ESP32_CAMERA_QR 使用教程

前言： ESP322-CAM 是我们新推出的一款开发板模组，只需要一个 ESP32 模组和摄像头即可组成一个摄像头系统，尺寸仅为 27x40mm，可广泛应用于各种物联网场合，适用于家庭智能设备、工业无线控制、无线监控、QR 无线识别，无线定位系统信号以及其它物联网应用，是物联网应用的理想解决方案。

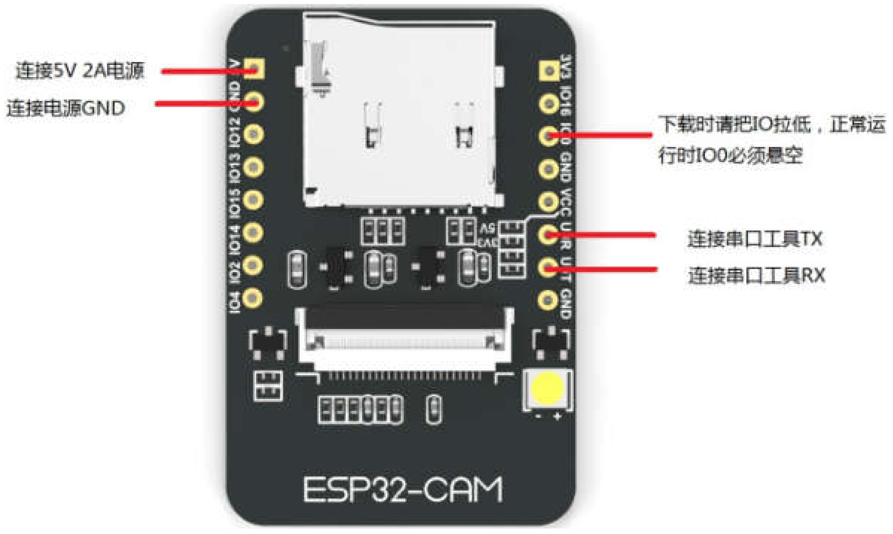
前期准备

1.串口工具

2.杜邦线

3.摄像头转接板

下载准备： 请按照下面图片接线



1.搭建 esp32 开发环境

自己搭建环境：https://esp-idf.readthedocs.io/zh_CN/latest/get-started/index.html

在虚拟机下操作步骤：

为了使开发者更快上手，我们把 ESP32,ESP8266 开发环境集成到 lubuntu 32 位虚拟机，该虚拟机在 VMware12 以上环境下打开，请广大开发者自行下载。

