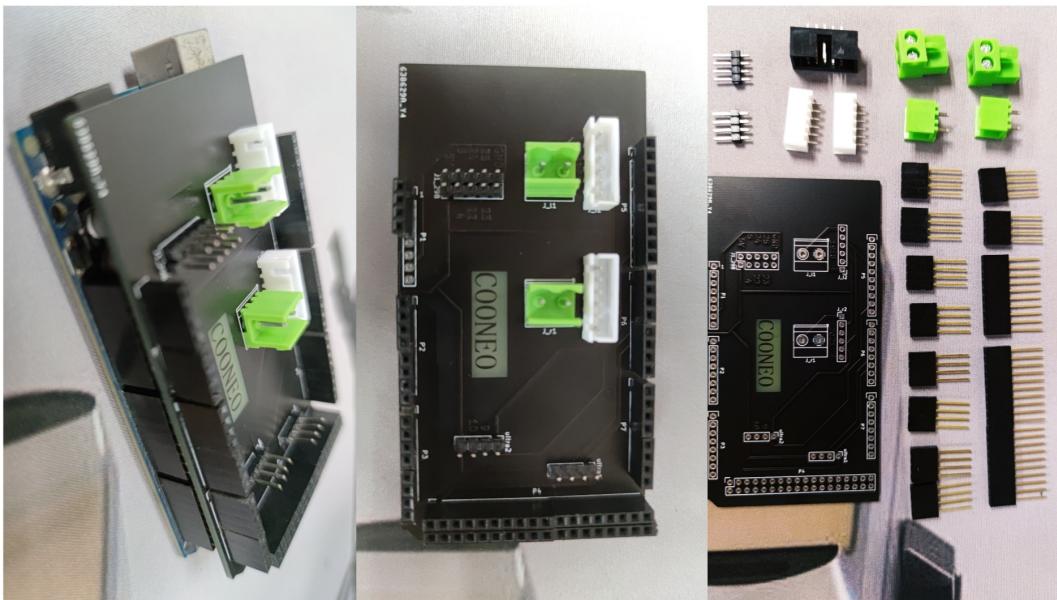


Arduino_Raspberry_ROS_Car Tutorials



chapter 1: Construction A ROS Car

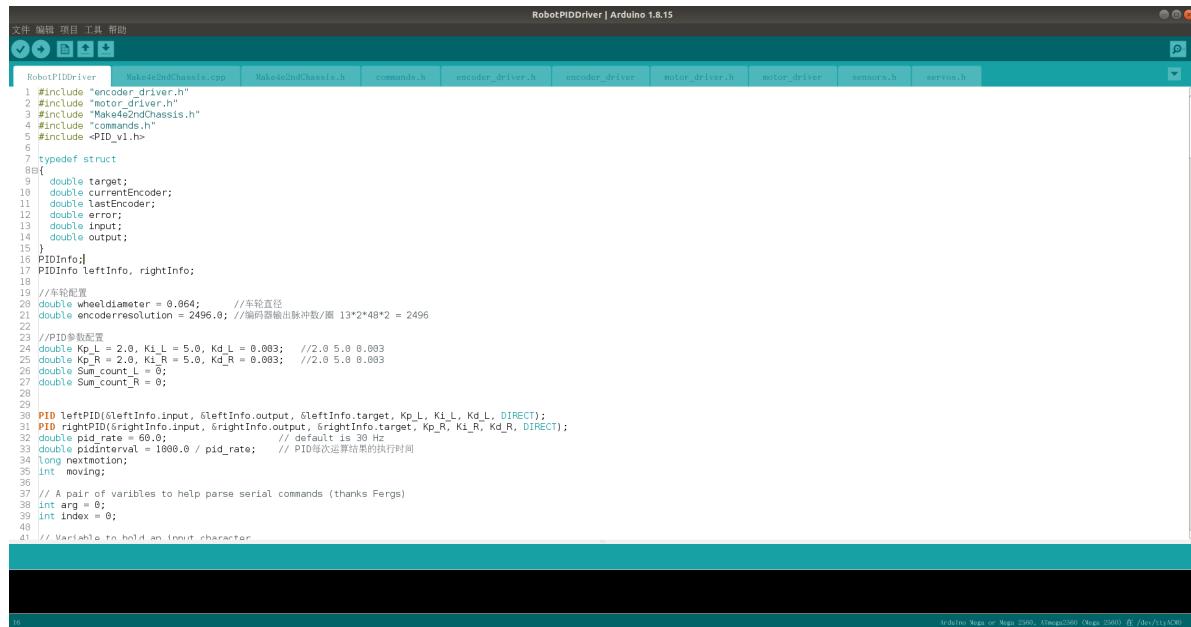


Download codes from Git:

```
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 'RobotPIDDriver' sketch open. The code is a C++ program for a robot using a PID controller. It includes headers for encoder, motor driver, commands, and PID. The main loop handles left and right wheel PID calculations based on target and current encoder values, with a fixed wheel diameter of 0.064 and an encoder resolution of 2496. It also manages serial command parsing and movement. The code is well-organized with comments explaining variables and functions.

