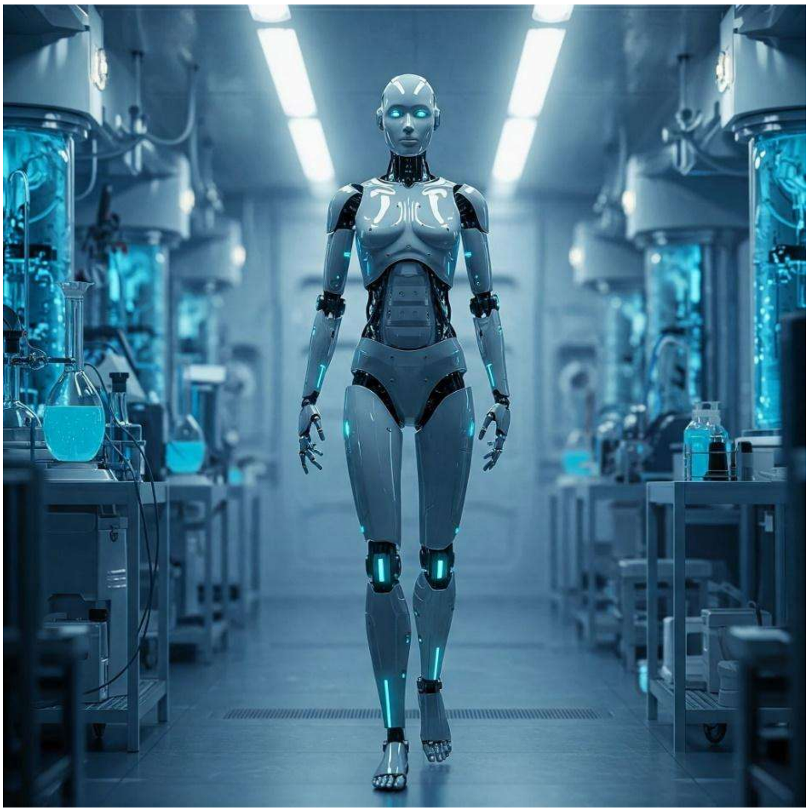
Interreg
Co-funded by the European Union
Slovakia – Austria

ICE-R: Inclusive and Creative Educations with Robotics and AI

Empowering Teachers and Students through Robotics and AI



Introduction to Robotics

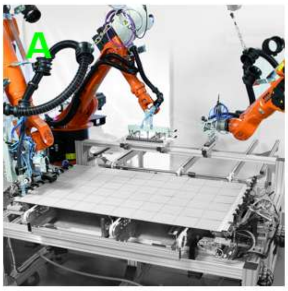
This (English) manual is just a draft version and we will improve it further.
Author will be glsd to receive any comments and corrections to this text.

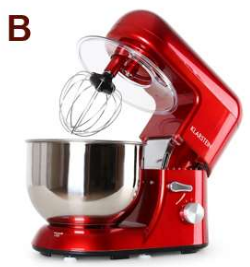
Author: Richard Balogh (balogh@elf.stuba.sk), Slovak University of Technology in Bratislava.

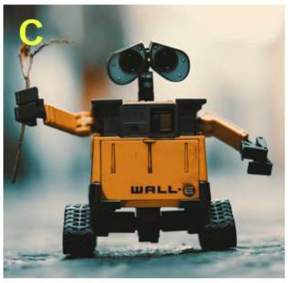
Published by CVTI SR during the ICE-R project

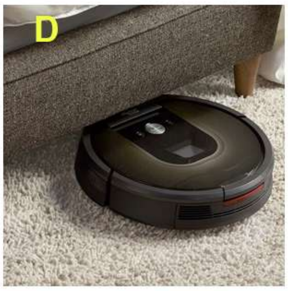
What is a Robot?

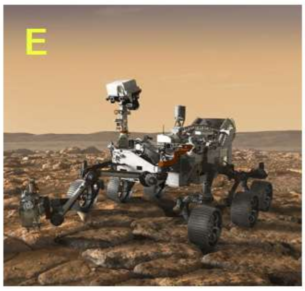
Definition: A robot is a programmable machine capable of carrying out a complex series of actions automatically.

