
Smart Garden

Release 2

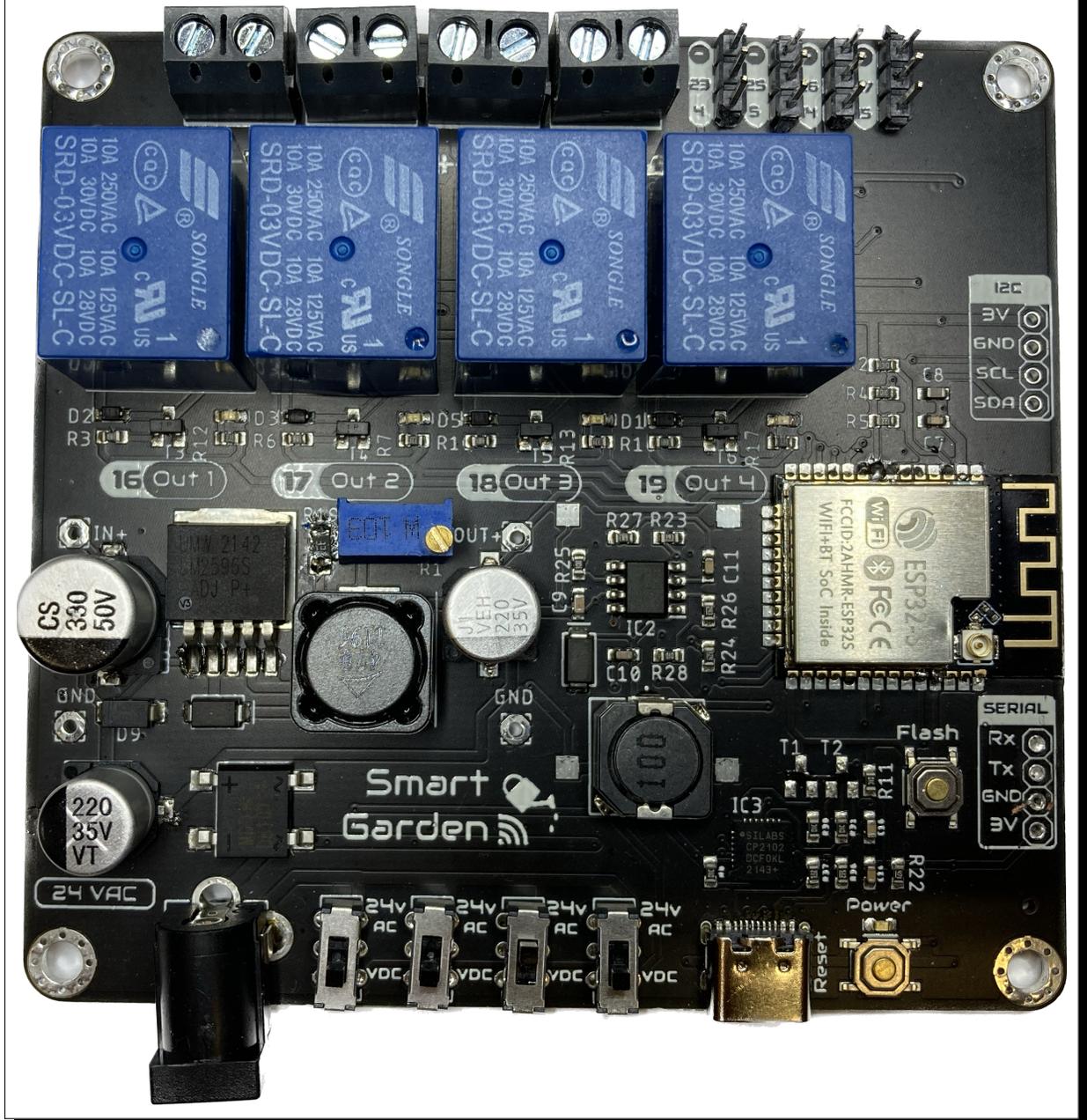
J.G.Aguado

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	10
2.4	FAQ	14
2.5	Schematics and drawings	15
2.6	Technical Specifications	16

Smart Garden V1



Welcome to the documentation page of the Smart Garden V1, an electronic board based on the ESP32 capable of controlling a combination of 4 irrigation valves and/or water pumps, supporting AC & DC voltages.