```
#include "encoder_driver.h"
#include "motor_driver.h"
#include "Make4x2ndChassis.h"
#include "commands.h"
#include <PID_v1.h>

typedef struct
{
    double target;
    double currentEncoder;
    double lastEncoder;
    double error;
    double input;
    double output;
} PIDInfo;

PIDInfo leftInfo, rightInfo;

//车轮配置
double wheeldiameter = 0.064; //车轮直径
double encoderresolution = 2496.0; //编码器输出脉冲数/圈 13*2*48*2 = 2496

//PID参数配置
double Kp_L = 2.0, Ki_L = 5.0, Kd_L = 0.003; //2.0 5.0 0.003
double Kp_R = 2.0, Ki_R = 5.0, Kd_R = 0.003; //2.0 5.0 0.003
double Sum_count_L = 0;
double Sum_count_R = 0;

PID(leftPID, &leftInfo.input, &leftInfo.output, &leftInfo.target, Kp_L, Ki_L, Kd_L, DIRECT);
PID(rightPID, &rightInfo.input, &rightInfo.output, &rightInfo.target, Kp_R, Ki_R, Kd_R, DIRECT);

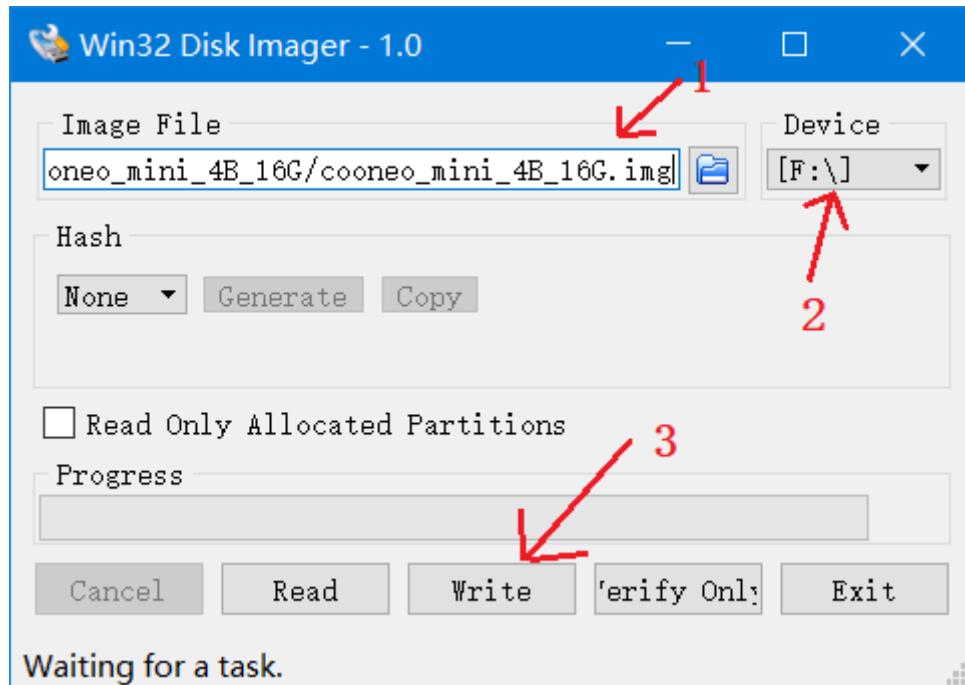
double pid_rate = 60.0; // default is 30 Hz
double pidinterval = 1000.0 / pid_rate; // PID每次运算消耗的执行时间
long nextmotion;
int in_moving;

