

# 如何将产品设计从 CC1310 移植到 CC1311 以及 345MHz/ 433MHz/915MHz 的多频段混合系统参考设计



Shuyang Zhong & Island Wei

## 摘要

由于 Flash 的增大，用户可以无需外扩 Flash，仅使用片上 Flash 对应用程序进行 OTA 升级，进一步降低 BOM。另外，如果客户对于传输距离具有更高的要求，在 CC1311 上 TI 还推出了 P 版本（功率增强版），其发射功率可以高达 +20dBm。

本篇文章通过介绍适配 CC1311 的开发工具，讲述了如何将基于 CC1310 的项目移植到 CC1311 平台，并以 TI 15.4 协议栈中的 Collector 和 Sensor 示例工程为例进行实验。

本文的主要内容有：

1. 如何将产品设计从 CC1310 移植到 CC1311
2. 如何使用 CC1311/CC110L 组合实现 915MHz/433MHz/345MHz 的多功能系统

## 内容

<b>1 如何将产品设计从 CC1310 移植到 CC1311.....</b>	<b>2</b>
1.1 硬件移植.....	2
1.2 软件移植.....	2
<b>2 如何使用 CC1311/CC110L 组合实现 915MHz/433MHz/345MHz 的多功能系统.....</b>	<b>5</b>
2.1 方案介绍.....	5
2.2 射频拓扑设计建议以及 BOM.....	5
2.3 Layout 设计.....	7
2.4 测试结果.....	9
<b>3 参考链接.....</b>	<b>11</b>

## 1 如何将产品设计从 CC1310 移植到 CC1311

### 1.1 硬件移植

TI SimpleLink 家族为减轻客户的移植难度，在相同封装相同版本间芯片管脚兼容（pin-to-pin），也就是如果您的新产品从 CC1310 RGZ (7mm × 7mm) VQFN 升级为 CC1311R3 RGZ (7mm × 7mm) VQFN，无需更改 PCB 设计，仅需更改晶振。需要注意的是 CC1311P RGZ 虽然封装与 CC1310R RGZ 相同，但因为内部设计有 PA，所以 R 和 P 版本并非管脚兼容，需要重新画板。

CC1311 有两个版本，CC1311R3 和 CC1311P3。其中 CC1311R3 的 RGZ 封装 (7x7mm QFN) 与相同封装的 CC1310 是 pin to pin 兼容的。如果之前使用的是 RGZ 封装的 CC1310，可直接 pin to pin 移植到 CC1311R3，PCB 布板不需要做任何改动。

在外围电路方面，唯一的改动是高速晶振。CC1311 需要一颗 48MHz 的晶振提供系统时钟，而 CC1310 是 24MHz。

具体的硬件移植指南请参考《[Hardware Migration From CC1310 to CC1312R](#)》。文档名称中是 CC1312R 是因为 CC1312R 芯片同样与 CC1310 和 CC1311 兼容，因此可直接使用此文档作为从 CC1310 到 CC1311R 移植的指南。

如果选用了 CC1311P3 版本，由于 P 版本带有内置功率放大器的原因，所以它与 CC1310 不能直接进行 pin to pin 替换。P 版本的硬件设计可参考《[CC13xx/CC26xx Hardware Configuration and PCB Design Considerations](#)》文档。

### 1.2 软件移植

CC1310 与 CC1311 的软件总体架构是相同的，但在 SDK 的迭代过程中，还是产生了一些软件上的区别。这一节中我们首先从软件整体架构角度列举二者软件上的不同点，以方便开发者从整体上把握软件移植工作。下一章中我们将通过实例阐述具体的移植流程。

