

Tasmota - Anpassung durch user_config

Inhaltsverzeichnis

- [I - Schnellstart mit user_config_override.h](#)
- [II - Erklärung und Hinweise](#)
 - [II.I - my_user_config.h](#)
 - [II.II - user_config_override.h](#)
- [III - FAQ](#)
 - [III.I - Welche Einstellungen sind vordefiniert?](#)
 - [III.II - Kann ich meine bearbeitete user_config_override.h wiederverwenden?](#)
 - [III.III - Warum immer paarweise #undef...#define ?](#)
 - [III.IV - Komplexere Bedingungen bildet man mit:](#)

I - Schnellstart mit [user_config_override.h](#)

Voraussetzung:

- **Atom** / **VSC** Installation mit PlatformIO oder **Arduino IDE**
- **Tasmota** Source-Code z.B. [Sonoff-Tasmota-development](#)

[MINIMAL_user_config_override.h.zip](#)

- für das schnelle Loslegen - wenige Einträge für eine Basis-Konfiguration

[UNIVERSAL_user_config_override.h.zip](#)

- alle wichtigen Zeilen für die eigene Konfiguration - wie in der [my_user_config.h](#)
 1. [user_config_override.h](#) in den Ordner `\sonoff` kopieren
 2. nur in [user_config_override.h](#) die Einstellungen auf **eigene Konfiguration** ändern
 3. [user_config_override.h](#) **aktivieren** (Details weiter unten) und Tasmota kompilieren

II - Erklärung und Hinweise

II.I - [my_user_config.h](#)

In der Datei `sonoff\my_user_config.h` werden die Einstellungen und Parameter für die Funktion von Tasmota festgehalten.

Die [my_user_config.h](#) (bis Tasmota v6.3.0 `user_config.h`) wird in zwei Abschnitte unterteilt:

- **Abschnitt 1 (SECTION 1):**

Darin sind Konfigurationsdaten für **WLAN**, **MQTT**, **Zeitzone**, **GeoPosition** usw. enthalten.

Alle hier vorgenommenen Einstellungen werden bei einem Update normalerweise **nicht** überschrieben und bleiben im Flash-Speicher **erhalten**.

Erst eine Änderung des CFG HOLDER oder das Löschen des Flash-Speichers bewirkt ein Überschreiben.

Mit Befehlen über die Konsole (Web / Seriell) oder MQTT können die Parameter **permanent** geändert bzw. **gespeichert** werden.

- **Abschnitt 2 (SECTION 2):**

Hier können **Features und Sensoren** aktiviert und deaktiviert werden, um Speicherplatz zu sparen.

Bei jedem Update, mit Änderungen in diesem Abschnitt, wird die **vorherige Konfiguration überschrieben**.

II.II - [user_config_override.h](#)

Es ist zwar möglich alle gewünschten persönlichen Einstellungen in der [my_user_config.h](#) vorzunehmen,

ALLE eigenen Parameter und Anpassungen sollte man aber in der [sonoffuser_config_override.h](#) eintragen.

Warum sollte man das tun?

[my_user_config.h](#) wird i. d. R. bei jedem Tasmota-Update **verändert**.

- Anpassungen müssen dann **jedes Mal neu** in der [my_user_config.h](#) vorgenommen und abgeglichen werden.

+ [user_config_override.h](#) kann bedenkenlos **von der einen zur anderen** Tasmota-Version mitgenommen werden

+ dabei sind, je nach Bedarf, keinerlei Änderungen notwendig

+ sie kann ganz auf persönliche Anforderungen zugeschnitten und übersichtlich gehalten werden

+ was man nie braucht, wird ausgeklammert

+ gleichbleibende Einstellungen können unverändert belassen werden

+ für Spezialfälle kann man verschiedene [user_config_override.h](#) Dateien anlegen

Wie funktioniert das?

Die [my_user_config.h](#) wird beim Kompilieren in Atom / VSC / Arduino-IDE **zuerst** abgearbeitet.

Erst **danach** werden die Definitionen in der [user_config_override.h](#) berücksichtigt.

Das bedeutet, was zuvor in der [my_user_config.h](#) mit `#define` definiert wurde, kann in der [user_config_override.h](#)

--> **deaktiviert** werden mit `#undef` oder

--> **neu definiert** werden mit `#undef / #define`

Wie macht man das?

