

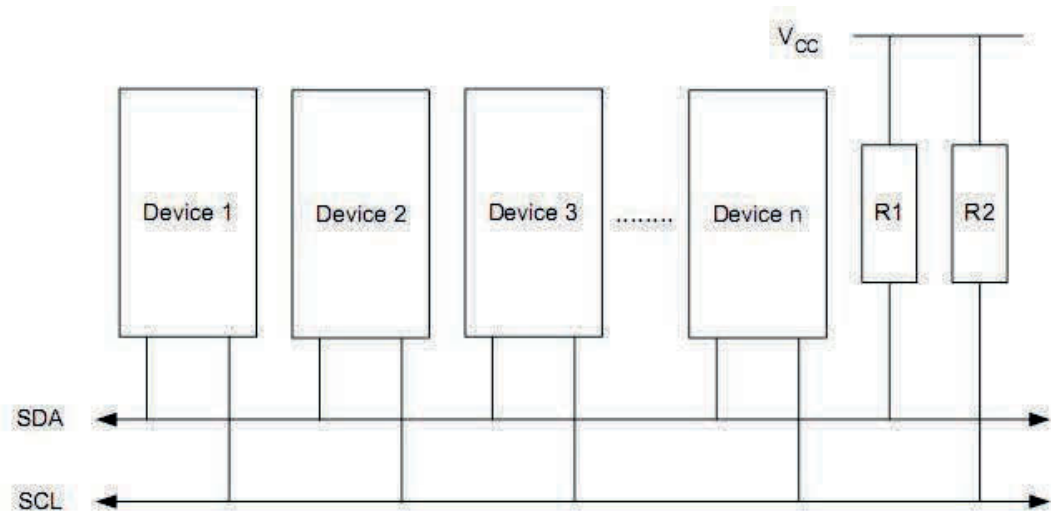
第 9 章 TWI/IIC

9.1 概述

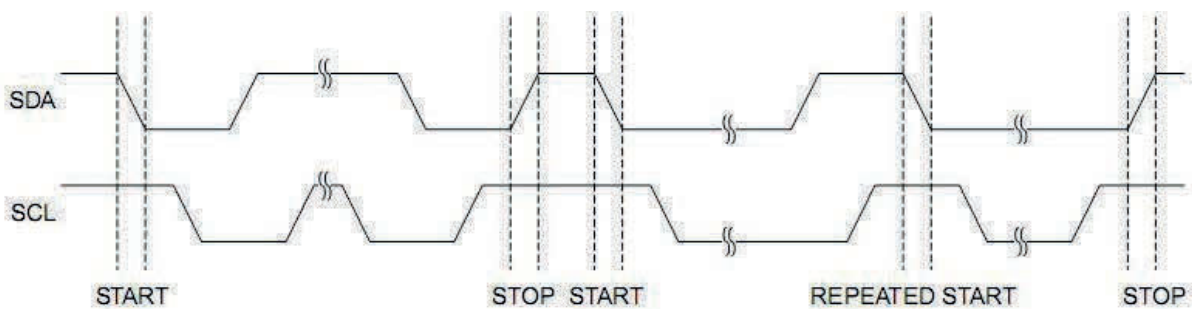
TWI 或者 IIC 是两线的总线。这原本是前几章的笔记，但是一开始因为觉得 IIC 太复杂了所以延迟到最后在练习。其实不是这样的，AVR 的 IIC 硬件，从手册上看是很复杂。但是只要慢慢的分类再分类就觉得很容易了。当然一开始确实要下一点功夫，但是原理理解了，程式跑起来后。你就会觉得，IIC 很有趣。

这一篇笔记 AT24C02A 作为我们的主角。

9.2 IIC 总线概念



IIC 的总线是由 SDA（串行数据）和 SCL（串行时钟）组成，有从机和主机之分。总线空闲时为高电平，所以 SDA 和 SCL 都必须使用上拉电阻将电平拉上。IIC 总线又有起始信号（START），终止信号（STOP），重复起始信号（REPEAT START）。



