

## AT32 USB Audio Development Note

### 文档说明

“AT32 USB Audio Development Note”完成对USB Audio Demo实现的说明，Demo包含了USB传输类型（Control，Isochronous）。

文档目标是让用户能够更快的使用Artery AT32Fxx系列的USB外设进行USB 设备开发。

目前这个文档介绍如下AT32Fxx USB 相关内容：

- AT32\_USB-FS-Device\_Driver：关于USB协议层的库
- Audio\_Composite\_Speaker\_MicroPhone：isochronous的使用，实现USB Speaker 和 USB MicroPhone

支持型号列表 T1

支持型号	AT32F403xx

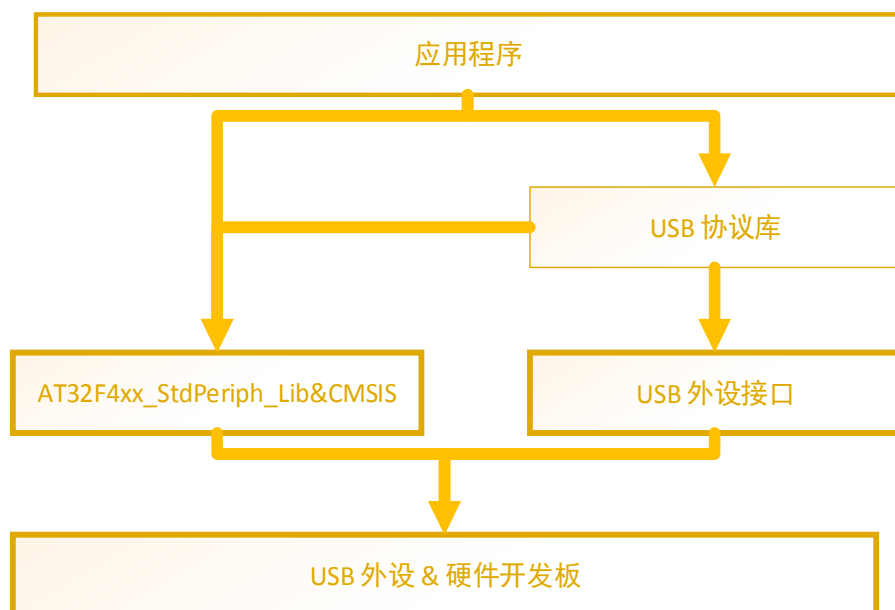
## 目录

<b>AT32 USB AUDIO DEVELOPMENT NOTE .....</b>	<b>1</b>
<b>1 AT32 USB-FS-DEVICE 协议库 .....</b>	<b>3</b>
1.1 AT32 USB 库文件 .....	3
<b>2 AUDIO_COMPOSITE_SPEAKER_MICROPHONE .....</b>	<b>4</b>
2.1 功能描述 .....	4
2.2 ISOCHRONOUS传输简介 .....	5
2.3 USB AUDIO DEVICE .....	5
2.4 AT32 USB AUDIO SPEAKER AND MICROPHONE.....	5
2.5 运行环境与使用资源 .....	6
2.6 软件实现 .....	6
2.6.1 USB时钟配置 .....	7
2.6.2 USB端点初始化 .....	8
2.6.3 Speaker 和 MicroPhone 设备配置 .....	9
2.6.4 MicroPhone 数据处理 .....	15
2.6.5 Speaker 数据处理 .....	15
<b>3 USB AUDIO测试使用说明 .....</b>	<b>17</b>
3.1 硬件资源 .....	17
3.2 使用方法步骤 .....	17
3.3 播放声音 .....	18
3.4 录音测试 .....	19
<b>4 版本历史.....</b>	<b>20</b>

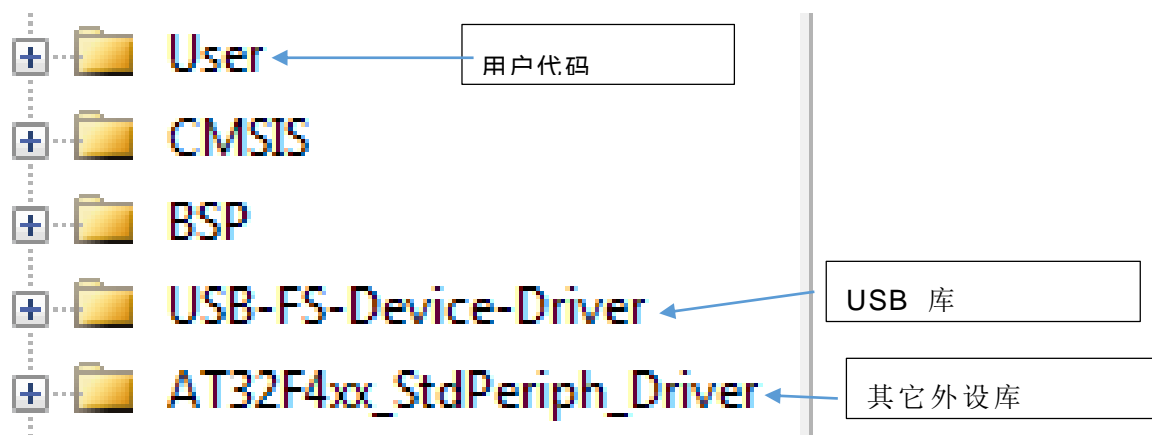
## 1 AT32 USB-FS-Device 协议库

这部分主要介绍AT32 USB库的结构和库的使用方法，AT32 USB是基于USB2.0全速设备，不支持USB2.0高速设备。这里库的作用是用来管理USB外设和实现USB的基本协议，使用户能够更快的上手开发。

