

Серия: API и внешняя интеграция радиооборудования Expert Electronics

12 January 2024

Указание правообладателя

Конфиденциальность

Вся информация, содержащаяся в данном документе открыта для публичного доступа. Она представлена с целью обеспечения технической информацией о протоколе TCI сторонних компаний и частных лиц, желающим интегрировать свое программное/аппаратное обеспечение с продуктами Expert Electronics.

Обязательства и условия

Авторские права (с) 2023 ООО «Эксперт Групп», г. Таганрог

Лицензия разрешает лицам, использующим интерфейс TCI (далее Интерфейс) и сопутствующую документацию к нему, безвозмездно использовать его в любом Программном Обеспечении без каких-либо ограничений и условий, включая неограниченное право на использование, копирование, изменение, слияние, публикацию, распространение, сублицензирование и/или продажу копий Программного Обеспечения с Интерфейсом в его составе, а также лицам, которым предоставляется данное Программное Обеспечение.

ИНТЕРФЕЙС ПРЕДОСТАВЛЯЕТСЯ «КАК ЕСТЬ», БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ЯВНО ВЫРАЖЕННЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ. ВКЛЮЧАЯ ГАРАНТИИ ТОВАРНОЙ ПРИГОДНОСТИ, СООТВЕТСТВИЯ ПО ЕГО КОНКРЕТНОМУ НАЗНАЧЕНИЮ И ОТСУТСТВИЯ НАРУШЕНИЙ, НО НЕ ОГРАНИЧИВАЯСЬ ИМИ. НИ В КАКОМ СЛУЧАЕ АВТОРЫ ИЛИ ПРАВООБЛАДАТЕЛИ НЕ НЕСУТ ОТВЕТСТВЕННОСТЬ ПО КАКИМ-ЛИБО ИСКАМ, ЗА УЩЕРБ ИЛИ ПО ИНЫМ ТРЕБОВАНИЯМ, В ТОМ ЧИСЛЕ, ПРИ ДЕЙСТВИИ КОНТРАКТА, ДЕЛИКТЕ ИЛИ ИНОЙ СИТУАЦИИ, ВОЗНИКШИМ ИЗ-ЗА ИСПОЛЬЗОВАНИЯ ИНТЕРФЕЙСА ИЛИ ИНЫХ ДЕЙСТВИЙ С ИНТЕРФЕЙСОМ.

Контакт

| | | | |
|----------|---|--------|--|
| Имя: | ООО «Эксперт Групп» | | |
| Тел: | +7 (8634) 43-13-01 | Email: | info@sunsdr.com |
| Адрес: | 347927, Russia, Rostov region, Taganrog, Polyakovskoe schosse, 16-3, ABK-1, off. 408 | | |
| Вэбсайт: | https://eesdr.com/ru/ | | |

Контроль версий документа

| Версия TCI | Комментарий | Дата | ExpertSDR3 |
|------------|--|----------------|------------|
| 1.9 | <p>Новые команды:</p> <p>AUDIO_STREAM_SAMPLE_TYPE</p> <p>AUDIO_STREAM_CHANNELS</p> <p>AUDIO_STREAM_SAMPLES</p> <p>TX_STREAM_AUDIO_BUFFERING</p> <p>Прочие изменения:</p> <p>Несколько важных обновлений в разделе 3.4 "Работа с аудио потоками через TCI".</p> | 29 июля 2022 | 0.13.0 |
| 1.9.1 | <p>Изменения в командах:</p> <ul style="list-style-type: none"> • KEYER | 31 мая 2023 | 1.0.4 |
| 2.0 | <p>Новые команды:</p> <ul style="list-style-type: none"> • VFO_LOCK • RX_CHANNEL_SENSORS <p>Прочие изменения:</p> <ul style="list-style-type: none"> • Доработана команда TRX | 12 января 2024 | 1.0.7 |
| | | | |
| | | | |

Содержание

| | | |
|-------|---|----|
| 1 | Введение | 4 |
| 1.1 | О чем этот документ | 4 |
| 1.2 | Цель документа | 4 |
| 1.3 | Для кого этот документ | 4 |
| 1.4 | Специфика | 4 |
| 2 | Обновление и Изменения в данной версии | 6 |
| 2.1 | Обновленные команды | 6 |
| 2.2 | Новые команды | 6 |
| 2.3 | Прочие изменения | 6 |
| 3 | Общая информация | 7 |
| 3.1 | Описание интерфейса | 7 |
| 3.2 | Работа в телеграфном режиме | 8 |
| 3.2.1 | CW макрос | 8 |
| 3.2.2 | CW сообщение | 9 |
| 3.3 | Типы TCI команд | 10 |
| 3.4 | Работа с аудио потоками через TCI | 11 |
| 3.5 | Особенности работы TCI сервера в ExpertSDR3 | 12 |
| 3.6 | Заключение | 13 |
| 4 | Список команд TCI | 14 |
| 4.1 | Команды инициализации | 14 |
| 4.2 | Команды двунаправленного управления | 16 |
| 4.3 | Команды одностороннего управления | 32 |
| 4.4 | Команды уведомления | 39 |
| 4.5 | Новые команды в этой версии | 43 |

1 Введение

1.1 О чем этот документ

Интерфейс управления трансивером TCI (от англ. TCI – Transceiver Control Interface) представляет собой сетевой интерфейс управления, передачи данных и синхронизации между трансивером/приемником, аппаратным журналом, контекст логгером, программами цифровых видов связи, скиммерами и другим программным обеспечением, а также внешними усилителями мощности, блоками диапазонных фильтров, антенными коммутаторами, контроллерами радиостанции и др. устройствами.

1.2 Цель документа

Этот документ описывает протокол TCI, для чего он и как им пользоваться.

1.3 Для кого этот документ

Целевой аудиторией этого документа являются программисты, кто будет заниматься внедрением TCI в их программы и устройства.

1.4 Специфика

TCI был создан в качестве современной альтернативы устаревшим интерфейсам СОМ-портов и звуковых кабелей, он использует полнодуплексный web-сокет протокол, который работает поверх соединения TCP и служит для обмена данными между сервером и клиентом, обеспечивая кроссплатформенность. В качестве сервера выступает трансивер, в качестве клиентов – все остальные программы и устройства. Сервер и клиенты могут находиться внутри одного компьютера (программа-сервер, аппаратный журнал и др. - клиенты) и/или в отдельных физических устройствах, объединенных через локальную сеть (классический трансивер, усилитель мощности, антенный коммутатор, блок ДПФ и т.п.).

Интерфейс TCI содержит основные команды управления трансивером (аналог САТ системы), принимает от клиентов CW макросы и передает их в эфир, выдает квадратурный поток трансивера клиентам, принимает споты от скиммеров и интернет кластеров, принимает/выдает аудио сигнал для работы в цифровых видах связи.

TCI использует расширяемую архитектуру и может быть дополнен новыми функциями и командами, при сохранении работоспособности старых. Таким образом интерфейс TCI может быть расширен и дополнен под конкретные нужды любого производителя программного обеспечения и/или производителя устройств (приемников, трансиверов, усилителей мощности, коммутаторов и др.). Наличие идентификатора устройства позволяет производителям трансиверов и приемников перейти на TCI интерфейс с сохранением обозначения модели устройства. Расширяемость интерфейса TCI позволяет создавать индивидуальный набор команд и функций для каждой

модели устройства, при этом сохраняя основной набор команд, присущий всем трансиверам.

Наша компания выступает за всеобщую унификацию обмена данными между устройствами и программами, создав для этого интерфейс TCI. Современные трансиверы и программное обеспечение должны общаться, используя один протокол – протокол TCI.

2 Обновление и Изменения в данной версии

2.1 Обновленные команды

- TRX

2.2 Новые команды

- VFO_LOCK
- RX_CHANNEL_SENSORS

2.3 Прочие изменения

3 Общая информация

3.1 Описание интерфейса

Любая команда представляет ASCII строку, которая содержит название команды и список аргументов, соответствующих данной команде. Имеются зарезервированные символы, которые не могут входить в название команды и в аргументы команды.

