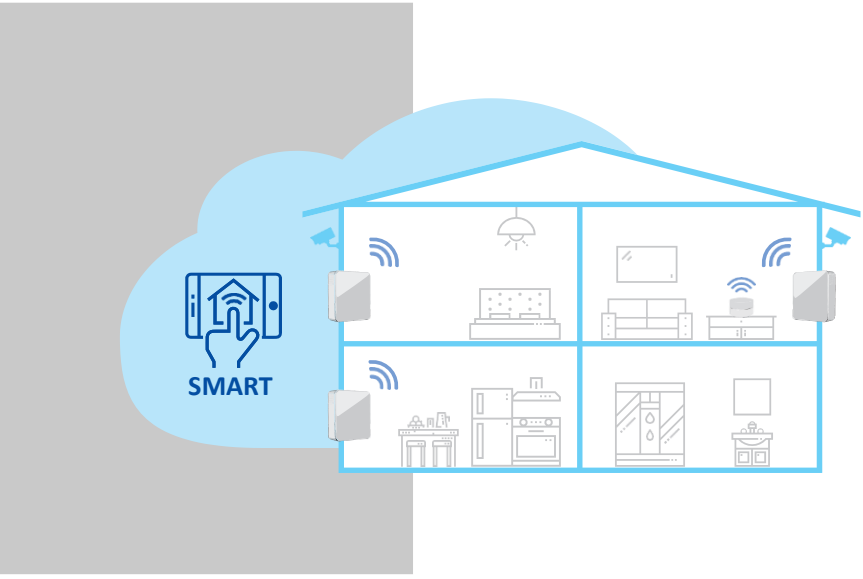
AlphaAir Freshbox 100 WiFi

Connection to a Smart Home System

March 2021



RadonTec GmbH
Hauptstraße 5
89426 Wittislingen - Germany
Tel: (+49) 9076 - 919 98 35
E-Mail: info@radontec.de
Website: radontec.de
Shop: radonshop.com

Version: 01

Scope of Contents

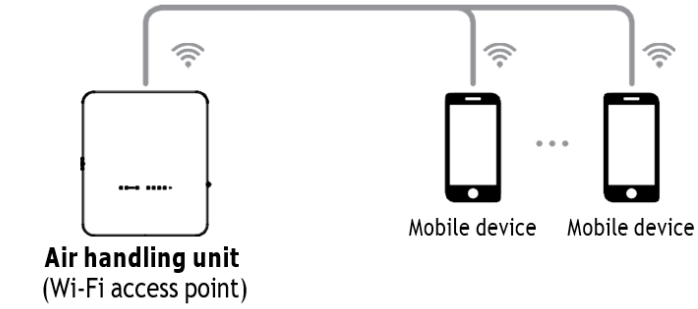
1 Introduction

The AlphaAir Freshbox 100 WiFi is a ventilation unit with pre-heating coil, enthalpy exchanger and WiFi connection. It can be integrated into your Smart Home in just a few steps.

2 Connection and adjustment

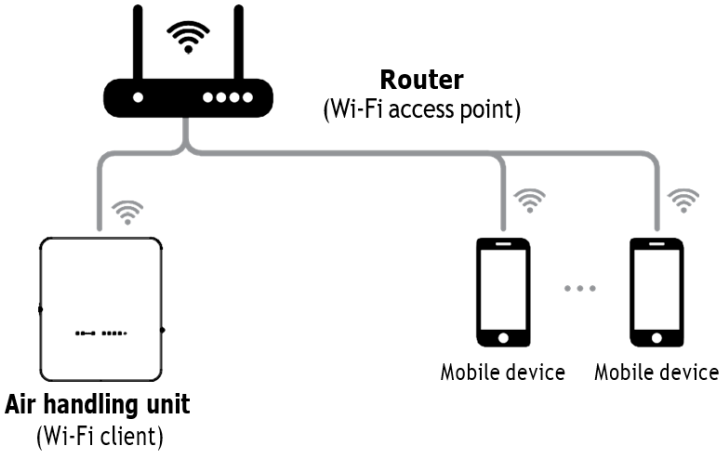
Example 1: Diagram of direct connection of the ventilation system to the Smart Home System without using a router. Set the ventilation system for WLAN operation in access point mode (see operating instructions of the ventilation system).

