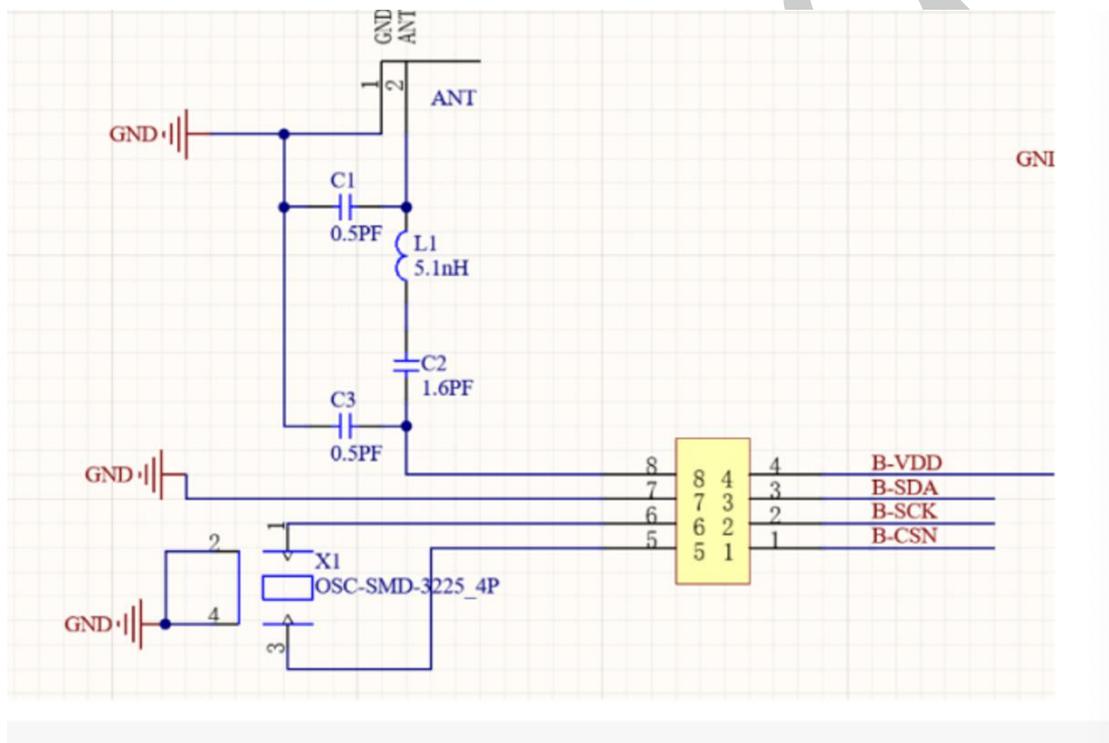




# XL2400T 系列 FCC 认证设计参考

硬件设计：



注：C1,C3可选择性加，建议预留元器件位置。

设计电路时可预留 C1,C3 电容的位置，测试时可根据需要进行添加。

## 设计建议:

1. 按常规射频电路设计，重点关注天线布局--天线远离数字电路，电源模块，减少干扰耦合；避免天线与 PCB 边缘过近，防止信号反射导致杂散辐射。
2. 在 RF 芯片电源引脚外接滤波电容（如 0.1  $\mu$ F 陶瓷电容 + 10  $\mu$ F 电解电容），滤除电源噪声对射频信号的干扰，降低杂散辐射。
3. RF 芯片电源引脚外接滤波电容（0.1uf 陶瓷电容+10uf 电解电容），滤除电源噪声对射频信号的干扰，降低杂散辐射。

## 软件关键配置:

1. 芯片初始化时，需关闭发射和接收的应答功能，修改 RF 芯片初始化代码；
2. 避免频繁调用频点设定函数，设定固定频点（如 2405MHz）后，每隔 2-3 秒调用一次即可，防止载波频点两侧出现干扰波；
3. 禁止完全不调用频点设定函数，否则会导致频点漂移。
4. 需覆盖高、中、低三个频段，每个频段对应数据发射、单载波、数据接收 3 种状态，共 9 种测试场景。
5. 固定选择三个认证频点：2405MHz（低）、2440MHz（中）、2480MHz（高）。

## 测试模式触发:

1. 选用 1 个闲置 IO 口，上电设为上拉输入；
2. 拉低该 IO 口进入测试模式，通过该 IO 口状态切换不同频点

和测试模式；

```
RESTART:
    LCALL    FT_TEST
    LCALL    IO_CONFIG
    LCALL    RAM_INIT
    LCALL    PWM_INIT
    LCALL    TIMER0_INIT
    LCALL    LOAD_RX_ROLL
    LCALL    DELAY_50MS
    LCALL    DELAY_50MS
    LCALL    INIT_RF
    LCALL    RF_CARRIER_TEST
```