CONTENTS

Safety guidelines

How to operate and manipulate the Smart Garden V1.

Getting started

First steps for configuring and working with the Smart Garden V1.

Schematics and drawings

Design files of the Smart Garden V1.

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 Garden V1

Revision

2

Date

April 12, 2023

2.1 Safety guidelines

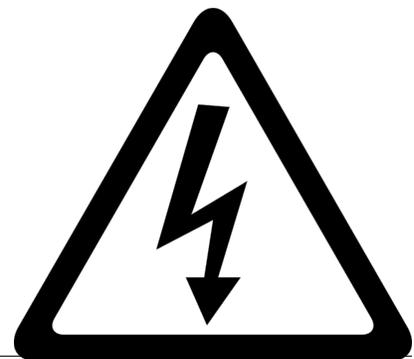
These safety guidelines, the operation instructions and the limit values listed in *Technical Specifications* are to be read carefully before operating the product.

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 Garden V1 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 Garden V1 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

Caution: Power the board only after making all the connections

The Smart Garden V1 is powered directly through a Power Jack connector located on the bottom part of the board. The recommended size for the input connector is a 5.5mm (outer) and 2.1mm (inner) diameter power jack.

As the Smart Garden V1 was designed to control sprinkles, it is expected to be powered with a 24VAC source, such as the **NIMO YL15-2401000 220VAC to 24VAC (1A)**.

Alternatively, and if you are not expecting to control AC sprinkles, powering the board from any VDC in the range of 12-24V is possible as well.

Note: Choose always a power supply that can deliver the same or more (within the *Technical Specifications* limits) voltage than your loads (solenoid valves/ pumps), as the DC/DC regulator operates as a step-down converter.

2.2.3 I/O

The Smart Garden V1 supports up to 4 independent *analog outputs* and 8 *digital inputs/outputs*:

Table 1: Pinout table

GPIO	Input	Output	Name
4			Auxiliar 1A
5			Auxiliar 2A
14			Auxiliar 3A
15			Auxiliar 4A
16			Output 1
17			Output 2
18			Output 3
19			Output 4
23			Auxiliar 1B
25			Auxiliar 2B
26			Auxiliar 3B
27			Auxiliar 4B

Outputs:

Each one of these outputs are individually controlled through relays, so they can only be on True/False states. However, the type of output voltage can be assigned previously via hardware:

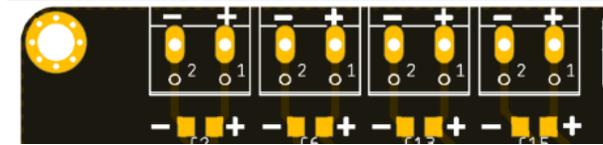
Next to the power jack connector, there are 4 slide switches that configure each of the 4 outputs as DC or AC:

- If the 24 VAC is selected, the controlled relay will output 0 or 24VAC. This is the way to control solenoid valves such as the [Hunter PGV series](#) or the [Gardena's Irrigation Valve 24V](#)
- If the VDC is selected, the controlled relay will output 0 or a regulated VDC voltage. This DC voltage can be adjusted through the blue trimmer in a range of 3.3 up to the input voltage through the jack connector. This is the way to control DC water pumps up to 10W.

The connection port of each of the outputs is located on the top left part of the board and is done through 3.5mm screw terminals. The polarity of the connection (for DC operations) is defined on the PCB silkscreen.

From left to right, the pin definition of each output is:

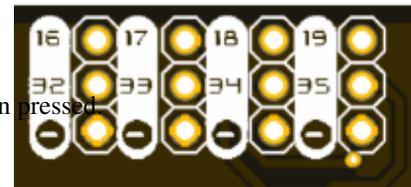
- Output 1**
GPIO16
- Output 2**
GPIO17
- Output 3**
GPIO18
- Output 4**
GPIO19



Digital inputs/outputs:

The connection port of these pins is located on the top right part of the board and is done through 2.54 pins.

These pins are grouped in 4 sets, containing:
:Auxiliar 1: *GND*, *GPIO23* & *GPIO4*
:Auxiliar 2: *GND*, *GPIO25* & *GPIO5*
:Auxiliar 3: *GND*, *GPIO26* & *GPIO14*
:Auxiliar 4: *GND*, *GPIO27* & *GPIO15*



Hint: These pins can be used to read if ring led powered pushbuttons has been pressed.

