

Subsystem Design

并联 IO 转 UART 桥接器



设计说明

许多应用需要同时捕获几个 GPIO 的状态变化，进行更新，然后通过 UART 将状态发送到主机。具有成本效益的微控制器 (MCU) 具有足够的 GPIO 资源，可以实施并联转串联，并通过 UART 实时向主机（例如 PC 端）发送数据。[下载此示例的代码。](#)

图 1-1 显示了该子系统的功能图。

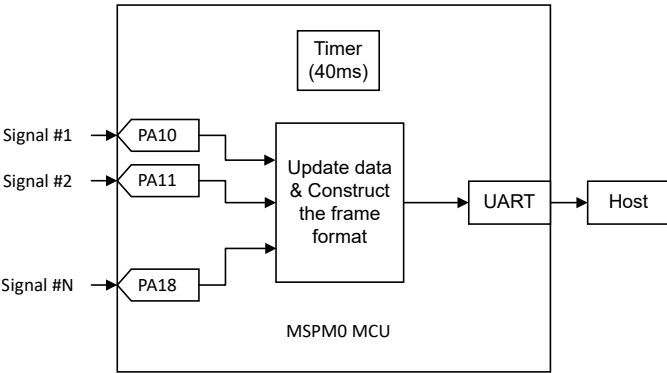


图 1-1. 子系统功能方框图

所需外设

此应用需要 9 个 GPIO、1 个计时器和 1 个 UART。

表 1-1. 所需外设

子块功能	外设使用	注释
IO 输入	9 引脚	在代码中称为 <i>GROUP1_IRQHandler</i>
计时器间隔	TIMG0	在代码中称为 <i>TIMG0_IRQHandler</i>
UART 输出	UART0	在代码中名为 <i>transmitPacketBlocking</i>

设计步骤

1. 捕获 9 个 GPIO 开关的状态。
2. 在数据段中填充这 9 位，并通过 UART 将一个已完成的帧传输到主机 PC。
3. 在检测到任何操作时或每 40ms 更新一次数据。

设计注意事项

此实施使用 9 个 GPIO 引脚 (PA10-PA18) 来捕获开关的状态，这些状态表示相应的操作，如表 1-2 所示：

表 1-2. 引脚与操作之间的对应关系

GPIO 引脚	操作
PA10	GPIO_Signal_10
PA11	GPIO_Signal_11
PA12	GPIO_Signal_12
PA13	GPIO_Signal_13
PA14	GPIO_Signal_14
PA15	GPIO_Signal_15
PA16	GPIO_Signal_16
PA17	GPIO_Signal_17
PA18	GPIO_Signal_18

在上面的引脚中，PA14 固定地连接到 LaunchPad 中的 S2，而在按下 S2 时，PA14 会下拉至地。对于其他引脚，每个引脚可通过 J11 连接至 S1，在按下 S1 时，该引脚可上拉至 3V3。例如，如果 S1 连接至 PA18 且同时按下两个开关，则数据会更新，如表 1-3 所示：

表 1-3. 9 个引脚的数据格式

	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GPIO 引脚								PA18	PA17	PA16	PA15	PA14	PA13	PA12	PA11	PA10
默认值	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
PA18&14	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

按下任何开关时，MCU 会立即更新数据段（2 个字节）和校验和，然后通过 UART 向 PC 发送使用以下格式的新数据。

如果没有每 40ms 按一次开关，MCU 会向 PC 发送当前状态。发送到 PC 的软件包采用表 1-4 中所示的格式：

表 1-4. UART 发送的数据包格式

字节	标头 (2 字节)		数据长度 (1 字节)	源 ID (1 字节)	目标 ID (1 字节)	命令 (1 字节)	数据索引 (1 字节)	数据 (N 字节)	校验和 (2 字节)	
值	0x5A	0xA5	N	0~63	0~63	0~255	0~255	数据	CSumL	CSumH

