UM2010 用户手册

版本：V1.3

广芯微电子（广州）股份有限公司

http://www.unicmicro.com/

条款协议

版本修订

版本	日期	描述
V1.0	2021.12.15	初始版
V1.1	2022.03.18	新增并更新寄存器列表
V1.2	2022.04.22	更新寄存器定义章节； 更新功能框图； 新增 RSSI 线性图； 更新电气参数； 更新 QFN20 封装尺寸图； 新增 RSSI，BUCK DCDC，GPIO 和中断及 FIFO 等相关描述的章节。
V1.3	2023.06.20	修改部分寄存器描述； 更新电气参数值，且表格重新排版； 调整功能描述、芯片运行、数据处理机制内容及排版

目录

表目录

图目录

1 系统概述

UM2010 是一款工作于 200MHz~960MHz 范围内的低功耗、高性能、单片集成的(G)FSK/OOK 无线收发机芯片。内部集成完整的射频接收机、射频发射机、频率综合器、调制解调器，只需配备简单、低成本的外围器件就可以获得良好的收发性能。

芯片支持灵活可设的数据包格式，支持自动应答和自动重发功能，支持跳频操作，同时集成了 FEC 功能。外部 MCU 可通过 SPI 对芯片进行控制，并访问内部收发各 128 bytes 的 TX/RX FIFO。

1.1 典型应用

- 工业传感及工业控制
- 安防系统
- 自动抄表
- 无线标签，无线门禁
- 遥控装置，无线玩具
- 智能交通，智慧城市，智能家居
- 智能门锁，资产追踪、无线监控等智能传感器终端应用

1.2 主要特性

- **功能特点**
  - 频率范围：200MHz~960MHz
  - 调制方式：(G)FSK, OOK
  - 数据率：0.1 ~ 300 kbps
  - 支持 NRZ、曼彻斯特、数据白化
  - 自动应答/自动重传
  - 支持 RSSI，0.5dB 检测精度
  - 可配置包处理机及 128-Byte TX/RX FIFO
  - AGC / AFC
  - 支持 FEC
- **发射功率**
  - -20dBm ~ +18dBm
- **发射电流 (F_{RF}=433.92MHz 非 BUCK 模式)**
  - 14mA @ 0dBm

- 22mA @ 10dBm
- 28mA @ 13dBm
- 58mA @ 18dBm

- **接收灵敏度 (F_{RF}=433.92MHz BUCK 模式)**
  - -127dBm @ 0.1kbps
  - -119dBm @ 1.2kbps
  - -109dBm @ 10kbps
  - -100dBm @ 100kbps
  - -93dBm @ 300kbps
- **接收灵敏度 (F_{RF} =433.92MHz 非 BUCK 模式)**
  - -130dBm @ 0.1kbps
  - -122dBm @ 1.2kbps
  - -112dBm @ 10kbps
  - -102dBm @ 100kbps
  - -97dBm @ 300kbps
- **接收电流 (F_{RF}=433.92MHz)**
  - BUCK 模式：6.5mA
  - 非 BUCK 模式：12mA
- **关断电流**
  - < 10nA
- **接口**
  - 标准四线 SPI 或三线 SPI，速率最高 16Mbps
  - 支持外部复位
  - 支持数据直通
- **电气参数**
  - 工作电压：1.8V~3.6V
  - 工作温度：-40°C ~85°C
  - ESD 保护：±3KV（HBM）
- **开发支持**
  - SDK：软件、文档、工具、参考设计
  - EVB 硬件开发板

2 功能框图

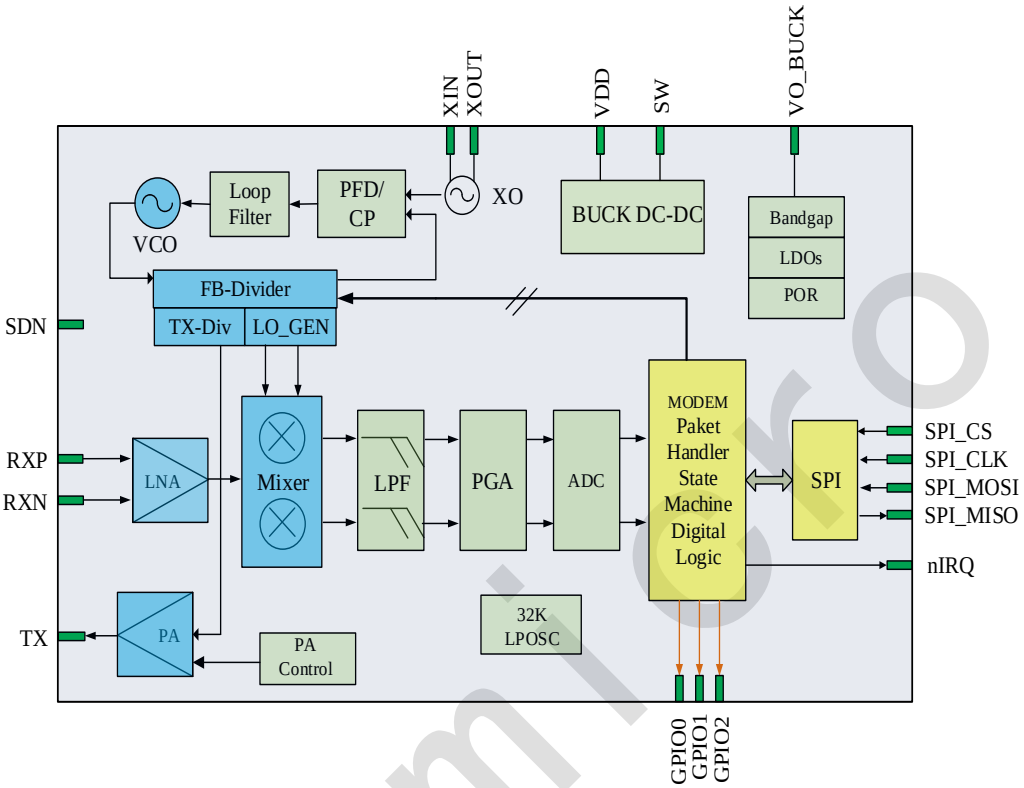


图 2-1：功能框图

3 封装及引脚定义

3.1 封装管脚分布

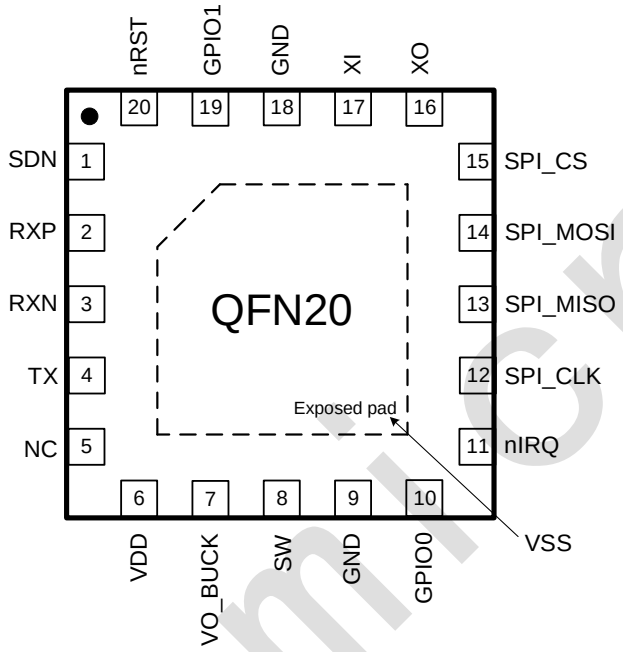


图 3-1：QFN20 封装管脚分布图

3.2 引脚功能描述

表 3-1：引脚功能说明

引脚编号	管脚名称	IO Type	功能描述
0	VSS	G	芯片地(LF PAD)公共地
1	SDN	DI	芯片关断使能，SDN 高电平时芯片处于关断模式
2	RXP	RFI	射频正端输入
3	RXN	RFI	射频负端输入
4	TX	RFO	射频 PA 输出
5	NC	NC	空脚，未连接任何内部电路
6	VDD	P	1.8V~3.6V 电源输入
7	VO_BUCK	AI	内部 LDO 供电电源，外接 BUCK DCDC 输出电压
8	SW	AIO	BUCK DCDC 电感接入端
9	GND	G	芯片地
10	GPIO0	DIO	可配置 GPIO0
11	nIRQ	DIO	可配置 GPIO，默认为中断输出
12	SPI_CLK	DI	SPI 时钟

引脚编号	管脚名称	IO Type	功能描述
13	SPI_MISO	DO	SPI 数据输出
14	SPI_MOSI	DIO	SPI 数据输入（或三线制输入输出）
15	SPI_CS	DI	片选信号
16	XO	AO	晶振输出
17	XI	AI	晶振输入
18	GND	G	芯片地
19	GPIO1	DIO	可配置 GPIO1
20	nRST	DIO	可配置 GPIO2，默认为外部复位引脚

说明：RF-射频信号；A –模拟信号；D –数字信号；I – Input；O – Output；G – Ground；P – Power。

4 寄存器定义

4.1 Reg00 Address:0x00 Default:0x1D

Bit	Name	Type	Description	Default
7	reserved	R/W	-	1'b0
6:0	rf_f0[30:24]	R/W	信道频率 f0[30:24]。 信道频率 rf_freq 设置公式： **rf_freq=rf_f0+ch_num*f_step** rf_f0 由{Reg00,Reg01,Reg02,Reg03}配置，单位为 MHz, 低 20bit 为小数； ch_num 由 Reg04 配置； 步进频率 f_step 由{Reg05,Reg06,Reg07}配置，单位为 MHz，低 20bit 为小数。	7'h1D

4.2 Reg01 Address:0x01 Default:0x78

Bit	Name	Type	Description	Default
7:0	rf_f0[23:16]	R/W	信道频率 f0[23:16]	8'h78

4.3 Reg02 Address:0x02 Default:0x00

Bit	Name	Type	Description	Default
7:0	rf_f0[15:8]	R/W	信道频率 f0[15:8]	8'h00

4.4 Reg03 Address:0x03 Default:0x00

Bit	Name	Type	Description	Default
7:0	rf_f0[7:0]	R/W	信道频率 f0[7:0]	8'h00

4.5 Reg04 Address:0x04 Default:0x00

Bit	Name	Type	Description	Default
7:0	ch_num	R/W	信道号 ch_num, 参考 Reg00 描述	8'h00

4.6 Reg05 Address:0x05 Default:0x03

Bit	Name	Type	Description	Default
7:0	f_step[23:16]	R/W	信道步进频率 f_step[23:16]，单位为 MHz，低 20bit 为小数，参考 Reg00 描述	8'h03

4.7 Reg06 Address:0x06 Default:0x33

Bit	Name	Type	Description	Default
7:0	f_step[15:8]	R/W	信道步进频率 f_step[15:8]，参考 Reg00 描述	8'h33

4.8 Reg07 Address:0x07 Default:0x33

Bit	Name	Type	Description	Default
7:0	f_step[7:0]	R/W	信道步进频率 f_step[7:0]，参考 Reg00 描述	8'h33

4.9 Reg08 Address:0x08 Default:0x10

Bit	Name	Type	Description	Default
7	Reserved	R/W	-	1'b0
6:0	ref_freq[30:24]	R/W	参考频率（晶振频率）设置，ref_freq 由{Reg08[6:0],Reg09,Reg0A,Reg0B}配置，单位为 MHz，低 24bit 为小数部分。	7'h10

4.10 Reg09 Address:0x09 Default:0x00

Bit	Name	Type	Description	Default
7:0	ref_freq[23:16]	R/W	参考 Reg08 描述	8'h00

4.11 Reg0A Address:0x0A Default:0x00

Bit	Name	Type	Description	Default
7:0	ref_freq[15:8]	R/W	参考 Reg08 描述	8'h00

4.12 Reg0B Address:0x0B Default:0x00

Bit	Name	Type	Description	Default
7:0	ref_freq[7:0]	R/W	参考 Reg08 描述	8'h00

4.13 Reg0C Address:0x0C Default:0x50

Bit	Name	Type	Description	Default
7:0	tx_preamble_len	R/W	发射 Preamble 长度，单位字节	8'h50

4.14 Reg0D Address:0x0D Default:0x30

Bit	Name	Type	Description	Default
7	payload_bit_order	R/W	Payload bit 顺序： 0：LSB 低位在前 1：MSB 高位在前	1'b0
6	mancester_inv	R/W	曼彻斯特编码： 0：上升沿为编码 1，下降沿为编码 0 1：下降沿为编码 1，上升沿为编码 0	1'b0
5	syncword_en	R/W	Syncword 同步字使能： 0：disable 1：enable	1'b1
4	preamble_en	R/W	设置 Preamble 前导码使能： 0：disable 1：enable 使能	1'b1
3:2	pkt_enc_type	R/W	数据包编码： 00：NRZ 01：曼彻斯特编码 10：无效 11：交织编码	2'b00
1:0	fec_type	R/W	数据包 FEC 类型： 00：NO FEC 01：1/3 FEC 10：2/3 FEC 11：1/2 FEC	2'b00

4.15 Reg0E Address:0x0E Default:0x32

Bit	Name	Type	Description	Default
7	length_byte_swap	R/W	包长度控制域为两个字节时，高低字节顺序： 0：低字节在前 1：高字节在前	1'b0
6	length_sel	R/W	包长度控制域字节配置： 0：1 字节 1：2 字节	1'b0
5	crc_en	R/W	CRC 使能： 0：disable 1：enable	1'b1

Bit	Name	Type	Description	Default
4	scramble_en	R/W	数据白化（扰码）使能： 0：disable 1：enable	1’b1
3	fifo_share_en	R/W	FIFO 共享设置： 0：RX 和 TX 各 128 字节 FIFO 1：RX 和 TX 共用 256 字节 FIFO	1’b0
2	direct_mode	R/W	直通模式使能： 0：disable 1：enable	1’b0
1:0	packet_mode	R/W	数据包控制模式： 00：模式 0，循环发射 TX FIFO 中的数据，需要 MCU 写命令退出接收或发射状态 01：模式 1，由寄存器控制数据包长度 10：模式 2，TX FIFO 中的第 1 个或 2 个字节作为包长度 11：模式 3，含有 length, address, seqnum 等多种数据域的模式	2’b10

4.16 Reg10 Address:0x10 Default:0x40

Bit	Name	Type	Description	Default
7:4	retx_times	R/W	Auto ACK 使能时，发射端未收到 ACK 包时自动重发的次数	4’h4
3	rssi_val_sel	R/W	读取 RSSI 时，选择读取的数据内容： 0：经过计算的 RSSI 值 1：原始信号幅度值	1’b0
2	preamble_int_en	R/W	接收 Preamble（前导码）匹配中断使能： 0：disable 1：enable	1’b0
1	syncword_int_en	R/W	接收 Syncword（同步字）匹配中断使能： 0：disable 1：enable	1’b0
0	int_pulse_sel	R/W	中断信号类型设置： 0：电平中断 1：脉冲中断（1μs）	1’b0

4.17 Reg11 Address:0x11 Default:0x00

Bit	Name	Type	Description	Default
7	int_polarity	R/W	中断信号极性选择： 0：high active 1：low active	1’b0
6	int_flag_clr	R/W	写‘1’清除中断标志	1’b0
5	miso_tri_opt	R/W	SPI 在非工作状态时，spi_miso 状态： 0：高阻 1：输出	1’b0
4:3	reserved	R/W	-	2’b00

Bit	Name	Type	Description	Default
2:0	brclk_sel	R/W	GPIO brclk 输出时钟选择： 000：rx_clk 接收 bit rate 时钟 001：xtal_clk 晶振时钟 010：xtal_clk/8 011：xtal_clk/16 100：频综时钟 101：tx_clk 发射 bit rate 时钟 110：内部 RC32K 时钟 111：ADC_clk	3’b000

4.18 Reg1B Address:0x1B Default:0x89

Bit	Name	Type	Description	Default
7	reserved	R/W	-	1’b1
6:4	buck_imax	R/W	BUCK 峰值电流设置： 000：80mA 001：110mA 010：140 mA 011：170 mA 100：200mA 101：230mA 110：260mA 111：290mA	3’b000
3:2	buck_vadj	R/W	BUCK 输出电压设置： 00：1.4V 01：1.5V 10：1.6V 11：1.7V	2’b10
1	buck_bp	R/W	BUCK 旁路模式： 0：disable 1：enable	1’b0
0	buck_en	R/W	BUCK 使能： 0：disable 1：enable	1’b1

4.19 Reg1D Address:0x1D Default:0x80

Bit	Name	Type	Description	Default
7:6	vdddig	R/W	数字 LDO 输出电压设置： 00：0.8V 01：1.0V 10：1.2V 11：1.4V	2’b10
5:0	reserved	R/W	-	6’h00

4.20 Reg20 Address:0x20 Default:0x50

Bit	Name	Type	Description	Default
7:6	frequency_band	R/W	工作频段设置： 00：800MHz~1GHz 频段 01：400MHz~500MHz 频段 10：267MHz~350MHz 频段 11：200MHz~250MHz 频段	2’b01
5:0	Reserved	R/W	-	6’h10

4.21 Reg21 Address:0x21 Default:0x60

Bit	Name	Type	Description	Default
7	reserved	R/W	-	1’b0
6	pa_ramp_en	R/W	PA RAMP 使能： 0：disable 1：enable	1’b1
5:0	pa_bias	R/W	PA_bias 控制： 000000：最小 …… 111111：最大	6’h20

4.22 Reg27 Address:0x27 Default:0x60

Bit	Name	Type	Description	Default
7	reserved	R/W	-	1’b0
6	pa_drvh	R/W	PA 前级驱动设置： 0：low 1：high	1’b1
5:0	pa_gain	R/W	PA 增益控制	6’h20

4.23 Reg28 Address:0x28 Default:0x19

Bit	Name	Type	Description	Default
7	rssi_Reg_clr	R/W	写‘1’清除被锁定的 RSSI	1’b0
6	rssi_sel	R/W	收到同步字后锁定 RSSI 值： 0：disable 1：enable	1’b0
5	rx_rssi_thr_en	R/W	接收时判断 RSSI，如果收到前导码的 RSSI 低于设置的 RSSI 门限值 （Reg2c），会复位解调器重新接收 0：disable 1：enable	1’b0

Bit	Name	Type	Description	Default
4	agc_en	R/W	AGC 使能： 0：disable 1：enable	1’b1
3:0	reserved	R/W	-	4’h9

