

Tasmota auf ESP8266 (D1 Wemos Mini) mit PN532

Changelog v12.3.1 Percy

Tasmota 12.3.1 by Theo Arends

<http://ota.tasmota.com/tasmota/release-12.3.1/>

<http://ota.tasmota.com/tasmota/release-12.3.1/tasmota-sensors.bin.gz>

16:26:39.110 CMD: SetOption19 on

16:26:39.114 RSL: RESULT = {"SetOption19":"ON"}

Tasmota

PN532 NFC reader

[arendst/tasmota](#)

- [V12.3.1](#)

<https://tasmota.github.io/docs/PN532/#tasmota-settings>

PN532 NFC reader

This feature is included only **in tasmota-sensors** (V12.3.1) and tasmota32 binaries

The PN532 is a highly integrated transceiver module for contactless communication at 13.56 MHz based on the 80C51 microcontroller core.

The datasheet for the PN532 chip is [available here](#).

Please note that although the datasheet mentions that the PN532 can be used on SPI, I²C and HSUART that only the HSU interface is implemented in the Tasmota driver.

Configuration

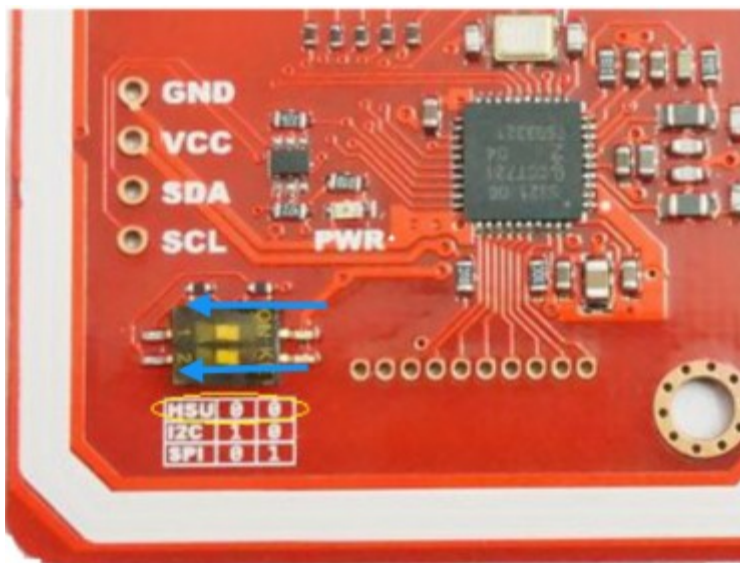
```
#define USE_PN532_DATA_FUNCTION
```

This function is experimental. There are limitations because it seems not all cards are supported by this driver and/or the PN532 module. See issue [4941](#) for more information. We are still researching the

Wiring

As mentioned earlier the PN532 breakout boards usually have pins broken out for all three protocols supported by the PN532 but we are only interested in the HSU interface as that is all the driver currently supports.

For this reason breakout boards have either micro dip switches as shown in the image below, or they have pads on the PC board which you need to bridge out with solder to select which mode the PN532 will operate in.



After selecting the correct protocol mode and connecting the HSU TX/RX pins of the PN532 to the pins you configured on your ESP8266 board you can power it up and the PN532 should be detected automatically.

PN53	ESP
2	

GND	GND
-----	-----

PN532	ESP
VCC	3.3V
SDA	GPIOx RX
SCL	GPIOy TX

Tasmota Settings

In the **Configuration -> Configure Module** page assign:

- GPIOx to PN532 Rx (139) (SDA)(D2)(GPIO4)
- GPIOy to PN532 Tx (138) (SCL)(D1)(GPIO5)