A maximum of eight control units can be connected.



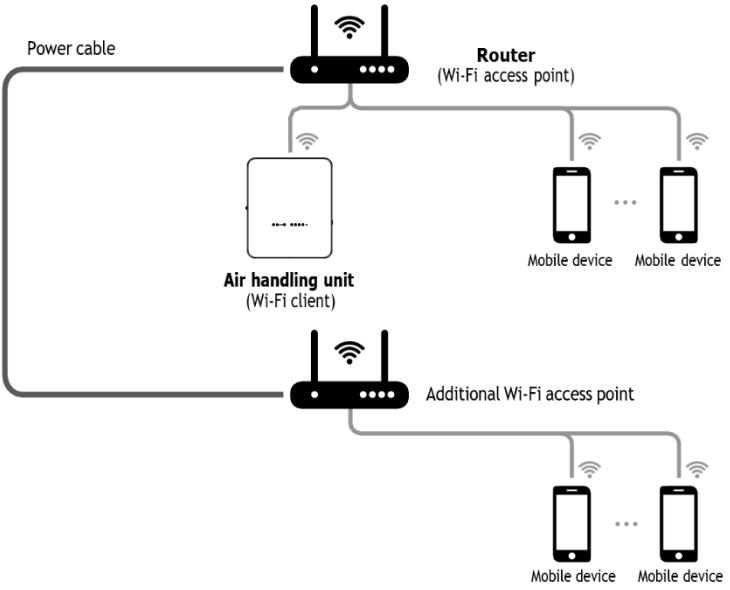
Example 2: Connection diagram with a router with a single WLAN access point.

Ventilation systems, mobile devices and the Smart Home System are connected to the router's WLAN access point.



Beispiel 3: Schema für direkten Anschluss des Smart Home

System with a router to which several WLAN access points are connected.



3 Network parameters

Data exchange takes place via the UDP protocol (with broadcast support).

IP address of the master system:

- 192.168.4.1: If the master unit is running without a router (connection scheme no. 1).
- If the master unit is connected via a router (connection scheme no. 2), the IP address is set via the app (see operating instructions of the unit) and can be determined statically or dynamically (DHCP).

Port of the master unit: 4000 Maximum packet size: 256 bytes

4 Package structure

0xFD **0xFD** Packet start character (2 bytes)

TYPE Protocol type (1 byte). Value = 0x02

SIZE ID ID block size (1 byte). Value = 0x10

ID ID number of the control unit. This number is listed on the label (16 characters) on the control board or the unit housing.
You can also replace the ID number with the code word "DEFAULT_DEVICEID". The ID number can be used

- To control when the master unit is running without a router (connection diagram no. 1).
- To search for master units in the network if a router is used (connection scheme no. 2). In this case, the installation only responds to two parameters: 0x007C and 0x00B9 (see parameter table).

SIZE PWD PWD block size (1 byte). Possible values: from 0x00 to 0x08.

PWD Password of the installation (permissible characters: "0... 9", "a... z" and "A... Z"). The default password is "1111". This password can be changed via the app in the menu Connection -> Local -> Settings (see operating instructions of the unit).

FUNC Function number (1 byte). It defines the action with the data and the DATA block structure:

- 0x01: Parameter read

- 0x02: Parameter write. The control unit does not send a response regarding the status of the specified parameters.

- 0x03: Parameter write with subsequent response from the control unit regarding the status of the specified parameters.

- 0x04: Parameter increment with subsequent response from the control unit regarding the status of the specified parameters.

- 0x05: Parameter decrement followed by a response from the control unit regarding the status of the specified parameters.

- 0x06: Response of the control unit to the request (FUNC = 0x01, 0x03, 0x04, 0x05).

DATA Data block. It consists of parameter numbers and their values:

If FUNC = 0x01 or 0x04 or 0x05:

P1	P2	Pn

If FUNC = 0x02 or 0x03 or 0x06:

P1	Value 1	P2	Value 2	Pn	Value n

Parameter numbers (see parameter table) consist of two bytes (high byte is virtual). By default, the high byte of each parameter number in each new packet corresponds to

0x00. The high byte can be changed within a single packet with the special command 0xFF (see below).

