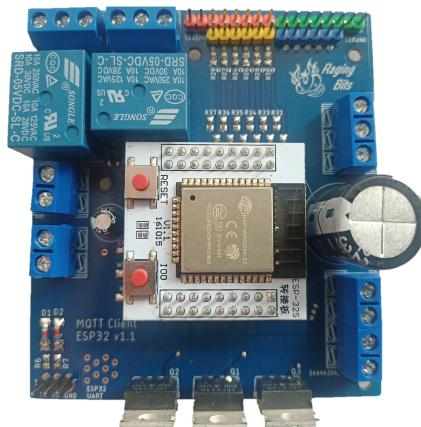


# Raging Bits MQTT

## Generic Device

v1.1



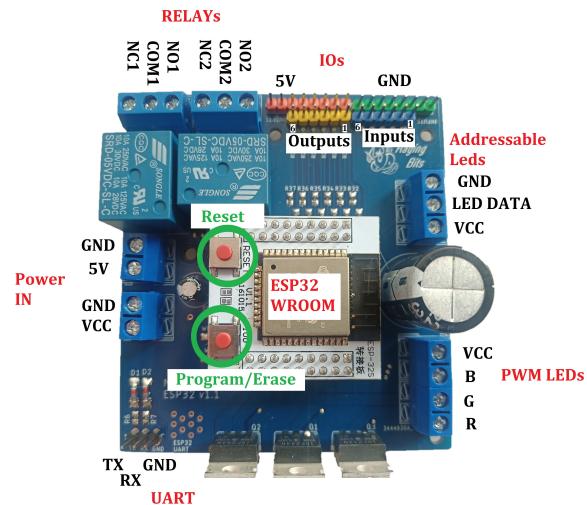
### Top level specs

- 1x UART
- 6x Inputs
- 6x Outputs
- 3x Power OpenDrain PWM outputs (RGB Up to 5A per channel)
- 1x Addressable LEDs interface (300 leds rgb leds)
- 2x Relays (Up to 25V)
- Independent power supply for power LEDs (24V max)

### ***Default Software***

- Full MQTT control for all interfaces
- WiFi Hotspot for simple setup
- Hotstart after first time setup

# Device hardware interfaces



MQTT Generic Device detail pinout

## Hardware IOs pinout from ESP to board functions.

### BOARD Function

WIFI\_LED  
 ERROR\_LED  
 WORK\_LED  
 PWM\_R  
 PWM\_G  
 PWM\_B  
 RELAY\_1  
 RELAY\_2  
 OUTPUT\_1  
 OUTPUT\_2  
 OUTPUT\_3  
 OUTPUT\_4  
 OUTPUT\_5  
 OUTPUT\_6  
 INPUT\_1  
 INPUT\_2  
 INPUT\_3  
 INPUT\_4  
 INPUT\_5  
 INPUT\_6  
 ADDRESSABLE\_LEDS\_DRIVER\_RESET  
 MEMORY\_ERASE\_REQUEST/PROGRAM  
 ESP32\_UART\_TX  
 ESP32\_UART\_RX  
 ADDRESSABLE\_LEDS\_DRIVER\_UART\_TX  
 ADDRESSABLE\_LEDS\_DRIVER\_UART\_RX

### ESP32 PIN

IO22	
IO21	
IO23	
IO4	
IO15	
IO2	
IO18	
IO19	
IO33	
IO26	
IO32	
IO25	
IO27	
IO12	
IO13	
IO14	
IO34	
IO36	
IO35	
IO39	
IO5	
IO0	
IO1	
IO3	
IO17	
IO16	

## **UART**

The Uart interface can be used for:

- ESP32 Programming
- MQTT Generic Device Debug output
- User Work

It has a maximum of 3.3v line interface level with protection.

A maximum of 0.5MBaud is advised.

When configured to user work, there will be no debug output after the main application starts running. The startup of the device up until the uart is reconfigured, will still be seen.

The default Uart debug speed is 115200 8N0

## **Power PWMs**

Each of the power PWMs output will be capable of up to 5A open drain driving to ground.

## **Power**

The system processors and logical control is powered by the 5V@0.5A input.

Consumptions:

- 80mA regular work with no relays nor hotspot active.
- 350MA with all relays and features active.

The Power PWMs and addressable LEDs bus will be powered by the VCC.

The VCC can be up to 25V.

## **Buttons**

Reset button, resets the unit.

IO0, is used for both programming the ESP32 and request the main application to erase any settings saved.

## **Inputs**

The device inputs have a 1Kohm input impedance and have a maximum supported input of 5V.

## **Outputs**

The device outputs are configured as open drain with an input protection load of 100ohm when have in RESET value, and are pulled up to 5V with 100kohm when have a SET value.

The maximum open drain RESET value current is of 15mA per channel.

**WARNING!! NEVER SHORT CIRCUIT THE OUTPUTS TO ANY VOLTAGE ABOVE ZERO!!**

This will permanently damage the outputs and most likely the device itself.

## **Addressable LEDs Bus**

The addressable LEDs bus will support mostly of the market regular brands.

SK and WS are currently guaranteed to work.

The data format is of RGB (3 components) formats ONLY. Can be set as RGB, GRB, BRG, BGR, etc... for strips of up to 300LEDs.

# MQTT Generic Device Default Software

## Device Setup

The device will have a Hotspot named

"rb\_wifi\_mqtt\_client\_xxxxxxxxxx" where **xxxxxxxxxx** is the device Unit ID.

Usually this ID is unique and pre configured from factory. It may be modified under specific conditions.

Once powered, the device will load the setup information and try to reconnect using it.

