


# TI德州仪器无线链接产品数据手册

**CC1101**

声明：此手册由信驰达科技进行翻译整理，旨在更方便快捷的为您提供帮助。译文难免会出现错误及误差，如对此有建议可通过邮件方式

联系我们，非常感谢您的反馈。凡译文手册注明“ RFStar”的作品，均为信驰达（RF-star®）公司合法拥有版权或有权使用的作品，未经本公司授权不得转载、摘编或利用其它方式用于商业用途。已经本公司授权使用作品的，应在授权范围内使用，并注明“信驰达科技提供”。违反上述声明者，本公司将拥有最终解释权。

**Shenzhen RF-star Technology Co., Ltd.**  
**TEL: 0755-86329829 FAX: 0755-86329413**

**<http://www.szrfstar.com>**

## 信驰达简介

信驰达科技 (RF-star) 是一家集合方案设计功能和核心器件供应的专业本地电子元器件分销商, 专注低功耗射频 LPRF 和低功耗 MCU 领域, 公司成立于2010年, 作为中国区唯一具有美国 TI 公司授予的 LPRF Product Reseller 和 Third Party 双重资质的公司, 一直引领着 LPRF 技术在国内的推广和应用, 是国内唯一一家可提供 LPRF 软硬件产品、技术支持、解决方案和核心元器件供应一条龙服务的专业化公司;

公司在美国新泽西州、中国深圳、上海、北京、天津、无锡、长沙、成都、重庆设有研发中心和办事处, 拥有资深的技术研发团队和销售团队以及 SMT 生产工厂。

无线射频器件用于低于1GHz 和2.4GHz 频段、ANT、蓝牙(Bluetooth)、低功耗蓝牙、射频识别 (RFID)、PurePath 无线音频、ZigBee、IEEE802.15.4、Zigbee RF4CE、6LoWPAN、Wi-Fi 的射频集成电路( RF IC ) 和专有协议。

产品市场应用: ZigBee 无线传感网络, 各种数据采集及遥测监控(含数据, 语音, 图像等), 可应用于安防、医疗、能源、水力、电力、交通监控、防盗, 无线自动抄表; 仪器仪表远程数据遥测、工业无线遥控; 消防安全自动报警、煤矿安全监控及人员定位; 汽车防盗、胎压检测, 四轮定位; 无线键盘、鼠标、打印机、游戏杆、遥控玩具、机器人等广泛的领域。适用于合乎全世界免费频段 315MHz、433 MHz、470MHz、868 MHz、915 MHz、2.4GHz, 符合 FCC、CE、SGS、RoHs 认证规范, 产品和信誉受到国内外顾客的一致好评。

RF-star 将一如既往, 为客户提供更多、更好的产品, 更具优势的技术服务, 良好的商务服务, 和更完善的物流服务。RF-star 将跨上一个新的平台, 获得更大的发展空间。RF-star 将继续本着“务实、诚信、学习、创新”的专业精神, 团结一致、奋勇开拓、锐意进取, 为成为全球无线射频技术绝对第一之产品、服务及解决方案提供者, 把科技与客户联系在一起, 为供应链注入动力, 并提供卓越的投资回报而不懈努力。

如果您在产品开发过程中发现技术难题以及高频困扰, 竭诚欢迎来电洽询。我们将为您提供技术支持和解决方案, 让您能更快把产品推向市场。

我们深信射频技术将会得到迅速的发展与普及, 我们愿意分享多年来在射频行业积累的经验与教训, 为无线的明天做出贡献。专业源于专注, 科技铸就未来。

---

## 低成本、低功耗 1GHz 以下无线收发器（CC1101）

---

### 应用

- | 基于 315/433/868/915 MHz ISM/SRD 的极低功耗的无线应用。
- | 无线报警和安全系统
- | 工业监视和控制

### 产品描述

CC1101 是低成本的 1GHz 以下的无线收发器，为极低功耗的无线应用而设计。电路主要设计为 ISM(工业、科学和医疗)和 SRD(短距离设备)，频段在 315、433、868 和 915，但是可以很容易的编程，使之工作在其他频率，在 300-348MHz，387-464 MHz 和 779-928 MHz 频段。

CC1101 是 CC1100 RF 收发器改良以及代码一致的版本。CC1101 的主要改进如下：

- | 改良的伪应答
- | 更好的关闭相位噪声，因而改善相邻信道功耗（ACP）的性能
- | 更高的输入饱和和级别
- | 改善输出功率斜面
- | 扩大工作频段：

CC1100: 400-464 MHz and 800-928MHz

CC1101: 387-464 MHz and 779-928MHz

## 10 4 线串口配置和数据接口

CC1101 通过 4 线 SPI 兼容接口 (SI, SO, SCLK 和 CSn) 进行配置, CC1101 作为从设备。这个接口同事用作读写缓冲器数据。SPI 接口上所有的数据传送都是先传送 MSB。

SPI 接口上的所有传送都是以一个头字节 (header byte) 开始, 包含一个读写位 (R/W), 一个突发(burst access)访问位 (B) 和 6 位地址位 ( $A_5 \sim A_0$ )。

在 SPI 总线上传输数据时, CSn 脚必须保持低电平。如果在发送头字节或者读写寄存器时 CSn 拉高, 传送将被取消。SPI 接口上地址和数据的发送时序图见图 12, 并参考表 19。

当 CSn 被拉低, MCU 在发送头字节之前, 必须等到 CC1101 的 SO 脚变为低电平。这说明晶振开始工作。除非芯片在 SLEEP 或者 XOFF 状态, SO 脚在 CSn 引脚被拉低后马上变为低电平。

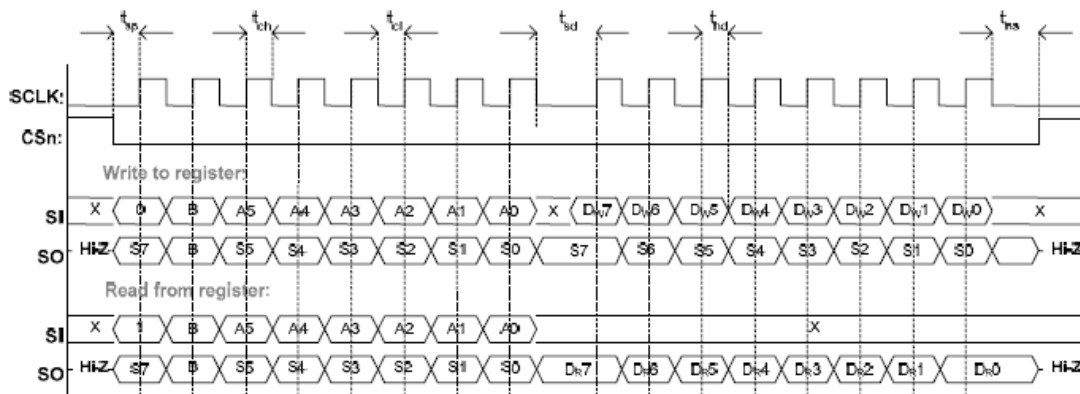


图 12: 配置寄存器读写操作

Parameter	Description	Min	Max	Units	
f <sub>SCLK</sub>	SCLK frequency 100 ns delay inserted between address byte and data byte (single access), or between address and data, and between each data byte (burst access).	-	10	MHz	
	SCLK frequency, single access No delay between address and data byte	-	9		
	SCLK frequency, burst access No delay between address and data byte, or between data bytes	-	6.5		
t <sub>sp,pd</sub>	CSn low to positive edge on SCLK, in power-down mode	150	-	μs	
t <sub>sp</sub>	CSn low to positive edge on SCLK, in active mode	20	-	ns	
t <sub>ch</sub>	Clock high	50	-	ns	
t <sub>cl</sub>	Clock low	50	-	ns	
t <sub>rise</sub>	Clock rise time	-	5	ns	
t <sub>fall</sub>	Clock fall time	-	5	ns	
t <sub>sd</sub>	Setup data (negative SCLK edge) to positive edge on SCLK (t <sub>su</sub> applies between address and data bytes, and between data bytes)	Single access	55	-	ns
		Burst access	76	-	ns
t <sub>hd</sub>	Hold data after positive edge on SCLK	20	-	ns	
t <sub>ns</sub>	Negative edge on SCLK to CSn high.	20	-	ns	

表 19: SPI 接口时序要求

## 10.1 芯片状态字节

当在 SPI 接口上发送头字节，数据字节或者命令选通（command strobe）时，CC1101 在 SO 引脚上发送芯片状态字节。状态字节包含对 MCU 有用的关键状态信号。第 1 位，S7，为 CHIP\_RDYn 信号，在 SCLK 的第一个上升沿之前，该信号必须变为低电平。CHIP\_RDYn 表示晶振已经开始工作。

第 6、5、4 位组成（STATE）状态值，该值反映芯片的状态。在空闲（IDLE）状态，XOSC 和数字核的电源被打开，但是其他模块全部掉电。频率和信道配置只能在芯片处于该状态时被更新。当芯片处于接收模式时，接收（RX）状态被激活。同样，当芯片处于发送模式时，发送（TX）状态被激活。

状态字节的最后 4 位（3: 0）包含 FIFO\_BYTES\_AVAILABLE。在读操作中（头字节的 R/W 位置 1），FIFO\_BYTES\_AVAILABLE 包含从 RX FIFO 可读到的数据字节数。在写操作中（头字节的 R/W 位置 0），FIFO\_BYTES\_AVAILABLE 包含可写入到 TX FIFO 中的字节数。当 FIFO\_BYTES\_AVAILABLE=15，15 个或者更多字节是可读的/空闲的。

表 20 为状态字节概要。

位	名称	描述		
7	CHIP_RDYn	保持高电平，直到电源和晶振稳定。当时用 SPI 接口时必须变为低电平。		
6:4	STATE[2:0]	显示当前主状态机器模式		
		值	状态	描述
		000	IDLE	空闲状态
		001	RX	接收模式
		010	TX	发送模式
		011	FSTXON	快速 TX 准备
		100	CALIBRATE	频率合成器校准运行中
		101	SETTLING	PLL 设置中
		110	RXFIFO_OVE RFLOW	RX FIFO 溢出。读出有用的数据，时用 SFRX 清洗 FIFO
		111	TXFIFO_UND ERFLOW	TX FIFO 下溢，使用 SFTX 命令
3:0	FIFO_BYTES_AVAILABLE[3:0]			

表 20：状态字节概要

## 10.2 寄存器访问

CC1101 的配置寄存器位于 SPI 地址的 0x00 ~ 0x2E。64 页的表 37 列出了所有配置寄存器。推荐使用 SmartRF® Studio 生成合适的寄存器设定值。每个寄存器的详细说明见 67 页的 29.1 和 29.2。所有配置寄存器都是可读写的。当写寄存器时，每次一个头字节或者数据字节在 SI 引脚上传送，一个状态字节都在 SO 引脚上传送。当读寄存器时，每次一个头字节在 SI 引脚上传送时，一个状态字节都在 SO 引脚上传送。

通过设置头字节的 burst 位（B）可以高效的实现寄存器的连续地址访问。地址位（A<sub>5</sub>~A<sub>0</sub>）

在内部地址指针中设置起始地址。指针通过每一个新的字节自动增加（每 8 个时钟脉冲）。突发访问可以是读，也可以是写访问，必须通过将 CS 脚拉高来停止操作。

在 0x30 - 0x3D 的寄存器地址范围，burst 位置 1 选择状态寄存器，而 burst 置 0 选择命令选通（command strobes）。详见 10.3。因此，突发访问不可用于状态寄存器的访问，一次只能访问一个状态寄存器。状态寄存器为只读。

## 10.3 读 SPI

## 10.4 命令选通（command strobes）

命令选通可以看做是 CC1101 的一个单字节指令。通过寻址一个命令选通寄存器，将启动内部序列。这些命令用来禁止晶振，使能接收，使能无线唤醒等。13 个命令选通的清单见 63 页的表 36。

**Note:** 一个 **SIDLE** 命令将清除所有的未定命令，直到达到 **IDLE** 状态。这意味着比如当无线处于接收状态时执行 **SIDLE** 命令，当无线达到 **IDLE** 状态之前执行任何其他的命令选通都是无效的。

命令选通寄存器通过发送单一的头字节来访问（不传送数据字节）。也就是说，只有一个 R/W 位，一个突发访问位（置 0）和 6 位地址位（在 0x30~0x3D 范围内）被传送。R/W 位可以是 1 或者 0，这将决定状态字节中的 FIFO\_BYTES\_AVAILABLE 区域如何确认。

当写命令选通时，状态字节将在 SO 引脚上被发送。

命令选通可以在 CSn 不被拉高的情况下，跟随在任何其他 SPI 访问之后。然而，如果一个 SRES 命令被执行，下一个头字节被执行之前必须等到 SO 引脚被拉低，见图 13。命令选通将立即被执行，除了 SPWD 和 SXOFF 命令在 CSn 拉高之后执行。

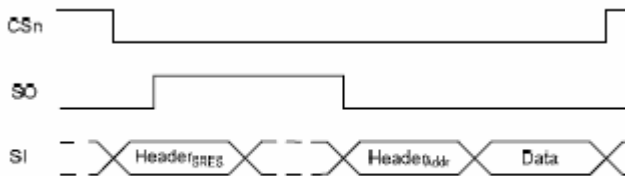


Figure 13: SRES Command Strobe

## 10.5 FIFO 访问

64 字节的 TX FIFO 和 64 字节的 RX FIFO 通过 0x3F 地址访问。当 R/W 位置 0 时，TX FIFO 被访问，当 R/W 位置 1 时 RX FIFO 被访问。TX FIFO 为只写，RX FIFO 为只读。

Burst 位用来确定 FIFO 访问时单字节访问还是突发访问。单字节访问方式是一个 burst 位置 0 的头字节和一个数据字节。数据字节之后，跟随一个新的头字节。因此，CSn 可以保持低电平。突发访问的方式是一个头字节和连续的数据字节，直到将 CSn 拉高来停止访问。

使用下列头字节来访问 FIFO:

Ø 0x3F: 单字节访问 TX FIFO

Ø 0x7F: 突发访问 TX FIFO



Ø 0xBF: 单字节访问 RX FIFO

Ø 0xFF: 突发访问 RX FIFO

当向 TX FIFO 写入数据时, 在每一个新的字节发送时状态字节在 SO 引脚上输出。(图 12) 当向 TX FIFO 写入数据时, 状态字节可以用来检测 TX FIFO 的下溢。

注意状态字节包含的空闲字节数是正在传送到 TX FIFO 的字节写入之前的值。当最后一个可以写入 TX FIFO 的字节在 SI 引脚上传送的时候, 同时在 SO 引脚上接收的状态字节将显示 TX FIFO 中有一个字节的空闲。

TX FIFO 可以通过 SFTX 命令刷新。同样 SFRX 命令可以刷新 RX FIFO。SFTX 和 SFRX 命令只能在 IDLE、TXFIFO\_UNDERFLOW、RXFIFO\_OVERFLOW 状态使用。在进入 SLEEP 状态时, 两个 FIFO 都将被刷新。

## 10.6 PATABLE 访问

0x3E 地址用来访问 PATABLE, 用来选择 PA 功率控制设置。SPI 在接收到该地址后, 传送 8 字节数据。通过编程 PATABLE, 可以完成对 PA 功率的增加和减少的控制, 以及 ASK 调制的修正以减小带宽。参照 samrtRF studio 推荐的修正和 PA 序列。参照 55 页的 Section 24 对 PA 功率输出编程的详细说明。