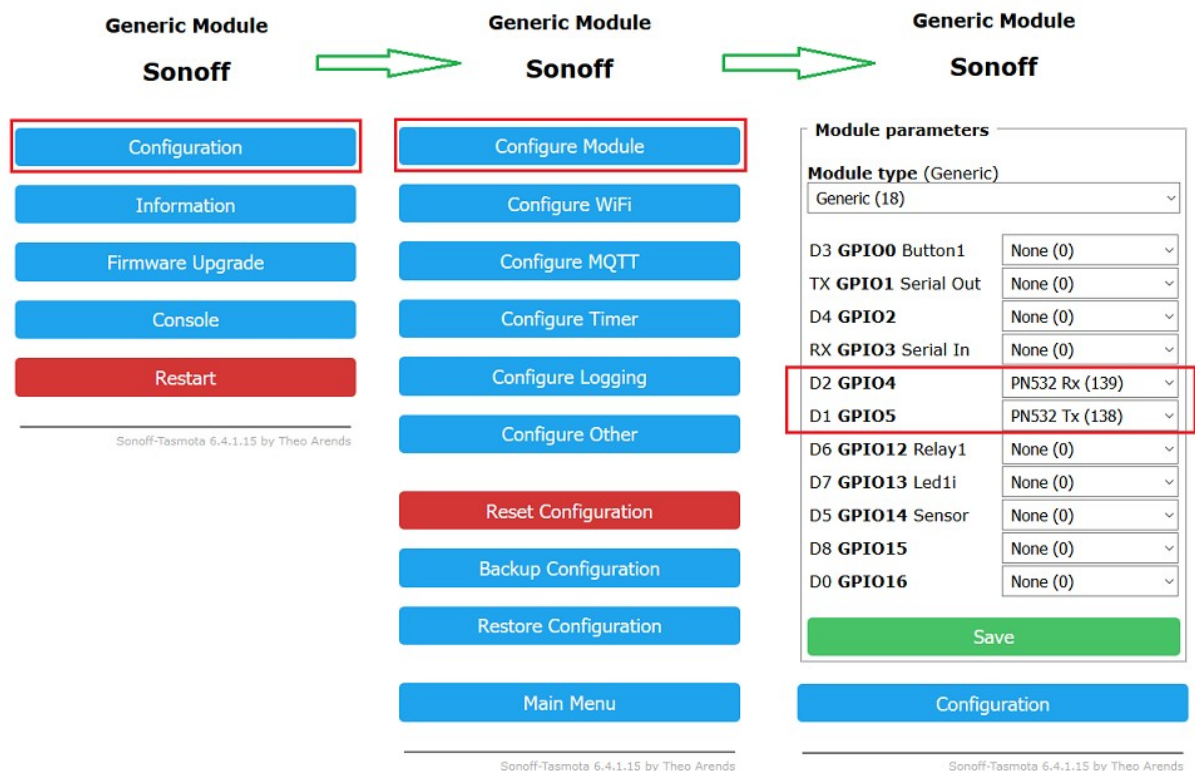
The module will reboot when you save this configuration. During start-up the following information should be visible in your console output:

```
00:00:00 NFC: PN532 NFC Reader detected V1.6
```

If the device was not found please check your wiring and configuration and confirm that everything is as it should be.

Example

Configured using Wemos D1 mini on pins: D1 (connected to PN532 SCL) and D2 (connected to PN532 SDA)



Usage

Tasmota will scan for a new card detect 4 times per second and if found will report it via immediate telemetry.

The output on the console will look similar to the below when a new card is detected

```
18:23:24 MQT: tele/tasmota/SENSOR = {"Time":"2019-01-10T18:23:24","PN532":{"UID":"94D8FC5F",
"DATA":""}}
18:23:24 MQT: stat/tasmota/RESULT = {"Event":"Done"}
18:23:25 MQT: stat/tasmota/RESULT = {"Event":"Done"}
```

The UID of the card/tag is reported and any text stored in BLOCK 1 of a Mifare Classic card or PAGE4-7 of a NTAG card (up to 16 characters in length) is reported in the DATA field of the JSON sent via telemetry. Please note that the DATA field field can contain only printable chars.

The content of the DATA on BLOCK 1 of a Mifare Classic (PAGE4-7 of a NTAG) card can be set as follows

Sensor40 WRITE,I LOVE TASMOTA

Once executed the very next card/tag that is presented to the reader will be programmed accordingly and the data will be retained on the card/tag until either changed or erased.

To erase the content of the DATA field the following command may be used

Sensor40 ERASE

Once executed the very next card/tag that is presented to the reader will have its BLOCK 1 (PAGE4-7 of a NTAG) erased.

For canceling previous command the following command may be used

Sensor40 CANCEL

Once executed will stop waiting next card/tag.

For NTAG213/215/216, NT3H2111/2211 or compatible are additionally available commands: AUTH, SET_PWD, UNSET_PWD for password operations.

Warning. *The password is a unsigned 32 bit and acknowledge is a unsigned 16 bit values.*

Sensor40 AUTH,password,ack

Set tasmota password phrase and password acknowledge for authentication.

Sensor40 SET_PWD Or Sensor40 SET_PWD,new_password Or Sensor40 SET_PWD,new_password,new_ack

Write current password and ack / new password and current ack / new password and new acknowledge to a NTAG21x tag.

Sensor40 UNSET_PWD

Clear password and ack in a NTAG21x tag.

Text logging of the above two actions are also presented during the process for information purposes.

Using the UID and DATA of a presented card

When a card is presented to the PN532 under normal operating conditions up to 3 ways of using the data is possible.

The first is the immediate telemetry generated which looks as follows. For not NTAG21x:

```
18:31:39 MQT: tele/tasmota/SENSOR = {"Time":"2019-01-10T18:31:39","PN532":{"UID":"94D8FC5F",  
"DATA":"ILOVETASMOTA"}}
```

For NTAG21x and correct password and acknowledge:

```
20:05:48 MQT: tele/tasmota/SENSOR = {"Time":"2023-02-24T20:05:48","PN532":  
{"UID":"53BB1F05110001","Data":"I love tasmota","Auth":"Ok"}}
```

For NTAG21x and correct password and not correct acknowledge:

```
20:05:48 MQT: tele/tasmota/SENSOR = {"Time":"2023-02-24T20:05:48","PN532":  
{"UID":"53BB1F05110001","Data":"I love tasmota","Auth":"Nok"}}
```

For NTAG21x and not password protection in a tag:

```
20:05:48 MQT: tele/tasmota/SENSOR = {"Time":"2023-02-24T20:05:48","PN532":  
{"UID":"53BB1F05110001","Data":"I love tasmota","Auth":"None"}}
```

Since this is an immediate telemetry generation as opposed to the sensor data you would normally be expected to be presented when the telemetry period occurs, this telemetry data is not directly usable on the device itself. It is generated and immediately transmitted over MQTT and the purpose of this is so that immediate action may be taken by any home automation software you are using with the data obtained from the card/tag as opposed to waiting for the telemetry period to expire and be sent with normal telemetry data.