4.24 Reg2B Address:0x2B Default:0xFF

Bit	Name	Type	Description	Default
7:6	reserved	R/W	-	2’b11
5:4	rxfe_gn	R/W	rxfe 增益控制	2’b11
3:2	filter_gn	R/W	Filter 增益控制	2’b11
1:0	pga_gn	R/W	PGA 增益控制	2’b11

4.25 Reg2C Address:0x2C Default:0xFF

Bit	Name	Type	Description	Default
7:0	rx_rssi_thr	R/W	接收 RSSI 门限值，参考 Reg28[5]	8’hFF

4.26 Reg2D Address:0x2D Default:0x10

Bit	Name	Type	Description	Default
7	cw_mode	R/W	发射载波模式： 0：disable 1：enable	1’b0
6:4	reserved	R/W	-	3’b001
3	ook_en	R/W	调制解调模式选择： 0：FSK 1：OOK	1’b0
2:0	reserved	R/W	-	3’b000

4.27 Reg30 Address:0x30 Default:0x80

Bit	Name	Type	Description	Default
7	sel_gau_o	R/W	发射高斯滤波器使能： 0：disable 1：enable	1’b1
6:4	reserved	R/W	-	3'b000
3:0	dev_set[11:8]	R/W	发射调制频偏 dev_set 的高 4 位	4’h0

4.28 Reg31 Address:0x31 Default:0x66

Bit	Name	Type	Description	Default
7:0	dev_set[7:0]	R/W	发射调制频偏 dev_set[11:0]={Reg30[3:0],Reg31[7:0} 例：频偏为正负 25KHz 即 0.025MHz，则 dev_set = 0.025*4096	8'h66

4.29 Reg37 Address:0x37 Default:0x06

Bit	Name	Type	Description	Default
7:0	csrst_len	R/W	在补偿功能使能时，找到信号的频谱后连续收到 Reg37*4 个前导后，则认为找到有效的 Preamble	8'h06

4.30 Reg38 Address:0x38 Default:0x91

Bit	Name	Type	Description	Default
7:5	reserved	R/W	-	3’b100
4	demod_comp_en	R/W	解调频偏补偿： 0：disable 1：enable	1’b1
3:0	reserved	R/W	-	4’h1

4.31 Reg44 Address:0x44 Default:0x05

Bit	Name	Type	Description	Default
7:6	reserved	R/W	-	2‘b00
5	wor_on	R/W	自动唤醒功能开启，即 RC32K 时钟模块使能： 0：disable 1：enable	1’b0
4	wor_trx_sel	R/W	自动唤醒后执行的命令： 0：RX 1：TX	1’b0
3:0	wor_clk_sel	R/W	自动唤醒功能计数器时钟分频选择： 0000：32k 0001：32k/2 0010：32k/4 --- 1111：32k/32768	4’h5

4.32 Reg45 Address:0x45 Default:0x00

Bit	Name	Type	Description	Default
7:0	wor_timer[15:8]	R/W	wor timer 高 8 位，参考 Reg46	8'h00

4.33 Reg46 Address:0x46 Default:0x60

Bit	Name	Type	Description	Default
7:0	wor_timer[7:0]	R/W	wor_timer 低 8 位， wor_time={Reg45,Reg46}为自动唤醒的计数周期，包括睡眠时间和工作时间	8’h60

4.34 Reg47 Address:0x47 Default:0x00

Bit	Name	Type	Description	Default
7:0	wor_rt_timer[15:8]	R/W	wor_rt_timer 高 8 位，参考 Reg48	8’h00

4.35 Reg48 Address:0x48 Default:0x3C

Bit	Name	Type	Description	Default
7:0	wor_rt_timer[7:0]	R/W	wor_rt_timer 低 8 位， wor_rt_timer={Reg47,Reg48}为唤醒后工作的时间	8’h3C

4.36 Reg4B Address:0x4B Default:0x45

Bit	Name	Type	Description	Default
7:6	rc32k_std[9:8]	R/W	RC32K 校准值	2'b01
5	rc32k_cal_en	R/W	RC32K 校准功能使能： 0：disable 1：enable	1'b0
4:0	reserved	R/W	-	5'h05

4.37 Reg4C Address:0x4C Default:0xF4

Bit	Name	Type	Description	Default
7:0	rc32k_std[7:0]	R/W	RC32K 校准值	8'hF4

4.38 Reg4D Address:0x4D Default:0x10

Bit	Name	Type	Description	Default
7	nirq_dir	R/W	nIRQ_flag IO 口方向： 0：output 1：input	1'b0
6	gpio_dir[0]	R/W	gpio[0] 方向： 0：output 1：input	1'b0
5	gpio_dir[1]	R/W	gpio[1] 方向： 0：output 1：input	1'b0
4	gpio_dir[2]	R/W	gpio[2] 方向，默认复位输入功能： 0：output 1：input	1'b1
3:0	nirq_sel	R/W	nIRQ 引脚输出功能选择： 0x0：pkt_flag(包括数据包完成中断 pkt_int, 接收前导码中断 preamble_int, 接收同步字中断 syncword_int) 0x1：pkt_int 0x2：preamble_int 0x3：syncword_int 0x4：fifo_flag 0x5：brclk（参考 Reg11[2:0]描述） 0x6：rxdata 0x7：txdata 0x8：tr_switch 0x9：tr_switch 反向 0xa：rxdata 0xb：txdata 0xc：wor_event 0xd：高电平 others: 低电平	4'h0

4.39 Reg4E Address:0x4E Default:0x45

Bit	Name	Type	Description	Default
7:4	gpio_0_sel	R/W	gpio[0]输出功能选择，参考 Reg4D[3:0]描述	4'h4
3:0	gpio_1_sel	R/W	gpio[1]输出功能选择，参考 Reg4D[3:0]描述	4'h5

4.40 Reg4F Address:0x4F Default:0x67

Bit	Name	Type	Description	Default
7:4	gpio_2_sel	R/W	gpio[2]输出功能选择，参考 Reg4D[3:0]描述	4'h6
3:0	reserved	R/W	-	4'h7

4.41 Reg50 Address:0x50 Default:0x00

Bit	Name	Type	Description	Default
7:0	rx_fifo_wr_ptr	R/W	RX FIFO 写指针，写 0x80 清除指针	8'h00

4.42 Reg51 Address:0x51 Default:0x00

Bit	Name	Type	Description	Default
7:0	rx_fifo_rd_ptr	R/W	RX FIFO 读指针，写 0x80 清除指针	8'h00

4.43 Reg52 Address:0x52 Default:0x00

Bit	Name	Type	Description	Default
7:0	rx_fifo_data	R/W	RX FIFO	8'h00

4.44 Reg53 Address:0x53 Default:0x00

Bit	Name	Type	Description	Default
7:0	tx_fifo_wr_ptr	R/W	TX FIFO 写指针，写 0x80 清除指针	8'h00

4.45 Reg54 Address:0x54 Default:0x00

Bit	Name	Type	Description	Default
7:0	tx_fifo_rd_ptr	R/W	TX FIFO 读指针，写 0x80 清除指针	8'h00

4.46 Reg55 Address:0x55 Default:0x00

Bit	Name	Type	Description	Default
7:0	tx_fifo_data	R/W	TX FIFO	8'h00

4.47 Reg56 Address:0x56 Default:0xB7

Bit	Name	Type	Description	Default
7:0	delta_rssi	R/W	计算 RSSI 时的预偏移量	8'hB7

4.48 Reg57 Address:0x57 Default:0x00

Bit	Name	Type	Description	Default
7:3	reserved	R/W	-	5'h00
2	spi_3w_sel	R/W	三线 SPI 选择： 0：4 线标准 SPI 1：3 线 SPI，数据线只使用 SPI_MOSI	1'b0
1	crc_err_st	R/W	接收 CRC 错误时，数据包的处理方法： 0：正常完成该包的接收，退出接收状态发出数据包中断和错误 CRC 标志 1：丢弃该包数据并重新进入接收状态	1'b0
0	reserved	R/W	-	1'b0

4.49 Reg58 Address:0x58 Default:0x00

Bit	Name	Type	Description	Default
7:5	reserved	R/W	-	3'b000
4:3	direct_tx_data_sel	R/W	direct mode 时，TX 数据源选择： 00：nIRQ 引脚 01：gpio[0] 1x：gpio[1]	2'b00
2:0	reserved	R/W	-	3'b000

4.50 Reg5A Address:0x5A Default:0x0F

Bit	Name	Type	Description	Default
7:4	reserved	R/W	-	4'h0
3:1	wor_ext_mode	R/W	自动唤醒模式下接收到以下有效信号自动退出休眠，（该功能需要 Reg5A[0]使能） 100：接收 RSSI 有效 010：接收 Syncword 有效 001：接收 Preamble 有效	3'b111
0	wor_ext_en	R/W	wor_ext_mode 使能： 0：disable 1：enable	1'b1

4.51 Reg5D Address:0x5D Default:0x00

Bit	Name	Type	Description	Default
7	nrst_sel	R/W	nRST 引脚功能设置： 0：nRST 1： GPIO2	1'b0
6	pull_up_en	R/W	nRST 引脚 Pull-UP 使能： 0：enable 1：disable	1'b0
5:0	reseved	R/W	-	5'h00

4.52 Reg60 Address:0x60 Default:0x80

Bit	Name	Type	Description	Default
7:0	command	R/W	0x80：IDLE command 0x40：TX Command 0x20：RX command 0x05：StandBy command 0x06：Reset command, 需要先使能 RegA3[7] 0x07：wor command 自动唤醒 0x10：FS command 打开频综	8'h80

4.53 Reg61 Address:0x61 Default:0x00

Bit	Name	Type	Description	Default
7	syncword_rec	R	接收到正确的 syncword	1'b0
6	preamble_rec	R	接收到有效的 preamble	1'b0
5	crc_error	R	高电平时表示 CRC 校验错误	1'b0
4	pkt_flag	R	pkt_flag	1'b0
3	fifo_flag	R	fifo_flag	1'b0
2:0	reserved	R	-	3'b000

4.54 Reg64 Address:0x64 Default:0x00

Bit	Name	Type	Description	Default
7:0	rssi	R	读出的值为实际 RSSI 的绝对值	8'h00

4.55 Reg66 Address:0x66 Default:0x37

Bit	Name	Type	Description	Default
7	rc32k_cal_done	R	RC32k 校准完成	1'b0
6:0	rc32k_ftrim	R	RC32K 校准完成后的 trimming 值	7'h37

4.56 Reg67 Address:0x67 Default:0x80

Bit	Name	Type	Description	Default
7	st_idle	R	IDLE 状态	1'b1
6	st_tx	R	发射状态	1'b0
5	st_rx	R	接收状态	1'b0
4	st_fson	R	频综开启状态	1'b0
3:0	reseved	R	-	4'h0

4.57 Reg68 Address:0x68 Default:0x00

Bit	Name	Type	Description	Default
7:2	reseved	R	-	6'h0
1	m3_seqnum_ok	R	数据包模式 3 时，接收 seqnum 匹配正确	1'b0
0	m3_addr_comp_ok	R	数据包模式 3 时，接收 address 匹配正确	1'b0

4.58 Reg69 Address:0x69 Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_rx_fcs2_value	R	数据包模式 3 时，接收 FCS2 数据	8'h00

4.59 Reg6A Address:0x6A Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_rx_payload_len[7:0]	R	数据包模式 3 时，接收 length 数据	8'h00

4.60 Reg6B Address:0x6B Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_rx_payload_len[15:8]	R	数据包模式 3 时，接收 length 数据	8'h00

4.61 Reg6C Address:0x6C Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_rx_addr[7:0]	R	数据包模式 3 时，接收 address	8'h00

4.62 Reg6D Address:0x6D Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_rx_addr[15:8]	R	数据包模式 3 时，接收 address	8'h00

4.63 Reg6E Address:0x6E Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_rx_addr[23:16]	R	数据包模式 3 时，接收 address	8'h00

4.64 Reg6F Address:0x6F Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_rx_addr[31:24]	R	数据包模式 3 时, 接收地址域数据	8'h00

4.65 Reg70 Address:0x70 Default:0xFF

Bit	Name	Type	Description	Default
7:0	scramble_init_val[7:0]	R/W	scramble 初始化值	8'hFF

4.66 Reg71 Address:0x71 Default:0x09

Bit	Name	Type	Description	Default
7	reserved	R/W	-	1'b0
6	srcamble_len	R/W	伪随机数长度选择： 0：PN9 1：PN7	1'b0
5	scramble_msb	R/W	选择首先输出的移位寄存器： 0：LSB 1：MSB	1'b0
4	scramble_type	R/W	伪随机数产生器类型： 0：抽出模式 1：插入模式	1'b0
3:1	scramble_poly	R/W	反馈抽头选择	3'b100
0	scramble_init_val[8]	R/W	scramble 初始化值最高位	1'b1

4.67 Reg72 Address:0x72 Default:0x01

Bit	Name	Type	Description	Default
7:5	reserved	R/W	-	3'b000
4:0	sync_thres	R/W	同步字检测允许的错误位数	5'h01

4.68 Reg73 Address:0x73 Default:0x55

Bit	Name	Type	Description	Default
7:0	preamble_val	R/W	发射前导码时的使用的数据	8'h55

4.69 Reg74 Address:0x74 Default:0x03

Bit	Name	Type	Description	Default
7	auto_ack	R/W	auto_ack 使能： 0：disable 1：enable	1'b0

Bit	Name	Type	Description	Default
6	ack_payload_en	R/W	发射 ACK 时带 payload 数据返回： 0：disable 1：enable	1’b0
5	cont_rx_en	R/W	控制接收状态机： 0：接收完成数据包后退出接收状态进入 IDLE 状态 1：接收完成数据包后重新再次进入接收状态	1’b0
4	sync_man_en	R/W	同步字曼彻斯特编码使能： 0：disable 1：enable	1’b0
3	sync_bit_order	R/W	同步字 bit 顺序： 0：字节低位 1：字节高位 MSB	1’b0
2:0	sync_len	R/W	同步字长度： 000：{sync_id_1} 001：{sync_id_1, sync_id_2} 010：{sync_id_1, sync_id_2, sync_id_3} 011：{sync_id_1, sync_id_2, sync_id_3, sync_id_4} 100：{sync_id_1, sync_id_2, sync_id_3, sync_id_4, sync_id_5} 101：{sync_id_1, sync_id_2, sync_id_3, sync_id_4, sync_id_5, sync_id_6} 110：{sync_id_1, sync_id_2, sync_id_3, sync_id_4, sync_id_5, sync_id_6, sync_id_7} 111：{sync_id_1, sync_id_2, sync_id_3, sync_id_4, sync_id_5, sync_id_6, sync_id_7, sync_id_8}	3’b011

4.70 Reg75 Address:0x75 Default:0xA7

Bit	Name	Type	Description	Default
7:0	sync_id_1	R/W	同步字字节 1	8’hA7

4.71 Reg76 Address:0x76 Default:0xA7

Bit	Name	Type	Description	Default
7:0	sync_id_2	R/W	同步字字节 2	8’hA7

4.72 Reg77 Address:0x77 Default:0x98

Bit	Name	Type	Description	Default
7:0	sync_id_3	R/W	同步字字节 3	8’h98

4.73 Reg78 Address:0x78 Default:0xF3

Bit	Name	Type	Description	Default
7:0	sync_id_4	R/W	同步字字节 4	8'hF3

4.74 Reg79 Address:0x79 Default:0x98

Bit	Name	Type	Description	Default
7:0	sync_id_5	R/W	同步字字节 5	8'h98

4.75 Reg7A Address:0x7A Default:0xF3

Bit	Name	Type	Description	Default
7:0	sync_id_6	R/W	同步字字节 6	8'hF3

4.76 Reg7B Address:0x7B Default:0x98

Bit	Name	Type	Description	Default
7:0	sync_id_7	R/W	同步字字节 7	8'h98

4.77 Reg7C Address:0x7C Default:0xF3

Bit	Name	Type	Description	Default
7:0	sync_id_8	R/W	同步字字节 8	8'hF3

4.78 Reg7D Address:0x7D Default:0x0E

Bit	Name	Type	Description	Default
7:0	payload_len[7:0]	R/W	payload 长度，在数据包模式 1 和数据包模式 3 有效	8'h0E

4.79 Reg7E Address:0x7E Default:0x00

Bit	Name	Type	Description	Default
7:0	payload_len[15:8]	R/W	payload 长度，在数据包模式 1 和数据包模式 3 有效	8'h00

4.80 Reg7F Address:0x7F Default:0x00

Bit	Name	Type	Description	Default
7:1	reserved	R/W	-	7'h0
0	page_sel	R/W	寄存器页地址： 0：Reg00~Reg7E 1：Reg80~RegFE	1'b0

4.81 Reg80 Address:0x80 Default:0x1D

Bit	Name	Type	Description	Default
7:6	reserved	R/W	-	2'b00
5	m3_length_en	R/W	数据包模式 3 时，数据长度域控制： 0：disable 1：enable	1'b0
4	m3_crc_sel	R/W	数据包模式 3 时，CRC 的校验范围： 0：All payload 1：仅 FIFO 中的数据	1'b1
3:0	m3_ack_node_ctrl	R/W	数据包模式 3 且 AUTO_ACK 功能开启，返回的 ACK 包控制： [3]：为'1'时使能返回接收到的数据长度 [2]：为'1'时使能返回接收到的地址 [1]：为'1'时使能返回接收到的 seqnum [0]：为'1'时使能返回接收到的 FCS2 数据	4'hD

4.82 Reg81 Address:0x81 Default:0x78

Bit	Name	Type	Description	Default
7:6	reserved	R/W	-	2'b01
5:4	m3_addr_size	R/W	数据包模式 3 时，地址长度： 00：1 字节 01：2 字节 10：3 字节 11：4 字节	2'b11
3	reserved	R/W	-	1'b1
2	m3_addr_split_mode	R/W	数据包模式 3 时，将地址分为源地址和目的地址，在返回 ACK 时，将收到的源地址和目的地址位置互换： 0：不互换 1：互换	1'b0
1	m3_addr_pos_sel	R/W	数据包模式 3 时，地址位置选择： 0：address 放在 length 之后 1：address 放在 length 之前	1'b0
0	m3_addr_en	R/W	数据包模式 3 时，地址域控制： 0：disable 1：enable	1'b0

