

# ANSCHLUSSANLEITUNG

---

**TwinFresh Expert RW-30 V.2**

**TwinFresh Expert RW1-50 V.2**

**TwinFresh Expert RW1-85 V.2**

**TwinFresh Expert RW1-100 V.2**

**TwinFresh Expert Duo RW1-30 V.2**

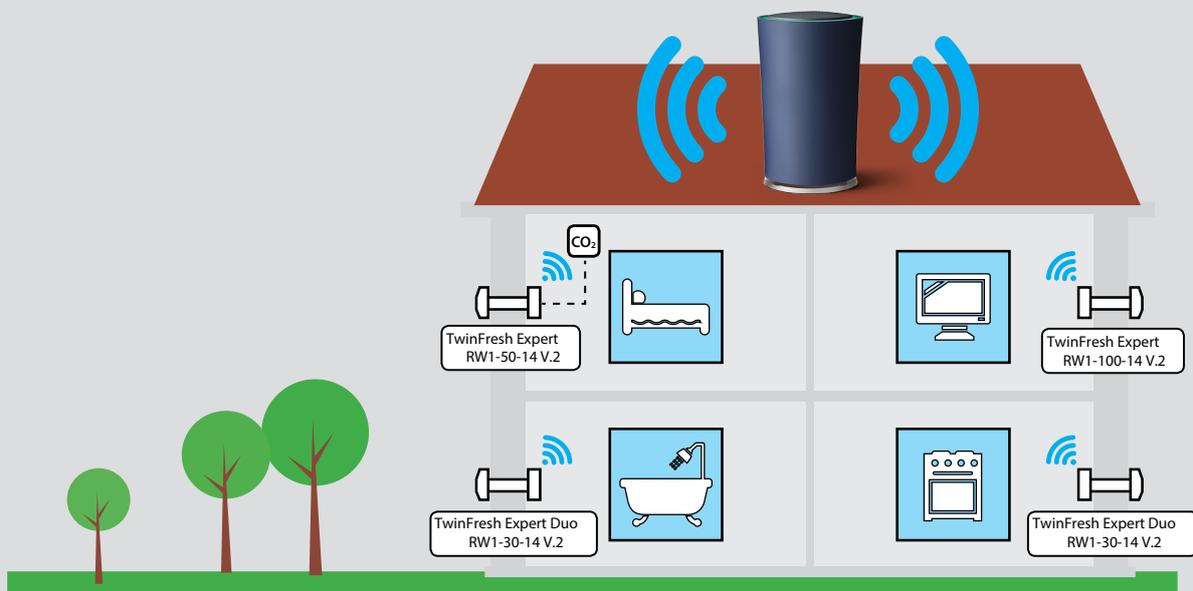
**TwinFresh Expert RW1-50 V.3**

**TwinFresh Style Wi-Fi**

**TwinFresh Style Wi-Fi Frost**

**TwinFresh Style Wi-Fi mini**

## Smart House



Anschluss an ein „Smart House“ System

---

## INHALT

Verwendungszweck.....	2
Netzparameter .....	3
Packetstruktur.....	4
Anwendungsbeispiele der speziellen Befehle im Datenblock.....	5
Beispiele eines kompletten Pakets .....	6
Parametertabelle .....	7
Beispiel der Paketverarbeitung, in C geschrieben .....	10

