ESP32-DevKit Getting Started Guide

This guide shows how to get started with the ESP32-DevKitDOGL128(TFT24/TFT28) development boards.



Overview

ESP32-DevKitDOGL128 is a small-sized ESP32-based very affordable development board using a powerful and popular ESP32-WROOM-32E-N8 module. Some of the I/O pins are broken out to the pin headers for easy interfacing. Developers can either connect sensors using a 2.54mm peripherals with jumper wires or mount ESP32-DevKitI2C-RGB or using the QWIIC (also known as STEMMA QT). This board is a great starting point for your Arduino, Platformio, ESP-IDF and Home Assistant projects.

To cover a wide range of user requirements, the following versions of ESP32-DevKit are available:

ESP32-DevKitDOGL128-BW (Version with Blue/White 128x64 DOGL128B LCD)

ESP32-DevKitDOGL128-YG (Version with Yellow/Green 128x64 DOGL128E LCD)

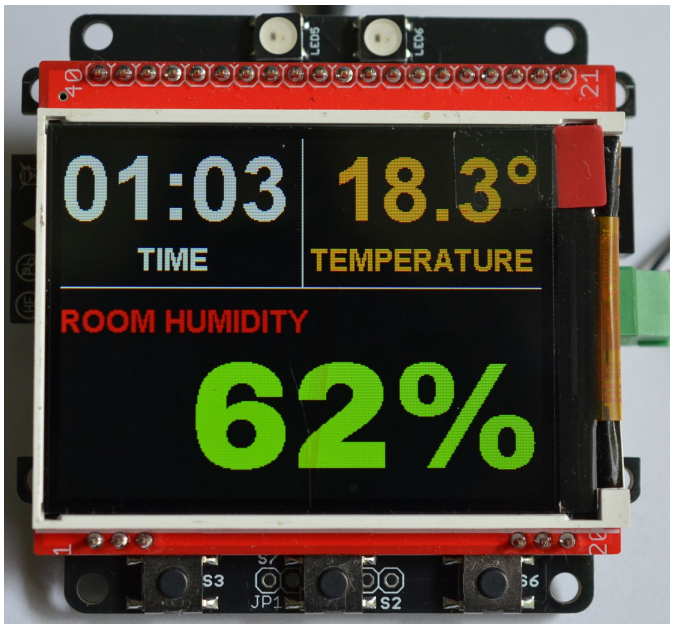
ESP32-DevKitTFT24 (Version with TFT 2,4“ 320x240 TFT full color display)

ESP32-DevKitTFT28 (Version with TFT 2,8“ 320x240 TFT full color display)

ESP32-S3-DevKitTFT24 (Version with TFT 2,4“ 320x240 TFT full color display)

ESP32-S3-DevKitTFT28-TP (Version with TFT 2,8“ 320x240 TFT full color display and RTP)

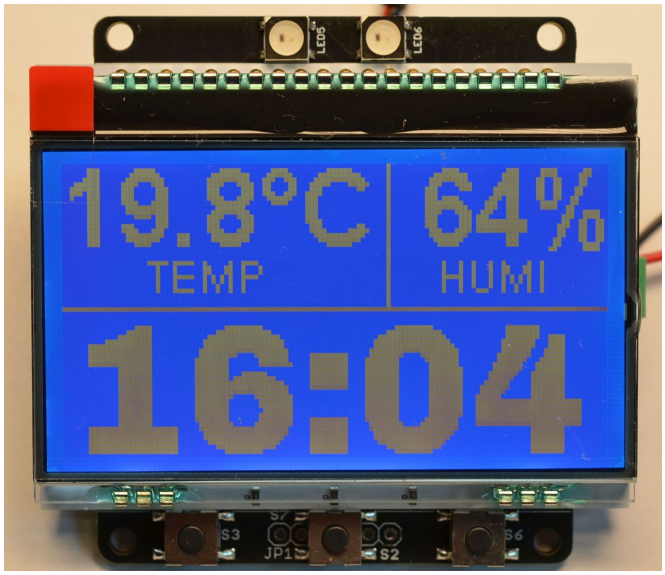
Many other versions are possible with different display color combinations from EA-DOGL128/DOGM128 families with different backlight options even with RGB and sensors (SGP4x footprint is available by default on PCB). For version with SGP40/SGP41 sensor please add SGP40/SGP41 suffix at order (sensor price will be added to base price).



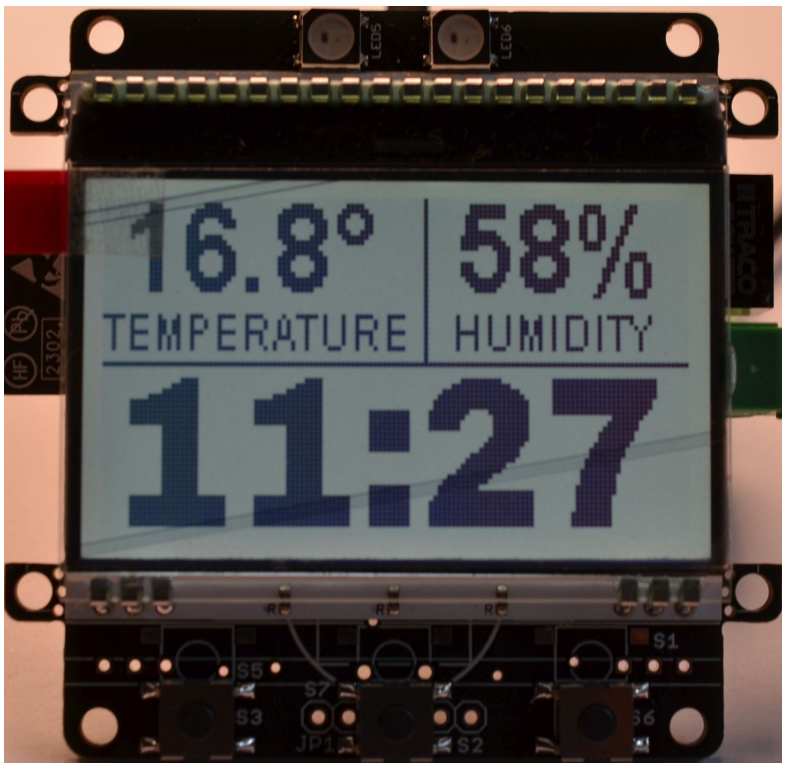
ESP32- DevKitTFT24 with ESP32-WROOM-32-N8 (front side)

