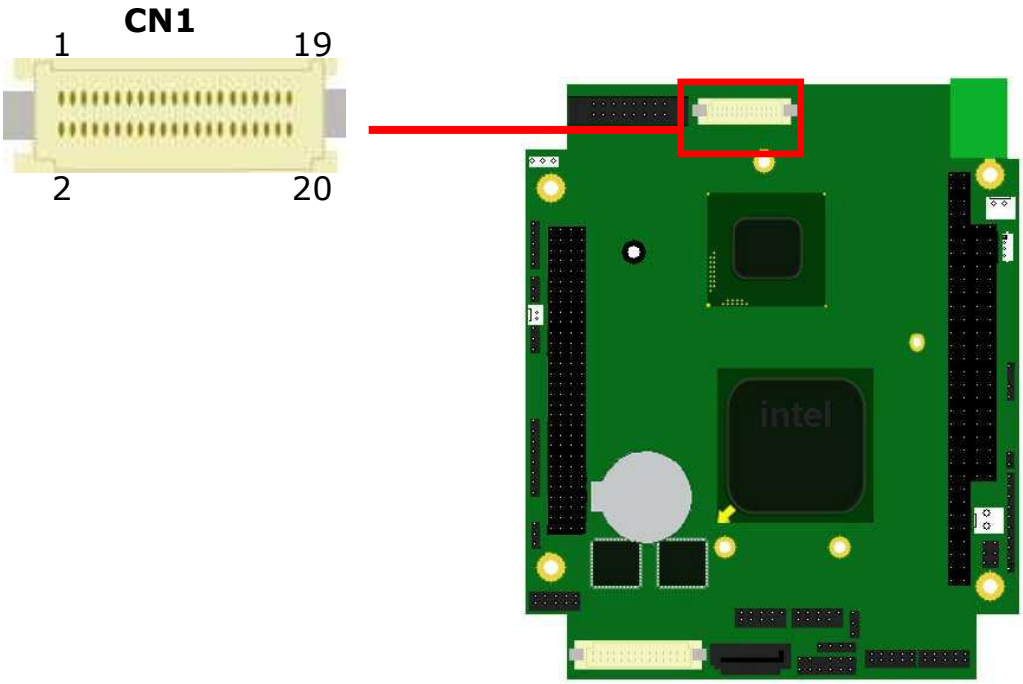
* Default setting



2.4 The pin define of onboard pin header

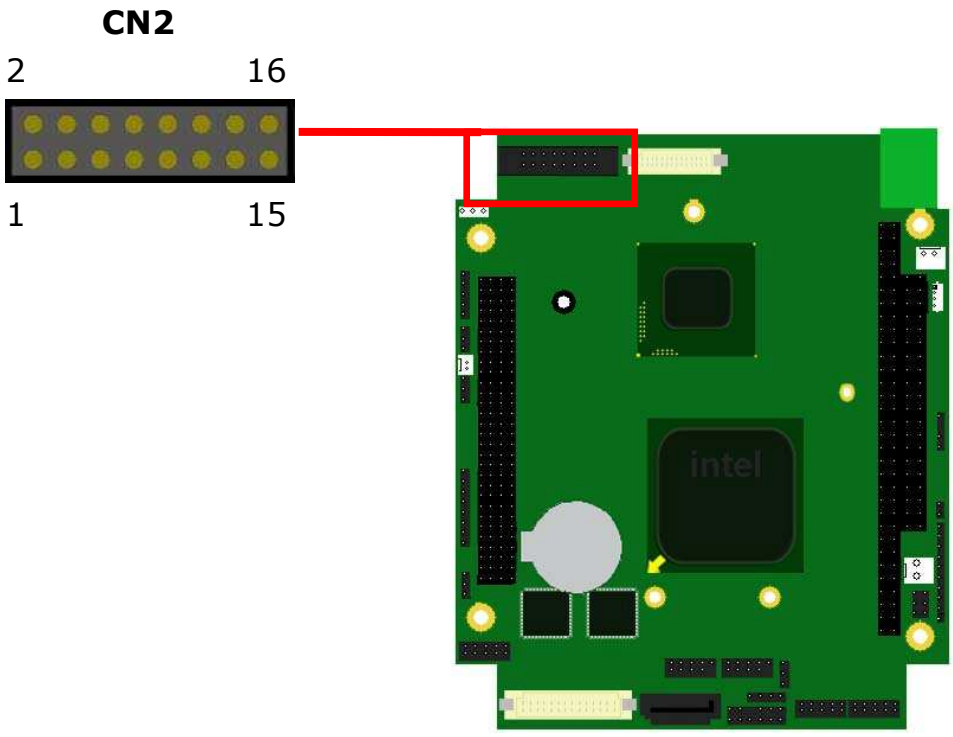
- 2.4.1 : CN1 for LVDS connector

CN6 : 2 x 10 header , connector type : DF13A-20DP-1.25V			
Pin	**Signal**	**Pin**	**Signal**
1	TXP0	2	TXN0
3	GND	4	GND
5	TXP1	6	TXN1
7	GND	8	VCC_LCD
9	TXP2	10	TXN2
11	CLKP	12	CLKN
13	GND	14	GND
15	TXP3	16	TXN3
17	LBKLT_EN_DELAY	18	VCC_LCD
19	DDC_DATA	20	DDC_CLK



