

# Подключение к ZET 030-I

## Описание протокола взаимодействия

---

<b>Установка подключения</b>	<b>2</b>
<b>Формат пакетов</b>	<b>3</b>
Пакет DEVICE_CONSOLE	5
Пакет DEVICE_TIME	5
Пакет STREAM_CONTROL	6
Пакет STREAM_TIME	7
Пакет STREAM_I24	7
Пакет FILE_OPERATION	9
Пакет FILE_DATA	10
Пакет FILE_RESULT	10
<b>Файл конфигурации</b>	<b>12</b>
<b>Сбор данных АЦП</b>	<b>15</b>
<b>Консольные команды</b>	<b>17</b>
Команда info	17
Команда reboot	17
Команда test	17

---

# Установка подключения

Используется протокол TCP/IPv4. Подключение производится по двум портам.

Таблица. Используемые порты TCP/IPv4

Имя порта	Номер	Описание
CMD	1832	Передача запросов и ответов. На данный момент номер не настраивается
ADC	CMD + 1	Прием данных от АЦП (от устройства)

Номер порта ADC определяется относительно номера порта CMD. По умолчанию: CMD 1832, ADC 1833.

Порт CMD двунаправленный, используется для отправки командных запросов и приема ответов на них.  
Порт ADC — однонаправленный, используется только для приема ответов с данными от АЦП устройства.

Процедура установки подключения:

1. Определить адрес IPv4 и номер командного порта CMD.
2. Установить соединение с командным портом CMD.
3. Установить соединение с портом данных ADC.

Адрес IPv4 и номер порта должны быть известны заранее. Если IP-адрес и порт неизвестны, можно сбросить сетевые настройки с помощью кнопки Reset на корпусе устройства, по умолчанию адрес 192.168.1.100, порт 1832.

Подключение к устройству считается установленным после соединения со всеми портами.

Отключение от устройства производится путем разрыва соединения со всеми портами.

Устройство со своей стороны разрывает соединение в следующих ситуациях:

- 1) закрытие соединения с любым из портов;
- 2) прием некорректного формата пакета;
- 3) изменение или сброс сетевых настроек;
- 4) длительное отсутствие связи по порту (механизм TCP Keep Alive).

# Формат пакетов

Каждый пакет начинается с заголовка фиксированного формата.

Здесь и далее типы полей, упоминаемые в таблицах, в большинстве своем соответствуют типам, определенным в заголовочном файле `<stdint.h>` для языка Си. Исключение составляет тип `ptr`, описание которого приводится ниже. Все значения записываются в порядке `little-endian`, младшим байтом вперед.

Таблица. Перечень используемых типов полей

Тип поля	Размер, байты	Описание
char	1	Строковый символ или часть символа в кодировке UTF-8
uint8_t	1	Беззнаковое 8-битное целое (0 .. +255)
uint16_t	2	Беззнаковое 16-битное целое (0 .. +65535)
int16_t	2	Знаковое 16-битное целое (-32768 .. +32767)
uint32_t	4	Беззнаковое 32-битное целое (0 .. +4294967295)
int32_t	4	32-битное целое со знаком (-2147483648 .. 2147483647)
uint64_t	8	Беззнаковое 64-битное целое (0 .. +18446744073709551615)
float	4	Число с плавающей запятой IEEE 754 одинарной точности
ptr	4	Относительный указатель (описание приводится ниже)

Заголовок пакета состоит из четырех 16-битных полей.

Таблица. Формат заголовка пакета

Смещение	Тип поля	Имя	Описание
0	uint16_t	full_size	Полный размер пакета в байтах, включая заголовок
2	uint16_t	token	Токен операции
4	uint16_t	code	Код пакета
6	uint16_t	root_size	Размер первого блока

Размер пакета указывается целиком, включая размер заголовка. Размер пакета должен быть кратен четырем. Минимально допустимый размер пакета — 8 байтов (только заголовок). Максимальный размер — 65532 байта, однако следует исходить из размера MTU для TCP/IP, обычно около 1400 байтов. Устройство поддерживает пакеты размером не более 2048 байтов.

