

摘要

嵌入式系统和电子系统使用串行接口在连接的器件之间传输数据。串行外围接口 (SPI) 是一种串行通信接口，它在微控制器 (MCU) 和一个或多个外围器件之间提供数据的同步传输。在 SPI 协议中，MCU 生成时钟信号 (SCLK)、选择信号 (nSCS) 和串行数据输出 (SDO) 信号 (例如传输到外围器件的数据)。外围器件通过串行数据输入 (SDI) 引脚接收数据信号。当选择信号激活时，从 MCU SDO 到外围 SDI 的信号数据会与时钟信号同步。外围器件还可以通过其自身向 MCU 中 SDI 的 SDO 输出将数据发送回 MCU。来自外围器件的这些输出也与时钟信号同步。SPI 是汽车应用中一种常见的接口形式，用于提高电子部件的灵活性、可配置性和故障报告能力。本报告描述了使用具有菊花链功能的 SPI 与多个外围器件 (例如电机驱动器器件) 进行同步串行通信的方法。

1 商标

所有商标均为其各自所有者的财产。

内容

1 商标.....	1
2 功能概述.....	2
3 实现详细信息.....	3
4 应用示例.....	5
5 总结.....	8
6 术语表.....	9
6.1 本文中使用的命名规则.....	9

插图清单

图 2-1. 以菊花链连接的三个电机驱动器器件.....	2
图 3-1. 带三个电机驱动器 (外围) 器件的 SPI 帧.....	3
图 3-2. 标头字节.....	3
图 3-3. 标头、状态、地址和数据字节的内容.....	4
图 3-4. MCU 和三个电机驱动器器件之间的 SPI 数据序列.....	4
图 4-1. 20 个器件的 SPI 菊花链实现示例.....	5
图 4-2. 20 个器件的 SPI 菊花链数据帧.....	5
图 4-3. 63 个器件的 SPI 菊花链方框图.....	6
图 4-4. 63 个器件的 SPI 菊花链数据帧.....	6
图 4-5. DRV8889-Q1 数据表中的 SPI 帧时序图.....	7
图 4-6. 用于控制 63 个以上器件的 SPI 方框图.....	7

2 功能概述

图 2-1 显示了菊花链配置，以在多个器件通过共享 SPI 总线与同一 MCU 进行通信时保持 GPIO 端口可用。MCU 必须配置为根据串联连接的器件数量生成芯片选择信号、报头信号和多个地址字段。此外，MCU 还生成时钟信号，使数据能够通过链同步传输。根据图 2-1，MCU 通过 SDO1 发送信号，该信号由链中的每个器件解码，并执行相应的命令。此处介绍了关于解码方法的详细信息。

