

Roy Zhuang
Allen Yin

Central FAE

摘要

毫米波成像雷达是高级驾驶辅助系统 (ADAS)、自动驾驶和工业自动化等领域的核心技术。与传统雷达相比,成像雷达利用多通道天线阵列实现更高的角分辨率,能够生成高精度的点云,从而更准确地识别和分类物体。德州仪器 (TI) 推出的高性能单芯片毫米波雷达 AWR2188 工作在 76~81 GHz 频段,集成了 8 路发射和 8 路接收通道,并内置完整的射频 (RF) 前端、锁相环 (PLL) 和模数转换器 (ADC),实现了高集成度和紧凑封装。本文将详细介绍 AWR2188 芯片的启动流程。我们将剖析从硬件上电到固件加载的全过程,旨在为开发者提供清晰、完整的操作指南,帮助其快速上手 AWR2188 的系统开发,并加速成像雷达应用的部署。

内容

1 AWR2188 介绍	2
2 硬件架构与设计	4
2.1 雷达系统架构.....	4
2.2 AWR2188 雷达硬件设计.....	4
2.3 SOP 引脚配置.....	5
2.4 AWR2188 复位与 RBL 启动流程.....	6
3 AWR2188 的外部处理器软件配置	8
3.1 外部处理器上电配置 MSS.....	8
3.2 DFP 镜像下载.....	9
3.3 MSS 应用程序 (Application) 启动与配置.....	9
4 AWR2188 SPI/I2C 配置与信号解析	11
4.1 SPI 启动配置与时序.....	11
4.2 I2C 启动配置与时序.....	13
5 参考文献	14
6 修改记录	14

插图清单

图 1-1. AWR2188 芯片系统框图.....	3
图 2-1. 卫星雷达架构.....	4
图 2-2. 边缘雷达架构.....	4
图 2-3. 卫星雷达架构硬件设计.....	5
图 2-4. AWR2188 上电时序.....	6
图 2-5. AWR2188 上电时序.....	7
图 3-1. AWR2188 软件执行流程.....	8
图 4-1. AWR2188 SPI 协议时序.....	12
图 4-2. AWR2188 下载 DFP 固件的 CMD_HEADER.....	12
图 4-3. AWR2188 下载 DFP 固件的 CMD_HEADER.....	13

表格清单

表 2-1. 通电检测(SOP)线路和引导模式.....	5
表 3-1. 外部处理器软件的 SOP 配置.....	8
表 3-2. AWR2188 应用程序 (Application) 启动模式配置.....	9
表 3-3. MSS 上电配置.....	9
表 3-4. RSS 上电配置.....	10

表 3-5. w_RssPowerUpCfg 上电配置.....	10
表 4-1. SPI 管脚配置.....	11
表 4-2. SPI 帧数据格式.....	11
表 4-3. SPI_CMD_ID.....	11
表 4-4. I2C 硬件管脚.....	13

1 AWR2188 介绍

成像雷达市场正处于高性能需求驱动的快速演进阶段，特别是在高级驾驶辅助系统 (ADAS)、自动驾驶领域和工业应用。早期，为了克服传统雷达分辨率低、无法形成精细点云的局限，行业开始探索成像雷达方案。以德州仪器 (TI) 第二代 AWR2243 毫米波传感器为例，需要通过多芯片级联的方式来增加天线数量，从而获得更多虚拟通道，以满足更高性能的成像需求，例如更远距离、更高分辨率。这种级联方案在性能上实现了突破，但同时也带来了系统复杂性的增加、功耗上升以及更长的开发周期等挑战。

市场的驱动力促使技术向更高集成度、更优性价比的方向发展，以实现成像雷达的普及。在此背景下，TI 推出了新一代 AWR2188 毫米波雷达芯片，代表了雷达技术向单芯片高性能集成的发展方向。AWR2188 集成了多达 8 个发射器和 8 个接收器，在单芯片上即可在通道数量、性能和功耗等方面优于两片 AWR2243 级联方案，从而显著简化了硬件设计，降低了系统成本和功耗。这一架构上的根本性创新，使得高性能成像雷达的开发变得更加高效和便捷，加速了其在各类应用场景中的部署。

AWR2188 的问世，不仅标志着成像雷达技术在集成度上的飞跃，也为自动驾驶和工业应用提供了更高效、更可靠的感知解决方案。其高通道数设计能够提供更精细的点云数据，提升对环境中小型物体、复杂场景的识别和区分能力。此外，单芯片集成带来的优势也体现在尺寸更小、系统级功耗更低和生产效率更高等方面，这对于空间受限的汽车电子和成本敏感的工业应用尤为关键。因此，AWR2188 的出现不仅是一次技术上的迭代，更是对市场需求的一次精准回应，为下一代高性能前向成像雷达的广泛应用铺平了道路。

