

Mobil gépek mechatronikája

Dr. Aradi Szilárd
Közlekedés- és Járműirányítási
Tanszék
Bevezetés

Időbeosztás, követelmények

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Dr. Aradi Szilárd, Doba Dániel
- Előadás: hétfő 12:15, St. 321A., kéthetente
- Gyak1: hétfő 12:15, St. 101., kéthetente
- Gyak1: hétfő 14:15, St. 101., kéthetente
- Lab1: kedd 10:15, St. 101., hetente
- Lab2: kedd 14:15, St. 101., hetente
- 1 ZH
- Féléves házi feladat

Tematika

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Mobil robotok alapjai
- Linux alapok
- Robot Operating System
- Objektumorientált programozás
- Python programozás
- Lokalizáció
- Térképezés
- Útvonaltervezés- és követés
- Gyakorlati példák

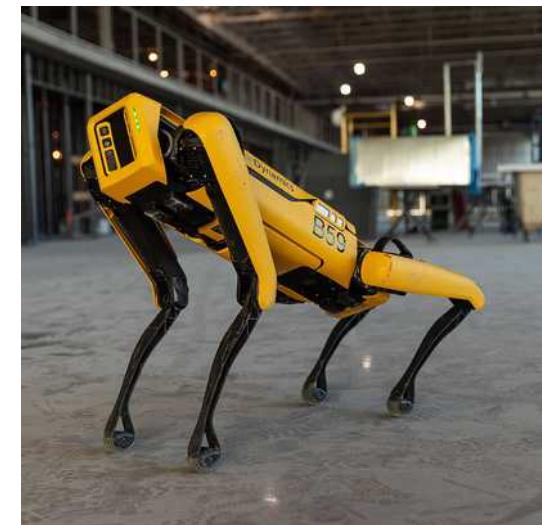
Mobil robotok

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Automatikus gépek, amelyek részben vagy teljesen önállóan képesek a helyváltoztatásra a megfelelő környezetükben.
- Széles körben elterjedtek a legkülönbözőbb ipari, szolgáltatási és háztartási területeken.
 - Beltéri robotok (pl.: kórházak, raktárak, egyéb ipari létesítmények, otthoni takarító robotok)
 - Áruszállítás
 - Speciális funkciók: takarítás, fertőtlenítés, őrzés stb.
 - Kültéri robotok (pl: áruszállítás, katonai és védelmi alkalmazás, katastrófavédelem)
 - Áruszállítás részben távirányítással
 - Katonai felderítés és szállítás
 - Keresési feladatok veszélyes környezetben



