Tasmota Sync Drivers

A GUIDE FOR INSTALLATION AND USE

Gary Milne

Contents

Tasmota Sync – Device Drivers for Hubitat

Automatic Synchronization between Tasmota and Hubitat without custom Tasmota

TLDR

These drivers allow Hubitat to synchronize with Tasmota (11 onwards) without the need for any special Tasmota compilation or use of HTTP hooks. A rule installed onto Tasmota by the Hubitat driver causes Tasmota to send a synchronization request whenever a monitored setting changes. Changes made on Tasmota will be reflected on Hubitat within 1-2 seconds. No polling or custom Tasmota build required. Interested? Read on.

Intro

I initially wrote a Tasmota device driver for the Smartthings environment which became obsolete when the Smartthings platform shutdown the use of all Classic drivers. When I came to Hubitat I was going to adapt my Tasmota driver until I found that Markus Li had already written a sophisticated App\Driver architecture for Tasmota integration. I liked what Markus had done so much that I wrote his T4HE documentation for him.

Fast forward a few years and things have changed.

Markus has been banned from Hubitat and there will be no further development or support for his software from him.

Markus' Tasmota drivers are linked to an alternate compilation of Tasmota which is forever fixed at Tasmota 8.X. Tasmota has since move to version 11.1 and keeps evolving.

Over a year of using Markus's software has highlighted several additional areas where I felt there could be improvement.

- The parent\child structure is unwieldy and cumbersome when almost every parent has only a single child device.
- There are several aspects of the drivers that are not fully functional, especially the bulb driver. Aspects such as Fade and CT simply do not work in my experience.
- Markus' drivers tend to be quite chatty and pass a lot of data resulting in a lot of logging and events recorded for each device. This is especially true for power managed or sensor devices.
- Markus' App\Parent\Child architecture is very dense, challenging to understand and difficult to customize without introducing unexpected consequences given it's universal nature.
- My original Smartthings classic driver had several conveniences such as Fade, CT and Color preset values that I found that I missed.

The Tasmota Sync driver family addresses all of these issues and most importantly frees people to upgrade to current versions of Tasmota across their devices. That said, it does not replicate all of the features of Markus' Tasmota drivers which handled auto discovery of devices and their capabilities. The Tasmota Sync drivers require the user to match the appropriate driver with the device.

Architecture

Quite simply the Hubitat driver for a given Tasmota device injects a rule into Tasmota that causes Tasmota to report any changes to the parameters of interest. Below is an example of a rule that monitors a dual plug. You don't need to understand it, just know it is required for Tasmota Sync to work. Each type of Tasmota device has a rule that is managed by the driver.

```
ON Power1#State DO backlog0 Var13 %value% ; RuleTimer1 1 ENDON
ON Power2#state DO backlog0 Var14 %value% ; RuleTimer1 1 ENDON
ON Rules#Timer=1 DO Var15 %Var13%,%Var14% ENDON
ON Var15#State$!%Var16% DO backlog ; Var16 %Var15% ; webquery http://192.168.0.200:39501
POST {'TSync':'True','Switch1':'%Var13%','Switch2':'%Var14%'} ENDON
```

Drivers

I have 7 completed drivers in the Tasmota Sync family all of which are built on the same core driver. All of these have been fully tested in my home environment and have been performing very well for the last few weeks.

1. Color Bulb
2. Dimmer Switch
3. Dual Switch\Dual Plug
4. Fan
5. Switch\Plug (PM optional)
6. Switch with Sensor
7. Sensor

The Tasmota Sync family drivers are located here: https://github.com/GaryMilne/Hubitat-Tasmota The Tasmota Sync drivers all share a lot of core functions and are internally consistent for easier code maintenance. Any changes to the "core" functions in the bulb driver would be incorporated in the other drivers prior to release.

Tasmota

This document does not attempt to provide any education on Tasmota, there are many other resources for that. For Tasmota Sync drivers to operate fully you need to be on Tasmota 11 as this supports the Webquery command and thus Tasmota Sync. If you need a little guidance on how to upgrade from Tasmota 8.X to 11.X you may find this guide useful, "How to Upgrade from Tasmota 8.X to Tasmota 11.X".

You can use Tasmota Sync drivers with prior versions of Tasmota, but this would only operate in polling mode which is not enabled by default. Once you have updated a device to Tasmota 11 you are ready to install the Tasmota Sync drivers into your Hubitat environment.

You can find the current releases of Tasmota here: http://ota.tasmota.com/tasmota/release/

Installing a New Device

This section gives a detailed example of how-to setup a new device. In this example we are using a bulb, but the same principles are true for all devices. Subsequent instructions for specific new devices will be in abbreviated form.

Driver Installation

The Tasmota Sync drivers all install like any other Hubitat driver. In this case I will use the Bulb Driver but the same steps are true for all of the Tasmota Sync Drivers.

- Go to your "Drivers Code" page and click on "New Driver".
- Click on "Import".
- Paste this URL into the field: **https://github.com/GaryMilne/Hubitat-Tasmota/blob/main/Color_Bulb.groovy**
- Click "Save".

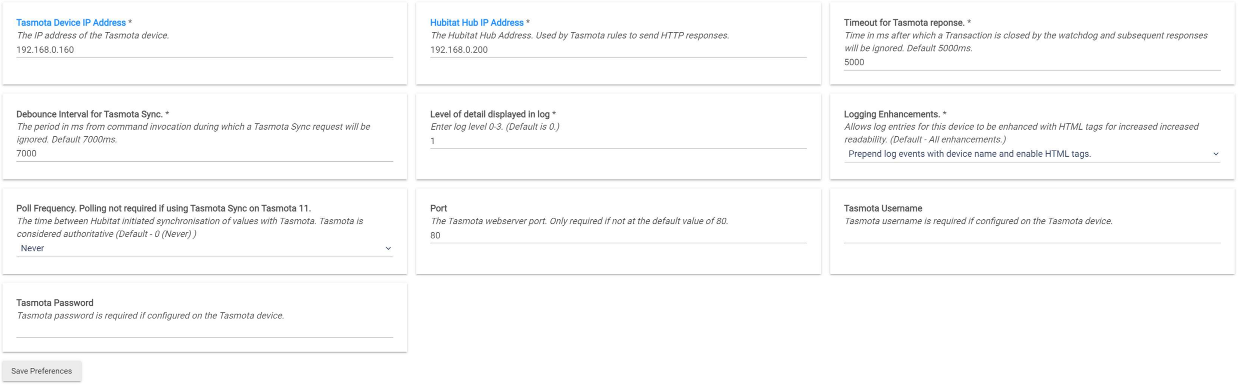
The driver should now be installed and be listed in your drivers list as "Tasmota Sync – Color Bulb".

Device Installation

Now you are ready to add the device to Hubitat. For this step you will need to know the IP addresses of the Tasmota Device and the Hubitat Hub.

- Go to your "Devices" page.
- Click on "Add Device"
- Click on "Virtual"
- Enter a device name and a device label
- For Type*, select Tasmota Sync – Color Bulb which will be down near the bottom of the list in the **User** category.
- Make any other changes that you wish and click "Save Device".

You will be taken to the device properties page. Scroll down to the "Preferences" section which will looks like this.



Tasmota Device IP Address *
The IP address of the Tasmota device.
192.168.0.160

Hubitat Hub IP Address *
The Hubitat Hub Address. Used by Tasmota rules to send HTTP responses.
192.168.0.200

Timeout for Tasmota reponse. *
Time in ms after which a Transaction is closed by the watchdog and subsequent responses will be ignored. Default 5000ms.
5000

Debounce Interval for Tasmota Sync. *
The period in ms from command invocation during which a Tasmota Sync request will be ignored. Default 7000ms.
7000

Level of detail displayed in log *
Enter log level 0-3. (Default is 0.)
1

Logging Enhancements. *
Allows log entries for this device to be enhanced with HTML tags for increased increased readability. (Default - All enhancements.)
Prepend log events with device name and enable HTML tags.

