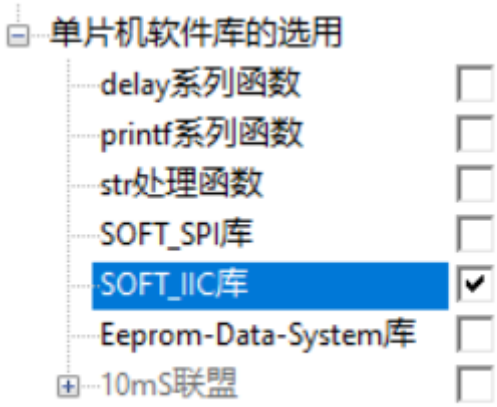
SOFT_IIC库

SOFT_IIC库就是用代码模拟的IIC，优点是IO口可以随意安排，缺点是速度会慢一些。在使用本库之前先到ecbm_core.h里使能。双击打开ecbm_core.h文件，然后进入图形化配置界面，使能SOFT_IIC库。



API

soft_iic_init

函数原型：void soft_iic_init(soft_iic_def * dev,u8 scl,u8 sda);

描述

软件IIC主机初始化函数。

输入

- scl：信号线对应的IO口。
- sda：数据线对应的IO口。

输出

- dev：软件IIC的器件信息包。

返回值

无

调用例程

```
#include "ecbm_core.h"//加载库函数的头文件。
soft_iic_def at24; //定义一个软件IIC操作对象，名字随意，但是推荐和目标器件有联系，
方便记忆。
void main(){//main函数，必须的。
    system_init();//系统初始化函数。
    soft_iic_init(&at24,D10,D11);//该对象所连接的引脚，先输入时钟脚SCL，再输入数据脚SDA。
    while(1){
    }
}
```

注意事项

1. 一定要先定义一个结构体实例，比如上面的at24。这个是接下来软件IIC操作的重要标识，先定义再执行本初始化函数。
2. 在没有利用中断的情况下，软件实现的IIC几乎没有办法做到实时从机。为了不占用宝贵的中断资源，以后也不会有软件IIC从机的开发计划。

soft_iic_set_pin

函数原型：void soft_iic_set_pin(soft_iic_def * dev);

描述

软件IIC引脚切换函数。在多器件的应用下，切换IIC函数的操作对象。

输入

- dev：切换的目标器件信息包。

输出

无

返回值

无

调用例程

```
soft_iic_set_pin(&sht30); //将IIC对象切换到名为SHT30的信息包。
val=get_temp(); //读取SHT30的温度数据。
soft_iic_set_pin(&at24c02);//将IIC对象切换到名为at24c02的信息包。
at24_write(0,val); //将温度数据写入到eeprom（24c02）里。
```

注意事项

1. 软件IIC的操作函数都是对默认器件的操作，所以当需要操作多个器件时，一定要先切换。
2. 如果只操作一个器件，或者多个器件都是在一条IIC总线上的话，也不需要执行本函数，因为在初始化的时候就已经弄好了。

操作函数

函数原型：

- void soft_iic_start(void);
- void soft_iic_stop(void);
- void soft_iic_write(u8 dat);
- void soft_iic_write_ack(void);
- void soft_iic_write_nack(void);
- u8 soft_iic_read(void);
- bit soft_iic_read_ack(void);

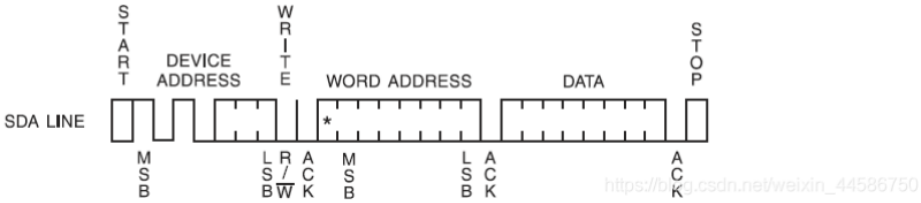
描述

软件IIC的操作相关函数，需要根据实际时序去调用。

调用例程

比如下图是AT24C02芯片的写一个字节的时序：

Figure 8. Byte Write



可以看出来时序动作是：【开始】【写器件地址和写入位】【等ACK】【写片内地址】【等ACK】【写数据】【等ACK】【结束】。所以可以得到如下代码。

```
void at24c02_write_byte(u8 addr,u8 dat){
    soft_iic_start(); //开始。
    soft_iic_write(0xA0);//写器件地址和写入位。
    soft_iic_read_ack(); //等从机ACK。
    soft_iic_write(addr);//写片内地址。
    soft_iic_read_ack(); //等从机ACK。
    soft_iic_write(dat); //写数据。
    soft_iic_read_ack(); //等从机ACK。
    soft_iic_stop(); //结束。
}
```

注意事项

无

优化建议

本库比较简单，所以没有可优化的地方。