

# USB AUDIO CARD

## Abstract:

This project completes a USB(Universal serial bus) audio card with Atmega162L-8PI.

In most computers, the audio device belongs to PCI device. If you want to add a new PCI audio device into your PC, you have to open the computer case, find an idle PCI slot , and plug this card in, then install the driver for the device. This deed is complex and dangerous, because any wrong operation may destroy your motherboard. Now USB audio device has been used in computer system more and more. It is simpler and much more convenient than PCI device.

With a USB B-type to A-type cable, my audio card can be easily connected with computer. After a few minutes, Windows system will tell you a new composite USB device is found, and then audio device driver is installed automatically. In the device manager, a USB Audio device appears. After you select the USB audio device as preferred audio device, windows system will transfer audio data to this card. After connecting earphone with this card, you can hear voice when you play a MP3 or WAV file.

This card has a indicator led, a reset key and 3 interfaces.

- a) A RS-232 interface, which is used for debugging.
- b) A USB B-Port, which is used to transfer data between the computer and audio card.
- c) A 3.5mm Audio connector, which is used to connect earphone or speaker.

This USB audio card has USB1.1 feature. I use the PDIUSB12 from Philip as the USB interface. Whose data transfer rate can achieve 1Mbits/s in isochronous mode. This chip just is a transceiver. So you can change it to be USB2.0 interface chip such as ISP1581 from Philips.

ATMEGA162L is a central processor. It mainly processes the USB audio protocol, and transfer data between USB module and AUDIO codec module. In the beginning, it also initialize usb chip, cpld , and audio codec.

CPLD is really a frequency divider and a 16bits parallel to serial converter. Which generate the syn clk and bit clock, and receive 16bits parallel data from Atmega162 and transmit serial data to Audio codec.

PCM3000E is 16/18-bits stereo audio codec, which can be programmed with 3-wire serial interface. Analog audio is produced.

TDA2822 is a dual audio power amplifier, which receives the audio from PCM3000E then drives the earphone or speaker.

ISL83220 is a RS232 transceiver, it is only used when debugging the program running in the atmega162L.

This audio card's power is supplied through USB port. The USB port provides 5V power. But some chips need 3.3V power, so I use a LM337 to produce 3.3V.

