

UniFlash CC3x20 和 CC3x3x SimpleLink™ Wi-Fi® 及 Internet-on-a-chip™ 解决方案 ImageCreator 和编程工具



摘要

CC3x20 和 CC3x3x 器件是 SimpleLink™ 微控制器 (MCU) 平台的一部分，该平台包括 Wi-Fi®、低功耗蓝牙®、低于 1Ghz 和主机 MCU，它们共用一个易于使用的通用型开发环境，其中包含单核软件开发套件 (SDK) 和丰富的工具集。借助一次性集成的 SimpleLink™ 平台，您可以将产品系列中的任何器件组合添加到自己的设计中，从而在设计要求变更时实现 100% 的代码重用。如需了解更多相关信息，请访问 www.ti.com.cn/simplelink/cn。

本用户指南介绍了德州仪器 (TI)™ 的 UniFlash CC3x20 和 CC3x3x SimpleLink™ Wi-Fi® 以及 Internet-on-a chip™ 解决方案 ImageCreator 和编程工具。

内容

1 引言.....	4
2 术语和概念.....	5
3 安装.....	6
4 ImageCreator 应用.....	7
5 快速入门.....	8
5.1 创建一个新工程.....	8
5.2 简单模式.....	9
5.3 添加 MCU 镜像.....	9
5.4 添加服务包.....	9
5.5 从打开的工程创建和编程镜像.....	9
6 使用.....	11
6.1 创建一个新工程.....	11
6.2 打开一个近期工程.....	11
6.3 管理工程.....	12
6.4 器件状态和设置.....	13
6.5 简单模式.....	15
6.6 高级模式.....	16
6.7 高级模式 - 常规设置.....	17
6.8 高级模式 - 系统设置.....	18
6.9 添加服务包.....	28
6.10 添加受信任的根证书目录.....	28
6.11 添加主机应用文件 (CC32xx).....	29
6.12 用户文件.....	30
6.13 器件文件浏览器.....	35
6.14 从一个工程中创建镜像.....	36
6.15 创建 OTA.....	36
6.16 保存一个镜像.....	37
6.17 Programming.bin 和 Programming.hex.....	37
6.18 从一个已打开的工程进行镜像编程.....	37
6.19 使用 .sli 文件对一个镜像进行编程.....	37
6.20 带密钥的安全镜像.....	39
7 命令行.....	40
7.1 工程命令.....	40
7.2 镜像命令.....	48

7.3 工具命令.....	49
7.4 器件命令.....	51
7.5 GUI 配置命令.....	51
7.6 GUI 命令附加参数.....	52
8 工具.....	53
8.1 证书签名请求 (CSR) (仅适用于 CC323xS/SF 器件)	53
8.2 签署文件.....	53
8.3 激活镜像.....	54
9 使用 CSR 实用程序.....	55
9.1 从器件获取 CSR 并复制到文件	55
9.2 替换工程中的 CSR 文件.....	57
10 天线 LaunchPad 的默认功率值.....	59
10.1 CC3x35 器件的默认值.....	59
10.2 CC3x35MOD 器件的默认值.....	61
11 修订历史记录.....	62

插图清单

图 1-1. 使用 ImageCreator 进行编程.....	4
图 4-1. 使用 UniFlash 打开 ImageCreator (1, 共 2).....	7
图 4-2. 使用 UniFlash 打开 ImageCreator (2).....	7
图 5-1. 新建工程.....	8
图 5-2. 器件类型.....	8
图 5-3. 简单模式 CC32xx.....	9
图 6-1. 打开近期工程.....	11
图 6-2. 工程管理.....	12
图 6-3. 器件状态：已断开连接.....	13
图 6-4. 器件状态：已连接.....	14
图 6-5. 简单模式 CC31xx.....	15
图 6-6. 简单模式 CC32xx.....	16
图 6-7. 器件状态高级模式.....	17
图 6-8. 工具提示.....	17
图 6-9. 16 字节密钥示例.....	18
图 6-10. 设置密钥文件名.....	18
图 6-11. PHY (2.4G) 校准模式.....	19
图 6-12. 监管域表 2.4G (1, 共 2).....	19
图 6-13. 监管域表 2.4G (2, 共 2).....	20
图 6-14. 监管域表.....	21
图 6-15. 共存和天线选择.....	23
图 6-16. 器件身份配置.....	24
图 6-17. 证书配置.....	24
图 6-18. 证书签名请求选项.....	25
图 6-19. 自签名证书选项.....	26
图 6-20. HTTP 服务器.....	28
图 6-21. 供应商证书目录.....	29
图 6-22. OTP 章节.....	29
图 6-23. MCU 镜像高级模式.....	29
图 6-24. “文件属性”对话框.....	31
图 6-25. 重命名文件.....	32
图 6-26. 删除文件或文件夹 (1, 共 2)	33
图 6-27. 删除文件或文件夹 (2, 共 2)	33
图 6-28. 用户文件操作监视器.....	34
图 6-29. 删除文件.....	35
图 6-30. 获取文件.....	35
图 6-31. OTA 私钥文件名.....	36
图 6-32. 保存镜像.....	37
图 6-33. 编程镜像.....	37
图 6-34. 编程镜像.....	38
图 6-35. 编程镜像.....	39

图 8-1. 打开工具.....	53
图 8-2. 证书签名请求.....	53
图 8-3. 签署文件.....	54
图 8-4. 激活镜像.....	54

表格清单

表 2-1. 术语和概念.....	5
表 6-1. TX 参数表.....	22
表 6-2. 标志选项.....	32
表 6-3. 其他文件属性.....	32
表 10-1. 2.4GHz 默认值.....	59
表 10-2. 5GHz 默认值.....	60
表 10-3. 2.4GHz 默认值.....	61
表 10-4. 5GHz 默认值.....	62

商标

SimpleLink™, 德州仪器 (TI)™, and Internet-on-a chip™ are trademarks of Texas Instruments.

Wi-Fi® are registered trademarks of Wi-Fi Alliance.

蓝牙® is a registered trademark of Bluetooth SIG.

所有商标均为其各自所有者的财产。

1 引言

ImageCreator 是 UniFlash 应用程序的一部分，用于创建编程镜像。ImageCreator 还可将编程镜像写入 SimpleLink™ CC3xx 器件。编程镜像是一个文件，包含 SimpleLink 器件配置以及运行此器件所需的文件。对于 SimpleLink™ CC32xx 无线微控制器 (MCU)，编程镜像还包含主机应用文件。

对于新的 SimpleLink™ 器件，应首先使用编程镜像对其进行编程。可将由 ImageCreator 创建的镜像作为生产程序的一部分或在开发阶段编程到器件中。镜像编程方法如下：

- 通过 UART 接口使用 ImageCreator 工具
- 通过串行闪存 SPI 接口使用一个现成的外部工具

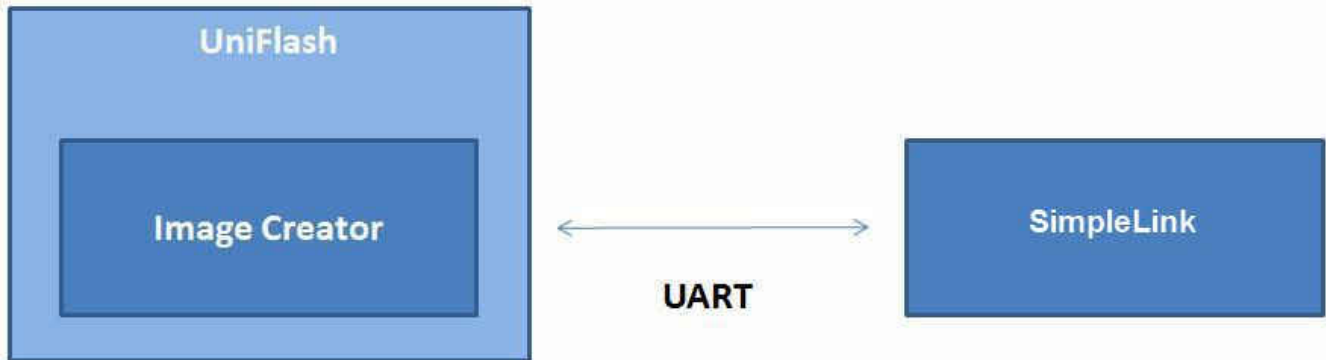


图 1-1. 使用 ImageCreator 进行编程

ImageCreator 的主要特性为：

- 支持 SimpleLink™ CC32xx 和 CC31xx 器件
- 可在生产模式或开发模式下创建镜像。生产图像会限制专用于开发的某些特性，例如 JTAG 接口或使用此工具访问单个文件。
- 创建加密编程图像
- 作为编程镜像创建的一部分，可实现以下功能：
 - 应用服务包和证书存储库
 - 定义器件配置，例如 Wi-Fi® 模式、IP 设置和配置等
 - 添加文件并应用每个文件的属性，例如安全设置和失效防护
- 连接到器件并检索其属性，例如器件类型、闪存大小和 MAC 地址
- 在开发过程中，此图像支持在线访问器件文件系统
- 可对器件进行编程，并可使用由 ImageCreator 的另一实例所创建的图像进行编程
- 使用命令行接口执行一些操作
- 工程管理能力：导入现有工程或将工程导出到其他机器

本文档介绍如何安装、操作和使用 UniFlash 中的 SimpleLink™ ImageCreator 工具。

2 术语和概念

表 2-1 列出了本文档中使用的术语和概念。

表 2-1. 术语和概念

术语或概念	说明
镜像	镜像是一个打包文件，其中包含服务包、系统配置文件、用户文件和主机程序（适用于 SimpleLink Wi-Fi CC32xx 无线 MCU）。编程镜像的创建是一个脱机进行的过程。
工程	工程是用于创建镜像文件的工作区。
连接	用户可连接到器件并获取其属性，例如器件 MAC 地址、安全类型等。
密钥	16 字节密钥用于镜像加密。
编程镜像文件类型	镜像文件以多种编码类型创建： <ul style="list-style-type: none"> • programming.bin（标准二进制）和 programming.hex（Intel 十六进制）文件用于通过外部串行闪存编程工具进行编程。 • Programming.ucf（TI 专有编码）用于主机编程。 • Programming.sli（TI 专有编码）用于 ImageCreator 编程。 • 注 <ul style="list-style-type: none"> - 加密镜像命名为 programming.encrypt.bin/hex/ucf/sli - 输出文件位于 <code>projects\\${project_name}\sl_image\Output</code> 路径下的 ImageCreator 安装目录中

3 安装

ImageCreator 是 UniFlash 应用的一部分。下载并运行从 <http://www.ti.com.cn/tool/cn/UNIFLASH> 获取的最新版 UniFlash 应用安装程序。

4 ImageCreator 应用

运行 UniFlash 应用。显示受支持的所有器件列表。从器件列表中选择 **CC31xx** 或 **CC32xx**，如图 4-1 所示。

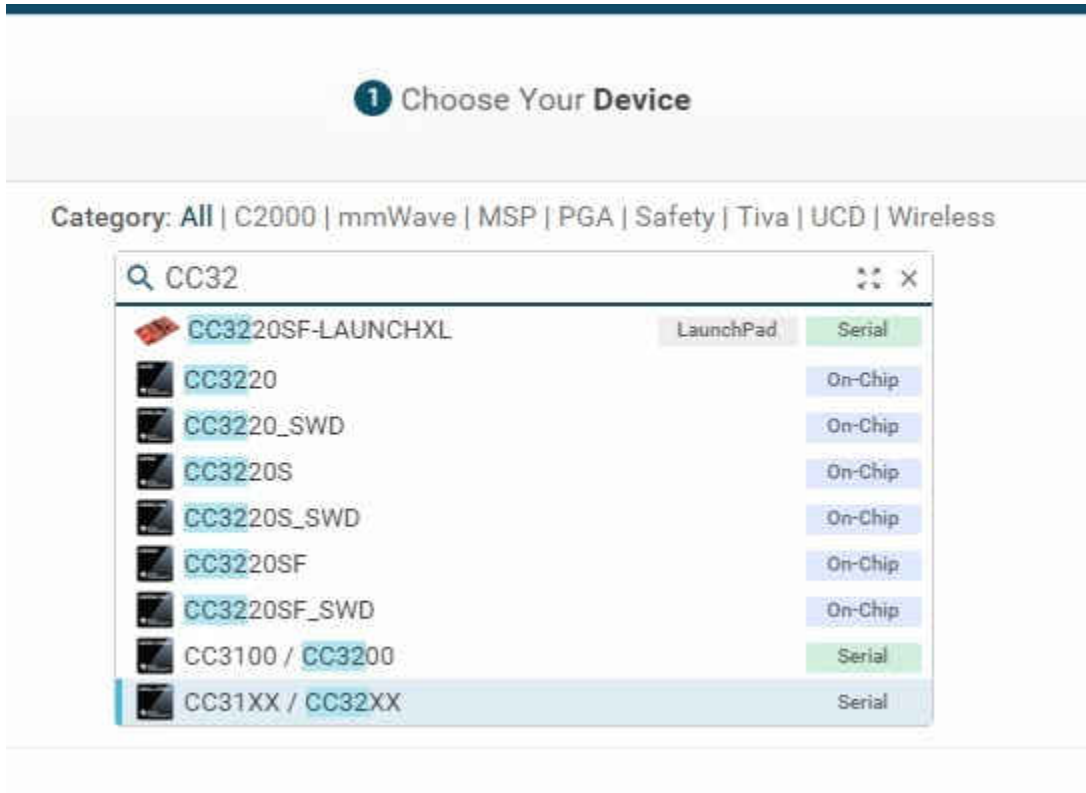


图 4-1. 使用 UniFlash 打开 ImageCreator (1, 共 2)

然后点击 “Start Image Creator” 按钮，如图 4-2 所示。



图 4-2. 使用 UniFlash 打开 ImageCreator (2)

5 快速入门

5.1 创建一个新工程

点击“Welcome”页面上的“New Project”按钮，如图 5-1 所示。

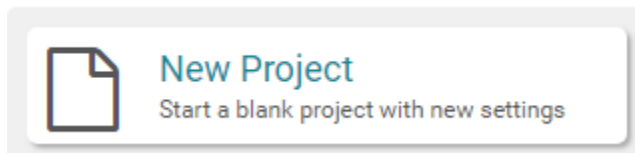


图 5-1. 新建工程

会显示“Create Project”窗口。

- Project name - 唯一工程名 (必填)
- Project Description - 简要描述工程 (可选)
- Device Type - 器件类型，如图 5-2 所示。