PATABLE 是一个 8 字节表, 定义 PA 功率控制使用的 8 个字节中的任意值。(通过 FRENDO.PA\_POWER 中的 3 位值选择) 表格可以从最低设置 (0) 到最高的 (7) 读写, 一次一个字节。一个索引指针用来控制访问表格。指针在每次读写表格的一个字节时自动增加, 并且在 CSn 拉高时自动设置为最小值。当达到最高值时自动复位。

依靠 burst 位, 访问 PATABLE 既可以是单字节访问也可以是突发访问。当时用突发访问时, 索引指针自动增加, 当指针达到 7 时, 自动复位为 0。R/W 位控制访问时读操作还是写操作。

注意, 当进入 SLEEP 状态时 PATABLE 的内容将全部消失, 除了第一个字节(index0)。

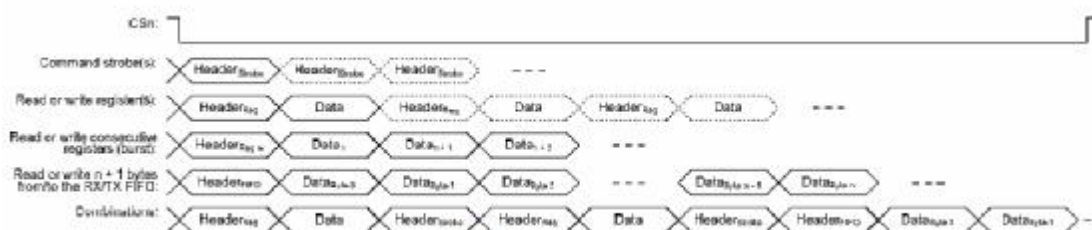


Figure 14: Register Access Types

## 11.MCU 接口和引脚配置

在典型系统中, CC1101 必须连接 MCU。MCU 必须具备:

- Ø 编程 CC1101 到不同的模式。
- Ø 读写数据缓冲器。
- Ø 通过 4 线 SPI 总线 (SI、SO、SCLK 和 CSn) 读取状态信息。

## 11.1 配置接口

MCU 使用 4 个 IO 口作为 SPI 配置接口。(SI、SO、SCLK 和 CSn) SPI 的描述见 28 页的 Section 10。

## 11.2 基本控制和状态脚

CC1101 有 2 个专用配置脚 (GDO0 和 GDO2) 和一个共用脚 (GDO1) 可以向控制软件输出有用的内部状态信息。这些脚可以用来向 MCU 产生中断。

GDO1 和 SPI 的 SO 引脚共用。GDO1/SO 的默认设置时 3 态输出。通过选择任何编程选项，GDO1/S 将变为普通引脚。当 CSn 拉低，该引脚将始终作为普通的 SO 功能引脚。

在同步和异步串行模式，在发送模式下，GDO0 引脚将作为串行 TX 数据输入引脚。

GDO0 也可以用作片上模拟温度传感器。通过外部 ADC 测量 GDO0 脚的电压，可以计算出温度。温度传感器的详细说明见 18 页的 Section 4.7。通过默认的 PTEST 寄存器设置

(0x7F)，如果频率合成器被允许温度传感器的的输出时可用的。(例如：MANCAL, FSTXON, RX 和 TX 状态) 在 IDLE 状态向 PTEST 寄存器写入 0xBF 允许模拟温度传感器是必要的。离开 IDLE 状态之前，必须恢复 PTEST 寄存器为默认值 (0x7F)。

## 11.3 可选的无线控制特性

CC1101 有一个可选的控制方式，通过 SPI 接口的 SI, SCLK 和 CSn 的重新使用。这个特性允许普通的三引脚控制无线的主要状态：SLEEP, IDLE, RX 和 TX。这些附加功能通过 MCSM0.PIN\_CTRL\_EN 配置为来使能。

状态变化遵从以下命令：

- n 如果 CSn 被拉高，SI 和 SCLK 依照表 21 设置好想要的状态。
- n 如果 CSn 被拉低，SI 和 SCLK 的状态被锁定，按照引脚的配置在内部产生一个 command strobe。

只允许在功能实现之后改变状态。例如，如果 SI 和 SCLK 设置为 RX 状态，并且已经由 CSn 触发，RX 将不能再重新启动。当 CSn 拉低，SI 和 SCLK 将作为普通的 SPI 功能。

除了 SPWD 外，所有引脚控制命令将立即执行，SPWD 延时到 CSn 拉高后执行。

CSn	SCLK	SI	Function
1	X	X	Chip unaffected by SCLK/SI
↓	0	0	Generates SPWD strobe
↓	0	1	Generates STX strobe
↓	1	0	Generates SIDLE strobe
↓	1	1	Generates SRX strobe
0	SPI mode	SPI mode	SPI mode (wakes up into IDLE if in SLEEP/XOFF)

Table 21: Optional Pin Control Coding



## 12 数据速率编程

发送或者接收的数据速率通过MDMCFG3.DRATE\_M 和MDMCFG4.DRATE\_E配置寄存器来编程。数据速率由下面的公式来计算。根据公式，数据速率的编程依赖于晶振频率。

$$R_{DATA} = \frac{(256 + DRATE\_M) \cdot 2^{DRATE\_E}}{2^{28}} \cdot f_{XOSC}$$

下面的方面可以计算出适合的数据速率：

$$DRATE\_E = \left\lceil \log_2 \left( \frac{R_{DATA} \cdot 2^{20}}{f_{XOSC}} \right) \right\rceil$$

$$DRATE\_M = \frac{R_{DATA} \cdot 2^{28}}{f_{XOSC} \cdot 2^{DRATE\_E}} - 256$$

如果DRATE\_M接近整数并且变为256，DRATE\_E加1并且DRATE\_M=0。数据速率可以设置为从1.2K~500K，按照表22的最小步值。

Min Data Rate [kBaud]	Typical Data Rate [kBaud]	Max Data Rate [kBaud]	Data rate Step Size [kBaud]
0.8	1.2 / 2.4	3.17	0.0062
3.17	4.8	6.35	0.0124
6.35	9.6	12.7	0.0248
12.7	19.6	25.4	0.0496
25.4	38.4	50.8	0.0992
50.8	76.8	101.6	0.1984
101.6	153.6	203.1	0.3967
203.1	250	406.3	0.7935
406.3	500	500	1.5869

Table 22: Data Rate Step Size

## 13 接收信道滤波带宽

为了区分不同信道带宽，接收信道滤波器是可设置的。MDMCFG4.CHANBW\_E 和MDMCFG4.CHANBW\_M 配置寄存器控制接收信道滤波器带宽，通过晶振频率来衡量。

下面的公式给出寄存器配置和信道滤波带宽之间的关系：

$$BW_{channel} = \frac{f_{XOSC}}{8 \cdot (4 + CHANBW\_M) \cdot 2^{CHANBW\_E}}$$

表 23 列出了 CC1101 支持的信道滤波带宽。

为了达到更好的性能,信道滤波带宽必须设置以至于信号带宽至多占 80%的信道滤波带宽。对于晶振精度的信道中心公差应该从信号带宽中减去。下面举例说明:

信道滤波带宽设置为 500kHz, 信号带宽保持在 500kHz 的 80%之内, 400kHz。发送和接收设备都在中心频率 915MHz 和  $\pm 20\text{ppm}$  频率误差, 总共的频率误差是 915MHz 的  $\pm 40\text{ppm}$ , 即  $\pm 37\text{kHz}$ 。如果整个发送的信号带宽达到接收的 400kHz 之内, 发送信号带宽的最大值为  $400\text{kHz} - 2 \times 37\text{kHz}$ , 即 326kHz。

通过修正发送和接收的频率偏移, 滤波带宽可以减小, 并且可以改善灵敏度。详见 DN005[20]和 Section14.1 。

MDMCFG4.	MDMCFG4.CHANBW_E			
CHANBW_M	00	01	10	11
00	812	406	203	102
01	650	325	162	81
10	541	270	135	68
11	464	232	116	58

**Table 23: Channel Filter Bandwidths [kHz]**  
(assuming a 26 MHz crystal)

## 14 解调器、符号同步器和数据判定

CC1101 包含一个高级的和高可配置的解调器。信道滤波和频率偏移补偿都可以数字化。

### 14.1 频率偏移补偿

CC1101 有非常好的频率分辨率。(表 12) 这个特性可以用来补偿频率偏移和温漂。

当时用 2-FSK、GFSK 或 MSK 调制时, 解调器将在发送和接收频率的一定界限内, 通过估计接收数据的中心, 补偿频率偏移。频率偏移补偿配置由 FOCCFG 寄存器控制。通过补偿发送和接收之间最大的频率偏移, 可以改善灵敏度, 见 DN005。

运算法则的跟踪范围是通过 FOCCFG.FOC\_LIMIT 配置寄存器选择一部分信道带宽。

如果FOCCFG.FOC\_BS\_CS\_GATE位设置, 频率补偿将被冻结, 直到carrier sense生效。这对于无线常时间处于RX状态而没有通信是非常有用的, since the algorithm may drift to the boundaries when trying to track noise。

追踪循环有两个增益要素, 影响设定时间和运算的噪声灵敏度。同步字节被检测到之前由FOCCFG.FOC\_PRE\_K设定增益, 同步字节检测到之后由FOCCFG.FOC\_POST\_K选择增益。

注意: ASK和OOK调制不支持频率偏移补偿。

在FREQUEST状态寄存器中的预置频率偏移值是可用的。可以用作永久的频率偏移补偿。通过把FREQUEST中的值写入FSCTRL0.FREQOFF, 平率合成器将依照预置频率偏移自动被调整。更详细的关于永久频率补偿计算参照DN015[13]。

## 14.2 位同步

## 14.3 字节同步

字节同步通过连续的同步字查询实现。同步字是一个 16 位配置区（可重复得到 32 位），在发送模式下通过调制器自动插入到包的起始。在同步字中 MSB 首先发送。解调器使用这个区域寻找比特流中的字节边界。同步字也有系统标识符的功能，因此如果 RX 中的同步字检测在寄存器 MDMCFG2 中被使能（Section 17.1），只有正确的预先确定的同步字的包才接收。同步字检测可以关联用户配置的 16 或者 32 位同步字。关联端可以设置为 15/16、16/16 或 30/32 位匹配。同步字能更好的使用前导质量指示机制的描述和/或载波监听条件。同步字通过 SYNC1 和 SYNC0 寄存器配置。

为了确保同步字检测的错误更少，一个前导质量指示（PQI）机制被用来仙子同步字。为确保同步字被接收，必须超过前导质量的门限值。详见 42 页 Section。

## 15 数据包处理的硬件支持

CC1101 有内置的硬件支持作为数据包导向的无线协议。

在发送模式，数据包的处理者可以配置增加下面的元素到 TX FIFO 中的数据包存储：

- n 一个可编程数量的前导字节。
- n 两字节同步字。可复制为 4 字节同步字。只插入前导字节或者同步字节是不可能的。
- n 一个覆盖数据区域的 CRC 校验计算。

推荐设置 4 字节前导和 4 字节同步字，除了 500k 波特率时推荐使用 8 字节前导。另外，在数据区域和附加的 2 字节 CRC 校验，以下可以被执行：

- n 使用 PN9 序列 Whitening 数据。
- n 通过使用交叉和编码数据（卷积编码）前向纠错。

在接收模式，数据包处理支持将是解构数据包，（如使能）以下可以被执行：

- n 前导检测
- n 同步字检测
- n CRC 计算和 CRC 检查
- n 一个字节地址检测
- n 包长度检查（根据设置的最大长度检查长度字节）
- n 反 Whitening
- n 反交叉和解码

另外，RSSI（接收信号强度指示）、LQI（链路质量指示）和 CRC 状态的两个状态字节可以附加在 RX FIFO。

Bit	Field Name	Description
7:0	RSSI	RSSI value

**Table 24: Received Packet Status Byte 1  
(first byte appended after the data)**

Bit	Field Name	Description
7	CRC_OK	1: CRC for received data OK (or CRC disabled) 0: CRC error in received data
6:0	LQI	Indicating the link quality

**Table 25: Received Packet Status Byte 2  
(second byte appended after the data)**

注意：控制数据包处理特性的寄存器区只能再 CC1101 处于 IDLE 状态时被修改。

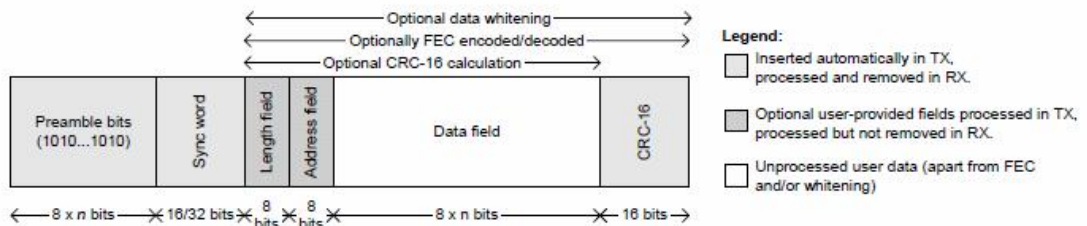
## 15.1 Data Whitening

将 PKTCTRL0.WHITE\_DATA 置 1，除了前导和同步字外，所有的数据在发送前将 XOR 一个 9 位的伪随机序列（PN9）。在接收结束后，数据将使用相同的伪随机序列 XOR。PN9 序列初始化全部为 1。

## 15.2 数据包格式

数据包的格式可以配置为由以下部分组成：

- n 前导
- n 同步字
- n 可选的长度字节
- n 可选的地址字节
- n 净荷
- n 可选的 2 字节 CRC



**Figure 16: Packet Format**

前导的格式是一个交叉的 1 和 0 的序列（10101010……）。前导的最小长度可以通过 MDMCFG1.NUM\_PREAMBLE 的值来设置。当使能发送，调制器将开始发送前导。当发送完编程数量的前导子节后，调制器开始发送同步字和 TX FIFO 中的可用数据。如果 TX FIFO 是空的，调制器将继续发送前导字节，直到第一个字节写入到 TX FIFO。调制器将接着发送同步字和数据字节。

同步字是在 SYNC1 和 SYNC0 寄存器中设置的 2 字节数据。同步字节提供了引入数据包的字同步。一个字节的同步字可以复制，通过设定 SYNC1 的值到前导格式。也可以通过设定 MDMCFG2.SYNC\_MODE 的值为 3 或 7 将同步字复制为 32 位。同步字将被复制 2 次。

CC1101 支持定长数据包协议和变长数据包协议。可变的或固定数据包长度模式可用于最长 255 字节数据包长度。再长的数据包，就必须使用无限数据包长度模式。

固定数据包长度模式通过设定 `PKTCTRL0.LENGTH_CONFIG=0` 来选择。期望的数据包长度通过 `PKTLEN` 寄存器来设置。

在可变数据包长度模式，`PKTCTRL0.LENGTH_CONFIG=1`，数据包长度由同步字后面的第一个字节配置。数据包长度定义为有效载荷，不包括长度字节和可选的 CRC。`PKTLEN` 用来配置允许接收的最大数据包长度。任何接收的数据包长度大于 `PKTLEN` 的值都将被丢弃。

`PKTCTRL0.LENGTH_CONFIG=2`，数据包长度将设置为无限长，发送和接收将持续到手动关闭。在下一部分描述中，将使用不同长度的配置来支持数据包格式。必须确保在发送前半个或多个字节的时候 TX 模式不被关闭。参考 CC1101 的勘误表。

## 15.2.1 任意长度区域配置

数据包长度寄存器 `PKTLEN`，在接收和发送时可以重新编程。结合固定数据包长度模式（`PKTCTRL0.LENGTH_CONFIG=0`），这使与变长数据包不同的长度配置成为可能。（在可变数据包长度模式下长度字节是同步字节后第一个字节）在接收开始，数据包长度设置为一个大的值。MCU 读取足够多的字节来解析数据包中的长度区域。然后依照这个值对 `PKTLEN` 进行设置。当数据包处理器中的字节指针等于 `PKTLEN` 寄存器的值时数据包结束。因此，MCU 必须在内部指针达到数据包长度之前确认正确的长度。

