

[TI] Integration Guide

- [Vorwort](#)
- [Nützliche Tools](#)
- [Wichtige Links](#)
- [Einleitung](#)
- [Die Basics](#)
 - [Geräte im Access Point Modus starten](#)
 - [Gerät ins lokale Wifi-Netzwerk einbinden](#)
 - [Einbindung per WebUI](#)
 - [Einbindung per REST-Call \(Webrequest\)](#)
 - [Geräte im Heimnetz finden](#)
 - [Konfiguration über den Webserver des Gerätes](#)
 - [Allgemein](#)
 - [Konfiguration von MQTT](#)
 - [Geräte-Konfiguration zurücksetzen](#)
 - [Per WebUI](#)
 - [Per REST-Call](#)
 - [Per Powercycle](#)
 - [Das Verhalten des Gerätes untersuchen / Debugging](#)
 - [Per WebUI](#)
 - [Per Syslog](#)
 - [Auslösen von Geräte-Aktionen](#)
 - [Allgemein](#)
 - [Spezifische Datenpunkte](#)
 - [Switch It](#)
 - [Shut It](#)
 - [Dimm It](#)
 - [Tasmota-Tuya-Helper](#)
 - [Tuya Protokoll](#)
 - [Tasmota Tuya](#)
 - [Per WebUI](#)
 - [Per REST-Call](#)
 - [Per MQTT](#)
 - [Besonderheiten](#)
 - [Switch It](#)
 - [WebUI-Anpassung](#)
 - [Dimm It](#)
 - [WebUI-Anpassung](#)
 - [Shut It](#)
 - [Kalibrierung](#)
 - [WebUI-Anpassung](#)
- [Firmware Update](#)
 - [Offizielle Tasmota-Firmware](#)

Vorwort

Dieses Dokument richtet sich an Leser, denen Tasmota (<https://github.com/arendst/Tasmota>) ein Begriff ist, die in Grundzügen ein gewisse Affinität zu Technik vorweisen können, die ein Basis-Wissen im Umgang mit Netzwerkgeräten haben und sich nicht scheuen ihr Wissen zu erweitern und durchaus im Stande sind Dokumentationen zu verstehen und anzuwenden. Weiterhin ist der Umgang mit curl (<https://everything.curl.dev/get>) und postman (<https://www.postman.com/downloads/>) oder vergleichbaren Tools Voraussetzung, um mit Hilfe dieses Guides voran zukommen und die vorliegenden Smarthome-Geräte in das Heimnetzwerk einzubinden.

Wenn im Folgenden von “Gerät” die Rede ist, dann ist damit ein *Switch It*, *Dimm It* oder *Shut It* gemeint und nicht etwa ein daran angeschlossenes Gerät.

Nützliche Tools

- curl
- Postman
- Portscanner

Wichtige Links

Tasmota Dokumentation → <https://tasmota.github.io/docs/>

Serial Protocol Tuya → <https://developer.tuya.com/en/docs/iot/tuya-cloud-universal-serial-port-access-protocol?id=K9hhi0xxtn9cb>

Tasmota-Tuya-Helper → <https://github.com/sillyfrog/Tasmota-Tuya-Helper>

Einleitung

Zu den aktuell verfügbaren It - Geräten zählen:

- **Switch It** - Schalten von Lasten
- **Shut It** - Steuern von Rolläden und Vorhängen
- **Dimm It** - Dimmen von Lampen

Alle aufgeführten Geräte lassen sich über die *my wesmartify* App in ein privates Heimnetzwerk aufnehmen und bieten über einen dahinterliegende Cloudservice entsprechende smarthome Features an. Die OpenCloudService-Plattform (OCS-Plattform) bietet dabei die Möglichkeit, Tasmota-basierte Geräte anzubinden und über eine App in Betrieb zu nehmen und zu steuern.

my wesmartify App

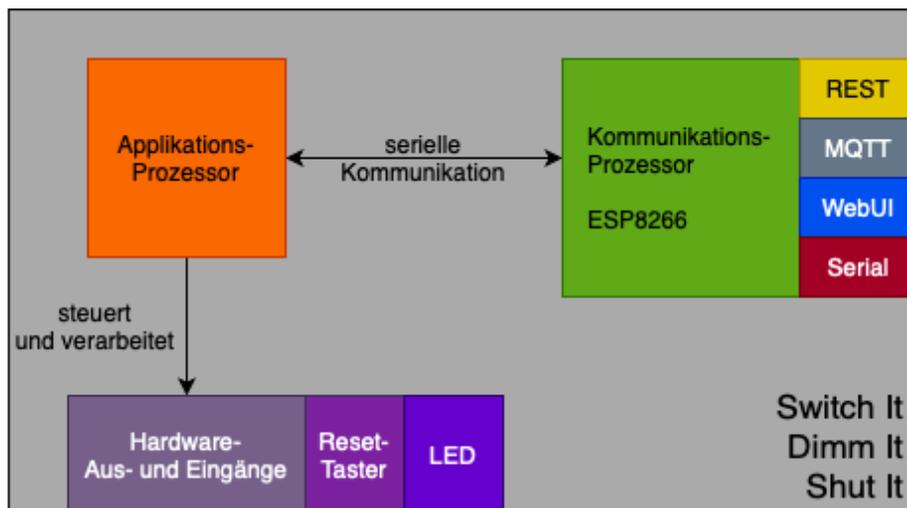
iOS: <https://apps.apple.com/de/app/my-wesmartify/id1637073690>

Android: <https://play.google.com/store/apps/details?id=de.wesmartify.mywesmartify&pli=1>

Um die Geräte ohne die my wesmartify App verwenden zu können, lassen sich mit Hilfe von REST-Calls/Webrequests (POST/GET), über MQTT oder die WebUI Aktionen am Gerät auslösen.

Auf den Geräten ist initial nach der Auslieferung die Firmware **Tasmota v12.4.0** installiert. Dies ist eine customized Basis-Firmware von Tasmota, die über einen angepassten Funktionsumfang verfügt, der bei der Erstellung der Firmware über Compiler-Schalter konfiguriert wurde. Die Geräte können so mit der OCS-Plattform der hinter der my wesmartify App steckt kommunizieren. Neben aktivierten Features wie AWS-Support und MQTT over TLS, sowie dem TUYA-Modul, sind viele andere Module deaktiviert (zum Beispiel der Support für verschiedenste Sensoren oder Services).

Das wichtigste vorweg, die angesprochenen Geräte basieren auf einer Zwei-Prozessor-Architektur. Sprich ein Prozessor (ESP8266) stellt den Kommunikationspart zur Außenwelt zur Verfügung, während der andere Prozessor für die Steuerung und Regelung der an das Gerät angeschlossenen (Hardware-) Komponenten zuständig ist.



