

树莓派机器人快速入门指南 V1.3

——带视觉全自主智能人形机器人训练平台 HBE-ROBONOVA-AI II

本教程基于 HBE-ROBONOVA-AI II，也可作为树莓派入门参考
原创作品，仅供个人研究使用，请勿转发！谢谢！



始于：2017 年 2 月 14 日

312 现代电子测量实验室、3-306 创新基地

YOBOT 团队-兰州交通大学

注意：所有资料仅供个人学习，本文档版权归刘亚龙所有，代码版权归 HBE-ROBONOVA-AI II 的生产公司所有，请勿转发！

声 明

本手册请务必从前往后逐字看，否则可能会得不到最终的结果，甚至损坏硬件！

资料内部的程序都是英文开发的，所以，放置资料的文件夹名称不要有中文或特殊符号，以防止程序运行不正常。

我尽量将入门教程写的通俗易懂，争取让小白菜也可以轻松学会 HBEROBONOVA-AI II 机器人平台与树莓派的入门操作。

本手册内容均由本人亲自编辑完成，本手册 V1.2 及之前的版本仅供本科学校（兰州交通大学）实验室（312 现代电子测量实验室、3-306 创新基地）人员使用！V1.3 及以后版本将公开以供大家一起进步。

西安电子科技大学

2020/01/02

刘亚龙

简介

咱实验室进购的这款机器人平台应用了树莓派这一开源平台，总体来说还是蛮好的，适合有一定基础的同学学习。但是，这个平台的资料有点不全，把它玩起来也是挺麻烦的，我查阅了好多资料，进行了实际的操作，特将入门步骤总结于此。

本文的目的是：教你学会用树莓派编写、编译一个 `helloworld` 程序、运行机器人系统内自带的踢球程序以及编辑踢球程序的自动运行脚本，并于文档最后附带了 VNC 教程资料。树莓派本身就是一个具有通用性的平台，可以用来实现一些嵌入式、图像处理、神经网络、单片机等方面的技术，把这个平台用好了，对个人以及团队的进步等都很有益处。

在这个教程里面我会尽量详细的带大家入门树莓派机器人（HBE-ROBONOVA-AI II），First of all，你需要明白整个机器人平台的工作原理。本平台分为树莓派上位机和单片机下位机两部分，树莓派处理数据后，通过串口发送动作对应的数字给单片机，单片机控制舵机输出动作，这是一个连续的过程。如果在本手册当中遇到看不懂的术语或者命令，这时候就应该找度娘去涨姿势了。

当然了，查找资料的地方有很多，常用的有度娘，某宝，Google，WIKI 等。Wait??? 某宝也行？是的呀，什么 USB-TTL 驱动一类的程序我都是在某宝上面找的，你先某宝 USBTTL，有的宝贝链接里面会有免费的资料链接。因此，某宝有时候比度娘还好用！

题外话：如果你发现系统的软硬件没有按照自己的意愿来工作，或者出现了无法解决的 BUG，一般情况下请参考苗老师的电子三定律：

- 1、硬件本身不会出错！
- 2、软件本身不会出错！
- 3、如果你觉得出现 BUG 是软硬件的问题，请参考以上两条！

所以，如果哪一步出现异常没有成功，请参考手册结尾的 FAQ 或自行上网查错，困难也是一种机会，把握机会提升自己，一起进步。

目 录

目录

一些话	5
可能会用到的周边.....	5
需要的技能.....	5
注 意	5
资料目录总览.....	6
树莓派 HelloWorld 编程	7
第一课 • 机器人平台原理.....	7
第二课 • 树莓派开机.....	7
第三课 • 树莓派基本操作.....	7
第四课 • 终端命令运行程序.....	8
第五课 • 修改程序.....	9
第六课 • 编译程序.....	10
树莓派机器人踢球程序.....	11
第七课 • 踢球程序.....	11
第八课 • 自动运行脚本.....	13
第九课 • 系统进程处理.....	14
第十课 • 系统开机自动运行.....	15
附 录	16
配置 VNC.....	16
建议.....	16
树莓派终端常用命令.....	16
FAQ.....	17
历史版本.....	17

