

ESP32 PRODUCT DETAILS

Features

- **Enhanced Connectivity:** Combines 2.4GHz Wi-Fi 6 (802.11ax), Bluetooth 5(LE), and IEEE 802.15.4 radio connectivity, allowing you to apply the Thread and Zigbee protocols.
- **Matter Native:** Supports building Matter-compliant smart home projects thanks to its enhanced connectivity, achieving interoperability
- **Security Encrypted on Chip:** Powered by ESP32-C6, it brings enhanced encrypted-on-chip security to your smart home projects via secure boot, encryption, and Trusted Execution Environment (TEE)
- **Outstanding RF performance:** Has an on-board antenna with up to 80m BLE/Wi-Fi range, while reserving an interface for external UFL antenna
- **Leveraging Power Consumption:** Comes with 4 working modes, with the lowest being 15 μ A in deep sleep mode, while also supporting lithium battery charge management.
- **Dual RISC-V Processors:** Incorporates two 32-bit RISC-V processors, with the high-performance processor running up to 160 MHz, and the low-power processor clocking up to 20
- **Classic XIAO Designs:** Remains the classic XIAO designs of the thumb-size form factor of 21 x 17.5mm, and single-sided mount, making it perfect for space-limited projects such as wearables

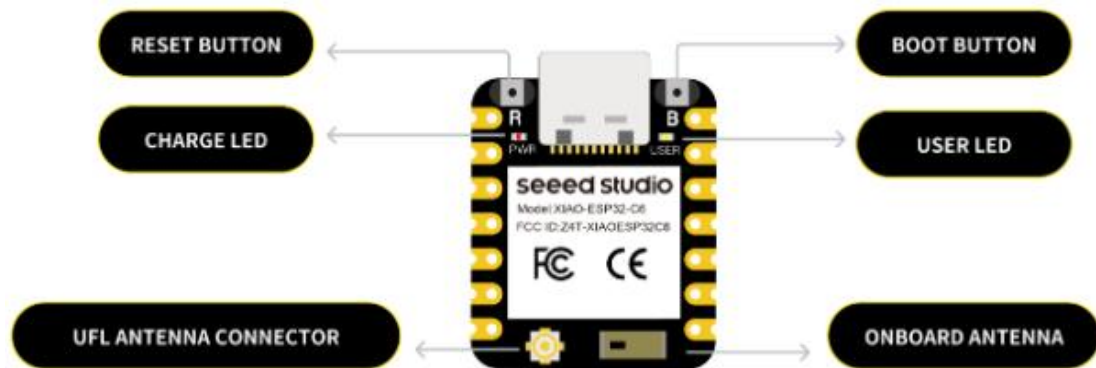


Description

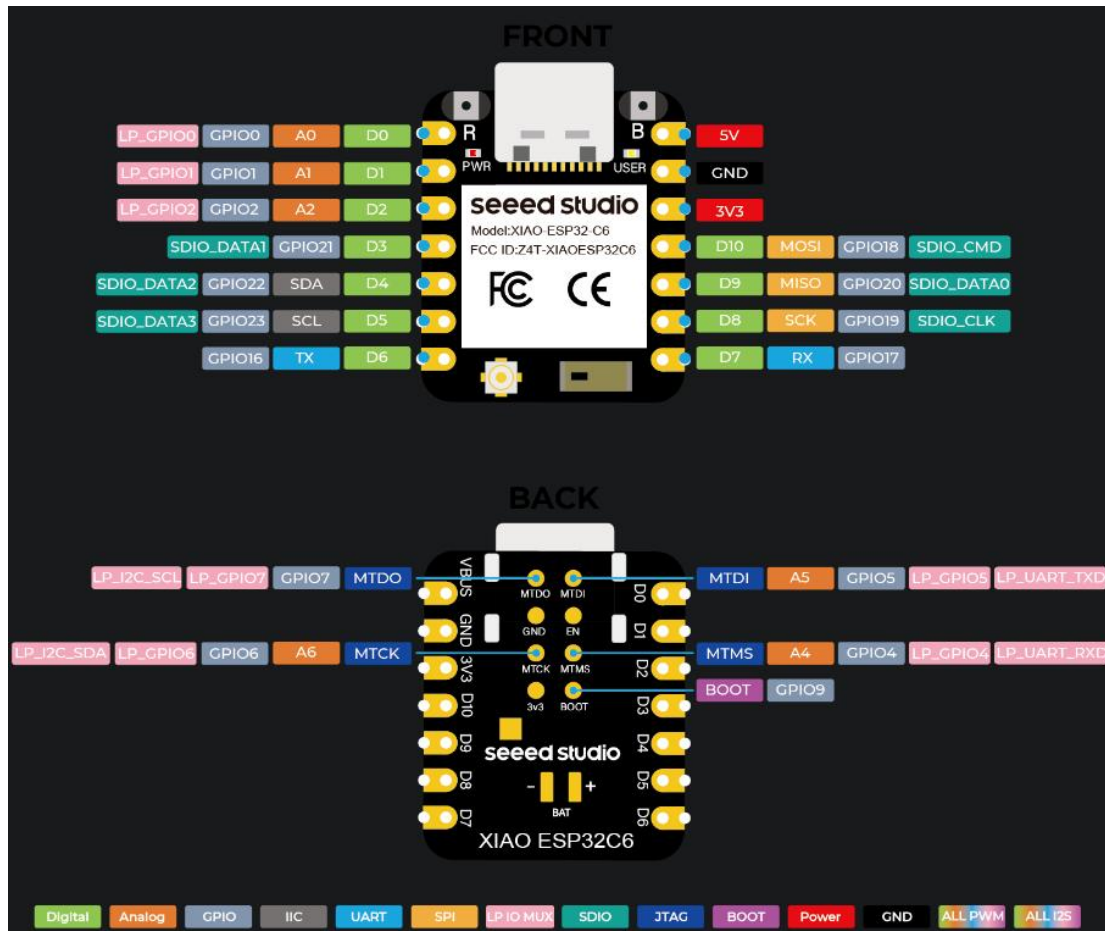
Seeed Studio XIAO ESP32C6 is powered by the highly-integrated ESP32-C6 SoC, built on two 32-bit RISC-V processors, with a high-performance (HP) processor with running up to 160 MHz, and a low-power (LP) 32-bit RISC-V processor, which can be clocked up to 20 MHz. There are 512KB SRAM and 4 MB Flash on the chip, allowing for more programming space, and bringing more possibilities to the IoT control scenarios.

