

Subsystem Design

连接二极管矩阵



1 说明

[连接二极管矩阵的示例](#)演示了如何使用矩阵格式减少在使用六个或更多 LED 时所需的 GPIO 引脚数量。此特定示例使用九个 LED 和六个 GPIO 来形成并控制一个 3×3 LED 矩阵。矩阵格式创建了一个每个 LED (或二极管) 使用两个 GPIO 的网格。这种格式在通过 LED 创建标牌或显示屏时特别有用。LED 矩阵的 GPIO 引脚分为行引脚和列引脚。如图 1-1 所示, 当行引脚连接 LED 的阴极时, 矩阵是公共行阴极。公共行阳极是指行引脚连接 LED 的阳极。根据 LED 矩阵中的 LED 配置, 行引脚和列引脚被设置为高电平有效或低电平有效。在此子系统示例中, 行引脚为低电平有效, 列引脚为高电平有效。为了使 LED 矩阵正常运行, 必须一次一行控制矩阵中的 LED。本示例的应用程序代码使用状态机连续循环遍历行, 以点亮和熄灭 LED。

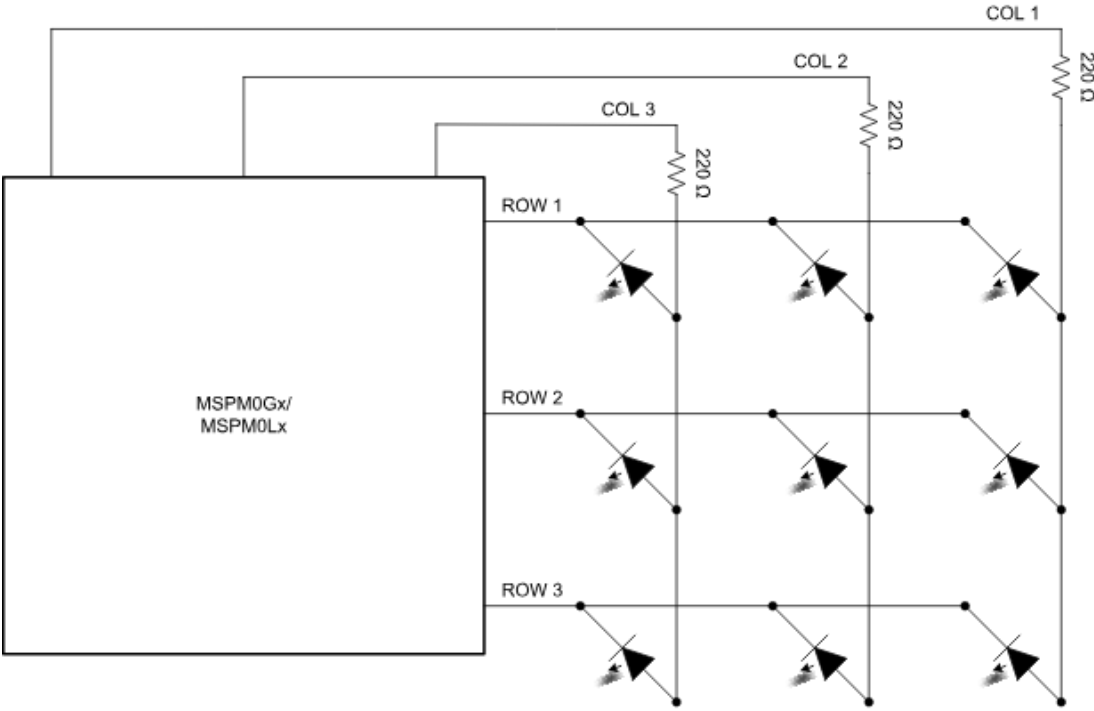


图 1-1. 子系统功能方框图

2 所需外设

此应用需要六个 GPIO 引脚和计时器中断。

表 2-1. 所需外设

子块功能	外设使用	注释
GPIO 子块	6 个 GPIO 引脚	此示例中使用的所有引脚都在同一端口上
计时器	计时器中断	计时器中断用于循环遍历 LED 矩阵上的行

3 设计步骤

1. 确定矩阵中使用的 LED 数量以及矩阵尺寸。矩阵尺寸决定了所需的 GPIO 引脚数。
2. 将 GPIO 引脚分为行引脚和列引脚。
3. 将所有行和列引脚配置为输出。
4. 通过采用所有列引脚 GPIO 值的逐位或运算来确定列引脚的掩码值。
5. 创建存储器表和存储器表更新函数。
6. 为行更新状态机创建枚举表以遍历行。
7. 配置计时器中断，并为行更新状态机编写应用程序代码，以使 LED 状态递增。
8. 编写应用程序代码，以设置显示周期，并在显示发生变化时使用新的列引脚值更新存储器表。

