Application Note

使用 MSPM0 通过 UART Over CAN 实现 TPS929xxx LED 驱动器控制

Helic Chi

摘要

本应用手册可作为准备和使用与 LP-MSPM0G3507 配对的 TPS929xxx-Q1 器件系列示例代码的指南。用户还可以使用系统配置工具 (SYSCONFIG) 轻松地将此代码移植到其他 MSPM0 器件。

内容

商标

1 引言

示例代码展示了点亮 TPS929120EVM、TPS929160EVM 和 TPS929240EVM 上的 LED 的功能。每个 TPS929xxxEVM 都有各自的示例代码。但是，唯一的区别在于在 led_driver.h 文件中选择所用的 LED 驱动器 IC。这有助于用户无需对示例代码进行任何修改即可点亮 EVM。

代码中有两种模式：动画和 EEPROM 编程。默认选择动画模式。节 4.2 介绍了如何在这两种模式之间切换。在动画模式下，根据预定义的顺序使用 6 种不同的动画。通过点击 LP-MSPM0G3507 上的按钮 S2，可在不同动画之间切换。在每次播放动画后，都会执行诊断以确定是否发生了任何故障。有关诊断的更多信息，请参阅节 4.3。

示例代码附带许多预定义的 API，这些 API 可用于更改 LED 驱动器的配置、执行诊断或构建自定义 FlexWire 命令。预定义的 API 会根据指定的系统自动调整。如需了解有关系统演示的更多详情，请参阅节 4.2。

2 硬件设置

本节介绍了硬件设置与每个器件特定 EVM 用户指南中的描述之间的差异。检查以下指南中的硬件设置：

- 工具页面：
  - TPS929120EVM
  - TPS929160EVM
  - TPS929240EVM
- 文档：
  - TPS929120EVM 用户指南
  - TPS929160EVM 用户指南
  - TPS929240EVM 用户指南

示例代码将 USB2ANY 替换为 LP-MSPM0G3507。可采用两种方法将 LP-MSPM0G3507 连接到 TPS929xxxEVM，此处使用 TPS929240EVM 进行演示。

表 2-1 中列出了这两种连接方法。图 2-1 中显示了 LP-MSPM0G3507 上的各个位置。首先，将 SW1 和 J14 的 PA9 连接在一起。除上述连接外，示例代码中还使用了 LP-MSPM0G3507 的开关 S1（青色）和 S2（绿色）。有关更多详细信息，请参阅节 4.1。在 TPS929xxxEVM 上，必须针对每种模式正确设置跳线。器件特定 EVM 用户指南对此进行了介绍。

表 2-1. 硬件连接

接口	电路板	UART-RX	UART-TX	+3.3V	GND	+5V
	LP-MSPM0G3507	PA9	PA8	J1-1	J1-22	J1-21
CAN	TPS929120CANEVM	J3-13	J3-14	J3-15	J3-16	+5V (J3-21)
UART	TPS929120EVM	J3-3	J3-4	J3-5	J3-6	\
	TPS929160EVM	J29-3	J29-4	J29-5	J29-6	\
	TPS929240EVM	J4-3	J4-4	J4-5	J4-6	\

