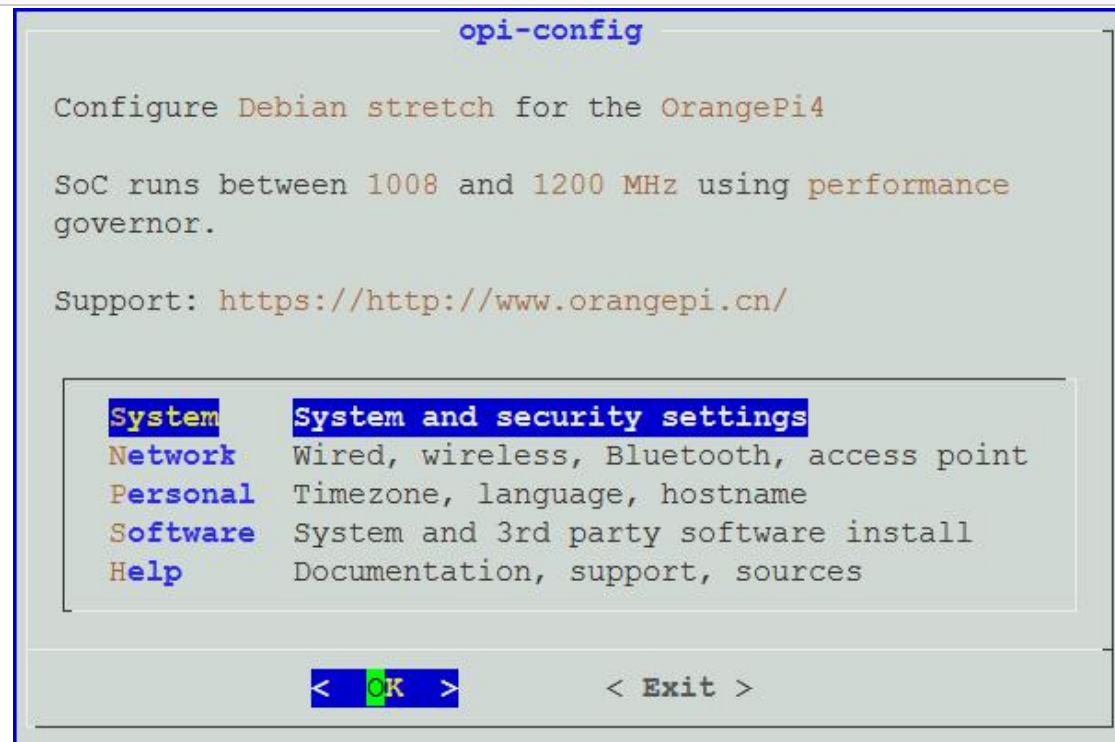


一、使用 **opi-config** 配置系统

终端 执行命令

```
opi-config
```



如果没有此命令按照以下方式安装

```
apt-get install -y dialog expect bc cpufrequtils figlet toilet  
lsb-release
```

```
git clone https://github.com/baiywt/opi_config_libs.git  
cp opi_config_libs /usr/local/sbin/ -rfa  
cp opi_config_libs/opi-config /usr/local/sbin -f  
rm -rf /etc/update-motd.d/*  
cp opi_config_libs/overlay/* / -rfa
```

二、使用 python 控制 GPIO

1、下载 OPi.GPIO

```
git clone https://github.com/baiywt/OPi.GPIO.git
```

2、安装

```
apt install python3-pip  
cd OPi.GPIO  
python3 setup.py install
```

3、测试

新建文件 test_gpio.py

```
import time  
import sys  
  
import OPi.GPIO as GPIO  
  
import orangepi.pi4  
BOARD = orangepi.pi4.BOARD  
GPIO.setmode(BOARD)  
  
pin=int(sys.argv[1])  
  
GPIO.setup(pin, GPIO.OUT)  
while True:  
    GPIO.output(pin, GPIO.HIGH)  
    time.sleep(1)  
    GPIO.output(pin, GPIO.LOW)  
    time.sleep(1)  
cd ..
```

```
python3 test_gpio.py 16
```

