

I2C 通信协议详解

Base info

I2C 通讯

I2C 通信引脚

I2C 总线信号电平

I2C 报文包格式

启动/停止条件

I2C 地址帧

7 位设备寻址

10 位设备寻址

I2C 数据帧

停止条件

I2C 数据传输步骤

I2C 应用

I2C 限制

I2C 有什么用

I2C 是如何工作的

如何解决 I2C 通信问题

I2C 通信协议详解

I²C（Inter-Integrated Circuit）字面上的意思是**集成电路之间**，它其实是**I²C Bus**简称，所以中文应该叫**集成电路总线**，它是一种串行通信总线，使用多主从架构

Base info

I2C（内部集成电路）是一种串行通信协议，允许将多个设备连接到一条总线上。它通常用于嵌入式系统和其他需要少量设备相互通信的应用程序。

数字设备之间的数据传输主要可以通过有线介质以两种方式进行，并行或串行。

微控制器使用许多串行协议与传感器和显示器等外部设备进行通信，I2C 通信就是其中之一。

在设备和系统之间定义通信模式的语言称为协议。它指定了电气方面、传输速率、位排序和位模式含义。

对于串行数据通信，使用不同的协议，例如 TWI、USB、SPI 或 UART 等。它在微控制器项目中有很多应用。

I2C 通讯

I2C（Inter-integrated circuit 发音为“eye-squared-see”）是一种允许从不同的芯片或电路与不同的主芯片通信的协议。

它仅用于短距离通信。它在电线的使用方面具有优势，因为它只需要 2 根电线，但仅这 2 根电线就可以支持 1008 个设备。

I2C 协议是飞利浦半导体在 1980 年发明的，旨在提供 CPU 和各种外围芯片之间的轻松板载通信。

I²C 代表内部集成电路。它是一种短距离串行接口，只需要两条总线即可进行双向数据传输。

它用于在短距离通信中将低速外围 IC 连接到微控制器。低速外设包括外部 EEPROM、数字传感器、I2C LCD 和温度传感器。

I²C 定义了三种附加模式：

在 1 MHz：“快速模式加”。

3.4 MHz：“高速模式”。

5 MHz：“超快模式”。

I2C 支持多主机通信，这意味着多个主机可以连接到总线并可以相互通信以及与从机通信。

它还支持时钟延展，允许从设备将时钟线保持在低电平以指示它需要更多时间来处理命令或数据。

I2C 通信引脚

I2C 是一种简单高效的协议，只需要两条线进行通信：一条时钟线和一条数据线。它通常用于需要低成本、低功耗通信的应用。

例如传感器、显示器和其他外围设备。可以将任意数量的从设备和任意数量的主设备连接到这两个信号。

与 UART 通信不同，此协议需要时钟信号，因为这是同步通信。通过使用时钟信号，它实现了高速数据传输率。

- SDA（串行数据）线：此线将数据发送到从设备。从一个设备发送到另一个设备的任何数据都通过 SDA 线。
- SCL（串行时钟）线：为数据传输提供必要的同步时钟。

总线主机产生时钟信号。我们还需要指定要发送的位数以及时钟线的时钟频率。

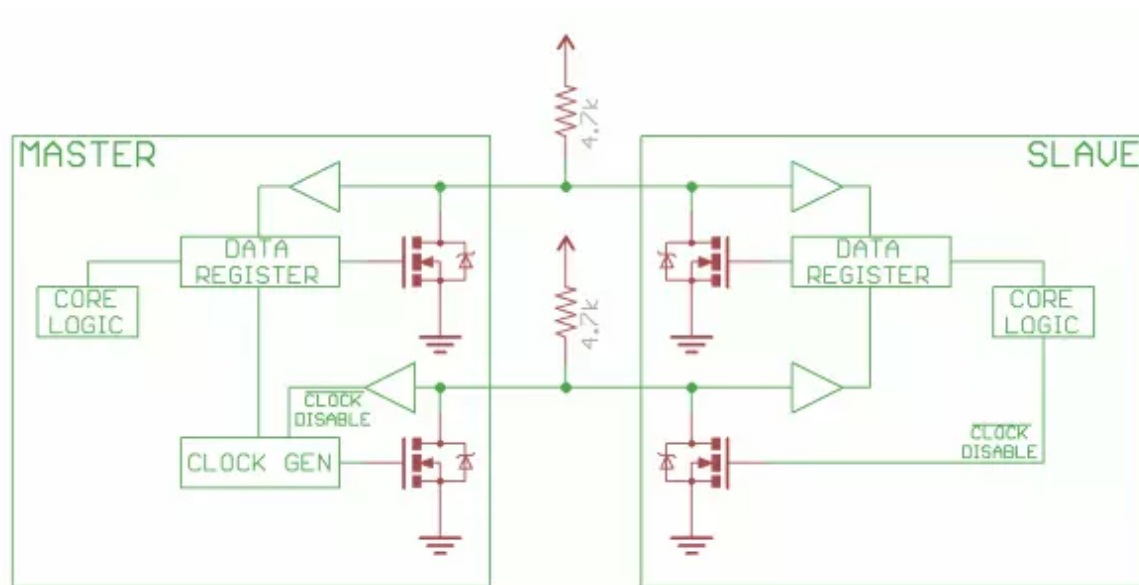
在某些情况下，从机变得缓慢。这可能是因为给它们的数据太长或者每个时钟周期的时间跨度太短。