Bei jeder neuen Version von Tasmota:

1. [user_config_override.h](#) in den Ordner `\sonoff` kopieren

Dazu wählt man z.B.:

[MINIMAL_user_config_override.h.zip](#)

[UNIVERSAL_user_config_override.h.zip](#)

oder eine, die man schon **zuvor** bearbeitet hat

(in `\sonoff` befindet sich die Beispiel-Datei: `user_config_override_sample.h`)

2. **Einstellungen** in der [user_config_override.h](#) bei Bedarf auf **eigene Konfiguration** ändern durch

- **Einfügen / Überschreiben** von Daten z.B. `#define STA_SSID1 ""` zu `#define STA_SSID1 "MeineRouterSSID"`

- **DEAKTIVIEREN** von einzelnen Funktionen und Sensoren durch Löschen von `//` (Kommentar aufheben) vor `#undef`

- **AKTIVIEREN** / Ausklammern durch Setzen von Zeilen als Kommentar durch `//` davor, also `//#undef` oder

durch die Kombination `#undef` und `#define`

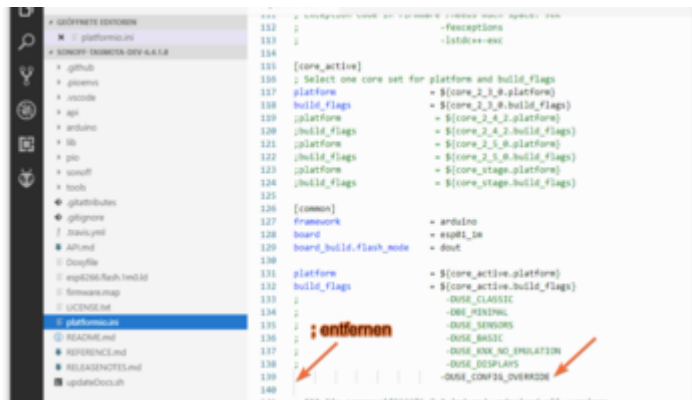
Weitere Infos und Erklärungen in den Kommentaren der [user_config_override.h](#)

3. [user_config_override.h](#) aktivieren und Tasmota kompilieren

Im Normalfall wird die [user_config_override.h](#) für Tasmota nicht berücksichtigt, deshalb müssen vor dem Kompilieren folgende Einträge gemacht werden:

Bei **Atom / VSC** in der **platformio.ini** eine Zeile hinzufügen (bzw. auskommentieren):

`-DUSE_CONFIG_OVERRIDE`



Für **Arduino-IDE** in der [my_user_config.h](#) folgende Zeile aktivieren (`//` davor löschen):

`#define USE_CONFIG_OVERRIDE`



III - FAQ

III.I - Welche Einstellungen sind vordefiniert?

[MINIMAL_user_config_override.h.zip](#)

Abschnitt 1:

NUR das Wichtigste für WLAN, MQTT, Zeit, Ort, Namen, Sprache...

Abschnitt 2:

Default werden die Features so übernommen, wie in der [my_user_config.h](#) vordefiniert.

Am ENDE kann 'Optionale Firmware Konfiguration' gewählt werden,

dadurch ergibt sich z.B. mit `#define USE_BASIC` eine 'kleine' Tasmota firmware.bin OHNE Sensoren.