4.83 Reg82 Address:0x82 Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_addr_val[7:0]	R/W	数据包模式 3 时，地址值	8'h00

4.84 Reg83 Address:0x83 Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_addr_val[15:8]	R/W	数据包模式 3 时，地址值	8'h00

4.85 Reg84 Address:0x84 Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_addr_val[23:16]	R/W	数据包模式 3 时，地址值	8'h00

4.86 Reg85 Address:0x85 Default:0x03

Bit	Name	Type	Description	Default
7:0	m3_addr_val[31:24]	R/W	数据包模式 3 时, 地址值	8'h03

4.87 Reg86 Address:0x86 Default:0x33

Bit	Name	Type	Description	Default
7:0	m3_addr_bit_mask[7:0]	R/W	数据包模式 3 时，地址过滤，mask 为'1'时对应地址位不作比较	8'h33

4.88 Reg87 Address:0x87 Default:0x33

Bit	Name	Type	Description	Default
7:0	m3_addr_bit_mask[15:8]	R/W	数据包模式 3 时，地址过滤，mask 为'1'时对应地址位不作比较	8'h33

4.89 Reg88 Address:0x88 Default:0x10

Bit	Name	Type	Description	Default
7:0	m3_addr_bit_mask[23:16]	R/W	数据包模式 3 时，地址过滤，mask 为'1'时对应地址位不作比较	8'h10

4.90 Reg89 Address:0x89 Default:0x00

Bit	Name	Type	Description	Default
7:0	m3_addr_bit_mask[31:24]	R/W	数据包模式 3 时，地址过滤，mask 为‘1’时对应地址位不作比较	8’h00

4.91 Reg8A Address:0x8A Default:0x00

Bit	Name	Type	Description	Default
7:6	reserved	R/W	-	2’b00
5	m3_seqnum_auto_inc	R/W	数据包模式 3 时，seqnum 配置方式： 0：seqnum 数据由{Reg8c,Reg8b}配置 1：由内部 seqnum 计数器自动加 1	1’b0
4	m3_seqnum_en	R/W	数据包模式 3 时，seqnum 数据使能： 0：disable 1：enable	1’b0
3	m3_seqnum_bit_size	R/W	数据包模式 3 时，seqnum 数据长度： 0：8bit 1：16bit	1’b0
2:0	reserved	R/W	-	3’b000

4.92 Reg8B Address:0x8B Default:0x00

Bit	Name	Type	Description	Default
7:0	seqnum_Reg[7:0]	R/W	数据包模式 3 时，seqnum 值	8’h00

4.93 Reg8C Address:0x8C Default:0x50

Bit	Name	Type	Description	Default
7:0	seqnum_Reg[15:8]	R/W	数据包模式 3 时，seqnum 值	8’h50

4.94 Reg8D Address:0x8D Default:0x30

Bit	Name	Type	Description	Default
7:1	reserved	R/W	-	7’h30
0	m3_fcs2_en	R/W	数据包模式 3 时，FCS2 数据域使能： 0：disable 1：enable	1’b0

4.95 Reg8E Address:0x8E Default:0x32

Bit	Name	Type	Description	Default
7	m3_fcs2_val	R/W	数据包模式 3 时，ACK request	1’b0
6:0	reserved	R/W	-	7’h32

4.96 Reg94 Address:0x94 Default:0x08

Bit	Name	Type	Description	Default
7:5	reserved	R/W	-	3’b000
4	crc_bit_order	R/W	CRC bit 顺序： 0：LSB 1：MSB	1'b0
3	crc_inv	R/W	CRC bit 取反： 0：disable 1：enable	1'b1
2	crc_man_en	R/W	CRC 曼彻斯特编码使能： 0：disable 1：enable	1'b0
1:0	crc_len	R/W	CRC 长度： 00：8 位 01：16 位 10：24 位 11：32 位	2'b00

4.97 Reg95 Address:0x95 Default:0x20

Bit	Name	Type	Description	Default
7:0	crc_poly[7:0］	R/W	CRC 多项式配置	8’h20

4.98 Reg96 Address:0x96 Default:0x26

Bit	Name	Type	Description	Default
7:0	crc_poly[15:8］	R/W	CRC 多项式配置	8’h26

4.99 Reg97 Address:0x97 Default:0x1F

Bit	Name	Type	Description	Default
7:0	crc_poly[23:16］	R/W	CRC 多项式配置	8’h1F

4.100 Reg98 Address:0x98 Default:0x68

Bit	Name	Type	Description	Default
7:0	crc_poly[31:24]	R/W	CRC 多项式配置	8'h68

4.101 Reg99 Address:0x99 Default:0x04

Bit	Name	Type	Description	Default
7:0	crc_init_val[7:0]	R/W	CRC 移位寄存器初始化值	8'h04

4.102 Reg9A Address:0x9A Default:0xAE

Bit	Name	Type	Description	Default
7:0	crc_init_val[15:8]	R/W	CRC 移位寄存器初始化值	8'hAE

4.103 Reg9B Address:0x9B Default:0x89

Bit	Name	Type	Description	Default
7:0	crc_init_val[23:16]	R/W	CRC 移位寄存器初始化值	8'h89

4.104 Reg9C Address:0x9C Default:0x84

Bit	Name	Type	Description	Default
7:0	crc_init_val[31:24]	R/W	CRC 移位寄存器初始化值	8'h84

4.105 Reg9D Address:0x9D Default:0x03

Bit	Name	Type	Description	Default
7	tx_data_inv	R/W	发射 bit 数据取反	1'b0
6	rx_data_inv	R/W	解调器输出的 bit 数据取反	1'b0
5	auto_det_tx_chl	R/W	发射前检测信道 RSSI： 0：disable 1：enable	1'b0
4	auto_det_tx_mode	R/W	发射检测信道模式： 0：检测次数满且信道仍然忙，退出发射模式，发射数据失败 1：持续检测，直到发射数据成功	1'b0
3:0	auto_det_times	R/W	发射前最大检测信道次数	4'h3

4.106 Reg9E Address:0x9E Default:0x80

Bit	Name	Type	Description	Default
7:0	chl_busy_thr	R/W	发射前检测信道门限值	8'h80

4.107 Reg9F Address:0x9F Default:0x26

Bit	Name	Type	Description	Default
7:6	reserved	R/W	-	2'b00
5:0	fifo_empty_thr	R/W	TX FIFO 空门限，FIFO 数据低于门限值时 fifo_flag 置'1'	6'h26

4.108 RegA0 Address:0xA0 Default:0x50

Bit	Name	Type	Description	Default
7:6	reserved	R/W	-	2'b00
5:0	fifo_full_thr	R/W	RX FIFO 满门限，RX FIFO 数据高于门限值时 fifo_flag 置'1'	6'h50

4.109 RegA3 Address:0xA3 Default:0x42

Bit	Name	Type	Description	Default
7	srst_cmd_enable	R/W	Reset Command 使能： 0：disable 1：enable	1'b0
6	reserved	R/W	-	1'b1
5:0	auto_ack_wait_time	R/W	auto ack 功能开启时，发射端发完数据包后等待 ACK 数据的时间，超时会重新发射数据	6'h02

4.110 RegE0 Address:0xE0 Default:0x80

Bit	Name	Type	Description	Default
7:0	rx_seqnum_val[7:0]	R	数据包模式 3 时，接收 seqnum 数据	8'h80

4.111 RegE1 Address:0xE1 Default:0x00

Bit	Name	Type	Description	Default
7:0	rx_seqnum_val[15:8]	R	数据包模式 3 时，接收 seqnum 数据	8'h00

4.112 RegE2 Address:0xE2 Default:0x01

Bit	Name	Type	Description	Default
7:0	seqnum_cnt[7:0]	R	seqnum_cnt	8’h01

4.113 RegE3 Address:0xE3 Default:0x40

Bit	Name	Type	Description	Default
7:0	seqnum_cnt[15:8]	R	seqnum_cnt	8’h40

5 功能描述

该芯片是一款高集成度的 sub-GHz 无线收发机。支持 OOK，2-(G)FSK 调制解调方式，支持 Direct 和 Packet 数据处理模式，支持标准四线 SPI 和自定义三线 SPI 接口。

5.1 接收机

该芯片内部集成了低功耗、高性能的低中频架构接收机。天线接收到的射频信号经过低噪声放大器放大之后，由正交混频器下变频至中频。I/Q 两路中频信号进一步通过低通滤波器（LPF）、可编程放大器（PGA）放大到合适的幅度，然后由 ADC 转换到数字域。

接收到的信号在数字域完成镜像抑制、中频滤波、基带滤波和 OOK/(G)FSK 解调。解调后的数据在直通模式下可以通过 GPIO 管脚直接输出，也可以在包模式下通过 SPI 从内部 FIFO 中读取。

5.2 发射机

该芯片发射机采用基于频率综合器的单点调制结构。内部的高效率功率放大器可以输出最大+18dBm 的功率，输出功率可以在-20dBm 至+18dBm 范围内调节，调节精度为 1dB。在(G)FSK 模式下，发射数据经过高斯滤波之后再送入频率综合器进行调制，使得发射频谱更为集中。为了降低 PA 开关过程中引起的频谱杂散和毛刺，并削弱对 VCO 的牵引，PA 的输出功率引入了缓慢升降机制（PA Ramp）。

5.3 频率综合器

该芯片内部集成了高精度的 Sigma-Delta 小数分频频率综合器，在 200MHz~960MHz 范围内产生精准的载波频率。载波频率可以通过两种方式进行设置：

1、 寄存器直接设置：在这种方式下，把所需的载波频率值直接写入相应的寄存器即可。

2、 查表方式。在这种方式下，可以按照以下公式进行频率设置：

$$F_C=F_0+N*F_{step}$$

其中，F_0、N 和 F_{step} 都可以通过寄存器进行设置。

- F_0 由{Reg00[6:0],Reg01,Reg02,Reg03}共 31bit 设置，高 11 位表示整数，低 20 位表示小数，单位为 MHz。
- N 为整数由 Reg04 设置。
- F_{step} 由{Reg05,Reg06,Reg07}设置，低 20bit 为小数，单位为 MHz。

如设置 F_c=433.92MHz，可以设置 F_0 = 433.92, N = 0, F_{step} = 0。即写入以下寄存器值：

Reg00=0x1B

Reg01=0x1E

Reg02=0xB8

Reg03=0x51

Reg04=0x00

Reg05=0x00

Reg06=0x00

Reg07=0x00

5.4 AGC

芯片的接收通路集成了 AGC 功能，Mixer、LPF、PGA 的增益受 AGC 环路调节。AGC 的环路控制在数字域完成，通过设定合理的 AGC 控制参数，芯片接收机的灵敏度、选择性和线性度可以达到最佳的性能。

5.5 RSSI

芯片内部集成的输入信号强度指示（RSSI）功能可以对天线端接收到的信号强度进行评估。RSSI 检测必须在 RX 状态下进行，检测到的是信道内的信号强度。

UM2010 内置信号检测模块，用于接收数据包时测量 ADC 的数据信号强度 RSSI_RAW，在 AGC(自动增益控制)使能时，使用 RSSI_RAW 值对接收链路增益自动调节，同时可自动计算出 RF 输入端的 RSSI(信号强度指示)值。也可以使用 RSSI_RAW 值手动调节（AGC 关闭）接收链路增益，并根据增益值自行计算 RSSI。RSSI_RAW 值或自动计算的 RSSI 值可以从寄存器 Reg64 读取。

5.5.1 RSSI 相关寄存器

RSSI 相关寄存器如下表所示：

表 5-1：RSSI 相关寄存器列表

Address	Bit	R/W	Name	Description
0x10	3	R/W	rssi_val_sel	读取 RSSI 时，选择读取的数据内容： 0：经过计算的 RSSI 值 1：原始信号幅度值
0x28	6	R/W	rssi_sel	收到同步字后锁定 RSSI 值： 0：disable 1：enable

Address	Bit	R/W	Name	Description
	5	R/W	rx_rssi_thr_en	接收时判断 RSSI，如果收到前导码的 RSSI 低于设置的 RSSI 门限值 （Reg2C)，会复位解调器重新接收： 0：disable 1：enable
	4	R/W	agc_en	AGC 使能： 0：disable 1：enable
0x2B	5:0	R/W	rx_gain	每个 step 为 6dbm,AGC 使能时该寄存器只读。
0x2C	7:0	R/W	rx_rssi_thr	接收 RSSI 门限值，参考 Reg28[5]
0x56	7:0	R/W	delta_rssi	计算 RSSI 时的预偏移量
0x64	7:0	R	rssi	读出的值为实际 RSSI 的绝对值

5.5.2 自动计算 RSSI

自动计算 RSSI 的寄存器优化值设置：

- Reg10[3] = 0
- Reg28[4] = 1
- Reg56 = 0xB7

在进入接收状态后，可以读取 Reg64 寄存器的值 rssi_value[7:0]，步进为 0.5dBm，且是真实信号 RSSI 的绝对值。

例如读取 Reg64 = 0x65，则 RSSI= -(rssi_value/2) = -50.5dBm。

如下图 F_{RF}=433.92MHz DR=10kbps F_{DEV}=22K 实测的 RSSI 值对应的输入功率。从图上看[-120dBm,-35 dBm]的线性区间内，输入功率值与读出的 RSSI 值有很好的对应关系，如果在某些频段或速率在线性区间内读出的值与输入功率值有固定的差值，可以写寄存器 Reg56 去微调偏移量。

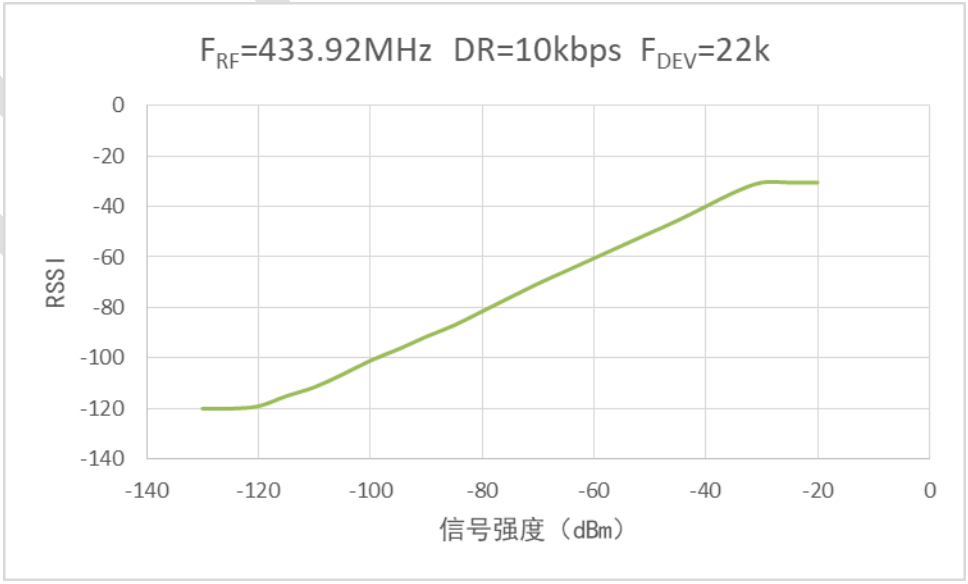


图 5-1：RSSI 线性图

UM2010 在整个接收数据包期间，RSSI 值实时更新到寄存器 Reg64，当退出接收状态 Reg64 值也恢复到 0；如果设置寄存器 Reg28[6]=1，则收到同步字后寄存器 Reg64 值锁定，RSSI 值不再更新，直到下次重新进入接收时 RSSI 才会被更新。

5.5.3 手动调节增益

在正常(G)FSK 模式时，都使用自动增益调节。在 OOK 模式时，可以手动调节增益并计算 RSSI 值。寄存器优化值设置如下：

- Reg10[3] = 1
- Reg28[6] = 0
- Reg28[4] = 0
- Reg2B[5:0] = 0x3F

进入接收状态后，读取 Reg64 的值，根据 Reg64 的值上下调节 Reg2B 的增益控制。Reg2B 对应增益表如下：

表 5-2：Reg2B 增益控制表

Reg2B[5:0]	Gain(dBm)
111111	54
111110	48
111101	42
111100	36
111000	30
110100	24
110000	18
100000	12
010000	6
000000	0

RSSI 计算公式：RSSI= (Reg64/2) – Gain + delta_rssi

5.5.4 RSSI 自动过滤功能

RSSI 自动过滤功能，即设置 RSSI 门限值，在解调器接收到前导信号时，会判断 RSSI 值，如接收到的信号强度低于设置的门限，则解调器自动复位，丢弃该信号重新进入接收状态。

如设置信号强度低于-96dBm 时，丢弃该包数据，则进入接收之前设置：

- Reg28[5] = 1
- Reg2C[7:0] = 0xC0

5.6 BUCK DCDC

该芯片内部集成了高效率的 BUCK DCDC converter。BUCK DCDC converter 有三种工作模式：BUCK 模式、旁路模式、关闭模式。不同的工作模式可以通过寄存器进行控制（数字寄存器读写不受 BUCK 工作模式的影响）。外围电路只需要添加一个 4.7μH 的电感和一个 2.2μF 电容，就可以把芯片配置成 BUCK 工作模式，在 BUCK 模式下，芯片的功耗可以显著降低。BUCK 的输出电压可以在 1.4V~1.7V 范围进行设置。在旁路模式下，SW 脚和 VDD 之间通过芯片内部的 PMOS 开关直接连通。

5.6.1 电源管理

UM2010 内部的电源管理单元如下图所示：

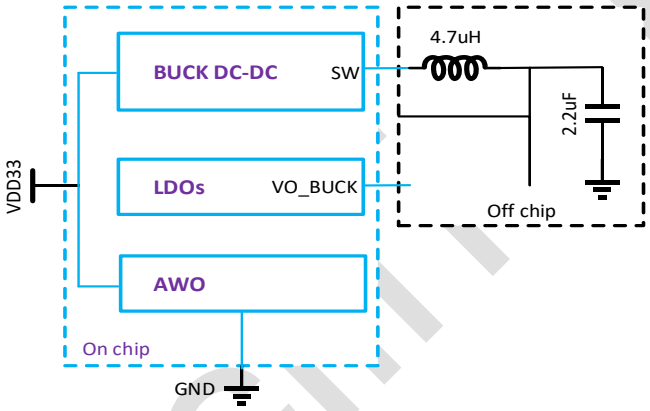


图 5-2：电源管理单元示意图

5.6.2 电源工作模式

UM2010 提供 BUCK 及非 BUCK 两种电源工作模式。

5.6.2.1 BUCK 模式

芯片 SW 脚和 VO_BUCK 脚之间接 4.7μH 功率电感，VO_BUCK 脚外接 2.2μF 对地电容。在这种模式下，芯片内部的 LDO 由 VO_BUCK 供电，AWO 模块（Always ON 模块，除了 shutdown 状态外，AWO 一直给芯片内部的数字模块供电）由外部电源直接供电。参考应用电路如下：

