

ESP32-DevKit12C Getting Started Guide

This guide shows how to start using the ESP32-DevKit12C-RGB development board.

Overview

ESP32-DevKit12C-RGB is a Flash disc sized ESP32-based development board using a powerful and affordable ESP32-WROOM-32E-N8 module. Some of the I/O pins are broken out to the pin headers on both sides for easy interfacing. Developers can either connect sensors using a 2.54mm peripherals with jumper wires or mount ESP32-DevKit12C-RGB on a breadboard. This board is a great starting point for your Arduino, Platformio, ESP-IDF and Home Assistant projects.

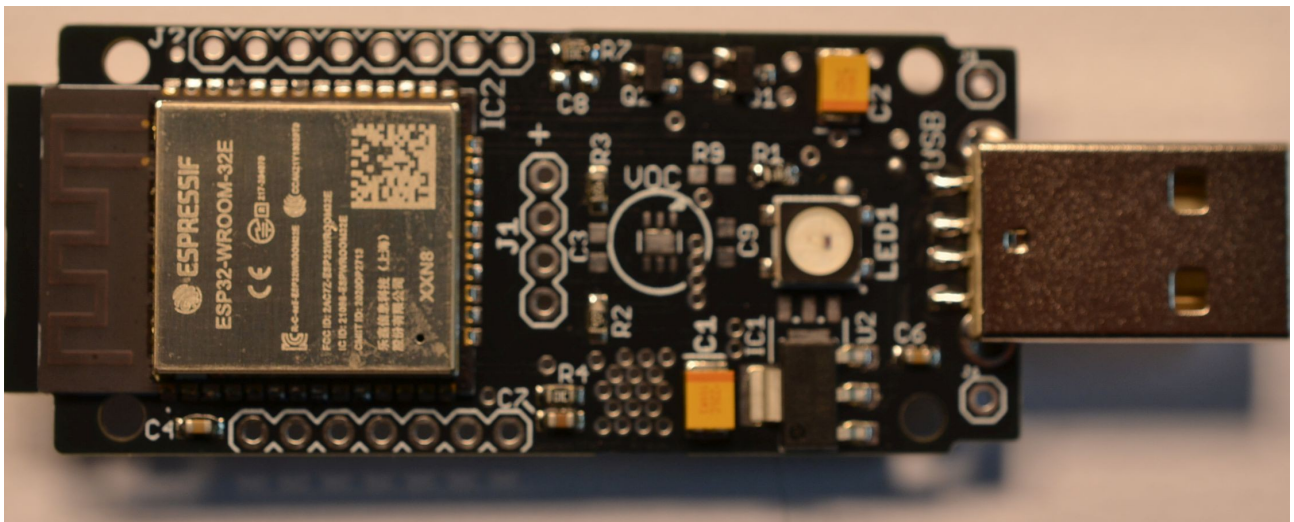
To cover a wide range of user requirements, the following versions of ESP32-DevKit12C are available:

ESP32-DevKit12C (Basic Version with WS2812B NeoPixel LED)

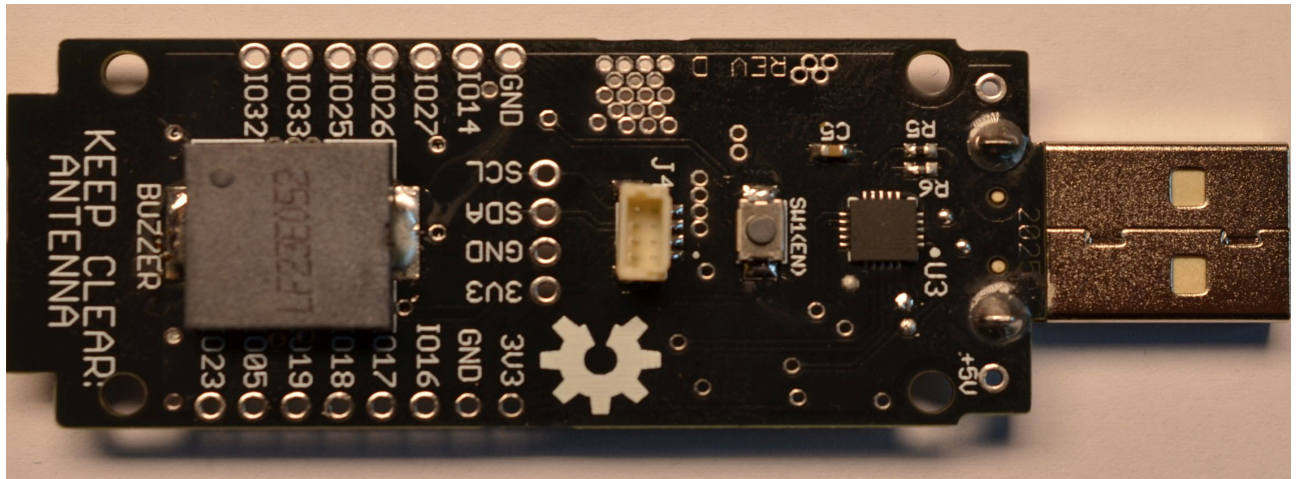
ESP32-DevKit12C-VOC with SGP40 VOC sensor SGP40

