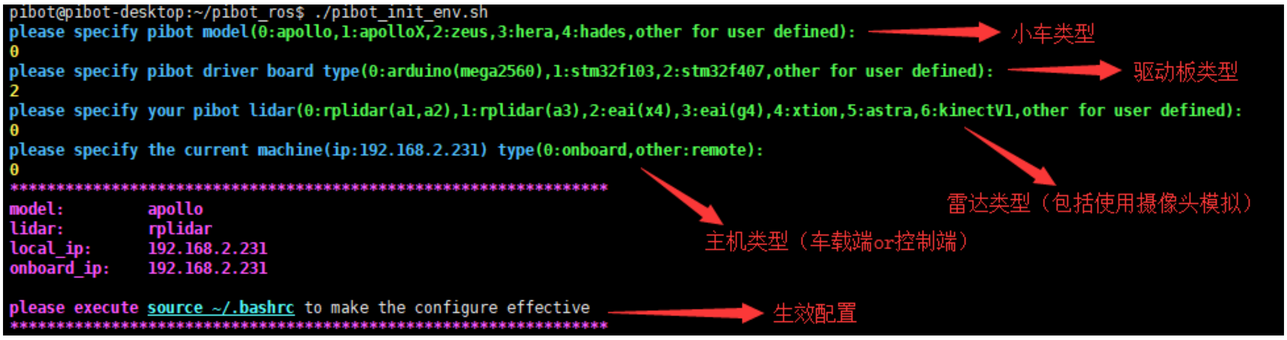
1.多机通讯

因车载主机限制(没有屏幕或者性能有限），无法启动`Rviz`查看建图等信息，ROS多机的通讯配置一文讲了具体的原理细节，即配置一个主机和一个从机，把从机的`ROS_MASTER_URI`指向主机即可，一般我把车载的主机配置指定为主机，而用于显示的PC或者虚拟机配置为从机。`PIBOT`提供了一键配置的脚本，按照提示选择即可PIBOT使用手册可以看到

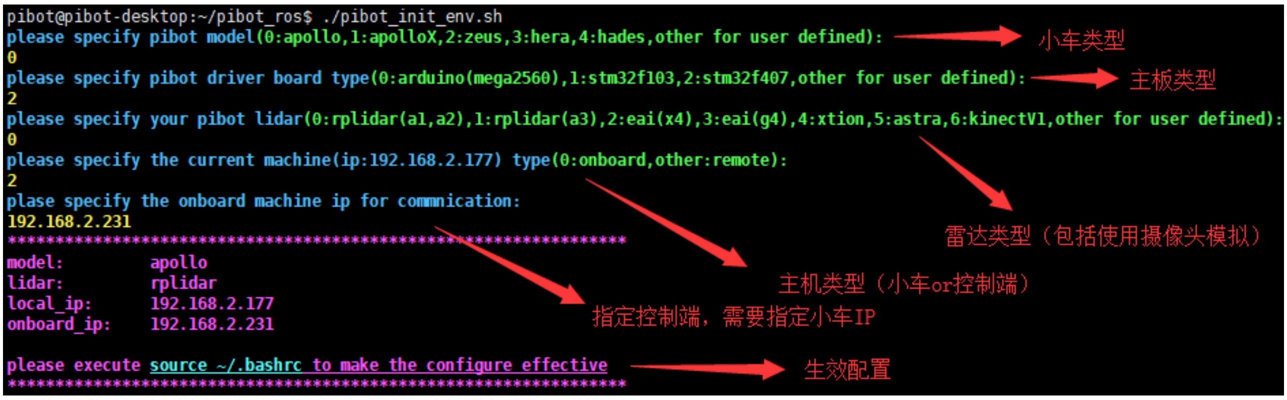
- 对于车载的主机



```
pibot@pibot-desktop:~/pibot_ros$ ./pibot_init_env.sh
please specify pibot model(0:apollo,1:apolloX,2:zeus,3:hera,4:hades,other for user defined):
0
please specify pibot driver board type(0:arduino(mega2560),1:stm32f103,2:stm32f407,other for user defined):
2
please specify your pibot lidar(0:rplidar(a1,a2),1:rplidar(a3),2:eai(x4),3:eai(g4),4:xtion,5:astra,6:kinectV1,other for user defined):
0
please specify the current machine(ip:192.168.2.231) type(0:onboard,other:remote):
0
********************************************************************
model:          apollo
lidar:          rplidar
local_ip:       192.168.2.231
onboard_ip:     192.168.2.231

please execute source ~/.bashrc to make the configure effective
********************************************************************
```