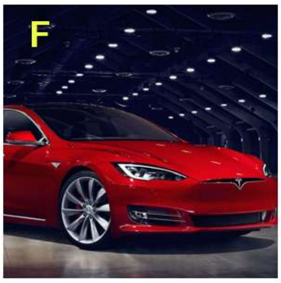


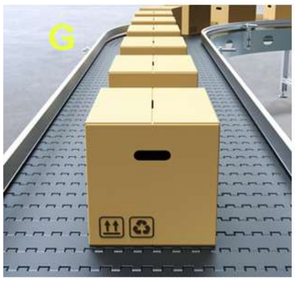


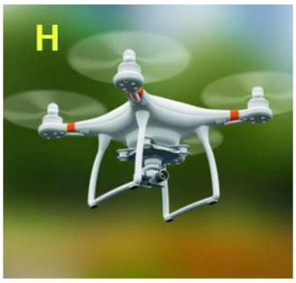








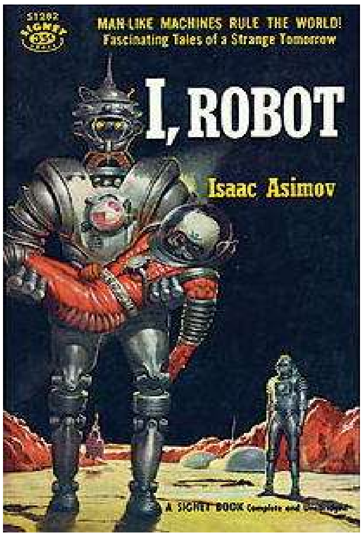




Types of Robots

- Industrial robots (assembly lines, manufacturing)
- Service robots (cleaning, delivery, healthcare)
- Mobile robots (drones, self-driving cars)
- Humanoid robots (human-like form)
- Educational robots (mbot2, etc.)

Robot Laws


S1202
SIGNET
35¢
MAN-LIKE MACHINES RULE THE WORLD!
Fascinating Tales of a Strange Tomorrow
I, ROBOT
Isaac Asimov
A SIGNET BOOK Complete and Unabridged

The Three Laws, presented to be from the fictional "Handbook of Robotics, 56th Edition, 2058 A.D.", are:

Basic Robot Components and Functions

A. Frame/Structure — Holds all the other components together.

- Provides the physical support and shape of the robot.
- Materials: Metal, plastic, etc.

B. Actuators (Motors and Servos) — Enable the robot to move its parts (wheels, arms, etc.).

- Motors: Provide continuous rotational motion.
- Servos: Provide precise angular motion.

C. Sensors — input devices that gather information from the environment, allow the robot to perceive its surroundings.

- Distance sensors (ultrasonic, infrared), light sensors, sound sensors.
- Cameras in various ranges
- Gyroscope/accelerometer.

