

1. General information

1.1. Description and characteristics

The world is evolving at a breakneck pace, and we are helping it do so. To support developers, innovators, enthusiasts, and others, we have created the Atom Black Edition product, and provide the opportunity to use our platform to develop new products and solutions in the world of automation or IoT. Our devices are designed with a modular architecture, giving you a possibility to choose any peripherals you need for your research or development.

We have developed two types of controllers in this series to suit any of your needs. We provide you with the ability to program under STM32 and ESP32 controllers, which meet the standards of stability and reliability in today's world.

This series can be used when writing your own software, as well as to use existing solutions popular in the world of automation and IoT (micropython, esphome.io, tasmota and others).

Our devices can be used by integrators, laboratories, startups, as well as educational institutions to train the next generation of engineers. Atom Black Edition series controllers are perfectly protected against overloads, erroneous connections and also have excellent protection in the form of the case, to prevent accidental contact with electrically conductive parts.

Atom Black Edition is based on the powerful and multifunctional module [Espressif ESP32-PICO-D4](#).

Features of ESP32-PICO-D4:

- Based on ESP32, 32-bit Xtensa® dual-core LX6 (600 DMIPS)
- 32 Mbps Flash, 520 KB RAM, 448 KB ROM, 1 KB eFuse
- Standards 802.11 b / g / ni and Bluetooth v4.2 BR / EDR / BLE
- Placing all the necessary components inside the chip
- Interfaces GPIO, UART, SPI, SDIO, I2C, PWM, I2S, IR, ADC, DAC
- Built-in Hall and temperature sensors

This solution opens up the possibility of using Atom series controllers in already established or new automation systems as a multifunctional I / O unit with extensive integration capabilities. The most popular platform for implementing a "smart home", where Atom controllers can be used based on the Espressif ESP32-PICO-D4 module - control system [Tasmota](#).

Tasmota is a system for managing devices based on ESP8266 / ESP32 (beta) using simple settings via a web-interface with advanced integration capabilities into home automation systems.

[Detailed characteristics ESP32-PICO-D4](#)

1.2. Modifications

All models of Atom controllers with proper configuration files are suitable for use in Tasmota automation systems. The difference from the basic models is that the controller uses a board based on ESP-Pico-D4 instead of the CPU board STM32. There are also no CAN and RS-485 interfaces. Connectors in the appropriate positions are used to program the controller with a USB cable.

Modifications with the OLED screen which can be easily adjustable in a configuration file are available.

2. Getting started - firmware, settings.

2.1. Software tools to upload the firmware to the device

To work in the automation system, the controller requires configuration, which is done by uploading to it a firmware file in *.bin format, followed by configuration via the web-interface. The controller can be used both as a separate control device and as part of the Home Assistant automation system using the MQTT and Tasmota integration software modules. In both of these cases, the firmware is uploaded using the USB-interface, all further settings are easier to do through the web-interface, using the home WiFi network.

At the software level, there are several options for downloading firmware to the controller:

- **Tasmotizer** - NEW flashing and firmware download tool just for Tasmota. (Windows, Linux or Mac)
- **Tasmota PyFlasher** - flashing tool intended for Tasmota. (Windows or Mac)
- **NodeMCU PyFlasher** - easy to use GUI flasher based on esptool.py. (Windows or Mac)
- **Esptool.py** - the official flashing tool from Espressif. (Requires Python)
- **Esptool executable** - Esptool in executable form, no Python required. (Windows, Linux or Mac)

For control and management, you can use the built-in web server, which allows you to monitor the status of inputs and outputs and information from the sensors, and provides a simplified interface for managing connected devices.

