

AYWHP XD050

AYWHP ESP32-C6 Mini Development Board User Manual

Model: XD050 | Brand: AYWHP

1. INTRODUCTION

This manual provides detailed instructions for the AYWHP ESP32-C6 Mini Development Board (Model XD050). This compact board is designed for Internet of Things (IoT) applications, smart home devices, wireless sensor networks, prototyping, programming education, and embedded systems. It features a powerful 32-bit RISC-V processor, WiFi 6, and Bluetooth 5 Low Energy connectivity.

2. SPECIFICATIONS

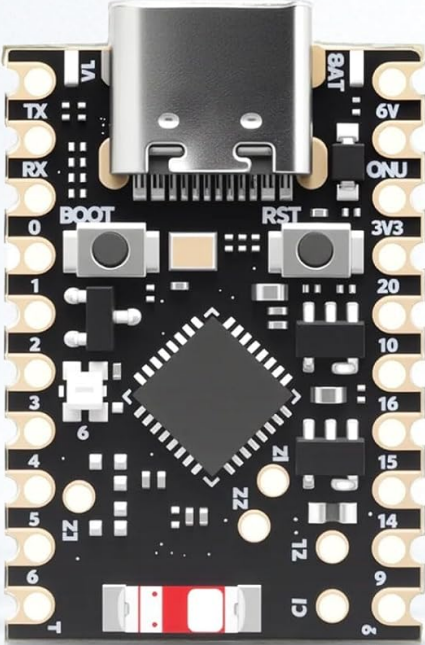
Feature	Detail
Processor	Single-core 32-bit RISC-V, 160 MHz clock frequency
Memory	320 KB ROM, 512 KB HP SRAM, 16 KB LP SRAM, 4 MB Flash
Wireless Connectivity	WiFi 6 (802.11ax) 2.4 GHz, Bluetooth 5 Low Energy (BLE)
Interfaces	USB Type-C, numerous digital I/O pins, SPI, I2C, UART
Operating Voltage	3.3 V
Dimensions	28mm x 18mm (approx. 1.1 x 0.7 inches)
Power Consumption	Supports various power-saving modes

Manufacturer	AYWHP
Model Number	XD050
Package Dimensions	11.8 x 7.8 x 1.4 cm; 10 g

3. BOARD LAYOUT AND PINOUT

The ESP32-C6 Mini Development Board features a compact design with clearly labeled pins and components. Understanding the layout is crucial for proper use and integration into your projects.

Leistung & Konnektivität



Hz

160MHz RISC-V



WiFi 6 (802.11ax)



Bluetooth 5 LE

Figure 3.1: Top-down view of the ESP32-C6 Mini Development Board, highlighting the USB-C port, BOOT and RST buttons, and pin headers for connectivity.

Produktdetails

Robustes PCB-Design. Goldkontakte für geringeren Widerstand und höhere Haltbarkeit.

