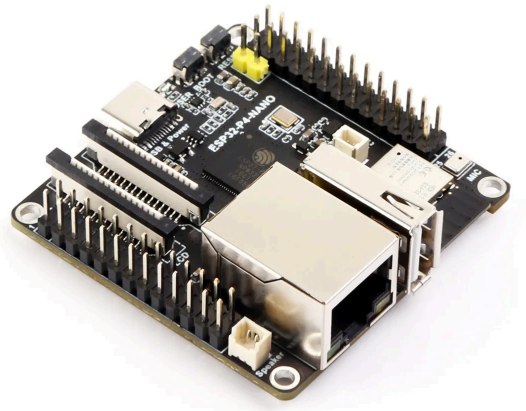
ESP32-P4-NANO



The **ESP32-P4-NANO** is a high-performance dual-core plus single-core RISC-V development board based on the ESP32-P4 chip. It supports rich human-machine interaction interfaces, including MIPI-CSI (with an integrated Image Signal Processor, ISP) and MIPI-DSI interfaces, as well as common peripherals such as SPI, I2S, I2C, LED PWM, MCPWM, RMT, ADC, UART, and TWAI. It also features USB OTG 2.0 HS, Ethernet, and SDIO Host 3.0 for high-speed connectivity. The ESP32-P4 chip integrates a digital signature peripheral and a dedicated key management unit to ensure data and operational security. The ESP32-P4-NANO is designed for high-performance and high-security applications, fully meeting the increased demands for HMI support, edge computing capability, and I/O connectivity in embedded applications.

SKU	Product
29026	ESP32-P4-NANO

SKU	Product
29027	ESP32-P4-NANO-KIT-A
29028	ESP32-P4-NANO-KIT-B
29030	ESP32-P4-NANO-KIT-C
29031	ESP32-P4-NANO-KIT-D

If you are looking for:

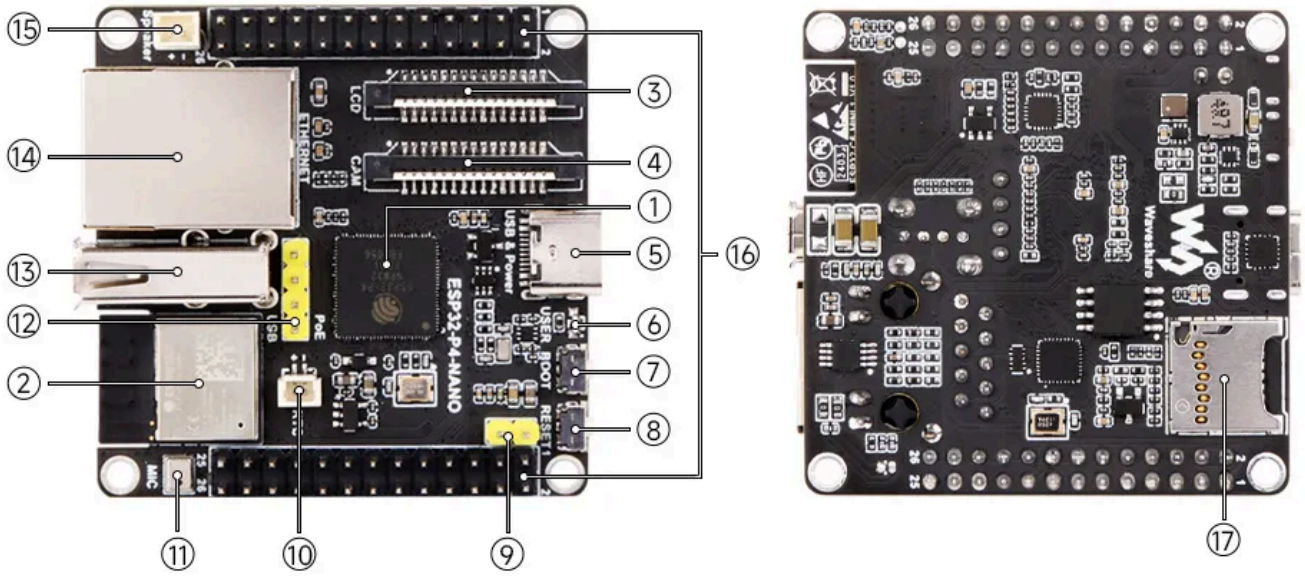
- **Coding with AI tools**: Click **Ask AI** and select an AI tool to ask questions about this page, or choose **Copy page** to copy its content and paste it into an AI tool for coding assistance.
- **Schematics, datasheets, and example code**: See Resources
- **ESP-IDF development**: See Working with ESP-IDF
- **Troubleshooting**: See Product FAQ or contact Technical Support