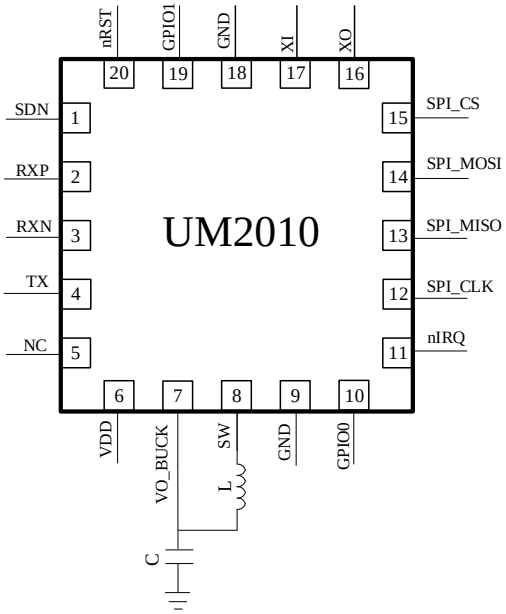


图 5-3： BUCK 工作模式参考应用

BUCK 工作模式下，芯片内部的 BUCK DC-DC 可配置成两种工作状态：

- BUCK 模式工作状态（Reg1B[1:0]=01）

  BUCK 模式工作状态下，BUCK DC-DC 将外部的输入电源电压转换成一个更低的输出电压从而提高电源效率，降低芯片的整体耗电电流。BUCK DC-DC 输出电压可以通过寄存器进行配置：

表 5-3：BUCK 输出电压设置

Reg1B[3:2]	Vout
00	1.4V
01	1.5V
10	1.6V
11	1.7V

  另外，BUCK DC-DC 可提供的最大峰值电流也可以通过寄存器进行配置：

表 5-4：BUCK 峰值电流设置

Reg1B[6:4]	Ipk
000	80mA
001	110mA
010	140mA
011	170mA
100	200mA
101	230mA
110	260mA
111	290mA

  在相同的外部供电电压条件下，BUCK DC-DC 的输出电压设得越低，整体接收电流越小。

- Bypass 模式（Reg1B[1:0]=10）

  在 Bypass 模式下，SW 脚和 VDD 之间通过芯片内部的 PMOS 开关直接连通。其工作模式类似

于非 BUCK 模式。BUCK 模式的状态机切换和非 BUCK 模式基本相同。唯一的区别在于：BUCK 模式在进入 standby 状态之前，需要先关闭 BUCK DC-DC，从 standby 状态唤醒进入 IDLE 状态之后再重新开启 BUCK DC-DC。

5.6.2.2 非 BUCK 模式

外部的 4.7μH 功率电感不需要，芯片 SW 脚悬空，VO_BUCK 脚和外部电源直连。在这种模式下，芯片内部的 LDO 由外部电源直接供电，BUCK DC-DC 需配置在关闭模式（Reg1B[1:0]=00）。参考应用电路如下：

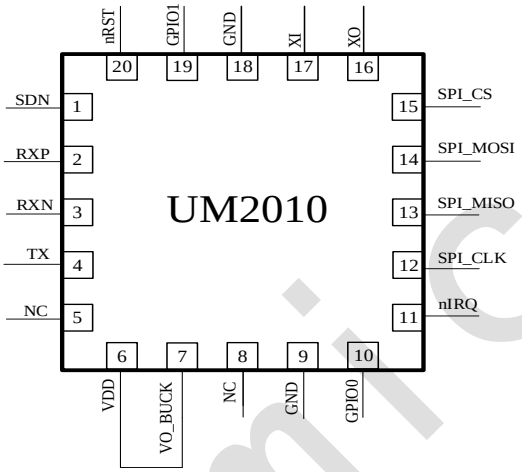


图 5-4：非 BUCK 工作模式应用参考

5.7 WOR 定时唤醒

芯片内部集成了定时唤醒器，内置一个 32KHz 频率的 RC 振荡器和一个 WOR 计数器。定时器采用内部的低功耗 32KHz 时钟源来运行。其唤醒过程可由下图所示。使能 WOR 命令后，芯片在工作状态和 Standby 状态之间自动定时切换，在 WOR time 时间窗口内，芯片处于 Standby 状态。在 TRX time 时间窗口内，芯片处于接收或发射状态（可通过寄存器来配置，WOR time 唤醒后芯片是进入 RX 状态还是 TX 状态）。WOR time 和 TRX time 的窗口时间都可以通过寄存器来设定。32KHz 时钟源具有自动校准功能。

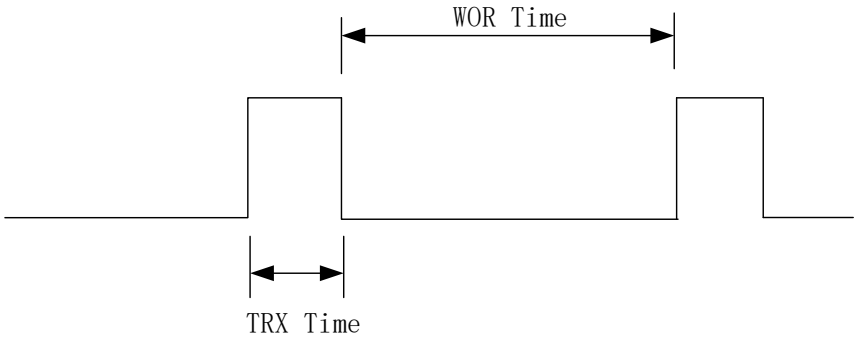


图 5-5：定时唤醒图

5.7.1 WOR 相关寄存器

表 5-5：WOR 相关寄存器列表

Address	Bit	R/W	Name	Description
0x44	5	R/W	wor_on	自动唤醒功能开启，即 RC32K 时钟模块使能： 0：disable 1：enable
	4	R/W	wor_trx_sel	自动唤醒后执行的命令： 0：RX 1：TX
	3:0	R/W	wor_clk_sel	自动唤醒功能计数器时钟分频选择： 0000：32k 0001：32k/2 0010：32k/4 … 1111：32k/32768
0x45	7:0	R/W	wor_timer[15:8]	wor_timer 为自动唤醒功能计数周期，包括 standby 时间和工作时间
0x46	7:0	R/W	wor_timer[7:0]	
0x47	7:0	R/W	wor_rt_timer[15:8]	wor_rt_timer 为唤醒后工作时间的计数长度
0x48	7:0	R/W	wor_rt_timer[7:0]	
0x4B	7:6	R/W	rc32k_std[9:8]	RC32K 校准值
	5	R/W	rc32k_cal_en	RC32K 校准功能使能： 0：disable 1：enable
0x4C	7:0	R/W	rc32k_std[7:0]	RC32K 校准值
0x5A	3:1	R/W	wor_ext_mode	自动唤醒模式下接收到以下有效信号自动退出休眠（该功能需要 Reg5A[0]使能）： 100：接收 RSSI 有效 010：接收 Syncword 有效 001：接收 Preamble 有效

Address	Bit	R/W	Name	Description
	0	R/W	wor_ext_en	wor_ext_mode 使能： 0：disable 1：enable
0x60	7:0	R/W	command	0x07：wor command 自动唤醒
0x66	7	R	rc32k_cal_done	RC32k 校准完成标志
	6:0	R	rc32k_ftrim	RC32K 校准完成后的 trimming 值

5.7.2 RC32K 模块校准说明

开启 WOR 功能会自动开启 RC32K 模块，所以开启 WOR 功能前配置好 32K 的校准相关寄存器。RC32K 校准寄存器 rc32k_std[9:0]的配置与参考时钟相关，计算方式如下：

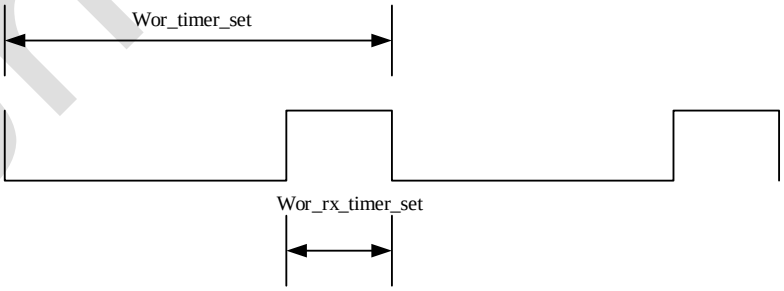
rc32k_std = (xtal_clk/2)/32000

xtal_clk 为参考时钟，如 30MHz，则计算的结果约为 469（HEX: 0x01D5），RC32K 配置流程如下：

1. Reg4A = 0x05
2. Reg4B = 0xE5
3. Reg4C = 0xD5
4. Reg44 = 0x25 //RC32K 开启
5. 等待 Reg66[7]为 1 则校准完成。

5.7.3 WOR 唤醒周期

WOR 唤醒周期由唤醒计数器 wor_timer 控制，该计数器的时钟源为 RC32K 或 RC32K 的分频时钟，由寄存器 wor_clk_sel 选择。



wor_timer_set 时间包括 standby 时间和工作时间，standby 时间由寄存器 wor_timer[15:0]配置；工作时间由寄存器 wor_rt_time[15:0]配置。在 RC32K 模块启动完成后，可写 WOR 命令启动 WOR 功能：Reg60 = 0x07。

5.7.4 WOR_EXT 模式

WOR 在正常唤醒工作固定的时间后会重新进入 standby 状态，如果在工作状态收到完整的数据包后会退出 WOR 功能。为了尽量减少工作的时间还能收到正确的数据包，可以使能 wor_ext 功能(Reg5A[0])，即在收到有效的 RSSI、Preamble 或 Syncword（Reg5A[3:1]）后，即使工作时间到也不退回到 standby 状态，直到接收完数据退出 WOR 命令。

5.8 系统复位

该芯片内部集成了上电复位（POR）、外部复位和软件复位。上电默认状态下，GPIO2 默认为外部复位信号(nRST)输入，只要 nRST 拉低就可以完成对芯片的复位。此外，还可以通过 SPI 写命令的方式对芯片进行复位操作。芯片一旦复位，内部所有配置都恢复到上电默认状态，MCU 需要对芯片重新进入初始化操作。

6 芯片运行

6.1 状态机控制图

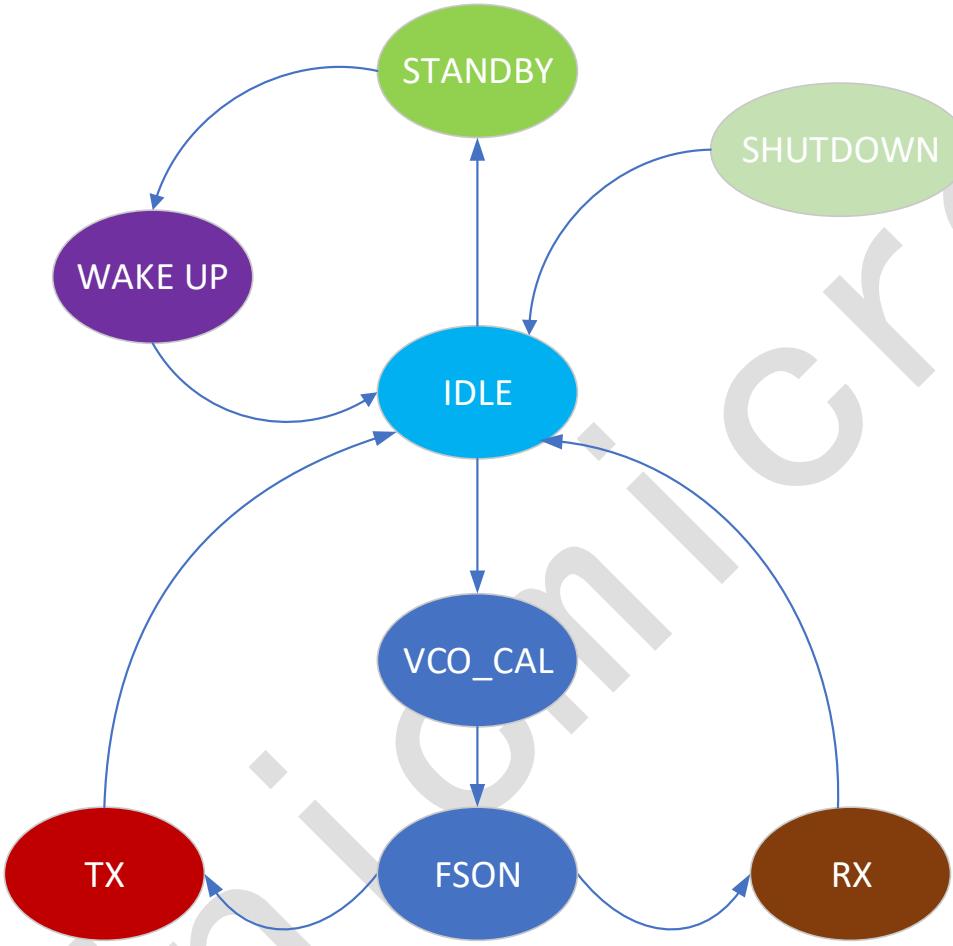


图 6-1：状态机控制图

6.2 工作模式

表 6-1：工作模式表

State/mode	Description	Command
SHUTDOWN	芯片处于关断状态	SDN 拉高
IDLE	空闲状态，复位后进入此状态	上电复位或发送 IDLE 命令
STANDBY	数字模块工作，其他模块关闭，寄存器保持	SPI 写 STANDBY 命令
FSON	频综打开状态，可快速进入接收或发送状态	SPI 写 FSON 命令
RX	接收数据状态	SPI 写 RX 命令
TX	发送数据状态	SPI 写 TX 命令

6.3 状态机说明

1. 关断状态（SHUTDOWN）

   当 SDN 管脚拉高之后，芯片进入关闭状态，芯片内部所有的电路模块都关闭，消耗电流小于 10nA。

2. IDLE 状态

   上电复位或发送 IDLE 命令后，芯片进入 IDLE 状态。在此状态下，晶体振荡器开启，等待 SPI 接口命令再执行其它的动作。

3. STANDBY 状态

   SPI 写命令进入 STANDBY 状态后，芯片的数字供电开启，所有的数字电路正常供电，其它模块都关闭，寄存器值可以保持。将 SPI_CS 拉低（2ms）后，芯片从 STANDBY 状态退出，进入 IDLE 状态。

4. 频综打开状态（FSON）

   频综打开并保持在这个状态，在此状态下，芯片收到 TX/RX 命令后会直接进入 TX/RX 状态。

5. 发送状态（TX）

   收到发射数据包命令后，芯片先打开 PLL 及 VCO，进行校准，等待至 PLL 达到要求的发射频段，如果自动信道检测功能打开，则在进入发射数据前先进行信道检测，如果空闲则进行发送数据包，如果信道忙，则下个动作可通过寄存器设置，是直接退出发送，还是继续检测 RSSI，直到把数据包发出。当数据包发出后，如果自动应答功能开启则切换到 RX_ACK 状态，以确定包有没有被接收方正确的接收，如果超出寄存器设定的时间没有收到应答包，则进行重发，重发最大次数可寄存器设置。

6. 接收状态（RX）

   收到 RX 命令后，芯片先打开 PLL，然后进行 VCO 校准并依次启动接收电路（LNA、Mixer、PGA 和 ADC）和数字解调器。当芯片收到数据包后，会给出 nIRQ 中断指示信号然后退回到 IDLE 状态或者 STANDBY 状态（定时唤醒模式下）。当 AUTO_ACK 功能开启时，芯片收到数据包后会自动发射 ACK 信号然后再退回到 IDLE 状态。

6.4 GPIO 和中断

UM2010 有 4 个 GPIO，分别是 nIRQ 和 GPIO[2:0]，每个 GPIO 都可以配置成不同的输入或者输出。芯片能产生两种中断信号，pkt_flag 和 FIFO_flag，该两个中断信号标志位都可读。pkt_flag 分为 4 个功能：前导匹配、同步字匹配、接收或发送包完成。FIFO_flag 表示 FIFO full 或 empty，在发送模式时表示 FIFO empty，在接收模式时表示 FIFO full。

6.4.1 GPIO 的配置

nIRQ 和 GPIO[2:0]为双向口，GPIO[2]默认为输入口作为 nRST 使用，拉低可复位整个数字部分。GPIO 配置相关寄存器如下表所示：

表 6-2：GPIO 配置相关寄存器列表

Address	Bit	R/W	Name	Description
0x4D	7	R/W	nirq_dir	nIRQ IO 口方向： 0：output 1：input
	6	R/W	gpio_dir[0]	gpio[0] 方向： 0：output 1：input
	5	R/W	gpio_dir[1]	gpio[1] 方向： 0：output 1：input
	4	R/W	gpio_dir[2]	gpio[2] 方向，默认复位输入功能： 0：output 1：input
	3:0	R/W	nirq_sel	nIRQ 引脚输出功能选择： 0x0：pkt_flag(包括数据包完成中断 pkt_int, 接收前导码中断 preamble_int, 接收同步字中断 syncword_int) 0x1：pkt_int 0x2：preamble_int 0x3：syncword_int 0x4：FIFO_flag 0x5：brclk（参考 Reg11[2:0]描述） 0x6：rxdata 0x7：txdata 0x8：tr_switch 0x9：tr_switch 反向 0xa：rxdata 0xb：txdata 0xc：wor_event 0xd：高电平 Others：低电平
0x4E	7:4	R/W	gpio_0_sel	gpio[0]输出功能选择，参考 Reg4D[3:0]描述
	3:0	R/W	gpio_1_sel	gpio[1]输出功能选择，参考 Reg4D[3:0]描述
0x4F	7:4	R/W	gpio_2_sel	gpio[2]输出功能选择，参考 Reg4D[3:0]描述

6.4.2 中断的配置和映射

GPIO 输出功能描述如下表所示：

表 6-3：GPIO 输出功能描述

Value	Name	Description
0x0	pkt_flag	pkt_flag (包括数据包完成中断 pkt_int, 接收前导码中断 preamble_int, 接收同步字中断 syncword_int)
0x1	pkt_int	包完成中断，包含接收完成和发射完成
0x2	preamble_int	接收 Preamble 有效中断
0x3	syncword_int	接收 Syncword 有效中断
0x4	FIFO_flag	RX FIFO 快满中断，TX FIFO 快空中断
0x5	brclk	测试时钟输出，由 Reg11[2:0]设置。 Reg11[2:0]： 000：rx_clk 接收同步时钟 001：xtal_clk 晶振时钟 010：xtal_clk/8 011：xtal_clk/16 100：频综时钟 101：tx_clk 发射同步时钟 110：内部 RC32K 时钟 111：ADC_clk
0x6	rxdata	解调器输出的串行数据
0x7	txdata	进入发射调制的串行数据
0x8	tr_switch	发射使能标志
0x9	tr_switch_inv	反向发射使能标志
0xa	rxdata	-
0xb	txdata	-
0xc	wor_event	自动唤醒状态标志
0xd	high_level	高电平
Others	low_level	低电平

