

UM2006A 用户手册

版本：V1.1



UNICMICRO
广芯微电子

广芯微电子（广州）股份有限公司

<http://www.unicmicro.com/>

条款协议

本档的所有部分，其著作权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本档的全部或部分组件。本档没有任何形式的担保、立场表达或其他暗示，若有任何因本档或其中提及的产品所有资讯所引起的直接或间接损失，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

目录

1	系统概述.....	1
1.1	典型应用.....	1
2	功能框图.....	2
3	封装及引脚定义.....	2
3.1	封装管脚分布.....	3
3.2	引脚功能描述.....	3
4	寄存器定义.....	5
4.1	Reg01 Address: 0x01 Default: 0x87.....	5
4.2	Reg02 Address: 0x02 Default: 0xCA.....	5
4.3	Reg03 Address: 0x03 Default: 0xF0.....	5
4.4	Reg04 Address: 0x04 Default: 0x00.....	6
4.5	Reg05 Address: 0x05 Default: 0x00.....	6
4.6	Reg06 Address: 0x06 Default: 0x60.....	6
4.7	Reg07 Address: 0x07 Default: 0x78.....	7
4.8	Reg08 Address: 0x08 Default: 0x88.....	7
4.9	Reg09 Address: 0x09 Default: 0x40.....	7
4.10	Reg0A Address: 0x0A Default: 0x00.....	8
4.11	Reg0B Address: 0x0B Default: 0x00.....	8
4.12	Reg0C Address: 0x0C Default: 0x09.....	8
4.13	Reg0D Address: 0x0D Default: 0x3D.....	8
4.14	Reg0E Address: 0x0E Default: 0x12.....	9
4.15	Reg0F Address: 0x0F Default: 0x70.....	9
4.16	Reg13 Address: 0x13 Default: 0x88.....	9
4.17	Reg16 Address: 0x16 Default: 0x32.....	9
4.18	Reg17 Address: 0x17 Default: 0x17.....	10
4.19	Reg18 Address: 0x18 Default: 0xD4.....	10
4.20	Reg19 Address: 0x19 Default: 0x2D.....	10
4.21	Reg1A Address: 0x1A Default: 0xD4.....	10
4.22	Reg1B Address: 0x1B Default: 0x2D.....	10
4.23	Reg1D Address: 0x1D Default: 0x80.....	11
4.24	Reg1E Address: 0x1E Default: 0x77.....	11
4.25	Reg1F Address: 0x1F Default: 0xC8.....	11
4.26	Reg20 Address: 0x20 Default: 0x80.....	11
4.27	Reg21 Address: 0x21 Default: 0xB2.....	11
4.28	Reg22 Address: 0x22 Default: 0x20.....	12

4.29	Reg23 Address: 0x23 Default: 0x1E.....	12
4.30	Reg24 Address: 0x24 Default: 0x01.....	12
4.31	Reg25 Address: 0x25 Default: 0x27.....	13
4.32	Reg26 Address: 0x26 Default: 0x20.....	13
4.33	Reg27 Address: 0x27 Default: 0x03.....	13
4.34	Reg28 Address: 0x28 Default: 0x70.....	14
4.35	Reg29 Address: 0x29 Default: 0x4D.....	14
4.36	Reg2B Address: 0x2B Default: 0xB5.....	15
4.37	Reg2D Address: 0x2D Default: 0x35.....	15
4.38	Reg2E Address: 0x2E Default: 0x05.....	16
4.39	Reg30 Address: 0x30 Default: 0x00.....	16
4.40	Reg40 Address: 0x40 Default: 0x00.....	16
4.41	Reg41 Address: 0x41 Default: 0x00.....	16
4.42	Reg42 Address: 0x42 Default: 0x00.....	17
4.43	Reg43 Address: 0x43 Default: 0x00.....	17
4.44	Reg44 Address: 0x44 Default: 0x00.....	17
4.45	Reg45 Address: 0x45 Default: 0x00.....	17
4.46	Reg46 Address: 0x46 Default: 0x00.....	17
4.47	Reg47 Address: 0x47 Default: 0x00.....	18
4.48	Reg48 Address: 0x48 Default: 0x00.....	18
4.49	Reg49 Address: 0x49 Default: 0x00.....	18
4.50	Reg50 Address: 0x50 Default: 0x00.....	18
4.51	Reg51 Address: 0x51 Default: 0x04.....	19
4.52	Reg52 Address: 0x52 Default: 0x00.....	19
4.53	Reg53 Address: 0x53 Default: 0xE5.....	19
4.54	Reg55 Address: 0x55 Default: 0xC0.....	19
4.55	Reg56 Address: 0x56 Default: 0x00.....	20
4.56	Reg57 Address: 0x57 Default: 0x7F.....	20
4.57	Reg58 Address: 0x58 Default: 0x00.....	20
4.58	Reg59 Address: 0x59 Default: 0xFF.....	21
4.59	Reg5a Address:0x5a Default:0x00.....	21
4.60	Reg60 Address:0x60 Default:0x00.....	21
5	功能描述.....	22
5.1	接收机.....	22
5.2	频率综合器.....	22
5.3	AGC.....	23
5.4	RSSI.....	24
5.5	AFC.....	25

5.6	PJD	25
5.7	CDR	26
5.8	RC32K	26
5.9	WOR.....	27
6	芯片运行.....	29
6.1	SPI接口	29
6.2	状态机控制	30
6.3	GPIO和中断.....	32
6.4	SLP超低功耗接收	33
7	数据处理机制.....	36
7.1	直通模式.....	36
7.2	Packet数据包模式.....	37
7.2.1	FIFO模式.....	38
7.2.2	UART模式.....	39
8	典型应用.....	41
8.1	参考电路图	41
8.2	器件参数.....	42
9	封装尺寸.....	43
9.1	QFN16 (3*3mm)	43
10	版本维护	44

表目录

表 3-1: 引脚功能说明	3
表 5-1: 频率综合器相关寄存器	23
表 5-2: AGC相关寄存器	23
表 5-3: RSSI相关寄存器	25
表 5-4: PJD相关寄存器	25
表 5-5: CDR相关寄存器	26
表 5-6: RC32K相关寄存器	27
表 5-7: WOR相关寄存器	28
表 6-1: SPI相关寄存器列表	30
表 6-2: 状态稳定时间	31
表 6-3: GPIO和中断相关寄存器列表	32
表 6-4: 收到唤醒信号后的唤醒时间和检测源控制	34
表 6-5: 低功耗接收相关寄存器列表	34
表 7-1: 数据处理相关寄存器	36
表 7-2: 直通模式相关寄存器	37
表 7-3: Packet模式相关寄存器	38
表 7-4: FIFO模式相关寄存器	39
表 7-5: UART模式相关寄存器	40

图目录

图 2-1: 功能框图	2
图 3-1: QFN16封装管脚分布图	3
图 5-1: RSSI与输入信号关系图	24
图 5-2: WOR流程图	27
图 6-1: SPI接口图.....	29
图 6-2: 基本模式状态图	31
图 6-3: 超低功耗接收流程图	33
图 7-1: 直通模式工作流程图	36
图 7-2: Packet 模式结构图	37
图 7-3: Packet模式工作流程图	38
图 8-1: 参考电路图	41
图 9-1: QFN16封装尺寸图.....	43

1 系统概述

UM2006A 是一款工作于 240MHz~1000MHz 范围内的低功耗、高性能、单片集成的 (G)FSK/OOK 无线接收芯片。适用于 ISM 频段 315/433/868/915 MHz 及其临近频点的无线接收应用。该芯片支持 0.5~40 kbps 数据率范围的 OOK 解调以及 0.5~250 kbps 数据率范围的 (G)FSK 解调。UM2006A 能够工作在 1.9 - 3.6 V 供电电压区间。当该芯片工作在 433.92 MHz 时，仅需 3.6 mA 电流便可实现 -109 dBm 的接收灵敏度。

芯片支持数据直通和数据包模式。数据包模式支持自适应数据率和固定配置数据率，Payload Data 支持 FIFO 读取和 UART 输出。

芯片通过空中唤醒功能 (WOR) 可以实现超低功耗接收 (SLP)，WOR 提供丰富的唤醒条件设置和开窗扩展功能，满足各种低功耗应用。

1.1 典型应用

- 工业传感及工业控制
- 安防系统
- 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，无线监控等智能传感器终端应用

