

Сервер-ретранслятор CCU Relay

29 августа 2019 г.

Содержание

1	Назначение	3
2	Подключение контроллера	5
2.1	Подключение к ccu.sh	5
2.2	Подключение к CCU Relay	6
3	Установка	6
3.1	Для Linux	7
3.2	Для Windows	7
4	Обновление	8
5	Настройка	9
5.1	Файл конфигурации	9
5.2	Создание самоподписанного сертификата и ДН-параметров	10
5.3	Настройка сервиса systemd в Linux	10
5.4	Настройка службы в Windows	11
6	Лог	11
7	Запуск	11
8	Проверка состояния	11
9	Остановка	12
10	Перезапуск	12
11	Лицензия	12

11.1	Получение	12
11.2	Обновление	13
11.3	Информация	13
12	HTTP API	13
12.1	Ключ доступа	13
12.2	Типы запросов	13
12.3	Ошибки	14
12.4	URL для входа в CCU Shell	14
12.5	Контроллеры онлайн	14
12.6	Идентификация контроллера	15
12.7	Системная информация и режим охраны	16
12.8	Режим охраны	17
	12.8.1 Получение	17
	12.8.2 Изменение	18
12.9	Входы	19
12.10	Границы входов	20
	12.10.1 Типы тревожных диапазонов	20
	12.10.2 Получение	20
	12.10.3 Изменение	21
12.11	Выходы	22
	12.11.1 Типы выходов	22
	12.11.2 Получение	22
	12.11.3 Управление	23
12.12	Профили	23
12.13	События	24
	12.13.1 Источники событий	24
	12.13.2 Получение	25
	12.13.3 Подтверждение	28
13	Modbus TCP	28
13.1	Настройка	28
13.2	Функции	29
	13.2.1 Read Coils (01)	29
	13.2.2 Read Discrete Inputs (02)	29
	13.2.3 Read Input Registers (04)	30
	13.2.4 Write Single Coil (05)	30
	13.2.5 Write Multiple Coils (15)	30
13.3	Ошибки	31

Внимание!

Для выполнения ряда приведенных команд могут потребоваться права пользователя *root* (через *sudo*) в Linux или права администратора в Windows.

1 Назначение

Сервер-ретранслятор *CCU Relay* позволяет запустить собственный независимый аналог сервиса *ssi.sh* для контроллеров:

- CCU422-S
- CCU825-S
- CCU825-S+
- CCU825-B
- CCU706-G

И предоставляет следующие возможности:

- Удаленное конфигурирование контроллеров с помощью веб-браузера
- Управление контроллерами, получение данных и событий через HTTP API, интегрирование контроллеров в собственные программные комплексы
- Подключение контроллеров к любой SCADA-системе через Modbus TCP

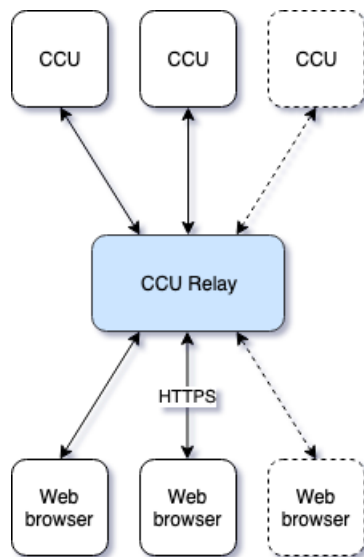


Рис. 1: Удаленное конфигурирование

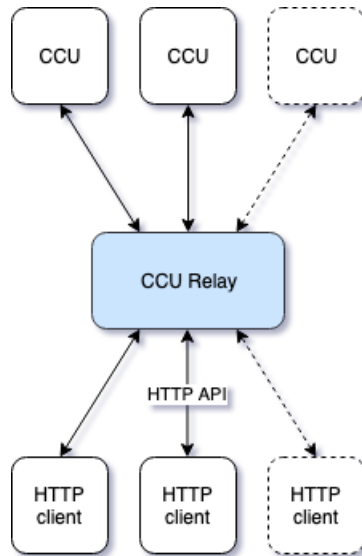


Рис. 2: Управление через HTTP API

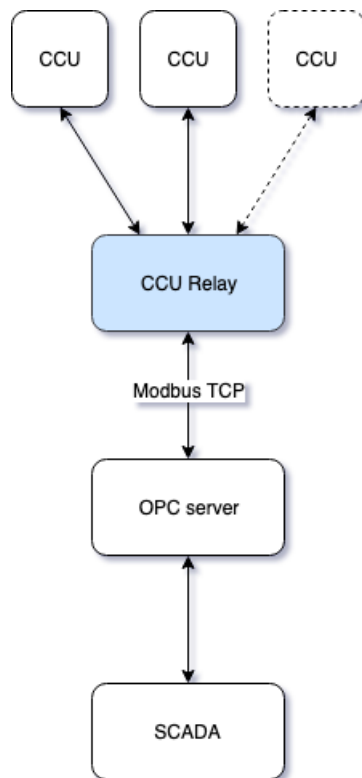


Рис. 3: Подключение к SCADA через Modbus TCP

2 Подключение контроллера

Контроллер может быть одновременно подключен к `ccu.sh` и трем собственным *CCU Relay* (соединения 1-3).