4 设计注意事项

1. **LED 数量和矩阵尺寸**：矩阵尺寸决定运行矩阵所需的 GPIO 引脚数。例如，一个 16 LED 矩阵可以在 4×4 矩阵中使用 8 个引脚，也可以在 2×8 矩阵中使用 10 个引脚。
2. **LED 配置**：行和列引脚的活动状态取决于矩阵处于公共行阴极还是公共行阳极。
3. **列引脚值**：列引脚值在存储器表中设置。确切的值由选择的引脚和相应的列掩码决定。为了便于设置，选择按数字顺序无间隙的引脚最简单。
4. **列和行引脚连接**：将引脚连接到 LED 矩阵时，如果行引脚从最顶部的行（向下移动）开始，而列引脚从最右侧的列（向左移动）开始，则应用程序编程非常简单。
5. **计时器中断**：中断的速度会影响显示周期以及每排 LED 在每个状态机周期中处于开启状态的时间。这个特定示例每 5ms 中断一次，防止人眼觉察到任何闪烁。
6. **更新存储器表**：更新存储器表的具体方法取决于应用程序。该示例将计数器（也称为显示周期）递增至指定值。当计数器达到该值时，存储器表会更新以设置新显示。

5 软件流程图

图 5-1 展示了该子系统示例的软件流程图，并解释了用于控制 LED 矩阵的计时器中断例程和状态机。

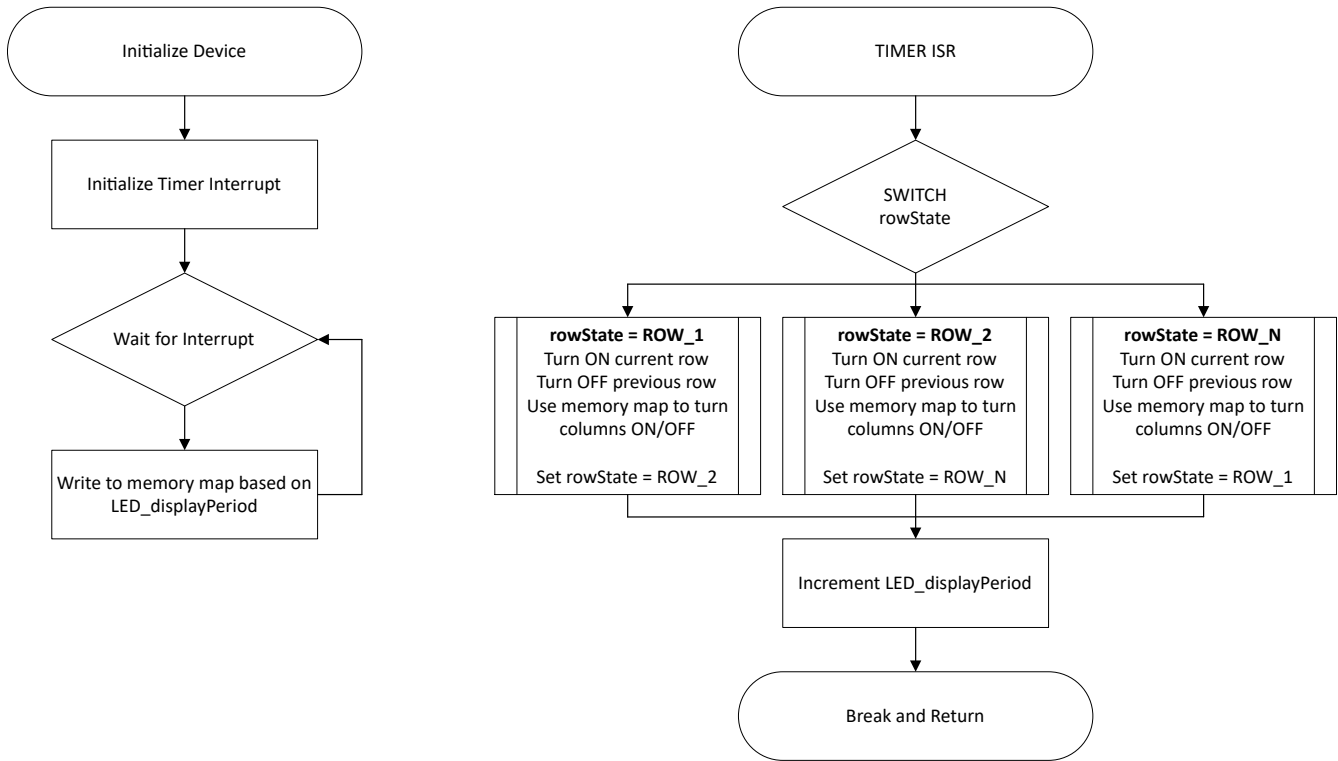


图 5-1. 应用软件流程图

6 应用代码

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。

该示例使用几个关键变量：行数、列掩码值、显示周期时长以及一个用于跟踪中断数的计数器。行数是用于构建存储器表数组的定义值。列掩码等效于所用的所有列引脚的 GPIO 值的按位或运算。列掩码与存储器表一同使用，用于确定在给定时间每行的哪些列引脚需要打开或关闭。显示周期变量乘以每个计时器中断的持续时间，以确定使用单个存储器表写入的时长。在本示例中，显示周期值定义为 100，相当于半秒的显示周期时间。计数器，即 gLedState，用于跟踪与显示周期值相关的中断数。这样可确保将存储器表写入每个显示周期。

```

#define NUMBER_OF_ROWS 3
#define COL_MASK 0x38
#define LED_DISPLAY_PERIOD 100 /* timer period = 5 ms, so display period = 500 ms */
volatile uint32_t gLedState = 0;
void LED_updateTable(uint8_t rowNumber, uint8_t LEDs);
  
```

下一段代码显示了枚举表以及计时器中断请求 (IRQ)。枚举表定义了 **rowState** 开关在计时器 IRQ 中循环的行状态。对于每个 **rowState** (或行引脚)，将打开当前行、关闭前一行，并通过将列掩码值与存储器表值进行比较来设置列。然后设置下一个 **rowState**。此示例按顺序从第一行循环到 **N** 行，然后返回到第一行。在离开计时器 IRQ 之前，**gLedState** 会递增以跟踪每个显示周期的中断数。

```
typedef enum {
    ROW_1,
    ROW_2,
    ROW_3
}rowNumber;

rowNumber rowState = ROW_1;

void LED_STATE_INST_IRQHandler(void) {
    switch (DL_TimerG_getPendingInterrupt(LED_STATE_INST)){
        case DL_TIMER_IIDX_ZERO:
            /* State machine to auto cycle from row 1 to row N and repeat */
            switch (rowState){
                case ROW_1:
                    /* Turn on ROW_1, Turn off ROW_3 */
                    DL_GPIO_clearPins(ROW_PORT, ROW_ROW_1_PIN);
                    DL_GPIO_setPins(ROW_PORT, ROW_ROW_3_PIN);

                    /* Set COLUMN values */
                    DL_GPIO_writePinsVal(COLUMN_PORT, COL_MASK, gLedMemoryTable[0]);
                    rowState = ROW_2;
