因为学习ESP8266(ESP-12S)，为此大大小小已经重装过ubuntu好几次了，特开一帖，记录ESP8266 SDK 3.0搭建过程，从装完操作系统开始一步一步搭建，希望大家能避免不必要的坑。

## Linux搭建环境

虚拟机：VMware Workstation 16 Pro

Ubuntu操作系统：ubuntu-18.04.5-desktop-amd64.ios

阿里云镜像：(比较接近我目前的版本)

<https://mirrors.aliyun.com/ubuntu-releases/bionic/ubuntu-18.04.6-desktop-amd64.iso>

搭建环境教程这边就不细说了，自行百度参考。

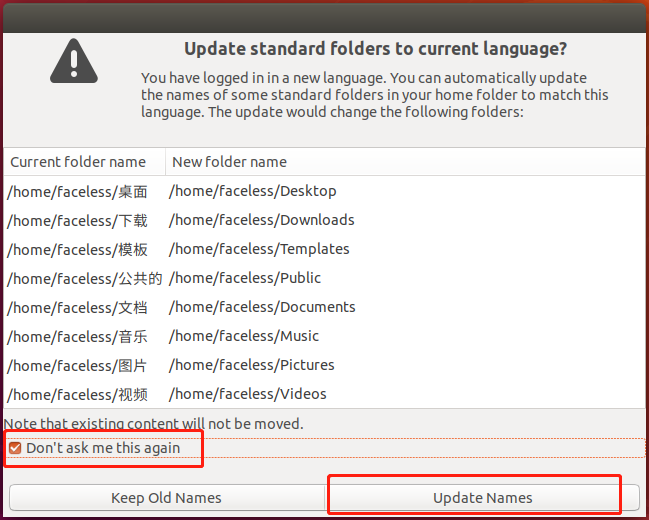
## 硬件准备

硬件型号：安信可ESP-12S

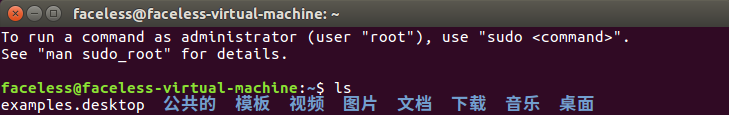
## Linux下软件安装

家目录下的中文文件名切换英文文件名，方便终端操作

1. 设置🡪区域与语言🡪选择英语🡪重启🡪弹窗勾选不再提醒并选择更新文件名



1. 设置🡪语言支持🡪选择中文🡪重启🡪完成





## Vmware tools

解决屏幕固定比例和部分操作问题

1. 虚拟机菜单栏🡪虚拟机🡪安装VMware Tools
2. 光盘文件在/media/faceless/下
3. 复制VMware Tools文件到当前目录

sudo cp -r /media/faceless/VMware\ Tools/ .

