Modem Develop Silver Bullet

**更改历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **状态** | **日期** | **责任人** | **更改原因** |
| 1.0 | 草稿 | 2022-05-2 | liuyongqi (刘永琪) | 创建 |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[Modem Develop Silver Bullet 1](#_Toc164157147)

[目录 2](#_Toc164157148)

[1 介绍 9](#_Toc164157149)

[1.1 文档目的 9](#_Toc164157150)

[1.2 术语和缩写 9](#_Toc164157151)

[1.2.1 术语 9](#_Toc164157152)

[1.3 参考文档 9](#_Toc164157153)

[2 ECM9100常见问题记录 10](#_Toc164157154)

[2.1 NAS 10](#_Toc164157155)

[2.1.1 手动搜网独占AT通道无法关机 10](#_Toc164157156)

[2.1.2 IPV6激活DNS未获取 10](#_Toc164157157)

[2.1.3 搜网过程 11](#_Toc164157158)

[2.1.4 AP频繁自动加载 11](#_Toc164157159)

[2.1.5 Service Request流程的结束 12](#_Toc164157160)

[2.2 RRC 12](#_Toc164157161)

[2.2.1 重选失败未将重选失败的小区BAR 300s 12](#_Toc164157162)

[2.2.2 Modification period 12](#_Toc164157163)

[3 协议流程学习 13](#_Toc164157164)

[3.1 基本的概念 13](#_Toc164157165)

[3.1.1 UE mode of operation 13](#_Toc164157166)

[3.2 LTE PHY 13](#_Toc164157167)

[3.2.1 Cell Search 14](#_Toc164157168)

[3.3 EMM 14](#_Toc164157169)

[3.3.1 概述 14](#_Toc164157170)

[3.3.2 术语记录 14](#_Toc164157171)

[3.3.3 EMM流程的类型 14](#_Toc164157172)

[3.3.4 EMM状态机 15](#_Toc164157173)

[3.3.5 EMM子层状态 15](#_Toc164157174)

[3.3.6 EPS update status 16](#_Toc164157175)

[3.4 ESM 17](#_Toc164157176)

[3.4.1 概述 17](#_Toc164157177)

[3.4.2 ESP流程的种类 17](#_Toc164157178)

[3.4.3 ESM子层的状态 18](#_Toc164157179)

[3.4.4 TFT 18](#_Toc164157180)

[3.5 IP地址分配 24](#_Toc164157181)

[3.5.1 通过NAS信令分配IP地址 24](#_Toc164157182)

[3.5.2 网络分配IP的处理过程 24](#_Toc164157183)

[3.5.3 IP头压缩 25](#_Toc164157184)

[3.5.4 ESM流程的地址处理原则 25](#_Toc164157185)

[3.6 PLMN Related 27](#_Toc164157186)

[3.6.1 高优先级PLMN搜索 27](#_Toc164157187)

[3.6.2 FPLMN处理 28](#_Toc164157188)

[3.7 RRC 28](#_Toc164157189)

[3.7.1 RRC Connection Request 28](#_Toc164157190)

[4 编解码 28](#_Toc164157191)

[5 科普类 28](#_Toc164157192)

[5.1.1 什么是MS、UE、MT、TE、TA 28](#_Toc164157193)

[5.1.2 Feature Phone和Smart Phone 28](#_Toc164157194)

[5.1.3 什么是Non-IP 29](#_Toc164157195)

[5.1.4 NIDD的实现 29](#_Toc164157196)

[5.1.5 CIoT EPS optimizations 30](#_Toc164157197)

[5.1.6 传输模式 30](#_Toc164157198)

[5.1.7 RNTI 30](#_Toc164157199)

[5.1.8 什么是DCI 31](#_Toc164157200)

[5.2 借助tshark解码任何码流 31](#_Toc164157201)

[5.2.1 未加密的ESM无法之间通过tshark解码 32](#_Toc164157202)

[6 语句积累 32](#_Toc164157203)

[7 PDCP代码学习 33](#_Toc164157204)

[7.1 协议内容学习 33](#_Toc164157205)

[7.1.1 UPDCP处理流程 33](#_Toc164157206)

[7.1.2 DPDCP处理流程 34](#_Toc164157207)

[7.2 代码学习 34](#_Toc164157208)

[7.2.1 判断IP包为TCP的ACK包 34](#_Toc164157209)

[7.2.2 检测Ping包 35](#_Toc164157210)

[7.2.3 PDCP关键函数 36](#_Toc164157211)

[7.2.4 PDCP的配置过程 37](#_Toc164157212)

[7.2.5 内存相关的函数操作 38](#_Toc164157213)

[7.2.6 PDCP关键Log 38](#_Toc164157214)

[8 RLC 38](#_Toc164157215)

[8.1 RLC TM模式 38](#_Toc164157216)

[9 MAC 38](#_Toc164157217)

[9.1 MAC PDU 38](#_Toc164157218)

[9.1.1 MAC PDU的组成部分 38](#_Toc164157219)

[9.1.2 不同的MAC子头 39](#_Toc164157220)

[9.1.3 MAC PDU的结构 39](#_Toc164157221)

[9.1.4 参数 39](#_Toc164157222)

[9.2 MAC控制单元 40](#_Toc164157223)

[9.2.1 BSR 40](#_Toc164157224)

[9.2.2 C-RNTI MAC Control Element 41](#_Toc164157225)

[9.2.3 DRX Command MAC Control Element 41](#_Toc164157226)

[9.2.4 UE Contention Resolution Identity MAC Control Element 41](#_Toc164157227)

[9.2.5 Timing Advance Command MAC Control Element 41](#_Toc164157228)

[9.2.6 Power Headroom MAC Control Element 41](#_Toc164157229)

[9.2.7 MCH Scheduling Information MAC Control Element 41](#_Toc164157230)

[10 RRC 41](#_Toc164157231)

[10.1 关于SRB 41](#_Toc164157232)

[10.2 测量控制 41](#_Toc164157233)

[11 IPRELAY 42](#_Toc164157234)

[11.1 发送数据过程 42](#_Toc164157235)

[11.1.1 IPC发送数据处理 42](#_Toc164157236)

