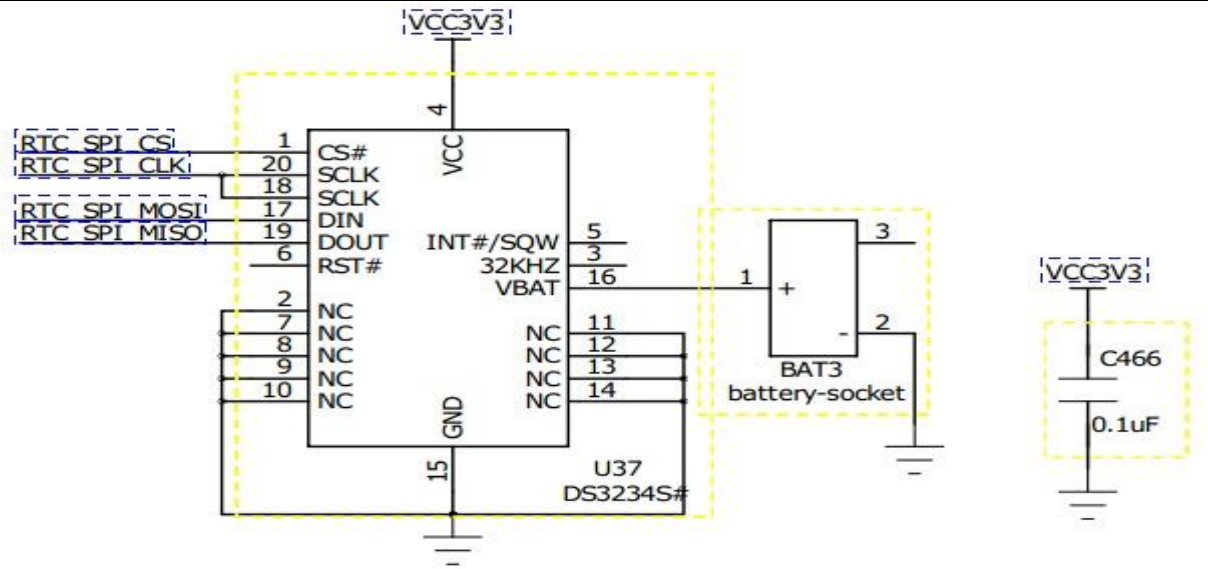


mx53 DS3234 驱动程序及测试程序说明

1、引脚定义说明

Ball	Pad name	Function	ALT mode	ALT function
W1	EIM_D20	RTC_SPI_CLK	ALT1	GPIO3_GPIO[20]
W2	EIM_D22	RTC_SPI_CS	ALT1	GPIO3_GPIO[22]
Y1	EIM_D23	RTC_SPI_MISO	ALT1	GPIO3_GPIO[23]
Y2	EIM_D24	RTC_SPI_MOSI	ALT1	GPIO3_GPIO[24]



2、源文件清单



北京栢园生物设备有限公司

Beijing Zheyuan Biological Equipment Co., Ltd.



源文件清单

设备	文件名称	功能
DS3234	imx5_ds3234_device.c	定义设备资源
	imx5_ds3234_driver.c	驱动实现
	Makefile	编译文件
	test_ds3234.c	读写控制寄存器的测试程序
	test_ds3234_readtime.c	读取实时时钟的测试程序，每秒读一次

设备文件清单

设备	设备文件	类型	主设备号	次设备号	用途
DS3234	/dev/imx5_ds3234	c	1988	0	实时时钟芯片

3、测试说明

DS3234 是一个具有 4 线制接口兼容 SPI 串行协议的实时时钟芯片，mx53 通过 GPIO 模拟时序与 DS3234 通信，读写其寄存器。为了方便用户层应用程序的编写，驱动层只实现了基本的读写操作。具体到寄存器的地址，寄存器数据分析，都放在应用层来做。

用户层程序通过标准的 read()和 write()系统调用操作 DS3234。为了跟驱动程序交互，需要定义一个寄存器结构体，如下：

```
typedef struct{
    char addr; /*register for ds3234*/
    char val; /*value for register*/
}
```

```
/*in read mode,the val will hold the value of physical register*/
```

```
/*in write mode,the val will be written to physical register*/
```

```
}IMX5_DS3234_REG;
```

例 1：读秒寄存器的值，从数据手册中可以查到秒寄存器的读地址为 0x00，则代码如下：

```
IMX5_DS3234_REG reg;  
reg.addr=0x00;  
read(fd,&reg,sizeof(IMX5_DS3234_REG));  
printf("read back seconds is :%d\n",reg.val);
```

