

ТЕХНИЧЕСКОЕ РУКОВОДСТВО

CCS_LOGGER

(с) Parabel
Версия 1.2

ОГЛАВЛЕНИЕ

1. Введение.....	2
2. Структура программного обеспечения	3
3. Установка программного обеспечения.....	4
4. Запуск программы, командная строка	6
5. Лицензия, процедура запроса лицензии.....	7
6. Конфигурация.....	8
6.1. INI ФАЙЛ.....	8
6.2. Секция common	8
6.3. Секция log	8
6.4. Секция linkset.....	9
6.5. Секция db	10
7. Подключение портов E1 адаптера к линиям E1.	11
7.1. Подключение к одной линии E1	12
7.2. Подключение к двум линиям E1 с отдельными каналами сигнализации	13
7.3. Подключение к двум линиям E1 с общим каналом сигнализации	14
7.4. Подключение к четырем линиям E1 с общим и резервным каналом сигнализации	15
8. Форматы выходных файлов	16
8.1. Метафайлы	16
8.2. Аудио файлы.....	17
8.3. База данных	18
8.4. Log файл	18

1. ВВЕДЕНИЕ

Назначение программы ccs_logger¹

Продукт предназначен для работы в составе программно-аппаратного комплекса, осуществляющего запись информации с линий E1, разбор протоколов общеканальной сигнализации SS7 и PRI, архивацию медиа сообщений.

Областями применения могут быть:

- диагностика сетей связи, выявление и локализация сбоев в сигнализации, проблем с настройкой телекоммуникационного оборудования;
- сбор и анализ статистики соединений, выявление нагруженных линий;

База сообщений, записанная с помощью ccs_logger, может быть использована совместно с программами ИИ, для оказания интеллектуальных услуг связи, например:

- анализ качества работы операторов в call-центрах
- анализ и выявление конфликтных ситуаций
- поиск диалогов по ключевым словам

Вывод информации

В результате работы ПО ccs_logger создаются следующие файлы:

- база данных SQLite с информацией о зафиксированных соединениях;
- файлы с аудио данными формата wav;
- лог файл с расшифровкой протокола обмена;

Форматы выходных файлов описаны на стр. 16

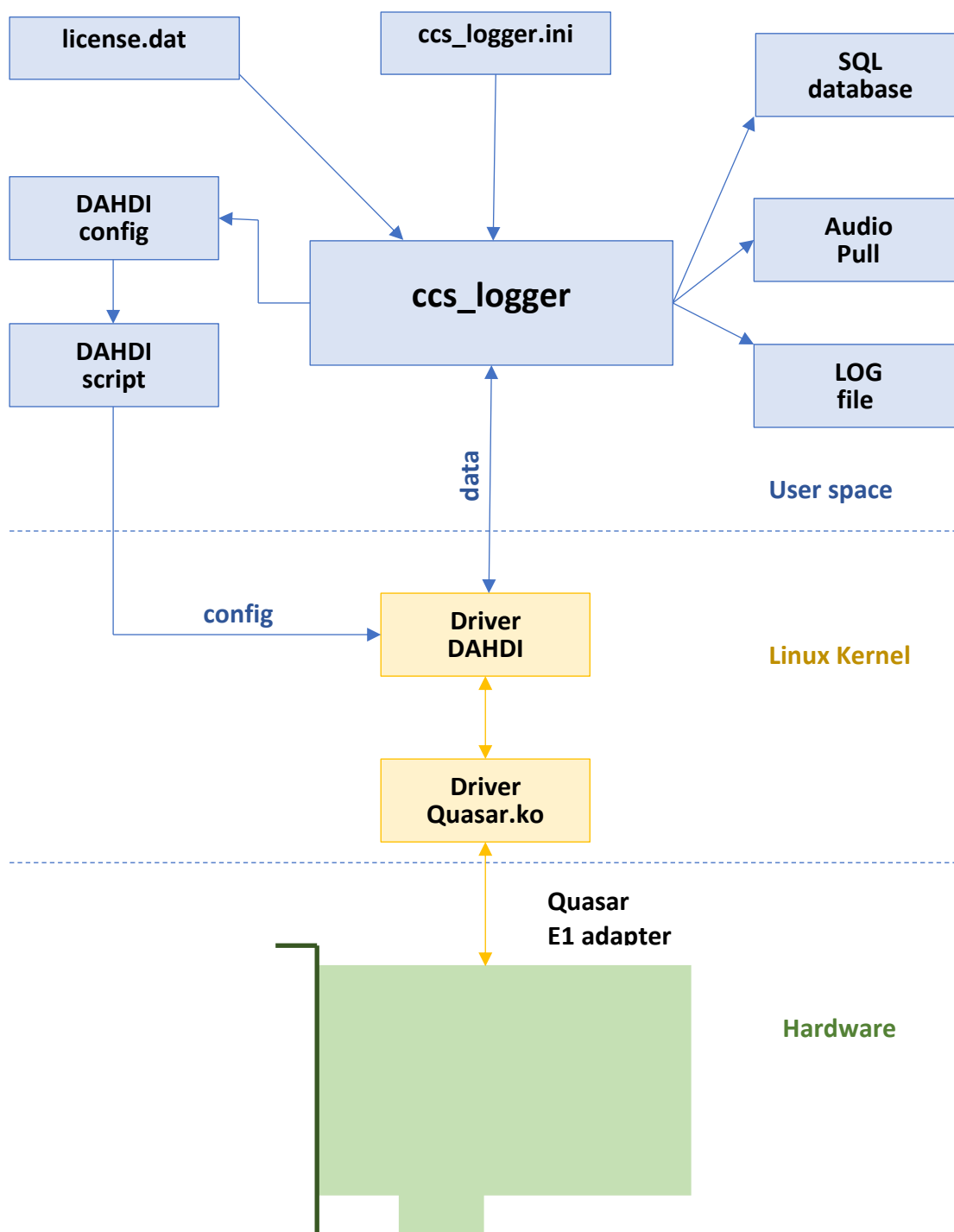
Необходимые условия для инсталляции

- компьютер с установленным адаптером Quasar
- развернутый дистрибутив ОС Linux x64
- пакет DAHDI с драйвером Quasar
- лицензия ccs_logger.
- стандартные средства компиляции Linux и заголовочные файлы

Процесс инсталляции ПО описан на стр. 4. Процесс получения лицензии описан на стр. 7. Подключение адаптера Quasar к линиям E1 описано на стр. 11.