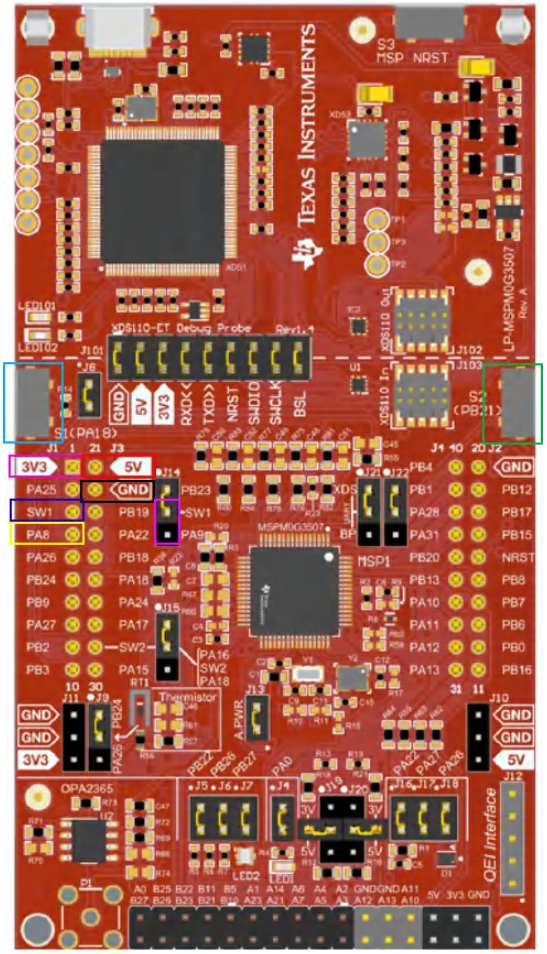

图 2-1. LP-MSPM0G3507 上的连接

3 软件设置

要为 LP-MSPM0G3507 设置软件，请执行以下步骤（在装有 Windows® 10 操作系统的计算机上演示）：

1. 下载并安装 Code Composer Studio™ (CCS)：Code Composer Studio 集成开发环境 (IDE)。按照在线指南安装 CCS 或使用 MSPM0 设计流程指南。
   a. 在安装过程中，当 CCS 安装程序位于 *Select Components* 页面时，选择 *MSPM0 32-bit Arm Cortex-M0+ General Purpose MCUs*。
2. 下载 MSPM0-SDK 并导入示例代码。
   a. 从 ti.com 下载 MSPM0-SDK。
   b. 在 CCS 中，依次点击“Project”和“Import CCS Projects”，选择 TPS929xxx 演示 SDK 文件夹路径：{SDK folder}\examples\nortos\LP_MSPM0G3507\demos\TPS929xxx_control_uart_over_can，即可导入演示代码。
3. 请参阅 TPS929120EVM、TPS929160EVM 或 TPS929240EVM 的用户指南以更改 EVM 地址。同步更改 system_info.h 和 system_info.c 文件中的器件编号 *DEVICE_CNT* 以及器件地址 *device_address*。
4. 构建和加载程序。
5. 按下 LP-MSPM0G3507 上的 S2 按钮，TPS929xxxEVM 将更改 LED 图形。
6. （可选）下载 EEPROM 配置工具。如果用户希望使用非默认数据对 EEPROM 进行编程，该工具非常适用。TPS929160/TPS929240 的工具还包括一个计算工具，用于计算选项卡 IF_CRC 中的 FlexWire 接口 CRC 值。对于每个受支持的器件，都有一个单独的链接：
   a. TPS929120/TPS929121：TPS92912x-Q1 EEPROM 配置工具。
   b. TPS929160/TPS929240：TPS929240-Q1 TPS929160-Q1 EEPROM 配置工具。

4 软件结构

4.1 流程图

示例代码中的简要流程如图 4-1 所示。在整个流程中，使用了 FlexWire 总线上的器件数量及地址。system_info.h 和 system_info.c 文件中指定了这些信息，节 4.2 中对此进行了更详细的说明。

