
Hong Ke Technology

kai

Nov 11, 2020

LANGUAGES/

1	Switch Between Languages/	1
2	Arduino Construction of development environment	3
3	ESP-IDF Construction of development environment	15
4	W800 Construction of development environment	17
5	WiFi	23
6	Bluetooth	25
7	Zigbee	27
8	LoRa	29
9	Sub1G	31
10	2.4G	33
11	Multi-Protocol	35
12	HK-D1-MiniPro4M	37
13	HK-D1-MiniPro16M	39
14	HK-ESP32-CAM	41
15	HK-ESP32-CAM-MB	47
16	HK-ESP32-MiniKit(BLACK)	49
17	HK-ESP32-MiniKit(BLUE)	51
18	HK-ESPTB	53
19	HK-RFTB	55

SWITCH BETWEEN LANGUAGES/

There are now versions in two languages.

- English/
- Chinese/

You can easily change from one language to another by the panel on the sidebar like below. Just click on the Read the Docs title button on the left-bottom corner if it is folded. En is an English docs and zh_CN is a Chinese docs

Read the Docs enzh_CN

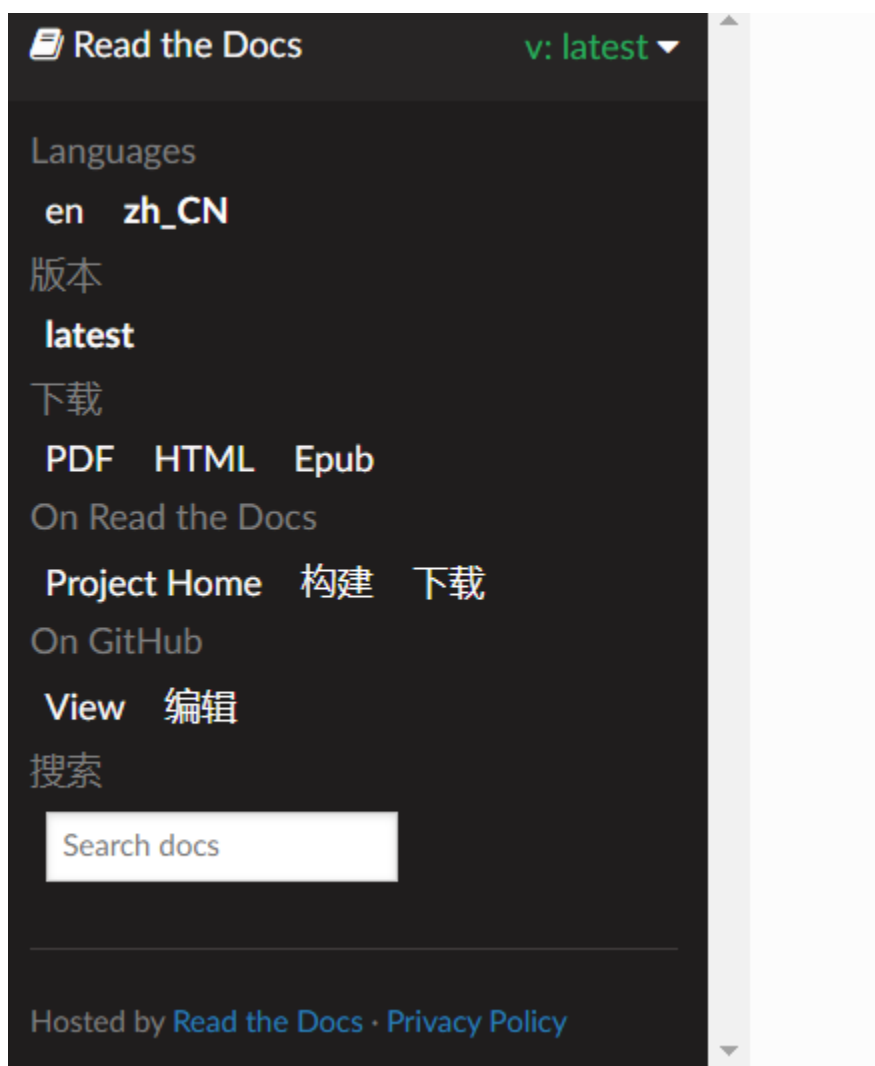


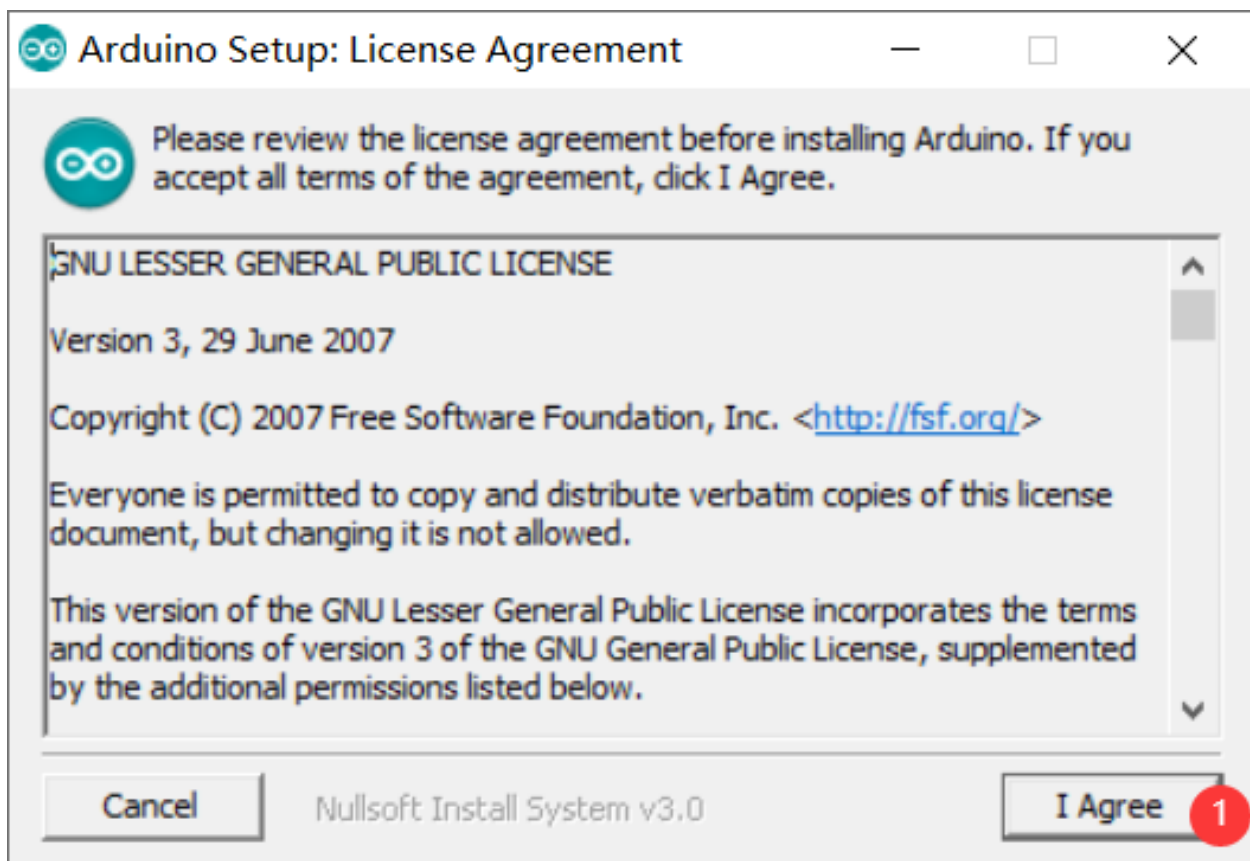
Fig. 1: Switch Between Languages/

ARDUINO CONSTRUCTION OF DEVELOPMENT ENVIRONMENT

2.1 Arduino Construction of development environment(ESP32)

You need to install Arduino, GIT, Python and other tools on your computer. The installation software section will not cover too much.

1Download and install arduinoide from the Arduino official website. Open the download package and click I agree

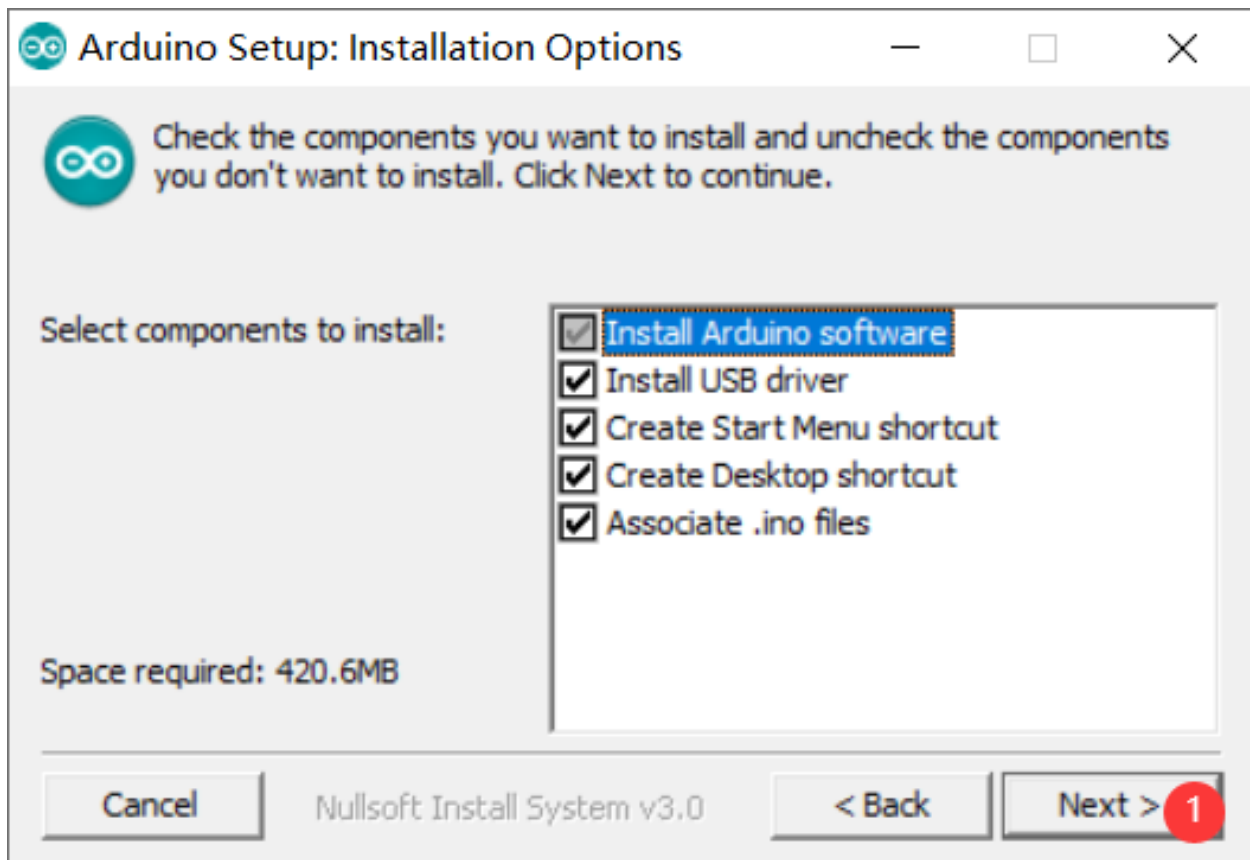


2Click Next

3The default is to install on the C drive. You can change the installation location by clicking Browse.