// A pair of variables to help parse serial commands (thanks Fergs)
int arg = 0;
int index = 0;
char inChar;
```

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

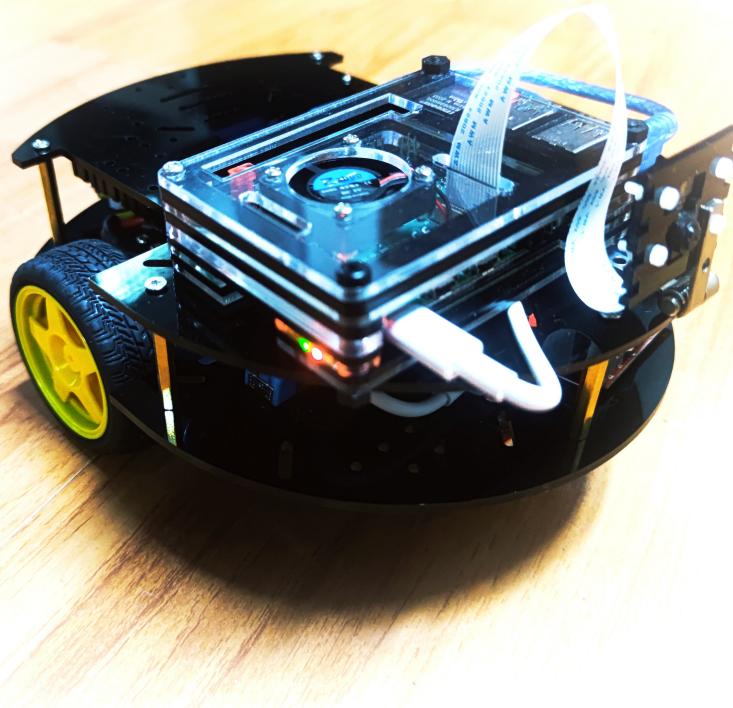
# copy ROS node in your home folder
cp -r Arduino_Raspberry_ROS_car/Raspberry_Pi_ROS_Node/catkin_ws ~/

# change *.py file's permission
sudo chmod 0777
Arduino_Raspberry_ROS_car/Raspberry_Pi_ROS_Node/catkin_ws/src/ros_arduino_bridge/ros_ard
