Introduction

Lesson 1.-2.

Materials needed:

- LEGO EV3 base robot assembled without sensors
- EV3 base robot sensors and LEGO pieces for building

Robots…what....how….why?

During many centuries, humans have been trying to create machines that would ease their work. During the last century, several machines which eased humans' physical work were created, e.g. excavators for digging, cranes for lifting heavy materials, different means of transport for moving fast, such as trains, automobiles, planes, etc. Nowadays, due to the development of intelligent machines, it would be possible to free humans from routine work completely. This allows to free more people for creative, intellectual work, such as creating robots.

All robots can make decisions and intelligent choices. In some more difficult situations, robots can communicate with humans and ask for additional information. Some robots are able to learn, i.e. collect experience and change their behavior to some limit without external help. All algorithms, which are the basis for robot's decisions, are programmed by the creators of a robot. Robots behave according to the algorithm and all their intelligence depends on their creators.

Co-funded by the Erasmus+ Programme of the European Union

For example, a car has the airbag system, which is based on the acceleration sensor. If the car engine is running, the airbag management system in a small computer is working and comparing the acceleration data. If the car loses speed faster than possible with brakes, it means that a crash has happened and the computer sends the signal to open the airbags. This is a simplified description as modern airbags have many other sensors. The computer takes into account the presence of a security seat, the weight of the traveller, and other important factors. The whole system is functioning automatically and does not need any human intervention, so it can be called an autonomous system. As the car needs a human driver, the car itself is not a robot; however, inside the car, there are many robotic subsystems in addition to the airbag subsystem.

Robots do not need to have human form - this common disbelief originates in science fiction films. Every robot has to have three parts: sensors, which receive signals from the surrounding environment, a management device (i.e. a brain), which analyses those signals and makes decisions, and an actuator, which executes the brain's commands. It is a widespread knowledge that a domestic washing machine is a robot. As a washing machine has sensors, a brain and actuators, it can indeed be called a robot - an independent device, which does not need any human help during one washing cycle. There are many sensors inside the washing machine, but let us have a closer look on how it warms the water. The washing machine receives a signal from the temperature sensor about the temperature of the water, and if the water is not warm enough, the brain gives a command to the heater to continue heating.

As robots have been created by humans to ease people's work, some rules have to apply in the relationship between humans and robots, which would forbid robots from fulfilling some commands. The author of many science fiction books, Isaac Asimov, was the first to formulate those rules in his short story Runaround already in 1942. Good robots fulfil the following rules:

- A robot can not harm a human with its actions or inactions.
- A robot has to follow the commands of a human, provided that these commands are not in contradiction with the first rule.
- A robot has to defend its existence, provided it is not in contradiction with the first and second rule.

For discussion: If you had to formulate rules for robots, what kind of rules would you add?

Robots can be divided into three generations according to their development level:

- First generation robots had a simple construction and no sensors (i.e. an understanding of the surrounding environment). They were able to do their work in unchanging conditions. For example, a first generation robot would be a robotic hand which lifts bottles into boxes in a brewery. The robot performs identical movements, but as it receives no signals from the surrounding environment, it has no feedback on its actions.Therefore, it can not work properly under changing conditions (no more bottles or boxes of different size). These kinds of robots are not called real robots anymore.
- Second generation robots can react to the changing environment, due to sensors. The efficiency of the robot now depends on its software. All robots used in this course belong to the second generation since we use sensors and program them to act independently.
- Third generation. This generation is still under development. Robots of the third generation can analyse situations and learn from their experience. For example, the robot from the movie “I, robot”, which is able to make conclusions and learn, is a third generation robot.

Manipulators

A manipulator is a device or a machine ruled by a human. Contradictory to common understanding, manipulators are not robots as they are directly driven by humans. Manipulators are not able to work independently. The information to drive the manipulator comes from the human mind; it is analysed and the decisions are made by a human driver. Consequently, manipulators are only actuators. Well-known manipulators are, for example, children’s remote control cars and remote control devices.

Sensors, microcontrollers and actuators

A sensor is a device, which measures a parameter and converts it into an electrical signal. From a simplified viewpoint, sensors are a robot’s sense organs, like human eyes, nose, ears, etc. In human’s neural networks, a signal moves in the form of electrical impulses. The diferent

components of a robot also communicate by using electrical signals. The robot receives an electrical signal from the sensor and behaves according to this signal. The more sensors a robot is using, the better we can shape the robot's behavior and the more tasks we can entrust it to do. To manage the information received from the sensors, the robot needs a controller. Robot's controller is often called a microcontroller as it is really small in dimensions. The controller is robot's brain. The signal from the sensor travels to the input of the microcontroller, where it is analyzed and according to this information, the robot makes its next moves. In the memory of the controller, there is an algorithm based on which the robot makes its decisions. The software for the microcontroller is composed according to the task the robot has to perform. Therefore, if we connected a video camera with the robot, we would have to write a software to recognize objects in its visual field. Based on the signal from the sensor, the microprocessor starts the actuator. The actuator can play a role in managing any influenceable device, e.g. motors, heating devices, etc. The actuators for a robot are like hands and legs for a human. The aim of actuators is to influence the environment. For example, a robot which controls the heating switches the heating on only if the temperature falls below the given critical barrier and switches it off again when the given temperature is reached.

Robots' area of usage

As robots are mechanical devices, they are tireless, which makes them the ideal candidates for routine work. This is why many industrial enterprises use robots for the tedious parts of their work. In addition, robots are not as sensitive to the working conditions as human workers are. They can be used in noisy, badly lighted workplaces, or even in workplaces which are dangerous for humans due to radiation, for example, etc. Robots can also be used in places where we can not send humans - at the bottom of an ocean, on Mars, etc. Robots are also used in entertainment, examples of which are intelligent toys, dancing robots, etc.

An overview of mechatronics

Mechatronics is a multifarious branch of technical science, which combines mechanics, electronics, computer technology, software technology, control theory, and systems' design with an aim to create a better end product. The name "mechatronic" is coined from the words "mechanics" and "electronics" - mechanics for building devices and electronics for managing them. The word has a Japanese origin.

There is a also a third term related to robotics and mechatronics - automation. Automation is defined as a field of technology which uses control theory, information and communication technology to reduce the need for human workforce in the manufacturing process.The word “automatisation” stems from the Anglo-American culture.

A robot is any system which has a sensor, a controller and an actuator. The sensor gives the controller information about the surrounding environment. The controller processes it and controls the device’s actions by ruling the actuator to fulfil tasks: to move something, switch something, etc.

How to use Mindstorms software?

With this material, EV3 User Guide is attached. Please check pages 42 and 46 to understand how to use the software. User Guide is available in Spanish, Norwegian and English.

What is this robot doing?

Start program number one in Robotic Systems folder on the robot. Ask for teacher’s help if needed. Try to interact with the robot when the program is running and guess what the program is commanding the robot to do. Write a description of the program.
It could be something like this:

This program makes the robot move when somebody shows the light sensor green color. The robot does not act upon seeing any other color.

Start programs 2-4 and try to guess the algorithm of each program. You have to interact with the robot to find out what is triggering it to move. Write a description of the program. Compare your story with the stories of your classmates. Were they the same or different?

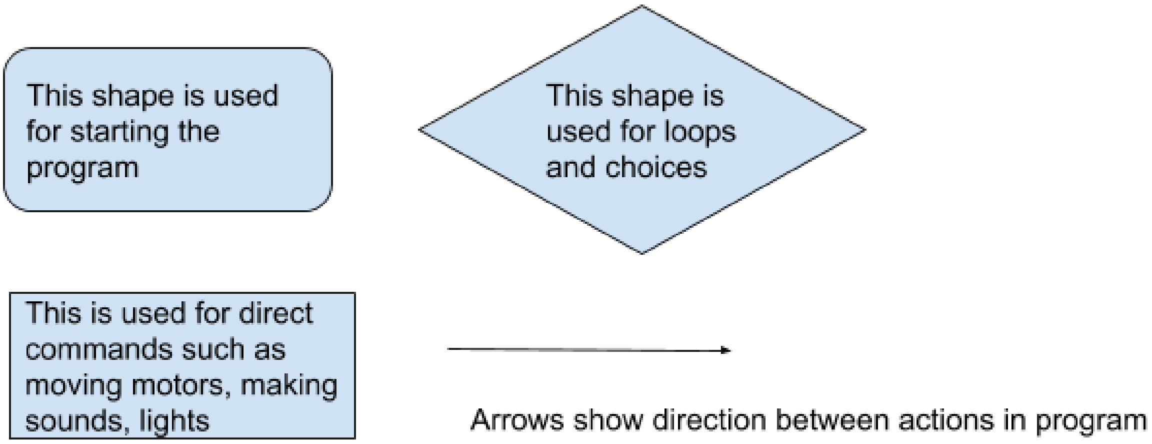
Program 2

Program 3

Program4

Program5

*Start programs 2-5 and try to guess the algorithm of each program. Draw an algorithm of the program. Compare your story with the stories of classmates. Do not open the computer to see the program. You have to guess it by the actions of the robot. Use the shapes given below.