Damit die beiden Prozessoren voneinander Wissen und miteinander Interagieren können, kommunizieren diese über eine serielle Schnittstelle und sprechen dabei ein simples Protokoll namens Tuya (<https://developer.tuya.com/en/docs/iot/tuya-cloud-universal-serial-port-access-protocol>).

Die initiale Firmware ist so (vor-)eingestellt, dass die Kommunikation zwischen den beiden Prozessoren und besagten Protokoll stattfinden kann.

Eine Änderung dieser Einstellung bedingt, eine nicht funktionsfähige (angeschlossene) Hardware.

Mit Hilfe dieses Guides sollte es möglich sein, ein Gespür für die Geräte und deren Eigenschaften, sowie deren Integration in ein Heimnetzwerk zu erhalten.

Die Basics

Ausgehend von einem werksfrischen aus der Packung entnommenem Switch/Shut/Dimm It, lassen sich die Geräte gleichermaßen in ein bestehendes Wifi-Netzwerk einbinden. Nach der Einbindung in das lokale Wifi lassen sich Kommandos an das Gerät senden.

Geräte im Access Point Modus starten

Ist das vorliegende Gerät bereits in ein bestehendes Wifi-Netzwerk integriert, kann durch einen Taster auf der Unterseite (Switch It und Shut It) bzw. den durch eine Aussparung an der Oberseite des Gehäuses erreichbaren Taster (Dimm It), die Firmware in den Zustand gebracht werden, dass für drei Minuten das Default Wifi-Netzwerk aufgespannt wird.

Nach Ablauf des drei minütigen Timeouts startet das Gerät neu und versucht sich mit den bereits hinterlegten Wifi-Daten zu verbinden.

In einem werksfrischen It Gerät oder in einem It Gerät, welches lediglich mit der *my wesmartify* App verwendet wurde, ist die Konfiguration gültig. Der Taster ist damit funktionsfähig!

Wurde die Konfiguration von Tasmota in Punkto Module / Template / GPIO angepasst, kann es sein, dass die Taster-Funktion nicht mehr gegeben ist.

Gerät ins lokale Wifi-Netzwerk einbinden

Ein werksfrisches It Gerät spannt ein Wifi-Netzwerk mit der SSID *my-wesmartify-XXXX*.

SSID: my-wesmartify-XXXX (kein Passwort)

Eine Verbindung kann mit jedem Wifi-fähigem Rechner, **ohne** die Eingabe eines **Passwortes** hergestellt werden. Nach erfolgreicher Wifi-Verbindung ist die IP-Adresse, unter der das Gerät erreichbar ist 192.168.4.1.

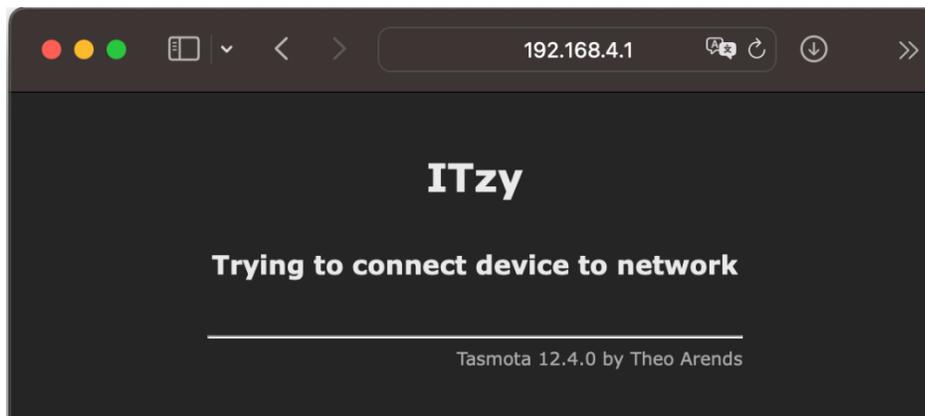
Default IP eines It Gerätes im Access Point Mode: **192.168.4.1**

Einbindung per WebUI

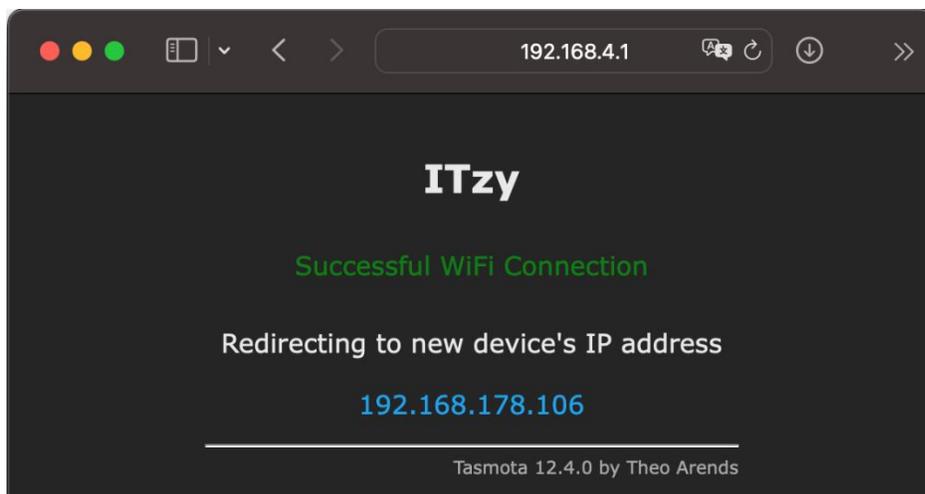
Die einfachste Art der Einbindung in ein bestehendes Wifi-Netzwerk erfolgt mit Hilfe des Webservers der auf den Geräten läuft. Mit einem beliebigen Browser kann unter der Verwendung der IP-Adresse <http://192.168.4.1> die WebUI des Gerätes erreicht werden.



Hier lassen sich nun die Wifi-Zugangsdaten konfigurieren. Nach der Auswahl des Netzwerkes und der Eingabe des Passwortes, startet das Gerät mit einem entsprechenden Hinweis neu.



Nach dem Neustart des Gerätes und dem erfolgreichen Versuch eine Wifi-Verbindung mit den konfigurierten Daten herzustellen, wird im Browser die neue IP (im Heimnetzwerk) des Gerätes angezeigt.



Das User interface des Webservers (WebUI) ist der Einstieg in die weitere individuelle Konfiguration/Parametrierung des Gerätes.

Einbindung per REST-Call (Webrequest)

Unter der Voraussetzung, dass die Verbindung in das durch ein It Gerät aufgespannte Netzwerk (my-wesmartify-XXXX) besteht, lassen sich die Daten des eigenen Heimnetzwerkes ebenfalls über Kommandos übermitteln.

An dieser Stelle sei auf die Dokumentation von Tasmota verwiesen, die die Möglichkeiten aller Kommandos aufzeigt.

