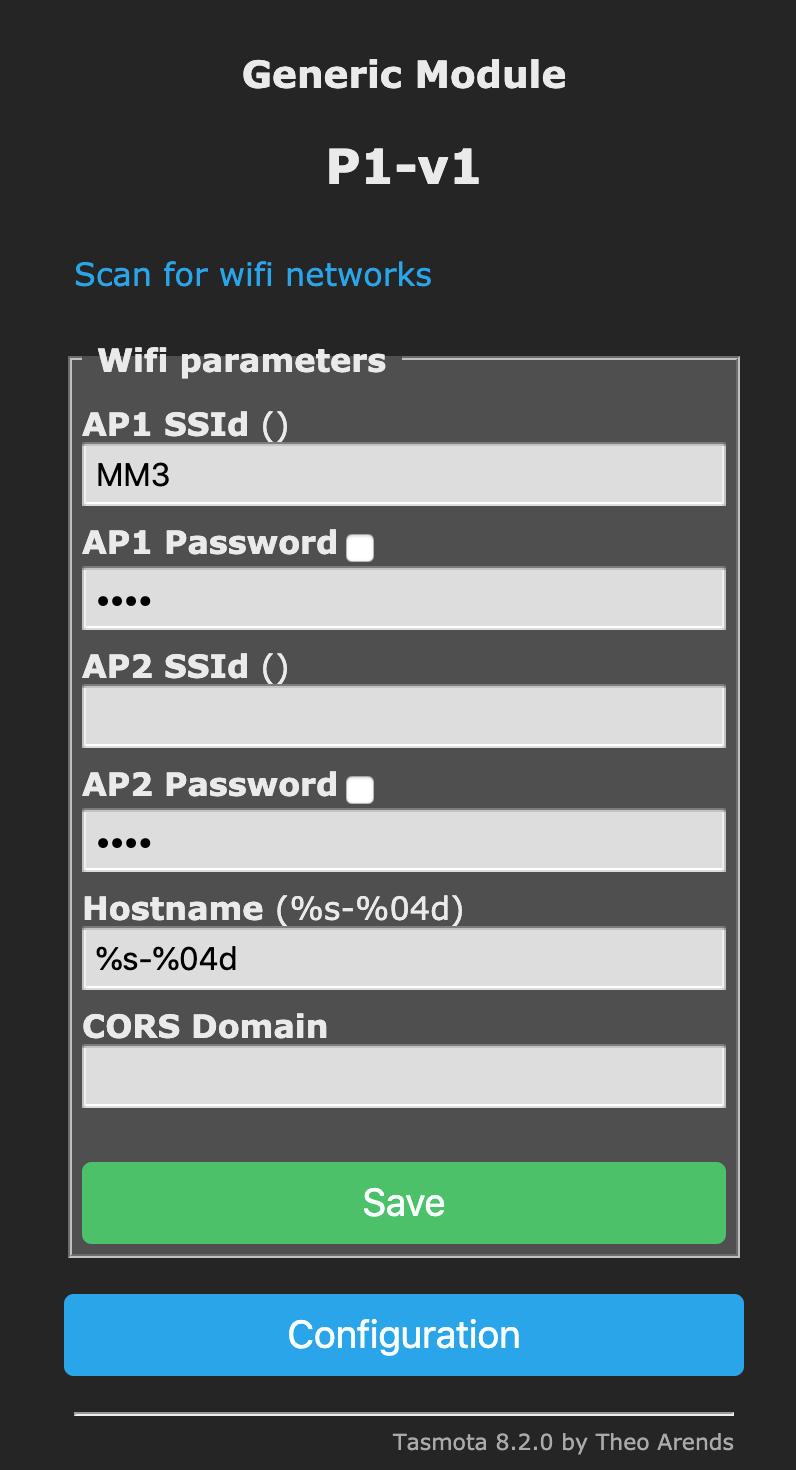
**P1-DSMR adapter Tasmota pre-installed**

De adapter komt met de laatste firmareversie van Tasmota. Op het moment van schrijven is dit 8.2.0.

**Configuratie Wifi**

Na het aansluiten van de adapter op je P1 ingang van je slimme meter gaat het blauwe ledje knipperen. De adapter bouwt als eerste een eigen Wifi hotspot op ten behoeve van configuratie. Deze hotspot is te herkennen door de Wifinaam Tasmota-?????.