Even version with EA-DOGM132 132x32 LCD display is available with EC11B15244A7 rotary encoder.

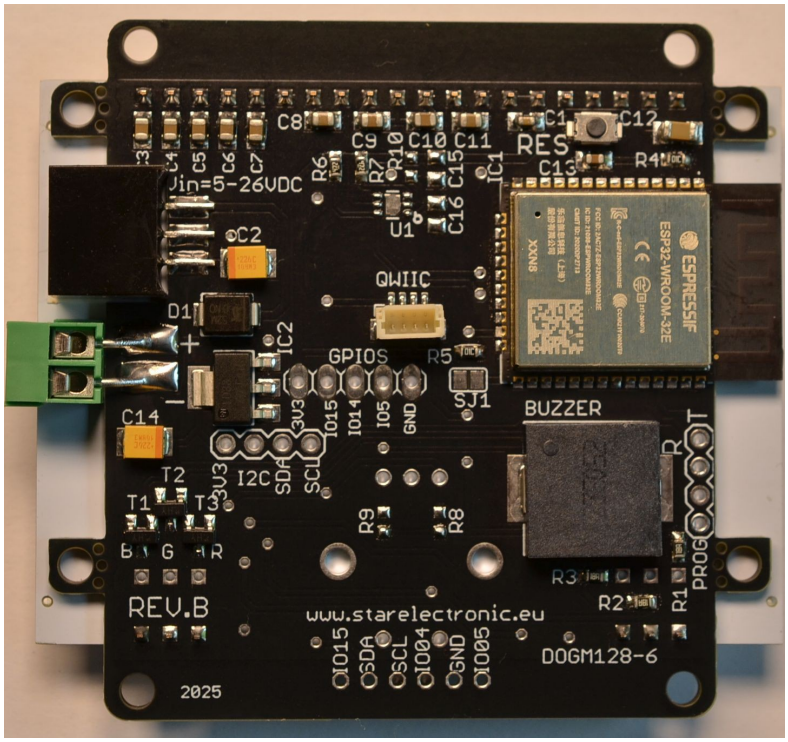
All versions are equipped with QWIIC/STEMMA QT compatible connector in order to allow developers use a wide range of available I2C sensors from different suppliers (Adafruit, Sparkfun, Laskakit, etc)



ESP32- DevKitDOGL128-BW with ESP32-WROOM-32-N8 (front side)



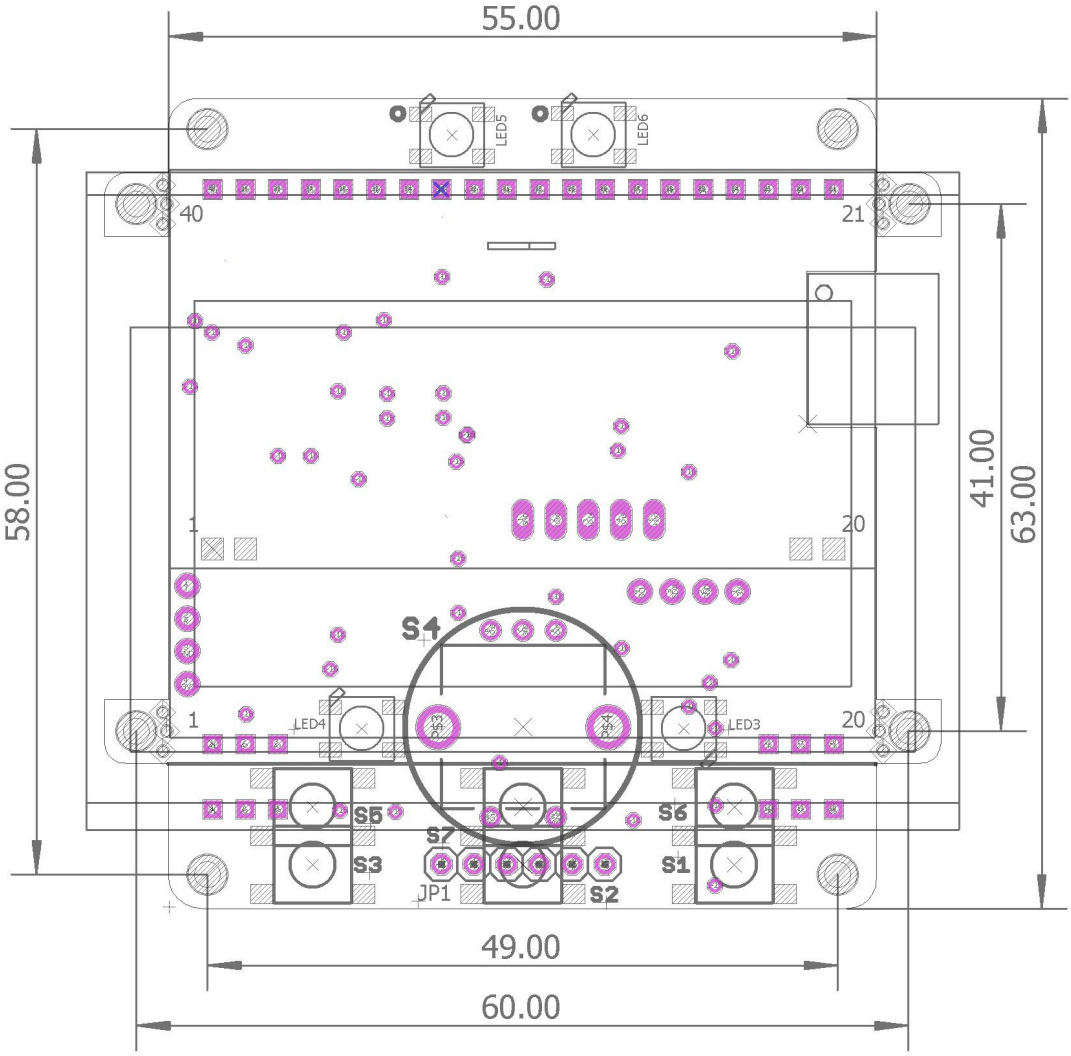
ESP32- DevKitDOGM128-POS with ESP32-WROOM-32-N8 (front side)



