# ESP32 系列教程之六：
# WSL 搭建 esp-idf 环境

Date: 2021.7.10

Version: V1.0

# 关于文档

本文档为 ESP32 教程系列，旨在为客户进行 ESP32 系列芯片开发提供环境搭建、工程示例演示等方面的参考文档及视频演示，降低 ESP32 系列芯片、模组开发的入门难度。

ESP32 教程系列文档主要参考于乐鑫官网提供的 ESP32 入门教程：https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32/get-started/index.html。由于个人经验有限，文档编写过程中难免存在失误、错漏之处，欢迎广大开发爱好者对本文档提出批评建议。

版本信息：

| 日期 | 版本 | 作者 | 说明 |
| --- | --- | --- | --- |
| 2021.7.10 | V1.0 | Amiliya | 首次发布 |

文档变更通知：以启明云端官网版本为准，恕不另行通知。

意见提交邮箱：amiliya@wireless-tag.com

# 目　录

# 1 WSL 搭建 esp-idf 环境

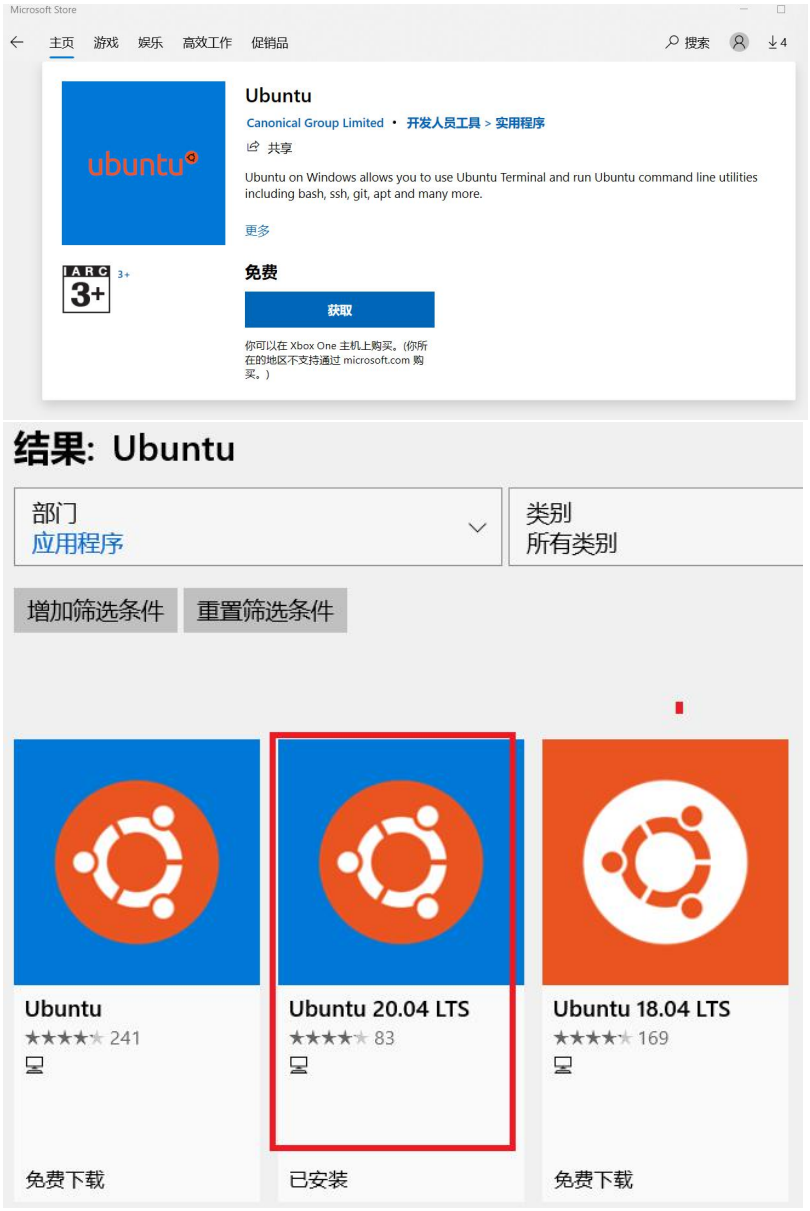
## 1.1 开启 WSL

快捷键 win+s 搜索：“启用或关闭 Windows 功能”，勾选“适用于 Linux 的 Windows 子系统”，确定后重启电脑。


## 1.2 准备 WSL

下载 Ubuntu20，以下两种方式任选一种（推荐方式一）。

方式一：快捷键 win+s 搜索：“Microsoft Store”，搜索 Ubuntu，下载 Ubuntu20.04




方式二：前往下方网址下载 Ubuntu20.04：（下载很慢，且容易失败）

https://docs.microsoft.com/zh-cn/windows/wsl/install-manual

## 1.3 工具准备

（1）更新“应用市场”

```
sudo apt-get update
```

（2）安装 esp-idf 相关应用

