

S2-LP 开发套件入门

引言

本文档介绍了 S2-LP 器件的开发套件以及相关硬件和软件组件。S2-LP 是超低功耗、低数据速率、低于 1 GHz 的射频收发器。

以下是基于 S2-LP 的开发套件：

1. STEVAL-FKI433V2，适用于 413-479 MHz
2. STEVAL-FKI868V2 和 X-NUCLEO-S2868A2，适用于 826-958 MHz
3. STEVAL-FKI512V1，适用于 452-527 MHz
4. STEVAL-FKI915V1，适用于 902-928 MHz，带外部 FEM
5. STEVAL-FKI001V1 双射频 BLE 和 sub-1 GHz 开发套件，带 BlueNRG-1 和 S2-LP
6. X-NUCLEO-S2915A1，适用于 915 MHz ISM 频段
7. STDES-MONARCH 参考设计，带 BlueNRG-2 和 S2-LP

1 概述

本节介绍了 S2-LP 套件的所有软件和硬件组件。

1.1 系统要求

STSW-S2LP-DK 应用程序 GUI 具有以下最低要求：

- 运行 Windows 操作系统（7、8 或 10）并采用 Intel®或 AMD®处理器的 PC
- 至少 1 GB 的空闲 RAM
- USB 端口
- 200 MB 可用硬盘空间
- Adobe Acrobat Reader 6.0 或更高版本

1.2 STSW-S2LP-DK 开发套件设置

启动 S2-LP DK-Setup-X.X.X.exe 文件，然后按照屏幕上的说明进行操作。SDK 默认安装在 C:\<user folder>\ST\S2-LP_DK x.x.x\中。

注意： 需要 IAR Embedded Workbench 8.32.1 或 MDK ARM Keil V5.26.2.0。

2 硬件说明

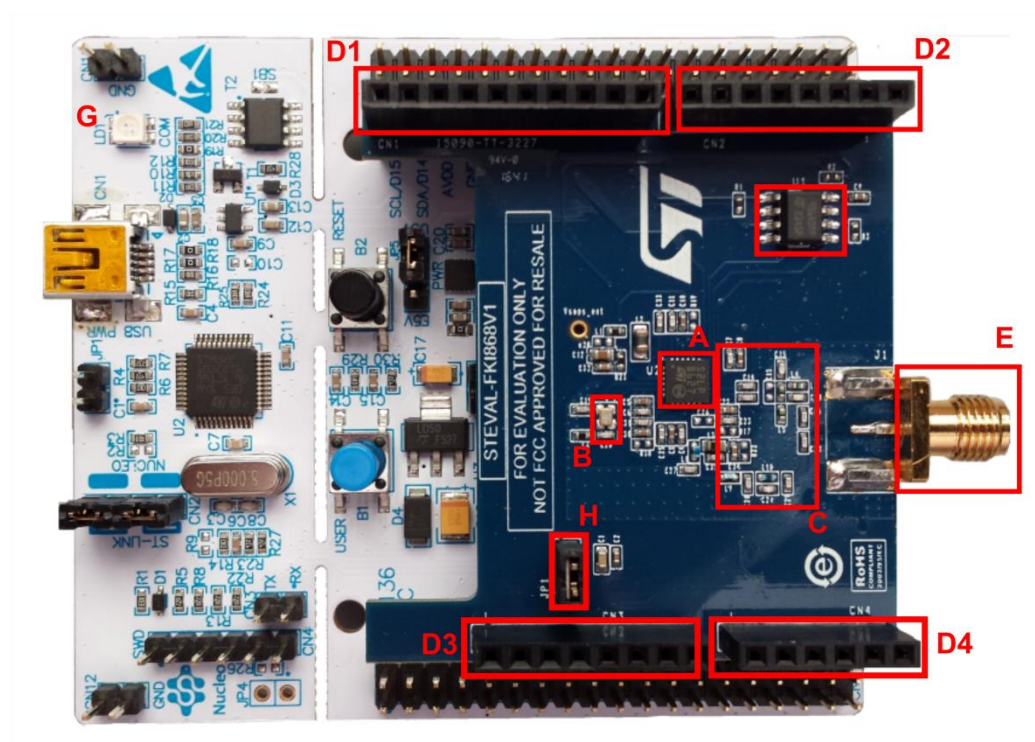
2.1 STEVAL-FKI868V2、STEVAL-FKI433V2 和 STEVAL-FKI512V1 评估板

STEVAL-FKI868V2、STEVAL-FKI433V2 和 STEVAL-FKI512V1 评估板分别设计用于 826-958 MHz、413-479 MHz 和 452-527 MHz 的工作频段。

板上的一些功能如下（参见图 1. STEVAL-FKI868V2 评估板特性）：

- S2-LP (A)
- 8 MHz 高频晶振 (B)
- 巴伦、匹配网络和谐波滤波器 (C)
- 两排兼容 Arduino 的连接器 (D1-4)
- SMA 连接器 (E)
- 用于存储生产数据的 EEPROM (F)
- NUCLEO-L152RE 或 NUCLEO-L053R8 评估板 (G)
- 用于 S2-LP 电流测量的跳线 (H)

图 1. STEVAL-FKI868V2 评估板特性



按下复位按钮后 STM32 Nucleo 开发板将复位。

2.1.1 S2-LP 连接

S2-LP 信号测试点分成两行，是兼容 Arduino 的连接器：CN1、CN3 和 CN2、CN4。

S2-LP 子板通过兼容 Arduino 的连接器的连接到 Nucleo 母板。

以下连接器和引脚名称用于 STEVAL-FKIxxxVx 原理图。

表 1. S2-LP: FKI868V2 和 FKI433V2 评估板引脚说明

引脚名称	引脚号	Arduino 连接器			
		CN1 (D1)	CN2 (D2)	CN3 (D3)	CN4 (D4)
VDD SMPS	1			引脚 4	
SMPS1	2				
SMPS2	3				
XOUT	4				
XIN	5				
SDN	6		引脚 8 (SDN)		
VDDANASYNTH	7			引脚 4	
VRSYNTH	8				
VREFVCO	9				
VDDVCOTX	10			引脚 4	
TX	11				
VRRF	12				
RXN	13				
RXP	14				
VDDRXDIG	15			引脚 4	
SDO	16	引脚 5 (MISO)			
SDI	17	引脚 4 (MOSI)			
SCLK	18		引脚 4 (SCK)		
CSN	19				引脚 2 (CS)
GPIO0	20				引脚 1 (GPIO0)
GPIO1	21				引脚 3 (GPIO1)
GPIO2	22				引脚 4 (GPIO2)
GPIO3	23				引脚 6 (GPIO3)
VSMPS3	24				
GND	25	引脚 7		引脚 6 和 7	

S2-LP 评估板包含用于 RF 匹配和巴伦的分立无源电路以及 S2-LP 正常运行所需的其他附加组件。

2.1.2 STEVAL-FKI868V2、STEVAL-FKI433V2 和 STEVAL-FKI512V1 电源

S2-LP 该板可通过 Nucleo 评估板的 mini USB 连接器供电。

安装 JP1 跳线时（图 1. STEVAL-FKI868V2 评估板特性中的 H），射频部分处于供电状态。

通过拆除此跳线并连接功率计，您可以测量 S2-LP 电流功耗。

2.2 STEVAL-FKI915V1 评估板

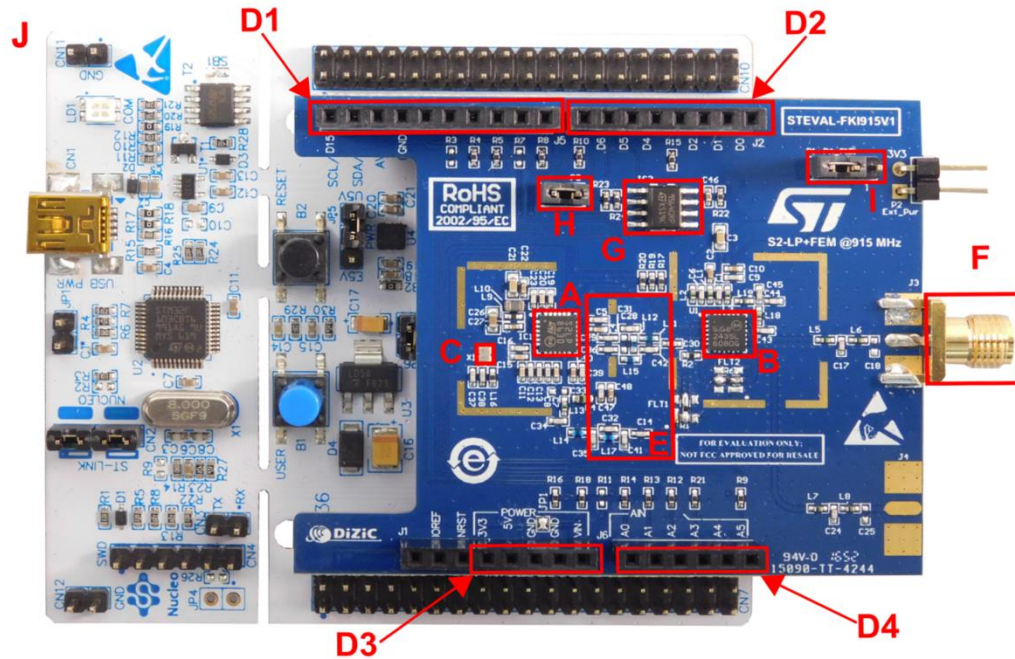
STEVAL-FKI915V1 评估板经过调整，可在 904-1055 MHz 的频段下工作。

STEVAL-FKI915V1 评估板特性（参见图 2. STEVAL-FKI915V1 评估板特性）：

- S2-LP (A)
- Skyworks SE2435L FEM (B)
- 高频 8 MHz 晶振 (C)
- 两排兼容 Arduino 的连接器 (D1-D4)

- 巴伦、匹配网络和谐波滤波器 (E)
- SMA 连接器 (F)
- 用于存储生产数据的 EEPROM (G)
- 用于 S2-LP 电流测量的跳线 (H)
- 用于 Skyworks SE2435L FEM 电流测量的跳线 (I)
- NUCLEO-L152RE 或 NUCLEO-L053R8 板 (J)

图 2. STEVAL-FKI915V1 评估板特性



2.2.1

S2-LP 连接

S2-LP 信号测试点分成两行，是兼容 Arduino 的连接器：CN1、CN3 和 CN2、CN4。

S2-LP 子板通过兼容 Arduino 的连接器连接到 STM32 Nucleo 母板。

以下连接器和引脚名称用于 STEVAL-FKI915V1 原理图。

表 2. S2-LP: FKI915Vx 评估板引脚说明

引脚名称	引脚号	Arduino 连接器			
		J5 (D1)	J2 (D2)	J1 (D3)	J6 (D4)
VDD SMPS	1			引脚 4	
SMPS1	2				
SMPS2	3				
XOUT	4				
XIN	5				
SDN	6		引脚 8 (SDN)		
VDDANASYNT	7			引脚 4	
VRSYNT	8				
VREFVCO	9				
VDDVCOTX	10			引脚 4	

引脚名称	引脚号	Arduino 连接器			
		J5 (D1)	J2 (D2)	J1 (D3)	J6 (D4)
TX	11				
VRRF	12				
RXN	13				
RXP	14				
VDDRXDIG	15			引脚 4	
SDO	16	引脚 5 (MISO)			
SDI	17	引脚 4 (MOSI)			
SCLK	18		引脚 4 (SCK)		
CSN	19				引脚 2 (CS)
GPIO0	20				引脚 1 (GPIO0)
GPIO1	21				引脚 3 (GPIO1)
GPIO2	22				引脚 4 (GPIO2)
GPIO3	23				引脚 6 (GPIO3)
VSMPS3	24				
GND	25	引脚 7		引脚 6 和 7	

2.2.2 STEVAL-FKI915V1 电源

射频前端通过两根不同的跳线提供：

- S2-LP 的 P3 (H, 图 2. STEVAL-FKI915V1 评估板特性)
- Skyworks FEM 的 P1 (I, 图 2. STEVAL-FKI915V1 评估板特性)

可通过以下方式安装 P1 跳线：

表 3. STEVAL-FKI915V1 扩展板跳线说明

P1 位置	备注
2-3	FEM 电源连接到 Nucleo 母板提供的 3.3 V 电压
1-2	FEM 通过外部电压供电，可通过 P2 连接器的引脚 2 提供该外部电压。

要测量射频部分的功耗，请将 P1 和 P3 上的两个电流相加。

2.3 X-NUCLEO-S2868A1

有关 X-NUCLEO-S2868A1 上电和 S2-LP 连接，请参阅 UM2405 “基于 S2-LP 射频收发器的 STM32 Nucleo X-NUCLEO-S2868A1 Sub-1 GHz 868 MHz RF 扩展板入门指南”，可通过 www.st.com 免费获取的。

2.4 X-NUCLEO-S2915A1

有关 X-NUCLEO-S2915A1 上电和 S2-LP 连接，请参阅 UM2641 “基于 S2-LP 射频收发器的 STM32 Nucleo X-NUCLEO-S2915A1 Sub-1 GHz 915 MHz RF 扩展板入门指南”，可通过 www.st.com 免费获取的。