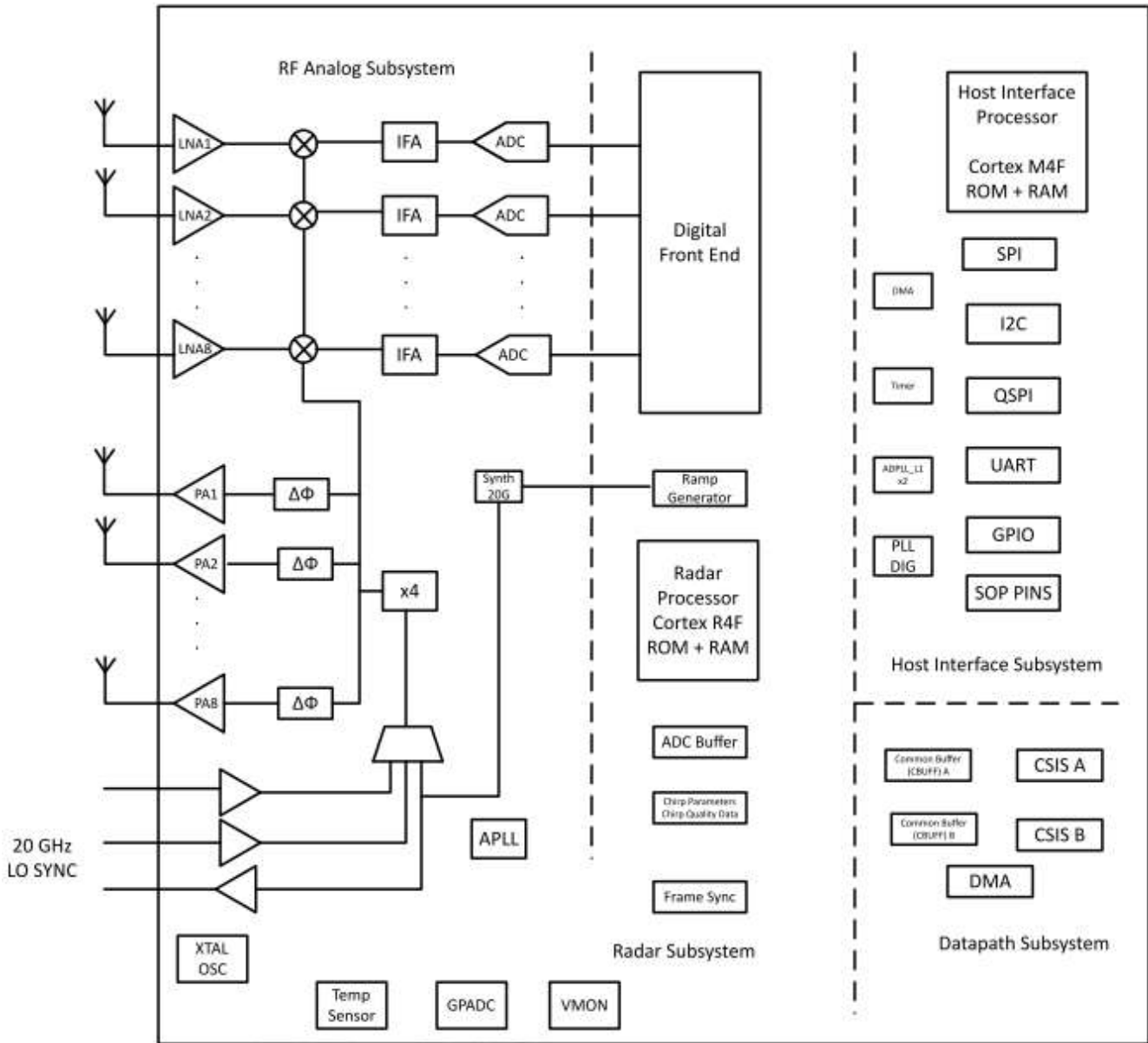


图 1-1. AWR2188 芯片系统框图

AWR2188 内部主要有以下部分构成：

- 毫米波射频模块：主要包括射频收发器和采样模块，包含接收低噪声放大器 (LNA)、发射放大器 (PA)、混频器、模数转换 (ADC)、本振发生器 (Synthesizer) 和锁相环 (APLL) 等；
- 雷达前端控制器：射频控制器的固件运行于内置的 Cortex R4F (RSS, Radar Subsystem 雷达子系统) 处理器，用于控制射频前端的参数配置、校准、监控，并对采样信号进行抽取滤波；
- 主机接口控制模块：内置 Cortex M4F (MSS) 运行 TI 提供的固件，用于提供 SPI 或者 I2C 接口与外部处理器进行通信，并通过 DMA 在 CSI 接口上发送 ADC 数据，以及与雷达前端控制器进行交互；

本文将介绍 AWR2188 的硬件配置以及启动过程，旨在最大程度地帮助客户快速上手和调试 AWR2188。

2 硬件架构与设计

2.1 雷达系统架构

目前市场上主流的毫米波雷达系统有两种：一种是卫星雷达架构。毫米波前端传感器芯片 MMIC 将采集到的原始数据直接通过高速接口 (Serdes/Ethernet) 传递给车身的中央域控处理器，在中央域控进行原始数据的处理。另一种是边缘雷达架构。即 MMIC 与处理器集成在一个板子或同一个芯片上，MMIC 将采集到的原始数据交给处理器处理成点云或目标数据，再通过 CAN-FD 或者以太网接口传递给车身的中央域控处理器进行下一步的判决。图 2-1 和 图 2-2 是这两种架构硬件框图。