Список зарезервированных символов: «:», «,», «;».

Структура команды:

1. Название команды;
2. Разделительный символ между названием команды и аргументами «:»;
3. Список аргументов через разделительный знак «,»;
4. Символ конца команды «;».

Если аргументов команда не имеет, то после названия команды ставится символ конца команды. Если команда неправильная, то она игнорируется. Регистр букв не имеет значения.

Программа ExpertSDR3 выполняет роль сервера, который может иметь одновременно несколько клиентских подключений, они будут синхронизироваться между собой сервером. При подключении к ExpertSDR3 клиент получает текущее состояние ExpertSDR3, сначала высылаются команды инициализации, затем параметры для установки состояния, такие как частота, модуляция и др.

Когда в программе ExpertSDR3(сервер) происходит изменение параметра, сервер уведомляет об этом всех подключённых клиентов, то есть клиентам не нужно постоянно опрашивать сервер, любое изменение состояния будет отправлено своевременно всем клиентам. Если клиент высылает новое состояние, то сервер его установит себе, а также вышлет всем клиентам, то есть сервер выполняет роль синхронизатора. Все клиенты, подключённые к серверу, будут автоматически синхронизированы. Такой способ работы позволяет минимизировать нагрузку на сеть, уменьшая трафик.

В протоколе TCI реализована передача IQ потока приёмника клиентам, которая нужна для работы специальных программ скиммеров, они автоматически находят станции и декодируют их во всей полосе, а ещё это позволяет записывать радиоэфир в файл.

Через TCI осуществляется передача НЧ аудиосигнала приёмника клиентам и обратно, то есть клиент может передавать НЧ аудиосигнал в ExpertSDR3 для излучения в эфир. Обмен аудио потоком предназначен для работы с цифровыми видами связи, когда кодирование и декодирование выполняется сторонними

программами, а также в речевых видах связи, в которых можно передавать в эфир аудио макросы, что очень востребовано в аппаратных журналах.

При работе в контекстах важно вести запись всей работы в эфир, для этого реализована отправка аудиопотока из линейного аудиовыхода всем клиентам. Полученный аудиопоток может быть записан в файл или воспроизведён звуковой картой ПК.

3.2 Работа в телеграфном режиме

Протокол TCI предусматривает работу телеграфом с помощью строковых команд (макросов). Команды делятся на два типа:

1. Макрос;
2. Сообщение.

3.2.1 CW макрос

Макрос – это набор символов, не содержащий правил, но подчиняющийся командам изменения скорости и передачи аббревиатуры, а также позволяющий работать в терминальном режиме.

Команда отправки макроса имеет вид: `cw_macros:arg1,arg2;`

`arg1` - номер программного передатчика;

`arg2` - текстовое сообщение.

Пример: `cw_macros:0,TU RA6LH 599;`

Если нужно внутри текста поместить аббревиатуру (слитно передать несколько букв), то символы нужно поместить между вертикальными скобками:

TEXT `|SK|` TEXT.

Если внутри текста нужно уменьшить скорость, то для этого используется символ «<», для увеличения скорости «>» соответственно. Шаг изменения скорости 5 wrpm, например:

`ANY TEXT > TEXT+5WPM >>TEXT+15WPM`

Чтобы передать строку «+5wrpmTU -5wrpmRA6LH +10wrpm599 004 SK» команда будет иметь вид:

`cw_macros:0,>TU >599 004 |SK|;`

Так как текстовые команды могут содержать запрещённые протоколом символы, то они заменяются на другие символы и на стороне сервера снова преобразуются обратно:

1. Символ «:» заменяется на «^»;
2. Символ «,» заменяется на «~»;
3. Символ «;» заменяется на «*».

При передаче макросов есть возможность работы в терминальном режиме, то есть после завершения передачи макроса трансивер остаётся на передаче. Включается/выключается терминальный режим следующей командой:

- `cw_terminal:true`; - включение;
- `cw_terminal:false`; - выключение.

В терминальном режиме, по факту начала передачи последней буквы в посылке, высылается следующая команда клиенту: `cw_macros_empty`;

Если во время проигрывания макроса выключить терминальный режим, то после завершения макроса трансивер перейдёт в режим приёма автоматически.

3.2.2 CW сообщение

Сообщение – это специальная команда, которая имеет более сложную структуру и состоит из трёх частей:

1. Префикс - текст перед позывным;
2. Позывной;
3. Суффикс - текст после позывного.

Команда отправки телеграфного сообщения с возможностью доотправки позывного:

`cw_msg:arg1,arg2,arg3,arg4;`

- `arg1` - номер программного передатчика
- `arg2` - префикс
- `arg3` - позывной
- `arg4` - суффикс

Чтобы передать строку «`TU RA6LH 599 004`» команда будет иметь вид:

`cw_msg:0,TU,RA6LH,599 004;`

Если нужно два раза повторить позывной «**TU RA6LH RA6LH 599 004**», то команда будет иметь вид:

```
cw_msg:0,TU,RA6LH$2,599 004;
```

Если при отправке сообщения ещё полностью не известен позывной и нужно будет его дописать или изменить, и возможно не один раз, то команда будет выглядеть так:

```
cw_msg:arg1;
```

Пример последовательной доотправки позывного:

1. `cw_msg:0,_,RA6$2,599 004;`
2. `cw_msg:RA6L;`
3. `cw_msg:RA6LH;`

Если редактирование позывного было выполнено после того, как была закончена передача позывного, то команда игнорируется. Процесс редактирования позывного осуществляется для еще не переданных символов. После того, как передача позывного завершена, клиенту отправляется команда, которая содержит финальный вариант позывного, переданного в эфир:

```
callsign_send:RA6LH;
```

Если во время передачи телеграфного макроса или сообщения потребовалось резко прекратить передачу, для этого есть команда:

```
cw_macros_stop;
```

3.3 Типы TCI команд

В протоколе предусмотрено несколько типов команд:

- Команды инициализации;
- Команды двунаправленного управления;
- Команды однонаправленного управления;
- Команды уведомления.

Команды инициализации отправляются клиенту при подключении, они сообщают клиенту основные параметры устройства, такие как диапазон рабочих частот, поддерживаемые виды связи и т.д.

Команды двунаправленного управления нужны для управления основными параметрами устройства и ExpertSDR3. Так как команды двунаправленные,

сервер будет выполнять синхронизацию параметров всех подключённых клиентов.

Команды однонаправленного управления используются для управления параметрами уникальными для каждого клиента, например, включение аудио потока, отправка спотов для отображения на панораме ExpertSDR3 и т.д.

Команды уведомления высылаются клиентам с некоторой периодичностью, это могут быть показания различных датчиков и измерителей сигналов.

3.4 Работа с аудио потоками через TCI

Благодаря протоколу [WebSocket](#) команды и аудиопотоки разделены между собой, команды передаются строками, а аудиопотоки передаются потоком байт. Аудиопоток передаётся блоками байт, у блока есть заголовок и поле данных.

