

Close the startup program

Close the startup program

rasberry pi

Temporarily close the startup program

Docker partial knowledge

Multiple terminals enter the same container

Run script

Permanently close the startup program

Exit the container

Run in the foreground

Background process

Startup script

Modify startup script

Run startup script

Turn off the Docker script that starts automatically at boot

Restore the Docker script that starts automatically at boot

Jetson motherboard

View the startup program

Temporarily close the startup program

Permanently close the startup program

Start the boot program

Since our image is factory-set to start the startup program, if you do not close the startup program and run other cases, some abnormal errors will occur!

The Dofbot of Raspberry Pi and Jetson motherboards have different methods of closing large startup programs!

raspberry pi

Temporarily close the startup program

Docker partial knowledge

Only some Docker commands are introduced here. For more details, please see the Docker chapter tutorial.

Multiple terminals enter the same container

Since the specified container will automatically start when we start it, we need to enter the same container and run the script that closes the startup program.

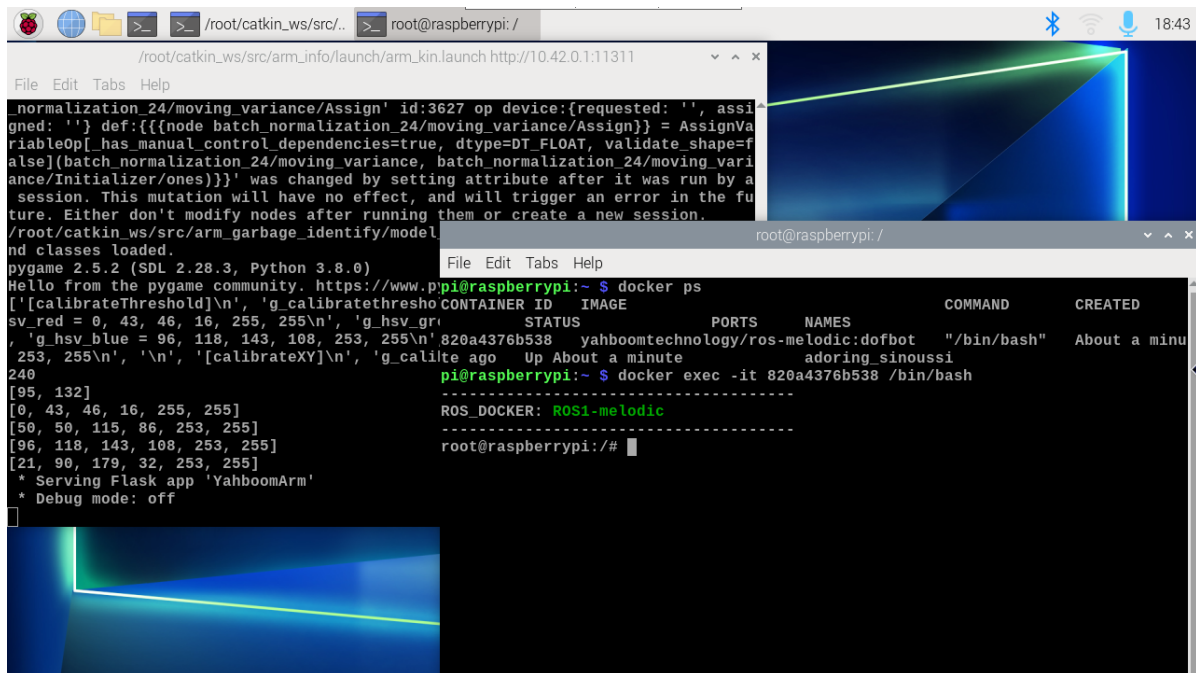
docker ps

Used to list currently running containers.

