

Documentation

MagicHome replacement firmware

Table of contents:

1. Initial configuration
 1. Hotspot and Login Information
 2. API Key
 3. Configure your WiFi
 4. Validate
2. Behavior and features
 1. Reset the configuration
 2. Modes, settings and default values
 3. Connection loss
3. HTTP API
 1. Available methods and usage
 2. Example
4. Telnet API
 1. Available methods and usage
 2. Example

1. Initial configuration

After powering on the device for the first time, if no valid configuration was found or the last configuration was deleted it will automatically go into config mode.

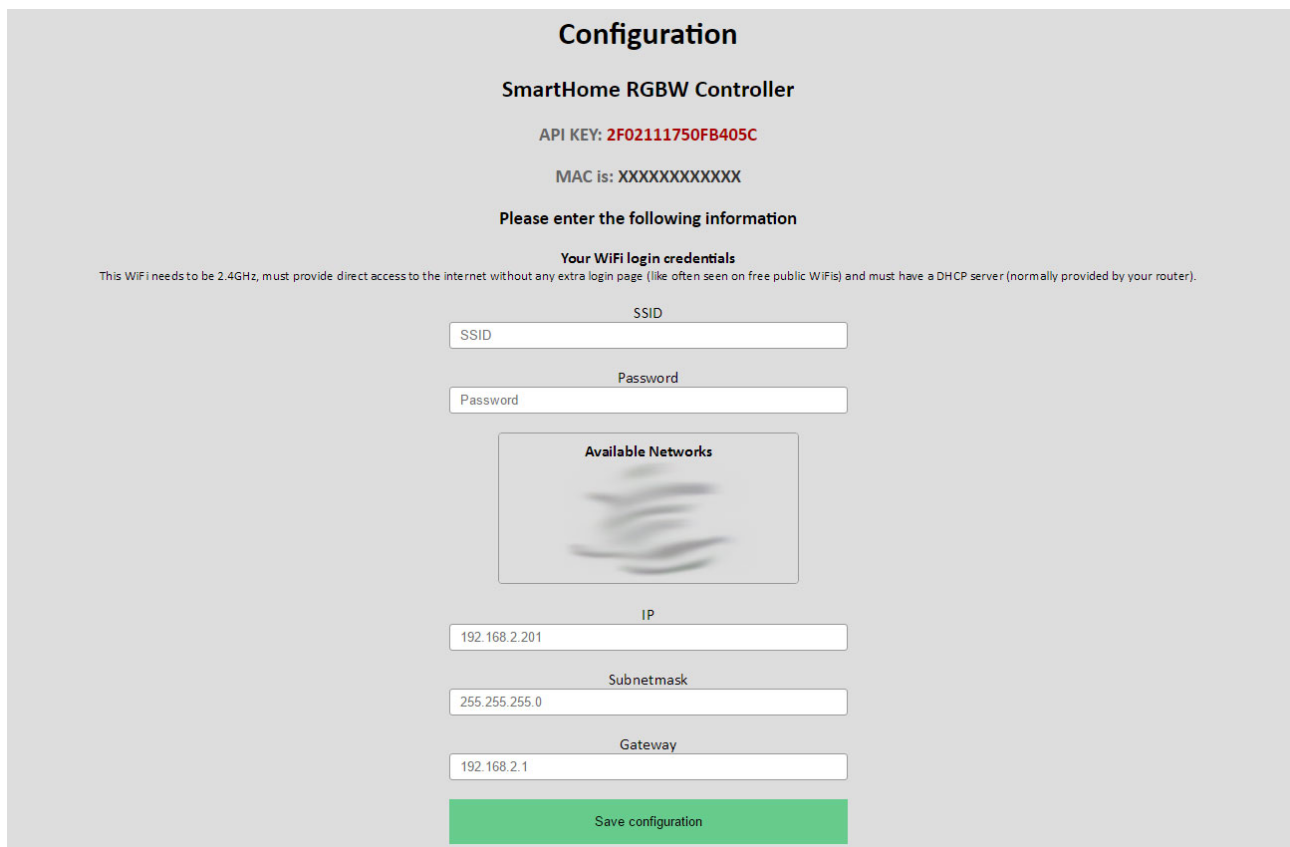
1.1 Hotspot and Login Information

After config mode has started the device will open a WiFi Hotspot named “SmartHome RGBW” followed by the MAC address of the controller.

You can connect to the WiFi using the password “espconfig”. Make sure to use DHCP.

Once logged in (with pc, laptop or smartphone) you may turn off mobile network and other network adapters to ensure no problems with failover functionality of the device you are using to connect.

Now you can open your browser and navigate to <http://10.10.10.10/> to get to the configuration page.



