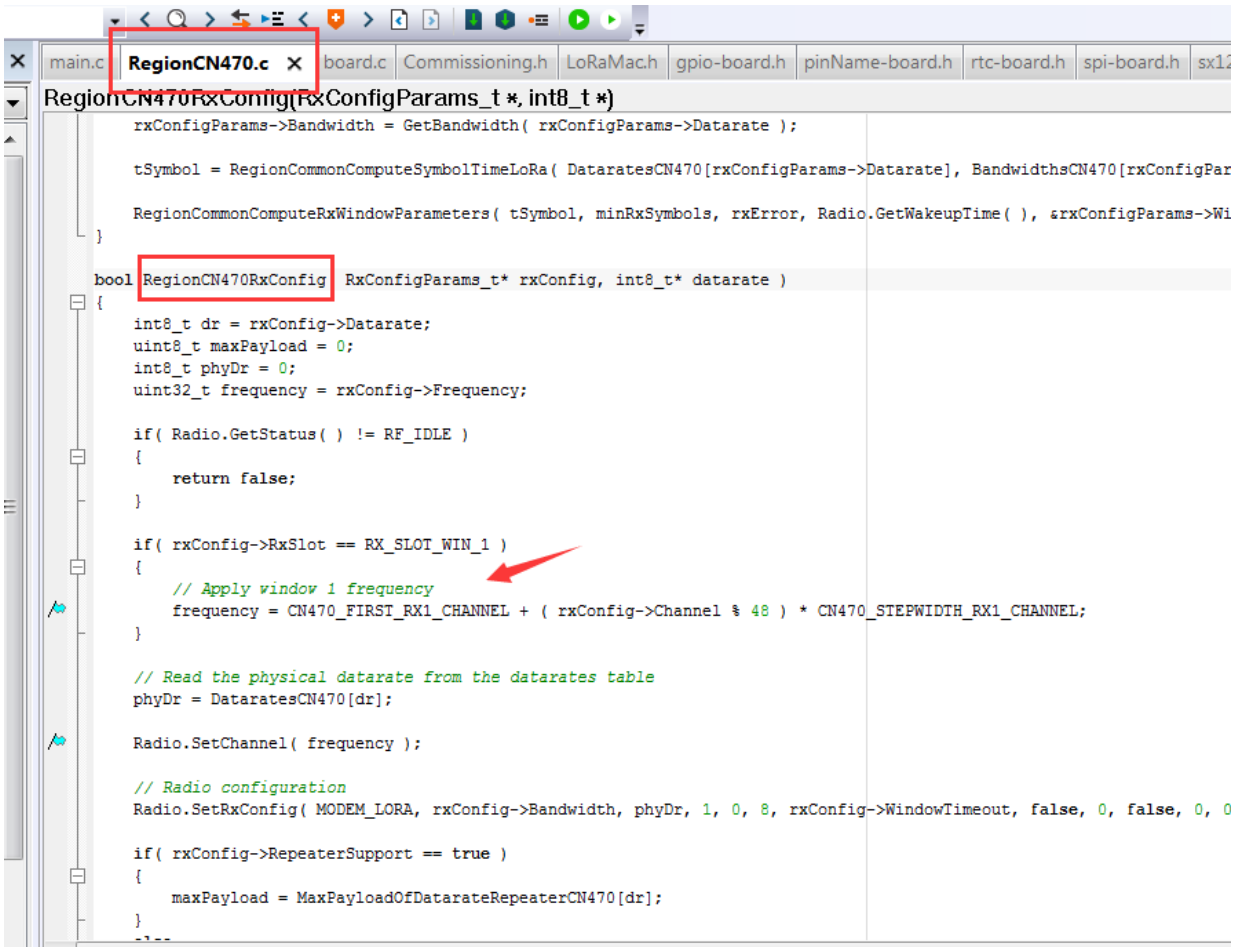
1. Semtech 提供的网关内置 ns,只支持同频收发，标准 CN470 规范是异频，所以使用网关内置 server 测试时，终端需要修改 rxwindow1,

