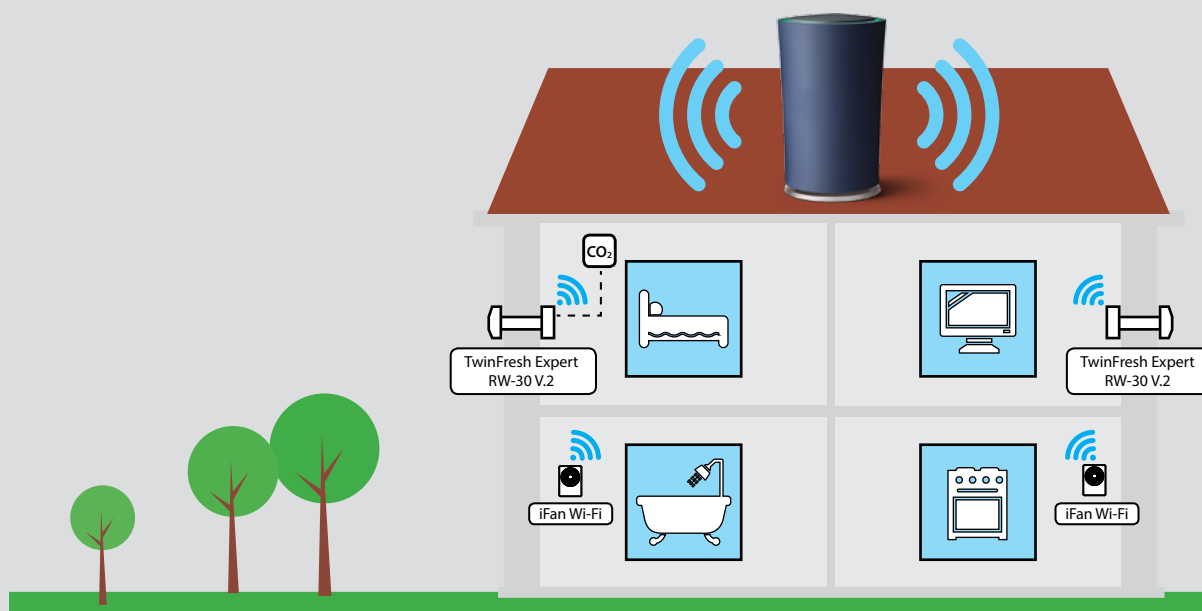


# Smart House



Підключення до системи «Розумний дім»