[11.1.2 AP\_DATA\_TASK的创建 42](#_Toc164157237)

[11.1.3 上行数据的发送流程 42](#_Toc164157238)

[11.1.4 TFT检查 43](#_Toc164157239)

[11.2 接收数据过程 43](#_Toc164157240)

[11.2.1 AP接收数据后的内存释放 43](#_Toc164157241)

[12 ESM代码部分 43](#_Toc164157242)

[12.1 PDPSession 43](#_Toc164157243)

[12.1.1 PdpContext\* pCurPdpContext 43](#_Toc164157244)

[13 eCat专题 43](#_Toc164157245)

[13.1.1 Layer\_Event eCat Scan 43](#_Toc164157246)

[13.1.2 多行文本显示 44](#_Toc164157247)

[14 ECM9500代码结构 44](#_Toc164157248)

[14.1 基本的概念、缩写 44](#_Toc164157249)

[14.2 入口函数 44](#_Toc164157250)

[14.3 ECM9500视频学习 44](#_Toc164157251)

[14.4 Platform\_overview录屏学习 44](#_Toc164157252)

[14.4.1 ESWIN交付/工具/PCTOOL/MORU-PCtool-GUI.exe 45](#_Toc164157253)

[14.4.2 armshrine 45](#_Toc164157254)

[14.4.3 省电相关 46](#_Toc164157255)

[14.4.4 如何编译kernel 46](#_Toc164157256)

[14.5 Innofidei\_P4\_platform\_SW\_Interface文档 47](#_Toc164157257)

[14.5.1 MODULE MANAGEMENT 47](#_Toc164157258)

[14.5.2 MEMORY MANAGEMENT 47](#_Toc164157259)

[14.5.3 Logging 48](#_Toc164157260)

[14.5.4 Platform\_Overview 48](#_Toc164157261)

[14.6 DSP\_Overview录屏学习 LTE\_DSP\_ARCH\_Design 49](#_Toc164157262)

[14.6.1 各个目录的作用 49](#_Toc164157263)

[14.6.2 代码工程 49](#_Toc164157264)

[14.6.3 LTE PHY state 51](#_Toc164157265)

[14.6.4 Tsk\_PSS\_SSS.c 51](#_Toc164157266)

[14.6.5 Tsk\_PBCH.c 51](#_Toc164157267)

[14.6.6 DSP和L1C的通信两种方式 51](#_Toc164157268)

[14.6.7 DSP-ARM Interface 52](#_Toc164157269)

[14.6.8 DSP-LTE\_HW interface 52](#_Toc164157270)

[14.7 LTE\_DSP\_HW\_Interface\_P4A 53](#_Toc164157271)

[14.7.1 系统模块图 53](#_Toc164157272)

[14.7.2 DSP-FPGA之间的接口 53](#_Toc164157273)

[14.7.3 中断和总线连接 53](#_Toc164157274)

[14.7.4 寄存器定义 53](#_Toc164157275)

[14.8 LTE UE L1 Interface Design Specification for P4(v2) 54](#_Toc164157276)

[14.8.1 System Overview 54](#_Toc164157277)

[14.8.2 L1 Interface 55](#_Toc164157278)

[14.8.3 L1 Interface Procedure 55](#_Toc164157279)

[14.9 FS&CS 58](#_Toc164157280)

[14.10 Platform\_MemInfo\_Dump\_etc 58](#_Toc164157281)

[14.10.1 平台内存分配相关的函数接口 58](#_Toc164157282)

[14.10.2 内存/程序运营情况 60](#_Toc164157283)

[14.10.3 Note 61](#_Toc164157284)

[14.11 PS\_AT&USIM&SMS培训视频 62](#_Toc164157285)

[14.12 PS+PHY软件编译环境演示 63](#_Toc164157286)

[14.13 代码中的module 63](#_Toc164157287)

[14.13.1 模块的定义 63](#_Toc164157288)

[14.13.2 Module.h文件中的枚举变量 63](#_Toc164157289)

[14.14 流程整理 64](#_Toc164157290)

[14.15 NAS代码整理 64](#_Toc164157291)

[14.15.1 EMmSession 64](#_Toc164157292)

[15 C/C++常见操作记录 64](#_Toc164157293)

[15.1 C++基础 64](#_Toc164157294)

[15.1.1 如何设置枚举类型的整型类型 64](#_Toc164157295)

[15.2 库函数 64](#_Toc164157296)

[15.2.1 realloc 64](#_Toc164157297)

[15.2.2 malloc\_usable\_size 65](#_Toc164157298)

[15.2.3 calloc 65](#_Toc164157299)

[15.2.4 srand/rand 65](#_Toc164157300)

[15.2.5 pcap\_loop 65](#_Toc164157301)

[15.2.6 getopt 66](#_Toc164157302)

[15.2.7 calloc 66](#_Toc164157303)

[15.2.8 strdup 66](#_Toc164157304)

[15.2.9 sprintf 66](#_Toc164157305)

[15.3 STL 66](#_Toc164157306)

[15.3.1 map 66](#_Toc164157307)

[15.3.2 hash\_map和map 66](#_Toc164157308)

[15.4 操作符相关 67](#_Toc164157309)

[15.4.1 #与## 67](#_Toc164157310)

[15.4.2 Enum 67](#_Toc164157311)

[15.4.3 类默认的属性类型 67](#_Toc164157312)

[15.4.4 operator 67](#_Toc164157313)

[15.4.5 sizeof 68](#_Toc164157314)

[15.4.6 do{}while(0) 68](#_Toc164157315)

[15.4.7 MemCheck实现的几个例子 68](#_Toc164157316)

[15.5 类相关 68](#_Toc164157317)

[15.5.1 C++关键字friend 68](#_Toc164157318)

[15.5.2 C++虚函数回顾 69](#_Toc164157319)

[15.5.3 类模板 69](#_Toc164157320)

[15.5.4 inline 69](#_Toc164157321)

[15.5.5 const 69](#_Toc164157322)

[15.6 内存操作 69](#_Toc164157323)

[15.7 线程 70](#_Toc164157324)

[15.7.1 线程锁 70](#_Toc164157325)

[15.8 文件系统 70](#_Toc164157326)