Dokumentation der möglichen Kommandos <https://tasmota.github.io/docs/Commands/>

Die Wifi-Daten lassen sich per POST an das It Gerät übertragen. Dazu kann zum Beispiel auf der Kommandozeile per `curl` folgender Befehl verwendet werden:

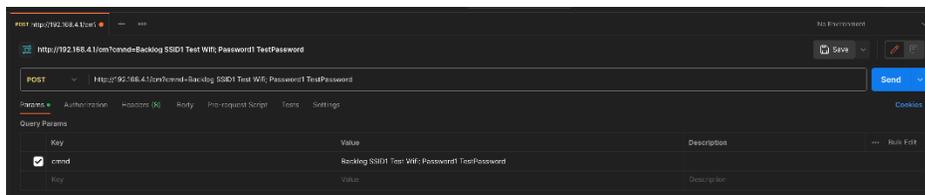
```
curl --location --request POST
'http://192.168.4.1/cm?cmd=Backlog%20SSID1%20Test%20Wifi%3BPassword1%20TestP
assword'
```

Innerhalb eines Web-Query (also dem Teil der URI nach dem `?`) ist es zulässig anstelle von Leerzeichen, die normalerweise als `“%20”` übertragen werden müssen, auch `“+”` zu übertragen (Google verwendet dies zum Beispiel in der Suchanfrage bei mehreren Wörtern).

Generell müssen ansonsten alle Sonderzeichen als `“%hex-Wert”` übertragen werden.

Ein online Web URI Encoder/Decoder hilft bei der Kodierung von Sonderzeichen <https://www.url-encode-decode.com>

Ebenso lässt sich per Postman ein entsprechender Befehl zusammenklicken (bei der Ausführung des Befehls codiert Postman die komplette URI entsprechend nach URL Encoding Standard <https://de.wikipedia.org/wiki/URL-Encoding>).

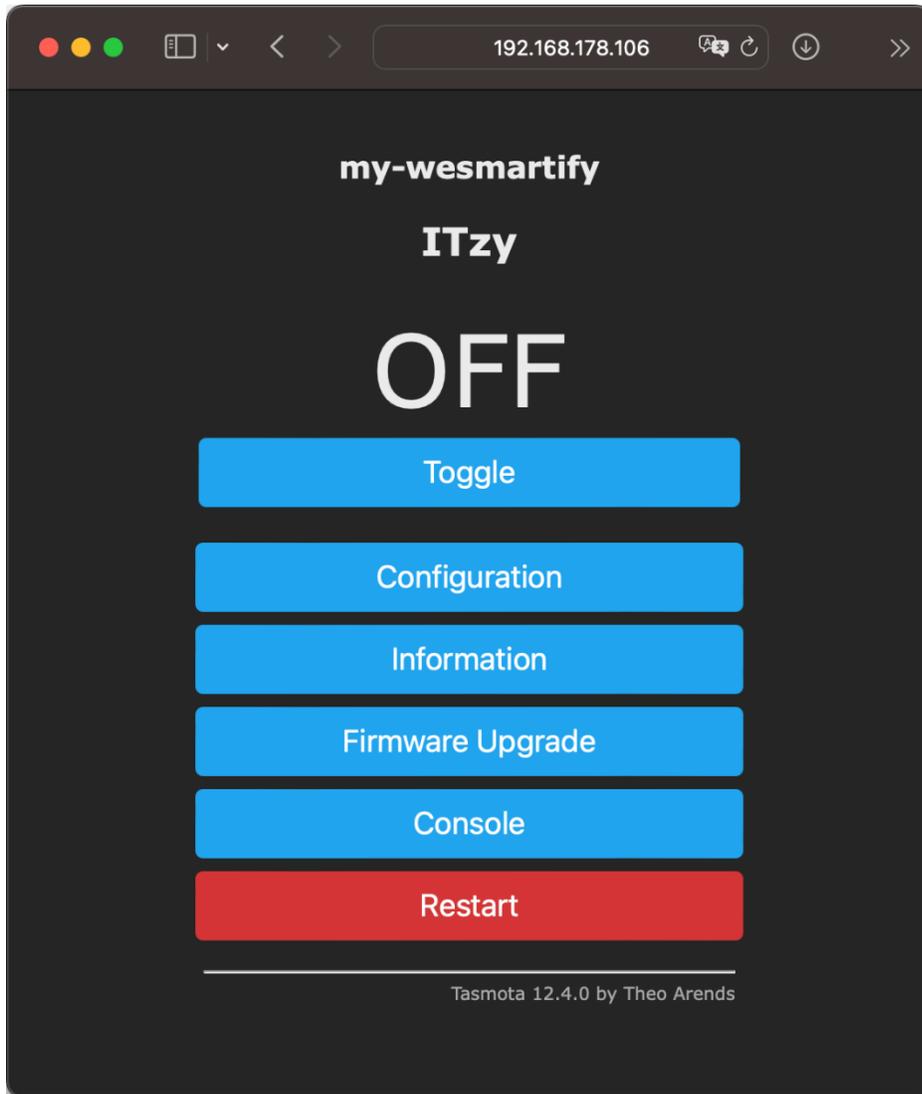


Geräte im Heimnetz finden

Generell lassen sich It Geräte, die bereits in das Heimnetzwerk eingebunden sind, mit einem beliebigen *Portscanner* (zum Beispiel <https://angryip.org>) finden. Der *Hostname* der It Geräte entspricht der *SSID* des aufgespannten Wifi-Netzwerkes im Access Point Mode. Außerdem lassen sich die Geräte per Web detect Namen *Tasmota/12.4.0 (ESP8266EX)* erkennen.

