

看门狗

配置内核中的硬件看门狗，使得一定时间内不喂狗就重启 Acadia 或 RPi 或 WRTnode，写一个程序或脚本保持一定频率的喂狗，当关闭这个程序或脚本时形成重启。实验报告要记录和表现出重启。

实现目的

掌握看门狗的概念；

掌握 Acadia 或 RPi 或 WRTnode 上编写看门狗程序的方法。

实验器材

硬件

Acadia 或 RPi 或 WRTnode 板一块；

5V/1A 电源一个；

microUSB 线一根；

USB-TTL 串口线一根（FT232RL 芯片或 PL2303 芯片）。

以下为自备（可选）器材：

PC（Windows/Mac OS/Linux）一台；

以太网线一根（可能还需要路由器等）。

软件

PC 上的 USB-TTL 串口线配套的驱动程序；

PC 上的串口终端软件，如 minicom、picocom、putty 等；

PC 上的 SSH 软件，如 putty 等。

实验步骤

编写看门狗程序并编译；

运行程序；

对程序进行验证。

实验过程与结果分析

1. 启动看门狗服务

```
pi@raspberrypi ~ $ sudo modprobe bcm2708_wdog
```

如图，/dev 文件下多了一个文件 watchdog

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

打开此文件后，一定时间不输入字符给这个文件，整个系统就会重启（输入 V 结束这个程序）

2. 写一段 shell 脚本喂狗

```
#!/bin/bash

for(;;)
do
sleep 5
echo "Feed" | tee /dev/watchdog
done
```

3. 运行脚本，过一段时间停止脚本，观察现象：

```
pi@raspberrypi ~ $ sudo bash feed.sh
Feed
Feed
Feed
Feed
Feed
Feed
Feed
Feed
Feed
Feed
ç^Cpi@raspberrypi ~ $
```

4. 过了一会，树莓派自动重启

```
pi@raspberrypi ~ $ Write failed: Host is down
```

感想

学习了一下看门狗的概念，看门狗在嵌入式开发中起到了很大的作用。一定程度上类似守护进程的概念，确保了嵌入式设备任务的完成。