

# ESP32-CAM

ESP32-CAM-MB WIFI bluetooth camera development board module



## Catalogue

|                            |   |
|----------------------------|---|
| Products .....             | 1 |
| Main Features .....        | 2 |
| .....                      | 3 |
| Product Parameters         |   |
| Instructions for use ..... | 4 |

# Products

The ESP32-CAM-MB WiFi Bluetooth Development Board is a multi-functional development board with an integrated ESP32 chip and camera module for IoT projects, especially those applications that require image capture and transmission.

The ESP32-CAM development board, features an ESP32-S chip, an OV2640 camera, micro SD card slot and several GPIOs for connecting peripherals. The module is a small-sized camera module that can work independently as the smallest system. The new WiFi+Bluetooth dual-mode development board designed based on ESP32 features an on-board PCB antenna, two high-performance 32-bit LX6 CPUs with 7-stage pipeline architecture, and an adjustable main frequency range of 80MHz to 240Mhz. ESP32-CAM is an 802.11b/g/n Wi-Fi + BT/BLE SoC module, with ultra-low power consumption, deep ESP32-CAM is an 802.11b/g/n Wi-Fi + BT/BLE SoC module with ultra-low power consumption and deep sleep current as low as 6mA, which makes it suitable for IoT applications with high power requirements. ESP32-CAM is a miniature module with camera function, equipped with OV2640 camera, GPIO for connecting peripherals, and micro-SD card for storing captured images, which can be directly plugged into the backplane.

As an IoT camera module based on the ESP32 chip, the ESP32-CAM-MB combines the functions of a microcontroller unit (MCU) and an image sensor, and is suitable for a wide range of application scenarios that require image capture and wireless transmission.

It can be widely used in a variety of IoT occasions, suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications, it is an ideal solution for IoT applications.

# Performance

Interface: Micro USB

Processor: Dual-core 32-bit LX6 microprocessor

Main Frequency: Up to 240 MHz

Computing power: up to 600 DMIPS

SPI Flash: 32mbit by default

Internal SRAM: 520 KB

External PSRAM: 4 MB/8 MB

Wi-Fi: 802.11b / G / n / e / i

Bluetooth: Bluetooth 4.2BR/EDR and BLE standards

Interface support (2Mbps): UART, SPI, I2C, PWM

TF card support: Maximum 4G

IO ports: 9

Serial Port Rate: Default 115200bps

Spectrum range: 2400 ~ 2483.5 MHz

Camera Sensor: OV2640 image sensor, 2MP

Image Output Format: JPEG (only support OV2640), BMP, GRAYSCALE

Transmit power:

802.11b:  $17 \pm 2$ dBm (@ 11mbps)

802.11g:  $14 \pm 2$ dBm (@ 54mbps)

802.11n:  $13 \pm 2$ dBm (@ MCS7)

Receive sensitivity:

CCK, 1mbps: -90dBm

CCK, 11mbps: -85dBm

6Mbps (1 / 2BPSK): -88dBm

54Mbps (3/464-QAM): -70dBm

MCS7 (65Mbps, 72.2 Mbps): -67dBm

# Power Consumption

Flash off: 180mA @ 5V

Flash on and brightness set to max: 310mA @ 5V.

Deep sleep: up to 6mA @ 5V minimum.

Moderm sleep: 20mA @ 5V minimum.

