

## **CC3200 低功耗机制及其在可视门铃中的应用**

---

蒋南

WCS FAE

### **概述**

智能可视门铃系统由于需要采用电池供电，同时需要保持长期的无线网络待机，故要使用低功耗的网络处理器。CC3200 具有超低功耗模式，非常适合此类应用。本文介绍了 CC3200 的休眠以及唤醒机制，阐述了如何实现应用系统在运行模式 ( ACTIVE ) 和超低功耗模式(LPDS)模式之间切换，以获得更小的代码，和更低的应用系统功耗。

## Contents

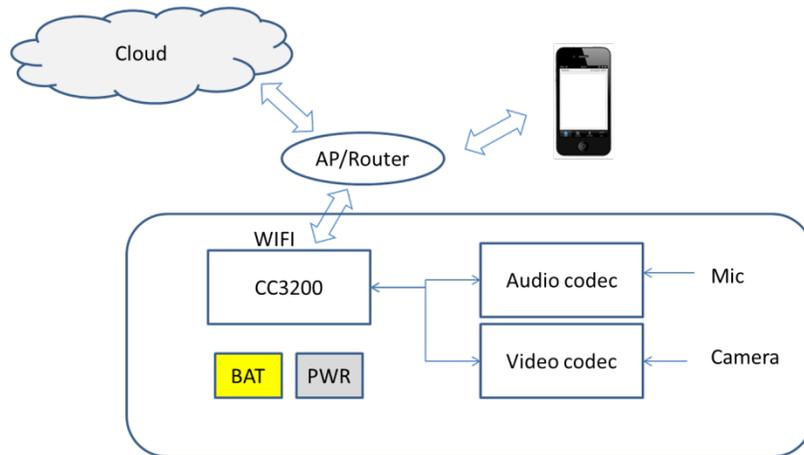
<b>1.</b>	<b>CC3200 简介</b> .....	<b>3</b>
<b>2.</b>	<b>智能可视门铃系统</b> .....	<b>3</b>
<b>3.</b>	<b>CC3200 的 LPDS 休眠/唤醒机制</b> .....	<b>4</b>
3.1.	IO 口唤醒.....	4
3.2.	休眠定时器中断唤醒 .....	4
3.3.	NWP 中断唤醒 .....	4
<b>4.</b>	<b>CC3200 的 LPDS 休眠/唤醒步骤</b> .....	<b>5</b>
<b>5.</b>	<b>LPDS 应用程序调用接口 ( API )</b> .....	<b>6</b>
5.1.	PRCMSRAMRetentionEnable( ulSramColSel, ulFlags) ; .....	6
5.2.	PRCMLPDSWakeupSourceEnable(ulLpdsWakeupSrc) .....	6
5.3.	PRCMLPDSRestoreInfoSet( ulStackPtr, ulProgCntr ).....	6
5.4.	PRCMLPDSIntervalSet(ulTicks) .....	7
5.5.	PRCMLPDSEnter(void).....	7
<b>6.</b>	<b>LPDS 实例</b> .....	<b>7</b>
<b>7.</b>	<b>关键代码解析</b> .....	<b>8</b>
<b>8.</b>	<b>References</b> .....	<b>10</b>

## 1. CC3200 简介

CC3200 是德州仪器针对物联网应用开发的一款微控制器，其在一颗芯片内部集成了 CORTEX-M4 内核，WIFI 网络处理器，256K RAM 以及大量外设。在 IOT 应用中，系统的功耗是一个重要的指标，CC3200 支持四种不同的运行模式：ACTIVE, SLEEP, DEEPSLEEP, LPDS(Low Power Dee Sleep)以达到在不同的场合整体最低功耗的目标。在这四种运行模式中，LPDS 模式提供在低于 1mA 的电流下保持 WIFI 网络连接的能力，因此成为最受关注的特性之一。