4I installed it under the Arduino folder on disk D (remember this installation path)

5Wait for the installation to complete. Turn on the Hardware under the installation path



6Create a new folder espressif

7Go to espressif. Create another folder for esp32

8Open Git Bash and go to this directory (if you already have Git installed)

Note: Note that the paths are “/” delimited

9Command line inputgit clone <https://github.com/espressif/arduino-esp32.git>

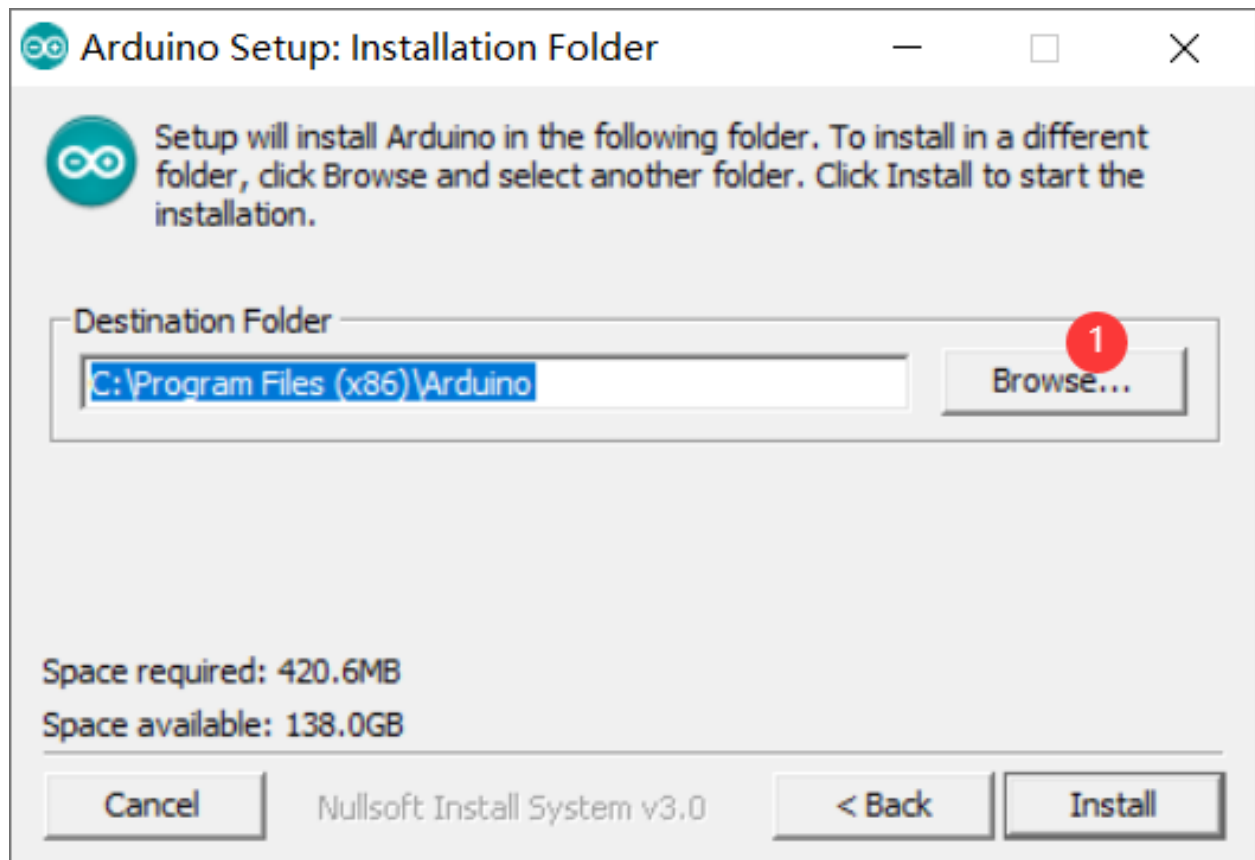
10Waiting for download to succeedOpen the arduino-esp32 in the espressif folder .open tools folder

11Double-click get.exeYou need Python installed on your computer.

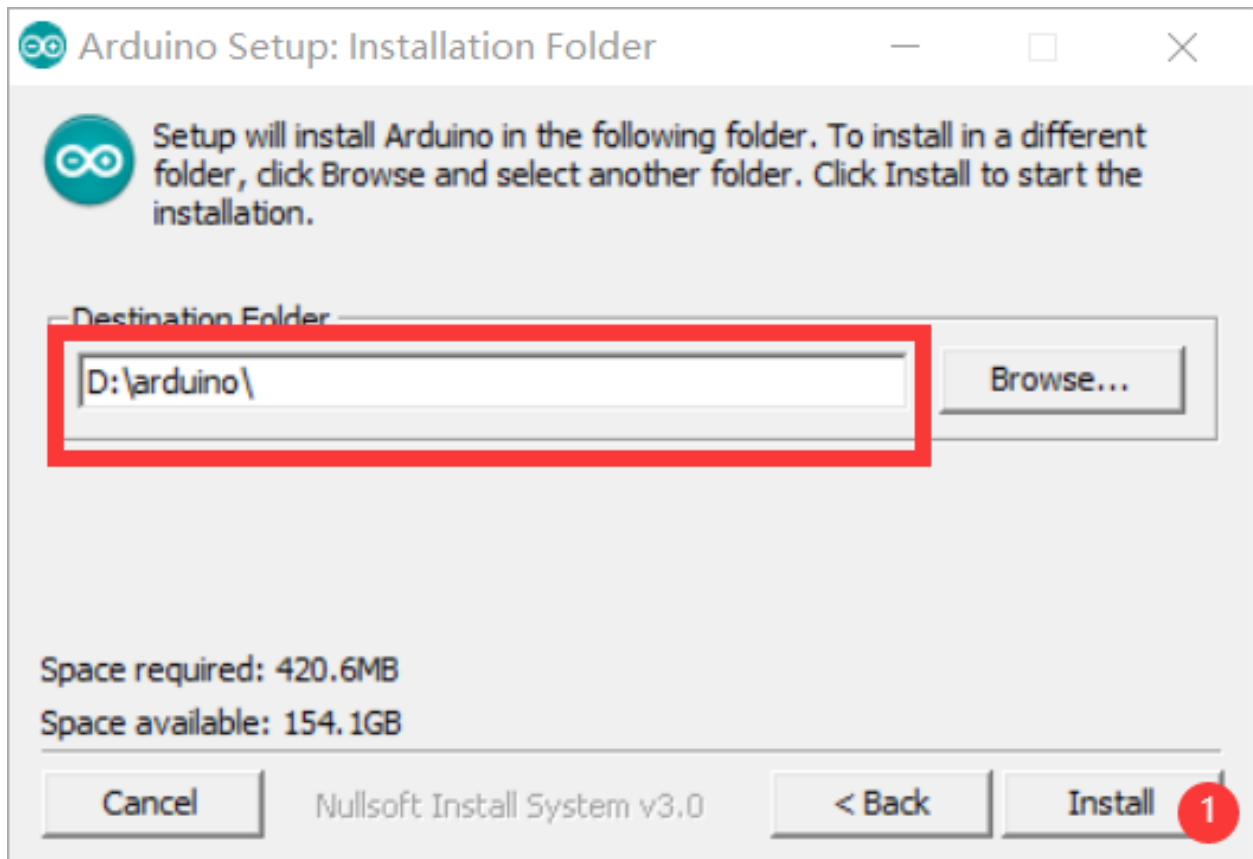
12Open ArduinoIDeselect development board

13Development board configuration

14Arduino development environment is built.You can select the appropriate example to burn to the appropriate development board









2.2 Arduino Construction of development environment(ESP8266)



data (D:) > arduino					▼	↺	🔍 搜索"ardi
名称	^	修改日期	类型	大小			
drivers		2020/11/6 14:20	文件夹				
examples		2020/11/6 14:20	文件夹				
hardware		2020/11/6 14:20	文件夹		1		
java		2020/11/6 14:20	文件夹				
lib		2020/11/6 14:20	文件夹				
libraries		2020/11/6 14:20	文件夹				
reference		2020/11/6 14:20	文件夹				
tools		2020/11/6 14:20	文件夹				
tools-builder		2020/11/6 14:20	文件夹				
arduino.exe		2017/8/18 22:55	应用程序	395 KB			
arduino.l4j.ini		2017/8/18 22:55	配置设置	1 KB			
arduino_debug.exe		2017/8/18 22:55	应用程序	393 KB			
arduino_debug.l4j.ini		2017/8/18 22:55	配置设置	1 KB			
arduino-1.8.4-windows.exe		2020/8/26 14:24	应用程序	92,546 KB			
arduino-builder.exe		2017/8/18 22:55	应用程序	3,214 KB			
libusb0.dll		2017/8/18 22:55	应用程序扩展	43 KB			
msvcp100.dll		2017/8/18 22:55	应用程序扩展	412 KB			
msvcr100.dll		2017/8/18 22:55	应用程序扩展	753 KB			
revisions.txt		2017/8/18 22:55	TXT 文件	84 KB			
uninstall.exe		2020/11/6 14:20	应用程序	404 KB			
wrapper-manifest.xml		2017/8/18 22:55	XML 文档	1 KB			

