

Requirements:

OS: Ubuntu 22.04

Python: 3.10.12

In Ubuntu Software:

Arduino

Terminator (sudo apt install terminator)

Brave (Optional, browser without adds)

Install pip:

sudo apt install python3-pip

Install VS Code:

sudo snap install code --classic

Install VS Code Extensions:

ROS – Developer has to be Microsoft

Python – Developer has to be Microsoft

Turn on auto save: File -> Auto Save OR File -> Preferences -> Settings → (Write “auto save”) -> Select afterDelay

Install Dynamixel Wizard:

https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_wizard2/

Install Git and configure:

<https://www.digitalocean.com/community/tutorials/how-to-install-git-on-ubuntu-22-04>

To create a github SSH authentication key follow this tutorial, but DO NOT PUT PASSWORD:

Hit Enter when asked for password:

<https://www.youtube.com/watch?v=WgZlv5HI44o>

Install NVIDIA drivers:

<https://www.cherryservers.com/blog/install-cuda-ubuntu>

(Follow until step 5)

Arduino Libraries:

Sketch → Include Libraries → Manage Libraries

Adafruit NeoPixel

Install ROS2 humble:

<https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html>

Follow Tutorials (include ROS2 installation):

<https://www.youtube.com/playlist?list=PLLSegLrePWgJudpPUof4-nVFHGkB62Izy>

This tutorial includes a lot more than installing ROS2 and subs and pubs examples, it does:

- adds ws sources to ~/.bashrc

To work with UFACTORY xArm 6:

- <https://www.ufactory.cc/ufactory-studio/>

- Download Linux

To install realsense-viewer:

Follow the following link until the part that states: "Uninstalling the packages"
https://github.com/IntelRealSense/librealsense/blob/master/doc/distribution_linux.md

To install librealsense and some more necessary libraries:

First follow these instructions:

<https://dev.intelrealsense.com/docs/compiling-librealsense-for-linux-ubuntu-guide>

Then follow the "Step 3: Install Intel® RealSense™ ROS2 wrapper":

<https://github.com/IntelRealSense/realsense-ros/blob/ros2-development/README.md>

Please check ANEX to see possible errors during COLCON BUILD:

To save videos for dataset:

- sudo apt-get install ffmpeg x264 libx264-dev
- ros2 run charmie_debug save_videos_for_dataset
- video is saved at home directory with date: "2024-07-04 00-14-19charmie.avi"
- to put in roboflow we must convert using ffmpeg to mp4 using:
ffmpeg -i 2024-07-04 00-14-19charmie.avi 2024-07-04_00-14-19_charmie.mp4

PACKAGES:

low_level:

pip install pyserial

ps4_controller:

pip install pyPS4Controller

neck_dynamixel:

pip install dynamixel-sdk

xarm:

Follow steps in: [https://github.com/xArm-Developer/xarm_ros2/tree/humble?](https://github.com/xArm-Developer/xarm_ros2/tree/humble?tab=readme-ov-file)

tab=readme-ov-file

sudo apt install ros-foxy-xacro ros-foxy-joint-state-publisher-gui

sudo apt install ros-foxy-gazebo-ros-pkgs

sudo apt install ros-foxy-ros2-control ros-foxy-ros2-controllers ros-foxy-gazebo-

ros2-control

yolos:

pip install ultralytics

to get the characteristics in yolo_pose:

pip install keras

pip install tensorflow

audio:

```
pip install SpeechRecognition
pip install pulsectl
sudo apt-get install portaudio19-dev
pip install PyAudio
    if you get an ffmpeg error:
        sudo apt update
        sudo apt install ffmpeg
```

receptionist:

```
pip install face_recognition
```

debug_visual:

```
python3 -m pip install pygame-widgets
```

speakers novos:

- pip install TTS
- export PATH=\$PATH:/home/utilizador/.local/bin (so tts commands can be used in terminal)
- pip install pydub
- (editar depois install espeak)
- pip install pygame

Possible ROS2 Errors:

When doing colcon build after creating the first package:

```
"/usr/lib/python3/dist-packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning:
setup.py install is deprecated. Use build and pip and other standards-based tools.
```

```
warnings.warn(
---"
```

```
pip3 list | grep setuptools
pip3 install setuptools==58.2.0
pip3 list | grep setuptools
```

If you have error such as the following when doing colcon build:

```
Finished <<< charmie_point_cloud [10.3s]
```

```
--- stderr: charmie_restaurant
```

```
/home/charmie/.local/lib/python3.10/site-packages/setuptools/command/easy_install.py:157:
```

```
EasyInstallDeprecationWarning: easy_install command is deprecated. Use build and pip and other
standards-based tools.
```

```
warnings.warn(
```

It means that you have the wrong version of setuptools. For ROS a specific version must be installed: setup tools version 58.2.0 (last version to work with ros2 python packages without any warnings)

```
pip install setuptools==58.2.0 ( all above versions do not work )
```

Please check for more info:

<https://answers.ros.org/question/396439/setuptoolsdeprecationwarning-setuptools-install-is-deprecated-use-build-and-pip-and-other-standards-based-tools/>

when doing colcon build realsense packages (on charmie_ws) it might get some erros related with gazebo, install gazebo using:

The error indicates that your workspace depends on the `gazebo_ros` package, but it is either not installed or not properly sourced in your environment. Here's how to resolve this issue:

1. Install the `gazebo_ros` Package

`gazebo_ros` is part of the ROS 2 Humble distribution. Install it by running:

```
sudo apt update
sudo apt install ros-humble-gazebo-ros-pkgs
```

2. Verify the Package Installation

Check if `gazebo_ros` is installed:

```
ros2 pkg list | grep gazebo_ros
```

You should see `gazebo_ros` and related packages listed.

