

CARLA-ROS 联合仿真环境搭建

我的电脑配置如下：

型号: 惠普暗影精灵 4 Plus

CPU: i7-8750H

GPU: GTX 1070 (GPU 驱动版本为 470)

RAM: 16G

系统: Ubuntu 20

1 配置 CARLA 运行环境

1.1 环境要求

安装 20.3 以上版本的 `pip`

```
# 安装 pip
sudo apt install python3-pip

# 确定 pip 的版本
pip3 -V

# 若版本不符合要求，需要升级 pip
pip3 install --upgrade pip
```

安装 `pygame` && `numpy` && `opencv`

```
pip3 install --user pygame numpy
pip3 install python-opencv
```

若 `pip install` 安装完报错 `WARNING: XXX is not on PATH`，需要在 `~/.bashrc` 的最后一行添加
`export PATH=/home/<username>/.local/bin/:$PATH`

```
sudo gedit ~/.bashrc

# username 需要更换为自己的用户名
export PATH=/home/<username>/.local/bin/:$PATH
```

保存并关闭 `gedit` 页面后，需要重新刷新环境变量使设置生效：

```
source ~/.bashrc
```

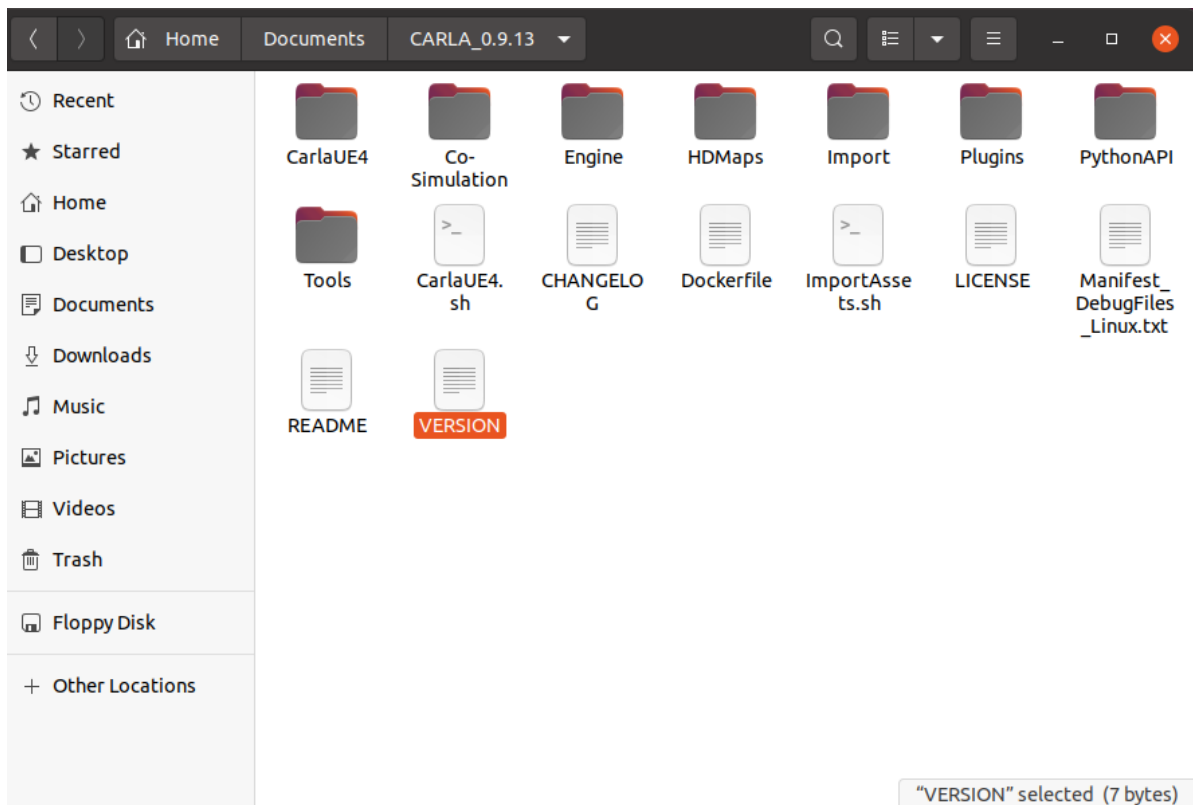
```
Open ▾ [icon] *.bashrc Save [icon] [icon] [icon] [icon]
88 #export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'
89
90 # some more ls aliases
91 alias ll='ls -alF'
92 alias la='ls -A'
93 alias l='ls -CF'
94
95 # Add an "alert" alias for long running commands.  Use like so:
96 #   sleep 10; alert
97 alias alert='notify-send --urgency=low -i "${? = 0 } && echo terminal || echo error)" "${-
(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[\;|]\s*alert$//'\`}'"
98
99 # Alias definitions.
100 # You may want to put all your additions into a separate file like
101 # ~/.bash_aliases, instead of adding them here directly.
102 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
103
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118
119 # >>> fishros initialize >>>
120 source /opt/ros/foxy/setup.bash
121 # <<< fishros initialize <<<
122
123 export PATH=/home/qrq/.local/bin/:$PATH
```

