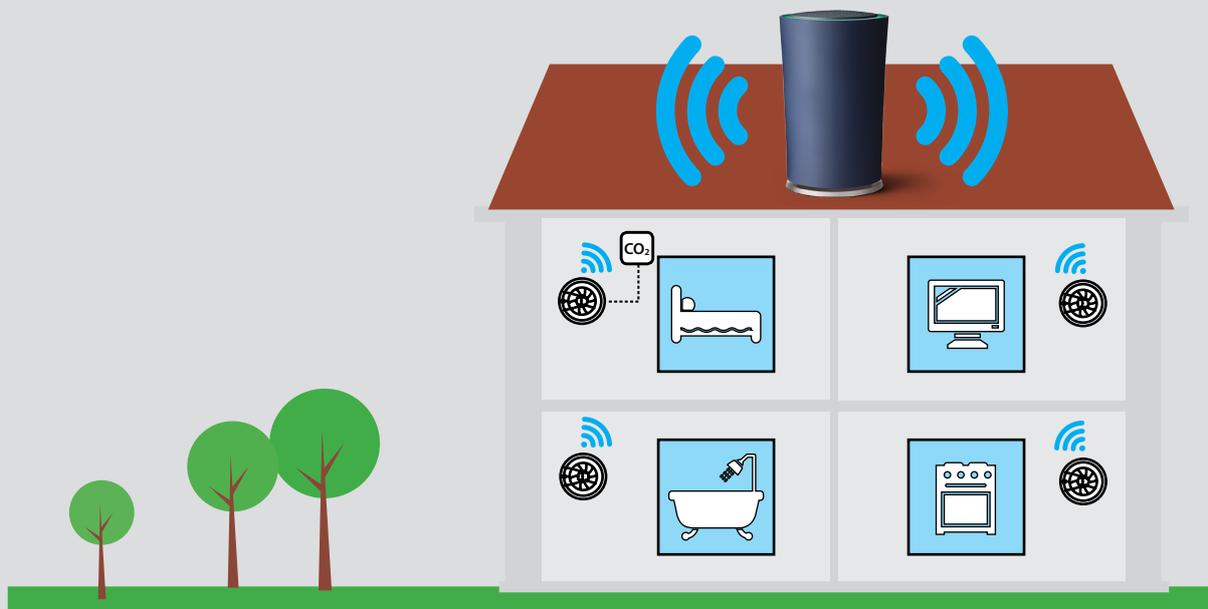


## Arc Smart

# Smart House



Verbindung mit dem Smart House-System

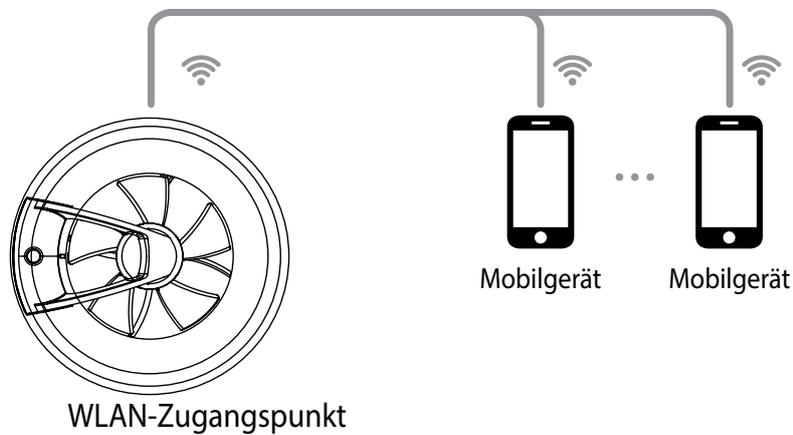
---

## INHALT

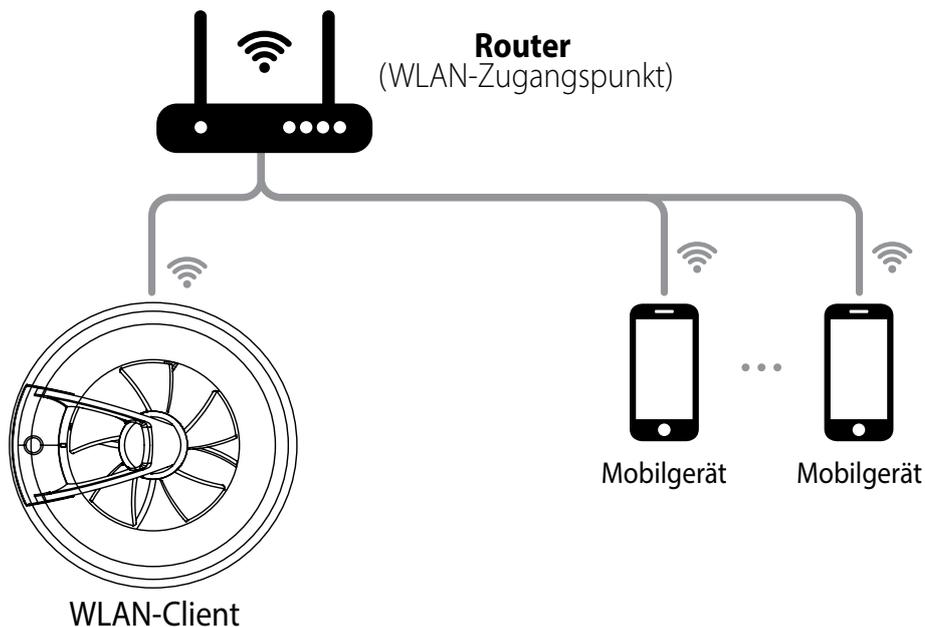
Verbindung und Einstellung .....	2
Netzwerkparameter .....	3
Paketstruktur .....	4
Beispiele für die Verwendung spezieller Befehle im Datenblock .....	5
Beispiele für ein vollständiges Paket .....	6
Tabelle der Parameter .....	7
Beispiel für Paketverarbeitung in C .....	10

## VERBINDUNG UND EINSTELLUNG

**Beispiel 1:** Schema für die direkte Verbindung des Ventilators mit dem BMS Smart House-System ohne Router. Konfigurieren Sie den Ventilator für den Betrieb über WLAN als Zugangspunkt (siehe Betriebsanleitung des Ventilators). Hinweis: Es können maximal acht Steuergeräte angeschlossen werden.



**Beispiel 2:** Schema für die Verbindung mit einem Router mit einem WLAN-Zugangspunkt. Der Ventilator, die Smartphones und das BMS Smart House-System werden mit dem WLAN-Zugangspunkt des Routers verbunden.



## NETZWERKPARAMETER

Der Datenaustausch erfolgt über das Transportprotokoll UDP (Broadcasting wird unterstützt).

IP-Adresse des Geräts:

- 192.168.4.1 – wenn das Gerät ohne Router betrieben wird (Verbindungsschema Nr. 1);
- wenn das Gerät mit einem Router verbunden ist (Verbindungsschema Nr. 2), wird die IP-Adresse über die Smartphone-App konfiguriert (siehe Betriebsanleitung des Geräts) und kann als statisch oder dynamisch (DHCP) eingestellt werden.

Der Geräteport ist 4000.

Die maximale Paketgröße beträgt 256 Byte.

## PAKETSTRUKTUR

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

**0xFD** **0xFD** – Paketanfang (2 Bytes).

**TYPE** – Protokolltyp (1 Byte). Der Wert ist 0x02.

**SIZE ID** – ID-Blockgröße (1 Byte). Der Wert ist 0x10.

**ID** – Steuereinheit-**ID**.  
Diese Nummer befindet sich auf einem Aufkleber (dargestellt als 16 Zeichen), der auf der Steuerplatine oder dem Gehäuse des Geräts angebracht ist.  
Es ist auch möglich, das Codewort „DEFAULT\_DEVICEID“ als ID-Nummer zu verwenden.  
Es kann verwendet werden:  
– zur Steuerung des Geräts, wenn es ohne Router betrieben wird (Verbindungsbeispiel Nr. 1);  
– zur Suche nach Geräten im Netzwerk, wenn ein Router verwendet wird (Verbindungsbeispiel Nr. 2);  
Gleichzeitig reagiert das Gerät nur auf zwei Parameter: 0x007C und 0x00B9 (siehe Parametertabelle).