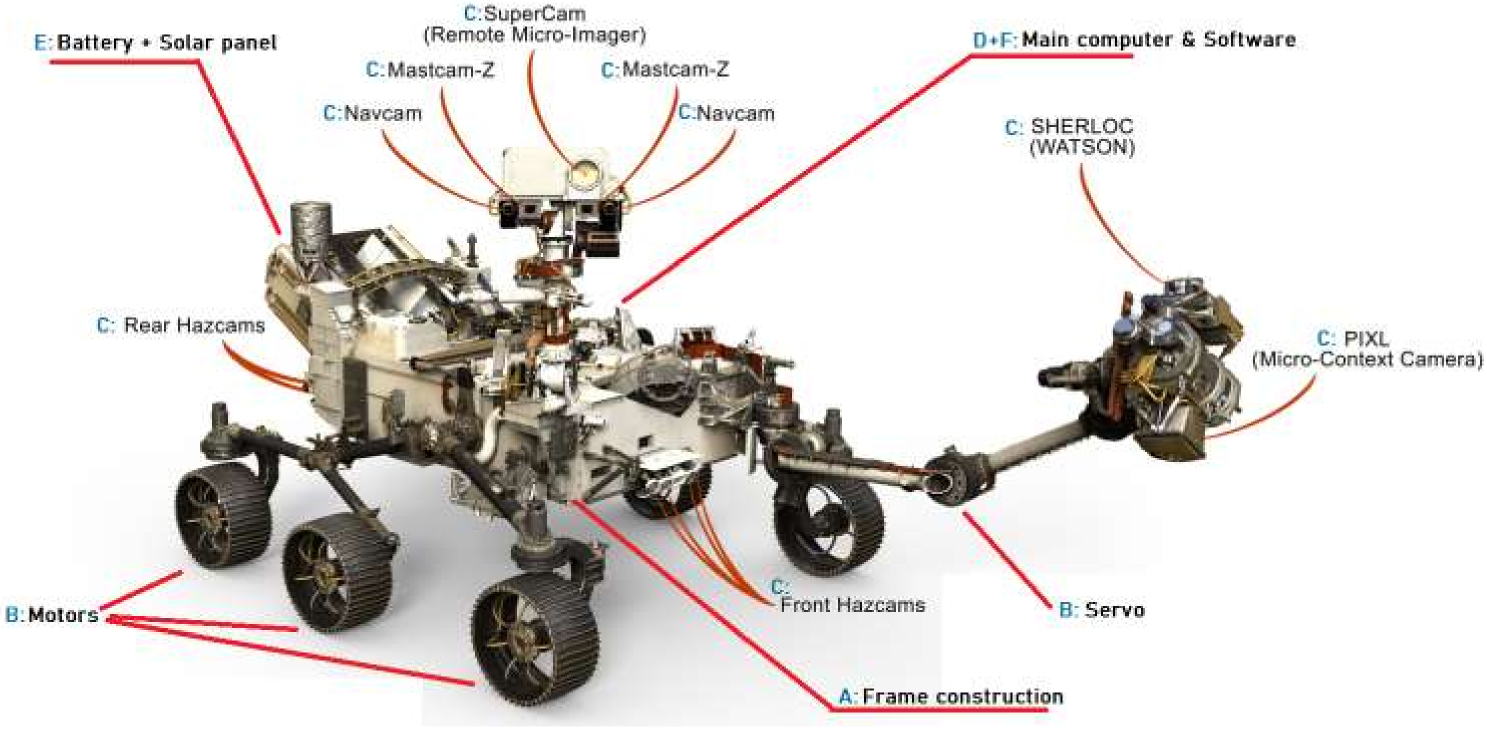
D. Control System (Microcontroller/Computer) — the "brain" of the robot. Makes decisions and executes commands.

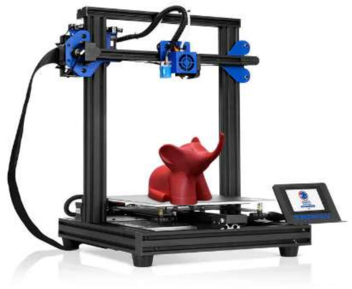
- Processes sensor data and controls actuators.
- Examples: Arduino, Raspberry Pi, CyberPi.

E. Power Source (Battery) — provides electrical energy to the robot. Powers the robot's operation.

- Types: Rechargeable batteries (Li-ion, NiMH).

F. Programming/Software — defines the robot's behavior. Allows the user to give instructions to the robot.





Is this a Robot?

Even that not strictly robots, they use the same technology:

- Sensors + Actuators + Control
- Embedded Microcontrollers
- Visual systems,
- Automation and AI

How Robots Work (Simplified Flow)

- **Sensing:** Sensors gather information from the environment.
- **Processing:** The control system processes the sensor data.

- **Decision-Making:** The control system makes decisions based on the program.
- **Action:** Actuators move the robot's parts according to the decisions.
- **Feedback:** Sensors provide feedback to the control system, allowing for adjustments.