1. 进入VMware Tools并解压VMwareTools-10.3.22-15902021.tar.gz

cd VMware\ Tools/

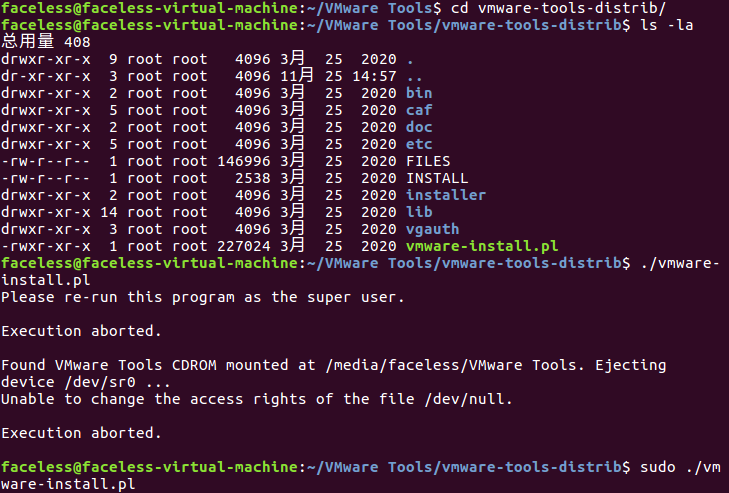
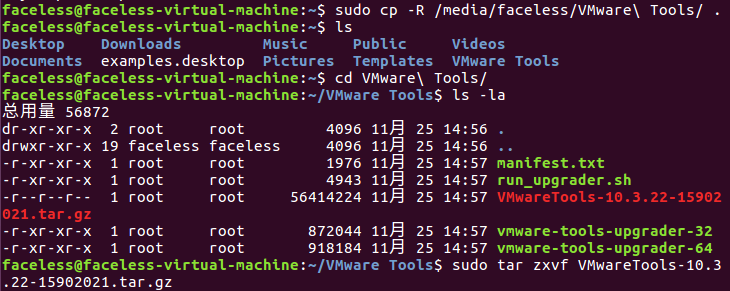
ls -la

sudo tar -zxvf VMwareTools-10.3.22-15902021.tar.gz

1. 执行sh文件安装VMware Tools

cd vmware-tools-distrib/

sudo ./vmware-install.pl



## Vim编辑器

sudo apt-get purge vim-common

sudo apt-get install vim

## 使用阿里云镜像源

备份linux下载源文件

sudo cp /etc/apt/sources.list /etc/apt/sources.list.backup

sudo vim /etc/apt/sources.list

阿里云Ubuntu镜像源获取

<https://developer.aliyun.com/mirror/ubuntu?spm=a2c6h.13651102.0.0.3e221b11kWcw0Z>

将符合改ubuntu版本的镜像源加入并按ESC再输入:wq!保存，最后需要更新一下源

sudo apt-get update

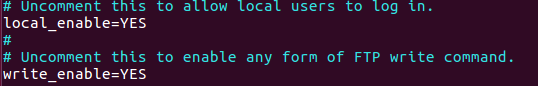
## Windows与ubuntu文件互传（FTP服务和FTP客户端FileZilla）

FTP服务

sudo apt-get install vsftpd

修改配置文件vsftpd.conf

sudo vim /etc/vsftpd.conf



sudo /etc/init.d/vsftpd restart

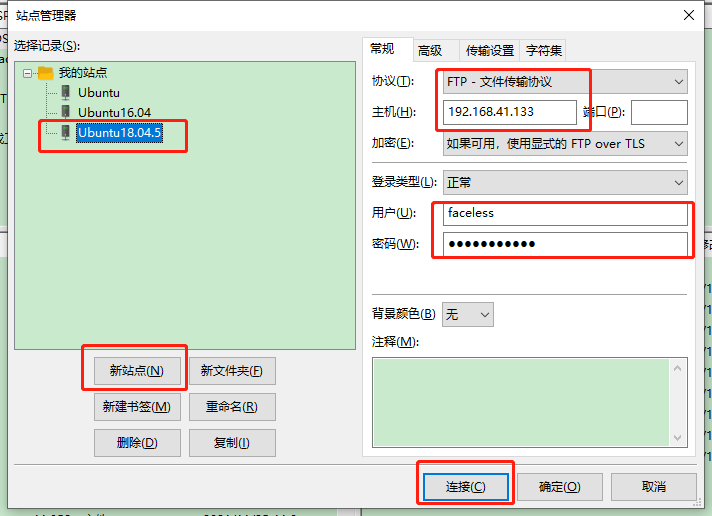


下载网络工具(使用ifconfig)