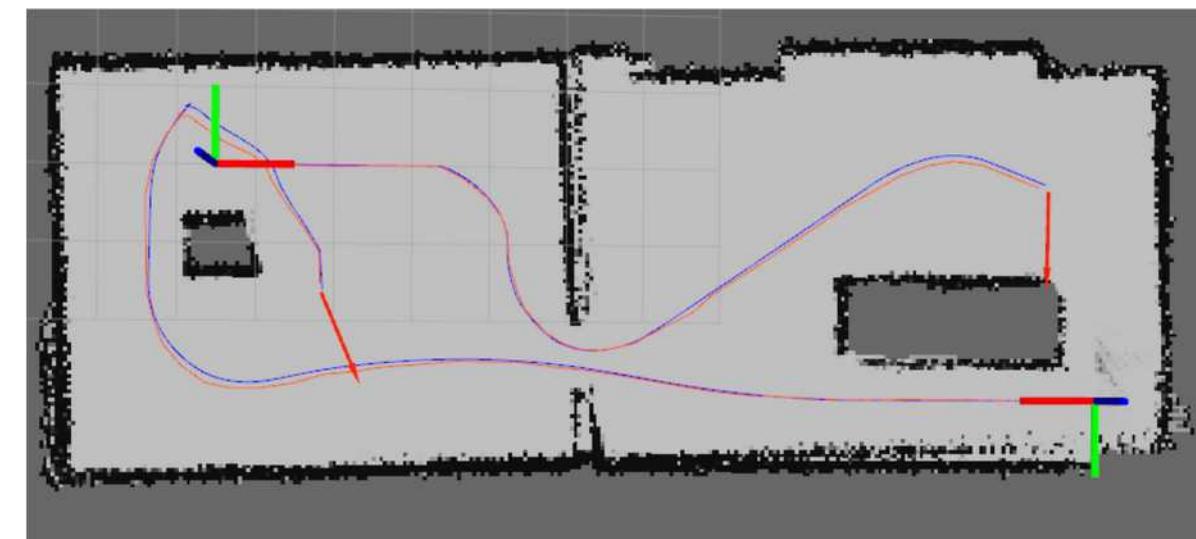
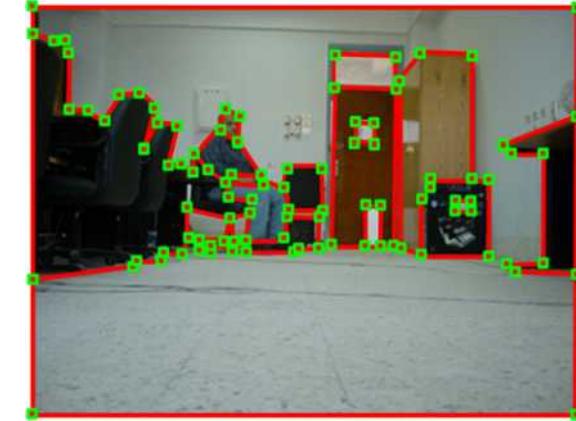
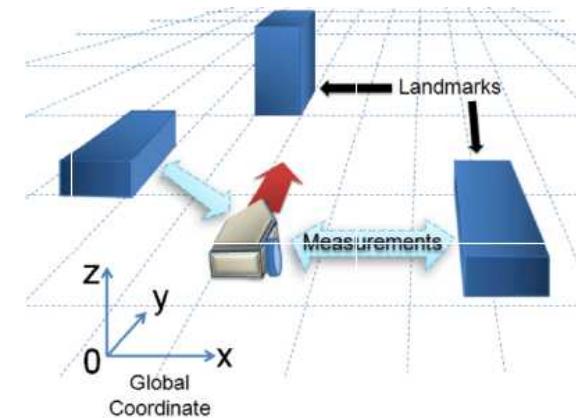
Funkcionális követelmények

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Hajtásszabályozás
 - Sebesség, szögsebesség
 - Lábak szabályozása, egyensúlyozás
- Lokalizáció (térképezés)
 - Pozíció meghatározása egy adott koordináta rendszerben (térfépen)
- Környezetérzékelés
 - Térképezés
 - Ütközéselkerülés
 - Objektumdetekció
- Útvonaltervezés
 - Globális és lokális
- Útvonalkövetés



