

使用

下载

x86\_64\_amd64平台

arm64平台

运行

基础镜像制作

镜像或产品配置文件生成

必看事项：

### ⚠ Caution

在amd 平台生仅限用户名为rpp的用户使用本程序，否则会导致桌面无法正常启动

将/etc/gdm3/custom.conf 中的用户修改为真实用户即可

## 使用

机器连接网络后下载快捷工具rpp\_easy\_tools

### 下载

#### x86\_64\_amd64平台

```
mkdir rpp_tools && cd rpp_tools/  
wget http://file.robotplusplus.com.cn/rpp_easy_tools/rpp_easy_tools_amd64_1.0.0.zip  
md5sum rpp_easy_tools.zip #检查md5  
unzip -x rpp_easy_tools.zip
```

#### arm64平台

```
mkdir rpp_tools && cd rpp_tools/  
wget http://file.robotplusplus.com.cn/rpp_easy_tools/rpp_easy_tools_arm64_1.0.0.zip  
md5sum rpp_easy_tools.zip #检查md5  
unzip -x rpp_easy_tools_arm64_1.0.0.zip
```

## 运行

### 基础镜像制作

#### 📌 Note

基础镜像制作需要在纯净系统中制作

纯净系统在安装完成后需要先保证补丁、依赖等完成更新

```
sudo apt update
sudo apt upgrade
sudo reboot
```

以zbox\_3080\_1804（索泰\_RTX3080\_ubuntu18.04）为例子

#### 1. 加载环境变量

```
cd ~/rpp_tools/rpp_easy_tools/robotplusplus_deploy_tool
source ./global_env.config
```

#### 2. 载入配置

```
cd config/
mv ./zbox_3080_1804.config base_image.config
```

#### 3. 执行程序，开始配置环境

```
rpp_base_generate [config name] # 执行rpp_base_generate根据需求选择对应config即可
```

等待程序执行完毕后，重启机器即可

#### ① Note

过程中需要输入密码

#### 4. realsense驱动安装

cuda 加速

##### 1. 加载代理

```
git config --global http.proxy 192.168.22.54:20172 #全局生效
git config --global https.proxy 192.168.22.54:20172 #全局生效
export http_proxy=192.168.22.54:20172
export https_proxy=192.168.22.54:20172
rpp_librealsense_install https://github.com v2.53.1 /usr/bin/python3.6 true 0
```

##### 2. 安装

```
rpp_librealsense_install https://github.com v2.53.1 /usr/bin/python3.6 true 0
```

无GPU

暂不安装

#### 5. 硬盘清理

- 清理rpp.log
- 启动bleachbit

bleachbit

修改 gedit .config/bleachbit为

```
[bleachbit]
auto_hide = True
auto_start = False
check_beta = False
check_online_updates = False
shred = False
exit_done = False
delete_confirmation = True
units_iec = False
first_start = False
version = 2.0
hashsalt =
daeff4e798eb83a217b7dc611493c106a76b49ad4970e396fbf5f711ed6beb69aad6fef37e511d98
36f8e02e0f0d7b28a0fd435ed16c0ebee3570889b36678eb

[hashpath]

[list/shred_drives]
0 = /home/rpp/.cache
1 = /tmp

[preserve_languages]
en = True

[tree]
apt.clean = True
apt = True
bash.history = True
bash = True
evolution = True
evolution.cache = True
firefox.dom = True
firefox = True
firefox.download_history = True
firefox.session_restore = True
firefox.backup = True
firefox.crash_reports = True
firefox.vacuum = True
firefox.site_preferences = True
firefox.cache = True
firefox.forms = True
firefox.cookies = True
gnome.search_history = True
gnome = True
journald.clean = True
journald = True
deepscan.tmp = True
deepscan = True
system.trash = True
system = True
system.rotated_logs = True
```

```
system.desktop_entry = True
system.cache = True
thumbnails.cache = True
thumbnails = True
```

#### Note

修改配置文件需要重启软件

或者手动选择至下图所示勾选后**点击清理**即可

## 6. 基础镜像快照制作

```
cd ~/rpp_tools
unzip -x rpp_fast_migrate_record.zipcd
cd rpp_fast_migrate_record/
./rpp_fast_migrate_record_rpp_ #启动程序
```

### 打开一个新终端

为了避免将工厂模式存入镜像中在制作镜像前需要将app改为用户模式

```
mv rpp_fast_migrate_record_rpp_fast_migrate_record
```