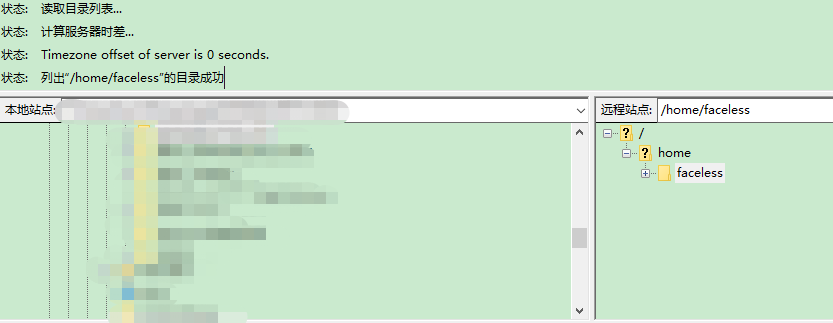
sudo apt install net-tools

FileZilla下载：<https://www.filezilla.cn/download/client>

在windows下打开FileZilla客户端，在菜单栏选择文件🡪站点管理🡪新建站点配置如图



此时连接成功，左侧为windows下的文件，右侧为ubuntu下的文件



## 基本环境

最新软件包文档<https://docs.espressif.com/projects/esp-idf/zh_CN/latest/esp32/get-started/linux-setup.html>

安装软件依赖包

sudo apt-get install git wget flex bison gperf python python-pip python-setuptools cmake ninja-build ccache libffi-dev libssl-dev dfu-util libusb-1.0-0 libncurses5-dev

python-serial python-click python-cryptography python-future python-pyparsing python-pyelftools

安装时间比较漫长，耐心等待…

## ESP8266工具链下载

xtensa-lx106-elf-gcc8\_4\_0-esp-2020r3-linux-amd64.tar.gz

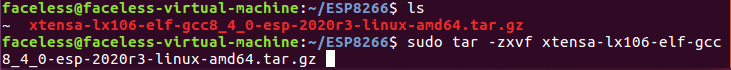
<https://github.com/espressif/ESP8266_RTOS_SDK>



这边我已经下载完了，通过FilaZilla客户端传到Ubuntu

解压工具链压缩包到当前目录

sudo tar -zxvf xtensa-lx106-elf-gcc8\_4\_0-esp-2020r3-linux-amd64.tar.gz



修改其权限

sudo chmod 777 -R xtensa-lx106-elf

## 获取ESP8266\_RTOS\_SDK代码

* 官方拉取代码

sudo git clone --recursive https://github.com/espressif/ESP8266\_RTOS\_SDK.git

* 这边建议使用半颗心脏博主贡献的源仓库

拉取源码

sudo git clone <https://gitee.com/xuhongv/ESP8266_RTOS_SDK.git>

修改ESP8266\_RTOS\_SDK权限

sudo chmod 777 -R ESP8266\_RTOS\_SDK/

由于该子模块不规范，需要对.gitmodules修改

cd ESP8266\_RTOS\_SDK/

sudo vim .gitmodules

将文件里面的内容修改成以下内容并按ESC再输入:wq!保存

[submodule "components/json/cJSON"]

path = components/json/cJSON

url = https://gitee.com/xuhongv/cJSON.git

[submodule "components/mbedtls/mbedtls"]

path = components/mbedtls/mbedtls

url = https://gitee.com/xuhongv/mbedtls.git

[submodule "components/lwip/lwip"]

path = components/lwip/lwip

url = https://gitee.com/xuhongv/esp-lwip.git

[submodule "components/mqtt/esp-mqtt"]

path = components/mqtt/esp-mqtt

url = https://gitee.com/xuhongv/esp-mqtt.git

[submodule "components/coap/libcoap"]

