



**mediola**<sup>®</sup>  
a.i.o. gateway

## Technische Dokumentation

Kommunikation & Kommandos

**Stand:**

September 2014

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>7</b>
<b>2</b>	<b>Ansteuerung .....</b>	<b>7</b>
<b>3</b>	<b>Parameter .....</b>	<b>8</b>
3.1	Infrarotcodes und Funkbefehle im mediola-Format senden .....	8
3.2	Systemcodes senden .....	9
3.1.1	intertechno® Funk-Komponenten .....	9
3.1.2	FS20 Funk-Komponenten .....	10
<b>4</b>	<b>Actions.....</b>	<b>13</b>
4.1	AddAction .....	13
4.2	DelAction .....	15
4.3	GetAll.....	15
4.4	DoAction .....	15
4.5	Reset.....	15
<b>5</b>	<b>Tasks .....</b>	<b>16</b>
5.1	AddTask .....	16
5.2	DelTask .....	16
5.3	GetAll.....	17
5.4	Reset.....	17

<b>6</b>	<b>Events .....</b>	<b>18</b>
6.1	AddEvent .....	18
6.2	DelEvent .....	20
6.3	GetAll.....	20
6.4	Reset.....	20
<b>7</b>	<b>Groups .....</b>	<b>21</b>
7.1	AddGroup .....	21
7.2	DelGroup .....	22
7.3	GetAll.....	22
7.4	Reset.....	22
7.5	ToggleGroup .....	22
7.6	Group aktivieren/deaktivieren .....	22
7.7	doGroup.....	22
<b>8</b>	<b>API Dokumentationen .....</b>	<b>23</b>
8.1	Bidirektionale Systeme.....	23
8.1.1	HomeMatic API .....	23
8.1.1.1	Komponente anlernen .....	23
8.1.1.2	Komponente steuern .....	24
8.1.1.3	Schaltaktor anmelden.....	24
8.1.1.4	Wandthermostat anmelden .....	26
8.1.1.5	Wetterstation anmelden .....	39
8.1.1.6	Wassersensor anmelden .....	41
8.1.1.7	Jalousieaktor konfigurieren.....	41

8.1.1.8	Sprachmodul Signalleuchte konfigurieren .....	41
8.1.1.9	Sprachmodul Sprachausgabe konfigurieren .....	42
8.1.1.10	Energiemesssteckdose .....	43
8.1.1.11	HomeMatic-Gerätetypen .....	44
8.1.1.12	HomeMatic-Befehle .....	46
<b>8.2</b>	<b>LED Controller I .....</b>	<b>48</b>
8.2.1	LED Controller I anlernen .....	48
8.2.2	LED Controller I steuern .....	48
<b>8.3</b>	<b>LED Controller II .....</b>	<b>50</b>
8.3.1	LED Controller II anlernen .....	50
8.3.2	LED Controller II steuern .....	50
<b>8.4</b>	<b>FHT80B API .....</b>	<b>52</b>
8.4.1	Add FHT80B .....	52
8.4.2	Delete FHT80B .....	52
8.4.3	Zustand verändern .....	53
8.4.4	Zustand abfragen .....	53
8.4.5	Reset .....	54
<b>8.5</b>	<b>Variablen .....</b>	<b>55</b>
8.5.1	SetVar .....	55
8.5.2	DelVar .....	56
8.5.3	GetVar .....	56
<b>8.6</b>	<b>Sensoren .....</b>	<b>57</b>
8.6.1	AddSensor .....	57
8.6.2	DelSensor .....	57

8.6.3	GetStates .....	57
8.7	Wettersensor .....	58
8.7.1	Wettersensor Innen WS300-S300TH .....	58
8.7.2	Wettersensor Aussen WS300-KS300 .....	58
8.8	Energie-Meter .....	59
8.8.1	AddMeter.....	59
8.8.2	DelMeter.....	59
8.8.3	GetStates .....	59
9	UDP.....	60

© 2014 – mediola – connected living AG

Alle Rechte vorbehalten.

Das Werk einschließlich all seiner Teile ist urheberrechtlich geschützt. Wird dieses Handbuch zusammen mit Software ausgeliefert, die einen Lizenzvertrag enthält, so werden sowohl dieses Handbuch als auch die Software unter Lizenz bereitgestellt und dürfen ausschließlich gemäß diesem Lizenzvertrag verwendet werden.

Dies gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, die Einspeicherung sowie das Bereithalten zur Nutzung in elektronischen Systemen.

Der Inhalt dieses Handbuchs dient ausschließlich Informationszwecken, kann ohne Vorankündigung geändert werden und ist nicht als Verpflichtung seitens tecnovum anzusehen. Die tecnovum AG gibt keine Gewähr oder Garantie hinsichtlich der Richtigkeit und Genauigkeit der Angaben in diesem Handbuch. Obwohl grundsätzlich bei der Zusammenstellung von Texten und Abbildungen mit größter Sorgfalt vorgegangen wurde, können Fehler und Irrtümer nicht vollständig ausgeschlossen werden. Die mediola AG kann für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch eine Haftung übernehmen.

In Beispielen verwendete Namen dienen ausschließlich Demonstrationszwecken. Übereinstimmungen mit tatsächlichen Unternehmen und Organisationen sind rein zufällig.

mediola und das mediola Logo sind eingetragene Marken der mediola AG. iPhone, iPod Touch und iPad sind eingetragene Marken von Apple Inc. Windows, Mediacenter Edition, Vista sind eingetragenen Marken der Microsoft Corporation. Android ist eine Marke von Google Inc. Alle anderen in der Dokumentation verwendeten Soft- und Hardwarebezeichnungen sowie Markennamen unterliegen im allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz der jeweiligen Firmen. Dieser Schutz wird hiermit anerkannt.

Stand: September 2014

mediola – connected living AG  
Rennbahnstr. 72 - 74  
60528 Frankfurt am Main

Tel.: 069-96870-324

Fax: 069-96870-326

Internet: [www.mediola.de](http://www.mediola.de)

Email: [info@mediola.de](mailto:info@mediola.de)

## 1 Einleitung

Ein mediola® AIO GATEWAY (Version V2/V3/V4) dient als Bindeglied zwischen der App AIO REMOTE und den zu steuernden Funkkomponenten mit 433Mhz und 868M MHz sowie nahezu beliebigen Infrarot (IR) Geräten. Der offen Systemarchitektur von mediola® folgend, basiert die Kommunikation daher auf HTTP-Befehlen. Somit lassen sich die mediola® AIO GATEWAYS auch in beliebig anderen Anwendungen einbinden.

Wie dies funktioniert, beschreibt das vorliegende Dokument, welches primär eine technische Dokumentation darstellt. Insofern richtet es sich vorwiegend an Anwender mit entsprechendem technischen Hintergrund.

## 2 Ansteuerung

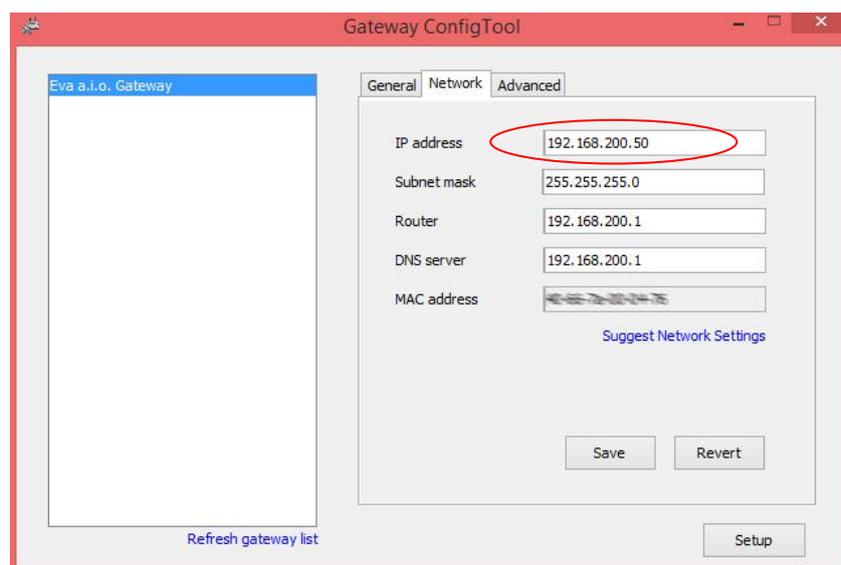
Das Absetzen eines Befehls basiert auf folgender Syntax:

<http://IP-Adresse-des-Gateways/command?PARAMETER>

bzw. (falls ein User-Passwort verwendet wird)

[http://IP-Adresse-des-Gateways/command?XC\\_USER=user&XC\\_PASS=PASSWORD&PARAMETER](http://IP-Adresse-des-Gateways/command?XC_USER=user&XC_PASS=PASSWORD&PARAMETER)

Die IP-Adresse des AIO GATEWAY lässt sich u.a. mit dem „Gateway-ConfigTool“ konfigurieren bzw. herausfinden. Dieses Tool<sup>1</sup> steht im Download-Bereich auf der mediola-Webseite<sup>1</sup> kostenlos zur Verfügung.