ESP32-DevKit12C-TFT with 1,8" 128x160pix TFT display (ST7735 driver)

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



ESP32-DEVKIT12C-RGB with ESP32-WROOM-32 module soldered



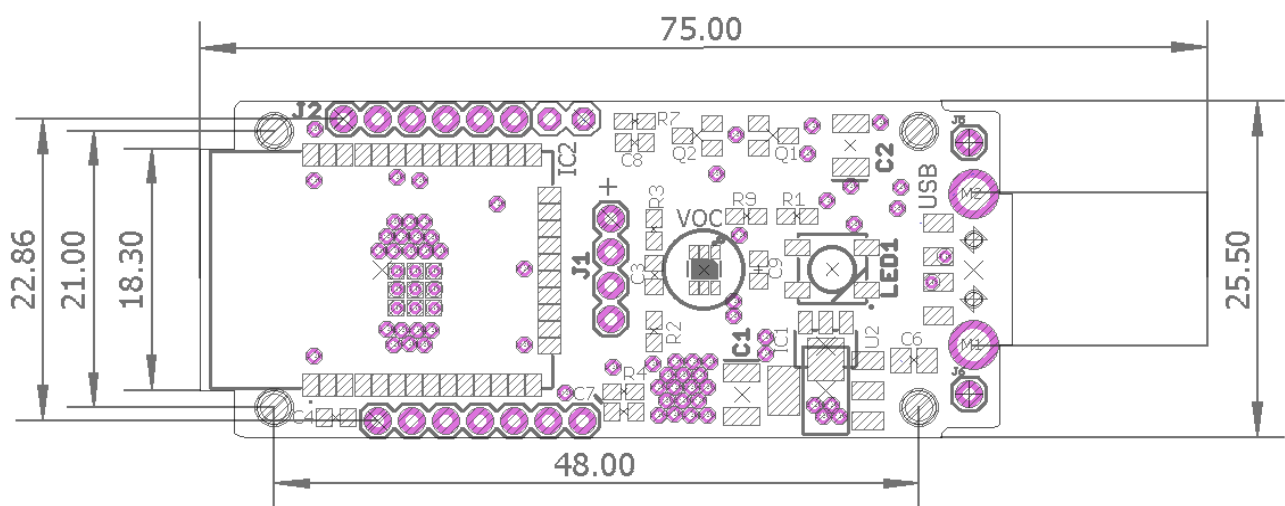
ESP32-DEVKIT12C-RGB bottom view



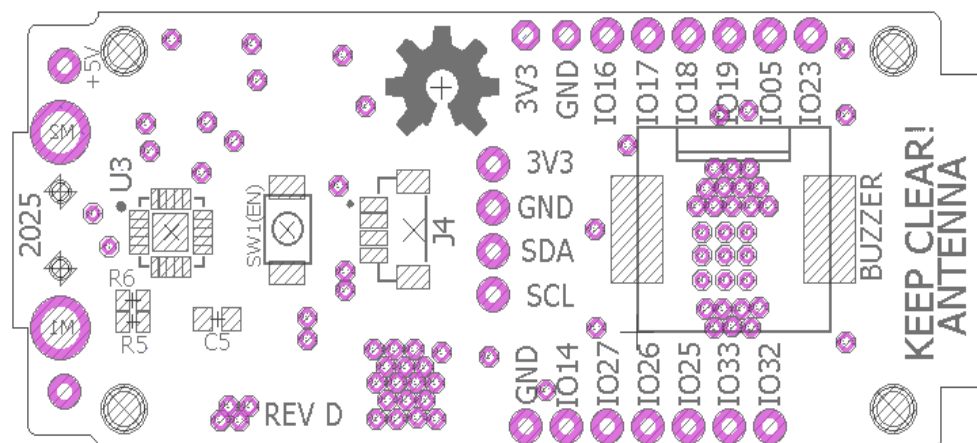
ESP32-DevKit12C-TFT with 1,8" 128x160pix TFT display (ST7735 driver)



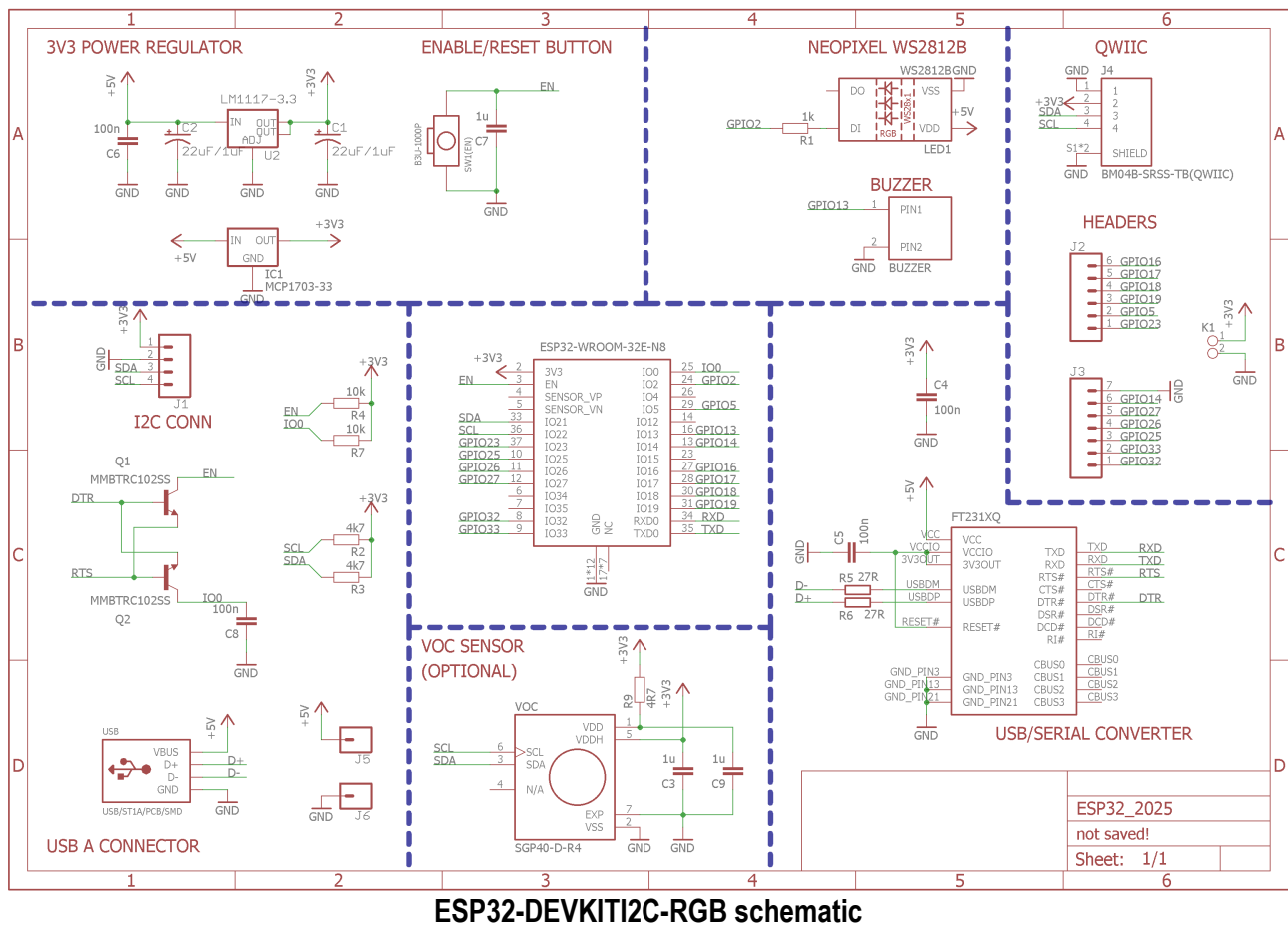
Key Component	Description
ESP32-WROOM-32E-N8	A module with ESP32 at its core. For more information, see ESP32-WROOM-32E Datasheet.
SW1(EN)	Reset button.
USB-to-UART Bridge	Single USB-UART bridge chip provides transfer rates of up to 3 Mbps. FT231XQ is used
USB Port Type A	USB interface. Power supply for the board as well as the communication interface between a computer and the ESP32-WROOM-32E module.
Neopixel WS2812B	WS2812B connected to GPIO02 of ESP32 module
GPIOs	Some of the pins on the ESP module are broken out to the pin headers on the board. GPIO05, 14, 16-19, 23, 25-27, 32, 33 are broken out for further use.
BUZZER	Buzzer connected to GPIO13