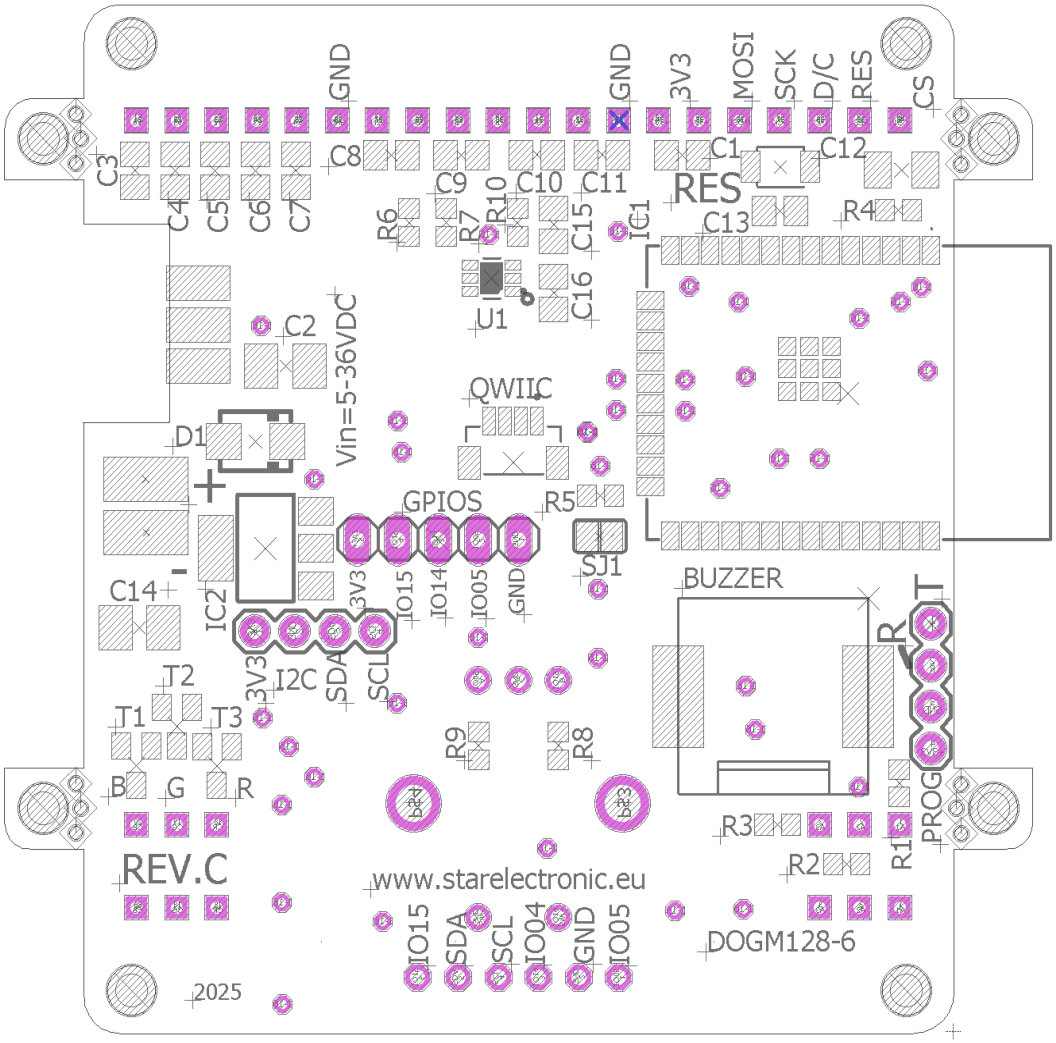
ESP32- DevKitDOGL128-BW with ESP32-WROOM-32-N8 (back side)

Key Component	Description
ESP32-WROOM-32E-N8	A module with ESP32 at its core. For more information, see ESP32-WROOM-32E-N8 Datasheet.
EA-DOGL128/TFT24/TFT28	LCD or TFT display
RES	Reset button (EN input of ESP32)
Neopixel WS2812B	WS2812B connected to GPIO32 of ESP32 module

