

# # jupyter notebooks学习使用

[清华镜像Anaconda] (<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>)  
[pycharm社区下载] (<http://www.jetbrains.com/pycharm/download/#section=windows>)  
[pycharm和anaconda B站安装教程视频] (<https://b23.tv/XzPD3J1>)  
[jupyter note使用教学] (<https://www.zhihu.com/zvideo/1497909776620761088>)

# STM32单片机

B站学习视频 ([https://www.bilibili.com/video/BV1yW411Y7Gw?share\\_source=copy\\_pc](https://www.bilibili.com/video/BV1yW411Y7Gw?share_source=copy_pc))

## 学习记录

### 工程建立

添加工程必要文件 Libraries->CMSIS -> CM3-> DeviceSupport-> ST-> STM32F10x-> startup-> arm->是启动文件，复制到工程模板（新建Start），回到STM32F10x（复制那三个文件stmxxxh、systemxxx.c、systemxxx.h）复制到Start,打开CM3->CoreSupport(将两个文件复制到Start)回到Keil将文件添加到工程->点击Target1,将Source Group1单击改名Start->右键选择添加已存在文件，打开Start文件夹，打开筛选器All files,添加启动文件（视频所用型号添加后缀md.s文件）和剩下所有.c.h文件->在工程选项添加该文件夹头文件路径（魔术棒按钮，C/C++,Include Paths栏点右边三个点按钮新建路径再点右三个点按钮添加Start路径，最后点ok）->打开新建工程文件夹添加新文件夹User（放main函数），接着keil里Target右键新建（组别）改名User添加文件选.c名叫main（注意路径选择Userwenj夹），在main里右键插入头文件stm32f10x.h(若用寄存器开发32，到此就完成工程建立) 扳手里Encoding->UTF-8可以防止中文乱码

### 错误

针对2-2新建工程中写了main函数进行编译时出现错误的问题：

D:\Keil5\ARM\PACK\Keil\STM32F1xx\_DFP\2.2.0\Device\Include\stm32f10x.h(483): error: #5: cannot open source input file "core\_cm3.h": No such file or directory 这是因为装的MDK版本太新了，比如MDK5.12/5.13，它们不会从MDK安装目录去查找头文件。所以导致这个错误。解决方法如下：把core\_cm3.h文件复制到keil软件安装的include目录下： D:\MDK\install\ARM\PACK\Keil\STM32F1xx\_DFP\2.2.0\Device\Include

In [ ]:

```

流水灯闪烁
void main(void)
{
    unsigned int i, j;
    *(unsigned int*) 0X40021018 |= (1<<3); //GPIOB时钟开启
    *(unsigned int*) 0x40010c00 |= (1<<(4*0)); //配置PB0为推挽10Mhz输出模式
    *(unsigned int*) 0x40010c00 |= (1<<(4*1)); //配置PB1为推挽10Mhz输出模式
    *(unsigned int*) 0x40010c00 |= (1<<(4*5)); //配置PB5为推挽10Mhz输出模式
    while(1)
    {
        *(unsigned int*) 0x40010c0c &= ~(1<<0); //配置PB0输出0
        for (i=1000;i>0;i--) {for (j=1000;j>0;j--) ;}
        *(unsigned int*) 0x40010c0c |= (1<<0); //配置PB0输出1, 因为PB0, PB1, PB5共用1个LED, 故在输出下一
        *(unsigned int*) 0x40010c0c &= ~(1<<1); //配置PB1输出0
        for (i=1000;i>0;i--) {for (j=1000;j>0;j--) ;}
        *(unsigned int*) 0x40010c0c |= (1<<1); //配置PB1输出1
        *(unsigned int*) 0x40010c0c &= ~(1<<5); //配置PB5输出0
        for (i=1000;i>0;i--) {for (j=1000;j>0;j--) ;}
        *(unsigned int*) 0x40010c0c |= (1<<5); //配置PB1输出1
    }
}

