

ITHO Ventilatie bedienen en monitoren met Home Assistant of Domoticz

Auteur: Arnout O.

Bijgewerkt door: Sten Vollebregt (2/2021)

[Wat heb je nodig ?](#)

[Hardware klaar maken voor gebruik](#)

[Software installatie ESPEasy](#)

[Software configuratie](#)

[Integratie Home Assistant](#)

[Integratie Domoticz](#)

[Dankwoord](#)

Wat heb je nodig ?

1. a. Home Assistant met MQTT server.

<https://www.home-assistant.io/getting-started/>

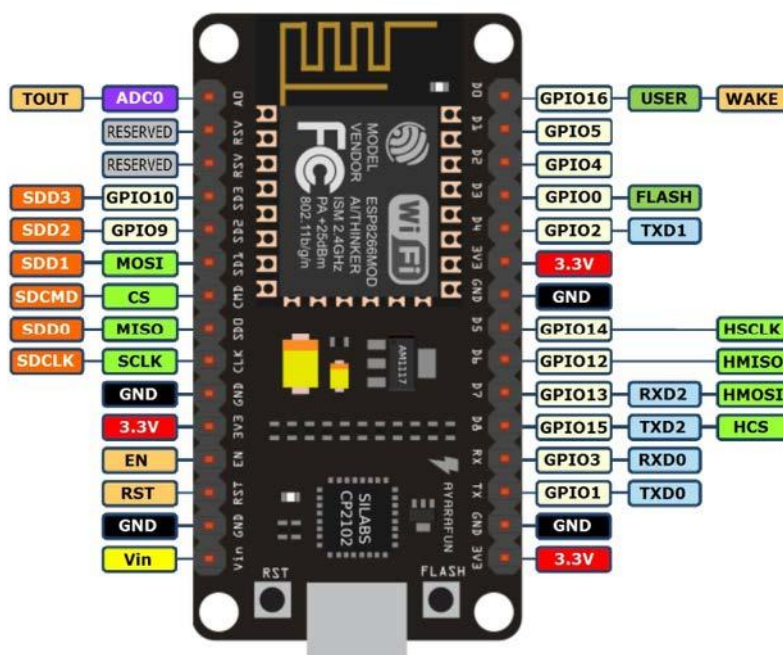
-of-

- b. Domoticz met MQTT server

<https://www.domoticz.com/wiki/MQTT>

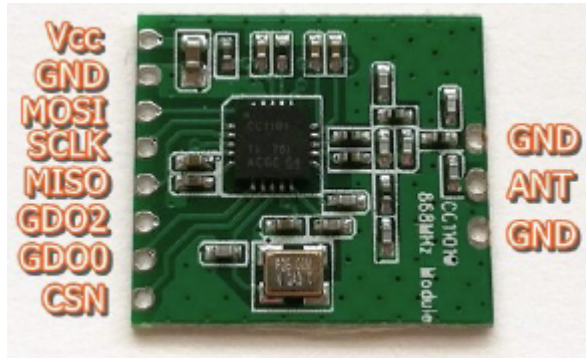
2. Een ESP bordje

[https://www.tinytronics.nl/shop/nl/communicatie/rf\(id\)-wifi-bt/esp8266-nodemcu-v2?search=NODEMCUCP2102](https://www.tinytronics.nl/shop/nl/communicatie/rf(id)-wifi-bt/esp8266-nodemcu-v2?search=NODEMCUCP2102)



3. Een CC1101 866 Mhz radio

<https://www.ebay.nl/itm/1PCS-NEW-CC1101-wireless-module-Long-Distance-Transmission-Antenna-868MHZ/381475961886?hash=item58d1bd3c1e:g:FmIAOSwAy5avdf9:rk:1:pf:0>



4. Breadboard DUPONT cable

https://nl.aliexpress.com/wholesale?catId=0&initiative_id=SB_20190119015514&SearchText=dupont+breadboard