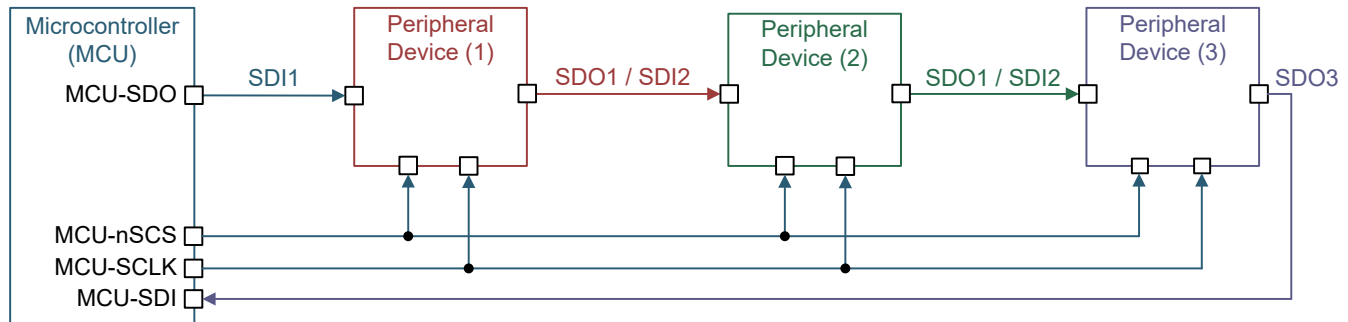


图 2-1. 以菊花链连接的三个电机驱动器器件

芯片选择信号 (MCU-nSCS) 定义了一个帧间隔，用于提供给电机驱动器器件。时钟信号用于控制 MCU 和电机驱动器器件之间串行数据的同步传输。MCU-SDO 是从 MCU 发送到链中第一个电机驱动器器件的数据，M-SDI 是从链中最后一个电机驱动器器件接收的数据。MCU-SDO 由标头字节、地址字节和数据字节组成。由链中最后一个电机驱动器器件生成的 MCU-SDI 由状态字节、标头字节 (与 MCU-SDO 相同) 和报告字节组成。

从 MCU 到电机驱动器器件，标头字段是在帧间隔中传输的第一个字段，并且指定以通信方式耦合到 MCU 的多个外围器件。标头字段后跟地址字段。多个地址字段将在帧间隔内传输。每个地址字段对应于不同的电机驱动器器件。由 MCU 在帧间隔中发送的第一个地址字段对应于接收标头字段的最后一个电机驱动器器件，反之亦然。地址字段后跟数据字段，这些数据字段携带要在给定电机驱动器器件中执行的信息。

3 实现详细信息

传统的菊花链配置允许减少微控制器上的终端数量，但限制了通信带宽。例如，在一些实现方案中，从外围器件读取则需要两个事务（两个通信帧）。在允许单帧读取的系统中，通过串行连接的外围器件的信息传输速度随着外围器件数量的增加而降低。

此处介绍的同步串行通信系统采用菊花链配置来减少微控制器上所需的终端数量，允许在单个帧中读取，并提供不受链中外围器件数量影响的传输比特率。一些实现方案在不引入额外协议开销的情况下使用单个外围器件提供操作。

图 2-1 显示了根据这部分所述菊花链实现的同步串行通信系统示例的方框图。SPI 系统包括一个 MCU 和一个或多个串联的电机驱动器器件，以与 MCU 进行通信。

例如，MCU 输出数据信号，并且链中的第一个电机驱动器器件接收数据信号 (SDI1)。在执行任何相关操作之后，它沿着输出信号 SDO1/SDI2 传递，该输出信号包括经由 SDI1 提供的部分数据。在整个链中重复相同的时序控制，直至到达最终的电机驱动器器件。

