

Subsystem Design

扫描比较器



1 说明

该子系统演示了如何在 MSPM0 微控制器中使用单个集成比较器和软件来表示多个比较器。该过程使设计人员能够最大限度地利用比较器功能，并且可以利用比器件上物理存在的比较器更多的理论比较器。该示例专门循环使用三种不同的比较器配置和输入引脚，同时设置三个输出引脚，结果如图 1-1 所示。

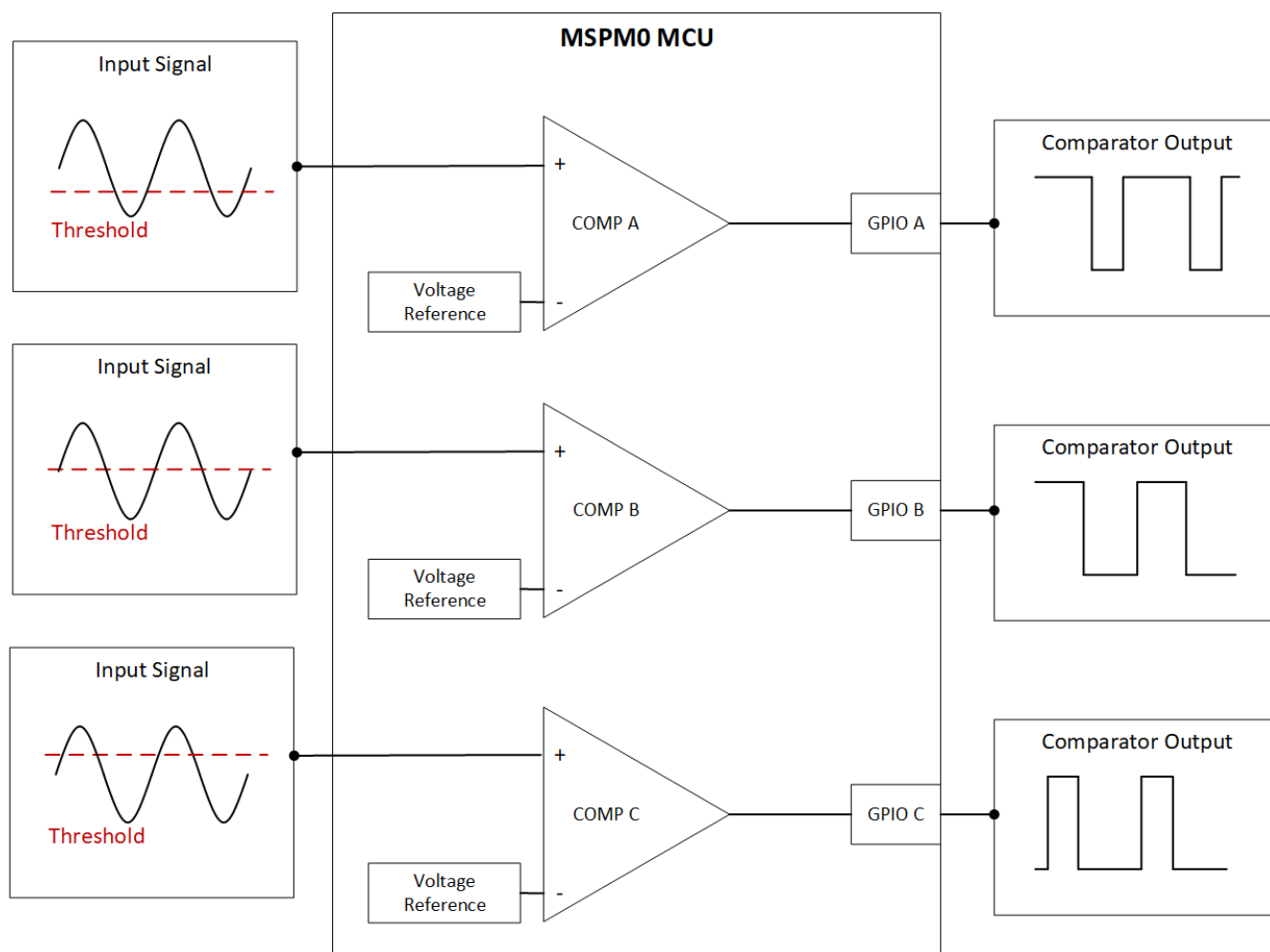


图 1-1. 扫描比较器子系统的理论功能

该示例利用 MSPM0 比较器的可定制 IO 多路复用，可以为同一比较器实现多个信号输入。该示例的三个信号输入位于 COMP_IN0+、COMP_IN0- 和 COMP_IN1- 引脚上，如图 1-2 所示。

