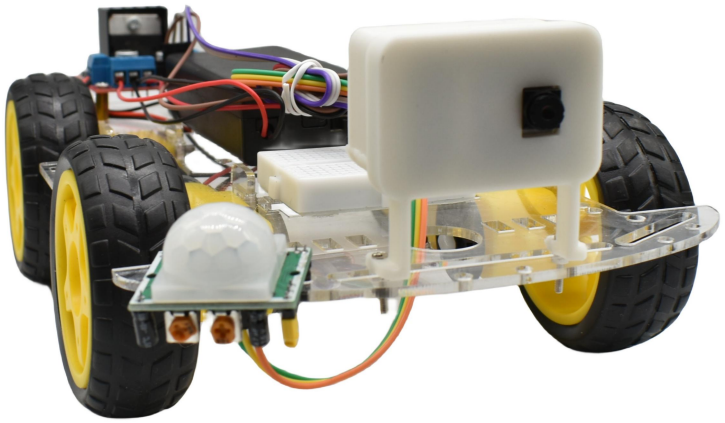
LROBRUYA

ESP32 CAM Car Kit



Content

Preface

Company Profile

Founded in 2014, Shenzhen Lonten Technology Co., Ltd. focuses on the design, research production of Electronics Module for robotics related products. Consisting of professional researchers and skilled engineers, our R&D team constantly strives for creative function and excellent user experience. The company's R&D investments on arduino kits raspberry pi kits, as well as 3D printer and robots that back up STEAM education.

Customer Service

Our self-owned factory is certificated with BSCI and SO, covering an area of 5,000 square meters, and achieving an annual production capacity of over 10,000 units. Our products are all certified to CE, FCC, and ROHS standards, have exported to more than 100 countries including, but not limited to France, the United States of America, Australia, Russia, the United Kingdom, Germany, Singapore, Egypt, and India, bringing technological innovation to all walks of life.

How to Install Arduino IDE

Introduction

The Arduino Integrated Development Environment (IDE) is the software side of the Arduino platform.

In this Project, you will learn how to setup your computer to use Arduino and how to set about the Projects that follow.

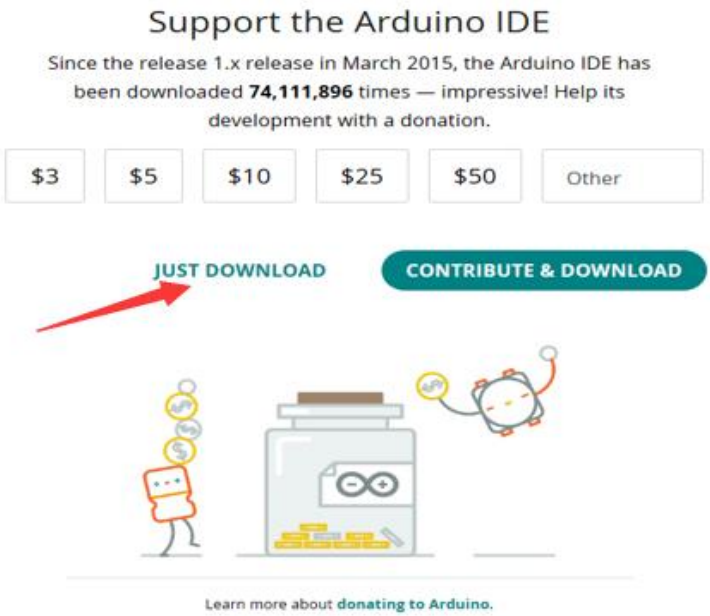
The Arduino software that you will use to program your Arduino is available for Windows, Mac and Linux. The installation process is different for all three platforms and unfortunately there is a certain amount of manual work to install the software.

STEP 1: Go to https://www.arduino.cc/en/software.

The version available at this website is usually the latest version, and the actual version may be newer than the version in the picture.

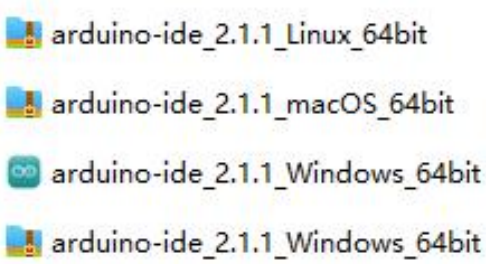
STEP2： Download the development software that is compatible with the operating. system of your computer. Take Windows as an example here.

Click Windows Win 10 and newer,64 bits.



Click JUST DOWNLOAD.

Also version 2.1.1 is available in the material we provided, and the versions of our materials are the latest versions when this course was made.

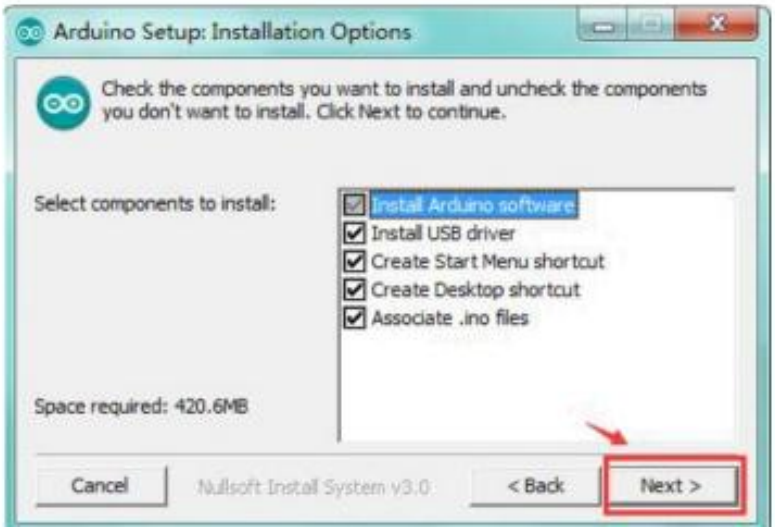


Installing Arduino (Windows)

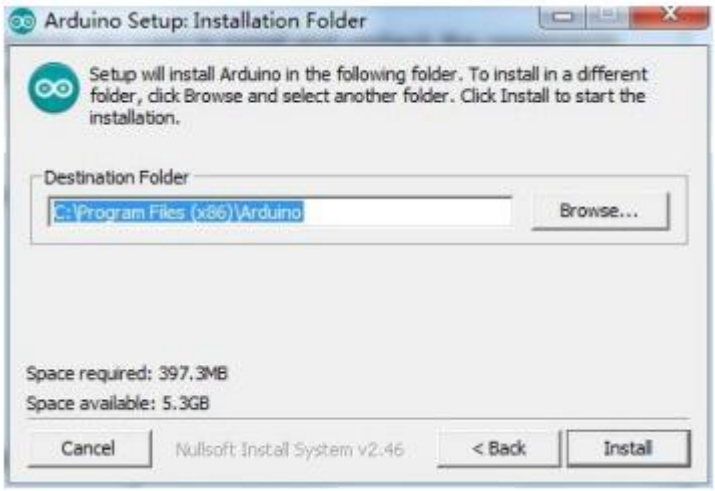
Install Arduino with the exe. Installation package.

arduino-ide_2.1.1_Windows_64bit

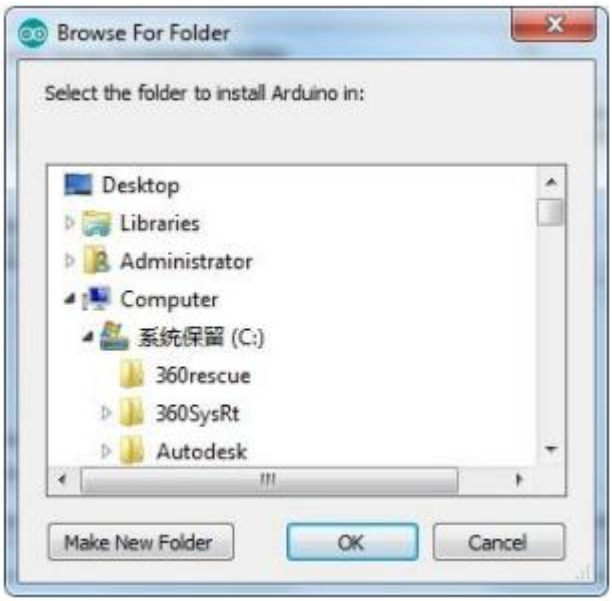
Click I Agree to see the following interface.



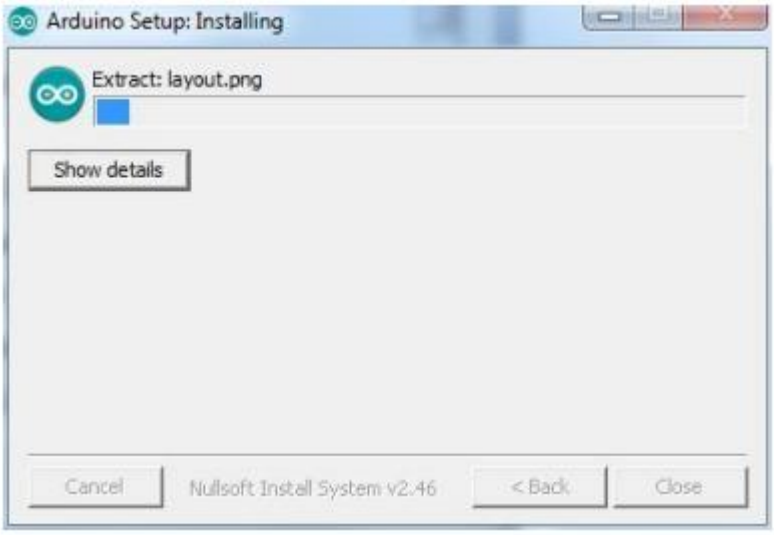
Click Next



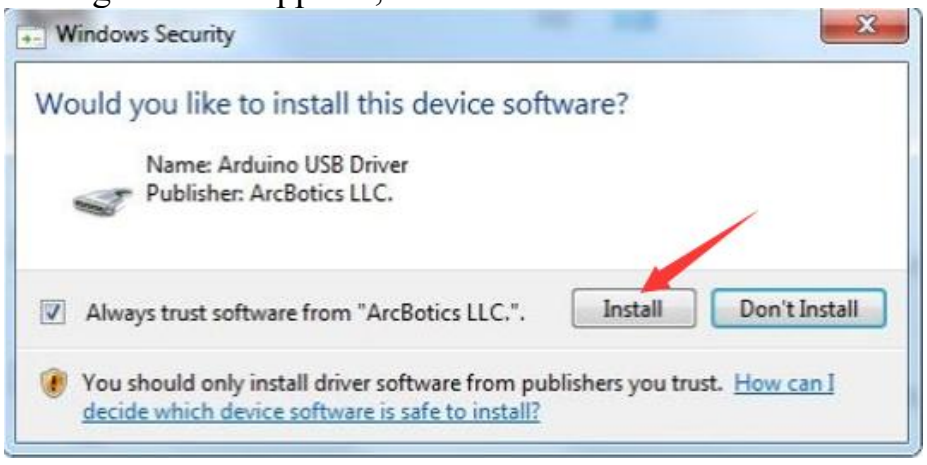
You can press Browse… to choose an installation path or directly type in the directory you want.



Click Install to initiate installation

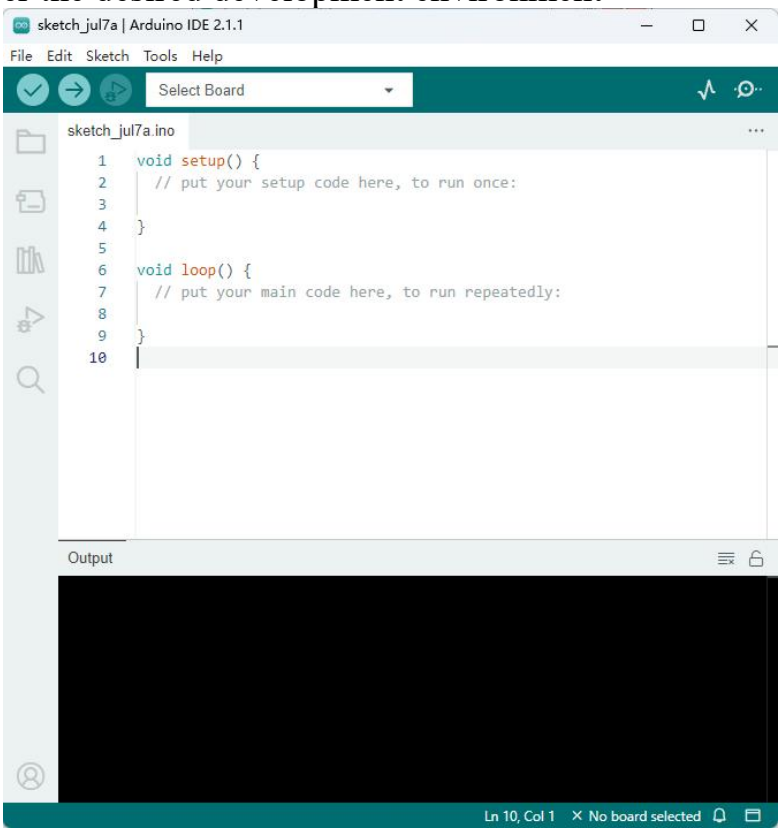


Finally, the following interface appears, click Install to finish the installation.