uino_python/src/ros_arduino_python/*
sudo chmod 0777
Arduino_Raspberry_ROS_car/Raspberry_Pi_ROS_Node/catkin_ws/src/ros_arduino_bridge/ros_ard
uino_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.

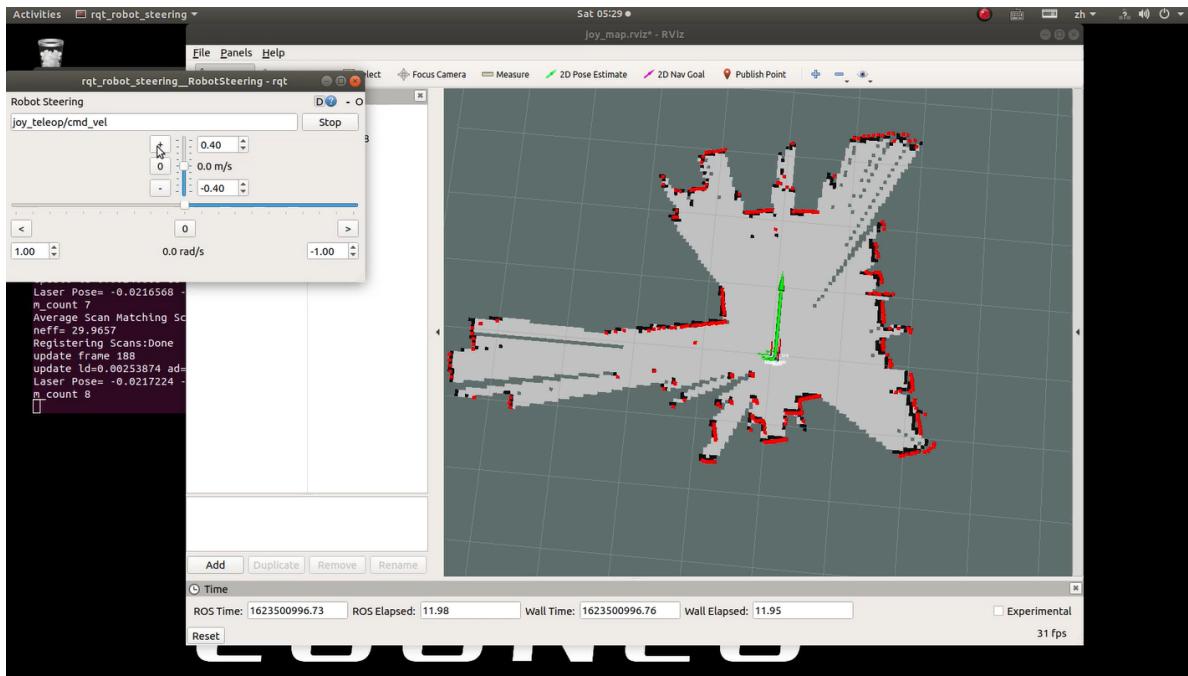


chapter 2: Gmapping with Arduino_Raspnerry_Car

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.

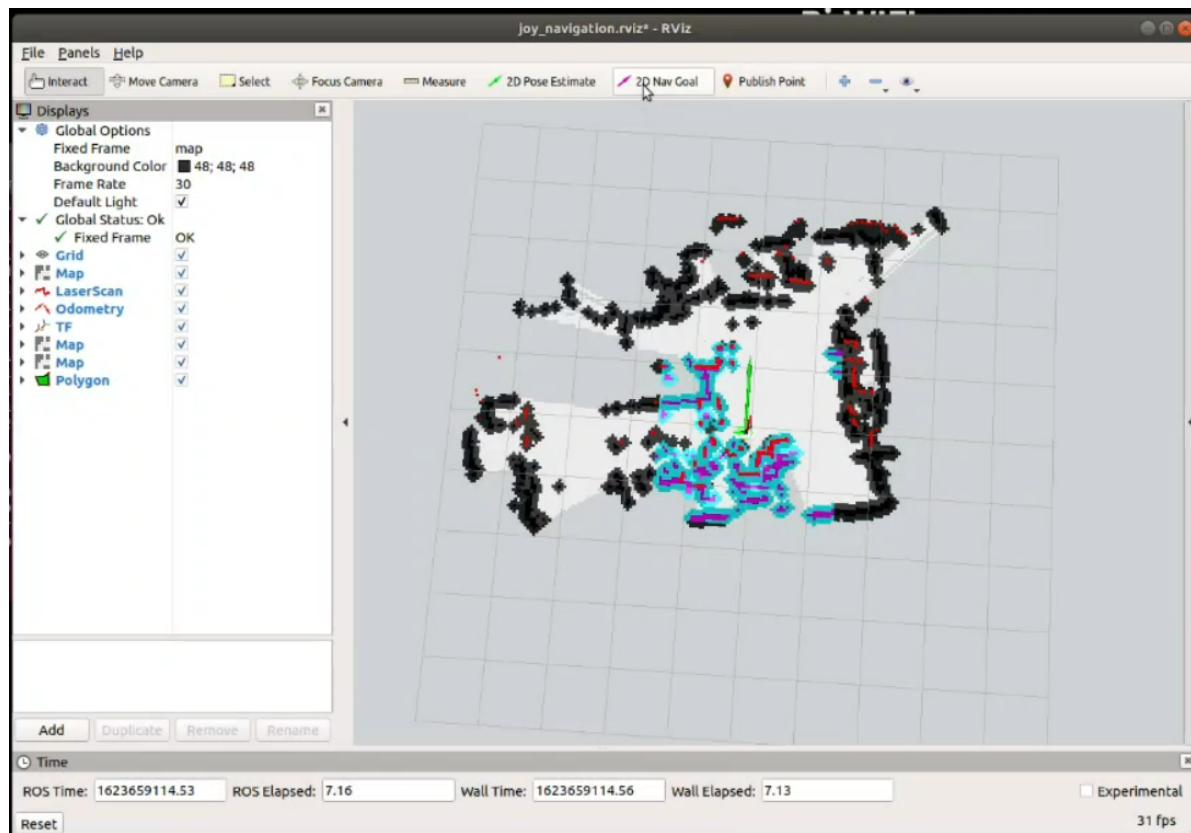
chapter 3 : Run ROS navigation stack

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.

