Manuals+
Q & A | Deep Search | Upload

Waveshare ESP32-P4-NANO-KIT-A

Waveshare ESP32-P4-NANO Development Board User Manual

Model: ESP32-P4-NANO-KIT-A

1. Introduction

This manual provides detailed instructions for the Waveshare ESP32-P4-NANO High-Performance Development Board. It is designed to assist users in understanding the board's features, setting it up, and utilizing its capabilities effectively. The ESP32-P4-NANO is based on the ESP32-P4 chip, featuring RISC-V dual-core and single-core processors, and is equipped with various human-machine interfaces.
This specific kit, ESP32-P4-NANO-KIT-A, includes the ESP32-P4-NANO board, an RPi Camera (B), an 8Ω 2W speaker, and an FFC 15PIN cable, providing a comprehensive platform for development projects.

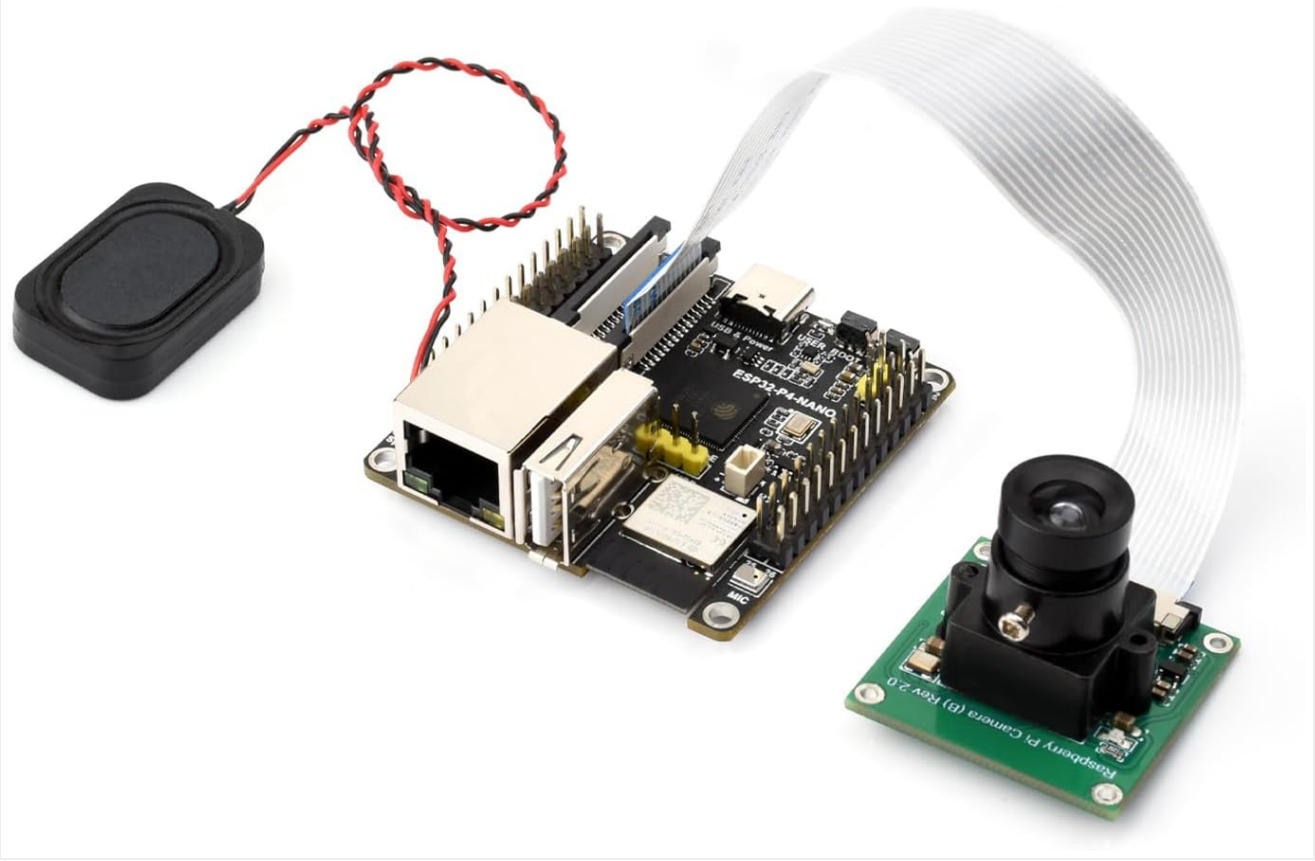


Image 1.1: The Waveshare ESP32-P4-NANO Development Board bundled with an RPi Camera and speaker.

2. Package Contents

Verify that all items listed below are present in your package. If any components are missing or damaged, please contact customer support.

- 1x ESP32-P4-NANO Development Board
- 1x RPi Camera (B)
- 1x 8Ω 2W Speaker
- 1x FFC 15PIN Cable

Image 2.1: Visual representation of the ESP32-P4-NANO-KIT-A package contents, including the development board, RPi Camera (B), 8Ω 2W speaker, and FFC 15PIN cable.