然后从机将时钟强制拉低，以便它们可以在下一个周期开始之前轻松准备数据。这种现象被称为“时钟拉伸”。

I2C 的总线驱动器是“漏极开路”，这意味着它们可以将信号线拉低，但不能将其拉高。

假设如果一个设备试图将线路拉低，而另一个设备试图将线路拉高，那么就不会有总线争用。这可以保护驱动器免受损坏，并防止系统过热。

如下图所示：



I2C 总线信号主从

当没有其他信号将其拉低时，连接在线路上的上拉电阻将该线路驱动为高信号。电阻器可能具有不同的值。

但通常我们从 4.7k 电阻器开始，然后根据我们的要求降低值，如果需要的话。

I2C 总线信号电平

有时，系统中的设备具有不同的电势，即一个设备的电压高于另一个设备的电压。但是可以在不使用电平转换电路的情况下连接这两个设备。

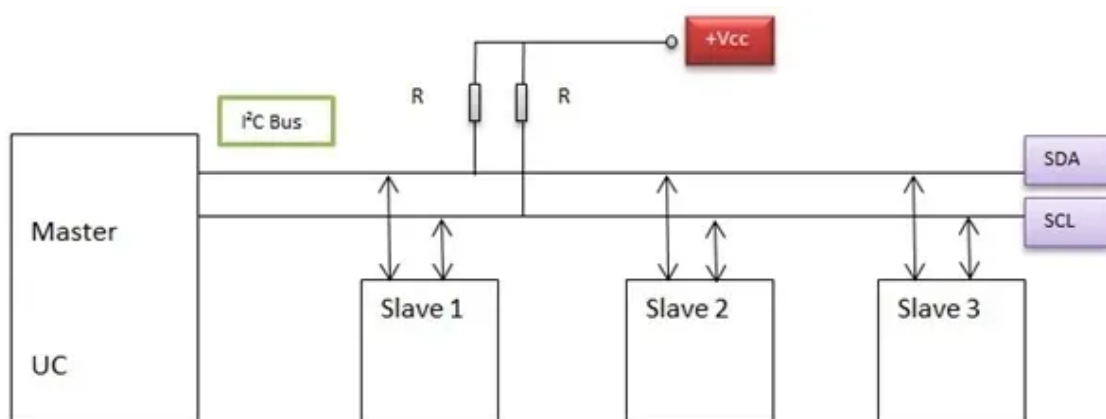
这是通过将上拉电阻连接到电压电平较低的设备来完成的。这种方法不一定适用于所有情况。

但在较低电压大于较高电压系统的输入电平的某些情况下它工作得很好。一个很好的例子就是 3.3V 加速度计和 5V Arduino。

I2C 报文包格式

数据由主设备或从设备在串行数据线 (SDA) 上的任一方向发送。只要主机可以开始数据传输，从机就会响应主机。

一条公共总线上可以有多个主控器，但一次只能有一个处于活动状态。SCL 时钟线始终由主机驱动。



I²C bus with a single master and three slaves

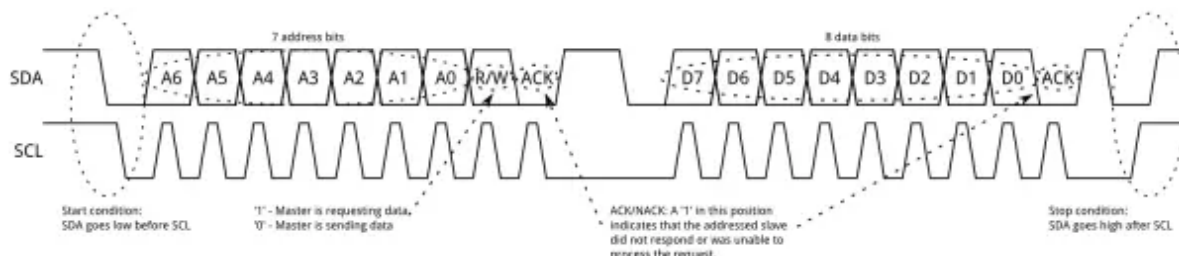
一个主机的I2C总线和三个从机

SCL 和 SDA 线都是漏极开路驱动器，并通过上拉电阻连接到正电源电压。I²C 设备只能将线路拉低，不能将其拉高。

当线上没有设备拉动时，会通过上拉电阻浮高。这就是上拉电阻在 I²C 中很重要的原因。

通过 I2C 进行的通信非常复杂。为了进行有效通信，总线上设备的信号必须遵守特定协议。

但幸运的是，大多数设备会自行处理这些小细节，我们可以只专注于我们想要传输的数据。



SCL SDA 信号

所有消息都分为两个框架：

- 地址帧。
- 数据帧。

在地址帧中，主机告知要将数据传输到的从机地址。数据帧由 8 位消息组成，这些消息从主机传递到从机或从从机传递到主机。

当 SCL 线变低时，数据被放置在 SDA 线上。当 SCL 线变高时，数据就会被采样。

启动/停止条件

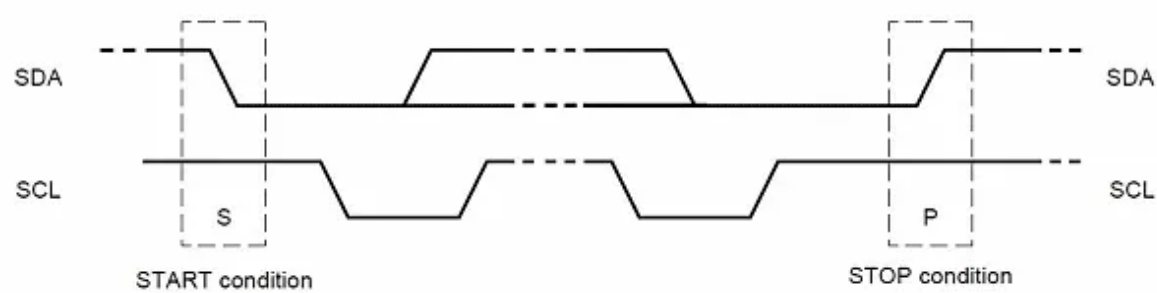
主设备发起的每个 I²C 命令都以 START 条件开始，以 STOP 条件结束。对于这两种情况。