P Low byte of the parameter number. Possible values: 0x00-0xFB. The values 0xFC-0xFF are special commands:

0xFC Change function number (FUNC). The following byte must be the new function number, ranging from 0x01 to 0x05. This command is used to organise several functions with different actions in a single package.

0xFD Parameter is not supported by the control unit. The following byte is the low byte of the unsupported parameter. This command is used in the control unit response (FUNC = 0x06) to an unsupported parameter read or write request.

0xFE Change the size of the parameter value Value for one of the following parameters. The following byte must be the new parameter size, followed by the low byte of the parameter number and then - by the value itself.

0xFF Change high byte for parameter numbers within a single packet. The following byte must be the new high byte.

Value Parameter value (1 byte by default). Byte order from low byte to high byte.

Chksum L	Chksum H

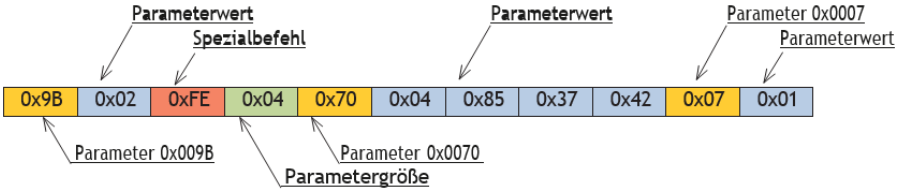
Checksum (2 bytes). This is calculated as the total number of bytes starting with the TYPE byte and ending with the last byte of the DATA block.

Chksum L: Low byte of the checksum.

Chksum H: High byte of the checksum.

5 Application examples of the special commands in the data block

Write request (FUNC = 0x03) for parameters with the numbers 0x009B, 0x0070 and 0x0007.



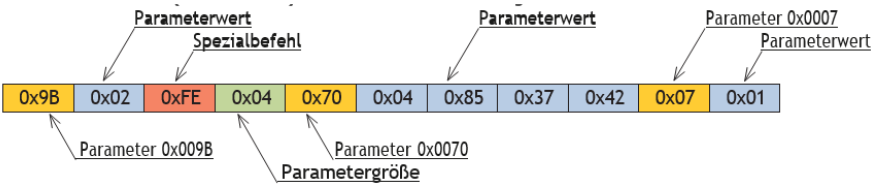
Details of the write request:

The parameter 0x009B must be assigned the value 0x02.

The parameter 0x0070 must be assigned the value 0x42378504. The value size is 4 bytes as specified by the special command 0xFE + 0x04.

The parameter 0x0007 must be assigned the value 0x01.

Response of the control unit (FUNC = 0x06) to the write request



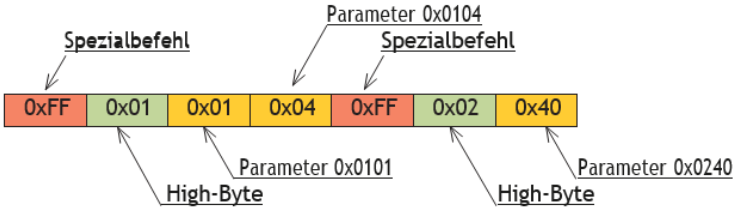
Response of the control unit:

Parameter 0x009B corresponds to 0x02.

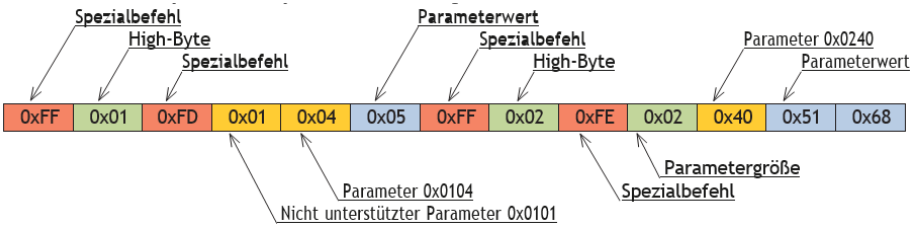
Parameter 0x0070 corresponds to 0x42378504. The value size is 4 bytes as specified by the special command 0xFE + 0x04.

