

# Robotics & XR

5 ECTS

Ilkka Jormanainen and Samuel Yigzaw  
{firstname.lastname}@uef.fi

Robotics applications

November 4, 2024

# Course contents

- Theme 1: The role of robotics in society and an introduction to ethical issues in robotics
- **Theme 2: Different types of robots and their application to real-world problems**
- Theme 3: Robot control theory
- Theme 4: Navigation
- Theme 5: Robotics & AI
- Theme 6: Extended Reality applications in robotics

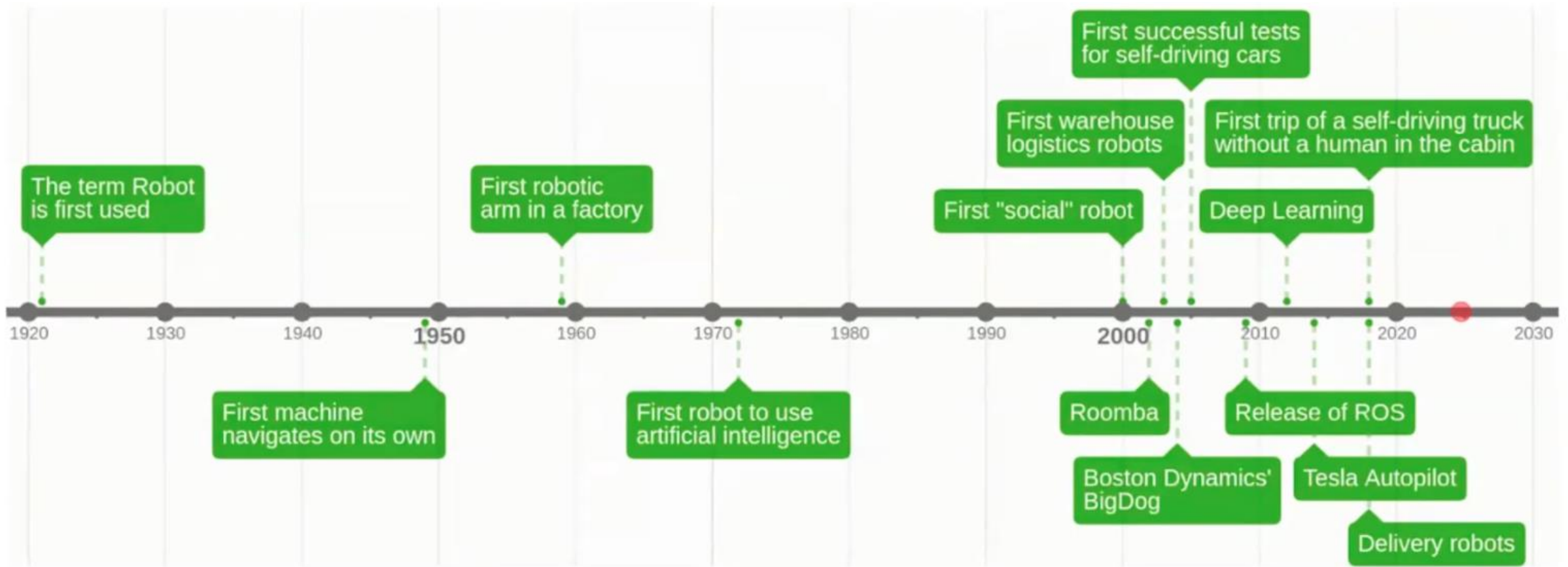
# Theme 2: Robotics applications

- *Week 45 (November 4 – 10)*
- Brief history of robotics
- Different type of robots and their applications in different fields
- Robotics frameworks and toolkits
- ROS2 application framework: Why and how?

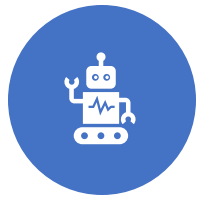
The logo for ROS 2, featuring a 3x3 grid of dots to the left of the text "ROS 2".