## 15.2.2 数据包长度大于 255

数据包自动控制寄存器 `PKTCTRL0`，可以在 TX 和 RX 期间重新编程。这就使发送和接收数据包长度超过 256 字节并且仍然使用数据包处理器硬件支持成为可能。在数据包开始时，必须激活无限数据包长度模式（`PKTCTRL0.LENGTH_CONFIG=2`）。在发送端，`PKTLEN` 寄存器设置为模（长度，256）。在接收端，MCU 读出足够长度的字节来解析出数据包长度并设置 `PKTLEN` 寄存器为模（长度，256）。当数据包中还有少于 256 字节数据，MCU 禁止数据包无限长度模式并且激活固定长度模式。当内部字节指针达到 `PKTLEN` 时，发送和接收结束。（无线进入的状态由 `TXOFF_MODE` 或 `RXOFF_MODE` 决定）自动 CRC 添加/校验也可以使用（通过设置 `PKTCTRL0.CRC_EN=1`）。

例如当发送一个 600 字节的数据包时，MCU 必须按照以下执行：

- | 设置 `PKTCTRL0.LENGTH_CONFIG=2`。
- | 重新设置 `PKTLEN` 寄存器模（600，256）=88。
- | 至少发送 345 字节（600-255），例如通过填充 64 字节 TX FIFO 6 次（发送 384 字节）。
- | 设置 `PKTCTRL0.LENGTH_CONFIG=0`。
- | 当数据包指针达到 88 发送完成。一共 600 字节被发送。



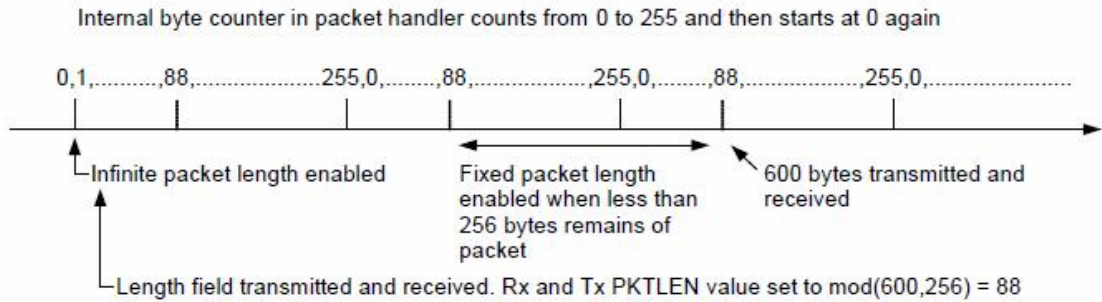


Figure 17: Packet Length &gt; 255

## 15.3 接收模式下的数据过滤

CC1101 支持三种不同类型的数据包过滤：地址过滤，最大长度过滤和 CRC 过滤。

### 15.3.1 地址过滤

设置 `PKTCTRL1.ADR_CHK` 为任意大于 0 的值使能数据包地址过滤。数据包处理器启动后将比较数据包中的目标地址字节和在 `ADDR` 寄存器中设置的节点地址或者当 `PKTCTRL1.ADR_CHK=10` 时的广播地址 `0x00` 以及 `PKTCTRL1.ADR_CHK=11` 时的广播地址 `0x00` 和 `0xFF`。如果接收的地址匹配到一个正确地址，数据包被接收并写入到 `RX FIFO`。如果地址匹配失败，数据包将被丢弃，重新启动接收模式（不管 `MCSM1.RXOFF_MODE` 的设置）。

如果当时用无限数据包长度模式和使能地址过滤时接收地址匹配到正确的地址，`0xFF` 将被写入到 `RX FIFO`，其次是地址字节和有效载荷。

### 15.3.2 最大长度过滤

在可变数据包模式，`PKTCTRL0.LENGTH_CONFIG=1`，`PKTLEN.PACKET_LENGTH` 寄存器用来设置最大允许的数据包长度。如果接收的长度字节大于这个值，数据包将被丢弃，重新启动接收模式（不管 `MCSM1.RXOFF_MODE` 的设置）。

### 15.3.3 CRC 过滤

当 CRC 校验失败时过滤数据包，通过设置 `PKTCTRL1.CRC_AUTOFLUSH=1` 使能。CRC 自动清除功能将在 CRC 校验失败时清空所有 `RX FIFO`。在自动清空 `RX FIFO` 后，下一个状态依赖于 `MCSM1.RXOFF_MODE` 的设置。

当使用自动清除功能，在可变数据包长度模式下最大数据包长度为 63 字节，在定长数据包长度模式下是 64 字节。注意当 `PKTCTRL1.APPEND_STATUS` 使能，最大允许数据包长度将减少 2 个字节，以确保在数据包的结尾附加的 2 字节的状态字节在 `RX FIFO` 中占用的空间。当 CRC 校验错误以后全部 `RX FIFO` 被清空，在接收正确的数据包前之前接收到的数据包必须从 `FIFO` 中读出。直到 CRC 校验正确 MCU 才从正确的数据包中读取数据。



## 15.4 发送模式的数据包处理

需要发送的有效载荷必须写入到TX FIFO。如果使能可变数据包长度第一个字节必须写长度字节。长度字节的值等于数据包的有效载荷（包含可选的地址字节）。如果在接收端使能地址识别，写入到TX FIFO中的第二个字节必须是地址字节。

如果使能固定数据包长度，写入到TX FIFO中的第一个字节必须是地址字节（接收端使用地址识别）。

调制器将首先发送预置数量的前导字节。如果TX FIFO中有可发送的数据，调制器将发送2字节（可选4字节）同步字，紧接着是TX FIFO中的有效载荷。如果使能CRC，校验时计算TX FIFO中的所有数据，然后跟随有效数据发送2个附加字节。若果在完整的数据包发送完成之前TX FIFO变为空的，无线将进入TXFIFO\_UNDERFLOW状态。退出该状态的唯一方法是运行SFTX命令。

发送下溢后再向TX FIFO写入数据不能重启TX模式。

如果whitening被使能，同步字后面的所有字节将被whitened。这必须在FEC/Interleaver之前完成。Whitening通过设置PKTCTRL0.WHITE\_DATA=1使能。

如果FEC/Interleaving被使能，同步字后面的所有数据在调制之前将被interleaver 和FEC编码加密。FEC通过设置MDMCFG1.FEC\_EN=1使能。

## 15.5 接收模式的数据包处理

在接收模式，解调器和数据包处理器将寻找一个有效的前导和同步字。当找到后，解调器将获得位和字节同步，并将接收第一个有效字节。

如果使能FEC/Interleaver，FEC解码器将开始解码第一个有效字节。Interleaver将在数据被其他任何处理之前解密。

如果使能whitening，在此基础上数据将被de-whitening。

当使能可变数据包长度模式时，第一个字节是长度字节。数据包处理器储存该值为数据包长度并且通过该长度字节指示接收的字节数量。如果使用固定数据包长度模式，数据包处理器将接收预置数量的字节。

接下来，数据包处理器可选的检测地址并且只有地址匹配才继续接收。如果使能自动CRC校验，数据包处理器计算CRC并与CRC校验结果相匹配。

在有效载荷的结尾，数据包处理器将可选的写入两个复交数据包状态字节，包括CRC状态、LQI和RSSI。

## 15.6 固件中的数据包处理

如果在固件中执行一个数据包导向的无线协议，MCU需要知道一个数据包什么时候收到/发送。加之，超过64字节长度的数据包，当接收时需要读取RX FIFO和当发送时TX FIFO需要重新填充。这就意味着MCU必须知道有多少个字节可以从RX FIFO中读取或者可以向TX FIFO中写入。有两种方法可以获得必要的状态信息。

### a) 中断驱动方法

当同步字接收/发送完成或者完整的数据包接收/发送通过设置IOCFGx.GDOx\_CFG=0x06，GDO脚可以用在RX和TX中产生一个中断。此外，

IOCFGx.GDOx\_CFG还有两个配置用来产生中断，源于有多少字节在RX FIFO或TX FIFO。IOCFGx.GDOx\_CFG=0x00和IOCFGx.GDOx\_CFG=0x01为RX FIFO，IOCFGx.GDOx\_CFG=0x02和IOCFGx.GDOx\_CFG=0x03为TX FIFO。更多信息查看表35。

#### b) SPI轮询

PKTSTATUS可以使用给定的速率查询以获得GDO0和GDO2当前值。RXBYTES 和 TXBYTES可以使用给定的速率查询以获得RX FIFO或者TX FIFO中的字节数。合二为一的，RX FIFO和TX FIFO中的字节数量可以从每次头字节、数据字节或命令选通在SPI总线上发送时MISO引脚返回的芯片状态字节读取。

推荐使用中断驱动方法因为高速率SPI轮询将减小RX的灵敏度。而且，在Section10.3和CC1101勘误表【4】中，当使用SPI轮询，单一读取PKTSTATUS、RXBYTES和TXBYTES有一个小的有限的错误的的可能性。当读取芯片状态字节有同样的问题。

关于SW的例子【9】和【10】参考TI网站。

## 16 调制方式

CC1101支持振幅、频率和相位移动调制。预期的调制方式在MDMCFG2.MOD\_FORMAT寄存器中设置。

另外，数据流可以通过调制进行曼彻斯特编码和解调器解码。该选项通过设置MDMCFG2.MANCHESTER\_EN=1使能。

注意：曼彻斯特编码不支持同事使用FEC/Interleaver选项或者当使用MSK调制时。

### 16.1 频移键控（FSK）

CC1101可以使用高斯型2-FSK（GFSK）。2-FSK信号通过BT=1的高斯滤波器，产生一个GFSK调制信号。这种频谱修整特性改善邻道功率和占用带宽。

在使用生硬的频移键控的真正的2-FSK系统，频谱是固有的宽阔。通过利用频移键控软化（softer），频谱可以显著的变窄。因此，使用GFSK在相同的带宽可以传输更高的数据速率。

当使用FSK/GFSK调制时，DEVIATN寄存器指定在接收中预期的引入信号频率背离并且在必须与TX背离相同。

频率背离通过DEVIATN寄存器中的DEVIATION\_M 和 DEVIATION\_E设置。该值时指数/尾数形式，由如下公式计算结果：

$$f_{dev} = \frac{f_{xosc}}{2^{17}} \cdot (8 + DEVIATION\_M) \cdot 2^{DEVIATION\_E}$$

Format	Symbol	Coding
2-FSK/GFSK	'0'	– Deviation
	'1'	+ Deviation

**Table 26: Symbol Encoding for 2-FSK/GFSK Modulation**

## 16.2 最小转换键控 (MSK)

当使用MSK，完整的传输（前导，同步字和有效载荷）将是MSK调制。

相移使用恒定的传输时间来完成。部分的信号周期可以通过改变相位来改善，通过DEVIATN.DEVIATION\_M来设置。

这相当于改变信号的波形。使用MSK时DEVIATN寄存器的设定在RX中不受影响。

当使用MSK，曼彻斯特编码/解码必须通过设定MDMCFG2来禁止。

## 16.3 振幅调制

CC1101支持两种不同模式的振幅调制：OOK和ASK。

OOK调制简单的转换PA开或关来分别调制1和0。

通过CC1101的使能ASK可编程调制深度（1和0之间的差异），和脉冲振幅的修整。脉冲修整产生一个更宽的带宽强制输出频谱。

当使用OOK/ASK，来自SmartRF Studio的AGC设置首选的FSK/MSK设置不是适宜的。DN022指导中有怎样从SmartRF Studio的首选设置中查找适宜的OOK/ASK设置。当使用OOK/ASK，DEVIATN的设置不影响TX和RX。

## 17 接收信号限定和链路质量信息

CC1101有几种可以用来增加正确的同步字被检测的几率的限定：

- I 同步字限定
- I PQT（前导质量门限）
- I RSSI（接收信号强度指示）
- I CS（载波监听）
- I CCA（净信道评估）
- I LQI（链路质量指示）

### 17.1 同步字限定

如果在MDMCFG2寄存器中使能接收同步字检测，在检测到有效的同步字之前CC1101不填充RX FIFO并且执行数据包过滤功能。同步字限定模式通过MDMCFG2.SYNC\_MODE设置，见表27的说明。表27中的载波监听的描述见Section17.4。

MDMCFG2. SYNC_MODE	Sync Word Qualifier Mode
000	No preamble/sync
001	15/16 sync word bits detected
010	16/16 sync word bits detected
011	30/32 sync word bits detected
100	No preamble/sync + carrier sense above threshold
101	15/16 + carrier sense above threshold
110	16/16 + carrier sense above threshold
111	30/32 + carrier sense above threshold

**Table 27: Sync Word Qualifier Mode**

## 17.2 前导质量门限（PQT）

在同步字接收之前必须一个质量高于编程门限的前导。

PQT的另一个应用时作为可选的RX终止时间的一个限定。见Section。

前导质量评估器通过每一次接收的不同于前一位的位增加内部计数器，通过每次接收的与前一位相同的位来减小。门限值通过PKTCTRL1.PQT寄存器配置。该计数器的4·PQT门限用来检测同步字。通过设置为0值，禁止同步字的前导质量限定。

GDO中的一个引脚通过设定IOCFGx.GDOx\_CFG=8可以用来检测“前导质量范围”信号。通过检测PKTSTATUS寄存器的PQT\_REACHED位也是可以确定前导质量的范围。当接收的信号超过PQT该信号/位assert。

## 17.3 RSSI

RSSI的值时在选择的通道中信号功率级别的估计。这个值基于RX中当前增益的设置和信道中的标准信号等级。

在RX模式，RSSI可以连续的从RSSI状态寄存器读取知道解调器检测到一个同步字（当同步字检测使能）。此时RSSI读取的值将被冻结知道芯片下一次进入RX状态。

注意：从无线进入RX模式到在RSSI寄存器检测到一个正确的RSSI值需要一定的时间。查看DN505[15]获得RSSI响应时间估算的详细信息。

RSSI的值以dBm和1/2 dB给出。RSSI更新速率， $f_{RSSI}$  依赖于接收滤波带宽和AGCCTRL0.FILTER\_LENGTH。

$$f_{RSSI} = \frac{2 \cdot BW_{channel}}{8 \cdot 2^{FILTER\_LENGTH}}$$

如果使能PKTCTRL1.APPEND\_STATUS，数据包最后的RSSI值自动添加到有效载荷后面的第一个附加字节。

从RSSI状态寄存器读出的RSSI值时2个补足的数字。下面的程序可以将RSSI转换为完整的功率级别（RSSI\_dBm）。

- 1) 读取RSSI状态寄存器
- 2) 将读到的数据从16进制数转换到二进制数（RSSI\_dec）
- 3) 如果 $RSSI\_dec \geq 128$ ， $RSSI\_dBm = (RSSI\_dec - 256)/2 - RSSI\_offset$
- 4) 如果 $RSSI\_dec < 128$ ， $RSSI\_dBm = RSSI\_dec / 2 - RSSI\_offset$

表28给出了RSSI\_offset的典型值。表18和19给出了不同数据速率下作为输入功率的功能的RSSI典型图。

Data rate [kBaude]	RSSI_offset [dB], 433 MHz	RSSI_offset [dB], 868 MHz
1.2	74	74
38.4	74	74
250	74	74
500	74	74

