

## 低功耗加速器 (LEA) 常见问题解答 (FAQ)

### 摘要

本文档解答了有关低功耗加速器 (LEA) 模块的常见问题；这种模块用于 MSP430™FRAM MCU 上的信号处理。它介绍了如何开始使用支持 LEA 模块的器件，以及如何将其用于执行高效的信号处理、矩阵乘法和其他运算。

### 内容

1	什么是用于信号处理的低功耗加速器 (LEA)?	1
2	LEA 模块是否支持浮点或定点运算?	1
3	哪些器件支持 LEA 模块?	2
4	LEA 模块能带来什么性能优势? 基准测试数据在哪里?	2
5	我应如何开始使用 LEA 模块?	2
6	MSP DSP 库是什么?	2
7	如何安装 DSPLib 和 DSPLib GUI?	2
8	我可以从哪里找到展示 LEA 模块的示例?	2
9	如果我习惯使用 Python 或 Matlab 来生成滤波器, 怎么办?	3
10	如何在我的程序中使用 LEA 模块?	3
11	如何使用 DSPLib GUI 生成我自己的滤波器系数?	4
12	如何将我在 DSPLib GUI 中生成的系数应用到代码?	4
13	TI 还有其他哪些利用 LEA 模块的配套资源, 比如 TI Designs 参考设计?	5
14	如果我还有关于 LEA 的问题, 应如何寻求帮助?	5
15	如果使用 LEA 模块进行计算, 我的函数需要多少能量或几个周期?	5
16	在 DSPLib API 中, 哪些 API 使用 LEA 函数? LEA 模块支持哪些函数?	5

### 商标

MSP430, MSP430Ware, Code Composer Studio, E2E are trademarks of Texas Instruments.  
All other trademarks are the property of their respective owners.

## 1 什么是用于信号处理的低功耗加速器 (LEA)?

LEA 模块是 MSP430 FRAM MCU 系列中新增的 32 位硬件加速器。此加速器可执行信号处理、矩阵乘法以及 FIR、IIR 和 FFT 等其他运算，这些运算通常需要在应用程序运行时花费大量时间和能量进行计算。LEA 模块是一款低功耗协处理器，无需 CPU 干预即可执行运算，并在运算完成时触发中断。MSP430FR599x MCU 系列具有共计 8KB 的 SRAM，其中 4KB 与 LEA 模块共享，用于存储数据输入、输出和参数。该协处理器依据配置期间提供的命令运行。这些命令是内存输入或输出缓冲器和运算类型的指针。MSP DSP 库中使用 LEA 命令确保最高效和易于使用的运算。有关更多信息，请参阅 5 节“我应如何开始使用 LEA 模块?”。有关 LEA 硬件和 MSP430FR599x MCU 系列的更多信息，请参阅《MSP430FR58xx、MSP430FR59xx、MSP430FR68xx 和 MSP430FR69xx 系列用户指南》。

## 2 LEA 模块是否支持浮点或定点运算?

LEA 模块支持实域和复域中的 16 位和 32 位定点运算。

### 3 哪些器件支持 LEA 模块？

LEA 模块当前可在 MSP430FR599x MCU 系列器件上使用，未来计划扩展至其他器件。有关 MSP430FR599x MCU 系列的更多信息，请访问 [MSP430FR5994 产品页面](#)。

### 4 LEA 模块能带来什么性能优势？基准测试数据在哪里？

相较于不使用加速器实现的基于矢量的软件信号处理算法，LEA 模块具有很多优势。LEA 模块的能效更高，执行多种不同的信号处理算法所需的时钟周期更少。要查看基准测试结果，请参阅《[对 MSP430™ MCU 上低功耗加速器的信号处理能力进行基准测试](#)》。《[设定 MCU 性能的新标准，同时最大限度降低能耗](#)》在较高层面探索了性能优势如何转化为总体系统性能。

### 5 我应如何开始使用 LEA 模块？

为确保易用性和高效率，可以使用[适用于 MSP MCU 的数字信号处理 \(DSP\) 库](#)来访问 LEA 模块的运算。以下几个问题说明了如何利用 MSP DSP 库来开始使用 LEA 模块。

### 6 MSP DSP 库是什么？

德州仪器 (TI) 数字信号处理 (DSP) 库是一组经高度优化的函数，可针对 MSP430 微控制器对定点数字执行许多常见的信号处理运算。该函数集通常用于以最低能耗和极高精度实时完成处理密集型转换的应用。该库可在定点运算中对 MSP 系列固有硬件进行最佳利用，从而实现极大的性能增益。