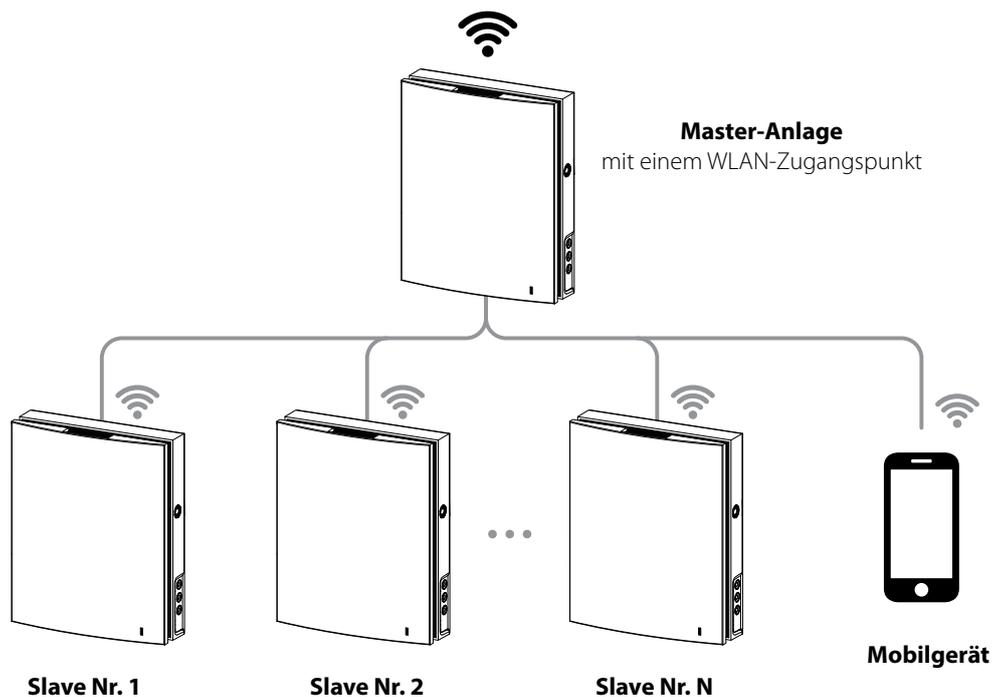
## VERWENDUNGSZWECK

In dieser Anleitung wird der Anschluss von Lüftungsanlagen der Serie TwinFresh Expert RW V.2 (V.3) und TwinFresh Style Wi-Fi an ein „Smart House“ System beschrieben.

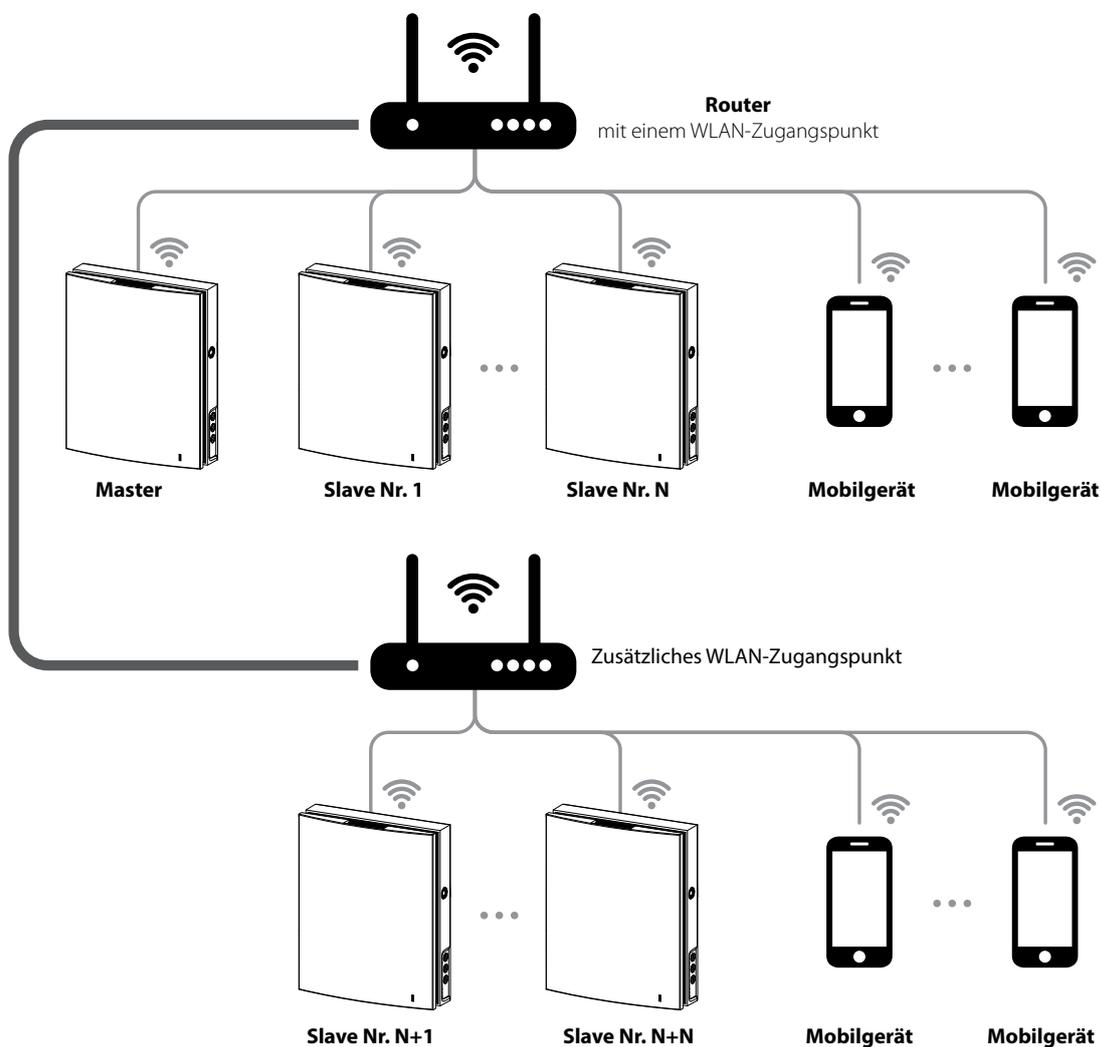
Die Steuerung erfolgt nur über Master-Anlagen. Slave-Anlagen, Mobilgeräte und „Smart House“ Systemkomponenten sind über WLAN mit Master-Anlagen verbunden. Slave-Anlagen werden über Befehle gesteuert, die von Master-Anlagen empfangen werden.