[15.8.1 FILE 70](#_Toc164157327)

[15.9 Makefile 71](#_Toc164157328)

[15.9.1 常见的变量 71](#_Toc164157329)

[15.9.2 链接符号 71](#_Toc164157330)

[15.9.3 make 72](#_Toc164157331)

[16 基本工具操作 72](#_Toc164157332)

[16.1 Win10常见操作 72](#_Toc164157333)

[16.1.1 快捷键 72](#_Toc164157334)

[16.1.2 Win10常见问题和解决方法 72](#_Toc164157335)

[16.2 VS Code操作 73](#_Toc164157336)

[16.2.1 VS Code快速添加注释 73](#_Toc164157337)

[16.2.2 快速打开终端 73](#_Toc164157338)

[16.2.3 快速格式化代码 73](#_Toc164157339)

[16.3 eCat工具 73](#_Toc164157340)

[16.3.1 eCat工程编译 73](#_Toc164157341)

[16.4 Eclipse/ZSP 73](#_Toc164157342)

[16.4.1 Eclipse中显示编译problem的方法： 73](#_Toc164157343)

[16.4.2 设置为UTF-8编码 74](#_Toc164157344)

[16.4.3 "IOConsole updater" has encounter a problem 74](#_Toc164157345)

[16.4.4 如何将多个.o文件连接为一个 74](#_Toc164157346)

[16.5 ZSP调试工具 74](#_Toc164157347)

[16.5.1 安装 74](#_Toc164157348)

[16.5.2 ZSP调试项目代码 76](#_Toc164157349)

[16.5.3 根据地址确定代码所在的行 76](#_Toc164157350)

[16.5.4 地址写入断点的设置 77](#_Toc164157351)

[16.5.5 启动调试停在\_start位置 77](#_Toc164157352)

[16.5.6 BSS段设置为0的操作 77](#_Toc164157353)

[16.5.7 ZSP打断点的方法 78](#_Toc164157354)

[16.6 梳理FreeRTOS内存管理流程 79](#_Toc164157355)

[16.6.1 关键的类型/变量 79](#_Toc164157356)

[16.7 GCC 83](#_Toc164157357)

[16.7.1 常见的报错 83](#_Toc164157358)

[16.8 Visual Studio 84](#_Toc164157359)

[16.8.1 Visual Studio预定义宏 84](#_Toc164157360)

[16.8.2 Visual Studio常见报错 85](#_Toc164157361)

[16.8.3 Visual Studio常见问题解决 86](#_Toc164157362)

[16.8.4 Visual Studio快捷键 86](#_Toc164157363)

[16.9 Source Insight 86](#_Toc164157364)

[16.9.1 常用快捷键 86](#_Toc164157365)

[16.9.2 将函数名的字体设置为大号字体，加粗，醒目 87](#_Toc164157366)

[16.9.3 如何在SI IDE中设置一个全局变量 87](#_Toc164157367)

[16.9.4 指定目录执行搜索 87](#_Toc164157368)

[16.10 Word 88](#_Toc164157369)

[16.10.1 如何将一个目录中的所有.doc文件批量转换为.docx格式文件 88](#_Toc164157370)

[16.11 Excel 88](#_Toc164157371)

[16.11.1 常见问题及对应的解决方法 88](#_Toc164157372)

[16.12 Git 89](#_Toc164157373)

[16.12.1 Git原理 89](#_Toc164157374)

[16.12.2 Git基本操作 89](#_Toc164157375)

[16.12.3 常见问题 89](#_Toc164157376)

[16.13 Jenkins 91](#_Toc164157377)

[16.13.1 Jenkins安装注意事项 91](#_Toc164157378)

[16.13.2 Jenkins邮件无法发送 93](#_Toc164157379)

[16.13.3 Jenkins自动邮件中添加文件内容 93](#_Toc164157380)

[16.13.4 Jenkins自动邮件中插入图片 93](#_Toc164157381)

[16.13.5 Jenkins设置通过GitLab提交触发 95](#_Toc164157382)

[16.13.6 Jenkins设置每次构建的名字 96](#_Toc164157383)

[16.14 测试工具使用 97](#_Toc164157384)

[16.14.1 RIDE自动化测试工具 97](#_Toc164157385)

[16.15 MAP文件分析 98](#_Toc164157386)

[16.15.1 基本概念 98](#_Toc164157387)

[16.16 PlantUML 98](#_Toc164157388)

[16.16.1 给箭头修改颜色 98](#_Toc164157389)

[16.16.2 文本对齐 99](#_Toc164157390)

[16.16.3 改变箭头的样式 99](#_Toc164157391)

[16.16.4 消息序列的自动编号 99](#_Toc164157392)

[16.16.5 添加标题 99](#_Toc164157393)

[16.16.6 分隔示意图 99](#_Toc164157394)

[16.16.7 组合消息 99](#_Toc164157395)

[16.16.8 注释信息 100](#_Toc164157396)

[16.16.9 分隔符 100](#_Toc164157397)

[16.16.10 引用 100](#_Toc164157398)

[16.16.11 延迟 101](#_Toc164157399)

[16.16.12 文本换行 101](#_Toc164157400)

[16.16.13 空间 101](#_Toc164157401)

[16.16.14 生命线的激活与撤销 101](#_Toc164157402)

[16.16.15 锚点和持续时间 102](#_Toc164157403)

[17 芯片公司 102](#_Toc164157404)

[17.1 半导体 102](#_Toc164157405)

[17.1.1 芯科科技 Silicon Lab 102](#_Toc164157406)

[17.1.2 ADI 102](#_Toc164157407)

[17.1.3 ASTRI 102](#_Toc164157408)

[17.1.4 创毅视讯 Innofidei 102](#_Toc164157409)

[17.1.5 瀚讯 102](#_Toc164157410)

[17.1.6 芯原微电子（上海）股份有限公司 102](#_Toc164157411)

[17.1.7 移芯通信 103](#_Toc164157412)

[17.1.8 芯翼 104](#_Toc164157413)

[17.1.9 翱捷ASR 104](#_Toc164157414)

[17.1.10 智联安 104](#_Toc164157415)

[17.1.11 瑞普康 104](#_Toc164157416)