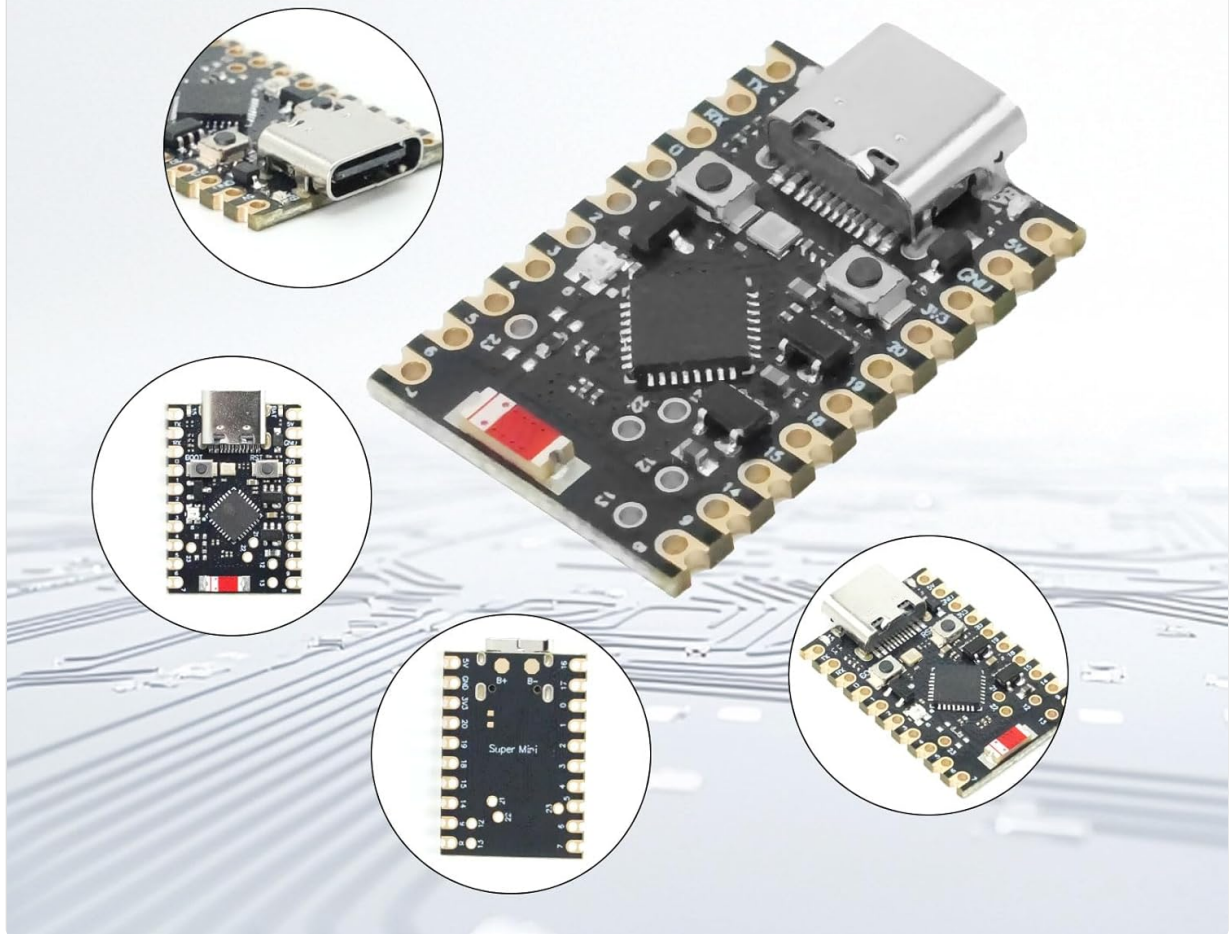


Figure 3.2: Product details showing the robust PCB design and gold contacts for improved durability and reduced resistance.