ROS 2

# A brief timeline of robotics applications



# Categorization of robots (one example)



Mobile robots



Collaborative robots  
(cobots)



Industrial robots



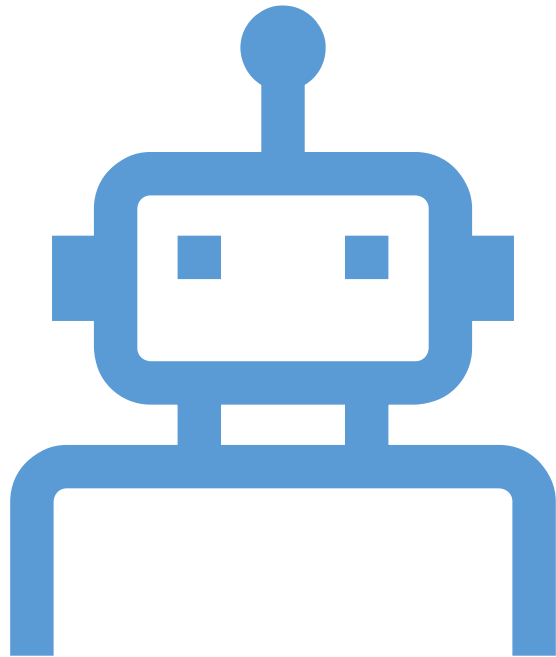
UAVs (Unmanned  
Aerial Vehicles)



Human or animal  
shaped robots



AUVs (Autonomous  
underwater vehicles)



# Categorization by the use

- Autonomous cars
- Domestic robots
- Social robots
- Service robots
- Medical robots
- Drones (aerial or underwater)
- Agriculture robots
- And so on!

# Definition of robotics systems



Robots shall consist of mechanical, electrical and electronic components



Actuators and sensors



Be physical agents affecting their environment



Capability to perform an action



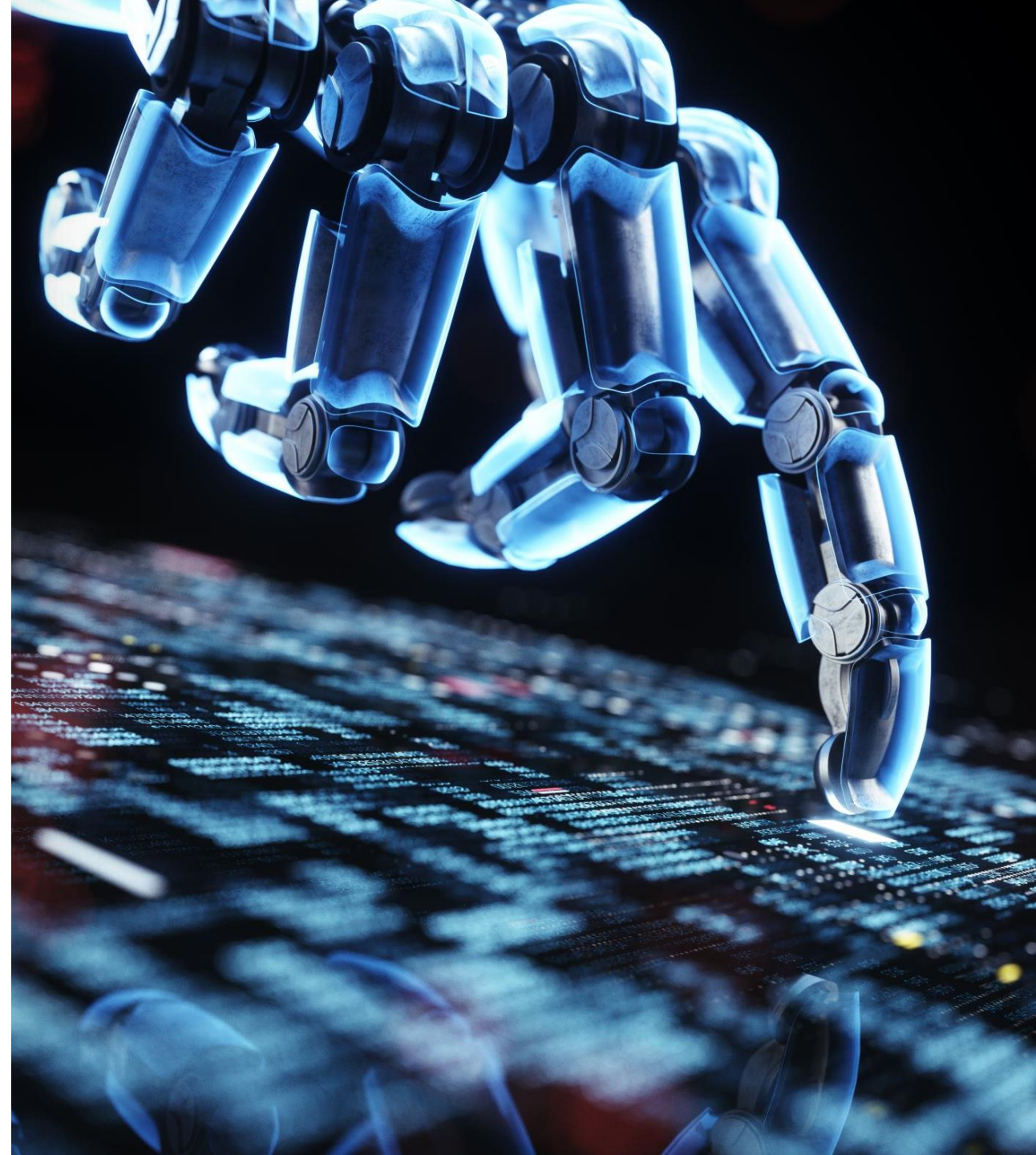
Have a well-defined task, e.g., to perform locomotion, manipulation, positioning, reconnaissance or delivery