2.1 Подключение к `ccu.sh`

Подключение к `ccu.sh` требуется при использовании ботов.

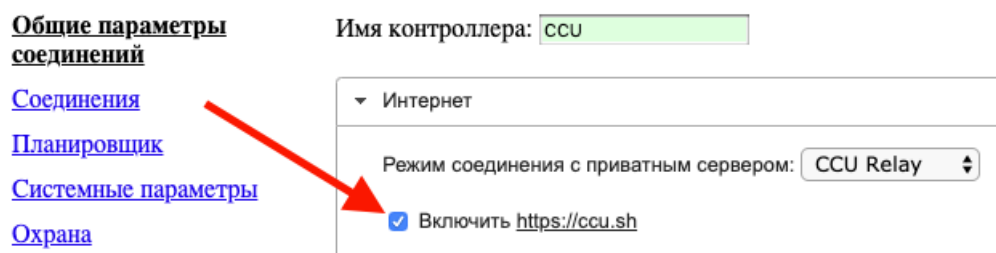


Рис. 4: Подключение к `ccu.sh`

Поддержка ботов включена для контроллеров модификаций В и G. Чтобы активировать опцию ботов для контроллеров модификаций S и S+, необходимо сделать запрос на support@radsel.ru, в котором отправить список IMEI контроллеров. Полученный текстовый лицензионный ключ вводится в инженерном меню контроллера.



Рис. 5: Активация опции ботов для модификаций S и S+

2.2 Подключение к CCU Relay

Общие параметры соединений

[Соединения](#)

[Планировщик](#)

[Системные параметры](#)

[Охрана](#)

Имя контроллера:

▼ Интернет

Режим соединения с частным сервером:

Включить <https://ccu.sh>

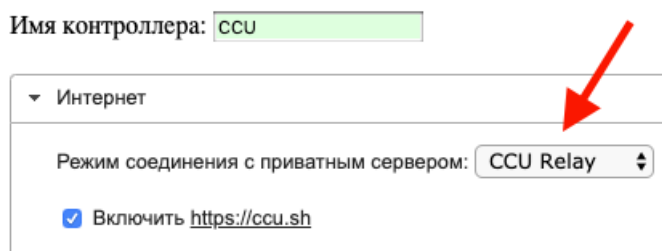


Рис. 6: Выбор режима соединения

В соединении 1-3 необходимо разрешить *CCU Relay*, ввести адрес и порт. Адрес (доменное имя или IP) должен соответствовать внешнему адресу компьютера, на котором работает *CCU Relay*, либо адресу роутера. Порт должен совпадать со значением *ccu_port*, заданным в файле конфигурации *CCU Relay* (см. пункт 5.1). Если используется роутер, то в нем необходимо пробросить данный порт на адрес компьютера с *CCU Relay*. Порт должен быть разрешен в сетевых экранах компьютера и роутера.

▼ Оповещение через CCU Relay TCP/IP

Разрешено

Параметры сервера

Тип адреса:

IP адрес:

TCP порт:

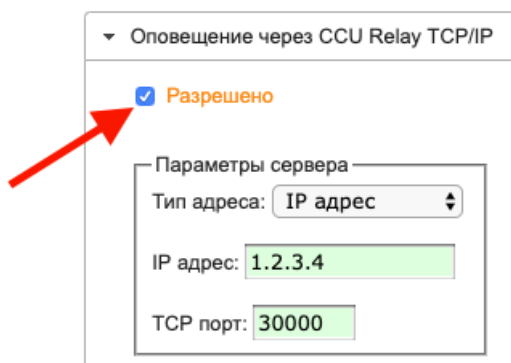


Рис. 7: Настройка адреса и порта CCU Relay

3 Установка

Сервер-ретранслятор *CCU Relay* поставляется в виде файла (см. таблицу 1). Поле *<version>* задает версию. Поддерживается только 64-битная архитектура.

Файл	Операционная система	Архитектура
ccu_relay-<version>-linux.tar.gz	Linux	x86-64
ccu_relay-<version>-windows.zip	Windows	x86-64

Таблица 1: Файлы CCU Relay

3.1 Для Linux

```
mkdir -p /opt/ccu_relay
tar xf ccu_relay-<version>-linux.tar.gz \
    -C /opt/ccu_relay --no-same-owner
```

После этого необходимо настроить сервис systemd (см. пункт 5.3).