XIAO ESP32C6 is Matter native thanks to its enhanced wireless connectivity. The wireless stack supports 2.4 GHz WiFi 6, Bluetooth® 5.3, Zigbee, and Thread (802.15.4). As the first XIAO member compatible with Thread, it's a perfect fit for building Matter-compliant projects, thus achieving interoperability in smart-home.

To better support your IoT projects, XIAO ESP32C6 not only provides seamless integration with mainstream cloud platforms like ESP Rain Maker, AWS IoT, Microsoft Azure, and Google Cloud, but also leverages security for your IoT applications. With its on-chip secure boot, flash encryption, identity protection, and Trusted Execution Environment (TEE), this tiny board ensures the desired level of security for developers looking to build smart, secure, and connected solutions.



This new XIAO is equipped with a high-performance onboard ceramic antenna with up to 80m BLE/Wi-Fi range, while it also reserves an interface for an external UFL antenna. At the same time, it also comes with an optimized power consumption management. Featuring four power modes and an onboard lithium battery charging management circuit, it works in the Deep Sleep mode with a current as low as 15 μ A, making it an excellent fit for remote, battery-powered applications.



Being the 8th member of the Seeed Studio XIAO family, XIAO ESP32C6 remains the classic XIAO design. It is designed to fit the 21 x 17.5mm, XIAO Standard Size, while remains its classic single-sided components mounting. Even being thumb-sized, it amazingly breaks out 15 total GPIO pins, including 11 digital I/Os for PWM pins and 4 analog I/Os for ADC pins. It supports UART, IIC, and SPI serial communication ports. All these features make it a perfect fit for either space-limited projects such as wearables, or a production-ready unit for your PCBA designs.

Getting started

First, we are going to connect XIAO ESP32C3 to the computer, connect an LED to the board and upload a simple code from Arduino IDE to check whether the board is functioning well by blinking the connected LED.

Hardware setup

You need to prepare the following:

- 1 x [Seeed Studio XIAO ESP32C6](#)
- 1 x Computer
- 1 x USB Type-C cable

Tip

Some USB cables can only supply power and cannot transfer data. If you don't have a USB cable or don't know if your USB cable can transmit data, you can check [Seeed USB Type-C support USB 3.1](#).

- **Step 1.** Connect XIAO ESP32C6 to your computer via a USB Type-C cable.
- **Step 2.** Connect an LED to D10 pin as follows
- Note: Make sure to connect a resistor (about 150Ω) in series to limit the current through the LED and to prevent excess current that can burn out the LED

Prepare the Software

Below I will list the system version, ESP-IDF version, and ESP-Matter version used in this article for reference. This is a stable version that has been tested to work properly.