Zorg dat je met je computer of mobiel device contact maakt met dit netwerk. Automatisch wordt een updatescherm getoond waarin de Wifi settings te zien zijn. Zie onderstaande plaatje.

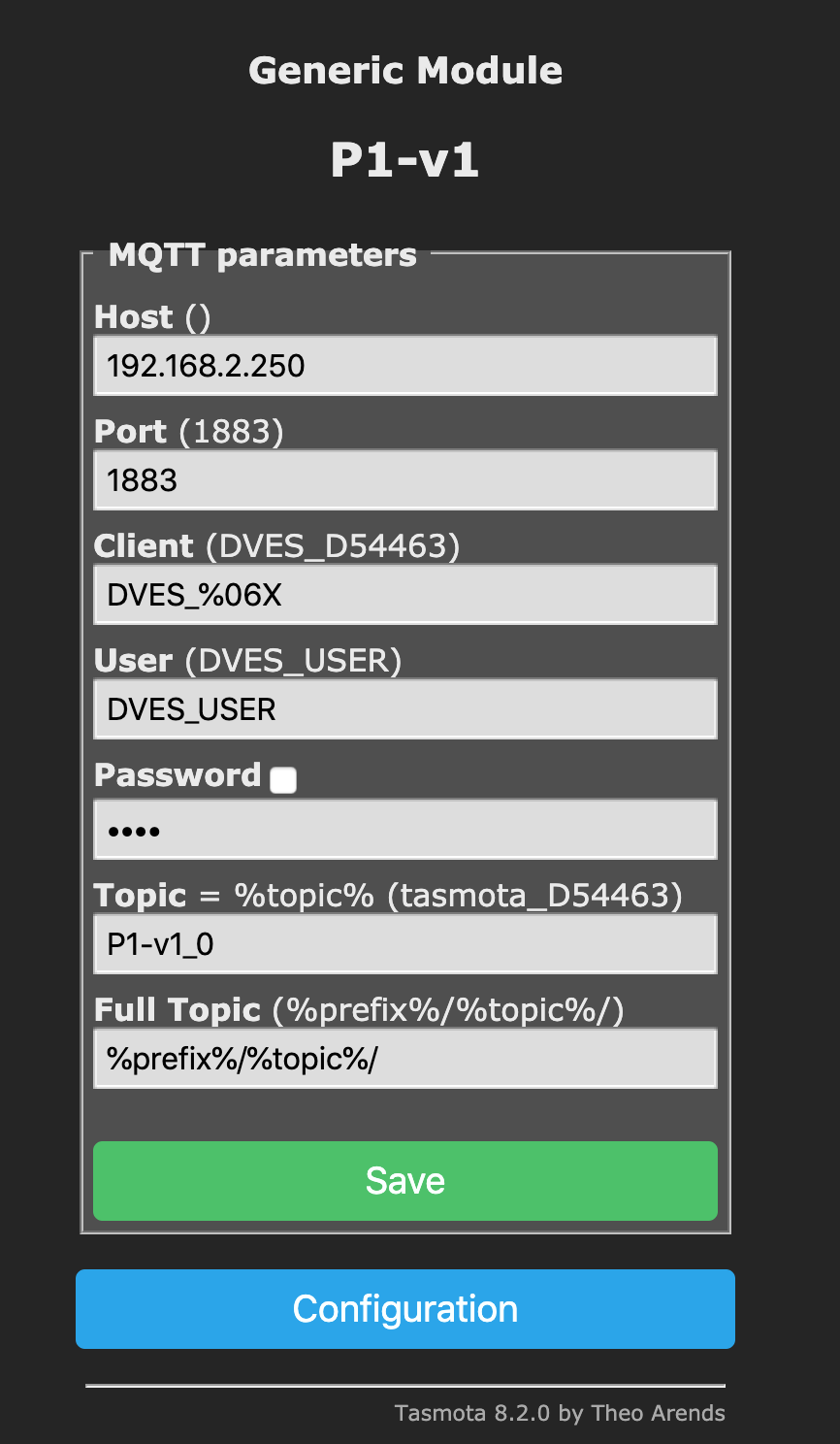


1. Via “scan for wifi kan je networks” gevonden en gekozen worden.
2. Vul daarna het bijbehorende wachtwoord in bij Password.
3. Druk op “Save”
4. Adapter zal op nieuwe opstarten.

Om de overige configuratie te doen dien je het ipadres van de adapter te weten. Dit adres kan je vinden in je router of via diverse andere hulpmiddelen om netwerkapparaten op te kunnen zoeken.