3.2 Для Windows

Сначала необходимо установить пакет Update for Visual C++ 2013 Redistributable Package¹. Файл vc_redist_x64.exe. В таблице для скачивания необходимо выбрать язык, например, Russian - Russia.

¹<https://support.microsoft.com/ru-ru/help/4032938/update-for-visual-c-2013-redistributable-package>

vcredist_x64.exe File Path

Locales	File Paths
Czech - Czech Republic	https://aka.ms/highdpimfc2013x64csy
German - Germany	https://aka.ms/highdpimfc2013x64deu
English - United States	https://aka.ms/highdpimfc2013x64enu
Spanish (traditional Sorts) - Spain	https://aka.ms/highdpimfc2013x64esn
French - France	https://aka.ms/highdpimfc2013x64fra
Italian - Italy	https://aka.ms/highdpimfc2013x64ita
Japanese - Japan	https://aka.ms/highdpimfc2013x64jpn
Korean - Korea	https://aka.ms/highdpimfc2013x64kor
Polish - Poland	https://aka.ms/highdpimfc2013x64plk
Portuguese - Brazil	https://aka.ms/highdpimfc2013x64ptb
<u>Russian - Russia</u>	<u>https://aka.ms/highdpimfc2013x64rus</u>
Turkish - Turkey	https://aka.ms/highdpimfc2013x64trk
Chinese - China	https://aka.ms/highdpimfc2013x64chs
Chinese - Taiwan	https://aka.ms/highdpimfc2013x64cht

Рис. 8: Файл vcredist_x64.exe

Распаковать файл `csu_relay-<version>-windows.zip` в директорию `C:\csu_relay` (путь может быть любым, но не должен содержать пробелы). После этого необходимо настроить службу (см. пункт 5.4).

4 Обновление

1. Если `csu_relay` запущен, то перед обновлением его необходимо остановить (см. пункт 9).
2. Для Windows необходимо удалить службу (см. пункт 5.4).
3. Сохранить директорию `config`, если в ней были изменения.
4. Распаковать новую версию вместо старой.
5. Восстановить директорию `config`, если это необходимо.

6. Для Windows необходимо добавить службу (см. пункт 5.4).
7. Запустить *ccu_relay* (см. пункт 7).

5 Настройка

5.1 Файл конфигурации

Листинг 1: config/ccu_relay.config

```
{host, "nohost"}.
{ccu_port, 30000}.
{https_port, 8443}.
% {ssl_cacertfile, "ssl/cacert.pem"}.
{ssl_certfile, "ssl/cert.pem"}.
{ssl_keyfile, "ssl/privkey.pem"}.
{ssl_dhfile, "ssl/dh.pem"}.
{http_api_key, "nokey"}.
{modbus, [{1, 502}]}.
{license_keyfile, "license.key"}.
```

Строка, которая начинается с символа %, является комментарием и игнорируется.

Параметр	Тип	Описание
host	строка	Имя хоста
ccu_port	число	CCU TCP-порт
https_port	число	HTTPS TCP-порт
ssl_cacertfile	строка	Файл корневого сертификата (опционально)
ssl_certfile	строка	Файл сертификата
ssl_keyfile	строка	Файл приватного ключа сертификата
ssl_dhfile	строка	Файл DH-параметров
http_api_key	строка	HTTP API ключ доступа
modbus	список пар	Каждая пара: {адрес шлюза, TCP-порт шлюза}
license_keyfile	строка	Файл лицензии

Таблица 2: Параметры конфигурации

Имя хоста *host* используется при получении URL для входа в CCU Shell (см. пункт 12.4).

TCP-порт *ccu_port* используется для подключения контроллеров CCU. Это же значение должно быть задано в конфигурации контроллера (см. пункт 2.2).

TCP-порт *https_port* используется для HTTP API.

TCP-порт со значением меньше 1024 считается привилегированным. Требуется запуск с правами пользователя *root* или администратора.

Для HTTPS требуется SSL-сертификат. В поставку входит самоподписанный сертификат. При необходимости данный сертификат можно пересоздать (см. пункт 5.2), либо использовать сертификат, выданный центром сертификации (CA). Сертификат и ключ сертификата должны быть в PEM-кодировке.

Путь к файлу может задаваться как:

- относительный путь (относительно директории *config*)
- абсолютный путь

Параметр конфигурации *modbus* описан в пункте 13.1.

5.2 Создание самоподписанного сертификата и ДН-параметров

```
openssl req -x509 -days 36500 -newkey rsa:2048 -nodes \  
    -keyout privkey.pem -out cert.pem  
openssl dhparam -out dh-params.pem 2048
```

5.3 Настройка сервиса systemd в Linux

Для автозапуска в современных системах Linux используется systemd.