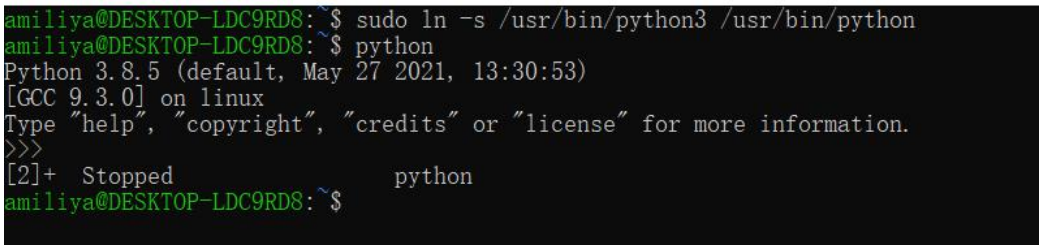
```
sudo apt-get install git wget flex bison gperf python3 python3-pip python3-setuptools cmake ninja-build ccache libffi-dev libssl-dev dfu-util libusb-1.0-0
```

（3）设置默认 python

```
sudo ln -s /usr/bin/python3 /usr/bin/python
```

（4）查看当前 python 版本（大于 3.6）

```
python
```

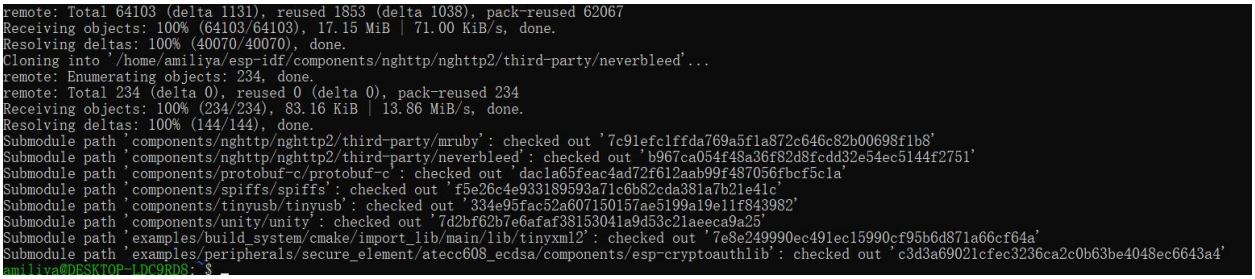



## 1.4 获取 ESP-IDF

使用命令下载 ESP-IDF

```
git clone -b release/v4.3 --recursive https://github.com.cnpmjs.org/espressif/esp-idf.git
```

（注：-b release/v4.3 表示 esp-idf 下载版本，需要下载其它版本或版本介绍请移至乐鑫官网查看具体介绍




## 1.5 安装工具包

进入 esp-idf 文件夹，运行./install.sh

若安装工具时速度慢，可在执行./install.sh 命令前执行以下命令，优先选择 Espressif 下载服务器进行下载：export IDF_GITHUB_ASSETS="dl.espressif.com/github_assets"

运行脚本激活 esp-idf

```
. ./export.sh
```

## 1.6 将 esp-idf 环境加入环境变量

复制并粘贴以下命令到 shell 配置文件中（`.profile`，`.bashrc`，`.zprofile` 等）

```
alias get_idf='. $HOME/esp_4.3/esp-idf/export.sh'
```

(注：文件路径需要正确，如文件夹名称 esp_4.3)

