**Imx6q 无源蜂鸣器（BUZZER）驱动开发过程**

找出蜂鸣器元器件： KLJ-7525 无源 ，因此需要输出 PWM 波才可以使蜂鸣器发声。

KLJ-7525技术参数：

电压：2.5~4.5伏

电流：小于90

声压：大于85DB

阻抗：15

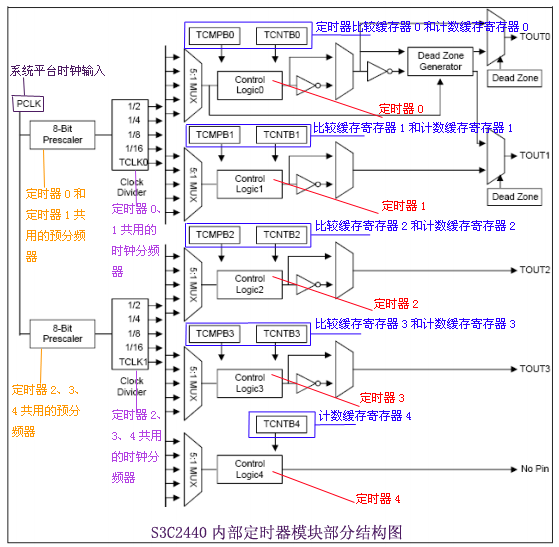
频率：2700HZ

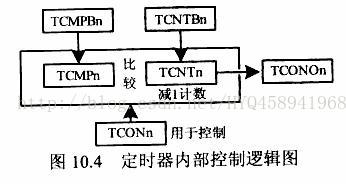
PWM （ Pulse Width Modulation ） —— 脉宽调制，它是利用微[控制器](http://www.chinabaike.com/z/a/index_47_1.html)的数字输出来对模拟电路进行控制的一种非常有效的技术

参考：

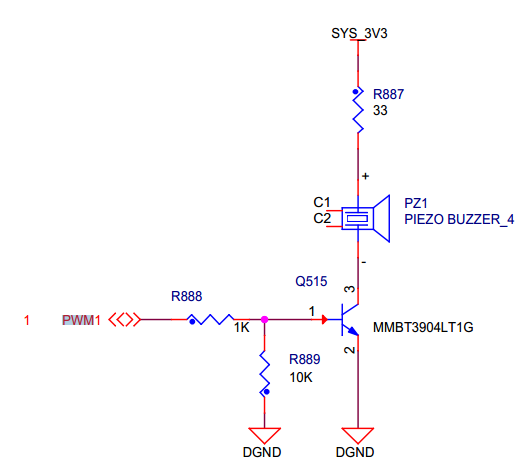
<http://blog.csdn.net/u011164819/article/details/49996791>

理解如下



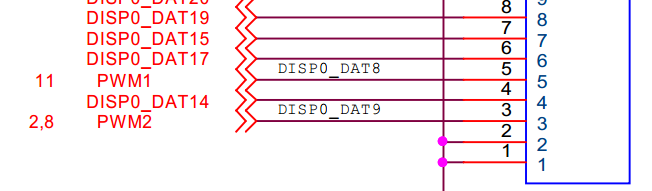


查看板子电路原理图，搜索关键词：pwm



得知：控制蜂鸣器的是 PWM1 信号

然后查看芯片原理图，查看那个管脚输出 PWM1



得知： DISP0\_DAT8 管脚

查阅 IMX6Q L3.0.35\_4.1.0\_130816\_docs 文档的

软件驱动参考手册： i.MX\_6Dual6Quad\_Linux\_Reference\_Manual.pdf

找到有关于 PWM 控制 QVGA面板的背光驱动开发例子

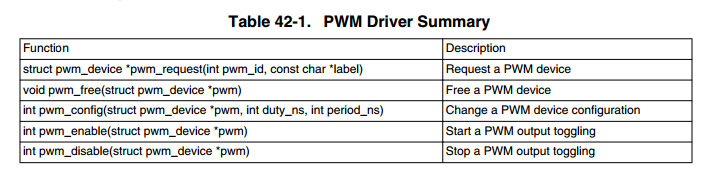
得知：

3.0.35 内核本身有关于控制 PWM 设备的接口函数

文件目录：

**\linux-3.0.35\include\linux\pwm.h**

**\linux-3.0.35\arch\arm\plat-mxc\pwm.c**



//申请一个pwm设备

struct pwm\_device \*pwm\_request(int pwm\_id, const char \*label);

//释放一个pwm设备

void pwm\_free(struct pwm\_device \*pwm);