5. Micro USB Voeding - 5V 1A (tenminste)

<https://www.tinytronics.nl/shop/nl/voedingen/5v/goobay-46600-micro-usb-voeding-5v-1a-zwart?search=micro%20USB>

6. CP210x USB to UART Bridge VCP Drivers (silabs

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

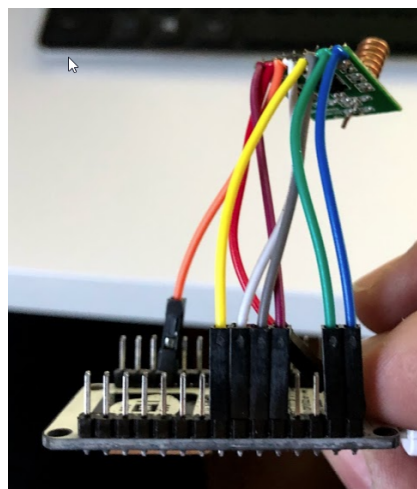
7. ESP Easy (letscontrolit)
<https://espeasy.readthedocs.io/en/latest/>
8. ESPEASY_Plugin_ITHO binary (gemaakt door jodur, aangepast door svollebrecht)
https://github.com/svollebrecht/ESPEASY_Plugin_ITHO

Hardware klaar maken voor gebruik

Voor het solderen van de dupont cable kun je de volgende layout gebruiken

CC11xx pin	ESP pins	Description
1 - VCC	VCC	3.3V
2 - GND	GND	Ground
3 - MOSI	13=D7	Data input to CC11xx
4 - SCK	14=D5	Clock pin
5 - MISO/GDO1	12=D6	Data output from CC11xx / serial clock from CC11xx
6 - GDO2	05=D1*	Interrupt pin
7 - GDO0	NC	Niet in gebruik
8 - CSN	15=D8	Chip select / (SPI_SS)

* Voor GDO2 kan je ook een andere pin gebruiken als D1 reeds in gebruik is voor I2C. Vermijd echter: D3 en D4 (beïnvloeden het opstarten van de ESP) en D0 (geen interrupts). Zorg er voor dat je I2C uit zet in ESPEasy als je D1 of D2 gebruikt. Gebruik ook niet GPIO-6 t/m 11 van de ESP, aangezien deze *meestal* gebruikt worden door de Flash ROM



Software installatie ESPEasy

1. Sluit ESP aan via usb en zet hem in flash modus. Houd het knopje flash ingedrukt terwijl je reset kort indrukt.
2. Flash de firmware met ESPEasy en de meest recente binary file van svollebregt https://github.com/svollebregt/ESPEASY_Plugin_ITHO.
3. Restart de ESP.
4. Verbindt via wifi met de ESP. SSID **ESP_Easy_0** en wachtwoord **configesp**
5. Een webpagina verschijnt automatisch (vermoedelijk ip 192.168.4.1) waarop je eigen wifi netwerk kunt instellen (als je meerdere AP's hebt dan verschijnen deze meerdere keren met hetzelfde SSID)
6. Verbindt opnieuw met je ESP via het IP adres dat hij heeft gekregen van jouw router (advies maak een mac adres reservering, zodat hij altijd hetzelfde IP adres heeft)
7. De volgende pagina verschijnt:

ESP Easy Mega: ITHO

[Main](#) [Config](#) [Controllers](#) [Hardware](#) [Devices](#) [Notifications](#) [Tools](#)

System Info	Value		System Info	Value	
Unit:	0		GIT version:	(custom)	
Local Time:	NTP disabled		Uptime:	0 days 19 hours 28 minutes	
Load:	10% (LC=22984)		Free Mem:	20304 (10680 - sendWebPageChunkedData)	
IP:	192.168.1.137		Wifi RSSI:	-79 dB	

Node List:	Name	Build	Type	IP	Age
------------	------	-------	------	----	-----

Powered by www.letscontrolit.com

Software configuratie

1. Stel onder **Config** de **Unit Name** in, ik heb gekozen voor **ITHO** en klik onderaan de pagina op **Submit**:

ESP Easy Mega: ITHO

[Main](#) [Config](#) [Controllers](#) [Hardware](#) [Devices](#)

Main Settings

Unit Name:

ITHO

Unit Number:

0

Admin Password:

2. Stel onder **Controllers** de gegevens van je MQTT server in. Klik op de eerste regel op **Edit** en vul jouw gegevens in, vink **Enabled** aan en klik op **Submit**..

