

BL1800RF

----2.4GHz 高集成度无线收发芯片

1. 概述

BL1800RF是一款工作在2.4GHz, 世界通用ISM频段的单片集成无线收发芯片。该芯片集成了射频接收器、射频发射器、频率综合器、GFSK调制器、GFSK解调器等功能模块。通过SPI或I2C接口可以对输出功率、频道选择以及协议进行灵活配置, 并且内置CRC、FEC、自动应答和自动重传机制, 可以大大简化系统设计并优化性能。外围电路简单, 只需搭配极少的被动器件就能达到优良的收发性能。

芯片主要特点:

低功耗

当工作在发射模式下(发射功率为0dBm)时电流消耗为15mA;

当工作在接收模式时电流消耗为16mA;

当工作在休眠模式时电流消耗低于1uA。

低成本

低成本系统解决方案;

内部集成上电复位和软件复位功能, 外围控制简单;

可采用双层或单层PCB, 也可采用印制板天线方式;

I2C接口

高性能

最高数据码率可达1Mbps.

最大输出功率7dBm, 灵敏度可达-89dBm

支持RSSI功能

正常配置下(发射功率7dBm), 空旷地带通信距离大于80米.

自动应答及自动重传

应用领域:

工业传感器及无线工控设备

智能家居设备

无线游戏设备

智能运动设备

智能电视遥控器

无线标签

无线门禁

无线遥控玩具

2. 极限值

表2.1 参数极限值

参数	符号	最小值	最大值	单位
工作温度	Top	-20	100	℃
存储温度	Tstor	-55	125	℃
工作电压	VDD	-0.3	3.7	V
输入射频信号强度	Pin_max		+10	dBm
ESD(人体模型)	ESD_HBM		2	KV