- 2.4.2 : CN2 for VGA connector

CN7 : 2 x 8 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	CRT_RED	2	CRT_GREEN
3	CRT_BLUE	4	+5VP0
5	GND	6	GND
7	GND	8	GND
9	+V_CRTCON	10	GND
11	V5P0	12	CRT_DDC_SDA
13	CRT_HSYNC	14	CRT_VSYNC
15	CRT_DDC_SLC	16	N/C

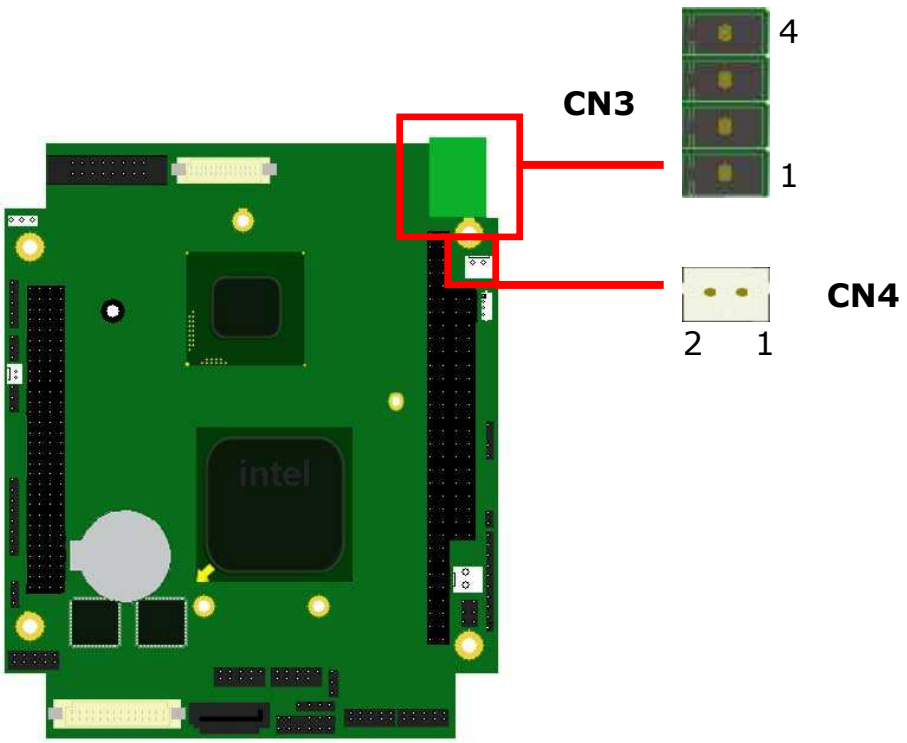


- 2.4.3 : CN3 for +5V DC input

CN9 : 1 x 4 header			
Pin	**Signal**	**Pin**	**Signal**
1	+V5P0	2	+V5P0
3	GND	4	GND

- 2.4.4 : CN4 for +12V DC input

CN12 : 1 x 2 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	GND	2	+12V

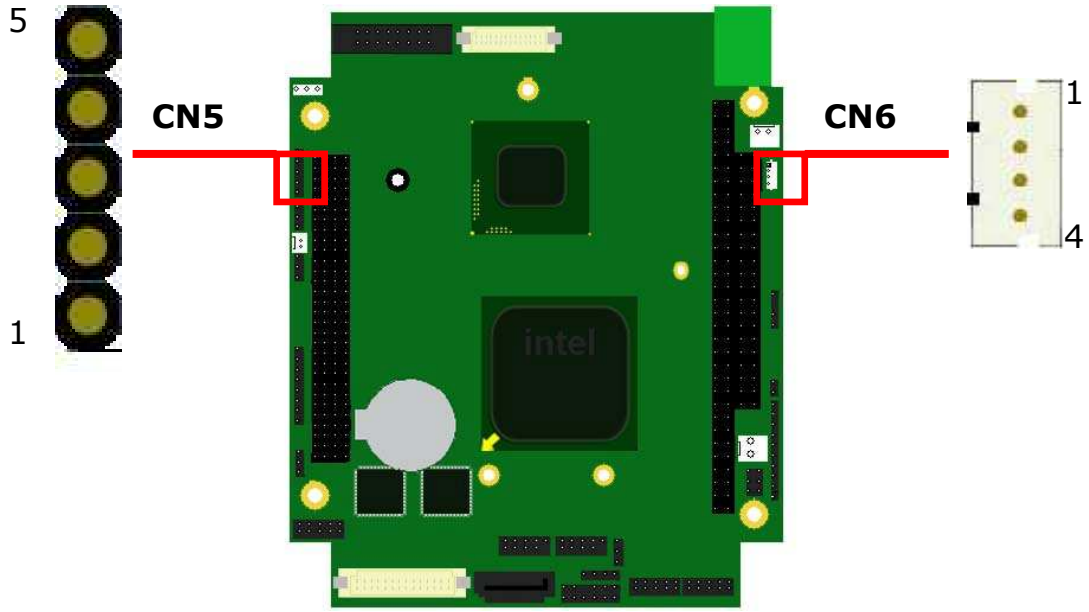


- 2.4.5 : CN5 for LVDS Backlight Control

CN12 : 1 x 5 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+12VP0	2	GND
3	LBKLT_EN_DELAY	4	LCD_BKB_CTRL
5	+5VP0		