path = components/coap/libcoap

url = https://gitee.com/xuhongv/libcoap.git

更新拉取子模块

sudo git submodule update --init --recursive

设置环境变量

工具链路径:/home/faceless/ESP8266/xtensa-lx106-elf



SDK路径:/home/faceless/ESP8266/ESP8266\_RTOS\_SDK



修改bashrc文件内容

sudo vim ~/.bashrc

添加以下内容并按ESC再输入:wq!保存

export PATH=$PATH:/home/faceless/ESP8266/xtensa-lx106-elf/bin

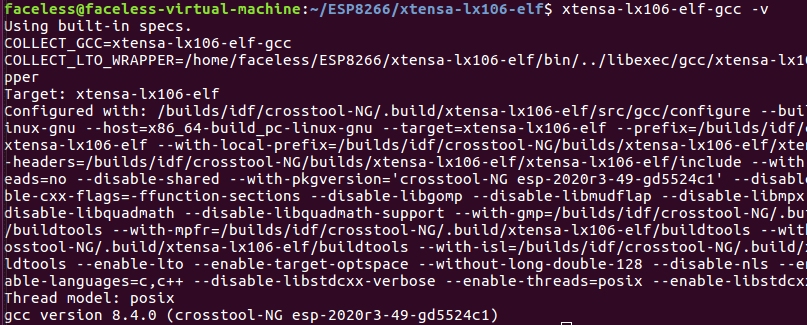
export IDF\_PATH=/home/faceless/ESP8266/ESP8266\_RTOS\_SDK

更新bashrc文件内容

source ~/.bashrc

查看工具链设置状态

xtensa-lx106-elf-gcc -v



查看SDK设置状态

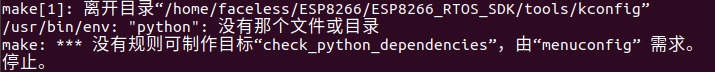
echo $IDF\_PATH



编译示例代码

cd examples/get-started/hello\_world/

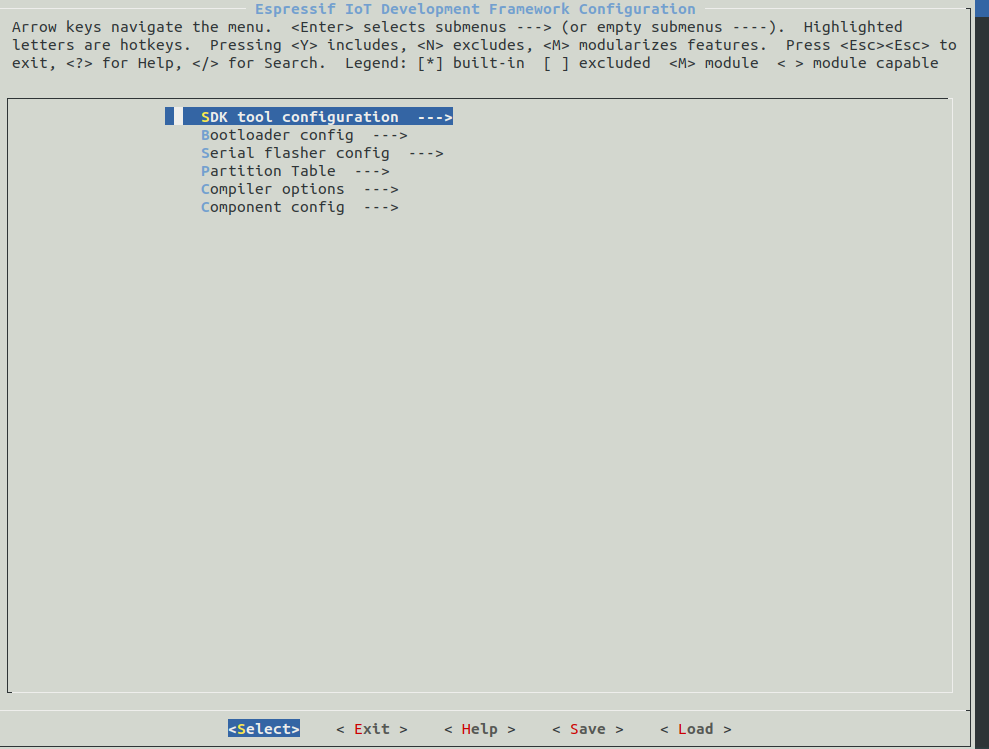
make menuconfig



之后make menuconfig还是会报错，根据提示需要更新requirements.txt文件

/usr/bin/python -m pip install --user -r /home/faceless/ESP8266/ESP8266\_RTOS\_SDK/requirements.txt