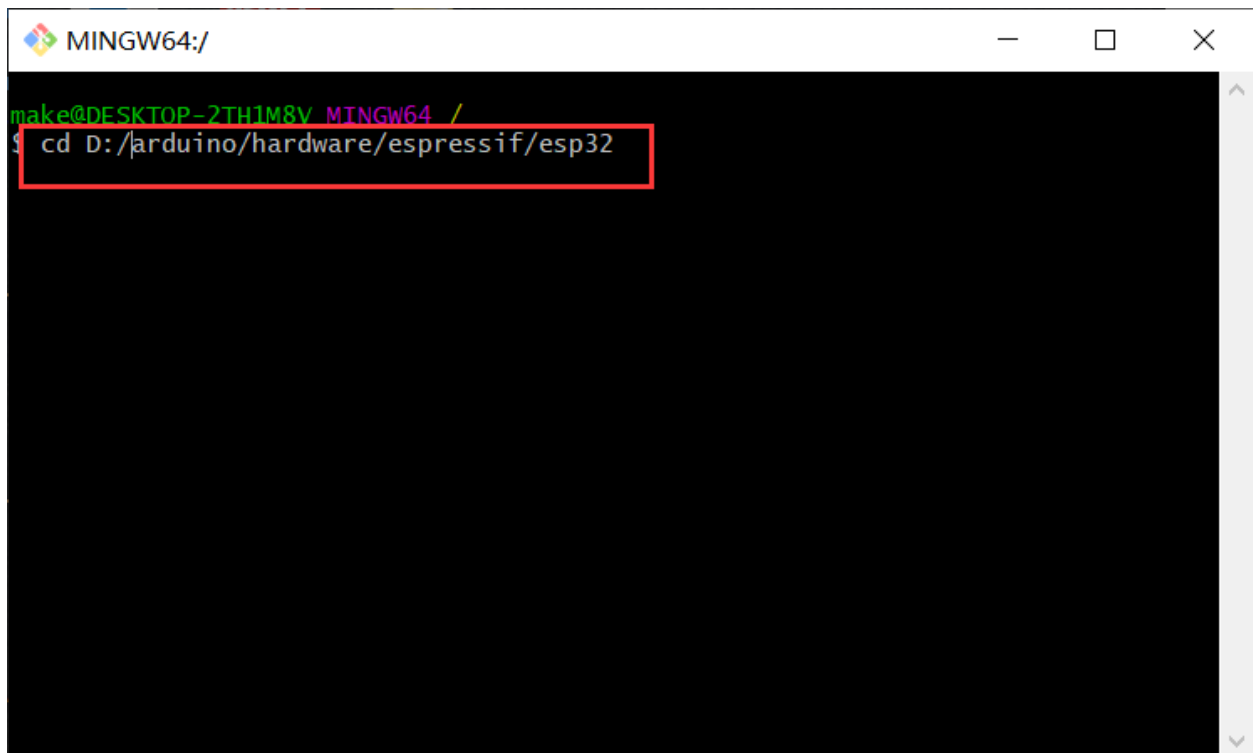
窗 > data (D:) > arduino > hardware > ▼ 🔄

名称	修改日期	类型	大小
 arduino	2020/11/6 14:43	文件夹	
 espressif	2020/11/6 14:44	文件夹	
 tools	2020/11/6 14:43	文件夹	
 package_index_bundled.json	2017/8/18 22:55	JSON File	9 KB
 platform.keys.rewrite.txt	2017/8/18 22:55	TXT 文件	8 KB
 platform.txt	2017/8/18 22:55	TXT 文件	1 KB

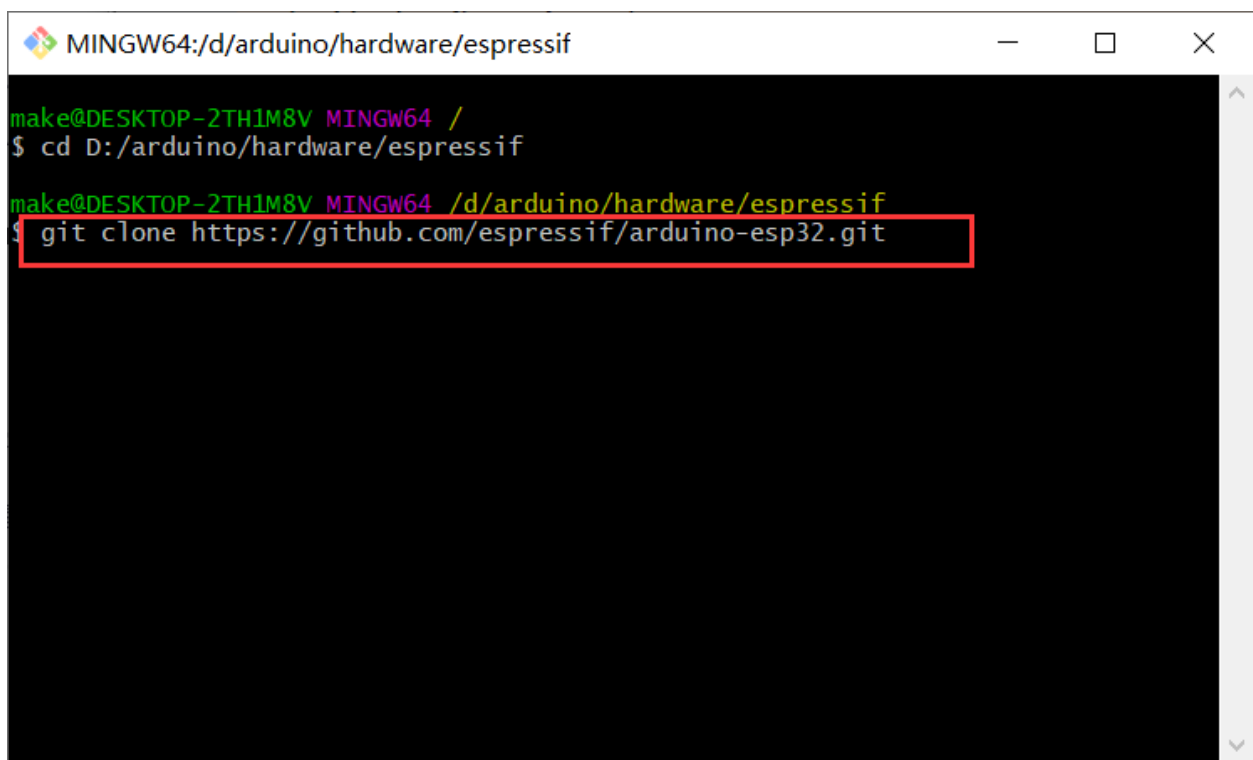


电脑 > data (D:) > arduino > hardware > espressif

名称	修改日期	类型	大小
esp32	2020/8/31 23:06	文件夹	



```
MINGW64:/  
make@DESKTOP-2TH1M8V MINGW64 /  
$ cd D:/arduino/hardware/espressif/esp32
```



A screenshot of a MINGW64 terminal window. The title bar shows the path "MINGW64:/d/arduino/hardware/espressif". The terminal content shows the following commands and prompts:

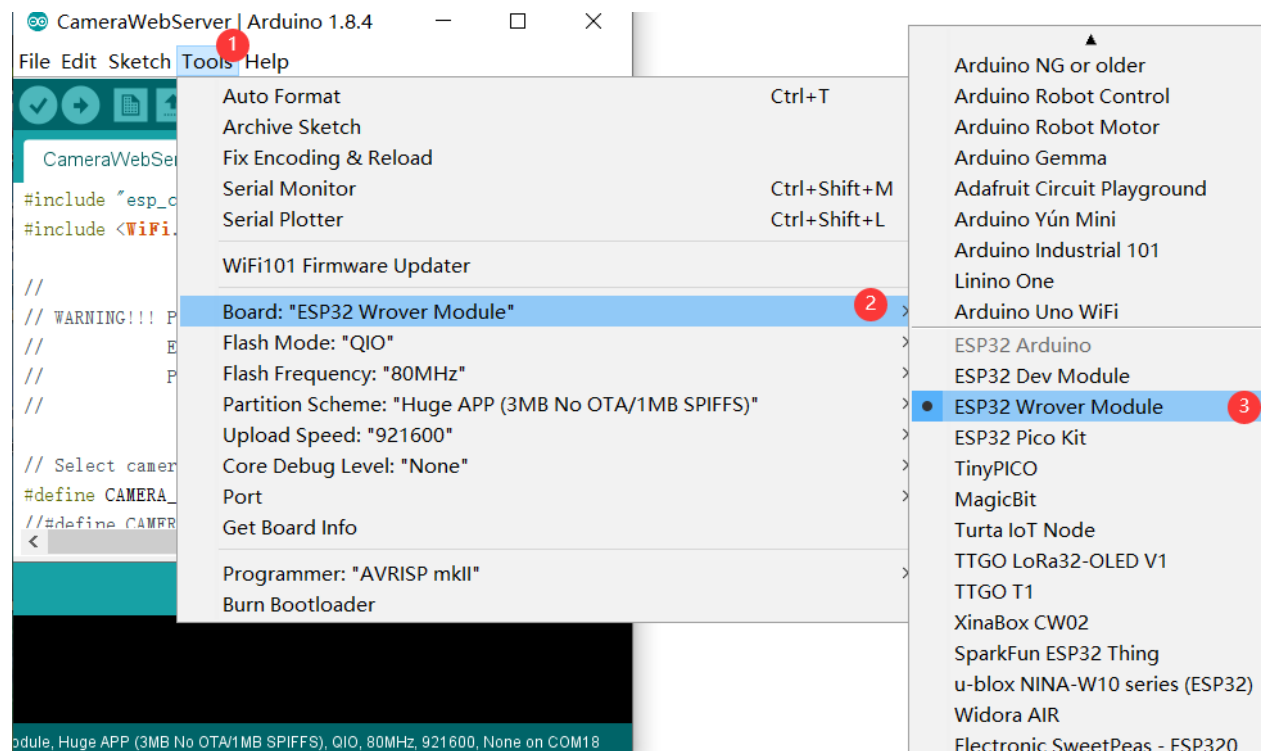
```
make@DESKTOP-2TH1M8V MINGW64 /  
$ cd D:/arduino/hardware/espressif  
make@DESKTOP-2TH1M8V MINGW64 /d/arduino/hardware/espressif  
$ git clone https://github.com/espressif/arduino-esp32.git
```