ESP Easy Mega: ITHO

Main

Config

Controllers

Hardware

Controller Settings

Protocol: OpenHAB MQTT ?

Locate Controller: Use IP address

Controller IP: ip.nummer.mqtt.server

Controller Port: 1883

Controller User: username.mqtt

Controller Password:

Controller Subscribe: %sysname%/#

Controller Publish: %sysname%/%tskname%/%c

Enabled: ☒

Close

Submit

Controller Subscribe is het topic waar ESP naar luistert. Wordt dus gebruikt om aan te sturen vanuit HomeAssistant

Controller Publish is het topic waarop de ESP zijn status adverteert

PS Ik heb **geen slash** staan voor de topics

%sysname% wordt vervangen door **ITHO**

%tskname% wordt straks **Fan**

3. Zorg er voor dat onder **Hardware** er een vinkje staat bij **Init SPI**

4. Stel onder **Devices** de gegevens van de remote in. Klik op de eerste regel (task) op **Edit** en kies onder **Device** voor **Itho ventilation remote**. Type bij **Name** "Fan". Vink **Enabled** aan. En kies bij **GPIO ← Interrupt pin** voor **GPIO-5 (D1)** (of de pin die je gebruikt voor GDO2, in onderstaande screenshot is dat D2).

Task Settings

Device: Itho ventilation remote ? i

Name:

Enabled: ☒

Sensor

GPIO ← Interrupt pin (CC1101 GDO2):

Remote RF Controls

Unit ID remote 1:

Unit ID remote 2:

Unit ID remote 3:

Enable RF receive log: ☒

Remote SYNC1 byte:

Note: Sync byte for remote, known good values: 170 (default, remote with timer) and 172 (remote

Data Acquisition

Send to Controller ☐

IDX:

Interval: [sec]

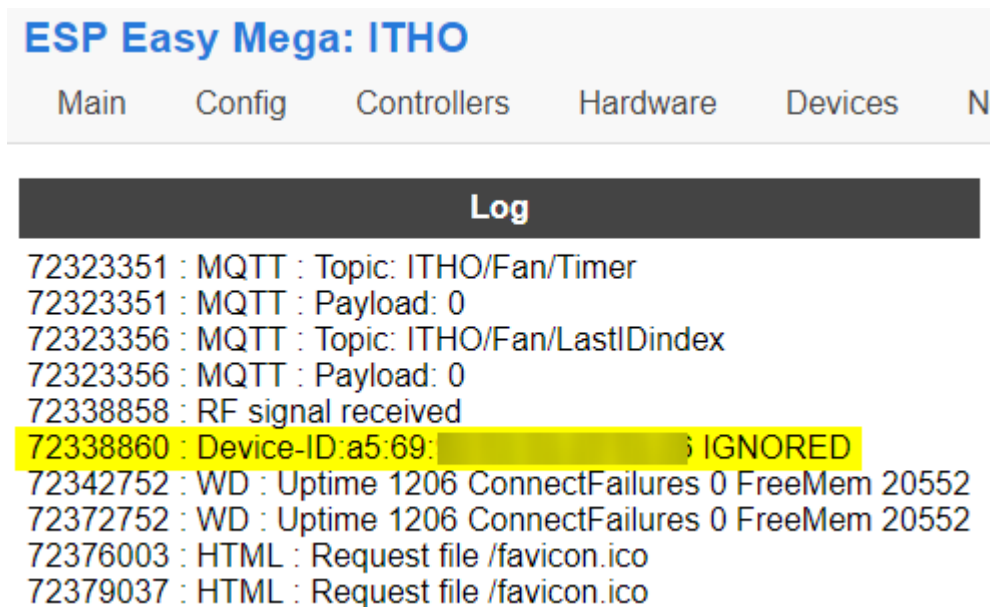
Values

#	Name
1	State
2	Timer
3	LastIDindex

Opmerkingen:

- Onder *Unit ID remote X* kan je de IDs van je Itho afstandsbedieningen plaatsen. **Let op:** bij de nieuwste versie (newlib) zijn dit nog maar 3 bytes gescheiden door komma's ipv ':'.
- Met *Enable RF receive log* zet je RF pakketjes loggen aan, dit is nodig voor het bepalen van je afstandsbediening (punt 5) maar daarna kan je het uitzetten
- *Remote SYNC1 byte* zet je standaard op 170. Mocht je Itho remote telkens een net iets andere waarde geven probeer dan 172. (deze optie is verdwenen in de newlib binary die dit niet meer nodig heeft)

- **Zet *Send to Controller* aan als je Home Assistant gebruikt, voor Domoticz is dit niet nodig**
5. ESP koppelen aan je ITHO Ventilatie box. Maak de ventilator even spanningsloos, hij komt nu 2 minuten in een modus waarin je een nieuwe afstandsbediening kunt koppelen.
 - a. Verstuur vanuit je browser het join commando <http://YourIP-adress/control?cmd=STATE.1111> Als het goed is hoor je de ventilator reageren.
 - b. Fan vol aan, <http://YourIP-adress/control?cmd=STATE.4>
 - c. Fan laag, <http://YourIP-adress/control?cmd=STATE.1>
 6. Zet je logging nu op **Debug** onder **Tools** -> **Advanced** -> **Web log level** en klik onderaan de pagina op **Submit**.
In **Tools** -> **Log** zie je als het goed is informatie voorbij komen als je een knop op je bestaande **originele afstandsbediening** drukt.



Bij mij is het kennelijk best druk op het 866 Mhz netwerk, dus ik moest een paar keer proberen voordat ik het juiste **Device-ID** had.

Als je dat **Device-ID** instelt bij **Devices** -> **Edit #1** -> **Remote RF devices** -> **Unit ID remote 1** dan kan je ESP ook de opdrachten ontvangen van je originele afstandsbediening van ITHO zodat Home Assistant of Domoticz ook op de hoogte wordt gebracht van de status via MQTT

Integratie Home Assistant

De volgende YAML heb ik als package toegevoegd aan Home Assistant.

Let op: als je er voor kiest om de MQTT Topic aan te passen, let er dan op dat dit *CASE Sensitive* is. FAN of fan is dus niet Fan.

1. Als je nog geen gebruik maakt van Packages binnen Homeassistant voeg dan in de configuration.yaml het volgende toe:

```
homeassistant:
  packages: !include_dir_named packages
```

2. Maak in de config directory van Home Assistant een directory **Packages** aan en een bestand itho.yaml (packages\itho.yaml).

Update: sinds HA 2021.4 is de code veranderd door een aanpassing in HA:

```
- platform: mqtt
name: Ecofan
command_topic: "ITHO/Fan/cmd"
state_topic : "ITHO/Fan/State"
state_value_template: "{% if value|float==0 %}State 0{%
endif %}{% if value|float >0 %}State 1{% endif %}"
payload_on: "State 1"
payload_off: "State 0"
optimistic: true
preset_modes:
  - "off"
  - "low"
  - "medium"
  - "high"
  - "full"
preset_mode_command_topic: "ITHO/Fan/cmd"
preset_mode_command_template: >
  {% if value == 'low' %}
  State 1
  {% elif value == 'medium' %}
  State 2
  {% elif value == 'high' %}
```

```

        State 3
        {% elif value == 'full' %}
        State 4
        {% else %}
        State 0
        {% endif %}
    preset_mode_state_topic: "ITHO/Fan/State"
    preset_mode_value_template: >
        {% if value_json.fan == low %}
        State 1
        {% elif value_json.fan == medium %}
        State 2
        {% elif value_json.fan == high %}
        State 3
        {% elif value_json.fan == full %}
        State 4
        {% else %}
        State 0
        {% endif %}