**SIZE PWD** – PWD-Blockgröße (1 Byte). Möglicher Wertebereich: von 0x00 bis 0x08.

**PWD** – Gerätepasswort (gültige Zeichen sind: „0...9“, „a...z“, „A...Z“). Das Standardpasswort ist 1111.

Dieses Passwort kann im Einstellungsmenü der Smartphone-App geändert werden.

**FUNC** – Funktionsnummer (1 Byte). Bestimmt die mit den Daten auszuführende Aktion und die Struktur des **DATA**-Blocks:  
0x01 – Lesen von Parametern;  
0x02 – Schreiben von Parametern. Die Steuereinheit antwortet nicht auf die Anfrage nach dem Status dieser Parameter;  
0x03 – Schreiben von Parametern mit der anschließenden Antwort der Steuereinheit über den Status der angegebenen Parameter;  
0x04 – Inkrementieren von Parametern mit der anschließenden Antwort der Steuereinheit über den Status der angegebenen Parameter;  
0x05 – Dekrementieren von Parametern mit der anschließenden Antwort der Steuereinheit über den Status der angegebenen Parameter;  
0x06 – Antwort der Steuereinheit auf die Anfrage (FUNC = 0x01, 0x03, 0x04, 0x05).

**DATA** – Datenblock. Es besteht aus Parameternummern und deren Werten:

wenn *FUNC* = 0x01 oder 0x04 oder 0x05:

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

wenn *FUNC* = 0x02 oder 0x03 oder 0x06:

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

Die Parameternummern (siehe Parametertabelle) bestehen grob aus zwei Bytes (das High-Byte ist virtuell). Standardmäßig ist das High-Byte jeder Parameternummer in jedem neuen Paket 0x00. Das High-Byte kann innerhalb desselben Pakets mit dem Spezialbefehl 0xFF (siehe unten) geändert werden.

**P** – das Low-Byte der Parameternummer. Der mögliche Wertebereich: 0x00 – 0xFB.

Die Werte 0xFC – 0xFF sind spezielle Befehle:

**0xFC** – Änderung der Funktionsnummer (FUNC). Das nächste Byte muss eine neue Funktionsnummer von 0x01 bis 0x05 sein. Es wird verwendet, um mehrere Funktionen mit verschiedenen Aktionen in einem einzigen Paket zu organisieren;

**0xFD** – der Parameter wird von der Steuereinheit nicht unterstützt.  
Das nächste Byte ist das Low-Byte des nicht unterstützten Parameters. Es wird verwendet, wenn die Steuereinheit auf eine Anforderung zum Lesen oder Schreiben eines nicht vorhandenen Parameters antwortet (FUNC = 0x06);

**0xFE** – die Größe des Parameterwerts Value für einen nachfolgenden Parameter zu ändern. Das nächste Byte muss die neue Parametergröße sein, gefolgt von dem niederwertigen Byte der Parameternummer und dann dem Wert Value selbst;

**0xFF** – das High-Byte für Parameternummern innerhalb desselben Pakets zu ändern.  
Das nächste Byte muss das neue High-Byte sein.