This shape is used for starting the program

This shape is used for loops and choices

This is used for direct commands such as moving motors, making sounds, lights

Arrows show direction between actions in program

Introduction to 21st century skills

The world is currently experiencing a 4th industrial revolution, one that emphasizes on automation, cyber-physical systems and the Internet of Things (IoT). Thus, 21st century skills such as digital and STEM (Science, Technology, Engineering, Mathematics) competences, analytical and critical thinking, team spirit and cooperation, are seen as necessary to improve our learning abilities and to better our future career opportunities, all while adapting and keeping up to date with the shifts in technology.

"21st century skills" are described as the skills that today's students will need, in order to be successful in this everchanging world. One of the most recognizable representations of these skills are the 4C's: communication, collaboration, critical thinking and creativity. However, 21st century skills also include social and emotional intelligence, technological literacy and problem-solving abilities.

Communication
Collaboration
Critical thinking
Creativity

***Reflection Activity 4C's:** Talk with a partner about why these skills will be particularly important and which of these skills you think you are the strongest in? (2min)*

These skills are becoming an important aspect of curriculums. The need for 21st century skills has spurred the integration of technology along with the emphasis on STEM in classrooms. These concepts help students develop the higher thinking skills that colleges and employers are looking for. Education is changing and placing greater importance on preparing students for the real world, which is why 21st century skills are more important than ever.

When students are engaged in hands-on STEM experiences, they build confidence, grow their knowledge and develop habits of learning. When adults coach these students, they encourage them to try, fail, and try again, while connecting STEM concepts to real-world examples. Through this curriculum we aim to make the students excited about working with robotics and to equip them with the skills and confidence they need to excel in the workforce of tomorrow.

The upcoming lessons will focus on combining the 4C's skills with developing technological know-how, including robot design, problem solving and algorithmic thinking.

Learning outcomes

Assess together with your friends, how did you reach following learning goals. What is the meaning of:

- Robot
- Manipulator
- Mechatronics
- Sensor
- Actuator
- Controller
- Generations of robots
- I. Asimov's laws
- 21st century skills

Co-funded by the Erasmus+ Programme of the European Union

Output devices - motors

Lesson 3.-4.

Materials needed:

- Assembled LEGO EV3 base robot without sensors
- Measure tape or longer ruler
- Calculator, Black Duct tape

Output devices – why do we need them?

Robots are using different output devices for moving, showing something or giving a feedback for the user and/or surrounding. A robot is using different motors for moving (like a car), climbing (like a spider), waving a hand or lifting something. Motors are used to move a robot from one place to another or to move robots' different parts. For example, a motor is moving a robotic arm in the factory even if the robot itself is stuck on the ground.

Other output devices like screens, lights, speakers, buzzers etc. give the user information or feedback. For example, we are using an ATM to take some money out, we use the screen to see which buttons to we need to push or what options do we have. We also use screen of our mobile phone to see a Youtube movie or to see which friend is calling.

A light is often used to give us some information or notification – for example when we push an elevator button and the button did not light up, we sure press it again, right? Or when there is a red light in the traffic light, we sure must stop. Or when the tiny red light from your mobile phone is on, it might mean that the battery will be soon empty.

Co-funded by the Erasmus+ Programme of the European Union

With the same purpose we also use sound to attract our attention. We use the speakers in our mobile phones to hear what the other side is speaking, or we use the speaker just to know that somebody is calling. Sometimes we hear the instructions on some device, like on the train to be sure that this is our stop.

So, all these output devices make our life easier, cleverer and safer.

In these 2 lessons we are focusing on different motors and actuators and are learning how to move the EV3 robot or how to move its' parts.

Motors and actuators

Motor or better to say electric motor is a device that converts electrical energy to mechanical energy. To choose the right motor for your robot, you must think of some parameters: the current specialty and consumption, dimensions of the motor, motor connecting options, control complexity, rotation angle measurement, rotation speed and torque.

The rotation speed unit RPM (Revolutions Per Minute), this means how many rotations a motor can do for one minute. The more RPM is the faster the motor rotates.

In this lesson we divide the electrical motors to two main categories: AC (Alternating Current) motor and DC (Direct Current) motor (see Figure X).

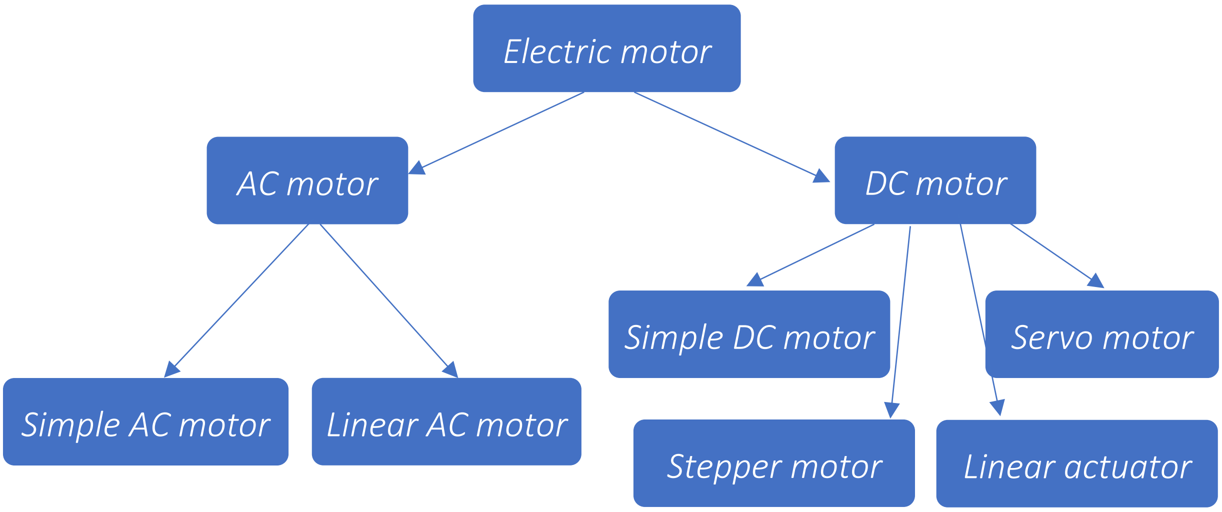


Figure 1. Different type of motors

AC motor converts Alternating current energy to mechanical energy. It uses typically 220V voltage with 50 Hz – the mains supply voltage. **Linear AC motor** converts electric energy to straight-line motion.

DC motor converts Direct current energy to mechanical energy. The rotation speed can be change with the supply voltage. The rotation angle of the simple DC motor can't be determined.

A **servo motor** is a simple DC motor that has a gearbox and feedback opportunity (rotation sensor) inside the motor and it has a controlling mechanism. A normal servo motor can usually rotate 180 degrees or 360 degrees, depending of the type. A continuous servo motor can rotate continuously.

A **stepper motor** is a DC motor that rotates with steps. A **linear actuator** is a DC motor that converts the electrical energy to mechanical straight-line motion.

Activities

Activity no1:

Use Google to see these different types of motors and find the suitable application on the right.

1. Simple AC motor	A. DVD drive
2. Linear Induction motor	B. LEGO Mindstorms EV3 Large motor
3. Simple DC motor	C. Ultra-high-speed train
4. Servo motor	D. Printer head
5. Continuous servo motor	E. Remotely controlled Toy Car
6. Stepper motor	F. Small robotic hand that moves 180°
7. Linear actuator	G. Big industrial conveyor

Fill in:

1. ….. ; **2. ….. ;** **3. ….. ;** **4. ….. ;**

5. …. ; **6. ….. ;** **7. ….. ;** **8. ….. ;**

LEGO Mindstorms EV3 robot has 2 Large and 1 Medium servo motors (see Figure 2.).

The rotor inside the Large motor rotates direct axis (alongside the motor) and the rotor in the Medium motor rotates cross axis (crosswise the motor).

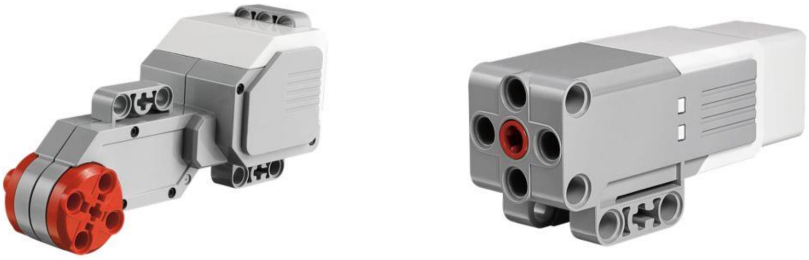


Figure 2. Left is the LEGO Mindstorms EV3 Large motor and right is the Medium motor

EV3 Large motor

There are multiple ways to program the Large motor to move:

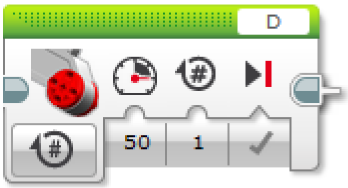


Figure 3. Large motor block

With this block you can move only one motor at a time.

The port number **D** shows the which motor is about to move.

Number **50** shows the Motor Power and number **1** the Duration in rotations. Meaning a motor is making 1 rotation with the power of 50.