2.5 STM32 Nucleo 板

2.5.1 按钮

电路板有一个按钮用于复位微控制器，另一个按钮可用于应用程序。

2.5.2 LED

具有以下可用 LED：

- LD1：绿色（STM32 Nucleo 板载 ST-LINK 电源指示灯）
- LD2：绿色（用户 LED）
- LD3：红色（微控制器电源指示灯）

2.5.3 板载 ST-LINK

ST-LINK/V2-1 编程和调试工具集成在 STM32 Nucleo 板上。

ST-LINK/V2-1 可启用 STM32 Nucleo 板 “mbed”。

2.5.4 STM32L053R8/STM32L152RE 微控制器

STM32 Nucleo 板载微控制器烧录了 S2-LP DK 固件，用于通过 GUI 或库示例驱动器件。

2.6 STEVAL-IDB007V2 和 STEVAL-IDB008V2 评估板

STEVAL-IDB007V2 和 STEVAL-IDB008V2 评估板的主要特性包括：

- 三个按钮：一个用于复位微控制器，另外两个可用于应用程序
- 三个 LED：
 - LD1：绿色（STM32 Nucleo 板载 ST-LINK 电源指示灯）
 - LD2：绿色（用户 LED）
 - LD3：蓝色（微控制器电源指示灯）
- 出色的接收器灵敏度：-88 dBm
- 极低功耗：7.7 mA RX 和 8.2 mA TX（+0 dBm）
- 新集成巴伦：BALF-NRG-02D3，它具有匹配网络和谐波滤波器
- 内置传感器：
 - 3D 数字加速度计
 - 3D 数字陀螺仪
 - 具有嵌入式温度传感器的 MEMS 压力传感器
- 电池座
- 板载 BlueNRG-1/BlueNRG-2 烧录了 S2-LP DK 固件，用于通过 GUI 或库样本来驱动设备

2.7 STEVAL-FKI001V1 评估板

STEVAL-FKI001V1 评估板的主要特性包括：

- 两个按钮：一个按钮用于复位微控制器，另一个按钮可用于应用程序
- 三个 LED：
 - LD1：绿色（STM32 Nucleo 板载 ST-LINK 电源指示灯）
 - LD2：绿色（用户 LED）
 - LD3：蓝色（微控制器电源指示灯）
- 出色的接收器灵敏度：-88 dBm
- 超低功耗：7 mA RX 和 10 mA TX（+10 dBm）
- 50 Ω 集成巴伦：BALF-NRG-01D3，具有匹配网络和谐波滤波器，BlueNRG-1 配套设备
- 板载 BlueNRG-1 烧录了 S2-LP DK 固件，用于通过 GUI 或库样本来驱动器件

2.8 硬件设置

第 1 步。 将天线连至 SMA 连接器

第 2 步. 确保子板的跳线配置正确（参见第 2.1.2 节“STEVAL- FKI868V2、STEVAL-FKI433V2 和 STEVAL-FKI512V1 电源”和第 2.2.2 节“STEVAL-FKI915V1 电源”）

第 3 步. 使用 USB 线将母板连接到 PC

3 GUI 软件说明

软件包中包含的 S2-LP DK GUI 是一个图形用户界面，可用于与 S2-LP 设备进行交互并评估其性能。

您可点击桌面上或者：开始 → STMicroelectronics → S2-LP DK X.X.X → S2-LP DK 下的 S2-LP GUI 图标，运行此实用程序

该版本的 GUI for S2-LP 专门针对 RF 性能评估，并仅提供 RF 测试窗口和设备配置参数操作。

3.1 安装

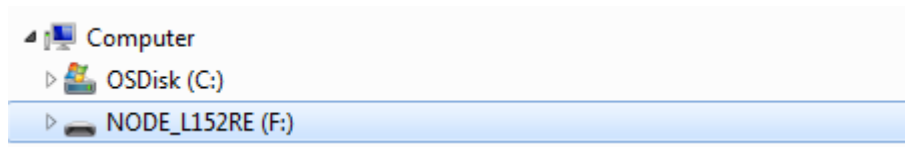
要使用 S2-LP GUI，请确保已正确设置硬件和软件（已安装 S2-LP DK）。要编程的固件位于二进制文件夹下的 S2-LP DK 软件包中。

要将固件下载到母板的内部 Flash 中，请遵循以下步骤：

- 将母板连接到 PC USB 端口
- 打开 S2-LP DK GUI
- 选择与板相关联的 COM 端口
- 打开“工具” → “固件升级”，并通过浏览选择适当的固件，然后点击“打开”，并等待固件下载。

或者，可通过将适当的固件拖放到 Windows 所识别的磁盘驱动器中（如下面的驱动器 F）来刷写母板。

图 3. NODE_L152RE 磁盘驱动器



3.2 详细说明

S2-LP DK GUI 仅可使用一个通过 USB 线连接到 PC 的 S2-LP DK - 母板以及 S2-LP RF - 子板。因此，连接到 PC 的每个板子，只能运行 S2-LP DK - GUI 的一个实例。图 4. 连接设置 1：1 台配备 S2-LP-DK GUI 的 PC 和图 5. 连接设置 2：2 台配备 S2-LP-DK GUI 的 PC 展示了使用一台或两台 PC 时的典型连接。

图 4. 连接设置 1：1 台配备 S2-LP-DK GUI 的 PC



图 5. 连接设置 2：2 台配备 S2-LP-DK GUI 的 PC

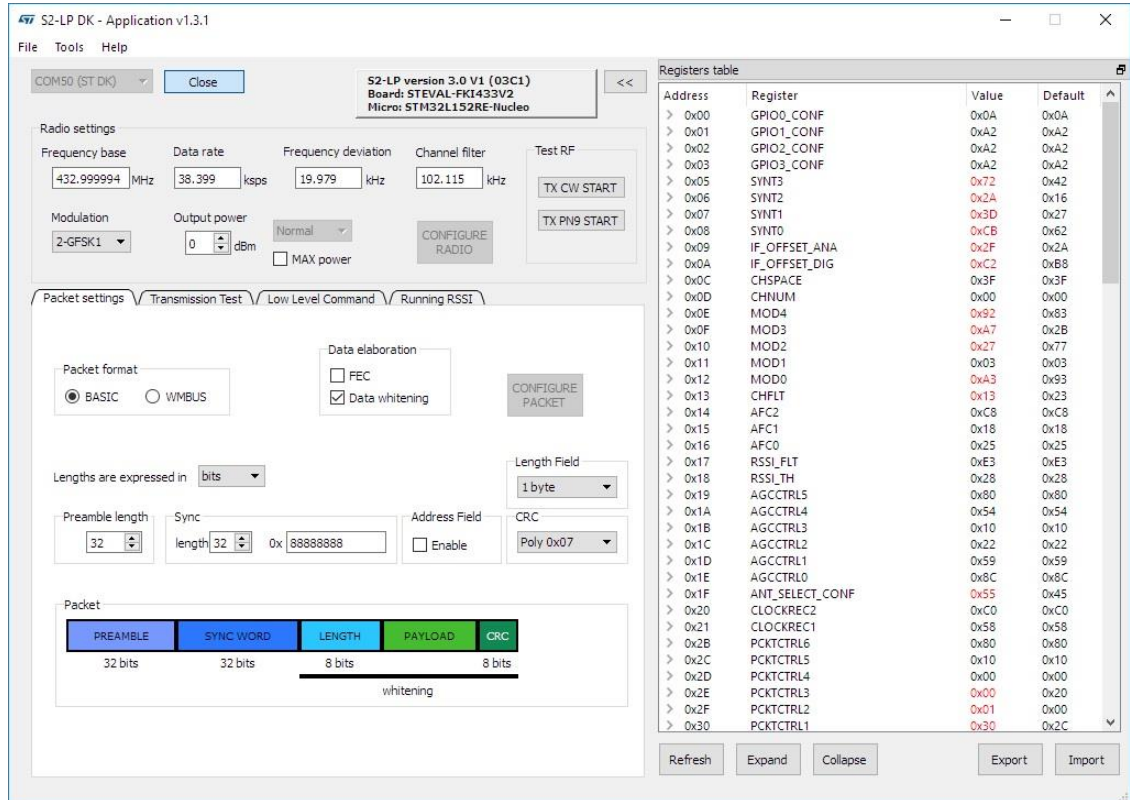


在测试期间，每个 S2-LP DK - 子板都可以充当发射器（TX）或接收器（RX）。

TX 设备在通信测试期间用作发射器；RX 设备在通信测试期间用作接收器。用户可将 S2-LP DK - 子板配置为 TX 设备或 RX 设备，并可以在运行测试前动态更改此选择。

当用户运行 S2-LP DK.exe 文件时，S2-LP DK - GUI 窗口如下所示：

图 6. S2-LP GUI 主窗口



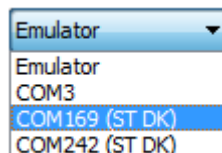
3.2.1 连接面板

在主窗口的顶部，用户可从下拉列表中选择适当的可用 COM 端口。选择正确的 S2-LP COM 端口并点击“Open（打开）”按钮后，将加载 S2-LP 的默认配置并在 S2-LP DK - GUI 上显示。

点击 COM 列表也会刷新可用的 COM 端口列表。

与 ST 开发套件关联的 COM 端口被标记为“（ST DK）”。

图 7. 可用的 COM 端口



3.2.2 射频设置面板

射频设置面板始终显示，通知用户以下内容：

- 频率；
- 调制方式；

- 数据速率；
- 频率偏差；
- 接收带宽
- 输出功率。

可根据这些限制更改这些字段（值可能会根据 XTAL 频率略有变化）：

- 频率
 - 中间频段：[413 - 479] MHz - [452 - 527] MHz
 - 高频段：[826 - 958] MHz - [904 - 1055] MHz
- 调制方式：
 - 2-FSK
 - 2-GFSK BT 0.5
 - 2-GFSK BT 1
 - 4-FSK
 - 4-GFSK BT 0.5
 - 4-GFSK BT 1
 - ASK
 - OOK
- 数据率范围：[0.3 - 250] kbps。
- 频率偏差范围：[0.793 - 761] kHz。
- 接收带宽范围：[1.1 – 769.3] kHz。
- 输出功率范围：[-30.0 14.0] dBm，如果选择正常配置（无外部 PA）（对于 STEVAL-FKI433V2、STEVAL-FKI868V2 或 X-NUCLEO-S2868A2）。
- 输出功率范围：[-5.0 28.0] dBm，如果选择 PA 配置（对于 STEVAL-FKI915V1 或 X-NUCLEO-S2915A1）。

点击“Configure radio（配置射频）”按钮，所有值均将被发送到设备，然后读取并显示。

图 8. S2-LP 射频设置

The image shows a 'Radio setting' window with the following fields and controls:

- Frequency base:** 868.000007 MHz
- Data rate:** 38.4 kbps
- Frequency deviation:** 20.027 kHz
- Channel filter:** 102.115 kHz
- Modulation:** 2-GFSK1 (dropdown menu)
- Output power:** 0 dBm (with up/down arrows)
- Normal:** (dropdown menu)
- MAX power:** (checkbox, currently unchecked)
- CONFIGURE RADIO:** (button)

3.2.3 RF 测试模式

TX CW 和 TX PN9 命令将 S2-LP 置于测试模式。

图 9. RF 测试模式按钮



两种测试都只需要将一台设备连接到 PC。

3.2.3.1 TX CW 测试

要启动该测试模式：

1. 选择所需的射频设置，然后通过点击“Configure radio（配置射频）”按钮加载这些设置；
2. 单击“TX CW START（TX CW 启动）”按钮。

在该测试模式中，S2-LP 以所选频率和所选输出功率传输载波（CW）。用户可在适当的 SMA 连接器上测量输出信号或测量 TX 状态电流功耗。

S2-LP 将保持在 TX 状态，直到点击“TX CW STOP（TX CW 停止）”按钮。

要更改频率或输出功率，请先停止运行测试，然后重复第 1 步和第 2 步，并在第 1 步中选择所需频率或输出功率。

3.2.3.2 TX PN9 测试

要启动该测试模式：

1. 选择所需的射频设置，然后通过点击“Configure radio（配置射频）”按钮加载这些设置；
2. 单击“TX PN9 START（TX PN9 启动）”按钮。

通过该测试模式，S2-LP 传输根据射频设置调制的 PN9 数据流。

用户可在适当的 SMA 连接器上测量输出信号或测量 TX 状态电流功耗。S2-LP 将保持在 TX 状态，直到点击“TX PN9 STOP（TX PN9 停止）”按钮。