### T2 USB应用结构



### T3 AT32工程结构



## 1.1 AT32 USB 库文件

### T4 USB库包含的文件列表

文件名	内容
usb_core	实现USB Specification 协议
usb_init	初始化USB全局变量
usb_int	处理USB传输完成中断

usb_mem	用户缓冲区和USB缓冲区的数据交换
usb_regs	实现对USB register的操作
usb_sil	实现用户缓冲和USB缓冲数据交换的封装

T5 USB用户接口文件列表

文件名	内容
usb_desc	实现应用程序对USB设备的描述
usb_istr	获取USB中断并进行处理
usb_prop	实现USB USER STANDARD REQUEST请求
usb_pwr	管理USB Power
hw_config	其它配置

## 2 Audio\_Composite\_Speaker\_MicroPhone

Audio例程运行在AT-START评估板上，Audio Speaker 和 MicroPhone是基于Audio Arduino Demo Board 进行实现，实验过程中需要连接AT-START和Audio Arduino Board，更多信息参考《UM\_Audio Arduino Daughter Board\_V1.0》。Audio 设备基于USB Audio Device 1.0，详细内容可参考《Universal Serial Bus Device Class Define for Audio Device V1.0》。

### 2.1 功能描述

实现一个USB Audio的复合设备，包含一个Speaker和一个MicroPhone，可以同时进行播放声音和录制声音，Speaker使用isochronous OUT传输类型，MicroPhone使用isochronous IN传输类型。

Speaker功能：

- 支持16K，48K采样率(默认只支持16K)
- 支持采样率切换
- 支持16bit采样
- 支持静音
- 支持音量调节

- 支持Feedback

MicroPhone功能：

- 支持16K，48K采样率（默认只支持16K）
- 支持采样率切换
- 支持16bit采样
- 支持静音
- 支持音量调节

**注意：**Speaker和Microphone的采样率必须相同，否则Speaker和MicroPhone会有一个功能不能正常使用。

## 2.2 isochronous传输简介

Isochronous传输主要用于传输数据量大，实时性要求高的场景，可以使Host到Device，也可以是Device到Host，在传输过程当中没有握手包ACK。

如下是一个isochronous OUT传输

Transfer 1		F	S	Isoch	OUT	ADDR	27	ENDP	2	Bytes Transferred	1291008	Isochronous Packet Info				Time Stamp									
												6724 packets ranging from 192 bytes to 192 bytes				0.001170632									
Transaction 1		F	S		OUT	ADDR	27	ENDP	2	T	0	Data	192 bytes	Time Stamp											
												0.001170632													
On	Packet 4		F	S	H	Sync	00000001	OUT	0x87	ADDR	27	ENDP	2	CRC5	0x17	EOP	250.000 ns	Pkt Len	35 Bits (5 Bytes)	Duration	2.917 us	Idle	351.330 ns	Time Stamp	0.001170632
	On	Packet 5		F	S	H	Sync	00000001	DATA0	0xC3	Data	192 bytes	CRC16	0x3FC8	EOP	250.000 ns	Pkt Len	1597 Bits (200 Bytes)	Duration	133.083 us	Time	994.616 us	Time Stamp	0.001173900	
Transaction 3		F	S		OUT	ADDR	27	ENDP	2	T	0	Data	192 bytes	Time	1.009 ms	Time Stamp									
												0.002168516													

如下是一个isochronous IN传输

Transfer		F	Isoch	ADDR	ENDP	Bytes Transferred		Isochronous Packet Info			Time Stamp		
4		S	IN	27	1	3242112		16886 packets ranging from 192 bytes to 192 bytes			3 . 596 482 082		
Transaction		F	IN	ADDR	ENDP	T	Data	Time Stamp					
3589		S	0x96	27	1	0	192 bytes	3 . 596 482 082					
D O	Packet	H	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Pkt Len	Duration	Idle	Time Stamp
	10791	S	00000001	0x96	27	1	0x0E	250.000 ns	35 Bits (5 Bytes)	2.917 us	467.330 ns	3 . 596 482 082	
D O	Packet	U	F	Sync	DATA0	Data	CRC16	EOP	Pkt Len	Duration	Time	Time Stamp	
	10792	S	00000001	0xC3	192 bytes	0x9C53	266.660 ns	1708 Bits (214 Bytes)	142.267 us	996.616 us	3 . 596 485 466		
Transaction		F	IN	ADDR	ENDP	T	Data	Time		Time Stamp			
3590		S	0x96	27	1	0	192 bytes	1.000 ms		3 . 597 482 082			

## 2.3 USB Audio Device

《Universal Serial Bus Device Class Define for Audio Device》中定义了USB Audio Device的实现。里面定义如何实现一个Audio设备以及一些控制如声音等。

## 2.4 AT32 USB Audio Speaker and MicroPhone

AT32 MCU通过USB与PC进行数据传输，并通过I2S与WM8988进行通信。



## 2.5 运行环境与使用资源

MCU外设资源：

USB：

- 端点0 Control：用于USB枚举和USB 控制（音量，采样率等）
- 使用端点1作为isochronous IN：AT32 MCU发送数据到PC，MicroPhone功能
- 使用端点2作为isochronous OUT：PC发送数据到AT32 MCU，Speaker功能

I2C：

- 使用I2C1发送控制信息到音频Board

I2S：

- 使用I2S1发送数据到音频Board（Speaker）
- 使用I2S2从音频board接收数据（MicroPhone）

DMA1：

- 使用DMA1\_Channel3传输I2S1的数据
- 使用DMA1\_Channel4传输I2S2的数据

TMR1：

- 使用TMR1 产生12MHz Clock 给音频Board

Audio属性：

- Audio 格式：TYPE I/PCM8 Format /Stereo
- Audio Resolution: 16Bit
- Audio Sample Frequency:16KHz/48KHz