docker run

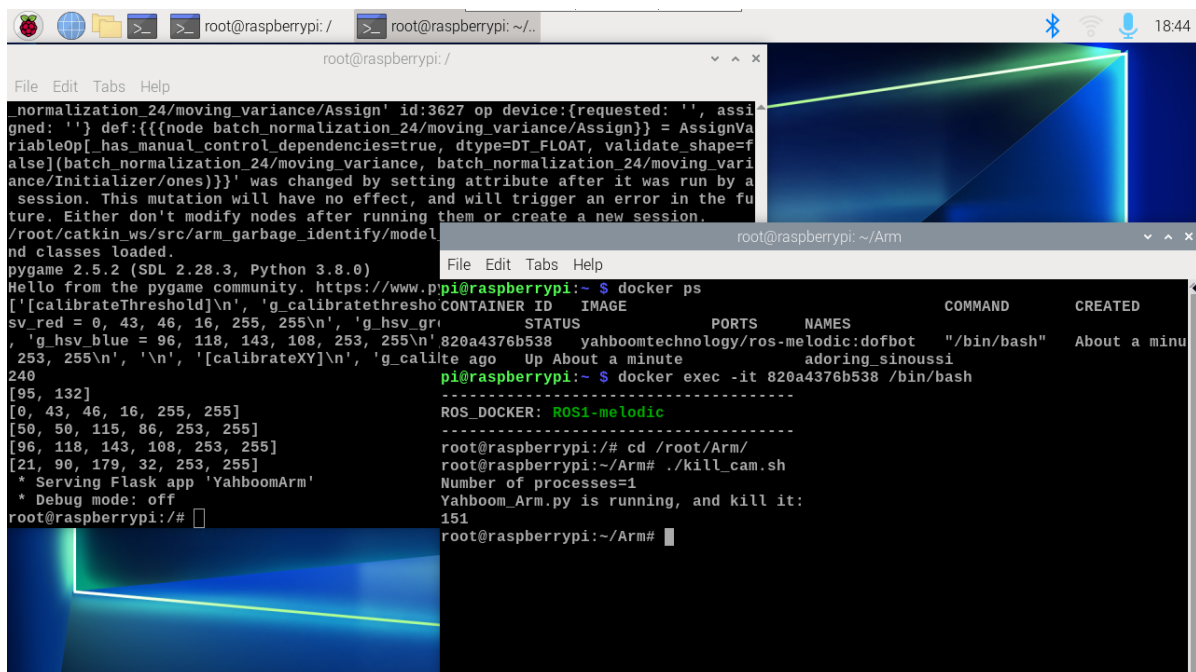
Used to create and start a new container instance.

```
# Start a container in interactive mode and execute the /bin/bash command within the container
docker exec -it 820a4376b538 /bin/bash
```