```

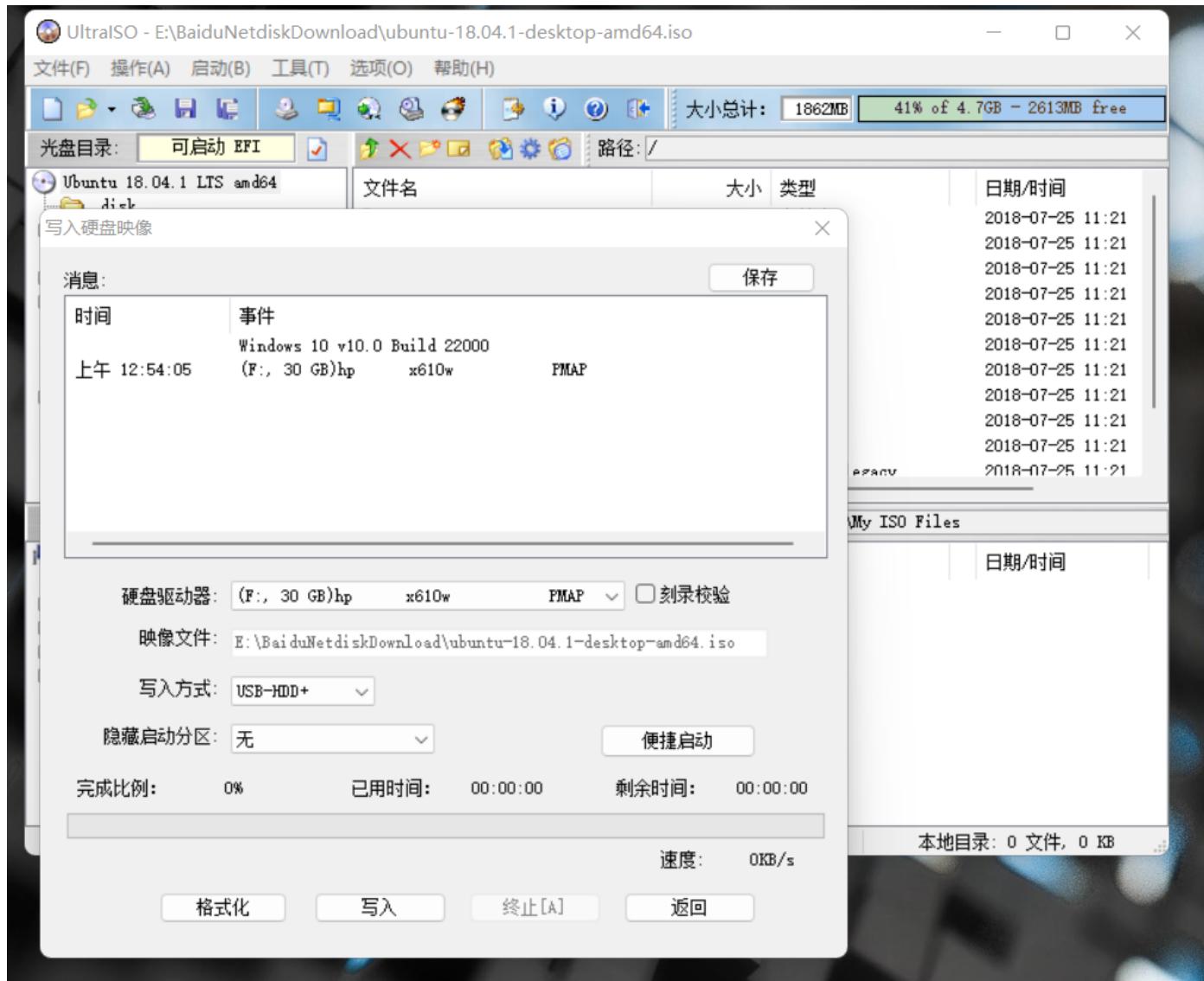
# Ubuntu装机记录

## 需要设备

U盘一个（提前备份数据） [Ubuntu 18.04 LTS 镜像](https://www.ubuntu.com/download/desktop) (<https://www.ubuntu.com/download/desktop>) [UltraISO](https://cn.ultraiso.net/xiazai.html) (<https://cn.ultraiso.net/xiazai.html>)