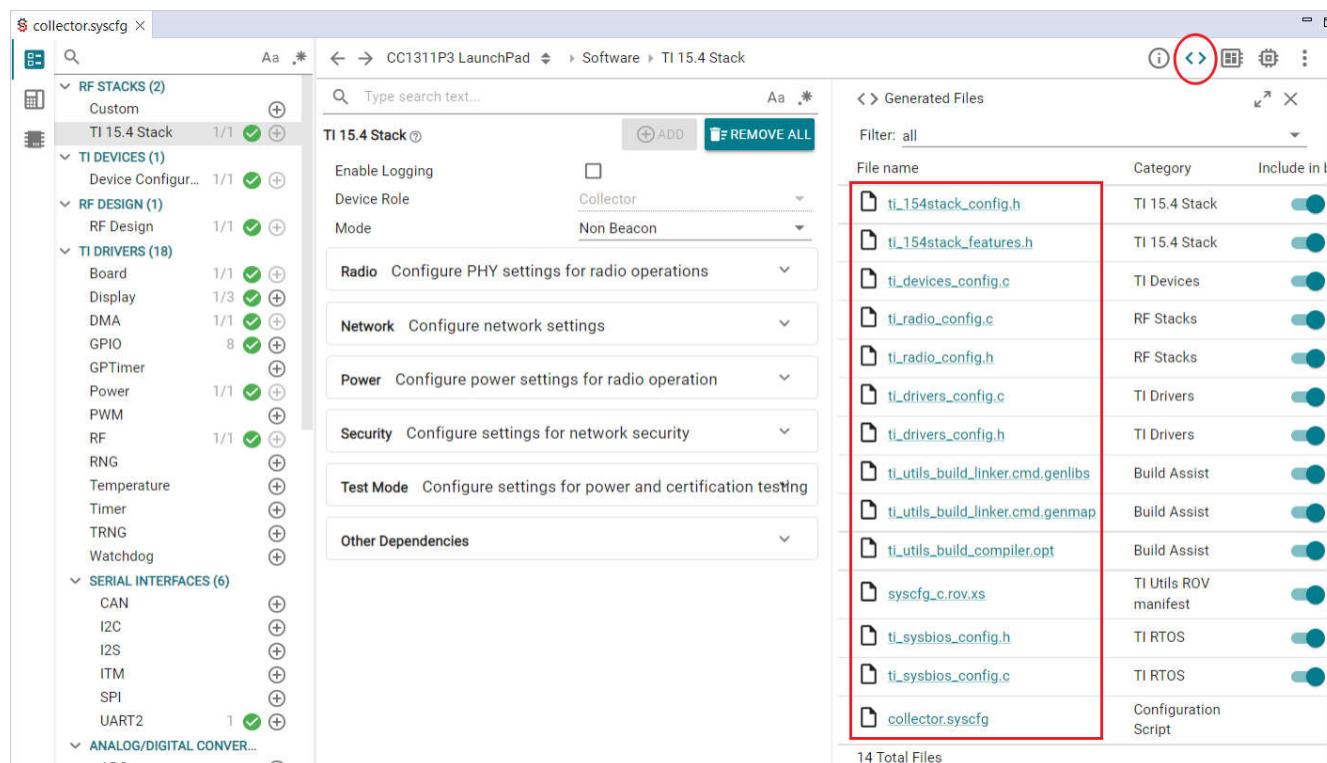
CC1310 与 CC1311 的软件主要有下列区别：

- SysConfig
- TI-RTOS 版本
- 外设驱动（包括 GPIO 和 UART）

#### 1.2.1 System Configuration (SysConfig)

从 CC13x2 这一代芯片开始，SDK 中引入了图形化代码配置工具 SysConfig，来帮助生成芯片硬件相关、以及协议栈相关的部分代码。CC1311 的软件工程也基于 SysConfig 工具。而 CC1310 SDK 中仍使用传统的全代码方式。因此，从 CC1310 向 CC1311 进行移植时，需要了解哪些代码由 SysConfig 生成，并将这些代码转移到 SysConfig 配置中。

我们来看一下 CC1311 工程中由 SysConfig 生成的文件列表，通过它们来介绍 SysConfig 的作用。以 collector\_LP\_CC1311P3\_tirtos7\_ticlang 工程为例，在 CCS 中双击打开 collector.syscfg，点击右上角 Generated Files 图标，可以预览 SysConfig 生成的文件列表：



SysConfig 共生成 14 个文件，其中与用户软件相关的主要有 .c 和 .h 文件，其它文件是工程配置和调试文件。下面表格中汇总了 SysConfig 生成代码的作用，以及在 CC1310 工程中该部分代码对应的位置：

SysConfig 生成文件	代码作用	在 CC1310 工程中对应文件
ti_154stack_config.h	TI15.4 协议栈配置	Application/subg/config.h
ti_154stack_features.h		Application/subg/features.h
ti_devices_config.c	芯片 CCFG 配置	ccfg.c
ti_radio_config.c	RF 物理层配置	Application/MAC/LowLevel/mac_settings.c
ti_radio_config.h		Application/MAC/LowLevel/mac_settings.h
ti_drivers_config.c	外设驱动配置	CC1310_LAUNCHXL.c
ti_drivers_config.h		CC1310_LAUNCHXL.h
ti_sysbios_config.c	TI-RTOS 配置	app.cfg
ti_sysbios_config.h		

更详细的移植步骤请参考 CC1311 SDK 文档中的 [SysConfig Migration](#) 部分。

### 1.2.2 TI-RTOS 移植到 TI-RTOS 7

CC1310 和 CC1311 的软件工程都依赖于 TI-RTOS 实时操作系统。从 CC1311 SDK 5.30 版本之后，TI-RTOS 的版本进行了升级，当前 CC1311 上使用的是 TI-RTOS 7，而 CC1310 上运行的是原来的 TI-RTOS，并未带有版本号命名。

相较 TI-RTOS，TI-RTOS 7 主要有如下升级：

- TI-RTOS 7 通过 SysConfig 生成配置代码
- TI-RTOS 7 不再依赖 XDSTools 工具进行编译，提高了编译速度
- TI-RTOS 7 占用的 flash 和 RAM 更小
- TI-RTOS 7 提升了 debug 体验，可直接进入 kernel 函数中进行单步调试

更详细的更新内容可参考以下链接：

[\[FAQ\] Why use TI-RTOS7? - Sub-1 GHz forum - Sub-1 GHz - TI E2E support forums](#)

对于从 CC1310 升级的用户来说，最大的区别就是 TI-RTOS 7 的相关配置由 SysConfig 生成，取代了之前的 cfg 配置文件。SysConfig 相关的内容在 2.2.1 节中已有介绍。另外在 CC1311 SDK 中也提供了详细的 TI-RTOS 移植步骤：

### [TI-RTOS to TI-RTOS7 Migration](#)

除此之外，例程中部分 RTOS 相关的 API 也发生了改变。CC1311 增加了一层 DPL ( Driver Porting Layer ) 来兼容不同的 RTOS 操作系统，并在例程中使用 DPL API 代替了原来的 TI-RTOS API。因此在移植时请注意以下 API 的变化：

- Clock 模块 API，包括 Clock\_Struct 结构体、Clock\_getTicks() 等，应替换为 ClockP\_Struct 和 ClockP\_getTicks()；
- util\_timer.c 中的 API，如 Timer\_start()、Timer\_stop() 等，应替换为 UtilTimer\_start() 和 UtilTimer\_stop() 等。

关于 DPL API 的详细介绍请参考 [SimpleLink MCU SDK Driver API Reference](#) 文档的 Driver Porting Layer 部分。

### **1.2.3 外设驱动代码移植**

CC1310 和 CC1311 的外设驱动都基于 TI Drivers API。在 TI Drivers 迭代过程中，部分驱动代码发生了变化，主要包括 GPIO 和 UART 两个部分。