例 2：将秒寄存器的值写成 0x45，从数据手册中可以查到秒寄存器的写地址为 0x80，则代码如下：

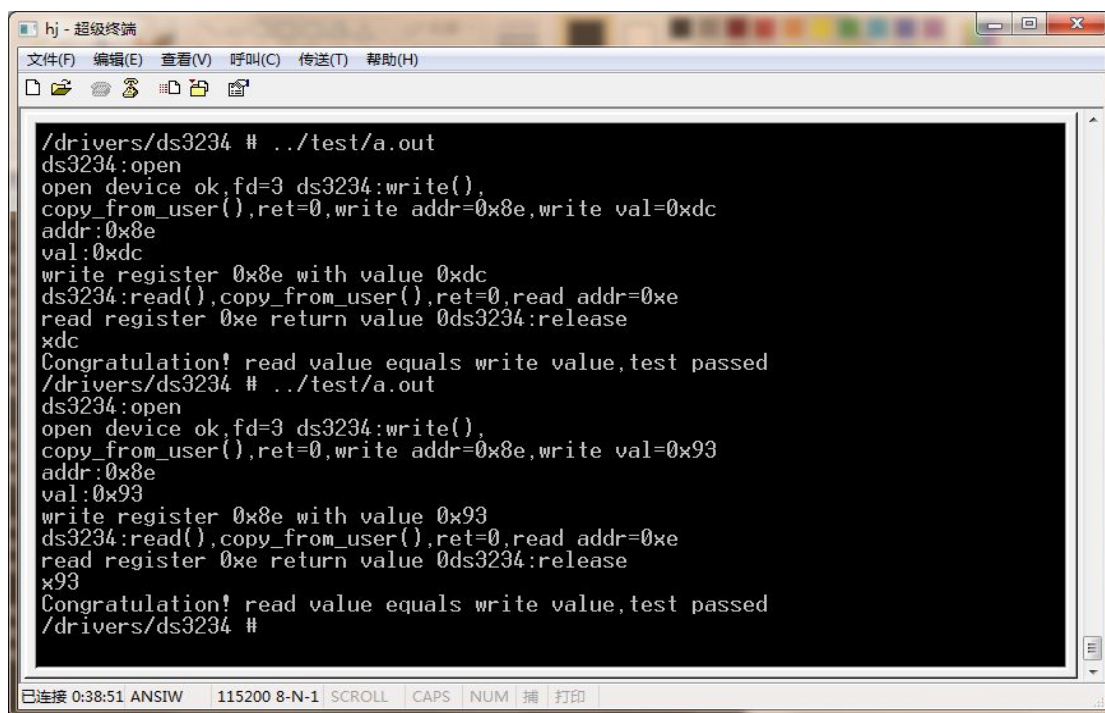
```
IMX5_DS3234_REG reg;  
reg.addr=0x80;  
reg.val=0x45;  
write(fd,&reg,sizeof(IMX5_DS3234_REG));
```

测试结果：

1、实时时钟测试

```
ds3234:read(),copy_from_user(),ret=0,read addr=0x1  
ds3234:read(),copy_from_user(),ret=0,read addr=0x0  
current time: 0:32:58  
ds3234:read(),copy_from_user(),ret=0,read addr=0x2  
ds3234:read(),copy_from_user(),ret=0,read addr=0x1  
ds3234:read(),copy_from_user(),ret=0,read addr=0x0  
current time: 0:32:59  
ds3234:read(),copy_from_user(),ret=0,read addr=0x2  
ds3234:read(),copy_from_user(),ret=0,read addr=0x1  
ds3234:read(),copy_from_user(),ret=0,read addr=0x0  
current time: 0:33:0  
ds3234:read(),copy_from_user(),ret=0,read addr=0x2  
ds3234:read(),copy_from_user(),ret=0,read addr=0x1  
ds3234:read(),copy_from_user(),ret=0,read addr=0x0  
current time: 0:33:1  
ds3234:read(),copy_from_user(),ret=0,read addr=0x2  
ds3234:read(),copy_from_user(),ret=0,read addr=0x1  
ds3234:read(),copy_from_user(),ret=0,read addr=0x0  
current time: 0:33:2  
ds3234:read(),copy_from_user(),ret=0,read addr=0x2  
ds3234:read(),copy_from_user(),ret=0,read addr=0x1  
ds3234:read(),copy_from_user(),ret=0,read addr=0x0  
current time: 0:33:3
```

2、读写寄存器测试



```
/drivers/ds3234 # ../test/a.out
ds3234:open
open device ok,fd=3 ds3234:write(),
copy_from_user(),ret=0,write addr=0x8e,write val=0xdc
addr:0x8e
val:0xdc
write register 0x8e with value 0xdc
ds3234:read(),copy_from_user(),ret=0,read addr=0xe
read register 0xe return value 0ds3234:release
xdc
Congratulation! read value equals write value, test passed
/drivers/ds3234 # ../test/a.out
ds3234:open
open device ok,fd=3 ds3234:write(),
copy_from_user(),ret=0,write addr=0x8e,write val=0x93
addr:0x8e
val:0x93
write register 0x8e with value 0x93
ds3234:read(),copy_from_user(),ret=0,read addr=0xe
read register 0xe return value 0ds3234:release
x93
Congratulation! read value equals write value, test passed
/drivers/ds3234 #
```

注意：

控制寄存器的 BIT 5 位用于启动温度转换。置位只发生在当前不在转换状态中。用户置位时应该先检测状态位 BSY。否则会置位失败！



shell.albert
2012/12/25