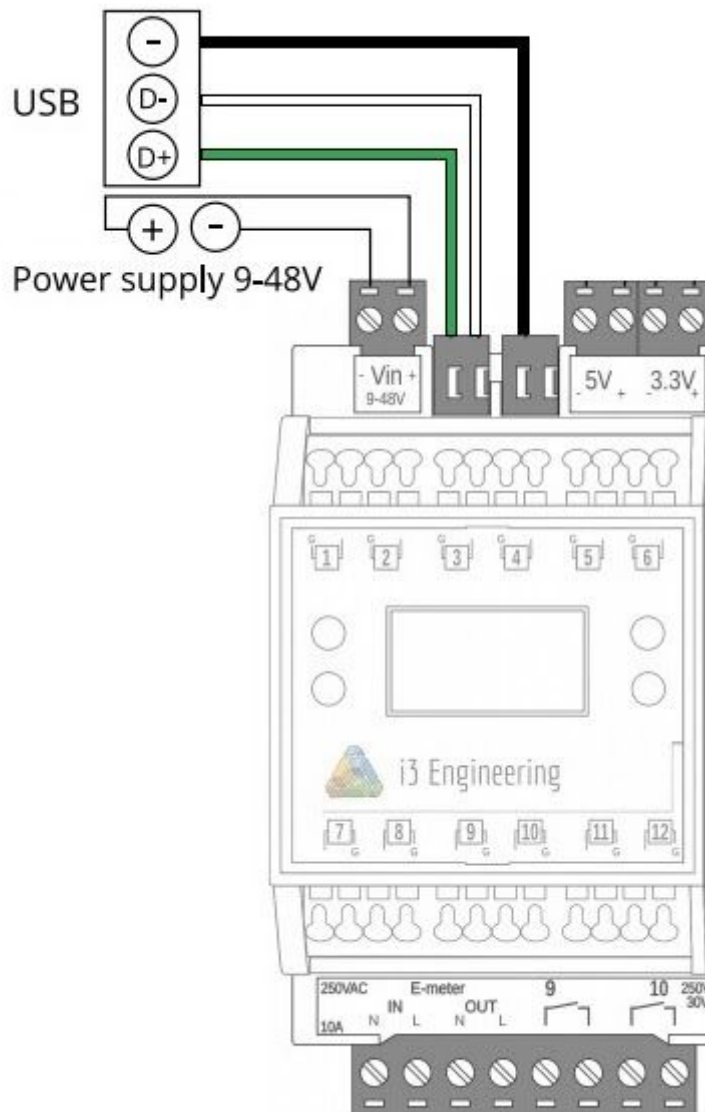
Incomparably more convenient and functional is the solution based on the integration of devices with Tasmota firmware into the home automation system which, in addition to advanced graphical interface for control and management provides data exchange between devices, to coordinate the system as a whole.

The list of possible integrations can be found on the relevant [project page](#)

2.2. Options for connecting the device to upload the firmware

Firmware upload via USB

1. Download and install the [built-in USB-UART adapter driver](#).
2. Connect the USB cable to the controller according to the diagram



3. Turn on power to the device.
4. Start the firmware upload process according to the [appropriate instructions](#)
5. Check that the controller is connected to the WiFi network.

OTA update

All the following [updates](#) of the controller connected to the home WiFi network will be done using OTA

3. Creating a configuration of the connected devices

3.1. Inputs settings

All of the equipment from the list, with the appropriate settings in the configuration file, presented on the pages of the Tasmota project can be connected to the inputs of the controller:

<https://tasmota.github.io/docs/Components/>

<https://tasmota.github.io/docs/Features/>

Inputs			
input type / input number	1, 2, 4	3	5-12
analog input	-	+	+
digital input	+		
digital output	+		
digital interface	+	+	-
optocoupler mode (selected by jumper)	+		

3.2. Outputs settings

Actuating devices that can be connected to the outputs of the controller, and their number, can be selected depending on the model of the controller:

Кількість та тип виходів						
	Hydrogen	Ferrum	Argon	Neon	Helium	Carbon
Вільні слоти для модулів	8					
Реле 5А	2	2	18	4	2	2
Реле 10А		8			4	
Реле 16А						6
АС димери					3	
MOSFET (DC <48V)				8		

The following interchangeable modules are available for the Atom Hydrogen Black Edition controller: relay 5A, DC MOSFET 48V 5A, 0-10V PWM.

Important! In the I / O configuration, the GPIO port numbering of the ESP32-PICO-D4 chip or the MCP23017 port expansion chip is specified, not the numbering of the physical pins of the chips or the numbering of the inputs / outputs of the controller. The table of correspondence between GPIO ports and inputs / outputs of the controller is given in the table below:

Таблиця відповідності входів/виходів контролера з портами GPIO

№ входу контролера	Порт GPIO	Мікросхема	Призначення
1	27	ESP32	in, out, 1-wire, PWM
2	26	ESP32	in, out, 1-wire, PWM, DAC
3	33	ESP32	in, out, 1-wire, PWM, ADC
4	25	ESP32	in, out, 1-wire, PWM, DAC
5	8	MCP23017	in, out
6	9	MCP23017	in, out
7	15	MCP23017	in, out
8	14	MCP23017	in, out
9	13	MCP23017	in, out
10	12	MCP23017	in, out
11	11	MCP23017	in, out
12	10	MCP23017	in, out
№ виходу контролера			
1	4	ESP32	Залежно від моделі контролера чи змінного модуля (таблиця Кількість та тип виходів)
2	9	ESP32	
3	5	ESP32	
4	10	ESP32	
5	2	ESP32	
6	22	ESP32	
7	23	ESP32	
8	32	ESP32	
9	5	MCP23017	реле
10	6	MCP23017	реле
White LED	3	MCP23017	Status LED
Red LED	4	MCP23017	Status LED
Внутрішні з'єднання			
I2C SDA	21	ESP32	I2C шина
i2C SCL	19	ESP32	I2C шина
Enable Modules	18	ESP32	активація модулів у Atom Hydrogen

4. Connecting the equipment

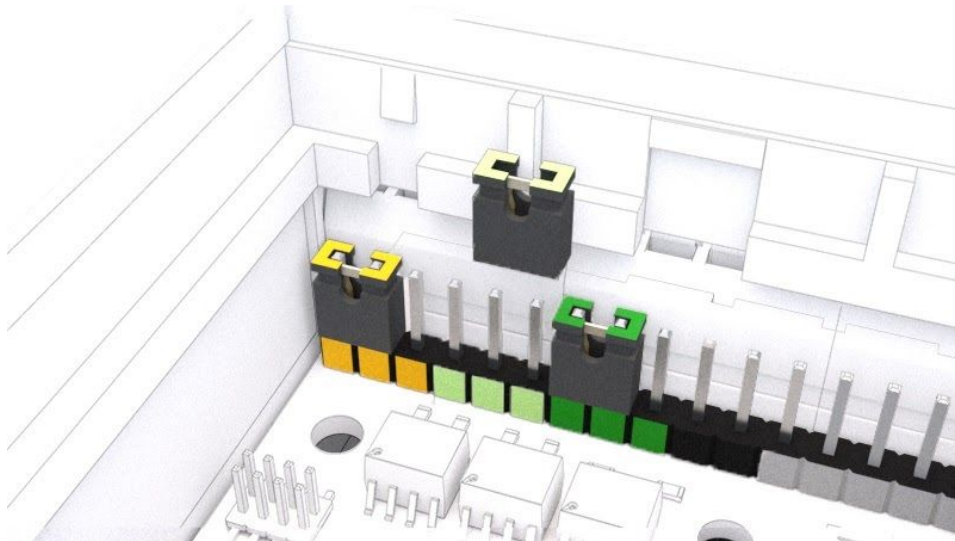
4.1. Connecting inputs.

Connect sensors, switches and other devices to the inputs of the controller. Inputs of the controller connectors are with spring-loaded contacts, which provide convenience and reliability of connection of conductors with a cross section up to 1.5 mm².