3. Set `CMAKE_PREFIX_PATH` if Needed

If you installed `gazebo_ros` but CMake still cannot find it, manually set the `CMAKE_PREFIX_PATH` to include the installation directory of `gazebo_ros`.

Add this to your `~/ .bashrc` or run it in your terminal:

```
export CMAKE_PREFIX_PATH=$CMAKE_PREFIX_PATH:/opt/ros/humble
```

Then source your `.bashrc`:

```
source ~/.bashrc
```

--- stderr: `xarm_controller` CMake Error at `CMakeLists.txt:38 (find_package)`: By not providing "Findhardware_interface.cmake" in `CMAKE_MODULE_PATH` this project has asked CMake to find a package configuration file provided by "hardware_interface", but CMake did not find one. Could not find a package configuration file provided by "hardware_interface" with any of the following names: `hardware_interfaceConfig.cmake` `hardware_interface-config.cmake` Add the installation prefix of "hardware_interface" to `CMAKE_PREFIX_PATH` or set "hardware_interface_DIR" to a directory containing one of the above files. If "hardware_interface" provides a separate development package or SDK, be sure it has been installed.

1. Install `ros-humble-ros2-control`

The `hardware_interface` package is provided by the `ros2_control` framework. Install it using:

```
sudo apt update
sudo apt install ros-humble-ros2-control ros-humble-ros2-controllers
```

The display FACE on CHARMIE does not map the input touchscreen to the display. What happens is that when you click on the display touchscreen it is like you are pressing with the mouse but on the PC screen and not on the display. To use the display CHARMIE face you have to change from: Ubuntu on Wayland to Ubuntu on Xorg. Wayland handles input devices and outputs differently, and currently **does not provide an official xinput-like way to map touchscreen input to specific displays**.

The test this switch you can do:

- **Log out** of your current session (don't restart — just log out).
- On the **login screen**, click your **username**.
- Before entering your password, in the **bottom-right corner** of the screen is a **gear icon** .
- It appears **after** you select your user, and **before** you type your password.
- Click the **gear icon** — a small menu will pop up.
- Choose **“Ubuntu on Xorg”** or **“Ubuntu (X11)”**.
- Now type your password and log in.

However if you restart the PC it goes back to Ubuntu on Wayland. To make this a definitive change you have to do the following:

You can **force GDM to always start Xorg** instead of Wayland by editing its config file:

```
sudo nano /etc/gdm3/custom.conf
```

Then **uncomment** (remove #) or set this line:

```
WaylandEnable=false
```

This tells GDM to **disable Wayland entirely**, so it always boots into Xorg for every user.

After saving (Ctrl+O), reboot.

Now it will **default to Xorg**, and your touchscreen mapping will always work.

Now run charmie face to test if you can press the face and it detects.

To check the id (device_id) of the display touchscreen, you can run:

```
xinput list
```

And to check the name (output_name) of the display touchscreen you can run:

```
xrandr
```

In sum after you switch to Ubuntu on Xorg you have to run the following command:

```
xinput map-to-output <device_id> <output_name>
```

For example in Tiago's PC it is:

```
xinput map-to-output 10 DP-1-2
```

I think the device_id may change from PC to PC but the output_name is the same.

The face_display already does this subprocess automatically to map the input touchscreen. So you only need to make sure you are on Ubuntu on Xorg.

Install Livox MID-360:

First Install the Livox Viewer 2 – Ubuntu:

<https://www.livoxtech.com/downloads>

For the Livox ROS Driver 2 and SDK, follow the steps in:

https://github.com/Livox-SDK/livox_ros_driver2

In src/livox_ros_driver2 there is an executable file named:

build_tr.sh

To build this package we must execute this file:

./build_tr.sh

this package from livox can not be built from the standard charmie_ws colcon build – symlink-install

Links for new team members:

RoboCup@Home official website:
<https://athome.robocup.org/>

RoboCup@Home official Github:
<https://github.com/RoboCupAtHome/>

Rulebook Link:
<https://robocupathome.github.io/RuleBook/rulebook/master.pdf>

CHARMIE Project Github:
https://github.com/SparkRibeiro21/charmie_ws

CHARMIE Dataset:
https://github.com/SparkRibeiro21/charmie_ws/blob/main/objects/LAR_objects/LAR%20Dataset%20Objects.pdf

Requirements nos vossos Pcs:

OS: Ubuntu 22.04 LTS (Aconselho vivamente que façam dualboot, máquina virtual vai dar erros no ROS)

ROS2: Humble

Para instalarem ROS2, aconselho a seguirem este set de tutoriais, eles instalam também o VSCode que é o interface que usamos para programar o robô, portanto é só seguir estes tutoriais. Além disso nestes tutoriais também vos explicam bem as bases todas que vão precisar de ROS2.

<https://www.youtube.com/playlist?list=PLLSegLrePWgJudpPUof4-nVFHGkB62Izy>

Arm (UFACTORY xArm 6): [xArm Humble Repo](#) To setup the arm in your workspace, please follow the preparation steps on the xArm Humble repository.
(Isto vemos depois, para já podem ir analisar as características do braço)

Terminator Keyboard Shortcuts:

Split Vertically	Ctrl + Shift + O
Split Horizontally	Ctrl + Shift + E
Switch Next Terminal	Ctrl + Shift + N
Switch Previous Terminal	Ctrl + Shift + P
New Tab	Ctrl + Shift + T
Next Tab	Ctrl + Tab
Previous Tab	Ctrl + Shift + Tab
Copy	Ctrl + Shift + C
Paste	Ctrl + Shift + V
Zoom In	Ctrl + +
Zoom Out	Ctrl + -
Reset Zoom	Ctrl + 0
Fullscreen Mode	F11
Close Terminal	Ctrl + Shift + W