- Host: Ubuntu 22.04 LTS (Jammy Jellyfish).
- ESP-IDF: Tags v5.2.1.
- ESP-Matter: main branch, as of 10 May 2024, commit bf56832.
- connectedhomeip: currently works with commit 13ab158f10, as of 10 May 2024.
- Git
- Visual Studio Code

Installation ESP-Matter Step by Step

Step 1. Install Dependencies

First, you need to install the required packages using `apt-get`. Open your terminal and execute the following command:

```
sudo apt-get install git gcc g++ pkg-config libssl-dev libdbus-1-dev \
    libglib2.0-dev libavahi-client-dev ninja-build python3-venv python3-dev \
    python3-pip unzip libgirepository1.0-dev libcairo2-dev libreadline-dev
```

This command installs various packages like `git`, compilers (`gcc`, `g++`), and libraries needed for building and running the Matter SDK.

Step 2. Clone the ESP-Matter Repository

Clone the repository from GitHub using the command with a depth of 1 to fetch only the latest snapshot:

```
cd ~/esp
git clone --depth 1 https://github.com/espressif/esp-matter.git
```

Change into the directory and initialize the required Git submodules:

```
cd esp-matter
git submodule update --init --depth 1
```

Navigate to the directory and run a Python script to manage submodules for specific platforms:connectedhomeip

```
cd ./connectedhomeip/connectedhomeip
./scripts/checkout_submodules.py --platform esp32 linux --shallow
```

This script updates submodules for both ESP32 and Linux platforms in a shallow manner (latest commit only).

Step 3. Install ESP-Matter

Return to the root directory, then run the installation script:esp-matter

```
cd ../../
./install.sh
```

This script will install additional dependencies specific to the ESP-Matter SDK.

Step 4. Set Environment Variables

Source the script to set up the environment variables needed for development:export.sh

```
source ./export.sh
```

This command configures your shell with necessary environment paths and variables.

Step 5 (Optional). Quick access to the ESP-Matter development environment

To add the provided aliases and environment variable settings to your file, follow these steps. This will configure your shell environment to easily switch between IDF and Matter development setups, and enable ccache for faster builds..bashrc

Open your terminal and use a text editor to open the file located in your home directory. You can use or any editor you prefer. For example:.bashrcnano

```
nano ~/.bashrc
```

Scroll to the bottom of the file and add the following lines:.bashrc

```
# Alias for setting up the ESP-Matter environment
alias get_matter='. ~/esp/esp-matter/export.sh'

# Enable ccache to speed up compilation
alias set_cache='export IDF_CCACHE_ENABLE=1'
```

After adding the lines, save the file and exit the text editor. If you're using `nano`, you can save by pressing `Ctrl+O`, hit to confirm, and then `Ctrl+X` to exit.

For the changes to take effect, you need to reload the file. You can do this by sourcing the file or closing and reopening your terminal. To source the file, use the following

```
source ~/.bashrc
```

command: `.bashrc.bashrc.bashrc`

Now you can run `get_matter` and `set_cache` to set up or refresh the esp-matter environment in any terminal session.

```
get_matter
set_cache
```

Application

- Secure and Connected Smart Home, enhancing everyday life through automation, remote control, and more.
- Space-limited and Battery-Powered Wearables, thanks to their thumb-size and low-power consumption.
- Wireless IoT Scenarios, enabling rapid, reliable data transmission.

Declaration here

- The device does not support BT hopping operation under DSS mode.

FCC

FCC Statement:

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference, and
- (2) this device must accept any interference received, including interference that may cause

undesired operation.

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

FCC Radiation Exposure Statement

This modular complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter. This modular must be installed and operated with a minimum distance of 20 cm between the radiator and user body.

The module is limited to OEM installation only

The OEM integrator is responsible for ensuring that the end-user has no manual instructions to remove or install module

If the FCC identification number is not visible when the module is installed inside another device, then the outside of the device into which the module is installed must also display a label referring to the enclosed module. This exterior label can use wording such as the following: "Contains Transmitter Module FCC ID: Z4T-XIAOESP32C6 Or Contains FCC ID: Z4T-XIAOESP32C6"

When the module is installed inside another device, the user manual of the host must contain below warning statements;

1. This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference.
- (2) This device must accept any interference received, including interference that may cause undesired operation.

2. Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

The devices must be installed and used in strict accordance with the manufacturer's instructions as described in the user documentation that comes with the product.

Any company of the host device which install this modular with limit modular approval should perform the test of radiated emission and spurious emission according to FCC part 15C : 15.247 requirement, Only if the test result comply with FCC part 15C : 15.247 requirement, then the host

can be sold legally.

Antennas:

Type	Gain
Ceramic chip antenna	4.97dBi
FPC antenna	1.23dBi
Rod antenna	2.42dBi

The antenna is permanently attached, can't be replaced. Choose whether to use the built-in ceramic antenna or external antenna through GPIO14. Send 0 to GPIO14 to use the built-in antenna, and send 1 to use the external antennaTrace antenna designs: Not applicable.