#### **1.2.3.1 PIN driver 移植到 GPIO driver**

在 C1310 SDK 中，GPIO 驱动代码基于 PIN driver 和 GPIO driver，在 CC1311 SDK 中对这两部分驱动代码进行了整合，提供了一套更加规范的 GPIO driver ( GPIO++ driver )。

CC1311 不再支持 PIN driver，因此 CC1310 中所有对 PIN driver 的调用需要进行移植。详细的移植步骤请参考 [GPIO++ Driver Porting Guide](#)。

#### **1.2.3.2 UART driver 移植到 UART2 driver**

C1311 SDK 中对 UART 驱动也进行了升级，从 UART driver 变为 UART2 driver。请注意此处命名中的 UART2 只是为了区分旧版本软件而进行的重新命名，与硬件中 UART 的编号并无关联。相比旧版 UART driver，UART2 driver 最大的提升是利用了 DMA 来进行 UART 收发，降低了收发时 CPU 的处理负担，从而提高了程序的整体效率。从 UART driver 到 UART2 driver 的移植步骤请参考 [UART to UART2 Porting Guide](#)。

## 2 如何使用 CC1311/CC110L 组合实现 915MHz/433MHz/345MHz 的多功能系统

### 2.1 方案介绍

现如今，在 Sub1G 组设备中，868MHz/915MHz 相对比 345MHz 更繁忙。当你设计网关并想与现有的著名设计兼容时，345MHz 在现有市场上占有一定份额，如 Honeywell 5800 series, 2GIG, Qolsys. 支持 OOK 调制器，并在与 SPDT 相同的射频路径下，切换一些上限以支持 433MHz。对于这种情况，我们建议使用 CC1311+CC110L 混合设计来满足这一要求。CC1311 和 CC110L 之间我们建议使用 SPI 接口，我们在数据表中有 CC110L 的所有文档，包括 SPI 接口的文档(第 5.5 章 4 线 SPI)。

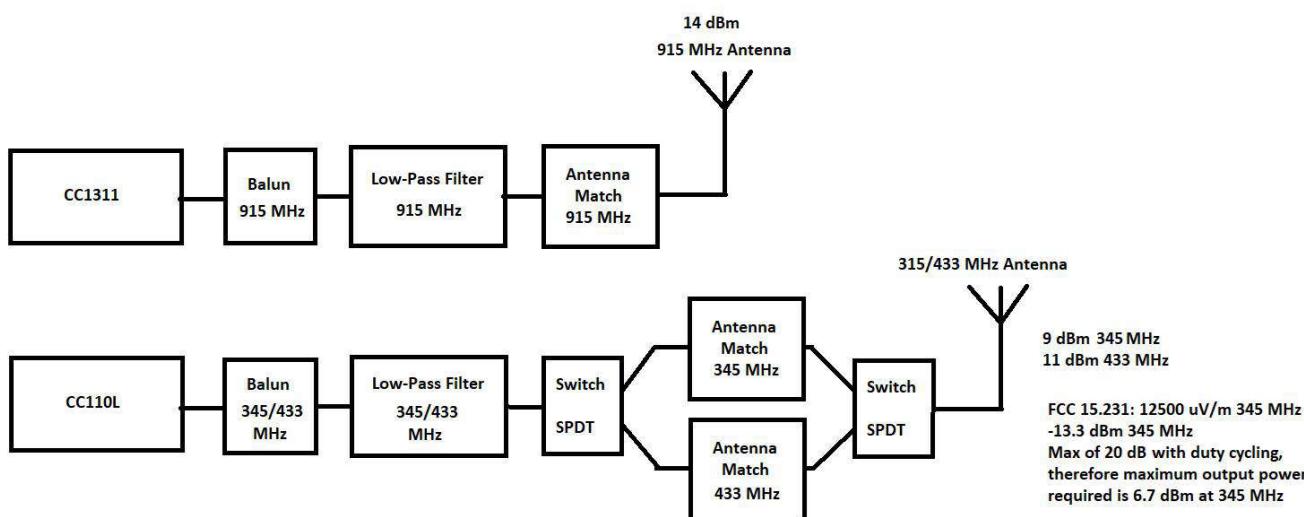
该混合系统的目标是在 868/915MHz 上使用 TI 15.4 协议栈，CC110L 支持 345MHz 上霍尼韦尔 5800 系列传感器的 OOK 调制，只有两个天线路径，并且应符合 FCC 的规定。

根据 FCC 15.231，在 345 MHz 时，最大输出功率为 12500 uV/m。这与 -13.3 dBm 相关，即使最大负载循环利用率为 +20 dB，这也使得最大输出功率为 +6.7 dBm，这对于非常短的数据包长度是允许的。根据 FCC 的规定，更长的数据包长度和输出功率将进一步降低。

CC110L 在 345 MHz 的输出功率为 +12 dBm，远大于 +6.7 dBm。由于在 345 MHz 和 433 MHz 双频工作才是关键，用户应该不需要 12/14 dBm 的输出功率。因此，345/433 MHz 天线必须有两个不同的天线匹配。根据 GND 平面的大小、天线尺寸和位置的不同，小型设备的天线带宽只能在 10-20 MHz 区域。因此，需要两个天线匹配才能覆盖 345 MHz 和 433 MHz。

