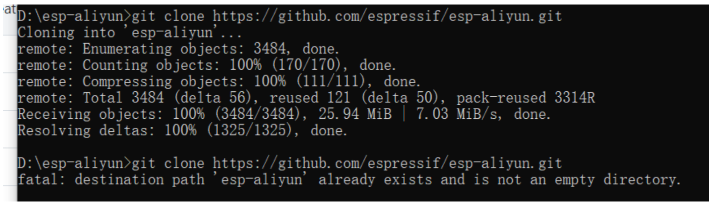
阿里云连接测试

1. 获取阿里云 sdk

下载阿里云 sdk

git clone https://github.com/espressif/esp-aliyun.git

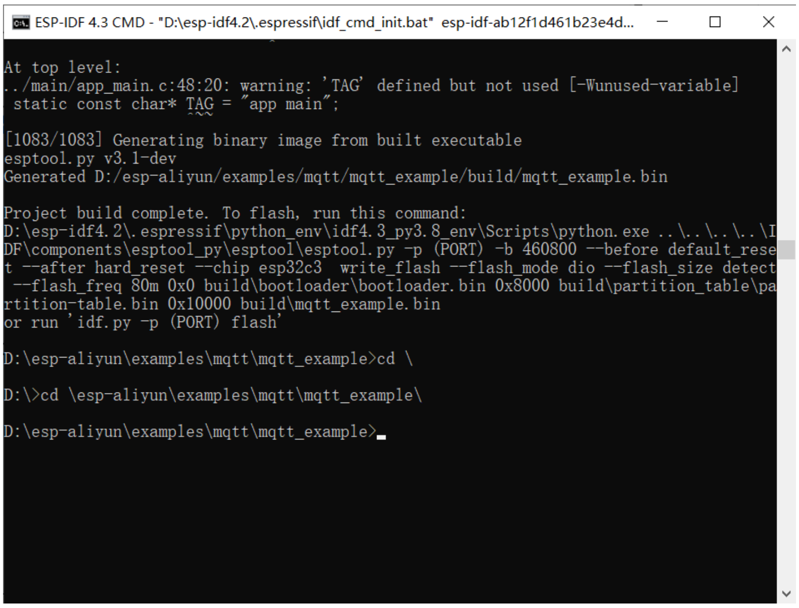


```
D:\esp-aliyun>git clone https://github.com/espressif/esp-aliyun.git
Cloning into 'esp-aliyun'...
remote: Enumerating objects: 3484, done.
remote: Counting objects: 100% (170/170), done.
remote: Compressing objects: 100% (111/111), done.
remote: Total 3484 (delta 56), reused 121 (delta 50), pack-reused 3314R
Receiving objects: 100% (3484/3484), 25.94 MiB | 7.03 MiB/s, done.
Resolving deltas: 100% (1325/1325), done.

D:\esp-aliyun>git clone https://github.com/espressif/esp-aliyun.git
fatal: destination path 'esp-aliyun' already exists and is not an empty directory.
```