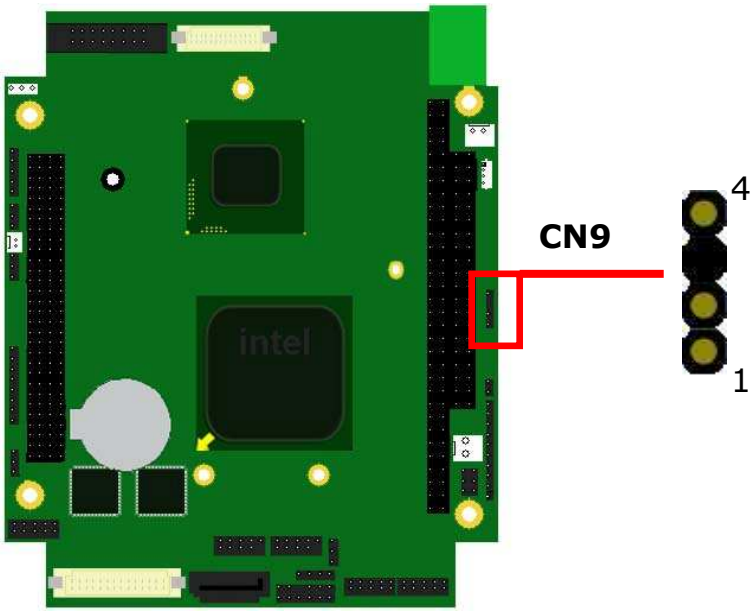
- 2.4.6 : CN6 for +5V_SB DC input and PS_ON

CN13 : 1 x 4 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+V5P0	2	+V5P0
3	GND	4	PS_ON



- 2.4.7 : CN9 for -5V & -12V DC input

CN13 : 1 x 4 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	-V5P0	2	GND
3	N/C	4	-V12P0

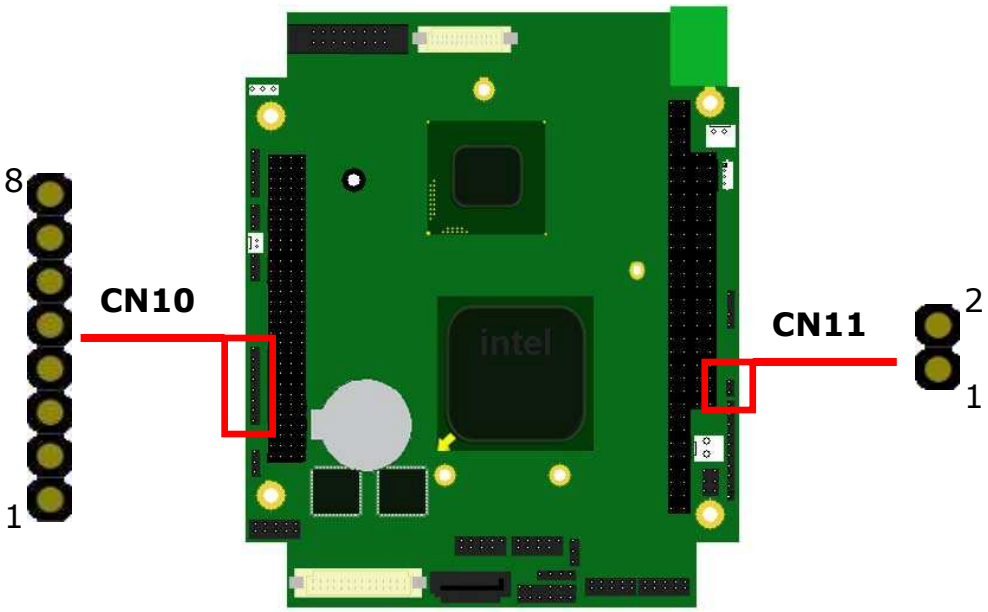


- 2.4.8 : CN10 for Front Panel

CN14 : 1 x 8 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	PWR_LED_N	2	GND
3	HDD_LED+	4	HDD_LE-
5	SYSRST_N	6	GND
7	SPKR_PU	8	BEEP_SPKR_R_N

CN11 for Power Button

CN15 : 1 x 2 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	PWRSW	2	GND

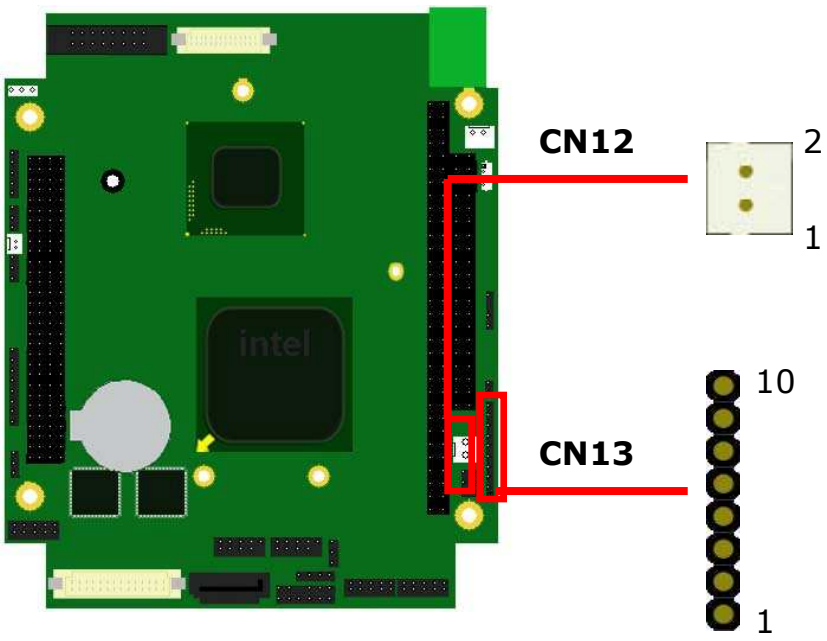


- 2.4.9 : CN12 for Fan connector

CN17 : 1 x 2 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+V5P0	2	GND

CN13 for GPIO ports

CN15 : 1 x 10 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+V3P3	2	GPIO10
3	GPIO11	4	GPIO12
5	GPIO13	6	GPIO14
7	GPIO15	8	GPIO16
9	GPIO17	10	GND

