

Subsystem Design

UART 转 I2C 桥接器



说明

图 1-1 展示了如何使用 MSPM0 作为 I2C 控制器，从 UART 接口向多个目标 I2C 外设传输数据或命令。传入的 UART 数据包经过专门格式化，便于过渡到 I2C 通信。图 1-1 可将通信中的错误传回主机设备。此示例的代码可在 [UART 转 I2C 桥接器子系统代码](#) 中找到。

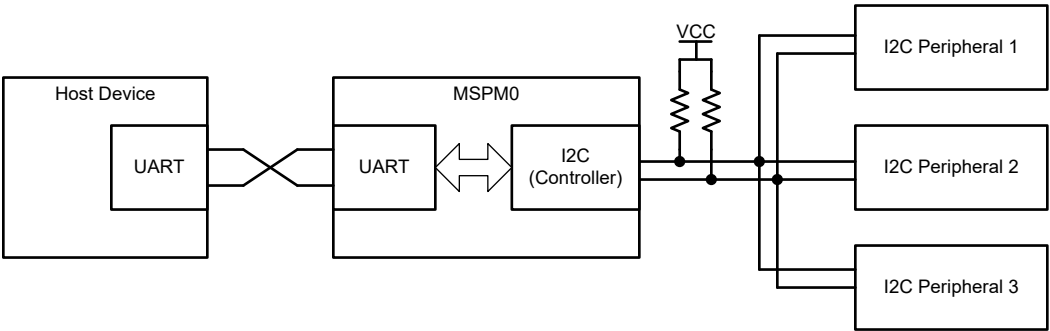


图 1-1. 子系统功能方框图

要求

应用此设计需要 UART 和 I2C 外设。

表 1-1. 所需外设

子块功能	外设使用	注释
UART TX/RX 接口	UART	在代码中称为 UART_Bridge_INST。默认波特率 9600。
I2C 控制器	I2C	在代码中称为 I2C_Bridge_INST。默认传输速率 100kHz。

设计步骤

1. 在 [Sysconfig](#) 中设置 UART 外设实例、I2C 外设实例和所需器件引脚的引脚输出。
2. 在 [SysConfig](#) 中设置 UART 波特率。默认波特率为 9600。
3. 在 [SysConfig](#) 中设置 I2C 时钟速度。默认值为 100kHz。
4. 定义桥接器处理的最大 I2C 数据包大小。
5. 定义关键的 UART 标头值 (可选)。
6. 自定义错误处理 (可选)。

设计注意事项

1. 通信速度。
 - a. 提高这两个接口的速度可增加数据吞吐量，减少数据冲突的可能性。
 - b. 如果 I2C 速度提高，则需要根据 I2C 规范调整外部上拉电阻以允许通信。
 - c. 以更高的速度重复的大数据包会影响整体系统性能。为提高桥接器利用率，有必要对该代码进行进一步优化。其他优化包括更高的器件运行速度、多个传输缓冲器、减小标头大小或简化状态机。

备注

[图 1-1](#) 示例仅使用 9600 波特 (UART) 和 100kHz (I2C) 的默认速度进行了测试。

2. UART 标头。
 - a. UART 数据包标头和起始字节值可针对您的应用进行自定义。德州仪器 (TI) 建议在典型数据传输开始时分配不太可能发生的值。
3. 错误处理。
 - a. 如果使用计算机终端监测 UART 总线，则将错误值对应于 ASCII 数值。
 - b. 确保主机 UART 器件可以读取错误值并知道相关含义，以便主机可以执行相应的操作。
 - c. 通过修改 *ErrorFlags* 结构类型添加其他错误类型，并在 *Uart_Bridge()* 中添加其他错误检测代码。
 - d. 当前实现检测有限的错误，并在 UART 接口上反馈相应的代码。然后，应用程序代码跳出当前通信状态机。用户可以添加额外的错误处理代码来更改发生错误时桥接器的行为。例如，发生 NACK 后重新发送 I2C 数据包。

备注

[图 1-1](#) 当前标记常见错误并为其分配数值，如 *ErrorFlags* 结构类型中定义。

软件流程图

图 1-2、图 1-3 和图 1-4 分别针对图 1-1 展示了 *UART* 桥接器主要功能、*Main()* 和 *UART ISR* 和 *I2C ISR* 的代码流程图。