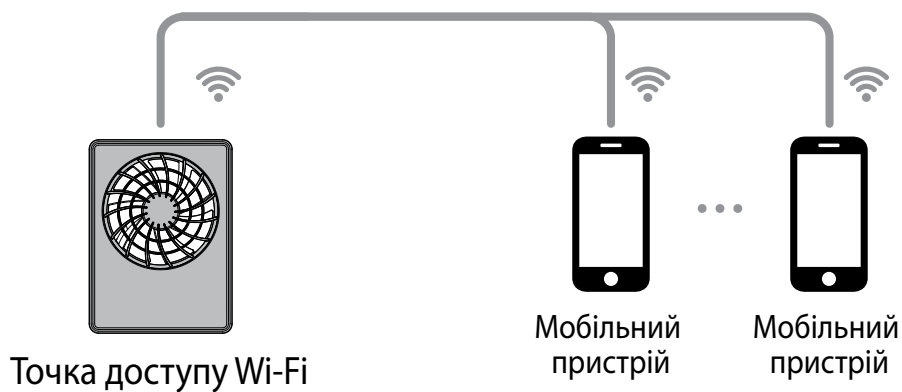
---

## ЗМІСТ

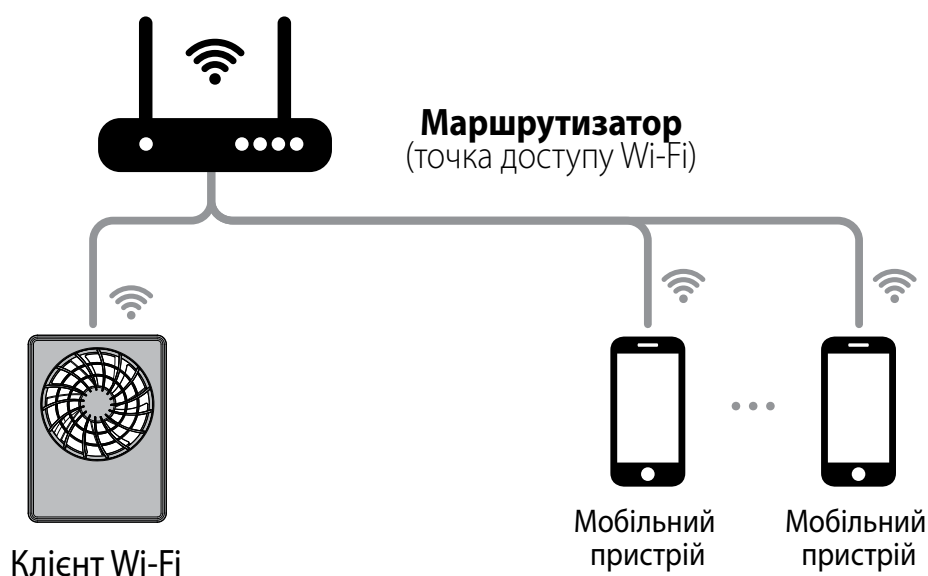
Підключення та налаштування .....	2
Параметри мережі.....	3
Структура пакета.....	4
Приклади використання спеціальних команд у блоці data .....	5
Приклади повного пакета.....	6
Таблиця параметрів .....	7
Приклад обробки пакетів мовою C.....	10

## ПІДКЛЮЧЕННЯ ТА НАЛАШТУВАННЯ

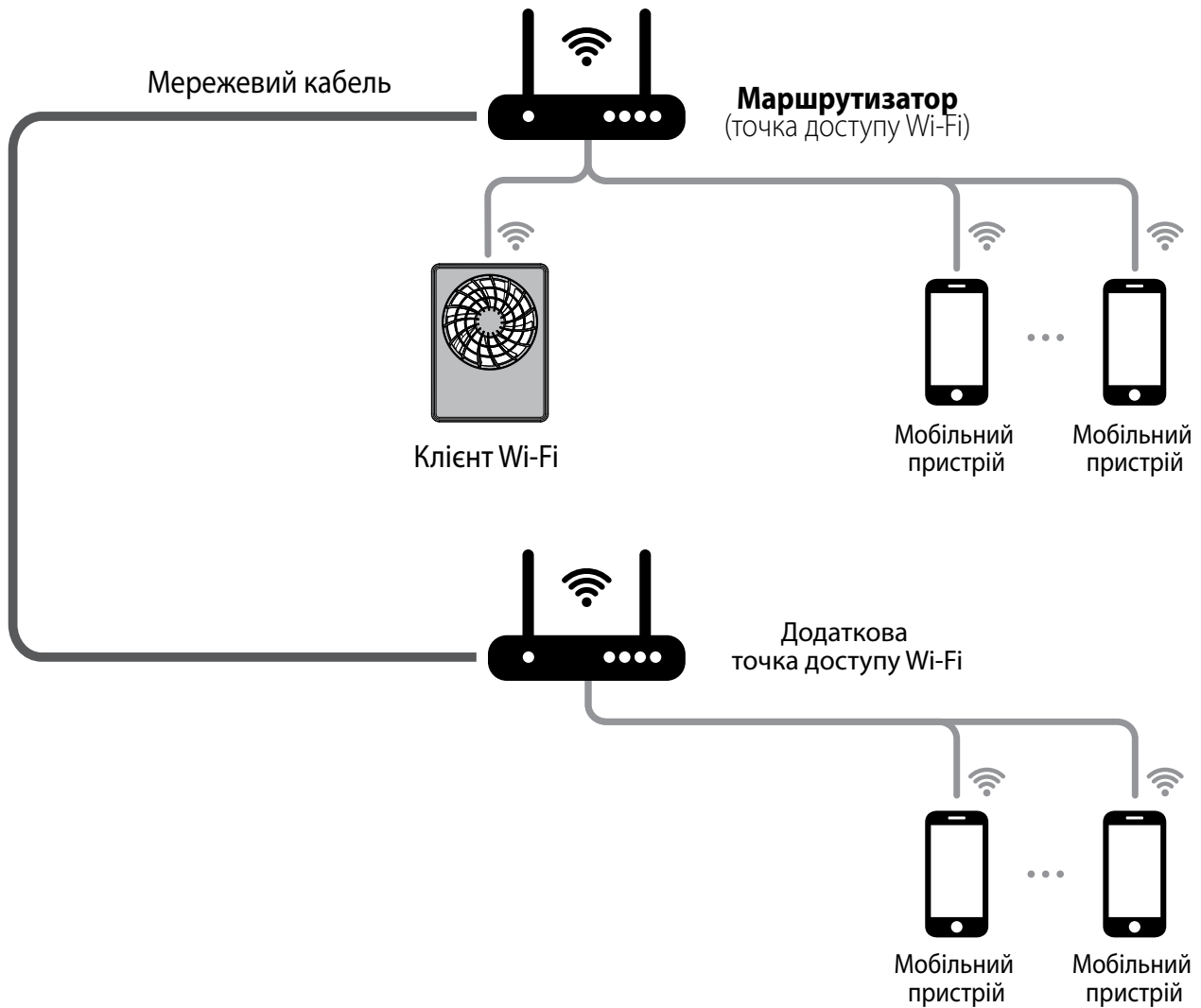
**Приклад 1:** схема прямого підключення вентилятора до системи BMS «Розумний дім» без використання маршрутизатора. Налаштуйте вентилятор на роботу Wi-Fi у режимі точки доступу (див. посібник користувача до вентилятора).  
Примітка: максимально можлива кількість пристроїв керування, які можна підключити, – вісім.