Size

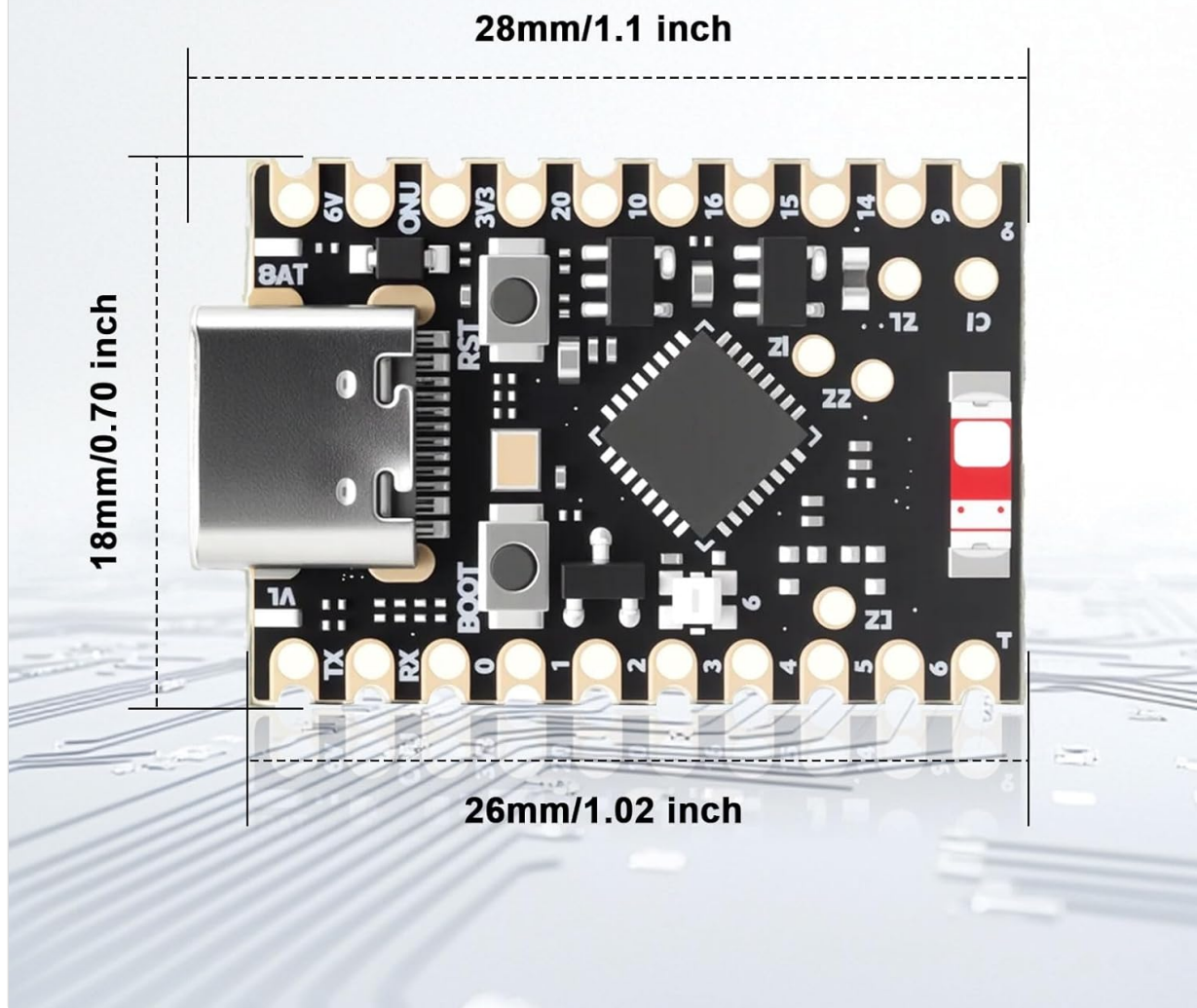


Figure 3.3: Physical dimensions of the board, measuring approximately 28mm (1.1 inches) in length and 18mm (0.70 inches) in width.

3.1 Functional Block Diagram

The functional block diagram provides an overview of the internal architecture of the ESP32-C6 chip, including the CPU system, wireless modules, peripherals, power management, and security features.

