

Технический перевод
Дистанционное управление

Стр. 1

5. Дистанционное управление.....	5.3
5.1. Введение.....	5.3
5.2. Переключение на дистанционное управление.....	5.4
5.2.1. Установка параметров передачи данных.....	5.4
5.2.2. Отображение удаленного управления.....	5.5
5.2.3. Возврат к непосредственному управлению.....	5.5
5.3. Краткие инструкции.....	5.6
5.4. Зависящие от устройства сообщения (команды и ответы).....	5.7
5.5. Структура и синтаксис зависящих от устройства сообщений.....	5.8
5.5.1. Введение в SCPI.....	5.8
5.5.2. Структура команд.....	5.8
5.5.3. Структура программного сообщения.....	5.10
5.5.4. Ответы на запросы.....	5.11
5.5.5. Параметры.....	5.11
5.5.6. Обзор синтаксиса элементов.....	5.13
5.6. Описание команд.....	5.14
5.6.1. Индексация.....	5.14
5.6.1-1. Таблица команд.....	5.14
5.6.2. Общие команды.....	5.15
5.6.3. Команды функции PLAY.....	5.17
5.6.4. Команды функции RECORD.....	5.26
5.6.5. Команды функции SETUP.....	5.28

5.6.6. Команды SCPI-регистра.....	5.32
5.7. Инструментальная модель и обработка команд.....	5.36
5.7.1. Входной блок.....	5.36
5.7.2. Идентификация команд.....	5.36
5.7.3. Набор данных и аппаратное устройство.....	5.37
5.7.4. Блок вывода.....	5.37
5.7.5. Последовательность команд и их синхронизация.....	5.37
5.8. Система отчетов о состоянии системы.....	5.38
5.8.1. Структура регистра статуса SCPI.....	5.38
5.8.2. Обзор регистра статуса.....	5.40
5.8.3. Описание регистра статуса.....	5.41
5.8.3.1. Биты статуса (STB) и службы включения запроса (SRE).....	5.41
5.8.3.2. Регистр статуса состояния (ESR) и статуса состояния событий (ESE).....	5.42
5.8.4. Использование отчетов о состоянии системы.....	5.43
5.8.4.1. Запрос на обслуживание (SRQ).....	5.43
5.8.4.2. Запросы.....	5.43
5.8.4.3. Запрос очереди ошибок.....	5.43
5.8.5. Сброс отчетов о состоянии системы.....	5.44
5.9. Принципы удаленного управления через Ethernet.....	5.45
5.9.1. Схема IEEE 1174 (режимы работы и последовательности управления).....	5.46
5.9.1.1. Режимы последовательного интерфейса IEEE 1174.0.....	5.46
5.9.1.2. Режимы последовательного интерфейса IEEE 1174.1.....	5.47

5.9.1.3. Спецификация IEEE для локальных сетей.....	5.48
5.9.1.4. Краткий обзор возможностей.....	5.49
5.9.1.5. Дополнительные секции управления Rohde&Schwarz.....	5.49
5.9.2. Передача последовательности управляющих команд.....	5.50
5.9.3. Функциональность.....	5.52
5.9.3.1. Сброс операнда.....	5.53
5.9.3.2. Заявка на сервис.....	5.54
5.9.3.3. Последовательный опрос.....	5.54
5.9.3.4. Механизм запуска.....	5.55
5.9.3.5. Управление потоком данных.....	5.55
5.9.3.6. Вторичные адреса.....	5.55
5.9.4. Команды Ethernet-драйвера в измерительном приборе.....	5.56
5.9.4.1. Структура драйвера в версии IEEE 1174.0.....	5.57
5.9.4.2. Структура драйвера в версии IEEE 1174.1.....	5.58
5.9.4.3. Структура драйвера в версии IEEE 1174.2.....	5.58
5.9.5. Пример дистанционного управления Ethernet.....	5.59
5.9.6. Дистанционное управление через RS-232-C интерфейс.....	5.63

Стр. 3

5. Дистанционное управление

5.1. Введение

Программа оборудована Ethernet-интерфейсом. Ethernet-интерфейс оснащен стандартом для 4-парного неэкранированного кабеля категории 5.

Используемый протокол передачи данных – TCP/IP.

С помощью дополнительного оборудования линии «DVRG_Remote.exe» (см. раздел 4.2.4.) возможно использование RS-232-C интерфейса в качестве средства дистанционного управления.

Разъемы расположены на задней панели, где контроллеры могут быть соединены для дистанционного управления.

Сообщения программы, переданные в информационный раздел протокола передачи данных, одинаковы для обоих интерфейсов. Набор команд поддерживает версию SCPI 1996.0 (стандартные команды для инструментов программирования). Стандарт SCPI основан на стандарте IEEE 488.2 и направлен на универсализацию устройство-ориентированных команд, обработки ошибок и регистров состояния системы.

Для четкого понимания этой главы требуется базовое знание SCPI-программирования и управления контроллерами.

Требования SCPI-стандарта относительно синтаксиса командного языка, обработки ошибок и конфигурации регистров состояния системы описаны в деталях в соответствующем разделе. Таблицы обеспечивают быстрый обзор команд, внедренных в аппарат, и присваивание бит регистрами состояния системы. Таблицы дополнены подробным описанием команд и регистров. Для понимания описания команд необходимо базовое знание инструкции по эксплуатации прибора.

Все примеры, использованные в описании сетевого управления, реализовывают стандартные панели совместимости интерфейса.

Стр. 4

5.2. Переключение на дистанционное управление

После включения питания прибор находится в управляемом вручную режиме функционирования. Переключение на дистанционное управление (REMOTE) осуществляется при получении команды через интерфейс дистанционного управления. DVRG остается на дистанционном управлении до тех пор, пока не осуществляется перевод вручную обратно на непосредственное управление. Переключение с непосредственного на дистанционное управление и наоборот не оказывает влияния на настройки прибора.

[изображение дисплея, на котором написано: управляется удаленно]

5.2.1. Настройка параметров передачи данных

Для обеспечения безошибочной и корректной передачи данных ее настройки должны быть одинаковы как на самом DVRG, так и на контроллере.

Вручную:

Открыть меню SETUP/COMM и установить требующиеся параметры передачи данных.

RS232: BAUDRATE (скорость передачи данных), DATABITS (биты данных), STOPBITS (то же), PARITY (контроль четности) и PROTOCOL (протокол).

Ethernet: IP ADDRESS (IP-адрес), SUBNET MASK (маска подсети), GATEWAY (шлюз), PORT (порт) и MODE (режим работы).

Выберите параметр, который хотите изменить, с помощью курсора, и начните ввод, нажав ENTER. Изменяйте значения, используя кнопки ↑ и ↓. Подтвердите ввод, нажав ENTER, или отмените его с помощью клавиши ESC. Закройте меню с помощью ESC.

Дистанционное управление:

Каждое сообщение программы должно заканчиваться регистром команд и переходом на новую строку (0x0D, 0x0A). Два символа – завершающие. Ответы устройства также заканчиваются этими знаками.

RS232:

SYSTEM:COMMunicate:SERial:RECEive:BAUD9600

SYSTEM:COMMunicate:SERial:RECEive:BITS 8

SYSTEM:COMMunicate:SERial:RECEive:SBITS 1

SYSTEM:COMMunicate:SERial:RECEive:PARity NONE

SYSTEM:COMMunicate:SERial:RECEive:PACe ACK

Ethernet:

SYSTEM:COMMunicate:INet:ADDRess 196, 168, 1, 4

SYSTEM:COMMunicate:INet:SMASk 255, 155, 255, 0

SYSTEM:COMMunicate:INet:GATeway 196, 168, 1, 0

SYSTEM:COMMunicate:INet:PORT 4002

SYSTEM:COMMunicate:INet: MODE 1

- Замечание: если параметры передачи данных интерфейса дистанционного управления должны быть изменены, убедитесь, что параметры передачи данных контроллера изменены соответствующим образом.
- Если вы хотите удаленно управлять DVRG через Ethernet, свяжитесь со своим администратором сети, чтобы определиться с необходимыми параметрами передачи данных. Неверные настройки могут вызвать ошибку сети и невозможность установить связь.

5.2.2. Просмотр дистанционного управления

При включенном дистанционном управлении отображается его меню.

5.2.3. Возвращение в режим непосредственного управления

Возвращение в режим непосредственного управления может быть осуществлено только через переднюю панель посредством выбора опции LOCAL в меню дистанционного управления.

- Замечание: перед переключением обработка команд должна быть завершена, потому что иначе прибор снова возвращается к работе в режиме дистанционного управления.
- Переключение на непосредственное управление может быть отменено командой SYSTem:KLOCK ON для предупреждения неумышленного переключения.
- Отмена переключения на непосредственное управление может быть осуществлена командой SYSTem:KLOCK OFF.

Стр. 6

5.3. Краткие указания

Нижеследующие краткие и простые инструкции позволяют пользователю быстро привести оборудование в рабочее состояние и настроить основные функции.

Каждое сообщение программы должно заканчиваться регистром команд и переходом на новую строку (0x0D, 0x0A). Два символа – завершающие. Ответы устройства также заканчиваются этими знаками.

Ethernet:

Соедините DVRG с 10-мегабитным разветвителем, используя экранированную витую пару с выходом RJ-45.

- В меню DVRG (пункт меню SETUP/COMM) установите параметры передачи данных: IP-адрес, маску подсети, шлюз, порт и режим работы. Свяжитесь со своим администратором сети, чтобы определиться с необходимыми параметрами передачи данных, поскольку неверные настройки могут вызвать ошибку сети и невозможность установить связь.

Откройте программу гнездового интерфейса.

Создайте двунаправленный канал типа SOCK_STREAM с помощью socket ()

Установите соединение с R&S DVRG с помощью connect ()

Посылайте или получайте информацию с помощью send () или recv ()

Пошлите набор команд R&S DVRG с помощью send ()

Вернитесь к непосредственному управлению: выберите опцию LOCAL в меню дистанционного управления REMOTE.

RS232:

Откройте программу “DVRG_Remote.exe” на R&S DVRG.

Соедините R&S DVRG с управляющим компьютером нуль-модемным кабелем.

На R&S DVRG (опция меню SETUP/COMM) установите бод 9600/мин, 8 бит, 1 стоп-бит, контроль по четности отсутствует и скорость не установлена.

Пошлите инструкцию DVRG.

Вернитесь к непосредственному управлению: в меню дистанционного управления REMOTE выберите опцию LOCAL.

Стр. 7

5.4. Зависящие от устройства сообщения (команды и ответы)

Зависящие от устройства сообщения кодируются с помощью ASCII/ISO. Они распределены в соответствии с направлением, в котором они посылаются через интерфейс.

Команды – это сообщения, посылаемые контроллером устройству. Они контролируют функции прибора и запрашивают информацию. Команды классифицируются следующими способами:

1) в соответствии с эффектом, оказываемым на устройство:

- Команды установления соединения: команды запуска настроек прибора, такие, напр., как сброс параметров устройства
- Запросы – запрашивают предоставление информации устройством, напр., для идентификации такового или активного входа

2) в соответствии с их предназначением в стандарте IEEE 488.2:

- Общие команды – строго заданы стандартом IEEE 488.2, как того требует их предназначение и номенклатура. Они включают в себя такие функции, как управление стандартизированными регистрами команд, настройками, регулировкой и пр.
- Устройство-ориентированные команды – относятся к функциям, которые зависят от характеристик прибора, напр., регулировка частоты. Часть этих команд также стандартизирована соответственно SCPI.