### Block Diagram

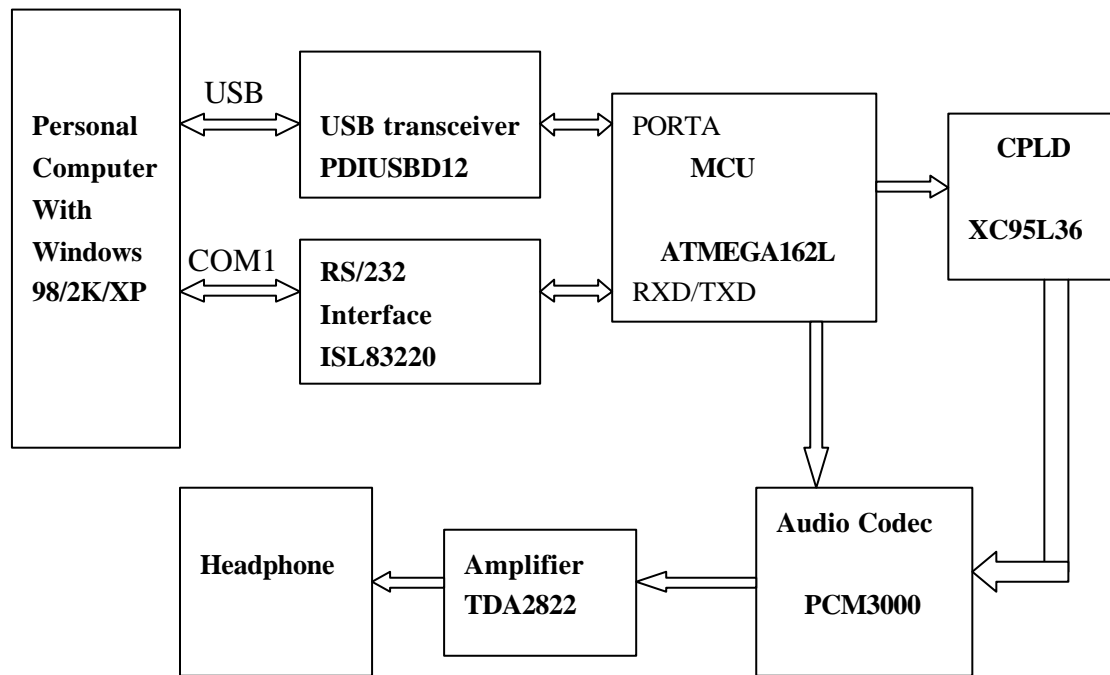
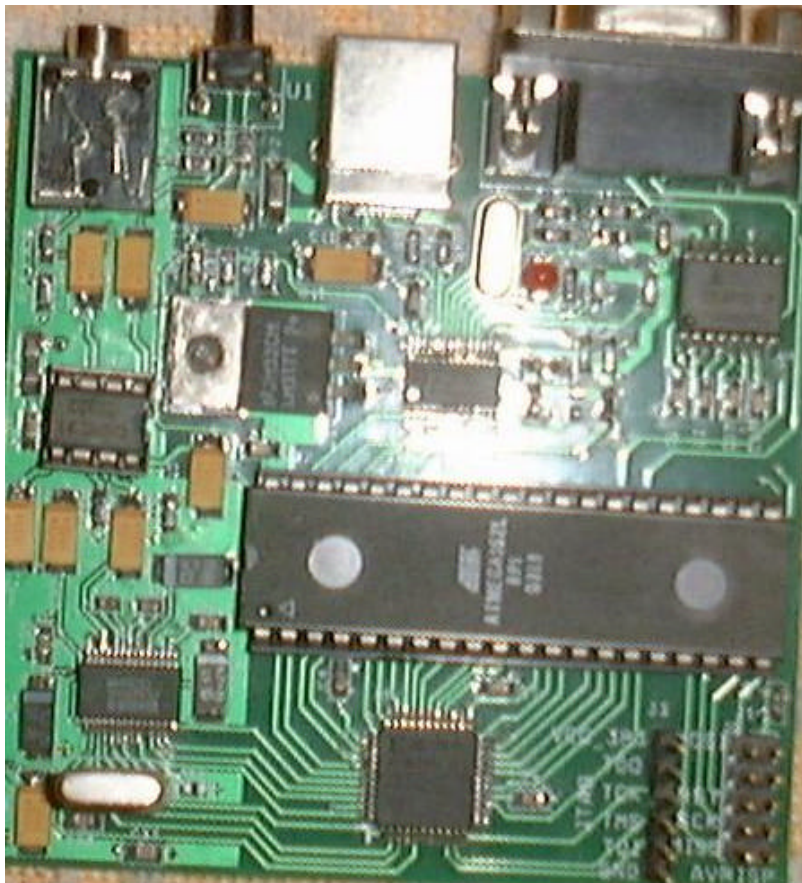