2 功能框图

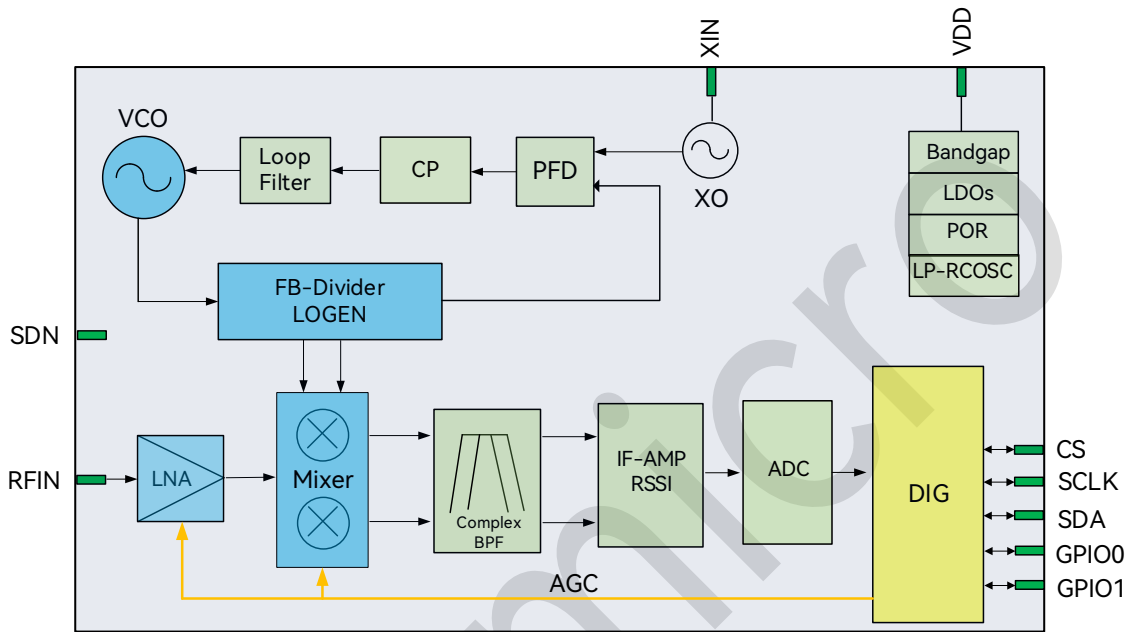


图 2-1: 功能框图

3 封装及引脚定义

3.1 封装管脚分布

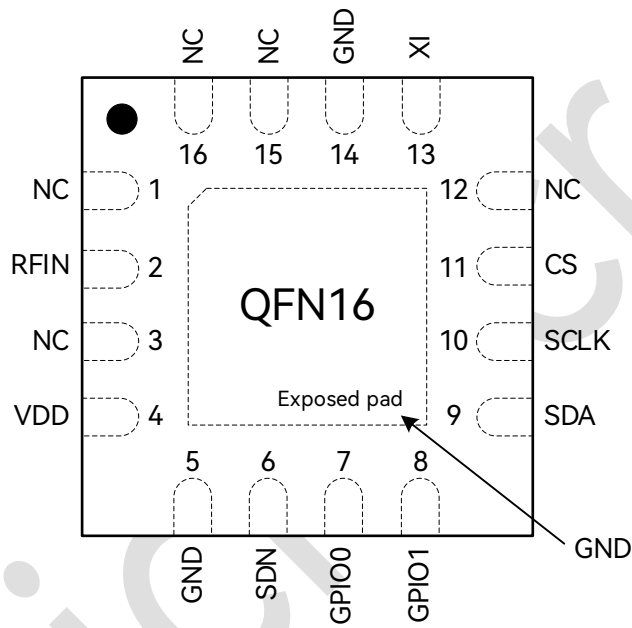


图 3-1: QFN16 封装管脚分布图

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
1	NC	NC	空脚，未连接内部任何电路
2	RFIN	RFI	射频输入
3	NC	NC	空脚，未连接内部任何电路
4	VDD	P	电源输入 (1.9~3.6V)
5	GND	G	芯片地
6	SDN	DI	关断引脚，高电平芯片处于关断状态
7	GPIO0	DIO	数字多功能引脚
8	GPIO1	DIO	数字多功能引脚

引脚编号	管脚名称	IO Type	功能描述
9	SDA	DIO	SPI 数据收发
10	SCLK	DI	SPI 时钟输入
11	CS	DI	SPI 片选输入
12	NC	NC	空脚，未连接内部任何电路
13	XI	AI	晶振输入，或外部参考时钟输入
14	GND	G	芯片地
15	NC	NC	空脚，未连接内部任何电路
16	NC	NC	空脚，未连接内部任何电路

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 Reg01 Address: 0x01 Default: 0x87

Bit	Name	Type	Description	Default
7:6	wkup_dly	W/R	从 SLEEP 唤醒后等待晶振稳定时间: 00: 75 μ s*1 01: 75 μ s*2 10: 75 μ s*3 11: 75 μ s*4	2'b10
5	sdm_en	W/R	SDM 模块使能: 0: 不使能 1: 使能	1'b0
4:0	reserved	-	-	5'h07

4.2 Reg02 Address: 0x02 Default: 0xCA

Bit	Name	Type	Description	Default
7:0	rc32k_cal_std	W/R	RC32K 时钟校准目标计数器值	8h'CA

4.3 Reg03 Address: 0x03 Default: 0xF0

Bit	Name	Type	Description	Default
7:6	duc_sel	W/R	Duty cycle 模式 (具体时间由寄存器配置): 00: off 01: 2s/1ms 10: 4s/1ms 11: 8s/1ms	2'b11
5:4	reserved	-	-	2'b11

Bit	Name	Type	Description	Default
3	fb_sel	W/R	频段选择: 0: 低频段 1: 高频段	1'b0
2	reserved	-	-	1'b0
1	fskdem_en	W/R	调制方式选择: 0: OOK 1: FSK	1'b0
0	uart_en	W/R	UART 发射模块使能, 在包模式接收时, 收到的 payload 数据通过 uart 模块发出, 默认 115200Hz 0: disable 1: enable	1'b0

4.4 Reg04 Address: 0x04 Default: 0x00

Bit	Name	Type	Description	Default
7:0	freq_frac[7:0]	W/R	频率分频比小数部分	8'h00

4.5 Reg05 Address: 0x05 Default: 0x00

Bit	Name	Type	Description	Default
7:0	freq_frac[15:8]	W/R	频率分频比小数部分	8'h00

4.6 Reg06 Address: 0x06 Default: 0x60

Bit	Name	Type	Description	Default
7:2	freq_ndiv[5:0]	W/R	频率分频比整数部分 $\text{freq_ndiv} = \text{round}(\text{freq}/\text{ref_freq})$, freq 为信道频率, ref_freq 为参考频率, round 为四舍五入	6h'18
1:0	freq_frac[17:16]	W/R	频率分频比小数部分 freq_frac 为分频比的小数部分, 最高位为符号位, 有效小数位为 17bit $\text{freq_frac} = \text{freq}/\text{ref_freq} - \text{round}(\text{freq}/\text{ref_freq})$, round 为四舍五入	2'b00

4.7 Reg07 Address: 0x07 Default: 0x78

Bit	Name	Type	Description	Default
7	rc32k_en	W/R	rc32k 功能使能,当 duty cycle 模式使能时 rc32k 功能也会使能	1'b0
6	rc32k_ftrim_sel	W/R	1: 选择寄存器 rc32k_frim 0: 选择 32K 校准值	1'b1
5	auto_cal_bpf_en	W/R	1: 选择 BPF 自动校准值 0: 选择寄存器 bpfcal_tune 值	1'b1
4	auto_cal_vco_en	W/R	1: 选择 VCO 自动校准值 0: 选择寄存器 VCO_FB	1'b1
3:0	vco_fb	W/R	auto_cal_vco为0时,vco_fb直接配置VCO band	4'h08

4.8 Reg08 Address: 0x08 Default: 0x88

Bit	Name	Type	Description	Default
7:0	rc32k_ftrim	W/R	寄存器配置 RC32K Trimming 值	8h'88

4.9 Reg09 Address: 0x09 Default: 0x40

Bit	Name	Type	Description	Default
7:6	reserved	-	-	2'b01
5	spi4w_en	W/R	四线 spi 选择, 选择一个 GPIO 作为 SPI 输出。 0: 3 wire 1: 4 wire	1'b0
4:3	maskdem_sel	W/R	GPIO 输出 demod_bitdata 信号时, maskdem_sel 选择 PJD_valide 有效或 sync_valid 有效或 rssi_valid 后再输出 demod_bit 00: 不屏蔽 01: rssi_valid 10: PJD_valid 11: sync_valid	2'b00
2:0	clkout_sel	W/R	调试, gpio 选择 clkout 信号, 该寄存器选择 一种 clk 输出 000: rxbit_clk	3'b000

Bit	Name	Type	Description	Default
			001: demod_out_ce 010: clk_adc 011: clk32k 100: fsm_clk 101: clk_reg 110: clk_fcal 111: synth_clk	

4.10 Reg0A Address: 0x0A Default: 0x00

Bit	Name	Type	Description	Default
7:0	wortimer_set[7:0]	W/R	WOR 功能, SLEEP 计数器总周期数, $(N+1) * t_{worclk}$	8'h00

4.11 Reg0B Address: 0x0B Default: 0x00

Bit	Name	Type	Description	Default
7:0	wortimer_set[15:8]	W/R	WOR 功能, SLEEP 计数器总周期数, $(N+1) * t_{worclk}$	8'h00

4.12 Reg0C Address: 0x0C Default: 0x09

Bit	Name	Type	Description	Default
7:0	wor_rtxtimer_set[7:0]	W/R	WOR 功能计算器唤醒后, 开窗 T1 计数长度, $(N+1) * t_{worclk}$	8'h09

4.13 Reg0D Address: 0x0D Default: 0x3D

Bit	Name	Type	Description	Default
7:0	wor_rtxtimer_set[15:8]	W/R	WOR 功能计算器唤醒后, 开窗 T1 计数长度, $(N+1) * t_{worclk}$	8'h3D

4.14 Reg0E Address: 0x0E Default: 0x12

Bit	Name	Type	Description	Default
7	rx_data_inv	W/R	接收数据反向	1'b0
6:4	reserved	-	-	3'b001
3:0	worclk_sel	W/R	WOR 功能计算器时钟分频选择 0000: 32k 0001: 32k/2 0010: 32k/4 --- 1111: 32k/32768	4'h02

