DORHEA 13051-1

DORHEA ESP32-S NodeMCU Development Board User Manual

1. Introduction

The DORHEA ESP32-S NodeMCU Development Board is a versatile 2.4GHz dual-mode WiFi and Bluetooth microcontroller, featuring the ESP-WROOM-32 module. This board integrates a dual-core processor and the CP2102 USB-to-serial chip, enabling stable and fast data transmission via a micro USB interface. It is designed for a wide range of applications, including home automation, wireless industrial control, and wireless positioning systems. The ESP32-S supports Lua programming, LWIP protocol, and FreeRTOS, offering flexibility for development. It operates in three modes: Access Point (AP), Station (STA), and a combination of both (AP + STA). Its low-power coprocessor allows for continuous monitoring of peripherals while the main CPU is powered down, optimizing energy consumption.

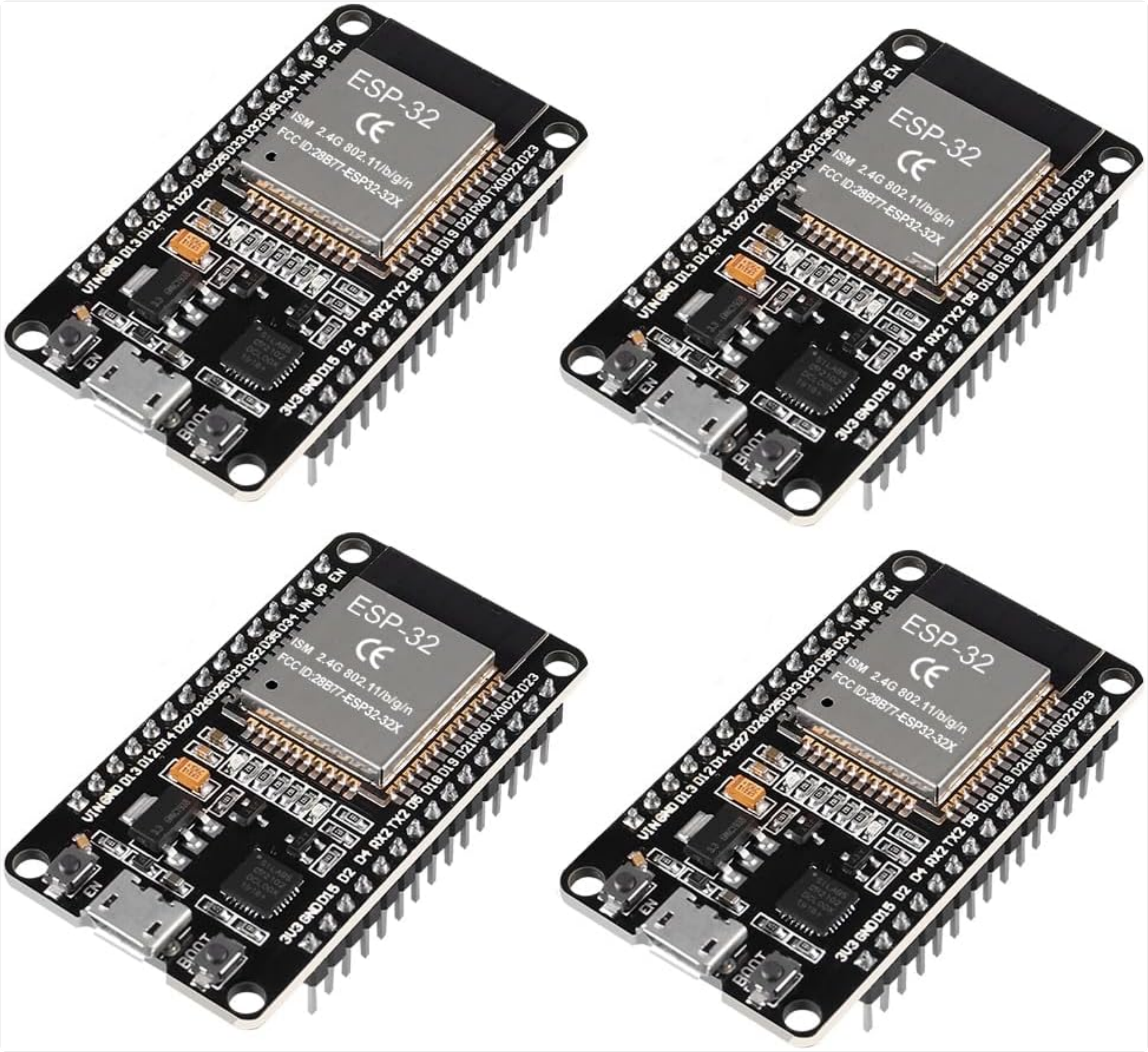


Figure 1: Front view of the DORHEA ESP32-S NodeMCU Development Board, showing the ESP-WROOM-32 module, micro USB port, and pin headers.