Листинг 2: `/etc/systemd/system/ccu_relay.service`

```
[Unit]  
Description=Runner for ccu_relay  
After=network.target  
  
[Service]  
WorkingDirectory=/opt/ccu_relay  
ExecStart=/opt/ccu_relay/bin/ccu_relay start  
ExecStop=/opt/ccu_relay/bin/ccu_relay stop  
User=root  
RemainAfterExit=yes  
  
[Install]  
WantedBy=multi-user.target
```

Необходимо создать unit-файл сервиса:

```
cp /opt/ccu_relay/config/systemd/ccu_relay.service /etc/systemd/system/
```

Разрешить сервис:

```
systemctl daemon-reload  
systemctl enable ccu_relay.service
```

После перезагрузки сервис должен запуститься автоматически. Для контроля см. пункт 8.

5.4 Настройка службы в Windows

Для добавления службы необходимо выполнить в командной строке *cmd*:

```
C:\ccu_relay\bin\ccu_relay.cmd install
```

Для удаления службы необходимо выполнить в командной строке *cmd*:

```
C:\ccu_relay\bin\ccu_relay.cmd uninstall
```

6 Лог

Директория лог-файлов:

- Linux: /opt/ccu_relay/log
- Windows: C:\ccu_relay\log

Основные типы лог-файлов:

- info.log — информационные сообщения
- error.log — сообщения об ошибках

7 Запуск

Для Linux:

```
/opt/ccu_relay/bin/ccu_relay start
```

Для Windows:

```
C:\ccu_relay\bin\ccu_relay.cmd start
```

В случае успешного запуска в *info.log* будет сообщение: *System started*.

В случае ошибки в *error.log* будет сообщение: *System startup failed*.

Для контроля см. пункт 8.

8 Проверка состояния

Для Linux:

- идентификатор процесса

```
/opt/ccu_relay/bin/ccu_relay pid
```

- проверка ping

```
/opt/ccu_relay/bin/ccu_relay ping
```

- статус сервиса systemd (см. пункт 5.3)

```
systemctl status ccu_relay
```

Для Windows:

- проверка ping

```
C:\ccu_relay\bin\ccu_relay.cmd ping
```

- проверка состояния в *Службах*.

9 Остановка

Для Linux:

```
/opt/ccu_relay/bin/ccu_relay stop
```

Для Windows:

```
C:\ccu_relay\bin\ccu_relay.cmd stop
```

В случае успешной остановки в *info.log* будет сообщение: *System stopped*.

10 Перезапуск

Перезапуск требуется после изменения файла конфигурации, если *ccu_relay* был запущен.

Для Linux:

```
/opt/ccu_relay/bin/ccu_relay restart
```

Для Windows:

```
C:\ccu_relay\bin\ccu_relay.cmd restart
```

11 Лицензия

11.1 Получение

Все используемые контроллеры ССУ должны быть зарегистрированы в лицензии. Для получения файла лицензии необходимо сделать запрос на support@radsel.ru, в котором отправить список IMEI контроллеров.

Команды *license_reload* и *license_info* работают только после запуска *ccu_relay*.

11.2 Обновление

Путь к файлу лицензии задается параметром конфигурации *license_keyfile* (см. таблицу 2). Полученный файл лицензии необходимо разместить по данному пути и выполнить следующую команду.

Для Linux:

```
/opt/ccu_relay/bin/ccu_relay license_reload
```

Для Windows:

```
C:\ccu_relay\bin\ccu_relay.cmd license_reload
```

В случае успешного обновления в *info.log* будет сообщение: *License file loaded*.

В случае ошибки в *error.log* будет сообщение: *Failed loading license file*.

11.3 Информация

Данная команда выдаст список IMEI контроллеров, которые зарегистрированы в лицензии.

Для Linux:

```
/opt/ccu_relay/bin/ccu_relay license_info
```

Для Windows:

```
C:\ccu_relay\bin\ccu_relay.cmd license_info
```

12 HTTP API

12.1 Ключ доступа

Запрос должен содержать заголовок *Authorization: Bearer <key>*. Ключ доступа *<key>* задается параметром конфигурации *http_api_key* (см. пункт 5.1).

12.2 Типы запросов

Поддерживаются следующие типы запросов:

- GET — получение данных
- POST — передача данных

Формат обмена данными JSON. Запрос POST должен содержать заголовок *Content-Type: application/json*. Параметры запроса POST передаются в теле запроса.