¹ По определению N308-ФЗ от 2 августа 2019 г, программное обеспечение (ПО) ccs_logger не является специальным техническим средством, предназначенным для негласного получения информации. Соответственно, производство и распространение данного ПО не подлежит лицензированию по 99-ФЗ "О лицензировании отдельных видов деятельности".

2. СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Выходными файлами для `ccs_logger` являются база данных (`SQL database`), директория с аудио данными (`Audio pull`), файл с историей соединений (`LOG file`). Конфигурация программы задается в файле `ccs_logger.ini`. На основе этой конфигурации автоматически создается конфигурация драйвера `DAHDI config`, которая скриптом `DAHDI script` загружается при старте программы.

3. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Поэтапно, для установки ПО **ccs_logger**, необходимо сделать следующее:

1. Установить пакет DAHDI с драйвером quasar. Готовая сборка находится на сайте производителя по ссылке <http://parabel.ru/download/>. Пакет представляет собой архив bz2 с исходными файлами, который необходимо распаковать и запустить скрипт **build.sh**, который выполнит компиляцию драйверов и утилит. Для успешной компиляции, предварительно необходимо установить следующие компоненты из репозитория Linux:

gcc - компилятор той же версии, которой собрано ядро ОС.

patch - утилита

linux-headers-XXX - заголовочные файлы ядра

linux-kbuild-XXX - скрипты сборки ядра

Кроме, того, в системе должен быть установлен linux драйвер **crc_ccitt**. Проверить его наличие в системе можно командами

```
lsmod | grep crc
```

Если драйвер не загружен, нужно проверить возможность его загрузки:

```
modprobe crc_ccitt
```

Так как программа **ccs_logger** работает по абсолютной ссылке на пакет **DAHDI**, нет необходимости устанавливать пакет в систему, скрипт **install.sh** не должен запускаться. Устанавливать **DAHDI** из стандартного репозитория также не следует, чтобы не получить конфликт версий.

2. Распаковать архив **ccs_logger** в рабочую директорию. В файле **dahdiStart.sh** отредактировать переменную **DAHDI_ROOT**, которая должна указывать на директорию с пакетом **DAHDI**.

При запуске скрипт **dahdiStart.sh** должен успешно загружать **DAHDI**, что можно проверить командой:

```
# lsmod | grep dahdi
```

```
dahdi          249856  1 quasar
```

```
crc_ccitt      16384  1 dahdi
```

Если это не так, возможно несколько вариантов:

- не установлен адаптер quasar
- указан неправильный путь к директории с пакетом **DAHDI**
- не скомпилировался **DAHDI**
- были взяты заголовочные файлы **linux-headers** от другого ядра

3. Установить библиотеку работы с ODBC из дистрибутива Linux

На примере Debian:

```
apt-get install unixodbc unixodbc-dev
```

ODBC является интерфейсом для доступа к сетевой или файловой SQL. В зависимости от базы данных нужно установить соответствующий драйвер.

Для базы **sqlite3** установить пакеты:

```
apt-get install sqlite3 libsqliteodbc
```

Для базы **postgre**

```
apt-get install odbc-postgresql
```

После установки драйверов БД, в файле **/etc/odbcinst.ini** будут находиться актуальные ссылки на библиотеки в вашей системе. Скопируйте файл **odbcinst.ini** в директорию с программой **ccs_logger**.

Локальный файл **odbc.ini** отредактировать в соответствии с содержимым **odbcinst** и атрибутами имеющейся базы данных:

Driver - указать имя драйвера из файла **odbcinst**

Database - указать путь к базе данных (для **sqlite**) или имя сетевой БД

UserName – указать имя пользователя сетевой БД

Password – пароль для доступа к БД

Port - порт tcp сетевой БД

Servename – адрес IP или сетевое имя сервера БД

Аналогичным образом может быть настроен доступ к большинству популярных БД - Microsoft Access, MySQL, Oracle, Microsoft SQL Server и т.д.

Обратите внимание, программа **ccs_logger** берет актуальные файлы **odbc.ini** и **odbcinst.ini** из локальной директории, где была запущена программа. Общесистемные файлы **/etc/odbc.ini** и **/etc/odbcinst.ini** игнорируются.

4. Если необходимо, можно дополнительно настроить **ccs_logger**, поменяв в **ini** файле пути по умолчанию на конфигурационный файл **dahdi.conf**, на скрипт запуска **dahdi**, имя базы данных соединений, путь к директории с аудио файлами.

5. Если имеется файл **license.dat**, поместите его в рабочую директорию **ccs_logger**, в противном случае получите лицензию по процедуре, описанной на стр.7.

6. В соответствии с подключением линий E1, сконфигурируйте секции **linkset** в файле **ini**, как описано на стр. 9.

