

LSD4RF-2F717N01

WOR 应用报告



型号: LSD4RF-2F717N01

提交日期 :2014.04

文件修改记录台帐

[illegible]

目 录

第一章 简介.....4

第二章 概述.....4

第三章 应用实例.....4

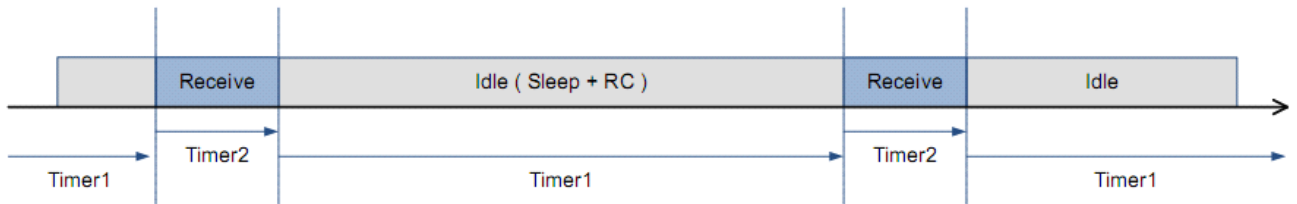


第一章 简介

本章主要是讲解 LSD4RF-2F717N01 WOR 应用报告，以及 WOR 注意事项。

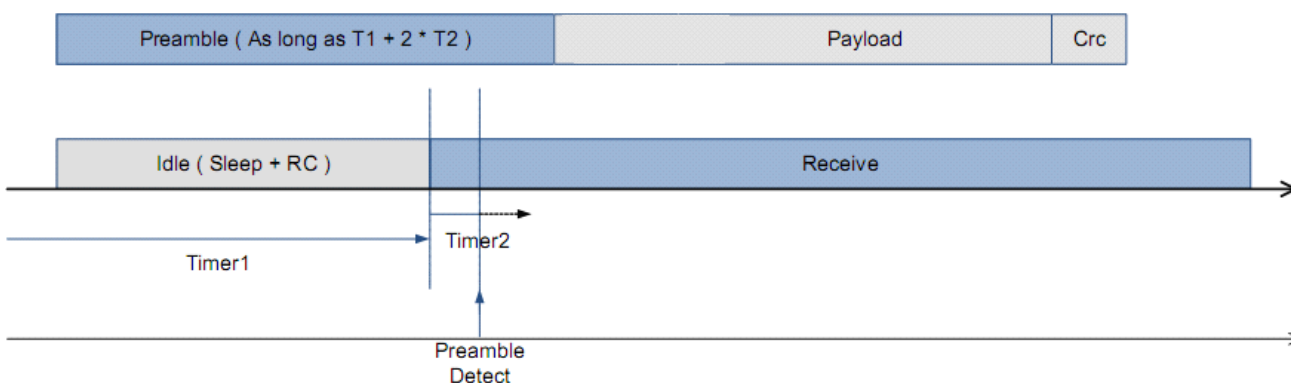
第二章 概述

LSD4RF-2F717N01 的 WOR 的工作原理如下图所示。



RF 逐时间周期的切换 sleep 模式与接收模式

这里我们定义 **Timer1** 为低功耗睡眠时间区间，**Timer2** 为功耗比较大的接收时间区间。



RF 唤醒 WOR 的工作原理

从上图可以看出，若要可靠唤醒 WOR，发送包的前导码必须大于 $T1+2*T2$ 才能可靠唤醒。其中负载部分可以定义为被唤醒的节点的 ID 等，可由客户自定义。

以上图示为 WOR 的工作过程，但需要注意的是，在 LORA 调制模式下，由于 LORA 模式是扩频传输，LORA 只能用软件 WOR 方式， $T2$ 时间的接收采用 CAD 方式实现。换句话说来解释就是，MCU 发命令让 RF 进入睡眠，MCU 定时 $T1$ 时间，时间到，MCU 发送命令让 RF 进入 CAD 模式，以此循环整个过程，即实现了软件 WOR 的过程。

第三章 应用实例

下面以 WOR CodeExamples 为例，介绍如何利用 DEMO 程序设计一个完整的 WOR 的过程。

1. 初始化 WOR。LSD_RF_WORInit(); //WOR 初始化

2. 调用 LSD_RF_WOR_Execute(0); //启动执行 WOR

启动 WOR 函数主要内容是 LSD_RF_Sleepmode(); //进入睡眠模式

ON_Sleep_Timerout(); //启动睡眠超时定时器

可以看出，启动 WOR 后，MCU 命令 RF 进入睡眠，同时开启了一个 4s 的定时器。

3. 四秒定时器时间到后，调用 LSD_RF_WOR_Execute(1); //启动 CAD 采样一次。 函数主要内容为。OFF_Sleep_Timerout(); //关闭睡眠超时定时器