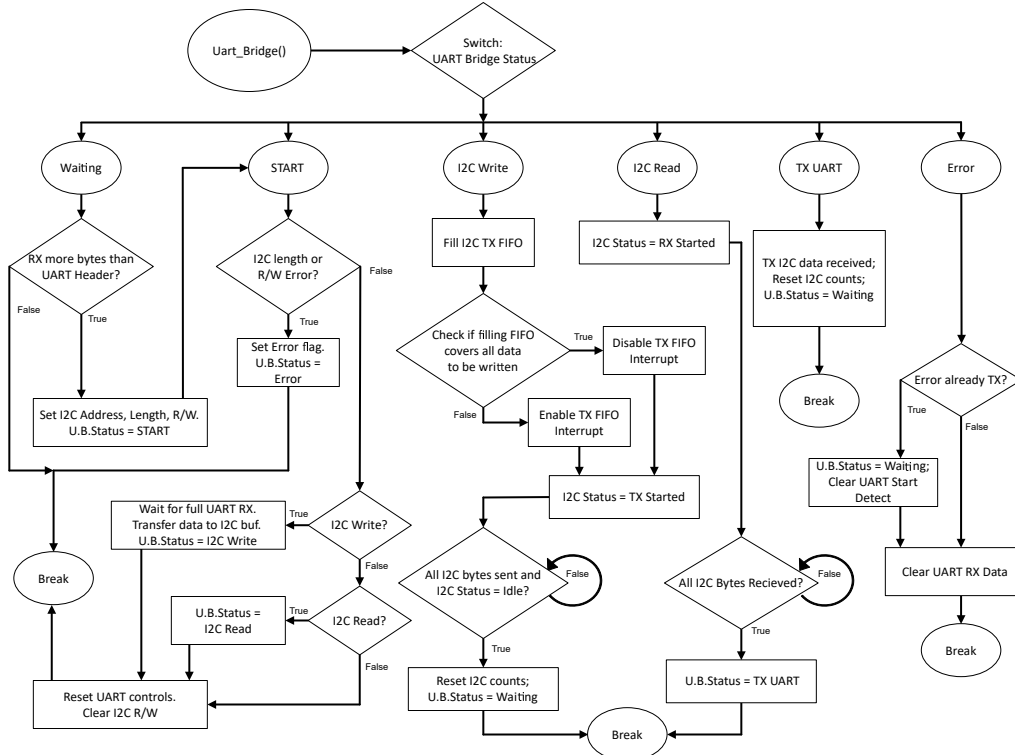


图 1-2. *Uart_Bridge()* 的软件流程图

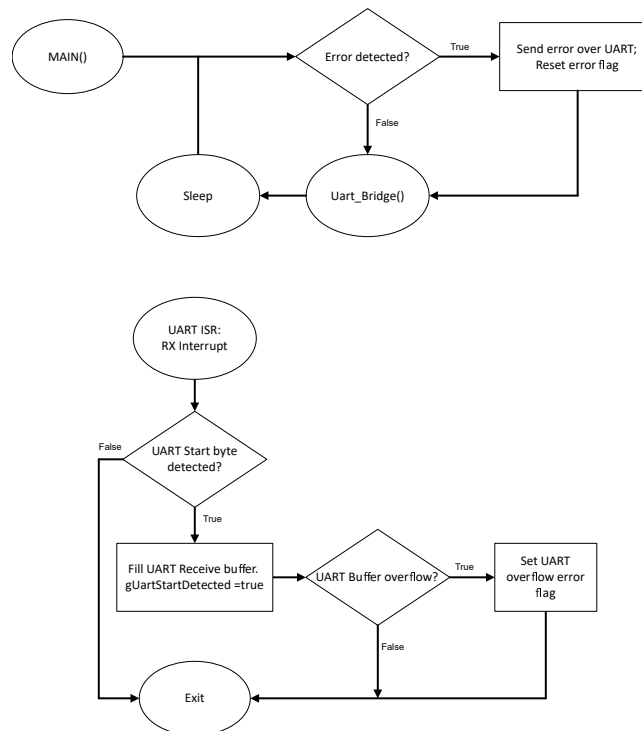


图 1-3. *MAIN* 循环和 *UART ISR* 的软件流程图

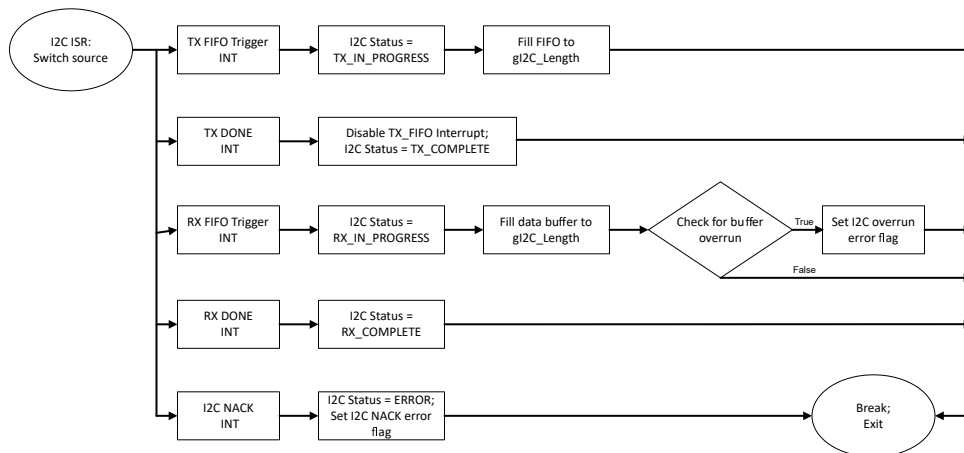


图 1-4. I2C ISR 的软件流程图

所需的 UART 数据包

图 1-5 展示了正确桥接至 I2C 接口所需的 UART 数据包。显示的值是图 1-1 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。在桥接器确认该值之前，UART 传输将被忽略。
- **I2C 地址**：主机与之通信的 I2C 目标的地址。
- **I2C 读取或写入指示器**：桥接器从目标 I2C 器件读取或写入的值。
- **消息长度 N**：传输的数据长度（单位：字节）。该值不能大于定义的 I2C 最大数据包长度。
- **D0, D1..., Dn**：桥接器内传输的数据。

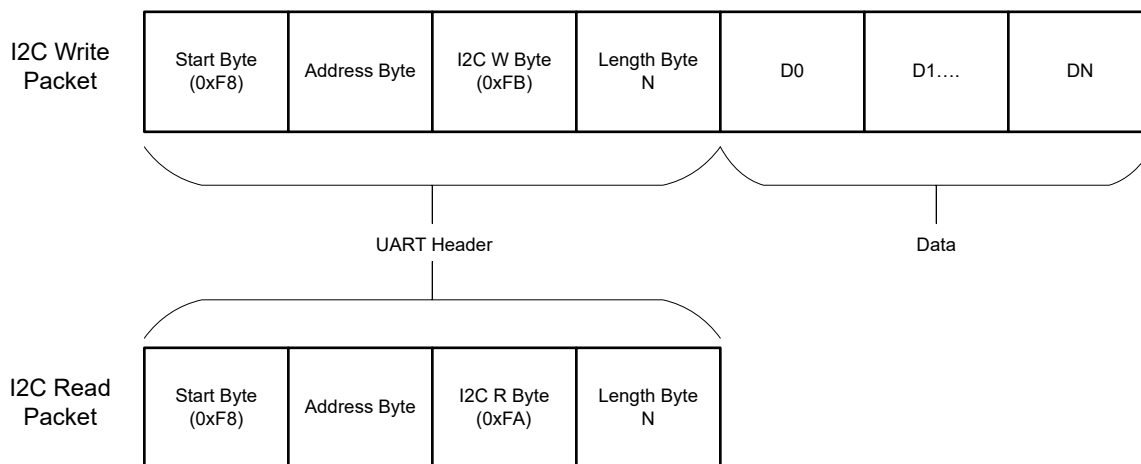


图 1-5. UART 桥接器数据包说明

器件配置

图 1-1 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。

应用代码

要更改 UART 数据包使用的特定值或最大 I2C 数据包大小，请修改文档开头的 **#defines**，如下代码块所示：

```
/* Define UART Header and Start Byte*/
#define UART_HEADER_LENGTH 0x03
#define UART_START_BYTE 0xF8
#define UART_READ_I2C_BYTE 0xFA
#define UART_WRITE_I2C_BYTE 0xFB
#define ADDRESS_INDEX 0x00
#define RW_INDEX 0x01
#define LENGTH_INDEX 0x02

/*Define max packet sizes*/
#define I2C_MAX_PACKET_SIZE 16