Передача данных производится пакетами в форме «запрос—ответ». Клиент (программное обеспечение, подключенное к устройству) начинает операцию, отправляя запрос, состоящий из одного или более пакетов, а устройство отвечает на него также одним или несколькими пакетами. Все пакеты в рамках одной операции «запрос—ответ» имеют один и тот же номер токена.

Рекомендуется для каждого нового запроса изменять значение токена, например, инкрементировать на один (с учетом разрядности 16 битов).

Код пакета определяет назначение пакета и формат дальнейшего содержимого, следующего после заголовка. Если от устройства приходит пакет с неизвестным кодом, то такой пакет можно проигнорировать и перейти к разбору следующего пакета, пропустив `full_size` байтов.

Таблица. Перечень кодов пакетов

Код пакета	Название	Описание
0x4344	DEVICE_CONSOLE	Консольная команда
0x5444	DEVICE_TIME	Время устройства
0x4353	STREAM_CONTROL	Управление потоком данных АЦП
0x5453	STREAM_TIME	Время потока АЦП
0x3349	STREAM_I24	Данные АЦП, покадровые
0x4F46	FILE_OPERATION	Начало файловой операции
0x4446	FILE_DATA	Порция данных обрабатываемого файла
0x5246	FILE_RESULT	Завершение файловой операции

После заголовка может следовать содержимое. Все содержимое условно делится на непрерывные блоки. Размер каждого блока кратен четырем.

Размер самого первого блока указывается в заголовке пакета. Если этот размер равен нулю, то содержимое отсутствует, и такой пакет называется пустым.

Каждое поле блока — это либо непосредственное значение какого-либо параметра, либо относительный указатель на другой блок (`ptr`).

Указатель на блок — это два 16-битных числа, следующих друг за другом.

Таблица. Формат относительного указателя

Смещение	Тип поля	Имя	Описание
0	<code>int16_t</code>	<code>offset</code>	Смещение указываемого объекта относительно адреса указателя
2	<code>uint16_t</code>	<code>size</code>	Размер указываемого объекта в байтах

Смещение на начало указываемого блока приводится относительно текущей позиции указателя. Смещение может быть как положительным, если указываемый блок находится после указателя, так и отрицательным, если блок находится до указателя. Если смещение равно нулю, то такой указатель считается нулевым, а указываемый блок отсутствует. Нулевые указатели используются для необязательных блоков.

Указатель и указываемый блок всегда выравниваются по границе в четыре байта.

Размер указываемого блока всегда приводится в байтах. Размер может иметь любую кратность (например, 7), но занимаемое этим блоком место в пакете всегда выравнивается до кратности четырем, а недостающие до кратности байты заполняются нулями.

Содержимое указываемого блока определяется заранее форматом того блока, в котором содержится сам указатель.

Блок может быть текстовым, то есть содержать текст в кодировке UTF-8. В конце текстового блока должен быть как минимум один нулевой байт. Нулевые байты допускаются также и в самом тексте, а не только в конце. Например, блок с текстом «Привет!» будет занимать в пакете 16 байтов: 13 байтов самого текста в кодировке UTF-8, один обязательный нулевой байт и два нулевых байта для дополнения блока до размера кратного четырем. При этом в указателе на этот блок будет указан размер 13.

Все блоки сформированы таким образом, чтобы их можно было использовать напрямую в памяти как структуры на языке Си.

Почти все пакеты, если это не указано явно, передаются по командному порту.

## Пакет DEVICE\_CONSOLE

Пакет с кодом DEVICE\_CONSOLE используется для выполнения консольной команды.

Таблица. Формат блока консольного управления DEVICE\_CONSOLE

Смещение	Тип	Имя	Описание
0	ptr	text_str	Указатель на текстовую строку с командой или ответом

Указываемая текстовая строка содержит команду или ответ на нее.

Команды передаются в виде «команда [аргументы...]». Команды и ответы передаются в нижнем регистре. Символы перевода строки (CR или LF) в конце строки не требуются.