3. Key Features

The ESP32-P4-NANO Development Board offers a robust set of features for advanced embedded applications:

- **High-performance MCU:** Features RISC-V 32-bit dual-core and single-core processors.
- **Memory:** Includes 128 KB HP ROM, 16 KB LP ROM, 768 KB HP L2MEM, 32 KB LP SRAM, 8 KB TCM.
- **Image and Voice Processing:** Provides interfaces for JPEG Codec, Pixel Processing Accelerator, Image Signal Processor, and H264 encoder.
- **Integrated PSRAM and Flash:** 32MB PSRAM in the chip's package, with onboard 16MB Nor Flash.
- **Extensive Peripherals:** Supports MIPI-CSI, MIPI-DSI, USB 2.0 OTG, Ethernet, SDIO 3.0 TF card slot, microphone, speaker header, and RTC battery header.
- **GPIOs:** 2*2*13 GPIO headers with 28 x programmable GPIOs.
- **Security Features:** Incorporates Secure Boot, Flash Encryption, cryptographic accelerators, and TRNG. Hardware access protection mechanisms enable Access Permission Management and Privilege Separation.
- **Wireless Connectivity:** Onboard ESP32-C6-MINI module extends 2.4GHz Wi-Fi 6 and Bluetooth 5/BLE.

Image 3.1: Overview of the ESP32-P4-NANO's core features, including RISC-V MCU, memory, and connectivity options.

4. Hardware Overview

4.1 Board Components

The following diagram identifies the main components and interfaces on the ESP32-P4-NANO development board.

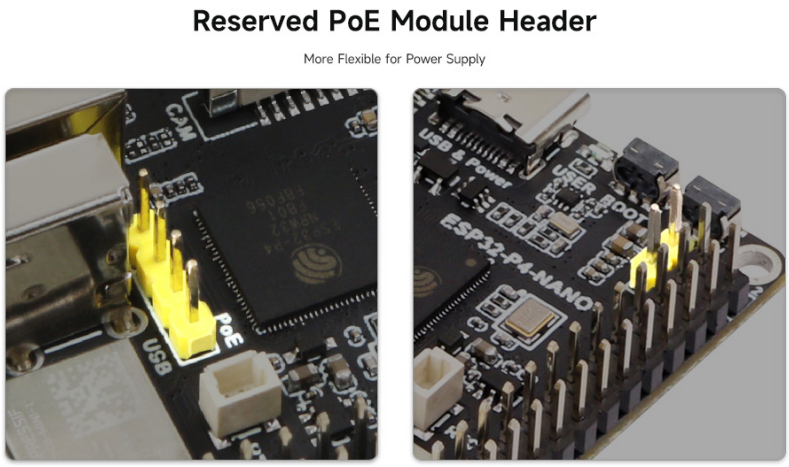


Image 4.1: Detailed view of the ESP32-P4-NANO board with numbered components and their descriptions.

1. ESP32-P4NRW32
2. ESP32-C6-MINI-1: SDIO interface protocol, extending Wi-Fi 6 and Bluetooth 5 for ESP32-P4-NANO
3. Display interface: MIPI 2-lane
4. Camera interface: MIPI 2-lane
5. USB Type-C connector: for power supply, program burning and debugging
6. USER-LED: power supply indicator
7. BOOT button: Press it when powering on or resetting to enter download mode
8. RESET button
9. PoE module / external power supply header: for connecting external 5V power supply or PoE module power supply
10. RTC battery header: for connecting rechargeable RTC battery (supports rechargeable RTC batteries only)
11. Onboard microphone
12. PoE module header
13. Type-A Port: USB OTG 2.0 High Speed port
14. RJ45 100M Ethernet port
15. Speaker header: MX1.25 2P connector, supports 8Ω 2W speaker
16. GPIO header
17. TF card slot: SDIO 3.0 interface protocol

4.2 Pin Definition

The pinout diagram illustrates the functionality of each GPIO pin on the development board, crucial for connecting external components and sensors.

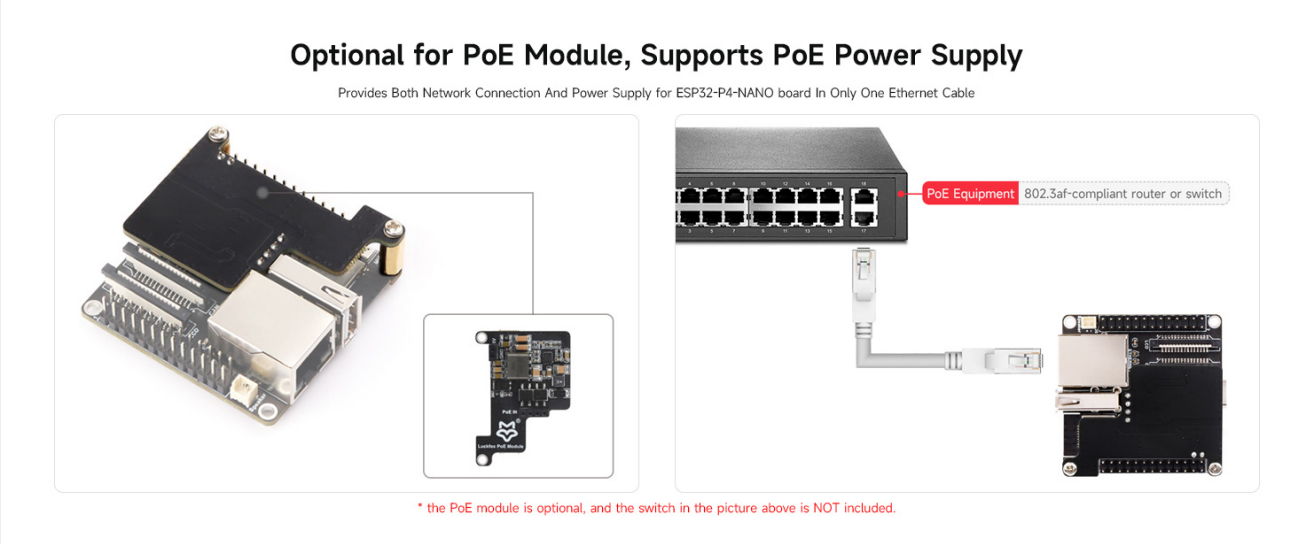


Image 4.2: Pin definition diagram showing power, GPIO, USB, UART, Touch Channel, ADC Channel, RTC Clock source, and ESP32-C6 GPIO assignments.