The screenshot shows the configuration interface for the SmartHome RGBW Controller. At the top, it displays the title "Configuration" and "SmartHome RGBW Controller". Below this, it shows the API KEY: 2F02111750FB405C and the MAC address: XXXXXXXXXXXX. A prompt asks the user to enter the following information: "Your WiFi login credentials". A note specifies that the WiFi must be 2.4GHz, provide direct internet access, and have a DHCP server. The form includes input fields for SSID, Password, IP (192.168.2.201), Subnetmask (255.255.255.0), and Gateway (192.168.2.1). There is also a section for "Available Networks" which is currently blank. A green "Save configuration" button is at the bottom.

Configuration

SmartHome RGBW Controller

API KEY: 2F02111750FB405C

MAC is: XXXXXXXXXXXX

Please enter the following information

Your WiFi login credentials

This WiFi needs to be 2.4GHz, must provide direct access to the internet without any extra login page (like often seen on free public WiFi) and must have a DHCP server (normally provided by your router).

SSID

Password

Available Networks

IP

192.168.2.201

Subnetmask

255.255.255.0

Gateway

192.168.2.1

Save configuration

1.2 API Key

You will see your controller specific API Key at the top of the page.

MAKE SURE TO SAVE IT, YOU NEED IT TO USE THE API!

1.3 Configure your WiFi

As you can see in the screenshot above, there are some fields you need to fill out.

- Your SSID
- Your Password
- A static IP address which matches your Network (i.e. 192.168.0.200)
- The Gateway (Usually your Router) (i.e. 192.168.0.1)
- Your subnet mask (i.e. 255.255.255.0)

After you filled everything, make sure to double check you entries. Then click on the “Save” button below the form.

The Controller will now try to connect to your WiFi. If there is a problem the controller will open the configuration WiFi again. In that case start over from point 1.1.

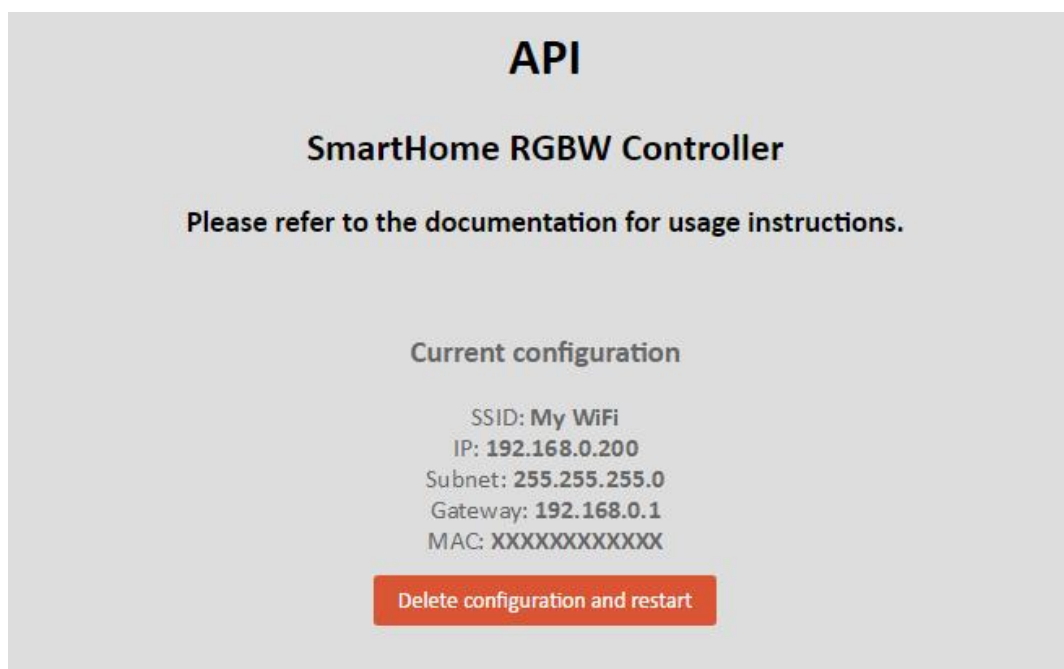
If everything goes fine and the configuration WiFi does not show up again you can connect to your normal WiFi again.

Make sure to revert any changes to interfaces and mobile networks on your phone if you made any.

1.4 Validate

If you have successfully connected to your WiFi again you can validate that the controller works by opening your browser and navigating to the IP you set during the configuration process earlier. (i.e. 192.168.0.200)

If you see a page like the one in the picture below you successfully connected the controller and you are ready to use it!



2. Behavior and features

2.1 Reset the configuration

As you saw in the last picture there is a reset button on the http index page of the controller. If you click it, the configuration will be deleted and the configuration WiFi will be opened again.