Таблица. Пример пакета DEVICE\_CONSOLE

Содержимое, hex	Значение	Описание
1800	24	Полный размер пакета
0100	1	Токен
4443	0x4344	Код DEVICE_CONSOLE
0400	4	Размер первого блока
0400 0A00	4 10	Указатель на строку, смещение 4, размер строки 10
746573742073686F 7274	«test short»	Команда «test short»
0000	0,0	Завершение строки и дополнение до кратности четырем

## Пакет DEVICE\_TIME

Пакет с кодом DEVICE\_TIME используется для получения или установки текущего времени устройства. Время указывается в формате UNIX epoch time — это количество секунд прошедших с 01.01.1970 00:00:00.

Таблица. Формат блока управления временем устройства DEVICE\_TIME

Смещение	Тип	Имя	Описание
0	uint64_t	time_sec	Время в формате UNIX epoch time

Пакет-запрос, содержащий время, используется для установки времени, пустой пакет-запрос — для получения текущего времени. В ответ всегда приходит пакет с текущим временем устройства.

Рекомендуется использовать глобальное время (UTC), но для удобства можно переключиться на локальное время (например, MSK = UTC+3).

Таблица. Пример пакета DEVICE\_TIME

Содержимое, hex	Значение	Описание
1000	16	Полный размер пакета
0200	2	Токен
4454	0x5444	Код DEVICE_CONSOLE
0800	8	Размер первого блока
8085746700000000	1735689600	Значение time_t, соответствующее 2025-01-01 00:00:00.

## Пакет STREAM\_CONTROL

Пакет с кодом STREAM\_CONTROL используется для запуска или остановки АЦП. После включения прибора АЦП не запущен. При запущенном АЦП устройство выдает снятые с него данные по порту ADC.

Таблица. Формат блока управления потоком STREAM\_CONTROL

Смещение	Тип	Имя	Описание
0	uint32_t	control	Управление потоком АЦП (0 — остановить, 1 — запустить)

Для запуска АЦП необходимо отправить пакет-запрос с полем control равным 1. После этого прибор будет отправлять пакеты-ответы с данными от АЦП по порту данных ADC, вплоть до остановки АЦП. Все такие пакеты считаются одной длительной операцией и имеют одинаковый токен.

Для остановки АЦП необходимо отправить пакет-запрос с полем control равным 0. При этом устройство еще может некоторое время отправлять пакеты АЦП, оставшиеся с последнего запуска, — их можно по желанию обрабатывать или игнорировать. Остановка АЦП будет также выполнена устройством автоматически, в случае невозможности опроса АЦП или передачи пакетов на ПК.

В ответ на запрос по командному порту CMD сначала отправляется пакет STREAM\_CONTROL для подтверждения команды, а затем, если АЦП был запущен, по порту данных ADC отправляются пакеты-ответы STREAM\_TIME и STREAM\_I24.

Таблица. Пример пакета STREAM\_CONTROL

Содержимое, hex	Значение	Описание
0C00	12	Полный размер пакета
0300	3	Токен запроса (рекомендуется менять)
5343	0x4353	Код STREAM_CONTROL
0400	4	Размер первого блока
01000000	1	Поле control (1 - запрос на запуск АЦП).

## Пакет STREAM\_TIME

Пакет STREAM\_TIME используется для передачи времени потоковых данных. В пакете указывается время в секундах.

Таблица. Формат блока информации о текущем времени потока STREAM\_TIME

Смещение	Тип	Имя	Описание
0	uint64_t	stream_time	Время потока в формате UNIX epoch time, с

Обычно ответ STREAM\_TIME выдается раз в секунду по порту данных ADC, в каждом следующем пакете время увеличивается на одну секунду. Между пакетами STREAM\_TIME передаются пакеты STREAM\_I24 таким образом, что суммарное количество кадров в них равно частоте дискретизации АЦП (например, при частоте дискретизации 100 кГц передается по 500 пакетов STREAM\_I24, каждый содержит по 200 кадров).

Таблица. Пример пакета STREAM\_TIME

Содержимое, hex	Значение	Описание
1000	16	Полный размер пакета
0300	3	Токен (соответствует текущей операции запуска АЦП)
5354	0x5453	Код STREAM_TIME
0800	8	Размер первого блока
7306756700000000	1735722611	Время 2025-01-01 09:10:11

## Пакет STREAM\_I24

Пакет с кодом STREAM\_I24 используются для передачи 24-битных кодов АЦП.