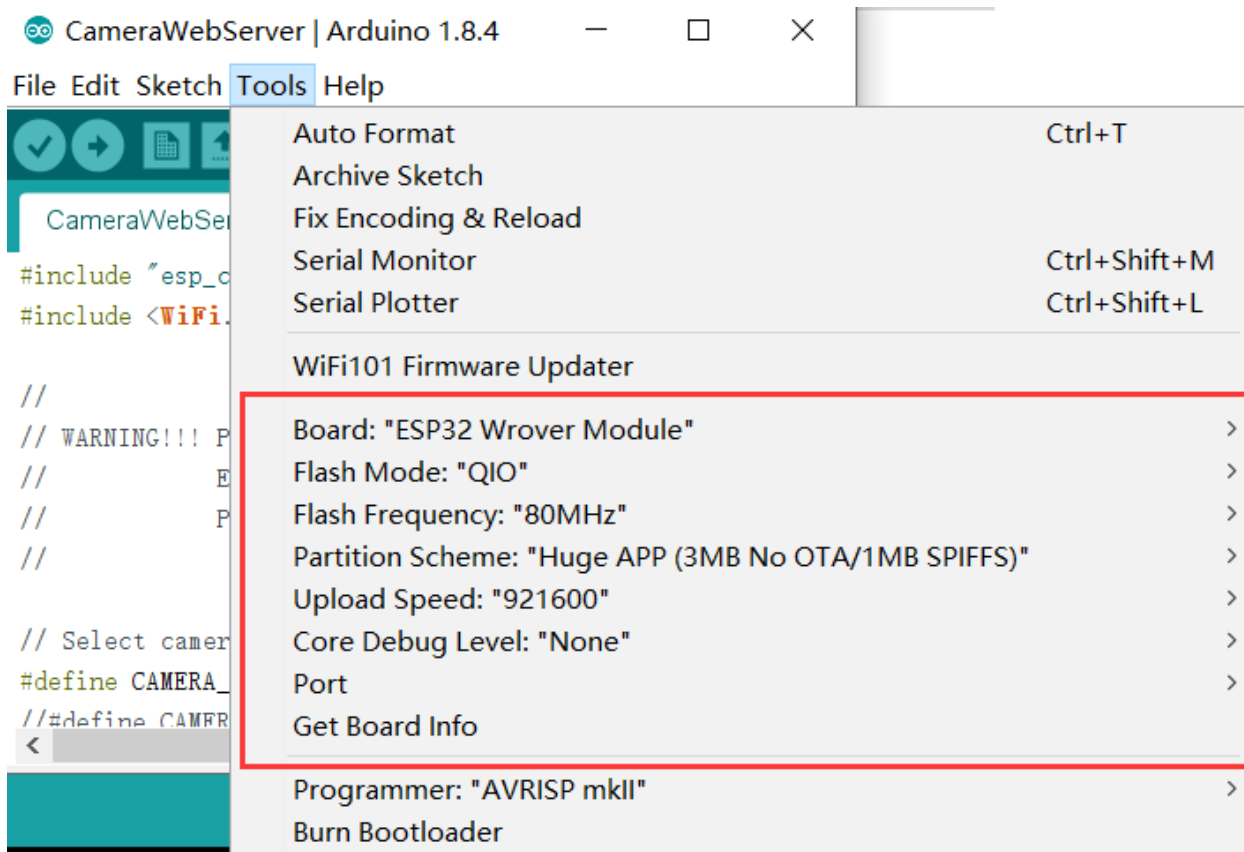
The last command, `git clone https://github.com/espressif/arduino-esp32.git`, is highlighted with a red rectangular box.

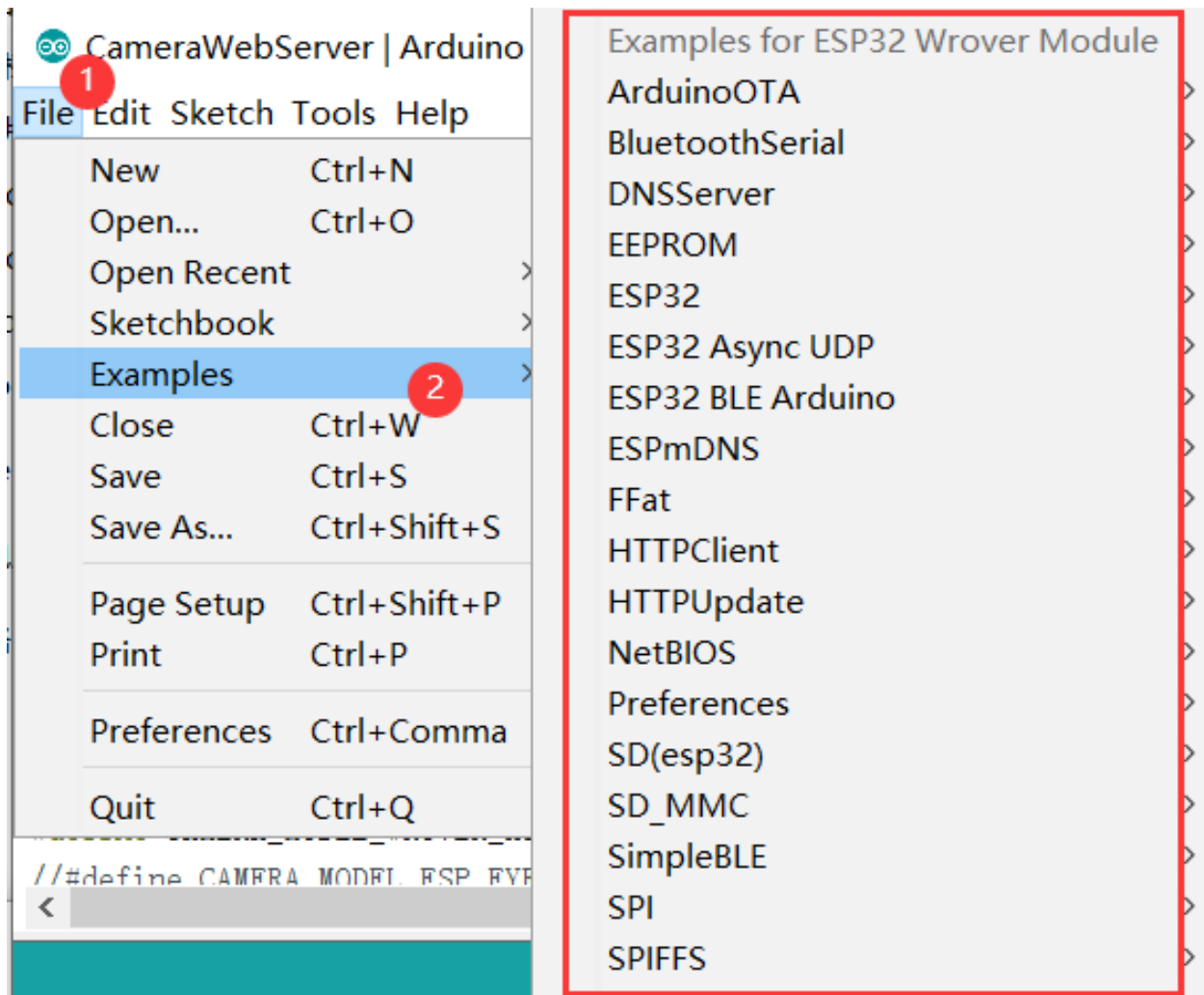
脑 > data (D:) > arduino > hardware > espressif > esp32					▼	🔄
名称	修改日期	类型	大小			
.github	2020/8/31 23:06	文件夹				
cores	2020/8/31 23:06	文件夹				
docs	2020/8/31 23:06	文件夹				
libraries	2020/8/31 23:06	文件夹				
package	2020/8/31 23:06	文件夹				
tools	2020/11/6 15:05	文件夹				
variants	2020/8/31 23:06	文件夹				
.gitignore	2020/8/31 23:06	GITIGNORE 文件	1 KB			
.gitmodules	2020/8/31 23:06	GITMODULES 文件	2 KB			
.travis.yml	2020/8/31 23:06	YML 文件	202 KB			
boards.txt	2020/8/31 23:06	TXT 文件	9 KB			
CMakeLists.txt	2020/8/31 23:06	TXT 文件	2 KB			
component.mk	2020/8/31 23:06	Makefile	9 KB			
Kconfig.projbuild	2020/8/31 23:06	PROJBUILD 文件	26 KB			
LICENSE.md	2020/8/31 23:06	MD 文件	1 KB			
Makefile.projbuild	2020/8/31 23:06	PROJBUILD 文件	12 KB			
package.json	2020/8/31 23:06	JSON File	0 KB			
platform.txt	2020/8/31 23:06	TXT 文件	4 KB			
programmers.txt	2020/8/31 23:06	TXT 文件				
README.md	2020/8/31 23:06	MD 文件				

data (D:) > arduino > hardware > espressif > esp32 > tools >

名称	修改日期	类型	大小
dist	2020/11/6 15:05	文件夹	
esptool	2020/11/6 15:05	文件夹	
partitions	2020/8/31 23:06	文件夹	
sdk	2020/8/31 23:06	文件夹	
xtensa-esp32-elf	2020/11/6 15:05	文件夹	
esptota.exe	2020/8/31 23:06	应用程序	3,936 KB
esptota.py	2020/8/31 23:06	PY 文件	10 KB
esptool.py	2020/8/31 23:06	PY 文件	141 KB
gen_esp32part.exe	2020/8/31 23:06	应用程序	3,262 KB
gen_esp32part.py	2020/8/31 23:06	PY 文件	20 KB
get.exe	2020/8/31 23:06	应用程序	5,090 KB
get.py	2020/8/31 23:06	PY 文件	6 KB
platformio-build.py	2020/8/31 23:06	PY 文件	11 KB







ESP-IDF CONSTRUCTION OF DEVELOPMENT ENVIRONMENT

3.1 ESP-IDF Construction of development environment(ESP32)

W800 CONSTRUCTION OF DEVELOPMENT ENVIRONMENT

4.1 SDK Command Line Compilation Guide

4.1.1 (1)Get Software Tools

Go get SDK.

