
Smart Lights

Release 2

J.G.Aguado

2023-04-12

SAFETY GUIDELINES

1	Contents	3
2	Technical support	5
2.1	Safety guidelines	5
2.2	Getting started	6
2.3	Programming	8
2.4	FAQ	11
2.5	Schematics and drawings	12
2.6	Technical Specifications	13

Note: You are on the documentation site of the X1 version. Please refer to X4 for documentation regarding the other board version.

Smart Lights X1



Welcome to the documentation page of the Smart Lights X1, an electronic board based on the ESP32 with the target of controlling a dimmable lights.

It can be powered through any power transformer in the range of $9V_{DC}$ to $24V_{DC}$ and being able to regulate a monochrome LED strip.

CONTENTS

Safety guidelines

How to operate and manipulate the Smart Lights X1.

Getting started

First steps for configuring and working with the Smart Lights X1.

Schematics and drawings

Design files of the Smart Lights X1.

TECHNICAL SUPPORT

If you have technical problems or cannot find the information that you need in the provided documentation, please contact me directly:

Author

J.G.Aguado

Contact

jon-garcia@hotmail.com

Board

Smart Lights X1

Revision

2

Date

April 12, 2023

2.1 Safety guidelines

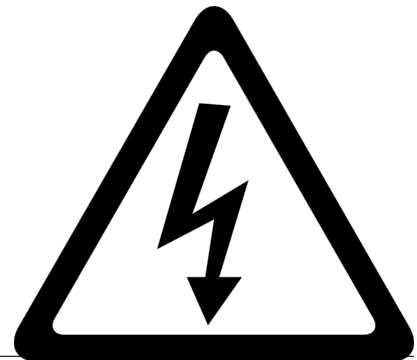
These safety guidelines, the operation instructions and the limit values listed in *Technical Specifications* are to be read carefully before operating the product.

Caution: This product has been designed for being used in domotic applications. It was not designed for any use involving serious risks or hazards that could lead to death, injury, serious physical damage, or loss of any kind without the implementation of exceptionally stringent safety precautions.

2.1.1 Risk of electric shock

The usage of this product could imply working with high voltage installations ($100 - 240V_{AC}$). Perform any action, specially in the area with this symbol on the board, with the appropriate means and always with safety.

If you have any question, please check with technical support (*Technical support*) before taking any action potentially dangerous.



2.1.2 Protection against ESD

The Smart Lights X1 contains components sensitive to ESD, which can be damaged by inappropriate handling. It's therefore necessary to provide some safety precautions against ESD during the handling and operation.

2.1.3 Usage, transport and storage

During the use, transport and storage, this product must be protected from any mechanical stress (forces applied over the board or it's components), temperatures out of the data-sheet's range, high environmental humidity or aggressive atmospheres.

Avoid using it in areas where it cannot be assured the previously mentioned environmental conditions.



2.1.4 Repairs

If any of the components on the board gets damaged, a reparation is possible. This repair can be performed by yourself if you identify the problem and have the means for desoldering & soldering. If this is not the case, just write to the *Technical support* mail to reach a solution.

2.1.5 Disposal

In accordance with national and local environmental protection and material recovery and recycling regulations, electronic devices that can no longer be used (in case a repair is not possible) must be disposed of separately and not with normal household garbage.

2.1.6 Restriction of Hazardous Substances



The PCB and the electronic components mounted on the Smart Lights X1 have been manufactured attending to the EU rules restricting the use of hazardous substances (RoHS) in electrical and electronic equipment to protect the environment and public health.

2.2 Getting started

2.2.1 Soldering

Important: Please note that some components in this board need to be soldered.

If you have never soldered or you want to improve your soldering techniques I recommend you the [Adafruit Guide To Excellent Soldering](#)

For better understanding where is located each component on the board check out the *PCB* layout with the interactive BOM.

2.2.2 Powering

The Smart Lights X1 can be powered in two ways: through the DC Power Jack **or** through the Serial port, but it is not recommended to be done simultaneously if you don't know exactly how the internal powering routes are set.

Caution: Power the board only after making all the connections

DC Power Jack

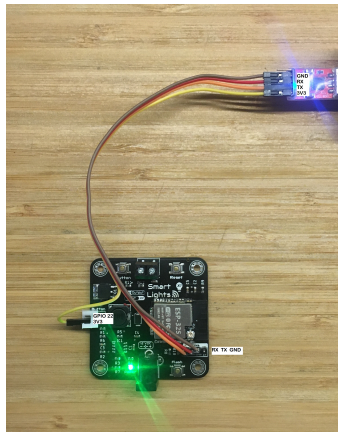
The DC Power Jack is a very common connector for home power supplies. The size of the one by default is a 5.5mm (outer) and 2.1mm (inner) diameter.