Таблица. Формат блока потоковых 24-битных кодов STREAM\_I24

Смещение	Тип	Имя	Описание
0	uint32_t	frame_counter	Счетчик кадров в пределах текущей секунды
4	ptr	data_ptr	Указатель на данные

Оцифровка производится кадрами, каждый кадр содержит по одному отсчету с каждого активного канала АЦП. Количество кадров за секунду соответствует частоте дискретизации АЦП.

Каждый кадр имеет время оцифровки, которое состоит из двух компонентов:

- stream\_time — время в секундах в формате UNIX epoch time;
- frame\_counter — номер кадра в пределах этой секунды (от 0 до  $FREQ - 1$ , включительно).

Счетчик кадров можно перевести в доли секунды путем деления на частоту дискретизации.

$$\text{double stream\_subsec} = ((\text{double})\text{frame\_counter}) / \text{FREQ};$$

Коды всегда передаются кадрами, в одном пакете кадры передаются целиком, то есть не бывает такого, чтобы в один пакет попала только часть одного кадра, а оставшаяся часть — в следующий пакет.

Например, если включено три канала (первый, второй и четвертый), то последовательность байтов будет следующей:

#### Листинг. Пример последовательности данных в пакете STREAM\_I24

frame0_channel0_byte0, frame0_channel0_byte1, frame0_channel0_byte2, // кадр 1, канал 1
frame0_channel11_byte0, frame0_channel11_byte1, frame0_channel11_byte2, // кадр 1, канал 1
frame0_channel12_byte0, frame0_channel12_byte1, frame0_channel12_byte2, // кадр 1, канал 1
frame1_channel0_byte0, frame1_channel0_byte1, frame1_channel0_byte2, // кадр 2, канал 1
frame1_channel11_byte0, frame1_channel11_byte1, frame1_channel11_byte2, // кадр 2, канал 1
frame1_channel12_byte0, frame1_channel12_byte1, frame1_channel12_byte2, // кадр 2, канал 1
... // кадры 3..(N-1)
frameN_channel0_byte0, frameN_channel0_byte1, frameN_channel0_byte2, // кадр N, канал 1
frameN_channel11_byte0, frameN_channel11_byte1, frameN_channel11_byte2, // кадр N, канал 1
frameN_channel12_byte0, frameN_channel12_byte1, frameN_channel12_byte2. // кадр N, канал 1

Количество кадров определяется исходя из размера данных в байтах и количества активных каналов:

$$\text{frame\_count} = \text{data\_ptr.size} / \text{channel\_count} / 3$$

Частота дискретизации и список активных каналов должны быть известны заранее из файла конфигурации.

Например, если размер данных 45, активно три канала, то в пакете будет  $45 / 3 / 3 = 5$  кадров.

Таблица. Пример пакета STREAM\_I24

Содержимое, hex	Значение	Описание
4000	64	Полный размер пакета
0300	3	Токен (соответствует текущей операции запуска АЦП)
4933	0x3349	Код STREAM_I24
0800	8	Размер первого блока
0A000000	10	Счетчик кадров 10 (10-й кадр с начала секунды)
0400	4	Указатель на данные, смещение 4,
2D00	45	размер 45 байтов ( $45 / 3 / 3 = 5$ кадров)
010000	+1	Кадр 1, канал 1
E80300	+1000	Кадр 1, канал 2
FEFFFF	-2	Кадр 1, канал 4
010000	+1	Кадр 2, канал 1
E90300	+1001	Кадр 2, канал 2
FEFFFF	-2	Кадр 2, канал 4
010000	+1	Кадр 3, канал 1
E80300	+1000	Кадр 3, канал 2
FEFFFF	-2	Кадр 3, канал 4
010000	+1	Кадр 4, канал 1
E80300	+1000	Кадр 4, канал 2
FEFFFF	-2	Кадр 4, канал 4
010000	+1	Кадр 5, канал 1
EA0300	+1002	Кадр 5, канал 2
FEFFFF	-2	Кадр 5, канал 4
000000	0,0,0	Дополнение до кратности четырем



## Пакет FILE\_OPERATION

Пакет с кодом FILE\_OPERATION используется для начала файловой операции на чтение, запись или удаление.