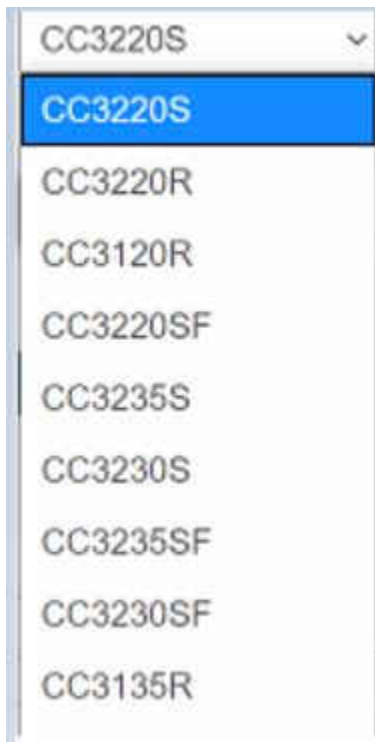


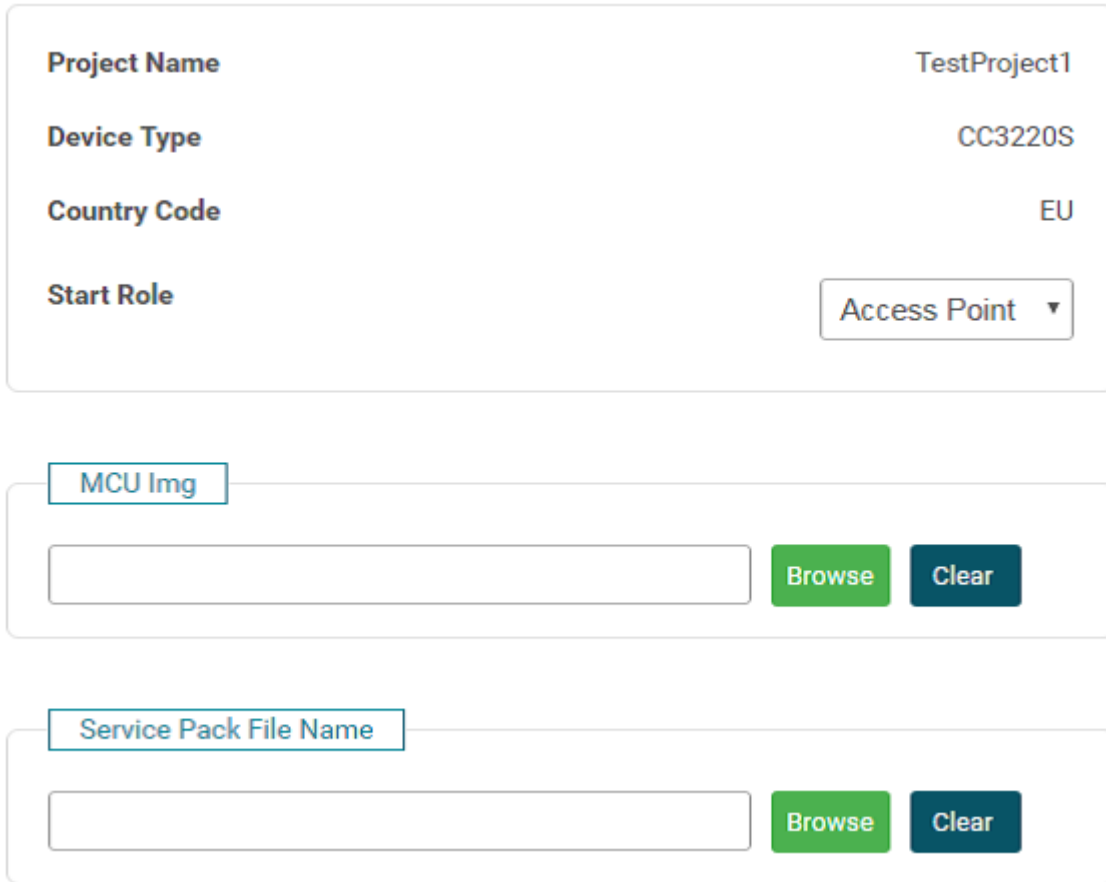
图 5-2. 器件类型

- Device mode - (生产/开发)
 - 默认为生产模式。在该模式下，用户无法使用 IDE 进行调试。
 - 开发模式用于 JTAG 接口和对单个文件的访问。此工程创建的镜像通过 MAC 地址用于特定器件。

用户应填写相关字段，并点击“Create Project”按钮。带有默认参数的一个新工程创建成功。

5.2 简单模式

ImageCreator 有两种模式：简单模式和高级模式。工程加载或创建成功后，ImageCreator 以简单模式打开工程（请参阅图 5-3）。



The screenshot displays the configuration page for a CC32xx device in simple mode. It includes the following elements:

- Project Name:** TestProject1
- Device Type:** CC3220S
- Country Code:** EU
- Start Role:** Access Point (with a dropdown arrow)
- MCU Img:** A text input field, a green 'Browse' button, and a dark blue 'Clear' button.
- Service Pack File Name:** A text input field, a green 'Browse' button, and a dark blue 'Clear' button.

图 5-3. 简单模式 CC32xx

5.3 添加 MCU 镜像

借助 SimpleLink ImageCreator，用户可为 CC32xx 器件添加主机应用文件（MCU 镜像）。在简单模式下，如果对 CC32xx 器件进行安全保护，那么 MCU 镜像上传时会自动由虚拟根证书签名。

添加完成后，显示如下文件名：

- CC32xxR/RS：mcuimg.bin
- CC32xxSF：mcuflashimg.bin

5.4 添加服务包

此服务包用于升级网络外围设备内部固件。服务包文件由 TI 提供，包含在 SDK 软件包中。服务包文件名为 sp_<release_versions_number>.bin，存放在 <SDK_PATH>\tools\cc32xx_tools\servicepack-cc3xXX 文件夹中。

TI 建议将服务包添加到编程镜像中，但这并非强制性要求。如果尚未对服务包进行编程，器件使用其出厂代码。

添加服务包时，用户选择文件存放位置。但是，ImageCreator 不保留原始文件的链接。若要修改服务包，应重新选择一个新的服务包文件。

5.5 从打开的工程创建和编程镜像

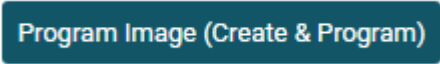
在生产模式下，编程前无需连接器件。

在开发模式下，应在创建镜像之前提供器件的 MAC 地址，并在编程前点击

 Connect

按钮以连接该器件。

若要对镜像进行编程，请点击

 Program Image (Create & Program)

按钮：编程镜像也会创建镜像，因此无需在编程之前创建镜像。

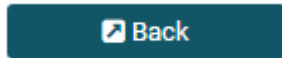
6 使用

6.1 创建一个新工程

请参阅节 5.1。

6.2 打开一个近期工程

直接从主“Welcome...”页面点击近期工程列表中的工程名，即可打开一个现有工程。可通过点击以下按钮导航到主页：



(请参阅图 6-1) 。

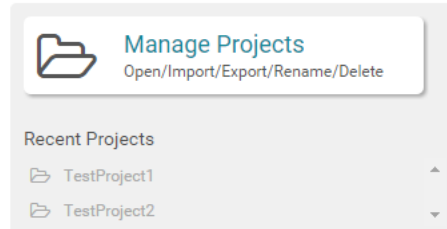


图 6-1. 打开近期工程

6.3 管理工程

点击“Manage Projects”按钮打开包含所有工程的列表，如图 6-1 所示。此时，屏幕会如图 6-2 所示。

Project Management

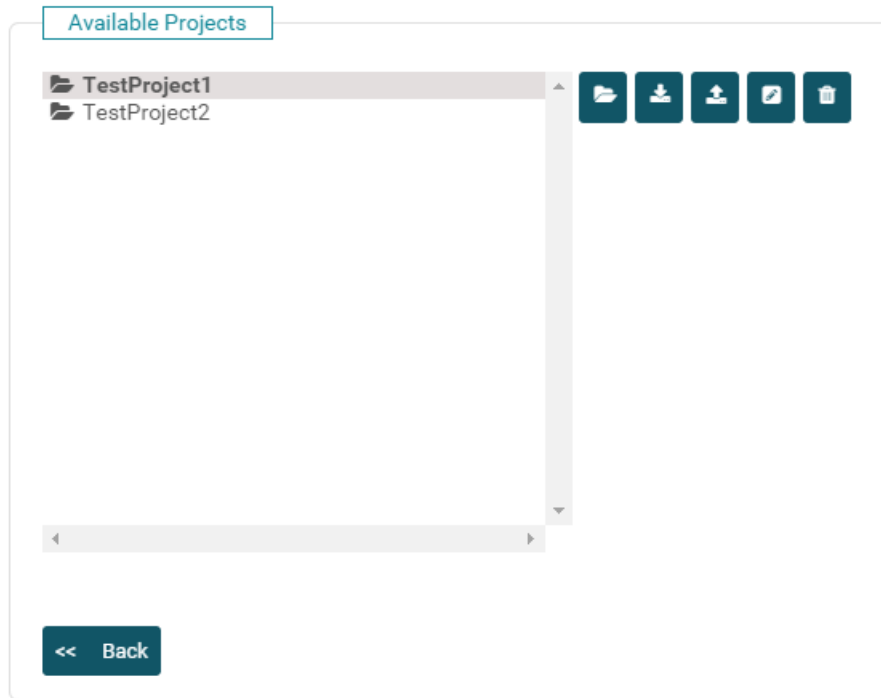


图 6-2. 工程管理

可用操作如下：

- 进入工程 - 按  进入工程。
- 从 ZIP 文件包导入工程 - 按  压缩文件。
- 导出工程 - 按  导出工程。
- 重命名工程 - 按  对工程重命名。
- 删除工程 - 按  删除工程。

6.4 器件状态和设置

可以在脱机状态下准备和创建编程镜像，例如，当器件未与 ImageCreator 应用计算机进行物理连接时（请参阅图 6-3）。

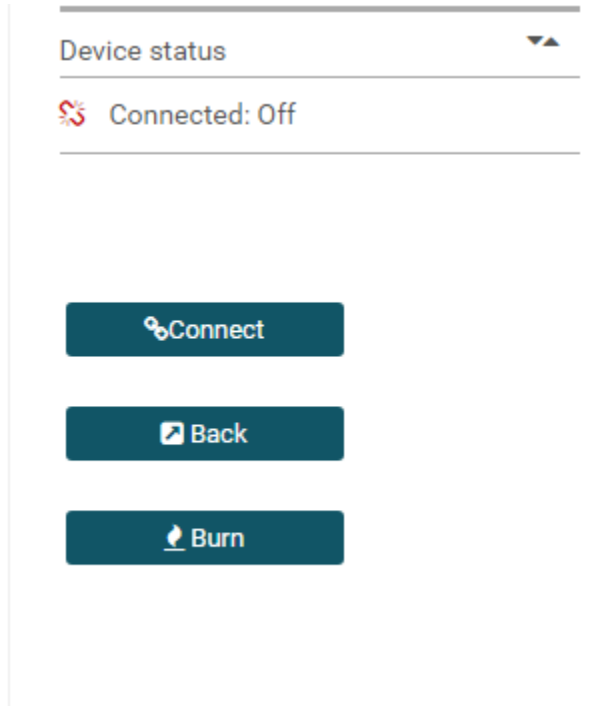


图 6-3. 器件状态：已断开连接

如果器件通过 UART 进行物理连接，用户可点击“Connect”按钮自动检测器件并进行初始连接。连接方法从器件检索设置并显示，如图 6-4 所示。

用户可选择生产模式或开发模式：

- 生产模式：创建的镜像在任意器件上均可编程。
 - 为确保器件安全，生产模式仅开放有限的操作功能：
 - 禁止使用 ImageCreator 对文件系统进行联机操作。
 - 禁止使用 JTAG（此规则对 CC32xx 器件适用）。
- 开发模式：需要目标器件的 MAC 地址才能对其进行编程。通过 ImageCreator 设置窗口设置目标器件的 MAC 地址。这种模式允许：
 - 浏览并修改器件文件系统（请参阅节 6.13）。
 - 允许使用 JTAG（此规则适用于 CC32xx 器件）。
 - 编程镜像文件只能用于对与镜像中设置的 MAC 地址相同的器件进行编程。

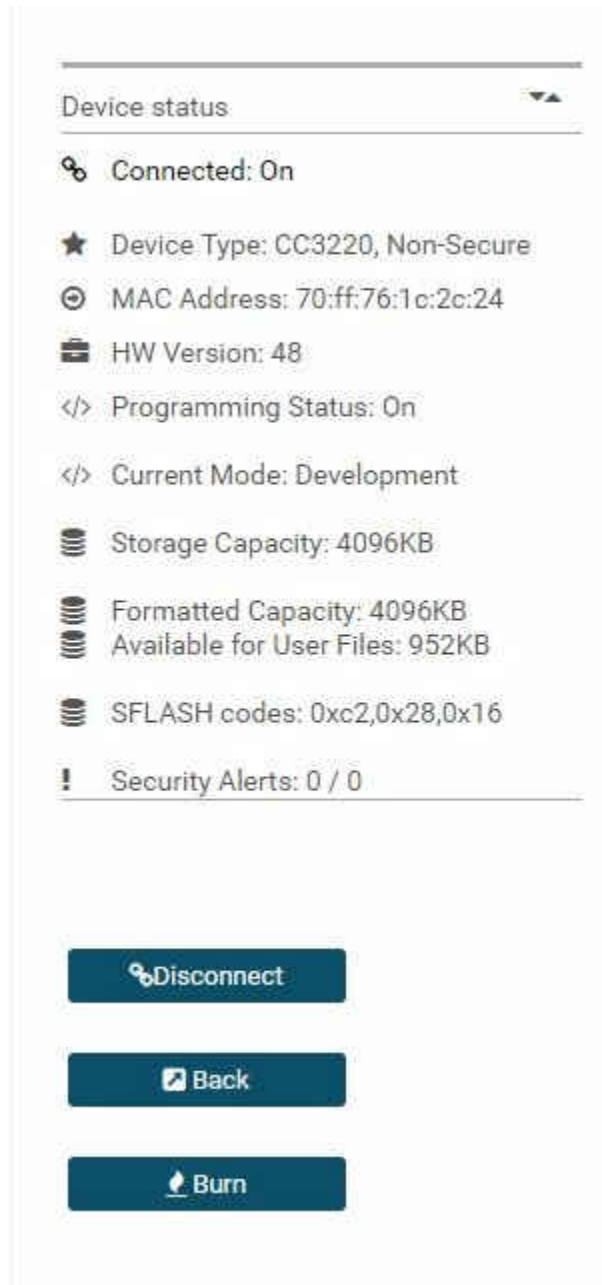
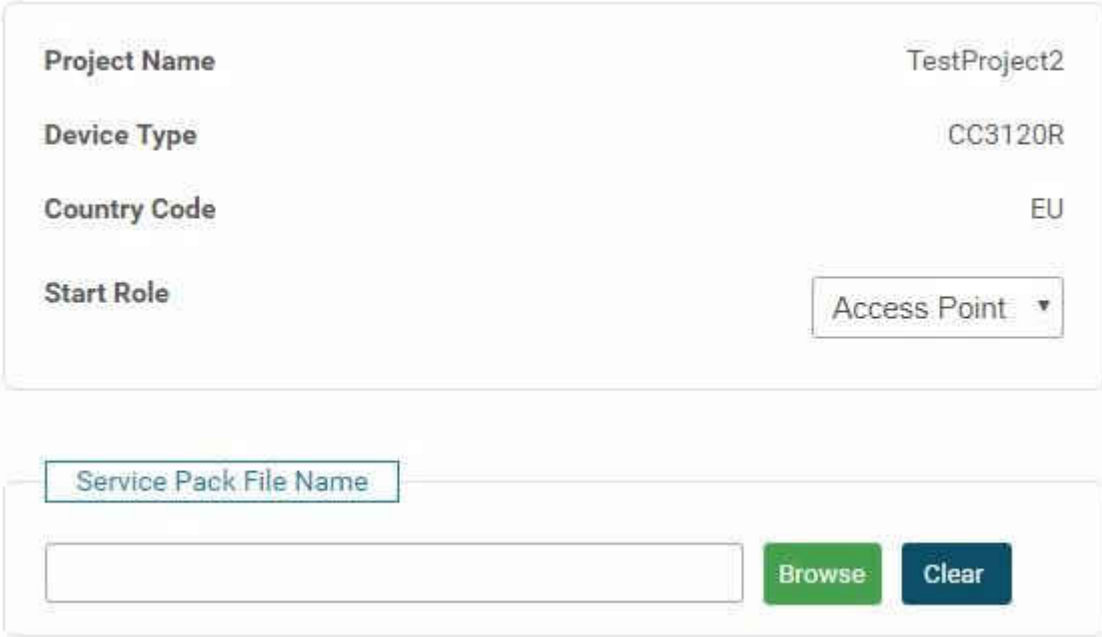


图 6-4. 器件状态：已连接

6.5 简单模式

ImageCreator 提供两种显示模式：简单模式和高级模式。工程加载或创建成功后，ImageCreator 以简单模式打开工程（请参阅图 6-5）。



The screenshot displays the configuration interface for the simple mode of ImageCreator. It features a form with the following fields and controls:

- Project Name:** TestProject2
- Device Type:** CC3120R
- Country Code:** EU
- Start Role:** Access Point (dropdown menu)
- Service Pack File Name:** A text input field with a "Browse" button and a "Clear" button.