4.3 Outline Dimensions

Refer to the following diagrams for the physical dimensions of the ESP32-P4-NANO board, measured in millimeters.

Image 4.3: Top and bottom view with precise measurements of the ESP32-P4-NANO development board.

5. Setup Instructions

This section guides you through the initial setup of your ESP32-P4-NANO Development Board and its bundled accessories.

5.1 Connecting the RPi Camera (B)

1. Locate the MIPI-CSI camera interface on the ESP32-P4-NANO board (refer to Image 4.1, component 4).
2. Carefully insert one end of the FFC 15PIN cable into the camera module's connector, ensuring the contacts face the correct direction.
3. Insert the other end of the FFC 15PIN cable into the MIPI-CSI connector on the development board. Ensure a secure connection.

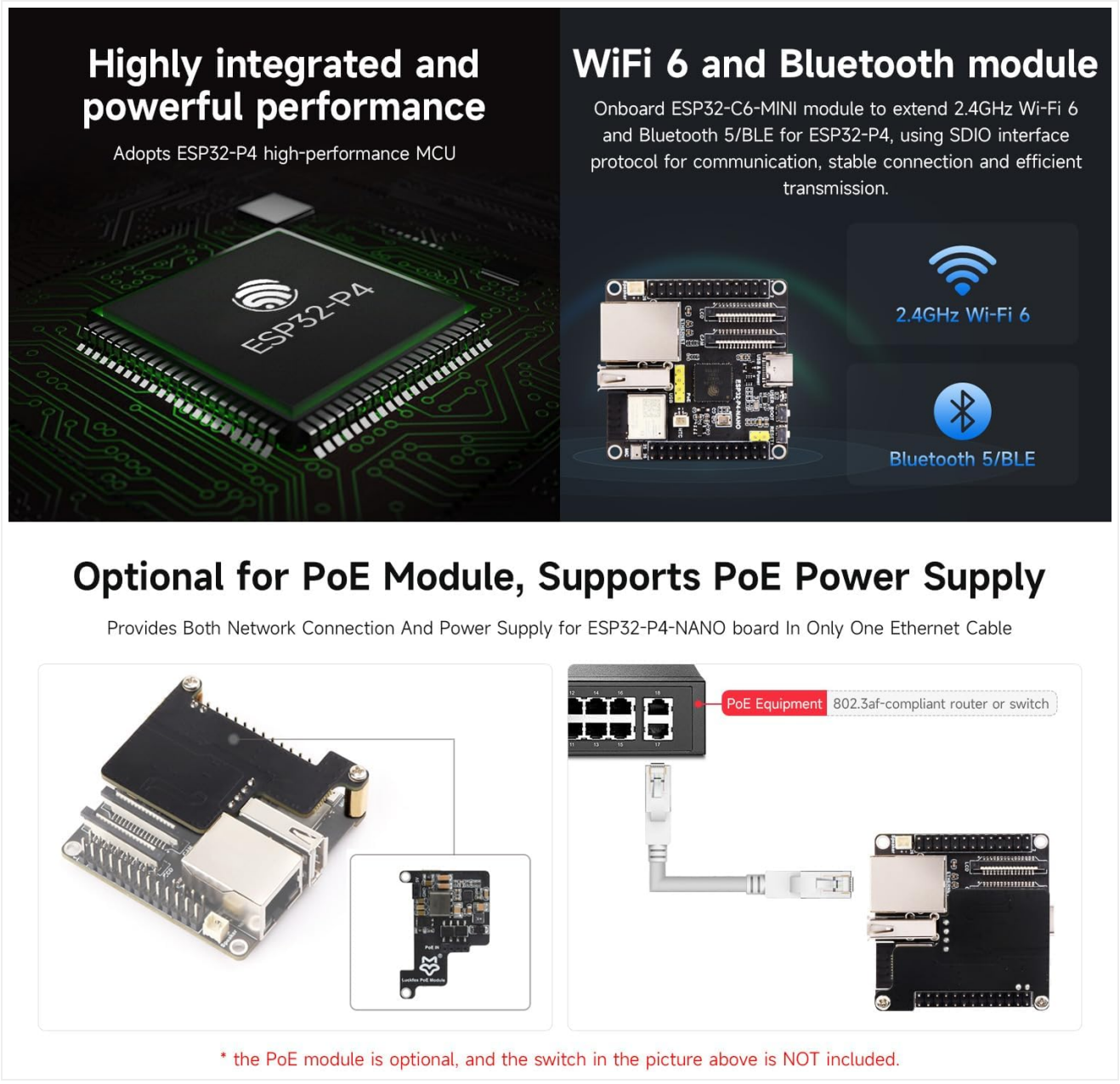


Image 5.1: The RPi Camera (B) module, compatible with Raspberry Pi series boards, featuring a 5MP OV5647 sensor.

5.2 Connecting the Speaker

1. Identify the speaker header (MX1.25 2P connector) on the development board (refer to Image 4.1, component 15).
2. Connect the 8Ω 2W speaker to this header. Ensure the polarity is correct if indicated.

5.3 Powering the Board

The board can be powered via the USB Type-C connector. Connect a compatible USB cable from your computer or a 5V power adapter to the USB Type-C port on the board (refer to Image 4.1, component 5).

