

# V3LTE系统使用说明

北京威视锐科技

## 公司简介



成立于2008年，专注于高性能可编程系统的研发与生产，面向无线通信、智能视觉和测试测量领域提供完整的解决方案。目前已经与超过20多个国家的300多个教育科研机构建立了合作关系。

作为全球最大的可编程器件(FPGA)公司XILINX的全球认证合作伙伴和授权培训中心，威视锐科技提供基于XILINX FPGA/SoC的全方位解决方案。同时威视锐也是全球模拟技术的领头企业ADI大学计划的大中国区合作伙伴，提供基于ADI最新产品和技术的教学和科研平台。

对于产业界客户，威视锐通过严格验证的核心模块、智能便携的测量仪器以及定制化的设计服务来加快产品研发周期。



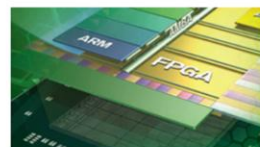
软件定义无线电SDR



软件定义测试仪器SDI



软件定义视觉平台SDVision



软件定义SoC平台SDSoC

同时，威视锐旗下的V3学院作为一家电子工程师教育培训机构，可以为高校学生和工程师提供量身定制的线上和线下培训课程，拥有北京和上海两个教学中心。多年以来，威视锐科技坚持“**Innovation for Research**”的发展理念，对于产业界客户，威视锐通过严格验证的核心模块、智能便携的测量仪器以及定制化的设计服务来加快产品研发周期。

## ■ 基于通用软件无线电技术的LTE系统

- **结构简单成本低。**一台PC集成射频前端，就可以实现一个独立的LTE基站+核心网的功能。而且还可以支持2\*2 MIMO，性能上可以达到下行149 Mbps。
- **外形小巧携带方便。**小型化工业PC，完全可实现基站+核心网。
- **配置灵活。**因为其软件定义的特性，LTE基站的很多参数可以随意修改定制，如频段、带宽、功率、增益等等。
- **可同时支持FDD和TDD两种制式。**两种LTE网络接入方式想要那种就哪种，甚至可在同一个基站中同时支持TDD和FDD，这可不是传统的基站可想象的了
- **既拥有商业网络的真实性同时满足开放软件可定义**

### 5G 无线系统关键技术

Cloud RAN

大规模MIMO

毫米波传输

高性能基带

### 威视锐解决方案



SoraSDR

基于x86服务器的高性能无线通信系统开发平台，提供开源SDK，面向LTE、WLAN的Cloud RAN原型开发



YunSDR

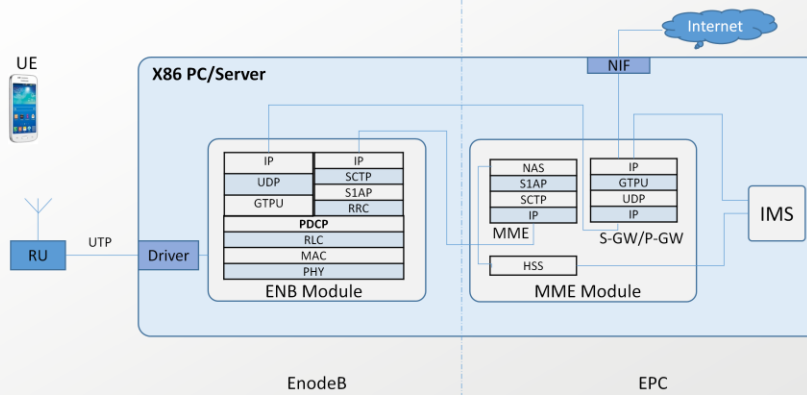
基于Zynq SoC的嵌入式实时通信开发平台，利用时钟同步技术，支持大规模MIMO和分布式MIMO的原型验证



IQXpert

面向60GHz毫米波频段无线通信开发平台，支持802.11ad和LTE毫米波系统，基带模拟带宽高达1.8GHz。

## ■ 系统架构



4

IMS(IP Multimedia Subsystem)即IP多媒体子系统

## ■ 基于软件无线电商用基站的系统参数

- LTE release 13 compliant
- 支持最多5CC CA
- 支持2\*2 MIMO, 在硬件支持的情况下理论上软件已可支持4\*2 MIMO
- 支持VoLTE高清语音、支持VoLTE视频
- 商用基站授权:
  - ✓ 分为ENB和MME两个部分, 可安装于一台计算机, 也可分开两台安装
  - ✓ 安装ENB模块的计算机, 建议采用主频>3.0Ghz CPU
- 支持开源ENB版本, 用户可以自由使用修改
- 对计算机的要求:
  - ✓ OS: Fedora 20~23 或 Ubuntu 14.04/16.04
  - ✓ i7 or xeon CPU, 内存建议8GB或以上, 硬盘建议500GB或以上