图 6-5. 简单模式 CC31xx

简单模式可简化必要的工程参数配置。

在简单模式下，如果对 CC32xx 器件进行安全保护，那么 MCU 镜像上传时会自动由虚拟根证书签名。

Project Name	TestProject1
Device Type	CC3220S
Country Code	EU
Start Role	Access Point ▾

MCU Img

Service Pack File Name

图 6-6. 简单模式 CC32xx

6.6 高级模式

高级模式允许用户对工程参数进行更广泛的修改和调整。切换到高级模式后，不同于简单模式（参阅图 6-5 或图 6-6），右侧栏会出现更多选项（参阅图 6-7）。

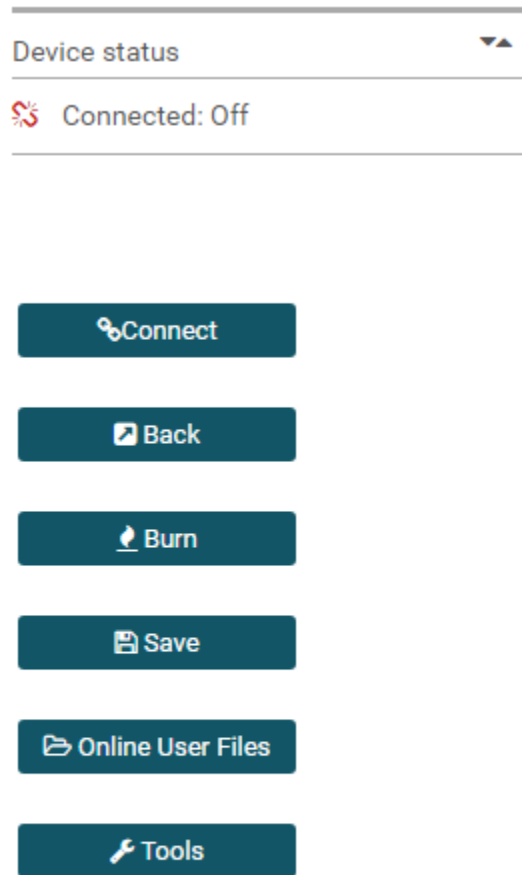


图 6-7. 器件状态高级模式

高级模式下，屏幕左侧以树形结构显示与配置页面的链接。点击该树形结构，即可快速导航至任意配置页面。页面中的字段包含工具提示，当鼠标移到问号图标上时会显示解释字段，如图 6-8 所示。

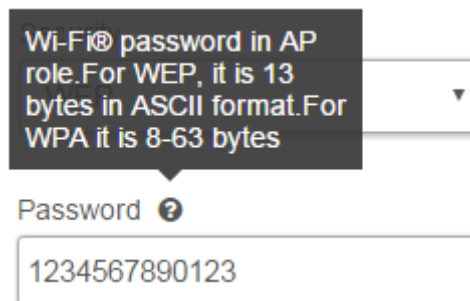


图 6-8. 工具提示

6.7 高级模式 - 常规设置

用户可对工程进行如下常规设置：

- 更改镜像模式（生产/开发）。请参阅节 6.4。
- 设置容量以及其他默认值。

- 创建加密镜像。请参阅节 6.7.1 和节 6.20。

6.7.1 创建一个加密镜像

ImageCreator 允许用户通过 AES-CTR 加密方式创建加密镜像。加密镜像只能与其密钥一起使用。

若要创建加密镜像，请创建一个包含 16 字节密钥的二进制文件，如图 6-9 所示。

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00000000:	11	22	33	44	55	66	77	88	99	00	11	22	33	44	55	66	."3DUfw♦♦.."3DUf

图 6-9. 16 字节密钥示例

然后，设置密钥文件名，如图 6-10 所示。

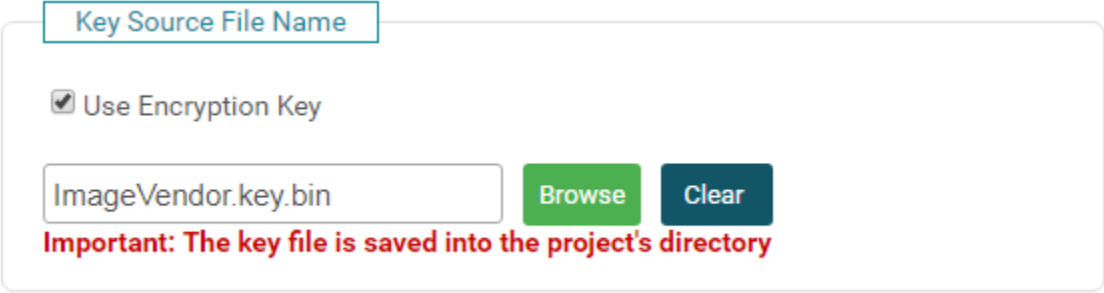


图 6-10. 设置密钥文件名

接下来，请参阅节 6.20。

6.8 高级模式 - 系统设置

本节介绍了配置系统设置的选项。

6.8.1 器件

6.8.1.1 无线设置

- PHY (2.4G) 校准模式 (请参阅节 6.8.1.1.1.1) 。
- 支持 5G 器件的配置选项：
 - 2.4G
 - TX 功率控制 (请参阅节 6.8.1.1.1.1)
 - 5G
 - PHY (5G) 校准模式 (请参阅节 6.8.1.1.2.1)
 - TX 功率控制 (请参阅节 6.8.1.1.2.2)
- 共存 (请参阅节 6.8.1.1.3)
- 天线选择 (请参阅节 6.8.1.1.3)

6.8.1.1.1 射频 2.4G

6.8.1.1.1.1 PHY (2.4G) 校准模式

若要修改校准，请选择如下校准模式之一 (如图 6-11 所示)：

- Normal
- Trigger
- Onetime

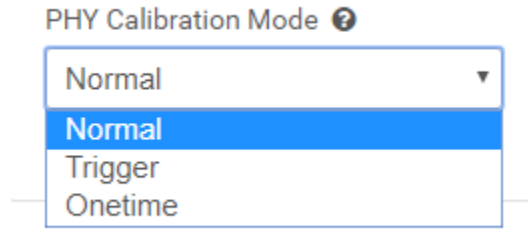


图 6-11. PHY (2.4G) 校准模式

对于低功耗应用，TI 建议选择“Trigger”模式而非“Onetime”模式，除非电流峰值限制构成绝对的约束。除非绝对必要或手动触发，否则“Trigger”模式不会触发校准。

“Normal”校准模式用于实现出色的射频性能，或在器件环境容易发生变化（温度变化）之时使用。

6.8.1.1.1.1 TX 功率控制 (2.4G)

为了修改输出功率，该工具允许用户按信道为指定的监管区域配置 2.4-GHz 频带传输功率级别。这有利于构建定制板（以及不同的 BOM）或频射跟踪损耗（这与 TI 参考设计中提供的信息不同，但 TI 强烈建议参考 TI 参考设计说明）。

若要打开监管域表（如图 6-12 和 图 6-13 所示），请点击射频 2.4G 部分中的“Configure”按钮。

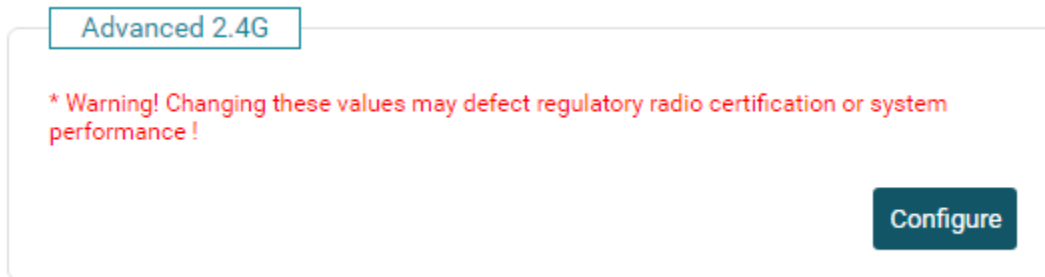


图 6-12. 监管域表 2.4G (1, 共 2)

Advanced RF 2.4G Settings

Regulatory Domain.

	<input type="checkbox"/> FCC BO Offset [dB]	<input type="checkbox"/> ETSI BO Offset [dB]	<input checked="" type="checkbox"/> JP BO Offset [dB]
Channel 1	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 2	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 3	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 4	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 5	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 6	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 7	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 8	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 9	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 10	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 11	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 12	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>
Channel 13	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>	11b <input type="text"/> L <input type="text"/> H <input type="text"/>

<< Back
Done

图 6-13. 监管域表 2.4G (2, 共 2)

回退偏移量 (BO) 决定了以 dB 为单位的配置值是来自默认 TI 设计的功率偏移，仅限于 EVM 和掩码约束。偏移量可以为正也可以为负，以使功率增加。若要更改特定的监管域回退偏移量，请选中相关复选框并根据所需的信道和速率组更改偏移量。

- 11b：仅 11b 速率
- H：高速率 (MCS7、54Mbps 和 48Mbps)
- L：低速率 (其余所有)

有效值为 -6[dB] 到 +6[dB]。

6.8.1.1.2 射频 5G

6.8.1.1.2.1 PHY (5G) 校准模式

仅支持 Normal 选项。

6.8.1.1.2.2 TX 功率控制

为了修改输出功率，ImageCreator 允许用户按信道为指定的监管区域配置 5GHz 频带传输功率级别。这有利于构建定制板 (以及不同的 BOM) 或频射跟踪损耗 (这与 TI 参考设计中提供的信息不同，但 TI 强烈建议参考 TI 参考设计说明)。

若要打开监管域表 (如图 6-14 所示以及表 6-1 所述)，请点击 5G 射频部分中的“Configure”按钮。

Advanced RF 5G Settings

Regulatory Domain.

	<input type="checkbox"/> FCC [dBm]	<input type="checkbox"/> ETSI [dBm]	<input type="checkbox"/> JP [dBm]	<input type="checkbox"/> Extra BO [dB]	<input type="checkbox"/> Ins.Loss [dB]
Channel 36	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	U-NII-1
Channel 40	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	TX <input type="text" value="0"/>
Channel 44	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	RX <input type="text" value="0"/>
Channel 48	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	AntG <input type="text" value="0"/>
Channel 52	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	U-NII-2A
Channel 56	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	TX <input type="text" value="0"/>
Channel 60	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	RX <input type="text" value="0"/>
Channel 64	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	AntG <input type="text" value="0"/>
Channel 100	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	U-NII-2C1
Channel 104	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	TX <input type="text" value="0"/>
Channel 108	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	RX <input type="text" value="0"/>
Channel 112	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	AntG <input type="text" value="0"/>
Channel 116	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	
Channel 120	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	U-NII-2C2
Channel 124	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	TX <input type="text" value="0"/>
Channel 128	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	RX <input type="text" value="0"/>
Channel 132	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	AntG <input type="text" value="0"/>
Channel 136	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	
Channel 140	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	
Channel 144	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	
Channel 149	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	U-NII-3
Channel 153	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	TX <input type="text" value="0"/>
Channel 157	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	RX <input type="text" value="0"/>
Channel 161	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	AntG <input type="text" value="0"/>
Channel 165	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	H <input type="text" value="0"/> L <input type="text" value="0"/>	

<< Back
Done

图 6-14. 监管域表

表 6-1. TX 参数表

列名	说明
FCC	在受 FCC 监管的国家/地区，在电路板走线损耗和天线增益之后，设置天线级别的输出功率上限（以 dBm 为单位）。
ETSI	在受 ETSI 监管的国家/地区，在电路板走线损耗和天线增益之后，设置天线级别的输出功率上限（以 dBm 为单位）。
JP	在受 JP 监管的国家/地区，在电路板走线损耗和天线增益之后，设置天线级别的输出功率上限（以 dBm 为单位）。
Extra BO	仅供 TI 使用。它采用额外的功率 BO，以牺牲输出功率的方式来提高 EVM 和掩码符合性。
o	H：高速率（MCS7、54Mbps 和 48Mbps）
o	L：低速率（其余所有）
Ins.Loss	设置电路板插入损耗（以 dB 为单位）。

参数修改示例：以下示例说明了如何针对某个监管域更改某个信道的 TX 输出功率级别。若要将信道 36 的限值设置为 12[dBm]，在天线增益之后，对 FCC 监管域进行如下操作：

1. 点击 FCC 复选框以进行编辑。
2. 填写除信道 36 以外的其它所有信道的默认值，如节 10 所述。
3. 在信道 36 下，填写 12。

返回默认值：取消选中相关复选框以返回默认值。根据节 10，未标记的复选框表示使用默认值。

6.8.1.1.3 共存和天线选择

为了使每个平台配置具有最大的灵活性，器件引脚支持多种分配共存和天线选择接口的方案。这些方案因器件系列（CC3135 和 CC3235x）的不同而略有差异。

共存模式：

- 关 - 不使用 BLE 共存（默认设置）
- 单天线 - 如果平台包含可在 BLE 和 Wi-Fi 之间共享单个天线的射频开关，选择此选项。此选项需要分配两个 GPIO，其中一个是从 BLE 到射频开关的输入，另一个是从 Wi-Fi 到射频开关的输出。
- 双天线 - 如果平台具有独立的 BLE 和 Wi-Fi 天线，选择此选项。在此模式下，BLE 在需要信道时向 Wi-Fi 发出信号，此时 Wi-Fi 停止正在进行的传输。此模式仅需要使用一个 I/O。