以 esp-aliyun\examples\mqtt\mqtt_example 为例

以下命令必须使用自己的路径

cd esp-aliyun\examples\mqtt\mqtt_example

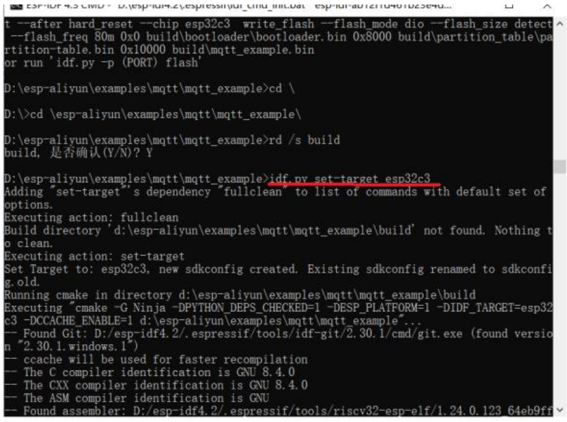


ESP-IDF 4.3 CMD - "D:\esp-idf4.2\.espressif\idf_cmd_init.bat" esp-idf-ab12f1d461b23e4d...

```
At top level:
../main/app_main.c:48:20: warning: 'TAG' defined but not used [-Wunused-variable]
 static const char* TAG = "app main";
                    ^~~

[1083/1083] Generating binary image from built executable
esptool.py v3.1-dev
Generated D:/esp-aliyun/examples/mqtt/mqtt_example/build/mqtt_example.bin

Project build complete. To flash, run this command:
D:\esp-idf4.2\.espressif\python_env\idf4.3_py3.8_env\Scripts\python.exe ..\..\..\..\I
DF\components\esptool_py\esptool\esptool.py -p (PORT) -b 460800 --before default_rese
t --after hard_reset --chip esp32c3  write_flash --flash_mode dio --flash_size detect
 --flash_freq 80m 0x0 build\bootloader\bootloader.bin 0x8000 build\partition_table\pa
rtition-table.bin 0x10000 build\mqtt_example.bin
or run 'idf.py -p (PORT) flash'

D:\esp-aliyun\examples\mqtt\mqtt_example>cd \

D:\>cd \esp-aliyun\examples\mqtt\mqtt_example\

D:\esp-aliyun\examples\mqtt\mqtt_example>_
```

2. 设置目标芯片为 esp32c3

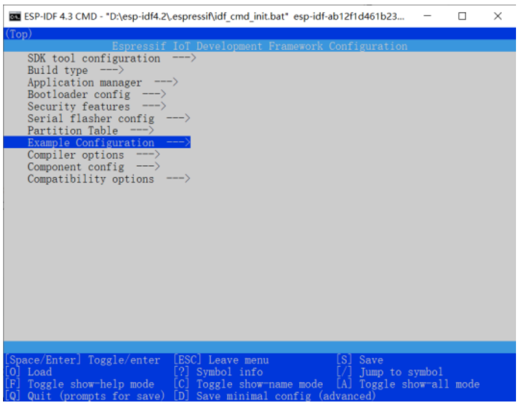
idf.py set-target esp32c3



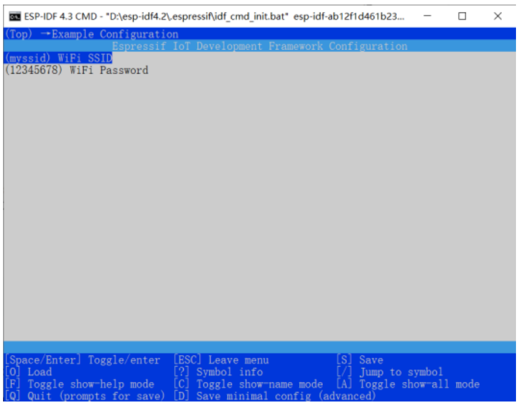
```
t --after hard_reset --chip esp32c3  write_flash --flash_mode dio --flash_size detect
 --flash_freq 80m 0x0 build\bootloader\bootloader.bin 0x8000 build\partition_table\pa
rtition-table.bin 0x10000 build\mqtt_example.bin
or run 'idf.py -p (PORT) flash'

D:\esp-aliyun\examples\mqtt\mqtt_example>cd \

D:\>cd \esp-aliyun\examples\mqtt\mqtt_example\

D:\esp-aliyun\examples\mqtt\mqtt_example>rd /s build
build, 是否确认(Y/N)? Y

D:\esp-aliyun\examples\mqtt\mqtt_example>idf.py set-target esp32c3
Adding "set-target"'s dependency "fullclean" to list of commands with default set of
options.
Executing action: fullclean
Build directory 'd:\esp-aliyun\examples\mqtt\mqtt_example\build' not found. Nothing t
o clean.
Executing action: set-target
Set Target to: esp32c3, new sdkconfig created. Existing sdkconfig renamed to sdkconfi
g.old.
Running cmake in directory d:\esp-aliyun\examples\mqtt\mqtt_example\build
Executing "cmake -G Ninja -DPYTHON_DEPS_CHECKED=1 -DESP_PLATFORM=1 -DIDF_TARGET=esp32
c3 -DCCACHE_ENABLE=1 d:\esp-aliyun\examples\mqtt\mqtt_example"...
-- Found Git: D:/esp-idf4.2/.espressif/tools/idf-git/2.30.1/cmd/git.exe (found versio
n "2.30.1.windows.1")
-- ccache will be used for faster recompilation
-- The C compiler identification is GNU 8.4.0
-- The CXX compiler identification is GNU 8.4.0
-- The ASM compiler identification is GNU
-- Found assembler: D:/esp-idf4.2/.espressif/tools/riscv32-esp-elf/1.24.0.123_64eb9ff
```