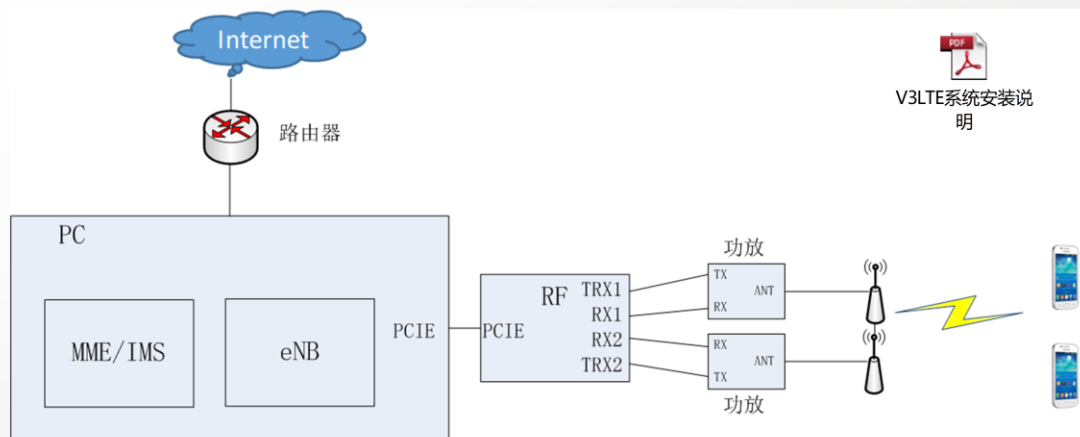
## ■ 基于软件无线电的NB-IOT基站

- 兼容NB-IoT release13
- 支持Single-tone和multi-tone类NB1 UE
- 支持15 kHz和3.75 kHz子载波间隔。
- 支持所有操作模式（带内，保护带和独立）。
- 可以在同一个eNodeB中同时使用多个NB-IoT和LTE小区。
- 支持多个覆盖级别
- 支持所有NPDCCH，NPDSCH，NPUSCH和NPRACH配置
- 支持控制平面CIoT优化
- 支持多DRB模式

## ■ 商业基站软件接口

- 系统连接
- 射频前端状态查看
- 配置文件查看与修改
- 运行MME、IMS、ENB
- 终端接入LTE100系统
- 日志查看
- Web UI使用

## Amari LTE100系统操作 ---系统连接



此示意图为MIMO2x2 + TDD功放的配置



## ■ Amari LTE100系统操作 ---射频前端信息

Linux终端下执行 `lspci -n` 查看是否识别到相应的pcie设备

Bash>lspci -n

```
00:00.0 0600: 8086:0100 (rev 09)
00:01.0 0604: 8086:0101 (rev 09)
00:02.0 0300: 8086:0102 (rev 09)
00:16.0 0780: 8086:1c3a (rev 04)
00:1a.0 0c03: 8086:1c2d (rev 05)
00:1b.0 0403: 8086:1c20 (rev 05)
00:1c.0 0604: 8086:1c10 (rev b5)
00:1c.1 0604: 8086:1c12 (rev b5)
00:1c.2 0604: 8086:1c14 (rev b5)
00:1c.3 0604: 8086:1c16 (rev b5)
00:1d.0 0c03: 8086:1c26 (rev 05)
00:1f.0 0601: 8086:1c4a (rev 05)
00:1f.2 0106: 8086:1c02 (rev 05)
00:1f.3 0c05: 8086:1c22 (rev 05)
01:00.0 0580: 10ee:7028
03:00.0 0c03: 1b21:1042
04:00.0 0c03: 1b21:1042
05:00.0 0200: 10ec:8168 (rev 06)
```

9

10ee:7028(7024)是射频前端的PCIE ID，如果没有识别到可以软重启，或者BIOS中检查相关设置

## ■ Amari LTE100系统操作 ---射频前端信息

在Linux终端执行如下命令查看PCIE板卡驱动是否安装成功

```
Bash>lsmod | grep riffa
```

```
riffa                36864  0
```

附:射频前端驱动手动安装

```
Bash>cd /opt/work/riffa/driver/linux
```

```
Bash>make
```

```
Bash>sudo make install
```

```
Bash>sudo modprobe riffa
```

10

如果驱动源码损毁可以由github手动下载

```
Bash>git clone https://github.com/v3best/riffa
```

不建议升级系统内核，每次升级内核都需要重新编译安装驱动，目前测试稳定版本的内核是4.10及以下版本

## ■ Amari LTE100系统操作 ---MME配置文件

### ➤LTE MME配置文件

/opt/work/2017-10-13/ltemme-linux-2017-10-13/config/1116-mme-ims.cfg

### ➤关键配置示例

#### log\_options:

"all.level=error,all.max\_size=0,nas.level=debug,nas.max\_size=1,slap.level=debug,slap.max\_size=1,rx.level=debug,rx.max\_size=1", log\_filename: "/tmp/mme.log",