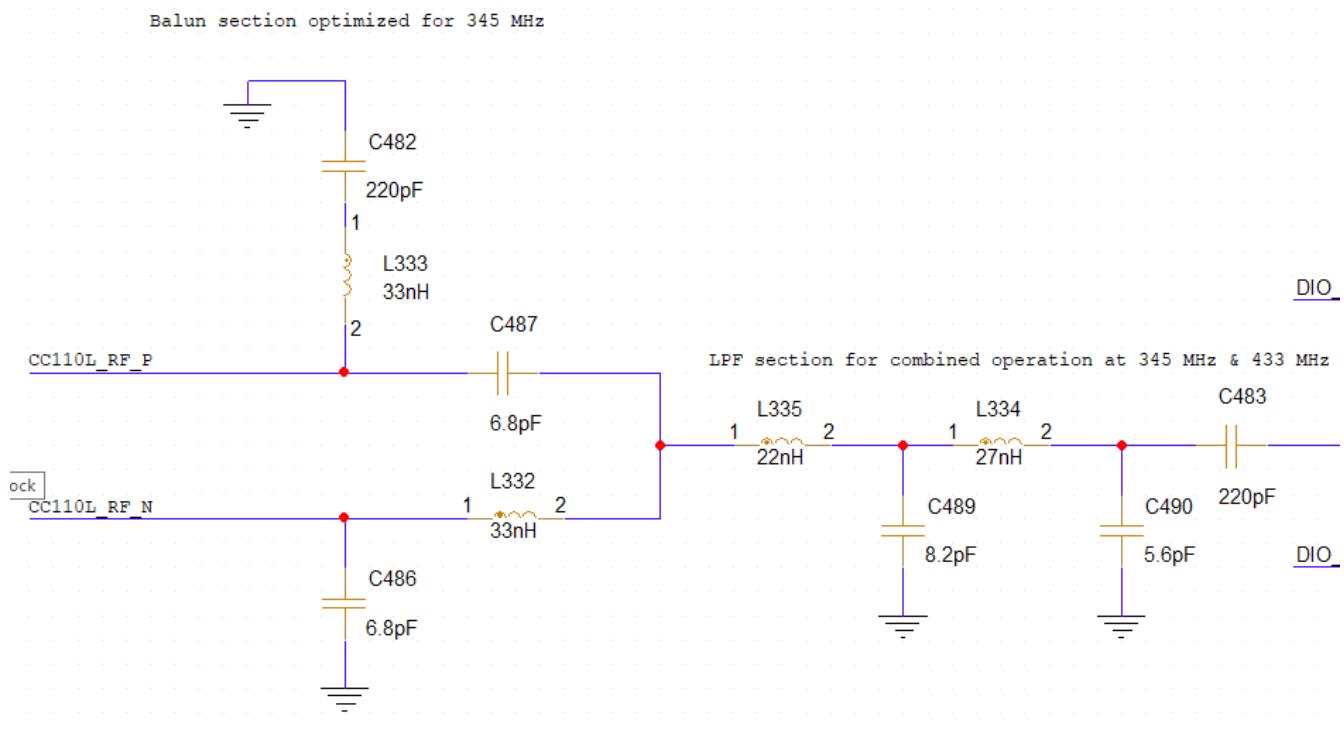
在 345MHz 和 433MHz 的最佳天线匹配下，使用一个物理天线将提供比单个天线更大的发射辐射输出功率。下面的框图只是使用 345 MHz 和 433 MHz 的一个天线，因为该天线在物理上会很大。也就是说，只使用一个天线 (345/433MHz) 而不是两个 345MHz 和 433MHz 的天线。如果可以使用两个天线来处理 345MHz 和 433MHz，那么只需要一个 SPDT 开关。

