

## 软件部分

发射接收流程:

1.发射流程:

⊙07 寄存器 0X00 0X00 (IDLE)

⊙52 寄存器 0X80 0X80

⊙08 寄存器 0X6C 0X90

⊙在写 50 寄存器后, 连续写需要发送的数据, 第一个字节代表发射的长度 (数据包最长 32byte)

⊙写完后 FIFO 再写 07 寄存器高字节 0X01, 低字节为设置的频点, 发射时间大约 3.5ms (62.5Kbps 15 个字节) 的时间, 在此时间无需对 RF 寄存器进行其他操作, 可处理应用其他的事情。

⊙07 寄存器 0X00 0X00 (IDLE)

需发数据再跳到第一步

2.接收流程:

⊙07 寄存器 0X00 0X00 (IDLE)

⊙52 寄存器 0X80 0X80

⊙08 寄存器 0X6C 0X90

⊙切换 0x38 寄存器配置(0x9C9F 和 0x9BFF 切换)

⊙07 寄存器 00, 高字节 0x00, 低字节最高位时置 1 为接收状态, BIT6~BIT0 设置频点, 进入接收。

⊙通过每 3ms(间隔可根据时序调整)进行一次读取 03 寄存器, 高字节的 5BIT 产生变化来判断是否收取信号, 没有信号则循环, 如果等待超时返回第一步

⊙如果收到数据, 写 07 寄存器 00 00 (空闲状态), 再读取 50 寄存器 FIFO 数据。第一字节是接收时的长度, 如是连续读取动作可把所需求的数据都读取出来(内部 FIFO 数据指针自动累加)。

为数据准确性, 对数据进行累加校验或者其他校验计算, 加深数据严谨性。

⊙用户处理收到的数据

⊙数据处理结束, 跳到第一步, 以此重复此类动作。

接收窗口要在发射信号前打开, 如果没有打开, 信号是无法收到的, 要做好通讯窗口对齐同步

通信条件需求:

⊙36 和 39 寄存器 (sync word) 地址配置、44 和 45 寄存器速率配置, 07 寄存器的频点 (低字节 BIT6~BIT0) 须相同、40 寄存器容错配置设置为 0x4402。

唤醒和睡眠:

⊙睡眠的时候操作写入 07 寄存器 00 00 (IDLE), 27 寄存器设置为 0X13 0X00, 35 寄存器的高字节的第 6BIT 位置 1 完成睡眠。

⊙唤醒的时候操作写入 0x38 寄存器 bit0 置 0 (0x9C9E 或者 0x9BFE), 延时大于 2mS, 再写入 0x38 寄存器 bit0 置 1(0x9C9F 或者 0x9BFF)。

注意事项:

1: RF76 的 IIC 设备地址为: 0x50 (写操作)、0x51 (读操作)

2: 读数据的时候, 寄存器地址位最高位必须写 1。写数据的时候, 寄存器地址位最高位必须为 0。

3: 写数据的时候, 高字节 CLK 准备数据在 SDA 上, 再写低字节。

4: 读数据的时候, 在 CLK 为高后 1uS 再去读取高低电平。STOP 停止位实现后 CLK 一定拉成低电平。

- 5: 在 RF76 的 IIC 协议中的 ACK, NACK 信号由单片机输出。在读取 FIFO 数据的时候 ACK、NACK 信号要求 CLK 高电平宽度不低于 200nS, 如果短于该时间值, 读取的数据是错误的。其他操作可以低于 200nS。
- 6: RF76 的 IIC 协议使能达到 M 级以上, TX 模式使能发射完成以后, 无需读取 03 寄存器! 建议 3ms (62.5Kbps 15 个字节) 内不要操作使能 TX 模式 (因为 TX 在此时间段还处于发射中)。
- RX 模式时不要频繁读取 03 寄存器 (PKT 判断寄存器), 会使 RX 模式收取信号时有所影响, 建议 3ms(间隔可根据时序调整)读取一次, 软件在此期间能够处理其他的工作。
- 7: RF76 需要注意到相邻 4M 发射的频点是泄露, 能够收取信号, 使用的时候需要注意到此问题。
- 8: 调试时需要注意到睡眠的情况下, 必须通过 IIC 使能唤醒 RF76, 而使能复位的情况下是没办法实现唤醒功能, 使能 TX/RX 模式的情况下, 使能 08 寄存器为 0X6C90。
- 9: 在 FIFO 读取, 假设读取长度为 15 个字节, 那么在第 14 个字节需要给 ack 应答信号, 读取第 15 个字节后, 要给 nack 信号。
- 10: 各个长度、速率所对发射时长, 如下: (数据为参考时长, 通过 IIC 读取得到, Reg32=0x4800)