本文仅针对 LPDS 特性做深入介绍，其他几种模式相对较简单，不在本文讨论范围，不再赘述。

## 2. 智能可视门铃系统



上图是一个智能型可视门铃的框图，除了具有传统可视门铃的音频视频处理能力外，该智能可视门铃需要能够与智能终端如手机通讯，也需要具备访问云端的能力，从而将访客的信息发送给终端或者云端，也可以被远程访问。在图中可以看出，门铃一般需要由电池供电。而该系统要求门铃保持无线网络待机，当有访客按门铃或者远端需要访问门铃时能够及时响应。由于门铃处于持续工作或者待机状态中，这样就要求主芯片需要能够保持低功耗无线网络待机，前面介绍了 CC3200 拥有超低功耗的待机能力，在此类场合下非常适用。为了获得低功耗待机 (<1mA)，需要使用 CC3200 的 LPDS 模式。

### 3. CC3200 的 LPDS 休眠/唤醒机制

当 CC3200 进入到 LPDS 模式后，所有的外设都将被复位同时失去时钟，MCU 也被挂起，但 NWP 并不受影响并可自行管理功耗以实现在保持网络连接的同时保持低功耗。

CC3200 进入 LPDS 后，可以通过三个途径将其唤醒：外部特定 IO 口中断，休眠定时器中断以及 NWP 中断。

#### 3.1. IO 口唤醒

CC3200 允许使用特定的 IO 口进行休眠唤醒，进入 LPDS 前需要调用 SDK 中提供的 API，选择正确的引脚以及中断唤醒触发方式：电平触发，上升沿触发，下降沿触发。在参数设置时需要注意 API 提供的 GPIO 宏定义与实际引脚的对应关系，在不同的引脚配置中会有所变化。在本例中，对应如下表所示：

API 参数	CC3200 实际引脚
PRCM_LPDS_GPIO2	PIN57
PRCM_LPDS_GPIO4	PIN59
PRCM_LPDS_GPIO13	PIN4
PRCM_LPDS_GPIO17	PIN8
PRCM_LPDS_GPIO11	PIN2
PRCM_LPDS_GPIO24	PIN1

**Figure -1. 唤醒引脚对照表**

#### 3.2. 休眠定时器中断唤醒

CC3200 内置专用的 32 位 LPDS 定时器，输入时钟为 32.768KHz。如果需要使用 LPDS 定时器唤醒 MCU，在进入 LPDS 前需要设定的唤醒时间并选择 LPDS 定时器做为唤醒源。

#### 3.3. NWP 中断唤醒

CC3200 中 MCU 和 NWP 各自独立工作，仅通过串口(UART/SPI)进行通信。当 MCU 进入 LPDS 时，NWP

仍可能会继续低功耗工作，以保持 WIFI 通信连接。当 NWP 接收到远端数据时，如果被使能，将会产生 NWP 中断，该中断可以用来唤醒 MCU。

#### 4. CC3200 的 LPDS 休眠/唤醒步骤

对于 C3200 中的 MCU 来说，每次从 LPDS 中唤醒实际相当于一次上电复位，不是真正的原地唤醒。但因为 CC3200 在休眠时用户可以选择保存部分或者全部的 SRAM，故可以通过特殊的方法来模拟真正的唤醒，从休眠前的地方继续执行下一条代码。

