
ESP32-S3 Supermini Development Board User Manual

Model: ESP32-S3 Supermini | Brand: TENSTAR ROBOT

1. INTRODUCTION

The TENSTAR ROBOT ESP32-S3 Supermini is an ultra-compact IoT development board built around the Espressif ESP32-S3 WiFi/Bluetooth dual-mode chip. This board features a powerful 32-bit RISC-V CPU operating at up to 160 MHz, offering robust computing capabilities for a wide range of applications. It provides excellent RF performance, supporting IEEE 802.11b/g/n WiFi and Bluetooth (BT) 5.0 protocols, and includes an external antenna for enhanced wireless signal strength. Designed with a simple solder-mount form factor, it integrates rich on-chip peripherals, making it ideal for low-power IoT devices and wireless wearables.

TENSTAR ROBOT

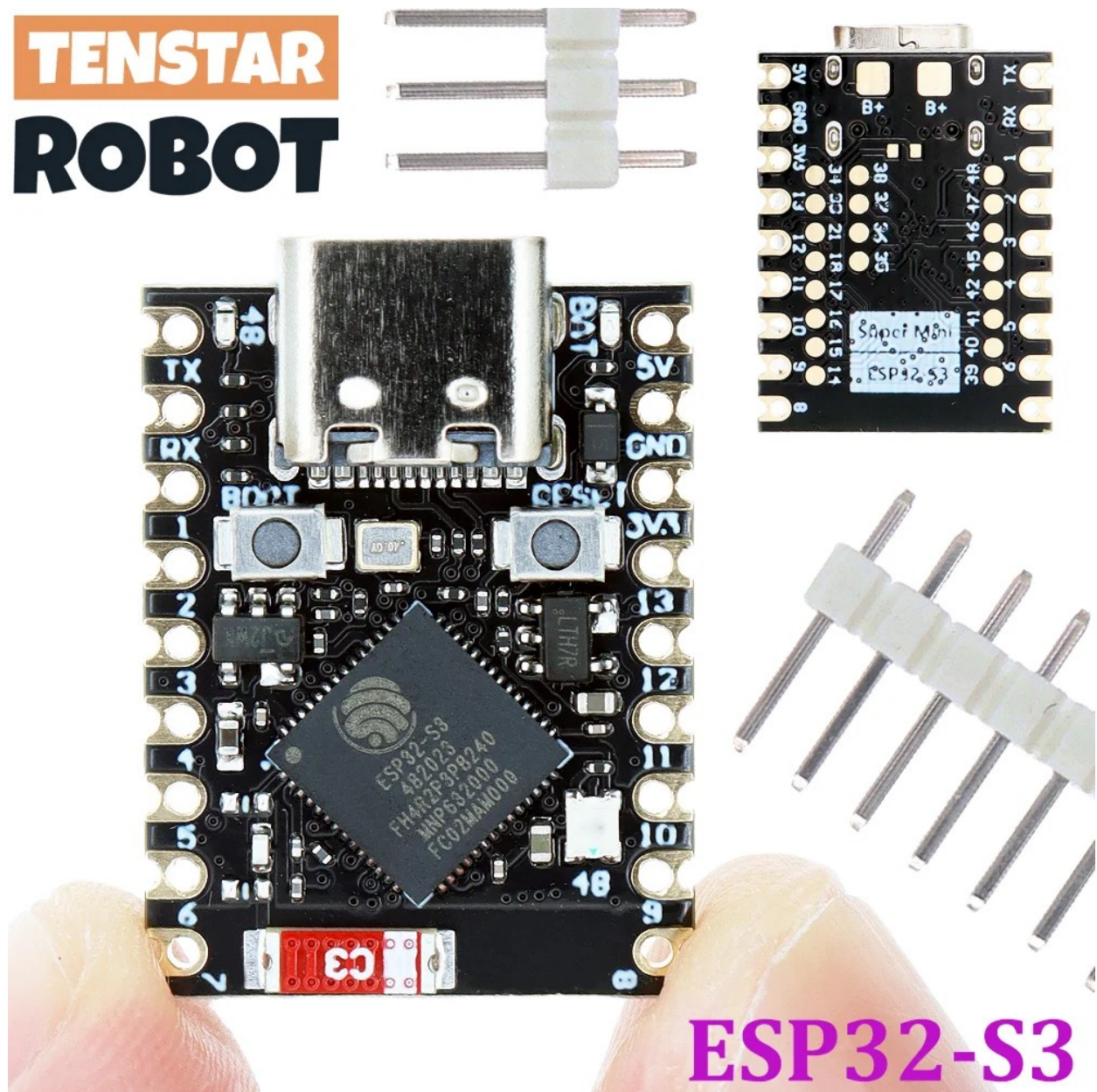


Figure 1: ESP32-S3 Supermini Development Board (Front View)