* 注意：强行超过一项或多项极限值使用会导致器件永久性损坏。

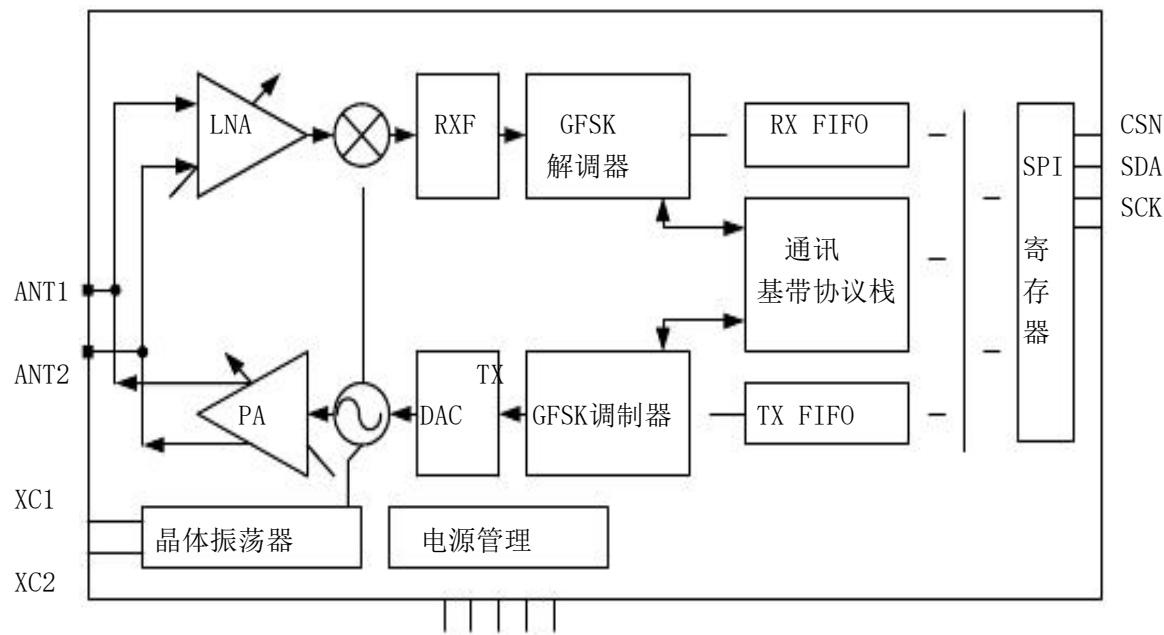
* 小心：芯片为静电敏感器件，操作时请遵守防护规则。

3. 主要电特性

表3.1 芯片主要电特性

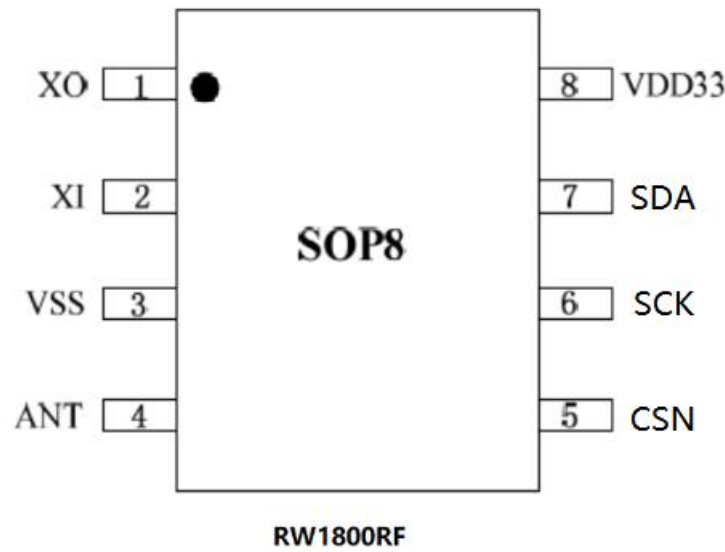
特性	条件 (VCC=3.3V, TA=25℃)	参数值			单位
		最小	典型	最大	
Icc	休眠		0.9		uA
	待机		1		mA
	发射(0dBm)		14		mA
	接收		16		mA
f _{OP}	工作频率	2402		2530	MHz
f _{XTAL}	晶振频率		12		MHz
C _{XTAL}	晶振负载电容		22		pF
SENS	灵敏度(0.1%BER)		-89		dBm
PRFC	输出功率范围	-24		7	dBm
VDD	供电电压	1.9	3	3.6	V
Cin	输入电容			10	pF

4. 系统结构方框图



BL1800RF 芯片框图

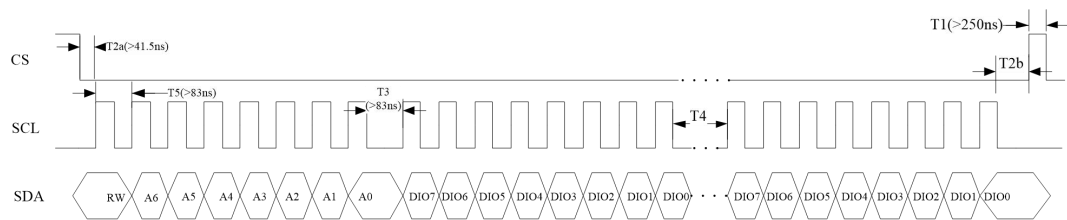
5. 引脚定义



脚位	脚位名称	管脚类型	描述
1	XTALp	AO	晶振输出
2	XTALn	AI	晶振输入
3	VSS	GND	接地
4	ANT	RFI	天线输入输出
5	CSN	DI	使能信号,低有效
6	SCK	DI	SPI时钟脚
7	SDA	DI/O	SPI数据脚
8	VDD	PWR	3.3V电源脚

6. SPI 接口

6.1 SPI 命令格式



自定义的三线制 SPI 接口：标准 SPI 接口中的 MOSI 和 MISO 共用一根数据线 SDA，其工作顺序和标准 SPI 接口一样。在使用三线 SPI 接口时，需要先写寄存器使能，即上电后将 reg0x2A 从 0XH001 改写为 0X2001。

Notes:

- 1、SPI 为下降沿采样数据，上升沿改变数据；
- 2、SPI 读写位： 写=0，读=1；
- 3、访问接收 FIFO 寄存器 0X28 时，可以按字节读。访问多个 FIFO 数据时可以用一个 CS 周期；
- 4、访问除 FIFO 外的其他寄存器时，一次要读写 16bits；
- 5、访问除 FIFO 外的其他多个寄存器时，可以用一个 CS 周期。此时，地址只要写一次，然后是 16bit 数据，当写完一个寄存器值后，芯片会自动增加寄存器地址。