ESP32-DEVKITI2C dimensions and component placement (TOP wiew)



ESP32-DEVKITI2C component placement (BOTTOM view)



ESP32-DEVKITI2C-RGB schematic

WiFi provisioning

The ESP32-DEVKITI2C-XX come with ESPHome (<https://esphome.io/>) firmware uploaded. That means, you can reach the board and upload your own configuration 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 „Improv via BLE“ button:

Improv via BLE

Connect device to Wi-Fi

You should see the following picture:

www.improv-wifi.com chce provést spárování

 esphome-web-b72750

 Prohledávání...

Spárovat

Zrušit

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

Configure Wi-Fi

Enter the credentials of the Wi-Fi network that you want esphome-web-b72750 to connect to.

Network Name
YOURWIFINETWORK

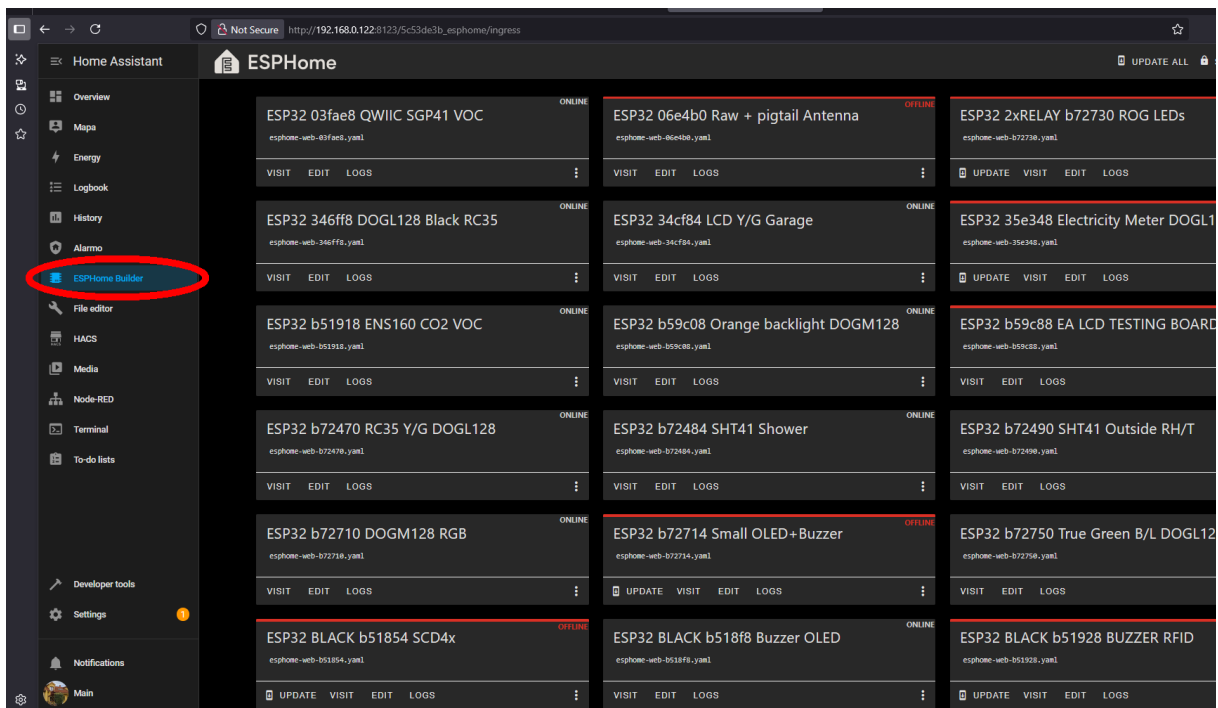
Password
.....

