## I2C协议

### 1 介绍

#### 1.1 概述

I2C总线具有广泛应用，多功能I2C总线用于各种控制体系结构，如系统管理总线(SMBus)、电源管理总线(PMBus)等。

I3C提供了与I2C的向后兼容性，提高速度和降低功耗。

#### 1.2 I2C协议特性

* 只需要两条总线线路；串行数据线(SDA)和串行时钟线(SCL)；
* 连接到总线的每个设备都是由唯一地址进行寻址；控制器可以作为数据发送方和接收方；
* 真正的多控制器总线，包括碰撞检测和仲裁；
* 串行、双向数据传输在标准模式下可以达到100kb；在快速模式下高达400kb；快速增强模式下高达1Mb；高速模式达到3.4Mb；
* 串行、单向数据传输在超快模式下高达5Mb；
* 片上滤波以保持数据完整性；
* 连接到同一总线的IC数量仅受最大总线电容的限制；

### 2 标准模式、快速模式和快速增强模式

串行数据(SDA)和串行时钟(SCL)在连接到总线的设备之间传递信息。每个设备都有一个唯一的地址并且可以根据设备的功能作为发送方或接收方。LCD驱动器可能只是一个接收器，而存储器可以同时接收和传输数据。除了发送方和接收方，当执行数据传输时，设备也可以被视为控制器或目标设备。控制器是在总线上启动数据传输并生成时钟信号的设备。任何被寻址的设备都被认为是目标设备。

* 发送方：向总线发送数据的设备；
* 接收方：从总线接收数据的设备；
* 控制器：发起传输、产生时钟信号并终止传输的设备；
* 目标设备：由控制器寻址的设备；
* 多个控制器：多个控制器可以同时尝试控制总线；
* 仲裁：以确保如果多个控制器同时试图控制总线，则只允许一个控制器这样做；
* 同步：同步两个或多个设备的时钟信号；

I2C总线是一种多控制器总线。这意味着可以将多个能够控制总线的设备连接到总线上。由于控制器通常是微控制器。

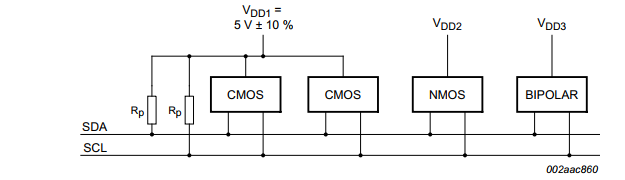
将多个微控制器连接到I2C总线意味着多个控制器可以同时尝试发起数据传输。通过仲裁程序来避免这种情况。此过程依赖于所有连接I2C总线的I2C接口。

如果两个或多个控制器试图将信息放到总线上，当另一个控制器产生低电平时，第一个产生高电平的控制器将失去仲裁。

I2C总线上时钟信号的产生始终由控制器负责；每个控制器在总线上传输数据时产生自己的时钟信号。来自控制器的总线时钟信号只有在慢速的目标设备控制时钟信号或被另一个控制器仲裁时才能改变。

#### 2.1 SDA和SCL信号

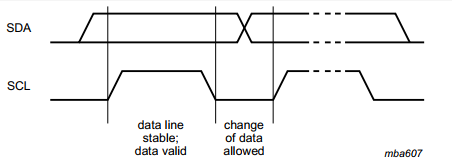
SDA和SCL都是双向线路，通过上拉电阻连接到正电源电压。当总线空闲时，两条线路都是高电平。连接到总线的设备的输出级必须有一个开路漏极或开路集电极来执行有线与功能。I2C总线的标准模式传输速率可达100kb；快速模式传输速率可达400kb；快速增强模式传输速率可达1Mb；高速模式传输速率可达3.4 Mb。总线电容限制连接到总线的接口数量。对于单个控制器，如果总线上没有设备，控制器的SCL输出可以是推挽式驱动器设计。



由于可以连接到I2C总线的不同设备(CMOS、NMOS、双极)，逻辑高电平和低电平不是固定的，而是取决于VDD。参考设置为VDD的30%和70%；VIL为0.3VDD, VIH为0.7VDD。一些传统设备的输入电平固定在VIL(1.5 V)和VIH(3.0 V)。

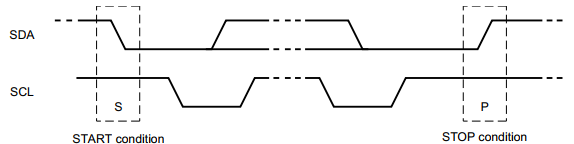
#### 2.2 数据有效性

在时钟的高电平时间段内，SDA上的数据必须保持稳定。只有当SCL线上的时钟信号为低电平时，SDA上的数据才会改变。每传输一个数据位，产生一个时钟脉冲。



#### 2.3 开始和停止条件

所有数据传输都以START位开始，并由STOP位终止。当SCL为高电平时，SDA上由高电平转换到低电平为START位。当SCL为高电平时，SDA上低电平到高电平的转换为STOP位。

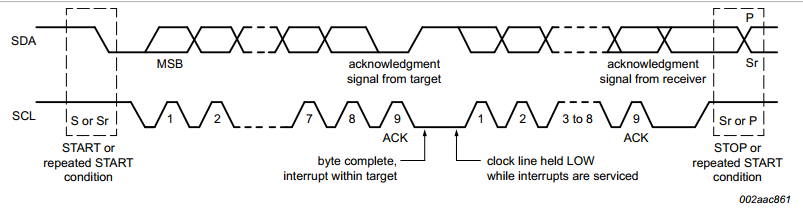


START位和STOP位总是由控制器生成。在START位之后，总线处于工作忙的状态。在STOP位后的某一时刻，总线处于空闲状态。如果生成了重复的START 位而不是STOP位，则总线将保持忙状态。