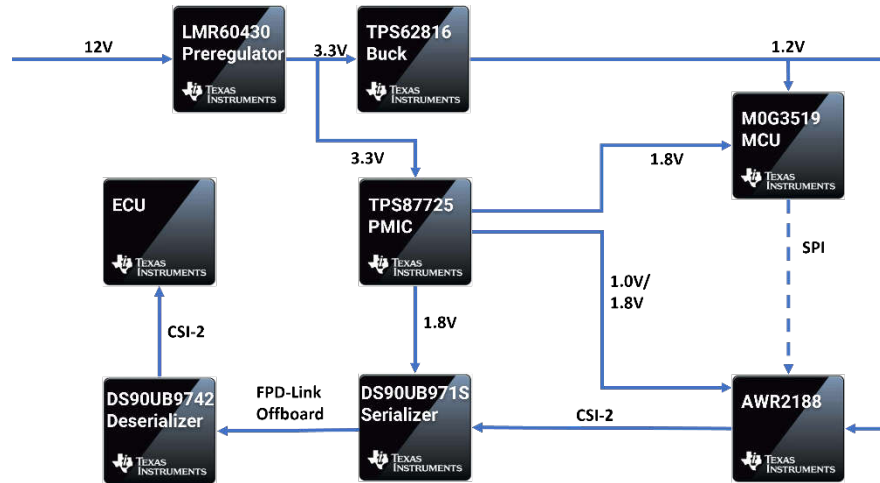


图 2-1. 卫星雷达架构

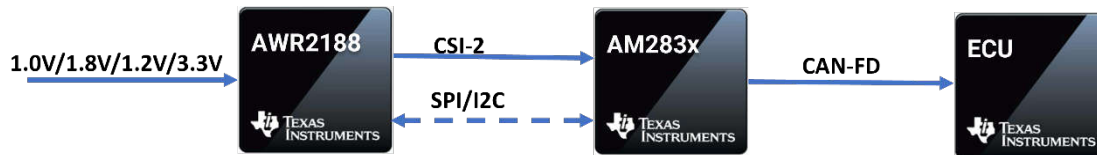


图 2-2. 边缘雷达架构

2.2 AWR2188 雷达硬件设计

面以 AWR2188 的卫星架构为例，介绍其硬件设计。图 2-3 中画出了 AWR2188 所需的全部信号引脚和电源相关器件(其中，TPS628503，TPS628501，TLV70723，LMR60430 是降压转换器，M0G3519 是 MCU 外部微控制器，LP87745 是 PMIC 电源管理芯片，DS90B971 是 FPDLINK 加串器)，其中连接 AWR2188 所必须连接的管脚有：nRESET、nError、SPI_BUS、HOST_IRQ、SPI_MOSI、SPI_MISO 和 SPI_CS (如果是 I2C 模式，详见 4.2 节 I2C 配置)。

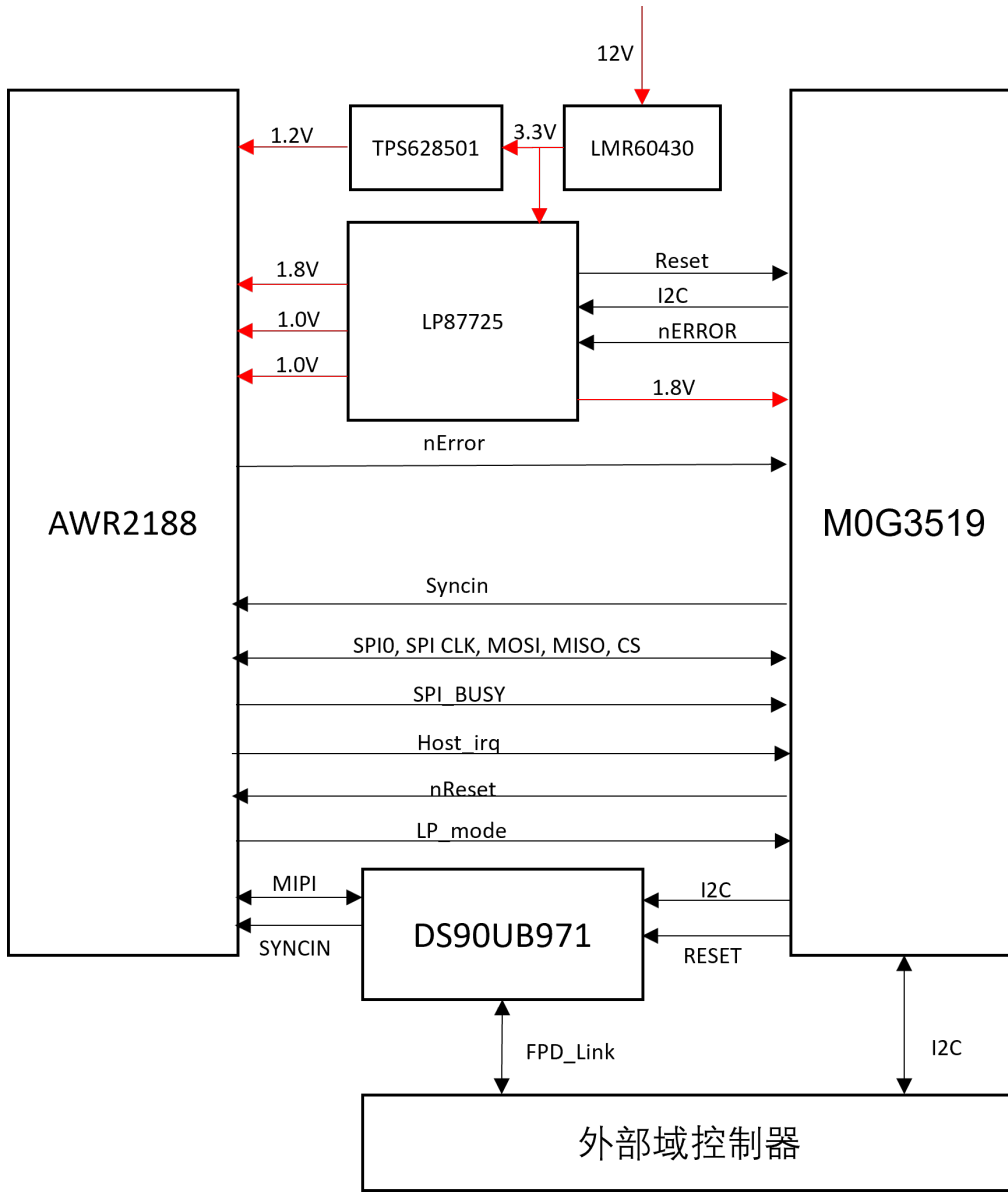


图 2-3. 卫星雷达架构硬件设计

更详细的原理图和参考设计请参考 AWR2188 卫星雷达硬件设计 [4]。

2.3 SOP 引脚配置

AWR2188 上电后的首个操作是读取通电检测 (SOP) 引脚的状态。表 2-1 为 AWR2188 各 SOP 模式以及对应的管脚。

表 2-1. 通电检测(SOP)线路和引导模式

功能	信号	介绍	管脚序号
SOP - SOP[4:0]	SOP0 (与 TDO 复用)	SOP 0 - controls boot behavior	AF8
	SOP1 (与 FRM_SYNCOUT 复用)	SOP 1 - controls boot behavior	AF3
	SOP2 (与 PMIC_CLKOUT 复用)	SOP 2 - controls boot behavior	AG5
	SOP3 (与 MCU_CLKOUT 复用)	SOP 3 - controls cascade/non-cascade	AG6
	SOP4 (与 QSPI_0 复用)	SOP 4 - controls 40 MHz or 50 MHz crystal	AG27