Poll Frequency. Polling not required if using Tasmota Sync on Tasmota 11.
The time between Hubitat initiated synchronisation of values with Tasmota. Tasmota is considered authoritative (Default - 0 (Never))
Never

Port
The Tasmota webserver port. Only required if not at the default value of 80.
80

Tasmota Username
Tasmota username is required if configured on the Tasmota device.

Tasmota Password
Tasmota password is required if configured on the Tasmota device.

You must complete the Tasmota Device IP Address and Hubitat Hub IP Address fields. All others are optional.

Common Settings

All Tasmota Sync drivers have the following options available in settings.

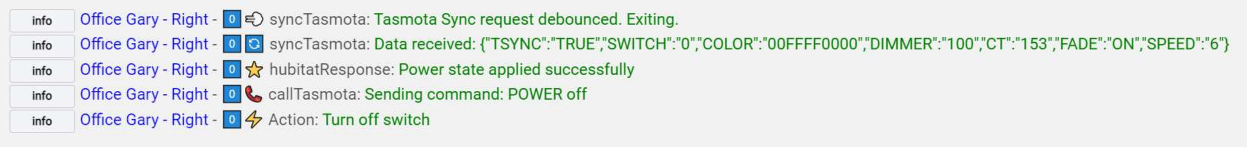
Timeout for Tasmota response: After the timeout period has ended Hubitat will ignore any late responses received from the Tasmota device.

Debounce Interval for Tasmota Sync: When Hubitat tells a Tasmota device to perform an action, such as switching on, Tasmota responds with a confirmation of the action requested and Hubitat reflects this change in its own state. However, the Tasmota Rule (if implemented) will detect the change and issue a Tasmota Sync request. When we already know the state of the Tasmota device these Tasmota Sync "echo requests" will be ignored within the specified period.

Level of detail displayed in log: Allows the user to specify the level of detail to be reported to the activity log. Logging levels correspond approximately to the following standard.

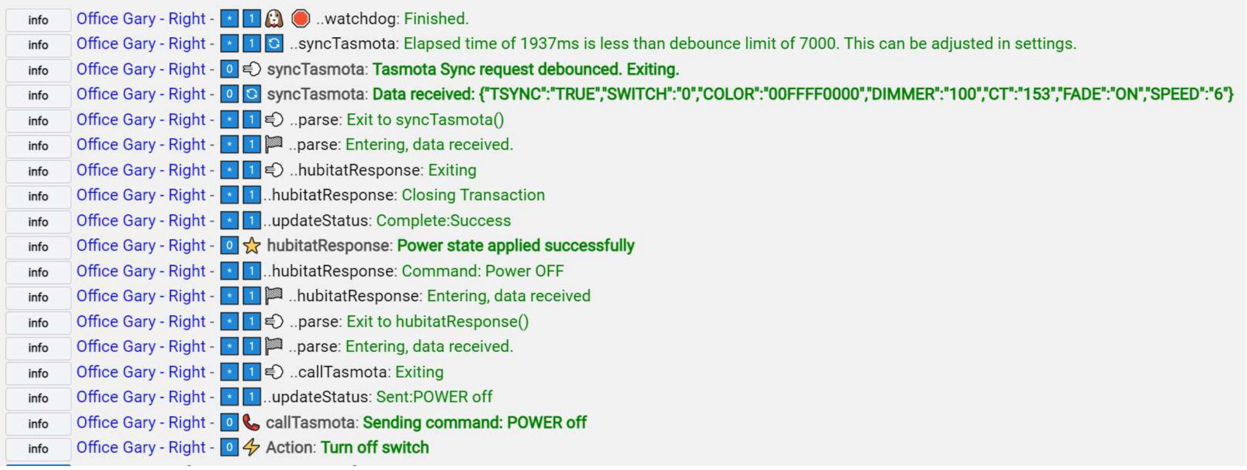
- -1 = All errors
- 0 = Actions and results
- 1 = Entering\Exiting functions
- 2 = Key variables
- 3 = Extended debugging with many variables displayed.

This is the log display when turning off a Tasmota bulb and logging level 0. In this case we have the device name turned on along with HTML enhancements.



```
info  Office Gary - Right - 0 syncTasmota: Tasmota Sync request debounced. Exiting.
info  Office Gary - Right - 0 syncTasmota: Data received: {"TSYNC":"TRUE","SWITCH":"0","COLOR":"00FFFF0000","DIMMER":"100","CT":"153","FADE":"ON","SPEED":"6"}
info  Office Gary - Right - 0 ☆ hubitatResponse: Power state applied successfully
info  Office Gary - Right - 0 ☎ callTasmota: Sending command: POWER off
info  Office Gary - Right - 0 ⚡ Action: Turn off switch
```

This is the log display when turning off a Tasmota bulb and logging level of 1. We get considerably more information. The leading numeric icons identify the log level that generated the message. The text after the icons indicates the function that generated the log entry. Finally, there is the message itself. Key actions are accompanied by an icon tile to make the flow of the log more intuitive.



```
info  Office Gary - Right - * 1 ..watchdog: Finished.
info  Office Gary - Right - * 1 ..syncTasmota: Elapsed time of 1937ms is less than debounce limit of 7000. This can be adjusted in settings.
info  Office Gary - Right - 0 syncTasmota: Tasmota Sync request debounced. Exiting.
info  Office Gary - Right - 0 syncTasmota: Data received: {"TSYNC":"TRUE","SWITCH":"0","COLOR":"00FFFF0000","DIMMER":"100","CT":"153","FADE":"ON","SPEED":"6"}
info  Office Gary - Right - * 1 ..parse: Exit to syncTasmota()
info  Office Gary - Right - * 1 ..parse: Entering, data received.
info  Office Gary - Right - * 1 ..hubitatResponse: Exiting
info  Office Gary - Right - * 1 ..hubitatResponse: Closing Transaction
info  Office Gary - Right - * 1 ..updateStatus: Complete:Success
info  Office Gary - Right - 0 ☆ hubitatResponse: Power state applied successfully
info  Office Gary - Right - * 1 ..hubitatResponse: Command: Power OFF
info  Office Gary - Right - * 1 ..hubitatResponse: Entering, data received
info  Office Gary - Right - * 1 ..parse: Exit to hubitatResponse()
info  Office Gary - Right - * 1 ..parse: Entering, data received.
info  Office Gary - Right - * 1 ..callTasmota: Exiting
info  Office Gary - Right - * 1 ..updateStatus: Sent:POWER off
info  Office Gary - Right - 0 ☎ callTasmota: Sending command: POWER off
info  Office Gary - Right - 0 ⚡ Action: Turn off switch
```

Note: If you use Hub mesh configured for a device the HTML enhancements will only display correctly on the devices’ home Hub. Log events on the other Hub will display the actual HTML tags. If this is troublesome you can disable the use of HTML tags in the driver settings as described next.

Logging Enhancements: This allows for the color coding of logged events for better readability. The options are:

- 0 - No Enhancements
- 1 - Prepend log events with device name.
- 2 - Enable HTML tags on logged events for this device.
- 3 - Prepend log events with device name and enable HTML tags.

I think these are quite intuitive given the above screenshots of the enhanced logging.

