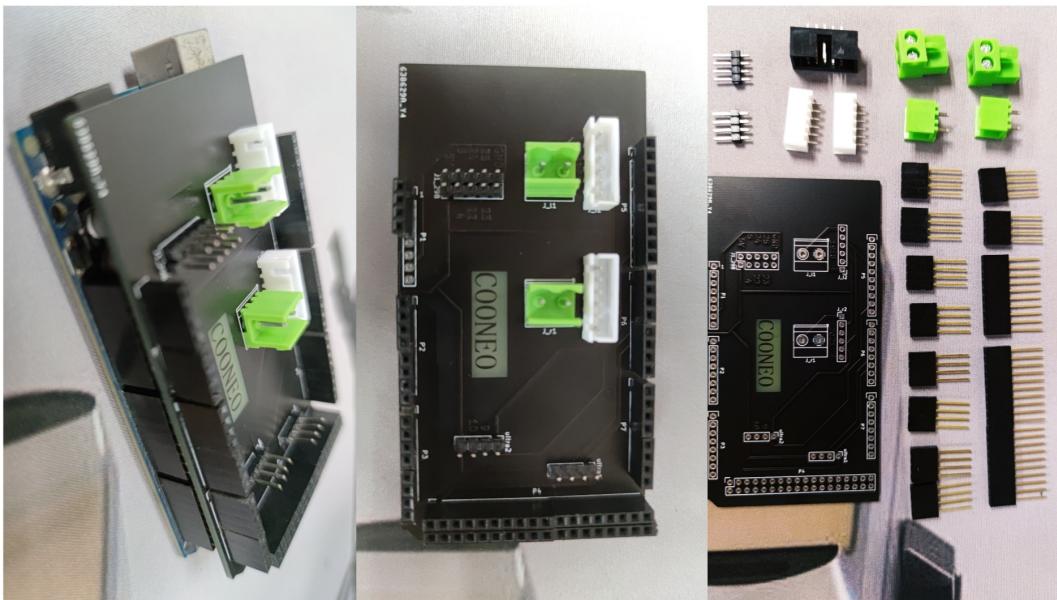


# Arduino\_Raspberry\_ROS\_Car Tutorials



## chapter 1: Construction A ROS Car

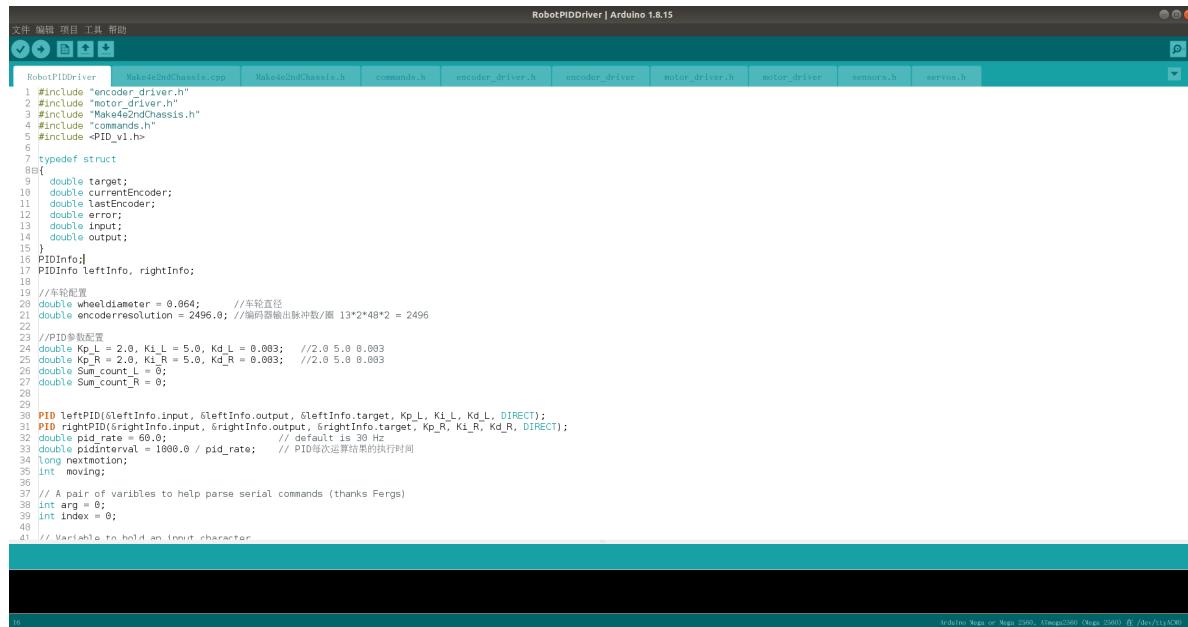


Download codes from Git:

```
git clone https://github.com/COONEO/Arduino_Raspberry_ROS_Car.git
```

# Step One: Download programs for Arduino Mega 2560

Install Arduino IDE in your computer and add library where in Arduino\_mega\_2560\_code/relative\_library folder.then,download the code into your Arduino\_mega\_2560 board.



The screenshot shows the Arduino IDE interface with the title "RobotPIDDriver | Arduino 1.8.15". The code editor displays the "RobotPIDDriver.ino" file. The code is a C++ program for a robot using a PID controller. It includes headers for encoder, motor driver, commands, and sensors. It defines a PIDInfo struct and initializes variables like wheel diameter, encoder resolution, and PID parameters (Kp, Ki, Kd) for both left and right motors. It also handles serial command parsing and movement logic. The status bar at the bottom indicates "Arduino Mega or Mega 2560, Atmega2560 (Mega 2560) (F) / (S) (L) (H)".

```
RobotPIDDriver | Arduino 1.8.15
文件 编辑 项目 工具 帮助