添加完成后需要刷新环境变量:

```
source ~/.bashrc
```

1.2 安装 CARLA

从 Github 下载 [CARLA 0.9.13](#) 并解压即可



1.3 运行 CARLA

为了顺利运行客户端，需要先添加 Carla client library

```
pip3 install carla
```

一切准备就绪后，便可以运行 Carla 了

```
# 尖括号需要修改为自己电脑上的路径，例如 cd /home/qrq/Carla
cd <path-to-carla>

# 首次运行 carla 需要先导入配置文件
./ImportAssets.sh

# 若要以低分辨率运行，可以使用 ./CarlaUE4.sh -quality-level=Low
./CarlaUE4.sh
```

此时可能会报错，需要安装一下 `libomp5`

```
sudo apt install libomp5
```

2 安装 carla_ros_bridge

2.1 ROS 2

carla_ros_bridge 对 ros 2 版本的要求比较刁钻，必须是 ROS 2 Foxy，其他版本可能会有些莫名其妙的问题。

ROS 2 的安装可以借助鱼香 ROS 的脚本一键安装：

```
wget http://fishros.com/install -O fishros && . fishros
```

安装完成后，直接创建工作空间，将 Github 上的 carla_ros_bridge 整个下载下来进行编译即可。

```
mkdir -p ~/carla-ros-bridge && cd ~/carla-ros-bridge  
git clone --recurse-submodules https://github.com/carla-simulator/ros-bridge.git  
src/ros-bridge  
source /opt/ros/foxy/setup.bash
```

功能包涉及到的依赖需要用 `rosdep` 来安装一下，但可能因为网络的原因安装不成功，这时候就需要再次请来鱼香 ROS 的脚本，用 `rosdepc` 来安装一下，随后就可以正常编译了。

```
# 注意：以下操作都需要在 ~/carla-ros-bridge 中进行  
wget http://fishros.com/install -O fishros && . fishros  
rosdep update  
rosdepc install --from-paths src --ignore-src -r  
colcon build
```

第一次编译时通常会报错 `stderr: pcl_recorder`，再编译一遍就好了。

在运行前，需要更新下 Python 的环境变量，将以下语句添加至 `~/.bashrc` 的末尾：

```
# 注意：以下操作都需要在 ~/carla-ros-bridge 中进行  
sudo gedit ~/.bashrc  
  
# 尖括号需要修改为自己电脑上的路径  
export CARLA_ROOT=<path-to-carla>  
export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/carla/dist/carla-  
<carla_version_and_arch>.egg:$CARLA_ROOT/PythonAPI/carla
```

```
Open  ▾  [+l]  *.bashrc  Save  ≡  -  □  ×

91 alias ll='ls -alF'
92 alias la='ls -A'
93 alias l='ls -CF'
94
95 # Add an "alert" alias for long running commands. Use like so:
96 #   sleep 10; alert
97 alias alert='notify-send --urgency=low -i "${?} = 0" && echo terminal || echo error)' "$-
(history|tail -n1|sed -e '\s/^s*[0-9]\+\s*//;s/[;&]\s*alert$//'\''"'
98
99 # Alias definitions.
100 # You may want to put all your additions into a separate file like
101 # ~/.bash_aliases, instead of adding them here directly.
102 # See /usr/share/doc/bash-doc/examples in the bash-doc package.
103
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112     if [ -f /usr/share/bash-completion/bash_completion ]; then
113         . /usr/share/bash-completion/bash_completion
114     elif [ -f /etc/bash_completion ]; then
115         . /etc/bash_completion
116     fi
117 fi
118
119 # >>> fishros initialize >>>
120 source /opt/ros/foxy/setup.bash
121 # <<< fishros initialize <<<
122
123 export CARLA_ROOT=/home/qrq/Documents/CARLA_0913
124 export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/carla/dist/carla-0.9.13-py3.7-linux-
x86_64.egg:$CARLA_ROOT/PythonAPI/carla
125 export PATH=/home/qrq/.local/bin/:$PATH$
```