| 字节 | 时长<br>uS@1Mbps | 时长<br>uS@250Kbps | 时长<br>uS@125Kbps | 时长<br>uS@62.5Kbps |
|----|----------------|------------------|------------------|-------------------|
| 1  | 347            | 543              | 816              | 1343              |
| 2  | 347            | 582              | 875              | 1481              |
| 3  | 367            | 602              | 934              | 1598              |
| 4  | 367            | 641              | 1012             | 1735              |
| 5  | 367            | 680              | 1070             | 1871              |
| 6  | 386            | 699              | 1129             | 1988              |
| 7  | 386            | 738              | 1187             | 2125              |
| 8  | 406            | 777              | 1265             | 2242              |
| 9  | 406            | 797              | 1324             | 2379              |
| 10 | 406            | 836              | 1382             | 2497              |
| 11 | 425            | 875              | 1461             | 2633              |
| 12 | 425            | 894              | 1520             | 2750              |
| 13 | 445            | 933              | 1578             | 2887              |
| 14 | 445            | 953              | 1637             | 3004              |
| 15 | 445            | 992              | 1715             | 3141              |
| 16 | 465            | 1031             | 1774             | 3277              |
| 17 | 465            | 1051             | 1832             | 3394              |
| 18 | 485            | 1090             | 1910             | 3532              |
| 19 | 485            | 1129             | 1969             | 3649              |
| 20 | 485            | 1148             | 2027             | 3786              |
| 21 | 504            | 1187             | 2086             | 3903              |
| 22 | 504            | 1226             | 2164             | 4039              |
| 23 | 524            | 1246             | 2222             | 4156              |
| 24 | 524            | 1285             | 2281             | 4293              |
| 25 | 524            | 1304             | 2359             | 4430              |
| 26 | 543            | 1343             | 2418             | 4548              |
| 27 | 543            | 1382             | 2477             | 4684              |
| 28 | 562            | 1402             | 2536             | 4801              |
| 29 | 562            | 1441             | 2614             | 4938              |

|    |     |      |      |      |
|----|-----|------|------|------|
| 30 | 582 | 1481 | 2672 | 5055 |
| 31 | 582 | 1500 | 2731 | 5192 |
| 32 | 582 | 1539 | 2789 | 5309 |

## 关于 0x38 寄存器

建议值:

- 1、0X9c9F, 这个配置下, 芯片内部的 LDO 电压低一点, 所以灵敏度会低, 阻塞小, 适应近距离接收
- 2、0x9bFF, 这个配置下, 芯片内部的 LDO 电压高一点, 所以灵敏度会高, 阻塞大, 适应远距离接收

切换配置值:

在具体的应用中, 需要远近距离通讯, 根据上面的建议值, 在使用过程中可以不断切换 0x38 寄存器的配置, 来做到兼顾远近距离通讯

参考代码:

```
bit RegSetCPL;
void RF76_EnterRx(unsigned char rf_channel)
{
    RF76_WriteReg(7,0,rf_channel);
    RF76_WriteReg(8,0x6c,0x90);
    RF76_WriteReg(52, 0x80, 0x80);
    if(RegSetCPL)
    {
        RegSetCPL=0;
        RF76_WriteReg(0x38, 0x9c, 0x9F);
    }
    else
    {
        RegSetCPL=1;
        RF76_WriteReg(0x38, 0x9b, 0xFF);
    }
    RF76_WriteReg(7,0,rf_channel|0X80);
}
```

功能操作:

```
case RF76_SLEEP:
    RF76_write_reg(7, 0x00, 0x00);
    RF76_write_reg(27, 0x13, 0x00);
    RF76_write_reg(35, 0x43, 0x00);
    break;
case RF76_WARKUP:
    RF76_write_reg(0x38, 0x9b, 0xfe);
```

```
    delay_MS(3);
    RF76_write_reg(0x38, 0x9b, 0xff);
    delay_MS(1);
    break;
case CLEAR_FIFO:
    RF76_write_reg(52, 0x80, 0x80);
    break;
case RF76_IDLE:
    RF76_write_reg(7, 0x00, 0x00);
    break;
case RF76_SPAN_1M:
    RF76_write_reg(44, 0x01, 0x01);    //
    RF76_write_reg(45, 0x05, 0x52);    //
    break;
case RF76_SPAN_512K:
    RF76_write_reg(44, 0x02, 0x01);    //
    RF76_write_reg(45, 0x05, 0x52);    //
    break;
case RF76_SPAN_250K:
    RF76_write_reg(44, 0x04, 0x01);    //
    RF76_write_reg(45, 0x05, 0x52);    //
    break;
case RF76_SPAN_125K:
    RF76_write_reg(44, 0x08, 0x01);    //
    RF76_write_reg(45, 0x05, 0x52);    //
    break;
case RF76_SPAN_62K:
    RF76_write_reg(44, 0x10, 0x01);    //
上电复位:
    delay_MS(100);
    RF76_write_reg(0x38, 0x00, 0x00);
    delay_MS(5);
    RF76_write_reg(0x38, 0x9b, 0xff);
    delay_MS(5);
```

