# Home Automation with Tasmota

## General

There are some general commands and features that Tasmota provides to make working with the devices easier.

## Groups

Using groups allows you the capability of sending the same command to multiple devices. For instance it is possible to turn on or off all of the Tasmota enabled devices by sending an MQTT command. Here are some examples:

turn on all devices

cmnd/sonoffs/power1 1

get current state of all devices (an invalid payload causes all devices to return their current state)

cmnd/sonoffs/power1 check

change the ssid and password on all devices

Backlog ssid1 <myssid>; password1 <mypassword>

## Garage Controller

### Introduction

In order to use a Sonoff type device as a garage door controller you must be able to provide a dry contact and have that dry contact close for approximately 1 second. This will instruct the garage door opener to open or close the garage door.

### Hardware

A regular sonoff basic can be modified to provide a dry contact by cutting the traces on the PCB that provides power to the contacts of the relay. This is tricky but possible. The better solution is to purchase a Sonoff SV (safe voltage).

The sonoff basic in the garage has a DHT22 connected to provide a temperature and humidy. It should be noted that the DHT22 and the AM2301 are equivalent.

### Software

The software is the regular version of Tasmota (sonoff.bin).

### Configuration

The sonoff basic is setup as follows:

|  |  |  |
| --- | --- | --- |
| Channel | Description | Setting |
| GPIO1 | Serial Out | Switch2 (10) |
| GPIO2 | N/A | None (00) |
| GPIO3 | Serial In | Switch3 (11) |
| GPIO4 | N/A | None (00) |
| GPIO14 | Sensor | AM2301 (02) |

# PulseTime

The pulse time command is used to ensure the relay is only on for a specified duration. In the case of the garage door opener this time is 1 second. The pulse time value is set in units of 0.1 seconds. To set the pulse time enter the following at the console:

PulseTime 10

# Door Status

The open and closed statuses are connected to the GPIO1 and GPIO3 inputs. These signals are dry contacts that are closed when the sensor is near. This means that when the door is “closed” the closed signal will show Off and the open signal will show On. The opposite is true when the door is “open”.

# MQTT

The data is transmitted via MQTT. The topic is set to “garage”. Under the Logging configuration the telemetry period must be set to 10. This ensures that the MQTT messages are transmitted at a faster rate (every 10 seconds).

## Bathroom Fan Control

### Introduction

The bathroom fan control uses a wifi enabled smart switch running the Tasmota firmware. The switch replaces the standard switch that controls power to the fan. The smart switch is connected to an AM2301 temperature/humidity sensor. Currently the logic for controlling the fan is run by NodeRed although this could be reconfigured to use rules running within Tasmota to remove the need for NodeRed.

Using Tasmota Rules feature it would be possible to move all logic for control into the smart switch. This would allow control even if communication with the MQTT broker was down.

### Hardware

The wifi switch has a DHT22 connected to provide a temperature and humidy. It should be noted that the DHT22 and the AM2301 are equivalent.

### Software

The software is the regular version of Tasmota (sonoff.bin).

### Configuration

The sonoff basic is setup as follows:

|  |  |  |
| --- | --- | --- |
| Channel | Description | Setting |
| GPIO1 | Serial Out | AM2301 (02) |
| GPIO2 | N/A | None (00) |
| GPIO3 | Serial In | None (00) |
| GPIO4 | N/A | None (00) |
| GPIO14 | Sensor | None (00) |

# PulseTime

The pulse time command is used to ensure the relay is only on for a specified duration. In the case of the garage door opener this time is 1 second. The pulse time value is set in units of 0.1 seconds. To set the pulse time enter the following at the console:

PulseTime 10

# MQTT

The data is transmitted via MQTT. The topic is set to “dn\_bath”. Under the Logging configuration the telemetry period must be set to 60. This ensures that the MQTT messages are transmitted at a faster rate (every 60 seconds).

# Local Control Logic

If performing the control within the smart switch was desired a Rule could be setup to provide this functionality.

This rule will turn the fan on when the humidity is greater than 90% and turn the fan off when the humidity is less than 85%. The problem with this method is that there is no way to turn the fan on manually and if it is very humid in the room naturally the fan may turn on when the bathroom isn’t in use.

rule1 on AM2301#Humidity>90 do Power1 1 endon on AM2301#Humidity<85 do Power1 0 endon

rule1 1

A way to fix this is to change the rule so that the fan turns on automatically when the humidity is high but then sets a timer to turn off after a time has expired.

rule1 on AM2301#Humidity>98 do Power1 1 endon

rule2 on switch1#state=1 do backlog power1 on; ruletimer1 600 endon on rules#timer=1 do power1 off endon

rule1 1

rule2 1

## Three Way Switch Control

### Introduction

The three way switch control uses two wifi enabled smart switches running the Tasmota firmware. One switch replaces the standard switch that controls the light while the other switch is independent and can be installed anywhere. The Tasmota Rules feature is used to provide all of the logic for control and is integrated into the smart switch. This allows control even if communication with the MQTT broker was down.

### Hardware

A standard wifi switch is used without any additional sensors.

### Software

The software is the regular version of Tasmota (sonoff.bin).

### Configuration

The sonoff basic is setup as follows:

|  |  |  |
| --- | --- | --- |
| Channel | Description | Setting |
| GPIO1 | Serial Out | None (00) |
| GPIO2 | N/A | None (00) |
| GPIO3 | Serial In | None (00) |
| GPIO4 | N/A | None (00) |
| GPIO14 | Sensor | None (00) |

# MQTT

The data is transmitted via MQTT. The topic is set according to the location of the switch. Under the Logging configuration the telemetry period must be set to 60. This ensures that the MQTT messages are transmitted at a faster rate (every 60 seconds).

# Local Control Logic

In order for the two switches to communicate each switch must be setup independently.

Master Switch (the one connected to the light)

The rule is used to set the state of the secondary switch to match the primary switch. Each time the primary switch state changes the message is published. This ensures the two state of the two switches remain in sync.

rule1 on power1#state do publish cmnd/<secondary topic>POWER %value% endon

rule1 1

Slave Switch (the one not connected to the light)

The rule is used to command the primary switch to toggle it’s state. If the primary switch is on it will turn off and vice versa.

rule1 on button1#state do publish cmnd/<primary topic>/POWER TOGGLE endon

rule1 1