<sup>1</sup> <http://mediola.com/dokumente.html>

<b>PASSWORD</b>	Das vom Nutzer gewählte Passwort (optional, wenn gesetzt)
<b>IP</b>	Die IPv4-Adresse des Gateways

Ein möglicher Aufruf könnte wie folgt aussehen (Statusabfrage):

[http://192.168.200.50/command?XC\\_FNC=GetStates](http://192.168.200.50/command?XC_FNC=GetStates)

### 3 Parameter

Der Parameter beschreibt den eigentlichen Befehl, der an das a.i.o gateway gesendet werden soll. Mit Hilfe dieser Parameter lassen sich IR-Codes und Funkbefehle versenden sowie Tasks und Events verwalten.

#### 3.1 Infrarotcodes und Funkbefehle im mediola-Format senden

<b>Parameter</b>	Send2
<b>Syntax</b>	<code>code=&lt;MEDIOLA_CODE&gt;&amp;XC_FNC=Send2[&amp;ir=&lt;IR_ID&gt;&amp;rf=00]</code>



Hinweis:

Wichtig beim Absetzen des Befehls ist die genaue Einhaltung der Parameter, andernfalls wird das a.i.o. gateway nicht reagieren.

<b>MEDIOLA_CODE</b>	Der abzusetzende Code im mediola®-Format. Diese beginnen immer mit 1908 - das ältere mediola-Format endet auf <b>001908!</b>
<b>IR_ID</b>	Für die optional ansteuerbaren externe Infrarot-Diode gilt: wird keine ID angegeben, werden automatisch die internen Sendedioden verwendet. Ansonsten erfolgt die Steuerung wie folgt: <ul style="list-style-type: none"> <li><b>01</b> - Interne Sendedioden</li> <li><b>02</b> - Externe Sendediode 1</li> <li><b>04</b> - Externe Sendediode 2</li> <li><b>08</b> - Externe Sendediode 3</li> </ul>

**Beispiele:**

[http://192.168.200.50/command?code=003700...34001908&XC\\_FNC=Send2](http://192.168.200.50/command?code=003700...34001908&XC_FNC=Send2)

Gleicher Befehl, aber zur Ansteuerung über die externe Sendediode Nr. 2:

[http://192.168.200.50/command?code=003700...34001908&XC\\_FNC=Send2&ir=04&rf=00](http://192.168.200.50/command?code=003700...34001908&XC_FNC=Send2&ir=04&rf=00)

### 3.2 Systemcodes senden

<b>Parameter</b>	SendSC
<b>Syntax</b>	XC_FNC=SendSC&type=<SYSTEM>&data=<CODE>

Mit Systemcodes (SC) können die verschiedenen Geräte adressiert werden, die direkt vom „a.i.o. gateway“ unterstützt werden.

<b>SYSTEM</b>	Das Identifikationskürzel des entsprechenden Gerätes z. B.:  <b>IT</b> - Intertechno® Funk-Komponenten <sup>2</sup>  <b>FS20</b> - FS20® Funk-Komponenten <sup>3</sup>
<b>CODE</b>	Der eigentliche Systemcode (siehe Details für „type“ in den entsprechenden Abschnitten).

#### 3.1.1 intertechno® Funk-Komponenten

<b>type</b>	IT
<b>data</b>	<family_code><device_code><action>

Der Systemcode setzt sich aus einem „Familien-Code“, dem individuellen „Geräte-Code“ und der gewünschten Aktion (Ein-/Ausschalten) zusammen.

<sup>2</sup> Intertechno®: <http://www.intertechno.at>  
<sup>3</sup> FS20®: <http://www.eq-3.de>

<b>family_code</b>	Der Familiencode besteht aus einem hexadezimalen Zeichen (0-9, A-F). 0 steht dabei für A, 1 für B usw. Bei einer intertechno <sup>®</sup> Steckdose kann man z.B. den Familiencode auf der Rückseite über einen Drehschalter von A-D einstellen.
<b>device_code</b>	Der individuelle Gerätecode lässt sich innerhalb einer Gruppe einstellen und besteht nur aus Zahlen (1-x). Auch dieser wird als hexadezimaler Zeichen (0-9, A-F) angegeben. Zu beachten ist, dass der Code bei 0 und nicht bei 1 beginnt, der eigentliche Gerätecode also mit -1 angegeben werden muss.
<b>action</b>	Die eigentliche Aktion die durchgeführt werden soll. Folgende Zeichen sind dabei gültig: <ul style="list-style-type: none"> <li><b>E</b> - Gerät einschalten</li> <li><b>6</b> - Gerät ausschalten</li> </ul>

**Beispiele:**

[http://192.168.200.50/command?XC\\_FNC=SendSC&type=IT&data=20E](http://192.168.200.50/command?XC_FNC=SendSC&type=IT&data=20E)

Schaltet eine Steckdose mit dem Familien-Code **C** (2) und dem individuellen Gerätecode **1** (0) **ein** (E).

[http://192.168.200.50/command?XC\\_FNC=SendSC&type=IT&data=046](http://192.168.200.50/command?XC_FNC=SendSC&type=IT&data=046)

Schaltet eine Steckdose mit dem Familien-Code **A** (0) und dem individuellen Gerätecode **5** (4) **aus** (6).

**3.1.2 FS20 Funk-Komponenten**

<b>type</b>	FS20
<b>data</b>	<house_code_1+2><device_code><action>

Der Systemcode setzt sich aus einem „Hauscode“, bestehend aus zwei Teilblöcken, dem individuellen „Geräte-Code“ und der gewünschten Aktion (Ein-/Ausschalten, Dimmen, Timer setzen, ...) zusammen.

<b>house_code</b>	<p>Der Hauscode eines FS20-Systems besteht aus zwei Teilblöcken (HC1 und HC2). Dieser muss in eine hexadezimale Zahl umgewandelt werden, was nach folgender Logik durchzuführen ist (Beispiel: HC1=1122, HC2=3344):</p> <ol style="list-style-type: none"> <li>1. HC1 und HC2 werden zusammengefasst 11223344</li> <li>2. Jede Stelle wird um eins (1) reduziert 00112233</li> <li>3. Die entstandene Zahl wird in Zweierblöcke unterteilt 00 11 22 33</li> <li>4. Die jeweils erste Zahl eines Blocks wird mit 4 multipliziert und mit der zweiten Zahl addiert 00 5 10 15</li> <li>5. Jeder Block wird nun in seine Hexadezimaldarstellung überführt (0-9, A-F) 0 5 A F = <b>05AF</b></li> </ol>
<b>device_code</b>	<p>Der 4-Zeichen lange Gerätecode wird nach demselben Schema wie der Hauscode berechnet. Beachten Sie, dass die FS20-Adressen 44xx, xx44 und 4444 besondere Bedeutungen haben.</p>
<b>action</b>	<p>Die eigentliche Aktion die durchgeführt werden kann. Der einzusetzende Hexadezimalcode kann folgender Übersicht entnommen werden:</p> <ul style="list-style-type: none"> <li><b>0000</b> - Ausschalten</li> <li><b>0100</b> - Auf 6,25% dimmen</li> <li><b>0200</b> - Auf 12,50% dimmen (im Creator ~10%)</li> <li><b>0300</b> - Auf 18,75% dimmen (im Creator ~20%)</li> <li><b>0400</b> - Auf 25,00% dimmen</li> <li><b>0500</b> - Auf 31,25% dimmen (im Creator ~30%)</li> <li><b>0600</b> - Auf 37,50% dimmen (im Creator ~40%)</li> <li><b>0700</b> - Auf 43,75% dimmen</li> <li><b>0800</b> - Auf 50,00% dimmen (im Creator ~50%)</li> <li><b>0900</b> - Auf 59,25% dimmen (im Creator ~60%)</li> </ul>

	<b>0A00</b> - Auf 62,50% dimmen
	<b>0B00</b> - Auf 68,75% dimmen (im Creator ~70%)
	<b>0C00</b> - Auf 75,00% dimmen
	<b>0D00</b> - Auf 81,25% dimmen (im Creator ~80%)
	<b>0E00</b> - Auf 87,50% dimmen (im Creator ~90%)
	<b>0F00</b> - Auf 93,75% dimmen
	<b>1000</b> - Einschalten
	<b>1100</b> - Auf den letzten, gespeicherten Wert einschalten
	<b>1200</b> - Ein <-> Ausschalten (Toggle)
	<b>1300</b> - Hochdimmen
	<b>1400</b> - Runterdimmen

## 4 Actions

### 4.1 AddAction

Anlegen einer neuen Action:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=AddAction](http://{IP-Adresse-des-Gateways}/command?XC_FNC=AddAction)