Parameter 0x0007 corresponds to 0x01.

Read request (FUNC = 0x01) for parameters with numbers 0x0101, 0x0104 and 0x0240



Response of the control unit (FUNC = 0x06) to the read request.



Response of the control unit:

Parameter 0x0101 is not supported by the control unit as indicated by special command 0xFD.

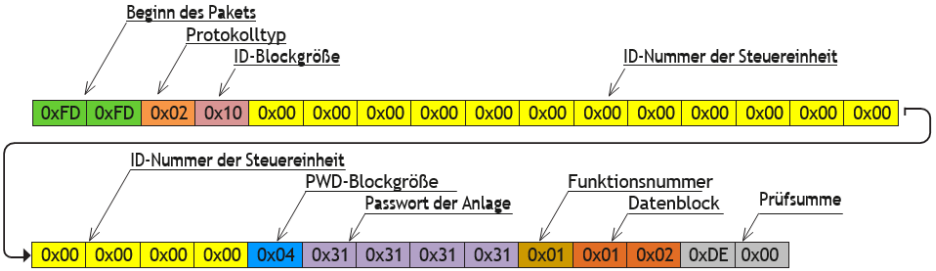
Parameter 0x0104 corresponds to 0x05.

Parameter 0x0240 corresponds to 0x6851. The value size is 2 bytes as specified by the special command 0xFE + 0x02.

6 Examples of a complete package

Sending the packet "Smart Home -> Control Unit".

This packet contains a read request (FUNC = 0x01) for parameters with numbers: 0x0001, 0x0002..

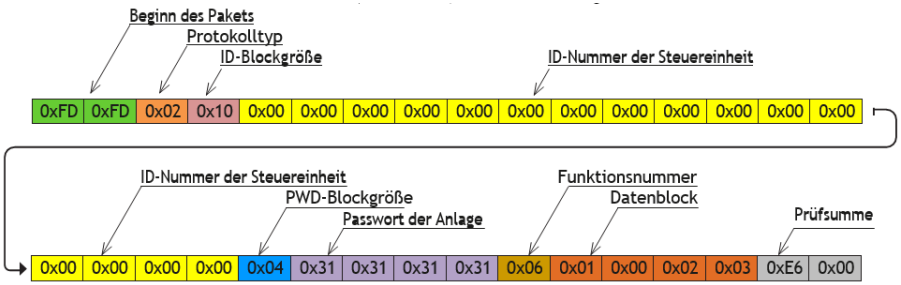


Request details:

Checksum: 0x00DE

Sending the packet "Control unit -> Smart Home".

This packet contains the response of the control unit (FUNC = 0x06) to the read request.



Response from the control unit:

Parameter 0x0001 corresponds to 0x00

Parameter 0x0002 corresponds to 0x03

Checksum: 0x00E6

7 Parameter table

Functions: R - 0x01 INC - 0x04 RW - 0x03
W - 0x02 DEC - 0x05

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
1/0x0001	R/W/RW	System On/Off	0: Off 1: On 2: Invert	1
2/0x0002	R/W/RW/INC/DEC	Ventilation stage operation	1: Ventilation level 1 2: Ventilation level 2 3: Ventilation level 3 4: Ventilation level 4 5: Ventilation level 5	1
3/0x0003	R/W/RW/INC/DEC	Maximum ventilation stage number	3, 5	1
6/0x0006	R	Boost mode. The system switches to the boost ventilation stage for the duration of the boost operation run-on time (see parameters 70, 71, 102).	0: Off 1: On 2: Invert	1
7/0x0007	R/W/RW	Timer On/Off	0: Off 1: On 2: Invert	1
8/0x0008	R/W/RW/INC/DEC	Timer operation	0: Standby 1: Ventilation level 1 2: Ventilation level 2 3: Ventilation level 3 4: Ventilation level 4 5: Ventilation level 5	1
9/0x000	R/W/RW/INC/DEC	Minute setpoint of the timer	0...59minutes	1

