# Arduino IDE1 development environment construction for ESP32

# CONTENTS

1. Download Arduino IDE1 software installation package3	
2. Arduino IDE1 software installation	
3. Arduino IDE1 software introduction	
3.1. Menu bar9	
3.1.1. File menu9	
3.1.2. Edit menu	
3.1.3. Sketch menu11	
3.1.4. Tools Menu12	
3.1.5. Help Menu	
3.2. Tool bar	
4. Install the Arduino-ESP32 core software library16	
4.1. Arduino IDE development board Manager is installed online	
4.2. Manual Offline installation	
<b>5.</b> Compile, download and run the ESP32 sample program20	
5.1. Install the USB-to-serial IC driver20	
5.2. Configure the development board21	
5.3. Compile, download, and run the program25	

# 1. Download the Arduino IDE1 software installation package

There are two versions of the Arduino IDE: Arduino IDE1 and Arduino IDE2. Arduino IDE1 is an old version that only supports Win7 and below systems, while Arduino IDE2 is a new version that supports Win10 and above systems. I'm only going to introduce the Arduino IDE1 here.

The Arduino IDE1 software installation package can be downloaded directly from the official website at:

https://www.arduino.cc/en/software

After entering the software download page of the official website, scroll down to find the Arduino IDE1 software installation package download column, as shown in the following picture:



Figure 1.1 Arduino IDE1 software installation package download interface 1

Select the appropriate version for your computer system from the download options. For example, if you use Windows, click "**Windows Win 7 and newer**" to download. You can also download ZIP files.

After clicking on the download option, a screen will pop up asking whether to provide team financial support, as shown below:



Figure 1.2 Arduino IDE1 software installation package download interface 2

You can ignore this option and simply click the "JUST DOWNLOAD" button.

After clicking the button, an interface will pop up asking whether to enter an email to receive Arduino information. You can ignore it and directly click the "**JUST DOWNLOAD**" button, as shown in the picture below:

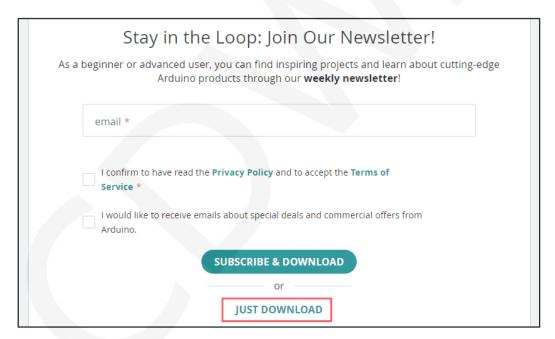


Figure 1.3 Arduino IDE1 software installation package download interface 3

After clicking the button, a "**New Download Task**" window will pop up. Click the "**Browse**" button to select the saving path of the software installation package, and then click the "**Download**" button to start downloading, as shown in the following picture:

新建下载任务		0
网址 https://downloads.arduir	no.cc/arduino-1.8.19-windows.exe	
名称 arduino-1.8.19-windows.	exe 111.	87 MB
下载到 D:\Arduino	剩: 150.89 GB 🔻	刘览
	下载并运行 取消	下载

Figure 1.4 Arduino IDE1 software installation package download task

# 2. Arduino IDE1 software installation

Find the path to save the Arduino IDE1 software installation package, and then double-click the exe file to enter the program installation (if the window asking whether to run the file pops up, directly click the "**Run**" button), as shown in the following picture:

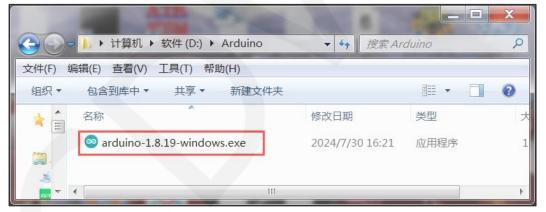


Figure 2.1 Arduino IDE1 software installation package exe file





Figure 2.2 Arduino IDE1 software installation license Agreement

Click the "I Agree" button to enter the installation content selection interface, select some basic software, drivers, and shortcuts to be installed. If you do not want to install a piece of software, remove the corresponding check mark. Generally, no operation is required, and you can keep the default, as shown in the following figure:

😎 Arduino Setup: Installation Options 📃 💻 💌			
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.			
Select components to install: Install Arduino software Install USB driver Create Start Menu shortcut Create Desktop shortcut Associate .ino files			
Space required: 541.6MB			
Cancel Nullsoft Install System v3.0 < Back Next >			

Figure 2.3 Selection of Arduino IDE1 software installation content

Then click "**Next**" button to enter the installation directory setting interface, click "**Browse**" button to select the installation directory or directly enter the installation directory, as shown in the following picture:

😎 Arduino Setup: Installation Folder	_ <b>D</b> X	
Setup will install Arduino in the following folder. To folder, click Browse and select another folder. Click installation.		
Destination Folder		
D:\Program Files (x86)\Arduino Browse		
Space required: 541.6MB		
Space available: 151.6GB		
Cancel Nullsoft Install System v3.0 < Ba	ack Install	