射频设计要求是在 14dBm 的最佳输出时为 345MHz，射频规格要求是 >10dBm 的 433MHz，我们建议采用以下拓扑：

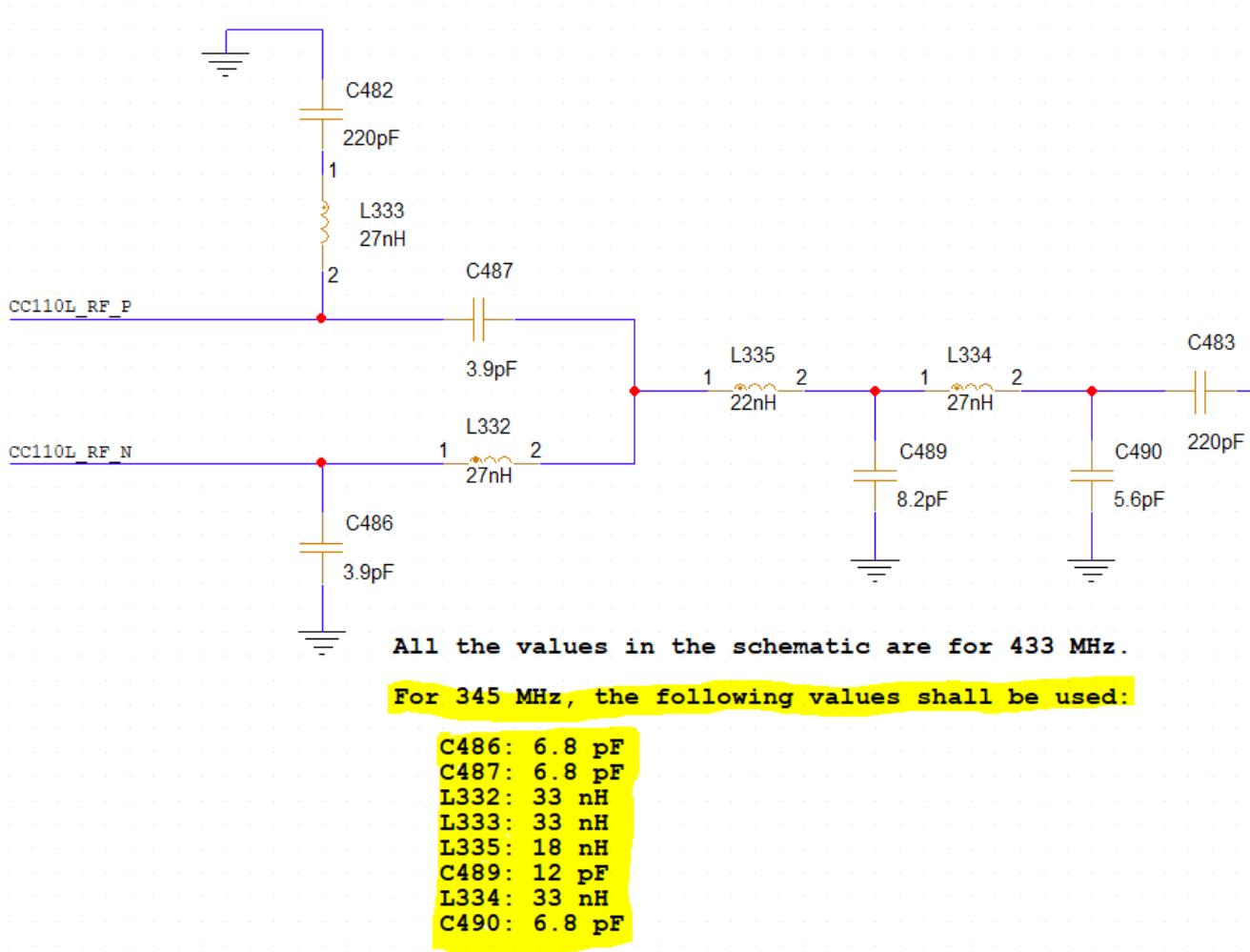


### 2.2 射频拓扑设计建议以及 BOM

BOM 主要为 345MHz 优化，但仍然需要在 433MHz 做些配置。

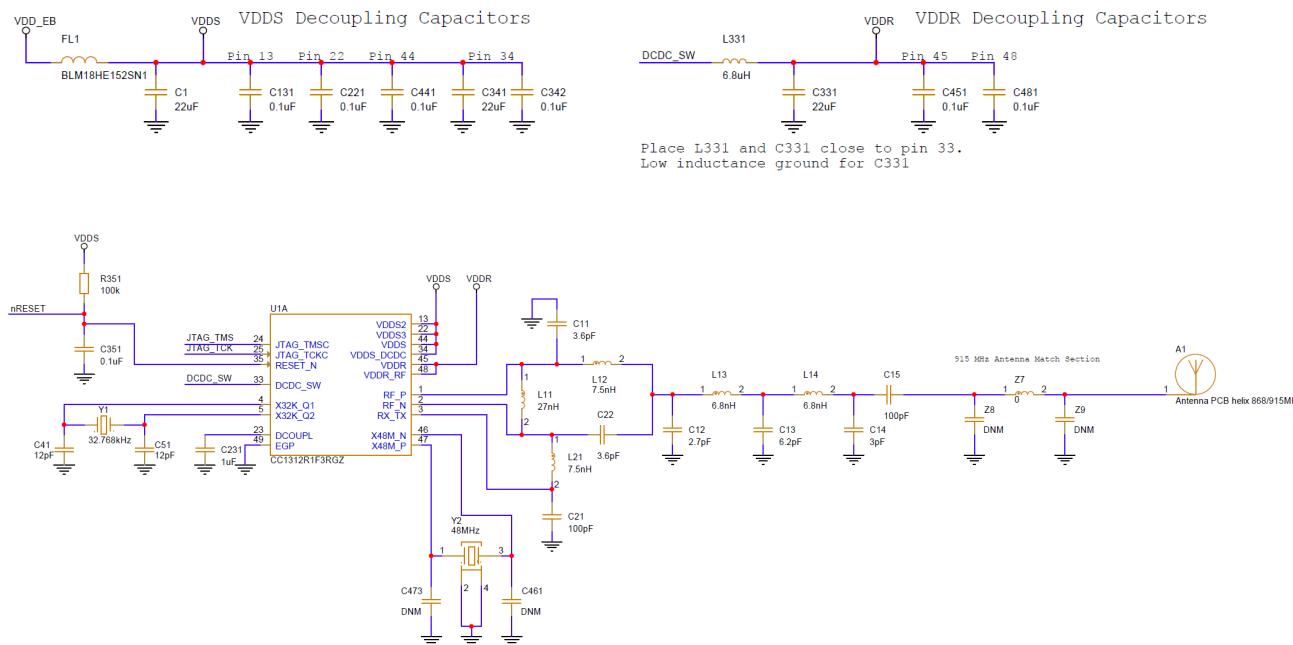


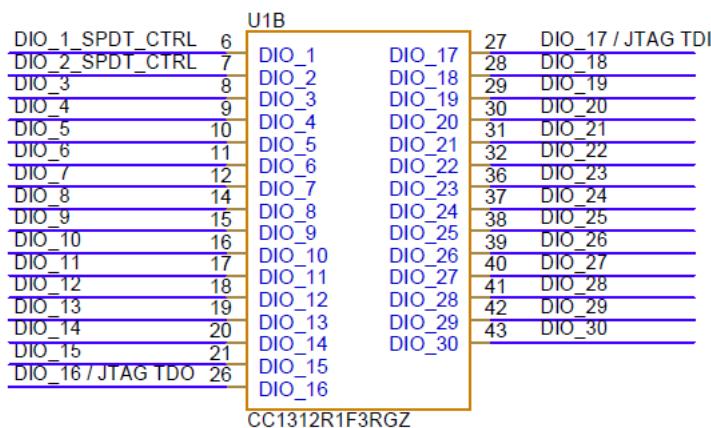
此 BOM 在 345 MHz 和 433 MHz 时允许 12 dBm。在 345 MHz 工作时，LPF 始终可以变得更清晰，但这将开始导致 433 MHz 时输出功率降低。如果您将为 345 MHz 也加上 LPF，那么 433 MHz 的输出功率将下降到约 8 dBm，但两个频段的谐波都将得到改善，特别是 345 MHz。