**Value** – den Wert des Parameters (standardmäßig 1 Byte). Die Byte-Reihenfolge ist von niedrig nach hoch.

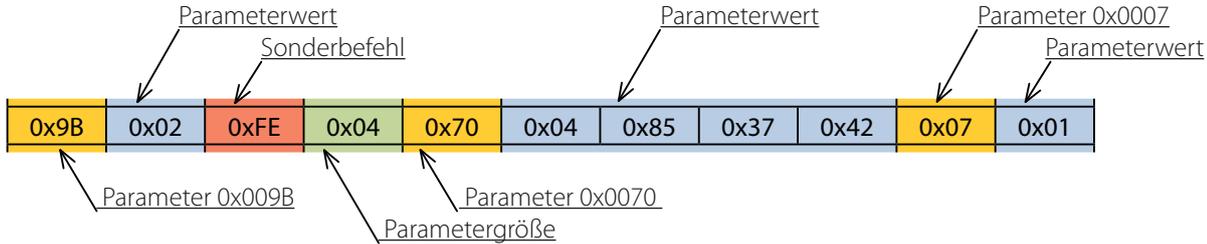
**Chksum L** **Chksum H** – Prüfsumme (2 Bytes). Sie wird als Summe der Bytes berechnet, beginnend mit dem TYPE-Byte und endend mit dem letzten Byte des **DATA**-Blocks.

**Checksum L** ist das niedrigste Byte der Prüfsumme.

**Checksum H** ist das höchste Byte der Prüfsumme.

## BEISPIELE FÜR DIE VERWENDUNG SPEZIELLER BEFEHLE IM DATENBLOCK

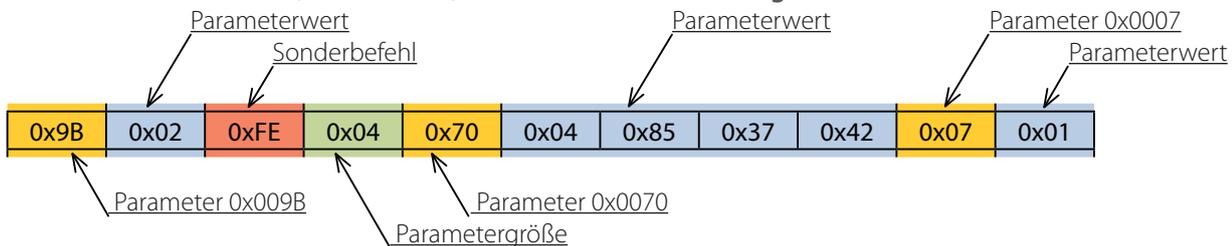
### Anforderung zum Schreiben (FUNC = 0x03) Parameter 0x009B, 0x0070, 0x0007



In der Schreibanforderung:

- Dem Parameter 0x02 den Wert 0x009B zuweisen.
- Dem Parameter 0x42378504 den Wert 0x0070 zuweisen. Die Wertgröße beträgt 4 Bytes, wie durch den Sonderbefehl 0xFE + 0x04 angegeben.
- Dem Parameter 0x01 den Wert 0x0007 zuweisen.

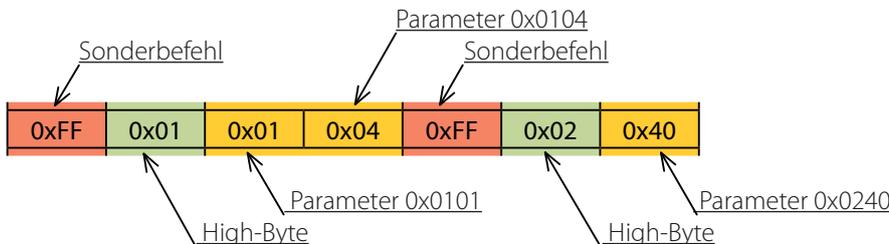
### Die Antwort der Steuereinheit (FUNC = 0x06) auf die Schreibanforderung