Konfiguration über den Webserver des Gerätes

Allgemein

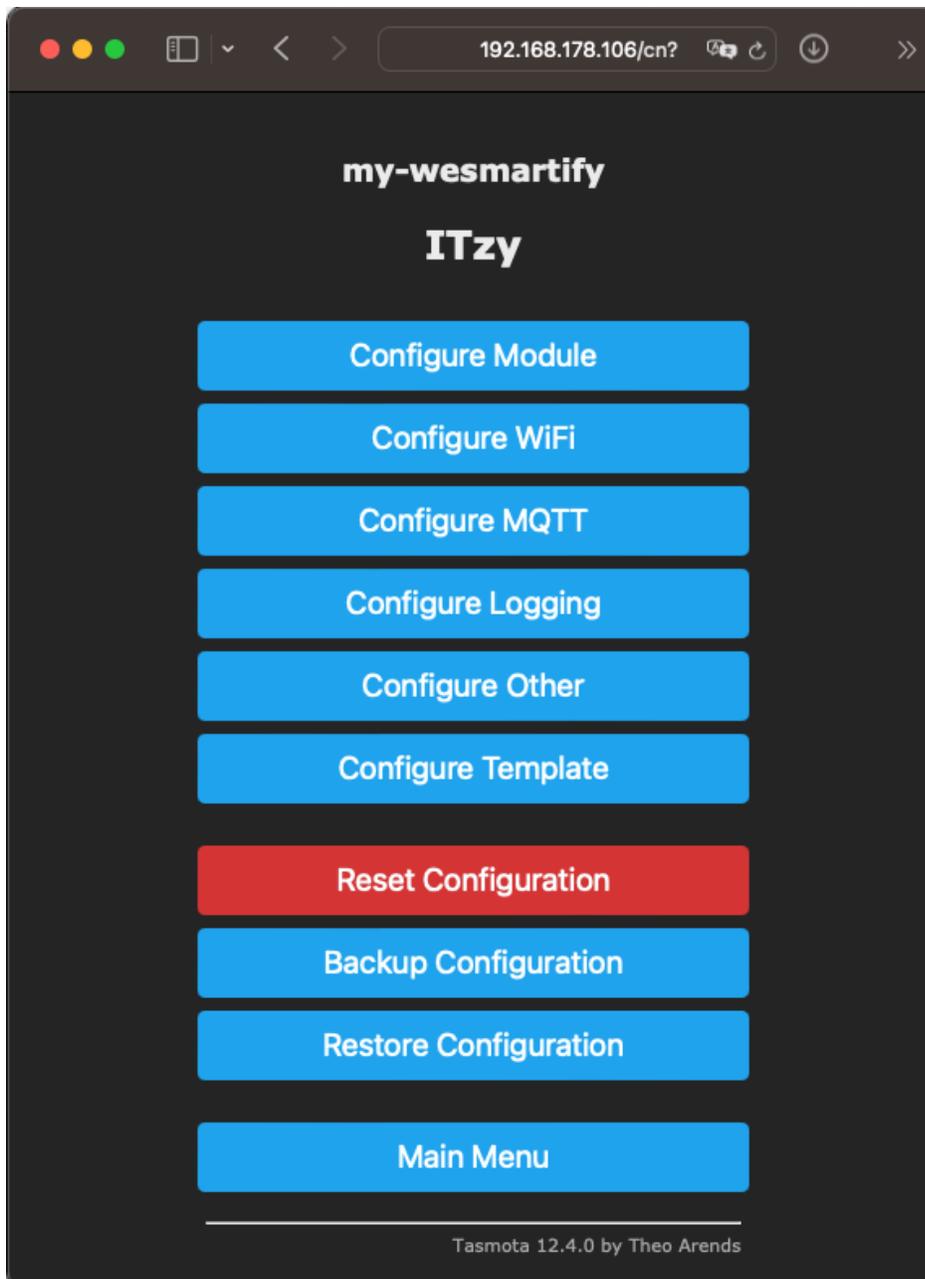


Hinter den UI (user interface) Komponenten verbergen sich folgende Informationen

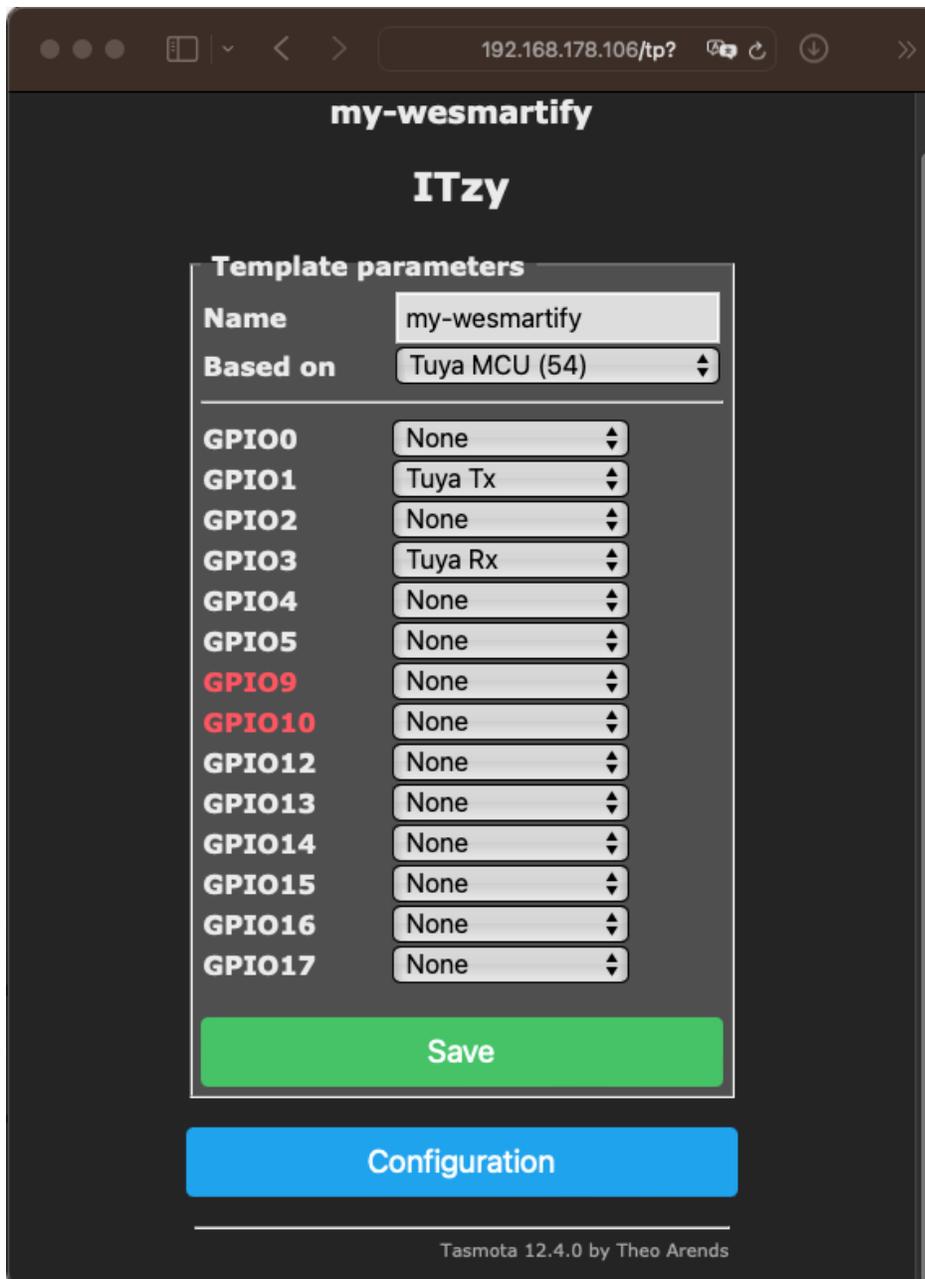
WebUI Komponente	Beschreibung
Status "OFF"	Status des Standard-Ausgangs (siehe nächste Komponente)