### Es stehen zwei Schemata des drahtlosen Anschlusses zur Verfügung:

1. Eine Master-Anlage mit einem WLAN-Zugangspunkt, der bis zu acht Slave-Verbindungen akzeptieren kann. Wenn alle verfügbaren acht Verbindungen der Master-Anlage verwendet werden, kann ein Mobilgerät oder ein „Smart House“ System keine Verbindung zur Master-Anlage herstellen.



2. Master-Anlagen, Slave-Anlagen, Mobilgeräte und ein „Smart House“ System stellen eine Verbindung zum WLAN-Zugangspunkt des Netzwerk-Routers her. In diesem Fall ist die maximale Anzahl verfügbarer WLAN-Verbindungen durch die Routerfunktionalität begrenzt. Wenn mehr Lüftungsanlagen erforderlich sind, als der Router akzeptieren kann, können Sie einen zusätzlichen WLAN-Zugangspunkt verwenden, um die verbleibenden Lüftungsanlagen anzuschließen. Das Router-Netzwerk kann mehrere Master-Anlagen enthalten, um die Steuerung nach Zonen zu ermöglichen.



**Die Verbindung wird über die App im Menü Verbindung -> WLAN-Einstellung hergestellt (siehe Betriebsanleitung des Geräts).**

## NETZPARAMETER

Der Datenaustausch erfolgt über das UDP-Protokoll (mit Broadcast-Unterstützung).

IP-Adresse der Master-Anlage:

- 192.168.4.1: Wenn die Master-Anlage ohne Router läuft (Anschlussschema Nr. 1).
- Wenn die Master-Anlage über einen Router verbunden ist (Anschlussschema Nr. 2), wird die IP-Adresse über die App eingestellt (siehe Betriebsanleitung des Geräts) und kann statisch oder dynamisch (DHCP) bestimmt werden.

Port der Master-Anlage: 4000

Maximale Paketgröße: 256 Bytes

## PACKETSTRUKTUR

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

**0xFD 0xFD** : Paketanfangszeichen (2 Bytes).

**TYPE** : Protokolltyp (1 Byte). Wert = 0x02.

**SIZE ID** : ID-Blockgröße (1 Byte). Wert = 0x10.

**ID** : ID-Nummer der Steuereinheit. Diese Nummer ist auf dem Etikett (16 Zeichen) auf der Steuerplatine oder dem Gerätegehäuse angeführt.

Sie können die ID-Nummer auch durch das Codewort „DEFAULT\_DEVICEID“ ersetzen. Die ID-Nummer kann verwendet werden:

- Zur Steuerung, wenn die Master-Anlage ohne Router läuft (Anschlussschema Nr. 1).
- Um im Netzwerk nach Master-Anlagen zu suchen, wenn ein Router verwendet wird (Anschlussschema Nr. 2). In diesem Fall reagiert die Anlage nur auf zwei Parameter: 0x007C und 0x00B9 (siehe Parametertabelle).

**SIZE PWD** : PWD-Blockgröße (1 Byte). Mögliche Werte: von 0x00 bis 0x08.

**PWD** : Passwort der Anlage (zulässige Zeichen: „0... 9“, „a... z“ und „A... Z“). Das Standardpasswort lautet „1111“.

Dieses Passwort kann über die App im Menü **Verbindung -> Lokal -> Einstellungen** geändert werden (siehe Betriebsanleitung des Geräts).

**FUNC** : die Funktionsnummer (1 Byte). Es definiert die Aktion mit den Daten und der **DATA**-Blockstruktur:

0x01: Parameterlesen.

0x02: Parameterschreiben. Die Steuereinheit sendet keine Antwort bezüglich des Status der angegebenen Parameter. Schreiben der Parameter.

0x03: Parameterschreiben mit anschließender Antwort der Steuereinheit bezüglich des Status der angegebenen Parameter.

0x04: Parameterinkrement mit anschließender Antwort der Steuereinheit bezüglich des Status der angegebenen Parameter.

0x05: Parameterdekrement mit anschließender Antwort der Steuereinheit bezüglich des Status der angegebenen Parameter.

0x06: Antwort der Steuereinheit auf die Anfrage (FUNC = 0x01, 0x03, 0x04, 0x05).

**DATA** : der Datenblock. Es besteht aus Parameternummern und ihren Werten:

Wenn FUNC = 0x01 oder 0x04 oder 0x05:

P1	P2	Pn
----	----	----

Wenn FUNC = 0x02 oder 0x03 oder 0x06:

P1	Value 1	P2	Value 2	Pn	Value n
----	---------	----	---------	----	---------

Parameternummern (siehe Parametertabelle) bestehen aus zwei Bytes (High-Byte ist virtuell). Standardmäßig entspricht das High-Byte jeder Parameternummer in jedem neuen Paket 0x00. Das High-Byte kann innerhalb eines einzelnen Pakets mit dem speziellen Befehl **0xFF** geändert werden (siehe unten).