                    break;
                case ROW_2:
                    /* Turn on ROW_2, Turn off ROW_1 */
                    DL_GPIO_clearPins(ROW_PORT, ROW_ROW_2_PIN);
                    DL_GPIO_setPins(ROW_PORT, ROW_ROW_1_PIN);

                    /* Set COLUMN values */
                    DL_GPIO_writePinsVal(COLUMN_PORT, COL_MASK, gLedMemoryTable[1]);
                    rowState = ROW_3;
                    break;
                case ROW_3:
                    /* Turn on ROW_3, Turn off ROW_2 */
                    DL_GPIO_clearPins(ROW_PORT, ROW_ROW_3_PIN);
                    DL_GPIO_setPins(ROW_PORT, ROW_ROW_2_PIN);

                    /* Set COLUMN values */
                    DL_GPIO_writePinsVal(COLUMN_PORT, COL_MASK, gLedMemoryTable[2]);
                    rowState = ROW_1;
                    break;
            }

            /* Increment LED_STATE */
            gLedState++;

            break;
        default:
            break;
    }
}
```

在主代码中，要做的就是每个显示周期写入存储器表。这种情况会无限重复。该特定代码使用二进制文件来确定哪个 LED 更容易打开，因为 1 和 0 的布局模仿矩阵布局。如果 LED 亮起，二进制值为 1；如果 LED 熄灭，则二进制值为 0。

```
while(1){
    __WFI();
    /* Flash TI on repeat in half second increments */
    if (gLedState == LED_DISPLAY_PERIOD){ /* Display "T" for one display period */
        LED_updateTable(1, 0b111);
        LED_updateTable(2, 0b010);
        LED_updateTable(3, 0b010);
    } else if (gLedState == LED_DISPLAY_PERIOD*2){ /* Blank for one display period */
        LED_updateTable(1, 0b000);
        LED_updateTable(2, 0b000);
        LED_updateTable(3, 0b000);
    } else if (gLedState == LED_DISPLAY_PERIOD*3){ /* Display "I" for one display period */
        LED_updateTable(1, 0b111);
        LED_updateTable(2, 0b010);
        LED_updateTable(3, 0b111);
    } else if (gLedState == LED_DISPLAY_PERIOD*4){ /* Blank for one display period */
        LED_updateTable(1, 0b000);
        LED_updateTable(2, 0b000);
        LED_updateTable(3, 0b000);
    } else if (gLedState > LED_DISPLAY_PERIOD*4){ /* Reset gLedState and start over */
        gLedState = 0;
    }
}
```