Figure 2.4 Selection of Arduino IDE1 software installation directory

Then click the "Install" button to start the installation, and you can see the change of the installation progress bar (if the window that needs to install some device software pops up during the installation, click the "Install" button directly), as shown in the following picture:

💿 Arduino Set	up: Installing		
Show details			
	3		
Cancel	Nullsoft Install System v3.0	< Back	Close

Figure 2.5 Arduino IDE1 software installation process

When the progress bar is Completed, the information "Completed" will be displayed, indicating that the software installation is complete. Click "Close" to finish the installation process, as shown in the following picture:

💿 Arduino Seti	up: Completed	
Completed	I	
Show details		
Cancel	Nullsoft Install System v3.0	< Back Close

Figure 2.6 Arduino IDE1 software installation complete

# 3. Arduino IDE1 software introduction

Arduino IDE1 has the functions of project creation, program code editing, debugging, compilation, upload, software library management, development board management, etc. The interface is shown as follows:

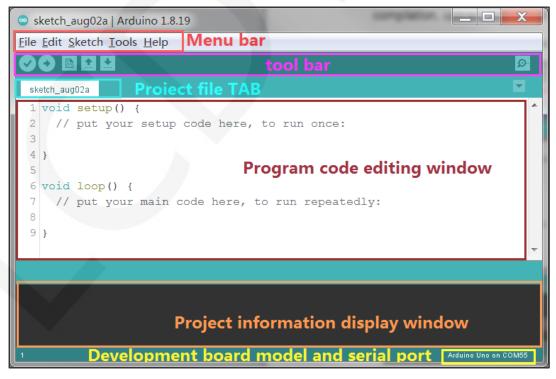


Figure 3.1 Arduino IDE1 interface

# 3.1. Menu bar

# 3.1.1. File menu

The menu contents of the menu bar file are shown as follows:

💿 s	ketch_aug02a	Arduino 1.8.19		X
<u>F</u> ile	<u>E</u> dit <u>S</u> ketch <u></u>	<u>T</u> ools <u>H</u> elp		
	New	Ctrl+N		Ø
	Open	Ctrl+O		
	Open Recent	۱.		
	Sketchbook	•	de here, to run once:	
	Examples	,	de nere, co fun once.	
		Ctrl+W		
	Save			
	Save As	Ctrl+Shift+S		
	Page Setup	Ctrl+Shift+P	e here, to run repeatedly:	
	Print	Ctrl+P		
	Preferences	Ctrl+Comma		~
	Quit	Ctrl+Q		
Γ				
1			Arduino Uno on CO	M55

Figure 3.2 Arduino IDE1 file menu bar

Basically, it is to create, open and save the project. You can select the **Examples** option to open the third-party software library and the sample program of the development board core library. Here the preferences menu is highlighted, click on the "**Preferences**" option, as shown below:

Preferences		<b>—X</b>		
Settings Network				
Sketchbook location:				
C:\Users\Administrator\Docum	ents\Arduino	Browse		
Editor language:	English (English)	(requires restart of Arduino)		
Editor font size:	16			
Interface scale:	Automatic 100 🔆 (requires restart of Arduino)			
Theme:	Default theme - (requires restart of Arduino)			
Show verbose output during:	Compilation Vupload			
Compiler warnings:	Default 🔻			
🗹 Display line numbers	Enable Code Folding			
Verify code after upload	🔲 Use external editor			
Check for updates on star	tup 📝 Save when verifying or up	oloading		
🔲 Use accessibility feature	5			
Additional Boards Manager UR	Ls:			
More preferences can be edited directly in the file				
C:\Users\Administrator\AppDa	ta\Local\Arduino15\preferences.txt			
(edit only when Arduino is n	ot running)			
		OK Cancel		

#### Figure 3.3 Arduino IDE1 Preferences menu

In the Preferences menu, you can make the following Settings:

- A. Sketchbook location, is the new project, the software default setting of the project save location, you can modify the location. The libraries directory in this location is used to store third-party software libraries.
- B. Editor language, you can set multiple languages, after the setting needs to restart the Arduino IDE software to take effect.
- C. Editor Interface Settings, keep the theme, code editing window and so on Settings.
- D. Compiler warnings, you can select "none", "default", "more", "all", etc., select "none", will not display any information, select "more" or "all", will display more comprehensive compilation information, but will slow down the compilation speed.
- E. Additional Boards Manager URLs, when you want to add the development board in the IDE's development board manager can not be found (non-Arduino official development board), you need to add the address of the development board here.
- F. more detailed setting options, you can open the "**preferences.txt**" file to view (directory see the preferences interface)

After setting, click the "**OK**" button to save.