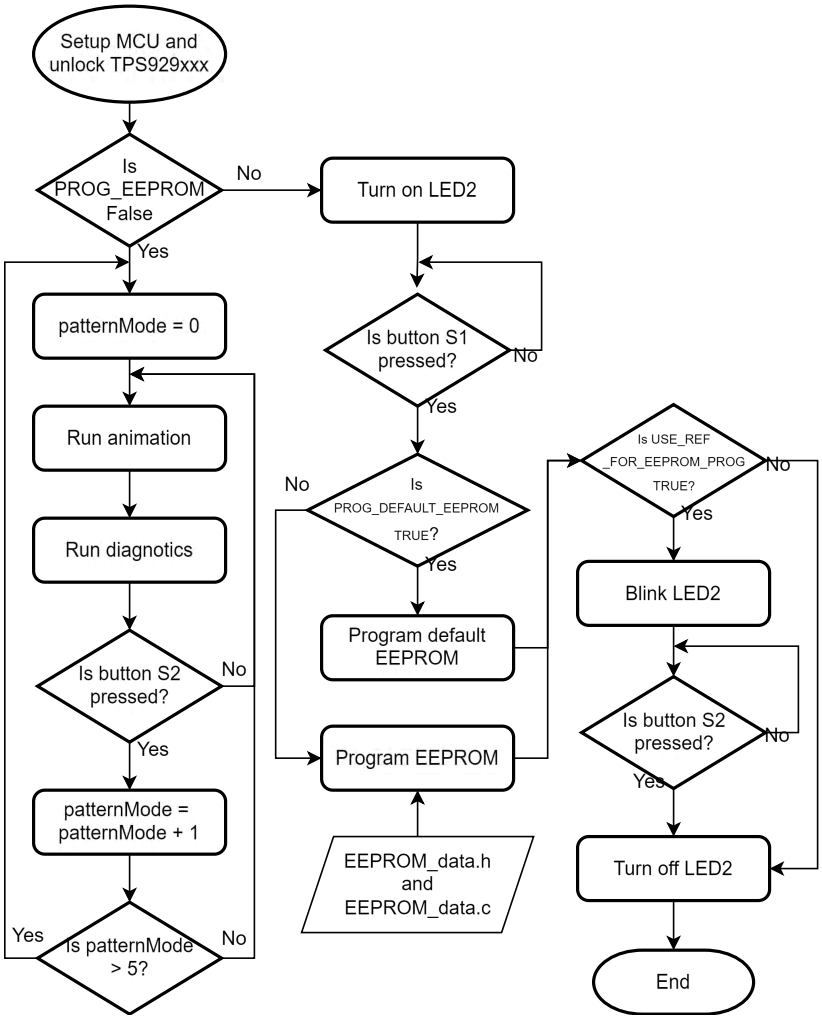

图 4-1. 软件流程图

设置 MCU 会配置 UART 接口并设置为 750000 波特。解锁 LED 驱动器后，M0 会检查是选择了动画模式还是 EEPROM 编程模式。节 4.2 介绍了如何在这两种模式之间切换。在动画模式期间，系统将执行 LED 图形，并在完成后检查诊断结果。如需详细了解诊断结果，请参阅节 4.3。诊断完成后，M0 会检查 LP-MSPM0G3507 上的按钮 S2 是否已按下。如果该按钮未按下，将再次执行同一 LED 图形。如果该按钮已按下，则执行下一个 LED 图形，直到所有 6 个图形都已执行，循环再次从第一个图形重新开始。

在 EEPROM 编程模式期间，LP-MSPM0G3507 上的按钮 S1 和 S2 以及 LED2 用于向用户提供反馈。选择非默认 EEPROM 编程后，eeprom_data.h 和 eeprom_data.c 文件用于对 EEPROM 进行编程。这些文件可由节 2 中提到的 EEPROM 配置工具自动生成。如需更多有关 EEPROM 编程的信息，请参阅节 4.4。

4.2 系统设置

本节介绍了示例代码如何设置不同的参数来识别系统的构建方式。第一部分是实际使用的 LED 驱动器 IC。在 led_driver.h 文件中，选择使用的 LED 驱动器 IC。示例代码默认包括 TPS929240。

示例代码还支持：

- TPS929120
- TPS929120A
- TPS929121
- TPS929121A
- TPS929160
- TPS929240
- TPS929240A

请注意，对于 *Q1* 器件，不会添加该后缀，仅使用基本产品名称。所选器件对于处理寄存器中不同的寄存器地址和字段非常重要。此外，在对默认 EEPROM 进行编程时，指定的 LED 驱动器 IC 是用于对默认值进行编程的驱动器 IC。这意味着当用户想要将 TPS929120 编程为 TPS929120A 时，必须在 led_driver.h 文件中选择 TPS929120A。

表 4-1 中汇总了影响系统设置的宏和变量。

表 4-1. 每个文件的宏和变量名称汇总

文件名	宏或变量名称	说明
system_info.h	DEVICE_CNT	FlexWire 总线上的器件数量
	CAN_USED	在 UART 或 UART 转 CAN 之间进行选择
	ALWAYS_CHECK_CRC	为所有非广播命令启用 CRC 检查功能
	PROG_EEPROM	启用 EEPROM 编程模式
	PROG_DEFAULT_EEPROM	对默认 EEPROM 值进行编程，而不是对自定义 EEPROM 值进行编程
	USE_REF_PIN_FOR_EEPROM_PROG	在 EEPROM 编程期间使用 REF 引脚
system_info.c	device_address	FlexWire 总线上的器件地址列表
FlexWire.c	rcvCrcError	如果接收到的 CRC 有错误则报告

在 system_info.h 文件中，FlexWire 总线上的器件数由宏 DEVICE_CNT 定义。示例代码仅支持 1 条 FlexWire 总线。

```
// Total devices on FlexWire bus #define
          DEVICE_CNT 1
```