7. Запустите программу на исполнение:

```
#!/ccs_logger -s
```

4. ЗАПУСК ПРОГРАММЫ, КОМАНДНАЯ СТРОКА

Запуск **ccs_logger** без параметров выводит краткую справку, программа завершается:

ccs_logger [options]

-l {file} generate file with license request, exit

-s start recording

-v print version (build number)

-h help

Запуск с ключом **-l** создает файл запроса лицензии, программа завершается. Особенности лицензирования описаны в соответствующем параграфе.

Запуск с ключом **-v** (без параметров) выводит версию программы (номер сборки).

Запуск с ключом **-s** (без параметров) переводит программу в основной рабочий режим, запись информации и сохранение ее в файлы осуществляется в соответствии с конфигурацией **ccs_logger.ini**.

Закончить работу программы можно нажатием **Ctrl-C**.

Нажатие любой клавиши приводит к выводу статистики работы:

```
### Linkset 0 ###
```

```
Counters:
```

```
Setup 5, Progress 5, Answered 4, Release 5, Release complete 5
```

```
Errors:
```

```
Format 0, Proto 1, Fops 0, Queue 0
```

В счетчиках статистики фиксируется количество событий с момента запуска программы:

Setup – протокольный пакет Setup (Q.931) или Initial address (SS7)

Progress – протокольный пакет Call proceed (Q.931) или Call progress (SS7)

Answered – протокольный пакет Connect ack (Q.931) или Answer (SS7)

Release – протокольный пакет Release (Q.931 или SS7)

Release complete – протокольный пакет Release complete (Q.931 или SS7)

Счетчики ошибок фиксируют события:

Format – ошибка формата, например, вышедший за границы параметр или неожиданный конец пакета

Proto – ошибка протокола, например, неподдерживаемый параметр или пакет

Fops – ошибка ввода-вывода в файл

Queue – переполнение внутренней очереди, скорее всего приводит к потере пакета

5. ЛИЦЕНЗИЯ, ПРОЦЕДУРА ЗАПРОСА ЛИЦЕНЗИИ

Лицензия дает право использовать один экземпляр программы `ccs_logger` в фиксированном аппаратном окружении. Под инсталляцией будем понимать установленные в одном сервере один или несколько адаптеров E1 серии Quasar, программу `ccs_logger` и лицензию для нее.

Лицензия выдается производителем на конкретную инсталляцию программы, работающую с одним или несколькими зарегистрированными адаптерами Quasar.

При этом верно следующее:

- лицензия выдается на работу `ccs_logger` с конкретными экземплярами адаптеров Quasar, с определенными серийными номерами.
- не допускается добавление в инсталляцию незарегистрированных адаптеров Quasar, в этом случае лицензия должна быть получена заново по процедуре запроса.
- в рамках текущей лицензии допускается удаление одного (нескольких) адаптеров из инсталляции. По крайней мере один зарегистрированный адаптер Quasar должен оставаться в системе.
- допускается перенос инсталляции с одного сервера на другой в рамках одной лицензии.

Процедура запроса лицензии.

Производитель может поставлять ПО `ccs_logger` как с готовой инсталляцией, так и отдельно. В последнем случае лицензию необходимо получить, отправив письмо с запросом на адрес технической поддержки производителя. Письмо должно содержать бинарный файл запроса лицензии, который генерируется программой `ccs_logger` при запуске с ключом `-l`:

```
ccs_logger -l {file}
```

где `file` – имя выходного файла, которое может быть произвольное.

Например,

```
ccs_logger -l a.dat
```

запишет в текущую директорию бинарный файл `a.dat` с запросом лицензии.

Программа может завершиться ошибкой:

```
No quasar adapter detected or no driver
```

Это будет означать, что программа не нашла ни одного установленного в сервер адаптера Quasar. Причиной может быть не только физическое отсутствие адаптеров, но и отсутствие загруженного драйвера к нему. В этом случае необходимо установить весь комплект программного обеспечения и запустить `ccs_logger` заново.

Файл с лицензией `license.dat` высылается производителем и должен быть скопирован в рабочую директорию, откуда запускается сама программа `ccs_logger`. При запуске программа проверит, соответствует ли инсталляция лицензии, и разрешит дальнейшее функционирование.

6. КОНФИГУРАЦИЯ

6.1. INI ФАЙЛ

Конфигурация описывается пользователем в файле **ccs_logger.ini** в формате стандартного ini файла и сводится к заданию значений переменным в виде

Parameter=value

где **Parameter** – имя переменной, **value** - значение переменной.

Содержимое ini файла разбито на секции:

[common] – пути к рабочим файлам и директориям

[log] – управление выводом в log файл

[linkset N] – описание группы E1 с номером N

Конфигурация считывается из ini файла один раз при старте программы. Для применения изменений конфигурации нужно перезапустить **ccs_logger**.

В каждой строке могут быть добавлены комментарии, после символа «;».

6.2. СЕКЦИЯ COMMON

В секции описываются пути к рабочим файлам и директориям.

dahdiConf = xxx - имя файла с конфигурацией драйвера dahdi. Имя может содержать путь. Файл создается автоматически при старте программы и затем используется скриптом, заданным в параметре dahdiScript.

DahdiScript = xxx – имя скрипта, который перезагружает драйвер dahdi и конфигурирует его. Скрипт создается пользователем и должен иметь права доступа на исполнение. Можно использовать скрипт dahdiStart.sh, идущий в комплекте с программой.

audioPath=xxx – путь к директории с аудиофайлами. Директория должна быть создана пользователем и иметь права доступа на запись. Путь к директории может быть как абсолютным, так и относительным.

saveAudio=1 – записывать аудиофайлы, =0 – не записывать

saveMeta = 1 – записывать метафайлы, = 0 – не записывать

startRecord = setup – начинать запись аудио файла сразу после начала соединения

startRecord = answer – начинать запись аудио файла после получения ответа абонента

6.3. СЕКЦИЯ LOG

timestamp=1 - печатать метку времени с каждым сообщением в лог. Метка будет иметь вид

2020-12-03 17:59:43.22>

timestamp=0 – метка времени не печатается.

info=1 – печатать информационные сообщения

warn=1 печатать предупреждения

lev2=1 печатать сообщения второго уровня протокола. В случае PRI это Q.921, в случае протокола SS7 это MTP2.

lev3=1 печатать сообщения третьего уровня протокола.

