

文件名称：PCX-9268 ubuntu GPIO\_mmu 驱动说明

文件编号：SLY-FAE-QW-055

版 本：A/00

制定日期：2022-04-18

制 定：黄飞强

审 批：梁 符

版本 修订 记录	版本	描述	修订人	日期
	A/00	第一次拟定	黄飞强	2022-4-18

## 一、驱动安装

进入9268\_GPIO 文件夹，打开 SHELL 终端，运行 insmod.sh 进行驱动的安装（需要root权限，如果第一次使用需要使用make进行驱动编译）

```

root@test-9502I09T01: /home/test/test
root@test-9502I09T01:/home/test/test# ls
gpio_drv.ko  gpio_drv_mmio  insmod.sh  read_gpio  rmmmod.sh  write_gpio
root@test-9502I09T01:/home/test/test# chmod 777 rmmmod.sh insmod.sh
root@test-9502I09T01:/home/test/test# ./insmod.sh
root@test-9502I09T01:/home/test/test#
    
```

没有出现报错，说明驱动安装成功。

如果需要每次开机自动安装驱动，需要编写开机自动运行脚本 /etc/init.d/gpio\_drv.sh 在脚本中添加以下内容（insmod [驱动位置（绝对路径）]）

```

gpio_drv.sh (/etc/init.d) - gedit
文件(F) 编辑(E) 查看(V) 搜索(S) 工具(T) 文档(D) 帮助(H)
打开(O) 保存(S)
#!/bin/bash
insmod /home/test/test/9168gpio_drv_mmio/gpio_drv.ko
    
```

```

root@test-6642I09T01:/etc/init.d
root@test-6642I09T01:/home/test/test/9168gpio_drv_mmio: cd /etc/init.d/
root@test-6642I09T01:/etc/init.d# gedit gpio_drv.sh
(gedit:3063): dconf-WARNING **: failed to commit changes to dconf: 连接已关闭
(gedit:3063): dconf-WARNING **: failed to commit changes to dconf: 连接已关闭
Error creating proxy: 连接已关闭 (g-io-error-quark, 18)
Error creating proxy: 连接已关闭 (g-io-error-quark, 18)
Error creating proxy: 连接已关闭 (g-io-error-quark, 18)
Error creating proxy: 连接已关闭 (g-io-error-quark, 18)
Error creating proxy: 连接已关闭 (g-io-error-quark, 18)
(gedit:3063): Glib-GIO-CRITICAL **: g_dbus_connection_register_object: assertion
'G_IS_DBUS_CONNECTION (connection)' failed
(gedit:3063): Glib-GIO-CRITICAL **: g_dbus_connection_register_object: assertion
'G_IS_DBUS_CONNECTION (connection)' failed
(gedit:3063): Glib-GIO-CRITICAL **: g_dbus_connection_get_unique_name: assertion
'G_IS_DBUS_CONNECTION (connection)' failed
(gedit:3063): dconf-WARNING **: failed to commit changes to dconf: 连接已关闭
(gedit:3063): dconf-WARNING **: failed to commit changes to dconf: 连接已关闭
    
```

使用 chmod +x gpio\_dr.sh 修改脚本权限（需要在root权限使用此命令）

运行脚本，查看能否进行驱动安装，

```

root@test-6642I09T01: /etc/init.d
root@test-6642I09T01:/etc/init.d# chmod +x gpio_drv.sh
root@test-6642I09T01:/etc/init.d# ./gpio_drv.sh
root@test-6642I09T01:/etc/init.d# ls /dev/gpio*
/dev/gpio0  /dev/gpio11  /dev/gpio14  /dev/gpio3  /dev/gpio6  /dev/gpio9
/dev/gpio1  /dev/gpio12  /dev/gpio15  /dev/gpio4  /dev/gpio7
/dev/gpio10 /dev/gpio13  /dev/gpio2   /dev/gpio5  /dev/gpio8
root@test-6642I09T01:/etc/init.d#
    
```

最后使用 `update-rc.d gpio_drv.sh defaults` 将 `gpio_drv.sh` 作为开机自启脚本。

```

root@test-6642I09T01: /etc/init.d
root@test-6642I09T01:/etc/init.d# update-rc.d gpio_drv.sh defaults
insserv: warning: script 'K01wdt.sh' missing LSB tags and overrides
insserv: warning: script 'gpio_drv.sh' missing LSB tags and overrides
insserv: warning: script 'wdt.sh' missing LSB tags and overrides
root@test-6642I09T01:/etc/init.d#
    
```

## 二、输出功能测试

进入 `write_gpio` 目录，运行 `make` 编译输出测试程序 `wgpio.c`

编译后生成 `wapp` 程序，使用方法：`./wapp [io设备名] on/off`

输出IO的设备名包括 `/dev/gpio0`、`/dev/gpio1`、`/dev/gpio2`、`/dev/gpio3`、`/dev/gpio4`、`/dev/gpio5`、`/dev/gpio6`、`/dev/gpio7`、`/dev/gpio8`、`/dev/gpio9`、`/dev/gpio10`、`/dev/gpio11`、`/dev/gpio12`、`/dev/gpio13`、`/dev/gpio14`、`/dev/gpio15`