Tamas Haidegger, "*Taxonomy and Standards in Robotics*". In Marcelo H. Ang, Oussama Khatib, and Bruno Siciliano (eds), *Encyclopedia of Robotics*, Springer Nature, 2021



# Robot definitions by formal standards

- ISO 8373 - Robots and robotic devices:  
*"programmed actuated mechanism with a degree of autonomy to perform locomotion, manipulation or positioning"*
- IEEE 1872 Standard Ontologies for Robotics and Automation: *"An agentive device in a broad sense, purposed to act in the physical world in order to accomplish one or more tasks. [...] A robot is composed of suitable mechanical and electronic parts. Robots might form social groups, where they interact to achieve a common goal. A robot (or a group of robots) can form robotic systems together with special environments geared to facilitate their work."*





# More definitions from ISO 1873

- Industrial robot: *“Automatically controlled, reprogrammable multipurpose manipulator(s), programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications”*
- Service robot: *“Robot that performs useful tasks for humans or equipment excluding industrial automation applications”*
- Medical robot: *“A robot with medical intended use”*



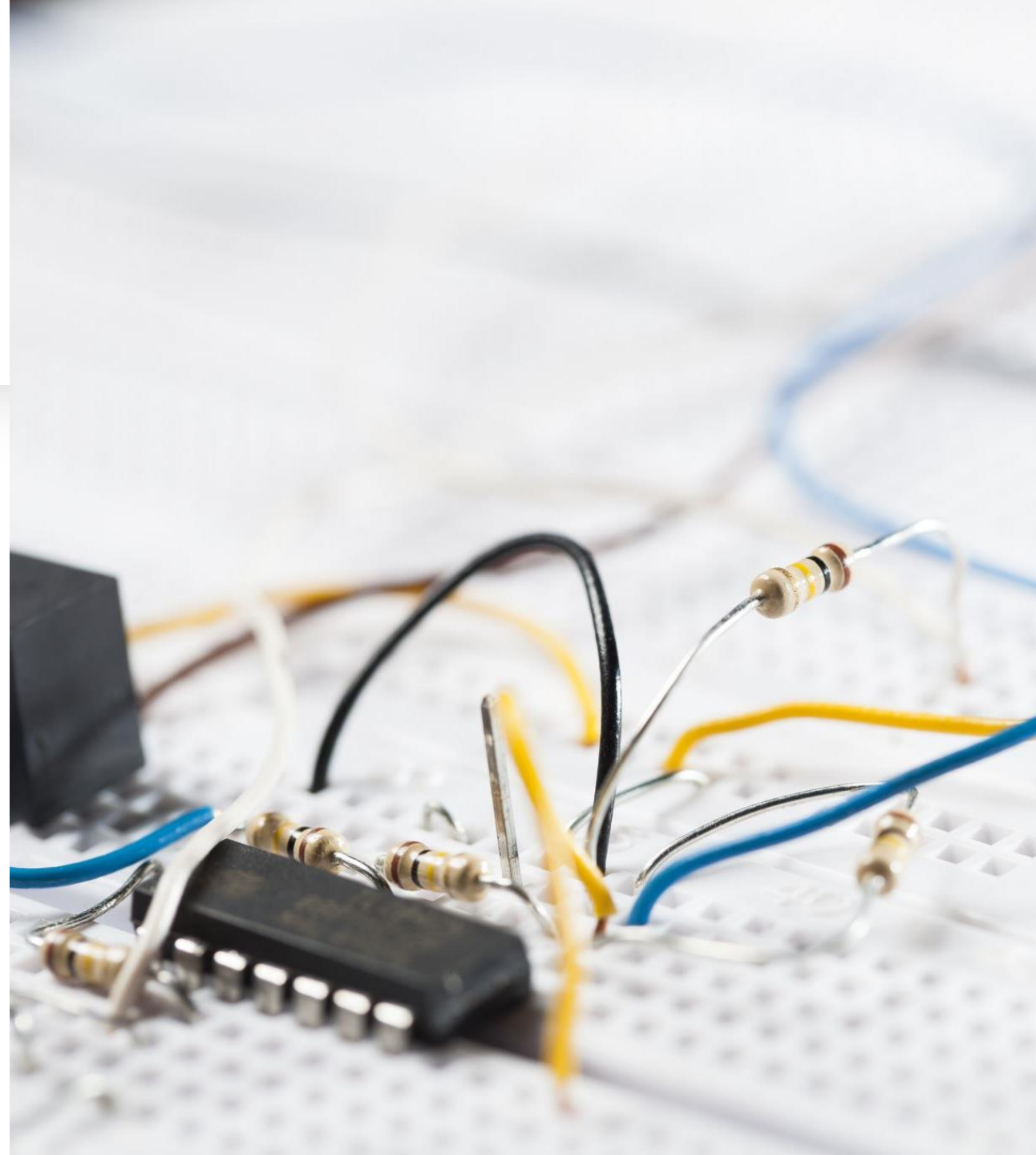
# What features different robots should have?



<https://padlet.com/ilkkajormanainen1/features-of-different-robot-applications-ihjumnrlmefgm5pr>

# Which components a robot system is built on?

- Sensors
- Actuators
- Mechanical structure
- Power source
- Communication capabilities
- Computational devices



# Sensors

- Sensors allow a robot to observe its surrounding
- Sensors "replace" human or animal senses and provide a robot system with capabilities to react external stimulus
- Sensors are everywhere – mobile phones, computers, cars, access control system, elevators, traffic lights, home appliances, ...
- Stimulus -> processing -> reaction
- Most typical sensors in robotics applications
  - Touch, distance, light, color, direction, IR, acceleration, tilting, sound, temperature, rotation, gyroscope
  - Laser-based range finders (LIDAR), stereo cameras, GPS

# Actuators

- Actuators convert energy (electricity) into physical motion
- Motors, linear actuators, solenoids, muscle wires, pneumatic and hydraulic
- Most robotics solutions have one or several type of motors
- Three main types of motors
  - DC motors, gear motors
  - Servo motors
  - Stepper motors





