

SPI、UART、I2C 通信的区别与应用

电子设备之间的通信就像人类之间的交流，双方都需要说相同的语言。在电子产品中，这些语言称为通信协议。

之前有单独地分享了 SPI、UART、I2C 通信的文章，这篇对它们做一些对比。

串行 VS 并行

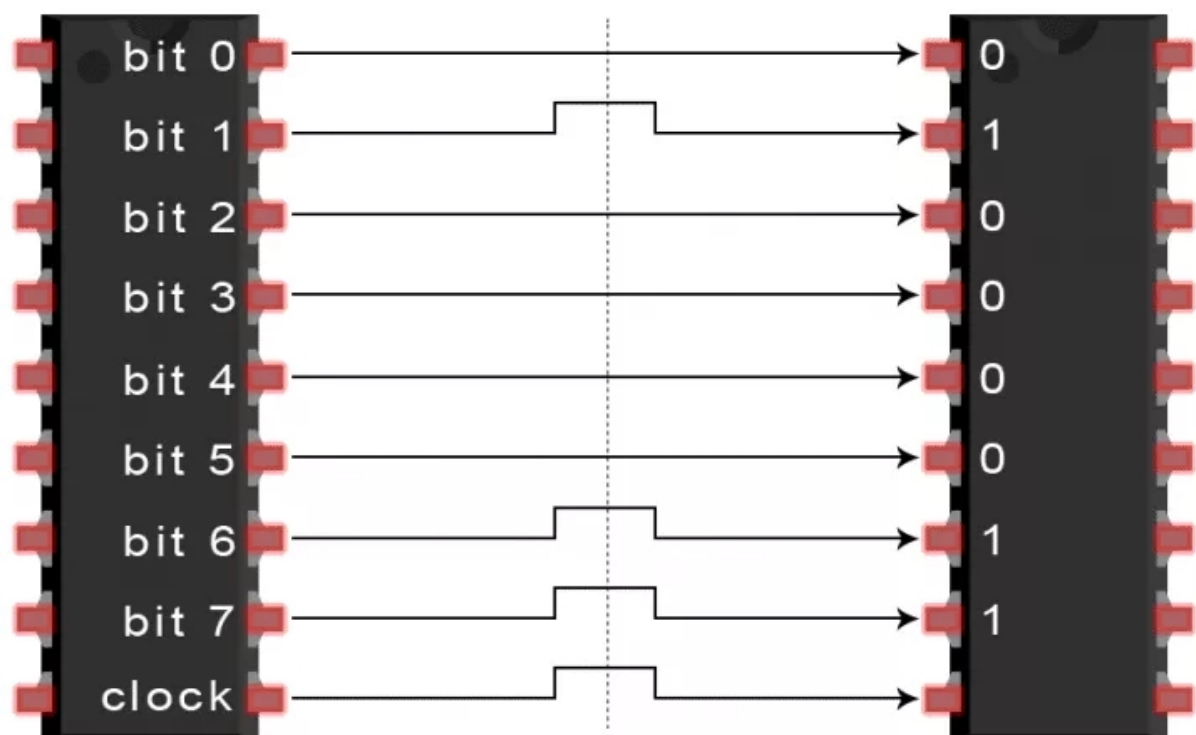
电子设备通过发送数据位从而实现相互交谈。位是二进制的，只能是 1 或 0。

通过电压的快速变化，位从一个设备传输到另一个设备。在以 5V 工作的系统中，

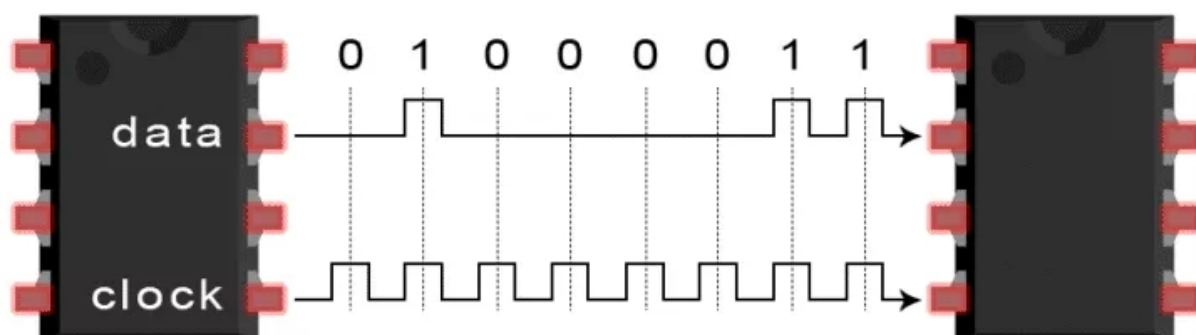
“0”通过 0V 的短脉冲进行通信，而“1”通过 5V 的短脉冲进行通信。数据位可

以通过并行或串行的形式进行传输。在并行通信中，数据位在导线上同时传输。

下图显示了二进制（01000011）中字母“C”的并行传输：



在串行通信中，位通过单根线一一发送。下图显示了二进制（01000011）中字母“C”的串行传输：



SPI 通信

SPI 是一种常见的设备通用通信协议。它有一个独特优势就是可以无中断传输数据，可以连续地发送或接收任意数量的位。而在 I2C 和 UART 中，数据以数据包的形式发送，有着限定位数。在 SPI 设备中，设备分为**主机与从机系统**。主机是控制设备（通常是微控制器），而从机（通常是传感器，显示器或存储芯片）从主机那获取指令。一套 SPI 通讯共包含四种信号线：**MOSI** (Master Output/Slave Input) – 信号线，主机输出，从机输入。**MISO** (Master Input/Slave Output) – 信号线，主机输入，从机输出。**SCLK** (Clock) – 时钟信号。**SS/CS** (Slave Select/Chip Select) – 片选信号。

Wires Used	4
Maximum Speed	Up to 10 Mbps
Synchronous or Asynchronous?	Synchronous
Serial or Parallel?	Serial
Max # of Masters	1
Max # of Slaves	Theoretically unlimited*