Figure 1 the block diagram of the usb audio card

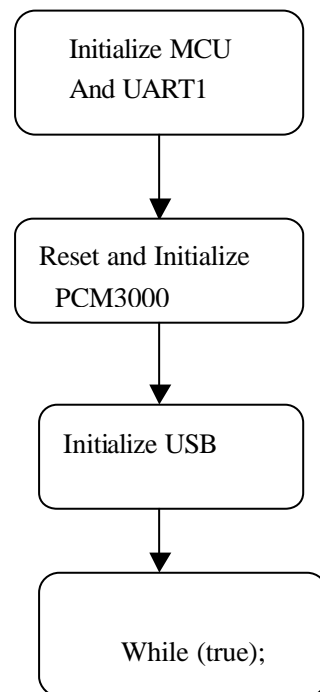
### Photos of the circuit



## The software structure of the MCU

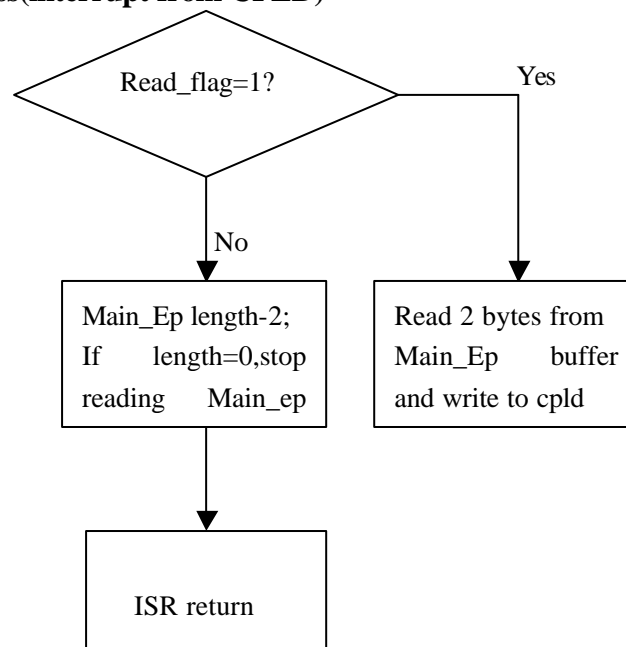
### 1:Main process

i



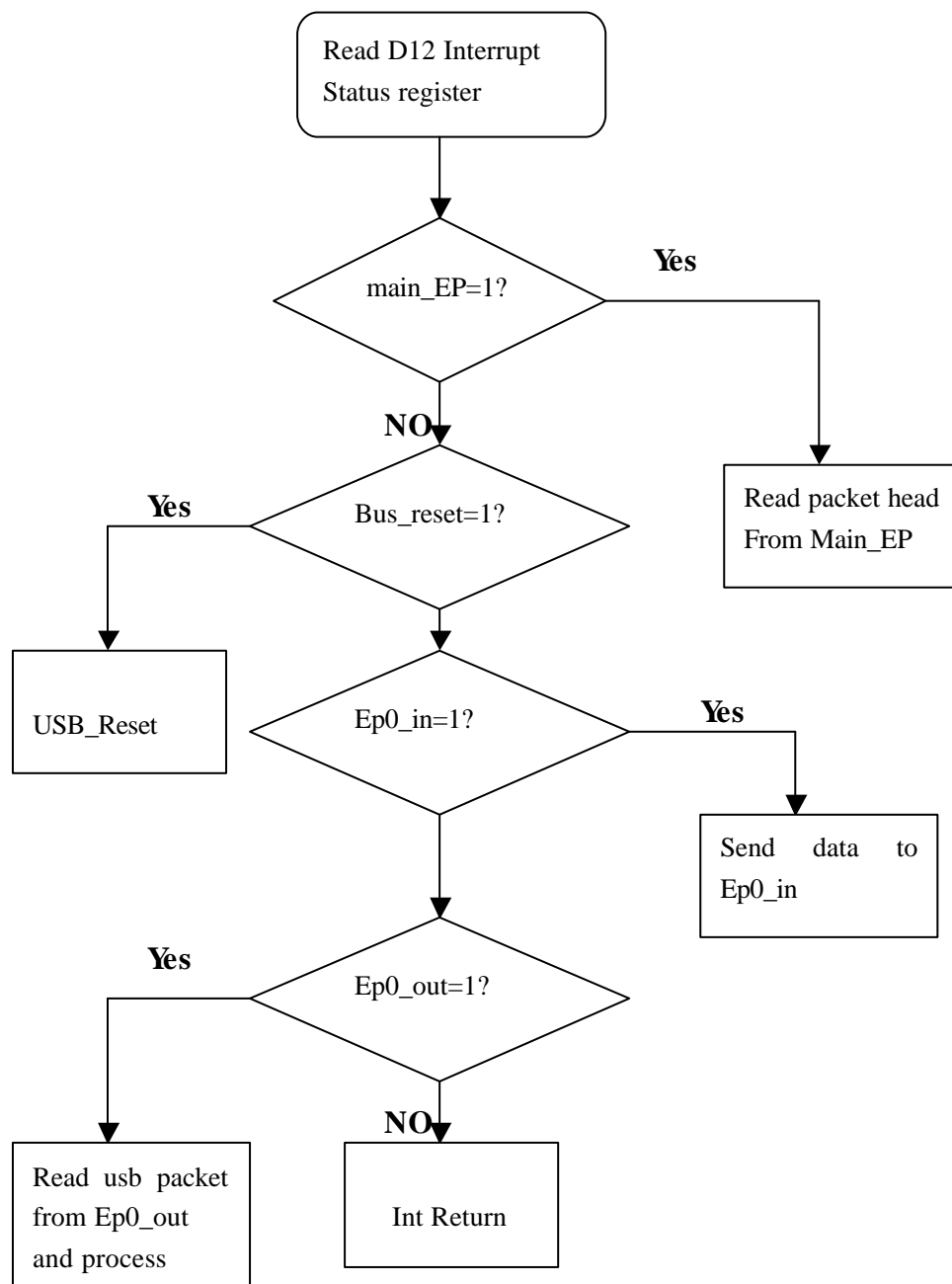
In the Main( ) function, first initialize the atmega162(port, uart), then reset the Audio codec and configure it's register group using 3 wires:CLK,DATA,LOAD.Last initialize the usb interface chip, set endpoints enable and select ISO OUT mode because the 44.1KHz/16bits audio format.