4.15 Reg0F Address: 0x0F Default: 0x70

Bit	Name	Type	Description	Default
7:5	reserved	-	-	3'b011
4:0	bpfcals_tune	W/R	BPF RC 常数寄存器配置	5'h10

4.16 Reg13 Address: 0x13 Default: 0x88

Bit	Name	Type	Description	Default
7:5	pjd_len	W/R	接收有效信号翻转次数检测, 检测有效产生 pjd_valid, 对寄存器 clr_sync 写'1'或重新进入接收清 pjd_valid。次数为 pjd_len*4	3'b100
4:0	reserved	-	-	5'h08

4.17 Reg16 Address: 0x16 Default: 0x32

Bit	Name	Type	Description	Default
7:0	dcmin_th	W/R	DC 计算时的 min 门限	8'h32

4.18 Reg17 Address: 0x17 Default: 0x17

Bit	Name	Type	Description	Default
7	sync_en	W/R	数据包解码使能，匹配 sync_data 后按字节接收数据 0: disable 1: enable	1'b0
6	sync_len	W/R	0: 16-bit sync_data 1: 32-bit sync_data	1'b0
5:0	reserved	-	-	6'h17

4.19 Reg18 Address: 0x18 Default: 0xD4

Bit	Name	Type	Description	Default
7:0	sync_data[7:0]	W/R	sync 数据 MSB	8'hD4

4.20 Reg19 Address: 0x19 Default: 0x2D

Bit	Name	Type	Description	Default
7:0	sync_data[15:8]	W/R	sync 数据 MSB	8'h2D

4.21 Reg1A Address: 0x1A Default: 0xD4

Bit	Name	Type	Description	Default
7:0	sync_data[23:16]	W/R	sync 数据 MSB	8'hD4

4.22 Reg1B Address: 0x1B Default: 0x2D

Bit	Name	Type	Description	Default
7:0	sync_data[31:24]	W/R	sync 数据 MSB	8'h2D

4.23 Reg1D Address: 0x1D Default: 0x80

Bit	Name	Type	Description	Default
7	auto_cdr_std	W/R	数据包解码时，自动计算数据率使能，数据包需先发射前导码	1'b1
6	agc_lock_en	W/R	agc 锁定使能，检测到 pjd_valid 或 sync_valid 有效锁定 AGC 增益 0: disable 1: enable	1'b0
5	dc_force_en	W/R	dc 直接赋值使能，值为 reg[16]dcmn_th	1'b0
4:0	reserved	-	-	5'h00

4.24 Reg1E Address: 0x1E Default: 0x77

Bit	Name	Type	Description	Default
7:4	agc_change_cnt	W/R	agc 计算点数设置，每个 step 为 16 个点	4'h70
3:0	reserved	-	-	4'h07

4.25 Reg1F Address: 0x1F Default: 0xC8

Bit	Name	Type	Description	Default
7:0	agc_hi_thr	W/R	agc 高门限配置	8'hC8

4.26 Reg20 Address: 0x20 Default: 0x80

Bit	Name	Type	Description	Default
7:0	agc_lo_thr	W/R	agc 低门限配置	8'h80

4.27 Reg21 Address: 0x21 Default: 0xB2

Bit	Name	Type	Description	Default
7:0	delta_rssi	W/R	rssi 偏移量	8'hB2

4.28 Reg22 Address: 0x22 Default: 0x20

Bit	Name	Type	Description	Default
7	agc_lock_sel	W/R	agc 锁定使能时，选择有效信号源： 0: sync_valid 1: pjd_valid	1'b0
6:0	reserved	-	-	7'h20

4.29 Reg23 Address: 0x23 Default: 0x1E

Bit	Name	Type	Description	Default
7:6	reserved	-	-	2'b00
5	agc_force_en	W/R	AGC 增益由寄存器 agc_force_ctrl 配置使能： 0: disable 1: enable	1'b0
4:0	agc_force_ctrl	W/R	agc 直接寄存器配置	5'h1E

4.30 Reg24 Address: 0x24 Default: 0x01

Bit	Name	Type	Description	Default
7:4	sda_sel	W/R	芯片 SDA 脚在非 SPI 工作期间的输出信号选择： 0000: demod_bitdata 0001: clkout_sel 0010: rxbit_data 0011: rxbyte_done 0100: rssi_valid 0101: pjd_valid 0110: sync_valid 0111: rxbyte_en 1000: uart_txd 1001: wor_event 1010: spi_so 1011: rx_en 1100: lim_i	4'h00

Bit	Name	Type	Description	Default
			1101: lim_q 1110: 1 1111: 0	
3:2	agc_end_gain_sel	W/R	agc 调节时, 结束 GAIN 选择	2'b00
1:0	agc_start_gain_sel	W/R	agc 调节时, 起始 GAIN 选择	2'b01

4.31 Reg25 Address: 0x25 Default: 0x27

Bit	Name	Type	Description	Default
7:4	reserved	-	-	4'h20
3:0	invalid_av_cnt	W/R	n 个无效信号后 (长 0 或长 1), dc 计算值 恢复到默认值, step 为 16	4'h07

4.32 Reg26 Address: 0x26 Default: 0x20

Bit	Name	Type	Description	Default
7:0	rxbyte_len	W/R	包模式接收时 payload 长度	8'h20

4.33 Reg27 Address: 0x27 Default: 0x03

Bit	Name	Type	Description	Default
7	invalid_av_clr_pkt	W/R	连续长零时, 自动清除包处理模块的 pjd_valid、 sync_valid、fifo: 0: 不清除 1: 自动清除	1'b0
6:3	reserved	-	-	4'h00
2:1	pjd_et_sel	W/R	自动计数数据率时, 计算精度选择	2'b01
0	duc_cmd_sel	W/R	duty cycle 模式, 唤醒后 RC32K 校准选择 0: 唤醒后重新校准 RC32K 1: 唤醒后不校准 RC32K, 只有在上电时校准一次	1'b1

4.34 Reg28 Address: 0x28 Default: 0x70

Bit	Name	Type	Description	Default
7:4	reserved	-	-	4'h70
3:2	mancst_type	W/R	数据包解码时，曼彻斯特编码选择： 00: 无 01: 上升沿为 1，下降沿为 0 10: 上升沿为 0，下降沿为 1 11: 差分曼彻斯特编码	2'b00
1	eom2idle_en	W/R	包接收模式时，接收 payload 完成时，自动进入 IDLE，需要参寄存器 clr_sync 写‘1’清除接收完成标志，然后自动重新进入 RX： 0: disable 1: enable	1'b0
0	eom_en	W/R	包接收模式时，payload 长度由寄存器 rxbyte_len 控制： 0: disable 1: enable	1'b0

4.35 Reg29 Address: 0x29 Default: 0x4D

Bit	Name	Type	Description	Default
7:6	vblo_mxr	W/R	混频器 LO 偏置电压调节： 00: MIN 11: MAX	2'b01
5	bpfvcm_sel	W/R	BPF 共模电压调节： 0: 500mV 1: 650mV	1'b0
4	bpfiq_sel	W/R	BPF 输入 I/Q 相位交换： 0: I 路超前 1: Q 路超前	1'b0
3	bpfi_sle	W/R	BPF 中频频点选择 0: 200kHz 1: 340kHz	1'b1
2	bpfhp_mode	W/R	BPF OP 工作电流选择 0: 400μA 1: 600μA	1'b1

Bit	Name	Type	Description	Default
1:0	bpfbw_sel	W/R	BPF 中频带宽选择 00: 240K 01: 360K 10: 480K 11: NA	2'b01

4.36 Reg2B Address: 0x2B Default: 0xB5

Bit	Name	Type	Description	Default
7	capbank_en	W/R	1: BPF Capbank 模块使能	1'b1
6:4	fcsl_vset	W/R	VCO 校准期间 Vtune 电压设置: 000: 0.6V 001: 0.7V ... 111: 1.3V	3'b011
3:2	rssl_lpbw	W/R	RSSI 低频滤波带宽设置: 00: min 11: max	2'b01
1:0	xosc_agc_sel	W/R	晶振驱动模块反馈电容大小选择: 00: 0fF 01: 77fF 10: 154fF 11: 231fF	2'b01

4.37 Reg2D Address: 0x2D Default: 0x35

Bit	Name	Type	Description	Default
7	vddawo_set	W/R	AWO-LDO 电压设置: 0: 1.9V 1: 1.6V	1'b0
6:0	reserved	-	-	7'h35

4.38 Reg2E Address: 0x2E Default: 0x05

Bit	Name	Type	Description	Default
7:6	gpio_oen	W/R	1: GPIO 输出使能	2'b00
5:4	gpio_dig_en	W/R	1: GPIO 数字 IO 模式使能	2'b00
3:0	reserved	-	-	4'h05

4.39 Reg30 Address: 0x30 Default: 0x00

Bit	Name	Type	Description	Default
7:4	gpio1_sel	W/R	信号选择同 SDA_SEL	4'h00
3:0	gpio0_sel	W/R	信号选择同 SDA_SEL	4'h00

4.40 Reg40 Address: 0x40 Default: 0x00

Bit	Name	Type	Description	Default
7	sync_valid	R	同步字匹配	1'b0
6	pjd_valid	R	数据翻转次数有效	1'b0
5	rss_i_valid	R	rss_i 有效	1'b0
4	pmu_en	R	工作状态: 0: sleep 1: work	1'b0
3:0	reserved	-	-	4'h00