适用于 MSP MCU 的 DSP 库 (DSPLib) 可供所有 MSP MCU 使用，能够为相应的函数自动选择和使用 LEA 模块。该库还提供预处理器宏，通过自动定位 DSPLib 数据结构并确保正确对齐来方便使用 LEA 模块。有关 DSP 库功能（例如预处理器宏和其他函数）的更多信息，请参阅《[MSP DSP 库 API 指南](#)》。

### 7 如何安装 DSPLib 和 DSPLib GUI？

可通过多种方式安装 DSPLib 和 DSPLib GUI 包。请访问 [MSP DSP 库页面](#)或访问最新版的 [MSP430Ware™](#) 软件。

在 MSP DSP 库主页上，单击“Get Software”并选择适合操作系统的 DSPLib 安装程序。

访问 MSP DSPLib 的另一方式是通过 [MSP430Ware](#) 软件包。下载 MSP430Ware 桌面版，方法是从 MSP430Ware 主页下载并安装到 Code Composer Studio™IDE 目录中，或从 Code Composer Studio IDE App Center 下载。如果运行的是 CCS 版本 6.2 或更高版本，请通过 CCS 中嵌入的新 Resource Explorer 来获取 MSP430Ware。

安装完成后，单击窗口顶部的“View”，单击“Resource Explorer”，然后查找“MSP430Ware”。接下来，单击“Libraries”→“DSPLib”，浏览提供的选项。在“DSPLib”中的“Examples Projects”中，有很多在编译成 MSP430FR599x MCU 系列器件目标时使用 LEA 模块的示例。单击 DSPLib GUI 以启动 DSPLib GUI，然后开始使用系数设计和生成滤波器结构，并将其导出为应用程序的 C 代码。

### 8 我可以从哪里找到展示 LEA 模块的示例？

展示 LEA 模块的示例显示在 MSP DSP 库包中。MSP DSP 库包含在 MSP430Ware v3.60 或更高版本中。可以在“MSP430Ware”>“Libraries”>“DSPLib”>“Example Projects”下找到 MSP DSP 库的示例。还可以通过在线版本的 [TI Resource Explorer](#) 来在线访问这些示例。由于只要目标器件连接外设，MSP DSP 库就会使用 LEA 模块，因此用户无需手动通知 DSP 库使用 LEA 模块。DSP 库可以自动完成所需工作，从而实现快速高效的信号处理。

## 9 如果我习惯使用 Python 或 Matlab 来生成滤波器，怎么办？

MSP DSP 库提供多种 Python 和 Matlab 示例，可帮助您生成各种滤波器系数。转到 MSP DSP 库安装位置的主目录，然后查找文件夹“scripts”。该文件夹包含专为帮助您生成自己的滤波器而设计的 Matlab 和 Python 脚本。请参阅《MSP DSP 库 API 指南》，详细了解 Matlab 和 Python 需要哪些包才能支持这些脚本。

## 10 如何在我的程序中使用 LEA 模块？

默认情况下，MSP DSP 库会检查目标器件的头文件，然后为支持 LEA 模块的 API 启用 LEA 模块。如果 LEA 模块可用，则 API 会设置并启用 LEA 模块，然后在任何函数调用结束后禁用它。要确定某函数是否使用 LEA 模块，请参阅《MSP DSP 库 API 指南》或参阅 16 节。这两个参考均指明了每个 API 是否使用 LEA 模块。

在使用 DSPLib API 之前，先完成以下步骤：

1. 指定输入和输出矢量，将矢量对齐以驻留在共享的 4KB LEA SRAM 内存中。
2. 为所选的一个或多个函数设置参数。
3. 调用所需的函数以执行该函数。

第一步，通过分配位置需要驻留在共享的 4KB LEA SRAM 内存中的数组，指定输入和输出内存位置。这可以通过使用 DSPLib 中提供的 DSPLIB\_DATA 宏来实现。例如，要为一个 256 点复数 FFT 分配内存，数据输入数组应包含 256 字的实值和 256 字的复值，共计 512 字（1024 字节）。因此，将使用以下代码片段。

```
#define SAMPLES                256
DSPLIB_DATA(input,MSP_ALIGN_CMPLX_FFT_Q15(SAMPLES))
_q15 input[SAMPLES*2];
```