[17.1.12 创芯汇联 104](#_Toc164157417)

[17.1.13 芯晟 104](#_Toc164157418)

[17.1.14 星思 104](#_Toc164157419)

[17.1.15 宇龙 104](#_Toc164157420)

[17.2 模组厂商 104](#_Toc164157421)

[17.2.1 广和通 104](#_Toc164157422)

[17.3 竞品分析 104](#_Toc164157423)

[17.3.1 EC618供参考 104](#_Toc164157424)

[18 周报 104](#_Toc164157425)

[18.1 2022年-5月 104](#_Toc164157426)

[18.1.1 Weekly 20 – 五月第二周 104](#_Toc164157427)

[18.1.2 Weekly21 – 五月第三周 104](#_Toc164157428)

[19 MORU原始平台代码学习 105](#_Toc164157429)

[19.1 makefile 105](#_Toc164157430)

[20 PS代码平台相关 105](#_Toc164157431)

[20.1 消息和模块 105](#_Toc164157432)

[20.2 AT命令相关模块 105](#_Toc164157433)

[20.2.1 MODULE\_ATCMD 105](#_Toc164157434)

[20.2.2 MODULE\_ATOUTPUT 105](#_Toc164157435)

[20.2.3 MODULE\_ATMT 105](#_Toc164157436)

[20.2.4 MODULE\_ATCMDLINE 105](#_Toc164157437)

[21 SMS模块学习 105](#_Toc164157438)

[21.1 SMS模块中的代码文件 105](#_Toc164157439)

[21.2 关键类定义 106](#_Toc164157440)

[21.2.1 SmsCommonConfig 106](#_Toc164157441)

[21.2.2 SmsMain.cpp文件 106](#_Toc164157442)

[21.3 函数接口定义 106](#_Toc164157443)

[21.4 SMS消息处理 107](#_Toc164157444)

[21.4.1 配置类消息的处理函数 107](#_Toc164157445)

[21.4.2 处理发送接收短信相关的消息 107](#_Toc164157446)

[21.5 配置类结构体说明 107](#_Toc164157447)

[21.6 收发短信相关结构体说明 107](#_Toc164157448)

[21.7 ATMT与SMS消息交互 107](#_Toc164157449)

[21.7.1 短信格式设置：AT+CMGF 107](#_Toc164157450)

[21.7.2 SMS发送给ATMT的消息 112](#_Toc164157451)

[22 内存泄漏统计工具 113](#_Toc164157452)

[23 AT命令 113](#_Toc164157453)

[23.1 代码中的AT命令 113](#_Toc164157454)

[23.1.1 关键结构体 113](#_Toc164157455)

[23.1.2 AT命令解析 114](#_Toc164157456)

[23.1.3 MtMain中的函数接口 114](#_Toc164157457)

[23.2 常见AT命令整理 116](#_Toc164157458)

[23.3 Weekly 22 – 五月第四周 116](#_Toc164157459)

[24 3GPP相关 118](#_Toc164157460)

[24.1 3GPP各个Release之间的功能差别 118](#_Toc164157461)

[25 RRC代码学习 118](#_Toc164157462)

[25.1 RRC信令无线承载 118](#_Toc164157463)

[25.2 各个逻辑信道上的RRC消息 118](#_Toc164157464)

[25.2.1 BCCH-BCH-Message 118](#_Toc164157465)

[*25.2.2* *BCCH-DL-SCH-Message* 119](#_Toc164157466)

[*25.2.3* *PCCH-Message* 119](#_Toc164157467)

[*25.2.4* *DL-CCCH-Message* 119](#_Toc164157468)

[25.2.5 *DL-DCCH-Message* 119](#_Toc164157469)

[25.2.6 *UL-CCCH-Message* 119](#_Toc164157470)

[*25.2.7* *UL-DCCH-Message* 119](#_Toc164157471)

[25.3 常用的常量定义 119](#_Toc164157472)

[25.3.1 RRC状态机相关 119](#_Toc164157473)

[25.4 RRC常用函数 120](#_Toc164157474)

[25.4.1 常用宏函数 120](#_Toc164157475)

[26 代码空间压缩 120](#_Toc164157476)

[26.1 各个内存分区的含义 120](#_Toc164157477)

[26.2 ESM部分 121](#_Toc164157478)

[26.2.1 定义部分 121](#_Toc164157479)

[27 Miscellaneous 121](#_Toc164157480)

[27.1 时钟同步方式 121](#_Toc164157481)

[27.2 AMT 121](#_Toc164157482)

[27.2.1 AGC 121](#_Toc164157483)

[27.2.2 AGF 122](#_Toc164157484)

[27.2.3 APC 122](#_Toc164157485)

[27.2.4 LOLeakage 122](#_Toc164157486)

[27.2.5 ACLR 122](#_Toc164157487)

[28 RTC功能 122](#_Toc164157488)

[28.1 CP侧与系统时间相关的流程 123](#_Toc164157489)

[28.2 NITZ 123](#_Toc164157490)

[28.3 NTP 123](#_Toc164157491)

[29 Linux常用操作 123](#_Toc164157492)

[29.1 命令行操作 123](#_Toc164157493)

[29.1.1 Ctrl+U 将光标前面的内容删掉。 123](#_Toc164157494)

[29.1.2 Ctrl+K 将光标后面的内容删掉。 123](#_Toc164157495)

[29.1.3 Ctrl+A 光标移动到命令行开始处。 123](#_Toc164157496)

[29.1.4 Ctrl+E 光标移动到命令行尾部。 123](#_Toc164157497)

[29.1.5 Ctrl+W 删除前面的单词 123](#_Toc164157498)

[29.1.6 Ctrl+Y 恢复删除前的命令 123](#_Toc164157499)

[29.1.7 Ctrl+P 上一条命令 124](#_Toc164157500)

[29.1.8 Ctrl+B 向左移动一个字符 124](#_Toc164157501)

[29.1.9 Ctrl+F 向右移动一个字符 124](#_Toc164157502)

[29.1.10 Ctrl+R 历史命令 124](#_Toc164157503)

[29.1.11 Ctrl+D 删除光标右方的字符 124](#_Toc164157504)

[29.1.12 Ctrl+L 清屏 124](#_Toc164157505)