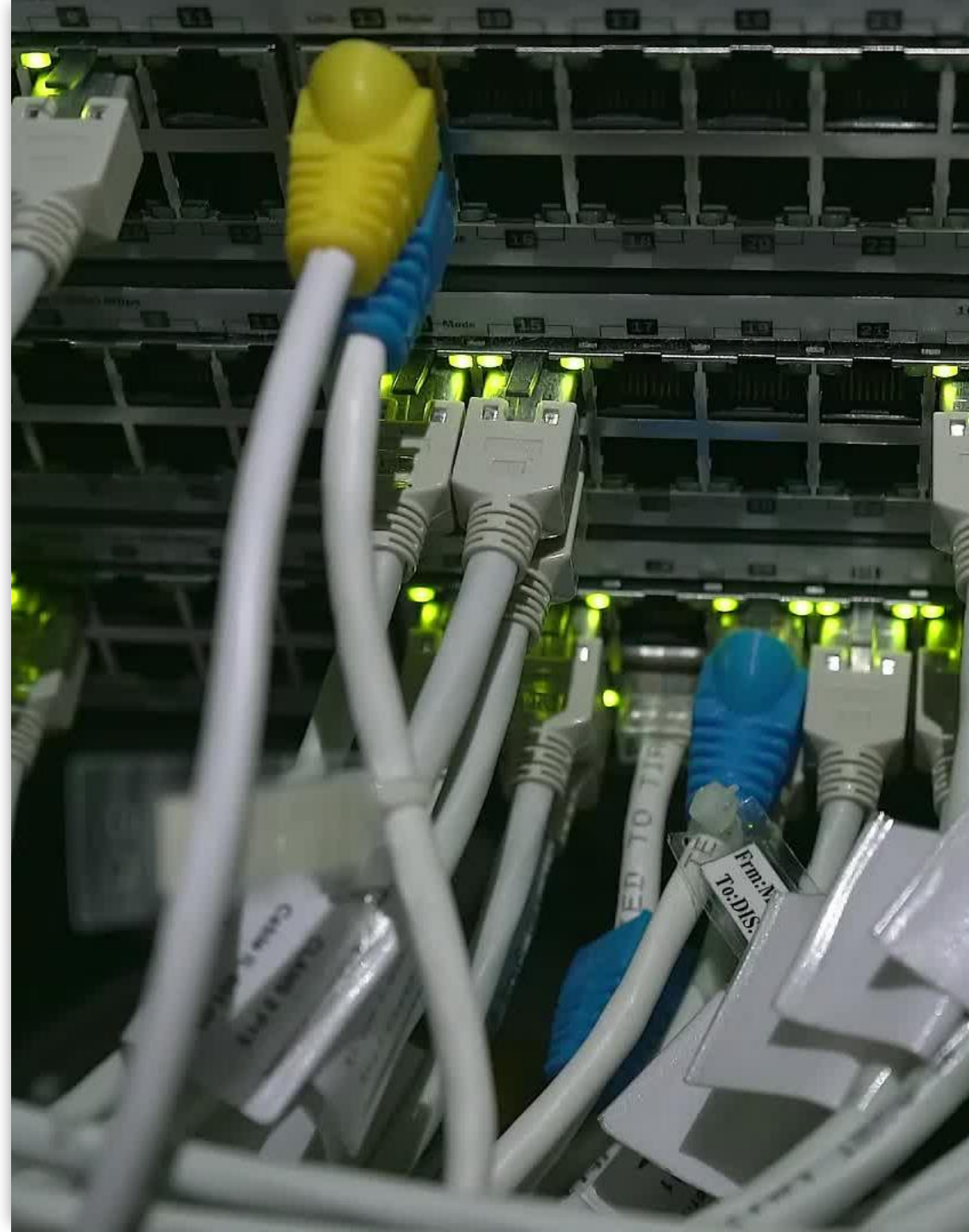
# Mechanical structure and power source

- Motors allow mobile robots to move
  - Rotating wheels or belts
  - Moving joints of robot's legs
  - Spinning rotators of drones
- Robot chassis to hold and protect components
- Autonomous robots need power (in most cases electricity)
- Several different type of power supplies depending on the use case



# Communication capabilities

- Short-range communication
  - Bluetooth, NFC, Zigbee, MQTT
- Long-range communication
  - TCP/IP based networks (WLAN, Ethernet), UDP, 4/5/6G, MODBUS
- Internal communication
  - Ethernet, DeviceNet (CAN), USB, serial port communication



# Computing devices

- STM32F103xx
- ATmega328
  - Used widely in Arduinos
- PIC16xxxx
  - Probably most popular 8-bit microcontroller
- Attiny85
  - Small form-factor, ideal for small
- TI MSP430 series
  - Especially good for low-power applications
- ESP8266/ESP32
  - Built-in communication capabilities
  - Variety of built-in interfaces
- Microprocessor – just a chip with transistors for computing operations
- Microcontroller – Single IC (integrated circuit) with processor, memory, programmable general purpose I/Os (GPIO)
- SoC (System on Chip) – Integrates functionalities for one chip (CPU, memory, GPU, signal processing)
- SBC (Single Board Computer) – “real” computers squeezed in small form factor (for example, Raspberry Pi, Nvidia Jetson series)
- Full PC:s in miniature size – Intel NUC, Apple Mac mini

# How to develop a complex robotics system?

- Building and testing a comprehensive robotics system with multiple sensors, actuators, communication devices and protocols may be an exhaustive task
- Modularization, abstraction, interfacing
- Multi-robotics systems (synchronization, task management)
- Use of established frameworks help!