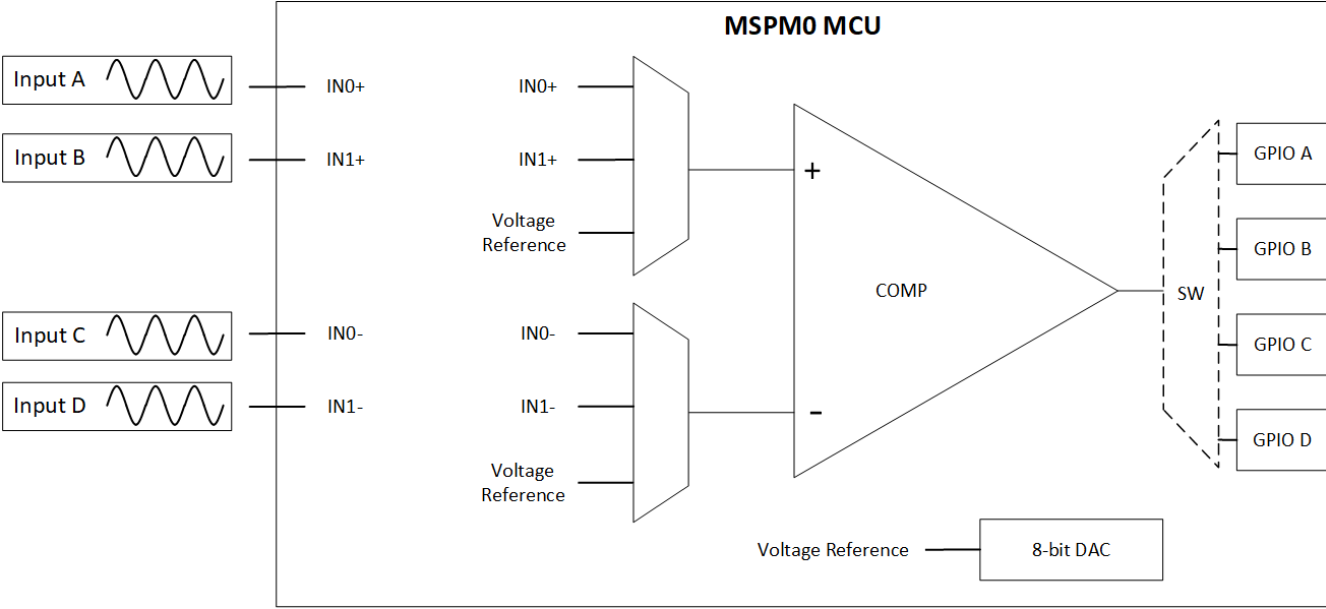


图 1-2. 比较器输入和输出多路复用器

2 所需外设

表 2-1 介绍了所需的集成 COMP 和 GPIO。

表 2-1. 所需外设

使用的外设	注释
比较器	在代码中称为 COMP_INST (包括 8 位基准 DAC)
GPIO	这三个 GPIO 分别称为引脚 A、B 和 C。

3 设计步骤

1. 根据设计要求确定比较器的多种配置，包括工作模式、通道输入和电压基准。
2. 利用 **SysConfig** 生成比较器配置代码。
3. 在 **SysConfig** 中配置所需的 **GPIO**。
4. 通过步骤 1 - 2 为每个比较器配置创建单独的函数。
5. 编写应用程序代码以调用每个配置设置、稳定时间延迟并将结果分配给相应的 **IO** 引脚。有关软件概览，请参阅图 5-1。