**P** ist das Low-Byte der Parameternummer. Mögliche Werte: 0x00–0xFB. Die Werte 0xFC–0xFF sind Spezialbefehle:

**0xFC** : Funktionsnummer ändern (**FUNC**). Das folgende Byte muss die neue Funktionsnummer sein, die von 0x01 bis 0x05 reicht. Dieser Befehl wird verwendet, um mehrere Funktionen mit unterschiedlichen Aktionen in einem einzigen Paket zu organisieren.

**0xFD** : Parameter wird von der Steuereinheit nicht unterstützt. Das nachfolgende Byte ist das Low-Byte des nicht unterstützten Parameters. Dieser Befehl wird in der Antwort der Steuereinheit (**FUNC** = 0x06) auf eine nicht unterstützte Lese- oder Schreibanforderung von Parametern verwendet.

**0xFE** : Die Größe des Parameterwerts **Value** für einen der folgenden Parameter ändern. Das nachfolgende Byte muss die neue Parametergröße sein, gefolgt vom Low-Byte der Parameternummer und dann – vom Wert (**Value**) selbst.

**0xFF** : Das High-Byte für Parameternummern innerhalb eines einzelnen Pakets ändern. Das nachfolgende Byte muss das neue High-Byte sein.

**Value** : Parameterwert (Standardgröße ist 1 Byte). Byteanordnung von Low-Byte zu High-Byte.

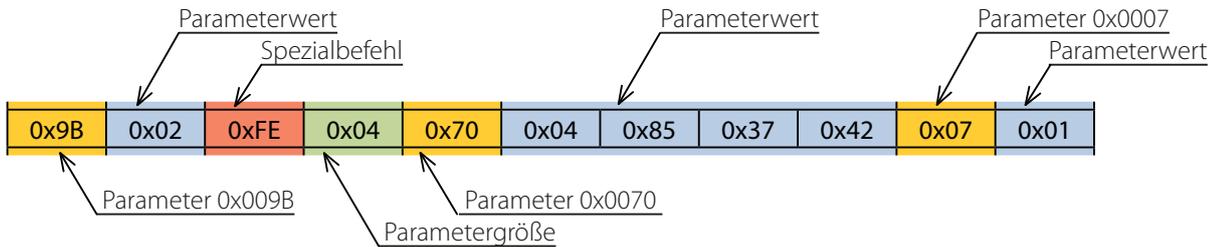
**Chksum L Chksum H** : Prüfsumme (2 Bytes). Dies wird als die Gesamtzahl von Bytes berechnet, die mit dem **TYPE**-Byte beginnen und mit dem letzten Byte des **DATA**-Blocks enden.

**Chksum L**: Low-Byte der Prüfsumme.

**Chksum H**: High-Byte der Prüfsumme.

## ANWENDUNGSBEISPIELE DER SPEZIELLEN BEFEHLE IM DATENBLOCK

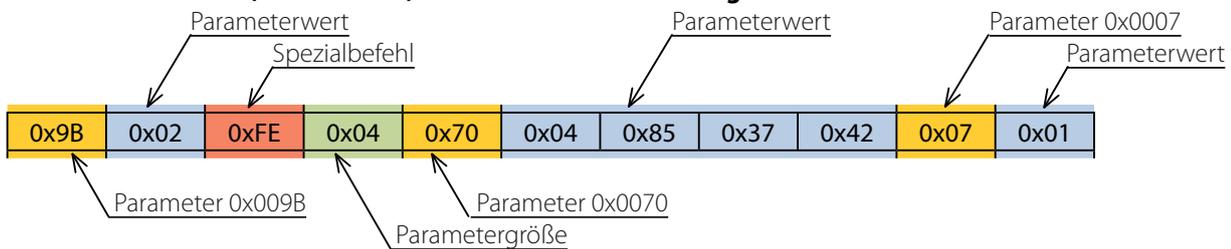
### Schreibanforderung (FUNC = 0x03) für Parameter mit den Nummern 0x009B, 0x0070 und 0x0007



Details der Schreibanforderung:

- Dem Parameter 0x009B muss der Wert 0x02 zugewiesen werden.
- Dem Parameter 0x0070 muss der Wert 0x42378504 zugewiesen werden. Die Wertgröße beträgt 4 Byte, wie durch den Spezialbefehl 0xFE + 0x04 angegeben.
- Dem Parameter 0x0007 muss der Wert 0x01 zugewiesen werden.