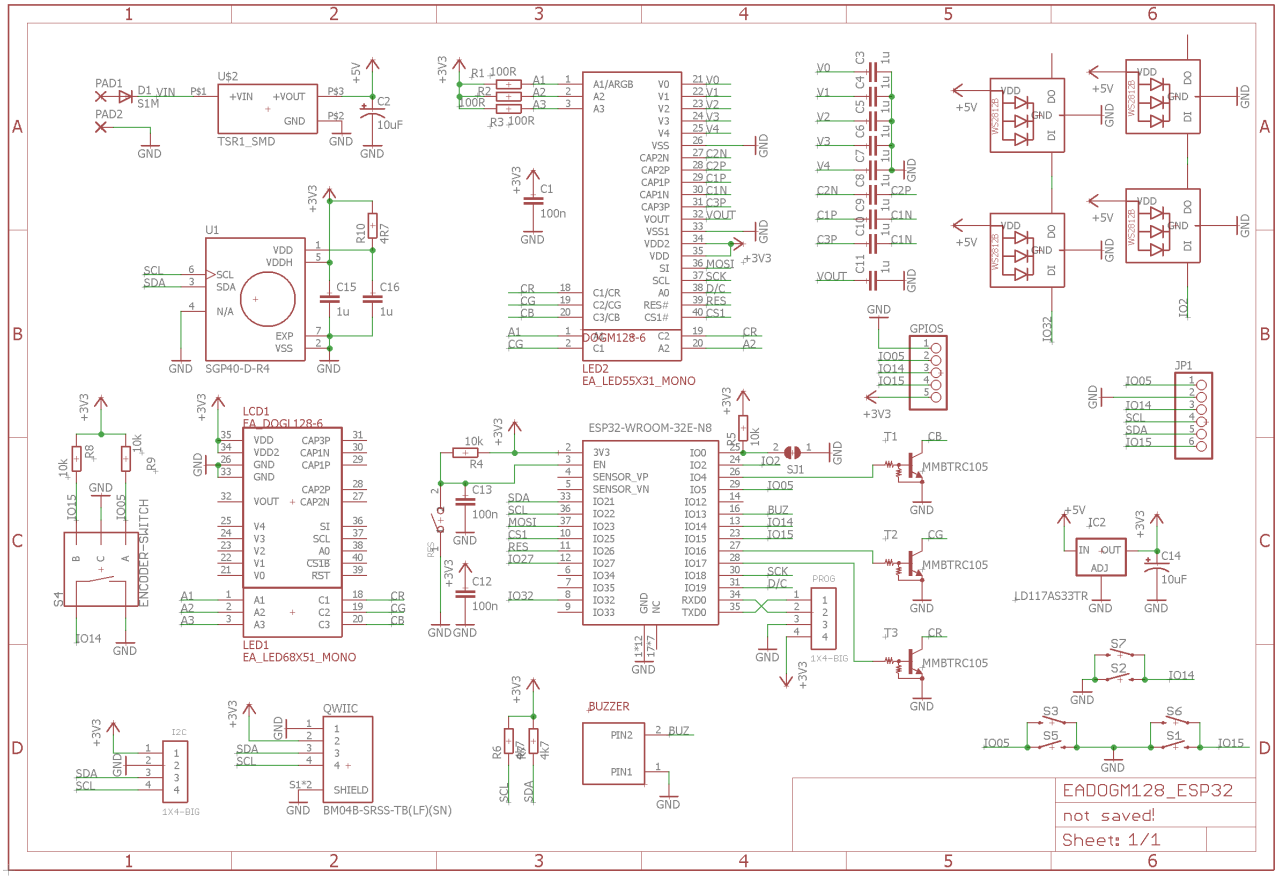
S1, S2, S3	Three push buttons for general use (connected to GPIO15, 14 and 05 respectively)
GPIOs	Some of the pins on the ESP module are broken out to the pin headers on the board. GPIO05, 14, 15, 22(SCL), 21(SDA) are broken out for further use.
BUZZER	Buzzer connected to GPIO13



ESP32-DevKitDOGL128-XX dimensions and component placement (TOP wiev)



ESP32-DevKitDOGL128-XX component placement (BOTTOM wiev)



ESP32- DevKitDOGL128-XX schematic

First steps - WiFi provisioning

The ESP32-DEVKITDOGL128-XX come with ESPHome (https://esphome.io/) firmware uploaded. That means, you can reach the board and upload your own configuration even using OTA. For first connection to WiFi network, power up the board and go to https://www.improv-wifi.com/ (make sure, you have bluetooth switched on on your notebook/smartphone) and click on „Connect device to Wi-Fi“ button:

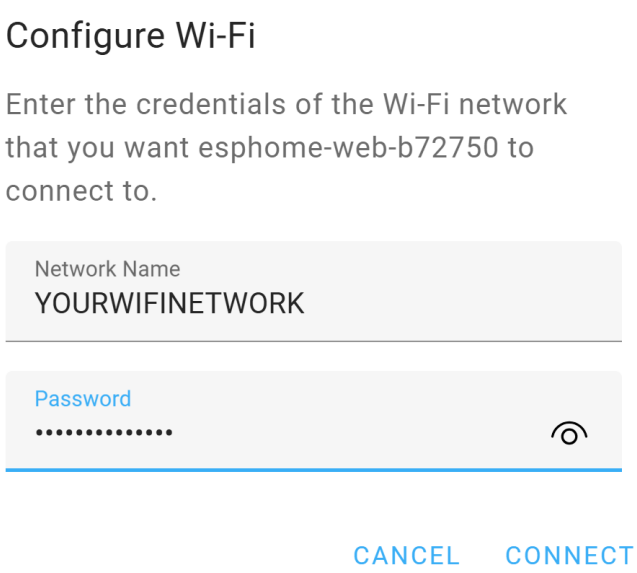
Improv via BLE

Connect device to Wi-Fi

You should see this picture:



Save the showed device name (**esphome-web-b72750 in this case**) for later use and click on „Pair“ button, then provide WiFi credentials (SSID and Password):