天线选择模式：

- 禁用
- 天线 1 - 静态选择天线 1
- 天线 2 - 静态选择天线 2
- 自动选择

请参阅图 6-15。

图 6-15. 共存和天线选择

6.8.1.2 器件身份 (DICE 和 CSR, 仅适用于 CC323xS/SF 器件)

设备标识组合引擎 (DICE) 是来自可信计算组织 (TCG) 的一个安全标准。它旨在提高物联网 (IoT) 安全性, 面向微控制器等器件。DICE 标准指定了一个基于加密器件身份的硬件和软件框架, 用于通过制造商提供的云服务器 (例如 Azure IoT 云服务) 进行身份验证和证明。

CC313x 和 CC323x 采用 DICE 协议, 这种协议为增强安全性和隐私性奠定了基础, 而无需添加昂贵的可信平台模块 (TPM)。在 CC313x 和 CC323x 上运行 DICE 实现方案, 通过使用 TLS 连接中的客户端证书链, 在制造商提供的云服务器中验证单个芯片身份和应用程序代码镜像。该链包含两个证书: 由器件 ID 密钥签名的别名证书和器件 ID 证书。

证书签名请求 (CSR) 是创建和签署证书的常用方法。它通过器件公钥和由私钥签名的数据创建。

德州仪器 (TI) 提供一种工具, 使器件能够在内部生成 PKCS #10 格式的 CSR, 从而简化了为 SimpleLink Wi-Fi 器件创建 CSR 的过程。

NOTE

该功能的 CSR 生成组件可替代节 9 中描述的 CSR 功能。

NOTE

DICE 功能要求 SP 版本为 4.4.1.3_3.1.0.5_3.1.0.19 或更高版本。

“Device Identity” 页面包含以下部分:

- 器件身份配置 (请参阅节 6.8.1.2.1)
- 证书配置 (请参阅节 6.8.1.2.2)
- 证书信息 (请参阅节 6.8.1.2.3)

6.8.1.2.1 器件身份配置

若要开启或关闭 DICE 功能，请选择下列其中一种模式：

- Enable DICE
- Disable DICE

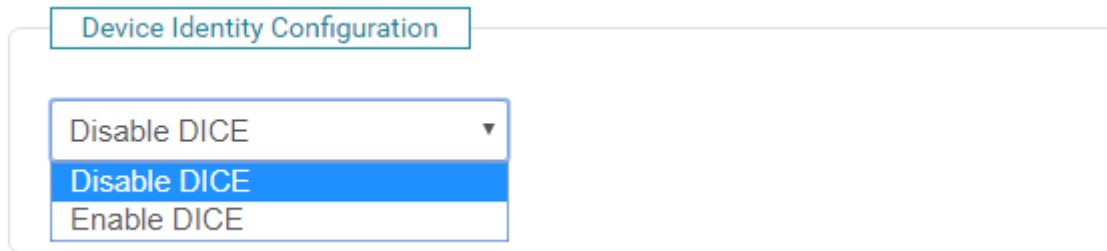


图 6-16. 器件身份配置

NOTE

选择“Enable DICE”时，无法禁用证书配置。

6.8.1.2.2 证书配置

证书配置可设置为如下模式之一。每种模式都会打开证书信息中的特定字段。

- 证书签名请求 (CSR) (参阅节 6.8.1.2.3.1)
- 自签名证书 (参阅节 6.8.1.2.3.2)
- 禁用 (仅当选“Disable DICE”时可用)

当“Certificate Configuration”已启用，且“Device Identity Configuration”设置为“Disable DICE”时，可用此功能中的 CSR 部分代替节 9 中的 CSR。

如果 DICE 已启用，则应为“Certificate Configuration”指定“CSR”或者“Self-Signed Certificate”。

也可在“Certificate Configuration”中添加一个允许重写证书文件的令牌。

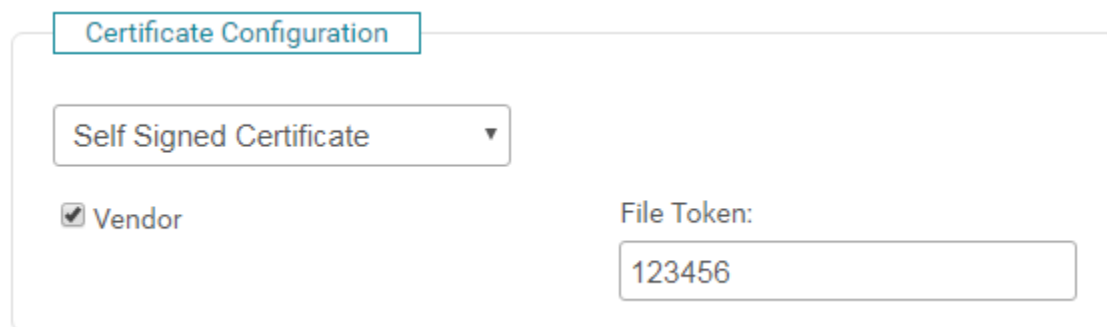


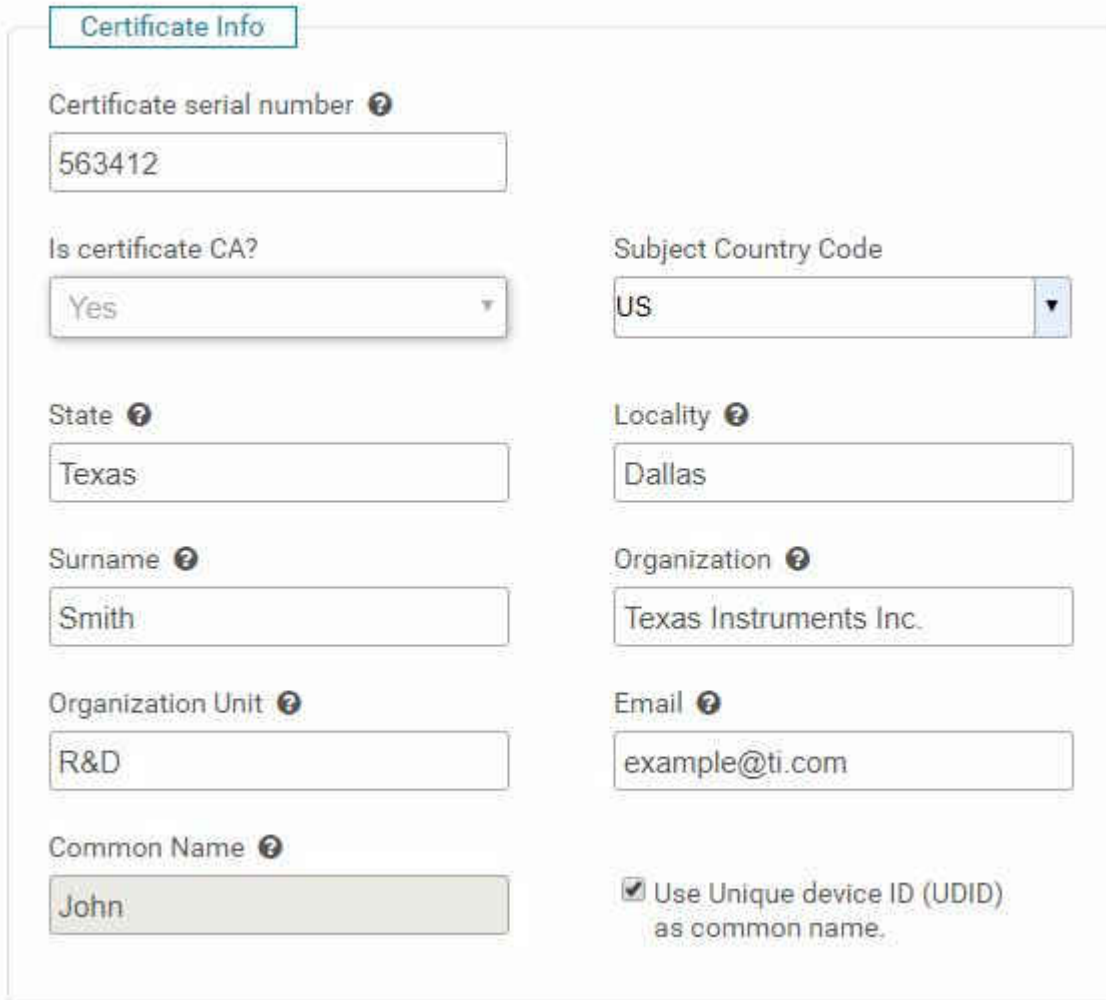
图 6-17. 证书配置

6.8.1.2.3 证书信息


6.8.1.2.3.1 证书签名请求选项

1. 序列号
2. 是否需要证书客户端验证？(选择 DICE 时，设置为 Yes (是))
3. 国家/地区代码
4. 州
5. 地点
6. 姓氏
7. 组织

8. 组织部门
9. 电子邮件
10. 通用名称
11. 将唯一器件 ID (UDID) 用作通用名称




Certificate Info


Certificate serial number 


563412


Is certificate CA?


Subject Country Code


State 


Locality 

Surname 

Organization 

Organization Unit 

Email 

Common Name 

Use Unique device ID (UDID) as common name.

图 6-18. 证书签名请求选项

6.8.1.2.3.2 自签名证书选项

1. 序列号
2. 有效期起始/结束时间
3. 是否需要证书客户端验证
4. 国家/地区代码
5. 州
6. 地点
7. 姓氏
8. 组织
9. 组织部门
10. 电子邮件
11. 通用名称
12. 将唯一器件 ID (UDID) 用作通用名称

Certificate Info

Certificate serial number ?

Certification validity start ? Certification validity end ?
Y M D Y M D

Is certificate CA? Subject Country Code

State ? Locality ?

Surname ? Organization ?

Organization Unit ? Email ?

Common Name ? Use Unique device ID (UDID) as common name.

图 6-19. 自签名证书选项

6.8.1.2.4 仅限 CSR 使用示例

本节介绍了如何创建 CSR 文件。

1. 根据节 6.8.1.2.1，必须选择“Disable DICE”。
2. 根据节 6.8.1.2.2，必须选择“Certificate Sign Request”。
3. 根据节 6.8.1.2.3.1 填写字段。
4. 在简单模式下（节 6.5）或高级模式对器件进行编程。
5. 使用节 8.1 所述的 CSR 读取工具提取 CSR 文件。输出文件名为“csr.der”。
6. 签署 CSR 文件。
7. 使用节 8.1 所述的写证书工具将已签名证书写回器件。
8. 继续使用已编程的应用。

6.8.1.2.5 DICE 用法示例

1. 根据节 6.8.1.2.1，必须选择“Enable DICE”。
2. 根据节 6.8.1.2.2，选择“Certificate Sign Request”以与制造商提供的支持 DICE 的云服务器一同使用。

3. 根据节 6.8.1.2.3.1 填写字段。
4. 在简单模式下 (节 6.5) 或高级模式下对器件进行编程。
5. 使用节 8.1 所述的 CSR 读取工具提取 CSR 文件。输出文件名为 “csr.der” (通过 CLI 命令使用 CSR 读取工具时可更改名称：请参阅节 7.3.1)。
6. 使用制造商提供的云服务器在 CSR 文件上添加签名。
7. 使用节 8.1 所述的写证书工具将已签名证书写回器件。
8. 若要安全连接到制造商的云服务器，请使用：
 - 创建时命名为 aliascert.pem 并存储在根目录中的 DICE 证书链。
 - 存储在 sys 目录中且名为 tempkey02.der 的 DICE 证书链的私钥。
9. 继续使用已编程的应用，连接到制造商的云服务器。

6.8.2 角色设置

- 常规设置
 - 器件模式：
 - 初始角色 (AP/P2P/站点)
 - 国家/地区代码
 - 器件名称
 - 连接策略：
 - 自动连接
 - 快速连接
 - Wi-Fi 直连
 - 自动配置
 - 自动配置外部确认
- STA/Wi-Fi 直连器件
 - WLAN 设置
 - 网络设置
- AP/Wi-Fi 直连
 - WLAN 设置
 - 网络设置

6.8.3 HTTP 服务器

通过 ImageCreator 配置 HTTP 服务器的选项如图 6-20 所示。

Primary Port

Secured

 Enable ROM Pages

Port Number

Secondary Port

Enable Secondary Port

Port Number

Security settings

HTTP server **certificate** file name

HTTP server **private key**

Important: The key file is saved into the project's directory

Enable Client Authentication
 client **CA certificate** file name

图 6-20. HTTP 服务器

6.9 添加服务包

此服务包用于升级器件软件。服务包文件由 TI 提供，包含在 SDK 软件包中。服务包文件名为 `sp_<release_version_number>.bin`，存放在 `<SDK_PATH>\tools\cc32xx_tools\servicepack-cc3xXX` 文件夹中。TI 建议将服务包添加到编程镜像中，但这并非强制性要求。如果尚未对服务包进行编程，器件使用其出厂代码。

添加服务包时，用户选择文件存放位置。但是，ImageCreator 不保留原始文件的链接。若要修改服务包，应重新选择一个新的服务包文件。

6.10 添加受信任的根证书目录

受信任的根证书文件由 TI 提供。TI store 包含受信任的已知根证书清单，以及已撤销的证书清单。您可以在 [CC3x20, CC3x3x SimpleLink™ Wi-Fi® Internet-on-a chip™ 解决方案内置安全功能应用报告](#) 中找到 TI 支持的证书清单。

有一个默认的由 ImageCreator 使用的受信任根证书目录用于 ImageCreator 安装。可通过选择不同的文件及其签名文件来覆盖默认的受信任根证书。ImageCreator 没有指向所选受信任根证书的原始文件的链接。若要修改受信任根证书的内容，请选择一个新文件。

6.10.1 供应商证书目录

备用启动加载程序允许客户在签署自己的固件镜像时使用自签名证书。

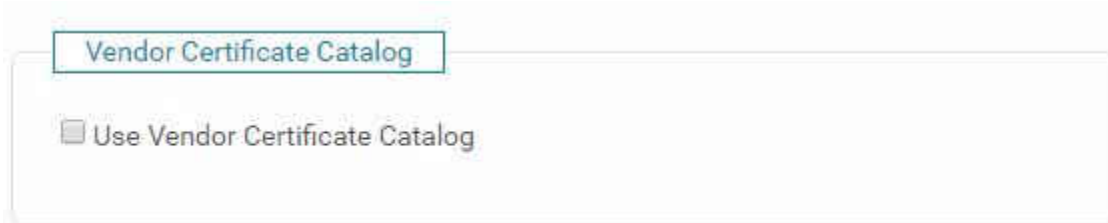


图 6-21. 供应商证书目录

6.10.2 OTP

信任根保存在一次性编程 (OTP) 存储器中，因此，它在编程后无法被替换。在本节中，供应商可添加器件特定的信息并添加签名，以验证硬件平台的真实性以及是否由供应商生产。如需更多信息，请参阅 [根据 SimpleLink™ WiFi® 器件用户指南进行供应商器件验证](#)。