**Приклад 2:** схема підключення з використанням маршрутизатора з однією точкою доступу Wi-Fi. Вентилятор, телефони та система BMS «Розумний дім» підключаються до Wi-Fi точки доступу маршрутизатора.



**Приклад 3:** схема підключення системи BMS «Розумний дім» з використанням маршрутизатора, до якого підключено декілька точок доступу Wi-Fi.



## ПАРАМЕТРИ МЕРЕЖІ

Обмін даними проводиться за транспортним протоколом UDP (підтримується широкомовлення).

IP-адреса керівного пристрою:

- 192.168.4.1 – коли керівний пристрій працює без маршрутизатора (схема підключення №1);
- у разі підключення керівного пристрою до маршрутизатора (схема підключення №2) IP-адреса налаштовується за допомогою мобільного додатка (див. паспорт на виріб) і може бути задана статичною або динамічною (DHCP).

Порт керівного пристрою – 4000.

Максимальний розмір пакета – 256 байт.

## СТРУКТУРА ПАКЕТА

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

**0xFD** **0xFD** – ознака початку пакета (2 байти).

**TYPE** – тип протоколу (1 байт). Значення = 0x02.

**SIZE ID** – розмір блока **ID** (1 байт). Значення = 0x10.

**ID** – ID-номер контролера. Цей номер знаходиться на наліпці (представлений у вигляді 16 char-символів), яка клеїться на плату керування або на корпус виробу.

Також можна використовувати у якості ID-номера кодове слово DEFAULT\_DEVICEID. Його можна застосувати:

- для керування, якщо керівний пристрій працює без маршрутизатора (схема підключення №1);
- для пошуку керівних пристроїв у мережі, якщо використовується маршрутизатор (схема підключення №2); водночас пристрій буде відповідати тільки на два параметри: 0x007C та 0x00B9 (див. таблицю параметрів).

**SIZE PWD** – розмір блока **PWD** (1 байт). Можливі значення: від 0x00 до 0x08.

**PWD** – пароль пристрою (допустимі символи: 0...9, a...z, A...Z). Пароль за замовчуванням – 1111.

Цей пароль можна змінити за допомогою мобільного додатка у меню **Підключення** → **Вдома** → **Налаштування** (див. паспорт на виріб).