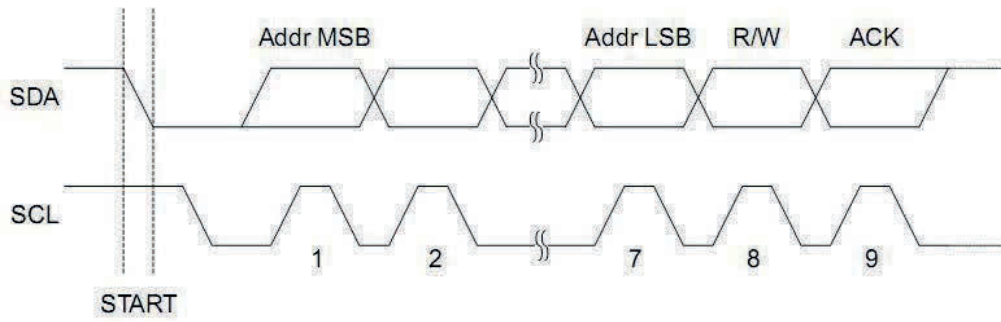
起始信号：SCL 持续高电平的时候，SDA 从高电平变成低电平。

终止信号：SCL 持续高电平的时候，SDA 从低电平变成高电平。

重复起始信号 这个比较难明白，在访问一个设备为了改变访问方向而不写入终止信号，第二次的起始信号称为重复起始信号

9.2 IIC 总线仲裁（寻址）

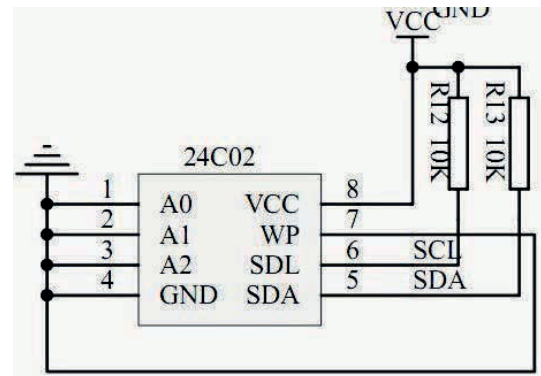
当主机要寻找从机时，会发送寻址，我习惯称为**硬件地址**。该地址包如下：



最高七位为从机地址，而最低位为硬件地址的行为。**最低位逻辑 1 为读从机设备，逻辑 0 为写从机设备**。不同的从机设备都有不同的硬件地址，硬件地址是由硬件厂商设定定的。如 AT24C02A，ATMEL 公司的串行 EEPROM。

2K	1	0	1	0	A ₂	A ₁	A ₀	R/W
	MSB			LSB				
4K	1	0	1	0	A ₂	A ₁	P ₀	R/W
8K	1	0	1	0	A ₂	P ₁	P ₀	R/W
16K	1	0	1	0	P ₂	P ₁	P ₀	R/W

AT24C02A 硬件地址的区分



HJ-2G 的 AT24C02A 的 A0~2 编程

AT24C02A 硬件地址是这样区分的：

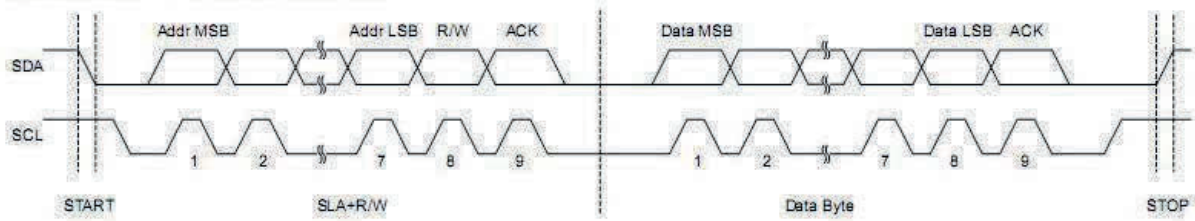
高四位（BIT4~7）为不可编程，AT24C02A 是 1010。低四位（BIT0~3）中的 BIT 1~3，亦即 A₀~A₂ 为可编程地址。HJ-2G 将 A₀~A₂ 引脚连地，所以该 AT24C02 的硬件地址为 1010 000x。一般上该 LSB 位都默认为逻辑 0，很习惯的将 AT24C02 的地址写成 1010 0000，亦即 0xA0。换一句话说，在同一个 IIC 总线上，类似 AT24Cxx 的设备可以连接 8 个。（话扯远了）