SPI 协议特点实际上，从机的数量受系统负载电容的限制，它会降低主机在电压电平之间准确切换的能力。**工作原理**

时钟信号

每个时钟周期传输一位数据，因此数据传输的速度取决于时钟信号的频率。时

钟信号由于是主机配置生成的，因此 SPI 通信始终由主机启动。 **设备共享时钟**

信号的任何通信协议都称为同步。 SPI 是一种同步通信协议，还有一些异步通信

不使用时钟信号。例如在 UART 通信中，双方都设置为预先配置的波特率，该

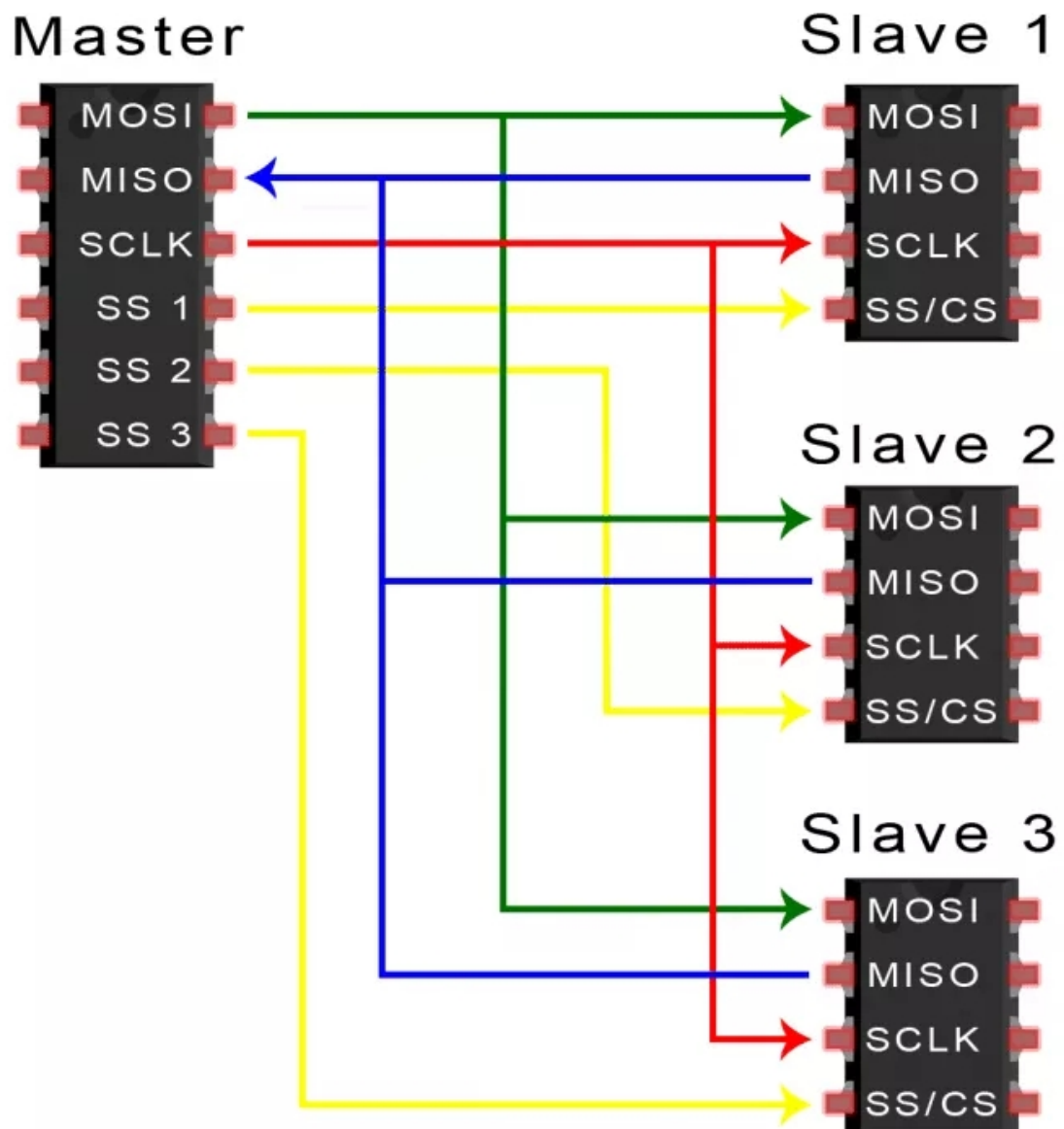
波特率决定了数据传输的速度和时序。

片选信号

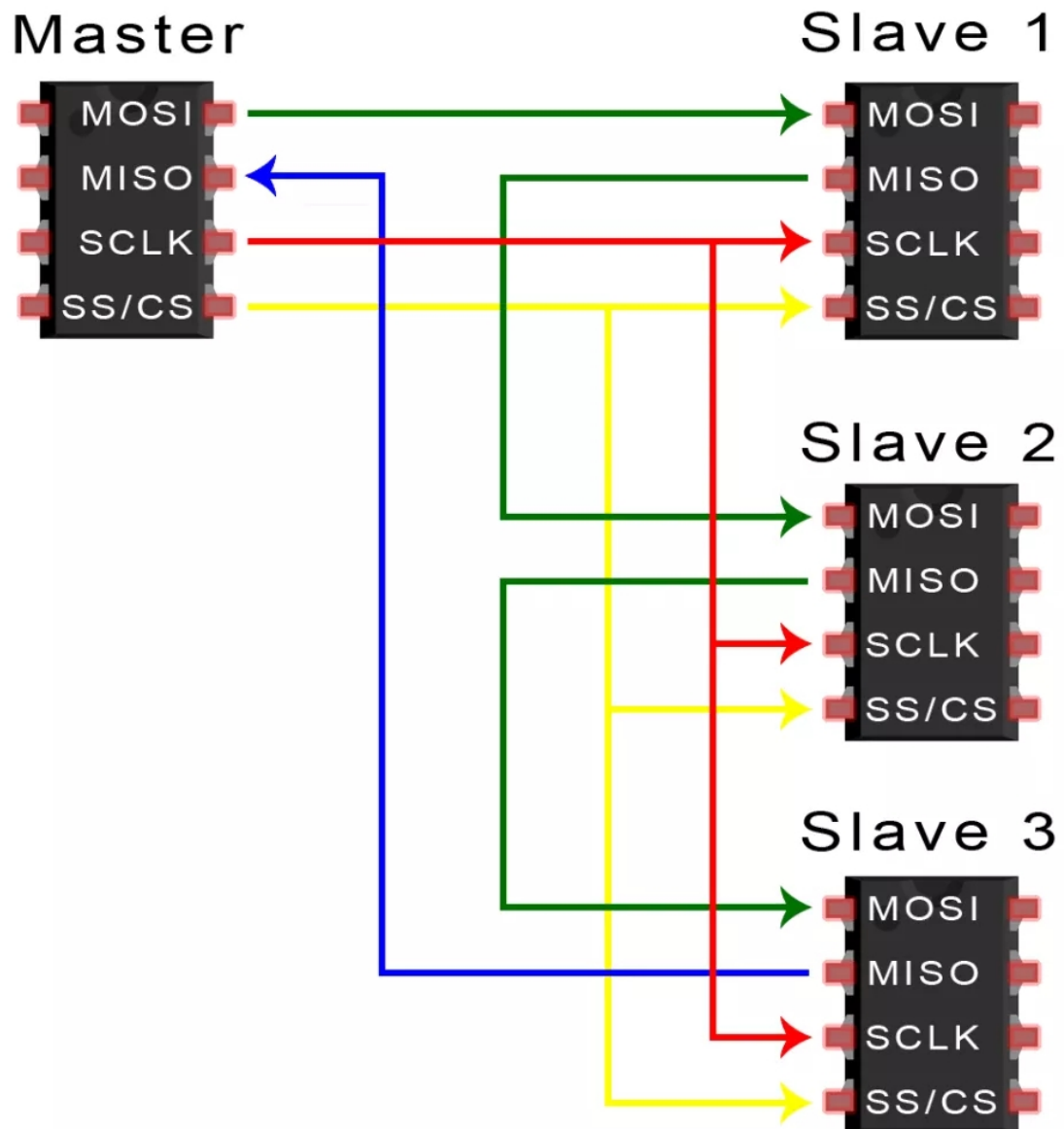
主机通过拉低从机的 CS/SS 来使能通信。 **在空闲/非传输状态下，片选线保持**

高电平。 在主机上可以存在多个 CS/SS 引脚，允许主机与多个不同的从机进行

通讯。



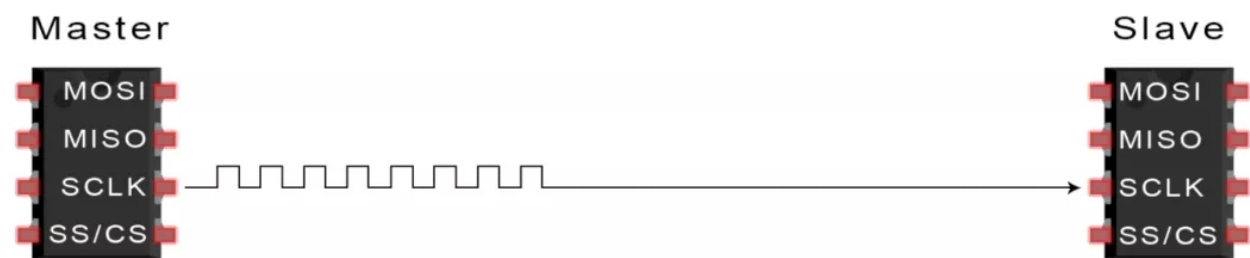
如果主机只有一个片选引脚可用，则可以通过以下方式连接这些从器件：



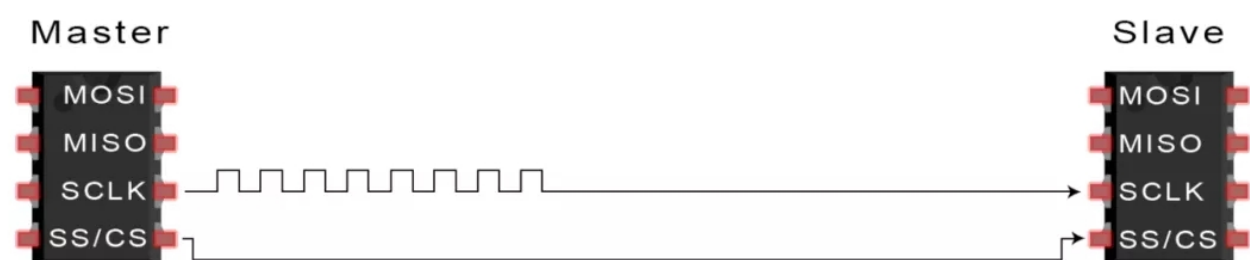
MOSI 和 MISO

主机通过 MOSI 以串行方式将数据发送给从机，从机也可以通过 MISO 将数据发送给主机，两者可以同时进行。所以理论上，**SPI 是一种全双工的通讯协议。**

