**04-驱动模块**

1. **驱动环境**

安装匹配头文件

|  |
| --- |
| Bash sudo apt-get install linux-headers-$(uname -r) |

2. **驱动编程**

2.1 **驱动加载**

两种方式被编译和加载：

1、直接编译进 Linux 内核，随同 Linux 启动时加载；

2、编译成一个可加载和删除的模块，使用 insmod 加载（ modprobe 和 insmod 命令类似，但依赖于

相关的配置文件）， rmmod 删除。

|  |
| --- |
| Bash  #include <linux/module.h> //所有模块都需要的头文件 #include <linux/init.h> // init&exit 相关宏 MODULE\_LICENSE("GPL"); static int \_\_init hello\_init (void) {  printk("Hello module init\n");  return 0; } static void \_\_exit hello\_exit (void) {  printk("Hello module exit\n"); } module\_init(hello\_init); module\_exit(hello\_exit); |

加载

|  |
| --- |
| Bash insmod ./hello.o |

卸载

|  |
| --- |
| Bash rmmod hello |

2.2 **驱动API**

1.设备注册和注销：

register\_chrdev()：注册字符设备。

unregister\_chrdev()：注销字符设备。

2.设备访问和控制：

file\_operations结构体：包含设备操作函数，如open()、read()、write()、ioctl()等。

request\_region()：请求设备IO内存区域。

release\_region()：释放设备IO内存区域。

request\_irq()：请求中断。

free\_irq()：释放中断。

3.内存管理：

kmalloc()：分配内核空间内存。

kfree()：释放内核空间内存。

4.设备驱动注册和注销：

platform\_driver\_register()：注册平台设备驱动程序。

platform\_driver\_unregister()：注销平台设备驱动程序。

pci\_register\_driver()：注册PCI设备驱动程序。

pci\_unregister\_driver()：注销PCI设备驱动程序。

5.设备节点操作：

device\_create()：创建设备节点。

device\_destroy()：销毁设备节点。

3. **uboot**

4. **USB驱动**

4.1 **查看事件**

|  |
| --- |
| C++ ls /dev/input |

4.2 **查看具体设备详情**

|  |
| --- |
| C++ sudo cat /porc/bus/input/devices    sudo apt-get install input\* sudo input-events x(event序列号) |

4.3 **lsusb 简单查看**

|  |
| --- |
| C++ /sys/bus/pci/devices/  /proc/bus/pci/devices/ |

是 Linux 文件系统中与 PCI 设备相关的路径。它们提供有关系统中 PCI 设备的信息。

/sys/bus/pci/devices/:

* 此目录包含有关系统中 PCI 设备的信息。
* 每个子目录对应于特定的 PCI 设备，您可以在其中找到各种文件，其中包含有关设备的信息，例如供应商和设备 ID、配置寄存器等。
* 这些信息由内核动态生成，可用于各种系统管理任务。

/proc/bus/pci/devices/:

* 在较早版本的 Linux 内核中，PCI 设备信息可在 /proc/bus/pci/devices/ 下找到。
* 但是，/proc/bus/pci/ 目录已被弃用，现代系统通常使用 /sys/bus/pci/ 目录来获取 PCI 设备信息。
* 如果在您的系统中找到 /proc/bus/pci/devices/，它很可能是指向 /sys/bus/pci/devices/ 的符号链接，或者可能仅用于兼容性原因。

总体而言，如果您在现代 Linux 系统上处理 PCI 设备信息，应使用 /sys/bus/pci/devices/ 目录。/proc/bus/pci/ 路径是一个已弃用的遗留位置，建议使用 /sys/ 路径。

5. **树莓派can驱动**

5.1 **相关驱动**

https://www.waveshare.net/wiki/2-CH\_CAN\_HAT#.E8.AF.B4.E6.98.8E

http://www.airspayce.com/mikem/bcm2835/

<https://www.fke6.com/html/84450.html>

5.2 **硬件**

由于树莓派本身并不支持CAN总线，因此使用SPI接口的CAN控制器，搭配一个收发器完成CAN功能

5.3 **驱动安装**

必要库函数

安装BCM2835， 打开树莓派终端，并运行一下指令

|  |
| --- |
| Bash wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.60.tar.gz tar zxvf bcm2835-1.60.tar.gz  cd bcm2835-1.60/ sudo ./configure sudo make sudo make check sudo make install |

安装wiringPi

|  |
| --- |
| Bash #打开树莓派终端，并运行以下指令 cd sudo apt-get install wiringpi #对于树莓派2019年5月之后的系统（早于之前的可不用执行），可能需要进行升级： wget https://project-downloads.drogon.net/wiringpi-latest.deb sudo dpkg -i wiringpi-latest.deb gpio -v #运行gpio -v会出现2.52版本，如果没有出现说明安装出错 #Bullseye分支系统使用如下命令： git clone https://github.com/WiringPi/WiringPi cd WiringPi ./build sudo gpio -v #运行gpio -v会出现2.70版本，如果没有出现说明安装出错 |

安装Python函数库

|  |
| --- |
| Bash #python2 sudo apt-get update sudo apt-get install python-pip sudo apt-get install python-pil sudo apt-get install python-numpy sudo pip install RPi.GPIO sudo pip install spidev sudo pip install python-can #python3 sudo apt-get update sudo apt-get install python3-pip sudo apt-get install python3-pil sudo apt-get install python3-numpy sudo pip3 install RPi.GPIO sudo pip3 install spidev  sudo pip3 install python-can |

5.4 **修改脚本config.txt**

树莓派：/boot/config.txt

ubuntu：/boot/firmware/config.txt

在最后一行加入如下：

|  |
| --- |
| Bash dtparam=spi=on dtoverlay=mcp2515-can1,oscillator=16000000,interrupt=25 dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=23 dtoverlay=spi-bcm2835-overlay |

重启树莓派以应用所有设置：

|  |
| --- |
| Bash sudo reboot |

5.5 **树莓派配置**

|  |
| --- |
| Bash sudo ip link set can0 up type can bitrate 1000000 sudo ip link set can1 up type can bitrate 1000000 sudo ifconfig can0 txqueuelen 65536 sudo ifconfig can1 txqueuelen 65536 |

5.6 **收发命令**

|  |
| --- |
| Bash apt install can-utils  #收  candump can0   #发 cansend can1 000#11.22.33.44 |

5.7 **MX6\_CAN**

|  |
| --- |
| Bash 1、配置修改 i.MX6U /boot/uEev.txt   dtoverlay=/lib/firmware/imx-fire-can1-overlay.dtbo dtoverlay=/lib/firmware/imx-fire-can2-overlay.dtbo 把注释去掉     2、注册can sudo apt updade  sudo apt-get -y install can-utils   3、设置can sudo ip link set can0 type can bitrate 1000000  sudo ip link set can0 up    sudo ip link set can1 type can bitrate 1000000 sudo ip link set can1 up    4、监测can candump can0  cansend can1 123#abcdabcd |