如果连接到总线的设备包含必要的接口硬件，则很容易通过这些检测START和STOP位。没有这种接口的微控制器必须在每个时钟周期对SDA线进行至少两次采样。

#### 2.4 数据格式

SDA上的每个字节必须是8位。每次传输可以传输的字节数不受限制。每个字节后面必须跟着一个确认位。数据首先传输最高有效位(MSB)。如果目标在执行其他功能之前不能接收或传输另一个完整字节的数据，可以保持时钟线SCL低电平以使控制器进入等待状态。然后，当目标准备好接收另一个字节的数据并释放时钟线SCL时，数据传输继续进行。



#### 2.5 确认(ACK)和不确认(NACK)

确认发生在每个字节之后。确认位允许接收方向发送方发出信号，表明字节已成功接收并且可以发送另一个字节。控制器产生所有时钟脉冲，包括确认第九个时钟脉冲。确认信号为发送方在确认时钟脉冲期间释放SDA线，接收方可以将SDA拉低并在该时钟脉冲的高期间保持稳定的低电平。

当SDA在第九个时钟脉冲期间保持高电平时，被定义为不确认信号。然后控制器可以生成一个STOP位来终止传输，或者一个重复的START为来开始一个新的传输。

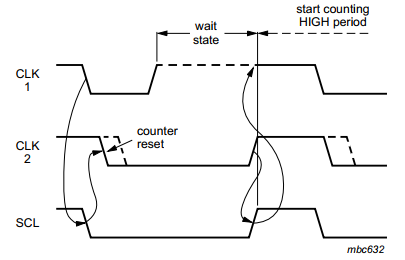
**不确认产生原因**：

* 总线上没有相应地址的接收者，因此没有设备响应确认；
* 接收方无法接收或发送，因为正在执行一些其它功能，还没有准备好开始与控制器通信；
* 在传输过程中，接收方得到无法识别的数据或命令；
* 在传输过程中，接收方不能再接收任何数据字节；

#### 2.6 时钟同步

两个控制器可能同时在一个空闲总线上开始传输，必须有一种方法来决定哪一个控制器控制总线进行传输。这是通过时钟同步和仲裁来完成的。单控制器系统不需要时钟同步和仲裁。

时钟同步采用将所有I2C接口SCL时钟信号的进行与操作。这意味着SCL线上的高到低转换导致相关控制器开始计数他们的低周期，一旦控制器时钟已经达到LOW，它将SCL线保持在那个状态，直到时钟达到HIGH状态。然而，如果另一个时钟仍然在它的LOW周期内，这个时钟的LOW到HIGH转换可能不会改变SCL线的状态。因此，SCL线被具有最长LOW周期的控制器保持为LOW。LOW周期较短的控制器在此时间进入HIGH等待状态。



当所有相关控制器都已计算出它们的LOW周期时，时钟线被释放并进入HIGH周期。然后，控制器时钟和SCL线的状态之间没有差异，所有控制器开始计算它们的HIGH周期。第一个完成HIGH周期的控制器再次拉低SCL线。

这样就生成了一个同步的SCL时钟，其LOW周期由时钟LOW周期最长的控制器决定，HIGH周期由时钟HIGH周期最短的控制器决定。

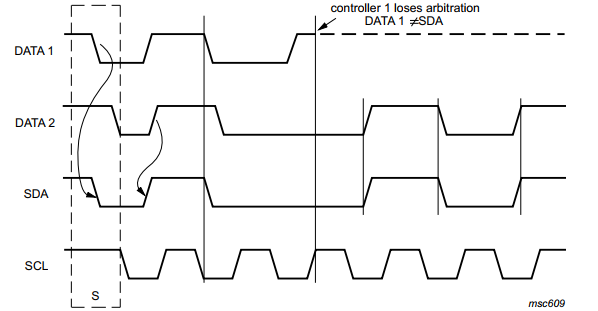
#### 2.7 仲裁

与同步一样，只有在系统中使用了多个控制器时才需要仲裁。目标设备不参与仲裁过程。控制器只有在总线空闲时才可以开始传输。两个控制器可以在START位的最小保持时间内生成START位，从而在总线上产生有效的START位。然后需要仲裁来确定哪个控制器进行数据传输。

仲裁是一位一位进行的。在每一个比特中，当SCL为高电平时，每个控制器都会检查SDA电平是否与它所发送的匹配。这个过程可能需要很多比特。两个控制器实际上可以毫无错误地完成整个事务，只要传输是相同的。当控制器第一次尝试发送HIGH，但检测到SDA为低电平时，控制器知道已经失去了仲裁并关闭其SDA输出。另一个控制器继续进行数据传输。

在仲裁过程中不会丢失任何信息。等其它控制器传输数据完后，丢失仲裁的控制器可以生成时钟信号并且在总线空闲时重新启动传输。

如果一个控制器可以作为目标设备，并且在寻址阶段失去了仲裁，那么获胜的控制器可能正在试图寻址它。因此，丢失控制器必须立即切换到目标设备模式。



如图显示了两个控制器的仲裁过程。根据有多少控制器连接到总线，可能涉及更多的控制器。当产生DATA1的控制器的内部数据电平与SDA线上的实际电平之间存在差异时，DATA1输出将被关闭。这不会影响获胜控制器发起的数据传输。

由于I2C总线的控制完全取决于相互竞争的控制器发送的地址和数据，因此总线上没有中央控制器，也没有任何优先顺序。

当一个控制器发送重复的START或STOP条件，而另一个控制器仍在发送数据时，仲裁过程仍在进行，这是未定义行为。

#### 2.8 时钟偏斜

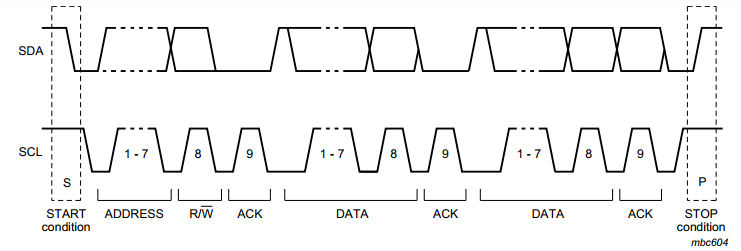
时钟通过将SCL设置为低电平来暂停传输。传输不能继续，直到SCL为高电平。时钟延长是可选的，事实上，大多数目标设备不能驱动SCL引脚，因此无法延长时钟。在字节级别上，设备可能能够以较快的速度接收数据字节，但需要更多的时间来存储接收到的字节或准备另一个字节来传输。然后，目标可以在接收和确认位后将SCL设置为低电平，迫使控制器进入等待状态，直到目标准备好后进行下一个字节传输。

