# 外设控制（点灯）应用

# 1. sysfs接口：

**sysfs是一种在Linux内核中提供文件系统接口的方式，它可以用于控制和监视GPIO引脚。使用sysfs接口可以通过读写特定的文件来操作GPIO引脚。**

## 1使用linux系统的GPIOsysfs直接进行IO控制

上一篇文章就有说到：[地址](https://mbb.eet-china.com/forum/topic/132518_1_1.html)

## 1.2使用shell脚本进行IO控制

#!/bin/bash

echo "hello world!"

echo "gpio44 loading user space!"

echo 44 > /sys/class/gpio/export

# set gpio44 out

echo out > /sys/class/gpio/gpio44/direction

# delay 1s

sleep 1

echo 0 > /sys/class/gpio/gpio44/value

sleep 1

echo 1 > /sys/class/gpio/gpio44/value

sleep 1

echo 0 > /sys/class/gpio/gpio44/value

echo "gpio44 loading user space!"

echo 44 > /sys/class/gpio/unexport

这个shell脚本是需要在系统root权限运行的，大致意思是gpio导入用户空间，设置gpio输出，延时1s，设置输出为0，延时1s，设置输出为1，延时1s，设置输出为0，gpio取消导入用户空间

## 1.3使用C语言进行IO控制

使用vim 创建一个test.c

vim test.c

使用gcc 编译test.c的可执行文件

gcc test.c

使用root权限运行。

sudo ./a.out

#include <fcntl.h>

#include <stdio.h>

#include <unistd.h>

#define GPIO44\_SET \

{ \

fd = open("/sys/class/gpio/gpio44/value", O\_WRONLY); \

write(fd, "1", sizeof("1")); \

close(fd); \

}

#define GPIO44\_RESET \

{ \

fd = open("/sys/class/gpio/gpio44/value", O\_WRONLY); \

write(fd, "0", sizeof("0")); \

close(fd); \

}

void read\_gpio(int \_\_fd, char \*\_\_value)

{

\_\_fd = open("/sys/class/gpio/gpio44/value", O\_RDONLY);

read(\_\_fd, \_\_value, 1);

close(\_\_fd);

printf("GPIO value: %c\n", \*\_\_value);

return;

}

int main()

{

int fd;

char value;

//chcek gpio44 exist

if (access("/sys/class/gpio/gpio44", F\_OK) == 0)

{

printf("loading gpio port\n");

fd = open("/sys/class/gpio/export", O\_WRONLY);

write(fd, "44", sizeof("44"));

close(fd);

}

printf("set gpio direction\n");

fd = open("/sys/class/gpio/gpio44/direction", O\_WRONLY);

write(fd, "out", sizeof("out"));

close(fd);

// set GPIO 44 value

printf("set gpio44 value \n");

// GPIO44\_RESET;

// fd = open("/sys/class/gpio/gpio44/value", O\_WRONLY);

// write(fd, "1", sizeof("1"));

// close(fd);

for (int i = 0; i < 10; i++)

{

read\_gpio(fd, &value);

GPIO44\_SET;

sleep(1);

read\_gpio(fd, &value);

GPIO44\_RESET;

sleep(1);

}

printf("clear GPIO44\n");

fd = open("/sys/class/gpio/unexport", O\_WRONLY);

write(fd, "44", sizeof("44"));

close(fd);

return 0;

}

输出为：

loading gpio port

set gpio direction

set gpio44 value

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

GPIO value: 0

GPIO value: 1

clear GPIO44

led将会闪烁。。。



总结：gpiosysfs的控制一般来说会使用c语言或者是shell进行控制，接下来使用VF2去实现一个驱动试试。