gtp\_addr: "127.0.1.100",

plmn: "00101",

### ➤详细配置参考

/opt/work/2017-10-13/ltemme-linux-2017-10-13/doc/ltemme.pdf

11

所有配置文件都是基于json格式

<http://json.org/>

**gtp\_addr:** "127.0.1.100",

设置接收GTP-U数据包的IP地址（和一个可选端口），默认端口是2152。

通常是连接到核心网络的网络接口的IP地址。

当mme和enb位于同一台主机上时，gtp\_addr一般设置为本地local地址

当mme和enb位于不同主机上时，gtp\_addr一般设置为与enb所在主机连接的网  
路出口对应的网卡IP

## ■ Amari LTE100系统操作 ---IMS配置文件

### ➤LTE IMS配置文件

/opt/work/2017-10-13/ltemme-linux-2017-10-13/config/1116-ims.cfg

### ➤关键配置示例

**log\_options:**

"all.level=error,sip.level=debug,sip.max\_size=1,ims.level=debug,rx.level=debug,rx.max\_size=1,cx.level=debug,cx.max\_size=1",

**log\_filename:** "/tmp/ims.log",

### ➤详细配置参考

/opt/work/2017-10-13/ltemme-linux-2017-10-13/doc/lteims.pdf

/opt/work/2017-10-13/ltemme-linux-2017-10-13/doc/appnote\_ims.pdf

12

appnote\_ims.pdf volte测试文档

■ **Amari LTE100系统操作 ---用户数据配置文件**

➤ **LTE 用户数据配置文件**

/opt/work/2017-10-13/ltemme-linux-2017-10-13/config/1116-ue\_db-ims.cfg

➤ **详细配置参考**

/opt/work/2017-10-13/ltemme-linux-2017-10-13/doc/ltemme.pdf

## ■ Amari LTE100系统操作 ---用户数据配置文件

### ➤参考配置举例

```
ue_db: [ {  
    sim_algo: "milenege",  
    imsi: "001010000000001",  
    opc: "000102030405060708090A0B0C0D0E0F",  
    amf: 0x9001,  
    sqn: "000000000000",  
    K: "00112233445566778899AABBCCDDEEFF",  
    impu: ["sip:001010000000001", "tel:080000001"],  
    impi: "001010000000001@ims.mnc001.mcc001.3gppnetwork.org",  
    multi_sim: false,  
}, {  
    sim_algo: "milenege",  
    imsi: "001010000000002",  
    opc: "000102030405060708090A0B0C0D0E0F",  
    amf: 0x9001,  
    sqn: "000000000000",  
    K: "00112233445566778899AABBCCDDEEFF",  
    impu: ["sip:001010000000002", "tel:080000002"],  
    impi: "001010000000002@ims.mnc001.mcc001.3gppnetwork.org",  
    multi_sim: false,  
}, {
```

14

可以指定opc和op

Volte呼叫号码可以在impu里设置

Impi为imsi@domain

Domain要与1116-ims.cfg中domain一致，也可以在本组用户信息中单独通过domain:""单独指定

Multi\_sim 为true时可以允许多个具有相同sim卡信息的终端接入，以imei区分

Tel：用户号码，自定义

## ■ Amari LTE100系统操作 ---ENB配置文件1

### ➤LTE ENB配置文件

/opt/work/2017-10-13/ltebn-linux-2017-10-13/config/1116-enb-tdd.cfg

### ➤关键配置示例

```
log_options:
"all.level=error,all.max_size=0,nas.level=debug,nas.max_size=1,slap.level=debug,slap.max_size=1,x2ap.level=debug,x2ap.max_size=1,rrc.level=debug,rrc.max_size=1",
log_filename: "/tmp/enb0.log",
mme_addr: "127.0.1.100",
gtp_addr: "127.0.1.1",
plmn_list: [ "00101", ],
dl_earfcn: 38050, /* 2600 MHz (band 38) */
```

。 。 。 。

15

为了配置方便，这里enb的配置文件分为了四个配置文件

1116-enb-tdd.cfg          单天线不加功放

1116-enb-tdd-pa.cfg      单天线加功放

1116-enb-tdd-mimo.cfg    mimo不加功放

1116-enb-tdd-mimo-pa.cfg mimo加功放

dl\_earfcn在线计算工具

[http://niviuk.free.fr/lte\\_band.php](http://niviuk.free.fr/lte_band.php)