SOP 模式	介绍	SOP[2:0]
功能模式 SPI	主要部署模式。通过 SPI 从外部处理器或者通过 QSPI 从 Flash 加载固件补丁，功能固件开始执行，设备通过 SPI 命令进行控制	0 0 1
设备管理模式	闪存编程模式。镜像文件（补丁）通过烧写工具下载到串行闪存(SFLASH)中，该工具通过 UART 传输镜像文件	1 0 1
功能模式 I2C	通过 I2C 加载固件补丁，然后功能固件开始执行	1 1 1

SOP 模式	介绍	SOP[3]
级联模式	0: 单片 AWR2188 1: 多片级联模式	1 或者 0

SOP 模式	介绍	SOP[4]
晶振选择	0: 40MHz 1: 50MHz	1 或者 0

2.4 AWR2188 复位与 RBL 启动流程

AWR2188 上电后，在 nReset 信号由低变高过程中，会将 SOP 硬件配置记录下来。复位结束后将会进入 RBL (ROM Bootloader ，上电后从 TI 固化的 ROM 区域执行 Bootloader) 阶段。

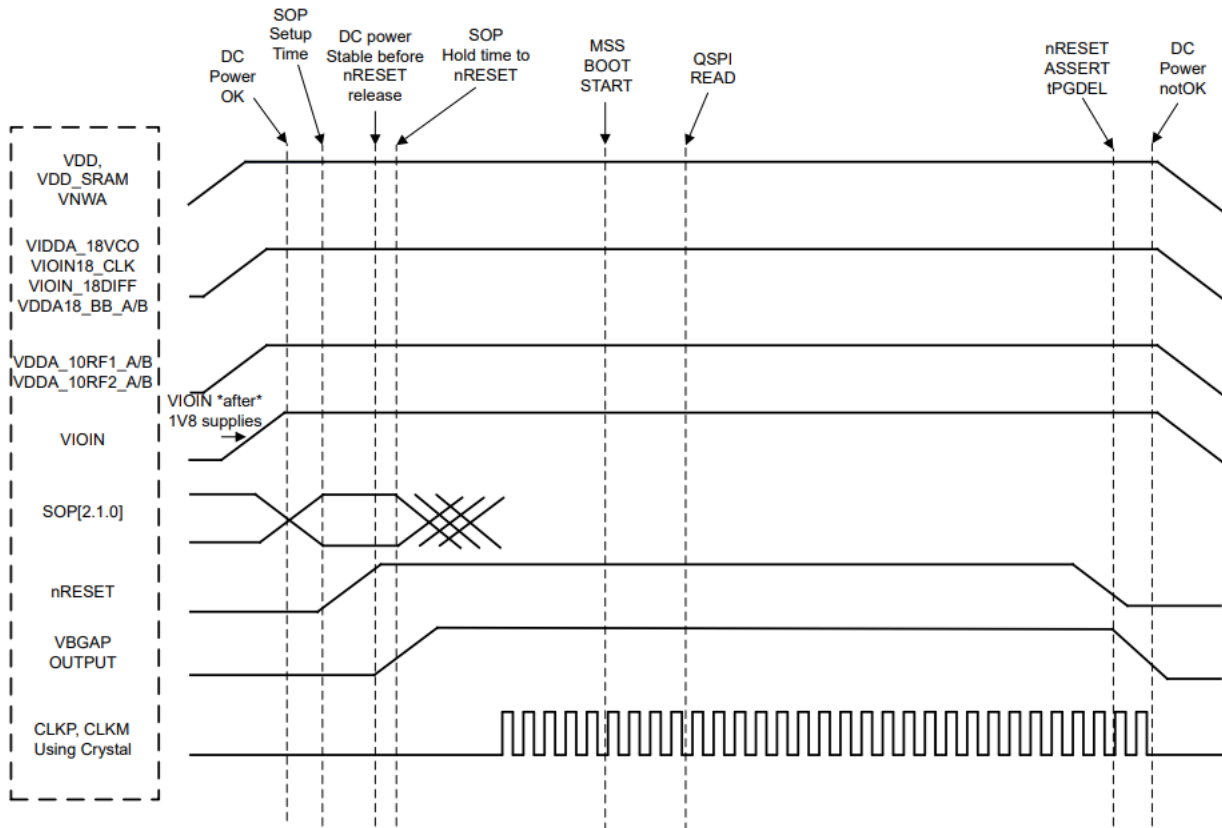


图 2-4. AWR2188 上电时序

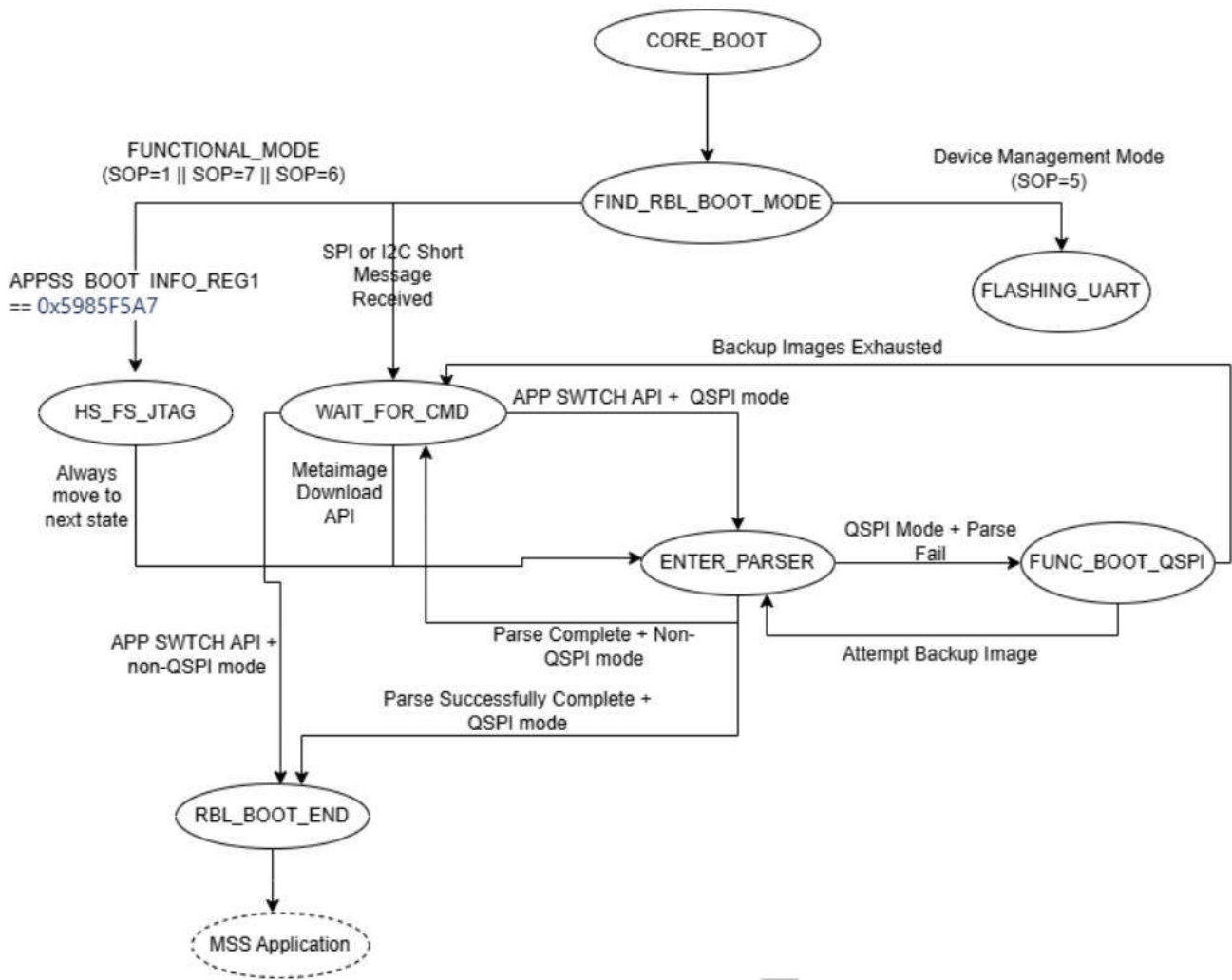


图 2-5. AWR2188 上电时序

