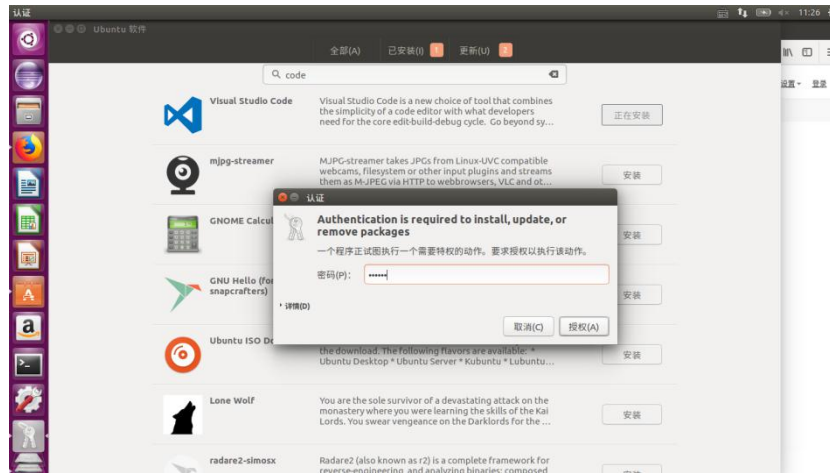


第一章 Ubuntu 搭建 ESP32 开发环境 (vscode)

1. Vscode 安装

ubuntu 软件中自带 vscode 的安装包，一键完成，不废话，没有的想办法解决。



2. 编译链操作

2.1. 下载

[官方最新地址](#)

有系统区别，下载对应的压缩到电脑

➤ for 64-bit Linux:

<https://dl.espressif.com/dl/xtensa-esp32-elf-linux64-1.22.0-80-g6c4433a-5.2.0.tar.gz>

➤ for 32-bit Linux:

<https://dl.espressif.com/dl/xtensa-esp32-elf-linux32-1.22.0-80-g6c4433a-5.2.0.tar.gz>

2.2. 解压

新建一个目录存放编译链的解压包 -> 进入目录 -> 解压。根据自己路径修改。

```
1 mkdir -p ~/esp
2 cd ~/esp
3 tar -xzf ~/Downloads/xtensa-esp32-elf-linux32-1.22.0-80-g6c4433a-5.2.0.tar.gz
```

3. ESP-IDF 操作

进入目录 -> git。

```
1 cd ~/esp
2 git clone --recursive https://github.com/espressif/esp-idf.git
```

4. 环境变量配置

打开配置文件。

```
1 gedit ~/.profile
```

添加编译链路径和 ESP-IDF 路径, 根据自己路径修改。

```
1 export PATH="$HOME/esp/xtensa-esp32-elf/bin:$PATH"
```

```
2 export IDF_PATH=~/.esp/esp-idf
```

5. 测试以上操作是否正常完美

5.1. 复制 ESP-IDF 中的工程到 esp 目录下,

```
1 cd ~/.esp
```

```
2 cp -r $IDF_PATH/examples/get-started/hello_world .
```

5.2. 编译, make all, 无错误说明环境正常。

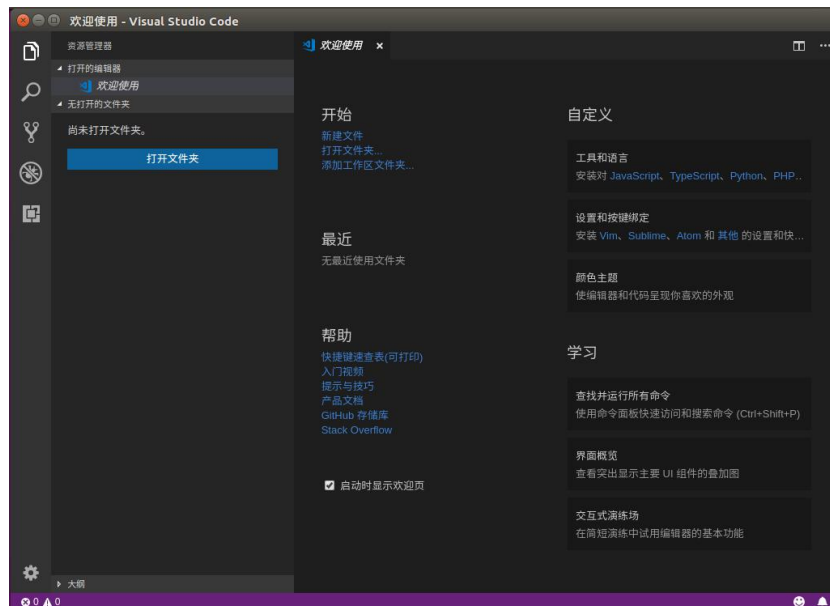
```
zs@zs-ThinkPad-X230: ~/.esp/hello_world
CC build/wpa_supplicant/src/crypto/crypto_internal-cipher.o
CC build/wpa_supplicant/port/os_xtensa.o
CC build/wpa_supplicant/src/fast_crypto/fast_aes-wrap.o
CC build/wpa_supplicant/src/fast_crypto/fast_sha256-internal.o
CC build/wpa_supplicant/src/fast_crypto/fast_aes-cbc.o
CC build/wpa_supplicant/src/fast_crypto/fast_crypto_internal.o
CC build/wpa_supplicant/src/fast_crypto/fast_sha256.o
CC build/wpa_supplicant/src/fast_crypto/fast_aes-unwrap.o
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
CC build/xtensa-debug-module/eri.o
CC build/xtensa-debug-module/trax.o
AR build/xtensa-debug-module/libxtensa-debug-module.a
LD build/hello-world.elf
esptool.py v2.1
To flash all build output, run 'make flash' or:
python /home/zs/esp32/sources/esp-idf/components/esptool_py/esptool/esptool.py
--chip esp32 --port /dev/ttyUSB0 --baud 115200 --before default_reset --after ha
rd_reset write_flash -z --flash_mode dio --flash_freq 40m --flash_size detect 0x
10000 /home/zs/esp/hello_world/build/bootloader/bootloader.bin 0x10000 /home/zs
/esp/hello_world/build/hello-world.bin 0x8000 /home/zs/esp/hello_world/build/pa
rtitions_singleapp.bin
zs@zs-ThinkPad-X230: ~/.esp/hello_world$
```