### 2:ISR\_number2 process(interrupt from CPLD)



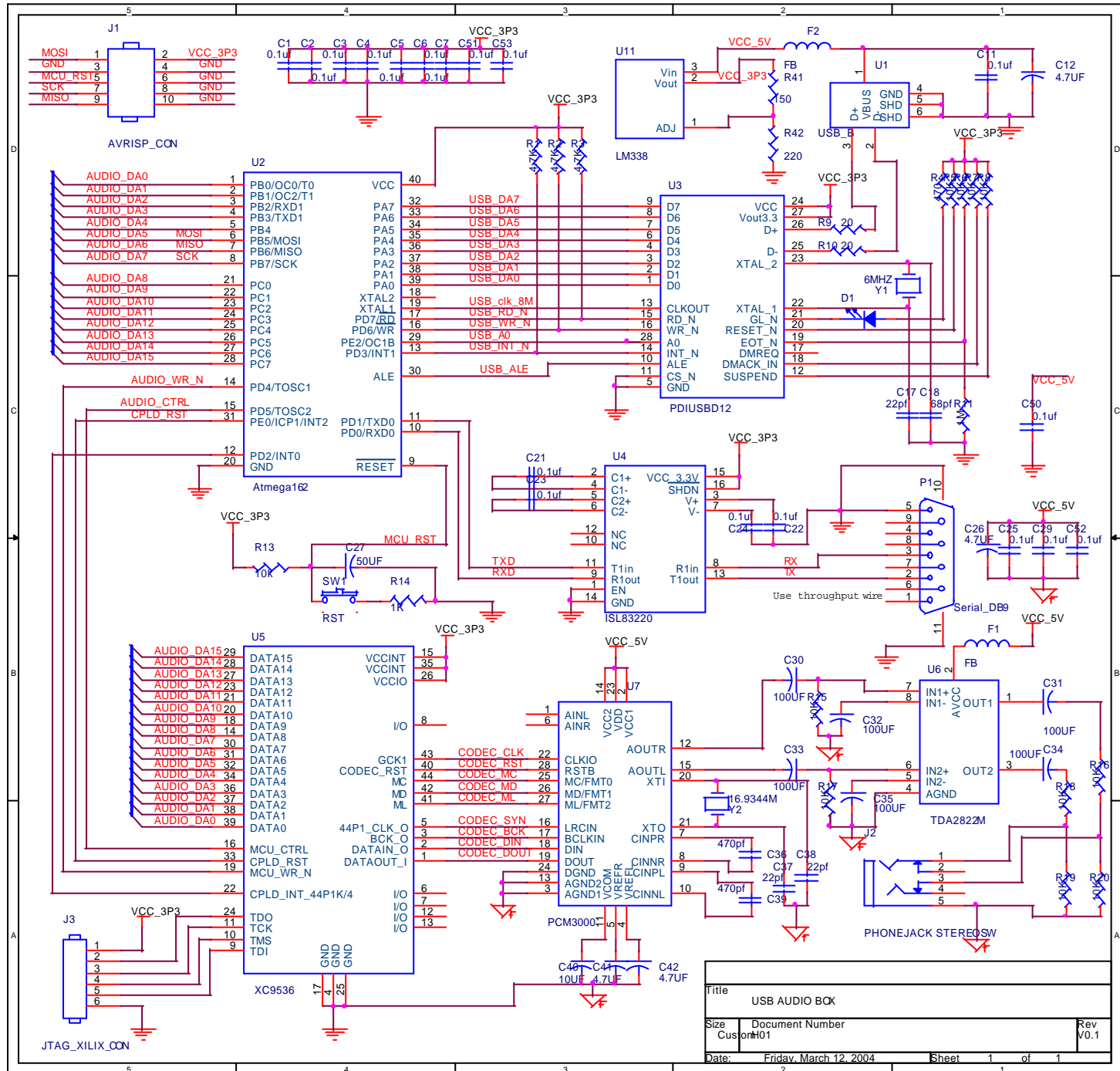
The audio data is transferred into Endpoint2\_out(Main\_Ep\_out) buffer. When CPLD request data, the MCU read one word (2 bytes) from buffer and place it in PB[7:0](low byte)&PC[7:0](high byte) .