RobotPIDDriver RobotPIDDriver.ino MakefileChassis.h Commands.h encoder_driver.h encoder_driver.h motor_driver.h motor_driver.h sensors.h servos.h
1 #include "encoder_driver.h"
2 #include "motor_driver.h"
3 #include "Make4x2ndChassis.h"
4 #include "commands.h"
5 #include <PID_v1.h>
6
7 typedef struct
8{
9    double target;
10   double currentEncoder;
11   double lastEncoder;
12   double error;
13   double input;
14   double output;
15 }
16 PIDInfo;
17 PIDInfo leftInfo, rightInfo;
18
19 //车轮配置
20 double wheelDiameter = 0.064; //车轮直径
21 double encoderResolution = 2496.0; //编码器输出脉冲数/圈 13*2*48*2 = 2496
22
23 //PID参数配置
24 double Kp_L = 2.0, Ki_L = 5.0, Kd_L = 0.003; //2.0 5.0 0.003
25 double Kp_R = 2.0, Ki_R = 5.0, Kd_R = 0.003; //2.0 5.0 0.003
26 double sum_count_L = 0;
27 double sum_count_R = 0;
28
29
30 PID leftPID(leftInfo.input, leftInfo.output, leftInfo.target, Kp_L, Ki_L, Kd_L, DIRECT);
31 PID rightPID(rightInfo.input, rightInfo.output, rightInfo.target, Kp_R, Ki_R, Kd_R, DIRECT);
32 double pid_rate = 60.0; // default is 30 Hz
33 double pidInterval = 1000.0 / pid_rate; // PID每次运算需要的执行时间
34 long nextMotion;
35 int moving;
36
37 // A pair of variables to help parse serial commands (thanks Fergs)
38 int arg = 0;
39 int index = 0;
40
41 // Variable to hold an input character
```