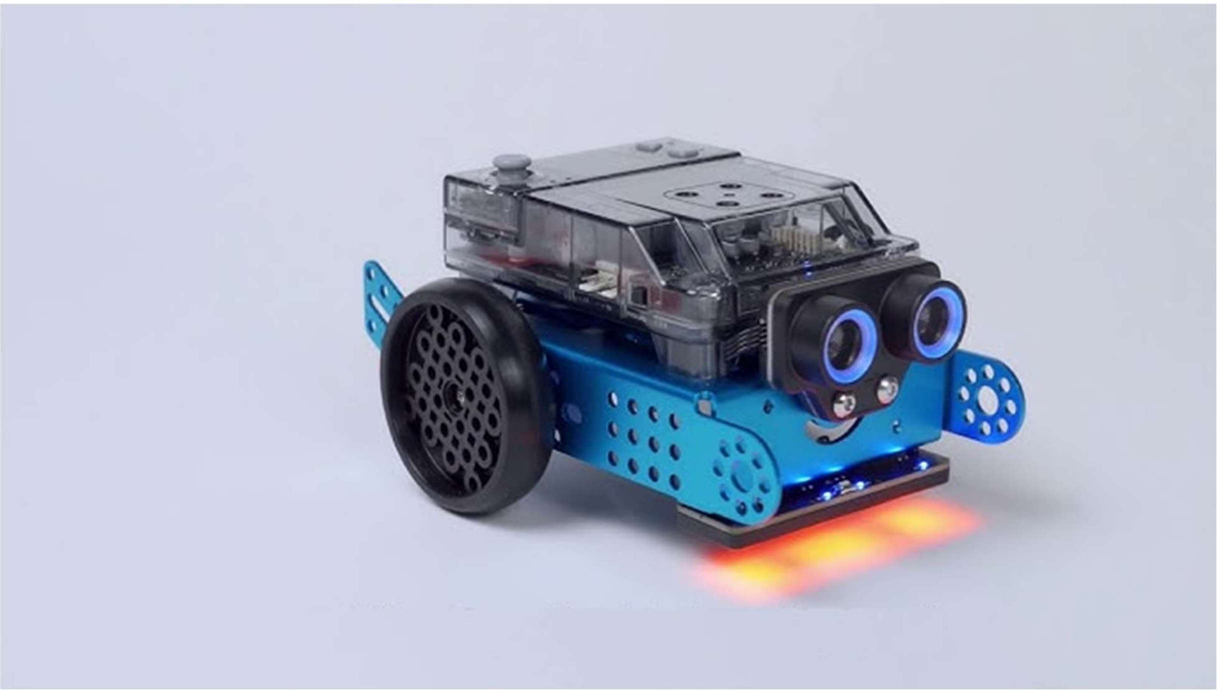
Applications of Robotics

- **Industry:** Automation, manufacturing, assembly lines.
- **Healthcare:** Surgical robots, rehabilitation, patient care.
- **Exploration:** Space exploration, underwater exploration.
- **Transportation:** Self-driving cars, drones.
- **Education:** Teaching STEM concepts, hands-on learning.
- **Home:** Vacuum cleaners, lawnmowers, entertainment.

Future of Robotics

- Artificial intelligence and machine learning.
- Human-robot interaction.
- Ethical considerations.