//控制物理引脚 16 1 秒一次翻转电平

三、安装中文输入法

Pi4 ubuntu18.04 安装中文输入法

测试镜像：OrangePi_4_ubuntu_bionic_desktop_linux4.4.179_v1.1.img

1、安装 ibus 输入法框架

在终端执行如下命令

```
apt-get install ibus ibus-clutter ibus-gtk ibus-gtk3 ibus-qt4
```

2、安装 ibus-pinyin

在终端执行如下命令

```
apt-get install ibus-pinyin ibus-sunpinyin
```

3、输入法的配置

点击左下角图标，选择 IBus Preferences



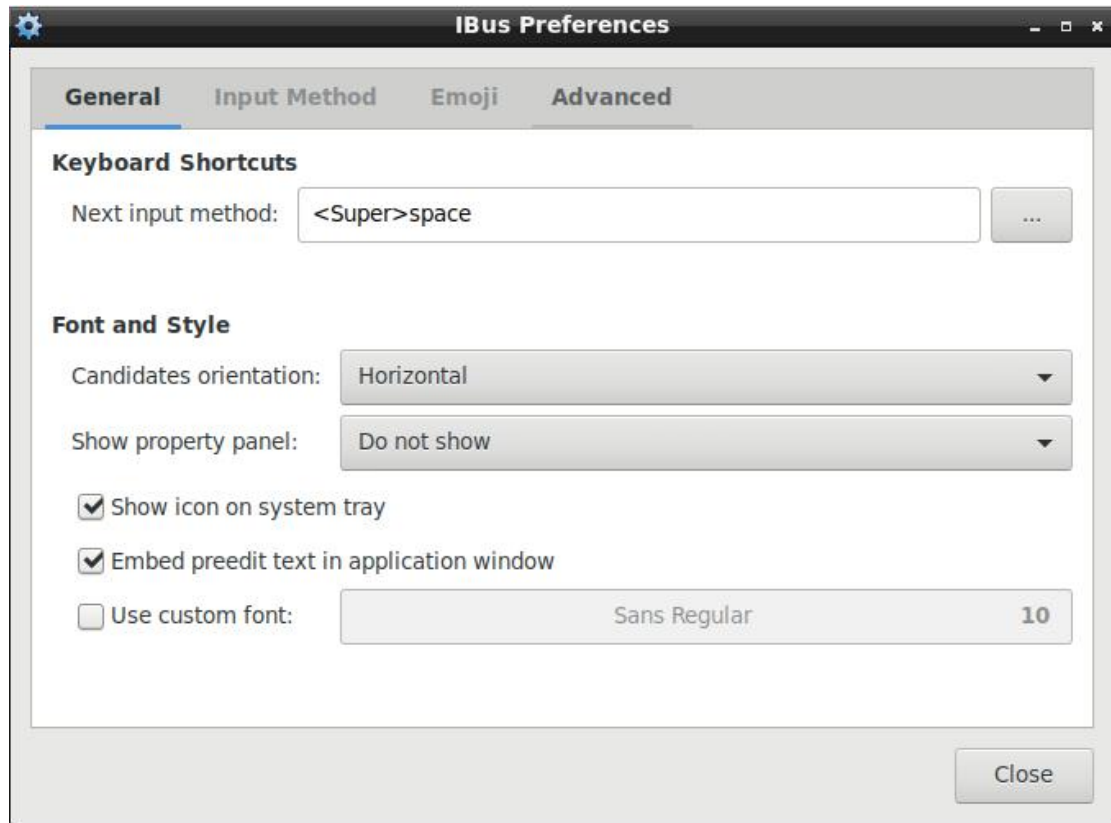
在弹出的窗口中选择 yes



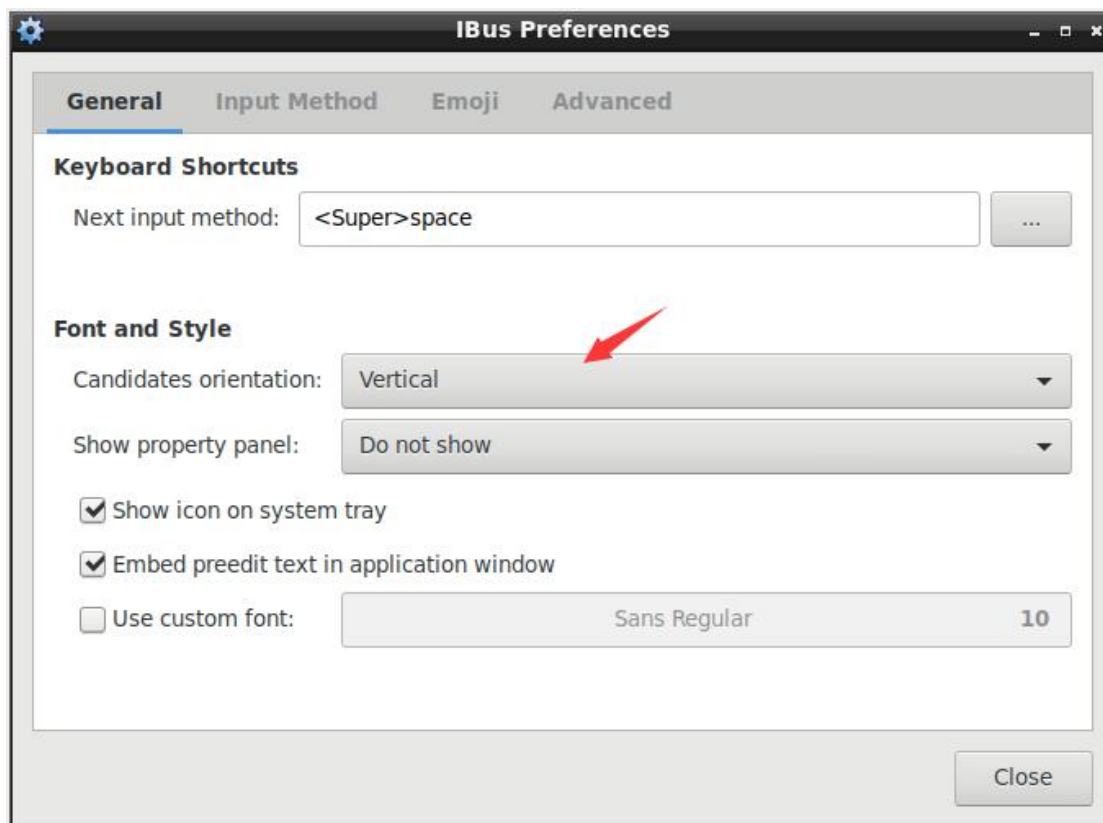
选择 ok



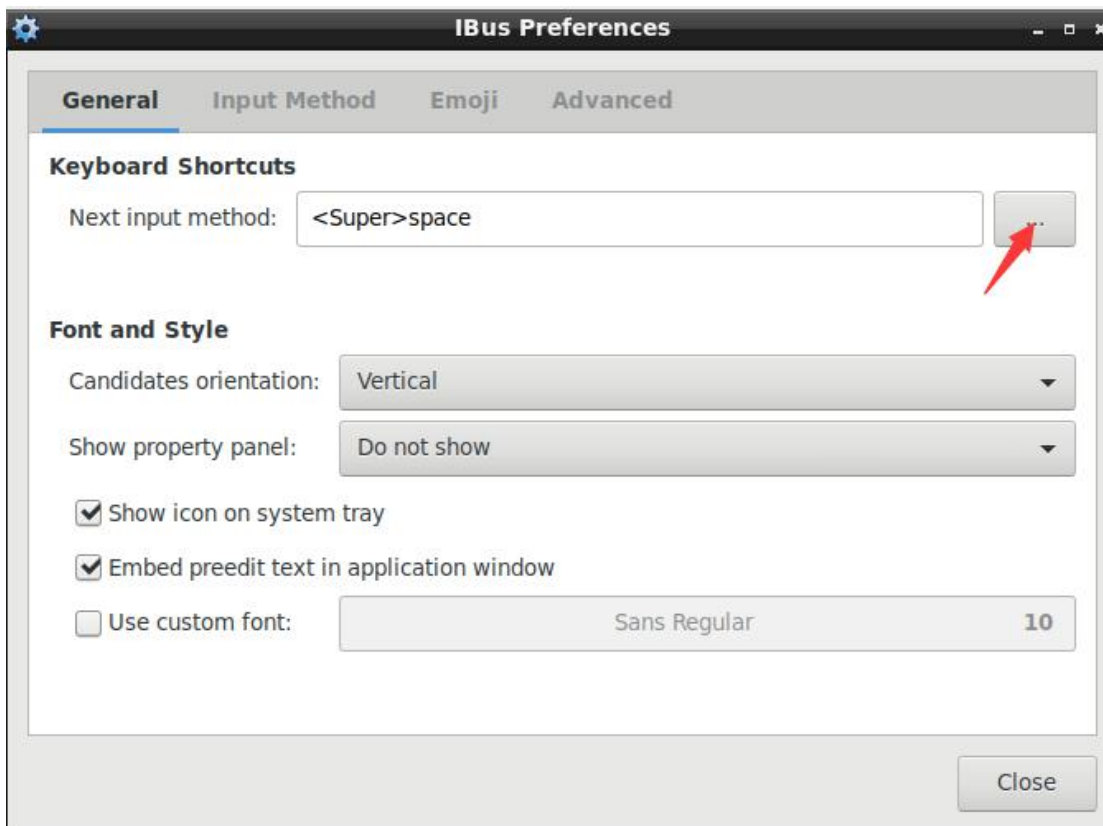
弹出配置界面



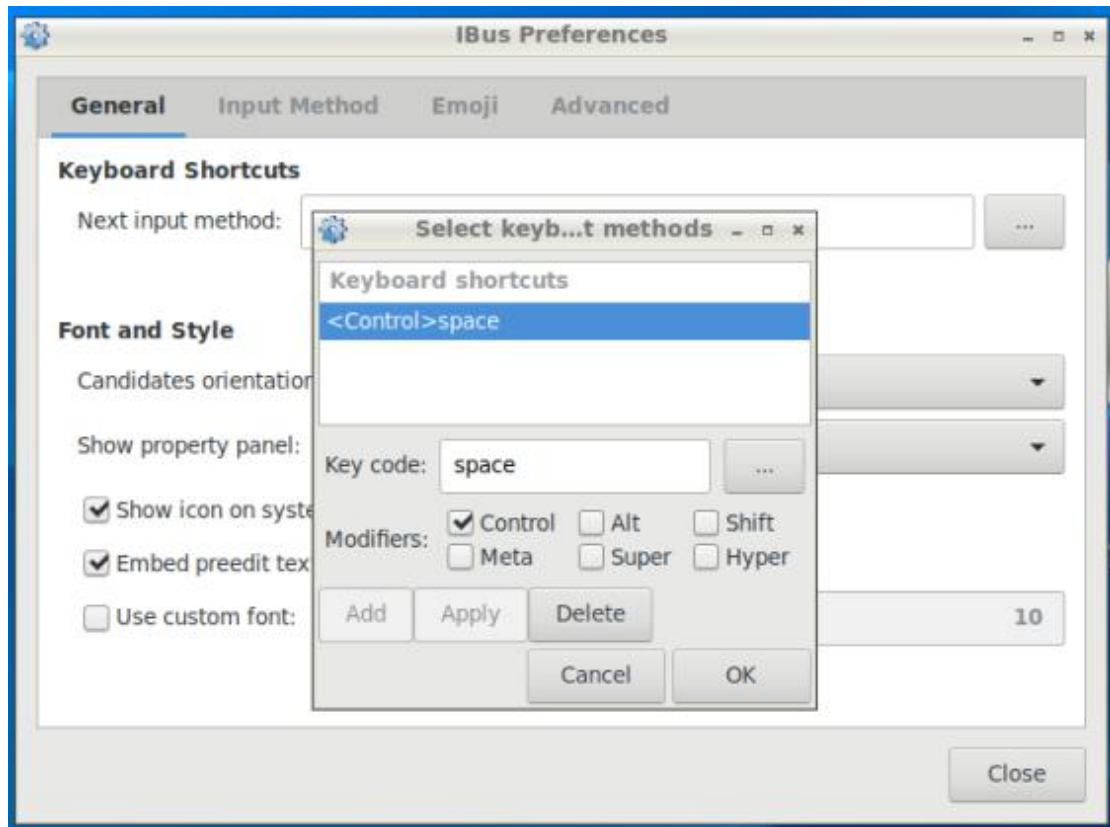
输入法方向选择 Vertical



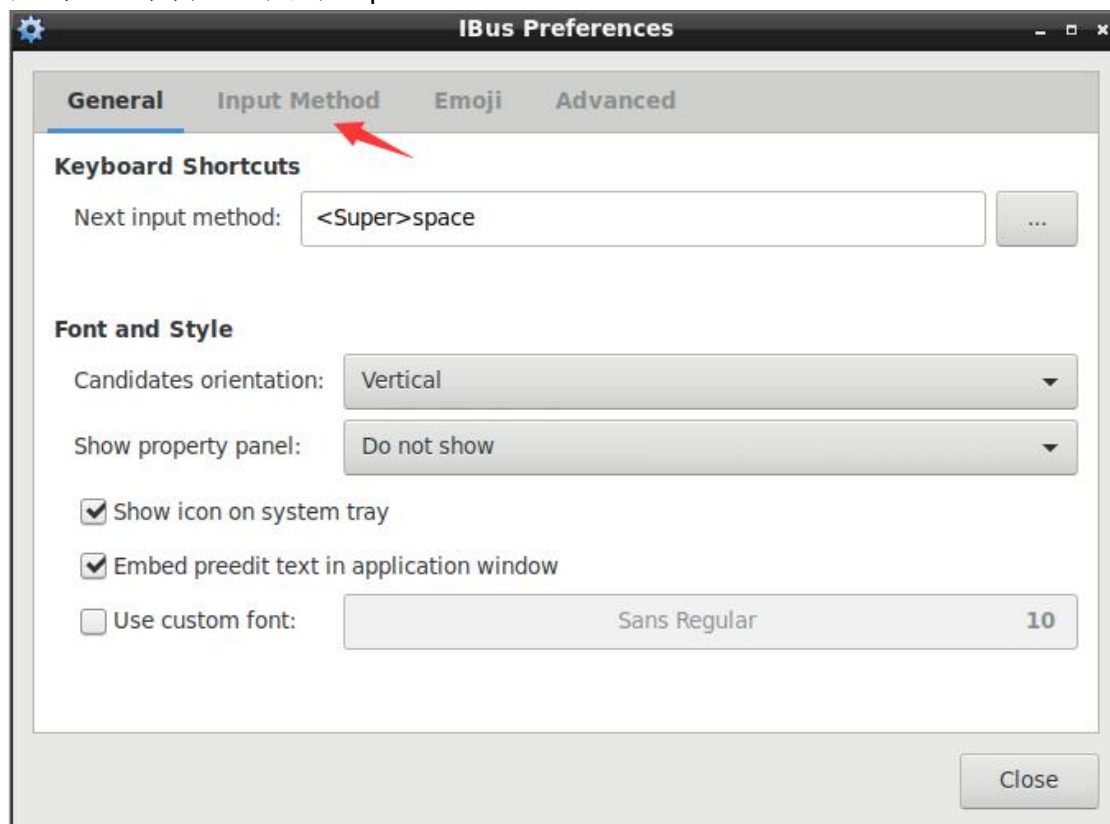
点击下图的按钮选择切换快捷键



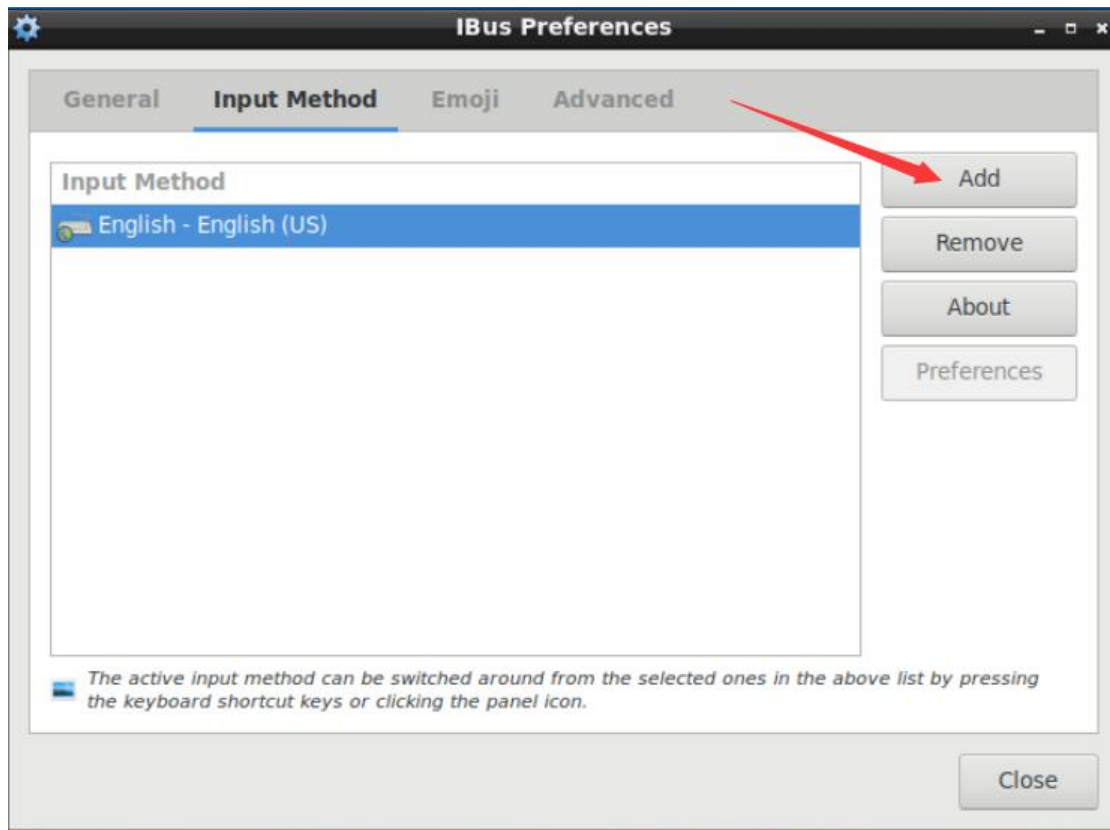
在弹出的窗口中勾选 Control，去掉 Super 前面的勾，然后点击 Apply。再点击 ok。