在位级上，微控制器之类的设备可以通过延长每个时钟低电平周期来降低总线时钟的速度。任何控制器的速度都适应于该设备的内部运行速率。

在高速模式下，此特性只能在字节级别上使用。

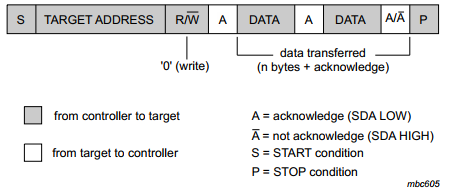
#### 2.9 目标地址和读写位

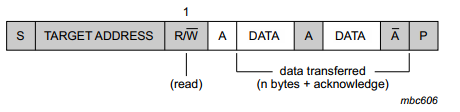
在START位之后，发送目标地址。该地址长7位，后面跟着第8位，表示进行读写操作。数据传输总是由控制器生成的STOP位。然而，如果一个控制器仍然希望在总线上通信，可以生成一个重复的START位为并在不首先生成STOP条件的情况下寻址另一个目标。

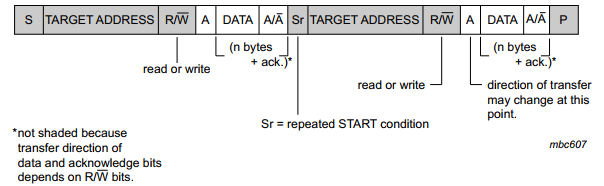


数据传输格式：

* 控制器(发送方)传输到目标设备(接收方)。传输方向没有改变。目标设备确认每个字节；
* 控制器在第一个字节后立即读取目标。在第一次确认时，控制器(发送方)变成控制器(接收方)，目标(接收方)变成目标(发送方)。第一个确认仍然由目标设备生成。控制器生成后续的确认。STOP位由控制器生成，在STOP位之前发送一个NACK位；
* 组合格式：在传输中的方向改变期间，START位和目标地址都是重复的，但是读写位改变。如果控制器(接收方)发送了一个重复的START位，它会在重复的START位之前发送一个NACK位；





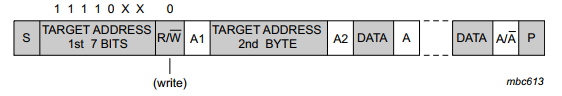


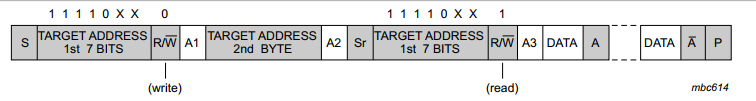
#### 2.10 10位地址

10位寻址扩展了可用地址的数量。具有7位和10位地址的设备可以连接到相同的I2C总线并且7位和10位寻址可以在所有总线工作模式中使用。

10位目标地址由START位或重复START位后的前两个字节组成。

第一个字节的前7位为组合11110xx，其中后两位为10位地址的两个最高有效位；第一个字节的第8位是决定消息方向的R/W位。虽然保留地址位1111xxx有8种可能的组合，但只有4种组合11110xx用于10位寻址。其余四个组合1111 1XX保留给未来的I2C总线扩展。





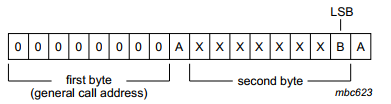
具有10位寻址的目标设备与具有7位寻址的目标设备以相同的方式工作。硬件控制器可以在“通用调用”之后传输它们的10位地址。在这种情况下，“通用调用”地址字节后面跟着两个连续的字节，其中包含控制器-发送器的10位地址。其中第一个DATA字节包含控制器地址的8个最低有效位。START字节0000 0001 (01h)可以以与7位寻址相同的方式在10位寻址之前。

#### 2.11 保留地址

本地系统内地址的分配由系统架构师负责，他们必须考虑总线上使用的设备以及未来与其他传统i2c总线的任何交互。例如，一个有7个用户可分配地址引脚的设备允许分配所有128个地址。如果知道保留地址永远不会用于其预期目的，则保留地址可用于目标地址。

#### 2.12 广播寻址

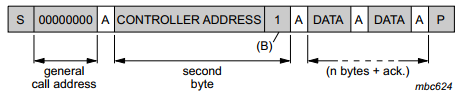
广播寻址用于同时寻址连接到I2C总线的设备。如果一个设备不需要任何广播的数据，可以通过发出不确认来忽略此地址。如果一个设备确实需要来自广播的数据，会确认这个地址并作为目标接收方。如果一个或多个设备响应，控制器实际上不知道有多少设备被确认。第二个字节和后面的字节由每个能够处理此数据的目标接收器确认。不能处理其中一个字节的目标必须通过不确认来忽略。同样，如果一个或多个目标确认，不确认将不会被控制器看到。



当比特B为0时，第二个字节的定义如下：

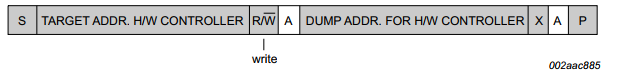
* 8’h6：硬件复位和写入目标地址的可编程部分；在接收到这个2字节序列后，所有设备广播地址重置并接收其地址的可编程部分。必须采取预防措施，以确保设备不会拉低SDA或SCL信号电平，因为这会阻塞总线；
* 8’h4：由硬件写入地址的可编程部分；
* 8’h0：该代码不允许用作第二个字节；