Since there are many power supplies with this connector it is important, before plugging any power supply to the board, checking out a few important details:

- The output voltage of the power supply has to be in the range of 5 to 24VDC.
- The power of the power supply is enough for the LEDs you want to power.
- The connector polarity is correct. Inner pole positive and outer negative



Serial port



Despite the normal usage of the board will be usually done through the DC Jack, whenever you need to program the board manually, it is recommendable to power it through the Serial port. However on the Smart Lights X1, the power regulator accepts a minimum voltage of 5V, which makes it unsuitable to bring down to 3.3V the 5V input voltage at Serial port.

Therefore, it is recommendable to make a connection as the figure whenever you want to program your Smart Lights X1 with an external USB-to-TTL programmer.

2.2.3 I/O

The Smart Lights X1 has the capability of reading 1 Input and controlling 1 output. :Input: *GPIO22* :Output: *GPIO21*
The input port is located on the left side, a 2.54 pitched connector that could be directly plugged to a pushbutton.

Note: The input pin is already pulled-down, through a 10k resistor, and connected on the PCB to the pushbuttons.

The output, located on the upper part of the board through a 3.5mm screw terminal, is PWM controlled through the power mosfet. Despite the PCB layout should allow a correct heat dissipation, an additional heatsink would be recommendable for applications where a high current is expected (long LED strips or high intensity LEDs)

2.2.4 Communications

In addition to the I/O mentioned before, there is also a direct connection to:

Serial bus:

Tx
GPIO1

Rx
GPIO3

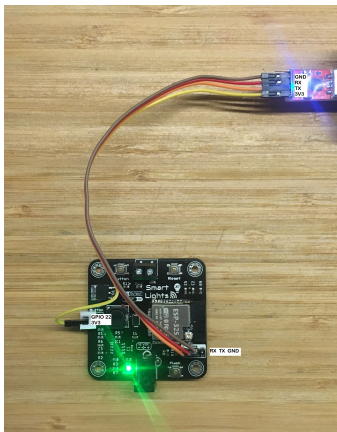
2.2.5 Enclosure

The Smart Lights X1 PCB can be mounted in a custom *3D printable enclosure* that you can print on your own 3D printer.

2.3 Programming

There are two main programming methods supported and tested with the Smart Lights X1:

- ESPHome
- Arduino



In both scenarios, you will first need to enter the board into flashing mode. For that, press and hold the *Flash* pushbutton while you reset the board (pressing once the *Reset* pushbutton).

Important: For flashing new firmwares, if the OTA support is not available, you will need an external UST-to-TTL module (like [this](#)) connected to the Serial port in a way that the 3V3 are delivered directly to the microcontroller, as in the figure.

2.3.1 ESPHome

ESPHome is a well known platform for programming ESP-based devices with a very little effort. It is configured via YAML files and supports a wide range of functionalities and sensors.

Important: For using ESPHome, and all its functionalities, you need to have a [Home Assistant](#) instance running in the same network as your Smart Lights X1.

The Smart Lights X1 already comes with an embedded version of ESPHome, that would only require an OTA update to get it ready to work in your network:

1. Power the board, and let it run for 1-2 minutes. When the board cannot connect to a WiFi network, it will create a fallback hotspot.
2. Use a smartphone or tablet and go to the WiFi settings, connect to the recently created *Smart-Lights-X1* hotspot with the password *smartlights*.
3. Access to the captive portal and open the browser if doesn't pop up automatically.
4. Enter your network settings and press *Save*.

Now, your ESPHome device is ready to be found by Home Assistant in your network. Add it from the ESPHome section to add and edit a customized configuration file.

As an example of such configuration file (and the one flashed on the factory settings of the Smart Lights X1) with all the I/O:

```

1 substitutions:
2   device_name: "smart-lights-x1"
3   friendly_name: "Smart Lights X1"
4   project_name: "smart.lights_x1"
5   project_version: "1.0"
6   ap_ssid: "Smart-Lights-X1"
7   ap_pwd: "smartlights"
8
9 esphome:
10   name: "${device_name}"
11   name_add_mac_suffix: true
12   project:
13     name: "${project_name}"

```

(continues on next page)