**mme\_addr:** "127.0.1.100",

对应mme配置文件中的gtp\_addr属性

**gtp\_addr:** "127.0.1.1",

eNB gtp绑定地址，如果eNB与MME不在同一台主机，那么必须设置为连接MME所在主机的本地网卡的IP

## ■ Amari LTE100系统操作 ---ENB配置文件2

### ➤关键配置示例

**n\_antenna\_dl:** 1, /\* number of DL antennas \*/

**n\_antenna\_ul:** 1, /\* number of UL antennas \*/

**n\_rb\_dl:** 25, /\* Bandwidth: 25: 5 MHz, 50: 10 MHz, 75: 15 MHz, 100: 20 MHz \*/

### ➤详细配置参考

/opt/work/2017-10-13/teenb-linux-2017-10-13/doc/teenb.pdf

16

为了配置方便，这里enb的配置文件分为了四个配置文件

1116-enb-tdd.cfg           单天线不加功放

1116-enb-tdd-pa.cfg       单天线加功放

1116-enb-tdd-mimo.cfg     mimo不加功放

1116-enb-tdd-mimo-pa.cfg  mimo加功放

Mimo2x2模式n\_antenna\_dl和n\_antenna\_ul都为2



## ■ Amari LTE100系统操作 ---射频配置文件

### ➤射频配置文件

/opt/work/2017-10-13/teenb-linux-2017-10-13/config/yunsdr/1chan.cfg

### ➤参考配置示例

```
rf_driver: {
    name:      "yunsdr",
    args:      "",
    auxdac1:   400,
    rx_antenna: "RX_ANTENNA_DISABLE",
    with_pa:   "DISABLE_PA",
    pa_switch: 500,
},
tx_gain: 80.0, /* TX gain (in dB) 0 to 89.8 dB */
rx_gain: 50.0, /* RX gain (in dB) 0 to 73 dB */
sample_rate: 11.52, /* 5M BW:7.68MHz,10M BW:11.52MHz,
                    20M BW:30.72MHz*/
```

17

auxdac1: 400,                      频偏校正  
rx\_antenna: "RX\_ANTENNA\_DISABLE", TRX端口的使能（射频板卡trx端口在RX\_ANTENNA\_ENABLE状态下即可以做接收端口又可以做发送端口）  
with\_pa: "DISABLE\_PA",              使能功放接口（使能功放后射频板卡trx端口接功放的tx端口，射频板卡rx端口接功放的rx端口，rx\_antenna选项就有板卡自动控制）  
pa\_switch: 500,                      TDD模式功放收发切换时间，以采样点为单位(可以结合采样率获知具体时间),可以保持默认值

采样率可以根据enb带宽设置

5M带宽： 7.68MHz采样率

10M带宽： 11.52MHz采样率

20M带宽： 30.72MHz采样率

## ■ Amari LTE100系统操作 ---运行

### ➤运行MME

```
Bash>cd /opt/work/2017-10-13/ltemme-linux-2017-10-13
```

```
Bash>sudo ./lte_init.sh [-e eth0]
```

```
Bash>sudo ./ltemme config/1116-mme-ims.cfg
```

### ➤运行IMS

```
Bash>cd /opt/work/2017-10-13/ltemme-linux-2017-10-13
```

```
Bash>sudo ./lte_init.sh [-e eth0]
```

```
Bash>sudo ./lteims config/1116-ims.cfg
```

### ➤运行ENB

```
Bash>cd /opt/work/2017-10-13/lteenb-linux-2017-10-13
```

```
Bash>sudo ./lte_init.sh
```

```
Bash>sudo ./lteenb config/1116-enb-tdd.cfg
```

18

```
Bash>sudo ./lte_init.sh [-e eth0]
```

[-e eth0] 代表为可选参数(实际命令输入时不带[]),

1116-enb-tdd.cfg          单天线不加功放

1116-enb-tdd-pa.cfg      单天线加功放

1116-enb-tdd-mimo.cfg    mimo不加功放

1116-enb-tdd-mimo-pa.cfg mimo加功放

为了运行方便, 这里针对不同配置提供了四个脚本文件

run\_enb.sh            单天线不加功放

run\_enb-pa.sh        单天线加功放

run\_enb-mimo.sh      mimo不加功放

run\_enb-mimo-pa.sh   mimo加功放

## Amari LTE100系统操作 ---终端接入1

➤ 扫描网络 → 开启数据漫游 → 开启volte高清通话 → 设置APN



19

有的型号终端在扫描网络后可能不会显示Amarisoft Network, 显示的网络名为Test plmn 1-1

网络名在mme的配置文件中可以配置  
network\_name: "Amarisoft Network",  
network\_short\_name: "Amarisoft",

