XL2400P 一对多 多对一使用笔记

说明：2.4G 通信收发双方需要地址、频点、数据宽度和管道一致，这也是实现一对多和多对一的关键。

1. 一个发送对多个接收

一个发送对多个接收，可以改变地址和频点两种方式实现，也可以是不同的频点+不同的地址实现。

硬件最多支持 6 个数据通道，通道 0~通道 5，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，第 5 位可以变化。

	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Data pipe 0(RX_ADDR_P0)	0xF1	0xD2	0xE6	0xA2	0x33
Data pipe 1(RX_ADDR_P1)	0xD3	0xD3	0xD3	0xD3	0xD3
Data pipe 2(RX_ADDR_P2)	0xD3	0xD3	0xD3	0xD3	0xD4
Data pipe 3(RX_ADDR_P3)	0xD3	0xD3	0xD3	0xD3	0xD5
Data pipe 4(RX_ADDR_P4)	0xD3	0xD3	0xD3	0xD3	0xD6
Data pipe 5(RX_ADDR_P5)	0xD3	0xD3	0xD3	0xD3	0xD7

从表中可以看 数据通道 0 的 5byte 总共 40 位的地址都是可配的；数据通道 1~5 的地址配置为 32 位共用地址（不数据通道 1 共用）+8 位各自的地址（最低字节）。

所以为了实现更多的通信，选择数据通道 0 来进行通信，这样地址的选择才最多。

```
109 //设置地址
110 void RF_Set_Address(unsigned char *AddrBuff)
111 {
112     Write_RF_Buff(W_REGISTER+TX_ADDR,AddrBuff , 5);
113     Write_RF_Buff(W_REGISTER+RX_ADDR_P0, AddrBuff ,5);
114 }
```

也可以是改变频点来实现一对多，也就是每个接收分配一个频点。

```
//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);
}
```

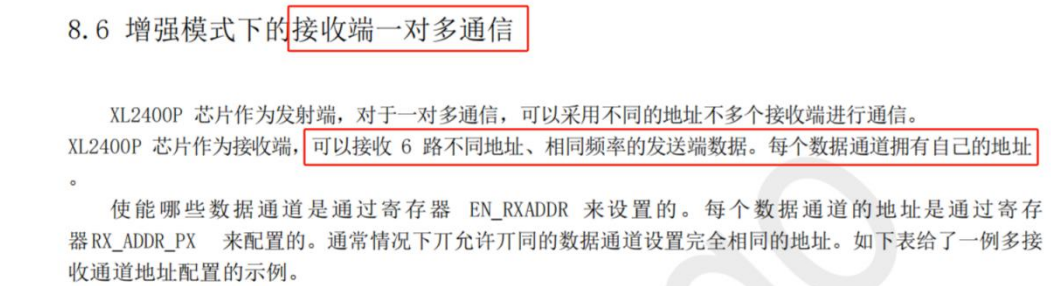
严格意义上说并不是一对多，因为发送的数据不是在同一时刻发出，而是发送完一次数据后，马上改变地址或者频点，发送给下一个接收，依次发送给所有的接收，因为发一次数据时间为毫秒级，在一定程度上可以理解为实现一对多发送数据。而且这种方式实现的一对多通信是可控的，发送端可以任意改变发送的数据和选择接收的对象。

如果接收不需要可控，也就是发送方发送一次数据，所有的接收都收到一次一样的的数据，那就可以将所有接收配置为一模一样的参数，也就是所有接收烧录同一份普通接收程序，这样实现的就是严格意义上的一对多，发射一次，所有的接收都能收到。

2. 多个发送对一个接收

配置多对一，只需要改变数据通道和地址的配置即可。

硬件最多支持 6 个数据通道，通道 0~通道 5，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，第 5 位可以变化。



8.6 增强模式下的接收端一对多通信

XL2400P 芯片作为发射端，对于一对多通信，可以采用不同的地址不多个接收端进行通信。XL2400P 芯片作为接收端，可以接收 6 路不同地址、相同频率的发送端数据。每个数据通道拥有自己的地址。

使能哪些数据通道是通过寄存器 EN_RXADDR 来设置的。每个数据通道的地址是通过寄存器RX_ADDR_PX 来配置的。通常情况下开允许开同的数据通道设置完全相同的地址。如下表给了一例多接收通道地址配置的示例。

发送端地址示例：

发送端 1（使用数据通道 0）地址

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

发送端 2（使用数据通道 1）地址

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

发送端 3（使用数据通道 2）地址

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

2 和 3 的地址高 4 位须保持一致。

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
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
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
void RF_Set_Address2_5(unsigned char *AddrBuff)
{
    Write_RF_Buff(W_REGISTER+RX_ADDR_P2TOP5, AddrBuff ,4);
}
```

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。

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