Name	Min	Typ.	Max	Description
T1	250ns			两次 SPI 访问的间隔时间
T2a, T2b	41.5ns			CS 和 SPI_CLK 的间隔
T3	Note 1			地址和数据间隔时间
T4	Note 1			高位字节和低位字节的时间间隔
T5	83ns			SPI_CLK 时钟周期
T6	Note 2			两个寄存器数据的时间间隔

Notes:

1. 在访问寄存器 0x28 中的 FIFO 数据时，芯片需要 450ns 去找到正确的读 FIFO 读取的指针地址。
2. 当读寄存器 0x28 中的 FIFO 数据时，至少需要等 450ns。
读其他寄存器时，T6min=41.5ns。

7. 状态机控制图

BL1800RF有4种工作模式，分别是休眠模式，待机模式，发射模式，接收模式

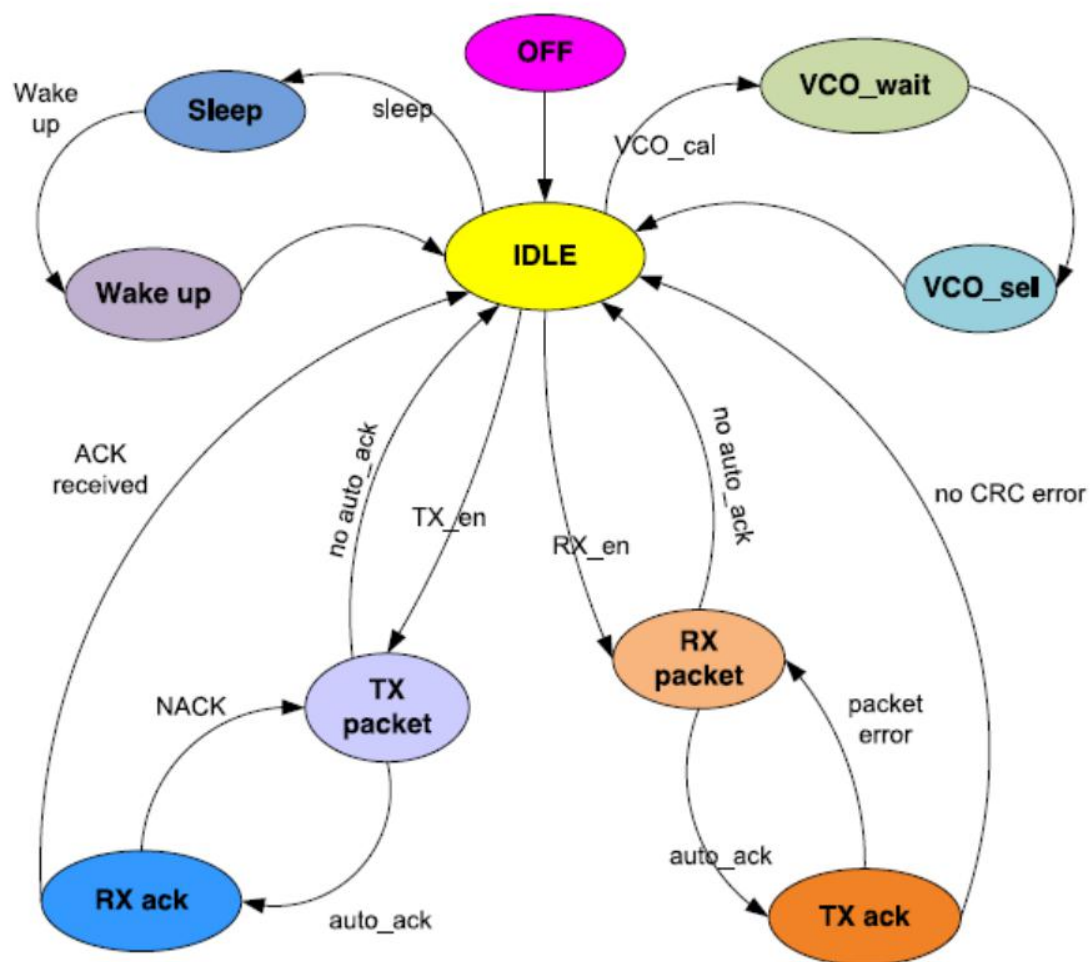


图7.1 状态机控制图

8. 通讯协议描述

8.1 数据包格式

前导码	地址	数据标识	数据	CRC
-----	----	------	----	-----

前导码: 1 字节.

