Sonoff Zigbee Bridge

Das Gerät stellt eine Schnittstelle zwischen WLAN (IEEE 802.11 b/g/n 2,4 GHz) und Zigbee 3.0 (IEEE 802.15.4 2,4GHz) dar. Geräte mit den Zigbee-Standard sind u.a. bei IKEA, Aldi, und Lidl in Form von Lampen oder Schaltern zu finden. Mit diesen Gateway kann man diese Zigbee-Geräte ohne ein weiteres herstellereigenes Gateway (meist nicht quelloffen) steuern.

1/5



Vorderseite

Rückseite

Platine

Um später die Software "Tasmota" nutzen zu können, müssen wir das Gerät öffnen. Dazu müssen auf der Rückseite die Gummifüße und dann die dahinterliegenden Schrauben entfernt werden.



Vorderseite [Klicken zum Vergrößern.]

Rückseite [Klicken zum Vergrößern.]

Auch hier ist es nicht notwendig Veränderungen auf der Platine vorzunehmen. Hier das Gerät auf der Webseite von "Zigbee Device Compatibility Repository".



Der serieller Anschluss wird mit den USB-to-Serial Adapter verbunden. (RX & TX muss vertauscht werden.)

Folgende Verbindungen müssen zum Adapter herstellt werden:

Platine: IO0 auf Adapter: GND Platine: ETX auf Adapter: RX Platine: ERX auf Adapter: TX Platine: 3V3 auf Adapter: 3,3V Platine: GND auf Adapter: GND

Steckbrücken-Kabel passen perfekt in die Durchkontaktierung auf der Platine der jeweiligen Pins.

Serieller Anschluß

Flash-Vorgang

2025/05/31 10:27

Mittels "Tasmota PyFlasher" wird die neue Firmware auf das Gerät gebracht.

Als Firmware wurde die Datei "tasmota-zbbridge.bin" gewählt. Wie man Tasmota (WLAN) einrichtet findet man hier.

W lashictor yr lash	er —	
Serial port	COM5	~ `
Tasmota firmware	F:\028\tasmota\tasmota-zbbridge.bin	Browse
Erase flash	○ no	
	Flash Tasmota	
Console	Chip is ESP8266EX Features: WiFi Crystal is 26MHz MAC: Uploading stub Running stub Stub running Configuring flash size Auto-detected Flash size: 2MB Erasing flash (this may take a while) Chip erase completed successfully in 2.5s Compressed 603424 bytes to 426175 Writing at 0x00010000 (18 %)	▲

Nachdem Tasmota funktionsfähig auf dem Gerät ist sollte die Firmware auf den Zigbee-Modul auf der Platine aktualisiert werden. Die neue Firmware für das Modul wird als Datei ncp-uartsw_6.7.8_115200.ota geladen

und über die eingebaute Funktion "Firmware Upgrade" als Datei hochgeladen.

Eine Info über das Update des Zigbee-Moduls ist hier zu finden.



Gerät anmelden

Über die "Console" (Webseite) ist es sehr einfach z.B. eine Zigbee-Lampe anzumelden. Dazu geben wir einfach folgenden Befehl ein:

<Code:sh > ZbPermitJoin 1 </Code>

Laut Anleitung zur Lampe muß man diese 3mal Ein- und Ausschalten um das Gerät in den Kopplungsmodus zu bringen. Wenig später erkennt die Sonoff Zigbee Bridge die Lampe und schreib in der Konsole: <Code:sh linenums:1> 16:22:02.328 MQT: sonzb-brg/tele/RESULT = {"ZbState":{"Status":21,"Message":"Pairing mode enabled"}} 16:22:02.331 MQT: sonzbbrg/stat/RESULT = {"ZbPermitJoin":"Done"} 16:22:18.580 ZIG: {"ZbEZSPReceived":"24009130D80E1D03008D150001000000"} 16:22:18.583 MQT: sonzbbrg/tele/RESULT = {"ZbState":{"Status":34,"IEEEAddr":"0x00159D00047E0ED2","ShortAddr":"0x4025","ParentNetwork ":"0x0000","JoinStatus":1,"Decision":0} 16:22:18.678 MQT: sonzb-brg/tele/RESULT = {"ZbState":{"Status":30,"IEEEAddr":"0x00159D00047E0ED2","ShortAddr":"0x4025","PowerSource":t rue,"ReceiveWhenIdle":true,"Security":false} 16:22:18.731 MQT: sonzb-brg/tele/RESULT = {"ZbState":{"Status":32,"ActiveEndpoints":["0x01","0xF2"]} 16:22:18.778 MQT: sonzbbrg/tele/SENSOR = {"ZbReceived":{"0x4025":{"Device":,0x4025","Modelld":"ZBT-ColorTemperature","Manufacturer":"MLI","Endpoint":1,"LinkQuality":79}} 16:22:18.930 ZIG: {"ZbEZSPReceived": "9B00D80E1D03008D150006" } 16:22:20.341 MQT: sonzb-brg/tele/RESULT =
{"ZbState": {"Status": 33, "Device": "0x4025", "Endpoint": "0x01", "ProfileId": "0x0104", "DeviceId": "0x0
10C", "DeviceVersion": 1, "InClusters": ["0x0000", "0x0003", "0x0004", "0x0005", "0x0006", "0x0008", "0
x0300", "0x1000"], "OutClusters": ["0x0019"] } 16:22:20.825 ZIG: Zigbee Devices Data saved in
EEPROM (41 bytes) 16:22:22.349 ZIG: auto-bind `ZbBind
{"Device": "0x4025", "Endpoint": 1, "Cluster": "0x0006" }` 16:22:22.453 MQT: sonzb-brg/tele/RESULT =
{"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:24.353 ZIG: autobind `ZbBind {"Device": "0x4025", "Endpoint": 1, "Cluster": "0x0008" }` 16:22:24.449 MQT: sonzbbrg/tele/RESULT = {"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:24.449 MQT: sonzbbrg/tele/RESULT = {"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:24.449 MQT: sonzbbrg/tele/RESULT = {"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:24.449 MQT: sonzbbrg/tele/RESULT = {"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:24.449 MQT: sonzbbrg/tele/RESULT = {"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:24.449 MQT: sonzbbrg/tele/RESULT = {"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:24.449 MQT: sonzbbrg/tele/RESULT = {"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:26.475 MQT: sonzb-brg/tele/RESULT =
{"ZbBind": {"Device": "0x4025", "Status": 0, "StatusMessage": "SUCCESS" } 16:22:28.341 ZIG: autobind `ZbSend {"Device": "0x4025", "Config": {"Power": {"MinInterval": 1, "MaxInterval": 3600} } }

Die Lampe wird mit der Adresse 0x4025 erfasst und darüber auch gesteuert werden.

Gerät steuern

Folgende Befehle werden in der Konsole (Weboberfläche) eingegeben:

Lampe einschalten <Code:sh > ZbSend { "device":"0x4025", "send":{"Power":"On"} } </Code>

Lampe ausschalten <Code:sh > ZbSend { "device":"0x4025", "send":{"Power":"Off"} } </Code>

Lampe dimmen Hier sind Werte zwischen 0 (Aus) und 254 (höchste Helligkeit) möglich.

<Code:sh > ZbSend { "device":"0x4025", "send":{"Dimmer":254} } </Code>

Lampe - Lichtfarbe ändern <Code:sh > ZbSend { "device":"0x4025", "send":{"CT":65534} }
</Code>

From: https://remo-web.de/ - **remo-web.de**

Permanent link: https://remo-web.de/doku.php?id=hardware:h0007

Last update: 2021/09/26 18:29