虚拟机推荐配置

1.2 核 CPU

2.至少 1G 内存

虚拟机账号：ai-thinker

密码：aithinker

功能

1.集成 ESP32,ESP8266 开发环境

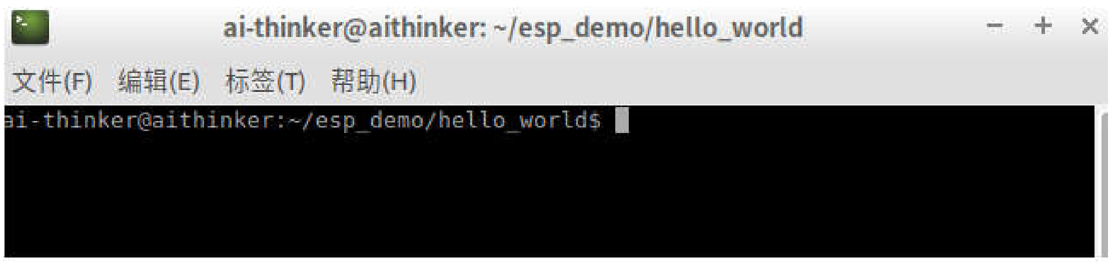
2.支持一键更新

操作步骤

1.下载镜像,地址虚拟机镜像

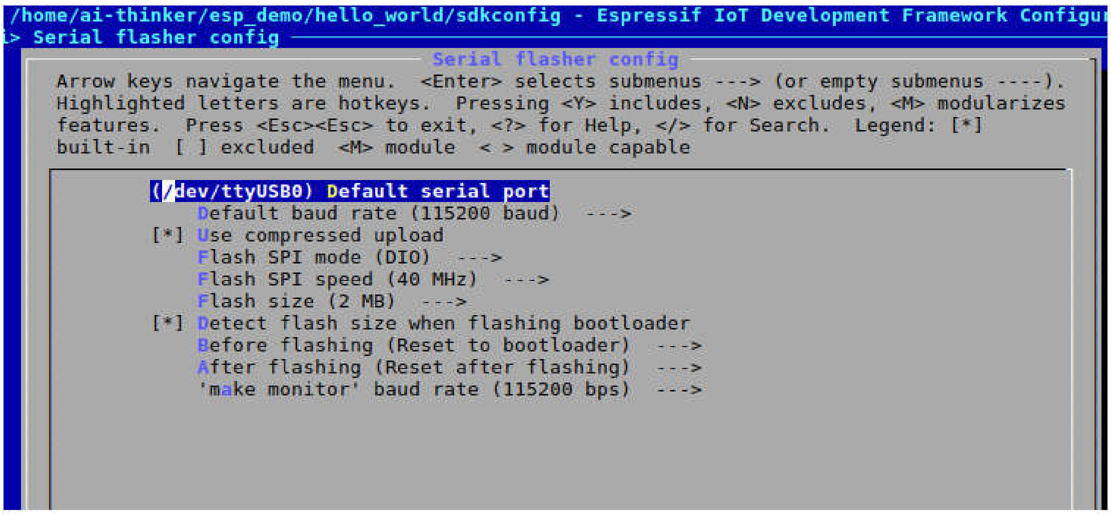
2.安装 VMware12，打开 VMware 软件，导入虚拟机镜像。

3.打开 LX 终端,进入/home/ai-thinker/esp_demo/hello_world 目录


ai-thinker@aithinker: ~/esp_demo/hello_world
文件(F) 编辑(E) 标签(T) 帮助(H)

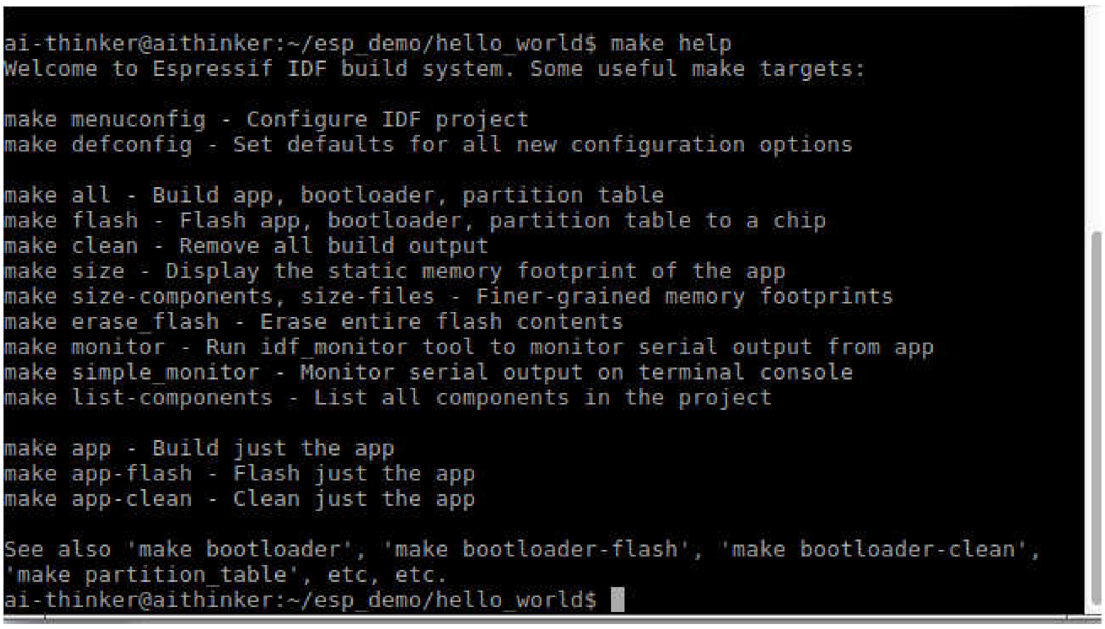
```
ai-thinker@aithinker:~/esp_demo/hello_world$
```