Button "Toggle"	Abhängig vom It-Gerät lässt sich mit dem Button der Standard-Ausgang (um-)schalten <ul style="list-style-type: none"> • Switch It → (Um-)schalten des Verbraucher-Ausgangs • Dimm It → (Um-)schalten des Lampen-Ausgangs • Shut It → keine Funktion
Button "Configuration"	Konfiguration aller Tasmota Einstellungen
Button "Information"	Informationen zur Firmware/(Kommunikationsprozessor-)Hardware des Gerätes
Button "Firmware Upgrade"	Firmware Update des Kommunikationsprozessors durchführen
Button "Console"	Logging und Kommando-Konsole
Button "Restart"	Startet das Gerät neu

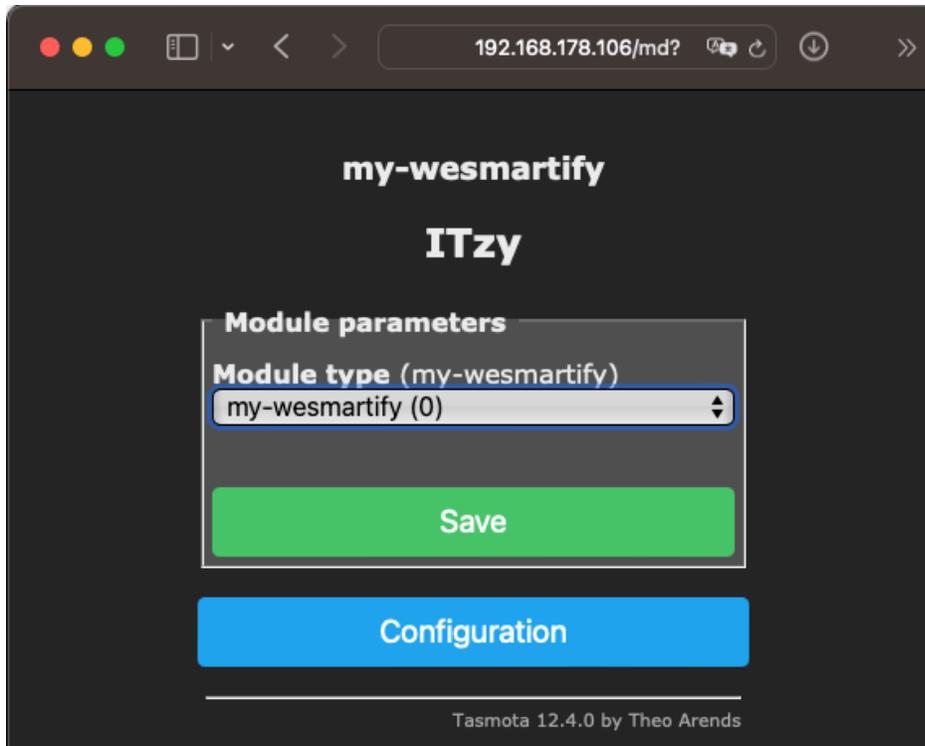
Unter dem Menüpunkt **Configuration** lassen sich weitere spezifische Einstellungen vornehmen. An dieser Stelle wird auf die Tasmota Dokumentation verwiesen. Es wird im Laufenden nur auf relevante Einstellungen näher eingegangen.



Unter dem Menüpunkt **Configure Template** ist die GPIO - Konfiguration (General purpose input output) des Gerätes hinterlegt. Diese Konfiguration ist das Herzstück für eine funktionierende Kommunikation zwischen den beiden Prozessoren. Wird sie für die GPIOs 1 und 3 verändert, so ist die Kommunikation nicht mehr sichergestellt. Ebenso muss die Einstellung **Based on** auf dem Eintrag **Tuya MCU (54)** ausgewählt sein.



Unter dem Menüpunkt **Configure Module** lassen sich verschiedene (vorkonfigurierte) Module, in Abhängigkeit von der zugrundeliegenden Hardware auswählen. Die Auswahl wiederum hat Einfluss auf die GPIO Konfiguration und einige Funktionen in der Firmware. Hier ist wichtig, dass der *Module type* auf das *Template my-wesmartify* eingestellt ist.



Mit der initialen Firmware Tasmota v12.4.0 lässt sich die WebUI nur geringfügig anpassen!

Bedingt lassen sich einzelne Elemente der WebUI hinzufügen. Sprich für den Shut, als auch den Dimm It lässt sich ein Slider-Element einfügen, um den Rolladen zu positionieren oder eine angeschlossene Lampe zu dimmen.

Um die WebUI weiter anzupassen, bedarf es Änderungen von Sourcecode, die nur durch das compilieren einer angepassten Tasmota Firmware erzeugt werden können.

Konfiguration von MQTT

Um eine MQTT Connection zu einem Broker aufzubauen, muss der Menüpunkt

Geräte-Konfiguration zurücksetzen

Per WebUI

Unter dem Menüpunkt **Configuration** findet sich der Eintrag **Reset Configuration**. Bei der Ausführung des Resets werden alle konfigurierten Einstellungen (inklusive der Wifi-Daten) zurücksetzt. Das It Gerät startet neu und öffnet sein Standard Wifi-Netzwerk (das Gerät ist im Access Point Modus).

Per REST-Call

Wahlweise per `curl` oder `Postman` lässt sich folgender Befehl zum Gerät senden, der zu einem Reset der Konfiguration führt.

```
curl --location --request POST 'http://192.168.178.106/cm?cmd=Reset%201'
```

Die IP-Adresse des Gerätes muss entsprechend angepasst werden.

Per Powercycle

Das It Gerät kann durch folgendes Vorgehen ebenfalls komplett zurück gesetzt werden:

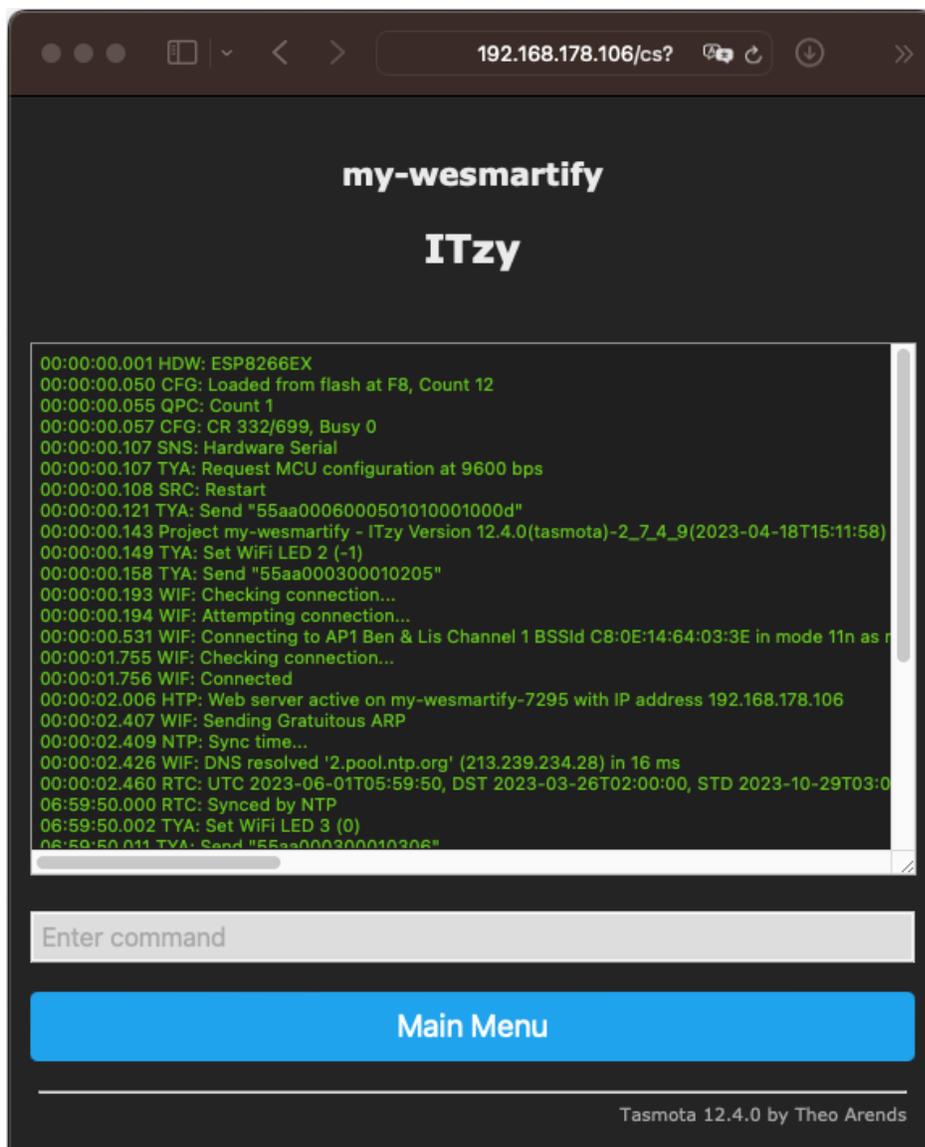
- It Gerät für mindestens 30 Sekunden ausschalten
- 6 maliges An- / Ausschalten der Spannungsversorgung
 - dabei darf die Zeit zwischen den An - und Ausschaltvorgängen nicht größer als 10 Sekunden sein
- nach dem (7. Mal) wieder einschalten ist das It Gerät zurückgesetzt

Siehe dazu Tasmota Dokumentation <https://tasmota.github.io/docs/Device-Recovery/#fast-power-cycle-device-recovery>

Das Verhalten des Gerätes untersuchen / Debugging

Per WebUI

Ausgehend vom Hauptmenü des Gerätes lassen sich über den Menüpunkt **Console** - neben dem Senden von Befehle an das Geräte - auch allerhand Logging Informationen mit lesen. Das Level an Logging-Ausgaben lässt sich begrenzen oder erweitern und kann bei Bedarf im Menü **Configuration** → **Configure Logging** → **Web log level** angepasst werden. Nach dem Neustart des Gerätes, der zum Beispiel über das Commando `Restart 1` in der "Enter command" Zeile des Loggong-Fensters ausgelöst werden kann, könnt der Log in etwas wie folgt aussehen.



Per Syslog

Um Debug-/Log-Informationen unabhängig von der WebUI des Gerätes zu erhalten, besteht die Möglichkeit einen Syslog-Server aufzusetzen. Die Tasmota-Firmware arbeitet dabei als Client, der zum Server Log-Nachrichten sendet. Es gibt eine Reihe Tools, mit dessen Hilfe ich Syslog-Nachrichten aufzeichnen lassen. Google hilft an dieser Stelle weiter.

Für eine simple Syslog-Konfiguration, ohne Syslog-Server-Tool, kann wie folgt beschrieben vorgegangen werden.

Im Menü **Configuration** → **Configure Logging** das Eingabefeld **Syslog host**, mit der **IP-Adresse** des Rechners im lokalen Heimnetz füllen, auf dem das Logging erfolgen soll. Der **Syslog port** kann auf default **514** belassen werden.

Auf dem Rechner, lässt sich - auf Unix-basierenden Systemen - in einer Terminal-Anwendung folgender Befehl aufrufen

```
sudo tcpdump -lns 0 -w - udp and port 514 | strings
```

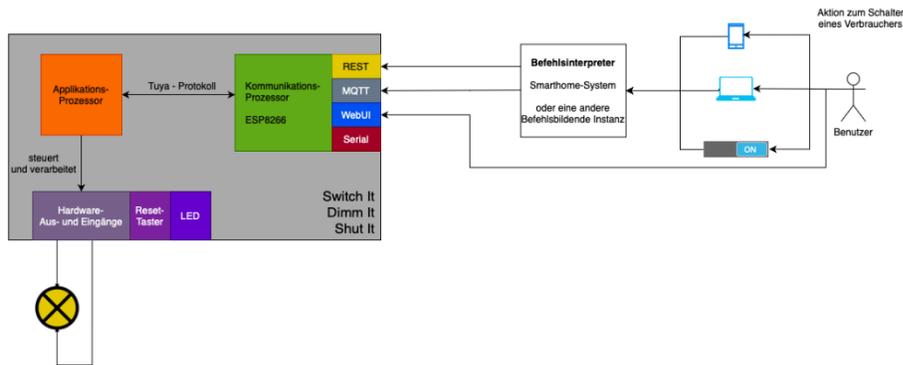
Der Befehl stellt alle UDP-Pakete, die der Rechner auf Port 514 empfängt und die leserliche Strings sind, im Terminal dar.

Der Einsatz eines Syslog-Server Tools ist natürlich komfortabler und lässt mehr Einstellungs- und Filtermöglichkeiten zu.

Auslösen von Geräte-Aktionen

Allgemein

Durch die zwei Prozessor-Architektur der Geräte, lassen sich Hardware-Aktionen nur indirekt mit Hilfe des **Tuya-Protokolls** bewirken. Jedes Gerät hat eigene **spezifische Datenpunkte**, die mit Hilfe der Tasmota-Firmware auf dem Kommunikationsprozessor, zum Applikationsprozessor übermittelt werden können und im Anschluss von diesem in Aktionen übersetzt werden. Die Wirkkette sieht wie folgt aus



Spezifische Datenpunkte

Um Aktionen in den Geräten auszulösen, ist das Wissen über deren spezifische Datenpunkte essentiell.