For the purpose of using card/tag data on the device itself you will need to use rules along with the events that are caused.

Example

Example rule for responding to a specific UID on the device when a card/tag matching a specific UID is presented

```
rule1 on PN532#UID=94D8FC5F do power on endon
```

Example

Example rule for responding to a specific DATA content that was previously programmed to one or more cards using the Sensor40 write,xxxx command

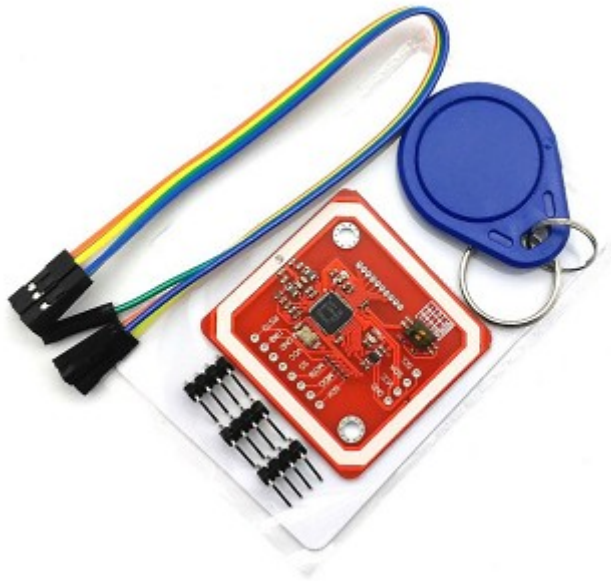
```
Rule 1 on EVENT#PN532_DATA=ILOVETASMOTA do power on endon  
18:41:12 MQT: tele/tasmota/SENSOR = {"Time":"2019-01-10T18:41:12","PN532":{"UID":"94D8FC5F",  
"DATA":"ILOVETASMOTA"}}
```

```
18:41:13 MQT: stat/tasmota/RESULT = {"Event":"Done"}  
18:41:13 MQT: stat/tasmota/RESULT = {"Event":"Done"}  
18:41:13 RUL: EVENT#PN532_DATA=ILOVETASMOTA performs "power on"  
18:41:13 MQT: stat/tasmota/RESULT = {"POWER":"ON"}
```

18:41:13 MQT: stat/tasmota/POWER = ON

Breakout Boards

Since the PN532 chip itself is surface mount and requires some external components to operate the best is probably to obtain a breakout board similar to the one below from your favourite online supplier.



Please make sure the breakout board you order has the HSU pins (SCL=TX and SDA=RX) broken out as it will be very difficult to add them manually - Usually they have them but for good measure just make sure.

→ Umsetzung:

https://www.msxfaq.de/sonst/bastelbude/rfid.htm#tuersteuern_mit_tasmota

Türsteuern mit Tasmota

Nachdem ich einige Projekte mit Tasmota im Eigenheim umgesetzt habe, habe ich auch den RFID-Leser mit dem Garagentoröffner verbunden. Dazu habe ich erst einmal die Voraussetzungen bei der Technik und der Firmware geschaffen.

- ESP8266 mit "Tasmota-sensors.bin" vorbereitet
Firmware über <https://tasmota.GitHub.io/install/> geflashed
Board von "Sonoff Basic" auf "Generic(0)" umgestellt.
WLAN, IP-Adresse und MQTT-Server gesetzt.
- PN532 Mode auf HSU umgestellt
Tasmota kennt nur den Mode und unterstützt bei dem Leser nicht SPI oder I2C
- PN532 NFC reader verkabelt
<https://tasmota.GitHub.io/docs/PN532>
Der Anschluss ist allein durch die vier Kabel schnell erfolgt. Die beiden Signalleitungen verbinden Sie mit zwei Digitalport, die sie dann in Tasmota entsprechend konfigurieren:

SCL = HSU_RX -> D2 (GPIO4) -> PN532 Rx
SDA = HSU_TX -> D1 (GPIO5) -> PN532 Tx
Vcc = Vcc
GND = GND

Beim Start sollte Tasmota auf der Konsole den Reader finden und wenn Sie eine Karte davor halten, sehen Sie die ID der Karte:

```
00:00:00.001 HDW: ESP8266EX
00:00:00.049 CFG: Loaded from flash at F8, Count 12
00:00:00.055 QPC: Count 1
00:00:00.065 Project tasmota - Tasmota Version 10.1.0(sensors)-2_7_4_9(2021-12-08T14:47:41)
00:00:00.078 NFC: PN532 NFC Reader detected v1.6
00:00:00.560 WIF: Connecting to AP1 IOT Channel 6 BSSId [REDACTED] in mode 11n as tasmota-
00:00:06.438 QPC: Reset
00:00:07.798 WIF: Connected
00:00:08.054 HTP: Web server active on tasmota-[REDACTED] with IP address 192.168.[REDACTED]
23:00:41.589 RSL: SENSOR = {"Time":"2022-01-04T23:00:41","PN532":{"UID":"F339973C"}}
23:01:19.509 RSL: SENSOR = {"Time":"2022-01-04T23:01:19","PN532":{"UID":"6761E0C7"}}
```

