

01. Bevezetés

Robot Operating System (ROS) bevezetés

A robot fogalma



- **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, which 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."

Mi a ROS?



I

- Open-source, robotikai célú middleware
- Modularitás, újra-felhasználhatóság (driverrek, algoritmusok, library-k, ...)
- Hardware absztrakció, ROS API
- C++ és Python támogatás
- Ubuntu Linux (kivéve ROS 2)
- Népes közösség

Történet



- 2000-es évek közepe, Stanford: robotikai célú rugalmas, dinamikus szoftverrendszer prototípusok fejlesztése
- 2007, Willow Garage: inkubáció, kialakult a ROS alapja BSD open-source licenz alatt
- Robotikai kutatások területén egyre inkább elterjedt, PR2
- 2012: Ipari robotika, ROS-Industrial
- 2017: ROS 2

Fejlesztőkörnyezet felállítása - Házi feladat



Ajánlott környezet:

- Ubuntu 22.04
- ROS2 Humble
- *IDE: QtCreator/CLion/VSCode*

Suggestion

Ha ragaszkodunk a Windows-hoz: WSL (Windows Subsystem for Linux): <https://docs.ros.org/en/humble/Tutorials/Advanced/Simulators/Webots/Installation-Windows.html>

ROS 2 Humble Hawksbill



1. Locale beállítása.

```
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. ROS 2 Humble telepítése

```
sudo apt install software-properties-common
sudo add-apt-repository universe
sudo apt update && sudo apt install curl -y
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
```

3. Ha ezzel megvagyunk, a következő paranccsal tesztelhetjük a ROS 2 telepítésünket:

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

4. A `source` parancs a környezeti változók beállításáért felelős, ezt minden új terminálablak megnyitásakor meg kell(ene) adni. Ez a parancs beilleszthető a `~/.bashrc` fájl végére, amely minden terminálablak megnyitásakor lefut, így nem kell mindig beírunk (ROS 2 lesz az alapértelmezett):

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

További csomagok

1. Az alábbi csomagokra is szükség lesz a félév során, így ezeket is érdemes feltelepíteni:

```
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 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

ROS csomagok fejlesztéséhez jelenleg az egyik leginkább használható IDE a QtCreator, melyhez ROS plugin is készült. Az installer az alábbi linken elérhető. A "18.04 **offline** installer"-t érdemes használni, ez működik Ubuntu 22.04-en is.

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

Ha letöltöttük, az IDE az alábbi paranccsal telepíthető (fontos, hogy `cd`zzünk be a letöltés helyére):

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

Amikor a telepítő kérdezi, hova telepítse, módosítsuk pl. `/home/<USER>/QtCreator` mappára. Ha a root-ba teleépítjük, nem fogjuk tudni futtatni. A telepítés után "Qt Creator (4.9.2)" néven keressük.

2. CLion

A CLion magasfokú ROS integrációval rendelkezik, a kurzus során ennek a használata a leginkább ajánlott. Ingyenes hallgatói licenz az alábbi linken igényelhető: <https://www.jetbrains.com/community/education/#students>

Telepítés után keressük meg a `/var/lib/snapd/desktop/applications/clion_clion.desktop` fájlt. A megfelelő sort írjuk át erre, így a terminál által beállított környezetet fogja használni az IDE:

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

3. Visual Studio

A Microsoft Visual Studio szintén támogatja a ROS-hoz készült forráskódokat, ez az IDE is használható a félév során.

Suggestion

Install **Terminator** terminal emulator:

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

Hasznos linkek

- <https://www.ros.org/>
- [ROS 2 Humble installation](#)
- [ROS Distributions](#)
- <http://wiki.ros.org/ROS/Tutorials>
- [CLion hallgatói licenz](#)
- [QtCreator + ROS plugin](#)
- [ROS 2 Humble installation on WSL](#)