4.41 Reg41 Address: 0x41 Default: 0x00

Bit	Name	Type	Description	Default
7:0	rxbyte_data	R	接收 payload 数据寄存器输出	8'h00

4.42 Reg42 Address: 0x42 Default: 0x00

Bit	Name	Type	Description	Default
7:0	dc_ave	R	dc 中值数据寄存器输出	8'h00

4.43 Reg43 Address: 0x43 Default: 0x00

Bit	Name	Type	Description	Default
7:0	demod_dout	R	demod 解调数据寄存器输出	8'h00

4.44 Reg44 Address: 0x44 Default: 0x00

Bit	Name	Type	Description	Default
7	rxbyte_done	R	数据写入完成, 可以读取数据	1'b0
6:5	reserved	-	-	2'b00
4:0	gainctrl	-	agc 增益控制寄存器输出显示	5'h00

4.45 Reg45 Address: 0x45 Default: 0x00

Bit	Name	Type	Description	Default
7:0	gain	R	agc gain 值寄存器输出显示	8'h00

4.46 Reg46 Address: 0x46 Default: 0x00

Bit	Name	Type	Description	Default
7:0	rssr	R	agc rssr 寄存器输出显示	8'h00

4.47 Reg47 Address: 0x47 Default: 0x00

Bit	Name	Type	Description	Default
7	wor_event	R	wor 工作状态: 0: sleep 1: RX	1'b0
6:5	reserved	-	-	2'b00
4:0	rc32k_ftrim	-	RC32kHz 校准值	5'h00

4.48 Reg48 Address: 0x48 Default: 0x00

Bit	Name	Type	Description	Default
7:0	rxbpf_tune	R	BPF 校准值	8'h00

4.49 Reg49 Address: 0x49 Default: 0x00

Bit	Name	Type	Description	Default
7	gpio_1	R	GPIO1 输入状态	1'b0
6	gpio_0	R	GPIO0 输入状态	1'b0
5:4	reserved	-	-	2'b00
3:0	vco_fcal	-	VCO 校准值	4'h00

4.50 Reg50 Address: 0x50 Default: 0x00

Bit	Name	Type	Description	Default
7:1	reserved	W/R	-	7'h00
0	sleep_cmd	W/R	0: work 1: sleep	1'b0

4.51 Reg51 Address: 0x51 Default: 0x04

Bit	Name	Type	Description	Default
7:4	reserved	W/R	-	4'h00
3:0	work command	W/R	0000: IDLE 0001: IDLE->RX 0010: IDLE->VCO_CAL-> RX 0100: IDLE->RCBPF_CAL->VCO_CAL->RX 1000: IDLE->RC32K_CAL->RCBPF_CAL->VCO_CAL->RX	4'h04

4.52 Reg52 Address: 0x52 Default: 0x00

Bit	Name	Type	Description	Default
7:3	reserved	W/R		5'h00
2	dig_reset	W	写'1'数字复位信号	1'b0
1	clr_rssi_lock	W	写'1'清除 rssi 信号	1'b0
0	clr_sync	W	写'1'清除 rssi_valid,pjd_valid、sync_valid、rxbyte_done 信号	1'b0

4.53 Reg53 Address: 0x53 Default: 0xE5

Bit	Name	Type	Description	Default
7:0	uart_bps	W/R	uart 波特率设置: 115.2Kbps	8'hE5

4.54 Reg55 Address: 0x55 Default: 0xC0

Bit	Name	Type	Description	Default
7:0	rssi_thr	W/R	RSSI valid 门限值	8'hC0

4.55 Reg56 Address: 0x56 Default: 0x00

Bit	Name	Type	Description	Default
7:4	reserved	-	-	4'h00
3	wor_t3_ext_mode	W/R	1: always rx 0: T3 扩展结束后进入 SLEEP	1'b0
2	wor_t2_ext_en	W/R	0: T2 扩展不打开, 无 T2 时间 1: syncword_extend, T2 扩展打开, 并且 sync 同步后进入 T3 时间	1'b0
1:0	wor_ext_sel	W/R	00: 无扩展, 如 wor_t2_ext_en 不使能, T1 后进入睡眠, 如 wor_t2_ext_en 使能, T2+T3 扩展时间 01: rssi_extend 在 T1 期间 rssi 有效, 进入 T2 (wor_t2_ext_en) 或者 T3 扩展 10: pjd_extend 在 T1 期间 pjd 有效, 进入 T2 (wor_t2_ext_en) 或者 T3 扩展 11: rssi_extend & pjd_extend 在 T1 期间 pjd 和 rssi 有效, 进入 T2 (wor_t2_ext_en) 或者 T3 扩展	2'b00

4.56 Reg57 Address: 0x57 Default: 0x7F

Bit	Name	Type	Description	Default
7:0	wor_t2_rx_set[7:0]	W/R	T2 扩展时间, $(N+1) * t_{worclk}$	8'h7F

4.57 Reg58 Address: 0x58 Default: 0x00

Bit	Name	Type	Description	Default
7:0	wor_t2_rx_set[15:8]	W/R	T2 扩展时间, $(N+1) * t_{worclk}$	8'h00

4.58 Reg59 Address: 0x59 Default: 0xFF

Bit	Name	Type	Description	Default
7:0	wor_t3_rx_set[7:0]	W/R	T3 扩展时间, $(N+1) * t_{worclk}$	8'hFF

4.59 Reg5a Address:0x5a Default:0x00

Bit	Name	Type	Description	Default
7:0	wor_t3_rx_set[15:8]	W/R	T3 扩展时间, $(N+1) * t_{worclk}$	8'h00

4.60 Reg60 Address:0x60 Default:0x00

Bit	Name	Type	Description	Default
7:0	rx_fifo_addr	R	FIFO 地址	8'h00

5 功能描述

5.1 接收机

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复数带通滤波器 (Complex-BPF) 进行滤波和镜像抑制，然后由限幅放大器放大后转为单 bit 信号，再送入数字域完成 FSK 解调；同时 RSSI 检测电路将输入信号转换成实时的 RSSI 值，由 ADC 转换到数字域再完成 OOK 解调。

数字电路负责 AGC 和 AFC 控制，并完成滤波和解调。解调后的数据可通过 IO 口直接输出（即直通模式），也可以通过包模式，读 FIFO 进行访问或 UART 输出。

5.2 频率综合器

芯片内部集成了低功耗、全集成的小数分频频率综合器，用于产生下变频混频所需的 I/Q 两路本地振荡信号。满足在 240~1000MHz 范围内产生任意工作频点，频率精度小于 100Hz。为了覆盖足够宽的频率范围，VCO 分为高低两个频段。在使用低频段（315/433MHz）时，寄存器控制位 reg0x03[3]要设为 0，此时 PLL 的参考频率为晶振频率 (f_{xtal}) 的一半；在使用高频段（868/915MHz）时，寄存器控制位 reg0x03[3]要设为 1，此时 PLL 的参考频率为晶振频率。根据所需的输出频率 (f_o)，计算出相应的分频比 (N_{div})，将分频比的整数部分值和小数部分值写到相应的寄存器，就完成了频点的设置。计算过程如下：

1、 低频段： $N_{div}=2* f_o/f_{xtal}$

2、 高频段： $N_{div}= f_o/f_{xtal}$

其中，寄存器 reg0x06[5:0]对应分频比整数部分的值，{reg0x06[1:0]， reg0x05[15:0]}对应分频比小数部分的数值。

相关寄存器如下:

表 5-1: 频率综合器相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x03	3	W/R	fb_sel	频段选择 0: 低频段 1: 高频段
0x05	7:0	W/R	freq_frac[15:8]	频率分频比小数部分
0x06	7:2	W/R	freq_ndiv[5:0]	频率分频比整数部分: $\text{freq_ndiv} = \text{round}(\text{freq}/\text{ref_freq})$, freq 为信道频率, ref_freq 为参考频率, round 为四舍五入
	1:0	W/R	freq_frac[17:16]	频率分频比小数部分: freq_frac 为分频比的小数部分, 最高位为符号位, 有效小数位为 17bit $\text{freq_frac} = \text{freq}/\text{ref_freq} - \text{round}(\text{freq}/\text{ref_freq})$, round 为四舍五入

5.3 AGC

UM2006A 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。

UM2006A 是一个采用低中频架构的 OOK 调制信号接收器。接收器的射频前端由一个低噪声放大器 (LNA)、I/Q 混频器 (Mixer)、中频滤波器 (IF Filter) 和一个宽带功率检测器组成 (WB Power Detector)。射频前端将天线上所感应的射频输入信号放大、下变频至中频, 以进行进一步处理。

借助于射频前端的宽带功率检测器和射频衰减网络, 自动增益控制 (AGC) 环路通过调节射频前端 Mixer、LPF、PGA 的增益, 即使在带外强干扰的环境下也能获得最佳的系统线性度、选择性和灵敏度性能。

相关寄存器如下:

表 5-2: AGC 相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x1D	6	W/R	agc_lock_en	AGC 锁定使能, 检测到 pjd_valid 或 sync_valid 有效锁定 AGC 增益
0x1E	7:4	W/R	agc_change_cnt	AGC 计算点数设置, 每个 step 为 16 个点

寄存器地址	位数	R/W	比特名	功能说明
0x1F	7:0	W/R	agc_hi_thr	AGC 高门限配置
0x20	7:0	W/R	agc_lo_thr	AGC 低门限配置
0x22	7	W/R	agc_lock_sel	AGC 锁定使能时，选择有效信号源： 0: sync_valid 1: pjd_valid
	5	W/R	agc_force_en	AGC 增益由寄存器 agc_force_ctrl 配置使能： 0: disable 1: enable
	4:0	W/R	agc_force_ctrl	AGC 直接寄存器配置
0x24	3:2	W/R	agc_end_gain_sel	AGC 调节时，结束 GAIN 选择
	1:0	W/R	agc_start_gain_sel	AGC 调节时，起始 GAIN 选择