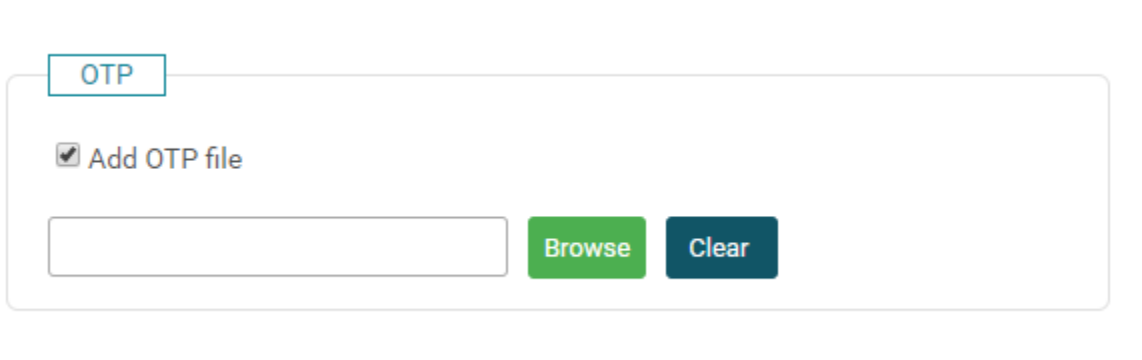


图 6-22. OTP 章节

6.11 添加主机应用文件 (CC32xx)

SimpleLink ImageCreator 允许为 CC32xx 器件添加主机应用文件。如需在简单模式下添加 MCU 镜像文件，请参阅 [节 6.5](#)。在高级模式下，在“User Files”页面，点击下拉菜单箭头并选择“Select MCU Image”，如 [图 6-23](#) 所示。



图 6-23. MCU 镜像高级模式

点击“Browse”按钮并从本地驱动中选择 MCU 镜像文件。会显示“File Properties”对话框。请参阅 [节 6.12.3](#)。

配置主机应用文件属性后，应点击“Save”按钮。主机应用文件创建成功，器件上显示如下文件名：

- CC32xx R/RS : /sys/mcuimg.bin
- CC32xxSF : /sys/mcuflashing.bin

对于安全器件，必须使用安全签名标志创建主机文件。若要在未来通过 OTA 对文件进行更新，用户必须通过公共写入标志打开该文件。另外，设置该文件的大小上限时，还应考虑到文件未来的增量，因为文件创建后无法更改大小上限。

6.11.1 CC32xxSF 器件的主机应用

CC3220SF 应用需要在主机文件开头添加 20 个字节的 SHA1。SHA1 是采用哈希算法对主机文件内容进行计算的结果。文件签名是根据主机文件内容计算的，包括 SHA1。创建步骤如下：

1. 在文件 (`host_final`) 开头添加 SHA1。
2. 计算主机签名 (`host_final` 文件的签名)。

ImageCreator 提供两种添加主机应用的方法：

- 应用文件 (包括 SHA1) 由用户创建并包含文件签名。输入为 `host_final` 文件以及 `host_final` 文件签名。
- `host_file` 文件 (不包含 SHA1) 由用户创建：
 1. ImageCreator 计算 SHA1 和 `host_final` 文件。
 2. ImageCreator 使用 DER 格式的私钥作为输入，计算 `host_final` 文件签名。

NOTE

通过设置，将主机应用的文件大小上限增加 20 个字节，以添加文件 SHA1。

6.12 用户文件

ImageCreator 支持文件操作，包括添加或删除文件、创建目录和查看文件属性，因此，可将用户文件添加到镜像中。

将光标移动到文件或目录图标上时，可进行文件操作。选择一个文件后，该文件将保存为工程的一部分。

ImageCreator 文件不链接到原先选择的文件。若要修改文件，应删除并重新添加文件中的内容。

6.12.1 已签名安全用户文件

对于已签名安全文件，ImageCreator 必须接收已签名证书和文件签名。有关如何检索已签名证书以及如何创建文件签名的详细信息，请参阅用户手册中的安全文件系统一节 (搜索 `sl_FsClose ()` 函数)。

当证书链接到另一个证书时，链接证书的名称应在原证书的“issued to”字段中。链接中的所有证书在添加由它们签名的文件之前都会被添加到工程中。

ImageCreator 通过如下方法添加已签名安全文件：


- 设置一个文件签名。
- 接收私钥。通过该密钥，ImageCreator 可创建文件签名。

NOTE

在这两种方法中，都必须提供包含公钥的已签名证书。

若要在未来通过 OTA 对文件进行更新，请使用带公共写入标志的添加文件。还可使用供应商令牌标志并定义文件主令牌。

6.12.2 添加一个文件


若要添加文件，用户应点击  图标，或从相应文件夹中拖放所需文件。

文件添加后，会显示“File Properties”对话框。请参阅图 6-24。

NOTE

根据规范中的要求，需要添加一个证书，该证书中，最后一行以 UNIX 结束行格式（仅使用“\n”）结束；任何其他符号都可能导致异常行为（例如，禁止使用 DOS 结束行格式“\r\n”）。

6.12.3 编辑一个文件

若要编辑文件属性，请选择文件并点击  图标。会显示“文件属性”对话框，如图 6-24 所示。

File Name:

Max File Size: (actual size: 121)

Failsafe
 Secure
 Static
 Vendor

File Token:

Public Write
 Public Read
 No Signature Test

Private Key File Name:

Certification File Name:

图 6-24. “文件属性”对话框

表 6-2 列出了各标志选项。

表 6-2. 标志选项

标志选项	说明
Failsafe	编辑文件时不会出现故障。有关更多详细信息，请参阅用户指南中的安全文件系统章节，可搜索： SL_FS_CREATE_FAILSAFE
Secure	文件在串行闪存上加密。有关更多详细信息，请参阅用户指南中的安全文件系统章节，可搜索： SL_FS_CREATE_SECURE
No Signature Test	仅与安全文件相关。安全文件默认需要一个签名。有关更多详细信息，请参阅用户指南中的安全文件系统章节，可搜索：SL_FS_CREATE_NOSIGNATURE
Static	仅与安全文件相关。每次打开文件进行写入时都不会替换令牌。有关更多详细信息，请参阅用户指南中的安全文件系统章节，可搜索：SL_FS_CREATE_STATIC_TOKEN
Vendor	仅与安全文件相关。主令牌由供应商设置。有关更多详细信息，请参阅用户指南中的安全文件系统章节，可搜索：SL_FS_CREATE_VENDOR_TOKEN
Public Write	仅与安全文件相关。文件可在没有令牌的情况下写入。有关更多详细信息，请参阅用户指南中的安全文件系统章节，可搜索：SL_FS_CREATE_PUBLIC_WRITE
Public Read	仅与安全文件相关。文件可在没有令牌的情况下读取。有关更多详细信息，请参阅用户指南中的安全文件系统章节，可搜索：SL_FS_CREATE_PUBLIC_READ
No Signature Test	仅与安全文件相关。安全文件默认需要一个签名。有关更多详细信息，请参阅用户指南中的安全文件系统章节，可搜索：SL_FS_CREATE_NOSIGNATURE

表 6-3. 其他文件属性

文件属性	说明
File Token	仅在使用供应商标志时相关。有安全保护的文件的令牌。
Signature File Name	仅在使用安全签名标志时相关，应通过在本地计算机上浏览来选择签名文件。文件名应出现在测试框中。
Certification File Name	仅在使用安全签名标志时相关，应显示先前添加的证书列表。用户应从列表中选择相关文件。
Maximum size	要为文件分配的存储空间大小。默认情况下，ImageCreator 将最大空间设置为文件的实际大小。若要在未来通过 OTA 对文件进行更新，请将空间大小上限设置为文件未来的大小上限（不能更改现有文件的文件大小上限）。器件将向上舍入文件的大小上限以关联串行闪存块的大小（4096 字节）；如需更多信息，请参阅用户手册中的安全文件系统章节。

若要对文件进行重命名，请使用“文件属性”对话框中的“File Name”字段，如图 6-25 所示。

File Name:

图 6-25. 重命名文件

6.12.4 添加一个文件夹

若要添加一个新的文件夹，请在根文件夹中找到目标位置并点击  图标。

6.12.5 删除一个文件或文件夹

勾选需要删除的文件或文件夹。在“Action”下拉菜单中，选择“Remove Selected”（请参阅图 6-26），然后点击“Apply”（请参阅图 6-27）。



图 6-26. 删除文件或文件夹 (1 , 共 2)

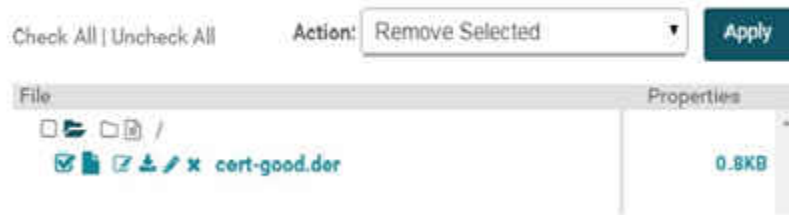



图 6-27. 删除文件或文件夹 (2 , 共 2)

用户也可直接点击“X”（删除）按钮删除单个文件或文件夹。

6.12.6 覆盖文件

用户可点击铅笔按钮 () 覆盖文件。

6.12.7 用户文件操作监视器

用户可将文件拖放到用户文件区域，如图 6-28 所示。在操作期间，用户文件操作监视器显示该操作过程。

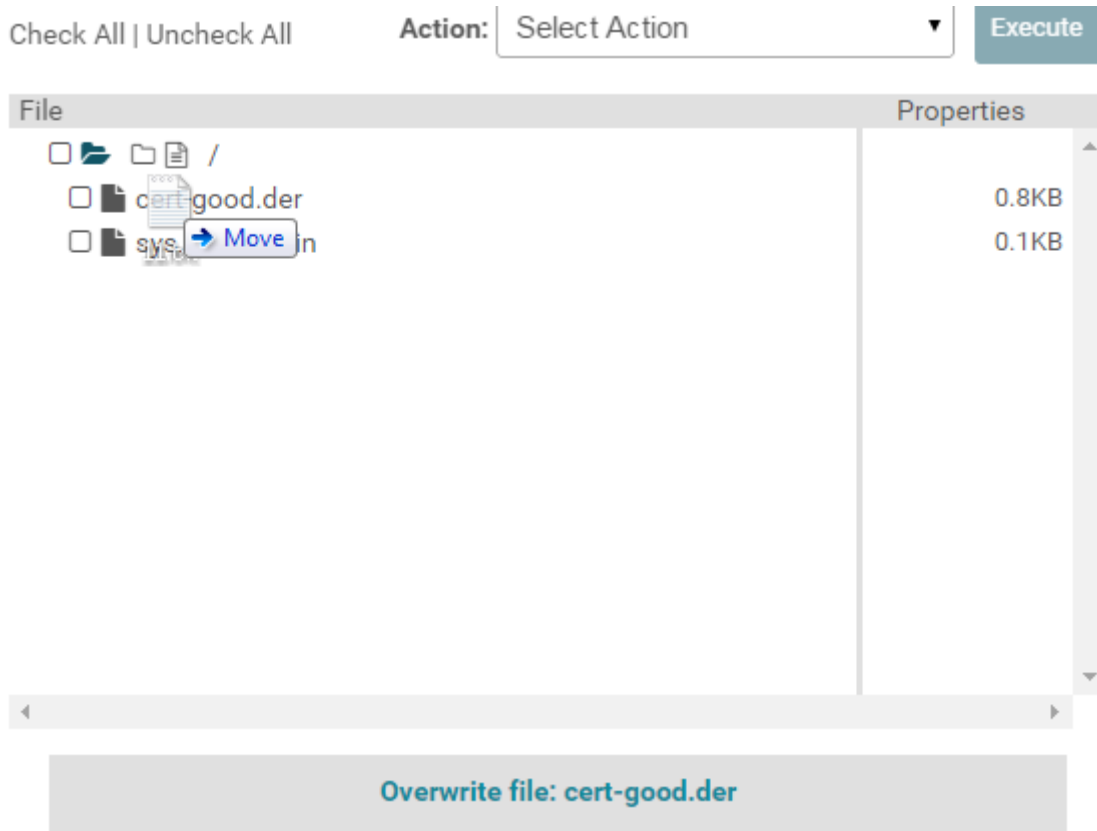


图 6-28. 用户文件操作监视器

6.13 器件文件浏览器

使用开发镜像编程的器件允许浏览其文件、用户文件和系统文件，并且可执行以下各节中列出的若干操作。


大多数系统文件无法在文件列表中查看，也没有读取/写入权限。

联机浏览器编辑器件串行闪存上的文件。此更改不影响使用由脱机浏览器准备的编程镜像文件。

若要查看联机文件浏览器，点击以下图标：




6.13.1 添加文件

若要添加文件，请点击  图标。

显示“File Properties”对话框。请参阅节 6.12。

6.13.2 编辑一个文件

移动光标到一个文件名处，并点击  图标。

显示“File Properties”对话框。请参阅图 6-24。

6.13.3 添加一个文件夹

若要添加新的文件夹，请在根文件夹中找到目标位置并点击  图标。

6.13.4 删除一个文件

若要删除文件，请将光标移动到需要删除的文件处并点击“Delete”按钮，如图 6-29 所示。如果文件有安全保护，将显示关于文件令牌的提示。



图 6-29. 删除文件

6.13.5 检索一个文件

若要在器件中检索（上传）文件，请将光标移到文件上并点击“Get File”按钮，如图 6-30 所示。如果文件有安全保护，将显示关于文件令牌的提示。

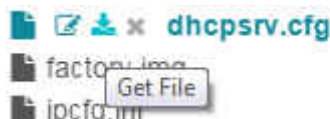
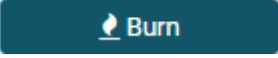



图 6-30. 获取文件

6.14 从一个工程中创建镜像

配置阶段结束时，可通过点击“Generate Image”按钮  完成镜像创建。在此阶段，将创建所有类型的编程镜像文件，并且这些文件可在 ImageCreator 安装目录下找到。

若要创建镜像，请点击“Create Image”按钮：。

若要了解如何创建加密镜像，请参阅节 6.7.1。

6.15 创建 OTA

若要创建 OTA 镜像，点击“Create OTA”按钮 。

会显示 OTA 安全对话框，如图 6-31 所示。

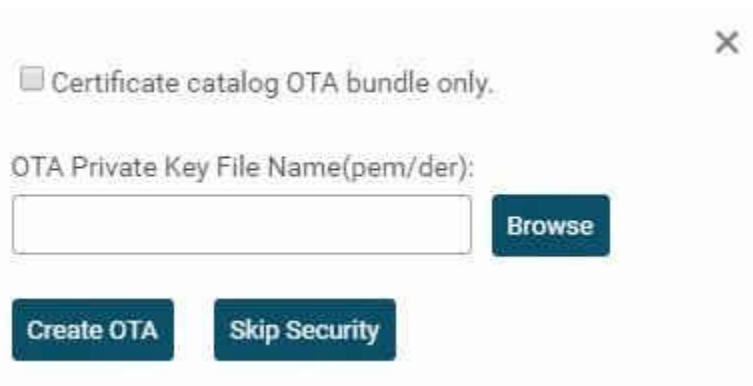


图 6-31. OTA 私钥文件名

6.15.1 创建带安全标志的一个 OTA

1. 点击“Browse”按钮并加载 OTA 私钥文件名（PEM 或 DER 文件）。
2. 点击“Create OTA”按钮。

6.15.2 创建无安全标志的 OTA

点击“Skip Security”按钮。

6.15.3 仅使用证书目录 OTA 捆绑包

如果必须更新证书目录，则应分两步执行 OTA。第一步，TAR 文件应仅包含证书目录及其签名。在这种情况下，必须勾选 Certificate catalog OTA bundle only 复选框。一旦存在证书目录，TAR 文件中不能包含其他文件。第二步，TAR 文件包含所需的所有其他文件，例如 MCU 镜像、服务包等。