Таблица. Формат блока файловой операции FILE\_OPERATION

Смещение	Тип	Имя	Описание
0	ptr	path_str	Указатель на строку, содержащую путь к файлу
4	uint32_t	operation	Код файловой операции

Таблица. Перечень кодов файловой операции

Код	Операция	Описание
0x44414F4C	LOAD	Запрос на загрузку файла из внутренней памяти устройства
0x45564153	SAVE	Запрос на сохранение файла во внутренней памяти устройства
0x544C4544	DELT	Запрос на удаление файла из внутренней памяти устройства

Файлы загружаются и сохраняются целиком.

В запросе на загрузку файла (LOAD) в ответ сначала приходит содержимое пакета в виде пакетов FILE\_DATA (при наличии такого содержимого), а в конце — пакет FILE\_RESULT.

В запросе на сохранение файла (SAVE) необходимо после пакета FILE\_OPERATION отправить содержимое файла в виде пакетов FILE\_DATA, последний пакет FILE\_DATA должен иметь нулевой указатель на данные, что трактуется как признак конца файла. Устройство попытается сохранить файл, при необходимости проверить на корректность (если это файл конфигурации) и затем ответит пакетом FILE\_RESULT.

В случае обнаружения ошибки устройство может завершить операцию и прислать ответ FILE\_RESULT с кодом ошибки до того, как будет отправлено все содержимое файла. В этом случае клиент может либо досрочно прекратить передачу оставшегося содержимого, либо продолжить отправлять данные, которые будут проигнорированы устройством.

В запросе на удаление файла (DELT) в ответ приходит пакет FILE\_RESULT.

Таблица. Пример пакета FILE\_OPERATION

Содержимое, hex	Значение	Описание
1800	24	Полный размер пакета
0500	5	Токен
464F	0x4F46	Код FILE_OPERATION
0800	8	Размер первого блока
0800 0800	8 8	Указатель на путь, смещение 8, размер строки 8
4C4F4144	0x44414F4C	Операция LOAD
636F6E66 2E786D6C	conf.xml	Путь к файлу «conf.xml»

000000000	0,0,0,0	Завершение строки и дополнение до кратности четырем
-----------	---------	---

## Пакет FILE\_DATA

Пакет с кодом FILE\_DATA используется для передачи очередной порции содержимого файла. Содержимое небольших файлов помещается в один пакет, для больших файлов требуется несколько пакетов, которые передаются друг за другом по порядку.

При загрузке файла (LOAD) пакеты FILE\_DATA передаются устройством в ответ на запрос, при сохранении (SAVE) — в сторону устройства как часть запроса. При удалении файла (DELT) пакеты FILE\_DATA не передаются. Пакеты FILE\_DATA также не передаются при обнаружении ошибки, например, если загружаемый файл не найден во внутренней памяти устройства.

Таблица. Формат блока с содержимым файла FILE\_DATA

Смещение	Тип	Имя	Описание
0	uint32_t	offset	Текущая позиция в файле
4	ptr	data_ptr	Указатель на содержимое файла

Текущая позиция относится к началу передаваемых в пакете данных. Поле позволяет проверить корректность последовательности пакетов FILE\_DATA. Значение offset в текущем пакете FILE\_DATA должно быть равно сумме значения offset и размера данных в предыдущем пакете FILE\_DATA.

Указатель data\_ptr может быть нулевым, в таком случае пакет становится маркером конца файла EOF. Пакет EOF необходимо отправлять последним при операции сохранения файла, чтобы устройство могло определить, что содержимое файла передано полностью, и приступить к завершению операции.

В операции загрузки файла маркер EOF возможен, но необязателен, так как признаком окончания файла может служить пакет FILE\_RESULT.

## Пакет FILE\_RESULT

Пакет с кодом FILE\_RESULT используется для передачи результата выполнения файловой операции. Пакет выдается последним в ответе на запрос FILE\_OPERATION, после возможного обмена пакетами FILE\_DATA.

Таблица. Формат блока с результатом файловой операции FILE\_RESULT

Смещение	Тип	Имя	Описание
0	ptr	path_str	Путь к файлу, операция к которому была запрошена
4	uint32_t	result	Код результата файловой операции

Поле path\_str дублирует путь, указанный в пакете FILE\_OPERATION.