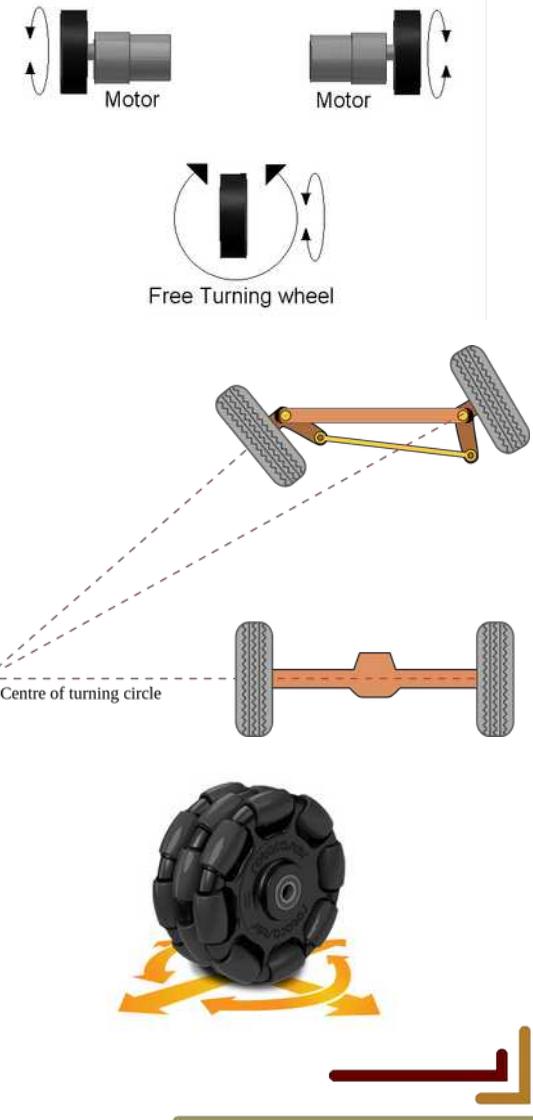
Komponensek

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Vezérlő számítógép
 - Beágyazott rendszer vagy PC-szerű vezérlőegység
 - Szoftver keretrendszer: op. rendszer, software stack vagy middleware
- Hajtás
 - Kerekek, lábak, lánctalp
 - Holonom vagy anholonom kinematika
 - Differenciál hajtás
 - Ackermann kormányzás
 - Omni wheel
- Szenzorok
 - Közelség- és ütközés: infravörös, kapacitív, ultrahang, nyomógomb
 - Távolság: ultrahang, infravörös, lidar
 - Sebesség, szögsebesség: kerék enkóder
 - Pozíció: GNSS, szenzor alapú lokalizáció, beltéri pozícionáló rendszer
 - Képalkotás: kamera, infrakamera, lidar
- Kiegészítő aktuátorok
- Tápellátás



Robot Operating System - ROS

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Mi az a ROS?
- ROS története
- ROS koncepció
- Támogatott program nyelvek
- Fájl rendszer és kód fordítás
- ROS elemek
 - Node
 - Topic
 - Launch fájlok
- Rosbag
- Időbélyegek, időzítések
- Vizualizáció
- Multimachine, multimaster - Hálózati kapcsolatok
- Transformáció - tf2
- Robotok leírása URDF formában, xacro
- Gazebo
- Dinamikus konfigurálás
- ROS robotok



Mi az a ROS?

Budapesti Műszaki és Gazdaságtudományi Egyetem

► Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- A ROS nem egy valódi operációs rendszer.
- A ROS egy ún. „middleware” szoftverkörnyezet, amit elsődlegesen robot fejlesztések felgyorsítására hoztak létre. A ROS biztosítja az alapvetően szükséges elemeket moduláris formában.
- A ROS fő erőssége a nyílt forráskód, és kifejezetten bátorít a kód újrahasználhatóságra.
- A ROS-t elsődlegesen kutatás-fejlesztési környezetben hasznosítják, de egyre több az ipari alkalmazás.



ROS koncepció

Budapesti Műszaki és Gazdaságtudományi Egyetem

► Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék



1. Csövezetékelés - Plumbing: az üzenetek küldés-fogadására egy rugalmas, gyorsan bővíthető, átalakítható rendszer lett kialakítva.
2. Eszközök - Tools: eszközök széles skálája áll rendelkezésre az adott rendszerek konfigurálására, indítására és megállítására, monitorozására, megjelenítésére, tesztelésére, mérési adatok tárolására.
3. Képességek - Capabilities: nagy mennyiségű könyvtár tartozik hozzá, melyek fókusza a különböző robot műveletek - mozgás, érzékelés, beavatkozás - megvalósítása.
4. Ökoszisztemá - Ecosystem: Nagy a közösségi támogatottsága, jelentős hangsúlyt helyezve az integrációra és a dokumentációra. Egyik példa erre a ros.org, ahol több ezer ROS könyvtár dokumentációja elérhető, valamint egy megbízható, sokak által használt fórum.