要更改频率、输出功率或调制方案，请先停止运行测试，然后重复第 1 步和第 2 步，并在第 1 步中选择所需频率、输出功率或调制方案。

3.3 包设置

选择“Panel setting（面板设置）”，以查看可用的数据包配置。

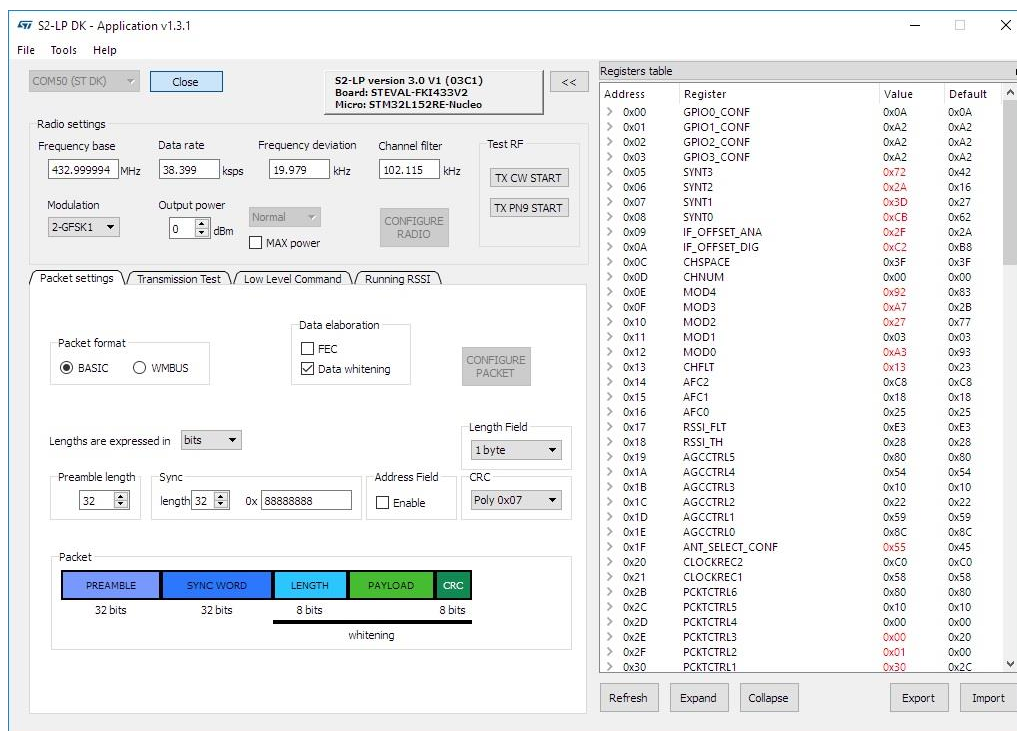
先通过按“Packet format（数据包格式）”面板中的其中一个单选按钮选择所需的数据包格式。

用户可以选择：

- BASIC
- WMBUS

每个分组格式提供不同的分组设置选项。

图 10. Basic 数据包设置面板



3.3.1

包设置：BASIC

该包的选项为（默认配置）：

- 前导码长度
- 同步码长度
- 同步码值
- CRC
- FEC
- 数据白化

可通过以下方式更改这些字段：

- 前导码长度间隔。
- 同步码长度间隔。
- CRC 可能为以下之一：
 - NO CRC。
 - Poly 0x07（1 字节）。
 - Poly 0x8005（2 字节）。
 - Poly 0x1021（2 字节）。
 - Poly 0x864CFB（3 字节）。
 - Poly 0x04C011BB7（4 字节）。

可根据所需设置选中“FEC”与“数据白化”；如果选中，则在传输过程中使用这些功能。

3.3.2 包设置：WMBUS

图 11. wM-Bus 包设置面板

如图所示，通过选择 wM-Bus，S2-LP 将某些参数用于所需的 wM-Bus 子模式。

可通过以下方式更改这些字段：

- 前导码长度：[0 - 1024] 芯片序列（01）。
- 后导码长度：[0 - 64] 芯片序列（01）。
- wM-Bus 子模式：
 - S1、S2、长头。
 - S1m、S2、T2、其他到仪表。
 - T1、T2、仪表到其他。
 - R2 短头文件。

3.4 发送测试

通过选择“发送测试”视图，用户可以访问所有可用的包测试，以便运行发送。

图 12. 发送测试面板

器件角色面板

在左下角的“Device role（器件角色）”面板中，您可以设置器件在发送过程中的主要角色（RX 和 TX）。

数据发送面板

数据发送面板具有通过发射器发送的十六进制值或字符。该字段的最大长度为 255 个字节（GUI 限制），代表发送的有效负载。如果选择 HEX 复选框，则该值必须添加为 07 08 09 0A，以此类推。如果选择 ASCII 复选框，则接受字符。

也可以通过点击“Random（随机）”生成一组随机字节。在这种情况下，随机序列长度等于通过有效负载长度字段所设置的长度。由于不确定随机生成的字符是否可以转换为 ASCII，因此始终以 HEX 格式进行表示。

RX 超时框

在 RX 超时框中，应将 RX 超时（以毫秒为单位）设置为足够大的值，以便能够接收完整的同步字（此时，超时定时器关闭）。可将其设为大于前导码持续时间与同步字长度的近似值。如果值为 0，则 RX 超时为无限，S2-LP 保持在 RX 状态，直到找到正确的同步字。

HEX 或 ASCII 单选按钮

收到的数据可以 HEX 或 ASCII 格式显示。如果设置为 ASCII 但收到非 ASCII 字符，则表达自动切换为 HEX 格式。

包长度模式

S2-LP 具有以下包长度模式：

- 变量 - 接收器从包本身检索包长度（发射器在字段中存储的信息）。要设置 VAR 模式（默认），必须将 PCKTCTRL2 寄存器的位 0 设为 1。

- 固定 - 需要接收器知道预期包的长度。要设置固定模式，必须复位 PCKTCTRL2 寄存器位，GUI 禁用 RX 选项卡上的长度字段。可通过寄存器表轻松完成该操作。如果选择固定模式，则发送测试中的 RX 选项卡将解锁有效负载长度字段，使其可由用户设置。

测试指示器面板

测试指示器面板显示有关发送/接收操作的所有结果：正确接收的数据包数量、数据包丢失、RSSI 值、自通信启动以来的 PER。

- 在右侧存在以下控件：
 - 总数据包：设置发射器将要发送的数据包数量或接收器预期接收的数据包数量（如果值为 0，则可以设置无限数量的数据包）。
 - 参考定时器：设置输入 RX 或 TX 的时段。
- 该测试定义了一个循环，其中：
 - 如果将 S2-LP 配置为发射器，它将发送数据包，然后器件进入空闲状态，直到在参考时间框中设置的时间段到期，然后重复循环。该操作的持续时间取决于数据率，并在数据包持续时间框中报告近似值。
 - 如果将 S2-LP 配置为接收器，则测试以类似方式工作：则器件在发射器进入 TX 状态的几毫秒之前进入 RX 状态，然后按写入 RX 超时框中的时间，等待数据包的同步字。如果收到数据包或 RX 超时到期，则 S2-LP 进入空闲状态（就绪），直到数据包速率框中所设置的时间段到期。在第一次通信期间，S2-LP 进入 RX 状态，以等待具有无限 RX 超时的第一个数据包（同步数据包）。
- RSSI 阈值[dBm]选项用于设置 RSSI 阈值。为实现良好的通信，请将接收器中的 RSSI 阈值设置成大于 RX 噪底。
- “Sense RSSI (RSSI 感应)” 按钮可用于根据所配置的中心频率与接收带宽来读取无线 RF 功率。如果在没有射频信号时点击“Sense RSSI (RSSI 感应)”按钮，此功能将显示环境中的 RF 噪声。

参考时间

将参考时间值设为大于数据包持续时间的值至关重要。否则，收到的数据包可能被截断或根本无法收到数据包。此外，两个器件的数据包速率必须相同。

开始/停止按钮

“Start（开始）”按钮用于运行测试，并在测试停止时变为“Stop（停止）”按钮。

低功耗模式

正常

在该模式中，微控制器将器件设为 RX 或 TX，空闲状态（当不在 RX 或 TX 态时，为 READY 态）。微控制器定时器用于实现参考定时器。当该定时器超时，MCU 将 S2-LP 设为活动模式（RX 或 TX）。

LDC

在该模式中，将 S2-LP 配置为低占空比模式。器件空闲状态为 SLEEP（睡眠）（在 RX 中，可以通过设置 PM_CONF0 寄存器中的 SLEEP_MODE_SEL 位来选择 SLEEP（睡眠）A 或 B）。

通过由内部低功耗 RC 振荡器作为时钟源的嵌入式唤醒定时器来产生唤醒事件。唤醒定时器值设置为等于 GUI 参考定时器值。

在 RX 中，使用同步功能自动重载。

监听（仅限 RX）

该选项用于配置启用快速 RX 中止和 LDC 的设备：点击启动按钮时，它将在微控制器（管理此模式的特定接收例程）中启动。

在监听模式中，务必设置适当的 RSSI 阈值，以使接收器在低功率下运行。RSSI 阈值必须高于设备所监听到的噪底（否则 RSSI 阈值将始终被置位，而使快速 RX 超时始终停止）。

Wake-Up（唤醒）定时器选值框可用于配置两个连续 RX 窗口之间的时间间隔。快速 RX 超时选值框可用于配置用于从信道监测 RSSI 的快速 RX 定时器。该定时器通过信道过滤器指数调整。此外，为将接收器设置为接收到每个数据包，必须将唤醒定时器配置为在前导码内至少唤醒设备两次。例如，可将其设置为小于前导码持续时间，可通过将发送的（以比特为单位）前导码长度除以数据速率来计算。

有关新 S2-LP 功耗 GUI 的详细信息，请参阅第 4 节 S2-LP 功耗工具。

文件名文本框

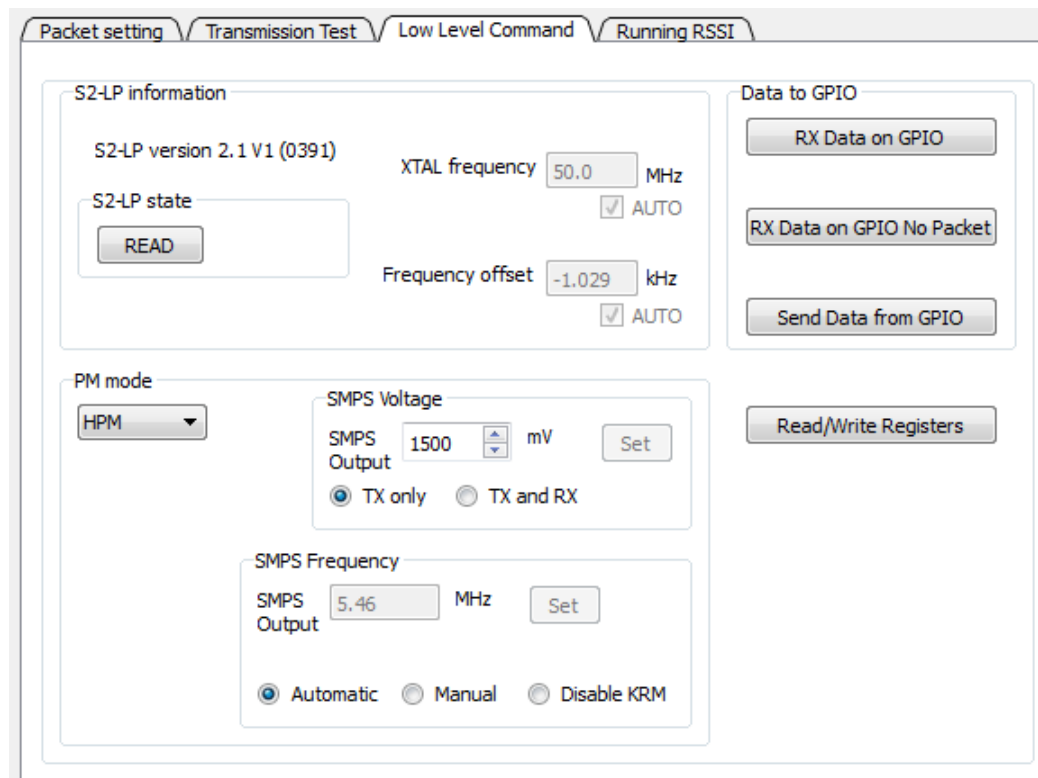
在发送测试面板底部，有一个供您在其中写入文件名的文本框，GUI 在该文件中保存当前测试的日志。

注意： 该操作在测试期间进行，所以务必在测试开始之前写入文件名。

3.5 底层命令

通过选择“Low Level Command（底层命令）”，您可以访问 S2-LP 测试模式，读取状态并设置 SMPS 输出电压。

图 13. 底层命令面板



可通过点击 S2-LP 状态读取按钮读取 S2-LP 的状态。也会显示芯片版本。

可通过 XTAL 频率文本框获取 XTAL 频率。

测试模式

可通过三个按钮设置一些特定测试模式：

- RX data in GPIO：配置 S2-LP GPIO_0 与 GPIO_1，分别发送收到的 RX 数据和时钟信号。这样一来，当 S2-LP 进入 RX 状态时，就能查看收到的数据包。
- RX data in GPIO No Packet：相当于 RX data in GPIO，唯一的区别在于旁路了 S2-LP 内部的载荷包处理器。

- send data from GPIO: 配置 S2-LP GPIO_0 与 GPIO_1，分别发送要传输的数据以及用于对数据进行采样的时钟。这样一来，当 S2-LP 进入 TX 状态时，便可以通过 GPIO 发送（而非通过 FIFO）加载的数据。

