

タブ 1

DayServiceRB-kata Ryzen9を自動起動

目標は、デイサービスで働くロボット。Ryzen9においても、私が不在時ロボットが使えるように、電源ボタンで起動した後、全ての起動プロセスを自動化する。

構成

・T8\$	Client PC CF-T8	Debian10
・noetic3\$	T8 Docker Container	
・Ryzen9\$	Server PC Minis Forum HX90	Debian12
・noetic\$	Ryzen9 Docker Container	
・W7day	DayServiceRB	Debian11

目次

[概要](#)

[自動ログイン](#)

[設定ファイルの作製](#)

[設定の適用](#)

[再起動](#)

[roslaunchの実行](#)

[.bashrcの編集](#)

[docker起動スクリプト作製](#)

[roslaunch起動スクリプト作製](#)

[実行権の付与](#)

[再起動](#)

[今後の予定](#)

[参考](#)

概要

基本、W7dayと同じ作用である。

 20250325DayServiceRB-kata W7dayを自動起動進める。

自動ログイン

設定ファイルの作製

```
yo@Ryzen9:~$ sudo mkdir -p
/etc/systemd/system/getty@tty1.service.d
yo@Ryzen9:~$ cd /etc/systemd/system/getty@tty1.service.d
yo@Ryzen9:/etc/systemd/system/getty@tty1.service.d$ sudo vi
autologin.conf
```

```
[Service]
```

```
ExecStart=
```

```
ExecStart=-/sbin/agetty --autologin yo --noclear %I $TERM
```

設定の適用

```
yo@Ryzen9:~$ sudo systemctl daemon-reexec
yo@Ryzen9:~$ sudo systemctl restart getty@tty1
```

再起動

```
yo@Ryzen9:~$ sudo reboot
```

以上の設定で、自動ログイン完了

roslaunchの実行

.bashrcの編集

.bashrcの最終行に以下を追記する。

```
yo@Ryzen9:~$ vi .bashrc
```

```
if [ -z "$DISPLAY" ] && [ "$(tty)" = "/dev/tty1" ]; then
    /home/yo/start_day_service_rb.sh;
fi
```

docker起動スクリプト作製

```
yo@Ryzen9:~$ vi start_day_service_rb.sh
```

```
docker start ros_noetic
docker exec ros_noetic /home/yo/start_all.sh
```

```
yo@Ryzen9:~$ chmod +x start_day_service_rb.sh
```

roslaunch起動スクリプト作製

```
noetic yo@Ryzen9:~$ vi start_all.sh
```

```
#!/bin/bash
source /opt/ros/noetic/setup.bash
source ~/catkin_ws/devel/setup.bash
#ros通信のための設定
export ROS_MASTER_URI=http://192.168.11.8:11311
export ROS_IP=192.168.11.8
roslaunch ros_ggml all-rev1.launch
```

実行権の付与

```
noetic yo@Ryzen9:~$ chmod +x start_all.sh
```

再起動

Ryzen9でroscoreが起動していれば、問題なく起動する。

今後の予定

1. PC名称Ryzen9のDocker noetic containerで/audio topicを購読する。
2. 購読した/audioよりVoskを用いてテキスト変換
3. 変換したテキストよりggmlで回答テキストの生成
4. pyopenjtalkを用いてテキスト音声変換
5. 音声を/audio topicに変換し、T8で購読/audio topicをスピーカに出力
6. T8の液晶にロボットの顔を表示し、ggml変換中に考え中の顔を表示する
7. servoを使った機体のデザイン決定
8. day_service_rb_kataの動きを作る
9. 実際の機体の作製
 - ・制御基板の検討

- ・腕の作製

- ・制御盤の固定

10. ros noeticで構成されるプログラムの起動をroslaunchで纏める

- ・全てを一つのlaunchファイルで構成するのではなく、各PC毎にlaunchファイルを作製する。

11. 考え中のときに、腕を動かす。

12. day_service_rb_kataの動き、ウェーブを作る。

13. 改善点

大きく分けて3点である。

a)音声再生時にノイズが入る

- ・音声再生時にノイズが発生することがある。

- pyopenjtalkをRyzen9からT8に移動

- pyopenjtalkからOpenjtalkに変更

- ・エラーメッセージが、出力されていた。

Geminiに確認すると、PCの負荷が大きくなり、音声出力の処理が間に合わないとのこと。なので、できるだけ軽く保つことで解決。

b)音声テキスト変換

- ・動作タイミング:再生時にマイクで拾ってしまう

- ・ggmlの起動について:反応しすぎるのはどうかと。

14.背面保護ケース作製

15電源ボタンからroslaunch自動起動

<ここまで終了>

13. 改善点

a)不用な言葉には回答しない。

- ・「えっと」の場合、ggmlを起動しない。

b)回答について、頭が悪い

- ・GPUを使った生成AIを用いる。

参考

☰ 20250325DayServiceRB-kata W7dayを自動起動