

# K210开发板学习

## 按照官方文档进行烧录

1. 主要参考的视频+官方文档:

<https://www.bilibili.com/video/BV144411J72P/>

<https://wiki.sipeed.com/soft/maixpy/zh/index.html>

2. 安装CH340驱动:

<https://blog.csdn.net/JAZZSOLDIER/article/details/70169732>

(Linux 不需要装驱动, 系统自带了, 使用 `ls /dev/ttyUSB*` 即可看到设备号)

然后在设备管理器的端口处可以看到安装完成:



3. 下载, 解压, 运行升级工具

[https://github.com/sipeed/kflash\\_gui/releases/tag/v1.6.7](https://github.com/sipeed/kflash_gui/releases/tag/v1.6.7)



#### 4. 固件下载

[https://dl.sipeed.com/shareURL/MAIX/MaixPy/release/master/maixpy\\_v0.6.2\\_72\\_g22a8555b5](https://dl.sipeed.com/shareURL/MAIX/MaixPy/release/master/maixpy_v0.6.2_72_g22a8555b5)

下maixpy\_\*.bin这个固件

#### 5. 固件烧录

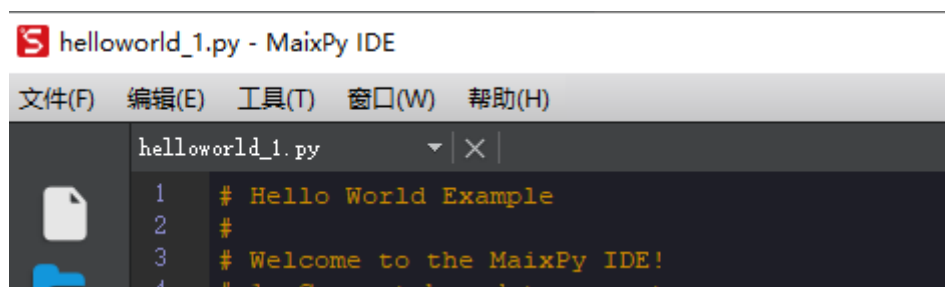
打开Kflash，选择刚刚下的bin文件，选择开发板为Dock，端口是CH340所在的端口