&id	Hexadezimal (1-Byte)	<p>Der mediola® a.i.o. creator bestimmt die ID der Action.</p> <p>Sie wird nicht mehr vom a.i.o. gateway ermittelt.</p> <p>Die id muss != 0 sein !!!</p>
&type	Hexadezimal (1-Byte)	<p>Typ der Action</p> <p>FS20 = FS20 Befehl senden</p> <p>CODE = Beliebigen Funk- oder Infrarotcode senden</p> <p>IT = Intertechno Befehl senden</p> <p>FHT80B = Modus und/oder Soll-Temperatur eines Raumregler ändern</p> <p>EVENT = Event Gruppe aktivieren/deaktivieren/toggeln</p> <p>ONOFF = Variable vom Typ ONOFF verändern</p> <p>KOPP = Kopp Befehl senden</p> <p>HE = HomeEasy Befehl senden</p> <p>INSTA = Gira Funkkomponenten</p> <p>GROUP = eine Gruppe des Task- &amp; Eventmanagers aktivieren/deaktivieren</p>
&code	<p>„FS20“, „CODE“, „IT“,</p> <p>„HTTP“, „AIO“, „ELRO“,</p> <p>„FHT80B“, "KOPP",</p> <p>"INSTA",</p> <p>"HE", "GROUP"</p>	<p><b>FS20</b></p> <p>Ein FS20 Befehl besteht aus 5 Byte.</p> <p>Alle Werte Hexadezimal angeben!</p> <p>hc1 = Hauscode1</p> <p>hc2 = Hauscode2</p> <p>adr = Adresse</p> <p>cmd = Befehl</p>

		<p>ext = Befehls-erweiterung</p> <p>Intertechno</p> <p>Ein Intertechno (IT) Befehl besteht aus 3 Zeichen.</p> <p>F = Family-Code (von 0 bis F)</p> <p>S = Single-Code (von 0 bis F)</p> <p>C = Command-Code (Ein = E, Aus = 6)</p> <p><b>AIO</b></p> <p>MediolaServerMessage</p> <ol style="list-style-type: none"> <li>1. Byte = 0x01 (Mediola Message Type)</li> <li>2. Byte = 0x01 (Version vom Taskplaber auf dem Gateway)</li> <li>3. Byte = Trigger ID (vom Gateway)</li> <li>4. Byte = Action ID (vom Gateway)</li> </ol> <p><b>FHT80B</b></p> <p>Ein FHT80B „Befehl“ besteht aus 4 Byte.</p> <p>Alle Werte Hexadezimal angeben!</p> <ol style="list-style-type: none"> <li>1. Byte: hc1 = Code 1 des Raumreglers</li> <li>2. Byte: hc2 = Code 2 des Raumreglers</li> <li>3. Byte: m = Modus (00 = Manuell, 01 = Automatisch, FF = Modus nicht verändern)</li> <li>4. Byte: soll = Die Soll-Temperatur wird wie in der FHT80B API unter Punkt 3 (Zustand verändern) codiert.</li> </ol> <p>Wenn die Soll-Temperatur nicht verändert werden soll, muss der Wert auf FF gesetzt werden.</p> <p><b>ONOFF</b></p> <ol style="list-style-type: none"> <li>1. Byte = die ID der ONOFF Variable</li> <li>2. Byte = 00-&gt;OFF</li> </ol> <p style="padding-left: 40px;">01-&gt;ON</p> <p style="padding-left: 40px;">0F-&gt;TOGGLE</p>
--	--	--

&rf		Siehe Abschnitt 3.1
&ir		Siehe Abschnitt 3.1

## 4.2 DelAction

Löschen einer Action vom Gateway:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=DelAction&id=](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=DelAction&id=)

## 4.3 GetAll

Gibt alle Actions, Events, Tasks und Groups aus:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=GetAll](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=GetAll)

```
{"sys":"action","id":"02","type":"FS20","code":"123456","rf":"00","ir":"00"}
```

## 4.4 DoAction

Führt die Action mit der ID 1 aus. (zum Testen)

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=doAction&id=01](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=doAction&id=01)

## 4.5 Reset

Löscht alle gespeicherten Actions

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=fEEPReset&type=03](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=fEEPReset&type=03)

## 5 Tasks

### 5.1 AddTask

Einen Task im Gateway anlegen:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=AddTask](http://{IP-Adresse-des-Gateways}/command?XC_FNC=AddTask)

&id	Hexadezimal (1-Byte)	Der mediola® a.i.o creator bestimmt die ID der Action. Sie wird nicht mehr vom a.i.o. Gateway ermittelt. Die id muss != 0 sein !!! Ein Event darf nicht die gleiche ID haben wie ein Task und umgekehrt.
&active	0 oder 1	Immer 1
&days	0 0 0 0 0 0 Mo Di Mi Do Fr Sa So	7-stellige Binärzahl. Jedes der 7 Bit steht für einen Wochentag. Eine 1 bedeutet, der Task ist an dem Wochentag aktiv, eine 0 bedeutet, der ist an diesem Wochentag nicht aktiv. MSB = Montag LSB = Sonntag
&time	HH:MM	Stunde und Minuten Angabe der Zeit an dem der Task ausgeführt werden soll.
dateStart	YYYY-MM-DD	Wird zurückgesetzt, wenn &dateStart = 2000-00-00 übergeben wird.
&dateEnd	YYYY-MM-DD	Wird zurückgesetzt, wenn &dateEnd = 2000-00-00 übergeben wird.

### 5.2 DelTask

Löscht einen Task vom Gateway:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=DelTask&id](http://{IP-Adresse-des-Gateways}/command?XC_FNC=DelTask&id)

### 5.3 GetAll

Gibt alle Actions, Events und Tasks aus.

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=GetAll](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=GetAll)

```
{"sys":"task","id":"01","active":"1","days":"1111111","time":"12:00","dateStart":"2011-08-01","dateEnd":"2011-08-06"}
```

### 5.4 Reset

Löscht alle gespeicherten Tasks:

[http://{{IP-Adresse-des-Gatewas}}/command?XC\\_FNC=fEEPReset&type=01](http://{{IP-Adresse-des-Gatewas}}/command?XC_FNC=fEEPReset&type=01)

## 6 Events

### 6.1 AddEvent

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=AddEvent](http://{IP-Adresse-des-Gateways}/command?XC_FNC=AddEvent)

&id	Hexadezimal (1-Byte)	<p>Der mediola® a.i.o. creator bestimmt die ID der Action.</p> <p>Sie wird nicht mehr vom a.i.o. Gateway ermittelt</p> <p>Die id muss != 0 sein !!!</p> <p>Ein Event darf nicht die gleiche ID haben wie ein Task und umgekehrt.</p>
&active	0 oder 1	Immer 1
&type	<p>„FS20“, „IT“, "KOPP",</p> <p>„SIAU“, "SIAUG",</p> <p>"SIAUR",</p> <p>"HE"</p>	<p>Typ des Ereignisses, das zum auslösen des Events führt.</p> <p>SIAU = Siegenia-Aubi Fensterkontakt</p>
&code	<p>Bei FS20:</p> <p>hc1   hc2   adr   cmd   ext</p> <p>Bei IT:</p> <p>F   S   C</p> <p>Bei SIAU:</p> <p>adr   state</p>	<p><b>FS20</b></p> <p>Ein FS20 Befehl bei dem das Event ausgeführt werden soll besteht aus 5 Byte.</p> <p>Alle Werte Hexadezimal angeben!</p> <p>hc1 = Hauscode1</p> <p>hc2 = Hauscode2</p> <p>adr = Adresse</p> <p>cmd = Befehl</p> <p>ext = Befehlsweiterung (meistens 00)</p> <p><b>IT</b></p> <p>Ein Intertechno (IT) Befehl bei dem das Event ausgeführt werden soll besteht aus 3 Zeichen.</p> <p>F = Family-Code (von 0 bis F)</p> <p>S = Single-Code (von 0 bis F)</p> <p>C = Command-Code (Ein = E, Aus = 6)</p>

		<p><b>SIAU</b></p> <p>Alle Werte Hexadezimal angeben!</p> <p>adr = 3 Byte Adresse z.b. 003b7c (FFFFFF → egal welches Fenster)</p> <p>state = 1 Byte des Zustandes bei dem das Event ausgeführt werden soll (4 höherwertigen Bits - Status)</p> <p>2 → Fenster geschlossen</p> <p>1 → Fenster gekippt</p> <p>0 → Fenster geöffnet</p> <p>F → egal / nicht auswerten</p> <p>(4 niederwertigen Bits - Fehler)</p> <p>0 → kein Fehler</p> <p>1 → Batteriewarnung</p> <p>2 → Seit 130 Minuten keine (alive) Meldung des Fensterkontakts erhalten</p> <p>F → egal / nicht auswerten</p> <p><b>SIAUG</b></p> <p>Siegenia Aubi Glasbruchmelder</p> <p>Alle Werte Hexadezimal angeben!</p> <p>adr = 3 Byte Adresse z.b. 003b7c (FFFFFF → egal welches Fenster)</p> <p>state = 1 Byte des Zustandes bei dem das Event ausgeführt werden soll (4 höherwertigen Bits - Status)</p> <p>1 → Alarm</p> <p>F → egal / nicht auswerten (4 niederwertigen Bits - Fehler)</p> <p>0 → kein Fehler</p> <p>1 → Batteriewarnung</p> <p>2 → Seit 130 Minuten keine (alive) Meldung des Fensterkontakts erhalten</p> <p>F → egal / nicht auswerten</p> <p><b>SIAUR</b></p> <p>Siegenia Aubi Rauchmelder</p> <p>Alle Werte Hexadezimal angeben!</p>
--	--	---

		adr = 3 Byte Adresse z.b. 003b7c (FFFFFF → egal welches Fenster) state = 1 Byte des Zustandes bei dem das Event ausgeführt werden soll (4 höherwertigen Bits - Status) 1 → Alarm 0 → kein Alarm mehr F → egal / nicht auswerten (4 niederwertigen Bits - Fehler) 0 → kein Fehler 1 → Batteriewarnung 2 → Seit 130 Minuten keine (alive) Meldung des Fensterkontakts erhalten F → egal / nicht auswerten
--	--	---

## 6.2 DelEvent

Löscht einen Event vom Gateway:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=DelEvent](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=DelEvent)

## 6.3 GetAll

Gibt alle Actions, Events und Tasks aus:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=GetAll](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=GetAll)

```
{"sys":"event","id":"01","type":"FS20","active":"1","code":"123456"}
```

## 6.4 Reset

Löscht alle gespeicherten Events:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=fEEReset&type=02](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=fEEReset&type=02)

## 7 Groups

Über eine Group, werden Actions mit einem oder mehreren Auslöser (Event oder Task) verknüpft. Eine Group verfügt über eine Liste von Auslösern (TriggerIDs) und über eine Liste von Actions (ActionIDs). In der Liste TriggerIDs werden die IDs von Tasks oder Events gespeichert die eine Group "auslösen" können. Tritt ein Event oder ein Task zu einer bestimmten Uhrzeit ein, löst die Group aus und führt nacheinander die Actions in der Liste ActionIDs aus.