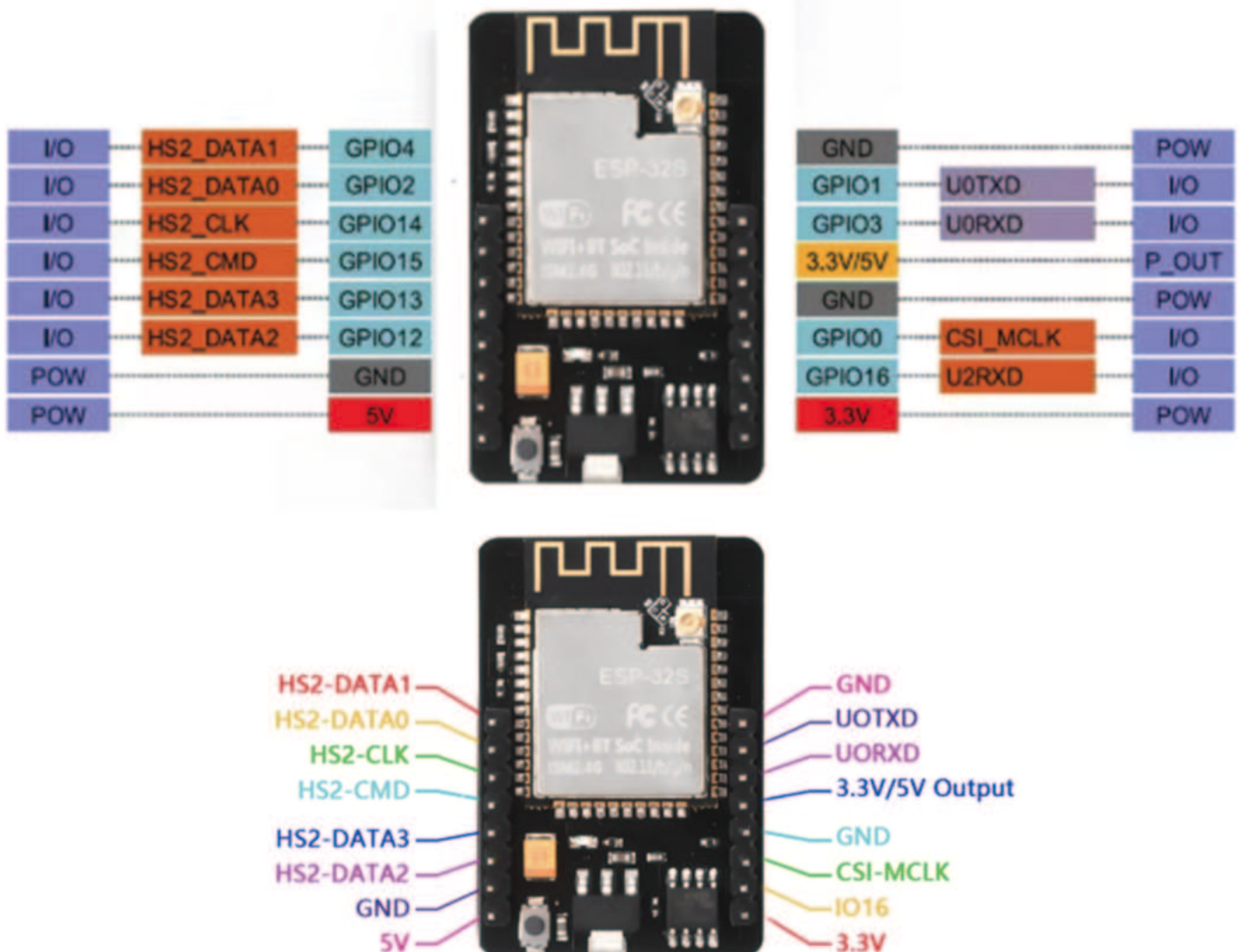
Light sleep: up to 6.7mA @ 5V minimum.