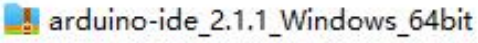
Next, the following icon appears on the desktop

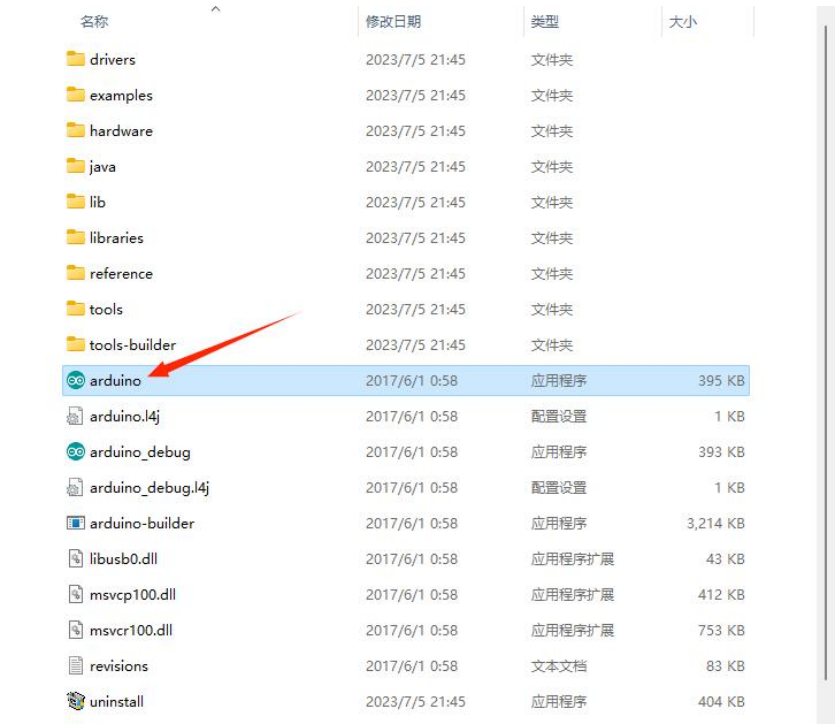
Double-click to enter the desired development environment

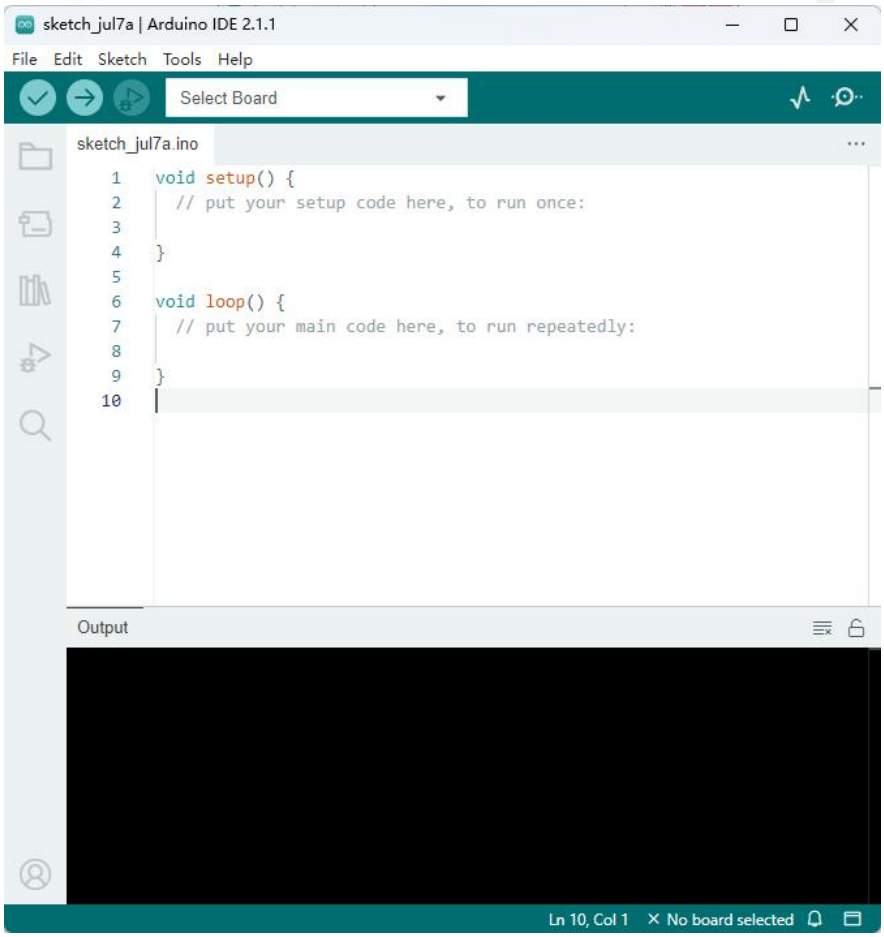


You may directly choose the installation package for installation and skip the contents below and jump to the next section. But if you want to learn some methods other than the installation package, please continue to read the section.

Unzip the zip file downloaded, Double-click to open the program and enter the desired development environment.

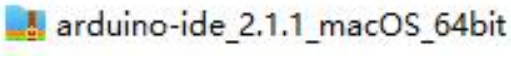






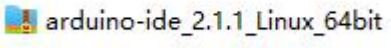
Installing Arduino (Mac OS X)

Download and Unzip the zip file, double click the Arduino.app to enter Arduino IDE; the system will ask you to install Java runtime library if you don't have it in your computer. Once the installation is complete you can run the Arduino IDE.



Installing Arduino (Linux)

You will have to use the make install command. If you are using the Ubuntu system, it is recommended to install Arduino IDE from the software center of Ubuntu.

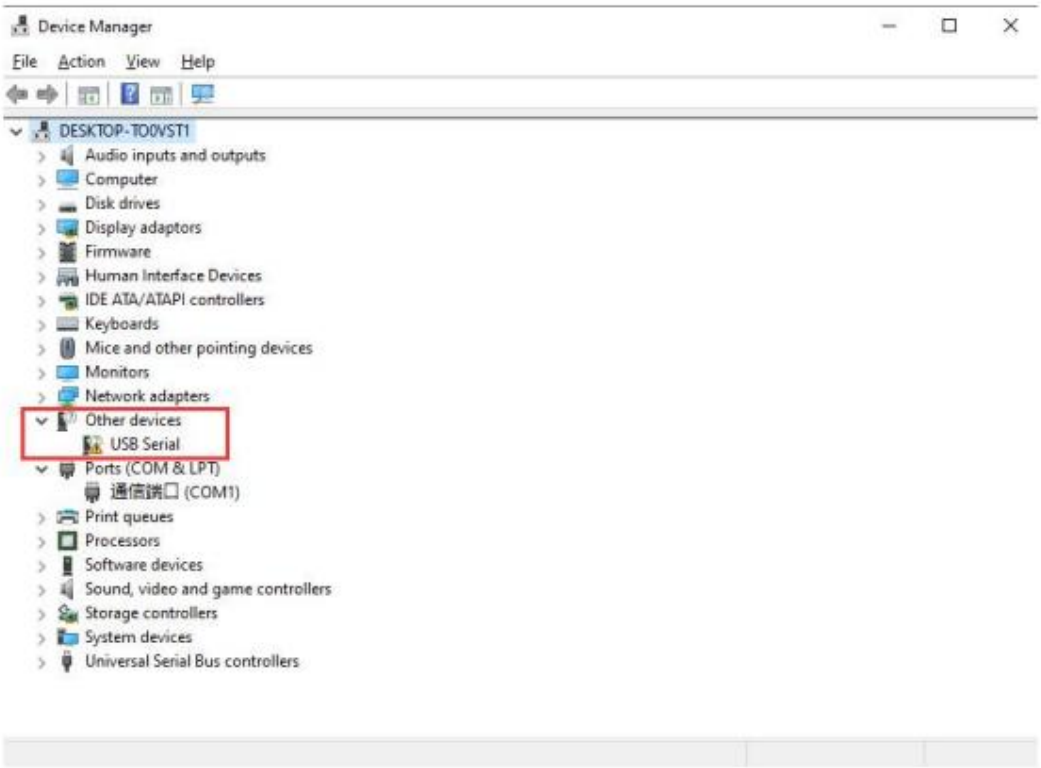


How to Install Arduino Driver

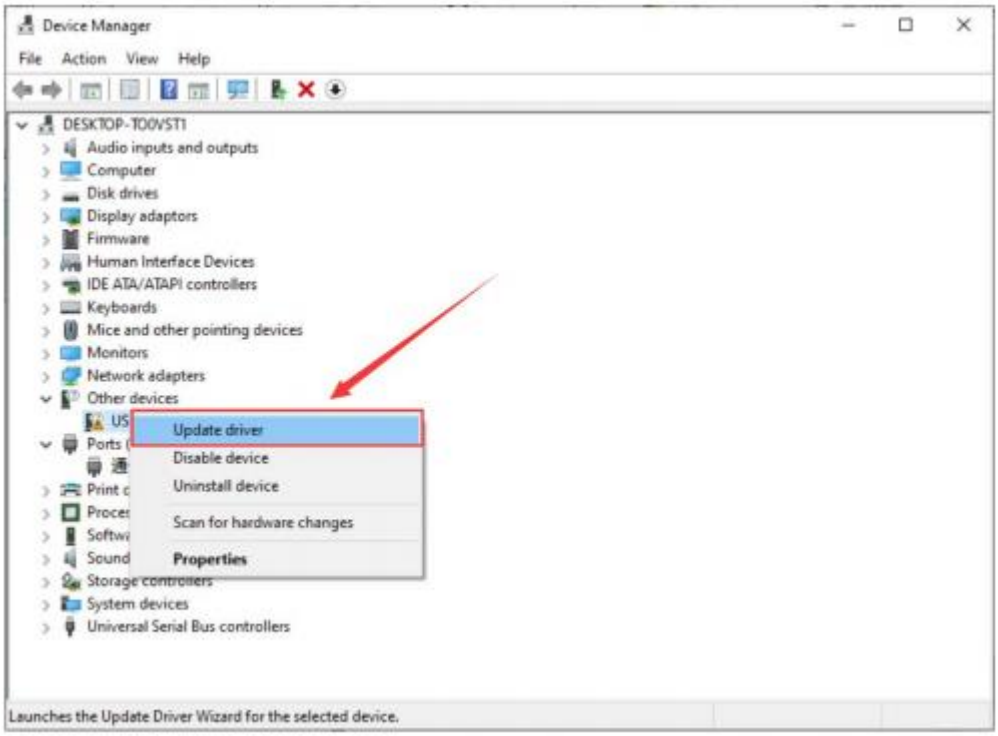
Using the ESP32-CAM-MB module makes it easier to upload code for the ESP32-CAM. The serial port conversion chip of
MB is CH340. So before uploading the code, you need to install the CH340 driver. Otherwise you won't be able to find the
correct COM port in the Arduino IDE.If your computer has already installed the CH340 driver, you can skip this step.

In different systems, the driver installation is similar. Here we start to install the driver on the Win10 system. You can find the “USB_Drive_CH341_3_1” folder in the information we provide, this is the driver file we want to install.

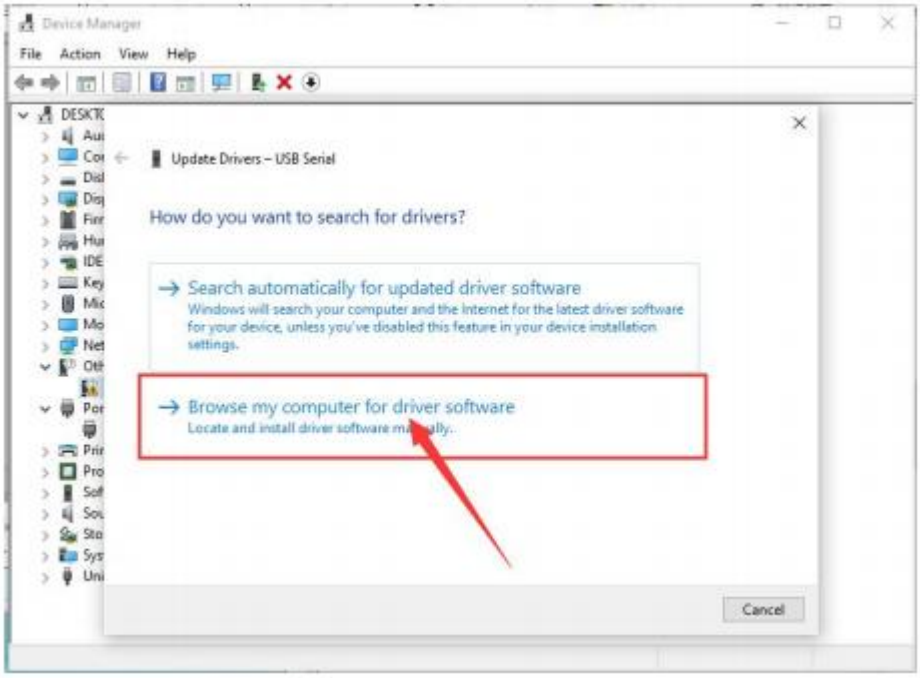
Plug one end of your USB cable into the Arduino UNO R3 Board and the other into a USB socket on your computer.
When you connect the Arduino UNOR3 Board to your computer at the first time, right click your “My Computer”—>for “Properties”—>click the “Device manager”, under Other devices, you should see the “USB-Serial” or “Unknown device ”.Or you can search for "devi" in your computer, or you can open the device manager of your computer.



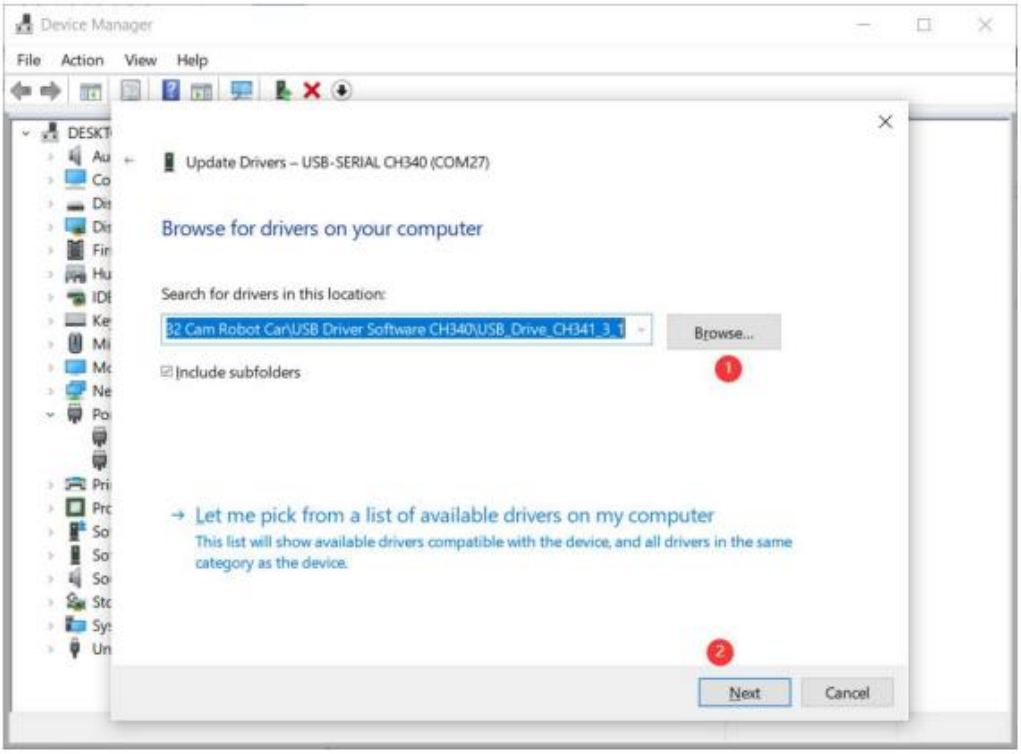
Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.