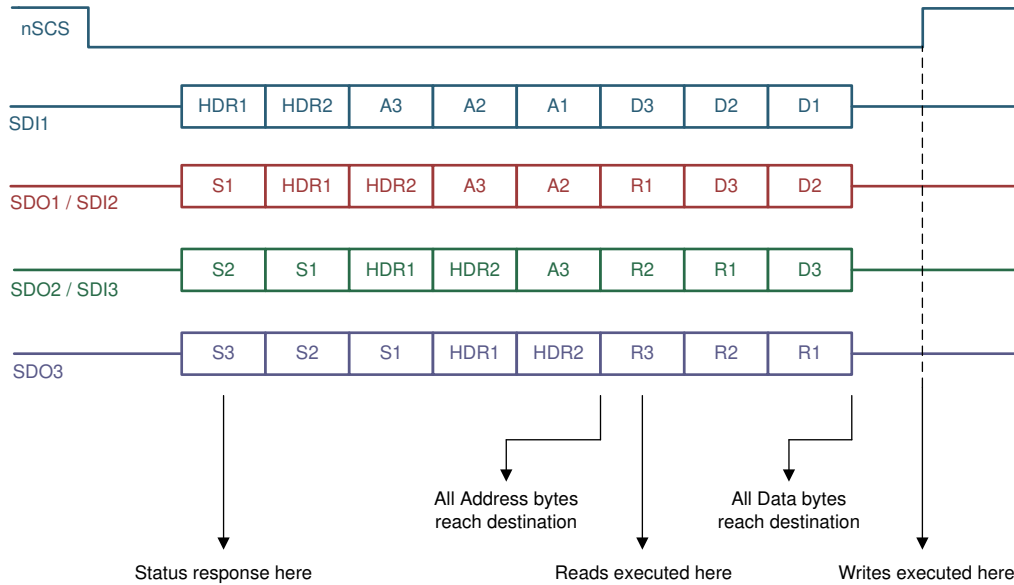


图 3-1. 带三个电机驱动器（外围）器件的 SPI 帧

图 3-1 显示了 MCU 与三个电机驱动器器件之间的数据传输和接收结构。链中的第一个器件按以下格式从 MCU 接收数据，以进行三器件配置：2 字节标头 (HDRx) 后跟 3 字节地址 (Ax) 后跟 3 字节数据 (Dx)。通过链中传送数据后，MCU 会按以下格式接收数据字符串，以进行三器件配置：3 字节状态 (Sx) 后跟 2 字节标头后跟 3 字节报告 (Rx)。

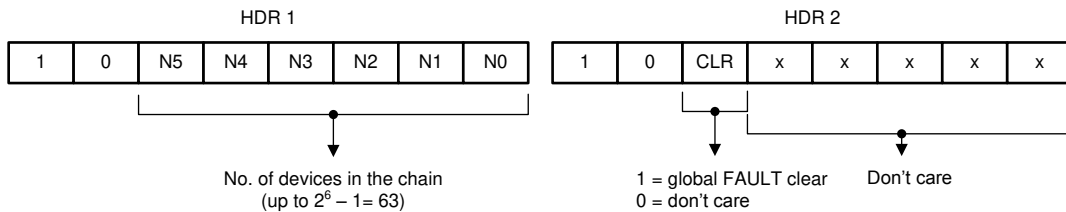


图 3-2. 标头字节

图 3-2 中显示的标头字节 (HDRx) 包含链中连接的器件数量信息，以及一个全局清除故障命令，该命令将清除芯片选择 (nSCS) 信号上升沿上所有器件的故障寄存器。对于两个 MSB，标头字节必须以 1 和 0 开头。标头识别值被布置在标头字段的开头，因此是由 MCU 发送的第一个值，该值由每个电机驱动器器件按照由其在链中的位置而确定的顺序接收。标头值 N5 到 N0 是 6 位，专用于显示链中的器件数。对于每个菊花链连接，最多可串行连接 63 个器件。HDR2 寄存器的 5 个 LSB 不用考虑位，MCU 可以使用这些位来确定菊花链连接的完整性。

地址字段 (A_x) 是来自 MCU 的地址字段的一种实现方式, 需要针对特定电机驱动器器件访问该地址字段。地址字段包含识别值 0, 此值将该字节标识为地址字段。它还包含读/写控制值 (R/W 位) 和字节中的地址值。识别值指定是读取还是写入与地址值相对应的电机驱动器器件的位置。例如, 如果 R/W 位设置为逻辑 1, 则将读取地址对应的地址值。如果 R/W 位为逻辑 0, 则与地址值对应的地址将与数据字段中的数据一同被写入。

数据字段 (D_x) 指定要写入电机驱动器器件地址值的值。例如, 在接收数据字段的帧结束时, 电机驱动器器件会将数据字段中包含的值写入对应于电机驱动器器件的地址字段中指定的地址。