idf.py menuconfig　　配置烧写串口和 WIFI_SSID, WIFI_PASSWORD

3． 芯片配置

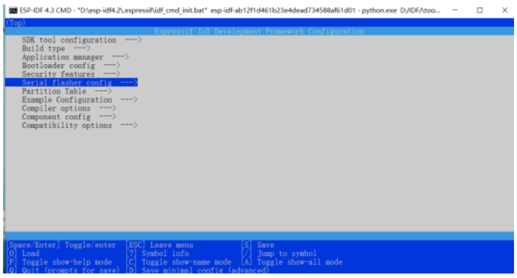


```
ESP-IDF 4.3 CMD - "D:\esp-idf4.2\.espressif\idf_cmd_init.bat"  esp-idf-ab12f1d461b23...
(Top)
                  Espressif IoT Development Framework Configuration
    SDK tool configuration  --->
    Build type  --->
    Application manager  --->
    Bootloader config  --->
    Security features  --->
    Serial flasher config  --->
    Partition Table  --->
    Example Configuration  --->
    Compiler options  --->
    Component config  --->
    Compatibility options  --->

[Space/Enter] Toggle/enter   [ESC] Leave menu           [S] Save
[O] Load                     [?] Symbol info            [/] Jump to symbol
[F] Toggle show-help mode    [C] Toggle show-name mode  [A] Toggle show-all mode
[Q] Quit (prompts for save)  [D] Save minimal config (advanced)
```

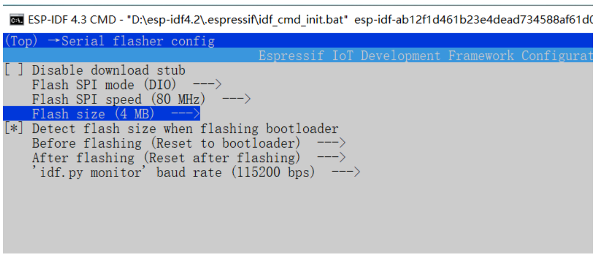


```
ESP-IDF 4.3 CMD - "D:\esp-idf4.2\.espressif\idf_cmd_init.bat"  esp-idf-ab12f1d461b23...
(Top) →Example Configuration
                  Espressif IoT Development Framework Configuration
(myssid) WiFi SSID
(12345678) WiFi Password

[Space/Enter] Toggle/enter   [ESC] Leave menu           [S] Save
[O] Load                     [?] Symbol info            [/] Jump to symbol
[F] Toggle show-help mode    [C] Toggle show-name mode  [A] Toggle show-all mode
[Q] Quit (prompts for save)  [D] Save minimal config (advanced)
```