## ■ Amari LTE100系统操作 ---终端接入2

### ➤APN设置

Name = Internet

APN = internet

APN type = internet,default

添加APN完成后选中此接入点，此接入点用于4G数据业务

Name = IMS

APN = ims

APN type = ims

此接入点用于VoLTE语音及视频通话，有的终端添加APN后可能不会显示出此接入点

### ➤参考文档

/opt/work/2017-10-13/Itemme-linux-2017-10-13/doc/appnote\_ims.pdf

20

终端成功鉴权接入网络后，添加相应APN就可以发起数据和语音(终端必须支持VoLTE业务)业务了

## ■ Amari LTE100系统操作 ---log查看1

➤MME log  
/tmp/mme.log

```
1 # ltemme version 2017-10-13, gcc 7.2.1
2 # Licensed to 'V3 TECH' [69a7d3098bdc75d5213c4b430d2293f4a57aa7f59fa7528580]
3 # Log file format:
4 # time layer dir [enb_id|mme_ue_id] message
5 14:22:10.846 [RX] - New connection from 127.0.1.100:44180
6 14:22:10.846 [RX] FROM 127.0.1.100:44180 Capabilities-Exchange-Request
7   Length = 200
8   Flags = 0x80 [ Request ]
9   Command code = 257 (Capabilities-Exchange)
10  Application-Id = 0 (Diameter common message)
11  Hop-by-hop Identifier = 3496929911
12  End-to-end Identifier = 4053006232
13  Origin-Host:
14    Code = 264
15    Flags = 0x40 [ Mandatory ]
16    Length = 41
17    Value = epc.mnc001.mcc001.3gppnetwork.org
18  Origin-Realm:
19    Code = 296
20    Flags = 0x40 [ Mandatory ]
21    Length = 37
22    Value = mnc001.mcc001.3gppnetwork.org
23  Host-IP-Address:
24    Code = 257
25    Flags = 0x40 [ Mandatory ]
26    Length = 14
27    Value = 127.0.1.100
28  Vendor-Id:
29    Code = 266
30    Flags = 0x40 [ Mandatory ]
31    Length = 12
```

## ■ Amari LTE100系统操作 ---log查看2

➤IMS log  
/tmp/ims.log

```
1 # lteims version 2017-10-13, gcc 7.2.1
2 # Licensed to 'V3 TECH' [098c10e7e107c4df02151d170f0216515c44000000adad00ce]
3 14:22:13.533 [CX] - Connecting to 127.0.1.100:3868
4 14:22:13.533 [RX] - Connecting to 127.0.1.100:3868
5 14:22:13.533 [RX] - Connected to 127.0.1.100:3868
6 14:22:13.533 [RX] TO 127.0.1.100:3868 Capabilities-Exchange-Request
7   Length = 180
8   Flags = 0x00 [ Request ]
9   Command code = 257 (Capabilities-Exchange)
10  Application-id = 0 (Diameter common message)
11  Hop-by-hop Identifier = 3300021563
12  End-to-end Identifier = 2574677228
13  Origin-Host:
14    Code = 264
15    Flags = 0x40 [ Mandatory ]
16    Length = 25
17    Value = ims.amarisoft.com
18  Origin-Realm:
19    Code = 296
20    Flags = 0x40 [ Mandatory ]
21    Length = 21
22    Value = amarisoft.com
23  Host-IP-Address:
24    Code = 257
25    Flags = 0x40 [ Mandatory ]
26    Length = 14
27    Value = 127.0.0.1
28  Vendor-Id:
29    Code = 266
```

## ■ Amari LTE100系统操作 ---log查看

### ➤ENB log

/tmp/enb0.log

```
1 # lteemb version 2017-10-13, gcc 7.2.1
2 # Licensed to 'V3 TECH' [69a7d3098bdc75d5213c4b430d2293f4a57aa7f59fa7528588]
3 # Log file format:
4 # time layer dir ue id [cell_id rnti sfn channel:] message
5 # Cell 0x02: earfcn=38850 pci=1 mode=TDD uldl_config=2 sp_config=0
6 # DL: n_rb_dl=50 cyclic_prefix=0
7 # UL: n_rb_ul=50 cyclic_prefix=0 prach_config_index=4 prach_freq_offset=4 delta_pucch_shift=2 n_rb_cqi=4 n_cs_an=0
8 # PUCCH allocation:
9 # Type      RBS      n
10 # 2/2a/2b   0      24
11 # Start of PUCCH ACK/NACK=12
12 # RBS reserved for PUCCH: 0 0 15 0 0 0 0 15 0 0
13 # SR resource count=48
14 # CQI resource count=192
15 # SRS resources: offsets=0 freqs=2 total=32
16 # GBR limits: DL=2.990 Mre/s UL=0.800 Mre/s
17 14:22:17.144 [SIAP] - Connecting to 127.0.1.100:36412
18 14:22:17.150 [SIAP] - Connected to 127.0.1.100:36412
19 14:22:17.150 [SIAP] TO 127.0.1.100:36412 S1 setup request
20   initiatingMessage: {
21     procedureCode id-S1Setup,
22     criticality reject,
23     value {
24       protocolIEs {
25         {
26           id id-Global-ENB-ID,
27           criticality reject,
28           value {
29             plmnIdentity '08F110' H,
30             enb-ID macroENB-ID: '1A2D0' H
31           }
32         },
33       }
34     }
35   }
```

23

## ■ Amari LTE100系统操作 ---WebUI

### ➤WebUI接口

Amarisoft LTE100 提供remote API用于访问MME、IMS、ENB, 通信协议使用WebSocket RFC 6455 (<https://tools.ietf.org/html/rfc6455>).