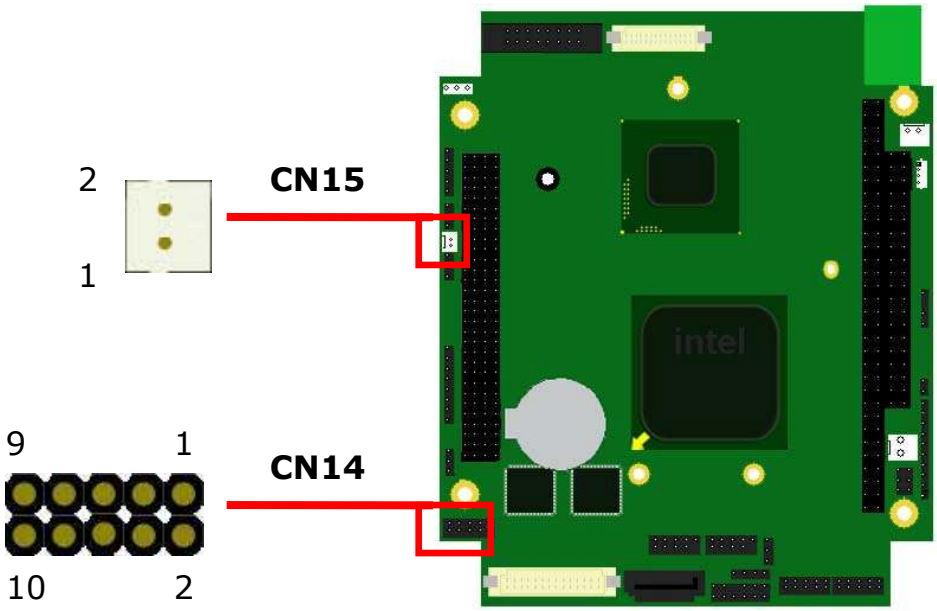


- 2.4.10 : CN14 for LPC connector

CN19 : 2 x 5 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+V3P3	2	LAD0
3	LAD1	4	LAD2
5	LAD3	6	LFRAME_N
7	PLTRST_N	8	+5VP0
9	PORT80_PCLK	10	GND

CN15 for Battery connector

CN19 : 1 x 2 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	BAT2_R	2	GND

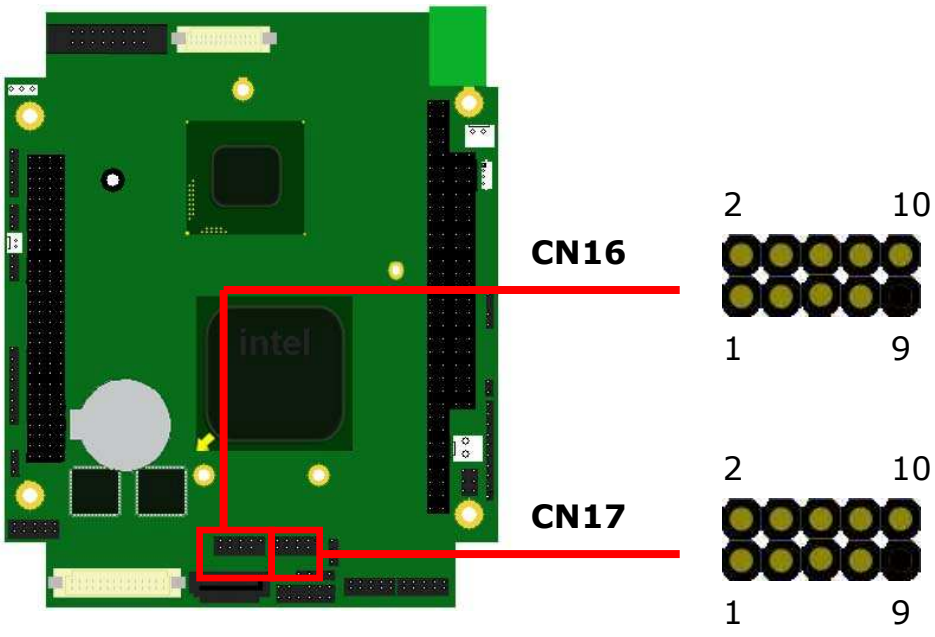


- 2.4.11 : CN16 for USB2 & USB3

CN20 : 2 x 5 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	USBV23	2	USBV23
3	USB_PN_2	4	USB_PN_3
5	USB_PP_2	6	USB_PP_3
7	GND	8	GND
9	N/C	10	GND

CN17 for USB0 & USB1

CN20 : 2 x 5 header , pitch 2.0 mm			
Pin	**Signal**	**Pin**	**Signal**
1	USBV01	2	USBV01
3	USB_PN_0	4	USB_PN_1
5	USB_PP_1	6	USB_PP_1
7	GND	8	GND
9	N/C	10	GND