以`.bashrc` 为例：

打开`.bashrc` 文件

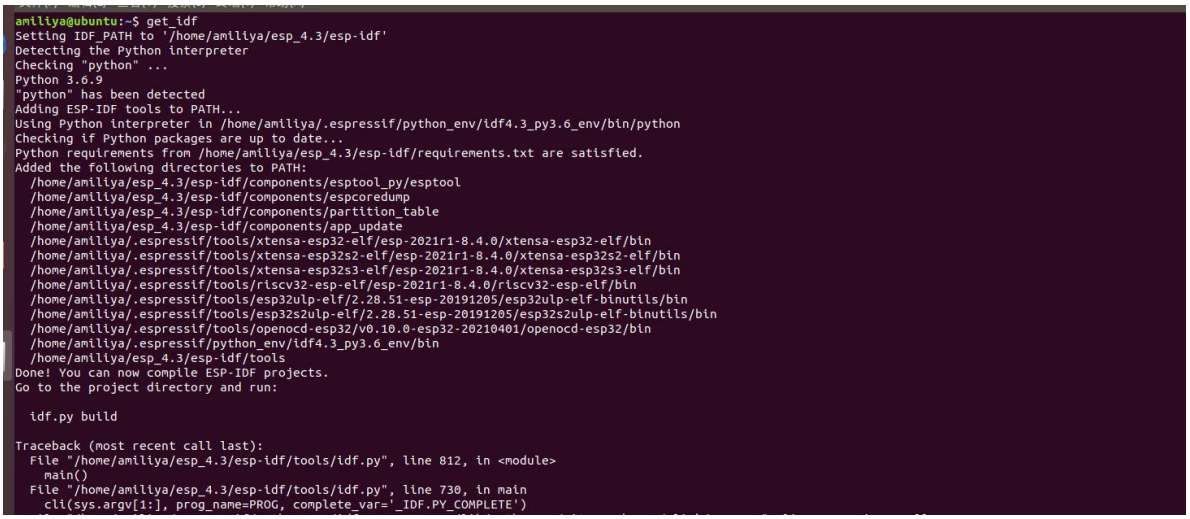
```
vim .bashrc
```

添加命令：（需要不同版本的 IDF 时都可以加入命令，只要变量名 `get_idf` 不同即可）

```
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

alias get_idf='. $HOME/esp_4.3/esp-idf/export.sh'
```

保存关闭，重启 Ubuntu 或使用命令刷新配置文件：`source ~/.bashrc`

最终效果如下：执行 `get_idf` 便可以让端口进入 esp-idf 环境：



# 2 运行 hello_world 示例

## 2.1 拷贝工程

将 `esp-idf/examples/get-started/` 目录下的 `hello_world` 示例拷贝到 `esp_4.3` 下

```
cp esp-idf/examples/get-started/hello_world/ ./ -r
```

```
amiliya@ubuntu:~/esp_4.3$ cp esp-idf/examples/get-started/hello_world/ ./ -r
amiliya@ubuntu:~/esp_4.3$ ls
esp-idf  hello_world
amiliya@ubuntu:~/esp_4.3$
```

## 2.2 工程配置

进入 hello_world 示例，进行工程配置：

```
cd ~/esp_4.3/hello_world
```

```
idf.py set-target esp32
```

(使用其它芯片请选择对应芯片类型，如：esp32c3)

(打开一个新项目后，应首先设置“目标”芯片 `idf.py set-target esp32`。注意，此操作将清除并初始化项目之前的编译和配置（如有）。 您也可以直接将“目标”配置为环境变量（此时可跳过该步骤）)

```
idf.py menuconfig
```