Poll Frequency: Polling is disabled by default as it is not necessary if you are using Tasmota 11. It is included for compatibility with older Tasmota versions. Polling will synchronize the Tasmota state with the Hubitat state using Tasmota as authoritative. Note: I have a couple of generic plugs that did not tolerate Tasmota 11.1 and would reboot frequently. I downgraded these and used polling as an alternative until there is a more compatible Tasmota version.

Port: Defaults to port 80. You can optionally specify an alternate port.

Tasmota Username: Defaults to blank, otherwise specify the username “Admin”.

Tasmota Password: Defaults to blank, otherwise specify the password you are using.

Click on “**Save Preferences**”

Testing the device

Do a quick test of the device by issuing some basic commands from Hubitat, on, off, etc. If the device is not responding review the log and if necessary, increase the logging level to 2 or 3 to troubleshoot the problem. The main issues to verify are the IP address of the device and Hub and ensuring that another Hubitat device is not already configured for that IP address.

Tasmota Sync Installation - Bulb

Once the device is responding to commands, we can configure Tasmota Sync. In summary, this process will install a rule in RULE3 on Tasmota. If you already have a rule in RULE3 it will be overwritten so, move it to another rule slot before proceeding. Also, Tasmota Sync uses %var% variables in the range from %var8% to %var16% so be aware those will also be overwritten.

Open the web page for your Tasmota Device and go to the Console.

On the Hubitat driver page for your newly installed device, click on “**Tasmota Inject Rule**”. You will see activity on your Tasmota console indicating that Rule3 has been installed and turned on. You can check the status of the rule by typing “rule3” at the console. It should show {“Rule3”:{“State”:”ON”, …….

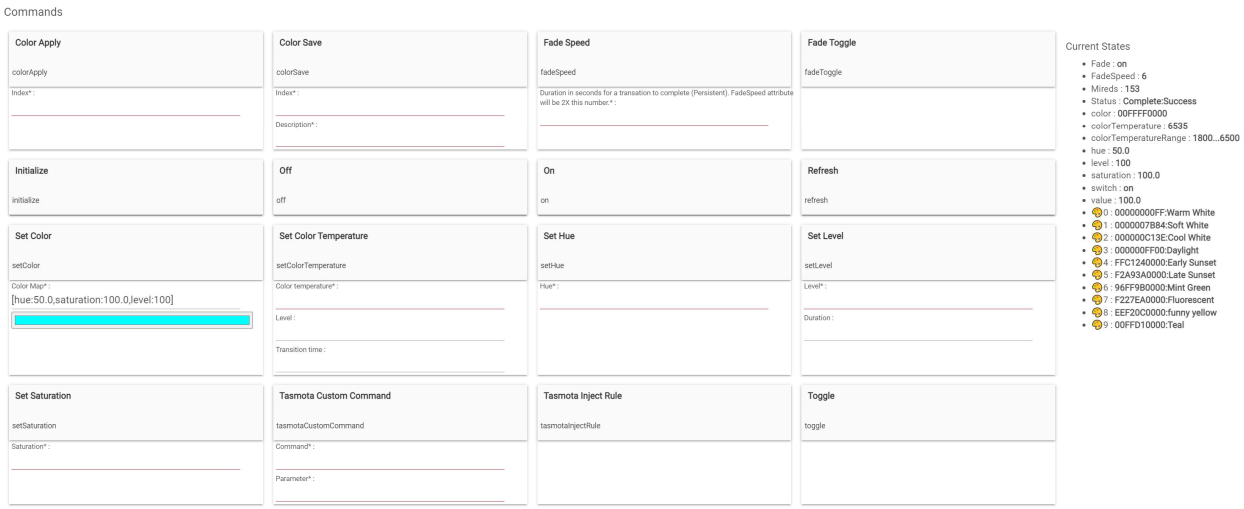
With Rule3 installed Tasmota is now ready to send monitored changes to Hubitat. For this device the monitored values are: Power, Dimmer, CT, Color, Fade, Speed (fadeSpeed).

Type “color 1” at the Tasmota console. The bulb should turn red and you will see activity on the Tasmota console. If you look at the attribute for “color” on Hubitat you will see it is FF00000000, which corresponds to red.

Type “dimmer 50” at the Tasmota console. The bulb will dim and if you check on Hubitat the value of dimmer will also be at 50. **Important:** You may also notice that when you changed the dimmer to 50 the value of the color also changed. This is not intuitive. The value of “FF” which corresponded to red has become “80”? Why, because “80” is 50% of “FF” in hexadecimal and that is how the RGB model operates.

Bulb Driver Capabilities

As you will probably have noticed the driver has a few unique capabilities.



Color Map* : [hue:50.0,saturation:100.0,level:100]

Current States

- Fade : **on**
- FadeSpeed : **6**
- Mireds : **153**
- Status : **Complete:Success**
- color : **00FFFF0000**
- colorTemperature : **6535**
- colorTemperatureRange : **1800...6500**
- hue : **50.0**
- level : **100**
- saturation : **100.0**
- switch : **on**
- value : **100.0**
- 🎨0 : **00000000FF:Warm White**
- 🎨1 : **0000007B84:Soft White**
- 🎨2 : **000000C13E:Cool White**
- 🎨3 : **000000FF00:Daylight**
- 🎨4 : **FFC1240000:Early Sunset**
- 🎨5 : **F2A93A0000:Late Sunset**
- 🎨6 : **96FF9B0000:Mint Green**
- 🎨7 : **F227EA0000:Fluorescent**
- 🎨8 : **EEF20C0000:funny yellow**
- 🎨9 : **00FFD10000:Teal**

Color Specific Commands

Color Presets

The first two functions, “Color Apply” and “Color Save” allow you to quickly save and recall color and CT favorites

To save a color use the “Set Color” or “Set Color Temperature” to first select the color that you wish to save. Once the bulb is the correct color, enter an index and description and click on “Color Save”. Check on the color list to the right and you should see your new color represented.

To apply this color, simply enter the index number and click “Color Apply”. This capability can be accessed in Hubitat Rules using a “Custom Action” and specifying the color index as a parameter.

Note: If the dimmer is at 100% the bulb driver uses Tasmota command “COLOR”. If the dimmer is < 100 then the driver uses Tasmota command ”COLOR2”, which uses a dimmer adjusted value for the color. In this case the color attribute will not match the value in the preset color because it is adjusted for the reduced dimmer level.

Fading

Tasmota has three parameters relating to the Fading of light bulbs.

1) Fade can either be on or off. To change the “Fade” state click on “**Fade Toggle**”.
2) Fade Speed (SPEED in Tasmota) is the amount of time in seconds that it takes an operation to transition. If Fadespeed is set to 10 then it will take the bulb 10 seconds to fully transition from On to Off or from red to blue etc. Fadespeed has no effect unless Fade is on. The value of Fadespeed is persistent. Note: Tasmota uses 0.5 sec increments so the FadeSpeed attribute will be 2X the number seconds entered.

3) Tasmota also has a “SPEED2” command which is a non-persistent version of SPEED intended for a single operation. Any Hubitat commands that include an option to specify “Transition Time” or “Duration” will use SPEED2 vs SPEED.

Color Functions

The functions for “Set Color”, “Set Color Temperature”, “Set Hue”, “Set Level” and “Set Saturation” are typical in color bulbs and need no explanation.

Tasmota Sync - Common Commands

These commands are common to all of the Tasmota Sync drivers.

Toggle

Simply toggles the power state of the device.

Refresh

This command issues a “STATE” request to Tasmota and processes the returned message to synchronize the Hubitat UI to the Tasmota device. This function is called when using polling and is often called when performing a “BACKLOG” function, as this does not return the results of the commands executed.