Funktionsblockdiagramm

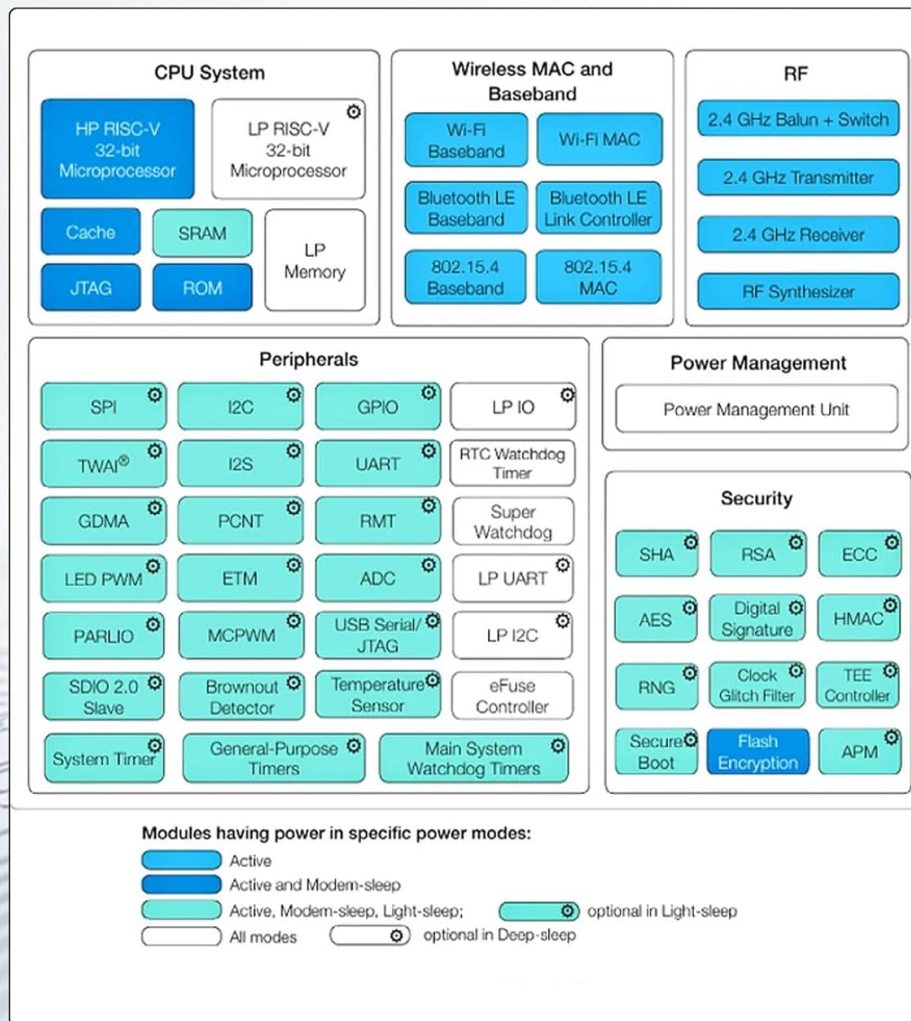


Figure 3.4: Functional Block Diagram of the ESP32-C6, detailing its core components and their interconnections.

3.2 Pinout Diagram

The pinout diagram illustrates the function of each pin on the development board. Refer to this diagram when connecting peripherals or integrating the board into custom circuits.