配置函数在 RegionCN470.c->RegionCN470RxConfig

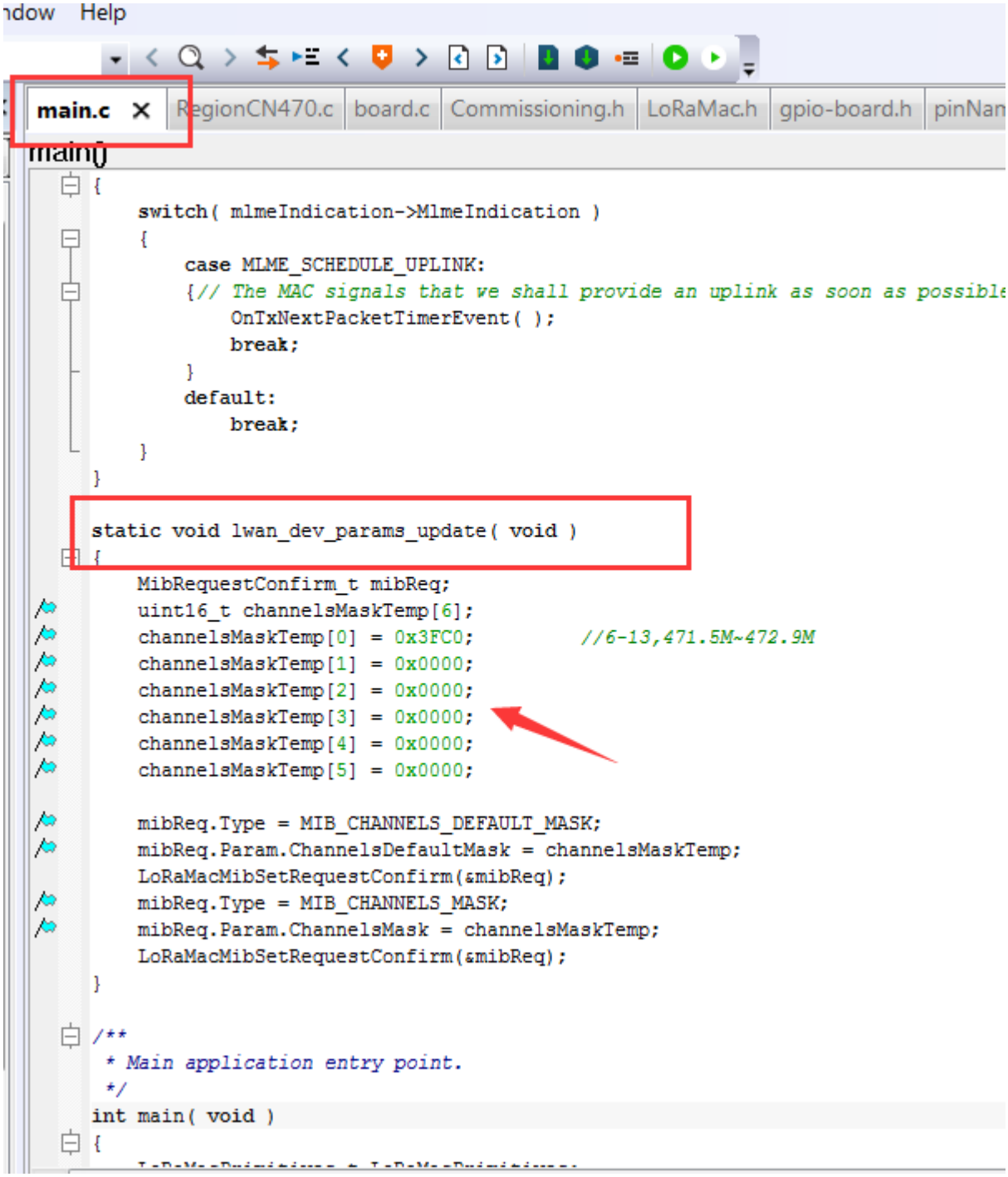


frequency=470300000+rxConfig->Channel*200000; //与发射频率相同

```
bool RegionCN470RxConfig( RxConfigParams_t* rxConfig, int8_t* datarate )
{
    int8_t dr = rxConfig->Datarate;
    uint8_t maxPayload = 0;
    int8_t phyDr = 0;
    uint32_t frequency = rxConfig->Frequency;

    if( Radio.GetStatus( ) != RF_IDLE )
    {
        return false;
    }

    if( rxConfig->RxSlot == RX_SLOT_WIN_1 )
    {
        // Apply window 1 frequency
//        frequency = CN470_FIRST_RX1_CHANNEL + ( rxConfig->Channel % 48 ) * CN470_STEPWIDTH_RX1_CHANNEL;   //标准CN470,异频
        frequency=470300000+rxConfig->Channel*200000;          //接收下发频率相同，同频
    }

    // Read the physical datarate from the datarates table
    phyDr = DataratesCN470[dr];
```