Tasmota Custom Command

This command allows the user to send simple commands to the Tasmota device. To test this out try typing “power” for the command and “off” for the parameter. This is just a convenience and not intended to replace the Tasmota console.

Note: If you do a browser refresh you will find the Tasmota response among the state variables. It will either be “thisMessage” or “lastMessage” depending on whether a Tasmota Sync request was generated by the command.

Tasmota Inject Rule

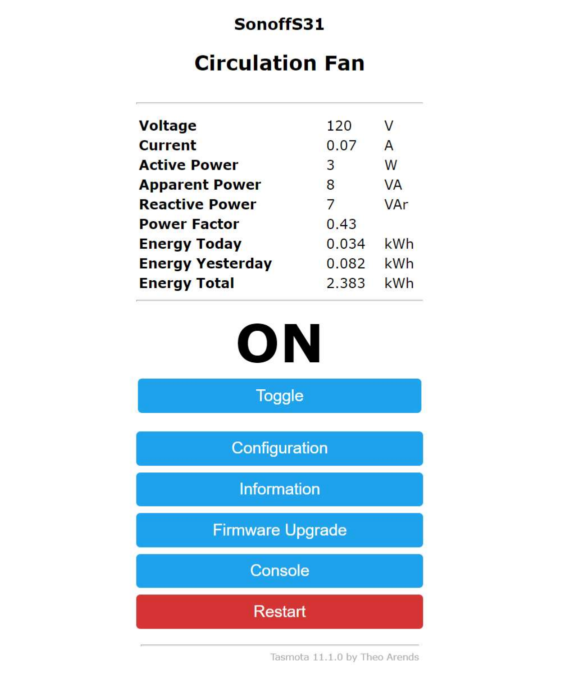
This simply injects the appropriate rule into the RULE3 position on the Tasmota device and enables it. Anything already in RULE3 will be overwritten. You can perform this action multiple times. Once you have installed RULE3 Tasmota Sync requests will be send to Hubitat whenever a monitored parameter changes.

To disable Tasmota Sync you can type “RULE3 OFF” at the Tasmota console or use the Tasmota Custom Command. To erase the rule, type ‘RULE3 “” ‘.

Tasmota Sync Installation – Switch\Plug (Power Monitoring optional)

Make sure you have performed the basic steps before continuing. This driver works the same for either switch or plug devices.

- You know the IP address of your switch\plug (PM optional) and the screen looks something like one of these.

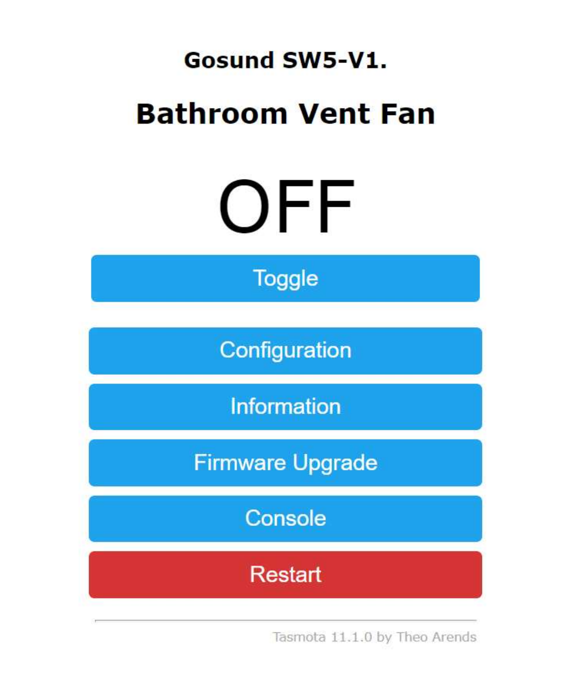


SonoffS31

Circulation Fan

Voltage	120	V
Current	0.07	A
Active Power	3	W
Apparent Power	8	VA
Reactive Power	7	VAr
Power Factor	0.43	
Energy Today	0.034	kWh
Energy Yesterday	0.082	kWh
Energy Total	2.383	kWh

ON



Gosund SW5-V1.

Bathroom Vent Fan

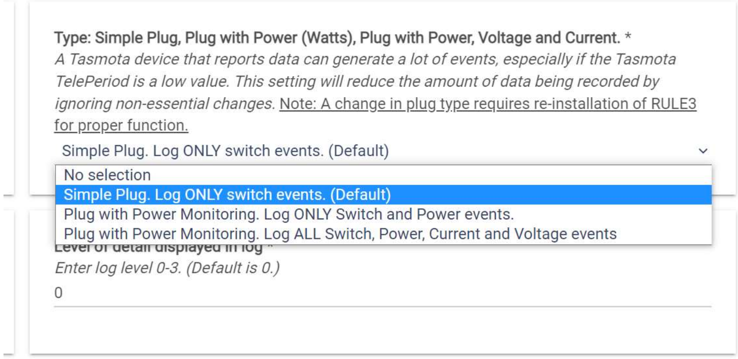
OFF

- You have installed Tasmota 11.0 or greater.
- You know the IP address of your hub.

Install the switch\plug driver using the steps previously described in the installation section of this document. The URL for the current version of the driver is:

https://github.com/GaryMilne/Hubitat-Tasmota/blob/main/Switch_Plug.groovy

Create the new device per the prior instructions and enter the IP address of the Tasmota device and the Hubitat Hub. In addition, you need to decide how this plug will operate. If it is a simple switch\plug just leave it at the default setting for “Type”. This driver



Type: Simple Plug, Plug with Power (Watts), Plug with Power, Voltage and Current. *
A Tasmota device that reports data can generate a lot of events, especially if the Tasmota TelePeriod is a low value. This setting will reduce the amount of data being recorded by ignoring non-essential changes. Note: A change in plug type requires re-installation of RULE3 for proper function.

Simple Plug. Log ONLY switch events. (Default)

No selection
Simple Plug. Log ONLY switch events. (Default)
Plug with Power Monitoring. Log ONLY Switch and Power events.
Plug with Power Monitoring. Log ALL Switch, Power, Current and Voltage events

Level of detail displayed in log *
Enter log level 0-3. (Default is 0.)
0

If the device does support Power Monitoring capabilities, you should select one of the two Power Monitoring options. You can select to monitor only switch & power (watts) or switch, power (watts), current and voltage.

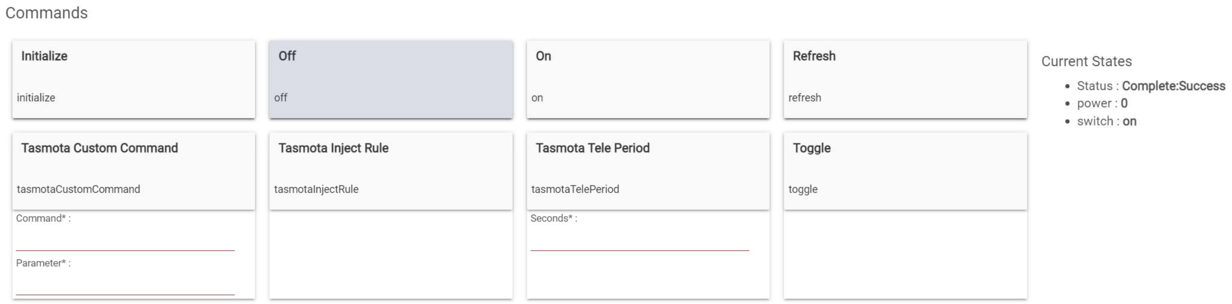
What’s the difference? If you only care about watts consumed because you are using it as an event mechanism such as for a washer or dryer, then monitoring only watts will greatly cut down on the number of events transmitted and recorded for the device.