Switch It

Datapoint-ID DP ID	Datapoint DP	Description	Transfer Type	Data Type	Value Range / Unit / etc.
1	switch	Schaltausgang schalten	Send and Report	bool	0 = off, 1 = on

7	countdown	Schaltausgang nach Zeit x umschalten	Send and Report	value	0 - 86400; step: 1, unit: seconds
---	-----------	---	--------------------	-------	--------------------------------------

Shut It

Datapoint- ID DP ID	Datapoint DP	Description	Transfer Type	Data Type	Value Range / Unit / etc.
1	control	Rolladen/Vorhang steuern	Send and Report	enum	open = 0, stop = 1, close = 2, continue = 3
2	desired curtain position	gewünschte Position des Rolladens/Vorhangs	Send and Report	value	0 - 100; step: 1; unit: %
3	current curtain position	aktuelle Position des Rolladens/Vorhangs	Report Only	value	0 - 100; step: 1; unit: %
12	fault	Fehler	Report Only	fault	motor_fault
16	calibration	Kalibrierung	Send and Report	enum	start = 0, open = 1, close = 2
101	calibration state	Kalibrierungsstatus	Report Only	enum	stanby = 0, calibrating = 1, success = 2, failed = 3

Dimm It

Datapoint- ID DP ID	Datapoint DP	Description	Transfer Type	Data Type	Value Range / Unit / etc.
1	switch	Schaltausgang schalten	Send and Report	bool	off = 0, on = 1
2	brightness	Helligkeit einstellen	Send and Report	value	10-1000; step: 1; unit: -
3	minimum brightness	Minimalwert der Helligkeit	Send and Report	value	10-1000; step: 1; unit: -

4	type of light source	Typ des Leuchtmittels	Send and Report	enum	led = 0, incandescent = 1, halogen = 2
5	maximum brightness	Maximalwert der Helligkeit	Send and Report	value	10-1000; step: 1; unit: -
6	countdown	Umschalten des Schaltausgang	Send and Report	value	0 - 86400; step: 1, unit: seconds

Tasmota-Tuya-Helper

Es gibt ein Tool für die WebUI von Tasmota, welches mit Hilfe eines einfachen Lesezeichens im Browser, die spezifischen Datenpunkte eines Tasmota-(Tuya)-Gerätes in tabellarischer Form darstellt. Das Github-Repository des Tools lautet <https://github.com/sillyfrog/Tasmota-Tuya-Helper>. Der Anleitung folgend, können über die WebUI Console, für einen *Dimm It* folgende Datenpunkt sichtbar gemacht werden. Das Lesezeichen ist ein Javascript Code Snippet, welches die Datenpunkte anfragt und dann in die WebUI von Tasmota einbettet.

my-wesmartify
ITzy

```

15:01:33.149 MQT: tasmota/hings/3e909b86-c60b-4ee9-aa00-b02a8e83d66f/tele/RESULT = {"TuyaReceived":{"Data":"55AA0307000802000400000001D","Cmdnd":"/","CmdndData":{},"FnId":11},"FnId":11}
15:01:33.150 TYA: RX Relay-1 -> MCU State: Off Current State:Off
15:01:33.220 MQT: tasmota/hings/3e909b86-c60b-4ee9-aa00-b02a8e83d66f/tele/RESULT = {"TuyaReceived":{"Data":"55AA0307000802000400000001D","Cmdnd":"/","CmdndData":{},"FnId":11},"FnId":11}
15:01:33.224 TYA: FnId=0 is set for dpId=6
15:01:33.226 TYA: RX value 0 from dpId 6
15:01:33.275 MQT: tasmota/hings/3e909b86-c60b-4ee9-aa00-b02a8e83d66f/tele/RESULT = {"TuyaReceived":{"Data":"55AA030700080200040000010822","Cmdnd":"/","CmdndData":{},"FnId":11},"FnId":11}
15:01:33.278 TYA: FnId=0 is set for dpId=2
15:01:33.280 TYA: RX value 264 from dpId 2
15:01:33.353 MQT: tasmota/hings/3e909b86-c60b-4ee9-aa00-b02a8e83d66f/tele/RESULT = {"TuyaReceived":{"Data":"55AA0307000803020004000000A24","Cmdnd":"/","CmdndData":{},"FnId":11},"FnId":11}
15:01:33.355 TYA: FnId=0 is set for dpId=3
15:01:33.357 TYA: RX value 10 from dpId 3
15:01:33.421 MQT: tasmota/hings/3e909b86-c60b-4ee9-aa00-b02a8e83d66f/tele/RESULT = {"TuyaReceived":{"Data":"55AA0307000805020004000003E807","Cmdnd":"/","CmdndData":{},"FnId":11},"FnId":11}
15:01:33.424 TYA: FnId=0 is set for dpId=5
15:01:33.428 TYA: RX value 1000 from dpId 5
15:01:33.482 MQT: tasmota/hings/3e909b86-c60b-4ee9-aa00-b02a8e83d66f/tele/RESULT = {"TuyaReceived":{"Data":"55AA030700050404000010118","Cmdnd":"/","CmdndData":{},"FnId":11},"FnId":11}
15:01:33.486 TYA: FnId=0 is set for dpId=4
15:01:34.351 TYA: Send *55aa00000000ff*
15:01:34.376 [TuyaReceived":{"Data":"55AA030000010104","Cmdnd":"/","CmdndData":{"01"}"}]
15:01:34.378 TYA: Heartbeat
15:01:45.350 TYA: Send *55aa00000000ff*
15:01:45.376 [TuyaReceived":{"Data":"55AA030000010104","Cmdnd":"/","CmdndData":{"01"}"}]
15:01:45.378 TYA: Heartbeat
15:01:48.825 WiFi: Checking connection...

```

Enter command

Consoles

DpId	Type	Last Val	Cur Val	Send Val	FnId
1	Boolean	Off	Off	On Off	11: Relay 1 (bool)
2	Integer	NaN (0xNaN)	264 (0x108)		
3	Integer	NaN (0xNaN)	10 (0xa)		
4	Enum	0	01		
5	Integer	NaN (0xNaN)	1000 (0x3e8)		
6	Integer	NaN (0xNaN)	0 (0x0)		

Current Configuration Command: Backlog TuyaMCU 11,1;
Run webLog 2 when complete to return console log to default.
See <https://tasmota.github.io/docs/TuyaMCU/> for a full explanation of all of the options.
See the [wiki for troubleshooting tips](#).
Designed for Tasmota v8.5.1 or later.

Tasmota 11.1.0.1 by Theo Arends

Das Tool zeigt ausschließlich die Datenpunkte an, die zum aktuellen Zeitpunkt vom Applikationsprozessor zurück geliefert werden.

Tuya Protokoll

Weiterhin ist es von Vorteil den Aufbau des Tuya-Protokolls verstanden zu haben. Es wird an dieser Stelle auf die Tuya - Doku verwiesen (siehe <https://wolkenmacher.atlassian.net/wiki/spaces/OCS/pages/361299972/TI+Integration+Guide#Wichtige-Links>), und hier nur auf das nötigste eingegangen.