#### 3.1.2. Edit menu

Menu bar editing menu interface is shown as follows:

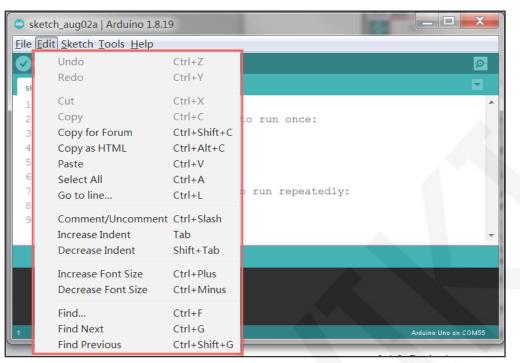


Figure 3.4 Arduino IDE1 Edit menu bar

The editing menu is mainly to copy, cut, undo, paste, find, modify the size of the

project file content and other editing operations.

#### 3.1.3. Sketch menu

Menu bar Sketch menu interface is shown as follows:

💿 sketch_a	ug02a   Arduino 1.8.19				x
File Edit	ketch Tools Help		1		
	Verify/Compile	Ctrl+R			Ø.
sketch a	Upload	Ctrl+U			-
1 voi	Upload Using Programmer				
2 /	Export compiled Binary	Ctrl+Alt+S	once:		
3	Show Sketch Folder	Ctrl+K			
4 }	Include Library	۰.			
5	Add File				
	100p() {				
8	put your main code her	e, to run r	epeatedly:		
9 }					
					-
1				Arduino Uno on CO	M55

Figure 3.5 Arduino IDE1 project menu bar

The Sketch menu bar is mainly used to compile, upload, export and load library

files for the project.

# 3.1.4. Tools Menu

The menu bar tools menu interface is shown as follows:

💿 sketch_aug02a	Arduino 1.8.19		
File Edit Sketch	pols Help		
sketch_aug02a 1 void setu 2 // put	Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries	Ctrl+T Ctrl+Shift+I	
2 // put 3 4 } 5	Serial Monitor Serial Plotter WiFi101 / WiFiNINA Firmware	Ctrl+Shift+M Ctrl+Shift+L e Updater	
6 <b>void</b> loop 7 // put 8 9 }	Board: "Arduino Uno" Port Get Board Info	) }	
	Programmer Burn Bootloader		•
1			Arduino Uno on COM55

Figure 3.6 Arduino IDE1 tool menu bar

The following Settings can be made in the tools menu:

A. Auto Format can automatically typeset the project code format,

such as alignment and so on.

- B. Archive Sketch can ZIP package and save the entire project file.
- C. Fix Encoding & Reload You can restore the program to its previous state and reload it.
- D. Manager Libraries can search, download and install the third-party software library, and click to enter, as shown in the following figure:

🗢 Library Manager	
Type All Topic All	Filter your search
A1P1c_Opta by Arduino Arduino IDE PLC runtime library for Arduino Opta This is the runtir the Arduino PLC IDE. <u>More info</u>	me library and plugins for supporting the Arduino Opta in           Version 1.1.0
AIPIC_PMC by Arduino Arduino IDE PLC runtime library for Arduino Portenta Machine Con the Arduino Portenta Machine Control in the Arduino PLC IDE. More info	trol This is the runtime library and plugins for supporting
Arduino Cloud Provider Examples by Arduino Examples of how to connect various Arduino boards to cloud provi More info	ders Close

Figure 3.7 Arduino IDE1 library Manager

In the library manager interface, you can filter the library according to the type and Topic of the software library, or you can directly output the name of the library to search the library. After the search is complete, select the library version and click the "**Install**" button to install. Finally, the library is installed in the

"C:\Users\Administrator\Documents\Arduino\libraries" directory (this is the default directory, of course, you can change it in the File -> Preferences interface, the actual user name of the computer is in red font). Of course, you can also install the software library without using the library manager. You can download the library manually (you need to extract it) and copy it to the

C:\Users\Administrator\Documents\Arduino\libraries directory.

E. Serial Monitor and Serial Plotter open the serial port interface, set the serial port baud rate, display the serial port output information, and send messages through the serial port. (Note that the development board should be connected, and the serial port can be used only after the serial port is correctly identified).

F. Board consists of two parts: Borads manager and Boards selection, as shown in the following figure:

sketch_aug02b	Arduino 1.8.19		A-U MEI ME
File Edit Sketch T	ools Help		
sketch_aug02b	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T	
1 void setu 2 // put 3 4 }	Manage Libraries Serial Monitor Serial Plotter	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L	
5 6 void loop	WiFi101 / WiFiNINA Firmwa	re Updater	
7 // put	Board: "Arduino Uno"	)	Boards Manager
8 9 }	Port Get Board Info		Arduino AVR Boards Arduino Renesas UNO R4 Boards
	Programmer Burn Bootloader	1	Arduino SAMD (32-bits ARM Cortex-M0+) Boards ESP32 Arduino

Figure 3.8 Arduino IDE1 development board menu

The Board option function is to select the currently used development board. Once the selection is successful, it will be displayed after the option. If the development version used does not exist, the core software library of the development board needs to be installed, in which case the development board manager is used.