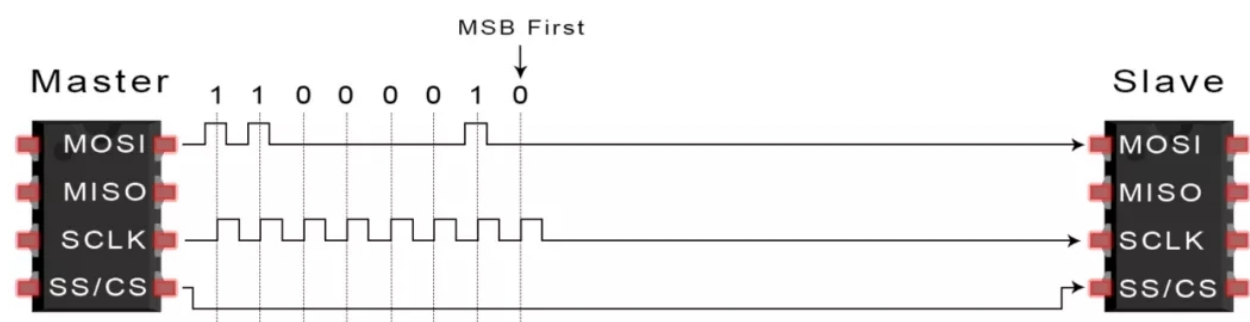
传输步骤 1. 主机输出时钟信号



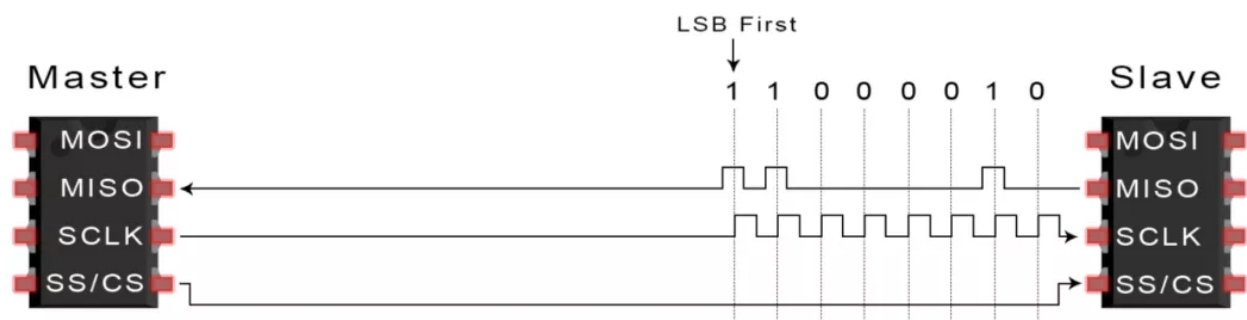
2. 主机拉低 SS / CS 引脚，激活从机



3. 主机通过 MOSI 将数据发送给从机



4. 如果需要响应，则从机通过 MISO 将数据返回给主机



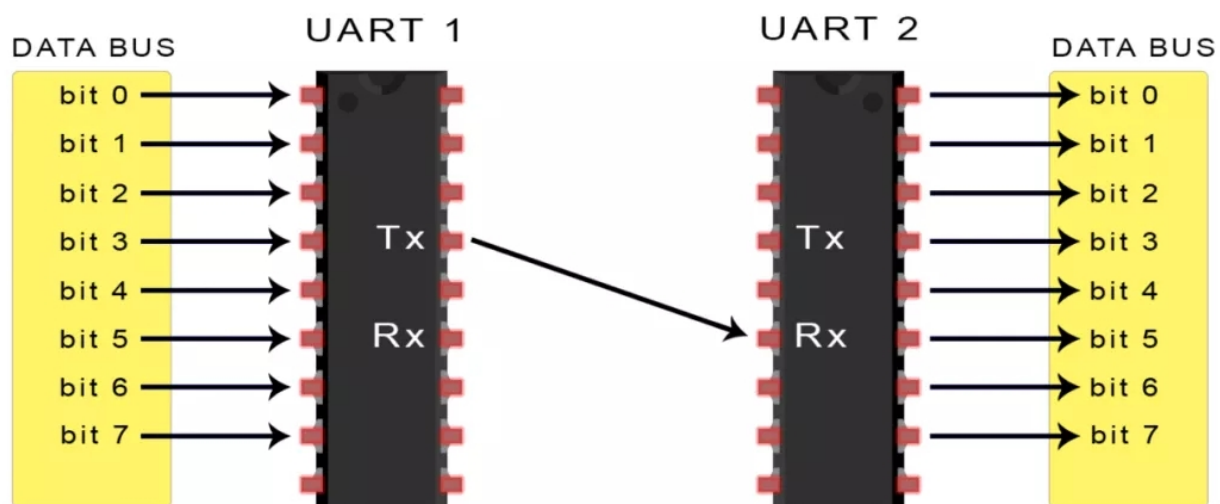
使用 SPI 有一些优点和缺点，如果在不同的通信协议之间进行选择，则应根据项目要求进行充分考量。 **优劣**

优点

SPI 通讯无起始位和停止位，因此数据可以连续流传输而不会中断；没有像 I2C 这样的复杂的从站寻址系统，数据传输速率比 I2C 更高（几乎快两倍）。独立的 MISO 和 MOSI 线路，可以同时发送和接收数据。

缺点

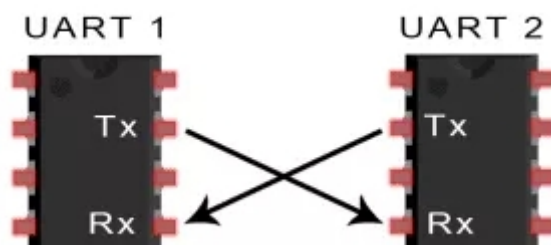
SPI 使用四根线（I2C 和 UART 使用两根线），没有信号接收成功的确认（I2C 拥有此功能），没有任何形式的错误检查（如 UART 中的奇偶校验位等）。



UART 代表通用异步接收器/发送器也称为串口通讯，它不像 SPI 和 I2C 这样的通信协议，而是微控制器中的物理电路或独立的 IC。UART 的主要目的是发送和接收串行数据，其最好的优点是它仅使用两条线在设备之间传输数据。UART 的原理很容易理解，但是如果您还没有阅读 SPI 通讯协议，那可能是一个不错的起点。

UART 通信

在 UART 通信中，两个 UART 直接相互通信。发送 UART 将控制设备（如 CPU）的并行数据转换为串行形式，以串行方式将其发送到接收 UART。只需要两条线即可在两个 UART 之间传输数据，数据从发送 UART 的 Tx 引脚流到接收 UART 的 Rx 引脚：



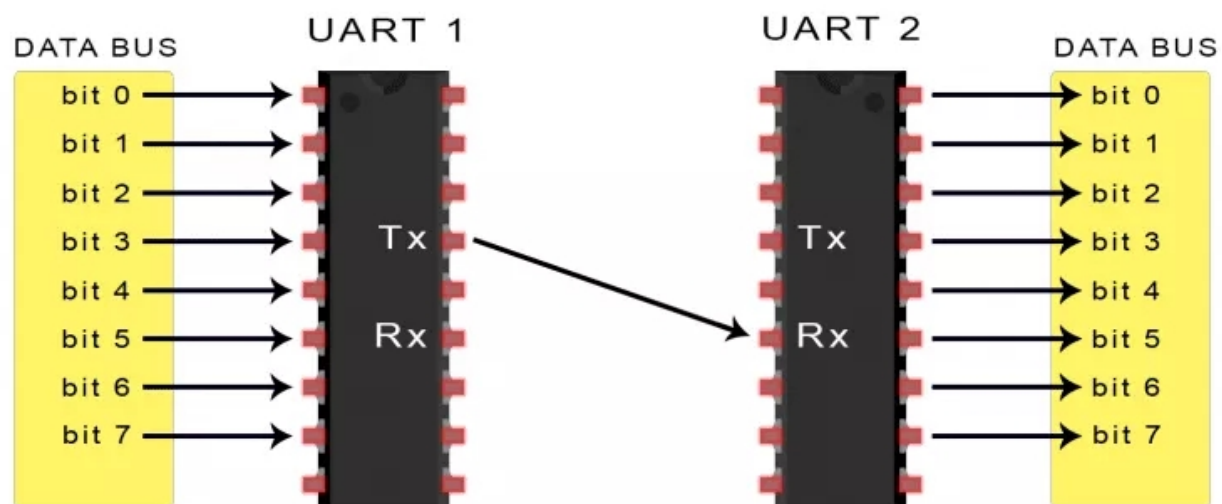
UART 属于异步通讯，这意味着没有时钟信号，取而代之的是在数据包中添加开始和停止位。这些位定义了数据包的开始和结束，因此接收 UART 知道何时读取这些数据。