**FUNC** – номер функції (1 байт). Визначає дію з даними та структуру блока **DATA**:

- 0x01 – читання параметрів;
- 0x02 – запис параметрів. Контролер не визначає відповідь про стан вказаних параметрів;
- 0x03 – запис параметрів з подальшою відповіддю контролера про стан вказаних параметрів;
- 0x04 – інкремент параметрів з подальшою відповіддю контролера про стан вказаних параметрів;
- 0x05 – декремент параметрів з подальшою відповіддю контролера про стан вказаних параметрів;
- 0x06 – відповідь контролера на запит (FUNC = 0x01, 0x03, 0x04, 0x05).

**DATA** – блок даних. Складається з номерів параметрів та їх значень:

якщо *FUNC* = 0x01 або 0x04 або 0x05:

<b>P1</b>	<b>P2</b>	<b>Pn</b>
-----------	-----------	-----------

якщо *FUNC* = 0x02 або 0x03 або 0x06:

<b>P1</b>	<b>Value 1</b>	<b>P2</b>	<b>Value 2</b>	<b>Pn</b>	<b>Value n</b>
-----------	----------------	-----------	----------------	-----------	----------------

Номери параметрів (див. таблицю параметрів) умовно складаються з двох байт (старший байт віртуальний). За замовчуванням старший байт кожного номера параметра у кожному новому пакеті дорівнює 0x00. Старший байт можна змінити у межах одного пакета за допомогою спеціальної команди 0xFF (див. нижче).

**P** – молодший байт номера параметра. Можливі значення: 0x00 – 0xFB. Значення 0xFC – 0xFF є спеціальними командами:

- 0xFC** – змінити номер функції (**FUNC**). Наступний байт повинен бути новим номером функції від 0x01 до 0x05. Використовується, щоб організувати в одному пакеті декілька функцій з різними діями;
- 0xFD** – параметр не підтримується контролером. Наступний байт – молодший байт непідтримуваного параметра. Використовується під час відповіді контролера (**FUNC** = 0x06) на запит читання або записи неіснуючого параметра;
- 0xFE** – змінити розмір значення параметра **Value** для одного наступного параметра. Наступним байтом повинен бути новий розмір параметра, за ним – молодший байт номера параметра, а далі – саме значення **Value**;
- 0xFF** – змінити старший байт для номерів параметрів у межах одного пакета. Наступним байтом повинен бути новий старший байт.

**Value** – значення параметра (за замовчуванням – 1 байт). Слідування байтів – від молодшого до старшого.

**Chksum L** **Chksum H** – контрольна сума (2 байти). Вона обчислюється як сума байтів, починаючи з байта **TYPE**

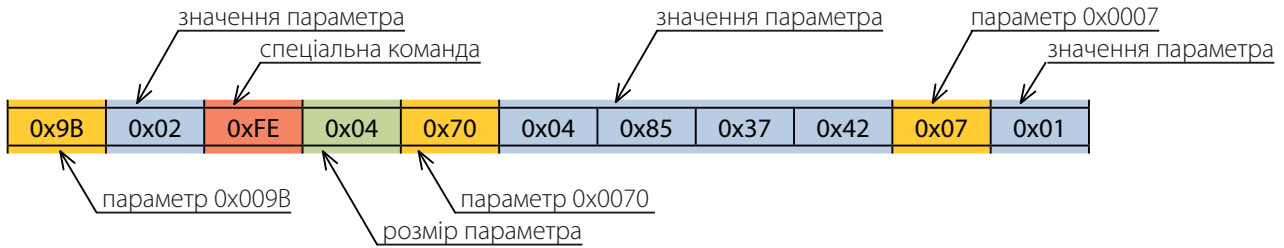
та закінчуючи останнім байтом блока **DATA**.