12.3 Ошибки

В случае ошибки ответом на запрос будет объект с параметрами *error_code* и *error_description*:

error_code	error_description	Описание
1	"CCU not found"	Контроллер с данным IMEI не найден
2	"CCU disconnected"	Контроллер отключился
3	"Timeout"	Превышено время ожидания ответа
4	"Too many requests"	Слишком много запросов
5	"Internal error"	Внутренняя ошибка

Таблица 3: Ошибки HTTP API

Пример ответа:

```
{
  "error_code": 1,
  "error_description": "CCU not found"
}
```

12.4 URL для входа в CCU Shell

Запрос:

```
GET /api/ccu/<imei>/auth
```

Ответ: строка/URL (действителен в течение 5 минут)

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \
  https://<host>/api/ccu/<imei>/auth
```

Пример ответа:

```
"https://nohost/login/3093c8ce5c3c7baf7decd55546f3fcf93316ece42eec410d9ec
20bf4f1fe303fe0cf28bcd745bfe8b07eca9c4a9cc5a8219fbe674e7360f6d00d94728433
3e8784dabe863a5cefc78748206f51a859885b470a0f08ca7fd23def56c009e9913fcd604
2de954562031c7a71590f645bd17e028fabe5f97b8ea726a8f14d695e5db184a06c8476"
```

12.5 Контроллеры онлайн

Запрос:

```
GET /api/ccu/online
```

Ответ: массив строк/IMEI

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \  
https://<host>/api/ccu/online
```

Пример ответа:

```
["865609024314173", "869158009723061"]
```

12.6 Идентификация контроллера

Запрос:

```
GET /api/ccu/<imei>/id
```

Ответ:

Параметр	Значение	Описание
id_dev_type	строка	Тип контроллера
id_dev_mod	строка	Модификация контроллера
id_ext_board	строка	Плата расширения
id_hw	строка	Аппаратная версия
id_fw	строка	Версия прошивки
id_boot_loader	строка	Версия загрузчика
id_build_date	строка	Дата сборки прошивки
id_country	строка	Код страны
id_sn	строка	Серийный номер
id_imei	строка	IMEI
id_gsm	строка	GSM

Таблица 4: Идентификация контроллера

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \  
https://<host>/api/ccu/<imei>/id
```

Пример ответа:

```
{  
  "id_boot_loader": "04.00",  
  "id_build_date": "May 24 2019",  
  "id_country": "RUS",  
  "id_dev_mod": "S",  
  "id_dev_type": "CCU825",  
  "id_ext_board": "--",
```

```

    "id_fw": "02.20",
    "id_gsm": "M680_1232_LQS13001_V018",
    "id_hw": "14.01",
    "id_imei": "865609024314173",
    "id_sn": "0BE0FF18469C2DAEB0550953871E00F5"
}

```

12.7 Системная информация и режим охраны

Запрос:

```
GET /api/ccu/<imei>/sysinfo
```

Ответ:

Параметр	Значение	Описание
sysinfo	таблица 6	Системная информация
mode	таблица 9 и 10	Режим охраны

Таблица 5: Системная информация и режим охраны

Параметр	Значение	Описание
sysinfo_gsm_signal	[таблица 7, число]	Статус и уровень GSM сигнала [дБм]
sysinfo_ext_pwr	число/null	Внешнее питание [В]
sysinfo_bat	[таблица 8, число/null]	Статус и заряд батареи [%]
sysinfo_temp	число/null	Температура платы [°C]
sysinfo_case	true/false/null	Корпус открыт/закрыт
sysinfo_balance	число/null	Баланс SIM-карты

Таблица 6: Системная информация

Если *null*, то значение не определено. Например, для *sysinfo_ext_pwr* это означает, что внешнее питание отключено.

Статус	Описание
"gsm_signal_weak"	Низкий
"gsm_signal_average"	Средний
"gsm_signal_good"	Хороший
"gsm_signal_excellent"	Отличный

Таблица 7: Статус GSM сигнала

Статус	Описание
"bat_ok"	В норме
"bat_not_used"	Не использовалась
"bat_disconnected"	Отключена
"bat_discharge_level1"	Разряжена до 1 уровня
"bat_discharge_level2"	Разряжена до 2 уровня

Таблица 8: Статус батареи

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \
  https://<host>/api/ccu/<imei>/sysinfo
```

Пример ответа:

```
{
  "mode": "disarm",
  "sysinfo": {
    "sysinfo_balance": 3019.989990234375,
    "sysinfo_bat": [
      "bat_not_used",
      null
    ],
    "sysinfo_case": true,
    "sysinfo_ext_pwr": 15.199999809265137,
    "sysinfo_gsm_signal": [
      "gsm_signal_excellent",
      -75
    ],
    "sysinfo_temp": 31
  }
}
```

12.8 Режим охраны

12.8.1 Получение

Запрос:

```
GET /api/ccu/<imei>/mode
```

Ответ:

Если контроллер работает в однораздельном режиме, то режим охраны представлен строкой:

Режим охраны	Описание
"disarm"	НАБЛЮДЕНИЕ
"arm"	ОХРАНА
"protect"	ЗАЩИТА

Таблица 9: Режим охраны

Если контроллер работает в многораздельном режиме, то режим охраны разделов представлен объектом:

Номер раздела	Значение	Описание
число в строковом представлении	таблица 9	Режим охраны раздела

Таблица 10: Режим охраны разделов

Количество разделов от 1 до 4. Если в конфигурации контроллера ни один раздел не имеет связанных входов, то объект *mode* будет пустым.