整个 RBL 阶段的流程图如 图 2-5 所示，在 SPI/I2C 功能模式下，AWR2188 的 nReset 由低变高后 AWR2188 进入 RBL 阶段。RBL 会根据用户的配置选择使用 QSPI 从 Flash 还是使用 SPI/I2C 从外部处理器下载 AWR2188 固件。固件下载完毕后，AWR2188 外部处理器会根据软件配置设置应用程序从 QSPI 启动还是通过 SPI/I2C 从外部处理器启动。

3 AWR2188 的外部处理器软件配置

为了帮助用户更好地理解 AWR2188 的启动流程与配置，本章以 M0 作为外部处理器为例，介绍 AWR2188 在 SPI (SOP = 001) 模式下的外部处理器软件配置。图 3-1 给出了 AWR2188 上电启动后的软件执行顺序。

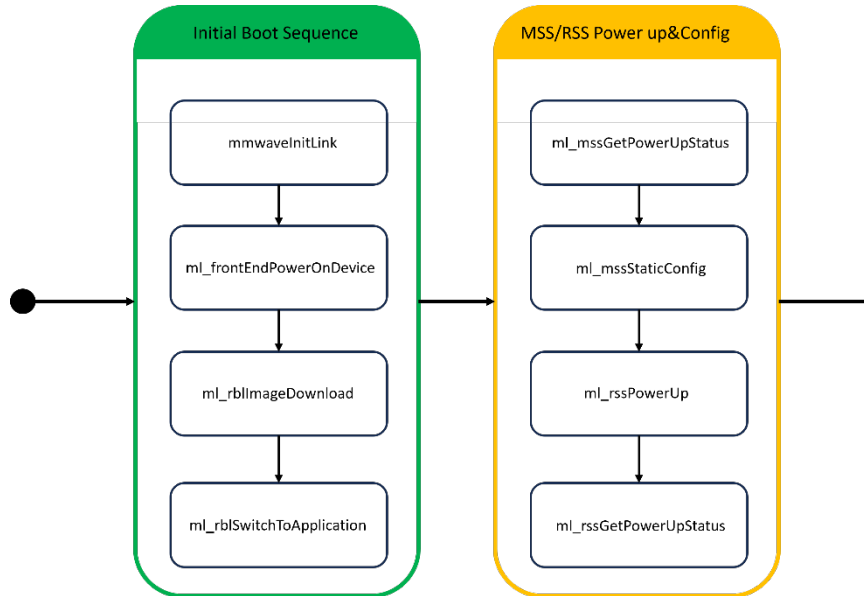


图 3-1. AWR2188 软件执行流程

图中的函数是在外部处理器中执行的，通过调用 mmWaveLink 框架中的接口函数来控制 AWR2188 中的各项功能。AWR2188 运行的固件包和运行代码均包含在 DFP (Device Firmware Package，设备固件包，包含固件、RSS 和 MSS 系统需要执行的代码) 中。

