启动开发板，复制server程序到开发板。

然后运行程序前需要先导入程序运行时所需的动态库：将libgcc\_s.so.1 libstdc++.so.6.0.27复制到/usr/lib

之后再创建软链接：

ln -s /usr/lib/libstdc++.so.6.0.27 /usr/lib/libstdc++.so.6

再加载设备驱动程序（运行/home/root/cis\_driver/register.sh），最后运行server。

新的ip地址

[http://192.168.1.115/](http://192.168.1.108/)



scp white.bmp black.bmp white\_b.bmp black\_b.bmp root@172.31.225.201:/home/root/cis\_app

诶，可以一个生成server1,一个生成server的呀，这样子很方便的比较了

scp server301 [root@172.31.225.131:/home/root/cis\_app](mailto:root@172.31.225.131:/home/root/cis_app)

scp scanner.ko [root@192.168.1.56:/home/root/cis\_d](mailto:root@172.31.225.131:/home/root/cis_app)river

scp black.bmp [root@172.31.225.131:/home/root/cis\_app](mailto:root@172.31.225.131:/home/root/cis_app)

scp server991 root@192.168.1.56:/home/root

传输命令

先配置一下环境先

Sptl

echo 'export PATH=$PATH:'`which arm-linux-gnueabihf-gcc | xargs dirname` >> ~/.bashrc

配置好这些就可以运行了

最新版的编译命令，先打开到board.cpp所在文件夹

arm-linux-gnueabihf-g++ -o server991 -DCPPHTTPLIB\_THREAD\_POOL\_COUNT=16 board.cpp bit.c -I. -lpthread -g -Wall

多向量运算的编译指令：

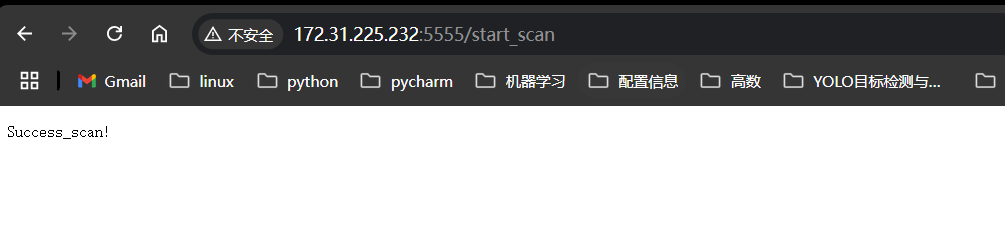
arm-linux-gnueabihf-g++ -o server991 -DCPPHTTPLIB\_THREAD\_POOL\_COUNT=16 board.cpp bit.c -I. -lpthread -g -Wall -mfpu=neon

目标IP假设为 **172.31.225.131**，端口号设置为**5555**，在浏览器上面的操作流程如下

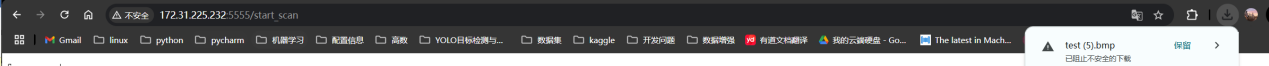
**直接在浏览器网站处输入即可**

启动扫描的http命令为

[http://172.31.225.131:5555/start\_scan](http://172.31.225.232:5555/start_scan)

获取对应bmp的命令为，此时会弹出下载提示

[http://172.31.225.131:5555/get\_bmp](http://172.31.225.232:5555/get_bmp)



设置间隔事件的命令为： 假如为100 (注意：设置的范围必须>=2)

<http://localhost:5555/get_data?number=100>

翻转扫描方向的命令为：

[http://172.31.225.131:5555/img\_flip](http://172.31.225.232:5555/img_flip)

start\_scan命令会将扫描后得到的原始图像进行一次简单的颜色矫正（这个目前是做全白的颜色矫正，现在还缺一个全黑的颜色矫正哇，其他全红，全绿，全蓝的要不要搞哇，需要重新生成.serve文件哇），该过程会使用到cis\_app/目录下的white.bmp文件。

若不需要进行颜色矫正，即想得到直接扫描的结果，则使用下面的命令：

扫描：

[http://172.31.225.131:5555/get\_img](http://172.31.225.232:5555/get_img)