```

Oude code:

```

sensor:
- platform: mqtt
  name: Tijd
  state_topic: "ITHO/Fan/Timer"
  value_template: "{{value}}"
  unit_of_measurement: 's'
- platform: mqtt
  name : Snelheid
  state_topic: "ITHO/Fan/State"
  value_template: >
    {% if value|float==0 %}Standby{% endif %}
    {% if value|float==1 %}Laag{% endif %}
    {% if value|float==2 %}Medium{% endif %}
    {% if value|float==3 %}Hoog{% endif %}
    {% if value|float==4 %}Full{% endif %}
    {% if value|float>=11 %}Hoog(T){% endif %}

fan:
- platform: mqtt

```

```

    command_topic: "ITHO/Fan/cmd"
    state_topic : "ITHO/Fan/State"
    speed_command_topic: "ITHO/Fan/cmd"
    speed_state_topic: "ITHO/Fan/State"
    state_value_template: "{% if value|float==0 %}State 0{%
endif %}{% if value|float >0 %}State 1{% endif %}"
    speed_value_template: "{% if value|float==0 %}State 0{%
endif %}{% if value|float >0 and value|float <=4 %}State
{{value}}{% endif %}{% if value|float >=11 %}State 3{%
endif %}"
    payload_on: "State 1"
    payload_off: "State 0"
    payload_low_speed: "State 1"
    payload_medium_speed: "State 2"
    payload_high_speed: "State 3"
    optimistic: "true"
    name: Afzuiging badkamer
    speeds:
      - low
      - medium
      - high

group:
  Ventilator:
    entities:
      - fan.afzuiging_badkamer
      - sensor.Snelheid
      - sensor.Tijd

```

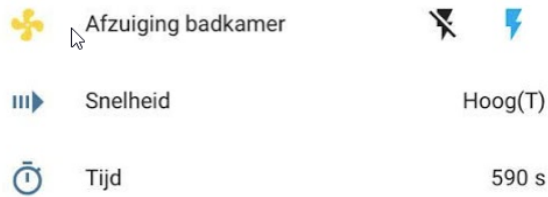
4. Voor mooie icoontjes bij **Tijd** en **Snelheid** kun je de **customize.yaml** nog aanpassen:

```

sensor.tijd:
  icon: mdi:timer
sensor.snelheid:
  icon: mdi:transfer-right

```

Ventilator



Integratie Domoticz

1. Domoticz maakt gebruik van een andere controller in ESPEasy. In plaats van openHAB moet je de Domoticz MQTT of Domoticz HTTP controller gebruiken. Deze handleiding gaat uit van de eerste. Voor meer info:
<https://www.letscontrolit.com/wiki/index.php/EasyProtocols>
2. In Domoticz kan je vervolgens een dummy switch aanmaken (https://www.domoticz.com/wiki/Dummy_for_virtual_Switches). Kies voor een Selector switch en gebruik bijvoorbeeld de volgende namen

Idx: 262

Name: Ventilatie

Switch Type: Selector

Switch Icon:  **Fan**
Airflow Producer

On Delay: 0 (Seconds) 0 = Disabled

Off Delay: 0 (Seconds) 0 = Disabled

Protected: ☐

Selector Style: ☒ Button set ☐ Select menu

Hide Off level: ☒

Selector Levels:

Level	Level name
0	Off
10	Low
20	Medium
30	High
40	Full
50	Timer 10
60	Timer 20
70	Timer 30