3.1 外部处理器上电配置 MSS

在软件侧，AWR2188 上电后外部处理器软件的第一个操作是通过 `ml_frontEndPowerOnDevice` 函数对 AWR2188 进行复位 (控制 AWR2188 的 `nReset` 引脚电平从高到低再到高)，并在复位期间通过控制 GPIO 输出电平，配置 AWR2188 的 SOP 和时钟，其中 SOP 的配置见 表 3-1，Clock 的配置见 表 2-1。此函数执行完毕后 AWR2188 离开复位状态正式进入 RBL 阶段，执行固化在 ROM 侧的代码，对 AWR2188 进行时钟等初始化，等待从外部处理器下载 DFP 镜像。

```

/*Power up*/
pwrup.c_PowerControl = M_T_ML_FE_DEVICE_POWER_UP; //上电宏
pwrup.c_SopBootCfg = M_ML_DEVICE_SOP_BOOT_SPI_MODE_0; //SPI 模式
pwrup.c_SopClkCfg =M_ML_DEVICE_SOP_CLOCK_50MHZ_OSC_OUT_DISABLE; //时钟配置

status = ml_frontEndPowerOnDevice(DEVICE_INDEX, &pwrup);

```

表 3-1. 外部处理器软件的 SOP 配置

配置	介绍
0x01	使用 SPI0 的功能模式
0x06	使用 SPI1 的功能模式
0x07	使用 I2C 的功能模式
0x05	使用 UART 的设备管理模式
0x03	Debug 模式

3.2 DFP 镜像下载

AWR2188 在 RBL 阶段会根据 SOP 判断是 QSPI 模式还是 SPI/I2C 模式，如果是 QSPI 模式，AWR2188 将 DFP 镜像从 Flash 复制到 AWR2188 的 RAM 中，如果是 SPI/I2C 模式，则需要外部处理器通过 SPI 或者 I2C 将 DFP 镜像传输给 AWR2188。下面以 SPI 模式为例，介绍 DFP 镜像下载。

```
/* Offset in the M0 flash where the metaimage for the 2188 is stored*/
imagecfg.w_ImageSrcAddress = 0x6F80;
/* Size of the metaimage*/
imagecfg.w_ImageSize = 0x32F48;
status = ml_rblImageDownload(DEVICE_INDEX,&imagecfg);
while(status!=0);
```

外部处理器会通过 ml_rblImageDownload 此函数将 DFP 镜像下载到 AWR2188 中，使用此函数下载镜像时需要注意的地方有两点：

1. 镜像在外部处理器中的地址：Flash 或者外部处理器侧 DFP 镜像的位置；
2. 镜像的大小。例如，镜像大小为 208712Bytes，示例代码中用十六进制来表示大小，即 0x32F48。

一般来说，被下载的固件是 TI 提供的 DFP 固件包，本地存放路径是："C:\ti\mmWave_DFP_04_00_xx_xx\firmware\WR2188\metaimage\mmwave_dfp_metaimage.bin"。

下载镜像完毕后，AWR2188 会返回一个校验值，保存在结构体 T_ML_MSS_APPLICATION_ENTRY_RSP.w_ParseStatus 中，用户可以根据这个返回值来判断镜像是否下载正常。如果镜像正常下载到 AWR2188，此校验值为 0x3E1A95C5。

由于镜像下载及后续控制均通过 SPI / I2C 协议完成，因此 AWR2188 的 SPI / I2C 协议解析放在第 4 章单独讨论。

3.3 MSS 应用程序 (Application) 启动与配置

AWR2188 的 MSS 上电是在外部处理器调用 frontEndPowerOnDevice 的过程中完成的。当镜像下载完毕后，AWR2188 会通过 ml_rblSwitchToApplication 切换到对应的应用程序启动模式。目前有两种应用程序启动模式可以配置，分别是 QSPI 模式和串行数据接口模式。这两者的区别在于 MSS 的应用程序通过 QSPI 接口中从 Flash 加载，还是外部处理器通过 SPI/I2C 协议进行加载。AWR2188 切换到对应的启动模式下之后会触发一个异步事件来通知外部处理器，可通过检查返回值 aehandler.h_MssAsyncStatus 是否为 1，来确认该异步事件是否正常触发。

```
/* Switch to Application*/
entryptr.h_BootMode = 0x00;
status = ml_rblSwitchToApplication(DEVICE_INDEX,&entryptr,&entryrsptr);
/* Ensure device power up*/
while((aehandler.h_MssAsyncStatus & 1) != 1);
```

表 3-2. AWR2188 应用程序 (Application) 启动模式配置

配置	介绍
0xAACE	从 QSPI 启动
0x0000	从 SPI/I2C 接口启动

切换到对应的应用程序启动模式后，AWR2188 将会检查 MSS 的上电状态并配置 MSS 的参数，MSS 配置完毕后 RSS 也会按照上电、检查上电状态和配置 RSS 的流程依次进行配置。其中，MSS 和 RSS 上电所需的关键配置见表 3-3、表 3-4 和表 3-5。

表 3-3. MSS 上电配置

名称	介绍	配置
c_PowerControl	将 nReset 设置为低电平来关闭电源	0
	将 nReset 设置为高电平来启动器件	1
c_SopBootCfg	见表 3-1	见表 3-1

表 3-3. MSS 上电配置 (续)

名称	介绍	配置
c_SopClkCfg	40MHz XTAL, OCS_CLOCKOUT enabled	0
	40MHz XTAL, OCS_CLOCKOUT disabled	1
	50MHz XTAL, OCS_CLOCKOUT disabled	2
	50MHz XTAL, OCS_CLOCKOUT enabled	3

表 3-4. RSS 上电配置

名称	介绍	配置
c_PowerOnMode	冷启动, RSS 会清除所有 RAM 中的所有数据, 固件变量也会被初始化为默认值。一般此模式适用于 RSS 首次上电或者 RSS 数据在断电后无法保存的情况	0x00
c_ClkSourceSel	RSS 的时钟来自外部晶源	0
	RSS 的时钟来自 PLL-Dig	1
	RSS 的时钟来自 Core-ADPLL clock	2
c_HostSubsystemSel	对于 AWR2188 这样的 MMIC, RSS 一次只能通信一个子系统	0
c_RefClkIndex	50 MHz XTAL	0
	40 MHz XTAL	1
	50 MHz BAW	2
h_CoreClockFreq	AWR2188 RSS 的 Core Clock 必须是 200MHz	200
w_RssPowerUpCfg	见表 3-5	见表 3-5