LSD_RF_CAD_Sample(); //启动 CAD 一次

可以看出，4s 的时间到后，MCU 命令 RF 进入 CAD 模式，采用一次。

4. CAD 采用结束后，LSD_RF_WOR_Execute(0); //重新进入睡眠模式

5. 以此循环整个过程，即实现了睡眠---->CAD----->睡眠的一个循环过程，也就是 WOR 的过程。

6. 唤醒包调用 LSD_RF_Awake(WakeAddr, 2);即可唤醒 WOR，WakeAddr 为唤醒包的载荷，可以自定义唤醒地址。

睡眠时间如下：



从上图显示，模块 4s 的睡眠，睡眠时间到，进入 CAD，以此循环整个过程。

CAD 时间如下：



4s 时间到后，进入 CAD 模式，从上图可以看出，整体分为三个部分，一个是切换过程，一个是接收过程，一个是计算过程，整个三个部分组合一体为一个 CAD 过程。

详细如下。

一：切换时间为 320us



二：CAD 的接收时间，约为 4.28ms



三：CAD 的计算时间

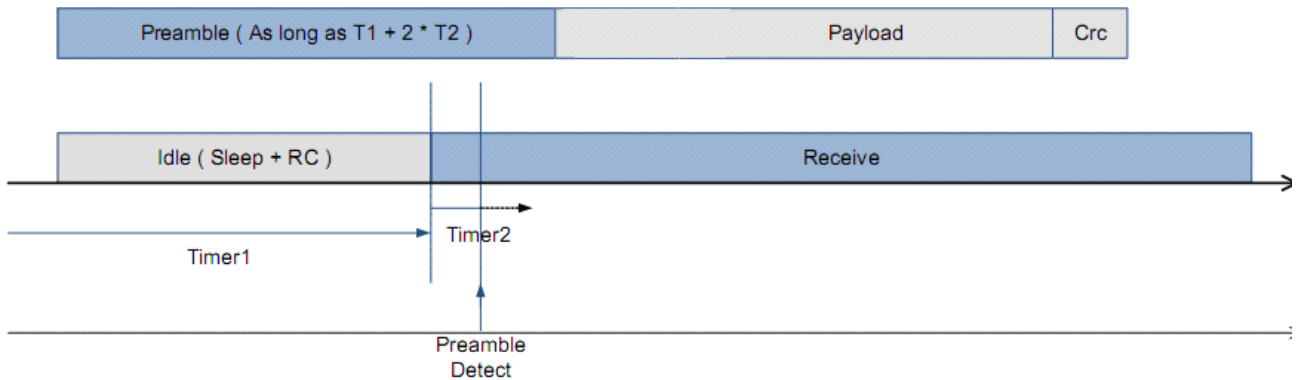


其中 CAD 的接收时间 = $(2^{\text{SF}} + 32) / \text{BW} = (2^9 + 32) / 125\text{K} = 4.35\text{ms}$ 。与时间测试结果基本一致。

平均功耗约 = $11000\text{mA} \times 4.3\text{ms} / 4000\text{ms} + 5500\text{mA} \times 2.8\text{ms} / 4000\text{ms} + 1.7\mu\text{A} + 3.5\mu\text{A}$ (MCU 全速定时器运行保守功耗) = $11.8\mu\text{A} + 3.85\mu\text{A} + 1.7\mu\text{A} + 3.5\mu\text{A}$ 约 = $21\mu\text{A}$ 左右。

从以上案例可以看出，整个 WOR 的过程的平均功耗，取决于 MCU 定时器定时睡眠的睡眠时间，这里的案例是采用的 4s，用户可以自己调整，而 CAD 的时间，取决于 SF 和 BW，DEMO 程序的 SF 和 BW 是固定的。用户不需要修改。

发送唤醒包需要注意的是如下，



从上图可以看出，若要保证成功唤醒，唤醒的前导码必须大于 $T1 + 2 * T2$ 。从刚才计算和截图可以知道前导时间 = $4000\text{ms} + 2 * (4.3 + 2.8)\text{ms} = 4015\text{ms}$ ，为了能可靠唤醒，需要大于 4015ms ，这里我取值为 4116ms 作为前导码的时间唤醒。因为每个 LORA 码源的时间是 $TS = 2^{\text{SF}} / BW = 2^9 / 125\text{K} = 4.1\text{ms/symb.}$ 所以可以计算出前导长度值 = $41116 / 4.1 = 0x03EC$ 。这样 `LSD_RF_Awake(uint8_t*cbuf,uint8_t csize)` 唤醒换算里的前导码就应该是。

```
SX1276Write( REG_LR_PREAMBLEMSB, 0x03); //set preamble length
```

```
SX1276Write( REG_LR_PREAMBLELSB, 0xEC); //set preamble length
```

若客户需要修改不同长度的前导，可根据以上计算方法得到不同的前导时间。

注意：对代码移植和使用过程中遇到疑问，欢迎联系利尔达 WSN 事业部 RF 团队提供支持。