2. Setup Guide

2.1 Connecting the Board

Connect the ESP32-S development board to your computer using a micro USB cable. The CP2102 chip facilitates the serial communication required for programming and debugging.

2.2 Driver Installation

The CP2102 USB-to-serial chip may require specific drivers for your operating system. These drivers are typically available from the Silicon Labs website. Install them before attempting to program the board.

2.3 Development Environment Setup

The ESP32-S board is compatible with various development environments, including the Arduino IDE and PlatformIO (within Visual Studio Code). Follow the respective environment's instructions to add ESP32 board support.

- **Arduino IDE:** Install the ESP32 board package through the Boards Manager.

- **PlatformIO:** Configure your project to use the ESP32 framework.

2.4 Programming Mode

To upload code to the board, it typically needs to be in programming mode. This is usually achieved by holding down the **BOOT** button while initiating the upload process in your IDE, then releasing it once the upload begins. Some environments or board configurations may not require manual button pressing.

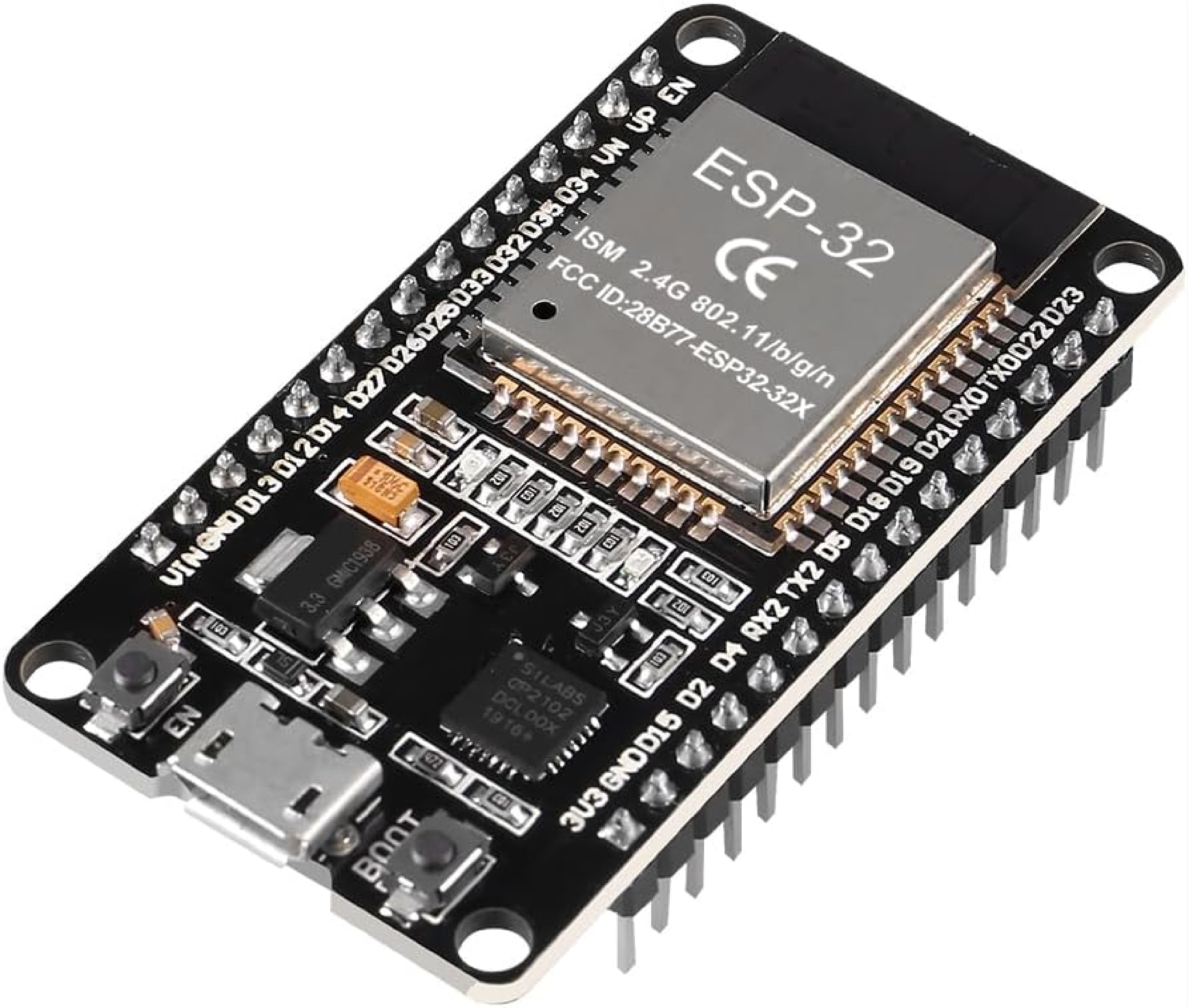

ESP-32