SCL 都必须很高。在启动条件之后，总线被认为忙并且只有在检测到停止条件后才能被另一个主机使用。

主设备将 SDA（串行数据）拉低并使 SCL（串行时钟）保持高电平以启动地址帧。它提醒所有从属设备传输即将开始。

如果有两个或更多主设备，则首先将 SDA 拉低的主设备将获得总线所有权。

- 启动条件：SDA 从高到低的转换。
- 停止条件：SDA 从低到高的转换。



I²C 地址帧

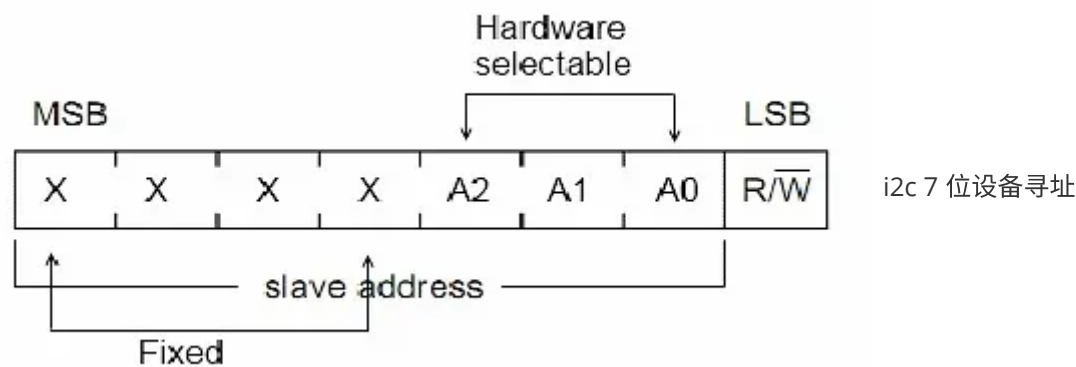
地址帧总是最先出现，因为如果没有地址，设备就不知道数据要传输到哪个从机。如果地址是 7 位，则首先读取 MSB（最高有效位），然后读取 R/W 位指示该操作是读操作还是写操作。

在启动条件之后，我们需要指定从地址，因为一个主设备可以向多个从设备发送数据。

每个从机都有一个唯一的地址。连接到总线的每个从机都可以通过唯一的 7 位或 10 位软件寻址。

7 位设备寻址

在启动条件之后，发送的第一个字节称为控制字节。控制字节的前七位构成从机地址。



- 前四位是固定的。
- 接下来的三位由硬件地址引脚（A0、A1 和 A2）设置，允许用户修改 I²C 地址。它允许多达八个设备在 I²C 总线上运行。
- 第八位（LSB）为数据方向位（R/W）
- LSB 中的 '0' 表示 Master 将向选定的 slave 写入信息
- LSB 中的 '1' 表示 Master 将从 slave 读取数据

- 当接收方获得其地址时，它必须生成确认信号。在此状态下，发送器应释放 SDA 总线以允许接收器获取它。
- 当接收器将 SDA 信号驱动为低电平时，它确认已接收到地址字节。如果接收器未将 SDA 驱动为低电平，则条件为无应答 (NACK) 且操作中止。确认 = 0 伏，无应答 = 高电压。

当发送一个控制字节时，系统中的每个设备/从机将前七个接收位与其地址进行比较。如果地址匹配，则设备认为自己已被主机寻址。

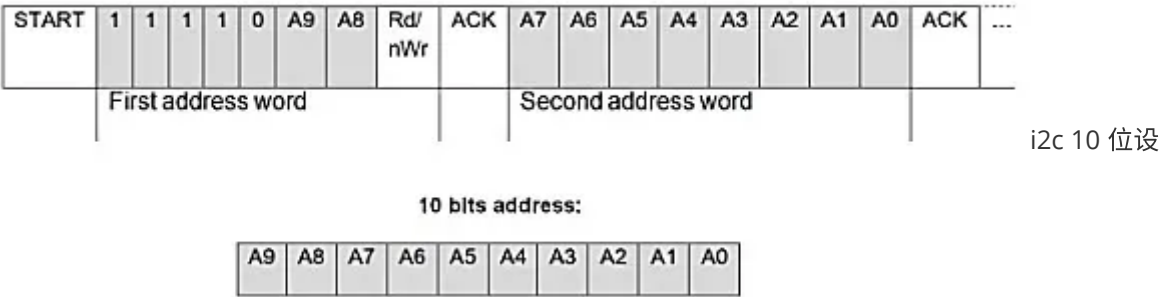
现在，它将根据 R/W 位的值充当从接收器或从发送器。

10 位设备寻址

每个 I²C 设备都必须有一个内置的 7 位地址。因此，据此，世界上将只有 128 种不同的 I²C 设备类型。

但是有更多不同的 I²C 设备，并且很可能有两个设备在 I²C 总线上具有相同的地址。为了克服这个问题，设备通常有多个内置地址。

我们可以通过设备上的外部配置引脚来选择这些地址。为了扩展可用设备地址的范围，I²C 规范推出了 10 位寻址方案。



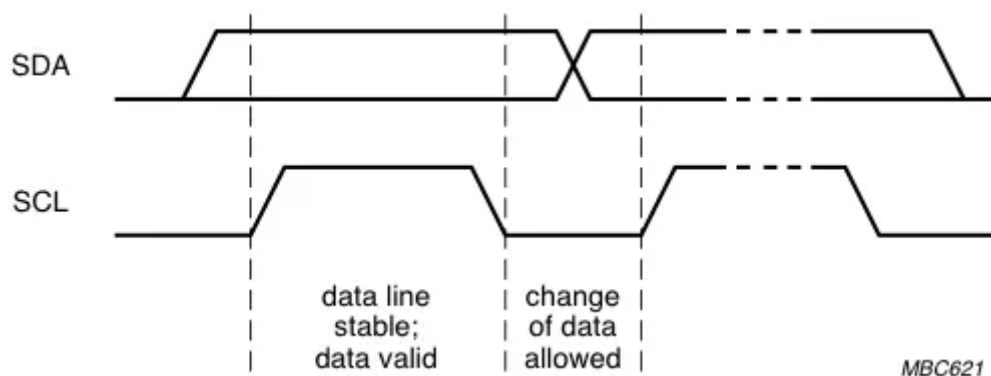
备寻址

- 两个地址字而不是一个地址字用于设备寻址。
- 第一个地址字包含常规代码，如 MSB 上的“11110”，以告知总线上的从机 10 位设备地址。之后，它包含 10 位地址和 R/W 位的两个 MSB。
- 第二个地址字包含 10 位地址的 8 个最低有效位 (LSB)。因此，添加这些位可确保向后兼容 7 位寻址方案。