地址: 32/40 bits.

数据标识: 4~9 bits.

数据: TX/RX的数据.

CRC: 16-bit CRC校验.

8.2 FIFO 长度

默认是32 个字节, 接收和发送共用, 也可设置成分开使用各32 个字节

8.3 清空FIFO 指针

接收FIFO 和发送FIFO 的读写指针都可通过对0x25, 0x26 相应寄存器写 ‘1’ 来清除
当接收到数据包, 读指针将指示 FIFO 中有多少 bytes 数据。

当接收器收到 SYNC 时, 接收FIFO 写指针会自动清0。

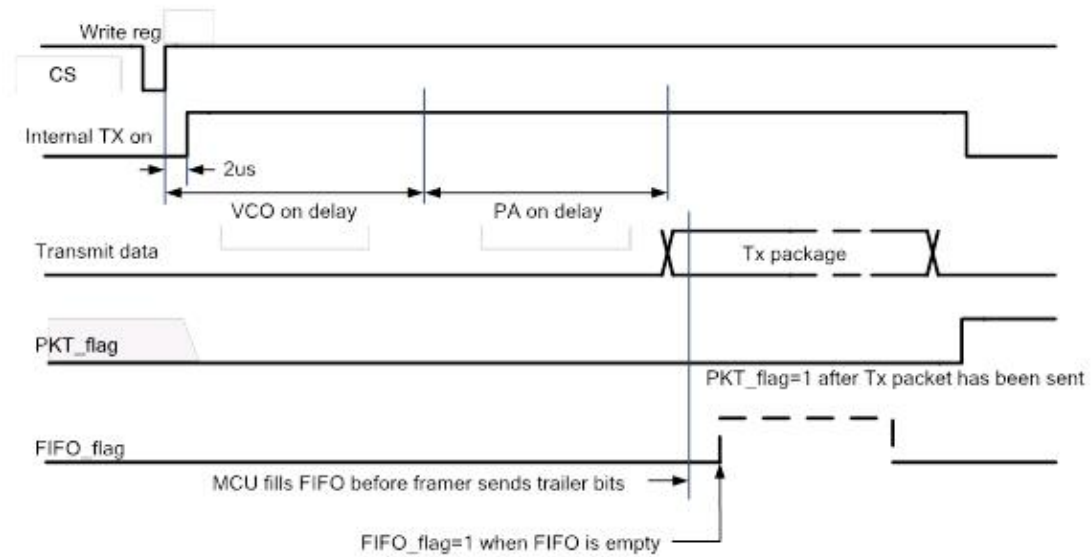
当发射器发送完数据后, 发射FIFO 读指针会自动清0。

8.4 数据包payload 长度

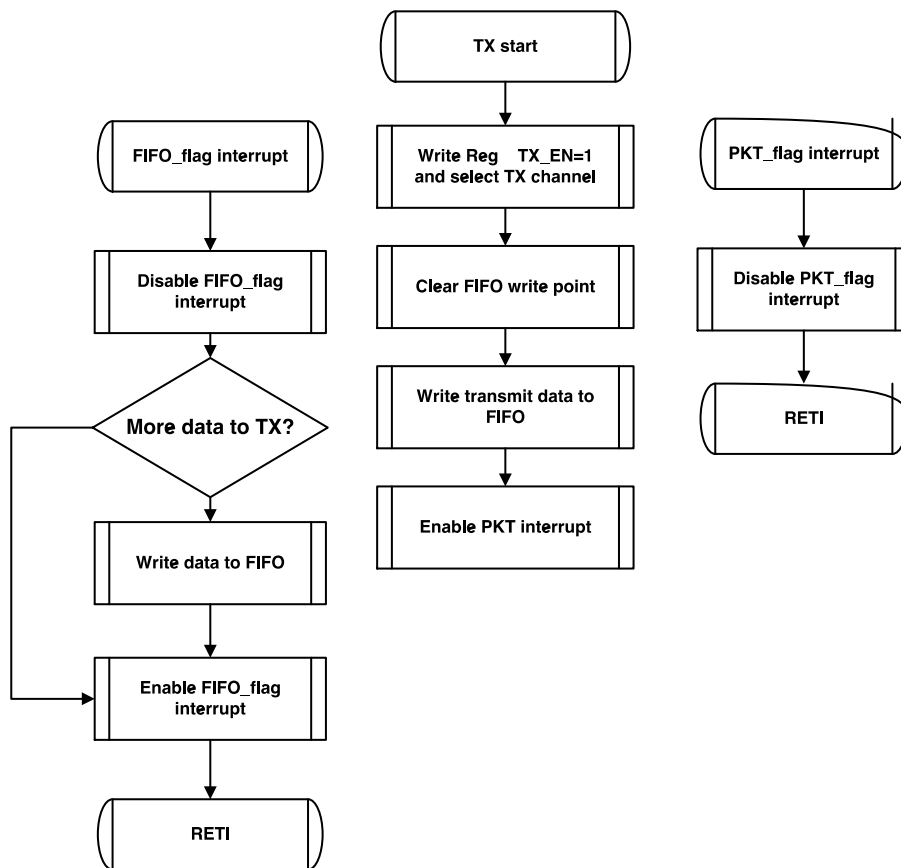
两种方式去确定 TX/RX 包长度。当寄存器0x01[2]=1 时, 内部状态机会根据 payload 第一个 byte 数据来检测包长度。如要发 8 个 byte, 第一个 byte 应写 8, 总字长为 9 个 byte。
当寄存器0x01[2]=0, 第一个 byte 数据没什么特殊意义。数据包长度将由 TX FIFO 何时为空或者何时清空 TX_EN 来决定。

8.5 发射时序

下面显示 TX 时序: 当 MCU 将寄存器 0x00[15]写为‘1’后, 同时写寄存器 0x00[6:0]设定好发射信道, 芯片将自动根据 payload 来产生包。MCU 需要在发射 trailer 前写入发射数据。^[1]