6.5 FIFO 缓冲区

芯片内置 256 字节 FIFO，分为两组 128 字节的 FIFO 作为发射和接收独立使用。FIFO 独立使用时，发射 FIFO 的数据不会被更新掉，下次如要发同样的数据就不需要再重新写入数据到 FIFO 了。也可以通过寄存器 Reg0E[3]设置成 256 字节共用模式，发射和接收都有 256 字节的 FIFO 可用。

在重新写一帧数据到发射 FIFO 时，应对发射 FIFO 写指针进行清零。在重新读一帧接收 FIFO 数据时，应对接收 FIFO 读指针进行清零。接收写指针和发射读指针芯片内部会自动清零。

6.5.1 FIFO 相关的寄存器

FIFO 相关的寄存器如下表所示：

表 6-4：FIFO 相关寄存器列表

Address	Bit	R/W	Name	Description
0x50	7:0	R/W	rx_fifo_wr_ptr	RX FIFO 写指针，写 0x80 清除指针
0x51	7:0	R/W	rx_fifo_rd_ptr	RX FIFO 读指针，写 0x80 清除指针

Address	Bit	R/W	Name	Description
0x52	7:0	R/W	rx_fifo_data	RX FIFO
0x53	7:0	R/W	tx_fifo_wr_ptr	TX FIFO 写指针，写 0x80 清除指针
0x54	7	R/W	tx_fifo_rd_ptr	TX FIFO 读指针，写 0x80 清除指针
0x55	7:0	R/W	tx_fifo_data	TX FIFO
0x0E	3	R/W	fifo_share_en	FIFO 共享设置： 0：RX 和 TX 各 128 字节 FIFO 1：RX 和 TX 共用 256 字节 FIFO
0x9F	5:0	R/W	fifo_empty_thr	TX FIFO 空门限，FIFO 数据低于门限值时 fifo_flag 置‘1’
0xA0	5:0	R/W	fifo_full_thr	RX FIFO 满门限，RX FIFO 数据高于门限值时 fifo_flag 置‘1

6.5.2 FIFO 的工作模式

芯片默认提供两个独立的 128-byte 的 FIFO，分别给 RX 和 TX 使用，两者互不相干。用户也可以将 fifo_share_en 设为 1，那么两个 FIFO 就可合成一个 256-byte 的 FIFO，在 TX 和 RX 下共同使用。

6.5.3 FIFO 的中断时序

对于 TX，设置 fifo_empty_thr 为将空阈值，只要发送的数据低于此门限值，fifo_flag 产生中断。对于 RX，设置 fifo_full_thr 为将满阈值，只要接收的的剩余空间低于此门限值，fifo_flag 产生中断。

6.5.4 FIFO 的应用场景

1. 在 RX 下接收数据

初始化寄存器优化值、数据率以及数据包的配置。并写 Reg60 = 0x20 进入接收，等 MCU 收到 nIRQ 中断后：

A. 读 Reg61 寄存器。

B. 判断 Reg61[7:4] 是否为 0xD，如果正确表明收到一包数据。

C. 清 RX FIFO 读指针，Reg51 = 0x80。

D. 读 RX FIFO 数据 Read Reg(0x52)。

E. 读 RSSI（Reg0x64）：根据需要可读出该包的 RSSI 值

F. 写接收命令重新进入接收数据状态。

2. 预先填好数据，进入 TX 发射

初始化寄存器优化值、数据率以及数据包的配置；

A. 清 TX FIFO 写指针：Reg53[7:0] = 0x80

B. 把数据写入 TX FIFO：Reg55 = data

C. 启动发射：Reg60 = 0x40

D. 发射完成：发射完成的判断可查询 Reg61[4]，如果为 1 表示发射完成。也可以通过 nIRQ 这个引脚产生的上升沿中断，来判断发射完成。

E. 如果需要再发射新的数据，重新清 TX FIFO 写指针，写入 TX FIFO 数据，然后启动发射。

3. 进入 TX 后，边填数据边发射

该场景适用于每次发送超出 FIFO 长度的数据包，在启动发射后，需要监控 fifo_flag 信号，该信号可以配置到引脚输出，也可以通过读 Reg61 寄存器获取；在 TX FIFO 快空时要及时写入数据到 TX FIFO，保证发射数据流的连续性。

4. 进入 RX 后，边读 FIFO 数据边接收

该场景适用于每次接收超出 FIFO 长度的数据包，在启动接收后，需要监控 fifo_flag 信号，该信号可以配置到引脚输出，也可以通过读 Reg61 寄存器获取；在 RX FIFO 快满时要及时读出数据，保证接收到的数据不被覆盖。

7 数据处理机制

芯片提供灵活可配置的数据控制模式，主要分为如下两大类。

- Direct 直通模式，在 TX 模式下，发射的串行数据直接从 GPIO 输入到发射模块将数据发出，在 RX 模式下，接收的串行数据可直接从 GPIO 口输出。
- Packet 数据包模式，在数据包模式下所有数据都要经过 FIFO，且支持各种数据包模式的控制，数据包模式又可以分成四种帧格式。

7.1 直通模式（Direct）

数据直通（direct）指的外部 MCU 通过 GPIO 输入发射数据或获取接收数据，在 TX/RX 启动前使能 Direct Mode，且将包模式配置为 Mode0，让芯片工作在循环状态，此时 TX/RX 的启动或停止完全由 MCU 的命令控制。

7.1.1 Direct 相关的寄存器

表 7-1：Direct 相关寄存器列表

Address	Bit	R/W	Name	Description
0x0E	2	R/W	Direct_mode	直通模式使能： 0：disable 1：enable

7.2 数据包（Packet）

数据包（Packet）中的 Payload 数据都是从 FIFO 读写，芯片内部实现了两个 128 字节的 FIFO，可分别作为发射和接收单独使用，也可连接成为 256 字节的的 FIFO 作为发射和接收共用。在数据包模式下有四种控制方式可选。在四种模式下，Preamble 和 Syncword 的配置都起作用，Preamble 最大可配置 256 个字节的长度，Syncword 可配置 0~8 个字节的长度。

数据包模式还可以对数据进行 Manchester、FEC、Whiten 编解码以及 CRC 校验。CRC 可配置 8、16、24、32 位四种长度校验，且 CRC 多项式任意可配置。Syncword 和 Payload 可配置高位优先或低位优先。CRC 和 Syncword 数据的 Manchester 编码可单独开关。

7.2.1 Mode 相关的寄存器

表 7-2：Mode 相关寄存器列表

Address	Bit	R/W	Name	Description
0x0E	1:0	R/W	Packet_mode	数据包控制模式： 00：模式 0，循环发射 TX FIFO 中的数据，需要 MCU 写命令退出接收或发射状态 01：模式 1，由寄存器控制数据包长度 10：模式 2，TX FIFO 中的第 1 个或 2 个字节作为包长度 11：模式 3，含有 length, address, seqnum 等多种数据域的模式

7.2.2 MODE 0

MODE 0 为 FIFO 循环模式。在此模式下启动 TX/RX，如果没有收到 MCU 的停止命令，TX 端会一直发射 TX FIFO 中的数据，RX 端会将接收到的数据不停地写入到 RX FIFO 中，并且 RX 端会检测 Preamble 和 Syncword。对于 TX 端在启动发射时会按照帧格式发射，帧格式有 Preamble、Syncword 和 Data 三个数据域，Data 数据来自 TX FIFO，且长度不受芯片本身控制。

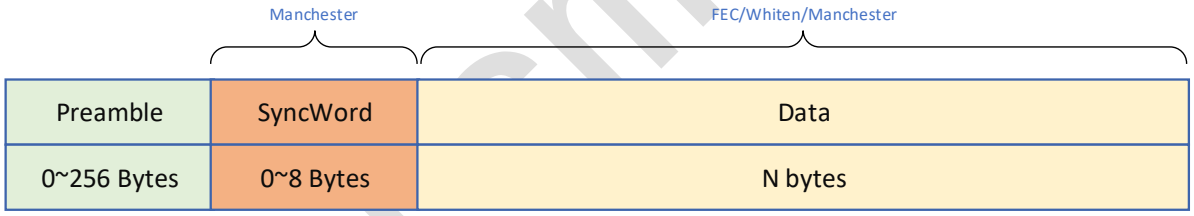


图 7-1：Mode 0 帧格式

7.2.3 MODE 1

MODE 1 帧格式包含 Preamble、Syncword、Data 和 CRC 域。发射 Data 数据来自 TX FIFO，数据长度由 Payload_len[15:0]寄存器控制。RX 端接收 Data 的长度同样由 Payload_len[15:0]寄存器控制。

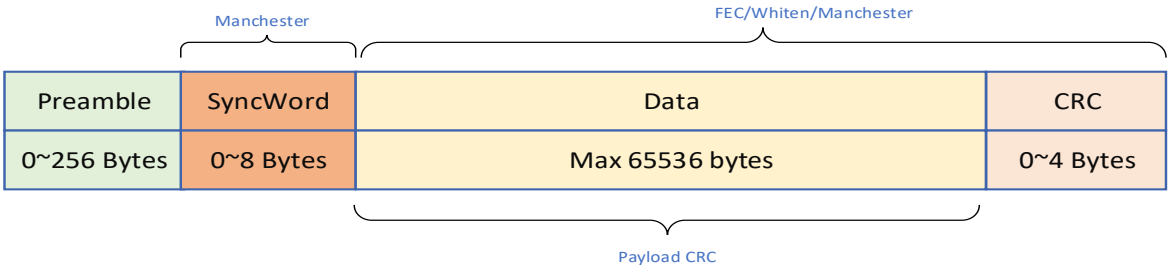


图 7-2：Mode 1 帧格式

7.2.4 MODE 2

MODE 2 帧格式包含 Preamble、Syncword、Length、Data 和 CRC 域。TX 时，Length 和 Data 都来自于 TX FIFO，Length 为 TX FIFO 的前 1 个字或前 2 个字节，Length 为两个字节时高低字节发射顺序可互换。在 RX 端，Length 为接收到的数据的前 1 个字或前 2 个字节，Length 为两个字节时高低字节顺序可互换，根据接收到的 Length 域数据去控制 Data 域的长度。

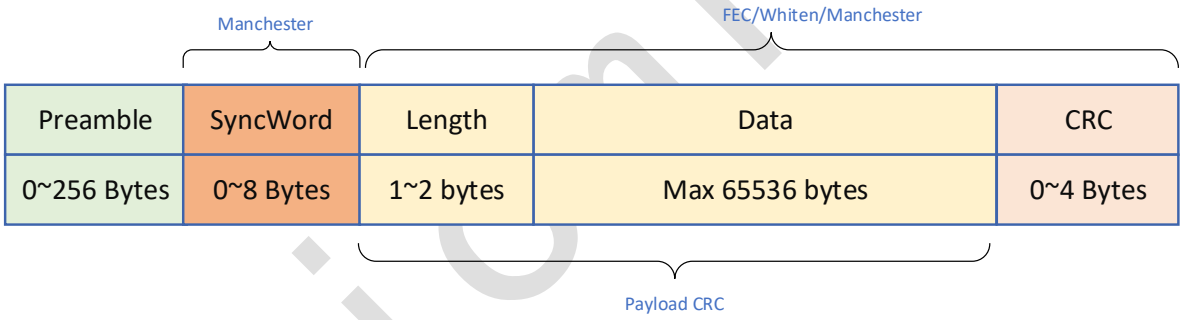


图 7-3：Mode 2 帧格式

7.2.5 MODE 3

MODE 3 帧格式包含 Preamble、Syncword、Length、Address、SeqNum、FCS2、Data 和 CRC 域。Length、Address、SeqNum、FCS2 作为 payload 域并且可以单独使能。Address 域可选择放置在 Length 之前。Length 数据来自 Payload_len[15:0]寄存器，且 Payload 长度范围包括 Length 到 CRC 之前的所有数据域。SeqNum 为数据包计数器，可由寄存器设置也可设置为自动增加。如果 FCS2[7]位使能 ACK 请求响应，当接收端接收到 ACK 请求时，会自动发出 ACK 数据包响应发射端。如果 Length 域没有使能，则接收端根据寄存器 Payload_len 来接收 Payload 数据。

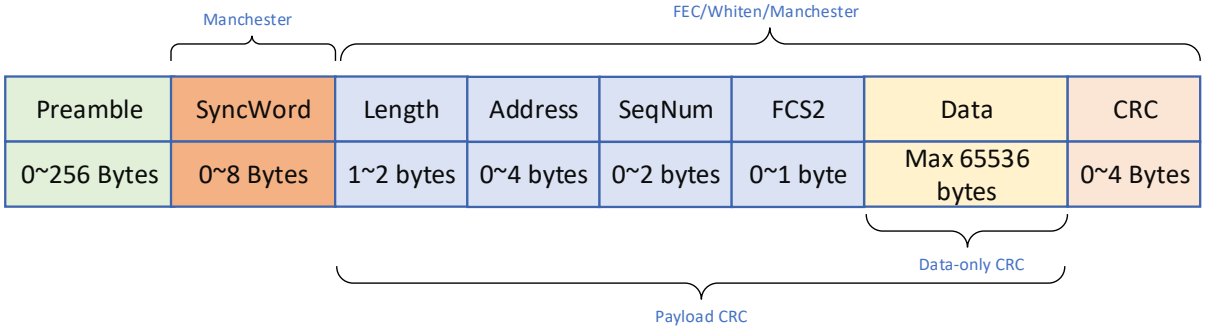


图 7-4：Mode 3 帧格式

7.3 Preamble 配置

对 TX 来说，Preamble 支持独立的使能，并且可控制发射的长度。对于 RX 来说，Preamble 检测成功会有 PREAM_OK 中断产生。用户可在有需要的时候去检测此中断。

7.3.1 Preamble 相关的寄存器

表 7-1：Preamble 配置相关寄存器列表

Address	Bit	R/W	Name	Description
0x0D	4	R/W	preamble_en	设置 preamble 前导码使能： 0：disable 1：enable 使能
0x0C	7:0	R/W	tx_preamble_len	发射 Preamble 长度，单位字节
0x10	2	R/W	preamble_int_en	接收 Preamble（前导码）匹配中断使能： 0：disable 1：enable
0x37	7:0	R/W	csrst_len	在补偿功能使能时，找到信号的频谱后连续收到 Reg37*4 个前导后，则认为找到有效的 preamble
0x38	4	R/W	demod_comp_en	解调频偏补偿： 0：disable 1：enable
0x73	7:0	R/W	preamble_val	发射前导码时的使用的数据

7.4 Sync Word 配置

Sync Word 支持独立使能，支持曼彻斯特编码，支持 Bit 顺序互换。在接收时支持同步字中断。

7.4.1 Sync Word 相关的寄存器

表 7-2：Sync 配置相关寄存器列表

Address	Bit	R/W	Name	Description
0x0D	5	R/W	syncword_en	Syncword 同步字使能： 0：disable 1：enable
0x10	1	R/W	syncword_int_en	接收 Syncword（同步字）匹配中断使能： 0：disable 1：enable
0x74	4	R/W	sync_man_en	同步字曼彻斯特编码使能： 0：disable 1：enable
	3	R/W	sync_bit_order	同步字 bit 顺序： 0：字节低位 LSB 1：字节高位 MSB
	2:0	R/W	sync_len	同步字长度： 000：{sync_id_1} 001：{sync_id_1, sync_id_2} 010：{sync_id_1, sync_id_2, sync_id_3} 011：{sync_id_1, sync_id_2, sync_id_3, sync_id_4} 100：{sync_id_1, sync_id_2, sync_id_3, sync_id_4, sync_id_5} 101：{sync_id_1, sync_id_2, sync_id_3, sync_id_4, sync_id_5, sync_id_6} 110：{sync_id_1, sync_id_2, sync_id_3, sync_id_4, sync_id_5, sync_id_6, sync_id_7} 111：{sync_id_1, sync_id_2, sync_id_3, sync_id_4, sync_id_5, sync_id_6, sync_id_7, sync_id_8}
0x75	7:0	R/W	sync_id_1	Sync Word 的值，根据不同的 sync_len 设置来填入不同的寄存器。
0x76	7:0	R/W	sync_id_2	
0x77	7:0	R/W	sync_id_3	
0x78	7:0	R/W	sync_id_4	
0x79	7:0	R/W	sync_id_5	
0x7A	7:0	R/W	sync_id_6	
0x7B	7:0	R/W	sync_id_7	
0x7C	7:0	R/W	sync_id_8	

7.5 Length 配置

在不同的包模式下，Length 域配置方式不一样。并且支持长度字节选择，高低位互换。

7.5.1 Length 相关的寄存器

表 7-3：Length 相关的寄存器列表

Address	Bit	R/W	Name	Description
0x0E	7	R/W	length_byte_swap	包长度控制域为两个字节时，高低字节顺序交换： 0：低字节在前 1：高字节在前
	6	R/W	Length_sel	选择 1 个或 2 个字节作为包长度控制域： 0：1 字节 1：2 字节
0x6A	7:0	R	m3_rx_payload_len[7:0]	数据包模式 3 时，接收 length 数据
0x6B	7:0	R	m3_rx_payload_len[15:8]	数据包模式 3 时，接收 length 数据
0x80	5	R/W	m3_length_en	数据包模式 3 时，数据长度域控制： 0：disable 1：enable
0x7D	7:0	R/W	payload_len[7:0]	payload 长度，在数据包模式 1 和数据包模式 3 有效
0x7E	7:0	R/W	payload_len[15:8]	payload 长度，在数据包模式 1 和数据包模式 3 有效
0x81	1	R/W	m3_addr_pos_sel	数据包模式 3 时，地址位置选择： 0：address 放在 length 之后 1：address 放在 length 之前

7.5.2 MODE 0 的 length 配置

在 MODE 0 中，对 TX，一直从循环 TX FIFO 中读出数据进行发射。对 RX，RX 端会将接收到的数据不停地写入到 RX FIFO 中。所以在 MODE 0 中 payload length 是无限制的。