如果之前的步骤都正确，则会显示下面的菜单：

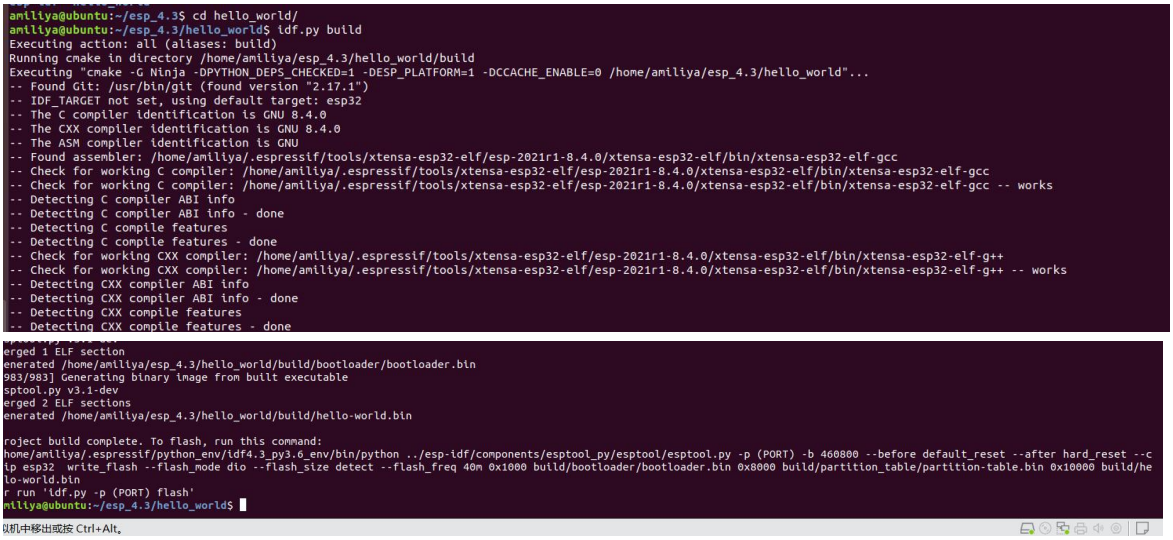
```
(Top)
                Espressif IoT Development Framework Configuration
    SDK tool configuration  --->
    Build type  --->
    Application manager  --->
    Bootloader config  --->
    Security features  --->
    Serial flasher config  --->
    Partition Table  --->
    Compiler options  --->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

*工程配置 — 主窗口*

您可以通过此菜单设置项目的具体变量，包括 Wi-Fi 网络名称、密码和处理器速度等. `hello_world` 示例项目会以默认配置运行，因此可以跳过使用 `menuconfig` 进行项目配置这一步骤。

## 2.3 编译工程