WiFi Networks

- NETGEAR
- Not your WIFI
- 3HuiTube_2.4Ghz
- Hello World

WiFi Settings

SSID
Password
Save

OTA Update

Browse...	No file selected.
Update	

(continued from previous page)

```
14     version: "${project_version}"
15
16   esp32:
17     board: esp32dev
18     framework:
19       type: arduino
20
21   # Enable logging
22   logger:
23
24   # Enable Home Assistant API
25   api:
26
27   # Enable Over The Air updates
28   ota:
29
30   #Public location of this yaml file
31   dashboard_import:
32     package_import_url: github://JGAguado/
33     ↪Smart_Lights/docs/source/files/
34     ↪configuration.yaml@X1
35     import_full_config: false
36
37   # Enable fallback hotspot (captive portal) ↪
38   ↪in case wifi connection fails
39   captive_portal:
40
41   esp32_improv:
42     authorizer: none
43
44   improv_serial:
45
46   wifi:
47     ap:
48       ssid: "${ap_ssid}"
49       password: "${ap_pwd}"
50
51   light:
52     - platform: monochromatic
53       name: "Light 1"
54       id: light_1
55       output: l1
56
57   output:
58     - platform: ledc
59       id: l1
60       pin: GPIO23
61
62   binary_sensor:
63     - platform: gpio
64       name: "PB1"
```

(continues on next page)

(continued from previous page)

```

63  pin:
64      number: GPIO32
65  on_press:
66      then:
67          - light.toggle: light_1

```

2.3.2 Arduino

If you are still interested in programming directly with the Arduino IDE, the procedure is no different than with any other ESP32 devices:

1. Open the Arduino IDE and go to File -> Preferences option.
2. Add to the *Additional Boards Manager URSLs* the url:

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.
↪ json
```

3. Close the preferences and open in the menu Tools -> Board -> Boards Manager.
4. Search for *esp32* and install it. This might take some time.
5. Now you can select the board *ESP32 Dev Module* as the target board. Leave the rest of parameters by default.
6. Select the correct port and remember to enter the board into flashing mode before uploading the sketch.

2.4 FAQ

After connecting everything and turning on the Smart Lights X1, the power LED doesn't turn on, why?

If the board is connected correctly and you cannot see the power LED on, disconnect the power immediately. Please check all the connections and measured with a multi-meter the resistance between 3V3 and gnd in any port (*I²C* or Serial). If the resistance is close to zero you have a short circuit somewhere in your board, please check your connections and any soldering you've made. If the problem persists reach [Technical support](#). In addition, remember that this version of the Smart Lights X1 (V1R2) cannot receive 5V through the Serial port "5V" pin, see the [Programming](#) section.

Can I upload ESPHome directly from the Home Assistance setup running on my Raspberry Pi?

Yes you can, and actually this is one of the simplest and more effective way to upload ESPHome.

Can I upload firmware if I don't have an external USB-to-TTL programmer?

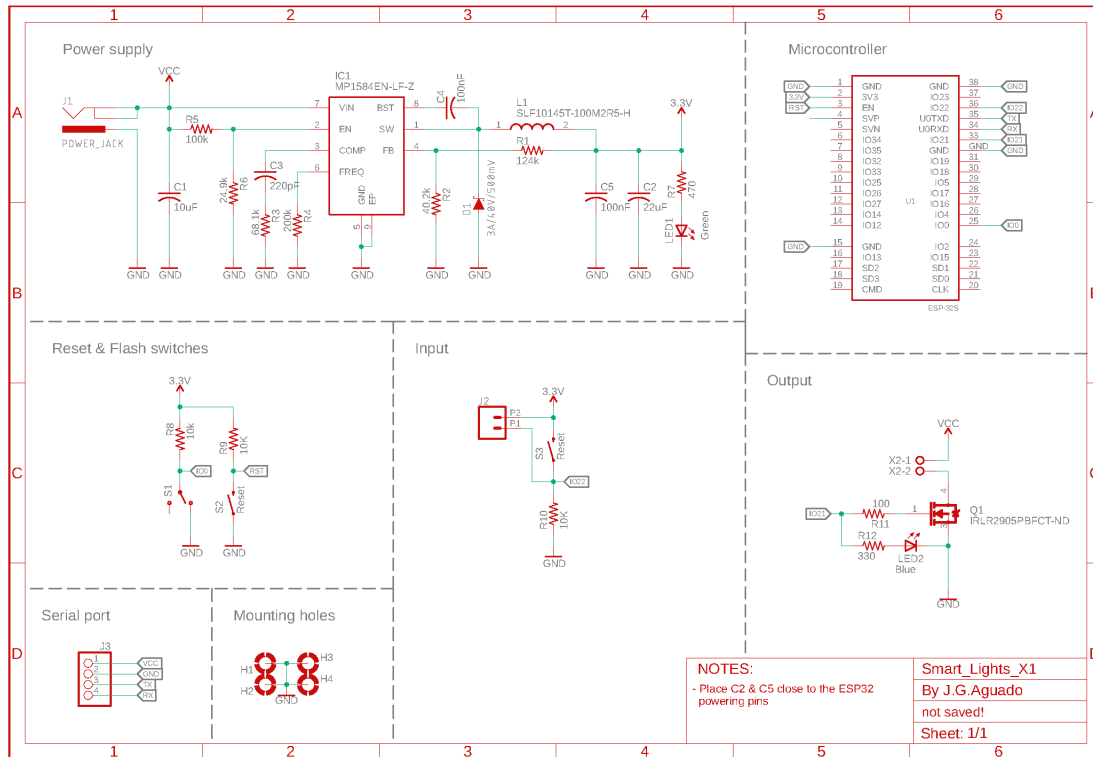
At the moment, this is the only way in order to minimize the components costs. But don't worry, in general it's a procedure that should be done just once.

I read that it can control 12VDC led strips and 24VDC, which one is the correct?

The electronics onboard are capable to power the ESP-32 in a range of 6 to 24VDC, meaning that is just up to your LED strips: if you have a 12VDC LED strip, just choose correctly the power supply for delivering 12VDC.

2.5 Schematics and drawings

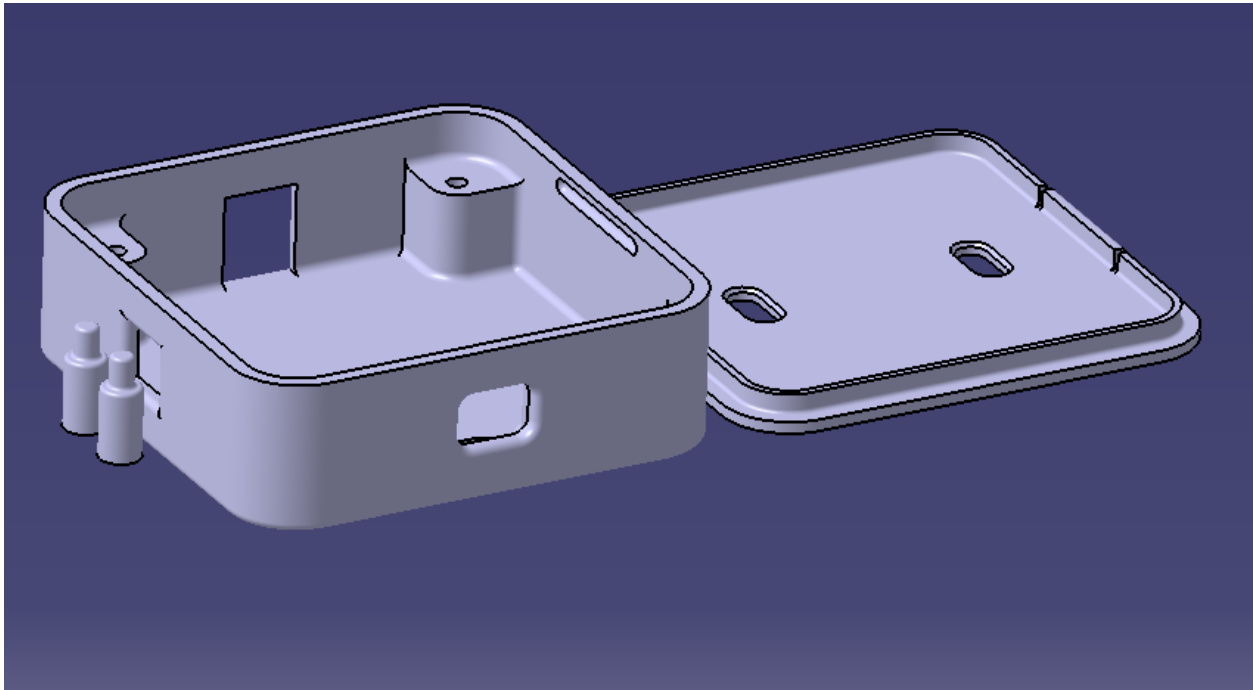
2.5.1 Electronic schematics



Download PDF

2.5.2 PCB

2.5.3 3D printable enclosure



STL Downloads:

- Bottom
- Top
- Button

2.6 Technical Specifications

2.6.1 Electrical

Parameter	Symbol	Min.	Max.	Unit
Supply voltage (DC)	V_{DC}	9	24	V
Power	P		10	W

2.6.2 Environment

Parameter	Symbol	Min.	Typ.	Max.	Unit
Working temperature ¹	T_{amb}	0	25	60	°C
Storage temperature ¹	T_{amb}	0	25	60	°C

¹ Based on components datasheets.