The Boards manager is to install the core software library of the development board. The interface is as follows: search the development board, select the version, and click the **"Install**" button to install the core software library of the development board.

😁 Boards Manager		X
Type All 🔻	Filter your search	
More Into		
Arduino Wbed OS Edge H by Arduino Boards included in this package: Arduino Edge Control. Online Help More Info	Goards 4.1.5 ▼ Install	]
Arduino Nbed OS Giga E by Arduino Boards included in this package: Arduino Giga. Online Help More Info	Goards	Ŧ
· ·	C	lose

Figure 3.9 Arduino IDE1 development board Manager

G. Port is to select the serial port connected to the development board, as shown

in the following figure, the serial port needs to be connected to the

development board will be displayed:

💿 sketch_aug02c   A	rduino 1.8.19		
File Edit Sketch Too	ols Help		
	Auto Format	Ctrl+T	
	Archive Sketch		
sketch_aug02c	Fix Encoding & Reload		
1 void setu	Manage Libraries	Ctrl+Shift+I	
2 // put 3	Serial Monitor	Ctrl+Shift+M	
4 }	Serial Plotter	Ctrl+Shift+L	
5 6 void loop	WiFi101 / WiFiNINA Firmware Updater		
7 // put	Board: "Arduino Uno"	Þ	
8	Port: "COM55 (Elecrow CrowPanel 7.0P)"		Serial ports
9 }	Get Board Info		COM1
	Programmer		✓ COM55 (Elecrow CrowPanel 7.0P)
	Burn Bootloader		
1			

Figure 3.10 Arduino IDE1 port

Some of the other options are not used, so keep the default Settings.

#### 3.1.5. Help Menu

The help menu interface is shown in the following figure, which mainly links to the software usage instructions on the official website.

File Edit Sketch Tools Help Getting Started sketch_aug02c void setup() 2 // put your 3 4 } 5 void loop() { 7 // put your 8 9 } Getting Started Environment Troubleshooting Reference Find in Reference Ctrl+Shift+F Frequently Asked Questions Visit Arduino.cc About Arduino * *	💿 sketch_aug02c   Ardui	ino 1.8.19	
sketch_aug02c   1   void setup()   2   2   4   3   5   6   void loop() {   7   7   7   7   9	File Edit Sketch Tools	Help	
9 }	<pre>sketch_aug02c 1 void setup() 2 // put your 3 4 } 5 6 void loop() { 7 // put your</pre>	Environment Troubleshooting Reference Find in Reference Ctrl+Shift+F Frequently Asked Questions Visit Arduino.cc	
	9 }		~
1 Arduino Uno on COM55			

Figure 3.11 Arduino IDE1 Help menu

# 3.2. Tool bar

The tool bar interface is shown as follows:

💿 sketch_aug02c   Arduino 1.8.19	
<u>Eile E</u> dit <u>S</u> ketch <u>T</u> ools <u>H</u> elp	
	Ø.
	· · · · · · · · · · · · · · · · · · ·
sken_an02 C Diz setup() { C Diz setup(	- Serial
4 } 5 6 void loop() {	Serial Monitor
7 // put your main code here, to run repeatedly:	
8	
9 }	-

Figure 3.12 Arduino IDE1 toolbar

- A. **Verify**: compile and check whether the program is correct, if correct, compile through, generate binary files.
- B. Upload: the compiler generates binary files and uploads them to the MCU of the development board.
- C. New, open, save: create, open or save a project file.
- D. Serial Monitor: The serial port window is displayed.

# 4. Install the Arduino-ESP32 core software library

Arduino-esp32 Core software library is a plug-in of Arduino platform, which is the Arduino platform ESP32 chip. The software development provides low-level support. Because the Arduino IDE does not support ESP32 by default, you must install the Arduino-ESP32 core software library.

There are two ways to install the Arduino-ESP32 core software library: online installation of the Arduino IDE development board manager and manual offline installation.

# 4.1. Arduino IDE development board Manager is installed online

A. Open the Arduino IDE software, click **Tools** -> **Board** -> **Boards Manager**, and input ESP32 in the search bar when the information of the development board is

loaded, the ESP32 search results will appear, as shown in the figure below. **Note**: If the search is less than ESP32 core library, you may need to click **file** - >

preferences, the attachment in the Additional Boards Manager URLs

"https://espressif.github.io/arduino-esp32/package\_esp32\_index.json" in the

input, Then follow the above steps to search.