[29.1.13 Ctrl+M Enter 124](#_Toc164157506)

[29.1.14 Ctrl+ 左箭头 向左移动一个单词 124](#_Toc164157507)

[29.1.15 Ctrl+ 右箭头 向右移动一个单词 124](#_Toc164157508)

[29.2 常用命令 124](#_Toc164157509)

[29.2.1 tee 124](#_Toc164157510)

[29.2.2 script 124](#_Toc164157511)

[29.2.3 ldd 124](#_Toc164157512)

[29.2.4 readelf 125](#_Toc164157513)

[29.2.5 file 125](#_Toc164157514)

[29.2.6 cat 125](#_Toc164157515)

[29.2.7 tac 125](#_Toc164157516)

[29.2.8 more 125](#_Toc164157517)

[29.2.9 less 125](#_Toc164157518)

[29.2.10 scp 125](#_Toc164157519)

[29.3 Linux常见问题 126](#_Toc164157520)

[29.3.1 Ubuntu中的时区不对，显示时间与北京时间相差8小时 126](#_Toc164157521)

[30 python常见操作 126](#_Toc164157522)

[30.1 常见函数使用 126](#_Toc164157523)

[30.1.1 range()函数的使用 126](#_Toc164157524)

[30.1.2 round() 126](#_Toc164157525)

[30.2 列表 126](#_Toc164157526)

[30.2.1 列表的遍历 126](#_Toc164157527)

[30.2.2 列表的增删 126](#_Toc164157528)

[30.3 时间 126](#_Toc164157529)

[30.4 文件 126](#_Toc164157530)

[30.4.1 获取指定文件的绝对路径 126](#_Toc164157531)

[30.4.2 当前目录切换 127](#_Toc164157532)

[30.4.3 文件重命名 127](#_Toc164157533)

[30.5 发布python程序 127](#_Toc164157534)

[31 协议重点内容 127](#_Toc164157535)

[32 Web学习 127](#_Toc164157536)

[32.1 ChatGPT的相关提问回答 127](#_Toc164157537)

[32.1.1 关于JavaScript 127](#_Toc164157538)

[32.1.2 Web前端和后端在Web应用程序中承担不同的任务 127](#_Toc164157539)

[32.1.3 Web后端技术 128](#_Toc164157540)

[32.1.4 Web相关的技术有哪些 128](#_Toc164157541)

[32.1.5 Web学习步骤 128](#_Toc164157542)

[32.2 HTML常见问题 128](#_Toc164157543)

[33 工具设计 128](#_Toc164157544)

[33.1 Map分析工具 128](#_Toc164157545)

[33.1.1 需求 128](#_Toc164157546)

[33.2 eCat Log工具分析 129](#_Toc164157547)

[33.2.1 DisplayText 129](#_Toc164157548)

[33.3 3gppdecoder 129](#_Toc164157549)

[33.4 SDR LTE 129](#_Toc164157550)

# 介绍

## 文档目的

本文档用于记录新人入职奕斯伟期间各种账号申请的方法和流程，以及一些常用资料的获取说明、常用工具的安装和配置，旨在加快新人的办公环境和开发环境的搭建，避开一些常见的问题，提升效率。

## 术语和缩写

### 术语

1. ESWIN：E-solution to win
2. OA：Office Automation，办公自动化
3. ETX：Exceed TurboX，一款远程连接到windows/Linux的工具。
4. 3GPP：3rd Generation Partnership Project，第三代合作伙伴计划
5. ETSI：European Telecommunications Standards Institute，欧洲电信标准化协会
6. JIRA：JIRA是Atlassian公司出品的项目与事务跟踪工具，被广泛应用于缺陷跟踪、客户服务、需求收集、流程审批、任务跟踪、项目跟踪和敏捷管理等工作领域。
7. Confluence：一个专业的企业知识管理与协同软件，也可以**用于构建企业**[**wiki**](https://baike.baidu.com/item/wiki/97755)。使用简单，但它强大的编辑和站点管理特征能够帮助团队成员之间共享信息、文档协作、集体讨论，信息推送。

## 参考文档

1.《上海新员工Note.docx》康国法 丁会彦

# ECM9100常见问题记录

## NAS

### 手动搜网独占AT通道无法关机

问题：[ECM9100-2446](http://jira.eswincomputing.com/browse/ECM9100-2446)【Modem测试】【MPW\_功能\_协议栈】cops=?期间，所有AT命令不能处理，需要支持关机命令

分析：AT是串行执行，前一个命令不返回，后一个命令无法执行。设计原因，暂不修改。

建议：可以把AT+COPS=?做成一个不阻塞的命令。这个期间可以执行一些与驻网无关的命令，如关机等。

### IPV6激活DNS未获取

参考对比机高通平台的PDN Connectivity Request消息设置不同IP类型的PCO即可。具体设置参考如下章节。

#### IPV4

IPCP、DNS Server IPv4 Address、IP Address allocation via NAS signaling、NWK Req Bearer Control indicator、IPv4 Link MTU Request（网络不回复此请求，可以忽略）

Note：

1. 在IPV4/IPV4V6类型的请求中，一定要带IPCP。
2. IP Address allocation via NAS signaling、NWK Req Bearer Control indicator这两个域也是必带的。