CANCEL CONNECT

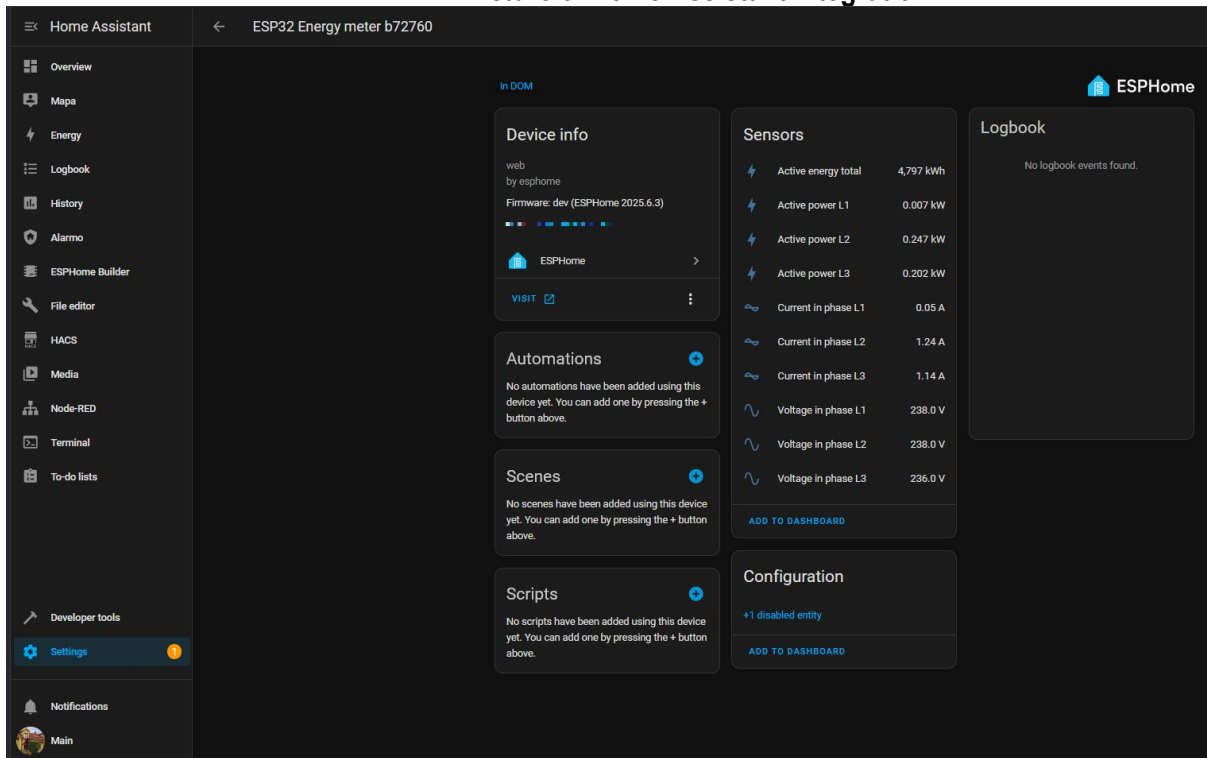
If the SSID and Password are correct, the board should connect to your WiFi network (it may take few dozens of seconds). Check the IP address of the device in the DHCP Client list of your WiFi router and navigate to this address using a browser. You can use also „<http://esphome-web-xxxxxx.local/>“ name in your browser (not supported by every browser) instead of the IP address. „xxxxxx“ is the part of the ESP32's module MAC address – „b72750“ in this case.