当接收 UART 检测到起始位时，它将以特定波特率的频率读取。波特率是数据传输速度的度量，以每秒比特数（bps）表示。两个 UART 必须以大约相同的波特率工作，发送和接收 UART 之间的波特率只能相差约 10%。

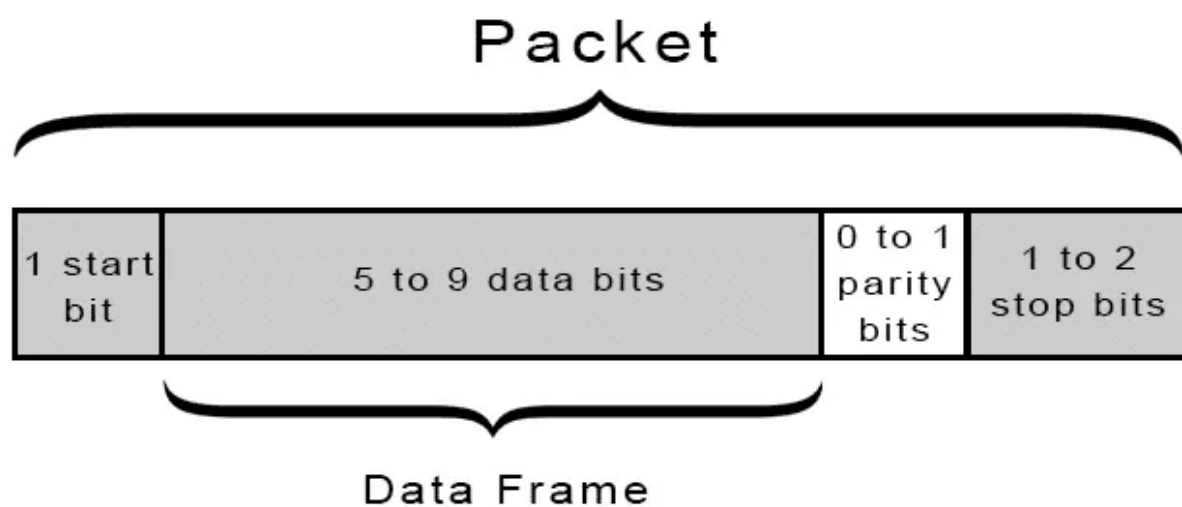
Wires Used	2
Maximum Speed	Any speed up to 115200 baud, usually 9600 baud
Synchronous or Asynchronous?	Asynchronous
Serial or Parallel?	Serial
Max # of Masters	1
Max # of Slaves	1

工作原理

发送 UART 从数据总线获取并行数据后，它会添加一个起始位，一个奇偶校验位和一个停止位来组成数据包并从 Tx 引脚上逐位串行输出，接收 UART 在其 Rx 引脚上逐位读取数据包。



UART 数据包含有 1 个起始位，5 至 9 个数据位（取决于 UART），一个可选的奇偶校验位以及 1 个或 2 个停止位：



起始位：

UART 数据传输线通常在不传输数据时保持在高电压电平。开始传输时发送

UART 在一个时钟周期内将传输线从高电平拉低到低电平，当接收 UART 检测到高电压到低电压转换时，它开始以波特率的频率读取数据帧中的位。

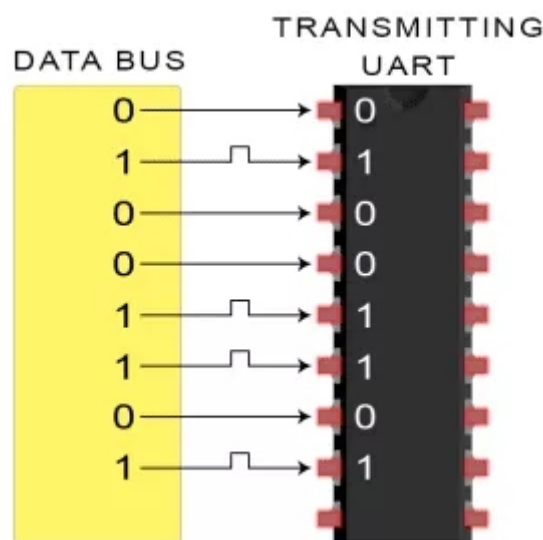
数据帧：数据帧内包含正在传输的实际数据。如果使用奇偶校验位，则可以是 5 位，最多 8 位。如果不使用奇偶校验位，则数据帧的长度可以为 9 位。

校验位：奇偶校验位是接收 UART 判断传输期间是否有任何数据更改的方式。

接收 UART 读取数据帧后，它将对值为 1 的位数进行计数，并检查总数是偶数还是奇数，是否与数据相匹配。

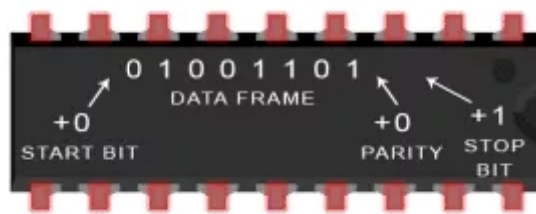
停止位：为了向数据包的结尾发出信号，发送 UART 将数据传输线从低电压驱动到高电压至少持续两位时间。

传输步骤发送 UART 从数据总线并行接收数据：



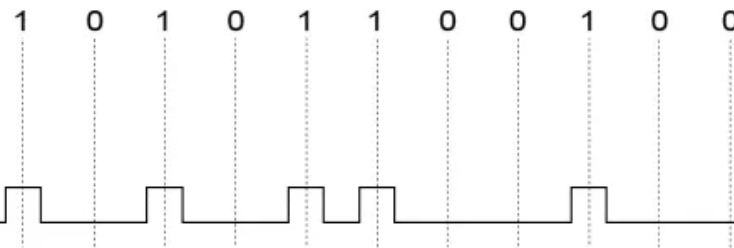
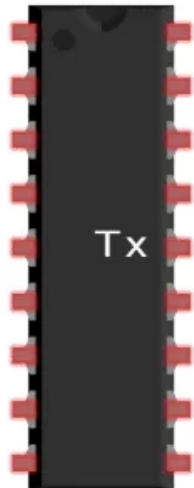
2.发送 UART 将起始位，奇偶校验位和停止位添加到数据帧：