#define UART_MAX_PACKET_SIZE (I2C_MAX_PACKET_SIZE + UART_HEADER_LENGTH)
```

代码中有几点是关于错误检测的注释。用户可以在代码中的这些点添加自定义错误处理和额外的错误报告。为简洁起见，此处并未包含所有错误处理代码交叉点。在实际操作中，请搜索代码中的注释，类似于下面代码块中的演示：

```
while (DL_I2C_isControllerRXFIFOEmpty(I2C_BRIDGE_INST) != true) {
    if (gI2C_Count < gI2C_Length) {
        gI2C_Data[gI2C_Count++] =
            DL_I2C_receiveControllerData(I2C_BRIDGE_INST);
    } else {
        /*
         * Ignore and remove from FIFO if the buffer is full
         * Optionally add error flag update
         */
        DL_I2C_receiveControllerData(I2C_BRIDGE_INST);
        gError = ERROR_I2C_OVERUN;
    }
}
```

其他资源

- 德州仪器 (TI)，[UART 转 I2C 桥接器子系统代码](#)
- 德州仪器 (TI)，[详细了解 TI Sysconfig](#) 工具
- 德州仪器 (TI)，[MSPM0 支持开发套件](#) 工具
- 德州仪器 (TI)，[MSPM0 Academy : UART](#) 培训
- 德州仪器 (TI)，[MSPM0 Academy : I2C](#) 培训

1 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 * (December 2023) to Revision A (August 2025)	Page
• 删除了“兼容器件”部分	1

商标

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