

# H1 hcrtos winusb(wcid)驱动使用说明

## H2 概述

本驱动支持winusb(wcid)驱动, PC Windows就可以通过免驱的方式, 利用usb线跟开发板进行数据传输.

另外支持用户修改winusb驱动中的vendor参数, 例如 `PID` / `VID` / `product_name` / `wcid vendor code` 等参数. 如果不做这些修改, 缺省的参数如下

winusb 缺省参数

- `PID` : 0xABCD
- `VID` : 0x0002
- `bcdDevice` : 0x0001
- `winusb vendor code` : 0x17 ## 即回应字符描述符index 0xEE时候的参数
- `manufacturer` : Hichip Inc
- `product` : hichip winusb
- 响应 `inusb vendor cod` 并且index为5时的
  - `prop_name` : DeviceInterfaceGUID
  - `prop_data` : {1D4B2365-4749-48EA-B38A-7C6FDDDD7E26}

## H2 使能winusb 驱动

打开menuconfig, 配置如下

- 使能winusb驱动 // 必须项目
  - `BR2_PACKAGE_PREBUILTS_USBDriver`
  - `BR2_PACKAGE_PREBUILTS_USBGADGETDRIVER`
  - `BR2_PACKAGE_PREBUILTS_USBGADGETDRIVER_WINUSB`
- 使能winusb vendor配置 // 非必须项目, 如果需要自定义PID/VID/product\_name这些参数才必要打开
  - `CONFIG_USB_VENDOR`

- CONFIG\_USB\_GADGET\_VENDOR
- CONFIG\_USB\_GADGET\_WINUSB\_VENDOR
- 测试winusb 驱动的demo示例 // 非必须项目, 仅是展示使用的demo示例
  - CONFIG\_CMDS\_USB
  - CONFIG\_CMDS\_USB\_GADGET\_WINUSB

使能以上menuconfig配置后, 编译命令如下

```
1 make kernel-rebuild cmds-rebuild all
```

## H2 winusb驱动使用说明

### H3 源码API

```
1 // from kernel/drivers/hcusb.h
2
3 /* 初始化usb端口 */
4 /*
5 ** brief: setup USB mode as HOST or GADGET
6 ** parm:
7 **     usb_port -- 0:usb0, 1:usb1
8 **     mode -- usb mode (MUSB_HOST, MUSB_PERIPHERAL, MUSB_OTG)
9 ** return:
10 **     0 -- successfully
11 **     !0 -- failure
12 */
13 int hcusb_set_mode(uint8_t usb_port, enum musb_mode mode);
14
15 /* 初始化winusb 驱动 */
16 int hcusb_gadget_winusb_init(void);
17 void hcusb_gadget_winusb_deinit(void);
18 int hcusb_gadget_winusb_specified_init(const char *udc_name);
```

### H3 使用简单示例

```
1 #include <kernel/drivers/hcusb.h>
2
3 void module_init_winusb(void)
```

```

4  {
5      //初始化usb端口为device 模式
6      hcusb_set_mode(USB_PORT_0, MUSB_PERIPHERAL);
7      hcusb_gadget_winusb_specified_init(get_udc_name(USB_PORT_0));
8  }
9
10 void module_deinit_winusb(void)
11 {
12     //反初始化winusb驱动, 相当于注销winusb驱动
13     hcusb_gadget_winusb_deinit();
14 }

```

### H3 通过/dev/winusb 收发数据

当winusb驱动成功初始化之后, 就会创建一个新的设备文件 `/dev/winusb`, 应用层就可以通过这个设备文件进行 `open` / `write` / `read` / `poll` / `close` 的操作

#### demo源码示例

具体请见 `components\cmds\source\usb\usb_gadget_cmd_winusb.c`,

以下示例就是将所收到的数据, 再重新发回去给PC, 相当于起到一个loopback的效果

```

1  int setup_usbd_winusb_demo(int argc, char **argv)
2  {
3      #define WINUSB_BUF_LEN (1024 * 4)
4      int fd, rd, wr, index;
5      struct pollfd pfd;
6      unsigned char buf[WINUSB_BUF_LEN];
7
8      fd = open("/dev/winusb", O_RDWR);
9      if(fd < 0) {
10         printf("Cannot open /dev/winusb\n");
11         return -1;
12     }
13     pfd.fd = fd;
14     pfd.events = POLLIN | POLLRDNORM;
15
16     while(1) {
17         memset(buf, 0, WINUSB_BUF_LEN);
18         if (poll(&pfd, 1, -1) ≤ 0) {
19             printf("⇒ poll abort ... \n");
20             break;
21         }
22     }

```

```

23         rd = read(fd, buf, WINUSB_BUF_LEN);
24         if(rd < 0) {
25             printf(" ==> read error ..\n");
26             break;
27         }
28
29         printf("read len: %d\n", rd);
30         for(index = 0; index < rd; index++) {
31             if(index % 16 == 0)
32                 printf("%.4xh: ", index);
33
34                 printf("%2.2x ", buf[index]);
35
36                 if(index % 16 == 15)
37                     printf("\n");
38         }
39         printf("\n");
40
41         wr = write(fd, buf, rd);
42         if(wr < 0) {
43             printf(" ==> write error ..\n");
44             break;
45         }
46         printf("write len: %d\n", wr);
47     }
48
49     close(fd);
50     printf("exit ... \n");
51     return 0;
52 }

```

## H2 Demo使用说明

### H3 menuconfig配置

测试winusb 驱动的demo示例 // 非必须项目, 仅是展示使用的demo示例

- CONFIG\_CMDS\_USB
- CONFIG\_CMDS\_USB\_GADGET\_WINUSB

配置完成后, 执行以下编译命令

```
1 make kernel-rebuild cmds-rebuild all
```

### H3 对应源码

- components/cmds/source/usb/usb\_gadget\_cmd\_winusb.c

### H3 命令行使用示例

```
1 ### -p 0 表示选择 usb#0 作为winusb端口,
2 ### -p 1 表示选择 usb#1 作为winusb端口,
3 $ usb g_winusb -p 0
4
5 ### 这个demo是展示如何使用本winusb驱动, 它会将从PC windows上收过来的数据再发回去
   给PC
6 $ usb g_winusb_demo
```

## H2 用户进行vendor自定义配置

本驱动也开放对应的源码, 让用户可以自定义对应的winusb vendor配置

### H3 menuconfig配置

使能winusb vendor配置 // 非必须项目, 如果需要自定义PID/VID/product\_name这些参数才必要打开

- CONFIG\_USB\_VENDOR
- CONFIG\_USB\_GADGET\_VENDOR
- CONFIG\_USB\_GADGET\_WINUSB\_VENDOR

配置完成后, 执行以下编译命令

```
1 make kernel-rebuild cmds-rebuild all
```

## H3 源码修改

### 源码位置

```
components\kernel\source\drivers\usb\gadget\legacy\winusb.c
```

### 源码修改说明

具体修改请见源码内部的注释说明

主要支持修改的内容如下

- PID
- VID
- bcdDevice
- winusb vendor code ## 即回应字符描述符index 0xEE时候的参数
- manufacturer
- product
- 响应 inusb vendor cod 并且index为5时的control传输
  - prop\_name
  - prop\_data