

SmartThings Device Handler Notes

Tasmota Plug with Power Monitoring ST Classic V0.91

This is the quick guide for the SmartThings device handler: "Tasmota Plug with Power Monitoring for ST Classic V0.91"

Tasmota firmware has a rich set of capabilities that are not being fully exploited by the currently available device handlers (DH) for Tasmota based plugs with Power Management. This DH makes a lot of the relevant power data available through the SmartThings Classic interface. It supports the following features:

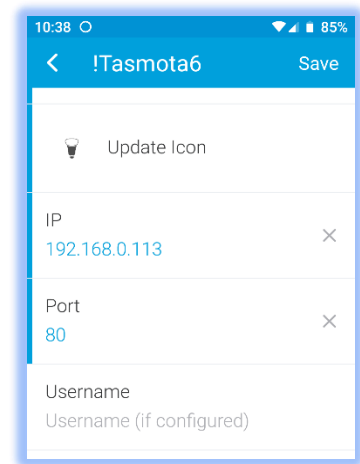
Capabilities
Plug <ul style="list-style-type: none">• Off\On
Monitor Device <ul style="list-style-type: none">• Off\On\Standby – based on power threshold
Power Stats <ul style="list-style-type: none">• Watts, Volts, Amps• Today, Yesterday Totals• Total Since, Total Cost• Apparent Power, Reactive Power, Power Factor• Reset Power Total
Device <ul style="list-style-type: none">• Last On• Last Off• Last Standby• State Duration (attribute)
Plug <ul style="list-style-type: none">• Time• Timezone – Set local or UTC• Uptime
WiFi Strength
Poll\Sync
Update DNI
WiFi Strength

*The Device Handler will sync from the device with the specified IP address.

Device Setup

After installation of the DH and the creation of a device you must configure the IP address of the device. All other values have reasonable defaults and can be adjusted later if desired.

After saving the configuration, the Device Network Interface (DNI) will be updated automatically and can be seen at the very bottom of the main screen. It should be in the form, DNI:XXXXXXXX:0050 where the X's are hex values.



The screenshot shows a mobile application interface for configuring a Tasmota device. The title bar is blue with a back arrow, the text '!Tasmota6', and a 'Save' button. The status bar at the top shows the time 10:38 and battery level 85%. The main content area has a white background with a light blue border. It contains an 'Update Icon' button with a lightbulb icon. Below this are three input fields: 'IP' with the value '192.168.0.113', 'Port' with the value '80', and 'Username' with the placeholder text 'Username (if configured)'. Each input field has a small 'x' icon to its right for clearing the text.