4 设计注意事项

1. 稳定时间：更新比较器的配置后，应用代码需要经过一段延迟，以便留出启用时间、**DAC** 稳定时间和传播延迟时间，然后再读取结果。在应用程序代码中设置延迟时，请参阅相应的 **MSPM0** 数据表的比较器规格部分。
2. 工作模式：比较器具有高速模式和低功耗模式。高速模式会消耗更多电流，但会缩短比较器读取之间的时间。低功耗模式需要更长的读取间延迟，但可以降低器件的电流消耗。作为参考，该示例使用高速模式。
3. 响应时间：当子系统循环使用多个比较器配置时，该过程会增加比较器的最大响应时间。最大响应时间是稳定时间延迟乘以仿真比较器配置数量的因子。标准响应时间 = x ；仿真响应时间 = 延迟 $\times$ 仿真比较器 ($45 \mu s$)

5 软件流程图

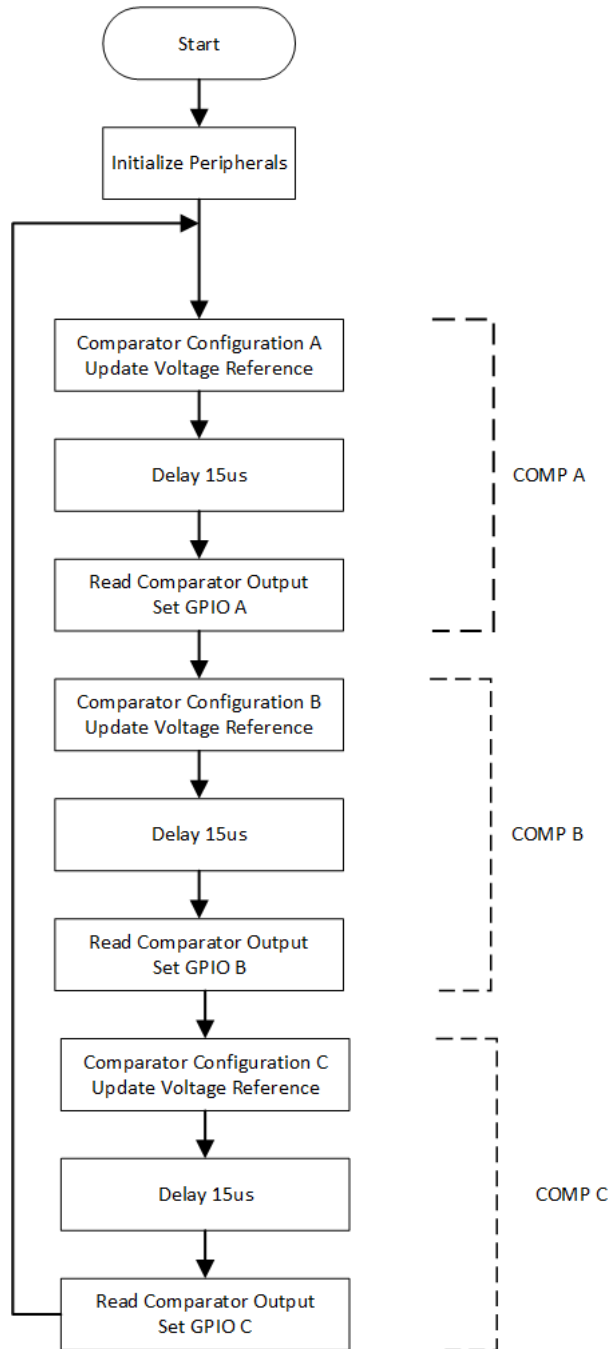


图 5-1. 应用软件流程图

6 应用代码

应用程序代码通过调用以下三个函数循环使用三个不同的比较器配置：update_comp_configA()、update_comp_configB() 和 update_comp_configC()。每次重新配置比较器之后，应用程序代码都会因传播和稳定延迟而延迟 15 μ s，然后再读取比较器输出并设置相应的 GPIO。

```
int main(void)
{
    //initialization
    SYSCFG_DL_init();
    DL_COMP_enable(COMP_INST);
    DL_SYSCCTL_enableSleepOnExit();

    while (1) {

        //0.5V reference
        update_comp_configA();
        delay_cycles(480); //15us delay for comp stabilization
        if (DL_COMP_getComparatorOutput(COMP_INST) == 1){
            DL_GPIO_setPins(COMP_OUTPUT_PORT,COMP_OUTPUT_A_PIN); //set GPIO high
        }else{
            DL_GPIO_clearPins(COMP_OUTPUT_PORT,COMP_OUTPUT_A_PIN); //set GPIO low
        }

        //1.0V reference
        update_comp_configB();
        delay_cycles(480); //15us delay for comp stabilization