Windows OS Windows OS by default does not support compilation of GNU - based Makefiles, so you need to build an environment that supports GNU - compiler toolchains [Click Download](#).Cygwin-based compilation environment. Unzip the file, open the executable, and the installation is complete

Linux OS Linux OS compiles directly in the shell, requiring the installation of common development libraries and tools on Linux.

4.1.2 (2)Connect Board

4.1.3 (3)Open Cygwin for WinnerMicro.Switch the path to the SDK root

4.1.4 (4)Make compiles complete firmware

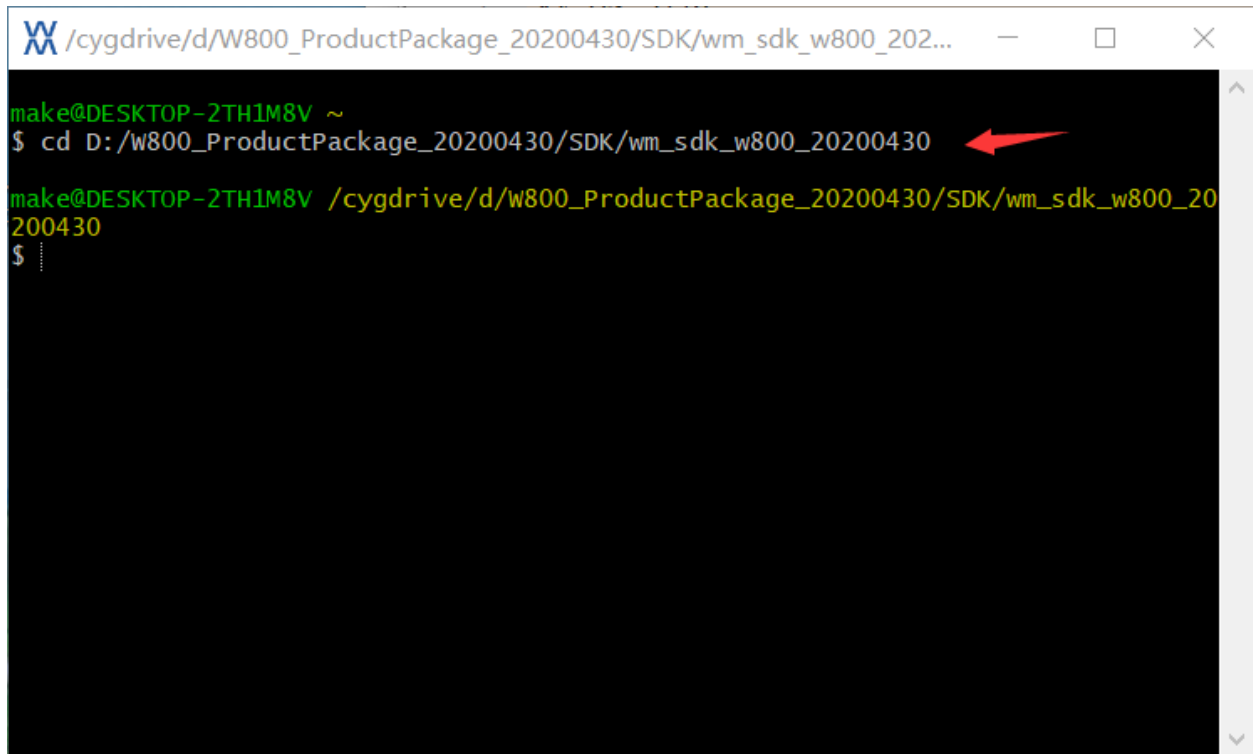
4.1.5 (5)Wait for the firmware to compile successfully

4.1.6 (6)Configuration parameter

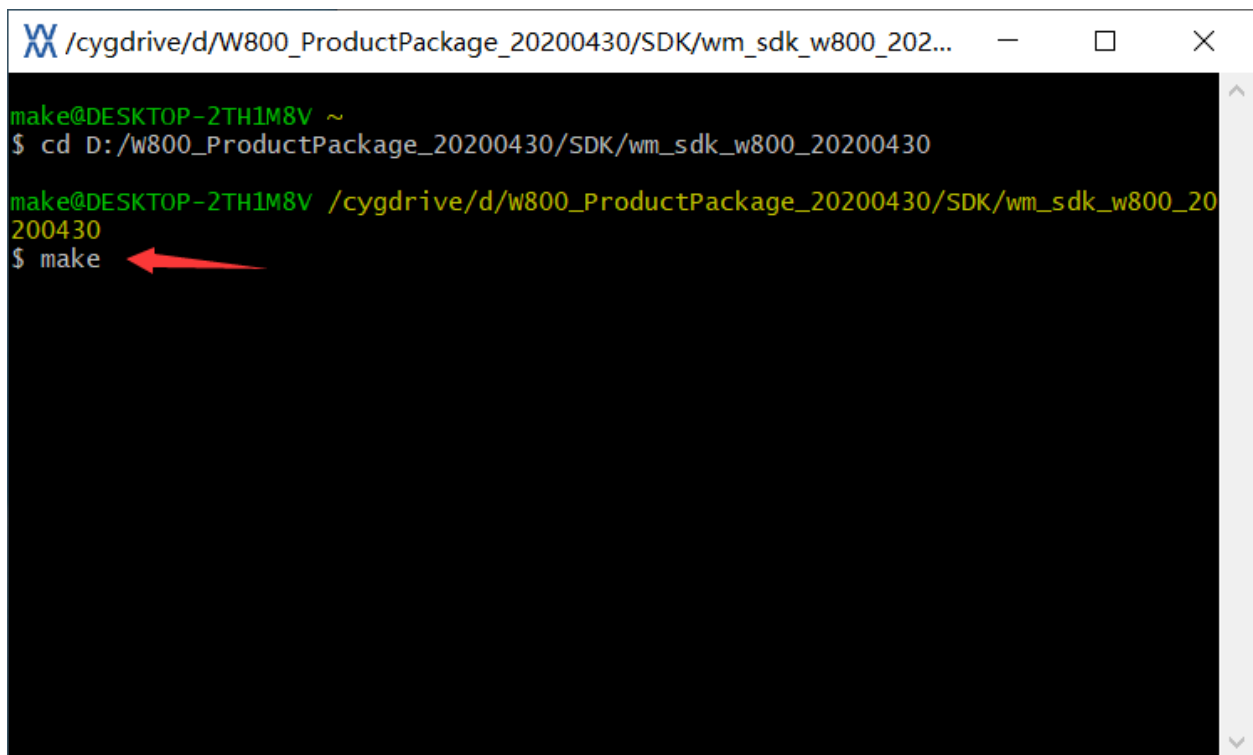
View serial port, command line inputmake list My serial port here is COM21.

Command line inputmake menuconfig

Configure the “Download Configuration” option in the compilation parameter, enter the serial number and the supported Baud rate. I configure serial port number COM21 here, baud rate 2000000, save and exit. The default serial port download baud rate of SDK is 115200. When the user’s download device supports 2M Baud rate, the baud rate is set to 2000000 to experience a faster download speed.