### 7.1 AddGroup

Anlegen einer neuen Group:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=AddGroup](http://{IP-Adresse-des-Gateways}/command?XC_FNC=AddGroup)

&id	Hexadezimal (1-Byte)	Der mediola® a.i.o. creator bestimmt die ID der Action. Sie wird nicht mehr vom a.i.o. gateway ermittelt. Die id muss != 0 sein !!!
&active	0 oder 1	0 = deaktiviert, 1 = aktiviert
&triggerIDs	Hexadezimal 1-Byte pro ID	Liste der IDs der Tasks/Events, bei denen die Liste der Actions ausgeführt werden soll. Angabe in Hexadezimal (1-Byte) Bsp: triggerids = 01020304A1 reagiert auf die Tasks/Events mit den IDs 1, 2, 3, 4 oder A1. Die IDs der Task und Events dürfen sich deshalb nicht überschneiden.
&actionIDs	Hexadezimal 1-Byte pro ID	Liste der ID der Actions, die ausgeführt werden sollen. Hexadezimal 1-Byte pro ID Bsp:actionids = 01020304A1 löst die Actions mit den Ids, 1, 2, 3, 4 und A1 aus

## 7.2 DelGroup

Löschen einer Group:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=DelGroup&id=](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=DelGroup&id=)

## 7.3 GetAll

Gibt alle Actions, Events, Tasks und Groups aus:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=GetAll](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=GetAll)

```
{"sys":"group","id":"01","active":"1","triggerids":"010203","actionids":"0102A1"}
```

## 7.4 Reset

Löscht alle gespeicherten Groups:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=fEEPReset&type=05](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=fEEPReset&type=05)

## 7.5 ToggleGroup

Wegen der Abwärtskompatibilität heißt der Befehl **toggleEvent**:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=toggleEvent&group=ID](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=toggleEvent&group=ID)

## 7.6 Group aktivieren/deaktivieren

Aktivieren/deaktivieren einer Gruppe:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=saveGroup&id=01&active=0](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=saveGroup&id=01&active=0)

## 7.7 doGroup

Alle Actions in der Liste ActionIDs werden ausgeführt! : (zu Testzwecken)

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=doGroup&id=01](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=doGroup&id=01)

## 8 API Dokumentationen

### 8.1 Bidirektionale Systeme

#### 8.1.1 HomeMatic API

##### 8.1.1.1 Komponente anlernen

Anlernen einer Komponente:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=learnSC&adr=HEQ0101981](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=learnSC&adr=HEQ0101981)

Bei erfolgreichem anmelden werden **{XC\_SUC}**, die Adresse des Schaltaktors und dessen Typ zurückgegeben:

```
{XC_SUC>{"adr":"130B99", "type":"0011"}
```

Anlernen einer batteriebetriebenen Komponente:

Das Lernen über die Seriennummer funktioniert in der Regel nicht mit batteriebetriebenen Komponenten.

Diese Komponenten müssen in den Anlernmodus gebracht werden, nach dem man den „learnSC“ Befehl ohne den Parameter „adr“ aufgerufen hat:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=learnSC](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=learnSC)

Bei erfolgreichem anmelden werden **{XC\_SUC}**, die Adresse der Komponente, die Seriennummer und dessen Typ zurückgegeben:

```
{XC_SUC}{\"adr\": \"1DE1E501\", \"snr\": \"JEQ0651128\", \"type\": \"0040\"}
```



Hinweis:

Angabe des Typs, erst ab Firmware Version 2.2.5

### 8.1.1.2 Komponente steuern

Die im Beispiel angelernte Steckdose lässt sich nun folgendermaßen steuern:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=SendSC&type=HM&data=130B990101](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=SendSC&type=HM&data=130B990101)

„data“ setzt sich folgendermaßen zusammen:

Adresse	Kanal	Befehl
130B99	01	01

Für weitere Geräte-Typen siehe Abschnitt 8.1.3.11 und für weitere Befehle siehe Abschnitt 8.1.3.12

### 8.1.1.3 Schaltaktor anmelden

Die HomeMatic Komponenten lassen sich auch als Sensoren anmelden, z.B. um ihren Status zu überwachen oder um die Daten von Wettersensoren auszuwerten.

Um den Status der Steckdose zu überwachen, muss diese folgendermaßen am Gateway angelernt werden:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=addSensor&type=HM&adr=130B9901](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=addSensor&type=HM&adr=130B9901)

„adr“ setzt sich folgendermaßen zusammen:

Adresse	Kanal
130B99	01

Verfügt eine Komponente über mehrere Kanäle, muss jeder Kanal einzeln angemeldet werden.

Den, auf dem Gateway gespeicherten, Status fragt man folgendermaßen ab:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=GetStates](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=GetStates)

Es wird ein JSON Array mit dem Status aller angemeldeten Geräte zurückgegeben:

```
{XC_SUC}[{"type":"HM","adr":"130b9901","state":"00:00"}]
```

„state“ setzt sich folgendermaßen zusammen:

Status	Trennzeichen	Letzte Meldung
00	:	00

Das Statusbyte wird je nach Gerätetyp unterschiedlich interpretiert. Es kann Werte zwischen 0x00 und 0xC8 (0 - 200) annehmen. Teilt man diesen Wert durch zwei erhält man den Prozentwert ( $0xC8 / 2 = 100,0\%$ ). Er gibt an zu wie viel Prozent ein Licht gedimmt ist, oder zu wieviel Prozent ein Rolladen heruntergefahren ist.

Ein Schaltaktor kann nur die Werte 0% oder 100% annehmen.

Das Byte „Letzte Meldung“ gibt die Zeit (in 5min Schritte) an, seit dem die letzten Daten empfangen wurden.

Meldet ein HomeMatic Gerät eine Statusänderung, sendet das Gateway eine UDP Nachricht über das Netzwerk an die Adresse 255.255.255.255 und Port-Nr. 1902.

Die Nutzdaten des UDP Pakets sehen folgendermaßen aus:

```
{XC_EVT}{"type":"HM","data":"130B9901C8"}
```

„data“ setzt sich folgendermaßen zusammen:

Adresse	Kanal	Status
130B99	01	C8

Status abfragen/aktualisieren eines bestimmten Gerätes über das Gateway:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=refreshhm&adr=130B9901](http://{IP-Adresse-des-Gateways}/command?XC_FNC=refreshhm&adr=130B9901)

Status abfragen/aktualisieren aller Geräte, welche am Gateway angemeldet sind:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=refreshhm](http://{IP-Adresse-des-Gateways}/command?XC_FNC=refreshhm)

Status Abfrage:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=GetStates](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=GetStates)

#### 8.1.1.4 Wandthermostat anmelden

Wandthermostet am Gateway anmelden:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=addSensor&type=HMFHT&adr=134FD601](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=addSensor&type=HMFHT&adr=134FD601)

Der 3-Byte Adresse „134FD6“ wird noch eine „01“ für den Kanal angehängt.

Status Abfrage:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=GetStates](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=GetStates)

Es wird ein JSON Array mit dem Status zurückgegeben:

```
{"type":"HMFHT","adr":"134FD601","state":"I:M:00D2:0107:35:64"}
```

„state“ setzt sich folgendermaßen zusammen:

Fehler	: Modus	: Soll-Temperatur	: Ist-Temperatur	: Luftfeuchtigkeit	: Ventilstellung
I	M	0x00D2	0x0107	0x35	0x64
0 = kein Fehler B = Batterie schwach I = noch keine Daten empfangen E = keine Funkverbindung	M = Manuell A = Auto B = Boost U = Urlaub	0x00D2 = 210 → 21,0°C	0x0107 = 263 → 26,3°C	0x35 = 53%	100%

Die Befehle zum Steuern des Wandthermostaten kann man der Tabelle von HomeMatic Befehlen entnehmen.