### ➤示例nodejs程序

/opt/work/2017-10-13/lteenb-linux-2017-10-13/doc/ws.js

### ➤示例程序运行

./ws.js 127.0.0.1:9000 '{"message": "config\_get"}'

### ➤参考文档

/opt/work/2017-10-13/lteenb-linux-2017-10-13/doc/ltemme.pdf



# openairinterface5G

## ■ OAI eNB

- git clone <https://gitlab.eurecom.fr/oai/openairinterface5g.git>
- cd openairinterface5g
- git checkout 17b9a9e917ce2a3a8c7004c7b9a221c350ddfe17
- git apply ../
- source oaienv
- ./cmake\_targets/build\_oai -I # install SW packages from internet
- ./cmake\_targets/build\_oai -w YUNSDR --eNB # compile eNB

## ■ OAI EPC

- `git clone https://gitlab.eurecom.fr/oai/openair-cn.git`
- `cd openair-cn`
- `git checkout 724542d0b59797b010af8c5df15af7f669c1e838`
- `git apply ../EPC.patch`

## ■ OAI EPC –安装第三方组件

- cd openair-cn
- source oaienv
- cd scripts
- ./build\_hss -l

设置MySQL 密码

freeDiameter 1.2.0 :yes

phpmyadmin:

选择light 或者 apache

Configure database for phpmyadmin with dbconfig-common: yes

password: 建议与MySQL 一致

## ■ OAI EPC –安装第三方组件

➤ ./build\_mme -i

安装freeDiameter 1.2.0: no

安装asn1c rev 1516 patched? <y/N>: no

安装libgtpnl ? <y/N>: yes

wireshark permissions: 按需配置

➤ ./build\_spgw -i

安装libgtpnl ? <y/N>: no

## ■ OAI EPC –编译EPC组件

- cd openair-cn
- source oaienv
- cd scripts
- ./build\_hss
- ./build\_mme
- ./build\_spgw

#### ■ OAI EPC –组件网络配置

HSS is on localhost: 127.0.0.1

eNB is on 127.0.0.10

MME is on 127.0.0.20

SPGW is on 127.0.0.30

realm: OpenAir5G.Alliance,

FQDN: hss.OpenAir5G.Alliance, mme.OpenAir5G.Alliance

## ■ OAI EPC –配置EPC

```
$sudo mkdir -p /usr/local/etc/oai  
$sudo cp -rp ../config_epc/* /usr/local/etc/oai  
$cd openair-cn  
$source oaienv  
$cd scripts  
$./check_hss_s6a_certificate /usr/local/etc/oai/freeDiameter hss.OpenAir5G.Alliance  
$./check_mme_s6a_certificate /usr/local/etc/oai/freeDiameter  
mme.OpenAir5G.Alliance
```

Sgi出口配置(选择连接Internet的网络接口)

```
/usr/local/etc/oai/spgw.conf:  
PGW_INTERFACE_NAME_FOR_SGI = "enp3s0";  
PGW_MASQUERADE_SGI = "yes";
```



#### ■ OAI EPC –HSS和SIM卡配置

MCC="208" ; MNC="92"; TAC = "1";

SIM信息按需配置，并导入到HSS的数据库

## ■ OAI EPC –运行EPC

### **Bash1:**

```
$cd openair-cn  
$source oaienv  
$cd scripts  
$./run_hss
```

### **Bash2:**

```
$cd openair-cn  
$source oaienv  
$cd scripts  
$./run_mme
```

### **Bash3:**

```
$cd openair-cn  
$source oaienv  
$cd scripts  
$sudo -E ./run_spgw
```

## ■ OAI EPC –运行eNB

```
$sudo bash  
$cd openairinterface5g  
$source oaienv  
$./cmake_targets/lte_build_oai/build/lte-softmodem -O <your config file>
```

35

如果UE接入后无法上网，可以先检查UE的APN设置  
然后在检查ubuntu的iptables转发策略  
sudo iptables --list-rules FORWARD  
可以追加一条转发策略  
sudo iptables -A FORWARD -j ACCEPT

# srsLTE

36

如果UE接入后无法上网，可以先检查UE的APN设置  
然后在检查ubuntu的iptables转发策略  
`sudo iptables --list-rules FORWARD`  
可以追加一条转发策略  
`sudo iptables -A FORWARD -j ACCEPT`

## ■ srsLTE -简介1

srsLTE 是由 SRS 开发的一款 SDR UE 和 eNodeB 的免费且开源的 LTE 库。该库采用最低的内置模块和外部依赖方式实现了高度模块化。它完全是由 C 语言编写。

兼容 LTE 第 8 发布版本;

✓FDD 配置;

✓测试带宽: 1.4、3、5、10、15 和 20 MHz;

✓传输方式 1 (单天线) 和 2 (发射分集);

✓UE 端的小区搜索和同步过程;

✓UE 和 eNodeB 端支持所有 DL 频道/信号: PSS、SSS、PBCH、PCFICH、PHICH、PDCCH、PDSCH;

