



Das nachfolgende Schema ist gültig für ALLE MD Beleuchtungsgeräte (aerolight, aerolightV2, smartlight, LaserLight). Das aerolightV2 benutzt bis zu acht Kanäle, bei allen anderen Geräten kann ein Dummy (beliebiger Wert) für den Kanal gesendet werden. Lichtmuster ist gerätebedingt unterschiedlich. Nicht jedes Gerät unterstützt immer alle Modi. Der Kanal setzt sich je Bit für einen Kanal zusammen. Bspw. Sie wollen die Kanäle 1,2,6 ansteuern, so müssen Sie nach binärem Werteschema $1+2+32 = 35$ als Wert für die Kanäle welche angesprochen werden sollen, übermitteln. Dabei unterscheidet sich der Wert je nach Übertragungsformat (siehe unten). Die Werte innerhalb des enum sind als ASCII (Text, String) zu übermitteln.

Alle Werten sind als Text/ASCII zu übermitteln, jedoch ist zu beachten,

- dass bei TCP/UDP der Kanal (bei Geräten mit nur 1 Kanal ist ein Dummywert zu setzen) (numerisch) als roher 8-Bitwert (0-255) übermitteln werden muss. Oftmals gibt es Funktionen (Chr, ChrW) die den numerischen Wert in Text wandeln. Der Kanalwert muss IMMER übermitteln werden, auch dort wo er nicht benötigt wird (bspw. POWER, FARBWERTE)
- dass bei HTTP Befehlen (PUT, GET) IMMER der Kanalwert übermitteln werden muss (ggf. einfach ein Dummywert). ALLE Zahlenwerte müssen als ASCII (Format HEX) übermitteln werden muss. Dabei ist bspw. der Wert 35_{DEC} eine 23_{HEX} . Wenn bspw. alle Kanäle angesprochen werden sollen (255) ist per HTTP FF zu senden. Außerdem muss eine Kennzeichnung (+DST:) vorgeschaltet werden sowie ein `\n\n` zum Schluss (nur bei HTTP PUT).

Das allg. gültige Protokoll lautet wie folgt (bspw. orange, Kanal 1,2,6 soll angeschaltet werden):
Dabei entspricht ein ASCII Zeichen einem String (Text) mit nur einem Zeichen!

TCP/UDP: Sende wie folgt ASCII 8 DEC 35 (oftmals ist der Sendestring so: "8" & Chr(35))

HTTP GET: Sende wie folgt: URL:PORT/?823, /? gibt einen GET-REQUEST an. (8 = Kommandowert, 23 = Hex-Wert für 35 für Kanalwert)

HTTP PUT: Sende wie folgt STRING "+DST:823\n\n" (8 = Kommandowert, 23 = Hex-Wert für 35 für Kanalwert)

Sie wollen senden: blau, Kanal 1 (HTTP PUT: "+DST:301\n\n"), blau, alle Kanäle (dec = 255) in HTTP PUT wäre das "+DST:3FF\n\n". Das „\n\n“ ist ein doppelter Zeilenumbruch und nur bei HTTP erforderlich.

ACHTUNG: BEI HTTP MUSS IMMER (!) EIN KANAL ANGEGEBEN WERDEN! BEI ALLEN BEFEHLEN ÜBER HTTP IST EIN KANALWERT ERFORDERLICH!!!! NUR BEI HTTP ERFORDERLICH!!!

Es wird auf allen Ebenen mit dem entsprechenden Wert geantwortet. Falls kein Rückgabewert vorhanden ist, wird HTTP 204 (No Content) gesendet. Bei einem falschen Steuerbefehl erscheint HTTP 501 (Not Implemented) als Antwort.