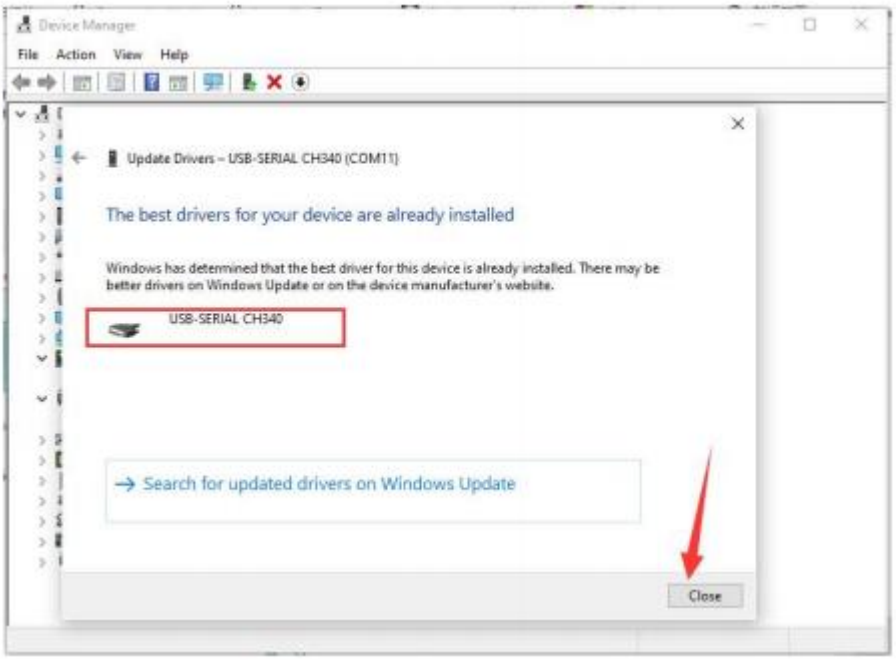
Then it will be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.



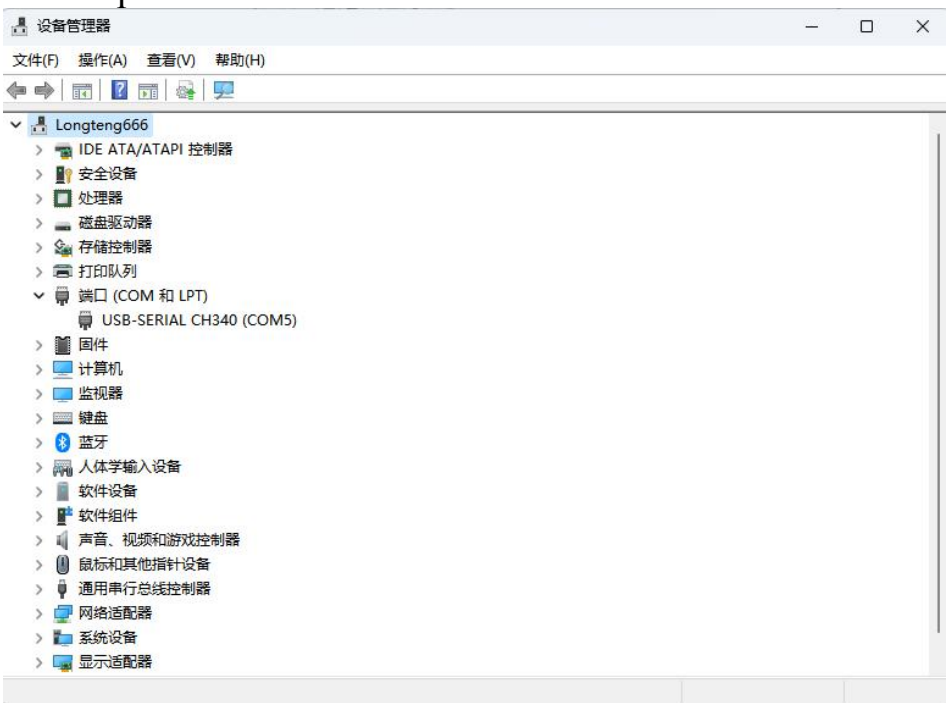
After that, select the browse option and navigate to the drive folder "USB_Drive_CH341_3_1", which can be found in the information we provide.



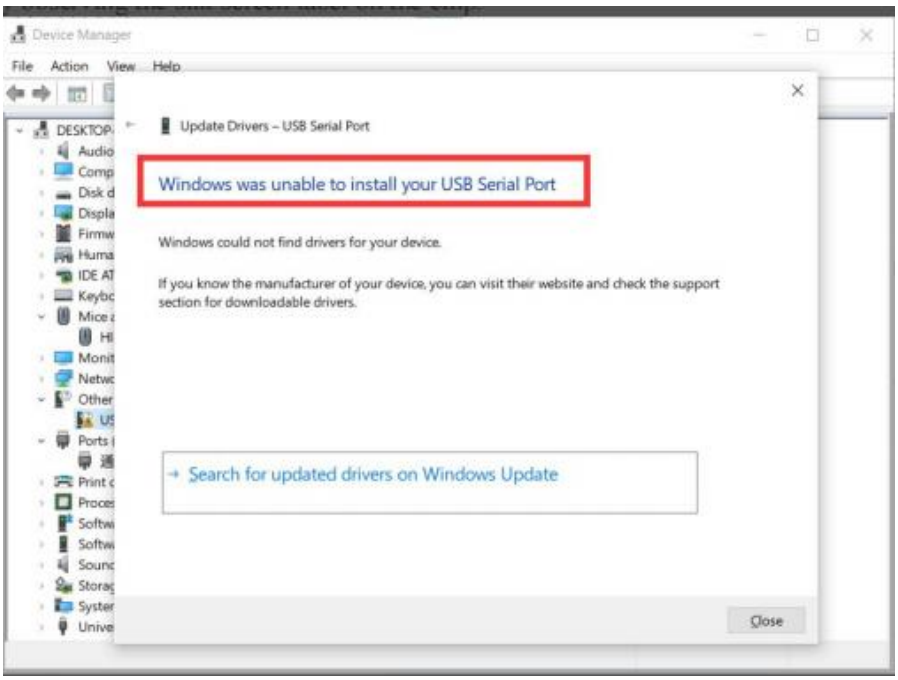
Once the software has been installed, you will get a confirmation message. Installation completed, click “Close”.



Up to now, the driver is installed well. Then you can right click “My Computer”—>for “Properties”—>click the “Device manager”, you should see the device as the figure shown below. Or you can search for "devi" in your computer, or you can open the device manager of your computer.



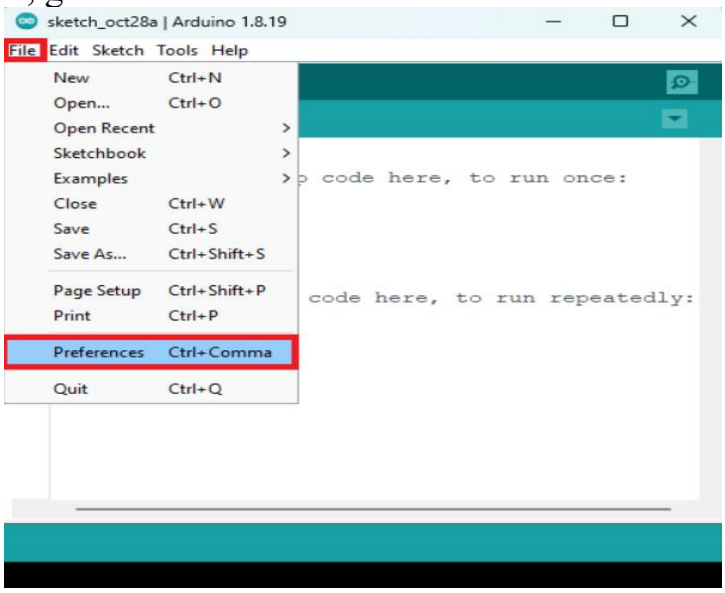
Note:If there is an error message: maybe the serial port conversion chip model of ESP32 CAM MB is "FT232RL", you can
refer to Install or update FTDI drivers. You can judge whether the serial port conversion chip is "CH340G" or "FT232RL"
by observing the silk screen label on the chip.



How to Install ESP32 Add-on in Arduino IDE

To install the ESP32 board in your Arduino IDE, follow these next instructions:

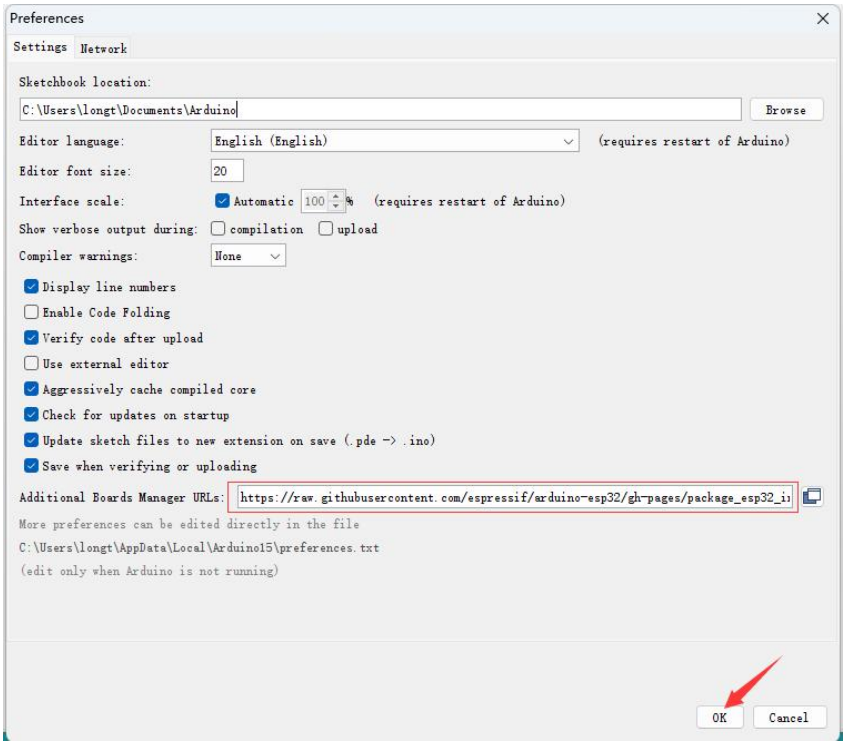
1.In your Arduino IDE, go to File> Preferences



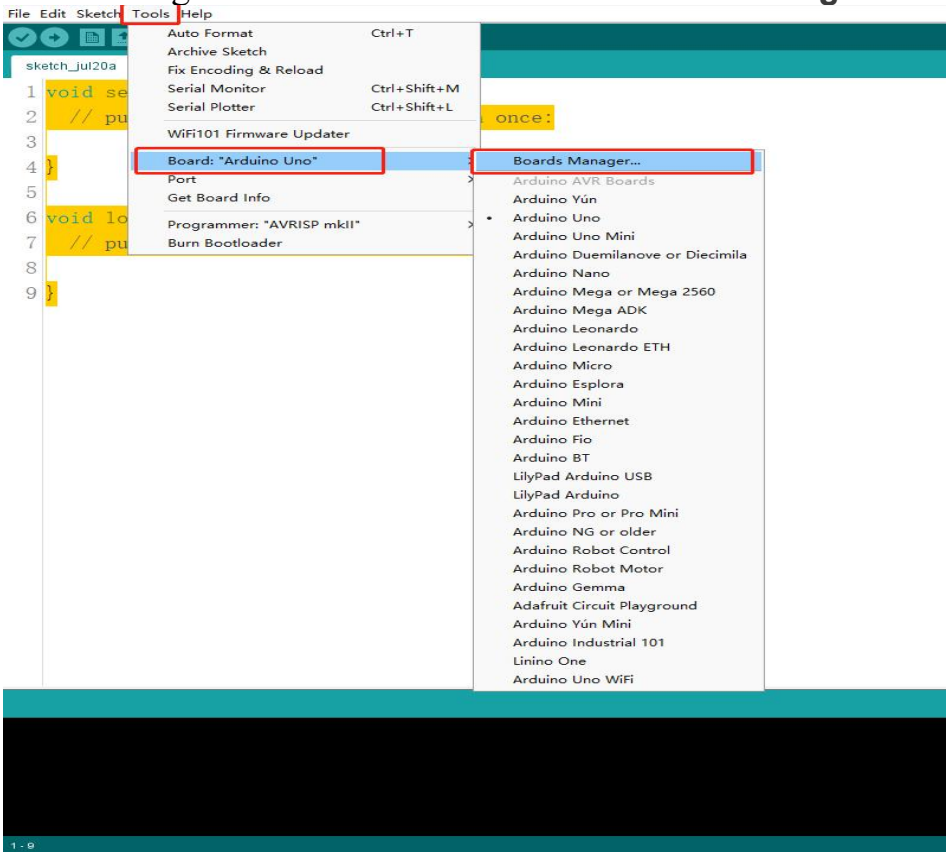
2.Enter the following into the “Additional Board Manager URLs” field:

`https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json`

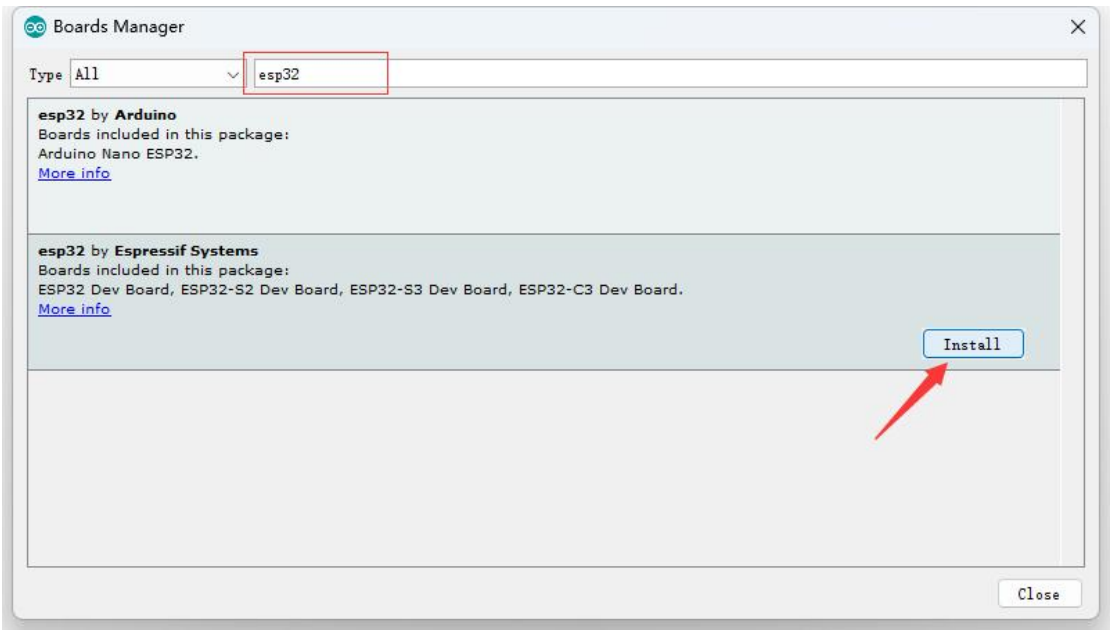
Then, click the “OK” button:



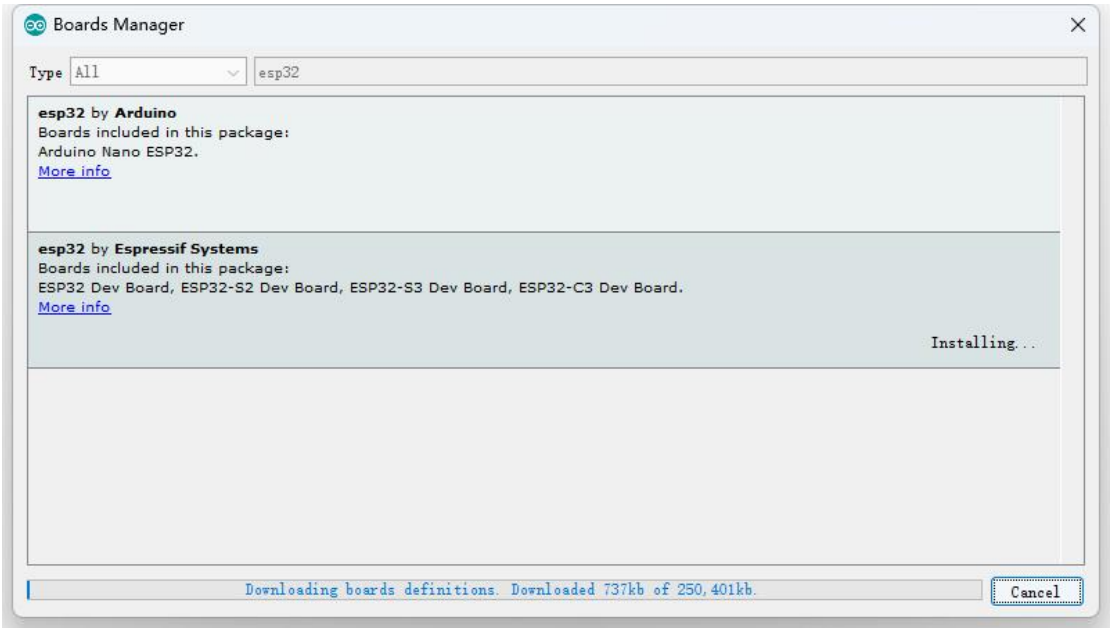
3.Open the Boards Manager. Go to **Tools > Board > Boards Manager…**



4. Search for **ESP32** and press install button for the “**ESP32 by Espressif Systems**”:



5.That’s it. It should be installed after a few seconds.