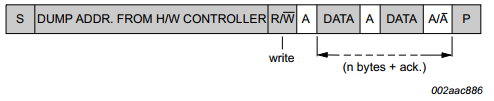
当位B为1时，2字节序列是“硬件广播”。这意味着序列是由硬件控制器设备传输的，例如键盘扫描仪，可以被编程来传输所需的目标地址。由于硬件控制器事先不知道消息必须传输到哪个设备，所以只能生成这个硬件广播和自己的地址——向系统标识自己。



第二字节中剩下的7位包含硬件控制器的地址。该地址由连接到总线上的智能设备(例如微控制器)识别，然后该设备接受来自硬件控制器的信息。如果硬件控制器也可以作为目标器，则目标器地址与控制器地址相同。

在一些系统中，另一种方法是在系统复位后将硬件控制器(发送方)设置为目标设备(接收方)。通过这种方式，系统配置控制器可以告诉硬件控制器(发送方)必须将数据发送到哪个地址。在此编程过程之后，硬件控制器保持在控制器(发送方)模式。





#### 2.13 硬件复位

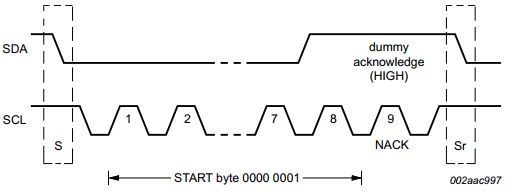
在广播 (0000 0000)之后，发送0000 0110 (06h)作为第二个字节会导致软件重置。此特性是可选的，并不是所有设备都响应此命令。在接收到这个2字节序列后，所有可以响应的设备重置并接收其地址的可编程部分。必须采取预防措施，以确保设备在此过程中不会拉下SDA或SCL线，因为这些低电平会阻塞总线。

#### 2.14 START位类型

微控制器可以通过两种方式连接到I2C总线。具有I2C总线接口的微控制器可以被编程为仅被来自总线的请求中断。当设备没有这样的接口时，必须通过软件不断地监视总线。显然，微控制器监视或轮询总线的次数越多，执行预期功能的时间就越少。因此，在快速的硬件设备和相对较慢的依赖软件轮询的微控制器之间存在速度差异。在这种情况下，数据传输之前可以有一个比正常情况长得多的启动过程。

**启动过程包括**：

1. START位；
2. 起始字节(0000 0001)；
3. 一个确认时钟脉冲(ACK)；
4. 重复START位；



在需要总线访问的控制器发送START条件位后，发送START字节(0000 0001)。因此，另一个微控制器可以以较低的采样率对SDA行进行采样，直到检测到START字节中的七个零之一。在SDA线路上检测到这个低电平后，微控制器可以切换到更高的采样率，以找到重复的START条件位，然后用于同步。

硬件接收器在接收到重复的START条件位时重置，因此忽略START字节。

在START字节之后生成一个与确认相关的时钟脉冲。这只是为了符合总线上使用的字节处理格式。任何设备都不允许确认START字节。

#### 2.15 清除总线

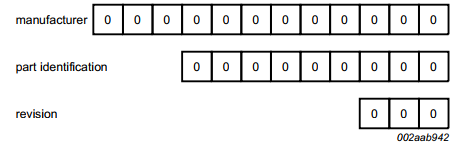
在时钟(SCL)一直处于低电平时，如果I2C设备有硬件复位输入，优先使用硬件复位信号重置总线。如果I2C设备没有硬件复位输入，则循环上电以激活内部电源重启(POR)电路。

如果数据线(SDA)一直处于低电平，控制器应该发送9个时钟脉冲。保持总线低电平的设备应该在这九个时钟内的某个时间释放。如果没有，则使用硬件复位或循环电源清除总线。

#### 2.16 设备ID

设备ID字段是一个可选的3字节只读数据：

* 12位每个制造商唯一标识；
* 由制造商指定的带有零件标识的9位；
* 由制造商指定的三个修正位；



**读取设备ID**：

1. START位；
2. 控制器发送设备ID总线地址，后面的读写位设置为0；
3. 控制器发送识别的目标设备的总线地址；
4. 控制器发送一个重复START位；
5. 控制器发送设备ID总线地址，后面的读写位设置为1；
6. 设备ID读取可以完成；
7. 控制器通过NACK最后一个字节结束读取序列，从而重置目标设备状态机并允许控制器发送STOP条件。注意:任何时候都可以通过发送NACK停止设备ID的读取；

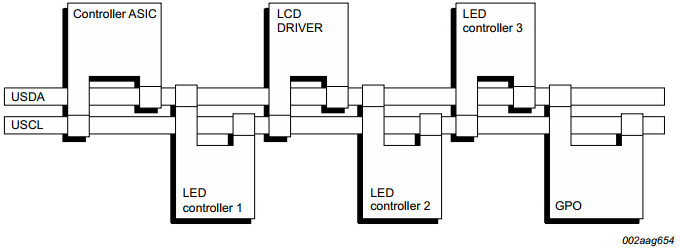
### 3 高速I2C协议

高速I2C总线是一种2线推挽串行总线，单向传输速度高达5Mb。高速I2C总线协议基于标准I2C总线协议，该协议由START、目标地址、命令位、第九个时钟和STOP位组成。命令位仅为写，第九个时钟上的数据位被驱动为高电平，由于总线的单向性质，忽略ACK周期。高速I2C总线由串行时钟(USCL)和串行数据(USDA)组成。

目标设备包含唯一的地址，并且仅作为接收器。LED驱动器可能只是接收器，可以支持高速I2C协议。

由于高速I2C总线使用推挽式驱动器，不具备多控制器功能。在高速I2C总线中，控制器是在总线上发起数据传输并生成时钟信号的唯一设备。处理的所有其他设备都被认为是目标设备。

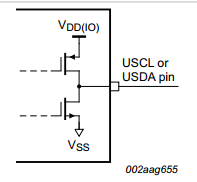
* 发送方：向总线发送数据的设备；
* 接收方：从总线接收数据的设备；
* 控制器：启动传输、产生时钟信号并终止传输的设备；
* 目标设备：控制器寻址的设备；