**Configuratie MQTT**

Via het hoofdmenu > Configuration kunnen diverse instellingen van de adapter ingesteld worden. Voor MQTT gebruik dien je naar Configure > MQTT te gaan. Zie een voorbeeld van een configuratie in onderstaande afbeelding



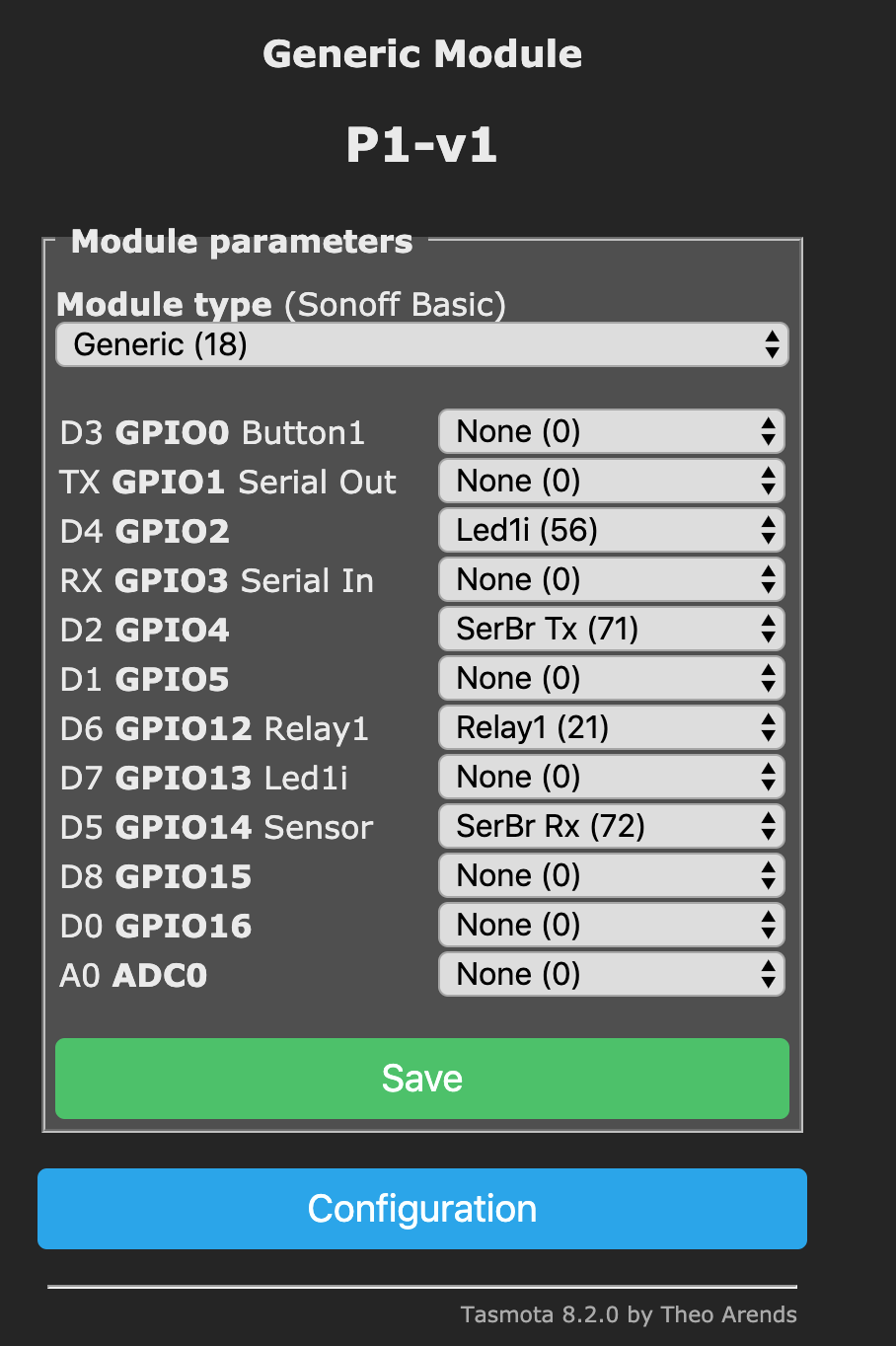
Er dient minimaal het onderstaande ingesteld te worden.