2. PRODUCT OVERVIEW

2.1 Key Features

- **Powerful CPU:** Espressif ESP32-S3, 32-bit RISC-V single-core processor, up to 160 MHz.
- **Wireless Connectivity:** WiFi (802.11b/g/n protocol, 2.4GHz, supports Station mode, SoftAP mode, SoftAP+Station mode, hybrid mode) and Bluetooth 5.0.
- **Ultra-low Power Consumption:** Deep sleep power consumption of

approximately 43μA.

- **Memory:** 512KB SRAM, 384KB ROM, built-in 4MFlash.
- **Chip Model:** ESP32-S3FH4R2.
- **Compact Size:** Ultra-small dimensions (22.5mm x 18mm), suitable for wearables and small projects.
- **Security Features:** Encryption hardware accelerators supporting AES-128/256, hashing, RSA, HMAC, digital signatures, and secure startup.
- **Rich Interfaces:** 1xI2C, 1xSPI, 2xUART, 11xGPIO (PWM capable), 4xADC.
- **Onboard Indicators:** RGB LED, blue LED.
- **Buttons:** Small reset button and a boot mode button.

2.2 Pinout Diagram

The following diagram illustrates the pin assignments and dimensions of the ESP32-S3 Supermini Development Board.

ESP32-S3 Supermini

The ESP32-S3 SuperMini is an IoT mini development board based on the Espressif ESP32-S3 WiFi/ Bluetooth dual-mode chip. The ESP32-S3 is a 32-bit RISC-V CPU that contains the FPU (floating point unit), which can perform 32-bit single-precision operations and has powerful computing power. It has excellent RF performance and supports IEEE 802.11b /g/n WiFi and Bluetooth 5 (LE) protocols. The board comes with an external antenna to enhance signal strength for wireless applications. It also has a small and delicate form factor combined with a single-sided surface mount design. It is equipped with a wealth of interfaces, with 11 digital I/ OS that can be used as PWM pins and 4 analog I/ OS that can be used as ADC pins. It supports four serial interfaces: UART, I2C and SPI. The board also has a small reset button and a boot loader mode button.

Based on the above features, ESP32S3SuperMini is positioned as a high-performance, low-power and cost-effective iot mini development board suitable for low-power iot applications and wireless wearable applications.

Product parameter

- **Powerful CPU:** ESP32-S3, 32-bit RISC-V single-core processor, running at up to 160 MHz
- **WiFi:** 802.11b/g/n protocol, 2.4GHz, support Station mode, SoftAP mode, SoftAP+Station mode, hybrid mode