这些器件的实际地址在 system_info.c 文件中指定。地址序列决定了 FlexWire 非广播写和读命令的顺序。因此，对于不同的器件地址序列，动画模式下的 LED 图形看起来会有所不同。

```
const uint16_t
          device_address[DEVICE_CNT] = {DEVICE_ADDR__1};
```

system_info.h 文件还定义了其他系统参数。

```
//
          Define if CAN or UART is used #define CAN_USED FALSE // When non-broadcast is transmitted,
          does the CRC need to be checked #define ALWAYS_CHECK_CRC FALSE
```

宏 *CAN_USED* 定义是否为 FlexWire 总线使用 UART 或 UART 转 CAN。这会影响在 MCU UART-RX 引脚上接收到的总字节数。

宏 *ALWAYS_CHECK_CRC* 定义对于接收到的反馈，是否每个非广播写入命令都需要检查 CRC。当检查 CRC 后发现其不正确时，全局变量 *rcvCrcError* 设置为 TRUE。在所有其他情况下，该变量设置为 FALSE。变量 *rcvCrcError* 在文件 FlexWire.c 中定义。

```
// When an error in CRC of the received data is observed, set this to TRUE unsigned
        int rcvCrcError;
```

4.3 诊断

示例代码提供了一个 API 来检测哪些器件存在开路、短路或单 LED 短路等故障。TPS929xxx_APIs.h 文件中定义了该 API 的原型。

```
void LED_Update_Chip_Status(unsigned int dev_addr_x);
```

该 API 会更新 system_info.h 中定义的变量 *chip_status*。对于器件 TPS929160-Q1 和 TPS929240-Q1，还有一个称为 VBAT 的额外电源引脚。因此，对于这些器件，该变量包括为此引脚测得的电压结果。此外，这些器件还包括一个称为电源欠压的额外故障类型。因此，这些器件包含标志 SUPUV。

```
struct chipStatus {
// Indicates open, short, and/or single-LED-short fault
uint16_t OUT_flag;
uint16_t SHORT_channels[MAX_CHANNEL_CNT];
uint16_t OPEN_channels[MAX_CHANNEL_CNT];
uint16_t SLS_channels[MAX_CHANNEL_CNT]; // Single-LED-short
uint16_t EEPCRC; // EEPROM CRC fault
uint16_t TSD; // Thermal Shutdown
uint16_t PRETSD; // Pre-thermal shutdown warning
uint16_t REF; // REF-pin fault
uint16_t LOWSUP; // Low supply
uint16_t POR; // Power-on-reset
#ifndef TPS92912X
uint16_t SUPUV; // Supply undervoltage
uint16_t VBAT_mV; // *1 mV
#endif
uint16_t VSUPPLY_mV; // *1 mV
uint16_t VLDO_mV; // *1 mV
uint16_t TEMPSNS_10mC; // *10 mC
uint16_t VREF_100uV; // *100 uV
uint16_t IREF_10nA; // *10 nA
};
// For diagnostics
extern struct chipStatus chip_status[];
```

在代码调试期间，可以按照“观察变量、表达式和寄存器”中的步骤在表达式视图中观察变量 *chip_status*。图 4-2 中描述了一个没有任何错误的示例。变量 *chip_status* 的第一个索引是 FlexWire 总线上 LED 驱动器的地址。总共有 16 个不同的地址。因此，索引的范围为 0 至 15。

图 4-3 中显示了一个带有短接的示例。TPS929240-Q1 具有地址 0x1 并设置了 OUT_Flag 标志。当数组 SHORT_channels 被扩展时，引脚 OUT2 上发生短路。

Memory Browser (x)= Variables Expressions Registers

Expression	Type	Value
chip_status	struct chipStatus[16]	[{OUT_flag=0,SHORT_channels=[0,0,0,0,0.....
[0]	struct chipStatus	{OUT_flag=0,SHORT_channels=[0,0,0,0,0...]...
[1]	struct chipStatus	{OUT_flag=0,SHORT_channels=[0,0,0,0,0...]...
OUT_flag	unsigned short	0
SHORT_channels	unsigned short[24]	[0,0,0,0,0...]
OPEN_channels	unsigned short[24]	[0,0,0,0,0...]
SLS_channels	unsigned short[24]	[0,0,0,0,0...]
EEPCRC	unsigned short	0
TSD	unsigned short	0
PRETSD	unsigned short	0
REF	unsigned short	0
LOWSUP	unsigned short	0
POR	unsigned short	0
SUPUV	unsigned short	0
VBAT_mV	unsigned short	0
VSUPPLY_mV	unsigned short	0
VLDO_mV	unsigned short	0
TEMPSNS_10mC	unsigned short	0
VREF_100uV	unsigned short	0
IREF_10nA	unsigned short	0
[2]	struct chipStatus	{OUT_flag=0,SHORT_channels=[0,0,0,0,0...]...