Ответы – сообщения, посылаемые прибором контроллеру по запросу. Они могут содержать результаты, настройки прибора или информацию о регистре устройства и его состоянии.

Структура и синтаксис зависящих от устройства сообщений описаны в соответствующем разделе. Команды приведены и объяснены в разделе 5.6. Для понимания описания команд необходимо базовое знание инструкции к R&S DVRG.

Стр. 8

5.5. Структура и синтаксис зависящих от устройства сообщений

5.5.1. Введение в SCPI

SCPI (стандартные команды для программируемых приборов) – стандартизированный набор команд для программирования, не зависящий от типа прибора или его производителя. Цель SCPI – максимально стандартизировать зависящие от устройств команды. Для этого была разработана модель прибора, которая объединяет одинаковые функции в

устройстве или нескольких устройствах. Системы команд были созданы и привязаны к этим функциям, чтобы была возможно обращаться к функциям с помощью одинаковых команд. Команды упорядочены в иерархической структуре.

На схеме 5-2 показана древовидная структура команд, использующая командный пакет SYSTem для контроля настроек прибора. Остальные примеры синтаксиса и структуры команд взяты из этой системы.

SCPI основана на стандарте IEEE 488.2, к примеру, здесь используется такой же синтаксис элементов, который обозначен в IEEE 488.2. Синтаксис ответов частично подчинен строгим правилам, изложенным в стандарте IEEE 488.2.

5.5.2. Структура команд

Команды состоят из заголовков и одного или нескольких параметров. Заголовок и параметры разделены пробелом (код ASCII от 0 до 9, от 11 до 32 в двоичной системе, пробел = 32). Заголовки могут состоять из нескольких ключевых слов. Форма запроса генерируется добавлением запроса напрямую к заголовку.

Замечание:

- Команды, использованные в приведенных ниже примерах, не обязательно могут выполняться для конкретного прибора.

Общие команды – независимые от устройства команды, состоящие из заголовка, которому предшествует звездочка (*). После заголовка может идти один или несколько параметров.

Пример:

- *RST – RESET, перезапускает прибор
- *ESE 235 – EVENT STATUS ENABLE, задает скорость считывания статуса состояния прибора
- *ESR? – EVENT STATUS QUERY, запрашивает состояние регистра состояния системы

Зависящие от устройства команды

Иерархия: зависящие от устройства команды подчиняются иерархической системе. Различные уровни представлены отдельными заголовками.

Заголовки высшего (корневого) уровня имеют только одно ключевое слово. Это ключевое слово оперирует все системой команд.

Пример: `SYSTem` – это ключевое слово указывает на всю систему команд `SYSTem`.

Для команд более низкого уровня должен быть обозначен полный путь, начиная с высшего уровня самой левой позиции. Индивидуальные ключевые слова должны быть разделены с помощью двоеточия.

Пример: `SYSTem:COMMunicate:SERial:RECeive:BAUD 9600`

Команда занимает пятый уровень системы `SYSTem` и устанавливает скорость передачи данных 9600 бод.

Стр. 9

Рис.5.2. [Блок-схема с примером иерархической структуры команд].

Представлена древовидная структура командной системы `SCPI`, в качестве примера показан блок `SYSTem`.

Полная и краткая формы: ключевые слова имеют полную и краткую формы. Они могут быть введены в одной из них, никакие другие сокращения не разрешено использовать.

Пример: `SYSTem:COMMunicate:SERial: BAUD 9600`

Следующая команда идентична первой: `SYST:COMM:SER:BAUD 9600`

Замечание:

- Краткая форма команд набирается в верхнем регистре, полная форма представляет собой все ключевое слово. Буквы как верхнего, так и нижнего регистра используются только для различия в инструкции к прибору, сам прибор не различает типы регистра.

Параметр: параметр должен быть отделен от заголовка пробелом. Если команда содержит в себе несколько параметров, они должны быть разделены запятой. Некоторые запросы разрешают спецификацию параметров `MINimum`, `MAXimum` и `DEFault`.

Пример: `SYSTem:COMM:SER:REC:BAUD? MAXimum RESPONSE: 19200`

Этот запрос показывает максимальное значение скорости передачи данных.

Числовой индекс: если в приборе есть несколько одинаковых функций или свойств, например входов, требуемая функция может быть выбрана с помощью индекса. Команды, запрошенные без индекса, воспринимаются как имеющие индекс 1.

Пример: `SYSTem:COMMunicate:SERial2:BAUD 4800`

Эта команда устанавливает скорость подключения последовательной передачи данных `SERial2 4800` бод.

Стр. 10

Дополнительные ключевые слова

В некоторых системах команд можно добавить или исключить определенные ключевые слова. Они показаны в инструкции к устройству в квадратных скобках. Для совместимости с SCPI-стандартом устройство должно распознавать полную длину команды. Некоторые команды становятся значительно короче при исключении определенных ключевых слов.

Пример: `SYSTem:COMMunicate:SERial [RECeive]:BAUD 4800`

Эта команда устанавливает скорость последовательной передачи данных `4800` бод.

Следующая команда идентична первой:

`SYSTem:COMMunicate:SERial:BAUD 4800`

Замечание:

- Дополнительное ключевое слово может не удалиться, если оно распространяется на несколько функций посредством числового индекса.

5.5.3. Структура программного сообщения

Программное сообщение может состоять из нескольких команд.

Программное сообщение оканчивается регистром команд и переходом на новую строку.

Несколько команд в программном сообщении разделяются точкой с запятой. Если следующая команда принадлежит другой системе команд, после точки с запятой идет двоеточие.

Пример: `SYSTem:TIME 20, 30, 00; :CONFigure:PLENght 188`

Это программное сообщение состоит из двух команд. Первая принадлежит системе SYSTem и используется для установки времени на системных часах. Вторая принадлежит системе CONFigure и выбирает длину пакета.

Если последовательно идущие команды принадлежат одной и той же системе и поэтому имеют один или несколько общих уровней, программное сообщение может быть укорочено. Вторая команда, следующая после точки с запятой, в таком случае начинается на уровне ниже общих. Двоеточие после точки с запятой в таком случае должно быть отменено.

Пример: SYSTem:COMM:SERial:BAUD 4800; : SYSTem:SOMM:SERial:BITS 8

Это программное сообщение состоит из двух команд, разделенных точкой с запятой. Обе команды принадлежат системе SYSTem и ее подсистеме COMMunicate:SERial, т.е. у них три общих уровня.

В сокращенном варианте программного сообщения вторая команда начинается уровнем ниже SYSTem:COMMunicate:SERial. Двоеточие после точки с запятой не требуется.

Сокращенная форма этого программного сообщения:

SYST:COMM:SER:BAUD 4800; BITS 8

Каждое новое программное сообщение начинается с полного пути.

Пример: SYSTem:COMMunicate:SERial:BAUD 4800

SYSTem:COMMunicate:SERial:BITS 8

Стр. 11

5.5.4. Ответы на запросы

Несмотря на точную отдельную спецификацию, запросы различаются для каждой команды установления соединения. Запрос создается добавлением знака вопроса к соответствующей команде. Некоторые правила SCPI, касающиеся ответов на запросы, строже представленных в стандарте IEEE 488.2:

1. Требуемый параметр посылается без заголовка

Пример: SYSTem:COMMunicate:SERial:BAUD ? Ответ: 4800

2. Минимальные и максимальные параметры могут быть запрошены с помощью специальных текстовых обозначений
Пример: SYSTem:COMMunicate:SERial:BAUD ? MINimum Ответ: 1200
3. Цифровые значения выводятся без объединения. Физические величины соотносятся с базовыми элементами.
Пример: READ:TS:BITRate ? Ответ: 38E6 для 38 Мбит/с
4. Значения логических функций возвращаются как 0 (выключено) и 1 (включено)
Пример: SYSTem:KLOCK ? Ответ: 0
5. Конкретная информация, выступающая в качестве параметра, возвращается в краткой форме.
Пример: SYSTem:COMM:SERial:PARity ? Ответ: нет

5.5.5. Параметры

Большинство команд требует специальных параметров. Параметры должны быть отделены от заголовка с помощью пробела. В качестве параметров также могут быть заданы цифровые значения, логические функции, численные данные, текстовые строки или блоки данных. Тип параметра, требующегося для конкретной команды, также как разрешенный диапазон значений, описаны вместе с командами.

Численные значения могут быть введены в любом общепринятом виде, напр., с помощью знака, отделяющей запятой, или экспоненты. Если значения превышают разрешение устройства. Они будут округлены.

Цифровая часть числа может включать в себя до 255 знаков, экспонента должна быть в диапазоне от -32000 до 32000. Экспонента выражается как «E» или «e». Экспонента не может быть использована в одиночку.

Физические величины могут выражаться приставками. Допускаемые коды для приставок– G (Гига), MA (мега, также возможны MOHM и MHZ), K (кило), M (милли), U (микро) и N (нано). Если не отмечена конкретная приставка, будет использована основная.

Специальные численные значения

Параметры MINimum, MAXimum и DEFault рассматриваются как специальные численные значения.

После запроса будет обозначено численное значение.

Пример:

Команда установки соединения: SYST:COMM:SER:BAUD MAX
SYST:COMM:SER:BAUD ?

Запрос:

MIN/MAX – указания минимальных и максимальных значений

DEF – указывает заданное значение, сохраненное в стираемой программируемой памяти. Эти значения совпадают с базовыми настройками, вызываемыми командой *RST.

NAN – Not A Number, нет номера, представлена как 9.91E37. Посылается только как ответ устройства. Цифровое значение не распознается. NAN обычно возникает при делении на ноль, вычитании бесконечно большого значения и обозначении несуществующей информации.

Стр. 12

Логические функции представлены в двух состояниях. Состояние ON (действительное) отображается как текстовое значение ON или некое численное значение, отличное от нуля. Состояние OFF (недействительное) отображается как текстовое значение OFF или как численное значение 0. Запросы предоставляют варианты 0 и 1.

Пример: запрос.

Команда установки соединения:

SYSTem:KLOCK ON

SYSTem:KLOCK ?

Ответ: 1

Численные данные подчиняются синтаксису данных ключевых слов. Они также имеют как краткую, так и полную формы. Параметры должны быть отделены от заголовка пробелом. Запрос выдает численные данные в краткой форме.

Пример:

Команда установки соединения:

SYSTem:CLOCK:EXTernal

Запрос: SYSTem:CLOCK ?

Ответ: EXT

Текстовые строки. Символы должны быть всегда заключены в одинарные или двойные кавычки.

Пример:

CONFigure:RECORD:NAME "USER" или

CONFigure:RECORD:NAME 'USER'

Блоки данных

Формат блока данных подходит для передачи больших объемов информации. Команда с параметром формата блока данных имеет следующую структуру: HEADer #45168xxxxxxxx

Символ ASCII «#» указывает на начало блока информации. Следующие цифры обозначают номер идущих далее знаков, описывающих длину информационного блока. В приведенном выше примере четыре знака передают длину 5168 байт. Информация, занимающая 5168 байт, следует далее. В процессе передачи этих данных все терминирующие резисторы и прочие управляющие символы игнорируются, пока полностью не передадутся эти 5168 байт.

Если блок данных возвращается после запроса, он всегда будет передан уровнем выше (<DEFINITE LENGTH ARBITRARY RESPONSE DATA>).

