

Docker と Ros2 によるシミュレーション環境の構築

1 これだけ知ってればまず動く！

PC に環境が出来あがった状態であることが前提のセクションです。まだ作ってない人は、3 とりあえず環境作ろう を先に見てください

1.1 フローチャート

windows(または Mac 上で)

`docker start hr_education` (または `docker start hr_arm_robot_simenv`)

ウェブブラウザで次を検索

`http://127.0.0.1:6080`

ウェブ上の linux 環境で Terminator を起動。Terminator 上で次を実行

`ros2 launch angle_control angle_control_gazebo_included.launch.py`

シミュレーション環境が現れます。

続いて別の Terminator を起動して、目的のフォルダに移動

`cd ~/hr_ws/src/crane_x7_hr_edu/move_robot/examples/` (自分のコードの場合は `examples` を `motions` に変更)

フォルダー下で用意したコードを実行します

`python3 your_code_name.py` (`your_code_name` は自分のコードの名前に変更)

自分のコードを編集したかったら次のコードで VSodium を開きましょう

`codium .`

終わったら必ずコンテナを必ず止めましょう。windows(または Mac 上で)

`docker stop hr_education`(または `docker stop hr_arm_robot_simenv`)

1.2 環境の立ち上げ方 (詳しく)

環境は docker でできているので、まずはコンテナを起動しましょう。起動する前に対象の名前を確認しておきます。windows powershell またはターミナルを開いて、

`docker ps -a`

を実行してください。多分図 12 のような表示が出ると思います (皆さんの PC だと上の一つだけかもしれないで

すが問題ありません)。IMAGE タグの下が hr_education, または hr_arm_robot_simenv となっているのが、前回

```
PS C:\Users\hibik> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d8ec7332772b	hr_education	"/bin/bash -c /entry..."	4 days ago	Exited (0) 3 days ago		hr_education
aa8df7077c1f	tiryoh/ros2-desktop-vnc:humble	"/bin/bash -c /entry..."	6 days ago	Exited (0) 4 days ago		great_villani

図 1 docker ps

作った Docker Container です。目的のコンテナが見つかったら、NAMES のところを確認してください。この場合 hr_education が使用するものです。

さて、コンテナをスタートさせるために次のコマンドを入力してください。

```
docekr start hr_education
```

図 2 のように表示されれば ok です。

```
PS C:\Users\hibik> docker start hr_education
hr_education
PS C:\Users\hibik>
```

図 2 docker start your_container_name

では、このコンテナのデスクトップ画面に行ってみましょう。ウェブブラウザを開いて、

<http://127.0.0.1:6080>

を検索してください。

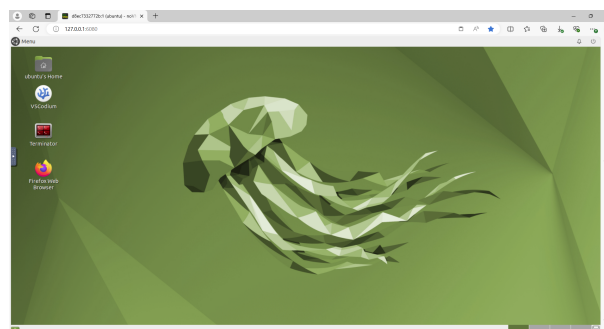


図 3 ubuntu desktop

図 3 のようになれば ok です。これから先はこのブラウザ上のデスクトップで作業を進めていきます。基本的にコマンドは Terminator というところに打ち込んでいきます。

ではまず、用意してあるアームロボットのシミュレーションを立ち上げてみましょう。Terminator を開いて

