XL2400P 的传输模式：

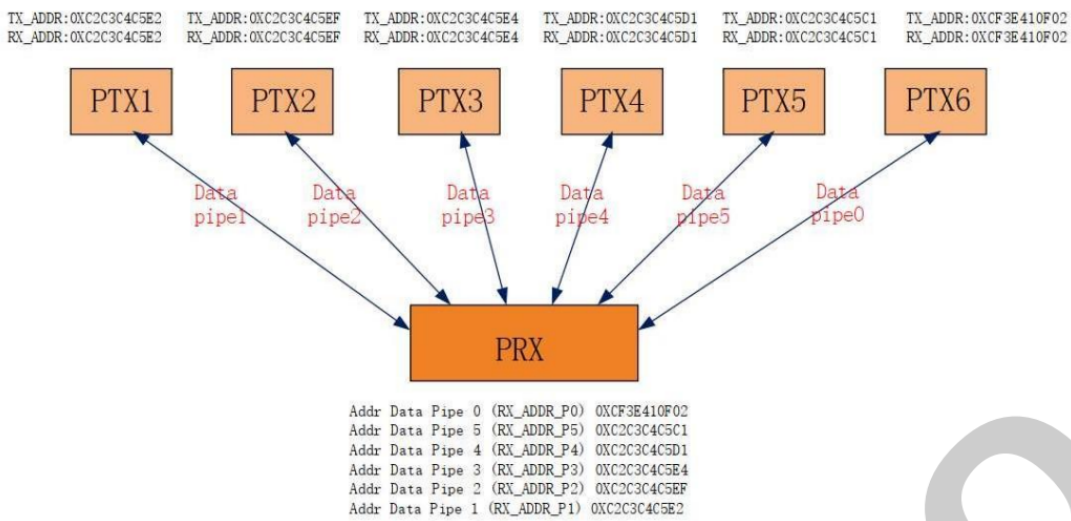
	数据模式	应答	多个发送 对 一个接收	一个发送 对 多个接收
模式1	固定	不带	支持	支持
模式2	动态	不带	支持	支持
模式3	固定	带	支持	支持
模式4	动态	带	支持	支持

1.多个发送对一个接收的模式说明

硬件最多支持6个数据通道

数据通道0的地址可以与其他5个 完全不一样

数据通道1-5的地址有要求 高4位必须保持一致 低1位可以不一样



多通道数据传输应答地址示例

发送端地址示例：

```
//发送端 0 地址
unsigned char RF_Test_Adrress[5]={0x22,0x33,0x44,0x55,0x66};//RF地址
```

```
//发送端 1地址
unsigned char RF_Test_Adrress[5]={0x11,0x11,0x11,0x11,0x11};//RF地址
```

```
//发送端 2 地址
unsigned char RF_Test_Adrress[5]={0x12,0x11,0x11,0x11,0x11};//RF地址
```

接收端地址示例：

```
//接收端地址示例

unsigned char RF_Test_Adrress0[5]={0x22,0x33,0x44,0x55,0x66};//RF数据通道0地址

unsigned char RF_Test_Adrress1[5]={0x11,0x11,0x11,0x11,0x11};//RF数据通道1地址

unsigned char RF_Test_Adrress2_5[4]={0x12,0x13,0x14,0x15};//RF数据通道2-5地址

//设置地址0
void RF_Set_Address0(unsigned char *AddrBuff)
{
    Write_RF_Buff(W_REGISTER+TX_ADDR,AddrBuff , 5);
    Write_RF_Buff(W_REGISTER+RX_ADDR_P0, AddrBuff ,5);
}

//设置地址1
void RF_Set_Address1(unsigned char *AddrBuff)
{
    Write_RF_Buff(W_REGISTER+TX_ADDR,AddrBuff , 5);
    Write_RF_Buff(W_REGISTER+RX_ADDR_P1, AddrBuff ,5);
}

```

```
//设置地址2-5
void RF_Set_Address2_5(unsigned char *AddrBuff)
{
    Write_RF_Buff(W_REGISTER+RX_ADDR_P2TOP5, AddrBuff ,4);
}
```

0A	RX_ADDR_P0	39:0	0xE7 E7E7 E7E7	R/W	RX 地址数据管道 0。最大 5 个字节。首先写入 LSB 字节。 SETUP_AW设置的使用的字节数。
0B	RX_ADDR_P1	39:0	0xC2 C2C2 C2C2	R/W	RX 地址数据管道 1。最大 6 个字节。首先写入 LSB 字节。 SETUP_AW设置的使用的字节数。
0C	RX_ADDR_P2TOP5				仅设置 LSB，MSB 字节使用 RX_ADDR_P1[39：8]
	RX_ADDR_P5	31:24	0xc6	R/W	RX地址数据管道5
	RX_ADDR_P4	23:16	0xc5	R/W	RX地址数据管道4
	RX_ADDR_P3	15:8	0xc4	R/W	RX地址数据管道3
	RX_ADDR_P2	7:0	0xc3	R/W	RX地址数据管道2