SMPS 电压

您可以在 SMPS 输出框中设置 SMPS 电压（以 mV 为单位）。

SMPS 频率

可设置 SMPS 频率（以 MHz 为单位）。

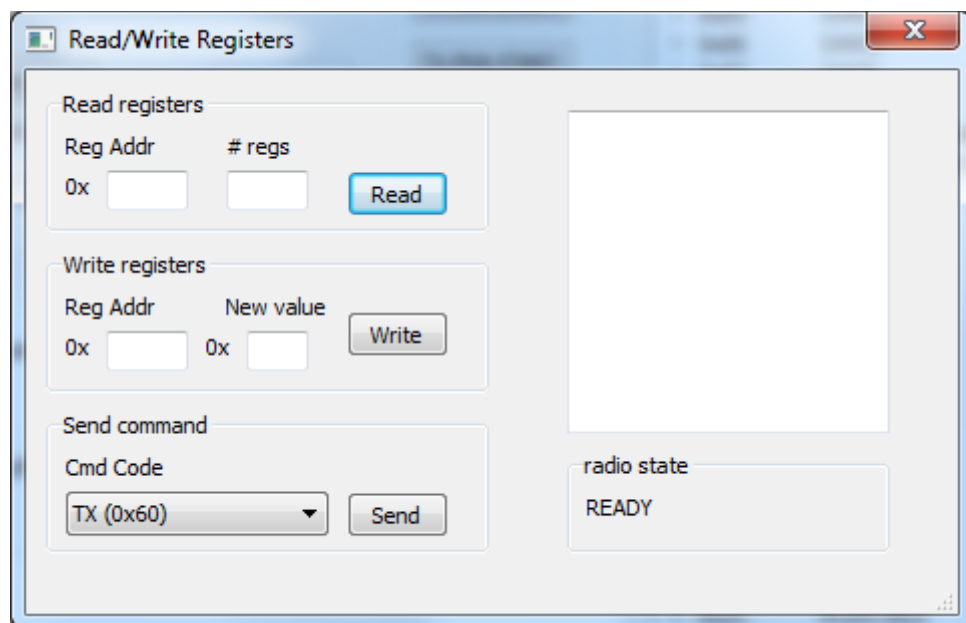
下面的选项可供使用：

- auto（自动）：根据器件状态（TX 或 RX）设置最佳频率；
- manual（手动）：用户可以指定 SMPS 频率；
- disable KRM（禁用 KRM）：SMPS 频率为固定值 $F_{dig}/4$ 。

读/写寄存器

尽管您可以通过寄存器表写入最常用的寄存器，但读/写寄存器按钮允许写入所有 S2-LP 寄存器。点击该按钮时，将显示以下窗口：

图 14. 读/写寄存器窗口

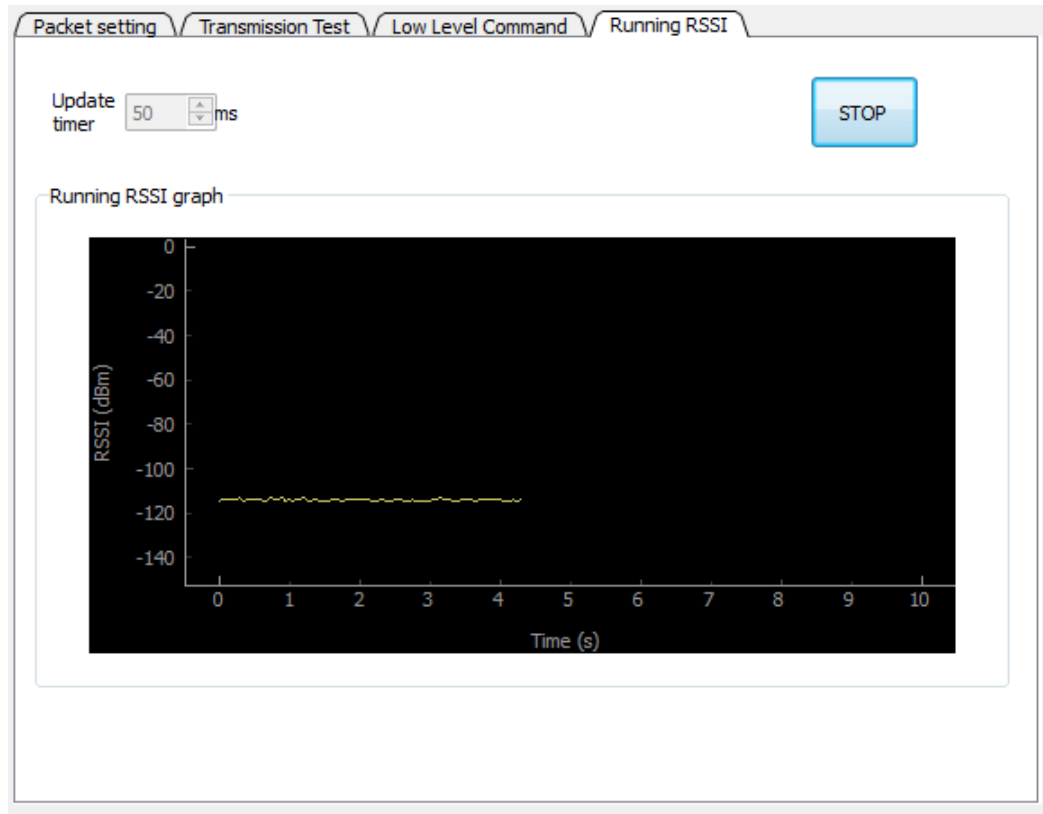


- 通过“Read registers（读寄存器）”框，您可以指定起始地址，以及要从起始地址开始读取的寄存器数量。
- 通过“Write registers（写寄存器）”框，可以指定单个寄存器的地址和值。

3.6 Running RSSI

“Running RSSI”使用户可通过使用 running RSSI 功能的 S2-LP 来测量信道上的功率。对 RSSI 值进行采样并绘制成图形（RSSI (dBm)对比时间(s)）。可通过更新定时器选值框来设置轮询间隔。

图 15. “Running RSSI” 选项卡



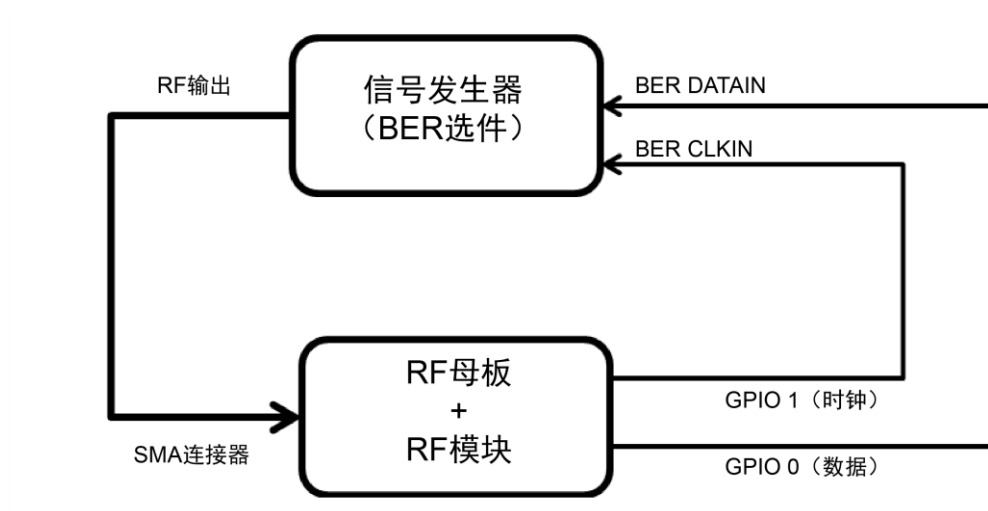
3.7 如何使用信号发生器来运行 BER 测试

借助“Low Level Command (底层命令)”选项卡，您可以将 S2-LP 置于 GPIO 口直接模式；从而使载荷包处理器完全被旁路，并使解调的数据以及相关时钟信号在两个 GPIO 上输出。

可通过“RX Data on GPIO No Packet”按钮来启用该模式。然后，这两个信号可用于具有 BER 选件的信号发生器，以允许根据特定的射频配置测量比特误码率（参见图 16. BER 测试平台架构）。

必须在时钟信号下降沿对数据进行采样。

图 16. BER 测试平台架构



3.8 寄存器表

在 GUI 右侧，默认显示寄存器表（可通过“<<”按钮隐藏/显示）。
寄存器表提供了一种快速且人性化的方式修改设备寄存器和位字段。

图 17. 寄存器表

Address	Register	Value	Default
▷ 0x07	SYNT1	0x27	0x27
▷ 0x08	SYNT0	0x62	0x62
▷ 0x09	IF_OFFSET_ANA	0x29	0x2A
▷ 0x0A	IF_OFFSET_DIG	0xB7	0xB8
▷ 0x0C	CHSPACE	0x3F	0x3F
▷ 0x0D	CHNUM	0x00	0x00
▷ 0x0E	MOD4	0x83	0x83
▷ 0x0F	MOD3	0x2B	0x2B
▷ 0x10	MOD2	0x27	0x77
▷ 0x11	MOD1	0x03	0x03
▷ 0x12	MOD0	0x93	0x93
▷ 0x13	CHFLT	0x23	0x23
▷ 0x14	AFC2	0xC8	0xC8
▷ 0x15	AFC1	0x18	0x18
▷ 0x16	AFC0	0x25	0x25
▷ 0x17	RSSI_FLT	0xE3	0xE3
▷ 0x18	RSSI_TH	0x28	0x28
▷ 0x1A	AGCTRL4	0x75	0x75
▷ 0x1B	AGCTRL3	0x00	0x00
▷ 0x1C	AGCTRL2	0x22	0x22
▷ 0x1D	AGCTRL1	0x7B	0x7B
▷ 0x1E	AGCTRL0	0x8A	0x8A
▷ 0x1F	ANT_SELECT_CONF	0x55	0x45
▷ 0x20	CLOCKREC2	0x00	0xC0
▷ 0x21	CLOCKREC1	0x58	0x58
▷ 0x2B	PCKTCTRL6	0x80	0x80
▷ 0x2C	PCKTCTRL5	0x20	0x10
▷ 0x2D	PCKTCTRL4	0x00	0x00
▷ 0x2E	PCKTCTRL3	0x00	0x20
▲ 0x2F	PCKTCTRL2	0x01	0x00
7:6	RESERVED	0x00	0x00
5	FCS_TYPE_4G	0x00	0x00
4	FEC_TYPE_4G/STOP_BIT	0x00	0x00
3	INT_EN_4G/START_BIT	0x00	0x00
2	MBUS_3OF6_EN	0x00	0x00
1	MANCHESTER_EN	0x00	0x00
0	FIX_VAR_LEN	0x01	0x00
▷ 0x30	PCKTCTRL1	0x30	0x2C

Refresh
Expand
Collapse
Export
Import

通过单击每个条目左侧的白色箭头，可以展开或压缩单个寄存器，以显示其逻辑字段。

修改字段时，相应的寄存器将自动写入 S2-LP。

此外，如果寄存器修改射频部分或数据包的参数，则用新字段值更新相应选项卡。通过双击寄存器表中的条目，也可以获得寄存器的详细描述。

选项卡底部有五个按钮：

- Refresh（刷新）：从设备读取所有寄存器值，并将其更新到选项卡中。
- Expand（展开）：展开所有寄存器的位字段。
- Collapse（折叠）：折叠所有位字段。
- Export（导出）：将寄存器配置保存到用户选择的文件中。
- Import（导入）：从用户选择的文件加载寄存器。可以 XML 和 txt 格式加载文件。

3.9 菜单栏

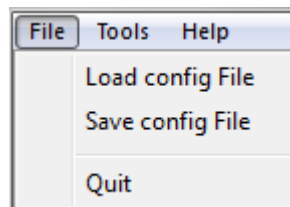
GUI 有一个可导出不同功能的菜单栏：

- 文件
- 工具包括：
 - 固件升级
 - 固件版本
 - Export code configuration（导出代码配置）
 - export production info（导出生产信息）
- 帮助

3.9.1 文件

文件菜单提供了以下列表：

图 18. 文件列表



3.9.1.1 保存并加载高级配置

保存选项允许将当前的射频和数据包配置保存在文件中，以便您可以轻松地重载这些配置。加载选项允许从文件加载存储的射频和数据包配置。

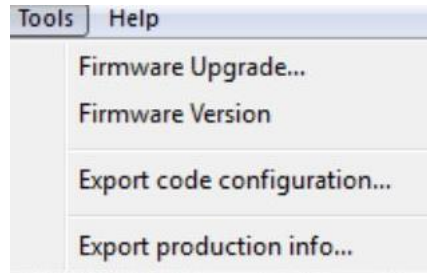
3.9.1.2 保存并加载寄存器配置

可通过寄存器表下方的导出和导入按钮获得这些功能。

3.9.2 工具

“工具”菜单提供了以下列表：

图 19. 工具列表



3.9.2.1 固件升级

S2-LP 固件允许通过 USB 端口执行自动固件升级。

升级固件的步骤如下：

1. 启动 S2-LP DK GUI。
2. 选择要升级的母板的 COM 端口。
3. 从 “Tools（工具）” → “Firmware Upgrade（固件升级）” 选择要加载的固件（.bin 或 .hex 格式）。
4. 单击 “OK（确定）”，固件将被烧录到板中。