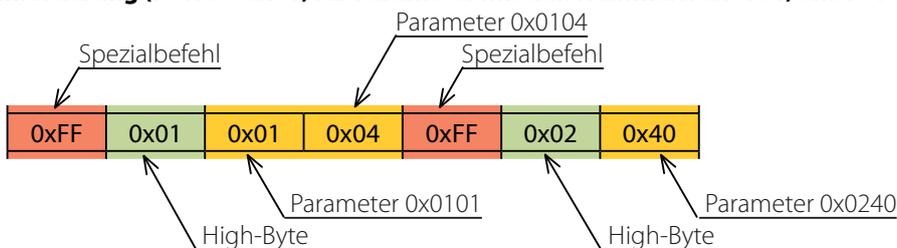
### Antwort der Steuereinheit (FUNC = 0x06) auf die Schreibanforderung



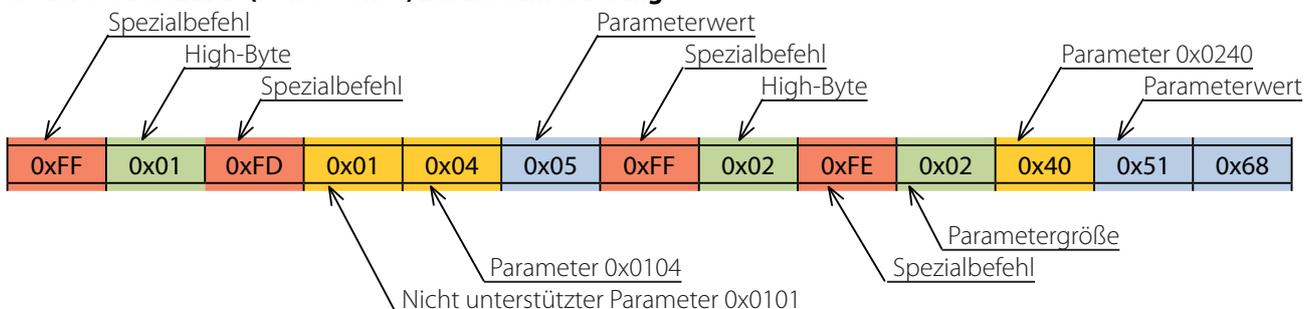
Antwort der Steuereinheit:

- Parameter 0x009B entspricht 0x02.
- Parameter 0x0070 entspricht 0x42378504. Die Wertgröße beträgt 4 Byte, wie durch den Spezialbefehl 0xFE + 0x04 angegeben.
- Parameter 0x0007 entspricht 0x01.

### Leseanforderung (FUNC = 0x01) für Parameter mit den Nummern 0x0101, 0x0104 und 0x0240



### Antwort der Steuereinheit (FUNC = 0x06) zur Leseanforderung



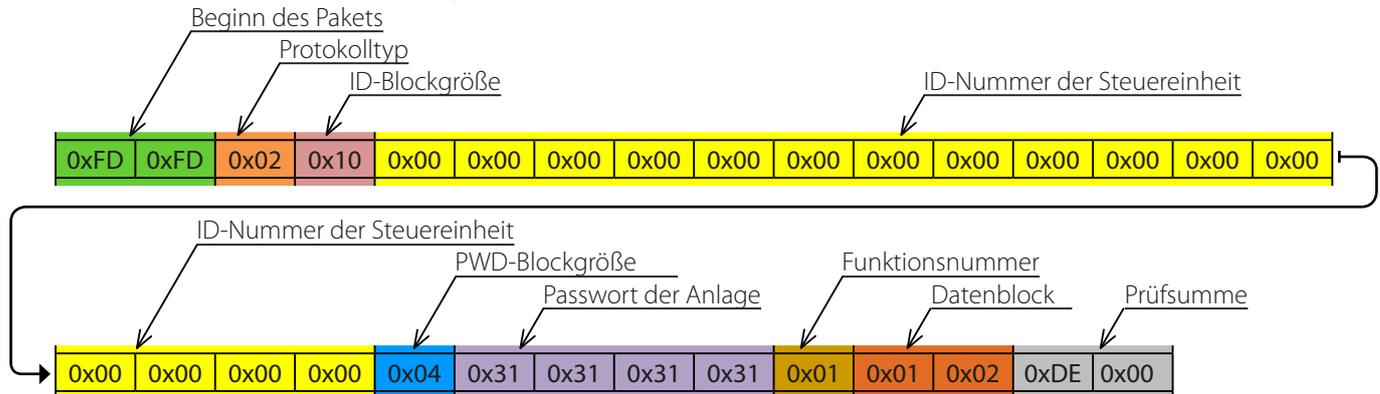
Antwort der Steuereinheit:

- Der Parameter 0x0101 wird von der Steuereinheit nicht unterstützt, wie durch den Spezialbefehl 0xFD angegeben.
- Parameter 0x0104 entspricht 0x05.
- Parameter 0x0240 entspricht 0x6851. Die Wertgröße beträgt 2 Byte, wie durch den Spezialbefehl 0xFE + 0x02 angegeben.

## BEISPIELE EINES KOMPLETTEN PAKETS

### Senden des Pakets „Smart House -> Steuereinheit“

Dieses Paket enthält eine Leseanforderung (FUNC = 0x01) für Parameter mit den Nummern: 0x0001, 0x0002.