CC3200 的休眠唤醒过程，如下图所示：

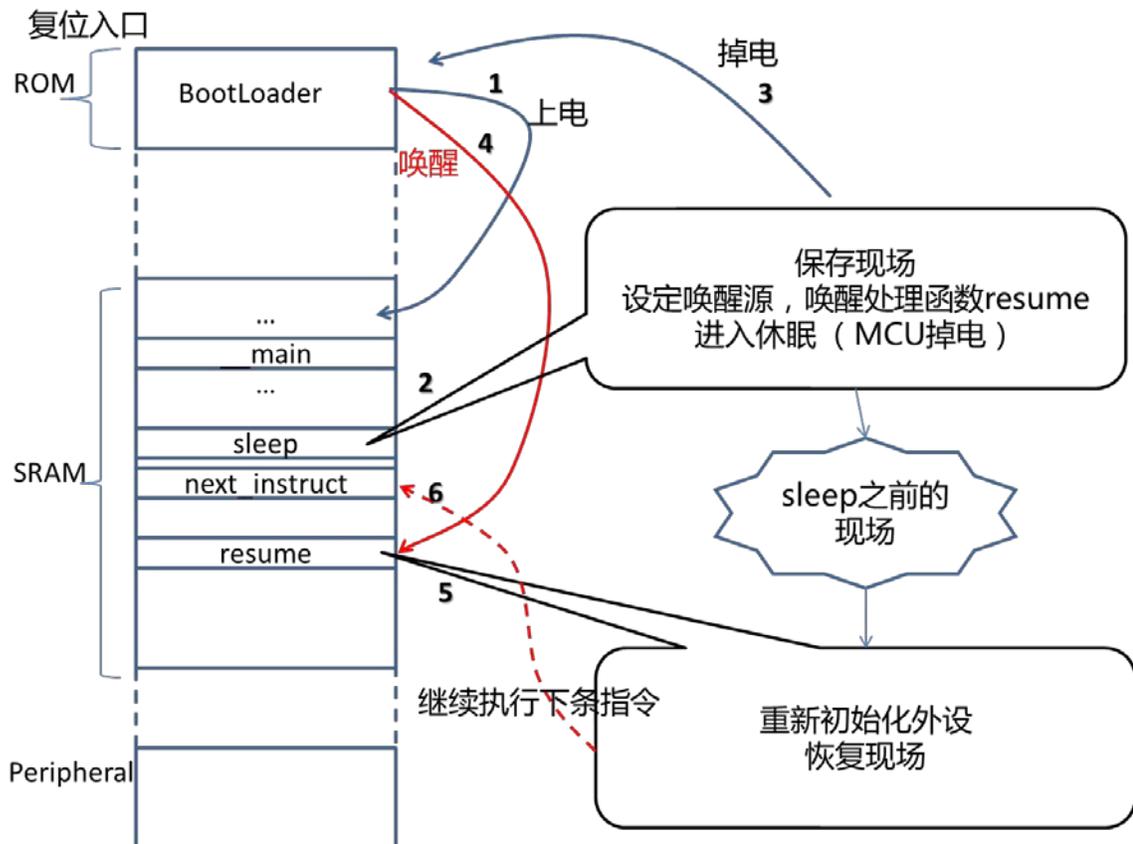


Figure -2. 休眠唤醒步骤分解

上图中的数字表示休眠/唤醒的过程：

1. 当 CC3200 上电时，MCU 从复位入口启动，即从 Bootloader 中开始运行。Bootloader 负责跳转至应用代码主函数入口，开始运行用户代码。
2. 当用户代码需要进入 LPDS 时，需要在执行休眠指令前保存当前现场。现场包括当前使用到的 RAM，相关 NVIC 中断控制寄存器，ARM CORTEX-M4 核心寄存器组（PC,堆栈，寄存器组等）以及已使用外设的状态。同时，需要指定一种或者多种唤醒源，以便在唤醒事件发生时能够唤醒 MCU。除夕之外，还要指定唤醒后的处理函数，以便唤醒后进行现场还原工作。
3. 保存了现场，执行休眠指令，MCU 进入掉电状态。此时所有外设模块都被复位，MCU 复位。
4. 当唤醒事件发生后，MCU 会进入上电状态，从 Bootloader 开始运行。由于在第 2 步指定了唤醒处理函数（resume），故 Bootloader 会把控制权交给 resume 函数（而不是应用代码入口）。
5. 在唤醒处理函数中，原来保存的现场会被恢复，包括外设，NVIC 相关寄存器，ARM 核心寄存器组等。
6. 当现场被恢复后，并从 resume 函数返回时，将从休眠处的下一条指令开始运行（即 next\_intstruct），从而实现了休眠/唤醒这一模拟过程。