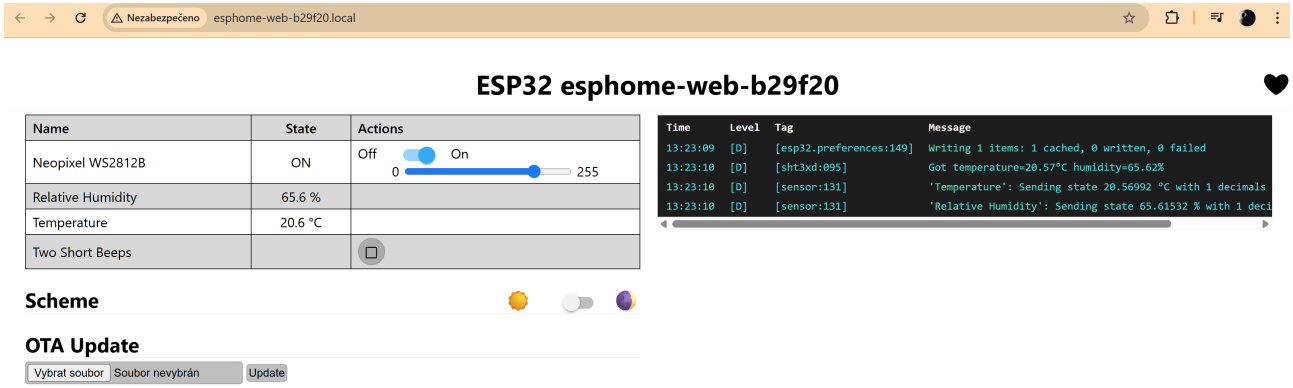


If the SSID and Password are correct, the board should connect to your WiFi network and show the correct time (Europe/Berlin timezone preconfigured). Now navigate to the esphome-web-xxxxxx.local in your browser (http://esphome-web-b72750.local in this case) or check the IP address of the device in the DHCP Client list of your WiFi router.

Warning

The Improv BLE software stack on the ESP32 consumes a significant amount of RAM on the device. Because of this fact the Improv BLE method is not available on boards equipped with TFT displays (ILI9341 controller). For these boards the captive_portal method is available - https://esphome.io/components/captive_portal/

Crashes are likely to occur if you include too many additional components in your device’s configuration. Memory-intensive components such as Voice Assistant and other audio components are most likely to cause issues.

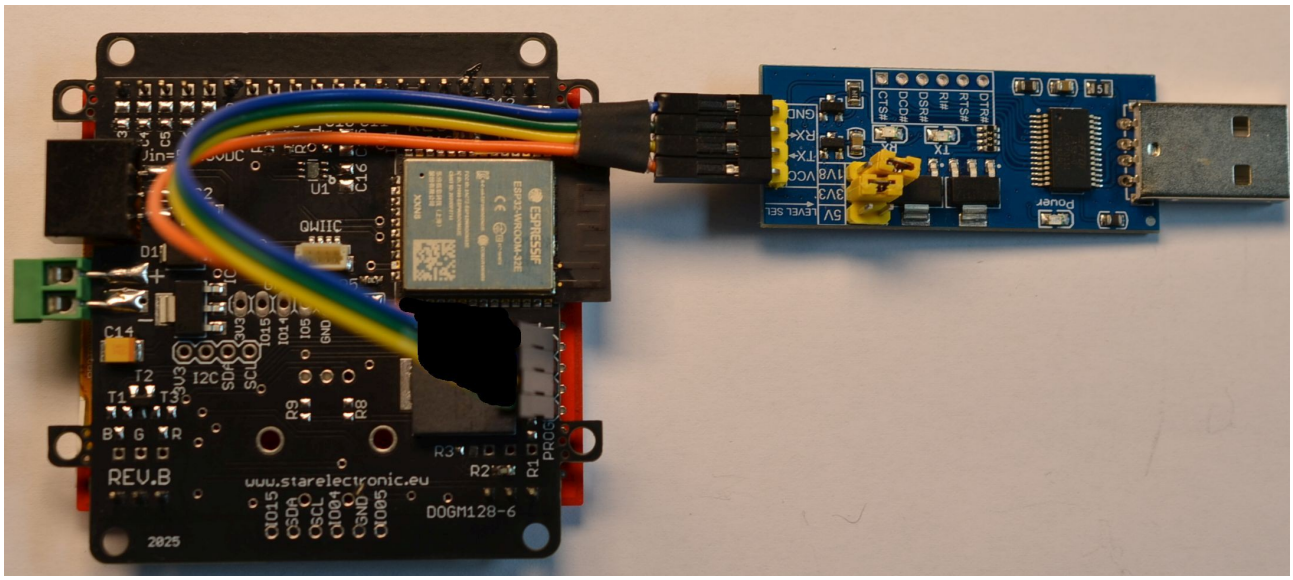


ESP32´s (ESPHome) internal webserver generated website with RH/T sensor connected

Now you can switch on and off remotely the LED and adjust its brightness using the slide bar. The embedded buzzer plays two short beeps if you click on the corresponding checkbox.