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \
  https://<host>/api/ccu/<imei>/mode
```

Пример ответа:

```
"disarm"
```

12.8.2 Изменение

Запрос:

```
POST /api/ccu/<imei>/mode
```

Параметр: см. таблицу 9 и 10

Ответ: см. таблицу 9 и 10

Пример запроса curl:

```
curl -k -X POST -d '"arm"' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <key>' \
  https://<host>/api/ccu/<imei>/mode
```

Пример ответа:

```
"arm"
```

12.9 Входы

Запрос:

```
GET /api/ccu/<imei>/inputs
```

Ответ:

Номер входа	Значение	Описание
число в строковом представлении	таблица 12	Состояние входа

Таблица 11: Состояние входов

Значение	Описание
false	Дискретный вход ПАССИВЕН
true	Дискретный вход АКТИВЕН
[false, число]	Аналоговый вход ПАССИВЕН, числовое значение
[true, число]	Аналоговый вход АКТИВЕН, числовое значение
[_ , null]	Показания недостоверны

Таблица 12: Состояние входа

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \  
https://<host>/api/ccu/<imei>/inputs
```

Пример ответа:

```
{  
  "1": [  
    false,  
    2099.0  
  ],  
  "2": true,  
  "3": [  
    true,  
    52.368743896484375  
  ],  
  "8": false  
}
```

12.10 Границы входов

12.10.1 Типы тревожных диапазонов

Тип	Количество границ
Низкий или высокий	2
Низкий	1
Средний	2
Высокий	1
Низкий гистерезисный	2
Высокий гистерезисный	2

Таблица 13: Типы тревожных диапазонов

12.10.2 Получение

Запрос:

```
GET /api/ccu/<imei>/inputs/limits
```

Ответ:

Номер входа	Значение	Описание
число в строковом представлении	[]	Для дискретного входа
	[число]	Для аналогового/RTD входа с одной границей
	[число, число]	Для аналогового/RTD входа с двумя границами

Таблица 14: Границы входов

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \
      https://<host>/api/ccu/<imei>/inputs/limits
```

Пример ответа:

```
{
  "1": [
    1430.0
  ],
  "2": [],
  "3": [
    30.097679138183594,
```

```
    70.4884033203125
  ],
  "8": []
}
```

12.10.3 Изменение

Запрос:

```
POST /api/ccu/<imei>/inputs/limits
```

Параметр: см. таблицу 14

Ответ: см. таблицу 14

Пример запроса curl:

```
curl -k -X POST -d '{"1": [1000.0], "3": [20, 55.5]}' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <key>' \
  https://<host>/api/ccu/<imei>/inputs/limits
```

Пример ответа:

```
{
  "1": [
    1000.0
  ],
  "3": [
    19.98779296875,
    55.4945068359375
  ]
}
```

12.11 Выходы

12.11.1 Типы выходов

Номер выхода	Тип
1	Реле 1
2	Реле 2
3	Выход 1
4	Выход 2
5	Выход 3
6	Выход 4
7	Выход 5

Таблица 15: Типы выходов

12.11.2 Получение

Запрос:

```
GET /api/ccu/<imei>/outputs
```

Ответ:

Номер выхода	Значение	Описание
число в строковом представлении	0	Выключен
	1	Включен

Таблица 16: Состояние выходов

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \  
https://<host>/api/ccu/<imei>/outputs
```

Пример ответа:

```
{  
  "1": 1,  
  "2": 1,  
  "3": 0,  
  "4": 0,  
  "5": 0,  
  "6": 0  
}
```

12.11.3 Управление

Запрос:

POST /api/ccu/<imei>/outputs

Параметр:

Номер выхода	Значение	Описание
число в строковом представлении	0	Выключить
	1	Включить
	2	Сценарий 1
	3	Сценарий 2
	4	Сценарий 3
	5	Сценарий 4
	6	Сценарий 5
	7	Сценарий 6
	8	Сценарий 7
	9	Сценарий 8
	10	Сценарий 9
	11	Сценарий 10
	12	Сценарий 11
	13	Сценарий 12
	14	Сценарий 13
15	Сценарий 14	