## 5. LPDS 应用程序调用接口（API）

德州仪器为 CC3200 提供了开发用 SDK, 当前最新版本为 1.1.0。SDK 提供了休眠控制的 API 接口，下面介绍主要的几个函数：

### 5.1. *PRCMSRAMRetentionEnable( ulSramColSel, ulFlags) ;*

CC3200 的 SRAM 被划分为 4 块，休眠时可以选择部分或者全部 SRAM 继续供电以保持内容不变。本例中选择全部保存（LPDS 下默认值）。

ulSramColSel – 使用该参数选择具体需要供电的 SRAM 块,位或操作。

ulFlags – 该参数用来指定 SRAM 的供电方案仅针对 DS,还是仅针对 LPDS，或者在两种情况下都使用相同的方案。

### 5.2. *PRCMLPDSWakeupSourceEnable(ulLpdsWakeupSrc)*

该函数用来选择 LPDS 唤醒的事件，可能是一种或者多种事件的组合。如前所述，事件包括 NWP,GPIO,LPDS Timer 三种。

### 5.3. *PRCMLPDSRestoreInfoSet( ulStackPtr, ulProgCntr )*

如前所述，在进入 LPDS 休眠前，需要保存当前现场，其中当前堆栈指针由 ulStackPtr 给出，注意的是如果使用了 RTOS，该值可能是 MSP 或者 PSP,视当前处于主进程/线程而定；同时，唤醒处理函数

指针由 ulProgCntr 给出。一旦发生唤醒事件，MCU 复位后，Bootloader 将根据这两个参数，跳转到唤醒处理函数。

#### 5.4. PRCMLPDSIntervalSet(ulTicks)

如果使用 LPDS 定时器中断做为唤醒触发事件，则需要为 LPDS 专用定时器设置溢出时间(参数 ulTicks)。ulTicks 以 1/32768 秒为单位计数，故 ulTicks 设为 32768 时，唤醒事件将在休眠后 1 秒钟后发生。

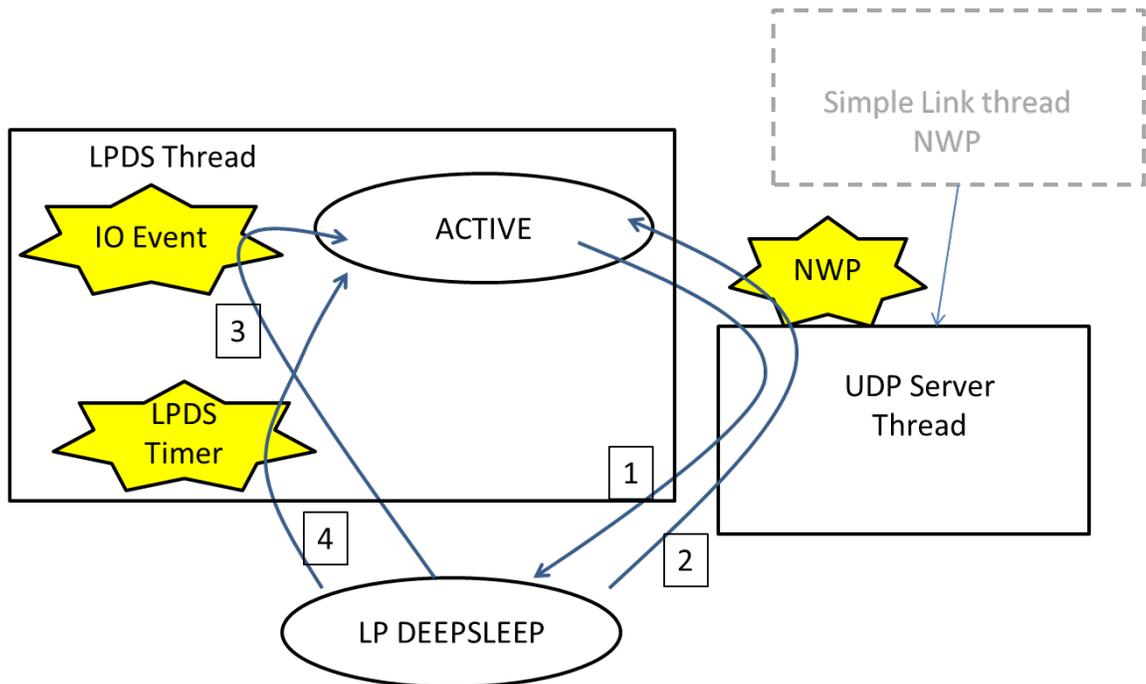
#### 5.5. PRCMLPDSEnter(void)

