

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 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 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;
}
```

2.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);
```