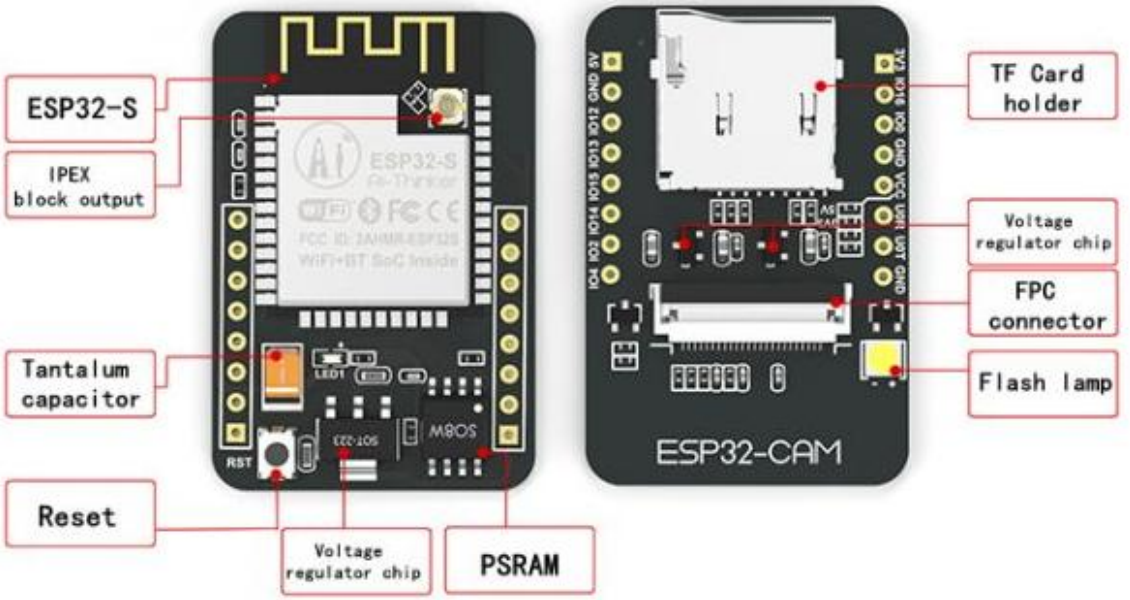
Project 1: ESP32 Cam real-time monitoring

In this project , we will real time monitoring on web pages through ESP32 Cam.

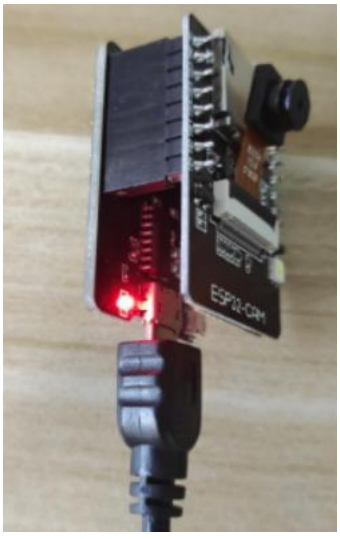
Required components:
ESP32-CAM with OV2640
FTDI programmer
Connecting line
5V power supply for ESP32-CAM

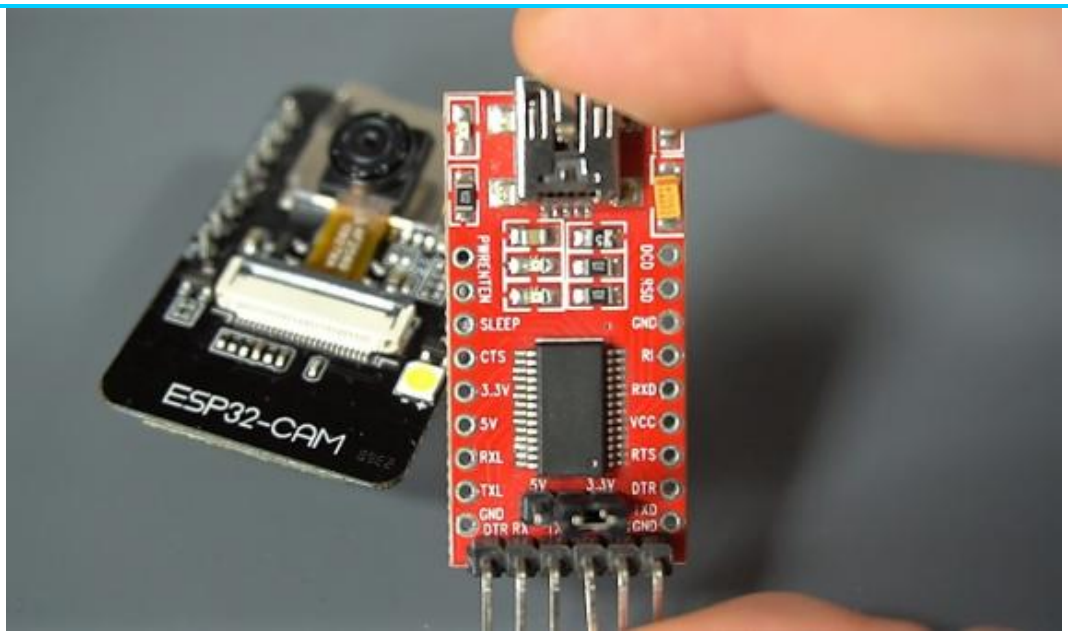
Introduction to ESP32-Cam:
The ESP32-CAM is a very small camera module with the ESP32-S chip, and the cost is less than $10. In addition to the OV2640 camera and several GPIOs for connecting peripherals, it also has a microSD card slot that can be used to store images taken with the camera or store files to be provided to the client.

The components of ESP32-Cam:



The ESP32-CAM is not equipped with an onboard micro USB connector like any onboard USB connector of NodeMCU-ESP8266, In this kit, there is an ESP32 expansion board that you can use to download via USB.Or so you can use an FTDI programmer to upload the code U0R and U0T pins (serial pins).





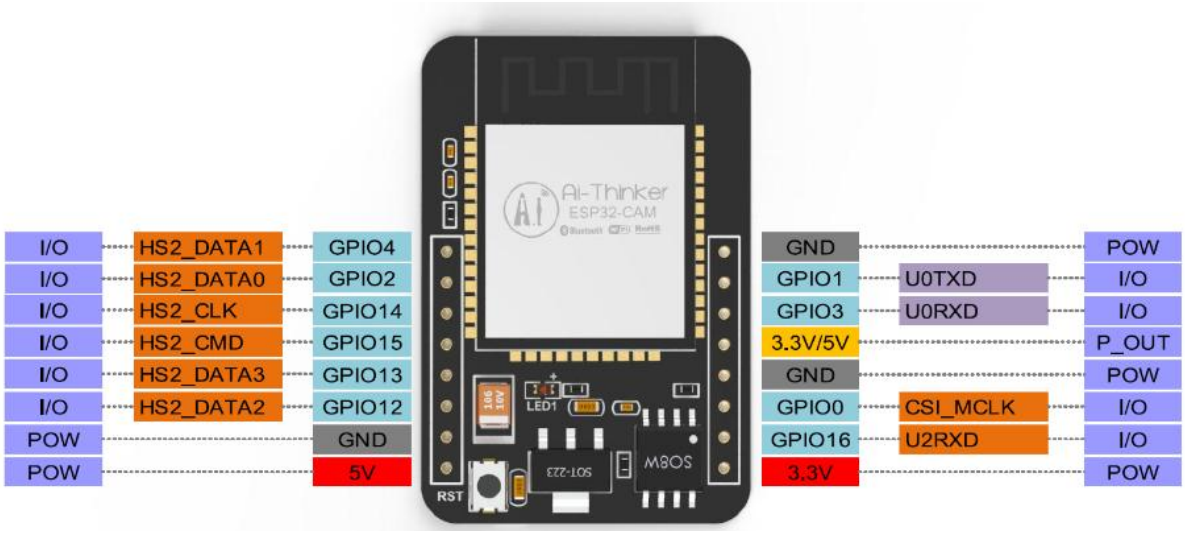
ESP32 camera features:

The following is a list of ESP32-Cam features:

- The smallest 802.11b/g/n Wi-Fi BT SoC module
- Low-power 32-bit CPU, which can also serve application processors
- The clock speed is up to 160MHz, and the aggregate computing capacity is upto 600 DMIPS
- Built-in 520 KB SRAM, external 4MPSRAM
- Support UART / SPI / I2C / PWM / ADC / DAC
- Support OV2640 and OV7670 cameras, built-in flash
- Support image upload via WiFI
- Support TF card
- Support multiple sleep modes

- Embedded Lwip and FreeRTOS
- Support STA / AP / STA + AP operation mode
- Support Smart Config / AirKiss technology
- Support serial port local and remote firmware upgrade (FOTA)

ESP32-Cam Pinout (AI-Thinker module):

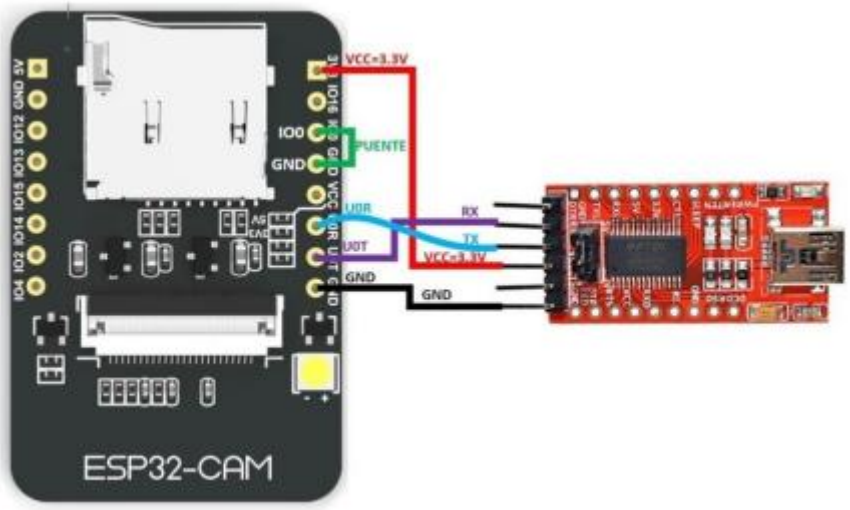


There are three GND pins and two pins for power supply: 3.3V or 5V.

GPIO 1 and GPIO 3 are serial pins. You need these pins to upload code to the board. In addition, GPIO 0 also plays an important role because it determines whether the ESP32 is in blinking mode. When GPIO 0 is connected to GND, ESP32 is in blinking mode.

Connect the ESP32-CAM board to your computer using an FTDI programmer.

Follow the next schematic diagram:



Important: GPIO 0 needs to be connected to GND so that you're able to upload code.

If the download fails, you can also use USB to download or connect the power cord to 5V

ESP32-CAM	FTDI Programmer
GND	GND
5V	VCC (5V)
U0R	TX
U0T	RX
GPIO 0	GND

To upload the code, follow the next steps:

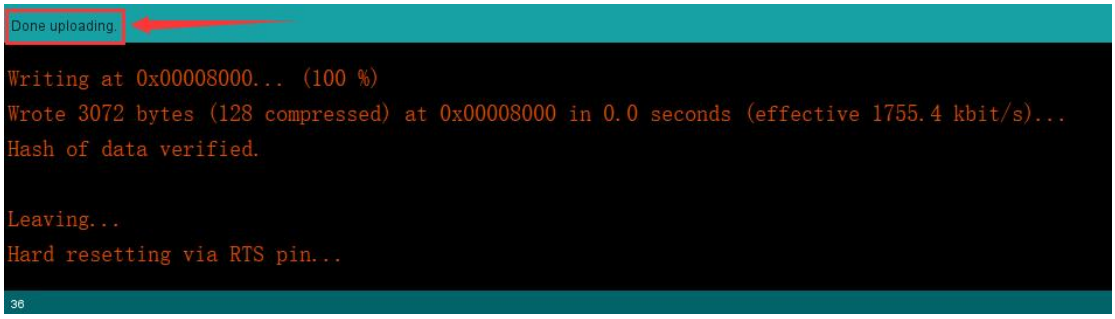
1) Go to **Tools** > **Board** and select **ESP32 Wrover Module**.

2) Go to **Tools** > **Port** and select the COM port the ESP32 is connected to.

3) Then, click the upload button to upload the code.

4) When you start to see these dots on the debugging window as shown below, press the ESP32-CAM on-board RST button.

After a few seconds, the code should be successfully uploaded to your board.



ESP32-CAM code upload

Here I choose to use USB for code download and FTDI programmer to view port output.

Note: Need to be modified in the above sketch:

You need to make some modifications before uploading the code.