Таблица 17: Управление выходами

Ответ: см. таблицу 16

Пример запроса curl:

```
curl -k -X POST -d '{"1":0,"6":1}' \  
  -H 'Content-Type: application/json' \  
  -H 'Authorization: Bearer <key>' \  
  https://<host>/api/ccu/<imei>/outputs
```

Пример ответа:

```
{  
  "1": 0,  
  "6": 1  
}
```

12.12 Профили

Запрос:

POST /api/ccu/<imei>/profile

Параметр: число (номер профиля для применения)

Ответ: число (номер примененного профиля)/null (если профиль запрещен)

Пример запроса curl:

```
curl -k -X POST -d 1 \  
  -H 'Content-Type: application/json' \  
  -H 'Authorization: Bearer <key>' \  
  https://<host>/api/ccu/<imei>/profile
```

Пример ответа:

1

12.13 События

12.13.1 Источники событий

В конфигурации контроллера в нужном соединении необходимо выбрать источники событий.

Оповещение через CCU Relay TCP/IP

Разрешено

Параметры сервера

Тип адреса: IP адрес

IP адрес: 1.2.3.4

TCP порт: 30000

Снять данные с передачи, если не переданы за: 11 мин

Передавать сообщения от следующих источников

In1	In2	In3	In4	In5	In6	In7	In8
--	--	--	--	--	--	--	--
In9	In10	In11	In12	In13	In14	In15	In16
--	--	--	--	--	--	--	--

Режим охраны

ОХРАНА

ЗАЩИТА

НАБЛЮДЕНИЕ

Системные события

Тестовые и информационные сообщения

Применение профиля

Рис. 9: Выбор источников событий

12.13.2 Получение

Для получения событий используется технология Server-Sent Events².

Запрос:

GET /api/ccu/events

Ответ:

Параметр	Значение	Описание
imei	строка	IMEI контроллера (опционально)
event	таблица 19	Событие
ack	число	Код подтверждения события (опционально)

Таблица 18: Событийный объект

Если событийный объект содержит параметр *ack*, то событие должно быть подтверждено (см. пункт 12.13.3).

²https://ru.wikipedia.org/wiki/Server-sent_events

Значение	Описание
"keepalive"	Периодическое событие для поддержания соединения
"online"	Контроллер подключился
"offline"	Контроллер отключился
"event_power_on"	Внешнее питание включено
"event_power_off"	Внешнее питание отключено
["event_battery_low", число]	Батарея разряжена, значение [%]
["event_balance_low", число]	Баланс снизился, значение
["event_temp_low", число]	Температура платы упала, значение [°C]
["event_temp_normal", число]	Температура платы вернулась в допустимый диапазон, значение [°C]
["event_temp_high", число]	Температура платы поднялась, значение [°C]
"event_case_open"	Корпус контроллера открыт
["event_test", <i>системная информация</i> (таблица 6), <i>режим охраны</i> (таблица 9 и 10)]	Тестовое сообщение
["event_info", <i>состояние входов</i> (таблица 11), <i>состояние выходов</i> (таблица 16)]	Информационное сообщение
["event_disarm", <i>источник события</i> (таблица 20), число/null]	Переведен в режим НАБЛЮДЕНИЕ, номер раздела или null
["event_arm", <i>источник события</i> (таблица 20), число/null]	Переведен в режим ОХРАНА, номер раздела или null
["event_protect", <i>источник события</i> (таблица 20), число/null]	Переведен в режим ЗАЩИТА, номер раздела или null
["event_input", число, <i>состояние входа</i> (таблица 12)]	Вход АКТИВЕН/ПАССИВЕН, номер входа
["event_profile_applied", число]	Профиль применен, номер профиля

Значение	Описание
"event_device_on"	Контроллер включен
"event_device_restart"	Контроллер перезапущен
"event_firmware_upgrade"	Прошивка обновлена
["event_ext_runtime_error", число]	Ошибка выполнения программы EХТ, код ошибки

Таблица 19: События

Источник события	Описание
"source_button"	Кнопкой
["source_touch_memory", строка]	Ключом, идентификатор ключа или пользователь
"source_input"	С помощью входа
"source_scheduler"	Планировщиком задач по времени
["source_dtmf", строка]	Через голосовое меню, пользователь
["source_sms", строка]	С помощью SMS команды, пользователь
["source_csd", строка]	Через CSD соединение, пользователь
["source_call", строка]	С помощью вызова без установки соединения, пользователь
["source_ccu_shell", строка]	Через CCU Shell, пользователь
"source_modbus"	Через GuardTracker RS-485
["source_bot", строка]	С помощью бота, пользователь
"source_ext"	Через программу EХТ

Таблица 20: Источник события (изменение режима охраны)

Пример запроса curl:

```
curl -k -X GET -H 'Authorization: Bearer <key>' \
https://<host>/api/ccu/events
```