I2C 数据帧

现在可以开始数据传输了。时钟脉冲将由主机定期产生。R/W 位将指示数据是否正在被主机读取或写入。

如果要读取数据，则从设备会将数据放在 SDA 上，如果要写入数据，则主设备会将数据放在 SDA 上。在大多数情况下，内部寄存器会根据读取或写入的次数自动递增。



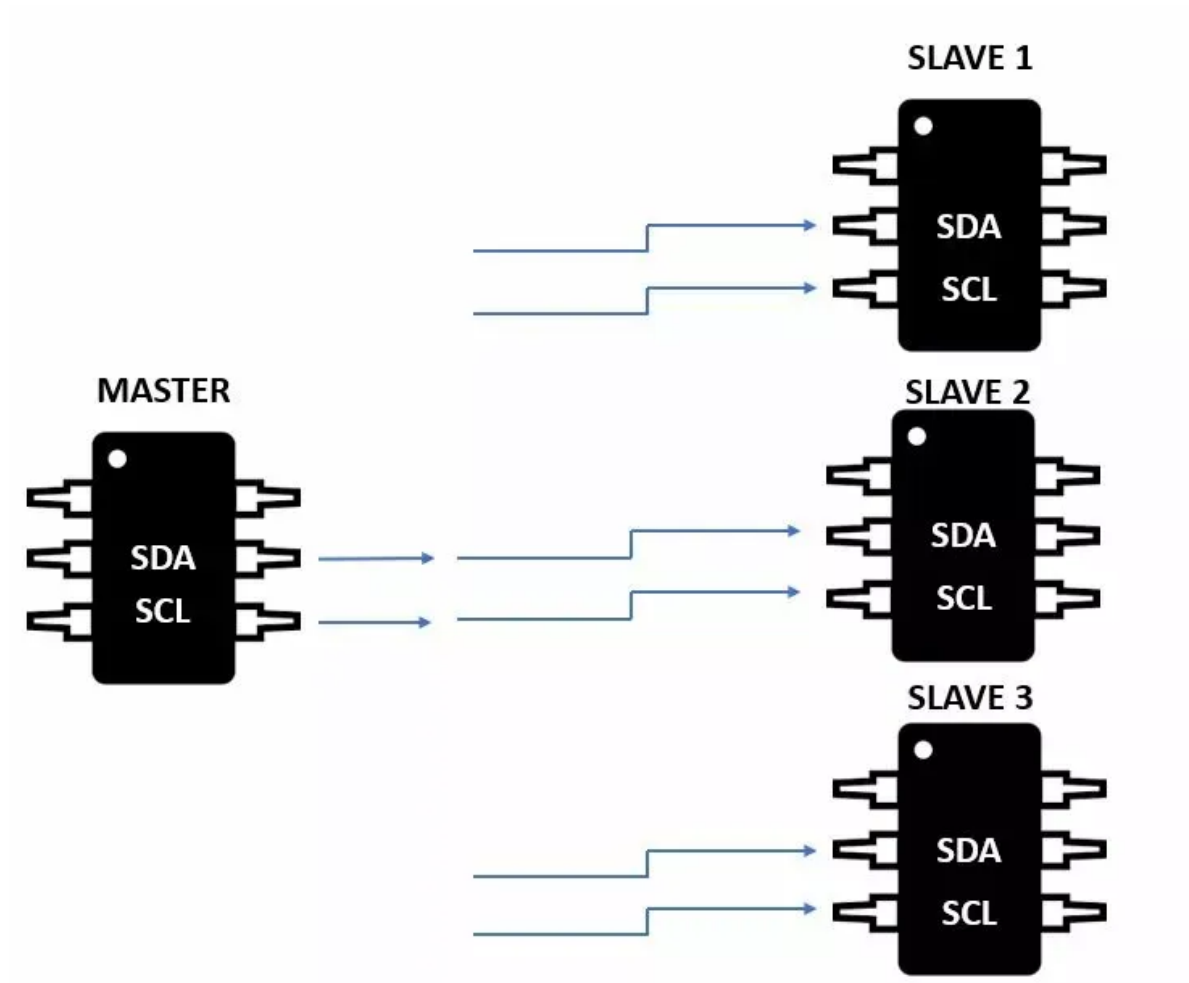
停止条件

所有数据连续传输后，主机将产生停止条件。SCL 应从低电平变为高电平，然后保持高电平，而 SDA 从低电平变为高电平，以定义停止条件。如上图所示。

I2C 数据传输步骤

现在让我们逐步讨论通过 I2C 协议传输数据。

为了启动 I2C 总线上的通信，主机必须首先发送一个启动条件。开始条件是时钟和数据线上的特定信号模式，它向总线上的所有设备指示新通信即将开始。



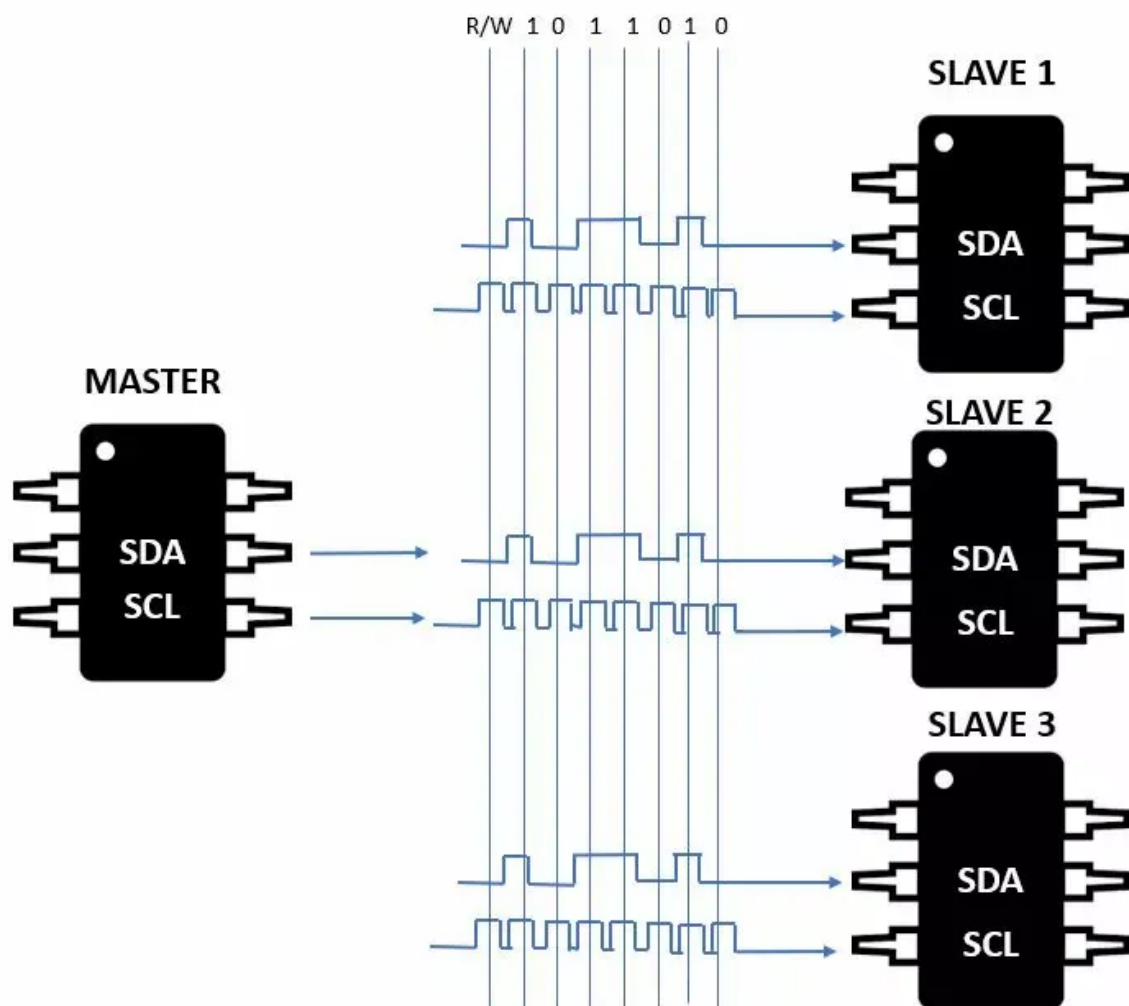