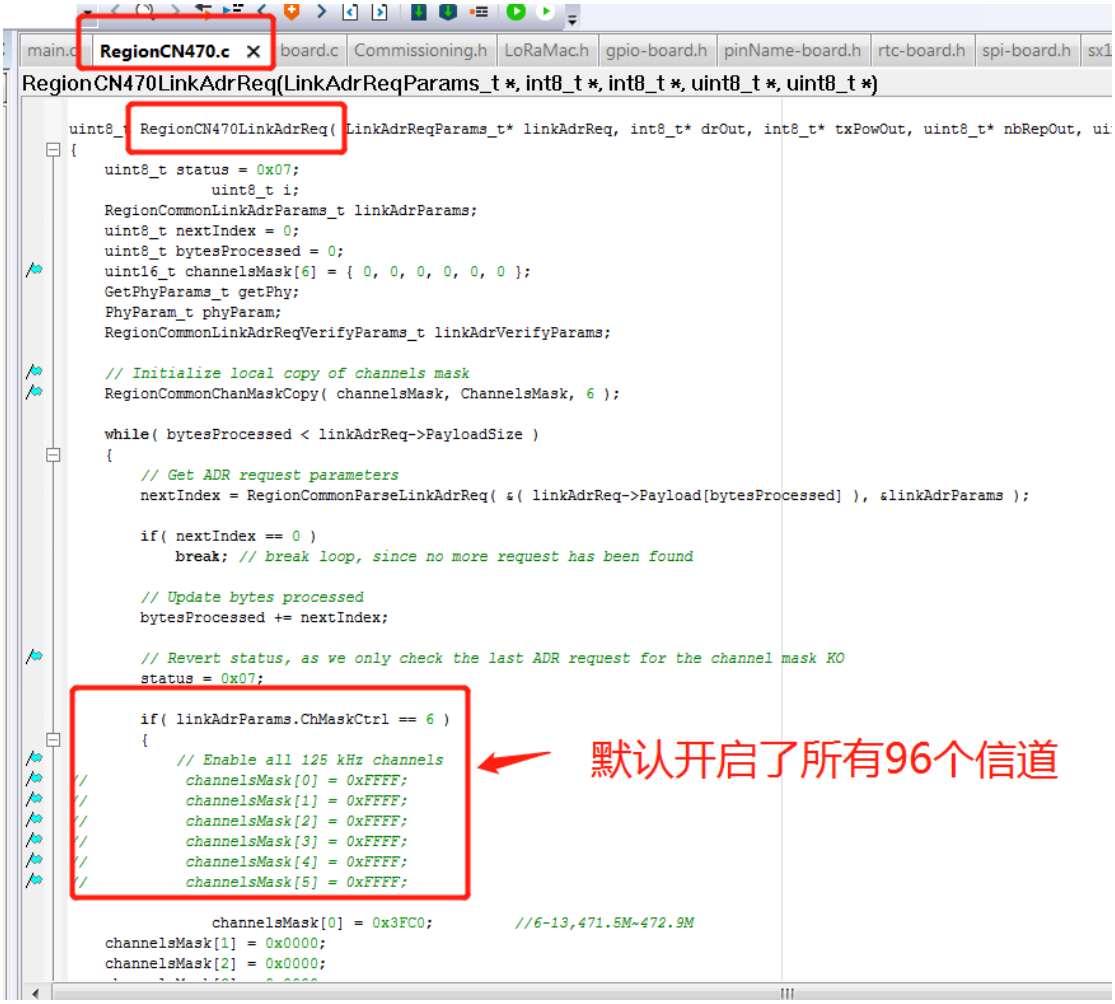
2. ADR 开启后，发射频率设置后，接收到数据频率又改变

正常我们配置节点发送频率的位置 main.c ->lawn_dev_param_update()