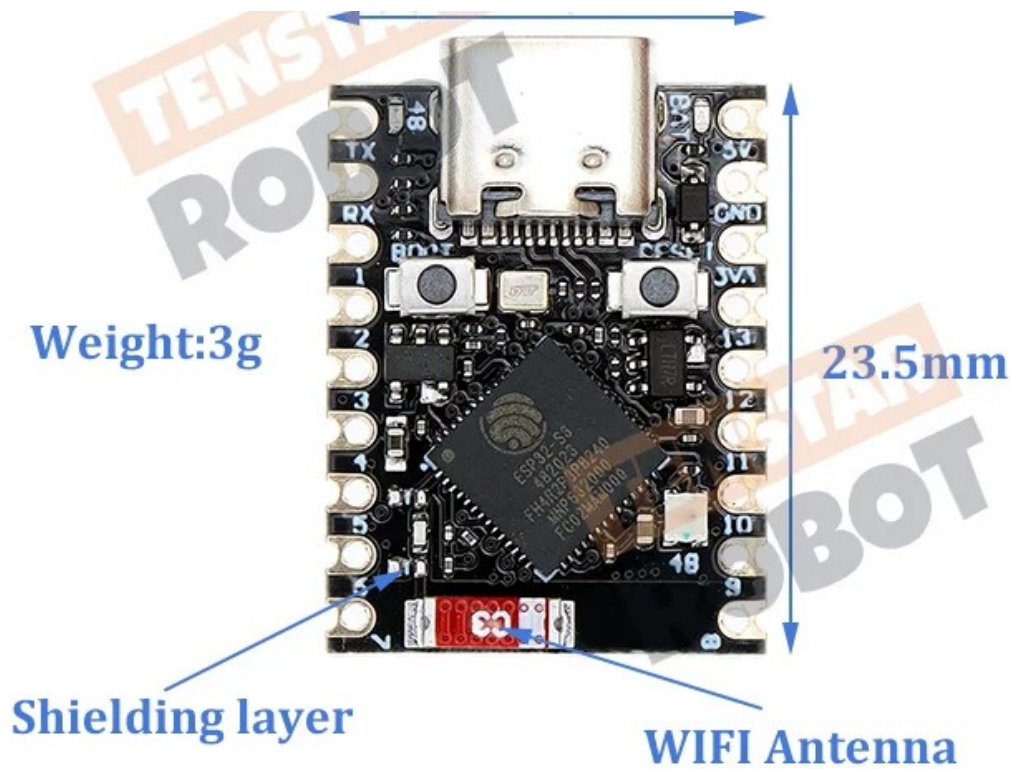


Figure 2: ESP32-S3 Supermini Pinout and Dimensions

Key pin functions include:

- **Power:** 5V (USB power), 3V3 (regulated 3.3V output), GND.
- **Serial Communication:** TX, RX (UART).
- **GPIO:** General Purpose Input/Output pins, many of which support PWM.
- **ADC:** Analog-to-Digital Converter pins.
- **I2C/SPI:** Dedicated pins for I2C and SPI communication protocols.
- **BOOT:** Boot mode button.
- **RST:** Reset button.

3. SPECIFICATIONS

Feature	Specification
Brand Name	TENSTAR ROBOT
Type	Module

Feature	Specification
Condition	New
Application	Computer, IoT
Supply Voltage	5V (via USB-C)
Dissipation Power	43μA (deep sleep)
Operating Temperature	-30°C to 50°C
Origin	Mainland China
Package Type	SMD
Dimensions	22.5mm x 18mm
Weight	3g
Wireless Protocols	WiFi (802.11b/g/n), Bluetooth 5.0
Flash Memory	4MFlash

4. SETUP INSTRUCTIONS

- 1. Connect to Computer:** Use a USB-C cable to connect the ESP32-S3 Supermini board to your computer.
- 2. Install IDE:** Download and install the Arduino IDE or your preferred Micropython development environment.
- 3. Install ESP32-S3 Board Support:**
 - For Arduino IDE, go to File > Preferences, add the ESP32 board manager URL, then go to Tools > Board > Boards Manager and install "esp32 by Espressif Systems".
 - For Micropython, follow the official Micropython documentation for ESP32-S3 to flash the firmware and set up your environment.
- 4. Select Board:** In your IDE, navigate to Tools > Board and select "ESP32S3 Dev Module".

5. **Select Port:** Go to Tools > Port and select the serial port corresponding to your connected ESP32-S3 board.

5. OPERATING INSTRUCTIONS

1. **Upload Code:** Write your code in the IDE and click the "Upload" button to flash it to the ESP32-S3 board.
2. **Enter Download Mode (if needed):** If the upload fails or the board is not recognized, you may need to manually put the board into download mode.
 - **Method 1:** Press and hold the BOOT button while powering on the board.
 - **Method 2:** Press and hold the BOOT button, then press and release the RESET button, and finally release the BOOT button. This sequence enables the ESP32-S3 to enter download mode. This may be required for each new connection or if port instability occurs.
3. **Execute Uploaded Program:** After a successful code upload, press the RESET button on the board to execute the newly uploaded program.
4. **Serial Communication:** To view serial output from your program, ensure "USB CDC On Boot" is set to "Enabled" in your IDE's toolbar settings.