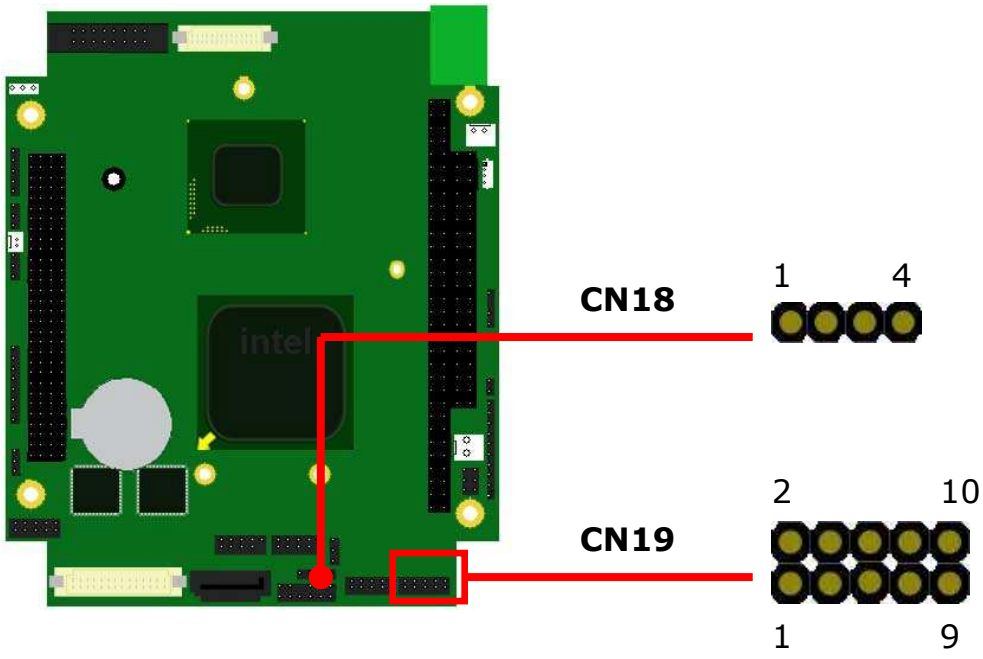


- 2.4.12 : CN18 for COM2 (RS422/485) Connector

CN21 : 1 x 4 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	485_RXD-	2	485RXD+
3	485TXD+	4	485TXD-

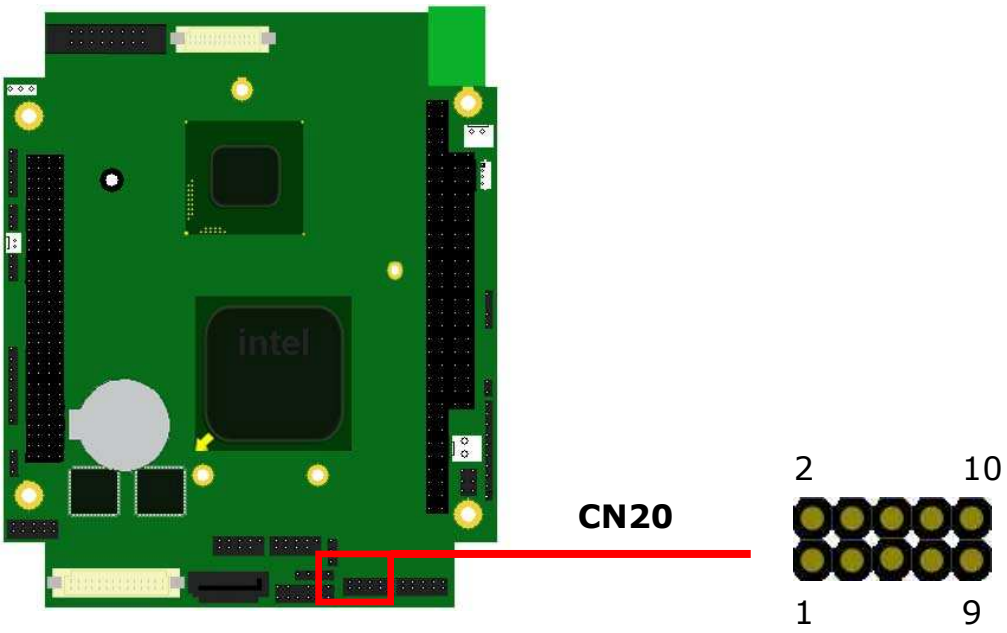
CN19 for COM1 (RS232) Connector

CN21 : 2 x 5 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	DCD_N_CON	2	DSR_N_CON
3	_SIN_CON	4	RTS_N_CON
5	SOUT_CON	6	CTS_N_CON
7	DTR_N_CON	8	RI_N_CON
9	GND	10	N.C