Run script

```
cd /root/Arm/
./kill_cam.sh
```

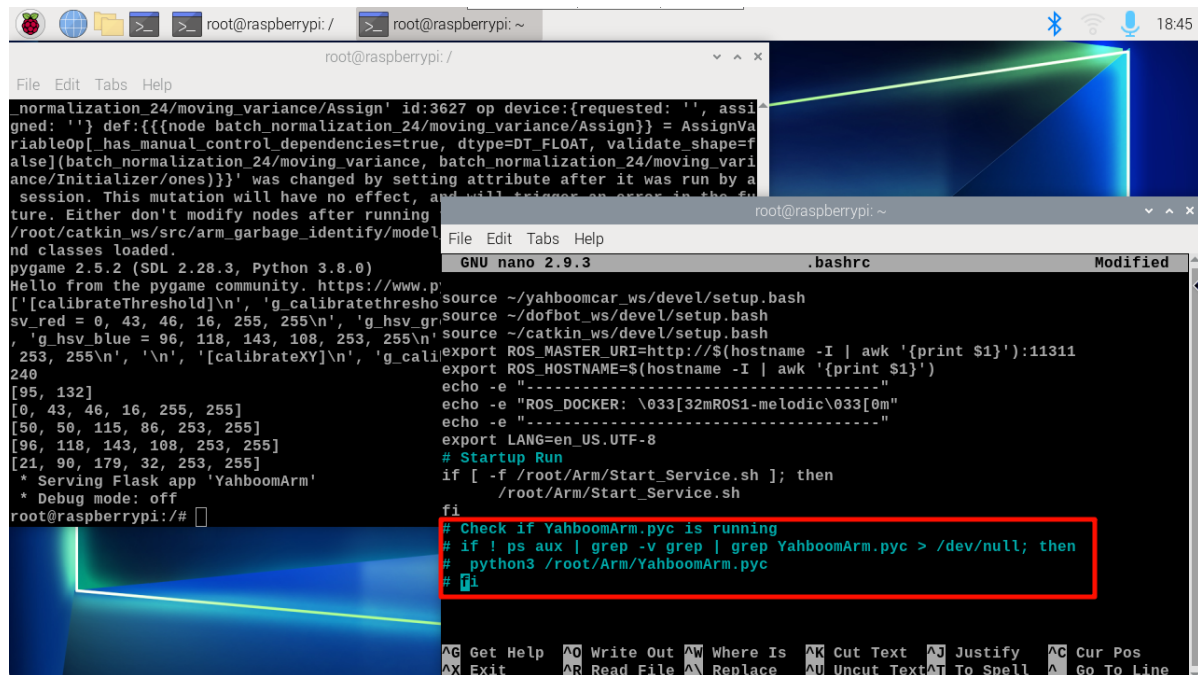


Temporarily shutting down a large program will only take effect on the currently running container. If you restart the container through a script, docker will rerun the startup program!

The script to start the large program is the .bashrc file added to docker, so every time a new container is started, the large startup program will be run. When multiple containers are entered and not closed correctly, the large program will prompt that the port is occupied!

Permanently close the startup program

Since our startup program is added to the .bashrc file, we need to comment or delete the relevant content of `YahboomArm.pyc` y in the .bashrc file: this content is at the bottom of the file



```
normalization 24/moving_variance/Assign' id:3627 op device:{requested: '', assigned: ''} def:{{{node batch_normalization_24/moving_variance/Assign}} = AssignVariableOp[has_manual_control_dependencies=true, dtype=DT_FLOAT, validate_shape=false][batch_normalization_24/moving_variance, batch_normalization_24/moving_variance/Initializer/ones]]' was changed by setting attribute after it was run by a session. This mutation will have no effect, and will trigger an error in the future. Either don't modify nodes after running /root/catkin_ws/src/arm_garbage_identify/model and classes loaded.
pygame 2.5.2 (SDL 2.28.3, Python 3.8.0)
Hello from the pygame community. https://www.pygame.org
['[calibrateThreshold]\n', 'g_calibratethresho
sv_red = 0, 43, 46, 16, 255, 255\n', 'g_hsv_gr
, 'g_hsv_blue = 96, 118, 143, 108, 253, 255\n'
253, 255\n', '\n', '[calibrateXY]\n', 'g_cal
240
[95, 132]
[0, 43, 46, 16, 255, 255]
[50, 50, 115, 86, 253, 255]
[96, 118, 143, 108, 253, 255]
[21, 90, 179, 32, 253, 255]
* Serving Flask app 'YahboomArm'
* Debug mode: off
root@raspberrypi:~#
```

```
GNU nano 2.9.3 .bashrc Modified
source ~/yahboomcar_ws/devel/setup.bash
source ~/dofbot_ws/devel/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://$(hostname -I | awk '{print $1}'):11311
export ROS_HOSTNAME=$(hostname -I | awk '{print $1}')
echo -e "-----"
echo -e "ROS_DOCKER: \033[32mROS1-melodic\033[0m"
echo -e "-----"
export LANG=en_US.UTF-8
# Startup Run
if [ -f /root/Arm/Start_Service.sh ]; then
    /root/Arm/Start_Service.sh
fi
# Check if YahboomArm.pyc is running
# if ! ps aux | grep -v grep | grep YahboomArm.pyc > /dev/null; then
#     python3 /root/Arm/YahboomArm.pyc
# fi
```

After the modification is completed, we need to submit a new version of the image:

First check the running containers:

```
docker ps
```

docker commit

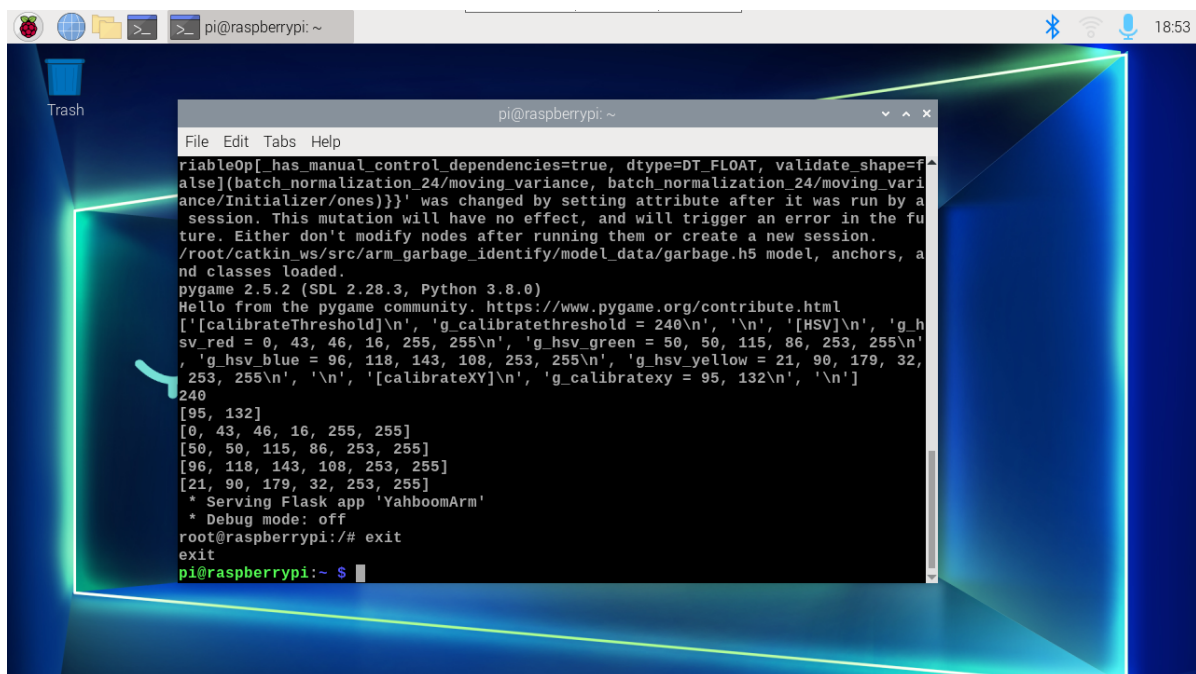
Used to save the current state of the container as a new image: Since docker already has 1.0 and dofbot labels, I will demonstrate using dofbot_v1.0 here.

```
# docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
docker commit 820a4376b538 yahboomtechnology/ros-melodic:dofbot_v1.0
```

Exit the container

Run in the foreground

If the container is running in the foreground, you can enter `exit` in the terminal to exit!

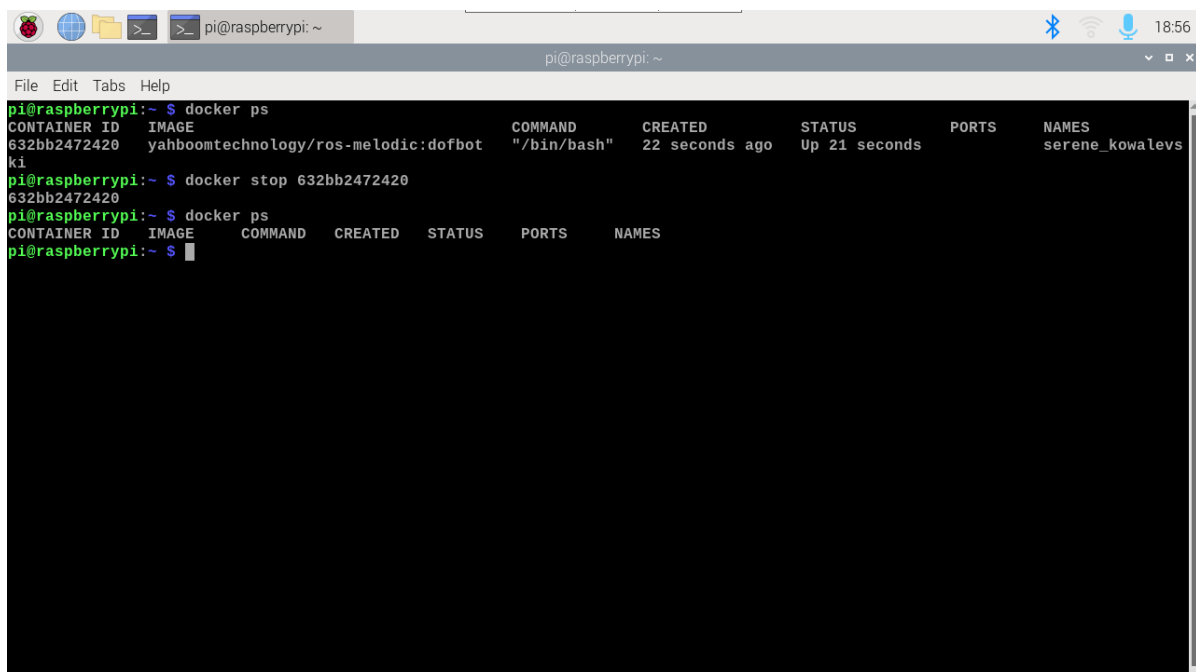


Background process

docker stop

Used to stop the specified container from running.

```
# docker stop [OPTIONS] CONTAINER
docker stop 632bb2472420
```