由于推挽输出上的总线争用，不允许将多个控制器连接到总线。如果系统中需要一个额外的控制器，必须与另一个控制器完全隔离，因为总线上一次只允许一个控制器。在总线上产生时钟信号始终由控制器设备负责，即控制器在总线上传输数据时产生时钟信号。来自控制器的总线时钟信号不能被目标设备的时钟所改变，仲裁和时钟同步过程在高速I2C总线中不存在。

#### 3.1 USDA和USCL信号

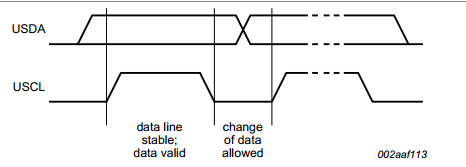
USDA和USCL都是单向线，具有推拉输出。当总线空闲时，两条线都被输出级的上层晶体管拉高。总线上的数据可以在高速模式下以高达5000Kb的速率传输。连接到总线的接口数量受总线负载、电缆端、连接器和反射的限制。



由于可以连接到总线的各种不同技术设备，逻辑低电平和高电平不是固定的，而是取决于VDD。输入参考水平设置为VDD的30%和70%；VIL为0.3VDD, VIH为0.7VDD。

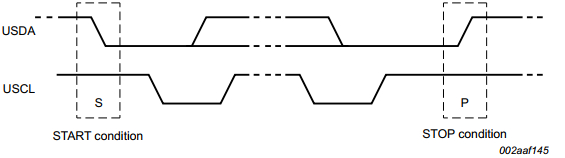
#### 3.2 数据有效

USDA线上的数据必须在时钟的HIGH时段保持稳定。只有当USCL线上的时钟信号为低电平时，数据线的电平才能改变。每传输一个数据位就产生一个时钟脉冲。



#### 3.3 开始和停止位

当总线处于空闲状态时，数据和时钟线都保持高电平状态。所有传输都以START位开始并由STOP位终止。当USCL为高电平，USDA上从高到低的转换USCL为START位。当USCL为高电平时，USDA上低电平到高电平的转换为STOP位。

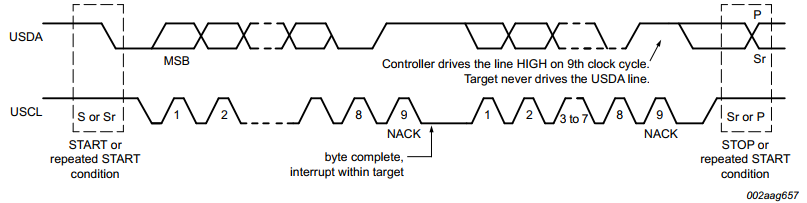


START和STOP位总是由控制器生成。在START位之后，总线处于工作状态。在STOP位之后，总线处于空闲状态。如果生成了重复的START位而不是STOP位，则总线将保持忙状态。在这方面，START和重复START位在功能上是相同的。

如果连接到总线的设备包含必要的接口硬件，则很容易检测START和STOP位。没有这种接口的微控制器必须在每个时钟周期对USDA线进行至少两次采样。

#### 3.4 数据格式

USDA上的每个字节必须是8位。每次传输可以传输的字节数不受限制。控制器在确认周期的每个字节后驱动USDA为高电平。首先传输数据最高有效位。如果目标设备不能接收数据，则不允许目标设备将USCL设置为低电平来使控制器进入等待状态。

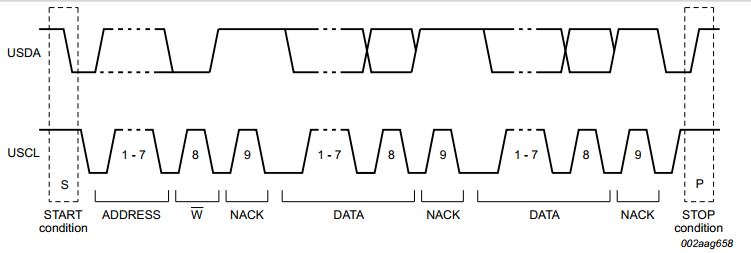


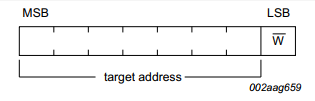
#### 3.5 确认(ACK)和不确认(NACK)

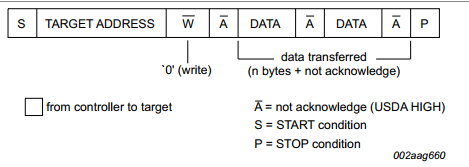
由于目标设备不能响应第九个时钟周期，因此不需要ACK和NACK。但是，时钟周期在高速I2C协议中被保留以兼容标准I2C总线协议。ACK和NACK也被称为第九个时钟周期。控制器产生所有时钟脉冲，包括第9个时钟脉冲。第九个数据位总是被驱动为高电平。目标设备在任何时候都不允许驱动SDA。

#### 3.6 目标地址和读写位

在START位之后，发送目标地址。该地址长7位，后面跟着第8位，这是读写；然而不允许读操作(START字节除外)，因为通信是单向的。数据传输总是由控制器生成的STOP位终止。然而，如果一个控制器仍然希望在总线上通信，可以生成一个重复的START位并在不首先生成STOP条件的情况下寻址另一个目标。





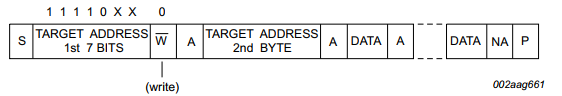


#### 3.7 10位地址

10位寻址扩展了设备地址的数量。7位和10位地址的设备可以连接到相同的I2C总线，7位和10位寻址可以在所有总线速度模式中使用。10位目标地址由START位后的前两个字节组成。