表 3-5. w_RssPowerUpCfg 上电配置

Bit	介绍
Bit[0]	Disable Boot Monitor; 1 为 Disable ; 0 为 Enable
Bit[1]	Disable RSS Logger; 1 为 Disable ; 0 为 Enable
Bit[2]	Disable Configuration Error Check; 1 为 Disable ; 0 为 Enable
Bit[3]	Disable RSS CRC; 1 为 Disable ; 0 为 Enable
Bit[4]	Disable watch dog; 1 为 Disable ; 0 为 Enable
Bit[5]	Disable RSS Core Clock Gating; 1 为 Disable ; 0 为 Enable
Bit[6]	Disable RSS Clock Scaling; 1 为 Disable ; 0 为 Enable
Bit[7]	Enable RSS Debug Log Async-Event; AWR2188 上此功能默认为 0
Bit[8]	Disable Chirp-RAM Memory Initialization; 1 为 Disable ; 0 为 Enable

```

status = ml_mssStaticConfig(DEVICE_INDEX,&mssstaticcfg); /*mss Static config*/
while(status!=0);
rsspwrupconfig.c_PowerOnMode = M_ML_RSS_POWER_UP_MODE_COLD;
rsspwrupconfig.c_ClkSourceSel = M_ML_RSS_CLK_SOURCE_DIG_PLL;
rsspwrupconfig.c_HostSubsystemSel = M_ML_RSS_SUB_SYSTEM_INDEX_DEFAULT;
rsspwrupconfig.c_RefClkIndex = M_ML_RSS_REF_CLK_INDEX_50MHZ_XTAL;
rsspwrupconfig.h_CoreClockFreq = 200; //200 MHz is frequency at which the RSS operates
rsspwrupconfig.w_RssPowerUpCfg = M_ML_RSS_PU_CFG_DISABLE_BOOT_MON | M_ML_RSS_PU_CFG_DISABLE_LOGGER |
M_ML_RSS_PU_CFG_ENABLE_ERROR_CHECK | M_ML_RSS_PU_CFG_DISABLE_CRC | M_ML_RSS_PU_CFG_DISABLE_WDT |
M_ML_RSS_PU_CFG_DISABLE_RSS_CLK_SCALING;
status = ml_rssPowerUp(DEVICE_INDEX,&rsspwrupconfig); /*RSS power up*/
    
```

在外部处理器侧用户只需要将 MSS/RSS 的参数设置好, 直接调用 *mssStaticConfig*、*rssPowerUp*、*rssStaticConfig* 将配置文件发送给 AWR2188 即可。表 3-3 和 表 3-4 列出了 MSS 和 RSS 必须配置的参数, 更详细的配置请参看 DFP 各个 API 接口说明文档。

4 AWR2188 SPI/I2C 配置与信号解析

AWR2188 无论是下载镜像、控制命令、命令校验，都离不开 SPI 或者 I2C 协议，理解这一部分对用户调试非常有帮助。

4.1 SPI 启动配置与时序

当 AWR2188 SOP[2:0]配置为 001 时，AWR2188 会通过 SPI 协议与外部处理器进行通信，SPI 模式下 AWR2188 使用的硬件管脚如 表 4-1。

表 4-1. SPI 管脚配置

信号名称	介绍	管脚序号
MSS_GPIO_0	被 SPI_BUSY 使用，用于 SPI (I2C) 和外部处理器的通信同步的控制	AF13
MSS_HOSTIRQ	被 HOST_IRQ 使用，用于触发异步事件	AF17
MSS_McSPIA_CLK	SPI 时钟信号	AG22
MSS_McSPIA_CS0	SPI 片选信号	AF20
MSS_McSPIA_MISO	AWR2188 SPI 输出数据线	AF22
MSS_McSPIA_MOSI	AWR2188 SPI 输入数据线	AF23

外部处理器通过 SPI 向 AWR2188 发送的信号会分为两部分：CMD_HEADER 和 CMD_LONG_MSG。其中 CMD_HEADER 的数据格式是固定的，主要的作用是告诉 AWR2188 CMD_LONG_MSG 的功能、数据大小、CRC 等信息，具体请参考 表 4-2。CMD_LONG_MSG 是此次 SPI 命令传输需要传递的数据或命令。除此之外，AWR2188 还可以使用 CMD_PATTERN (Pattern 信号，MISO 返回一个校验信号来通知 MOSI 成功通信) 功能，在外部处理器发送命令时，可以根据 AWR2188 的返回值来确定 SPI 命令是否正常下达。命令发送和命令响应的流程类似。当 AWR2188 向外部处理器发送 RESP_HEADER，RESP_LONG_MSG，外部处理器回应 RESP_PATTERN。

表 4-2. SPI 帧数据格式

信号名称	Bytes
HDR_CHKSUM (Header Checksum)	2
SPI_CMD_ID (SPI Command ID)	1
CMD_FLAGS (Command Flags)	1
RW_TMT (Read/Write Timeout)	1
AUX_INFO (Auxiliary Information)	1
MSG_SIZE (Message Size)	2
MSG_CRC (Message CRC)	4
SHORT_MSG (Short Message)	4
LONG_MSG (Long Message)	N

表 4-2 需要注意的是 SPI_CMD_ID，可以通过 SPI_CMD_ID 直接读取到此命令代表含义，各个 CMD 的含义如 表 4-3 所示。

表 4-3. SPI_CMD_ID

接口函数名称	SPI_CMD_ID
SPI error info GET command	0x00
SPI error info GET response	0x80
MSS control SET/GET SPI command	0x01
MSS control SET/GET SPI response	0x81
MSS memory write SPI command	0x02
MSS memory write SPI response	0x82
MSS handling RSS control GET SPI command	0x04
MSS handling RSS control GET SPI response	0x84