Features

- Processor
  - Equipped with a RISC-V 32-bit dual-core processor (HP system), featuring DSP and ISA extensions, a Floating-Point Unit (FPU), with a main frequency of up to 360 MHz.
  - Equipped with a RISC-V 32-bit single-core processor (LP system), with a main frequency of up to 40 MHz.
  - Equipped with an ESP32-C6 WIFI/BT co-processor, expanding WIFI 6/Bluetooth 5 and other functions through SDIO
- Memory
  - 128 KB of high-performance (HP) system read-only memory (ROM).
  - 16 KB of low-power (LP) system read-only memory (ROM).
  - 768 KB of high-performance (HP) L2 memory (L2MEM).
  - 32 KB of low-power (LP) SRAM.
  - 8 KB of system tightly coupled memory (TCM).
  - 32 MB PSRAM stacked in the package, onboard 16MB Nor Flash
- Peripheral Interfaces
  - 2×2×13 headers bring out 28 remaining programmable GPIOs

- Onboard Type-A USB 2.0 OTG interface, 100Mbps Ethernet interface, SDIO3.0 TF card slot, Type-C UART programming port for convenient use in different scenarios
- Onboard speaker interface and microphone, enabling ideal audio functionality using the Codec chip and amplifier chip
- Onboard MIPI-CSI high-definition camera interface, supporting Full HD 1080P image capture and encoding, integrated Image Signal Processor (ISP), H.264 video encoder, supports H.264 & JPEG video encoding (1080P @30fps), facilitating applications in computer vision, machine vision, and other fields
- Onboard MIPI-DSI high-definition display interface, integrated Pixel Processing Accelerator (PPA), 2D graphics acceleration controller (2D DMA), supports JPEG image decoding (1080P @30fps), providing strong support for high-definition display and smooth HMI experiences, suitable for applications such as smart home control panels, industrial control panels, and vending machines
- Reserved PoE module interface, making the board's power supply more flexible. With only a single Ethernet cable connected to a PoE device, the ESP32-P4-NANO series can achieve both networking and power delivery