第一个字节的前7位是组合11110xx，其中最后两位(XX)是10位地址的两个最高有效位；第一个字节的第8位是读写位。

虽然保留地址位1111xxx有8种可能的组合，但只有4种组合11110xx用于10位寻址。其余四个组合1111 1XX保留给未来的I2C总线。



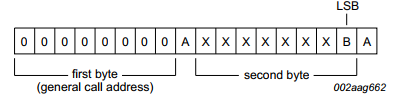
START字节8’h1以与7位寻址相同的方式在10位寻址之前。

#### 3.8 保留地址

高速I2C总线与其他I2C速度模式总线具有不同的物理层。因此可用的目标地址范围是不同的。

#### 3.9 广播寻址

广播寻址用于同时寻址连接到I2C总线的每个设备。但是，如果一个设备不需要广播的数据，可以忽略这个地址。如果一个设备确实需要广播的数据，可以作为目标设备。控制器实际上并不知道有多少设备响应这个广播。第二个字节和后面的字节由每个响应广播的目标设备接收。不能处理其中一个字节的目标设备必须忽略该广播。



当比特B为0时，第二个字节的定义如下：

* 8’h06：硬件复位和写入地址可编程部分。在接收到这个2字节序列后，所有可以响应广播地址的设备复位并接收其地址的可编程部分；
* 8’h04：硬件写入目标地址可编程部分。行为如上，但设备不复位；
* 8’h00：该代码不允许用作第二个字节；

当位B为1时，2字节序列将被忽略。

#### 3.10 软件复位

在广播8’h00之后，发送8’h06作为第二个字节会导致软件复位。此特性是可选的，并不是所有设备都响应此命令。在接收到这个2字节序列后，所有可以响应广播地址的设备复位并接收其地址的可编程部分。

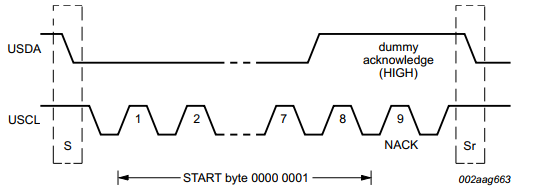
#### 3.11 START字节

微控制器可以通过两种方式连接到I2C总线。具有片上硬件I2C总线接口的微控制器可以被编程为仅被来自总线的请求中断。当设备没有这样的接口时，必须通过软件不断地监视总线。显然，微控制器监视或轮询总线的次数越多，执行预期功能的时间就越少。

因此，在快速的硬件设备和相对较慢的依赖软件轮询的微控制器之间存在速度差异。在这种情况下，数据传输之前可以有一个比正常情况长得多的启动过程。

启动过程包括：

* START位；
* 起始字节8’h01；
* 一个不确认时钟脉冲(NACK)；
* 重复START位；



在需要总线访问的控制器发送START条件S后，发送START字节(0000 0001)。因此，另一个微控制器可以以较低的采样率对USDA行进行采样，直到检测到START字节中的七个零中的一个。在USDA线上检测到这个低电平后，微控制器可以切换到更高的采样率，以找到重复的START位，然后用于同步。硬件在接收到重复的START位时重置，因此忽略START字节。在START字节之后生成一个与确认相关的时钟脉冲。这只是为了符合总线上使用的字节处理格式。任何设备都不允许确认START字节。

#### 3.12 无响应目标复位

在目标设备无响应的情况下，优先使用软件复位命令或硬件复位信号重置目标设备。如果目标不支持这些特性，则循环给设备上电以激活电源复位(POR)电路。

高速I2C协议不支持设备ID字段。

### 4 I2C总线的其他用途

I2C总线被用作多种系统架构的通信协议。除了基本I2C规范之外，这些体系结构还添加了命令集和特定于应用程序的扩展。一般来说，简单的i2c总线设备(如I/O扩展器)可以用于这些体系结构中的任何一个，因为协议和物理接口是相同的。

#### 4.1 CBUS兼容性

CBUS接收器可以连接到标准模式i2c总线。除此之外，CBUS的DLEN必须被连接并省略确认位。通常，I2C传输是8位字节的序列；CBUS兼容的设备有不同的格式。

在混合总线结构中，I2C总线设备必须不响应CBUS消息。因此，预留了一个特殊CBUS地址(8’b0000\_001X)。传输完CBUS地址后，DLEN信号被激活并发送CBUS格式的传输。在STOP位之后，所有设备再次准备好接受数据。

控制器(发送方)在发送CBUS地址后可以发送CBUS格式消息。传输由所有设备都能识别的STOP位结束。

注:如果已知CBUS配置，且未预见到扩展使用CBUS兼容设备，则允许设计人员根据所使用设备的特定要求调整保持时间。

#### 4.2 SMBus系统管理总线

SMBus使用I2C硬件和I2C硬件寻址，但增加了用于构建特殊系统的二级软件。特别是，规范包括一个地址解析协议，可以进行动态地址分配。硬件和软件的动态重新配置允许总线设备热插拔并立即使用，而无需重新启动系统。这些设备被自动识别，并被分配唯一的地址。这一优势产生了即插即用的用户界面。在这两个协议中，系统主机和系统中所有其他具有控制器或目标的名称和功能的设备之间有一个非常有用的区别。

SMBus目前在大多数PC中用作系统管理总线。由英特尔等公司开发，修改了I2C的一些电气和软件特性，以更好地适应便携式设备。

##### 4.2.1 I2C / SMBus兼容

SMBus和I2C协议基本相同。一个SMBus控制器能够控制I2C设备，反之亦然。SMBus时钟频率在10KHZ到100KHZ，而I2C则取决于具体模式。这意味着运行在10KHZ以下的I2C总线不符合SMBus标准。

逻辑电平也略有不同。SMBus协议中逻辑低电平约为0.8 V和高电平约为2.1 V，而I2C为VDD的 30%和70%。

##### 4.2.2 超时特性

SMBus有一个超时特性。如果通信时间过长，可以复位设备。这解释了最低时钟频率为10KHZ以防止总线锁定。I2C可以称为“DC”总线，这意味着当控制器访问目标设备时，目标设备在执行某些例行程序时延长了控制器时钟。这将通知控制器，目标正忙，但不想失去通信。目标设备在任务完成后将继续传输。i2c总线协议中没有限制该延迟的时间，而对于SMBus系统，将被限制为35毫秒。