CE
ISM 2.4G 802.11/b/g/n
FCC ID:28B77-ESP32-32X
VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
SILABS CP2102 DCL00X 1916+
EN
BOOT

Figure 2: Angled view of the ESP32-S development board, showing the ESP-WROOM-32 module and micro USB port.

3. Operating Instructions

3.1 Basic Operation

Once programmed, the ESP32-S board will execute the uploaded code. The board features two touch buttons: one for **Reset** (labeled EN) and another to enable the module to enter program mode (labeled BOOT).

3.2 WiFi and Bluetooth Modes

The ESP32-S supports three operational modes for wireless communication:

- **AP (Access Point) Mode:** The ESP32-S acts as a Wi-Fi hotspot, allowing other devices to connect to it.
- **STA (Station) Mode:** The ESP32-S connects to an existing Wi-Fi network.
- **AP + STA Mode:** The ESP32-S simultaneously functions as both an Access Point and a Station.

3.3 Pinout Diagram

Refer to the pinout diagram for proper connection of external components and sensors. Understanding the GPIO assignments, power pins, and special functions (ADC, DAC, Touch) is crucial for project development.

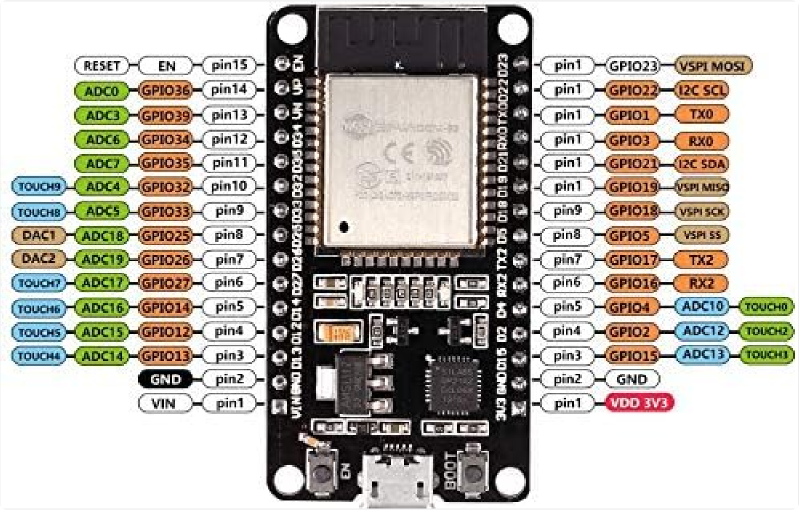


Figure 3: Detailed pinout diagram of the ESP32-S NodeMCU board.