3. 进入接口后首先扫描按键，判断认证触发的按键 IO 口是否处于按下状态，如果处于按下状态并超过一定时间（50MS 左右），即认为进入 FCC 认证模式。

```
RF_CARRIER_TEST:
    LCALL    KEY_SCAN                ;按键扫描
    MOVR     R_KEY_BUF, T0_A
    XORIA    0X01
    BTRSS    STATUS, Z_B
    RET
    LCALL    DELAY_50MS
    LCALL    KEY_SCAN                ;延时50ms后再次扫描
    MOVR     R_KEY_BUF, T0_A
    XORIA    0X01
    BTRSS    STATUS, Z_B
    RET
    MOVIA    1                        ;进入FCC认证模式
    MOVAR    R_SEND_CARRIER
    MOVAR    R_KEY_OLD
    MOVAR    R_KEY_STATE
    LCALL    RF_SET_TX_MODE          ;设为发射模式
    LCALL    CARRIER_SET_HANDLER
```

4. 进入认证模式后认证模式一共有 9 种测试模式，工作在三个

频点 2405、2440、2480. 默认前三种为数据发送模式，中间三种为单载波模式，最后三种为数据接收模式。每按一次按键顺序切换一种模式。其中工作频点, 工作模式和模式顺序可以根据实际情况调整。另外可以通过 LED 灯的闪烁次数获知当前处于哪种测试模式

```

RF_CARRIER_LOOP:
    BTRSS    F_TMR_8MS
    LGOTO    RF_CARRIER_LOOP
    BCR      F_TMR_8MS
    CLRWDT
    LCALL    KEY_SCAN           ;模式切换按键扫描
    LCALL    KEY_DEBOUNCE
    LCALL    KEY_HANDLE
    LCALL    CARRIER_SET_HANDLER ;9种测试模式切换

    BTRSS    F_TMR_16MS_TEST
    LGOTO    RF_CARRIER_LOOP
    BCR      F_TMR_16MS_TEST
    LCALL    CARRIER_MODE_LED_SET ;模式状态LED灯指示
    LGOTO    RF_CARRIER_LOOP
    
```

模式切换按键检测:

模式切换按键检测:

```

KEY_HANDLE:
    MOVR    R_KEY_STATE, TO_A
    ANDIA   0X01
    XORAR   R_KEY_OLD, TO_A
    BTRSC   STATUS, Z_B
    RET
    MOVR    R_KEY_STATE, TO_A
    ANDIA   0X01
    MOVAR   R_KEY_OLD

    MOVR    R_KEY_OLD, TO_A
    BTRSC   STATUS, Z_B
    RET
    BSR     F_CARRIER_CHANGE
    INCR    R_SEND_CARRIER, TO_R ;模式切换变量, 每按一次按键切换一次模式, 1-9循环
    MOVIA   10
    SUBAR   R_SEND_CARRIER, TO_A
    BTRSS   STATUS, C_B
    LGOTO   LED_TIME_SET
    MOVIA   1
    MOVAR   R_SEND_CARRIER

LED_TIME_SET:
    MOVR    R_SEND_CARRIER, TO_A
    MOVAR   R_LED_LOOP_TIME ;LED灯闪烁次数显示当前处于哪种模式
    RET

```

9 种工作模式切换:

```

CARRIER_SET_HANDLER:
    MOVIA   7
    MOVAR   PCHBUF
    MOVR    R_SEND_CARRIER, TO_A ;模式切换变量
    ADDAR   PCL, TO_R

    LGOTO   CARRIER_SET_NEXT_SETP
    LGOTO   CARRIER_SET_NEXT_SETP
    LGOTO   CARRIER_SET_NEXT_SETP_00
    LGOTO   CARRIER_SET_NEXT_SETP_01
    LGOTO   CARRIER_SET_NEXT_SETP_02
    LGOTO   CARRIER_SET_NEXT_SETP_03
    LGOTO   CARRIER_SET_NEXT_SETP_04
    LGOTO   CARRIER_SET_NEXT_SETP_05
    LGOTO   CARRIER_SET_NEXT_SETP_06
    LGOTO   CARRIER_SET_NEXT_SETP_07

```

模式 1: 载波模式, 频率 2.405GHz;

模式 2: 载波模式, 频率 2.440GHz;