如果包长度超过 FIFO 长度, MCU 需要多次写 FIFO 数据。FIFO flag (reg0x12[13])表示 FIFO 是不是为空。



发射时序图



MCU发送数据包流程图

在上面的流程图中，先使能发送状态机，再写发送数据到发送 FIFO 中，这样可以提高 MCU 效率，但要保证数据包在发送 TRAILER 之前写入所有数据；如果要写入的时间需要很长，则应先写好所有数据再启动发送状态。

//////// 发射流程例子 //////////

```

write reg[0x0a] = 0x2053;    //初始化寄存器，即写入需要优化的寄存器值
write reg[0x03] = 0x5810;

write reg[0x26] = 0x8080;    //清空发送 FIFO
write reg[0x27] = 0x0501;    //写 0x05 0x01 0x02 0x03 0x04 0x0 到 FIFO，第一个字节表示长度
write reg[0x27] = 0x0203;
write reg[0x27] = 0x0405;
write reg[0x00] = 0x80XX;    //启动发送使能，同时设置频道，低 7 位为频道号，

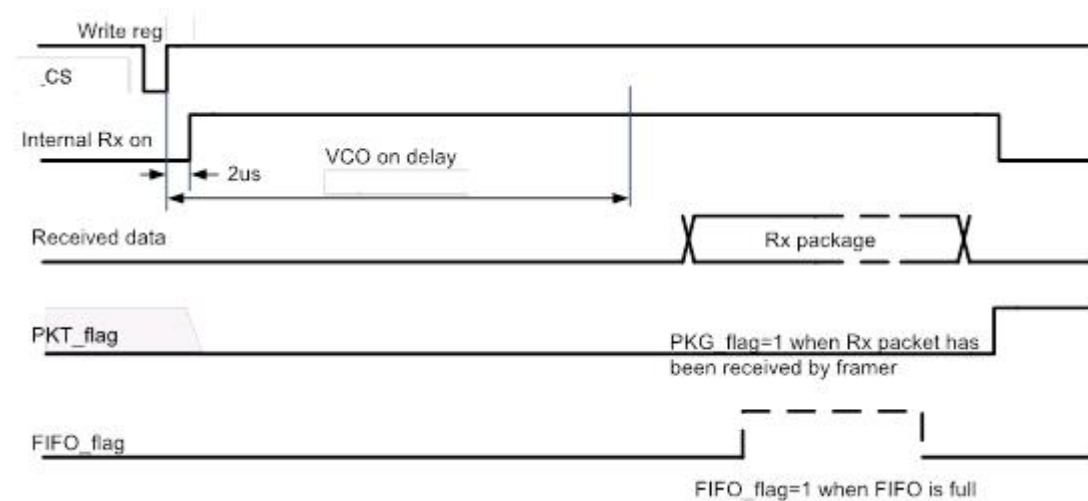
```