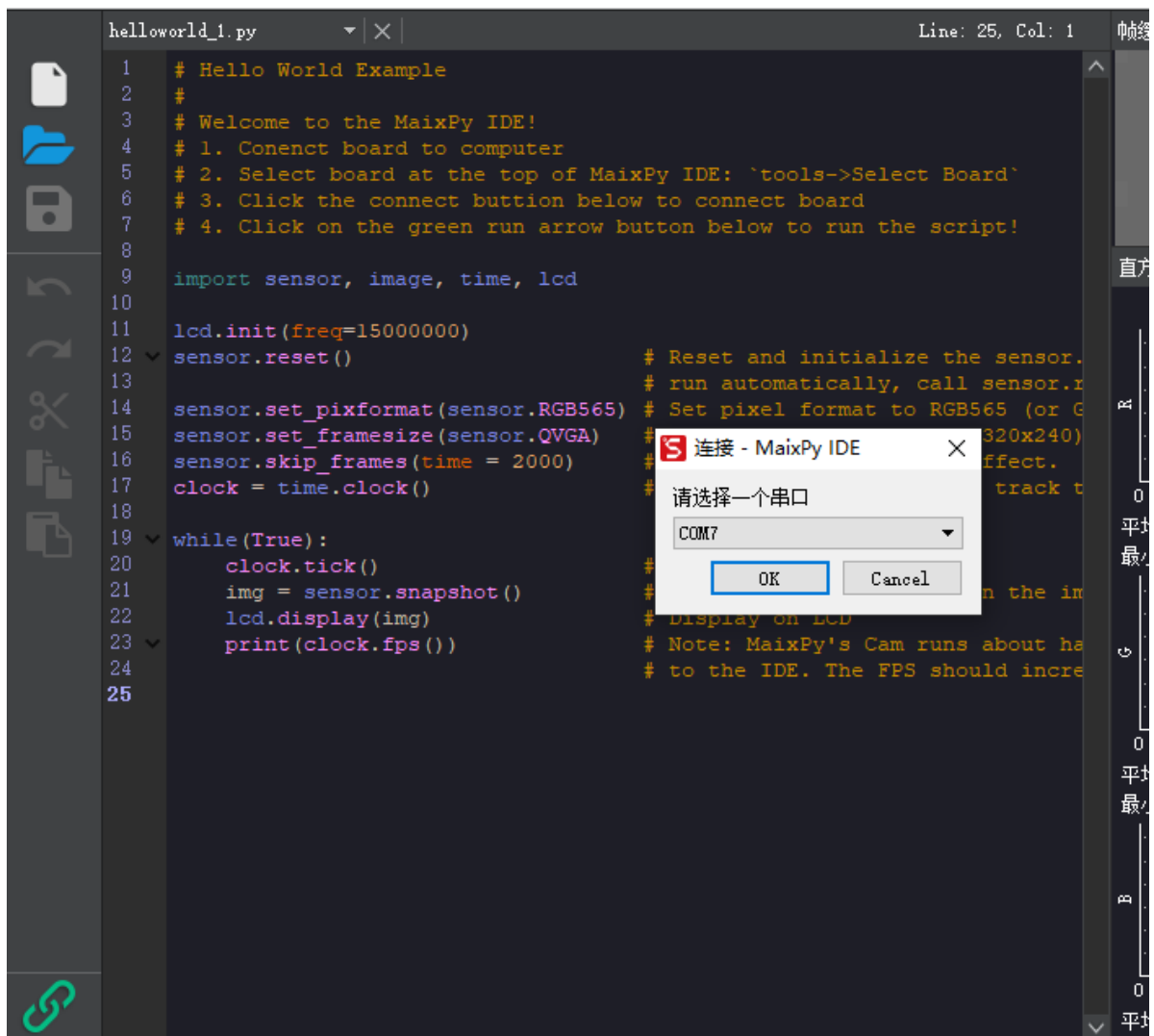
## 6. 下载IDE

<https://dl.sipeed.com/MAIX/MaixPy/ide/>

## 7. 在IDE上连接主板，运行程序



在工具这一栏选择开发板为Dock



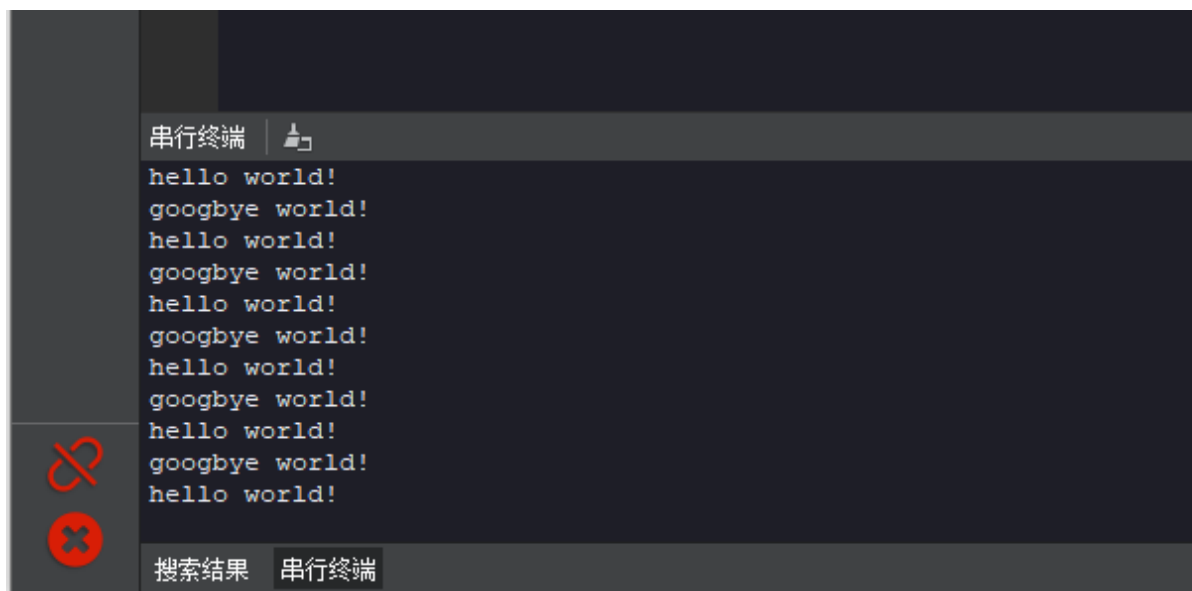
点左下角的绿色回形针，记得选串口为刚刚下载的CH340 driver的串口。

#### 8. Hello world

连接完成后，我们跑下面这一段python代码：

```
while(True):  
    print("hello world!")  
    print("googbye world!")
```