4.输入 make menuconfig,在 Serial flasher config→Default serial port 修改串口端口。



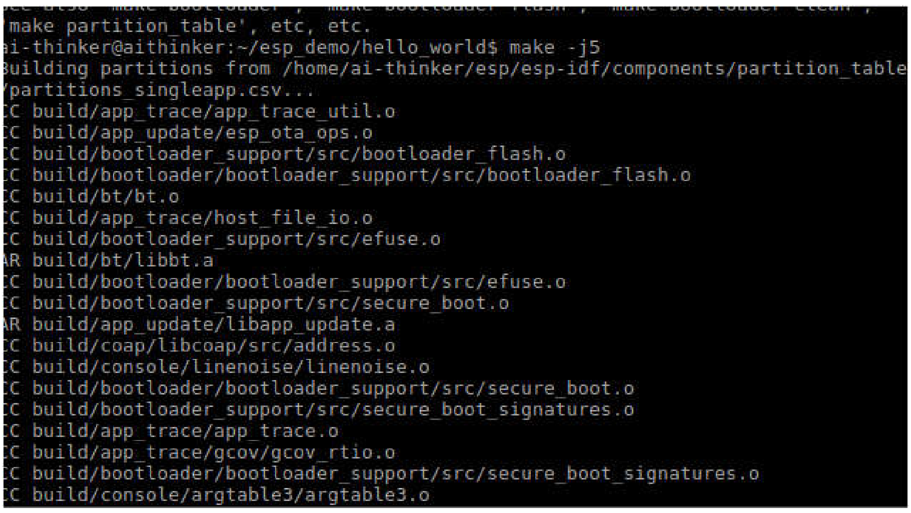
```
/home/ai-thinker/esp_demo/hello_world/sdkconfig - Espressif IoT Development Framework Configu
> Serial flasher config
                              Serial flasher config
    Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
    Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
    features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]
    built-in  [ ] excluded  <M> module  < > module capable

            (/dev/ttyUSB0) Default serial port
                Default baud rate (115200 baud)  --->
            [*] Use compressed upload
                Flash SPI mode (DIO)  --->
                Flash SPI speed (40 MHz)  --->
                Flash size (2 MB)  --->
            [*] Detect flash size when flashing bootloader
                Before flashing (Reset to bootloader)  --->
                After flashing (Reset after flashing)  --->
                'make monitor' baud rate (115200 bps)  --->
```

5.输入 make help 即可查看编译指令

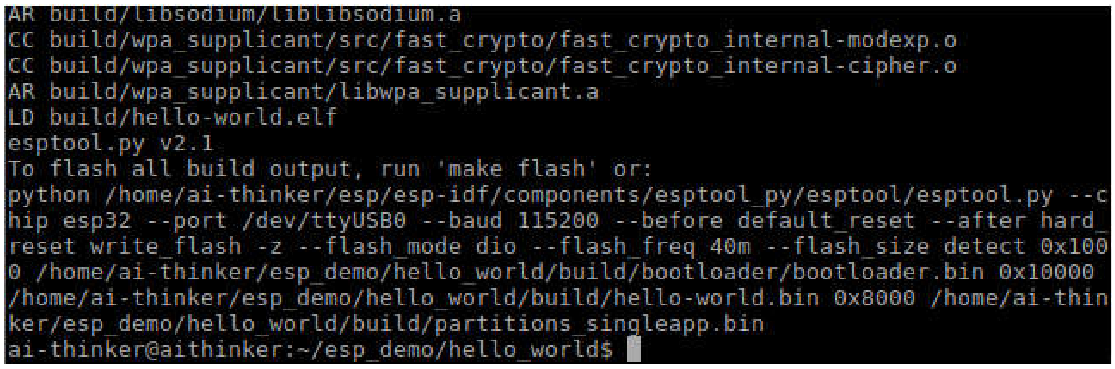


```
ai-thinker@aithinker:~/esp_demo/hello_world$ make help
Welcome to Espressif IDF build system. Some useful make targets:

make menuconfig - Configure IDF project
make defconfig - Set defaults for all new configuration options

make all - Build app, bootloader, partition table
make flash - Flash app, bootloader, partition table to a chip
make clean - Remove all build output
make size - Display the static memory footprint of the app
make size-components, size-files - Finer-grained memory footprints
make erase_flash - Erase entire flash contents
make monitor - Run idf_monitor tool to monitor serial output from app
make simple_monitor - Monitor serial output on terminal console
make list-components - List all components in the project

make app - Build just the app
make app-flash - Flash just the app
make app-clean - Clean just the app

See also 'make bootloader', 'make bootloader-flash', 'make bootloader-clean',
'make partition_table', etc, etc.
ai-thinker@aithinker:~/esp_demo/hello_world$ 
```

