雨燕调试流程说明

**北京智行者科技有限公司**

目 录

[1 硬件准备 2](#_Toc500167088)

[2 系统准备 2](#_Toc500167089)

[3 模块调试 3](#_Toc500167090)

[4 系统联调 4](#_Toc500167091)

[5 迭代测试 4](#_Toc500167092)

# 硬件准备

当一辆新车改装成自动驾驶车辆时，首先需要改装好以下几项硬件系统：

1. EPS转向系统；
2. 刹车油门控制系统；
3. MCU底层控制器安装；
4. 传感器安装与配置，包括感知传感器（如激光雷达、毫米波雷达和视觉传感器等）和定位传感器（如GPS、IMU和轮速计等）；
5. 工控机、路由器、交换机、CANET等设备安装，其中工控机安装前，需配置如下系统和软件：

* Ubuntu系统，用户名和密码统一，开机自启，屏幕常亮
* ROS系统，以及各个支持AVPT编译运行的插件
* 绿盾加密系统
* 安装vim、sublime、ssh等常用软件
* Teamviewer IP及其登录密码设置
* 工控机本机IP

路由器需要如下配置：

* 路由器IP
* 路由器命名
* 密码设置

CANET需要如下配置：

* 主机IP及其端口号配置
* 传感器IP及其端口号配置

1. 线束整理和通信测试。

# 系统准备

在不同的车型及应用场景下，自动驾驶系统需要在launch/avos.launch中选择相应的车型参数及场景地图，若没有相应的车型参数或场景地图，需要做如下操作：

1. 新车型参数配置：拷贝其它车型的launch文件，修改如下几个模块的参数：
2. 车型名称和车身固定参数；
3. MCU参数；
4. Pathpalnner/motionplanner/prediction/steercontrol/drivercontrol/localization/perception等各算法模块参数；
5. TF坐标转换参数。
6. 新场景地图设置。配置测试场地地图文件，如果没有设定场地的地图文件，需要采集该场地的底图及参考路径，详见附录1（base-map）和附录2（ref-map）。

# 模块调试

自动驾驶系统的调试内容可分为五大模块：感知、定位、决策规划、控制和交互。而当一版稳定的自动驾驶系统（AVPT）应用在新车上时，首先需要配置定位模块（GPS），然后需要适配调试的是感知和控制模块，当感知测试完毕后，可供决策规划、纵向控制进行联调测试，大致模块调试执行顺序如下：

第一步：定位模块

目前只需GPS定位，分为4g和基站两种模式，都需要配置可用；

第二步：各独立模块的调试

1. 感知模块：
2. 激光雷达、毫米波雷达、摄像头等传感器参数的标定、网络设置；
3. 各传感器对周围车辆、行人等障碍物的检测；
4. 传感器检测障碍物的融合；
5. 横向控制：

路径跟踪模块只需要规划模块进行原始路径的提取即可进行单项功能的测试。

1. 纵向控制：

可以不依赖其它部分进行定速、停车控制调试；

1. 人机交互

进行新地图显示和路径规划等交互操作。

第三步：模块联调

1. 感知模块完成后，需要进行纵向跟车、切入、切出测试。
2. 感知模块完成后，决策规划模块，需要根据周围障碍物信息进行避障换道测试。

# 系统联调

当各个模块基本适配新车后，则需要进行集成测试，其目的是进行各模块的配合测试，当上游功能不满足下游功能需求时，则需要两个模块之间的完善、协调和配合，以完成设定功能。