Beispiel:

Soll-Temperatur auf 21°C stellen:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=SendSC&type=HM&data=134FD601112A](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=SendSC&type=HM&data=134FD601112A)

Boost Modus aktivieren:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=SendSC&type=HM&data=134FD6011C03](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=SendSC&type=HM&data=134FD6011C03)

**Wochenprofil setzen**

Pro Tag sind maximal 13 unterschiedliche Zeitabschnitte mit jeweils eigenen Temperaturen einstellbar. Der erste Zeitabschnitt startet immer bei 0:00 Uhr. Der letzte Zeitabschnitt muss bei 0:00 Enden. Die Endzeit eines Zeitabschnitts ist dann gleichzeitig die Startzeit des folgenden Zeitabschnitts.

Beispiel für Samstag:

- 1. Zeitabschnitt 00:00 – 05:00    18.0°C
- 2. Zeitabschnitt 05:00 – 08:00    22.0°C
- 3. Zeitabschnitt 08:00 – 17:00    18.0°C
- 4. Zeitabschnitt 17:00 – 24:00    22.0°C

Register	Bedeutung	Wert
Samstag		
0x14 Bit 7 – 1	Temperatur 1. Zeitabschnitt	0x24 = 36 => 36 / 2 = 18°C
0x14 Bit 0	Bit 8 von Register 0x15	0

0x15	Endzeit 1. Zeitabschnitt	$0x03C = 60 \Rightarrow 60 * 5 = 300\text{min} \Rightarrow 05:00$
0x16 Bit 7 - 1	Temperatur 2. Zeitabschnitt	$0x29 = 44 \Rightarrow 44 / 2 \Rightarrow 22^{\circ}\text{C}$
0x16 Bit 0	Bit 8 von Register 0x17	0
0x17	Endzeit 2. Zeitabschnitt	$0x060 = 96 \Rightarrow 96 * 5 = 480\text{min} \Rightarrow 08:00$
0x18 Bit 7 - 1	Temperatur 3. Zeitabschnitt	$0x24 = 36 \Rightarrow 36 / 2 = 18^{\circ}\text{C}$
0x18 Bit 0	Bit 8 von Register 0x19	0
0x19	Endzeit 3. Zeitabschnitt	$0x0CC = 204$ $\Rightarrow 204 * 5 = 1020\text{min} \Rightarrow 17:00$
0x1A Bit 7 - 1	Temperatur 4. Zeitabschnitt	0x2C
0x1A Bit 0	Bit 8 von Register 0x1B	1
0x1B	Endzeit 4. Zeitabschnitt	$0x120 = 288 \Rightarrow 288 * 5 = 1440\text{min} \Rightarrow 24:00$
0x1C	Temperatur 5. Zeitabschnitt	
0x1D	Endzeit 5. Zeitabschnitt	
0x1E	Temperatur 6. Zeitabschnitt	
0x1F	Endzeit 6. Zeitabschnitt	
0x20	Temperatur 7. Zeitabschnitt	
0x21	Endzeit 7. Zeitabschnitt	

0x22	Temperatur 8. Zeitabschnitt	
0x23	Endzeit 8. Zeitabschnitt	
0x24	Temperatur 9. Zeitabschnitt	
0x25	Endzeit 9. Zeitabschnitt	
0x26	Temperatur 10. Zeitabschnitt	
0x27	Endzeit 10. Zeitabschnitt	
0x28	Temperatur 11. Zeitabschnitt	
0x29	Endzeit 11. Zeitabschnitt	
0x2A	Temperatur 12. Zeitabschnitt	
0x2B	Endzeit 12. Zeitabschnitt	
0x2C	Temperatur 13. Zeitabschnitt	
0x2D	Endzeit 13. Zeitabschnitt	

Beispiel:

Tagesprofil setzen für Samstag:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=SendSC&type=HM&data=134FD6011E1448153C16581760184819CC1A581B201C451D20](http://{IP-Adresse-des-Gateways}/command?XC_FNC=SendSC&type=HM&data=134FD6011E1448153C16581760184819CC1A581B201C451D20)

Wochenprofil abfragen:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=GetStates&config=1](http://{IP-Adresse-des-Gateways}/command?XC_FNC=GetStates&config=1)



0x2F	Endzeit 1. Zeitabschnitt
0x30	Temperatur 2. Zeitabschnitt
0x31	Endzeit 2. Zeitabschnitt
0x32	Temperatur 3. Zeitabschnitt
0x33	Endzeit 3. Zeitabschnitt
0x34	Temperatur 4. Zeitabschnitt
0x35	Endzeit 4. Zeitabschnitt
0x36	Temperatur 5. Zeitabschnitt
0x37	Endzeit 5. Zeitabschnitt
0x38	Temperatur 6. Zeitabschnitt
0x39	Endzeit 6. Zeitabschnitt
0x3A	Temperatur 7. Zeitabschnitt
0x3B	Endzeit 7. Zeitabschnitt
0x3C	Temperatur 8. Zeitabschnitt
0x3D	Endzeit 8. Zeitabschnitt
0x3E	Temperatur 9. Zeitabschnitt
0x3F	Endzeit 9. Zeitabschnitt
0x40	Temperatur 10. Zeitabschnitt
0x41	Endzeit 10. Zeitabschnitt
0x42	Temperatur 11. Zeitabschnitt
0x43	Endzeit 11. Zeitabschnitt

0x44	Temperatur 12. Zeitabschnitt
0x45	Endzeit 12. Zeitabschnitt
0x46	Temperatur 13. Zeitabschnitt
0x47	Endzeit 13. Zeitabschnitt

Register	Bedeutung
Montag	
0x48	Temperatur 1. Zeitabschnitt
0x49	Endzeit 1. Zeitabschnitt
0x4A	Temperatur 2. Zeitabschnitt
0x4B	Endzeit 2. Zeitabschnitt
0x4C	Temperatur 3. Zeitabschnitt
0x4D	Endzeit 3. Zeitabschnitt
0x4E	Temperatur 4. Zeitabschnitt
0x4F	Endzeit 4. Zeitabschnitt
0x50	Temperatur 5. Zeitabschnitt
0x51	Endzeit 5. Zeitabschnitt
0x52	Temperatur 6. Zeitabschnitt
0x53	Endzeit 6. Zeitabschnitt
0x54	Temperatur 7. Zeitabschnitt
0x55	Endzeit 7. Zeitabschnitt

0x56	Temperatur 8. Zeitabschnitt
0x57	Endzeit 8. Zeitabschnitt
0x58	Temperatur 9. Zeitabschnitt
0x59	Endzeit 9. Zeitabschnitt
0x5A	Temperatur 10. Zeitabschnitt
0x5B	Endzeit 10. Zeitabschnitt
0x5C	Temperatur 11. Zeitabschnitt
0x5D	Endzeit 11. Zeitabschnitt
0x5E	Temperatur 12. Zeitabschnitt
0x5F	Endzeit 12. Zeitabschnitt
0x60	Temperatur 13. Zeitabschnitt
0x61	Endzeit 13. Zeitabschnitt

Register	Bedeutung
Dienstag	
0x62	Temperatur 1. Zeitabschnitt
0x63	Endzeit 1. Zeitabschnitt
0x64	Temperatur 2. Zeitabschnitt
0x65	Endzeit 2. Zeitabschnitt
0x66	Temperatur 3. Zeitabschnitt
0x67	Endzeit 3. Zeitabschnitt

0x68	Temperatur 4. Zeitabschnitt
0x69	Endzeit 4. Zeitabschnitt
0x6A	Temperatur 5. Zeitabschnitt
0x6B	Endzeit 5. Zeitabschnitt
0x6C	Temperatur 6. Zeitabschnitt
0x6D	Endzeit 6. Zeitabschnitt
0x6E	Temperatur 7. Zeitabschnitt
0x6F	Endzeit 7. Zeitabschnitt
0x70	Temperatur 8. Zeitabschnitt
0x71	Endzeit 8. Zeitabschnitt
0x72	Temperatur 9. Zeitabschnitt
0x73	Endzeit 9. Zeitabschnitt
0x74	Temperatur 10. Zeitabschnitt
0x75	Endzeit 10. Zeitabschnitt
0x76	Temperatur 11 Zeitabschnitt
0x77	Endzeit 11. Zeitabschnitt
0x78	Temperatur 12. Zeitabschnitt
0x79	Endzeit 12. Zeitabschnitt
0x7A	Temperatur 13. Zeitabschnitt
0x7B	Endzeit 13. Zeitabschnitt

Register	Bedeutung
Mittwoch	
0x7C	Temperatur 1. Zeitabschnitt
0x7D	Endzeit 1. Zeitabschnitt
0x7E	Temperatur 2. Zeitabschnitt
0x7F	Endzeit 2. Zeitabschnitt
0x80	Temperatur 3. Zeitabschnitt
0x81	Endzeit 3. Zeitabschnitt
0x82	Temperatur 4. Zeitabschnitt
0x83	Endzeit 4. Zeitabschnitt
0x84	Temperatur 5. Zeitabschnitt
0x85	Endzeit 5. Zeitabschnitt
0x86	Temperatur 6. Zeitabschnitt
0x87	Endzeit 6. Zeitabschnitt
0x88	Temperatur 7. Zeitabschnitt
0x89	Endzeit 7. Zeitabschnitt
0x8A	Temperatur 8. Zeitabschnitt
0x8B	Endzeit 8. Zeitabschnitt
0x8C	Temperatur 9. Zeitabschnitt
0x8D	Endzeit 9. Zeitabschnitt
0x8E	Temperatur 10. Zeitabschnitt