If you do care about current and voltage, then obviously select the last option.

Note: You can change between these options but once the current and voltage attributes have been populated, they will persist forever and be displayed as “—” if you switch to only monitoring switch and power (watts). A plug that has fluctuating input voltage, which is quite common, will generate a lot of events that are not particularly useful.

Everything else is optional, see prior description for details on other settings. Remember to click on “**Save Preferences**”

Once you have done that the device is created, and the screen looks something like this.



Current States

- Status : **Complete:Success**
- power : **0**
- switch : **on**

Plug Command Descriptions

Off: Turns off the switch. The attached device will lose power.

On: Turns on the switch. The attached device will receive power.

Toggle: Turns power on\off to the attached device.

Tasmota Tele Period: This sets the Tasmota “TelePeriod” in seconds. More about this later.

For Initialize, Refresh, Tasmota Custom Command and Tasmota Inject Rule see prior descriptions.

Test the Plug

From the Hubitat Driver screen you should be able to turn the switch on\off. If this is not working turn up the logging detail and check the logs. Note: If a load is connected to the plug you should see power statistics.

Tasmota Sync (Plug\Switch)

On the Hubitat driver page for your newly installed device click on “**Tasmota Inject Rule**”. With Rule3 installed Tasmota is now ready to send monitored changes to Hubitat. Monitored changes for this device are “switch”, “power”, “current” and “voltage” depending on your installation choices. Make some changes to Tasmota and you will see the changes reflected in Hubitat within a second or two.

TelePeriod (Telemetry Period)

This is the frequency at which changes in the monitored values of power, current and voltage will be sent to Hubitat. A Teleperiod of 60 means that changes will be sent at 60 second intervals, but ONLY if the underlying value changes. This greatly reduces excess processing of unnecessary data. This makes the use of a lower Teleperiod more acceptable when monitoring appliances. Teleperiod does not affect switch events which are always transmitted immediately.

Relevant Tasmota Commands for Power Monitoring Plugs

You can modify the resolution of the Volt, Amp and Power information. Default values are as follows on a Sonoff S31.

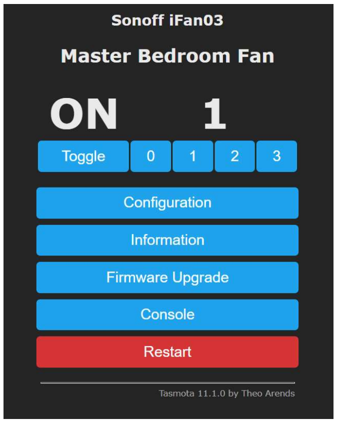
- Wattres = 0
- Voltres = 0
- Ampres = 3

For example, the resolution of 0 for Voltres simply means that it takes a full volt swing (i.e. 119 to 120) for Tasmota to register that as a change and report it. Ampres = 3 means that it is sensitive to changes of 1000th of an amp (~ 1/10th of a Watt at 120V) which is too sensitive IMO. At Ampres = 2 it is sensitive to 100th of an amp (~ 1 Watt at 120V) changes which is more reasonable.

Tasmota Sync Installation – Fan

Make sure you have performed the basic steps before continuing.

- You know the IP address of your Tasmota Fan controller and the screen looks something like this.



Sonoff iFan03

Master Bedroom Fan

ON 1

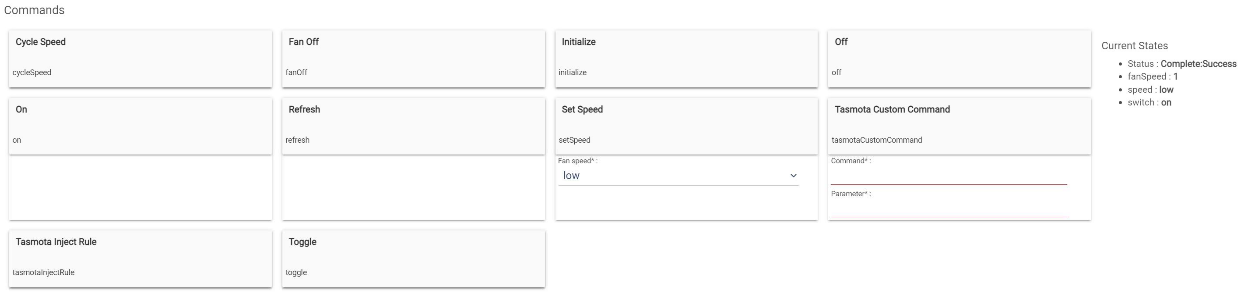
- You have installed Tasmota 11.0 or greater.
- You know the IP address of your hub.

Install the fan driver using the steps previously described in the installation section of this document. The URL for the current version of the driver is:

https://github.com/GaryMilne/Hubitat-Tasmota/blob/main/Fan.groovy

Create the new device per the prior instructions and enter the IP address of the Tasmota device and the Hubitat Hub. Everything else is optional, see prior description for details on other settings. Remember to click on “Save Preferences”

Once you have done that the device is created, and the screen looks something like this.



Fan speed* : low

Current States

- Status : Complete:Success
- fanSpeed : 1
- speed : low
- switch : on

Fan Command Descriptions

Cycle Speed: The fan will cycle to the next speed in the sequence Off – Low – Med – High - Off

Fan Off: Turns the fan off regardless of current speed.

Off: Turns off the switch which is usually associated with a light fixture. Does not affect the fan.

On: Turns on the switch which is usually associated with a light fixture. Does not affect the fan.

Set Speed: Sets the Fan to a specific speed. This is a generic Hubitat control that lists 5 possible speeds. These are converted to one of Tasmota's 3 unique speeds.

Toggle: Turns power on\off to the light fixture.

For Initialize, Refresh, Tasmota Custom Command and Tasmota Inject Rule see prior descriptions.

Test the Fan

From the Hubitat Driver screen you should be able to turn the light on\off and select different fan speeds. If this is not working turn up the logging detail and check the logs.

Tasmota Sync (Fan)

On the Hubitat driver page for your newly installed device click on "**Tasmota Inject Rule**". With Rule3 installed Tasmota is now ready to send monitored changes to Hubitat. Monitored changes for this device are "power" and "fanspeed". Make some changes to Tasmota and you will see the changes reflected in Hubitat within a second or two.

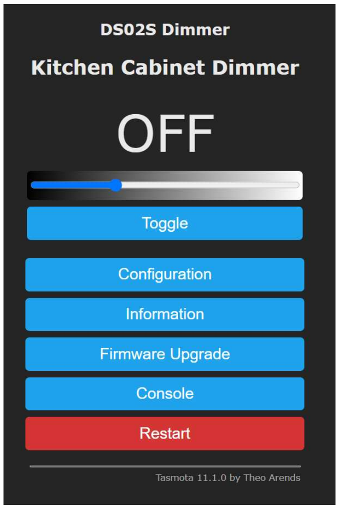
Relevant Tasmota Commands for Fans

To disable the beep on fan speed change for Sonoff iFan type "SetOption67 off" at the Tasmota console.

Tasmota Sync Installation – Dimmer Switch

Make sure you have performed the basic steps before continuing.

- You know the IP address of your Tasmota dimmer switch and the screen looks something like this.