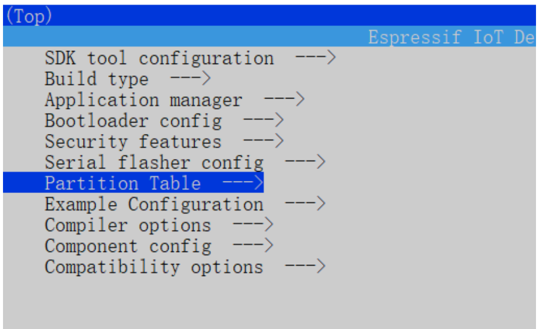
这里的 wifi 名 与密码 所用到的 wifi 需要联网,



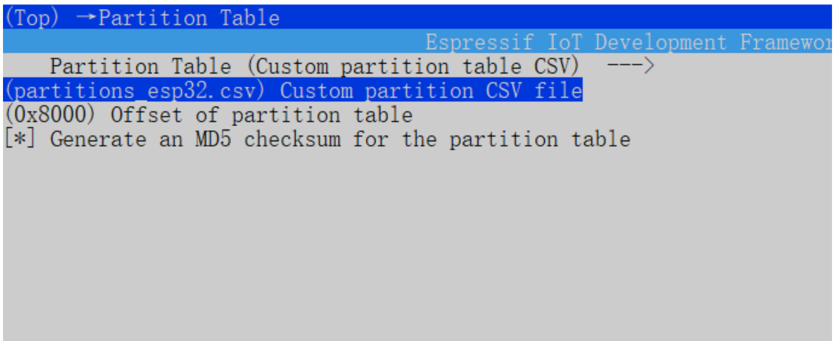
```
ESP-IDF 4.3 CMD - "D:\esp-idf4.2\.espressif\idf_cmd_init.bat"  esp-idf-ab12f1d461b23e4dead734588af61d01 - python.exe  D:/IDF/\too...
(Top)
                  Espressif IoT Development Framework Configuration
    SDK tool configuration  --->
    Build type  --->
    Application manager  --->
    Bootloader config  --->
    Security features  --->
    Serial flasher config  --->
    Partition Table  --->
    Example Configuration  --->
    Compiler options  --->
    Component config  --->
    Compatibility options  --->

[Space/Enter] Toggle/enter   [ESC] Leave menu           [S] Save
[O] Load                     [?] Symbol info            [/] Jump to symbol
[F] Toggle show-help mode    [C] Toggle show-name mode  [A] Toggle show-all mode
[Q] Quit (prompts for save)  [D] Save minimal config (advanced)
```



```
ESP-IDF 4.3 CMD - "D:\esp-idf4.2\.espressif\idf_cmd_init.bat"  esp-idf-ab12f1d461b23e4dead734588af61d0
(Top) →Serial flasher config
                                  Espressif IoT Development Framework Configurat
[ ] Disable download stub
    Flash SPI mode (DIO)  --->
    Flash SPI speed (80 MHz)  --->
    Flash size (4 MB)  --->
[*] Detect flash size when flashing bootloader
    Before flashing (Reset to bootloader)  --->
    After flashing (Reset after flashing)  --->
    'idf.py monitor' baud rate (115200 bps)  --->
```



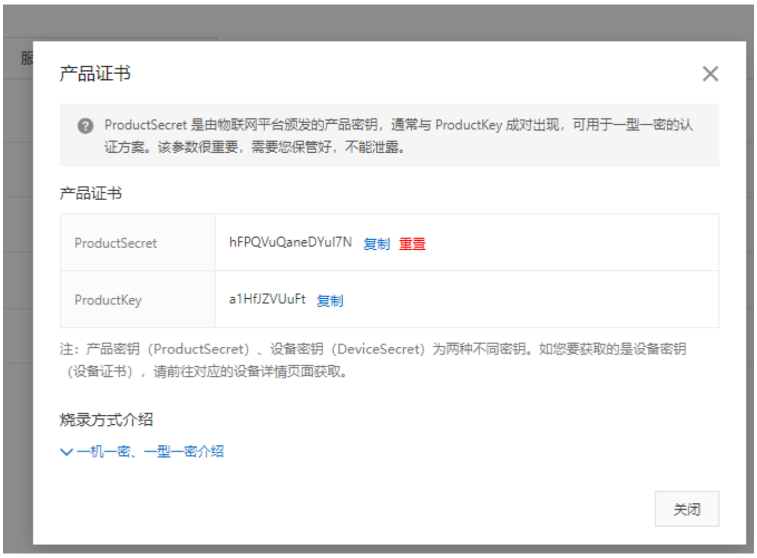
```
(Top)
                                          Espressif IoT De
    SDK tool configuration  --->
    Build type  --->
    Application manager  --->
    Bootloader config  --->
    Security features  --->
    Serial flasher config  --->
    Partition Table  --->
    Example Configuration  --->
    Compiler options  --->
    Component config  --->
    Compatibility options  --->
```



```
(Top) →Partition Table
                                          Espressif IoT Development Framewo
    Partition Table (Custom partition table CSV)  --->
(partitions_esp32.csv) Custom partition CSV file
(0x8000) Offset of partition table
[*] Generate an MD5 checksum for the partition table
```

4. 获取阿里云的三元组加 ProductSecret

接着去阿里云物联网平台创建产品及产品下的设备 获取它的 4 个参数：



产品证书

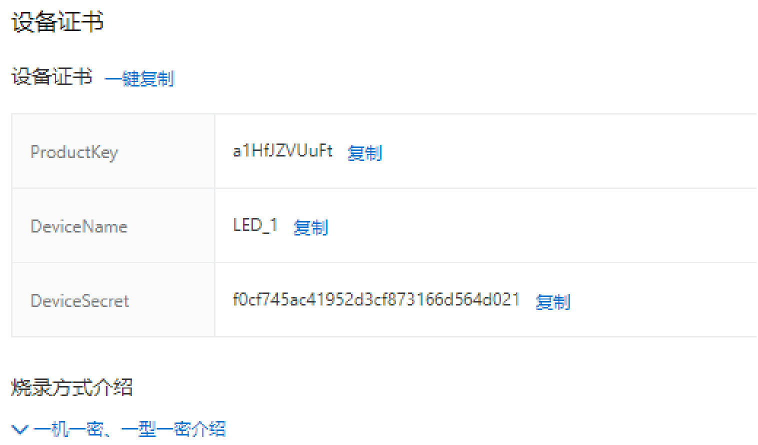
ProductSecret 是由物联网平台颁发的产品密钥，通常与 ProductKey 成对出现，可用于一型一密的认证方案。该参数很重要，需要您保管好，不能泄露。