        if (DL_COMP_getComparatorOutput(COMP_INST) == 1){
            DL_GPIO_setPins(COMP_OUTPUT_PORT,COMP_OUTPUT_B_PIN); //set GPIO high
        }else{
            DL_GPIO_clearPins(COMP_OUTPUT_PORT,COMP_OUTPUT_B_PIN); //set GPIO low
        }

        //1.5V reference
        update_comp_configC();
        delay_cycles(480); //15us delay for comp stabilization
        if (DL_COMP_getComparatorOutput(COMP_INST) == 1){
            DL_GPIO_setPins(COMP_OUTPUT_PORT,COMP_OUTPUT_C_PIN); //set GPIO high
        }else{
            DL_GPIO_clearPins(COMP_OUTPUT_PORT,COMP_OUTPUT_C_PIN); //set GPIO low
        }
    }
}
```

图 6-1. 扫描比较器 Main.C

7 结果

图 7-1 显示了扫描比较器子系统示例的结果。仿真比较器 A、B 和 C 的基准电压分别被设置为 0.5V、1.0V 和 1.5V。在三个仿真比较器中的每一个上测量了是相同的输入信号。

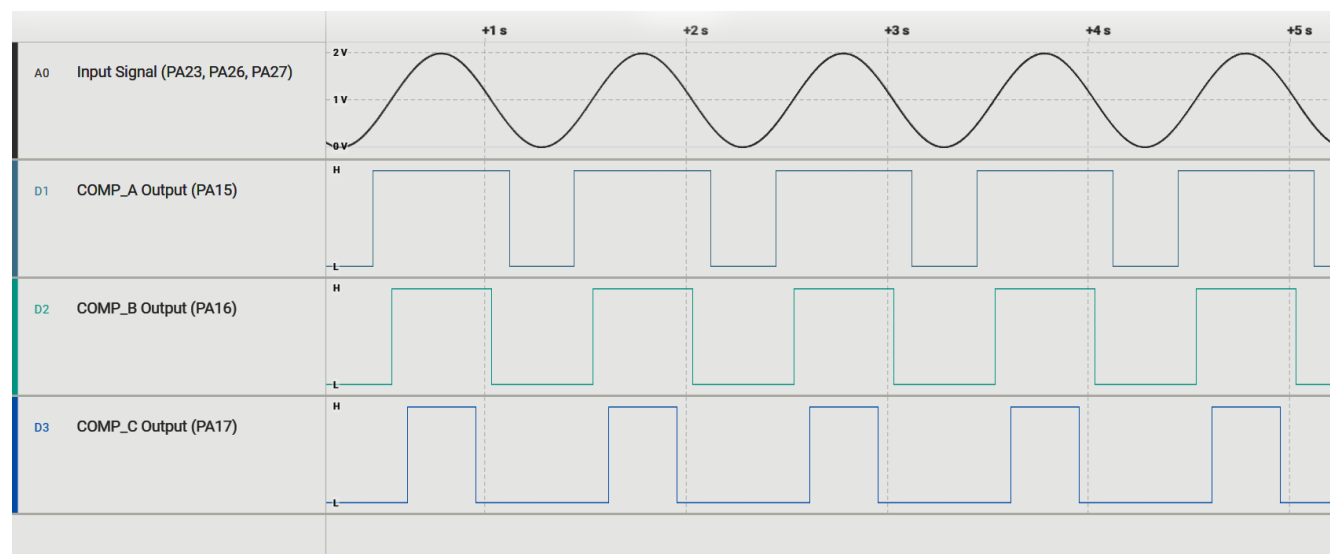


图 7-1. 结果

示波器读数表明，一个物理比较器能够模拟三个同时运行的比较器。通过更改 `comp_hal.c` 中的函数，可以编辑该示例代码以适应不同的比较器数量和配置。

8 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 Academy](#)

9 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。

10 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 * (September 2024) to Revision A (August 2025)	Page
• 删除了“兼容器件”部分	3

商标

LaunchPad™ and E2E™ are trademarks 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