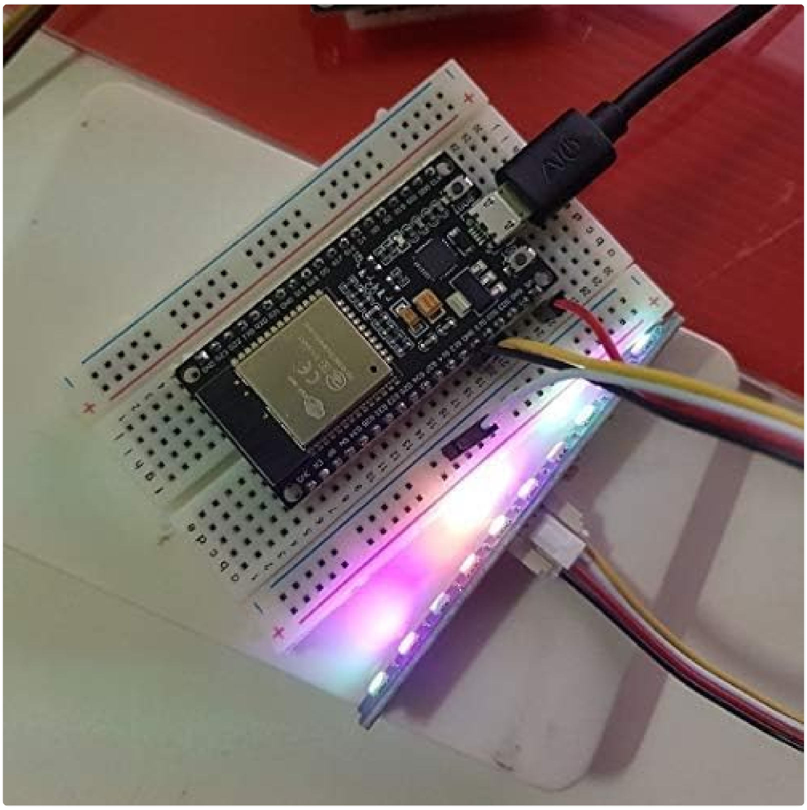


Figure 4: The ESP32-S development board connected to a breadboard with an LED strip, demonstrating a typical application setup.

4. Maintenance

To ensure the longevity and reliable operation of your ESP32-S development board, follow these general maintenance guidelines:

- **Handle with Care:** Avoid dropping the board or subjecting it to excessive physical stress.
- **Static Discharge:** Always handle the board in an environment free of static electricity to prevent damage to sensitive components.
- **Cleanliness:** Keep the board free from dust, dirt, and moisture. Use a soft, dry brush or compressed air for

cleaning.

- **Power Supply:** Ensure a stable and appropriate power supply. The board can draw between 500-600mA instantaneously when connecting to Wi-Fi. Using an inadequate power supply (e.g., some computer USB ports or hubs) may lead to reboots or connectivity issues.
- **Voltage Regulation:** While the board can handle input voltages up to 20V, the onboard LDO regulator has limited heat dissipation. If using a higher voltage power supply, consider stepping down the voltage to 5V with a switching regulator before connecting to the VIN pin to prevent overheating and potential failure.

5. Troubleshooting

This section addresses common issues encountered during the use of the ESP32-S development board.

5.1 Wi-Fi Connectivity Issues

- **Frequent Disconnections/Reboots:** If the board reboots or disconnects from Wi-Fi frequently, especially during connection attempts, check your power supply. The board requires significant current (500-600mA) during Wi-Fi initialization. Ensure your USB port or external power supply can provide sufficient, stable current.
- **No Wi-Fi Connection:** Verify your Wi-Fi credentials in the code. Ensure the antenna is not obstructed.

5.2 Programming and Flashing Problems

- **Failed Uploads:** Ensure the correct COM port is selected in your IDE. If using Arduino IDE, ensure the correct board type (ESP32 Dev Module) is selected. For some setups, you may need to manually hold the **BOOT** button during the upload process.
- **Driver Issues:** Confirm that the CP2102 drivers are correctly installed and recognized by your operating system.
- **"Could not connect to an Espressif device":** This error often indicates a communication problem. Check USB cable, drivers, and ensure the board is in programming mode.
- **Memory Loss/Bricking:** In rare cases, the flash memory may become corrupted. Attempting to reflash the board with a known working program is the first step. If persistent issues occur, the board may be damaged.