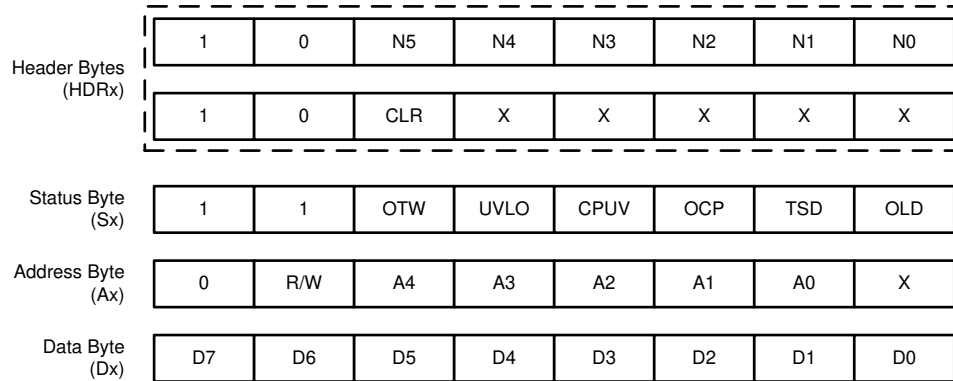


图 3-3. 标头、状态、地址和数据字节的内容

当芯片选择信号激活时, 电机驱动器器件开始接收时钟信号, 并且在每个时钟脉冲时, 电机驱动器器件都会发送状态值。例如, 当电机驱动器器件接收标头字段时, 同时也在发送状态位, 该状态位为菊花链中的每个器件提供关于故障状态寄存器的信息, 因此 MCU 不必重新启动另一个读取命令来从给定电机驱动器器件读取故障状态。这减少了从 MCU 读取命令的数量, 并使系统更有效地确定器件标记的故障条件。

状态字段包含标识值和状态值。对于两个 MSB, 标识值必须以 1 和 1 开头。状态字段包含生成状态字节的电机驱动器器件的全局故障位标识。在图 3-3 中, 显示的全局故障位以 DRV8873-Q1 器件为例。这六个全局故障位位于两个识别位之后, 根据电机驱动器器件的不同而不同。

当数据通过器件时, 它通过计算接收到的总字节数后跟第一个标头字节来确定自身在链中的位置。例如, 在这种三器件配置中, 链中的器件 2 在接收 HDR1 字节之前先接收两个状态字节, 然后再接收 HDR2 字节。根据这两个状态字节, 电机驱动器器件可以确定其在链中的位置为第二位。根据 HDR2 字节, 每个器件可以确定链中连接了多少个器件。这样, 数据仅将相关的地址和数据字节加载到其缓冲区中, 并绕过其他位。该协议可实现更快的通信, 而不会因为链中连接多达 63 个器件而增加系统延迟。图 3-3 显示了状态字节的编码。

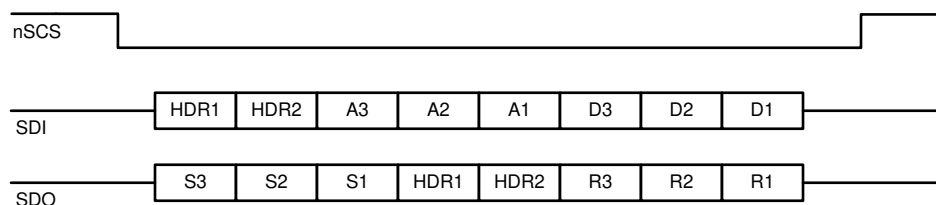


图 3-4. MCU 和三个电机驱动器器件之间的 SPI 数据序列

图 3-4 显示了链中连接的外围器件的 MCU 和电机驱动器器件之间的数据序列。头字节数始终保持为 2, 但其他字节 (状态、地址、数据和报告) 将根据串联器件的数量进行缩放。这种菊花链配置还允许 MCU 控制两个或多个电机驱动器器件, 而无需更改 SDI 数据结构。如果 MCU 仅控制链中的一个器件, 则不需要标头字节和菊花链配置。特定器件数据表介绍了与单个外围器件进行通信的应用案例。