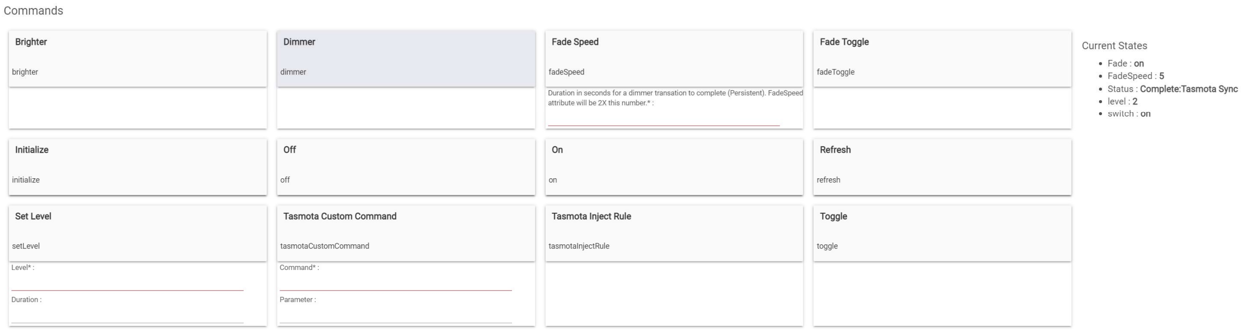
- You have installed Tasmota 11.0 or greater.
- You know the IP address of your hub.

Install the dimmer switch driver using the steps previously described in the installation section of this document. The URL for the current version of the driver is:

https://github.com/GaryMilne/Hubitat-Tasmota/blob/main/Dimmer_Switch.groovy

Create the new device per the prior instructions and enter the IP address of the Tasmota device and the Hubitat Hub. Everything else is optional, see prior description for details on other settings. Remember to click on “Save Preferences”

Once you have done that the device is created, and the screen looks something like this.



Duration in seconds for a dimmer transation to complete (Persistent). FadeSpeed attribute will be 2X this number.* :

Level* :

Duration :

Command* :

Parameter :

Current States

- Fade : on
- FadeSpeed : 5
- Status : Complete:Tasmota Sync
- level : 2
- switch : on

Dimmer Switch Command Descriptions

Brighter: Increments the brightness of the dimmer. See dimmerStep below.

Dimmer: Decrements the brightness of the dimmer. See dimmerStep below.

Off: Turns off the switch and removes power from the attached device.

On: Turns on the switch and applies power to the attached device.

Set Level: Sets the dimmer to a specific level in the range from 0 (off) to 100.

Toggle: Turns switch into the opposite state.

Fade_Toggle: Turns the Tasmota fade function on or off.

Fade Speed: The amount of time in seconds that it takes a dimming operation to complete. If Fadespeed is set to 10 then it will take the dimmer 10 seconds to fully transition from 30% to 100% brightness. Fadespeed has no effect unless Fade is on. The value of Fadespeed is persistent. Note: Tasmota uses 0.5 sec increments, so the FadeSpeed attribute will be 2X the number of seconds entered.

For Initialize, Refresh, Tasmota Custom Command and Tasmota Inject Rule see prior descriptions.

Test the Dimmer Switch

From the Hubitat Driver screen you should be able to turn the switch on\off and select different dimmer levels speeds. If this is not working turn up the logging detail and check the logs.

Tasmota Sync (Dimmer Switch)

On the Hubitat driver page for your newly installed device click on “**Tasmota Inject Rule**”. With Rule3 installed Tasmota is now ready to send monitored changes to Hubitat. Monitored changes for a fan device are “power”, “dimmer”, “fade” and “speed”. Make some changes to Tasmota and you will see the changes reflected in Hubitat within a second or two.

Relevant Tasmota Commands for Dimmer Switches

You can set the internal dimmer range if your device does not dim smoothly at either end of the dimmer range.

- dimmerRange min,max (0-1000). Example: dimmerRange 50,800

Note: You will still reference dimmer levels in the range of 0 – 100 with these applied.

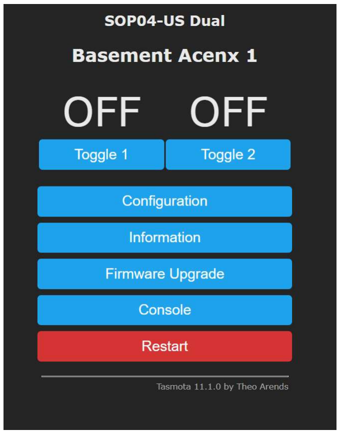
You can change the dimmer step increment with:

- dimmerStep value (5). Example: dimmerStep 5 will mean that the dimmer will increase\decrease 5% with each button press.

Tasmota Sync Installation – Dual Switch \Dual Plug

Make sure you have performed the basic steps before continuing.

- You know the IP address of your Tasmota dual switch\dual plug and the screen looks something like this.



SOP04-US Dual

Basement Acenx 1

OFF OFF

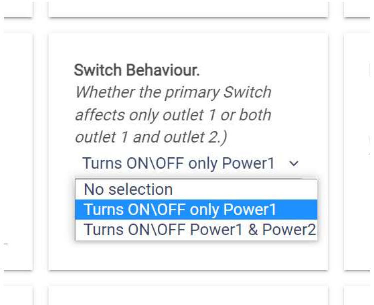
- You have installed Tasmota 11.0 or greater.
- You know the IP address of your hub.

Install the dual switch\dual plug driver using the steps previously described in the installation section of this document. The URL for the current version of the driver is:

https://github.com/GaryMilne/Hubitat-Tasmota/blob/main/Dual_Switch_Plug.groovy

Create the new device per the prior instructions and enter the IP address of the Tasmota device and the Hubitat Hub. There is one additional setting that needs to be covered.

In Hubitat, a driver can only be one instance of a switch, dimmer, bulb etc. Where a device has two or more instances of a capability, only the first one is truly identifiable by Hubitat. This setting controls whether a Hubitat switch() command will only turn off outlet 1 or turn off both outlets 1 and 2. You can change this preference at any time.



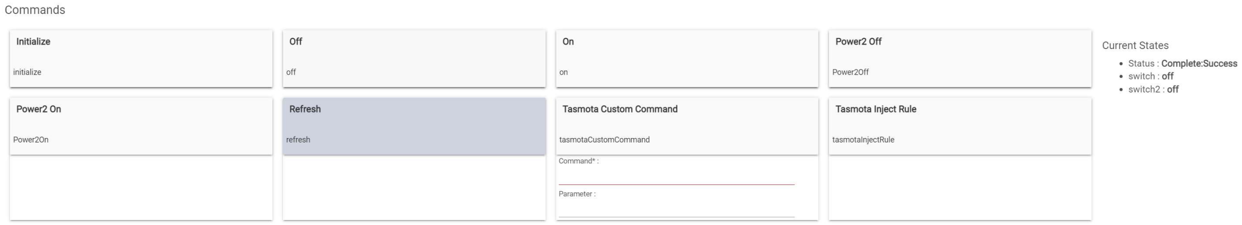
Switch Behaviour.

Whether the primary Switch affects only outlet 1 or both outlet 1 and outlet 2.)

Turns ON\OFF only Power1

Everything else is optional, see prior description for details on other settings. Remember to click on “**Save Preferences**”

Once you have done that the device is created and the screen looks something like this.



Current States

- Status : Complete:Success
- switch : off
- switch2 : off

Dual Switch\Dual Plug Command Descriptions

Off: Turns off the first switch and removes power from the attached device.

On: Turns on the first switch and applies power to the attached device.

