

Subsystem Design

UART 转 SPI 桥接器



1 说明

该子系统演示了如何实现将 MSPM0 器件作为通用异步接收器/发送器 (UART) 到串行外设接口 (SPI) 的桥接器。传入的 UART 数据包应采用特定格式，以便于 SPI 通信。本示例还能够确定错误条件并将其传回 UART 器件。本示例的代码位于 [MSPM0 SDK](#) 中。

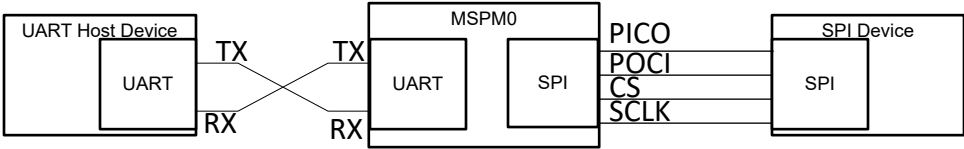


图 1-1. 系统功能方框图

2 所需外设

表 2-1. 所需外设

使用的外设	注释
UART	在代码中称为 UART_BRIDGE_INST
SPI	在代码中称为 SPI_0_INST

3 设计步骤

1. 在 SysConfig 中设置 SPI 模块。将器件置于控制器模式，并将其余设置保留为默认值。在 Advanced Configuration 选项卡中，确保将 RX FIFO 阈值水平设置为 *RX FIFO contains ≥ 1 entry*。确保将 TX FIFO 阈值级别设置为 *TX FIFO contains ≤ 2 entries*。现在导航到 Interrupt configuration 选项卡，并启用 Receive、Transmit、RX Timeout、Parity Error、Receive FIFO Overflow、Receive FIFO Full 和 Transmit FIFO Underflow 中断。
2. 在 SysConfig 中设置 UART 模块。将波特率设置为 9600。启用 Receive 中断。

4 设计注意事项

1. 在应用程序代码中，请确保根据应用程序的要求检查 SPI 和 UART 最大数据包大小。
2. 要提高 UART 波特率，请调整 SysConfig UART 选项卡中标记为 *Target Baud Rate* 的值。在此下方，观察计算得出的波特率变化以反映目标波特率。这可以使用可用的时钟和分频器进行计算。
3. 检查错误标志并进行适当处理。UART 和 I²C 外设都能够引发信息性错误中断。为了方便调试，该子系统在引发错误代码时使用枚举和全局变量来保存错误代码。在实际应用中，应在代码中处理错误，以确保错误不会使工程中断。
4. 工程的当前形式定义了数据包的所有格式化部分，例如 *UART_START_BYTE*、*UART_READ_SPI_BYTE* 和 *UART_WRITE_SPI_BYTE*。这些代码都附有定义，指定这些命令在数据包标头中的位置。可以更改本实现中的值。确保 UART 起始字节和读取/写入字节不是应用中预期的字节。

5 软件流程图

图 5-1 展示了本示例的代码流程图，并说明了不同的 UART 桥接器等待状态以及器件在每个状态下采取的操作。该流程图还展示了 UART 和 SPI 的中断服务例程。