Примеры событий:

```
data:{"event":"online","imei":"865609024314173"}
data:{"ack":17696,"event":["event_arm",["source_ccu_shell","admin"]},
```

```
    null], "imei": "865609024314173"}
data: {"event": "keepalive"}
data: {"ack": 1792, "event": ["event_input", 1, [true, 5.155066967010498]],
      "imei": "865609024314173"}
```

12.13.3 Подтверждение

Запрос:

```
POST /api/ccu/<imei>/events/ack
```

Параметр: массив чисел

Ответ: нет

Пример запроса curl:

```
curl -k -X POST -d '[1,2]' \
      -H 'Content-Type: application/json' \
      -H 'Authorization: Bearer <key>' \
      https://<host>/api/ccu/<imei>/events/ack
```

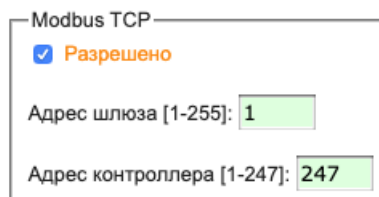
13 Modbus TCP

13.1 Настройка

Параметр конфигурации *modbus* (см. таблицу 2) позволяет настроить до 255 шлюзов Modbus TCP. Адрес шлюза задается числом от 1 до 255. Соответствующий адрес шлюза и адрес устройства также должны быть заданы в конфигурации контроллера на странице *Общие параметры соединений* (см. рисунок 10). Настройка более одного шлюза позволяет обойти ограничение Modbus (адрес устройства ограничен диапазоном от 1 до 247) и при необходимости подключить более 247 контроллеров. После запуска шлюза в *info.log* будет сообщение вида: *Modbus TCP Gateway {1,502} started*.

Листинг 3: Пример настройки двух шлюзов Modbus TCP

```
{modbus, [{1, 502}, {2, 503}]}
```



Modbus TCP

Разрешено

Адрес шлюза [1-255]: 1

Адрес контроллера [1-247]: 247

Рис. 10: Настройка Modbus TCP в конфигурации контроллера

13.2 Функции

Учитывая особенности работы GPRS, рекомендуется задавать в SCADA период опроса каждого контроллера не менее 10 секунд.

13.2.1 Read Coils (01)

Дискретные выходы контроллера отображаются на Coils (Регистры флагов).

Адрес	Тип
0	Реле 1
1	Реле 2
2	ВЫХОД 1
3	ВЫХОД 2
4	ВЫХОД 3
5	ВЫХОД 4
6	ВЫХОД 5

Таблица 21: Дискретные выходы

13.2.2 Read Discrete Inputs (02)

Дискретные входы контроллера отображаются на Discrete Inputs (Дискретные входы).

Адрес	Тип
0	Вход 1
1	Вход 2
2	Вход 3
3	Вход 4
4	Вход 5
5	Вход 6
6	Вход 7
7	Вход 8
8	Вход 9
9	Вход 10
10	Вход 11
11	Вход 12
12	Вход 13
13	Вход 14
14	Вход 15
15	Вход 16

Таблица 22: Дискретные входы

13.2.3 Read Input Registers (04)

Аналоговые входы контроллера отображаются на Input Registers (Регистры ввода). Значением входа является *float*³ размером 32 бита (4 байта). Один вход занимает два 16-битных регистра. Порядок байтов: от старшего к младшему (big-endian). Например, число 3.14 в данном формате имеет вид 4048F5C3₁₆.

Регистр 0		Регистр 1	
Старший байт	Младший байт	Старший байт	Младший байт
40	48	F5	C3

Таблица 23: Вход 1 — значение 3.14

Адрес	Тип
0	Вход 1
2	Вход 2
4	Вход 3
6	Вход 4
8	Вход 5
10	Вход 6
12	Вход 7
14	Вход 8
16	Вход 9
18	Вход 10
20	Вход 11
22	Вход 12
24	Вход 13
26	Вход 14
28	Вход 15
30	Вход 16

Таблица 24: Аналоговые входы

13.2.4 Write Single Coil (05)

См. пункт 13.2.1.

13.2.5 Write Multiple Coils (15)

См. пункт 13.2.1.

³Число одинарной точности, стандарт IEEE 754.

13.3 Ошибки

Код	Ошибка	Описание
01	Illegal Function	Функция не поддерживается
02	Illegal Data Address	Неверный адрес
03	Illegal Data Value	Неверное значение
04	Slave Device Failure	Ошибка передачи данных
06	Slave Device Busy	Слишком много запросов
10	Gateway Path Unavailable	В лицензии не включена поддержка Modbus для данного контроллера
11	Gateway Target Device Failed To Respond	Контроллер не в сети или превышено время ожидания ответа

Таблица 25: Ошибки Modbus