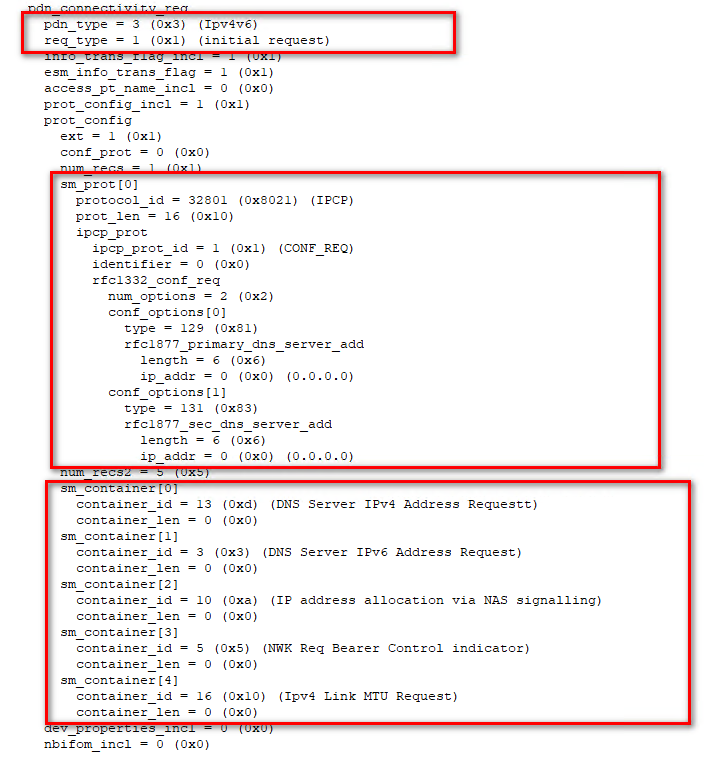


#### IPV4V6

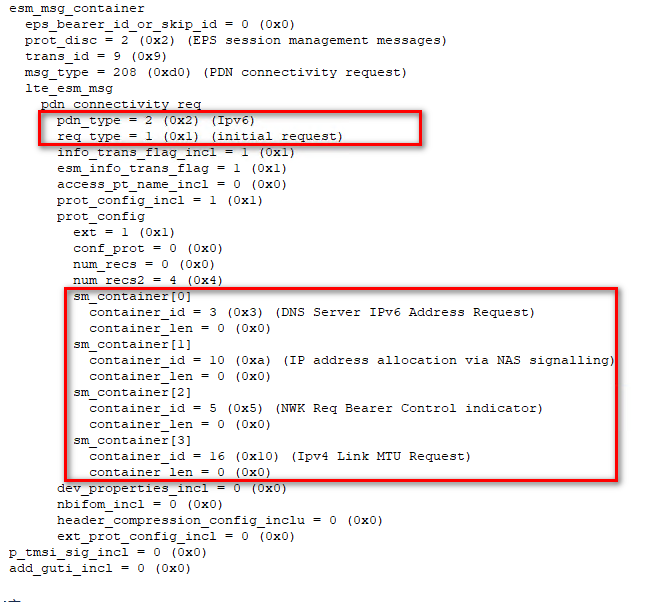
IPCP、DNS Server IPv4 Address、IP Address allocation via NAS signaling、NWK Req Bearer Control indicator、IPv4 Link MTU Request（网络不回复此请求，可以忽略）

Note：

同IPV4中的说明。



#### IPV6



### 搜网过程

#### 自动搜网过程

1. RPLMN、EHPLMN、HPLMN、EPLMN搜索，执行全BandScan
2. Step1失败，Search Available PLMN，执行各个自带的搜索，搜到了就上报，EMM根据PLMN的优先顺序选择驻留

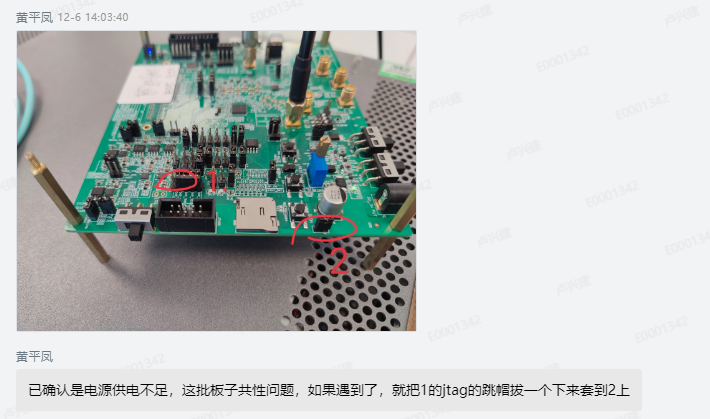
#### 手动搜网过程

Search Available PLMN，设置搜网类型为MANUAL，所有频点搜完后上报所有搜到的PLMN。

### AP频繁自动加载

问题原因：供电不足

处理方法：在如下图中接上跳线帽



### Service Request流程的结束

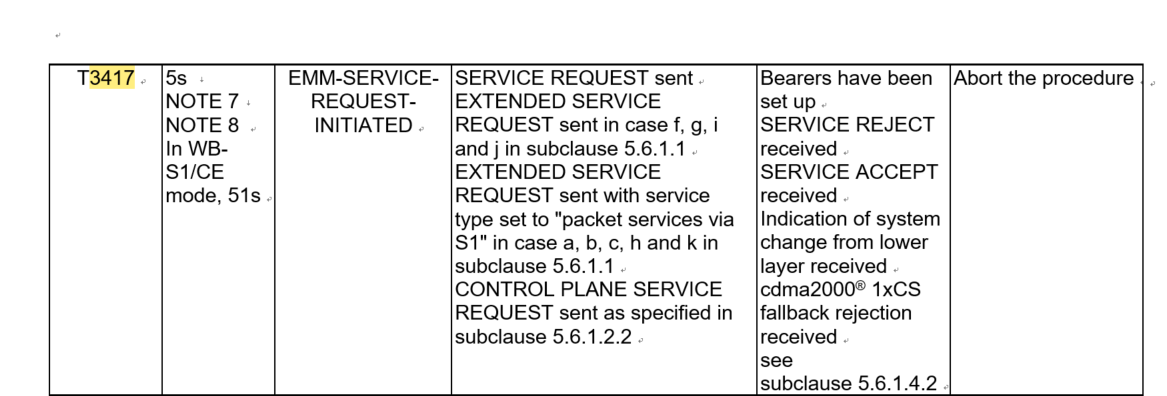
EMM发起Service Request流程后，启动T3417定时器（5s）

发起的原因：

1. 网络有下行信令需要发送（同过Paging消息触发SR）
2. UE有上行信令发送
3. UE或网络有用户数据发送并且当前处于EMM-IDLE状态

Service Request流程的结束：

1. T3417定时器超时，中止Service Request流程
2. Bearers更新了（通过RRCConnectionReconfiguration更新）
3. 收到SERVICE REJECT消息
4. 收到SERVICE ACCEPT消息



## RRC

### 重选失败未将重选失败的小区BAR 300s

[ECM9100-2497](http://jira.eswincomputing.com/browse/ECM9100-2497) 【Modem测试】【MPW\_功能\_协议栈】重选到非HPMN/EHPLMN的小区，驻留失败，没有将该频点加入到bar list中300s

根据协议36304 5.2.4.4节如下字段：

If the highest ranked cell or best cell according to absolute priority reselection rules is an intra-frequency or interfrequency cell which is not suitable for a CN type due to being part of the "list of forbidden TAs for roaming" or belonging to a PLMN which is not indicated as being equivalent to the registered PLMN, the UE shall not consider this cell and other cells on the same frequency, as candidates for reselection for the CN type for a maximum of 300s. If the UE enters into state any cell selection, any limitation shall be removed. If the UE is redirected under E-UTRAN control to a frequency for which the timer is running, any limitation on that frequency shall be removed.