Produktdaten

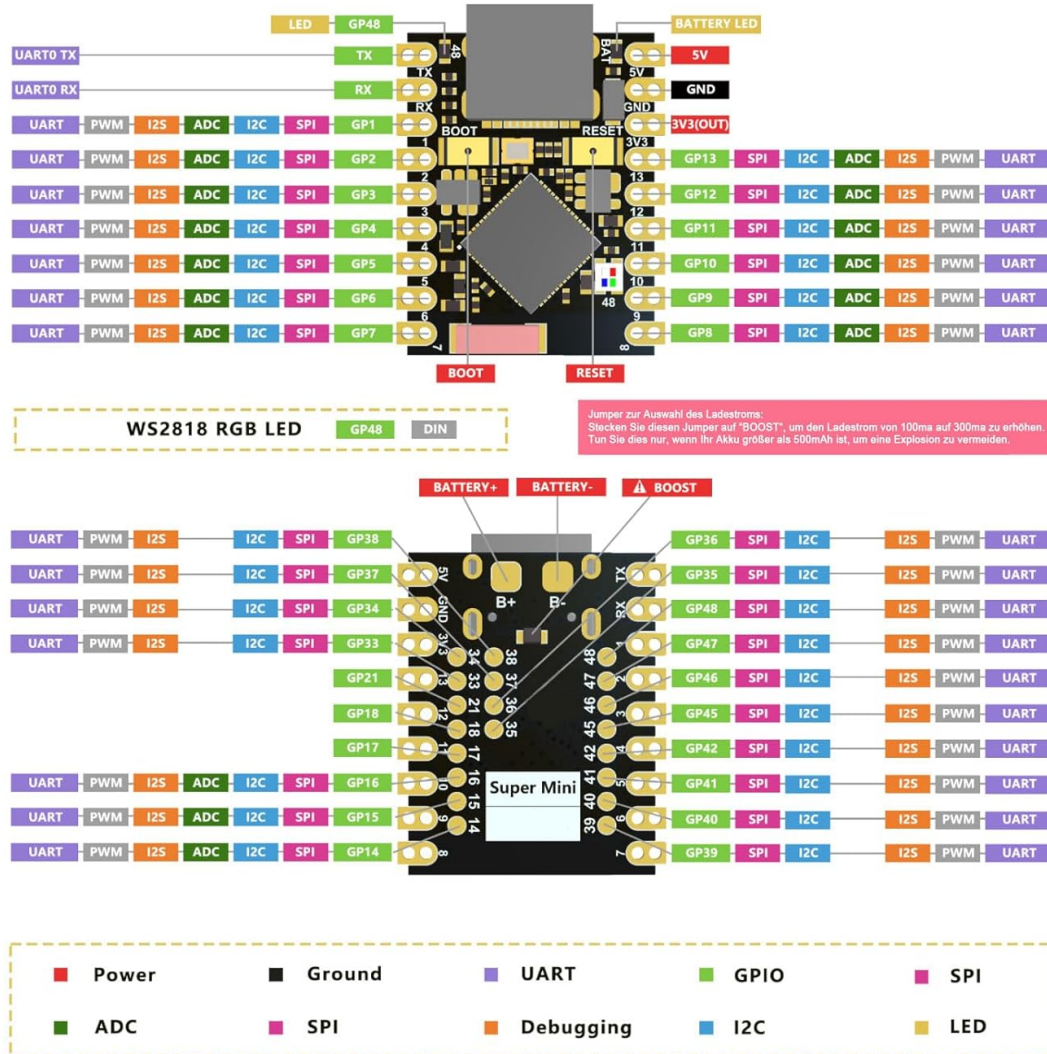


Figure 3.5: Pinout diagram of the ESP32-C6 Mini, indicating power, ground, UART, GPIO, SPI, ADC, debugging, I2C, and LED pins. Note the WS2818 RGB LED on GP48.

Key pin functions include:

- **Power Pins:** 3V3 (3.3V output), 5V (5V input/output), GND (Ground).
- **GPIO Pins:** General Purpose Input/Output pins, configurable for various digital functions.
- **UART:** Universal Asynchronous Receiver/Transmitter for serial communication (TX, RX).
- **SPI:** Serial Peripheral Interface for high-speed communication with peripherals.
- **I2C:** Inter-Integrated Circuit for communication with sensors and other devices.
- **PWM:** Pulse Width Modulation for controlling motors or LEDs.
- **ADC:** Analog-to-Digital Converter for reading analog sensor values.
- **BOOT Button:** Used to enter bootloader mode for flashing firmware.
- **RST Button:** Resets the ESP32-C6 chip.

4. SETUP INSTRUCTIONS

4.1 Connecting the Board

1. Connect the ESP32-C6 Mini Development Board to your computer using a USB Type-C cable.
2. Ensure the cable is capable of data transfer, not just charging.
3. The board should power on, indicated by a power LED (if present).

4.2 Driver Installation

In most modern operating systems (Windows 10/11, macOS, Linux), the necessary USB-to-Serial drivers for the ESP32-C6 are automatically installed. If the board is not recognized, you may need to manually install drivers for the USB-to-Serial chip (e.g., CH340 or CP210x, depending on the specific board revision). Refer to the manufacturer's website or common ESP32 documentation for driver downloads.

4.3 Development Environment Setup

The ESP32-C6 Mini supports various development environments, including the Arduino IDE and ESP-IDF.

