

Tasmota Sync Drivers

A GUIDE FOR INSTALLATION AND USE

Gary Milne

Contents

TLDR	2
Intro.....	2
Architecture	3
Drivers	3
Tasmota.....	3
Installing a New Device.....	4
Driver Installation	4
Device Installation.....	4
Common Settings.....	5
Testing the device	7
Tasmota Sync Installation - Bulb.....	7
Bulb Driver Capabilities.....	7
Color Specific Commands	8
Color Presets	8
Fading.....	8
Color Functions	8
Tasmota Sync Common Commands	9
Toggle.....	9
Refresh	9
Tasmota Custom Command.....	9
Tasmota Inject Rule	9
Tasmota Sync Installation – Fan	10
Fan Command Descriptions.....	10
Test the Fan.....	11
Tasmota Sync (Fan).....	11

Tasmota Sync – Device Drivers for Hubitat

Automatic Synchronization between Tasmota and Hubitat without custom Tasmota

TLDR

These drivers allow native Tasmota (11 onwards) to synchronize with Hubitat. 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 8.1. 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 switch. 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 5 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. Switch \ Switch with PM
3. Dimmer Switch
4. Dual Switch
5. Switch with Sensor

The Tasmota Sync family drivers are located here: <https://github.com/GaryMilne/Hubitat-Tasmota> Initially I am only releasing the bulb driver to get the code tested on a broader range of devices. 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. You can use Tasmota Sync drivers with prior versions of Tasmota but this would only operate in polling mode which needs to be enabled.

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

Once you have updated a device to Tasmota 11 you are ready to install the Tasmota Sync drivers into your Hubitat environment.

Installing a New Device

This section gives a detailed example of you to setup a new device. In this example we are using a bulb but the same principles are true for all of the 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://raw.githubusercontent.com/GaryMilne/Hubitat-Tasmota/main/Tasmota%20Sync%20Bulb%20-%20Current%20Version.txt>
- 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 look 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 response. * 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 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.		
Save Preferences		

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 it's 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" can 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 it uses "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, 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.

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 %var10% to %var16% so beware 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 a bulb 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.

Bulb Driver Capabilities

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

Commands

Color Apply colorApply index* : 	Color Save colorSave index* : Description* : 	Fade Speed fadeSpeed Duration in seconds for a transition to complete (Persistent). Fadespeed attribute will be 2X this number* : 	Fade Toggle fadeToggle 	Current States <ul style="list-style-type: none"> Fade : on FadeSpeed : 6 Mins : 153 Status : Complete:Success color : 00FFFF0000 colorTemperature : 6695 colorTemperatureRange : 1800...6500 hue : 50.0 level : 100 saturation : 100.0 switch : on value : 100.0 0 : 00000000FF:Warm White 1 : 0000007884:Soft White 2 : 000000C13E:Cool White 3 : 000000FF00:Daylight 4 : FFC1240000:Early Sunset 5 : F2A93A0000:Late Sunset 6 : 96FF980000:Mint Green 7 : F227EA0000:Fluorescent 8 : EEF20C0000:funny yellow 9 : 00FFD10000:Teal
Initialize initialize 	Off off 	On on 	Refresh refresh 	
Set Color setColor Color Map* : [hue:50.0,saturation:100.0,level:100] 	Set Color Temperature setColorTemperature Color temperature* : Level : Transition time : 	Set Hue setHue Hue* : 	Set Level setLevel Level* : Duration : 	
Set Saturation setSaturation Saturation* : 	Tasmota Custom Command tasmotaCustomCommand Command* : Parameter* : 	Tasmota Inject Rule tasmotaInjectRule 	Toggle toggle 	

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.

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 sometimes 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

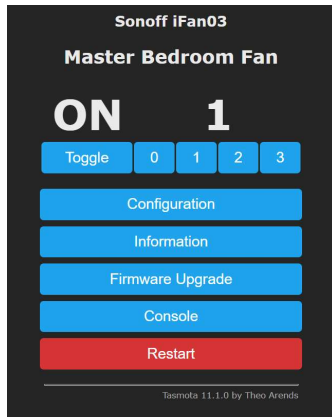
This simply injects the appropriate rule into the RULE3 position on the Tasmota device and enables it. Anything already in RULE3 will be overwritten. You can perform this action multiple times. Once you have installed RULE3 Tasmota Sync requests will be sent to Hubitat whenever a monitored parameter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



- 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 fan driver is:

<https://raw.githubusercontent.com/GaryMilne/Hubitat-Tasmota/main/Tasmota%20Sync%20Fan%20-%20Current%20Version.txt>

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.

Commands

Cycle Speed cycleSpeed	Fan Off fanOff	Initialize initialize	Off off
On on	Refresh refresh	Set Speed setSpeed Fan speed*: low	Tasmota Custom Command tasmotaCustomCommand Command*: Parameter*:
Tasmota Inject Rule tasmotainjectrule	Toggle toggle		

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.

Initialize: Used in the creation of the Fan Driver and sets initial settings.

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 a fan device are "power" and "fanspeed". Make some changes to Tasmota and you will see the changes reflected in Hubitat within a second or two.

I think this documentation should be sufficient to get most people up and running. I'll update this further when I release my other Tasmota Sync drivers.