SMBus协议假设如果某次传输花了太长时间，那么这意味着总线上有问题，所有设备必须进行复位。目标设备不允许长时间保持时钟为低电平。

##### 4.2.3 SMBus 1.0和SMBus 2.0的区别

SMBus规范定义了两类电气特性：低功率和高功率。第一类最初是在SMBus 1.0和1.1规范中定义的，设计时主要考虑的是智能电池，但也可以与其他低功耗设备一起使用。

2.0版本引入了替代更高功率的电气特性集。当需要更高的驱动能力时，该类适用于使用，例如PCI外接卡上的SMBus设备。设备可以由总线VDD或另一种电源供电(如智能电池)。只要遵守其类别的SMBus电气规范，就可以相互操作。

#### 4.3 PMBus电源管理总线

PMBus是通过SMBus在电源转换器和系统主机之间进行通信的标准方式，以提供对电源转换器的更智能控制。PMBus规范定义了一组标准的设备命令，以便来自多个源的设备能够相同地工作。PMBus设备使用SMBus 1.1版本和扩展进行传输。

### 5 总线速度

最初，I2C总线仅限于100Kb的操作。随着时间的推移，规范中增加其它速度模式，现在有五个传输速度模式。标准模式、快速模式、快速增强模式、高速模式和超速模式设备是向下兼容的，即任何设备都可以在较低的总线速度下运行。超速模式设备与以前的版本不兼容，因为总线是单向的。

**双向总线**：

* 标准模式：码速可达100Kb；
* 快速模式：码速可达400Kb；
* 快速增强模式：码速可达1Mb；
* 高速模式：码速可达3.4 Mb；

**单向总线**：

* 超速模式：比特率高达5Mb；

#### 5.1 快速模式

快速模式可以使用高达400Kb的速度接收和传输。最低要求是可以同步400 kbit/s传输；然后，可以延长SCL信号的低电平周期，以减缓传输。SDA和SCL上的协议、格式、逻辑电平和最大容性负载与标准模式I2C总线规范相同。快速模式设备向下兼容，可以在0到100Kb总线系统中与标准模式设备通信。然而，由于标准模式器件不能向上兼容；不应该被合并到快速模式I2C总线系统中。

快速模式I2C总线规范具有以下附加特性：

* 最大比特率增加到400Kb；
* 调整了SDA和SCL信号的时序。不需要与其他总线系统(如CBUS)兼容；
* 快速模式器件的输入在SDA和SCL输入处包含峰值抑制和施密特触发器；
* 快速模式器件的输出缓冲区包含SDA和SCL信号下降沿的斜率控制；
* 如果快速模式设备的电源被切断，SDA和SCL引脚必须是开漏的，这样就不会阻塞总线线路；

连接到总线线路的外部上拉设备必须适应快速模式I2C总线更短的最大允许上升时间。对于负载高达200PF的母线，每条母线的上拉装置可以是一个电阻；对于200pf到400pf之间的总线负载，上拉装置可以是电流源(最大3mA)或开关电阻电路。

#### 5.2 快速增强模式

快速增强模式器件提高了i2c总线传输速度和总线电容。设备可以高达1Mb的比特率传输信息，但在混合速度总线系统中，仍然完全向下兼容快速或标准模式设备的双向通信。与快速或标准模式系统保持相同的串行总线协议和数据格式。设备还提供比快速或标准模式设备更大的驱动电流，允许它们驱动更长或负载更重的总线。

快速增强模式部件中的驱动器足够强大，可以满足时序规范，与标准模式部件具有相同的400 pF负载。为了向后兼容标准模式，还能容忍标准模式部件的1μs上升时间。

#### 5.3 高速模式

高速模式设备明显提高了i2c总线传输速度。高速模式设备可以高达3.4 Mb的比特率传输信息，但在混合速度总线系统中，仍然完全向下兼容快速增强模式、快速模式或标准模式设备进行双向通信。除了在高速模式传输期间不执行仲裁和时钟同步外，与快速和标准模式系统保持相同的串行总线协议和数据格式。

#### 5.4 超速模式

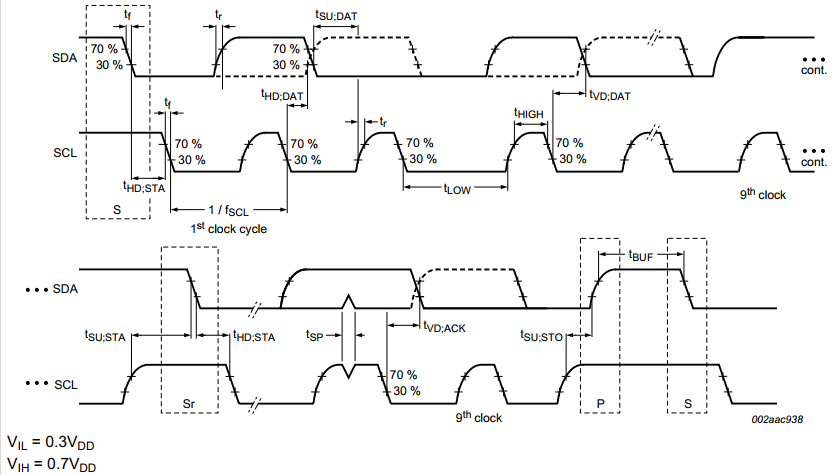
超速模式设备提高了i2c总线传输速度。超速模式设备可以高达5Mb的比特率传输信息。超速模式设备提供推挽驱动器，消除了上拉电阻，允许更高的传输速率。

总线协议和数据格式与标准模式、快速模式和快速增强模式系统保持一致。超速总线设备不兼容双向i2c总线设备。

### 6 电气规范和时序

#### 6.1 标准、快速和快速增强模式设备

规定的SCL时钟的最小高电平和低电平周期决定了标准模式设备的最大比特传输速率为100Kb；快速模式设备为400Kb；快速增强模式为1000Kb。设备必须能够以自己的最大比特率进行传输，要么能够以该速度发送或接收，要么应用时钟同步，强制控制器进入等待状态并延长SCL信号的低电平时间。在后一种情况下，比特传输速率降低。

