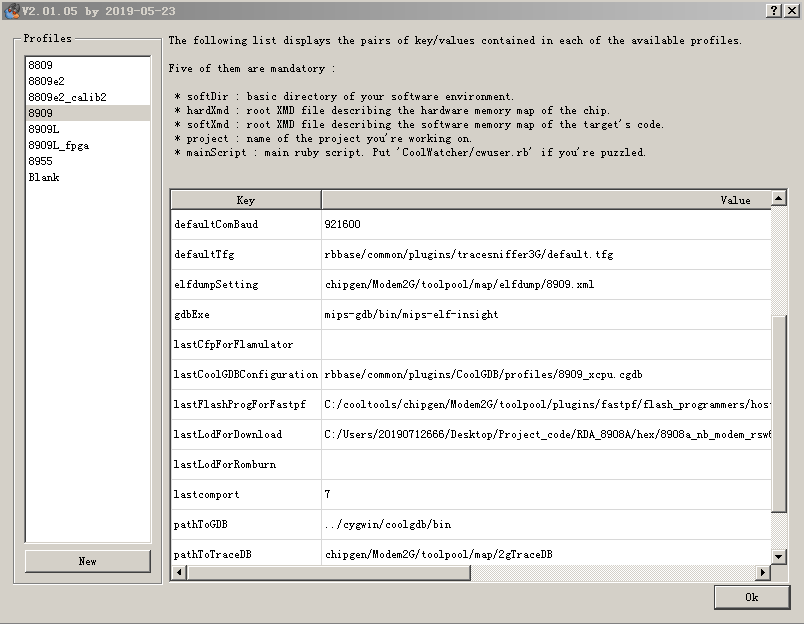
**模块的LOG调试方法的积累**

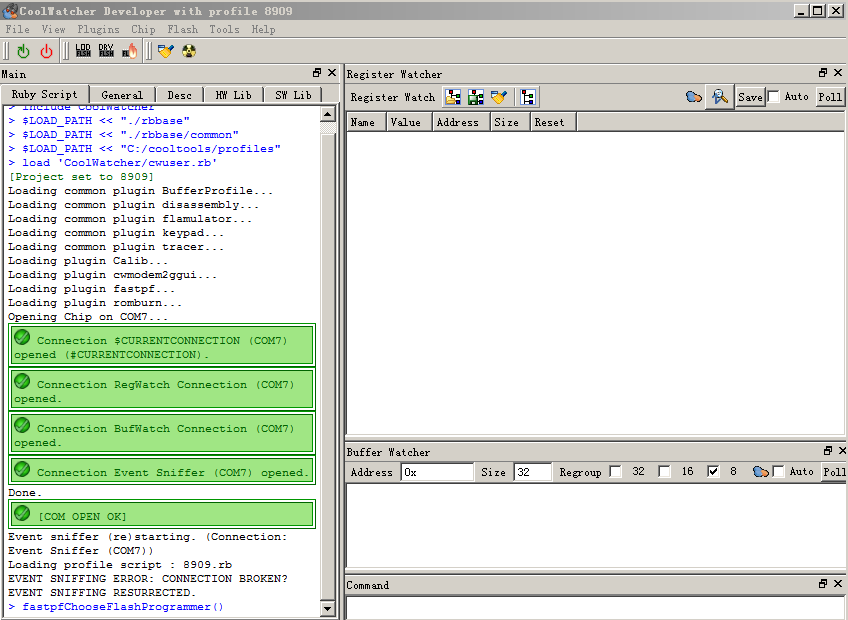
—— 李宣廷

# 1．coolwatcher的使用

打开coolwatcher，N21模块选择8909，端口选择模块host所对应的端口。

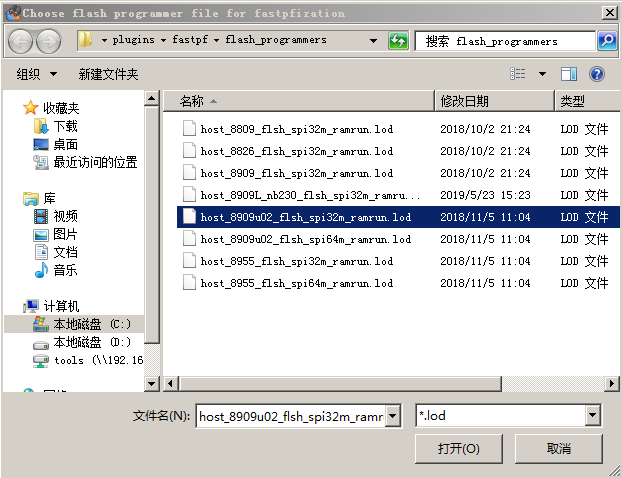


成功打开如下图所示

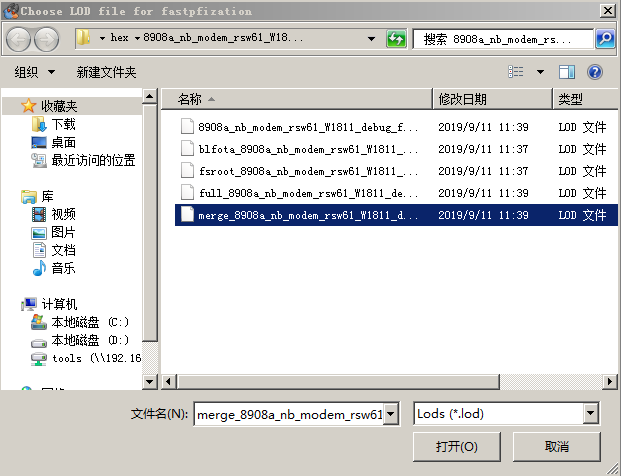


# 2.模块版本的烧写

点击，选择内容如下所示。



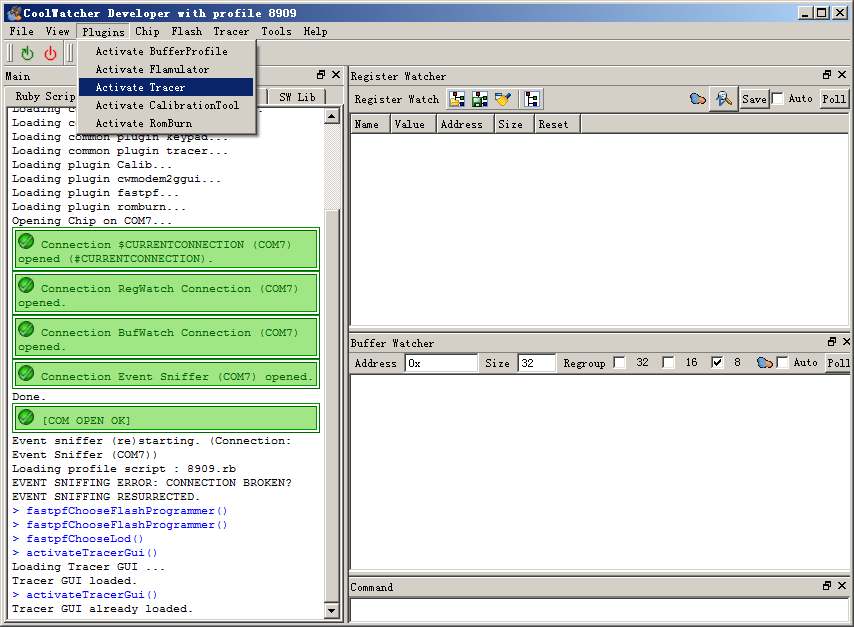
点击，选择

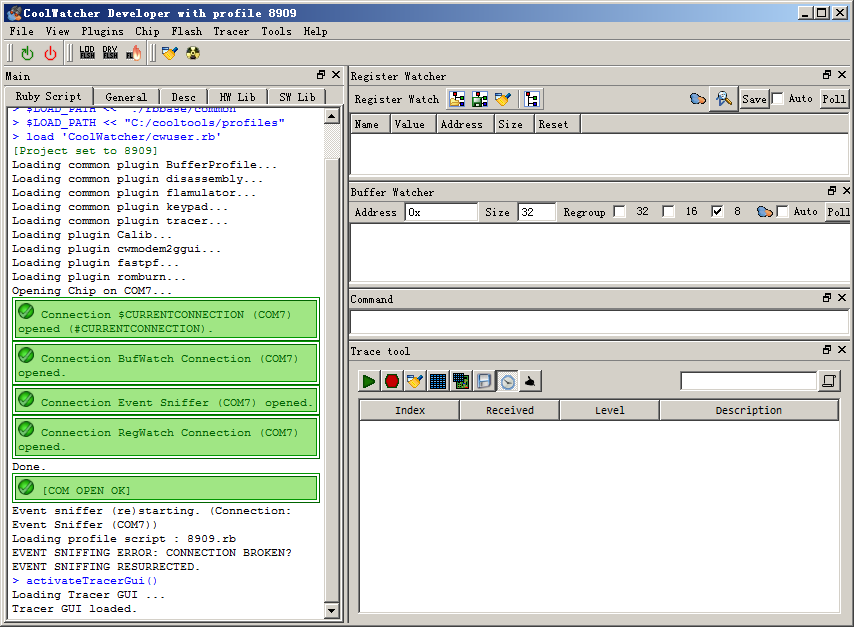


点击进行烧写，等进度条到100%，表示烧写完毕。

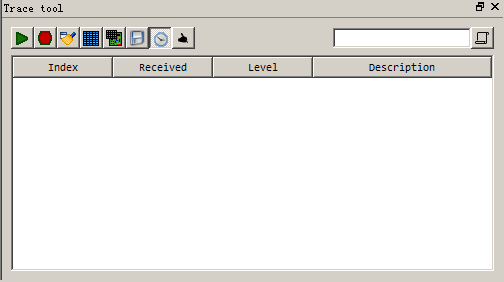
# 3.log信息抓取

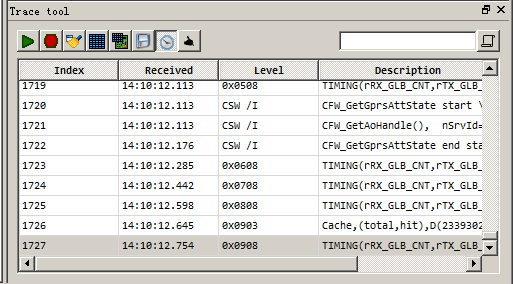
coolwatcher连接模块后，点击Plugins->Activate Tracer,打开Trace tool.





点击绿色三角形后开始进行log信息的输出。点击红色的圆后，停止抓取log信息。