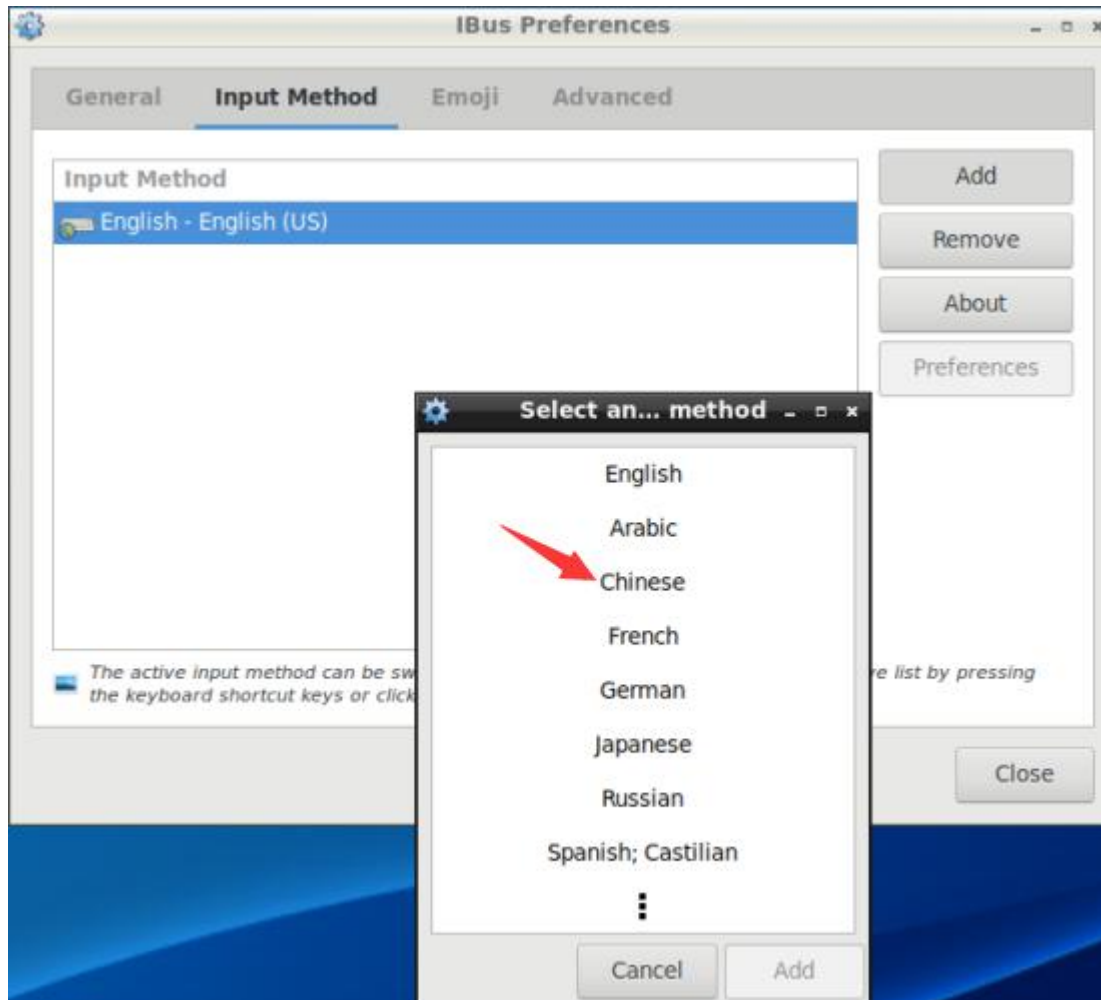
回到上一个窗口，点击 Input Method。



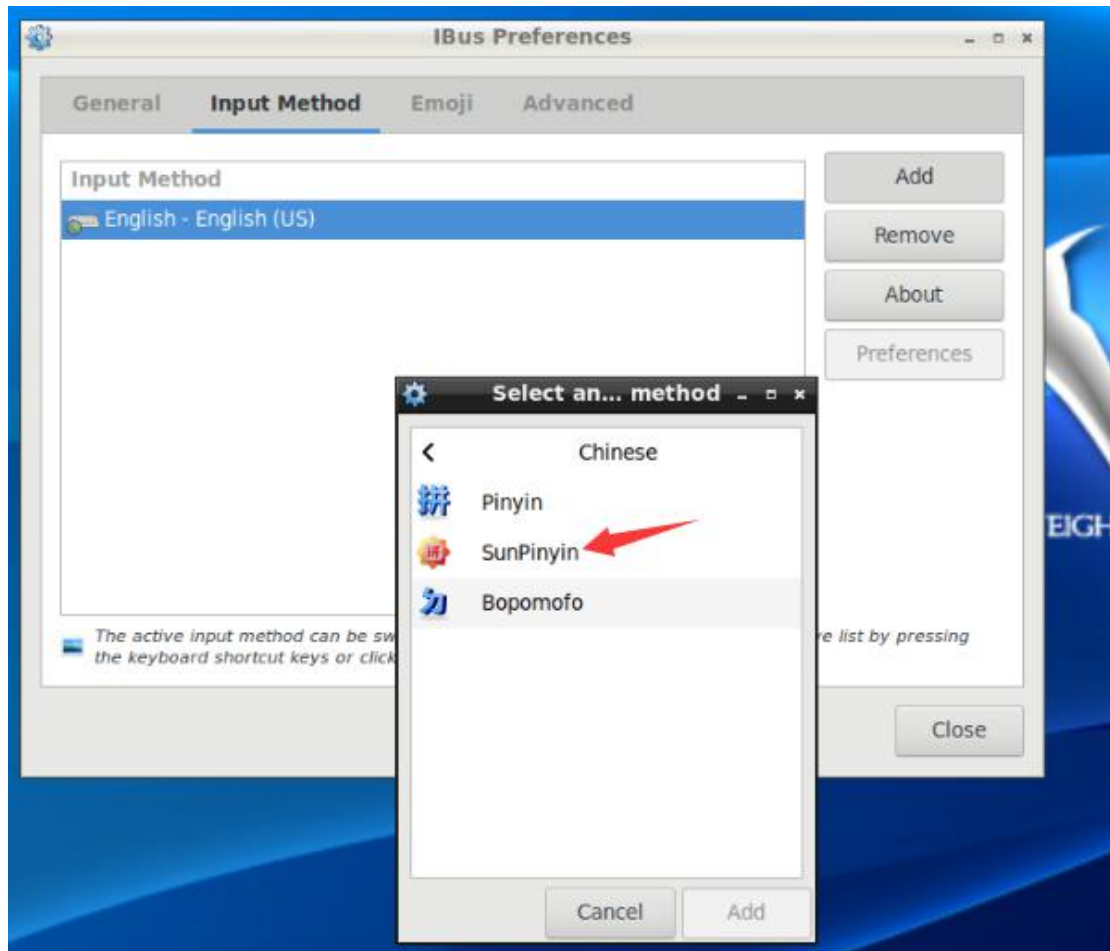
如下图，只有英文。点击 Add 添加中文输入法。



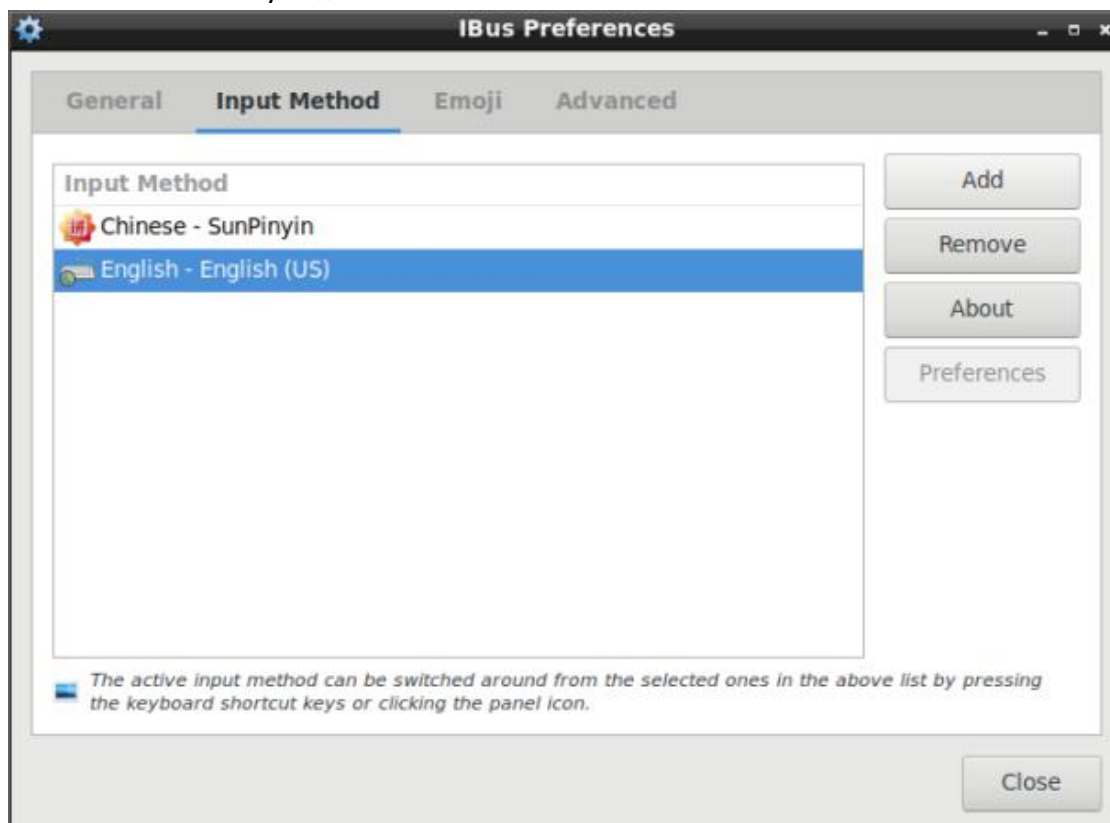
选择 Chinese



选择 Sunpinyin, 点击 Add



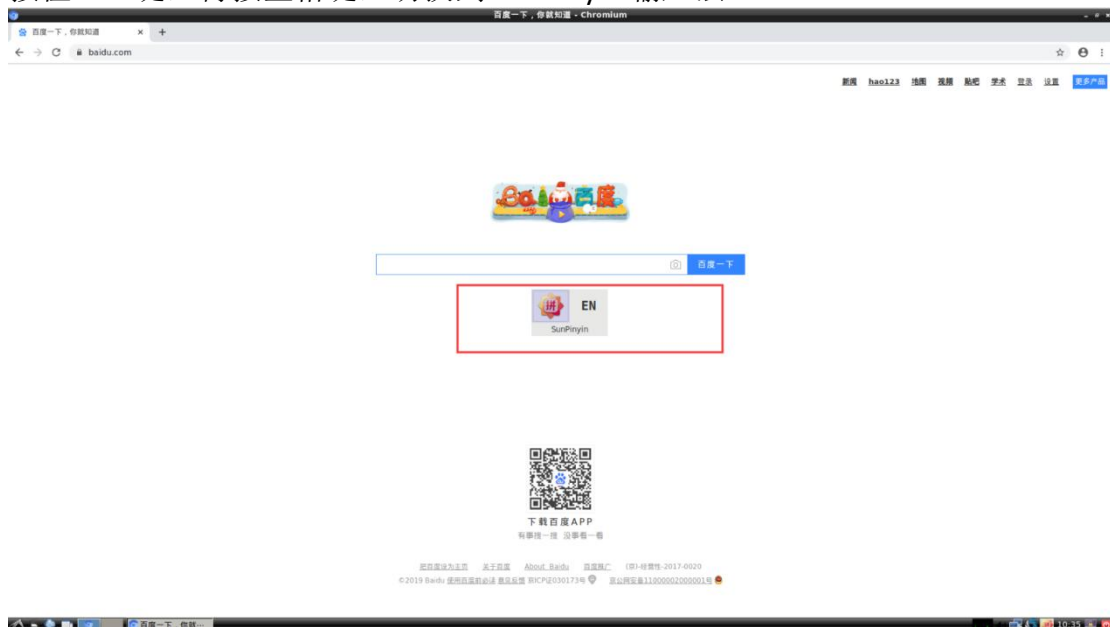
成功添加了 SunPinyin 输入法

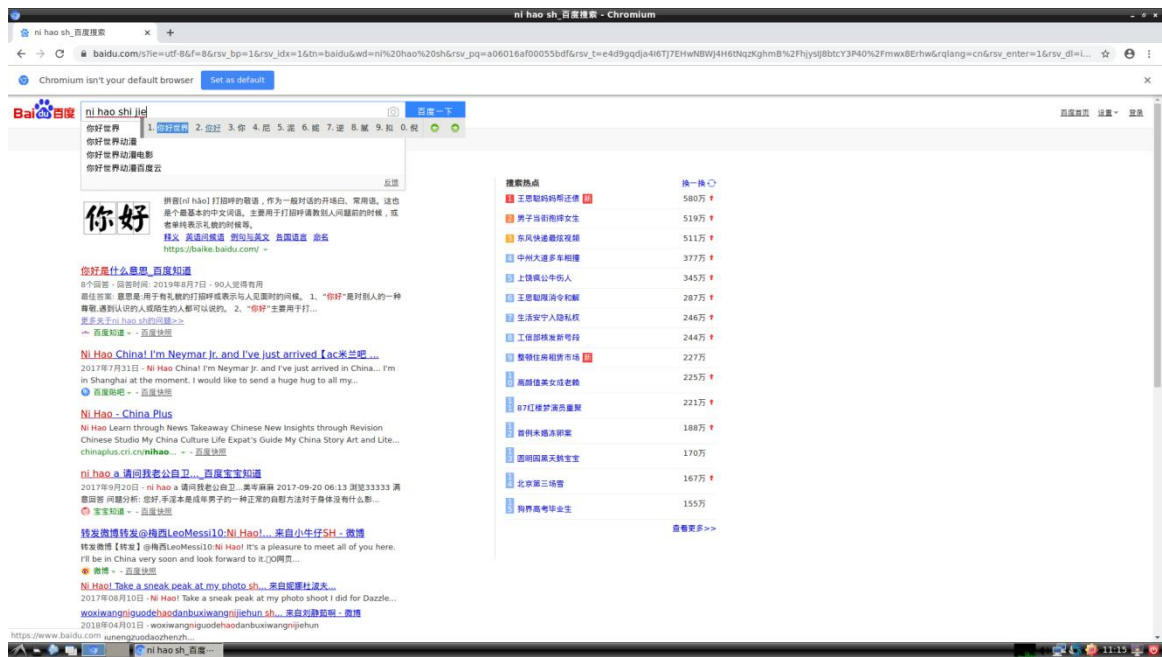


打开浏览器



按住 Ctrl 键，再按空格键，切换到 SunPinyin 输入法





四、安装 QT 及其使用示例

1、安装 QT5

```
sudo apt-get update
sudo apt-get install qt5-default
sudo apt-get install qtcreator
```

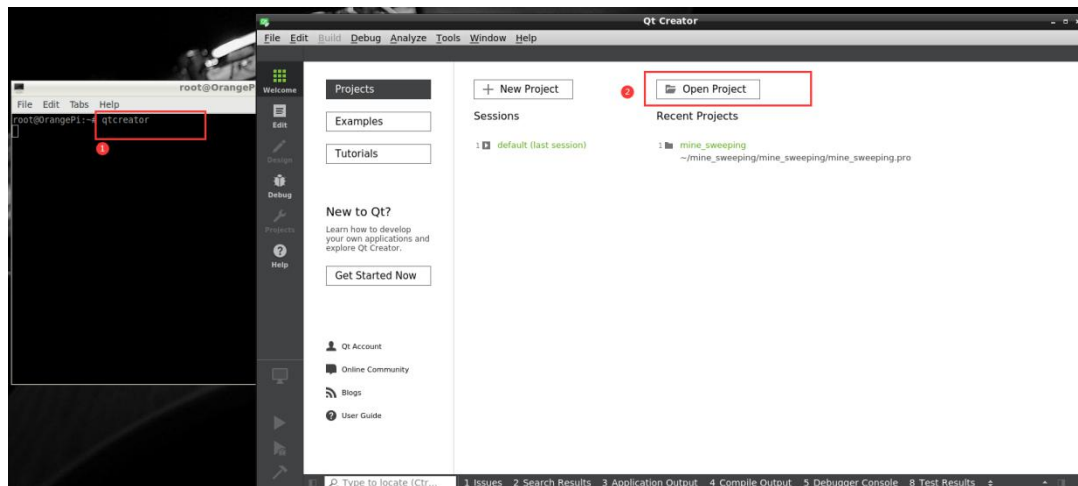
检查安装

```
orange@orange:~$ qmake -v
QMake version 3.1
Using Qt version 5.9.5 in /usr/lib/aarch64-linux-gnu
```

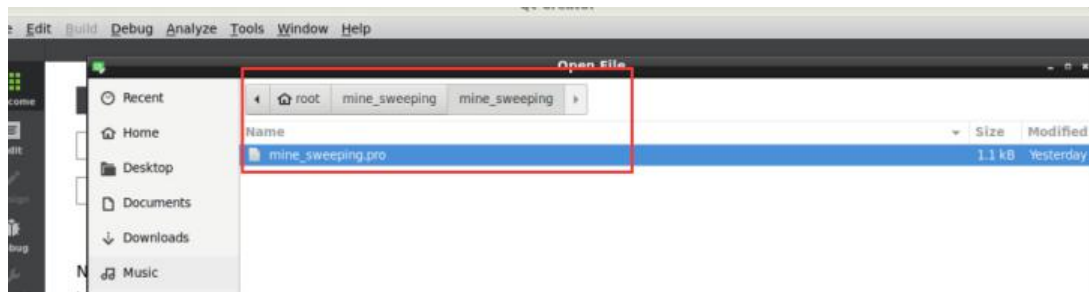
2、QT 使用示例：扫雷游戏

```
git clone https://github.com/qmeng0207/mine_sweeping
```

