Project 1

本项目希望大家根据 PID 控制方法实现一个巡航控制系统。这是 ROS 1.0 对应的系统框架，仅需要大家实现 `src/vehicle_control/src/pid_controller.cpp` 文件中的 todo 部分，即 PID 控制实现以及重置 PID 参数。

```
// /**to-do**/ 计算 PID 输出，实现PID控制
double PIDController::Control(const double error, const double dt) {

}

// /**to-do**/ 重置PID参数
void PIDController::Reset() {

}
```

实现 todo 部分后，需到项目的根目录运行 `catkin build` 进行编译，编译通过后运行如下命令：

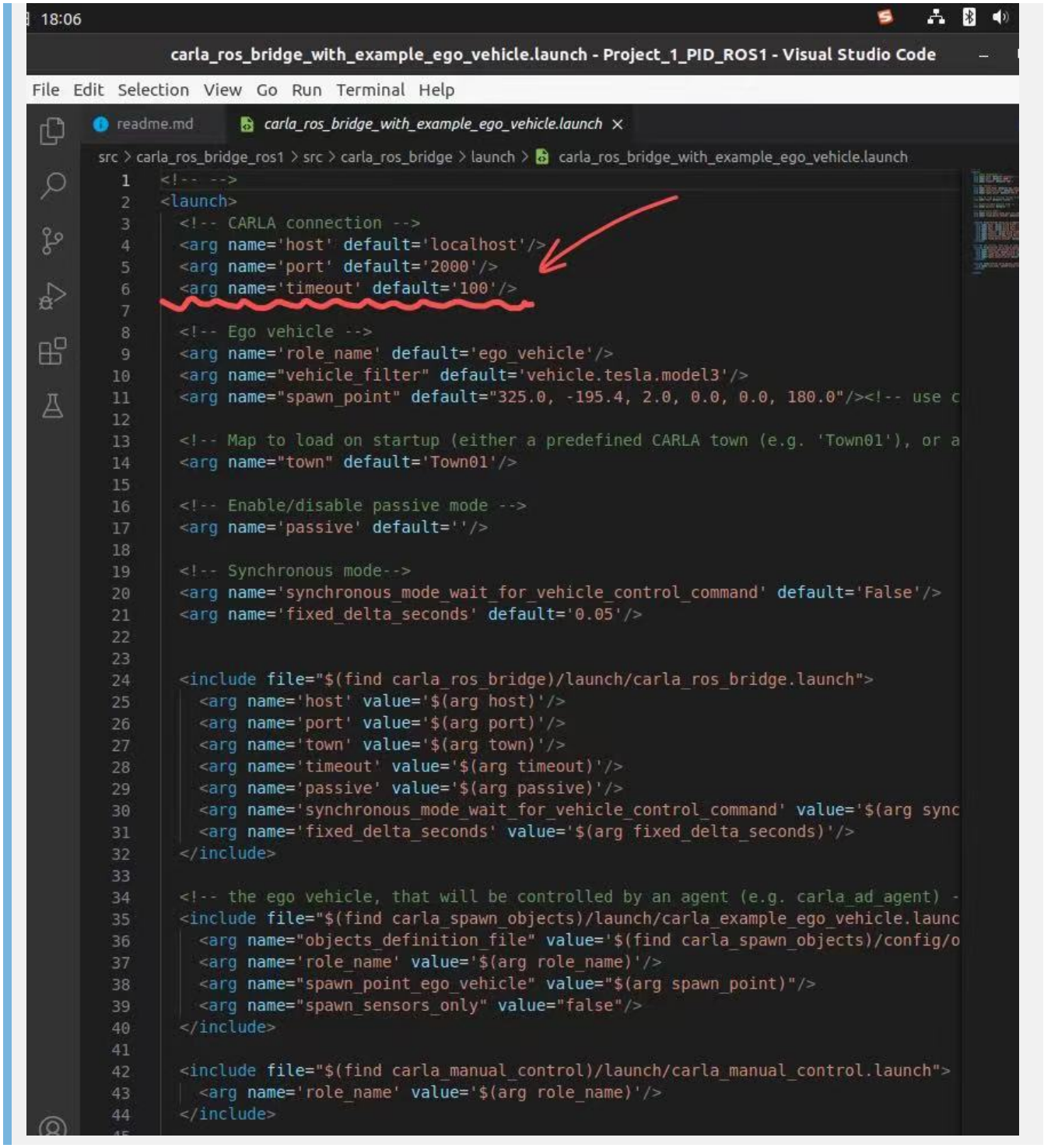
控制台1：

```
source ./devel/setup.bash
roslaunch carla_ros_bridge carla_ros_bridge_with_example_vehicle.launch
```

控制台2：

```
source ./devel/setup.bash
rosrun vehicle_control vehicle_control
```

> **特别注意：**
>
> 项目使用到的地图为 Town01，若启动 `carla_ros_bridge` 后地图没切换到 Town01，则说明 `.launch` 文件中的 `timeout` 参数时间给小了，需要调大 `timeout` 参数。


18:06
carla_ros_bridge_with_example_ego_vehicle.launch - Project_1_PID_ROS1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
readme.md
carla_ros_bridge_with_example_ego_vehicle.launch
src > carla_ros_bridge_ros1 > src > carla_ros_bridge > launch > carla_ros_bridge_with_example_ego_vehicle.launch
<!-- -->
<launch>
<!-- CARLA connection -->
<arg name='host' default='localhost'/>
<arg name='port' default='2000'/>
<arg name='timeout' default='100'/>
<!-- Ego vehicle -->
<arg name='role_name' default='ego_vehicle'/>
<arg name="vehicle_filter" default='vehicle.tesla.model3'/>
<arg name="spawn_point" default="325.0, -195.4, 2.0, 0.0, 0.0, 180.0"/><!-- use c
<!-- Map to load on startup (either a predefined CARLA town (e.g. 'Town01'), or a
<arg name="town" default='Town01'/>
<!-- Enable/disable passive mode -->
<arg name='passive' default=''/>
<!-- Synchronous mode-->
<arg name='synchronous_mode_wait_for_vehicle_control_command' default='False'/>
<arg name='fixed_delta_seconds' default='0.05'/>
<include file="$(find carla_ros_bridge)/launch/carla_ros_bridge.launch">
<arg name='host' value='$(arg host)'/>
<arg name='port' value='$(arg port)'/>
<arg name='town' value='$(arg town)'/>
<arg name='timeout' value='$(arg timeout)'/>
<arg name='passive' value='$(arg passive)'/>
<arg name='synchronous_mode_wait_for_vehicle_control_command' value='$(arg sync
<arg name='fixed_delta_seconds' value='$(arg fixed_delta_seconds)'/>
</include>
<!-- the ego vehicle, that will be controlled by an agent (e.g. carla_ad_agent) -
<include file="$(find carla_spawn_objects)/launch/carla_example_ego_vehicle.launc
<arg name="objects_definition_file" value='$(find carla_spawn_objects)/config/o
<arg name='role_name' value='$(arg role_name)'/>
<arg name="spawn_point_ego_vehicle" value="$(arg spawn_point)"/>
<arg name="spawn_sensors_only" value="false"/>
</include>
<include file="$(find carla_manual_control)/launch/carla_manual_control.launch">
<arg name='role_name' value='$(arg role_name)'/>
</include>