Поле result содержит результат выполнения файловой операции.

Таблица. Перечень кодов результатов файловой операции

Код	Результат	Описание
0	OK	Операция выполнена успешно
1	BUSY	Прибор занят другой файловой операцией
2	NOT_FOUND	Запрошенный путь некорректен или ведет к отсутствующему файлу
3	IO_ERROR	Ошибка при обращении к памяти устройства
4	NOT_SUPPORTED	Операция не поддерживается (неизвестный код операции)
5	FORMAT_ERROR	Ошибка формата (в случае сохранения файла конфигурации conf.xml)
6	CANCELLED	Операция отменена, например, из-за начала другой операции

# Файл конфигурации

Все настройки и основные характеристики устройства, включая калибровочные поправки, хранятся в памяти устройства в файле конфигурации. Имя файла конфигурации — conf.xml, формат — XML, кодировка — UTF-8.

После подключения к устройству необходимо загрузить содержимое файла из памяти устройства, с помощью запроса FILE\_OPERATION с кодом операции LOAD и путем conf.xml. Содержимое можно хранить в оперативной памяти программного обеспечения или на диске.

В случае необходимости можно изменить некоторые настройки и записать новое содержимое обратно в память устройства. Для записи необходимо отправить запрос FILE\_OPERATION с кодом операции SAVE и путем conf.xml, а затем — серию пакетов FILE\_DATA с новым содержимым файла. Устройство проверит предлагаемые настройки и либо сохранит у себя весь файл целиком, либо полностью его отклонит. Для проверки успешности применения настроек рекомендуется снова загрузить содержимое файла с помощью файловой операции LOAD.

Сохранение файла необходимо проводить при остановленном АЦП.

Программное обеспечение может по своему усмотрению добавить в файл собственные элементы, при условии, что эти элементы не используются устройством.

Листинг. Пример файла конфигурации

```
<?xml version="1.0"?>
<Config version="1.2">
  <Device name="ZET 030-I" type="30" serial="23001">
    <Description label="" />
    <DigitalResolChanADC>4.65661e-09,4.65661e-09,4.65661e-09,4.65661e-09</DigitalResolChanADC>
    <Ethernet method="static" addr="192.168.1.100/24" ftp="no" />
    <Freq>25000</Freq>
    <Channel>0xf</Channel>
    <HCPChannel>0x0</HCPChannel>
    <KodAmplify>0,0,0,0</KodAmplify>
    <Recorder start="auto" />
    <RecordMinutes>0</RecordMinutes>
  </Device>
</Config>
```

Таблица. Перечень элементов конфигурационного файла

Элемент	Описание
Description	Описание устройства, используется для удобства идентификации, если необходимо назначить устройства пользовательское имя.
DigitalResolChanADC	Поканальный список коэффициентов преобразования кодов в вольты.  Коэффициенты применяются к 32-битным кодам. Чтобы преобразовать в вольты 24-битный код АЦП, полученный от прибора, необходимо сначала привести его к 32-битному, затем умножить на соответствующий коэффициент преобразования и поделить на текущий коэффициент усиления (1 или 30):  $AdcCode32[n] = AdcCode24[n] * 256;$ $U[n] = AdcCode32[n] * DigitalResolChanADC[n] / GainRatio[n] \quad \{volts\}$
Ethernet	Сетевые настройки.