S2-LP DK Binaries 目录包含用于运行 GUI S2LP_CLI_NUCLEO 的固件。

或者，也可以直接将 S2LP_CLI_NUCLEO 固件复制到与要烧录的母板相对应的硬盘驱动器中。

如果 GUI 发现固件不一致，则会发出警告并提示用户升级固件。

3.9.2.2 固件版本

“固件版本” 显示了运行在微控制器上的当前固件版本。

固件版本格式为 x.y.z，并具有用于识别 beta 版本的 BETA 选项，以及用于识别 alpha 版本的 ALPHA 选项。

注意： beta 版早于最终版，并与最终版具有相同的版本号，即 2.0.0_BETA 略早于 2.0.0 版。

3.9.2.3 Export code configuration（导出代码配置）

此选项将生成 C 语言指令列表，用于向 S2-LP 寄存器写入新值。

例如，用户可以使用 GUI 快速找到设备所需的配置，然后使用该工具获得一个 C 语言代码片段，该代码片段可以轻松包含在微控制器上运行的程序中。

以下是通过默认配置获得的 C 文件的示例：

```
#include <stdint.h>
/**
 * The SPI interface is platform dependent, this means that it should be implemented according
 * to the used hardware.
 * The function S2LPSpiWriteRegisters(uint8_t address, uint8_t n_regs, uint8_t* buffer) must be
 * implemented.
 * An example of implementation (for the SDK_EVAL motherboards of the S2-LP kit) can be found
 * into the file: Projects/Drivers/BSP/SDK_Eval_STM32L/Src/SDK_EVAL_Spi_Driver.c
 * It is advisable to implement also the read registers, the command strobe and the 2 FIFO
 * functions to read and write (no reference in this template code but needed in applications
 * using the FIFO of the S2-LP).
 */
/* list of the command codes of S2-LP */
#define COMMAND_TX ((uint8_t)(0x60)) /*!< Start to transmit; valid only from READY */
#define COMMAND_RX ((uint8_t)(0x61)) /*!< Start to receive; valid only from READY */
#define COMMAND_READY ((uint8_t)(0x62)) /*!< Go to READY; valid only from STANDBY or SLEEP
or LOCK */
#define COMMAND_STANDBY ((uint8_t)(0x63)) /*!< Go to STANDBY; valid only from READY */
#define COMMAND_SLEEP ((uint8_t)(0x64)) /*!< Go to SLEEP; valid only from READY */
#define COMMAND_LOCKRX ((uint8_t)(0x65)) /*!< Go to LOCK state by using the RX configuration
of the synth; valid only from READY */
#define COMMAND_LOCKTX ((uint8_t)(0x66)) /*!< Go to LOCK state by using the TX configuration
of the synth; valid only from READY */
#define COMMAND_SABORT ((uint8_t)(0x67)) /*!< Force exit form TX or RX states and go to
READY state; valid only from TX or RX */
#define COMMAND_SRES ((uint8_t)(0x70)) /*!< Reset of all digital part, except SPI registers
*/
```

```
#define COMMAND_FLUSHRXFIFO ((uint8_t)(0x71)) /*!< Clean the RX FIFO; valid from all states
*/
#define COMMAND_FLUSHTXFIFO ((uint8_t)(0x72)) /*!< Clean the TX FIFO; valid from all states
*/
/* This is the function that initializes the S2-LP with the configuration that the user has
exported using the GUI */
void SpiritBaseConfiguration(void)
{
uint8_t tmp[5];
tmp[0]= 0x92; /* reg. GPIO0_CONF (0x00) */
tmp[1]= 0x52; /* reg. GPIO1_CONF (0x01) */
tmp[2]= 0x2A; /* reg. GPIO2_CONF (0x02) */
S2LPSpiWriteRegisters(0x00, 3, tmp);
tmp[0]= 0x2B; /* reg. SYNT2 (0x06) */
tmp[1]= 0x85; /* reg. SYNT1 (0x07) */
tmp[2]= 0x1F; /* reg. SYNT0 (0x08) */
tmp[3]= 0x2F; /* reg. IF_OFFSET_ANA (0x09) */
tmp[4]= 0xC2; /* reg. IF_OFFSET_DIG (0x0A) */
S2LPSpiWriteRegisters(0x06, 5, tmp);
tmp[0]= 0x92; /* reg. MOD4 (0x0E) */
tmp[1]= 0xA7; /* reg. MOD3 (0x0F) */
tmp[2]= 0x27; /* reg. MOD2 (0x10) */
S2LPSpiWriteRegisters(0x0E, 3, tmp);
tmp[0]= 0xA3; /* reg. MOD0 (0x12) */
tmp[1]= 0x13; /* reg. CHFLT (0x13) */
S2LPSpiWriteRegisters(0x12, 2, tmp);
tmp[0]= 0x55; /* reg. ANT_SELECT_CONF (0x1F) */
tmp[1]= 0x00; /* reg. CLOCKREC2 (0x20) */
S2LPSpiWriteRegisters(0x1F, 2, tmp);
tmp[0]= 0x20; /* reg. PCKTCTRL5 (0x2C) */
S2LPSpiWriteRegisters(0x2C, 1, tmp);
tmp[0]= 0x00; /* reg. PCKTCTRL3 (0x2E) */
tmp[1]= 0x01; /* reg. PCKTCTRL2 (0x2F) */
tmp[2]= 0x30; /* reg. PCKTCTRL1 (0x30) */
S2LPSpiWriteRegisters(0x2E, 3, tmp);
tmp[0]= 0x01; /* reg. PROTOCOL1 (0x3A) */
S2LPSpiWriteRegisters(0x3A, 1, tmp);
tmp[0]= 0x41; /* reg. PCKT_FLT_OPTIONS (0x40) */
S2LPSpiWriteRegisters(0x40, 1, tmp);
tmp[0]= 0x00; /* reg. FAST_RX_TIMER (0x54) */
S2LPSpiWriteRegisters(0x54, 1, tmp);
tmp[0]= 0x1D; /* reg. PA_POWER8 (0x5A) */
S2LPSpiWriteRegisters(0x5A, 1, tmp);
tmp[0]= 0x07; /* reg. PA_POWER0 (0x62) */
tmp[1]= 0x01; /* reg. PA_CONFIG1 (0x63) */
S2LPSpiWriteRegisters(0x62, 2, tmp);
}
```

根据应用程序，可手动将其修改为宏或简单指令块。

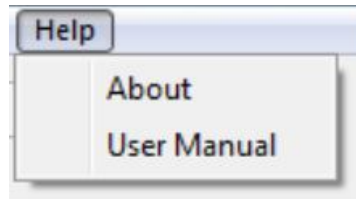
3.9.2.4 Export production info (导出生产信息)

每个 STEVAL-FKI 板均配有 E2PROM，在制造阶段向其中存储了一些数据，可通过点击“Export production info (导出生产信息)”将这些数据导出到文件中。

3.9.3 帮助

您可以通过“Help (帮助)”按钮访问用户手册。

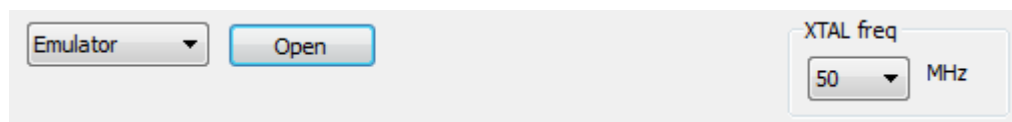
图 20. Help 菜单



3.9.4 设备模拟器

在未将任何板连接到 PC 的情况下，可通过“Emulator（模拟器）”按钮模拟设备：选中“Emulator（模拟器）”之后，用户可以使用 GUI，就像一块板已连接到 PC 一样。

图 21. S2-LP 仿真器



由于没有实际的电路板，用户应使用仅在该情况下激活的选项卡手动指定 XTAL 频率（否则通过微控制器自动计算）。

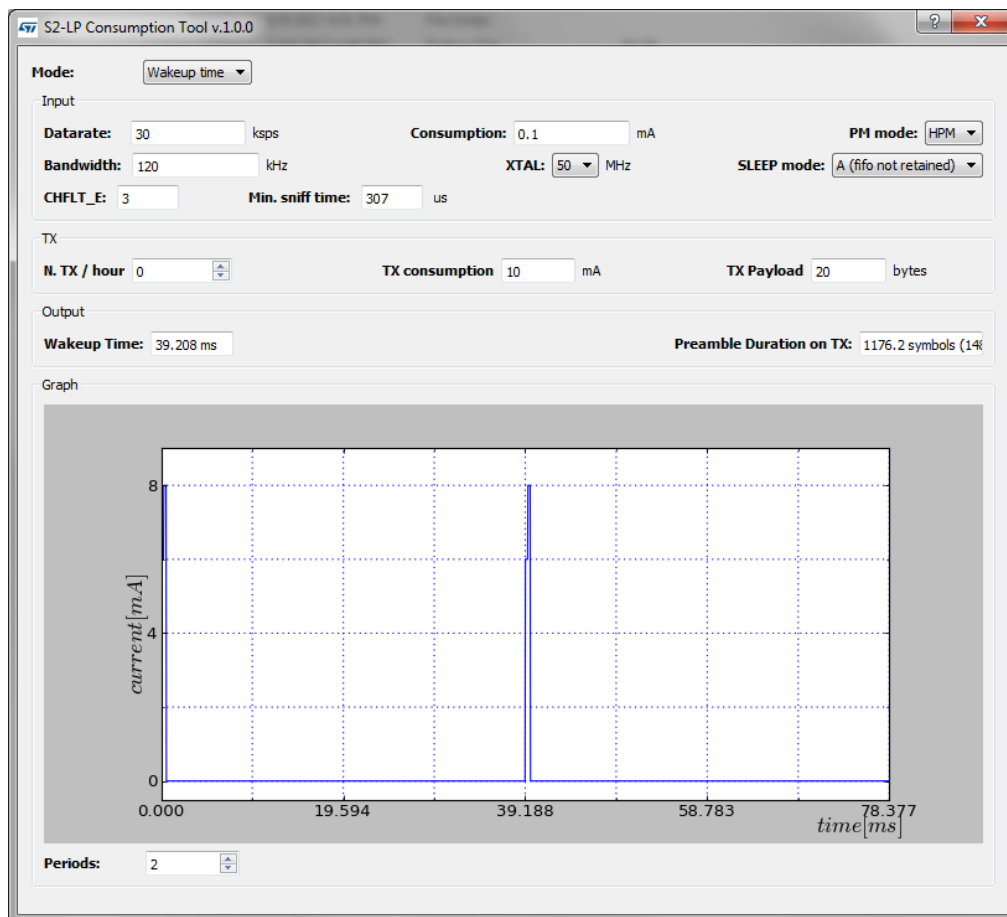
随后，单击“Open（打开）”按钮，与设备配置相关的所有一切均应像已连接设备那样运行。

因而，用户可以轻松地选择自己的配置，并查看或保存在其固件中保持相同配置所需的寄存器值。为此，可以将该功能与“Export code configuration（导出代码配置）”工具配合使用。

4 S2-LP 电流评估工具

S2-LP 电流评估工具是一个图形用户界面，用于模拟监听模式下的 S2-LP 电流功耗。由于它只是模拟，因此不能认为它可以替代提供实际数据的芯片测量值。

图 22. S2-LP 电流评估工具 - GUI 概述



该 GUI 可以设备两种模式：

- 唤醒
- 电流功耗

4.1 唤醒模式

在唤醒模式中，GUI 的作用是计算达到所需功耗所需的唤醒时间。输入参数为：

- 所需的通信数据速率
- 目标功耗
- 接收带宽
- XTAL 频率
- PM 模式 (HPM/LPM)
- 睡眠模式 (A 或 B)

图 23. 唤醒模式输入

Input		Consumption: <input type="text"/> mA		PM mode: <input type="text" value="HPM"/>
Datarate: <input type="text"/> kbps	Bandwidth: <input type="text"/> kHz	XTAL: <input type="text" value="52"/> MHz	SLEEP mode: <input type="text" value="A (fifo not retained)"/>	
CHFLT_E: <input type="text"/>	Min. sniff time: <input type="text"/> us			

输出为唤醒时间以及 TX 上用于确保每个数据包均已正确接收的前导码最短长度。

图 24. 唤醒模式输出

Output	
Wakeup Time: <input type="text"/>	Preamble Duration on TX: <input type="text"/>

4.2 功耗模式

在功耗模式中，GUI 将给定唤醒时间的功耗计算为输入数据。

输入参数为：

- 所需的通信数据速率
- 唤醒时间
- 接收带宽
- XTAL 频率
- PM 模式（HPM/LPM）
- 睡眠模式（A 或 B）

图 25. 功耗模式输入

Input		Wakeup Time: <input type="text"/> ms		PM mode: <input type="text" value="HPM"/>
Datarate: <input type="text"/> kbps	Bandwidth: <input type="text"/> kHz	XTAL: <input type="text" value="52"/> MHz	SLEEP mode: <input type="text" value="A (fifo not retained)"/>	
CHFLT_E: <input type="text"/>	Min. sniff time: <input type="text"/> us			

输出为电流功耗与电池持续指示。

图 26. 功耗模式输出

Output	
Wakeup Time: <input type="text"/>	Preamble Duration on TX: <input type="text"/>

对于这两种模式，一旦填入所有输入字段，GUI 就会执行计算。

4.3 TX

GUI 还考虑了由于包传输而导致的功耗。

从 TX 部分，可以指定每小时的传输次数，TX 电流和 TX 有效负载（以字节为单位）。

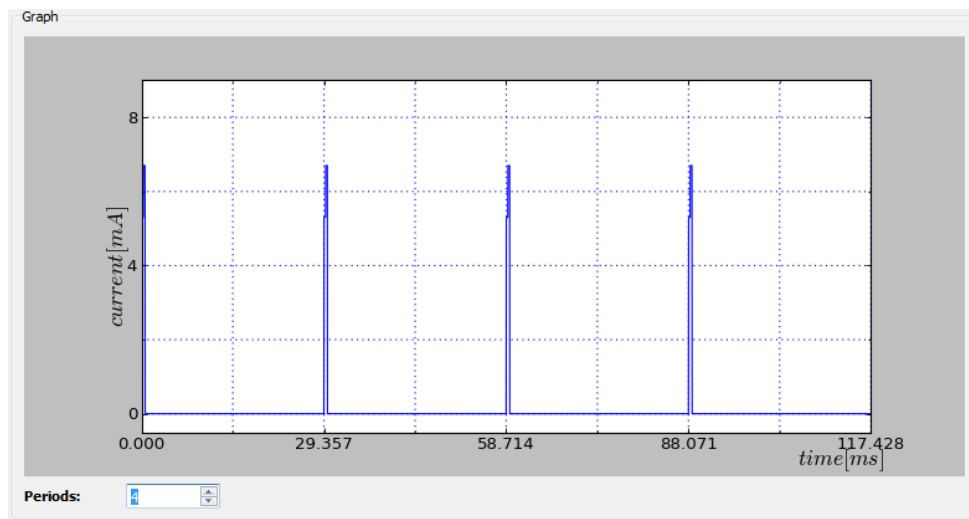
图 27. TX 部分

TX		
N. TX / hour <input type="text" value="0"/>	TX consumption <input type="text"/> mA	TX Payload <input type="text"/> bytes

4.4 当前配置文件图

下图显示了 RX 唤醒周期。

图 28. 电流图



5 固件示例

固件包中可用的所有固件示例均已经过配置，可与支持的平台一起工作。对于定制产品，用户仅需修改适当的配置文件，即可匹配其配置。

STM32 Nucleo

默认情况下，文件位于 C:\<user folder>\ST\S2-LP_DK
x.x.x\Projects\Drivers\S2LP_Middleware\STM32\inc.