While(等待pkt_flag(reg0x16[6])为高，表示发送完成)

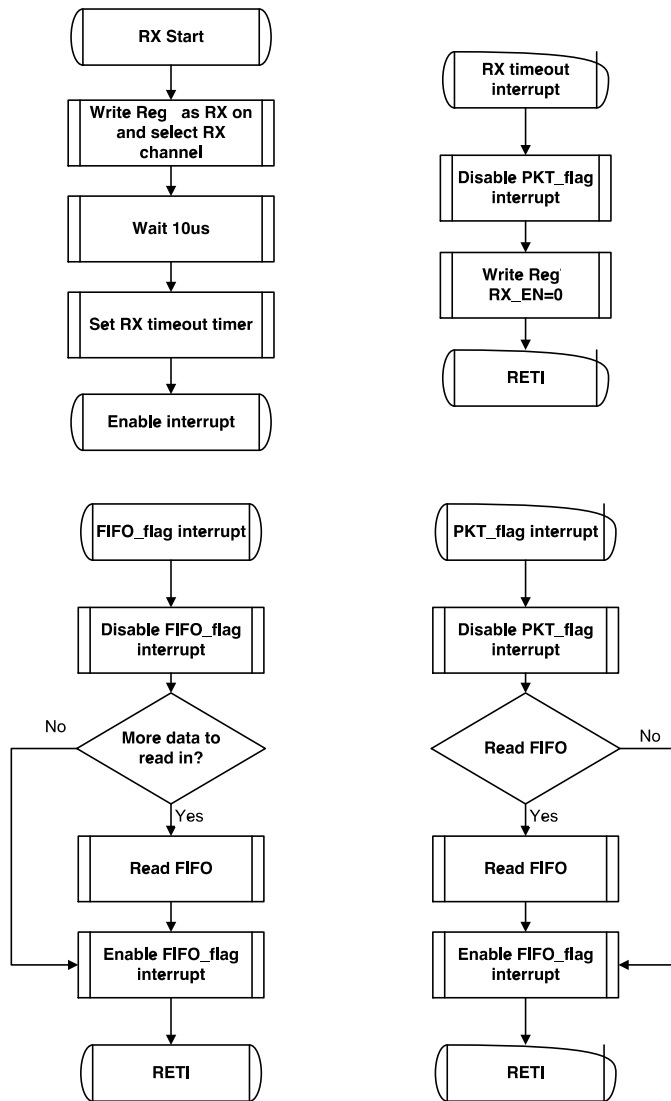
8.6 接收时序

下面显示 RX 接收时序。当 MCU 将寄存器 0x00[14]写为 1 并且选择好接收器信道，芯片将打开 RX 并等待正确的 syncword。当收到正确的 syncword，芯片将自动开始处理数据包。当数据包处理完毕，状态机将进入 IDLE。当接收到的数据包长度长于 63 bytes，FIFO flag 将起作用，意味着 MCU 必须从 FIFO 中读取数据。

在弱信号、多径和远距离时，不一定能收到正确的 syncword。为了避免出现死机情况，MCU 需要做一个定时器。在大多数应用中，数据包是在一定时间窗口内可以收到的，如果没收到系统要有定时器恢复到正常模式。



接收时序图



MCU接收数据包流程图

//////// 接收流程例子 //////////

```

write reg[0x0a] = 0x2053;    //初始化寄存器，即写入需要优化的寄存器值
write reg[0x03] = 0x5810;

```

```

write reg[0x25] = 0x8080;    //清空接收 FIFO
write reg[0x00] = 0x40XX;    //启动接收使能，同时设置频道，低 7 位为频道号

```

```

//等待 pkt_flag( reg0x16[6] )为高，表示接收到一帧数据，开始从接收 FIFO 读数据,读出的第
//一个字节为数据长度

```

```

read reg[0x28]
read reg[0x28].....