	<p>Настраивается способ получения IP-адреса, IP-адрес и маска подсети по умолчанию, включение протокола FTP.</p> <p>Настройки передаются в атрибутах:</p> <ul style="list-style-type: none"> <li>- method="static": статические IP-адрес и маска подсети</li> <li>- method="dhcp": динамическое получение IP-адреса и маски подсети</li> <li>- addr="&lt;ip&gt;": IP-адрес, маска подсети 255.255.255.0</li> <li>- addr="&lt;ip&gt;/&lt;subnet_length&gt;": IP-адрес и маска подсети (в виде числа)</li> <li>- ftp="yes": включение FTP для чтения файлов на SD-карте</li> <li>- ftp="no": отключение FTP</li> </ul>
Freq	<p>Частота дискретизации АЦП, Гц.</p> <p>Допустимые значения: 1000, 3125, 6250, 12500, 25000, 50000, 100000, 200000, 400000.</p> <p>Необходимо учесть, что при выборе высоких частот дискретизации устройству не хватит пропускной способности для передачи данных в полном объеме. Проверенный максимум передачи данных по TCP/IPv4 — 400 кГц суммарно, то есть 100 кГц по всем четырем каналам или 400 кГц, но только с одним активным каналом.</p>
Channel	<p>Битовая маска активных каналов.</p> <p>Для включения канала 1 необходимо выставить бит 0x0001, для канала 2 — бит 0x0002, для канала 3 — 0x0004, для канала 4 — 0x0008.</p> <p>Значение 0x000F означает, что включены все четыре канала. Значение 0x0000 не допускается (нельзя отключить все каналы).</p>
HCPCChannel	<p>Битовая маска включения IEPЕ.</p> <p>Для включения IEPЕ в канале необходимо установить соответствующий бит, по аналогии с маской Channel. Настройка учитывается только для активных каналов.</p>
KodAmplify	<p>Поканальный список индексов коэффициентов усиления.</p> <p>Для каждого канала выставляется КУ по соответствующему индексу:</p> <ul style="list-style-type: none"> <li>- индекс 0: КУ 1 (GainRatio = 1)</li> <li>- индекс 1: КУ 30 (GainRatio = 30)</li> </ul>
Recorder	<p>Режим запуска автономной записи.</p> <p>Режим выбирается через атрибут start.</p> <p>Возможные варианты:</p> <ul style="list-style-type: none"> <li>- start="auto": начинать запись автоматически после подачи питания;</li> <li>- start="button": начинать и останавливать запись по нажатию кнопки.</li> </ul> <p>Автономная запись возможна, когда к прибору нет подключения со стороны ПК. Во время записи создаются файлы с расширением .zsdata. Размер каждого файла ограничен (не более 2000 ГБ), при достижении макс. размера будет создан новый файл, и запись продолжена.</p>
RecordMinutes	<p>Длительность каждой записи в минутах.</p>

	<p>Если указать 0, то запись будет непрерывной. Запись будут продолжаться вплоть до отключения питания или подключения к устройству со стороны ПК, а также по нажатию кнопки в режиме button.</p> <p>Если указать ненулевое значение, то каждая запись будет записываться указанное число минут, затем запись будет либо остановлена до очередного нажатия кнопки (в режиме button), либо перезапущена на такое же время (в режиме auto).</p> <p>Максимально можно задать 30 минут. В более поздних версиях прошивках максимальная длительность увеличена до 1500 минут (25 часов).</p>
--	---

# Сбор данных АЦП

Для сбора данных с устройства необходимо выполнить следующие действия:

- 1) подключиться к устройству;
- 2) считать файл конфигурации;
- 3) при необходимости настроить измерительные параметры;
- 4) подать команду на запуск АЦП;
- 5) начать прием пакетов с очередными кодами, считанными с выбранных каналов АЦП;
- 6) преобразовывать получаемые коды АЦП в напряжение на входе.

Пакет `STREAM_CONTROL` используется для запуска или остановки АЦП.

Пакет `STREAM_CONTROL` передается по порту `CMD`, а пакеты `STREAM_TIME` и `STREAM_I24` — по порту `ADC`.

Потоковые данные АЦП выдаются в пакетах `STREAM_I24`. По тому же порту `ADC` выдаются пакеты `STREAM_TIME`, которые могут быть полезны, если необходима привязка данных ко времени.

При обработке пакетов `STREAM_I24` и `STREAM_TIME` следует игнорировать пакеты с другими токенами.

В случае необходимости изменения настроек, необходимо остановить АЦП, сохранить новое содержимое файла и затем запустить АЦП.

Преобразование получаемых кодов АЦП в напряжение на входе выполняется путем умножения кода на коэффициент преобразования, соответствующий каналу, и учесть текущий коэффициент усиления. Все эти коэффициенты берутся из файла конфигурации. При этом необходимо 24-битные коды преобразовать в 32-битные.