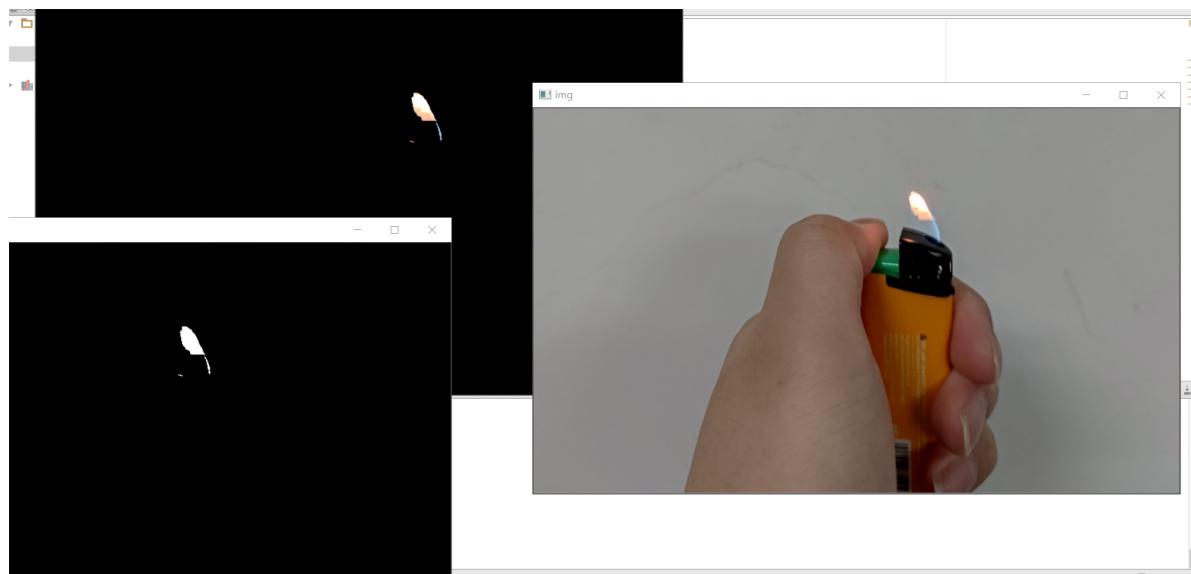


and then, click the "2D Nav Goal" button to select a Goal pose. for details, please see the "demo_videos/03_ROS_Navigation_function.mp4" file.

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.

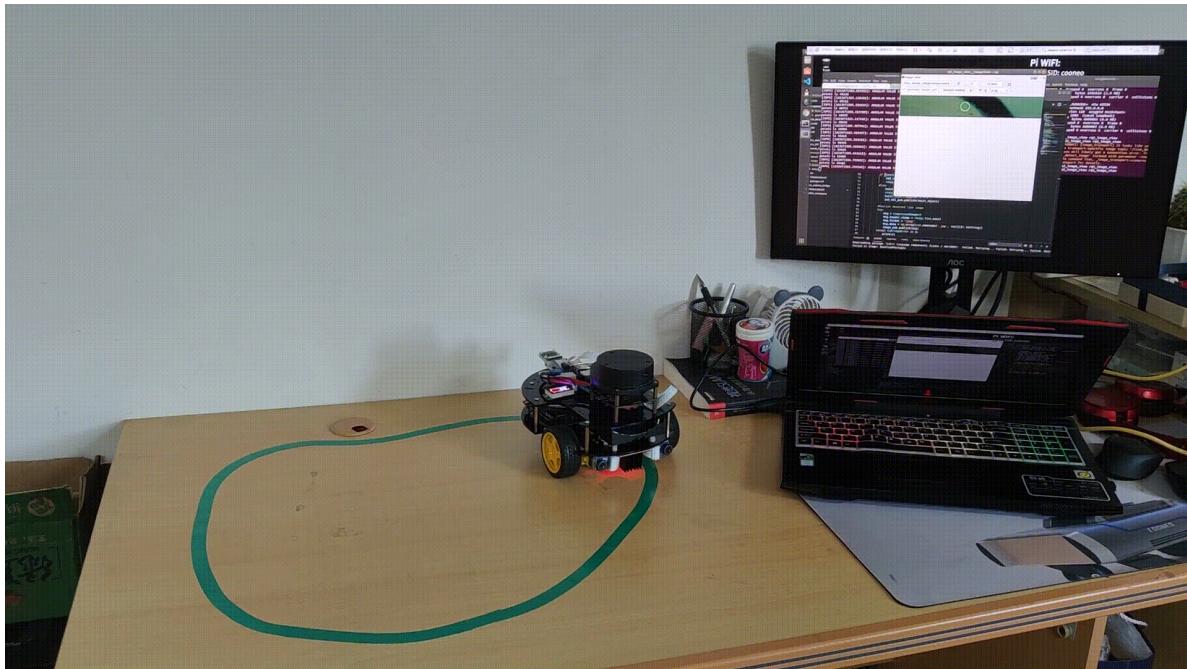
chapter 5 : automatic following the line

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

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](#)