Если `info/warn/lev2/lev3 = 0`, то соответствующий тип сообщений не печатается в лог.

console=1 – в консоль будет выводиться копия сообщений, попавших в лог-файл.

filename=xxx – имя лог-файла. Если файл не существует, он будет создан. Существующий файл будет дополняться с конца.

6.4. СЕКЦИЯ LINKSET

Секция **linkset** описывает группу целевых E1 линий, мониторинг которых осуществляется с помощью оборудования Quasar. Группа может включать один или несколько (до 4) линий E1, управляемых общим сигнальным каналом. К целевым E1 линиям подключены порты адаптера Quasar.

В заголовке секции, через пробел, должен указываться номер группы, например

[linkset 3]

Может быть описано несколько групп, максимальное количество определяется лицензией.

В секции описываются следующие параметры.

numslots = xxx – количество таймслотов в группе.

Параметр может принимать фиксированные значения - 32,64,96,128, что соответствует от 1 до 4 подключенных к мониторингу линий, что, в свою очередь, соответствует от 2 до 8 портов на адаптере Quasar (с учетом двух направлений – rx, tx). Подробнее схему подключения см на стр. 11.

dslot0 = xxx – номер таймслота с основным каналом сигнализации.

Нумерация таймслотов осуществляется от нуля, каждому E1 соответствует 32 таймслота. Таймслотам 0,32,64 и 96 соответствует разметочный канал E1, содержащий метку начала фрейма. Нумерация сквозная от нулевого таймслота первой линии E1 в группе. Например, если сигнализация идет по 17 таймслоту третьей линии E1, то нужно указать `dslot0=81`.

dslot1 = xxx – номер таймслота с резервным каналом сигнализации.

Правила нумерации такие же как для `dslot0`. Параметр не обязательный – если резервного канала нет, строку `dslot1` можно закомментировать.

proto = xxx – протокол общеканальной сигнализации

Может быть два варианта `pri` или `ss7`.

law = xxx

Может принимать значения **a** или **m**. Параметр используется только для конфигурирования драйвера DAHDI.

ifid0=xxx

ifid1=xxx

ifid2=xxx

ifid3=xxx – номер интерфейса E1 в группе, параметр относится **только** к PRI протоколу. Ifid0 соответствует младшему E1 порту в группе, ifid3 – старшему порту в группе.

Значение параметра соответствует номеру интерфейса в протоколе Q.931 (PRI). Номер интерфейса назначается административно при конфигурации внешней АТС и может принимать значения 0..127. На одиночных E1 линиях чаще всего используется указание интерфейса «по умолчанию», в этом случае параметры ifid нужно не указывать. Если группа содержит два E1 на общем сигнальном канале, указываются только параметры ifd0, ifd1, остальные игнорируются. Если группа содержит три E1, нужно указать ifid0, ifid1, ifid2. Соответственно, для группы из 4-х E1 нужно указать ifid0..ifid3.

6.5. СЕКЦИЯ DB

Секция db содержит параметры:

dbName = xxx

где xxx – имя базы данных, заданное в файле odbc.ini

saveDb = 1 – сохранять информацию в базу данных, =0 – не сохранять

Если saveDb = 1 и при записи в БД произошла ошибка, ccs_logger завершается.

7. ПОДКЛЮЧЕНИЕ ПОРТОВ E1 АДАПТЕРА К ЛИНИЯМ E1.

Для корректной работы программы `ccs_logger`, порты адаптеров Quasar должны быть подключены к линиям E1 с соблюдением определенных правил. В данном параграфе рассмотрим несколько схем:

- подключение к одной линии E1
- подключение к двум линиям E1 с отдельными каналами сигнализации
- подключение к двум линиям E1 с общим каналом сигнализации
- подключение к четырем линиям E1 с общим каналом сигнализации и резервным каналом

Под **линией E1** далее будем понимать линию, соединяющую коммутационное оборудование PBX1 и PBX2, с которой необходимо записать информацию. Физически линия состоит из двух витых пар, передающих информацию в двух направлениях. Направление PBX1 -> PBX2 обозначим TX, направление PBX2 -> PBX1 обозначим RX. На рисунках одну витую пару будем обозначать одной графической линией.

Под **портом E1** будем понимать порт адаптера Quasar, установленного в систему под управлением программы `ccs_logger`. Порт имеет один передатчик и один приемник. Для съема информации с линии E1 передатчик порта не используется, но необходимо два приемника для направлений RX, TX. Соответственно, на одну линию E1 приходится два порта адаптера Quasar.

В `ini` файле программы `ccs_logger` описывается конфигурация **линий E1** (секция `linkset`), при этом подразумевается, что подключение этих линий к портам будет следовать правилам:

- если `linkset` описывает одну линию, порт P адаптера соответствует RX, порт P+1 соответствует TX;
- если `linkset` описывает n линий (с общим сигнальным каналом), для подключения используется 2*n портов, причем к нечетным портам P1, P3.. Pn подключается направление RX (RX1, RX2 ... RXn), к четным портам P2, P4, Pn+1 подключается направление TX (TX1, TX2 .. TXn).
- `linkset` с большим номером соответствует портам с большим номером.