Startup script

Modify startup script

Startup script for self-starting at boot: Docker_Ros.sh

```
#!/bin/bash

# wait for the Docker service to start
```

```

while true; do
    if systemctl is-active --quiet docker; then
        echo "Docker service has been started"
        break
    else
        echo "The Docker service has not started, waiting..."
        sleep 1
    fi
done

# Docker start
xhost +

docker run -it --rm \
--privileged=true \
--net=host \
--env="DISPLAY" \
--env="QT_X11_NO_MITSHM=1" \
-v /tmp/.X11-unix:/tmp/.X11-unix \
--security-opt apparmor:unconfined \
-v /home/pi/Temp:/root/Temp \
--device=/dev/video0 \
--device=/dev/i2c-1 \
yahboomtechnology/ros-melodic:dofbot /bin/bash

```

Modify it to our newly submitted docker image:

```

#!/bin/bash

# wait for the Docker service to start
while true; do
    if systemctl is-active --quiet docker; then
        echo "Docker service has been started"
        break
    else
        echo "The Docker service has not started, waiting..."
        sleep 1
    fi
done

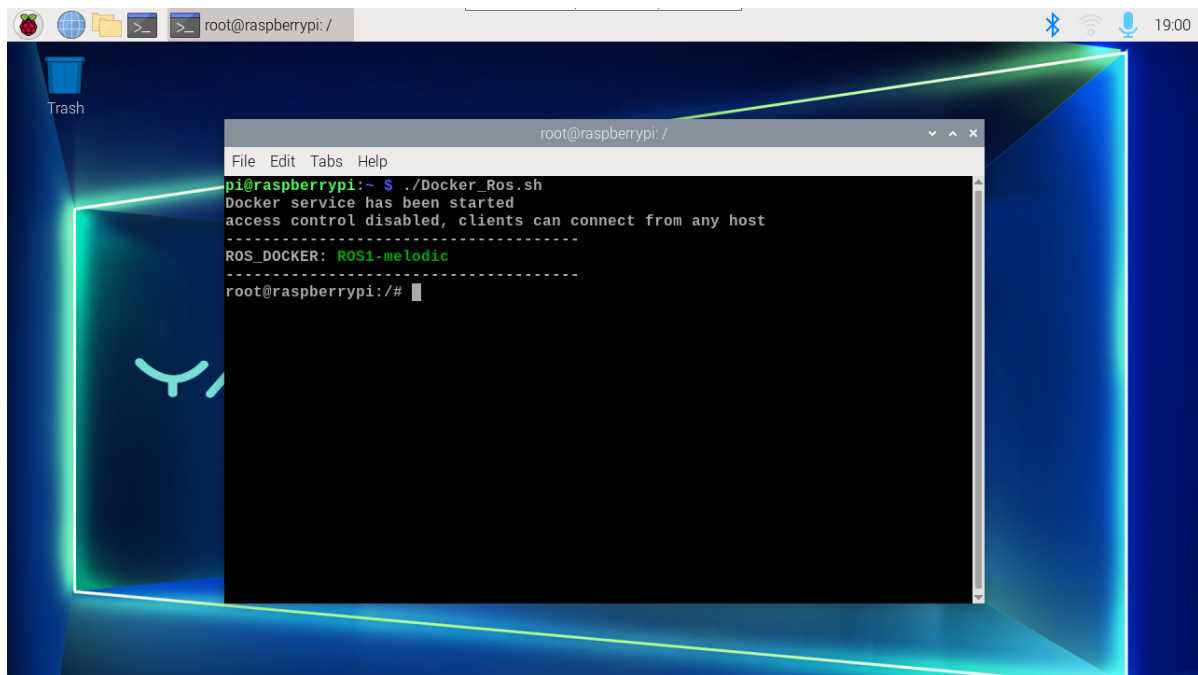
# Docker start
xhost+

docker run -it --rm \
--privileged=true \
--net=host \
--env="DISPLAY" \
--env="QT_X11_NO_MITSHM=1" \
-v /tmp/.X11-unix:/tmp/.X11-unix \
--security-opt apparmor:unconfined \
-v /home/pi/Temp:/root/Temp \
--device=/dev/video0 \
--device=/dev/i2c-1 \
yahboomtechnology/ros-melodic:dofbot_v1.0 /bin/bash