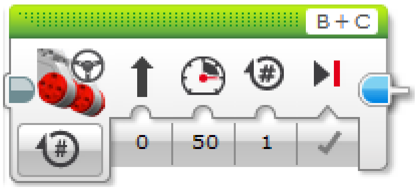


Figure 4. Move Steering block

With this block you can move your robot with two motors. Move Steering block works in a way that motor B is a "master" and motor C is a "slave" and according to the Steering value the robot is moving to different directions. "Slave" is trying to follow the rotations of "master".

Table 1. Different values of Steering with the Motor Power of 50 (and with EV3 Base robot)

Steering value	Robot direction	Steering value	Robot direction
0	Straight forward		
-25	Robot turns left ~45 deg.	25	Robot turns right ~45 deg.
-50	Robot turns left ~90 deg.	50	Robot turns right ~90 deg.
-100	Robot turns left ~180 deg.	100	Robot turns right ~180 deg.

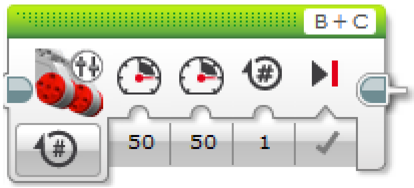
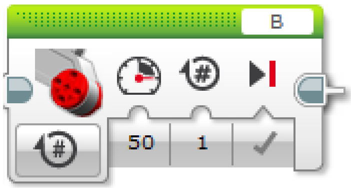


Figure 5. Move Tank block

With this block you can give a different power to both motors at the same time. When the Power values are equally positive and over 0, then the robot is moving straight forward when under 0, robot is moving straight backward and with other combinations the robot is turning accordingly. Try yourself!

There are multiple ways to turn the robot**:**



With this block only the motor in port B is moving. This means that the robot is turning over the middle point of motor C and the turning radius **r** is the distance between these two wheels. (See Figure 6.)

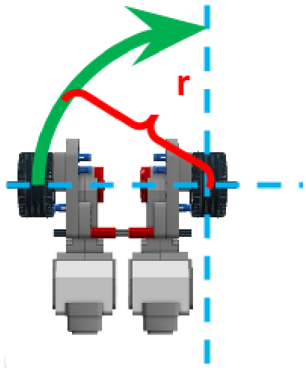
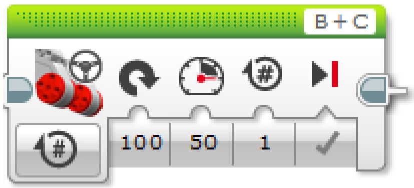


Figure 6. Robot turning with one motor (Author: Rasmus Kits)



With this setup the robot is turning on spot. With Steering value 100 its turning right and with -100 left. On spot means that the robot is turning over its middle point and not any more over the wheel middle point. The turning radius **r** is the distance between the wheel and robot middle point. (See Figure 7.)

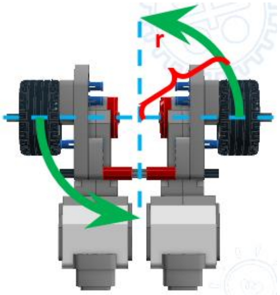
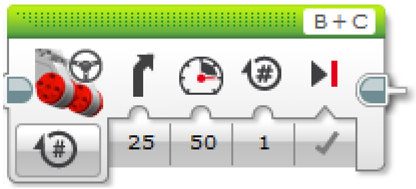


Figure 7. Robot turning over its middle point. (Author: Rasmus Kits)



With this setup the robot is turning with two motors and the turning radius and directions depends on the *Steering* value. The smaller the value is the bigger the turning radius is. Positive value turns robot right, negative value turns robot left. The Steering value 0 asks the robot to go straight forward.

Activity no2

Use the LEGO Mindstorms EV3 programming software to move your base robot:

- 1 rotation forward
- 5 rotations forward and turn right with duration of 2 rotations
- 3 rotations forward, then turn the robot left or right with duration of 380 degrees and drive backwards 3 rotations.
- Try the *Steering* values in Table 1 and see what happens? Use your own *Steering* values also.

Activity no3

Now use the ruler or the measure tape to measure the distance how far the robot moved. Make at least three tests.

Fill in:

Test 1. ………. cm;

Test 2. ………. cm;

Test 3. ………. cm;

Was there any difference?

Now the teacher will measure out a distance (over 100 cm) and puts two duct tapes – one in one side and the other into other side. So, the distance between these lines are the one that teacher thought. Then the teacher gives you the exact number and you have 1 minute to calculate and program your robot to drive from one duct tape to another. The closer you get the better result.

Make a small competition with your classmates and see who gets closest.

Hint: *Cross multiplication might come handy.*

Name at least three factors that affect the results.

1. ………………………….…

2. ……………………………..

3. ……………………………..

Activity no4

Make the robot to drive a square. The side length of the square must be ~35cm.

What two ways there is to achieve this?

1. ……………………….

2. ……………………….

Hint: *You can try to use the* ***Loop*** *(in orange* ***Flow Control*** *palette ->* ***Loop****).*

What should be the loop condition? And why?

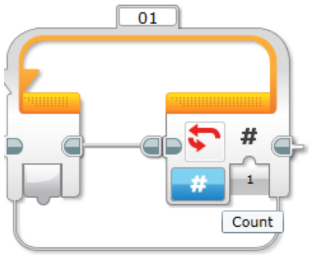
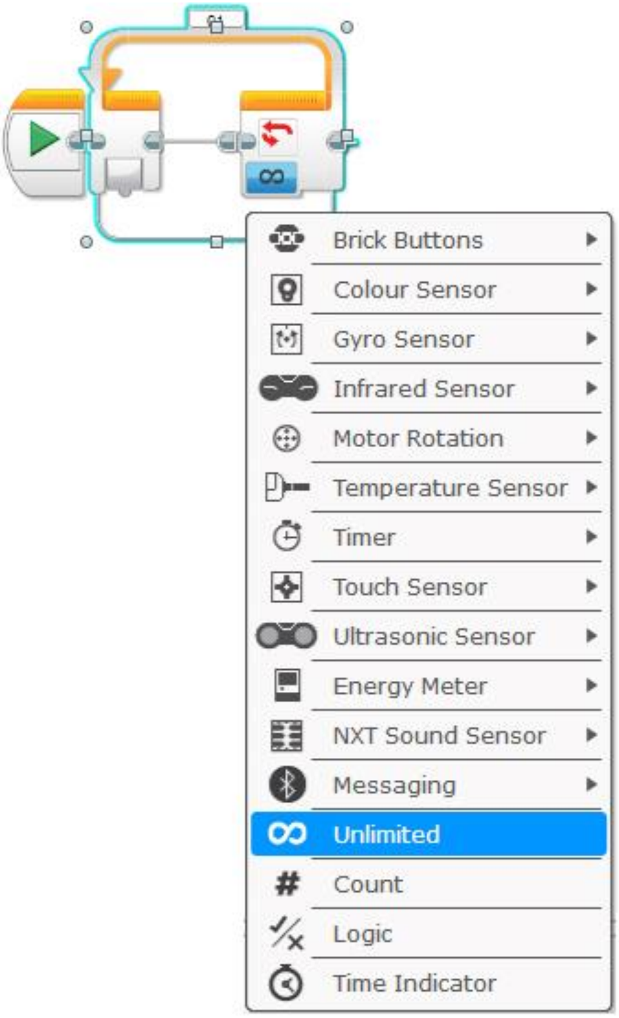


Figure 8. Programming block for Loop

Hint: *You can get familiar with the Loop block after the Learning outcomes section.*

Loop block



You can change the Loop condition by pushing the Infinity symbol (∞).

<u>You can choose between these conditions:</u>

Unlimited is the default condition for the Loop block, meaning that the loop will run unlimited and end only when the User stops the program (or the battery of the robot will deplete).

Time Indicator means that the loop will repeat until a certain time will pass. 5 seconds is the default value. You can also use a decimal point for less or more accurate time.

Count means that the loop will repeat certain cycles. For example, count 5 means that the loop will repeat 5 times.

Logic means that the loop will repeat until the Logic value will be True (by default) or False when the User changes it. This means that inside or outside of the Loop a Logic operation must be done and then the result must be wired into Loop condition.

Sensors (Starting with Brick Buttons and ending with Messaging). You can use the value from the sensors to end the Loop. All kind of different type of sensors can be used for that. Be aware that the sensor value must match exactly with the condition when the program reaches to the end of the Loop otherwise the loop condition is not reached, and the loop will not stop.

Learning outcomes

Assess together with your partner, how did you reach following learning goals and how did you use math to achieve those.

What is the meaning of:

- Rotor
- Motor
- Continuous servo
- Direct axis vs cross axis
- Loop block

Output devices – motors 2

Lesson 5.-6.

Materials needed:

- Assembled LEGO EV3 base robot with grabber in front (with Medium motor)
- Assembled EV3 block made of LEGO bricks
- LEGO minifigure

In this lesson we are learning how to control the LEGO Mindstorms EV3 Medium motor and how to use the Large motors for more advanced tasks.

Activities

Activity no1:

Use the LEGO Mindstorms EV3 programming software to move the front grabber down and up. Use the Medium motor block to do it. (See Figure 1).

Think about it:

Did you manage it with first time? If not, what happened?

Did the program was stuck on some point? Why?