Структура блока на языке СИ:

```
struct Stream
{
    uint32_t receiver;           // номер приёмника
    uint32_t sample_rate;       // частота дискретизации
    uint32_t format;            // тип сэмплов определён в StreamType
    uint32_t codec;             // алгоритм сжатия (не реализовано), всегда 0
    uint32_t crc;               // контрольная сумма (не реализовано), всегда 0
    uint32_t length;           // количество отсчётов
    uint32_t type;              // тип потока
    uint32_t channels;          // количество каналов
    uint32_t reserv[8];         // зарезервировано
    uint8_t data[16384];        // сэмплы
};
```

Тип потока определяется перечислением:

```
enum StreamType
{
    IQ_STREAM = 0,              // Поток IQ сигнала приёмника
    RX_AUDIO_STREAM = 1,        // Аудиопоток приёмника
    TX_AUDIO_STREAM = 2,        // Аудиопоток для передатчика
    TX_CHRONO = 3,              // Поток маркеров времени для передачи аудио сигнала
    LINEOUT_STREAM = 4          // Аудиопоток линейного аудио выхода
}

enum SampleType
{
    INT16 = 0, // 16 битный целочисленный знаковый тип
    INT24 = 1, // 24 битный целочисленный знаковый тип
    INT32 = 2, // 32 битный целочисленный знаковый тип
}
```

```
FLOAT32 = 3, // 32 битный целочисленный знаковый тип  
}
```

После подключения клиент может включить получение IQ потока командой **IQ_START** после отправки этой команды ExpertSDR3 начинает отправлять IQ поток. В поле **Stream.length** fie указывается количество вещественных отсчётов в поле **uint8_t data[]**, так как сэмплы комплексные, то количество комплексных отсчётов вычисляется как **Stream.length/Stream.channels**.

Аудиопоток приёмника полностью повторяет IQ поток, но отличие заключается в возможности управлять некоторыми параметрами, такими как:

- количество каналов
- формат сэмплов
- количество сэмплов передаваемых в одном пакете

Когда выбраны виды связи **DIGL/DIGU** будет передаваться комплексный сигнал, если выбрано количество каналов 2 или вещественный, если выбран 1 канал. Аудиопоток линейного аудио выхода повторяет обычный аудио поток, но ему нельзя устанавливать параметры.

Отправка аудиопотока передатчику имеет определённые особенности: ExpertSDR3 отправляет временной маркер **TX_CHRONO** который уведомляет клиента о необходимости отправить аудиосигнал, помеченный как **TX_AUDIO_STREAM** с заданным количеством отсчётов в поле **Stream.length**. Временные маркеры отправляются без ожидания ответа от клиента. Если сигнал на отправку в ExpertSDR3 ещё не готов, то клиент может не отправлять ответ или может отправить сигнал с обнулёнными отсчётами, что соответствует отсутствию сигнала - такой вариант предпочтительнее.

По сравнению с версией 1.8 произошли изменения в структуре Stream, но бинарная совместимость осталось, все программы, работающие с протоколом 1.8 будут работать с версией 1.9.

3.5 Особенности работы TCI сервера в ExpertSDR3

Сервер выполняет синхронизацию всех подключённых к нему клиентов – это задача очень сложная, поэтому в работе сервера имеются особенности, которые нужно учитывать при написании собственного клиента.

Количество команд довольно большое и увеличивается с выходом каждой новой версии ExpertSDR3, но это не мешает эффективно выполнять синхронизацию всех подключённых клиентов. В основе лежит иерархия, однозначно определяющая ведущего и ведомого. Если несколько клиентов попытаются управлять одним и тем же параметром одновременно, приоритет будет у того, кто был первым, но наивысший приоритет всегда у оператора ExpertSDR3. Даже если один из клиентов управляет одним из параметров, оператор всегда может перехватить управление.

Важно знать, что в ExpertSDR3 реализовано запоминание некоторых настроек по диапазонам и видам связи. Когда меняется частотный диапазон клиентам отправляются все изменённые параметры в определённой последовательности. Например, если клиент задаёт частоту, которая соответствует другому частотному диапазону, ExpertSDR3 установит эту частоту и сменит частотный диапазон, затем программа восстановит настройки и отправит их подключённым клиентам. Когда ExpertSDR3 является инициатором изменения параметра, он захватывает управление над ним на 200 мс и клиенты не могут его изменить в течении этого времени. Это же правило работает если один из клиентов управляет параметром, в то время как остальные клиенты могут только получать уведомления, но перехватить управление не могут до тех пор, пока не пройдёт 200 мс после того, как захвативший управление клиент не прекратит менять параметр.

Работа с CW макросами тоже имеет особенности, так как работа с макросами проектировалась с акцентом на работу в контестах, но это не мешает использованию макросов в повседневной работе в эфире. В протоколе реализовано две команды для передачи текстового сообщения в CW, первая и самая простая **CW_MACROS** передаёт текст в эфир, во время передачи макроса можно добавить в очередь ещё макросы, и они все будут переданы в эфир в порядке добавления. Вторая команда **CW_MSG**, имеет более сложную структуру, префикс, позывной и суффикс. До того, как позывной не передан в эфир его можно редактировать, но если во время передачи такого сообщения повторно отправить **CW_MSG** с префиксом, позывным и суффиксом, тогда текущая сессия будет прекращена и сразу начнётся новая сессия передачи сообщения. Если во время передачи **CW_MSG** послать **CW_MACROS** то он помещается в очередь на отправку после завершения передачи сообщения. Если во время передачи **CW_MACROS** послать команду **CW_MSG**, в таком случае немедленно будет прекращена передача макроса и начнётся передача сообщения, так как сообщение имеет приоритет выше.

3.6 Заключение

Протокол TCI постоянно развивается. С выходом каждой новой версии программы ExpertSDR3, расширяется количество команд, что позволяет увеличить функциональность программ и устройств, подключаемых к ExpertSDR3.

Команда разработчиков всегда прислушивается к мнению пользователей ExpertSDR3. Мы стараемся делать наш продукт удобным как для разработчиков программного обеспечения, позволяя им не тратить много времени на интеграцию своих программ и устройств, так и для рядовых пользователей, которым важна не техническая сторона, а удобство и простота использования.

Каждый из вас может внести вклад в развитие программы. Ваши предложения, пожелания и замечания помогут нам сделать ExpertSDR3 лучше.

4 Список команд TCI

4.1 Команды инициализации

| | | |
|------------|-----------------------------|---|
| VFO_LIMITS | Диапазон рабочих частот. | |
| Ответ | VFO_LIMITS:arg1,arg2; | arg1 – нижняя граничная частота, Гц. arg2 – верхняя граничная частота, Гц. |
| Тип | инициализация | |
| Пример | VFO_LIMITS:10000,300000000; | |

| | | |
|------------|--|---|
| IF_LIMITS | Граничные значения частот фильтра ПЧ (только для VFOA). | |
| Ответ | IF_LIMITS:arg1,arg2; | arg1 – нижняя граничная частота, Гц. arg2 – верхняя граничная частота, Гц. |
| Тип | инициализация | |
| Пример | IF_LIMITS:-48000,48000; | |
| Дополнение | Высылается при подключении и изменении частоты дискретизации устройства. | |

| | | |
|-----------|---|--|
| TRX_COUNT | Количество приёмников (трансиверов) в устройстве. | |
| Ответ | TRX_COUNT:arg1; | arg1 – количество приёмников/трансиверов (физических или программных). |
| Тип | инициализация | |
| Пример | TRX_COUNT:2; | |

| | | |
|---------------|--|-----------------------------------|
| CHANNEL_COUNT | Количество каналов приёма в одном приёмнике (A/B/C). | |
| Ответ | CHANNEL_COUNT:arg1; | arg1 – количество каналов приёма. |
| Тип | инициализация | |
| Пример | CHANNEL_COUNT:2; | |

| | | |
|--------|----------------------|-----------------------------|
| DEVICE | Название устройства. | |
| Ответ | DEVICE:arg1; | arg1 – название устройства. |
| Тип | инициализация | |
| Пример | DEVICE:SunSDR2DX; | |

| | | |
|--------------|--|---|
| RECEIVE_ONLY | Определяет устройство как приёмник или передатчик. | |
| Ответ | RECEIVE_ONLY:arg1; | arg1 – только приём (true), трансивер (false). |
| Тип | инициализация | |
| Пример | RECEIVE_ONLY:true; | |

| | | |
|------------------|---|------------------------------------|
| MODULATIONS_LIST | Список поддерживаемых видов связи. | |
| Ответ | MODULATIONS_LIST:arg1,arg2, ... ,argN; | Вид связи передаётся названием. |
| Тип | инициализация | |
| Пример | MODULATIONS_LIST:AM,LSB,USB,FM; | |

| | | |
|----------|------------------------------------|--|
| PROTOCOL | Информация о версии протокола TCI. | |
| Ответ | PROTOCOL:arg1,arg2; | arg1 – название программы. arg2 – версия протокола. |
| Тип | инициализация | |
| Пример | PROTOCOL:ExpertSDR3,1.7; | |

| | | |
|-------|--|--|
| READY | Высылается после команд инициализации. | |
| Ответ | READY; | |
| Тип | инициализация | |