Die vorgehaltenen Karten können Sie dann per MQTT oder SYSLOG absenden. Interessant ist aber auch ein "autarker" Betrieb. Hier können Sie mit den Tasmota-Regeln eine rein lokale Lösung bauen.

Allerdings ist dies nur bedingt sicher, das RFID-Karten recht einfach kopiert werden können und es ist unbequem, da sie die IDs per Konsole als Regel anlegen müssen.

Ehe ich eine Regel anlege, muss ich einen weiteren Pin als Schaltausgang definieren, an dem ich dann ein Relais anklemmen, welches z.B. den Türöffner für einige Sekunden betätigt. Ich habe dazu einfach D3 GPIO als Relay1 definiert.

Nun muss ich aber Tasmota noch sagen, dass das Relay nach z.B. 0,7 Sekunden automatisch wieder abfällt. Das geht durch die Eingabe von "Pulsetime1 7".

- Tasmota: Commands: Control

<https://tasmota.GitHub.io/docs/Commands/#control>

Ich habe dann über die WebUI das Relay eingeschaltet und nach 1 Sekunde ist es wieder abgefallen:

```
> pulsetime1 70
23:29:51.233 CMD: pulsetime1 7
23:29:51.238 MQT: stat/tasmota_5F2757/RESULT = {"PulseTime1":{"Set":7,"Remaining":7}}
23:29:53.325 MQT: stat/tasmota_5F2757/RESULT = {"POWER":"OFF"}
23:29:53.329 MQT: stat/tasmota_5F2757/POWER = OFF
23:30:58.876 MQT: stat/tasmota_5F2757/RESULT = {"POWER":"ON"}
23:30:58.878 MQT: stat/tasmota_5F2757/POWER = ON
23:31:00.949 MQT: stat/tasmota_5F2757/RESULT = {"POWER":"OFF"}
23:31:00.952 MQT: stat/tasmota_5F2757/POWER = OFF
```

Nun muss ich nur noch einmal alle erlaubten Karten kurz davor halten, um die IDs der Karten zu erfassen. Die brauche ich nun gleich in den Regeln.

- Tasmota Commands: Rules

<https://tasmota.GitHub.io/docs/Commands/#rules>

- Tasmota Rule Syntax

<https://tasmota.GitHub.io/docs/Rules/>

- TasmotaGarage:Rules: Door Opener

<https://tasmota.GitHub.io/docs/Rules/#garage-door-opener>

"SetOption0 0" verhindert, dass der Status der Relays etc. beim Neustart wieder hergestellt wird und "PowerOnState 0" stellt sicher, dass das Relay nach dem Einschalte erst mal aus ist. Nicht dass ein Stromausfall sprichwörtlich Tür und Tor öffnet und das Relay soll nach 0,7 Sek wieder abfallen:

```
SetOption0 0
PowerOnState 0
Pulsetime1 7
Rule1 0
Rule1
ON PN532#UID=6ACB4BB5 DO Power1 1 ENDON
ON PN532#UID=67618D62 DO Power1 1 ENDON
ON PN532#UID=F339E0C7 DO Power1 1 ENDON
ON PN532#UID=7FF4973C DO Power1 1 ENDON
Rule1 1
```

Die Regel 1 enthält einfach alle IDs der RFID-Karten, um das Relay anziehen zu lassen. Interessant finde ich aber, wie viele Karten so einfach ausgelesen werden

können. Die meisten Bankkarten bleiben stumm, weil der Leser wohl nicht die richtige Frage stellt, aber die DB Finkster-Karte, die Pader Bäderkarte, diverse Kantinenkarten von Kunden und sogar der Personalausweis senden eine eindeutige ID unterschiedlicher Länge. Sogar mein iPhone 11 reagiert bei Annäherung und fragt nach der Bezahlkarte (aber nicht andere hinterlegte Karten) und liefert nach Bestätigung eine Seriennummer die sich aber bei jedem Vorgang ändert.

Auch wenn der ePerso eigentlich sicher ist, bedeutet es nicht, da nicht jemand eine Karte mit der gleichen "einfachen" Seriennummer herstellt.