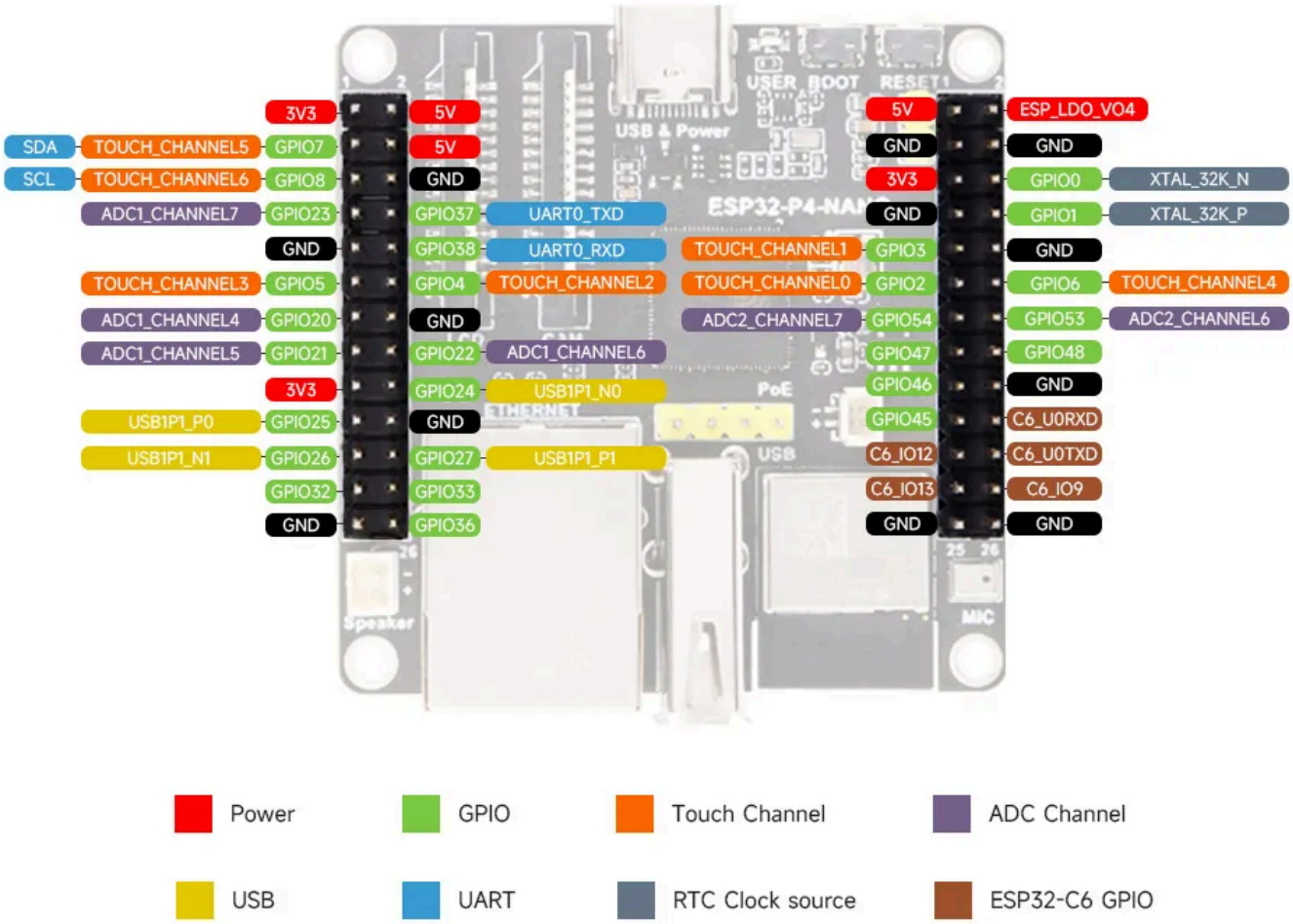
Hardware Description



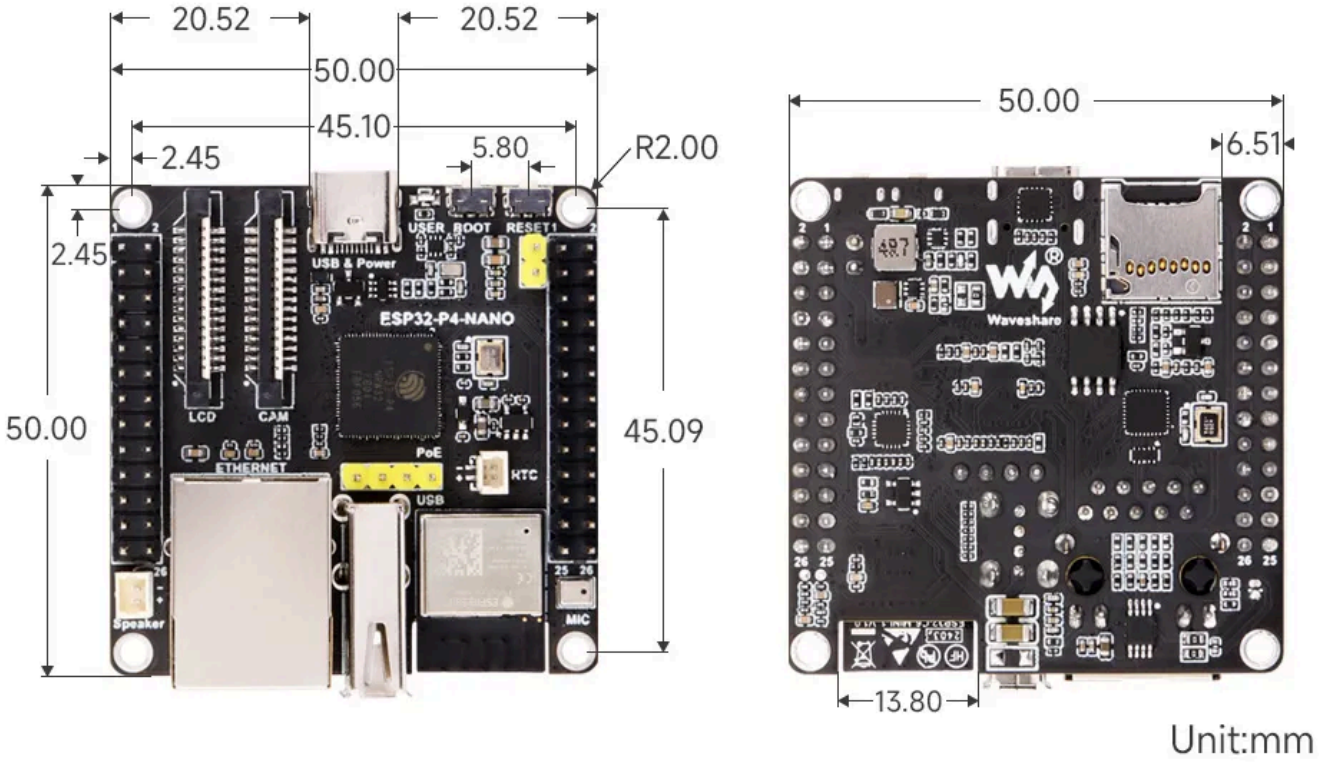
1. **ESP32-P4NRW32X** ESP32-P4 with 32MB stacked PSRAM
2. **ESP32-C6-MINI-1** SDIO interface protocol, expands ESP32-P4-NANO with Wi-Fi 6 and Bluetooth 5
3. **Display Interface** MIPI-DSI (2-lane), compatible with `5` / `7` / `8` / `10.1` inch DSI screens