4.3.1 Arduino IDE Setup

1. Download and install the Arduino IDE from the official Arduino website.
2. Open the Arduino IDE, go to **File > Preferences**.
3. In the "Additional Boards Manager URLs" field, add the ESP32 board manager URL (e.g., https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json).
4. Go to **Tools > Board > Boards Manager...** Search for "ESP32" and install the "esp32 by Espressif Systems" package.
5. Select your board: Go to **Tools > Board > ESP32 Arduino** and choose the appropriate ESP32-C6 board variant (e.g., "ESP32C6 Dev Module").
6. Select the correct COM port (Windows) or /dev/ttyUSBx (Linux/macOS) under **Tools > Port**.

4.3.2 ESP-IDF Setup

For advanced development, Espressif's IoT Development Framework (ESP-IDF) provides a robust environment. Follow the official ESP-IDF Getting Started Guide for detailed installation instructions for your operating system.

5. OPERATING INSTRUCTIONS

5.1 Uploading a Program

1. Ensure your development environment is set up and the board is connected.
2. In Arduino IDE, open an example sketch (e.g., **File > Examples > 01.Basics > Blink**).
3. Click the "Upload" button (right arrow icon).
4. If the upload fails, you may need to manually put the board into bootloader mode:
 - Press and hold the **BOOT** button.
 - While holding **BOOT**, press and release the **RST** button.
 - Release the **BOOT** button. The board is now in flashing mode.
 - Attempt to upload the sketch again.

5.2 Using Wireless Connectivity (WiFi & Bluetooth)

The ESP32-C6 supports both WiFi 6 and Bluetooth 5 LE. Libraries for these functionalities are available in both Arduino IDE (e.g., WiFi.h, BluetoothSerial.h) and ESP-IDF.

- **WiFi:** Use the WiFi library to connect to access points, create soft APs, or implement web servers.
- **Bluetooth LE:** Utilize the Bluetooth Low Energy capabilities for low-power communication with other BLE devices.

Refer to the extensive documentation and examples provided by Espressif and the Arduino community for specific code implementations.

6. MAINTENANCE

6.1 Handling Precautions

- Avoid touching the board's components unnecessarily, especially when powered.
- Handle the board by its edges to prevent damage from electrostatic discharge (ESD).
- Do not expose the board to extreme temperatures, humidity, or direct sunlight.

6.2 Storage

When not in use, store the development board in an anti-static bag or a protective enclosure to prevent dust accumulation and ESD damage.

6.3 Cleaning

If cleaning is necessary, use a soft, dry brush or compressed air to remove dust. For stubborn dirt, use isopropyl alcohol and a cotton swab, ensuring the board is unpowered and completely dry before re-applying power.

7. TROUBLESHOOTING

7.1 Board Not Detected by Computer

- **Check USB Cable:** Ensure the USB Type-C cable is a data cable, not just a charging cable. Try a different cable.
- **Check USB Port:** Try connecting to a different USB port on your computer.
- **Driver Installation:** Verify that the correct USB-to-Serial drivers are installed for your operating system.
- **Restart Computer:** Sometimes a simple restart can resolve connectivity issues.

7.2 Program Upload Errors

- **Incorrect Port/Board Selection:** Ensure the correct COM port (or /dev/ttyUSBx) and board type are selected in your IDE.
- **Bootloader Mode:** Manually put the board into bootloader mode by holding BOOT, pressing RST, then releasing RST and BOOT.
- **Cable Issues:** A faulty USB cable can cause upload failures. Try another cable.
- **Power Supply:** Ensure the board is receiving sufficient power.

7.3 WiFi/Bluetooth Connection Issues

- **Code Verification:** Double-check your code for correct SSID/password (for WiFi) or device addresses (for Bluetooth).
- **Antenna:** Ensure the onboard antenna is not obstructed.

- **Interference:** Minimize interference from other 2.4 GHz devices.
- **Power:** Insufficient power can affect wireless performance.

8. WARRANTY AND SUPPORT

This product is covered by a standard manufacturer's warranty. For specific warranty terms and conditions, please refer to the documentation provided with your purchase or contact AYWHP customer support.

For technical support, programming resources, and community forums, please refer to the extensive online resources available for ESP32 development boards, including the official Espressif documentation and the Arduino ESP32 community.