6.输入 make -j5 即可编译源码。

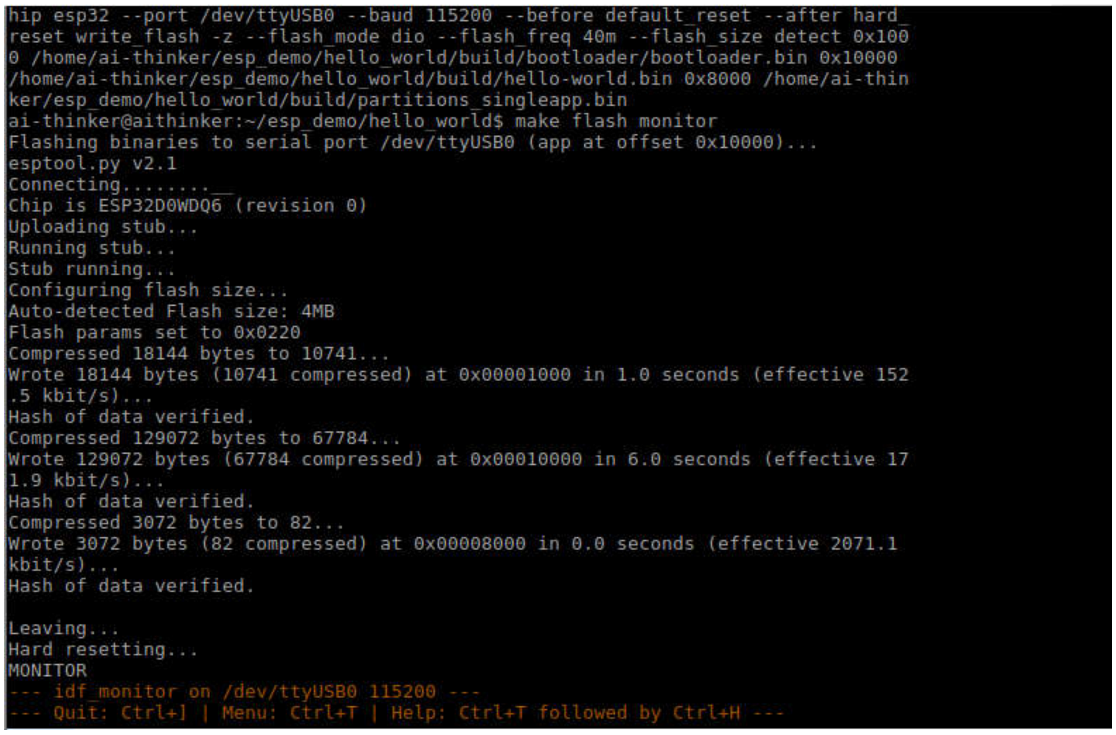


```
'make partition_table', etc, etc.
i-thinker@aithinker:~/esp_demo/hello_world$ make -j5
Building partitions from /home/ai-thinker/esp/esp-idf/components/partition_table
/partitions_singleapp.csv...
C build/app_trace/app_trace_util.o
C build/app_update/esp_ota_ops.o
C build/bootloader_support/src/bootloader_flash.o
C build/bootloader/bootloader_support/src/bootloader_flash.o
C build/bt/bt.o
C build/app_trace/host_file_io.o
C build/bootloader_support/src/efuse.o
R build/bt/libbt.a
C build/bootloader/bootloader_support/src/efuse.o
C build/bootloader_support/src/secure_boot.o
R build/app_update/libapp_update.a
C build/coap/libcoap/src/address.o
C build/console/linenoise/linenoise.o
C build/bootloader/bootloader_support/src/secure_boot.o
C build/bootloader_support/src/secure_boot_signatures.o
C build/app_trace/app_trace.o
C build/app_trace/gcov/gcov_rtio.o
C build/bootloader/bootloader_support/src/secure_boot_signatures.o
C build/console/argtable3/argtable3.o
```



```
AR build/libsodium/liblibsodium.a
CC build/wpa_supplicant/src/fast_crypto/fast_crypto_internal-modexp.o
CC build/wpa_supplicant/src/fast_crypto/fast_crypto_internal-cipher.o
AR build/wpa_supplicant/libwpa_supplicant.a
LD build/hello-world.elf
esptool.py v2.1
To flash all build output, run 'make flash' or:
python /home/ai-thinker/esp/esp-idf/components/esptool_py/esptool/esptool.py --c
hip esp32 --port /dev/ttyUSB0 --baud 115200 --before default_reset --after hard_
reset write_flash -z --flash_mode dio --flash_freq 40m --flash_size detect 0x100
0 /home/ai-thinker/esp_demo/hello_world/build/bootloader/bootloader.bin 0x10000
/home/ai-thinker/esp_demo/hello_world/build/hello-world.bin 0x8000 /home/ai-thin
ker/esp_demo/hello_world/build/partitions_singleapp.bin
ai-thinker@aithinker:~/esp_demo/hello_world$
```