Информация о портах, как их предполагает использовать программа, выводится при старте в консоли и в `log`-файле, например:

```
LINKSET 0 (Rx) ->PORT 1
```

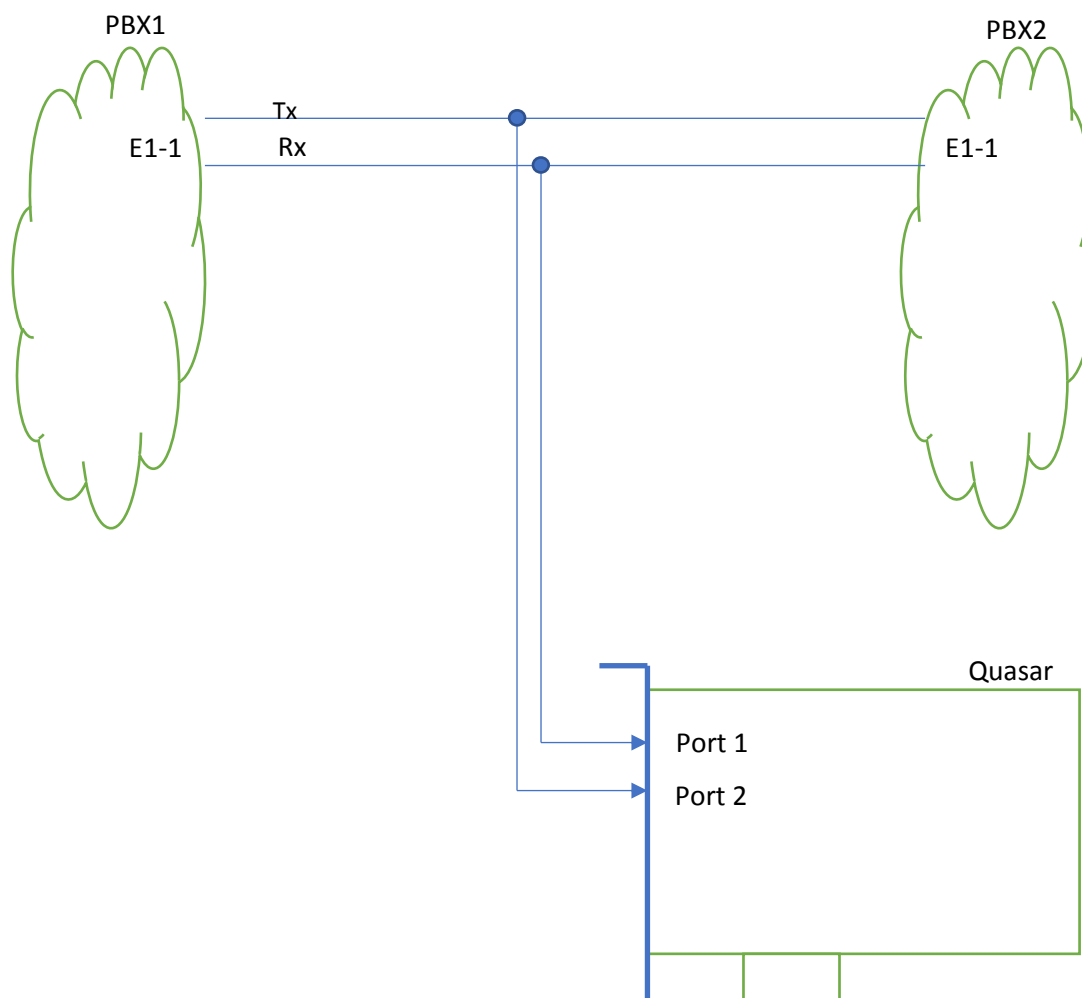
```
LINKSET 0 (Tx) ->PORT 2
```

```
LINKSET 0 (Rx) ->PORT 3
```

```
LINKSET 0 (Tx) ->PORT 4
```

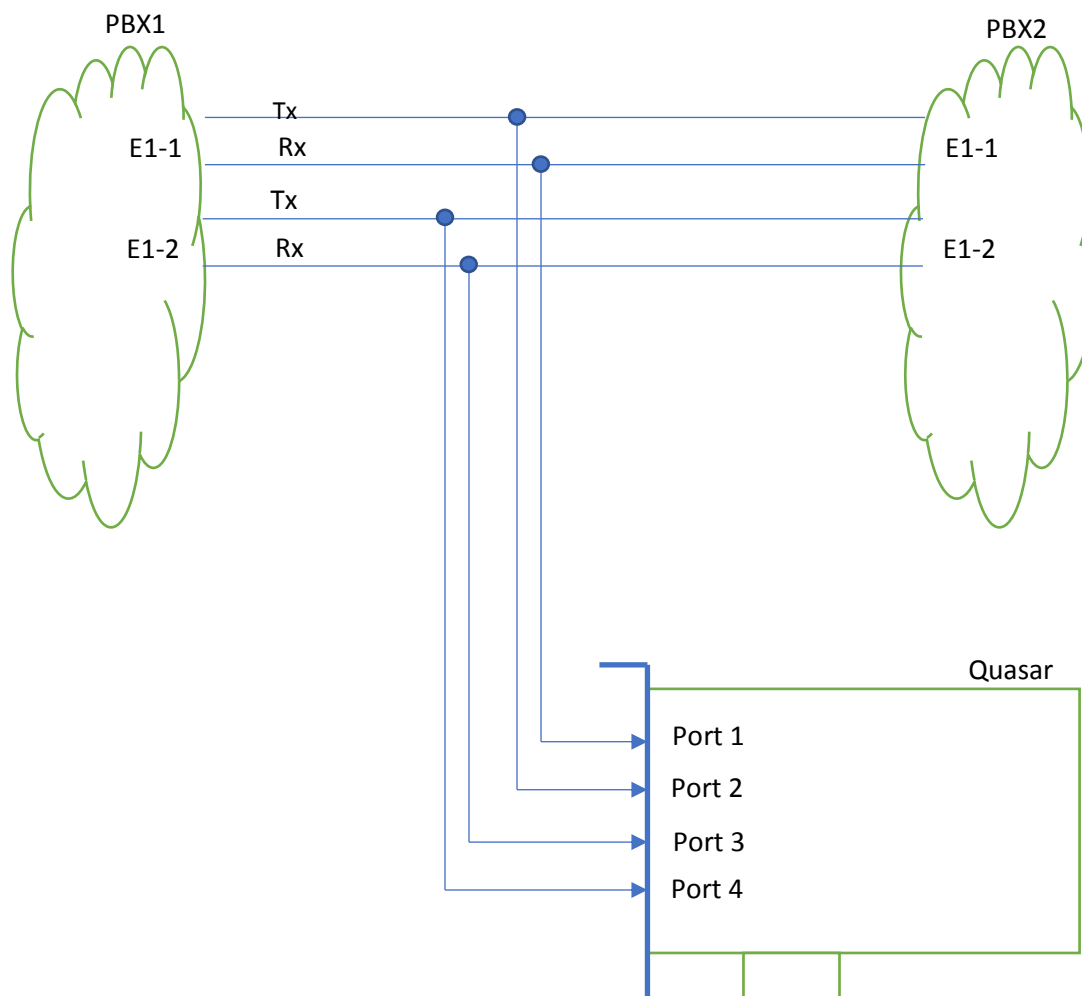
Далее перейдем к рассмотрению конкретных примеров подключения.