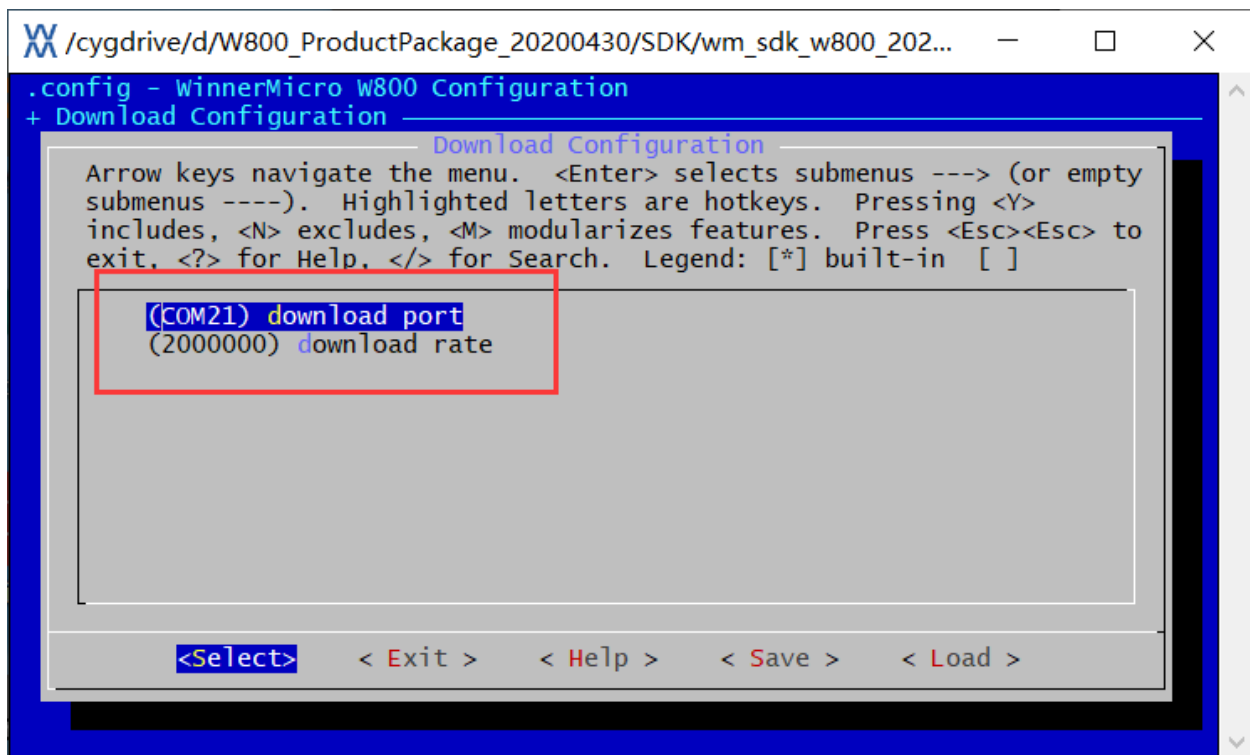
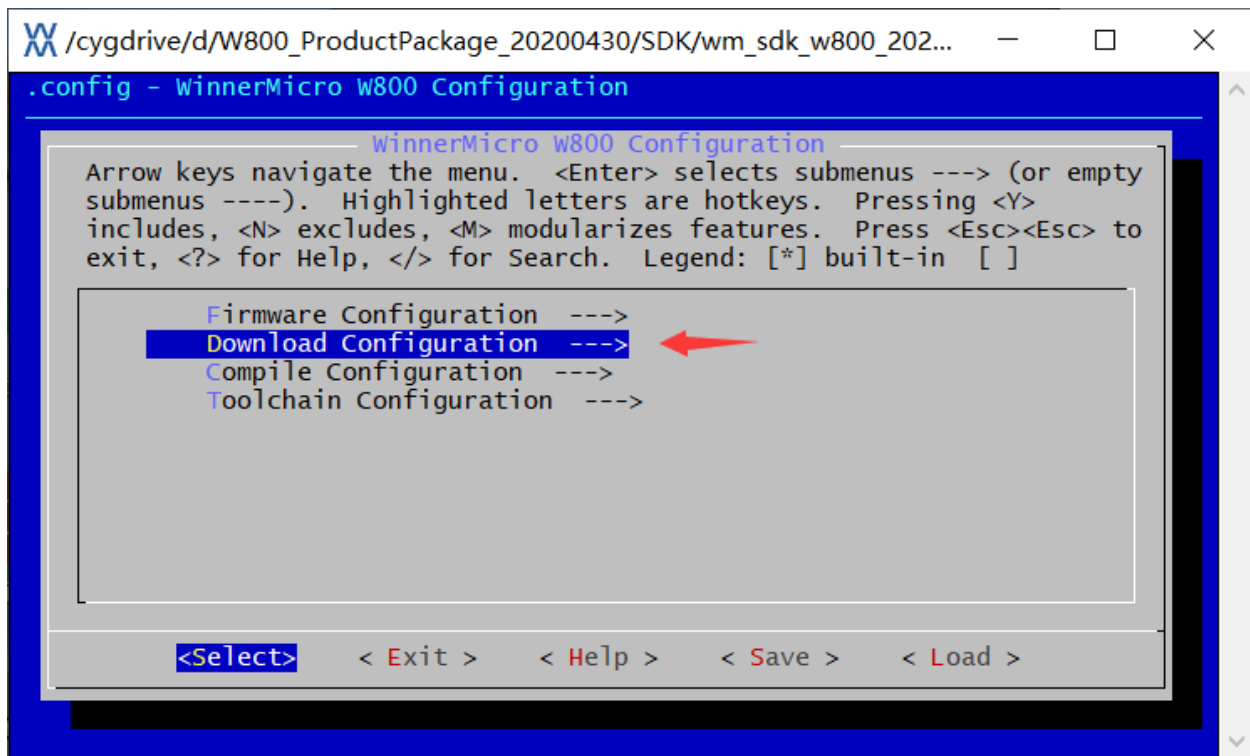
A terminal window with a title bar showing the path `/cygdrive/d/W800_ProductPackage_20200430/SDK/wm_sdk_w800_202...`. The prompt is `make@DESKTOP-2TH1M8V ~`. The command `$ cd D:/W800_ProductPackage_20200430/SDK/wm_sdk_w800_20200430` is entered, with a red arrow pointing to the path. The next line shows the prompt has changed to `make@DESKTOP-2TH1M8V /cygdrive/d/W800_ProductPackage_20200430/SDK/wm_sdk_w800_20200430`. The prompt is followed by a vertical ellipsis `$...`.



A terminal window with a title bar showing the path `/cygdrive/d/W800_ProductPackage_20200430/SDK/wm_sdk_w800_202...`. The prompt is `make@DESKTOP-2TH1M8V ~`. The command `$ cd D:/W800_ProductPackage_20200430/SDK/wm_sdk_w800_20200430` is entered. The next line shows the prompt has changed to `make@DESKTOP-2TH1M8V /cygdrive/d/W800_ProductPackage_20200430/SDK/wm_sdk_w800_20200430`. The command `$ make` is entered, with a red arrow pointing to the word `make`.

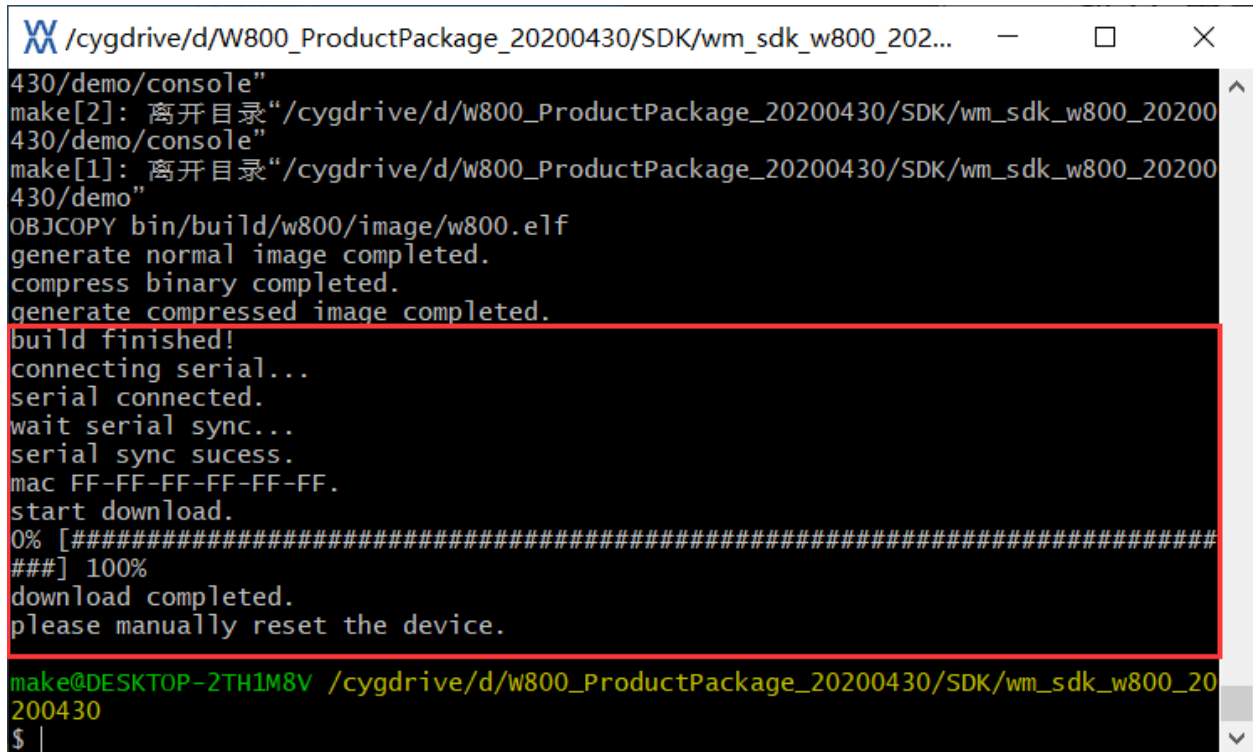
```
/cygdrive/d/W800_ProductPackage_20200430/SDK/wm_sdk_w800_202...  —  □  ×  
430/src/app/web"  
make[2]: 进入目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/src/app/wm_atcmd"  
make[2]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/src/app/wm_atcmd"  
make[1]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/src/app"  
make[1]: 进入目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/demo"  
make[2]: 进入目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/demo/console"  
make[2]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/demo/console"  
make[1]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/demo"  
OBJCOPY bin/build/w800/image/w800.elf  
generate normal image completed.  
compress binary completed.  
generate compressed image completed.  
build finished!  
make@DESKTOP-2TH1M8V /cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20  
200430  
$
```

```
/cygdrive/d/W800_ProductPackage_20200430/SDK/wm_sdk_w800_202...  —  □  ×  
make[2]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/src/app/wm_atcmd"  
make[1]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/src/app"  
make[1]: 进入目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/demo"  
make[2]: 进入目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/demo/console"  
make[2]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/demo/console"  
make[1]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200  
430/demo"  
OBJCOPY bin/build/w800/image/w800.elf  
generate normal image completed.  
compress binary completed.  
generate compressed image completed.  
build finished!  
make@DESKTOP-2TH1M8V /cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20  
200430  
$ make list  
COM21  
make@DESKTOP-2TH1M8V /cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20
```



4.1.7 (7)Upload the firmware

Command line inputmake flash (Once the compilation is complete, burn the second Boot and user code, the firmware in .fls format)



```

/cygdrive/d/W800_ProductPackage_20200430/SDK/wm_sdk_w800_202...
430/demo/console"
make[2]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200
430/demo/console"
make[1]: 离开目录"/cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20200
430/demo"
OBJCOPY bin/build/w800/image/w800.elf
generate normal image completed.
compress binary completed.
generate compressed image completed.
build finished!
connecting serial...
serial connected.
wait serial sync...
serial sync sucess.
mac FF-FF-FF-FF-FF-FF.
start download.
0% [#####]
###] 100%
download completed.
please manually reset the device.

make@DESKTOP-2TH1M8V /cygdrive/d/w800_ProductPackage_20200430/SDK/wm_sdk_w800_20
200430
$ |

```

Other commands.

make image Once the compilation is complete, only the user code, the firmware in .img format, is burned

Note: If this operation is performed and requires secondary boot file support, you will also need to burn secondary boot firmware.