Code

```
1. /*
2. user_config_override.h - user configuration overrides my_user_config.h for Sonoff-Tasmota
3. Copyright (C) 2019 Theo Arends
6. This program is free software: you can redistribute it and/or modify
7. it under the terms of the GNU General Public License as published by
8. the Free Software Foundation, either version 3 of the License, or
9. (at your option) any later version.
10. This program is distributed in the hope that it will be useful,
12. but WITHOUT ANY WARRANTY; without even the implied warranty of
13. MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14. GNU General Public License for more details.
16. You should have received a copy of the GNU General Public License
17. along with this program. If not, see <http://www.gnu.org/licenses/>.
18. */
20. #ifndef _USER_CONFIG_OVERRIDE_H_
21. #define _USER_CONFIG_OVERRIDE_H_
22. // force the compiler to show a warning to confirm that this file is included
24. #warning **** user_config_override.h: Using Settings from this File ****
26. /*****\
27. * USAGE:
28. * To modify the stock configuration without changing the my_user_config.h file:
29. * (1) copy this file to "user_config_override.h" (It will be ignored by Git)
30. * (2) define your own settings below
31. * (3) for platformio:
32. * define USE_CONFIG_OVERRIDE as a build flags.
33. * ie1 : export PLATFORMIO_BUILD_FLAGS='-DUSE_CONFIG_OVERRIDE'
34. * ie2 : enable in file platformio.ini "build_flags = -WI,-Tesp8266.flash.1m0.ld -
    DUSE_CONFIG_OVERRIDE"
35. * for Arduino IDE:
36. * enable define USE_CONFIG_OVERRIDE in my_user_config.h
37. ****
38. * ATTENTION:
39. * - Changes to SECTION1 PARAMETER defines will only override flash settings if you change define
    CFG HOLDER.
40. * - Expect compiler warnings when no ifdef/undef/endif sequence is used.
41. * - You still need to update my_user_config.h for major define USE_MQTT_TLS.
```

```

42. * - All parameters can be persistent changed online using commands via MQTT, WebConsole or
    Serial.
43. \*****/
44. // ++++++
45. // +++ Aenderungen erst ab hier vornehmen. +++
46. // ++++++
47. // +++ _MINIMAL_user_config_override.h +++
48. // +++ NoitaercX +++ 13.01.2019 16:00:00 +++
49. // ++++++
50. // +++ Kommentare von mir werden durch '+++' gekennzeichnet. +++
51. // +++ ALLES was hier NICHT geaendert werden soll, +++
52. // +++ kann als Kommentar // oder /* */ auskommentiert werden +++
53. // +++ und bleibt dann so, wie in der my_user_config.h definiert.+++
54. // ++++++
55. // ++++++
56. // ++++++
57. // +++ Abschnitt 1 (SECTION 1) +++
58. // ++++++
59. // ++++++
60. // +++ Eigene WLAN-Einstellungen ++++++
61. #undef STA_SSID1 // [SSID1] Wifi SSID
62. #define STA_SSID1 "" // "MeinRouter"
63. #undef STA_PASS1 // [Password1] Wifi password
64. #define STA_PASS1 "" // "MeinPasswort"
65. // -- Project -----
66. #undef PROJECT // +++ Default Topic
67. #define PROJECT "meinsonoff" // PROJECT is used as the default topic delimiter
68. // +++ Eigene MQTT-Einstellungen ++++++
69. #undef MQTT_USE // +++ mit '0' ist MQTT beim Start deaktiviert
70. #define MQTT_USE 0 // [SetOption3] Select default MQTT use (0 = Off, 1 = On)
71. #undef MQTT_HOST
72. #define MQTT_HOST "192.168.x.x" // [MqttHost] +++ MQTT-Host Einstellung
73. #undef MQTT_PORT
74. #define MQTT_PORT 1883 // [MqttPort] +++ MQTT-Port Einstellung
75. // -- HTTP -----
76. #undef FRIENDLY_NAME // +++ angezeigter Name des Geraetes
77. #define FRIENDLY_NAME "MeinSonoff" // [FriendlyName] Friendlyname up to 32 characters used by
    webpages and Alexa
78. // +++ Location ----- Orts-Einstellung -----
79. // +++ 'Berlin'
80. #undef LATITUDE
81. #define LATITUDE 52.520006 // [Latitude] +++ Breitengrad-Angabe fuer die Berechnung von Astro-
    Zeiten
82. #undef LONGITUDE
83. #define LONGITUDE 13.404954 // [Longitude] +++ Laengengrad-Angabe fuer die Berechnung von
    Astro-Zeiten
84. // -- Application -----
85. #undef APP_TIMEZONE // +++ Zeitzone: 99 = Sommer-/Winterzeit
86. #define APP_TIMEZONE 99 // [Timezone] +1 hour (Amsterdam) (-13 .. 14 = hours from UTC, 99 =
    use TIME_DST/TIME_STD)
87. //
88. // ++++++
89. // +++ Befehle in eckigen Klammern koennen direkt in der Tasmota Konsole oder +++
90. // +++ per MQTT zur Abfrage bzw. Konfiguration eingegeben werden z.B. [SSID1] +++