0x8F	Endzeit 10. Zeitabschnitt
0x90	Temperatur 11 Zeitabschnitt
0x91	Endzeit 11. Zeitabschnitt
0x92	Temperatur 12. Zeitabschnitt
0x93	Endzeit 12. Zeitabschnitt
0x94	Temperatur 13. Zeitabschnitt
0x95	Endzeit 13. Zeitabschnitt

Register	Bedeutung
Donnerstag	
0x96	Temperatur 1. Zeitabschnitt
0x97	Endzeit 1. Zeitabschnitt
0x98	Temperatur 2. Zeitabschnitt
0x99	Endzeit 2. Zeitabschnitt
0x9A	Temperatur 3. Zeitabschnitt
0x9B	Endzeit 3. Zeitabschnitt
0x9C	Temperatur 4. Zeitabschnitt
0x9D	Endzeit 4. Zeitabschnitt
0x9E	Temperatur 5. Zeitabschnitt

0x9F	Endzeit 5. Zeitabschnitt
0xA0	Temperatur 6. Zeitabschnitt
0xA1	Endzeit 6. Zeitabschnitt
0xA2	Temperatur 7. Zeitabschnitt
0xA3	Endzeit 7. Zeitabschnitt
0xA4	Temperatur 8. Zeitabschnitt
0xA5	Endzeit 8. Zeitabschnitt
0xA6	Temperatur 9. Zeitabschnitt
0xA7	Endzeit 9. Zeitabschnitt
0xA8	Temperatur 10. Zeitabschnitt
0xA9	Endzeit 10. Zeitabschnitt
0xAA	Temperatur 11 Zeitabschnitt
0xAB	Endzeit 11. Zeitabschnitt
0xAC	Temperatur 12. Zeitabschnitt
0xAD	Endzeit 12. Zeitabschnitt
0xAE	Temperatur 13. Zeitabschnitt
0xAF	Endzeit 13. Zeitabschnitt

Register	Bedeutung
Freitag	
0xB0	Temperatur 1. Zeitabschnitt

0xB1	Endzeit 1. Zeitabschnitt
0xB2	Temperatur 2. Zeitabschnitt
0xB3	Endzeit 2. Zeitabschnitt
0xB4	Temperatur 3. Zeitabschnitt
0xB5	Endzeit 3. Zeitabschnitt
0xB6	Temperatur 4. Zeitabschnitt
0xB7	Endzeit 4. Zeitabschnitt
0xB8	Temperatur 5. Zeitabschnitt
0xB9	Endzeit 5. Zeitabschnitt
0xBA	Temperatur 6. Zeitabschnitt
0xBB	Endzeit 6. Zeitabschnitt
0xBC	Temperatur 7. Zeitabschnitt
0xBD	Endzeit 7. Zeitabschnitt
0xBE	Temperatur 8. Zeitabschnitt
0xBF	Endzeit 8. Zeitabschnitt
0xC0	Temperatur 9. Zeitabschnitt
0xC1	Endzeit 9. Zeitabschnitt
0xC2	Temperatur 10. Zeitabschnitt
0xC3	Endzeit 10. Zeitabschnitt
0xC4	Temperatur 11. Zeitabschnitt
0xC5	Endzeit 11. Zeitabschnitt

0xC6	Temperatur 12. Zeitabschnitt
0xC7	Endzeit 12. Zeitabschnitt
0xC8	Temperatur 13. Zeitabschnitt
0xC9	Endzeit 13. Zeitabschnitt

### 8.1.1.5 Wetterstation anmelden

Momentan werden folgende Wettersensoren unterstützt:

Temperatur- und Feuchtesensor:

HM-WDS40-TH-I (Typ-Nr: 0x003F)

HM-WDS10-TH-O (Typ-Nr: 0x003D)

Große Wetterstation:

HM-WDS100-C6-O (Typ-Nr: 0x0040)

Sensor am Gateway anmelden:

Diese Sensoren müssen mit ihrer Typ-Nr. angemeldet werden.

Beispiel:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=addSensor&type=HM&adr=003D134FD601](http://{IP-Adresse-des-Gateways}/command?XC_FNC=addSensor&type=HM&adr=003D134FD601)

Nach der Adresse wird noch die Kanalnummer „01“ angehängt.

Das Statusrückgabeformat für die beiden Temperatur- und Feuchtesensoren bei „GetStates“ ist identisch.

Es wird ein JSON Array mit dem Status zurückgegeben:

```
{"type": "HM", "adr": "134FD601", "state": "00F2:2C:00"}
```

„state“ setzt sich folgendermaßen zusammen:

Temperatur	:	Feuchtigkeit	:	Status/Alive
0x00F2		0x2C		00
0x00F2 = 242 → 24,2°C		0x2C = 44 → 44%		Letzte Meldung vor x min  0xFE → Battery low  0xFF → Timeout

Bei der großen Wetterstation wird der folgender JSON Array mit dem Status zurückgegeben:

```
{"type": "HM", "adr": "1DE1E501", "state": "00EC:2C:0D4:C:41:000B:000B:0000:05A:0A:0:00"}
```

„state“ setzt sich folgendermaßen zusammen:

Temperatur	:	Feuchtigkeit	:	Windgeschwindigkeit	:	Schwankungsbreite	:	Windrichtung
0x00F2		0x2C		0x0D4		0xC		0x41
0x00F2 = 242 → 24,2°C		0x2C = 44 → 44%		0x0D4 = 212 → 21,2km/h		0xC = 12 → 12 * 5,5 = 66°		0x41 = 65 → 65 * 5° = 325°

:	Regenmenge	:	Regenmenge heute	:	Regenmenge gestern
	0x00B		0x00B		0x000
	0x00B = 11 → 11 * 295ml = 3245 ml → 3,2mm				

:	Sonnenscheindauer	:	Helligkeit	:	Regen	:	Status/Alive
	0x05A		0x0A		0x0		0x00
	0x05A = 90 → 90min		0x0A = 10 → 10		0 → kein Regen		Letztes Packet vor x min

			1 → regen	0xFE → Battery low 0xFF → Timeout
--	--	--	-----------	--------------------------------------

### 8.1.1.6 Wassersensor anmelden

Wassersensor am Gateway anmelden:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=addSensor&type=HM&adr=00451DE4E101](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=addSensor&type=HM&adr=00451DE4E101)

Es wird ein JSON Array mit dem Status zurückgegeben:

```
{"type":"HM","adr":"1DE4E101","state":"00:00"}
```

„state“ setzt sich folgendermaßen zusammen:

Status	:	Status/Alive
0x00 → trocken		Letzte Meldung vor x Minuten
0x64 → feucht		0xFE → Battery low
0xC8 → nass		0xFF → Timeout

### 8.1.1.7 Jalousieaktor konfigurieren

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=hblindcal&adr=14983E01&data=007800820505](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=hblindcal&adr=14983E01&data=007800820505)

### 8.1.1.8 Sprachmodul Signalleuchte konfigurieren

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=SendSC&type=HM&adr=1E3695011B281103](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=SendSC&type=HM&adr=1E3695011B281103)

„adr“ setzt sich folgendermaßen zusammen:

Geräteadresse	Kanal	Befehl	Virtueller Kanal	Farbe	Wiederholungen
0x1E3695	0x01	0x1B	0x28	0x11	0x03
	Signalleuchte hat immer Kanal 1	Konfigurieren des Sprachmoduls	Kanal über den die Eingestellte Aktion angesteuert wird	0x11 = Rot kurz 0x12 = Rot lang 0x21 = Grün kurz 0x22 = Grün lang 0x31 = Orange kurz 0x32 = Orange lang	Wie oft die eingestellte Farbe blinken soll

Virtuellen Kanal 0x28 auslösen:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=SendSC&type=HM&data=1E36952819](http://{IP-Adresse-des-Gateways}/command?XC_FNC=SendSC&type=HM&data=1E36952819)

#### 8.1.1.9 Sprachmodul Sprachausgabe konfigurieren

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=SendSC&type=HM&adr=1E3695021B01030464](http://{IP-Adresse-des-Gateways}/command?XC_FNC=SendSC&type=HM&adr=1E3695021B01030464)

„adr“ setzt sich folgendermaßen zusammen:

Geräteadresse	Kanal	Befehl	Virtueller Kanal	Klang Nr
0x1E3695	0x02	0x1B	0x01	0x03
	Sprachausgabe hat immer Kanal 2	Konfigurieren des Sprachmoduls	Kanal über den die Eingestellte Aktion angesteuert wird	Klang Nr mit der der virtuelle Kanal verknüpft werden soll

Wiederholungen	Lautstärke
0x04	0x64

Wie oft die eingestellte Sprachdatei abgespielt werden soll	Lautstärke in Prozent. Einstellbar in 10% Schritten.  10% = 0x0A  20% = 0x14  100% = 0x64
---	---

Virtuellen Kanal 0x01 auslösen:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=SendSC&type=HM&data=1E36950119](http://{IP-Adresse-des-Gateways}/command?XC_FNC=SendSC&type=HM&data=1E36950119)