硬件地址比较常用的写法，如 SLA+R 或者 SLA+W。该字表示 Slave Address + Read，Slave Address + Write。

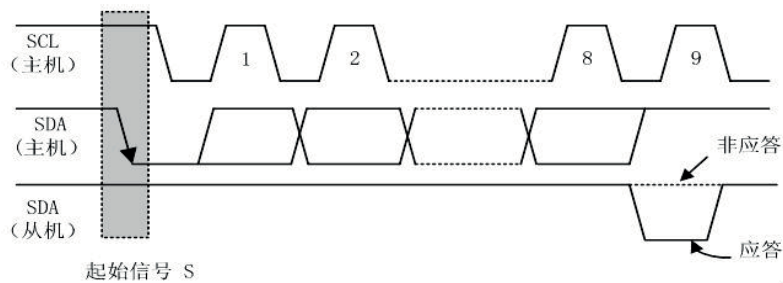
9.3 IIC 总线 • 应答信号

IIC 总线数据都是一次传送一个字节，MSB 先传输。而传输的次序为 硬件地址==>数据字节

Figure 81. 典型的数据传送



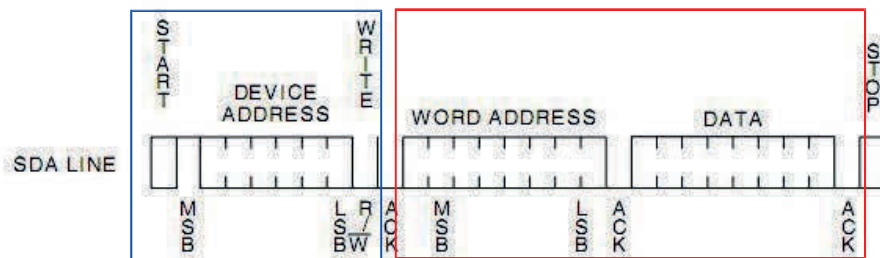
如上图，在数据传送的开始之间会主机会先给总线一个起始信号，然后广播硬件地址。如果该硬件地址与某从机设备吻合，那么该某从机设备必须回应（Response），而该回应就是应答信号(ACK)。



如上图，当主机给从机发送硬件地址，从机吻合后或者从机从主机接收数据字节后，从机必须回应主机，应答为“收到了”，非应答为“沉默，没收到，没反应，或者失败”。在总线的行为上，从机给主机的应答是拉低 SDA，而不应答是对 SDA 不作反应。换一句话说，每当主机发送一字节数据（管他是什么数据），从机就必须回应一次。当主机对从机一次性的访问结束后，就会发送终止信号，告诉从机“完了”。

9.3 IIC 总线 • 写硬件地址，写数据

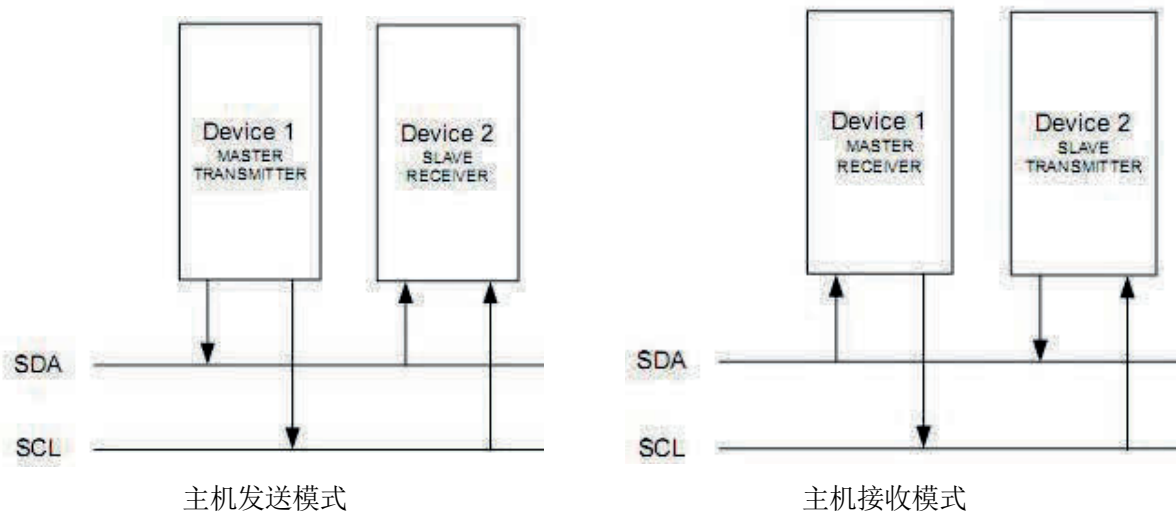
从上图“典型的数据”中，我们知道典型的 IIC 数据传送，是先写硬件地址，再来是写数据。那么写硬件地址的定义是什么？写硬件地址的定义是，主机在总线上寻址从机。而写数据的定义是主机对从机进行访问。



如上图 AT24C02A 的写操作中（写不管该时序是什么意思）。蓝色框图中为写硬件地址，而红色框图为写数据。该概念很重要，因为 AVR 的 IIC 资源会将 IIC 分成四种模式。AVR 的 IIC 状态寄存器有写硬件地址和写数据的区分。

9.4 四种模式

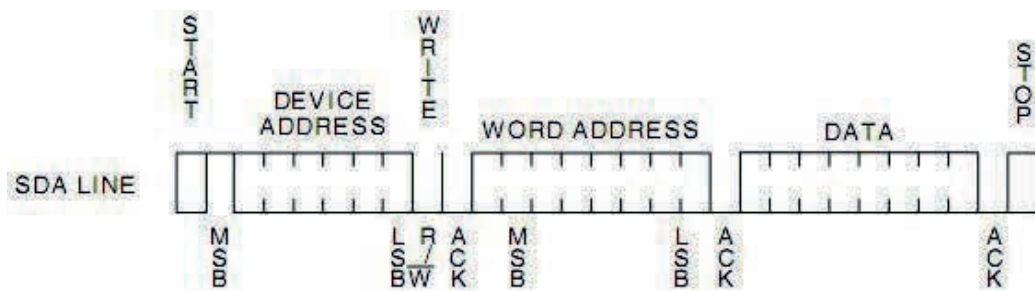
AVR 的 IIC 模块有四种的操作模式：**主机发送**，**主机接收**，从机发送，从机接收。而前两个就是我们要学习的模式。