```
ros2 launch angle_control angle_control_gazebo_included.launch.py
```

を実行しましょう (図 4)

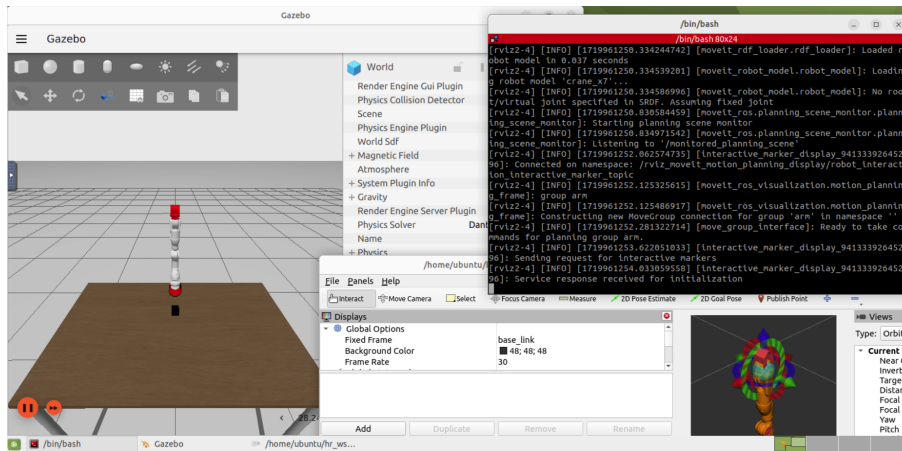


図 4 simulator

初回は時間がかかります。また、ウェブ上で動かしているのがバグることも多いので、うまくいってなさそうだったら `ctrl+C` で一旦停止し、もう一度実行してください。

では、プログラムのコードがあるところに行きましょう。新しい Terminator を開き、次のコマンドで目的の場所まで行きます。

```
cd ~/hr_ws/src/crane_x7_hr_edu/move_robot/examples/
```

ではコードを実行してみましょう。

```
python your_code_name.py
```

`your_code_name` は実行したいコードの名前に置き換えてください。今回はあらかじめ用意してある `wave.py` を使用しています。図 5 のように 2 つの Terminator に謎の文字列が表示されて、ロボットが動き出します。

動かなかったらシミュレータを再度動かしてみてください 1.2

これで環境を立ち上げることができるようになりました!お疲れ様です。

2 コードの書き方

さて、コードを書いていきましょう。皆さんのコードは `motions` というところに書いていってもらいます。まずは目的の場所に移動して

```
cd ~/hr_ws/src/crane_x7_hr_edu/move_robot/
```

VScode を起動しましょう 図 6。

```
codium .
```

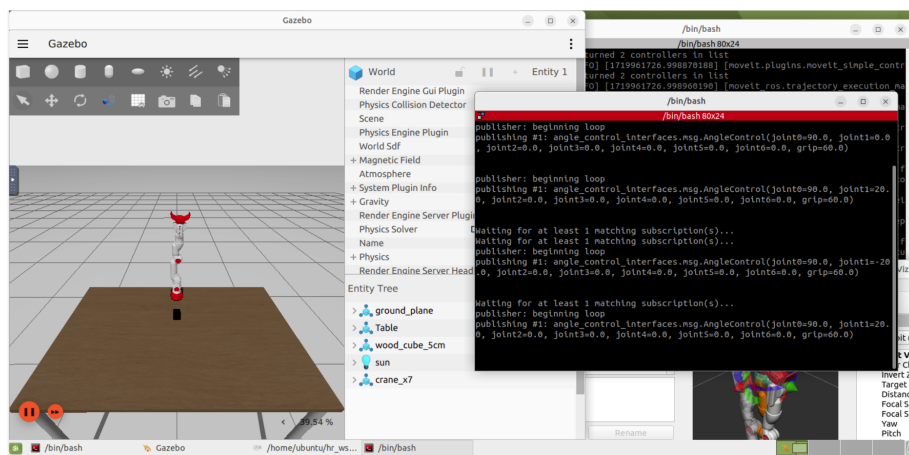


図5 move robot

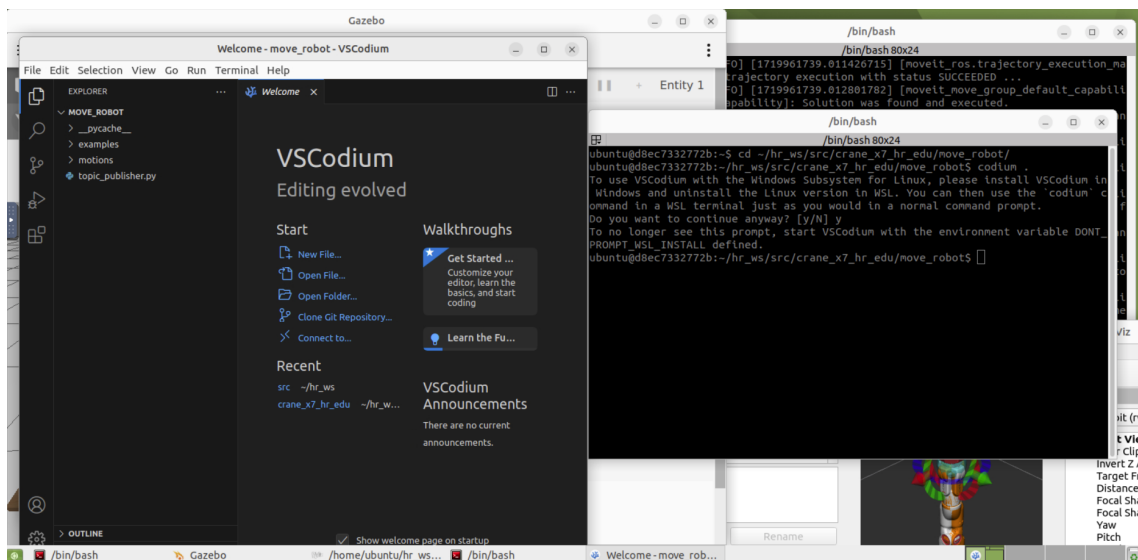


図6 open vscode

コードを書きやすいように色々下設定を済ませてあるので、まずは作成した.py ファイルに次の呪文を書き入れてください。python ファイルは、motions フォルダ (なかったら作ってください) 直下に入れましょう

