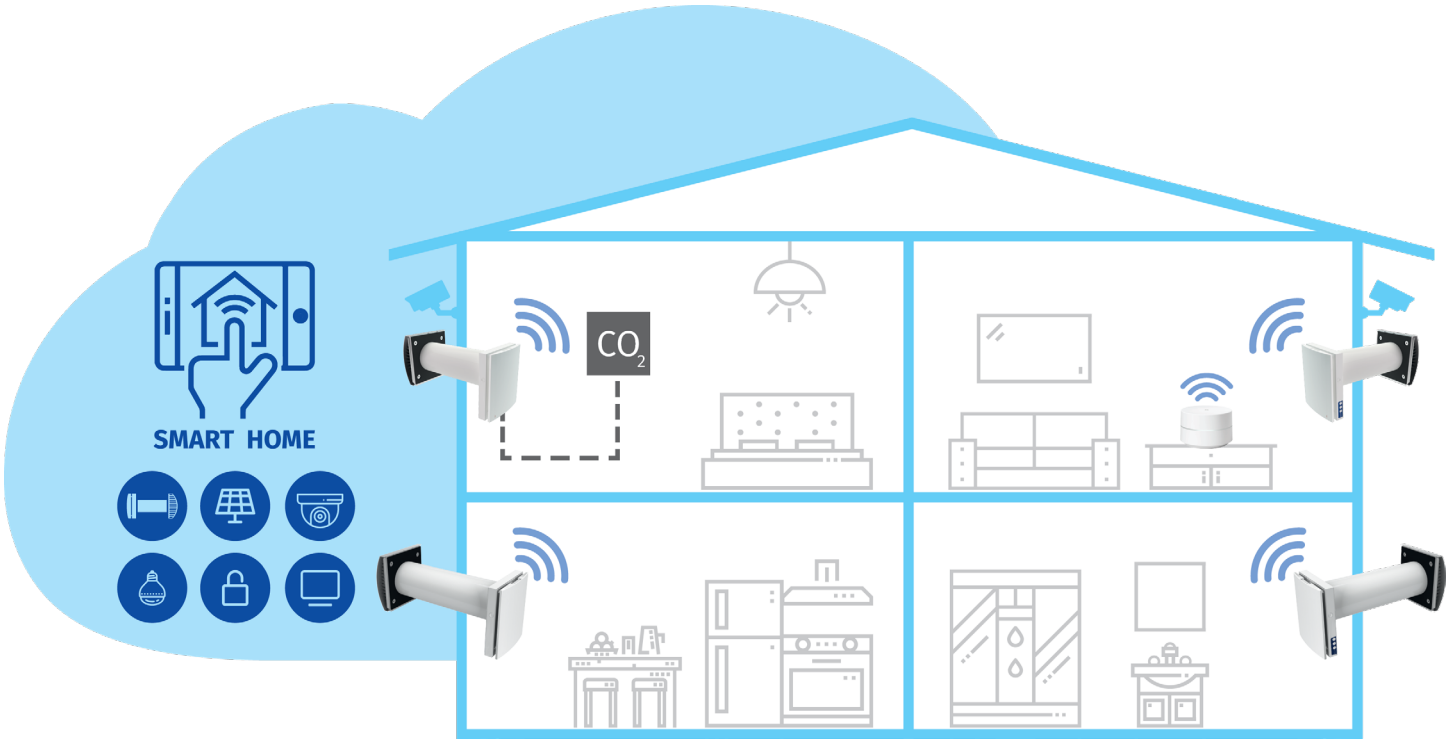


ПОДКЛЮЧЕНИЕ К СИСТЕМЕ «УМНЫЙ ДОМ»