5.4 Development Environment Setup

For programming and development, it is recommended to use the Espressif IoT Development Framework (ESP-IDF) or the Arduino IDE with ESP32 board support. Refer to the official Waveshare Wiki for detailed instructions on setting up your development environment and obtaining example code.

Image 5.2: Examples of how various kits, including KIT-A, can be connected, demonstrating the board with camera and speaker.

6. Operating Instructions

Once the board is set up and programmed, follow these general guidelines for operation.

6.1 Basic Operation

- **Power On:** Connect the USB Type-C cable to a power source. The USER-LED (component 6) should illuminate, indicating power.
- **Program Upload:** Use your chosen IDE (ESP-IDF or Arduino) to compile and upload your code to the board via the USB Type-C port.
- **Resetting the Board:** Press the RESET button (component 8) to restart the loaded program.
- **Download Mode:** To enter download mode for flashing new firmware, press and hold the BOOT button (component 7) while pressing and releasing the RESET button, then release the BOOT button.

6.2 Utilizing Peripherals

- **Camera:** Access the RPi Camera (B) through the MIPI-CSI interface using appropriate libraries and code examples from the Waveshare Wiki or Espressif documentation.
- **Speaker:** Generate audio output via the speaker header. Ensure your code includes audio playback functionalities.
- **Networking:** Utilize the 100Mbps Ethernet port (component 14) for wired network connectivity or the integrated Wi-Fi 6 and Bluetooth 5/BLE for wireless communication.
- **Storage:** Insert a TF card into the SDIO 3.0 TF card slot (component 17) for external storage.
- **GPIOs:** Connect various sensors, actuators, and other modules to the programmable GPIO headers (component 16) as required by your project.

7. Maintenance

Proper maintenance ensures the longevity and reliable operation of your development board.

- **Handling:** Always handle the board by its edges to avoid touching sensitive components. Use anti-static precautions when working with the board.
- **Cleaning:** Keep the board clean and free from dust and debris. Use a soft, dry brush or compressed air for cleaning. Avoid liquid cleaners.
- **Storage:** Store the board in an anti-static bag in a dry, cool environment when not in use.
- **Power Supply:** Use only recommended 5V power supplies. Over-voltage can damage the board.
- **Firmware Updates:** Regularly check the Waveshare Wiki for firmware updates and security patches to ensure optimal performance and security.

8. Troubleshooting

This section addresses common issues you might encounter.

- **Board Not Powering On:**
  - Ensure the USB Type-C cable is securely connected to both the board and a functional power source.
  - Verify that the power source provides 5V.
  - Check if the USER-LED (component 6) illuminates.
- **Program Upload Failure:**
  - Confirm that the correct board and port are selected in your IDE.
  - Try entering download mode manually by holding the BOOT button during reset.
  - Ensure all necessary drivers are installed for the USB-to-serial converter.
- **Camera Interface Issues:**

- Some users have reported challenges with the camera interface functionality. Waveshare acknowledges this and suggests monitoring Espressif's official resources for potential software fixes or updated examples.
- Ensure the FFC cable is correctly seated and not damaged.
- Verify that your code correctly initializes and accesses the MIPI-CSI interface.

- **No Wi-Fi/Bluetooth Connectivity:**
  - Check your code for correct Wi-Fi/Bluetooth initialization and credentials.
  - Ensure the ESP32-C6-MINI module is properly functioning.

For further assistance, consult the official Waveshare Wiki or contact their technical support.

9. Specifications

Feature	Detail
Brand	Waveshare
Model Name	ESP32-P4-NANO-KIT-A
Processor	ESP32-P4 (RISC-V 32-bit dual-core and single-core)
RAM	32MB PSRAM
Flash Memory	16MB Nor Flash
Wireless Connectivity	Wi-Fi 6 (802.11ax), Bluetooth 5/BLE (via ESP32-C6-MINI)
Ethernet	100Mbps RJ45
USB	USB 2.0 OTG (Type-A Port), USB Type-C (Power/Debug)
Camera Interface	MIPI-CSI 2-lane
Display Interface	MIPI-DSI 2-lane
Storage	SDIO 3.0 TF card slot
GPIOs	28 programmable GPIOs (2*2*13 headers)
Audio	Onboard microphone, Speaker header (8Ω 2W speaker included in kit)
Dimensions	50 x 50 mm (approximate board size)
Weight	~9.5 ounces (total item weight)

10. Support and Resources

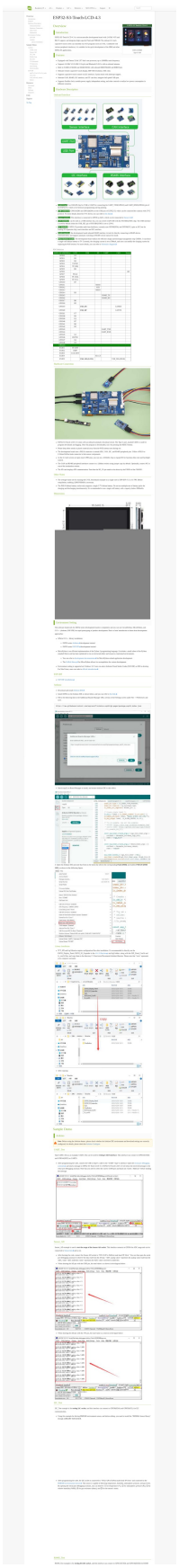
For the most up-to-date information, detailed tutorials, and example code, please refer to the official Waveshare Wiki. This resource is regularly updated and provides comprehensive documentation for all Waveshare products.

