**RTL-SDR驱动程序简析**

* **RTL-SDR驱动简介**

这里的驱动指Osmocom工程师开发的一套命令行工具，包括，在附件中有。且代码开源，可以到官网Wiki下载。官网还提供了GNURadio的模块支持。

其中rtl\_sdr是驱动的主体，可以接收数据流并保存到文件中。还有rtl\_fm等工具，可以接收FM广播等，源代码在Linux下安装libusb-1.0\_0-dev后可以顺利编译，Windows下没有实验。使用方法在Osmocom的Wiki（https://osmocom.org/projects/rtl-sdr/wiki）上可以找到，具体可看驱动程序目录下的README文件。

实际上不了解其原理也没关系，主要是寄存器的各种操作配置，和后端信号处理基本无关（指rtl\_sdr主体，rtl\_fm等工具则涉及到解调算法，就不具体说明了），如果看着更混乱了，跳过即可。

之后我们有空将尝试使用LabView、GNURadio或其它语言等，实现接收到信号的处理工作，跳过硬件操作部分，把目光放在具体应用上——硬件平台的搭建只是开始，RTL-SDR的真正灵魂在软件层面，不重复造轮子是优良品德。

**以下是简单分析后顺手写的笔记，较乱较长且有多处不确定的地方，不再修订格式和语言严谨性，权作记录，可作参考。**

* **驱动程序主线流程简单分析**

主要参考Osmocom提供的驱动，目前没有太深入分析，大致看了librtlsdr.c、rtl\_sdr.h/.c和tuner\_r82xx.c的内容。源代码在Ubuntu安装好libusb-1.0\_0-dev后可以完美编译，也可以使用官方编译好的可执行文件。

可以从中提取出控制RTL-SDR的一个大概流程。rtl\_sdr是CLI程序，官网给出的一个示例用法如下：./rtl\_sdr /tmp/capture.bin -s 1.8e6 -f 392e6。作用是开启RTL\_SDR后设置采样率为1.8MSps，中心频率为392MHz，捕获数据存入/tmp/capture.bin中。

**rtl\_sdr.c**的主函数中，先用**rtlsdr\_get\_device\_count**等搜索设备，并用**rtlsdr\_open**打开设备。参数s和f分别设置samp\_rate和frequency，在后面又分别调用**rtlsdr\_set\_sample\_rate**和**rtlsdr\_set\_center\_freq**来设置这两个参数。然后用**rtlsdr\_set\_tuner\_gain(\_mode)**设置增益模式和增益。最后while里用**rtlsdr\_read\_sync**循环捕获数据，异步则async。

**rtlsdr\_open**中首先是USB设备相关配置，不清楚细节，略。打开设备后进行一次虚写，验证连接，失败则重置USB设备。

接下来调用**rtlsdr\_init\_baseband**，其内部依次进行：

1. USB SIE（0x2000）的配置。分别配置USB\_SYSCTL、USB\_EAP\_MAXPKT和USB\_ETA\_CTL寄存器。设置为Full Packet Mode、开启DMA、最大Packet为2Bytes，还有手册上没有的一些东西。
2. 开启Demodulator各部分的电源。配置了DEMOD\_CTL\_1和DEMOD\_CTL，前者手册上没有。后者的配置开启了PLL、ADC\_I和ADC\_Q，并释放Reset标志位。（应该就是一起全开了，虽然后面因为使用R820T2而用不到Q通道，但电源还是开了，为了兼容其它Tuner；后面也没有关闭，估计是可以关闭的）
3. 重置Demodulator。是把Page1/0x01的第3位置位（0x14）再复位（0x10）。手册上没写，注释说是重置。
4. 关闭Spectrum Inversion和Adjacent Channel Rejection。前者后边又开起来了，后者手册没有。
5. 清零DDC shift和IF Frequency的寄存器。前者不知道，后者是0x09到0x1B。
6. 设置FIR系数。手册没有，估计滤波器用的吧。
7. **【重要】**开启**SDR Mode**。然而手册也没有，是Page0/0x19设为0x05。注释称其bit5还同时关闭了DAGC。还发现了一个test mode，在librtlsdr.c里面的**rtlsdr\_set\_testmode**里，是Page0/0x19设为0x03开启，0x05关闭，显然它们是同一个东西，所以叫“测试模式”是正确的，当然也不排除是驱动作者自己起的名字而非官方。
8. 初始化FSM state-holding register。有限状态机状态保持？
9. 关闭AGC。注释说好像没起作用，不过这个手册上写了。
10. 关闭RF\_AGC和IF\_AGC。这里是初始化部分，有些后边还会改。
11. 关闭PID滤波器。手册有，但不知道干什么的。
12. opt\_adc\_iq设为0，也就是后面提到的交换I、Q通道。这里设为0就是不交换。
13. 开启Zero-IF模式和一系列东西。后边因为用R820T2又关了。
14. 关闭TP\_CKO输出的4.096MHz时钟。好像是用来测试ADC的。

下一步检查Tuner型号，**rtlsdr\_set\_i2c\_repeater(dev, 1)**开启RTL2832U的I2C总线，发送指令从Tuner获取CHECK信息。一个一个型号试过去，找到了就设置dev->tuner\_type。本方案中应该找到R820T型号。其中有两个型号好像还要对GPIO5操作，暂时不管。