MSP\_ALIGN\_CMPLX\_FFT\_Q15(SAMPLES) 是一个宏，有助于根据样本数自动计算 16 位复数 FFT 的对齐方式。就如何正确对齐 MSP DSP 库中其他函数的数据，如需了解更多深入示例，请参阅 MSP DSP 库中的示例以及《API 指南》。

在应用程序分配内存后，必须设置所选函数的参数。这些参数在不同函数之间可能不同，并在《MSP DSP 库 API 指南》中进行了介绍。

在设置合适的参数后，调用所需的函数，然后库将处理剩余事项并返回结果。

## 11 如何使用 DSPLib GUI 生成我自己的滤波器系数？

要生成滤波器系数，请打开 DSPLib GUI，在屏幕左侧的“Design”面板中设计滤波器，然后转到顶部“File”下拉菜单 → “Export Filter”。在指定的位置将会以“Design”面板中给出的名称创建一个 .c 文件和一个 .h 文件。在 DSPLib GUI 中设计滤波器的一种简单方法是在“Example Filters”下拉菜单中选择一个预先设计的滤波器（请参阅图 1）。

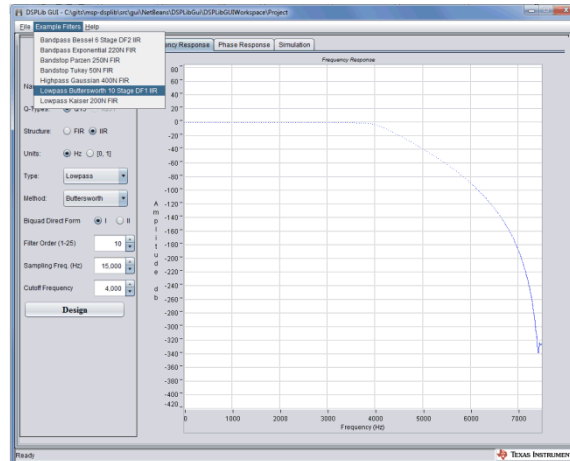


图 1. 在 DSPLib GUI 中选择预定义的低通巴特沃斯 IIR 滤波器

## 12 如何将我在 DSPLib GUI 中生成的系数应用到代码？

要在代码中使用滤波器系数，请使用顶部的“File”下拉菜单 → “Export Filter”，将滤波器从 DSPLib GUI 中导出，然后从 DSPLib 中打开一个示例项目。右键单击该项目，单击“Add files”，添加生成的 filter\_name.c 和 filter\_name.h 文件，然后将代码行 #include "filter\_name.h" 添加到源代码中（请参阅图 2）。然后便可以在 filter\_name 下引用生成的系数。

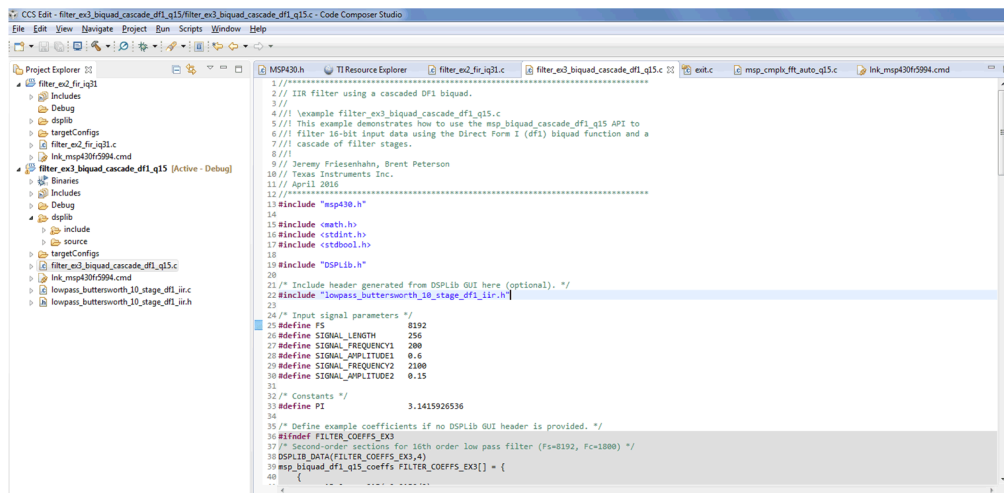


图 2. 将生成的滤波器添加到 CCS DSPLib 示例项目中

### 13 TI 还有其他哪些利用 LEA 模块的配套资源，比如 TI Designs 参考设计？

TI 不断扩充一个称为 TI Designs 的示例 应用系列，旨在展示各种 TI 技术。就 MSP430FR599x MCU 系列而言，TI 当前推出了以下 TI Designs 参考设计，未来还将推出更多。