一些话

可能会用到的周边

- 1、树莓派板子
- 2、带 AV 或 HDMI 输入的显示器及相关线材
- 3、鼠标
- 4、键盘

需要的技能

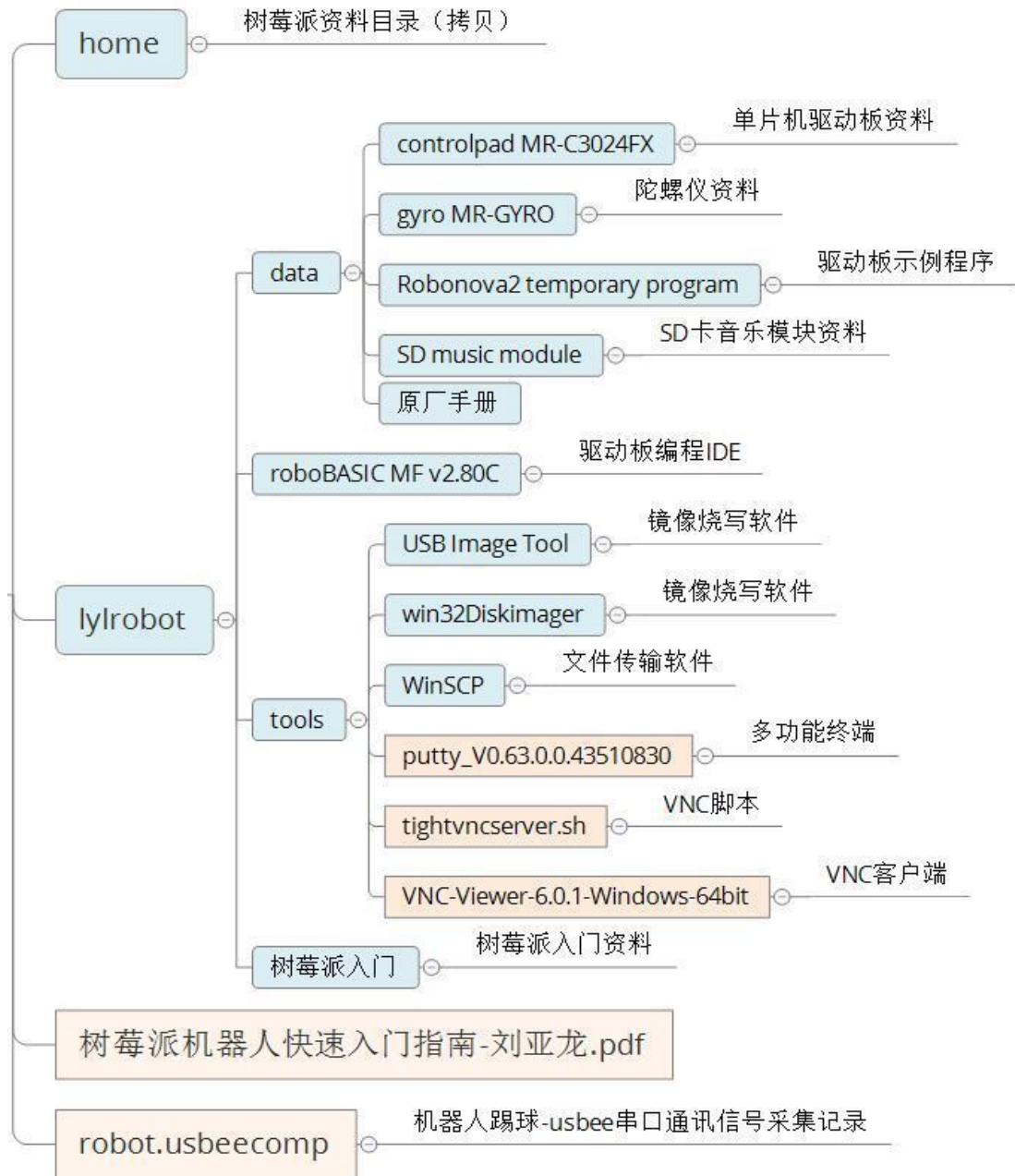
- 1、英语
- 2、C、C++
- 3、电脑基础
- 4、硬件基础

（建议先有这些基础，没学好这些，想要操作树莓派的话是比较痛苦的）

注 意

- 1、首要事项：树莓派质量很好，但是，许多玩家都出现过意外事故。其中之一就是尽量不要在 HDMI 的接口上直接插 HDMI-VGA 转换器。即尽量不要使用 HDMI 口向外部的 HDMI 设备供电！不然就有可能因为供电问题而永久烧毁树莓派主板！
- 2、树莓派如果使用了小型的无线网卡，更新软件时距离你家的路由器不要太远，否则速度会很慢。
- 3、打开 MOT 等需要显示图形窗口的程序，只能在本机的 Terminal 上输入命令才行，在 SSH 终端输这类命令没有效果。