5.4 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。RSSI 值可以通过寄存器 (reg0x46) 直接读取。

UM2006A 可以设置 RSSI_THR[7:0] 的门限值，与检测到的 RSSI 值进行对比，如果 RSSI 检测值大于门限就输出 1，否则输出 0。对比的结果可以输出到 RSSI VLD 中断，也可以用于辅助内部超低功耗 (WOR) 模式的运作。

RSSI 与输入信号的关系图如下：

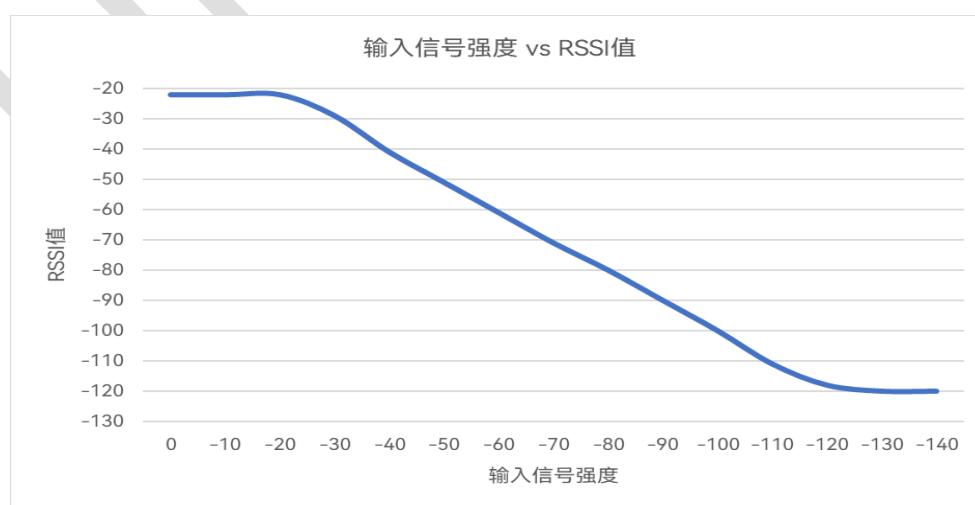


图 5-1: RSSI 与输入信号关系图

相关寄存器如下:

表 5-3: RSSI 相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x21	7:0	R/W	delta_rssi	计算 RSSI 时的预偏移量
0x46	7:0	R	rssi	读出的值为实际 RSSI 的绝对值
0x55	7:0	W/R	rssi_thr	RSSI valid 门限值

5.5 AFC

AFC 功能可以帮助接收机在接收信号的时候, 用很短的时间就能去除 TX 和 RX 之间的频率差异, 以达到更高的灵敏度。

TX 和 RX 之间的频率差异通常是由两者所用晶体的频率偏差导致的, UM2006A 可以通过 AFC 的调节, 把晶体老化导致的频率偏差影响降低到最小, 从而延长产品使用寿命。

5.6 PJD

PJD 是指 Preamble Jump Detector, 即前导码跳变检测器。在芯片进行自适应数据率模式恢复数据时, 可用于通过观察接收信号的跳变特性, 来识别接收的是噪声还是有用信号, 确定 CDR 模块的数据率参数; 还可以通过 PJD 进行直通模式的去噪处理, 使直通输出数据更干净易于辨认有用信息。

相关寄存器如下:

表 5-4: PJD 相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x13	7:5	W/R	pjd_len	接收有效信号翻转次数检测, 检测有效产生 pjd_valid, 对寄存器 clr_sync 写 '1' 或重新进入接收清 pjd_valid。次数为 pjd_len*4
0x27	2:1	W/R	pjd_et_sel	自动计数数据率时, 计算精度选择
0x40	6	R	pjd_valid	数据翻转次数有效

5.7 CDR

在数据包模式下 CDR 系统的基本任务是在接收数据的同时，恢复出与数据率同步的时钟信号，既在芯片内部用于解码，也可以输出到 GPIO 给用户用于采样数据。所以，CDR 的任务很简单也很重要，如果恢复出来的时钟频率跟实际传输的数据率有误差，就会在接收时造成数据采集错误，出现误码，以及解码出错。CDR 的数据率参数来源于寄存器配置，或来源于 PJD 模块自动计算产生，使用 PJD 模块时，信号中至少有两个字节以上的 preamble。

相关寄存器如下：

表 5-5: CDR 相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x1C	7:0	W/R	cdr_std	包解码时的数据率配置
0x1D	7	W/R	auto_cdr_std	数据包解码时，自动计算数据率使能，数据包需先发射前导码

5.8 RC32K

芯片内置一个 32kHz 频率的 RC 振荡器，在 WOR 功能下，RC32kHz 时钟源具有自动校准功能。

手动校准 RC32kHz 流程：

1. 写寄存器：Reg07 = 0xB8
2. 写寄存器：Reg09 = 0x43
3. 写寄存器：Reg30 = 0x11
4. 写寄存器：Reg2E = 0xF5
5. 写寄存器：Reg51 = 0x00
6. 写寄存器：Reg51 = 0x08
7. 延时 7ms 等待校准完成
8. 可通过 GPIO0 查看 RC32kHz 校准结果

相关寄存器如下:

表 5-6: RC32K 相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x02	7:0	W/R	rc32k_cal_std	RC32K 时钟校准目标计数器值
0x07	7	W/R	rc32k_en	rc32k 功能使能, 当 duty cycle 模式使能时 rc32k 功能也会使能
	6	W/R	rc32k_ftrim_sel	1: 选择寄存器 rc32k_frim 0: 选择 32K 校准值
0x08	7:0	W/R	rc32k_ftrim	寄存器配置 RC32K Trimming 值

5.9 WOR

芯片内部集成了定时唤醒器, 集成了 WOR 计数器。定时器采用内部的低功耗 32KHz 时钟源来运行。其工作过程可由下图所示。使能 WOR 命令后, 芯片在工作状态和 SLEEP 状态之间自动定时切换, 在休眠时间窗口内, 芯片处于 SLEEP 状态。在唤醒时间窗口内, 芯片处于接收状态 (包含晶振起振时间)。休眠和唤醒的窗口时间都可以通过寄存器来设定。开启 WOR (duty cycle) 也需要打开 rc32K 使能。WOR 唤醒条件配置详见[6.4 SLP超低功耗接收](#)。

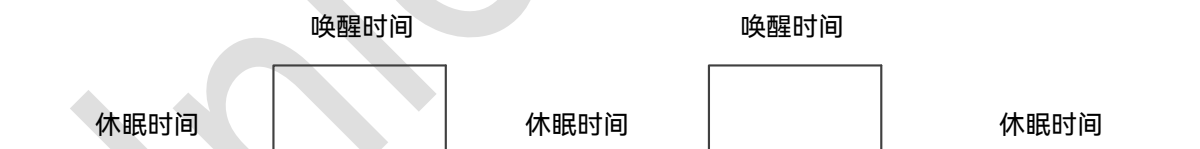


图 5-2: WOR 流程图

相关寄存器如下:

表 5-7: WOR 相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x03	7:6	W/R	duc_sel	duty cycle 模式 (具体时间由寄存器配置): 00: off 01: 2s/1ms 10: 4s/1ms 11: 8s/1ms
0x0E	3:0	W/R	worclk_sel	WOR 功能计算器时钟分频选择: 0000: 32k 0001: 32k/2 0010: 32k/4 --- 1111: 32k/32768
0x27	0	W/R	duc_cmd_sel	duty cycle 模式, 唤醒后 RC32K 校准选择 0: 唤醒后重新校准 RC32K 1: 唤醒后不校准 RC32K, 只有在上电时校准一次

6 芯片运行

6.1 SPI 接口

芯片内置 SPI 从机模块，默认通过三线 SPI 接口 CS、SCLK、SDA 与主机 MCU 进行通讯。SPI 接口可在最大 4MHz 下工作运行。SPI 接口在时钟下降沿变化数据，在时钟上升沿采样数据 (CPOL=1, CPHA=0)。地址和数据都是从 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，后面跟一个字节的数据。地址的 bit7 为读写位，R/W=0 表示读寄存器，R/W=1 表示写寄存器。如果读写连续地址或者访问 FIFO 时，可以在一个 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。

SPI 接口不需要外部时钟就可正常读写寄存器。

SPI 默认是 3 线制，SDA 同时用于数据输入和输出，在读寄存器数据时，芯片会在地址和数据之间对 SDA 的方向进行切换。

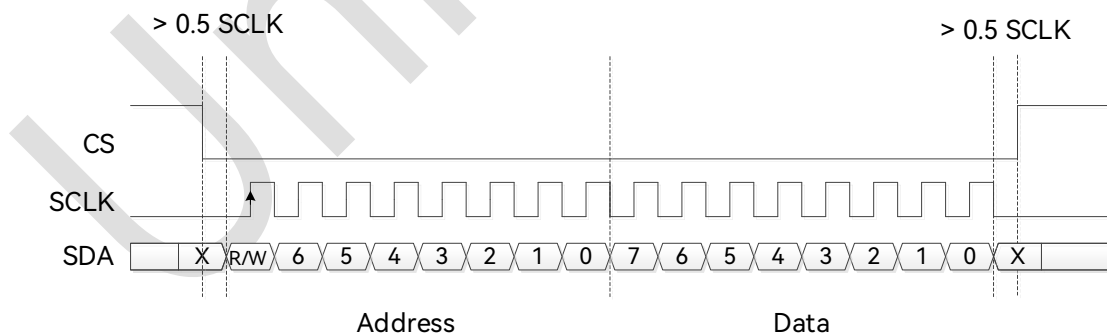


图 6-1: SPI 接口图

注：建议 MCU 在 SCLK 下降沿之前就改变 IO 接口方向，避免 IO 在转换过程中同时为输出，产生电气冲突，引起系统工作异常。

相关寄存器如下:

表 6-1: SPI 相关寄存器列表

寄存器地址	位数	R/W	比特名	功能说明
0x09	5	W/R	spi4w_en	四线 spi 选择, 选择一个 GPIO 作为 SPI 输出。 0: 3 wire 1: 4 wire
0x30	7:4	W/R	gpio1_sel	GPIO1 信号选择: 1010: spi_so, 4 线 SPI 数据输出
	3:0	W/R	gpio0_sel	GPIO0 信号选择: 1010: spi_so, 4 线 SPI 数据输出

6.2 状态机控制

上电后自动进行 RC 校准->VCO 校准->RX, 并一直停留在 RX 状态;

在 RX 状态, SPI 可发送 IDLE 命令使状态机回到 IDLE 状态, 在 IDLE 状态状态机可以进入 RX 状态或 SLEEP 状态。RX 命令分为三种:

1. 不做任何校准操作直接进入 RX
2. 完成 VCO 校准后再进入 RX
3. 完成 RC 校准和 VCO 校准后再进入 RX

SLEEP 和 RX 也可以直接切换, 切换过程中, 芯片内部会自动经历 IDLE 状态。

RX 可以配置成 DutyCycle 模式。进入该模式后, 芯片先自动完成 32K 校准 (可选)、RC 校准和 VCO 校准, 然后进入 RX 状态, 同时内部计数器开始工作, 达到预设的时间后进入 SLEEP 状态。SLEEP 状态超时后, 芯片会自动唤醒然后按相同的流程再次进入 RX 状态, 如此循环往复, 实现 dutycycle 工作模式。

基本模式状态图, 上电后芯片自动进入并停留在 RX 状态。

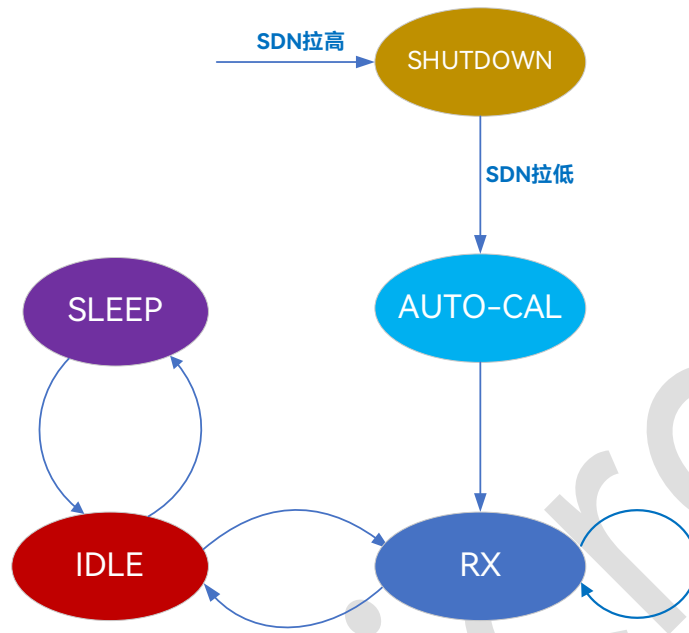


图 6-2: 基本模式状态图

表 6-2: 状态稳定时间

符号	描述	参数以及条件	最小值	典型值	最大值	单位
$t_{SDN-IDLE}$	SHUTDOWN 到 RX	-	-	2000	-	μs
$t_{IDLE-RX}$	IDLE 到 RX	Without cal	-	10	-	μs
		With VCO Cal	-	150	-	μs
		With RC/VCO Cal	-	300	-	μs
		With 32K/RC/VCO Cal	-	5000	-	μs
$t_{IDLE-SLP}$	IDLE 到 SLEEP	-	-	10	-	μs
$t_{SLP-IDLE}$	SLEEP 到 IDLE	-	-	500	-	μs
t_{SLP-RX}	SLEEP 到 RX	With VCO Cal	-	650	-	μs

注: $t_{SLP-IDLE}$, t_{SLP-RX} 时间和晶体本身的特性有很大关系。

6.3 GPIO 和中断

芯片有 2 个 GPIO，作为输出时，可输出信号和中断，信号包括时钟、解调数据、串口输出等。中断包括 RSSI 有效、PJD 有效、同步字有效、接收单字节完成，数据包接收完成中断。

SDA 输出默认为 demod_bitdata 的直通信号，其他输出可以修改 0x24[7:4]来输出不同信号。

GPIO 作为输出数字功能时，需要先对输出使能位 reg0x2e[7:6]置位和数字功能位 reg0x2e[5:4]置位。

相关寄存器如下：

表 6-3: GPIO 和中断相关寄存器列表

寄存器地址	位数	R/W	比特名	功能说明
0x09	4:3	W/R	maskdem_sel	GPIO 输出 demod_bitdata 信号时, maskdem_sel 选择 PJD_valide 有效或 sync_valid 有效或 rssi_valid 后再输出 demod_bit: 00: 不屏蔽 01: rssi_valid 10: PJD_valid 11: sync_valid
	2:0	W/R	clkout_sel	调试, gpio 选择 clkout 信号, 该寄存器选择一种 clk 输出: 000: rxbit_clk 001: demod_out_ce 010: clk_adc 011: clk32k 100: fsm_clk 101: clk_reg 110: clk_fcal 111: synth_clk
0x24	7:4	W/R	sda_sel	芯片 SDA 脚在非 SPI 工作期间的输出信号选择: 0000: demod_bitdata, 直通解调数据输出 0001: clkout_sel, 时钟输出选择, 请参考寄存器 0x09<2:0> 0010: rxbit_data, 接收数据 (CDR 后) 输出 0011: rxbyte_done, 接收数据长度匹配完成 0100: rssi_valid, RSSI 有效输出

寄存器地址	位数	R/W	比特名	功能说明
				0101: pjd_valid, PJD 有效输出 0110: sync_valid, 同步字有效输出 0111: rxbyte_en, 接收单字节完成 1000: uart_txd, 串口输出数据 TXD 1001: wor_event, WOR 功能中断 1010: spi_so, 4 线 SPI 数据输出 1011: rx_en, 接收使能状态 1100: XX, 未定义 1101: XX, 未定义 1110: Logic 1, 输出逻辑高 1111: Logic 0, 输出逻辑低
0x2E	7:6	W/R	gpio_oen	1: GPIO 输出使能
	5:4	W/R	gpio_dig_en	1: GPIO 数字 IO 模式使能
0x30	7:4	W/R	gpio1_sel	信号选择同 SDA_SEL
	3:0	W/R	gpio0_sel	信号选择同 SDA_SEL

6.4 SLP 超低功耗接收

使能 WOR 命令后, 芯片在工作状态和 SLEEP 状态之间自动定时切换, 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 worclk_sel 选择, 其工作流程图如下:



图 6-3: 超低功耗接收流程图

唤醒时间是由 T1、T2、T3 三个时间段组成, 当未收到唤醒信号时, 唤醒时间由 T1 控制, 当收到唤醒信号后, 唤醒时间根据 Reg56[2:0]配置控制。

唤醒监测源分为 RSSI 检测、PJD 检测、Syncword 检测, 唤醒时间由 T1、T2、T3 控制。

收到唤醒信号后的唤醒时间和检测源控制如下表:

表 6-4: 收到唤醒信号后的唤醒时间和检测源控制

Reg56[2:0]	功能说明	RX 延长条件
000	窗口时间为 T1, T1 时间结束后重新进入 SLEEP	固定 T1 时间
001	T1 窗口时间内, 如果检测到有效信号, 则 RX 进入 T3 延长窗口	RSSI 有效, 需要设置 RSSI 门限
010	T1 窗口时间内, 如果检测到有效信号, 则 RX 进入 T3 延长窗口	PJD 有效, PJD 只有自适应数据率模式有效
011	T1 窗口时间内, 如果检测到有效信号, 则 RX 进入 T3 延长窗口	RSSI 和 PJD 同时有效
100	没有 T1 窗口时间, 从 SLEEP 直接进入 T2 窗口时间, T2 窗口时间内如果检测到有效信号, 则 RX 直接进入 T3 延长窗口	T2 时间内 Syncword 有效
101	T1 窗口时间内, 如果检测到有效信号, 则 RX 进入 T2 窗口时间, 在 T2 窗口时间内如果检测到 SYNC 有效, 则 RX 再进入 T3 窗口时间	在 T1 内 RSSI 有效进入 T2, 在 T2 内 SYNC 有效进入 T3
110	T1 窗口时间内, 如果检测到有效信号, 则进入 T2 窗口时间, 在 T2 窗口时间内检测到 SYNC 有效再进入 T3 窗口时间	在 T1 内 PJD 有效进入 T2, 在 T2 内 SYNC 有进入 T3
111	T1 窗口时间内, 如果检测到有效信号, 则进入 T2 窗口时间, 在 T2 窗口时间内检测到 SYNC 有效再进入 T3 窗口时间	在 T1 内 RSSI+PJD 有效进入 T2, 在 T2 内 SYNC 有效进入 T3