- De volgende stap is om een verandering van de ESPEasy naar Domoticz te versturen. Helaas wordt dit niet standaard ondersteund in de MQTT of HTTP Domoticz controller. Het is echter heel eenvoudig om dit via de ESPEasy rules te doen (<https://espeasy.readthedocs.io/en/latest/Rules/Rules.html>):

```
on Itho#State do
  if [Itho#State]=0
    Publish domoticz/in,'{"command":"switchlight","idx":IDX,"switchcmd":"Set Level","level":0}'
  elseif [Itho#State]=1
    Publish domoticz/in,'{"command":"switchlight","idx":IDX,"switchcmd":"Set Level","level":10}'
  elseif [Itho#State]=2
    Publish domoticz/in,'{"command":"switchlight","idx":IDX,"switchcmd":"Set Level","level":20}'
```

```

elseif [Itho#State]=3
  Publish domoticz/in,'{"command":"switchlight","idx":IDX,"switchcmd":"Set Level","level":30}'
elseif [Itho#State]=4
  Publish domoticz/in,'{"command":"switchlight","idx":IDX,"switchcmd":"Set Level","level":40}'
elseif [Itho#State]=13
  if [Itho#Timer]=599
    Publish domoticz/in,'{"command":"switchlight","idx":IDX,"switchcmd":"Set Level","level":50}'
  endif
elseif [Itho#State]=23
  if [Itho#Timer]=1199
    Publish domoticz/in,'{"command":"switchlight","idx":IDX,"switchcmd":"Set Level","level":60}'
  endif
elseif [Itho#State]=33
  if [Itho#Timer]=1799
    Publish domoticz/in,'{"command":"switchlight","idx":IDX,"switchcmd":"Set Level","level":70}'
  endif
endif
endof
endon

```

Hier moet je voor IDX het nummer van je dummy switch invullen. Dit scriptje zal bij een verandering (en 1 keer per minuut) de stand van je Itho box zoals bekend bij de ESP doorsturen naar je Domoticz door de level van je switch aan te passen.

4. Nu moeten we nog Domoticz commando's naar de ESP kunnen laten sturen, zodat je ook via Domoticz je ventilatie kan aansturen. Hiervoor maak ik gebruik van een dzVents script:

```

return {
  on = {
    devices = {
      'Ventilatie'
    },
    data =
    {
      previousLevel = { initial = 10 }
    },
    execute = function(domoticz, device)
      if domoticz.data.previousLevel ~= device.level then
        domoticz.log('Fan state change')
        if device.level == 10 then
          domoticz.log('Fan now Low')
          domoticz.openURL('http://<espi>/control?cmd=STATE,1')
          domoticz.data.previousLevel = 10
        elseif device.level == 20 then
          domoticz.log('Fan now Medium')
          domoticz.openURL('http://<espi>/control?cmd=STATE,2')
          domoticz.data.previousLevel = 20
        elseif device.level == 30 then
          domoticz.log('Fan now High')
          domoticz.openURL('http://<espi>/control?cmd=STATE,3')
          domoticz.data.previousLevel = 30
        elseif device.level == 40 then
          domoticz.log('Fan now Full')
          domoticz.openURL('http://<espi>/control?cmd=STATE,4')
          domoticz.data.previousLevel = 40
        elseif device.level == 50 then
          domoticz.log('Fan now Timer 10')
          domoticz.openURL('http://<espi>/control?cmd=STATE,13')

```

```

        domoticz.data.previousLevel = 50
    elseif device.level == 60 then
        domoticz.log('Fan now Timer 20')
        domoticz.openURL('http://<espi>/control?cmd=STATE,23')
        domoticz.data.previousLevel = 60
    elseif device.level == 70 then
        domoticz.log('Fan now Timer 30')
        domoticz.openURL('http://<espi>/control?cmd=STATE,33')
        domoticz.data.previousLevel = 70
    end
end
end
}

```

Hier vul je voor <espi> het IP adres van je ESP in. Bij een verandering stuurt Domoticz een commando naar de ESP om de fan aan te sturen.

Voor meer info over dzVents:

https://www.domoticz.com/wiki/DzVents:_next_generation_Lua_scripting

Dankwoord

Deze handleiding heb ik gemaakt met behulp van de documentatie die ik heb gehad van mijn collega Mories, online van jodur en natuurlijk mijn zwager Joost die zeer handig is met de soldeerbout ;-)

Meer info: https://gathering.tweakers.net/forum/list_message/66189038