4.2 Команды двунаправленного управления

| | | |
|------------|----------------------------|--|
| START | Старт устройства. | |
| Установить | START; | |
| Тип | двунаправленное управление | |
| Пример | START; | |

| | | |
|------------|----------------------------|--|
| STOP | Остановка устройства. | |
| Установить | STOP; | |
| Тип | двунаправленное управление | |
| Пример | STOP; | |

| | | |
|------------|--|--|
| DDS | Управление центральной частотой настройки приёмника. | |
| Установить | DDS:arg1,arg2; | arg1 – номер приёмника; arg2 – частота настройки, Гц. |
| Прочитать | DDS:arg1; | |
| Ответ | DDS:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | DDS:0; DDS:0,7100000; | |

| | | |
|------------|---|--|
| IF | Управление частотой настройки фильтра ПЧ в пределах панорамы. | |
| Установить | IF:arg1,arg2,arg3; | arg1 – номер приёмника; arg2 – номер канала (A / B); arg3 – частота настройки, Гц. |
| Прочитать | IF:arg1,arg2; | |
| Ответ | IF:arg1,arg2,arg3; | |
| Тип | двунаправленное управление | |
| Пример | IF:0,1; IF:0,1,12500; IF:0,1,-17550; | |

| | | |
|------------|--|--|
| VFO | Управление частотой настройки приёмника. | |
| Установить | VFO:arg1,arg2,arg3; | arg1 – номер приёмника; arg2 – номер канала (A/B); arg3 – частота настройки, Гц. |
| Прочитать | VFO:arg1,arg2; | |
| Ответ | VFO:arg1,arg2,arg3; | |
| Тип | двунаправленное управление | |
| Пример | VFO:0,1,7100000; | |

| | | |
|------------|------------------------------------|---|
| MODULATION | Управление видами связи. | |
| Установить | MODULATION:arg1,arg2; | arg1 – номер приёмника; arg2 – вид связи (строка). |
| Прочитать | MODULATION:arg1; | |
| Ответ | MODULATION:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | MODULATION:0,LSB; MODULATION:1; | |

| | | |
|------------|--|--|
| | MODULATION:1,NFM; | |
| Дополнение | Список поддерживаемых видов связи высылается клиенту при подключении к ExpertSDR3. | |

| | | |
|------------|---|---|
| TRX | Переключение режимов приём передача. | |
| Установить | TRX:arg1,arg2,arg3; | arg1 – номер передатчика; arg2 – флаг включения. arg3 – источник сигнала (необязательно) tci - брать сигнал из TCI mic1 - брать сигнал из MIC1 mic2 - брать сигнал из MIC2 micpc - брать сигнал из MIC PC Ecoder2 - брать сигнал из E-Coder2 Если аргумент отсутствует, то сигнал берется из микрофона. |
| Прочитать | TRX:arg1; | |
| Ответ | TRX:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | TRX:0,true; TRX:0,true,tci; TRX:0,false; TRX:0,true,micpc; TRX:0,true,ecoder2; TRX:0,true,mic2; TRX:1; | |
| Дополнение | <p>Сигнал на передачу всегда берется из выбранного в программе микрофона. Если сторонняя программа подключенная по TCI хочет передавать свой аудиосигнал, то необходимо указать третий аргумент - TCI. Это работает если включен аудиопоток по TCI, иначе берется сигнал микрофона.</p> <p>Если требуется указать определённый вход на время передачи, тогда можно указать один из микрофонных входов, если указан не TCI, тогда включение аудио потока по TCI необязательно.</p> | |

| | | |
|------------|---|---------------------------|
| TUNE | Переключение режимов приём и излучение несущей. | |
| Установить | TUNE:arg1,arg2; | arg1 – номер передатчика; |
| Прочитать | TUNE:arg1; | |

| | | |
|--------|--|------------------------|
| Ответ | TUNE:arg1,arg2; | arg2 – флаг включения. |
| Тип | двунаправленное управление | |
| Пример | TUNE:0,true; TUNE:0,false; TUNE:1; | |

| | | |
|------------|--|---|
| DRIVE | Управление мощностью передатчика. | |
| Установить | DRIVE:arg1,arg2; | arg1 – номер передатчика; arg2 – значение излучаемой мощности от 0 до 100. |
| Прочитать | DRIVE:arg1; | |
| Ответ | DRIVE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | DRIVE:0,30; DRIVE:0,75; DRIVE:1; | |

| | | |
|------------|--|---|
| TUNE_DRIVE | Управление мощностью передатчика в режиме излучения несущей. | |
| Установить | TUNE_DRIVE:arg1,arg2; | arg1 – номер передатчика; arg2 – значение излучаемой мощности от 0 до 100. |
| Прочитать | TUNE_DRIVE:arg1; | |
| Ответ | TUNE_DRIVE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | TUNE_DRIVE:0,30; | |

| | | |
|------------|---------------------------------|-------------------------|
| RIT_ENABLE | Включение расстройки по приёму. | |
| Установить | RIT_ENABLE:arg1,arg2; | arg1 – номер приёмника; |

| | | |
|-----------|-------------------------------------|------------------------|
| Прочитать | RIT_ENABLE:arg1; | arg2 – флаг включения. |
| Ответ | RIT_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RIT_ENABLE:0,true; RIT_ENABLE:1; | |

| | | |
|------------|-----------------------------------|---|
| XIT_ENABLE | Включение расстройки по передаче. | |
| Установить | XIT_ENABLE:arg1,arg2; | arg1 – номер передатчика; arg2 – флаг включения. |
| Прочитать | XIT_ENABLE:arg1; | |
| Ответ | XIT_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | XIT_ENABLE:0,true; | |

| | | |
|--------------|--|---|
| SPLIT_ENABLE | Включение режима передачи на частоте канала В. | |
| Установить | SPLIT_ENABLE:arg1,arg2; | arg1 – номер передатчика; arg2 – флаг включения. |
| Прочитать | SPLIT_ENABLE:arg1; | |
| Ответ | SPLIT_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | SPLIT_ENABLE:0,true; SPLIT_ENABLE:1; | |

| | | |
|------------|---|-------------------------|
| RIT_OFFSET | Управление частотой расстройки приёмника. | |
| Установить | RIT_OFFSET:arg1,arg2; | arg1 – номер приёмника; |

| | | |
|-----------|------------------------------------|--------------------------------|
| Прочитать | RIT_OFFSET:arg1; | arg2 – частота расстройки, Гц. |
| Ответ | RIT_OFFSET:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RIT_OFFSET:0,500; RIT_OFFSET:1; | |

| | | |
|------------|---|---|
| XIT_OFFSET | Управление частотой расстройки передатчика. | |
| Установить | XIT_OFFSET:arg1,arg2; | arg1 – номер передатчика; arg2 – частота расстройки, Гц. |
| Прочитать | XIT_OFFSET:arg1; | |
| Ответ | XIT_OFFSET:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | XIT_OFFSET:0,-350; XIT_OFFSET:1; | |