设置地址0是写入0A寄存器 设置地址1 是写入0B寄存器 设置地址2-5是写0C寄存器

2.固定包长与动态包长

需要配置两个寄存器 1C 1D

1C寄存器是配置需要使用 动态长度的数据通道

1C	DYNPD				动态有效载荷长度
	Reserved	7:6	00	R/W	Unused
	DPL_P5	5	0	R/W	设置 1 以启用动态有效负载 长度数据管道 5（需要 EN_DPL & ENAA_P5)
	DPL_P4	4	0	R/W	设置 1 以启用动态有效负载 长度数据管道 4（需要 EN_DPL & ENAA_P4)
	DPL_P3	3	0	R/W	设置 1 以启用动态有效负载 长度数据管道 3（需要 EN_DPL & ENAA_P3)
	DPL_P2	2	0	R/W	设置 1 以启用动态有效负载 长度数据管道 2（需要 EN_DPL & ENAA_P2)
	DPL_P1	1	0	R/W	设置 1 以启用动态有效负载 长度数据管道 1（需要 EN_DPL & ENAA_P1)
	DPL_P0	0	0	R/W	设置 1 以启用动态有效负载 长度数据管道 0（需要 EN_DPL & ENAA_P0)

1D寄存器是使能动态长度 第5位是使能长包功能 最大128字节 不使能第5位默认最大32字节

1D	FEATURE				特征
	STAT_SETUP [1:0]	7:6	00	R/W	在命令输入期间调整SDO的输出 00：默认值，SDO输出为状态 01：RX读出模式，SDO输出MAX_RT和TX_FULL位被RSSI1和RSSI2读出取代 10：FIFO读出模式，SDO输出STATUS_FIFO 11：未使用，与 00 相同
	EN_LONG_PLD	5	0	R/W	写 1 启用长有效负载功能，最大长度为 128 字节 可选
	EN_FEC	4	1	R/W	写1 启用 FEC（加绕功能）
	EN_WHITEN	3	1	R/W	写1 启用白化 功能
	EN_DPL	2	1	R/W	写1启用动态有效负载长度
	EN_ACK_PAY	1	0	R/W	写1在 ACK 上启用有效负载
	EN_DYN_ACK	0	0	R/W	写1启用 W_TX_PAYLOAD_NOACK 命令

3.ACK应答配置

01寄存器用于配置应答

01	EN_AA				自动应答设置
	REG_LOCK	47:16	0	R/W	位 16： Reg00 锁定位（0：解锁， 1：锁定） 位 17： Reg01 锁定位（0：解锁， 1：锁定） 位18： Reg02 锁定位（0：解锁， 1：锁定） 位 19： Reg03 锁定位（0：解锁， 1：锁定） ..... bit47： reg1f 锁定位（0：解锁， 1：锁定）
	REG_LOCK_KEY	15:8	0	W	写这个寄存器0x5C，REG_LOCK可以设置，读这个寄存器会返回0x00
	Reserved	7	0	R/W	Unused
	TO_RF_PULSE_SPI	6	0		to RF module
	ENAA_P5	5	1	R/W	在数据管道 5 上启用应答
	ENAA_P4	4	1	R/W	在数据管道 4 上启用应答
	ENAA_P3	3	1	R/W	在数据管道 3 上启用应答
	ENAA_P2	2	1	R/W	在数据管道 2 上启用应答
	ENAA_P1	1	1	R/W	在数据管道 1 上启用应答
	ENAA_P0	0	1	R/W	在数据管道 0 上启用应答

```c
void RF_Init(void)
{
    unsigned char RF_Init_Buff[16]={0};

    SPI_Write_Reg(W_REGISTER+CFG_TOP,0x02);
    DelayMs(2);

    SPI_Write_Reg(W_REGISTER+CFG_TOP,0x3E);
    DelayMs(2);

    /* 配置模拟寄存器 */
    Read_RF_Buff(ANALOG_CFG3 , RF_Init_Buff , 6);
    RF_Init_Buff[5] =((RF_Init_Buff[5]&0xff) | 0x6d);
    Write_RF_Buff(W_REGISTER+ANALOG_CFG3 , RF_Init_Buff , 6);

    /* 配置应答数据通道道 */
    SPI_Write_Reg(W_REGISTER + EN_AA,     0x00);//不使用应答

    /* 配置使能地址 */
    SPI_Write_Reg(W_REGISTER + EN_RXADDR, 0x3f);//使能数据通道0-5

    /* 配置地址长度 */
    SPI_Write_Reg(W_REGISTER + SETUP_AW,  0xaf);//配置地址长度为5字节

    /* 配置重传次数和时间间隔 */
    SPI_Write_Reg(W_REGISTER + SETUP_RETR,0x33);//

    /* 配置通讯速率 */
    SPI_Write_Reg(W_REGISTER + RF_SETUP, C_DR_250K);//配置通讯速率为250Kpbs

    /* 配置数据通道0&数据通道1 接收包长度 */
    RF_Init_Buff[0] = RF_PACKET_SIZE;
    RF_Init_Buff[1] = RF_PACKET_SIZE;
    Write_RF_Buff(W_REGISTER+RX_PW_PX, RF_Init_Buff ,2);

    /* 配置PIPE动态长度使能位 */
    SPI_Write_Reg(W_REGISTER+DYNPD, 0x00);//不使能
```