6.16 保存一个镜像

成功创建一个镜像后，可点击“Save Image”按钮。

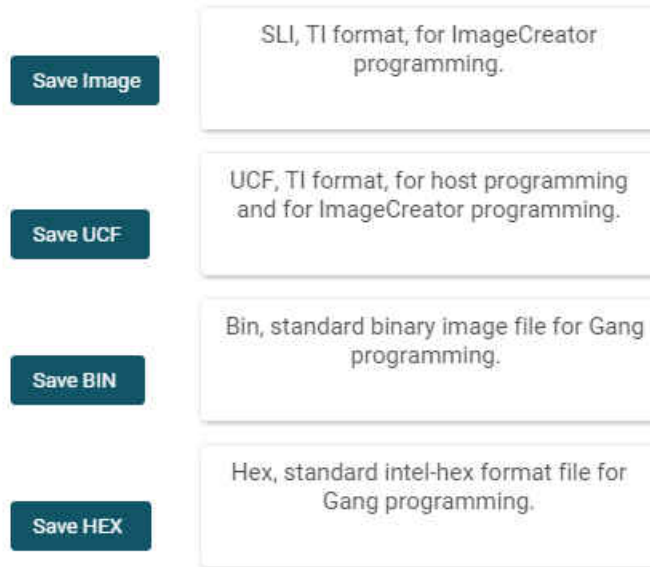


图 6-32. 保存镜像

6.17 Programming.bin 和 Programming.hex

标准二进制和 Intel 十六进制文件用于通过外部串行闪存编程工具进行编程。

- Programming.ucf (TI 专有编码) 用于主机编程。
- Programming.sli (TI 专有编码) 用于 ImageCreator 编程。

6.18 从一个已打开的工程进行镜像编程

请参阅节 5.5。

6.19 使用 .sli 文件对一个镜像进行编程

创建 .sli 文件后，ImageCreator 的任何实例都可使用它来对器件进行编程。若要使用已存在的 .sli 文件，点击“Welcome”页面上的“Program Image”按钮，如图 6-33 所示。

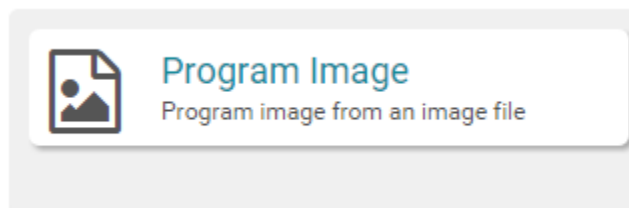


图 6-33. 编程镜像

显示“Program Image”对话框。设置要编程的 .sli 文件，并点击“Program Image”按钮。

若要对有安全保护的 .sli 文件镜像进行编程，请设置用于镜像加密的密钥文件（请参阅节 6.7.1），然后点击“Program Image”按钮（请参阅图 6-34）。

Program Image

Image File Name

CC3200_Provisioning_1.0.11.3.0.1.4.sli

Image Key File Name

Secondary Bootloader

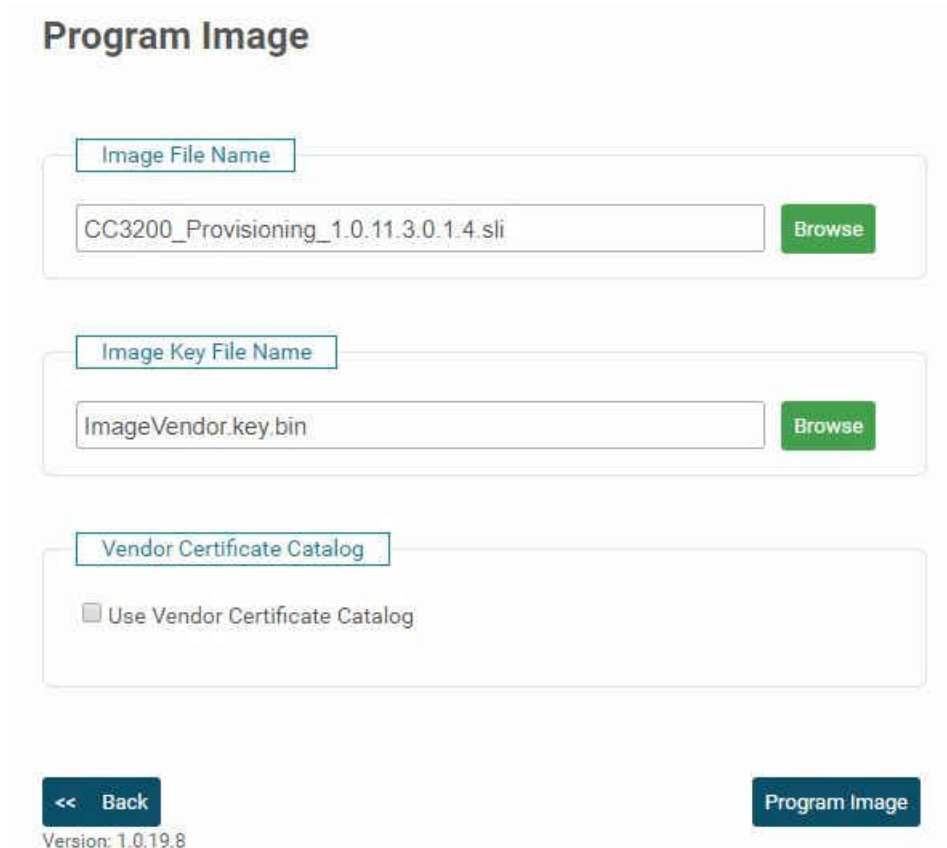
Use secondary bootloader

图 6-34. 编程镜像

签名文件工具位于工具窗格中。在用户选择私钥（有关受支持的密钥格式的更多信息，请参阅文件系统用户手册）以及要签名的文件后，可获得二进制或 base 64 的已签名文件（请参阅节 8.3）。

6.20 带密钥的安全镜像

若要对有安全保护的 .sli 文件镜像进行编程，请设置用于镜像加密的密钥文件（请参阅节 6.7.1），然后点击“Program Image”按钮（请参阅图 6-35）。



Program Image

Image File Name
CC3200_Provisioning_1.0.11.3.0.1.4.sli **Browse**

Image Key File Name
ImageVendor.key.bin **Browse**

Vendor Certificate Catalog
 Use Vendor Certificate Catalog

<< **Back** **Program Image**

Version: 1.0.19.8

图 6-35. 编程镜像

7 命令行

导航到 UniFlash 安装目录：

- Windows 系统：

```
cd c:\ti\uniflash_X.X
```

- Linux 系统：

```
cd /home/YOUR_USER/ti/uniflash_4.0
```

- Mac OS X 系统：

```
cd /Users/YOUR_USER/ti/uniflash_X.X
```

使用 `dslite shell` 脚本以在 `cc31xx/cc32xx` 模式下发送命令：

- Windows 系统：

```
dslite.bat --mode cc31xx COMMAND

dslite.bat --mode cc32xx COMMAND
```

- Linux 和 Mac OS X 系统：

```
./dslite.sh --mode cc31xx COMMAND

./dslite.sh --mode cc32xx COMMAND
```

7.1 工程命令

7.1.1 添加一个文件或设置 MCU 镜像

基本命令：

```
project add_file --name PROJECT_NAME --file MCU_FILENAME.bin --mcu

-或-

project add_file --name PROJECT_NAME --file MCU_FILENAME.bin --fs_path /path/filename.ext
```

必填参数：

<code>--name PROJECT_NAME</code>	需要使用的工程名称
<code>--file FILENAME</code>	需要添加的文件
<code>--fs_path /path/filename.ext</code> -或- <code>--mcu</code>	SimpleLink 文件系统中的文件路径或名称 该文件是一个 MCU 镜像

可选参数：

<code>--sign SIGNATURE_FILENAME</code> -或- <code>--priv PRIVATE_KEY_FILENAME</code>	用于对 MCU 镜像文件进行签名的签名文件 用于为该文件生成签名的私钥
<code>--flags flag1,flag2,...</code>	文件标志。可用值包括： <code>failsafe</code> 、 <code>secure</code> 、 <code>nosignaturetest</code> 、 <code>static</code> 、 <code>vendor</code> 、 <code>publicread</code> 、 <code>publicwrite</code> 、 <code>nofailsafe</code> 和 <code>nopublicwrite</code> 。 其中， <code>nofailsafe</code> 和 <code>nopublicwrite</code> 为否定标志。默认设置为 <code>failsafe</code> 和 <code>publicwrite</code> 为 <code>true</code> 。要禁用该值，请设置为否定标志。
<code>--token TOKEN_NUMBER</code>	文件令牌，32 位无符号整数，与供应商标志一起使用
<code>--max_size MAX_SIZE_IN_BYTES</code>	为 SimpleLink 文件系统中的文件分配的最大字节数。
<code>--cert CERT_NAME</code>	需要使用的（从器件文件系统获取的）证书文件； 若省略，该文件名不变。若需擦除，请使用 <code>--cert ""</code> 命令。
<code>--overwrite</code>	若该文件已经存在，强制覆盖该文件。
<code>--project_path PROJECT_PATH</code>	工程文件夹的路径
<code>--cfg_json PATH_CFG_JSON_FILE</code>	<code>cfg.json</code> 文件的完整路径

注：

- 如果使用“`--mcu`”命令，安全属性和最大文件大小将根据工程类型自动选择，但也可通过“`--flags`”和“`--max_size`”选项覆盖。
- 如果工程中已经存在该文件，则该命令提示出错并会退出；若需强制覆盖，请使用“`--overwrite`”命令。

示例：

设置 MCU 镜像：

```
project add_file --name MY_PROJECT --file MCU_FILENAME.bin --mcu
```

添加文件：

```
project add_file --name MY_PROJECT --file MY_TEXT_FILE.txt --fs_path /mydir/myfilename.txt --  
  
flags "failsafe,publicwrite"
```

设置安全 MCU：

- 首先，添加证书文件（证书始终位于 SimpleLink 文件系统的根目录中）：

```
project add_file --name MY_PROJECT --file MY_CA_CERT.der --fs_path CA_CERT
```

- 设置 MCU 镜像，并使用非默认文件夹中的工程私钥签名：

```
project add_file --name MY_PROJECT --project_path FULL_PROJECTS_PATH --file MCU_FILENAME.bin --  
mcu --priv MY_PRIVATE_KEY_FILENAME.key --  
  
cert CA_CERT
```

7.1.2 设置服务包

服务包文件由 TI 提供，包含在 SDK 软件包中。服务包文件名为 `sp_<release_version_number>.bin`，存放在 `<SDK_PATH>\tools\cc32xx_tools\servicepack-cc3xXX` 文件夹中。

基本命令：

```
project set_sp --name PROJECT_NAME --file SP_FILENAME.bin
```

必填参数：

--name PROJECT_NAME	需要使用的工程名称
--file FILENAME	服务包 .bin 文件

可选参数：

--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

7.1.3 编程镜像 (来自一个工程)

基本命令：

```
project program --name PROJECT_NAME
```

必填参数：

--name PROJECT_NAME	需要使用的工程名称
---------------------	-----------

可选参数：

--vendor_cert	使用供应商证书目录
--otp_file OTP_INF_FILE	OTP 文件
--port COM<port_number>	需要使用的 COM 端口
--reconfig RECONFIG_FILENAME.json	动态应用重配置文件 (无需保存到工程)
-dev	工程处于开发模式时确认编程
--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径
--script_path SCRIPT_PATH	开关机脚本文件夹的完整路径

注：

- 如果工程处于开发模式下，则不允许在未传递 “--dev” 参数的情况下进行编程。
- 可选择将 COM 端口作为参数调用。在这种情况下，用户需要提供并使用 `power_off_com.py` 或 `power_on_com.py` 来重置器件。

示例：

```
project program --name PROJECT_NAME --project_path PROJECT_PATH --script_path SCRIPT_PATH --port COM11
```

7.1.4 设置受信任的根证书目录

基本命令：

```
project set_certstore --name PROJECT_NAME --file CERT_STORE.lst --sign certstore.lst.signed
```

必填参数：

--name PROJECT_NAME	需要使用的工程名称
--file CERT_STORE.lst	受信任的根证书目录文件
--sign CERT_STORE.lst.signed	受信任的根证书目录签名文件

可选参数：

--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 将“--file”和“--sign”指定为空时，会切回使用所提供的默认文件。例如：`project set_certstore --name PROJECT_NAME --file "" --sign ""`

7.1.5 导出工程

基本命令：

```
project export --name PROJECT_NAME --file EXPORTED_PROJECT.zip
```

必填参数：

--name PROJECT_NAME	需要使用的工程名称
--file EXPORTED_PROJECT.zip -或- --path PATH	已导出工程的存档文件名 带有含时间戳的文件名的存档写入路径

可选参数：

--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 使用“--path”命令可为存档生成带时间戳的文件名，并在给定路径中创建文件。

7.1.6 导入工程

基本命令：

```
project import --file EXPORTED_PROJECT.zip
```

必填参数：

--file EXPORTED_PROJECT.zip	已导出工程的存档文件
-----------------------------	------------

可选参数：

--overwrite	强制覆盖同名的现有工程
--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 如果存在同名工程，则该命令将拒绝导入；若需强制覆盖，请使用“--overwrite”命令。

7.1.7 克隆工程

基本命令：

```
project clone --name PROJECT_NAME --new NEW_PROJECT_NAME
```

必填参数：

--name PROJECT_NAME	需要克隆的工程名称
---------------------	-----------

<code>--new NEW_PROJECT_NAME</code>	克隆工程的新工程名称
-------------------------------------	------------

可选参数：

<code>--overwrite</code>	强制覆盖同名的现有工程
<code>--with_key</code>	将加密密钥复制到克隆工程
<code>--project_path PROJECT_PATH</code>	工程文件夹的路径
<code>--cfg_json CFG_JSON_PATH</code>	cfg.json 文件的完整路径

注：

- 如果存在同名工程，则该命令拒绝导入；若需强制覆盖，请使用“`--overwrite`”命令。
- 如果工程有加密密钥，除非使用“`--with_key`”命令，否则不会将加密密钥复制到克隆工程。

7.1.8 新建工程

基本命令：

```
project new --name NEW_PROJECT_NAME
```

必填参数：

<code>--name NEW_PROJECT_NAME</code>	需要创建的工程名称
--------------------------------------	-----------