VENTO

RU **РУКОВОДСТВО ПО ПОДКЛЮЧЕНИЮ**

BLAUBERG

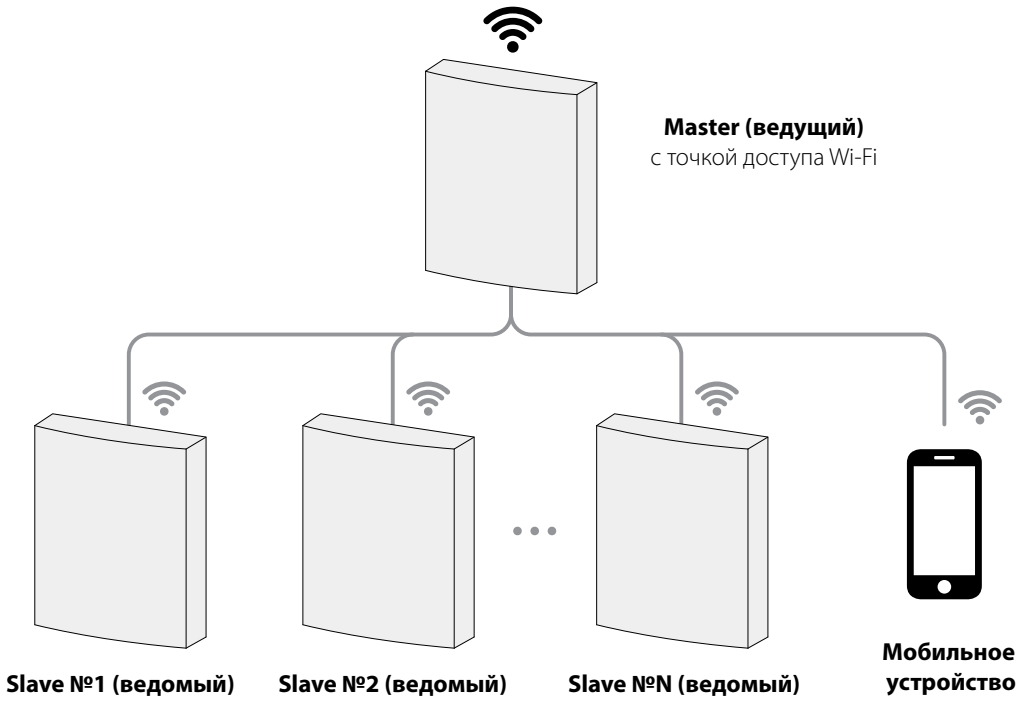
СОДЕРЖАНИЕ

Назначение 2
Параметры сети 3
Структура пакета 4
Примеры использования специальных команд в блоке DATA 5
Примеры полного пакета 6
Таблица параметров 7
Пример обработки пакетов на языке C 10

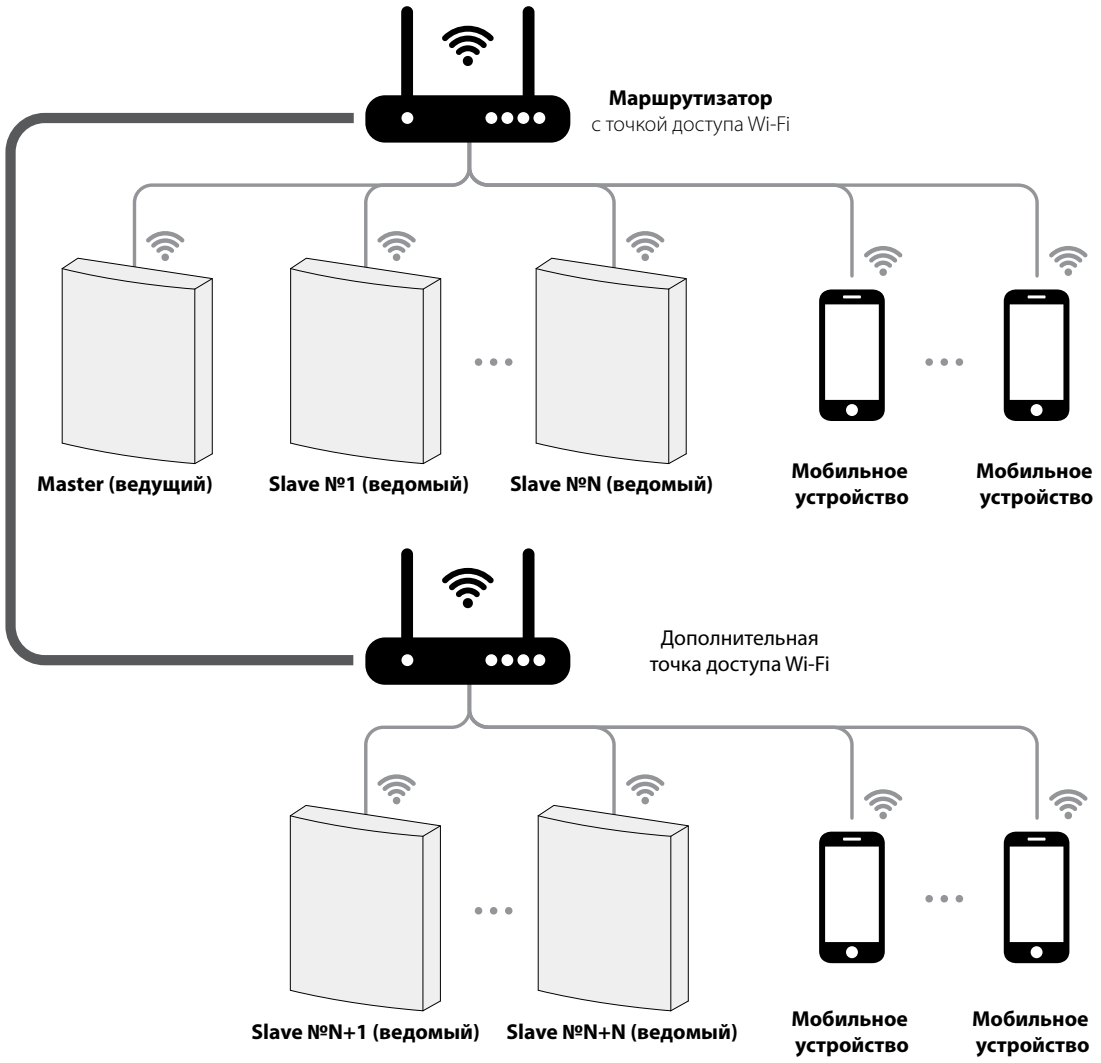
НАЗНАЧЕНИЕ

Настоящее руководство предназначено для подключения установок серии VENTO к системе «Умный дом».
Управление осуществляется только ведущими устройствами (Master). С помощью Wi-Fi к ним подсоединяются ведомые устройства (Slave), телефоны и система «Умный дом». Управление ведомыми устройствами (Slave) осуществляется командами от ведущих устройств (Master).