```
idf.py build
```

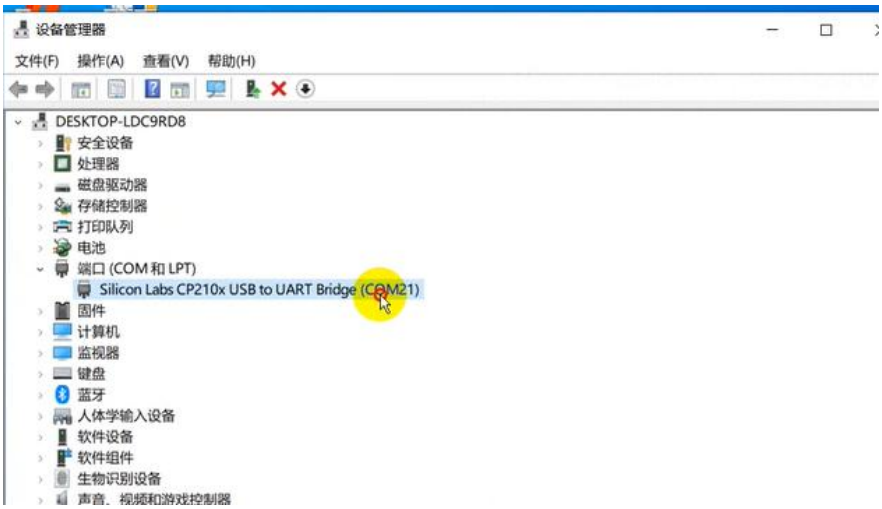



## 2.4 硬件连接

用 USB 线将 ESP32 模组与电脑连接：

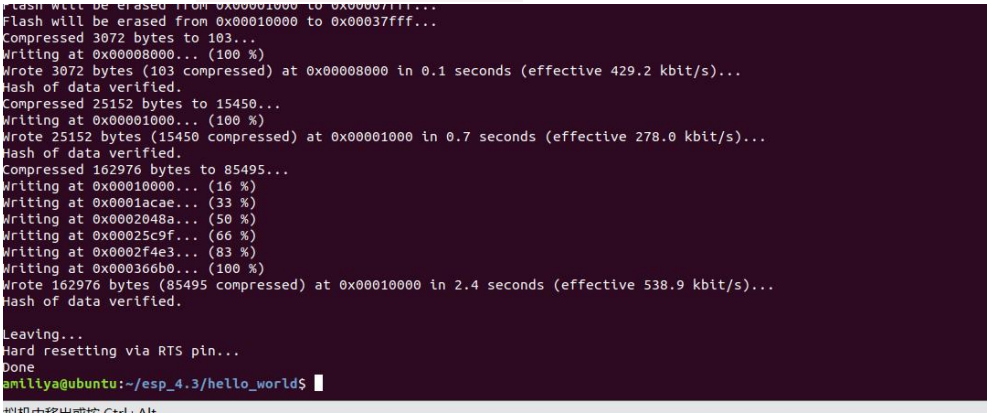
## 2.5 查看端口

在开始菜单中选择设备管理器查看当前端口号：



## 2.6 工程烧录

进行烧录：`idf.py -p /dev/ttyS21 flash`

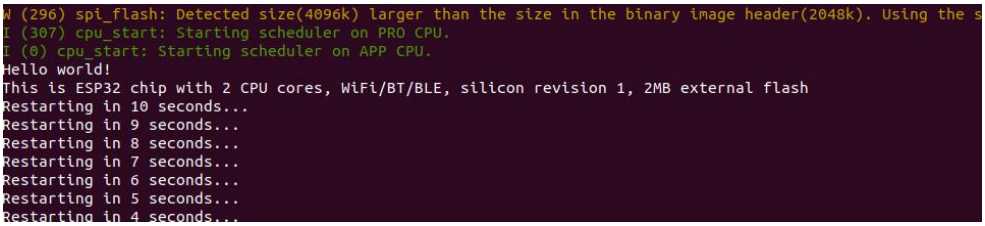



烧录成功！

(WSL 中的端口与 Windows 端口的对应关系：COM21 - S21)

## 2.7 监视工程

查看监视器：`idf.py -p /dev/ttyS21 monitor`




成功打印 hello_world.

# 3 参考视频

（注：为保证视频质量，演示视频并未录制声音，同时对于一些长时间下载过程进行了缩略录制，视频仅作为本文档参考文件。）

## 3.1 WSL 搭建 esp-idf 环境

ESP32系列教程（
10）WSL搭建esp

## 3.2 VS Code 远程连接 WSL

ESP32系列教程（
11）：VS Code远

# 4 后　记

## 4.1 注意事项

（1）Ubuntu 版本建议在 18.0 以上，搭建 esp-idf 环境需要保证网络通畅

（2）python 版本必须保证 3.6 以上且为系统默认 python

（3）环境变量可添加多个，可通过调用不同变量名选择不同版本的 IDF

（4）工程配置中选择目标芯片的过程对某些芯片是必要的，如 ESP32-C3

（5）进行 ESP32-C3 的开发必需使用最新版（V4.3）IDF 的支持

## 4.2 常见问题

由于 esp-idf 环境搭建过程中由于各种原因容易遇到各种问题，此章节整理了一些常见的问题以供大家参考，其它相关问题可提交至意见提交邮箱。

### 4.2.1 ESP-IDF 的获取

部分开发者由于网络等原因无法从 github 获取 esp-idf，可参考下方步骤从国内的 gitee 获取 esp-idf。

（1）获取 esp-idf

```
git clone -b release/v4.3 https://gitee.com/EspressifSystems/esp-idf.git
```

```
amiliya@DESKTOP-LDC9RD8:~/esp$ git clone -b release/v4.3 https://gitee.com/EspressifSystems/esp-idf.git
正克隆到 'esp-idf'...
remote: Enumerating objects: 30044, done.
remote: Counting objects: 100% (30044/30044), done.
remote: Compressing objects: 100% (11812/11812), done.
remote: Total 250881 (delta 20441), reused 24664 (delta 17190), pack-reused 220837
接收对象中: 100% (250881/250881), 139.96 MiB | 5.57 MiB/s, 完成.
处理 delta 中: 100% (185486/185486), 完成.
正在更新文件: 100% (8306/8306), 完成.
amiliya@DESKTOP-LDC9RD8:~/esp$ cd esp-idf/
```

（2）下载工具