ROS történelem

Budapesti Műszaki és Gazdaságtudományi Egyetem

► Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- A ROS alapjait Standford-i Egyetem Phd hallgatói fektették le.
- 2007-ben az amerikai Willow Garage átvette fejlesztést és teljes értékű framework-é nőtte ki magát.
- 2010-ben került kiadásra az első hivatalos verzió, majd félévente új főverziók követték.
- 2013-ban Open Source Robotics Foundation (OSRF) átvette a karbantartási és fejlesztési feladatokat, és ekkor állnak át évenkénti frissítési ciklusra. minden második évben kiadott verziók az Ubuntu Linux rendszerekhez hasonlóan kibővített támogatási ciklussal rendelkeznek, mely 5 évet jelent.
- 2017-től az OSRF Open Robotics néven fut tovább.
- 2017-ben kiadták a ROS 2-t, aminek az egyik legfontosabb újítása, hogy támogat valós idejű megoldásokat.
- 2020-ban kiadták az utolsó ROS 1 verziót, a Noetic Ninjemys-t, amely 2025-ig támogatott.
- A továbbiakban a ROS 2-ből lesznek új verziók.

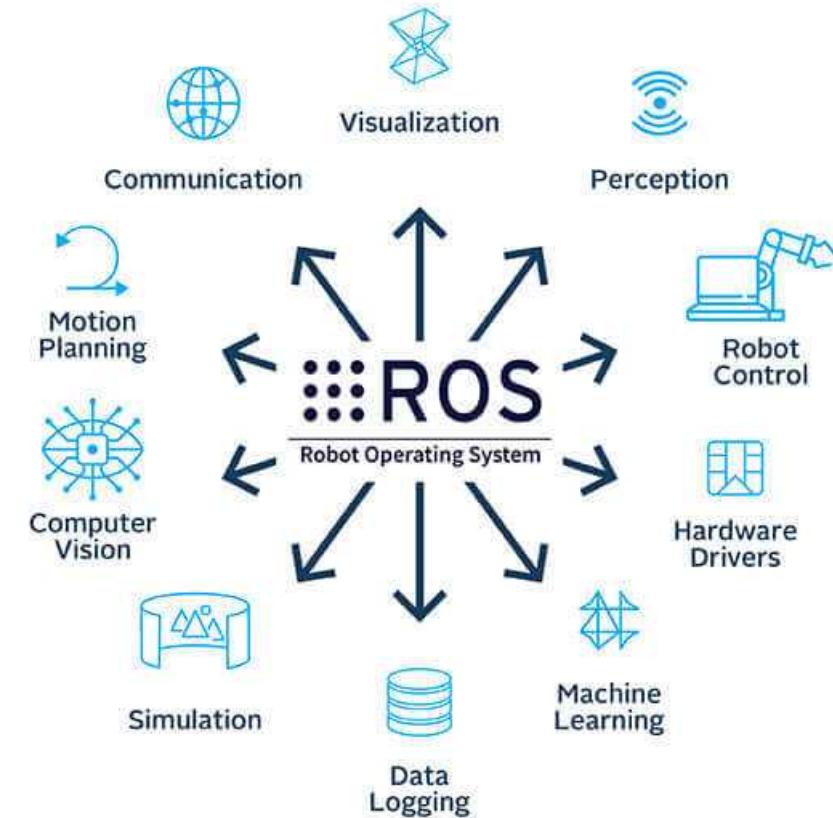
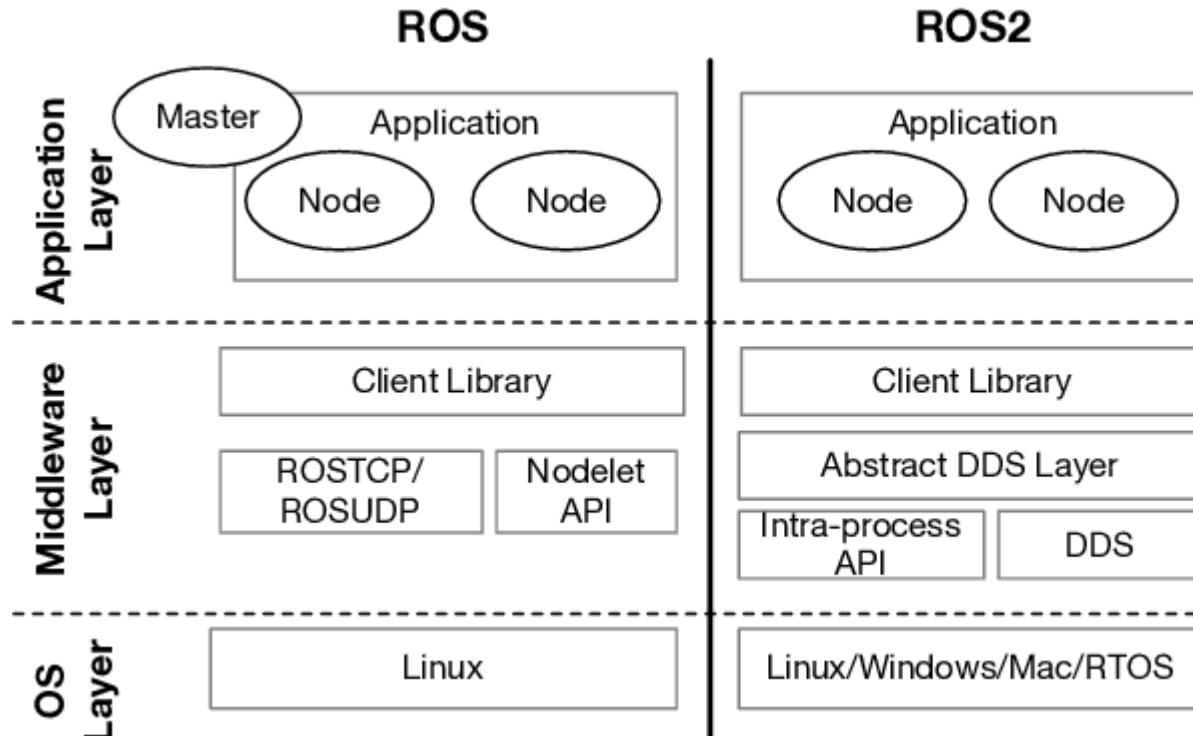


ROS felépítés I.