`machice type`为0即为主机

- 对于用于显示的主机或者虚拟机



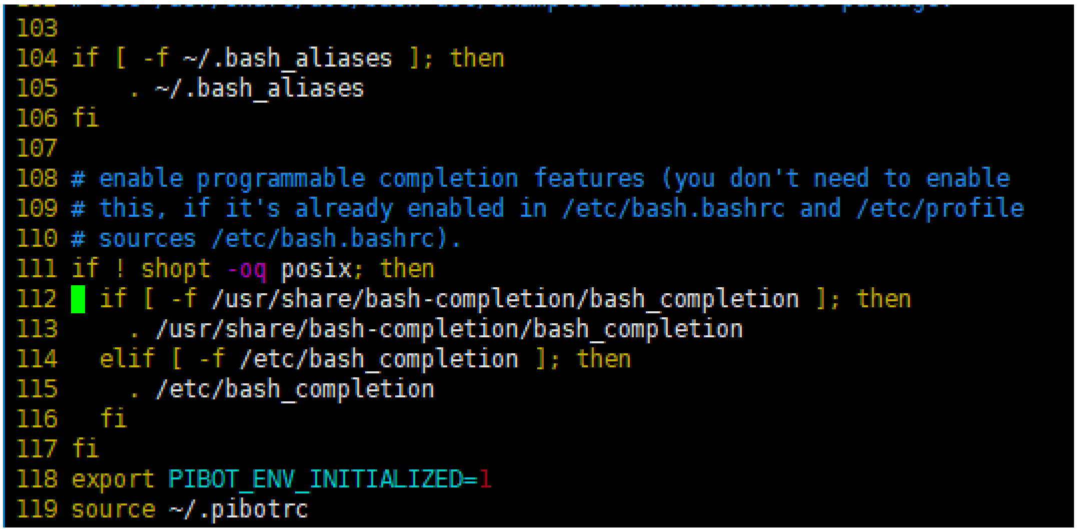
```
pibot@pibot-desktop:~/pibot_ros$ ./pibot_init_env.sh
please specify pibot model(0:apollo,1:apolloX,2:zeus,3:hera,4:hades,other for user defined):
0
please specify pibot driver board type(0:arduino(mega2560),1:stm32f103,2:stm32f407,other for user defined):
2
please specify your pibot lidar(0:rplidar(a1,a2),1:rplidar(a3),2:eai(x4),3:eai(g4),4:xtion,5:astra,6:kinectV1,other for user defined):
0
please specify the current machine(ip:192.168.2.177) type(0:onboard,other:remote):
2
plase specify the onboard machine ip for commnication:
192.168.2.231
********************************************************************
model:          apollo
lidar:          rplidar
local_ip:       192.168.2.177
onboard_ip:     192.168.2.231

please execute source ~/.bashrc to make the configure effective
********************************************************************
```

`machice type`为非0即为从机，同是作为从机，需要新增一个配置主机`IP`

2.pibot_init_env脚本

具体看下pibot_init_env做了什么

- 添加`PIBOT_ENV_INITIALIZED`环境变量
- 添加`source ~/.pibotrc`至`~/.basrc` 根据PIBOT_ENV_INITIALIZED是否定义，保证`source ~/.pibotrc`只被添加一次 执行`pibot_init_env`后`~/.bashrc`文件如下（118 119行）