硬件环境：

- 参考《UM\_Audio Arduino Daughter Board\_V1.0》

## 2.6 软件实现

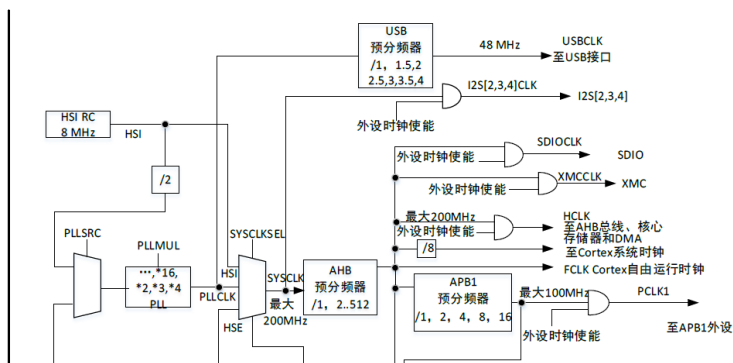
- USB外设的初始化，如时钟配置，端点配置，端点缓冲区的配置
- USB作为Speaker设备的描述

- USB作为MicroPhone设备的描述

## 2.6.1 USB时钟配置

USB 48MHz时钟来源：

USB时钟根据PLLCLK分频得到，分频参数可以设置为0, 1, 1.5, 2, 2.5, 3, 3.5, 4



代码实现：

根据不同的系统频率，配置不同的USB分频参数。

```
void Set_USBClock(uint8_t bDivByPRE)
{
    switch (SystemCoreClock)
    {
        case 48000000:
            RCC_USBCLKConfig(RCC_USBCLKSelection_PLL_Div1);
            break;
        case 72000000:
            RCC_USBCLKConfig(RCC_USBCLKSelection_PLL_Div1_5);
            break;
        case 96000000:
            RCC_USBCLKConfig(RCC_USBCLKSelection_PLL_Div2);
            break;
        case 120000000:
            RCC_USBCLKConfig(RCC_USBCLKSelection_PLL_Div2_5);
            break;
        case 144000000:
            RCC_USBCLKConfig(RCC_USBCLKSelection_PLL_Div3);
            break;
        case 168000000:
            RCC_USBCLKConfig(RCC_USBCLKSelection_PLL_Div3_5);
            break;
        case 192000000:
            RCC_USBCLKConfig(RCC_USBCLKSelection_PLL_Div4);
            break;
        default:
            break;
    }
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_USB, ENABLE);
}
```

## 2.6.2 USB端点初始化

对要使用到的端点进行初始化,包括端点类型和端点最大支持的长度等。端点初始化在设备收到主机发送Reset信号时进行初始化。

```
void USB_Audio_Reset()
{
    /* Set USB_Audio device as not configured state */
    pInformation->Current_Configuration = 0;
    /* Current Feature initialization */
    pInformation->Current_Feature = Audio_ConfigDescriptor[7];
    _SetBUFTBL(BTABLE_ADDRESS);
    /* Initialize Endpoint 0 */ 初始化端点0为控制端点
    SetEPTType(ENDP0, EP_CONTROL);
    SetEPTxStatus(ENDP0, EP_TX_NAK);
    SetEPRxAddr(ENDP0, ENDP0_RXADDR);
    SetEPRxCount(ENDP0, Device_Property.MaxPacketSize);
    SetEPTxAddr(ENDP0, ENDP0_TXADDR);
    Clear_Status_Out(ENDP0);
    SetEPRxValid(ENDP0);
    /* Initialize Endpoint 1 */
    SetEPTType(ENDP1, EP_ISOCHRONOUS); 初始化端点1为同步IN端点
    SetEPDblBuffAddr(ENDP1, ENDP1_BUF0Addr, ENDP1_BUF1Addr);
    SetEPDblBuffCount(ENDP1, EP_DBUF_IN, USB_EDNP_MAX_PACKET);
    ClearDTOG_RX(ENDP1);
    ClearDTOG_TX(ENDP1);
    ToggleDTOG_RX(ENDP1);
    SetEPRxStatus(ENDP1, EP_RX_DIS);
    SetEPTxStatus(ENDP1, EP_TX_VALID);
    /* Initialize Endpoint 2 */
    SetEPTType(ENDP2, EP_ISOCHRONOUS); 初始化端点2为同步OUT端点
    SetEPDblBuffAddr(ENDP2, ENDP2_BUF0Addr, ENDP2_BUF1Addr);
    SetEPDblBuffCount(ENDP2, EP_DBUF_OUT, USB_EDNP_MAX_PACKET);
    ClearDTOG_RX(ENDP2);
    ClearDTOG_TX(ENDP2);
    ToggleDTOG_TX(ENDP2);
    SetEPRxStatus(ENDP2, EP_RX_VALID);
    SetEPTxStatus(ENDP2, EP_TX_DIS);

    SetEPTType(ENDP3, EP_ISOCHRONOUS); 初始化端点3作为feedback端点
    SetEPDblBuffAddr(ENDP3, ENDP3_BUF0Addr, ENDP3_BUF1Addr);
    SetEPDblBuffCount(ENDP3, EP_DBUF_IN, USB_EDNP_MAX_PACKET);
    ClearDTOG_RX(ENDP3);
    ClearDTOG_TX(ENDP3);
    ToggleDTOG_RX(ENDP3);
    SetEPRxStatus(ENDP3, EP_RX_DIS);
    SetEPTxStatus(ENDP3, EP_TX_DIS);
    SetEPRxValid(ENDP0);
    /* Set this device to response on default address */
    SetDeviceAddress(0);

    bDeviceState = ATTACHED;
}
```

```
/* EP0 */
/* rx/tx buffer base address */
#define ENDP0_RXADDR (0x20)
#define ENDP0_TXADDR (0x60)

/* EP1 */
/* buffer base address */
#define ENDP1_BUF0Addr (0x220)
#define ENDP1_BUF1Addr (0x220)
为了防止USB SRAM在同时支持Microphone和Speaker的情况下不够用,这里将BUF0和BUF1使用同一块USB SRAM。
/* EP2 */
/* buffer base address */
#define ENDP2_BUF0Addr (0xA0)
#define ENDP2_BUF1Addr (0x160)

/* EP3 */
/* feedback buffer base address */
#define ENDP3_BUF0Addr (0x2E0)
#define ENDP3_BUF1Addr (0x2E0)

/*USB ISO Max Length*/
#define USB_EDNP_MAX_PACKET 0x120
```