TRANSMITTING UART

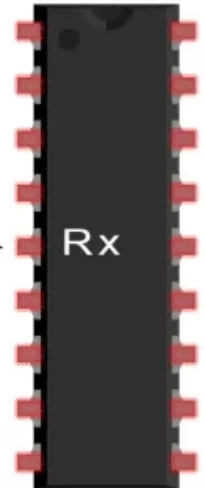


3.整个数据包从发送 UART 串行发送到接收 UART。接收 UART 以预先配置的波特率对数据线进行采样：

TRANSMITTING UART

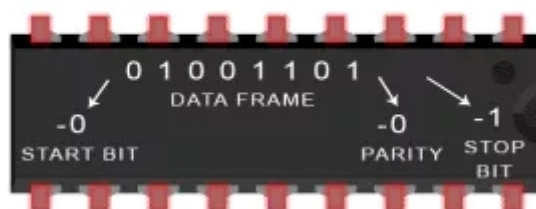


RECEIVING UART



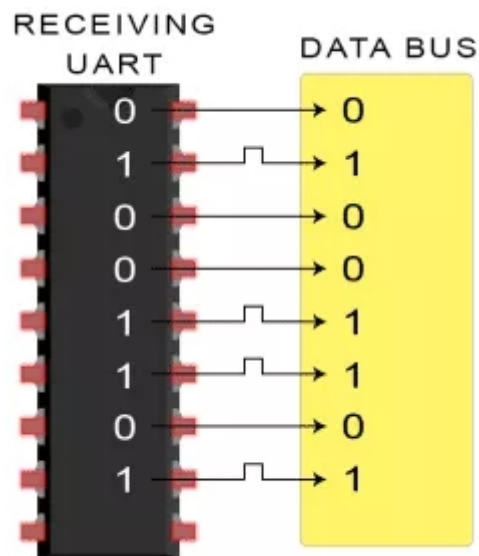
4.接收 UART 丢弃数据帧中的起始位，奇偶校验位和停止位：

RECEIVING UART



5.接收 UART 将串行数据转换回并行数据，并将其传输到接收端的数据总线：

优劣没有任何通信协
议是完美的，但是



UART 非常擅长于其工作。以下是一些利弊，可帮助您确定它们是否适合您的
项目需求：

优点

- 仅使用两根电线
- 无需时钟信号
- 具有奇偶校验位以允许进行错误检查
- 只要双方都设置好数据包的结构

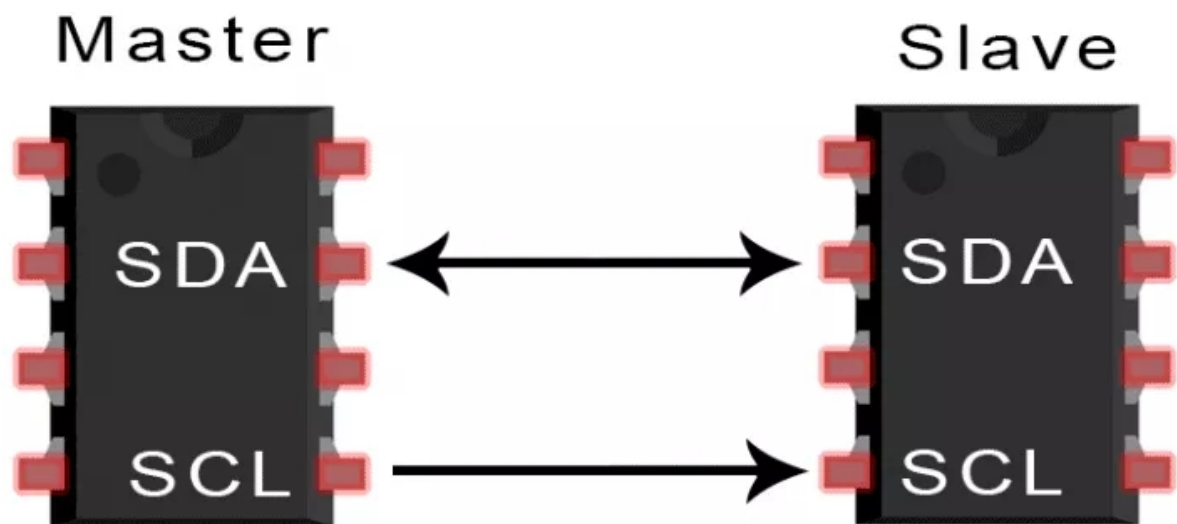
- 有据可查并得到广泛使用的方法

缺点

- 数据帧的大小最大为 9 位
- 不支持多个从属系统或多个主系统
- 每个 UART 的波特率必须在彼此的 10%之内

I2C 通信

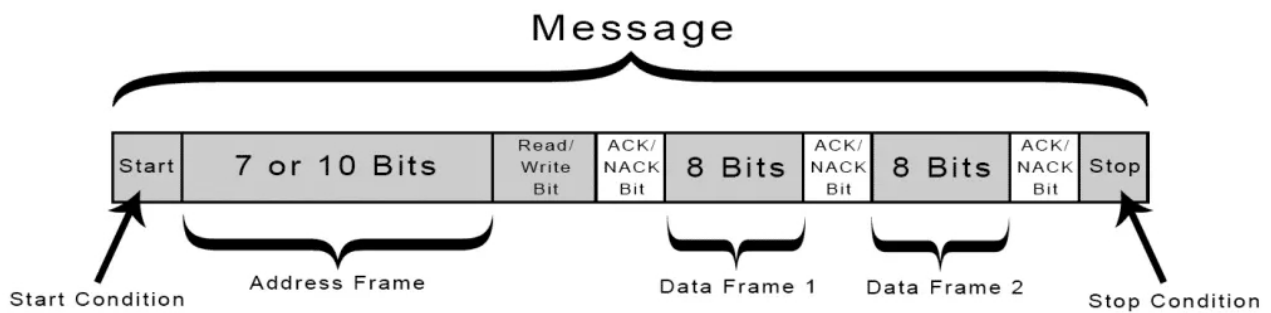
I2C 总线是由 Philips 公司开发的一种简单、双向二线制同步串行总线。它只需要两根线即可传送信息。它结合了 SPI 和 UART 的优点，您可以将多个从机连接到单个主机（如 SPI 那样），也可以使用多个主机控制一个或多个从机。当您想让多个微控制器将数据记录到单个存储卡或将文本显示到单个 LCD 时，这将非常有用。



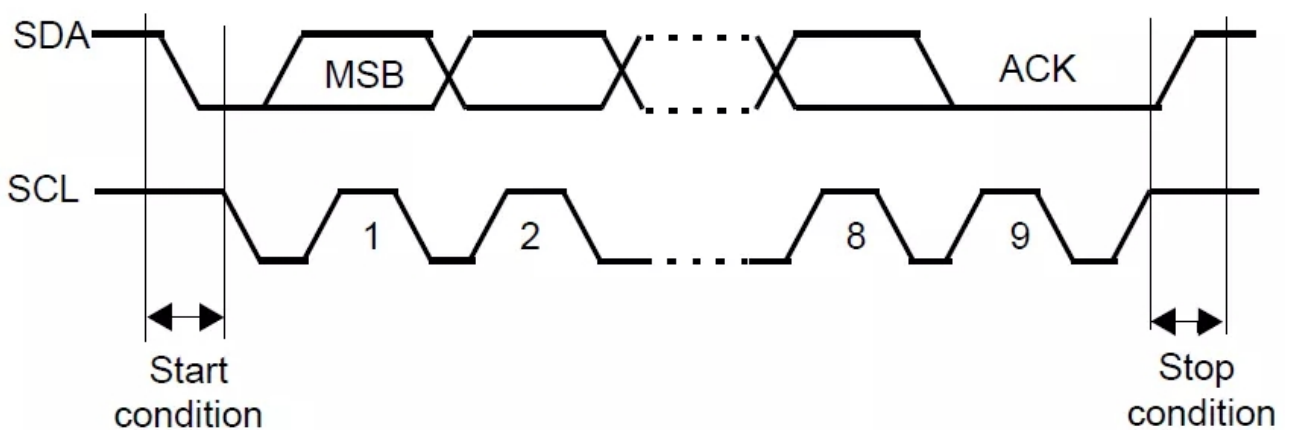
SDA (Serial Data) - 数据线。SCL (Serial Clock) - 时钟线。I2C 是串行通信协议，因此数据沿着 SDA 一点一点地传输。与 SPI 一样，I2C 也需要时钟同步信号且时钟始终由主机控制。

Wires Used	2
Maximum Speed	Standard mode= 100 kbps
	Fast mode= 400 kbps
	High speed mode= 3.4 Mbps
	Ultra fast mode= 5 Mbps
Synchronous or Asynchronous?	Synchronous
Serial or Parallel?	Serial
Max # of Masters	Unlimited
Max # of Slaves	1008

工作原理 I2C 的数据传输是以多个 msg 的形式进行，每个 msg 都包含从机的二进制地址帧，以及一个或多个数据帧，还包括开始条件和停止条件，读/写位和数据帧之间的 ACK / NACK 位：



启动条件：当 SCL 是高电平时，SDA 从高电平向低电平切换。**停止条件：**当 SCL 是高电平时，SDA 由低电平向高电平切换。



地址帧：每个从属设备唯一的 7 位或 10 位序列，用于主从设备之间的地址识别。

读/写位：一位，如果主机是向从机发送数据则为低电平，请求数据则为高电平。

ACK/NACK：消息中的每个帧后均带有一个 ACK/NACK 位。如果成功接收到地址帧或数据帧，接收设备会返回一个 ACK 位用于表示确认。

寻址

由于 I2C 没有像 SPI 那样的片选线，因此它需要使用另一种方式来确认某一个从设备，而这个方式就是 —— 寻址。主机将要通信的从机地址发送给每个从机，然后每个从机将其与自己的地址进行比较。如果地址匹配，它将向主机发送一个低电平 ACK 位。如果不匹配，则不执行任何操作，SDA 线保持高电平。

