MSP430F5438A&SI4463通用板调试手册

2014-08-19

1：对于红灯、绿灯、蜂鸣器、三路外部电源（两路AAT3221，一路AAT4280）增加了管脚功能的选择，分别在驱动代码中PINSEL做了设置，都分别设置成为了一般的IO管脚

2：增加UA0 RS232/TTL转换芯片打开与关闭驱动代码

3：增加UA3 RS232/TTL转换芯片打开与关闭驱动代码

2014-08-20

1:完成射频模块管脚宏定义

2014-08-21

1：完成射频模块的物理层部分的驱动代码

2：修改了使RF进入到不同状态的驱动代码，从html版数据手册REQUEST\_DEVICE\_STATE命令中可以得到使用该命令的时候不可能从该命令的返回数据中得到SLEEP或者STANDBY模式，因为这两种状态下使用SPI命令的时候，首先就要先确定RF工作在SPI\_ACTIVE模式下

2014-08-22：

1：修改StartTX函数，在CHANGE\_STATE命令中，有CHANGE\_TX命令，但是无法写入CHANNEL NUMBER等参数，所以在SetRFTx()函数中，不适用CHANGE\_STATE命令，修改成为START\_TX命令，并可以一次性写入频道号，以及转换状态。

2：要注意SI4463和SI4432工作区别，SI4432可以通过命令直接跳频到某一个频点，但是SI44463的改变频道号在START\_TX中，所以必须要该命令中才能改变工作频点。

\*\*注意\*\*在RF驱动代码中，写了连个发送函数，待回校后进行测试，确定函数的使用是否正确。

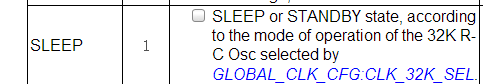
3:在使用SetRFStartTx()函数中，开始发送的是载波模式，防止未完成前面的写入工作就进行发送，发送发送错误（待测试），在函数中还没有添加调制模式的代码，等待后续代码的添加

4：StartRFRx()函数中改变形参

2014-08-23

1：查看数据手册START\_RX命令，当RX\_LEN[15：0]的值不为0的时候，不使用packet handler,当参数被配置成为0的时候，使用packet handler ,对应2014-08-22的问题4，改变形参只有工作频道号

2：改变RF的状态，CHANGE\_STATE命令中，通过查看进入到SLEEP模式下



当使能CLK\_32K\_SEL时进入的是SLEEP模式下

未使能CLK\_32K\_SEL时进入的是STANDBY模式，

通过参数的配置，可以判断使RF进入到不同状态的函数方法应该如何书写，注意参数的不同

2014-9-27

1:完成了大数据的发送与接收的调试

遇到的问题：前面几天遇到的问题是大数据的接收不正确，读取到很多重复的数据

修改方法：默认当接收数据的时候，设计PKT\_FIELD\_1\_LENGTH\_0\_7的值设置成为255，也就是该值的最大设置 PKT\_FIELD\_1\_LENGTH\_7\_0(255);

注意：该寄存器的设计不可以在正接受数据的过程中进行设置，这样会使接收数据不正确，具体原因正在进一步查找

后续修改方法：参考AN626（PACKET HANDLER）手册的方法进行修改

2014-09-28：

在完成上次的测试之后，发现一个问题，当发送数据的长度不确定，接收数据的字段长度一直设置成为255之后，测试发现当接收几次数据之后，便不能够正常的接受数据，正在调试中

测试发现：如果设置的字段长度与发送长度是一致的时候，此时可以正确的接受数据，如果不是同一个长度，则接受过程中就会很容易出现错误。

调试解决方法：设置variable packet length，第一个字段设置成为长度字段，第二个字段为数据字段，

设置方法：在TX过程中，FIELD\_2的发送长度要根据数据段的实际长度进行改变，在RX接受过程中,FIELD\_2字段设置的值为接收的最大长度，因为程序中设置的最大接收长度为255字节，所以在接受过程中该字段的值设置成为255；

2014-09-29

问题介绍：

使用CHANGE\_STATE命令进入到SLEEP模式下，但是没有给出如何使用CHANGE\_STATE命令进入到Standby模式下，所以在射频层驱动中，在使用进入到Standby和Sleep时，均进入到了sleep模式下

2014-11-22

问题介绍：

使用GetWirelessPacket();函数无法正确的接收到无线数据。

原因分析：

int GetWirelessPacket(unsigned int overtime)

