

任务 23:看门狗

课程名称: 嵌入式 实验类型: 综合

实验项目名称: 任务 23

学生姓名: 种晓雅 专业: 计科 1202 学号: 3120102154

同组学生姓名: _____ 指导老师: 蔡铭

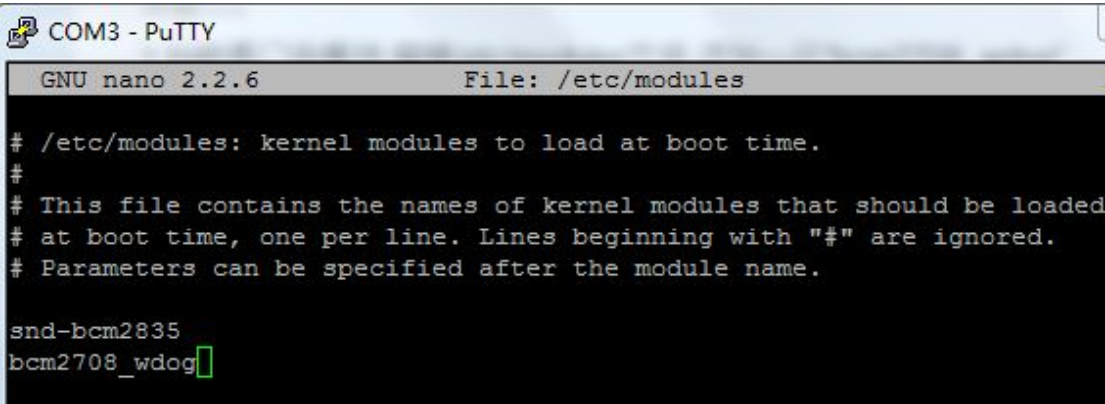
实验地点: 寝室 实验日期: 2015 年 3 月 30 日

实验内容

喂狗原理:喂狗就是在程序中打开了/dev/watchdog,写入除了 'V' 字符的任意内容。在程序中要避免重启,就要每隔一段时间往设备中写一次数据。

1. 加载看门狗模块,编辑 /etc/modules 文件,添加一行 "bcm2708 wdog"

```
pi@raspberrypi:~$ sudo modprobe bcm2708_wdog
pi@raspberrypi:~$ sudo nano /etc/modules
```



```
COM3 - PuTTY
GNU nano 2.2.6      File: /etc/modules

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
bcm2708_wdog
```

3.启动看门狗

`sudo /etc/init.d/watchdog start`

可通过 `dir /dev` 查看

```
pi@raspberrypi: ~
autofs      loop6      ram14      tty11      tty32      tty53      vcs
block       loop7      ram15      tty12      tty33      tty54      vcs1
btrfs-control loop-control ram2      tty13      tty34      tty55      vcs2
bus         MAKEDEV    ram3      tty14      tty35      tty56      vcs3
cachefiles  mapper     ram4      tty15      tty36      tty57      vcs4
char        mem        ram5      tty16      tty37      tty58      vcs5
console     memory_bandwidth ram6      tty17      tty38      tty59      vcs6
cpu_dma_latency mmcblk0    ram7      tty18      tty39      tty6       vcsa
cuse        mmcblk0p1 ram8      tty19      tty4       tty60      vcsa1
disk        mmcblk0p2 ram9      tty2       tty40      tty61      vcsa2
fb0         net        random    tty20      tty41      tty62      vcsa3
fd          network_latency raw       tty21      tty42      tty63      vcsa4
full        network_throughput root      tty22      tty43      tty7       vcsa5
fuse        null       shm       tty23      tty44      tty8       vcsa6
input       ppp        snd       tty24      tty45      tty9       vcsa
kmsg        ptmx       sndstat   tty25      tty46      ttyAMA0    vchi
log         pts        stderr    tty26      tty47      ttyprintk  watchdog
loop0       ram0       stdin     tty27      tty48      uinput     xconsole
loop1       ram1       stdout    tty28      tty49      urandom     zero
loop2       ram10      tty       tty29      tty5       vc-cma
loop3       ram11      tty0      tty3       tty50      vchiq
loop4       ram12      tty1      tty30      tty51      vcio
loop5       ram13      tty10     tty31      tty52      vc-mem
pi@raspberrypi ~ $
```

4.编写打开和自动喂狗的程序

```
pi@raspberrypi ~ $ nano feed.c
```

```
GNU nano 2.2.6      File: feed.c

#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<fcntl.h>
int main(void)
{
    int fd=open("/dev/watchdog",O_WRONLY);
    int res=0;
    if(fd==-1){
        perror("open error");
        exit(EXIT_FAILURE);
    }
    while(1){
        printf("feed dog\n");
        res=write(fd,"a",1);
        if(res!=1){
            res=-1;
            break;
        }
        sleep(5);
    }
    close(fd);
    return res;
}
```

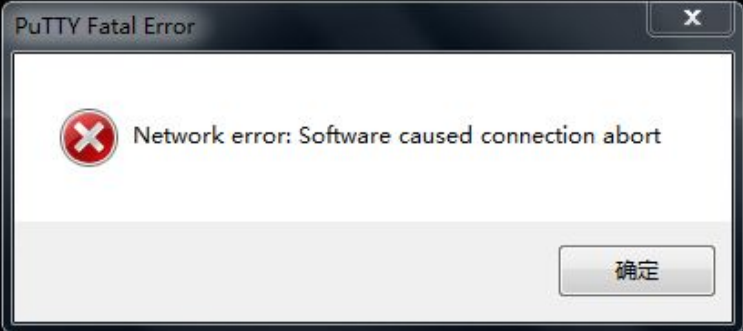
5.编译运行，每 5 秒显示 feed dog

```
pi@raspberrypi ~ $ gcc feed.c -o feed

pi@raspberrypi ~ $ sudo ./feed
feed dog
feed dog
feed dog
█
```

6.终止程序，则出现错误，需要重启

```
pi@raspberrypi ~ $ sudo ./feed
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
feed dog
^Z
[1]+  Stopped                  sudo ./feed
pi@raspberrypi ~ $ █
```

A screenshot of a PuTTY Fatal Error dialog box. The title bar reads "PuTTY Fatal Error" with a close button (X) on the right. The main area contains a red circular icon with a white 'X' and the text "Network error: Software caused connection abort". At the bottom right, there is a button labeled "确定" (OK).