Table 28: Typical RSSI\_offset Values

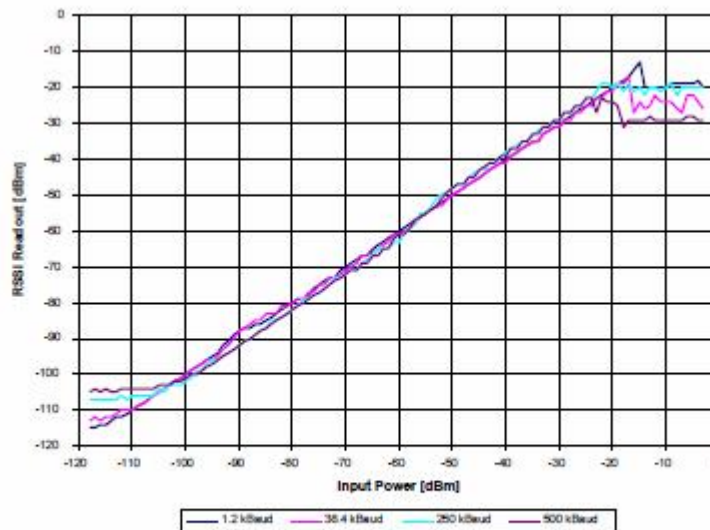


Figure 18: Typical RSSI Value vs. Input Power Level for Different Data Rates at 433 MHz



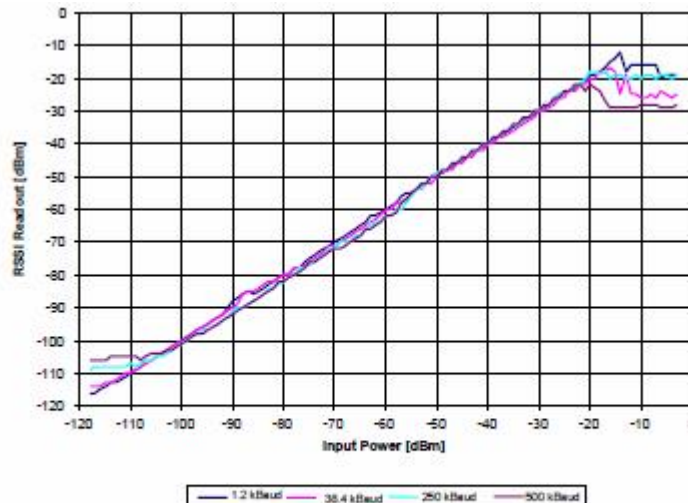


Figure 19: Typical RSSI Value vs. Input Power Level for Different Data Rates at 868 MHz

## 17.4 载波监听（CS）

CS用来作为同步字限定和净信道评估。CS可以基于以下两种可分别调整的条件来声明：

- I 当RSSI在编程的绝对门限之上CS assert，当RSSI在同一个门限之下CS de-assert。
- I 当RSSI以编程数量的dB从一个RSSI到下一个增加的时候CS assert，当RSSI以同一个数量减少的时候CS de-assert。这个设定不依靠绝对的信号等级，因此对于经常编号的噪声环境中检测信号时非常有用的。

CS可以用作同步字限定，需要信号等级高于门限值去执行同步字搜索，通过设定MDMCFG2来设置。CS信号可以通过设定IOCFGx.GDOx\_CFG=14使用GDO的一个脚来检测，在状态寄存器的PKTSTATUS.CS位。

CS的其他应用包括TX-if-CCA功能(Section17.5)和可选的快速RX终止(Section19.7)。

CS还可以用来避免与其他ISM频段的RF信号源冲突。

### 17.4.1 CS 绝对门限

绝对门限值关系到RSSI的值，依赖于以下寄存器：

- AGCCTRL2.MAX\_LNA\_GAIN
- AGCCTRL2.MAX\_DVGA\_GAIN
- AGCCTRL1.CARRIER\_SENSE\_ABS\_THR
- AGCCTRL2.MAGN\_TARGET

通过设定AGCCTRL2.MAX\_LNA\_GAIN 和AGCCTRL2.MAX\_DVGA\_GAIN, 绝对门限值可以使用CARRIER\_SENSE\_ABS\_THR 以1dB的步进调整±7dB。

MAGN\_TARGET的设定折中于blocker tolerance/selectivity和灵敏度之间。该值设定信道到解调器中的信号等级。增加该值减少blockers的空间，因此关闭选择。强烈推荐使用SmartRF Studio产生正确的MAGN\_TARGET配置。表29和表30显示了CS门限在2.4kBuad到250kBuad速率时的典型RSSI读出值。默认CARRIER\_SENSE\_ABS\_THR=0 (0 dB) 和



MAGN\_TARGET=3 (33 dB)。在其他波特率下，用户必须生成相似的表格来查询CS绝对门限值。

		MAX_DVGA_GAIN[1:0]			
		00	01	10	11
MAX_LNA_GAIN[2:0]	000	-97.5	-91.5	-85.5	-79.5
	001	-94	-88	-82.5	-76
	010	-90.5	-84.5	-78.5	-72.5
	011	-88	-82.5	-76.5	-70.5
	100	-85.5	-80	-73.5	-68
	101	-84	-78	-72	-66
	110	-82	-76	-70	-64
	111	-79	-73.5	-67	-61

**Table 29: Typical RSSI Value in dBm at CS Threshold with Default MAGN\_TARGET at 2.4 kBaud, 868 MHz**

		MAX_DVGA_GAIN[1:0]			
		00	01	10	11
MAX_LNA_GAIN[2:0]	000	-90.5	-84.5	-78.5	-72.5
	001	-88	-82	-76	-70
	010	-84.5	-78.5	-72	-66
	011	-82.5	-76.5	-70	-64
	100	-80.5	-74.5	-68	-62
	101	-78	-72	-66	-60
	110	-76.5	-70	-64	-58
	111	-74.5	-68	-62	-56

**Table 30: Typical RSSI Value in dBm at CS Threshold with Default MAGN\_TARGET at 250 kBaud, 868 MHz**

如果门限值设置高了，只有强的信号才能通过，门限值向上调整必须首先减小MAX\_LNA\_GAIN值，接着是MAX\_DVGA\_GAIN值。这样可以降低接收后的功耗，避免太高的增益设定。

## 17.4.2 CS 相对门限

相对门限检测标准信号水平中的突变。这个设定不依赖于绝对信号水平，因此适用于检测定时变化的噪声环境中的信号。寄存器AGCCTRL1.CARRIER\_SENSE\_REL\_THR用来使能/禁止相对CS，并用来选择门限的6dB, 10dB或者14dB的RSSI变化。

## 17.5 净信道评估（CCA）

CCA用来显示当前信道是空闲还是繁忙。通过设定IOCFGx.GDOx\_CFG=0x09，当前的CCA状态在任何一个GDO引脚都可以查看。

MCSM1.CCA\_MODE用来选择当确定CCA时使用的模式。

当CC1101在RX状态时给出STX 或 SFSTXON命令选通，仅能在净信道条件完成才能进入TX或FSTXON状态。此外，芯片将仍处于RX状态。如果信道接下来变为可用的，在一个新的命令选通通过SPI接口发送之前，无线不会进入TX或FSTXON状态。这个特性叫做TX-if-CCA。可以编程设置4种CCA条件：

- l 始终（禁止CCA，总是进入TX）
- l 如果RSSI低于门限值
- l 除非正常接收一个数据包
- l 以上两个（RSSI低于门限值和没有正常接收一个数据包）

## 17.4 链路质量指示（LQI）

信号链路质量指示是一个公制的接收信号的当前质量。如果使能

PKTCTRL1.APPEND\_STATUS, 这个值自动添加到有线载荷后面的最后一个字节里。这个值也可以从LQI状态寄存器中读取。LQI gives an estimate of how easily a received signal can be demodulated by accumulating the magnitude of the error between ideal constellations and the received signal over the 64 symbols immediately following the sync word. LQI也可以用来作为链路质量的相对测量（高的值好于低的值的链路），该值依赖于调制模式。

## 18 使用交叉前向纠错

### 18.1 前向纠错（FEC）

CC1101内部支持前向纠错。使能这个选项，设置MDMCFG1.FEC\_EN=1。**FEC仅支持固定数据包长度模式**，即当PKTCTRL0.LENGTH\_CONFIG=0。FEC被用在数据区金额CRC字来减少总的位误差率，当工作在灵敏度极限附近时。冗余被以这样的方法加入到发送的数据，在接收中可以从存在位错误的数据中回复原始数据。

FEC的使用可以在很低的信噪比（SNR）下正确接收，因此在接收带宽不变的情况下可以扩大通信范围。给定的SNR，使用FEC减少位错误率（BER）。包错误率（PER）与BER的关系：

$$PER = 1 - (1 - BER)^{packet\_length}$$

较低的BER可以允许较长的数据包，或者给定长度的高的数据包百分比，更能传送成功。最后，在实际的ISM无线环境中，暂时的和随时间变化的现象将会产生偶然的错误，在其他号的接收条件下。FEC能够掩盖这种错误，并且与编码数据的交叉相组合，在不好的接收条件下可以可以保持长周期的正确接收（突发错误）。

CC1101的FEC设计采用回旋编码，基于k个输入位产生n位和m个最近的输入位形成的编码流，在每个编码状态之间可以经受一定数量的位错误（m位窗口）。

回旋编码器强制长度为m=4的1/2编码速率。编码器编码一个输入位产生2个输出位；因此有效数据速率减半。这就意味着当使用FEC想要传送相同的有效数据速率，使用两倍的无线数据速率时必要的。这将需要更高的接收带宽，因此降低了灵敏度。换句话说，使用FEC改善接收，更高的接收带宽将成为降低灵敏度的阻碍因数。

### 18.2 交叉

通过无线信道接收的数据经常产生冲突错误和随时间变化的信号强度。为了增加应对多位的错误，当FEC使能时使用交叉。反交叉之后，在接收数据流中的连续范围内的错误将会变成单一的错误伸展开。

CC1101使用矩阵交叉，见图20。片上交叉和反交叉缓存是4\*4的。在发送器中，回旋编码器中的1/2速率的数据位被写到矩阵的行，但是被传送的位序列是从矩阵的列读出的。相反的，在接收器中，接收的符号被写入到矩阵的行，传送到回旋解码器的数据时从矩阵的列中读取的。

当FEC合交叉使用时，至少有一个附加的字节作为格子终止。加起来，在空气中发送的数据数量必须是交叉缓冲器的倍数（2字节）。数据包控制部件在数据包的结尾自动插入1或2个附加字节，以至于被交叉的数据数量是偶数。注意这些附加字节对于用户是看不到的，

在接收数据包进入RX FIFO之前别移除。

当FEC和交叉使用时，最小的数据有效载荷是2字节。

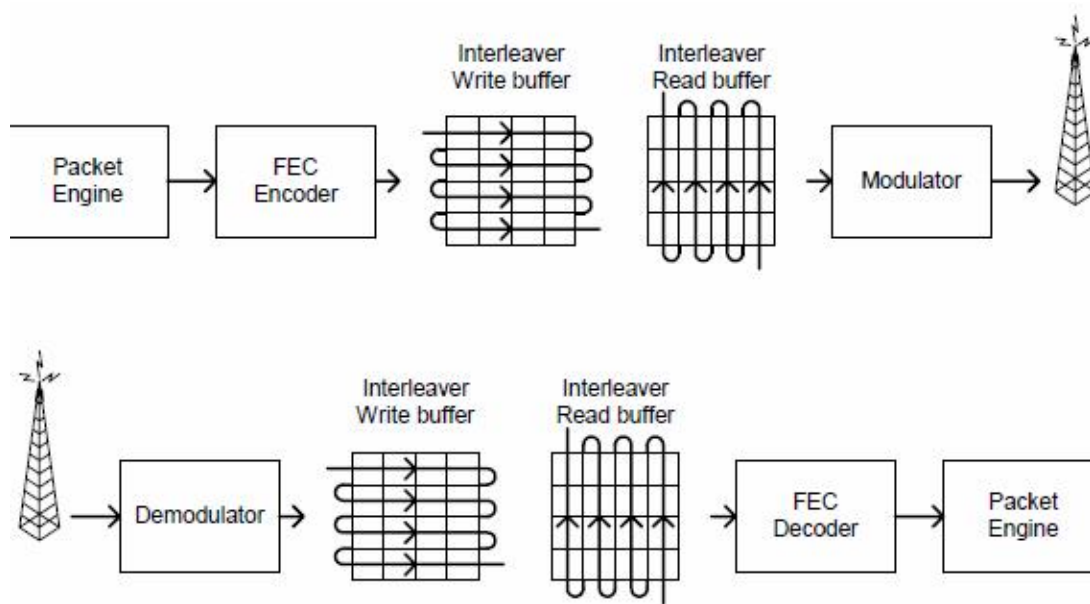


Figure 20: General Principle of Matrix Interleaving

## 19 无线控制

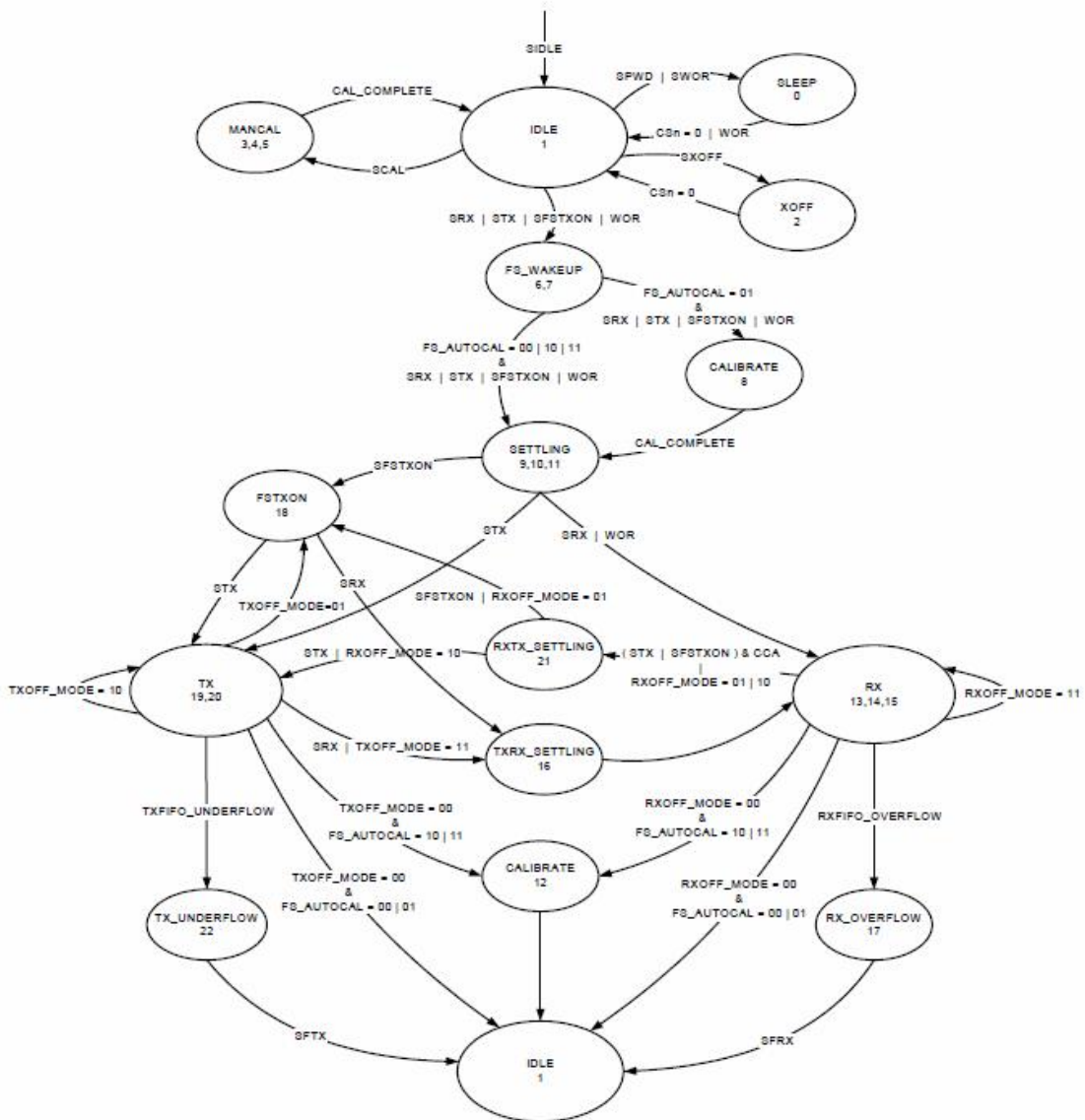


Figure 21: Complete Radio Control State Diagram

CC1101 有一个内部状态机用来切换不同的工作状态（模式）。状态的转换时任意的，通过使用命令选通或者内部事件像 TC FIFO 下溢等。

简单状态图，包含典型用法和电流消耗，见 27 页图 10。完整的无线控制状态图见图 21。所涉及到的状态数量可以从 MARCSTATE 状态寄存器读到，这个寄存器主要是用作测试的目的。