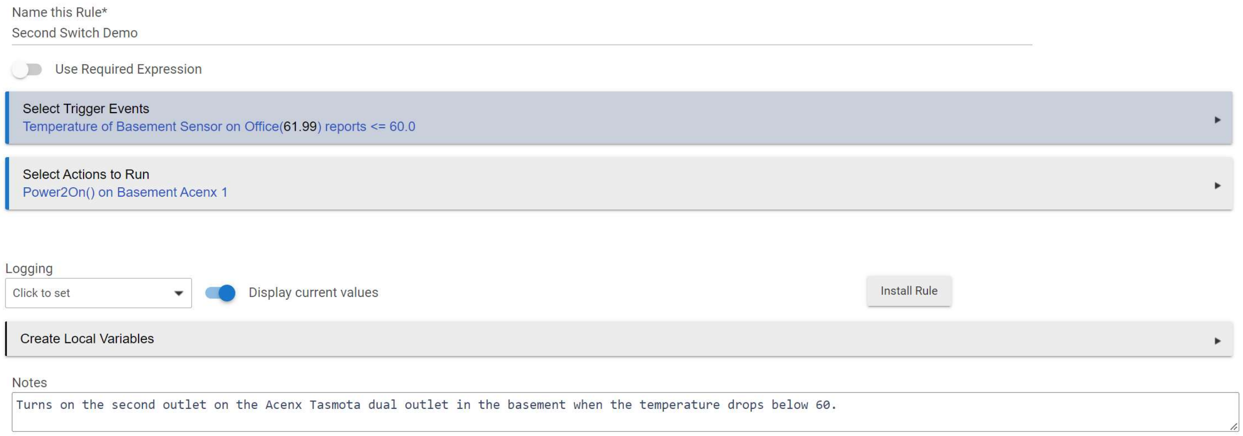
Power2Off: Turns off the second switch and removes power from the attached device.

Power2On: Turns on the second switch and applies power to the attached device.

For Initialize, Refresh, Tasmota Custom Command and Tasmota Inject Rule see prior descriptions.

Accessing Secondary Switch\Plug

It is worth repeating that only one switch will show up in a Hubitat capability list picker. To turn switch2 on\off using rules you must select a custom action command. In the screenshot below the rule turns on the second outlet of a device when the temperature drops to 60.



Name this Rule*
Second Switch Demo

Use Required Expression

Select Trigger Events
Temperature of Basement Sensor on Office(61.99) reports <= 60.0

Select Actions to Run
Power2On() on Basement Acenx 1

Logging
Click to set

Display current values

Notes
Turns on the second outlet on the Acenx Tasmota dual outlet in the basement when the temperature drops below 60.

Test the Dual Switch\Dual Plug

From the Hubitat Driver screen you should be able to turn the different switches on and off. If this is not working turn up the logging detail and check the logs.

Tasmota Sync (Dual Switch\Dual Plug)

On the Hubitat driver page for your newly installed device click on “**Tasmota Inject Rule**”. With Rule3 installed Tasmota is now ready to send monitored changes to Hubitat. Monitored changes for this device are “power1” and “power2”. Make some changes to Tasmota and you will see the changes reflected in Hubitat within a second or two.

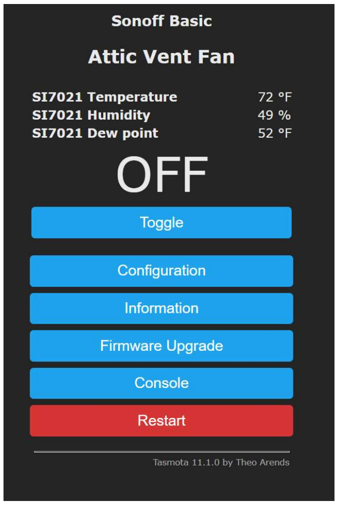
Relevant Tasmota Commands for Dual Switch\Dual Plug

None at this time.

Tasmota Sync Installation – Switch with Sensor

Make sure you have performed the basic steps before continuing.

- You know the IP address of your Tasmota switch with sensor and the screen looks something like this.



Sonoff Basic

Attic Vent Fan

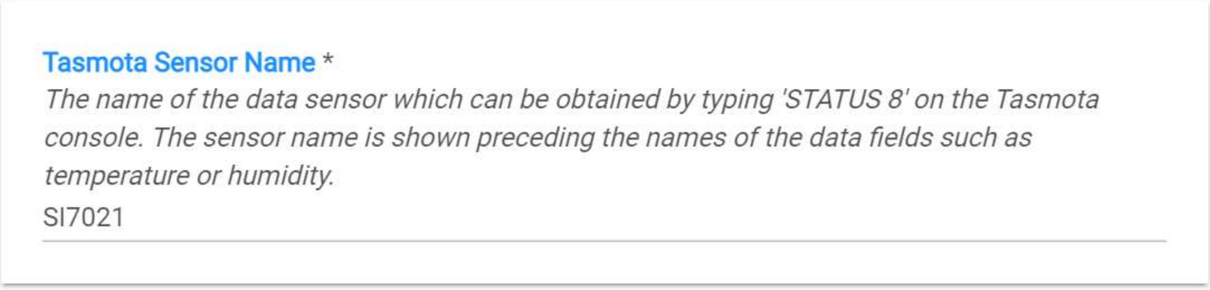
SI7021 Temperature	72 °F
SI7021 Humidity	49 %
SI7021 Dew point	52 °F

OFF

- You have installed Tasmota 11.0 or greater.
- You know the IP address of your hub.

Install the Switch with T&H Sensor driver using the steps previously described in the installation section of this document. The URL for the current version of the driver is: **https://github.com/GaryMilne/Hubitat-Tasmota/blob/main/Switch_with_Sensor.groovy**

Create the new device per the prior instructions and enter the IP address of the Tasmota device and the Hubitat Hub. There is one additional setting that needs to be covered. We must tell Hubitat the name of the sensor we wish to pull data from.



Tasmota Sensor Name *

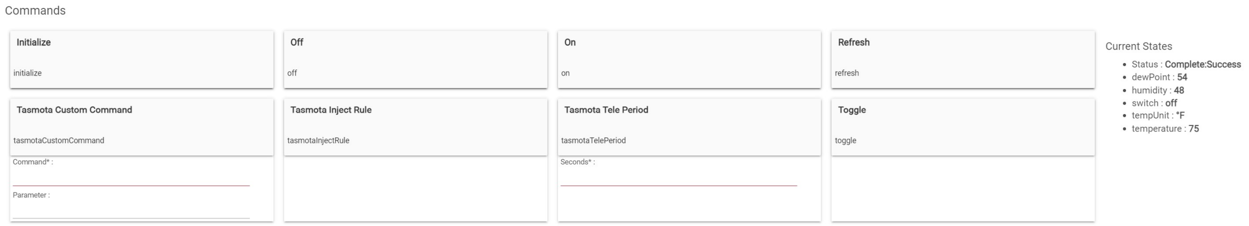
The name of the data sensor which can be obtained by typing 'STATUS 8' on the Tasmota console. The sensor name is shown preceding the names of the data fields such as temperature or humidity.

SI7021

As you can see from the main screen above my sensor name was “SI7021”, so I enter that into the field. In this case Temperature, Humidity and Dewpoint are collected. The driver is capable of handling multiple types of sensors and a variety of data types but not all are enabled in the current iteration as I do not have the hardware to guarantee their error free operation. If you are reading this and have a sensor that does not work 100% please let me know.

Everything else is optional, see prior description for details on other settings. Remember to click on “**Save Preferences**”

Once you have done that the device is created, and the screen looks something like this.



Current States

- Status : **Complete:Success**
- dewPoint : **54**
- humidity : **48**
- switch : **off**
- tempUnit : **°F**
- temperature : **75**

Switch with Sensor Command Descriptions

Off: Turns off the first switch and removes power from the attached device.

On: Turns on the first switch and applies power to the attached device.

TelePeriod: The period in seconds at which changes will be reported to Hubitat.