AWR2188 的 SPI 每次会收发两个字节，片选信号 (SPI_CS) 可在每 1 个字节或每 2 个字节后拉高一次。需要注意的是在上图的示例中的外部处理器：M0 是小端设备，但是 M0 的 SPI 模块是大端设备，因此每次发送数据时都需要将数据进行字节交换。使用其他大端或者小端的外部处理器，只需要保证 SPI 上是大端格式的数据即可。

外部处理器的 SPI 每次发送两个字节，SPI_CMD_ID 和 CMD_FLAG 同时发送并交换字节序，所以第 3 个字节的 SPI_CMD_ID 在 SPI MOSI 上表现出来的是第 4 个字节。每次 CMD_HEADER 发送完毕后，外部处理器会接收到来自 AWR2188 的 CMD_PATTERN，此校验信号由固定格式的 16 个字节组成：

```
const UINT8 c_CmdPattern[M_RL_PKT_HDR_SIZE] =
{
    0xCDU, 0xABU, 0xBAU, 0xDCU, 0xCDU, 0xABU, 0xBAU, 0xDCU,
    0xCDU, 0xABU, 0xBAU, 0xDCU, 0xCDU, 0xABU, 0xBAU, 0xDCU
};
```

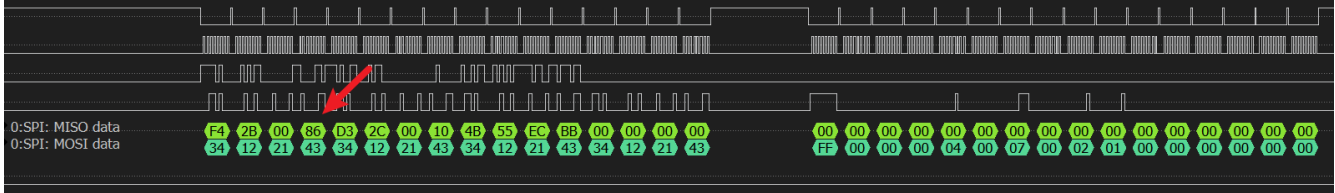


图 4-3. AWR2188 下载 DFP 固件的 CMD_HEADER

同理，按照图 4-3 相同的 SPI 命令格式可以解析出 AWR2188 在 MISO 上发送给外部处理器的响应命令，根据表 4-2 和 AWR2188 的 RESP_HEADER (MISO 线) 的第 4 个字节 0x86 可判断出镜像已下载完毕，此时外部处理器发送给 AWR2188 RESP_PATTERN：

```
const UINT8 c_RespPattern[M_RL_PKT_HDR_SIZE] =
{
    0x34U, 0x12U, 0x21U, 0x43U, 0x34U, 0x12U, 0x21U, 0x43U,
    0x34U, 0x12U, 0x21U, 0x43U, 0x34U, 0x12U, 0x21U, 0x43U
};
```

总之，AWR2188 是通过 SPI 协议被外部处理器控制，因此所有的控制命令均可以使用逻辑分析仪结合表 4-2 解析出来，从而判断 AWR2188 当前的状态。

```
const UINT8 c_RespPattern[M_RL_PKT_HDR_SIZE] =
{
    0x34U, 0x12U, 0x21U, 0x43U, 0x34U, 0x12U, 0x21U, 0x43U,
    0x34U, 0x12U, 0x21U, 0x43U, 0x34U, 0x12U, 0x21U, 0x43U
};
```

4.2 I2C 启动配置与时序

当 AWR2188 SOP[2:0]配置为 111 时，AWR2188 会通过 I2C 协议与外部处理器进行通信。AWR2188 上使用的引脚如表 4-4。在 I2C 协议中，SPI_BUSY 和 HOST_IRQ 也都是必需的，用于同步 I2C 信号和触发异步事件。

表 4-4. I2C 硬件管脚

SIGNAL_NAME	Description	APE PIN
MSS_GPIO_0	被 SPI_BUSY 使用，I2C/SPI 的控制信号	AF13
MSS_HOSTIRQ	被 HOST_IRQ 使用，用于触发异步事件	AF17
MSS_I2CA_SCL	I2C 的时钟信号	AF22
MSS_I2CA_SDA	I2C 的数据信号	AF23
MSS_GPIO_2	在 SOP_I2C 中被 I2C_ADDR_0 使用	AF6
MSS_GPIO_4	在 SOP_I2C 中被 I2C_ADDR_1 使用	AF20
MSS_GPIO_5	在 SOP_I2C 中被 I2C_ADDR_2 使用	AG22

AWR2188 的 I2C 地址按照下面的格式配置，I2C 地址的最后三位可由用户根据需要自定义，从 0101000 到 0101111 均可。

AWR2188 I2C ADDRESS						
0	1	0	1	MSS_GPIO_5	MSS_GPIO_4	MSS_GPIO_2

整个 I2C 协议与 SPI 协议是一致的，SPI 与 I2C 使用完全相同的底层驱动，因此 AWR2188 的 I2C 协议可以参考 SPI 协议进行分析。但是由于 AWR2188 只有一根数据线 (I2C_SDA)，因此 I2C 模式下会关闭校验信号，外部处理器不会收到 AWR2188 返回的校验信号。

5 参考文献

- [AWR2188 Single Chip 8x8 Cascadable 76-to-81 GHz Transceiver in LOP package datasheet](#)
- [AWR2243 引导加载程序流程](#)
- [AWR2188 EVM 硬件参考设计](#)
- [AWR2188 卫星雷达参考设计](#)

6 修改记录

Version	Date	Author	Notes
0.1	Oct 24 th 2025	Roy Zhuang / Allen Yin	First draft
0.2	Mar 15 th 2026	Roy Zhuang / Allen Yin	Second draft
*	June 12 th 2026	Roy Zhuang / Allen Yin	initial release

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、与某特定用途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安全、安保法规或其他要求。

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。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 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。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月