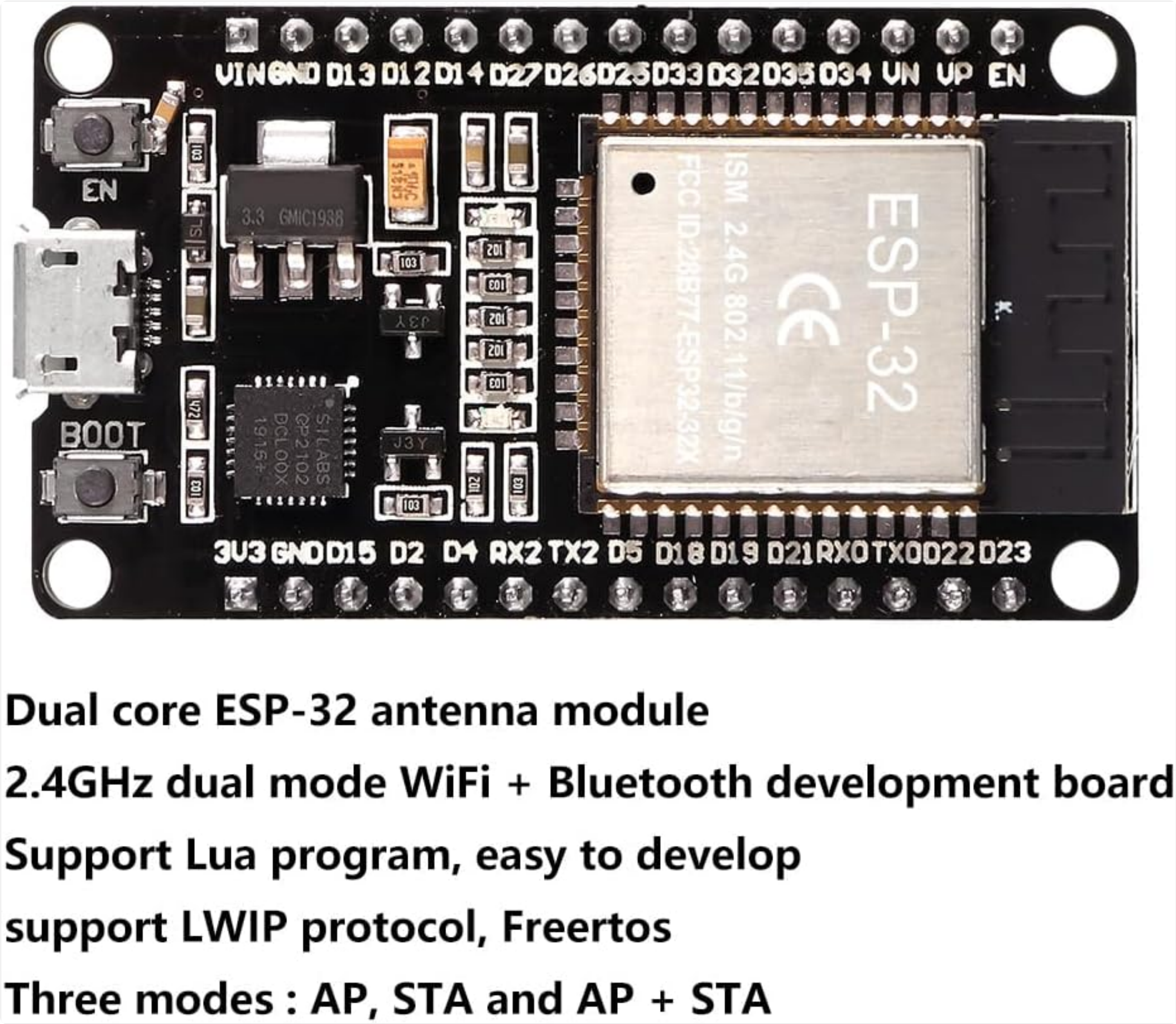
5.3 Bluetooth Functionality

- **Bluetooth Not Working:** Verify that your code correctly initializes and uses the Bluetooth module. Ensure there are no hardware conflicts with other components connected to the board.

5.4 Silkscreen Error

- **Incorrect Pin Label:** Be aware that the GPIO pin G33, located between G25 and G32, may be incorrectly marked as G23 on the silkscreen. This is a labeling error only and does not affect the physical pin's functionality.