### 19.1 上电启动序列

当电源给电，系统必须复位。通过下面描述的 2 个序列中的一个来完成，即自动上电复位（POR）或者手动复位。自动上电复位或手动复位之后，推荐转换 GDO0 引脚上输出的信号。默认设置是以 CLK\_XOSC/192 的频率输出一个时钟信号。然而，为了在 TX 和 RX 中优化执行，必须选择 58 页表 35 中的配置来配置 GDO 脚。

### 19.1.1 自动 POR

CC1101 内部包含上电复位电路。最小的需求见表 15，完整的上电复位功能必须遵循。当 CHIP\_RDYn 变低内部上电序列完成。在 CSn 拉低后在 SO 引脚上可以查看 CHIP\_RDYn。

当 CC1101 复位完成，芯片将进入 IDLE 状态，晶振开始运行。如果芯片在上电之后有足够的时间让晶振稳定，在捕获 CSn 变低之后 SO 引脚将马上变低。如果在复位完成之前 CSn 被拉低，SO 引脚将首先变高，指示晶振没有稳定，变低之前见图 22。

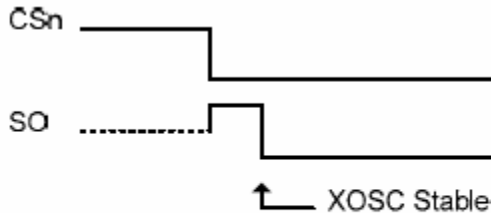


Figure 22: Power-On Reset

### 19.1.2 手动复位

CC1101 上的其它全部复位时使用 SRES 命令选通。通过执行这个选通，所有内部寄存器和状态恢复默认，IDLE 状态。手动复位序列如下（图 23）：

- l 设置 SCLK=1, SI=0, 避免引脚控制模式可能出现的问题。
- l 选通 CS 低/高。
- l 保持 CS 为低，然后拉高，至少 40us 才可以拉低
- l 将 CSn 拉低等到 SO 拉低（CHIP\_RDYn）。
- l 在 SI 线上发布 SRES 选通。
- l 当 SO 再一次拉低，复位完成，芯片进入 IDLE 状态。

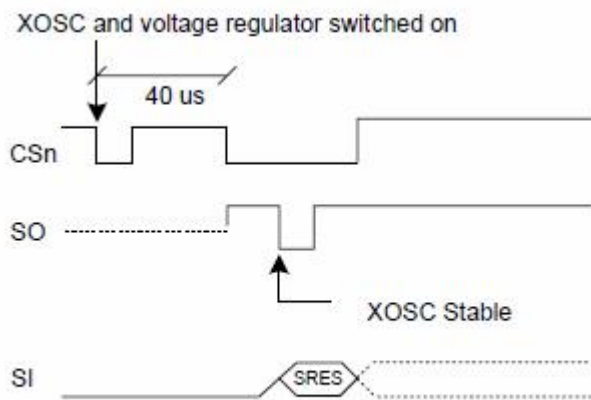


Figure 23: Power-On Reset with SRES

注意：以上复位程序只需要在电源给电之后。如果用户在这之后想要复位 CC1101，只需要执行 SRES 命令选通。



## 19.2 晶振控制

如果设置 `MCSM0.XOSC_FORCE_ON`, 晶体振荡器 (XOSC) 是自动控制的或者一直在线。

在自动模式下, 如果执行 `SXOFF` 或 `SPWD` 命令选通, XOSC 将被关闭; 状态机接着讲变为 `XOFF` 或者 `SLEEP`。这只能从 `IDLE` 状态执行。当 `CSn` 被释放 (变高) XOSC 将被关闭。当 `CSn` 被拉低 XOSC 将再一次被打开。状态机将变为 `IDLE` 状态。在 `SPI` 接口准备使用之前, `SPI` 接口的 `SO` 引脚必须被拉低。

如果 XOSC 被强制开, 即使在 `SLEEP` 状态, 晶振也将保持开的状态。

晶振的启动时间依赖于晶振的 `ESR` 和负载电容。晶振的电气规格可以从 16 页的 `Section4.4` 查找。

## 19.3 电压调整器控制

数字和的电压调整器由无线控制器控制。当芯片进入低功耗的 `SLEEP` 状态时, 电压调整器被禁止。这发生在 `SPWD` 命令选通被发送到 `SPI` 接口 `CSn` 被释放后。接着芯片进入 `SLEEP` 状态。再一次设置 `CSn` 为低将打开调整器和晶振, 使芯片进入 `IDLE` 状态。

如果使能无线唤醒, `WOR` 模块将控制电压调整器, 见 `Section19.5` 描述。

## 19.4 活跃模式

CC1101 有两个活跃的模式: 接收和发送。这些模式直接通过 `MCU` 使用 `SRX` 和 `STX` 命令滤波, 或者自动无线唤醒来激活。

频率合成器必须被有规律的校准。CC1101 有一个手动校准选项 (使用 `SCAL` 滤波) 和 3 个自动校准选项, 由 `MCSM0.FS_AUTOCAL` 设定来控制:

- I 当从 `IDLE` 进入 `RX` 或 `TX` (或 `FSTXON`) 时校准
- I 当从 `RX` 或 `TX` 自动进入 `IDLE` 时校准
- I 每四次从 `RX` 或 `TX` 自动进入 `IDLE` 时校准

如果无线从 `TX` 或 `RX` 使用 `SIDLE` 命令进入 `IDLE`, 不执行校准。校准占用一定数量的 XOSC 周期; 校准的详细时序见表 31。

当 `RX` 被激活, 芯片将一直保持在接收模式, 直到一个数据包被成功的接收或者定时器满 `RX` 中止 (见 `Section19.7`)。降低错误的同步字被检测的可能性可以通过使用 `PQT`, `CS`, 最大同步字长度和 `Section17` 描述的同步字限制。一个数据包被成功接收之后, 无线控制器进入的模式由 `MCSM1.RXOFF_MODE` 来设定。可能的目的地是:

- I `IDLE`
- I `FSTXON`: 同步合成器开并且准备在 `TX` 频率。使用 `STX` 激活 `TX`
- I `TX`: 开始发送前导
- I `RX`: 开始搜索一个新的数据包

注意: 当 `MCSM1.RXOFF_MODE=11`, 并且一个数据包已经被接收, 一个有效的 `RSSI` 值再一次被呈现在 `RSSI` 寄存器之前需要一定的时间, 即使无线从来没有退出 `RX` 模式。

这个时间是 `RSSI` 的响应时间, 论述见 `DN505`。

同理, 当 `TX` 被激活, 芯片一直保持在 `TX` 状态直到当前数据包被成功的发送。接下来状态将按照 `MCSM1.TXOFF_MODE` 来设定。可能的目的地与 `RX` 相同。



MCU 也可以手动的通过使用命令滤波从 RX 到 TX 转换状态。如果无线控制器当前在发送状态，使用 SRX 命令，当前的传送将终止，转换到 RX 模式。

如果无线控制器在 RX 模式，执行 STX 或 SFSTXON 命令，TX-if-CCA 功能将被使用。如果信道为非空闲，芯片将保持在 RX。MCSM1.CCA\_MODE 配置控制 CCA 的条件。详见 46 页 Section17.5。

SIDLE 命令可以一直用来强制无线控制器进入 IDLE 状态。

## 19.5 无线唤醒（WOR）

附加的 WOR 够哦拿给你允许 CC1101 周期性的从 SLEEP 状态唤醒，不需要 MCU 的交互监听输入的数据包。

当 SWOR 命令在 SPI 接口上发送，当 CSn 被释放 CC1101 将持续 SLEEP 状态。在 SWOR 命令被使能之前必须使能 RC 振荡器，作为 WOR 定时器的时钟源。片上定时器将设置 CC1101 进入 IDLE 状态，然后 RX 状态。在编程好的 RX 时间之后，芯片将回到 SLEEP 状态，除非接收到一个数据包。详见图 24 和 Section19.7。

要退出 WOR 模式，设置 CC1101 进入 IDLE 状态。

使用 GDO 引脚当接收到数据包后 CC1101 可以发信号给 MCU。如果一个数据包被接收到，MCSM1.RXOFF\_MODE 将决定数据包接收完成之后的动作。当 MCU 读取数据包，它可以使用 SWOR 命令将芯片从 IDLE 状态转回到 SLEEP 状态。

注意：FIFO 在 SLEEP 状态释放其中的内容。

WOR 定时器有两个事件，事件 0 和事件 1。在 WOR 激活的 SLEEP 状态，达到事件 0 将打开数字校准器，启动晶振。在编程好的超时后事件 1 跟随事件 0。

两个连续的事件 0 之间的时间由 WOREVT1.EVENT0 和 WOREVT0.EVENT0 的尾数值编程，典型值由 WORCTRL.WOR\_RES 设置。公式为：

$$t_{Event0} = \frac{750}{f_{XOSC}} \cdot EVENT0 \cdot 2^{5 \cdot WOR\_RES}$$

事件 1 的超时时间由 WORCTRL.EVENT1 配置。图 24 显示事件 0 超时和事件 1 超时的时序关系。

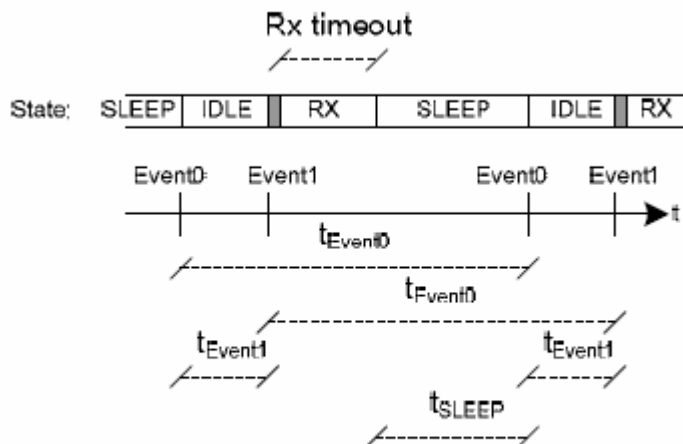


Figure 24: Event 0 and Event 1 Relationship

从 CC1101 进入 SLEEP 状态直到下一个被编程的事件 0 出现的时间，图 24 的  $t_{\text{SLEEP}}$ ，当使用 26MHz 晶振的时候必须大于 11.08ms，在使用 27MHz 晶振的时候必须大于 10.67ms。如果  $t_{\text{SLEEP}}$  小于 11.08 (10.67) ms，连续事件 0 有产生太早的可能。

$$\frac{750}{f_{\text{XOSC}}} \cdot 128 \text{ seconds}$$

AN047 详细的说明了使用 WOR 时的工作原理和不同的寄存器相关。

## 19.5.1 RC 振荡器和时序

用于 WOR 功能的低功耗 RC 振荡器的频率随着温度和供电电压而变化。为了尽量保持频率的稳定，RC 振荡器要尽可能的被校准，当 XOSC 在运行而且芯片不处于 SLEEP 状态。当电源和 XOSC 使能，WOR 定时器使用的时钟和 XOSC 是分开的。当芯片进入睡眠状态，RC 振荡器使用最好的有效校准结果。RC 振荡器的频率锁定在主晶振频率的 750 分频。

在无线唤醒频繁的应用中，典型的每秒几次，尽可能的做到 RC 振荡器校准一次病关掉校准来降低功耗。这可以通过设定 WORCTRL.RC\_CAL=0，这需从 RCCTRL0\_STATUS 和 RCCTRL1\_STATUS 读出 RC 振荡器的校准值，并分别写回到 RCCTRL0 和 RCCTRL1。如果 RC 振荡器校准被关闭，如果温度和电源电压改变需要再一次手动打开。更多详细资料参考 AN047。

## 19.6 时序

在 CC1101 中无线控制器控制着大多数的时序，像合成器校准，PLL 锁定时间，RX/TX 转换时间。从 IDLE 到 RX 和从 IDLE 到 TX 的时序是恒定的，依赖于自动校准的设置。RX/TX 和 TX/RX 转换的时间也是恒定的。校准时间是恒定的 18739 个时钟周期。表 31 显示关键状态转换的晶振时钟周期的时序。

上电时间和 XOSC 启动时间是可变的，但是在表 10 的极限规定之内。

注意，在跳频传播的频谱或者多信道协议，校准时间可以从 721us 减少到 50us。详见 Section28.2。

Description	XOSC Periods	26 MHz Crystal
IDLE to RX, no calibration	2298	88.4μs
IDLE to RX, with calibration	~21037	809μs
IDLE to TX/FSTXON, no calibration	2298	88.4μs
IDLE to TX/FSTXON, with calibration	~21037	809μs
TX to RX switch	560	21.5μs
RX to TX switch	250	9.6μs
RX or TX to IDLE, no calibration	2	0.1μs
RX or TX to IDLE, with calibration	~18739	721μs
Manual calibration	~18739	721μs

Table 31: State Transition Timing

## 19.7 接收终止定时器

CC1101 有一个附加的功能，在编程的时间之后自动终止 RX。这个功能的主要应用时无线唤醒，但是对其他应用也是有用的。当进入 RX 状态启动终止定时器。超时时间由编程 MCSM2.RX\_TIME 来设置。当定时器终止，无线控制器将检查留在 RX 的条件；如果条件不满足，RX 将终止。

可编程的条件为：

- I MCSM2.RX\_TIME\_QUAL=0:如果发现同步字，继续接收
- I MCSM2.RX\_TIME\_QUAL=1:如果发现同步字或者前导质量在门限之上 (PQT)，继续接收

如果当使能接收器的时候系统期望启动传送，可以使用 MCSM2.RX\_TIME\_RSSI 功能。如果第一个有效的 CS 样本显示没有载波 (RSSI 低于门限)，无线控制器将终止 RX。

对于 ASK/OOK 调制，没有载波监听只能考虑有效的 8 个信号周期之后。因此在 ASK/OOK 模式当 1 信号时 8 个或更少的时候可以使用 MCSM2.RX\_TIME\_RSSI。

当 MCSM2.RX\_TIME\_RSSI 功能被使用，如果 RX 终止对于没有 CS 或者当使用 MCSM2.RX\_TIME 超时功能没有发现同步字，如果禁止 WOR 芯片将回到 IDLE 状态，如果使能 WOR 将回到 SLEEP 状态。另外，MCSM1.RXOFF\_MODE 的设置决定 RX 结束时的状态。这意味着一旦接收到同步字芯片将不会自动回到 SLEEP 状态。因此推荐当使用 WOR 模式时，始终唤醒 MCU。这可以通过编程 GDO 输出脚中第一个选择输出信号 6 (58 页表 35) 来实现，并且设置 MCU 从 GDO 脚边沿触发中断唤醒。