上行 96 个信道，channelsMaskTemp[]来配置，1 对应开发开启，0 关闭；

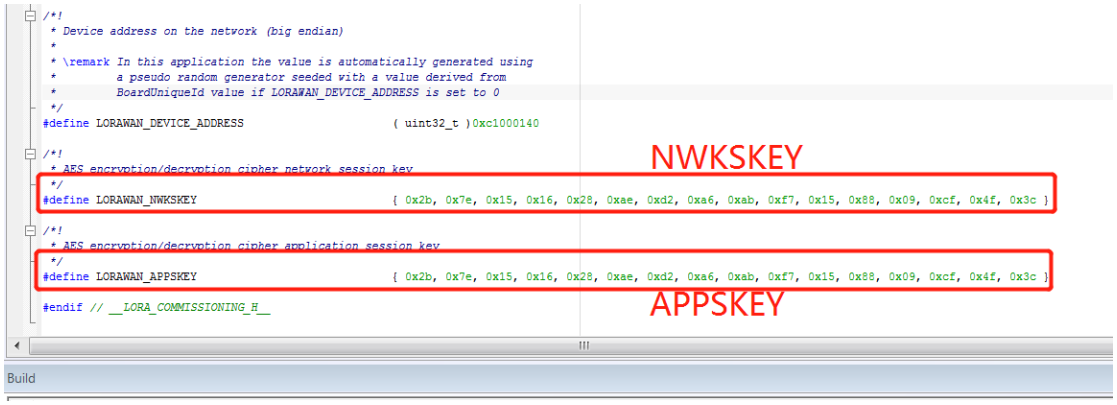
还需要修改一个地方，RegionCN470.c->RegionCN470LinkAdrReq()