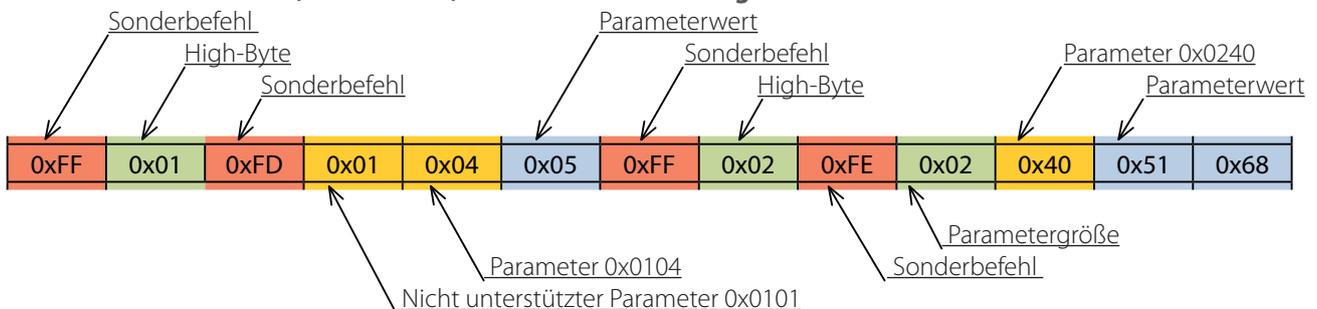
In der Antwort der Steuereinheit:

- Parameter 0x009B hat den Wert 0x02.
- Parameter 0x0070 hat den Wert 0x42378504.
- Parameter 0x0007 hat den Wert 0x01.

### Leseanforderung (FUNC = 0x01) der Parameter 0x0101, 0x0104, 0x0240



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



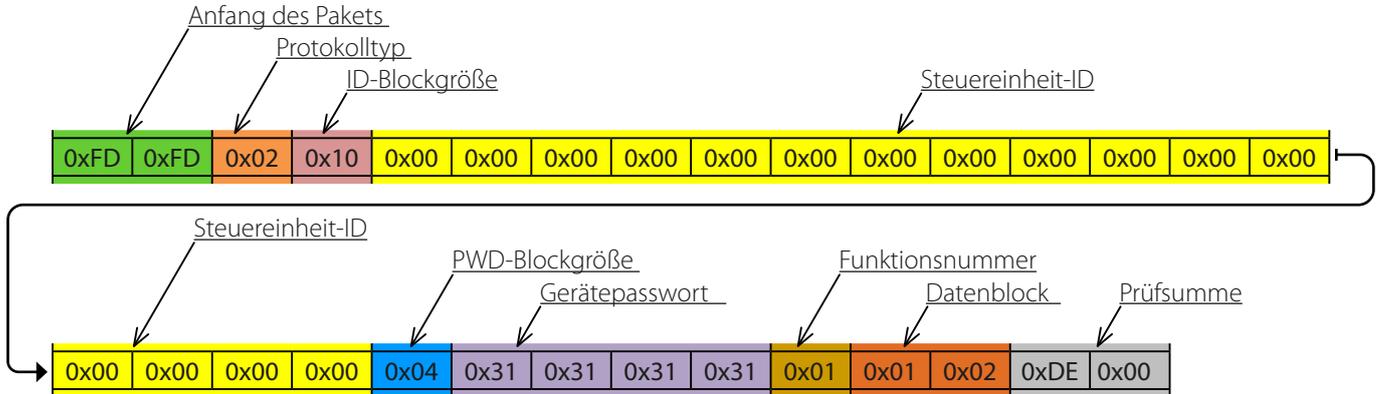
In der Antwort der Steuereinheit:

- Der Parameter 0x0101 wird von der Steuereinheit nicht unterstützt. Dies wird durch den Sonderbefehl 0xFD angezeigt.
- Parameter 0x0104 hat den Wert 0x05.
- Parameter 0x0240 hat den Wert 0x6851. Die Wertgröße beträgt 2 Bytes, wie durch den Sonderbefehl 0xFE + 0x02 angezeigt.

## BEISPIELE FÜR EIN VOLLSTÄNDIGES PAKET

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

Dieses Paket stellt eine Leseanforderung (FUNC = 0x01) für die folgenden Parameter dar: 0x0001, 0x0002.