5.3. 下载, make flash 前需要 make menuconfig 配置下载串口号

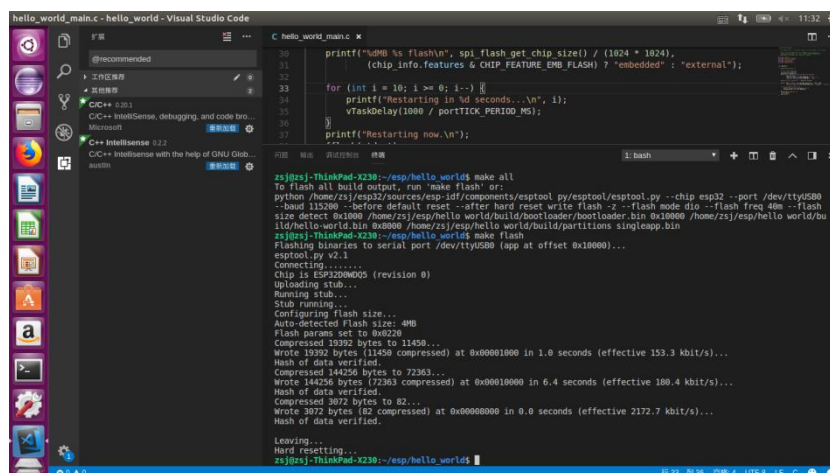
```
zs@zs-ThinkPad-X230: ~/.esp/hello_world
rtitions_singleapp.bin
zs@zs-ThinkPad-X230: ~/.esp/hello_world$ make flash
Flashing binaries to serial port /dev/ttyUSB0 (app at offset 0x10000)...
esptool.py v2.1
Connecting...
Chip is ESP32D0WDQ5 (revision 0)
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0220
Compressed 19392 bytes to 11450...
Wrote 19392 bytes (11450 compressed) at 0x00001000 in 1.0 seconds (effective 152
.9 kbit/s)...
Hash of data verified.
Compressed 144256 bytes to 72363...
Wrote 144256 bytes (72363 compressed) at 0x00001000 in 6.4 seconds (effective 18
0.5 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 82...
Wrote 3072 bytes (82 compressed) at 0x00008000 in 0.0 seconds (effective 2161.9
kbit/s)...
Hash of data verified.
Leaving...
Hard resetting...
zs@zs-ThinkPad-X230: ~/.esp/hello_world$
```

6. Vscode 操作

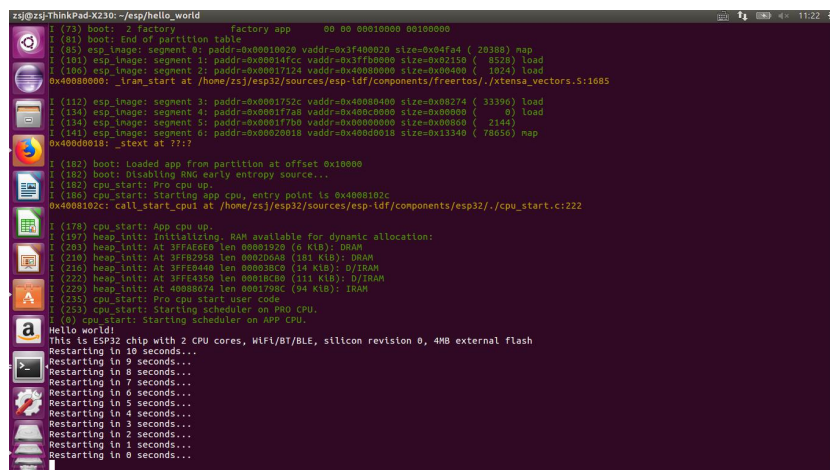
6.1. 打开 vscode 软件，打开刚才的 helloworld 文件夹。



6.2. Ctrl+` 打开终端，输入 make all 编译，make flash 下载。太简单了



6.3. 串口监测，运行正常



6.4. ubuntu 搭建 ESP32 环境 (vscode) 总结

- ubuntu 下搭建环境特别简单。
- 对比 ubuntu 和 Windows 速度

指令	ubuntu	Windows(未优化)
make all	56 秒	5 分钟
make flash	13 秒	24 秒

速度说明:

- 其中没有对 make 其他指令对比
 - 电脑不同编译速度也不一样
- 对比可以得出:
 - 开发产品使用 ubuntu 搭建 ESP32 开发环境
 - 有 ubuntu 系统的使用 ubuntu 搭建 ESP32 开发环境
 - 体验使用 windows 搭建 ESP32 开发环境, 方便