产品证书

ProductSecret	hFPQVuQaneDYul7N 复制 重置
ProductKey	a1HfJZVUuFt 复制

注：产品密钥（ProductSecret）、设备密钥（DeviceSecret）为两种不同密钥。如您要获取的是设备密钥（设备证书），请前往对应的设备详情页面获取。

烧录方式介绍

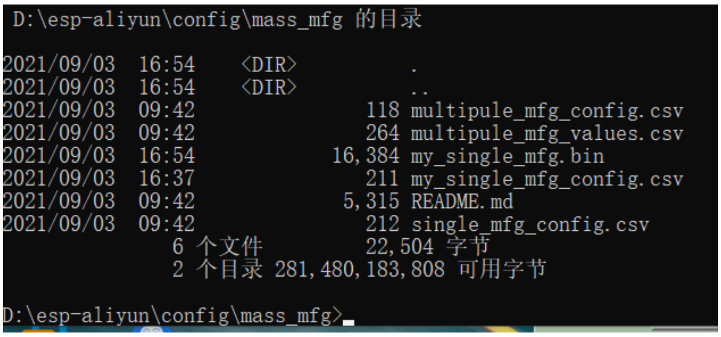


设备证书

设备证书 一键复制

ProductKey	a1HfJZVUuFt 复制
DeviceName	LED_1 复制
DeviceSecret	f0cf745ac41952d3cf873166d564d021 复制

烧录方式介绍



```
 D:\esp-aliyun\config\mass_mfg 的目录

2021/09/03  16:54    <DIR>          .
2021/09/03  16:54    <DIR>          ..
2021/09/03  09:42               118 multipule_mfg_config.csv
2021/09/03  09:42               264 multipule_mfg_values.csv
2021/09/03  16:54            16,384 my_single_mfg.bin
2021/09/03  16:37               211 my_single_mfg_config.csv
2021/09/03  09:42             5,315 README.md
2021/09/03  09:42               212 single_mfg_config.csv
               6 个文件         22,504 字节
               2 个目录 281,480,183,808 可用字节

D:\esp-aliyun\config\mass_mfg>
```

cd 到该目录下 cd esp-aliyun\config\mass_mfg

5.　生成可执行 bin 文件

单 bin 生成

在调试过程中，建议使用该方式。

mass_mfg 目录中有一个参考配置：single_mfg_config.csv，请复制自己的配置文件，如 my_single_mfg_config.csv。

cp single_mfg_config.csv my_single_mfg_config.csv

使用自己的 ProductKey、ProductSecret、DeviceName、DeviceSecret 对 my_single_mfg_config.csv 进行修改:

```
key,type,encoding,value
aliyun-key,namespace,,
DeviceName,data,string,config
DeviceSecret,data,string,dsj3RuY74pgCBJ3zczKz1LaLK7RGApqh
ProductKey,data,string,a10BnLLzGv4
ProductSecret,data,string,pVfLpS1u3A9JM0go
```

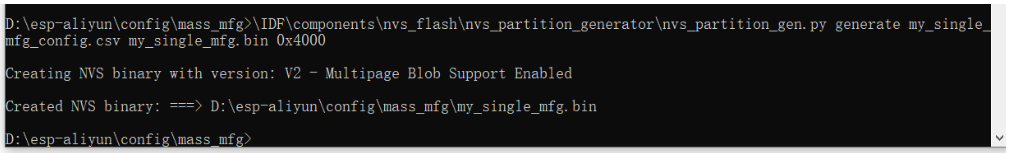
将 config，dsj3RuY74pgCBJ3zczKz1LaLK7RGApqh，a10BnLLzGv4，pVfLpS1u3A9JM0go 为你修改的值。

6. 烧入到目标开发板

%userprofile%\components\nvs_flash\nvs_partition_generator\nvs_partition_gen.py generate my_single_mfg_config.csv my_single_mfg.bin 0x4000