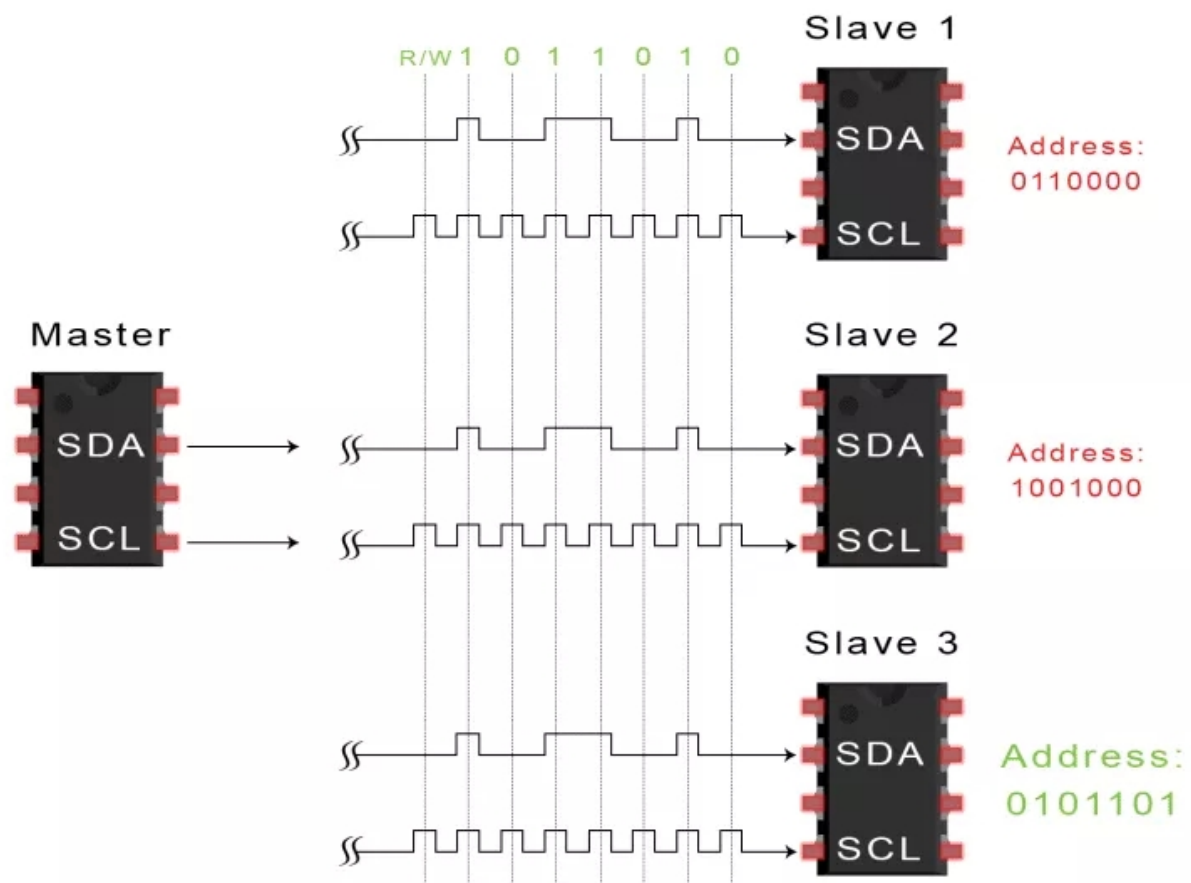
读/写位

地址帧的末尾包含一个读/写位。如果主机要向从机发送数据，则为低电平。如果是主机向从机请求数据，则为高电平。

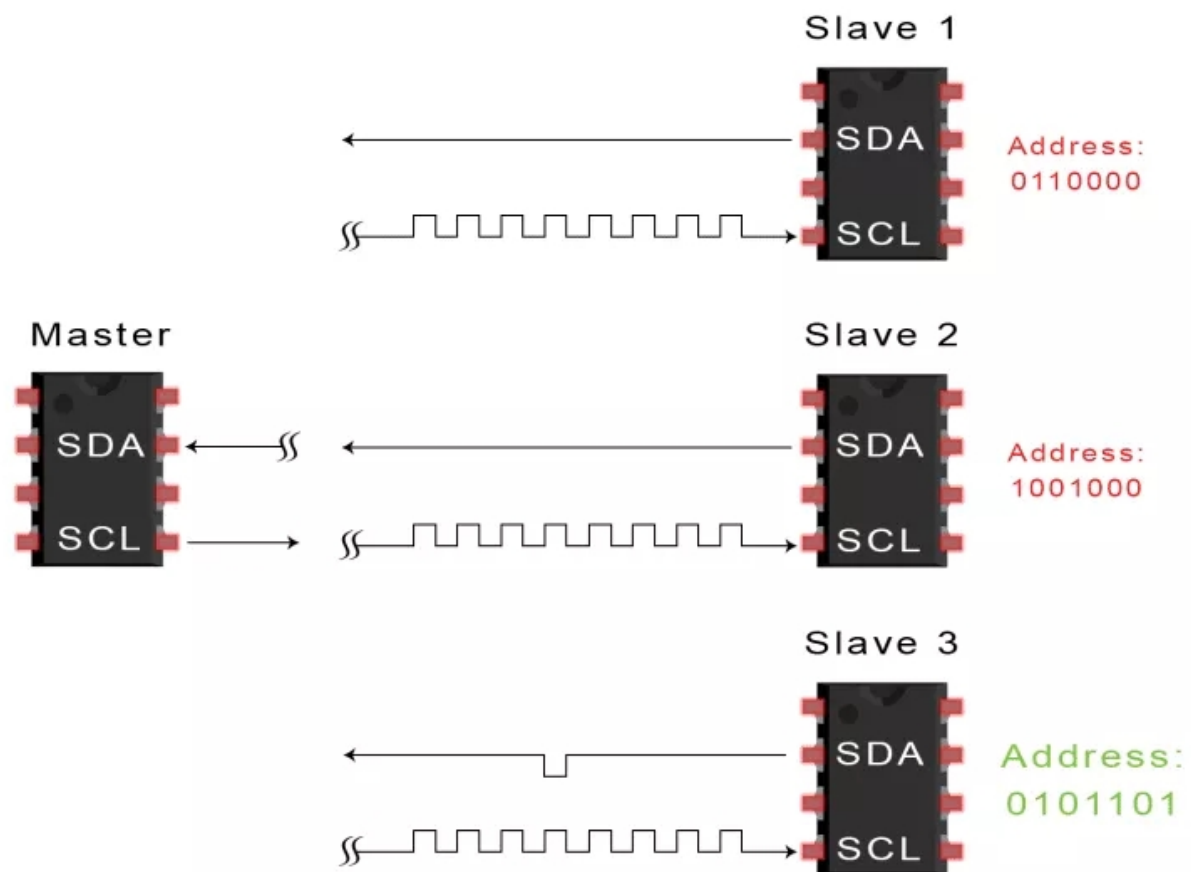
数据帧

当主机检测到从机的 ACK 位后，就可以发送第一个数据帧了。数据帧始终为 8 位，每个数据帧后紧跟一个 ACK / NACK 位，来验证接收状态。当发送完所有数据帧后，主机可以向从机发送停止条件来终止通信。**传输步骤** 1. 在 SCL 线为高电平时，主机通过将 SDA 线从高电平切换到低电平来启动总线通信。