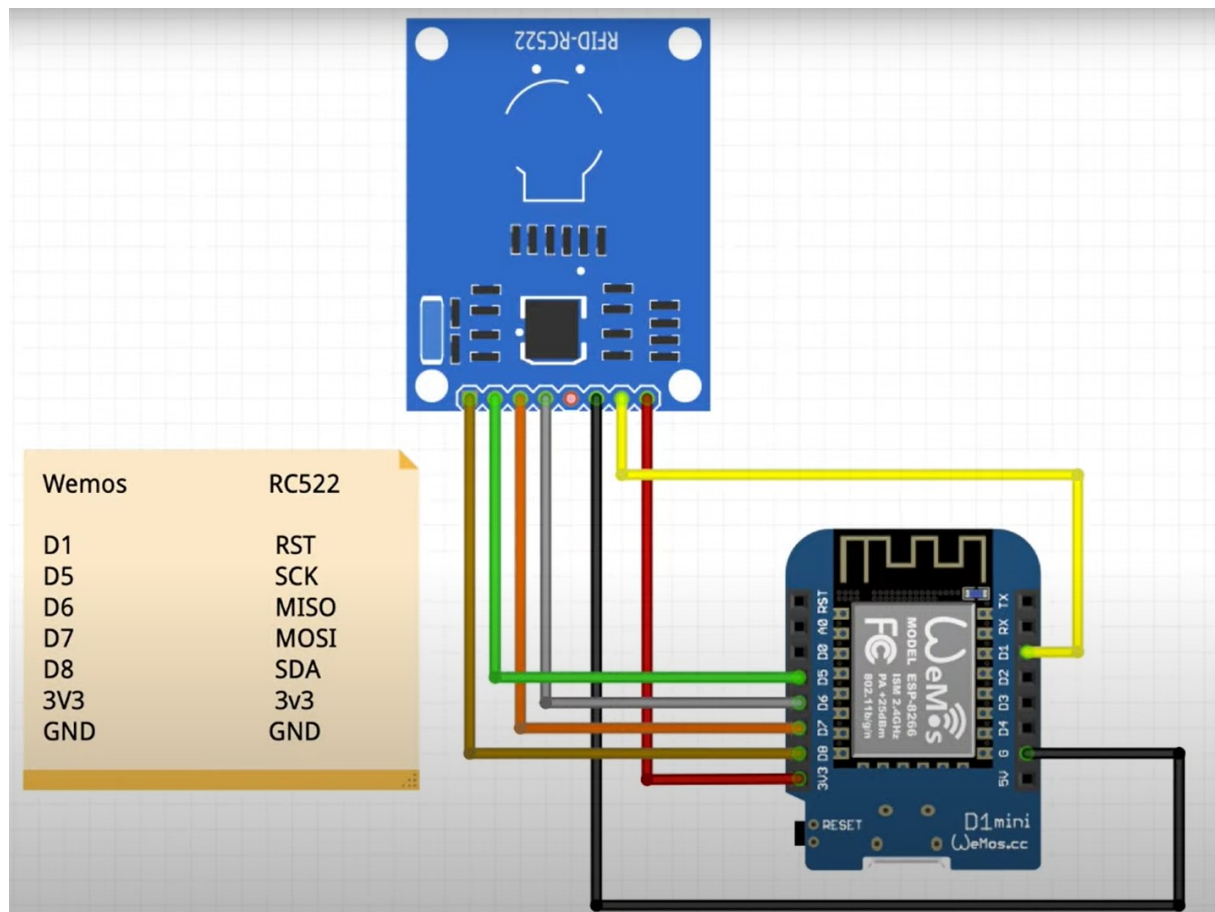
- PN532 NFC reader
<https://tasmota.GitHub.io/docs/PN532>
- HowTo: Tasmota – Automatisches ausschalten des Ausgangs konfigurieren
<https://nerdiy.de/tasmota-nachlaufen-eines-relais-einstellen/>
- Hörrmann Supramatic mit ESP8266 / Tasmota und 433 MHz ansteuern
<https://hoeser-medien.de/2020/05/hoerrmann-supramatic-mit-esp8266-tasmota-und-433-mhz-ansteuern/>
- Hörmann Garage Door Opener mit ESP8266 (direkt über Bus, ohne UAP1 !)
<https://bestofcpp.com/repo/steff393-hgdo>
<https://GitHub.com/steff393/hgdo>
- buergerservice.org: der eID-Türöffner MyPersoDoor
<https://www.youtube.com/watch?v=JQLZyo-OX18>
Nutzt allerdings den ePerso per App (?)