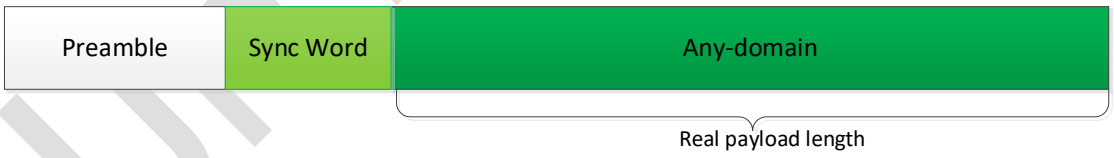


图 7-5：MODE 0

7.5.3 MODE 1 的 length 配置

在 MODE 1 中，payload_len 来自寄存器{Reg7D,Reg7E},Real payload length 的实际长度为配置的寄存器 Payload_len 加一，例如 Real payload length 为 8 bytes，则 Payload_len 配置为 7。

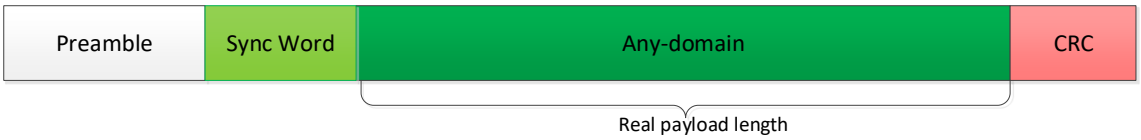


图 7-6：MODE 1

7.5.4 MODE 2 的 length 配置

在 MODE 2 中，payload length 由 FIFO 第一个或前两个字节配置，length 表示在 length 后面跟随的数据字节数量。

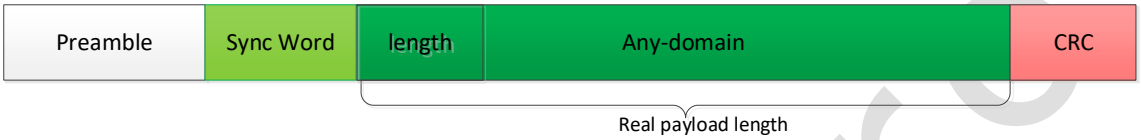


图 7-7：MODE 2

7.5.5 MODE 3 的 length 配置

在 MODE 3 中，根据 Length 使能或者不使能来分为两种格式：固定包格式和可变包格式，对于 TX 来说，length 都来自寄存器，对于 RX 来说，如果 length 域不使能，则来自寄存器，如果 length 域使能，则来自数据包格式的 length。

7.5.5.1 固定包格式

在 MODE 3 中，Length 域不使能(Reg80[5]=0，m3_length_en=0)的情况下，对于 TX 和 RX 来说，Real payload length 来自寄存器{Reg7D,Reg7E}, Real payload length 的实际长度为配置的寄存器 Payload_len 加一。

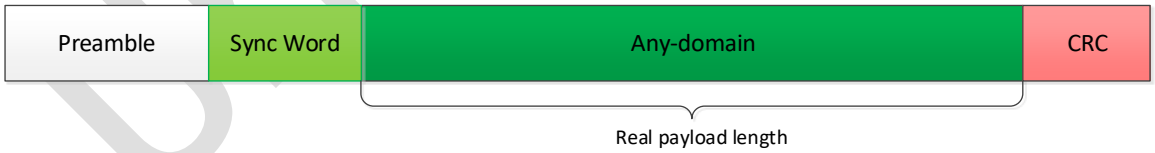


图 7-8： m3_length_en=0 时 MODE 3 帧格式

当 length 域不使能（m3_length_en=0）时，对于 TX 和 RX 来说，Payload 结构 = Address + Seqnum + FCS2 + Data，其中 Address / Seqnum / FCS2 单独配置使能，示例如下：

- **举例1：**

  Payload_len<15:0> = 17

  Addr_en = 1

  Addr_size <1:0> = 2

  Seqnum_en = 1

Seqnum_size = 1

Fcs2_en = 1

此处，Payload 包含 Address + Seqnum + FCS2 + Data 共 4 部分，Payload 的总长度是 17 + 1 = 18，Address 的长度是 2 (Addr) + 1 = 3，Sequence Num 的长度是 1 + 1 = 2，FCS2 长度固定为 1，所以，Data 的长度是 18 -3 -2 -1 = 12 个 byte。

- **举例2：**

Payload_len <15:0> = 17

Addr_en = 1

Addr_size<1:0> = 2

Seqnum_EN = 0

Seqnum_size = 1

Fcs2_en = 0

此处，Address + Seqnum + FCS2 不存在，那么 Data 的长度就是 18。

7.5.5.2 可变包格式

在 MODE 3 中，Length 域使能(Reg80[5]=1，m3_length_en=1)的情况下，对于 TX 来时，payload length 来自寄存器{Reg7D,Reg7E}，对于 RX 来说，payload length 来自数据包的 Length 域，根据 Length 域的位置不同及 Address 域使能来决定 payload length 的计算。

- **Address 不使能（m3_addr_en=0）**

当 Address 不使能的情况下，Payload 结构 = Length + Seqnum + FCS2 + Data，其中 Seqnum、FCS2 可以独立配置使能，payload Length 包括 length+Seqnum_FCS2+Data。

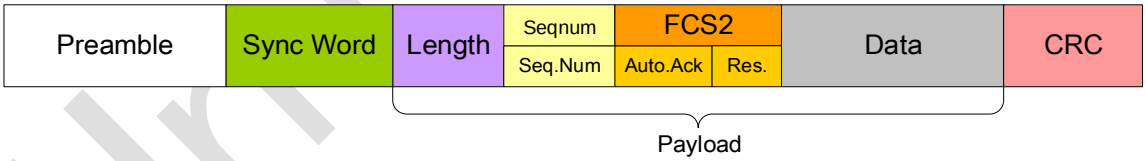


图 7-9：可变长度包格式，Address 不存在

当 Length 域使能，Address 不使能（m3_length_en=1，m3_addr_en=0），整个 payload 的长度，包括 length+Seqnum_FCS2+Data，对于 TX 来说，Length 的内容由 Payload_len <15:0>寄存器配置。如果 Length_size = 0，即 Payload_len <7:0>有效，最大值为 255，Length Byte 本身是 1 个 byte 的长度。如果 Length_size = 1，即 Payload_len<15:0>有效，最大值为 65535，Length Byte 本身是 2 个 byte 的长度，对于 RX 来说，RX 需要解析 length 域，length+1 表示 Payload 的长度。示例如下：

举例：

Payload_len <15:0> = 17

Length_size = 0

Seqnum_EN = 1

Seqnum_size = 1

Fcs2_en = 1

此处，Payload 包含 Length + Seqnum + FCS2 + Data 共 4 部分，Payload 的总长度是 17 + 1 = 18，Length 的长度是 1，Sequence Num 的长度是 1 + 1 = 2，FCS2 长度固定为 1，所以，Data 的长度是 18 - 1 - 2 - 1 = 14 个 byte。

如果 Seqnum + FCS2 不存在，那么 Data 的长度就是 18 - 1 = 17 个 byte。

如果 length_size = 1，即 Length 的长度是 2，所以，Data 的长度是 18 - 2 - 2 - 1 = 13 个 byte。

如果 Seqnum + FCS2 不存在，那么 Data 的长度就是 18 - 2 = 16 个 byte。

- **Address 存在，且 Addr_pos_sel = 1（Address 在 Length Byte 之前）**

ADDR 放在 LENGTH 之前，ADDR 单独罗列出来，Payload 结构 = Address + Length Byte + Seqnum + FCS2 + Data，Real payload length 的实际长度为配置的寄存器 Payload_len 加一。

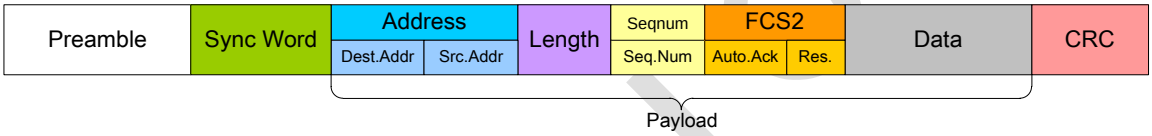


图 7-10： Address 存在，Address 在 Length Byte 之前

当 Address 存在，且 Addr_pos_sel = 1（Address 在 Length Byte 之前）时，对于 TX 来说，Payload 长度 = (addr_size<1:0> + 1) + 1 + payload_length<15:0>，其中 Address 的长度根据 addr_size<1:0>确定，payload_length<15:0>的内容就等于 Length Byte 的内容，payload_length+1 指的是除 Address 外的 Payload 的长度。如果 length_size = 0，即 payload_length<7:0>有效，最大值为 255，Length Byte 本身是 1 个 byte 的长度。如果 length_size = 1，即 payload_length<15:0>有效，最大值为 65535，Length Byte 本身是 2 个 byte 的长度。对于 RX 来说，RX 需要解析 length 域，length+1 表示除 Address 外的 Payload 的长度。示例如下：

举例：

Payload_length<15:0> = 17

Addr_en= 1

Addr_size<1:0> = 2

Length_size = 0

Seqnum_EN = 1

Seqnum_size = 1

FCS2_EN = 1

此处，Payload 包含 Address + (Length + FCS1 + FCS2 + Data)共 5 部分，其中，Address 的长度是 2 (Dest.Addr) + 1 = 3，Length 的长度是 1，Sequence Num 的长度是 1 + 1 = 2，FCS2 长

度固定为 1，那么 Payload 的总长度是 3 + (17 + 1) = 21，所以，Data 的长度是 18 - 1 - 2 - 1 = 14 个 byte。

如果 Seqnum + FCS2 不存在，那么 Data 的长度就是 18 - 1 = 17 个 byte。

如果 length_size = 1，即 Length 的长度是 2，所以，Data 的长度是 18 - 2 - 2 - 1 = 13 个 byte。

如果 Seqnum + FCS2 不存在，那么 Data 的长度就是 18 - 2 = 16 个 byte。

- **Address 存在，且 Addr_pos_sel = 0（Address 在 Length Byte 之后）**

ADDR 放在 LENGTH 之后，同 MODE 1，Payload 结构 = Length Byte + Address + Seqnum + FCS2 + Data，Real payload length 的实际长度为配置的寄存器 Payload_len 加一。

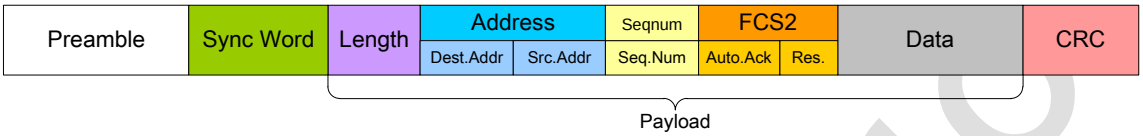


图 7-11：Address 存在，Address 在 Length Byte 之后

当 Address 存在，且 Addr_pos_sel = 0（Address 在 Length Byte 之后）时，对于 TX 来说，Payload 长度：1 + payload_length<15:0>，其中 payload_length<15:0>的内容就等于 Length Byte 的内容，指的是 Payload 的长度。如果 length_size = 0，即 payload_length<7:0>有效，最大值为 255，Length Byte 本身是 1 个 byte 的长度。如果 length_size = 1，即 payload_length<15:0>有效，最大值为 65535，Length Byte 本身是 2 个 byte 的长度。对于 RX 来说，RX 需要要解析 length 域，length+1 表示 Payload 的长度。示例如下：

举例：

Payload_length<15:0> = 17

Addr_en = 1

Addr_size<1:0> = 2

Length_size = 0

Seq_en = 1

Seqnum_size = 1

FCS2_EN = 1

此处，Payload 包含 Length + Address + Seqnum + FCS2 + Data 共 5 部分，Payload 的总长度是 17 + 1 = 18，Length 的长度是 1，Address 的长度是 2 (Addr) + 1 = 3，Sequence Num 的长度是 1 + 1 = 2，FCS2 长度固定为 1，所以，Data 的长度是 18 - 1 - 3 -2 - 1 = 11 个 byte。

如果 Seqnum + FCS2 不存在，那么 Data 的长度就是 18 - 1 -3 = 14 个 byte。

如果 length_size = 1，即 Length 的长度是 2，所以，Data 的长度是 18 - 2 -3 - 2 - 1 = 10 个 byte。

如果 Seqnum + FCS2 不存在，那么 Data 的长度就是 18 - 2 -3 = 13 个 byte。

7.6 Address 配置（仅 MODE 3）

7.6.1 Address 相关的寄存器

表 7-4：Address 配置相关寄存器列表

Address	Bit	R/W	Name	Description
0x81	5:4	R/W	m3_addr_size	数据包模式 3 时，地址长度： 00：1 字节 01：2 字节 10：3 字节 11：4 字节
	2	R/W	m3_addr_split_mode	数据包模式 3 时，将地址分为源地址和目的地址，在返回 ACK 时，将收到的源地址和目的地址位置互换： 0：不互换 1：互换
	0	R/W	m3_addr_en	数据包模式 3 时，地址域控制： 0：disable 1：enable
0x82	7:0	R/W	m3_addr_val[7:0]	数据包模式 3 时, 地址值
0x83	7:0	R/W	m3_addr_val[15:8]	
0x84	7:0	R/W	m3_addr_val[23:16]	
0x85	7:0	R/W	m3_addr_val[31:24]	
0x86	7:0	R/W	m3_addr_bit_mask[7:0]	数据包模式 3 时，地址过滤，mask 为‘1’时对应地址位不作比较
0x87	7:0	R/W	m3_addr_bit_mask[15:8]	
0x88	7:0	R/W	m3_addr_bit_mask[23:16]	
0x89	7:0	R/W	m3_addr_bit_mask[31:24]	

表 7-5：M3_ADDR_SIZE 选择

ADDR_SIZE	ADDR_VALUE			
	<31:24>	<23:16>	<15:8>	<7:0>
0	√			
1	√	√		
2	√	√	√	
3	√	√	√	√

7.7 seqnum 配置（仅 MODE 3）

在 MODE 3 中，在发射 Sequence Num 作为包结构内容一并发出，在接收中 Sequence Num

自动填入到映射的寄存器中，可以直接读取寄存器值。

7.7.1 seqnum 相关的寄存器

表 7-6：seqnum 相关的寄存器列表

Address	Bit	R/W	Name	Description
0x8A	5	R/W	m3_seqnum_auto_inc	数据包模式 3 时，seqnum 配置方式： 0：seqnum 数据由{Reg8c,Reg8b}配置 1：由内部 seqnum 计数器自动加 1
	4	R/W	m3_seqnum_en	数据包模式 3 时，seqnum 数据使能： 0：disable 1：enable
	3	R/W	m3_seqnum_bit_size	数据包模式 3 时，seqnum 数据长度： 0：8bit 1：16bit
0x8B	7:0	R/W	seqnum_Reg[7:0]	数据包模式 3 时，seqnum 值
0x8C	7:0	R/W	seqnum_Reg[15:8]	

7.8 FCS2 配置（MODE 3 的 ACK 模式）

7.8.1 FCS2 相关的寄存器

表 7-7：FCS2 相关的寄存器列表

Address	Bit	R/W	Name	Description
0x8D	0	R/W	m3_fcs2_en	数据包模式 3 时，FCS2 数据域使能： 0：disable 1：enable
0x8E	7	R/W	m3_fcs2_value	数据包模式 3 时，FCS2 数据 ACK request
0x80	3:0	R/W	m3_ack_node_ctrl	数据包模式 3 且 AUTO_ACK 功能开启，返回的的 ACK 包控制： [3]：为‘1’时使能返回接收到的数据长度 [2]：为‘1’时使能返回接收到的地址 [1]：为‘1’时使能返回接收到的 seqnum [0]：为‘1’时使能返回接收到的 FCS2 数据
0xA3	5:0	R/W	auto_ack_wait_time	auto ack 功能开启时，发射端发完数据包后等待 ACK 数据的时间，超时会重新发射数据。
0x74	7	R/W	auto_ack	auto_ack 使能： 0：disable 1：enable
	6	R/W	ack_payload_en	发射 ACK 时带 payload 数据返回：

Address	Bit	R/W	Name	Description
				0：disable 1：enable

7.8.2 AUTO ACK

接收端具有自动应答功能，在收到一个数据包后，如果该包要求回复 ACK 包（auto_ack 打开，在模式 3 时 m3_fcs2_value[7]需要为 1），则芯片直接自动切换到发射模式并发回 ACK 包。ACK 的包格式其中一种表示：

图 7-12：ACK 的包格式

ACK 数据包包含 Preamble 和 Sync Word，其他域需要 ack_m3_mode_ctrl（Reg80[3:0]）和 ack_payload_en(Reg74[6])来决定是否打开，其中，Address 域将接收到的 DEST ADDR 和 SRC ADDR 位置交换，如有 seq 域，将接收到的 Sequence Num 直接返回。

对于 Master 端，如果要求 Slave 发送应答包，模式 1 和 2 直接打开 auto_ack，而模式 3 需要将 m3_fcs2_value 设置为 1，发送完当前包后自动进入接收状态，接收 Slave 发回的 ACK 包。如果发射端超时未收到 ACK 包则重新发射数据包。

7.9 Payload Data 配置

Payload Data 支持 NRZ、曼彻斯特、数据白化编解码以及高低位互换。

7.9.1 Payload Data 相关的寄存器

Address	Bit	R/W	Name	Description
0x74	4	R/W	sync_man_en	同步字曼彻斯特编码使能： 0：disable 1：enable
0x0D	7	R/W	Payload_bit_order	Payload bit 顺序： 0：LSB 低位在前 1：MSB 高位在前
	3:2	R/W	pkt_enc_type	数据包编码： 00：NRZ 01：曼彻斯特编码 10：无效 11：交织编码

Address	Bit	R/W	Name	Description
0x94	2	R/W	crc_man_en	CRC 曼彻斯特编码使能： 0：disable 1：enable
0x0D	6	R/W	Mancester_inv	曼彻斯特编码： 0：上升沿为编码 1，下降沿为编码 0 1：下降沿为编码 1，上升沿为编码 0
0x0E	4	R/W	scramble_en	数据白化使能： 0：disable 1：enable
0x71	6	R/W	scramble_len	伪随机数长度选择： 0：PN9 1：PN7
	5	R/W	scramble_msb	选择首先输出的移位寄存器： 0：LSB 1：MSB
	4	R/W	scramble_type	伪随机数产生器类型： 0：抽出模式 1：插入模式
	3:1	R/W	scramble_poly	反馈抽头选择
	0	R/W	scramble_init_val[8]	scramble 初始化值最高位
0x70	7:0	R/W	scramble_init_val[7:0]	scramble 初始化值

