

MK4 ESP32-CAM with HomeAssistant integration



Piotr W

[VIEW IN BROWSER](#)

updated 9. 10. 2023 | published 9. 10. 2023

Summary

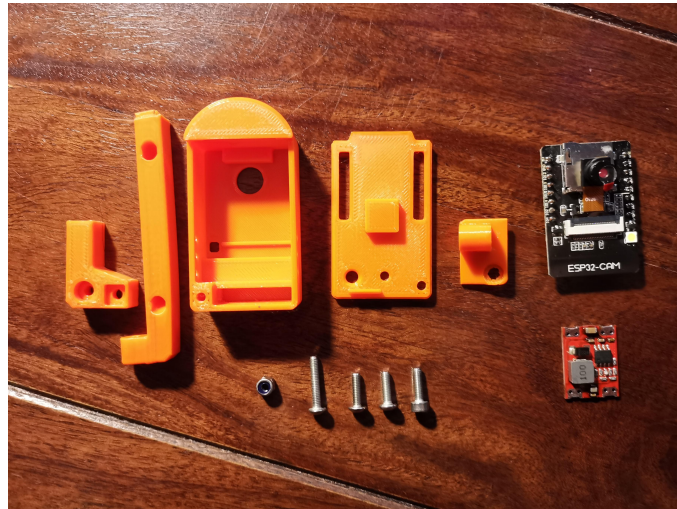
Simply and small MK4 enclosure for the ESP32-CAM + flashlight.
HomeAssistant ESPHOME code below.

[3D Printers](#) > [Prusa Parts & Upgrades](#)

Tags: [esp32](#) [esp32cam](#) [homeassistant](#) [mk4](#) [mk4upgrade](#)
[mk4mod](#)

The enclosure contains a buck converter and an esp32-cam. All the necessary parts you can see in the photo below. You can use any DC-DC buck converter up to 20x16x4mm and 5V or 3.3V output. There is enough room in the case for a small OPTIONAL heatsink (up to 9x9x3mm) to transfer heat from the esp32, there are also holes to ensure air movement. The pins are accessible externally to program the device directly in case of failure. Additional pins can be used for any purpose, e.g. to control the custom printer lighting. The power supply is connected to the motherboard, USE the RIGHT PAIR of power connectors, the left of them are for powering the heat bed and the voltage on them can be unstable. Use the wiring sleeve for the X axis motor to route the cables. SUPPORTS REQUIRED.

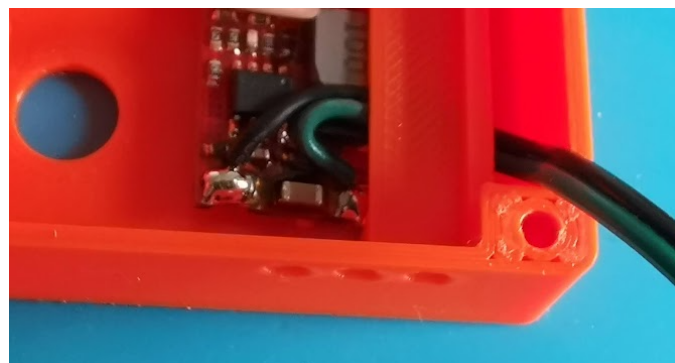
The shorter screws are M3x10, the longer one is M3x16.



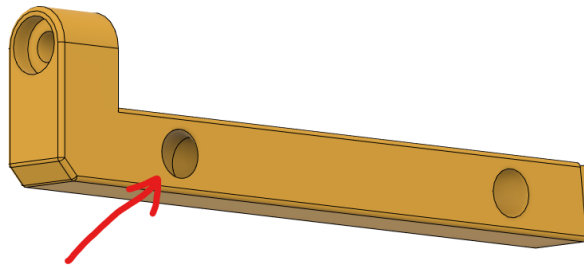
OPTIONAL



The buck must be placed according to the picture, it must be pushed all the way under the cover.



There is a hole in the arm to route the cable, I did not use it.



You will need a programmer and ESPHome software to prepare the chip. You can also use the Arduino IDE and share the video stream directly to Prusa Link/Connect (you can easily find how to do this). The code provides also a LED management (on/off). You will need to rotate video stream or card in HA, it is not possible in ESP.

Use the configuration appropriate to your chip. The following example is for the Ai-Thinker Camera (https://esphome.io/components/esp32_camera).

```
esphome: name: esp32-cam #OPTIONAL #esp32_camera_web_server: # -
port: 80 # mode: stream esp32: board: esp32dev framework: type:
arduino # Enable logging logger: # Enable Home Assistant API api: ota:
password: "USE YOURS" wifi: ssid: !secret wifi_ssid password: !secret
wifi_password captive_portal: esp32_camera: external_clock: pin: GPIO0
frequency: 20MHz i2c_pins: sda: GPIO26 scl: GPIO27 data_pins: [GPIO5,
GPIO18, GPIO19, GPIO21, GPIO36, GPIO39, GPIO34, GPIO35] vsync_pin:
GPIO25 href_pin: GPIO23 pixel_clock_pin: GPIO22 power_down_pin: GPIO32
name: MK4_ESP32-CAM # Flashlight switch: - platform: gpio name:
"Flashlight" pin: GPIO4
```

Model files



case.stl

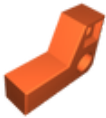




cover.stl



connector.stl



base.stl



arm.stl



light-coveruse-black-filament.stl



lighttuse-transparent-filament.stl

License ©

This work is licensed under a
[Creative Commons \(4.0 International License\)](https://creativecommons.org/licenses/by-nc/4.0/)



Attribution-NonCommercial

- ✗ | Sharing without ATTRIBUTION
- ✓ | Remix Culture allowed
- ✗ | Commercial Use
- ✗ | Free Cultural Works
- ✗ | Meets Open Definition