## 2.3 Layout 设计

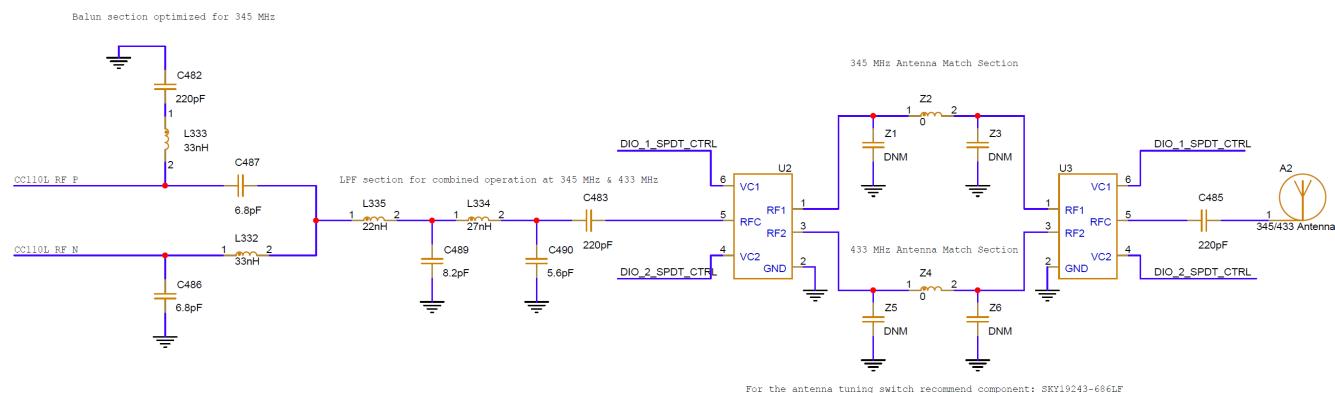
### 2.3.1 CC1311 部分





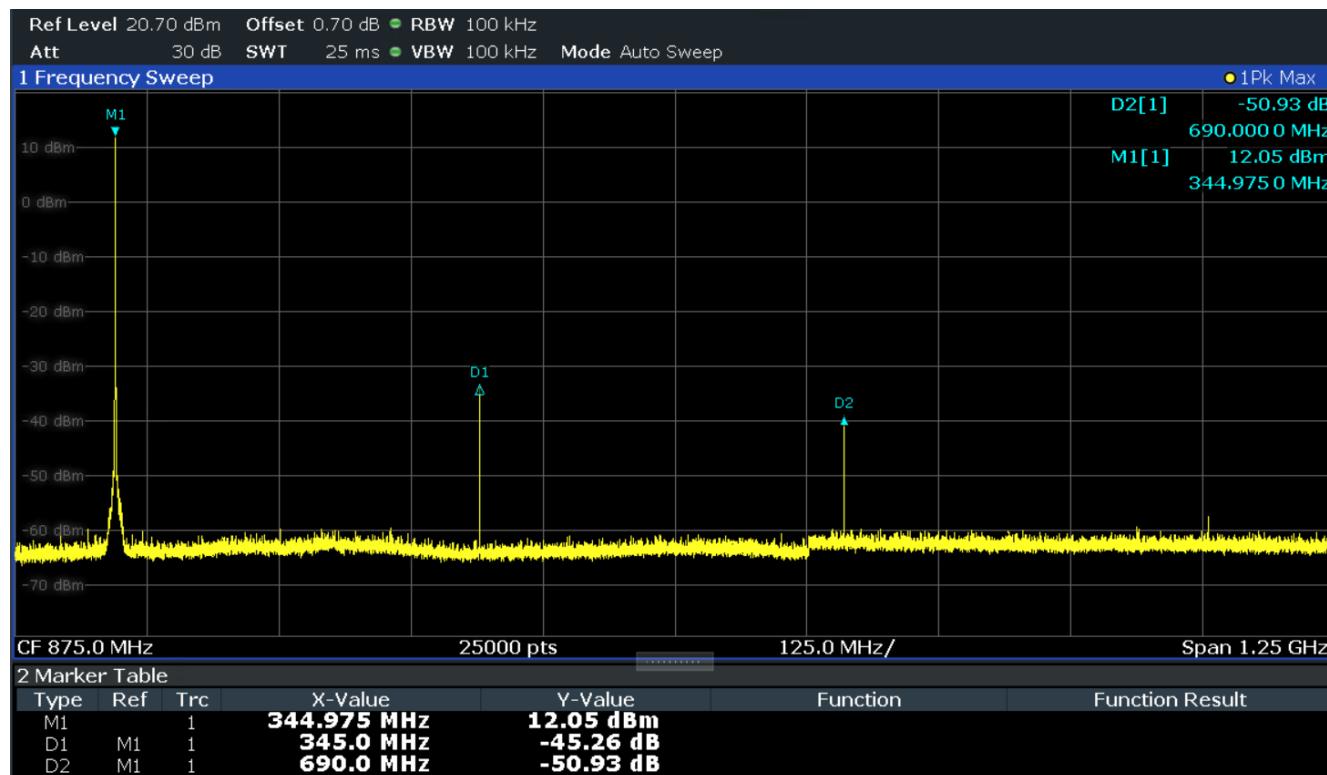
### 2.3.2 CC110L 部分

CC110L RF Front End for 345 MHz &amp; 433 MHz Single Antenna