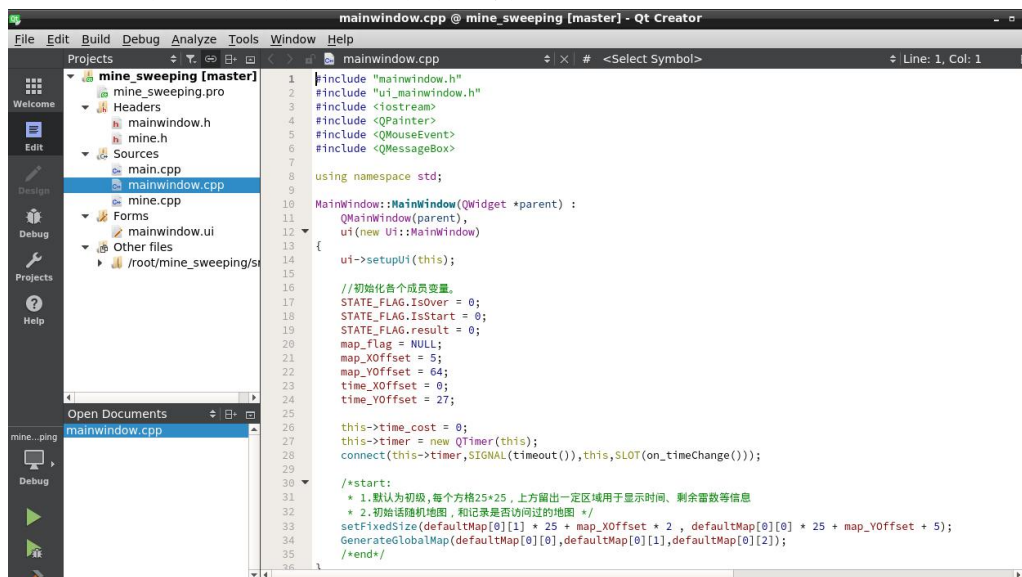
通过 Terminal 打开 qtcreator
如下图：

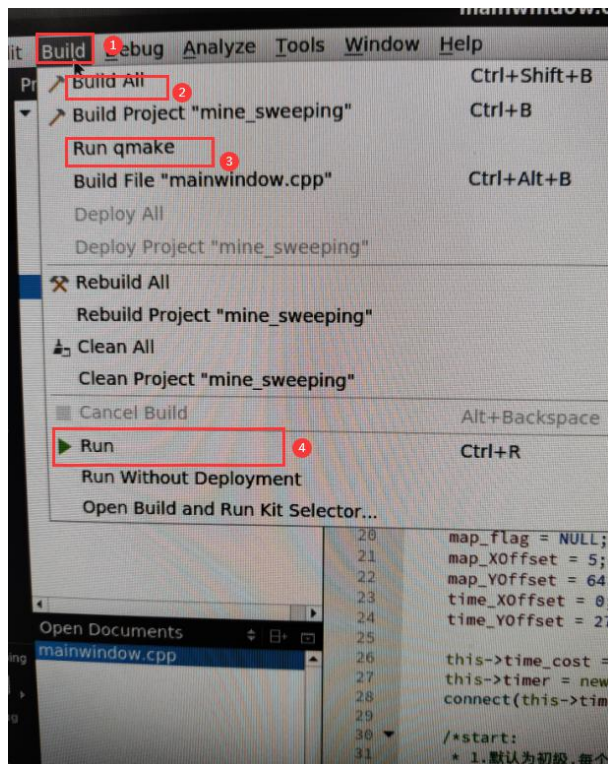


点击 Open Project 选择刚刚 git clone 下来的扫雷源码



Build 和运行此源码就能玩扫雷游戏了。





五、查看及控制 CPU 频率

Rockchip RK3399 为 6 核 ARM® 64 位处理器，主频高达 2.0GHz，基于 big.LITTLE 大小核架构：双核 Cortex-A72(大核)+四核 Cortex-A53(小核)

1、CPU 命令

①、查看 CPU0~3 支持的频率

```
root@OrangePi:~# cat
/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
408000 600000 816000 1008000 1200000 1416000
```

②、查看 cpu4~5 支持的频率

```
root@OrangePi:~# cat
/sys/devices/system/cpu/cpu4/cpufreq/scaling_available_frequencies
408000 600000 816000 1008000 1200000 1416000 1608000 1800000
```

③、设置 CPU0~5 的最高频率

```
echo 816000 > /sys/devices/system/cpu/cpu0 (0~5)
/cpufreq/scaling_max_freq
```

④、查看 CPU0~5 的当前频率

```
root@OrangePi:~# cat
/sys/devices/system/cpu/cpu[012345]/cpufreq/cpuinfo_cur_freq
408000
408000
408000
408000
408000
408000
```

⑤、查看当前 CPU 的温度

```
root@OrangePi:~# cat /sys/class/thermal/thermal_zone[01]/temp
48750
48750
```

⑥、CPU ID

```
root@OrangePi:~# hexdump /sys/bus/nvmem/devices/rockchip-efuse0/nvmem
00000000 4b52 9933 fe91 5421 4356 3432 2e38 3030
00000010 0000 0000 060b 230c 1e18 1915 0008 0000
00000020 0000 0010 0000 0000 0000 0000 0000 0000
00000030 0000 0000 0000 0000 0000 0000 0000 0000
*
00000080
```

⑦、查看的是 cpu 的统计信息

```
root@OrangePi:~# lscpu
Architecture:      aarch64
Byte Order:        Little Endian
CPU(s):            6
On-line CPU(s) list:  0-5
Thread(s) per core:  1
Core(s) per socket:  3
Socket(s):         2
Vendor ID:         ARM
```

Model:	4
Model name:	Cortex-A53
Stepping:	r0p4
CPU max MHz:	1800.0000
CPU min MHz:	408.0000
BogoMIPS:	48.00

六、集成 OpenCV

1、OpenCV 介绍

OpenCV 是一个广受欢迎的开源计算机视觉库，它提供了很多函数，实现了很多计算机视觉算法，算法从最基本的滤波到高级的物体检测皆有涵盖。OpenCV 的设计目标是执行速度尽量快，主要关注实时应用，它采用优化的 C/C++ 代码编写，能够充分利用多核处理器的优势，其主要目标是构建一个简单易用的计算机视觉框架，以帮助开发人员更便捷地设计更复杂的计算机视觉相关应用程序。

OpenCV 于 1999 年由 Intel 建立，如今由 Willow Garage 提供支持。它是一个基于开源发行的跨平台计算机视觉库，可以运行在 Linux、Windows、Mac OS、Android、iOS、Maemo、FreeBSD、OpenBSD 等操作系统上。

OpenCV 可用于解决如下领域的问题：

人机交互、物体识别、图像分区、人脸识别、动作识别、运动跟踪、机器人、运动分析、机器视觉、结构分析、汽车安全驾驶

2、安装 OpenCV

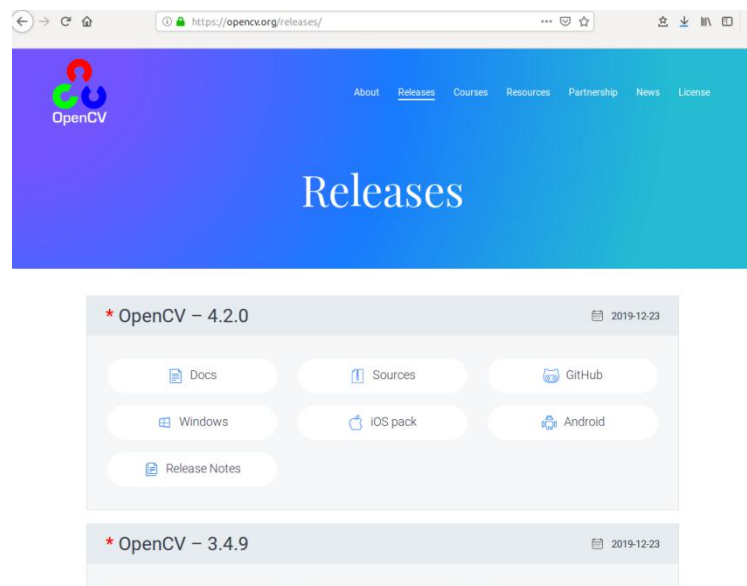
①首先安装必要的软件和依赖库等

```
sudo apt-get update
sudo apt-get install cmake git

sudo apt-get install cmake qt5-default libvtk6-dev zlib1g-dev
libjpeg-dev libwebp-dev libpng-dev libtiff5-dev libopenexr-dev
libgdal-dev libdc1394-22-dev libavcodec-dev libavformat-dev
libswscale-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev
yasm libopencore-amrnb-dev libopencore-amrwb-dev
```

②下载 OpenCV 源码

在 OpenCV 官网 <https://opencv.org/releases/> 选择你需要的版本和下载方式下载。



这里使用的是 OpenCV - 4.2.0 版本。

③编译

```
cd opencv-4.2.0
mkdir build
cd build
cmake -D CMAKE_BUILD_TYPE=RELEASE -D \
CMAKE_INSTALL_PREFIX=/usr/local/opencv ..
make -j4
make install
```

```
ldconfig
```

简单测试示例

```
import cv2 as cv
img = cv.imread("/usr/share/lxde/wallpapers/orangepi.jpg")
cv.namedWindow("Image")
cv.imshow("Image", img)
cv.waitKey(0)
cv.destroyAllWindows()
```

七、集成 XRDP 远程桌面

1、安装 XRDP 远程桌面

```
sudo apt-get install xrdp
sudo apt-get install vnc4server tightvncserver
```

2、启动 xrdp

```
/etc/init.d/xrdp start
```

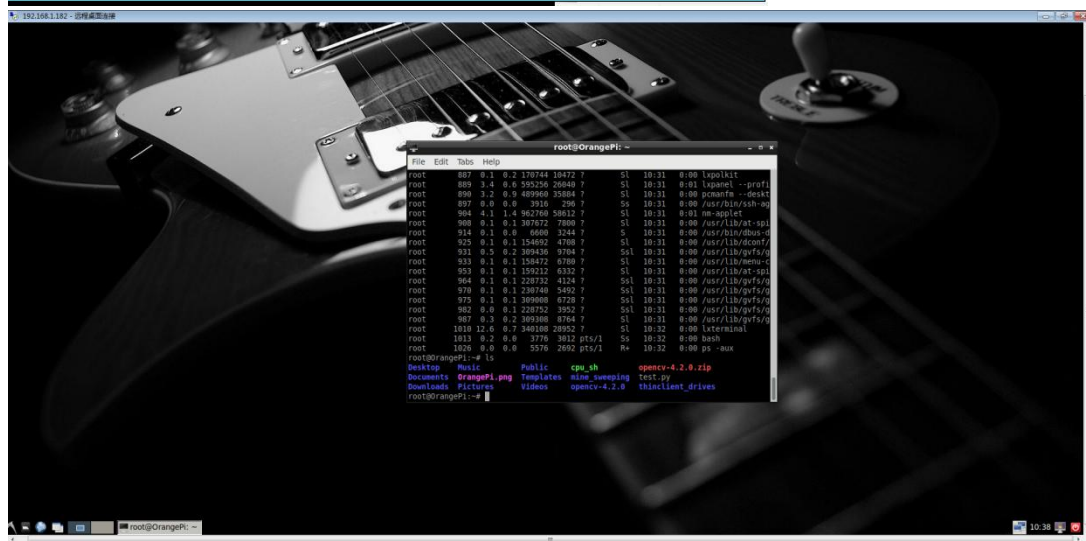
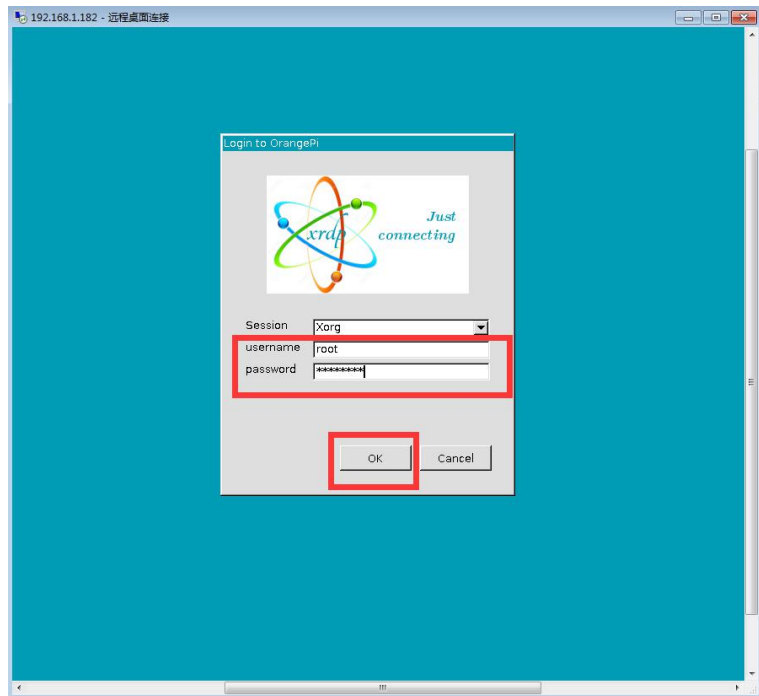
需要设置密码