当完成模块间的配合测试后，需要进行全工况的自动驾驶测试，即所有功能开启，目标是使车辆能够完成设定工况下的完全自动驾驶。

# 迭代测试

为了使自动驾驶系统更加稳定、完善，交付客户，以及向产品化方向发展，常态化的测试总结是一定需要的。

1. 当集成测试完成后，需要进行周期性的常态化测试。常态化测试应该具备以下条件：

（1）设置多工况的固定场景，包含常见的几大工况，例如：自由循迹、定速巡航、跟车、切入切出、避障换道、触发换道、全局路径规划、倒库、人机交互、红绿灯等；

（2）开放道路下的社会车辆和行人的融合测试；

（3）紧急工况下的应对处理；

进行一次常态化测试后，如果出现不满足要求的功能，则在下一次常态化测试前进行功能完善，并在下一次测试中重点测试。

1. 常态化测试执行方案：

常态化系统测试方案可借鉴蜗必达每周测试流程，具体分为：

1. 每两周进行两次设定工况环境下的自动驾驶测试，并与实际道路车辆和行人融合测试，由项目管理部主导，调试人员辅助完成；
2. 每次测试至少循环五次；
3. 全程记录相关数据，包含摄像头数据；
4. 当遇到功能不正常时，采用bag标记软件，截取问题出现时刻前后一定时间内的数据，并语音标记描述问题现象；
5. 各模块负责人根据问题现象认领，查找原因并在下次测试前解决该问题；
6. 下次全工况测试时，重点测试上次出问题部分；
7. 如果问题不再复现，说明该问题已得到解决；如果再次出现，则集中力量优先解决该问题。
8. 制作问题列表，记录出每次常态化测试中出现的问题，列出问题现象、责任人、问题原因、解决方案、解决状态等，方便后续分析历史问题记录等；
9. 质检部测试用例测试：

质检部每两周进行一次测试用例测试，用于全面分析各功能模块的性能，并整理出测试报告，量化各数据指标，不满足要求的进行针对性改进完善。

**附录1：basemap地图的采集和制作**

第一步是basemap采集和制作。操作步骤如下：（1）确定测试场地的大小和路线分布，最终确定基准点；（2）对每条路的basemap边界进行采集；（3）对采集边界进行离线处理生成basemap。

整个basemap的制作过程分为：在线采集数据和离线生成basemap。接下来对其进行详细的介绍：

**1 制作basemap涉及的工程文件：**

1. 车型launch，位置**[src/launch/paramfile]**，涉及参数如下：

|  |
| --- |
| <param name = "IF\_SAVE\_PATH" value = "0" type = "int"/>  <param name = "FILE\_PATH" value = "/home/idriver/test" type = "string"/> |

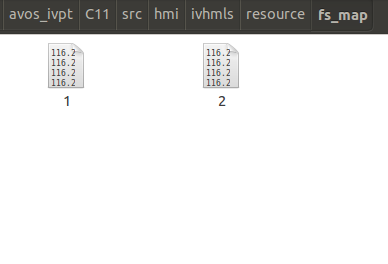
1. 地图采集ivhdmap.launch，位置**[src/hmi/ivhdmap/resource/[地图名称]/ivhdmap.launch]**，涉及设置basemap的顶点，具体参数如下：

|  |
| --- |
| <param name="iabasemaptllon" value = "116.23048029999999642" type = "double"/>  <param name="iabasemaptllat" value = "40.162281380000003139" type = "double"/> |

1. 离线生成basemap文件，节点位置**[src/hmi/ivhmls]**

**2 在线采集数据步骤：**

1. 准备事项：观察gps信号状态是否正常；
2. 设置地图顶点：如果在采集地图之前知道顶点，则需要设置**[src/hmi/ivhdmap/resource/[地图名称]/ivhdmap.launch]**文件中的顶点；如果不知道，跳过该步骤；
3. 修改记录数据标志位：在**车型launch**中，将IF\_SAVE\_PATH的value值设为**1**，并观察存储路径下是否生成新文件（txt文件）；
4. 路线采集：启动工程 ，围绕场地边缘，缓慢行驶，一圈完毕后，将产生的文件复制到**[ivhmls/resource/fs\_map]**文件夹中；
5. 重复步骤4，并在**[ivhmls/resource/fs\_map]**路径下重命名为 1,2,.....等连续数字，如下图所示，至此在线采集工作完毕；



**3 离线生成basemap**

1. 修改ivhmls.launch：

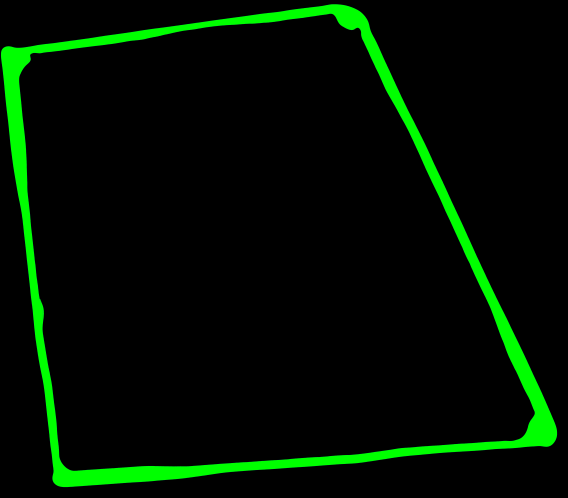
①如果预先知道地图顶点：修改**[src/hmi/ivhmls/ivhmls.launch]**中第9~11行的数据，设置basemap顶点位置和ivhdmap.launch中一致，生成basemap的方式为xy，即xy\_or\_gps" 的value = "0"；