✓UE 端支持所有 UL 频道/信号: PRACH、PUSCH、PUCCH、SRS;

✓基于频率 ZF 和 MMSE 均衡器;

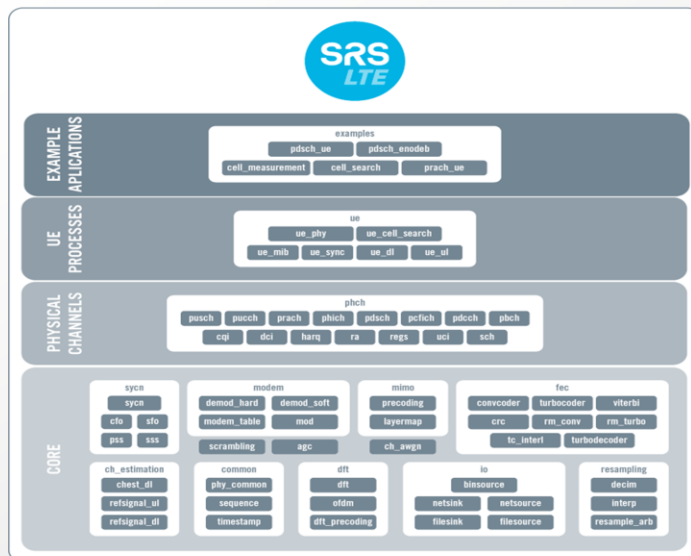
✓可用于 Intel SSE4.1/AVX (+100 Mbps) 和 C 标准 (+25Mbps) 的高度优化的 Turbo 解码器;

✓。。。

37

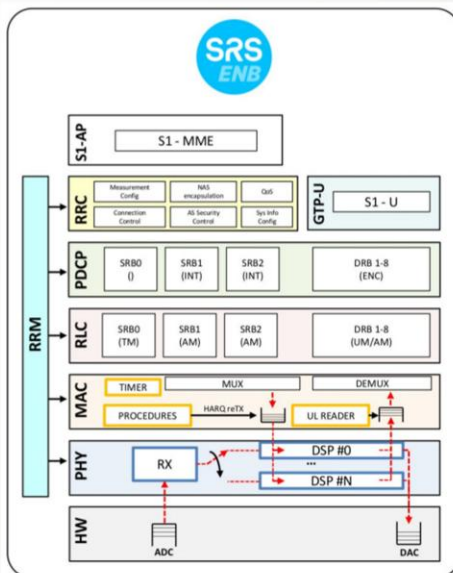
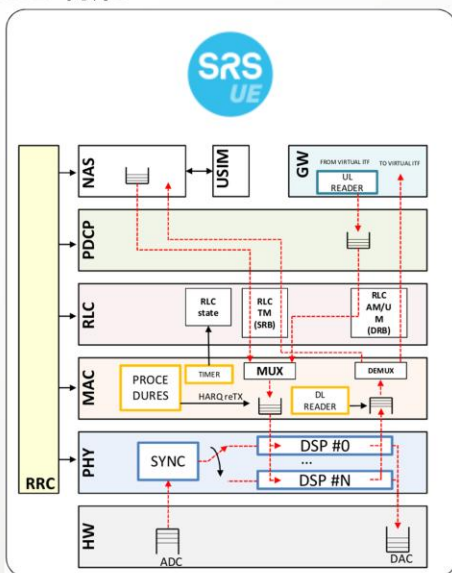
<http://www.softwareradiosystems.com/products/#srslte>

## ■ srsLTE -简介2



38

## ■ srsLTE -简介3



#### ■ srsLTE –准备 安装srsgui

```
$sudo apt-get install libboost-system-dev libboost-test-dev libboost-thread-dev  
libqwt-dev libqt4-dev  
$git clone https://github.com/srslte/srsgui  
$cd srsgui  
$mkdir build  
$cd build  
$cmake ../  
$make  
$sudo make install  
$sudo ldconfig
```



#### ■ srsENB –准备 安装volk

```
$git clone https://github.com/gnuradio/volk  
$cd volk  
$mkdir build  
$cd build  
$cmake ../  
$make  
$make test  
$sudo make install  
$sudo ldconfig
```

### ■ srsENB –编译运行

```
$sudo apt-get install cmake libfftw3-dev libmbedtls-dev libboost-all-dev libconfig++-dev libsctp-dev
```

```
$git clone https://github.com/srsLTE/srsLTE.git
```

```
$cd srsLTE
```

```
$git checkout 74b9159bdc42391f8f7489dc01e1510c26165018
```

```
$git apply ../v3_pcie.patch
```

```
$mkdir build
```

```
$cmake ../
```

```
$make -j8
```

```
$cd ../srsenb/
```

```
$cp drb.conf.example drb.conf
```

```
$cp enb.conf.example enb.conf
```

```
$cp rr.conf.example rr.conf
```

```
$cp sib.conf.example sib.conf
```

```
$sudo ../build/srsenb/src/srsenb enb.conf
```