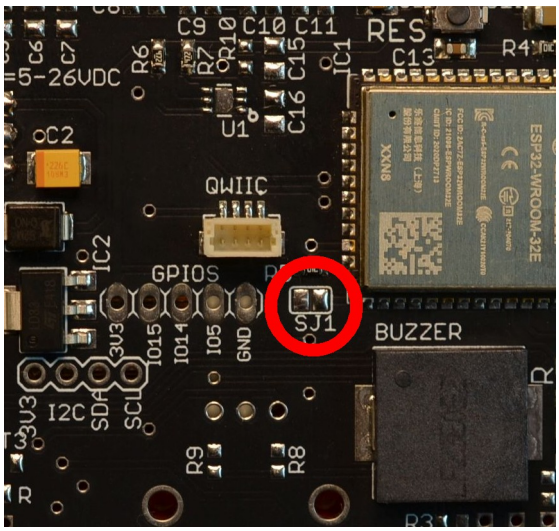
Board programming and troubleshooting

Normally the compiled .bin file could be uploaded wirelessly to the board using OTA, but sometimes there could be a bug in the uploaded code which prevents following uploads. For this case there is a „PROG“ interface available on the board, which is connected directly to RX and TX pins of ESP32 and you can use this interface to upload or erase the code in ESP32´s internal Flash memory. The best way is to use a USB/UART converter (must be 3,3V!):



ESP32-DEVKITTFT24 with connected USB/UART converter with FT232R chip

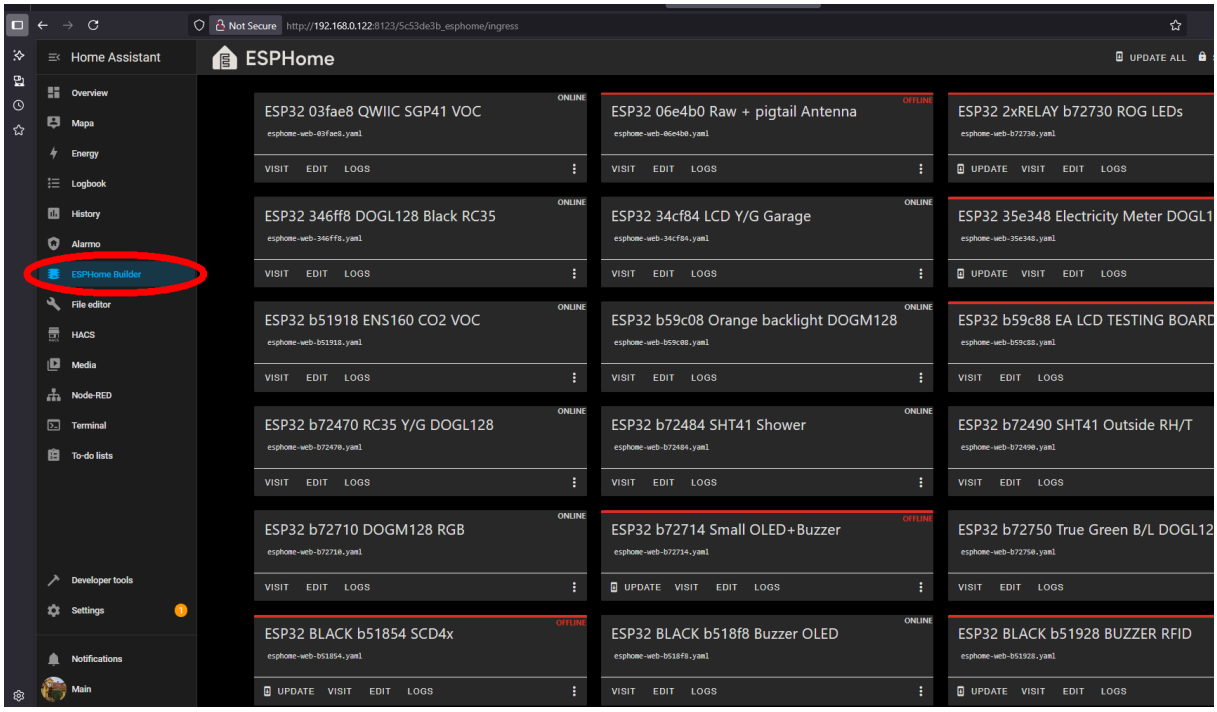
Before code upload short SJ1 solder joint (GPIO0 pin of ESP32). After code upload, remove the solder joint in order to run the uploaded code:



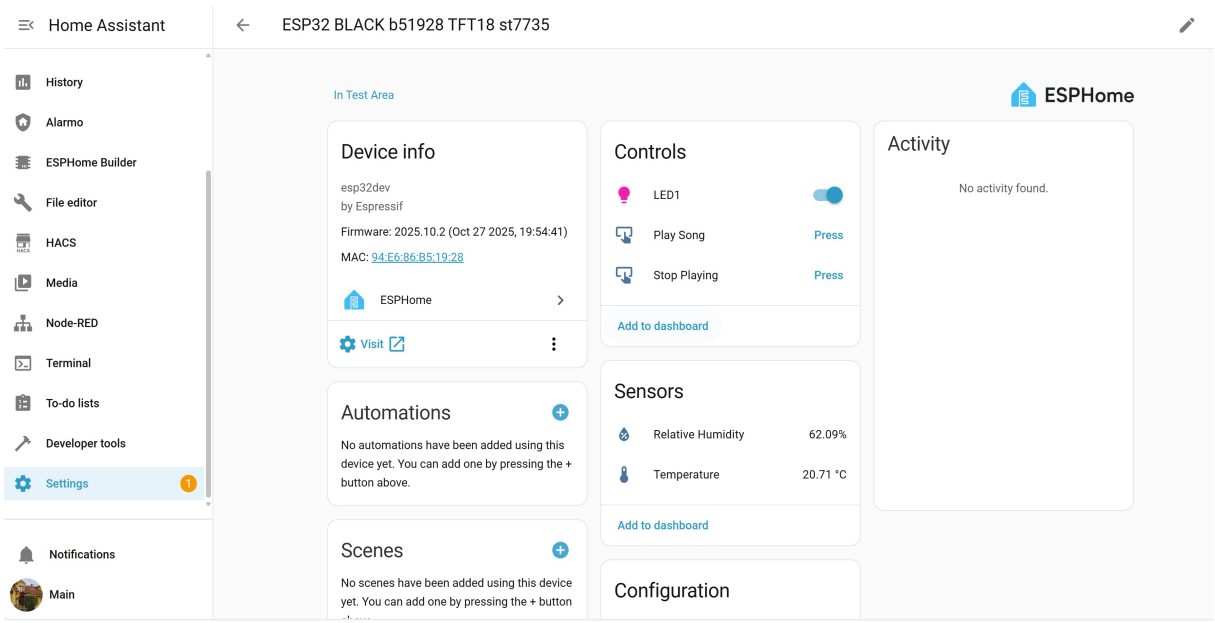
SJ1 solder joint for code upload using seriál programming