4. **Camera Interface** MIPI-CSI (2-lane), compatible with cameras such as `OV5647`
5. **USB Type-C Interface** For power supply, program flashing, and serial debugging (via USB-to-UART bridge)
6. **USER-LED** Power indicator
7. **BOOT Button** Press during power-up or reset 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 batteries (rechargeable only)
11. **SMD Microphone**
12. **PoE Module Header**
13. **Type-A Port** USB OTG 2.0 High Speed interface
14. **100M RJ45 Ethernet Port**
15. **Speaker Interface** MX1.25 2P connector, supports 8Ω 2W speaker
16. **GPIO Header**
17. **TF Card Slot** SDIO 3.0 interface protocol

Pinout Definition


3V3
5V
SDA
TOUCH_CHANNEL5
GPIO7
5V
SCL
TOUCH_CHANNEL6
GPIO8
GND
ADC1_CHANNEL7
GPIO23
GPIO37
UART0_TXD
GND
GPIO38
UART0_RXD
TOUCH_CHANNEL3
GPIO5
GPIO4
TOUCH_CHANNEL2
ADC1_CHANNEL4
GPIO20
GND
ADC1_CHANNEL5
GPIO21
GPIO22
ADC1_CHANNEL6
3V3
GPIO24
USB1P1_N0
USB1P1_P0
GPIO25
GND
USB1P1_N1
GPIO26
GPIO27
USB1P1_P1
GPIO32
GPIO33
GND
GPIO36
USER_BOOT
RESET
USB & Power
ESP32-P4-NANO
TOUCH_CHANNEL1
GPIO3
TOUCH_CHANNEL0
GPIO2
ADC2_CHANNEL7
GPIO54
GPIO47
GPIO46
GPIO45
C6_IO12
C6_IO13
PoE
ETHERNET
USB
Speaker
MIC
5V
GND
3V3
GND
ESP_LDO_VO4
GND
GPIO0
XTAL_32K_N
GPIO1
XTAL_32K_P
GND
GPIO6
TOUCH_CHANNEL4
GPIO53
ADC2_CHANNEL6
GPIO48
GND
C6_U0RXD
C6_U0TXD
C6_IO9
GND
Power
GPIO
Touch Channel
ADC Channel
USB
UART
RTC Clock source
ESP32-C6 GPIO

Dimensions



Development Tools

Arduino generally offers a gentler learning curve and may be more approachable for beginners and hobbyists. ESP-IDF provides advanced tooling and finer control over system behavior, making it better suited to complex projects and applications with demanding performance requirements.

> **WARNING**
>
> ESP32-P4 currently has limited adaptation on the Arduino platform. To ensure development stability, it is recommended to use ESP-IDF for development at this stage. If you have requirements for using the Arduino platform, you can participate in the discussion on GitHub: Support of Arduino

- **ESP-IDF**, short for Espressif IoT Development Framework, is a development framework provided by Espressif for the ESP series chips. It is developed using the C language, including a compiler, debugger, and flashing tool, etc., and can be developed via the command lines or through an integrated development environment (such as Visual Studio Code with the Espressif IDF plugin). The plugin offers features such as code navigation, project management, and debugging, etc. We recommend using VS Code for development. For the

specific configuration process, please refer to the **Working with ESP-IDF**. The tutorial also provides relevant example programs for reference.

Give Feedback