main.c
RegionCN470.c
board.c
Commissioning.h
LoRaMac.h
gpio-board.h
pinName-board.h
rtc-board.h
spi-board.h
RegionCN470LinkAdrReq(LinkAdrReqParams_t *, int8_t *, int8_t *, uint8_t *, uint8_t *)
uint8_t RegionCN470LinkAdrReq(LinkAdrReqParams_t* linkAdrReq, int8_t* drOut, int8_t* txPowOut, uint8_t* nbRepOut, ui
{
uint8_t status = 0x07;
uint8_t i;
RegionCommonLinkAdrParams_t linkAdrParams;
uint8_t nextIndex = 0;
uint8_t bytesProcessed = 0;
uint16_t channelsMask[6] = { 0, 0, 0, 0, 0, 0 };
GetPhyParams_t getPhy;
PhyParam_t phyParam;
RegionCommonLinkAdrReqVerifyParams_t linkAdrVerifyParams;
// Initialize local copy of channels mask
RegionCommonChanMaskCopy(channelsMask, ChannelsMask, 6);
while(bytesProcessed < linkAdrReq->PayloadSize)
{
// Get ADR request parameters
nextIndex = RegionCommonParseLinkAdrReq(&(linkAdrReq->Payload[bytesProcessed]), &linkAdrParams);
if(nextIndex == 0)
break; // break loop, since no more request has been found
// Update bytes processed
bytesProcessed += nextIndex;
// Revert status, as we only check the last ADR request for the channel mask KO
status = 0x07;
if(linkAdrParams.ChMaskCtrl == 6)
{
// Enable all 125 kHz channels
// channelsMask[0] = 0xFFFF;
// channelsMask[1] = 0xFFFF;
// channelsMask[2] = 0xFFFF;
// channelsMask[3] = 0xFFFF;
// channelsMask[4] = 0xFFFF;
// channelsMask[5] = 0xFFFF;
channelsMask[0] = 0x3FC0; //6-13,471.5M~472.9M
channelsMask[1] = 0x0000;
channelsMask[2] = 0x0000;
默认开启了所有96个信道

需要修改成：


main.c
RegionCN470.c
board.c
Commissioning.h
LoRaMac.h
gpio-board.h
pinName-board.h
RegionCN470LinkAdrReq(LinkAdrReqParams_t *, int8_t *, int8_t *, uint8_t *, uint8_t *)
// Revert status, as we only check the last ADR request for the channel mask KO
status = 0x07;
if(linkAdrParams.ChMaskCtrl == 6)
{
// Enable all 125 kHz channels
// channelsMask[0] = 0xFFFF;
// channelsMask[1] = 0xFFFF;
// channelsMask[2] = 0xFFFF;
// channelsMask[3] = 0xFFFF;
// channelsMask[4] = 0xFFFF;
// channelsMask[5] = 0xFFFF;
channelsMask[0] = 0x3FC0; //6-13,471.5M~472.9M
channelsMask[1] = 0x0000;
channelsMask[2] = 0x0000;
channelsMask[3] = 0x0000;
channelsMask[4] = 0x0000;
channelsMask[5] = 0x0000;
}
else if(linkAdrParams.ChMaskCtrl == 7)
{
status &= 0xFE; // Channel mask KO
}
else
{

3. 网关入网的相关参数位置 DEVEUI,DEVADDR,APPSKEY,NEWSKEY,ABP/OTAA,APPEUI,APPKEY
主要在 Commissioning.h 文件，



4. LoRaWAN 发送周期设置 main.c->main()


main.c
RegionCN470.c
board.c
Commissioning.h
LoRaMac.h
gpio-board.h
pinName-board.h
rtc-board.h
spi-board.h
sx126x-board.h
main()
mibReq.Type = MIB_NETWORK_JOINED;
mibReq.Param.IsNetworkJoined = true;
LoRaMacMibSetRequestConfirm(&mibReq);
DeviceState = DEVICE_STATE_SEND;
#endif
break;
}
case DEVICE_STATE_SEND:
{
if(NextTx == true)
{
/* LED TX ON */
GPIO_LOW(LED_TX_PORT,LED_TX_PIN);
PrepareTxFrame(AppPort);
NextTx = SendFrame( );
/* LED TX OFF */
GPIO_HIGH(LED_TX_PORT,LED_TX_PIN);
}
// Schedule next packet transmission
TxDutyCycleTime = APP_TX_DUTYCYCLE + randr(0, APP_TX_DUTYCYCLE_RND);
break;
}
case DEVICE_STATE_CYCLE:
{
DeviceState = DEVICE_STATE_SLEEP;
// Schedule next packet transmission
TimerSetValue(&TxNextPacketTimer, TxDutyCycleTime);
TimerStart(&TxNextPacketTimer);
break;
}
case DEVICE_STATE_SLEEP:
ms 发送周期=固定+随机延时
Build