Um Datenpunkte zu Senden, ist der Aufbau einer Nachricht wie folgt zu gestalten.

Field	Länge (Byte)	Instructions
Header	2	0x55AA

Protocol version	1	0x00 = Senden / 0x03 = Empfangen
Command ID	1	0x04
Data length	2	Länge der folgenden Daten in Bytes (ohne die Checksumme)
DP-ID	1	Die Datenpunkt-ID (siehe Spezifische Datenpunkte)
Type	1	0x00 = Rohdaten 0x01 = Boolean 0x02 = Integer
data length	2	Die Länge ist abhängig vom gewählten Datentyp 0x00 = variable Länge 0x01 = Länge 1 0x02 = Länge 4
data	datalength	Daten
Checksum	1	Summe aller Bytes außer der Checksumme, Modulo 256

Tasmota Tuya

Tasmota stellt einen einfachen Befehls-Satz für die Kommunikation mit einem Tuya-Applikationsprozessor zur Verfügung, als auch einen Befehlssatz der auf Protokollebene agieren lässt. Ein Befehl aus dem vereinfachten Befehlssatz (**TuyaSendx**) führt zu einer vollständig kodierten Tuya-Nachrichten, die in Richtung des Applikationsprozessors gesendet wird. Befehle auf Protokollebene (**SerialSendx**) lass die direkte Kommunikation mit dem Applikationsprozessor zu.

Eine vollständige Liste aller Tuya-Befehle ist der aktuellen Tasmota Doku zu entnehmen. Anbei Auswahl an Tasmota-Befehlen die zur Steuerung der Geräte notwendig ist.

	Tasmota Befehl	Beispiel
vereinfachter Befehlssatz	TuyaSend1 <dpID>, <bool>	TuyaSend1 1, 1 -> Setzt den Datenpunkt mit der ID 1 (des Types Boolean) auf 1 TuyaSend1 25, 0 -> Setzt den Datenpunkt mit der ID 25 (des Types Boolean) auf 0
	TuyaSend2 <dpID>, <int>	TuyaSend2 7, 1000 -> Setzt den Datenpunkt mit der ID 7 (des Types Integer) auf 1000

	TuyaSend4 <dpId>, <enum>	TuyaSend4 1, 2 → Setzt den Datenpunkt mit der ID 1 (des Types Enum) auf den Enum-Value 2
	TuyaSendx	siehe Tasmota Dokumentation
Protokollebene	SerialSend5 <hex-data>	SerialSend5 55AA0006000501010001010E -> Setzt den Datenpunkt mit der ID 1 (des Types Boolean) auf 1 SerialSend5 55AA0006000501010001000D -> Setzt den Datenpunkt mit der ID 1 (des Types Boolean) auf 0

Welcher Befehlssatz verwendet wird, um mit dem Applikationsprozessor zu interagieren ist egal und obliegt dem Benutzer.

Per WebUI

Aus dem Hauptmenü heraus wird die bereits erwähnte **Console** angesteuert, um Befehle zum Applikationsprozessor senden zu können. Die Eingabe des Befehls

```
TuyaSend1 1,1
```

führt zum Schalten des Aktorausgangs an einem Switch It und würde ebenfalls eine angeschlossene Lampe an einem Dimm It einschalten.

Auf Protokoll-Ebene würde das Kommando wie folgt aussehen

```
SerialSend5 55AA0006000501010001010E
```

Das Kommando

```
TuyaSend2 7,10
```

würde bei einem Switch It den internen Counter auf 10 Sekunden setzen, der nach Ablauf der Zeit zu einer Deaktivierung des Ausgangs führt.

Per REST-Call

Wie auch schon bei der Übertragung der Wifi Credentials, können Daten per REST wie folgt allgemeingültig übertragen werden:

```
curl --location --request POST '<ip-address-of-your-device>/cm?cmd=<your-command>'
```

Die spitzen Klammern <> sind dabei nicht mit einzugeben. Außerdem erhält man auf einen gesendeten Befehl hin eine Antwort (im Body), der direkt von curl auf der Kommandozeile angezeigt wird.

Zum Einschalten des Ausgangs an einem Switch It Gerät, muss also der Befehl

```
curl --location --request POST '192.168.178.106/cm?cmnd=TuyaSend1%201%2C1'
{"TuyaSend":"Done"}
```

übertragen werden. Analog zum Deaktivieren des Schaltausganges sieht der Befehl wie folgt aus

```
curl --location --request POST '192.168.178.106/cm?cmnd=TuyaSend1%201%2C0'
{"TuyaSend":"Done"}
```

Beobachtet man dabei die WebUI, so ist die Änderung des Zustands zu erkennen (WebUI toggelt zwischen ON/OFF).

Auf Protokollebene schaut das Kommando zum **Einschalten des Switch It Ausgangs**, wie folgt aus

```
curl --location --request POST
'192.168.178.106/cm?cmnd=SerialSend5%2055AA0006000501010001010E'
{"SerialSend":"Done"}
```

Per MQTT

Besonderheiten

Switch It

WebUI-Anpassung

Der Umschalttimer (Wechsel zwischen On ↔ Off des Schaltausganges) lässt sich wie folgt visualisieren

```
TuyaMcu 81,7
```

Dimm It

Die Hardware-Eingänge I1 und I2 sind Tastereingänge und auch nur als solche zu verwenden. Über die Hardware-Eingänge lässt sich die Dimmstufe nicht anpassen.

WebUI-Anpassung

Um die WebUI von Tasmota entsprechend anzupassen, kann mit den folgenden Befehlen, ein Slider zur Regulation der Helligkeit integriert werden.

```
TuyaMcu 21,2
```

Der Umschalttimer lässt sich wie folgt visualisieren

TuyaMcu 81,6

Shut It

Kalibrierung

Für eine Kalibrierung sind folgende Schritte durchzuführen

- Rolladen komplett hochfahren TuyaSend4 1,0 oder manuell
- Rolladen Kalibrierung starten TuyaSend4 16,0
- Datenpunkt 101 sollte auf 2 wechseln

WebUI-Anpassung

Um die WebUI von Tasmota entsprechend anzupassen, kann mit den folgenden Befehlen, ein Slider zur Positionierung des Rolladens integriert werden.

TuyaMcu 21,2

TuyaMcu 27,3

Firmware Update

Offizielle Tasmota-Firmware

Die Firmware auf den It Geräte ist angepasst, so dass diese mit der OpenCloudService-Plattform kommunizieren kann. **Wird eine offizielle Tasmota-Firmware geladen (per OTA oder per File), so kann das It Gerät zukünftig nicht mehr über die wesmartify App bedient werden.** Alle hier beschriebenen Funktionen sollten jedoch weiterhin gegeben sein und im Rahmen von Tasmota anwendbar sein.