Home Assistant integration

This device could be easily integrated into your Home Assistant simply by adding it under ESPHome Integration. After connecting WOH-01 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 ESP32-DevKit12C-TFT with 1,8" 128x160pix TFT display (ST7735 driver):

esphome:

```
name: esphome-web-b72750
friendly_name: ESP32 BLACK b72750 TFT18 st7735
min_version: 2025.10.0
name_add_mac_suffix: false
```

esp32:

```

board: esp32dev
framework:
  type: arduino

# Enable logging
logger:

# Enable Home Assistant API
api:

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

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

# Allow provisioning Wi-Fi via serial
improv_serial:

# Sets up Bluetooth LE (Only on ESP32) to allow the user
# to provision wifi credentials to the device.
esp32_improv:
  authorizer: none

wifi:
  # Set up a wifi access point
  ap: {}

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

rtttl:
- output: gpio_buzzer

# Template button
button:
- platform: template
  name: "Play Song"
  on_press:
    then:
      - rtttl.play: 'PinkPanther:d=4, o=5, b=160:8d#, 8e, 2p, 8f#, 8g, 2p, 8d#, 8e, 16p, 8f#, 8g, 16p, 8c6, 8b, 16p, 8d#, 8e, 16p, 8b, 2a#, 2p, 16a, 16g, 16e, 16d, 2e'
      - platform: template
        name: "Stop Playing"
        on_press:
          then:

```

- rtttl.stop:

I2C interface definition

i2c:

sda: GPIO21

scl: GPIO22

scan: true

id: bus_a

RH/T sensor

sensor:

- platform: sht4x

temperature:

name: "Temperature"

id: sht_temp

humidity:

name: "Relative Humidity"

id: sht_humi

Neopixel LED definition

light:

- platform: neopixelbus

type: GRB

variant: WS2812

pin: GPIO02

num_leds: 1

name: "NeoPixel"

id: LED1

font:

- file: "Arial_Bold.ttf"

id: arialbold

size: 60

- 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: 50

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%

SNTP server definition

time:

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

Configuration entry for RC522 RFID reader

spi:

- clk_pin: GPIO18
- mosi_pin: GPIO23

Example minimal configuration entry

display:

- platform: st7735
 - model: "IN1TR_18BLACKTAB"
 - dc_pin: GPIO5
 - reset_pin: GPIO33
 - cs_pin: GPIO16

```

device_width: 128
device_height: 160
col_start: 0
row_start: 0
eight_bit_color: True
update_interval: 5s
id: TFT18
rotation: 180
pages:
- id: page1
  lambda: |-
    // Print time in HH:MM format
    it.strftime(0, 0, id(arialmedium), TextAlign::TOP_LEFT, "%H:%M", id(sntp_time).now());
    it.printf(0, 50, id(arialmedium), id(ORANGE), TextAlign::TOP_LEFT, "%.1f°", id(sht_temp).state);
    it.printf(62, 60, id(arialmini), TextAlign::CENTER, "TIME");
    it.printf(62, 100, id(arialmini), id(ORANGE), TextAlign::CENTER, "TEMP");
    it.line(0, 40, 128, 40);

- id: page2
  lambda: |-
    // Print time in HH:MM format
    it.strftime(0, 0, id(arialmedium), TextAlign::TOP_LEFT, "%H:%M", id(sntp_time).now());
    it.printf(0, 50, id(arialmedium), id(GREEN), TextAlign::TOP_LEFT, "%.0f%% ", id(sht_humi).state);
    it.printf(62, 60, id(arialmini), TextAlign::CENTER, "TIME");
    it.printf(62, 100, id(arialmini), id(GREEN), TextAlign::CENTER, "HUMI");
    it.line(0, 40, 128, 40);

interval:
- interval: 10s
  then:
    - display.page.show_next: TFT18
    - component.update: TFT18

```

For technical support please contact info@starelectronic.eu