**Chksum L** – молодший байт контрольної суми.

**Chksum H** – старший байт контрольної суми.

## ПРИКЛАДИ ВИКОРИСТАННЯ СПЕЦІАЛЬНИХ КОМАНД У БЛОЦІ DATA

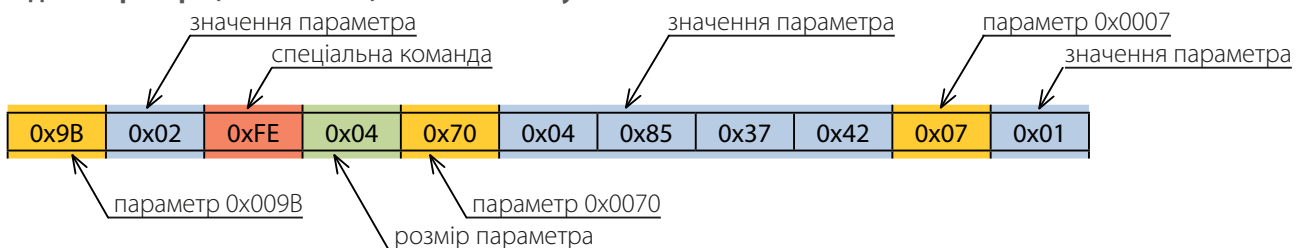
### Запит на запис (FUNC = 0x03) параметрів номер 0x009B, 0x0070, 0x0007



У запиті на запис наступне:

- Параметру 0x009B присвоїти значення 0x02.
- Параметру 0x0070 присвоїти значення 0x42378504. Розмір значення – 4 байти, на це вказує спеціальна команда 0xFE + 0x04.
- Параметру 0x0007 присвоїти значення 0x01.

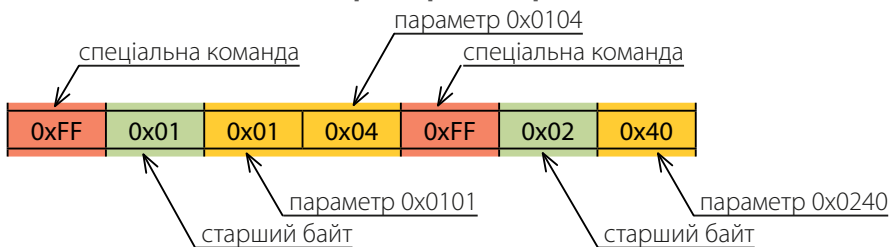
### Відповідь контролера (FUNC = 0x06) на запит запису



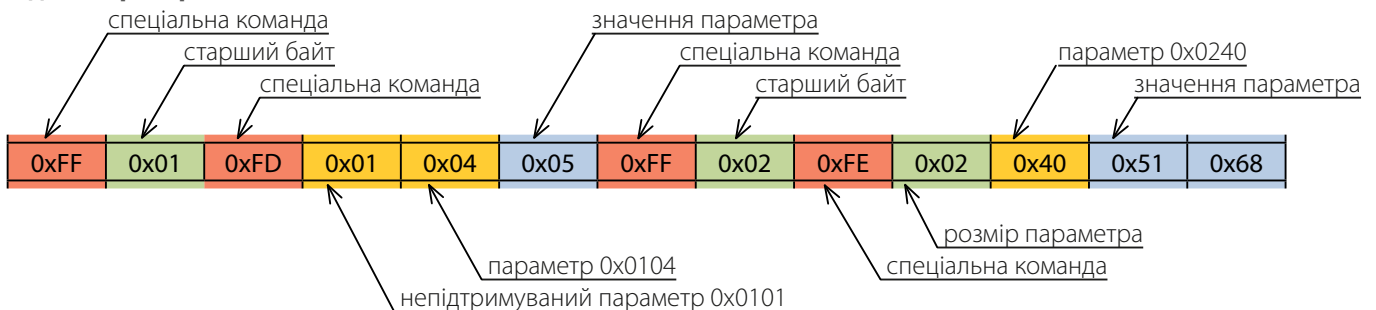
У відповіді контролера наступне:

- Параметр 0x009B має значення 0x02.
- Параметр 0x0070 має значення 0x42378504. Розмір значення – 4 байти, на це вказує спеціальна команда 0xFE + 0x04.
- Параметр 0x0007 має значення 0x01.