可选参数：

<code>--device</code>	器件类型 (CC3220S、CC3220R、CC3120R、CC3220SF、CC3235S、CC3235SF 和 CC3135R)
<code>--mode</code>	器件模式 (生产/开发)
<code>--description</code>	工程说明：
<code>--project_path PROJECT_PATH</code>	工程文件夹的路径
<code>--cfg_json CFG_JSON_PATH</code>	cfg.json 文件的完整路径

示例：

```
project new --name PROJ_NAME --device DEVICE_TYPE --mode MODE --description DESC
```

注：

- 默认值为 CC32xx。生产模式无工程描述。

7.1.9 创建镜像 (从工程中)

基本命令：

```
project create_image --name PROJECT_NAME --sli_file IMAGE_FILENAME.sli
```

必填参数：

<code>--name PROJECT_NAME</code>	需要使用的工程名称
<code>--sli_file IMAGE_FILENAME.sli</code>	用于编写 SLI 文件的镜像文件名

可选参数：

<code>--ucf_file IMAGE_UCF_FILENAME.ucf</code>	用于编写的 UCF 镜像文件名
<code>--bin_file IMAGE_BIN_FILENAME.bin</code>	用于编写的 BIN 镜像文件名
<code>--hex_file IMAGE_HEX_FILENAME.hex</code>	用于编写的 HEX 镜像文件名
<code>--reconfig RECONFIG_FILENAME.json</code>	动态应用重配置文件 (无需保存到工程)

--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

7.1.10 重新配置工程

基本命令：

```
project reconfig --name PROJECT_NAME --file RECONFIG_FILENAME.json
```

必填参数：

--name PROJECT_NAME	需要使用的工程名称
--file RECONFIG_FILENAME.json	应用重置文件

可选参数：

--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 新配置被应用并保存到工程中。
- reconfig.json 文件中包含如下参数：
 - "macAddress"
 - "startRole"
 - "countryCode"
 - "devMac"
 - "apSsid"
 - "apPassword"
 - "deviceName"
 - "staNetwork"
 - "ip"
 - "mask"
 - "gateway"
 - "dns"
 - "dhcp"
 - "apNetwork"
 - "ip"
 - "mask"
 - "gateway"
 - "dns"
 - "startIp"
 - "lastIp"
 - "https"
 - "prim_port_secured"
 - "access_rom"
 - "prim_port_val"
 - "sec_port_enable"
 - "sec_port_val"
 - "access_ca_cert"
 - "privatekey_file_name"
 - "certificate_file_name"
 - "ca_certificate_file_name"
 - "dice_csr"

- "enable_dice"
 - "create_csr"
 - "use_self_signed_cert"
 - "starts_day"
 - "starts_month"
 - "starts_year"
 - "ends_day"
 - "ends_month"
 - "ends_year"
 - "csr_vendor"
 - "csr_token"
 - "csr_use_udid_as_common_name"
 - "common_name"
 - "certificate_number"
 - "is_certificate_CA"
 - "country_code"
 - "state"
 - "locality"
 - "surname"
 - "organisation"
 - "organisation_unit"
 - "email"
- 每个文件都应以 SimpleLink 名称开头。示例：

```

[ {
  "SimpleLink":
  [
    {
      "devMac"      : "CC:BB:00:00:00:AA",
      "apSsid"      : "test",
      "apPassword"  : "password",
      "deviceName"  : "device1",
      "staNetwork":
      [
        {
          "ip": "192.168.10.11",
          "mask": "255.255.0.0"
        }
      ],
      "apNetwork":
      [
        {
          "ip": "10.0.0.10",
          "mask": "255.0.0.0"
        }
      ],
      "https":
      [
        {
          "prim_port_secured": false,
          "access_rom": true,
          "privatekey_file_name": "C:\\\\Bugs\\\\Rog\\\\dummy-root-ca-cert",
          "certificate_file_name": "C:\\\\Bugs\\\\Rog\\\\dummy-root-ca-certa",
          "ca_certificate_file_name": "C:\\\\Bugs\\\\Rog\\\\dummy-root-ca-certb"
        }
      ],
      "dice_csr":
      [
        {
          "enable_dice"           : true,
          "create_csr"            : true,
          "use_self_signed_cert"  : false,
          "starts_day"            : "01",
          "starts_month"          : "01",
          "starts_year"           : "2019",
          "ends_day"              : "31",
          "ends_month"            : "12",
          "ends_year"             : "2019",
          "csr_vendor"            : true,
          "csr_token"              : "12213443",
          "csr_use_udid_as_common_name" : true,
          "common_name"           : "Name",
          "certificate_number"     : "1218",
          "is_certificate_CA"      : true,
          "country_code"          : "US",
          "state"                  : "Texas",
          "locality"              : "Dallas",
          "surname"                : "Beres",
          "organisation"           : "Texas Instruments",
          "organisation_unit"      : "Team",
          "email"                  : "email@ti.com"
        }
      ]
    }
  ]
} ]

```

7.1.11 列出可用工程

基本命令：

```
project list
```

可选参数：

--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 输出所有可用工程
- 除非在 `cfg.json` 文件中指定，否则工程目录位于：
 - Win 7/10 - `c:\Users\the_user\.SLImageCreator\projects`
 - Linux - `/home/the_user/.SLImageCreator/projects`
 - Mac OS X - `/Users/the_user/.SLImageCreator/projects`

7.1.12 从工程中创建 OTA 存档

基本命令：

```
project create_ota --name PROJECT_NAME --file TAR_FILE
```

-或-

```
project create_ota --name PROJECT_NAME --path TAR_PATH
```

必填参数：

--name PROJECT_NAME	需要使用的工程名称
--file -或- --path	目标 TAR 文件的文件路径 目标路径 - 将创建 <PROJECT_NAME>.tar 文件

可选参数：

--project_path PROJECT_PATH	工程文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径
--priv	用于安全 OTA 的私钥文件名
--cert_catalog	仅设置证书目录，忽略其他文件

7.2 镜像命令

7.2.1 编程镜像

基本命令：

```
image program --file IMAGE_FILENAME.sli
```

必填参数：

--file IMAGE_FILENAME.sli	镜像文件名
---------------------------	-------

可选参数：

--vendor_cert	使用供应商证书目录
--otp_file <OTP_FILE>	提供 OTP 文件

--key KEY_FILENAME	密钥文件名
--port COM<port_number>	需要使用的 COM 端口
--script_path SCRIPT_PATH	开关机脚本文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 可选择将 COM 端口作为参数调用。在这种情况下，用户需要提供并使用 power_off_com.py 或 power_on_com.py 重置器件。

7.3 工具命令

7.3.1 读取 CSR 文件 (仅适用于 CC323xS/SF 器件)

可使用该命令替代从节 9 获取的 CSR。

基本命令：

```
tools get_csr --out_file OUT_FILENAME
```

必填参数：

--out_file OUT_FILENAME	CSR 输出文件名
-------------------------	-----------

有关 CSR 配置，请参阅节 7.1.10。

7.3.2 编写证书文件 (仅适用于 CC323xS/SF 器件)

可使用该命令替代从节 9 获取的 CSR。

基本命令：

```
tools set_csr --file CERT_FILENAME
```

必填参数：

--file CERT_FILENAME	用于写入器件的证书文件名
----------------------	--------------

7.3.3 签署文件

基本命令：

```
tools sign --file FILENAME --priv PRIVATE_KEY_FILENAME --out_file OUT_FILENAME
```

必填参数：

--file FILENAME	待签署文件
--priv PRIVATE_KEY_FILENAME	用于签名的密钥
--out_file SIGNATURE_FILENAME	签名文件名

可选参数：

--fmt	"BINARY_SHA1"/"BINARY_SHA2"/"BASE64"
-------	--------------------------------------

7.3.4 激活镜像

基本命令：

```
tools activate --key KEY_FILENAME
```

必填参数：

<code>--key KEY_FILENAME</code>	密钥文件名
---------------------------------	-------

可选参数：

<code>--port COM<port_number></code>	需要使用的 COM 端口
<code>--script_path SCRIPT_PATH</code>	开关机脚本文件夹的路径
<code>--cfg_json CFG_JSON_PATH</code>	cfg.json 文件的完整路径

注：

- 可选择将 COM 端口作为参数调用。在这种情况下，用户需要提供并使用 `power_off_com.py` 或 `power_on_com.py` 来重置器件。

7.3.5 创建 otp.meta

基本命令：

```
tools meta --cert CERT_FILE_NAME --out_file OUTPUT_FILE --mac "112233445566" -usechain
```

必填参数：

<code>--cert CERT_FILE_NAME</code>	供应商证书文件名
<code>--out_file OUTPUT_FILE</code>	输出文件名

可选参数：

<code>--mac MAC_ADDRESS</code>	MAC 地址，由 12 个十六进制数字组成
<code>--usechain</code>	使用第二签名。

7.3.6 创建 otp.inf 文件

基本命令：

```
tools inf --algo 2 --sign1 SIGNATURE --sign2 SIGNATURE_2 --meta META_FILE --out_file OUTPUT_FILE
```

必填参数：

<code>--algo ALGO</code>	ALGO 取值：1 - RSASHA1，2 - RSASHA256
<code>--sign1 SIGNATURE</code>	自签名文件名
<code>--meta META_FILE</code>	元节文件名
<code>--out_file OUTPUT_FILE</code>	输出文件名

可选参数：

<code>--sign2 SIGNATURE2</code>	供应商链签名文件名应设置使用元命令的链标志（参阅节 7.3.5）。否则会忽略签名。
---------------------------------	---

7.3.7 创建证书目录

基本命令：

```
tools make_cert_catalog --cert_folder CERT_FOLDER --out_file OUTPUT_FILE
```

必填参数：

<code>--cert_folder</code>	供应商证书文件名
<code>--out_file OUTPUT_FILE</code>	输出文件名

注：

- 所有证书都必须采用 DER 格式才能创建证书存储文件。

7.4 器件命令

7.4.1 获取器件信息

基本命令：

```
device info
```

可选参数：

--json	以 JSON 格式打印到 <code>stderr</code>
--port COM<port_number>	需要使用的 COM 端口
--script_path SCRIPT_PATH	开关机脚本文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 可选择将 COM 端口作为参数调用。在这种情况下，用户需要提供并使用 `power_off_com.py` 或 `power_on_com.py` 来重置器件。
- 使用 “--json” 将带有信息的 JSON 对象打印到 `stderr`，从而允许脚本轻松捕获信息并对其进行反序列化。

示例：

```
device info --json
```

- 捕获 `stderr`
- 若返回码为 0，则对从 `stderr` 中捕获的文字进行反序列化，使其成为一个 JSON 对象。

7.4.2 恢复出厂镜像

基本命令：

```
device restore
```

可选参数：

--defaults_only	仅恢复默认值
--port COM<port_number>	需要使用的 COM 端口
--script_path SCRIPT_PATH	开关机脚本文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 可选择将 COM 端口作为参数调用。在这种情况下，用户需要提供并使用 `power_off_com.py` 或 `power_on_com.py` 来重置器件。

7.5 GUI 配置命令

7.5.1 配置 GUI

基本命令：

```
gui_cfg
```

可选参数：

--project_path PROJECT_PATH	工程文件夹的路径
--port COM<port_number>	需要使用的 COM 端口

--script_path SCRIPT_PATH	开关机脚本文件夹的路径
--cfg_json CFG_JSON_PATH	cfg.json 文件的完整路径

注：

- 该命令允许为 GUI 设置 COM 端口。
- 可选择将 COM 端口作为参数调用。在这种情况下，用户需要提供并使用 power_off_com.py 或 power_on_com.py 来重置器件。
- cfg.json 文件支持以下参数：

projectDir	工程文件夹的路径
tempDir	日志文件夹的路径
scriptDir	开关机脚本文件夹的路径

7.6 GUI 命令附加参数

7.6.1 获得 ImageCreator 版本

-v or -version

示例：

```
SLImageCreator.exe -v
```

7.6.2 静音打印模式

-q or --quiet

-q or --quiet 应在命令之前。

示例：

```
SLImageCreator.exe -q -v
```

8 工具

点击“Welcome”页面上的“Tools”按钮，如图 8-1 所示。

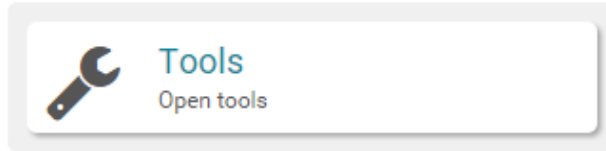


图 8-1. 打开工具

或在工程中点击“Tools”按钮 。

8.1 证书签名请求 (CSR) (仅适用于 CC323xS/SF 器件)

这些命令可代替节 9 中的 CSR。

使用此工具，用户可从器件读取 CSR 文件或编写最终签名证书。此工具支持 DER 和 PEM 格式。

NOTE

使用“Read CSR”选项时，下载文件将被命名为 `csr.der`。

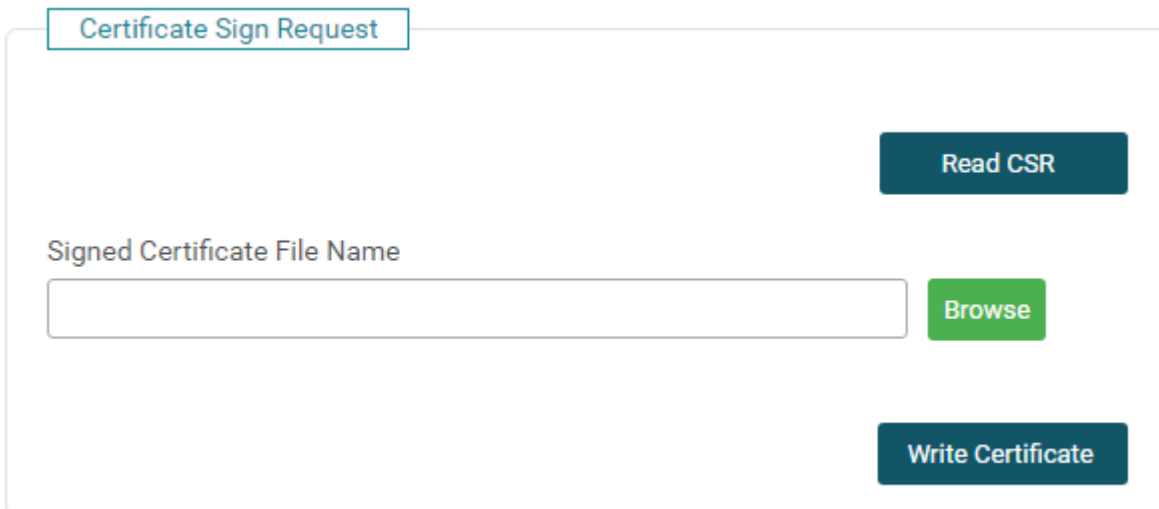
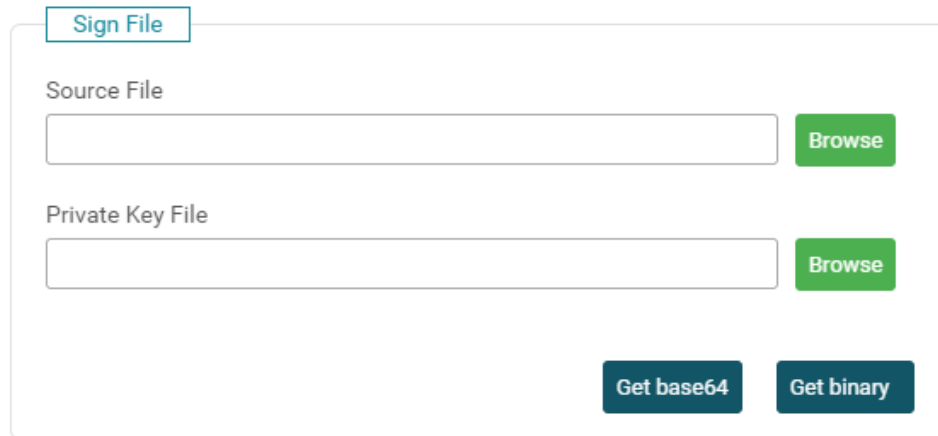


图 8-2. 证书签名请求

有关 CSR 配置，请参阅节 6.8.1.2.3。

8.2 签署文件

通过使用图 8-3 所示的工具，用户可对带有私钥的文件进行签名，并输出二进制 (SHA2) 或 base-64 形式的已签名文件。若要创建二进制 SHA1 签名，请使用命令行（请参阅节 7.3.3）。



The interface is titled "Sign File" in a light blue box. It contains two input fields: "Source File" and "Private Key File". Each field has a corresponding green "Browse" button to its right. At the bottom right of the interface, there are two dark blue buttons: "Get base64" and "Get binary".