一旦执行本函数，CC3200 将进入 LPDS 模式。因此该函数应该是休眠前最后调用的一个函数，并且永远不会返回。

### 6. LPDS 实例

本文通过一个实例来说明如何实现 CC3200 的 LPDS,该例程使用 FreeRtos 并通过 LPDS 监控线程和 UPD Server 线程来演示休眠及如何通过 GPIO/LPDS Timer/NWP 唤醒。

下图描述了在例程中，MCU 的状态如何在不同事件作用下发生转换。



**Figure -3. 状态转换图**

上图中的数字表示了状态切换过程：

1. 应用程序运行的初始状态为 ACTIVE, UDP 服务器等待外部 UDP 数据包，一旦接收到 UDP 数据，则 UDP 服务线程发消息给 LPDS 线程，LPDS 线程执行 LPDS 命令，MCU 进入休眠。
2. 当 MCU 进入休眠后，NWP 任然在工作，如果 NWP 接收到 UPD 数据，则产生 NWP 事件，MCU 被唤醒，返回 LPDS 线程. LPDS 线程保持 ACTIVE 约一秒后，重新执行 LPDS 命令，MCU 进入休眠。
3. 上述 1,2 演示了通过 NWP 来唤醒 MCU 的过程。当 MCU 处于休眠时，如果外部 IO 口产生中断事件，则 MCU 被唤醒，控制权重新返回 LPDS 线程。
4. 当 MCU 处于休眠时，如果在休眠前设定了 LPDS 定时器，则当定时器事件发生时，MCU 被唤醒。本例程中，在进入休眠前，会将 LPDS 定时器溢出时间设定为 30 秒，当系统休眠超过 30 秒后被唤醒并一直保持唤醒，直到下一个 UPD 数据包到来（回到步骤 1）。

## 7. 关键代码解析

在实例代码中，完成休眠和唤醒的函数分别是 sleep 和 resume\_from\_LPDS 函数，下面做一简要介绍，其他代码请参考本文所附源码。

需要重点注意的是，在 resume\_from\_LPDS 函数中，需要重新初始化外设。由于本例中的 NWP 与 MCU 的通信是通过 SPI, 故在唤醒后需要重新初始化 SPI 接口，以使得 NWP 可以将数据发送给 MCU 从而避免 ARM 总线错误，造成程序锁死。另外，由于本例使用串口做为调试输出接口，故唤醒后也需要重新初始化串口。同理，如果应用程序在休眠前使用了其他外设，也需要——初始化。

```
void sleep() {
    IntMasterDisable();

    PRCMLPDSWakeupSourceEnable(PRCM_LPDS_TIMER);
    PRCMLPDSWakeupSourceEnable(PRCM_LPDS_HOST_IRQ);

    PRCMLPDSWakeupSourceEnable(PRCM_LPDS_GPIO);
    PRCMLPDSWakeUpGPIOSelect(PRCM_LPDS_GPIO13,PRCM_LPDS_RISE_EDGE);

    PRCMLPDSIntervalSet(32768 * 30);
    MAP_PRCMSRAMRetentionEnable(
    PRCM_SRAM_COL_1 | PRCM_SRAM_COL_2 | PRCM_SRAM_COL_3 | PRCM_SRAM_COL_4,
```

```

    PRCM_SRAM_LPDS_RET);

    back_up_nvic_regs();

    INTRODUCE_SYNC_BARRIER()
    BACK_UP_ARM_REGISTERS()
    if (vault_arm_registers.control == 0) {
    MAP_PRCMLPDSRestoreInfoSet(vault_arm_registers.msp, resume_from_LPDS);
    } else {
    MAP_PRCMLPDSRestoreInfoSet(vault_arm_registers.psp, resume_from_LPDS);
    }

    IntMasterEnable();
    PRCMLPDSEnter();
    ... ..
}

void resume_from_LPDS (void) {
    board_init(); //初始化IO,时钟等
    spi_init(); //初始化NWP使用的SPI接口

    INTRODUCE_SYNC_BARRIER()

    RESTORE_ARM_REGISTERS() //恢复ARM内核寄存器组

    INTRODUCE_SYNC_BARRIER()

    cc_restore_soc_data(); //恢复NVIC中断控制寄存器相关寄存器

    MAP_IntMasterEnable();

    debug_port_init(); //初始化调试接口
    return;
}