7.插上 ESP32 模块，检测串口工具时候连上电脑，然后输入命令 make flash monitor,即可下载二进制文件到 ESP32 模块上，并且下载成功后可以查看串口调试信息。



```
hip esp32 --port /dev/ttyUSB0 --baud 115200 --before default_reset --after hard_
reset write_flash -z --flash_mode dio --flash_freq 40m --flash_size detect 0x100
0 /home/ai-thinker/esp_demo/hello_world/build/bootloader/bootloader.bin 0x10000
/home/ai-thinker/esp_demo/hello_world/build/hello-world.bin 0x8000 /home/ai-thin
ker/esp_demo/hello_world/build/partitions_singleapp.bin
ai-thinker@aithinker:~/esp_demo/hello_world$ make flash monitor
Flashing binaries to serial port /dev/ttyUSB0 (app at offset 0x10000)...
esptool.py v2.1
Connecting........_
Chip is ESP32D0WDQ6 (revision 0)
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0220
Compressed 18144 bytes to 10741...
Wrote 18144 bytes (10741 compressed) at 0x00001000 in 1.0 seconds (effective 152
.5 kbit/s)...
Hash of data verified.
Compressed 129072 bytes to 67784...
Wrote 129072 bytes (67784 compressed) at 0x00010000 in 6.0 seconds (effective 17
1.9 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 82...
Wrote 3072 bytes (82 compressed) at 0x00008000 in 0.0 seconds (effective 2071.1
kbit/s)...
Hash of data verified.

Leaving...
Hard resetting...
MONITOR
--- idf_monitor on /dev/ttyUSB0 115200 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
```

注意： 请勿随便随便修改/home/ai-thinker/esp 目录下文件，否则可能编译出错。

本 demo 使用 esp-idf 版本是 3.01-rc(c2b39f4a5f4234d3276bec40d42132589739d655)