7.1. ПОДКЛЮЧЕНИЕ К ОДНОЙ ЛИНИИ E1



```
[linkset 0]
numslots = 32
dslot0 = 16
proto = ss7
law = a
```

7.2. ПОДКЛЮЧЕНИЕ К ДВУМ ЛИНИЯМ E1 С РАЗДЕЛЬНЫМИ КАНАЛАМИ СИГНАЛИЗАЦИИ

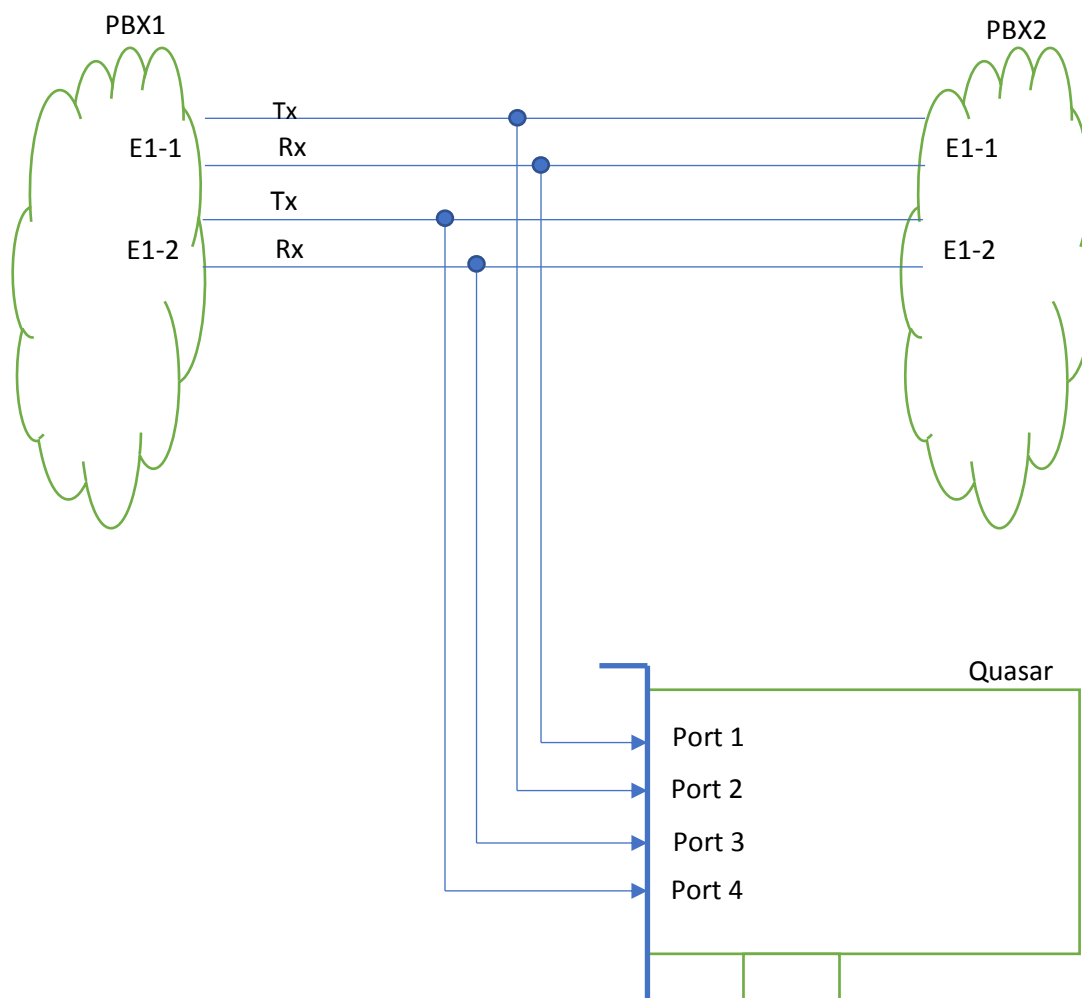


Так как обе линии E1 имеют свой сигнальный канал, нужно описать две секции linkset. Схема подключения линий к портам RX1-TX1-RX2-TX2

```
[linkset 0]
numslots = 32
dslot0 = 16
proto = ss7
law = a
```

```
[linkset 1]
numslots = 32
dslot0 = 16
proto = ss7
law = a
```