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
10/0x000A	R/W/RW/INC/DEC	Hour setpoint of the timer	0...23 hours	1
11/0x000B	R	Current countdown time of the timer	Byte 1: seconds (0...59) Byte 2: minutes (0...59) Byte 3: hours (0...23)	3
13/0x000D	R/W/RW/INC/DEC	Room temperature setpoint in timer mode	0: ventilation only, +15...+30 °C	1
20/0x0014	R/W/RW	Control with a BOOST switch	0: Off 1: On 2: Invert	1
21/0x0015	R/W/RW	Control via fire detector	0: Off 1: On 2: Invert	1
24/0x0018	R/W/RW/INC/DEC	Room temperature setpoint in standard mode	+15...+30 °C	1
29/0x001D	R/W/RW/INC/DEC	Selection of a temperature sensor to control the room temperature	0: in the extract air duct (ExAirIn) 1: external sensor in the control panel (Ext) 2: in the supply air duct (SuAirOut)	1
30/0x001E	R	Current temperature with which the room temperature is controlled	-32768: the sensor is missing +3276 7: short circuit	signed 2 (must be divided by 10)
31/0x001F	R	Current supply air temperature at the system inlet	-32768: the sensor is missing +32767: short circuit	signed 2 (must be divided by 10)

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
32/0x0020	R	Current supply air temperature at the system outlet (after the heat exchanger/after the reheating coil)	-32768: the sensor is missing +32767: short circuit	signed 2 (must be divided by 10)
33/0x0021	R	Current extract air temperature at the system inlet	-32768: the sensor is missing +32767: short circuit	signed 2 (must be divided by 10)
34/0x0022	R	Current extract air temperature at the system outlet	-32768: the sensor is missing +32767: short circuit	signed 2 (must be divided by 10)
50/0x0032	R	Current status of the boost switch	0: off 1: On	1
51/0x0033	R	Current status of the fire detector	0: off 1: On	1
54/0x0036	R/W/RW/INC/DEC	Minimum ventilation level of the fan	0...100 %	1
55/0x0037	R/W/RW/INC/DEC	Minimum ventilation level of the fan	0...100 %	1
58/0x003A	R/W/RW/INC/DEC	Ventilation stage of the supply air fan in operation of the first ventilation stage	min…max %	1
59/0x003B	R/W/RW/INC/DEC	Ventilation stage of the extract air fan in operation of the first ventilation stage	min...max %	1

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
60/0x003C	R/W/RW/INC/DEC	Ventilation stage of the supply air fan in operation of the second ventilation stage	min...max %	1
61/0x003D	R/W/RW/INC/DEC	Ventilation stage of the extract air fan in operation of the second ventilation stage	min...max %	1
62/0x003E	R/W/RW/INC/DEC	Ventilation stage of the supply air fan in operation of the third ventilation stage	min...max %	1
63/0x003F	R/W/RW/INC/DEC	Ventilation stage of the extract air fan in operation of the third ventilation stage	min...max %	1
64/0x0040	R/W/RW/INC/DEC	Ventilation stage of the supply air fan in operation of the fourth ventilation stage	min...max %	1
65/0x0041	R/W/RW/INC/DEC	Ventilation stage of the extract air fan in operation of the fourth ventilation stage	min...max %	1
66/0x0042	R/W/RW/INC/DEC	Ventilation stage of the supply air fan in operation of the fifth ventilation stage	min...max %	1
65/0x0041	R/W/RW/INC/DEC	Ventilation stage of the extract air fan in operation of the fourth ventilation stage	min...max %	1

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
66/0x0042	R/W/RW/INC/DEC	Ventilation stage of the supply air fan in operation of the fifth ventilation stage	min...max %	1
67/0x0043	R/W/RW/INC/DEC	Ventilation stage of the extract air fan in operation of the fifth ventilation stage	min...max %	1
69/0x0045	R/W/RW/INC/DEC	Ventilation stage of the fans during flushing of electric heating coils	min...max %	1
70/0x0046	R/W/RW/INC/DEC	Ventilation stage of the supply air fan in BOOST mode	min...max %	1
71/0x0047	R/W/RW/INC/DEC	Ventilation stage of the extract air fan in the BOOST	min...max %	1
96/0x0060	R/W/RW/INC/DEC	Type of reheating coil	0: switch off 1: electric (fixed value)	1
99/0x0063	R/W/RW/INC/DEC	Time setting of the filter change timer	0, 70...365 days at 5-day intervals	2
100/0x0064	R	Countdown of the timer until filter change	Byte 1: Minutes (0...59) Byte 2: Hours (0...23) Byte 3: Days (0...365)	4
101/0x0065	W	Reset countdown of the timer until filter change	Each byte	1