[使用 MSP430 FRAM 微控制器的滤波与信号处理参考设计](#) TI 设计展示了 MSP430 FRAM 微控制器 (MCU) 上的低功耗加速器 (LEA) 在执行高级滤波和信号处理以及在 16 位 MCU 上维持极低功耗的出色性能。针对 256 点复数 FFT，LEA 模块相较传统 C 实现方式将速度提高了 13.8 倍。LEA 模块还以 20kHz 的高音频采样率提供实时 FIR 滤波性能。

[使用 MSP430 FRAM 微控制器进行 EEPROM 仿真和传感](#) TI 设计描述了如何通过使用 MSP430 超低功耗微控制器 (MCU) 上使用铁电随机存取存储器 (FRAM) 技术并结合采用在使用 MCU 时可启用的其他传感功能来实现 EEPROM 仿真。此参考设计同时支持通过 I<sup>2</sup>C 和 SPI 接口连接至主机处理器，可进行多个从机寻址。

随着其他支持 LEA 模块的 TI Designs 参考设计的发布，这些设计将可通过 [TI Designs 网站](#) 进行访问。

### 14 如果我还有关于 LEA 的问题，应如何寻求帮助？

如果您仍有关于 LEA 模块的问题，请在 [TI E2E™ 社区论坛](#) 上搜索答案。MSP MCU 有一个专门的 E2E 论坛，用于解答有关 MSP 的问题。除拥有一个非常乐于助人和知识渊博的 MSP 粉丝社区外，TI 还拥有一支 MSP 工程师团队，他们致力于提供器件相关支持，确保您可以获得快速而有意义的响应。请访问 <https://e2e.ti.com/support/microcontrollers/msp430/>。

### 15 如果使用 LEA 模块进行计算，我的函数需要多少能量或几个周期？

无论是在 16 位平台上还是在一些 32 位平台上，与软件支持的算法相比，LEA 模块在计算各种信号处理算法方面的整体效率更高，速度也更快。《[对 MSP430™ MCU 上低功耗加速器的信号处理能力进行基准测试](#)》列出了 API 以及 LEA 模块计算函数所需时间的计算公式。请记住，在每次计算中，都会有一小部分时间用于启用和禁用 LEA 模块。

### 16 在 DSPLib API 中，哪些 API 使用 LEA 函数？LEA 模块支持哪些函数？

函数	说明	LEA 支持
msp_add_q15	两个实数源矢量相加	是
msp_add_iq31	两个实数源矢量相加	是
msp_sub_q15	两个实数源矢量相减	是
msp_sub_iq31	两个实数源矢量相减	是
msp_mpy_q15	两个实数源矢量相乘	是
msp_mpy_iq31	两个实数源矢量相乘	是
msp_mac_q15	实数源矢量相乘并累加	是
msp_mac_iq31	实数源矢量相乘并累加	是
msp_neg_q15	源矢量取反	是
msp_neg_iq31	源矢量取反	是
msp_abs_q15	实数源矢量的绝对值	否
msp_abs_iq31	实数源矢量的绝对值	否
msp_offset_q15	实数源矢量的恒定偏移	是
msp_offset_iq31	实数源矢量的恒定偏移	是
msp_scale_q15	缩放实数源矢量	否
msp_scale_iq31	缩放实数源矢量	否
msp_shift_q15	实数源矢量按位移位	是
msp_shift_iq31	实数源矢量按位移位	是