<https://www.youtube.com/watch?v=4g-9ERAf6uQ>

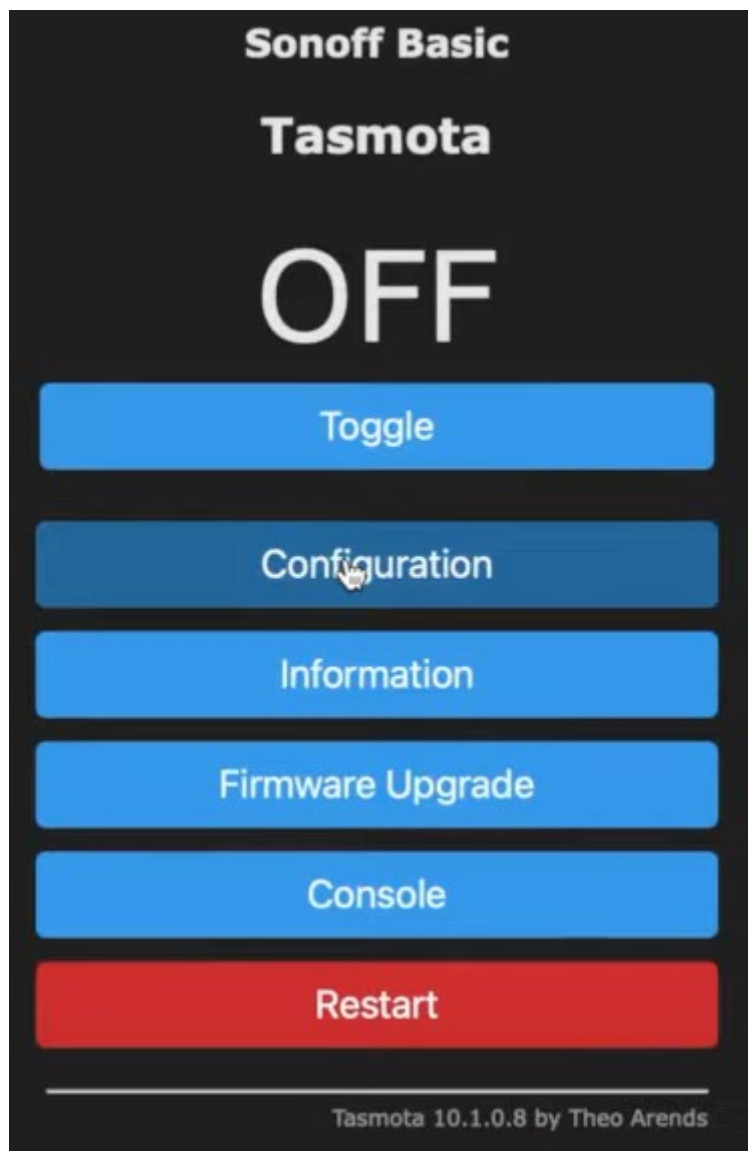
Accessori per Home Assistant - Lettura card e portachiavi con modulo RFID RC522

Tasmota Firmware:

<https://drive.google.com/file/d/1uaEX...>



Konfiguration:



Sonoff Basic

Tasmota

Configure Module

Configure WiFi

Configure MQTT

Configure Domoticz 

Configure Timer

Configure Logging

Configure Other

Configure Template

Reset Configuration

Backup Configuration

Restore Configuration

Main Menu

Sonoff Basic

Tasmota

Module parameters

Module type (Sonoff Basic)

Generic (18) ▾

GPIO1 None ▾

GPIO2 None ▾

GPIO3 None ▾

GPIO4 None ▾

GPIO14 None ▾

Save

Configuration

Tasmota 10.1.0.8 by Theo Arends

Xiaomi Philips (48)

SYF05 (69)

YTF IR Bridge (62)

Witty Cloud (32)

Generic (18)

Configure Moudule:

Module parameters

Module type (Generic)
Generic (18) ▼

D3 GPIO0	None ▼
TX GPIO1	None ▼
D4 GPIO2	None ▼
RX GPIO3	None ▼
D2 GPIO4	None ▼
D1 GPIO5	RC522 Rst ▼
D6 GPIO12	SPI MISO ▼
D7 GPIO13	SPI MOSI ▼
D5 GPIO14	SPI CLK ▼
D8 GPIO15	RC522 CS ▼
D0 GPIO16	None ▼
A0 GPIO17	None ▼

Save

Configure Other

Generic Tasmota

Other parameters

Template

```
{"NAME":"Generic","GPIO":[1,1,1,1,1,1,1,1,
```

☐ **Activate**

Web Admin Password ☐

••••

☒ **HTTP API enable**

☒ **MQTT enable**

Device Name (Tasmota)