下载链接 https://github.com/espressif/esp-idf/releases/tag/v3.0.1-rc

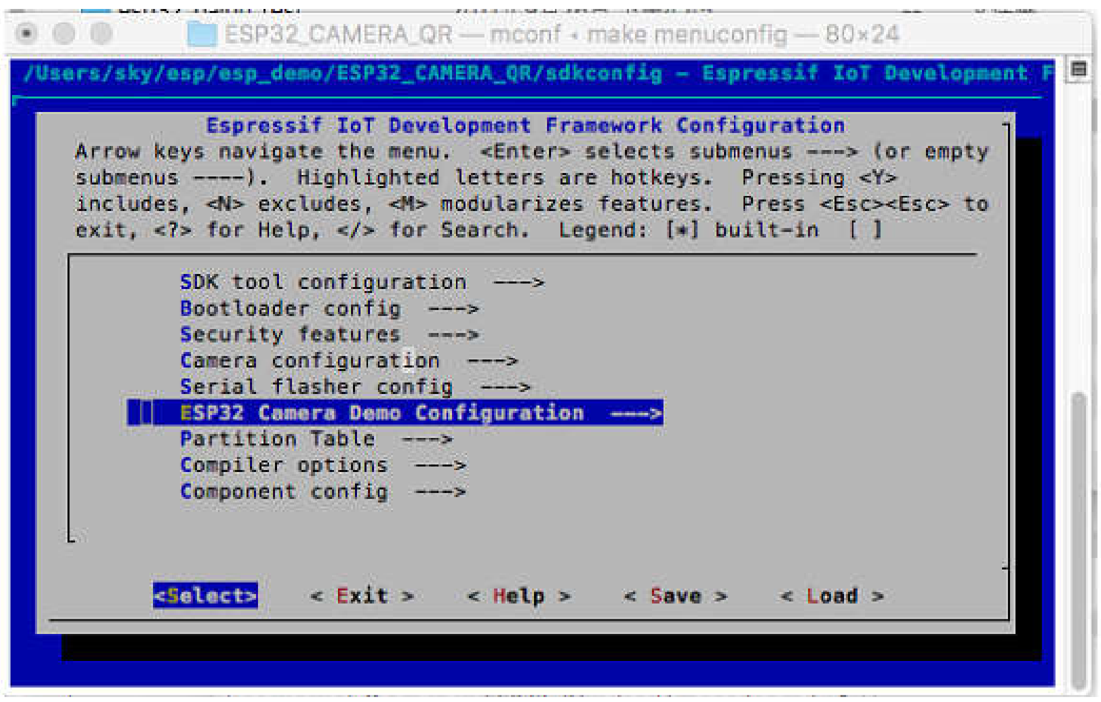
2.下载工程

下载代码：git clone https://github.com/donny681/ESP32_CAMERA_QR.git

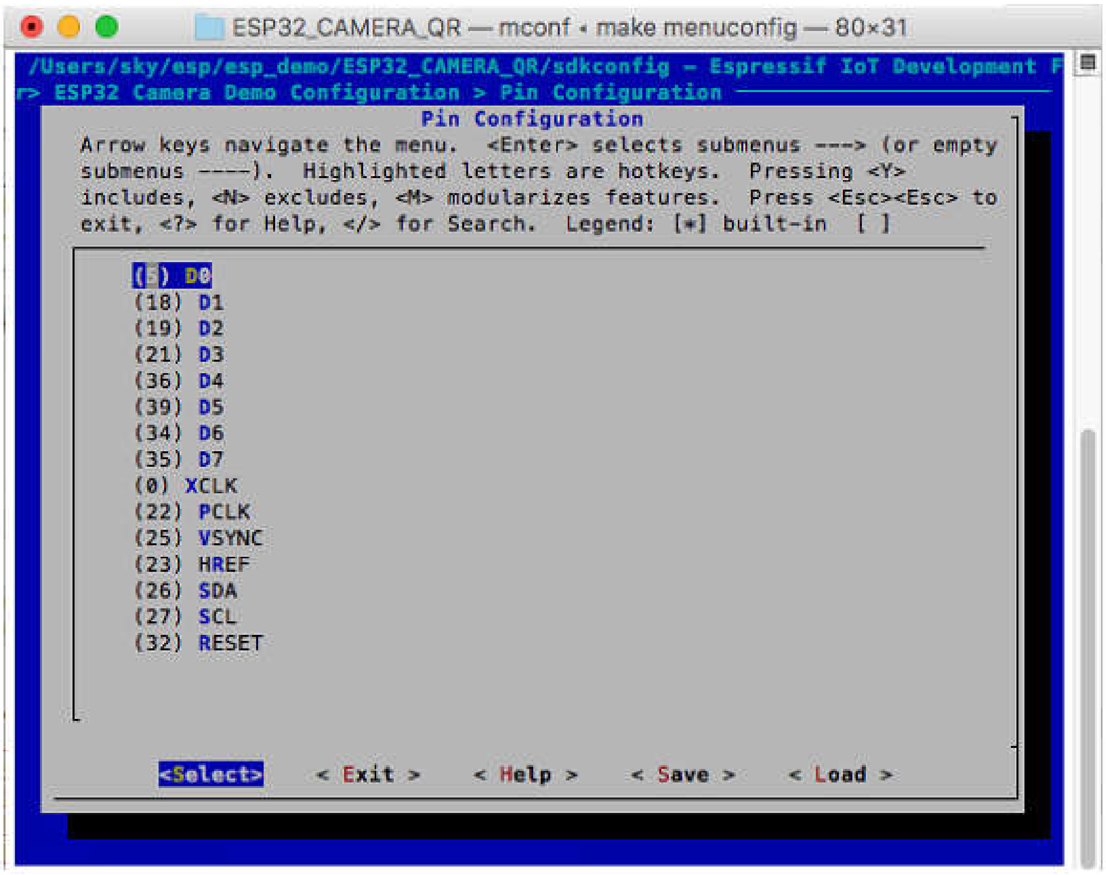
下载子模块：git submodule update --init

3.修改工程参数

在终端输入"make menuconfig"，配置 WiFi 和摄像头参数



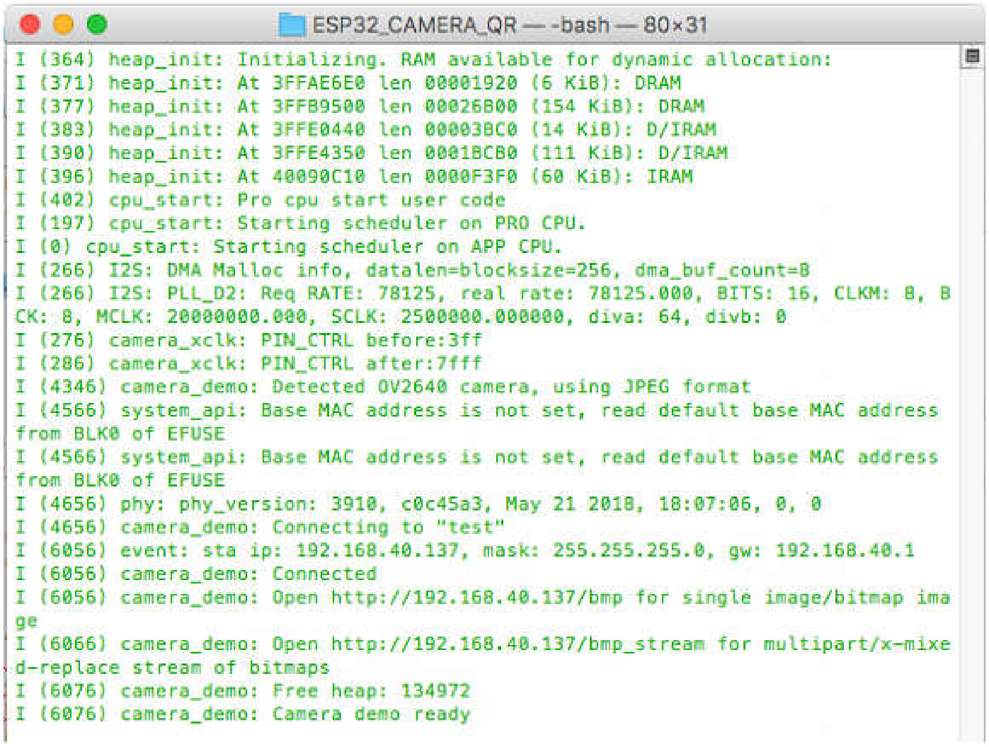
```
ESP32_CAMERA_QR — mconf ◂ make menuconfig — 80×24
/Users/sky/esp/esp_demo/ESP32_CAMERA_QR/sdkconfig - Espressif IoT Development F
            Espressif IoT Development Framework Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]

          SDK tool configuration  --->
          Bootloader config  --->
          Security features  --->
          Camera configuration  --->
          Serial flasher config  --->
          ESP32 Camera Demo Configuration  --->
          Partition Table  --->
