```
Neato(r) Turtle Cheatsheet Launch Scripts       by Ross Lunan (arlunan@ieee.org)
For SBC Robot (Raspberry Pi models) and Remote Workstation Desktop Computer which can be a Ubuntu
22.04 (Jammy) Desktop Machine or MAC Virtual Machine
This document on my Mac: /Users/Ross/Documents/ROS Projects/Cheatsheets & Install/Botvac

Based on Neato Turtle in "Homebrewed Robots!" by Camp Peavy
https://github.com/cpeavy2/botvac_node

Note: These scripts assume navigation2 is installed from Debian Repository on Desktop Machine and
map.yaml files are saved at /home/ubuntu/Desktop/maps where Username = ubuntu > change these scripts
if username is different
Location to save maps: mkdir -p /home/ubuntu/Desktop/maps
Optional Game Controller (Logitech F710 or Sony PS4): For F710 Input Standard Select "X" (NOT "D") &
Mode "Flight" Green LED OFF, location of configuration .yaml files: On Remote Desktop or SBC Robot -
mkdir -p /home/ubuntu/Desktop/config .
Optional ROS 2 USB Camera Package ros_drivers/usb_camera installed on SBC Robot with custom low
resolution & framerate configuration file "params10.yaml" saved to ~/Desktop/usb_camera/config
__________________________________________________________________________
ROBOT with cable plugged between USB Ports on RasPi Robot and Neato Botvac Diagnostic port (ls -l /
dev/ttyACM0)

From Desktop Machine, e.g. $ ssh sudo ssh ubuntu@rp4-ub22h.bv.localinto the SBC Robot

$ ros2 launch botvac_node botvac_base.launch.py

Optional Game Controller Joystick connected to ROBOT, provided ps4.config.yaml or f710.config.yaml
are saved to /user/ubuntu/Desktop/config folder

$ ros2 launch teleop_twist_joy teleop-launch.py config_filepath:='/home/ubuntu/Desktop/config/
ps4.config.yaml'
OR
$ ros2 launch teleop_twist_joy teleop-launch.py config_filepath:='/home/ubuntu/Desktop/config/
f710.config.yaml'

Optional Camera Image from a WebCam connected to a Robot USB port. Run from SBC RasPi Robot:
$ ros2 run usb_cam usb_cam_node_exe --ros-args --params-file /home/ubuntu/Desktop/usb_cam/config/
params10.yaml
____________________________________________________________________________
WORKSTATION Navigation & SLAM Launch SLAM Toolbox, rviz & Teleop in separate windows
Rename map.yaml & map.pgm when creating a new map

$ ros2 launch nav2_bringup bringup_launch.py use_sim_time:=False autostart:=True map:=/home/ubuntu/
Desktop/maps/map.yaml slam:=True
$ ros2 launch nav2_bringup rviz_launch.py . Launches rviz with nav2_default_view.rviz. Might need to
click "Robot Model"

$ ros2 run teleop_twist_keyboard teleop_twist_keyboard
_____________________________________________________________________________
With Optional Game Controller Joystick connected to either MACVM or Laptop Desktop, configureprovided
ps4.config.yaml or f710.config.yaml saved to /user/ubuntu/Desktop/config folder

$ ros2 launch teleop_twist_joy teleop-launch.py config_filepath:='/home/ubuntu/Desktop/config/
ps4.config.yaml'
OR
$ ros2 launch teleop_twist_joy teleop-launch.py config_filepath:='/home/ubuntu/Desktop/config/
f710.config.yaml'

MAC VM $ ros2 launch teleop_twist_joy teleop-launch.py config_filepath:='/home/ubuntu/Desktop/config/
f710.config.yaml' joy_dev:='/dev/input/js1'
Laptop Ubuntu 22.04 $ ros2 launch teleop_twist_joy teleop-launch.py config_filepath:='/home/ubuntu/
Desktop/config/f710.config.yaml' joy_dev:='/dev/input/js0'
To test Joystick: $ ros2 topic echo /joy OR
$ ros2 topic echo /cmd_vel
__________________________________________________________________________
Optional Camera Image from a WebCam connected to the a Robot USB port
$ ros2 run rqt_image_view rqt_image_view
____________________________________________________________________________
Save the MAP

$ ros2 run nav2_map_server map_saver_cli -f /home/ubuntu/Desktop/maps/map --free 0.196 --ros-args -p
save_map_timeout:=5000.0
Kill the Toolbox and rviz by closing Terminal Windows with CTRL-c
```

Relaunch SLAM

```
$ ros2 launch nav2_bringup bringup_launch.py use_sim_time:=False autostart:=True map:=/home/ubuntu/Desktop/maps/map.yaml
```

```
$ ros2 launch nav2_bringup rviz_launch.py
```

Click 2D Pose Estimate and place robot on map
Set Navigational Goal, Set 1 or more Waypoints

Diagnostics

```
$ ros2 run rqt_gui rqt_gui
$ rqt_gui  Select Plugins - Topics - Node Graph - Visualization
$ ros2 run rqt_graph rqt_graph
```

When Done: On the RasPi Robot, STOP the botvac_node with ctrl-C .