1. Host
2. Port (default 1883)
3. Topic van deze adapter
4. Eventueel wachtwoord en Username
5. Wederom weer op Save drukken en de adapter start opnieuw op

**Configuratie Uitlezen P1 interface**

Standaard is dit al ingesteld op de adapter. Mocht er iets gebeuren met de update of wil je hem opnieuw instellen dan staan hieronder de verschillende parameters.

#1 module parameters; zie hieronder.



#2 Overige configuratie

De overige parameters worden via de Console ingesteld. Ga in het hoofdmenu > Console.

Type hierin de volgende commando’s:

* SerialDelimiter 44 (“/” is scheidingsteken … dwz mqtt bericht start met “/” in plaats van “//”
* SBaudrate 115200
* SSerialSend 2
* Rule1 on System#Boot do backlog power1 0; ruletimer1 5 endon on power1#state=1 do backlog ruletimer1 5; delay 1; power1 0; endon on rules#timer=1 do power1 1 endon Rule1 1
* power 1

Bovenstaande setup vraagt elke 5 seconde een request (Datarequest signaal). Door ruletimer1 de 5 (2x) aan te passen naar een ander geheel getal kan je de update frequentie aanpassen. Het snelst kan dit 1x per seconde. Dan kan rule1 helemaal verwijderd worden en de power op 1 zetten. De DataRequest wordt dan altijd hoog gehouden door de adapter. Let dan wel op dat de adapter bij reboot automatisch de datarequest hoog zet. Dit kan op diverse manieren maar makkelijkst is met onderstaande rule. Delay 1 is nodig omdat anders een deel van het telegram verloren gaat.

* Rule1 on System#Boot do power1 1 endon

**Update Tasmota**

Om de module te updaten is het voldoende om in het volgende te doen:

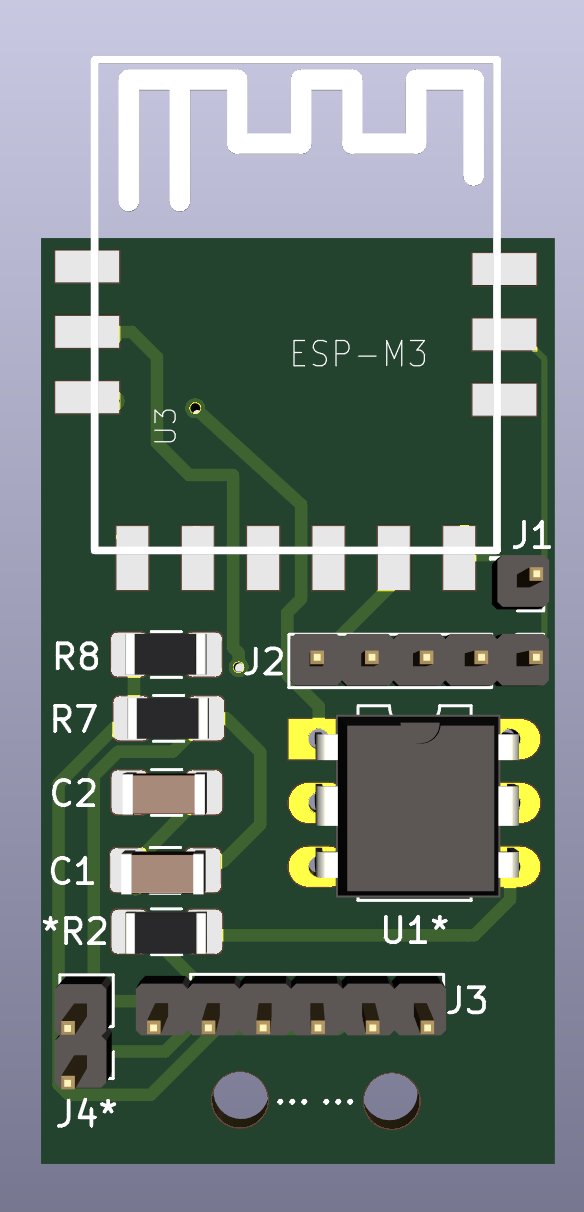
* Kies in hoofdmenu voor Firmware Update
* Kies in volgende menu voor eerste Start Upgrade knop
* De ingevulde OTA url is <http://thehackbox.org/tasmota/release/tasmota-lite.bin>

Het is wel nodig dat de module een verbinding naar buiten kan maken. In de module lang niet geupdate is. Bijvoorbeeld 1 jaar dan kan het zo zijn dat de module niet meer via de OAT url te updaten is. Vaak dient er dan eerst een tussen versie (of meerdere) geïnstalleerd te worden. Dit is terug te vinden op de site van Tasmota. Deze tussen versie(s) dient dan via een file upload geïnstalleerd te worden.

