

# CCU relay

7 мая 2019 г.

## Содержание

<b>1</b>	<b>Установка</b>	<b>3</b>
<b>2</b>	<b>Обновление</b>	<b>3</b>
<b>3</b>	<b>Настройка</b>	<b>3</b>
3.1	Файл конфигурации . . . . .	3
3.2	Создание самоподписанного сертификата и ДН-параметров . . . . .	4
3.3	Настройка сервиса systemd . . . . .	4
<b>4</b>	<b>Запуск</b>	<b>5</b>
<b>5</b>	<b>Перезапуск</b>	<b>5</b>
<b>6</b>	<b>Остановка</b>	<b>5</b>
<b>7</b>	<b>Проверка работоспособности</b>	<b>6</b>
<b>8</b>	<b>Лицензия</b>	<b>6</b>
8.1	Получение . . . . .	6
8.2	Обновление . . . . .	6
8.3	Информация . . . . .	6
<b>9</b>	<b>HTTP API</b>	<b>7</b>
9.1	Ключ доступа . . . . .	7
9.2	Типы запросов . . . . .	7
9.3	Ошибки . . . . .	7
9.4	URL для входа в CCU shell . . . . .	7
9.5	Список IMEI контроллеров онлайн . . . . .	8
9.6	Идентификация контроллера . . . . .	8
9.7	Системная информация и режим охраны . . . . .	8
9.8	Режим охраны . . . . .	10
9.8.1	Получение . . . . .	10

9.8.2	Изменение . . . . .	10
9.9	Входы . . . . .	11
9.10	Границы входов . . . . .	11
9.10.1	Типы тревожных диапазонов . . . . .	11
9.10.2	Получение . . . . .	12
9.10.3	Изменение . . . . .	12
9.11	Выходы . . . . .	13
9.11.1	Типы выходов . . . . .	13
9.11.2	Получение . . . . .	13
9.11.3	Управление . . . . .	13
9.12	Профили . . . . .	14
9.13	События . . . . .	15
9.13.1	Получение . . . . .	15
9.13.2	Подтверждение . . . . .	18

## Внимание!

Для выполнения ряда приведенных команд может потребоваться запуск от пользователя *root* или использование *sudo*.

## 1 Установка

*ccu\_relay* поставляется в виде файла *ccu\_relay-<version>.tar.gz*. Поле *<version>* определяет версию.

---

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

---

## 2 Обновление

1. Если *ccu\_relay* запущен, то перед обновлением его необходимо остановить (см. пункт 6).
2. Сохранить директорию */opt/ccu\_relay/config*, если в ней были изменены файлы.
3. Распаковать новую версию вместо или поверх старой.
4. Восстановить директорию */opt/ccu\_relay/config*, если это необходимо.
5. Запустить *ccu\_relay* (см. пункт 4).

## 3 Настройка

### 3.1 Файл конфигурации

Листинг 1: */opt/ccu\_relay/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"}.
{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
license_keyfile	строка	Файл лицензии

Таблица 1: Параметры конфигурации

Имя хоста *host* используется при получении URL для входа в CCU shell (см. пункт 9.4).

TCP-порт *ccu\_port* используется для подключения контроллеров CCU. Это же значение должно быть задано в конфигурации контроллера.

TCP-порт *https\_port* используется для HTTP API.

TCP-порт со значением меньше 1024 считается привилегированным. Требуется запуск от root.

Для HTTPS требуется SSL-сертификат. В поставку входит самоподписанный сертификат. При необходимости данный сертификат можно пересоздать (см. пункт 3.2), 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 */opt/ccu\_relay/config*)
- абсолютный путь (должен начинаться с символа */*)

## 3.2 Создание самоподписанного сертификата и DH-параметров

---

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

---

## 3.3 Настройка сервиса systemd

Для автозапуска в современных системах Linux используется systemd. Необходимо создать unit-файл сервиса: скопировать файл */opt/ccu\_relay/config/systemd/ccu\_relay.service* в */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
```

---

Разрешить сервис:

---

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

---

После перезагрузки сервис должен запуститься автоматически. Для контроля см. пункт 7.

## 4 Запуск

---

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

---

В */opt/ccu\_relay/log/info.log* будет сообщение *System started*.

В случае ошибки в */opt/ccu\_relay/log/error.log* будет сообщение *System startup failed*.

Для контроля см. пункт 7.

## 5 Перезапуск

---

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

---

Перезапуск требуется после изменения файла конфигурации, если *ccu\_relay* был запущен.

## 6 Остановка

---

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

---

В */opt/ccu\_relay/log/info.log* будет сообщение *System stopped*.

## 7 Проверка работоспособности

Для проверки работоспособности могут использоваться команды:

- идентификатор процесса

---

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

---

- ping

---

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

---

- статус сервиса systemd (если выполнялась настройка сервиса, как в пункте 3.3)

---