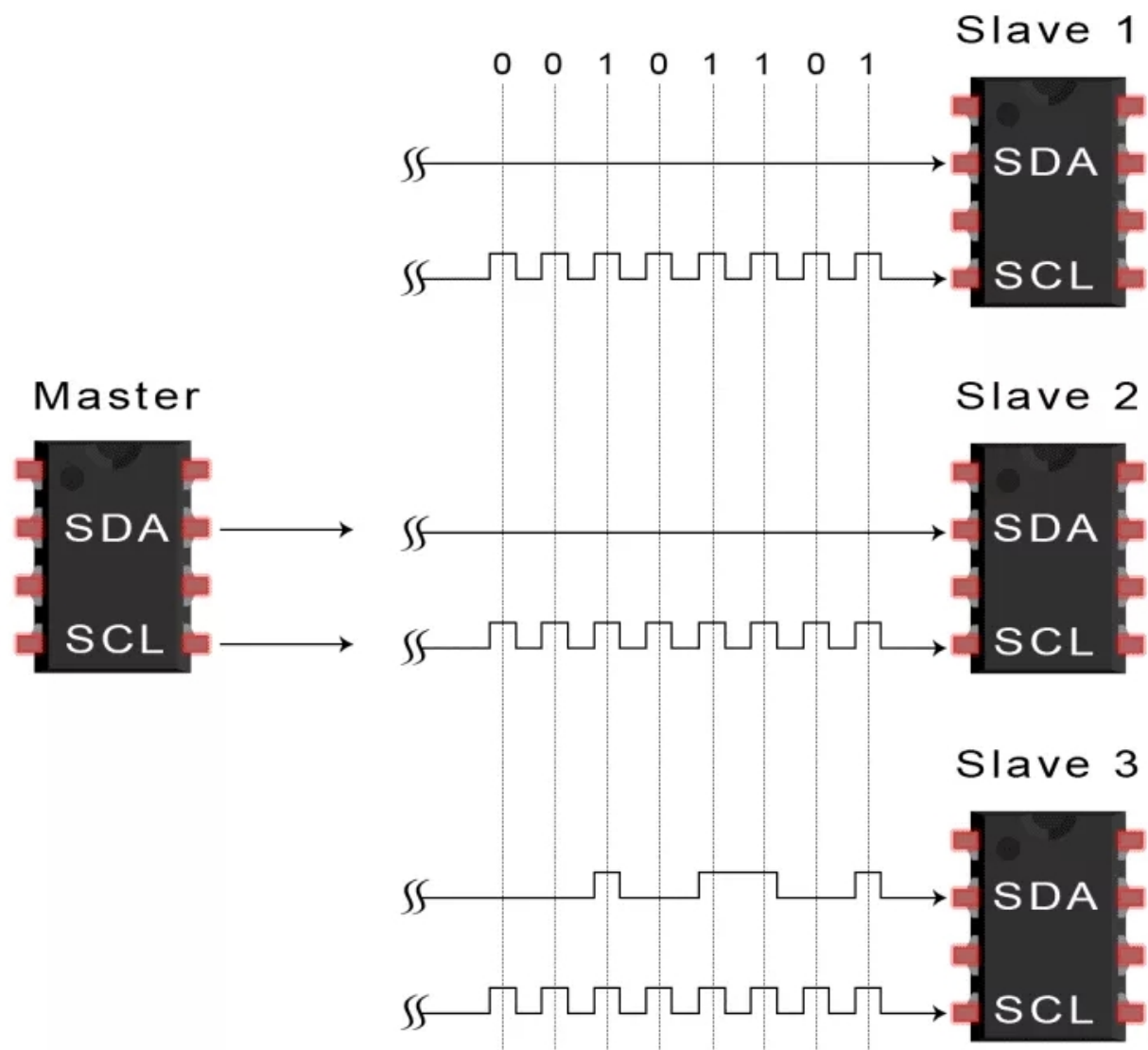
2. 主机向总线发送要与之通信的从机的 7 位或 10 位地址，以及读/写位：



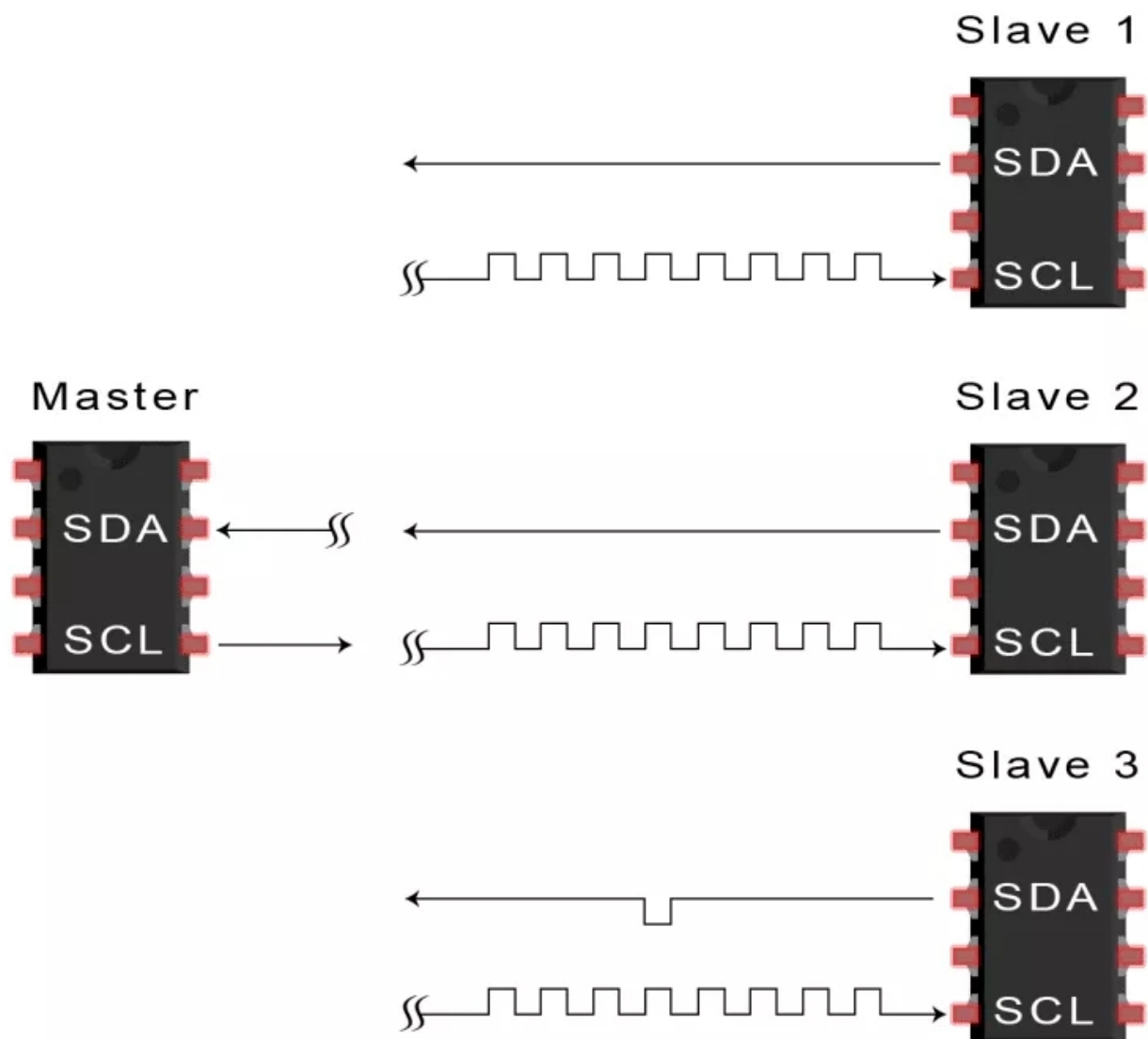
3. 每个从机将主机发送的地址与其自己的地址进行比较。如果地址匹配，则从机通过将 SDA 线拉低一位返回一个 ACK 位。如果主机的地址与从机的地址不匹配，则从机将 SDA 线拉高。



4. 主机发送或接收数据帧:



5. 传输完每个数据帧后，接收设备将另一个 ACK 位返回给发送方，以确认已成功接收到该帧：

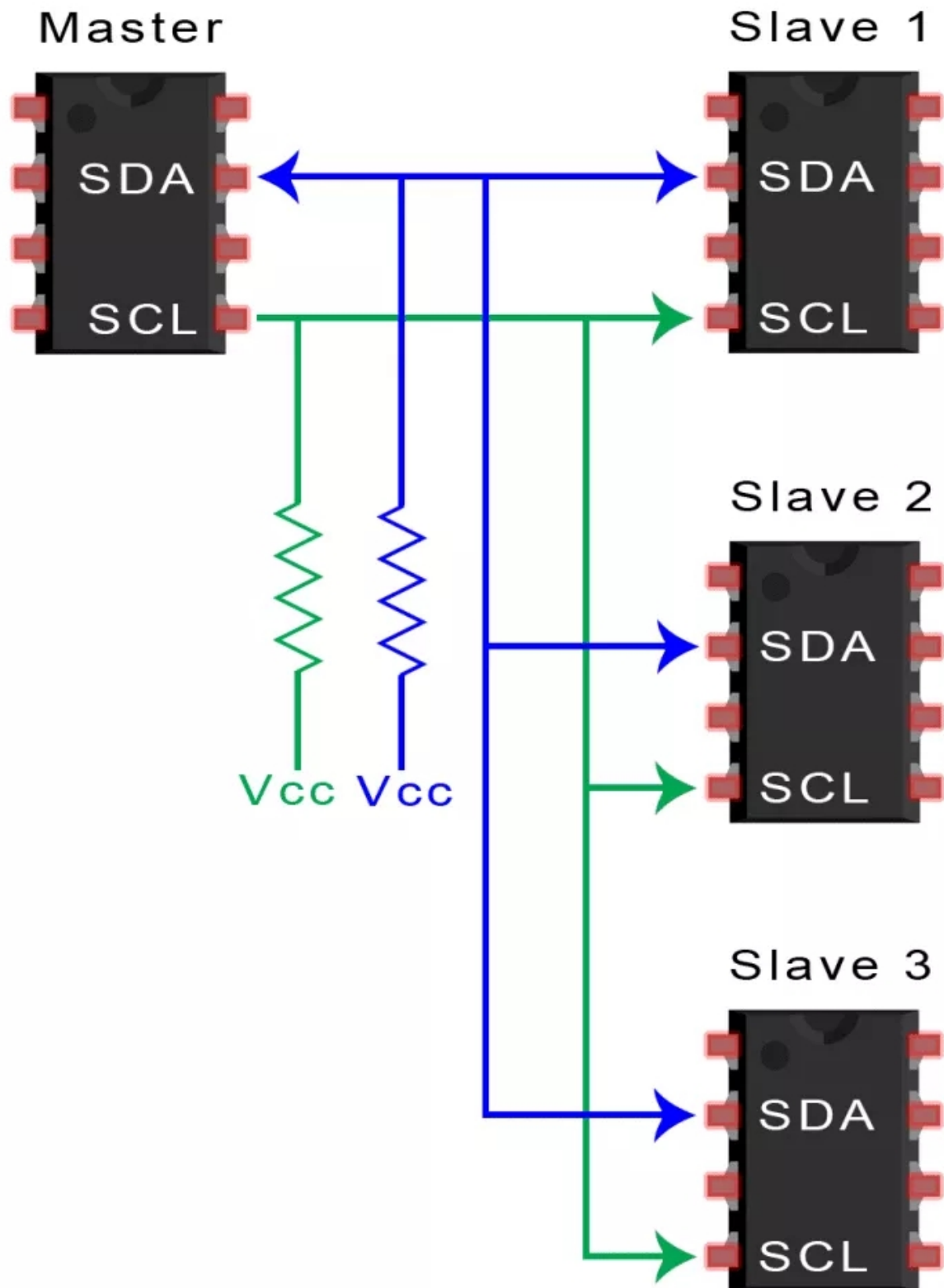


6. 随后主机将 SCL 切换为高电平，然后再将 SDA 切换为高电平，从而向从机发送停止条件。

单个主机 VS 多个从机

由于 I2C 使用寻址功能，可以通过一个主机控制多个从机。使用 7 位地址时，最多可以使用 128(2⁷)个唯一地址。使用 10 位地址并不常见，但可以提供

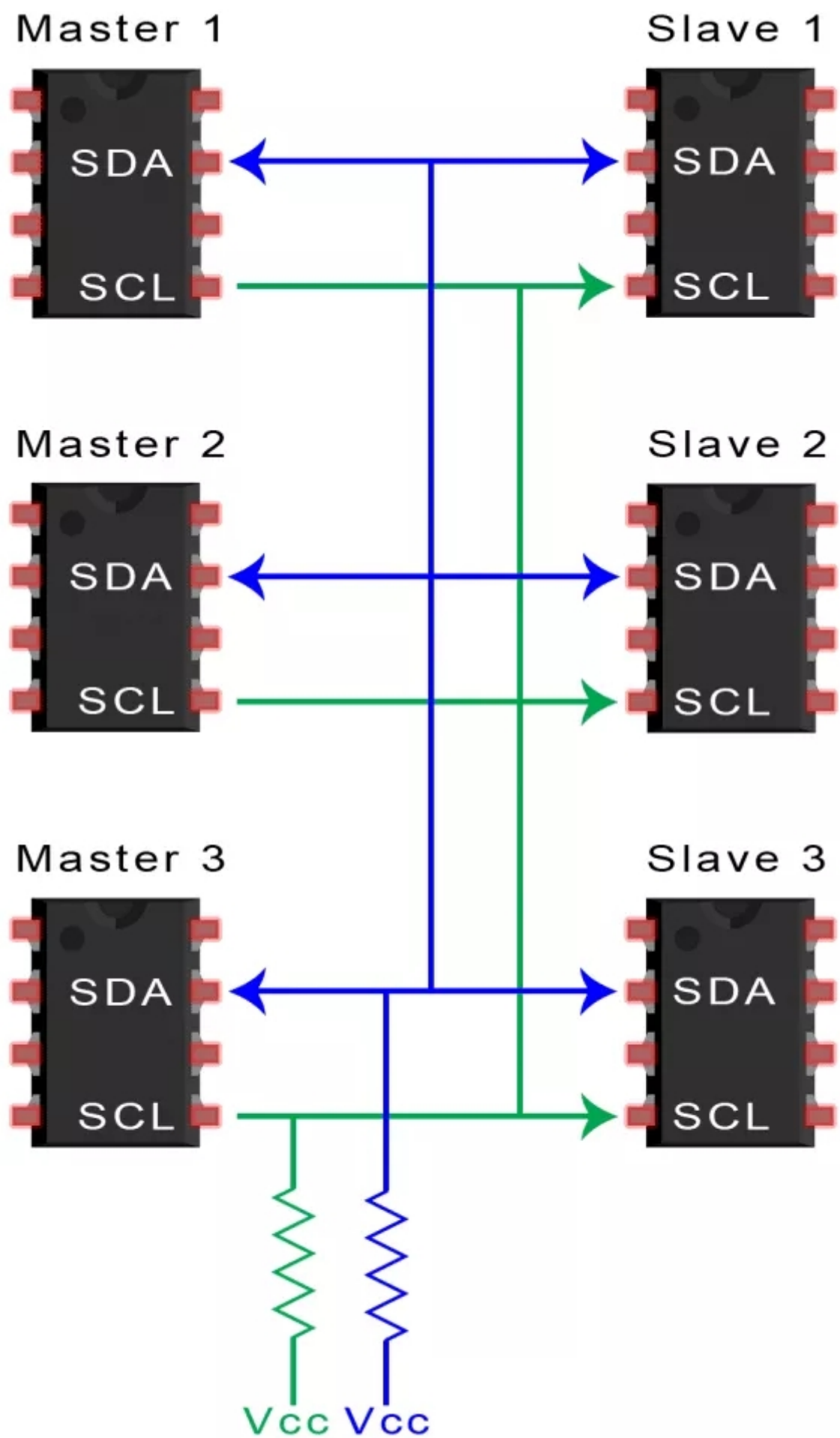
1,024(2¹⁰)个唯一地址。如果要将多个从机连接到单个主机时，请使用 4.7K 欧的上拉电阻将它们连接，例如将 SDA 和 SCL 线连接到 Vcc：



多个主机 VS 多个从机

I2C 支持多个主机同时与多个从机相连，当两个主机试图通过 SDA 线路同时发送或接收数据时，就会出现冲突。因此每个主机都需要在发送消息之前检测

SDA 线是低电平还是高电平。如果 SDA 线为低电平，则意味着另一个主机正在控制总线。如果 SDA 线高，则可以安全地发送数据。如果要将多个主机连接到多个从机，请使用 4.7K 欧的上拉电阻将 SDA 和 SCL 线连接到 Vcc：



优劣与其他协议相比，I2C 可能听起来很复杂。以下是一些利弊，可帮助您确定它们是否适合您的项目需求：**优点**仅使用两根电线支持多个主机和多个从机
每个 UART 的波特率必须在彼此的 10% 之内硬件比 UART 更简单众所周知且被广泛使用的协议**缺点**数据传输速率比 SPI 慢数据帧的大小限制为 8 位