时间参数计算公式如下:

$$\text{时间} = (N + 1) * 2^{\text{分频系数}} / \text{clk}$$

如 RC=32kHz, 分频系数为 1, N 可以通过寄存器设置值, 如 T1 设置为 0x000A, 时间 $t = (0x000A+1) * 2^1 / 32\text{kHz} = 0.6875\text{ms}$

在 T3 时间结束后, 可以选择重新进入 SLEEP 继续执行 WOR 功能, 或者一直停留在 RX。

相关寄存器如下:

表 6-5: 低功耗接收相关寄存器列表

寄存器地址	位数	R/W	比特名	功能说明
0x03	7:6	W/R	duc_sel	Duty cycle 模式 (具体时间由寄存器配置): 00: off 01: 2s/1ms 10: 4s/1ms 11: 8s/1ms

寄存器地址	位数	R/W	比特名	功能说明
0x0E	3:0	W/R	worclk_sel	WOR 功能计算器时钟分频选择 0000: 32k 0001: 32k/2 0010: 32k/4 --- 1111: 32k/32768
0x51	3:0	W/R	work command	0000: IDLE 0001: IDLE->RX 0010: IDLE->VCO_CAL->RX 0100: IDLE->RCBPF_CAL->VCO_CAL->RX 1000: IDLE->RC32K_CAL->RCBPF_CAL->VCO_CAL->RX
0x0A	7:0	W/R	wortimer_set[7:0]	WOR 功能, SLEEP 计数器总周期数, $(N+1) * t_{worclk}$
0x0B	7:0	W/R	wor_rtxtimer_set[15:8]	WOR 功能计算器唤醒后, 开窗 T1 计数长度, $(N+1) * t_{worclk}$
0x0C	7:0	W/R	wor_rtxtimer_set[7:0]	WOR 功能计算器唤醒后, 开窗 T1 计数长度, $(N+1) * t_{worclk}$
0x0D	7:0	W/R	wor_rtxtimer_set[15:8]	WOR 功能计算器唤醒后, 开窗 T1 计数长度, $(N+1) * t_{worclk}$
0x56	3	W/R	wor_t3_ext_mode	1: always rx 0: T3 扩展结束后进入 SLEEP
	2	W/R	wor_t2_ext_en	0: T2 扩展不打开, 无 T2 时间 1: syncword_extend, T2 扩展打开, 并且 sync 同步后进入 T3 时间
	1:0	W/R	wor_ext_sel	00: 无扩展, 如 wor_t2_ext_en 不使能, T1 后进入睡眠, 如 wor_t2_ext_en 使能, T2+T3 扩展时间 01: rssi_extend 在 T1 期间 rssi 有效, 进入 T2 (wor_t2_ext_en) 或者 T3 扩展 10: pjd_extend 在 T1 期间 pjd 有效, 进入 T2 (wor_t2_ext_en) 或者 T3 扩展 11: rssi_extend & pjd_extend 在 T1 期间 pjd 和 rssi 有效, 进入 T2 (wor_t2_ext_en) 或者 T3 扩展
0x57	7:0	W/R	wor_t2_rx_set[7:0]	T2 扩展时间, $(N+1) * t_{worclk}$
0x58	7:0	W/R	wor_t2_rx_set[15:8]	T2 扩展时间, $(N+1) * t_{worclk}$
0x59	7:0	W/R	wor_t3_rx_set[7:0]	T3 扩展时间, $(N+1) * t_{worclk}$
0x5A	7:0	W/R	wor_t3_rx_set[15:8]	T3 扩展时间, $(N+1) * t_{worclk}$

7 数据处理机制

UM2006A 支持直通数据和数据包模式。数据包模式接收到的 Payload 数据可以选择从 FIFO 读出或从 UART 输出。当 Sync_en 为 disable 时，为直通模式。当 Sync_en 为 enable 时，为数据包模式。芯片默认为直通模式。

相关寄存器如下：

表 7-1: 数据处理相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x17	7	W/R	sync_en	数据包解码使能，匹配 sync_data 后按字节接收数据 0: disable 1: enable

7.1 直通模式

Direct 直通模式是指芯片接收到信号进行解调后，通过 SDA 或者 GPIO 直接输出。

在 Direct 直通模式下，FIFO 和 UART 不工作。系统上电时默认为直通模式，在 CS 为高期间，SDA 为默认直通的数据输出，数据时钟可以通过配置对应寄存器，使其从 GPIO0 或者 GPIO1 输出。

直通模式工作流程如下：

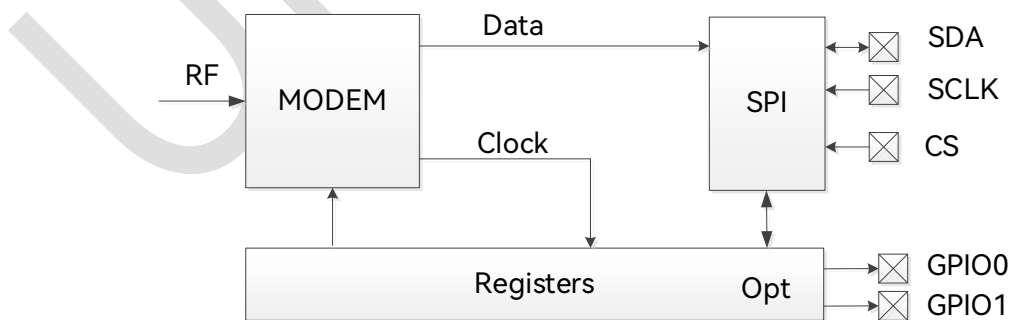


图 7-1: 直通模式工作流程图

相关寄存器如下:

表 7-2: 直通模式相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x24	7:4	W/R	sda_sel	芯片 SDA 脚在 CS 为高时输出信号选择 0000: demod_bitdata, 直通解调数据输出
0x30	7:4	W/R	gpio1_sel	GPIO1 信号选择: 0000: demod_bitdata, 直通解调数据输出
	3:0	W/R	gpio0_sel	GPIO0 信号选择: 0000: demod_bitdata, 直通解调数据输出

7.2 Packet 数据包模式

Packet 数据包模式由 Preamble、Syncword、Payload Data 三部分组成, 如下图:

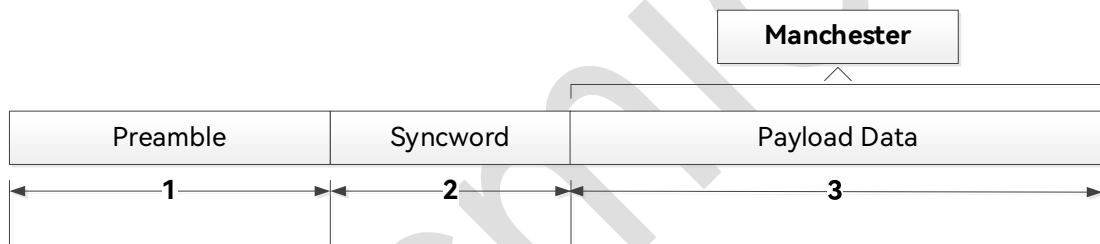


图 7-2: Packet 模式结构图

Syncword 的长度支持 16bits 和 32bits。

Payload 数据支持曼彻斯特解码和差分曼彻斯特解码, Packet 数据包模式的 Payload 长度由寄存器 Reg0x28[0]控制, 当 Reg0x28 [0] = 0 时, 芯片一直处于接收状态接收数据; 当 Reg0x28[0]=1 时, 数据包长度由寄存器 Reg26(rxbyte_len)控制。数据包接收完 payload 后, 芯片的状态由寄存器 reg0x28[1]控制, 当 reg0x28[1] = 0 时, 芯片保持在接收状态; 当 reg0x28[1] =1 时, 芯片进入 IDLE 状态。Payload 数据支持 FIFO 读取和 UART 输出。

数据包工作流程如下：

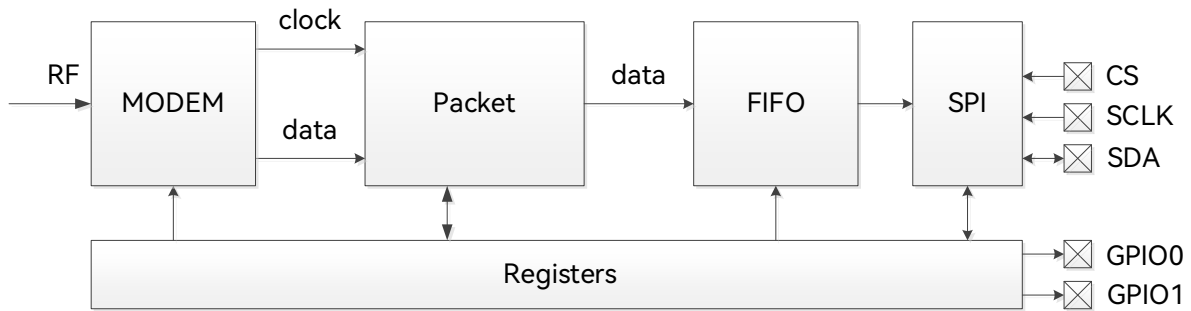


图 7-3: Packet 模式工作流程图

相关寄存器如下：

表 7-3: Packet 模式相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x17	6	sync_len	W/R	0: 16-bit sync_data 1: 32-bit sync_data
0x26	7:0	rxbyte_len	W/R	包模式接收时 payload 长度
0x28	3:2	mancst_type	W/R	数据包解码时，曼彻斯特编码选择： 00: 无 01: 上升沿为 1，下降沿为 0 10: 上升沿为 0，下降沿为 1 11: 差分曼彻斯特编码
	1	eom2idle_en	W/R	包接收模式时，接收 payload 完成时，自动进入 IDLE，需要参寄存器 clr_sync 写‘1’清除接收完成标志，然后自动重新进入 RX： 0: disable 1: enable
	0	eom_en	W/R	包接收模式时，payload 长度由寄存器 rxbyte_len 控制： 0: disable 1: enable