ype All	▼ ESP32					
Arduino ESP32 Bo	oards					-
by <b>Arduino</b> Boards included in this p Arduino Nano ESP32.	oackage:					
More Info						
oan20						
-			-			
esp32 by Espressif Systems Boards included in this p		2 Day Board 55022-02	Dour Roard Arduino	Naco 55022		
by Espressif Systems Boards included in this p		33 Dev Board, ESP32-C3 I	Dev Board, Arduino	Nano ESP32.		
by <b>Espressif Systems</b> Boards included in this p ESP32 Dev Board, ESP32		33 Dev Board, ESP32-C3 I	Dev Board, Arduino	Nano ESP32.	Install	
by <b>Espressif Systems</b> Boards included in this p ESP32 Dev Board, ESP32		33 Dev Board, ESP32-C3 I	Dev Board, Arduino		Install	
by <b>Espressif Systems</b> Boards included in this p ESP32 Dev Board, ESP32		33 Dev Board, ESP32-C3 I	Dev Board, Arduino	3. 0. 3 ▼ 3. 0. 3 ▲ 3. 0. 2 =	Install	
by <b>Espressif Systems</b> Boards included in this p ESP32 Dev Board, ESP32		33 Dev Board, ESP32-C3 I	Dev Board, Arduino	3. 0. 3 ▼ 3. 0. 3 ▲ 3. 0. 2 3. 0. 1	Install	
by <b>Espressif Systems</b> Boards included in this p ESP32 Dev Board, ESP32		33 Dev Board, ESP32-C3 I	Dev Board, Arduino	3. 0. 3 ▼ 3. 0. 3 ▲ 3. 0. 2 =	Install	

Figure 4.1 Arduino-ESP32 core software library search

B. select "esp32 by Espressif Systems", then select the version number, and finally click Install, as shown below.

**Note**: Version 3.0 was developed based on ESP32 idf 5.1 and version 2.0 was developed based on ESP32 idf 4.4. The two versions have different apis for Bluetooth, timer, I2S driver, LEDC driver, timer, and other software. Therefore, the example of version 2.0 involves the above apis. If the 3.0 version is used, an error will be reported. Pay attention to the version selection.

The installation takes a long time, and the download may fail during the installation. Therefore, you need to try the installation several times.

The downloaded installation files are compressed and saved in

# "C:\Users\Administrator\AppData\Local\Arduino15\staging\packages"

directory(The red part is the actual user name of the computer, **AppData** directory is a hidden directory, you need to click the folder menu bar **tools** ->

www.lcdwiki.com

Folder options -> View -> select Show hidden files, folders and drives, and

then click **OK** to save)

😔 Boards Manager	×	
Type All 💌 ESP32		
Arduino ESP32 Boards		
by <b>Arduino</b> Boards included in this package: Arduino Nano ESP32. <u>More Info</u>		
esp32 by Espressif Systems Boards included in this package: ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, ESP32-C3 Dev Board, Arduino Nano ESP32. More Info	Installing	
	-	
Verifying archive integrity	Cancel	

Figure 4.2 Installation of Arduino-ESP32 core software library

C. After installation, close the development board manager, click Tools -> Board,

you can see the "ESP32 Arduino" option, click this option, you can see many

ESP32 development boards, as shown in the following picture:

💿 sketch_aug02a   /	Arduino 1.8.19		And and the survey of the survey of	
File Edit Sketch To	pols Help			
sketch_aug02a	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T		
1 void setu 2 // put 3 4 } 5	Manage Libraries Serial Monitor Serial Plotter WiFi101 / WiFiNINA Firmware Upd	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L		
6 void loop 7 // put 8 9 }	Board: *Arduino Uno* Port Get Board Info Programmer Burn Bootloader		Boards Manager Arduino AVR Boards Arduino Renesas UNO R4 Boards Arduino SAMD (32-bits ARM Cortex-M0+) Boards ESP32 Arduino	∆     ESP32H2 Dev Module     ESP32C6 Dev Module     ESP32C3 Dev Module     ESP32C3 Dev Module     ESP32C3 Dev Module     ESP32S2 Dev Module     ESP322 Dev Module     ESP322Wrover Module     ESP32 Wrover Module     ESP32 PICO-D4     ESP32S3 Dev Module Octal (WROOM2)     ESP32S3 Dev Module Octal (WROOM2)     ESP32S3 CAM LCD     ESP32S3 CAM LCD     ESP32S2 Native US8     ESP32S2 Native US8     ESP32 Wrover Kit (all versions)     Aventen S3 Sync     UM BLING

Figure 4.3 Selection of ESP32 development board

# 4.2. Manual Offline installation

The Developer Board Manager online installation is to download the installation

```
www.lcdwiki.com
```

file zip from github, and then unzip the installation. In a poor network environment, access to github may fail, resulting in an online installation failure. A manual offline installation method is provided as follows:

**Note**: First make sure you have 7-Zip installed on your computer.

A. Download the installation file package from the following URL:

http://www.lcdwiki.com/res/Arduino ESP32/Arduino esp32 3.0.3.exe http://www.lcdwiki.com/res/Arduino ESP32/Arduino esp32 2.0.17.exe

Among them, 3.0.3 is ESP32 core software library 3.0 version, 2.0.17 is ESP32 core software library 2.0 version, choose to download according to demand.