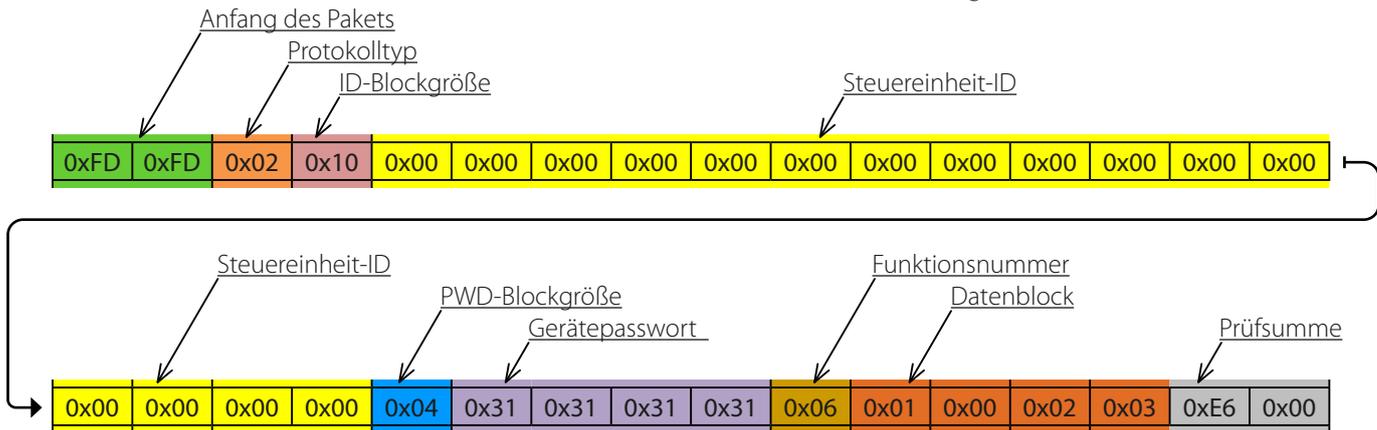


Die Anforderung umfasst:

- Prüfsumme: 0x00DE.

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

Dieses Paket enthält die Antwort der Steuereinheit (FUNC = 0x06) auf eine Leseanforderung.



In der Antwort der Steuereinheit:

- Parameter 0x0001 hat den Wert 0x00.
- Parameter 0x0002 hat den Wert 0x03.
- Prüfsumme: 0x00E6.

**TABELLE DER PARAMETER**

Funktionen:		R – 0x01	INC – 0x04	RW – 0x03	W – 0x02	DEC – 0x05
Parameter- nummer, Dec./Hex.	Funktionen	Beschreibung	Möglicher Wertebereich	Größe, Bytes		
6/0x0006	R/W/RW	Boost-Modus	0: AUS 1: EIN 2: invertieren	1		
7/0x0007	R	Anzeige für den Betrieb des Nachlaufschalters	0: AUS 1: EIN	1		
11/0x000B	R	Aktuelle Countdown-Zeit des Boost-Timers	1. Byte: Sekunden (0...59) 2. Byte: Minuten (0...59) 3. Byte: Stunden (0...23)	3		
15/0x000F	R/W/RW/INC/DEC	Steuerung durch Feuchtigkeitssensor	0: AUS 1: automatisch 2: manuelle Einstellung (siehe Parameter 25)	1		
25/0x0019	R/W/RW/INC/DEC	Einstellung des Feuchtigkeitssollwerts	40...80 RH %	1		
33/0x0021	R	Aktuelle Raumtemperatur	- 32768: kein Sensor + 32767: Kurzschluss	signed 2 (muss durch 10 geteilt werden, eine Dezimalstelle)		
36/0x0024	R	Aktuelle RTC-Batteriespannung	0...5000 mV	2		
37/0x0025	R	Aktuelle Raumfeuchtigkeit	0...100 RH %	1		
75/0x004B	R	Drehzahl des Ventilators	0...5000 U/min	2		
102/0x0066	R/W/RW/INC/DEC	Einstellung der Nachlaufzeit	0...60 min	1		
111/0x006F	R/W/RW	RTC-Zeit	1. Byte: Sekunden (0...59) 2. Byte: Minuten (0...59) 3. Byte: Stunden (0...23)	3		
124/0x007C	R	Suche nach Geräten im lokalen Ethernet-Netzwerk. Die Antwort enthält die Geräte-ID-Nummer.	Text ("0...9", "A...F")	16		
125/0x007D	R/W/RW	Gerätepasswort	Text ("0...9", "a...z", "A...Z")	0-8		
131/0x0083	R	Warnanzeige für schwache Batterie	0: AUS 1: EIN	1		
133/0x0085	R/W/RW	Erlaubnis, über einen Cloud-Server zu arbeiten	0: AUS 1: EIN 2: invertieren	1		
134/0x0086	R	Version und Datum der Hauptfirmware der Steuereinheit	1. Byte: Firmwareversion (major) 2. Byte: Firmwareversion (minor) 3. Byte: Tag 4. Byte: Monat 5., 6. Byte: Jahr	6		
135/0x0087	W	Alle Parameter auf Werkseinstellungen zurücksetzen	1: ausführen	1		
148/0x0094	R/W/RW/INC/DEC	WLAN- Betriebsmodus	1: Client 2: Zugangspunkt	1		
149/0x0095	R/W/RW	WLAN-Name im Client-Modus	Text	1 ... 32		
150/0x0096	R/W/RW	WLAN-Passwort	Text	8 ... 64		
153/0x0099	R/W/RW	Typ der WLAN-Datenverschlüsselung	48: OPEN 50: WPA_PSK 51: WPA2_PSK 52: WPA_WPA2_PSK	1		
154/0x009A	R/W/RW/INC/DEC	WLAN-Frequenzkanal	1 ... 13	1		