Die IP finden Sie über Ihren Router oder mittels unserer App. Der Port bei allen MD Beleuchtungsgeräten lautet: 2020 bei TCP/http und 4040 bei UDP. Hierüber lässt sich außerdem der interne Gerätenamen sowie die IP abrufen. Außerdem kann die IP ebenso über den UDP-Befehl „Y“ bestimmt werden. Hiermit senden alle Geräte Ihren Namen und Ihre IP sowie Netmask. Ansprechen nachfolgendem Schema: `http://192.168.xxx.yyy:2020` bei TCP/HTTP.



```
enum _receiveData_enum
```

```
{  
    POWER = '0',  
    SMOOTH = '1',  
    RED = '2',  
    BLUE = '3',  
    GREEN = '4',  
    ROSE = '5',  
    BLOODORANGE = '6',  
    YELLOW = '7',  
    ORANGE = '8',  
    CYAN = '9',  
  
    LIGHTBLUE = 'a',  
    LILA = 'b',  
    WARMWHITE = 'c',  
    WHITE = 'd',  
    LIMETTE = 'e',  
    VEILCHEN = 'f',  
    SMARTGREEN = 'g',  
    ICEBLUE = 'h',  
    SENSORS = 'i', //Sensor is Lightsensor, Touchsensor on/off  
    RESET = 'j',  
    NEXT = 'k', //next color if smooth mode active  
    SMOOTHTIME = 'l', //l + time value (e.g. 30 = 30 minutes)  
    TIME_MODE = 'm', //only Smartlight!!!  
    LIMOGELB = 'n',  
    VERSION = 'z',  
  
    EXTCOLORRGB = 'A', //Protokoll: A + RGB-Values  
    STROBE_FADE = 'B', //Switch between strobe/fade mode  
    FADE_TIMEUP = 'C', //Fade time increase  
    FADE_TIMEDOWN = 'D', //Fade time decrease  
    MANBRIGHTNESS = 'E', //only if sensor off, Protokoll: E + DimmVal  
    BONUSCOLOR1 = 'F',  
    BONUSCOLOR2 = 'G',  
    BONUSCOLOR3 = 'H',  
    BONUSCOLOR1SAVE = 'I', //actual color (e.g. from EXTCOLORRGB) saved  
    BONUSCOLOR2SAVE = 'J',  
    BONUSCOLOR3SAVE = 'K',  
    NETWORK_REGISTRATION = 'L', //Protokoll can be requested!!  
    GETSATEKOORS = 'M', //koors of the bonuscolors 1-3 to center in app  
    STARTSLEEPTIMER = 'N', //N + time value (e.g. 30 = 30 mins) + 0/1*1  
    WECKFUNKTION = 'O', //O + Weckcolor (e.g. Blue = 3) + hour + min  
    LICHTMUSTER = 'P', //P+(0-2 as ASCII in TCP and HTTP)  
    ShowTimeDateWeather = 'R', //Get all web informations  
    TIMERCONTROL = 'v', //v + Weckcolor + hourON + minON + hourOFF + minOFF  
    CHANGE_TIME_MODE = 'S', //only Smartlight!!! Protocoll requestable  
    MANVOLUME = 'T', //only smartlight, Protokoll: T + 8 bit Values  
    MUTE_NONmute_AUDIO = 'V', //only smartlight, act/deact muting of sound  
    NETWORK_NAME_CHANGINGS = 'X', //Neuer Gerätename nach X, Umlaute ersetzen!  
    GET_IP_NAME = 'Y', //DHCP IP und Gerätename abfragen  
    GET_DATA_STATE = 'Z', //Got Data: Value 1 = Wecker active?,  
                                     Value 2 = Strobe mode active?,  
                                     Value 3 = Sensor active?  
                                     Value 4 = Power on?  
                                     Value 5 = Lichtmuster active?  
                                     Value 6 = Timer active?  
                                     Value 7 = Sleeptimer active?  
  
    Bei Smartlight kommen hier alle  
    Werte für die Zeiteinfädung sowie  
    die dazugehörigen Farben als  
    decimale Wert.
```



```
Value 8 = Smoothzeit  
Value 9 = Weckkanal  
Value 10 = Weckfarbe  
Value 11 = Weckstunde  
Value 12 = Weckminute  
Value 13 = Timerfarbe  
Value 14 = Timer Stunde an  
Value 15 = Timer Minute an  
Value 16 = Timer Stunde aus  
Value 17 = Timer Minute aus
```

```
ShowCurrenColorBright = 'W' //Get current RGB+Brightness levels from device  
                        //R-Value,G-Value,B-Value,Brightness;...(next channel)  
                        //sending each channel values  
                        //if smartlight, R-Value,G-Value,B-Value,Brightness,Volume  
                        //Example (e.g. aerolight): 125,255,0,50  
  
TOGGLE_STATUS_LED =   '*', //Activate/Deactivate Status LED (if existing) for Session  
}receiveData;
```

```
*1 0 = single timer (one hit), 1 = always timer (always active and reload by sending  
new values (0/1 always as text/ASCII))
```