For Initialize, Refresh, Tasmota Custom Command and Tasmota Inject Rule see prior descriptions.

Test the Switch

From the Hubitat Driver screen you should be able to turn the switches on and off. If this is not working turn up the logging detail and check the logs.

Tasmota Sync (Switch with Sensor)

On the Hubitat driver page for your newly installed device click on “**Tasmota Inject Rule**”. With Rule3 installed Tasmota is now ready to send monitored changes to Hubitat. Monitored changes for this device are “power1” and any data type (i.e. temperature, pressure, illuminance, humidity) retrieved from the sensor. Go to the Tasmota console and check that when the Teleperiod executes, new data is being sent to Hubitat. Any changes should be reflected in Hubitat in a second of two. To make things go faster do “teleperiod 10” but remember that you still need the sensor data to change before anything will be sent to Hubitat. Remember to change teleperiod back to a reasonable value.

TelePeriod (Telemetry Period)

This is the frequency at which changes in the monitored values of sensor data will be sent to Hubitat. A Teleperiod of 60 means that changes will be sent at 60 second intervals but ONLY if the underlying value changes. This greatly reduces excess processing of unnecessary data and allows for more frequent updates. I use a Teleperiod of 300 seconds as changes to these values are not that rapid or critical in general. Changes to the state of the switch are always transmitted immediately regardless of Teleperiod.

Relevant Tasmota Commands for Switch with Sensor

You can adjust the resolution of temperature and humidity readings using the commands:

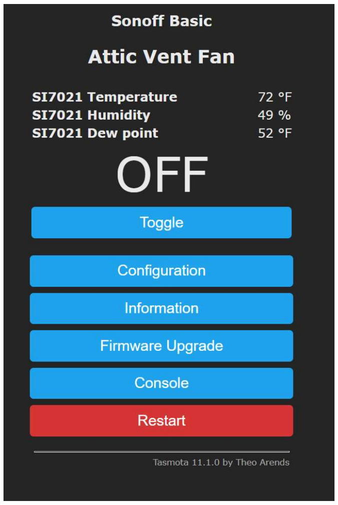
- TempRes
- HumRes

By default, these are both at 0 meaning a full 1 degree or 1% humidity change is required to trigger an event.

Tasmota Sync Installation –Sensor

Make sure you have performed the basic steps before continuing.

- You know the IP address of your Tasmota switch with sensor and the screen looks something like this.



Sonoff Basic

Attic Vent Fan

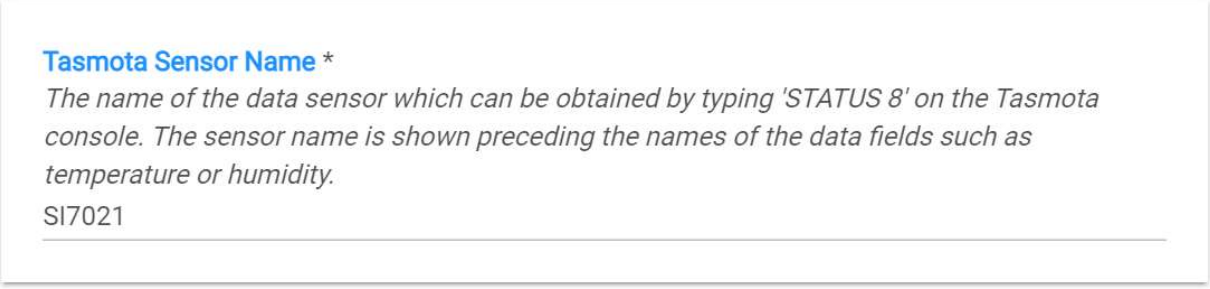
```
SI7021 Temperature    72 °F
SI7021 Humidity       49 %
SI7021 Dew point      52 °F
```

OFF

- You have installed Tasmota 11.0 or greater.
- You know the IP address of your hub.

Install the Switch with T&H Sensor driver using the steps previously described in the installation section of this document. The URL for the current version of the driver is: **https://github.com/GaryMilne/Hubitat-Tasmota/blob/main/Sensor.groovy**

Create the new device per the prior instructions and enter the IP address of the Tasmota device and the Hubitat Hub. There is one additional setting that needs to be covered. We must tell Hubitat the name of the sensor we wish to pull data from.



Tasmota Sensor Name *

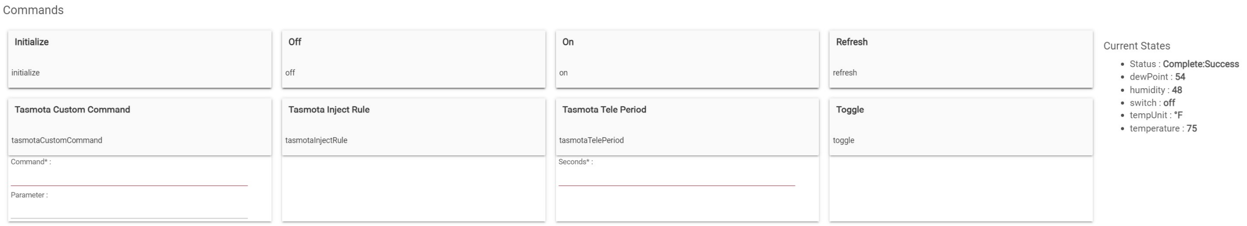
The name of the data sensor which can be obtained by typing 'STATUS 8' on the Tasmota console. The sensor name is shown preceding the names of the data fields such as temperature or humidity.

SI7021

As you can see from the main screen above my sensor name was “SI7021”, so I enter that into the field. In this case Temperature, Humidity and Dewpoint are collected. The driver is capable of handling multiple types of sensors and a variety of data types but not all are enabled in the current iteration as I do not have the hardware to guarantee their error free operation. If you are reading this and have a sensor that does not work 100% please let me know.

Everything else is optional, see prior description for details on other settings. Remember to click on “**Save Preferences**”

Once you have done that the device is created, and the screen looks something like this.



Current States

- Status : **Complete:Success**
- dewPoint : **54**
- humidity : **48**
- switch : **off**
- tempUnit : **°F**
- temperature : **75**

Sensor Command Descriptions

TelePeriod: The period in seconds at which changes will be reported to Hubitat.

For Initialize, Refresh, Tasmota Custom Command and Tasmota Inject Rule see prior descriptions.

Tasmota Sync (Sensor)

On the Hubitat driver page for your newly installed device click on “**Tasmota Inject Rule**”. With Rule3 installed Tasmota is now ready to send monitored changes to Hubitat. Monitored changes for this device are any data type (i.e. temperature, pressure, illuminance, humidity) retrieved from the sensor. Go to the Tasmota console and check that when the Teleperiod executes, new data is being sent to Hubitat. Any changes should be reflected in Hubitat in a second of two. To make things go faster do “teleperiod 10” but remember that you still need the sensor data to change before anything will be sent to Hubitat. Remember to change teleperiod back to a reasonable value.

TelePeriod (Telemetry Period)

This is the frequency at which changes in the monitored values of sensor data will be sent to Hubitat. A Teleperiod of 60 means that changes will be sent at 60 second intervals but ONLY if an underlying value changes. This greatly reduces excess processing of unnecessary data and allows for more frequent updates. I use a Teleperiod of 300 seconds as changes to these values are not that rapid or critical in general.

Relevant Tasmota Commands for Sensor

You can adjust the resolution and unit designation of sensor data using the commands:

- TempRes
- TempUnit
- HumRes
- PressRes
- PressUnit

By default, these are at 0 meaning a full 1 degree or 1% humidity change is required to trigger an event.