Parameter- nummer, Dec./Hex.	Funktionen	Beschreibung	Möglicher Wertebereich	Größe, Bytes
155/0x009B	R/W/RW	DHCP des WLAN-Moduls	0: STATIC 1: DHCP 2: invertieren	1
156/0x009C	R/W/RW	Die angegebene IP-Adresse des WLAN-Moduls	1: Byte: 0...255 2: Byte: 0...255 3: Byte: 0...255 4: Byte: 0...255	4
157/0x009D	R/W/RW	Subnetzmaske des WLAN-Moduls	1: Byte: 0...255 2: Byte: 0...255 3: Byte: 0...255 4: Byte: 0...255	4
158/0x009E	R/W/RW	Standardgateway des WLAN-Moduls	1: Byte: 0...255 2: Byte: 0...255 3: Byte: 0...255 4: Byte: 0...255	4
160/0x00A0	W	Übernehmen Sie neue WLAN-Einstellungen und verlassen Sie den Einstellungsmodus des WLAN-Moduls	1: ausführen	1
162/0x00A2	W	Verlassen Sie den Einstellungsmodus des WLAN-Moduls, ohne neue WLAN-Einstellungen anzuwenden	1: ausführen	1
163/0x00A3	R	Aktuelle IP-Adresse des WLAN-Moduls	1: Byte: 0...255 2: Byte: 0...255 3: Byte: 0...255 4: Byte: 0...255	4
185/0x00B9	R	Gerätetyp	<b>0: 65535</b> <b>13: Arc Smart</b>	2
252/0x00FC	<b>Sonderbefehle</b>			
253/0x00FD				
254/0x00FE				
255/0x00FF				
772/0x0304	R	Indikator für übermäßige Luftfeuchtigkeit	0: AUS 1: EIN	1
781/0x030D	R/W/RW	Modus 24	0: AUS 1: EIN 2: invertieren	1
782/0x030E	R	Auslöseanzeige des Lichtsensors	0: AUS 1: EIN	1
783/0x030F	R	Auslöseanzeige des Bewegungssensors	0: AUS 1: EIN	1
784/0x0310	R	Auslöseanzeige der Intervalllüftung	0: AUS 1: EIN	1
785/0x0311	R	Anzeige für den Betrieb im Silent-Modus	0: AUS 1: EIN	1
786/0x0312	R	Anzeige für die Verschlechterung der Luftqualität	0: AUS 1: EIN	1
787/0x0313	R/W/RW	Steuerung durch Lichtsensor	0: AUS 1: EIN 2: invertieren	1
788/0x0314	R/W/RW	Steuerung durch Bewegungssensor	0: AUS 1: EIN 2: invertieren	1