B. After the installation file is downloaded, double-click the file to pop up the decompression window, and enter

"C:\Users\Administrator\AppData\Local\Arduino15\packages" in the Extract to text box (the red part is the actual user name of the computer). Then click the "Extract" button to extract and install the file, as shown below:

**Note**: If the **esp32** folder already exists in the destination folder, you need to delete the **esp32** folder before decompressing the file.

2 7-Zip self-extracting archive	X
Extract to:	
C:\Users\Administrator\AppData\Local\Arduino15\packages	
Extract Cance	

Figure 4.4 Offline installation of Arduino-ESP32 core software library

C. After the file is installed, open the Arduino IED software again, click **Tools** -> **Development Board**, you can see the "**ESP32 Arduino**" option, click this option, you can see many ESP32 development boards, which is consistent with the online installation step C.

# 5. Compile, download and run the ESP32 sample program

#### 5.1. Install the USB-to-serial IC driver

When using the serial port to download the program, you need to install the USB to serial port IC driver on the computer, if no, the computer cannot identify the serial port. Different development boards use different USB-to-serial IC, and corresponding drivers need to be installed. The CH340C USB-to-serial driver IC is used here, so to install the CH340 driver, the steps are as follows (if already installed, you can skip the following steps) :

A. Locate the **USB-SERIAL\_CH340.zip** package in the **7-Tool\_software** folder and decompress it.

B. go to the folder after decompression, double-click "**CH341SER.EXE**" executable program, pop up the installation window, and then click "Install" button to continue the installation, as shown in the following picture:

🔒 驱动安装()	X64)	
驱动安装	麦/卸载	
选择INI	F文件:	CH341SER.INF
3	安装	WCH.CN
Ť.	甲载	<u> </u>
書	帮助	

Figure 5.1 Installing the CH340C driver

C. After the installation is successful, click the window OK button to exit. Connect the computer USB to the development board and power on, and then enter the computer device manager, you can see that the CH340 port is identified under the port, as shown in the following picture:

→ 设备管理器	X
文件(F) 操作(A) 查看(V) 帮助(H)	
CSL-PC	
▷ · 🕞 IDE ATA/ATAPI 控制器	
▶ 🚔 WSD 打印提供程序	
▶ 🛄 处理器	
▶□ 磁盘驱动器	
┛ ‴ 端口 (COM 和 LPT)	
USB-SERIAL CH340 (COM55)	
「「」通信端口 (COM1)	
▶ - 場 计算机	
▶ - 型 监视器	
▶	
▶ 🖏 人体学输入设备	
▶ ⑩ 声音、视频和游戏控制器	
▶ ② 鼠标和其他指针设备	

Figure 5.2 Identifying CH340 ports

# 5.2. Configure the development board

After you create or open an existing sample program in the Arduino IDE, you first need to configure the development board. The steps are as follows:

A. Connect the development board to the USB port of the computer and power on,

then select the model of the target development board, here select ESP32, click the

"Tools" button, select Board ->ESP32 Arduino->ESP32 Dev Module,As shown in the picture below:

ile Edit Sketch T				
sketch_aug02a void setu // put }	Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter WiFi101 / WiFiNINA Firmware U	Ctrl+T Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L odater		
6 void loop 7 // put	Board: "Arduino Uno"		Boards Manager	Δ
8	Port	1	Arduino AVR Boards	ESP32H2 Dev Module
9 }	Get Board Info		Arduino Renesas UNO R4 Boards	ESP32C6 Dev Module
	Programmer		Arduino SAMD (32-bits ARM Cortex-M0+) Boards	ESP32S3 Dev Module
	Burn Bootloader		ESP32 Arduino	ESP32C3 Dev Module
				ESP32S2 Dev Module
			L	ESP32 Dev Module ESP32-WROOM-DA Module
				ESP32-WROOM-DA Module ESP32 Wrover Module
				ESP32 Wrover Wodule ESP32 PICO-D4
				ESP32S3 Dev Module Octal (WROOM2)
				ESP32-S3-Box
				ESP32-S3-USB-OTG
				ESP32S3 CAM LCD
				ESP32S2 Native USB
				ESP32 Wrover Kit (all versions)
				Aventen S3 Sync
				UM BLING

Figure 5.3 Select ESP32 development board model

B. Click the "Tools" button, you can see the default configuration of ESP32

development board, as shown in the following picture:

#### LCDWIKI ESP32 Arduino IDE1 development environment construction CR20

💿 sketch_aug02a   A		
File Edit Sketch To	ols Help	
	Auto Format Archive Sketch	Ctrl+T
sketch_aug02a	Fix Encoding & Reload	
1 void setu	Manage Libraries	Ctrl+Shift+I
2 // put	Serial Monitor	Ctrl+Shift+M
3 4 }	Serial Plotter	Ctrl+Shift+L
5 6 void loop	WiFi101 / WiFiNINA Firmware Updater	
7 // put	Board: "ESP32 Dev Module"	•
8	Upload Speed: "921600"	+
9}	CPU Frequency: "240MHz (WiFi/BT)"	+
	Flash Frequency: "80MHz"	+
	Flash Mode: "QIO"	÷.
	Flash Size: "4MB (32Mb)"	+
	Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"	+
	Core Debug Level: "None"	+
	PSRAM: "Disabled"	+
	Arduino Runs On: "Core 1"	+
	Events Run On: "Core 1"	+
	Erase All Flash Before Sketch Upload: "Disabled"	+
	JTAG Adapter: "Disabled"	•
	Zigbee Mode: "Disabled"	•
	Port: "COM55 (Elecrow CrowPanel 7.0P)"	•
	Get Board Info	
	Programmer	Þ
	Burn Bootloader	