{

int ret;//返回值

unsigned char maxRSSI = 0;

unsigned char RSSI = 0;

SetRFRx(); //====在这里执行了清除FIFO的指令

//CLR\_NIRQ\_IFG;//清除中断标记

EINT\_NIRQ;

SetTimer0A0\_ACLK();

Timer0A0\_Delay(PHY\_ACLK\_1MS); //搜寻60ms

while((g\_iWirelessRxFlag == 1)&&(g\_uPHYTimer0A0Count<overtime)) //限定时间内等待无线数据

{

RSSI = GetRSSI(); //获取最大的RSSI

// RSSI=GetFastLatchRSSI();

if(RSSI >maxRSSI)

maxRSSI = RSSI;

}

StopTimer0A0();

DINT\_NIRQ;

g\_uchRSSI = maxRSSI;

ret = g\_iWirelessRxFlag; //先将返回值返回的然后在将标志位改变为默认值的

g\_iWirelessRxFlag = 1;

//if(ret == 0)//做CRC校验

//{

// if( GenerateCRC(g\_uchWirelessRxBuf,g\_uchWirelessRxBuf[0])!=0 ) //CRC校验错误

// return -3;

//}

return ret;

}

在上述函数中，在等待过程中会去获取RSSI的值，在SI4463中读取RSSI值的时候，占用了太多的时钟，导致无线数据无法正确的读出。

可以调用快速读取寄存的函数读取RSSI,

**二：重点注意事项：**

程序错误状态：在程序的最初测试中，对数据包字段进行正常的设置后，对于所发送的数据帧的设置为前四个字节为工程字，后面的字节为数据帧，且数据帧的第一个字节为数据的长度，但是在测试过程中，发现数据总是无法正常的接受。

修改方式：1：**在进行发送数据的时候，不要一次性写满FIFO**，很有可能会出现溢出，导致数据发送的不正确 2：参考AN626数据手册，在对**数据帧格式的设置有一定的要求**，到数据包格式设置为不固定数据格式的时候，先接收到的数据为发送的总数据帧的长度。

**例程：**

PKT\_LEN(0x2a); //长度字节配置成为一个字节

//这里设置为表示数据帧长度的字节为一个字节，数据长度字节加入到FIFO中，详细的介绍参考介绍SPI的数据手册。

//数据帧的格式

TXBuffer[0]=length; //第一个字节要存储数据帧的总长度

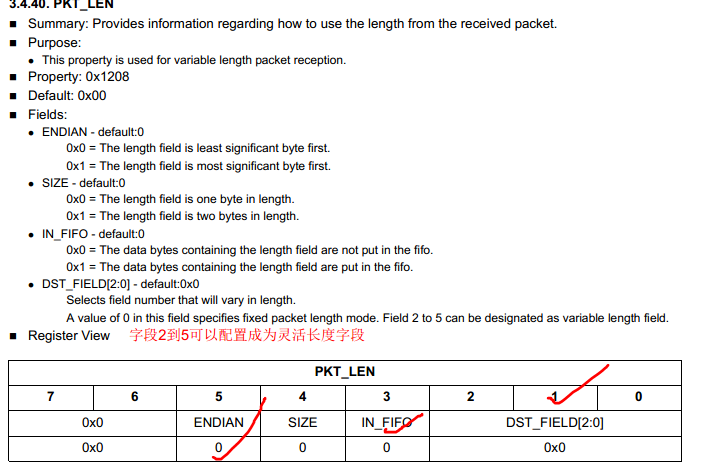
//这里与SI4432所开发的驱动层程序有一点不同

TXBuffer[1] = g\_PHYDevInfo.IntegratorID.ucIntegratorID[0]; //集成商ID

TXBuffer[2] = g\_PHYDevInfo.IntegratorID.ucIntegratorID[1];

TXBuffer[3] = g\_PHYDevInfo.ProjectID.ucProjectID[0]; //设备工程ID

TXBuffer[4] = g\_PHYDevInfo.ProjectID.ucProjectID[1];

这里当SIZE字段的值改变时，数据帧格式要跟着改变，当SIZE=1时，数据帧的前两个字节要用来表示数据帧长度，如数据帧长度为20字节时，TXBuffer[0]=0x00,

TXBuffer[1]=20,后面的数据帧数组下标都要加1。

**未完成的事情：**

在发送与接收函数中针对定义两种不同长度的SIZE所对应的代码为完成，针对是否#ifdef LengthFieldInTwoByte的代码需要继续完善。