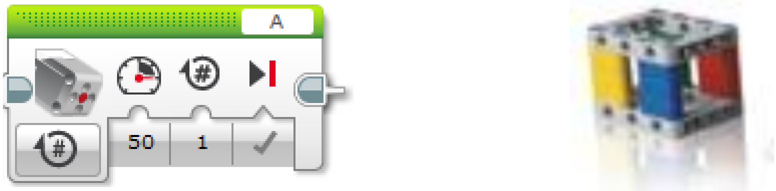


Figure 1. Medium Motor block on the left and assembled LEGO block on the right

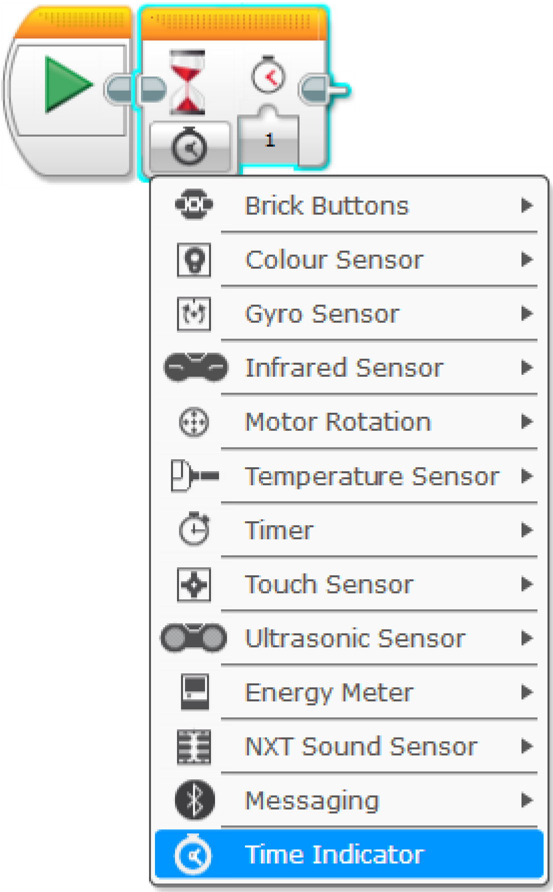
Co-funded by the Erasmus+ Programme of the European Union

Activity no2:

Put the assembled LEGO block (see Figure 1) into some distance and try to drive the robot towards the block, then grab it, stop for 2 seconds and then drive back to starting position and release it.

***Hint**. For stopping use the Wait block (**Flow Control** -> **Wait**).*

Wait block



You can use ***Wait*** block to wait for certain condition to be true. You can choose between these conditions:

Time Indicator is the default condition. This means that the program is waiting for a certain amount of Time. Default is 1 second. You can also use a decimal point for more accurate time.

Sensors are the also very often used. This means that the robot is waiting for a sensor to be activated (like touch sensor for example) or a certain sensor value is reached (for example, the program is waiting for the distance of Ultrasonic sensors to reach more than 50 cm and then the program continues.

You can use all the LEGO Mindstorms EV3 sensors, Timer value, LEGO Mindstorms NXT Sound sensor, Bluetooth message or even EV3 Brick Buttons.

Activity no3:

Smooth Tank – use the *Move Tank* programming block to move the robot straight forward 2 rotations.

Think about it:

What are the main differences between the *Move Steering* and *Move Tank* blocks? Name some cases where to use one or another.

Now try to make a robot that accelerates smoothly. Use the *Move Tank* block. For accelerating use the *Loop Index* (see Figure 2.)

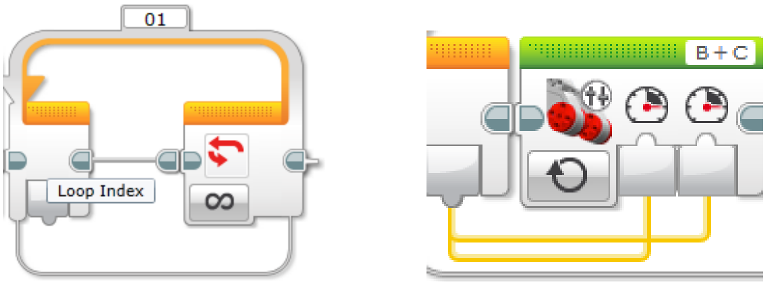


Figure 2. Using a Loop and Loop Index (left) and Loop index and Move Tank sample (right)

What else should the program have?

If available, put the LEGO minifigure standing on top of the robot and try to accelerate so, that the LEGO minifigure stays up.

* Try to stop the robot also smoothly. You can use the *Math* block to do so.

Learning outcomes

Assess together with your partner, how did you reach following learning goals and how did you use math to achieve these.

What is:

- the difference between *Move Steering* and *Move Tank*
- the *Loop Index*
- the requirement of using a *Motor On* in motor block condition

Output devices – feedback

Lesson 7.-8.

Materials needed:

- Assembled LEGO EV3 base robot

In this lesson we are learning about different output devices that are giving audiovisual feedback for the user.

Indicators and Displays

Different robots use visual indicators to give the feedback. We can use a simple light to give a notification or the display to give information, instructions, or just playing some fun stuff.

The easiest indicator is a Light Emitted Diode (LED) that can be controlled directly by the controller output (see Figure 1). Switching the LED on and off is often the first task when programming a microcontroller.

LED is a semiconductor light source that emits light when current flows through it. Different color LEDs need different supply voltages.

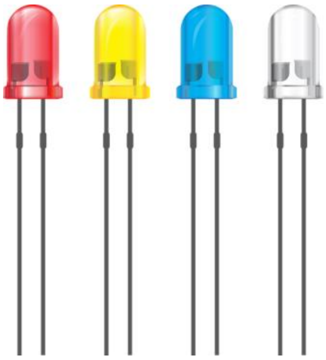


Figure 1. Different color LED-s

Think about it:

Name some of the applications at your home that uses a single LED for feedback.

Share your thoughts with other classmates.

Sometimes we can’t give full information with just one LED, so we need to use them more and combined. Next step is to use a **7-segment number indicator** that consist of multiple LEDs (see Figure 2).

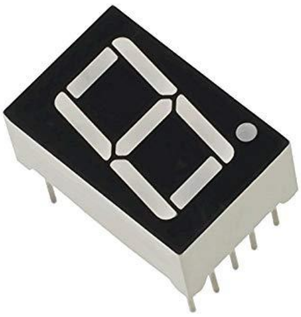


Figure 2. A 7-segment LED indicator

These indicators are used for showing numbers and sometimes letters and are commonly used in small electronics, calculators etc. when there is no need to show large information.

When there is a need to show more text then for robotics and electronics people use a LED matrix display (see Figure 3).

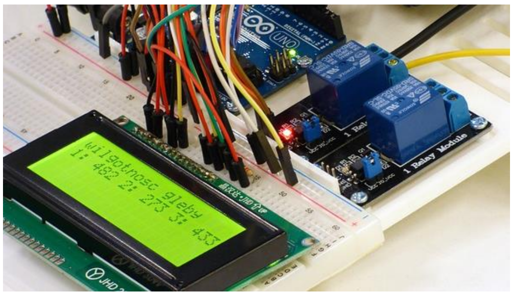


Figure 3. LED matrix display

Simple matrix displays are usually one-colored and might not even have a backlight, just like the Mindstorms EV3 robots. More complicated LED displays are more expensive, harder to control and use more complicated technologies.

Think about it:

Name some of the devices at your home that uses a LED display to show information.

Share your thoughts with other classmates.

Speaker

Sometimes the robots or systems do not need to show any information or attract the user with visual feedback, for example a simple Smoke detector in the ceiling has a LED for showing that it is active but when there is smoke near the device, it activates a very loud sound to alarm people. Sometimes different devices use a speaker to give instructions or error codes as well and some robots are just communicating with the user. Another example is a modern car that has parking sensors. The closer the object is the more frequently the beeping sound plays.

Activities

Activity no1:

Your LEGO Mindstorms EV3 robot has LED lights in the buttons area. Usually there is a red light on, when the robot is starting or closing and green light when the robot is ON and waiting the user. Green light flashes when the program inside the robot is active.

Your first task is to program your robot to show red light for 3 seconds, green light for 2 seconds and orange light for 1 second. You can use the *Brick Status Light* block and *Wait* block (see Figure 3).

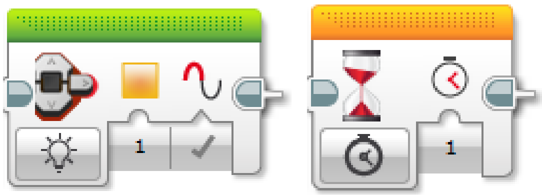


Figure 3. The *Brick Status Light* block on the left and *Wait* block on the right

Activity no2:

Next task is to imitate a Traffic light. Use your friends or Google to figure out the cycle of a traffic light. Then program your robot to act like a traffic light. Cycle will start with red light ON.

> **Think about it**:
>
> What do you need to modify your traffic light to listen also to the pedestrians that want to cross the street?

Activity no3:

Next, we are going to use the LEGO Mindstorms EV3 screen to display different things. First try to show a Smiley Face on the EV3 screen. You can use the *Display* block (see Figure 4).