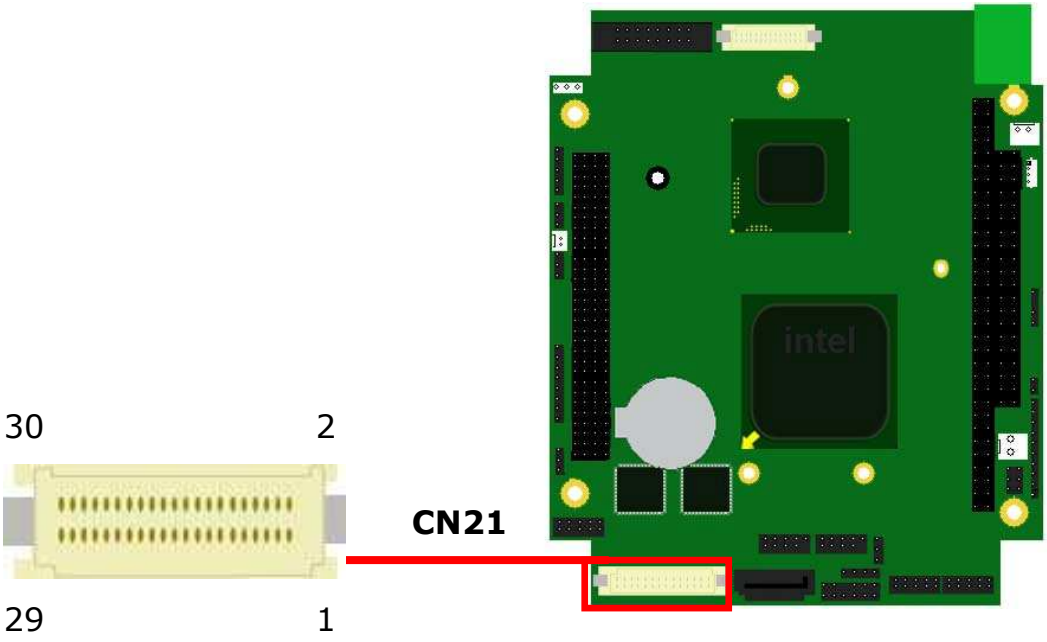
- 2.4.13 : CN20 for COM2 (RS232) Connector

CN20 : 2 x 5 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	DCD_N_CON	2	DSR_N_CON
3	SIN_CON	4	RTS_N_CON
5	SOUT_CON	6	CTS_N_CON
7	DTR_N_CON	8	RI_N_CON
9	GND	10	N/C



- 2.4.14 : CN21 for LAN1 , LAN2 Connector

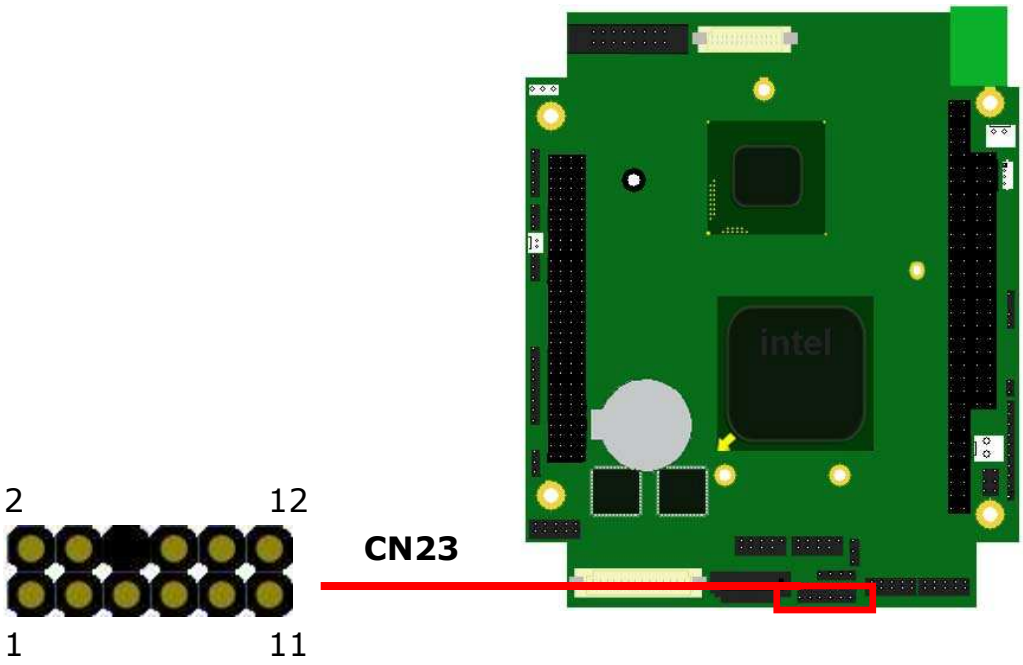
CN20 : 2 x 15 header , connector type : DF13A-30DP-1.25V			
Pin	**Signal**	**Pin**	**Signal**
1	LAN1_L_MDIP2	2	LAN1_L_MDIP0
3	LAN1_L_MDIN2	4	LAN1_L_MDIN0
5	LAN1_L_MDIP3	6	LAN1_L_MDIP1
7	LAN1_L_MDIN3	8	LAN1_L_MDIN1
9	+V3P3	10	GND
11	LAN1_ACT_N_R	12	LAN1_LINK_N
13	LAN1_LINK100_N_R	14	LAN1_LINK1000_N
15	GND	16	GND
17	LAN2_L_MDIP0	18	LAN2_L_MDIP2
19	LAN2_L_MDIN0	20	LAN2_L_MDIN2
21	LAN2_L_MDIP1	22	LAN2_L_MDIP3
23	LAN2_L_MDIN1	24	LAN2_L_MDIN3
25	LAN2_ACT_N_R	26	LAN2_LINK_N
27	LAN2_LINK100_N	28	LAN2_LINK1000_N
29	GND	30	GND



