



## 1、OTG 模块

### 1.1 简介

O18 芯片集成了 USB 2.0 全速控制器，支持 HOST 和 DEVICE 功能。

HOST 模式下可以实现读写 U 盘，连接蓝牙 dongle 等功能。

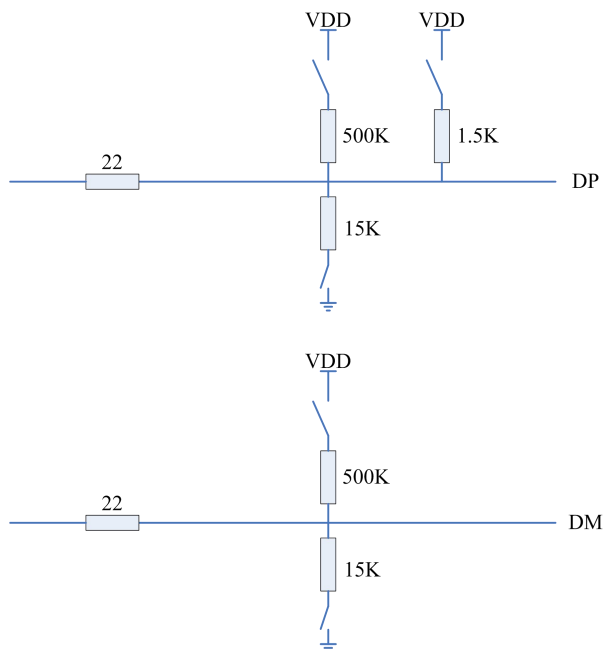
DEVICE 模式下可以实现连接电脑做读卡器、USB 声卡、反向控制等功能。

客户也可以通过调用本模块提供的基本函数实现其他协议的功能。



## 1.2 硬件资源

### 1.2.1 硬件连接



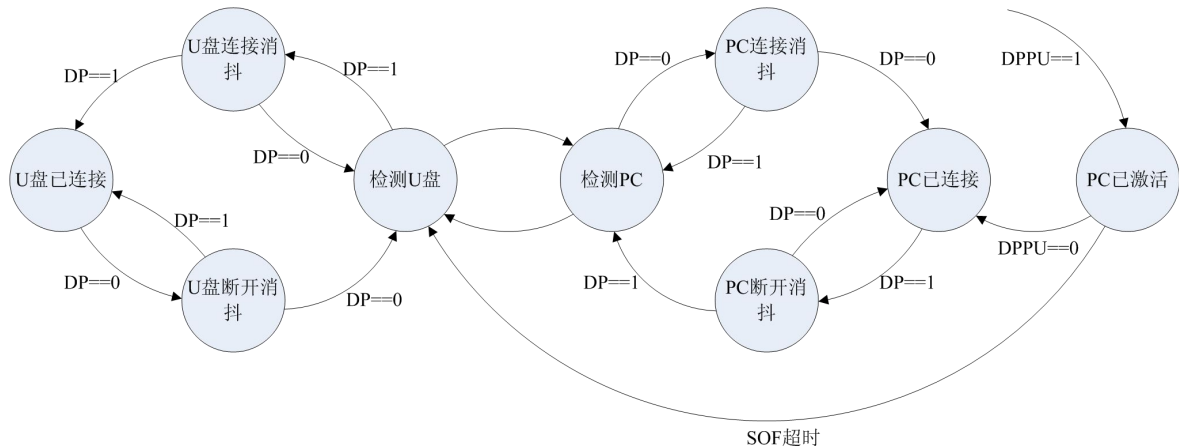
### 1.2.2 端点资源

端点号	DIR	FIFO	HOST 用途	DEVICE 用途	
0	T/R	64	CONTROL	CONTROL	
1	T/R	64	INT IN	INT IN	
2	T/R	64	BULK IN	BULK IN	
3	T/R	128	BULK OUT	BULK OUT	
4	T/R	256	ISO OUT	ISO IN	
5	R	1024	ISO IN	ISO OUT	



## 1.3 连接检测

### 1.3.1 连接检测状态机



2 个 USB 端口分别有一个检测状态机在运转，互相独立。

### 1.2.1 接口函数

`VOID OTGLinkCheck(VOID);`

//在定时器中断处理函数中每毫秒调用一次

`VOID UsbSetCurrentPort(BYTE PortNum);`

//设置当前有效的 USB 端口

//有 2 组 GPIO 可以复用为 USB 端口，调用本函数设置当前有效 USB 端口后，所有的 USB 操作都会针对当前这个端口。

`BOOL UsbHost1IsLink(VOID);`

//检查 USB 端口 1 是否连接了一个 USB 设备

`BOOL UsbHost2IsLink(VOID);`

//检查 USB 端口 2 是否连接了一个 USB 设备

`BOOL UsbHostIsLink(VOID);`

//检查当前选中的 USB 端口是否连接了一个 USB 设备

`BOOL UsbDevice1IsLink();`

//检查 USB 端口 1 是否连接了一个 USB 主机

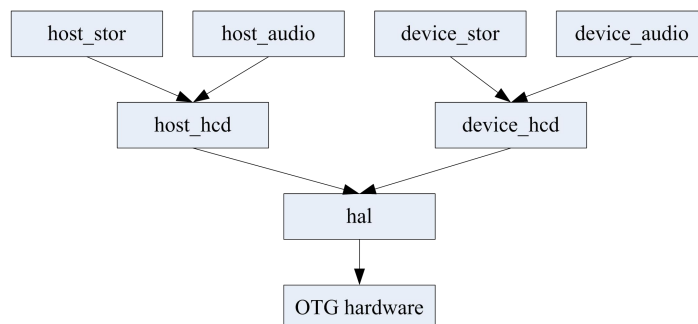


```
    BOOL UsbDevice2IsLink();  
//检查 USB 端口 2 是否连接了一个 USB 主机
```

```
BOOL UsbDeviceIsLink();  
//检查当前选中的 USB 端口是否连接了一个 USB 主机
```