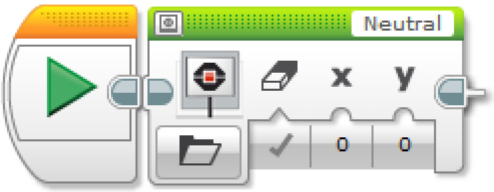


Figure 4. Display block in EV3 program

Did the program work for first time? What should you add to see the Smiley Face longer?

Activity no4:

Make a 30 second screensaver or animation. You can use different preloaded LEGO Image files, make your own or use the Display block settings to move one picture on your EV3 screen.

Activity no5:

Your LEGO Mindstorms EV3 robot has a built-in speaker. Program your robot to bark like a dog. You can use the *Sound* block (see figure 5).

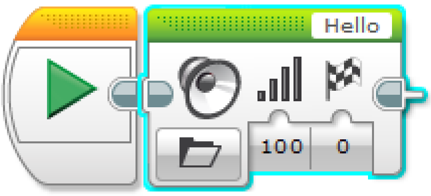


Figure 5. Sound block in EV3 program

Activity no6:

Use *Play Note* or *Play Tone* in your *Sound* block to program your robot to play a well-known song (for example a “Mary had a little lamb” etc. Use Google to find the *Notes* for the song.

Learning outcomes

- Students know what 3-base LED colors they can use with EV3 brick.
- They know how to use pre-made figures in Display block.
- They know what files, tones and notes they can use with Sound block.

Sensors

Lesson 9.-10.

Materials needed:

- LEGO EV3 base robot assembled with all sensors

Sensors are senses for robots

Humans would not be able to exist on Earth without senses. Imagine that one moment you would not hear, smell, feel touch, see or taste any more. Would it be possible for you to receive any information or feedback from the world?

Robots are like us – we want them to navigate and operate in the same world we live in. We have to give them sensors to detect objects, distances, sound etc.

Can you link what sensors robots would have to use to be like humans?

Human sense	Corresponding sensor
Hearing	
Seeing	
Touching	
Tasting	
Smelling	

Did you manage to find answer for also senses like tasting and smelling? Is the science there already?

Google it if you haven't done it so far!

Co-funded by the Erasmus+ Programme of the European Union

How are robots feeling?

Pair up with your friend and try out the following:

1. Close your eyes and try to walk around the classroom or wherever possible
2. Your friend will tap you on the shoulder when distance between you and the obstacles in front of you is less than half a meter.
3. See if you can navigate to the door
4. Change roles

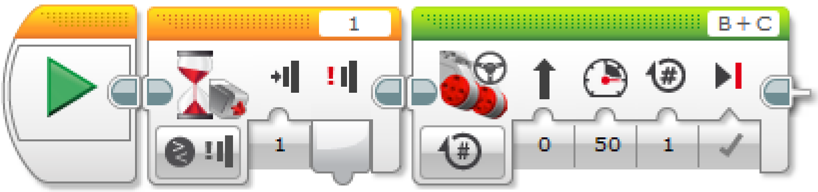
How did you feel when your eyes were closed? If you did or did not reach the door – it was much easier for you compared to a robot since you already knew the plan of the room or in other words – map.
In this exercise you imitated a EV3 robot with touch sensor.

Analog sensors

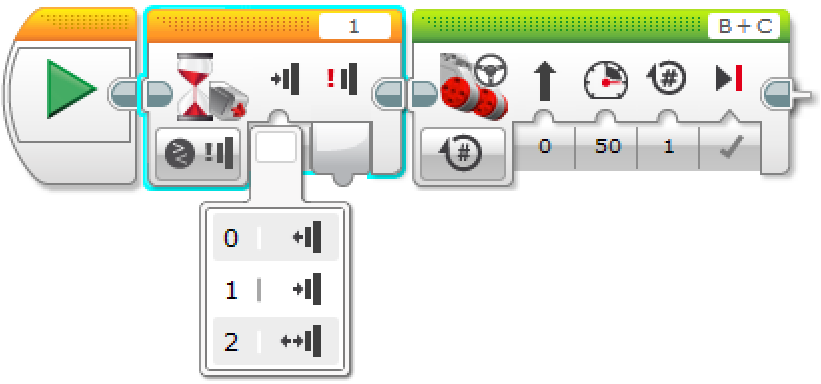
Sensors are divided into two main categories – analog and digital. Analog sensors are the ones that send electrical voltage to the robot controller. Voltage value is directly depending on physical value that the sensor is measuring. This might be temperature, force, light and so on. Analog sensor does not tell the controller that temperature is 30 °C, it will put out 2,4 V for example. Voltage responding to temperature depends on the sensor and its characteristics. Since the microcontroller is a digital device, it cannot be connected with analog sensors without Analog to Digital Converter (ADC). ADC is a device that converts analog sensor's signal into a digital signal, in other words, a sequence of bits. Usually, microcontrollers have a built-in ADC and they need nothing more than an analog sensor to be connected with a suitable leg of microcontroller to start using them.

What is the only analog sensor EV3 robot has? Why do you think so?

1. Make a robot to drive forward when touch sensor is pressed. Use following example:



Second orange block is a wait block (figure above), it means that robot will do nothing besides waiting for that certain condition to be fulfilled. Now robot is just waiting for somebody to press touch sensor – then it continues to execute next block. Only orange color blocks are waiting blocks.

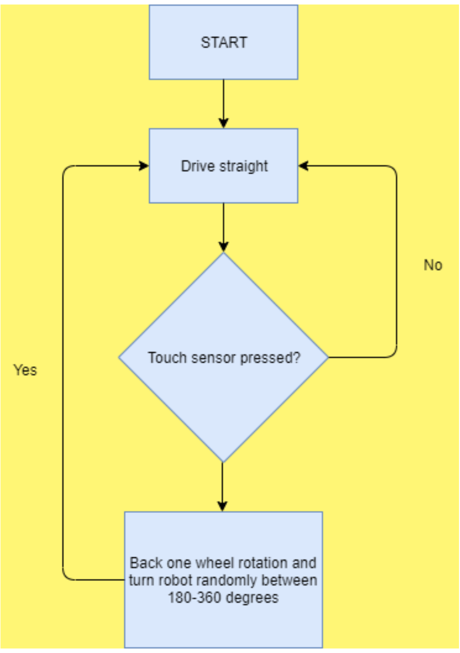


See that touch sensor has three states (figure above):

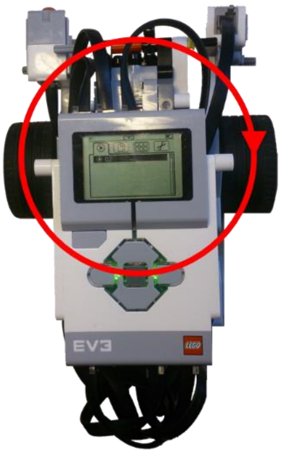
- 0 – react only when button is released
- 1 – react only when button is pressed
- 2 – react only when button is pressed and released

Co-funded by the Erasmus+ Programme of the European Union

2. Program a robot with touch sensor to follow an algorithm:


START
Drive straight
Touch sensor pressed?
No
Yes
Back one wheel rotation and turn robot randomly between 180-360 degrees

For advanced pupils:
Did you make robot wheels turn 180-360 degrees or did you make robot to turn 180-360 degrees?
Correct: **Robot** has to turn this amount of degrees.
Remake your program if needed.
Use formula below to calculate how much robot wheels need to turn for robot to spin 180-360.
C$_{wheel}$ is perimeter of EV3 robot wheel
Δrobot$_{angle}$ is angle of what robot needs to turn
R$_{robot}$ is distance between wheels or points where robot turns (see picture)



$$Motor_{B_{degrees}} = \frac{l}{C_{wheel}} \times 360° = \frac{\Delta robot_{angle} \times 2 \times r_{robot} \times \pi \times 360°}{360° \times C_{wheel}} = \frac{\Delta robot_{angle} \times 2 \times r_{robot} \times \pi}{C_{wheel}}$$

What is minimum amount for wheels to turn for robot to turn 180°? ....
What is maximum amount for wheels to turn for robot to turn 360°? ....

Learning outcomes

Assess together with your partner, how did you reach following learning goals. What is the meaning of:

- Sensors for robots
- Analog sensors
- AD converter

- Waiting block
- Touch sensor (touch sensor states)

Sensors

Lesson 11.-12.

Materials needed:

- LEGO EV3 base robot assembled with sensors

Digital sensors

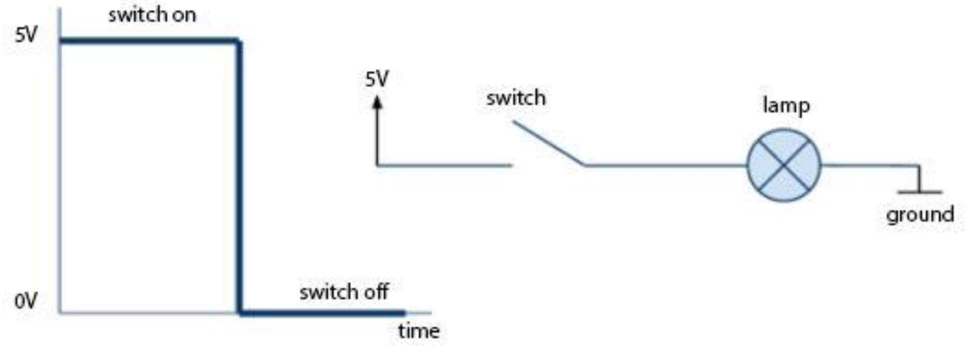
You learned last time what are analog sensors. Now we dive into digital sensors.