Существует две схемы беспроводного подключения:
1. Ведущее устройство со своей точкой доступа Wi-Fi, к которой можно подключить максимум восемь устройств. Если к точке доступа ведущего устройства подключить восемь ведомых устройств, то телефон или система «Умный дом» не сможет к ней подключиться!



2. Ведущие устройства, ведомые устройства, телефоны и система «Умный дом» подключаются к Wi-Fi точке доступа маршрутизатора. В таком случае ограничение на максимальное количество подключаемых устройств к Wi-Fi зависит от возможностей маршрутизатора. Если технические характеристики маршрутизатора не позволяют подключить необходимое количество проветривателей, можно использовать дополнительную точку доступа Wi-Fi для подключения остальных проветривателей. Также можно подключить в сеть маршрутизатора несколько ведущих устройств (Master) для организации зонального управления.



Настройка подключения производится с помощью мобильного приложения в меню Подключение – > Настройка Wi-Fi (см. паспорт на изделие).

ПАРАМЕТРЫ СЕТИ

Обмен данными производится по транспортному протоколу UDP (поддерживается широковещание).

IP-адрес ведущего устройства:
- 192.168.4.1 – когда ведущее устройство работает без маршрутизатора (схема подключения №1);
- в случае подключения ведущего устройства к маршрутизатору (схема подключения №2) IP-адрес настраивается с помощью мобильного приложения (см. паспорт на изделие) и может быть задан статическим или динамическим (DHCP).

Порт ведущего устройства – 4000.

Максимальный размер пакета – 256 байт.

СТРУКТУРА ПАКЕТА

0xFD	0xFD	TYPE	SIZE ID	ID	SIZE PWD	PWD	FUNC	DATA	Chksum L	Chksum H

0xFD **0xFD** – признак начала пакета (2 байта).

TYPE – тип протокола (1 байт). Значение = 0x02.

SIZE ID – размер блока **ID** (1 байт). Значение = 0x10.

ID – ID-номер контроллера. Этот номер находится на наклейке (представлен в виде 16 char-символов), которая клеится на плату управления или на корпус изделия.

Также можно использовать в качестве ID-номера кодовое слово "DEFAULT_DEVICEID". Его можно применить:

- для управления, если ведущее устройство работает без маршрутизатора (схема подключения №1);
- для поиска ведущих устройств в сети, если используется маршрутизатор (схема подключения №2); при этом устройство будет отвечать только на два параметра: 0x007C и 0x00B9 (см. таблицу параметров).

SIZE PWD – размер блока **PWD** (1 байт). Возможные значения: от 0x00 до 0x08.

PWD – пароль устройства (допустимые символы: "0…9", "a…z", "A…Z"). Пароль по умолчанию –1111.
Этот пароль можно изменить с помощью мобильного приложения в меню **Подключение –> Дома –> Настройки** (см. паспорт на изделие).

FUNC – номер функции (1 байт). Определяет действие с данными и структуру блока **DATA:**
0x01 – чтение параметров;
0x02 – запись параметров. Контроллер не отправляет ответ о состоянии указанных параметров;
0x03 – запись параметров с последующим ответом контроллера о состоянии указанных параметров;
0x04 – инкремент параметров с последующим ответом контроллера о состоянии указанных параметров;
0x05 – декремент параметров с последующим ответом контроллера о состоянии указанных параметров;
0x06 – ответ контроллера на запрос (FUNC = 0x01,0x03,0x04,0x05).

DATA – блок данных. Состоит из номеров параметров и их значений:

если FUNC = 0x01 или 0x04 или 0x05:

P1	P2	Pn

если FUNC = 0x02 или 0x03 или 0x06:

P1	Value 1	P2	Value 2	Pn	Value n

Номера параметров (см. таблицу параметров) условно состоят из двух байт (старший байт виртуальный). По умолчанию старший байт каждого номера параметра в каждом новом пакете равен 0x00. Старший байт можно изменить в пределах одного пакета с помощью специальной команды **0xFF** (см. ниже).