| | | |
|-------------------|---|---|
| RX_CHANNEL_ENABLE | Включение дополнительного канала приёмника. | |
| Установить | RX_CHANNEL_ENABLE:arg1,arg2,arg3; | arg1 – номер приёмника; arg2 – номер канала. arg3 – флаг включения. |
| Прочитать | RX_CHANNEL_ENABLE:arg1,arg2; | |
| Ответ | RX_CHANNEL_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_CHANNEL_ENABLE:0,1,true; RX_CHANNEL_ENABLE:0,1; | |
| Дополнение | Канал А всегда включён, управлять можно только каналом В. | |

| | | |
|----------------|---|--|
| RX_FILTER_BAND | Управление шириной фильтра основной селекции. | |
| Установить | RX_FILTER_BAND:arg1,arg2,arg3; | arg1 – номер приёмника. arg2 – нижняя частота, Гц. arg3 – верхняя частота, Гц. |
| Прочитать | RX_FILTER_BAND:arg1; | |
| Ответ | RX_FILTER_BAND:arg1,arg2,arg3; | |
| Тип | двунаправленное управление | |
| Пример | RX_FILTER_BAND:0,30,2700; RX_FILTER_BAND:1,-2900,-70; RX_FILTER_BAND:0; | |

| | | |
|-----------------|---|--|
| CW_MACROS_SPEED | Управление скоростью телеграфирования для макросов. | |
| Установить | CW_MACROS_SPEED:arg1; | arg1 – скорость телеграфирования, WPM. |
| Прочитать | CW_MACROS_SPEED; | |
| Ответ | CW_MACROS_SPEED:arg1; | |
| Тип | двунаправленное управление | |
| Пример | CW_MACROS_SPEED:42; CW_MACROS_SPEED; | |

| | | |
|-----------------|---|---|
| CW_MACROS_DELAY | Управление задержкой перед началом телеграфирования после переключения на передачу. | |
| Установить | CW_MACROS_DELAY:arg1; | arg1 – задержка перед началом телеграфирования, мс. |
| Прочитать | CW_MACROS_DELAY; | |
| Ответ | CW_MACROS_DELAY:arg1; | |
| Тип | двунаправленное управление | |
| Пример | CW_MACROS_DELAY:100; | |

| | | |
|--|------------------|--|
| | CW_MACROS_DELAY; | |
|--|------------------|--|

| | | |
|----------------|---|--|
| CW_KEYER_SPEED | Управление скоростью телеграфирования ключом. | |
| Установить | CW_KEYER_SPEED:arg1; | arg1 – скорость телеграфирования, WPM. |
| Тип | двунаправленное управление | |
| Пример | CW_KEYER_SPEED:35; CW_KEYER_SPEED:42; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|------------|--|---|
| VOLUME | Управление общей громкостью программы. | |
| Установить | VOLUME:arg1; | arg1 – значение громкости, дБ. Диапазон значений от -60 до 0 дБ, при значении -60 дБ звук отсутствует. |
| Прочитать | VOLUME; | |
| Ответ | VOLUME:arg1; | |
| Тип | двунаправленное управление | |
| Пример | VOLUME:-12; VOLUME; | |

| | | |
|------------|---------------------------------------|---|
| MUTE | Выключение/включение общей громкости. | |
| Установить | MUTE:arg1; | arg1 – флаг состояния (вкл - true, выкл - false). |
| Прочитать | MUTE; | |
| Ответ | MUTE:arg1; | |
| Тип | двунаправленное управление | |
| Пример | MUTE:true; | |

| | | |
|--|----------------------|--|
| | MUTE:false; MUTE; | |
|--|----------------------|--|

| | | |
|------------|--|--|
| RX_MUTE | Выключение/включение громкости отдельного приемника. | |
| Установить | RX_MUTE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл – true, выкл – false). |
| Прочитать | RX_MUTE:arg1; | |
| Ответ | RX_MUTE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_MUTE:0,true; RX_MUTE:1,false; RX_MUTE:0; | |

| | | |
|------------|---|---|
| RX_VOLUME | Управление громкостью каждого канала приёмника. | |
| Установить | RX_VOLUME:arg1,arg2,arg3; | arg1 – номер приёмника. arg2 – номер канала. arg3 – уровень громкости, дБ. Диапазон значений от -60 до 0 дБ, при значении -60 дБ звук отсутствует. |
| Прочитать | RX_VOLUME:arg1,arg2; | |
| Ответ | RX_VOLUME:arg1,arg2,arg3; | |
| Тип | двунаправленное управление | |
| Пример | RX_VOLUME:0,1,-6; RX_VOLUME:0,0; | |

| | | |
|------------|--|---|
| RX_BALANCE | Управление балансом громкости в каждом канале приемника. | |
| Установить | RX_BALANCE:arg1,arg2,arg3; | arg1 – номер приёмника. arg2 – номер канала. |
| Прочитать | RX_BALANCE:arg1,arg2; | |
| Ответ | RX_BALANCE:arg1,arg2,arg3; | |

| | | |
|--------|---|---|
| Тип | двунаправленное управление | arg3 – уровень громкости, дБ. |
| Пример | RX_BALANCE:0,1,-6; RX_BALANCE:0,0,12; RX_BALANCE:0,1; | Диапазон значений от -40 до +40 дБ, отрицательные значения уменьшают громкость левого канала, а правый не меняется, положительные значения уменьшают громкость правого канала, а левый не меняется. |

| | | |
|------------|---|---|
| MON_VOLUME | Управление громкостью самоконтроля в режиме передачи. | |
| Установить | MON_VOLUME:arg1; | arg1 – значение громкости, дБ. Диапазон значений от -60 до 0 дБ, при значении -60 дБ звук отсутствует. |
| Прочитать | MON_VOLUME; | |
| Ответ | MON_VOLUME:arg1; | |
| Тип | двунаправленное управление | |
| Пример | MON_VOLUME:-12; MON_VOLUME; | |

| | | |
|------------|--|---|
| MON_ENABLE | Enable/disable monitoring in TX mode. | |
| Set | MON_ENABLE:arg1; | arg1 – status indicator (enable – true, disable – false). |
| Read | MON_ENABLE; | |
| Reply | MON_ENABLE:arg1; | |
| Type | Bidirectional control | |
| Example | MON_ENABLE:true; MON_ENABLE:false; MON_ENABLE; | |

| | | |
|------------|--|---|
| AGC_MODE | Управление режимом работы АРУ приёмника. | |
| Установить | AGC_MODE:arg1,arg2; | arg1 – номер приёмника. arg2 – режим работы. Список поддерживаемых режимов: <ul style="list-style-type: none"> • normal; • fast; • off. |
| Прочитать | AGC_MODE:arg1; | |
| Ответ | AGC_MODE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | AGC_MODE:0,normal; AGC_MODE:1; | |

| | | |
|------------|-------------------------------------|---|
| AGC_GAIN | Управление усилением АРУ приёмника. | |
| Установить | AGC_GAIN:arg1,arg2; | arg1 – номер приёмника. arg2 – значение усиления, дБ. Диапазон значений от -20 до 120 дБ. |
| Прочитать | AGC_GAIN:arg1; | |
| Ответ | AGC_GAIN:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | AGC_GAIN:0,87; AGC_GAIN:1; | |

| | | |
|--------------|---|--|
| RX_NB_ENABLE | Включение/Выключение подавителя импульсных помех приёмника. | |
| Установить | RX_NB_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл – true, выкл – false). |
| Прочитать | RX_NB_ENABLE:arg1; | |
| Ответ | RX_NB_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_NB_ENABLE:0,true; RX_NB_ENABLE:1; | |

| | | |
|-------------|---|--|
| RX_NB_PARAM | Управление параметрами подавителя импульсных помех приёмника. | |
| Установить | RX_NB_PARAM:arg1,arg2,arg3; | arg1 – номер приёмника. arg2 – порог срабатывания, диапазон значений: 1 ... 100. arg3 – длительность импульса, диапазон значений: 1 ... 300. |
| Прочитать | RX_NB_PARAM:arg1; | |
| Ответ | RX_NB_PARAM:arg1,arg2,arg3; | |
| Тип | двунаправленное управление | |
| Пример | RX_NB_PARAM:0,70,25; RX_NB_PARAM:1; | |