4 应用示例

以下部分包含使用菊花链配置的其他应用示例。以下列表概述了系统设计过程中需要考虑的一些要点：

- 必须为每个 SPI 事务写入或读取整个器件链。
- HDRx 字节必须始终包含链中的器件总数。每个器件都统计时钟周期数，直到接收到标头字节。这就是每个器件查找链中器件的数量及其相对位置的方法。
- 每个 SPI 事务只能读取或写入每个器件的一个寄存器。

中链写入器件

在此示例中，链中有 20 个器件，但微控制器只需要写入第 10 个器件，如图 4-1 中所示。

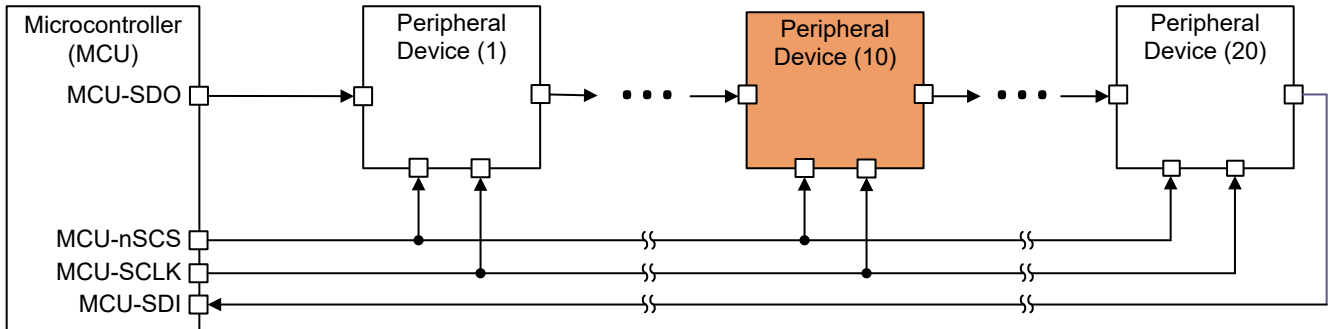


图 4-1. 20 个器件的 SPI 菊花链实现示例

如果没有其他器件需要更新，则地址字节 (A1-A9 和 A11-A20) 中的 R/W 位可以设置为“读取”。A10 的 R/W 位将设置为“写入”。按照 A10 字节的选择，每个事务将只在器件 10 中写入一个寄存器。写入该地址位置的数据以 D10 字节发送。如果微控制器需要更新器件 10 中的多个地址，它将需要为每个附加地址更新发送整个 SPI 事务。图 4-2 显示了写入器件 10 的每个地址的 SPI 事务示例。

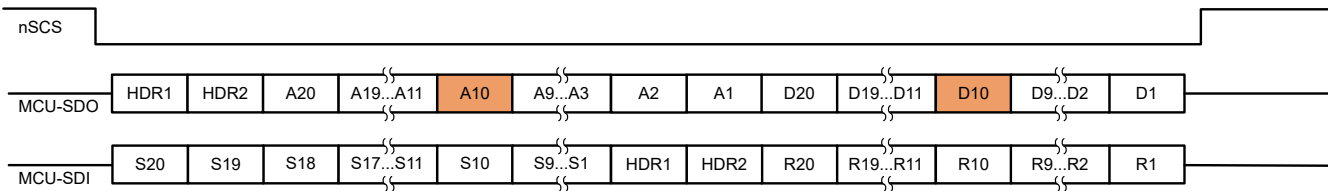


图 4-2. 20 个器件的 SPI 菊花链数据帧

在每个 SPI 事务中，微控制器可以在每个器件中读取或写入一个寄存器。如果多个器件需要更新，则微控制器可以将这些地址中的 R/W 位设置为“写入”。

每个 SPI 事务的时间长度

有时，SPI 事务时间可能是具有长菊花链的应用中的一个关键约束。这部分计算每个事务所需的时间量。图 4-3 显示了此应用示例的方框图。表 4-1 显示了对示例应用的约束。图 4-4 显示了此示例的 SPI 事务数据帧。