P – младший байт номера параметра. Возможные значения: 0x00 – 0xFB. Значения 0xFC – 0xFF являются *специальными командами:*

0xFC – изменить номер функции (**FUNC**). Следующий байт должен быть новым номером функции от 0x01 до 0x05. Используется, чтобы организовать в одном пакете несколько функций с разными действиями;

0xFD – параметр не поддерживается контроллером. Следующий байт – младший байт неподдерживаемого параметра. Используется при ответе контроллера (**FUNC** = 0x06) на запрос чтения или записи несуществующего параметра;

0xFE – изменить размер значения параметра **Value** для одного следующего параметра. Следующим байтом должен быть новый размер параметра, за ним – младший байт номера параметра, а далее – само значение **Value**;

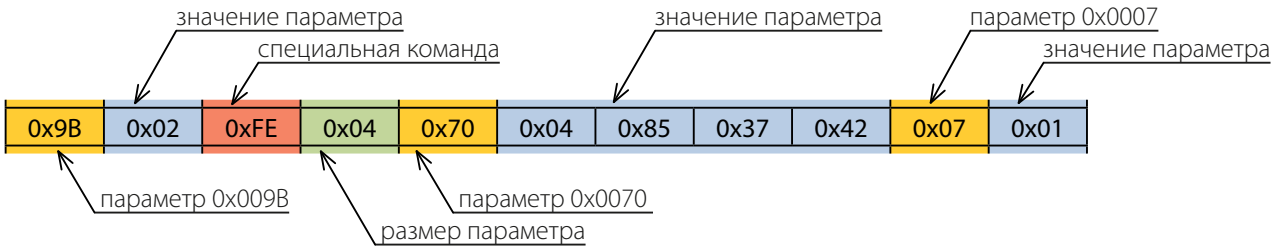
0xFF – изменить старший байт для номеров параметров в пределах одного пакета. Следующим байтом должен быть новый старший байт.

Value – значение параметра (по умолчанию 1 байт). Следование байт от младшего к старшему.

Chksum L **Chksum H** – контрольная сумма (2 байта). Она вычисляется как сумма байт, начиная с байта **TYPE** и заканчивая последним байтом блока **DATA**.
Chksum L – младший байт контрольной суммы.
Chksum H – старший байт контрольной суммы.

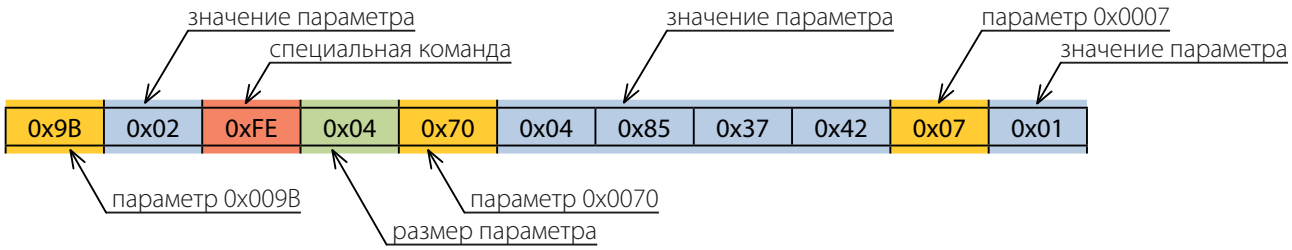
ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ СПЕЦИАЛЬНЫХ КОМАНД В БЛОКЕ DATA

Запрос на запись (FUNC = 0x03) параметров номер 0x009B, 0x0070, 0x0007



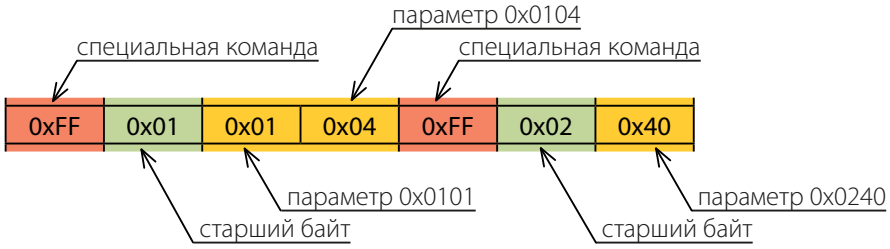
В запросе на запись следующее:
- Параметру 0x009B присвоить значение 0x02.
- Параметру 0x0070 присвоить значение 0x42378504. Размер значения – 4 байта, на это указывает специальная команда 0xFE + 0x04.
- Параметру 0x0007 присвоить значение 0x01.

Ответ контроллера (FUNC = 0x06) на запрос записи

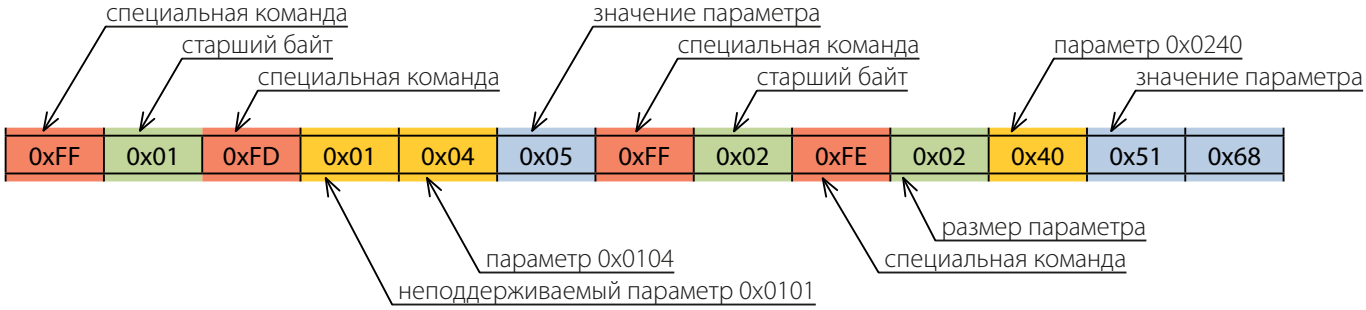


В ответе контроллера следующее:
- Параметр 0x009B имеет значение 0x02.
- Параметр 0x0070 имеет значение 0x42378504. Размер значения – 4 байта, на это указывает специальная команда 0xFE + 0x04.
- Параметр 0x0007 имеет значение 0x01.