//配置pwm设备

int pwm\_config(struct pwm\_device \*pwm, int duty\_ns, int period\_ns);

//开启pwm设备(使能)

int pwm\_enable(struct pwm\_device \*pwm);

//关闭pwm设备（使能）

void pwm\_disable(struct pwm\_device \*pwm);

配置**内核 menuconfig**

关于 PWM 控制 QVGA面板的背光驱动 Backlight Driver 的配置

• System Type > Enable PWM driver （这点需要开启）

• Select the following option to enable the Backlight driver:

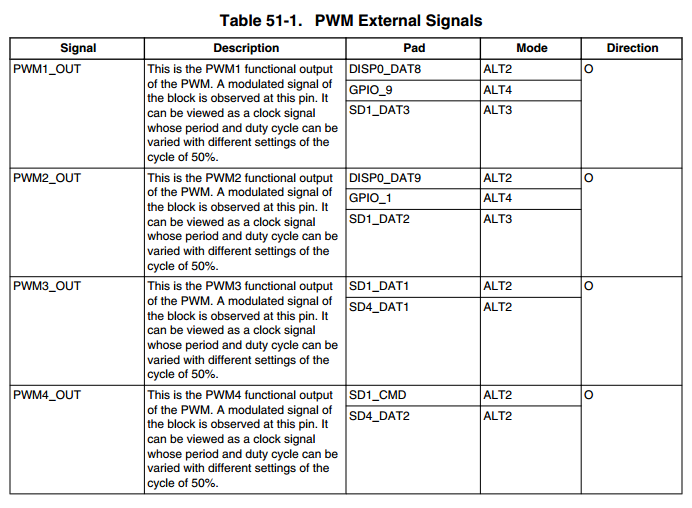
Device Drivers > Graphics support > Backlight & LCD device support > Generic PWM based Backlight Driver （这个得关闭，后面会遇到一个错误再解释）

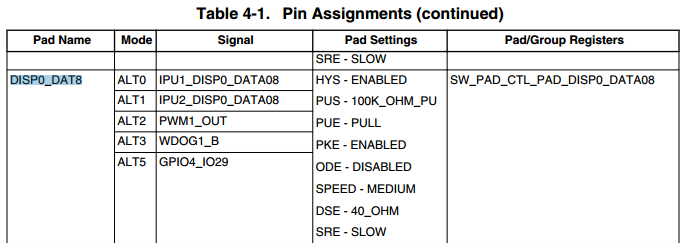
在做的过程中，又了解到 管脚复用（一个管脚有多种模式）

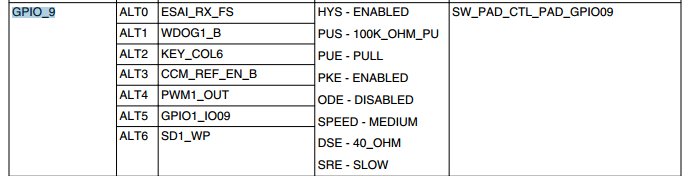
管脚查找实例 <http://www.jianshu.com/p/3c2053508342>