However if you can't access the website due to IP miss-configuration you can always turn off your WiFi AP/Router and reboot the controller. Once it recognizes that the WiFi which is saved is not reachable it will automatically start configuration mode again. You can edit your settings like described in step 1.1.

2.2 Modes, settings and default values

The Firmware has 3 basic modes

- Off (mode=0) [default]
- On (mode=1)
- RGB Fader (mode=2)

You can set the following parameters at any time:

- Red (r=0 ... r=255) [default=0]
- Green (g=0 ... g=255) [default=0]
- Blue (b=0 ... b=255) [default=0]
- White (w=0 ... w=255) [default=0]
- Time (time=0 ... time=10000) [default=2000] (in milliseconds)
- Frequency (freq=100 ... freq=25000) [default=100] (in Hz, HTTP API only, auto-saved)

You can read and set these values (except reading the frequency, you should not change it too often anyway)

There is also a way to save all of these values to the EEPROM so they will be restored after reboot.

Optionally you can turn on autosave mode which will automatically do it for you (this may lead to flicker since it's software PWM and the EEPROM access time leads to lags in the PWM generator)

More to that in section 3.

2.3 Connection loss

If your WiFi router has a failure and the WiFi network is not available the controller continuously tries to reconnect to the network which is saved on it until you reboot the controller. (then it would start configuration mode due to unavailability of the saved network **after 10 minutes.**)

However. If you do not restart the controller it will automatically reconnect to you WiFi once it's available again.

3. HTTP API

3.1 Available methods and usage

GET method:

This gets the current state of the controller in JSON format.

You can call it via the following URL: "http://<ip>/get".

There is no API Key required.

The result will look like this:

```
{
  r: 0,
  g: 127,
  b: 0,
  w: 0,
  mode: 2,
  time: 1000,
  mac: "XXXXXXXXXXXX",
  autosave: false
}
```

SET method:

This sets the variables described in 2.2.

You can call it via the following URL: "http://<ip>/set".

API Key is required. (key parameter, see example in 3.2)

The result will look like this:

```
{
  success: true
}
```

Extras: You can add the parameter "save" with the value "1" to immediately save your changes to the EEPROM (see example in 3.2)

You can change the autosave behavior with the parameter "autosave", set it to "1" or "0"

3.2 Example

http://192.168.0.200/set?key=2F0211175F5B405C&mode=2&time=5000&save=1

Sets mode to RGB Fade, time to 5 seconds and saves the changes to the EEPROM.

http://192.168.0.200/set?key=2F0211175F5B405C&mode=1&r=255&g=0&b=0&w=0

Sets mode to ON, r to full brightness and g & b off.

(There will be a transition time of 5s due to the changes in the last call)

http://192.168.0.200/set?key=2F0211175F5B405C&r=500

Will return an error message because r is outside of the valid range.

http://192.168.0.200/set?key=2F0211175F5B405C&autosave=1

Turns autosave mode on and automatically saves the EEPROM.

http://192.168.0.200/set?key=2F0211175F5B405C&freq=150

Changes the PWM frequency to 150Hz.

4. Telnet API

The Telnet API runs on port 23 and every command has to be ended with \n (new line).

4.1 Available methods and usage

To authenticate simply send the API Key. [session related]

Methods WITHOUT authentication:

“ping” – (returns “pong”)

“enable response” – (enables response “ok” on led change, save and autosave [session related])

“disable response” – (disables same response)

“get” – (returns current status in this format: [r|g|b|w|mode|time|mac|autosave], look at 4.2 for example)

“exit” – (closes session, you should always do this!)

Methods WITH authentication:

“save” – (saves the EEPROM)

“enable autosave” – (Enables autosave)

“disable autosave” – (Disables autosave)

To SET something use the following commands:

r=value , g=value , b=value , w=value , m=value (mode) , t=value (time)

these can be concatenated by the pipe symbol “|”

i.e. r=100|b=200|g=0|m=1 (see examples in 4.2 for more info)

4.2 Example

(white: received from controller. blue: sent to controller)

welcome message	welcome
request ping	ping
answer	pong
request status	get
recieved status	0 0 0 0 0 2000 5CCF7FC67CE5 0
enable response	enable response
confirmation	ok
invalid API Key	asdfg
answer	invalid apikey
valid API Key	2F021007505B405C
answer	authorized
change settings	m=1 r=255 g=0
confirmation	ok
change settings	m=0
confirmation	ok
request status	get
recieved status	255 0 0 0 0 2000 5CCF7FC67CE5 0
change settings	m=1
confirmation	ok
request status	get
recieved status	255 0 0 0 1 2000 5CCF7FC67CE5 0
change settings	m=0
confirmation	ok