Стр. 13

5.5.6. Обзор элементов синтаксиса

‘ : ’ Двоеточие разделяет ключевые слова команды. В программном сообщении двоеточие, следующее за точкой с запятой, отмечает высший иерархический уровень команды.

‘ ; ’ Точка с запятой разделяет две команды в программном сообщении.

‘ , ’ Запятая разделяет несколько параметров команды.

‘ ? ’ Знак вопроса формирует запрос.

‘ * ’ Звездочка обозначает общую команду.

- ‘ ”” ’ Кавычки указывают на конец и начало строки символов.
- ‘ # ’ Решетка указывает на начало блока информации.
- ‘ ’ Пробел (код ASCII от 0 до 9, от 11 до 32 в двоичной системе, пробел = 32) разделяет заголовок и параметры.

Стр. 14

5.6. Описание команд

5.6.1. Система счисления

В следующих разделах все команды, выполняемые DVRG, сведены в таблицы в соответствии с их режимом функционирования и командной системы. Все они описаны в деталях. Система счисления длиннее в стандарте SCPI.

В этом разделе все команды, выполняемые устройством, представлены в табличной форме.

5.6.1.1. Таблица команд

Команда: столбец команд предоставляет обзор команд и их иерархической структуры (см. подробнее).

Параметр: столбец параметров выводит заданные параметры и диапазон их значений.

Блок: столбец блоков показывает основные блоки физических величин.

Замечания: столбец замечаний указывает:

- имеет ли команда форму запроса
- существует ли команда только в форме вопроса
- реализуется ли команда исключительно в параметре конкретного инструмента

Отступы

Различные уровни командной иерархии SCPI показаны в таблице отступами вправо. Чем ниже уровень, тем больше будет отступ вправо. Следует отметить, что полные обозначения команд включают в себя также наивысшие уровни.

Верхний/нижний регистр

Знаки верхнего и нижнего регистра используются для того, чтобы различать полную и краткую формы ключевых слов команды в ее описании. Сам прибор к регистру не чувствителен.

Специальные символы

| Вертикальная черта в обозначениях параметров используется для разделения альтернативных опций и должна восприниматься как «или». Эффект команды различается в соответствии с заданным параметром.

Пример: выбор параметров команды

```
SYSTem:COMM:SER:PAR NONE | EVEN | ODD
```

[] Ключевые слова в квадратных скобках могут быть исключены из соединенных заголовков. В целях совместимости со стандартом SCPI инструмент должен быть в состоянии распознать полную длину команды. Параметры в квадратных скобках также могут быть вставлены или исключены из команды.

{ } Параметры в фигурных скобках могут быть включены в команду один раз, несколько раз или вообще проигнорированы.

< > Обозначение в угловых скобках – это суммарное описание символьных данных, сведенных в таблицу из-за их длины.

Стр. 15

5.6.2. Общие команды

Общие команды основаны на стандарте IEEE 488.2 (IEC 625.2). Конкретная команда дает одинаковый эффект в разных инструментах. Заголовки этих команд состоят из звездочки “*”, за которой следуют три буквы. Многие общие команды соотносятся с отчетом о состоянии системы, описанном в разделе 5.8 – Отчет о состоянии системы.

Таблица 5.6-1. Общие команды

Команда	Параметр/ответ	Блок	Замечания

*CLS			Нет запроса
*ESE	0...255		
*ESR?			Только запрос
*IDN?			Только запрос
*OPC	0 1		
*PSC			
*RST			
*SRE	0...255		
*STB?			Только запрос
*WAI			

*CLS – CLEAR STATUS

Четкий статус устанавливает байты состояния (STB), стандартный регистр состояния системы (ESR) и часть EVENT регистра QUESTIONABLE и OPERATION

на нуле. Команда не влияет на включенные и движущиеся части регистра. Она очищает только выходной буфер.

*ESE от 0 до 255

Статус состояния включения устанавливает определенное значение регистра статуса состояния. Запрос *ESE? выдает содержимое регистра статуса состояния в десятичной форме.

*ESR?

Стандартный запрос статуса состояния выдает содержимое регистра статуса состояния в десятичной форме (от 0 до 255) и затем устанавливает регистр на нуле.

*IDN?

Запрос идентификации для определения устройства.

Пример ответа: “Rohde&Swarz, DVRG, 2.0”

2.0 – версия прошивки

*OPC

Операция завершена устанавливает 0 бит в регистре статуса состояния, если все предшествующие команды были выполнены. Это значение может быть использовано для вызова запроса на обслуживание (см. раздел 5.8, система отчетов о состоянии системы).

Стр. 16 – дубль стр. 15. Следовательно, следующая стр. – 17-я.

Стр. 17

*OPC?

Завершение операции запроса помещает сообщение “1” в буфер вывода, как только выполняются все команды, предшествующие этой (см. раздел 5.8, система отчетов о состоянии системы).

*PSC 0|1

Включение очистки статуса определяет, сохранены ли компоненты регистра ENABLE, или они были очищены при выключении питания.

*PSC = 0 – заставляет статус регистра сохранить его компоненты. С надлежащей конфигурацией регистров состояния ESE и SRE запрос на обслуживание запускается при включении питания.

*PSC ≠ 0 – очищает регистр. Запрос *PSC? считывает содержимое включенной очистки статуса. Ответ может быть 0 или 1.

*RST

Сброс приводит устройство в режим работы по умолчанию. Настройки умолчания представлены вместе с описанием команд.

*SRE от 0 до 255

Запрос на обслуживание устанавливает определенное значение запроса на обслуживание регистра включения. Эта команда определяет состояние, при котором срабатывает запрос на обслуживание. Запрос *SRE? выводит содержимое запроса на обслуживание регистра в десятичной системе. Значение неиспользуемого бита 6 (бит MSS) всегда равно нулю.

*STB?

Считывание байта состояния выводит содержимое байта состояния в десятичной системе.

*WAI

Подождите, чтобы продолжить – разрешает выполнение команд только после того, как будут выполнены все предыдущие команды и установлены сигналы (см. раздел 5.8, система отчетов о состоянии системы и “*OPC”).

Стр. 18

5.6.3. Команды функции PLAY

Таблица 5.6-2. Команды функции PLAY

Команда	Параметр/ответ	Блок	Замечания
CONFigure	1 0 или ON OFF		Начинает воспроизведение. Текущий файл загружается в RAM для воспроизведения и
:GENerator	/		
:PLAY[?]	1 0		

			прекращения такового.
CONFigure :GENerator :PAUSE[?]	1 0 или ON OFF / 1 0		Останавливает воспроизведение (без сброса повторного воспроизведения) 1=пауза активна, т.е. удерживается
CONFigure :GENerator :SEEK[?]	BEGin END / BEG END INB		Возвращает воспроизведение к началу/концу. Возвращает к позиции BEG, END или INB (in between, между).
CONFigure :GENerator :SEEK :POSition[?]	ddd.ddd / ddd.ddd	с	Устанавливает воспроизведение в позиции n. n мин:0.000 n разрешение: 1/1000 с n максимальное: длина последовательности (см. READ:FILL:LENG)
CONFigure :RATE :TS[?]	ddd.dddddd / ddd.dddddd	Мбит/с	Устанавливает TS- скорость передачи данных для файлов GTS и TRP. Минимум: 0.625 Мбит/с. Максимум: 160 Мбит/с. Разрешение: 1бит/с

CONFigure :RATE :TS :MIN?	/ ddd.dddddd	Мбит/с	Запрашивает минимальную TS-скорость передачи данных для файлов GTS и TRP. Для GTS минимум – TS DATARATE, в противном случае REC DATARATE
CONFigure :PLENght[?]	188 204 208 / 188 204 208	Байты	Устанавливает величину пакета транспортного потока MPEG для файлов GTS и TRP.
CONFigure :JITTer :PCR[?]	1 0или ON OFF / 1 0		Устанавливает колебание PCR во включенном или выключенном положении для GTS-файлов
CONFigure :JITTer :PCR :WAVeform[?]	SINE SQUare TRIangle / SIN SQU TRI		Устанавливает волнообразное PCR-колебание для GTS-файлов
CONFigure :JITTer :PCR :FREQuency[?]	dddddd.ddd / dddddd.ddd	Гц	Устанавливает частоту PCR-колебаний для GTS-файлов. Минимум: 1МГц Максимум: 100 кГц
CONFigure	dddddd	us	Устанавливает

:JITter :PCR :AMPLitude[?]	/ dddd		амплитуду PCR-колебаний для GTS-файлов Минимум: 1us Максимум: 10000us
CONFigure :GENerator ASIFormat[?]	CONTInuous PACKet / CONT PACK		Выбирают ASI-формат
READ :ARRay :PROGram[?]	- / <prog.no> {, ...}		Считывает номера программ, включенных в транспортный поток (от 1 до 65535)
READ :ARRay :PROGram :PID[?]	<prog.no> / <pid> {, ...}		Запрашивает PID-номера (от 0 до 8191) всех программных элементов. Первая цифра – PMT PID, вторая - PCR PID.

Стр. 19

Команда	Параметр/ответ	Блок	Замечания
READ [:SCALar] :PROGram :NAME?	<prog.no> / <name>		Запрашивает наименование программы
READ [:SCALar]	<pid> /		Запрашивает тип элементарного потока.

:PID :TYPE?	V625 V525 AUD DATA VTXT SUBT		V625 – видео 625 строк развертки V525 – видео 525 строк развертки AUD – аудио DATA – информация VTXT – видеотекст SUBT - субтитры
READ [:SCALar] :TS :BITRate?	<prog.no> / От 0.0 до 99.999E6	Биты	Запрашивает скорость транспортного потока
MMEMory :MSIS[?]	drive name / drive name		Выбирает или запрашивает текущие имена дисков как символы (напр., C)
MMEMory :MSIS :NAME?	- / <data media name>		Указывает имя видеоданных конкретного диска, и мя устройства для CFS-драйверов
MMEMory :MCFS[?]	[<index number>] / <drive name>		Определяет букву диска CFS-устройства. Несколько таких устройств должны быть обозначены индексным номером от 0 до n.

			Индексный номер, равный 0, установлен по умолчанию и может быть опущен.
MMEMory :CDIRectory[?]	<path name> / <path name>		Переводит на установленное значение или отображает текущий каталог (напр., “\DVRG\TRP_FILE S\USER”)
MMEMory :MDIRectory[?]	<path name>		Указывает заданный путь к каталогу
MMEMory :RDIRectory[?]	<path name>		Удаляет выбранный каталог (только если он пуст)
MMEMory :CATalog?	[<path name>] / Список файлов или названий каталогов, напр., D, “Dir1”, “Dir2”, {...}, F, “File1”, “File2”, {...}		Идентифицирует содержимое данного каталога (каталога и имен файлов). Code. Char. D – предыдущего списка каталогов, Code Char F – предыдущего списка файлов. Если путь к файлу особенный, он

			будет просканирован.
MMEMory :CATalog :ALL?	<path name> / Список файлов или каталогов + информация: D, "Dir1", "yyyy,mm,dd,hh,ss,attr", "Dir2", {...,....,....}, F, "File1", "yyyy,mm,dd,hh,ss,size,attr", "File2", {...,....,....}		Показывает содержимое данного каталога; для каталогов – дату создания, информацию и атрибуты, к тому же для файлов – размер.
MMEMory :CATalog :FIRSt?	<path name>		Вызывает первый подкаталог или файл в данном каталоге

Стр. 20 – дубль стр. 19. Следовательно, следующая стр. – 21.