在弹出的gui中的基础状态工具集中点击1-基础状态制作，并根据提示操作即可

### arm

由于arm平台暂时没有合适的部署镜像的方案，以mmc1为例其部署会将开发环境(基础镜像)和标准产品功能产品同时拉取部署：

```
cd rpp_easy_tools/robotplusplus_deploy_tool/
source ./global_env.config
rpp_nx_base_generate mmc1_t503_image.config
```

## 镜像或产品配置文件生成

#### Important

无gpu型号机器在生成配置文件时必须去除掉或注释**nvidia\_gpu\_model**

```
xxx_image:
env_map:
robuster_git_user: rpp:shihe321
local_drivers_repos: $HOME/rpp_ws/src/drivers
local_robot_repos: $HOME/rpp_ws/src/robot
local_slam_repos: $HOME/rpp_ws/src/slam
local_navigation_repos: $HOME/rpp_ws/src/navigation
local_manipulator_repos: $HOME/rpp_ws/src/manipulator
```

```

local_application_repos: $HOME/rpp_ws/src/application
local_vision_repos: $HOME/rpp_ws/src/vision
local_3rdparty_repos: $HOME/3rdparty
#nvidia_gpu_model: "3080" #机器没有gpu 务必注释此行后再进行生成!!!!!!
car_mode: robot
system_architecture: Linux_arm64 # or Linux_arm64
PS1_STRING: \\[\033[1;32m\\]\u\\[\033[1;36m\\]<\\[\033[1;32m\\]$car_mode\\[\033[1;36m\\]>:\\
[\033[1;34m\\]\w\\[\033[0m\\]\\$ '
apt_source_url: https://mirrors.tuna.tsinghua.edu.cn
pip_source_url: https://pypi.tuna.tsinghua.edu.cn/simple
rpp_tools_link: http://file.robotplusplus.com.cn/common_tools/1.0.0
nvidia_toolkit_link: http://file.robotplusplus.com.cn/nvidia/ubuntu18.04-cuda-11.4.0-cudnn-
8.4.1.50-tensorrt-8.4.1.5-amd64/
git_3rdparty_link: https://git.robotplusplus.com.cn/third-party
ros1_source_url: https://mirrors.tuna.tsinghua.edu.cn/ros/ubuntu/
ros_distro_version: melodic

```

### Important

注意不同平台的机器人需要更改**system\_architecture**

```

xxx_image:
env_map:
robuster_git_user: rpp:shihe321
local_drivers_repos: $HOME/rpp_ws/src/drivers
local_robot_repos: $HOME/rpp_ws/src/robot
local_slam_repos: $HOME/rpp_ws/src/slam
local_navigation_repos: $HOME/rpp_ws/src/navigation
local_manipulator_repos: $HOME/rpp_ws/src/manipulator
local_application_repos: $HOME/rpp_ws/src/application
local_vision_repos: $HOME/rpp_ws/src/vision
local_3rdparty_repos: $HOME/3rdparty
#nvidia_gpu_model: "3080" #机器没有gpu 务必注释此行后再进行生成!!!!!!
car_mode: robot
system_architecture: Linux_arm64 # or Linux_arm64 #根据cpu架构进行选择
PS1_STRING: \\[\033[1;32m\\]\u\\[\033[1;36m\\]<\\[\033[1;32m\\]$car_mode\\[\033[1;36m\\]>:\\
[\033[1;34m\\]\w\\[\033[0m\\]\\$ '
apt_source_url: https://mirrors.tuna.tsinghua.edu.cn
pip_source_url: https://pypi.tuna.tsinghua.edu.cn/simple
rpp_tools_link: http://file.robotplusplus.com.cn/common_tools/1.0.0
nvidia_toolkit_link: http://file.robotplusplus.com.cn/nvidia/ubuntu18.04-cuda-11.4.0-cudnn-
8.4.1.50-tensorrt-8.4.1.5-amd64/
git_3rdparty_link: https://git.robotplusplus.com.cn/third-party
ros1_source_url: https://mirrors.tuna.tsinghua.edu.cn/ros/ubuntu/
ros_distro_version: melodic

```

## 必看事项:

制作基础镜像后务必，测试其基础快照是否正常！

测试方法如下：

1. 制作完基础快照后，上传至guidang测试文件夹
2. 使用**rpp\_product\_generate**命令刷入工控机接近的产品镜像（如nodka-mr2000）
3. 产品相关写入完制作出厂快照，并提取相应增量备份,上传至测试文件夹
4. 用另一台工控机(b)刷入刚上传的测试文件夹中的基础镜像
5. 在工控机b中尝试使用增量备份，若恢复成功则本次基础镜像制作成功，失败请联系相关维护人