i2c 数据传输步骤

发送开始条件后，主机发送它要与之通信的从机的地址。如前所述，每个从设备都有一个唯一的 7 位地址，主设备通过设置地址的 LSB 来指定是否要读取或写入从设备。

从设备在总线上侦听自己的地址，如果接收到自己的地址，则用确认信号进行响应。

确认信号是数据线上的一个脉冲，被从设备拉低一个时钟周期。该信号向主机表明从机已收到其地址并准备好进行通信。

如果从机收不到自己的地址，或者忙而无法通信，则不发送应答信号，通信终止。然后主机可以稍后重试通信或与不同的从机通信。



i2c 数据传输步骤

在主机发送开始条件和它要与之通信的从机地址后，如果从机收到自己的地址，则以应答信号响应。

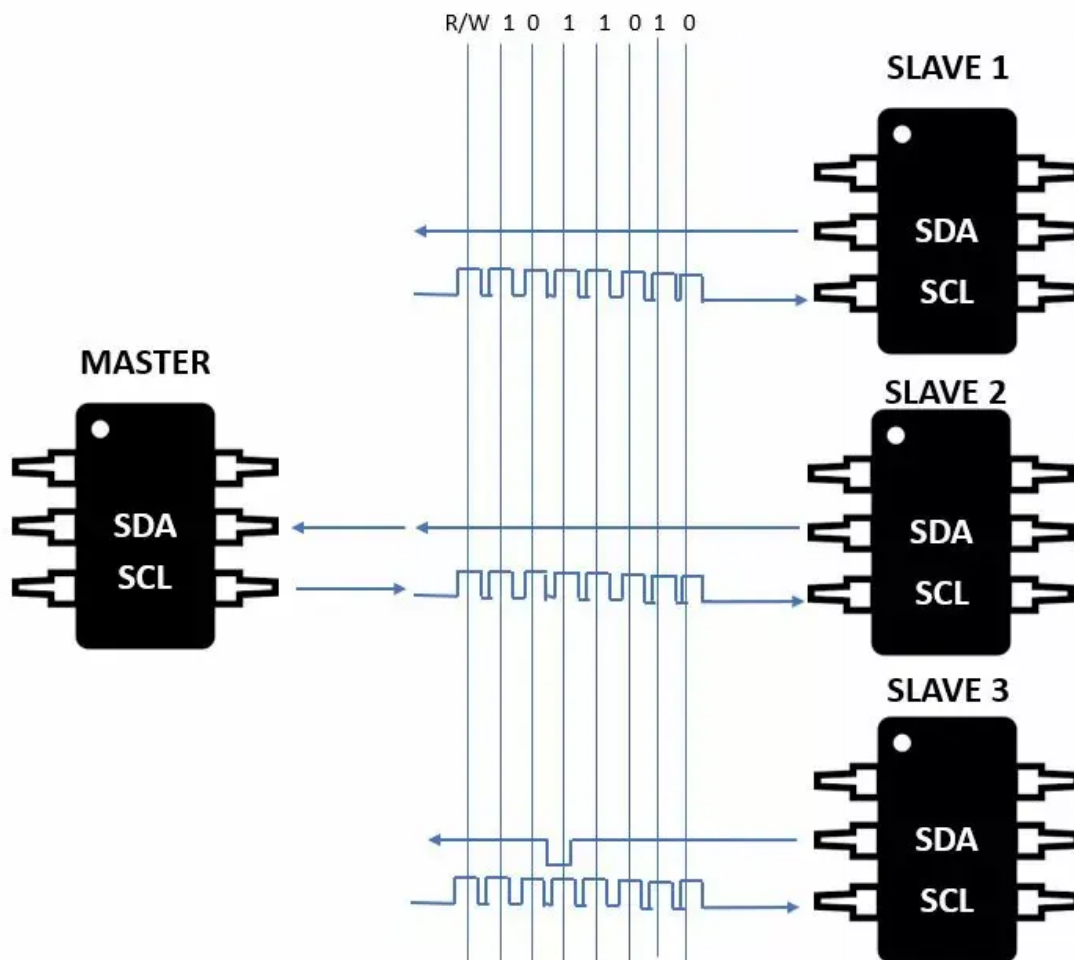
确认信号是数据线上的一个脉冲，被从设备拉低一个时钟周期。该信号向主机表明从机已收到其地址并准备好进行通信。