Security: VPA/VPA2/VPA2-Enterprise/VPS

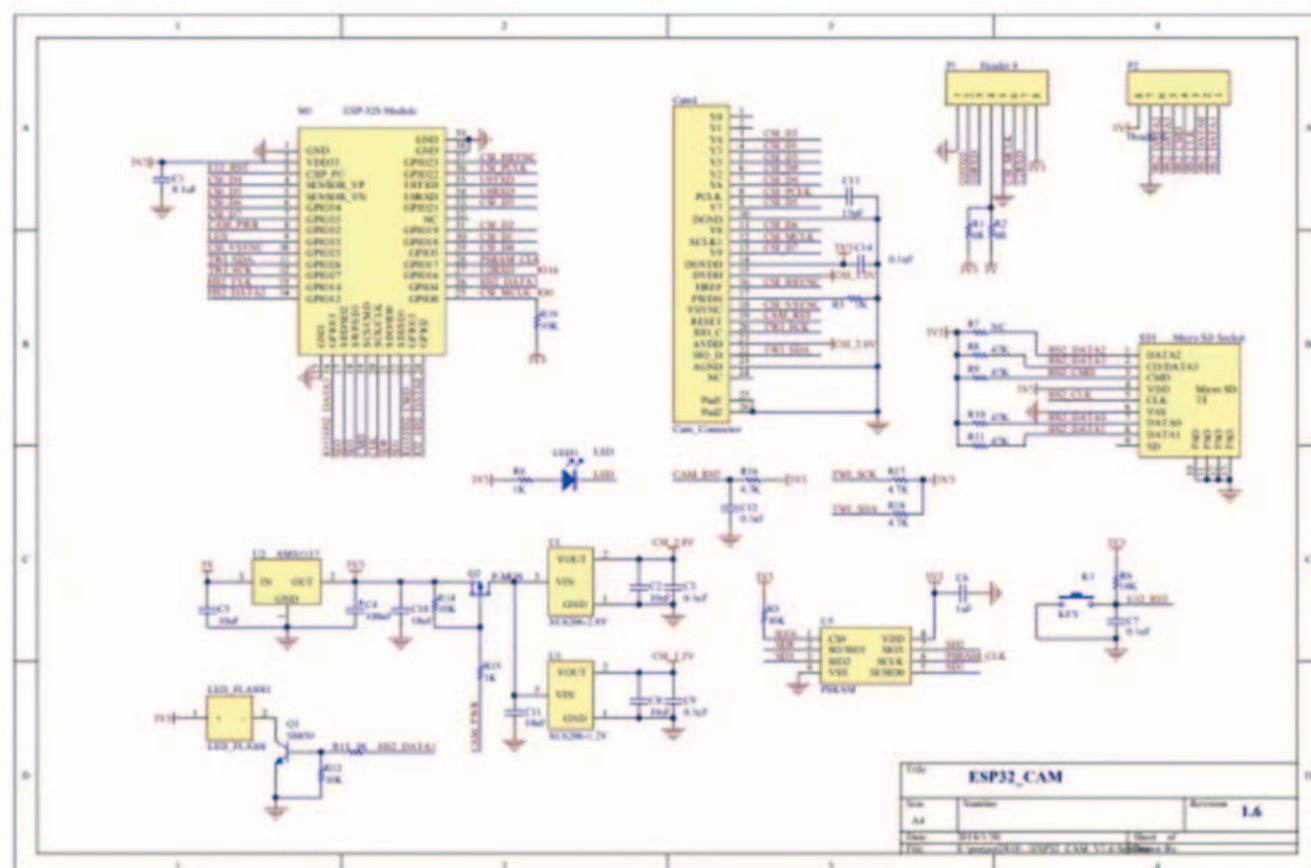
Operating temperature: -20°C ~ 70

Storage environment: -40°C ~ 125°C, <90% RH

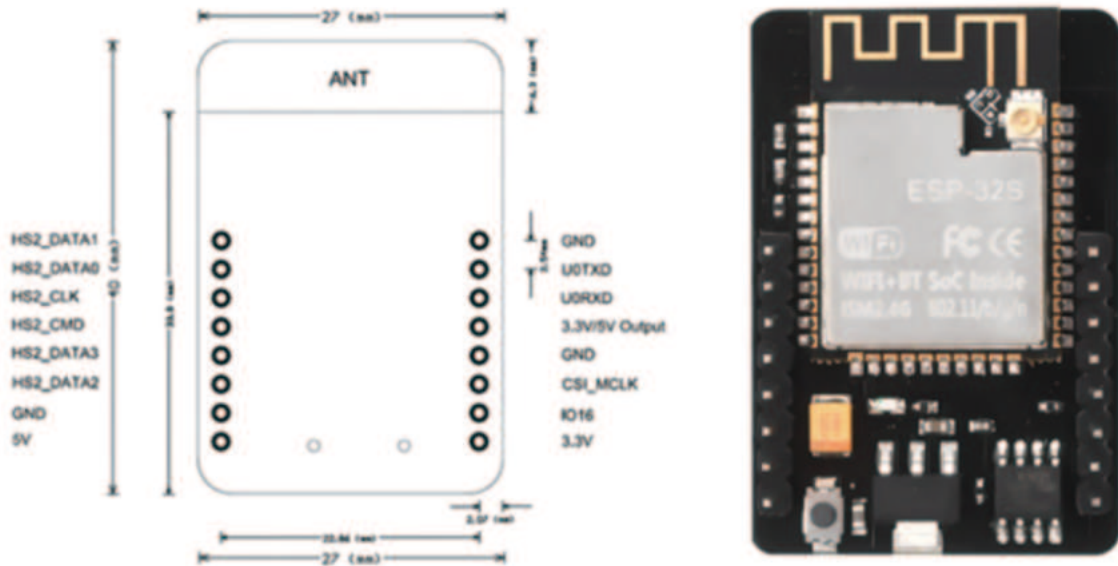
## Product Pinout



| CAM       | ESP32 | SD          | ESP32 |
|-----------|-------|-------------|-------|
| D0        | PIN5  | CLK         | PIN14 |
| D1        | PIN18 | CMD         | PIN15 |
| D2        | PIN19 | DATA0       | PIN2  |
| D3        | PIN21 | DATA1/flash | PIN4  |
| D4        | PIN36 | DATA2       | PIN12 |
| D5        | PIN39 | DATA3       | PIN13 |
| D6        | PIN34 |             |       |
| D7        | PIN35 |             |       |
| XCLK      | PIN0  |             |       |
| PCLK      | PIN22 |             |       |
| VSYN      | PIN25 |             |       |
| HREF      | PIN23 |             |       |
| SDA       | PIN26 |             |       |
| SCL       | PIN27 |             |       |
| POWER PIN | PIN32 |             |       |



# Product Size



## Instructions for use

### Preparation Stage

1. Hardware preparation: Make sure you have the ESP32-CAM module, USB to TTL module (for data transfer), power adapter or battery, and necessary connecting cables such as Dupont cable.
2. Software preparation: Install Arduino IDE. If your version of Arduino IDE does not include ESP32 development support, you need to additionally install the ESP32 development board package. You can find the Manage IDE Packages option in the Arduino IDE Preferences, then search and install the 'ESP32' package.

### Connecting ESP32-CAM

1. Connect the ESP32-CAM to the USB to TTL module: Use a Dupont cable to connect the TX, RX, GND and VCC pins of the ESP32-CAM to the corresponding pins of the USB to TTL module.
2. Connect the power supply: ESP32-CAM needs 3.3V power supply. You can use the power output of the USB to TTL module or connect it directly to an external power adapter.

3. Connect to computer: Connect the USB to TTL module to your computer via USB cable.

## Arduino IDE Setup

1. Select development board: Under the 'Tools' menu of Arduino IDE, select the type of development board as 'AI Thinker ESP32-CAM'. 2. Select Serial Port: Again under the 'Tools' menu, select the correct serial port, which is usually the USB port you connect the ESP32-CAM to.

## Code Configuration

```
// -----  
// #define CAMERA_MODEL_WROVER_KIT // Has PSRAM  
// #define CAMERA_MODEL_ESP_EYE // Has PSRAM  
// #define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM  
// #define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM  