### 2.6.3 Speaker 和 MicroPhone 设备配置

主要实现对设备的描述，主机可以通过设备描述了解设备的具体功能。对应Speaker和MicroPhone的功能，用户可以选择具体需要那个功能或者两个功能都需要，设备的描述会根据当前配置进行修改。

#### 功能配置

可以通过at32\_audio\_conf.h中配置当前需要实现的功能。配置包括是否实现Speaker和MicroPhone，是否支持16KHz和48KHz采样率。配置代码如下：

```
/*Enable Speaker or Micrphone, 1 is Enable, 0 is Disable*/ 配置是否支持Speaker或者Micrphone
#define ENABLE_AT32_USB_SPEAKER 1
#define ENABLE_AT32_USB_MICRPHONE 1

/*Support 16K freq or 48K Freq, 1 is Enable, 0 is Disable*/配置是否支持16K或者48K采用率
#define ENABLE_AT32_USB_AUDIO_FREQ_16K 1
#define ENABLE_AT32_USB_AUDIO_FREQ_48K 0

/*Support Audio Freq*/
#define AT32_AUDIO_FREQ_16K 16000
#define AT32_AUDIO_FREQ_48K 48000

/*Support Channel*/
#define AT32_AUDIO_STEREO 2

/*Support bit Wide*/
#define AT32_AUDIO_BITW_16 16

/*Freq 16K*/ 如下为当前默认配置
#define AT32_DEFAULT_AUDIO_FREQ AT32_AUDIO_FREQ_16K
/*Bit Wide 16bit*/
#define AT32_DEFAULT_AUDIO_BITW AT32_AUDIO_BITW_16
/*Channel 2 channel, stereo*/
#define AT32_DEFAULT_AUDIO_CHR AT32_AUDIO_STEREO
```

#### Audio设备描述-》usb\_desc.c

字段	值	描述
bLength	0x12	Size of this descriptor in bytes
bDescriptorType	0x01	DEVICE Descriptor type
bcdUSB	0x0200	USB Specification Release Number
bDeviceClass	0x00	Device defined at interface level
bDeviceSubClass	0x00	Unused
bDeviceProtocol	0x00	Unused
bMaxPacketSize0	0x40	Maximum packet size for endpoint 0 0x40 bytes
idVendor	0x2E3C	Vendor ID
idProduct	0x5730	Product ID
bcdDevice	0x0200	Device release number
iManufacturer	0x01	Index of string descriptor describing Manufacturer
iProduct	0x02	Index of string descriptor describing product
iSerialNumber	0x03	Index of string descriptor describing the device serial number
bNumConfigurations	0x01	Number of possible configurations

#### Audio 配置描述-》usb\_desc.c

字段	值	描述
bLength	0x09	Size of this descriptor in bytes
bDescriptorType	0x02	CONFIGURATION Descriptor type
wTotalLength	0xC0 ( 注释1 )	Total length of data returned for this configuration
bNumberInterface	0x02 (注释2)	Number of interfaces supported of this configuration
bConfigurationValue	0x01	Value to use as SetConfiguration() argument
iConfiguration	0x00	Index of string descriptor describing this configuration
bmAttributes	0xc0	Configuration characteristics
bMaxPower	0x32	bMaxPower

注释1：wTotalLength根据配置不同而不同（支持的采样率，是否支持Micpone/Speaker）

注释2：bNumberInterface，根据要是实现的功能不同而不同(是否支持Micpone/Speaker)

## 标准AC接口描述-》usb\_desc.c

字段	值	描述
bLength	0x0A	Size of this descriptor in bytes
bDescriptorType	0x04	INTERFACE Descriptor type
bInterfaceNumber	0x00	Number of this interface
bAlternateSetting	0x00	Index of this setting
bNumEndpoints	0x00	0 endpoint
bInterfaceClass	0x01	AUDIO
bInterfaceSubClass	0x01	AUDIO CONTROL
bInterfaceProtocol	0x00	Protocol code(assigned by the USB)
iInterface	0x00	Index of string descriptor describing this interface

## Audio Class-specific AC接口描述-》usb\_desc.c

字段	值	描述
bLength	0x0A	Size of this descriptor
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x01	HEADER descriptor subtype
bcdADC	0x0100	Audio Device Class Specificaion Release Number
wTotalLength	0x46	Total number of bytes returned for class-specific
bIncollection	0x02	The number of AudioStreaming and MIDIStreaming interfaces
baInterfaceNr	0x02	Interface number of the first AudioStreaming
baInterfaceNr	0x01	Interface number of the second AudioStreaming