Команда	Параметр/ответ	Блок	Замечания
MMEMory :CATalog :NEXT?	<Путь к файлу, включающий в себя дополнительную маску D, F, A>		Опция D вызывает следующий подкаталог данного каталога. Опция F вызывает следующий файл данного каталога. Опция A вызывает следующий подкаталог или файл в данном каталоге.
MMEMory :CATalog			Прекращает выполнение данного процесса.

:CLOSE?			
MMEMory :CATalog :FREE?			Показывает свободный объем памяти в байтах
MMEMory :COPY	<имя исходного файла>, <имя файла назначения>		Копирует исходный файл в файл назначения
MMEMory :REName	<имя исходного файла>, <имя файла назначения>		Переименовывает исходный файл в файл назначения
MMEMory :NEW	<имя файла>		Создает новое имя файла с файлом назначения <имя файла>. Имя файла также может включать в себя путь к нему, который создается при соответствующем запросе
MMEMory :DELETE	<имя файла>		Удаляет выбранный файл
MMEMory :INFO?	[<имя файла>] / Имя файла, размер, дата создания, атрибуты		Указанная краткая информация по конкретному файлу. Без передачи значения: показывает информацию о данном файле
MMEMory :SElect[?]	[<имя файла>] /		Выбирает указанный файл как текущий и идентифицирует

	Информация заголовка, <...>		заголовок (количество передаваемых данных, зависящих от типа файла, напр., расширения названий)
MMEMory :SAVE	-		Копирует содержимое оперативной памяти (RECORD) в текущий файл
READ :FILE :NAME?	<имя файла>		Считывает имя выбранного файла
READ :FILE :DATE?	yyyy,mm,dd		Считывает дату создания выбранного файла
READ :FILE :VERSion?	dd.dd		Считывает версию данного GTS-файла
READ :FILE :STANdard?	625 525 NDEF		Считывает стандарт данного GTS-файла
READ :FILE :MODE?	GTS TRP 601		Показывает режим текущего файла

READ :FILE :LENGth?	dddd.ddd	s	Считывает последовательную длину файла (от 0 до 2E31)
-----------------------------------	----------	---	---

Стр. 22

Команда	Параметр/ответ	Блок	Замечания
READ :FILE :FRAMe :COUNT?	dddd		Считывает количество кадров в файлах ITU-R601
READ :GENerator :LOAD :POSition?	ddd		Считывает процесс загрузки, выдавая его как число между 0 и 100
READ :GENerator :GETHeader :POSition?	ddd		Считывает процесс загрузки нового заголовка файла, выдавая его как число между 0 и 100
READ :GENerator :CDIR?	<имена путей>		Считывает использующийся в данный момент путь

Стр. 23

CONFigure:GENErator:PLAY 0|1 или OFF|ON Значение *RST: 0

Команда начинает или останавливает воспроизведение

Пример: CONF/GEN/PLAY 1

CONFigure:GENErator/PAUSE 0|1 или OFF|ON Значение *RST: 0

Команда приостанавливает повторение воспроизведения (1/ON) или включает его снова (0/OFF)

Пример: CONF/GEN/PAUS 1

CONFigure:GENErator:SEEK BEGin|END Значение *RST: BEG

Команда устанавливает воспроизведение на начальную или конечную позиции

Пример: CONF:GEN:SEEK:BEG

CONFigure:GENErator:SEEK:POSition

Команда устанавливает начало воспроизведения в определенной точке потока (от 0 до максимальной длины последовательности) Значение *RST: 0.000

Пример: CONF:GEN:SEEK:POS 0.300s

CONFigure:RATE:TS от 0.625 до 160 Мбит/с Значение *RST: 20 Мбит/с

Команда устанавливает скорость передачи данных GTS и TRP

Пример: CONF:RATE:TS 30Mbit/s

CONFigure:RATE:TS:MIN?

Запрос считывает минимальную скорость передачи данных транспортных потоков GTS и TRP.

Пример: CONF:RATE:TS:MIN?

Ответ: 30Mbit/s

CONFigure:PLENgth 188|204|208 Значение *RST: 188

Команда устанавливает длину пакета транспортных потоков GTS и TRP

Пример: CONF:PLEN: 208

CONFigure:JITTer:PCR 0|1 или OFF|ON Значение *RST: 0

Команда включает или выключает колебания PCR

Пример: CONF:JITT:PCR ON

CONFigure:JITTer:PCR:WAVeform SINE|SQUare|TRIangle Значение *RST: SIN

Команда устанавливает волнообразные PCR-колебания

Пример: CONF:JITT:PCR SIN

CONFigure:JITTer:PCR:FREQuency 1 МГц до 100 кГц Значение *RST: 100 МГц

Команда устанавливает частоту PCR-колебаний

Пример: CONF:JITT:PCR:FREQ 1kHz

CONFigure:JITTer:PCR:AMPLitude от 1 до 10000µs Значение *RST: 1µs

Команда устанавливает амплитуду PCR-колебаний

Пример: CONF:JITT:PCR:AMPL 10 µs

CONFigure:GENerator:ASIFormat CONTinuous|PACKet Значение *RST: CONT

Команда устанавливает формат ASI

Пример: CONF:GEN:ASIF:CONT

READ:ARRay:PROGram?

Этот запрос считывает в таблице номеров программы, участвующие в транспортном потоке. (от 1 до 65535). Если ни одна программа не доступна, появляется значение NAN.

Пример: READ:ARR:PROG? Ответ: 1, 2, 3

Стр. 24

READ:ARRay:PROGram:PID? <имя программы>

Запрос считывает PID всех элементарных потоков, связанных с переданными номерами программ.

Формат: PMT PID программы

PCR PID программы

PID первого элементарного потока

PID второго элементарного потока

..

..

Если нет ни одной программы с таким номером, генерируется SCPI-ошибка 262, “нет совпадений”.

Пример: READ:ARR:PROG:PID? 2 Ответ: 200, 201, 64, 65, 66

READ[:SCALar]:PROG:NAME? <номер программы>

Этот запрос считывает имя программы, связанное с числом используемой программы. Если нет ни одной программы с этим номером, генерируется SCPI-ошибка 262, “нет совпадений”.

Пример: READ:PROG:NAME? 1 Ответ: ARD

READ[:SCALar]:PID:TYPE? <PID>

Запрос считывает элементарные потоки, связанные с выбранным PID.

Возвращаемое значение: 625 видео с 625-строчной разверткой

525 видео с 525-строчной разверткой

UD аудио

ATA данные

ТХТ видеотекст

UBT субтитры

Если нет элементарного потока для данного PID, генерируется SCPI-ошибка 262, “нет совпадений”.

Пример: READ:PID:TYPE? 64 Ответ: V625

READ[:SCALar]:TS:BITRate?

Запрос считывает скорость передачи данного транспортного потока.

Пример: READ:TS:BITR? Ответ: 30 000000

READ[:SCALar]:PROGram:BITRate? <номер программы>

Эта команда запрашивает скорость передачи данных конкретной программы. Если такой программы не имеется, то генерируется SCPI-ошибка 262, “нет совпадений”.

Пример: READ:PROG:BITR? 2 Ответ: 3 000000

Стр. 25

MMEMory:MSIS <имя диска>

Эта команда выбирает или запрашивает имя данного диска.

- Замечание: путь (за исключением принадлежащей диску части) обозначается по-старому и должен быть установлен в корневой папке с помощью MMEM:CDIR “\”

Пример: MMEM:MSIS “C:”

MMEMory:MSIS:NAME?

Запрос идентифицирует медиа-данные выбранного диска и имя устройства для CFS-устройств.

Пример: MMEM:MSIS:NAME? Ответ: CFS1

MMEMory:MCFS? <порядковый номер >

Запрос считывает букву, которой обозначается диск CFS-устройства. Несколько CFS-дисков отмечаются порядковыми номерами от 0 до N. Порядковый номер, равный нулю, может быть опущен.

Пример: MMEM:MCFS? Ответ: E

MMEMory:CDIRectory <путь к файлу>

Команда запускает текущий диск. Переход в корневой каталог представлен параметром “\”, в противном случае указанные строки принимаются за путь, связанный с тем же диском.

- Замечание: не должны упоминаться имена дисков.

Пример: MMEM:CDIR “\DVRG\TRP_FILES\USER” полный путь

MMEM:CDIR “USER1” краткий путь

MMEMory:MDIRectory <имя файла>

Команда создает новые указанные разделы. Имя файла связано с текущим путем, кроме случаев, когда оно начинается с символа “\”.

Пример: MMEM:MDIR “DVRG\TRP_FILES\USER”

MMEMory:RDIRectory <имя файла>

Команда удаляет выбранный каталог, если он пуст. Имя файла связано с текущим путем, кроме случаев, когда оно начинается с символа “\”.

Пример: MMEM:RDIR “USER1”

MMEMory:CATalog?

Этот запрос считывает содержимое выбранного каталога. Списку каталогов предшествует обозначение “D”, списку файлов – обозначение “F”. Файлы и данные каталога представлены заключенными в кавычки строками.

Пример: MMEM:CAT? Ответ: D, “Dir1”, “Dir2”, ... , F, “File1”, “File2, ...”

Стр. 26

MMEMory:CATalog:ALL?

Запрос считывает содержимое выбранного каталога: подкаталоги и файлы, даты создания, время, атрибуты и размер файлов.

Пример: MMEM:CAT:ALL? Ответ: D, “Dir1”, уууу,мм,дд,hh,мм,ss,attr, Dir2”, уууу...

“File1”, уууу,мм,дд,hh,мм,ss,size,attr, ”File2”, ...

- Замечание: attr – битовое поле атрибутов. Используются следующие обозначения:

NORMAL	BIT0=1
DIR	BIT1=2
FILE	BIT2=4
RDONLY	BIT3=8

HIDDEN BIT4=16

SYSTEM BIT5=32

ARCH BIT6=64

MMEMory:CATalog:FIRSt? <имена файлов>

Команда вызывает первый подкаталог или файл выбранного каталога.

Пример: MMEM:CAT:FIRSt? Ответ: D, "Dirname", уууу,мм,дд,хх,мм,сс,аттр

F, "Filename", уууу,мм,дд,хх,мм,сс,аттр

MMEMory:CATalog:NEXt? <имена файлов>

Команда вызывает следующий подкаталог или файл в данном каталоге.

Пример: MMEM:CAT:NEXt? Ответ: D, "Dirname", уууу,мм,дд,хх,мм,сс,аттр

F "Filename", уууу,мм,дд,хх,мм,сс,аттр

MMEMory:CATalog:CLoSe

Команда прерывает текущий процесс.

Пример: MMEM:CAT:CLoSe

MMEMory:CATalog:FRee?

Команда показывает количество свободной памяти в байтах.

Пример: MMEM:CAT:FRee? Ответ: 12390000

MMEMory:COpy <имя исходного файла> <имя файла назначения>

Команда копирует исходный файл в файл назначения. Если файл назначения отсутствует, генерируется SCSI-ошибка 256, «Имя файла не найдено»

Пример: MMEM:COpy "FILE1" "FILE2"

MMEMory:REName <имя исходного файла> <имя файла назначения>

Команда переименовывает исходный файл в файл назначения. Если файл назначения отсутствует, генерируется SCSI-ошибка 256, «Имя файла не найдено»

Пример: MMEM:REn "FILE1" "FILE2"

MMEMory:NEW <имя файла>

Команда воссоздает переданный файл. Имя файла связано с путем к нему, если не начинается с символа “\”.