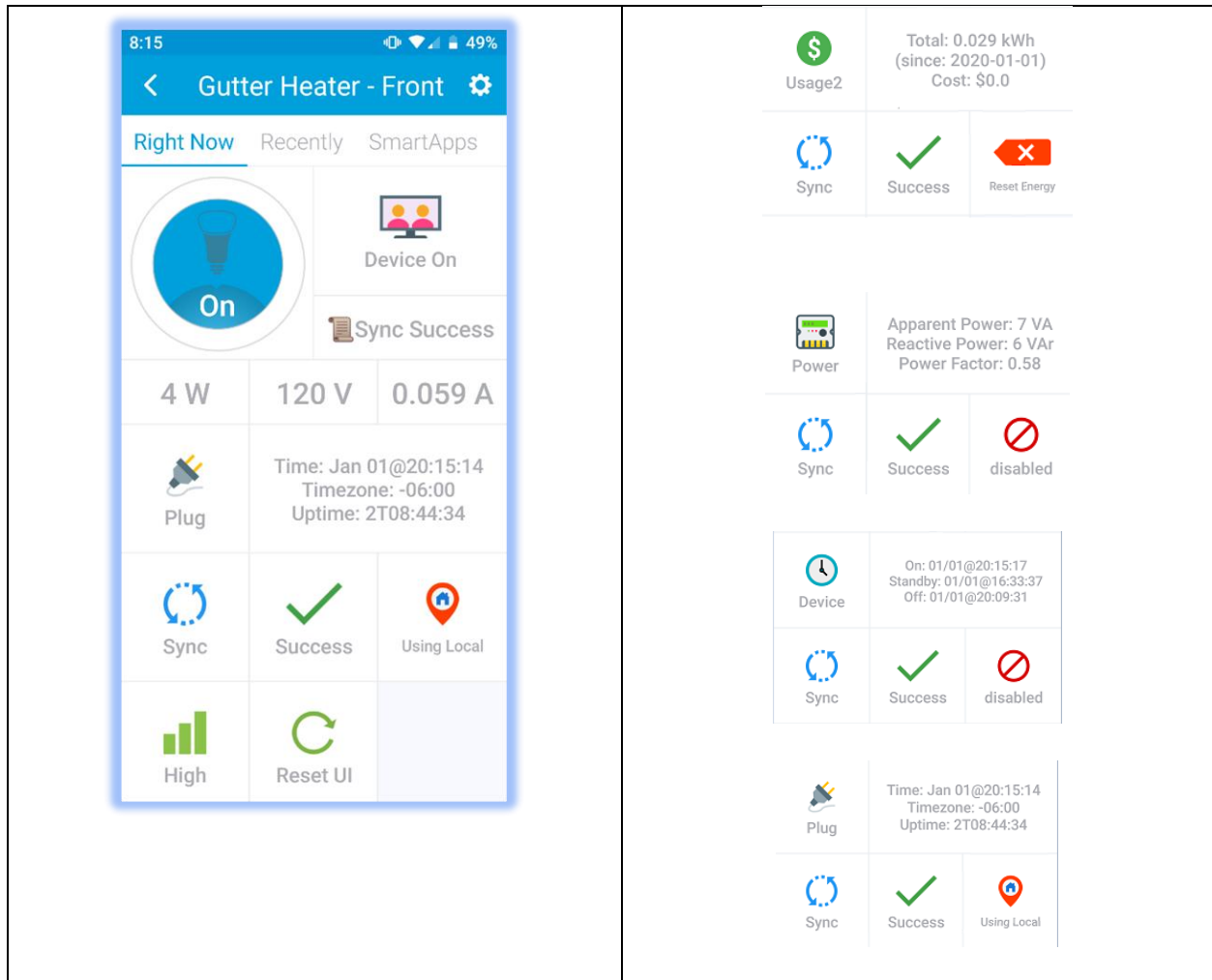


DNI:C0A80071:0050

If this does not update correctly then click on the DNI button to force the DNI to be re-calculated and applied.

Once the DNI is set properly you can now sync the plug to the UI using the Sync tile. If you see a "System Busy" message it typically means that the background polling process is already running. If it stays this way for more than 15 seconds click the Reset UI to clear it.

Quick Guide



Most of the controls are self-explanatory or will become obvious with a little playing but here is the quick guide.

On\Off Button: Self-explanatory.

Device On\Off\Standby: Indicates the state of the downstream device based upon power consumption.

- Zero Watts = Off
- Between zero and Standby Power Threshold* watts = Standby
- Greater than Standby Power Threshold = On

*Standby Power Threshold is configurable in Settings.

Message: Displays system messages.

Current Power Conditions: Watts, Volts and Amps are view only and will lag other operations as they need a refresh to indicate a change.

Usage1, Usage2, Power, Device, Plug: This button cycles through these 5 options. Information related to the presently selected state will be displayed on the tile to its right.

In the Usage2 state (total power since last reset) a tile is enabled called Reset Energy whose purpose is obvious.

In the Plug state a tile is enabled which allows you to toggle the plug between local time (as defined in settings i.e. -06:00 or +01:00) and UTC time (00:00). The main reason for doing this is to have the energy statistics information correct for your timezone.

Sync: Forces a sync between the UI and the Plug. The UI settings will be changed to match those of the plug. This tile will also show as active when the background poll process is active (every 1 min by default).

Status (Success): This is a status tile that shows the status of a request. The status will be one of the following; idle, send, wait, receive, success or fail. A request is not complete until the status reads either Fail or Success.

WiFi: Clicking this tile will request the WiFi signal strength from the plug and display the appropriate icon for High, Medium or Low. A device operating with a "Low" Wifi status is more likely to have communication issues.

Reset UI: In the event of the UI becoming out of sync or a system busy message lasting more than 15 seconds this tile performs a reset and terminates the current command.

DNI:XXXXXXXX:0050 This is the Device Network Address that is comprised of the IP address in hex format and the HTTP port in hex format (0050 is port 80). This should be calculated and applied when the IP address is assigned and preferences saved. If the DNI tile shows the name you originally assigned it when creating the device, then there is something wrong. Pressing the DNI tile will cause the DNI to be recalculated and applied if it is incorrect. The only time this process is problematic is if you are creating and deleting multiple devices in short order that use the same IP address. Check the log and check that no other device is already assigned that IP address.

Icon Credits: This tile does nothing other than acknowledge that all tile icons used in this DH are courtesy of Icons8 (www.icons8.com).

Backgrounder

SmartThings HTTP calls are asynchronous which makes for a challenging paradigm. It is not uncommon for an HTTP request to fail or timeout somewhere within the SmartThings platform, in which case the device handler never receives a response to process. A tricky problem because the programmer cannot distinguish between a request time out (typically device off) and a response time out (typically network issue). Two very different scenarios, leaving the device in different states, with no obvious distinction.

In this DH I have turned these stateless calls into a quasi-synchronous transacted model by waiting for a response to an HTTP call and correlating the returned state to the requested state.

Because of this synchronous mode we wait (in Reliable mode) until we get a response or a response times out before we permanently commit the changes to the UI. Because of this transacted model it is only possible to execute one command at a time. The DH will block any additional commands until the current command has cleared. This will be indicated on screen as shown below with a System Busy message. When the current command is finished the status will clear.

Logging: You can easily tweak the amount of logging that goes on by changing the log threshold. The normal level of logging is zero, increasing it to 1 or 2 will increase the amount of logging.

```
//Function to selectively log activity based on varios logging levels. Nor
//Loglevels are cumulative: -1 All errors, 0 = All user actions plus statu
private log(name, message, loglevel){

    //This is a quick way to filter out messages based on loglevel
    def threshold = 0
    if (loglevel > threshold) {return}

    //This is a quick way to filter out messages from a given function
    def filterlist = "listofnamestofilter: function1, anothername"
    if (filterlist.contains(name) == true) {return}
```

Troubleshooting

This software is near completion but for now I consider it to be a late Beta. I wanted to make it available before I head out of town for a few months with limited access. Can't promise any support in the interim period but I will respond to eMails.

If you have some Groovy knowledge the first thing to do is to look at the SmartThings Logging for the device for any exception errors. You can temporarily turn up the logging as described earlier. Any null values would be of special interest.

You can also go to the Tasmota console on the device that should be receiving the commands and see if they have been appeared and been executed without error.

Written by: Gary Milne on January 1st, 2019.