## MicroPhone Input Terminal描述-》usb\_desc.c

字段	值	描述
bLength	0x0C	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x02	INPUT_TERMINAL descriptor subtype
bTerminalID	0x01	ID of this Input Terminal
wTerminalType	0x0201	Terminal is Microphone
bAssocTerminal	0x00	No association
bNrChannels	0x02	Two channel
wChannelConfig	0x0003	Stereo
iChannelNames	0x00	Unused
iTerminal	0x00	Unused

## MicroPhone Feature Unit描述-》usb\_desc.c

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x06	FEATURE Unit ID descriptor subtype
bUnitID	0x02	ID of this Unit ID
bSourceID	0x01	Input Terminal ID
bControlSize	0x01	Control size
bmaControl	0x01	MUTE is Support
bmaControl	0x02	Volume is Support
iTerminal	0x00	Unused

## MicroPhone Output Terminal描述-》usb\_desc.c

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x03	OUTPUT_TERMINAL descriptor subtype
bTerminalID	0x03	ID of this Output Terminal
wTerminalType	0x0101	USB Streaming
bAssocTerminal	0x00	Unused
bSourceID	0x02	Feature Unit ID
iTerminal	0x00	Unused

## Speaker Input Terminal描述-》usb\_desc.c

字段	值	描述
bLength	0x0C	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x02	INPUT_TERMINAL descriptor subtype
bTerminalID	0x04	ID of this Input Terminal
wTerminalType	0x0101	USB Streaming
bAssocTerminal	0x00	No association
bNrChannels	0x02	Two channel
wChannelConfig	0x0003	Stereo
iChannelNames	0x00	Unused
iTerminal	0x00	Unused

## Speaker Feature Unit描述-》usb\_desc.c

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x06	FEATURE Unit ID descriptor subtype
bUnitID	0x05	ID of this Unit ID
bSourceID	0x04	Input Terminal ID
bControlSize	0x01	Control size
bmaControl	0x01	MUTE is Support
bmaControl	0x02	Volume is Support
iTerminal	0x00	Unuse

## Speaker Output Terminal描述-》usb\_desc.c

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type

bDescriptorSubtype	0x03	OUTPUT_TERMINAL descriptor subtype
bTerminalID	0x06	ID of this Output Terminal
wTerminalType	0x0301	Speaker
bAssocTerminal	0x00	Unused
bSourceID	0x05	Feature Unit ID
iTerminal	0x00	Unused

### MicroPhone Standard AS interface描述-》usb\_desc.c

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x04	CS_INTERFACE descriptor type
bInterfaceNumber	0x01	Index of this interface
bAlternateSetting	0x00	Index of this alternate setting
bNumEndpoint	0x00	0 endpoint
bInterfaceClass	0x01	AUDIO
bInterfaceSubClass	0x02	AUDIO_STREAMING
bInterfaceProtocol	0x00	Unused
iInterface	0x00	Unused

### MicroPhone Standard AS interface描述-》usb\_desc.c

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x04	CS_INTERFACE descriptor type
bInterfaceNumber	0x01	Index of this interface
bAlternateSetting	0x01	Index of this alternate setting
bNumEndpoint	0x01	1 endpoint
bInterfaceClass	0x01	AUDIO
bInterfaceSubClass	0x02	AUDIO_STREAMING
bInterfaceProtocol	0x00	Unused
iInterface	0x00	Unused

### MicroPhone Class-Specific AS Interface描述-》usb\_desc.c

字段	值	描述
bLength	0x07	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x01	GENERAL subtype
bTerminalLink	0x03	Unit ID of the Output Terminal
bDelay	0x01	Interface delay
wFormatTag	0x0001	PCM Format

### MicroPhone Type I Format type描述-》usb\_desc.c

字段	值	描述
bLength	0x0E	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x02	FORMAT_TYPE subtype
bFormatType	0x01	FORMAT_TYPE_I
bNrChannels	0x02	Two Channel
bSubFrameSize	0x02	Two bytes per audio subframe
bBitResolution	16	16 bits per sample
bSamFreqType	2	Two frequency supported
tSamFreq	0x003E80	16000HZ
tSamFreq	0x00BB80	48000HZ

**MicroPhone Standard Endpoint描述-》usb\_desc.c**

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x05	ENDPOINT descriptor
bEndpointAddress	0x81	IN Endpoint 1
bmAttributes	0x05	Isochronous
wMaxPacketSize	0x0120	288 bytes per packet
bInterval	0x01	One packet per frame
bRefresh	0x00	Unused
bSynchAddress	0x00	Unused

**MicroPhone Class-specific iso Endpoint描述-》usb\_desc.c**

字段	值	描述
bLength	0x07	Size of this descriptor,in bytes.
bDescriptorType	0x25	ENDPOINT descriptor
bDescriptorSubtype	0x01	GENERAL subtype
bmAttributes	0x00	No sampling frequency control
bLockDelayUnits	0x00	Unused
wLockDelay	0x0000	Unused

**Speaker Standard AS interface描述-》usb\_desc.c**

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x04	CS_INTERFACE descriptor type
bInterfaceNumber	0x02	Index of this interface
bAlternateSetting	0x00	Index of this alternate setting
bNumEndpoint	0x00	0 endpoint
bInterfaceClass	0x01	AUDIO
bInterfaceSubClass	0x02	AUDIO_STREAMING
bInterfaceProtocol	0x00	Unused
iInterface	0x00	Unused

**Speaker Standard AS interface描述-》usb\_desc.c**

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x04	CS_INTERFACE descriptor type
bInterfaceNumber	0x02	Index of this interface
bAlternateSetting	0x01	Index of this alternate setting
bNumEndpoint	0x02	2 endpoint,feed back
bInterfaceClass	0x01	AUDIO
bInterfaceSubClass	0x02	AUDIO_STREAMING
bInterfaceProtocol	0x00	Unused
iInterface	0x00	Unused