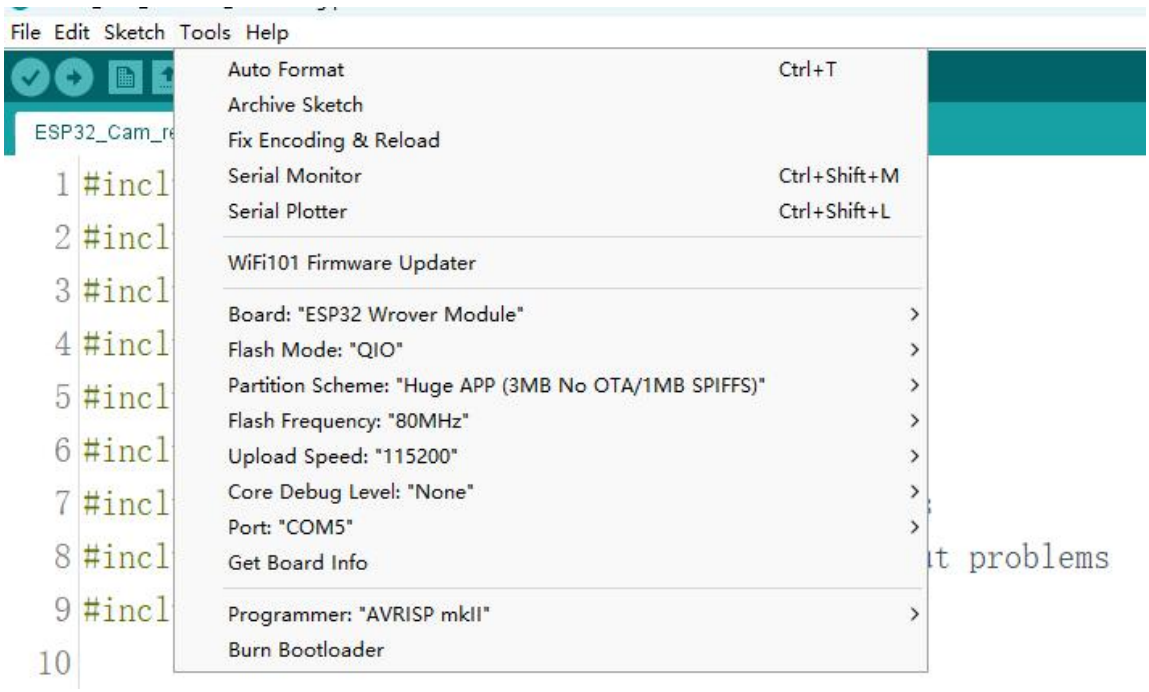
Insert the network credentials into the following variables:

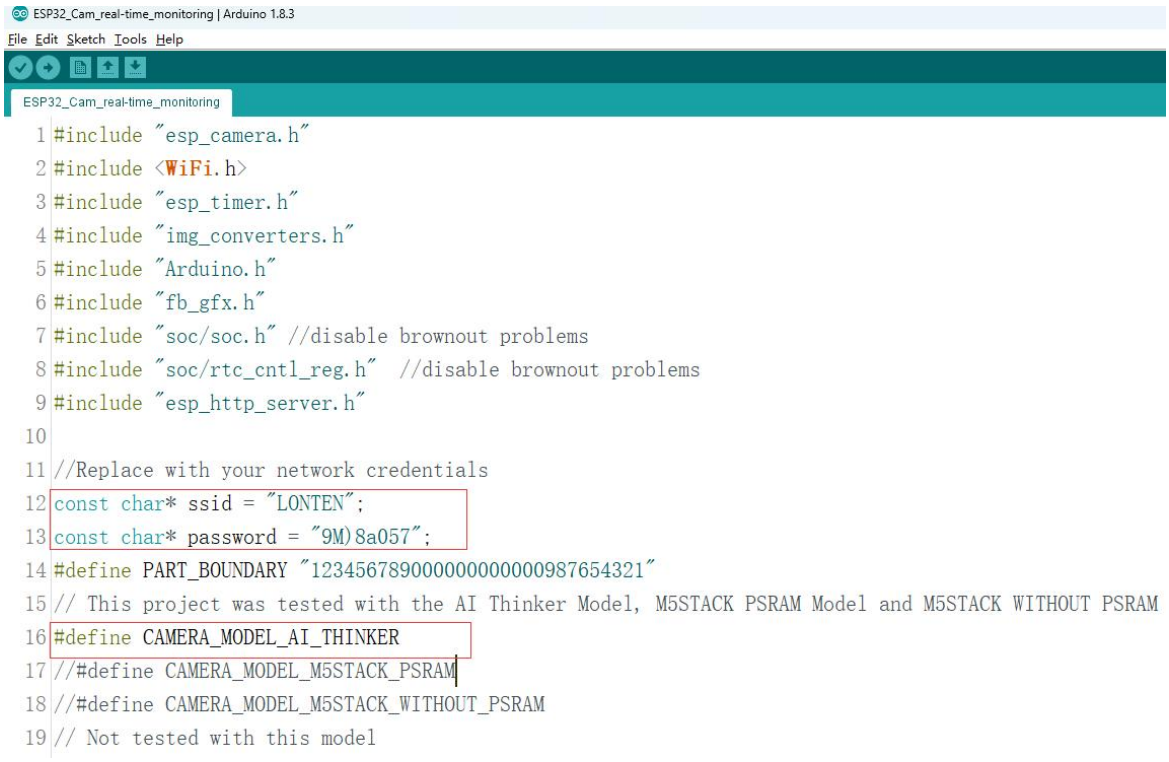
```
const char* ssid = "REPLACE_WITH_YOUR_WIFI_SSID";
const char* password = "REPLACE_WITH_YOUR_WIFI_PASSWORD";
```

Then, make sure to select the correct camera module.

Our correct choice is the **AI-THINKER model**.

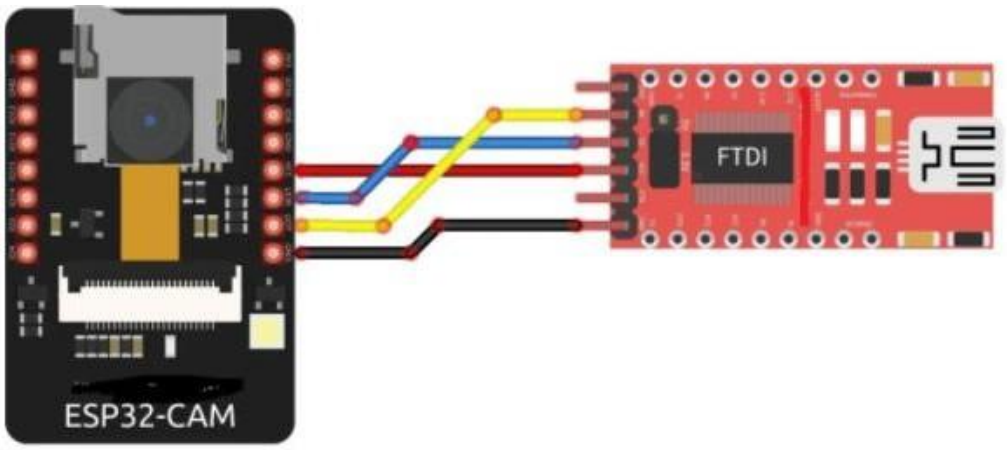
Just like this:





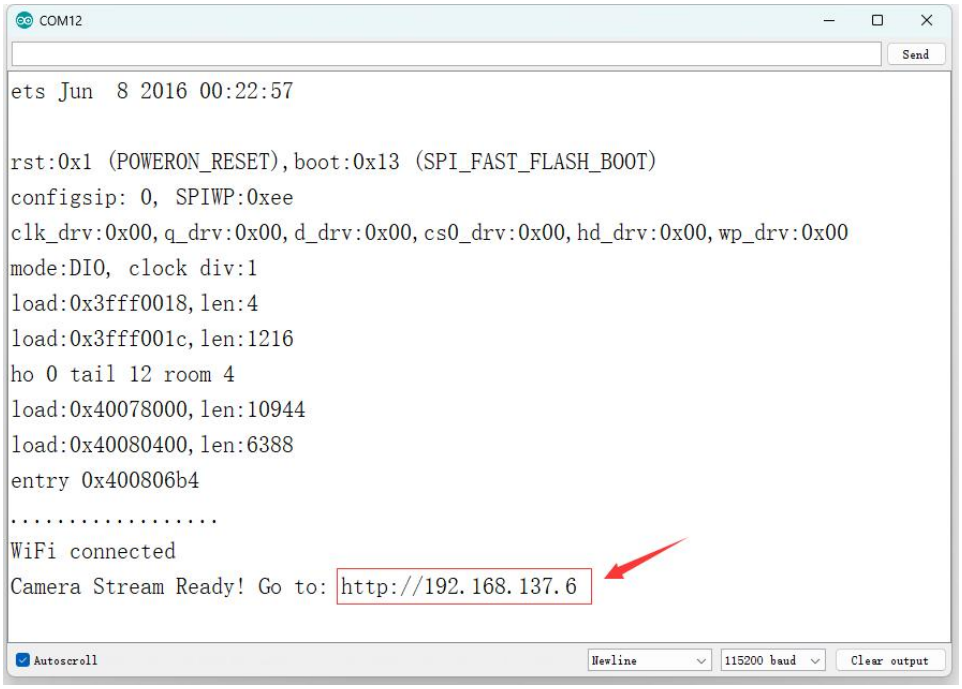
After a few seconds, the code should be successfully uploaded to your circuit board.

Obtain the local IP address of ESP32-Cam Server



Turn on the serial monitor at a baud rate of 115200. Press the reset button on the ESP32-CAM board.

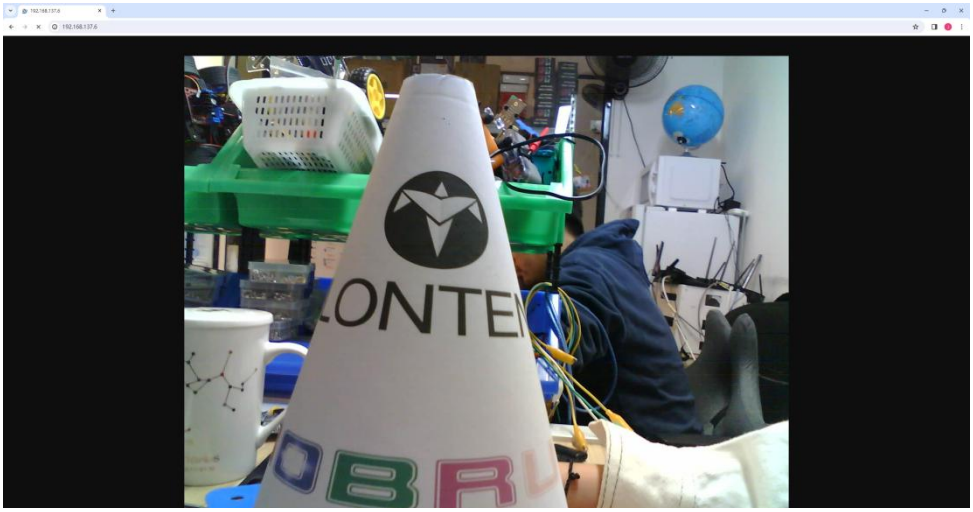
As shown below, the "WiFi connection status", the ESP32 IP address of the "Camera" module and the Web server will be printed in the "Serial Monitor".



Video streaming server

Now you can access the camera streaming server on the local network. Open your favorite browser, enter the IP address of ESP32-CAM, and press Enter to load the Video Streaming page.and you can see that the image quality is good.

NOTE:Must be on the same local area network.

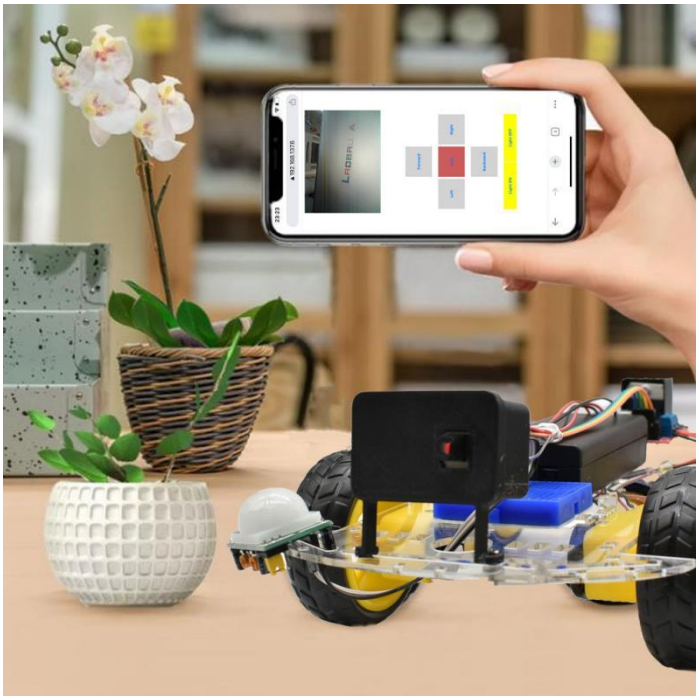


When we install the equipment in the monitoring area, to protect the equipment from environmental dust and moisture, you can also design your own interesting shell and print it through the 3D printer.

Wish you a successful experiment!

Project 2:ESP32 Camera Surveillance Car

This project will introduce you to a highly interactive ESP32 CAM game. Use ESP32 CAM to make a camera programmable car with LED light control and motor control.