Пример: MMEM:NEW “\DVRG\USER\TEST01.GTS”

MMEMory:DELeTe <имя файла>

Команда удаляет передаваемый файл. Имя файла связано с путем к нему, если не начинается с символа “\”. Если файл отсутствует, генерируется SCPI-ошибка 256, «Имя файла не найдено».

Пример: MMEM:DEL “TEST01.GTS”

Стр. 27

MMEMory:INFO? [<имя файла>]

Запрос считывает описание информации, связанной с данным именем файла: дату создания, время, размер, атрибуты. Если не передается ни одно имя файла, используется имя текущего файла из меню GENERATOR. Если запрашиваемый файл отсутствует, генерируется SCPI-ошибка 256, «Имя файла не найдено».

Пример: MMEM:INFO? “TEST01.GTS” Ответ: “File name”,
уууу,mm,dd,hh,mm,ss,size,attr

MMEMory:SELeCt <имя файла>

Команда выбирает имя передающегося файла как текущий файл и загружает основную информацию о нем. Эта информация выводится в MEMM:SEL?

Информация заголовков GTS-файлов: день, месяц, год, версия, стандарт, длина последовательности, минимальная скорость передачи данных, длина пакета, колебания.

Информация заголовков TRP-файлов: день, месяц, год, длина последовательности, скорость передачи данных, скорость записи данных, длина пакета.

Информация заголовков 601-файлов: день, месяц, год, стандарт, номер кадра, длина последовательности.

Пример:

MMEM:SEL "TEST01.GTS"

READ:FILE:NAME?

Запрос выдает имя данного файла как строку в меню GENERATOR.

Пример: READ:FILE:NAME? Ответ: "TEST01.GTS"

READ:FILE:DATE?

Запрос считывает дату создания выбранного файла в формате гггг,мм,дд

Пример: READ:FILE:DATE? Ответ: 2000,01,20

READ:FILE:VERSion?

Запрос выдает номер версии GTS-файла в виде строки символов, NDEF (не определено) выдается для остальных файлов.

Пример: READ:FILE:VERS? Ответ: "01.00"

READ:FILE:STANdard?

Запрос выдает видеостандарт текущего GTS-файла в виде строки символов, NDEF (не определено) выдается для остальных файлов.

Пример: READ:FILE:STAN? Ответ: "625" – стандарт BG/PAL, частота развертки 625 строк

"525" – стандарт M/NTSC, частота развертки 525 строк

READ:FILE:MODE?

Запрос выдает режим (GTS, TRP, SDI) выбранного файла в виде строки символов. NDEF (не определено) выдается для остальных файлов.

Пример: READ:FILE:MODE? Ответ: "GTS"

READ:FILE:LENGth?

Запрос считывает длину последовательности текущего файла в секундах. NDEF (не определено) выдается для остальных файлов.

Пример: READ:FILE:LENG? Ответ: 9.760

READ:FILE:FRAMe:COUNt?

Запрос считывает количество кадров для ITU-R 601 файлов. NDEF (не определено) выдается для остальных файлов.

Пример: READ:FILE:FRAM:COUN? Ответ: 2000

READ:GENerator:LOAD:POSition?

Запрос считывает процесс загрузки в диапазоне от 0 до 100.

Пример: READ:GEN:LOAD:POS? Ответ: 50

READ:GENerator:GETHeader:POSition?

Запрос считывает процесс создания нового заголовка файла, выдавая результат в диапазоне от 0 до 100.

Пример: READ:GEN:GETH:POS? Ответ: 50

READ:GENerator:CDIR?

Запрос считывает текущий путь каталога

Пример: READ:GEN:CDIR? Ответ: F:\TRP\TEST

Стр. 28

5.6.4. Команды функции RECORD

Таблица 5.6-3

Команда	Параметр/ответ	Блок	Замечания
CONFigure :RECORD :INPut[?]	AREar AFRont SREar SDI ARE AFR SRE SDI		Выбирает вход записи - ASI Rear - ASI Front - SPI Rear - SDI
CONFigure :RECORD	ddd /	Мбайт	Устанавливает размер записи

:SIZE[?]	ddd		
CONFigure :RECORD :NAME[?]	<имя файла> / <имя файла>		Отмечает имя файла, в котором должна быть сохранена запись. Если последняя буква имени – а, то имя увеличивается для каждой записи: USER01, USER02, ... Возвращенное значение: генерируйте имя при помощи символов подстановки
CONFigure :RECORD :MODE[?]	CONTInuous SINGle / CONT/SING		Выбирает режим записи
CONFigure :RECORD :STOP :EXtern[?]	DISable SINGle MULtiple / DIS SING MULT		Отключает внешний сигнал остановки, включает его для записи или начинает новую запись после каждой остановки
CONFigure :RECORD :STOP :DELay[?]	ddd / ddd	PCT	Устанавливает задержку воспроизведения
CONFigure :RECORD	1 0 или ON OFF /		Начинает или останавливает запись

:START[?]	1 0		
CONFigure :RECORD :ABORT			Прерывает запись без учета задержки. Выводимое значение отсутствует.
READ :RECORD :TIME :POSITION?	ddddddd	ms	Позиция в процессе записи в ms
READ :RECORD :TIME :STOP?	dddddd	s	После команды записи: «Начало» эта команда позволяет считывать задержки вплоть до конца записи (в секундах)
READ :RECORD :SAVE :POSITION?	ddd		После команды «Сохранить» позволяет считывать процесс сохранения. Значения вывода от 0 до 100
READ :RECORD :NAME?	<имя файла>		Считывает имя файла, присвоенное ему во время сохранения, напр., шаблон. Заполняется числовыми значениями
READ :RECORD :CDIR?	<путь к файлу>		Считывает путь к текущей записи

Стр. 29

CONFigure:RECOrd:INPut AREar|AFRont|SREar Значение *RST: ARE

Команда выбирает или запрашивает вход записи.

ARE ASI Rear

AFR ASI Front

SRE SPI Rear

Пример: CONF:REC:INP AFR

CONFigure:RECOrd:SIZE от 0 до максимума памяти в Мбайтах

Значение *RST: 60 Мбайт

Команда устанавливает или запрашивает размер записи в Мбайтах.

Пример: CONF:REC:SIZE 20 MB

CONFigure:RECOrd:NAME <имя файла> Значение *RST: USER??

Команда устанавливает имя файла для сохранения записи. Имя файла может быть дополнено знаком вопроса в качестве шаблона. Номера шаблонов считываются в процессе сохранения. Фактическое имя файла может быть определено с помощью READ:REC:NAME?

Пример: CONF:REC:NAME "USER1"

CONFigure:RECOrd:MODE CONTInuous|SINGle Значение *RST: CONT

Команда запрашивает или устанавливает режим записи.

Пример: CONF:REC:MODE SING

CONFigure:RECOrd:STOP:EXTern DISable|SINGle:MULTiple Значение *RST: DIS

Отключает внешний сигнал остановки (DISable), включает его для записи (SINGle) или начинает новую запись после каждой остановки (MULTiple).

Пример: CONF:REC:STOP:EXT SING

CONFigure:RECOrd:STOP:DELaY от 0 до 100% Значение *RST: 10%

Команда запрашивает или устанавливает задержку между внешней командой остановки и фактической остановкой записи.

Пример: CONF:REC:STOP:DEL 20%

CONFigure:RECOrd:STARt 1|0 или ON|OFF Значение *RST: 0

Команда начинает или останавливает воспроизведение, а также запрашивает его состояние.

Пример: CONF:REC:STAR 1

CONFigure:RECOrd:ABORt

Команда прерывает запись без учета задержки.

Пример: CONF:REC:ABOR

READ:RECOrd:TIME:POSition?

Запрос считывает процесс записи.

Пример: READ:REC:TIME:POS? Ответ: 2000

READ:RECOrd:TIME:STOP?

После команды CONFigure:RECOrd:STARt OFF этот запрос позволяет считывать задержку вплоть до конца записи.

Пример: READ:REC:TIME:STOP? Ответ: 1000

READ:RECOrd:SAVE:POSition?

Запрос считывает процесс сохранения (после команды MMEMOry:SAVE).

Пример: READ:REC:SAVE:POS? Ответ: 20

READ:RECOrd:NAME?

Запрос считывает имя созданного файла. Шаблоны (“?”), если таковые имеются, отмечены числовыми значениями. Названия с шаблонами устанавливаются или считываются с помощью запроса CONF:REC:NAME[?]

Пример: READ:REC:NAME? Ответ: “USER02.TRP”

READ:RECOrd:CDIR?

Команда считывает путь к текущей записи.

Пример: READ:REC:CDIR? Ответ:“F:\REC\TEST”

Стр. 30

5.6.5. Команды функции SETUP

Таблица 5.6-4

Команда	Параметр/ответ	Блок	Заметки
SYSTem :DATE[?]	yyyy,mm,dd / yyyy,mm,dd		Устанавливает дату.
SYSTem :TIME[?]	hh,mm,ss / hh,mm,ss		Устанавливает время.
SYSTem :CLOCK[?]	INTernal EXTernal / INT EXT		Устанавливает внутренний сигнал синхронизации времени.
SYSTem :COMMunicate :SERial [:RECeive] :BAUD[?]	4800 9600 19200 38400 57800 115200 / 4800 9600 19200 38400 57800 115200		Устанавливает скорость передачи данных последовательного интерфейса
SYSTem :COMMunicate :SERial [:RECeive] :BITS[?]	7 8 / 7 8		Количество битов данных

SYSTEM :COMMunicate :SERial [:RECeive] :SBITs[?]	1 1.5 2 / 1 1.5 2	Количество стоп- битов
SYSTEM :COMMunicate :SERial [:RECeive] :PARity[?]	NONE EVEN ODD ZERO MARK / NONE EVEN ODD ZERO MARK	Контроль четности
SYSTEM :COMMunicate :SERial [:RECeive] :PACE[?]	NONE XONE ACK / NONE XONE ACK	Протокол: NO XON/XOFF RTS/CTS
SYSTEM :COMMunicate :INET :ADDRess[?]	<addr0>,<addr1>,<addr2>,<addr3 > / <addr0>,<addr1>,<addr2>,<addr3 > VAL REB	IP-адреса для дистанционного управления Ethernet для удаленного клиента. VAL или REBoot добавляется для запроса, зависящего от того, было ли активировано новое соединение или требуется

			перезагрузка.
SYSTEM :COMMunicate :INET :SMASk[?]	<addr0>,<addr1>,<addr2>,<addr3 > / <addr0>,<addr1>,<addr2>,<addr3 > VAL REB		Маска дистанционного управления Ethernet для удаленного клиента.
SYSTEM :COMMunicate :INET :GATeway[?]	<addr0>,<addr1>,<addr2>,<addr3 > / <addr0>,<addr1>,<addr2>,<addr3 > VAL REB		Адрес шлюза для дистанционного управления Ethernet для удаленного клиента.
SYSTEM :COMMunicate :INET :MACaddress[?]	<строка символов>		Запрашивает MAC-адрес сетевой карты

Стр. 31

SYSTEM :COMMunicate :INET :PORT[?]	<порт> / <порт>		Порт для дистанционного управления Ethernet для удаленного клиента.
---	-----------------------	--	--

SYSTEM :COMMunicate :INET :MODE[?]	<Режим чтения> / <Режим чтения>	Устанавливает или считывает номер использованного протокола IEEE/1174.1 или 2
SYSTEM :KLOCK[?]	1 0 или ON OFF / 1 0	Включает или выключает клавиатуру передней панели
SYSTEM :ERRor?	<строка символов>	Запрашивает очереди ошибок
SYSTEM :VERSion?	<строка символов>	Запрашивает SCPI-версию
SYSTEM :VERSion :ABOut?	<строка символов>	Запрашивает версию устройства, модель DVGRб версию SW, версию BIOS память устройства
SYSTEM :VERSion :DIAGnose?	<строка символов>	Запрашивает аппаратную диагностику из автозагрузки
READ :REMote :STATe?	0 1	Выдает 0: DVRG находится в режиме непосредственного

			<p>управления</p> <p>Выдает 1: DVRG находится в режиме дистанционного управления</p>
--	--	--	--

Стр. 32

SYSTem:DATE <yyyy,mm,dd>

Команда запрашивает конкретную дату или устанавливает ее.