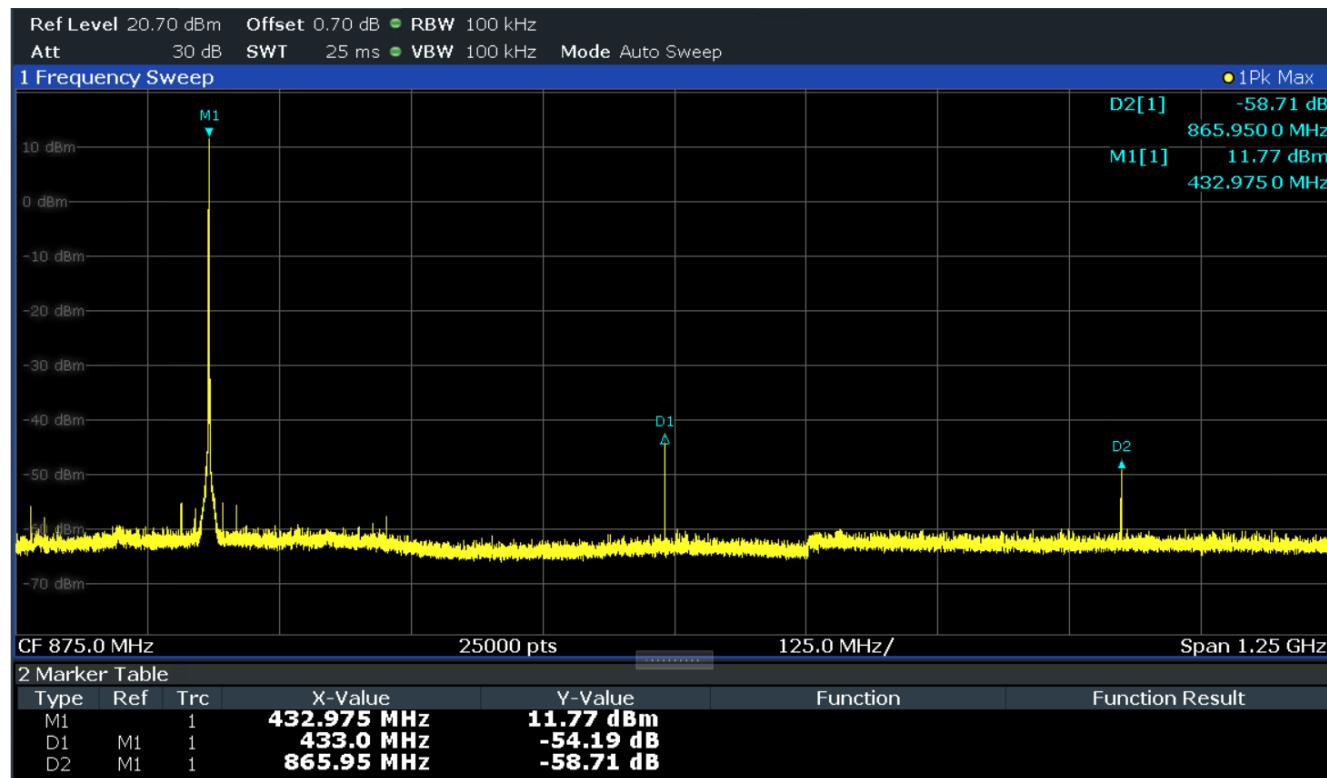


## 2.4 测试结果

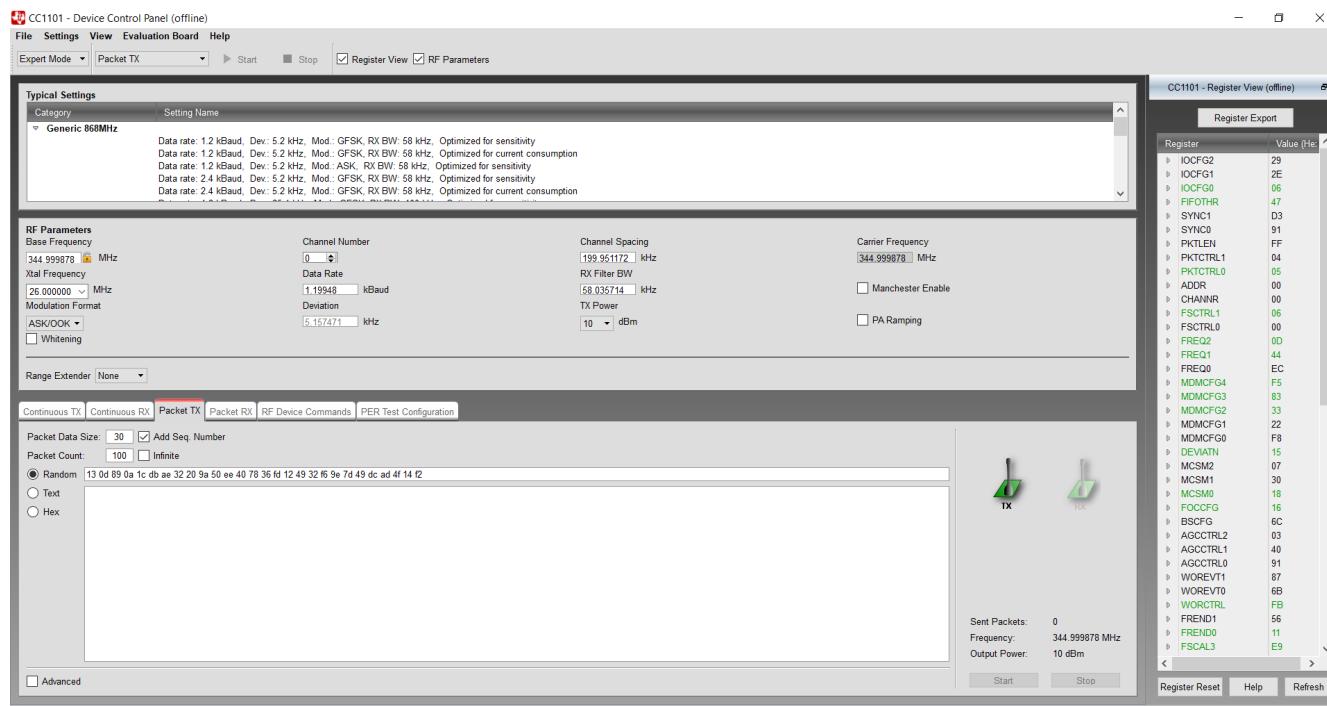
BALUN 部分 BOM 改变后的 345 MHz 波形；PA Setting = 0xC2 下 433 MHz 输出频率。