**Speaker Class-Specific AS Interface描述-》usb\_desc.c**

字段	值	描述
bLength	0x07	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x01	GENERAL subtype
bTerminalLink	0x04	ID
bDelay	0x01	Interface delay

wFormatTag	0x0001	PCM Format
------------	--------	------------

**Speaker Type I Format type描述-》usb\_desc.c**

字段	值	描述
bLength	0x0E	Size of this descriptor,in bytes.
bDescriptorType	0x24	CS_INTERFACE descriptor type
bDescriptorSubtype	0x02	FORMAT_TYPE subtype
bFormatType	0x01	FORMAT_TYPE_I
bNrChannels	0x02	Two Channel
bSubFrameSize	0x02	Two bytes per audio subframe
bBitResolution	16	16 bits per sample
bSamFreqType	2	Two frequency supported
tSamFreq	0x003E80	16000HZ
tSamFreq	0x00BB80	48000HZ

**Speaker Standard Endpoint描述-》usb\_desc.c**

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x05	ENDPOINT descriptor
bEndpointAddress	0x82	OUT Endpoint 2
bmAttributes	0x05	Isochronous
wMaxPacketSize	0x0120	288 bytes per packet
bInterval	0x01	One packet per frame
bRefresh	0x00	Unused
bSynchAddress	0x83	Feed back endpoint 3

**Speaker Class-specific iso Endpoint描述-》usb\_desc.c**

字段	值	描述
bLength	0x07	Size of this descriptor,in bytes.
bDescriptorType	0x25	ENDPOINT descriptor
bDescriptorSubtype	0x01	GENERAL subtype
bmAttributes	0x00	No sampling frequency control
bLockDelayUnits	0x00	Unused
wLockDelay	0x0000	Unused

**Speaker Feed back Endpoint描述-》usb\_desc.c**

字段	值	描述
bLength	0x09	Size of this descriptor,in bytes.
bDescriptorType	0x05	ENDPOINT descriptor
bEndpointAddress	0x83	OUT Endpoint 2
bmAttributes	0x11	Isochronous
wMaxPacketSize	0x0003	288 bytes per packet
bInterval	0x01	One packet per frame
bRefresh	0x08	Refresh time
bSynchAddress	0x00	Unused

## 2.6.4 MicroPhone 数据处理

Mircphone数据从USB 设备端发送到HOST端 ,发送数据频率为1ms发送一帧数据 ,一帧数据的大小根据采样频率和位宽觉得。一帧数据大小=(采样频率 / 1000 ) \* (位宽 / 8 ) \* 通道数。Demo通过端点1进行IN传输。

如下是软件实现过程：

```
void EP1_IN_Callback(void)
{
    uint8_t *u8pBuffer = Audio_GetMicrophoneBuffer(); /*获取当前数据Microphone buffer*/
    uint32_t u32SendLen = Audio_GetMicrophoneLen(); /*获取当前帧数据长度*/
    if ( u8pBuffer == 0 ) /*没有数据，直接返回*/
        return ;
    if (GetENDPOINT(ENDP1) & EP_DTOG_RX) /*ISO 端点使用双缓冲*/
    {
        /*Write data to ENDP1_BUF0Addr buffer*/
        UserToPMABufferCopy(u8pBuffer, ENDP1_BUF0Addr, u32SendLen); /*发送数据到HOST*/
        SetEPDblBuf0Count(ENDP1, EP_DBUF_IN, u32SendLen);
    }
    else
    {
        /*Write data to ENDP1_BUF1Addr buffer*/
        UserToPMABufferCopy(u8pBuffer, ENDP1_BUF1Addr, u32SendLen); /*发送数据到HOST*/
        SetEPDblBuf1Count(ENDP1, EP_DBUF_IN, u32SendLen);
    }
    FreeUserBuffer(ENDP1, EP_DBUF_IN);
}
```

## 2.6.5 Speaker 数据处理

Speaker数据从HOST端发送到设备端，发送数据频率为1ms发送一帧数据，一帧数据的大小根据采样频率和位宽觉得。一帧数据大小=(采样频率 / 1000 ) \* (位宽 / 8 ) \* 通道数。Demo通过端点2进行OUT传输，由于存在时钟不同步的问题，Speaker加上了Feedback功能，Feedback反映一段时间内设备的真实采样率，主机根据设备反馈的真实采样率，调整发数据的频率。

如下是软件实现过程：

```
void EP2_OUT_Callback(void)
{
    uint16_t Data_Len;          /* data length*/
    if (GetENDPOINT(ENDP2) & EP_DTOG_RX) /*read from ENDP1_BUF0Addr buffer*/
    {
        Data_Len = GetEPDbIBuf0Count(ENDP2);
        PMAToUserBufferCopy(u8EndPBuffer, ENDP2_BUF0Addr, Data_Len); /*获取 Speaker数据*/
    }
    else /*read from ENDP1_BUF1Addr buffer*/
    {
        Data_Len = GetEPDbIBuf1Count(ENDP2);
        PMAToUserBufferCopy(u8EndPBuffer, ENDP2_BUF1Addr, Data_Len); /*获取 Speaker数据*/
    }
    Audio_WriteDataToSpeakerBuffer(u8EndPBuffer, Data_Len); /*将数据写入播放缓冲区*/
    FreeUserBuffer(ENDP2, EP_DBUF_OUT);
}
/*Feed back 数据*/
void EP3_IN_Callback(void)
{
    uint8_t *u8pBuffer = Audio_GetSpeakerFeedback(); /*获取当前feedback*/
    uint32_t u32SendLen = 3; /*固定为3Byte*/
    if (GetENDPOINT(ENDP3) & EP_DTOG_RX)
    /*read from ENDP1_BUF0Addr buffer*/
    {
        UserToPMABufferCopy(u8pBuffer, ENDP3_BUF0Addr, u32SendLen);
        SetEPDbIBuf0Count(ENDP3, EP_DBUF_IN, u32SendLen);
    }
    else
    {
        UserToPMABufferCopy(u8pBuffer, ENDP3_BUF1Addr, u32SendLen);
        SetEPDbIBuf1Count(ENDP3, EP_DBUF_IN, u32SendLen);
    }
    FreeUserBuffer(ENDP3, EP_DBUF_IN);
}
```