|  |
| --- |
| <param name = "iabasemaptllon" value = "116.33882011999999406" type = "double"/>  <param name = "iabasemaptllat" value = "40.078692179999" type = "double"/>  <param name = "xy\_or\_gps" value = "0" type = "int"/> <!--0:using xy 1:using gps--> |

②如果预先不知道地图顶点：只修改**[src/hmi/ivhmls/ivhmls.launch]**中第11行的数据，生成basemap的方式为gps，即xy\_or\_gps" 的value = "1"；

|  |
| --- |
| <param name = "xy\_or\_gps" value = "1" type = "int"/> <!--0:using xy 1:using gps--> |

1. 在终端里面输入**[roslaunch ivhmls ivhmls.launch]** 即可在位置**[src/hmi/ivhmls/resource/fs\_basemap]**中生成basemap图片；
2. 用Windows画图板或者Linux下的GIMP(附录3)打开产生的freespace文件，用喷涂工具，将freespace内部喷成纯绿色，即RGB为（0,255,0）， 即可完成freespace的制作，最终结果如下图：



1. 将步骤（3）生成的图片（如上图）拷贝到对应地图的base\_map文件夹下；

若初始不知道地图顶点，需要将**[src/hmi/ivhmls/TL\_point.txt]**文件中的值（离线生成）拷贝到对应地图的ivhdmap.launch中；若知道顶点，跳过此步骤；

**附录2：refrence path路线关系的制定和采集**

1. **手绘地图和标记**

确定地图每条大路和路段，以及它们的标号，用手绘方式进行记录；以下用沙河测试场地的“8”字型地图作详细说明。



图 1 沙河测试场地“8”字型地图

地图标记步骤：

A、首先对全部的道路进行大路的划分，如图1中红色虚线框内为一大路，标号从1001开始。另外每一个转弯或者掉头按照一条大路来记，每一个通过路口也按一条大路记；

B、针对每一个大路，要具体划分成一个个的小路段，划分的原则是，遇到车道有分叉或者合并的长车道就要在此处切开，如图1中每一个黑色虚线即表示把车道切分成小路段的标记。小路段的标记从1开始，为了方便，可以按照一定的顺序来标记。

说明：图中的每一个黑色箭头即表示一个小路段，而且其表明了小路段的行驶方向，为了作图方便，黑色箭头表示的小路段可能与实际结构化道路的车道线有出入，但并不影响地图标记与采集。

1. **建立refpath的关系**

具体包含大路和大路之间的关系、大路和小路段之间的关系、小路段和小路段之间的关系，以及它们各自的速度、长度、转弯属性等信息；

* 1. **大路与大路间关系：**

所在文件名relationpathplanner2，在数据文件中大路间关系如图2所示，具体说明，第一列的roadID要按照顺序把所有的大路标记都列出，每一行的第二列至第五列分别为第一列大路所连接的大路标记，有几条就列出几条。每一行有5位有效位一位预留位。如1007按照行驶顺序可以到达1008和1014，所以1007这一行就是1007,1008,1014,-1,-1,-1 。其余类推。

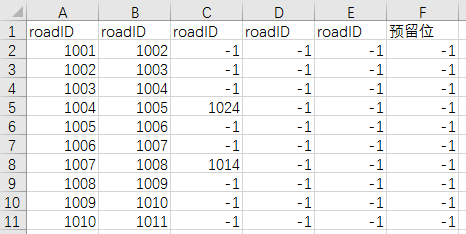


图 2 大路与大路间关系示例

* 1. **大路与小路段间关系：**

所在文件名relationpathplanner1，在数据文件中大路与小路段关系如图3所示，具体说明，第一列的roadID要按照顺序把所有的大路标记都列出，每一行的第二列为该大路的长度（单位：m），第三列为该大路预设速度（单位：m/s）第四列为该大路的属性（-1：非路口路段，0：路口调头，1：路口左转，2：路口直行，3：路口右转），第五列到第十二列为该大路所包含的所有小路段。每一行有12位有效位。如1005长度200，速度15，非路口路段，包含11~18路段，所以有1005,200,15,-1,8,9,10,11,12,13,14,15,16,17,18。其余类推。