**Flashen**

Flashen dient via een FTDI interface te gebeuren. Op J2 zitten de aansluitingen voor deze interface. De pinout is :

1. Flash (vierkante pad / Rode pijl); Flash naar GND en opnieuw opstarten om in de program mode te komen
2. 3.3Volt
3. GND
4. TX
5. RX



J1 is de reset. Reset = GND

**Instellen OpenHab 2.4+**

Zie setup hieronder. Let op! Er is meer data beschikbaar die getoond kan worden.

**Mqtt.things:**

// P1 - slimmemeter2

Thing topic slimmemeter2 "Slimmemeter2" @ "Hal" {

Channels:

Type string : p1 "kWh Teller 1" [ stateTopic="tele/P1-v1\_0/RESULT", transformationPattern="JSONPATH:SSerialReceived" ]

}

**Default.items:**

String P1 "test [%s]" {channel="mqtt:topic:mosquitto:slimmemeter2:p1" }

Number P1\_P "Actueel vermogen [%s]" (GRP\_energy)

Number P1\_GAS "Gasmeter: [%.1f]"

Number P1\_T1 "kWh Teller 1: [%s]"

Number P1\_T2 "kWh Teller 1: [%s]"

Number P1\_I "Stroom: [%.2f A]"

**Default.rules:**

rule "P1 P"

when

Item P1 received update

then

if (P1.state == NULL || P1 == "") return;

val rawString = P1.state.toString

//P1\_I 1-0:31.7.0(002\*A)

val rawString3 = transform("REGEX", ".\*([0-9]{3}).A.\*", rawString)

if ( rawString3 != NULL && rawString3 !== null && rawString3 != "") {

P1\_I.postUpdate(rawString3)

}

//P1\_T1 1-0:1.8.1(000013.000\*kWh) (Totaal geleverd tarief 1 (nacht))

rawString3 = transform("REGEX", ".\*1.8.1.([0-9]{6}.[0-9]{3}).kWh.\*", rawString)

if ( rawString3 != NULL && rawString3 !== null && rawString3 != "") {

P1\_T1.postUpdate(rawString3)

}

//P1\_T2 1-0:1.8.2(000084.000\*kWh) (Totaal verbruik tarief 2 (dag))

rawString3 = transform("REGEX", ".\*1.8.2.([0-9]{6}.[0-9]{3}).kWh.\*", rawString)

if ( rawString3 != NULL && rawString3 !== null && rawString3 != "") {

P1\_T2.postUpdate(rawString3)

}

//P1\_GAS 0-1:24.2.1(191222151005W)(02728.449\*m3)

rawString3 = transform("REGEX", ".\*[0-9]{12}.\*([0-9]{5}.[0-9]{3}).m3.\*", rawString)

if ( rawString3 != NULL && rawString3 !== null && rawString3 != "") {

P1\_GAS.postUpdate(rawString3)

}

// P1\_P 1-0:21.7.0(00.523\*kW)

rawString3 = transform("REGEX", ".\*21.7.0.([0-9]{2}.[0-9]{3}).kW.\*", rawString)

if ( rawString3 != NULL && rawString3 !== null && rawString3 != "") {

P1\_P.postUpdate(rawString3)

}

end

**Default.sitemap:**

Frame label="Verbruik" {

Text item=P1\_T1 label="kWh Teller 1: [%.1f kWh]"

Text item=P1\_T2 label="kWh Teller 2: [%.1f kWh]"

Text item=P1\_P label="Actuele vermogen: [%.3f kW]"

Text item=P1\_I label="Actuele stroom: [%.1f A]"

Text item=P1\_GAS label="Gasmeter: [%.1f m3]"

Group item=GRP\_energy

Switch item=P\_charts label="Vermogen" icon=line mappings=[0="Hide", 1="Hour", 2="Day", 3="Week"]

Chart item=Power period=h refresh=5000 visibility=[P\_charts==1]

Chart item=Power period=D refresh=30000 visibility=[P\_charts==2]

Chart item=Power period=W refresh=30000 visibility=[P\_charts==3]

}

**Resultaat**