### Запит на читання (FUNC = 0x01) параметрів номер 0x0101, 0x0104, 0x0240



### Відповідь контролера (FUNC = 0x06) на запит читання



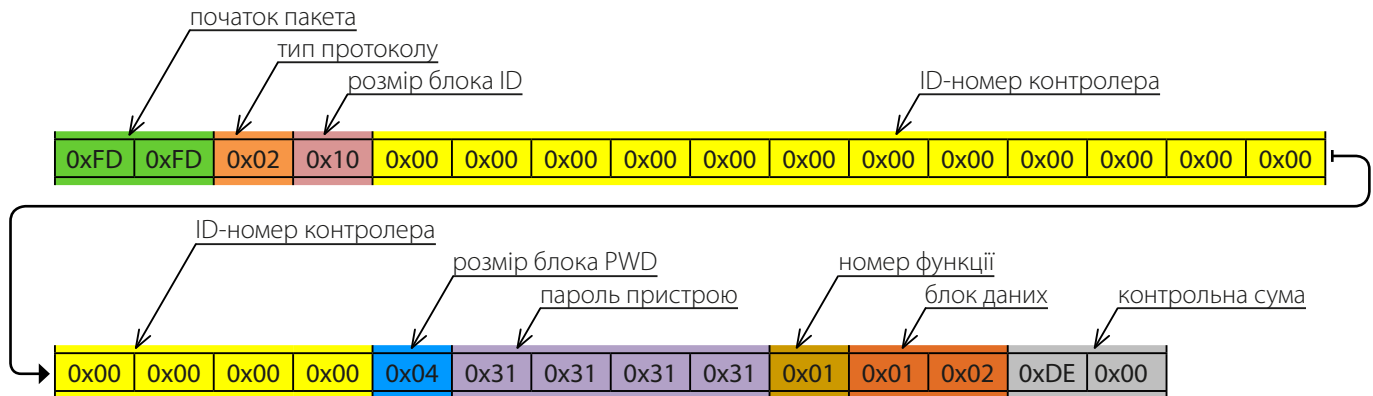
У відповіді контролера наступне:

- Параметр 0x0101 не підтримується контролером, на це вказує спеціальна команда 0xFD.
- Параметр 0x0104 має значення 0x05.
- Параметр 0x0240 має значення 0x6851. Розмір значення – 2 байти, на це вказує спеціальна команда 0xFE + 0x02.

## ПРИКЛАДИ ПОВНОГО ПАКЕТА

### Надсилання пакета «Розумний дім → Контролер»

У цьому пакеті запит на читання (FUNC = 0x01) параметрів номер: 0x0001, 0x0002.

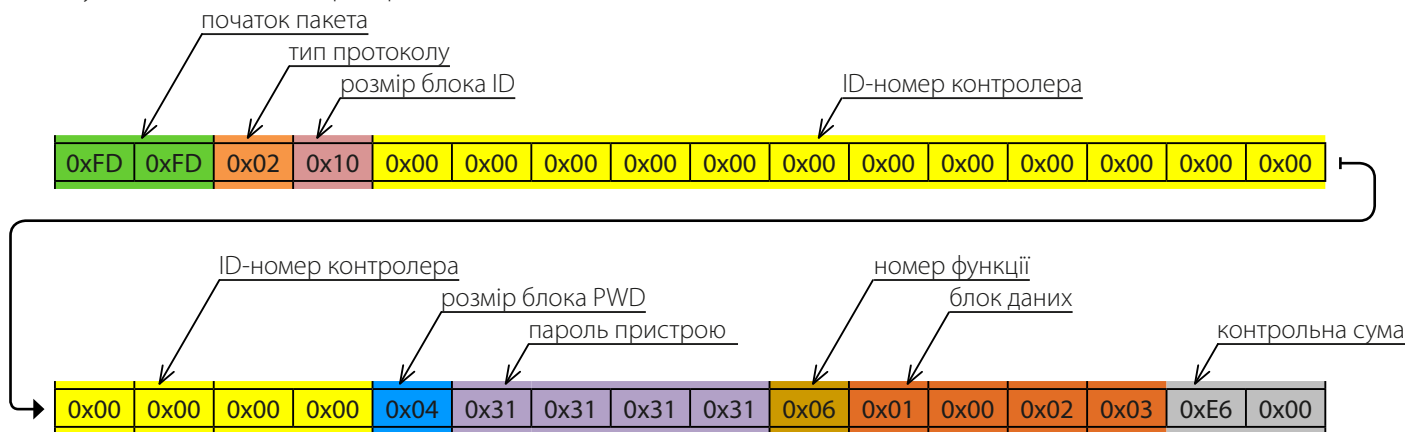


У запиті:

- Контрольна сума: 0x00DE.