- **Official Waveshare Wiki:** https://www.waveshare.com/wiki (Please note: specific product page link may vary, search for "ESP32-P4-NANO")
- **Technical Support:** If you encounter issues not covered in this manual or the Wiki, please contact

Waveshare customer support through their official website.

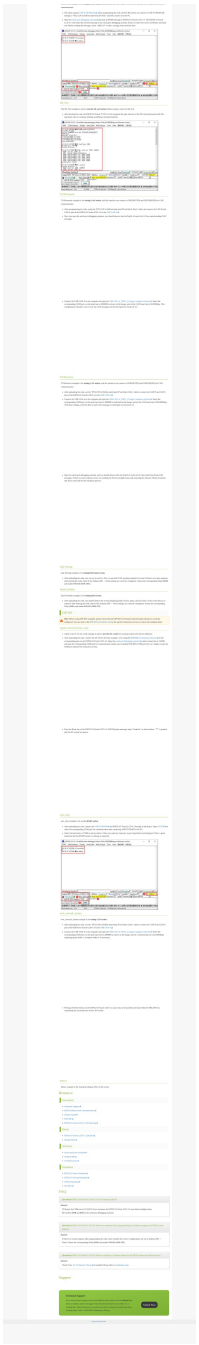
Information regarding warranty is typically provided with the product packaging or available on the manufacturer's official website.

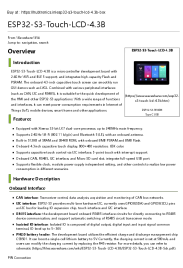
Related Documents - ESP32-P4-NANO-KIT-A



[Waveshare ESP32-S3-Touch-LCD-4.3 Development Board: Features & Guide]

Explore the Waveshare ESP32-S3-Touch-LCD-4.3, a powerful microcontroller development board featuring a 4.3-inch capacitive touch display, WiFi, BLE 5, and multiple interfaces like CAN, RS485, and I2C. Learn about its hardware, setup, and sample demos for HMI development.





[ESP32-S3-Touch-LCD-4.3B: Development Board Overview and Setup Guide]

Explore the ESP32-S3-Touch-LCD-4.3B, a powerful microcontroller development board from Waveshare. This guide covers its features, hardware description, interface details, and provides instructions for setting up the development environment using ESP-IDF and VSCode.

RGB-Matrix-P4-64x32

This product is a bare screen and needs to be displayed with the main control board such as Raspberry Pi, ESP32, and Arduino. For details, please refer to the list.

Overview

Introduction

This product is a 64 × 32 RGB LED matrix panel, with 2048 RGB LEDs onboard, 4mm pitch, supports Raspberry Pi and Arduino, etc. It provides supporting open-source demos and tutorials, a dubstle for makers or electronics Enthusiasts to start learning, or DIY secondary development into other desktop or wall-mounted display applications.

Features

- 2048 individual RGB LEDs, full-color display, adjustable brightness.
- 64 × 32 pixels, 4mm pitch, allows displaying text, colorful images, or animation.
- The dimension is 256 × 128mm, which is readerable and suitable for DIY desktop display or wall mount display.
- Onboard two HUB75 headers, one for controller data input, one for output, and chain support.
- Provides open-source development resources and tutorials, for use with Raspberry Pi, Arduino, and so on.

Specifications

DIMENSIONS	256mm × 128mm
PIXELS	64 × 32=2048 DOTS
PITCH	4mm
PIXEL FORM	SMD1turn
VIEWING ANGLE	120°
CONTROL TYPE	synchronization
DRIVING	1/16 scan
HEADER	HUB75
POWER SUPPLY	5V / 4A （With header input）
POWER	120W

Header Definition

PIN	DESCRIPTION	PIN	DESCRIPTION
VCC	5V power input	GND	Ground
R1	R higher bit data	R2	R lower bit data
G1	G higher bit data	G2	G lower bit data
B1	B higher bit data	B2	B lower bit data
A	A line selection	B	B line selection
C	C line selection	D	D line selection
E	E line selection	CLK	clock input
LAT/STB	latch pin	OE	output enable

Note: The power supply ports (VCC and GND) of the display screen are 5V power supply, do not connect to other power supply voltages, so as not to burn the display screen.

Usage Scenarios

DIY Maker Desktop Or Wall Mount Display, Signboard, Environment Monitor...

Raspberry Pi Instructions

Please refer to the github information: Demo, wiring reference

User Guides for Arduino Mega

Hardware Connection

Prepare materials

- RGB-Matrix-P3-64x32
- Arduino Mega (please purchase it separately)

Hardware Connection Diagram

Software Settings

- Download example
- After connecting the wires according to the hardware connection diagram, the software settings are as follows:

RGB text display

The effect of running the instance is shown in the following figure:

【Function Description】

- UI
  - Can display icons
  - Can display text content such as text and numbers

Working with ESP32

Preparation

- RGB-Matrix-P4-64x32
- 5V/4A或以上5V5 (not included)

Hardware Connection

Software Setting

- Download the demo
- Copy the folder under libraries in AuroraDemo to the libraries in the installation directory of the Arduino IDE
- After connecting the wires according to the hardware connection diagram, the software settings are as follows:

- Note: If you don't have the corresponding board, you can operate as below:

```
//Link to file in ACE board manager addition >> paste:  https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json >> OK
```

Perform

```
#include<Wapper.h>//Display basic shapes
#include<Fonts.h>//Displays various fonts
#include<Pictures.h>//Display pictures
#include<Scroll.h>//Display scrolling
#include<Effects.h>//Display various animation effects
#Check to see if the device is connected to the ACE board successfully
#Check to see if the device is connected to the ACE board successfully
```

Working with Pico

Hardware Connection

Preparation

- RGB-Matrix-P4-64x32
- Raspberry Pi Pico (It must be purchased separately. If not, it is recommended to buy one with soldered pin headers, which is convenient for wiring.)