# How to develop a complex robotics system?

- Building and testing a comprehensive robotics system with multiple sensors, actuators, communication devices and protocols may be an exhaustive task
- Modularization, abstraction, interfacing
- Multi-robotics systems (synchronization, task management)
- Use of established frameworks help!

 ROS 2



# Example: Making a robot to move

```
digitalWrite(6, HIGH);  
digitalWrite(7, LOW);  
analogWrite(10, 128);
```

Arduino with  
L289N H-Bridge

# Example: Making a robot to move

```
Adafruit_DCMotor *leftMotor = AFMS.getMotor(1);  
Adafruit_DCMotor *rightMotor = AFMS.getMotor(2);  
leftMotor->setSpeed(128);  
rightMotor->setSpeed(128);
```

```
digitalWrite(6, HIGH);  
digitalWrite(7, LOW);  
analogWrite(10, 128);
```

Arduino with I2C  
motor driver

Arduino with  
L289N H-Bridge

# Example: Making a robot to move

```
ros2 topic pub -r 1 /cmd_vel
geometry_msgs/msg/Twist "{linear: {x: 0.5, y: 0.0,
z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
```

ROS2 topic for  
setting velocity  
for a robot

```
Adafruit_DCMotor *leftMotor = AFMS.getMotor(1);
Adafruit_DCMotor *rightMotor = AFMS.getMotor(2);
leftMotor->setSpeed(128);
rightMotor->setSpeed(128);
```

Arduino with I2C  
motor driver

```
digitalWrite(6, HIGH);
digitalWrite(7, LOW);
analogWrite(10, 128);
```

Arduino with  
L289N H-Bridge

# Example: Making a robot to move

ABSTRACTION

```
ros2 topic pub -r 1 /cmd_vel
geometry_msgs/msg/Twist "{linear: {x: 0.5, y: 0.0,
z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
```

ROS2 topic for  
setting velocity  
for a robot

```
Adafruit_DCMotor *leftMotor = AFMS.getMotor(1);
Adafruit_DCMotor *rightMotor = AFMS.getMotor(2);
leftMotor->setSpeed(128);
rightMotor->setSpeed(128);
```

Arduino with I2C  
motor driver

```
digitalWrite(6, HIGH);
digitalWrite(7, LOW);
analogWrite(10, 128);
```

Arduino with  
L289N H-Bridge

# Example: Making a robot to move

ABSTRACTION

```
ros2 topic pub -r 1 /cmd_vel
geometry_msgs/msg/Twist "{linear: {x: 0.5, y: 0.0,
z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
```

```
Adafruit_DCMotor *leftMotor = AFMS.getMotor(1);
Adafruit_DCMotor *rightMotor = AFMS.getMotor(2);
leftMotor->setSpeed(128);
rightMotor->setSpeed(128);
```

```
digitalWrite(6, HIGH);
digitalWrite(7, LOW);
analogWrite(10, 128);
```

ROS2 topic for  
setting velocity  
for a robot

Arduino with I2C  
motor driver

Arduino with  
L289N H-Bridge

APPLICATION

HARDWARE



# What is ROS(2)?

- Robotics Operating System
- Open-source framework for building complex robotics solutions
- Strong community
  - Proven use cases
  - Fast delivery of patches and new features
  - Wide range of supported hardware
- Industry "standard"
- Multi-platform
- Note ROS vs. ROS2

# Next time... Wednesday Nov 7 at 8:15

- What is ROS?
- Why the command in the previous slide was so complicated?
- Should I learn it by heart?
- Why all this fuzz about ROS?
- *<put your question here – George and Janne will be in your service!>*