Pair up with your friend and try out the following:

5. Close your eyes and try to walk around the classroom or wherever possible
6. Your friend will tell you distances between you and the obstacles in front of you in every 5 seconds
7. See if you can navigate to the door
8. Change roles

Main difference between this exercise and “walk-with-touch-sensor” is that now you get actual reading from the sensor instead of raw value.
In this exercise you imitated a EV3 robot with ultrasonic distance sensor. Robot is getting distance reading from the sensor more than 15 times a second.

Co-funded by the Erasmus+ Programme of the European Union

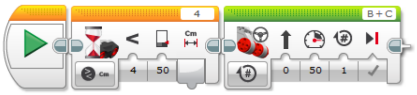


The work principle of a digital sensor.

An elementary digital signal signifies the presence or absence of voltage, in other words, logical 1 or 0. We can use the simple electrical chain (figure above) as an example, since the lamp is connected to the power supply through a switch. In this case, we can conditionally say that the lamp is lit for logical 1 and the lamp is not lit for logical 0.

Ultrasonic sensor

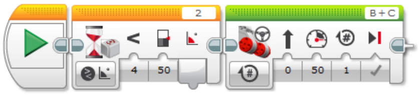
Try out the program example below. This will make the robot wait with ultrasonic sensor until obstacle is closer than 50 cm and then drive forward. Make sure that ultrasonic sensor is attached in port 4.



Can you remake the program so robot drives forward until it detect something closer than 50 cm?

Gyro sensor

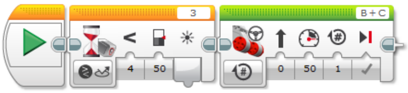
Try out the program example below. This will make the robot wait to be turned on spot at least 50° and then drive forward. Robot uses gyro sensor for that. In case your robot does not work properly, try resetting gyro by removing sensor wire from controller for 2 seconds and then putting it gently back.



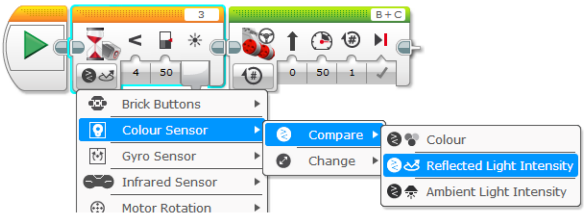
Gyro sensor can also detect speed of angle change degrees/sec.

Light sensor

Try out program example below. Robot will not move until light sensor sees something darker than 50 %. You can try out different surfaces and see which surface makes robot to move.



Can you remake the program, so robot drives forward until light sensor detects dark surface?

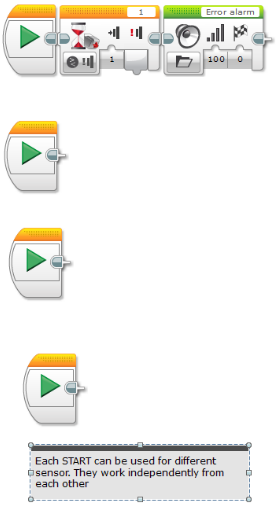


Light sensor has ability to detect colors and ambient light. See from the picture on the left.

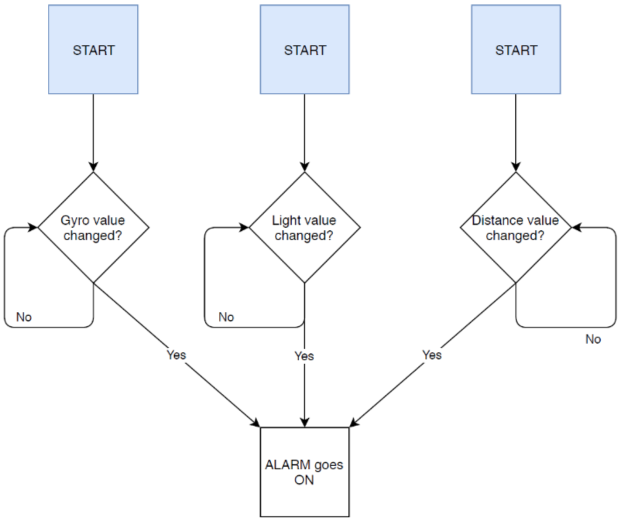
Burglar alarm

When you have something nice, there is always somebody that wants to take it away from you. Your task is to use all possible sensors and methods to make your robot steel proof. Build most advanced robot burglar alarm.

1. Put robot on the table
2. Start the burglar alarm program
3. Robot should make a sound when somebody tries to remove it from the table.



Each START can be used for different sensor. They work independently from each other



Co-funded by the Erasmus+ Programme of the European Union

See the example program (from teacher) and make it to work the way you want. Describe the program in the box below.
You have to use parallel processing in this program. It means that there are programs that work in parallel separately from each other.

When ready, make a competition between other groups and let them try to “steel” your robot. Only rule is that wires of the sensors cannot be removed.

Learning outcomes

Assess together with your friends, how did you reach following learning goals. What is the meaning of:

- Digital sensors
- Parallel processing
- EV3 digital sensor
- Could touch sensor also be considered digital?
- Could button on EV3 controller also to be considered as sensors?

Co-funded by the Erasmus+ Programme of the European Union

Digital sensors

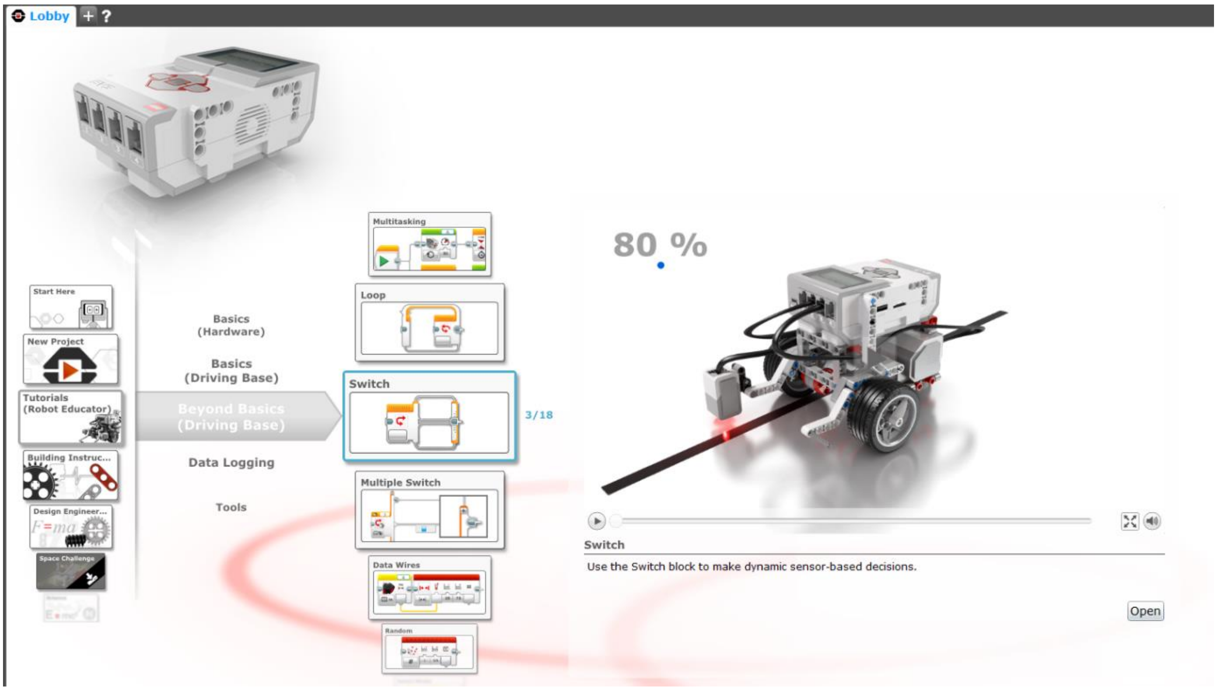
Lesson 13.-14.

Materials needed:

- LEGO EV3 base robot assembled with sensors
- Black tape and light surface

Line following

Line following is one of the main ways to make robots to go where needed. Let's learn the basics by LEGO Mindstorms software:



Switch

Use the Switch block to make dynamic sensor-based decisions.

Open Switch activity and follow instructions. Make the robot to follow a line and see if you can make robot even go faster!

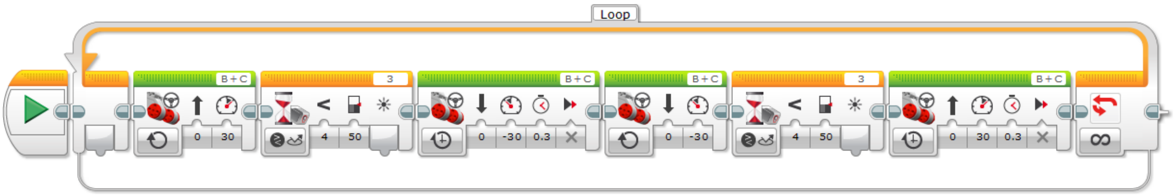
Describe your line following algorithm in the box below.