| | | |
|---------------|---|--|
| RX_BIN_ENABLE | Включение/Выключение режима псевдостерео приёмника. | |
| Установить | RX_BIN_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл - true, выкл - false). |
| Прочитать | RX_BIN_ENABLE:arg1; | |
| Ответ | RX_BIN_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_BIN_ENABLE:0,true; RX_BIN_ENABLE:1; | |

| | | |
|--------------|---|--|
| RX_NR_ENABLE | Включение/Выключение фильтра подавляющего постоянную помеху в НЧ сигнале приёмника. | |
| Установить | RX_NR_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл - true, выкл - false). |
| Прочитать | RX_NR_ENABLE:arg1; | |
| Ответ | RX_NR_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_NR_ENABLE:0,true; | |

| | | |
|--|-----------------|--|
| | RX_NR_ENABLE:1; | |
|--|-----------------|--|

| | | |
|---------------|--|--|
| RX_ANC_ENABLE | Включение/Выключение процессора выделяющего речевой сигнал из шума НЧ сигнала приёмника. | |
| Установить | RX_ANC_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл - true, выкл - false). |
| Прочитать | RX_ANC_ENABLE:arg1; | |
| Ответ | RX_ANC_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_ANC_ENABLE:0,true; RX_ANC_ENABLE:1; | |

| | | |
|---------------|---|--|
| RX_ANF_ENABLE | Включение/Выключение фильтра подавляющего тональные сигналы в НЧ сигнале приёмника. | |
| Установить | RX_ANF_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл - true, выкл - false). |
| Прочитать | RX_ANF_ENABLE:arg1; | |
| Ответ | RX_ANF_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_ANF_ENABLE:0,true; RX_ANF_ENABLE:1; | |

| | | |
|---------------|---|--|
| RX_APF_ENABLE | Включение/Выключение фильтра, выделяющего полезный сигнал в НЧ сигнале приёмника. | |
| Установить | RX_APF_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл - true, выкл - false). |
| Прочитать | RX_APF_ENABLE:arg1; | |
| Ответ | RX_APF_ENABLE:arg1,arg2; | |

| | | |
|--------|---|--|
| Тип | двунаправленное управление | |
| Пример | RX_APF_ENABLE:0,true; RX_APF_ENABLE:1; | |

| | | |
|---------------|---|--|
| RX_DSE_ENABLE | Включение/Выключение процессора выполняющего пространственное позиционирование телеграфного сигнала относительно частоты настройки. | |
| Установить | RX_DSE_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл – true, выкл – false). |
| Прочитать | RX_DSE_ENABLE:arg1; | |
| Ответ | RX_DSE_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_DSE_ENABLE:0,true; RX_DSE_ENABLE:1; | |

| | | |
|--------------|---|--|
| RX_NF_ENABLE | Включение/Выключение модуля диапазонных notch фильтров. | |
| Установить | RX_NF_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл – true, выкл – false). |
| Прочитать | RX_NF_ENABLE:arg1; | |
| Ответ | RX_NF_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | RX_NF_ENABLE:0,true; RX_NF_ENABLE:1; | |

| | | |
|------------|---|-------------------------|
| LOCK | Управление блокировкой частоты настройки. | |
| Установить | LOCK:arg1,arg2; | arg1 – номер приёмника. |
| Прочитать | LOCK:arg1; | |

| | | |
|--------|----------------------------|--|
| Ответ | LOCK:arg1,arg2; | arg2 – флаг состояния (вкл – true, выкл – false). |
| Тип | двунаправленное управление | |
| Пример | LOCK:0,true; LOCK:1; | |

| | | |
|------------|---|---|
| SQL_ENABLE | Включение/Выключение порогового шумоподавителя. | |
| Установить | SQL_ENABLE:arg1,arg2; | arg1 – номер приёмника. arg2 – флаг состояния (вкл – true, выкл – false). |
| Прочитать | SQL_ENABLE:arg1; | |
| Ответ | SQL_ENABLE:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | SQL_ENABLE:0,true; SQL_ENABLE:1; | |

| | | |
|------------|--|---|
| SQL_LEVEL | Управление порогом срабатывания порогового шумоподавителя. | |
| Установить | SQL_LEVEL:arg1,arg2; | arg1 – номер приёмника. arg2 – порог срабатывания, дБ. Диапазон значений от -140 до 0 дБ. |
| Прочитать | SQL_LEVEL:arg1; | |
| Ответ | SQL_LEVEL:arg1,arg2; | |
| Тип | двунаправленное управление | |
| Пример | SQL_LEVEL:0,-83; SQL_LEVEL:1; | |

| | | |
|-------------|---|---|
| DIGL_OFFSET | Управление смещением частоты для вида связи DIGL. | |
| Установить | DIGL_OFFSET:arg1; | arg1 – частота смещения, Гц. Диапазон значений от 0 Гц до 4 кГц. |
| Прочитать | DIGL_OFFSET; | |
| Ответ | DIGL_OFFSET:arg1; | |
| Тип | двунаправленное управление | |
| Пример | DIGL_OFFSET:1500; DIGL_OFFSET; | |

| | | |
|-------------|---|---|
| DIGU_OFFSET | Управление смещением частоты для вида связи DIGU. | |
| Установить | DIGU_OFFSET:arg1; | arg1 – частота смещения, Гц. Диапазон значений от 0 Гц до 4 кГц. |
| Прочитать | DIGU_OFFSET; | |
| Ответ | DIGU_OFFSET:arg1; | |
| Тип | двунаправленное управление | |
| Пример | DIGU_OFFSET:2200; DIGU_OFFSET; | |

4.3 Команды однонаправленного управления

| | | |
|------------|---|--|
| TX_ENABLE | Уведомляет клиентов о том, что передача разрешена или запрещена. | |
| Ответ | TX_ENABLE:arg1,arg2; | arg1 – номер приёмника/трансивера; arg2 – передача разрешена (true)/передача запрещена (false). |
| Тип | однонаправленное управление | |
| Пример | TX_ENABLE:0,true; | |
| Дополнение | Отправляется клиенту при подключении, если используется трансивер, тогда будет отправляться при смене диапазона, если разрешение на использование передатчика будет меняться. | |

| | | |
|--------------------|--|---|
| CW_MACROS_SPEED_UP | Увеличение скорости телеграфирования для макросов. | |
| Установить | CW_MACROS_SPEED_UP:arg1; | arg1 – количество WPM на сколько увеличиться скорость телеграфирования. |
| Тип | однонаправленное управление | |
| Пример | CW_MACROS_SPEED_UP:7; CW_MACROS_SPEED_UP:2; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|----------------------|--|--|
| CW_MACROS_SPEED_DOWN | Уменьшение скорости телеграфирования для макросов. | |
| Установить | CW_MACROS_SPEED_DOWN:arg1; | arg1 – количество WPM на сколько уменьшится скорость телеграфирования. |
| Тип | однонаправленное управление | |
| Пример | CW_MACROS_SPEED_DOWN:7; CW_MACROS_SPEED_DOWN:2; | |

| | | |
|------------|-------------------------------|--|
| Дополнение | Отправляется только клиентом. | |
|------------|-------------------------------|--|