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

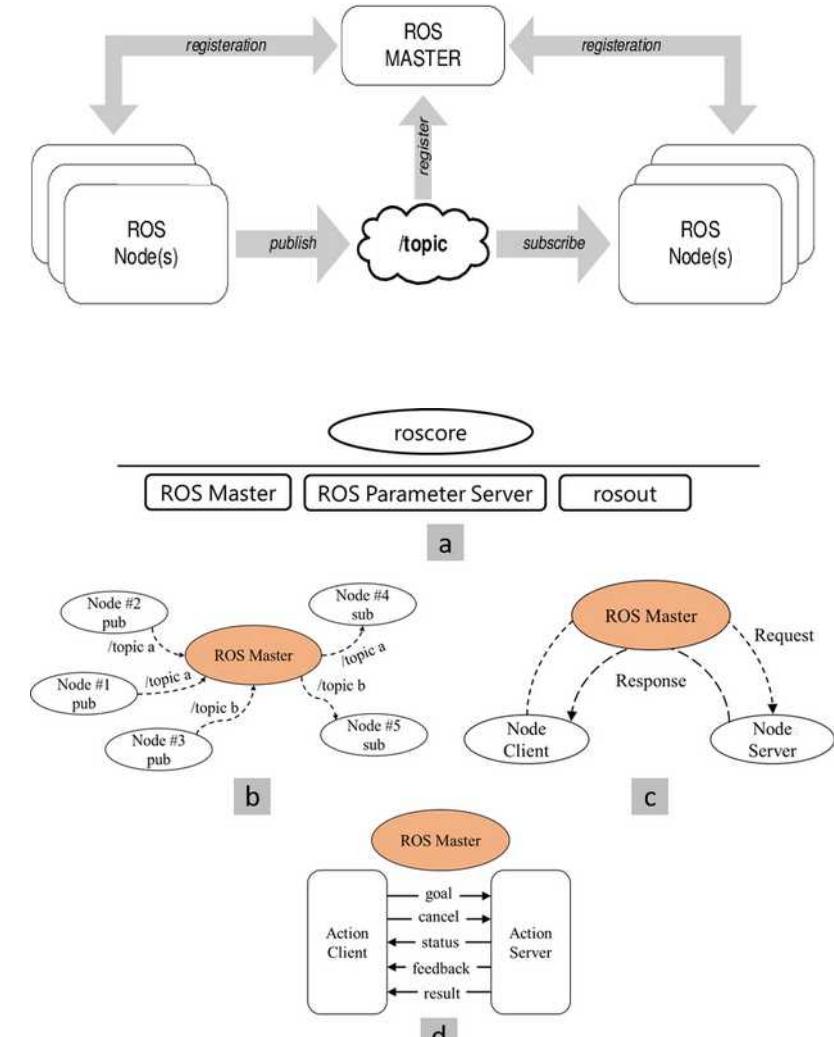


ROS felépítés II.

- ROS Master, felel a node-k és a topic-ok regisztrálásáért, a paraméter szerver kezeléséért, valamint az alapvető funkciók kezeléséért. (a. ábra)
- A paraméter szerver egy olyan szolgáltatása a ROS Masternek, mely egy helyen mindenki számára látható módon tárol adatokat. Ezek általában statikus vagy fél-statikus információk, pl.: kalibrációs vagy állapot jelző információk. (a. ábra)
- **A ROS 2-ben NINCS master és paraméter szerver. A feladatokat decentralizáltan oldják meg a Data Distribution Service-szel (DDS) és Distributed Parameter API-val..**
- A rosout egy rendszer szintű logger. (a. ábra)
- A node-k egyedi folyamatok, melyek a rendszert alkotják. Egy node általában egy jól körül határolt feladatért felel ezzel is segítve egy moduláris rendszer kialakítását. (b. ábra)
- A topic-ok a kommunikációs csatornák a node-k között, melyek adott üzenet leírással/definícióval rendelkeznek. Az üzenet küldő regisztrálja a topic-ot a ROS Master-nél és az esemény vagy idő vezérlésnek megfelelően küldi az üzeneteket. A fogadó fél feliratkozik az adott topic-ra és képesé válik az üzenetek fogadására. (Felső ábra és b. ábra)

A gyakorlatban az üzent továbbítás peer-to-peer kommunikációt jelent, a ROS Master csak az ip címek és az adott folyamat által nyitott port címeket tárolja, és adja át a küldő oldalnak. Az adat továbbítás aszinkron TCP kommunikációnak felel meg.

Lehetőség van több küldő és fogadó node definiálására is.