在 windows 上打开远程桌面软件



点击连接，

4、输入用户和密码，点击“OK”，就连接上了，会直接进入 Pi4 桌面。



八、USB 摄像头使用

1、UVC 英菲克 USB 网络摄像头

①直接将摄像头插入 Pi4 的 USB2.0 或 3.0，然后就可以查询到 USB 摄像头的节点了：

```
root@OrangePi:~# ls /dev/video10 -la
```

```
crw-rw----+ 1 root video 81, 15 Jan  2 13:15 /dev/video10
```

我的节点是 video10，设备节点在你的板子上可能不是/dev/video10，你需要检查一下。

②设备节点出来后，就可以使用 USB 摄像头了。

使用 motion 来使用 USB 摄像头：

安装 motion

```
Sudo apt-get install motion
```

修改配置

```
$ sudo nano /etc/motion/motion.conf
```

```
$ stream_localhost off
```

在 motion.conf 配置文件中请注意 USB 摄像头的设备节点是 video?多少，并相应修改

创建文件夹，用于保存图片

```
$ mkdir ~/motion
```

修改权限

```
$ chmod 777 motion
```

继续修改配置

```
$ sudo nano /etc/default/motion
$ start_motion_daemon=yes
```

启动服务器

```
$ sudo /etc/init.d/motion start
```

最后一步，在浏览器中输入

localhost:8081

即可查看摄像头输出的图像

九、用摄像头+开发板实现视频推流

此章节介绍在 Pi4 RK3399 的 linux 平台，用摄像头采集实时视频数据流，推送至 RTMP 监听服务器，用客户端访问这个 RTMP 服务器就可以观看视频了。

系统版本：Ubuntu 18.04.1 LTS (GNU/Linux 4.4.179-gb623ef72-dirty aarch64)桌面版
镜像：OrangePi_4_ubuntu_bionic_desktop_linux4.4.179_v1.2.img

需要使用 ov13850 摄像头一个，可到 orangepi 官方购买。

1、搭建 RTMP 流直播服务器

在开发板上安装 nginx 和 nginx-rtmp-module 模块来作为 RTMP 服务端，依次执行以下命令：

```
root@OrangePi:~# apt-get update
root@OrangePi:~# apt-get -y install nginx
root@OrangePi:~# apt-get -y remove nginx
root@OrangePi:~# apt-get clean
root@OrangePi:~# rm -fr /etc/nginx/*

root@OrangePi:~# apt-get install -y curl build-essential libpcre3
libpcre3-dev libpcre++-dev zlib1g-dev libcurl4-openssl-dev libssl-dev
```

```
root@OrangePi:~# mkdir -p /var/www
root@OrangePi:~# rm -fr nginx_src
root@OrangePi:~# mkdir -p nginx_src
root@OrangePi:~# cd nginx_src/
root@OrangePi:~/nginx_src# ls
root@OrangePi:~/nginx_src# NGINXSRC=$PWD
root@OrangePi:~/nginx_src# wget
http://nginx.org/download/nginx-1.13.8.tar.gz
root@OrangePi:~/nginx_src# git clone
https://github.com/arut/nginx-rtmp-module.git
```

```
cd nginx-rtmp-module/
git checkout -b ngl.13.8 791b6136f02bc9613daf178723ac09f4df5a3bbf
```

```
root@OrangePi:~/nginx_src# tar -xvf nginx-1.13.8.tar.gz
root@OrangePi:~/nginx_src# cd nginx-1.13.8/
```

```
./configure --prefix=/var/www --sbin-path=/usr/sbin/nginx
--conf-path=/etc/nginx/nginx.conf --pid-path=/var/run/nginx.pid
--error-log-path=/var/log/nginx/error.log
--http-log-path=/var/log/nginx/access.log --with-http_ssl_module
--without-http_proxy_module --add-module=$NGINXSRC/nginx-rtmp-module
```

编译 nginx

```
root@OrangePi:~/nginx_src/nginx-1.13.8# make
root@OrangePi:~/nginx_src/nginx-1.13.8# make install
```

查看版本信息:

```
root@OrangePi:~/nginx_src/nginx-1.13.8# nginx -v
nginx version: nginx/1.13.8
```

2、配置 RTMP 服务器

编辑 nginx 配置文件:

```
root@OrangePi:~# vim /etc/nginx/nginx.conf
```

在末尾添加如下 RTMP 配置:

```
rtmp {  
    server {  
        listen 1935;  
        chunk_size 4096;  
        application live {  
            live on;  
            record off;  
        }  
    }  
}
```

重启 nginx 服务让配置生效:

```
root@OrangePi:~# systemctl restart nginx
```

3、验证 RTMP 服务器

使用测试视频来确定 RTMP 服务器能否正常工作:

```
root@OrangePi:~# gst-launch-1.0 videotestsrc is-live=true !  
videoconvert ! x264enc bitrate=1000 tune=zerolatency ! video/x-h264 !  
h264parse ! video/x-h264 ! queue ! flvmux streamable=true name=mux !  
rtmpsink sync=false location='rtmp://192.168.1.160:1935/live/test'
```

注意: 需要将上述 IP 改为自己实际的 IP。

出现如下信息。

```
Setting pipeline to PAUSED ...  
Pipeline is live and does not need PREROLL ...  
Setting pipeline to PLAYING ...  
New clock: GstSystemClock  
Redistribute latency...
```

再在主机 PC 的 Ubuntu 系统下, 使用 rtmpdump 测试从 Pi4 拉去视频流:

①安装 rtmpdump:

安装依赖:

```
sudo apt-get install openssl
sudo apt-get install libssl-dev
sudo apt-get install zlib1g-dev
```

下载 rtmpdump 源码并编译安装:

```
git clone git://git.ffmpeg.org/rtmpdump
cd rtmpdump/
make
sudo make install
sudo ldconfig
ldd rtmpdump
```

②使用 rtmpdump 测试:

```
rtmpdump -r "rtmp://192.168.1.160/live/test" -V -z -o out.flv
```

按 **Ctrl+C** 暂停采集, 查看生成的 **out.flv** 文件, 可以正常播放说明 RTMP 服务器是正常工作的:

```
dale@dale:~/dale/share/Drivers-package/rtmpdump$ ls -lh out.flv
-rw-rw-r-- 1 dale dale 652K 1月 2 18:05 out.flv
```

4、推送 OV13850 摄像头图像到 RTMP 服务器

```
root@OrangePi:~# gst-launch-1.0 rkisp io-mode=1 device=/dev/video1 !
video/x-raw,format=NV12,width=1280,height=720,framerate=30/1 ! queue !
mpph264enc ! h264parse ! flvmux streamable=true ! queue ! rtmpsink
sync=false location='rtmp://192.168.1.160:1935/live/test'
```

注: 将上述 IP 地址改成自己板子的。

Setting pipeline to PAUSED ...

mpi: mpp version: Without VCS info

mpp_rt: NOT found ion allocator

mpp_rt: found drm allocator

media get entity by name: lens is null

Pipeline is live and does not need PREROLL ...

[XCORE]:XCAM ERROR rkisp_device.cpp:72: atomisp device(/dev/video1) try subdev
format failed

Setting pipeline to PLAYING ...

New clock: GstSystemClock

h264e_api: h264e_config MPP_ENC_SET_RC_CFG bps 3456000 [3240000 : 3672000]

5、播放视频流

①在手机上播放

如果手机是 iPhone，安装一个 SLDP Player 的免费软件，在软件上添加流地址 `rtmp://192.168.1.160/live/test` 即可查看实时图像：



②通过页面播放

创建一个播放页面，达到只需用浏览器打开网址 `http://开发板的 IP 地址`，即可浏览直播画面的目的，以下为创建步骤：

```
root@OrangePi:~# mkdir -p strobe_src
root@OrangePi:~# cd strobe_src/
root@OrangePi:~/strobe_src# wget
http://downloads.sourceforge.net/project/smp.adobe/Strobe%20Media%20P
layback%201.6%20Release%20%28source%20and%20binaries%29/StrobeMediaPl
ayback_1.6.328-full.zip
root@OrangePi:~/strobe_src# unzip StrobeMediaPlayback_1.6.328-full.zip
root@OrangePi:~/strobe_src# cp -r for\ Flash\ Player\ 10.1
/var/www/html/strobe
```

编辑 `/var/www/html/index.html`，填入如下内容，记得把下面的“开发板的 IP 地址”替换成你的实际的 IP 地址，IP 地址可以通过 `ifconfig` 命令查看：

```
root@OrangePi:/var/www/html# cat index.html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
```

```

<title>OrangePi Live</title>
<script type="text/javascript"
src="strobe/lib/swfobject.js"></script>
<script type="text/javascript">
    // Create a StrobeMediaPlayback configuration
    var parameters = {
    // src:
"http://players.edgesuite.net/videos/big_buck_bunny/bbb_448x252.mp4",
    src: "rtmp://192.168.1.160/live/test",
    autoPlay: true,
        controlBarAutoHide: false,
        playButtonOverlay: true,
        showVideoInfoOverlayOnStartup: false,
        optimizeBuffering : false,
        initialBufferTime : 0.1,
        expandedBufferTime : 0.1,
        minContinuousPlayback : 0.1,
        poster: "strobe/images/poster.png"
    };

    // Embed the player SWF:
    swfobject.embedSWF
    ( "strobe/StrobeMediaPlayback.swf"
      , "strobeMediaPlayback"
      , 640
      , 480
      , "10.1.0"
      , {}
      , parameters
      , { allowFullScreen: "true"}
      , { name: "strobeMediaPlayback" }
    );
</script>
</head>
<body>
    <div id="strobeMediaPlayback">
        <p>Alternative content</p>
    </div>
</body>
</html>