```
systemctl status ccu_relay
```

---

## 8 Лицензия

### 8.1 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*.

### 8.2 Обновление

Путь к файлу лицензии задается параметром конфигурации *license\_keyfile*. Полученный файл лицензии необходимо разместить по данному пути и выполнить команду:

---

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

---

В */opt/ccu\_relay/log/info.log* будет сообщение *License file loaded*.

В случае ошибки в */opt/ccu\_relay/log/error.log* будет сообщение *Failed loading license file*.

### 8.3 Информация

---

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

---

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

## 9 HTTP API

### 9.1 Ключ доступа

Запрос должен содержать заголовок *Authorization: Bearer <key>*. Ключ доступа *<key>* задается параметром конфигурации *http\_api\_key* (см. пункт 3.1).

### 9.2 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.

### 9.3 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 2: Ошибки

### 9.4 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
```

---

## 9.5 Список IMEI контроллеров 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
```

---

## 9.6 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 3: Идентификация контроллера

Пример запроса curl:

---

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

---

## 9.7 Системная информация и режим охраны

Запрос:

---

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

---



Ответ:

Параметр	Значение	Описание
sysinfo	таблица 5	Системная информация
mode	таблица 8 и 9	Режим охраны

Таблица 4: Системная информация и режим охраны

Параметр	Значение	Описание
sysinfo_gsm_signal	[таблица 6, число]	Статус и уровень GSM сигнала [дБм]
sysinfo_ext_pwr	число/null	Внешнее питание [В]
sysinfo_bat	[таблица 7, 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 5: 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 6: 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 7: Статус батареи

Пример запроса curl:

---

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

---

## 9.8 Режим охраны

### 9.8.1 Получение

Запрос:

---

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

---

Ответ:

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

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

Таблица 8: Режим охраны

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

Номер раздела	Значение	Описание
число в строковом представлении	таблица 8	Режим охраны раздела

Таблица 9: 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
```

---

### 9.8.2 Изменение

Запрос:

---

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

---

Параметр: см. таблицу 8 и 9

Ответ: см. таблицу 8 и 9

Пример запроса curl:

---

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

---

## 9.9 Входы

Запрос:

---

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

---

Ответ:

Номер входа	Значение	Описание
число в строковом представлении	таблица 11	Состояние входа

Таблица 10: 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 11: Состояние входа

Пример запроса curl:

---

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

---

## 9.10 Границы входов

### 9.10.1 Типы тревожных диапазонов

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

Таблица 12: Типы тревожных диапазонов

### 9.10.2 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 13: Границы входов

Пример запроса curl:

---

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

---

### 9.10.3 Изменение

Запрос:

---

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

---

Параметр: см. таблицу 13

Ответ: см. таблицу 13

Пример запроса curl:

---

```
curl -k -X POST -d '{"1": [70.1], "2": [20, 22.5]}' \
-H 'Content-Type: application/json' \
-H 'Authorization: Bearer <key>' \
https://<host>/api/ccu/<imei>/inputs/limits
```

---

## 9.11 Выходы

### 9.11.1 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 14: Типы выходов

### 9.11.2 Получение

Запрос:

---

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

---

Ответ:

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

Таблица 15: Состояние выходов

Пример запроса curl:

---

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

---

### 9.11.3 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 16: Управление выходами

Ответ: см. таблицу 15

Пример запроса curl:

---

```
curl -k -X POST -d '{"1":1,"2":0}' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Bearer <key>' \
  https://<host>/api/ccu/<imei>/outputs
```

---

## 9.12 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
```

---

## 9.13 События

### 9.13.1 Получение

Для получения событий используется технология Server Sent Events.

Запрос:

---

GET /api/ccu/events

---

Ответ:

Параметр	Значение	Описание
imei	строка	IMEI контроллера (опционально)
event	таблица 18	Событие
ack	число	Код подтверждения события (опционально)

Таблица 17: Событийный объект

Если событийный объект содержит параметр *ack*, то событие должно быть подтверждено (см. пункт 9.13.2).

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

Значение	Описание
["event_test", <i>системная информация</i> (таблица 5), <i>режим охраны</i> (таблица 8 и 9)]	Тестовое сообщение
["event_info", <i>состояние входов</i> (таблица 10), <i>состояние выходов</i> (таблица 15)]	Информационное сообщение
["event_disarm", <i>источник события</i> (таблица 19), число/null]	Переведен в режим НАБЛЮДЕНИЕ, номер раздела или null
["event_arm", <i>источник события</i> (таблица 19), число/null]	Переведен в режим ОХРАНА, номер раздела или null
["event_protect", <i>источник события</i> (таблица 19), число/null]	Переведен в режим ЗАЩИТА, номер раздела или null
["event_input", число, <i>состояние входа</i> (таблица 11)]	Вход АКТИВЕН/ПАССИВЕН, номер входа
["event_profile_applied", число]	Профиль применен, номер профиля
"event_device_on"	Контроллер включен
"event_device_restart"	Контроллер перезапущен
"event_firmware_upgrade"	Прошивка обновлена
["event_ext_runtime_error", число]	Ошибка выполнения программы EXT, код ошибки

Таблица 18: 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 Modbus
["source_bot", строка]	С помощью бота, пользователь
"source_ext"	Через программу EXT

Таблица 19: 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":"012345678901234"}
data:{"ack":17696,"event":["event_arm",["source_ccu_shell","admin"],
null],"imei":"012345678901234"}
data:{"event":"keepalive"}
data:{"ack":1792,"event":["event_input",1,[true,5.155066967010498]],
"imei":"012345678901234"}
```

### 9.13.2 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
```

---