

H1 hcRTOS USB gadget mass-storage 驱动说明

H2 menuconfig配置

使能此USB功能

- BR2_PACKAGE_PREBUILTS_USBDRIVER
- BR2_PACKAGE_PREBUILTS_USBGADGETDRIVER

若使用终端命令来使用此功能

- CONFIG_CMDS_USB
- CONFIG_CMDS_USB_GADGET_MSC
- CONFIG_DVR_RAMDISK ## 使用RAMDISK 功能

H2 mass-storage

支持使用 RAMDISK , SD/MMC, U盘 来作为 mass-storage 的卷轴, 以下展示如何使用命令或者源码API

H3 上电后使用ramdisk来作为mass-storage

```

1  nsh
2  mkrd -m 0 -s 512 20480    #创建10MB的ramdisk
3  mkfatfs /dev/ram0
4  exit
5  usb
6  g_mass_storage -p 0 /dev/ram0 # 指定使用usb#0
7  g_mass_storage -p 1 /dev/ram0 # 指定使用usb#1
8
9  其他选项:
10 g_mass_storage -h          # 查看g_mass_storage 命令帮助说明
11 g_mass_storage -s -p 0 # 关闭usb#0 的mass-storage功能
12 g_mass_storage -s -p 1 # 关闭usb#1 的mass-storage功能

```

H3 源码API说明

使用示例可见 `components\cmds\source\usb\usb_gadget_cmd_msc.c`

```

1  // from kernel/drivers/hcusb.h
2
3  // 配置或者查看usb HOST或GADGET 模式
4  int hcusb_set_mode(uint8_t usb_port, enum musb_mode mode);
5  enum musb_mode hcusb_get_mode(uint8_t usb_port);
6
7  // 使能配置 mass-storage 功能, 默认使用的是 USB#0
8  int hcusb_gadget_msg_init(char **path, uint8_t luns);
9
10 // 指定具体的usb端口, 设置其为 mass-storage 功能
11 int hcusb_gadget_msg_specified_init(const char *udc_name, char **path,
    int luns);
12
13 // 关闭 mass-storage 功能
14 int hcusb_gadget_msg_deinit(void);

```

以下为简单的使用例程, 使用ramdisk 和 sd卡的空间来作为卷轴

```

1  #include <kernel/drivers/hcusb.h>
2
3  int setup_usbd_mass_storage(int argc, char **argv)
4  {
5      int usb_port = 1;    // 使用 usb#1
6      char *path[] = {

```

```
7         "/dev/ram0",
8         "/dev/mmcblk0"
9     };
10
11     hcusb_set_mode(usb_port, MUSB_PERIPHERAL);
12     hcusb_gadget_msg_specified_init(
13         get_udc_name(usb_port),
14         path, 2);
15 }
16
17
18 int deinit_usbd_mass_storage(int argc, char **argv)
19 {
20     int usb_port = 1; //使用 usb#1
21     hcusb_set_mode(usb_port, MUSB_HOST);
22     hcusb_gadget_msg_deinit();
23 }
```