42

### ■ srsENB –eNB配置文件enb.conf

```
# eNB configuration
# enb_id:      20-bit eNB identifier.
# cell_id:     8-bit cell identifier.
# tac:         16-bit Tracking Area Code.
# mcc:         Mobile Country Code
# mnc:         Mobile Network Code
# mme_addr:    IP address of MME for S1 connection
# gtp_bind_addr: Local IP address to bind for GTP connection
# n_prb:       Number of Physical Resource Blocks (6,15,25,50,75,100)
[enb]
enb_id = 0x19B
cell_id = 0x01
phy_cell_id = 1
tac = 0x0001
mcc = 001
mnc = 01
mme_addr = 127.0.1.100
gtp_bind_addr = 127.0.1.1
n_prb = 25
```

43

mme\_addr:MME所在主机IP,  
gtp\_bind\_addr：如果eNB与MME不在同一台主机，那么必须设置为连接MME所在主机的本地网卡的IP

### ■ srsENB –eNB配置文件enb.conf

```
# RF configuration
# dl_eearfcn: EARFCN code for DL
# tx_gain: Transmit gain (dB).
# rx_gain: Optional receive gain (dB). If disabled, AGC if enabled
#
# Optional parameters:
# device_name: Device driver family. Supported options: "auto" (uses first found), "UHD" or
# "bladeRF" or "yunsdr"
[rf]
dl_eearfcn = 1650 # Band3
tx_gain = 80
rx_gain = 65

device_name = yunsdr
#device_args = auto
#time_adv_nsamples = auto
#burst_preamble_us = auto
```

44

device\_name = yunsdr  
Yunsdr指的是PCIE的板卡

**tx\_gain:** TX gain (in dB) 0 to 89.8 dB

**rx\_gain:** RX gain (in dB) 0 to 73 dB

## ■ srsENB---终端接入 with Amarisoft MME

### ➤APN设置

Name = Internet

APN = internet

APN type = 未设置

添加APN完成后选中此接入点，此接入点用于4G数据业务

45

APN type不要设置，否则srsenb程序在建立承载时会崩溃

## ■ srsUE---运行

### ➤ 运行

```
$sudo ../build/srsue/src/srsue ue.conf
```

### ➤ 配置

```
# RF configuration
```

```
#
```

```
# dl_earfcn: Downlink EARFCN code.
```

```
# freq_offset: Uplink and Downlink optional frequency offset (in Hz)
```

```
# tx_gain: Transmit gain (dB).
```

```
# rx_gain: Optional receive gain (dB). If disabled, AGC if enabled
```

46

目前版本yunsdr设备不支持ue功能，所以使用b210

#### ■ srsUE---配置

```
[rf]
dl_earfcn = 1650
freq_offset = 0
tx_gain = 80
rx_gain = 70
device_name = UHD
```

47

freq\_offset：ue与enb的频偏校正，可在srsue运行扫描小区是的频偏估计值CFO获知

#### ■ srsUE---虚拟USIM配置

```
# USIM configuration
# algo: Authentication algorithm (xor/milenage)
# op: 128-bit Operator Variant Algorithm Configuration Field (hex)
# amf: 16-bit Authentication Management Field (hex)
# k: 128-bit subscriber key (hex)
# imsi: 15 digit International Mobile Subscriber Identity
# imei: 15 digit International Mobile Station Equipment Identity
[usim]
algo = milenage
op = 63BFA50EE6523365FF14C1F45F88737D
amf = 8000
k = 00112233445566778899aabbccddeeff
imsi = 001010123456789
imei = 353490069873319
```

48

虚拟USIM卡信息配置，与MME的HSS数据库中一致才能接入





过滤S1AP协议，提取非接入层信令

## 典型客户

 **1000+**  
行业客户

 **200+**  
大学和研究机构

 **20+**  
国家和地区

**Microsoft**

**Duke**  
UNIVERSITY

 **北京大学**

 **清华大学**  
Tsinghua University

 **北京邮电大学**  
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

**SONY**



 **复旦大学**  
FUDAN UNIVERSITY



 **北京交通大学**

**FUJITSU**

 **UNIVERSITY OF CAMBRIDGE**

 **北京工业大学**

**ZTE中兴**



 **浙江大学**  
Zhejiang University

**NOKIA**  
Connecting People



## 更多信息

威视锐: <http://www.v3t.com.cn>  
V3学院: <http://www.v3edu.org>

## ■ 基于通用SDR的LTE系统应用场景——芯片开发集成测试验证

- 基于LTE软基站的灵活、低成本、升级简便等特性，Amari LTE100产品已经开始被芯片开发厂商用于LTE终端芯片开发的集成测试验证系统
- 通过运行于x86上的软基站，可以非常方便的为测试模块提供LTE基带信号，而且基带信号可以软件配置的方式进行动态的调整
- Spreadtrum公司已经在Synopsys的2017年初的会议上发布了他们的相应解决方案