Запрос на чтение (FUNC = 0x01) параметров номер 0x0101, 0x0104, 0x0240



Ответ контроллера (FUNC = 0x06) на запрос чтения

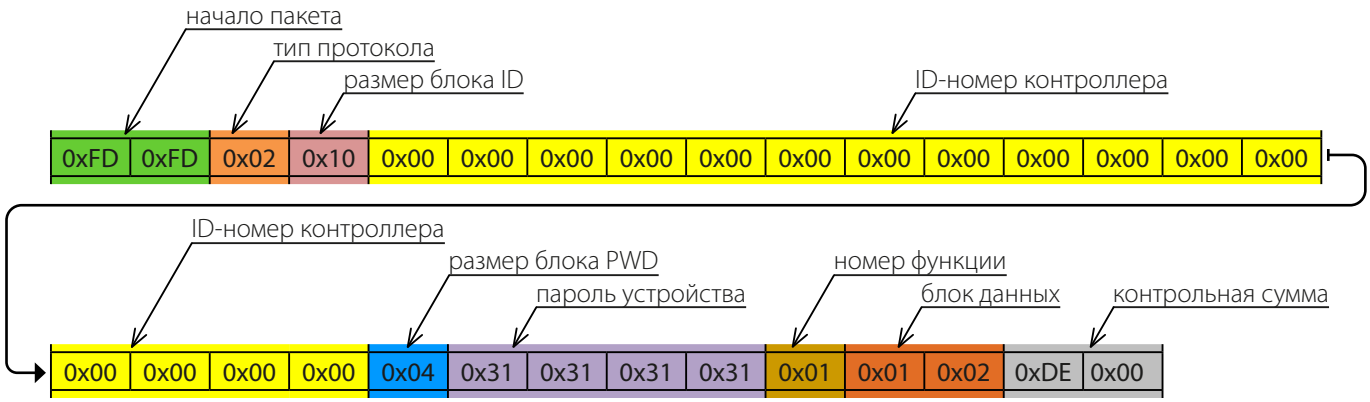


В ответе контроллера следующее:
- Параметр 0x0101 не поддерживается контроллером, на это указывает специальная команда 0xFD.
- Параметр 0x0104 имеет значение 0x05.
- Параметр 0x0240 имеет значение 0x6851. Размер значения – 2 байта, на это указывает специальная команда 0xFE + 0x02.

ПРИМЕРЫ ПОЛНОГО ПАКЕТА

Отправка пакета «Умный дом –> Контроллер»

В этом пакете запрос на чтение (FUNC = 0x01) параметров номер: 0x0001, 0x0002.

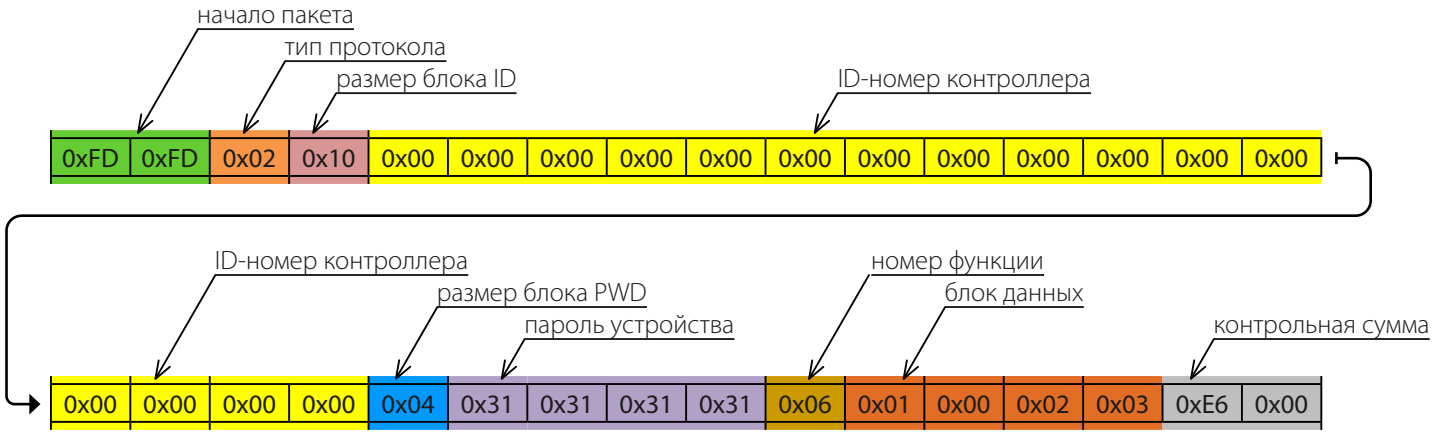


В запросе:

- Контрольная сумма: 0x00DE.