### Надсилання пакета «Контролер → Розумний дім»

У цьому пакеті відповідь контролера (FUNC = 0x06) на запит читання.



У відповіді контролера:

- Параметр 0x0001 має значення 0x00.
- Параметр 0x0002 має значення 0x03.
- Контрольна сума: 0x00E6.

## ТАБЛИЦЯ ПАРАМЕТРІВ

### Функції:

R – 0x01

INC – 0x04

RW – 0x03

W – 0x02

DEC – 0x05

Номер параметра, Дес./Hex.	Функції	Опис	Можливі значення	Розмір, байт
1/0x0001	R/W/RW	Увімк./вимк. вентилятор	0 — вимк 1 — увімк 2 — інвертувати	1
2/0x0002	R	Стан батарейки	0 — розряджена (відсутня) 1 — заряд у нормі	1
3/0x0003	R/W/RW	Вибір режиму 24 години	0 — вимк 1 — увімк 2 — інвертувати	1
4/0x0004	R	Поточне значення швидкості вентилятора (об/хв)	0...6000 RPM	2
5/0x0005	R/W/RW	Увімк./Вимк. режим BOOST	0 — вимк 1 — увімк 2 — інвертувати	1
6/0x0006	R	Поточне значення часу відліку таймера BOOST у секундах	0...86400 секунд	3
7/0x0007	R	Поточний статус вбудованого таймера	0 — вимк 1 — увімк	1
8/0x0008	R	Поточний статус роботи вентилятора за датчиком вологості	0 — вимк 1 — увімк	1
10/0x000A	R	Поточний статус роботи вентилятора за датчиком температури	0 — вимк 1 — увімк	1
11/0x000B	R	Поточний статус роботи вентилятора за датчиком руху	0 — вимк 1 — увімк	1
12/0x000C	R	Поточний статус роботи вентилятора за сигналом від зовнішнього вимикача	0 — вимк 1 — увімк	1
13/0x000D	R	Поточний статус роботи вентилятора у режимі інтервального провітрювання	0 — вимк 1 — увімк	1
14/0x000E	R	Поточний статус роботи вентилятора у режимі SILENT	0 — вимк 1 — увімк	1
15/0x000F	R/W/RW	Дозвіл роботи за датчиком вологості	0 — вимк 1 — в автоматичному режимі 2 — у ручному режимі	1
17/0x0011	R/W/RW	Дозвіл роботи за датчиком температури	0 — вимк 1 — увімк 2 — інвертувати	1
18/0x0012	R/W/RW	Дозвіл роботи за датчиком руху	0 — вимк 1 — увімк 2 — інвертувати	1
19/0x0013	R/W/RW	Дозвіл роботи за сигналом від зовнішнього вимикача	0 — вимк 1 — увімк 2 — інвертувати	1
24/0x0018	R/W/RW/INC/DEC	Встановлення швидкості Max	30...100 %	1
26/0x001A	R/W/RW/INC/DEC	Встановлення швидкості Silent	30...100 %	1