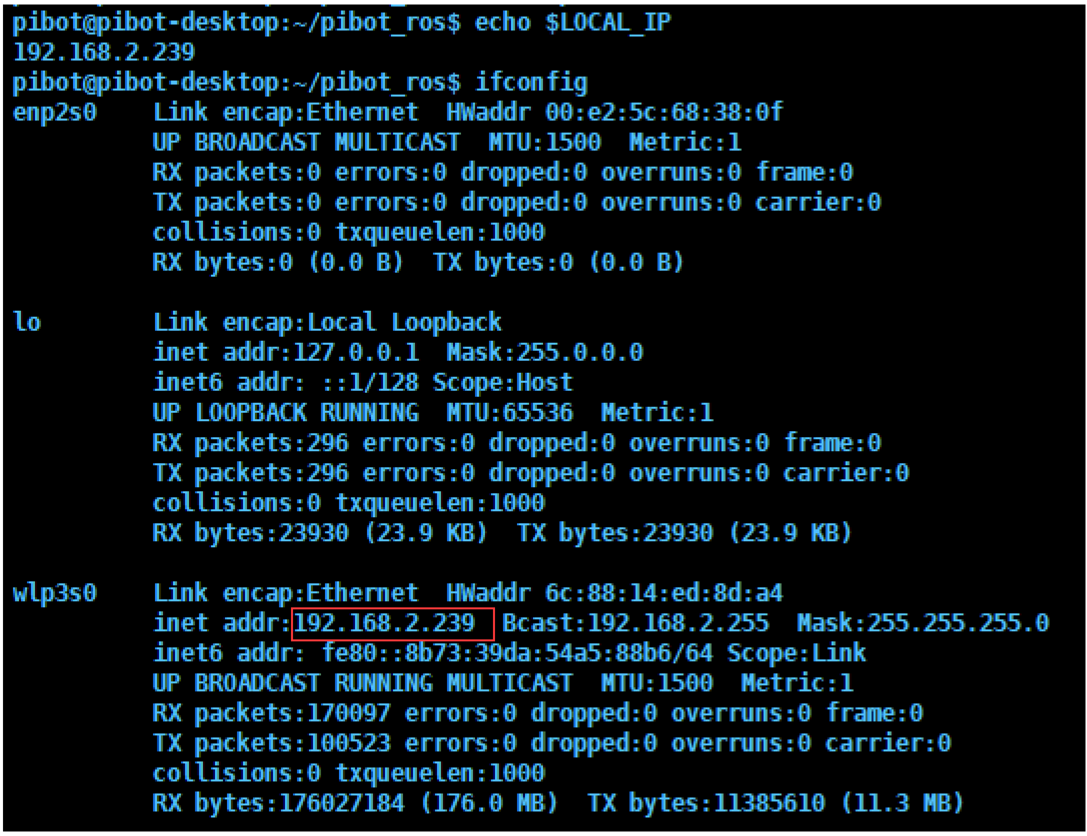
```

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
export PIBOT_ENV_INITIALIZED=1
source ~/.pibotrc
```

- 添加`udev rules` 可以在usb插入根据PID/VID生成`/dev/pibot`和`/dev/rplidar`等软连接，而不需指定具体的`/dev/ttyUSBn`
- 根据驱动板型号设置波特率 最终执行`python ros_ws/src/pibot_bringup/scripts/set_baud.py 115200`或者`python ros_ws/src/pibot_bringup/scripts/set_baud.py 921600`
- 添加`~/.pibotrc`文件

```
source /opt/ros/kinetic/setup.bash
LOCAL_IP=`ip addr | grep 'state UP' -A2 | tail -n1 | awk '{print $2}' | awk -F/ '{print $1}'`
export ROS_IP=`echo $LOCAL_IP`
export ROS_HOSTNAME=`echo $LOCAL_IP`
export PIBOT_MODEL=hades
export PIBOT_LIDAR=rplidar
export PIBOT_BOARD=stm32f4
export ROS_MASTER_URI=http://`echo $LOCAL_IP`:11311
source ~/pibot_ros/ros_ws/devel/setup.bash
```