- 2.4.15 : CN23 for HDA Connector

Note : Need to purchase AEWIN's audio module & cable to make audio function work

CN20 : 2 x 6 header , pitch 2.00 mm			
Pin	**Signal**	**Pin**	**Signal**
1	+V5P0	2	GND
3	GND	4	HDA_BIT_CLK
5	+V3P3	6	N/C
7	HDA_SDIN0	8	HDA_SYNC
9	GND	10	HDA_RST_N
11	HDA_SDOUT	12	HDA_SDIN1



3.1 GPIO Sample Program for DOS environment

```
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <stdlib.h>
#include <inlines/pc.h>

#define index_port  0x2E                    //Super IO Index port address
#define data_port    0x2F                   //Super IO Data port address
#define GPIO_port 0x378
#define GPIO_read_portGPIO_port + 3

void Enter_sio_config();
void Exit_sio_config();
void ENABLE_GPIO();
void Input_mode();
void Output_mode();
void help();

int main(int argc, char *argv[])
{
      int data_rw8;

            if (argc<2){
            help();
            return;
      }

      ENABLE_GPIO();

            if(strcmp(argv[1], "-i") == 0){
            Input_mode();
            data_rw8 = inportb(GPIO_read_port);
            if ((data_rw8&0x01) == 0x00)
                  printf("GPI0 -> Low\n");
```

```
else
        printf("GPI0 -> High\n");

if ((data_rw8&0x02) == 0x00)
        printf("GPI1 -> Low\n");
else
        printf("GPI1 -> High\n");

if ((data_rw8&0x04) == 0x00)
        printf("GPI2 -> Low\n");
else
        printf("GPI2 -> High\n");

if ((data_rw8&0x08) == 0x00)
        printf("GPI3 -> Low\n");
else
        printf("GPI3 -> High\n");

if ((data_rw8&0x10) == 0x00)
        printf("GPI4 -> Low\n");
else
        printf("GPI4 -> High\n");

if ((data_rw8&0x20) == 0x00)
        printf("GPI5 -> Low\n");
else
        printf("GPI5 -> High\n");

if ((data_rw8&0x40) == 0x00)
        printf("GPI6 -> Low\n");
else
        printf("GPI6 -> High\n");

if ((data_rw8&0x80) == 0x00)
        printf("GPI7 -> Low\n");
else
        printf("GPI7 -> High\n");
```

```
        }

    else if(strcmp(argv[1], "-h0") == 0){
        printf("GPO0 -> High\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xFE;
        data_rw8 |= 0x01;
        outportb(GPIO_port, data_rw8);
    }
    else if(strcmp(argv[1], "-l0") == 0){
        printf("GPO0 -> Low\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xFE;
        outportb(GPIO_port, data_rw8);
    }

    else if(strcmp(argv[1], "-h1") == 0){
        printf("GPO1 -> High\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xFD;
        data_rw8 |= 0x02;
        outportb(GPIO_port, data_rw8);
    }
    else if(strcmp(argv[1], "-l1") == 0){
        printf("GPO1 -> Low\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xFD;
        outportb(GPIO_port, data_rw8);
    }

    else if(strcmp(argv[1], "-h2") == 0){
        printf("GPO2 -> High\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xFB;
        data_rw8 |= 0x04;
        outportb(GPIO_port, data_rw8);
```

```
}
else if(strcmp(argv[1], "-l2") == 0){
        printf("GPO2 -> Low\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xFB;
        outportb(GPIO_port, data_rw8);
}

else if(strcmp(argv[1], "-h3") == 0){
        printf("GPO3 -> High\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xF7;
        data_rw8 |= 0x08;
        outportb(GPIO_port, data_rw8);
}
else if(strcmp(argv[1], "-l3") == 0){
        printf("GPO3 -> Low\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xF7;
        outportb(GPIO_port, data_rw8);
}

else if(strcmp(argv[1], "-h4") == 0){
        printf("GPO4 -> High\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xEF;
        data_rw8 |= 0x10;
        outportb(GPIO_port, data_rw8);
}
else if(strcmp(argv[1], "-l4") == 0){
        printf("GPO4 -> Low\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xEF;
        outportb(GPIO_port, data_rw8);
}

else if(strcmp(argv[1], "-h5") == 0){
```

```
        printf("GPO5 -> High\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xDF;
        data_rw8 |= 0x20;
        outportb(GPIO_port, data_rw8);
}
else if(strcmp(argv[1], "-l5") == 0){
        printf("GPO5 -> Low\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xDF;
        outportb(GPIO_port, data_rw8);
}

else if(strcmp(argv[1], "-h6") == 0){
        printf("GPO6 -> High\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xBF;
        data_rw8 |= 0x40;
        outportb(GPIO_port, data_rw8);
}
else if(strcmp(argv[1], "-l6") == 0){
        printf("GPO6 -> Low\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0xBF;
        outportb(GPIO_port, data_rw8);
}

else if(strcmp(argv[1], "-h7") == 0){
        printf("GPO7 -> High\n");
        Output_mode();
        data_rw8 = inportb(GPIO_port)&0x7F;
        data_rw8 |= 0x80;
        outportb(GPIO_port, data_rw8);
}
else if(strcmp(argv[1], "-l7") == 0){
        printf("GPO7 -> Low\n");
        Output_mode();
```

```
        data_rw8 = inportb(GPIO_port)&0x7F;
        outportb(GPIO_port, data_rw8);
    }

    else{
        help();
        return;
    }

    return;
}

void Enter_sio_config()                 //Enter F81865 Configuration
{
    outportb(index_port, 0x87);
    delay(1);                   //Delay some time
    outportb(index_port, 0x87);
    outportb(index_port, 0x07);             //Super IO Selct Bank Register Number
    outportb(data_port , 0x03);             //Select logical device 7
    outportb(index_port, 0x30);
    outportb(data_port , 0x01);
}

void Exit_sio_config()                  //Exit F81865 Configuration
{
    outportb(index_port, 0xAA);
}

void ENABLE_GPIO()
{
    int BusNum    = 0x00;   //bus0
    int DevFunc = 0xA3;     //device20, function3
    int RegNum    = 0x44;   //reg44
    long int data_rw32;

    data_rw32 = (BusNum << 8) + (DevFunc << 8);
    data_rw32 += (RegNum & 0xFC);
```

```
	data_rw32 |= 0x80000000;
	outportl(0xCF8, data_rw32);
	outportb(0xCFC, 0xC1);
}

void Input_mode()
{
	Enter_sio_config();
	outportb(index_port, 0xF0);
	outportb(data_port,    0x41);
	Exit_sio_config();
}

void Output_mode()
{
	Enter_sio_config();
	outportb(index_port, 0xF0);
	outportb(data_port,    0x40);
	Exit_sio_config();
}

void help()
{
	printf("AEWIN GPIO Program\n");
	printf("Usage: GPIO -i    (Show GPI Settings)\n");
	printf("Usage: GPIO -hx (Set GPO Value to high)\n");
	printf("                        x = 0 ~ 7\n");
	printf("Usage: GPIO -lx (Set GPO Value to low)\n");
	printf("                        x = 0 ~ 7\n");
}
```