终于可以了！make menuconfig

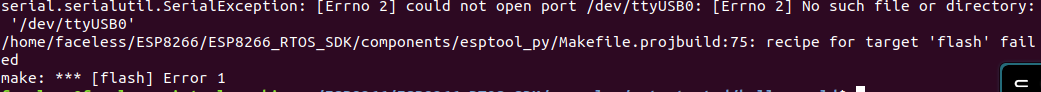


编译代码

make all

烧写代码

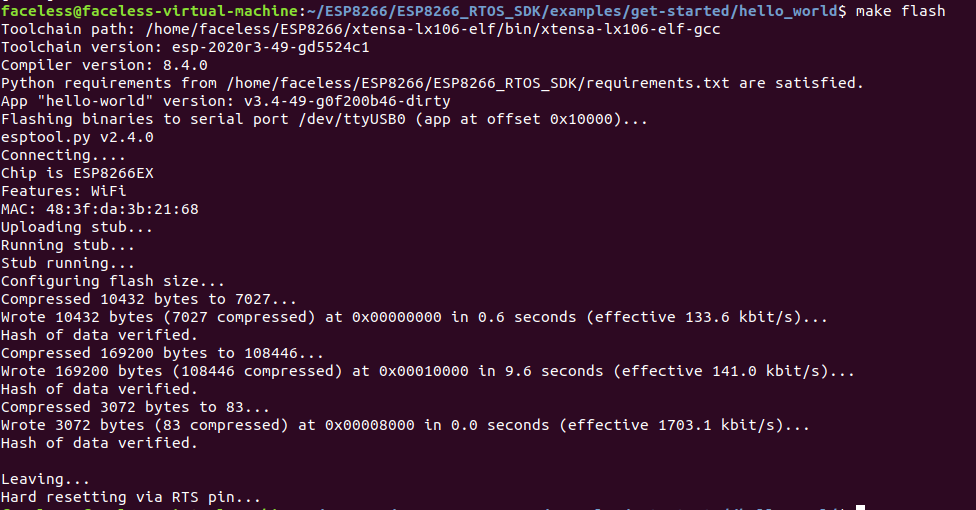
make flash



以上问题可能有两种

* 硬件未连接，将硬件连接到虚拟机下
* 未设置串口权限，需要sudo chmod 777 /dev/ttyUSB0

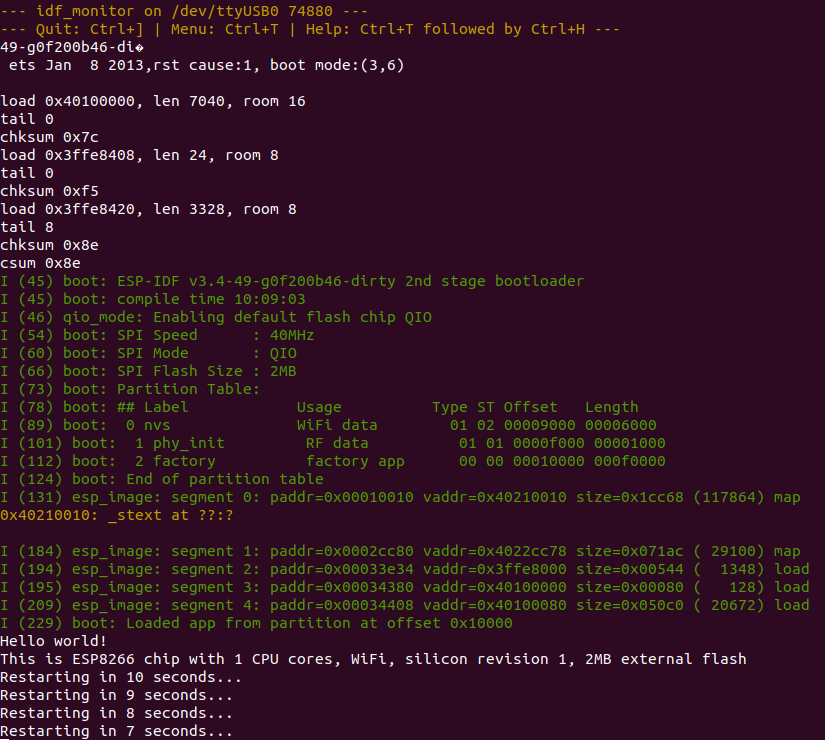
若问题解决的话 会连接硬件并烧写程序

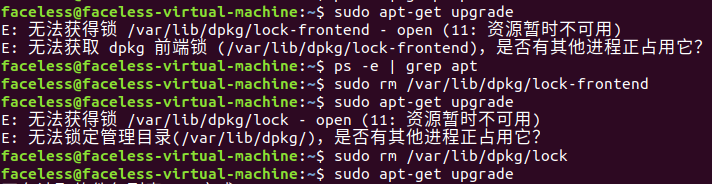


串口显示

make monitor

一般我直接使用make flash monitor





以上内容参考

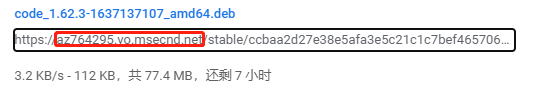
博主半颗心脏<https://blog.csdn.net/xh870189248/article/details/104736261?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522162399911016780274118798%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fblog.%2522%257D&request_id=162399911016780274118798&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~first_rank_v2~rank_v29-1-104736261.nonecase&utm_term=esp8266+linux&spm=1018.2226.3001.4450>

【正点原子】I.MX6U嵌入式Linux驱动开发指南V1.5/第四章 开发环境搭建

vscode