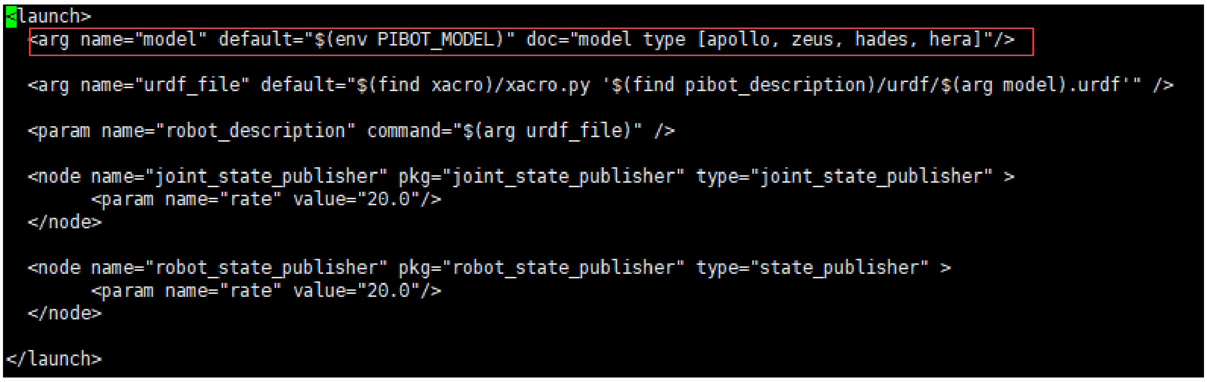
- `/opt/ros/kinetic/setup.bash` 生效`ROS`的环境变量,如果没安装好`ROS`会报该文件不存在的错误

- 第二行给LOCAL_IP赋值，我们尝试直接在终端输出该变量`echo $LOCAL_IP`

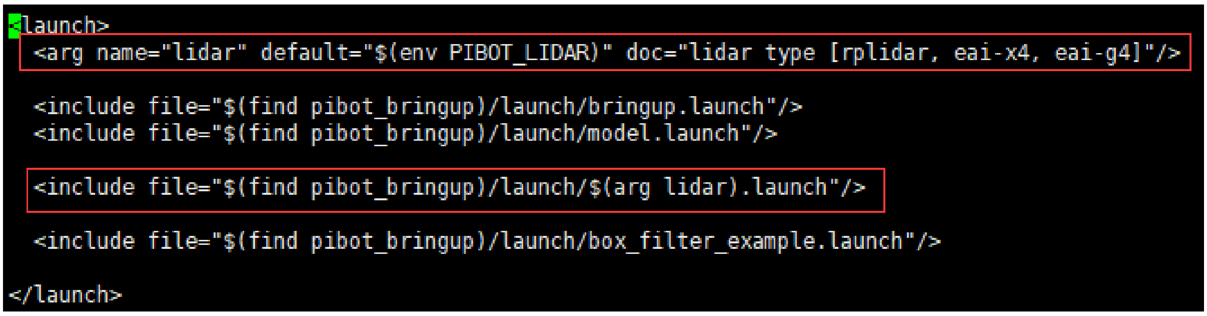
  ```
  pibot@pibot-desktop:~/pibot_ros$ echo $LOCAL_IP
  192.168.2.239
  pibot@pibot-desktop:~/pibot_ros$ ifconfig
  enp2s0    Link encap:Ethernet  HWaddr 00:e2:5c:68:38:0f
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

  lo        Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:296 errors:0 dropped:0 overruns:0 frame:0
            TX packets:296 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:23930 (23.9 KB)  TX bytes:23930 (23.9 KB)

  wlp3s0    Link encap:Ethernet  HWaddr 6c:88:14:ed:8d:a4
            inet addr:192.168.2.239  Bcast:192.168.2.255  Mask:255.255.255.0
            inet6 addr: fe80::8b73:39da:54a5:88b6/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:170097 errors:0 dropped:0 overruns:0 frame:0
            TX packets:100523 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:176027184 (176.0 MB)  TX bytes:11385610 (11.3 MB)
  ```