Пример: SYST:DATE 2000, 01, 20

SYSTem:TIME <hh,mm,ss>

Команда запрашивает конкретное время или устанавливает его.

Пример: SYST:TIME 15, 02, 00

SYSTem:CLOCK INTernal|EXTernal Значение *RST:INT

Команда устанавливает режим внутренних или внешних часов.

Пример: SYST:CLOC EXT

SYSTem:COMMunicate:SERial[:RECEive]:BAUD

4800|9600|19200|38400|57800|115200

Значение *RST: 9600

Команда устанавливает или запрашивает скорость передачи данных для дистанционного управления через RS232. Если передается другое значение, то генерируется SCPI-ошибка 224, «Недопустимое значение параметра».

Пример: SYST:COMM:SER:BAUD 19200

SYSTem:COMMunicate:SERial[:RECEive]:BITS 7|8 Значение *RST: 8

Команда запрашивает или устанавливает число бит данных для дистанционного управления через RS232. Если передается другое значение, то генерируется SCPI-ошибка 224, «Недопустимое значение параметра».

Пример: SYST:COMM:SER:BITS 7

SYSTem:COMMunicate:SERial[:RECeive]:SBITs 1|1.5|2 Значение *RST: 1

Команда устанавливает или запрашивает стоп-биты данных для дистанционного управления через RS232. Если передается другое значение, то генерируется SCPI-ошибка 224, «Недопустимое значение параметра».

Пример: SYST:COMM:SER:SBIT 2

SYSTem:COMMunicate:SERial[:RECeive]:PARity
NONE|EVEN|ODD|ZERO|MARK

Значение *RST: EVEN

Команда устанавливает или запрашивает четность для дистанционного управления через RS232. Если передается другое значение, то генерируется SCPI-ошибка 224, «Недопустимое значение параметра».

Пример: SYST:COMM:SER:PAR ODD

SYSTem:COMMunicate:SERial[:RECeive]:PACE NONE|XON|ACK

Значение *RST: ACK

Команда устанавливает или запрашивает протокол взаимной идентификации для дистанционного управления через RS232.

Протокол: NONE – нет протокола

XON – XON|XOFF

ACK – RTS|CTS

Пример: SYST:COMM:SER:PACE NONE

SYSTem:COMMunicate:INET:ADDRes <addr1> <addr2> <addr3> <addr4>

Команда устанавливает или запрашивает IP-адреса для дистанционного управления Ethernet для удаленного клиента. VAL или REBoot добавляется для запроса, зависящего от того, было ли активировано новое соединение или требуется перезагрузка.

Пример: SYST:COMM:INET:ADDR 127, 100, 80, 10

SYSTem:COMMunicate:INET:SMASk <addr1> <addr2> <addr3> <addr4>

Команда устанавливает или запрашивает маску подсети для сетевого протокола интерфейса.

Пример: SYST:COMM:INET:SMAS 255,255,0,0

SYSTem:COMMunicate:INET:GATeway <addr1> <addr2> <addr3> <addr4>

Команда устанавливает или запрашивает шлюз сетевого протокола интерфейса.

Пример: SYST:COMM:INET:GAT 192, 168, 1, 0

Стр. 33

SYSTem:COMMunicate:INET:MACaddress?

Запрашивает аппаратный MAC-адрес сетевой карты.

Пример: SYST:COMM:INET:MAC? Ответ: AABBCCDDEEFF...

SYSTem:COMMunicate:INET:PORT от 0 до 9999

Команда устанавливает или запрашивает номер порта для сетевого протокола интерфейса.

Пример: SYST:COMM:INET:PORT 4002

SYSTem:COMMunicate:INET:MODE 1|2

Команда устанавливает или запрашивает номер использованного протокола IEEE/1174.1 или 2.

Пример: SYST:COMM:INET:MODE 1

SYSTem:CLOCK 1|0 или ON|OFF

Команда блокирует клавиатуру (KLOCK 1) или разблокирует ее (KLOCK 0). Запрос определяет статус состояния клавиатуры.

Пример: SYST:KLOC 1

SYSTem:ERRor?

Запрашивает последнюю отметку в очереди ошибок. Положительные номера ошибок обозначают зависящие от устройства ошибки, отрицательные номера ошибок – определяемые SCPI сообщения об ошибках (см. раздел 5.10, Сообщения об ошибках интерфейса дистанционного управления).

Пример: SYST:ERRor? Ответ: -221, “разворачивается конфликт”.

SYSTem:VERSion?

Запрос выдает поддерживаемую устройством SCPI-версию.

Пример: SYST:VERS? Ответ: 1995.0

SYSTem:VERSion:ABOut?

Запрашивает версию устройства. Строка ответа содержит версию DVRG, версию прошивки, информацию о BIOS и памяти.

Пример: SYST:ABO? Ответ: “VAR:02, 01.00, 30.11.99, 128”

SYSTem:VERSion:DIAGnose?

Запрашивает аппаратную диагностику из автозагрузки.

Пример: SYST:VERS:DIAG? Ответ: “STARTUP OK”

READ:REMOte:STATe?

Запрашивает, находится ли DVRG находится в режиме непосредственного управления (0) или в режиме дистанционного управления (1).

Пример: READ:REM:STAT? Ответ: 1

Стр. 34

5.6.6. Команды SCPI-регистра

Команды для контролирования SCPI-зависимых и устройство-ориентированных регистров статуса STATus. Функции отдельных регистров описаны в разделе 5.8, Отчеты о состоянии системы.

Таблица 5.6-5. Команды SCPI-регистра.

Команда	Параметр/ответ	Блок	Замечания

STATUS PRESet			Нет запроса
STATUS :QUEue [:NEXT]?			Запрос об ошибке или событии
:OPERation [:EVENT]? :CONDition? :ENABle :PTRansition :NTRansition	От 0 до 32767 От 0 до 32767 От 0 до 32767		Только запрос Только запрос
:QUEStionable [:EVENT]? :CONDiton? :ENABle :PTRansition :NTRansition	От 0 до 32767 От 0 до 32767 От 0 до 32767		Только запрос Только запрос
:QUEStionable :INPut [:EVENT]? :CONDition? :ENABle :PTRansition	От 0 до 32767 От 0 до 32767		Только запрос Только запрос

:NTRansition	От 0 до 32767		
:QUEStionable			Только запрос
:REFinput			
[:EVENT]?			
:CONDition?			Только запрос
:ENABle	От 0 до 32767		
:PTRansition	От 0 до 32767		
:NTRansition	От 0 до 32767		

Стр. 35

STATus:PRESet

Команды PRESet сбрасывают регистры ENABle и переходные регистры статуса SCPI к конкретному значению. Это относится к регистрам STATus:OPERation и STATus:QUEStionable. Все регистры PTRansition устанавливаются на значении 32767, а NTRansition регистры – на нуле, т.е. все переходы от 0 до 1 сигнализируют о регистре CONDition. Категория ENABle регистра SCPI устанавливается на нуле, т.е. все события тех регистров не передаются.

Пример: STAT:PRES

STATus:QUEue:[:NEXT]?

Этот запрос выдает следующее значение из очереди ошибок или событий и удаляет его из этого списка. Положительные номера ошибок обозначают зависящие от устройства ошибки, отрицательные номера ошибок – определяемые SCPI сообщения об ошибках. Если список пуст, DVRG выдает значение 0, «нет ошибок». Если очередь слишком заполнена, DVRG выдает результат -350, «слишком много ошибок».

Пример: STAT:QUE? Ответ: 0, “No Error”

STATus:OPERation

Это узловое событие обеспечивает команды, контролирующие регистр SCPI-STATus:OPERation.

STATus:OPERation[:EVENT]?

Запрашивает содержимое регистра EVENT системы регистров STATus:OPERation. Прочтение регистра EVENT очищает их.

Пример: STAT:OPER? Ответ: 512

STATus:OPERation:CONDition?

Этот запрос выдает содержимое регистра CONDition системы регистров STATus:OPERation. Считывание регистра CONDition не очищает их.

Пример: STAT:OPER:COND? Ответ: 0

STATus:OPERation:ENABle от 0 до 32767

Устанавливает биты в регистре ENABle системы регистров STATus:OPERation. Регистр выбирает и активирует отдельные биты регистра EVENT для итоговых битов байтов состояния. Значение PRESET равно нулю.

Пример: STAT:OPER:ENAB 32

STATus:OPERation:PTRansition от 0 до 32767

Устанавливает отрицательные фильтры перехода битов CONDition системы регистров STATus:OPERation. Если бит PTRansition равен 1, то соответствующий бит в регистре EVENT устанавливается при передаче бита CONDition в диапазоне от 1 до 0. Значение PRESET равно 32767.

STATus:OPERation:NTRansition от 0 до 32767

Устанавливает отрицательные фильтры перехода битов CONDition системы регистров STATus:OPERation. Если бит NTRansition равен 1, то соответствующий бит в регистре EVENT устанавливается при передаче бита CONDition в диапазоне от 1 до 0. Значение PRESET равно 0.

Пример: STAT:OPER:NTR 0

STATus:QUESTionable

Это узловое событие обеспечивает команды, контролирующие регистр SCPI-STATus:QUEStionable.

Стр. 36

STATus:QUEStionable[:EVENT]?

Запрашивает содержимое регистра EVENT системы регистров STATus:QUEStionable. Считывание регистра EVENT очищает его.

Пример: STAT:QUES? Ответ: 0

STATus:QUEStionable:CONDition?

Запрашивает содержимое регистра CONDition системы регистров STATus:QUEStionable. Считывание регистра CONDition не очищает его.

Пример: STAT:QUES:COND? Ответ: 0

STATus:QUEStionable:ENABle от 0 до 32767

Устанавливает биты в регистре ENABle системы регистров STATus:QUEStionable. Этот регистр выбирает и активирует отдельные биты регистра EVENT для итоговых битов байтов состояния. Значение PRESET равно 32767.

Пример: STAT:QUEST:ENAB 128

STATus:QUEStionable:PTRansition от 0 до 32767

Устанавливает отрицательные фильтры перехода битов CONDition системы регистров STATus:QUEStionable. Если бит PTRansition равен 1, то соответствующий бит в регистре EVENT устанавливается при передаче бита CONDition в диапазоне от 1 до 0. Значение PRESET равно 32767.

Пример: STAT:QUEST:PTR 128

STATus:QUEStionable:NTRansition от 0 до 32767

Устанавливает отрицательные фильтры перехода битов CONDition системы регистров STATus:QUEStionable. Если бит NTRansition равен 1, то соответствующий бит в регистре EVENT устанавливается при передаче бита CONDition в диапазоне от 1 до 0. Значение PRESET равно 0.