```

Run startup script

```
./Docker_Ros.sh
```



In this way, the big boot program will not be started. If you need to start the big boot program, change the specified container of the `Docker_Ros.sh` script to `yahboomtechnology/ros-melodic:dofbot /bin/bash`

That's it!

Turn off the Docker script that starts automatically at boot

Enter the following command to restart:

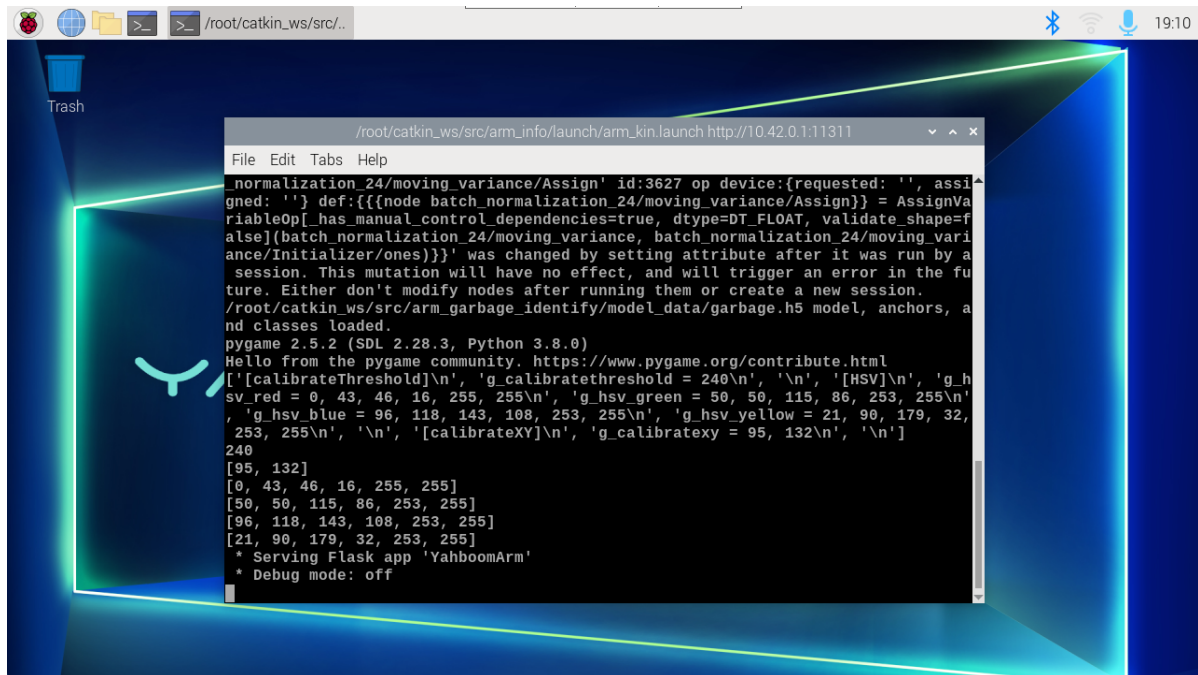
```
sudo rm -rf /home/pi/.config/autostart/Docker_Ros.desktop
```



Restore the Docker script that starts automatically at boot

Enter the following command to restart:

```
sudo cp Docker_Ros.desktop /home/pi/.config/autostart/
```



Jetson motherboard

View the startup program

Open a terminal and run:

```
ps -ef | grep YahboomArm.pyc
```

Temporarily close the startup program

Open a terminal and run:

```
sh ~/Dofbot/kill_YahboomArm.sh
```

Permanently close the startup program

Open a terminal and run:

```
sudo systemctl stop yb-bigProgram.service
sudo systemctl disable yb-bigProgram.service
```

Start the boot program

Open a terminal and run:

```
sudo systemctl restart yb-bigProgram.service
sudo systemctl enable yb-bigProgram.service
```