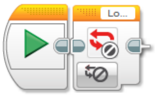
Safe robots

Now that you know how to make a robot to follow a line – can you build a factory robot that delivers goods from one place to another?

1. Robot will take a merchandise from one place (marked with black line) and deliver it to another black line. You can put merchandise on top of robot and take it off by hand.
2. What if there happens to be a person on the way of a robot? Can your robot detect human being and stop and play alarm?

Here you can find sample programm below. But it does not work perfectly. Lower block is called loop interrupt and it stops upper loop. Can you make it work?





For advanced pupils: Variables* – code lock

This tasks if for you if you are motivated to solve more difficult tasks.
Make four digits PIN code

1. EV3 buttons are for PIN Code
   - Up - 1
   - Down - 2
   - Left - 3
   - Right - 4
2. Set the code (for ex. up, up, down, down)
3. We press the code
4. Compare the code with original
5. If it is correct, release the robot

Ask teacher to show you example program!

Learning outcomes

This is the end of sensors part. Assess together with your friends, how did you reach following learning goals.

- Sensors for robots
- Analog sensors
- Digital sensors
- Parallel processing
- EV3 digital sensor
- Could touch sensor also be considered digital?
- Could button on EV3 controller also to be considered as sensors?
- Using variables for programs
- AD converter
- Loop interrupt

Developing STEM competence with robotics

Lesson 15-16: Teambuilding – Algorithmic thinking

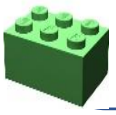
Learning outcomes:

- 21st century skills (esp. communication, collaboration and crea
- Knowledge to the term algorithm and algorithmic thinking.

Materials needed:

Tape or chalk

Paper and pencil



Codes with example

Algorithms and algorithmic thinking

An algorithm is a sequence of commands that is executed in a given order. It can be seen as a detailed recipe. An example of an algorithm could be making a cup of tea. This algorithm would describe the detailed steps of “boil water, find your cup, add a tea bag, pour water into cup until full, steep for x minutes, drink tea from cup.“ When this algorithm has been developed you can repeat it every time you want to drink a cup of tea. Are there any suggestions for other algorithms describing actions in your daily life?

Algorithmic thinking involves breaking bigger complex problems into smaller and more manageable subtasks. It also includes organizing and analyzing data in a logical manner and creating algorithms/recipes to solve complex problems.

Developing algorithmic thinking requires that the students learn to approach problems in a systematic manner. It also requires that the students can suggest solutions that computers using proper programming, are able to help solve.

Algorithmic thinking involves

- the detailed steps that is needed to solve a problem
- use programming skills to make the computer solve the problem

Activity 1: Algorithmic thinking – Obstacle course Alfa

Time estimation: Introduction: 5 mins

Make algorithm: 10 mins

Execute task in front of class: 5 mins

Codes with example

Code	How to use code	Description
New Player	New Player Alex	Alex stand in the A square
Forward	Alex forward	Alex moves one step forward
Back	Alex back	Alex moves one step back
Left	Alex left	Alex standing in the same square, turning left
Right	Alex right	Alex standing in the same square, turning right
Jump	Alex jump	Alex jumps over the square in front of him, landing two squares ahead of him

Steps

- Teacher draws a 5 x 5 grid using tape or chalk on floor or outside, the grid presented here is only an example
- A: Start point, Z: End point, X: dangerous zone
- Put an A and a Z in two opposite corners, then mark five dangerous zones with X
- Describe the grid and the codes above to the students
- Divide the class into pairs
- Give each pair paper and pens
- Make the pairs write down the algorithm for their solution, line by line using the codes and examples
- After 10 mins the teacher choose two pairs that executes their solution in front of the class

A			**X**	
		X		
X				**X**
		X		**Z**

Activity 1: Algorithmic thinking – Obstacle course Bravo

Time estimation: Introduction: 5 mins

Make algorithm: 10 mins

Execute task in front of class: 5 mins

Reflection: 5 mins

In this task, we add another letter, for example S. The class decides in advance what commands this letter represents.

Example:

S = Wave, Clap and Smile

When the students start working with their algorithm taking the player from A to Z, the task now also includes the S letter and the player needs to “Wave, Clap and Smile” each time he or she steps into a S-square.

The teacher decides how many S there will be in the grid.

A	S		X	
		X		S
	S			
X		S	S	X
S		X		Z

Reflection in plenum:

- How did you use algorithmic thinking, solving this task?
- How many unique solutions do you think there is, for the player to reach the endpoint?
- Which experiences from this task do you think is important when programming in pairs later?

Lesson 16: Teambuilding – Building the Cuboid

Learning outcomes:

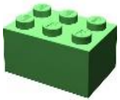
- Working with 21st century skills to improve communication and collaboration abilities.
- Knowledge of the terms driver and navigator, as used in programming.

Materials needed:

LEGO Mindstorms edu EV3 kit (teacher prepares bags with the 34 parts that is needed to build the Cuboid in advance)

Zip-lock bags or similar (one per pair of students)

Cuboid Building instructions:



https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-cuboid-dc93b2e60bed2981e76b3bac9ea04558.pdf

Time estimation

Introduction: 10 min

Task: 20-35 min

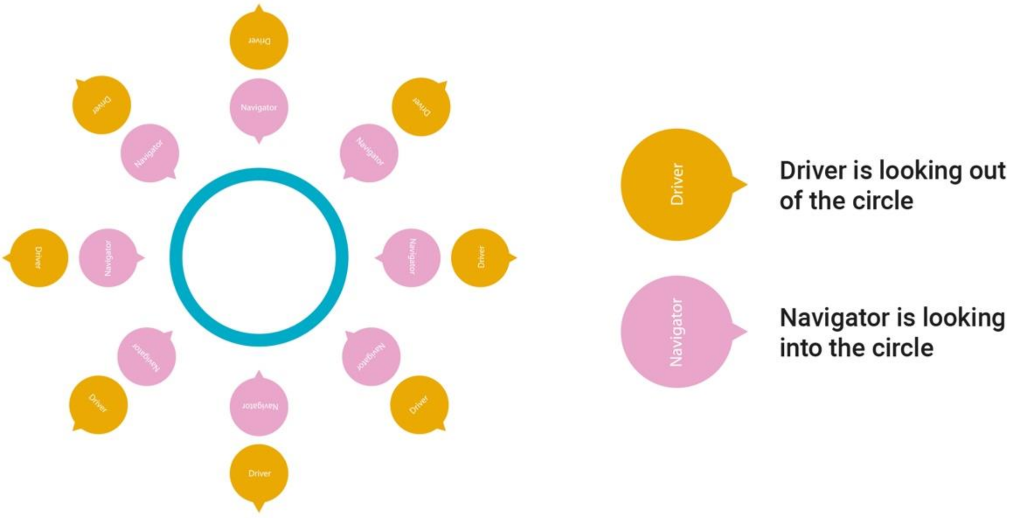
Reflection: 10 min

Concept Introduction - Driver and navigator

Working in pairs with robotics and programming has several advantages, and often leads to good discussions and better programming.

The students will have one of the two roles "*driver*" and "*navigator*" while programming. These two roles are introduced early on to make the cooperation easier and to understand why we use these concepts. Both roles are important to solve this task. The driver will in this task do the building, while the navigator will do the explaining part. In programming, the driver is the programmer while an attentive navigator has the overview and gives the driver advice and ideas.

Co-funded by the Erasmus+ Programme of the European Union


Driver
Navigator
Driver is looking out of the circle
Navigator is looking into the circle

To minimize disturbance, each group should be placed sitting back to back in a big circle and with some distance to each other.

Activity:

- Start by dividing the class into pairs. Each pair will be sitting back to back in a big circle with the builder/driver looking out of the circle
- The navigator will be given a computer and the building instructions above from the teacher
- The driver will be given the 34 LEGO Mindstorms EV3 parts from the teacher
- The navigator explains to the driver how to build the cuboid
- The driver cannot see the building instruction, and is only allowed to use the explanations from the navigator

Tip 1: Students in each pair give suitable names to the LEGO bricks in order to better explain and understand each other.

Tip 2: Leave the Cuboid complete afterwards, as you are going to use it in some of the robotics lessons.

Reflection in plenum:

- What was the hardest part of this task?
- Which strategies was used by the different pairs to solve the task?
- Which experiences from this task do you think is important when programming in pairs later?

Learning outcomes

Reflection Notes:

Assess and reflect together with your teammate, how well did you reach the learning goals? What parts did your team do well while solving this task? How could other parts of the problem-solving process be improved upon next time?

Driving Base

Lesson 17.-18.