Figure 5.4 ESP32 development board configuration

Here are the configuration parameters:

Upload Speed: Code upload rate. Available parameters are 51200, 230400,

256000, 115200, and 921600.Select according to the maximum rate supported by the USB-to-serial port on the development board, such as the one used here The CH340C supports a maximum rate of 2Mbps. Therefore, the maximum rate is 921600.

CPU Frequency: CPU clock frequency, optional parameters: 240MHz(WiFi/BT),

160MHz(WiFi/BT),80MHz(WiFi/BT), 40MHz, 26MHz, 20MHz, 13MHz, 10MHz. Generally speaking,The higher the frequency, the higher the power consumption, which can be selected according to the demand. Power consumption is not considered here, direct selection Select a maximum frequency of 240MHz for optimal performance. Points to note: 240MHz, 160MHz,These three frequencies of 80MHz can ensure the normal operation of WiFi and BT, and other frequencies cannot be guaranteed Verify the normal operation of WiFi and BT, and only ensure the basic functions of the CPU.

- Flash Frequency: Clock frequency of the Flash SPI bus mounted on the ESP32. Optional parameters are: 80MHz,40MHz. In order to improve the Flash read and write speed, the high frequency of 80MHz is generally selected.
- Flash Mode: Flash communication mode mounted on the ESP32. Available parameters are QIO and DIO. QIO uses four SPI data lines for Flash writing and reading. DIO uses two SPI data lines for Flash writing and reading. Select the connection mode based on the actual Flash connection mode. Here we use 4 SPI data lines for Flash write and read, so QIO is selected.
- Flash Size: Flash capacity mounted on ESP32. The value can be 4MB(32Mb), 8MB(64Mb), or 4MB(32 MB),2MB(16Mb), 16MB(128Mb). The value is selected based on the actual capacity of the Flash. The Flash is 4MB, so choose 4MB(32Mb).
- Partition Scheme: Method of mounting Flash space partition on ESP32. To make better use of Flash,The Arduino IDE has designed more than a dozen partitioning methods, which are not introduced here. If you are interested, you can learn by yourself. The Flash used here is 4MB. Generally, "Default 4MB with spiffs(1.2MB APP/1.5MB SPIFFS)" is selected. If there are many project files, the compiled binary file is relatively large. You can select "Huge APP(3MB No OTA/1MB SPIFFS)".

Core Debug Level: Arduino kernel debug log level, output through the serial port. The options are as follows: None,Errors, Warn, Info, Debug, and Verbose. Among them:

None: No debug log is displayed;

Error: Only debug logs with error levels are output;

Warn: Only debug logs of warning or higher levels are

www.lcdwiki.com

generated;

Info: Only debugging logs of the information and higher levels are displayed;

Debug: Displays only debug logs of a higher level;

Verbose: Displays debugging logs of all levels in kernel

debugging;

In general, you do not need to pay attention to kernel debug logs unless you are developing some kernel-related functionality. So the choice here is nothing.

- PSRAM: The options are Disabled and Enabled. Some ESP32 divide In addition to the built-in SRAM, external PSRAM is attached for memory expansion. In this case, select Enabled.Some ESP32s only have internal SRAM. In this case, select Disabled. The ESP32 used here does not have PSRAM mounted, so select Disabled.
- Arduino Runs On: Configure the ESP32 kernel that runs the Arduino Core task code. The optional parameters are Core0 and Core1. The ESP32 has two cores, Core0 and Core1, each of which can run different code tasks. You can choose according to the situation, the default is Core1.
- Event Runs On: Configure the ESP32 kernel that the Arduino interrupt event runs on. The optional parameters are Core0 and Core1. You can choose according to the situation, the default is Core1. The kernel configured here can be the same as that configured in Arduino Runs On, or it can be different. When configured in the same way, the ESP32 power consumption can be reduced. When configured differently, the efficiency of the program can be improved.

Erase ALL Flash Before Sketch Upload: Specifies whether to erase the entire Flash when uploading code. The options are Disabled and Enabled. If Disabled is selected, full erasure is not required, and

if Enabled is selected, full erasure is required. If full erase is selected, the rate of code upload slows down. In addition, frequent full erase affects the service life of the Flash. Therefore, Disabled is selected.

JTAG Adapter: The options are Disabled, FTDI Adapter, and ESP USB Bridge. Debugging code with JTAG is easier, but the Arduino IDE does not support ESP32 debugging. So select "Disabled".