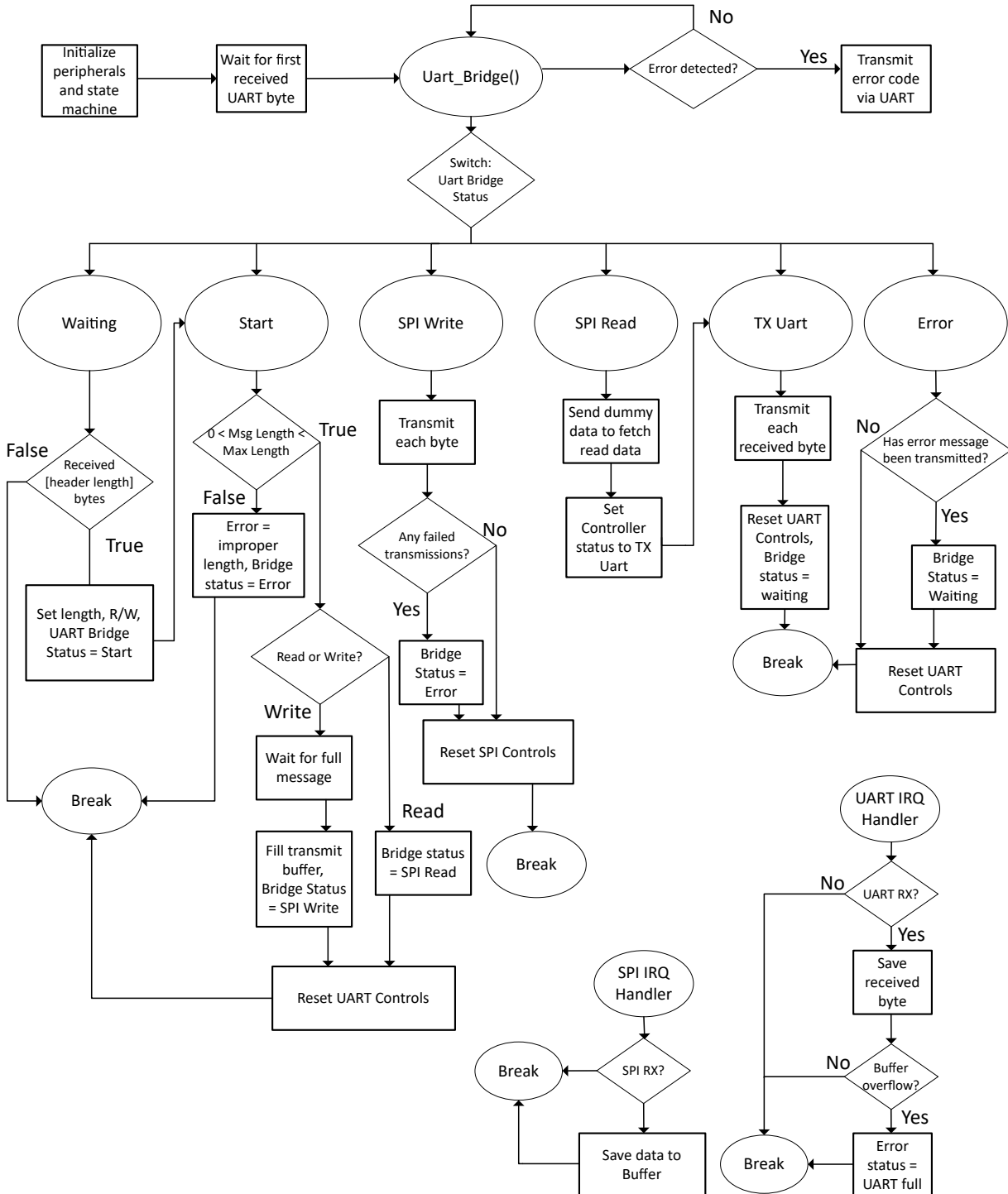


图 5-1. 软件流程图

5.1 器件配置

该应用利用 TI 系统配置工具 (SYSCONFIG) 图形界面来生成器件外设的配置代码。使用图形界面配置器件外设可简化应用原型设计过程。

软件流程图中描述的代码位于 `uart_to_spi_bridge.c` 文件中。

5.2 所需的 UART 数据包

图 5-2 展示了使用 SPI 执行读写操作所需的 UART 数据包。显示的值是示例中定义的默认标头值。

- **起始字节**：桥接器用来指示新事务开始的值。在桥接器检测到该值之前，UART 传输将被忽略。
- **SPI 读取或写入指示器**：此值会告知桥接器是对 SPI 器件执行读取还是写入操作。
- **消息长度 N**：传输的数据长度（以字节为单位）。
- **D0, D1, ..., D(N - 1)**：数据正在传输到桥接器