表 4. STM32 Nucleo 开发板配置文件

需要编辑的文件	板 (MCU)
S2LP_Nucleo64_STM32_L0xx_AUTO.h	NUCLEO-L053R8 (STM32L0)
S2LP_Nucleo64_STM32_L1xx_AUTO.h	NUCLEO-L152RE (STM32L1)

BlueNRG

默认情况下，文件位于 C:\<user folder>\ST\S2-LP_DK
x.x.x\Projects\Drivers\S2LP_Middleware\BlueNRG\inc.

表 5. BlueNRG 评估板配置文件

需要编辑的文件	板 (BlueNRG)
S2LP_FKI001V1.h	STEVAL-FKI001V1 (BlueNRG-1)
S2LP_IDB00xV2_AUTO.h	STEVAL-IDB00xV2 (BlueNRG-1/2)

需要配置的参数

文件包含 S2-LP 与 MCU 之间的配置接口 (SPI、GPIO、SDN)，以及与 S2-LP 强相关的其他参数，如下表所示。

用户应根据配置设置预处理器定义，并在需要时重新映射连接。

表 6. 需要配置的参数列表

参数	值
EEPROM_PRESENT	EEPROM_YES
	EEPROM_NO
S2LP_FEM_PRESENT	S2LP_FEM_YES
	S2LP_FEM_NO
	S2LP_FEM_AUTO
TCXO_PRESENT	TCXO_YES
	TCXO_NO
	TCXO_AUTO

用户平台配置

代替重写默认头文件，用户可以使用默认文件作为模板并设置 S2LP_Middleware_Config.h 中所述的 USER_EVAL_PLATFORM 预处理器定义，从而创建自定义文件（例如，USER_Platform_Configuration.h），（默认位于 C:\<user folder>\ST\S2-LP_DK x.x.x\Projects\Drivers\S2LP_Middleware\STM32(BlueNRG)\inc.）

5.1 命令行接口（CLI）

此 S2-LP CLI 演示固件展示了如何使用命令行界面将命令从不同的 MCU 发送至 S2-LP。GUI 使用相同的固件来驱动 S2-LP 和执行评估程序。

要使用无 GUI 的 CLI，请遵循如下步骤。

第 1 步。 打开波特率为 115200 8-N-1 的 COM 端口。
打开一个简单的命令 shell。

第 2 步。 输入“help”，以读取完整的命令列表。

第 3 步。 键入所需命令名称，后跟参数（如有必要）。

5.1.1 CLI 指令、调制类型和 S2-LP 模式

表 7. CLI 指令

指令	参数	说明
S2LPRadioInit	modulation_select : 调制类型 ⁽¹⁾ data_rate : 数据速率 (sym/s) fdev: 频率偏移 (Hz) rx_bandwidth: 接收带宽 (Hz) xtal_frequency: 建议值为 0 (在这种情况下, 使用预设的 XTAL 值)	初始化 S2-LP 射频
S2LPRadioGetInfo	无	获取射频参数
S2LPRadioGetXtalFrequency	无	获取晶振频率 (Hz)
S2LPRadioSetFrequencyBase	frequency_base : 基频 (Hz)。	初始化 S2-LP 射频基频
S2LPRadioGetFrequencyBase	无	获取射频基频 (Hz)
S2LPRadioSetModulation	modulation_scheme: 调制类型 ⁽¹⁾	初始化 S2-LP 调制方式
S2LPRadioGetModulation	无	获取 S2-LP 调制方式
S2LPRadioSetPALeveldBm	pa_level: 指示 dBm 值 pa_index: PA 寄存器索引 (0 到 7)	设置 S2-LP PA 功率等级, 将传递的索引设为最大索引
S2LPRadioGetPALeveldBm	pa_index: PA 寄存器索引 (0 到 7)	获取 S2-LP PA 等级 (dBm)
S2LPRadioSetPALevelNdBm	pa_level: 指示 dBm 值 pa_index: PA 寄存器索引 (0 到 7)	设置 S2-LP PA 等级
S2LPRadioSetPALevelMaxInd	pa_index: PA 寄存器索引 (0 到 7)	设置 S2-LP PA 最大索引
S2LPGetPktFrmt	无	返回数据包格式
S2LPPktBasicInit	preamble_length : 前导码长度 (位)	初始化 Basic 数据包格式

指令	参数	说明
	sync_length: 同步字长度 (位) sync_word: 同步字 fix_var_length: (0: 固定, 1: 可变) extended_length: 仅适于可变长度 (0: 1 字节地址, 1: 2 字节地址) crc_mode ⁽²⁾ address: 指定地址是否必须使用 (1) 或不使用 (0) fec: 指定 FEC 是否必须使用 (1) 或不使用 (0) whitening: 指定白化是否必须使用 (1) 或不使用 (0)	
S2LPPktBasicGetInfo	无	返回 Basic 数据包信息
S2LPPktStackInit	preamble_length: 前导码长度 (位) sync_length: 同步字长度 (位) sync_word: 同步字 fix_var_length: (0: 固定, 1: 可变) extended_length: 仅适于可变长度 (0: 1 字节地址, 1: 2 字节地址) crc_mode ⁽²⁾ fec: 指定 FEC 是否必须使用 (1) 或不使用 (0) whitening: 指定白化是否必须使用 (1) 或不使用 (0)	初始化 STack 数据包格式
S2LPPktStackGetInfo	无	返回 STack 数据包信息
S2LPPktBasicSetPayloadLength	payload_len: 要设置的有效负载长度	设置 Basic 数据包的有效负载长度
S2LPPktBasicGetPayloadLength	无	获取 Basic 数据包的有效负载长度
S2LPPktStackSetPayloadLength	payload_len: 要设置的有效负载长度	设置 STack 数据包的有效负载长度
S2LPPktStackGetPayloadLength	无	获取 STack 数据包的有效负载长度
S2LPPktMbusSetPayloadLength	payload_len: 要设置的有效负载长度	设置 wM-Bus 数据包的有效负载长度
S2LPPktMbusGetPayloadLength	无	获取 STack 数据包的有效负载长度
S2LPPktMbusInit	wmbus_submode ⁽³⁾	初始化 wM-Bus 数据包格式

指令	参数	说明
	preamble_length: 前导码长度 (码片) postamble_length: 前导码长度 (码片)	
S2LPPktMbusGetInfo	无	返回 wM-Bus 数据包信息
S2LPTimerSetRxTimeoutUs	rx_timeout: RX 超时 (μs)	设置 RX 超时
S2LPTimerGetRxTimeout	无	获取 RX 超时
S2LPGpioInit	gpio_pin: 0:GPIO0, 1:GPIO1, 2:GPIO2, 3:GPIO3 gpio_mode ⁽⁴⁾ gpio_io function ⁽⁵⁾	设置 S2-LP GPIO
S2LPIrq	irq 码 ⁽⁶⁾ en: 1: 使能; 0: 禁用	设置 S2-LP GPIO
S2LPIrqGetStatus	无	获取 S2-LP IRQ 状态
S2LPQiGetRssidBm	无	获取 RSSI dBm 值
S2LPGetVersion	无	获取 S2-LP 版本
S2LPWhitening	白化: 1: 使能; 0: 禁用	设置 S2-LP 白化
S2LPDirectRfSetTxMode	tx_mode ⁽⁷⁾	设置 S2-LP DirectRf 模式 (TX)
S2LPDirectRfSetRxMode	rx_mode ⁽⁸⁾	设置 S2-LP DirectRf 模式 (RX)
S2LPLinearFifoSetAlmostFullThresholdRx	ae_thr: FIFO 接近全满阈值	设置 RX 接近全满阈值
S2LPLinearFifoSetAlmostEmptyThresholdTx	ae_thr: FIFO 几乎已空阈值	设置 TX 几乎已空阈值
S2LPTimerSetWakeUpTimerUs	wake_up: 定时器 (us)	设置唤醒定时器
S2LPTimerLdcrMode	en: 1: 使能; 0: 禁用 en_autoreload: 1: 使能; 0: 禁用	启用 LDCR 模式
S2LPCalibrateRco	无	校准 RCO
S2LPGetNBytesReportAll	en: 1: 使能; 0: 禁用	报告所有数据包数据
S2LPGetNBytes	n_bytes: 要接收的字节数	获取 n 字节例程。接收所需字节数
S2LPGetNBytesBatch	ref_timer: 将每隔 ref_timer ms 发送一次数据。 num_packets: 数据包必须发送的次数	获取 n 字节批处理例程。接收所需的数据包数
S2LPSendPattern	buffer: 要发送的数据 n_bytes: 字节数	以循环模式发送缓冲区 (适用于测试极性模式)
S2LPSendPatternRaw	raw_buffer: 字节原始流	以循环方式发送缓冲区 (适用于测试极性模式)

指令	参数	说明
S2LPSendNBytes	buffer: 将通过 FIFO 发送的数据	根据配置的 FIFO TX 模式（直接、数据包等）发送 n 字节
S2LPSendNBytesBatch	ref_timer : 每隔 ref_timer ms 发送一次数据 num_packets: 数据包必须发送的次数 buffer: 将通过 FIFO 发送完成的数据	根据配置的 FIFO TX 模式（直接、数据包等）以批量模式发送 n 字节
S2LPGetRssiRunBatch	ref_timer : RSSI 读取 ref_timer ms	返回 RSSI 值
S2LPSendPatternGpio	pattern: 要发送的数据	使用 TX 直接 GPIO 口模式发送一个样本（4 字节）
SdkEvalIrqHandler	blocking: 如果为 1, 则阻止, 直至中断被置高（或停止 cmd 输入）, 如果为 0, 则不阻止。	等待 S2-LP IRQ 发起, 发起时返回 IRQ_STATUS。如果其他中断在 IRQ 队列中, 则会弹出并返回最后一个中断。
S2LPGetLibVersion	无	获取 S2-LP 库版本
S2LPSendBatchLP	en: 1: 使能; 0: 禁用	这会导致在 SendNbytesBatch 例程中使用低功耗待机模式
S2LPGetBatchLP	en: 1: 使能; 0: 禁用	这会导致在 GetNbytesBatch 例程中使用低功耗待机模式
S2LPGetRcoFrequency	无	获取 RCO 频率
SdkEvalSpiCommandStrobes	command_code ⁽⁹⁾	向 S2-LP 发送指令
SdkEvalSpiReadRegisters	address: 寄存器地址 n_regs: 读取的寄存器数量	从设备读取寄存器
SdkEvalSpiWriteRegisters	address: 寄存器地址 values: 要作为块写入的值	将寄存器写入设备
SdkEvalSpiReadFifo	n_bytes: 从 RX FIFO 读取的字节数	读取 RX FIFO
SdkEvalSpiWriteFifo	values: 将作为块写入 FIFO 的值	写入 TX FIFO
S2MGpioIrqConfiguration	gpio_pin: 0: GPIO0, 1: GPIO1, 2: GPIO2, 3: GPIO3 en: 1: 使能; 0: 禁用	在 EXTI（外部中断）模式下配置 uC 的 GPIO
S2MGpioGetValue	gpio_pin: 0: GPIO0, 1: GPIO1, 2: GPIO2, 3: GPIO3	获取配置为输入的 GPIO 的值
SdkEvalSdn	Value: 置入 SDN（1, 引脚高电平）, 退出 SDN（0, 引脚低电平）	驱动 S2-LP 关机引脚