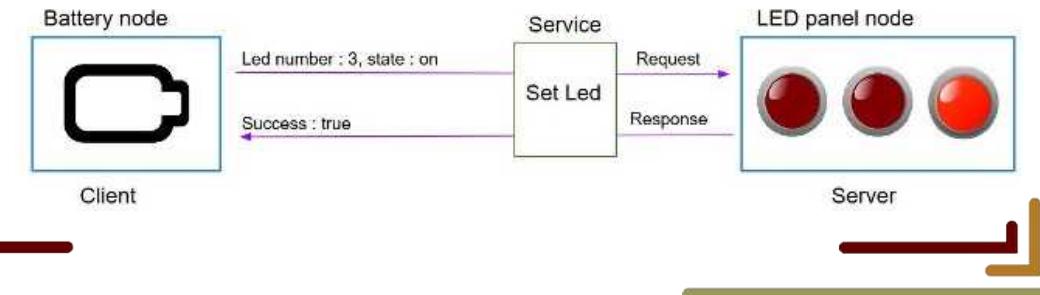
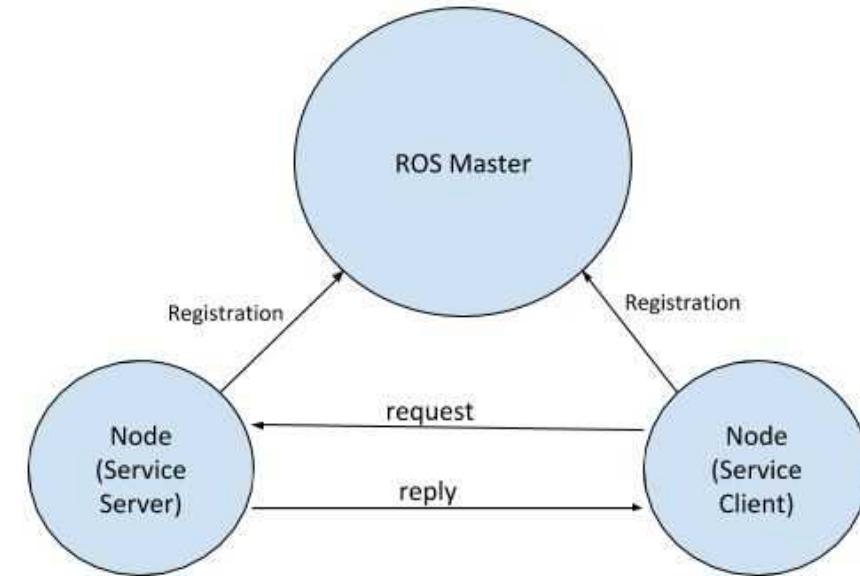
ROS felépítés III.

Budapesti Műszaki és Gazdaságtudományi Egyetem

► Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- A ROS service-ek egy „blocking” kliens-szerver rendszert alkotnak.
- Ezzel lehetőség van az adott kérés és a válaszként jött üzenet közötti kapcsolat definiálására.
- Egy service servernek több kliense is lehet.
- Hátrányuk, hogy szinkronizáltak. A kliens elküldi a kérést, ami blokkolja a folyamatát a válasz beérkezéséig. Azaz, ha a válasz üzenetre sokat kell várni, akkor a kliens folyamata áll.



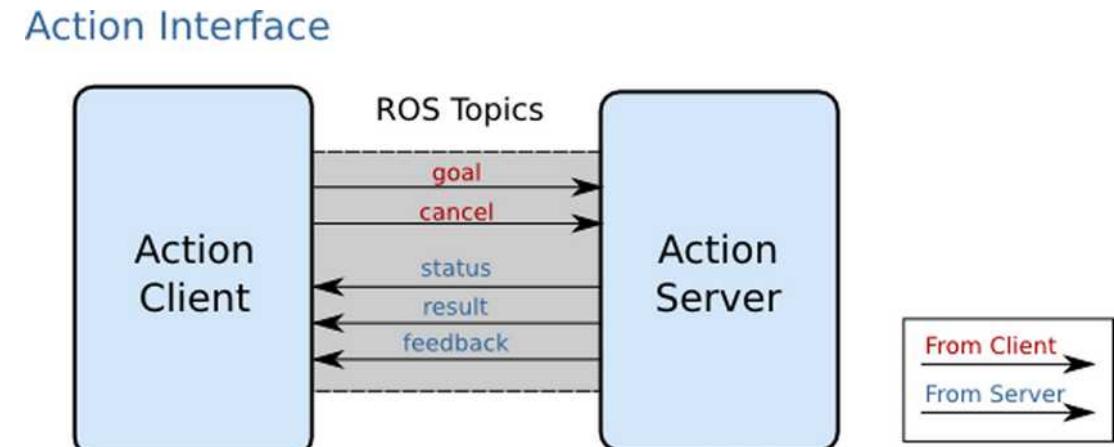
ROS felépítés IV.