不推荐使用虚拟机，如果有条件，尽量使用原生 Ubuntu20 系统

对于使用 VMWare 虚拟机用户：此时需要下载一份 Linux 版本的 Carla，并解压到虚拟机的某处，这里的 `<path-to-carla>` 则应改为虚拟机中的 Carla 根目录。

对于使用 WSL2 子系统的用户：此时需要下载一份 Linux 版本的 Carla，并将 Linux 版 Carla 中 `PythonAPI/carla/dist/` 文件夹下的两个 `egg` 文件拷贝到 Windows 主机里 Carla 根目录下的 `PythonAPI/carla/dist` 文件夹下，并将 `<path-to-carla>` 修改为 `/mnt/<path-to-carla>`，其中的 `path-to-carla` 为 windows 下 Carla 的根目录。

我的 Carla 在 Windows 中的地址为 `C:\ProgramFiles\windowsNoEditor`，则此时的 `<path-to-carla>` 应当改为 `/mnt/C/ProgramFiles/windowsNoEditor`，并且在 `C:\ProgramFiles\windowsNoEditor\PythonAPI\carla\dist` 下应当存在四个文件，两个 `.whl` 和两个从 Linux 版 Carla 的安装包里拷贝出来的 `.egg`

这时候就可以启动 launch 文件了：

先开启一个终端运行 Carla

```
# 注意：以下操作都需要在 carla 仿真器的根目录中进行
./CarlaUE4.sh
```

再 ctrl + shift + t 开一个新终端，刷新环境变量并启动 carla_ros_bridge

```
# 注意：以下操作都需要在 ~/carla-ros-bridge 中进行
source ./install/setup.bash
ros2 launch carla_ros_bridge carla_ros_bridge.launch.py
```

不推荐使用虚拟机，如果有条件，尽量使用原生 Ubuntu20 系统

对于使用 VMWare 虚拟机用户：此时需要把网络的连接方式改为==桥接==，后续操作与 WSL2 用户一致。

对于使用 WSL2 子系统的用户：此时需要将 `./src/carla_ros_bridge/launch` 文件夹中的 `.launch` 文件以及 `.launch.py` 中的 `localhost` 改为 Windows 主机的 IPv4 地址（一共四个文件）。修改完后需要 `colcon build` 一下以使配置生效。

```
7 def generate_launch_description():
8     ld = launch.LaunchDescription([
9         launch.actions.DeclareLaunchArgument(
10             name='host',
11             default_value='192.168.50.101'
12         ),
13         launch.actions.DeclareLaunchArgument(
14             name='port',
15             default_value='2000'
16         ),
17         launch.actions.DeclareLaunchArgument(
18             name='timeout',
19             default_value='2'
20         )
21     ])
22     return ld
```

carla_ros_bridge.launch
~/Documents/carla-ros-bridge/src/carla_ros_bridge/launch

```
1 <!-- -->
2 <launch>
3 <!-- global parameter to set CARLA as time source -->
4 <param name="use_sim_time" value="True"/>
5
6 <!-- the network connection for the python connection to CARLA -->
7 <arg name='host' default='192.168.50.101' />
8 <arg name='port' default='2000' />
9 <arg name='timeout' default='2' />
10 <!-- -->
```

我在运行时遇到了如下问题：

Error: time-out of 10000ms while waiting for the simulator,
make sure the simulator is ready and connected to localhost:2000