资料目录总览



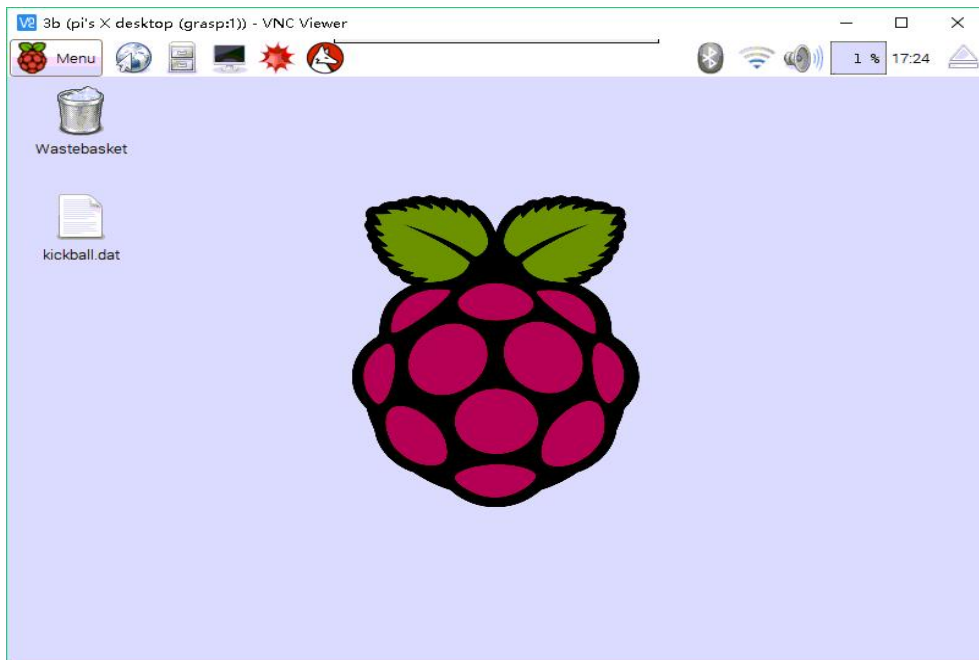
树莓派 HelloWorld 编程

第一课 • 机器人平台原理

树莓派通过 `opencv` 处理摄像头视频，之后将机器人的动作编号通过串口发送给 `AVR` 单片机，单片机控制舵机转动。

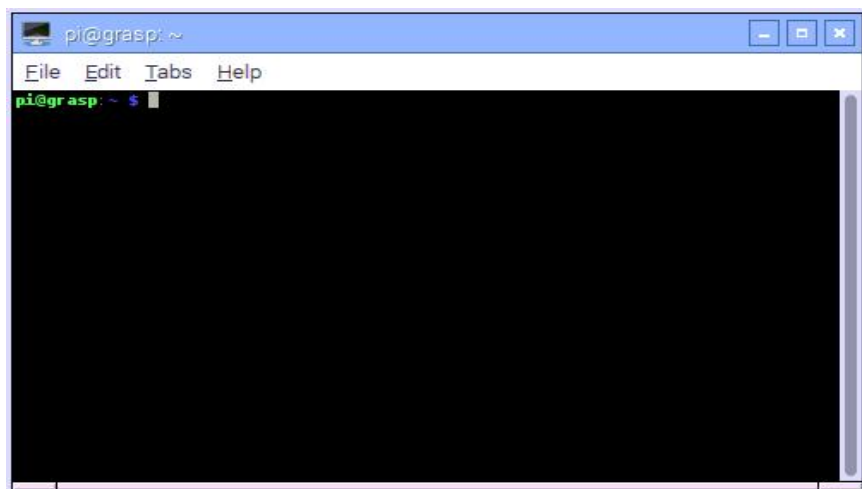
第二课 • 树莓派开机

给树莓派插上显示器（AV 或 HDMI，不推荐 4.3 寸液晶屏，屏太小看不清代码）、鼠标和键盘，并上电（建议 5V2A），等待开机进入桌面：



第三课 • 树莓派基本操作

打开 `shell`，左上角的这个图标 ，也就是命令行，也可以叫终端：



先记住不用操作：用户名是 `pi` 密码是 `xxxxxx`，之后输入 `sudo passwd` 可以改密码，菜单栏里面可以用鼠标改字体大小什么的，鼠标点几下就好了。

输入 ls 可以查看当前目录下有哪些文件：

```
pi@grasp: ~ $ ls
Desktop  Downloads  Pictures  python_games  Videos
Documents  Music      Public    Templates      working
```

当前所在目录为 pi 的用户目录

说明：linux 是一个用户一个目录，而且硬盘不分区，都是文件夹，软件用指令安装不用担心安装位置，系统全自动分配，文件怎么都不会乱，高度有序，和 windows 不一样。

查看目录可以 cd 后面加目录，不知道目录可以按 Tab 键查看当前目录选项，超级好用：

```
pi@grasp: ~ $ cd working/Book_exam_PI/
10.3 MovingObjectDetect/ 6.5.2 Video_exam2/
10.4 MovingObjectTracking/ 7.2.4 Gray_Scale/
11.2 ShapeDetection/ 7.2.5 SubSampling/
11.3 ShapeRecognitionRobot/ 7.2.6 window/
12. ComebackHome/ 7.3.1 Gray_Resolution/
13. FighterRobot/ 7.3.2 binary_image/
5.2 Hello/ 7.4.1 Smoothing/
5.2 Hello-makefile/ 7.4.2 Laplacian/
5.3 neck_spin/ 7.5 Edge_detection/
5.4 Commad/ 8.1.2 Brightness/
5.5 Walking/ 8.2 SensingBrightness/
5.6 Taekwondo/ 9.3 Color_detect/
5.7 motor_angle/ 9.4 ColorTracking/
5.8 robot_test/ kick/
6.3 ImageServer_libjpeg_mem/ test/
6.5.1 Video_exam/
```