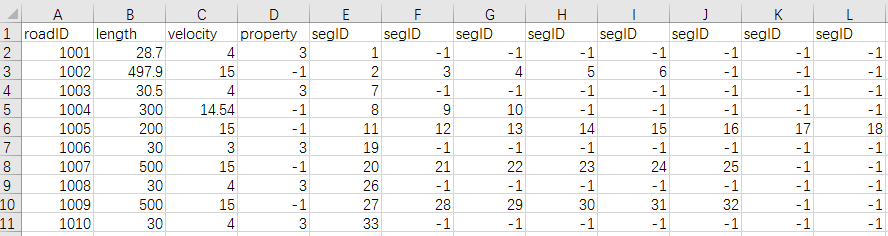


图 3 大路与小路间关系示例

* 1. **小路段与小路段间关系：**

所在文件名relationbegin\_var，在数据文件中小路段与小路段关系如图4所示，具体说明，第一列的segID要按照顺序把所有的大路标记都列出，每一行的第二列为该路段的长度（单位：m），第三列为该路段预设速度（单位：m/s）第四列到第八列，共五列为该路段的父路段列表，第九列到第十三列，共五列为该路段的子路段列表。第十四列为该路段左侧平行路段id（若无，为-1），第十五列为该路段id，第十六列为该路段右侧路段id（若无，为-1），第十七和十八列为空都为-1，第十九列为路段前进或后退属性（0为前进，1为后退）。每一行有19位有效位。如路段9长度300，速度15，无父路段，子路段有12，左侧平行为10，右侧平行为8，前进，所以有：9,300,15,-1,-1,-1,-1,-1,12,-1,-1,-1,-1,10,9,8,-1,-1,0,。其余类推。