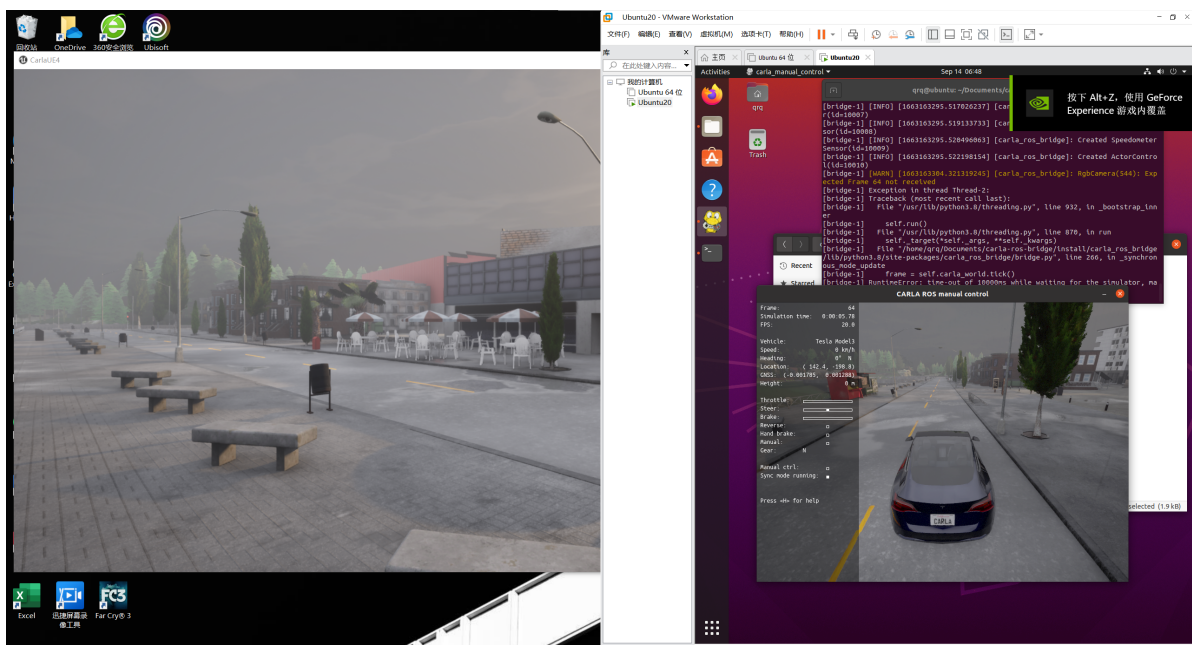
此时需要回到启动 Carla 的窗口，先用 `Ctrl + C` 终止程序运行，再将 Carla 的端口号手动改为 2000：

```
./CarlaUE4.sh -carla-rpc-port=2000
```

这时候再运行 `carla_ros_bridge` 就没问题了，可以启动如下的 launch 文件畅玩 Carla 了：

```
ros2 launch carla_ros_bridge carla_ros_bridge_with_example_vehicle.launch.py
```

这里放一张运行成功的截图用于参考：



该文档为深蓝学院自动驾驶控制与规划课程配套文档，由课程助教朱旺与邱润其共同整理。未经允许，禁止上传到网上。

2.2 ROS 1

不推荐使用 ROS 1，如果有条件，尽量使用 Ubuntu 20 + ROS 2 Foxy 的方案

2.2.1 一些准备工作

ROS 1 的 carla_ros_bridge 安装与 ROS 2 并无太大差异：

对于 Ubuntu 18 用户，需要针对 Python 2 进行设置，至于 1.1 节提到的针对 Python 3 的环境设置最好也完成一下，至于不设置会不会出问题我还没测试过。

```
# 安装 python 2 的 pip
sudo apt install python-pip

# 确定 pip 版本
pip -V

# 升级 pip
pip install --upgrade pip

# 为 Python 2 安装 pygame、numpy、opencv
pip install --user pygame numpy
pip install opencv-python

# 打开 bashrc 文件
sudo gedit ~/.bashrc

# 在文件末尾指定 carla 依赖库的路径
## 尖括号需要修改为自己电脑上的路径
## 注意：<carla_version_and_arch> 记得选择 py2.7 的版本
export CARLA_ROOT=<path-to-carla>
export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/carla/dist/carla-
<carla_version_and_arch>.egg:$CARLA_ROOT/PythonAPI/carla
```

对于 Ubuntu 20 用户，在完成 1.1 节的基础上，只用在 `bashrc` 文件后添加 carla 依赖库的路径即可，不用管 Python 2 的事：

```
# 打开 bashrc 文件
sudo gedit ~/.bashrc

# 在文件末尾指定 carla 依赖库的路径
## 尖括号需要修改为自己电脑上的路径
## 注意：<carla_version_and_arch> 记得选择 py3.7 的版本
export CARLA_ROOT=<path-to-carla>
export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/carla/dist/carla-
<carla_version_and_arch>.egg:$CARLA_ROOT/PythonAPI/carla
```

2.2.2 安装 carla_ros_bridge

ROS 1 可以借助鱼香 ROS 的脚本一键安装：

```
wget http://fishros.com/install -O fishros && . fishros
```

安装完成后，直接创建工作空间，将 Github 上的 carla_ros_bridge 整个下载下来进行编译即可。
该文档为深蓝学院自动驾驶控制与规划课程配套文档，由课程助教朱旺与邱润其共同整理。未经允许，禁止上传到网上。


```
mkdir -p ~/carla-ros-bridge/catkin_ws/src
git clone --recurse-submodules https://github.com/carla-simulator/ros-bridge.git
catkin_ws/src/ros-bridge

# Ubuntu 18 用户选择 melodic, Ubuntu 20 用户选择 noetic, 例: source
/opt/ros/melodic/setup.bash
source /opt/ros/<melodic/noetic>/setup.bash
```