3.2 Watchdog timer Sample Program for DOS environment

```
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <stdlib.h>
#include <inlines/pc.h>

#define index_port  0x2E                    //Super IO Index port address
#define data_port    0x2F                   //Super IO Data port address

void Enter_sio_config();
void Exit_sio_config();
void help();

int main(int argc, char *argv[])
{
      int data_rw8;

            if (argc<2){
            help();
            return;
      }

            if(strcmp(argv[1], "-s") == 0){
                  //Show Watchdog Register Settings
            Enter_sio_config();
            outportb(index_port, 0xF5);
            data_rw8 = inportb(data_port)&0x08;
            if(data_rw8 == 0x00){
                  //second mode
                  outportb(index_port, 0xF6);
                  data_rw8 = inportb(data_port);
                  printf("Second mode: %d second\n", data_rw8);
            }
            else{
```

```
            //minute mode
            outportb(index_port, 0xF6);
            data_rw8 = inportb(data_port);
            printf("Minute mode: %d minute\n", data_rw8);
        }
        }
    else if(strcmp(argv[1], "-t") ==0 ){
        //Set Time-out Value
        if(argv[2] == NULL){
            help();
            return;
        }
        else{
            Enter_sio_config();
            outportb(index_port, 0xFA);
            data_rw8 = inportb(data_port)|0x01;
            outportb(data_port, data_rw8);

            outportb(index_port, 0xF5);
            data_rw8 = inportb(data_port)&0xF7|0x31;
            outportb(data_port, data_rw8);
                sscanf(argv[2], "%d", &data_rw8);
            outportb(index_port, 0xF6);
            outportb(data_port,    data_rw8);
                printf("Watchdog Timer will count down for %d second(s)\n", data_rw8);
        }
    }
    else if(strcmp(argv[1], "-m") ==0 ){
        //Set Time-out Value
        if(argv[2] == NULL){
            help();
            return;
        }
        else{
            Enter_sio_config();
            outportb(index_port, 0xFA);
            data_rw8 = inportb(data_port)|0x01;
```

```
                outportb(data_port, data_rw8);

                outportb(index_port, 0xF5);
                data_rw8 = inportb(data_port)&0xF7|0x31;
                data_rw8 |= 0x08;
                outportb(data_port, data_rw8);
                    sscanf(argv[2], "%d", &data_rw8);
                outportb(index_port, 0xF6);
                outportb(data_port,    data_rw8);
                    printf("Watchdog Timer will count down for %d minute(s)\n", data_rw8);
            }
        }
        Exit_sio_config();
        return;
}

void Enter_sio_config()                         //Enter F81865 Configuration
{
        outportb(index_port, 0x87);
        delay(1);                       //Delay some time
        outportb(index_port, 0x87);
        outportb(index_port, 0x07);             //Super IO Selct Bank Register Number
        outportb(data_port , 0x07);             //Select logical device 7
        outportb(index_port, 0x30);
        outportb(data_port , 0x01);
}

void Exit_sio_config()                          //Exit F81865 Configuration
{
        outportb(index_port, 0xAA);
}

void help()
{
        printf("AEWIN Watchdog Timer Program\n");
        printf("Usage: WDT -s         (Show Watchdog Register Settings)\n");
        printf("Usage: WDT -t xxx (Set Time-out Value)\n");
```

```
    printf("                xxx = 1 ~ 255 seconds\n");
    printf("                xxx = 0 : Time-out Disable \n");
    printf("Usage: WDT -m xxx (Set Time-out Value)\n");
    printf("                xxx = 1 ~ 255 minutes\n");
    printf("                xxx = 0 : Time-out Disable \n");
}
```