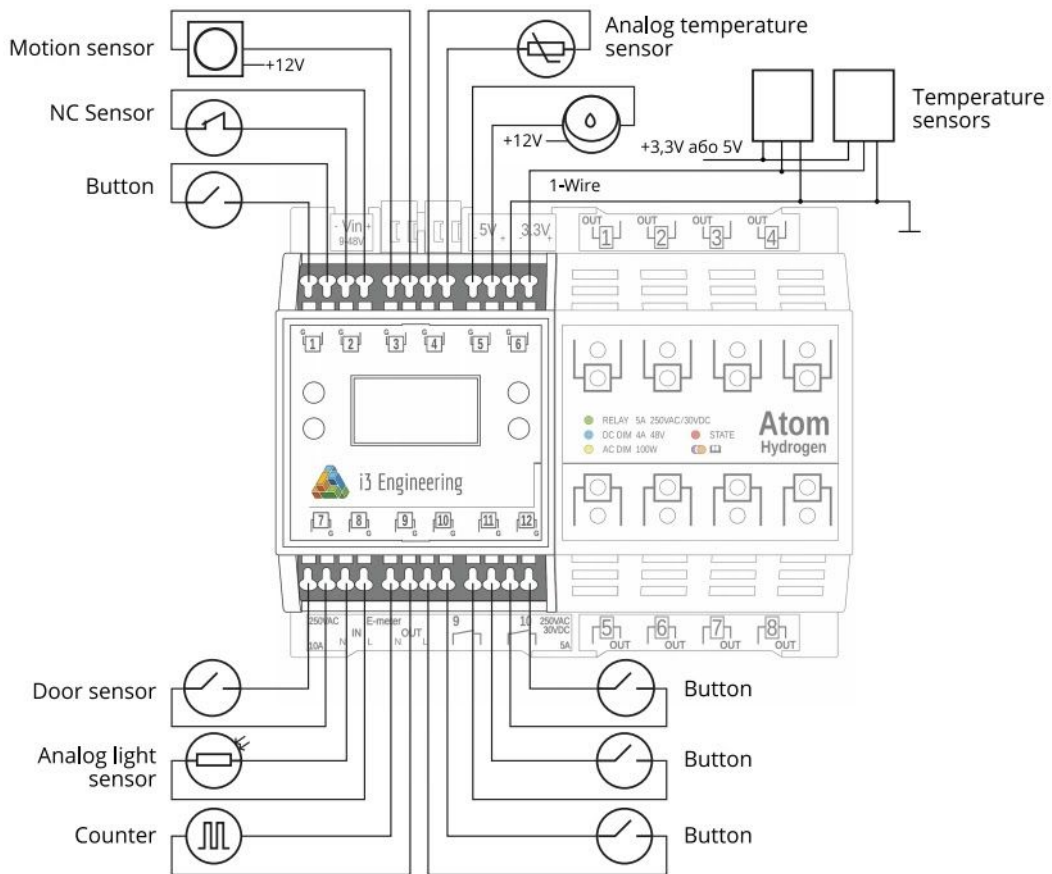
Important: the inputs are numbered and each input has a "common" output marking (marked with the letter "G"), which is connected to the "minus" of the power supply.

3.3V and 5V controller outputs are provided for power supply of sensors. If the system uses sensors or actuators with a different operating voltage, use additional power supplies or the power supply that powers the controller.

By default, all inputs with jumpers are configured in "Optocoupler" ("OP") mode. This is necessary if switches, discrete sensors or other devices are connected to the controller, which due to the specifics of operation, large length of the connection cable, other potentially dangerous factors can generate voltage values exceeding 3.3V at the input of the controller. If analog sensors or a bus with 1-wire devices are connected to the input, the input will be used in digital output mode, the inputs must be reconfigured to direct signal mode. To do this, use a flat-blade screwdriver to remove the front panel of the controller, remove the input display board, and switch the required jumpers on the input board to the "Dir" position.



Example of connecting inputs:

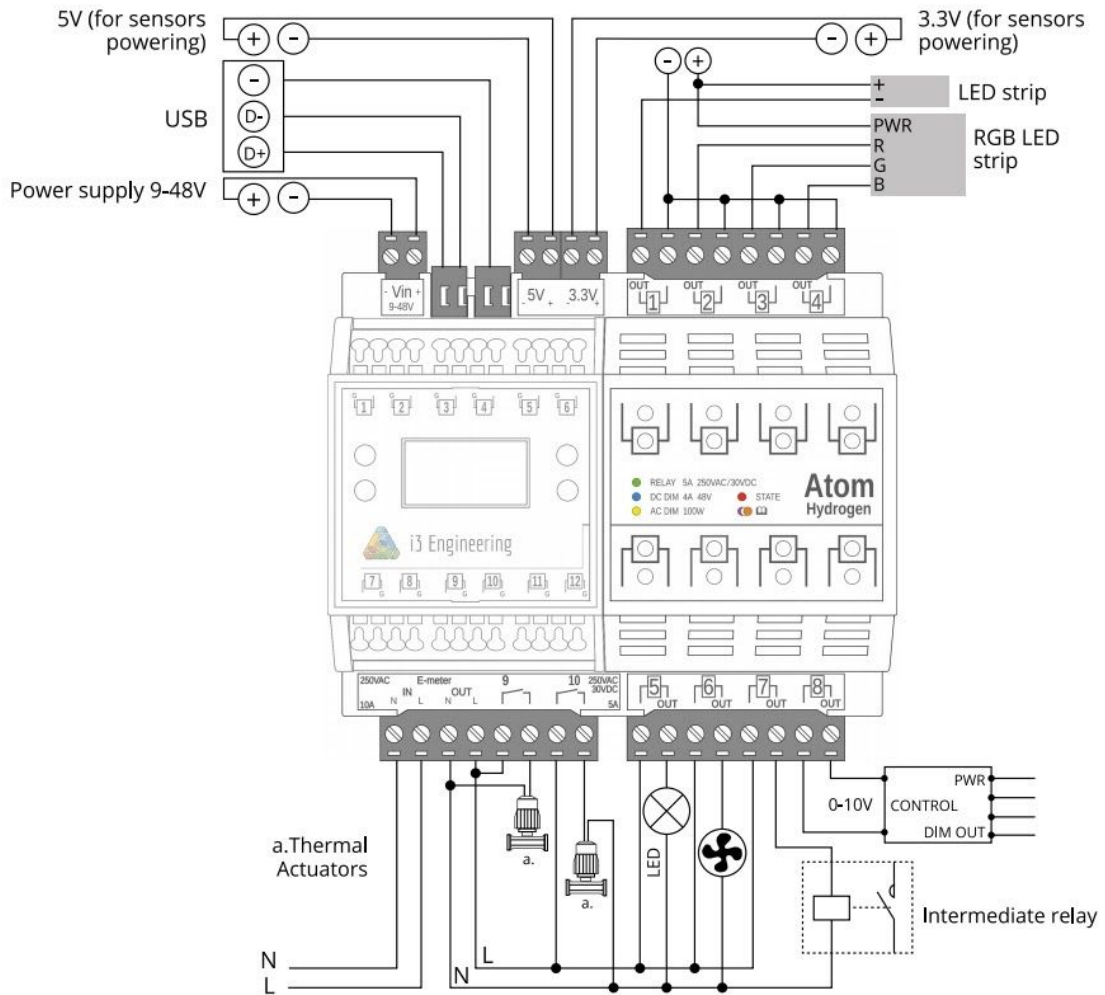


4.2. Connecting outputs

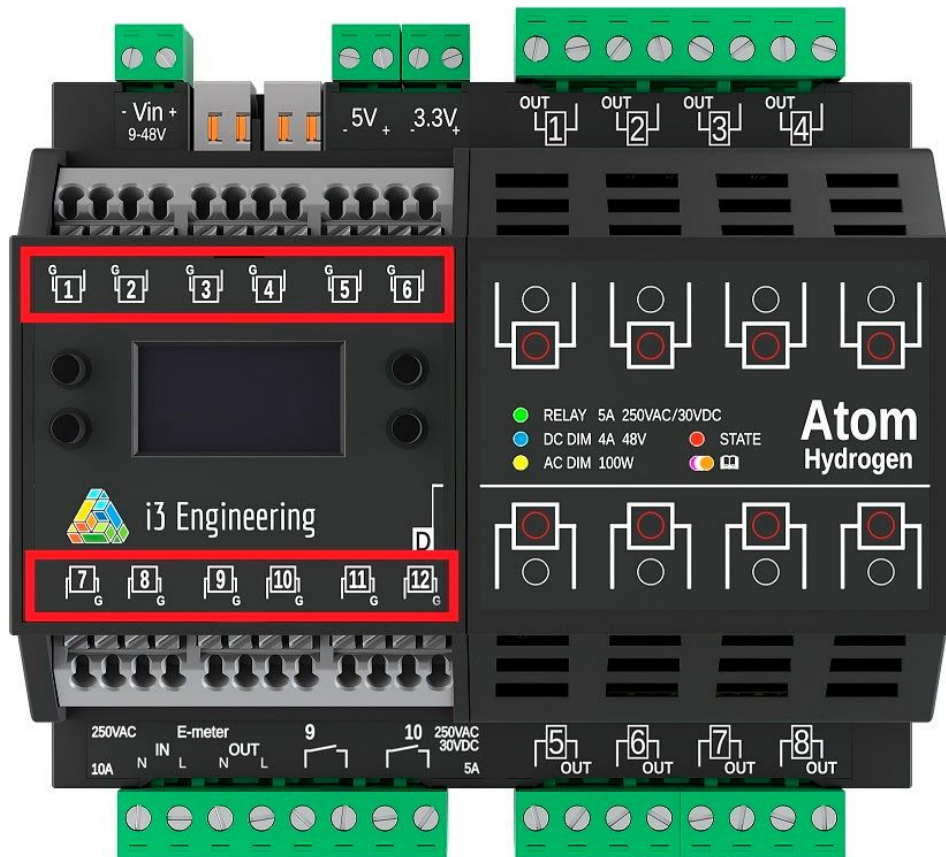
Connect the 220V phase and neutral conductors to the "L" and "N" connectors.