Analog inputs:

In order to provide a feedback of the DC regulated voltage that would be output, pin *GPIO32* offers a measurement of the real voltage once regulated.

Since it is designed in a voltage divider configuration for avoiding the input read above 3.3V to the microcontroller, the correspondent equation for obtaining the real voltage is as follows:

$$V_{real} = \frac{11.2k\Omega}{1.2k\Omega} \cdot V_{measured} = 9.33 \cdot V_{measured}$$

2.2.4 Communications

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

IIC (I^2C) bus:

- SDA**
GPIO21
- SCL**
GPIO22

Serial bus:

Tx
GPI01

Rx
GPI03

2.2.5 Enclosure

The Smart Garden V1 has been designed to fit in the electronics enclosure LK-PLC01, compatible with DIN rails and screws, and it is recommended for indoors only.



External size
115x90x40mm

Material

ABS Plastic

Color

Transparent cover, black or beige base

2.3 Programming

There are many firmwares supported with the Smart Garden V1 (almost any supporting the ESP-32). Some of the tested working are:

- ESPHome
- Tasmota
- Arduino

In any case scenario, and regardless you are using the USB port or the Serial port for programming it, 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).

Caution: When flashing the board, make sure its only powered by the USB/Serial port.

Note: It is possible that after flashing your board and while rebooting, the ESP-32 has some issues to boot. This is due to a higher demand of power while

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 Garden V1.

The Smart Garden V1 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-Garden* hotspot with the password *smartgarden*.
3. Access to the captive portal and open the browser if doesn't pop up automatically.
4. Enter your network settings and press *Save*.

WiFi Networks

- NETGEAR
- Not your WiFi
- 3HuTube 2.4Ghz
- Hello World

WiFi Settings

SSID
Password

Save

OTA Update

Browse... No file selected.

Update

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 Garden V1) with all the I/O:

```

1 substitutions:
2   device_name: "smart-garden"
3   friendly_name: "Smart Garden"
4   project_name: "smart.garden"
5   project_version: "1.0"
6   ap_ssid: "Smart-Garden"
7   ap_pwd: "smartgarden"
8
9 esphome:
10  name: "${device_name}"
11  name_add_mac_suffix: true
12  project:
13    name: "${project_name}"
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_Garden/docs/source/files/
34   ↪configuration.yaml
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:

```

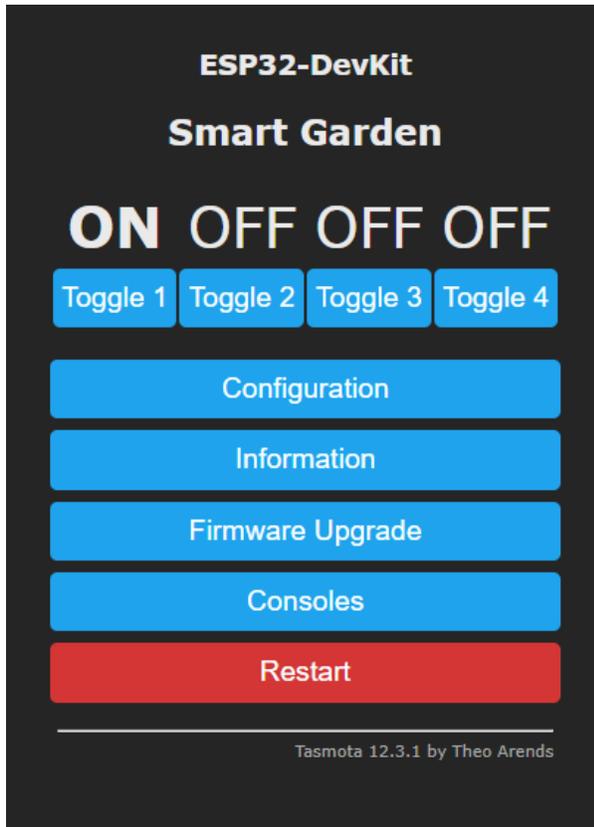
(continues on next page)