%userprofile% 系统变量可以替换成自己的路径

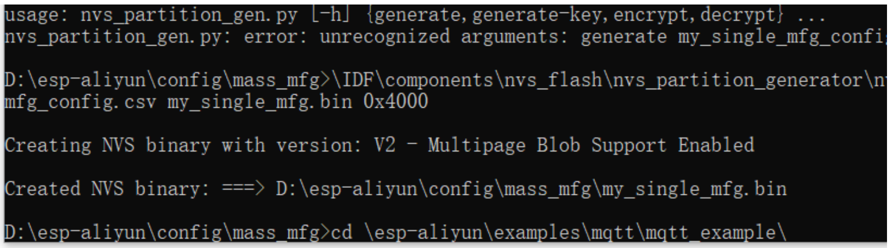
在目录下生成可执行 my_single_mfg.bin



```
D:\esp-aliyun\config\mass_mfg>\IDF\components\nvs_flash\nvs_partition_generator\nvs_partition_gen.py generate my_single_
mfg_config.csv my_single_mfg.bin 0x4000

Creating NVS binary with version: V2 - Multipage Blob Support Enabled

Created NVS binary: ===> D:\esp-aliyun\config\mass_mfg\my_single_mfg.bin

D:\esp-aliyun\config\mass_mfg>
```

cd 到 esp-aliyun 刚刚的工程目录下

cd \esp-aliyun\examples\mqtt\mqtt_example\



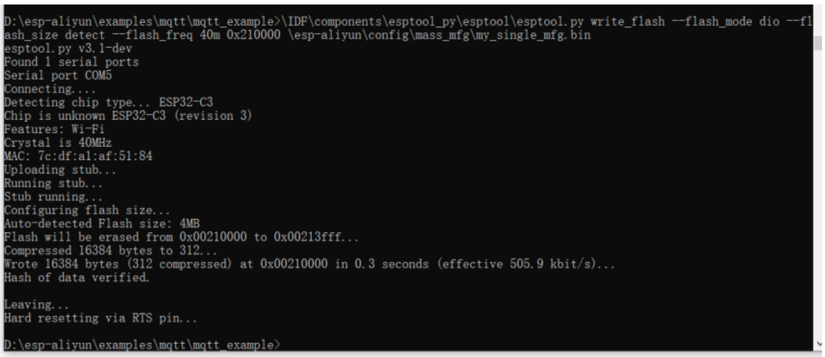
```
usage: nvs_partition_gen.py [-h] {generate,generate-key,encrypt,decrypt} ...
nvs_partition_gen.py: error: unrecognized arguments: generate my_single_mfg_confi

D:\esp-aliyun\config\mass_mfg>\IDF\components\nvs_flash\nvs_partition_generator\n
mfg_config.csv my_single_mfg.bin 0x4000

Creating NVS binary with version: V2 - Multipage Blob Support Enabled

Created NVS binary: ===> D:\esp-aliyun\config\mass_mfg\my_single_mfg.bin

D:\esp-aliyun\config\mass_mfg>cd \esp-aliyun\examples\mqtt\mqtt_example\
```

必须先烧入一遍程序 idf.py -p COM5 flash (需把 COM5 改成自己的端口号)

运行如下命令 将刚刚生成的 bin 文件烧入到板子的 0x210000 物理地址 (均需使用自己的 IDF 路径)

%userprofile%\components\esptool_py\esptool\esptool.py write_flash --flash_mode dio --flash_size detect --flash_freq 40m 0x210000 \esp-aliyun\config\mass_mfg\my_single_mfg.bin