Home Assistant integration

This device could be easily integrated into your Home Assistant simply by adding it under ESPHome Integration. After connecting to your home network go to your ESPHome menu from Home Assistant:



Picture 3. Home Assistant integration



Picture 5. HA – Settings - Devices & Services – Integrations – ESPHome (API)

Example YAML file for using with ESPHome

```
esphome:
  name: esphome-web-b35f38
  friendly_name: ESP32 esphome-web-b35f38
  min_version: 2025.9.0
  name_add_mac_suffix: false
  compile_process_limit: 1

esp32:
  variant: esp32
  flash_size: 8MB
  framework:
    type: arduino

# Enable logging
logger:

# Enable Home Assistant API
api:

# Allow Over-The-Air updates
ota:
- platform: esphome
- platform: web_server

wifi:
  ap: {}

# Enable Improv over BLE
esp32_improv:
  authorizer: none
```

```yaml
# Allow internal web server
web_server:

external_components:
  - source:
      type: local
      path: my_components

output:
  - platform: ledc
    pin: GPIO13
    id: gpio_buzzer

rtttl:
  - output: gpio_buzzer

button:
  - platform: template
    name: "Two Short Beeps"
    on_press:
      then:
        - rtttl.play: 'two_short:d=4,o=5,b=100:16e6,16e6'
  - platform: template
    name: "Play Alarm"
    on_press:
      then:
        - rtttl.play: 'siren:d=8,o=5,b=100:d,e,d,e,d,e,d,e'

light:
  - platform: neopixelbus
    type: GRB
    variant: WS2811
    pin: GPIO32
    num_leds: 2
    name: "NeoPixel LED"

font:
  - file: "gfonts://Archivo Black"
    id: arialbold
    size: 40

  - file: "arial.ttf"
    id: arial
    size: 24

  - file: "Arial_Bold.ttf"
    id: arialsmall
    size: 14

  - file: "Arial_Bold.ttf"
    id: arialmini
```

```
    size: 9

  - file: "Arial_Bold.ttf"
    id: arialmedium
    size: 24

# RGB backlight
switch:
  - platform: gpio
    pin: GPIO17
    id: red
    name: "RED BL"
    restore_mode: RESTORE_DEFAULT_ON
  - platform: gpio
    pin: GPIO16
    id: green
    name: "GREEN BL"
    restore_mode: RESTORE_DEFAULT_ON
  - platform: gpio
    pin: GPIO04
    id: blue
    name: "BLUE BL"
    restore_mode: RESTORE_DEFAULT_ON

sensor:
  - platform: sht4x
    temperature:
      name: "Temperature °C"
      id: sht4x_temperature
    humidity:
      name: "Relative Humidity %"
      id: sht4x_humidity
    update_interval: 10s

time:
  - platform: sntp
    id: sntp_time
    timezone: Europe/Berlin
    servers:
     - 0.pool.ntp.org
     - 1.pool.ntp.org
     - 2.pool.ntp.org

i2c:
  sda: GPIO21
  scl: GPIO22
  scan: true
  id: bus_a

# Example minimal configuration entry
spi:
```

```
  clk_pin: GPIO18
  mosi_pin: GPIO23

# Configuration entry for DOGM128-6 LCD
display:
  - platform: st7567_spi_DOGL128
    dc_pin: GPIO19
    cs_pin: GPIO25
    reset_pin: GPIO26
    id: DOGM128
    lambda: |-
      // Print time in HH:MM format
      it.strftime(62, 48, id(arialbold), TextAlign::CENTER, "%H:%M",   id(sntp_time).now());
      it.printf(4, -4, id(arialmedium), TextAlign::TOP_LEFT, "%.1f°", id(sht4x_temperature).state);
      it.printf(77, -4, id(arialmedium), TextAlign::TOP_LEFT, "%.0f%%", id(sht4x_humidity).state);
      it.printf(0, 20, id(arialmini), TextAlign::TOP_LEFT, "TEMPERATURE");
      it.printf(120, 20, id(arialmini), TextAlign::TOP_RIGHT, "HUMIDITY");
      // Draw a line from [0,20] to [128,20]
      it.line(0, 31, 128, 31);
      it.line(70, 0, 70, 31);
      id(DOGM128).set_contrast(40);
```