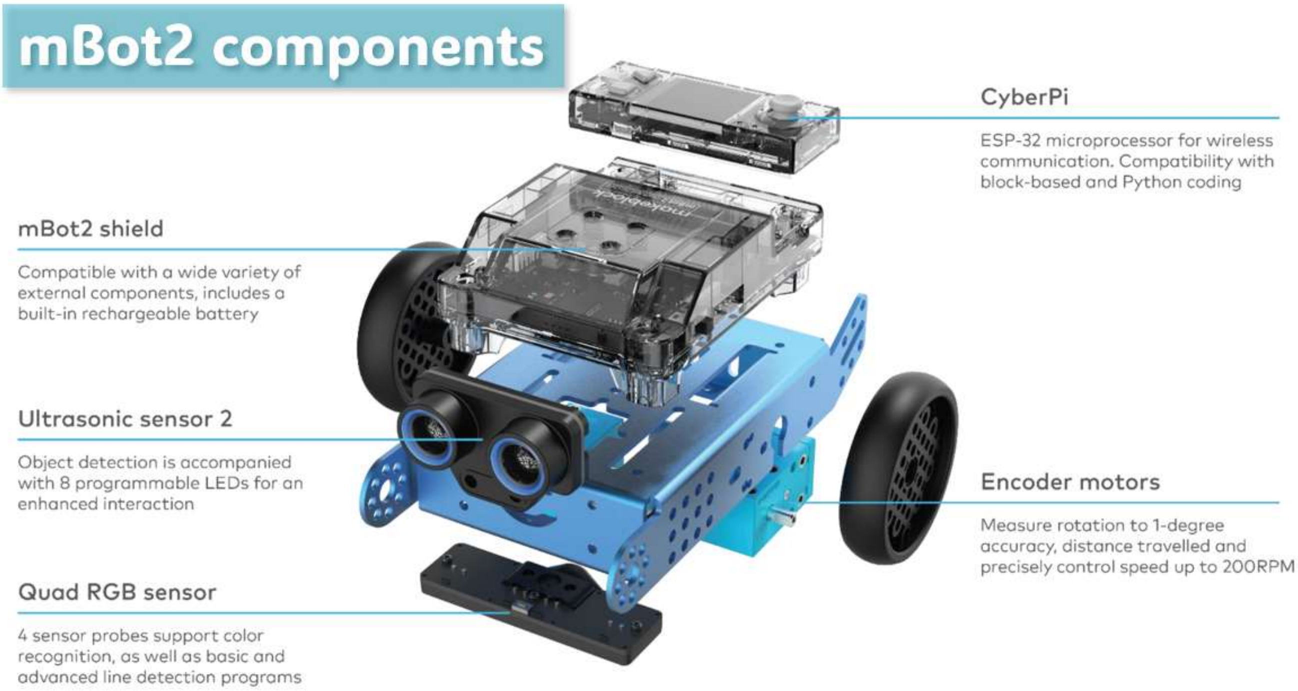
Robot mBot2



The mBot2 is designed to introduce students to robotics, AI, IoT, and data science. It's intended for use in educational settings, from lower secondary to upper secondary levels and beyond. It utilizes the CyberPi, a microcontroller with network capabilities and built-in sensors. It supports block-based coding with mBlock, making it accessible for beginners. It also allows for a smooth transition to object-oriented coding with Python. The mBot2 is designed to be highly expandable, so its capabilities can be increased with additional modules.

Key Technical Parameters:

- **Connectivity:**
  - Wi-Fi.
  - Bluetooth.
- **Processor:**
  - ESP32-WROVER-B.
- **Display:**
  - 1.44-inch color display.
- **Power:**
  - Rechargeable Li-ion battery.
- **Expandability:**
  - mBuild module compatibility.
  - Expandable sensor and motor ports.
- **Coding:**
  - mBlock (block-based).
  - Python.



mBot2 components

CyberPi

ESP-32 microprocessor for wireless communication. Compatibility with block-based and Python coding

mBot2 shield

Compatible with a wide variety of external components, includes a built-in rechargeable battery

Ultrasonic sensor 2

Object detection is accompanied with 8 programmable LEDs for an enhanced interaction

Encoder motors

Measure rotation to 1-degree accuracy, distance travelled and precisely control speed up to 200RPM

Quad RGB sensor

4 sensor probes support color recognition, as well as basic and advanced line detection programs

Core Components:

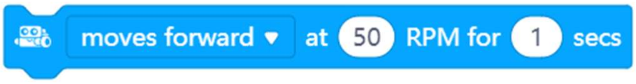
- **CyberPi:** This is the main control board.
    - Processor: ESP32-WROVER-B.
    - Wireless Connectivity: Wi-Fi and Bluetooth.
    - Display: 1.44-inch full-color display.
    - Sensors:
        - Light sensor.
        - Microphone.
        - 3-axis gyroscope and accelerometer.
    - Audio: Speaker.
    - LEDs: RGB LED strip.
- **mBot2 Shield:**
    - Provides motor and servo control.
    - Battery: Built-in rechargeable Li-ion battery (2500mAh).
    - Motor Ports:
        - 2 encoder motor ports.
        - 2 DC motor ports.
        - 4 servo ports.
- **Sensors:**
    - Ultrasonic Sensor 2: For distance measurement.
    - Quad RGB Sensor: For line and color detection.
- **Motors:**
    - Encoder motors: For precise movement control.

Hands-on Activities

Lesson 1:

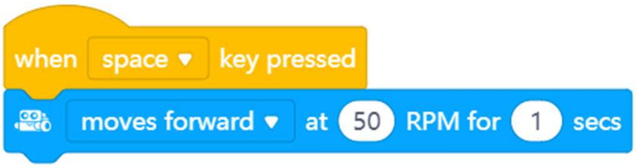
Let's move!