下载地址<https://code.visualstudio.com/>

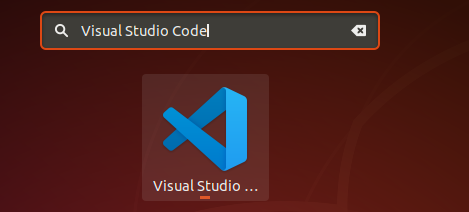
在该网站内可以选择相应版本，国外网站下载很慢，容易下到一半就出现网络连接错误，所以这边需要使用国内镜像下载，将图示红框内地址替换成vscode.cdn.azure.cn即可高速下载，几秒就搞定，通过FilaZilla传到ubuntu



安装code\_1.62.3-1637137107\_amd64.deb

sudo dpkg -i code\_1.62.3-1637137107\_amd64.deb

在应用程序中搜索Visual Studio Code即可找到该软件



vscode插件安装

vscode支持

我们需要按照的插件有下面几个：  
1)、 C/C++，这个肯定是必须的。  
2)、 C/C++ Snippets，即 C/C++重用代码块。  
3)、 C/C++ Advanced Lint,即 C/C++静态检测 。  
4)、 Code Runner，即代码运行。  
5)、 Include AutoComplete，即自动头文件包含。  
6)、 Rainbow Brackets，彩虹花括号，有助于阅读代码。  
//7)、 One Dark Pro， VSCode 的主题。  
//8)、 GBKtoUTF8，将 GBK 转换为 UTF8。  
9)、 ARM，即支持 ARM 汇编语法高亮显示。  
10)、 Chinese(Simplified)，即中文环境。  
11)、 vscode-icons， VSCode 图标插件，主要是资源管理器下各个文件夹的图标。  
//12)、 compareit，比较插件，可以用于比较两个文件的差异。  
//13)、 DeviceTree，设备树语法插件。  
14)、 TabNine，一款 AI 自动补全插件，强烈推荐，谁用谁知道

C/C++ Extension Pack