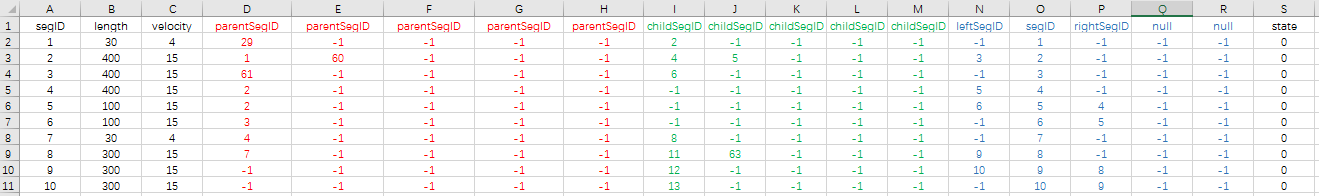


图 4 小路段间的关系示例

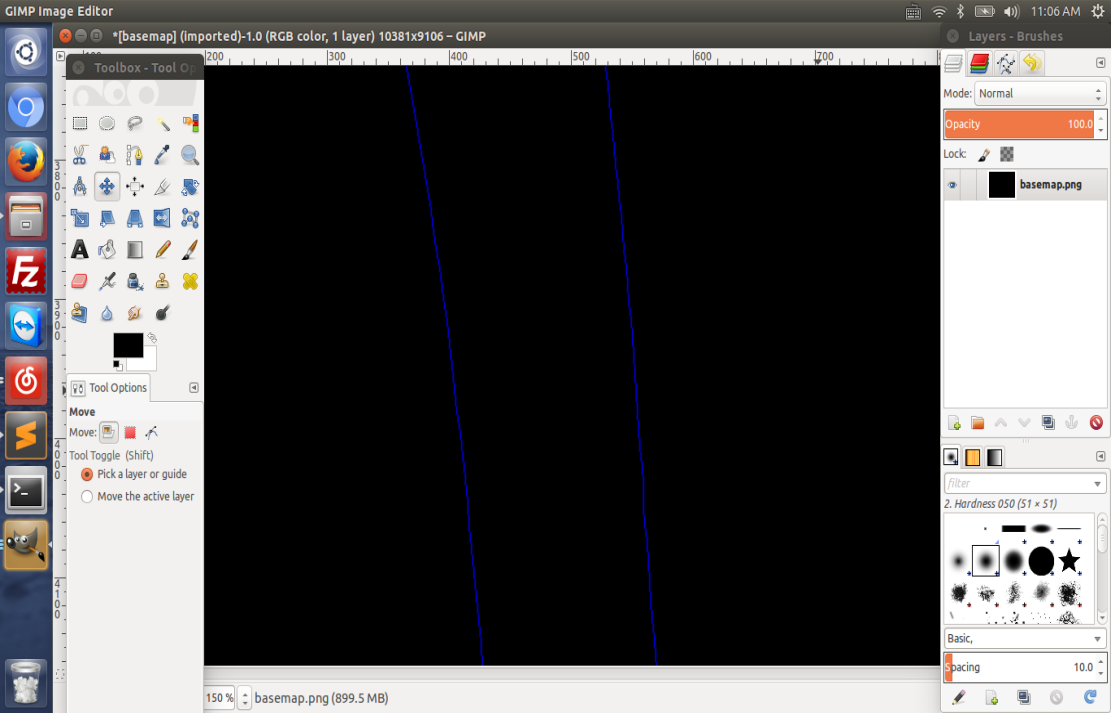
1. **采集所有路段**

按照小路段的标号，进行路段采集，并命名为相应的地图文件名称。

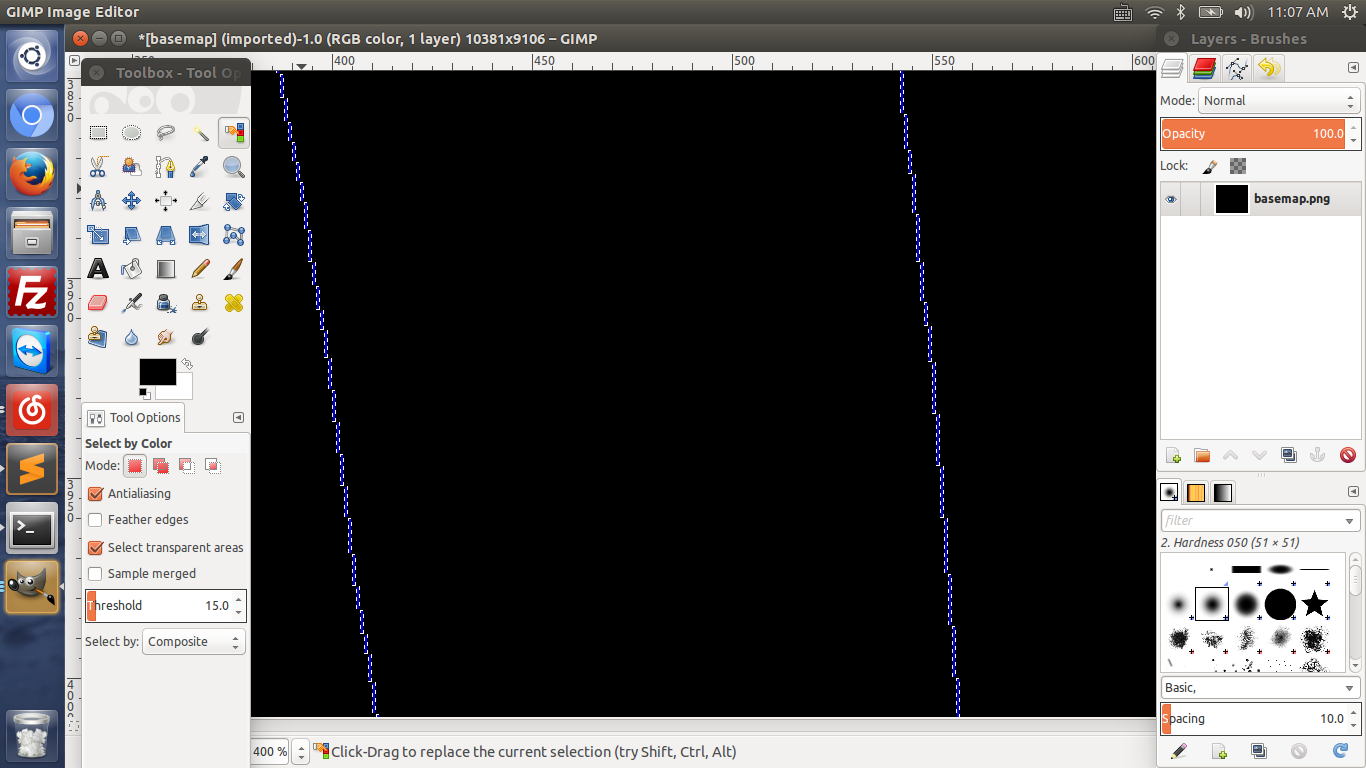
每一个采集的小路段命名格式为：路段标号-seg，如6号路段的地图文件命名为6-seg，并存放在整个地图文件中的reference\_map文件下。