可以看到这里有很多程序文件夹，因为这个树莓派是二次开发过的。我现在对他的操作已经算是三次开发了，现在我已经搞好了编程环境，可以直接进行编程，十分方便。

第四课 • 终端命令运行程序

我们先运行一下 HelloWorld：

输入 `sudo ./working/Book_exam_PI/5.2\ Hello/bulid/Hello`

```
pi@grasp: ~ $ sudo ./working/Book_exam_PI/5.2\ Hello/build/Hello
Hello! Wolrdddddddddddddddd!
```

这是一个打印字符串的程序，这个文件可能我已经修改过了，所以显示的是这样子。不论怎么样，会有字符串输出的。除非操作不对。

解释：

sudo 表示用 root 权限执行（不建议直接切换到 root 用户，root 权限过大，防止误操作系统关键文件，要用超级权限的时候直接前面加个 sudo 就好了，养成好习惯）

./表示运行程序

working/Book_exam_PI/5.2\ Hello/bulid/是目录

最后一个 Hello 是可执行文件

你看我输了这么大一串，其实我只按了几个键，文件夹几个开头的字母，之后按 Tab，系统会自动查找匹配的文件夹即文件，之后自动补全名称，如果有几个文件开头都一样，按下 Tab 会显示匹配的所有文件，继续手动输入直到不冲突。

这个目录里面可以试着运行各种程序看看，除了 HelloWorld 什么的程序，其他的一般都是用到了串口，所有结果都是通过串口输出去的。

第五课 • 修改程序

我们来修改一下这个 HelloWorld 吧，之后编译程序，让他输出别的字符串：
首先打开到这个目录下，省的每次操作都输入很长的目录。

```
pi@grasp:~$ cd working/Book_exam_PI/5.2\ Hello
pi@grasp:~/working/Book_exam_PI/5.2 Hello$ ls
build CMakeLists.txt CMakeLists.txt~ Hello.cpp
```

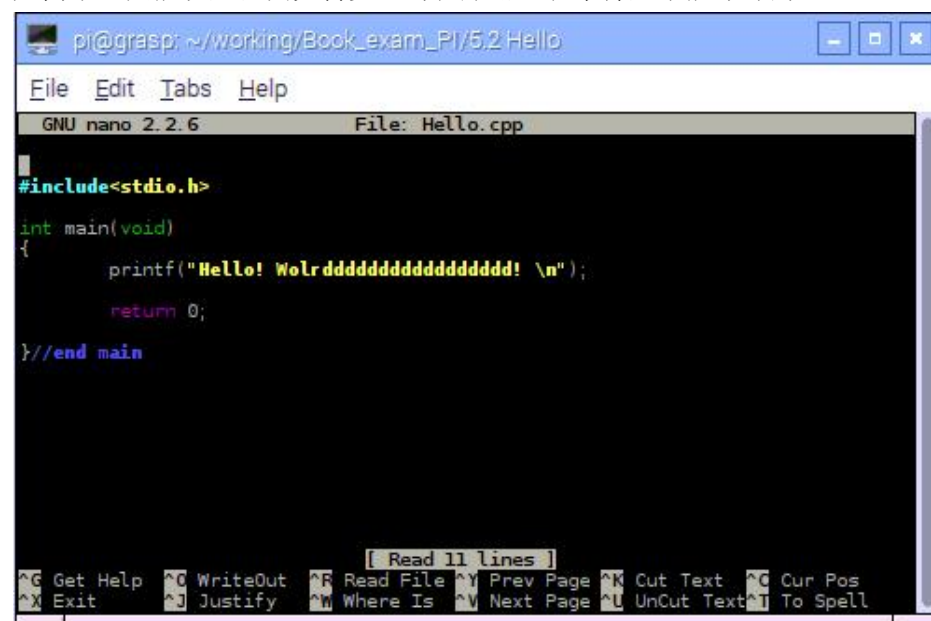
里面的 Hello.cpp 就是我们要操作的文件了

我还是新手，vi 太强大，用的不熟，还是用 nano 吧，nano 一向以小巧友好著称。

sudo nano Hello.cpp

```
pi@grasp:~/working/Book_exam_PI/5.2 Hello$ sudo nano Hello.cpp
```

怎么样，这个打开之后，代码还是很平易近人的吧。写过单片机程序的人都不陌生，别嫌这个太简单，只能说是世面见得少。写程序嘛，从来都是不能浮躁的。



方向键移动，直接打字就好了，蛮好用的，相对 vim，这样的编辑效率很低

```
#include<stdio.h>

int main(void)
{
    printf("Hello! Mei Zi! \n");
    return 0;
}

//end main
```

修改好了看下方的菜单，按下 ctrl+x 保存
询问是否确定

```
Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
^Y Yes
^N No      ^C Cancel
```

看菜单，按 y 保存。当然，不保存和取消分别就是 ctrl+n 与 ctrl+c 了询问是否改名，咱就不改了，直接按回车

```
File Name to Write: Hello.cpp
^G Get Help      M-D DOS Format      M-A Append          M-B Backup File
^C Cancel        M-M Mac Format      M-P Prepend
```