点击左下角运行，并点开串行终端，可以看到结果被正确运行。



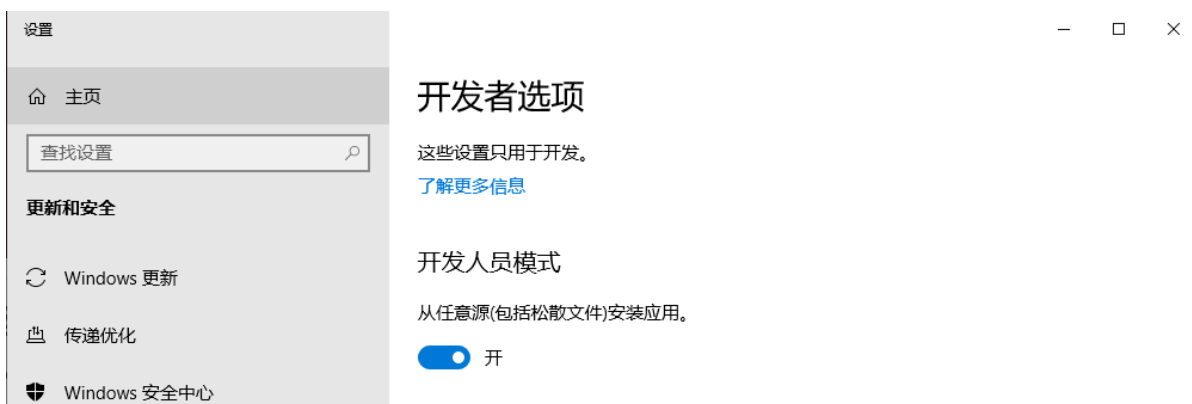
## 编译，烧录开源kernel

这一步的目标是对上一届的一等奖作品xv6-k210(<https://github.com/SKTT1Ryze/xv6-k210>)进行编译。

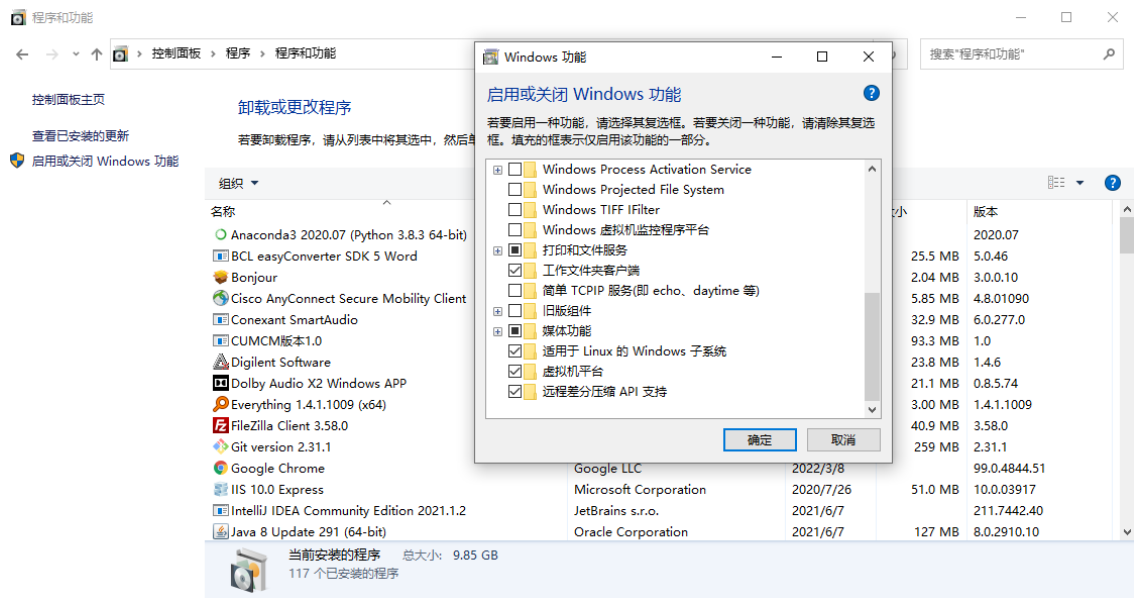
其中，它用到了 `riscv64-unknown-elf` 这个编译工具，这个编译工具是在linux环境下运行的，所以我的第一部是在windows电脑下安装Ubuntu 20.04子系统（WSL）：