Отправка пакета «Контроллер –> Умный дом»

В этом пакете ответ контроллера (FUNC = 0x06) на запрос чтения.



В ответе контроллера:

- Параметр 0x0001 имеет значение 0x00.
- Параметр 0x0002 имеет значение 0x03.
- Контрольная сумма: 0x00E6.

ТАБЛИЦА ПАРАМЕТРОВ

Функции:	**R** – 0x01	**INC** – 0x04	**RW** – 0x03	**W** – 0x02	**DEC** – 0x05

Номер параметра, Dec./Hex.	Функции	Описание	Возможные значения	Размер, байт
1/0x0001	R/W/RW	Вкл/выкл установку	0 – выкл, 1 – вкл, 2 – инвертировать	1
2/0x0002	R/W/RW/INC/DEC	Номер скорости	1 – первая скорость, 2 – вторая скорость, 3 – третья скорость, 255 – режим ручной настройки скорости (см. 68 параметр)	1
6/0x0006	R	Состояние режима Boost	0 – выкл, 1 – вкл	1
7/0x0007	R/W/RW/INC/DEC	Режим «Таймер» (см. 770 и 771 параметр)	0 – выкл, 1 – ночной режим, 2 – режим «Вечеринка»	1
11/0x000B	R	Текущее время обратного отсчета режима «Таймер»	1-й байт – секунды (0…59), 2-й байт – минуты (0…59), 3-й байт – часы (0…23)	3
15/0x000F	R/W/RW	Активация датчика влажности	0 – выкл, 1 – вкл, 2 – инвертировать	1
20/0x0014	R/W/RW	Активация релейного датчика	0 – выкл, 1 – вкл, 2 – инвертировать	1
22/0x0016	R/W/RW	Активация датчика 0–10 В*	0 – выкл, 1 – вкл, 2 – инвертировать	1
25/0x0019	R/W/RW/INC/DEC	Уставка порога влажности	40…80 RH%	1
36/0x0024	R	Текущее напряжение батарейки RTC	0…5000 mV	2
37/0x0025	R	Текущая влажность	0…100 RH%	1
45/0x002D	R	Текущий уровень датчика 0–10 В*	0…100 %	1
50/0x0032	R	Текущее состояние релейного датчика	0 – выкл, 1 – вкл	1
58/0x003A	R/W/RW/INC/DEC	Скорость приточного вентилятора в режиме 1-й скорости**	10…255	1
59/0x003B	R/W/RW/INC/DEC	Скорость вытяжного вентилятора в режиме 1-й скорости**	10…255	1
60/0x003C	R/W/RW/INC/DEC	Скорость приточного вентилятора в режиме 2-й скорости**	10…255	1
61/0x003D	R/W/RW/INC/DEC	Скорость вытяжного вентилятора в режиме 2-й скорости**	10…255	1
62/0x003E	R/W/RW/INC/DEC	Скорость приточного вентилятора в режиме 3-й скорости**	10…255	1
63/0x003F	R/W/RW/INC/DEC	Скорость вытяжного вентилятора в режиме 3-й скорости**	10…255	1
68/0x0044	R/W/RW/INC/DEC	Скорость вентиляторов в режиме ручной настройки скорости	0…255	1
74/0x004A	R	Обороты вентилятора №1	0…5000 об/мин	2
75/0x004B	R	Обороты вентилятора №2	0…5000 об/мин	2
99/0x0063	R/W/RW/INC/DEC	Установка времени таймера замены фильтра**	70…365 дней	2
100/0x0064	R	Время обратного отсчета таймера до замены фильтра	1-й байт – минут (0…59), 2-й байт – часов (0…23), 3-й байт – дней (0…181)	3