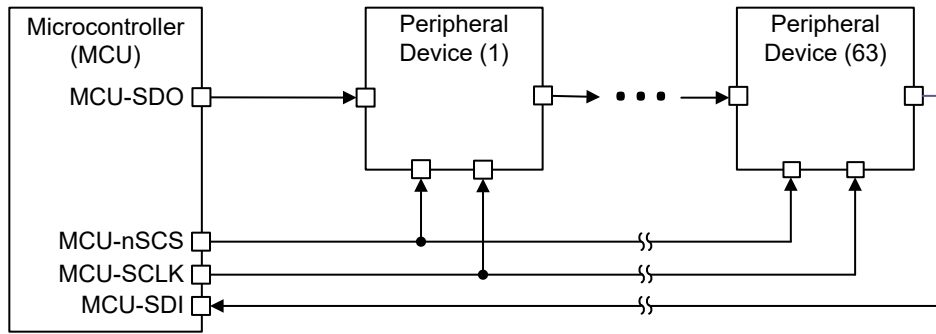


图 4-3. 63 个器件的 SPI 菊花链方框图

表 4-1. SPI 定时计算参数

参数	说明	示例值
N_{devices}	器件数	63
f_{CLK}	SPI 时钟频率	5MHz (位/秒)

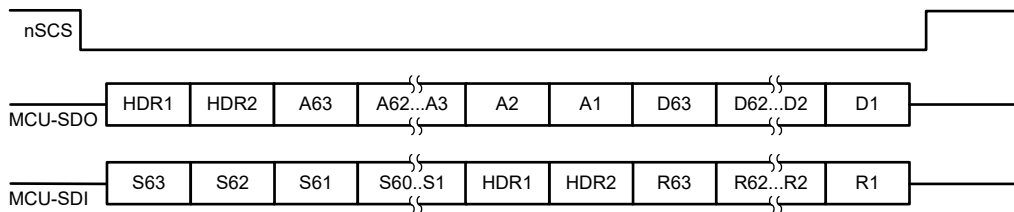


图 4-4. 63 个器件的 SPI 菊花链数据帧

在 SPI 事务中，传输 SPI 帧中的字节花费的时间最多。方程式 1 计算每个 SPI 事务中的总位数。方程式 2 计算链中所有数据所需的总时间。对于大多数 SPI 定时计算而言，方程式 2 应足够。

$$N_{\text{bits}} = N_{\text{HDR1}} + N_{\text{HDR2}} + N_{\text{ADDRESSES}} + N_{\text{DATA}} = 8 + 8 + 8 * N_{\text{devices}} + 8 * N_{\text{devices}} = 16 + 16 * 63 = 1,024 \text{ 位} \quad (1)$$

$$t_{\text{bits}} = N_{\text{bits}} / f_{\text{CLK}} = N_{\text{bits}} * (t_{\text{CLKH}} + t_{\text{CLKL}}) = 1,024 \text{ 位} / 5\text{MHz} = 0.2048\text{ms} \quad (2)$$

方程式 3 计算每个 SPI 帧所需的完整时长，包括初始 nSCS 设置时间 ($t_{\text{su(nSCS)}}$) 和最终 nSCS 保持时间 ($t_{\text{h(nSCS)}}$)。图 4-5 显示了 SPI 逻辑所需的其他 SPI 定时参数。公式中的具体数值为示例值。具体值取决于特定的外围器件，可在器件数据表中找到。

$$t_{\text{SPI_FRAME}} = t_{\text{bits}} + t_{\text{su(nSCS)}} + t_{\text{h(nSCS)}} = 0.2048\text{ms} + 100\text{ns} + 100\text{ns} = 0.2050\text{ms} \quad (3)$$