7 硬件设计

此特定子系统示例需要九个 LED、三个电阻器和至少六根导线。要设置矩阵，请将 LED 排列成 3 × 3 行。将每排 LED 的阴极连接在一起。然后，将每列 LED 的阳极连接在一起。将一个 220 Ω 电阻器连接到每条列线。然后，根据器件配置，将行线和列线连接到正确的器件引脚。有关连接指南，请参阅图 1-1。

8 结果

图 8-1 展示了此应用中“T”显示周期的预期结果。该图的上半部分显示了应用程序状态机在每个中断中遍历每一行的情况下每行的状态。图的下半部分显示了整个周期内的复合图像。这就是人眼看到的矩阵。

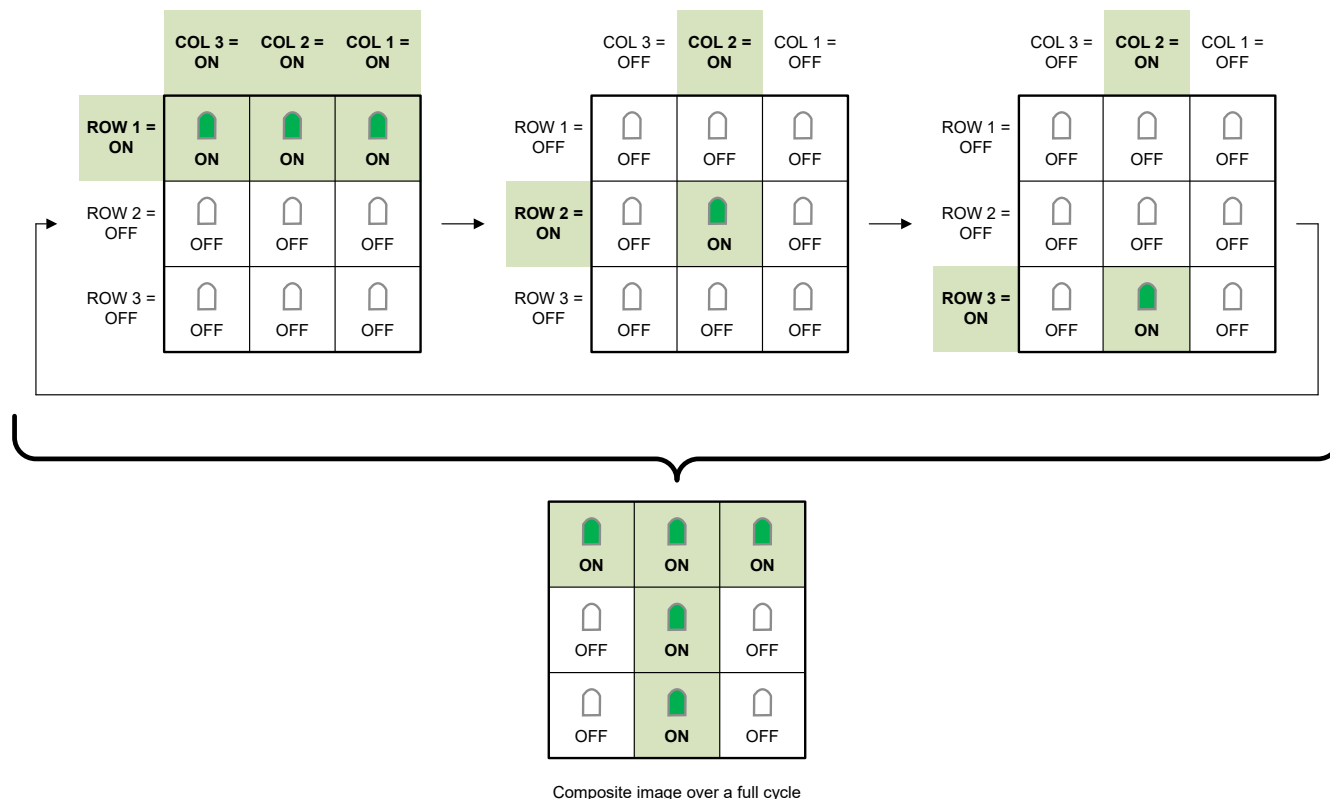


图 8-1. 结果

9 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](#)

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	2

11 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。

12 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