**PORT**: Select the serial port number connected to the ESP32 development board.

Generally, it will be automatically identified.

#### 5.3. Compile, download, and run the program

The example that comes with the Arduino-ESP32 core software library is used as a demonstration. You can also create a new project to compile, download and run, and you can open an already completed project to operate.

A. Open Arduino IDE, click the "File" button, and select

Examples->ESP32->ChipID->GetChipID, as shown in the following picture:

😊 sketc	h_aug02a	Arduino 1.8.19	)					-	designment of the local division of	
File Edit Sketch Tools Help										
Ope		Ctrl+N Ctrl+O								
_	mples					:				
Clo: Sav	se e	Ctrl+W Ctrl+S Ctrl+Shift+S		DNSServer EEPROM ESP Insights ESP RainMaker	> > >					
Pag Prin		Ctrl+Shift+P Ctrl+P	Ģ	ESP32 ESP32 Async UDP	1		AnalogOut AnalogRead	Þ		
Pre	ferences	Ctrl+Comma		ESP_I2S ESP_NOW			AnalogReadContinu ArduinoStackSize	ous		
Qui	t	Ctrl+Q		ESP_SR			Camera	ŀ		
				ESPmDNS			ChipID		GetChipID	
				Ethernet	1		CI	Þ		
				FFat	1		DeepSleep	Þ		
				HTTPClient			FreeRTOS	Þ		
				HTTPUpdate			GPIO	Þ		
				HTTPUpdateServe	er i		HWCDC_Events			
				LittleFS	1		MacAddress	Þ		
				NetBIOS	1		ResetReason	Þ		

Figure 5.5 ESP32 sample program

B. click the "Upload" button, you can see "compiling Sketch..." Prompt, as

shown in the picture below:



Figure 5.6 ESP32 project compilation

C. After the compilation is successful, "**Uploading...**" will be displayed. At the same time, the information output window will output a compilation success message, as shown in the following figure:

😇 GetChipID   Arduino 1.8.19	
Eile Edit Sketch Tools Help	
	ø
GetChipID	
<pre>1 /* The true ESP32 chip ID is essentially its MAC address. 2 This sketch provides an alternate chip ID that matches 3 the output of the ESP.getChipId() function on ESP8266 4 (i.e. a 32-bit integer matching the last 3 bytes of 5 the MAC address. This is less unique than the 6 MAC address chip ID, but is helpful when you need 7 an identifier that can be no more than a 32-bit integer 8 (like for switchcase). </pre>	*
Uploading Wrote 0x4000000 bytes to file C:\Users\ADMINI~1\AppData\Local\Temp\ardu "C:\\Users\\Administrator\\AppData\Local\\Arduino15\\packages\\esp32\ Sketch uses 283433 bytes (21%) of program storage space. Maximum is 13 Slobal variables use 20248 bytes (6%) of dynamic memory, leaving 30743 C:\Users\Administrator\AppData\Local\Arduino15\packages\esp32\tools\esp	<u>\\tools\\esp-x3</u> 310720 bytes.

Figure 5.7 ESP32 project compiled successfully

D. After the upload is successful, the message "**Upload Success**" will appear, and the information output window will output the uploaded information and the program operation prompt, as shown in the following figure:

💿 GetChipID   Arduino 1.8.19	x
<u>Eile E</u> dit <u>S</u> ketch <u>T</u> ools <u>H</u> elp	
	Ø
GetChipID	
<pre>1 /* The true ESP32 chip ID is essentially its MAC address. 2 This sketch provides an alternate chip ID that matches 3 the output of the ESP.getChipId() function on ESP8266 4 (i.e. a 32-bit integer matching the last 3 bytes of 5 the MAC address. This is less unique than the 6 MAC address chip ID, but is helpful when you need 7 an identifier that can be no more than a 32-bit integer 8 (like for switchcase).</pre>	+ III
•	Þ
Done uploading.         Writing at 0x0004c297 (90 %)         Writing at 0x0005180a (100 %)         Wrote 283792 bytes (158496 compressed) at 0x00010000 in 2.6 seconds (effective 862.         Hash of data verified.         Leaving         Hard resetting via RTS pin	4
visbled, Disabled, Default 4MB with spiffs (1 2MB APP/1 5MB SPIFFS), 240MHz (WIFIBT), QLO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled, Disabled on COM	155

Figure 5.8 ESP32 project uploaded and running successfully

E. click the menu bar tool -> Serial port monitor, the serial port interface pops

up, set the baud rate to 115200, you can see the serial terminal information

output, then the program runs successfully, as shown in the following picture:

© COM55			
			Send
ESP32 Chip model = ESP32-DOWD Rev 301			
This chip has 2 cores			
Chip ID: 5369700			
	r		
🔽 Autoscroll 🔲 Show timestamp	Newline -	115200 baud 🔻	Clear output

Figure 5.9 Serial port output of ESP32 program