Пример: STAT:QUES:NTR 0

STATus:QUEStionable:INPut

Это узловое событие обеспечивает команды для администрирования зависящего от устройства STATus: регистр QUEStionable:INPut. Это регистр идентифицирует сомнительные устройства через измерительный вход.

STATus:QUEStionable:INPut[:EVENT]?

Запрашивает содержимое регистра EVENT системы регистров STATus:QUEStionable:INPut. Считывание регистра EVENT очищает его.

Пример: STAT:QUES:INP? Ответ: 0

STATus:QUEStionable:INPut:CONDition?

Запрашивает содержимое регистра CONDitional системы регистров STATus:QUEStionable:INPut. Считывание регистра CONDition не очищает его.

Пример: STAT:QUES:INP:COND? Ответ: 0

STATus:QUEStionable:INPut:ENABle от 0 до 32767

Устанавливает биты в регистре ENABle системы регистров STATus:QUEStionable:INPut. Регистр выбирает и активирует отдельные биты регистра EVENT для итоговых битов байтов состояния регистра STATus:QUEStionable. Значение PRESET равно 32767.

Пример: STAT:QUES:INP:ENAB 128

Стр. 37

STATus:QUEStionable:INPut:PTRansition от 0 до 32767

Устанавливает отрицательные фильтры перехода битов CONDition системы регистров STATus:QUEStionable:INPut. Если бит PTRansition равен 1, то соответствующий бит в регистре EVENT устанавливается при передаче бита CONDition в диапазоне от 0 до 1. Значение PRESET равно 32767.

Пример: STAT:QUES:INP:PTR 128

STATus:QUEStionable:INPut:NTRansition от 0 до 32767

Устанавливает отрицательные фильтры перехода битов CONDition системы регистров STATus:QUEStionable:INPut. Если бит NTRansition равен 1, то

соответствующий бит в регистре EVENT устанавливается при передаче бита CONDITION в диапазоне от 1 до 0. Значение PRESET равно нулю.

Пример: STAT:QUES:INP:NTR 0

STATUS:QUESTIONABLE:REFinput

Это узловое событие обеспечивает команды, необходимые для администрирования зависящего от устройства STATUS: регистр QUESTIONABLE:REFinput. Этот регистр отмечает спорные состояния устройства при его включении.

STATUS:QUESTIONABLE:REFinput[:EVENT]?

Запрашивает содержимое регистра EVENT системы регистров STATUS:QUESTIONABLE:REFinput. Считывание регистра EVENT очищает его.

Пример: STAT:QUES:REF? Ответ: 0

STATUS:QUESTIONABLE:REFinput:CONDITION?

Запрашивает содержимое регистра CONDITION системы регистров STATUS:QUESTIONABLE:REFinput. Считывание регистра CONDITION не очищает его.

Пример: STAT:QUES:REF:COND? Ответ: 0

STATUS:QUESTIONABLE:REFinput:ENABLE

Устанавливает биты в регистре ENABLE системы регистров STATUS:QUESTIONABLE:REFinput. Регистр выбирает и активирует отдельные биты регистра EVENT для итоговых битов байтов состояния регистра STATUS:QUESTIONABLE. Значение PRESET равно 32767.

Пример: STAT:QUES:REF:ENAB 128

STATUS:QUESTIONABLE:REFinput:PTRansition

Устанавливает отрицательные фильтры перехода битов CONDITION системы регистров STATUS:QUESTIONABLE:REFinput. Если бит PTRansition равен 1, то соответствующий бит в регистре EVENT устанавливается при передаче бита CONDITION в диапазоне от 0 до 1. Значение PRESET равно 32767.

Пример: STAT:QUES:REF:PTR 128

STATus:QUEStionable:REFinput:NTRansition

Устанавливает отрицательные фильтры перехода битов CONDition системы регистров STATus:QUEStionable:REFinput. Если бит NTRansition равен 1, то соответствующий бит в регистре EVENT устанавливается при передаче бита CONDition в диапазоне от 1 до 0. Значение PRESET равно 0.

Пример: STAT:QUES:REF:NTR 0

Стр. 38

5.7. Инструментальная модель и обработка команд

Эта инструментальная модель, показанная на схеме 5-3, была подготовлена для обзора выполнения команд удаленного управления. Отдельные компоненты выполняются одновременно и независимо друг от друга. Они связаны друг с другом смыслом передаваемых сообщений.

5.7.1. Блок входа

Блок входа получает команды в виде символов из интерфейса дистанционного управления и сохраняет их во буфере входа. Блок входа посылает сообщение идентификатору команд, как только заполняется буфер, или как только принимается ограничитель (см. приложение А).

5.7.2. Идентификация команд

Идентификатор команд анализирует информацию, полученную из блока входа. Данные обрабатываются той же последовательности, в которой они были получены. Синтаксические ошибки в команде распознаются и передаются системе отчетов о состоянии устройства. Соблюдая синтаксис теста, диапазон информационных значений проверяется, и настройки устанавливаются соответствующим образом. Только когда команда уже полностью выполнена, следующая команда начинает обрабатываться идентификатором команд.

Стр. 39

5.7.3. Набор данных и аппаратное устройство

Термин «аппаратное устройство» отсылает нас к той части прибора, которая представляет генерирующую или записывающую функции. Набор данных

содержит все параметры, необходимые для настройки аппаратного устройства.

Команды установки вызывают изменения в наборе данных. Прежде чем информация поступает в набор данных, она проверяется на совместимость с другой информацией и аппаратным устройством. Если оказывается, что какая-то настройка невозможна, сообщение об ошибке выполнения посылается в систему отчетов о состоянии системы, и эта настройка игнорируется. После успешного завершения проверки настройка будет немедленно выполнена.

Запросы заставляют управление набором данных посылать желаемую информацию в блок выхода.

5.7.4. Блок выхода

Блок вывода данных собирает информацию, запрашиваемую контроллером, из управления набором данных. Он предоставляет информацию в строке, соответствующей правилам SCPI, и делает ее доступной в буфере вывода.

5.7.5. Последовательность команд и их синхронизация

Все команды немедленно исполняются. Не существует параллельной обработки команд. Тем не менее, пользователь может определять последовательность исполнения команд. Команда *WAI не оказывает влияния на последовательность исполнения команд.

Если длинное программное сообщение прерывается *OPC или *OPC?, о конце обработки команды сообщается контроллеру соответствующим программным сообщением.

Таблица 5.7-1. Синхронизация с *OPC, *OPC?

Команда	Действие после аппаратной настройки	Программирование контроллера
*OPC	Установка настройки бита завершения работы	Устанавливает бит 0 в ESE Устанавливает бит 5 в SRE Ожидает запроса на обслуживание (SRQ)

*ОРС?	Записывает единицу в буфер вывода	Адресует устройство в качестве говорящего
-------	-----------------------------------	---

Стр. 40

5.8. Отчеты о состоянии системы

Система отчетов о состоянии хранит всю информацию о текущем рабочем состоянии прибора и ошибках. Информация хранится в регистре статуса и журнале ошибок. Содержимое регистра статуса и журнала ошибок может быть запрошено через дистанционное управление.

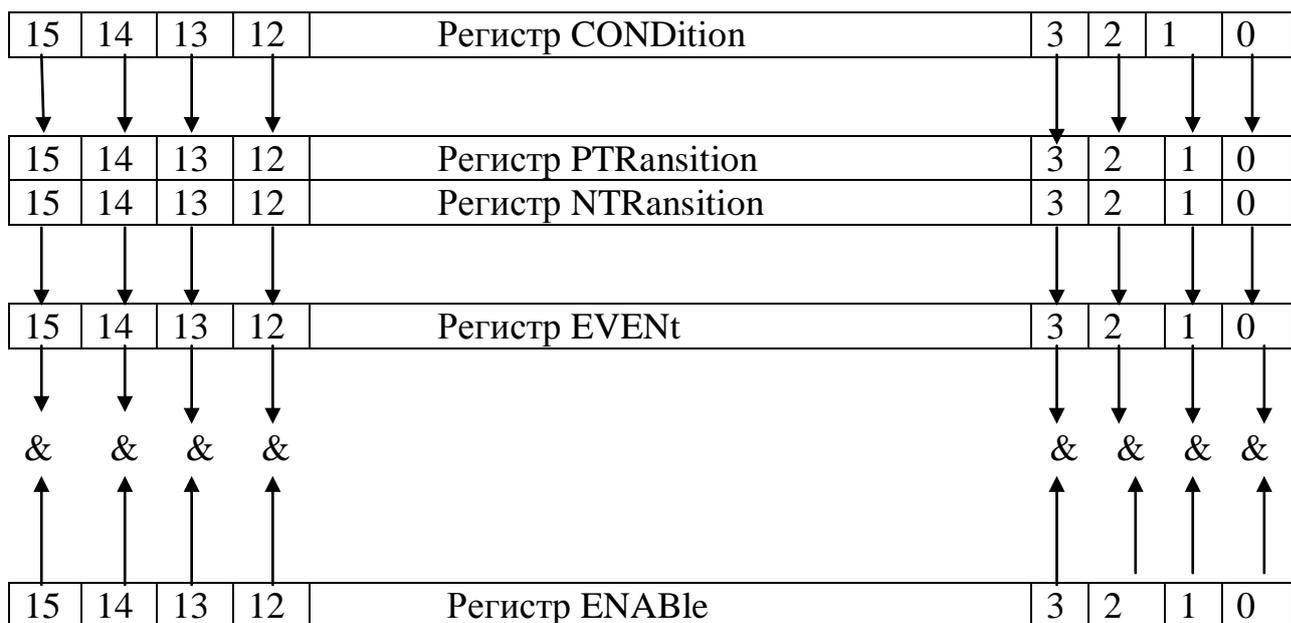
Вся информация иерархически структурирована. Высший уровень формируется регистром байтов состояния (STB), определяющимся стандартом IEEE 488.2 и соответствующим регистром включения запроса на обслуживание (SRE). STB получает информацию из регистра статуса стандартных событий (ESR), тоже соответствующего стандарту IEEE 488.2, а также из регистра статуса включения стандартных событий (ESE). Также информация приходит из зависящих от SCPI регистров STATus:OPERation и STATus:QUEStionable, содержащих более подробную информацию об устройстве.

Буфер вывода содержит сообщения, переданные устройством контроллеру. Они не являются частью отчетов о состоянии системы, но определяют значение бита MAV в STB.