## 1.4 软件框架



host\_hcd 提供 HOST 模式下基本的收发数据包接口函数。

device\_hcd 提供 DEVICE 模式下基本的收发数据包接口函数。

host\_stor 提供 HOST 模式下读写 U 盘的的接口函数。

host\_audio 提供 HOST 模式下与 ipod 的通信协议。

device\_stor 提供读卡器功能接口。

device\_audio 提供 USB 声卡功能接口。



## 1.5 HOST 模式接口函数

### 1.5.1 host\_hcd 模块接口

本模块提供 OTG 控制器工作在 HOST 模式时，一组基本函数接口。

上层通过调用这些基本函数可以实现某种具体的协议功能，比如读写 U 盘，iPhone Docking 等功能。

```
BOOL HostUsbOpen(VOID);
```

```
BOOL HostIsLink(VOID);
```

```
VOID HostUsbPortReset(VOID);
```

```
VOID HostUsbSetAddress(BYTE Address); // Set device address
```

```
BOOL HostUsbSetupPacket(BYTE* SetupPacket, BYTE Direction, BYTE* Buf, WORD Len);
```

```
BOOL HostRcvPacket(PIPE_INFO* Pipe, BYTE* Buf, WORD Len, WORD TimeOUT);
```

```
BOOL HostSendPacket(PIPE_INFO* Pipe, BYTE* Buf, WORD Len, WORD TimeOUT);
```

### 1.5.2 host\_stor 模块接口

本模块提供读写 U 盘的接口函数。

```
//init usb host module
```

```
VOID HostInit(VOID);
```

```
// get usb host link status
```

```
BOOL HostIsLink(VOID);
```

```
//usb host device status
```

```
//VOID HostPoll(BYTE isAutoProcessDevice);
```

```
BOOL HostStorInit(VOID);
```

```
//usb host read storage block
```

```
BOOL HostReadBlock(DWORD lba, VOID* buf, BYTE size);
```

```
//write blocks
```

```
BOOL HostWriteBlock(DWORD lba, VOID* buf, BYTE size);
```

```
//get storage device block size
```

```
WORD HostGetBlockSize(VOID);
```

```
//get storage device last lba number
```

```
DWORD HostGetLastLBA(VOID);
```



## 1.6 DEVICE 模式接口函数

### 1.6.1 device\_hcd 模块接口

本模块提供 OTG 控制器工作在 DEVICE 模式时，一组基本函数接口。  
上层通过调用这些基本函数可以实现某种具体的协议功能，比如读卡器、USB 声卡等功能。

```
BOOL UsbDeviceIsLink(VOID);

// Device initial
VOID UsbDeviceInit(VOID);

// Device software disconnect.
VOID UsbDeviceDisConnect(VOID);

// Send stall
VOID UsbDeviceSendStall(BYTE EndpointNum);

// Set receive endpoint to ISO mode.
VOID UsbDeviceSetRcvISO(VOID);

// Reset endpoint
VOID UsbDeviceResetEndpoint(BYTE EndpointNum);

// Set device address
VOID UsbDeviceSetAddress(BYTE Address);

// Get bus event
BYTE UsbDeviceGetBusEvent(VOID);

// Send data to control endpoint
VOID UsbDeviceControlSend(BYTE* Addr, BYTE Len);

// Receive data from control endpoint
BYTE UsbDeviceControlRcv(BYTE* Buf, BYTE MaxLen);

// Receive setup packet from control endpoint
BYTE UsbDeviceSetupRcv(BYTE* Buf, BYTE MaxLen);

// Send data to bulk in endpoint
VOID UsbDeviceBulkSend(BYTE* Addr, WORD Len);

// Receive data from bulk out endpoint
WORD UsbDeviceBulkRcv(BYTE* Buf, WORD Len);
```



## 1.6.2 device\_audio 模块接口

```
//pc volume: 0 --- 999
#define AUDIO_MAX_VOLUME 999

//pc command
#define PC_CMD_NEXT          0x04
#define PC_CMD_PRE           0x08
#define PC_CMD_MUTE          0x10
#define PC_CMD_PLAY_PAUSE    0x40
#define PC_CMD_STOP          0x80

//usb audio device initial
VOID DeviceAudioInit(VOID);

//usb audio device process
VOID DeviceRequestProcess(VOID);

//send command to pc
BOOL DeviceAudioSendPcCmd(BYTE PcCmd);
```

## 1.6.3 device\_stor 模块接口

```
//mass-storage device initial
VOID DeviceStorInit(VOID);

//mass-storage device process
VOID DeviceStorProcess(VOID);

BOOL DeviceStorIsPrevent(VOID);

BOOL DeviceStorIsStopped(VOID);
```