功能包涉及到的依赖需要用 `rosdep` 来安装一下，但可能因为网络的原因安装不成功，这时候就需要再次请来鱼香 ROS 的脚本，用 `rosdep` 来安装一下，随后就可以正常编译了。

```
cd ~/carla-ros-bridge/catkin_ws

# 注意：以下操作都需要在 ~/carla-ros-bridge/catkin_ws 中进行
wget http://fishros.com/install -O fishros && . fishros
rosdep update
rosdep install --from-paths src --ignore-src -r
catkin build
```

如果遇到编译报错的状况，可以尝试再编译一次，如果还报错，就根据报错信息 google 一下吧。

在启动 Carla 仿真器后，新开一个终端，运行如下命令：

```
roslaunch carla_ros_bridge carla_ros_bridge_with_example_ego_vehicle.launch
```

若有在虚拟机中使用 ROS 1 的用户，还请回看 2.1 节中对虚拟机的设置说明，唯一的区别在于 launch 文件的位置不同，需要将 `carla_ros_bridge.launch` 和 `carla_ros_bridge_with_example_vehicle.launch` 中的 `localhost` 修改为主机的 `IPv4` 地址。

运行后若出现了 pygame 的控制界面，说明安装成功，请尽情享受吧！

3 ROS 2 版本的课程作业——Project 1 PID controller

3.1 准备工作

1. 完成Ubuntu20.04系统安装，Carla 0.9.13安装，ros 2(foxy)安装以及官方carla-ros-bridge的编译测试。
2. 课程作业是在官方carla-ros-bridge的基础上进行优化改进来的，因此在下载作业需要的课程文件前，请删除原来的carla-ros-bridge文件夹或者将课程上下载的压缩包放到和官方bridge不同的目录下。
3. 从课程官网下载所需的课程文件，或者通过下列命令克隆仓库到本地，仓库托管在gitee上，后续课程作业也会更新到课程中，大家可以通过注册gitee的方式，然后fork一份仓库到自己的账户下。

```
git clone https://gitee.com/wangwangzhu/carla-ros-bridge.git
```

4. 启动流程

1. 新开终端 A：

1. 启动carla 服务器端（参考前文）

2. 新开终端 B：

1. cd /carla-ros-bridge

2. source source_env.sh(如果还没有编译过工程，这一步会报错文件找不到，直接忽略)
 3. colcon build
 4. source source_env.sh
 5. ros2 launch carla_shenlan_bridge_ego_vis carla_bridge_ego_vehicle.launch.py
3. 新开终端 C:
1. source source_env.sh
 2. ros2 run carla_shenlan_pid_controller carla_shenlan_pid_controller_node
4. 新开终端 D:
1. source source_env.sh
 2. rqt(用于可视化控制指令以及速度曲线)
 3. 使用rqt的plot插件可视化下列三个消息，其中 `vehicle_status` 的 `velocity` 为车辆实际速度，`target_velocity` 的 `velocity` 为车辆的期望速度，将这两个速度曲线可视化到一个plot里面，可以直观的对比速度控制器性能。

<input type="checkbox"/> /carla/ego_vehicle/speedometer	
<input type="checkbox"/> /carla/ego_vehicle/vehicle_control_cmd	carla_msgs/msg/CarlaEgoVehicleControl
▶ header	std_msgs/Header
throttle	float
steer	float
brake	float
hand_brake	boolean
reverse	boolean
gear	int32
manual_gear_shift	boolean
<input type="checkbox"/> /carla/ego_vehicle/vehicle_info	carla_msgs/msg/CarlaEgoVehicleInfo
<input checked="" type="checkbox"/> /carla/ego_vehicle/vehicle_status	carla_msgs/msg/CarlaEgoVehicleStatus
▶ header	std_msgs/Header
velocity	float
acceleration	geometry_msgs/Accel
orientation	geometry_msgs/Quaternion
control	carla_msgs/CarlaEgoVehicleControl
<input type="checkbox"/> /carla/ego_vehicle/target_velocity	carla_msgs/msg/CarlaVehicleTargetVelocity
<input checked="" type="checkbox"/> /carla/ego_vehicle/target_velocity	carla_msgs/msg/CarlaVehicleTargetVelocity
▶ header	std_msgs/Header
velocity	float