```
git clone https://gitee.com/EspressifSystems/esp-gitee-tools.git
```

```
amiliya@DESKTOP-LDC9RD8:~/esp$ git clone https://gitee.com/EspressifSystems/esp-gitee-tools.git
正克隆到 'esp-gitee-tools'...
remote: Enumerating objects: 48, done.
remote: Counting objects: 100% (48/48), done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 48 (delta 19), reused 0 (delta 0), pack-reused 0
展开对象中: 100% (48/48), 15.36 KiB | 249.00 KiB/s, 完成.
amiliya@DESKTOP-LDC9RD8:~/esp$ ls
esp-gitee-tools  esp-idf
amiliya@DESKTOP-LDC9RD8:~/esp$
```

（3）下载子模块和工具链

进入工具文件夹，保存路径

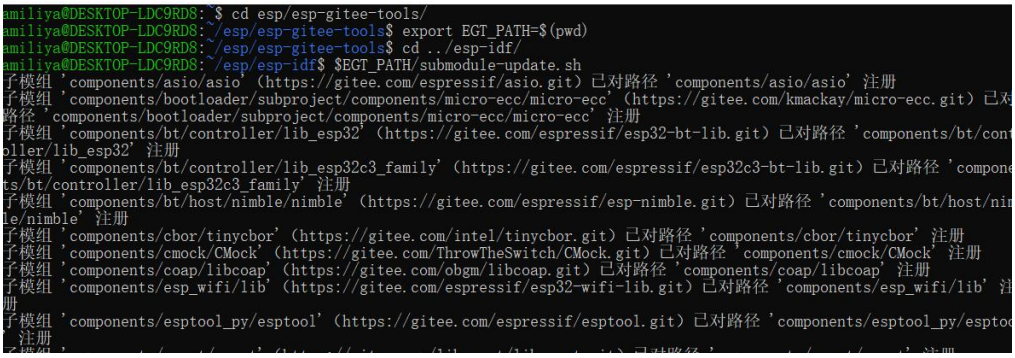
```
cd esp-gitee-tools
```

```
export EGT_PATH=$(pwd)
```

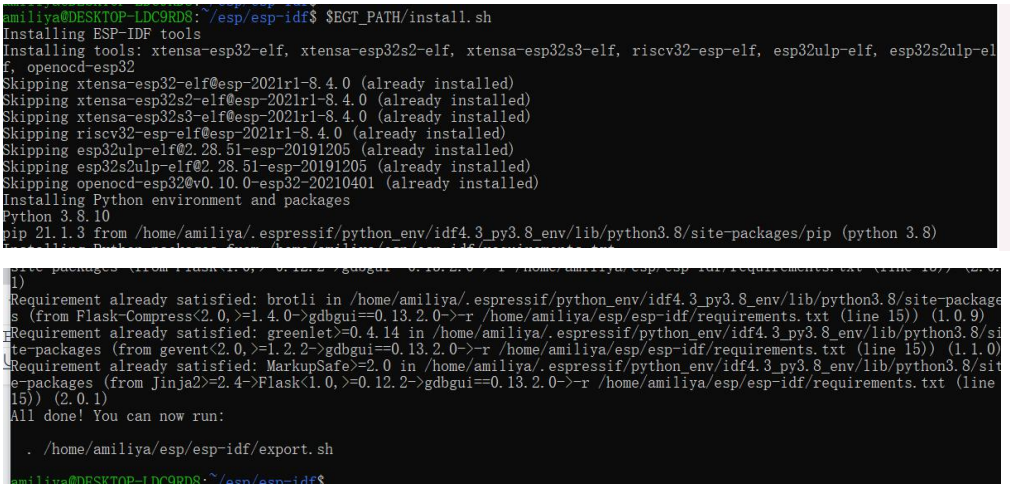
进入 esp-idf 文件夹，下载子模块和工具链