(continued from previous page)

```
44  ap:
45    ssid: "${ap_ssid}"
46    password: "${ap_pwd}"
47
48  # These relays will automatically turn off
49  ↪after 2 minutes
49  switch:
50    - platform: gpio
51      pin: 16
52      id: relay_1
53      name: "Zone 1"
54      icon: "mdi:sprinkler-variant"
55      on_turn_on:
56        - delay: 120s
57        - switch.turn_off: relay_1
58
59    - platform: gpio
60      pin: 17
61      id: relay_2
62      name: "Zone 2"
63      icon: "mdi:sprinkler-variant"
64      on_turn_on:
65        - delay: 120s
66        - switch.turn_off: relay_2
67
68    - platform: gpio
69      pin: 18
70      id: relay_3
71      name: "Zone 3"
72      icon: "mdi:sprinkler-variant"
73      on_turn_on:
74        - delay: 120s
75        - switch.turn_off: relay_3
76
77    - platform: gpio
78      pin: 19
79      id: relay_4
80      name: "Zone 4"
81      icon: "mdi:sprinkler-variant"
82      on_turn_on:
83        - delay: 120s
84        - switch.turn_off: relay_4
85
86  sensor:
87    - platform: adc
88      pin: A4
89      name: "VCC"
90      update_interval: 60s
91      filters:
92        - calibrate_linear:
93          - 0 -> 0
94          - 0.55 -> 5
```

2.3.2 Tasmota

As an alternative easy-to-use and still powerfull, you can flash Tasmota into the Smart Garden V1 directly. This option is highly recommendable if you want to have an Alexa compatible device (based on a Hue emulated bridge) without the need of a full Home Assistant setup.



The easiest process of flashing Tasmota into your Smart Garden V1 goes as follows:

1. Plug the Smart Garden V1 into your computer's USB.
2. Go to the [Tasmota Web Installer](#)
3. Follow the process selecting ESP32 as device and select the correct port (remember to enter the board into flashing mode).
4. After the process is completed, press the reset button.
5. Once rebooted, the Smart Garden V1 will create a wifi hotspot containing the name *Tasmota*. Connect and enter your WLAN setup on the captive portal.
6. Get the IP assigned to the board (check your router's connected devices) and enter it into your browser for accessing to the Tasmota's WebUI.

Now you can customize your board as you want. Let's first assign the output class '*Relay*' to the corresponding outputs 16, 17, 18 and 19.

In addition, some of these other commands can be entered through the **Console** (*Consoles -> Console*):

- **Timed output:** With the command `PulseTime3 300` you can make that the output 3 turns off after 5 minutes (300s). This option is highly indicated as a safety feature to prevent your home to get flooded.
- **Interlocks:** With the command `interlock on` you can make that ONLY one of the 4 outputs can become active at the same time. For selecting the group of the for relays just use the command `interlock 1,2,3,4`. This mode can become usefull when your home's

water pressure is not high enough for holding more than one watering zone at the same time.

In any case, you can find these and much more commands to customize your Tasmota's flashed Smart Garden V1 on the [official Tasmota documentation](#)

And, by the way...

If you want Alexa to directly recognize your Smart Garden V1 to directly control each output through voice controls, go to *Configuration -> Configure Other* and select the option 'Hue Bridge multi device' emulation.

2.3.3 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 Garden V1, 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](#)

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 through the Serial bus?

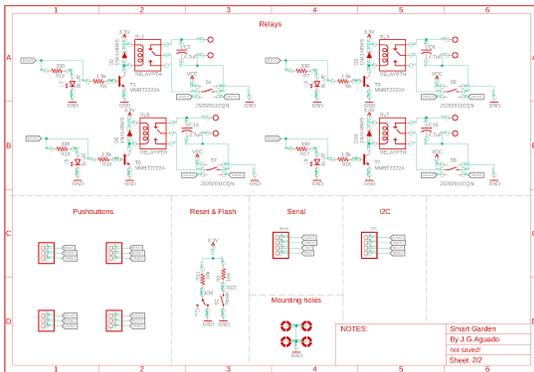
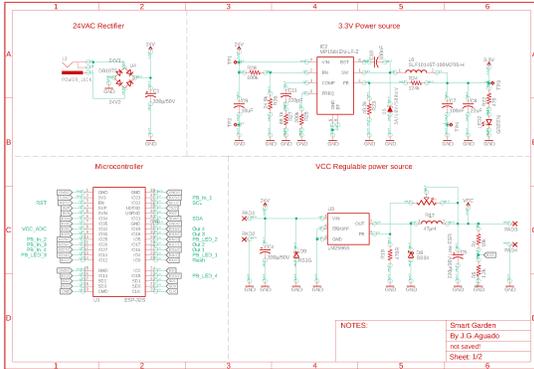
Yes you can, however this is only recommended if you have troubles using the USB port and you are familiar with the procedure.