图 4-2. TPS929240-Q1 的 chip_status（无错误）示例

Memory Browser (x)= Variables Expressions Registers

Expression	Type	Value
chip_status	struct chipStatus[16]	[{OUT_flag=0,SHORT_channels=[0,0,0,0,0.....
[0]	struct chipStatus	{OUT_flag=0,SHORT_channels=[0,0,0,0,0...]...
[1]	struct chipStatus	{OUT_flag=1,SHORT_channels=[0,0,1,0,0...]...
OUT_flag	unsigned short	1
SHORT_channels	unsigned short[24]	[0,0,1,0,0...]
[0]	unsigned short	0
[1]	unsigned short	0
[2]	unsigned short	1
[3]	unsigned short	0

图 4-3. TPS929240-Q1 的 chip_status（有错误）示例

图 4-4 中展示了 TPS929240-Q1 中出现低电源电压警告 (V(SUPPLY) < V(ADCLOWSUPTH)) 时的示例。已为地址为 0x1 的器件设置了标志 LOWSUP。此外，对于该警告，电源电压由 ADC 测量并在诊断中报告。在本例中，测量得到的结果为 4804mV。

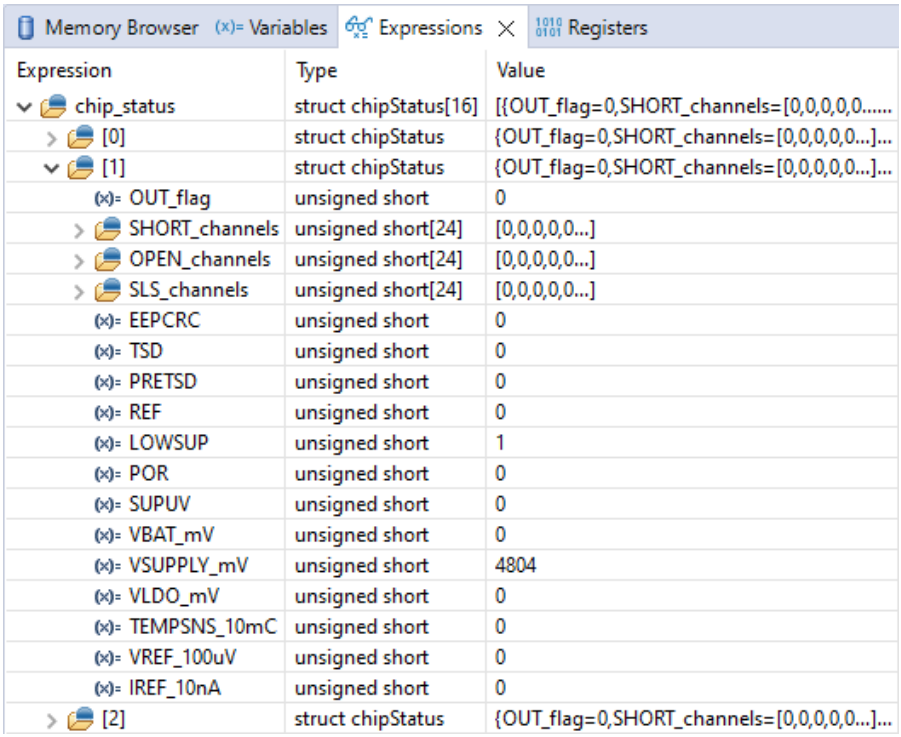

图 4-4. TPS929240-Q1 的 chip_status（具有低电源电压）示例

4.4 EEPROM 编程

示例代码包括对 EEPROM 进行编程的功能。此功能由 system_info.h 文件中定义的宏启用。

```
// When set to 1, the EEPROM programming routine is executed instead of normal program
#define PROG_EEPROM             (FALSE)
// When set to 1, program the EEPROM to the default value
#define PROG_DEFAULT_EEPROM         (FALSE)
// Use external device address settings for EEPROM programming
#define USE_REF_PIN_FOR_EEPROM_PROG     (FALSE)
```