7.9.2 Payload_bit_order 的示例

- Payload_bit_order = 0 表示发送时把 Payload 的每个 byte 自身从 LSB 到 MSB 的顺序发送或进行 Manchester/Whiten 编码。
- Payload_bit_order = 1 则表示把 Payload 的每个 byte 自身从 MSB 到 LSB 的顺序发送或进行 Manchester/Whiten 编码。

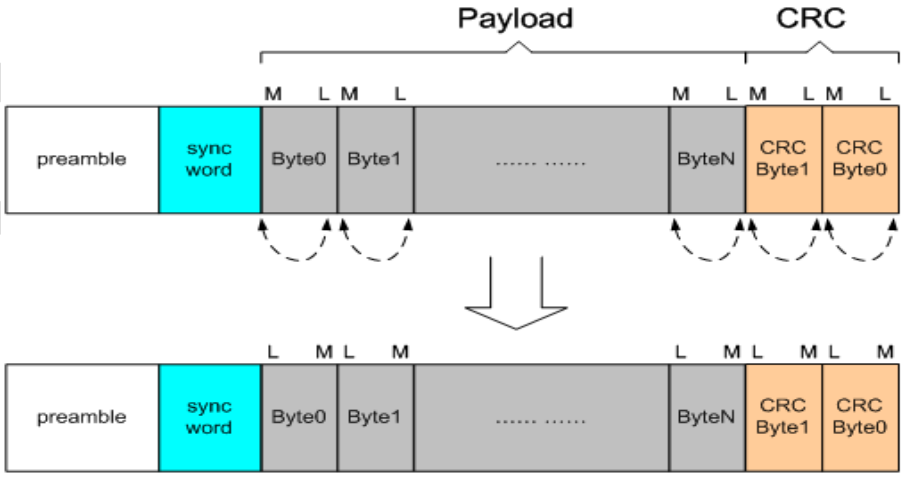
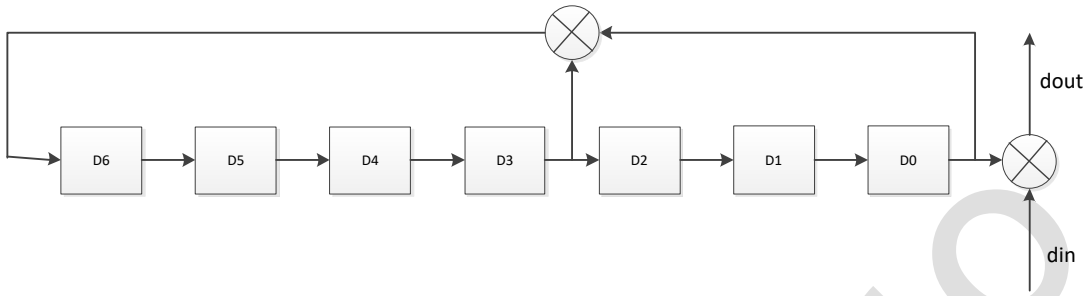


图 7-13：PAYLOAD_BIT_ORDER 操作

7.9.3 数据白化示例

数据白化随机数产生器有 PN7 或 PN9 可选择，结构上也有抽出或插入模式；数据输出可选最低位或最高位。

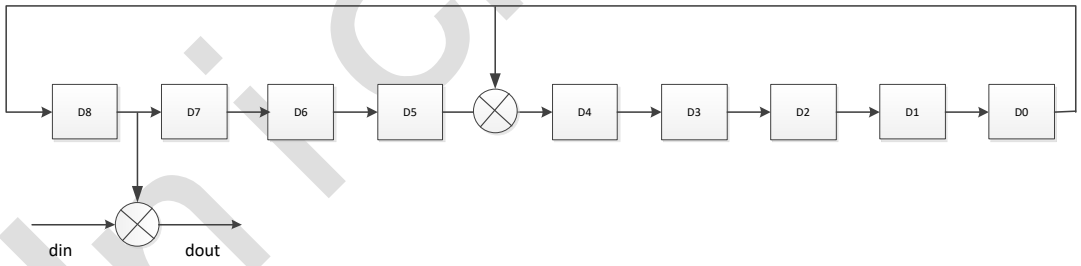
- PN7 的抽出模式



如上图，移位顺序不变，从高位移位到低位，对应配置为：

- Reg0E[4] = 1 打开白化使能。
- Reg71[6] = 1 选择 PN7。
- Reg71[5] = 0 选择 LSB 和数据做异或。
- Reg71[4]= 0 选择抽出模式。
- Reg71[3:1] = ‘b010 选择在 D2 输入处抽出，即 Reg71[3:1]+1 的寄存器输出处，例如举例为 2，则在 D3 输出处抽出或者插入。

- PN9 的插入模式



如上图，移位顺序不变，从高位移位到低位，对应配置为：

- Reg0E[4] = 1 打开白化使能。
- Reg71[6] = 0 选择 PN9。
- Reg71[5] = 1 选择 MSB 和数据做异或。
- Reg71[4]= 1 选择插入模式。
- Reg71[3:1] = ‘b100 选择在 D4 输入处插入，即 Reg71[3:1]+1 的寄存器输出处，例如举例为 4，则在 D5 输出处抽出或者插入。

7.10 CRC 配置

7.10.1 CRC 相关的寄存器

表 7-8：CRC 相关的寄存器列表

Address	Bit	R/W	Name	Description
0x0E	5	R/W	CRC_en	CRC 使能： 0：disable 1：enable
0x80	4	R/W	m3_crc_sel	数据包模式 3 时，CRC 的校验范围： 0：All payload 1：仅 FIFO 中的数据
0x94	4	R/W	crc_bit_order	CRC bit 顺序： 0：LSB 1：MSB
	3	R/W	crc_inv	CRC bit 取反： 0：disable 1：enable
	1:0	R/W	crc_len	CRC 长度： 00：8 位 01：16 位 10：24 位 11：32 位
0x95	7:0	R/W	crc_poly[7:0］	CRC 多项式配置
0x96	7:0	R/W	crc_poly[15:8］	
0x97	7:0	R/W	crc_poly[23:16］	
0x98	7:0	R/W	crc_poly[31:24］	
0x99	7:0	R/W	crc_init_val[7:0]	CRC 移位寄存器初始化值
0x9A	7:0	R/W	crc_init_val[15:8]	
0x9B	7:0	R/W	crc_init_val[23:16]	
0x9C	7:0	R/W	crc_init_val[31:24]	

7.10.2 CRC 配置说明

- **M3_crc_sel（Reg80[4]）**

  M3_crc_sel 用来指定在包模式 3 时 CRC 的编解码校验对象。

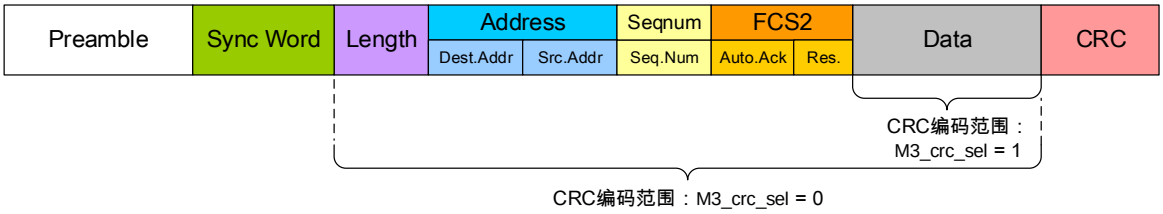


图 7-14：CRC 编码范围

- **crc_inv（Reg94[3]）**

  crc_inv 是将 CRC 的每一位都进行逻辑取反，原来是 0 变为 1，原来是 1 变为 0。

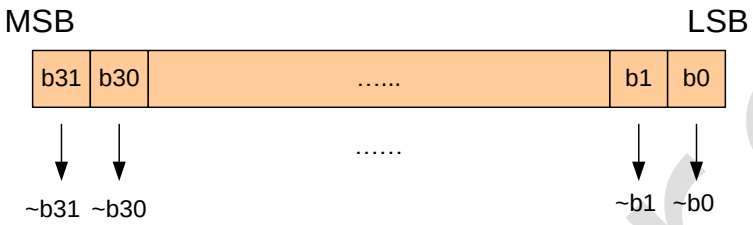


图 7-15：CRC_INV

- **crc_bit_order(Reg94[4])**

  crc_bit_order 是将 CRC 的 b31-b0 开始送出或者从 b0- b31 发送选项。

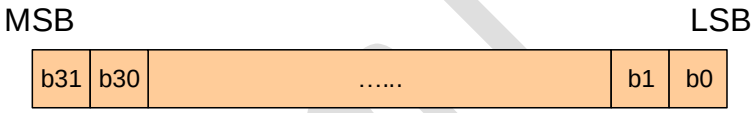


图 7-16：CRC_BIT_ORDER

7.10.3 常用 CRC 标准和多项式

表 7-9：常用 CRC 标准和多项式列表

名称	生成多项式	数值式	简记式	标准引用
CRC-4	x^4+x+1	0x1’3	0x3	ITU G.704
CRC-8	$x^8+x^5+x^4+1$	0x1’31	0x31	-
CRC-8	x^8+x^2+x1+1	0x1’07	0x07	-
CRC-8	$x^8+x^6+x^4+x^3+x^2+x1$	0x1’5E	0x5E	-
CRC-12	$x^{12}+x^{11}+x^3+x^2+x+1$	0x1’80F	0x80F	-
CRC-32c	$X^{32}+x^{28}+x^{27}+…x^8+x^6+1$	0x1’1EDC6F41	0x1EDC6F41	SCTP
CRC-16	$x^{16}+x^{15}+x^2+1$	0x1’8005	0x8005	IBM SDLC
CRC-CCITT	$x^{16}+x^{12}+x^5+1$	0x1’1021	0x1021	ISO HDLC，ITU X.25，V.34/V.41/V.42，PPP-FCS
CRC-32	$X^{32}+x^{26}+x^{23}+…x^2+x+1$	0x1’04C11DB7	0x04C11DB7	ZIP，RAR，IEEE 802 LAN /FDDI，IEEE1394，PPP-FCS

7.10.4 配置 CRC8 和 CRC16 示例

1. CRC8 的 $X^8+X^2+X^1+1$ 配置示例（简记式为 0x07）

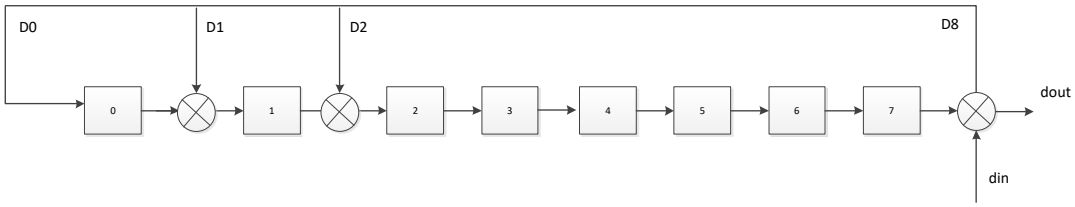


图 7-17：CRC8 的配置示例

如上图所示，对应配置为：

- Reg94[1:0] = 0x01；crc_len 设置 8bit
- Reg95 = 0x07
- Reg96 = 0x00
- Reg97 = 0x00
- Reg98 = 0x00
- Reg99，Reg9A，Reg9B，Reg9C 根据实际情况配置 CRC 的初始值。

2. CRC16 的 $X^{16}+X^{12}+X^5+1$ 的配置示例（简记式为 0x1021）

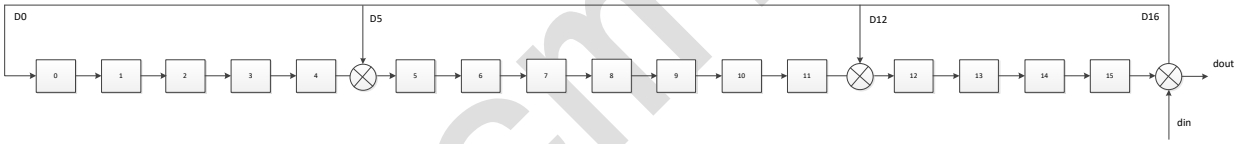


图 7-18：CRC16 的配置示例

如上图所示，对应配置为：

- Reg94[1:0] = 0x02; crc_len 设置 16bit
- Reg95 = 0x21
- Reg96 = 0x10
- Reg97 = 0x00
- Reg98 = 0x00
- Reg99，Reg9A，Reg9B，Reg9C 根据实际情况配置 CRC 的初始值

8 接口说明

8.1 SPI 接口

芯片内置 SPI 从机模块，通过标准 4 线 SPI 接口与主机 MCU 进行通讯。SPI_CS、SPI_CLK、SPI_MOSI 和 SPI_MISO。SPI 接口可在最大 16MHz 下工作运行。SPI 接口模式的时钟极性为正，在时钟下降沿采样数据，在时钟上升沿输出数据，地址和数据部分都是从 MSB 开始传送。芯片内部访问都是以 SPI 读写寄存器的方式，第一个字节为地址，后面跟一个字节的寄存器数据。如果在访问 FIFO 对应的地址时，可以在一个 SPI_CS 为低的周期内按字节方式连续的访问，SPI 接口控制器会自动增加访问地址，在访问 FIFO 数据时，地址和数据之前至少要等 3 个芯片的系统时钟，以便芯片确定 FIFO 指针地址。

在没有晶振时钟时，SPI 接口不能写数据，但仍然可以读寄存器数据。当访问寄存器的时候，SPI_CS 要拉低。然后首先发送一个 R/W 位，之后是 7 位的寄存器地址。RW=0 表示写，RW=1 表示读。

SPI 默认是 4 线的，在上电后可配置成 3 线。在 SPI 3 线模式下，SPI_MOSI 同时用于数据输入和输出，在读寄存器数据时，接口会在地址和数据之间对 SPI_MOSI 的方向进行切换。

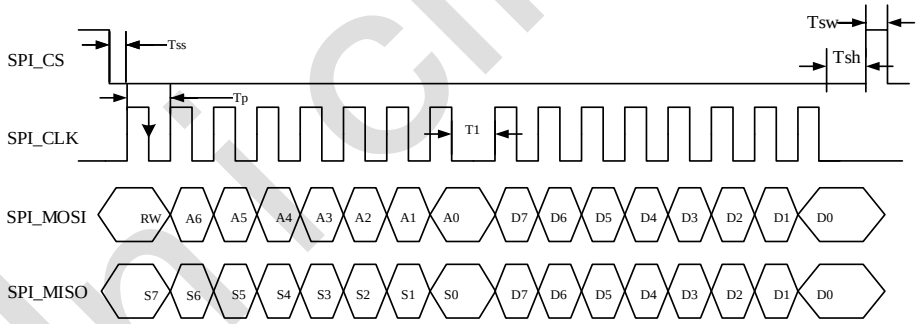


图 8-1：SPI 读写寄存器时序图

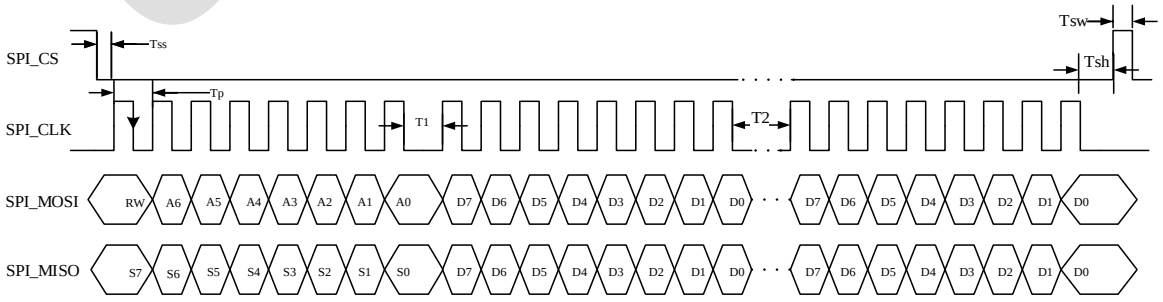


图 8-2：SPI 读写 FIFO 时序图

表 8-1：SPI 时序参数

Name	Min	Description
Tsw	100ns	两次 SPI 访问的间隔时间
Tss,Tsh	31.25ns	SPI_CS 和 SPI_CLK 的间隔时间
T1	32ns	地址和数据间隔时间
T2	32ns	两个寄存器数据读取时间间隔
Tp	62.5ns	SPI_CLK 时钟周期