Листинг. Пример обработки пакета `STREAM_I24` с преобразованием кодов в вольты

```
uint32_t DigitalResolChanADC[4]; // список коэффициентов преобразования из файла конфигурации
uint32_t Channel; // битовая маска активных каналов из файла конфигурации
uint32_t KodAmplify[4]; // список настройки КУ из файла конфигурации

// Определяем количество активных каналов и вычисляем обобщенный коэффициент преобразования
uint32_t channel_count = 0;
float ratio[4];
for (int ch = 0; ch < 4; ch++) {
    if (Channel & (1u << ch)) {
        channel_count++;
        ratio[ch] = DigitalResolChanADC[ch] / (KodAmplify[ch] ? 30 : 1);
    }
}

const uint8_t *stream_data; // указатель на начало данных в пакете STREAM_I24
uint32_t stream_data_size; // размер данных в пакете STREAM_I24
uint32_t frame_count = stream_data_size / channel_count / 3; // определяем количество кадров

union {
    uint8_t bytes[4];
    int32_t code;
} sample;
```

```

for (int ch = 0; ch < 4; ch++) { // по всем каналам
    if (Channel & (1u << ch)) { // только если канал активен
        const uint8_t *src = stream_data;
        for (int i = 0; i < frame_count; i++) {
            // приводим 24-битный к 32-битному с сохранением знака, младший байт нулевой
            sample.bytes[0] = 0;
            sample.bytes[1] = src[0];
            sample.bytes[2] = src[1];
            sample.bytes[3] = src[2];

            // формируем значение по каналу в вольтах
            float volts = ((float)sample.code) * ratio[ch];

            // переходим к следующему каналу
            src += channel_count * 3;
        }

        stream_data += 3; // переходим к данным следующего активного канала
    }
}

```



# Консольные команды

Настройки, передаваемые через файл конфигурации, сохраняются в постоянной памяти устройства и применяются после сброса питания или перезагрузки устройства. Консольные команды предназначены для обращения к устройству без изменения файла конфигурации, например, для получения какой-либо оперативной информации или для выполнения определенных действий, таких как перезагрузка устройства или активация функций самотестирования.

Команды управления передаются в пакетах DEVICE\_CONSOLE. Все команды текстовые, в кодировке UTF-8, в нижнем регистре.

В ответ на команду приходит также текстовый ответ. В случае ошибки, например, если команда не распознана или не поддерживается, выдает строка «error». В случае успешного выполнения команды выдается запрошенная информации или строка «ok», если команда подразумевает выполнение действия без получения информации.

## Команда info

Команда info позволяет запросить общую информацию об устройстве, такую как имя устройства, серийный номер и версию устройства (аппаратную и программную).

Таблица. Перечень запрашиваемой информации

Вариант	Описание
info name	Получить имя устройства, например, «ZET 030-I».
info serial	Получить серийный номер устройства, например, «23001».
info version	Получить версию устройства, в формате «hw.fw.date», где hw — номер аппаратной версии, fw — номер программной версии (прошивки), date — дата выпуска прошивки в виде шести цифр.

## Команда reboot

Команда reboot используется для перезагрузки устройства без физического сброса питания. Непосредственно перезапуск происходит через несколько секунд после отправки команды. Во время ожидания перезагрузки устройство перестает отвечать на новые запросы, а также может разорвать подключение.

## Команда test

Команда test используется для активации функций самотестирования устройства. При активации на входы каналов внутри устройства подается определенный сигнал, позволяющий оценить какую-либо характеристику АЦП по этому каналу, например, уровень собственных шумов. Тестирование включается сразу на все каналы. При этом желательно отсоединить внешние датчики.

Самотестирование не сохраняется и отключается при сбросе питания и после перезагрузки.

Команду можно подать как при остановленном, так и при уже запущенном АЦП. Таким образом, тестирование можно переключать «на лету».

Таблица. Перечень вариантов тестирования

Вариант	Описание
test off	Отключение тестов
test short	Закоротка входов между собой через резистор 50 Ом. Используется, например, для измерения собственных шумов АЦП
test sqr	Подача на входы сигнала в форме меандра, с частотой 4 Гц. Используется, например, для общей проверки измерений на адекватность
test neg	Подача на входы постоянного отрицательного уровня (то есть уровень на положительном входе ниже уровня на отрицательном). Используется, например, для проверки измерения отрицательных значений