获取对应的bmp：

[http://172.31.225.131:5555/get\_origin\_bmp](http://172.31.225.232:5555/get_origin_bmp)

<http://192.168.1.56:5555/get_origin_bmp>

<http://192.168.1.56:5555/chose_change_1>

[http://192.168.1.56:5555/chose\_change\_0](http://192.168.1.56:5555/chose_change_1)

重复两次这个指令才能切换输入的值哦

若想重新扫描并更换用于颜色矫正的white.bmp文件，输入：

[http://172.31.225.131:5555/get\_white\_img](http://172.31.225.232:5555/get_white_img)

此时会进行一次扫描，并将扫描结果作为新的white.bmp替换掉原来的文件。

想查看white.bmp文件，可以通过以下命令获取：

[http://172.31.225.131:5555/get\_white\_bmp](http://172.31.225.232:5555/get_white_bmp)

[http://172.31.225.131:5555/get\_black\_img](http://172.31.225.232:5555/get_white_img)

此时会进行一次扫描，并将扫描结果作为新的black.bmp替换掉原来的文件。

想查看black.bmp文件，可以通过以下命令获取：

[http://172.31.225.131:5555/get\_black\_bmp](http://172.31.225.232:5555/get_white_bmp)

新的white.bmp,和新的black.bmp要生效需要更新内部参数，所以需要执行：

[http://172.31.225.131:5555/update\_weight](http://172.31.225.232:5555/update_weight)

加入选择命令的

[http://172.31.225.131:5555/](http://172.31.225.232:5555/update_weight)chose\_a

这个代表控制扫描仪a的

[http://172.31.225.131:5555/](http://172.31.225.232:5555/update_weight)chose\_b

这个代表控制扫描仪b的

注：这二者已经更新黑白校正的参数了

<http://192.168.1.108/>

新的IP地址更换一下就行

想查看扫描仪B的white\_b.bmp文件，可以通过以下命令获取：

[http://172.31.225.131:5555/get\_white\_b\_bmp](http://172.31.225.232:5555/get_white_bmp)

想查看扫描仪B的black\_b.bmp文件，可以通过以下命令获取：

[http://172.31.225.131:5555/get\_black\_b\_bmp](http://172.31.225.232:5555/get_white_bmp)

curl <http://192.168.1.108:5555/chose_a>

[http://192.168.1.108:5555/get\_img](http://192.168.1.108:5555/chose_a)

[http://192.168.1.108:5555/get\_origin\_bmp](http://192.168.1.108:5555/chose_a)

最新增加的一些功能哇！！！！！！！！！！！

<http://192.168.1.56:5555/get_origin_bmp>

[http://192.168.1.56:5555/get\_b\_bmp](http://172.31.225.131:5555/get_origin_b_bmp)

这些是获取b的哇

[http://192.168.1.56:5555/get\_white\_a\_img](http://172.31.225.131:5555/get_white_a_img)

这个是只扫a的白图

[http://192.168.1.56:5555/get\_white\_](http://172.31.225.131:5555/get_white_a_img)bmp

这个是拿到a的白图

[http://192.168.1.56:5555/get\_black\_](http://172.31.225.131:5555/get_white_a_img)bmp

这个是拿到a的黑图

[http://192.168.1.56:5555/get\_white\_b\_img](http://172.31.225.131:5555/get_white_a_img)

这个是只扫b的白图

[http://192.168.1.56:5555/get\_white\_both\_img](http://172.31.225.131:5555/get_white_a_img)

这个是两个都取的

（改动说明：白图，黑图的获取改了逻辑了，为了方便，特意设置了三种情况的，只要a的，只要b的，都要的，注：这三种情况，两个都会发光，但是前两个只会生成自己那一路的参考照片，第三种是都生成的 然后那个update\_weight，正常用就行，我加了command\_save去追踪情况逻辑了，会根据当前的情况来选择跟新什么参数的，不放心的话，可以调一下command的指令哦）

然后那个get\_img两个都会生成的，生成两个origin,origin\_b的图片的哦

Start\_scan也是的，处理也是会一起处理两个照片的，得到两个res的照片，所以会慢一点的

建议先尝试一下get\_img的吧

[http://172.31.225.131:5555/](http://172.31.225.232:5555/update_weight)get\_black\_offset

[http://172.31.225.131:5555/chose\_the\_same](http://172.31.225.232:5555/update_weight)