Номер параметра, Dec./Hex.	Функции	Описание	Возможные значения	Размер, байт
101/0x0065	W	Сбросить время обратного отсчета таймера до замены фильтра	Любой байт	1
102/0x0066	R/W/RW/INC/DEC	Уставка задержки выключения Boost режима	0…60 минут	1
111/0x006F	R/W/RW	Время RTC	1-й байт – секунды RTC (0…59), 2-й байт – минуты RTC (0…59), 3-й байт – часы RTC (0…23)	3
112/0x0070	R/W/RW	Календарь RTC	1-й байт – число RTC (1…31), 2-й байт – день недели RTC (1…7), 3-й байт – месяц RTC (1…12), 4-й байт – год RTC (0…99)	4
114/0x0072	R/W/RW	Режим «Недельное расписание»	0 – выкл, 1 – вкл, 2 – инвертировать	1
119/0x0077	R/W/RW	Настройка расписания. В запросе на чтение необходимо воспользоваться специальной командой 0xFE и указать размер значения параметра 0x02 для выбора необходимого дня недели и номера периода времени. В запросе на запись и в ответе контроллера используются все 6 байт. Начало первого периода времени всегда 00:00, а начало каждого последующего периода является концом предыдущего. Конец последнего периода времени всегда 24:00	**1-й** байт – день недели: 0 – все дни (только запись), 1 – понедельник, 2 – вторник, 3 – среда, 4 – четверг, 5 – пятница, 6 – суббота, 7 – воскресенье, 8 – Пн…Пт (только запись), 9 – Сб…Вс (только запись) **2-й** байт – номер периода: 1…4 **3-й** байт – номер скорости: 0 – standby 1…3 **4-й** байт – зарезервирован: любой байт **5-й** байт – минуты конца периода: 0…59 **6-й** байт – часы конца периода: 0…23	6
124/0x007C	R	Поиск устройств в локальной сети, ID-номер	Текст ("0…9", "A…F")	16
125/0x007D	R/W/RW	Пароль устройства	Текст ("0…9", "a…z", "A…Z")	0–8
126/0x007E	R	Моточасы	1-й байт – минут (0…59), 2-й байт – часов (0…23), 3-й, 4-й байт – дней (0…65535)	4
128/0x0080	W	Сбросить аварии	Любой байт	1
131/0x0083	R	Индикатор наличия аварии/ предупреждения	0 – нет 1 – авария (имеет больший приоритет) 2 – предупреждение	1
133/0x0085	R/W/RW	Разрешение работы через облачный сервер	0 – выкл, 1 – вкл, 2 – инвертировать	1
134/0x0086	R	Версия и дата основной прошивки контроллера	1-й байт – версия прошивки (major) 2-й байт – версия прошивки (minor), 3-й байт – день, 4-й байт – месяц, 5-й, 6-й байт – год	6

Номер параметра, Dec./Hex.	Функции	Описание	Возможные значения	Размер, байт
135/0x0087	W	Восстановить до заводских настроек	Любой байт	1
136/0x0088	R	Индикатор замены фильтра	0 – фильтр не требует замены, 1 – замените фильтр	1
148/0x0094	R/W/RW/INC/DEC	Режим работы Wi-Fi	1 – Client, 2 – Access Point	1
149/0x0095	R/W/RW	Имя Wi-Fi в режиме Client	Текст	1…32
150/0x0096	R/W/RW	Пароль Wi-Fi	Текст	8…64
153/0x0099	R/W/RW	Тип шифрования данных Wi-Fi	48 – OPEN, 50 – WPA_PSK, 51 – WPA2_PSK, 52 – WPA_WPA2_PSK	1
154/0x009A	R/W/RW/INC/DEC	Частотный канал Wi-Fi	1…13	1
155/0x009B	R/W/RW	DHCP Wi-Fi модуля	0 – STATIC, 1 – DHCP, 2 – инвертировать	1
156/0x009C	R/W/RW	Заданный IP-адрес Wi-Fi модуля	**1-й** байт – 0…255, **2-й** байт – 0…255, **3-й** байт – 0…255, **4-й** байт – 0…255	4
157/0x009D	R/W/RW	Маска подсети WI-FI модуля	**1-й** байт – 0…255, **2-й** байт – 0…255, **3-й** байт – 0…255, **4-й** байт – 0…255	4
158/0x009E	R/W/RW	Основной шлюз WI-FI модуля	**1-й** байт – 0…255, **2-й** байт – 0…255, **3-й** байт – 0…255, **4-й** байт – 0…255	4
160/0x00A0	W	Применить новые параметры Wi-Fi и выйти из режима Setup Mode	Любой байт	1
162/0x00A2	W	Выйти из режима Setup Mode без применения новых параметров Wi-Fi	Любой байт	1
163/0x00A3	R	Текущий IP-адрес Wi-Fi модуля	**1-й** байт – 0…255, **2-й** байт – 0…255, **3-й** байт – 0…255, **4-й** байт – 0…255	4
183/0x00B7	R/W/RW/INC/DEC	Режим работы проветривателя	0 – проветривание, 1 – регенерация, 2 – приток	1
184/0x00B8	R/W/RW/INC/DEC	Уставка порога датчика 0–10 В*	5…100 %	1
185/0x00B9	R	Тип устройства	0…65535, 3 - VENTO Expert A50-1 W V.2, VENTO Expert A85-1 W V.2, VENTO Expert A100-1 W V.2 4 - VENTO Expert Duo A30-1 W V.2 5 - VENTO Expert A30 W V.2 14 - VENTO Style 26 - VENTO inHOME (старый) 27 - VENTO inHOME 100 28 - VENTO inHOME 160	2
252/0x00FC	**Специальные команды**			
253/0x00FD				
254/0x00FE				
255/0x00FF				
298/0x012A	R/W/RW	Восстановить настройки скорости вентиляторов для всех режимов до заводских	1 – выполнить	1
770/0x0302	R/W/RW	Уставка таймера для ночного режима	1-й байт – минуты (0…59), 2-й байт – часы (0…23)	2
771/0x0303	R/W/RW	Уставка таймера для режима «Вечеринка»	1-й байт – минуты (0…59), 2-й байт – часы (0…23)	2