在我们提供的驱动基础上 将EN_AA (01寄存器) 如果配置为 0x3F 就是在所有数据通道上启用应答

0x3F对应 二进制 0011 1111

```c
//设置频点
void RF_Set_Chn(unsigned char Chn)
{
    SPI_Write_Reg(W_REGISTER + EN_AA, 0x00);
    SPI_Write_Reg(W_REGISTER + RF_CH, Chn + 0x60);
    SPI_Write_Reg(W_REGISTER + EN_AA, 0x40);
}
```

然后设置频点的函数 也需要更改 上面的改成和初始化一样的值 下面的 再原有的值上第7位置1

0x3F 0011 1111

0x7F 0111 1111

07	STATUS				状态（从SDO引脚读出,在SPI命令字输入期间）； SDO输出可以调整
	Reserved	7	0	R/W	保留，不使用
	RX_DR	6	0	R/W	数据RX FIFO中断，在新数据进入FIFO时候触发，写入 1 以清除位
	TX_DS	5	0	R/W	数据发送 TX FIFO 中断。在传输数据包完成时触发。 如果激活了自动 ACK，则仅当ACK收到时候触发。写入 1 以清除位
	MAX_RT	4	0	R/W	TX 重传的最大次数中断。写入 1 以清除位。如果 MAX_RT触发，它必须清除以启用进一步操作
	RX_P_NO[2:0]	3:1	111	R	可用于从rx_fifo读取的有效载荷的数据管道编号 000~101：数据管道数（0~5）
	TX_FULL	0	0	R	0: TX FIFO 可用 1: TX FIFO 满

启用应答机制以后 发送数据出去以后 状态寄存器 会变化 只有当接收端 回传应答 以后 第5位才会触发

4.操作指令说明

```
/********************** 操作指令 **********
#define R_REG                    0x00
#define R_REGISTER               0x00//读寄存器指
#define W_REG                    0x20//写寄存器指
#define W_REGISTER               0x20//写寄存器指
#define R_RX_PLOAD           0x61//读接收数据，读
#define W_TX_PLOAD           0xA0//写发射数据，写
#define FLUSH_TX             0xE1//清 TX FIFO
#define FLUSH_RX             0xE2//清 RX FIFO
#define REUSE_TX_PL          0xE3//用在 PTX 端，

#define R_RX_PL_WID          0x60//读 RX FIFO 最
#define W_ACK_PLOAD          0xA8//
#define W_TX_PLOAD_NOACK     0xB0//
#define CMD_NOP              0xFF//空操作。可用来

/* STATUS Interrupt status */
#define RX_DR            (0x40)//接收到数据中断标
#define TX_DS            (0x20)//发送数据完成中断
#define MAX_RT           (0x10)//达到最大发送次数
```

指令	命令字 二进制	后带数据	操作
R_REGISTER	000A AAAA	1到5 低字节在前	读寄存器 AAAAA=5位表示寄存器地址
W_REGISTER	001A AAAA	1到5 低字节在前	写寄存器 AAAAA=5位表示寄存器地址
R_RX_PAYLOAD	0110 0001	1 to 32/64低字节在前	读接收数据，读操作通常由第 0 字节开始，读完过后数据将从 RX FIFO 中删除，接收模式下执行
W_TX_PAYLOAD	1010 0000	1 to 32/64低字节在前	写发射数据，写操作通常由 0 字节开始。
FLUSH_TX	1110 0001		清 TX FIFO
FLUSH_RX	1110 0010		清 RX FIFO
R_RX_PL_WID	0110 0000		读 RX FIFO 最顶部 RX-payload 数据宽度

NOP	1111 1111		无操作

5.对码思路

所谓的对码 就是 改变RF的地址 让地址不一致的其他端 接收不到数据 核心就是设置RF地址

将RF地址 存在MCU的Flash里面 上电初始化的时候 将它读取出来 然后设置为RF地址