模式 3: 载波模式, 频率 2.480GHz;

模式 4：非载波模式，频率 2.405GHz；

模式 5：非载波模式，频率 2.440GHz；

模式 6：非载波模式，频率 2.480GHz；

模式 7：本振测试，频率 2.405GHz；

模式 8：本振测试，频率 2.440GHz；

模式 9：本振测试，频率 2.480GHz；

当前测试模式 LED 灯指示：

```
CARRIER_MODE_LED_SET:                ;测试模式闪灯
    INCR      R_LED_CNT, TO_R
    MOVR      R_LED_LOOP_TIME, TO_A
    BTRSC     STATUS, Z_B
    LGOTO     LED_NEXT_TIME_SET

    MOVIA     6
    SUBAR     R_LED_CNT, TO_A
    BTRSC     STATUS, C_B
    LGOTO     CARRIER_LED_ONE_00
    BSR      IO_LED
    RET

CARRIER_LED_ONE_00:
    MOVIA     10
    SUBAR     R_LED_CNT, TO_A
    BTRSC     STATUS, C_B
    LGOTO     CARRIER_LED_ONE_01
    BCR      IO_LED
    RET

CARRIER_LED_ONE_01:
    DECR     R_LED_LOOP_TIME, TO_R
    CLRR     R_LED_CNT
    RET

LED_NEXT_TIME_SET:
    MOVIA     100
    SUBAR     R_LED_CNT, TO_A
    BTRSS    STATUS, C_B
    RET
    CLRR     R_LED_CNT
    MOVR     R_SEND_CARRIER, TO_A
    MOVAR    R_LED_LOOP_TIME
    RET
```

注：测试可能会出现 RF 功率过大的情况，可以在 RF 初始化接口中调整测试时使用的通信功率，一般在 1M 通信速率下功率不能超过 0db，在 250K 速率下功率不能超过 6DB.

5. 发射模式建议采用 2 个按键同时按下作为触发条件，接收模式采用 1 个按键触发，避免误触发。

## XL2400T:

### 一、软件设计

过 FCC 需要修改的寄存器值：

寄存器名称	Bit 位	方向	默认值	修改值
ANALOGCFG0	<12>	R/W	0	1
ANALOGCFG1	<7: 6>	R/W	11	00
ANALOGCFG2	<12>	R/W	0	1

### 二、代码函数实现

```
void ModifyAnalogRegisters(void) {
```

```
    uint8_t gRfBuffer[ANALOG_REG_SIZE] = {0}; // 16 字节缓冲区，初始化为 0
```

```
    /* 1. 修改 ANALOG_CFG0 寄存器：位 12 设置为 1 */
```

```

RF_Read_Buff(ANALOG_CFG0, gRfBuffer, ANALOG_REG_SIZE); // 读取当前值

RF_DelayMs(1); // 延迟 1ms

// 位 12 位于字节 1（索引 1）的位 4（12/8=1 余 4）

gRfBuffer[1] = (gRfBuffer[1] | 0x10); // 使用按位或设置位 4 为 1（0x10 =
00010000）

RF_Write_Buff(W_REGISTER + ANALOG_CFG0, gRfBuffer, ANALOG_REG_SIZE);

// 写回寄存器

RF_DelayMs(1);

/* 2. 修改 ANALOG_CFG1 寄存器：位 7:6 设置为 00 */
RF_Read_Buff(ANALOG_CFG1, gRfBuffer, ANALOG_REG_SIZE); RF_DelayMs(1);

// 位 7:6 位于字节 0（索引 0）的高位：掩码 0xC0（11000000），清空后设置
为 00

gRfBuffer[0] = (gRfBuffer[0] & 0x3F); // 清空位 7:6（0x3F = 00111111，保留低
6 位）

// 由于目标值为 00，清空后无需额外设置（00 对应值 0）

RF_Write_Buff(W_REGISTER + ANALOG_CFG1, gRfBuffer, ANALOG_REG_SIZE);

RF_DelayMs(1);

/* 3. 修改 ANALOG_CFG2 寄存器：位 12 设置为 1 */
RF_Read_Buff(ANALOG_CFG2, gRfBuffer, ANALOG_REG_SIZE);

RF_DelayMs(1);

gRfBuffer[1] = (gRfBuffer[1] | 0x10); // 类似 ANALOG_CFG0，设置位 12（字节 1
的位 4）

```

```

RF_Write_Buff(W_REGISTER + ANALOG_CFG2, gRfBuffer, ANALOG_REG_SIZE);

RF_DelayMs(1);

}

