

杂项设备（misc device）

原创

[monkea123](#) 于 2020-10-17 20:03:29 发布 463 收藏 1

分类专栏: [嵌入式 linux](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/monkea123/article/details/109137398>

版权



[嵌入式](#) 同时被 2 个专栏收录

50 篇文章 3 订阅

订阅专栏



[linux](#)

40 篇文章 1 订阅

订阅专栏

杂项设备（misc device）

杂项设备也是嵌入式系统中用得比较多的一种设备驱动。在 Linux 内核的include/linux目录下有miscdevice.h文件, 要把自己定义的misc device从设备定义在这里。其实是因为这些字符设备不符合预先确定的字符设备范畴, 所有这些设备采用主编号10, 一起归于misc device, 其实misc_register就是用主标号10调用register_chrdev()的, 只不过misc是将一些字符设备存放在misc类中。换句话说, misc设备其实也就是特殊的字符设备。

定义一个 MISC 设备(miscdevice 类型)以后我们需要设置 minor、name 和 fops 这三个成员变量。minor 表示子设备号, MISC 设备的主设备号为 10, 这个是固定的, 需要用户指定子设备号, Linux 系统已经预定义了一些 MISC 设备的子设备号

```

#define PSMOUSE_MINOR 1
#define MS_BUSMOUSE_MINOR 2 /* unused */
#define ATIXL_BUSMOUSE_MINOR 3 /* unused */
/*#define AMIGAMOUSE_MINOR 4 FIXME OBSOLETE */
#define ATARIMOUSE_MINOR 5 /* unused */
#define SUN_MOUSE_MINOR 6 /* unused */
#define APOLLO_MOUSE_MINOR 7 /* unused */
#define PC110PAD_MINOR 9 /* unused */
/*#define ADB_MOUSE_MINOR 10 FIXME OBSOLETE */
#define WATCHDOG_MINOR 130 /* Watchdog timer */
#define TEMP_MINOR 131 /* Temperature Sensor */
#define RTC_MINOR 135
#define EFI_RTC_MINOR 136 /* EFI Time services */
#define VHCI_MINOR 137
#define SUN_OPENPROM_MINOR 139
#define DMAPI_MINOR 140 /* unused */
#define NVRAM_MINOR 144
#define SGI_MMTIMER 153
#define STORE_QUEUE_MINOR 155 /* unused */
#define I2O_MINOR 166
#define MICROCODE_MINOR 184
#define VFIO_MINOR 196
#define TUN_MINOR 200
#define CUSE_MINOR 203
#define MWAVE_MINOR 219 /* ACP/Mwave Modem */
#define MPT_MINOR 220
#define MPT2SAS_MINOR 221
#define MPT3SAS_MINOR 222
#define UINPUT_MINOR 223
#define MISC_MCELOG_MINOR 227
#define HPET_MINOR 228
#define FUSE_MINOR 229
#define KVM_MINOR 232
#define BTRFS_MINOR 234
#define AUTOFS_MINOR 235
#define MAPPER_CTRL_MINOR 236
#define LOOP_CTRL_MINOR 237
#define VHOST_NET_MINOR 238
#define UHID_MINOR 239
#define MISC_DYNAMIC_MINOR 255

```

name 就是此 MISC 设备名字，当此设备注册成功以后就会在/dev 目录下生成一个名为 name的设备文件。fops 就是字符设备的操作集合，MISC 设备驱动最终是需要使用用户提供的 fops操作集合。

关于device_create, class_create 作用：class_create函数在misc.c中的模块初始化中被调用，现在一起说一下。这两个函数看起来很陌生，没有在ldd3中发现过，看源代码的时候发现class_create会调用底层组件

(2) 函数接口：

int misc_register(struct miscdevice * misc) 完成misc设备注册。

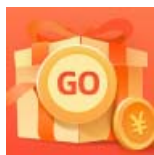
int misc_deregister(struct miscdevice *misc) 完成misc设备注销。

说明

(1) misc设备完成的是用户层到底层文件操作函数的映射，底层操作函数和具体设备的映射需要驱动自己完成，可以使用设备地址或者结合其他总线接口，例如i2c的ifly设备，可以在底层函数中，直接调用i2c相关接口函数实现。

(2) misc设备被实现为一个子系统，在子系统初始化阶段调用misc_init，设置了主设备号以及misc设备的文件操作。此时的misc设备文件操作和具体设备文件操作不同，这里的open函数需要根据不同的此设备号选择相应具体设备的底层文件操作。

(3) 在misc设备注册时，系统自动创建设备文件/dev/misc_dev，这需要Linux设备模型以及udev的支持。从linux内核2.6的某个版本之后，devfs不复存在，udev成为devfs的替代。相比devfs，udev有很多优势。



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)