6. MAINTENANCE

- **Storage:** Store the development board in a dry, anti-static environment away from direct sunlight and extreme temperatures.
- **Handling:** Handle the board with care to avoid damaging components or bending pins. Avoid touching the exposed circuitry unnecessarily.
- **Cleaning:** If necessary, gently clean the board with a soft, dry brush or compressed air. Do not use liquids or abrasive cleaners.
- **Power Supply:** Always use a stable 5V power supply via the USB-C port. Ensure the power source can provide sufficient current for your

application.

7. TROUBLESHOOTING

Refer to the following common issues and their solutions:

FAQ

■ Com is not recognized on Arduino

Enter the download mode: **Method 1:** Press and hold BOOT to power on. **Method 2:** Press and hold down the BOOT button of the ESP32S3, press the RESET button, release the RESET button, and then release the BOOT button to enable the ESP32S3 to enter download mode. (Each connection needs to re-enter the download mode, sometimes press once, the port instability will be disconnected, you can judge by the port identification sound)

■ The program will not run after uploading

After the upload succeeds, you need to press the Reset button to execute the upload

■ Serial port cannot print

Set the USB CDC On Boot on the toolbar to Enabled

■ IDE board options:

ESP32s3 Dev Module

Figure 3: Troubleshooting Guide

- **COM Port Not Recognized on Arduino:**
 - **Method 1:** Press and hold the BOOT button while powering on the board.
 - **Method 2:** Press and hold the BOOT button, then press the RESET button, release the RESET button, and finally release the BOOT button. This will enable the ESP32-S3 to enter download mode. You may need to re-enter download mode for each connection, especially if the port becomes unstable.
- **Program Will Not Run After Uploading:**
 - After a successful upload, you must press the RESET button on the board to execute the uploaded program.
- **Serial Port Cannot Print:**
 - Ensure that "USB CDC On Boot" is set to "Enabled" in your IDE's toolbar settings.

- **Incorrect IDE Board Option:**

- Always select "ESP32S3 Dev Module" in your IDE's board selection menu.

8. USER TIPS

- For reliable flashing, always ensure the board is in download mode before attempting to upload new firmware. The BOOT button sequence is crucial.
- If you experience intermittent serial communication, double-check your "USB CDC On Boot" setting and try re-entering download mode.
- When experimenting with new code, always perform a hard reset (press the RESET button) after uploading to ensure the new program starts correctly.

9. WARRANTY AND SUPPORT

This product is covered by the standard manufacturer's warranty against defects in materials and workmanship. For technical support, documentation, or further assistance, please refer to the TENSTAR ROBOT official channels or the Espressif Systems documentation for the ESP32-S3 chip. Keep your purchase receipt for warranty claims.

	mode for ESP32-C3 modules.
	ESP32-S3 DevKits Documentation Comprehensive documentation for Espressif's ESP32-S3 series development boards, including the ESP32-S3-DevKitC-1, ESP32-S3-DevKitM-1, ESP32-S3-USB-OTG, and ESP32-S3-LCD-EV-Board. This guide provides getting started instructions, hardware references, revision details, and related resources for developers.
	ESP32-S3 Development Kits Documentation Espressif Systems Comprehensive documentation for Espressif Systems' ESP32-S3 series development kits, covering DevKitC-1, DevKitM-1, USB-OTG, LCD-EV-Board, and USB-Bridge. Includes hardware guides, features, and getting started information.
	ESP32-S3 Development Boards Documentation Espressif Systems Comprehensive documentation for Espressif Systems' ESP32-S3 development boards, including DevKitC-1, DevKitM-1, USB-OTG, LCD-EV-Board, and USB-Bridge. Guides cover setup, hardware, and application development.
	ESP32 Hardware Design Guidelines - Espressif Systems Comprehensive hardware design guidelines for the ESP32 series of chips, modules, and development boards, covering schematic design, PCB layout, and application examples.
	ESP32-S3 ESP-IDF Programming Guide Comprehensive guide for developers on setting up and using the Espressif ESP-IDF framework with the ESP32-S3 SoC, covering hardware setup, API references, and project development for IoT applications.

ASK A QUESTION ABOUT THIS MANUAL

Ask about setup, troubleshooting, compatibility, parts, safety, or missing instructions. Manuals+ will review the question and use this page's manual context to help answer it.