7.3. ПОДКЛЮЧЕНИЕ К ДВУМ ЛИНИЯМ E1 С ОБЩИМ КАНАЛОМ СИГНАЛИЗАЦИИ

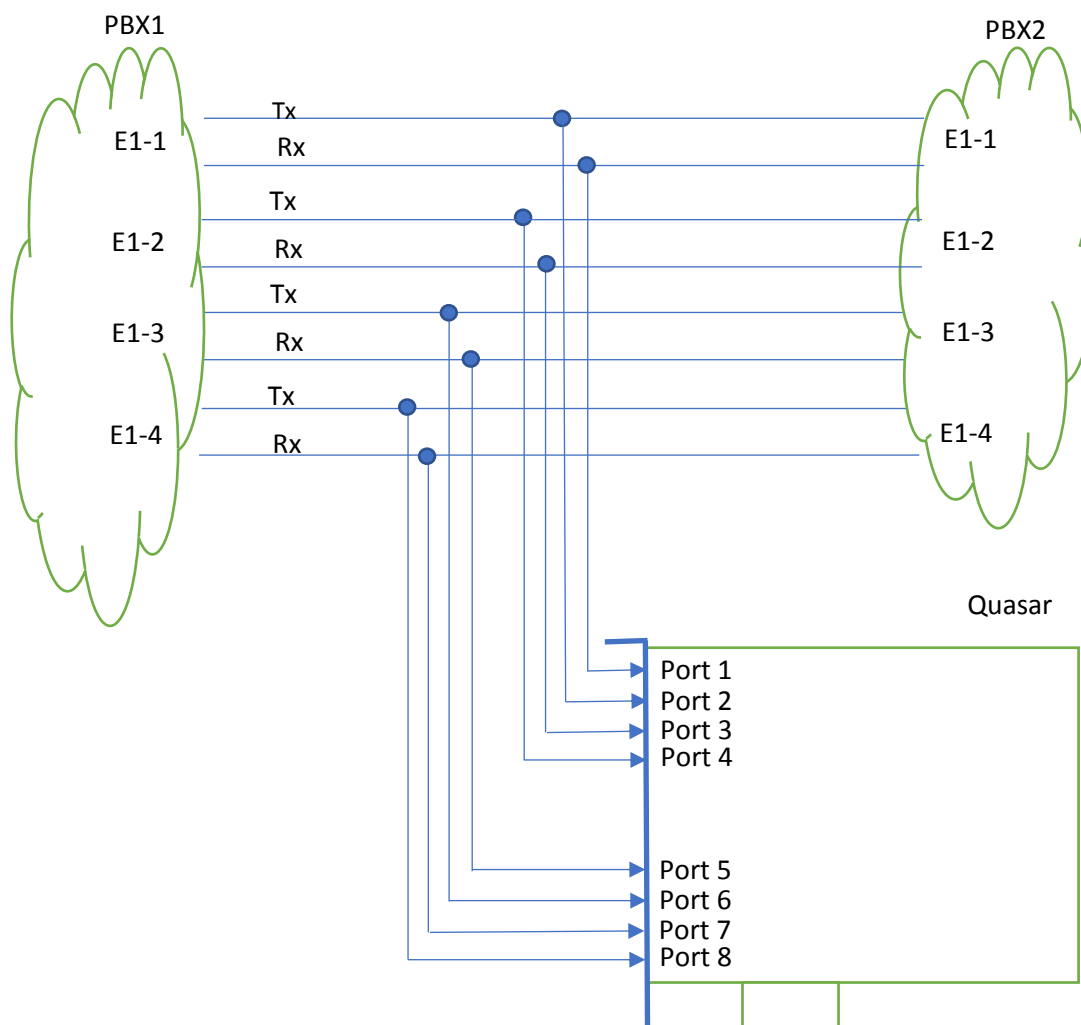


Обе линии E1 имеют общий сигнальный канал (например, 16 на линии E1-1), группа должна быть описана в одной секции linkset. Схема подключения линий к портам RX1-TX1-RX2-TX2.

```
[linkset 0]
numslots = 64
dslot0 = 16
proto = pri
law = a
ifid0 = 2
ifid1 = 4
```

Если используется сигнализация PRI, возможно понадобится указать ifid для каждой линии. Id интерфейса назначается администратором PBX1 или PBX2.

7.4. ПОДКЛЮЧЕНИЕ К ЧЕТЫРЕМ ЛИНИЯМ E1 С ОБЩИМ И РЕЗЕРВНЫМ КАНАЛОМ СИГНАЛИЗАЦИИ



```
[linkset 0]
numslots = 128
dslot0 = 16
dslot1 = 48
proto = ss7
law = a
ifid0 = 2
ifid1 = 4
ifid2 = 6
ifid3 = 7
```

Четыре линии имеют общий сигнальный канал (16) и резервный сигнальный канал (48), группа должна быть описана в одной секции linkset. Схема подключения линий к портам RX1-TX1-RX2-TX2-RX3-TX3-RX4-TX4.

Если используется сигнализация PRI, возможно понадобится указать ifid для каждой линии. Id интерфейса назначается администратором PBX1 или PBX2.

8. ФОРМАТЫ ВЫХОДНЫХ ФАЙЛОВ

8.1. МЕТАФАЙЛЫ

Метафайл соответствует одному законченному телефонному вызову, детектированному системой. Каждый метафайл имеет уникальный цифровой идентификатор звонка, который также является именем файла. Метафайл также имеет парный файл с аудиоданными, а котором записано содержимое вызова. Имена соответствующих метафайла и медиафайла совпадают и эквивалентны идентификатору. Метафайлы имеют расширение **.meta**, например 16068878173880240011.meta.