## 20 数据 FIFO

CC1101 包含两个 64 字节的 FIFO，一个用来接收数据，一个用来发送数据。SPI 接口用来从 RX FIFO 读取数据，向 TX FIFO 写入数据。Section10.5 包含 SPI FIFO 访问的详细说明。FIFO 控制器将检测 RX FIFO 的上溢和 TX FIFO 的下溢。

当向 TX FIFO 写入数据时 MCU 的责任是避免 TX FIFO 下溢。TX FIFO 溢出将在 TX FIFO 的内容中产生一个错误。

同样，在读取 RX FIFO 时 MCU 要避免读取经过 RX FIFO 的空值，一旦 RX FIFO 下溢将在读出的 RX FIFO 数据中产生一个错误。

当传送 SPI 头字节在 SO 引脚上激活的芯片状态字节，如果访问是读操作时包含 RX FIFO 的填充等级，如果访问是写操作时包含 TX FIFO 的填充等级。详见 30 页 Section10.1。

RX FIFO 和 TX FIFO 中的字节数量可以通过状态寄存器 RXBYTES.NUM\_RXBYTES 和 TXBYTES.NUM\_TXBYTES 读取。如果一个要被接收的字节正好在 RX FIFO 的最后一个字节通过 SPI 接口被读出的时候写入到 RX FIFO，RX FIFO 的指针将不会适时的更新，最后读出的字节将被复制。为了避免这个问题，在数据包的最后一个字节被接收之前 RX FIFO 绝对不能空。

对于数据包长度少于 64 字节，在读出 RX FIFO 的数据之前等到数据包被完全接收。

如果数据包长度大于 64 字节，MCU 必须检测有多少个字节可以从 RX FIFO 读出 (RXBYTES.NUM\_RXBYTES-1)。下面的软件步骤可以被用到：

1. 以一个速率度 RXBYTES.NUM\_RXBYTES，保证至少读 2 次 RF 接收到的字节数直到返回两次相同的值，存储到 n。
2. 如果 n<#数据包中的剩余字节，从 RX FIFO 读出 n-1 个字节。

3. 重复步骤 1 和 2 直到  $n = \#$ 数据包中的剩余字节。
  4. 从 RX FIFO 中读出数据包的剩余字节。
- 4 位的 FIFOTH.R.FIFO\_THR 设定用来编程 FIFO 中的门限指针。

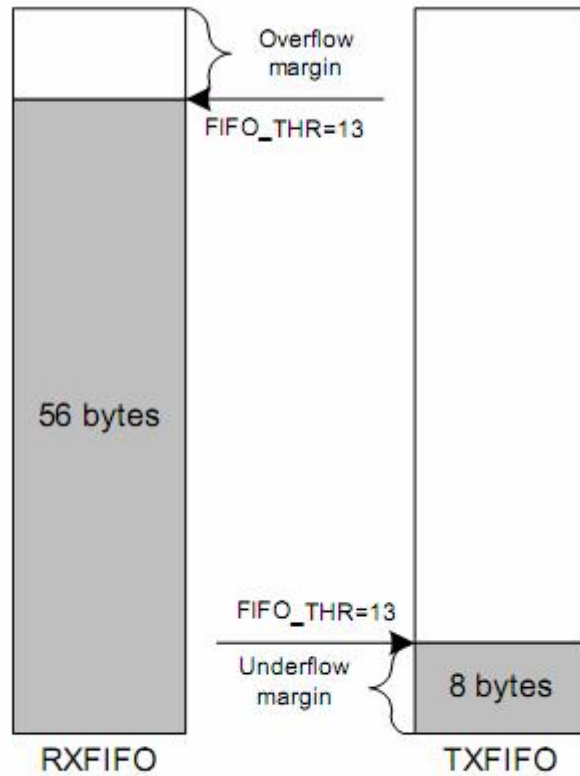
表 32 列出了 16 个 FIFO\_THR 的设置和 RX TX FIFO 的相应门限。该门限值被反相编码到 RX FIFO 和 TX FIFO 中。当达到门限值给出相等的上溢和下溢条件。

FIFO_THR	Bytes in TX FIFO	Bytes in RX FIFO
0 (0000)	61	4
1 (0001)	57	8
2 (0010)	53	12
3 (0011)	49	16
4 (0100)	45	20
5 (0101)	41	24
6 (0110)	37	28
7 (0111)	33	32
8 (1000)	29	36
9 (1001)	25	40
10 (1010)	21	44
11 (1011)	17	48
12 (1100)	13	52
13 (1101)	9	56
14 (1110)	5	60
15 (1111)	1	64

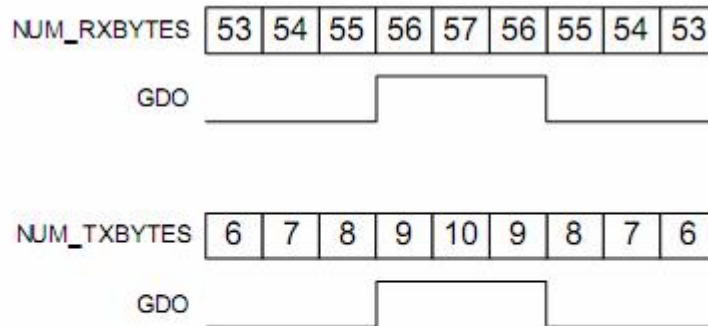
**Table 32: FIFO\_THR Settings and the Corresponding FIFO Thresholds**

当 FIFO 中的字节数等于或高于门限时一个信号将被声明。这个信号可以在 GDO 引脚上查看。(58 页表 35)

图 25 显示了当在 FIFO\_THR=13 门限信号触发时的 RX FIFO 和 TX FIFO 中的字节数。图 26 显示了各自 FIFO 被填充在门限之上的 GDO 引脚的信号，和在 FIFO\_THR=13 的排除情况。



**Figure 25 Example of FIFOs at Threshold**



**Figure 26: Number of Bytes in FIFO vs. the GDO Signal (GDOx\_CFG=0x00 in RX and GDOx\_CFG=0x02 in TX, FIFO\_THR=13)**

## 21 频率设置

CC1101 的频率设置设计最小化设置需要信道导向系统。

使用信道号设置一个系统，预置信道空间通过 MDMCFG0.CHANSPC\_M 和 MDMCFG1.CHANSPC\_E 寄存器来配置。分别为底数和指数。基本或启动频率通过位于 FREQ2, FREQ1 和 FREQ0 寄存器的 24 位频率字来配置。这个字将被设置为最低信道中心频率。



$$f_{carrier} = \frac{f_{XOSC}}{2^{16}} \cdot (FREQ + CHAN \cdot ((256 + CHANSPC - M) \cdot 2^{CHANSPC - E - 2}))$$

使用 26MHz 晶振醉倒的信道间隔为 405kHz。想要获得 1MHz 信道间隔，一个方法是使用 333kHz 信道间隔并在 CHANNR.CHAN 中每个三个信道选择。

首选的 IF 频率通过设置 FSCTRL1.FREQ\_IF 寄存器。IF 频率由下面给出：

$$f_{IF} = \frac{f_{XOSC}}{2^{10}} \cdot FREQ\_IF$$

注意：SmartRF® Studio 根据信道间隔和信道滤波带宽自动计算最合适的 FSCTRL1.FREQ\_IF 寄存器设定值。

当频率合成器运行的时候任何频率设置寄存器改变，合成器可能会给出一个不想要的响应。因此，必须当无线处于 IDLE 状态时更新频率设置。

## 22 VCO

VCO 完全是片上的。

### 22.1 VCO 和 PLL 自校准

VCO 的特性随着温度和供电电压的变化而改变，和预置工作频率一样。为了保证可靠运行，CC1101 包含了频率合成器自校准电路。校准必须是有规律的，必须执行在上电之后和使用一个新的频率（或信道）之前。完成 PLL 校准所需要的 XOSC 周期的数量见 52 页表 31。

校准可以手动或自动开始。合成器在每一次启动或者关闭时可以自动被校准。这由寄存器来配置。在手动模式，当在 IDLE 模式使用 SCAL 命令时校准开始。

注意：在 SLEEP 模式校准值将被保持，因此从 SLEEP 模式唤醒之后仍然有效，直到供电电压或者温度发生改变。

为了检测 PLL 被锁定，用户可以配置寄存器 IOCFGx.GDOx\_CFG=0x0A，使用锁定检测器输出激活在 GDOx 引脚使 MCU 产生中断。在 GDOx 引脚上一个正的传送意味着 PLL 被锁定。第二种选择是读寄存器 FSCAL1。如果寄存器中的内容不同于 0x3F 说明 PLL 被锁定。

更多的操作，源代码可以包含一个检测，如果第一次 PLL 没有锁定那么重新校准 PLL 直到锁定完成。

## 23 电压调整器

CC1101 包含几个片上线性稳压器来产生低功耗模式所需要的电压。这些电压调整器用户是看不到的，可以作为不同模式的主要部分看待。用户必须保证不超过表 1 和表 16 的绝对最大值和所需要的引脚电压。

通过设置 CSn 为低，数字核的电压调整器打开，晶振启动。SPI 接口的 SO 引脚必须在 SCLK 的第一个正边沿被拉低（启动时间见表 19）。

如果芯片被编程进入 power-down 模式（执行 SPWD 命令），在 CSn 拉高后电源将关闭。



当 CSn 被拉低，电源和晶振将再一次打开。

数字核的电压调整器需要一个外部的退藕电容。

电压调整器的输出只能用来驱动 CC1101。

## 24 输出功率编程

设备的 RF 输出功率等级有两个可编程的级别，见图 27。专用寄存器 PATABLE 可以提出 8 个用户选择的输出功率设置。3 位 FRENDO.PA\_POWER 值选择 PATABLE 进入使用。这两个级别的功能可灵活的用在 PA 的功率在传送启动的时候 ramp up 和 ramp down，像 ASK 模式修整一样。PATABLE 中的所有 PA 功率配置从 index0 一直到 FRENDO.PA\_POWER 值被使用。

数据包开始和结束时的功率斜率可以通过设定 FRENDO.PA\_POWER=0 来关闭，接下来编程预期的输出功率到 PATABLE 中的 index 0。

如果使用 OOK 调制，逻辑 0 和逻辑 1 的功率等级必须分别编程到 index0 和 index1。

表 33 包含了不同输出等级的 PATABLE 推荐值和频率带宽。DN013 给出了不同频率带宽的完整表格。不推荐使用 0x61 到 0x6F 的 PA 设定。表 34 包含了默认 PATABLE 设置(0xC6)的输出功率和电流消耗。

PATABLE 的详细说明见 32 页的 Section10.6。如果想写入到 PATABLE[0]之外的其他入口必须使用突发访问模式。

注意：当进入 SLEEP 状态时除了第一个字节（index0）PATABLE 中的内容全部丢失。

	315 MHz		433 MHz		868 MHz		915 MHz	
Output Power [dBm]	Setting	Current Consumption, Typ. [mA]	Setting	Current Consumption, Typ. [mA]	Setting	Current Consumption, Typ. [mA]	Setting	Current Consumption, Typ. [mA]
-30	0x12	10.9	0x12	11.9	0x03	12.1	0x03	12.0
-20	0x0D	11.4	0x0E	12.4	0x0F	12.7	0x0E	12.6
-15	0x1C	12.0	0x1D	13.1	0x1E	13.4	0x1E	13.4
-10	0x34	13.5	0x34	14.4	0x27	15.0	0x27	14.9
-5	0x69	12.8	0x68	13.8	0x67	14.4	0x39	17.7
0	0x51	15.0	0x60	15.9	0x50	16.9	0x8E	16.7
5	0x85	18.3	0x84	19.4	0x81	21.0	0xCD	24.3
7	0xCB	22.1	0xC8	24.2	0xCB	26.8	0xC7	26.9
10	0xC2	26.9	0xC0	29.1	0xC2	32.4	0xC0	31.8

Table 33: Optimum PATABLE Settings for Various Output Power Levels and Frequency Bands

	315 MHz		433 MHz		868 MHz		915 MHz	
Default Power Setting	Output Power [dBm]	Current Consumption, Typ. [mA]	Output Power [dBm]	Current Consumption, Typ. [mA]	Output Power [dBm]	Current Consumption, Typ. [mA]	Output Power [dBm]	Current Consumption, Typ. [mA]
0xC6	8.5	24.4	7.8	25.2	8.5	29.5	7.2	27.4

Table 34: Output Power and Current Consumption for Default PATABLE Setting

## 25 修整和 PA 斜率

ASK 调制，上传 8 个功率设置用来修整。该模式包含一个计数器，当传送 1 时增加二档传送 0 时减少。计数器在等于 8 次信号速率的速率下计数。计数器饱和在

FREND0.PA\_POWER 和 0。

计数值作为查看功率表的索引。因此，当 ASK 模式激活时为了使用整个功率表，FREND0.PA\_POWER=7。ASK 信号的修整依赖于 PATABLE 的配置。图 28 显示了一些 ASK 修整的例子。

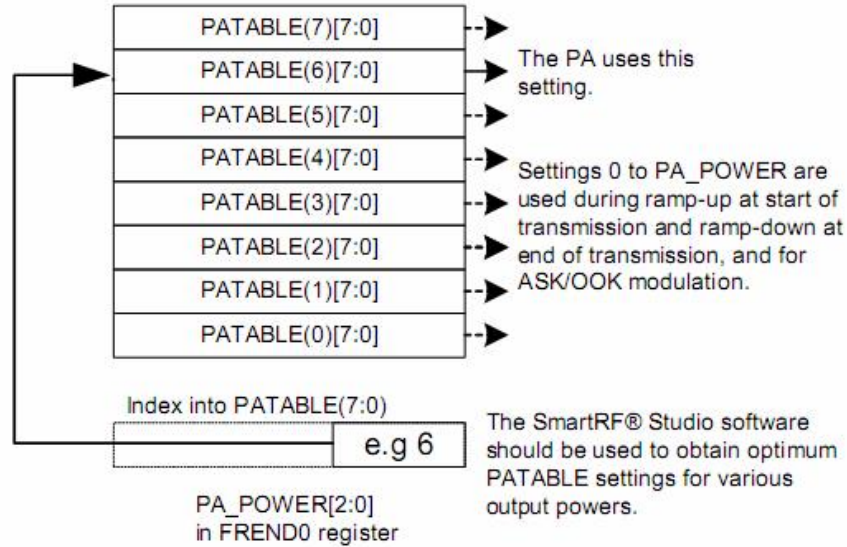


Figure 27: PA\_POWER and PATABLE

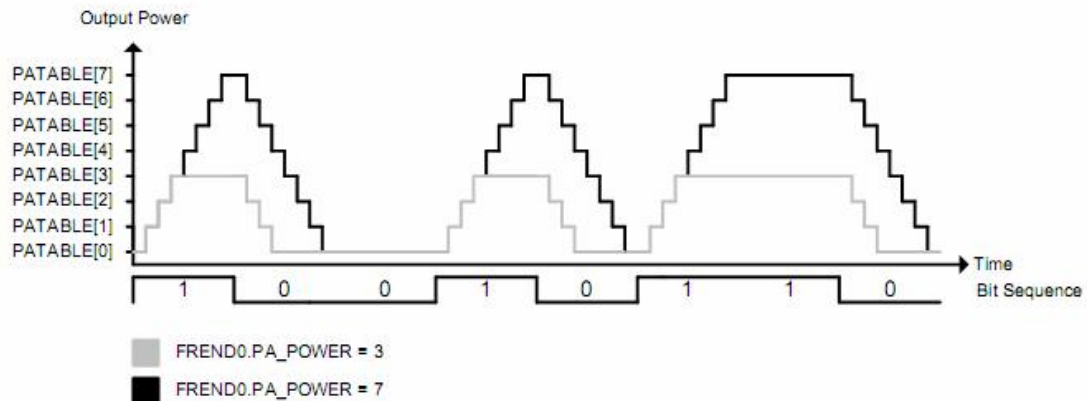


Figure 28: Shaping of ASK Signal

## 26 基本用途/测试输出控制引脚

三个数字输出引脚 GDO0、GDO1 和 GDO2 为普通控制引脚，由 IOCFG0.GDO0\_CFG，IOCFG1.GDO1\_CFG 和 IOCFG2.GDO2\_CFG 分别配置。表 35 显示了可以在 GDO 引脚上监视的不同信号。这些信号可以作为 MCU 的输入信号。