          Compiler options  --->
          Component config  --->

      <Select>    < Exit >    < Help >    < Save >    < Load >
```



```
ESP32_CAMERA_QR — mconf ◂ make menuconfig — 80×31
/Users/sky/esp/esp_demo/ESP32_CAMERA_QR/sdkconfig - Espressif IoT Development F
r> ESP32 Camera Demo Configuration > Pin Configuration
                            Pin Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]

      (5) D0
      (18) D1
      (19) D2
      (21) D3
      (36) D4
      (39) D5
      (34) D6
      (35) D7
      (0) XCLK
      (22) PCLK
      (25) VSYNC
      (23) HREF
      (26) SDA
      (27) SCL
      (32) RESET

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

4.打开工程文件 app_main.c，修改 CAMERA_FRAME_SIZE，CAMERA_FRAME_SIZE（照片大小）宏定义（默认配置 JPEG 格式）

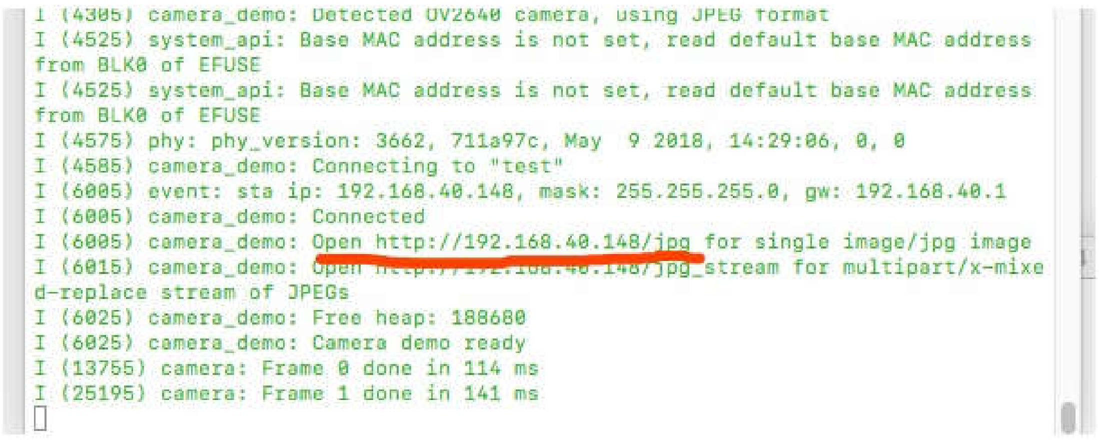
5.在终端输入"make flash monitor"，编译工程，并且烧录。



