【嵌⼊式⼯程师⾯试⾼频问题】你知道IIC吗（附程序说明）

在嵌⼊式⾯试中，需要能够熟悉IIC的⼯作原理，使⽤场景，并且需要能够⼿撕IIC的时序图和写⼀个IIC读写程序。

IIC的⼀些⼯作原理、时序图在《》这篇博客介绍已经很全⾯了，我这⾥主要就是补充IIC的代码部分。顺便补充⼀下IIC和SPI以及UART的⼀些区别。

IIC读写特点

IIC最⼤的特点就是只使⽤两根线进⾏通讯，⽀持多个主服务器和多个从服务器。

IIC只有两根线，⽽且⼜不像SPI⼀样有⽚选线，那怎么区分地址呢？其实IIC总线上每个设备都有⾃⼰的地址，有的是芯⽚⾥⾯定死的，如果⾃⼰⽤GPIO模拟则可以⾃⼰定⼀个地址。



IIC、SPI、UART区别

SPI：4根线，全双⼯同步，速率最⼤可10Mbps，

I2C：2根线，半双⼯同步，速率⼀般是1Mbps，

UART：2根线，全双⼯异步，速率⼀般不超过20Kbps

UART和SPI、I2C两者最⼤的区别在于后⾯两者是同步的，要有统⼀的时钟线，⽽UART不需要，接线⽅便，但收发双⽅都需要配置相同的波特率，即设备要有⾃⼰的时钟源；

I2C协议中是有地址规则的，因此在⼀对多通信的时候更有优势。常⽤来连接传感器；

SPI编程简单，在设备不多，端⼝充⾜，对通信速度要求不⾼的场合可以应⽤，较常⽤来连接EEPROM、FLASH或液晶显⽰器。

SPI优点：⽀持全双⼯操作；操作简单；数据传输速率较⾼。SPI没有定义速度限制，⼀般的实现通常能达到甚⾄超过10 Mbps。

SPI缺点：需要占⽤主机较多的⼝线（每个从机都需要⼀根⽚选线）；只⽀持单个主机；没有指定的流控制，没有应答机制确认是否接收到数据。

I2C优点：只使⽤两根电线，⽀持多个主服务器和多个从服务器，ACK / NACK位确认每个帧都已成功传输，硬件没有UART那么复杂。

I2C缺点：数据传输速率⽐SPI慢(IIC 最⾼的速度也就快速+模式（1 Mbps）和⾼速模式（3.4 Mbps）)，数据帧的⼤⼩限制为8位，实现⽐SPI更复杂的硬件。

UART优点：只使⽤两根电线，不需要时钟信号，有⼀个奇偶校验位，只要双⽅设置后，就可以改变数据包的结构。

UART缺点：数据帧的⼤⼩限制为最多9位，不⽀持多个从属或多个主系统，速度最慢。

IIC⼋种状态和对应代码

其实IIC的通信就是通过两根线的⼋种状态完成的。\*\*两根线怎么会有⼋种状态？\*\*其实⼀根线有⾼电平、低电平、上升沿、下降沿四种状态。



数据线和时钟线组成的8种状态。



下⾯的代码就是⽤IO⼝去模拟IIC。实验是在STM32F103芯⽚做的，是能够跑通的。



SDA\_OUT();//sda输出

IIC\_SDA=1;

IIC\_SCL=1;

delay\_us(4);

IIC\_SDA=0;//START:when CLK is high,DATA change form high to low

delay\_us(4);

IIC\_SCL=0;//铅住I2C总线，准备发送或接收数据

}



//IIC发送⼀个字节

//返回从机有⽆应答

//1，有应答

//0，⽆应答

void IIC\_Send\_Byte(u8 txd)

{

u8 t;

SDA\_OUT();

IIC\_SCL=0;//拉低时钟开始数据传输

for(t=0;t<8;t++)

{

IIC\_SDA=(txd&0x80)>>7;//往SDA上放数据，放txd最⾼位

txd<<=1;

delay\_us(2);//对TEA5767这三个延时都是必须的

IIC\_SCL=1;

delay\_us(2);

IIC\_SCL=0;

delay\_us(2);

}

}



//等待应答信号到来

//返回值：1，接收应答失败

// 0，接收应答成功

u8 IIC\_Wait\_Ack(void)

{

u8 ucErrTime=0;

SDA\_IN();//SDA设置为输⼊

IIC\_SDA=1;delay\_us(1);//SDA设置为1，释放SDA总线

IIC\_SCL=1;delay\_us(1);

while(READ\_SDA)//读SDA，如果SDA⼀直为1，则⽆应答；如果SDA为0，则应答成功{

ucErrTime++;

if(ucErrTime>250)

{

IIC\_Stop();

return1;

}

}

IIC\_SCL=0;//时钟输出0

return0;

}



IIC\_SCL=0;

SDA\_OUT();

IIC\_SDA=0;

delay\_us(2);

IIC\_SCL=1;

delay\_us(2);

IIC\_SCL=0;

}



//读数据

//返回值就是接收到的⼀个字节

u8 IIC\_Read\_Byte(unsigned char ack)

{

unsigned char i,receive=0;

SDA\_IN();//SDA设置为输⼊

for(i=0;i<8;i++)

{

IIC\_SCL=0;

delay\_us(2);

IIC\_SCL=1;//SCL为⾼的时候读取数据

receive<<=1;//每读⼀次，receive左移⼀位

if(READ\_SDA)receive++;//读取SDA放在receive最低位

delay\_us(1);

}

if(!ack)

IIC\_NAck();//发送nACK

else

IIC\_Ack();//发送ACK

return receive;

}



//产⽣IIC停⽌信号

void IIC\_Stop(void)

{

SDA\_OUT();//sda输出

IIC\_SCL=0;

IIC\_SDA=0;//STOP:when CLK is high DATA change form low to high delay\_us(4);

IIC\_SCL=1;

IIC\_SDA=1;//发送I2C总线结束信号

delay\_us(4);

}