Parameter- nummer, Dec./Hex.	Funktionen	Beschreibung	Möglicher Wertebereich	Größe, Bytes
789/0x0315	R/W/RW/INC/DEC	Steuerung durch Luftqualitätssensor	0: AUS 1: automatisch 2: manuelle Einstellung (siehe Parameter 799)	1
790/0x0316	R/W/RW	Intervalllüftungsmodus	0: AUS 1: EIN 2: invertieren	1
791/0x0317	R/W/RW	Silent-Modus	0: AUS 1: EIN 2: invertieren	1
792/0x0318	R/W/RW	Beginn des Silent-Modus	1. Byte: Sekunden (0...59) 2. Byte: Minuten (0...59) 3. Byte: Stunden (0...23)	3
793/0x0319	R/W/RW	Ende des Silent-Modus	1. Byte: Sekunden (0...59) 2. Byte: Minuten (0...59) 3. Byte: Stunden (0...23)	3
794/0x031A	R/W/RW	Luftstrom bei Auslösung des RH-Sensors	3: 60 m <sup>3</sup> /h 4: 90 m <sup>3</sup> /h 5: 115 m <sup>3</sup> /h	1
795/0x031B	R/W/RW	Luftstrom bei Auslösung des Bewegungs-/Lichtensors	2: 40 m <sup>3</sup> /h 3: 60 m <sup>3</sup> /h 4: 90 m <sup>3</sup> /h 5: 115 m <sup>3</sup> /h	1
796/0x031C	R/W/RW	Luftstrom bei Auslösung des Luftqualitätssensors	3: 60 m <sup>3</sup> /h 4: 90 m <sup>3</sup> /h 5: 115 m <sup>3</sup> /h	1
797/0x031D	R/W/RW	Luftstrom während der Intervalllüftung	1: 20 m <sup>3</sup> /h 2: 40 m <sup>3</sup> /h 3: 60 m <sup>3</sup> /h	1
798/0x031E	R/W/RW	Luftstrom im 24-Stunden-Modus	1: 20 m <sup>3</sup> /h 2: 40 m <sup>3</sup> /h 3: 60 m <sup>3</sup> /h	1
799/0x031F	R/W/RW/INC/DEC	Sollwert des Luftqualitätssensors	50: 500 Index	2
800/0x0320	R	Aktuelles Luftqualitätsniveau	0: 500 Index	2
803/0x0323	R	Übertemperaturanzeige	0: AUS 1: EIN	1
804/0x0324	R/W/RW	Steuerung des Temperatursensors	0: AUS 1: EIN 2: invertieren	1
805/0x0325	R/W/RW/INC/DEC	Sollwert des Temperatursensors	18 – 36 °C	1
815/0x032F	R/W/RW	Luftstrom bei Auslösung des Temperatursensors	3: 60 m <sup>3</sup> /h 4: 90 m <sup>3</sup> /h 5: 115 m <sup>3</sup> /h	1

## BEISPIEL FÜR PAKETVERARBEITUNG IN C

```
//===== 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"; // Steuereinheit-ID

//***** Prüfen der Prüfsumme und Starten 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
            {
                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, an der der FUNC-Block beginnt
                //***** FUNC i 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) // falls die Funktionsnummer nicht unterstützt wird
                            break;
                        continue;
                    }
                    //=== Neuer Wert des High-Bytes für Parameternummern
                    else if(receive_data[r_pos] == BGCP_CMD_PAGE)
                    {

```

```
        page = receive_data[++r_pos];
        continue;
    }
    //=== neuer Parametergrößenwert
    else if(receive_data[r_pos] == BGCP_CMD_SIZE)
    {
        param_size = receive_data[++r_pos];
        r_pos++;
    }
    //=== wenn der Parameter nicht unterstützt wird
    else if(receive_data[r_pos] == BGCP_CMD_NOT_SUP)
    {
        r_pos++;
        //***** Verarbeitung nicht unterstützter 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 unterstützter 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;
}
//*****//
}
}
}
```



**VENTS**