GDO1 在 SPI 接口中作为 SO 引脚，因此在这个引脚上的输出配置只有在 CSn 为高时有效。GDO1 的默认值是 3 态，这对于 SPI 接口与其他设备共用时非常有用的。

GDO0 的默认值是输出 135-141kHz 时钟 (XOSC 频率 192 分频)。XOSC 在上电复位以后打开，可以为 MCU 提供时钟，在系统内置包含一个晶振。当 MCU 上电并且运行，可以通过写 IOCFG0.GDO0\_CFG 来改变时钟频率。

通过将 128 (0x80) 写入 IOCFG0 寄存器, 片上模拟温度传感器可以被使能。GDO0 引脚上的电压和温度成比例。温度传感器的详细说明见 18 页的 Section4.7。

如果 IOCFGx.GDOx\_CFG 的设置小于 0x20 并且 IOCFGx\_GDOx\_INV 为 0(1), 在 SLEEP 模式 GDO0 和 GDO2 将由硬件置 0 (1), GDO1 将有硬件置 1 (0)。这些信号将由硬件保持知道 CHIP\_RDYn 变低。

如果 IOCFGx.GDOx\_CFG 设置为 0x20 或更高, GDO 引脚在 SLEEP 状态也工作在设置的状态。例如, 如果 IOCFG1.GDO1\_CFG=0x2E, GDO1 在所有状态都是高阻抗。

GDOx_CFG[5:0]	描述
0 (0x00)	与 RX FIFO 相关: 当 RX FIFO 被填充到或超过 RX FIFO 门限置 1。当 RX FIFO 排除到低于相同门限时置 0。
1 (0x01)	与 RX FIFO 相关: 当 RX FIFO 被填充到或超过 RX FIFO 门限或者数据包的结束符到达时置 1。当 RX FIFO 为空时置 0
2 (0x02)	与 TX FIFO 相关: 当 TX FIFO 被填充到或超过 TX FIFO 门限置 1。当 TX FIFO 排除到低于相同门限时置 0。
3 (0x03)	与 TX FIFO 相关: 当 TX FIFO 满置 1。当 TX FIFO 排除到低于相同门限时置 0。
4 (0x04)	当 RX FIFO 上溢置 1。当 FIFO 被清洗置 0
5 (0x05)	当 TX FIFO 下溢置 1。当 FIFO 被清洗置 0。
6 (0x06)	当同步字被发送/接收置 1, 数据包结束时置 0。在 RX, 当可选的地址检测错误或者 RX FIFO 上溢将置 0。在 TX 当 TX FIFO 下溢将置 0
7 (0x07)	当数据包接收完成并且 CRC OK 置 1。当第一个字节从 RX FIFO 被读取置 0。
8 (0x08)	前导质量到达。当 PQI 大于配置的 PQT 时置 1。
9 (0x09)	CCA。当 RSSI 级别低于门限值时为高。(依赖于当前 CCA_MODE 设定)
10 (0x0A)	锁定检测输出。如果锁定检测器输出有一个正的传输或者频繁的逻辑高电平则 PPL 在锁定状态。为了检测 PLL 锁定, 锁定检测器输出必须作为 MCU 的中断信号。
11 (0x0B)	串行时钟。在同步串行模式用作同步数据。 在 RX 模式, 当 GDOx_INV=0 数据建立在下降沿。 在 TX 模式, 当 GDOx_INV=0, CC1101 的数据采样在串行时钟的上升沿。
12 (0x0C)	串行同步数据输出。用在同步串行模式。
13 (0x0D)	串行数据输出。用在同步串行模式。
14 (0x0E)	CS。如果 RSSI 级别在门限之上为高电平。
15 (0x0F)	CRC_OK。最后的 CRC 对照匹配。当进入/重启 RX 模式时清 0。
16 (0x10)	保留—用于测试。
17 (0x11)	保留—用于测试。
18 (0x12)	保留—用于测试。
19 (0x13)	保留—用于测试。
20 (0x14)	保留—用于测试。
21 (0x15)	保留—用于测试。
22 (0x16)	RX_HARD_DATA[1]。可与 RX_SYMBOL_TICK 一起使用, 用来选择串行 RX 输出。
23 (0x17)	RX_HARD_DATA[0]。可与 RX_SYMBOL_TICK 一起使用, 用来选择串行 RX 输出。
24 (0x18)	保留—用于测试。
25 (0x19)	保留—用于测试。
26 (0x1A)	保留—用于测试。
27 (0x1B)	PA_PD。注意: PA_PD 在 SLEEP 和 TX 模式有相同的信号等级。为了控制外部 PA 或 RX/TX

	切换在 SLEEP 状态下的应用推荐使用 GDOx_CFGx=0x2F 来代替。	
28 (0x1C)	LNA_PD。注意：LN_PD 在 SLEEP 和 RX 模式有相同的信号等级。为了控制外部 LNA 或 RX/TX 切换在 SLEEP 状态下的应用推荐使用 GDOx_CFGx=0x2F 来代替。	
29 (0x1D)	RX_SYMBOL_TICK。可与 RX_HARD_DATA 一起使用，用来选择串行 RX 输出。	
30 (0x1E)	保留—用于测试。	
31 (0x1F)	保留—用于测试。	
32 (0x20)	保留—用于测试。	
33 (0x21)	保留—用于测试。	
34 (0x22)	保留—用于测试。	
35 (0x23)	保留—用于测试。	
36 (0x24)	WOR_EVT0	
37 (0x25)	WOR_EVT1	
38 (0x26)	保留—用于测试。	
39 (0x27)	CLK_32k	
40 (0x28)	保留—用于测试。	
41 (0x29)	CHIP_RDYn	
42 (0x2A)	保留—用于测试。	
43 (0x2B)	XOSC_STABLE	
44 (0x2C)	保留—用于测试。	
45 (0x2D)	GDO0_Z_EN_N。当这个输出为 0，GDO0 被配置为输入（作为串行 TX 数据）。	
46 (0x2E)	高阻抗（三态）。	
47 (0x2F)	硬件置 0 (通过配置 GDOx_INV=1 完成 HW1)。可以用来控制外部 LNA/PA 或 RX/TX 切换。	
48 (0x30)	CLK_XOSC/1	<p>注意：有 3 个 GDO 引脚，但是在任何时候只能有一个 CLK_XOSC/n 被选择作为输出。如果 CLK_XOSC/n 在一个 GDO 引脚上被监控，其他两个 GDO 引脚必须配置为小于 0x30 的值。GDO0 默认值为 CLK_XOSC/192。</p> <p>为了优化 RF 的性能，当无线处于 RX 和 TX 模式时不能使用这个信号。</p>
49 (0x31)	CLK_XOSC/1.5	
50 (0x32)	CLK_XOSC/2	
51 (0x33)	CLK_XOSC/3	
52 (0x34)	CLK_XOSC/4	
53 (0x35)	CLK_XOSC/6	
54 (0x36)	CLK_XOSC/8	
55 (0x37)	CLK_XOSC/12	
56 (0x38)	CLK_XOSC/16	
57 (0x39)	CLK_XOSC/24	
58 (0x3A)	CLK_XOSC/32	
59 (0x3B)	CLK_XOSC/48	
60 (0x3C)	CLK_XOSC/64	
61 (0x3D)	CLK_XOSC/96	
62 (0x3E)	CLK_XOSC/128	
63 (0x3F)	CLK_XOSC/192	

表 35: GDOx 信号选择 (x=0,1,2)

## 27 异步和同步串行操作

CC1101 中包含的一些特性和工作模式是和以前的 Chipcon 产品或其他存在的 RF 通信系统向后兼容的。在新系统中，推荐使用内在的数据包处理器特性，他们可以给出更多的充足的通信，值得卸下 MCU，更简单的软件发展。

### 27.1 异步操作

CC1101 包含的异步传输时为了向后兼容已经使用异步传输的系统。

当使能异步传输，CC1101 中包含的几个支持 MCU 的机制将被禁止，像硬件数据包处理器，FIFO 缓存等等。异步传输模式不允许使用 data whitener, interleaver 和 FEC，也不能使用曼彻斯特编码。MSK 不支持异步传输。

配置 PKTCTRL0.PKT\_FORMAT=3 使能串行异步模式。在 TX，GDO0 用作数据输入 (TX 数据)。数据输出可以使用 GDO0，GDO1 和 GDO2。通过设置 IOCFG0.GDO0\_CFG, IOCFG1.GDO1\_CFG 和 IOCFG2.GDO2\_CFG。

CC1101 采样异步输入的级别 8 倍于配置的数据速率。异步数据流的时序需求在位周期的错误必须小于 1/8 的编程数据速率。

### 27.2 串行同步操作

配置 PKTCTRL0.PKT\_FORMAT=1 使能串行同步模式。在串行同步模式，数据通过 2 线串行接口传输。CC1101 提供一个用来在数据输入线建立新的数据或在数据输出线采样数据的时钟。数据输入 (TX 数据) 在 GDO0 引脚。当 TX 激活时这个脚自动配置为输入。TX 的执行时间为 8 位。数据输出引脚可以是任意一个 GDO 引脚。通过 IOCFG0.GDO0\_CFG, IOCFG1.GDO1\_CFG 和 IOCFG2.GDO2\_CFG 配置。从开始接收到在接收数据输出引脚上有可用的数据的时间等于 9 个位。

前导和同步字插入/检测可以激活也可以不激活，依赖于同步模式的 MDMCFG2.SYNC\_MODE 配置。

如果禁止前导和同步字，其他所有的数据包处理特性金额 FEC 都必须禁止。MCU 必须通过软件插入和检测前导和同步字。

如果前导和同步字的插入/检测使能，所有的数据包处理特性和 FEC 都可以使用。只有一个除外，地址过滤特性在串行同步模式是不可用的。

当在串行同步模式使用数据包处理特性，CC1101 将插入和检测前导和同步字，MCU 值需要供给/获得数据载荷。这等价于推荐的 FIFO 操作模式。

一个二选一的 RX 输出选项是配置任何一个 GDO 引脚的 RX\_SYMBOL\_TICK 和 RX\_HARD\_DATA，见表 35。RX\_HARD\_DATA[1:0]是硬件确定信号。RX\_HARD\_DATA[1:0]包含 4-ary 调制模式，而 RX\_HARD\_DATA[1]包含 2-ary 调制模式。RX\_SYMBOL\_TICK 信号是信号时钟，高电平作为 1/2 信号周期，无论何时一个新的信号出现在硬件或软件数据输出。这个选项可以用在同步和异步接口。



## 28 系统注意事项和指导

### 28.1 SRD 规章

### 28.2 频率跳变和多信道系统

315MHz、433MHz、868MHz 或 915MHz 频段在工业、办公和家庭环境等很多系统中共用。因此推荐使用频率跳频扩频 (FHSS) 或多信道协议, 因为频率的差异可以减少和工作在同一个频段的其他设备之间的冲突。FHSS 也可以抗多径衰减。

CC1101 非常适合 FHSS 和多信道系统, 由于它的灵活的频率合成器和有效的通信接口。使用数据包处理器支持和数据缓冲器也更有利于这样的系统, 像这些特性可以显著的减轻主控制器的负担。

当 CC1101 执行频率跳变时, 电荷泵电流, VCO 电流和 VCO 容量组校准数据在每一个频率都是必须的。有 3 种方法可以从芯片获得校准数据:

- 1) 每次跳变运行频率跳变校准。PLL 校准时间大约 720us。每次频率跳变之间的消隐期大约是 810us。
- 2) 每次跳变运行不带校准的快速频率跳变, 可以通过在启动时执行必要的校准和将 FSCAL3, FSCAL2 和 FSCAL1 的值作为结果存在 MCU 的存储器中。VCO 电流校准的 FSCAL1 寄存器值在每一个 RF 频率查询。在 FSCAL2 和 FSCAL3 中的 VCO 电流校准值和电荷泵电流校准值不依赖于 RF 频率, 因此这两个寄存器的值可以使用在所有 RF 频率。每两次频率跳变之间, 校准过程可以通过写入符合下一个 RF 频率的值到 FSCAL3, FSCAL2 和 FSCAL1 寄存器来代替。PLL 启动时间大约 90us。每次频率跳变之间的消隐期大约 90us。
- 3) 在启动时在单一频率下运行校准。接下来向 FSCAL3[5:4]写入 0 来禁止电荷泵校准。写完 FSCAL3[5:4]之后, 在每次频率跳变 MCSM0.FS\_AUTOCAL=1 执行 SRX (或 STX)。即 VCO 电流和 VCO 容量校准完成, 但是不校准电荷泵电流。当电荷泵电流校准被禁止校准时间从 720us 减少到 150us。每次频率跳变之间的消隐期大约 240us。

这是消隐时间和存储器空间的交替使用, 需要存储校准数据在非易失性存储器中。方案 2 给出了最小的消隐时间, 但是需要更多的存储空间来存储校准值。这个方案也需要电源电压和温度不会有很大变化。方案 3 给出了比方案 1 大约少于 570us 的消隐时间。

TEST0.VCO\_SEL\_CAL\_EN 的推荐设定随频率改变。这就意味着在校准之前必须使用 SmartRF® Studio 来获得详细频率的正确设置, 无论使用哪一种校准方法。

注意: TESTn (n=0,1 或 2) 的内容在 SLEEP 状态下不保持, 因此当从 SLEEP 状态返回时必须重新写这些寄存器。

### 28.3 当不使用扩频时的宽带调制

### 28.4 无线 MBUS

## 28.5 数据突发传输

CC1101 的最大数据速率时工作在突发访问。一个低的平均速率链接（如 10K）可以通过使用较高的空中数据速率来实现。缓冲数据并且以更高的速率（500k）突发传输能减少在活动模式中的时间，因而可以减少平均电流消耗。

减少在活动模式的时间也可以减少与同频率范围内的其他系统产生冲突的可能。

注意：对于高的数据速率的突发访问比较低速率的灵敏度和传输范围都小。

## 28.6 连续传输

在数据流应用中，CC1101 可以达到在 500k 有效数据速率下连续传输。在闭环 PLL 下调制完成，传输的数据长度没有限制（在一些收发器中使用开环调制要避免这样的连续数据流并减小有效数据速率）。

## 28.7 低功耗系统

像 CC1101 使用 500K 波特率多信道的性能，而没有外部 SAW 或环路滤波器，极低功耗系统是可以实现的。在 CC1101EM 参考设计中使用一个 HC-49 型的 SMD 晶振。晶振的封装极大的影响价格。在有尺寸限制的 PCB 设计中，需要使用一个小的胆识不贵的晶振。

## 28.8 电池工作系统

在低功耗应用中，当 CC1101 不活动时必须使用晶振核关闭的 SLEEP 状态。如果启动时间达到临界，在 SLEEP 状态关闭晶振核的运行时必要的。在低功耗应用用使用 WOR 功能是必要的。