Code block:



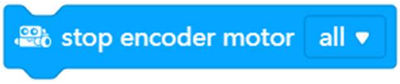
```
moves forward ▾ at 50 RPM for 1 secs
```

This code block allows you to move the mBot2 forward, backward, left and right at a specific rotational speed of the wheels and for a number of seconds. The example below shows how you can make the mBot2 move for two seconds at a speed of 50 rotations per minute. This is useful, for example, when the mBot2 needs to push a load forward.



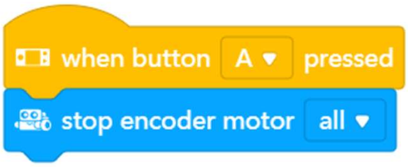
```
when space ▾ key pressed
moves forward ▾ at 50 RPM for 1 secs
```

Code block:



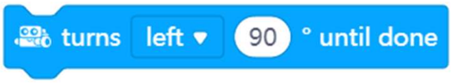
```
stop encoder motor all ▾
```

This code block allows you to make your robot stop moving. This can be a useful block during a testing phase. If a program does not work as expected you can use this block to make the robot stop immediately. For example, you can make the robot stop when the A button of the CyberPi is pressed. To do this, use the programming example below.



```
when button A ▾ pressed
stop encoder motor all ▾
```

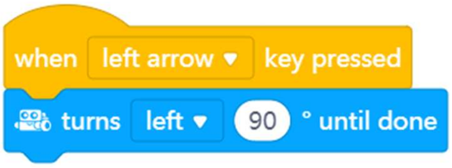
Code block:



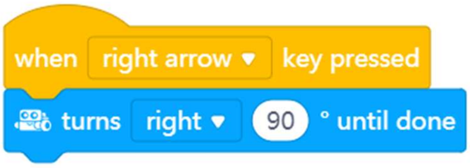
```
turns left ▾ 90 ° until done
```

With this code block you can make the mBot2 rotate by a number of degrees, and you can choose whether the rotation should be to the left or to the right.

When the programming example below is set, the mBot2 can be controlled by using the arrow keys. When pressing the right arrow, the mBot2 will turn 90 degrees to the right, when pressing the left arrow, it will turn 90 degrees to the left.

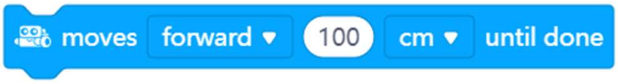


```
when left arrow ▾ key pressed
turns left ▾ 90 ° until done
```



```
when right arrow ▾ key pressed
turns right ▾ 90 ° until done
```

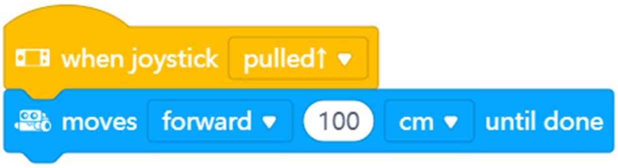
Code block:



```
moves forward ▾ 100 cm ▾ until done
```

With this code block you can make the mBot2 move forward or backward a certain distance.

When you use the programming example below, the mBot2 will move 100 cm forward. To start this, move the joystick of the CyberPi upwards.



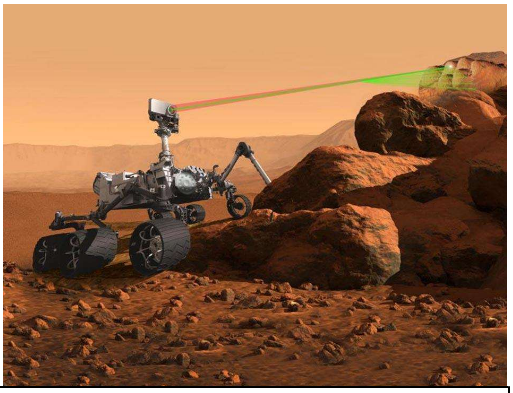
```
when joystick pulled↑ ▾
moves forward ▾ 100 cm ▾ until done
```

Trying it out

It is now time to put your mBot2 to work. Take an A3 paper and draw a maze that your mBot2 has to drive through. Don't make it too complicated and take into account the width of the robot. The mBot2 doesn't have to find the right route by itself. The mBot2 just needs to drive the route that you program

Lesson 2:

Sensing = data



What does this sensor do?