I want to use the IIC port for expanding the possibilities, how can I do it?

Once you have located the sensor that you want to connect to the IIC port, check that is compatible with the 3.3V power supply from the Smart Garden V1. If this is the case, just connect it directly, power the board and update the firmware!

2.5 Schematics and drawings

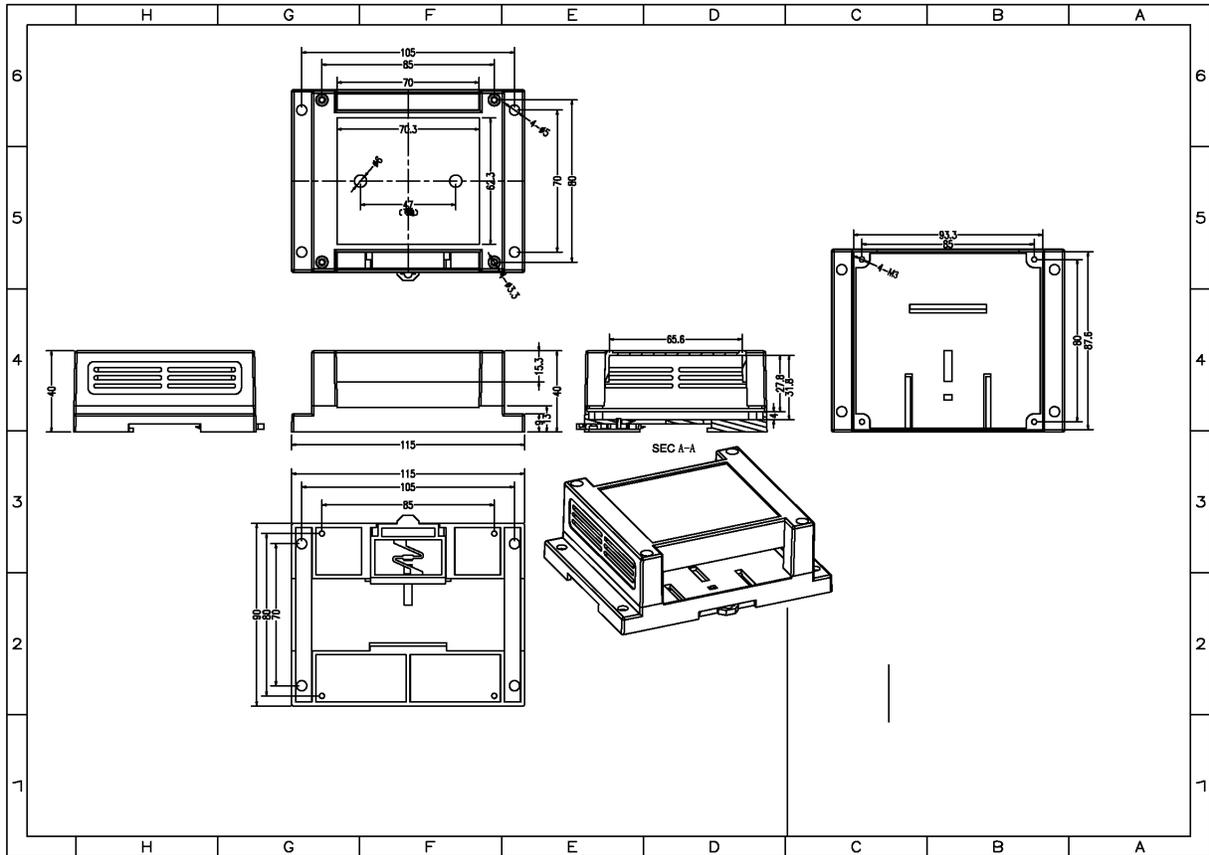
2.5.1 Electronic schematics



[Download PDF](#)

2.5.2 PCB

2.5.3 Enclosure drawings



[Download PDF](#)

2.6 Technical Specifications

2.6.1 Electrical

Parameter	Symbol	Min.	Max.	Unit
Supply voltage (AC)	V_{AC}	12	24	V
Supply voltage (DC)	V_{DC}	12	24	V
Power	P		15	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.