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
102/0x0066	R/W/RW/INC/DEC	Setpoint of the overrun time of the boost mode	0…60 min	1
103/0x0067	R/W/RW/INC/DEC	Setpoint of the switch-on delay of the boost mode	0…15 min	1
104/0x0068	R/W/RW	Temperature control in standard mode	0: Off 1: On 2: invert	1
106/0x006A	R	Temperature TE5	-32768: the sensor is missing +32767: short circuit	signed 2 (must be divided by 10)
111/0x006F	R/W/RW	RTC time	Byte 1: RTC seconds Byte 2: RTC minutes Byte 3: RTC hours	3
112/0x0070	R/W/RW	RTC calendar	Byte 1: RTC date Byte 2: RTC day of the week Byte 3: RTC month Byte 4: RTC year	4
114/0x0072	R/W/RW	Time-controlled operation	0: Off 1: On 2: invert	1
115/0x0073	R	Ventilation level of the time-controlled operation	0: standby 1: ventilation level 1 2: ventilation level 2 3: ventilation level 3 4: ventilation level 4 5: ventilation level 5	1

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
116/0x0074	R	Temperature setting of the timed operation	0: ventilation only,	1
119/0x0077	R/W/RW	Schedule setting	+15...+30 °C Byte 1: Weekday: 0: all ages (write only) 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Sunday 8: Mon...Fri (write only) 9: Sat...Sun (write only) Byte 2: Period number: 1...4 Byte 3: Ventilation stage number: 0: standby 1...5 Byte 4: Temperature 0: ventilation only, +15...+30 °C Byte 5: minutes to the end of the interval: 0...59 Byte 6: hours to the end of the distance: 0...23	6

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
124/0x007C	R	Searching for installations in the local network Ethernet	Text ("0...9", "A...F")	16
125/0x007D	R/W/RW	Password of the installation for the network Ethernet	Text ("0...9", "a...z", "A...Z")	0-8
126/0x007E	R	Operating hours	Byte 1: Minutes (0...59) Byte 2: Hours (0...23) Byte 3 and 4: Days (0...65535)	4
127/0x007F	R	List of current alarms/warnings	Byte 1: Code Byte 2: Type: 1: Alarm 2: Warning	0, 2 , 4...
128/0x0080	W	Reset alarms	Each byte	1
129/0x0081	R	Status of the heating register	0: Off 1: On	1
131/0x0083	R	Alarm/warning display	0: No alarms 1: Alarm (highest priority) 2: Warning	1
133/0x0085	R/W/RW	Control via cloud server	0: Off 1: On 2: Invert	1
134/0x0086	R	Version and date of the base firmware of the control unit	Byte 1: Firmware version (major) Byte 2: Firmware version (minor) Byte 3: Day Byte 4: Month Byte 5 and 6: Year	6

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
135/0x0087	W	Reset factory settings	Each byte	1
136/0x0088	R	Filter status	0: clean 3: Triggering the filter change timer	1
147/0x0093	R	Presence of a WLAN module on the board	0: does not exist 1: does exist	1
148/0x0094	R/W/RW	WLAN operation	1: client 2: access point	1
149/0x0095	R/W/RW	WLAN name in client mode	Text	1…32
150/0x0096	R/W/RW	WLAN password	Text	8 … 64
153/0x0099	R/W/RW	WLAN encryption technology	48: OPEN 50: WPA_PSK 51: WPA2_PSK 52: WPA_WPA2_PSK	1
154/0x009A	R/W/RW	WLAN frequency channel	1…13	1
155/0x009B	R/W/RW	WLAN module DHCP	0: STATIC 1: DHCP 2: invert	1
156/0x009C	R/W/RW	Assigned IP address of the WLAN module	Byte 1: 0…255, Byte 2: 0…255, Byte 3: 0…255, Byte 4: 0…255	4
157/0x009D	R/W/RW	Subnet mask of the WLAN module	Byte 1: 0…255, Byte 2: 0…255, Byte 3: 0…255, Byte 4: 0…255	4