```

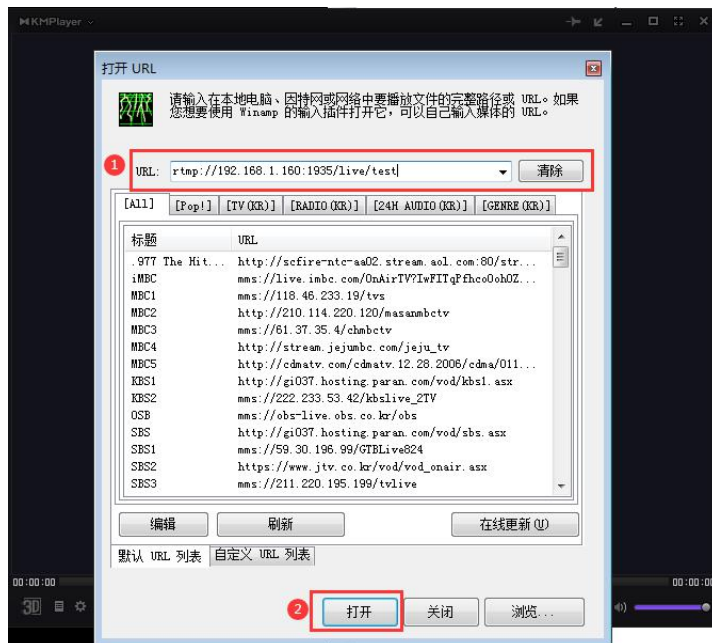
最后，在电脑上用浏览器打开网址 [http://开发板的 IP 地址](http://开发板的IP地址) 即可进入播放界面

③在 Windows 中使用 Kmplayer Plus 软件播放

在 Windows 中安装 Kmplayer Plus 软件，打开软件并添加流地址，即可查看实时

图像。

打开 Kmpayer Plus 并按 Ctrl+u 在 URL 地址栏添加流地址：
rtmp://192.168.1.160:1935/live/test，再点击“打开”即出现直播画面：



十、在 orangepi 4 上使用 Docker

1. 下载安装包

```
wget
https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/arm
64/containerd.io_1.2.6-3_arm64.deb
wget
https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/arm
64/docker-ce-cli_19.03.2~3-0~ubuntu-bionic_arm64.deb
wget
https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/arm
64/docker-ce_19.03.2~3-0~ubuntu-bionic_arm64.deb
```

2. 安装 docker

```
sudo dpkg -i *.deb
```

3. Docker 使用

(1) 直接运行

终端上执行命令

```
sudo docker run hello-world
```

出现如下信息

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
be6e184261a6: Pull complete
```

```
Digest:
sha256:4fe721ccc2e8dc7362278a29dc660d833570ec2682f4e4194f4ee23e415e10
64
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

直接执行以下命令可以进到 **ubuntu docker** 中

```
root@OrangePi:~# docker run -it ubuntu bash
```

```
0;root@0a8c24c3c7de: /root@0a8c24c3c7de:/# ls
```

```
bin    dev    home   media  opt    root   sbin   sys    usr
```

```
boot   etc    lib    mnt    proc   run    srv    tmp    var
```

(2) 搜索到支持 **arm64v8** 格式的可用的 **image** 然后执行

执行 **docker search arm64v8** 搜到 **arm64v8/nginx**

然后就可以运行 **docker run -it arm64v8/nginx bash**

(3) 搜索到格式不支持的 **image** 重新 **build** 成 **arm64** 格式的再运行

例如 执行 **docker search ubuntu** 找到 **eclipse/ubuntu_jdk8** 以后

如果直接执行 **docker run -it eclipse/ubuntu_jdk8 bash** 会报格式不对的错，

在本地建立 **docker** 目录，然后建立如下内容的两个文件分别叫

Dockfile 和 sources.list 的文件放进去

Dockerfile:

FROM armhf/debian

COPY ./sources.list /etc/apt/sources.list

RUN apt-get update

RUN echo "root:fa" | chpasswd

USER root

RUN echo "all done."

sources.list:

```
deb [arch=armhf] http://ftp2.cn.debian.org/debian/ jessie main
contrib non-free
```

```
deb-src [arch=armhf] http://ftp2.cn.debian.org/debian/ jessie main
contrib non-free
```

```
# deb [arch=armhf] http://ftp2.cn.debian.org/debian/
jessie-updates main contrib non-free
```

```
# deb-src [arch=armhf] http://ftp2.cn.debian.org/debian/
jessie-updates main contrib non-free
```

然后执行一下 `docker build -t "eclipse/ubuntu_jdk8" docker` 这个命令 再执行 `docker run -it eclipse/ubuntu_jdk8 bash` 就可以运行了

(4) 使用 raspberry pi 的 docker image

在以下网址

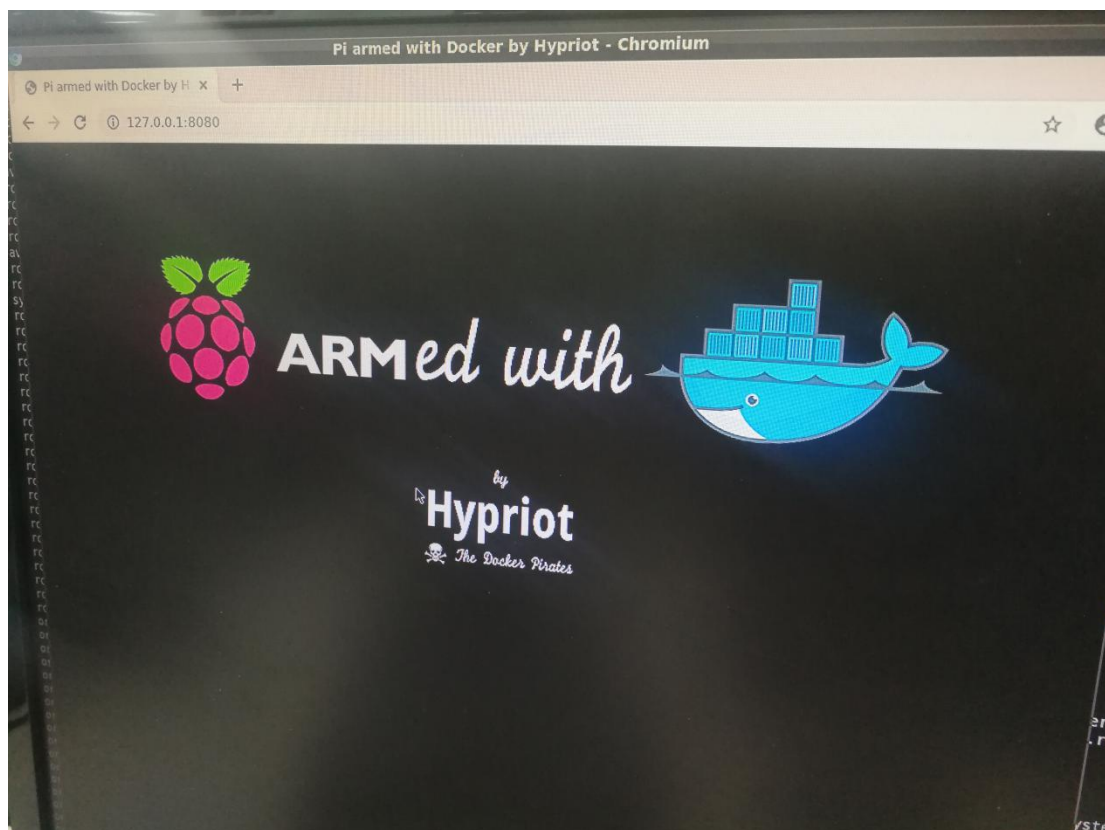
<https://hub.docker.com/u/hypriot/> 可以找到很多支持 Raspberry pi 的 docker image 比如 rpi-busybox-httpd , 这些 OrangePi 4 也是支持的

执行一下命令后,

```
docker run -d -p 8080:80 hypriot/rpi-busybox-httpd
```

```
0cbdce32ec6c639e55b6a8d8e09c030e7c1acc994dd02308af51fa8be9d041e5
```

可以通过浏览器 127.0.0.1:8080 访问到服务器页面如下



十一、mjpg-streamer 网络视频流服务器

```
git clone https://github.com/jacksonliam/mjpg-streamer
sudo apt-get install cmake libjpeg8-dev
sudo apt-get install gcc g++

cd mjpg-streamer-experimental
make
sudo make install
```