### Modification period

SIB2消息中的RadioResourceConfigCommonSIB（*RadioResourceConfigCommon类型*）

|  |
| --- |
| BCCH-Config ::= SEQUENCE {  modificationPeriodCoeff ENUMERATED {n2, n4, n8, n16}  }  PCCH-Config ::= SEQUENCE {  defaultPagingCycle ENUMERATED {  rf32, rf64, rf128, rf256},  nB ENUMERATED {  fourT, twoT, oneT, halfT, quarterT, oneEighthT,  oneSixteenthT, oneThirtySecondT}  } |

修改周期的计算为：

|  |
| --- |
| ***modificationPeriodCoeff***  Actual modification period, expressed in number of radio frames= *modificationPeriodCoeff* \* *defaultPagingCycle*. n2 corresponds to value 2, n4 corresponds to value 4, n8 corresponds to value 8 and n16 corresponds to value 16. |

modification period = *modificationPeriodCoeff* \* *defaultPagingCycle，单位为无线帧（10ms）*

**问题：如果下发Paging指示系统消息修改，则UE至多在一个修改周期之后重新接收新的系统消息！**



图 1 系统消息的修改

重点内容：

1. 修改周期（modification period）：在一个修改周期内，系统消息可以多次发送，但内容一致，下一个修改周期，网络可以修改系统消息的内容；
2. 修改周期边界（modification period boundaries）：SFN mod m = 0，其中m为修改周期的长度；
3. UE收到Paging中的系统消息修改通知（*systemInfoModification*）后，将在下一个修改周期开始时立即获取系统消息；
4. UE在获取到新的系统消息前使用之前获取到的系统消息；
5. 修改周期的计算方法：*modification period = modificationPeriodCoeff* \* *defaultPagingCycle*，单位为无线帧（10ms），均在SIB2中；
6. SIB1消息中的*systemInfoValueTag指示系统消息中是否发生了变化；*
7. UE在确认系统消息有效3小时后，应认为存储的系统消息无效；

# 工具类

## 工具设计相关

### eCat串口不支持关闭后再开启

[ECM9100-790](http://jira.eswincomputing.com/browse/ECM9100-790)【Modem测试】【FPGA\_功能\_工具】串口关闭再开启后，log输出异常

解决：log需要支持串口关闭再打开，能够正常输出

### log定位相关

ECM9100-825【Modem测试】【FPGA\_功能\_工具】log抽取出来后，双击不能定位到原有log

# NAS相关

## NAS相关

### AT匹配处理

问题：ECM9100-793【Modem测试】【FPGA\_功能\_工具】AT命令无法下发

解决：ATMT在查找AT命令字符串时出错导致无法解析。关键原因：AT命令的匹配处理问题。

### 搜网流程长导致的上传状态卡死

问题：ECM9100-795【Modem测试】【FPGA\_功能\_工具】AT命令at+cfun=1后at工具页面显示error

解决：

问题原因：

步骤：

1，开机发起attach

2，底层搜网需要比较长时间（79s）

3，输入AT+CFUN=0定制搜网并退出，此时EMM没有将状态信息更新到PDP

4，再次发起AT+CFUN=1，由于步骤3 PDP没有退出状态，导致返回ERROR

解决方案：

在步骤3时EMM更新状态到PDP

问题：ECM9100-796【Modem测试】【FPGA\_功能\_协议栈】开机SI接收完成后，未发起随机接入流程

### 关机并没有关闭所有的模块，再次开机出现功能问题

ECM9100-799【Modem测试】【FPGA\_功能\_协议栈】开机注册完成后发at+cfun=0关机，再at+cfun=1开机搜索不到小区，log打印仍然有pucch打印。

ECM9100-808【Modem测试】【FPGA\_功能\_协议栈】开机RRC连接释放后，终端打印UE POWER OFF，之后下发at命令无响应。

解决：RRC只有关机消息，没有开机消息，导致状态无法在开机时清除。

### PDN激活参数设置错误

ECM9100-811【Modem测试】【FPGA\_功能\_协议栈】开机RRCConnectionSetupComplete消息基站已经收到，但没有解出attach\_req消息

解决：没有设置正确的PDN Type；

### EMM关机定时5s后返回关机响应（协议规定）

ECM9100-845【Modem测试】【FPGA\_功能\_协议栈】终端丢网后，发送at+cfun=0无法关机

AT+CFUN=0没有返回的问题，现场分析是由于降频，detach时设置的5s超时延迟到40s，等待超过40s可以返回

### AT发送失败

ECM9100-867【Modem测试】【FPGA\_功能\_协议栈】开机终端回复RRC Connection Setup Complete后，不发attach request，非必现