5.8.1. Структура регистра SCPI

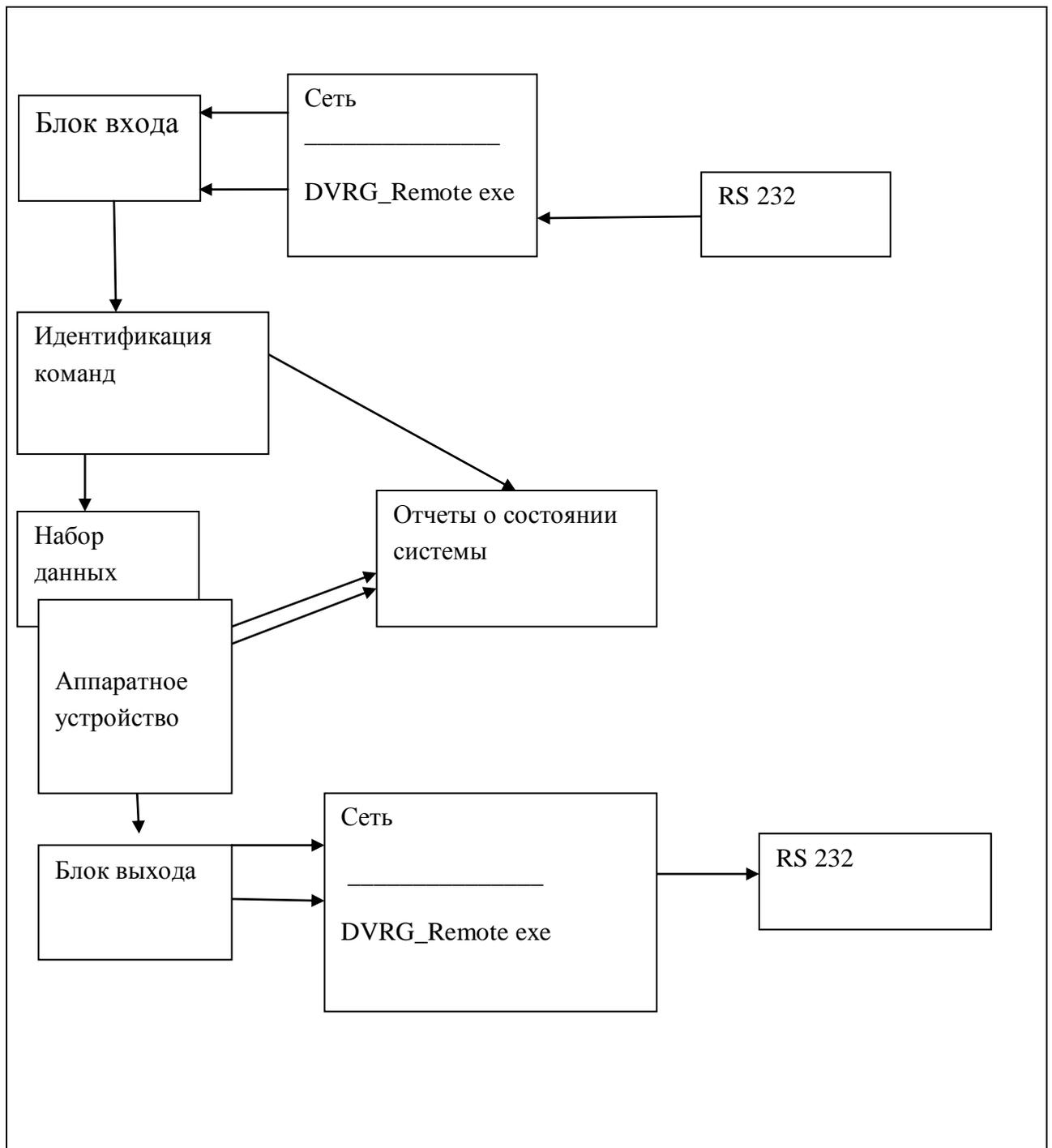
Каждый SCPI-регрстр состоит из пяти 16-битных регистров, поддерживающих различные функции. Отдельные биты не зависят друг от друга, т.е. каждому состоянию оборудования присваивается разрядное число, одинаковое для всех регистров. К примеру, пятый бит регистра STATus:OPERation присваивается «Ожиданию действия» во всех пяти регистрах. Пятнадцатый бит (наиболее значимый) устанавливается на нуле во всех регистрах статуса SCPI. Таким образом, содержимое регистров может обрабатываться контроллером программ как положительное число.

Схема 5-4



Где стрелка обозначает переход к регистру более высокого уровня, & - логическое «и».

Схема 5-3. Инструментальная модель дистанционного управления



Стр. 41

Регистр CONDition напрямую заполняется аппаратными средствами или суммарными битами нижеследующего регистра. Его содержимое отражает текущий статус устройства. Этот регистр может быть только прочтен, но не написан или очищен.

Регистр PTRansition (Positive Transition Register) работает как фильтр передачи данных. Когда бит регистра CONDition изменяется с 0 на 1, соответствующий PTR-бит определяет, будет ли установлен бит EVENT на 1.

PTR-бит =1: устанавливается бит EVENT

PTR-бит =0: бит EVENT не устанавливается.

Этот регистр может быть написан и прочитан. Считывание этого регистра не изменяет его содержание.

Регистр NTRansition (Negative Transition Register) также работает как фильтр передачи данных. При изменении бита регистра CONDition с 1 на 0 соответствующий NTR-бит определяет, будет ли установлен бит EVENT на 1.

NTR-бит =1: устанавливается бит EVENT

NTR-бит =0: бит EVENT не устанавливается.

Этот регистр может быть написан и прочитан. Считывание этого регистра не изменяет его содержание.

Регистр EVENT сообщает, произошло ли событие с его последнего считывания, этот регистр является «памятью» регистра CONDition. Он регистрирует только те события, которые прошли через транзитные фильтры. Регистр EVENT постоянно обновляется устройством. Пользователь может только прочитать его; считывание этого регистра очищает его содержание. Этот регистр часто называют общим.

Регистр ENABle определяет, влияет ли бит EVENT на суммарный бит (см. ниже). Каждый бит регистра EVENT связан (ANDed, символ "&") с соответствующим битом ENABle. События всех логических операций этого регистра маркируются ORed (символ "+") и передаются суммарному биту.

Бит ENAB = 0: соответствующий бит EVENT не оказывает влияния на суммарный бит.

Бит ENAB = 1: если соответствующий бит EVENT =1, то суммарный бит также устанавливается на 1.

Этот регистр может быть прочитан или написан по запросу. Его считывание не изменяет его содержимое.

Суммарный бит. Как указано выше, суммарный бит для каждого регистра получается из регистров EVENT и ENABLE. Результат вводится в бит регистра CONDition следующего по рангу регистра. Устройство автоматически генерирует суммарный бит для каждого регистра. Событие, напр., пропуск входного сигнала, таким образом может вызвать запрос на обслуживание через все иерархические уровни.

Стр. 42

5.8.2. Обзор регистра статуса

Схема 5-5 (Обзор регистра статуса) – см. оригинал. Список терминов, употребляющихся в схеме:

- Напряжение
- Текущее
- Время
- Мощность
- Температура
- Частота
- Фаза
- Модуляция
- Калибровка
- Прием сообщений FSW
- Доступно для дизайнера
- Резюме инструмента
- Командное предупреждение
- Настройка
- Классификация
- Развертка
- Коррекция
- Операция выполнена
- Запрос на управление
- Ошибка запроса
- Ошибка устройства
- Ошибка использования
- Ошибка команды
- Запрос пользователя

- Включение питания

Стр. 43

5.8.3. Описание регистра статуса

5.8.3.1. Биты статуса (STB) и службы включения запроса (SRE)

Байт статуса уже определен стандартом IEEE 488.2. Он находится в корне регистра статуса SCPI. Предопределенные биты стандарта IEEE 488.2 остаются неизменными. Третий и седьмой биты новые – это суммарные биты регистров статуса QUEStionable и OPERationable. Несмотря на то, что STB интегрирован в иерархию SCPI, есть некоторые нюансы.

Функция регистра SRE соответствует той же функции регистра STB ENABLE. Суммарный бит регистра STB – его собственный бит 6. STB не имеет регистра EVENt, он напрямую представляет состояние устройства в регистре CONDition.

Регистры PTRansition и NTRansition не имеют значений и не определяются.

Каждый бит в STB соответствует биту в SRE. Шестой бит системы SRE игнорируется. Если бит устанавливается в системе SRE и соответствующий бит в системе STB меняется с нуля на единицу, генерируется запрос на обслуживание (SRQ), который запускает перерыв в работе контроллера, проверяет, был ли контроллер настроен соответствующим образом и может ли он продолжать работу.

Байт статуса считывается посредством запроса *STB? SRE может быть запущено командой *SRE и считано посредством запроса *SRE?

Таблица 5.8-1. Значение битов, используемых в байтах статуса

Номер бита	Значение
2	Очередь ошибок не пуста. Этот бит устанавливается, когда в журнале ошибок появляется запись. Если этот бит активирован системой SRE, каждая запись в журнале ошибок генерирует запрос на обслуживание. Также ошибка может быть распознана и рассмотрена в деталях при совершении запроса в очередь ошибок.
3	Суммарный бит статуса QUEStionable Этот бит устанавливается, если бит EVENt присутствует в регистре QUEStionable, а соответствующий бит в регистре

	ENABLE устанавливается на 1. Установленный бит отражает сомнительное состояние устройства, которое может быть рассмотрено в деталях посредством запроса в регистр QUEStionable.
4	Бит MAV (сообщение доступно) Этот бит устанавливается, если в буфере вывода есть доступные к прочтению сообщения.
5	Бит ESB Суммарный бит регистра статуса событий. Этот бит устанавливается, если один из битов регистра статуса событий установлен и активирован в регистре статуса выполняемых событий (бит ENABLE=1). Появление этого бита отражает серьезную ошибку, которая может быть рассмотрена в деталях посредством запроса в регистр статуса событий.
6	Бит MSS (итоговый бит статуса) Этот бит устанавливается, если прибор запускает запрос на обслуживание. Это случается, когда один из остальных битов данного регистра встречается с несовместимым битом регистра SRE.
7	Суммарный бит регистра статуса OPERation Этот бит устанавливается, если бит EVENt фиксируется в регистре статуса OPERation, и соответствующий бит ENABLE равен 1. Установленный бит отражает, что в устройстве протекает некий процесс. Более подробная информация о процессе может быть предоставлена посредством запроса регистра статуса OPERation.

Стр. 44

5.8.3.2. Регистр статуса состояния (ESR) и статуса состояния событий (ESE)

Несмотря на то, что ESR соотносится со стандартом IEEE 488.2, это вполне хорошо вписывается в модель регистра SCPI. Формы ESE соотносятся с регистром ENABLE. Все биты PTRansition устанавливаются на логическом значении "1", а все биты NTRansition – на значении "0". Регистр статуса состояния может быть считан с помощью запроса *ESR? Регистр статуса состояния событий может быть установлен посредством команды *ESE и считан с помощью запроса *ESE?

Таблица 5.8-2

Значения битов, используемых в регистре статуса состояния

Номер бита	Значение
0	Операция завершена После приема команды *ОРС устанавливается этот бит, как только выполняются все предшествующие команды.
2	Ошибка запроса Этот бит устанавливается, если контроллер намеревается считать информацию с устройства без посылы запроса на действие. Частая причина этого – неисправный запрос, который невозможно обработать.
3	Зависящая от устройства ошибка Этот бит устанавливается, если происходит зависящая от устройства ошибка. Сообщение об ошибке с номером от -300 до -399 или с положительным номером отражает, что ошибка будет внесена в журнал ошибок. Подробнее см. раздел 5.10.1
4	Ошибка использования Этот бит устанавливается, если синтаксис полученной команды верный, но из-за каких-то условий команда не может быть обработана. Сообщение об ошибке с номером от -200 до -300 подробно описывает ошибку и заносит ее в журнал ошибок (см. раздел 5.10.1)
5	Ошибка команды Этот бит устанавливается, если получена неопределенная команда или команда с неверным синтаксисом. Сообщение об ошибке с номером от -100 до -200 подробно описывает ошибку и заносит ее в журнал ошибок (см. раздел 5.10.1)
6	Запрос пользователя Этот бит устанавливается при переходе с дистанционного управления на непосредственное, т.е. когда устройство переключается на ручное управление с помощью клавиши LOCAL.
7	Включение Этот бит устанавливается при включении питания прибора.