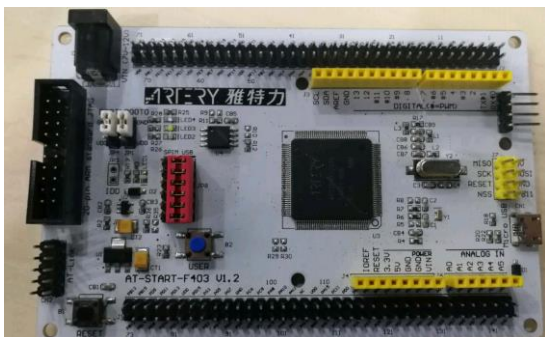


### 3 USB Audio测试使用说明

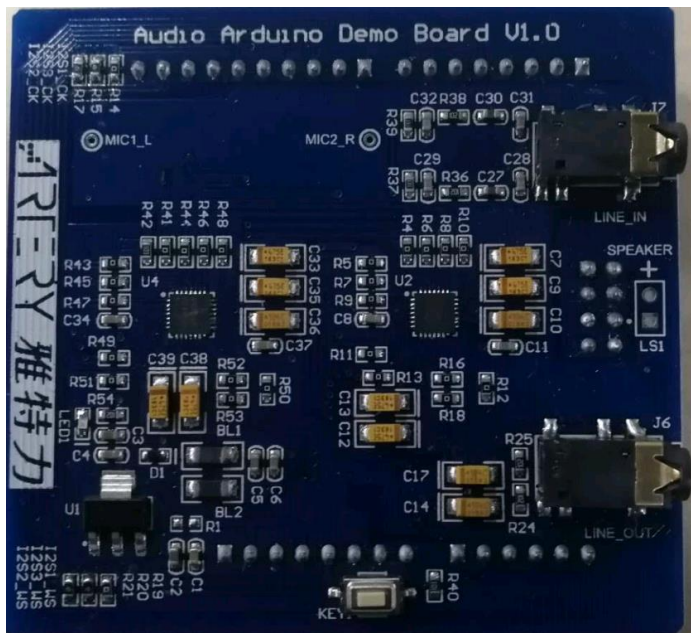
本节介绍如何使用USB Audio Demo进行播放声音和进行录音。

#### 3.1 硬件资源

- 指示灯 LED2/LED3/LED4
- USB(PA11/PA12)
- AT-START-F403 V1.2 实验板



- Audio Arduino Demo Board V1.0



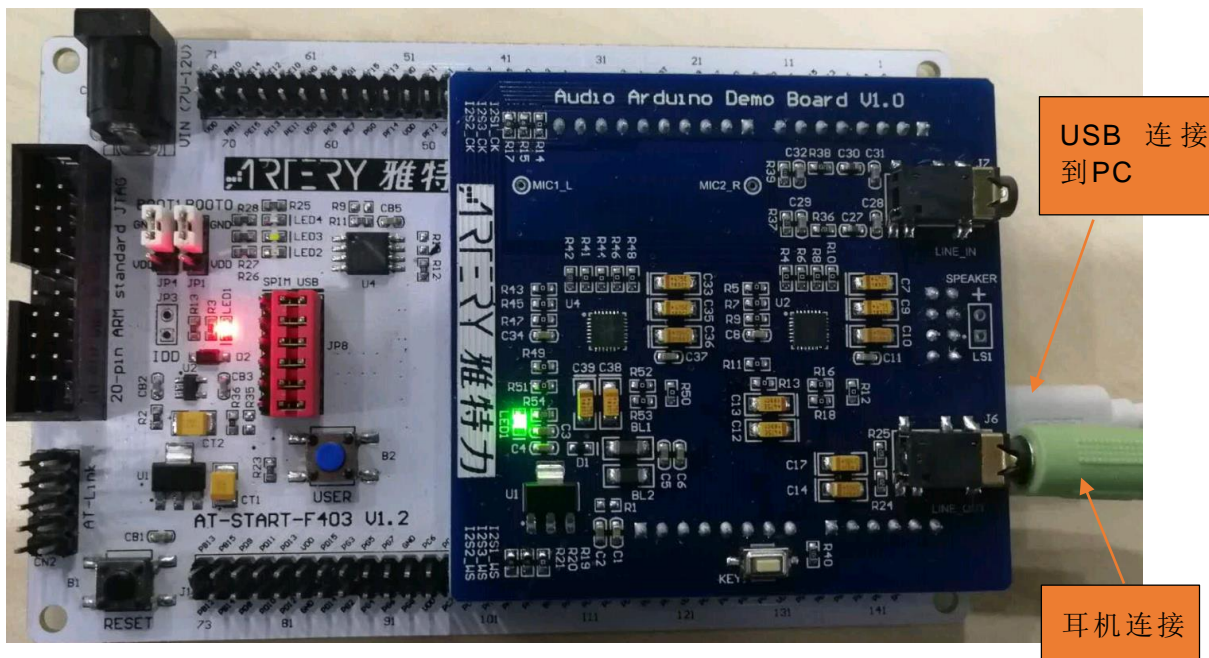
- 耳机

#### 3.2 使用方法步骤

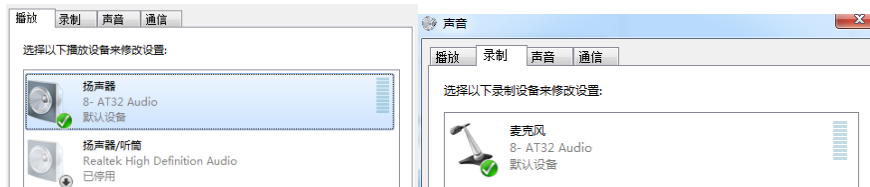
- 连接AT-START-F403 V1.2 和Audio Arduino Demo Board