Required components:

OV2640 ESP32 - CAM

FTDI programmer

Connecting line

L298N Motor Drive Module

18650 battery box

L298N Motor Drive Module:

Product parameters:

1. Driver chip: L298N dual H-bridge DC motor driver chip
2. The power supply range of the drive part of the terminal Vs: +5V～＋35V; if the board needs to be powered, the power supply range Vs: +7V～+35V
3. The peak current Io of the driving part: 2A
4. The power supply range Vss of the logic part terminal: +5V～＋ 7V (+5V can be taken from the board)
5. Logic part operating current range: 0～36mA
6. Control signal input voltage range:

Low level: -0.3V≤Vin≤1.5V

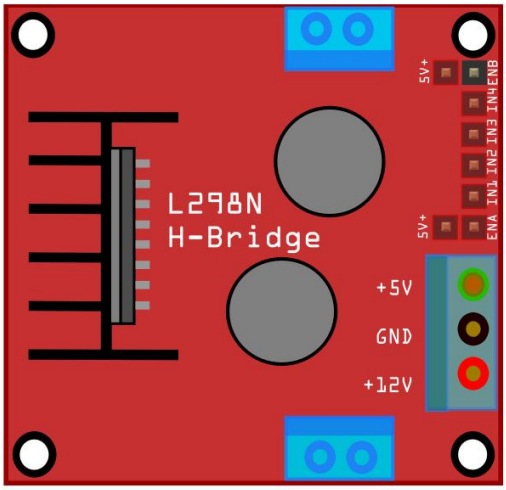
High level: 2.3V≤Vin≤Vss

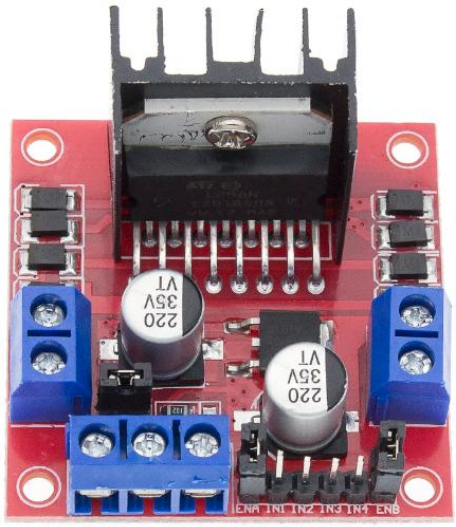
7. Enable signal input voltage range:

Low level: -0.3≤Vin≤1.5V (control signal is invalid)

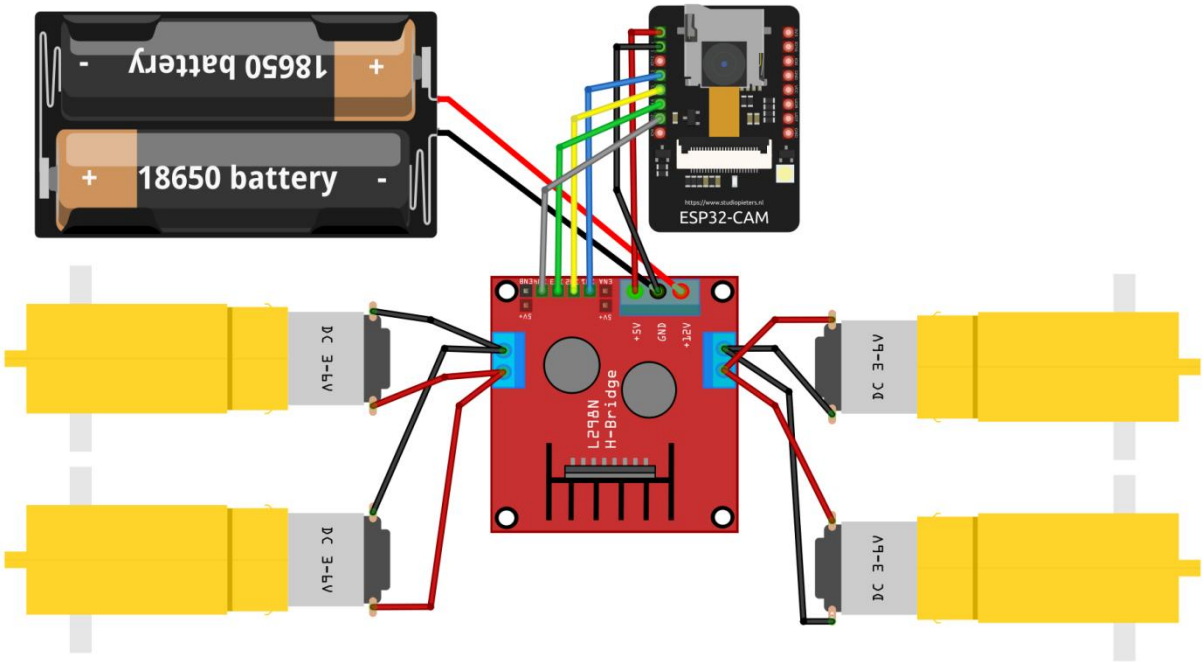
High level: 2.3V≤Vin≤Vss (control signal is valid)

8. Maximum power consumption: 20W (when temperature T=75℃)

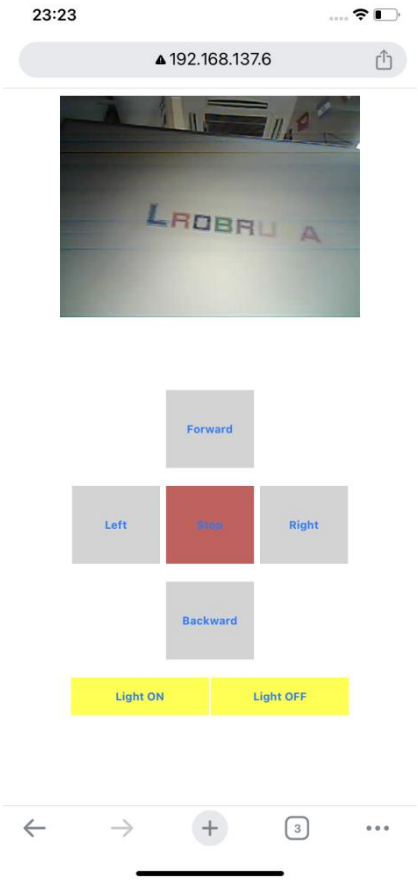




Trolley circuit wiring diagram:



After downloading the code, you can control and monitor the car.



Project 3:ESP32-CAM Take Photo and Save to MicroSD Card

Learn how to take photos with the ESP32-CAM board and save them to a microSD card using Arduino IDE. When you press the ESP32-CAM RESET button, it wakes up, takes a photo and saves it in the microSD card.

Required components:

OV2640 ESP32 - CAM

FTDI programmer

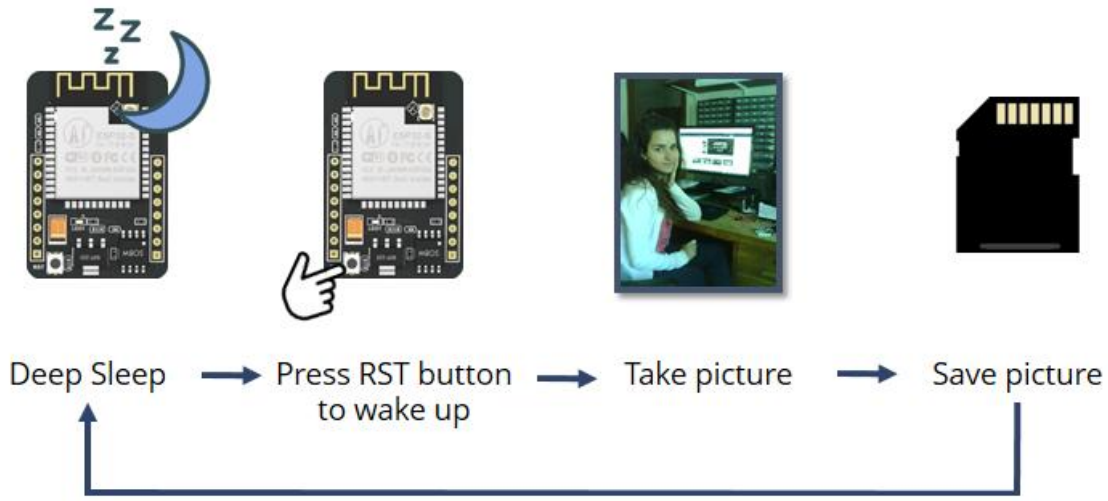
MicroSD card

Connecting line

18650 battery box

Project Overview：

Here is a quick overview on how the project works.



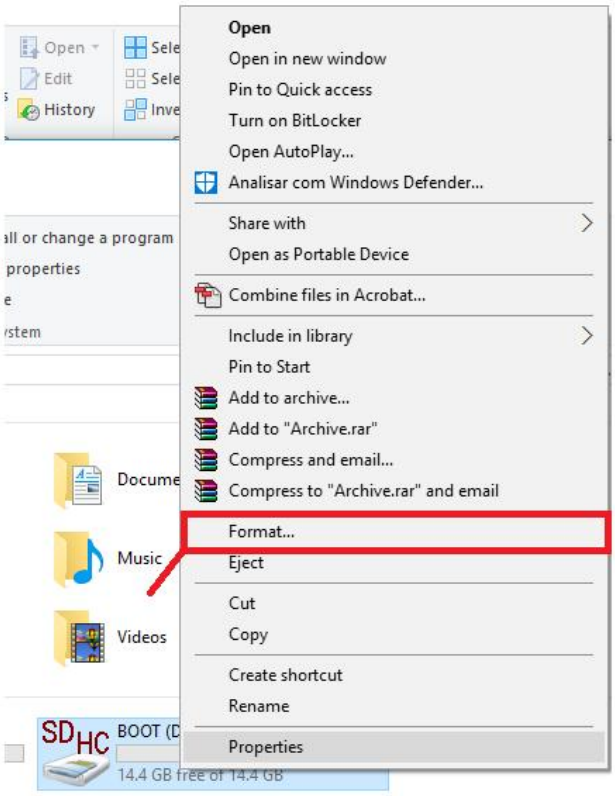
- The ESP32-CAM is in deep sleep mode
- Press the RESET button to wake up the board
- The camera takes a photo
- The photo is saved in the microSD card with the name: pictureX.jpg, where X corresponds to the picture number

  The picture number will be saved in the ESP32 flash memory so that it is not erased during RESET and we can keep track of the number of photos taken.

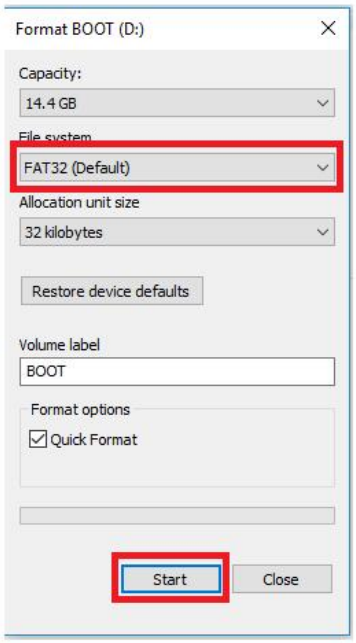
Formatting MicroSD Card

The first thing we recommend doing is formatting your microSD card. You can use the Windows formatter tool or any other microSD formatter software.

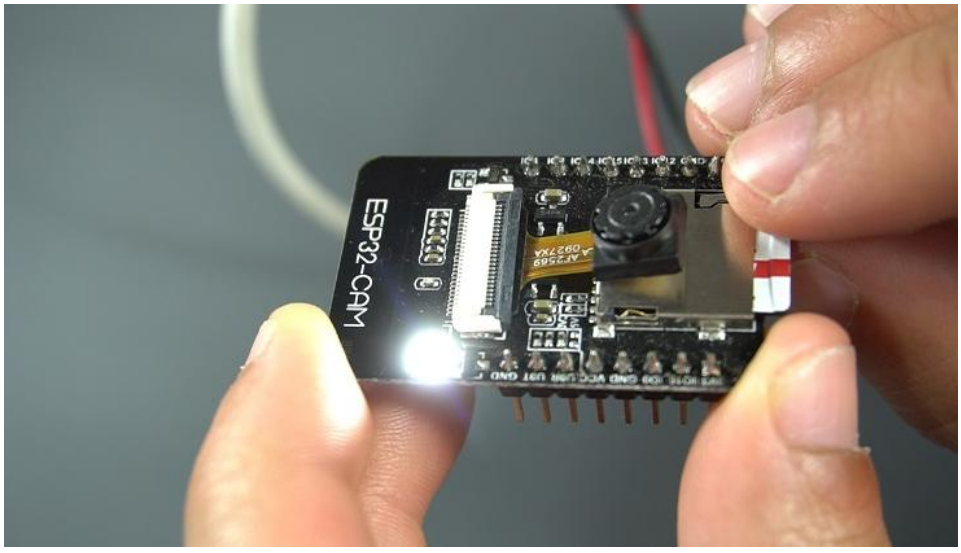
1. Insert the microSD card in your computer. Go to **My Computer** and right click in the SD card. Select **Format** as shown in figure below.



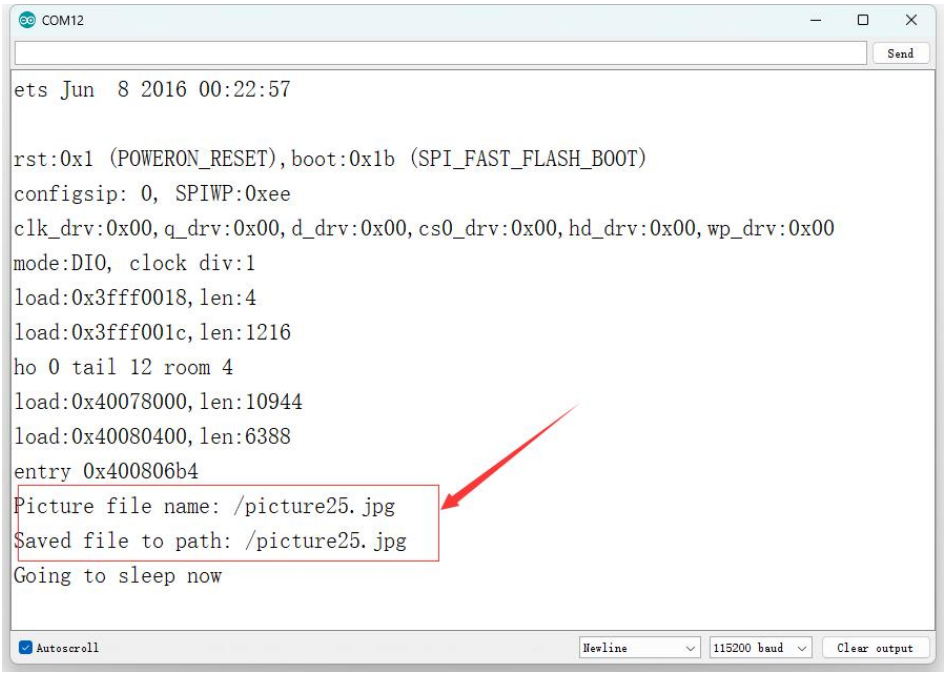
2. A new window pops up. Select **FAT32**, press **Start** to initialize the formatting process and follow the onscreen instructions.



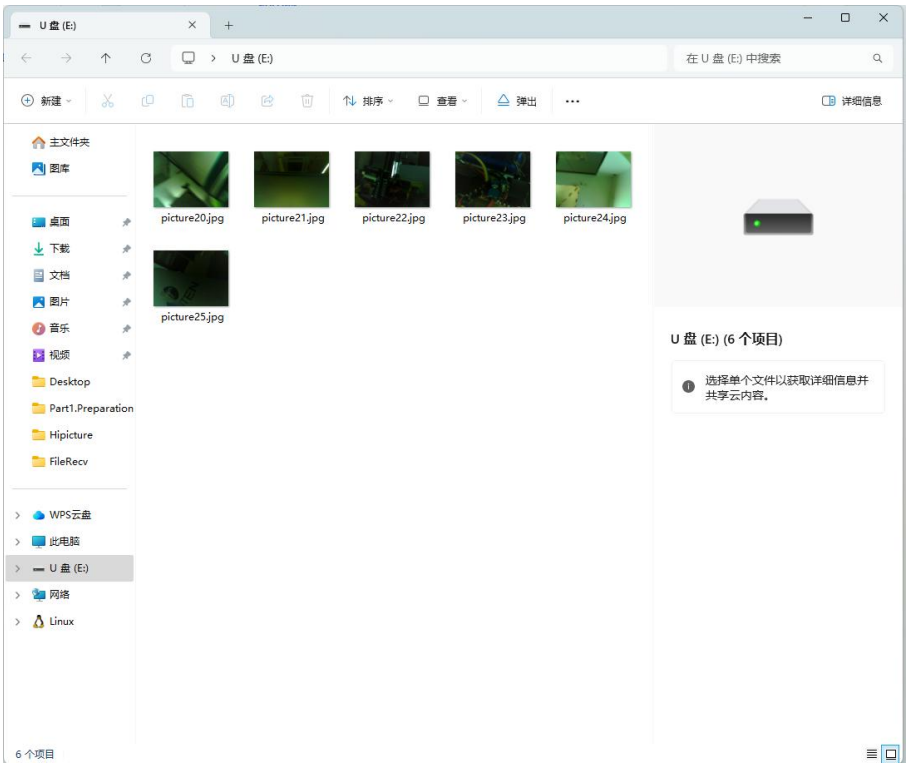
Demonstration



Check the Arduino IDE Serial Monitor window to see if everything is working as expected. As you can see, the picture was successfully saved in the microSD card.