指令	参数	说明
SdkEvalRfboardIdentification	xtal: 如果该值为 0, 则会自动检测 XTAL, 否则将其设为该值	通过从 EEPROM 获取一些信息来识别 RF 板, 从 EEPROM 获取一些信息并配置库
FEMSetBypass	旁路: 0: 无旁路, 1: 有旁路	将 FEM 设为旁路模式 (如果可用)
CliGetTimer	无	获取 CLI 定时器
CliResetTimer	无	复位 CLI 定时器
SdkEvalLedHandler	led: 0: LED1, 1: LED2, 2: LED3, 3: LED4, 4: LED5	将 LED 设为开或关 (根据母板配置)
EepromStatus	无	获取 EEPROM 状态值
EepromWritePage	page: 页码 offset: 开始从页面内写入的偏移量 values: 列出要写入的值	将值或值列表写入 EEPROM
EepromReadPage	page: 页码 offset: 开始从页面内部读取的偏移量 n_bytes: 要读取的字节数	从 EEPROM 读取值或值列表
SdkEvalGetVersion	无	获取母板版本
TimeMeasureService	gpio_pin: 0: GPIO0, 1: GPIO1, 2: GPIO2, 3: GPIO3	启用 GPIO 上的时间测量服务
GetTimeMeasure	gpio_pin: 0: GPIO0, 1: GPIO1, 2: GPIO2, 3: GPIO3	如果启用 TimeMeasureServiceAction, 则获取迄今为止所获得的时间测量结果
SdkEvalIrqRaised	blocking: 如果为 0, 则立即返回	等待设备中断。当 GPIO 发出 IRQ 时返回。不读取 IRQ_STATUS 寄存器
SdkEvalHiZ	无	将 GPIO 和 SPI 设为高阻抗配置
SdkComBaudrate	波特率: 值 (bps)	设置串行波特率

1. 请参考表 8. 调制类型。
2. 请参考表 15. CRC 模式。
3. 请参考表 16. WMBus 子模式。
4. 请参考表 9. S2-LP GPIO 模式。
5. 请参考表 10. S2-LP GPIO I/O 功能。
6. 请参考表 11. S2-LP IRQ。
7. 请参考表 13. 直接 TX 模式。
8. 请参考表 12. 直接 RX 模式。
9. 请参考表 14. S2-LP 指令。

表 8. 调制类型

调制类型	值
2-FSK 调制	0x00
4-FSK 调制	0x10
选择 BT = 0.5 的 2GFSK 调制	0xA0

调制类型	值
选择 BT = 1 的 2GFSK 调制	0x20
选择 BT = 0.5 的 G4FSK 调制	0xB0
选择 BT = 1 的 G4FSK 调制	0x30
OOK 调制	0x50
极性模式	0x60
无调制 (载波)	0x70

表 9. S2-LP GPIO 模式

GPIO 模式	值
GPIO 上的数字输入	1
GPIO 上的数字输出 (低电流)	2
GPIO 上的数字输出 (高电流)	3

表 10. S2-LP GPIO I/O 功能

输入/输出配置	GPIO 模式	值
如果配置为输出	nIRQ (中断请求, 低电平有效), POR 后的默认配置	0x00
	POR 反转 (低电平有效)	0x08
	唤醒定时器超时: 当 WUT 过期时为 “1”	0x10
	电池低电量检测: 当电池电量低于阈值设置时为 “1”	0x18
	TX 数据内部时钟输出 (在其上升沿对 TX 数据采样)	0x20
	TX 状态指示: 当 S2LP1 进入 TX 状态时为 “1”	0x28
	TX/RX FIFO 几乎已空标志	0x30
	TX/RX FIFO 接近全满标志	0x38
	RX 数据输出	0x40
	RX 时钟输出 (从接收的数据恢复)	0x48
	RX 状态指示: 当解调器开启时为 “1”	0x50
	器件未处于睡眠或待机模式时 (高电平有效)	0x58
	器件待机时 (高电平有效)	0x60
	用于天线分集的天线开关	0x68
	检测到有效前导码标志	0x70
	检测到同步 WordSync 字标志	0x78
	RSSI 高于阈值	0x80
	MCU 时钟	0x88
	TX 或 RX 模式指示器 (用于使能外部 PA)	0x90
	VDD (用于模拟 MCU 的额外 GPIO, 可通过 SPI 编程)	0x98
	GND (用于模拟 MCU 的额外 GPIO, 可通过 SPI 编程)	0xA0
	外部 SMPS 使能信号 (高电平有效)	0xA8
	器件处于睡眠模式 (高电平有效)	0xB0
	器件处于就绪状态 (高电平有效)	0xB8

输入/输出配置	GPIO 模式	值
如果配置为输出	器件处于锁定模式（高电平有效）	0xC0
	器件正在等待锁定（高电平有效）	0xC8
	TX_DATA_OOK 信号（在 OOK 模拟平滑模式下生成的内部控制信号）	0xD0
	器件正在等待来自 XO 的 READY2 高电平信号	0xD8
	器件正在等待允许 PM 块建立的定时器超时	0xE0
	器件正在等待 VCO 校准结束	0xE8
	器件启用完整的 SYNTH 块回路	0xF0
如果配置为输入	TX 指令	0x00
	RX 指令	0x08
	用于直接调制的 TX 数据输入	0x10
	从外部输入唤醒	0x18
	34.7 kHz 的外部时钟（用于 LDC 模式时序）	0x20

表 11. S2-LP IRQ

IRQ 模式	值
IRQ: RX 数据就绪	0x00000001
RX 数据被丢弃（在过滤时）	0x00000002
TX 数据已发送	0x00000004
达到了最大 TX 次数	0x00000008
CRC 错误	0x00000010
TX FIFO 下溢/上溢错误	0x00000020
RX FIFO 下溢/上溢错误	0x00000040
TX FIFO 接近全满	0x00000080
TX FIFO 几乎已空	0x00000100
RX FIFO 接近全满	0x00000200
RX FIFO 几乎已空	0x00000400
CCA 期间的最大回退次数	0x00000800
检测到有效前导码	0x00001000
检测到同步字	0x00002000
RSSI 高于阈值	0x00002000
LDC 模式下的唤醒超时	0x00008000
READY 状态	0x00010000
MCU_CK_CONF_CLOCK_TAIL_X 时钟周期后的待机状态	0x00020000
电池电量低于阈值	0x00040000
上电复位	0x00080000
欠压事件（精确和非精确）	0x00100000
LOCK 状态	0x00200000
RX 操作超时	0x10000000
RX 监听操作超时	0x20000000

IRQ 模式	值
所有 IRQ	0x7FFFFFFF

表 12. 直接 RX 模式

RX 模式	值
正常 RX 模式	0x00
直接 RX FIFO 模式	0x10
直接 RX GPIO 模式	0x20

表 13. 直接 TX 模式

TX 模式	值
正常 TX 模式	0x00
直接 TX FIFO 模式	0x04
直接 TX GPIO 模式	0x08
PN9 TX 模式	0x0C

表 14. S2-LP 指令

指令	值
CMD_TX - 开始传输，仅在就绪状态下有效	0x60
CMD_RX - 开始接收，仅在就绪状态下有效	0x61
CMD_READY - 进入就绪状态；仅在待机、睡眠或锁定状态下有效	0x62
CMD_STANDBY - 进入待机状态，仅在就绪状态下有效	0x63
CMD_SLEEP - 进入睡眠状态，仅在就绪状态下有效	0x64
CMD_LOCKRX - 使用 synth 的 RX 配置进入锁定状态，仅在就绪状态下有效	0x65
CMD_LOCKTX - 使用 synth 的 TX 配置进入锁定状态，仅在就绪状态下有效	0x66
CMD_SABORT - 强制从 TX 或 RX 状态退出，并进入就绪状态，仅在 TX 或 RX 状态下有效	0x67
CMD_LDC_RELOAD - LDC 模式：用存储在 LDC_PRESCALER / COUNTER 寄存器中的值重新加载 LDC 定时器，在所有状态下均有效	0x68
CMD_RCO_CALIB - 开始（或重新开始）RCO 校准	0x69
CMD_SRES - 复位除 SPI 寄存器以外的所有数字部分	0x70
CMD_FLUSH_RX_FIFO - 清除 RX FIFO，在所有状态下均有效	0x71
CMD_FLUSH_TX_FIFO - 清除 TX FIFO，在所有状态下均有效	0x72
MD_SEQUENCE_UPDATE - 自动重传：用存储在 PROTOCOL[2] 寄存器中的值重新加载包序列计数器，在所有状态下均有效	0x73

表 15. CRC 模式

CRC 模式	值
无 CRC	0x00
Poly 0x07	0x20
Poly 0x8005	0x40
Poly 0x1021	0x60

CRC 模式	值
Poly 0x864CFB	0x80
Poly 0x04C011BB7	0xA0

表 16. WMBus 子模式

WMBus 模式	值
WMBUS_SUBMODE_NOT_CONFIGURED	0
WMBUS_SUBMODE_S1_S2_LONG_HEADER	1
WMBUS_SUBMODE_S1_M_S2_T2_OTHER_TO_METER	2
WMBUS_SUBMODE_T1_T2_METER_TO_OTHER	3
WMBUS_SUBMODE_R2_SHORT_HEADER	4

5.1.2 IAR 项目

工作区文件被称为 CLI_Project.eww，位于 STM32 或 BlueNRG 文件夹下的 EWARM 文件夹中的 Projects/Projects_Cube/S2-LP_DK/S2LP_CLI_Project 目录中。

要使用 IAR Embedded Workbench for ARM 项目：

- 第 1 步。** 打开 Embedded Workbench for ARM 并选择 “File（文件）” > “Open（打开）” > “Workspace（工作区）” 菜单。
- 第 2 步。** 打开 IAR 项目 .../Projects/Projects_Cube/S2-LP_DK/S2LP_CLI_Project，并选择 STM32 或 BlueNRG 文件夹下的 EWARM/CLI_Project.eww 文件
- 第 3 步。** 选择要构建的所需配置
- 第 4 步。** 选择下载与调试按钮，以重新编译并链接整个应用程序、下载相关固件并进入调试模式。

5.1.3 MDK-ARM KEIL 项目

工作区文件被称为 CLI_Project.uvprojx，位于 STM32 或 BlueNRG 文件夹下的 MDK-ARM 文件夹中的 Projects/Projects_Cube/S2-LP_DK/S2LP_CLI_Project 目录中。

要使用 KEIL uVision for ARM 项目：

- 第 1 步。** 打开 KEIL uVision for ARM，并选择[项目]>[打开项目]菜单
- 第 2 步。** 打开 KEIL 项目 .../Projects/Projects_Cube/S2-LP_DK/S2LP_CLI_Project，并选择 STM32 或 BlueNRG 文件夹下的 MDK-ARM/CLI_Project.uvprojx 文件
- 第 3 步。** 选择要构建的所需配置
- 第 4 步。** 选择[项目]>[重建全部目标文件]，以重新编译并链接整个应用程序。
- 第 5 步。** 选择[项目]>[下载]，以下载相关二进制图像。

5.2 库示例

S2-LP 开发工具包包含一组简单的示例，用于说明如何使用某些 S2-LP 功能（如分组处理器、低功耗模式、自动重传、CSMA 引擎等）。

每个示例至少包括两个名为 A 和 B 的程序，分别用作发射器和接收器：

- Basic generic：有效负载长度小于 FIFO 大小（128 字节）的 S2-LP BASIC 数据包交互。
- FIFO handler：有效负载长度大于 FIFO 大小的 S2-LP BASIC 数据包交互。
- wM-Bus STD：采用类似于 wM-Bus 格式的数据包交互。
- Sniff：使用接收器的快速 RX 终止功能的数据包交互。
- LDCR：使用低占空比模式的数据包交互。
- Stack LLP：使用 S2-LP S2ack 数据包的嵌入式链路层功能进行通信（自动确认与自动重传）。
- CSMA：当干扰源（第三个节点发送载波来实现）干扰通信时，在两个节点之间交换数据包。

- Chat: 在两个节点之间交换字符串。对于该示例，角色是对称的，因此不必具有 A 和 B 程序。

这些示例以源格式提供，并作为 IAR（所需工具链为 IAR Embedded Workbench for ARM（EWARM）工具链（V7.40.3 或更高版本））和 MDK-ARM Keil 项目（所需工具链为 V5.17 或更高版本）。