Diagram

Software Setting

- Download the device
- After connecting the wire according to the diagram, you can set the software:

environment building

Performance

- After setting up the CircuitPython environment, you can copy all the contents of the CircuitPython directory in the downloaded Pico example to the recognized USB flash drive, and you can run the example (there are 16 demos in this code).
- The PSD folder is a file in ".psd" format, which can be used to modify the text image that needs to be moved. The modified image needs to be saved as "FloralBed" and the format needs to be converted to "BMP" format.

【Function】

- Display text
- Set scrolling effect

Resources

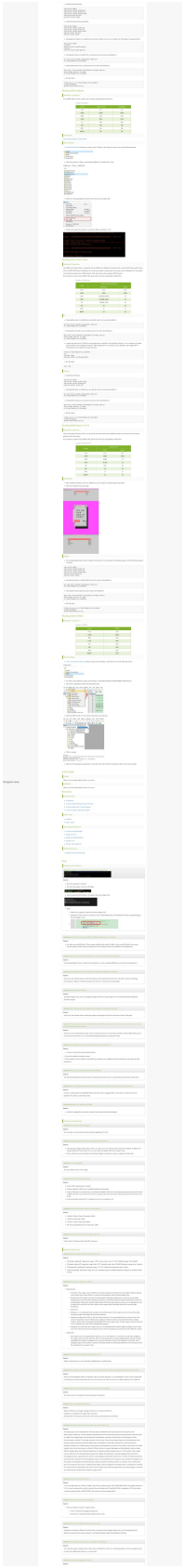
- Demo Sample
- 2D Drawing
- windows-arduino-1.8.15

Support

If you require technical support, please go to the page and open a ticket.

[Waveshare RGB-Matrix-P4-64x32: 64x32 LED Matrix Panel for DIY Projects]

Discover the Waveshare RGB-Matrix-P4-64x32, a 64x32 RGB LED matrix panel with 4mm pitch, perfect for DIY desktop or wall-mounted displays. It is compatible with Raspberry Pi, Arduino, and ESP32, offering extensive open-source resources.



[Waveshare 2.13inch e-Paper HAT (B) User Manual and Technical Guide]

Comprehensive guide for the Waveshare 2.13inch e-Paper HAT (B), covering hardware connections, software setup, programming principles, and troubleshooting for Raspberry Pi, Arduino, Jetson Nano, and STM32.

[Waveshare 7.5-inch E-Paper HAT User Manual and Guide]

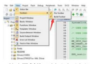
This comprehensive user manual provides detailed information on the Waveshare 7.5-inch E-Paper HAT (V1/V2), an 800x480 resolution display module utilizing Microencapsulated Electrophoretic Display technology. It covers hardware connections, SPI communication, working principles, and integration with Raspberry Pi, Arduino, Jetson Nano, Sunrise X3 Pi, STM32, ESP32, and ESP8266. Essential precautions, resources, and FAQs are included for optimal use.

DIN	PA7
CLK	PA5
CS	PA4
DC	PA2
RST	PA1
BUSY	PA3
PWR	PA6

Run The Program

- Click to download the demo, and then unzip it into the E-Paper_code directory to see the following content.

- Use Keil to open epd-demo.uvprojx in the E-Paper_code\STM32\STM32-F103ZET6\MDK-ARM directory
- Open Keil's compilation toolbar (usually already open).



- Select the EPD_7in5_V2_test at the location shown in the picture.

- Click to compile.

- Make sure the appropriate programmer is connected, then click LOAD to download the demo to the microcontroller.

ESP32/8266

ESP32

There is a lot of content, please click the URL below to view: https://www.waveshare.com/wiki/E-Paper_ESP32_Driver_Board

ESP8266

There is a lot of content, please click the URL below to view: https://www.waveshare.com/wiki/E-Paper_ESP8266_Driver_Board

Resources

Documentation

- 7.5inch e-Paper V2 Specification (current version)
- 7.5inch-e-paper V1 Specification
- Schematic

Demo code

- Demo (E-Paper_code.zip)
- Github

Development Resources

- E-Paper Floyd-Steinberg
- Image2Lcd.7z
- Image2Lcd Image Modulo
- Zimo221.7z
- E-Paper_API_Analysis

Related Resources

- Raspberry Pi Documentation

FAQ

Question about Software

Question: The Raspberry Pi runs the Python program and the following occurs?

Answer:

- Enter the command: ls /dev/spi*
- The result may appear as shown in the figure.

- This is because the SPI interface is occupied in the /boot/config.txt file.

- Steps:
  - Delete the occupation of spi0-0 in the /boot/config.txt file.
  - Modify the location shown in the picture in the /e-Paper/RaspberryPi_JetsonNano/lib/waveshare_epd/epdconfig.py file and change it to 0,1.
  - 

Question: stm32 drives the e-paper screen, the MDK compilation display space is not enough?

Answer:

Our demo uses stm32f103zet6. If the customer modifies other models in MDK, such as stm32f103rbt6, the ram space becomes smaller, and the stack size and heap size in the startup file need to be modified on the original basis.

Question: After multiple positions are brushed, the font is lighter after brushing several times?

Answer:

In this case, the customer needs to reduce the position of the round brush and clear the screen after 5 rounds of brushing (increasing the voltage of VCOM can improve the color, but it will increase the afterimage).