软件流程图

图 1-2 展示了主循环 (即主函数) 以及 GPIO 中断处理 (即 GROUP1_IRQHandler 函数) 的代码流程图。

TIMG0 中断处理非常简单, 即每 40ms 进入一次计时器中断并发送当前数据。

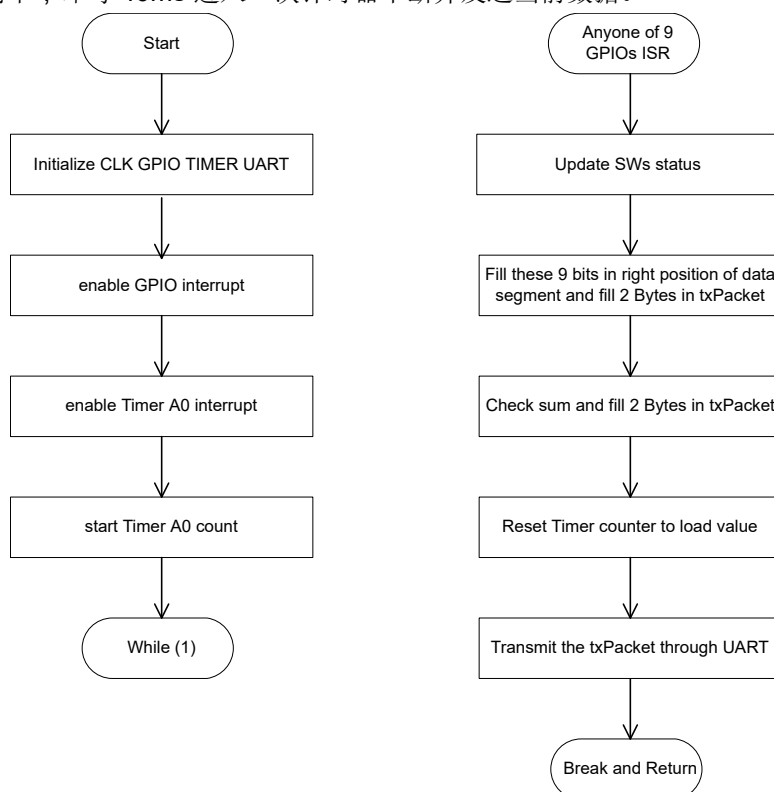


图 1-2. 应用软件流程图

应用代码

主循环

```

SYSCFG_DL_init();
NVIC_EnableIRQ(GPIO_MULTIPLE_GPIOA_INT_IRQN);
NVIC_EnableIRQ(TIMER_0_INST_INT_IRQN);
DL_TimerG_startCounter(TIMER_0_INST);
while (1) {
    __WFI();
}
  
```

TIMG0_IRQHandler

```

switch (DL_TimerG_getPendingInterrupt(TIMER_0_INST)) {
    case DL_TIMER_IIDX_ZERO:
        transmitPacketBlocking(gTxPacket, UART_PACKET_SIZE);
        break;
}
  
```

GPIO GROUP1_IRQHandler

```

if (DL_Interrupt_getPendingGroup(DL_INTERRUPT_GROUP_1)) {
    dataStatus = (GPIOA->DIN31_0);
    dataTemp = (dataStatus >> 10);
    gTxPacket[7] = dataTemp >> 8;
    gTxPacket[8] = dataTemp & 0xFF;

    siganlChecksum = checkSum1ByteIn2ByteOut((gTxPacket+2), 7);
}
  
```

```

gTxPacket[10] = siganlChecksum >> 8;
gTxPacket[9] = siganlChecksum & 0xFF;

DL_TimerG_stopCounter(TIMER_0_INST);
DL_TimerG_setTimerCount(TIMER_0_INST, TIMER_0_INST_LOAD_VALUE);
DL_TimerG_startCounter(TIMER_0_INST);

transmitPacketBlocking(gTxPacket, UART_PACKET_SIZE);
}

```

结果

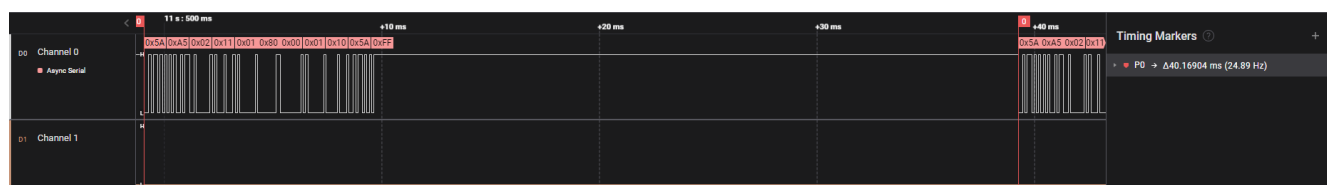
使用逻辑分析来捕捉数据流并显示更多详细信息。

通道 0 ----> UART Tx

通道 1 ----> PA18

下图显示：

如果未按下开关，MCU 每 40ms 发送一次默认值



按下 S1 时，PA18 会检测到上升沿，并进行数据更新。然后 MCU 每 40ms 发送一次更新数据。



如果检测到上升沿，但最后一个数据包尚未完成，则 MCU 在完成最后一次传输后发送数据更新。



其他资源

- [下载 MSPM0 SDK](#)
- [了解有关 SysConfig 的更多信息](#)
- [MSPM0G 技术参考手册 \(TRM\)](#)
- [MSPM0L 技术参考手册 \(TRM\)](#)
- [MSPM0G LaunchPad 开发套件](#)
- [MSPM0L LaunchPad 开发套件](#)
- [MSPM0 TIMER Academy](#)
- [MSPM0 UART Academy](#)

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 * (March 2024) 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