如果需要快速连续写入多个 SPI 帧，则必须在每个帧之间观察到 $t_{\text{HI_nSCS}}$ 和 $t_{\text{dis(nSCS)}}$ 。方程式 4 计算每个 SPI 事务的完整时长。

$$t_{\text{SPI_TRANSACTION}} = t_{\text{SPI_FRAME}} + t_{\text{HI_nSCS}} + t_{\text{dis(nSCS)}} = 0.2050\text{ms} + 600\text{ns} + 30\text{ns} = 0.20563\text{ms} \quad (4)$$

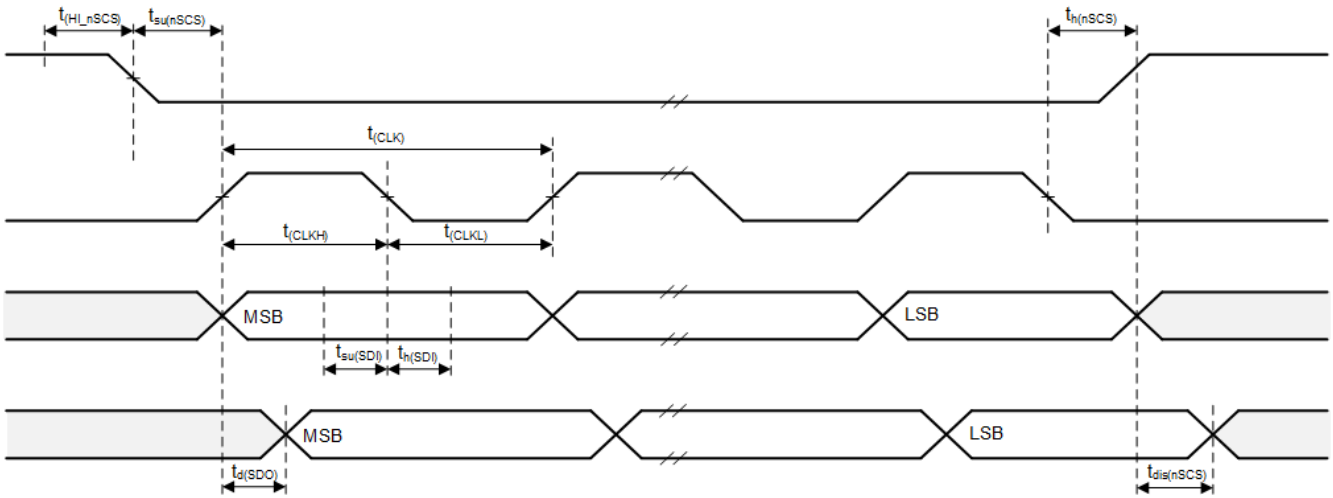


图 4-5. DRV8889-Q1 数据表中的 SPI 帧时序图

控制超过 63 个器件

为了控制 63 个以上的器件，微控制器需要为多条 nSCS 线路提供单独的 GPIO。这允许控制器支持多个菊花链，同时尽可能减少 SPI 通信所需的总 GPIO 引脚数。图 4 显示了这一点。若要与特定链进行通信，必须将特定 nSCS 信号拉低，而其他信号保持高电平。图 4-6 显示了这方面的一个示例。