第六课 • 编译程序

程序写好了，接下来就是编译了

打开 build 文件夹

```
pi@grasp: ~/working/Book_exam_PI/5.2 Hello $ cd build/
```

里面的 Makefile 就是编译的设置了。因为我们编译的文件还是 Hello.cpp，也没有调用什么大的库，所以编译配置（Makefile）就不用改了。想要深入研究的请参考配套的红皮书上 P74 及以后的内容。

```
pi@grasp: ~/working/Book_exam_PI/5.2 Hello/build $ ls
CMakeCache.txt  CMakeFiles  cmake_install.cmake  Hello  Makefile
```

直接在这个地方输入 make

看到这个就是编译成功了，有错误的根据报错提示自己改去。

```
pi@grasp: ~/working/Book_exam_PI/5.2 Hello/build $ make
Scanning dependencies of target Hello
[100%] Building CXX object CMakeFiles/Hello.dir/Hello.cpp.o
Linking CXX executable Hello
[100%] Built target Hello
```

再次运行

Hello! Mei Zi!出现了，至此成功！

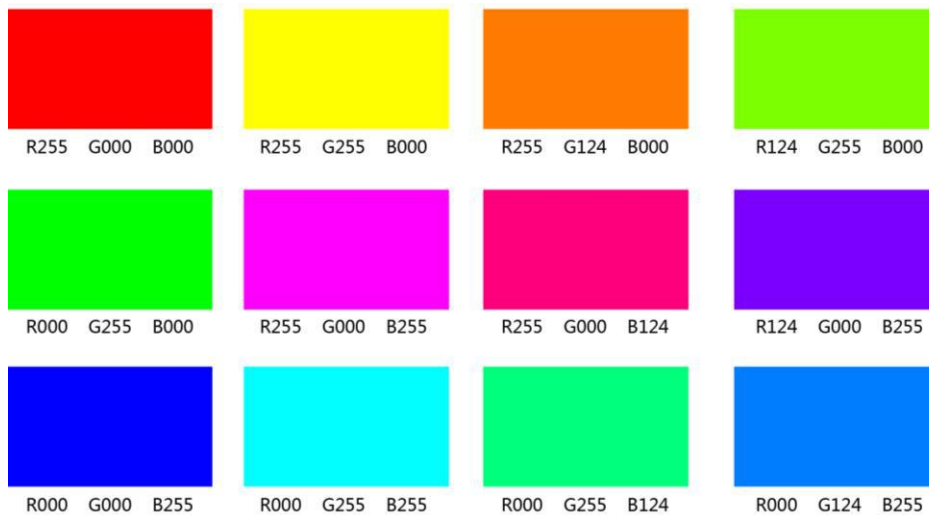
树莓派机器人踢球程序

第七课 • 踢球程序

接下来我们运行踢球程序

```
pi@grasp:~$ cd working/kickBall/build/
pi@grasp:~/working/kickBall/build$ ls
CMakeCache.txt CMakeFiles cmake_install.cmake Makefile MOT
pi@grasp:~/working/kickBall/build$ ./MOT
Xlib: extension "RANDR" missing on display ":1.0".
```