确认Tuner型号后goto到found标签，设置两块芯片时钟一致，然后针对Tuner开始操作。对于RTLSDR\_TUNER\_R820T依次会：

1. 首先禁用Zero-IF模式（Page1/0xB1设为0x1A），因为用R820T2就只输出一路接到I通道，而不使用Q通道。这其实同时也开启了DC estimation and cancellation、IQ compensation、IQ estimation for compensation。
2. 第二步（只）使能I通道ADC输入（Page0/0x08设为0x4D）。此处感觉是手册上十七页关于AD\_EN\_regx的描述把I和Q写反了【TODO】。（注：检查发现还有寄存器opt\_adc\_iq可以配置后交换I和Q通道，但是确认了下程序中使用的是正常顺序，故怀疑手册标反）
3. 然后调用**rtlsdr\_set\_if\_freq**设置RTL2832U这边的中频频率为3.57MHz。函数内为设置Page1寄存器0x19到0x1B处22位的值，这个22位数值就代表IF Frequency。11…111（22个1）代表Xtal的频率（28.8MHz），0代表0，故程序中是(freq\*TWO\_POW(22)) / rtl\_xtal，但为什么乘-1不知道。
4. 最后开启Spectrum Inversion（Page1/0x15设为0x01）。这也不知道作用。

完成后调用tuner的init函数（此处则对应r82xx\_init函数）。**r82xx\_init**中，首先会将寄存器的初始值（r82xx\_init\_array，包含从05到1F的初始配置，没有细看）整体写入R820T2。然后调用**r82xx\_set\_tv\_standard**和**r82xx\_sysfreq\_sel**。

其中**r82xx\_set\_tv\_standard**传入了bw为3（所以f\_symbol约为三点几……？），函数中就会配置一堆参数，其他都不认识，就一个if\_khz=3570比较眼熟。然后会备份一下寄存器当前状态到priv->regs（刚用r82xx\_init\_array初始化完的状态）。接下来是几个手册没提的，看起来和CHECK、VERISON、TEST相关，应该不是很重要。最后设了个标定过程，说会强制标定一次，具体内容不清楚。

**r82xx\_sysfreq\_sel**完全不清楚，后面还有LNA相关的配置，前面不清楚。

最后关闭I2C Repeater，到此RTL-SDR开启完成。

接下来是**rtlsdr\_set\_sample\_rate**。因为前面提到的Resample的原因，驱动中给它设定了一个上限MAX\_SAMP\_RATE，为3.2MSps（据说实测不丢样本到不了），如果设定值超过该上限则固定为该值。

然后应该是整数转换的精度考虑，驱动中把输入的采样率先转22位二进制数（2的22次方为4194304，设置重采样率相关内容手册P20有部分提及，虽然不知道是什么原理，不过好像是BW\*8/7=f\_symbol…？），去掉末两位再转回rate值，以此获得实际设置的采样率数值。随后设置Bandwidth，不过使用R820T好像不用设。

这里实际上设置的是Resampler的采样率。下一步即设置寄存器rsamp\_ratio，在Page1/{0x9F,0xA2}的[27:2]位。下面还有一步**rtlsdr\_set\_sample\_freq\_correction**，看起来是一些误差纠正，传入ppm，涉及到的寄存器手册没提。

最后通过Page1/0x01重置芯片，并且**rtlsdr\_set\_offset\_tuning**重新计算offset frequency，不太清楚具体干什么的，但该函数在使用R820T时直接跳出了，实际没有执行。

然后是**rtlsdr\_set\_center\_freq**。里面判断是否直采，是的话就是直接用前面用过的**rtlsdr\_set\_if\_freq**设置中频，不是的话就调用tuner的set\_freq函数。我们使用Tuner，所以不是直采模式，此处则对应执行**r82xx\_set\_freq**函数，其内部依次：

1. 先用**r82xx\_set\_mux**配置选择器。总之是配置R820T各个模块。具体项目写在草稿纸，写不动了。
2. 再用**r82xx\_set\_pll**配置锁相环。大致结构了解，但具体配置不明。比如pll autotune之类东西不知道是什么。
3. 后边的东西如果用R820T似乎不用管。

**注：**（1）和（2）里两个函数里边传入的freq，来源都是lo\_freq = freq + priv->int\_freq，而这个int\_freq在**r82xx\_set\_tv\_standard**里边被设为中频（3.57MHz之类的东西），这个加号就有点不太懂，应该需要看了**r82xx\_set\_tv\_standard**之后再说【TODO】。

还有**rtlsdr\_set\_tuner\_gain\_mode**。传0则为手动模式，1为自动模式，内部就是调用tuner的**set\_gain\_mode**函数，此处对应**r82xx\_set\_gain**函数。手动模式下还会用**rtlsdr\_set\_tuner\_gain**设置增益，具体略；**自动增益控制**（AGC）模式下，会开启几个寄存器标志位（是R820T2的，不是2832的）：0x05[4] LNA\_GAIN\_MODE设为0（auto），0x07[4] MIXGAIN\_MODE设为1（auto），还有一个VGA\_MODE它没设为由VAGC\_Pin控制，而是用VGA\_CODE固定为了+26.5dB，查看其他地方也没发现有改回来，所以IF\_AGC那路其实没用到……？感觉不太对，再说【TODO】。

后面就是循环捕获，暂略。

* **驱动的其他部分、整体框架等**

暂略。