start.sh 有使用的例子，在 Pi4 上插入 uvc 摄像头，对应的设备节点应该是 video10。

做如下修改。

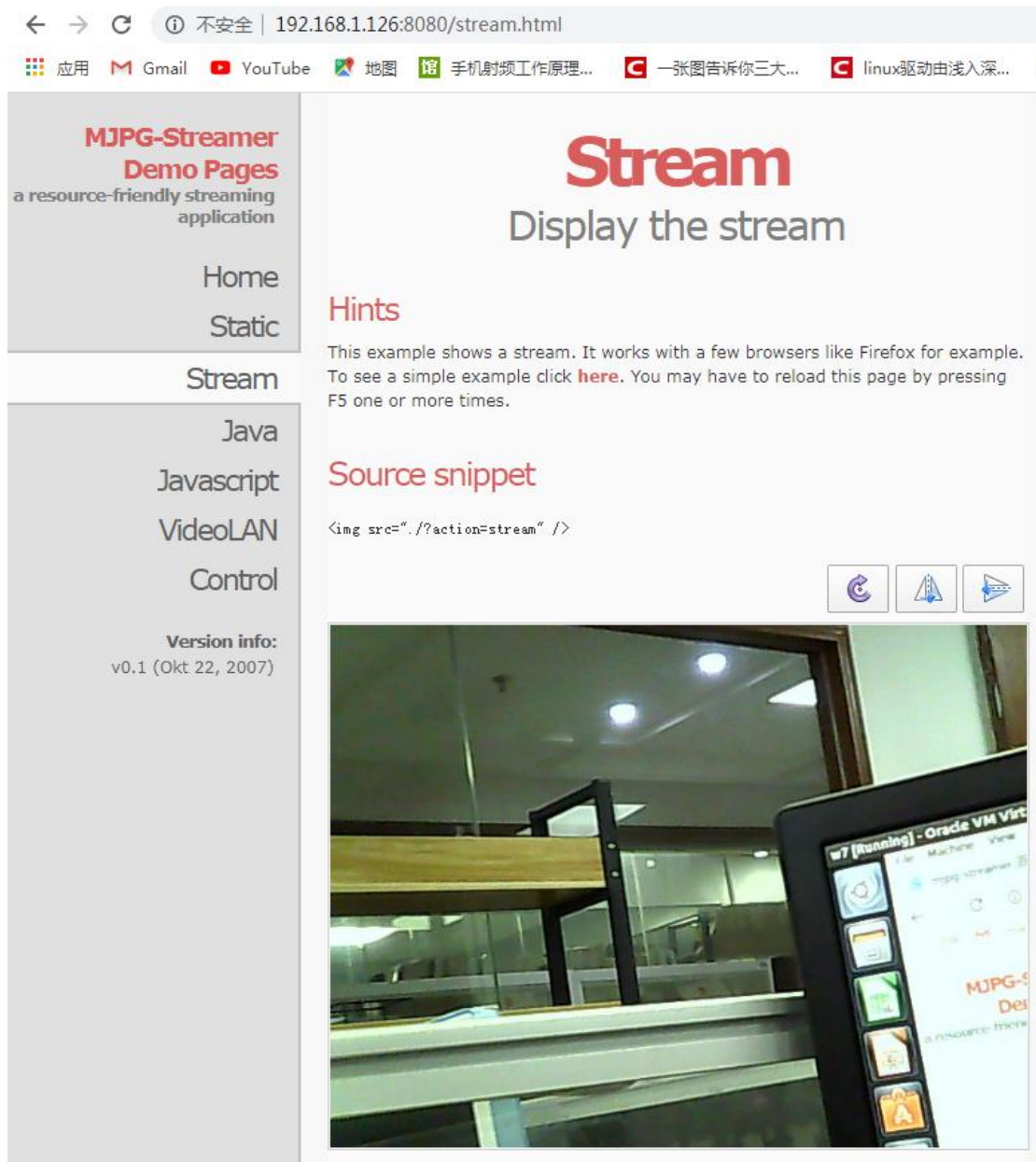
```
export LD_LIBRARY_PATH="$PWD"
#./mjpg_streamer -i "input_uvc.so --help"

#./mjpg_streamer -i "./input_uvc.so" -o "./output_http.so -w ./www"
#./mjpg_streamer -i "./input_uvc.so -n -f 30 -r 1280x960" -o "./output_http.so -w ./www"
./mjpg_streamer -i "./input_uvc.so -n -f 30 -r 640x480 -d /dev/video10" -o "./output_http.so -w ./www" &
#./mjpg_streamer -i "./input_uvc.so -d /dev/video0" -i "./input_uvc.so -d /dev/video1" -o "./output_http.so -w ./www"
#valgrind ./mjpg_streamer -i "./input_uvc.so" -o "./output_http.so -w ./www"
```

执行 start.sh。有如下信息

```
root@OrangePi:~/mjpg-streamer/mjpg-streamer-experimental# ./start.sh
root@OrangePi:~/mjpg-streamer/mjpg-streamer-experimental#
MJPEG Streamer Version: git rev:
501f6362c5afddcfb41055f97ae484252c85c912
i: Using V4L2 device.: /dev/video10
i: Desired Resolution: 640 x 480
i: Frames Per Second.: 30
i: Format.....: JPEG
i: TV-Norm.....: DEFAULT
i: Could not obtain the requested pixelformat: MJPG , driver gave us:
YUYV
... will try to handle this by checking against supported formats.
... Falling back to YUV mode (consider using -yuv option). Note that
this requires much more CPU power
o: www-folder-path.....: ./www/
o: HTTP TCP port.....: 8080
o: HTTP Listen Address..: (null)
o: username:password....: disabled
o: commands.....: enabled
```

浏览器输入板子 IP 地址加端口号。如 192.168.1.126:8080



系统环境: OrangePi_4_ubuntu_bionic_desktop_linux4.4.179_v1.2.img
硬件环境: V1.3

十二、安装 opencv

apt-get 方式安装

运行以下命令

```
apt-get install libopencv-dev  
apt-get install python-opencv
```

测试程序 test.py

```
import cv2 as cv  
img = cv.imread("/usr/share/lxde/wallpapers/orangepi.jpg")  
cv.namedWindow("Image")  
cv.imshow("Image", img)  
cv.waitKey(0)  
cv.destroyAllWindows()
```

运行测试程序

```
python test.py
```

十三、集成 caffe

1、安装依赖项

```
sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev  
libopencv-dev libhdf5-serial-dev protobuf-compiler  
  
sudo apt-get install --no-install-recommends libboost-all-dev  
sudo apt-get install libopenblas-dev liblapack-dev libatlas-base-dev  
sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
```

2、下载 Caffe 并修改配置参数

```
git clone https://github.com/BVLC/caffe  
cd caffe  
cp Makefile.config.example Makefile.config
```

修改 Makefile.config 中的参数：

将 `#CPU_ONLY := 1` 改为 `CPU_ONLY := 1`

将 `#OPENCV_VERSION := 3` 改为 `OPENCV_VERSION := 3`

3、安装 caffe

```
mkdir build  
cd build  
cmake ..  
make all -j1  
make pycaffe -j1  
make test -j1  
make runtest -j1
```

4、运行简单示例

运行 caffe 自带的例子—mnist 实例：

mnist 是一个手写数字库。mnist 最初用于支票上的手写数字识别，现在成了 DL 的入门练习库。针对 mnist 识别的专门模型是 Lenet，算是最早的 cnn 模型了。mnist 数据训练样本为 60000 张，测试样本为 10000 张，每个样本为 28*28 大小的黑白图片，手写数字为 0-9，因此分为 10 类。

①首先下载 mnist 数据

```
root@OrangePi:~/caffe# sh data/mnist/get_mnist.sh
```

②运行成功后，在 data/mnist/目录下有四个文件：

```
root@OrangePi:~/caffe/data/mnist# ls
t10k-labels-idx1-ubyte    //训练集对应标注
train-labels-idx1-ubyte   //测试集对应标注
t10k-images-idx3-ubyte    //训练集样本
train-images-idx3-ubyte   //测试集样本
```

③这些数据不能在 caffe 中直接使用，需要转换成 LMDB 数据：

```
root@OrangePi:~/caffe# sh examples/mnist/create_mnist.sh
```

④如果想运行 leveldb 数据，请运行 examples/siamese/ 文件夹下面的程序。而 examples/mnist/ 文件夹是运行 lmdb 数据转换成功后，会在 examples/mnist/目录下，生成两个文件夹，分别是 mnist_train_lmdb 和 mnist_test_lmdb，里面存放的 data.mdb 和 lock.mdb，就是我们需要的运行数据。

⑤接下来修改配置文件 lenet_solver.prototxt

```
root@OrangePi:~/caffe/examples/mnist# vi lenet_solver.prototxt
```

根据需要，在 max_iter 处设置最大迭代次数，以及最后一行 solver_mode，改成 CPU

⑥最后是运行这个例子：

```
root@OrangePi:~/caffe# time sh examples/mnist/train_lenet.sh
```

```
I0110 06:28:06.117972 25078 caffe.cpp:197] Use CPU.
```

```
I0110 06:28:06.118988 25078 solver.cpp:45] Initializing solver from parameters:
```

```
test_iter: 100
```

```
test_interval: 500
```

```
base_lr: 0.01
```

```
display: 100
```

```
max_iter: 10000
```

```
lr_policy: "inv"
gamma: 0.0001
power: 0.75
momentum: 0.9
weight_decay: 0.0005
snapshot: 5000
snapshot_prefix: "examples/mnist/lenet"
solver_mode: CPU
net: "examples/mnist/lenet_train_test.prototxt"
.....
.....
.....
I0110 07:25:37.016746 25078 sgd_solver.cpp:284] Snapshotting solver state to binary proto file
examples/mnist/lenet_iter_10000.solverstate
I0110 07:25:37.146054 25078 solver.cpp:327] Iteration 10000, loss = 0.00200602
I0110 07:25:37.146195 25078 solver.cpp:347] Iteration 10000, Testing net (#0)
I0110 07:25:54.299634 25080 data_layer.cpp:73] Restarting data prefetching from start.
I0110 07:25:55.010598 25078 solver.cpp:414]      Test net output #0: accuracy = 0.9913
I0110 07:25:55.010757 25078 solver.cpp:414]      Test net output #1: loss = 0.0273203 (* 1 =
0.0273203 loss)
I0110 07:25:55.010777 25078 solver.cpp:332] Optimization Done.
I0110 07:25:55.010793 25078 caffe.cpp:250] Optimization Done.

real 57m48.984s
user 58m6.479s
sys 0m1.977s
```

⑦运行结论

根据以上运行结果可知此次 CPU 运行时间大约为 58 分钟，精度为 99%左右。

十四、集成 TensorFlow

TensorFlow: 1.14.0

Python 环境: python2.7

1、python 环境与依赖项的安装

此次选择预装的 python2.7，不再安装 python3.x

安装依赖：

```
sudo apt-get install python-pip python-dev libatlas-base-dev
```

2、安装 TensorFlow

安装 TensorFlow 有多种方式，这里选择较为方便直接命令安装。

步骤：

从以下下载 TensorFlow 的 pip 安装包

tensorflow-1.14.0-cp27-none-linux_aarch64.whl

<https://github.com/lhelontra/tensorflow-on-arm/releases/tag/v1.14.0>

使用 pip 安装：

`pip install tensorflow-1.14.0-cp27-none-linux_aarch64.whl`

注意，安装过程中可能由于网络的原因下载不下来有些依赖项请根据 log 显示的依赖版本下载下来安装上去后，再安装 TensorFlow。

vi test.py

```
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
#sess = tf.Session()
sess = tf.compat.v1.Session()
print(sess.run(hello))
```

运行测试程序

```
root@OrangePi:~# python test.py
Hello, TensorFlow!
```

注：此次只使用了 python2.7 环境安装 TensorFlow，python3.5 以及 python3.6 都是可以的，请自行实验。