ps:新增了适配我们Arduino扩展板的程序，见文件夹“**RobotPIDDriver\_tb6612**”。配置过程参见如下的两篇微信公众号推文。

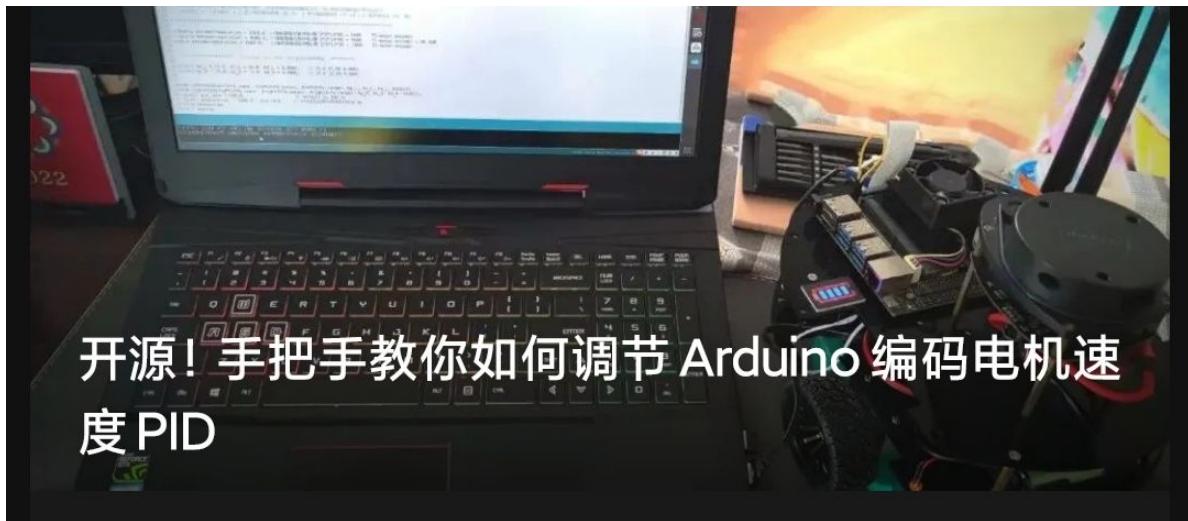
2022/1/1 17:09

...

开源！手把手教你驱动Arduino+ROS小车的电机 原创



264 3 1

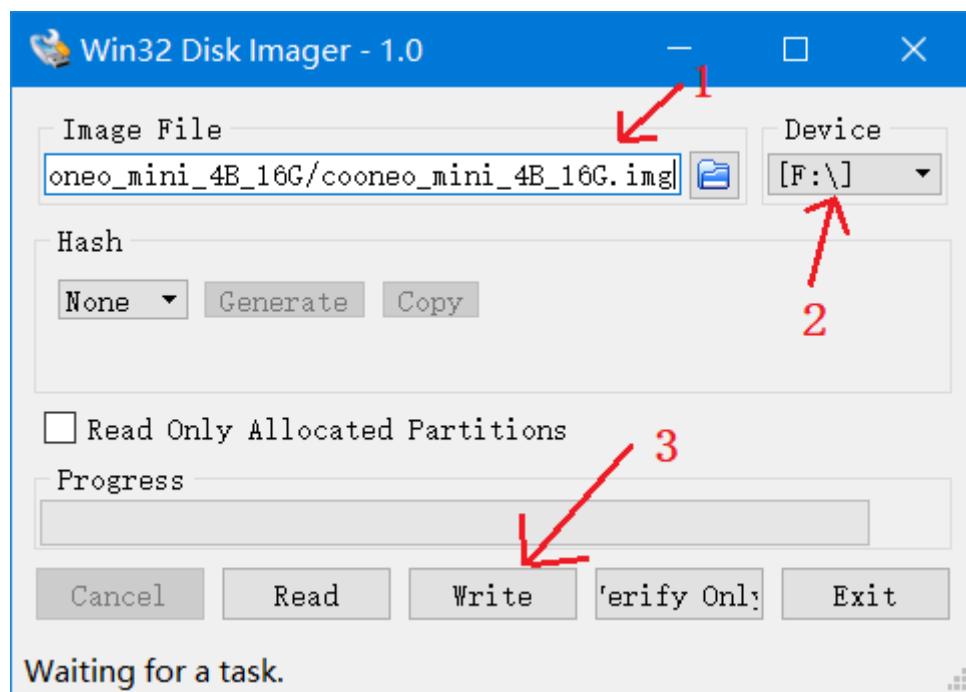


开源！手把手教你如何调节 Arduino 编码电机速度 PID

## Step Two: FLASH OS and LAUNCH ROS NODE

1. flash Ubuntu OS into your Pi 4B board (By Win32DiskImager.exe)

The OS img can be finding in our **Wechat Official Account COONEO**. Process like this:



2. launch ROS node in Raspberry Pi

```
#connect Raspberry Pi and Arduino
sudo chmod 0777 /dev/ttyACM0

#open a Terminal && download codes
git clone https://github.com/COONEO/Arduino_Raspberry_ROS_Car.git

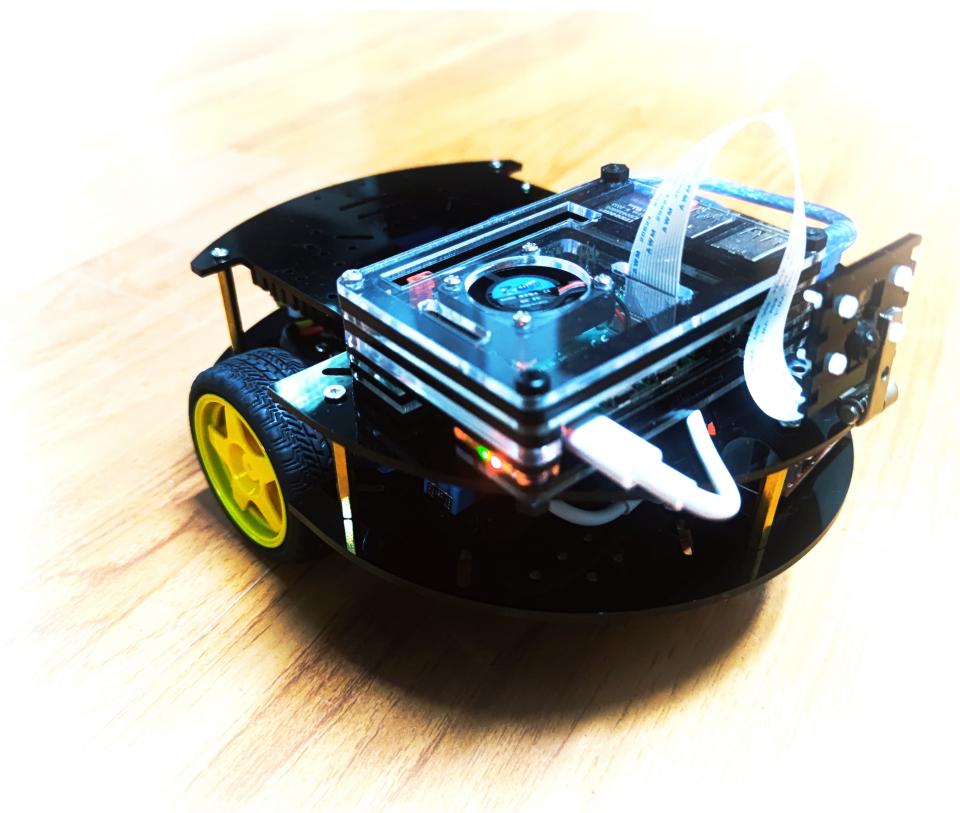
# copy ROS node in your home folder
cp -r Arduino_Raspberry_ROS_car/Raspberry_Pi_ROS_Node/catkin_ws ~/

# change *.py file's permission
```

```
sudo chmod 0777
Arduino_Raspberry_ROS_car/Raspberry_Pi_ROS_Node/catkin_ws/src/ros_arduino_bridge/ros_arduino_python/src/ros_arduino_python/*
sudo chmod 0777
Arduino_Raspberry_ROS_car/Raspberry_Pi_ROS_Node/catkin_ws/src/ros_arduino_bridge/ros_arduino_python/nodes/arduino_node.py

cd catkin_ws
catkin_make
source devel/setup.bash
roslaunch ros_arduino_python arduino.launch
```

In the end, you can publish Topic "cmd\_vel" msg to control ROS car running.