查看频偏流程:

- 1、上电初始化配置
- 2、进入接收模式(比如频点为 2450MHz)
- 3、频谱仪设置中心频点为 2451MHz，SPAN 设置 300KHz，LEVEL 设置-20dBm，按下 PEAK 便可得到频点频率值。
- 4、以 2451MHz 计算出频偏值

频率微调:

28 寄存器 (0x1C)，默认值 0x1800 时使用我们 3225 晶体正偏接近 1M，具体值根据晶体参数不同有差异

可调节范围为八档 0x1800 至 0x1807 每加一档频偏往负调节大概 300 多 KHz

发射功率配置表:

| Reg0x09 | 功率<br>(dBm) | 峰值电流(mA) |
|---------|-------------|----------|
| 0x7830  | 8.9         | 32       |
| 0x0830  | 7.8         | 29       |
| 0x0930  | 6.22        | 24       |
| 0x0A30  | 4.9         | 20       |
| 0x0B30  | 3.35        | 19       |
| 0x0C30  | 1.15        | 17       |
| 0x0D30  | -1.65       | 16       |
| 0x0E30  | -4.47       | 15       |
| 0x0F30  | -6.97       | 14       |
| 0x3F30  | -9.17       | 13       |
| 0x0FB0  | -9.62       | 13       |

## 硬件部分

### 晶体电路：

RF76 是单端晶体电路，对 XTAL\_IN 和 GND 的寄生电容电感敏感，设计 PCB 需注意，如下：

- ◎晶体下不要走电源属性网络。
- ◎天线与晶体距离不能靠太近，会相互干扰，需分开摆放。  
( 如需 24MHz 晶体样品，可与我司申请样品 )

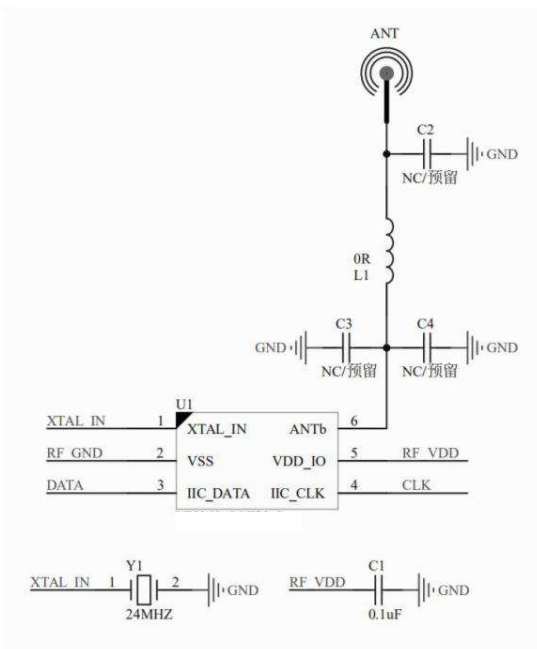
### 电源电路：

- ◎滤波电路尽量靠近芯片相应管脚，例如 RFVDD 旁的 1uF 电容。
- ◎电源纹波最大不要大于 100mV，带载能力大于 100mA，最低不能低于 2.1V。
- ◎大电流线路尽量分开处理，分流走线到公共 GND 上。
- ◎为确保 RFVDD 滤波干净，要增加 RC 滤波电路，再单独供给 RFVDD。