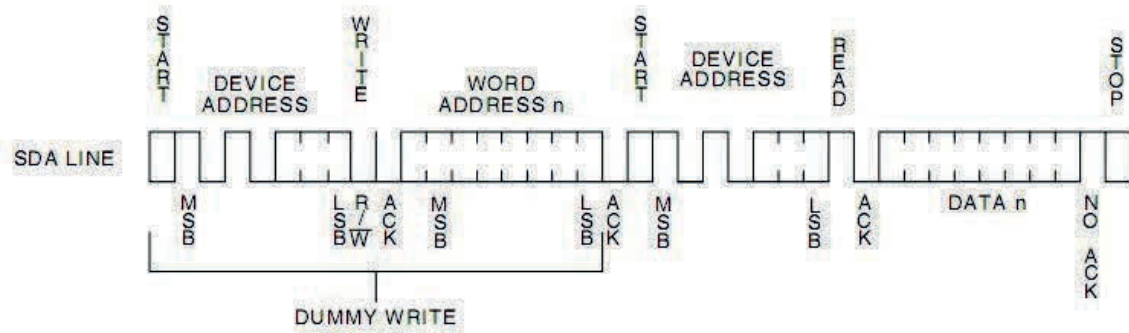
无论是主机接收还是主机发送，**SCL 总线的控制都是主机所拥有**。

9.5 熟悉 起始信号，重复起始信号，应答信号，终止信号

在还没有接触寄存器之前，再来加深一些 IIC 典型的传输方式。我依然以 ATMEL 公司的 AT24C02A 为例。我们假设一个情况，AVR 单片机是主机而 AT24C02A 是从机的关系。（无视 SCL）



上图，是 AT24C02A 的**写操作**，在写操作开始之前，先是起始信号，然后主机广播硬件地址（从机地址|写硬件地址），由于是**写操作的行为**，硬件地址的 **LSB 必须为逻辑 0**，表示该硬件地址是写的行为，然后等待从机的应答。当主机接收到从机的应答信号后，会进行写数据，主机输送页地址，等待从机的应答，然后发送数据，等待应答。主机最后发一个终止信号表示放弃对总线的拥有，或者表示一次性的操作已经结束。



再来也是一个很经典的例子。上图是 AT24C02A 的读操作，图上的时序，写硬件地址行为的方向明显有改变。硬件地址先是写行为，然后硬件地址改变成读行为。

首先依然是主机给个起始信号，然后写硬件地址，刚开始主机为了选择页地址，主机把写硬件地址的行为设置位“写”，然后等待从机的应答，再然后输送页地址，等待从从机的应答。

重点来了：

主机为了不~~放弃~~一次性的操作，而它又必须改变写硬件地址的方向，这时主机发一个重复起始信号，然后进行写硬件地址操作，注意该硬件地址的行为已经是“读”了（硬件地址的 LSB 为逻辑 1）。从机接收后回应主机，然后将本身的数据输送给主机。最后呢，主机从从机接收数据后，“它已经满意了”，然后无视应答强制发送一个终止信号。

在这里有几点重点需要注意：

第一，主机可以改变写硬件地址的方向，而无需发次当前对从机的操作，只要发重复起始信号几个。

第二，当主机为接收模式时（写硬件地址为读行为），SCL 线依然是主机拥有着（这表明主机依然是主机），主机接收一个字节的数据后，可以根据自己的意愿，要继续接收数据呢？还是不要继续接收数据呢？如果是后者，主机可以无视应答，强制结束该操作。

9.6 关于 IIC | TWI 的寄存器

IIC 有关的寄存器，虽然不算多，但是由于模式众多，尤其是状态寄存器更是猥琐。所以，在这里仅焦距在同一个方向讨论而已。

第一：TWI 控制寄存器 TWCR

Bit	7	6	5	4	3	2	1	0	
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE	TWCR
读 / 写	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
初始值	0	0	0	0	0	0	0	0	

TWINT，亦即中断标志位。该位为逻辑 1 时表示上一次的~~操作~~已经结束或者 TWI 模块结束工作。换句话说，TWINT 位就像 TWI 功能的“开始键”，要 TWI 工作该位必须清零。而清零的方式就是为它写 1。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/568116116016006120>