```

```

97. //
+++++
98. // +++ Spracheinstellung +++++
100. #undef MY_LANGUAGE
101. #define MY_LANGUAGE de-DE // German in Germany
102. //
+++++
104. // +++ Abschnitt 2 (SECTION 2) +++
105. // +++ Falls die Funktion in der my_user_config.h aktiviert wurde mit #define +++
106. // +++ dann kann sie hier deaktiviert werden mit #undef +++
107. //
+++++
108. //#undef USE_ENERGY_SENSOR // Disable energy sensors
110. #undef USE_ARDUINO_OTA // Disable support for Arduino OTA
111. #undef USE_WPS // Disable support for WPS as initial wifi configuration tool
112. #undef USE_SMARTCONFIG // Disable support for Wifi SmartConfig as initial wifi configuration tool
113. #undef USE_MQTT_TLS // Disable TLS support won't work as the MQTTHost is not set
114. //#undef USE_CUSTOM // Disable Custom features
116. //#undef USE_DOMOTICZ // Disable Domoticz (+6k code, +0.3k mem)
117. #undef USE_HOME_ASSISTANT // Disable Home Assistant Discovery Support (+2k code)
118. #undef USE_KNX // Disable KNX IP Protocol Support (+9.4k code, +3k7 mem)
119. #undef USE_KNX_WEB_MENU // Disable KNX WEB MENU (+8.3k code, +144 mem)
120. // -- Rules ----- // +++ Achtung: dadurch kann die Verwendung von Rules
deaktiviert werden
122. //#undef USE_RULES // Disable support for rules (+4k4 code)
123. //
+++++
125. // +++ Sensoren +++ +++
126. //
+++++
127. // -- I2C sensors ----- // +++ I2C kann hier mit #undef USE_I2C komplett
deaktiviert werden
128. //#undef USE_I2C // I2C using library wire (+10k code, 0k2 mem, 124 iram)
129. //
+++++
132. // +++ OPTIONALE FIRMWARE KONFIGURATION +++
133. // +++ KEINE oder EINE auswaehlen +++
134. // +++ KEINE: entspricht den Einstellungen in my_user_config.h und den oben vorgenommenen. +++
135. // +++ EINE: die dann enthaltenen Features und Sensoren sind in 'RELEASENOTES.md' +++
136. // +++ beschrieben, die Einstellungen von Abschnitt 2 werden dadurch ueberschrieben. +++
137. //
+++++
138. //#define FIRMWARE_MINIMAL // +++ 1. Tasmota sonoff-minimal: Minimal Firmware fuer OTA
140. //#define FIRMWARE_CLASSIC // +++ 2. Tasmota sonoff-classic: mit WPS, SmartConfig und
WifiManager
141. //#define FIRMWARE_SENSORS // +++ 3. Tasmota sonoff-sensors: mit den meisten Sensoren
142. //#define FIRMWARE_BASIC // +++ 5. Tasmota sonoff-basic: OHNE Sensoren
143. //#define FIRMWARE_DISPLAYS // +++ 6. Tasmota sonoff-display: mit aktivierten Display Treibern
144. //
+++++
146. // +++ ENDE von Abschnitt 2 (SECTION 2) +++
147. // +++ Folgende Zeilen nicht aendern +++

```

```
148. //
    ++++++
149. #endif // _USER_CONFIG_OVERRIDE_H_
```

Alles anzeigen

[UNIVERSAL_user_config_override.h.zip](#)

Abschnitt 1:

ALLES Wichtige für WLAN, MQTT, Zeit, Ort, Namen, Sprache usw. ist aus-/abwählbar.

Abschnitt 2:

Eine Grundauswahl mit einigen, der wichtigsten, Sensoren ist voreingestellt.

III.II - Kann ich meine bearbeitete [user_config_override.h](#) wiederverwenden?

Ja und uneingeschränkt, ohne weitere Veränderungen, vom einen `\sonoff` Ordner zum nächsten (Punkt 3. [user_config_override.h](#) aktivieren muss erneut durchgeführt werden).

III.III - Warum immer paarweise `#undef...#define` ?

Wenn ein Wert bereits definiert wurde, muss er vorerst auf 'nicht definiert' gesetzt werden, um ihn neu festzulegen --> bei Mehrfach-Definitionen kommt es sonst zu Fehlern beim Kompilieren.

III.IV - Komplexere Bedingungen bildet man mit:

```
#ifdef ... // +++ falls definiert ...
    #undef ... // +++ hebe Definition auf ...
    #define ... // +++ definiere ...
#endif

#ifdef ... // +++ falls nicht definiert ...
    #define ... // +++ definiere ...
#endif
```

Weitere Beschreibungen und Fragen --> [Auskommentieren der my_user_config.h](#)

P.S.:

Kommentare zu **Korrekturen** und **Verbesserungsvorschlägen** sind SEHR willkommen und werden von mir mit Sicherheit überdacht.

Wenn dir der Beitrag geholfen hat, freut mich ein '👍' natürlich besonders 😊