```

8.7 自动应答和自动重传

该功能可配置的寄存器有：

写寄存器 0x01[3] = 1，使能 auto-ack 功能。

写寄存器 0x0b[11:8]，设置重发次数，默认值是 3，即第一次发送失败后，再重发 2 次。

写寄存器 0x0b[5:0]，设置等待 ACK 时间，默认是 0x3f，每个 step 是 16us，总共大约为 1ms，表示等待 1ms 时间如果没收到 ACK 包且重发次数没有到最大值就会重新发送数据包。

使能和不使能 AUTO-ACK 功能对应的 PKT_FLAG 的拉起时间也不一样：

发射方收到 ACK 包后或未收到 ACK 包但重发次数已最大，这时会拉起 PKT_FLAG，退出发射状态，MCU 在 PKT_FLAG 拉高后，可读寄存器 0x16[7]位，如果为 1 即收到 ACK 包。

接收方在收到数据包后再发出 ACK 包，然后退出接收状态，拉起 PKT_FLAG 信号。

ACK包的时间约为150us。

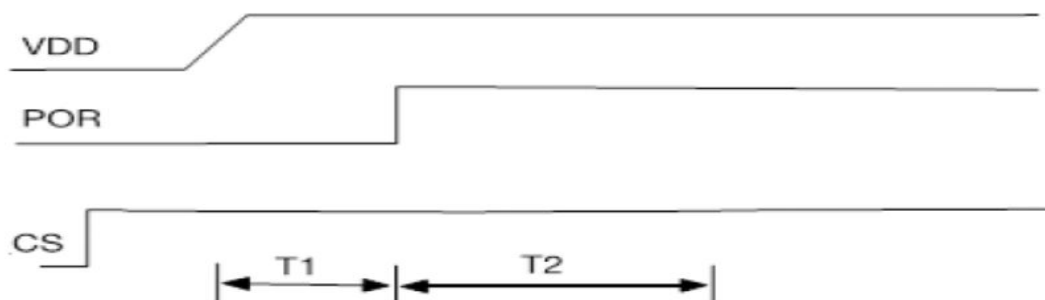
8.8 PA 输出功率设置

PA 的输出功率可以通过 reg0x02 寄存器来设置。最大输出功率可到+7dBm，最小可到-24dBm。下表列出部分功率配置（如果需要其它功率配置，请联系公司技术人员）：

Reg0x02	Pout (dBm)	Ivcc (mA)
0x4060	7	34
0x4061	6	30
0x4065	3	23
0x4067	1	20
0x2020	0	16
0x2061	-2	15
0x2064	-5	13
0x2066	-7	12
0x2068	-10	11.6
0x306A	-13	11
0x307A	-16	9
0x306F	-24	8

9. 注意事项

9.1 上电和寄存器初始化数据



1. T1 时间为上电复位时间，约0.5ms.
2. T2 是晶振稳定时间，约为1.5ms
3. 编程时需要延时0.5ms+1.5ms=2ms 后再对芯片进行操作, 初始化寄存器
4. 软件复位：先写0x1e[0]寄存器为‘1’，使能软件复位功能，再对0x01[7]寄存器写‘1’就可以完成复位操作。

9.2 进入睡眠模式及唤醒

当CS拉高，并且MCU 写寄存器 0x01[15] 为“1”后，芯片进入 sleep mode，此时消耗电流 $<1\mu\text{A}$ ，sleep mode下寄存器的值能够保持。

当CS拉低后，芯片自动唤醒进入IDLE状态。

MCU 要拉低CS一段时间(等待晶体稳定，约1.5ms)，再读写数据。

9.3 RF 电路的PCB板需要注意什么

首先要保证RF芯片的电源稳定，干净。电源上的稳压和去耦合电容尽量靠近芯片的管脚，电源线和地线尽量短。可用大面积的铜层作为地线用，没有用到的地方可以和地线相连作为地线用。如果RF芯片3.3V供电电源的噪声或干扰比较大，建议经过RC或LC滤波之后再给RF芯片供电。

晶振尽量靠近XI和X0脚，连接晶振的线尽量短以减少噪声干扰和分布电容对晶振的影响。晶振外壳应良好接地以减少对外辐射提高抗干扰能力。