| | | |
|------------|---|---|
| SPOT | Передача спота для отображения. | |
| Установить | SPOT:arg1,arg2,arg3,arg4,arg5; | arg1 – позывной. arg2 – вид связи. arg3 – частота, Гц. arg4 – цвет ARGB. arg5 – дополнительный текст. |
| Тип | однаправленное управление | |
| Пример | SPOT:RN6LHF,CW,7100000,16711680,ANY_TEXT; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|-------------|-------------------------------|------------------|
| SPOT_DELETE | Удалить спот. | |
| Установить | SPOT_DELETE:arg1; | arg1 – позывной. |
| Тип | однаправленное управление | |
| Пример | SPOT_DELETE:RN6LHF; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|---------------|---|--|
| IQ_SAMPLERATE | Управление частотой дискретизации IQ сигнала. | |
| Установить | IQ_SAMPLERATE:arg1; | arg1 – частота дискретизации, Гц. |
| Ответ | IQ_SAMPLERATE:arg1; | |
| Тип | однаправленное управление | Поддерживаемые частоты дискретизации: 48/96/192/384 кГц |
| Пример | IQ_SAMPLERATE:48000; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|------------------|---|---|
| AUDIO_SAMPLERATE | Управление частотой дискретизации НЧ аудио сигнала. | |
| Установить | AUDIO_SAMPLERATE:arg1; | arg1 – частота дискретизации, Гц. Поддерживаемые частоты дискретизации: 8/12/24/48 кГц |
| Ответ | AUDIO_SAMPLERATE:arg1; | |
| Тип | однаправленное управление | |
| Пример | AUDIO_SAMPLERATE:12000; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|------------|-------------------------------|-------------------------|
| IQ_START | Запустить поток сигнала IQ. | |
| Установить | IQ_START:arg1; | arg1 – номер приёмника. |
| Тип | однаправленное управление | |
| Пример | IQ_START:0; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|------------|-------------------------------|-------------------------|
| IQ_STOP | Остановить поток сигнала IQ. | |
| Установить | IQ_STOP:arg1; | arg1 – номер приёмника. |
| Тип | однонаправленное управление | |
| Пример | IQ_STOP:0; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|-------------|----------------------------------|-------------------------|
| AUDIO_START | Запустить поток НЧ аудиосигнала. | |
| Установить | AUDIO_START:arg1; | arg1 – номер приёмника. |
| Тип | однаправленное управление | |

| | | |
|------------|-------------------------------|--|
| Пример | AUDIO_START:0; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|------------|-----------------------------------|-------------------------|
| AUDIO_STOP | Остановить поток НЧ аудиосигнала. | |
| Установить | AUDIO_STOP:arg1; | arg1 – номер приёмника. |
| Тип | однаправленное управление | |
| Пример | AUDIO_STOP:0; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|----------------|---|-------------------------|
| LINE_OUT_START | Запустить аудиопоток из линейного аудио выхода. | |
| Установить | LINE_OUT_START:arg1; | arg1 – номер приёмника. |
| Тип | однаправленное управление | |
| Пример | LINE_OUT_START:0; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|---------------|--|-------------------------|
| LINE_OUT_STOP | Остановить аудиопоток из линейного аудио выхода. | |
| Установить | LINE_OUT_STOP:arg1; | arg1 – номер приёмника. |
| Тип | однаправленное управление | |
| Пример | LINE_OUT_STOP:0; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|-------------------------|---|---|
| LINE_OUT_RECORDER_START | Начать запись аудиопотока из линейного аудио выхода. | |
| Установить | LINE_OUT_RECORDER_START:arg1,arg2; | arg1 – номер приёмника. arg2 – максимальное время записи (не более 300 сек). |
| Тип | однаправленное управление | |
| Пример | LINE_OUT_RECORDER_START:0,250; | |
| Дополнение | Отправляется только клиентом. По истечении времени запись удаляется, чтобы сохранить запись в файл необходимо в заданном временном интервале выслать команду LINE_OUT_RECORDER_SAVE. | |

| | | |
|------------------------|--|---|
| LINE_OUT_RECORDER_SAVE | Сохранить запись аудиопотока из линейного аудио выхода в файл. | |
| Установить | LINE_OUT_RECORDER_SAVE:arg1,arg2; | arg1 – номер приёмника. arg2 – полное имя файла. |
| Тип | однаправленное управление | |
| Пример | LINE_OUT_RECORDER_SAVE:0,home/user_name/record_dir/file_name.wav; LINE_OUT_RECORDER_SAVE:0,home/user_name/record_dir/file_name.mp3; LINE_OUT_RECORDER_SAVE:0,D:\record_dir\file_name.mp3; | |
| Дополнение | Отправляется только клиентом. Поддерживается формат записи WAVE и MP3. Для windows ОС путь содержит запрещённый символ ":", он заменяется на символ " ", также можно использовать прямой и обратный слэш в пути к файлу. | |

| | | |
|-------------------------|---|-------------------------|
| LINE_OUT_RECORDER_BREAK | Прекратить запись аудиопотока из линейного аудио выхода и удалить накопленную запись. | |
| Установить | LINE_OUT_RECORDER_BREAK:arg1; | arg1 – номер приёмника. |
| Тип | однаправленное управление | |
| Пример | LINE_OUT_RECORDER_BREAK:0; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|------------|-------------------------------|--|
| SPOT_CLEAR | Удаление всех спотов. | |
| Установить | SPOT_CLEAR; | |
| Тип | однонаправленное управление | |
| Пример | SPOT_CLEAR; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|---------------------------|--|---|
| TX_STREAM_AUDIO_BUFFERING | Установка времени буферизации передаваемого сигнала. | |
| Установить | TX_STREAM_AUDIO_BUFFERING:arg1; | arg1 – время накопления, мс значение можно задать от 50 до 500 |
| Тип | однонаправленное управление | |
| Пример | TX_STREAM_AUDIO_BUFFERING:150; TX_STREAM_AUDIO_BUFFERING:270; | |
| Дополнение | Отправляется только клиентом. По умолчанию установлено 50 мс. | |

| | | |
|----------------------|---|--|
| AUDIO_STREAM_SAMPLES | Установка количества сэмплов передаваемых в одном пакете. | |
| Установить | AUDIO_STREAM_SAMPLES:arg1; | arg1 – количество сэмплов указываемое в поле Stream.length значение можно задать от 100 до 2048 сэмплов |
| Тип | однаправленное управление | |
| Пример | AUDIO_STREAM_SAMPLES:200; AUDIO_STREAM_SAMPLES:512; | |
| Дополнение | <p>Отправляется только клиентом.</p> <p>По умолчанию для каждой частоты дискретизации установлено своё количество сэмплов (указываемое в поле Stream.length):</p> <ul style="list-style-type: none">• при 48 кГц - 2048 сэмплов• при 24 кГц - 1024 сэмплов• при 12 кГц - 512 сэмплов• при 8 кГц - 256 сэмплов <p>После установки количества сэмплов это значение будет применяться для всех частот дискретизации.</p> <p>Рекомендуется задавать количество сэмплов таким, чтобы длительность воспроизведения была не меньше 10 мс иначе сигнал, передаваемый на передачу, может иметь повреждения.</p> <p>Рекомендуемое минимальное количество сэмплов:</p> <ul style="list-style-type: none">• 48 кГц - 512 сэмплов• 24 кГц - 256 сэмплов• 12 кГц - 128 сэмплов• 8 кГц - 100 сэмплов | |

| | | |
|-----------------------|--|---|
| AUDIO_STREAM_CHANNELS | Установка количества каналов аудио потока. | |
| Установить | AUDIO_STREAM_CHANNELS:arg1; | arg1 – количество каналов Поддерживается 1 или 2 канала. |
| Тип | однаправленное управление | |
| Пример | AUDIO_STREAM_CHANNELS:1; AUDIO_STREAM_CHANNELS:2; | |
| Дополнение | Отправляется только клиентом. По умолчанию установлено количество каналов равное 2. | |

| | | |
|--------------------------|--|-------------------------------------|
| AUDIO_STREAM_SAMPLE_TYPE | Установка формата сэмплов для аудио потока. | |
| Установить | AUDIO_STREAM_SAMPLE_TYPE:arg1; | arg1 – идентификатор формата сэмпла |
| Тип | однонаправленное управление | Поддерживаемые форматы: |
| Пример | AUDIO_STREAM_SAMPLE_TYPE:int24; AUDIO_STREAM_SAMPLE_TYPE:float32; | |
| Дополнение | Отправляется только клиентом. Формат сэмпла по умолчанию float32. | |