```
import sys
```

```
sys.path.append("..")
```

```
from topic_publisher import angle_write
```

これで関数「angle_write」が使えるようになります。

angle_write() 関数は大きさが8のリストを受け取り、ロボットに角度情報を送るプログラムです。リストの0か

ら 6 までは関節角度を、7 がハンドの角度を指定します。例として wave.py を見てみましょう。

```
import sys

sys.path.append("..")
from topic_publisher import angle_write

wave = [
    [90,0,0,0,0,0,0,60],
    [90,20,0,0,0,0,0,60],
    [90,-20,0,0,0,0,0,60],
    [90,20,0,0,0,0,0,60],
    [90,-20,0,0,0,0,0,60],
    [90,0,0,0,0,0,0,60]
]

for value in wave:
    angle_write(value)
```

wave=... のところで 6 つの角度を順番に指定した 2 重リストを作っています。後は for 文で指定した角度を順番に回し、ロボットを動かしています。ここで、機体の限界を超える角度を入力してもロボットは反応しません。また、指示角度にロボットが動ききるまでは次の命令を受け付けない仕様になっています。

3 とりあえず環境作ろう

環境を作るため、Git と Docker をダウンロードする必要があります。

Mac の人は Git と Docker Desktop をそれぞれ調べ、ダウンロードしてください。

Windows の人は powershell を開いて次のコマンドをそれぞれ打ち込みましょう。

```
winget install Git.Git
winget install Docker.DockerDesktop
```

かなり時間がかかりますがダウンロードできます。

ダウンロードが終わったら、PC を再起動してください。

再起動後にもう一度 powershell を開き

```
git --version
docker --version
```

をそれぞれ実行しましょう。図 7 のような表示が出たら ok です。

それでは、Docker の Image を作成していきましょう。まずはいつもロボメカの作業をしているディレクトリに移動してください。移動先で次のコマンドを実行します。

```
git clone -b master https://github.com/rmf-hru-wg/hr_arm_robot_simenv.git
```

```

PS C:\Users\hibik> git --version
git version 2.42.0.windows.2
PS C:\Users\hibik> docker --version
Docker version 26.1.4, build 5650f9b
PS C:\Users\hibik> |

```

図7 version information

ディレクトリで

```
ls ./hr_arm_robot_simenv
```

を実行して図8のように表示されたら成功です。

```

PS C:\Users\hibik\Documents\Robomech\HR\Induction\Docke> ls ./hr_arm_robot_simenv

ディレクトリ: C:\Users\hibik\Documents\Robomech\HR\Induction\Docke\hr_arm_robot_simenv

Mode                LastWriteTime         Length Name
----                -
d-----          2024/06/28    14:09             src
d-----          2024/07/02    17:15      tex_documents
-a-----          2024/06/29     9:20        824 Dockerfile

```

図8 ls check

では Image を作っていきましょう。まずは先ほど clone したディレクトリに移動して