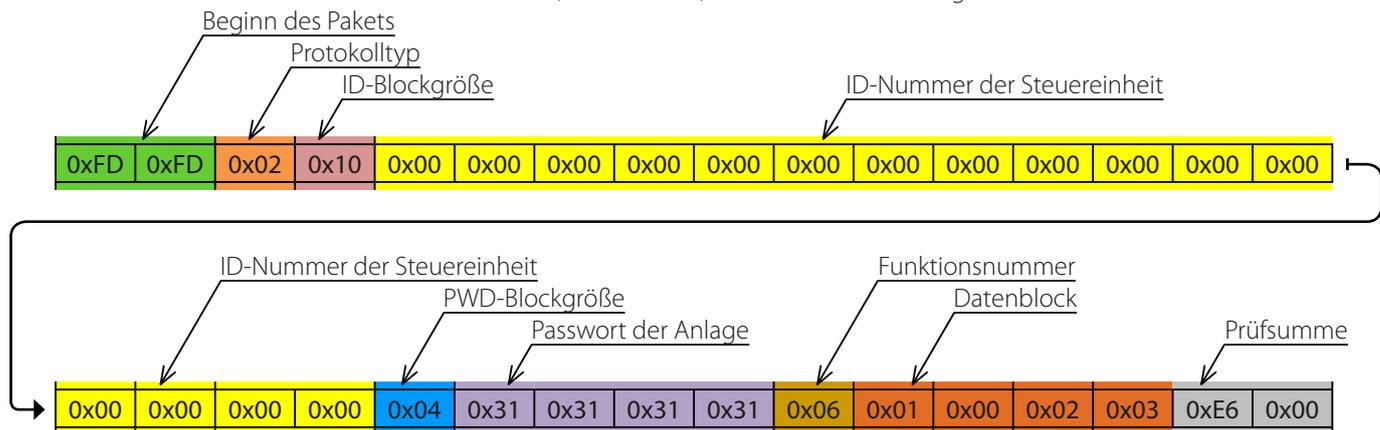


Anfragedetails:

- Prüfsumme: 0x00DE.

### Senden des Pakets „Steuereinheit -> Smart House“

Dieses Paket enthält die Antwort der Steuereinheit (FUNC = 0x06) zur Schreibanforderung.



Antwort der Steuereinheit:

- Parameter 0x0001 entspricht 0x00.
- Parameter 0x0002 entspricht 0x03.
- Prüfsumme: 0x00E6.

## PARAMETERTABELLE

Funktionen:		R – 0x01	INC – 0x04	RW – 0x03	W – 0x02	DEC – 0x05
Parameter- Nummer, Dez./Hex.	Funktionen	Beschreibung	Mögliche Werte	Größe, Bytes		
1/0x0001	R/W/RW	Anlage Ein/Aus	0: Aus 1: Ein 2: invertieren	1		
2/0x0002	R/W/RW/INC/DEC	Nummer der Lüftungsstufe	1: Lüftungsstufe 1 2: Lüftungsstufe 2 3: Lüftungsstufe 3 255: Betrieb der manuellen Einstellung der Lüftungsstufe (siehe Parameter 68)	1		
6/0x0006	R	Boost-Betriebstatus	0: Aus 1: Ein	1		
7/0x0007	R/W/RW/INC/DEC	Timerbetrieb (siehe Parameter 770 und 771)	0: Aus 1: Nachtbetrieb 2: Partybetrieb	1		
11/0x000B	R	Aktueller Countdown des Timerbetriebs	Byte 1: Sekunden (0...59) Byte 2: Minuten (0...59) Byte 3: Stunden (0...23)	3		
15/0x000F	R/W/RW	Aktivierung des Feuchtigkeitssensors	0: Aus 1: Ein 2: invertieren	1		
20/0x0014	R/W/RW	Aktivierung des Relais-Sensors	0: Aus 1: Ein 2: invertieren	1		
22/0x0016	R/W/RW	Aktivierung des Sensors 0–10 V*	0: Aus 1: Ein 2: invertieren	1		
25/0x0019	R/W/RW/INC/DEC	Feuchtesollwert	40...80 RH%	1		
36/0x0024	R	Aktuelle RTC-Batteriespannung	0...5000 mV	2		
37/0x0025	R	Aktuelle Feuchte	0...100 RH%	1		
45/0x002D	R	Aktueller Signalwert des Sensors 0–10 V*	0...100 %	1		
50/0x0032	R	Aktueller Wert des Relais-Sensors	0: Aus 1: Ein	1		
58/0x003A	R/W/RW/INC/DEC	Lüftungsstufe des Zuluftventilators in der 1. Stufe**	10...255	1		
59/0x003B	R/W/RW/INC/DEC	Lüftungsstufe des Abluftventilators in der 1. Stufe**	10...255	1		
60/0x003C	R/W/RW/INC/DEC	Lüftungsstufe des Zuluftventilators in der 2. Stufe**	10...255	1		
61/0x003D	R/W/RW/INC/DEC	Lüftungsstufe des Abluftventilators in der 2. Stufe**	10...255	1		
62/0x003E	R/W/RW/INC/DEC	Lüftungsstufe des Zuluftventilators in der 3. Stufe**	10...255	1		
63/0x003F	R/W/RW/INC/DEC	Lüftungsstufe des Abluftventilators in der 3. Stufe**	10...255	1		
68/0x0044	R/W/RW/INC/DEC	Lüftungsstufe des Ventilators im Betrieb der manuellen Einstellung der Lüftungsstufe	0...255	1		
74/0x004A	R	Lüftungsstufe des Ventilators Nr. 1	0...5000 Drehzahl	2		
75/0x004B	R	Lüftungsstufe des Ventilators Nr. 2	0...5000 Drehzahl	2		
99/0x0063	R/W/RW/INC/DEC	Einstellung der Zeit für den Filterwechsel**	70...365 Tage	2		
100/0x0064	R	Countdown des Timers bis zum Filterwechsel	Byte 1: Minuten (0...59) Byte 2: Stunden (0...23) Byte 3: Tage (0...181)	3		