颜色测试图片



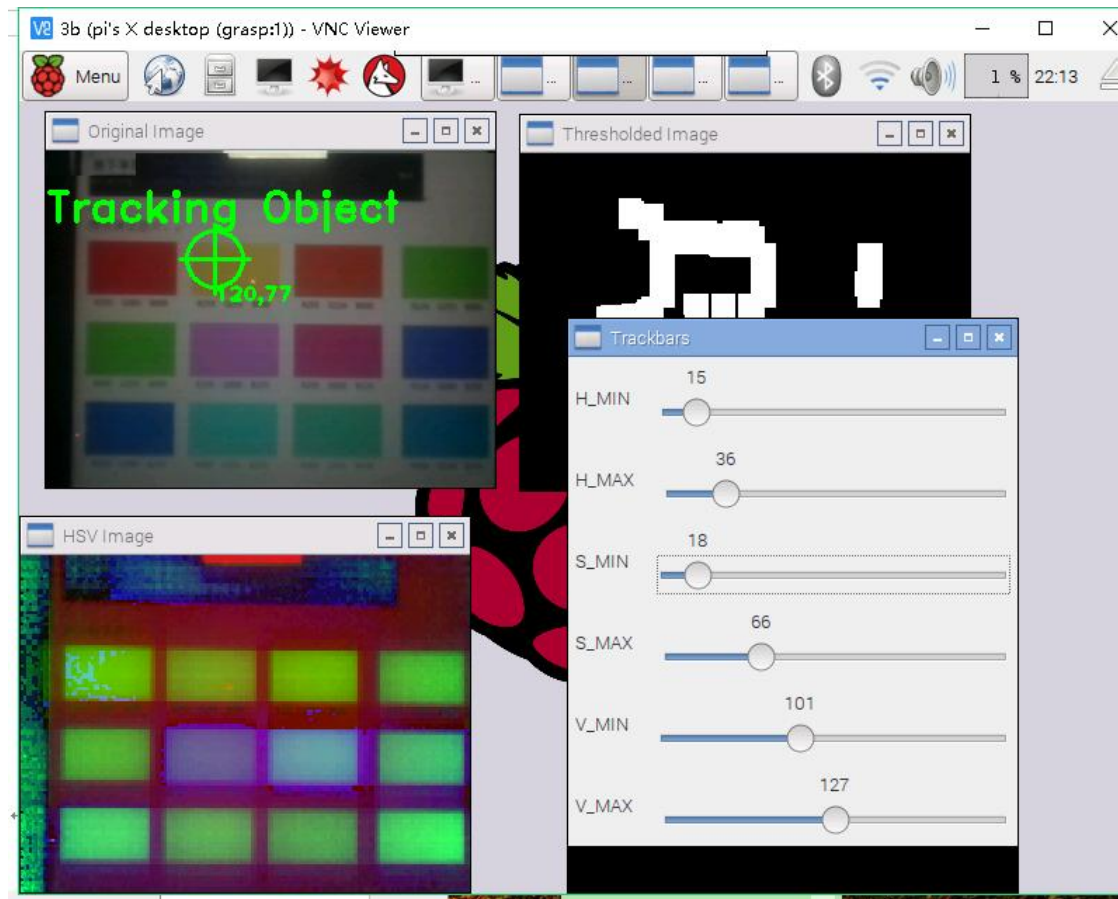
树莓派采集到的图像



一开始我也不会用这个程序，通过查阅它的源代码（objectTracer.cpp），发现还有个菜单可以调节参数，在 MOT 窗口下按下 s 即可调出菜单。

```
pi@grasp:~$ cd working/kickBall/
pi@grasp:~/working/kickBall$ ls
build CMakeLists.txt objectTracer.cpp uart_api.cpp uart_api.h
```

```
if (key == 's') calibrationMode = !calibrationMode;//if user presses 's', toggle calibration mode
```



菜单包括二值化图像，HSV 域图像，原始图像和调节菜单。

摄像头对准颜色鲜艳的目标，调节合适了就可以追踪到颜色对象，并通过串口给底板发对象位置数据，驱动机器人追逐对象。

你得到的结果和我给的图片有细微差别也是正常的，毕竟这些程序我有些微调测试过。

源代码也是用的 C 类语言，蛮好的，下面可以得到源代码的一些重要参数。

```
int main(int argc, char** argv){

    objectTracerClass *objectTC = new objectTracerClass();
    user_uart_open("ttyAMA0");
    user_uart_config(115200);
    namedWindow(windowName);
    //set mouse callback function to be active on "Webcam Feed" window
    //we pass the handle to our "frame" matrix so that we can draw a rectangle to it
    //as the user clicks and drags the mouse
    cv::setMouseCallback(windowName, clickAndDrag_Rectangle, &cameraFeed);
    //initiate mouse move and drag to false
    mouseIsDragging = false;
    mouseMove = false;
    rectangleSelected = false;

    objectTC->imageProcess();
    return 0;
}
```