备注

读取数据包仅包含标头。进行读取时，不需要在数据包之后发送数据。桥接器器件会自动向 SPI 外设发送正确数量的虚拟数据，以获取读取的数据。

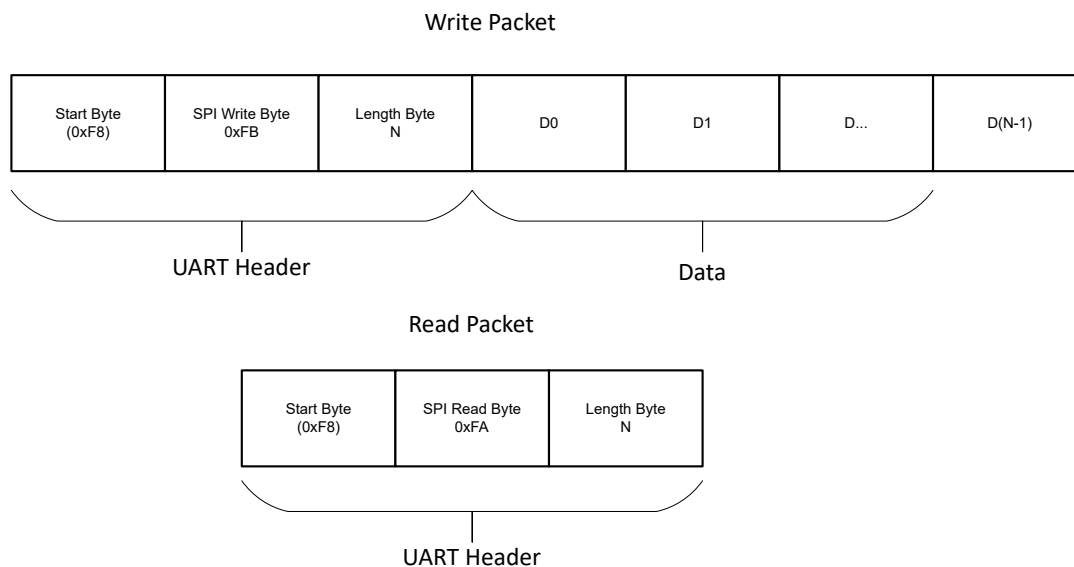


图 5-2. UART 写入和读取数据包格式

6 应用代码

某些用户希望更改 UART 数据包头使用的具体值，或更改最大数据包大小。这种更改可以通过修改 `uart_to_spi_bridge.c` 文件开头的 `#define` 值来实现，如以下代码所示。

```
/* Define UART Header and Start Byte*/
#define UART_HEADER_LENGTH 0x02
#define UART_START_BYTE 0xF8
#define UART_READ_SPI_BYTE 0xFA
#define UART_WRITE_SPI_BYTE 0xFB
#define RW_INDEX 0x00
#define LENGTH_INDEX 0x01

/*Define max packet sizes*/
#define SPI_MAX_PACKET_SIZE (16)
#define UART_MAX_PACKET_SIZE (SPI_MAX_PACKET_SIZE + UART_HEADER_LENGTH)
```

代码的许多部分旨在用于错误检测和处理。在代码中的这些点上，用户可以使用额外的错误处理或报告来实现更稳健的应用。例如，以下代码段演示了一种检查 SPI 传输中是否存在错误的方法，可在发生错误时设置错误标志。用户可以退出发送并更改此处的 UART 桥接器状态以反映错误。代码中的这一部分和许多其他部分都有考虑错误处理的选项。

```
for(int i = 0; i < gMsgLength; i++){
    if(!DL_SPI_transmitDataCheck8(SPI_0_INST, gSPIData[i])){
        gError = ERROR_SPI_WRITE_FAILED;
    }
}
```

7 其他资源

- 德州仪器 (TI), [下载 MSPM0 SDK](#)
- 德州仪器 (TI), [详细了解 SysConfig](#)
- 德州仪器 (TI), [MSPM0L LaunchPad™](#)
- 德州仪器 (TI), [MSPM0G LaunchPad™](#)
- 德州仪器 (TI), [MSPM0 SPI Academy](#)
- 德州仪器 (TI), [MSPM0 UART Academy](#)

8 E2E

请访问 [TI 的 E2E](#) 支持论坛来查看讨论并发布新主题，以获得在设计中使用 MSPM0 器件的技术支持。

9 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (January 2024) to Revision A (August 2025)	Page
• 删除了“兼容器件”部分	2

10 商标

LaunchPad™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

重要通知和免责声明

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。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
版权所有 © 2025，德州仪器 (TI) 公司