

IIC 总线

支持标准模式（高达 100K 比特/秒）

快速模式（高达 400K 比特/秒）

3 个 IIC 接口，读写分别有 32BYTE 的 FIFO，

有两个 DMA 通道和 1 个中断线

IIC0 支持唤醒

支持 7 位和 10 位地址格式

模块的复位：

- 1、通过 I2C_SYS 寄存器
- 2、I2C_CON 的 I2C_EN 位

发送数据：

- 1、轮询发送数据，查询准备标志(I2C_IRQSTATUS_RAW 的 位 XRDY)
- 2、发送数据，写 I2C_DATA 寄存器

接收数据：

- 1、标志位(I2C_IRQSTATUS_RAW 的 位 RRDY)
- 2、读取数据 I2C_DATA

寄存器：

I2C_REVNB_LO：模块版本号

I2C_REVNB_HI

I2C_SYSC：模块复位，自动待机，唤醒，待机模式，时钟选择位

中断相关的：

I2C_IRQSTATUS_RAW：显示所有信息（启用和禁用中断）

I2C_IRQSTATUS：显示当前的中断状态信息，写 1 清除

AT：仲裁失败	NACK：无应答中断	ARDY：准备就绪
RRDY：接收就绪	XRDY：发送就绪	GC：通用 IRQ？
STC：启动条件检测	AERR：访问错误	BF：总线空闲
AAS：确认地址从站	XUDF：传送溢出	ROVR：接收溢出
BB：总线忙	RDR：接收完成？	XDR：发送完成？

I2C_IRQENABLE_SET：中断使能

I2C_IRQENABLE_CLR：中断清除

I2C_WE：唤醒寄存器

I2C_DMARXENABLE_SET：使用 DMA 接收

I2C_DMATXENABLE_SET: 使用 DMA 发送

I2C_DMARXENABLE_CLR:

I2C_DMATXENABLE_CLR:

I2C_DMARXWAKE_EN: DMA 接收唤醒

I2C_DMATXWAKE_EN: DMA 发送唤醒

I2C_SYSS: 复位状态

I2C_BUF: 接收、发送的 FIFO 触发值，启用 DMA，FIFO 复位

作者：A Xian

qq：281453291

I2C_CNT: 设置接收或发送的字节数，此寄存器不包括设置 7Bit 地址的，直接是操作数据

I2C_DATA: 读写数据寄存器

I2C_CON:

I2C_EN: IIC 使能位，=1 开启 IIC 功能

OPMODE: 速度操作模式选择，只有 1 种模式，快速和标准模式

STB: 发送 1 个启动 Bit

MST: 主模式还是从模式

TRX: 在从模式下，是接收还是发送，和 MST 位一起决定

The operating modes are defined as follows:

MST	TRX	Operating Modes
0	x	Slave receiver
0	x	Slave transmitter
1	0	Master receiver
1	1	Master transmitter

XSA: 扩展从地址配置，7 位地址模式或 10 位地址模式

XOA0: 我们是从机，本机的地址配置，是 7 位还是 10 位

XOA1、XOA2、XOA3: 同上

作者：A Xian

STP: 产生停止条件，=1 产生

qq：281453291

STT: 产生 S 信号，和 STP 一起决定状态

STT、STP 根据这两位判断是读还是写，也就是下面的那个图片，最后 1 个 BIT 发出 R 或 W，操作 STT 和 STP 前，必须要把从机地址写好，因为操作后他会自动发送出启动信号和从机地址（这里包括 RW 位）

STT	STP	Conditions	Bus Activities
1	0	Start	S-A-D
0	1	Stop	P
1	1	Start-Stop (DCOUNT= n)	S-A-D..(n)..D-P
1	0	Start (DCOUNT= n)	S-A-D..(n)..D

1: 0 模式，可以用于重发 START 信号，就是在读模式，先发出主机地址，然后发出要读取的字地址，也可以用于发送 START 信号，然后开始接收从机发来的数据（在 MASTER

模式下，此模式要设置 MST 和 TRX)

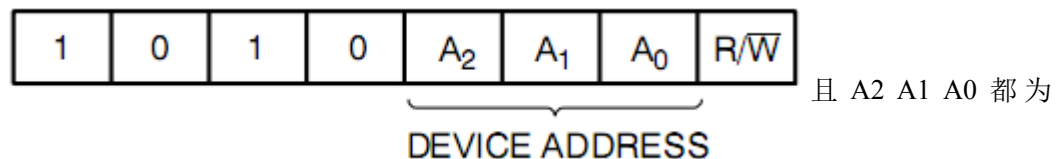
1: 1 模式，写 IIC 设备，可以使用此模式，停止条件根据 DCNT 寄存器的值

MST: 是设置主机还是从机模式 作者: A Xian

TRX: 是设置接收还是发送模式 qq: 281453291

I2C_OA: IIC 模块自身的地址

I2C_SA: IIC 模块从机的 7 位或 10 地址，假设有一个 IIC 设备，地址表如下:



0，那么设置 SA=0x50，最后一位 RW 位不计算在内，最后一位不用写

时钟设置寄存器

I2C_PSC: 预分频时钟，内部时钟，就是 I2C 模块的使用的时钟

I2C_SCLL: 输出的时钟 L

I2C_SCLH: 输出的时钟 H

CPU 给 IIC 模块的时钟是 48M，那么假设要输出 100K 的时钟，可以将 PSC 设置为 4，然后模块就有 12M 的时钟了，在分别设置 SCLL 和 SCLH，让其输出 100K，

I2C_SYSTEST: 测试模式 作者: A Xian

强制输出时钟，读取输出 SDA qq: 281453291

ST_EN: 测试模式启用停止

TMODE: 测试模式

I2C_BUFSTAT: buf 状态（发送和接收），FIFO 内部总个数

本机地址配置寄存器 在 slave 模式下使用

I2C_OA1、I2C_OA2、I2C_OA3: 本机地址 1

I2C_ACTOA: 是否激活了本机地址

I2C_SBLOCK: 作为 SLAVE 模式，自动判断 oa 地址是否正确

操作:

作为 MASTER 模式，要对设备的读写，设置好波特率，FIFO

- 1、设置好设备的地址格式，7BIT，还是 10BIT 地址，其中地址的最低 1 个 BIT 不用设置，且设置地址的时候当做没有这位地址，不用保留
- 2、设置好 CNT 寄存器，要发送多少个数据，或者要接收多少个数据
- 3、设置好 CON 寄存器，特别是，主传送模式，主接收模式，发送数据的格式等

CON 寄存器，根据 MST 和 TXR 位，设置为主接收或发送模式，这里接收或发送模式，会自动设置上面那条的最低位，代表读写，所以上面那个地址最低位我们不用关心了，这里会帮我们自动设置好

STT 和 STP 控制 START，DATA，STOP 信号，也要设置好 作者：A Xian

4、开始写或者读 DATA 寄存器，操作之前同时要判断状态位 qq：281453291

所有的操作主要还是在 CON 寄存器和 SA 寄存器上，例如是发送 START 还是发送 STOP 信号，都是在 STT 和 STP 位控制的，所以这两个位要在所有配置都完成后，才可以操作，然后在操作 DATA 寄存器，发送读取数据