// #define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM  
// #define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM  
// #define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM  
// #define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM  
#define CAMERA_MODEL_AI_THINKER // Has PSRAM  
// #define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM  
// #define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM  
// ** Espressif Internal Boards **  
// #define CAMERA_MODEL_ESP32_CAM_BOARD  
// #define CAMERA_MODEL_ESP32S2_CAM_BOARD  
// #define CAMERA_MODEL_ESP32S3_CAM_LCD  
// #define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM  
// #define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM  
#include "camera_pins.h"
```

## Burning Sample Programs

Select Examples: Arduino IDE comes with some examples, you can select 'File' -> 'Examples' -> 'ESP32' -> 'CameraWebServer' in 'WebCam'.

**Upload Code:** After confirming the hardware connection and IDE settings are correct, click the Upload button to upload the sample program to ESP32-CAM.

### Test ESP32-CAM

**Find the IP address of ESP32-CAM:** After the upload is completed, use the serial monitor of Arduino IDE to find the IP address assigned by ESP32-CAM.

**Access the web interface:** Enter the IP address of the ESP32-CAM into your computer's web browser, and you should see a simple web interface showing live images captured by the camera.

**Extended Functions** Once you have mastered the basics, you can start exploring more advanced features such as face detection, object recognition, cloud storage, mobile device control, etc.

### **Caution**

- Make sure that the IO0 pin of the ESP32-CAM is not pulled low before burning the code, otherwise the development board will enter Bootloader mode and cannot upload the code properly.
- When connecting and disconnecting the ESP32-CAM, be sure to disconnect the power supply first to avoid damaging the circuit.
- When the upload fails or the ESP32-CAM does not respond, check that the hardware connections are correct and that the correct development board and serial port are selected.

Keep in mind that the ESP32-CAM-MB may have its own specific hardware features or firmware versions, so the above steps may need to be adapted to the specific product manual. If you encounter specific problems during use, consulting the official documentation or online forums can be very helpful.

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.
- This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.