Question: When the e-paper screen is in deep sleep mode, there will be a problem that the screen refresh will not be clean when it wakes up for the first time. How can I solve it?

Answer:

The process of re-awakening the e-ink screen is actually the process of re-powering. Therefore, when the EPD wakes up, the screen must be cleared first, so as to avoid the afterimage phenomenon to the greatest extent.

Question: When testing the demo, it has been stuck in e-Paper busy?

Answer:

It may be caused by the unsuccessful SPI driver.
1. First check whether the wiring is correct.
2. Check whether the spi is turned on and whether the parameters are configured correctly (spi baud rate, spi mode, and other parameters).

Question: Why can't the image be displayed after full brushing?

Answer:

The full refresh initialization function needs to be added when the ink screen is switched from partial refresh to full refresh.

Question: Why is the printing information normal when running a Python program, but the ink screen does not respond?

Answer:

It may be a demo based on the BCM2835 library that has run the C language before. At this time, you need to restart the Raspberry Pi and then run the Python demo.

Question: ImportError: No module named Image?

Answer:

Install the imaging library using the command sudo apt-get install python-imaging

Question: When to transmit Data Start Transmission 1 and when to use Data Start Transmission 2. In KW mode, what is "OLD" data and what is "NEW" data. ?

Answer:

If it is a three-color screen, when you need to transmit B/W data, use Data Start Transmission 1, when transmitting RED data, use Data Start Transmission 2, because the 2.13inch e-Paper HAT (D) can only display black and white, Therefore, when working in KW mode Data Start Transmission 1, the last data is sent, that is, the so-called old, and 0x00 is directly sent in the program. When Data Start Transmission 2, the data that needs to be refreshed, the so-called "NEW" is sent. data.

Question: The e-Paper display is too dark or too light?

Answer:

You can adjust the value of Vcom in the program to change the display contrast, and the screen improvement effect with local brushing is particularly obvious.

Question: e-Paper shows black border?

Answer:

The border display color can be set through the Border Waveform Control register or the VCOM AND DATA INTERVAL SETTING register.

Question: When the ink screen is in deep sleep mode, the first time the screen refreshes will be unclean. How can I solve it?

Answer:

The process of re-awakening the e-paper screen is actually the process of supplying the power again. Therefore, when the EPD wakes up, it must first clear the screen, so as to avoid the afterimage phenomenon to the greatest extent.

Question about Hardware

Question: Can Arduino 5V drive the ink screen?

Answer:

Yes, now there is a level conversion chip onboard, supporting a 5V drive.

Question: What should be paid attention to when designing the driver board?

Answer:

- The rated input voltage of the e-paper screen is 2.3~3.6V. If it is a 5V system, level conversion is required. In addition, the voltage should not be lower than 2.5V, so as not to affect the display effect of the e-paper screen.
- Device selection can use the model in the schematic diagram we provide or choose according to the data sheet.

Question: Can I use analog SPI?

Answer:

Yes, pay attention to the correct timing.

Question: Why is the BUSY pin always busy?

Answer:

- Check if SPI communication is normal.
- Confirm whether the BUSY pin is normally initialized to input mode.
- It may be that there is no normal reset, try to shorten the duration of the low level during reset (because the power-off switch is added to the drive circuit, the reset low level is too long, which will cause the drive board to power off and cause the reset to fail).
- If the busy function sends the 0x71 command, you can try to comment it out.

Question: What is the specification of the screen cable interface?

Answer:

- 1.64inch, 2.36inch, 3inch, 0.5mm pitch, 26Pin.
- 1.02inch, 0.5mm pitch, 30Pin.
- 4.37inch, 7.3inch, 0.5mm pitch, 50Pin.
- The rest (non-parallel ports) are 0.5mm pitch, 24Pin.

Question: What type of connector does the ink screen use?

Answer:

Cable socket 0.5-24pin rear-flip 2.0H (FPC connector).

Question about Screen

Question: What is the usage environment of the e-paper screen?

Answer:

- 【Working conditions】Temperature range: 0~50°C; Humidity range: 35%~65%RH.
- 【Storage conditions】Temperature range: below 30°C; Humidity range: below 55%RH; Maximum storage time: 6 months.
- 【Transportation conditions】Temperature range: -25~70°C; Maximum transportation time: 10 days.
- 【After unpacking】Temperature range: 20°C±5°C; Humidity range: 50±5%RH; Maximum storage time: Assemble within 72 hours.

Question: Precautions for E-paper screen refresh

Answer:

- Refresh mode
  - Full refresh: The e-paper screen will flicker several times during the refresh process (the number of flickers depends on the refresh time), and the flicker is to remove the afterimage to achieve the best display effect.
  - Partial refresh: The e-paper screen has no flickering effect during the refresh process. Users who use the partial brushing function note that after refreshing several times, a full refresh operation should be performed to remove the residual image, otherwise the residual image problem will become more and more serious, or even damage the screen (currently only some black and white e-paper screens support partial refreshing, please refer to product page description).
  - Refresh rate
  - During use, it is recommended that customers set the refresh interval of the e-paper screen to at least 180 seconds (except for products that support the local brush function).
  - During the standby process (that is, after the refresh operation), it is recommended that the customer set the e-paper screen to sleep mode, or power off (the power supply part of the e-paper screen can be disconnected with an analog switch) to reduce power consumption and prolong the life of the e-paper screen. (If some e-paper screens are powered on for a long time, the screen will be damaged beyond repair.)
  - During the use of the three-color e-paper screen, it is recommended that customers update the display screen at least once every 24 hours (if the screen remains the same screen for a long time, the screen burn will be difficult to repair).
- Application
  - The e-paper screen is recommended for indoor use. If it is used outdoors, it is necessary to avoid direct sunlight on the e-paper screen, and at the same time, take UV protection measures, because charged particles will dry out under strong light for a long time, resulting in loss of activity and failure to refresh. This situation is irreversible. When designing e-paper screen products, customers should pay attention to determining whether the use environment meets the requirements of an e-paper screen.