4.4 Команды уведомления

| | | |
|-----------------|--|---|
| CLICKED_ON_SPOT | Уведомляет о клике мышью по споту на панораме ExpertSDR3. (устаревшая команда) | |
| Установить | CLICKED_ON_SPOT:arg1,arg2; | arg1 – позывной. arg2 – частота, Гц. |
| Тип | однонаправленное управление | |
| Пример | CLICKED_ON_SPOT:RN6LHF,7147500; | |
| Дополнение | Отправляется только сервером. | |

| | | |
|--------------------|---|----------------------------|
| RX_CLICKED_ON_SPOT | Уведомляет о клике мышью по споту на панораме ExpertSDR3. | |
| Установить | RX_CLICKED_ON_SPOT:arg1,arg2,arg3,arg4; | arg1 – номер приёмника. |
| Тип | однаправленное управление | arg2 – номер канала (A/B). |
| Пример | RX_CLICKED_ON_SPOT:0,1,RN6LHF,7147500; | arg3 – позывной. |
| | | arg4 – частота, Гц. |

| | | |
|------------|-------------------------------|--|
| Дополнение | Отправляется только сервером. | |
|------------|-------------------------------|--|

| | | |
|---------------|-------------------------------|---|
| TX_FOOTSWITCH | Сигнал нажатия на педаль РТТ. | |
| Установить | TX_FOOTCWATCH:arg1,arg2; | arg1 – номер приёмника. arg2 – состояние педали (нажата - true, отжата - false) |
| Тип | однонаправленное управление | |
| Пример | TX_FOOTCWATCH:0,true; | |
| Дополнение | Отправляется только сервером. | |

| | | |
|--------------|--|------------------------------|
| TX_FREQUENCY | Уведомление о текущей частоте передатчика. | |
| Установить | TX_FREQUENCY:arg1; | arg1 – частота передачи, Гц. |
| Тип | однаправленное управление | |
| Пример | TX_FREQUENCY:7140000; | |
| Дополнение | Отправляется только сервером. | |

| | | |
|------------|---|--|
| APP_FOCUS | Статус главного окна программы ExpertSDR3 (в фокусе или нет). | |
| Установить | APP_FOCUS:arg1; | arg1 – состояние фокуса. (в фокусе - true, не в фокусе - false) |
| Тип | однаправленное управление | |
| Пример | APP_FOCUS:true; | |
| Дополнение | Отправляется только сервером. | |

| | | |
|--------------|--|--|
| SET_IN_FOCUS | Сделать главное окно программы ExpertSDR3 активным (в фокусе). | |
| Установить | SET_IN_FOCUS; | |
| Тип | однаправленное управление | |

| | | |
|------------|-------------------------------|--|
| Пример | SET_IN_FOCUS:true; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|------------|--|--|
| KEYER | Сигнал нажатия на телеграфный ключ (не автоматический). | |
| Установить | KEYER:arg1,arg2,arg3; | arg1 – номер передатчика. |
| Тип | однаправленное управление | arg2 – состояние нажатия (нажата - true, отжата - false) |
| Пример | KEYER:0,true,120; | arg3 – длительность предыдущего знака, в мс. |
| Дополнение | <p>Отправляется только клиентом.</p> <p>Эта команда была модифицирована для работы с RadioSync. Основное назначение передача телеграфных макросов через COM-порт, для сохранения длительности точек и тире.</p> <p>Алгоритм работы:</p> <p>При первом нажатии на “клоподав” TCI клиент высылает команду <code>keyer:0,true,0;</code> в ExpertSDR3, что инициирует передачу телеграфного знака. TCI клиент фиксирует время начала этого события и когда приходит событие отпускания ключа, TCI клиент фиксирует разницу во времени между двумя событиями, высылая команду, например: <code>keyer:0,false,142;</code> где третий аргумент это зафиксированный временной интервал. Это сделано для того, чтобы телеграфное аппаратное ядро знало точное время проигрывания каждого знака. Все последующие нажатия и отжатия ключа сопровождаются определением временных интервалов, что позволяет минимизировать «пьяного матроса».</p> | |

| | | |
|-------------------|--|---|
| RX_SENSORS_ENABLE | Включение отправки уровня сигнала в полосе приёма. | |
| Установить | RX_SENSORS_ENABLE:arg1,arg2; | arg1 –флаг включения (вкл - true, выкл - false) |
| Тип | однаправленное управление | |
| Пример | RX_SENSORS_ENABLE:true; RX_SENSORS_ENABLE:true,200; | arg2 – интервал отправки, мс (от 30 до 1000 мс, |

| | | |
|------------|-------------------------------|--------------------------|
| | | необязательный параметр) |
| Дополнение | Отправляется только клиентом. | |

| | | |
|-------------------|--|---|
| TX_SENSORS_ENABLE | Включение отправки показаний измерителей передатчика. | |
| Установить | TX_SENSORS_ENABLE:arg1,arg2; | arg1 –флаг включения (вкл - true, выкл - false) arg2 – интервал отправки, мс (от 30 до 1000 мс, необязательный параметр) |
| Тип | однаправленное управление | |
| Пример | TX_SENSORS_ENABLE:true; TX_SENSORS_ENABLE:true,200; | |
| Дополнение | Отправляется только клиентом. | |

| | | |
|------------|---|---|
| RX_SENSORS | Показания измерителя мощности сигнала в полосе приёма. | |
| Установить | RX_SENSORS:arg1,arg2; | arg1 – номер приёмника arg2 – мощность сигнала, дБм |
| Тип | однаправленное управление | |
| Пример | RX_SENSORS:0,-71.5; RX_SENSORS:1,-112.7; | |
| Дополнение | Отправляется только сервером. Команда устарела и была заменена RX_CHANNEL_SENSORS, будет удалена в будущем. | |

| | | |
|------------|--------------------------------------|---|
| TX_SENSORS | Показания измерителей передатчика. | |
| Установить | TX_SENSORS:arg1,arg2,arg3,arg4,arg5; | arg1 – номер передатчика arg2 – уровень микрофонного сигнала, дБм arg3 – мощность сигнала на выходе передатчика, Вт (RMS) arg4 – пиковая мощность сигнала на выходе |
| Тип | однаправленное управление | |
| Пример | TX_SENSORS:0, -27.2,47.4.67.5,1.7; | |

| | | |
|------------|-------------------------------|-------------------------------|
| | | передатчика, Вт arg5 – КСВ |
| Дополнение | Отправляется только сервером. | |

4.5 Новые команды в этой версии

| | | |
|------------|--|---|
| VFO_LOCK | Уведомление о блокировке управления частотой клиентом. | |
| Установить | VFO_LOCK:arg1,arg2,arg3; | arg1 – номер приёмника; arg2 – номер канала (A/B); arg3 – флаг блокировки (true/false). |
| Прочитать | VFO_LOCK:arg1,arg2; | |
| Ответ | VFO_LOCK:arg1,arg2,arg3; | |
| Тип | Однонаправленное управление | |
| Пример | VFO_LOCK:0,1,true; VFO_LOCK:1,1,false; | |
| Дополнение | Отправляется только сервером. | |

| | | |
|--------------------|---|--|
| RX_CHANNEL_SENSORS | Показания измерителя мощности сигнала в полосе приёма. | |
| Установить | RX_CHANNEL_SENSORS:arg1,arg2,arg3; | arg1 – номер приёмника arg2 – номер канала (A/B); arg3 – мощность сигнала, дБм |
| Тип | однонаправленное управление | |
| Пример | RX_CHANNEL_SENSORS:0,0,-71.5; RX_CHANNEL_SENSORS:1,1,-112.7; | |
| Дополнение | Отправляется только сервером. | |