Example YAML file with TFT2,4“ ILI9341 using in Home Assistant

```
esphome:

  name: esphome-web-b700e8
  friendly_name: ESP32 2,4" TFT b700e8 Portrait
  min_version: 2025.5.0
  name_add_mac_suffix: false
  compile_process_limit: 1

esp32:
  board: esp32dev
  framework:
    type: esp-idf

# Enable logging
logger:

wifi:
  # Set up a wifi access point
  ap:
    ssid: "Esphome-Web-b700e8"
    password: ""

# Enable Home Assistant API
api:

# Allow Over-The-Air updates
```

```
ota:
  - platform: esphome
  - platform: web_server

# To have a "next url" for improv serial
web_server:

captive_portal:

font:
  - file: "Arial_Bold.ttf"
    id: arialbold
    size: 96

  - file: "arial.ttf"
    id: arial
    size: 60

  - file: "Arial_Bold.ttf"
    id: arialsmall
    size: 30

  - file: "Arial_Bold.ttf"
    id: arialmini
    size: 20

  - file: "Arial_Bold.ttf"
    id: arialmedium
    size: 80

color:
  - id: RED
    red: 100%
    green: 3%
    blue: 5%
  - id: GREEN
    red: 0%
    green: 100%
    blue: 0%
  - id: BLUE
    red: 0%
    green: 0%
    blue: 100%
  - id: ORANGE
    red: 100%
    green: 73%
    blue: 0%
  - id: YELLOWGREEN
    red: 52%
    green: 99%
    blue: 0%
```

```
  - id: PINK
    red: 100%
    green: 0%
    blue: 66%
  - id: BLACK
    red: 0%
    green: 0%
    blue: 0%

sensor:
  - platform: homeassistant
    id: LOCAL_humi
    entity_id: sensor.esphome_web_b72710_rh
    internal: false
  - platform: homeassistant
    id: AIR_temp
    entity_id: sensor.ems_esp_thermostat_inttemp1
    internal: false
  - platform: homeassistant
    id: AIR_humidity
    entity_id: sensor.esphome_web_b72490_relative_humidity
    internal: false
  - platform: homeassistant
    id: OUT_temp
    entity_id: sensor.esphome_web_b72490_temperature
    internal: false
  - platform: homeassistant
    id: DHW_temp
    entity_id: sensor.ems_esp_temperaturesensor_28_808e_5a10_0000
    internal: false

binary_sensor:
  - platform: homeassistant
    id: DAYTIME
#    entity_id: binary_sensor.day_night
    entity_id: switch.day_night
    internal: false

time:
  - platform: sntp
    id: sntp_time
    timezone: Europe/Berlin
    servers:
     - 0.pool.ntp.org
     - 1.pool.ntp.org
     - 2.pool.ntp.org

# Example configuration entry - single controller
spi:
  clk_pin: GPIO18
  mosi_pin: GPIO23
```

```
# Example configuration for TFT2,4“ with ILI9341 controller and two periodically
# switching pages

display:
  - platform: ili9xxx
    model: ILI9341
    dc_pin: GPIO19
    reset_pin: GPIO26
    cs_pin: GPIO25
    color_palette: 8BIT
    color_order: BGR
    invert_colors: true
    show_test_card: false
    id: TFT24
    rotation: 0
    pages:
      - id: page1
        lambda: |-
          // Print time in HH:MM format
          it.strftime(120, 40, id(arialmedium), TextAlign::CENTER, "%H:%M", id(sntp_time).now());
          it.printf(240, 90, id(arialmedium), id(ORANGE), TextAlign::TOP_RIGHT, "%.1f°", id(OUT_temp).state);
          it.printf(5, 94, id(arialmini), TextAlign::TOP_LEFT, "O");
          it.printf(5, 118, id(arialmini), TextAlign::TOP_LEFT, "U");
          it.printf(5, 142, id(arialmini), TextAlign::TOP_LEFT, "T TMP");
          // Draw a line from [0,20] to [128,20]
          it.line(0, 90, 240, 90);
          it.line(0, 180, 240, 180);
          if (id(DAYTIME).state) {
            it.filled_circle(200, 215, 20);
            it.filled_circle(210, 205, 20, BLACK);
          } else {
            it.filled_circle(200, 215, 20, ORANGE);
          }
          // Print DHW temperature
          it.printf(5, 215, id(arialmini), id(RED), TextAlign::BASELINE_LEFT, "DHW");
          it.printf(5, 245, id(arialmini), id(RED), TextAlign::BASELINE_LEFT, "TEMPERATURE");
          if (id(DHW_temp).has_state()) {
          it.printf(240, 320, id(arialmedium), id(YELLOWGREEN), TextAlign::BASELINE_RIGHT, "%.1f°C",
id(DHW_temp).state);
          }
      - id: page2
        lambda: |-
          // Print time in HH:MM format
          it.strftime(120, 40, id(arialmedium), TextAlign::CENTER, "%H:%M", id(sntp_time).now());
          it.printf(240, 90, id(arialmedium), id(GREEN), TextAlign::TOP_RIGHT, "%.0f%% ",
id(AIR_humidity).state);
          it.printf(5, 94, id(arialmini), TextAlign::TOP_LEFT, "O");
          it.printf(5, 118, id(arialmini), TextAlign::TOP_LEFT, "U");
          it.printf(5, 142, id(arialmini), TextAlign::TOP_LEFT, "T  RH");
          // Draw a line from [0,20] to [128,20]
```

```
        it.line(0, 90, 240, 90);
        it.line(0, 180, 240, 180);
        if (id(DAYTIME).state) {
          it.filled_circle(200, 215, 20);
          it.filled_circle(210, 205, 20, BLACK);
        } else {
          it.filled_circle(200, 215, 20, ORANGE);
        }
        it.printf(5, 215, id(arialmini), id(RED), TextAlign::BASELINE_LEFT, "LIVING ROOM");
        it.printf(5, 245, id(arialmini), id(RED), TextAlign::BASELINE_LEFT, "TEMPERATURE");
        it.printf(240, 320, id(arialmedium), id(ORANGE), TextAlign::BASELINE_RIGHT, "%.1f%°C",
id(AIR_temp).state);

interval:
  - interval: 10s
    then:
      - display.page.show_next: TFT24
      - component.update: TFT24
```

For technical support and other versions please contact info@starelectronic.eu