To see the photos taken, remove the microSD card from the microSD card slot and insert it into your computer. You should have all the photos saved.



The quality of your photo depends on your lighting conditions. Too much light can ruin your photos and dark environments will result in many black pixels.

Project 4: ESP32-CAM PIR Motion Detector with Photo Capture (saves to microSD card)

In this project, we're going to make a motion sensor detector with photo capture using an ESP32-CAM. When your PIR sensor detects motion, it wakes up, takes a photo and saves it in the microSD card.

Required components:

OV2640 ESP32 - CAM

FTDI programmer

MicroSD card

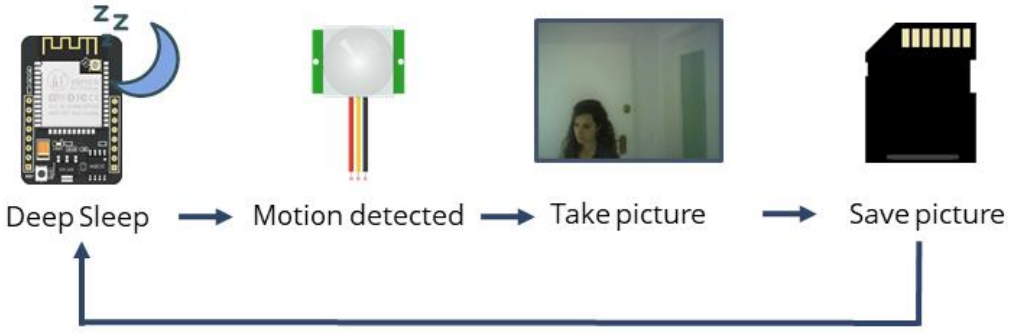
PIR motion sensor

Connecting line

18650 battery box

Project Overview：

Here is a quick overview on how the project works.

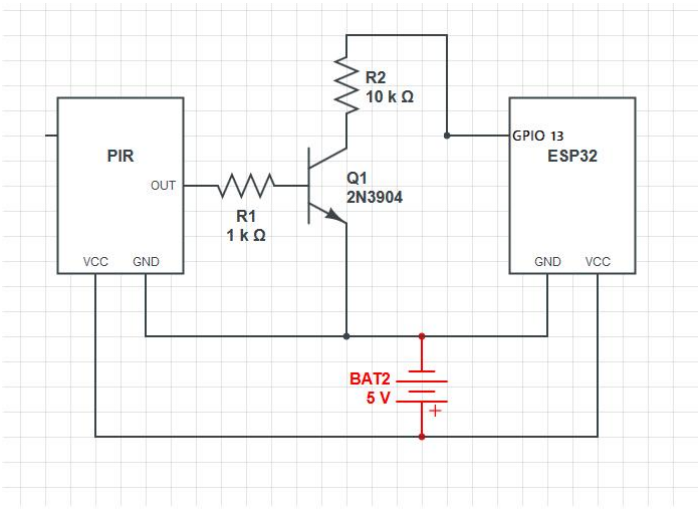


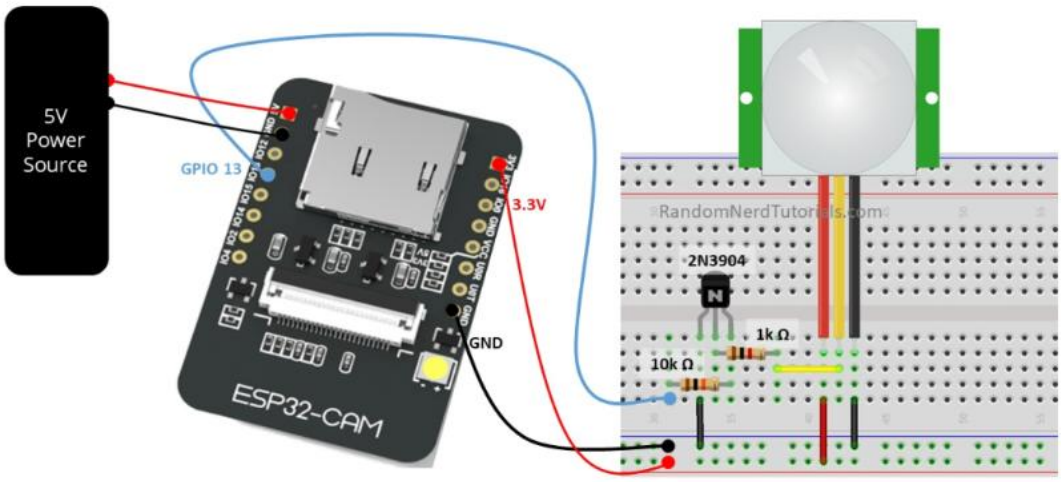
- The ESP32-CAM is in deep sleep mode with external wake up enabled.
- When motion is detected, the PIR motion sensor sends a signal to wake up the ESP32.
- The ESP32-CAM takes a photo and saves it on the microSD card.

  It goes back to deep sleep mode until a new signal from the PIR motion sensor is received

schematic diagram：

Assemble all parts as shown below:



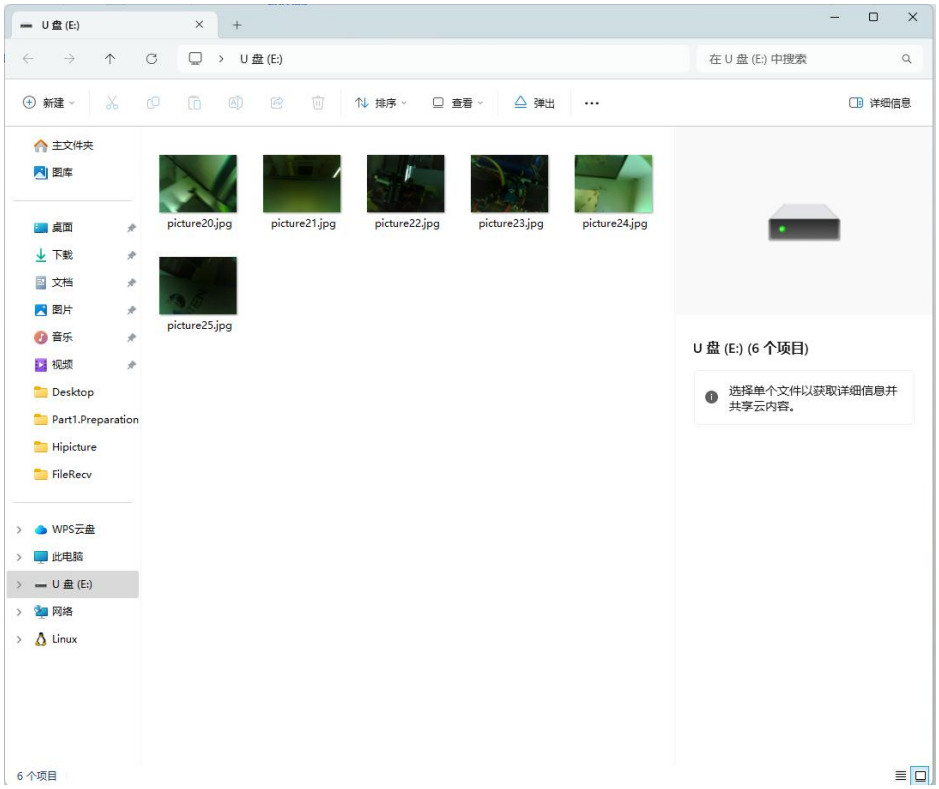


Demonstration

After uploading the code and assembling the circuit, insert a formatted microSD card and power the circuit.

Then, press the Reset (RST) button and it should start working. When it detects movement, it turns on the flash, takes a picture and saves it on a microSD card.

The circuit was tested several times to ensure its normal operation. The microSD card is then inserted into the computer to view the captured photos.



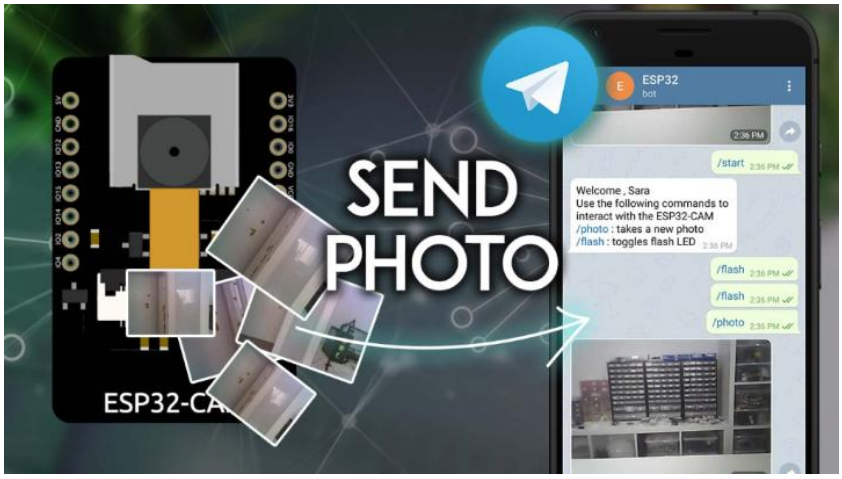
This is a real case:

When we install the equipment in the surveillance area, in order to protect the equipment from environmental dust and moisture, you can also design an interesting shell by yourself and print it out through a 3D printer; install this equipment in the surveillance area, as shown in the picture below. The image quality is pretty good.

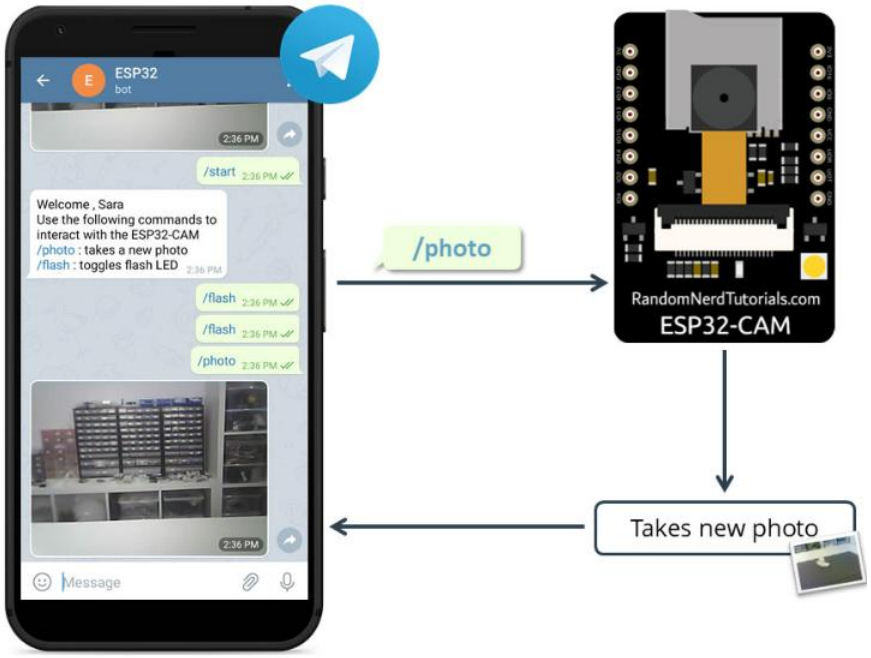
I wish you a successful experiment!

Project 5:Telegram: ESP32-CAM Take and Send Photo (Arduino IDE)

In this tutorial, you'll create a Telegram bot to interact with the ESP32-CAM to request a new photo. You can request a new photo using your Telegram account from anywhere. You just need to have access to the internet on your smartphone.



Project Overview:



Here's an overview of the project you'll build:

- You'll create a Telegram bot for your ESP32-CAM;
- You can start a conversation with the ESP32-CAM bot;
- When you send the message **/photo** to the ESP32-CAM bot, the ESP32-CAM board receives the message, takes a new photo, and responds with that photo;
- You can send the message **/flash** to toggle the ESP32-CAM's LED flash;
- You can send the **/start** message to receive a welcome message with the commands to control the board;

- The ESP32-CAM will only respond to messages coming from your Telegram account ID.

This is a simple project but shows how you can use Telegram in your IoT and Home Automation projects. The idea is to apply the concepts learned in your own projects.

Introducing Telegram

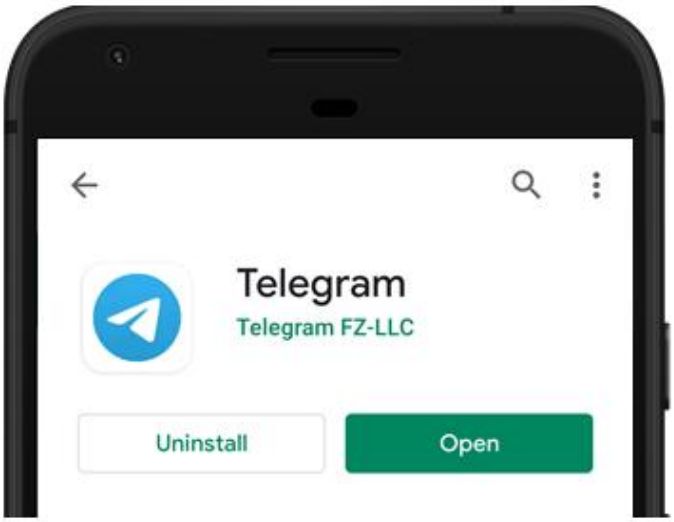
Telegram Messenger is a cloud-based instant messaging and voice over IP service. You can easily install it in your smartphone (Android and iPhone) or computer (PC, Mac and Linux). It's free and without any ads. Telegram allows you to create bots that you can interact with.

"Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and inline requests. You control your bots using HTTPS requests to Telegram Bot API".

The ESP32-CAM will interact with the Telegram bot to receive and handle the messages, and send responses. In this tutorial, you'll learn how to use Telegram to send messages to your bot to request a new photo taken with the ESP32-CAM. You can receive the photo wherever you are (you just need Telegram and access to the internet).