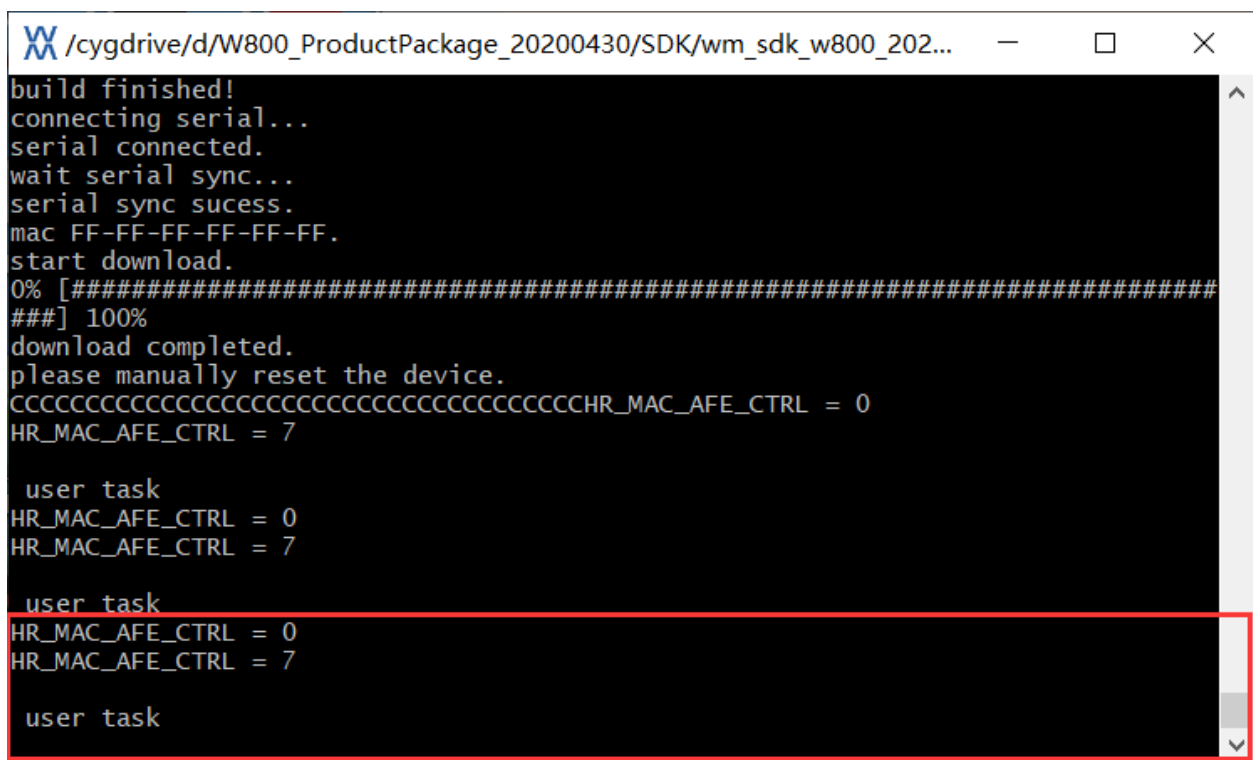
make down Do not compile, directly burn firmware in .fls format.

4.1.8 (8)Run the firmware

It captures the logs generated when the device is working properly and outputs them to the screen.

Command line inputmake run Press reset to view the output

Note: Type Make help from the command line to see what the command does.



```
/cygdrive/d/W800_ProductPackage_20200430/SDK/wm_sdk_w800_202...
build finished!
connecting serial...
serial connected.
wait serial sync...
serial sync sucess.
mac FF-FF-FF-FF-FF-FF.
start download.
0% [#####]
###] 100%
download completed.
please manually reset the device.
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCHR_MAC_AFE_CTRL = 0
HR_MAC_AFE_CTRL = 7

user task
HR_MAC_AFE_CTRL = 0
HR_MAC_AFE_CTRL = 7

user task
HR_MAC_AFE_CTRL = 0
HR_MAC_AFE_CTRL = 7

user task
```


5.1 HK-01

5.1.1

5.2 HK-01B

5.2.1

5.3 HK-01S

5.3.1

5.4 HK-03

5.4.1

5.5 HK-05

5.5.1

5.6 HK-07

5.6.1

5.7 HK-07S

5.7.1

5.8 HK-12E

5.8.1

5.9 HK-12F-L2

5.9.1

BLUETOOTH

6.1 HK-BT01

6.1.1

6.2 HK-BT12

6.2.1

ZIGBEE

7.1 HK-ZB12

7.1.1

7.2 HK-ZB01

7.2.1

LORA

8.1 HK-Ra07

8.1.1

8.2 HK-Ra07-I

8.2.1

9.1 HK-Si01

9.1.1

9.2 HK-Si12

9.2.1

2.4G

10.1 HK-24DP

10.2 HK-24MiNi

MULTI-PROTOCOL

11.1 HK-32D

11.1.1

11.2 HK-32U

11.2.1

11.3 HK-32S

11.3.1

11.4 HK-32S-I

11.4.1

11.5 HK-32B

11.5.1

11.6 HK-WROVER

11.6.1

11.7 HK-WROVER-I

11.7.1

HK-D1-MINIPRO4M

12.1

HK-D1-MINIPRO16M

13.1

HK-ESP32-CAM

14.1 Product Overview

The HK-ESP32-CAM module is a 39.8*27* small camera module designed by Hongke. This module can work independently as the minimum system. New WiFi+ Bluetooth dual mode development board based on ESP32, with PCB onboard antenna, with two high-performance 32-bit LX6Cpus, at level 7 Pipeline architecture, main frequency adjustment range from 80MHz to 240Mhz. Ultra-low power consumption, deep sleep current as low as 6mA. Ultra-small 802.11b/g/ N Wi-Fi + BT/BLE SoC module.

HK-ESP32-CAM can be widely used in a variety of Internet of Things occasions, suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, Wireless positioning system signal and other iot applications are ideal solutions for iot applications.

The HK-ESP32-CAM is DIP encapsulated and can be used by directly plugging into the bottom plate. The reliable connection mode is convenient for all kinds of Internet of Things hardware terminals.

Ultra-small 802.11b/g/ N Wi-Fi + BT/BLE SoC module.

14.2 Product Characteristics

Using low power dual core 32-bit CPU, it can be used as application processor

The main frequency is up to 240mhz, and the computing power is up to 600 DMIPS

Built in 520kb SRAM and external 8MB PSRAM

Support UART/SPI/I2C/PWM/ADC/DAC interfaces

Support ov2640 and ov7670 cameras with built-in flash

Support image WiFi upload

Support TF Card

Support multiple sleep modes

Embedded LwIP and FreeRTOS

Support STA/AP/sta+AP working mode

Support smart config / airkiss one click distribution network

Support secondary development

14.3 Application Scenarios

Picture transmission of home intelligent equipment

Wireless monitoring

Smart agriculture

QR wireless identification

Wireless positioning system signal

And other Internet of things applications

14.4 Technical Specifications

Parameter	description
Working voltage	4.75-5.25v
Spi-flash	The default of 32 mbit
RAM	Internal 520kb + External 8MB PSRAM
Wi-Fi	802.11b/g/n/e/i
Bluetooth	Bluetooth 4.2br/edr and ble
Support interface(2Mbps)	UART, SPI, I2C, PWM
Support TF card	Maximum 4G support
IO	9
Serial port rate	115200bps by default
Spectral Range	2400 ~ 2483.5mhz
Antenna form	PCB antenna on board, gain 2dBI
Image output format	JPEG (ov2640 only), BMP, grayscale
Pack-age	DIP-16
Transmitting power	802.11b: 17 ± 2 dBm (@11Mbps) 802.11g: 14 ± 2 dBm (@54Mbps) 802.11n: 13 ± 2 dBm (@MCS7)
Receiving Sensitivity	CCK,1 Mbps:-90dBm CCK,11 Mbps:-85dBm 6Mbps(1/2BPSK):-88dBm 54Mbps(3/464-QAM):-70dBm MCS7(65Mbps,72.2Mbps):-67dBm Power consumption turn off flash :180mA@5V Turn on the flash and turn on the maximum brightness :310mA@5V Deep sleep: the minimum power consumption can reach 6mA@5V Modem sleep: lowest attainable 20mA@5V Light sleep: up to 6 7mA@5V
Security	WPA/WPA2/WPA2 enterprise/WPS
Working temperature	-20 °C ~ 70 °C

14.5 Pin definition

CAM PIN	ESP32 PIN
D0	PIN5
D1	PIN18
D2	PIN19
D3	PIN21
D4	PIN36
D5	PIN39
D6	PIN34
D7	PIN0
XCLK	PIN22
PCLK	PIN25
VSYNC	PIN23
SDA	PIN26
SCL	PIN27
POWER PIN	PIN32