```
I (364) heap_init: Initializing. RAM available for dynamic allocation:
I (371) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (377) heap_init: At 3FFB9500 len 00026B00 (154 KiB): DRAM
I (383) heap_init: At 3FFE0440 len 00003BC0 (14 KiB): D/IRAM
I (390) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (396) heap_init: At 40090C10 len 0000F3F0 (60 KiB): IRAM
I (402) cpu_start: Pro cpu start user code
I (197) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
I (266) I2S: DMA Malloc info, datalen=blocksize=256, dma_buf_count=8
I (266) I2S: PLL_D2: Req RATE: 78125, real rate: 78125.000, BITS: 16, CLKM: 8, B
CK: 8, MCLK: 20000000.000, SCLK: 2500000.000000, diva: 64, divb: 0
I (276) camera_xclk: PIN_CTRL before:3ff
I (286) camera_xclk: PIN_CTRL after:7fff
I (4346) camera_demo: Detected OV2640 camera, using JPEG format
I (4566) system_api: Base MAC address is not set, read default base MAC address
from BLK0 of EFUSE
I (4566) system_api: Base MAC address is not set, read default base MAC address
from BLK0 of EFUSE
I (4656) phy: phy_version: 3910, c0c45a3, May 21 2018, 18:07:06, 0, 0
I (4656) camera_demo: Connecting to "test"
I (6056) event: sta ip: 192.168.40.137, mask: 255.255.255.0, gw: 192.168.40.1
I (6056) camera_demo: Connected
I (6056) camera_demo: Open http://192.168.40.137/bmp for single image/bitmap ima
ge
I (6066) camera_demo: Open http://192.168.40.137/bmp_stream for multipart/x-mixe
d-replace stream of bitmaps
I (6076) camera_demo: Free heap: 134972
I (6076) camera_demo: Camera demo ready
```

6.查看串口信息，或者模组 ip 信息，然后输入 http://模组 ip 地址+"/jpg"即可获取图像，请确保电脑，模组在同一个局域网下。

例如本例子模组获取地址是 192.168.40.148，请看以下截图



```
I (4305) camera_demo: Detected OV2640 camera, using JPEG format
I (4525) system_api: Base MAC address is not set, read default base MAC address
from BLK0 of EFUSE
I (4525) system_api: Base MAC address is not set, read default base MAC address
from BLK0 of EFUSE
I (4575) phy: phy_version: 3662, 711a97c, May  9 2018, 14:29:06, 0, 0
I (4585) camera_demo: Connecting to "test"
I (6005) event: sta ip: 192.168.40.148, mask: 255.255.255.0, gw: 192.168.40.1
I (6005) camera_demo: Connected
I (6005) camera_demo: Open http://192.168.40.148/jpg for single image/jpg image
I (6015) camera_demo: Open http://192.168.40.148/jpg_stream for multipart/x-mixe
d-replace stream of JPEGs
I (6025) camera_demo: Free heap: 188680
I (6025) camera_demo: Camera demo ready
I (13755) camera: Frame 0 done in 114 ms
I (25195) camera: Frame 1 done in 141 ms
```

在浏览器中输入"http://192.168.40.148/jpg" ，即可收到图像信息