例如：`./wapp /dev/gpio0 on` 功能是将 `gpio0` 拉高

`./wapp /dev/gpio4 off` 功能是将 `gpio4` 拉低

```

root@test-Default-string: /home/test/test/write_gpio
root@test-Default-string:/home/test/test/write_gpio# ls
high.sh low.sh Makefile wapp wgpio.c
root@test-Default-string:/home/test/test/write_gpio# chmod 777 high.sh low.sh
root@test-Default-string:/home/test/test/write_gpio# make clean && make
rm -rf wapp
gcc wgpio.c -o wapp
root@test-Default-string:/home/test/test/write_gpio# ls /dev/gpio
gpio0  gpio10  gpio12  gpio14  gpio2  gpio4  gpio6  gpio8
gpio1  gpio11  gpio13  gpio15  gpio3  gpio5  gpio7  gpio9
root@test-Default-string:/home/test/test/write_gpio# ./wapp /dev/gpio0 on
the val value is:1
write ok====:0
root@test-Default-string:/home/test/test/write_gpio# ./wapp /dev/gpio4 off
the val value is:0
write ok====:0
root@test-Default-string:/home/test/test/write_gpio#
    
```

其他测试脚本：

**high.sh** : 将输出IO全部拉高

**low.sh** : 将输出IO全部拉低

```
./wapp /dev/gpio0 on #对应丝印DO1
./wapp /dev/gpio1 on #对应丝印DO2
./wapp /dev/gpio2 on #对应丝印DO3
./wapp /dev/gpio3 on #对应丝印DO4
./wapp /dev/gpio4 on #对应丝印DO5
./wapp /dev/gpio5 on #对应丝印DO6
./wapp /dev/gpio6 on #对应丝印DO7
./wapp /dev/gpio7 on #对应丝印DO8
./wapp /dev/gpio8 on #对应丝印DO9
./wapp /dev/gpio9 on #对应丝印DO10
./wapp /dev/gpio10 on #对应丝印DO11
./wapp /dev/gpio11 on #对应丝印DO12
./wapp /dev/gpio12 on #对应丝印DO13
./wapp /dev/gpio13 on #对应丝印DO14
./wapp /dev/gpio14 on #对应丝印DO15
./wapp /dev/gpio15 on #对应丝印DO16
```

注意：如果 high.sh 和 low.sh 不能运行，需要先使用 `chmod 777 high.sh low.sh` 将他们设置为可执行权限。

```
root@test-Default-string: /home/test/test/write_gpio
root@test-Default-string:/home/test/test/write_gpio# chmod 777 high.sh low.sh
root@test-Default-string:/home/test/test/write_gpio# ls
high.sh low.sh Makefile wapp wgpio.c
root@test-Default-string:/home/test/test/write_gpio# ./high.sh
the val value is:1
write ok====:0
the val value is:1
write ok====:0
the val value is:1
write ok====:0
the val value is:1
write ok====:0
the val value is:1
write ok====:0
```


```
root@test-Default-string: /home/test/test/write_gpio
the val value is:1
write ok====:0
the val value is:1
write ok====:0
the val value is:1
write ok====:0
root@test-Default-string:/home/test/test/write_gpio# ./low.sh
the val value is:0
write ok====:0
the val value is:0
write ok====:0
the val value is:0
write ok====:0
the val value is:0
write ok====:0
the val value is:0
write ok====:0
the val value is:0
write ok====:0
the val value is:0
write ok====:0
the val value is:0
write ok====:0
root@test-Default-string:/home/test/test/write_gpio#
```

### 三、输入功能测试

进入 read\_gpio 目录，运行 make 编译输入测试程序 rgpio.c

编译后生成 rapp 程序，功能：自动循环读取所以输入IO的状态（默认低电平，低电平有效）

备注：所有的输入IO包括 /dev/gpio16、/dev/gpio17、/dev/gpio18、/dev/gpio19、/dev/gpio20、/dev/gpio21、/dev/gpio22、/dev/gpio23、/dev/gpio24、/dev/gpio25 /dev/gpio26、/dev/gpio27、/dev/gpio28、/dev/gpio29、/dev/gpio30、/dev/gpio31（对应丝印DI1~DI16）



```
root@test-Default-string: /home/test/test/read_gpio
root@test-Default-string:/home/test/test/read_gpio# ./rapp
the gpio8 value is:1
the gpio9 value is:1
the gpio10 value is:1
the gpio11 value is:1
the gpio12 value is:0
the gpio13 value is:1
the gpio14 value is:1
the gpio15 value is:1
```

### 四、GPIO开机初始化（BIOS）

GPIO 开机状态初始化步骤：

1. 开机进入Bios 界面
2. 选择并进入Advanced—General Purpose Input Outut
3. Set All GPIOs I/O 作用是选择GPIO 状态，GPIO 全部置为输入（Input），GPIO 全部置为输出（Output），对GPIO 逐个手动设置(Manual Mode)，Linux系统输入输出功能固定，修改该项不会生效。
4. Set All GPIOs Level 作用是设置所有GPIO 输出高电平(High)/低电平(Low)。
5. GPIO逐个手动设置(Manual Mode)。
6. 单个GPIO 设置可选Input(输入) or output(输出)，Level 高/低(注:Level 设为高，则用万用表测量GPIO 口会是低)