```
cd hr_arm_robot_simenv
```

image を作ります。

```
docker image build -t hr_arm_robot_simenv .
```

```

PS C:\Users\hibik\Documents\Robomech\HR\Induction\Docke> cd ./hr_arm_robot_simenv\
PS C:\Users\hibik\Documents\Robomech\HR\Induction\Docke\hr_arm_robot_simenv> docker image build -t hr_arm_robot_simenv .
[*] Building 2.4s (8/10)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 863B
=> [internal] load metadata for docker.io/tiryoh/ros2-desktop-vnc:humble
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 1.75kB
=> [1/6] FROM docker.io/tiryoh/ros2-desktop-vnc:humble
=> CACHED [2/6] RUN apt-get update && apt-get install -y sudo && apt-get install -y nano && rm -rf /var/lib/apt/lists
=> CACHED [3/6] RUN mkdir -p /home/ubuntu/hr_ws/src && cd /home/ubuntu/hr_ws/src && git clone -b ros2 https://github.com/rt-net/crane_x7_ros.git &
=> CACHED [4/6] COPY src/crane_x7_hr_edu/ /home/ubuntu/hr_ws/src/crane_x7_hr_edu/
=> [5/6] RUN cd /home/ubuntu/hr_ws && source /opt/ros/humble/setup.bash && colcon build

```

図9 build image

図9のように青い行がたくさん表示されたら成功です。30分くらいかかるので気長に待ちましょう。なんかうまくいかなかった場合は8トラブルシューティングへ

では作成したイメージを確認しましょう

```
docker images
```

図 10 のように表示されたら ok です。皆さんの場合は REPOSITORY の下の名前が hr.education ではなく、hr_arm_robot_simenv となるはずです。

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hr.education	latest	a8e79fba4126	5 days ago	7.96GB
tiryoh/ros2-desktop-vnc	humble	8c5383c5f835	9 days ago	7.24GB
osrf/ros	humble-desktop-full	59961398ad13	3 weeks ago	3.83GB

図 10 images

ではこの image を実際に使えるようにしてみましょう。

```
docker run -p 6080:80 --name hr.education hr_arm_robot_simenv
```

大体図 11 のようなものが表示されれば ok です。

```
* enable custom user: ubuntu
useradd: warning: the home directory /home/ubuntu already exists.
useradd: Not copying any file from skel directory into it.
  set default password to "ubuntu"
grep: /home/ubuntu/.bashrc: No such file or directory
=====
NOTE 1: --security-opt seccomp=unconfined flag is required to launch Ubuntu Jammy based image.
See https://github.com/tiryoh/docker-ros2-desktop-vnc/pull/56
NOTE 2: Before stopping to commit docker container to new docker image, log out first.
See https://github.com/tiryoh/docker-ros2-desktop-vnc/issue/131
=====
2024-07-03 00:20:36,678 INFO Included extra file "/etc/supervisor/conf.d/supervisord.conf" during parsing
2024-07-03 00:20:36,678 INFO Set uid to user 0 succeeded
2024-07-03 00:20:36,687 INFO RPC interface 'supervisor' initialized
2024-07-03 00:20:36,687 CRIT Server 'unix_http_server' running without any HTTP authentication checking
2024-07-03 00:20:36,687 INFO supervisord started with pid 43
2024-07-03 00:20:37,691 INFO spawned: 'novnc' with pid 44
2024-07-03 00:20:37,694 INFO spawned: 'vnc' with pid 45
2024-07-03 00:20:37,694 WARN received SIGINT indicating exit request
2024-07-03 00:20:37,697 INFO waiting for novnc, vnc to die
2024-07-03 00:20:37,698 INFO stopped: vnc (terminated by SIGTERM)
2024-07-03 00:20:37,699 INFO stopped: novnc (terminated by SIGTERM)
```

図 11 docker run

これで container が動き出しました。確認してみましょう。ほかの shell を開いて、

```
docker ps
```

図 12 のように表示されていれば ok です。大事なのは NAMES のところで、ここが hr.education なら大丈夫です。

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d8ec7332772b	hr.education	"/bin/bash -c /entry..."	5 days ago	Up 1 second	0.0.0.0:6080->80/tcp	hr.education

図 12 docker ps

使い終わったら必ず