BALUN 部分 BOM 改变后的 345 MHz 波形；PA Setting = 0xC0 下 433 MHz 输出频率。



综上，PA 寄存器应该被设置为 0xC2。

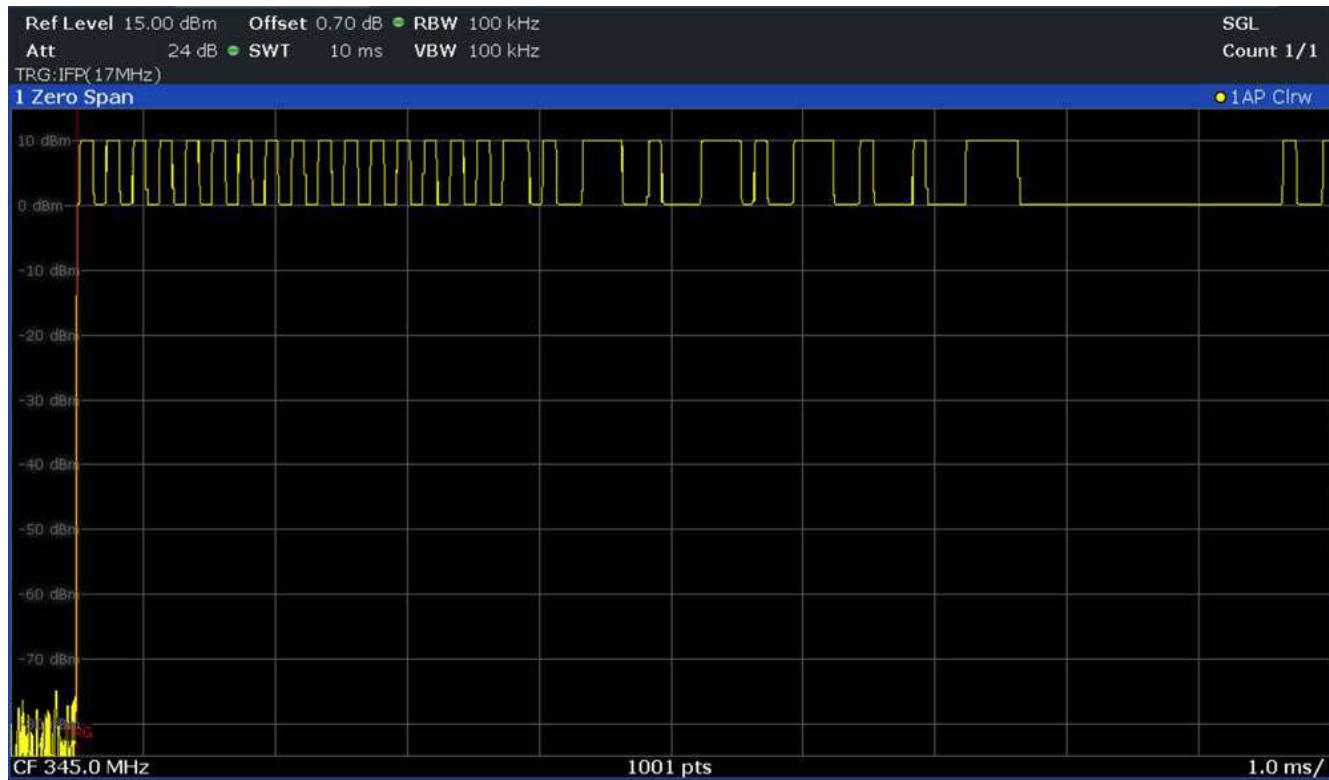


OOK 是 CC110L 支持的最简单的 ASK 调制形式。在这里，名称刚刚从 CC1101 的 ASK/OOK 更改为 CC110L 的 OOK。我们在实验室中测试将调制深度的振幅更改为 0.5。这应该只是改变 PA\_TABLE0 寄存器。将其设置为 0x51，则 0 位的输出功率为 0 dBm。为了改变 0 位 PA\_TABLE0 的功率，需要修改。默认情况下，设置为 0x00，即 PA 关闭。最终，我们在 345MHz 频段上得到了您建议的匹配条件 (OOK/1.2kbps/RXB 58kHz) 下的 -104.5dBm 的灵敏度 (1%BER)。

您可以使用 Smart RF Studio 对 CC110L 的调制进行配置，请阅读 [app note](#) 为 OOK 进行调整，在 Smart RF Studio 里默认的配置是为了 (G) FSK 并非为了 OOK。



下图是在该设计拓扑下 10kbps 的 ASK 调制的波形：



### 3 参考链接

- [CC1311R3](#)
- [CC1311P3](#)
- [CC110L](#)
- [Smart RF Studio 7](#)

## 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出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