运行这个函数的时候会占用树莓派的串口，串口 SSH 什么的就用不了了。

你会发现按 MOT 的关闭按钮是关不了的，没事，这是我们在终端里面启动的程序，在终端里面直接按下 ctrl + c 就可以停止程序了。

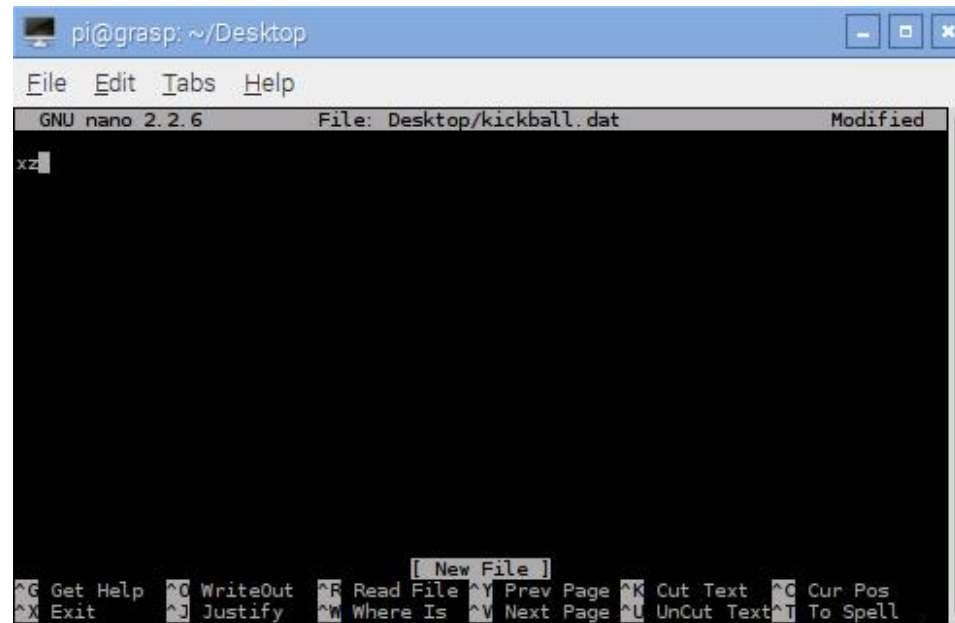
第八课 • 自动运行脚本

下面我们写个脚本，点一下就可以运行：

打开终端，输入：

```
pi@grasp: ~ $ sudo nano Desktop/kickball.dat
```

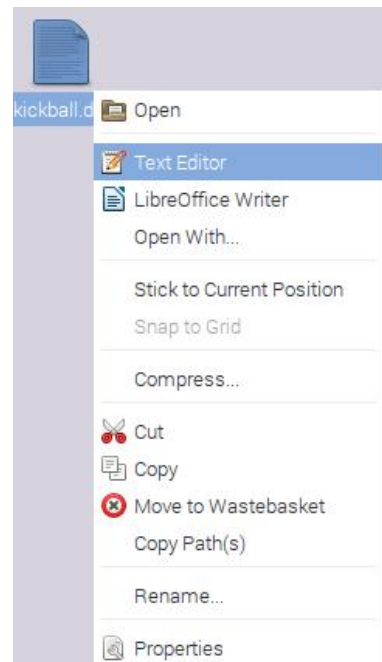
打开后随便输入一些东西，保存后退出



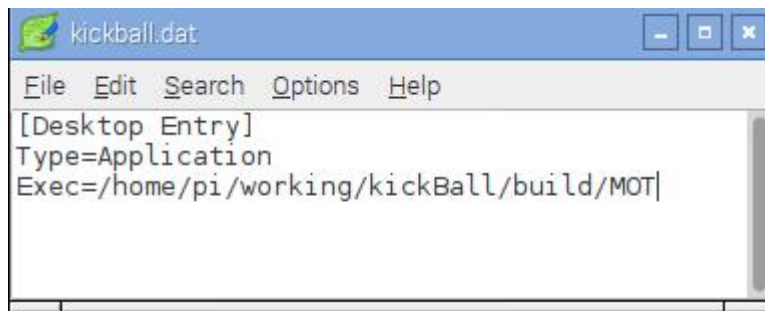
这样就在桌面上建立了一个 **dat** 脚本



右击选择文本编辑器，进行编辑



输入如下内容：



ps :也可以直接在 nano 里面输入，一样的。

Ctrl + s 保存后退出,之后双机鼠标即可自动运行 MOT

你会发现又关不掉了。。。。。。请看第九课的内容

第九课 • 系统进程处理

打开终端，输入如下内容，查看系统进程：

```
pi@grasp: ~ $ ps -ef|grep kick
pi      1160      1  1 20:18 ?        00:00:01 /home/pi/working/kickBall/build/
MOT
pi      1183  1172  0 20:20 pts/0    00:00:00 grep --color=auto kick
pi@grasp: ~ $ kill 1160
```

ps -ef 表示列出所有进程，|表示同时代入后面的参数 grep kick 表示筛选出 kick 内容的进程，我这里进程号是 1160，输入 kill 1160 结束对应进程即可。