Номер параметра, Dec./Hex.	Функции	Описание	Возможные значения	Размер, байт
772/0x0304	R	Состояние датчика влажности	0 – не превышает уставку, 1 – превышает уставку	1
773/0x0305	R	Состояние датчика 0–10 В*	0 – не превышает уставку, 1 – превышает уставку	1
810/0x032A	R/W/RW	Разрешение работы Boost режима в режиме пассивной вентиляции (только для ТФ Атмо)	0 - выкл, 1 - вкл, 2 - инвертировать	1
811/0x032B	R/W/RW	Режим пассивной вентиляции (только для ТФ Атмо). Имеет больший приоритет над параметром 183	0 - выкл, 1 - вкл, 2 - инвертировать	1

*Доступно для всех моделей кроме VENTO Expert A30 W V.2

**Доступно для VENTO Expert A50-1 W V.3

ПРИМЕР ОБРАБОТКИ ПАКЕТОВ НА ЯЗЫКЕ С

```
//================ Специальные команды ==================//
#define BGCP_CMD_PAGE                                   0xFF
#define BGCP_CMD_FUNC                                   0xFC
#define BGCP_CMD_SIZE                                   0xFE
#define BGCP_CMD_NOT_SUP                                0xFD
//=======================================================//

#define BGCP_FUNC_RESP                                  0x06

uint8_t receive_data[256];
uint16_t  receive_data_size;
uint8_t State_Power;
uint8_t State_Speed_mode;
char current_id[17] = "002D6E1B34565815"; // ID-номер контроллера

//********** Проверка контрольной суммы и начало пакета ***********//
uint8_t check_protocol(uint8_t *data, uint16_t size)
{
    uint16_t i, chksum1 = 0, chksum2 = 0;
    if((data[0] == 0xFD) && (data[1] == 0xFD))
    {
        for(i = 2; i <= size-3; i++)
            chksum1 += data[i];
        chksum2 = (uint16_t)(data[size-1] << 8) | (uint16_t)(data[size-2]);
        if(chksum1 == chksum2)
            return 1;
        else
            return 0;
    }
    else
        return 0;
}
//*************************************************************//


int main(void)
{
    ...

    if(check_protocol(receive_data, receive_data_size) == 1) // Контрольная сумма
    {
        if(receive_data[2] == 0x02) // Тип протокола
        {
            if(memcmp(&receive_data[4], current_id, receive_data[3]) == 0) // ID-номер
            {
                uint16_t jump_size = 0, page = 0, param, param_size, r_pos;
                uint8_t flag_check_func = 1, BGCP_func;
```

```
                r_pos = 4 + receive_data[3];
                r_pos += 1 + receive_data[r_pos]; // Место в массиве, где начинается блок FUNC
                //********************* FUNC и DATA *********************//
                for(; r_pos < receive_data_size - 2; r_pos++)
                {
                    //============ Специальные команды ============//
                    param_size = 1;
                    //=== новый номер функции
                    if((flag_check_func == 1) || (receive_data[r_pos] == BGCP_CMD_FUNC))
                    {
                        if(receive_data[r_pos] == BGCP_CMD_FUNC)
                           r_pos++;
                        flag_check_func = 0;
                        BGCP_func = receive_data[r_pos];
                        if(BGCP_func != BGCP_FUNC_RESP) // если номер функции не поддерживается
                           break;
                        continue;
                    }
                    //=== новое значение старшего байта для номеров параметров
                    else if(receive_data[r_pos] == BGCP_CMD_PAGE)
                    {
                       page = receive_data[++r_pos];
                       continue;
                    }
                    //=== новое значение размера параметра
                    else if(receive_data[r_pos] == BGCP_CMD_SIZE)
                    {
                        param_size = receive_data[++r_pos];
                        r_pos++;
                    }
                    //=== если параметр не поддерживается
                    else if(receive_data[r_pos] == BGCP_CMD_NOT_SUP)
                    {
                        r_pos++;
                        //******* обработка неподдерживаемых параметров ******//
                        param = (uint16_t)(page << 8) | (uint16_t)(receive_data[r_pos]);
                        switch(param)
                        {
                           case 0x0001:
                              break;
                           case 0x0002:
                              break;
                            ...
                        }
                        //************************//
                        continue;
                    }
                    jump_size = param_size;
                    //==================================//

                    //******* обработка поддерживаемых параметров ******//
                    param = (uint16_t)(page << 8) | (uint16_t)(receive_data[r_pos]);
                    switch(param)
                    {
                        case 0x0001:
                            State_Power = receive_data[r_pos+1];
                            break;
                        case 0x0002:
                            State_Speed_mode = receive_data[r_pos+1];
                            break;
                        ...
                    }
                    //************************//
                    r_pos += jump_size;
                }
                //*********************************************************//
            }
        }
    }
}
```

BLAUBERG

https://blaubergventilatoren.net/
B133-4-1RU-02