### IIC 走线：

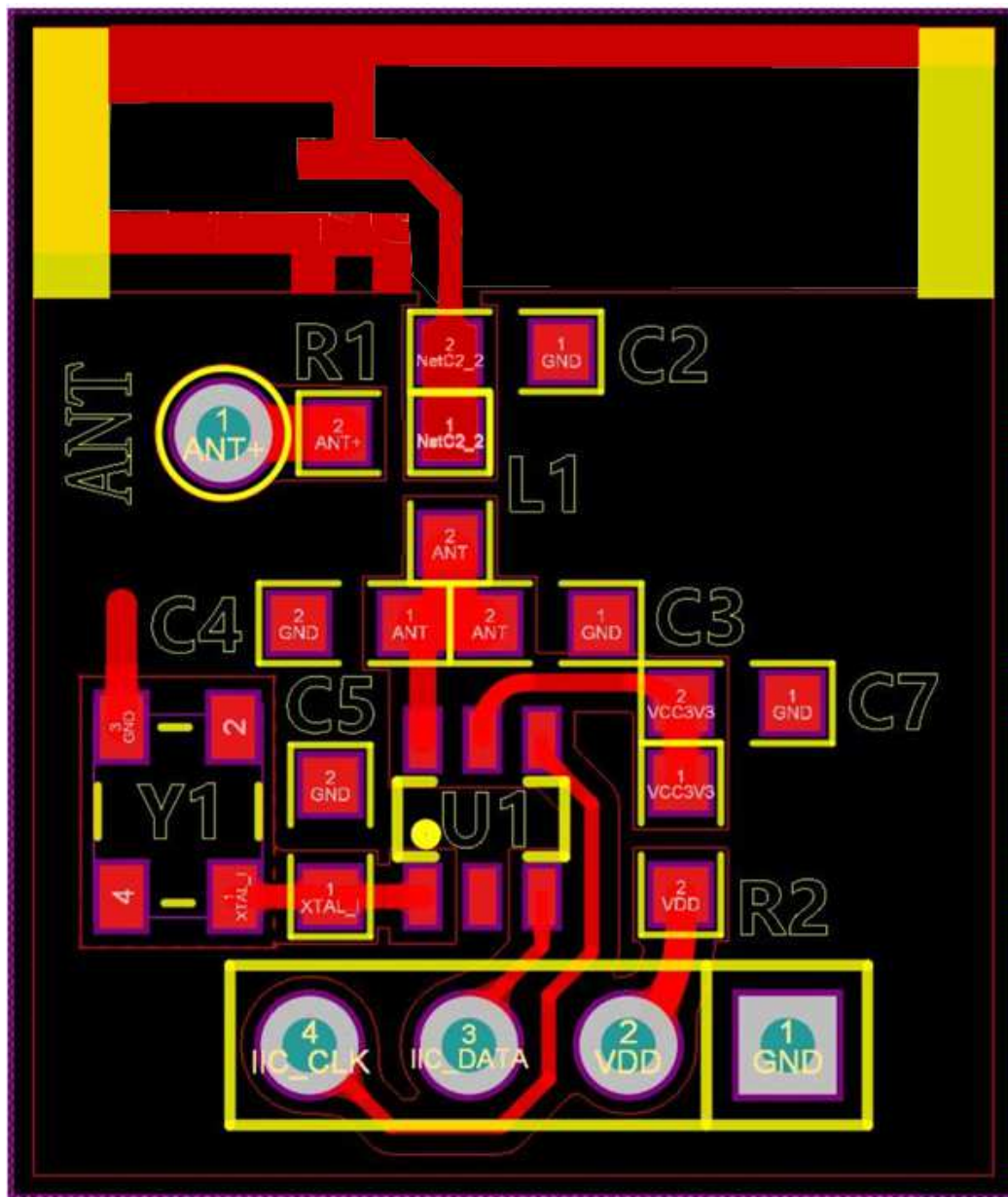
- ◎IIC 总线到 MCU 端走线尽量短，两端用完整地包裹以增强抗干扰性。

### FCC 认证电路：



- ◎如上图所示：在芯片射频脚跟天线之间预留该电路即可。（建议按照参考图摆放位置）
- ◎RF76 最大发射功率理论值为 10dBm，实际输出功率跟 PCB 设计与芯片封装有直接关系。

PCB 参考设计:



◎R1=1.2pF（跨接33mm 导线天线时可用）

◎R2=10R、C7=1uF（RC 滤波电路）

◎Y1=24MHz、C5=2pF（晶体端匹配电容可以调节频偏以及降低芯片自身本振辐射从而进一步提升通讯质量）

◎L1=1.2pF、C2/C3/C4=NC（FCC 认证预留电路）

◎如不需过认证，在 TX RX 端可以在L1 位置放置1.2pF 左右电容，可改善芯片端到天线端的匹配度。具体值可根据不同 PCB 版型适当调整。