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
158/0x009E	R/W/RW	Main gateway of the WLAN module	Byte 1: 0…255, Byte 2: 0…255, Byte 3: 0…255, Byte 4: 0…255	4
159/0x009F	R/W/RW	DNS server address for WLAN module	Byte 1: 0…255, Byte 2: 0…255, Byte 3: 0…255, Byte 4: 0…255	4
160/0x00A0	W	Accept new WLAN parameters and exit the setting mode of the WLAN module.	Each Byte	1
161/0x00A1	R	Connection status of the WLAN module to the access point of the router	0: not connected 1: connected	1
162/0x00A2	W	Exit the setting mode of the WLAN module without using the new parameters.	Each Byte	1
163/0x00A3	R	Current IP address of the WLAN module	Byte 1: 0…255, Byte 2: 0…255, Byte 3: 0…255, Byte 4: 0…255	4
182/0x00B6	R	Flush status of the electric heating register (pre-heating, post-heating)	0: off 1: on	1
185/0x00B9	R	System type	0x0002	2

Parameter number, dec./hex.	Functions	Description	Possible values	Size, bytes
240/0x00F0	R/W/RW/INC/DEC	Recirculation damper	0: Switch off recovery I: Switch on recovery (only for systems with recovery)	1
252/0x00FC		Special command		
253/0x00FD		Special command		
254/0x00FE		Special command		
255/0x00FF		Special command		
273/0x0111	R	Control panel type		2
274/0x0112	R	Version and date of the basic firmware of the control panel	Byte 1: Firmware version (major) Byte 2: Firmware version (minor) Byte 3: Day Byte 4: Month Byte 5 and 6: Year	6
1024/0x0400	R/W/RW	Brightness setpoint of the key illumination	0...80 (20-100%)	1
1025/0x0401	R/W/RW	Switching on/off the sounder on the board	0: Off 1: On	1
1026/0x0402	R/W/RW	Selection of the lighting mode	0: static operation 1: dynamic operation	1

8 Example of package processing, written in c

```
//================ Special commands ================//
#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"; // Controller ID

//********* Checksum and start of packet check ***********//
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) // Checksum
    {
        if(receive_data[2] == 0x02) // Protocol type
        {
            if(memcmp(&receive_data[4], current_id, receive_data[3]) == 0) // ID
            {
                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]; // Position in array where FUNC block begins
                //******************** FUNC and DATA ********************//
                for(; r_pos < receive_data_size - 2; r_pos++)
                {
                    //========== Special commands ==========//
                    param_size = 1;
                    //=== New function number
                    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) // If the function number is not supported
                           break;
                        continue;
                    }
                    //=== New lead byte value for parameter numbers
                    else if(receive_data[r_pos] == BGCP_CMD_PAGE)
                    {
```

```
                    page = receive_data[++r_pos];
                    continue;
                }
                //=== New parameter size value
                else if(receive_data[r_pos] == BGCP_CMD_SIZE)
                {
                    param_size = receive_data[++r_pos];
                    r_pos++;
                }
                //=== If the parameter is not supported
                else if(receive_data[r_pos] == BGCP_CMD_NOT_SUP)
                {
                    r_pos++;
                    //******* Processing of non-supported parameters ******//
                    param = (uint16_t)(page << 8) | (uint16_t)(receive_data[r_pos]);
                    switch(param)
                    {
                       case 0x0001:
                          break;
                       case 0x0002:
                          break;
                        ...
                    }
                    //***********************//
                    continue;
                }
                jump_size = param_size;
                //===================================//

                //******* Processing of supported parameters ******//
                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;
            }
            //*************************************************************//
        }
    }
}
```

9 Support and Contact

9.1 Troubleshooting/FAQ

You will find answers to the most frequently asked questions on our website:

https://www.radontec.de

9.2 Contact Us

Should you have any further questions or require further help and technical support, please do not hesitate to contact us.

RadonTec GmbH
Hauptstraße 5
89426 Wittislingen - Germany
Tel: (+49) 9076 - 919 98 35
E-Mail: info@radontec.de
Website: radontec.de
Shop: radonshop.com