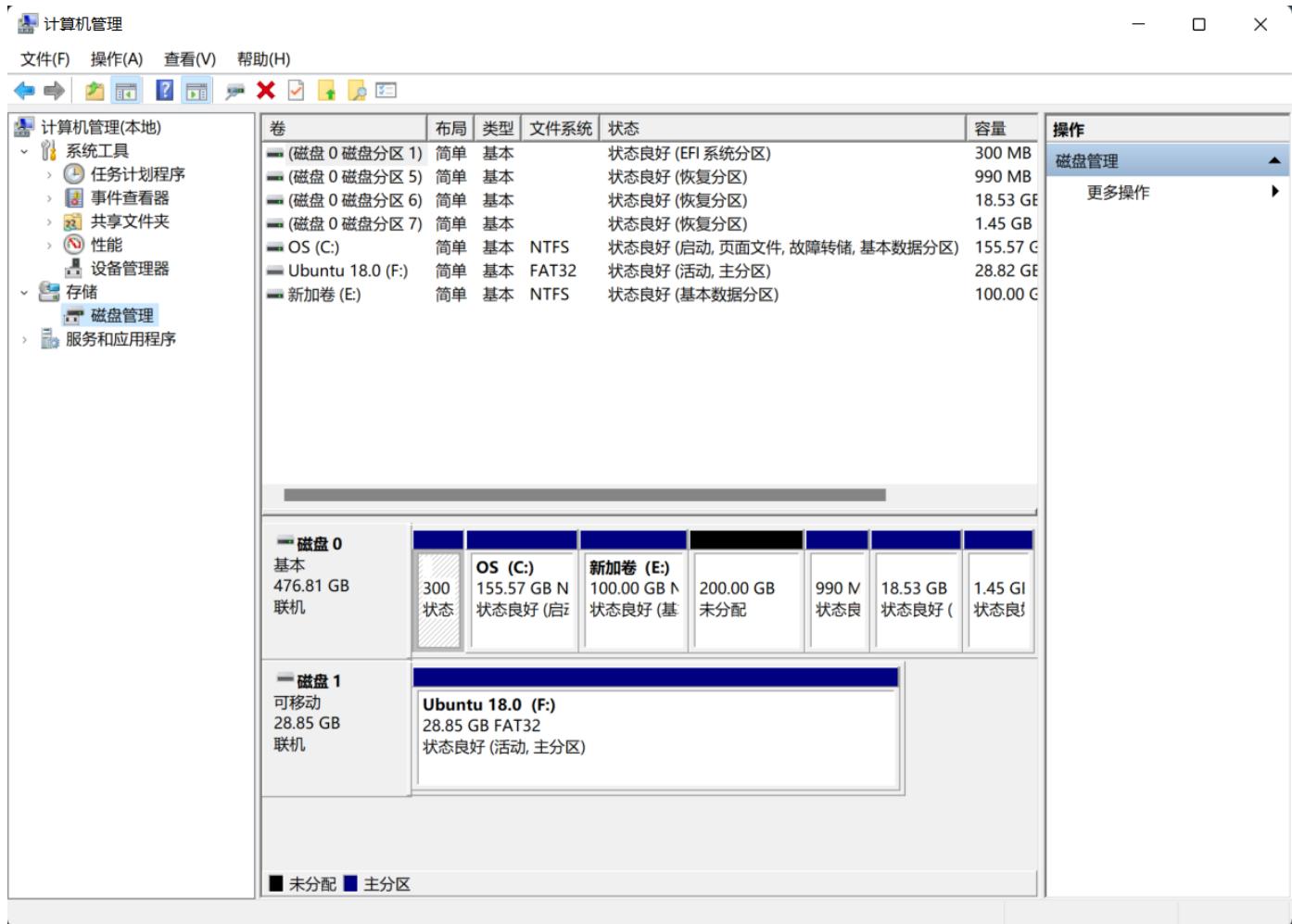
## 制作U盘启动盘

- 1) 安装制作工具UltraISO
- 2) 插入用来做启动盘的U盘（最好是usb3.0接口，16GB或以上），并清空里面的文件
- 3) 打开安装好的UltraISO，点击继续试用按钮工作界面
- 4) 进入工作界面后，点击菜单栏文件(F)，在弹出的选项卡里点击打开
- 5) 在弹出的文件选择对话框中找到下载好的 Ubuntu18.04.1 LTS 镜像文件
- 6) 点击菜单栏上的启动，在弹出的选项卡里点击写入硬盘映像
- 7) 在弹出的新窗口中，硬盘驱动器：选择刚刚插入的U盘，写入方式：设置为USB-HDD+，其余需要勾选的不管，采用默认的设置就行，然后可以格式化一下 最后点击最下面一栏的写入按钮，打开后如下图所示：



## 给 Ubuntu 分配硬盘空间

1) 鼠标右键计算机，在弹出来选项卡中选择管理，接着在弹出来的窗口左侧点击 存储/磁盘管理 2) 在要安装的目标磁盘中，通过删除卷和删除分区操作腾出一块未分配的磁盘空间作为安装区，我要安装的位置是 磁盘1，所以我在 磁盘1 中整合出了 200GB 的空间用来安装 Ubuntu18.04.1），这一步弄好后如下图所示：



## 安装 Ubuntu18.04

1.设置启动项 关闭要安装 Ubuntu18.04.1 的目标主机，然后插入启动盘，接着开机，迅速的按住 F2直到进入 bios 设置界面 进入Boot Manager后，选择 EFI USB 作为启动项，回车 至此就进入了安装程序，选择 Install Ubuntu，回车直接安装，然后根据系统提示安装即可（挂载点放在根目录下）

## 遇到的问题

### 显示无wifi适配器 亮度无法调节

升级linux内核版本 [linux内核官网 \(<https://www.kernel.org/>\)](https://www.kernel.org/) 打开终端，进入刚下载好的文件所在目录下，解压缩刚下载的文件