Parameter-Nummer, Dez./Hex.	Funktionen	Beschreibung	Mögliche Werte	Größe, Bytes
101/0x0065	W	Countdown des Timers bis zum Filterwechsel zurücksetzen	Jedes Byte	1
102/0x0066	R/W/RW/INC/DEC	Sollwert der Nachlaufzeit des Boostbetriebs	0...60 Minuten	1
111/0x006F	R/W/RW	RTC-Zeit	Byte 1: RTC-Sekunden (0...59) Byte 2: RTC-Minuten (0...59) Byte 3: RTC-Stunden (0...23)	3
112/0x0070	R/W/RW	RTC-Kalender	Byte 1: RTC-Datum (1...31) Byte 2: RTC-Wochentag (1...7) Byte 3: RTC-Monat (1...12) Byte 4: RTC-Jahr (0...99)	4
114/0x0072	R/W/RW	Zeitgesteuerter Betrieb	0: Aus 1: Ein 2: invertieren	1
119/0x0077	R/W/RW	<p>Zeitplan-Einstellung</p> <p>Verwenden Sie in der Leseanforderung den Spezialbefehl 0xFE und geben Sie die Größe des Parameterwerts 0x02 an, der für die Auswahl des erforderlichen Wochentags und der Zeitraumnummer erforderlich ist.</p> <p>Die Schreibanforderung und die Steuerungsantwort verwenden alle 6 Bytes.</p> <p>Der erste Zeitraum beginnt immer um 00:00 Uhr, während der Beginn jedes folgenden Zeitraums am Ende des vorherigen beginnt.</p> <p>Der letzte Zeitraum endet immer um 24:00.</p>	<p><b>Byte 1:</b> Wochentag: 0: alle Tage (write only) 1: Montag 2: Dienstag 3: Mittwoch 4: Donnerstag 5: Freitag 6: Samstag 7: Sonntag 8: Mo...Fr (write only) 9: Sa...So (write only)</p> <p><b>Byte 2:</b> Periodennummer: 1...4</p> <p><b>Byte 3:</b> Nummer der Lüftungsstufe: 0: standby 1...3</p> <p><b>Byte 4:</b> reserviert Jedes Byte</p> <p><b>Byte 5:</b> Minuten bis zum Ende der Periode: 0...59</p> <p><b>Byte 6:</b> Stunden bis zum Ende der Periode: 0...23</p>	6
124/0x007C	R	Suche der Anlagen im Netzwerk, ID	Text („0...9“, „A...F“)	16
125/0x007D	R/W/RW	Passwort der Anlage	Text („0...9“, „a...z“, „A...Z“)	0-8
126/0x007E	R	Betriebsstunden	Byte 1: Minuten (0...59) Byte 2: Stunden (0...23) Byte 3 und 4: Tage (0...65535)	4
128/0x0080	W	Alarmer zurücksetzen	Jedes Byte	1
131/0x0083	R	Alarm/Warnung-Anzeige	0: Nein 1: Alarm (höchste Priorität) 2: Warnung	1
133/0x0085	R/W/RW	Zulassung des Betriebs über Cloud-Server	0: Aus 1: Ein 2: invertieren	1
134/0x0086	R	Firmware-Version und Datum der Firmware-Version der Steuereinheit	Byte 1: Firmware-Version (major) Byte 2: Firmware-Version (minor) Byte 3: Tag Byte 4: Monat Byte 5 und 6: Jahr	6