### 8.1.1.10 Energiemessteckdose

Modell: HM-ES-PMSw1-PIType: 0x00AC

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=addSensor&type=HM&adr=00AC2516C001](http://{IP-Adresse-des-Gateways}/command?XC_FNC=addSensor&type=HM&adr=00AC2516C001)

Es wird ein JSON Array mit dem Status zurückgegeben:

```
{"type":"HM","adr":"2516C001","state":"1:00EA:0640:0074:08A6:01:00:00"}
```

„state“ setzt sich folgendermaßen zusammen:

Gültig	: Energie	: Leistung	: Strom	: Spannung
1	0x00EA	0x0640	0x0074	0x08A6
1 = Gültige Werte 0 = ungültige Werte	0x00EA = 234 => 234 / 10 = 23,4 Wh	0x0640 = 1600 1600 / 100 = 16,0W	0x0074 = 116mA	0x08A6 = 2214 => 2214 / 10 = 221,4 V

: Frequenz	: Status
0x01	0x00

Signed	0x00 = off
0x01 => 0,01Hz	0xC8 = on
50Hz + 0,01Hz = 50,01Hz	
0xFE = -0,02Hz	
=> 50Hz - 0,02Hz = 49,98 Hz	

#### 8.1.1.11 HomeMatic-Gerätetypen

Gerätetyp	Typ-Nr	Beschreibung	Geräte
HM_WDS10_TH_O	0x003D	outdoor radio-controlled temperature/humidity sensor	Kleine Wetterstation
HM_WDS40_TH_I	0x003F	indoor radio-controlled temperature and humidity sensor	Kleine Wetterstation
HM_Sec_WDS	0x0045	radio-controlled water detection sensor	Water detection
HM_LC_BI1_FM	0x0005	radio-controlled blind actuator 1-channel (flush-mount)	Blind
HM_LC_Sw1_PI	0x0011	radio-controlled socket adapter switch actuator 1-channel	Switch
HM_Sec_SD	0x0042	radio-controlled smoke detector	Smoke Sensor
HM_Sec_RHS	0x0030	Rotary Handle Sensor	Window
HM_CC_RT_DN	0x0095		
HM_TC_IT_WM_W_EU	0x00AD		
HM_CC_TC	0x0039	ClimateControl-ThermoControl	FHT
HM_LC_Sw2_FM	0x0009	radio-controlled switch actuator 2-channel	Switch

		(flush-mount)	
HM_LC_Sw1_FM	0x0004	radio-controlled switch actuator 1-channel (flush-mount)	Switch
HM_Sec_SC	0x002F	Shutter Contact	contact
HM_Sec_SC_2	0x00B1	Shutter Contact	contact
HM_LC_Dim1L_PI	0x0013	radio-controlled socket adapter 1-channel leading edge	dimmer
HM_LC_Sw1PBU_FM	0x069		
HM_LC_Dim1TPBU_FM	0x0068		
HM_LC_BI1PBU_FM	0x006A		
HM_WDS100_C6_O	0x0040	radio-controlled weather data sensor (OC3)	Große Wetterstation

<https://github.com/mhop/fhem-mirror/blob/master/fhem/FHEM/HMConfig.pm>

### 8.1.1.12 HomeMatic-Befehle

Funktion	Befehl	Aktor
on	01	Schaltaktor
off	02	Schaltaktor, Dimmaktor
toggle	03	Schaltaktor
toggleTo	04 Beispiel: 10% = 040A 60% = 043C	Dimmaktor
onTo	05 <b>Beispiel:</b> 10% = 050A 60% = 053C	Dimmaktor
up	06	Jalousieaktor
down	07	Jalousieaktor
moveUp	08	Jalousieaktor
moveDown	09	Jalousieaktor
moveTo	0A <b>Beispiel:</b> 10% = 0A0A 60% = 0A3C	Jalousieaktor
stop	0B	Jalousieaktor
dim_up	0F	Dimmaktor
dim down	10	Dimmaktor
tempTo	11	Thermostat Beispiel: 21°C Soll-Temperatur = 21,0 * 2 =

		42 = 0x2A Command: 112A
tempUp	12	Thermostat
tempDown	13	Thermostat
stepUp	14 Beispiel: 100ms = 140A	Jalousieaktor
stepDown	15	Jalousieaktor
<b>Funktion</b>	<b>Befehl</b>	<b>Aktor</b>
on	01	Schaltaktor

## 8.2 LED Controller I

### 8.2.1 LED Controller I anlernen

Anmelden eines LED Controllers an das a.i.o. gateway:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=learnSC&type=LS](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=learnSC&type=LS)

### 8.2.2 LED Controller I steuern

Angelernte LED Controller steuern:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=SendSC&type=LS&data=1110C0](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=SendSC&type=LS&data=1110C0)

„data“ setzt sich folgendermaßen zusammen:

Adresse	Befehl
111	0C0

Folgende Befehle werden unterstützt:

Funktion	Befehl
Power	0C0
Up	003
Down	00C
Play/Pause	030
Red	00F
Green	033
Blue	0C3

White	03C
Orange	0CC
Yellow	0F0
Cyan	303
Purple	C03
Auto	30C
Jump3	C0C
Fade3	C30
Flash	CC0
Jump7	C0C
Fade7	300
SpeedUp	330
SpeedDown	C00

## 8.3 LED Controller II

### 8.3.1 LED Controller II anlernen

Anmelden eines LED Controllers an das a.i.o. gateway:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=learnSC&type=L2](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=learnSC&type=L2)

### 8.3.2 LED Controller II steuern

Angelernte LED Controller steuern:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=SendSC&type=L2&data=818d1401](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=SendSC&type=L2&data=818d1401)

„data“ setzt sich folgendermaßen zusammen:

Bei einem 1-Byte Befehl (2 ASCII Zeichen):

$data = ((Adresse \& 0xFFFFFE) \ll 8) + Befehl;$

Bei einem 2-Byte Befehl (4 ASCII Zeichen):

$data = ((Adresse \& 0xFFFFFE) \ll 16) + Befehl;$

Folgende Befehle werden unterstützt:

Funktion	Befehl
Power	02
PowerOn	01
PowerOff	02
Left	05
Left/hold	07

Right	06
Up	03
Down	04
Red	0198
Purple	28
Blue	50
Cyan	98
Green	F0
Yellow	0150
White	06
Orange	0170
Auto	07
color	RGB Farbwert

## 8.4 FHT80B API

Es können maximal 4 FHT80b Raumregler an einem a.i.o gateway angemeldet werden. Jeder Raumregler ist mit einem zweiteiligen Sicherheitscode/Adresse (C1 und C2) versehen. C1 und C2 können Dezimalzahlen von 1-99 annehmen.

Um den Sicherheitscode eines Raumreglers auszulesen, muss man die „PROG“ Taste so lange gedrückt halten bis „Sond“ in dem Display angezeigt wird. Danach wählt man mit dem Stellrad die Sonderfunktion „CodE“ aus und drückt die „PROG“-Taste einmal kurz. Es wird der Code 1 (C1) angezeigt. Ein weiterer kurzer Tastendruck auf die „PROG“ Taste und es wird der Code 2 (C2) angezeigt. Damit sich ein Gateway an einem FHT80b Raumregler anmelden kann, muss die Sonderfunktion „CEnt“ auf „nA“ gestellt werden. Dazu hält man die „PROG“-Taste wieder so lange gedrückt bis im Display „Sond“ angezeigt wird. Danach wählt man mit dem Stellrad die Sonderfunktion „CEnt“ aus und drückt kurz die „PROG“-Taste. Dann wählt man mit dem Stellrad „nA“ aus und betätigt kurz die „PROG“ Taste. Nach erfolgreicher Anmeldung wechselt der Status von „CEnt“ auf „ON“.

Das a.i.o. gateway meldet sich zusätzlich mit einem Zentralencode an dem FHT80b Raumregler an, dadurch wird sichergestellt, dass der Raumregler nachher nur noch mit dieser Zentrale zusammenarbeitet. Der Zentralencode kann eine Dezimalzahl von 1-99 annehmen. Momentan wird die Ist-Temperatur, die Soll-Temperatur, der Modus und der Batteriestatus vom a.i.o. gateway ausgewertet.

### 8.4.1 Add FHT80B

Anmelden eines Raumreglers an das a.i.o. gateway:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=AddFHT80b](http://{IP-Adresse-des-Gateways}/command?XC_FNC=AddFHT80b)

&hc1	Hexadezimal (1-Byte)	Code 1 (C1)
&hc2	Hexadezimal (1-Byte)	Code 2 (C2)

Der Sicherheitscode/Adresse wird im EEPROM gespeichert. Ist dies erfolgreich, wird {XC\_SUC} zurückgegeben.

### 8.4.2 Delete FHT80B

Löschen eines Raumreglers von einem a.i.o. gateway:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=DelFHT80b](http://{IP-Adresse-des-Gateways}/command?XC_FNC=DelFHT80b)

&hc1	Hexadezimal (1-Byte)	Code 1 (C1)
&hc2	Hexadezimal (1-Byte)	Code 2 (C2)

Wurde der Raumregler erfolgreich gelöscht wird ein **{XC\_SUC}** zurückgegeben. Ist beim Löschen ein Fehler aufgetreten wird **{XC\_ERR}** zurückgegeben.