For the first time, there will be no data so the user will need to connect a wifi and browser capable device to the hotspot.

Once connected, open a browser and in the address bar set the address to  
" 1.2.3.4 " and Enter/Proceed.

A webpage will open requesting the following information:

- WiFi details, SSID and passphrase, where to connect to.
- Network Broker IP and Port.
- RGB data format.
- Functionality of the Serial port, user dedicated interface or default ESP32 debug output.
  - If the Uart is enabled for User dedication functionality, the ESP32 debug logs will no longer output as long as the main code is running, as it will redirect the port.

Once the data is filled in, press connect.

At this point the device will save the data and try to connect to the WiFi and to the MQTT broker.

Once connected the Hotspot will turn off.

## Erase device saved data

If the setup has not been correctly set or needs to be modified, press the ESP32 button labelled IO0 (Program/Erase button), until the Error LED lights up (about 5seconds), indicating the data has been erased.

Once the button is released, the unit will reset and be ready at default state.

At this point follow **Device Setup** instructions again.

## **MQTT Services**

The MQTT services provided allow to control the Relays, power PWMs, IOs outputs, addressable LEDs colour and Uart output when enabled for user work.

The Inputs and Uart input, if enabled for user work, will automatically generate MQTT messages with the updated data.

### **Relays**

The relays can be controlled using the MQTT Topic

"**rb\_wifi\_mqtt\_client\_XXXXXX relay\_set**" where **XXXXXX** is the device Unit ID.

The message supported data formats are:

"AB", "A B", "ABCD", "C D AB", "A B C D", Where A and C are Relay Numbers "1" and "2", and B and D are the Relays "1" and/or "0" states. The numbers can have or not spaces in between but the order of Relay+State MUST be respected.

For example:

Set Relay one ON: "11" or "1 1"

Set Relay one OFF: "10" or "1 0"

Set Relay one OFF and relay two ON: "1021" or "21 1 0" or "2 1 10" or "1 0 2 1"

### **Power PWMs and Addressable LEDs**

The PWMs/LEDs can be controlled using the MQTT Topic

"**rb\_wifi\_mqtt\_client\_XXXXXX rgb\_set**" where **XXXXXX** is the device Unit ID.

Both Addressable LEDs and Power PWMs will have the same data set.

The Addressable LEDs cannot be individually addressed, these will all have the same colour set by the RGB command. The data sent to the Addressable LEDs will follow the RGB format set at setup, although the Power PWMs will just be set with the command data.

The PWMs will have a 240Hz frequency capable of 256 different steps.

The message supported data formats are:

"#RRGGBB" or "r g b" where RR, GG and BB are hexadecimal format of colour component Red, Green and Blue. r, g and b are either decimal or hexadecimal format.

For example:

Set pure white: "#FFFFFF" or "255 255 255" or "255 255 0xff" or "255 0xFF 0xff"

Set pure Blue: "#0000FF" or "0 0 255" or "0 0 0xff" or "0 0x00 0xff"

### **Outputs set/reset**

The Outputs can be controlled using the MQTT Topic

"**rb\_wifi\_mqtt\_client\_XXXXXX outputs\_set**" where **XXXXXX** is the device Unit ID.

The message supported data formats are:

"AB", "A B", "ABCD", "C D AB", "A B CD GH D E" etc.

Where A , C, E, G... are Output Numbers from "1" to "6", and B, D, F, H are the Outputs states "1" and/or "0". The numbers can have or not spaces in between but the order of Output+State MUST be respected.

For example:

Set Output 2 ON: "21" or "2 1"

Set Output 2 OFF, 1 ON, 5 OFF and 4 ON: "20 1 150 4 1" or "20115041" or "2 0 1 1 5 0 4 1" etc...

## Inputs reading

The Inputs state request can be done using the MQTT Topic

"rb\_wifi\_mqtt\_client\_XXXXXX\_Inputs\_get" where **XXXXXX** is the device Unit ID.

This message has no data to be sent.

## Inputs reading reply

The Inputs state information can be received when subscribing to the MQTT Topic

"rb\_wifi\_mqtt\_client\_XXXXXX\_Inputs" where **XXXXXX** is the device Unit ID.

The message data format is:

"A B".

Where A is Input Numbers from "1" to "6" and B is the state "1" or "0".

For example:

Received information of Input 2 ON: "2 1"

Received information of Input 5 OFF: "5 0"

## Uart input

The user work Uart input data can be received when subscribing to the MQTT Topic

"rb\_wifi\_mqtt\_client\_XXXXXX\_uart\_in" where **XXXXXX** is the device Unit ID.

The message data content is the Uart input data.

## **Uart output**

The user work Uart output data can be sent through the MQTT Topic  
"**rb\_wifi\_mqtt\_client\_XXXXXX uart\_out**" where **XXXXXX** is the device Unit ID.  
The message data content is the Uart output data.

## **Reprogramming the device**

This procedure will erase the device main application, and shall be **done at user own risk**.

- 1) Power the device.
- 2) Press and hold both Reset and IO0
- 3) Release Reset
- 4) Release IO0
- 5) Use Arduino IDE, espressif tool or simple programming too to reprogram the device using ISP through the uart.

## **References**

[https://github.com/RagingBits/MQTT\\_GenDevice](https://github.com/RagingBits/MQTT_GenDevice)

<https://randomnerdtutorials.com/esp32-pinout-reference-gpios>

[https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)

<https://www.espressif.com/en/products/modules/esp32>

[https://github.com/RagingBits/ESP32\\_Wroom\\_Tools](https://github.com/RagingBits/ESP32_Wroom_Tools)

## Device pictures of interest