In [ ]:

```
sudo tar -xavf linux-5.10.90.tar.xz # linux-5.10.90.tar.xz是刚下载的文件名
```

## 安装依赖

In [ ]:

```
sudo apt install build-essential
sudo apt install gcc
sudo apt install make
sudo apt install openssl
sudo apt install pkg-config
sudo apt install libssl-dev
sudo apt install libncurses5-dev
sudo apt install libc6-dev
sudo apt install libelf-dev
sudo apt install bison
sudo apt install flex
```

进入解压后的linux-5.10.90/目录，准备编译：

In [ ]:

```
cd linux-5.10.90/
sudo cp /boot/config-5.4.0-20-generic .config # 在ubuntu的根目录下寻找文件 /boot/config-... 查看一
sudo make menuconfig
```

出现窗口，选择Exit退出（即默认所有选项设置设置），新跳出的窗口选择Yes

### 编译内核

In [ ]:

```
sudo make -j16 # -j16表示同时开16个编译命令, 可以加快编译速度。CPU有多少线程就可以开几个编译命令
sudo make modules_install
```

### 安装

In [ ]:

```
cd ..
sudo mv linux-5.10.90/ /usr/src/
cd /usr/src/linux-5.10.90/
sudo make install
sudo mkinitramfs -o /boot/initrd.img-5.10.90
sudo update-initramfs -c -k 5.10.90
sudo update-grub2
sudo shutdown -r now
uname -a
```

**很无奈 经过了一系列操作，也浏览遍了CSDN上的帖子依然没有拯救我，只能先用网线连着吧，希望去学校后与老师和师兄们探讨能解决这一问题**

## NanoPC-T4学习记录

[NanoPC-T4开发手册](http://wiki.friendlyelec.com/wiki/index.php/NanoPC-T4/zh) ([原理图](http://wiki.friendlyelec.com/wiki/images/e/e0/NanoPC-T4-1902-Schematic.pdf))

(<http://wiki.friendlyelec.com/wiki/images/e/e0/NanoPC-T4-1902-Schematic.pdf>) 百度网盘资料下载地址

(<https://download.friendlyelec.com/NanoPC-T4>) 1)按照NanoPC-T4开发手册擦除和执行烧写即可，默认用户名

pi 密码pi root用户 密码fa (内存卡在刷机后内存空间会被格式占有，如果觉得内存不够可外联固态硬盘，重新分配挂载点)

## # python学习记录

[[python\\_B站学习视频](https://www.bilibili.com/video/BV1wD4y1o7AS?p=2&share_source=copy_pc)] ([https://www.bilibili.com/video/BV1wD4y1o7AS?p=2&share\\_source=copy\\_pc](https://www.bilibili.com/video/BV1wD4y1o7AS?p=2&share_source=copy_pc))

注意事项：

- 1) 应使用小写的Python变量名。虽然在变量名中使用大写字母不会导致错误，但是大写字母在变量名中有特殊含义。
- 2) 慎用小写字母l和大写字母0，因为它们可能被人错看成数字1和0。
- 3) 看到Python代码以普通句子的颜色显示，或者普通句子以Python代码的颜色显示时，就可能意味着文件中存在引号不匹配的情况。
- 4) 方法remove()只删除第一个指定的值。如果要删除的值可能在列表中出现多次，就需要使用循环来确保将每个值都删除。
- 5) 调用函数sorted()后，列表元素的排列顺序并没有变）。如果要按与字母顺序相反的顺序显示列表，也可向函数sorted()传递参数reverse=True。
- 6) Python计算列表元素数时从1开始，因此确定列表长度时，应该不会遇到差一错误。
- 7) 发生索引错误却找不到解决办法时，将列表或其长度打印出来。列表可能以为的截然不同，在程序对其进行动态处理时尤其如此。通过查看列表或其包含的元素数，可帮助找出这种逻辑错误。
- 8) 可在表示切片的方括号内指定第三个值。这个值告诉Python 在指定范围内每隔多少元素提取一个。
- 9) 严格地说，元组是由逗号标识的，圆括号只是让元组看起来更整洁、更清晰。如果你要定义只包含一个元素的元组，必须在这个元素后面加上逗号。创建只包含一个元素的元组通常没有意义，但自动生成的元组有可能只有一个元素。
- 10) 字典中元素的排列顺序与定义时相同。如果将字典打印出来或遍历其元素，将发现元素的排列顺序与添加顺序相同。
- 11) 调用get()时，如果没有指定第二个参数且指定的键不存在，Python将返回值None。这个特殊值表示没有相应的值。None 并非错误，而是一个表示所需值不存在的特殊值。
- 12) 使用关键字实参时，务必准确指定函数定义中的形参名。