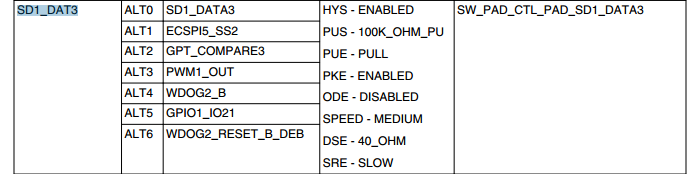
查阅 IMX6Q L4.1.15\_doc

参考手册 i.MX 6Dual 6Quad Applications Processor Reference Manual









由上得：

有3个管脚可以控制 PWM1\_OUT

DISP0\_DAT8

GPIO\_9

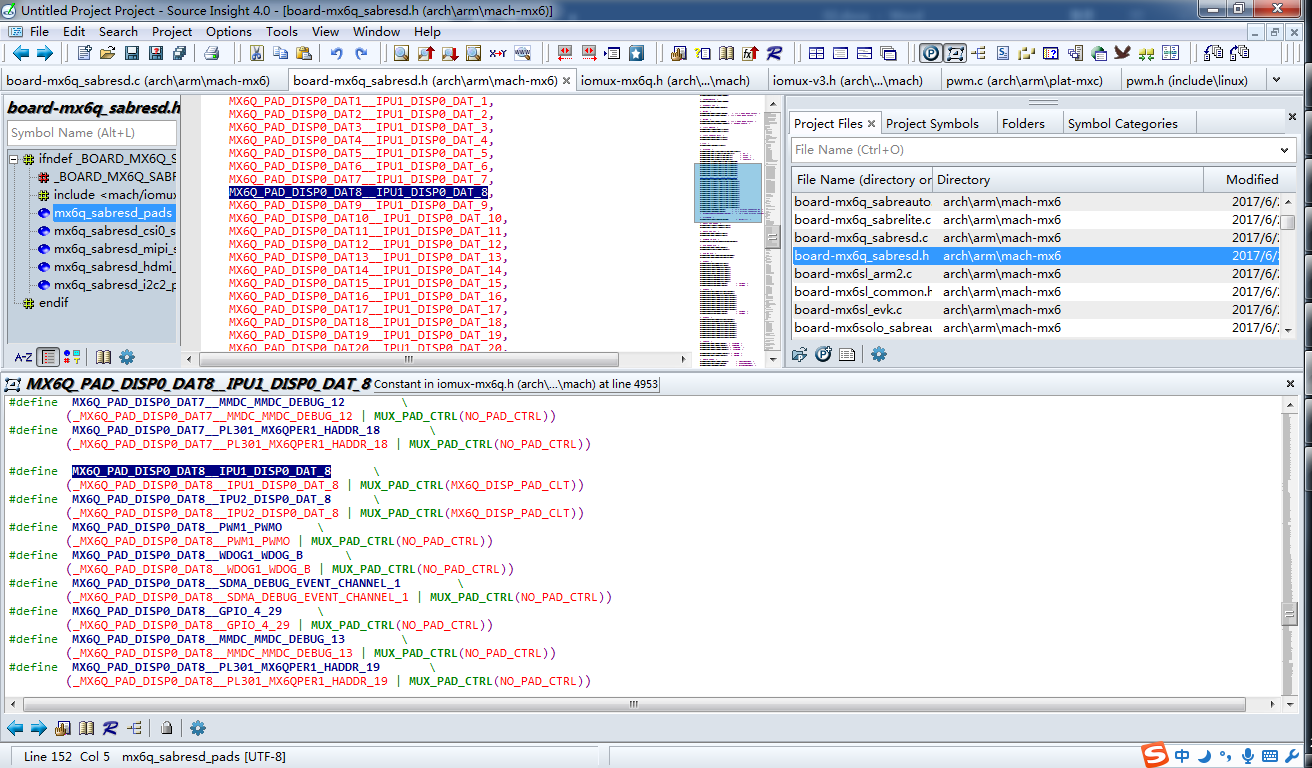
SD1\_DAT3

举例：原理图中连接蜂鸣器的是 DISP0\_DAT8 ，然而 DISP0\_DAT8 管脚有多种模式，其中 ALT2 才是对应 PWM1\_OUT 的，因此如果要使用该管脚控制 PWM1 ，就要到对应芯片的头文件修改该管脚的输出模式

对应 IMX6Q 芯片关于 DISP0\_DAT8 管脚模式的修改是在：

linux-3.0.35/arch/arm/mach-mx6/board-mx6q\_sabresd.h（或者其他board-mx6q\_xxx.h）

用SourceInsight 软件打开文件，搜索DISP0\_DAT8，双击搜索宏，点击进去



复制 MX6Q\_PAD\_DISP0\_DAT8\_\_PWM1\_PWMO

粘贴覆盖原 MX6Q\_PAD\_DISP0\_DAT8\_\_IPU1\_DISP0\_DAT\_8

重新编译内核 uImage 镜像并烧写（总共修改地方是上面提及的 menuconfig 、 管脚模式）

**简单混杂设备驱动结构**

#include <linux/init.h>

#include <linux/kernel.h>

#include <linux/module.h>

#include <linux/miscdevice.h>

#include <linux/fs.h>

#define DEVICE\_NAME "my\_buzzer" //设备节点（设备文件）名字的定义

//ioctl函数

static long pwm\_ioctl(struct file \*file, unsigned int cmd, unsigned long arg){

switch(cmd)

{

case 1:break;

}

return 0;

}

//设备的文件操作结构体

static struct file\_operations pwm\_fops = {

.owner = THIS\_MODULE,

.unlocked\_ioctl = pwm\_ioctl,

}

//注册混杂设备的3个必须参数

static struct miscdevice misc = {

.minor = MISC\_DYNAMIC\_MINOR,

.name = DEVICE\_NAME,

.fops = &pwm\_fops,

}

static int \_\_init pwm\_init(void){

misc\_register(&misc); //注册 misc 设备

//misc\_register 函数会自动创建设备节点(设备文件),无需mknod手动创建设备文件。

}

static void \_\_exit pwm\_exit(void){

misc\_deregister(&misc); ////注销 misc 设备

}

module\_init(pwm\_init);

module\_exit(pwm\_exit);