6. Specifications


VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
BOOT
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
ESP-32
ISM 2.4G 802.11/b/g/n
FCC ID:2AB77-ESP32-32X
SILABS CP2102 DCL00X 1915+
3.3 GMIC1938

Dual core ESP-32 antenna module

2.4GHz dual mode WiFi + Bluetooth development board

Support Lua program, easy to develop

support LWIP protocol, Freertos

Three modes : AP, STA and AP + STA

Figure 5: Top-down view of the ESP32-S development board with key features labeled.

Feature	Detail
Processor	Dual-core 2.4 GHz Espressif
Wireless Connectivity	Wi-Fi (2.4GHz 802.11 b/g/n), Bluetooth
USB-to-Serial Chip	CP2102
Operating System Support	FreeRTOS
Memory Storage Capacity	4 MB
Product Dimensions (LxWxH)	1.96 x 1.10 x 0.59 inches
Item Weight	1.58 ounces
Model Number	13051-1
Color	30pin micro

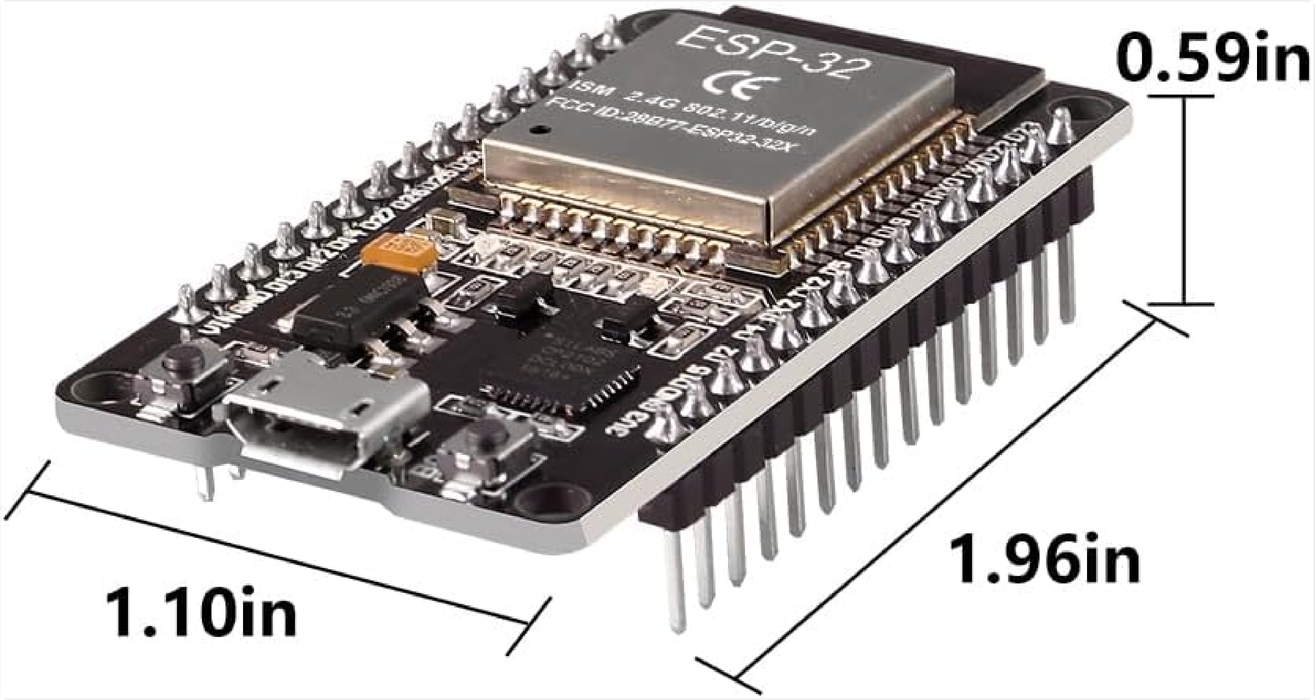


Figure 6: Side view of the ESP32-S development board with dimensions labeled.

7. Warranty and Support

7.1 Warranty Information

Specific warranty details for the DORHEA ESP32-S NodeMCU Development Board are not explicitly provided in the product information. For warranty inquiries, please contact the manufacturer or your point of purchase.

7.2 Technical Support

For technical assistance, troubleshooting beyond this manual, or further product information, please refer to the DORHEA brand store or contact their customer support channels. You can often find additional resources and community forums online for ESP32 development boards.