Question: What is the refresh rate/lifetime of the e-paper screen?

Answer:

Ideally, with normal use, it can be refreshed 1,000,000 times (1 million times).

Question: After using for a period of time, the screen refresh (full refresh) has a serious afterimage problem that cannot be repaired?

Answer:

Power on the development board for a long time, after each refresh operation, it is recommended to set the screen to sleep mode or directly power off processing, otherwise, the screen may burn out when the screen is in a high voltage state for a long time.

Question: After the e-paper screen enters deep sleep mode, can it be refreshed again?

Answer:

Yes, but you need to re-initialize the electronic paper with software.

Question: Why is the image displayed offset?

Answer:

Maybe the SPI rate is too high, resulting in data loss, try to reduce the SPI rate.
Insufficient or unstable power supply leads to data loss.
The data cable is too long to cause data loss, the extension cable should not exceed 20cm.

Question: What is the waveform file of the e-paper screen and what does it do?

Answer:

The display gray scale of electrophoretic electronic paper is determined by the spatial position of the particles in the Microcapsule or Microcup. The electrophoresis phenomenon occurs between black particles and white particles under the action of voltage. This voltage sequence that promotes the electrophoretic movement of the particles is the driving force of the electronic paper. waveform. The driving waveform is the core part of the electronic paper display, and the optimization of the driving waveform will directly affect the display effect of the display. The driving waveform file is used to describe the parameters formed by the voltage sequence that promotes the electrophoretic movement of the particles, and it needs to be called regularly when the electronic paper is refreshed. Different batches of e-paper diaphragms and electrophoretic matrices require different voltage values when driving the display due to materials, manufacturing processes, etc. The waveform of the e-paper screen is reflected in the relationship between grayscale, voltage, and temperature. Generally speaking, after each batch of electrophoresis matrix is generated, there will be a corresponding waveform file in the form of a .wbf file. The film manufacturer will provide the waveform file and electrophoresis matrix to the manufacturer of the electronic paper screen, and then the manufacturer of the electronic paper screen integrates the protection board, substrate, and driver and then provides it to customers; if the waveform file does not correspond to the screen, it is likely that the display cannot be displayed or the display effect is unsatisfactory. Generally, the waveform file has OTP built into the driver IC of the ink screen when leaving the factory, and some programs we provide also called external waveform files to drive the e-paper screen.

Question: What do LUT and OTP stand for?

Answer:

LUT is the abbreviation of LOOK UP TABLE, and OTP is the abbreviation of ONE TIME PROGRAM. The original intention of LUT is to load waveform files, and the waveform files are divided into OTP and REGISTER. Among them, OTP is the built-in waveform storage method, and REGISTER is the external waveform storage method.

Question: What is the process of refreshing e-paper partially?

Answer:

- There are mainly two types of ink screens.
  - One is to refresh the background image first.
  - The other is to alternately refresh old data and new data.

Question: How to refresh in different positions at the same time?

Answer:

Simultaneous brushing in different locations needs to be operated in the program design, that is, first brushing the data of different locations into the electronic paper IC, and finally doing the Update/TurnOnDisplay uniformly.

Question: Are bare screens shipped with a film?

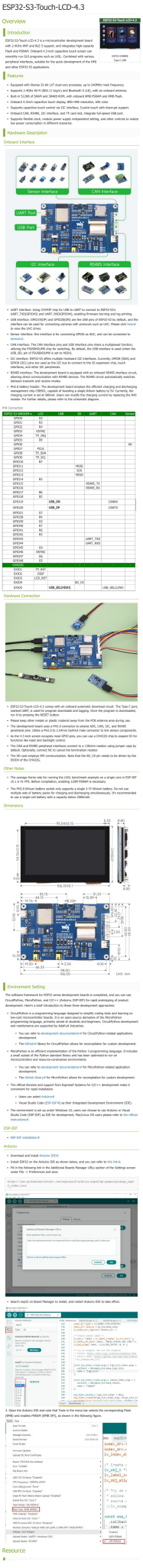
Answer:

with film.

Question: Does e-paper have a built-in temperature sensor?

Answer:

At present, all screens have built-in temperature sensors, and you can also use an external LM75 temperature sensor with IIC pins.


ESP32-S3-Touch-LCD-4.3

Overview

Introduction

Features

Hardware Description

Onboard Interface

Sensor Interface
CAN Interface
UART Port
USB Port
I2C Interface
RS485 Interface

Hardware Connection

Other Notes

Dimensions

Environment Setting

ESP-IDF

Arduino

Resource

[ESP32-S3-Touch-LCD-4.3 Development Board Overview and Setup]

A comprehensive guide to the ESP32-S3-Touch-LCD-4.3 development board, detailing its features, hardware description, pin connections, environmental setup for CircuitPython, MicroPython, and C/C++ (Arduino, ESP-IDF), and configuration within the Arduino IDE.

Document

- ESP32 Arduino Core's documentation
- arduino-esp32
- FAT-FS

Demo

Software

Datasheet

- ESP32-S3 Series Datasheet
- ESP32-S3 Wroom Datasheet
- CH343 Datasheet
- TXS0108