9 应用参考

9.1 BUCK 模式

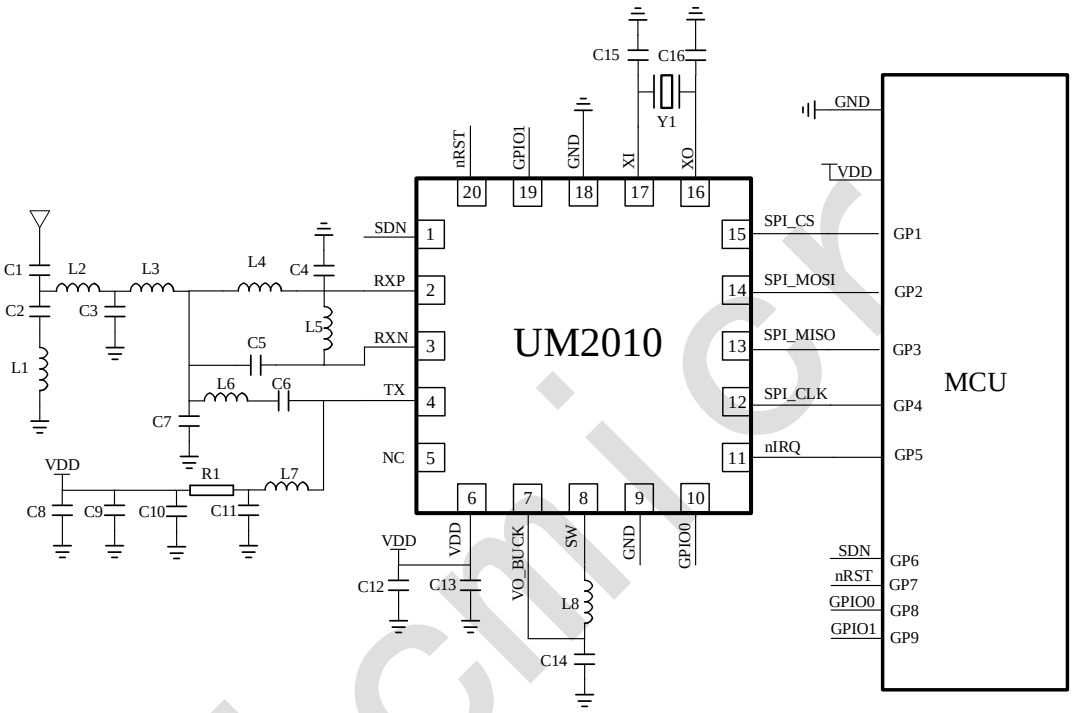


图 9-1：BUCK 模式应用参考电路图

9.2 非 BUCK 模式

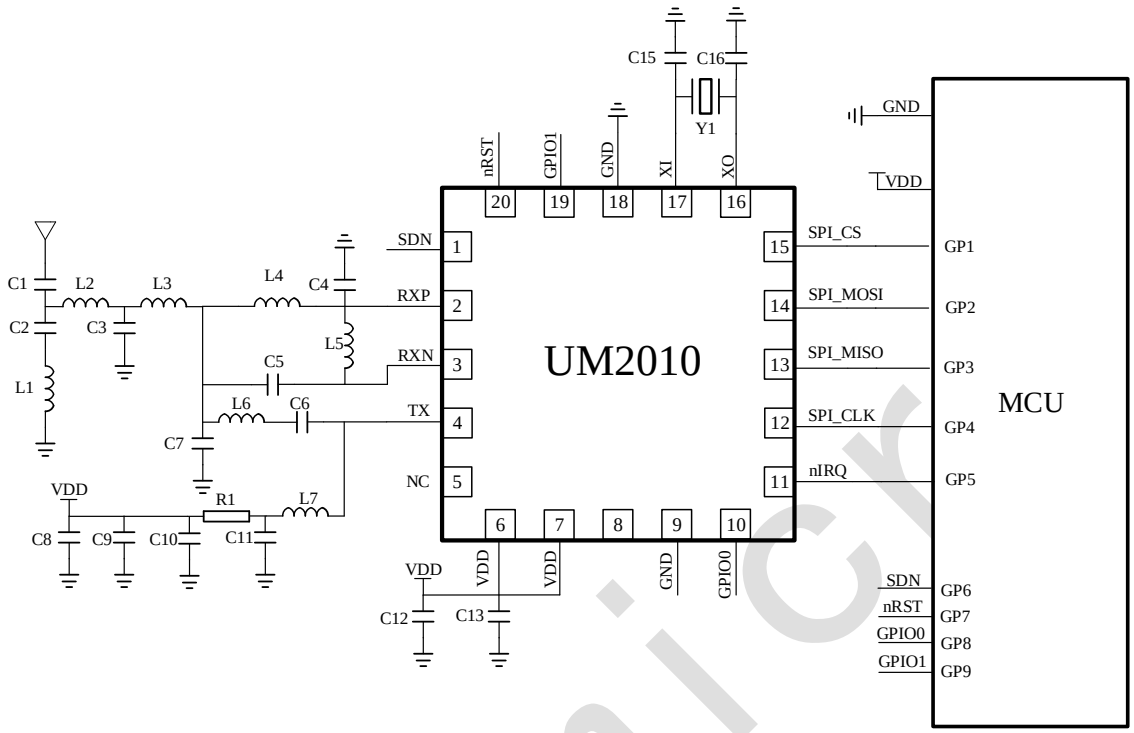


图 9-2：非 BUCK 模式应用参考电路图

9.3 天线匹配参考参数

表 9-1：天线匹配参考参数表

位号	描述	元件值				单位
		315	433.92	868	915	MHz
C1	±10%,0402, X7R,50V,YAGEO	22	20pF	5.6	5.6	nH
C2	±5%, 0402, C0G, 50 V,YAGEO	5.6	3	3.3	3.3	pF
C3	±5%, 0402, C0G, 50 V,YAGEO	8.2	8.2	6	6	pF
C4	±5%, 0402, C0G, 50 V,YAGEO	4.7	8.2	3.3	3.3	pF
C5	±5% ,0402,NP0, 50V,YAGEO	4.7	3	2.7	2.7	pF
C6	±1%,0402,NPO,50V,muRata	8.2	5.6	5	5	pF
C7	±5%,0402,NPO,50V,YAGEO	3	3.9	5.6	5.6	pF
C8	±10%,0402,X7R,50V,FH(风华)	470	470	470	470	pF
C9	±5%, 0603 NP0, 50 V,SAMSUNG	100	100	100	100	nF
C10	±10%,0402,X7R,50V,SAMSUNG	2.2	2.2	2.2	2.2	μF
C11	±10%,0402,X7R,50V,SAMSUNG	2.2	2.2	2.2	2.2	μF
C12	±10%,0402,X7R,50V,SAMSUNG	100	100	100	100	nF

位号	描述	元件值				单位
		315	433.92	868	915	MHz
C13	±5%, 0402, C0G, 50 V,SAMSUNG	470	470	470	470	pF
C14	±5%, 0402, C0G, 50 V,SAMSUNG	10	10	10	10	uF
C15	±10%,0402,X7R,50V,YAGEO	20	20	20	20	pF
C16	±10%,0402,X7R,50V,YAGEO	20	20	20	20	pF
L1	±2%,0402,贴片绕线电感,Sunlord	10	10	0R	0R	nH
L2	±2%,0402,贴片绕线电感,Sunlord	47	33	10	10	nH
L3	±2%,0402,贴片绕线电感,Sunlord	22	22	8.2	8.2	nH
L4	±5%,0402,贴片绕线电感,Sunlord	82	33	10	10	nH
L5	±5%,0402,贴片绕线电感,Sunlord	68	33	12	12	nH
L6	±5%,0603,贴片绕线电感,Sunlord	33	22	8.2	8.2	nH
L7	±5%,0603,贴片绕线电感,Sunlord	220	180	100	100	nH
L8	±5%,0805,贴片绕线电感,Sunlord	4.7	4.7	4.7	4.7	μH
R1	±5%,0402，贴片电阻,YAGEO	4.7	4.7	4.7	4.7	Ω
Y1	5032 贴片无源晶振 ±10PPM 20pF,YXC(扬兴晶振)	30	30	30	30	MHz

10 电气参数

10.1 绝对最大额定值

外部条件如果超过“绝对最大额定值”列表中给出的值，可能会导致器件永久性地损坏。这里只是给出能承受永久性损坏的最大载荷，并不意味着在此条件下器件的功能性操作无误。器件长期工作在最大值条件下会影响器件的可靠性。

表 10-1：芯片绝对最大额定值

符号	描述	最小值	典型值	最大值	单位	备注
V_{DD}	-	-0.3	-	+3.6	V	-
T_J	结温	-40	-	+125	℃	-
T_{stg}	存储温度	-50	-	+150	℃	-
I_{LATH}	Latch up 电流	-100	-	+100	mA	Norm: Jedec78
ESD	静电放电	-3	-	+3	KV	HBM

10.2 主要电气特性

10.2.1 通用工作条件

除非特别说明外，T_c=25℃，F=433.92MHz，GFSK，V_{DD}=3.3V

表 10-2：主要电气特性参数

符号	描述	参数以及条件	最小值	典型值	最大值	单位
V_{DD}	电源电压	-	1.8	3.3	3.6	V
T_c	工作温度	-	-40	-	85	℃
F_{RF}	工作频率范围	-	200	-	960	MHz
DR	数据率	-	0.1	-	300	kbps

10.2.2 功耗

表 10-3：功耗参数

符号	描述	参数以及条件	最小值	典型值	最大值	单位
$I_{STANDBY}$	休眠模式电流	-	-	40	-	μA
$I_{shutdown}$	关断电流	-	-	10	-	nA
I_{IDLE}	IDLE 状态工作电流 (BUCK 模式)	F_{RF}=315MHz	-	0.8	-	mA
		F_{RF}=433.92MHz	-	0.75	-	mA
		F_{RF}=868MHz	-	0.85	-	mA

符号	描述	参数以及条件	最小值	典型值	最大值	单位
		F_{RF}=915MHz	-	0.85	-	mA
	IDLE 状态工作电流 (非 BUCK 模式)	F_{RF}=315MHz	-	1.2	-	mA
		F_{RF}=433.92MHz	-	1.0	-	mA
		F_{RF}=868MHz	-	1.35	-	mA
		F_{RF}=915MHz	-	1.35	-	mA
I_{RX}	接收状态工作电流 (BUCK 模式)	F_{RF}=315MHz	-	6.5	-	mA
		F_{RF}=433.92MHz	-	6.5	-	mA
		F_{RF}=868MHz	-	7.0	-	mA
		F_{RF}=915MHz	-	7.0	-	mA
	接收状态工作电流 (非 BUCK 模式)	F_{RF}=315MHz	-	11.5	-	mA
		F_{RF}=433.92MHz	-	12	-	mA
		F_{RF}=868MHz	-	12.5	-	mA
		F_{RF}=915MHz	-	12.5	-	mA
I_{TX}	发射电流 @315MHz (非 BUCK 模式)	+18dBm	-	60	-	mA
		+13dBm	-	28	-	mA
		+10dBm	-	20	-	mA
		+0dBm	-	14	-	mA
	发射电流 @433.92MHz (非 BUCK 模式)	+18dBm	-	58	-	mA
		+13dBm	-	28	-	mA
		+10dBm	-	22	-	mA
		+0dBm	-	14	-	mA
	发射电流 @868MHz (非 BUCK 模式)	+18dBm	-	56	-	mA
		+13dBm	-	27	-	mA
		+10dBm	-	22	-	mA
		+0dBm	-	14	-	mA
	发射电流 @915MHz (非 BUCK 模式)	+18dBm	-	56	-	mA
		+13dBm	-	27	-	mA
		+10dBm	-	22	-	mA
		+0dBm	-	14	-	mA
I_{FS}	PLL tune 状态电流 (BUCK 模式)	F_{RF}=315MHz	-	5.1	-	mA
		F_{RF}=433.92MHz	-	4.9	-	mA
		F_{RF}=868MHz	-	5.6	-	mA
		F_{RF}=915MHz	-	5.6	-	mA
	PLL tune 状态电流 (非 BUCK 模式)	F_{RF}=315MHz	-	7.5	-	mA
		F_{RF}=433.92MHz	-	8.2	-	mA
		F_{RF}=868MHz	-	9.0	-	mA
		F_{RF}=915MHz	-	9.0	-	mA

10.2.3 接收特性

表 10-4：接收特性

符号	描述	参数以及条件	最小值	典型值	最大值	单位
SEN	接收灵敏度@315MHz (BER<0.1%) (BUCK 模式)	DR=0.1kbps F_{DEV}=0.3kHz	-	-129	-	dBm
		DR=1.2kbps F_{DEV}=2.5kHz	-	-119	-	dBm
		DR=10kbps F_{DEV}=22kHz	-	-110	-	dBm
		DR=100kbps F_{DEV}=50kHz	-	-99	-	dBm
		DR=300kbps F_{DEV}=300kHz	-	-94	-	dBm
	接收灵敏度@315MHz (BER<0.1%) (非 BUCK 模式)	DR=0.1kbps F_{DEV}=0.3kHz	-	-131	-	dBm
		DR=1.2kbps F_{DEV}=2.5kHz	-	-122	-	dBm
		DR=10kbps F_{DEV}=22kHz	-	-113	-	dBm
		DR=100kbps F_{DEV}=50kHz	-	-102	-	dBm
		DR=300kbps F_{DEV}=300kHz	-	-96	-	dBm
	接收灵敏度@433.92MHz (BER<0.1%) (BUCK 模式)	DR=0.1kbps F_{DEV}=0.3kHz	-	-127	-	dBm
		DR=1.2kbps F_{DEV}=2.5kHz	-	-119	-	dBm
		DR=10kbps F_{DEV}=22kHz	-	-109	-	dBm
		DR=100kbps F_{DEV}=50kHz	-	-100	-	dBm
		DR=300kbps F_{DEV}=300kHz	-	-93	-	dBm
	接收灵敏度@433.92MHz (BER<0.1%) (非 BUCK 模式)	DR=0.1kbps F_{DEV}=0.3kHz	-	-130	-	dBm
		DR=1.2kbps F_{DEV}=2.5kHz	-	-122	-	dBm
		DR=10kbps F_{DEV}=22kHz	-	-112	-	dBm
		DR=100kbps F_{DEV}=50kHz	-	-102	-	dBm
		DR=300kbps F_{DEV}=300kHz	-	-97	-	dBm
	接收灵敏度@868MHz (BER<0.1%) (BUCK 模式)	DR=1.2kbps F_{DEV}=2.5kHz	-	-117	-	dBm
		DR=10kbps F_{DEV}=22kHz	-	-108	-	dBm
		DR=100kbps F_{DEV}=50kHz	-	-98	-	dBm
		DR=300kbps F_{DEV}=300kHz	-	-92	-	dBm
	接收灵敏度	DR=1.2kbps F_{DEV}=2.5kHz	-	-120	-	dBm

符号	描述	参数以及条件	最小值	典型值	最大值	单位
	@868MHz (BER<0.1%) (非 BUCK 模式)	DR=10kbps F_{DEV}=22kHz	-	-112	-	dBm
		DR=100kbps F_{DEV}=50kHz	-	-101	-	dBm
		DR=300kbps F_{DEV}=300kHz	-	-96	-	dBm
	接收灵敏度 @915MHz (BER<0.1%) (BUCK 模式)	DR=1.2kbps F_{DEV}=2.5kHz	-	-117	-	dBm
		DR=10kbps F_{DEV}=22kHz	-	-108	-	dBm
		DR=100kbps F_{DEV}=50kHz	-	-97	-	dBm
		DR=300kbps F_{DEV}=300kHz	-	-93	-	dBm
	接收灵敏度 @915MHz (BER<0.1%) (非 BUCK 模式)	DR=1.2kbps F_{DEV}=2.5kHz	-	-120	-	dBm
		DR=10kbps F_{DEV}=22kHz	-	-112	-	dBm
		DR=100kbps F_{DEV}=50kHz	-	-100	-	dBm
		DR=300kbps F_{DEV}=300kHz	-	-96	-	dBm
Pin_max	最大输入信号功率	-	-	+10	-	dBm
Co_REJ	同频干扰	-	-	9	-	dB
Im_REJ	镜像抑制	-	-	-35	-	dB
1CH_REJ	第一邻道抑制	200KHz 信道间隔，带相同调制的干扰	-	-42	-	dB
2CH_REJ	第二邻道抑制	400KHz 信道间隔，带相同调制的干扰	-	-46	-	dB
3CH_REJ	第三邻道抑制	600KHz 信道间隔，带相同调制的干扰	-	-48	-	dB
Block	阻塞	10MHz 偏移, 连续波干扰	-	-72	-	dB

10.2.4 发射特性

表 10-5：发射特性

符号	描述	参数以及条件	最小值	典型值	最大值	单位
P_{out}	输出功率	-	-20	-	+18	dBm
P_{step}	输出功率调节	-	-	1	-	dB

10.2.5 频率综合器特性

表 10-6：频率综合器特性

符号	描述	参数以及条件	最小值	典型值	最大值	单位
F_{XTAL}	晶振参考频率	-	-	26	30	MHz
F	输出频率范围	-	200	-	960	MHz
F_{RES}	输出频率精度	F_{RF}=433.92MHz	-	12	-	Hz
T_{stable}	频率稳定时间	-	-	150	-	μs
PN	相位噪声	100KHz 频率偏移	-	-97	-	dBc/Hz
		500KHz 频率偏移	-	-115	-	dBc/Hz
		1MHz 频率偏移	-	-120	-	dBc/Hz

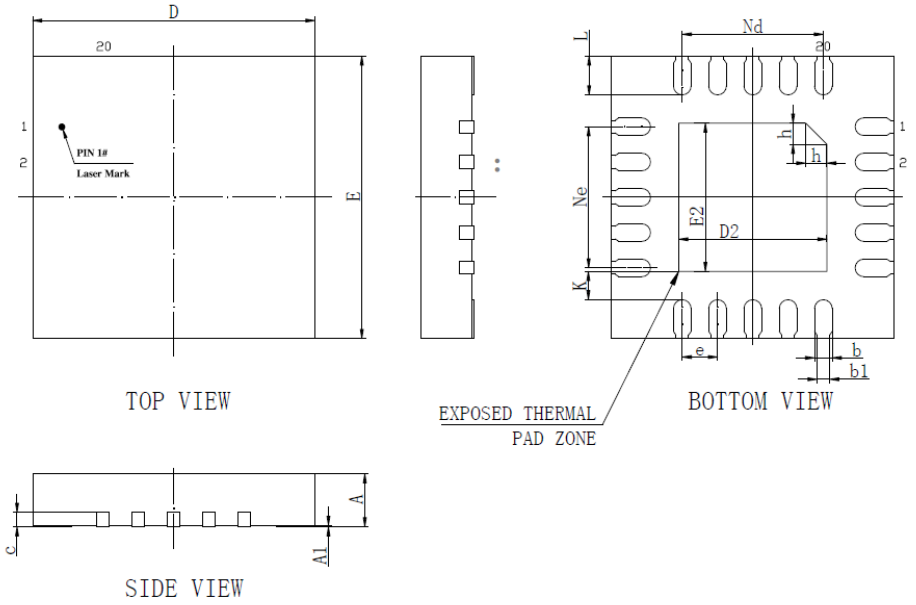
10.2.6 数字 IO 输入输出特性

表 10-7：数字 IO 输入输出特性

符号	描述	参数以及条件	最小值	典型值	最大值	单位
V_{IH}	高电平输入	-	0.8*V_{DD}	-	V_{DD}	V
V_{IL}	低电平输入	-	0	-	0.2*V_{DD}	V
I_{LEAK}	输入漏电流	-	-	-	100	nA
V_{OH}	高电平输出	1mA 负载电流	V_{DD}-0.4	-	-	V
V_{OL}	低电平输出	1mA 负载电流	-	-	V_{SS}+0.4	V

11 封装尺寸

11.1 QFN20（4*4mm）



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.20	0.25	0.30
b1	0.18REF		
c	0.203REF		
D	3.90	4.00	4.10
D2	2.00	2.10	2.20
e	0.50BSC		
Nd	2.00BSC		
Ne	2.00BSC		
E	3.90	4.00	4.10
E2	2.00	2.10	2.20
L	0.50	0.55	0.60
h	0.25	0.30	0.35
K	0.40REF		

图 11-1：QFN20 封装图