```

**注：**将该函数放在初始化即可。

## XL2417D、XL2417U：

### 一、 软件设计

过 FCC 需要修改的寄存器值：

寄存器名称	Bit 位	方向	默认值	修改值	进制
RF_SETUP	<7>	R/W	0	1	二进制
RF_SETUP	<13: 8>	R/W	-	000000~111111	二进制
EN_AA	<7: 0>	R/W	3F	00	十六进制
SETUP_RETR	<7: 0>	R/W	33	00	十六进制
FEATURE	<7: 0>	R/W	-	00	十六进制
RF_CH	<7: 0>	R/W	-	00~53	十六进制

## 二、 代码函数实现

```
// 单载波输出配置函数
void RF_Enable_CarrierTest(void)
{
    unsigned char config_buff[4] = {0};

    // 1. 进入待机模式
    RF_CE_Low();
    DelayMs(1);

    // 2. 配置为单载波输出模式
    // 读取当前 RF_SETUP 寄存器配置
    Read_RF_Buff(R_REGISTER + RF_SETUP, config_buff, 2);

    // 设置单载波模式使能位
    config_buff[0] |= (1 << 7); // 设置 EN_CW 位为 1, 使能连续载波

    // 写入配置
    Write_RF_Buff(W_REGISTER + RF_SETUP, config_buff, 2);

    // 3. 禁用自动应答和重传
    SPI_Write_Reg(W_REGISTER + EN_AA, 0x00); // 禁用所有管道的自动应答
    SPI_Write_Reg(W_REGISTER + SETUP_RETR, 0x00); // 禁用自动重传

    // 4. 配置 FEATURE 寄存器, 禁用数据包处理功能
    SPI_Write_Reg(W_REGISTER + FEATURE, 0x00);

    // 5. 清空 FIFO
    SPI_Write_Reg(FLUSH_TX, CMD_NOP);
    SPI_Write_Reg(FLUSH_RX, CMD_NOP);

    printf("Carrier test mode enabled\n");
}

void RF_Disable_CarrierTest(void)
{
    unsigned char config_buff[4] = {0};

    // 1. 进入待机模式
    RF_CE_Low();
```

```

DelayMs(1);

// 2. 禁用单载波模式
Read_RF_Buff(R_REGISTER + RF_SETUP, config_buff, 2);
config_buff[0] &= ~(1 << 7); // 清除 EN_CW 位
Write_RF_Buff(W_REGISTER + RF_SETUP, config_buff, 2);

// 3. 恢复正常的通信配置
SPI_Write_Reg(W_REGISTER + EN_AA, 0x3F); // 使能所有管道的自动应答
SPI_Write_Reg(W_REGISTER + SETUP_RETR, 0x33); // 使能自动重传

printf("Carrier test mode disabled\n");
}

void RF_Set_CarrierFrequency(unsigned char channel)
{
    // 设置载波频率 (2400 + channel) MHz
    // XL2400T 支持 2400-2483MHz, channel 范围 0-83
    if(channel > 83) channel = 83;

    SPI_Write_Reg(W_REGISTER + RF_CH, channel);

    printf("Carrier frequency set to %d MHz\n", 2400 + channel);
}

void RF_Set_CarrierPower(unsigned char power)
{
    unsigned char power_buff[3] = {0};

    // 读取当前 RF_SETUP 寄存器配置
    Read_RF_Buff(R_REGISTER + RF_SETUP, power_buff, 2);

    // 清除功率设置位 (bit13-8)
    power_buff[1] &= 0xC0; // 保留高位, 清除功率位

    // 设置新的功率值
    // 功率配置在 RF_SETUP 寄存器的 bit13-8
    power_buff[1] |= (power & 0x3F); // 确保功率值在有效范围内

    Write_RF_Buff(W_REGISTER + RF_SETUP, power_buff, 2);

    printf("Carrier power configured\n");
}

```

```
// 完整的单载波测试初始化函数
void RF_CarrierTest_Init(unsigned char channel, unsigned char power)
{
    // 芯片复位
    RF_Reset();
    DelayMs(10);

    // 基本射频初始化
    RF_Init();
    DelayMs(5);

    // 设置载波频率
    RF_Set_CarrierFrequency(channel);

    // 设置发射功率
    RF_Set_CarrierPower(power);

    // 使能单载波输出
    RF_Enable_CarrierTest();

    // 进入发射模式
    RF_Tx_Mode();

    printf("Carrier test initialization complete\n");
    printf("Frequency: %d MHz, Power level: 0x%02X\n", 2400 + channel, power);
}
```