Connect the devices controlled by the controller to the outputs. Each output is implemented in the form of a 2-pin connector with screw terminals. One of the contacts is labeled "OUT". The load is connected according to the type of output.

Example of connecting outputs:

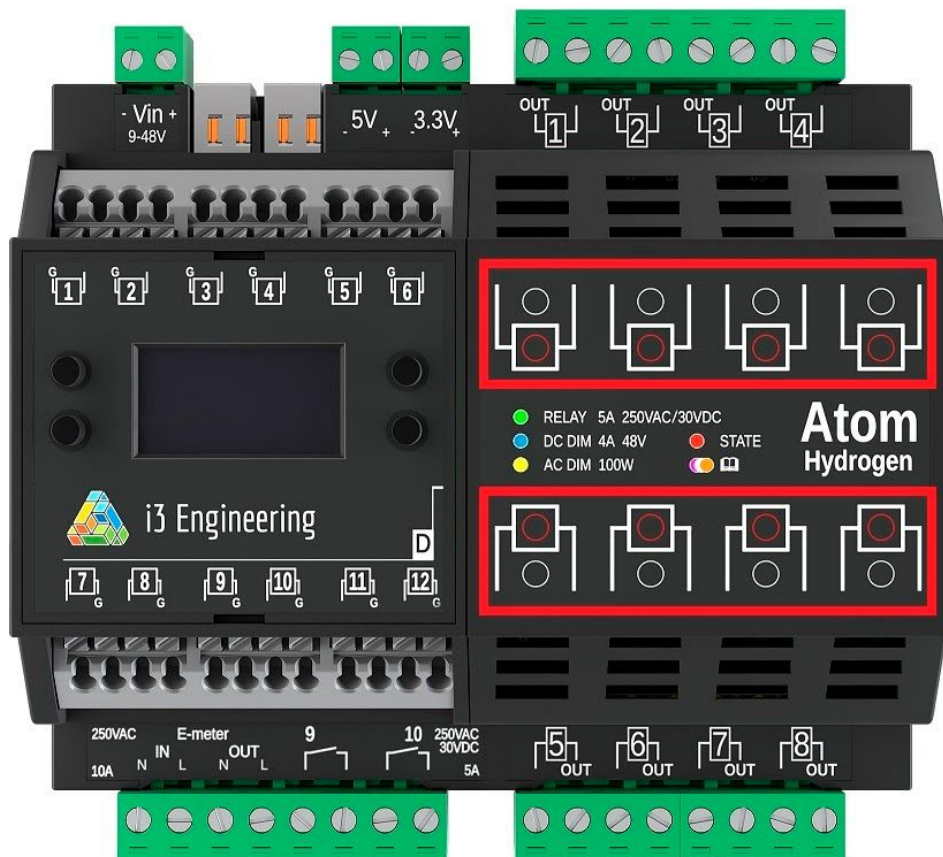


After connecting all devices, you should check their operation. It is convenient to use input and output indicators for this.



Input indicators - show the status of each input;

OLED display - for monitoring and adjustment, display of system parameters of the device, information from sensors.



Output status indicator - red LED that displays the active output status of the controller;

Output type indicators - the color of its glow corresponds to the output type:

- Yellow - Triac or AC MOSFET
- Blue - DC MOSFET
- Green - Relay

5. Automation configuration

The purpose of any automation system is, obviously, the ability to configure all devices to work according to certain rules and algorithms.

In the Tasmota system it is possible to configure tons of such algorithms (automations) using different devices, different logic of their work, using groups of devices and a large number of options for relationships between devices.

Examples of automation implementation are described here:

<https://tasmota.github.io/docs/Rules/>

6. Additional information

6.2. Expansion of functionality

At present, most of the functionality of the Atom series controllers is implemented for the Tasmota system.

Planning to implement the following functionality:

- Using Triac and AC MOSFET interchangeable modules
- Control of current consumption at the outputs equipped with Hall sensors,
- Activate control of the controller using the buttons on the front panel of the controller
- Use of controllers in similar automation systems that support ESP32 - Tasmota and 1mSmart.

6.3. Examples of implementations of various functions

<https://tasmota.github.io/docs/Tutorials/>