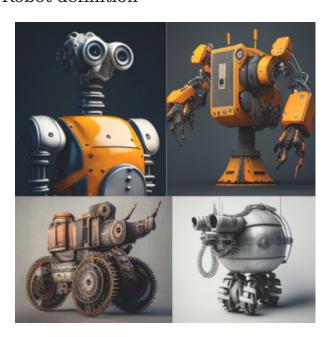
# 01. Introduction

# Robot Operating System (ROS) introduction

### Robot definition



- Joseph Engelberger, pioneer in industrial robotics: "I can't define a robot, but I know one when I see one."
- Wikipedia: "A robot is a machine—especially one programmable by a computer— capable of carrying out a complex series of actions automatically. Robots can be guided by an external control device or the control may be embedded within. Robots may be constructed on the lines of human form, but most robots are machines designed to perform a task with no regard to their aesthetics."
- ISO 8373:2012 Robots and robotic devices Vocabulary, FDIS 2012: "A robot is an actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment, to perform intended tasks."
- Rodney Brooks, Founder and CTO, Rethink Robotics: "A robot is some sort of device, wich has sensors those sensors the world, does some sort of computation, decides on an action, and then does that action based on the

sensory input, which makes some change out in the world, outside its body. Comment: the part "make some change outside its body" discriminates a washing machine from e.g. a Roomba."

• Tamás Haidegger, Encyclopedia of Robotics: "A robot is a complex mechatronic system enabled with electronics, sensors, actuators and software, executing tasks with a certain degree of autonomy. It may be preprogrammed, teleoperated or carrying out computations to make decisions."

### What is ROS?



- Open-source, robotics themed middleware
- Modularity, reusability (drivers, algorithms, libraries, ...)
- Hardware abstraction, ROS API
- C++ és Python support
- Ubuntu Linux (except ROS 2)
- Great community

## History



- Mid 2000s, Stanford: robotics themed, flexible, dynamic framework for prototype development
- 2007, Willow Garage: incubation, the core of ROS under BSD license
- Spread in robotics research, PR2
- 2012: Industrial robotics, ROS-Industrial
- 2017: ROS 2

Development system build -- homework



## Recommended environment:

- Ubuntu 20.04
- ROS1 Noetic
- ROS2 Foxy
- $\bullet \ IDE: QtCreator/CLion/VSCode$

## **ROS 1 Noetic**



#### 1. Install ROS Noetic

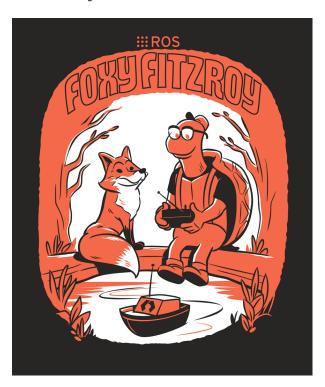
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu \$(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
sudo apt install curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add sudo apt update
sudo apt install ros-noetic-desktop-full
source /opt/ros/noetic/setup.bash

### 2. ROS 1 dependencies

 $sudo\ apt\ install\ python 3-rosinstall\ python 3-rosinstall\ python 3-rosinstall-generator\ python 3-wstool\ build-essential\ sudo\ rosdep\ init\ rosdep\ update$ 

3. Once we are done with this, we can test our ROS 1 installation with the following command:

## **ROS 2 Foxy**



### 1. Setup locale.

locale # check for UTF-8

sudo apt update && sudo apt install locales sudo locale-gen en\_US en\_US.UTF-8 sudo update-locale LC\_ALL=en\_US.UTF-8 LANG=en\_US.UTF-8 export LANG=en\_US.UTF-8

locale # verify settings

### 2. Install ROS 2 Foxy

sudo apt install software-properties-common sudo add-apt-repository universe sudo apt update && sudo apt install curl sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg echo "deb [arch=\$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu \$(./etc/os-release && echo

\$UBUNTU\_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
sudo apt update
sudo apt upgrade
sudo apt install ros-foxy-desktop python3-argcomplete ros-dev-tools ros-foxy-moveit\*
ros-foxy-control\*

3. Once we are done with this, we can test our ROS 2 installation with the following command:

```
source /opt/ros/foxy/setup.bash
ros2 run demo nodes py talker
```

4. The source command is responsible for setting the environment variables, which must be entered every time a new terminal window is opened. This command can be pasted at the end of the ~/.bashrc file, which will run every time a terminal window is opened, so we don't have to type it all the time (ROS 2 will be the default):

```
echo "source /opt/ros/foxy/setup.bash" >> ~/.bashrc
```

## Further packages

1. The following packages will also be needed during the semester, so it is worth installing them as well:

sudo apt install libxml2-dev libraw1394-dev libncurses5-dev qtcreator swig sox espeak cmake-curses-gui cmake-qt-gui git subversion gfortran libcppunit-dev libqt5xmlpatterns5-dev python3-catkin-tools python3-osrf-pycommon libasound2-dev libgl1-mesa-dev xorg-dev ros-foxy-turtlebot3\*

#### IDE

#### 1. QtCreator

QtCreator is currently one of the most usable IDEs for developing ROS packages, for which a ROS plugin has also been created. The installer is available at the link below. You should use the "18.04 **offline** installer", it also works on Ubunutu 20.04.

#### https://ros-qtc-plugin.readthedocs.io/en/latest/\_source/How-to-Install-Users.html

Once downloaded, the IDE can be installed with the following command (it is important to insert cd in the download location):

```
```bash chmod +x qtcreator-ros-bionic-latest-offline-installer.run sudo ./qtcreator-ros-bionic-latest-offline-installer.run

```

When the installer asks where to install it, change to e.g `/home/<USER>/QtCreator` mappára. Ha a root-ba teléepítjük, nem fogjuk tudni futtatni. A telepítés után "Qt Creator (4.9.2)" néven keressük.
```

#### 1. CLion

CLion has a high degree of ROS integration, the use of which is the most recommended during the course. A free student license can be requested at the following link: https://www.jetbrains.com/community/education/#students

After installation, find the <code>/var/lib/snapd/desktop/applications/clion\_clion.desktop</code> file. We rewrite the appropriate line for this, so the environment set by the terminal will be used by the IDE:

```
```bash
Exec=bash -i -c "/snap/bin/clion" %f
```
```

#### 1. Visual Studio

Microsoft Visual Studio also supports source codes for ROS, this IDE can also be used during the semester.



#### Install **Terminator** terminal emulator:

```
sudo apt update
sudo apt install terminator
```

# Useful links

- https://www.ros.org/
- ROS 1 Noetic installation
- ROS 2 Foxy installation
- ROS Distributions
- http://wiki.ros.org/ROS/Tutorials
- CLion hallgatói licensz
- QtCreator + ROS plugin
- IROB virtual tour