RFID Reader

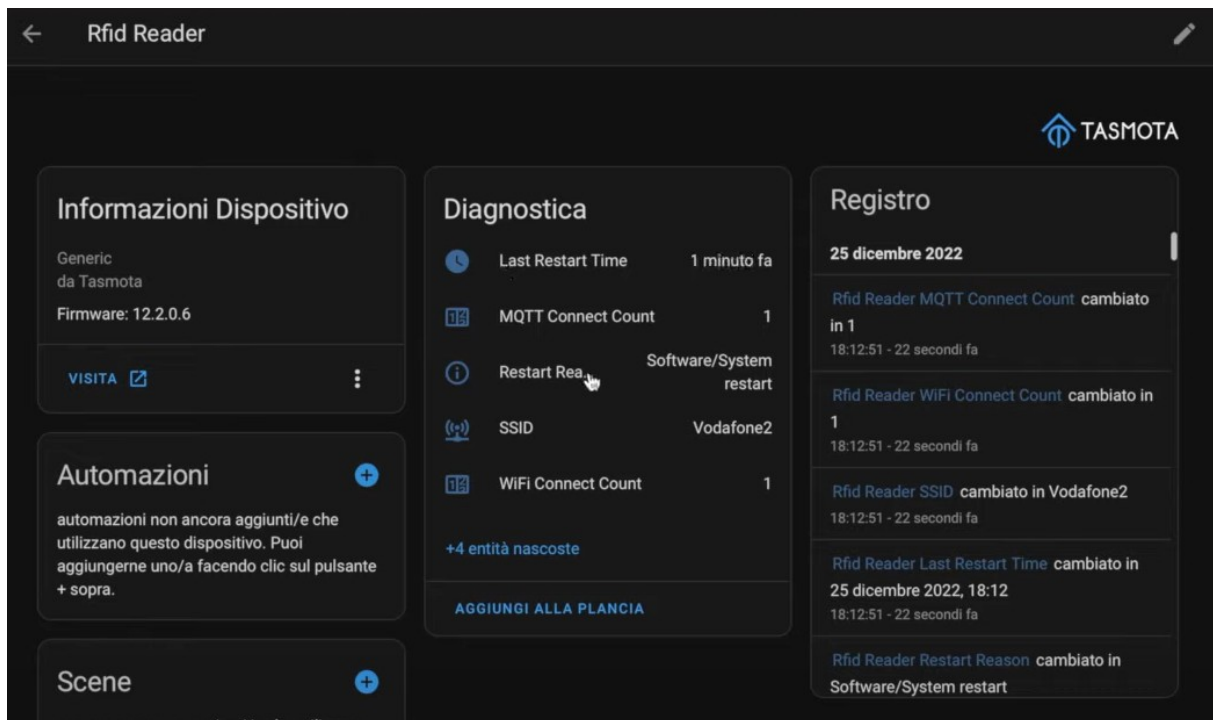
Friendly Name 1 (Tasmota)

RFID Reader

Emulation

- ☒ **None**
- ☐ **Belkin WeMo** single device
- ☐ **Hue Bridge** multi device

Save



file editor

/config/sensori.yaml

Hinzufügen:

```
- platform: mqtt
  name: "RFID"
  state_topic: "tele/rfid/SENSOR"
  value_template: "{{value_json['RC522']['UID']}}"
  device:
    connections: [[mac, '%macaddr%']]
```

<https://forum.creationx.de/forum/index.php?thread/3974-pn532-verschwindet-aus-tasmota/>

An meinem ESP8266 habe ich mir einen RFID (RDM6300) und einen NFC (PN532) angeschlossen.

Beides über die aktuelle Tasmota Software 12.4.0 eingerichtet und es läuft soweit. Unregelmäßig alle paar Stunden verschwindet der PN532 aber aus der Anzeige und wird dann auch nicht mehr ausgelesen.

Erst nach Neustart ist er wieder da und funktioniert.

Habt ihr eine Idee, warum der immer rausgeschmissen wird? Gibt es da eine Lösung für?

Im Moment resette ich den ESP8266 jede Stunde einmal damit ich damit arbeiten/testen kann.

Links mit dem Fehler - rechts nach Reset wieder OK:

Generic

Klingelmodul-RFID

RDM6300 UID

00000000

OFF

Toggle

Configuration

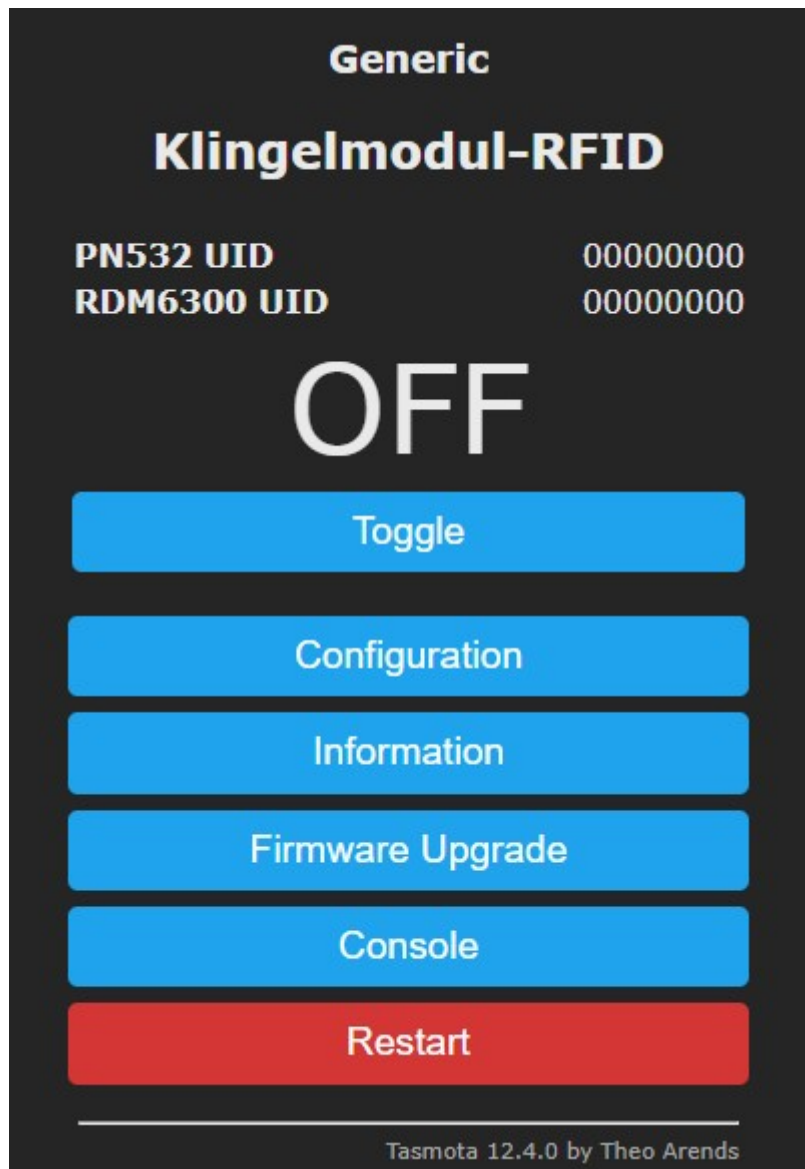
Information

Firmware Upgrade

Console

Restart

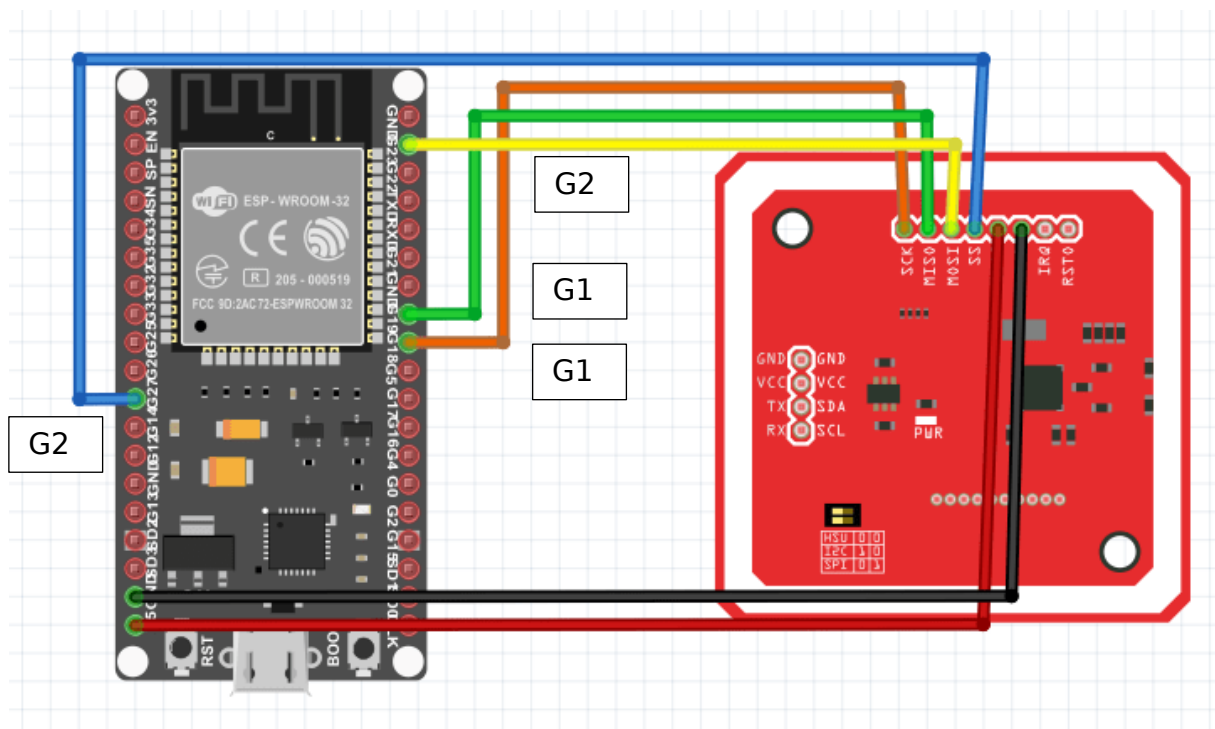
Tasmota 12.4.0 by Theo Arends



<https://forum.creationx.de/forum/index.php?thread/3963-binaries-deutsch-mit-sml-und-script/&postID=42092#post42092>

<https://community.home-assistant.io/t/no-tag-scanned-esp32-with-pn532-testes-oder-spi-and-i2c/348601>

Oct '21



[clod986Clod](#)

I have the issue with the circuit above. The connection between ESP32 and PN523.

[Gompman](#)

I tested with another esp32 board and another rfid board. Same result. I switched back to esp8266 d1 mini. Works like a charm.

[ibarbechIván Barbecho Delgado](#)

I have the same problem with pn532_i2c, pn532_spi never worked for me. I found a solution for pn532_i2c, but it doesn't work for pn532_spi. The solution is add 4.7k pullup resistor to 3.3v for sda and scl.

PN532 stops working after few hours on WEMOS D1 Mini #14236

<https://github.com/arendst/Tasmota/issues/14236>

[DarthWeber](#) commented [on Jan 11, 2022](#)

After soldering the connection instead of using breakout cables the connections seems to be stable. Thanks for all feedbacks.