Метафайлы содержат следующие поля:

ID 16068878173880240011 – уникальный идентификатор вызова

LINKSET 0 – номер группы E1, на которой записан звонок

SIG - сигнализация

BCHAN – номер таймслота

NUMCALLED – номер вызываемого абонента

NUMCALLING – номер вызывающего абонента

TIME – время и дата начала вызова, определяется по первому пакету сигнализации

DURATION – длительность звонка (в секундах)

TSETUP – абсолютная метка времени начала вызова (секунды)

TPROGRESS – абсолютная метка времени подтверждения вызова встречной АТС (секунды)

TRELEASE - абсолютная метка времени окончания вызова (секунды)

TANSWER - абсолютная метка времени ответа абонента (секунды)

CAUSE – причина окончания вызова, в терминологии протокола сигнализации

CAUSECODE – код причины окончания вызова, в десятичном виде

RELEASEDIR – кто закончил вызов, “I” – инициатор вызова, “D” – получатель вызова

CALLDIR – направление вызова, “>” – инициатор на линии RX, “<” - инициатор на линии TX

PROTO – специфические поля для соответствующего протокола:

PRI.CRVALUE – call reference value

PRI.CODING – кодировка вызова, в соответствии с протоколом

SS7.SPC – source point code

SS7.DPC – destination point code

SS7.CIC – circuit identification code

Пример meta-файла:

ID 16294360859244850001

LINKSET 0

SIG SS7

BCHAN 1

NUMCALLED 310

NUMCALLING 13202

TIME 2021-08-20 11:08:05.92

DURATION 0017.94

TSETUP 1629436085.92

TPROGRESS 1629436086.07

TRELEASE 1629436103.86

TANSWER 1629436091.08

CAUSE Normal call clearing

CAUSECODE 16

CALLDIR >

RELEASEDIR I

PROTO SS7.SPC=599/SS7.DPC=650/SS7.CIC=1

8.2. АУДИО ФАЙЛЫ

Аудиоданные сохраняются в файлах формата wav, со следующими характеристиками:

- частота дискретизации 8 кгц
- Количество каналов – 2
- Кодировка A-law для протокола SS7, для PRI определяется протоколом.

Имя файла эквивалентно идентификатору вызова и является уникальным.

Файл создается системой после получения по сигнальному каналу пакета установления соединения. С этого момента идет непрерывная запись аудиоданных в файл, из указанных в протоколе таймслотов. В один канал wav файла записывается сторона приемника, в другой канал – сторона передатчика. Запись заканчивается после получения пакета разрыва соединения. Файлы пишутся в директорию, указанную в параметре audioPath (см. описание ini-файла).

8.3. БАЗА ДАННЫХ

Информация, которая записывается в метафайлах, почти целиком дублируется в базе данных SQL. Имя базы данных задаются в параметре dbName (см. описание ini файла). Доступ к БД осуществляется через унифицированный интерфейс ODBC, благодаря чему можно организовать запись в большинство популярных БД – SQLite, Postgre, Microsoft Access, MySQL, Oracle, Microsoft SQL Server и т.д.

База данных позволяет искать и сортировать записи по различным характеристикам – номерам телефонов, времени и длительности вызова.

8.4. LOG ФАЙЛ

В Log файл выводятся сообщения протокольного уровня, что позволяет непосредственно наблюдать установление соединения. Пример log файла приведен ниже:

```
2020-12-02 11:54:29.53> Q931 MSG: SETUP (5), CALL REF: 201
2020-12-02 11:54:29.53> Q931 IE: BEARER (4), length=3
2020-12-02 11:54:29.53> Q931 IE: speech 64kbit/s, coding a-law
2020-12-02 11:54:29.53> Q931 IE: CHANNEL ID (24), length=4
2020-12-02 11:54:29.53> Q931 IE: Chan ID: ifid=2, timeslot=22
2020-12-02 11:54:29.53> Q931 IE: ? (40), length=4
2020-12-02 11:54:29.53> Q931 IE: CALLING PARTY (108), length=2
2020-12-02 11:54:29.53> Q931 IE: Calling number:
2020-12-02 11:54:29.53> Q931 IE: CALLED PARTY (112), length=3
2020-12-02 11:54:29.53> Q931 IE: Called number: 99
2020-12-02 11:54:29.53> Q931 IE: ? (161), length=-1
2020-12-02 11:54:29.54> Q931 MSG: CALL PROCEEDING (2), CALL REF: 201
2020-12-02 11:54:29.54> Q931 IE: CHANNEL ID (24), length=4
2020-12-02 11:54:29.54> Q931 IE: Chan ID: ifid=2, timeslot=22
2020-12-02 11:54:29.57> Q931 MSG: CONNECT (7), CALL REF: 201
2020-12-02 11:54:29.57> Q931 IE: CHANNEL ID (24), length=4
2020-12-02 11:54:29.57> Q931 IE: Chan ID: ifid=2, timeslot=22
2020-12-02 11:54:29.57> Q931 IE: PROGRESS INDICATOR (30), length=2
2020-12-02 11:54:29.58> Q931 MSG: CONNECT ACK (15), CALL REF: 201
2020-12-02 11:54:32.04> Q931 MSG: DISCONNECT (69), CALL REF: 192
2020-12-02 11:54:32.04> Q931 IE: CAUSE (8), length=2
```

2020-12-02 11:54:32.04> Q931 IE: Cause: Normal call clearing

В зависимости от настроек в ini-файле, в лог файл может выводиться информация 2 или 3 уровня протокола, файл может дублироваться в консоль.