```

## 8. References

1. *CC3200 SimpleLink Wi-Fi and Internet-of-Things Solution, TRM(SWRU367B)*
2. *CC3200 SimpleLink Wi-Fi and Internet-of-Things Solution, PROG GUIDE(SWRU369B)*
3. *CC32xx Power Management Framework*

## 重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内, 且 TI 认为有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定, 否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予的直接或间接侵权作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从 TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是 TI 的专利权或其它知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时, 如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分, 则会失去相关 TI 组件或服务的所有明示或暗示授权, 且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意, 尽管任何应用相关信息或支持仍可能由 TI 提供, 但他们将独自负责满足与其产品及其应用中使用 TI 产品相关的所有法律、法规和安全相关要求。客户声明并同意, 他们具备制定与实施安全措施所需的全部专业技术和知识, 可预见故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中, 为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此, 此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III (或类似的生命攸关医疗设备) 的授权许可, 除非各方授权官员已经达成了专门管控此类使用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同意, 对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用, 其风险由客户单独承担, 并且由客户独自负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品, 这些产品主要用于汽车。在任何情况下, 因使用非指定产品而无法达到 ISO/TS16949 要求, TI 不承担任何责任。

	产品		应用
数字音频	<a href="http://www.ti.com.cn/audio">www.ti.com.cn/audio</a>	通信与电信	<a href="http://www.ti.com.cn/telecom">www.ti.com.cn/telecom</a>
放大器和线性器件	<a href="http://www.ti.com.cn/amplifiers">www.ti.com.cn/amplifiers</a>	计算机及周边	<a href="http://www.ti.com.cn/computer">www.ti.com.cn/computer</a>
数据转换器	<a href="http://www.ti.com.cn/dataconverters">www.ti.com.cn/dataconverters</a>	消费电子	<a href="http://www.ti.com.cn/consumer-apps">www.ti.com.cn/consumer-apps</a>
DLP® 产品	<a href="http://www.dlp.com">www.dlp.com</a>	能源	<a href="http://www.ti.com.cn/energy">www.ti.com.cn/energy</a>
DSP - 数字信号处理器	<a href="http://www.ti.com.cn/dsp">www.ti.com.cn/dsp</a>	工业应用	<a href="http://www.ti.com.cn/industrial">www.ti.com.cn/industrial</a>
时钟和计时器	<a href="http://www.ti.com.cn/clockandtimers">www.ti.com.cn/clockandtimers</a>	医疗电子	<a href="http://www.ti.com.cn/medical">www.ti.com.cn/medical</a>
接口	<a href="http://www.ti.com.cn/interface">www.ti.com.cn/interface</a>	安防应用	<a href="http://www.ti.com.cn/security">www.ti.com.cn/security</a>
逻辑	<a href="http://www.ti.com.cn/logic">www.ti.com.cn/logic</a>	汽车电子	<a href="http://www.ti.com.cn/automotive">www.ti.com.cn/automotive</a>
电源管理	<a href="http://www.ti.com.cn/power">www.ti.com.cn/power</a>	视频和影像	<a href="http://www.ti.com.cn/video">www.ti.com.cn/video</a>
微控制器 (MCU)	<a href="http://www.ti.com.cn/microcontrollers">www.ti.com.cn/microcontrollers</a>		
RFID 系统	<a href="http://www.ti.com.cn/rfidsys">www.ti.com.cn/rfidsys</a>		
OMAP应用处理器	<a href="http://www.ti.com.cn/omap">www.ti.com.cn/omap</a>		
无线连通性	<a href="http://www.ti.com.cn/wirelessconnectivity">www.ti.com.cn/wirelessconnectivity</a>	德州仪器在线技术支持社区	<a href="http://www.deyisupport.com">www.deyisupport.com</a>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2016, Texas Instruments Incorporated