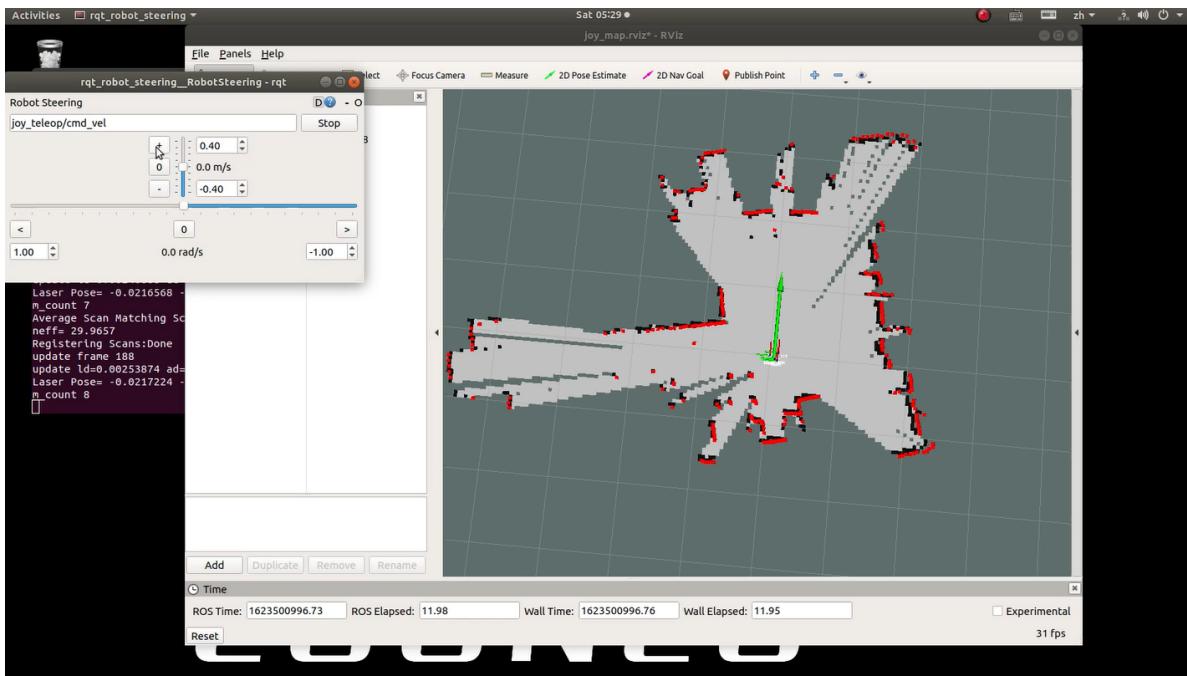


## chapter 2: Gmapping with Arduino\_Raspberry\_Car

Step 1 : launch gmapping launch file and watching.

```
# open a Terminal
cd catkin_ws
source devel/setup.bash
roslaunch launch_file gmapping_ekf.launch

# open a Rviz && Visual a map
rosrun rviz rviz
```



Step 2 : save the map

```
# open a Terminal
# cd in your folder,P.S.
cd catkin_ws/src/launch_file/map/
rosrun map_server map_saver -f your_map_name
```

and then, the folder will create two files, they are **your\_map\_name.pgm** && **your\_map\_name.yaml** file.