# 4.log信息的修改和增加

要修改和增加log，首先要知道log输出函数所在的位置。用sscom打开模块的uart口，发送at+ledmode=1。然后在trace tool里面查找有关ledmode的log，找到：

AT CMD0 line: at+linemode=1

AT CMD parse start amd len=8: +ledmode

AT CMD0 run +LEDMODE type=0 param=1

以AT CMD0 line: at+linemode=1为例，进行log信息的修改和添加。

在代码中搜索“AT CMD%d line:”，找到此log输出所在的函数。

static void at\_CmdRunCommandBuff(AT\_CMD\_ENGINE\_T \*th)

{

AT\_CMD\_BUF \*pCmdBuf = &th->module.cmd\_buf;

COS\_LOGXI(COS\_LOGPAR\_IS, 0x100000d6, "AT CMD%d line: %s", th->channelId, pCmdBuf->DataBuf);

int result = at\_ModuleProcessLine(pCmdBuf->DataBuf, pCmdBuf->DataLen, th);

…

…

}

其中，COS\_LOGXI(COS\_LOGPAR\_IS, 0x100000d6, "AT CMD%d line: %s", th->channelId, pCmdBuf->DataBuf);为log输出的代码。

在修改log信息时，出现的问题是：修改"AT CMD%d line: %s"，log输出没有变化。后来通过师父讲解，查看0x100000d6这个参数代表着什么。打开cooltools\chipgen\Modem2G\toolpool\map\2gTraceDB\traceDb.yaml后发现，原来COS\_LOGXI的第三个参数并不是log内容，根据我的理解和测试，第三个参数是显示一下第二个参数里面所写的log信息的内容。若修改log信息，需要修改traceDb.yaml文件里面对应的信息；同理，添加log信息，需要在traceDb.yaml文件里，写入新的log内容。

修改完log信息后，双击neoway\_build\_N21.bat文件进行编译，编译通过后显示===============build Success!!=====================此时，在hex文件里会出现新编译的文件，通过第2部分所写的内容，进行新版本的烧写。以下是对log信息的修改和添加的测试。

0x100052e5: "lxt logadd test"

0x100052e6: "lxt logadd test, AT CMD%d line: %s"

0x100052e7: "lxt logadd test%d,%s"