MODULE\_LICENSE("GPL");

蜂鸣器驱动：my\_buzzer.c

测试程序：my\_buzzer\_test.c

Makefile

交叉编译地址

/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib-2011.12/fsl-linaro-toolchain/bin/arm-none-linux-gnueabi-gcc

**Make .ko 错误**

**error: unknown field 'ioctl' specified in initializer**

<http://blog.csdn.net/yusiguyuan/article/details/16829299>

long (\*unlocked\_ioctl) (struct file \*, unsigned int, unsigned long);

long (\*compat\_ioctl) (struct file \*, unsigned int, unsigned long);

出错1：

当把编译好的驱动模块 .ok 文件用 insmod 命令加载进去时，程序崩了

出错信息：

root@freescale /my$ insmod buzzer\_test.ko

-- pwm\_init

-- pwm\_request\_1 success

Unable to handle kernel NULL pointer dereference at virtual address 00000014

pgd = ba074000

[00000014] \*pgd=4f8f0831, \*pte=00000000, \*ppte=00000000

Internal error: Oops: 17 [#1] PREEMPT SMP

Modules linked in: buzzer\_test(+)

CPU: 2 Not tainted (3.0.35-2666-gbdde708 #3)

PC is at pwm\_config+0x28/0x150

LR is at pwm\_init+0x44/0xe8 [buzzer\_test]

pc : [<80065e58>] lr : [<7f002044>] psr: 20000013

sp : bfc67e88 ip : 00000001 fp : 800a5144

r10: 7f002000 r9 : 7f000394 r8 : 00000001

r7 : 00000000 r6 : fffffff0 r5 : bfc66000 r4 : 0007a120

r3 : 00000001 r2 : 0007a120 r1 : 00000000 r0 : fffffff0

Flags: nzCv IRQs on FIQs on Mode SVC\_32 ISA ARM Segment user

Control: 10c53c7d Table: 4a07404a DAC: 00000015

Process insmod (pid: 3284, stack limit = 0xbfc662f0)

Stack: (0xbfc67e88 to 0xbfc68000)

7e80: 7f0004d4 bfc66000 00000001 00000000 00000001 7f000394

7ea0: 7f002000 7f002044 80af3e80 8003c4e8 00000001 800ebdc8 8bf44de0 7f002000

7ec0: 80000013 8050bd94 00000002 ba26fa40 00000000 800a7b80 00000001 7f000394

7ee0: 7f0003dc 00000001 ba1c55c0 00000001 7f000394 800a505c 800a5144 800a7338

7f00: 7f0003a0 2ab0b000 00000000 00000124 00000000 000a7008 7f0004c0 7f0003a0

7f20: 0000dc12 c0af8000 0000167e c0af8b08 c0af89c1 c0af9548 ba1c5500 000004d8

7f40: 00000558 00000000 00000000 0000001d 0000001e 00000015 00000012 00000010

7f60: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 80a49d24

7f80: 00000003 0000167e 7ed72dd4 7ed72eda 00000080 80042144 bfc66000 00000000

7fa0: 00000000 80041fc0 0000167e 7ed72dd4 000a7018 0000167e 000a7008 7ed72eda

7fc0: 0000167e 7ed72dd4 7ed72eda 00000080 000001de 00000000 2ab0b000 00000000

7fe0: 7ed72c10 7ed72c00 0001a32c 2acf0490 60000010 000a7018 00000000 00000000

[<80065e58>] (pwm\_config+0x28/0x150) from [<7f002044>] (pwm\_init+0x44/0xe8 [buzzer\_test])

[<7f002044>] (pwm\_init+0x44/0xe8 [buzzer\_test]) from [<8003c4e8>] (do\_one\_initcall+0x34/0x174)

[<8003c4e8>] (do\_one\_initcall+0x34/0x174) from [<800a7338>]

程序执行到驱动初始化函数的pwm\_config函数，后面的调试信息全没了。

原因：我第一次加载驱动，不懂管脚模式的修改，而且 QVGA面板的背光驱动 Backlight Driver 占用了PWM ，后来我去内核的menuconfig 把Backlight Driver关闭了就可以成功加载

Device Drivers > Graphics support > Backlight & LCD device support > Generic PWM based Backlight Driver （这个得关闭）

出错2：

蜂鸣器驱动模块是成功加载了，但不管测试程序怎么测，蜂鸣器都不出声

原因：就是之前没改管脚模式

问题：