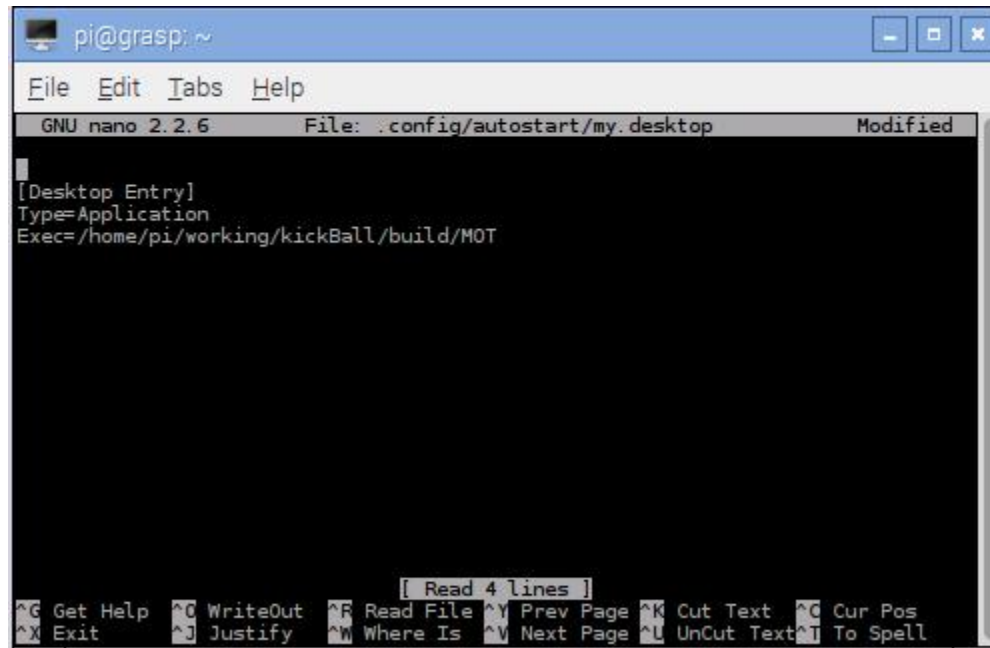
第十课 • 系统开机自动运行

上面虽然可以鼠标点一下就可以启动，但是还是不够自动，下面我们设置一下，开机后马上启动 MOT，开机自启动的方式有很多，这里我介绍一种：

终端输入：

```
pi@grasp:~$ sudo nano .config/autostart/my.desktop
```

输入如下代码：



The screenshot shows a terminal window titled 'pi@grasp:~'. The nano text editor is open, editing the file '.config/autostart/my.desktop'. The editor's status bar at the top indicates 'GNU nano 2.2.6' and 'File: .config/autostart/my.desktop'. The content of the file is as follows:

```
[Desktop Entry]
Type=Application
Exec=/home/pi/working/kickBall/build/MOT
```

The bottom of the terminal shows the nano editor's command palette with various shortcuts like '^G Get Help', '^O WriteOut', '^F Read File', etc.

保存退出

终端输入 `sudo reboot` 即可重启，重启后自动运行 MOT

如果不想开机自动启动了。

打开刚才编辑的 `my.Desktop`，前面都加上 `#` 号，注释掉这些代码即可。

```
#[Desktop Entry]
#Type=Application
#Exec=/home/pi/working/kickBall/build/MOT
```


附录

配置 VNC

VNC(Virtual Network Console)是虚拟网络控制台，在树莓派上面开启了 VNC，连接到路由器，那么连接在同一路由器上的电脑就可以远程桌面控制树莓派。

总之有了 VNC，树莓派只要供电就行，再也不用把鼠标、键盘和显示器插到树莓派上了。

简要安装介绍：[网上的树莓派 VNC 教程](#)（网上教程和我的教程两个参考着一起看，开机启动 VNC 网上教程里面有）

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install tightvncserver
```

（过会儿出现一个选项，选 y，按回车，之后等着，大约 5 分钟之内装完）

```
vncpasswd
```

```
passwd:xxxxxxx
```

```
passwd:xxxxxxx
```

出现 view-only 模式密码设置，选择 n，按回车

关于 VNC 无法复制和粘贴，可以度娘网上的方法，基本只有几种方法。不过我试下来还是不行，只好 VNC 配合 WINSXP 或 SSH 一起用了，记得先设置好 root 的密码和相关配置用以 SSH 登录和 SCP 登录。

HDMI 强制输出设置可以调整 VNC 分辨率，自己度娘去，网上教程很多的。

建议

usb 转 ttl 转换器：我只推荐 24I01 芯片的 USB-TTL，其他 PL2303、FT232 等渣渣芯片懒得说。驱动自己度娘去，别懒，这种驱动什么的超级多的。

树莓派终端常用命令

clear 可以清除屏幕

cd 打开目录

cd..返回上级目录

./执行可执行文件

ls 查看目录

ls -a 查看当前目录所有文件

ls -all 查看当前目录文件详细信息

nano filename 打开文件

vi filename 打开文件

nano 和 vi/vim 都是很厉害的编辑器，用来编辑各种文档，vi 用熟了可以全键盘编辑文档，比什么 word 记笔记高效多了。

Kill (PID)结束进程

```
sudo halt
```

 关机

```
sudo /etc/init.d/networking restart
```

 重启网络

```
sudo reboot
```

 重启

FAQ

1. `make[2]: Warning: File 'MOT' has modification time 12054 s in the future`
`make[2]: warning: Clock skew detected. Your build may be incomplete.`
系统在编译时发现文件日期超前!常发生在交叉编译的情况下。需要调整文件修改时间，系统时间，和电脑时间。
- 2、无线情况下 SSH 突然掉线，连不上：干等一会儿就好。。。
- 3、开机一直停在命令行界面，且命令行显示 `Ext4-fs error.....block109210.....bad entry indictionary...`。出错什么的：问题不常见，可能是 TF 卡本身的问题，有几率可以手动修复，具体方法请自行参考度娘，蛮麻烦的，我搭建的系统用了好久，忘了备份镜像，有次突然开机就出现 `block error` 什么的，一查是 TF 卡问题，可能是 TF 卡有个存储块坏了，用 `ubuntu` 修了一遍，没有任何效果，所以，建议还是换一张卡重来吧。
- 4、无法开机或者系统运行不稳定，请移除外设，排除供电不足的可能。

历史版本

- [1].2017/2/1.第一版 V1.0.刘亚龙.江苏-苏州;
- [2].2017/2/14.第二版 V1.1 刘亚龙.江苏-苏州;
- [3].2017/10/24.第三版 V1.2 刘亚龙.西安-西安电子科技大学;
- [4] 2020/01/02.不忘初心第四版 V1.3 刘亚龙.西安-西安电子科技大学;

注意：所有资料仅供个人学习，本文档版权归刘亚龙所有，代码版权归 HBE-ROBONOVA-AI II 的生产公司所有，请勿转发！