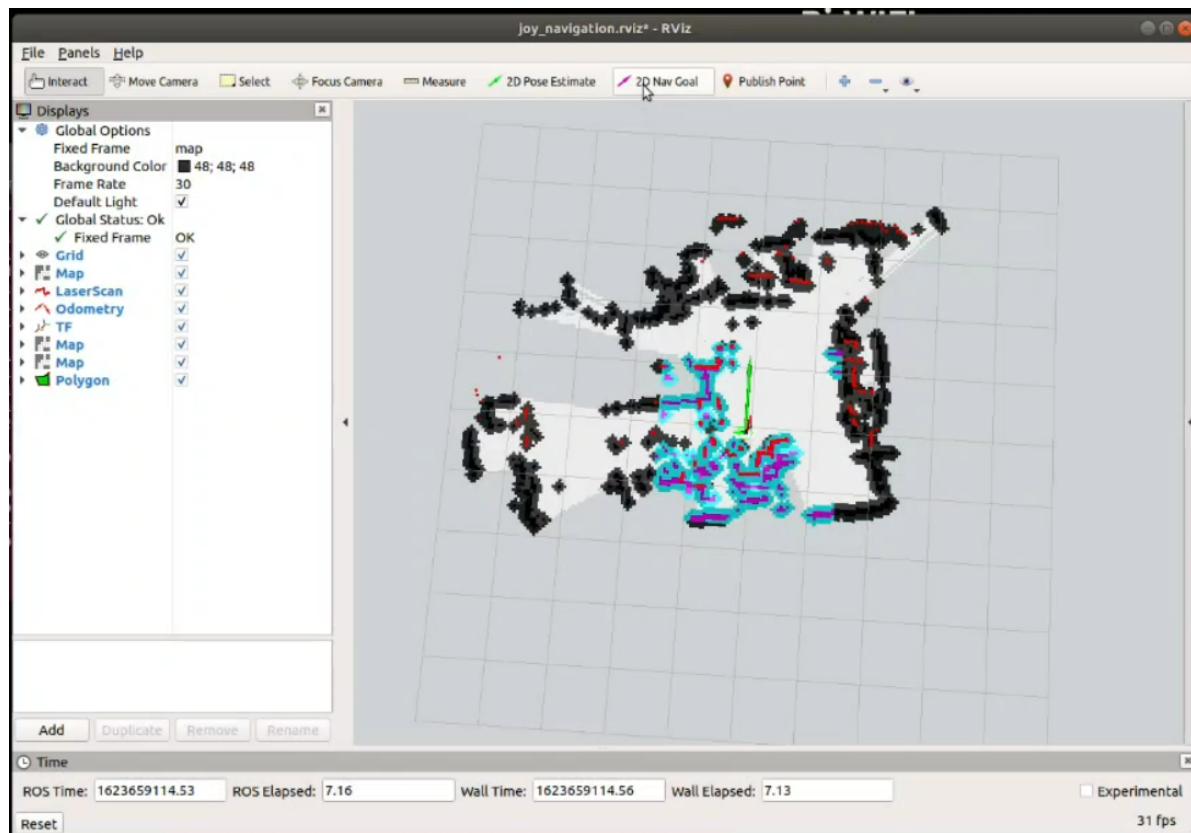
## chapter 3 : Run ROS navigation stack

Step 1 : launch navigation\_ekf.launch file.

```
# open a Terminal
cd catkin_ws
source devel/setup.bash
roslaunch launch_file navigation_ekf.launch

# and open another Terminal
rosrun rviz rviz
```

select topics like this in Rviz.

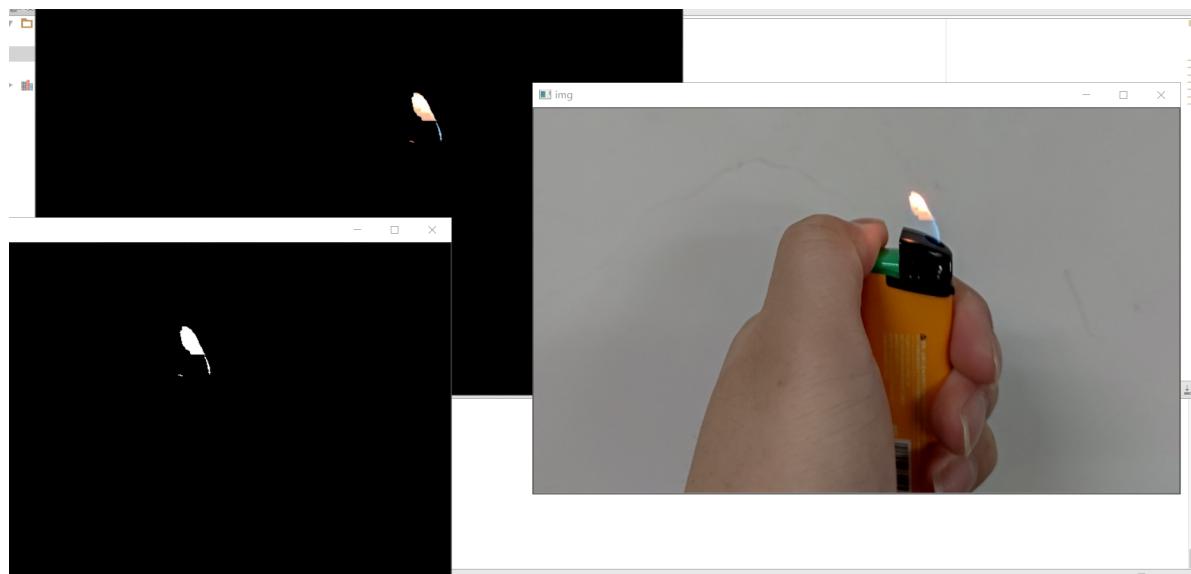


and then, click the "2D Nav Goal" button to select a Goal pose. for details, please see the "demo\_videos/03\_ROS\_Navigation\_function.mp4" file.