Parameter-Nummer, Dez./Hex.	Funktionen	Beschreibung	Mögliche Werte	Größe, Bytes
135/0x0087	W	Werkseinstellungen zurücksetzen	Jedes Byte	1
136/0x0088	R	Filterwechselanzeige	0: Filterwechsel wird nicht benötigt 1: Filterwechsel	1
148/0x0094	R/W/RW/INC/DEC	WLAN-Betrieb	1: Client 2: Access Point	1
149/0x0095	R/W/RW	WLAN-Name im Client-Betrieb	Text	1...32
150/0x0096	R/W/RW	WLAN-Passwort	Text	8...64
153/0x0099	R/W/RW	WLAN-Verschlüsselungstechnologie	48: OPEN 50: WPA_PSK 51: WPA2_PSK 52: WPA_WPA2_PSK	1
154/0x009A	R/W/RW/INC/DEC	WLAN-Kanalfrequenz	1...13	1
155/0x009B	R/W/RW	WLAN-Modul DHCP	0: STATIC 1: DHCP 2: invertieren	1
156/0x009C	R/W/RW	Zugewiesene IP-Adresse des WLAN-Moduls	<b>Byte 1:</b> 0...255, <b>Byte 2:</b> 0...255, <b>Byte 3:</b> 0...255, <b>Byte 4:</b> 0...255	4
157/0x009D	R/W/RW	Subnetzmaske des WLAN-Moduls	<b>Byte 1:</b> 0...255, <b>Byte 2:</b> 0...255, <b>Byte 3:</b> 0...255, <b>Byte 4:</b> 0...255	4
158/0x009E	R/W/RW	Haupt-Gateway des WLAN-Moduls	<b>Byte 1:</b> 0...255, <b>Byte 2:</b> 0...255, <b>Byte 3:</b> 0...255, <b>Byte 4:</b> 0...255	4
160/0x00A0	W	WLAN-Parameter übernehmen und Setup-Modus verlassen	Jedes Byte	1
162/0x00A2	W	Setup-Modus ohne Verwendung der neuen Parameter verlassen	Jedes Byte	1
163/0x00A3	R	Aktuelle IP-Adresse des WLAN-Moduls	<b>Byte 1:</b> 0...255, <b>Byte 2:</b> 0...255, <b>Byte 3:</b> 0...255, <b>Byte 4:</b> 0...255	4
183/0x00B7	R/W/RW/INC/DEC	Betriebsart des Ventilators	0: Lüftung 1: Wärmerückgewinnung 2: Zuluft	1
184/0x00B8	R/W/RW/INC/DEC	Sollwert des Sensors 0–10 V*	5...100 %	1
185/0x00B9	R	Anlagentyp	3: TwinFresh Expert RW1-50 V.2 TwinFresh Expert RW1-85 V.2 TwinFresh Expert RW1-100 V.2 4: TwinFresh Expert Duo RW1-30 V.2 5: TwinFresh Expert RW-30 V.2	2
252/0x00FC	<b>Spezialbefehle</b>			
253/0x00FD				
254/0x00FE				
255/0x00FF				
770/0x0302	R/W/RW	Sollwert des Timers für den Nachtbetrieb	Byte 1: Minuten (0...59), Byte 2: Stunden (0...23)	2
771/0x0303	R/W/RW	Sollwert des Timers für den Partybetrieb	Byte 1: Minuten (0...59), Byte 2: Stunden (0...23)	2
772/0x0304	R	Feuchtigkeitssensorzustand	0: unter dem Sollwert 1: über dem Sollwert	1
773/0x0305	R	Zustand des 0–10 V* Sensors	0: unter dem Sollwert 1: über dem Sollwert	1

\* Verfügbar für alle Modelle außer TwinFresh Expert RV-30 V.2

\*\* Verfügbar für TwinFresh Expert RV1-50 V.3, TwinFresh Style Wi-Fi, TwinFresh Style Wi-Fi Frost, TwinFresh Style Wi-Fi mini

## BEISPIEL DER PAKETVERARBEITUNG, IN C GESCHRIEBEN

```
//===== Spezialbefehle =====//
#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-Nummer der Steuereinheit

//***** Anfang der Prüfsumme und Beginn des Pakets *****//
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) // Prüfsumme
    {
        if(receive_data[2] == 0x02) // Protokolltyp
        {
            if(memcmp(&receive_data[4], current_id, receive_data[3]) == 0) // ID-Nummer
            {
                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 im Array, wo FUNC-Block startet
                //***** FUNC und DATA *****//
                for(; r_pos < receive_data_size - 2; r_pos++)
                {
                    //===== Spezialbefehle =====//
                    param_size = 1;
                    //=== Neue Funktionsnummer
                    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) // Wenn die Funktionsnummer nicht unterstützt wird
                            break;
                        continue;
                    }
                    //=== Neuer High-Byte-Wert für Parameternummer
                    else if(receive_data[r_pos] == BGCP_CMD_PAGE)
                    {

```

```
    page = receive_data[++r_pos];
    continue;
}
//=== neuer Wert der Parametergröße
else if(receive_data[r_pos] == BGCP_CMD_SIZE)
{
    param_size = receive_data[++r_pos];
    r_pos++;
}
//=== falls der Parameter nicht unterstützt wird
else if(receive_data[r_pos] == BGCP_CMD_NOT_SUP)
{
    r_pos++;
    //***** Verarbeitung der nicht unterstützten Parameter *****/
    param = (uint16_t)(page << 8) | (uint16_t)(receive_data[r_pos]);
    switch(param)
    {
        case 0x0001:
            break;
        case 0x0002:
            break;
        ...
    }
    //*****//
    continue;
}
jump_size = param_size;
//=====//

//***** Verarbeitung der unterstützten Parameter *****/
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;
}
//*****//
}
}
}
```