### 安装WSL

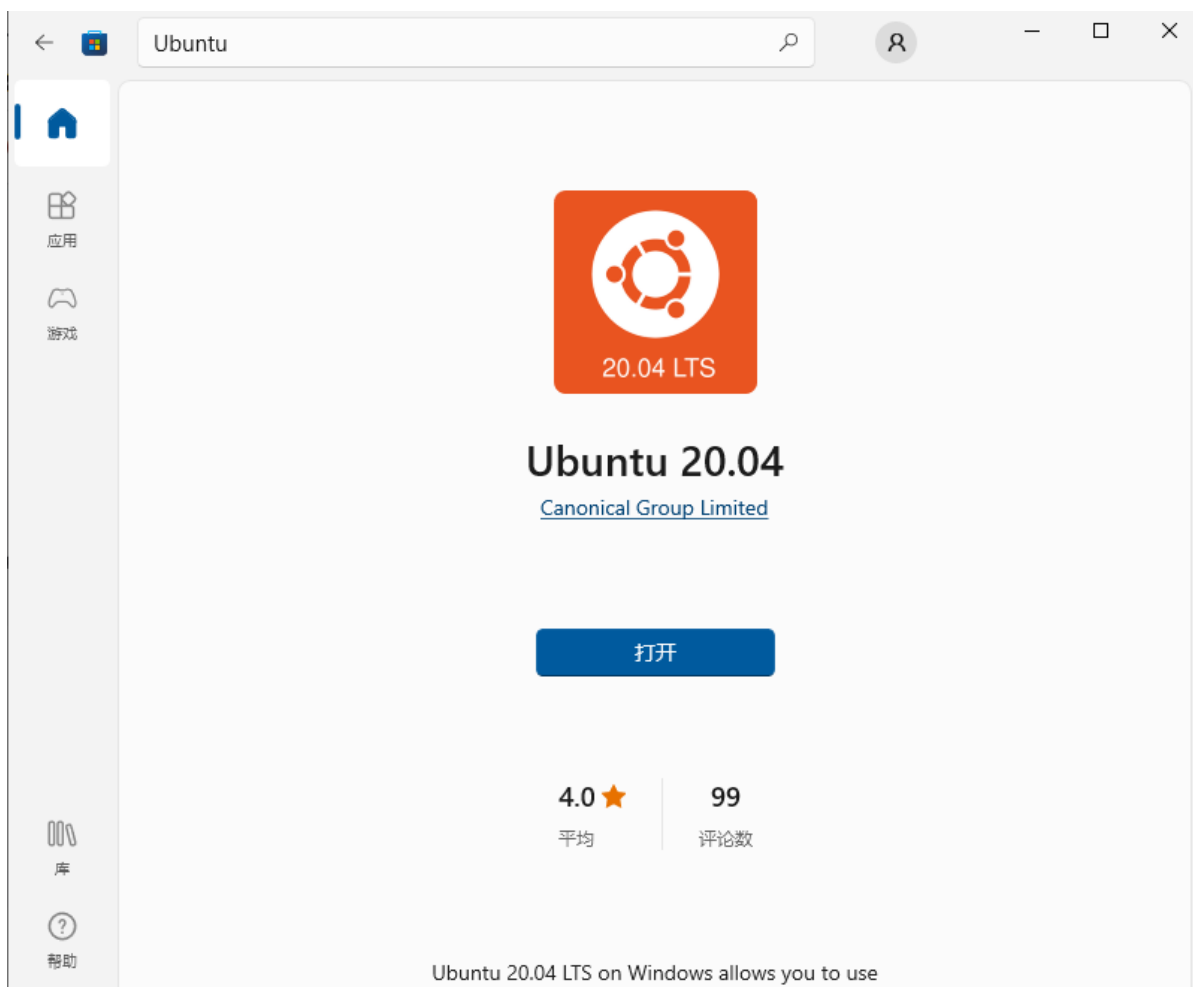
1. 设置 - 开发人员模式 - 开



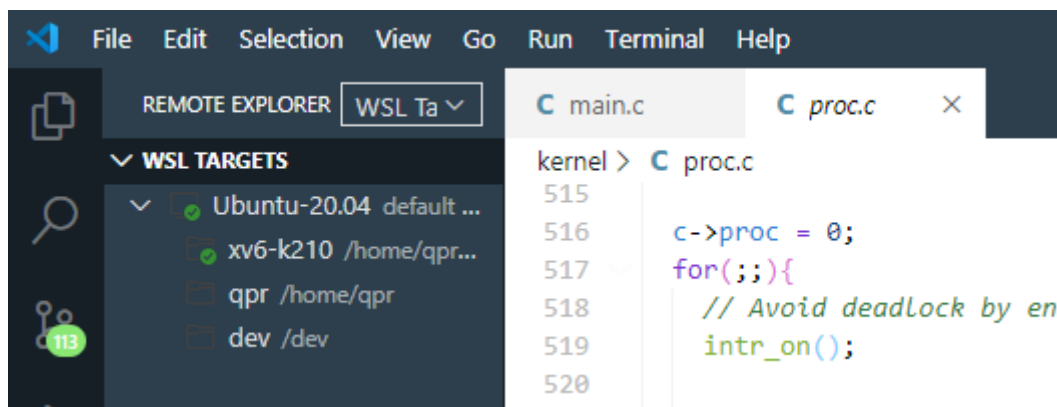
2. 控制面板-程序-程序功能-启动或关闭Windows功能 - 把“适用于Linux的Windows子系统选上”



3. 在应用商店里下载Ubuntu,



4. 下载完后设置账户和密码后, 可以用VSC连上WSL: Remote-WLS-Connect



5. WSL换源，不然安装太慢了

换源可以参考这一篇[https://blog.csdn.net/qg\\_42476834/article/details/120939510](https://blog.csdn.net/qg_42476834/article/details/120939510)

6. 然后把make, gcc那些工具都下好

7. Ubuntu 20.04的好处是它可以一行命令把 `riscv64-unknown-elf` 下好

```
sudo apt-get install gcc-riscv64-unknown-elf
```

省非常多的事!