**附录3：GIMP软件使用**

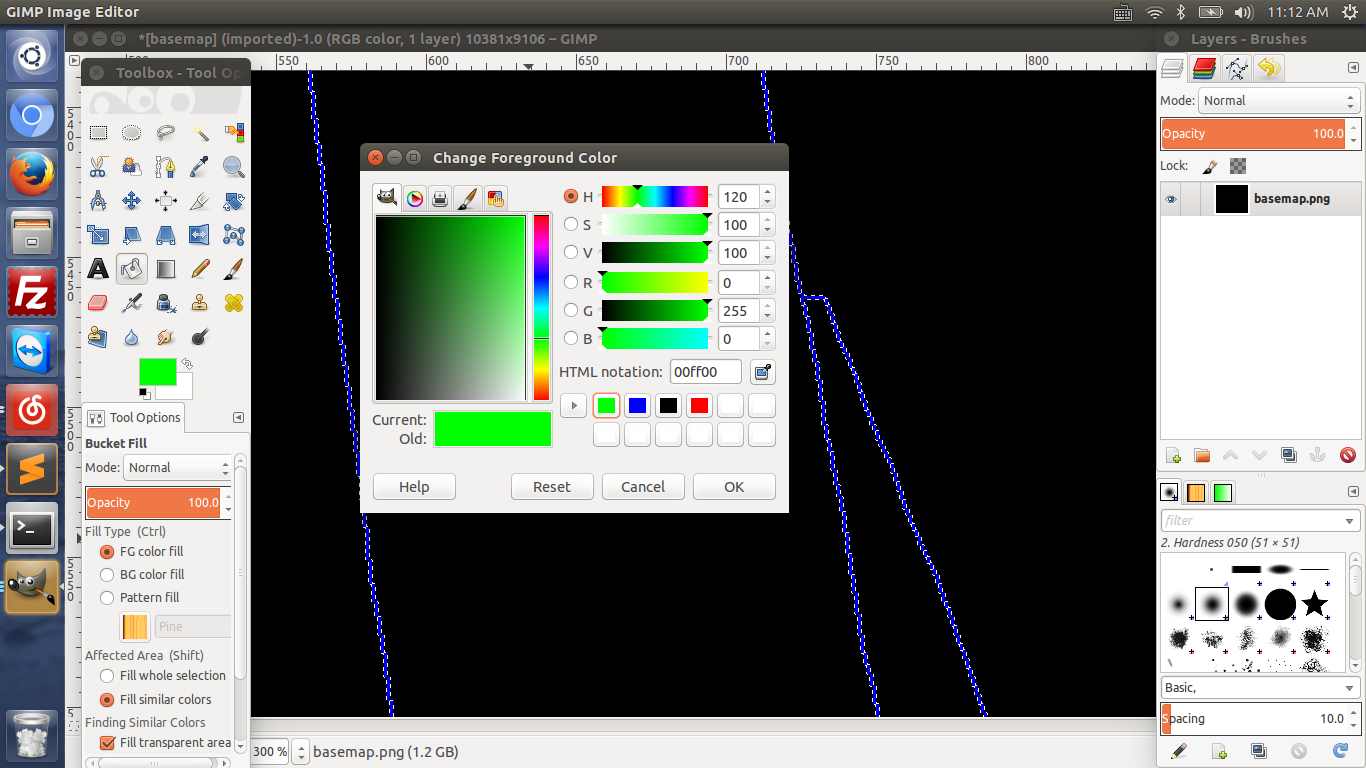
1. 联网在Linux的Software Center中安装GIMP软件。
2. 用GIMP打开ivhmls.launch生成的图片，如下：