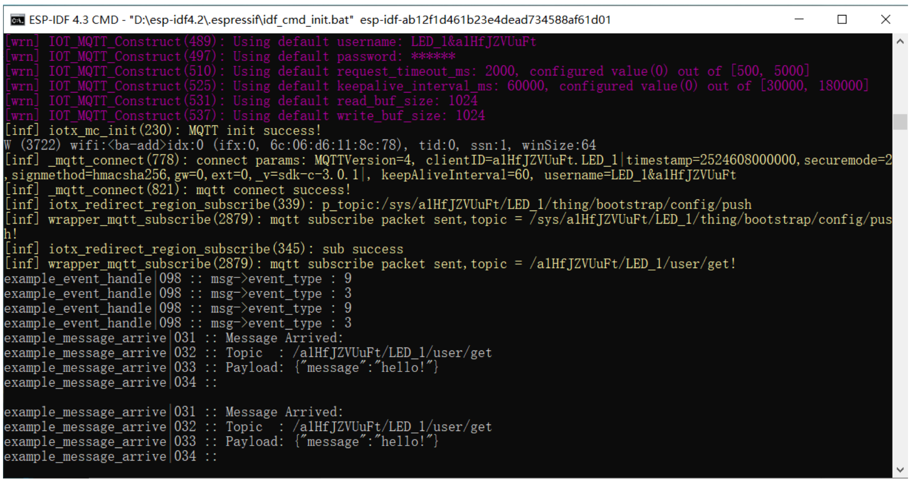
\esp-aliyun\必须替换成自己的 aliyun 存放路径



```
D:\esp-aliyun\examples\mqtt\mqtt_example>\IDF\components\esptool_py\esptool\esptool.py write_flash --flash_mode dio --fl
ash_size detect --flash_freq 40m 0x210000 \esp-aliyun\config\mass_mfg\my_single_mfg.bin
esptool.py v3.1-dev
Found 1 serial ports
Serial port COM5
Connecting....
Detecting chip type... ESP32-C3
Chip is unknown ESP32-C3 (revision 3)
Features: Wi-Fi
Crystal is 40MHz
MAC: 7c:df:a1:af:51:84
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Flash will be erased from 0x00210000 to 0x00213fff...
Compressed 16384 bytes to 312...
Wrote 16384 bytes (312 compressed) at 0x00210000 in 0.3 seconds (effective 505.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

D:\esp-aliyun\examples\mqtt\mqtt_example>
```

7. 实现

与阿里云通信便完成了


ESP-IDF 4.3 CMD - "D:\esp-idf4.2\.espressif\idf_cmd_init.bat" esp-idf-ab12f1d461b23e4dead734588af61d01

```
[wrn] IOT_MQTT_Construct(489): Using default username: LED_1&a1HfJZVUuFt
[wrn] IOT_MQTT_Construct(497): Using default password: ******
[wrn] IOT_MQTT_Construct(510): Using default request_timeout_ms: 2000, configured value(0) out of [500, 5000]
[wrn] IOT_MQTT_Construct(525): Using default keepalive_interval_ms: 60000, configured value(0) out of [30000, 180000]
[wrn] IOT_MQTT_Construct(531): Using default read_buf_size: 1024
[wrn] IOT_MQTT_Construct(537): Using default write_buf_size: 1024
[inf] iotx_mc_init(230): MQTT init success!
W (3722) wifi:<ba-add>idx:0 (ifx:0, 6c:06:d6:11:8c:78), tid:0, ssn:1, winSize:64
[inf] _mqtt_connect(778): connect params: MQTTVersion=4, clientID=a1HfJZVUuFt.LED_1|timestamp=2524608000000,securemode=2
,signmethod=hmacsha256,gw=0,ext=0,_v=sdk-c-3.0.1|, keepAliveInterval=60, username=LED_1&a1HfJZVUuFt
[inf] _mqtt_connect(821): mqtt connect success!
[inf] iotx_redirect_region_subscribe(339): p_topic:/sys/a1HfJZVUuFt/LED_1/thing/bootstrap/config/push
[inf] wrapper_mqtt_subscribe(2879): mqtt subscribe packet sent,topic = /sys/a1HfJZVUuFt/LED_1/thing/bootstrap/config/pus
h!
[inf] iotx_redirect_region_subscribe(345): sub success
[inf] wrapper_mqtt_subscribe(2879): mqtt subscribe packet sent,topic = /a1HfJZVUuFt/LED_1/user/get!
example_event_handle|098 :: msg->event_type : 9
example_event_handle|098 :: msg->event_type : 3
example_event_handle|098 :: msg->event_type : 9
example_event_handle|098 :: msg->event_type : 3
example_message_arrive|031 :: Message Arrived:
example_message_arrive|032 :: Topic  : /a1HfJZVUuFt/LED_1/user/get
example_message_arrive|033 :: Payload: {"message":"hello!"}
example_message_arrive|034 ::

example_message_arrive|031 :: Message Arrived:
example_message_arrive|032 :: Topic  : /a1HfJZVUuFt/LED_1/user/get
example_message_arrive|033 :: Payload: {"message":"hello!"}
example_message_arrive|034 ::
```