```
cd ~/esp-idf/
```

```
$EGT_PATH/submodule-update.sh
```




```
$EGT_PATH/install.sh
```

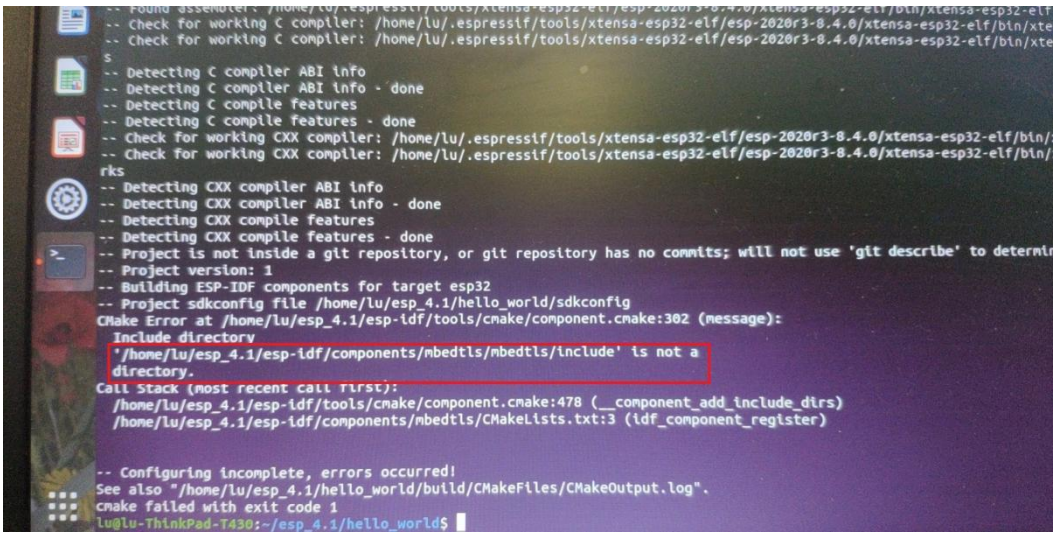






## 4.2.2 子模组缺失

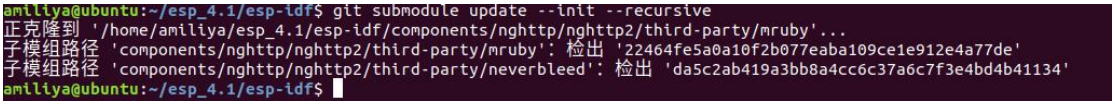
绝大多数开发者所遇到的无法启动菜单、工程编译失败等问题都源于 ESP-IDF 子模组未下载完全，因此保证子模组的完全下载对完整搭建 esp-idf 环境至关重要。




缺少相应子模块时的错误 log

在 esp-idf 文件夹下使用以下命令下载缺失子模组

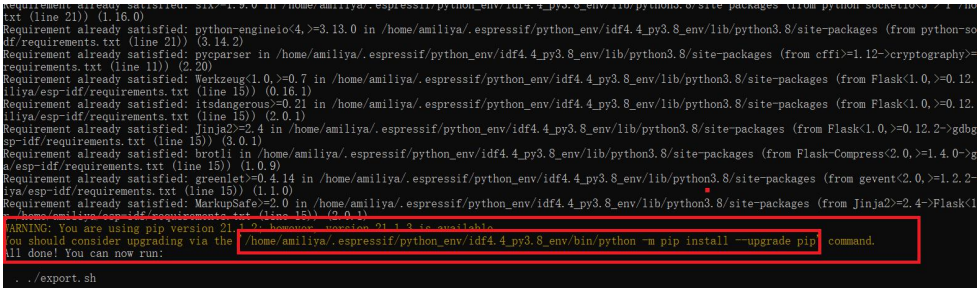
```
git submodule update --init --recursive
```




若命令长时间无响应可取消后重试，此命令也可用于检测子模组是否下载完全（下载完全则直接返回命令行），也可进入缺失子模组的目录下单独克隆该子模组。

### 4.2.3 工具链设置

少部分开发者在搭建环境时会遇到此类问题，可依据错误提示 log 做出相应处理，以下仅供参考。




按照提示执行相应命令可消除警告（pip 版本问题）

### 4.2.3 烧录问题

在 WSL 下进行烧录时出现以下错误：

按照提示执行相应命令或更改波特率可正常进行烧录

```
idf.py -p  /dev/ttyS21  -b 1152000 flash
```