## Chapter 4 : Fire detect

```
# open a Terminal
cd catkin_ws
source devel/setup.bash
fire_detect_cpp.launch #or "fire_detect.launch"

# and open another Terminal
rosrun rviz rviz
```



for more details,please see the "demo\_videos/04\_Fire\_detect\_based\_on\_color.mp4" file. or search the "COONEO" in your Bilibili.

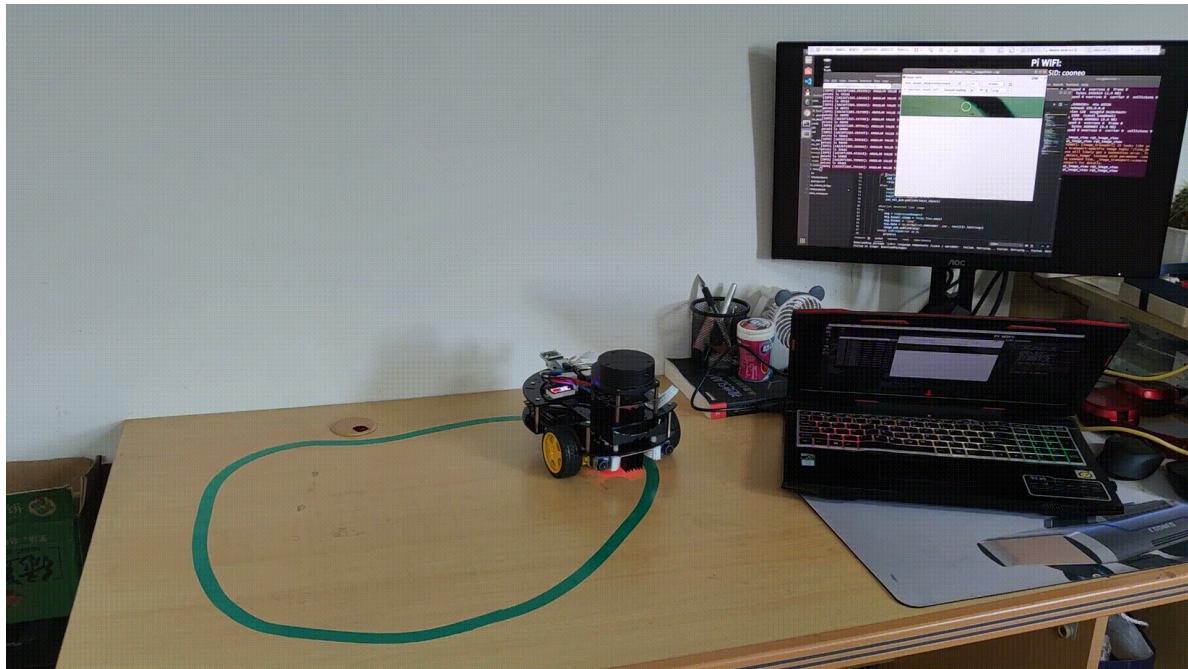
## chapter 5 : automatic following the line

Step 1 : launch the ros\_arduino\_bridge node.

```
# open a terminal  
cd catkin_ws  
source devel/setup.bash  
  
roslaunch ros_arduino_python arduino.launch
```

Step 2 : launch the line\_track launch file.

```
# open another terminal  
cd catkin_ws  
source devel/setup.bash  
roslaunch line_track linetrack_red.launch
```



2021.09.25

author:ZhaoXiang Lee

COONEO Co.,Ltd

Web:<http://cooneo.cc>

E: [cooneo@outlook.com](mailto:cooneo@outlook.com)

For more details,you can search "COONEO" in your WeChat.



微信搜一搜

COONEO

or search "COONEO" in Bilibili.



511  
粉丝

2  
关注

453  
获赞

特别关注



COONEO 🔴

微博/公众号: COONEO

详情