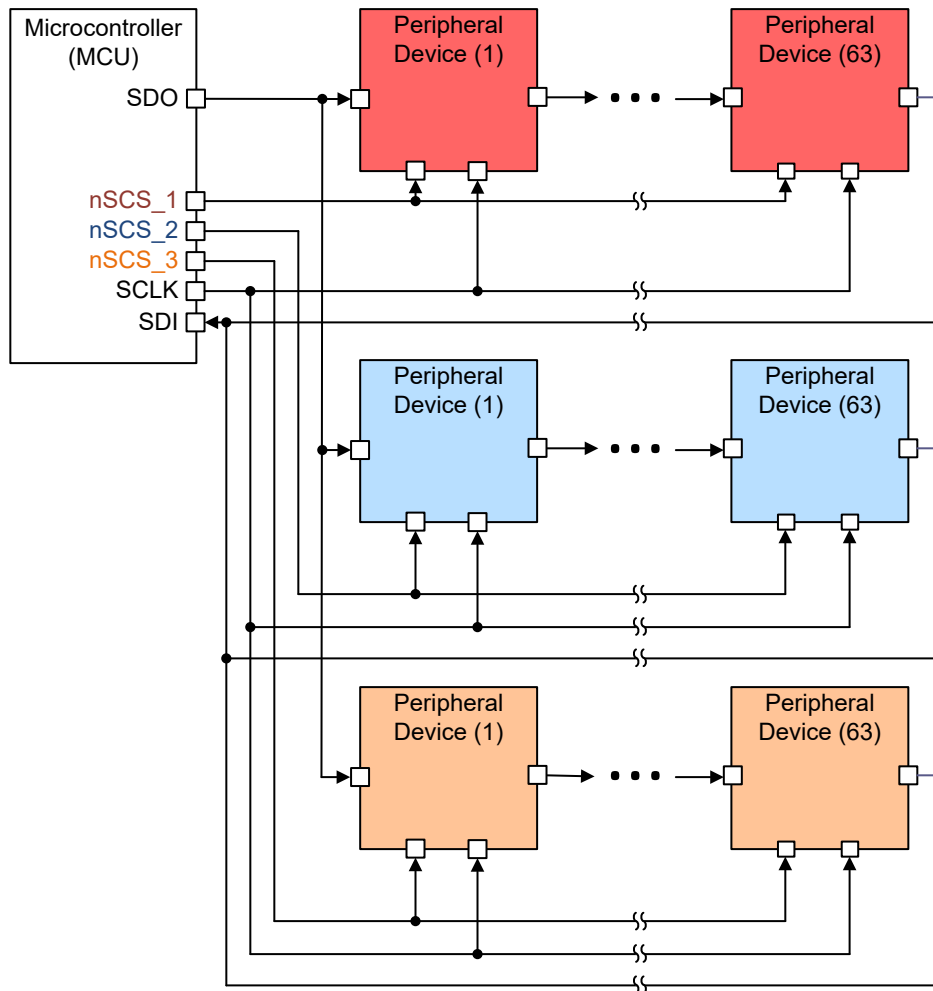


图 4-6. 用于控制 63 个以上器件的 SPI 方框图

5 总结

如本报告所述，实施菊花链具有某些优势。这些优势包括：

- 速度：大量 SPI 外围器件（多达 63 个）可以连接在单个 SPI 总线上，而无需降低 SPI 事务的频率。
- 相同的帧响应：对于读和写命令，每个电机驱动器件在其 SDO 线（MCU-SDI 线）上响应给定寄存器地址处的当前数据。这允许每个器件更新 MCU，而不需要额外的读取事务。
- 稳健性：微控制器发送的标头字节在经过整个链后返回微控制器。这允许微控制器持续检查链连接的完整性。

更多相关信息，请参阅以下数据表。

- [DRV8873-Q1 汽车类 H 桥电机驱动器 数据表](#)
- [具有高级诊断功能的 DRV89xx-Q1 汽车多通道半桥驱动器 数据表](#)
- [具有集成电流检测、1/256 微步进和失速检测功能的 DRV8889-Q1、DRV8889A-Q1 汽车步进驱动器 数据表](#)
- [具有集成电流检测和 1/256 微步进功能的 DRV8899-Q1 汽车步进驱动器 数据表](#)
- [具有集成电流检测、1/256 微步进、STEP 接口、智能调优技术和失速检测功能的 DRV8434S 步进电机驱动器 数据表](#)

6 术语表

6.1 本文档中使用的命名规则

本文档中使用了以下首字母缩略词和首字母缩写：

MCU	微控制器
SPI	串行外围接口
MSB	最高有效位
LSB	最低有效位
SDO	串行数据输出
SDI	串行数据输入
GPIO	通用输入输出

更全面的术语、首字母缩略词和定义列表，请参阅 [TI 术语表](#)。

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司