



# 01. Introduction

## Robot Operating System (ROS) introduction

### The definition of robot

- **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 pre-programmed, teleoperated or carrying out computations to make decisions."*

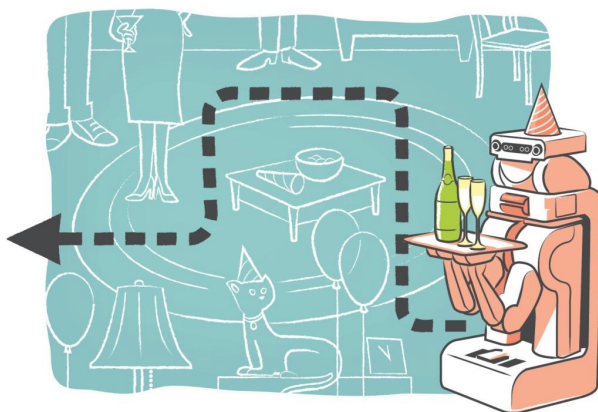
What is ROS?

# 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 reserach, PR2

- 2012: Industrial robotics, ROS-Industrial
- 2017: ROS 2

## Development system build -- homework

Recommended environment:

- Ubuntu 22.04
- ROS2 Humble
- *IDE: QtCreator*

### ROS 2 Humble Hawksbill



Setup locale.

```
```bash
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
```

---
```

#### 1. ROS 2 Humble install

```

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-humble-desktop
sudo apt install ros-dev-tools

```

## 2. Test the new ROS 2 install:

```

source /opt/ros/humble/setup.bash
ros2 run demo_nodes_py talker

```

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

```

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

```

## Further packages

1. We will also need the following packages during the semester, so these should be installed 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-osrf-pycommon libasound2-dev libgl1-mesa-dev xorg-
dev python3-vcstool python3-colcon-common-extensions python3-pykd1 python3-pyudev
libxml2-dev libraw1394-dev libncurses5-dev qtcreator swig sox espeak cmake-curses-
gui cmake-qt-gui git subversion gfortran libcppunit-dev libqt5xmlpatterns5-dev
libbluetooth-dev ros-humble-joint-state-publisher* ros-humble-xacro gfortran-9

```

# IDE

## 1. QtCreator

Currently, one of the most widely used IDEs for developing ROS packages is QtCreator, for which a ROS plugin has been developed. The installer is available at the link below. You should use the "18.04 **offline** installer", it also works on Ubuntu 22.04.

[https://ros-qtc-plugin.readthedocs.io/en/latest/\\_source/How-to-Install-Users.html](https://ros-qtc-plugin.readthedocs.io/en/latest/_source/How-to-Install-Users.html)

Once downloaded, the IDE can be installed using the command below (it is important to put `cd` in the download location):

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

When the installer asks you where to install it, change it to e.g. `/home/<USER>/QtCreator`. If you put it in root, you will not be able to run it. After installation, look for `Qt Creator (4.9.2)'.

## 2. CLion

CLion has a high level of ROS integration, and its use is most recommended for this course. A free student license can be obtained at <https://www.jetbrains.com/community/education/#students>

After installation, browse to the file `/var/lib/snapd/desktop/applications/clion_clion.desktop`. Copy the appropriate line here so that the IDE will use the environment set by the terminal:

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

## 3. Visual Studio

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



### Suggestion

Install **Terminator** terminal emulator:

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

## Links

- <https://www.ros.org/>
- [ROS 2 Humble installation](#)
- [QtCreator + ROS plugin](#)