解决：复现时的场景是**串口被占用的误操作导致at+cfun=0没有发下来**

### PDP上下文空指针（PDP上下文建立的逻辑问题）

ECM9100-1022【Modem测试】【FPGA\_功能\_协议栈】终端收到仪表下发的DRB重配消息后CP死机

解决：死机原因，pdp在收到网侧激活专用承载时打log访问空指针pCurPdpContext

# RRC相关

### 驻留准则不满足

ECM9100-814【Modem测试】【FPGA\_功能\_协议栈】收到系统消息修改的paging，终端没有收系统消息和发起RRC建链请求

解决：

收到系统消息修改的paging，paging通过imsi寻呼，终端接收系统消息和并发起RRC建链请求；

paging通过s-tmsi寻呼，paging通过s-tmsi寻呼;，终端接收系统消息和并发起RRC建链请求；

不是代码原因，因为phy打桩的测量结果不满足S驻留准则；修改打桩测量结果或修改准则配置流程正常；

### L1C与PHY消息不是4字节对齐导致的消息解码错误

ECM9100-836【Modem测试】【FPGA\_功能\_协议栈】连接态进idle态，RRC release后走指定频点搜网流程很慢

### 配置的异频测量频点不支持，所以不起测

ECM9100-853【Modem测试】【FPGA\_功能\_协议栈】空闲态满足异频测量门限，终端没有启动异频测量

解决：根据当前的问题，异频频点不在支持的band合法范围内，因此无法进行配置，需要更新配置复测，后续有问题可继续跟踪

相关的频点说明：

|  |
| --- |
| PRS第1阶段必选支持的频段，  TDD 34 38 39 40 41  FDD 1 3 5 8  @胡浩，2GHz：3GPP定义为Band34，完整范围是2010-2025MHz，中国移动获得的是全部，最初获批是用于部署TD-SCDMA，为3G频段。现在中国移动TD-SCDMA已基本退网，该频段也就清空腾退出来了，一些地区用来部署TD-LTE，扩大4G网络的容量。  2.6GHz：3GPP定义为Band41（4G频段）、n41（5G频段），完整范围是2496-2690MHz，中国移动获得的是2515-2675MHz部分，最初获批是用于部署TD-LTE，为4G频段。现在中国移动用于部署TD-LTE和5G，是其4G网络和5G网络主力覆盖频段。2496-2690MHz此频段也是FDD band7的频段范围。  1，TD-LTE最开始使用的是2575-2635（60MHz），为了部署5G，现在已经将2575-2615MHz这40MHz退频；  2，5G NR现网使用2515-2615MHz（100MHz）；  3，剩余2615-2675MHz这60MHz现网暂时还没大规模使用。 |

### RRC模块关机处理不彻底

ECM9100-868【Modem测试】【FPGA\_功能\_协议栈】开机失败。小区搜索完成后，终端直接发送了一条RRC CONNECTION REESTABLISHMENT REQUEST

解决：开机后，小区搜索后，发起重建是因为上一次的**ue上下文没有释放导致**的；加入开关机功能后，此现象应该消失

### 在连接态收到了Paging消息且本身的S-TMSI包含在了PagingRecordList中

ECM9100-871【Modem测试】【FPGA\_功能\_协议栈】终端在连接态下收到寻呼后，直接RRC释放去了idle态，流程异常

解决：在连接态收到Paging消息确实存在断开RRC连接的处理，协议中并未对此种场景的处理有任何的描述。

回归Paging消息的作用：在连接态收到Paging仅处理系统信息变化的场景，因为Paging消息在UE空闲态时用于触发建链，若寻呼的时IMSI，则触发UE重新发起注册，若为连接态收到了S-TMSI，则UE的行为规范未定义，这里之间忽略即可！

### 系统信息更新失败

ECM9100-995【Modem测试】【FPGA\_功能\_协议栈】终端收到寻呼后不发起随机接入

解决：收到paging消息后，由于需要进行**系统消息更新**，这里接收sib，将nas发送的建链请求放到了buffer中，需要等收到所有需要更新的sibx后再给网侧发送建链请求，但是我们现有代码**是不支持sib10,11,12的**,，导致等待sibx定时器一直重启，并一直等待接收sib10,11,12。可以先看看是否需要支持sib10,11,12，若不用支持就需要修改用例，若需要支持就需要添加SIB10，11，12相应的处理函数。

Note：这里需要支持SIBX消息的部分解码，不支持解析的忽略掉！

### 完整性保护失败

ECM9100-1029【Modem测试】【FPGA\_功能\_协议栈】开机建链失败，基站下发UeCapability后，终端返回RRcConnectionReestablishmentRequest

解决：EIA1 Integrity failure! local MAC\_I=0xae684671, calc MAC\_I=0x65c6ac6，消息的完整性保护检查失败，之间触发了RRC重建，完整性保护算法计算问题。

### 重选过程中导致寻呼漏收

ECM9100-1030【Modem测试】【FPGA\_功能\_物理层】FDD异频重选用例，收不到寻呼

解决：寻呼收不到，是因为重选到假小区，失败后回服务小区也失败，然后逐个对历史频点进行固定频点搜索和指定小区搜索，从而耗时太长，用例已将目标小区信号降低，从而漏收寻呼。ps临时修改，在重选失败回服务小区失败后尝试驻留最新历史频点，可解决此问题，已合入svn。后续会继续对重选策略进行优化。

### 重建MAC-I计算失败

ECM9100-1032【Modem测试】【FPGA\_功能\_协议栈】重建请求mac-I检测错误

解决：重建过程中计算mac-i时，完保算法的入参为全1,但是目前在函数updcp\_rrc\_short\_MAC\_I\_report\_proc中调用updcp\_hw\_integrity\_protection\_proc时的完保入参的direction为0，与协议不符

# 物理层性能

### 同频小区RSRP相差3dB干扰严重

ECM9100-952 【Modem测试】【FPGA\_功能\_物理层】开机小区搜索失败，sib1 fail

这个用例配置了两个小区相差3db，最新代码测试结果强小区的SIB1偶现解错，重传之后能正常解对，弱小区的解不到SIB1的DCI, 原因是同频干扰。 删除一个小区后SIB1也会偶现出错，抓数分析后算法定位是由于码率太高导致RV版本为1和2时候解错概率比较大。性能问题后续算法分析。

### PSS门限设置问题导致真实小区被过滤

ECM9100-1017【Modem测试】【FPGA\_功能\_物理层】TDD开机bandscan小区搜索，弱小区未搜到

解决：加log分析结果是门限值过高导致第二个真实小区的峰值没有过门限，降低门限后能搜到第二个真实小区，而且第二个真实小区的峰值远高于假小区，目前扫频门限值设置不合适，跟算法讨论后修改。算法通过仿真结果觉得将扫频门限的一个系数设置成20比较合理，已修改验证目前能搜到两个小区。