图 8-3. 签署文件

8.3 激活镜像

使用如图 8-4 所示的工具，用户可激活已编程的加密镜像。



The interface is titled "Activate Image" in a light blue box. It contains one input field labeled "Image Key File Name" with a green "Browse" button to its right. At the bottom right of the interface, there is a dark blue "Activate" button.

图 8-4. 激活镜像

9 使用 CSR 实用程序

SimpleLink™ Wi-Fi® 的特殊安全功能之一是每个器件有唯一的密钥对。此功能支持签名和验证等加密实用程序，而无需从主机应用直接访问器件的私钥。

这个唯一的密钥对也可用于 TLS 握手中的相互认证。对于这种功能，仅仅拥有器件的唯一密钥对是不够的，但器件必须有服务器支持的权威或信任链签名证书。在大多数情况下，若要创建此证书，仅访问器件的公钥是不够的。创建和签署证书的常用方法是使用证书签名请求 (CSR)，这要求在创建过程中使用私钥对某些数据进行签名。

德州仪器 (TI) 简化了这个流程，并提供了一个工具来获取器件内部生成的 PKCS #10 格式的 CSR。本附录介绍了如何从器件获取 CSR 以及如何对已签名证书进行编程。

9.1 从器件获取 CSR 并复制到文件

若要创建 CSR 文件，请使用 `get_csr.bat file` 命令。对于 Linux/MAC OS 版本，请使用 `./get_csr.sh`。其中，`get_csr.*` 表示一个脚本，用于根据器件类型创建工程，将工程编程到器件，并执行 CSR 实用程序。

9.1.1 编辑 `get_csr.bat` 文件

1. 设置 SDK 路径和服务包名。

```

18 set SDKINSTALLPATH=C:\ti\simplelink_cc32xx_sdk_1_60_00_04
19 rem *****
20 set SPNAME=sp_3.6.0.3_2.0.0.0_2.2.0.6.bin
21 set SP_PATH=%SDKINSTALLPATH%/tools/cc32xx_tools/servicepack-cc3x20
  
```

2. 设置证书清单和签名。

```

rem Certificate store list/signature and path
set CERT_LST=certcatalogPlayGround20160911.lst
set CERT_LST_BIN=certcatalogPlayGround20160911.lst.signed_3220.bin
set CERT_LST_PATH="%SDKINSTALLPATH%/tools/cc32xx_tools/certificate-playground"
  
```

3. 设置证书和密钥名称和路径。

```

22 set DUMMY_CERT_NAME=dummy-root-ca-cert
23 set DUMMY_KEY_NAME=dummy-root-ca-cert-key
24 set DUMMY_CERT_PATH=%SDKINSTALLPATH%/tools/cc32xx_tools/certificate-playground
  
```

4. 设置 CSR 证书的参数。

```

26 rem Certificate serial number (up to 8 bytes)
27 set CERT_SERIAL_NUM=0111000
28 rem Validity period in days (> 0)
29 set VALIDITY=2
30 rem Is certificate CA? (0-No/1-Yes )
31 set ISCA=1
32 rem Subject country( 2 capital letters, i.e US)
33 set COUNTRY="US"
34 rem Subject state (max size is 64)
35 set STATE="State"
36 rem Subject locality (max size is 64)
37 set LOCALITY="Locality"
38 rem Subject surname (max size is 64)
39 set SURNAME="SURNAME"
40 rem Subject organization (max size is 64)
41 set ORGANIZATION="Organization name"
42 rem Subject organization unit (max size is 64)
43 set ORG_UNIT="unit name"
44 rem Subject common name (max size is 64)
45 set NAME="Name"
46 rem Subject email (max size is 64)
47 set EMAIL="email@email.com"

```

5. 验证路径与参数。

从 55 行到 122 行。

6. 设置可执行文件。

```

117 set RUNCMD=SLImageCreator.exe
118 set XDSRESET=xds110reset.exe
119 set CSREXE=csr.exe

```

7. 创建一个新的 ImageCreator 工程。

```

124 echo Creating New Project
125 %RUNCMD% -q project new --name %PROJNAME% --device %PROJDEVICE% --description "project for csr" --overwrite
126

```

8. 设置服务包、证书以及 MCU 镜像。

从 129 行到 145 行。

9. 编程新工程。

```
144 echo Program the image directly from the Project
145 if [%COMPORT%]==[] (
146     %RUNCMD% -q project program --name %PROJNAME%
147 ) else (
148     %RUNCMD% -q project program --port %COMPORT% --name %PROJNAME%
149 )
```

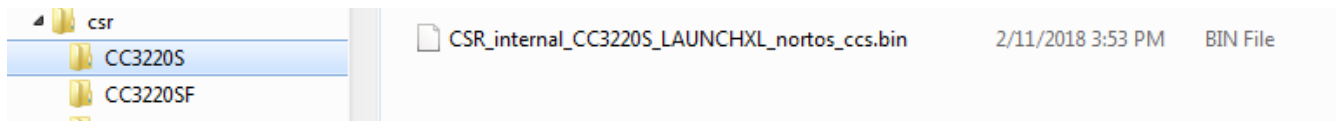
10. 重置器件并等待 10 秒钟。

```
153 echo.
154 echo.
155 echo sleep 10
156 echo.
157 timeout 10 >nul
158
159
160
161 echo.
162 echo reset device
163 echo.
164 %XDSRESET%
```

11. 运行 CSR 实用程序。

编程后，脚本执行 CSR 实用程序 (csr.exe/csr, 从 176 行到 186 行)。此实用程序通过 RS232 与器件交互，并发送脚本输入以在输出文件夹中创建 csr.pem 文件。

相关 MCU 文件存储在 CC3220S/SF 和 CC3235S/SF 文件夹中。



9.1.2 使用 get_csr.bat 文件

调用 get_csr.bat 命令：

提供器件类型 (CC3220SF/CC3220S/CC3235SF/CC3235S)

```
sr>get_csr.bat CC3220SF
```

为避免对 COM 端口进行自动检测，请提供一个 COM 端口作为参数，以便调用 get_csr.bat 命令：

```
sr>get_csr.bat CC3220SF COM13
```

9.2 替换工程中的 CSR 文件

若要替换或添加一个新的 csr.pem 文件，请使用 set_csr.bat (./set_csr.sh) 文件。该批处理命令从工程中删除旧文件 (如果存在) 并添加一个新文件。

9.2.1 用法

```
set_csr <proj_name> <pem_file_name_in_the_project_file_system> <pem_file_source>
set_cert <proj_name> <pem_file_name_in_the_project_file_system> <pem_file_source>
set_cert.bat CC3220SF_CSR_csr_new.pem C:\ImageCreator\CSR\Input\csr_new.pem
```

9.2.2 脚本参数

```
26 set PROJNAME=%1
27 set FILENAME=%2
28 set FILESOURCE=%3
29
```

9.2.3 从工程中删除旧的用户文件

```
39 echo.
40 echo Deleting old pem file from the project
41 echo.
42 %RUNCMD% -q project del_file --name %PROJNAME% --file %FILENAME%
```

9.2.4 添加新文件

```
44 echo Adding csr file to the project
45 echo.
46 %RUNCMD% project add_file --name %PROJNAME% --fs_path %FILENAME% --file %FILESOURCE%
47
```

10 天线 LaunchPad 的默认功率值

10.1 CC3x35 器件的默认值

表 10-1. 2.4GHz 默认值

信道	FCC BO 偏移 [dB]			ETSI BO 偏移 [dB]			JP BO 偏移 [dB]		
	11b	L	H	11b	L	H	11b	L	H
1 [2412MHz]	0	0	0	0	0	0	0	0	0
2 [2417MHz]	0	0	0	0	0	0	0	0	0
3 [2422MHz]	0	0	0	0	0	0	0	0	0
4 [2427MHz]	0	0	0	0	0	0	0	0	0
5 [2432MHz]	0	0	0	0	0	0	0	0	0
6 [2437MHz]	0	0	0	0	0	0	0	0	0
7 [2442MHz]	0	0	0	0	0	0	0	0	0
8 [2447MHz]	0	0	0	0	0	0	0	0	0
9 [2452MHz]	0	0	0	0	0	0	0	0	0
10 [2457MHz]	0	0	0	0	0	0	0	0	0
11 [2462MHz]	0	0	0	0	0	0	0	0	0
12 [2467MHz]	0	0	0	0	0	0	0	0	0
13 [2472MHz]	0	0	0	0	0	0	0	0	0

表 10-2. 5GHz 默认值

信道	子带	FCC [dBm]	ETSI [dBm]	JP [dBm]	额外回退 [dB]		插入损耗 [dB]		
					高	低	TX	RX	天线增益
36 [5180MHz]	U-NII1	14.5	0	0	0	0	4.2	4.2	3.2
40 [5200MHz]		16.125	0	0	0	0			
44 [5220MHz]		16	0	0	0	0			
48 [5240MHz]		16.25	0	0	0	0			
52 [5260MHz]	U-NII-2A	16.125	0	0	0	0	4.2	4.2	3.2
54 [5280MHz]		16	0	0	0	0			
60 [5300MHz]		14.5	0	0	0	0			
64 [5320MHz]		13.625	0	0	0	0			
100 [5500MHz]	U-NII-2C1	13.5	0	0	0	0	4.2	4.2	3.2
104 [5520MHz]		17	0	0	0	0			
108 [5540MHz]		17.125	0	0	0	0			
112 [5560MHz]		17	0	0	0	0			
116 [5580MHz]		0	0	0	0	0			
120 [5600MHz]	U-NII-2C2	0	0	0	0	0	4.2	4.2	3.2
124 [5620MHz]		0	0	0	0	0			
128 [5640MHz]		0	0	0	0	0			
132 [5660MHz]		0	0	0	0	0			
136 [5680MHz]		0	0	0	0	0			
140 [5700MHz]		12.375	0	0	0	0			
144 [5720MHz]		0	0	0	0	0			
149 [5745MHz]	U-NII-2C3	16.5	0	0	0	0	4.2	4.2	3.2
153 [5765MHz]		17	0	0	0	0			
157 [5785MHz]		0	0	0	0	0			
161 [5805MHz]		0	0	0	0	0			
165 [5825MHz]		15.125	0	0	0	0			
169 [5845MHz]		0	0	0	0	0			

10.2 CC3x35MOD 器件的默认值

WARNING

此器件的配置已获得美国联邦通信委员会 (FCC) 设备授权，FCC 标识符为：Z64-CC3235MOD。对器件软件或配置的任意修改都可能导致器件性能发生变化并超出当前提及的 FCC 授权范围。如果您修改器件软件或配置，则可能需要在分销或营销器件或产品之前寻求 FCC 和其他监管授权。

表 10-3. 2.4GHz 默认值

信道	FCC BO 偏移 [dB]			ETSI BO 偏移 [dB]			JP BO 偏移 [dB]		
	11b	L	H	11b	L	H	11b	L	H
1 [2412MHz]	0	0	0	0	0	0	0	0	0
2 [2417MHz]	0	0	0	0	0	0	0	0	0
3 [2422MHz]	0	0	0	0	0	0	0	0	0
4 [2427MHz]	0	0	0	0	0	0	0	0	0
5 [2432MHz]	0	0	0	0	0	0	0	0	0
6 [2437MHz]	0	0	0	0	0	0	0	0	0
7 [2442MHz]	0	0	0	0	0	0	0	0	0
8 [2447MHz]	0	0	0	0	0	0	0	0	0
9 [2452MHz]	0	0	0	0	0	0	0	0	0
10 [2457MHz]	0	0	0	0	0	0	0	0	0
11 [2462MHz]	0	0	0	0	0	0	0	0	0
12 [2467MHz]	0	0	0	0	0	0	0	0	0
13 [2472MHz]	0	0	0	0	0	0	0	0	0

表 10-4. 5GHz 默认值

信道	子带	FCC [dBm]	ETSI [dBm]	JP [dBm]	额外回退 [dB]		插入损耗 [dB]		
					高	低	TX	RX	天线增益
36 [5180MHz]	U-NII1	14.5	0	0	0	0	3	3	4.5
40 [5200MHz]		16.125	0	0	0	0			
44 [5220MHz]		16	0	0	0	0			
48 [5240MHz]		16.25	0	0	0	0			
52 [5260MHz]	U-NII-2A	16.125	12.75	12.75	0	0	3	3	4.5
54 [5280MHz]		16	12.75	12.75	0	0			
60 [5300MHz]		13	12.75	12.75	0	0			
64 [5320MHz]		12.125	12.75	12.75	0	0			
100 [5500MHz]	U-NII-2C1	13.5	12.75	0	0	0	3	3	4.5
104 [5520MHz]		17	12.75	0	0	0			
108 [5540MHz]		17.125	12.75	0	0	0			
112 [5560MHz]		17	12.75	0	0	0			
116 [5580MHz]		14.25	12.75	0	0	0			
120 [5600MHz]	U-NII-2C2	0	12.75	0	0	0	3	3	4.5
124 [5620MHz]		0	12.75	0	0	0			
128 [5640MHz]		0	12.75	0	0	0			
132 [5660MHz]		0	12.75	0	0	0			
136 [5680MHz]		0	12.75	0	0	0			
140 [5700MHz]		12.375	12.75	0	0	0			
144 [5720MHz]		0	0	0	0	0			
149 [5745MHz]	U-NII-2C3	16.5	0	0	0	0	3	3	4.5
153 [5765MHz]		17	0	0	0	0			
157 [5785MHz]		0	0	0	0	0			
161 [5805MHz]		0	0	0	0	0			
165 [5825MHz]		15.125	0	0	0	0			
169 [5845MHz]		0	0	0	0	0			

11 修订历史记录

Changes from Revision G (February 2020) to Revision H (August 2020)
Page

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

Copyright © 2022，德州仪器 (TI) 公司