8. 把kernel给clone下来;

```
git clone https://github.com/SKTT1Ryze/xv6-k210
```

9. 在编译前，我们需要根据我们当下系统改一些东西：

- 设备串口：

在Makefile中有这样的一段话：

```
image = $T/kernel.bin
k210 = $T/k210.bin
k210-serialport := /dev/ttyUSB0
```

然而我们的设备可能并不是ttyUSB0。用 `ls /dev` 查看设备：

```
qpr@DESKTOP-8SGMT82:~/xv6-k210$ ls /dev
block  ttyS109  ttyS121  ttyS134  ttyS147  ttyS160  ttyS172  ttyS185  ttyS25  ttyS38  ttyS50  ttyS63  ttyS76  ttyS89
brg    ttyS11  ttyS122  ttyS135  ttyS148  ttyS161  ttyS173  ttyS186  ttyS26  ttyS39  ttyS51  ttyS64  ttyS77  ttyS90
fd     ttyS0   ttyS110  ttyS123  ttyS136  ttyS149  ttyS162  ttyS174  ttyS187  ttyS27  ttyS40  ttyS52  ttyS65  ttyS78  ttyS91
lansg  ttyS1   ttyS111  ttyS124  ttyS137  ttyS150  ttyS163  ttyS175  ttyS188  ttyS28  ttyS41  ttyS53  ttyS66  ttyS79  ttyS92
lss    ttyS10  ttyS112  ttyS125  ttyS138  ttyS151  ttyS164  ttyS176  ttyS189  ttyS29  ttyS42  ttyS54  ttyS67  ttyS80  ttyS93
null   ttyS100  ttyS113  ttyS126  ttyS139  ttyS152  ttyS165  ttyS177  ttyS190  ttyS30  ttyS43  ttyS55  ttyS68  ttyS81  ttyS94
ptmx   ttyS101  ttyS114  ttyS127  ttyS140  ttyS153  ttyS166  ttyS178  ttyS191  ttyS31  ttyS44  ttyS56  ttyS69  ttyS82  ttyS95
pts    ttyS102  ttyS115  ttyS128  ttyS141  ttyS154  ttyS167  ttyS179  ttyS192  ttyS32  ttyS45  ttyS57  ttyS70  ttyS83  ttyS96
random ttyS103  ttyS116  ttyS129  ttyS142  ttyS155  ttyS168  ttyS180  ttyS20  ttyS33  ttyS46  ttyS58  ttyS71  ttyS84  ttyS97
shm    ttyS104  ttyS117  ttyS130  ttyS143  ttyS156  ttyS169  ttyS181  ttyS21  ttyS34  ttyS47  ttyS59  ttyS72  ttyS85  ttyS98
stderr ttyS105  ttyS118  ttyS131  ttyS144  ttyS157  ttyS170  ttyS182  ttyS22  ttyS35  ttyS48  ttyS60  ttyS73  ttyS86  ttyS99
stdin  ttyS106  ttyS119  ttyS132  ttyS145  ttyS158  ttyS171  ttyS183  ttyS23  ttyS36  ttyS49  ttyS61  ttyS74  ttyS87  urandom
stdout ttyS107  ttyS120  ttyS133  ttyS146  ttyS159  ttyS172  ttyS184  ttyS24  ttyS37  ttyS50  ttyS62  ttyS75  ttyS88  zero
tty    ttyS108  ttyS121  ttyS134  ttyS147  ttyS160  ttyS173  ttyS186  ttyS26  ttyS39  ttyS51  ttyS64  ttyS77  ttyS90
```

果然不是。这里WSL和原windows系统的串口对应规则是：COM x 就对应ttySx。在第一部分我们就知道我们k210设备连的是COM7，对应的是ttyS7，所以我们将Makefile那段话改成：

```
image = $T/kernel.bin
k210 = $T/k210.bin
k210-serialport := /dev/ttyS7
```

- (暂时)把SD卡读写功能去掉