### 3:ISR\_number3 process (interrupt from USB D12)



Once the MCU set the Softconnect flag in the USB chip, Windows system will send setup packets to the USB control endpoint(Ep0) one by one. Then USB interrupt MCU to process these packets. MCU will read packet from USB EP0 buffer and acknowledge these request such as get\_device\_descriptor,get\_configure\_descriptor and so on. In this process, MCU notify Windows system that it is a usb audio device. If Windows system finds this is an available device, USB audio device driver will be loaded. At the late time, you can use the USB audio device as your PCI audio card.

### Circuit schematic chart



## Software code samples

```
//*****
//*****USB AUDIO CARD SOURCE CODE IN ATMEGA162*****
//Written by Haventchen
//2004-3-10 21:08
//ICC-AVR application builder: 2004-3-10 21:51:30
// Target: M162
// Crystal: 8.0000Mhz

//*****
//**                               MAIN PROCESS
//*****

void main(void)
{extern int _textmode;
_textmode = 1;
/**Reset the PCM3000*****
PORTE=0x06;
delay_time(40);
PORTE=0x07; /*******
/**initial the Atmega162*****
init_devices();
uart0_init();
/**initial the PCM3000*****
    pcm_init(0x0035);          //reg0 ,
    pcm_init(0x0135);          //reg0 ,ld
    /*******
    pcm_init(0x0235);          //reg1 ,
    pcm_init(0x0335);          //reg1 ,ld
    /*******
    pcm_init(0x0404);          //reg2
    pcm_init(0x0608);          //reg2,ld
    /**initial the PDIUSB D12
    dl2_init();
    /**Enable Interrupt
    SEI();
    while(1);
}
```

```

//*****
//*                               ISR from CPLD
//*****
#pragma interrupt_handler int0_isr:2
void int0_isr(void) //external interupt on INT0
{
    if(DAC_enable)
    {
        if(flag)
        {
            PORTB=DATA;
            _NOP();
            _NOP();
            PORTC=DATA;
            flag=0;
        }
        else
        {
            flag=1;
            transmit_status.Main_len=transmit_status.Main_len-2;
            if(transmit_status.Main_len==0)
            {
                COMMAND=0xf2;
                DAC_enable=0;
            }
        }
    }
}

//*****
//*                               ISR from USB Transceiver
//*****
#pragma interrupt_handler int1_isr:3
void int1_isr(void)
{
    unsigned char intreg1;
    unsigned char intreg2;
    unsigned char ep_buffer;
    COMMAND=0xf4; //read int reg;
    _NOP();
    _NOP();
    intreg1=DATA;
    _NOP();
    _NOP();
    intreg2=DATA;
}

```

```

    if (intreg1&MAIN_OUT)
    {COMMAND=0x44;//read last status
    _NOP();
    _NOP();
    ep_buffer=DATA;

    COMMAND=0x04;//select ep2_out
    _NOP();
    _NOP();
    ep_buffer=DATA;

    COMMAND=0xf0;//read the ep2_out buffer
    _NOP();
    _NOP();
    ep_buffer=DATA; //this byte is length
    _NOP();
    _NOP();
    transmit_status.Main_len=DATA;
    _NOP();
    _NOP();
    DAC_enable=1;    //enable the DAC
    }
    else if (intreg1&BUS_RST) bus_reset();
    else if (intreg1&EP0_IN) ep0_in();
    else if (intreg1&EP0_OUT) ep0_out();
    }

```