当宏 *PROG_EEPROM* 被定义为 TRUE 时，*EEPROM* 编程模式被启用。示例代码可以对指定 LED 驱动器 IC 或自定义设置的默认 EEPROM 值进行编程。当 *PROG_DEFAULT_EEPROM* 宏定义为 FALSE 时，会对自定义设置进行编程。此设置在 eeprom_data.h 和 eeprom_data.c 文件中指定。这些文件可由节 3 中提到的 EEPROM 配置工具自动生成。

LED 驱动器 IC 支持两种针对单独芯片选择的方法，即通过拉高 REF 引脚或通过使用地址引脚配置器件地址来实现。当 *USE_REF_PIN_FOR_EEPROM_PROG* 宏被定义为 TRUE 时，REF 引脚在编程期间被拉高。当 *USE_REF_PIN_FOR_EEPROM_PROG* 被定义为 FALSE 时，使用当前器件地址。TI 建议使用当前器件地址。

当代码进入 EEPROM 编程例程时，M0 会使 LP-MSPM0G3507 上的 LED2 (PB27) 亮起。当宏 *USE_REF_PIN_FOR_EEPROM_PROG* 定义为 TRUE 时，REF 引脚会在 LED2 亮起后被上拉。表 4-2 中列出了为每个 EVM 上拉 REF 引脚所需的跳线。

LED2 亮起后，需按下 LP-MSPM0G3507 上的按钮 S1 以开始编程。当使用当前器件地址时，LED2 会在编程完毕后熄灭。

当使用 REF 引脚时，LED2 会在编程完毕后开始闪烁。此时，移除 REF 引脚上的上拉电阻，然后需要按下 LP-MSPM0G3507 上的按钮 S2。然后，LED2 熄灭。

表 4-2. 在 EEPROM 编程期间使用 REF 引脚时要设置的 EVM 跳线

EVM	跳线
TPS929120EVM	J2 位置 2 和 3 (+5V)
TPS929160EVM	J52 位置 2 和 3 (VLDO)
TPS929240EVM	J10 位置 2 和 3 (VLDO)

5 总结

汽车照明广泛用于汽车前照灯、尾灯和环境照明。随着汽车照明领域对动画的需求不断增加，必须对 LED 进行单独控制。因此，需要更出色的 MCU 和设计来满足新的要求。MSPM0G 系列搭配 TPS929xxxEVM，可实现独立 LED 控制设计。

本应用手册介绍了 LP-MSPM0G3507 和 TPS929xxxEVM 之间的硬件连接以及如何从零开始设置测试和调试环境，包括 IDE 和 SDK 安装以及代码导入步骤。

此外，本应用手册还提供了软件图、软件系统参数设置和演示代码的基本功能，并展示了如何通过简单的步骤测试芯片错误状态并提供结果用于比较。

6 参考资料

1. 德州仪器 (TI)，TPS929xxx-Q1 示例代码用户指南。
2. 德州仪器 (TI)，TPS929240-Q1 具有 FlexWire 接口的高侧 (O)LED 驱动器数据表。
3. 德州仪器 (TI)，TPS929160-Q1 具有 FlexWire 接口的高侧 (O)LED 驱动器数据表。
4. 德州仪器 (TI)，TPS929120-Q1 具有 FlexWire 接口的高侧 (O)LED 驱动器数据表。
5. 德州仪器 (TI)，TPS929240EVM 用户指南。
6. 德州仪器 (TI)，TPS929160EVM 用户指南。
7. 德州仪器 (TI)，TPS929120EVM 用户指南。
8. 德州仪器 (TI)，MSPM0G350x 具有 CAN-FD 接口的混合信号微控制器数据表。
9. 德州仪器 (TI)，MSPM0 G 系列 80MHz 微控制器技术参考手册用户指南。
10. 德州仪器 (TI)，MSPM0G3507 LaunchPad 开发套件用户指南 (LP-MSPM0G3507) EVM 用户指南。

重要声明和免责声明

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出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 产品的应用。严禁对这些资源进行其他复制或展示。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