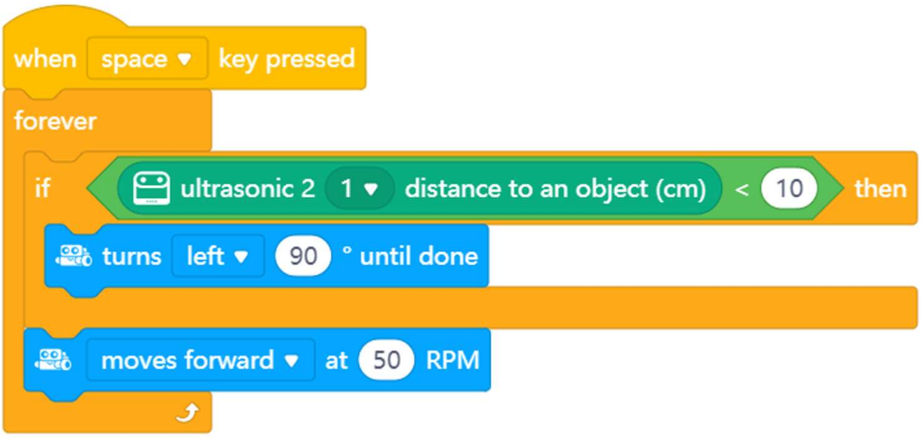
Sound, which is vibration in the form of variations in density and pressure, propagates in air at a constant speed in the form of waves (speed of sound in air approx. 334 m/s). The higher the vibration, the higher the sound appears - up to a limit above which people can no longer perceive this sound. These very high vibrations are called ultrasound. Since the speed of sound in the air is known, sound can be used to determine distances to objects; to do this, one emits a sound and measures the time until this sound is reflected back from the object. This reflected sound wave is also called an echo. Ultrasounds usually are used for this purpose.

For example, an ultrasonic sensor is often used:

- for an imaging procedure, e.g. in pregnancy (each pixel is a distance measurement) or
- in the control of robots to prevent collisions

In the programming example below, the ultrasonic sensor is used to prevent the mBot2 from driving into an obstacle. When the mBot2 is less than 10 cm from an obstacle, the robot makes a 90° turn to the left and then simply continues driving. You can extend this programming example so that the mBot2 moves randomly through a classroom without bumping into tables and chairs.

Programming example



```
when space ▾ key pressed
forever
  if < ultrasonic 2 1 ▾ distance to an object (cm) < 10 > then
    turns left ▾ 90 ° until done
  moves forward ▾ at 50 RPM
```

Wrapping out

- What do you think turned out well?
- What could be better?
- Which parts of the lesson did you find easy and which did you find more difficult?
- What would you like more explanation about?
- Who could help you with that?

Vocabulary of Basic Terms

Actuator: A component that produces motion (e.g., motors, servos).

Algorithm: A step-by-step procedure for solving a problem or completing a task.

Automation: The use of technology to perform tasks with minimal human intervention.

Collaborative Robot (Cobot): A robot designed to work alongside humans in a shared workspace, with built-in safety features to prevent injuries.

Debugging: The process of identifying and fixing errors in a program or robot system.

Degrees of Freedom (DOF): The number of independent movements a robot can make.

Emergency Stop (E-Stop): A safety mechanism that immediately halts all robot operations in case of an emergency.

Encoder: A sensor that provides feedback on the position or speed of a motor.

End Effector: The device at the end of a robotic arm that interacts with the environment (e.g., a gripper).

Feedback: Information returned to a control system to adjust or correct its operation.

Human-Robot Interaction (HRI): The study of how humans and robots interact.

Loop: A programming structure that repeats a set of instructions.

Microcontroller: A small computer on a single integrated circuit that controls a robot's functions.

Motor: A device that converts electrical energy into rotational mechanical motion.

Navigation: The ability of a robot to move from one location to another.

Payload: The weight or load a robot can carry.

Program: A set of instructions that tells a robot what to do.

Risk Assessment: A systematic process to identify and evaluate potential hazards associated with robot operation.

Robot: A programmable machine that can perform tasks automatically.

Safe Zones/Work Cells: Designated areas where robots operate, separated from human workers by physical barriers or safety systems.

Safety Sensors: Sensors specifically designed to detect hazardous conditions or prevent collisions (e.g., laser scanners, light curtains).

Sensor: A device that detects and measures physical quantities (e.g., light, sound, distance).

Servo: A type of motor that allows precise control of angular position.