  可以看到得到的值跟`ifconfig`查看的一致，可以知道该值为当前`IP`
- 第三四行分别`ROS_IP`和`ROS_HOSTNAME` 这里使用的本地IP
- 第五六七行分别模型名称，雷达名称，驱动板名称
  - 对于模型名称`pibot_bringup/launch/model.launch`会使用到该变量

    ```
    <launch>
      <arg name="model" default="$(env PIBOT_MODEL)" doc="model type [apollo, zeus, hades, hera]"/>

      <arg name="urdf_file" default="$(find xacro)/xacro.py '$(find pibot_description)/urdf/$(arg model).urdf'" />

      <param name="robot_description" command="$(arg urdf_file)" />

      <node name="joint_state_publisher" pkg="joint_state_publisher" type="joint_state_publisher" >
            <param name="rate" value="20.0"/>
      </node>

      <node name="robot_state_publisher" pkg="robot_state_publisher" type="state_publisher" >
            <param name="rate" value="20.0"/>
      </node>

    </launch>
    ```

    从而加载对应的模型文件
  - 对于雷达名称`pibot_bringup/launch/robot.launch`会使用到该变量

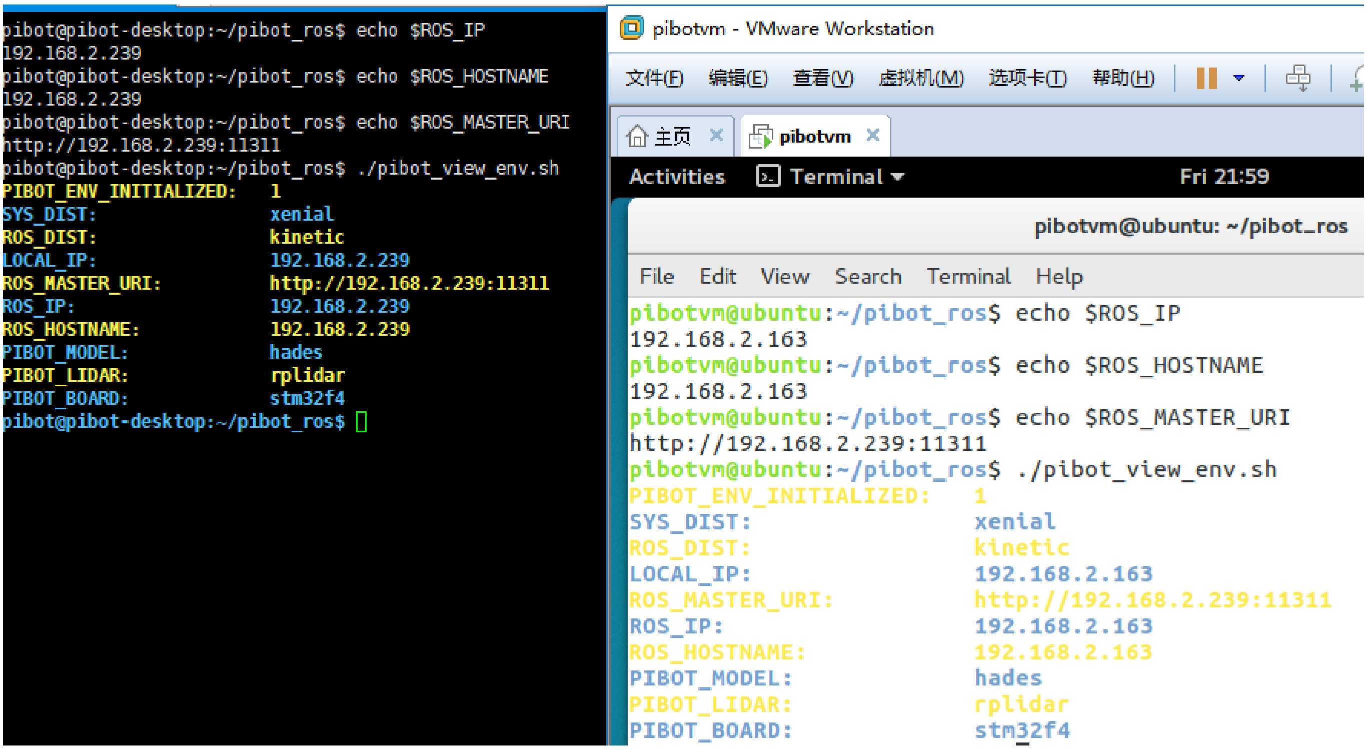
    ```
    <launch>
      <arg name="lidar" default="$(env PIBOT_LIDAR)" doc="lidar type [rplidar, eai-x4, eai-g4]"/>