- 下载Audio Demo到实验板（工程路径：  
SourceCode\USB\_Audio\_Demo\Project\Audio\_Composite\_Speaker\_MicroPhone
- 连接USB到PC Host（目前测试在WIN7系统下）

开发板连接示意图

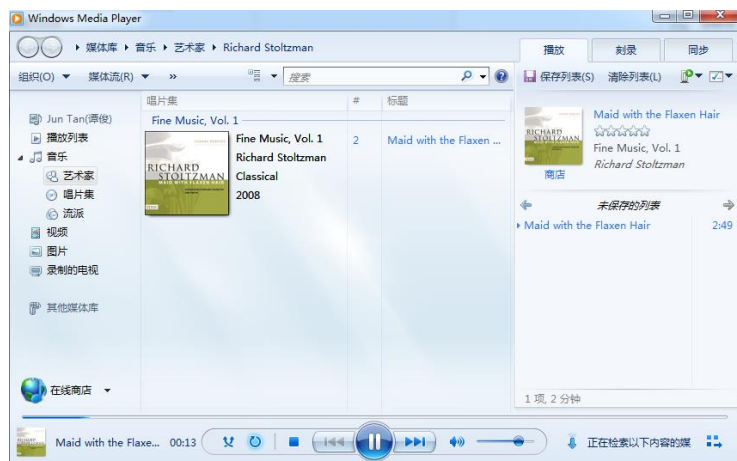


- 如果PC成功识别到Audio设备，在声音控制器里面会看到AT32 Audio的设备，分别是一个扬声器和一个麦克风的设备。



## 3.3 播放声音

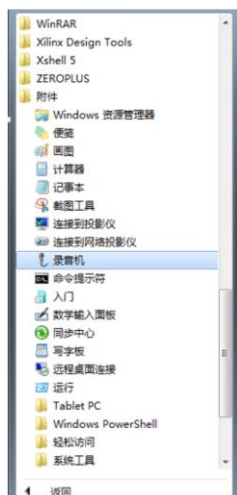
打开Windows Media Player（或者其它播放软件），随意播放一首音乐，可通过耳机听到播放的声音。



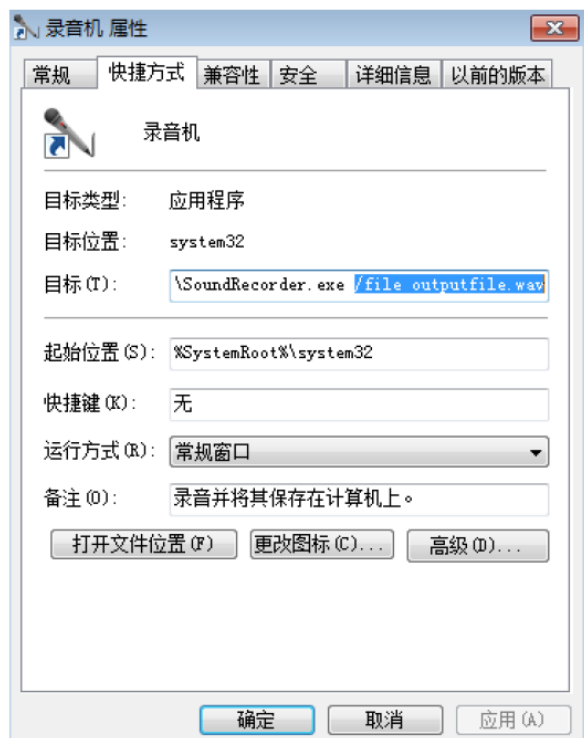
### 3.4 录音测试

通过Windows自带“录音机”进行测试，测试步骤：

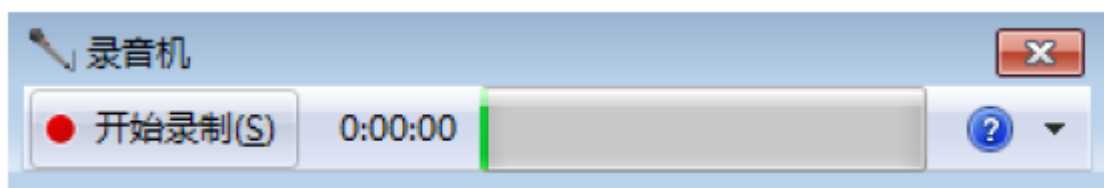
- 找到Window->“附件”->“录音机”，右键->属性



- 在目标选项里加入“ /file outputfile.wav” ，点击应用确定



- 打开录音机->开始录音



## 4 版本历史

日期	版本	变更
2019.01.09	1.0.0	最初版本

**重要通知 - 请仔细阅读**

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力的产品不得应用于武器。此外，雅特力产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且 / 或（D）航天应用或航天环境。如果雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，采购商仍将独自承担因此而导致的任何风险，雅特力的产品规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML或JAN正式认证产品适用于航天应用。

经销的雅特力产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2019 雅特力科技（重庆）有限公司 保留所有权利