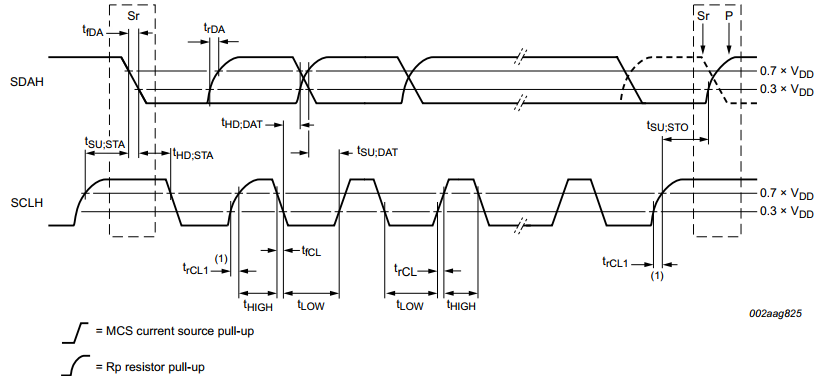


#### 6.2 高速模式设备

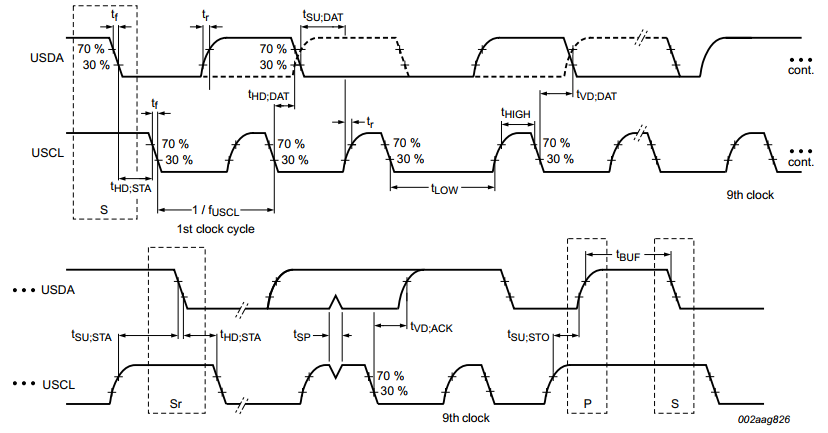
正常START位在高速模式下不存在。地址位、读写位、确认位和数据位都是一样的。只有确认位之后的第一个SCLH时钟信号的上升沿有较大的值，因为外部电阻必须在没有内部电流源的帮助下上拉SCL。

内部生成的SCLH信号的低电平和高电平电平周期分别为200 ns和100 ns,。高速模式控制器满足外部SCLH时钟脉冲的要求时最大比特率为3.4 Mb。因此，高速模式控制器可以使用10MHZ的基本频率或10MHZ的倍数来生成SCLH信号。对SCLH时钟的最大高电平和低电平周期没有限制，最低比特率也没有限制。

对于电容负载，每条线电容最高可达100 pF，允许最大比特率为3.4 Mb。在线上的容性负载较高时，比特率逐渐降低。容性总线负载400pF时，允许最大比特率为1.7 Mb。对于100pF到400pF之间的容性总线负载，定时参数必须线性插值。上升和下降时间与传输线路SDAH和SCLH的最大传播时间一致，以防止端反射。



#### 6.3 超速模式设备



### 7 I2C总线设备电气连接

#### 7.1 上拉电阻大小

总线电容是导线、连接件和引脚的总电容。由于指定的上升时间，这种电容限制了上拉电阻的最大值。

为了计算RC时间常数，考虑与VDD相关的输入阈值VIH为0.7VDD和VIL为0.3VDD。那么V(t)为VDD (1 - e-t / RC)，其中t是充电开始的时间，RC是时间常数。

由于标准模式和快速模式指定的最小电流为3mA，或快速增强模式为20mA，电源电压限制了电阻R的最小值。



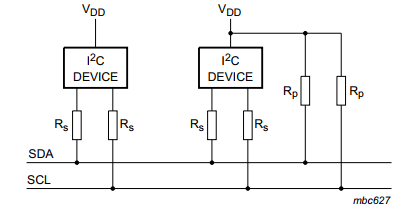
#### 7.2 总线最大电容

总线电容限值用于限制上升时间的减少，并允许在额定频率下工作。虽然大多数设计可以很容易地保持在这个限制内，但有些应用程序可能会超过这个限制。系统设计人员可以采用几种策略来处理总线电容过大的问题。

* 降低SCL频率：总线以较低的速度运行；
* 更高的驱动输出：具有更高驱动电流的设备；
* 总线缓冲器：有许多可用的总线缓冲器设备，可以将总线划分为段，以便每个段的电容低于允许的限制；
* 开关上拉电路：开关上拉电路可以通过在需要时交替切换低值上拉来加速上升边；

#### 7.3 串联保护电阻

如图45所示，例如300 Ω的串联电阻可用于保护SDA和SCL线路上的高压尖峰。如果使用串联电阻，设计人员必须将额外的电阻加入到R和总线线电容的计算中。

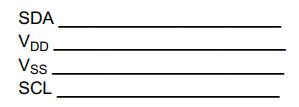


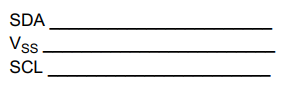
#### 7.4 输入漏极电流

每个输入输出连接的高电平输入电流最大值不能超过10μA。由于高电平需要0.2VDD的噪声裕度，这个输入电流限制了R的最大值。这个限制取决于VDD。

#### 7.5 布线

一般来说，布线必须使总线之间的串扰和干扰最小化。由于上拉的阻抗相对较高，总线在高电平时最容易受到串扰和干扰。





这些布线也导致SDA和SCL线具有相同的电容载荷。如果使用带有VSS或VDD层的PCB，则可以省略VSS和VDD线。