### 8.4.3 Zustand verändern

Ändern der Zustände Modus und Soll-Temperatur:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=SendFHT80b](http://{IP-Adresse-des-Gateways}/command?XC_FNC=SendFHT80b)

&hc1	Hexadezimal (1-Byte)	Code 1 (C1)
&hc2	Hexadezimal (1-Byte)	Code 2 (C2)
&tsoll	Dezimalzahl	Die Soll-Temperatur kann zwischen 6,0°C und 30°C in 0,5°C Schritte genau eingestellt werden. Um keine Kommazahlen übertragen zu müssen wird die Soll-Temperatur mit dem Faktor 2 multipliziert. (Beispiel: 22,5°C = 45) Um das Ventil des Heizkörpers vollständig zu schließen wird die Soll-Temperatur auf 11 (5,5°C) gesetzt. Um das Ventil vollständig zu öffnen wird die Soll-Temperatur auf 61 (30,5°) gesetzt.
&modus	Dezimalzahl	Der Modus kann 3 Zustände annehmen: 0 = Manuell, 1 = Automatisch, 2 = Urlaub (wird nicht unterstützt)
&t	Binärzahl	0 = 0,5°C weniger 1 = 0,5°C höher

Konnten die neuen Werte übernommen werden wird **{XC\_SUC}** zurückgegeben.

### 8.4.4 Zustand abfragen

Abfragen der Zustände (IST-Temperatur, Soll-Temperatur, Modus) aller angemeldeten FHT80b Raumregler (maximal 4):

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=GetStates](http://{IP-Adresse-des-Gateways}/command?XC_FNC=GetStates)

Es wird ein JSON Array zurückgegeben:

```
[{"type":"fht80b", "adr":"C1.C2", "state":"M0096:0036:0"}]
```

#### Extended - Status des angemeldeten Fensterkontakts & Ventilstellung:

```
{"type":"FHT80BX", "adr":"4B.19", "state":"0:47:1"}
```

Das erste Zeichen bei „state“, gibt den Status des Fensters an.

0 -> geschlossen

1 -> geöffnet

Das Byte nach dem Doppelpunkt gibt die Ventilstellung in Prozent an - **Hexadezimaler Wert**

Der letzte Wert ist entweder 0 oder 1 und gibt an, ob jemals eine Nachricht des Raumreglers empfangen wurde.

#### 8.4.5 Reset

Löscht alle angemeldeten FHTs

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=fEEPReset&type=06](http://{IP-Adresse-des-Gateways}/command?XC_FNC=fEEPReset&type=06)

adr	Hexadezimal	Code 1 (C1).Code2 (C2)
state	String	Das erste Zeichen gibt den Modus an. Ein 'M' steht für manuell ein 'A' steht für automatisch. Danach folgt die Soll-Temperatur (2-Byte) hexadezimal x 10 Nach dem Doppelpunkt folgt die Ist-Temperatur (2-Byte) hexadezimal x 10. Die Ist-Temperatur kann über 30,0 °C liegen! Nach dem letzten Doppelpunkt kann ein Fehler codiert werden. '0' = kein Fehler '1' = Anmeldung noch nicht abgeschlossen 'B' = Batterie leer 'E' = Lange nichts vom Raumregler gehört

## 8.5 Variablen

### 8.5.1 SetVar

Anlegen einer neuen Variablen oder den Wert einer vorhandenen Variablen ändern.

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=SetVar](http://{IP-Adresse-des-Gateways}/command?XC_FNC=SetVar)

&id	Hexadezimal (1-Byte)		Der a.i.o. creator bestimmt die ID. Die id muss != 0 sein !!!
&type	type	values	Typ der Variablen
	"ONOFF"	on / off / toggle	
	"INT"	4 Byte	
	"FLOAT"	4 Byte	
	"STRING"	Maximal 15 Zeichen	
&value	bei INT & FLOAT Hexadezimal (4-Byte) bei Typ ONOFF gibt es: on, off, toggle		

Existiert die Variable mit der ID schon, wird sie überschrieben!

Beispiel:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=setVar&id=01&type=ONOFF&value=on](http://{IP-Adresse-des-Gateways}/command?XC_FNC=setVar&id=01&type=ONOFF&value=on)

[http://192.168.55.207/command?XC\\_FNC=setVar&id=01&type=int&value=A1B4DC22](http://192.168.55.207/command?XC_FNC=setVar&id=01&type=int&value=A1B4DC22)

## 8.5.2 DelVar

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=DelVar&id=](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=DelVar&id=)

## 8.5.3 GetVar

Abfragen der Variable funktioniert über GetStates:

```
{"type":"ONOFF","adr":"01","state":"on"}
```

```
{"type":"INT","adr":"01","state":"00000002"}
```

Im Feld type steht der Typ.

Im Feld adr steht die ID.

Im Feld state steht die Value (Immer Hexadezimal 4-Byte) außer beim Typ ONOFF.

## 8.6 Sensoren

### 8.6.1 AddSensor

Hinzufügen eines neuen Sensors:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=AddSensor&type=FS20&adr=9bc201](http://{IP-Adresse-des-Gateways}/command?XC_FNC=AddSensor&type=FS20&adr=9bc201)

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=AddSensor&type=IT&adr=00](http://{IP-Adresse-des-Gateways}/command?XC_FNC=AddSensor&type=IT&adr=00)

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=AddSensor&type=WS300&adr=01](http://{IP-Adresse-des-Gateways}/command?XC_FNC=AddSensor&type=WS300&adr=01)

&type	String	FS20, IT, WS300
&adr	Hexadezimal	Adresse des Sensors. Bei FS20 3-Byte: HC1, HC2, ADR Bei IT 1-Byte Bei WS300 1-Byte

### 8.6.2 DelSensor

Löschen eines Sensors:

[http://{IP-Adresse-des-Gateways}/command?XC\\_FNC=DelSensor&type=FS20&adr=9bc201](http://{IP-Adresse-des-Gateways}/command?XC_FNC=DelSensor&type=FS20&adr=9bc201)

### 8.6.3 GetStates

Status abfragen geht über GetStates.

```
{"type":"FS20","adr":"63b300","state":"FF:01"}
```

state: Das erste Byte gibt den Status an. Das zweite Byte gibt an, vor wie vielen Minuten der letzte Status empfangen wurde. (Wert x 5min)

## 8.7 Wettersensor

### 8.7.1 Wettersensor Innen WS300-S300TH

```
{"type":"WS300","adr":"13","state":"0044:031D:00"}
```

Die ersten zwei Byte geben die Temperatur in °C in Hexadezimaler Schreibweise multipliziert mit 10 an.

0x0044 = 6,8°C

Das dritte und vierte Byte gibt die Luftfeuchte in % in Hexadezimaler Schreibweise multipliziert mit 10 an.

0x031D = 79,7%

Das letzte Byte gibt an, vor wie vielen Minuten der letzte Status empfangen wurde. (Wert x 5min)

### 8.7.2 Wettersensor Aussen WS300-KS300

```
{"type":"WS300","adr":"e0","state":"0000:0000:0000:0000:00"}
```

Die ersten zwei Byte geben die Temperatur in °C in Hexadezimaler Schreibweise multipliziert mit 10 an.

0x0044 = 6,8°C

Das dritte und vierte Byte gibt die Luftfeuchte in % in Hexadezimaler Schreibweise multipliziert mit 10 an.

0x031D = 79,7%

Das fünfte und sechste Byte gibt die Windstärke an.

Das siebte und achte Byte gibt die Regenmenge an.

Das letzte Byte gibt an, vor wie vielen Minuten der letzte Status empfangen wurde. (Wert x 5min)

## 8.8 Energie-Meter

### 8.8.1 AddMeter

Hinzufügen eines neuen Sensors:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=AddMeter&type=02&adr=01&rpw=004B](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=AddMeter&type=02&adr=01&rpw=004B)

&type	1-Byte Hex	<p>0x01 = Siemens</p> <p>0x02 = ELV Ferraris</p> <p>0x03 = ELV Plug</p> <p>0x04 = ELV Gaszähler</p>
&adr	Hexadezimal	Adresse des Meters, bei ELV 1-Byte (Hexadezimal)
&rpw	2-Byte Hexadezimal	Umdrehungen pro kWh

z.B. Siemens AMIS:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=AddMeter&type=01&adr=050018454c2d1e02XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=AddMeter&type=01&adr=050018454c2d1e02XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX)

### 8.8.2 DelMeter

Löschen eines Sensors:

[http://{{IP-Adresse-des-Gateways}}/command?XC\\_FNC=DelMeter&type=02&adr=02](http://{{IP-Adresse-des-Gateways}}/command?XC_FNC=DelMeter&type=02&adr=02)

### 8.8.3 GetStates

Status abfragen geht über GetStates.

```
{"type": "EM", "adr": "02.01", "state": "000024D2:0006:02EE"}
```

adr: 1. Byte = Typ des Zählers

2. Byte = Adresse des Zählers

state: die ersten 4-Byte = Gesamtverbrauch in Wattstunden

die nächsten 2-Byte = Verbrauch in den letzten 5min in Wattstunden

die letzten 2-Byte = Umdrehungen pro kWh (nur beim ELV Ferraris Zähler)

## 9 UDP

```
{XC_EVT}{"type":"EM","data":"02.02"}
```

1. Byte = Typ

2. Byte = Adresse