# iic & spi 接口及操作说明

## 1. IIC接口说明

### 1.1概述

AC54系列芯片包含2个硬件iic模块，如果不够用可添加软件iic模块，软件iic个数不限。

### 1.2 iic接口

注册iic设备

在板级文件board\_xxx.c文件中定义iic设备信息

/\* 硬件iic0 \*/

HW\_IIC0\_PLATFORM\_DATA\_BEGIN(hw\_iic0\_data)

.clk\_pin = IO\_PORTB\_04,

.dat\_pin = IO\_PORTB\_03,

.baudrate = 0x1f,

HW\_IIC0\_PLATFORM\_DATA\_END()

/\* 硬件iic1 \*/

HW\_IIC1\_PLATFORM\_DATA\_BEGIN(hw\_iic1\_data)

.clk\_pin = IO\_PORTB\_00,

.dat\_pin = IO\_PORTB\_01,

.baudrate = 0x1f,

HW\_IIC1\_PLATFORM\_DATA\_END()

/\* 软件件iic2 \*/

SW\_IIC\_PLATFORM\_DATA\_BEGIN(sw\_iic2\_data)

.clk\_pin = IO\_PORTB\_00,

.dat\_pin = IO\_PORTB\_01,

.sw\_iic\_delay = 50, //io翻转延时，相当于波特率

SW\_IIC\_PLATFORM\_DATA\_END()

/\* 软件件iic3 \*/

SW\_IIC\_PLATFORM\_DATA\_BEGIN(sw\_iic3\_data)

.clk\_pin = IO\_PORTB\_00,

.dat\_pin = IO\_PORTB\_01,

.sw\_iic\_delay = 50,

SW\_IIC\_PLATFORM\_DATA\_END()

在REGISTER\_DEVICES(device\_table) = { 中添加设备

{“iic0”, &iic\_dev\_ops, (void \*)hw\_iic0\_data },

{“iic1”, &iic\_dev\_ops, (void \*)hw\_iic1\_data },

{“iic2”, &iic\_dev\_ops, (void \*)sw\_iic2\_data },

{“iic3”, &iic\_dev\_ops, (void \*)sw\_iic3\_data },

### 1.3 iic操作步骤

打开设备

void \*fd = dev\_open(“iic0”, NULL);

发送ioctl命令

dev\_ioctl(fd, IIC\_IOCTL\_START, 0); //开始发送，获取IIC控制权

dev\_ioctl(fd, IIC\_IOCTL\_TX\_WITH\_START\_BIT, dat1);

dev\_ioctl(fd, IIC\_IOCTL\_TX, dat2);

dev\_ioctl(fd, IIC\_IOCTL\_TX\_WITH\_STOP\_BIT, dat3);

/\*

....

\*/

dev\_ioctl(fd, IIC\_IOCTL\_STOP, 0); //结束发送，让出IIC控制权

关闭设备

dev\_close(fd);

支持的IOCTL命令

IIC\_IOCTL\_TX\_START\_BIT

IIC\_IOCTL\_TX\_WITH\_START\_BIT

IIC\_IOCTL\_TX\_STOP\_BIT

IIC\_IOCTL\_TX

IIC\_IOCTL\_TX\_WITH\_STOP\_BIT

IIC\_IOCTL\_RX

IIC\_IOCTL\_RX\_WITH\_STOP\_BIT

IIC\_IOCTL\_RX\_WITH\_NOACK

IIC\_IOCTL\_RX\_WITH\_ACK

IIC\_IOCTL\_SET\_NORMAT\_RATE //dma方式的波特率

IIC\_IOCTL\_START

IIC\_IOCTL\_STOP

扩展IOCTL命令

如果需要扩展iic的命令，可通过实现下面函数的方式实现：

int iic\_ioctl\_ex(struct iic\_device \*iic, int cmd, int arg)

{

switch(cmd) {

case IOCTL\_xxxx1: //自定义命令

/\*

命令实现

软件iic：(struct software\_iic \*)iic->hw

硬件iic：(struct hardware\_iic \*)iic->hw

\*/

break;

case IOCTL\_xxxx2:

break;

default:

return false;

}

return true;

}

## spi接口说明

### 2.1 概述

AC54系列芯片包含3个spi模块

### 2.2 注册spi设备

在板级文件board\_xxx.c文件中定义spi设备信息

SPI0\_PLATFORM\_DATA\_BEGIN(spi0\_data)

.clk = 60000000,

.mode = SPI\_DUAL\_MODE,

.port = SPI\_PORTA\_0\_4\_PORTH15,

SPI0\_PLATFORM\_DATA\_END()

/\*

clk：时钟要能够被clk\_get(“spi”) 整除

mode: spi 模式

port: 出口选择，请参考include\_lib/system/cpu/AC54xx/asm/spi.h

\*/

SPI1\_PLATFORM\_DATA\_BEGIN(spi1\_data)

.clk = 60000000,

.mode = SPI\_DUAL\_MODE,

.port = SPI\_PORTC\_4\_6,

SPI1\_PLATFORM\_DATA\_END()

SPI2\_PLATFORM\_DATA\_BEGIN(spi2\_data)

.clk = 60000000,

.mode = SPI\_DUAL\_MODE,

.port = SPI\_PORTD\_0\_5,

SPI2\_PLATFORM\_DATA\_END()

在REGISTER\_DEVICES(device\_table) = { 中添加设备

{“spi0”, &spi\_dev\_ops, (void \*)spi0\_data },

{“spi1”, &spi\_dev\_ops, (void \*)spi1\_data },

{“spi2”, &spi\_dev\_ops, (void \*)spi2\_data },

### 2.3 操作步骤

打开设备

void \*fd = dev\_open(“spi1”, NULL);

更换cs脚

void cs\_pin\_func(int cs)

{

gpio\_direction\_output(IO\_PORTX\_XX, cs);

}

dev\_ioctl(fd, IOCTL\_SET\_CS\_PORT\_FUNC, cs\_pin\_func);

cs脚控制

dev\_ioctl(fd, IOCTL\_SET\_CS, 0)

读写

u8 byte;

/\*

读写1byte

\*/

dev\_ioctl(fd, IOCTL\_SPI\_READ\_BYTE, &byte);

dev\_ioctl(fd, IOCTL\_SPI\_WRITE\_BYTE, 0xaa);

/\*

dma读写

\*/

dev\_read(fd, buf, len);

dev\_write(fd, buf, len);

关闭设备

dev\_close(fd);