      <include file="$(find pibot_bringup)/launch/bringup.launch"/>
      <include file="$(find pibot_bringup)/launch/model.launch"/>

      <include file="$(find pibot_bringup)/launch/$(arg lidar).launch"/>

      <include file="$(find pibot_bringup)/launch/box_filter_example.launch"/>

    </launch>
    ```

    从而加载对于的雷达文件
  - 驱动板名称尚未使用（只在运行是设置波特率）
- `export ROS_MASTER_URI=xxxx` 这里是主机和从机的唯一区别的地方，

  - 对于主机可以看到直接使用本地的`IP`,`export ROS_MASTER_URI=http://`echo $LOCAL_IP`:11311`
  - 对于从机直接使用的手动如输入的`IP`,`export ROS_MASTER_URI=http://192.168.2.231:11311`
- 最后一行即为生效`PIBOT`驱动包的环境变量 **这里需要编译，不然会提示文件不存在**

3.总结

其实如ROS多机的通讯配置所讲就是设置了`ROS_IP` `ROS_HOSTNAME`和`ROS_MASTER_URI`三个环境变量，前2个主机从机都是本机IP，后一个主机为本机IP，从机为主机IP，我们可以输出这几个变量或者使用`pibot_view_env`查看



```
pibot@pibot-desktop:~/pibot_ros$ echo $ROS_IP
192.168.2.239
pibot@pibot-desktop:~/pibot_ros$ echo $ROS_HOSTNAME
192.168.2.239
pibot@pibot-desktop:~/pibot_ros$ echo $ROS_MASTER_URI
http://192.168.2.239:11311
pibot@pibot-desktop:~/pibot_ros$ ./pibot_view_env.sh
PIBOT_ENV_INITIALIZED:   1
SYS_DIST:                xenial
ROS_DIST:                kinetic
LOCAL_IP:                192.168.2.239
ROS_MASTER_URI:          http://192.168.2.239:11311
ROS_IP:                  192.168.2.239
ROS_HOSTNAME:            192.168.2.239
PIBOT_MODEL:             hades
PIBOT_LIDAR:             rplidar
PIBOT_BOARD:             stm32f4
pibot@pibot-desktop:~/pibot_ros$
```

pibotvm - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
主页
pibotvm
Activities
Terminal
Fri 21:59
pibotvm@ubuntu: ~/pibot_ros
File Edit View Search Terminal Help

```
pibotvm@ubuntu:~/pibot_ros$ echo $ROS_IP
192.168.2.163
pibotvm@ubuntu:~/pibot_ros$ echo $ROS_HOSTNAME
192.168.2.163
pibotvm@ubuntu:~/pibot_ros$ echo $ROS_MASTER_URI
http://192.168.2.239:11311
pibotvm@ubuntu:~/pibot_ros$ ./pibot_view_env.sh
PIBOT_ENV_INITIALIZED:   1
SYS_DIST:                xenial
ROS_DIST:                kinetic
LOCAL_IP:                192.168.2.163
ROS_MASTER_URI:          http://192.168.2.239:11311
ROS_IP:                  192.168.2.163
ROS_HOSTNAME:            192.168.2.163
PIBOT_MODEL:             hades
PIBOT_LIDAR:             rplidar
PIBOT_BOARD:             stm32f4
```