函数	说明	LEA 支持
mzp_max_q15	源矢量的有符号最大值	是
mzp_max_iq31	源矢量的有符号最大值	是
mzp_max_uq15	源矢量的无符号最大值	是
mzp_max_uq31	源矢量的无符号最大值	是
mzp_min_q15	源矢量的有符号最小值	是
mzp_min_iq31	源矢量的有符号最小值	是
mzp_min_uq15	源矢量的无符号最小值	是
mzp_min_uq31	源矢量的无符号最小值	是
mzp_cmplx_add_q15	两个复数源矢量相加	是
mzp_cmplx_add_iq31	两个复数源矢量相加	是
mzp_cmplx_sub_q15	两个复数源矢量相减	是
mzp_cmplx_sub_iq31	两个复数源矢量相减	是
mzp_cmplx_mpy_q15	两个复数源矢量相乘	是
mzp_cmplx_mpy_iq31	复数源矢量相乘	否
mzp_cmplx_mpy_real_q15	复数源矢量与实数源矢量相乘	否
mzp_cmplx_mpy_real_iq31	复数源矢量与实数源矢量相乘	是
mzp_cmplx_mac_q15	复数源矢量相乘并累加	是
mzp_cmplx_mac_iq31	复数源矢量相乘并累加	否
mzp_cmplx_conj_q15	源矢量共轭	否
mzp_cmplx_conj_iq31	源矢量共轭	是
mzp_cmplx_scale_q15	缩放复数源矢量	否
mzp_cmplx_scale_iq31	缩放复数源矢量	否
mzp_cmplx_shift_q15	复数源矢量按位移位	是
mzp_cmplx_shift_iq31	复数源矢量按位移位	是
mzp_matrix_add_q15	两个实数源矩阵相加	是
mzp_matrix_add_iq31	两个实数源矩阵相加	是
mzp_matrix_sub_q15	两个实数源矩阵相减	是
mzp_matrix_sub_iq31	两个实数源矩阵相减	是
mzp_matrix_mpy_q15	两个实数源矩阵相乘	是
mzp_matrix_mpy_iq31	两个实数源矩阵相乘	否
mzp_matrix_trans_q15	源矩阵转置	否
mzp_matrix_trans_iq31	源矩阵转置	否
mzp_matrix_neg_q15	源矩阵取反	是
mzp_matrix_neg_iq31	源矩阵取反	是
mzp_matrix_abs_q15	实数源矩阵的绝对值	否
mzp_matrix_abs_iq31	实数源矩阵的绝对值	否
mzp_matrix_offset_q15	实数源矩阵的恒定偏移	是
mzp_matrix_offset_iq31	实数源矩阵的恒定偏移	是
mzp_matrix_scale_q15	缩放实数源矩阵	否
mzp_matrix_scale_iq31	缩放实数源矩阵	否
mzp_matrix_shift_q15	实数源矩阵按位移位	是
mzp_matrix_shift_iq31	实数源矩阵按位移位	是
mzp_fir_q15	使用实系数对源矢量执行离散时间卷积以应用 FIR 滤波器	是
mzp_fir_iq31	使用实系数对源矢量执行离散时间卷积以应用 FIR 滤波器	是
mzp_cmplx_fir_q15	使用复系数对复数源矢量执行离散时间卷积以应用 FIR 滤波器	是
mzp_cmplx_fir_iq31	使用复系数对复数源矢量执行离散时间卷积以应用 FIR 滤波器	是
mzp_biquad_df1_q15	二阶直接形式 1 双二阶滤波器	是
mzp_biquad_df2_q15	二阶直接形式 2 双二阶滤波器	是