如果从机收不到自己的地址，或者忙而无法通信，则不发送应答信号，通信终止。然后主站可以稍后重试通信或与不同的从站通信。

如果收到确认信号，则主机可以继续向从机发送命令或数据。从机将以确认信号响应以指示它已接收到数据。这个过程可以重复多次来发送和接收多个字节的数据。

重要的是要注意确认信号在某些情况下是可选的。如果某些从机无法接收或处理数据，或者如果它们被配置为在无应答模式下运行。

则它们可能不会发送应答信号。在这些情况下，主机可能需要使用其他方法来检测数据是否已收到或检查错误。



i2c 数据传输步骤

在从机响应确认信号以表明它已准备好进行通信后，主机可以向从机发送命令或数据。命令或数据作为数据线上的一系列位发送，每个位都由主机的时钟信号计时。

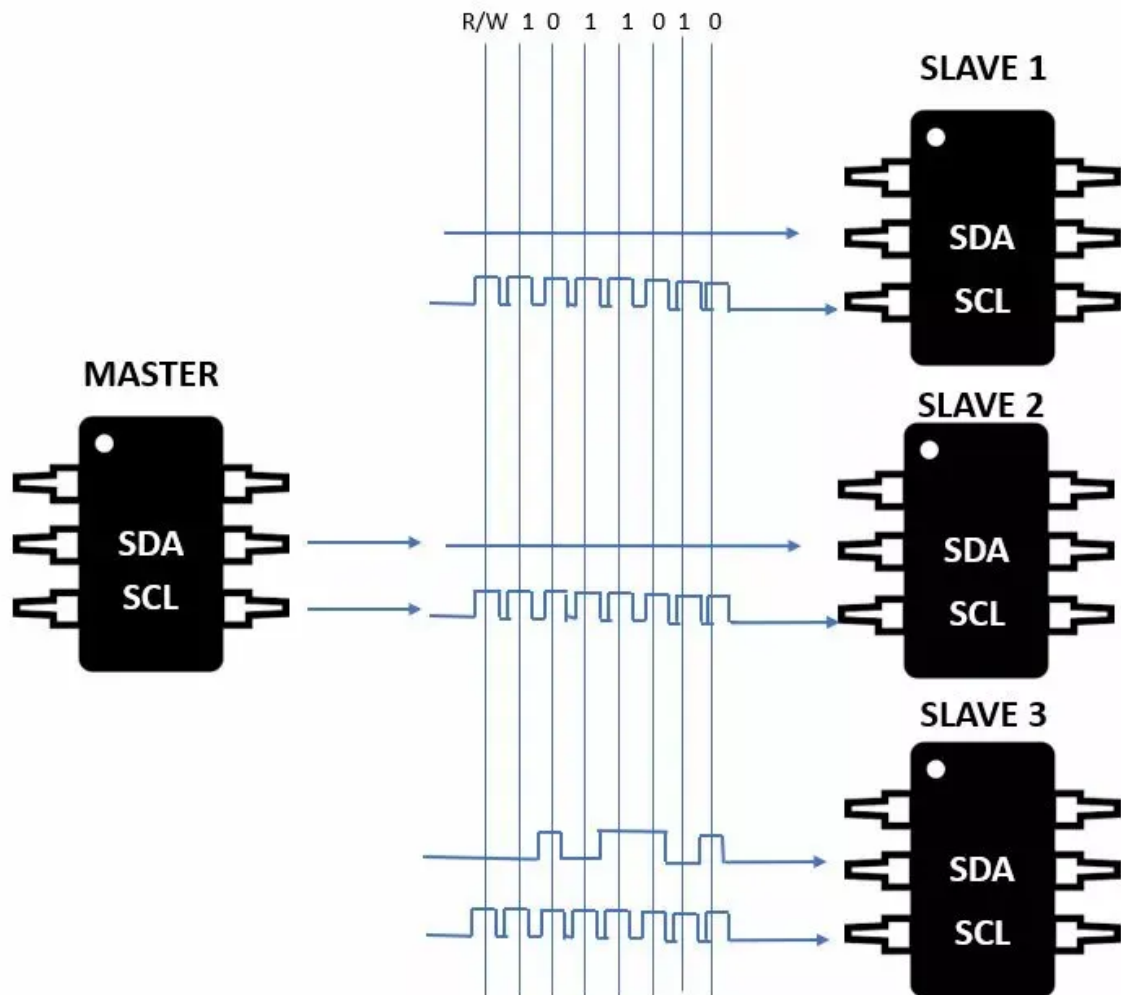
从机侦听数据线上的数据并以确认信号响应以表明它已接收到数据。这个过程可以重复多次来发送和接收多个字节的数据。