## 28.9 提高输出功率

在一些应用中扩展链路范围可能是必要的。增加哦一个外部功率放大器（PA）是最有效的方法。PA 必须插入到天线和不匹配变压器与匹配电路之间。在接收模式必须使用两个 T/R 开关隔离 PA，详见图 29。

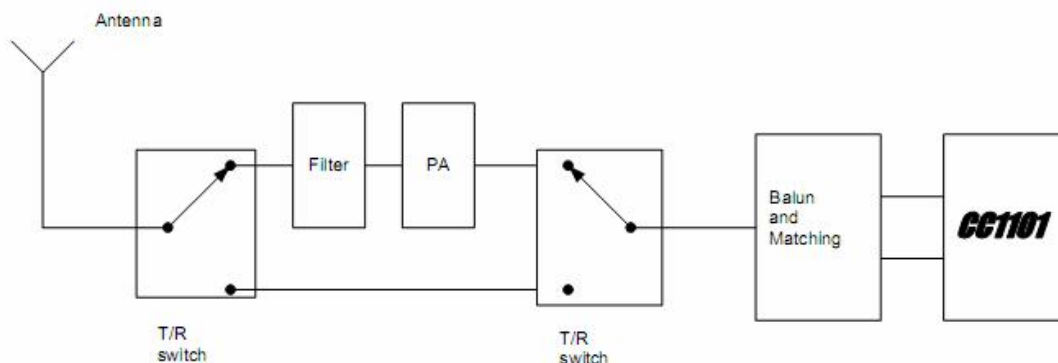


Figure 29: Block Diagram of **CC1101** Usage with External Power Amplifier

## 29 配置寄存器

CC1101 的配置通过编程 8 位寄存器来实现。基于选择好系统参数的最适宜的配置数据的最省力的方法是使用 SmartRF® Studio 软件。接下来的表格中给出寄存器的完整描述。芯片复位后，所有寄存器恢复为表中的默认值。最适合的寄存器设置可能不同于默认值。上电复位后，不同于默认值的寄存器需要听过 SPI 接口编程。

表 36 有 12 个命令滤波寄存器。访问这些寄存器开始改变内部状态或模式。表 37 有 47 个标准 8 位配置寄存器。有一些寄存器仅用来测试的目的，CC1101 的正常操作不需要配置。

表 38 有 12 个状态寄存器。这些寄存器为只读，包含 CC1101 的状态。

两个 FIFO 通过一个 8 位寄存器来访问。写操作写入 TX FIFO，读操作从 RX FIFO 读取。

在头字节发送和当写数据到寄存器或 TX FIFO，在 SO 线上返回一个状态字节。这个状态字节见 30 页表 20 的描述。

表 39 概述 SPI 地址空间。地址的使用通过增加基址到左侧和突发以及读/写位在上面。注意突发位在基址 0x2F 前和后有不同含义。

地址	命令名称	描述
0x30	SRES	复位芯片
0x31	SFSTXON	使能和校准频率合成器（如果 MCSM0.FS_AUTOCAL=1）。如果在 RX(使能 CCA)，仅当合成器运行时转到等待状态(RX/TX 快速转换)。
0x32	SXOFF	关闭晶振
0x33	SCAL	校准频率合成器并关闭。SCAL 可以从 IDLE 模式执行而不需要设置收到校准模式（MCSM0.FS_AUTOCAL=0）。
0x34	SRX	使能 RX。如果来自 IDLE 状态并且 MCSM0.FS_AUTOCAL=1 首先执行校准。
0x35	STX	在 IDLE 状态：使能 TX。如果 MCSM0.FS_AUTOCAL=1 首先执行校准。如果在 RX 状态并使能 CCA 仅在信道为空时转到 TX。
0x36	SIDLE	离开 RX/TX 模式，关闭频率合成器，如果可用离开 WOR。
0x38	SWOR	如果 WORCTRL.RC_PD=0 像 Section19.5 的描述一样开始自动 RX 轮询序列（WOR）。
0x39	SPWD	当 CSn 拉高进入掉电模式。
0x3A	SFRX	清洗 RX FIFO 缓冲器。只在 IDLE 或 RXFIFO_OVERFLOW 状态运行 SFRX。

0x3B	SFTX	清洗 TX FIFO 缓冲器。只在 IDLE 或 TXFIFO_UNDERFLOW 状态运行 SFTX。
0x3C	SWORRST	复位实时时钟到 Event1 值。
0x3D	SMOP	无操作，可以用来访问获取芯片状态字节。

表 36：命令滤波

地址	寄存器	描述	睡眠状态保持
0x00	IOCFG2	GDO2 输出引脚配置	是
0x01	IOCFG1	GDO1 输出引脚配置	是
0x02	IOCFG0	GDO0 输出引脚配置	是
0x03	FIFOTHR	RX FIFO 和 TX FIFO 门限	是
0x04	SYNC1	同步字，高字节	是
0x05	SYNC0	同步字，低字节	是
0x06	PKTLEN	数据包长度	是
0x07	PKTCTRL1	数据包自动控制	是
0x08	PKTCTRL0	数据包自动控制	是
0x09	ADDR	设备地址	是
0x0A	CHANNR	信道号	是
0x0B	FSCTRL1	频率合成器控制	是
0x0C	FSCTRL0	频率合成器控制	是
0x0D	FREQ2	频率控制字，高字节	是
0x0E	FREQ1	频率控制字，中间字节	是
0x0F	FREQ0	频率控制字，低字节	是
0x10	MDMCFG4	Modem 配置	是
0x11	MDMCFG3	Modem 配置	是
0x12	MDMCFG2	Modem 配置	是
0x13	MDMCFG1	Modem 配置	是
0x14	MDMCFG0	Modem 配置	是
0x15	DEVIATN	Modem 背离设定	是
0x16	MCSM2	主无线控制状态机配置	是
0x17	MCSM1	主无线控制状态机配置	是
0x18	MCSM0	主无线控制状态机配置	是
0x19	FOCCFG	频率偏移补偿配置	是
0x1A	BSCFG	位同步配置	是
0x1B	AGCTRL2	AGC 控制	是
0x1C	AGCTRL1	AGC 控制	是
0x1D	AGCTRL0	AGC 控制	是
0x1E	WOREVT1	事件 0 超时高字节	是
0x1F	WOREVT0	事件 0 超时低字节	是
0x20	WORCTRL	WOR 控制	是
0x21	FREND1	前端 RX 配置	是
0x22	FREND0	前端 TX 配置	是
0x23	FSCAL3	频率合成器校准	是

0x24	SCAL2	频率合成器校准	是
0x25	SCAL1	频率合成器校准	是
0x26	SCAL0	频率合成器校准	是
0x27	RCCTRL1	RC 振荡器配置	是
0x28	RCCTRL0	RC 振荡器配置	是
0x29	FSTEST	频率合成器校准控制	否
0x2A	PTEST	产品测试	否
0x2B	AGCTEST	AGC 测试	否
0x2C	TEST2	各方面测试设定	否
0x2D	TEST1	各方面测试设定	否
0x2E	TEST0	各方面测试设定	否

表 37 配置寄存器总览

地址	寄存器	描述
0x30 (0xF0)	PARTNUM	CC1101 零件号
0x31 (0xF1)	VERSION	当前版本号
0x32 (0xF2)	FREQEST	频率偏移评估
0x33 (0xF3)	LQI	解调器链路质量估计
0x34 (0xF4)	RSSI	接收信号强度指示
0x35 (0xF5)	MARCSTATE	控制状态机状态
0x36 (0xF6)	WORTIME1	WOR 定时器高字节
0x37 (0xF7)	WORTIME0	WOR 定时器低字节
0x38 (0xF8)	PKTSTATUS	当前 GDOx 状态和数据包状态
0x39 (0xF9)	VCO_VC_DAC	PLL 校准模块的当前设置
0x3A (0xFA)	TXBYTES	下溢和 TX FIFO 中的字节数
0x3B (0xFB)	RXBYTES	上溢和 RX FIFO 中的字节数
0x3C (0xFC)	RCCTRL1_STATUS	上一次 RC 振荡器校准结果
0x3D (0xFD)	RCCTRL0_STATUS	上一次 RC 振荡器校准结果

表 38 状态寄存器总览



	Write		Read		
	Single Byte	Burst	Single Byte	Burst	
	+0x00	+0x40	+0x80	+0xC0	
0x00			IOCFG2		RW configuration registers, burst access possible
0x01			IOCFG1		
0x02			IOCFG0		
0x03			FIFOTHR		
0x04			SYNC1		
0x05			SYNC0		
0x06			PKTLEN		
0x07			PKTCTRL1		
0x08			PKTCTRL0		
0x09			ADDR		
0x0A			CHANNR		
0x0B			FSCTRL1		
0x0C			FSCTRL0		
0x0D			FREQ2		
0x0E			FREQ1		
0x0F			FREQ0		
0x10			MDMCFG4		
0x11			MDMCFG3		
0x12			MDMCFG2		
0x13			MDMCFG1		
0x14			MDMCFG0		
0x15			DEVIATN		
0x16			MCSM2		
0x17			MCSM1		
0x18			MCSM0		
0x19			FOCCFG		
0x1A			BSCFG		
0x1B			AGCCTRL2		
0x1C			AGCCTRL1		
0x1D			AGCCTRL0		
0x1E			WOREVT1		
0x1F			WOREVT0		
0x20			WORCTRL		
0x21			FREND1		
0x22			FREND0		
0x23			FSCAL3		
0x24			FSCAL2		
0x25			FSCAL1		
0x26			FSCAL0		
0x27			RCCTRL1		Command Strobes, Status registers (read only) and multi byte registers
0x28			RCCTRL0		
0x29			FS TEST		
0x2A			PTEST		
0x2B			AGCTEST		
0x2C			TEST2		
0x2D			TEST1		
0x2E			TEST0		
0x2F					
0x30	SRES		SRES	PARTNUM	
0x31	SFSTXON		SFSTXON	VERSION	
0x32	SXOFF		SXOFF	FREQEST	
0x33	SCAL		SCAL	LOI	
0x34	SRX		SRX	RSSI	
0x35	STX		STX	MARCSTATE	
0x36	SIDLE		SIDLE	WORTIME1	
0x37				WORTIME0	
0x38	SWOR		SWOR	PKTSTATUS	
0x39	SPWD		SPWD	VCO_VC_DAC	
0x3A	SFRX		SFRX	TXBYTES	
0x3B	SFTX		SFTX	RXBYTES	
0x3C	SVORRST		SVORRST	RCCTRL1_STATUS	
0x3D	SNOP		SNOP	RCCTRL0_STATUS	
0x3E	PATABLE	PATABLE	PATABLE	PATABLE	
0x3F	TX FIFO	TX FIFO	RX FIFO	RX FIFO	

表 39 SPI 地址空间



## 重要声明

德州仪器(TI) 及其下属子公司有权在不事先通知的情况下, 随时对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权随时中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的硬件产品的性能符合TI 标准保修的适用规范。仅在TI 保证的范围内, 且TI 认为有必要时才会使用测试或其它质量控制技术。除非政府做出了硬性规定, 否则没有必要对每种产品的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何TI 专利权、版权、屏蔽作品权或其它与使用了TI 产品或服务的组合设备、机器、流程相关的TI 知识产权中授予的直接或隐含权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是TI 的专利权或其它知识产权方面的许可。

对于TI 的产品手册或数据表, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。在复制信息的过程中对内容的篡改属于非法的、欺诈性商业行为。TI 对此类篡改过的文件不承担任何责任。

在转售TI 产品或服务时, 如果存在对产品或服务参数的虚假陈述, 则会失去相关TI 产品或服务的明示或暗示授权, 且这是非法的、欺诈性商业行为。TI 对此类虚假陈述不承担任何责任。

TI 产品未获得用于关键的安全应用中的授权, 例如生命支持应用(在该类应用中一旦TI 产品故障将预计造成重大的人员伤亡), 除非各方官员已经达成了专门管控此类使用的协议。购买者的购买行为即表示, 他们具备有关其应用安全以及规章衍生所需的所有专业技术和知识, 并且认可和同意, 尽管任何应用相关信息或支持仍可能由TI 提供, 但他们将独力负责满足在关键安全应用中使用其产品 & TI 产品所需的所有法律、法规和安全相关要求。此外, 购买者必须全额赔偿因在此类关键安全应用中使用TI 产品而对TI 及其代表造成的损失。

TI 产品并非设计或专门用于军事/航空应用, 以及环境方面的产品, 除非TI 特别注明该产品属于“军用”或“增强型塑料”产品。只有TI 指定的军用产品才满足军用规格。购买者认可并同意, 对TI 未指定军用的产品进行军事方面的应用, 风险由购买者单独承担, 并且独力负责在此类相关使用中满足所有法律和法规要求。

TI 产品并非设计或专门用于汽车应用以及环境方面的产品, 除非TI 特别注明该产品符合ISO/TS 16949 要求。购买者认可并同意, 如果他们在汽车应用中使用任何未被指定的产品, TI 对未能满足应用所需要求不承担任何责任。

可访问以下URL 地址以获取有关其它TI 产品和应用解决方案的信息:

产品	应用		
数字音频	<a href="http://www.ti.com.cn/audio">www.ti.com.cn/audio</a>	通信与电信	<a href="http://www.ti.com.cn/telecom">www.ti.com.cn/telecom</a>
放大器和线性器件	<a href="http://www.ti.com.cn/amplifiers">www.ti.com.cn/amplifiers</a>	计算机及周边	<a href="http://www.ti.com.cn/computer">www.ti.com.cn/computer</a>
数据转换器	<a href="http://www.ti.com.cn/dataconverters">www.ti.com.cn/dataconverters</a>	消费电子	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
DLP® 产品	<a href="http://www.dlp.com">www.dlp.com</a>	能源	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
DSP - 数字信号处理器	<a href="http://www.ti.com.cn/dsp">www.ti.com.cn/dsp</a>	工业应用	<a href="http://www.ti.com.cn/industrial">www.ti.com.cn/industrial</a>
时钟和计时器	<a href="http://www.ti.com.cn/clockandtimers">www.ti.com.cn/clockandtimers</a>	医疗电子	<a href="http://www.ti.com.cn/medical">www.ti.com.cn/medical</a>
接口	<a href="http://www.ti.com.cn/interface">www.ti.com.cn/interface</a>	安防应用	<a href="http://www.ti.com.cn/security">www.ti.com.cn/security</a>
逻辑	<a href="http://www.ti.com.cn/logic">www.ti.com.cn/logic</a>	汽车电子	<a href="http://www.ti.com.cn/automotive">www.ti.com.cn/automotive</a>
电源管理	<a href="http://www.ti.com.cn/power">www.ti.com.cn/power</a>	视频和影像	<a href="http://www.ti.com.cn/video">www.ti.com.cn/video</a>
微控制器 (MCU)	<a href="http://www.ti.com.cn/microcontrollers">www.ti.com.cn/microcontrollers</a>		
RFID 系统	<a href="http://www.ti.com.cn/rfidsys">www.ti.com.cn/rfidsys</a>		
OMAP 机动性处理器	<a href="http://www.ti.com/omap">www.ti.com/omap</a>		
无线连通性	<a href="http://www.ti.com.cn/wirelessconnectivity">www.ti.com.cn/wirelessconnectivity</a>		
德州仪器在线技术支持社区		<a href="http://www.deyisupport.com">www.deyisupport.com</a>	

邮寄地址: 上海市浦东新区世纪大道 1568 号, 中建大厦 32 楼 邮政编码: 200122  
Copyright © 2012 德州仪器 半导体技术(上海)有限公司