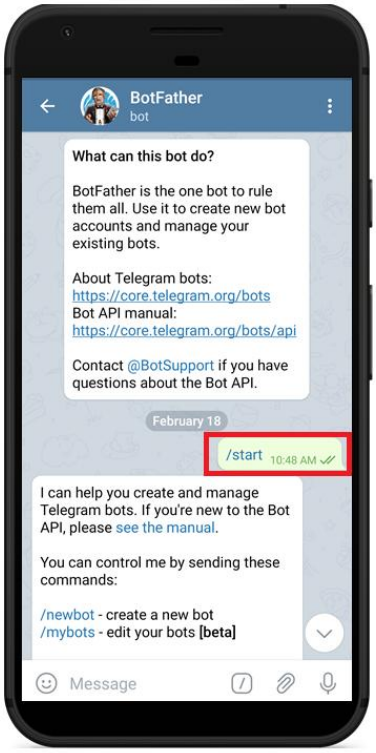
Creating a Telegram Bot

Go to Google Play or App Store, download and install **Telegram**.



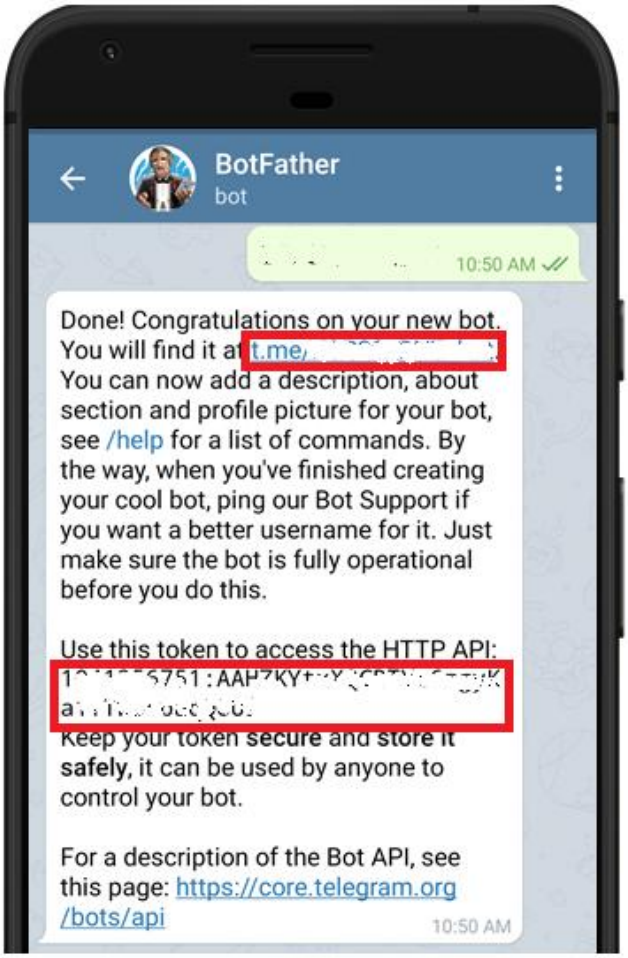
Open Telegram and follow the next steps to create a Telegram Bot. First, search for "**botfather**" and click the BotFather as shown below. Or open this link t.me/botfather in your smartphone.

The following window should open and you'll be prompted to click the **start** button.



Type **/newbot** and follow the instructions to create your bot. Give it a name and username.

If your bot is successfully created, you'll receive a message with a link to access the bot and the **bot token**. Save the bot token because you'll need it so that the ESP32 can interact with the bot.

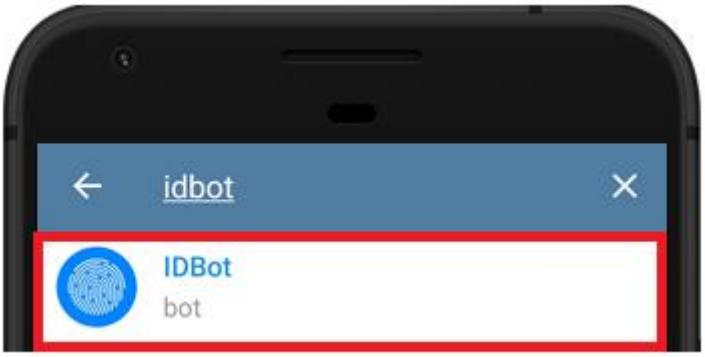


Get Your Telegram User ID

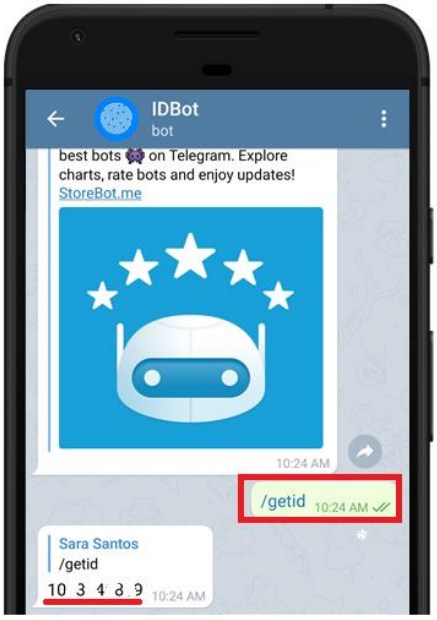
Anyone that knows your bot username can interact with it. To make sure that we ignore messages that are not from our Telegram account (or any authorized users), you can get your Telegram User ID. Then, when your

telegram bot receives a message, the ESP can check whether the sender ID corresponds to your User ID and handle the message or ignore it.

In your Telegram account, search for "**IDBot**" or open this link t.me/myidbot in your smartphone.



Start a conversation with that bot and type **/getid**. You will get a reply back with your user ID. Save that **user ID**, because you'll need it later in this tutorial.



Preparing Arduino IDE

We'll program the ESP32 board using Arduino IDE, so make sure you have them installed in your Arduino IDE.

- Installing the ESP32 Board in Arduino IDE (Windows, Mac OS X, Linux)

Universal Telegram Bot Library

To interact with the Telegram bot, we'll use the Universal Telegram Bot Library created by Brian Lough which provides an easy interface for the Telegram Bot API.
Follow the next steps to install the latest release of the library.

1. Click here to download the Universal Arduino Telegram Bot library.
2. Go to **Sketch** > **Include Library** > **Add.ZIP Library...**
3. Add the library you've just downloaded.

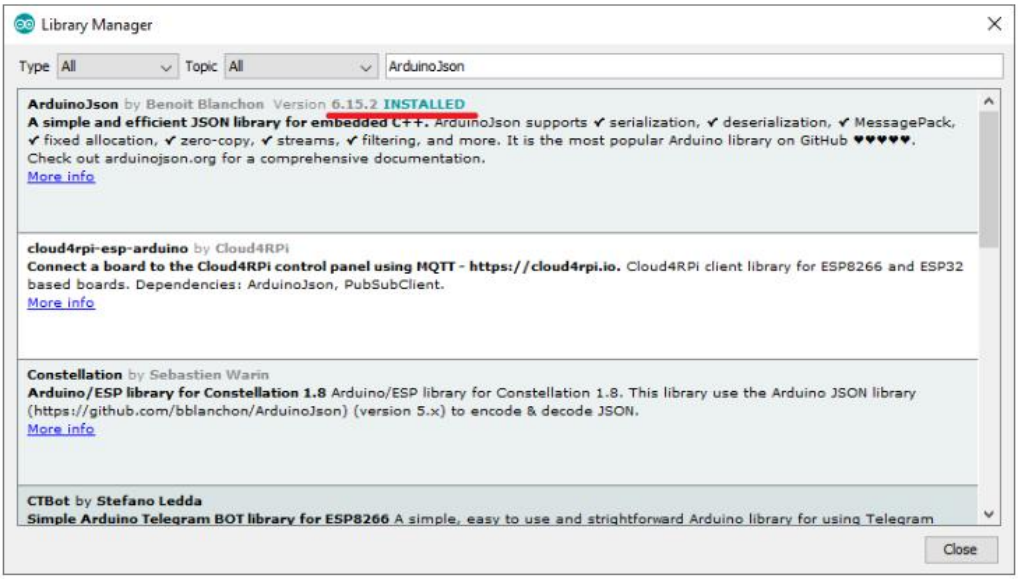
Important: don't install the library through the Arduino Library Manager because it might install a deprecated version.
For all the details about the library, take a look at the Universal Arduino Telegram Bot Library GitHub **page**.

ArduinoJson Library

You also have to install the ArduinoJson library. Follow the next steps to install the library.

1. Go to **Sketch** > **Include Library** > **Manage Libraries**.
2. Search for "ArduinoJson".
3. Install the library.

We're using ArduinoJson library version 6.15.2.



Demonstration

Upload the code to your ESP32-CAM board. Don't forget to go to **Tools** > **Board** and select the board you're using. Go to **Tools** > **Port** and select the COM port your board is connected to.

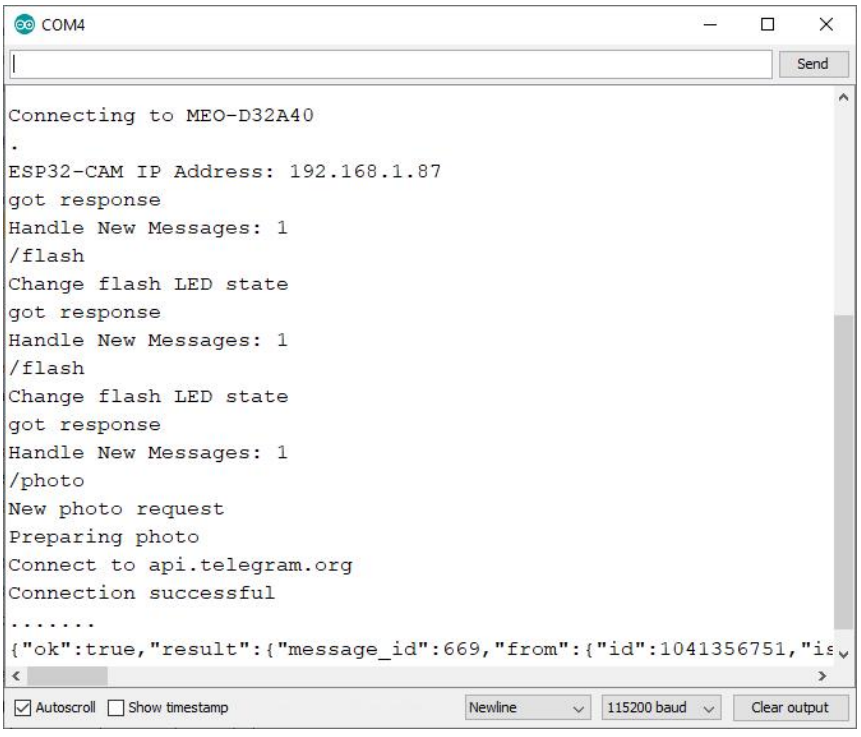
After uploading the code, press the ESP32-CAM on-board RST button so that it starts running the code. Then, you can open the Serial Monitor to check what's happening in the background.

Go to your Telegram account and open a conversation with your bot.

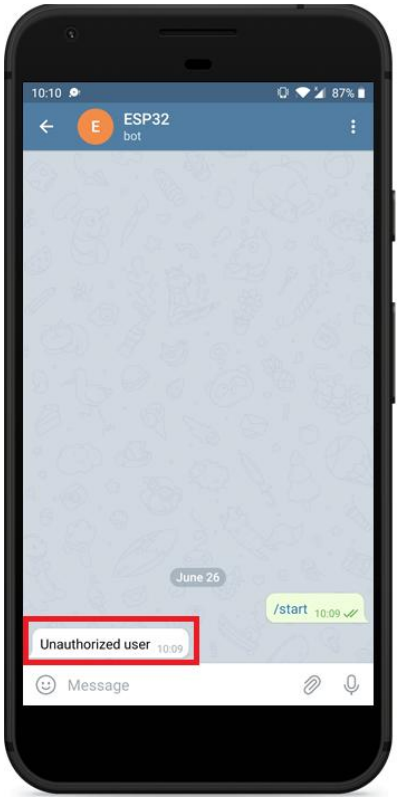
Send the following commands and see the bot responding:

- **/start** shows the welcome message with the valid commands;
- **/flash** inverts the state of the LED flash;
- **/photo** takes a new photo and sends it to your Telegram account.

At the same time, on the Serial Monitor, you should see that the ESP32-CAM is receiving the messages.



If you try to interact with your bot from another account, you'll get the "Unauthorized user" message.



Project 6: ESP32-CAM takes photos and displays them on a web server

In this project, Learn how to use the ESP32-CAM development board to build a web server, which allows you to send commands to take photos and visualize the latest photos in a browser saved in SPIFFS. We also added an option to rotate the image when necessary.

Required components:

OV2640 ESP32 - CAM

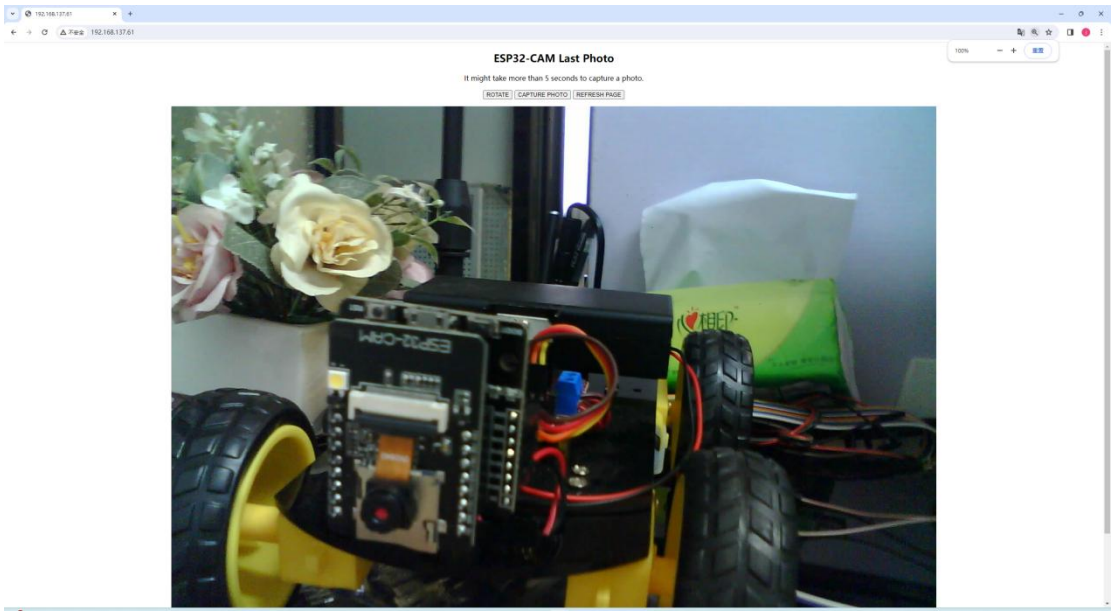
FTDI programmer

Connecting line

18650 battery box

Demonstration

After uploading the code and assembling the circuit, insert a formatted microSD card and power the circuit.



NOTE:At the beginning, there may not be any visuals. At this point, we can click on the CAPURE PHOTO button to take photos, and use the REFRESH PAGE button to obtain the previous photo taken.