重要的是要注意确认信号在某些情况下是可选的。如果某些从机无法接收或处理数据，或者如果它们被配置为在无应答模式下运行。

则它们可能不会发送应答信号。在这些情况下，主机可能需要使用其他方法来检测是否接收到数据或检查错误。

除了发送和接收数据外，主机还可以向从机发送特殊命令以控制其行为或请求特定信息。

例如，主机可以发送命令来读取从机上的寄存器，或者将数据写入特定的内存位置。从机用适当的数据或确认信号来响应这些命令，以表明命令已被接收和执行。



i2c 数据传输步骤

在 I2C 通信中，主机和从机可以相互发送和接收多个字节的数据。这在需要在设备之间传输大量数据的应用程序中很有用。

要发送多个字节的数据，主机可以简单地一个接一个地发送每个字节，从机在每个字节后用一个应答信号进行响应，以表明它已经接收到数据。

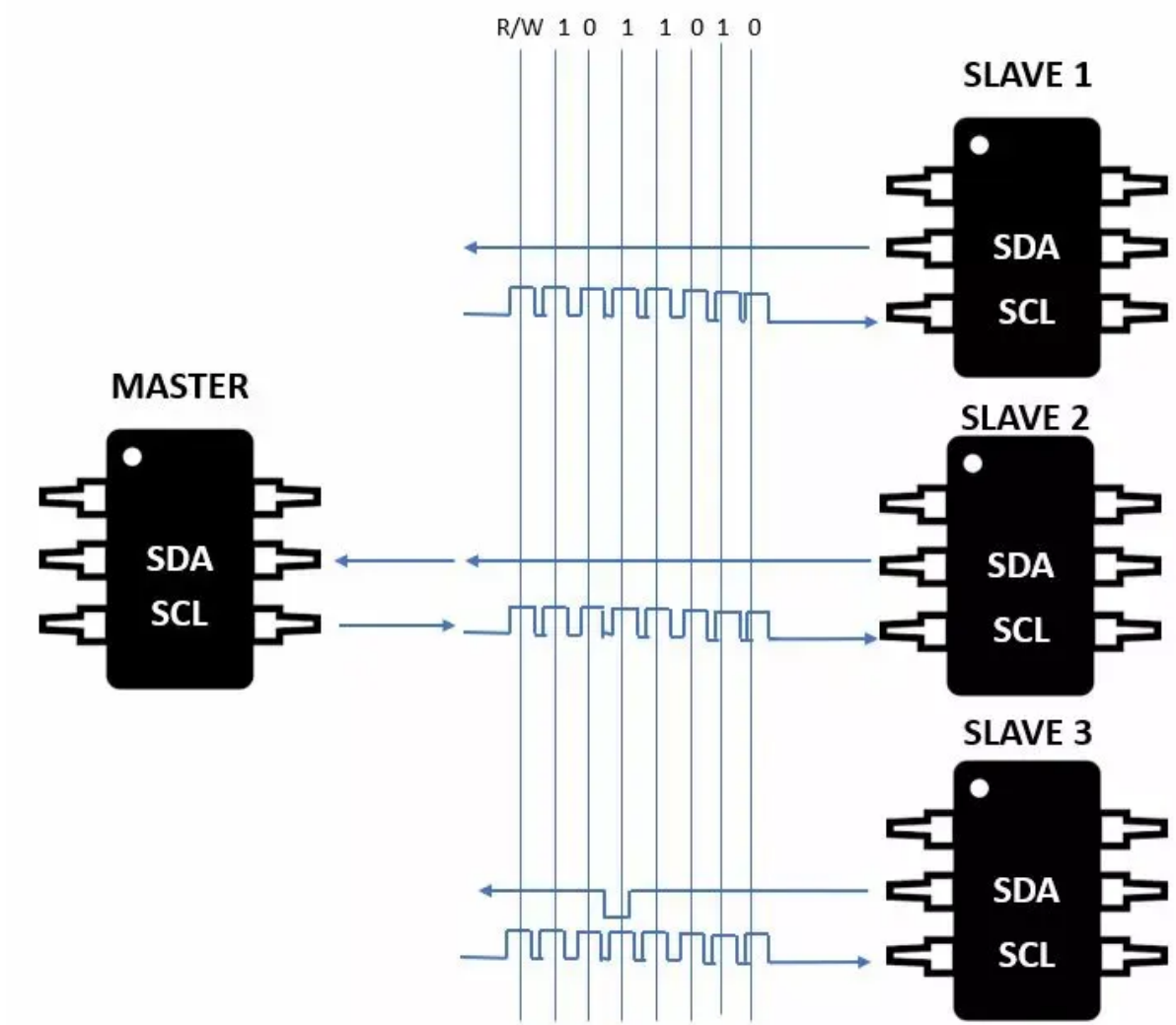
从机也可以类似的方式向主机发送多个字节的数据，主机在接收到每个字节时进行确认。

重要的是要注意确认信号在某些情况下是可选的。如果某些从机无法接收或处理数据，或者如果它们被配置为在无应答模式下运行。

则它们可能不会发送应答信号。在这些情况下，主机可能需要使用其他方法来检测是否接收到数据或检查错误。

除了发送和接收数据外，主机还可以向从机发送特殊命令以控制其行为或请求特定信息。

例如，主机可以发送命令来读取从机上的寄存器，或者将数据写入特定的内存位置。从机用适当的数据或确认信号来响应这些命令，以表明命令已被接收和执行。



i2c 数据传输步骤

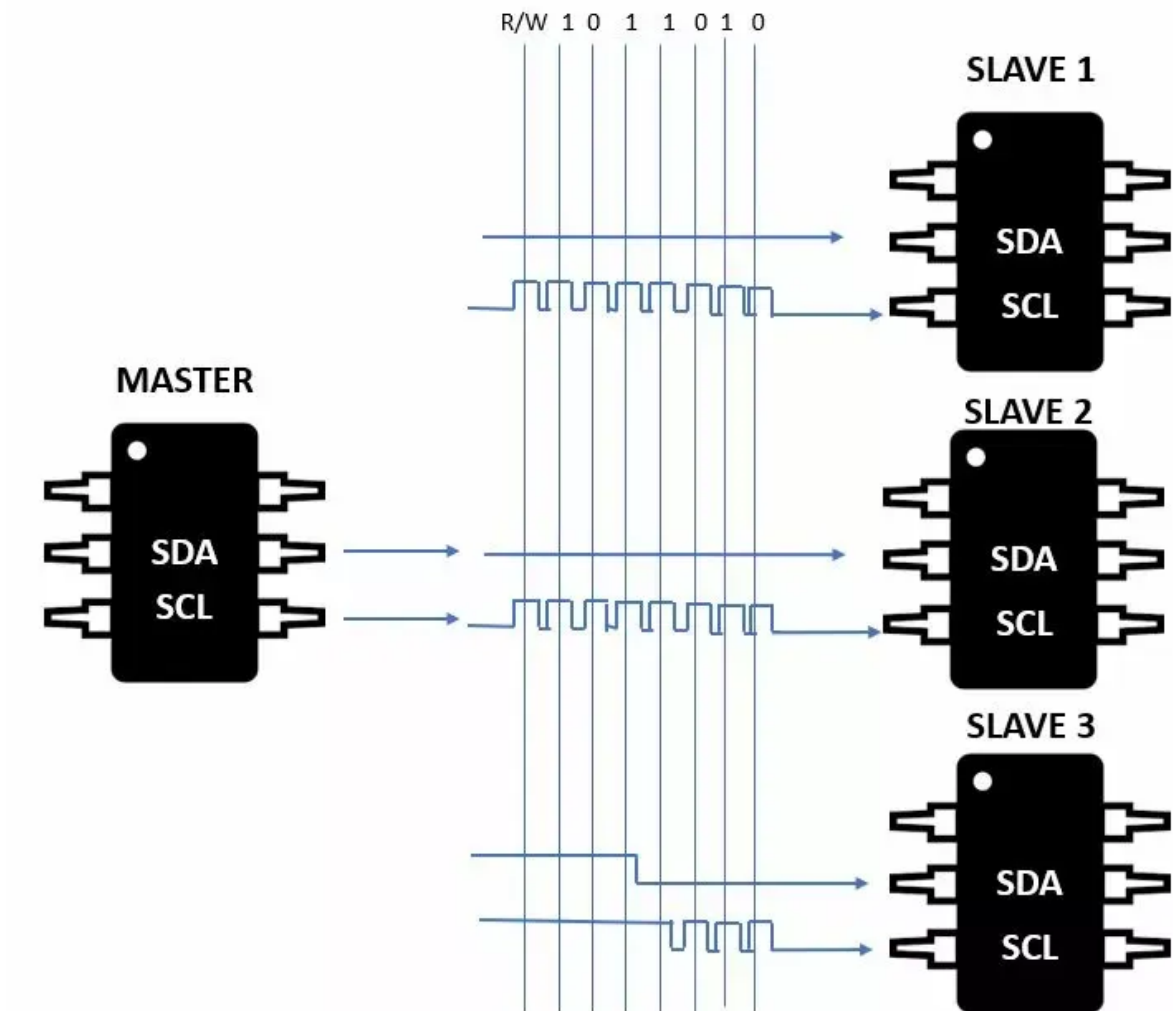
主机完成与从机的通信后，发送停止条件以结束传输。停止条件是时钟和数据线上的特定信号模式，它向总线上的所有设备指示通信已完成。

停止条件告诉从机它可以停止侦听数据并可以返回到低功耗状态。它还释放总线，以便其他主机或从机可以发起通信。

发送停止条件后，主机可以通过发送开始条件和不同从机的地址来启动新的通信，或者它可以进入低功耗状态以节省能量。

重要的是要注意停止条件在某些情况下是可选的。如果某些设备被配置为在连续模式下运行，则它们可能不会发送停止条件。

在这种模式下，通信正在进行并且数据不间断地连续传输。在这些情况下，主机或从机可能需要使用其他方法来检测传输结束或终止通信。



i2c 数据传输步骤

I2C 应用

I2C 是各种应用中广泛使用的协议，包括：

嵌入式系统：I2C 通常用于嵌入式系统，用于与传感器、显示器和其他外围设备进行通信。它的简单性和低功耗使其成为这些类型应用的理想选择。

