

Da ich mit dem Verständnis zur der DeepSleep-Funktion ein wenig Schwierigkeiten hatte und auch im Netz viel Beiträge zu Problemen mit dem DeepSleep bei Tasmota fand, habe ich meine funktionierende Variante, hier als Hilfestellung für andere Verzweifelte ;) dokumentiert.

Als Quelle für meine Beschreibung habe ich das Tasmota-Doc genutzt: [Link](#)

## Erläuterungen:

### 1. Schlafen legen bzw. DeepSleep-Funktion aktivieren

Der DeepSleep wird eingeleitet durch folgenden Befehl in der Konsole:

DeepSleepTime [Schlafdauer in Sekunden] zB **DeepSleepTime 60**, damit schläft er 60Sek.

Das bedeutet, folgende Befehle eingeben:

**Savedata 1** (Alle Eingabewerte werden jede Sekunde gespeichert)

**DeepSleepTime 60** (Die DeepSleep-Funktion mit 60 Sekunden wird aktiviert)

### 2. Aufwecken

Es gibt 2 Möglichkeiten:

- Beim ESP kurz die Spannungsversorgung unterbrechen und wieder verbinden.  
Er bleibt danach nicht dauerhaft wach, sondern macht mit der DeepSleep-Funktion weiter.
- Der Reset-Anschluß RST wird mit dem GPIO16 permanent mit einer Drahtbrücke verbunden und über einen 10k Pullup-Widerstand an 3,3V gelegt, siehe Schaltungsaufbau.  
Jetzt kann der ESP nach Ablauf der DeepSleepTime sich selbst aufwecken. Anschließend macht er mit der DeepSleep-Funktion weiter und legt sich wieder schlafen.

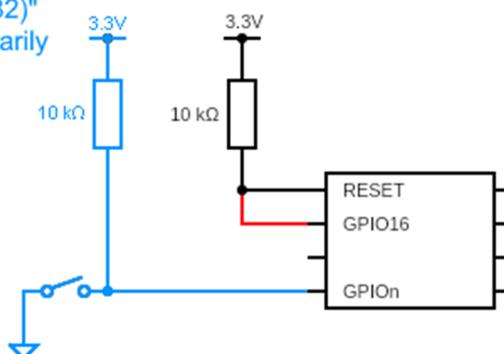
### 3. Deepsleep-Funktion deaktivieren

Es gibt 2 Möglichkeiten:

- In der Konsole folgenden Befehl eingeben: **DeepSleepTime 0**  
Voraussetzung dafür ist, dass der ESP wach ist. Er bleibt anschließend wach.
- Mit Hilfe eines Schalters:  
Dazu einen freien GPIO zB.14 als **DeepSleep (182)** konfigurieren. Hinweis: GPIO00, GPIO15 und GPIO16 dürfen dafür nicht verwendet werden.  
An dem ausgewählten GPIO wird ein 10k Pullup-Widerstand gegen 3,3V und ein Schalter gegen GND angeschlossen, siehe Schaltungsaufbau.  
Wird der Schalter geschlossen, so wird *beim nächsten Aufwachen* des ESPs, zB. durch die DeepSleep-Funktion, der ESP *wach bleiben, solange der Schalter geschlossen ist.*

### 4. Schaltungsaufbau

Option : Use pin as "DeepSleep (182)" to block temporarily DeepSleep



## Wichtige Einstellungen:

### 1. Boot Loop Detektion vom ESP

Diese könnte zu Problemen führen und würde sie grundsätzlich wie folgt einstellen:

[SetOption36 0](#) (Weitere siehe hier [Nr. 6890](#))

### 2. Telemetry period Wert in den Tasmota Einstellungen

Der Wert definiert in welchen Zeitabständen die Wert erfasst werden.

Beispiel:

Bei einem Wert von 30, würden ohne DeepSleep alle 30s die Werte erfasst.

Bei aktiviertem DeepSleep, würden nach dem Aufwecken die Werte erfasst UND nach den eingestellten 30s ein zweites Mal. Anschließend geht der ESP wieder in den DeepSleep. Dies kann man im Konsolen-Screenshot gut sehen.

Des Weiteren hat man bei einem Wert von 30, ein Zeitfenster von 30s, um in der Konsole Befehle abzusetzen, zB. um die DeepSleepTime zu verändern, sehr praktisch!

```
00:00:00 CFG: Loaded from flash at F6, Count 42
00:00:00 Project tasmota WemosD1_01 Version 8.3.1(sensors)-2_7_1
00:00:00 SR04: Mode 1
00:00:00 WIF: Connecting to AP1 FRITZBox-FF4 Channel 1 BSSId E8:DF:70:64:44:7C in mode 11N as WemosD1_01...
00:00:02 WIF: Connected
00:00:02 HTP: Web server active on WemosD1_01 with IP address 192.168.168.105
10:24:08 MQT: Attempting connection...
10:24:08 MQT: Connected
10:24:08 MQT: tele/WemosD1_01/LWT = Online (retained)
10:24:08 MQT: cmd/WemosD1_01/POWER =
10:24:12 MQT: tele/WemosD1_01/STATE = {"Time":"2020-06-12T10:24:12","Uptime":"0T00:00:10","UptimeSec":10,"Heap":25,"SleepMode":"Dynamic","Sleep":50}
10:24:12 MQT: tele/WemosD1_01/SENSOR = {"Time":"2020-06-12T10:24:12","SR04":{"Distance":82.982}}
10:24:42 MQT: tele/WemosD1_01/STATE = {"Time":"2020-06-12T10:24:42","Uptime":"0T00:00:40","UptimeSec":40,"Heap":24,"SleepMode":"Dynamic","Sleep":50}
10:24:42 MQT: tele/WemosD1_01/SENSOR = {"Time":"2020-06-12T10:24:42","SR04":{"Distance":82.561}}
10:24:42 MQT: stat/WemosD1_01/RESULT = {"DeepSleep":{"Time":"2020-06-12T10:25:00","Epoch":1591953900}}
```

## Meine Einstellungen:

Ich nutze den DeepSleep für meine Ultraschall-Niveau-Messung einer Zisterne.

The image displays two screenshots of the Tasmota configuration interface for a WemosD1\_01 module. The left screenshot shows the 'Module parameters' section, where various GPIO pins are configured. The right screenshot shows the 'Logging parameters' section, where the 'Telemetry period' is set to 30. Both screenshots have a 'Save' button and a 'Configuration' button at the bottom.

Module parameter	Value
Module type (Sonoff Basic)	Generic (18)
D3 GPIO0 Button1	None (0)
TX GPIO1 Serial Out	None (0)
D4 GPIO2	None (0)
RX GPIO3 Serial In	None (0)
D2 GPIO4	None (0)
D1 GPIO5	None (0)
D6 GPIO12 Relay1	SR04 Ech/RX (74)
D7 GPIO13 Led1i	SR04 Tri/TX (73)
D5 GPIO14 Sensor	DeepSleep (182)
D8 GPIO15	None (0)
D0 GPIO16	None (0)
A0 ADC0	None (0)

Logging parameter	Value
Serial log level (Info)	2 Info
Web log level (Info)	2 Info
Mqtt log level (None)	0 None
Syslog level (None)	0 None
Syslog host ( )	
Syslog port (514)	514
Telemetry period (300)	30