函数	说明	LEA 支持
msp_biquad_df2_ext_q15	二阶直接形式 2 双二阶滤波器, 使用直流偏置以及最小和最大跟踪进行扩展	是
msp_biquad_cascade_df1_q15	级联直接形式 1 双二阶滤波器	是
msp_biquad_cascade_df2_q15	级联直接形式 2 双二阶滤波器	是
msp_biquad_cascade_df2_ext_q15	级联直接形式 2 双二阶滤波器, 使用直流偏置以及最小和最大跟踪进行扩展	是
msp_fft_auto_q15	有自动缩放功能的实数正向 FFT 函数	是
msp_fft_fixed_q15	有固定缩放功能 (每级缩放 2) 的实数正向 FFT 函数	是
msp_fft_iq31	无缩放功能的实数正向 FFT 函数	是
msp_ifft_auto_q15	有自动缩放功能的实数结果逆向 FFT 函数	是
msp_ifft_fixed_q15	有固定缩放功能 (每级缩放 2) 的实数结果逆向 FFT 函数	是
msp_ifft_iq31	无缩放功能的实数结果逆向 FFT 函数	是
msp_cmplx_fft_auto_q15	有自动缩放功能的复数正向 FFT 函数	是
msp_cmplx_fft_fixed_q15	有固定缩放功能 (每级缩放 2) 的复数正向 FFT 函数	是
msp_cmplx_fft_iq31	无缩放功能的复数正向 FFT 函数	是
msp_cmplx_ifft_auto_q15	有自动缩放功能的复数逆向 FFT 函数	是
msp_cmplx_ifft_fixed_q15	有固定缩放功能 (每级缩放 2) 的复数逆向 FFT 函数	是
msp_cmplx_ifft_iq31	无缩放功能的复数逆向 FFT 函数	是
msp_copy_q15	实数 Q15 矢量复制	是
msp_copy_iq31	实数 IQ31 矢量复制	是
msp_fill_q15	用常数填充实数 Q15 矢量	是
msp_fill_iq31	用常数填充实数 IQ31 矢量	是
msp_cmplx_fill_q15	用常数填充复数 Q15 矢量	是
msp_cmplx_fill_iq31	用常数填充复数 IQ31 矢量	是
msp_deinterleave_q15	从多通道源中提取单个通道	是
msp_deinterleave_iq31	从多通道源中提取单个通道	是
msp_cmplx_bitrev_q15	复数位反转函数	是
msp_cmplx_bitrev_iq31	复数位反转函数	是
msp_cmplx_q15	将一个 Q15 实数矢量转换为拥有复数分量	是
msp_cmplx_iq31	将一个 IQ31 实数矢量转换为拥有复数分量	是
msp_interleave_q15	将单个通道插入一个多通道目标	是
msp_interleave_iq31	将单个通道插入一个多通道目标	是
msp_iq31_to_q15	将 IQ31 矢量转换为 Q15 格式	是
msp_q15_to_iq31	将 Q15 矢量转换为 IQ31 格式	是
msp_sinusoid_q15	生成具有指定振幅和频率的正弦信号	是

## 重要声明和免责声明

TI 均以“原样”提供技术性 & 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，不保证其中不含任何瑕疵，且不做任何明示或暗示的担保，包括但不限于对适销性、适合某特定用途或不侵犯任何第三方知识产权的暗示担保。

所述资源可供专业开发人员应用 TI 产品进行设计使用。您将对以下行为独自承担全部责任：(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安全、安保或其他要求。所述资源如有变更，恕不另行通知。TI 对您使用所述资源的授权仅限于开发资源所涉及 TI 产品的相关应用。除此之外不得复制或展示所述资源，也不提供其它 TI 或任何第三方的知识产权授权许可。如因使用所述资源而产生任何索赔、赔偿、成本、损失及债务等，TI 对此概不负责，并且您须赔偿由此对 TI 及其代表造成的损害。

TI 所提供产品均受 TI 的销售条款 (<http://www.ti.com.cn/zh-cn/legal/termsofsale.html>) 以及 [ti.com.cn](http://www.ti.com.cn) 上或随附 TI 产品提供的其他可适用条款的约束。TI 提供所述资源并不扩展或以其他方式更改 TI 针对 TI 产品所发布的可适用的担保范围或担保免责声明。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122  
Copyright © 2019 德州仪器半导体技术（上海）有限公司



## 重要声明和免责声明

TI 均以“原样”提供技术性数据（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，不保证其中不含任何瑕疵，且不做任何明示或暗示的担保，包括但不限于对适销性、适合某特定用途或不侵犯任何第三方知识产权的暗示担保。

所述资源可供专业开发人员应用TI 产品进行设计使用。您将对以下行为独自承担全部责任：(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安全、安保或其他要求。所述资源如有变更，恕不另行通知。TI 对您使用所述资源的授权仅限于开发资源所涉及TI 产品的相关应用。除此之外不得复制或展示所述资源，也不提供其它TI 或任何第三方的知识产权授权许可。如因使用所述资源而产生任何索赔、赔偿、成本、损失及债务等，TI 对此概不负责，并且您须赔偿由此对TI 及其代表造成的损害。

TI 所提供产品均受TI 的销售条款 (<http://www.ti.com.cn/zh-cn/legal/termsofsale.html>) 以及ti.com.cn上或随附TI产品提供的其他可适用条款的约束。TI提供所述资源并不扩展或以其他方式更改TI 针对TI 产品所发布的可适用的担保范围或担保免责声明。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122  
Copyright © 2019 德州仪器半导体技术（上海）有限公司