SD PIN	ESP32 PIN
CLK	PIN14
CMD	PIN15
DATA0	PIN2
DATA1/flash lamp	PIN4
DATA2	PIN12
DATA3	PIN13

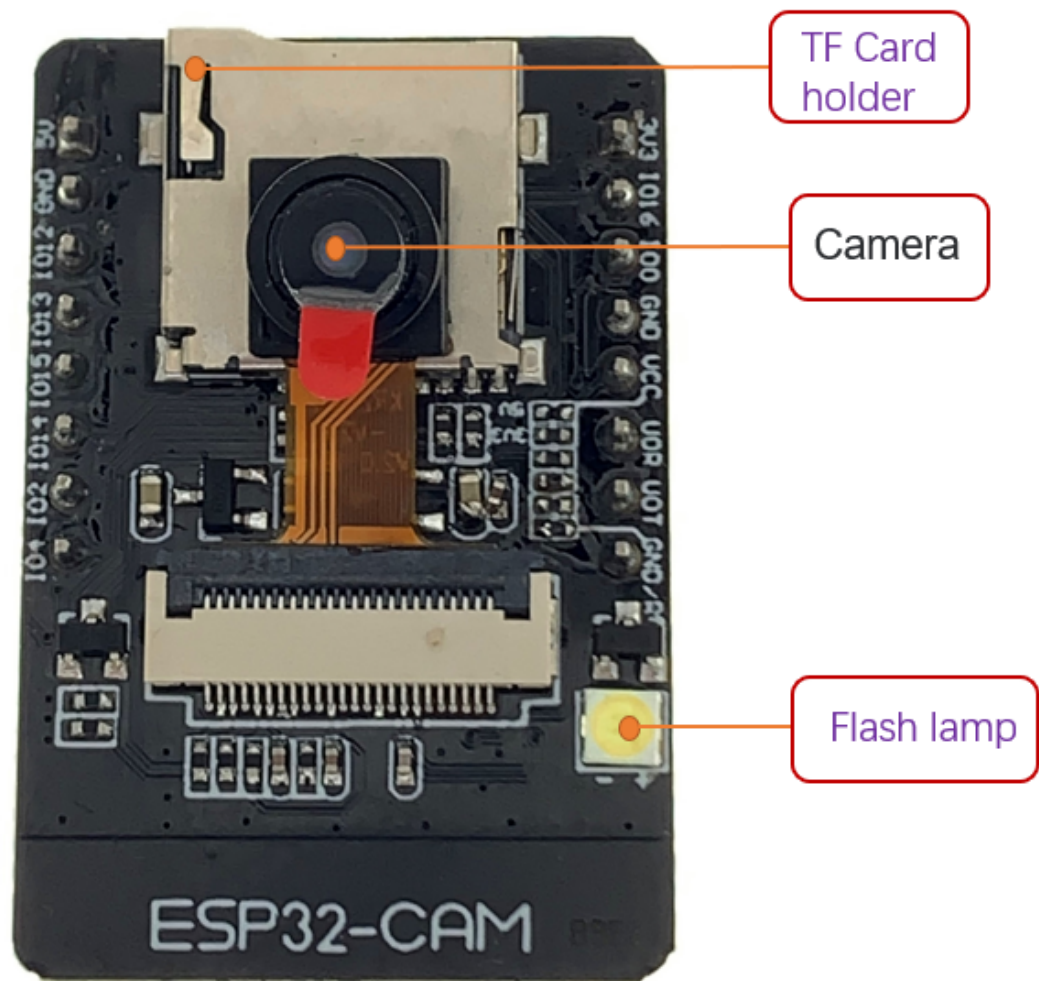


Fig. 1: HK-ESP32-CAM

HK-ESP32-CAM-MB

HK-ESP32-CAM-MB is a base designed by Hongke for hk-esp32-cam small camera module.

ESP32-CAM-MB is equipped with USB to serial interface, which is convenient for users to burn programs. This base and HK-ESP32-CAM can form a module which can work independently as the minimum system.

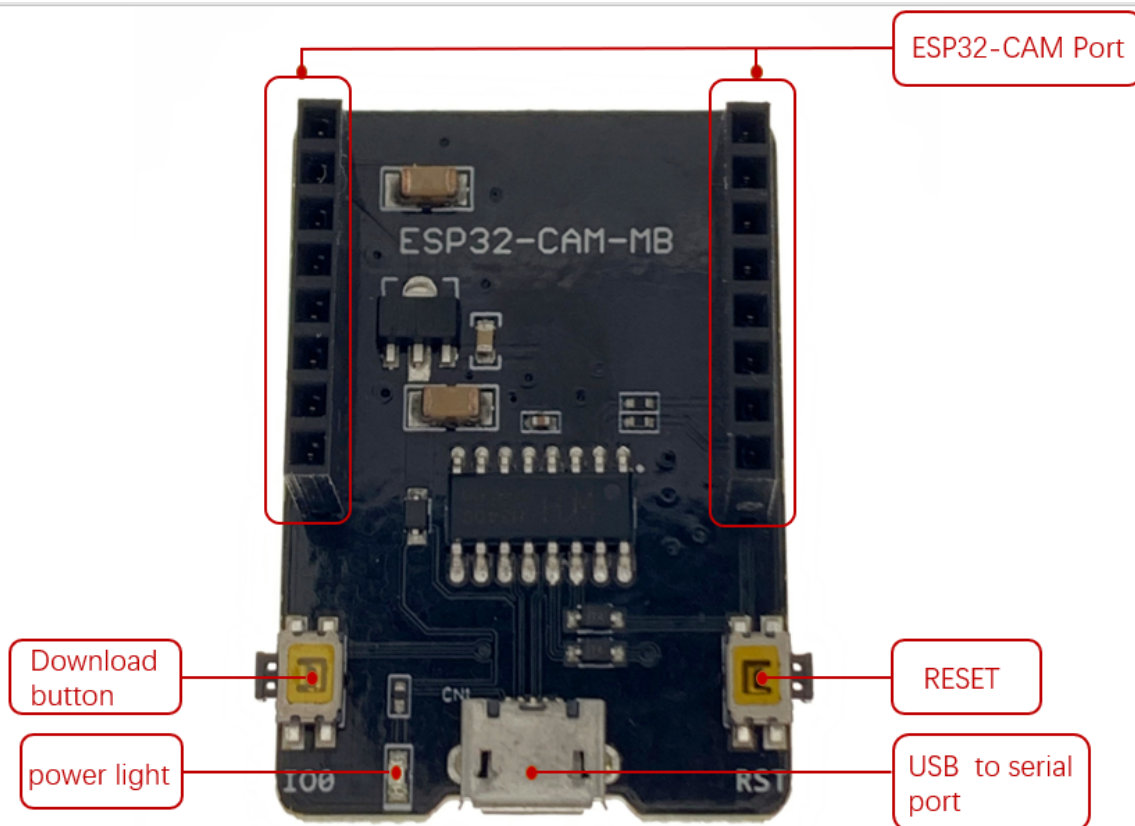


Fig. 1: HK-ESP32-CAM-MB

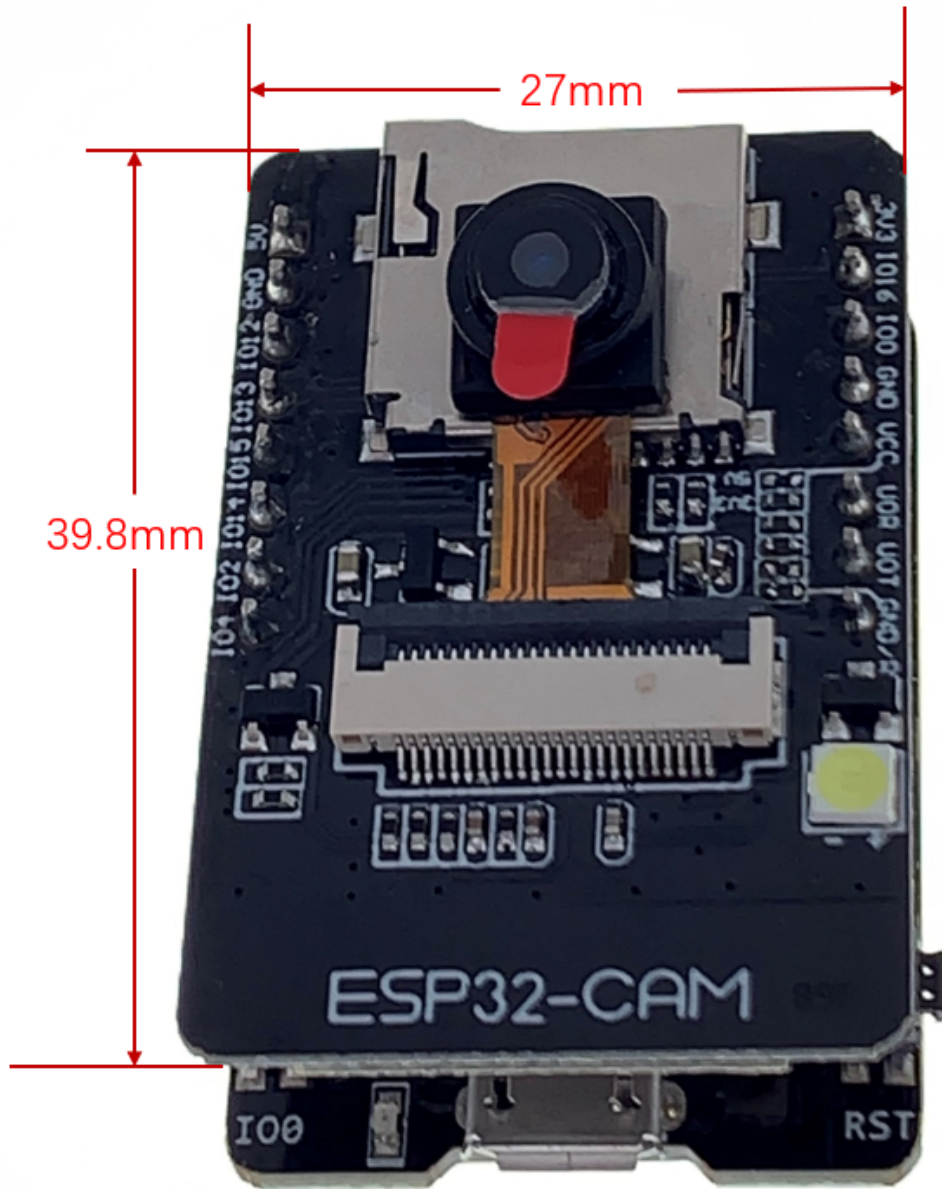


Fig. 2: HK-ESP32-CAM module

HK-ESP32-MINIKIT(BLACK)

16.1

HK-ESP32-MINIKIT(BLUE)

17.1

18.1

19.1