7.2.1 FIFO 模式

芯片的 FIFO 共 32 bytes，当芯片每接收到一个字节的 Payload 数据后，就会写入 FIFO 中。

如果接收到的 Payload 数据的长度超过 32bytes，FIFO 写满后不再写入 FIFO 中。当芯片接收完成

中断产生后，MCU 可通过 SPI 直接读取 FIFO 数据。读取 FIFO 数据时，访问 FIFO 对应的地址，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，从而增加读取数据的效率。

相关寄存器如下：

表 7-4: FIFO 模式相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x24	7:4	W/R	sda_sel	芯片 SDA 脚在 CS 为高时输出信号选择 0011: rxbyte_done,接收数据长度匹配完成
0x30	7:4	W/R	gpio1_sel	GPIO1 信号选择: 0011: rxbyte_done, 接收数据长度匹配完成
	3:0	W/R	gpio0_sel	GPIO0 信号选择: 0011: rxbyte_done, 接收数据长度匹配完成
0x44	7	R	rxbyte_done	数据写入完成，可以读取数据
0x52	0	W	clr_sync	写‘1’清除 rssi_valid,pjd_valid、sync_valid、 rxbyte_done 信号
0x60	7:0	R	rx_fifo_addr	FIFO 地址

7.2.2 UART 模式

芯片可选 SDA 或一个 GPIO 作为 UART 输出，当每接收到一个字节的 Payload 数据后，就自动通过串口输出给 MCU 进行数据解析，默认 UART 数据率 115200，1 个起始位，1 个停止位，无校验位，修改寄存器可改变 UART 波特率，波特率计算公式如下：

$$\text{Reg53} = (f_{\text{xtal}} + 0.5\text{MHz}) / \text{uart_bps} - 1$$

其中： f_{xtal} 为晶振，uart_bps 为波特率。

如：晶振为 26MHz，uart_bps 为 115200bps

$$\text{Reg53} = (26+0.5) * 1000000 / 115200 - 1 = 0xE5$$

相关寄存器如下：

表 7-5: UART 模式相关寄存器

寄存器地址	位数	R/W	比特名	功能说明
0x03	0	W/R	uart_en	UART 发射模块使能, 在包模式接收时, 收到的 payload 数据通过 uart 模块发出, 默认 115200Hz 0: disable 1: enable
0x24	7:4	W/R	sda_sel	芯片 SDA 脚在 CS 为高时输出信号选择 1000: uart_txd
0x30	7:4	W/R	gpio1_sel	GPIO1 信号选择: 1000: uart_txd
	3:0	W/R	gpio0_sel	GPIO0 信号选择: 1000: uart_txd
0x53	7:0	W/R	uart_bps	UART 波特率设置: 115.2kbps

8 典型应用

8.1 参考电路图

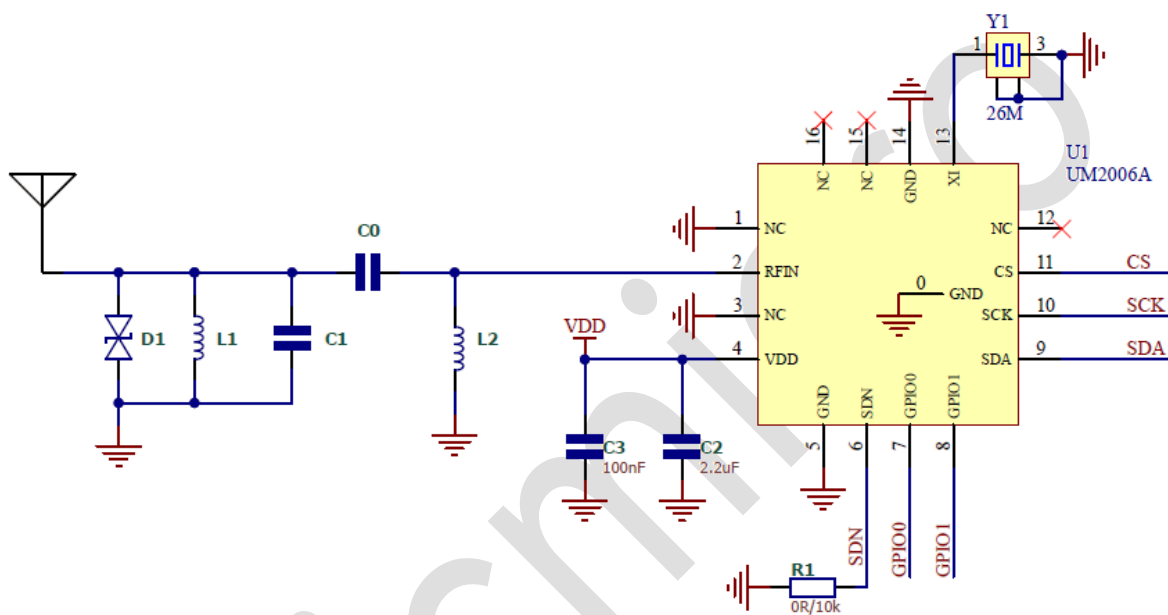


图 8-1: 参考电路图

应用说明:

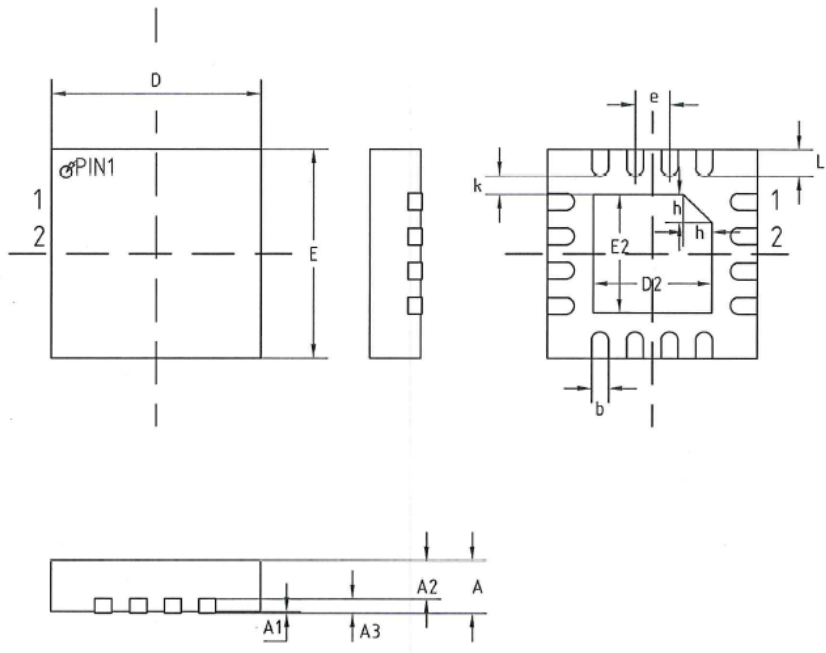
1. SDN 为芯片关断功能脚，高电平关断。如果该引脚受 MCU 控制，建议接一个 10k 左右的下拉电阻。如果应用中不需要关断功能，可以将 SDN 脚接地；
2. D1 为 ESD 保护管，是否需要，可根据应用要求来定。

8.2 器件参数

位号	描述	元件值				单位
		315	433	868	915	MHz
C0	±5%, 0402, X7R, 50V	3.9	2.7	1.8	1.5	pF
C1	±5%, 0402, NP0, 50V	12	10	4.7	3.9	pF
C2	±10%, 0402, C0G, 50 V	2.2	2.2	2.2	2.2	uF
C3	±10%, 0402, X7R, 50V	100	100	100	100	nF
R1	±10%, 0402, 50V	10	10	10	10	kΩ
L1	±5%, 0402, 高频电感, Sunlord	33	18	12	8.2	nH
L2	±5%, 0402, 高频电感, Sunlord	47	33	10	10	nH
D1	ESD 保护二极管	NC	NC	NC	NC	NC
Y1	3225 贴片无源晶振±20PPM 18pF, YXC (扬兴晶振)	26	26	26	26	MHz

9 封装尺寸

9.1 QFN16 (3*3mm)



Dimensions/mm			
SYMBOL	MIN	NOMINAL	MAX
A	0.70	0.75	0.80
A1	0	0.02	0.05
A2	-	0.55	-
A3	0.203 REF		
b	0.18	0.24	0.30
D	3 BSC		
E	3 BSC		
e	0.50 BSC		
D2	1.6	1.7	1.8
E2	1.6	1.7	1.8
K	0.20 BCS		
L	0.30	0.40	0.50
h	0.35	0.40	0.45

图 9-1: QFN16 封装尺寸图

10 版本维护

版本	日期	描述
V1.0	2024.10.17	初始版
V1.1	2025.05.16	更新 SLP 超低功耗接收相关参数及部分器件参数。