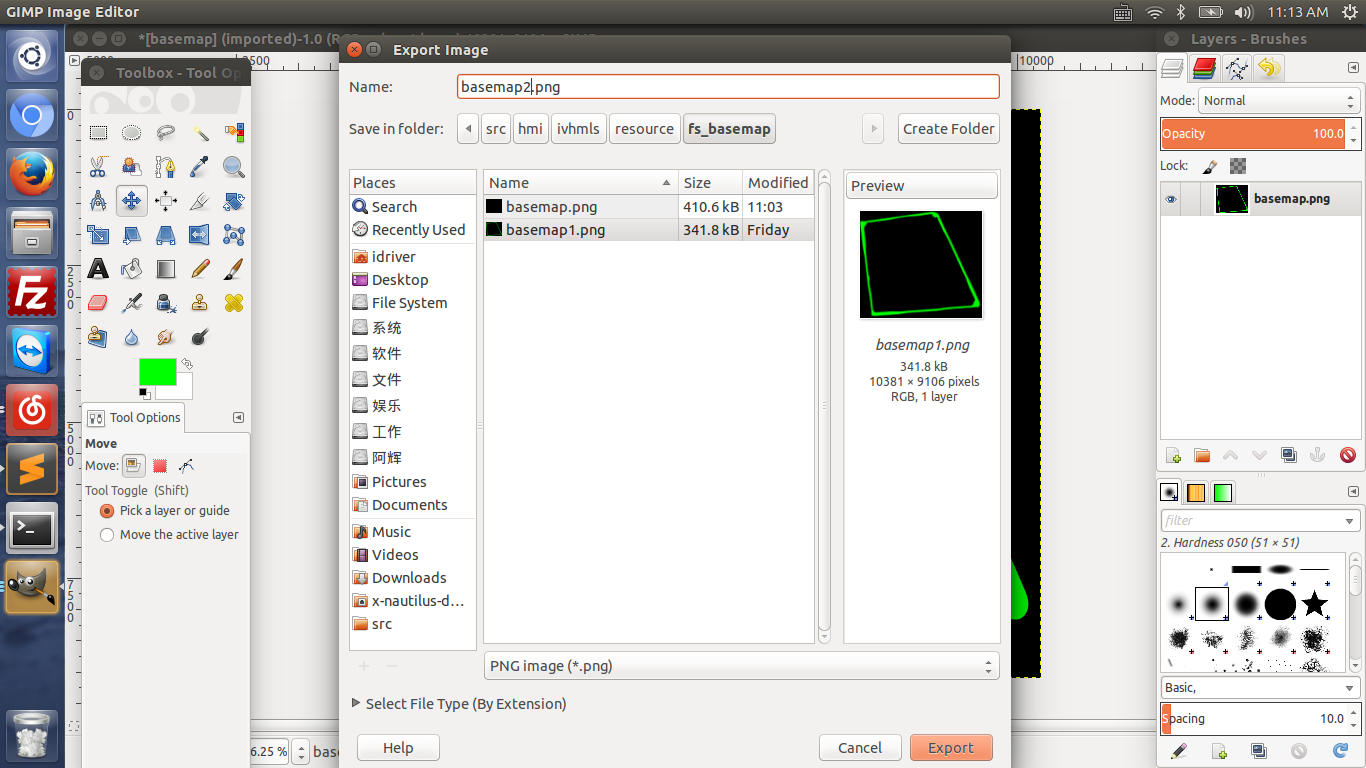
1. 检查采集的路段是否收尾相连（区域封闭），若没有需要使用画笔工具将收尾相连
2. 使用【像素选择工具】，选中上图中的蓝色线，如下图：



1. 在GIMP的菜单栏点击Select->Invert，进行反向选择
2. 使用【油漆桶填充工具】，设置颜色绿色，然后填充线中间的区域，如下：



1. 在GIMP的菜单栏点击File->Export As，在弹出的窗口里面下拉选择生成图片格式为png，如下：



-------------------------------------------END--------------------------------------