Budapesti Műszaki és Gazdaságtudományi Egyetem

► Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- A ROS actionök alkotnak.
- A kliens küld egy kérést (goal) a szervernek.
- A szerver válaszol rá (feedback).
- Ha szerver végzett, akkor küldi az eredményt (result).
- Menetközben a kliens.
- További lehetőség a kérés visszavonás a kliens részéről (cancel) és a státusz küldése a szerver részéről (status).



Támogatott nyelvek

Budapesti Műszaki és Gazdaságtudományi Egyetem

► Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Támogatott nyelvek :

- C++ 14
- Python 3.8
- Matlab/Simulink
- Common LISP, Java, Ruby, R, Go, stb.

- Ezek közül a **pirossal** jelöltek, amiket a fejlesztők közvetlenül támogatnak.
- Legyünk körültekintőek, mert a c++ és Python implementációk között lehet különbség.

Például:

- on_shutdown() - képesek vagyunk felül írni a program leállításért felelős függvényét Python-ban, c++ ilyen függvény nem létezik, helyette az alap signal kezelő függvényt lehet használni.
- rospy.is_shutdown(), ros::ok(), ros::isShuttingDown()
- NodeHandle() - olyan objektumok, melyek kezelik a program ROS-sal való kapcsolatát. Ezek csak c++-ban léteznek, pythonban nem.

```
#!/usr/bin/env python3

import rospy
from std_msgs.msg import String

def callback(msg):
    pass

if __name__ == '__main__':
    rospy.init_node('get_mes')
    sub_mes = rospy.Subscriber('/topic_name', String, callback)
    r = rospy.Rate(10)
    while not rospy.is_shutdown():
        r.sleep()
```

```
#include "ros/ros.h"
#include "std_msgs/String.h"

void callback(const std_msgs::String::ConstPtr& msg){ }

int main(int argc, char **argv){
    ros::init(argc, argv, "get_mes");
    ros::NodeHandle n;
    ros::Subscriber sub = n.subscribe("/topic_name",
        1000, chatterCallback);
    ros::Rate r(10);
    while(ros::ok()){
        r.sleep();
    }
    return 0;
}
```

ROS robotok

Budapesti Műszaki és Gazdaságtudományi Egyetem

Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Open Robotics
 - TurtleBot 3
- Husarion
 - Rosbot
- Clearpath Robotics
 - Turtlebot 4
 - Jackal, Husky, Warthog
- CogniTeam
 - Hamster
- F1tenth

TurtleBot 3 Family

Burger



Waffle



Waffle Pi



Összefoglalás

Budapesti Műszaki és Gazdaságtudományi Egyetem

► Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

- Mobil robotok
 - Differenciál hajtású robotok kinematikája és irányítása
 - Alapvető robotikai funkciók és algoritmusok
 - Lokalizáció
 - Térképezés
 - Útvonaltervezés és -követés
- Robot Operating System (ROS) alkalmazása
 - Linux
 - C++
 - Objektumorientált programozás
 - ROS működése és alkalmazása

Vége

Budapesti Műszaki és Gazdaságtudományi Egyetem

► Közlekedésmérnöki és Járműmérnöki Kar

Közlekedés- és Járműirányítási Tanszék

Köszönöm a figyelmet!