工业自动化：I2C 在工业自动化系统中用于连接传感器、执行器和其他设备以控制和监视过程。

消费电子产品：I2C 用于一系列消费电子设备，例如智能手机、平板电脑和笔记本电脑，以与传感器、显示器和其他外围设备进行通信。

医疗设备：I2C 用于医疗设备，例如监控系统、血糖仪和便携式氧气浓缩器，用于与传感器和其他外围设备进行通信。

汽车：I2C 在汽车系统中用于与传感器、显示器和其他外围设备通信，例如发动机控制系统、信息娱乐系统和安全系统。

航空航天：I2C 在航空航天应用中用于与传感器、显示器和其他外围设备通信，例如飞机航空电子系统和卫星通信系统。

I2C 限制

I2C 是一种简单高效的协议，广泛用于嵌入式系统和其他应用程序。但是，它确实有一些您应该注意的限制：

有限距离：I2C 专为短距离通信而设计，通常为几米或更短。可以使用中继器或其他技术来延长总线的长度，但这会增加系统的成本和复杂性。

设备数量有限：I2C 在一条总线上最多支持 128 个设备，这在某些应用程序中可能会受到限制。但是，可以使用多条总线来连接更多的设备，或者可以使用其他协议，例如 SPI（串行外设接口）。

限速：I2C通信的速度受时钟信号的限制，时钟信号由主机产生。I2C 的最大速度通常为几 MHz，这对于某些应用来说可能不够。

功率有限：I2C是为低功率通信而设计的，不适合需要大功率或高压通信的应用。

复杂性：I2C 的实现可能比其他一些串行通信协议更复杂，尤其是在具有多个主机或从机的系统中。它可能需要额外的硬件和软件支持来管理通信并确保设备正确同步。

I2C 有什么用

I2C 是一种串行通信协议，用于将多个设备连接到一条总线。它通常用于嵌入式系统和其他需要少量设备相互通信的应用程序，例如传感器、显示器和其他外围设备。

I2C 是如何工作的

I2C 的工作原理是允许主设备控制总线上的通信以及向从设备发送数据和从从设备接收数据。

每个设备都有一个唯一的 7 位地址，主设备通过发送开始条件和它要与之通信的从设备的地址来启动通信。

从机响应确认信号，表明它已收到地址，然后主机可以向从机发送数据。主机发送停止条件终止通信。

如何解决 I2C 通信问题

要排除 I2C 通信问题，您可以尝试以下步骤：

- 检查设备是否正确连接到总线以及是否使用了正确的地址。
- 确认设备配置正确并且以相同速度通信。
- 使用示波器或逻辑分析仪监视总线上的信号，确保开始和停止条件、地址和数据正确传输。
- 检查是否有任何硬件问题，例如损坏的连接器和断线。
- 检查是否有任何软件问题，例如不正确的驱动程序或库配置。
- 检查是否有任何可能引起干扰的外部因素，例如电磁干扰或电源问题。