```
docker stop hr.education
```

で動作を止めましょう。PC に悪いです。

次にコンテナを再起動したいときは

```
docekr start hr.education
```

で始めることができます。ではコンテナを再度起動して (docker start hr.education) デスクトップに接続しましょう。ブラウザで

`http://127.0.0.1:6080`

を検索してください。開けたら Terminator を起動しましょう。開けたら、

```
echo 'source /home/ubuntu/hr_ws/install/local_setup.bash' >> ~/.bashrc
```

と打ち込んでください (図 13)。

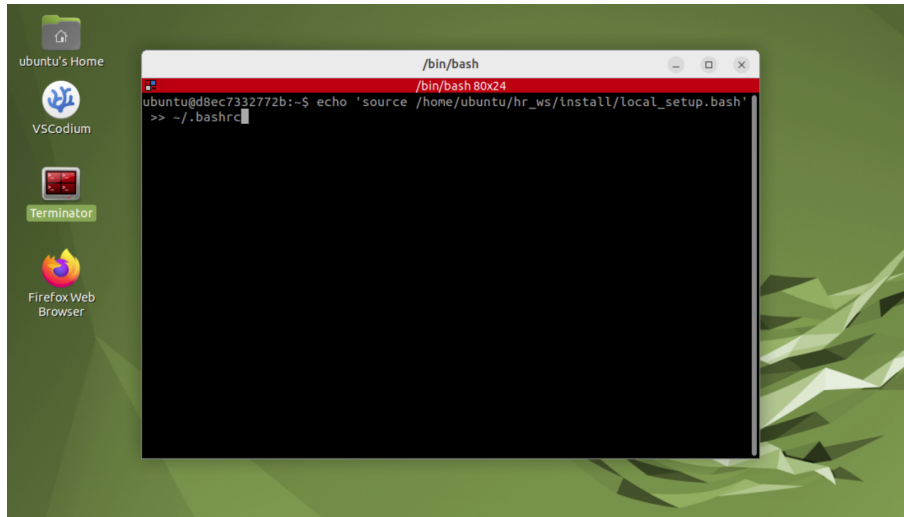


図 13 echo

Terminator を再起動したら環境完成です。

4 Docker とは

後で書きます。

5 Ros とは

後で書きます。

6 書いてあるコードの説明

後で書きます。

7 これらを使うと何ができるか？

後で書きます。

8 トラブルシューティング

8.1 ありそうなミス

ubuntu のデスクトップ (127.0.0.1:6080) では貼り付けのコマンドは Ctrl+Shift+V、コピーは Ctrl+Shift+C。

~を書き忘れている。

echo コマンドを打った後に Terminator を再起動してない

python3 じゃなくて python って打ってる

python3 を実行するディレクトリが違う (example または motions じゃないと動かない)

8.2 docker image build で Dockerfile not found と言われた

git clone が間違っている可能性があります。一度 hr_arm_robot_simenv フォルダを削除し、再度

git clone -b master https://...

を実行しましょう。-b master を忘れないようにしましょう

8.3 docker image build が colcon build のところで止まる

PC のスペックが足りてない可能性があります。Dockerfile を次のように編集してください。

```
FROM tiryoh/ros2-desktop-vnc:humble
```

```
RUN apt-get update \
```

```
    && apt-get install -y sudo \
```

```
    && apt-get install -y nano && rm -rf /var/lib/apt/lists
```

```
RUN mkdir -p /home/ubuntu/hr_ws/src && cd /home/ubuntu/hr_ws/src \
```

```
    # Download crane_x7 repositories
```

```
    && git clone -b ros2 https://github.com/rt-net/crane_x7_ros.git \
```

```
    && git clone -b ros2 https://github.com/rt-net/crane_x7_description.git \
```

```
    # Install dependencies
```

```
    && rosdep -y update && apt-get update \
```

```
    && rosdep install -r -y -i --from-paths .
```

```
# copy some source files
```

```
COPY src/crane_x7_hr_edu/ /home/ubuntu/hr_ws/src/crane_x7_hr_edu/
```

```
# build all the file
```

```
RUN cd /home/ubuntu/hr_ws \
```

```
&& source /opt/ros/humble/setup.bash \  
&& colcon build --packages-skip angle_conntrol \  
&& colcon build --packages-select angle_control
```

```
RUN echo 'source /home/ubuntu/hr_ws/install/local_setup.bash' >> ~/.bashrc
```