5.2.1 IAR 项目

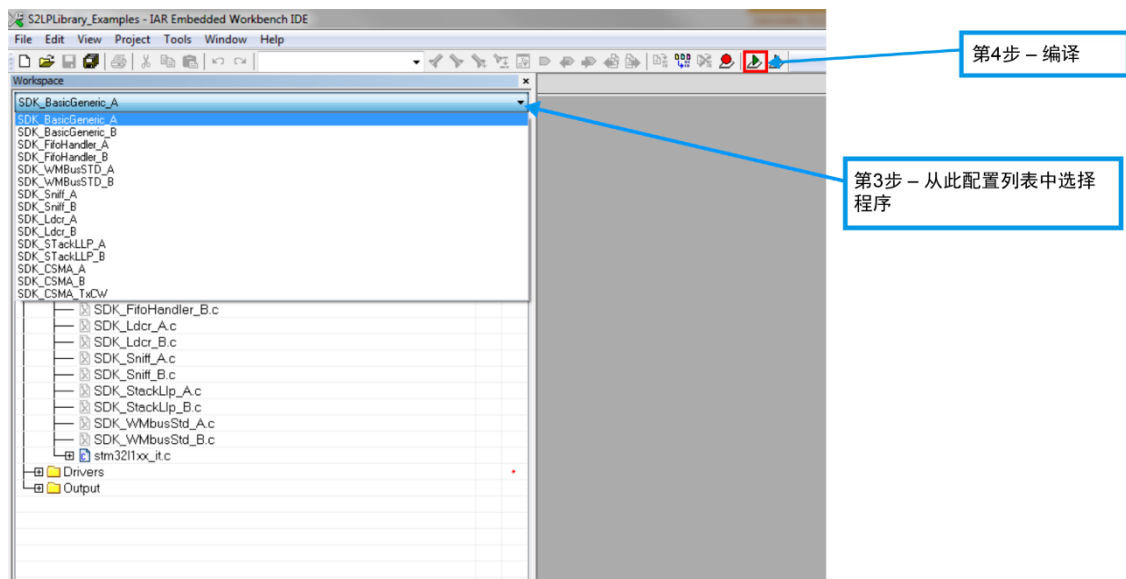
工作区文件被称为 S2LPLibrary_Examples.eww，位于 STM32 或 BlueNRG 文件夹下的 EWARM 文件夹中的 Projects/Projects_Cube/S2-LP_DK/S2LPLibrary_Examples 目录中。

每个程序都是一个 IAR 配置，并可以使用适用于 STM32L053R8/STM32L152RE 的嵌入式 ST-LINK 或适用于 BlueNRG-1/BlueNRG-2 板的外置 ST-LINK 编译和烧录到母板上。

要使用 IAR Embedded Workbench for ARM 项目：

- 第 1 步.** 打开 Embedded Workbench for ARM 并选择 “File（文件）” > “Open（打开）” > “Workspace（工作区）” 菜单。
- 第 2 步.** 打开 IAR 项目 .../Projects/Projects_Cube/S2-LP_DK/S2LPLibrary_Examples，并选择 STM32 或 BlueNRG 文件夹下的 EWARM/S2LPLibrary_Examples.eww 文件
- 第 3 步.** 选择要构建的所需配置
- 第 4 步.** 选择下载与调试按钮，以重新编译并链接整个应用程序、下载相关固件并进入调试模式。

图 29. IAR 项目

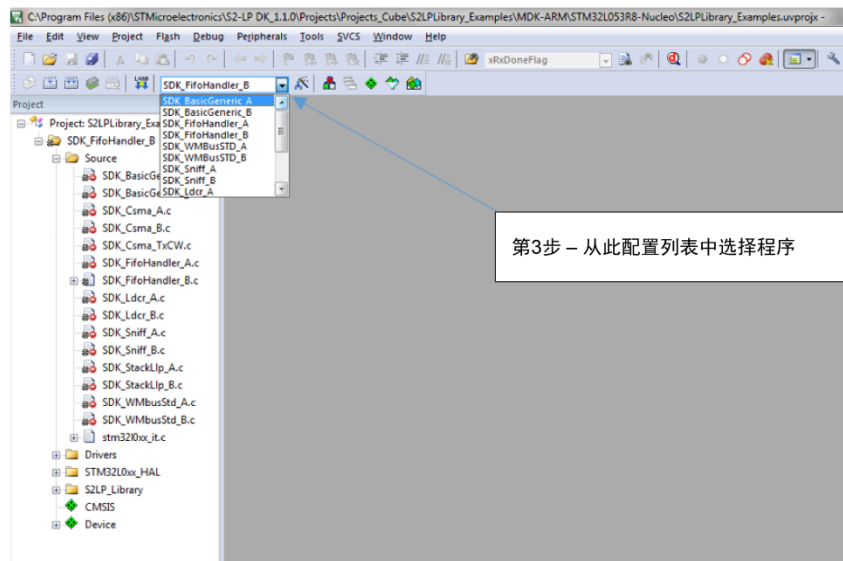


5.2.2 MDK-ARM KEIL 项目

要使用 KEIL uVision 5 for ARM 项目：

- 第 1 步.** 打开 KEIL uVision 5 for ARM，并选择 “Project（项目）” -> “Open Project（打开项目）” 菜单
- 第 2 步.** 打开 KEIL 项目.../Projects/Projects_Cube/S2-LP_DK/S2LPLibrary_Examples，并选择 STM32 或 BlueNRG 文件夹下的 MDK-ARM/S2LPLibrary_Examples.uvprojx 文件
- 第 3 步.** 选择要构建的所需配置

图 30. Keil 项目



第 4 步. 选择[项目]>[重建全部目标文件], 以重新编译并链接整个应用程序。

第 5 步. 选择[项目]>[下载], 以下载相关二进制图像。

6 版本说明

S2-LP 开发套件软件包的版本说明包含在 Documents/S2LP_DK_release_notes/Release_Notes.html（html 格式）中。

打开 Documents/index.html 文件，以获取全局文档索引。

7 授权

可通过 Documents/index.html 文件访问 S2-LP 开发套件软件包许可文件。

版本历史

表 17. 文档版本历史

日期	版本	变更
2016 年 12 月 19 日	1	初始版本。
2017 年 2 月 01 日	2	更新了 1.2 节：“STSW-S2LP-DK 开发套件设置”，第 2.1 节：“STEVAL-FKI868V1 和 STEVAL-FKI433V1 评估板”，第 2.2 节：“STEVAL-FKI915V1 评估板”，图 2：“STEVAL-FKI915V1 评估板特性”，第 2.3.4 节：“STM32L 微控制器”，第 3.1 节：“安装”，第 3.9.2.1 节：“固件升级”，第 4 节：“固件示例” 增加了第 4.1 节：“IAR 项目”和第 4.2 节：“MDK-ARM Keil 项目”
2017 年 7 月 26 日	3	更新了图 6：“S2-LP GUI 主窗口”，第 3.2.1 节：“连接面板”，图 10：“基本数据包设置面板”，第 3.3.2 节：“包设置：WMBUS”，第 5 节：“固件示例”和第 3.4 节：“发送测试”。 增加了第 4 节：“S2-LP 电流评估工具”，第 4.1 节：“唤醒模式”，第 4.2 节：“功耗模式”，第 4.3 节：“TX”和第 4.4 节：“当前配置文件图”。
2018 年 5 月 23 日	4	更新了频段并支持 HW。 添加新器件：STEVAL-FKI512V1。
2019 年 7 月 3 日	5	更新了简介、第 1.1 节：“系统要求”、第 2.1 节：“STEVAL-FKI868V2、STEVALFKI433V2 和 STEVAL-FKI512V1 评估板”，第 2.1.2 节：“STEVAL-FKI868V2、STEVALFKI433V2 和 STEVAL-FKI512V1 电源”、第 2.2 节：“STEVAL-FKI915V1 评估板”、第 2.2.2 节“STEVAL-FKI915V1 电源”、第 3.1 节：“安装”、第 3.2.2 节：“射频设置面板”、第 3.9.2.1 节：“固件升级”、第 5.2.1 节：“IAR 项目”和第 5.2.2 节：“MDK-ARM KEIL 项目”。 添加了第 2.5 节“STEVAL-IDB007V2 和 STEVAL-IDB008V2 评估板”、第 2.6 节“STEVAL-FKI001V1 评估板”、第 5.1 节：“命令行界面（CLI）”、第 5.1.1 节：“CLI 命令、调制类型和 S2-LP 模式”、第 5.1.2 节：“IAR 项目”以及第 5.1.3 节：“MDK-ARM KEIL 项目”。
2020 年 3 月 9 日	6	更新了简介，第 1.1 节“系统要求”，图 6. S2-LP GUI 主窗口以及第 3.2.2 节“射频设置面板”。 增加了第 2.4 节“X-NUCLEO-S2915A1”。
2021 年 5 月 27 日	7	更新了第 5 节“固件示例”。
2022 年 11 月 23 日	8	更新了第 1.2 节“STSW-S2LP-DK 开发套件设置”和第 5 节“固件示例”。

目录

1	概述	2
1.1	系统要求	2
1.2	STSW-S2LP-DK 开发套件设置	2
2	硬件说明	3
2.1	STEVAL-FKI868V2、STEVAL-FKI433V2 和 STEVAL-FKI512V1 评估板	3
2.1.1	S2-LP 连接	3
2.1.2	STEVAL-FKI868V2、STEVAL-FKI433V2 和 STEVAL-FKI512V1 电源	4
2.2	STEVAL-FKI915V1 评估板	4
2.2.1	S2-LP 连接	5
2.2.2	STEVAL-FKI915V1 电源	6
2.3	X-NUCLEO-S2868A1	6
2.4	X-NUCLEO-S2915A1	6
2.5	STM32 Nucleo 板	6
2.5.1	按钮	6
2.5.2	LED	7
2.5.3	板载 ST-LINK	7
2.5.4	STM32L053R8/STM32L152RE 微控制器	7
2.6	STEVAL-IDB007V2 和 STEVAL-IDB008V2 评估板	7
2.7	STEVAL-FKI001V1 评估板	7
2.8	硬件设置	7
3	GUI 软件说明	9
3.1	安装	9
3.2	详细说明	9
3.2.1	连接面板	10
3.2.2	射频设置面板	10
3.2.3	RF 测试模式	11
3.3	包设置	12
3.3.1	包设置: BASIC	13
3.3.2	包设置: WMBUS	14
3.4	发送测试	14
3.5	底层命令	17
3.6	Running RSSI	18
3.7	如何使用信号发生器来运行 BER 测试	19
3.8	寄存器表	20
3.9	菜单栏	22

3.9.1	文件.....	22
3.9.2	工具.....	22
3.9.3	帮助.....	24
3.9.4	设备模拟器.....	25
4	S2-LP 电流评估工具.....	26
4.1	唤醒模式.....	26
4.2	功耗模式.....	27
4.3	TX.....	27
4.4	当前配置文件图.....	28
5	固件示例.....	29
5.1	命令行接口（CLI）.....	30
5.1.1	CLI 指令、调制类型和 S2-LP 模式.....	30
5.1.2	IAR 项目.....	38
5.1.3	MDK-ARM KEIL 项目.....	38
5.2	库示例.....	38
5.2.1	IAR 项目.....	39
5.2.2	MDK-ARM KEIL 项目.....	39
6	版本说明.....	41
7	授权.....	42
	版本历史.....	43
	表格索引.....	46
	图片目录.....	47

表格索引

表 1.	S2-LP: FKI868V2 和 FKI433V2 评估板引脚说明	4
表 2.	S2-LP: FKI915Vx 评估板引脚说明	5
表 3.	STEVAL-FKI915V1 扩展板跳线说明	6
表 4.	STM32 Nucleo 开发板配置文件	29
表 5.	BlueNRG 评估板配置文件	29
表 6.	需要配置的参数列表	29
表 7.	CLI 指令	30
表 8.	调制类型	34
表 9.	S2-LP GPIO 模式	35
表 10.	S2-LP GPIO I/O 功能	35
表 11.	S2-LP IRQ	36
表 12.	直接 RX 模式	37
表 13.	直接 TX 模式	37
表 14.	S2-LP 指令	37
表 15.	CRC 模式	37
表 16.	WMBus 子模式	38
表 17.	文档版本历史	43

图片目录

图 1.	STEVAL-FKI868V2 评估板特性	3
图 2.	STEVAL-FKI915V1 评估板特性	5
图 3.	NODE_L152RE 磁盘驱动器	9
图 4.	连接设置 1: 1 台配备 S2-LP-DK GUI 的 PC	9
图 5.	连接设置 2: 2 台配备 S2-LP-DK GUI 的 PC	9
图 6.	S2-LP GUI 主窗口	10
图 7.	可用的 COM 端口	10
图 8.	S2-LP 射频设置	11
图 9.	RF 测试模式按钮	12
图 10.	Basic 数据包设置面板	13
图 11.	wM-Bus 包设置面板	14
图 12.	发送测试面板	15
图 13.	底层命令面板	17
图 14.	读/写寄存器窗口	18
图 15.	“Running RSSI” 选项卡	19
图 16.	BER 测试平台架构	19
图 17.	寄存器表	21
图 18.	文件列表	22
图 19.	工具列表	23
图 20.	Help 菜单	25
图 21.	S2-LP 仿真器	25
图 22.	S2-LP 电流评估工具 - GUI 概述	26
图 23.	唤醒模式输入	27
图 24.	唤醒模式输出	27
图 25.	功耗模式输入	27
图 26.	功耗模式输出	27
图 27.	TX 部分	27
图 28.	电流图	28
图 29.	IAR 项目	39
图 30.	Keil 项目	40

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“意法半导体”）保留随时对 ST 产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于意法半导体产品的最新信息。意法半导体产品的销售依照订单确认时的相关意法半导体销售条款。

买方自行负责对意法半导体产品的选择和使用，意法半导体概不承担与应用协助或买方产品设计相关的任何责任。

意法半导体不对任何知识产权进行任何明示或默示的授权或许可。

转售的意法半导体产品如有不同于此处提供的信息的规定，将导致意法半导体针对该产品授予的任何保证失效。

ST 和 ST 标志是意法半导体的商标。关于意法半导体商标的其他信息，请访问 www.st.com/trademarks。其他所有产品或服务名称是其各自所有者的财产。本文档中的信息取代本文档所有早期版本中提供的信息。

© 2023 STMicroelectronics - 保留所有权利