Номер параметра, Дес./Hex.	Функції	Опис	Можливі значення	Розмір, байт
27/0x001B	R/W/RW/INC/DEC	Встановлення швидкості інтервального провітрювання	30...100 %	1
29/0x001D	R/W/RW	Активування режиму інтервального провітрювання	0 — вимк 1 — увімк 2 — інвертувати	1
30/0x001E	R/W/RW	Активування режиму Silent Mode	0 — вимк 1 — увімк 2 — інвертувати	1
31/0x001F	R/W/RW	Час старту роботи режиму Silent Mode у секундах	0...86400 секунд	3
32/0x0020	R/W/RW	Час закінчення роботи режиму Silent Mode у секундах	0...86400 секунд	3
33/0x0021	R/W/RW	Поточний час внутрішнього годинника вентилятора у секундах	0...86400 секунд	3
35/0x0023	R/W/RW/INC/DEC	Встановлене значення таймера затримки вимкнення/BOOST	0 — вимк 2 — 5 хвилин 3 — 15 хвилин 4 — 30 хвилин 6 — 60 хвилин	1
36/0x0024	R/W/RW/INC/DEC	Встановлене значення затримки увімкнення	0 — вимк 1 — 2 хвилини 2 — 5 хвилин	1
37/0x0025	W	Скидання параметрів до заводських значень	Будь-який байт	1
124/0x007C	R	Пошук пристроїв у локальній мережі Ethernet	Текст ("0...9", "A...F")	16
134/0x0086	R	Версія та дата основної прошивки контролера	1-й байт — версія прошивки (major) 2-й байт — версія прошивки (minor) 3-й байт — день 4-й байт — місяць 5-й, 6-й байт — рік	6
148/0x0094	R/W/RW	Режим роботи Wi-Fi	1 — client 2 — access point	1
149/0x0095	R/W/RW	Ім'я Wi-Fi у режимі Client	Текст	1 ... 32
150/0x0096	R/W/RW	Пароль Wi-Fi	Текст	8 ... 64
153/0x0099	R/W/RW	Тип шифрування даних Wi-Fi	48 — OPEN 50 — WPA_PSK 51 — WPA2_PSK 52 — WPA_WPA2_PSK	1
154/0x009A	R/W/RW	Частотний канал Wi-Fi	1...13	1
155/0x009B	R/W/RW	DHCP Wi-Fi модуля	0 — STATIC 1 — DHCP 2 — інвертувати	1
156/0x009C	R/W/RW	Задана IP-адреса Wi-Fi модуля	1-й байт — 0...255 2-й байт — 0...255 3-й байт — 0...255 4-й байт — 0...255	4



Номер параметра, Дес./Hex.	Функції	Опис	Можливі значення	Розмір, байт
157/0x009D	R/W/RW	Маска підмережі 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	Основний 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	Застосувати нові параметри Wi-Fi та вийти з режиму налаштування Wi-Fi модуля	Будь-який байт	1
163/0x00A3	R	Поточна IP-адреса Wi-Fi модуля	0...255	4
185/0x00B9	R	Тип пристрою		2

## ПРИКЛАД ОБРОБКИ ПАКЕТІВ МОВОЮ С

```
//===== Спеціальні команди =====//
#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 контролера

//***** Перевірка контрольної суми та початок 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) // Контрольна сума
    {
        if(receive_data[2] == 0x02) // Тип протоколу
        {
            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]; // Місце у масиві, де починається блок FUNC
                //***** FUNC і DATA *****//
                for(; r_pos < receive_data_size - 2; r_pos++)
                {
                    //===== Спеціальні команди =====//
                    param_size = 1;
                    //=== новий номер функції
                    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) // якщо номер функції не підтримується
                            break;
                        continue;
                    }
                    //=== нове значення старшого байта для номерів параметрів
                    else if(receive_data[r_pos] == BGCP_CMD_PAGE)
                    {

```

```
        page = receive_data[++r_pos];
        continue;
    }
    //== нове значення розміру параметра
    else if(receive_data[r_pos] == BGCP_CMD_SIZE)
    {
        param_size = receive_data[++r_pos];
        r_pos++;
    }
    //== якщо параметр не підтримується
    else if(receive_data[r_pos] == BGCP_CMD_NOT_SUP)
    {
        r_pos++;
        //***** обробка непідтримуваних параметрів *****//
        param = (uint16_t)(page << 8) | (uint16_t)(receive_data[r_pos]);
        switch(param)
        {
            case 0x0001:
                break;
            case 0x0002:
                break;
            ...
        }
        //*****//
        continue;
    }
    jump_size = param_size;
    //=====//

    //***** обробка підтримуваних параметрів *****//
    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;
}
//*****//
}
}
}
```