Materials needed:

- LEGO Mindstorms Edu EV3 kit
- Video: https://education.lego.com/en-us/lessons/ev3-dep/make-it-move-with-wheels#Planitem0

A robot moves

Use this video https://education.lego.com/en-us/lessons/ev3-dep/make-it-move-with-wheels#Planitem0

Walking robots - Discussion

1. Look at the ways that machines and robots move with and without wheels. Choose one example and trace how the movement transfers from one part to another. *This question challenges students to observe and describe how a robot physically moves.*
2. Compare the way the two walking robots move. What do you notice about how they balance as they walk? *The ZI insect robot lifts one leg on one side but not on the other. The humanoid robot is maintains its balance in several ways (e.g., moving slowly, adjusting foot position, holding hands out and down to the side).*

Wheeled robots - Theory

Wheeled robots are robots that navigate around the ground using motorized wheels to propel themselves. This design is simpler than using treads or legs and by using wheels they are easier to design, build, and program for movement in flat, not-so-rugged terrain. They are also more well controlled than other types of robots. Disadvantages of wheeled robots are that they can not navigate well over obstacles, such as rocky terrain, sharp declines, or areas with low friction. Wheeled robots are most popular among the consumer market, their differential steering provides low cost and simplicity. Robots can have any number of wheels, but three wheels are sufficient for static and dynamic balance. Additional wheels can add to balance;

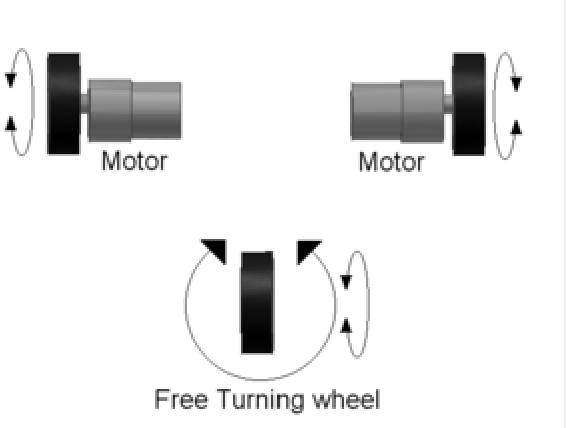
Co-funded by the Erasmus+ Programme of the European Union

however, additional mechanisms will be required to keep all the wheels in the ground, when the terrain is not flat.

Most wheeled robots use differential steering, which uses separately driven wheels for movement. They can change direction by rotating each wheel at a different speed. There may be additional wheels that are not driven by a motor these extra wheels help keep it balanced.

3-wheeled vehicles

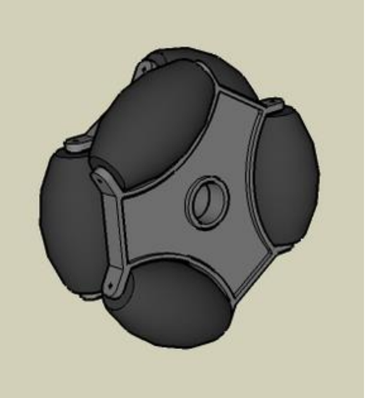
3-wheeled robots may be of two types: differentially steered (2 powered wheels with an additional free rotating wheel to keep the body in balance) or 2 wheels powered by a single source and a powered steering for the third wheel. In the case of differentially steered wheels, the robot direction may be changed by varying the relative rate of rotation of the two separately driven wheels. If both the wheels are driven in the same direction and speed, the robot will go straight. Otherwise, depending on the speed of rotation and its direction, the center of rotation may fall anywhere in the line joining the two wheels.



The center of gravity in this type of robot has to lay inside the triangle formed by the wheels. If too heavy of a mass is mounted to the side of the free rotating wheel, the robot will tip over.

Omni Wheels

Another option for wheeled robots that makes it easier for robots with wheels not all mounted on the same axis to have Omni Wheels. An omni wheel is like many smaller wheels making up a large one, the smaller ones have axis perpendicular to the axis of the core wheel. This allows the wheels to move in two directions, and the ability to move holonomically, which means it can instantaneously move in any direction. Unlike a car, which moves non-holnomicallly and has to be in motion to change heading. Omni-wheeled robots can move in at any angle in any direction, without rotating beforehand. Some omni wheel robots use a triangular platform, with the three wheels spaced at 60 degree angles. Advantages of using 3 wheels and not 4 are that its cheaper, and 3 points are guaranteed to be on the same plane, so each wheel in contact with the ground, but only one wheel will be rotating in the direction of travel. The disadvantages of using Omni wheels is that they have poor efficiency due to not all the wheels rotating in the direction of movement, which also

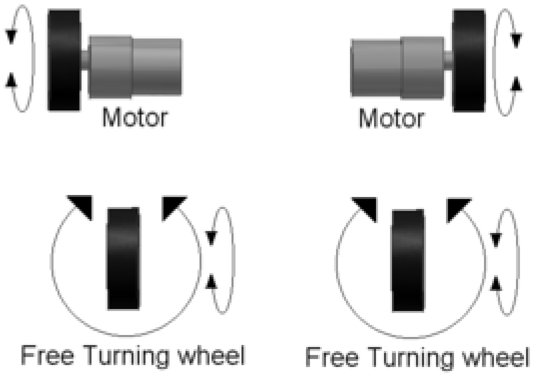


causes loss from friction, and are more computationally complex because of the angle calculations of movement.

4-wheeled vehicles

2 powered, 2 free rotating wheels

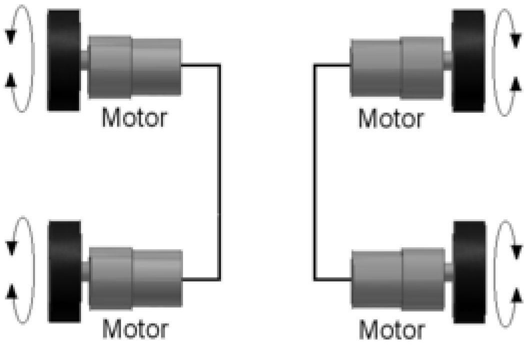
Same as the Differentially steered ones above but with 2 free rotating wheels for extra balance.



More stable than the three wheel version since the center of gravity has to remain inside the rectangle formed by the four wheels instead of a triangle. This leaves a larger useful space. Still it's advisable to keep the center of gravity to the middle of the rectangle as this is the most stable configuration, especially when taking sharp turns or moving over a non-level surface.

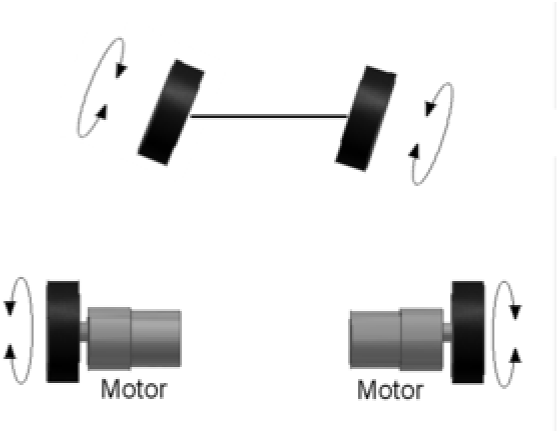
2-by-2 powered wheels for tank-like movement

This kind of robot uses 2 pairs of powered wheels. Each pair (connected by a line) turn in the same direction. The tricky part of this kind of propulsion is getting all the wheels to turn with the same speed. If the wheels in a pair aren't running with the same speed, the slower one will slip (inefficient). If the pairs don't run at the same speed the robot won't be able to drive straight. A good design will have to incorporate some form of car-like steering.



Car-like steering

Differential Steering



This method allows the robot to turn in the same way a car does. This is a far harder method to build and makes dead reckoning much harder as well. This system does have an advantage over previous methods when your robot is powered by a combustion engine: It only needs one motor (and a servo for steering of course). The previous methods would require either 2 motors or a very complicated gearbox, since they require 2 output axles with independent speed and direction of rotation.

Pros and Cons

Type	Pros	Cons
Tracks (Differential Drive)	Good mobility for rough terrain, low ground pressure	Complex mechanism, high power usage when turning
Skid Steer 4 wheel	Very simple, high weight capacity	High power usage, requires skidding/slipping
Differential drive 2 wheel + Passive Caster(s)	Easy	Lower weight designs, less precise controls. bad for obstacles/bumps
2 wheel + 1 Powered Steering Caster	Easy mechanical, powered steering wheel for control	Lower weight designs, potentially a lot of weight on steering wheel.
Ackerman Steering	Wheels do not need to slip to turn. Fixed rear wheels makes control geometry easier.	Increased motor count

Omni	Non-holonomic, light, simple, Best performance when traveling diagonal, number of wheels can be varied	Motion can be bumpy, sensitive to nonsmooth terrain, low torque for pushing
Legs (2,4,6,etc..)	Ease of steeping over difficult terrain	Hard, complex, high power

Design, build and program

Activity no1:
Design, build, and program a robot that can:

- Move itself a distance of 1 m
- Use at least one motor
- Use wheels for locomotion
- Display the distance it moved

Building Ideas:

- Large Motor and Wheel
- Bevel Gears

If a team needs more help use the instructions of the Driving Base by Lego Education:

- Driving Base

Investigate different ways of Controlling Driving Base moving in a straight line.

- Program 1 : Use the Move Steering Block to program your robot move a distance of 1 m.The diameter of the standard wheel is 56mm.

- Program 2: Make the Driving Base do the exact same movement using the Move Tank Block.

What difference do you notice ? Note this below

- Program 3: Display the distance it moved.

Co-funded by the Erasmus+ Programme of the European Union

Activity no2:
Your robot needs to move a distance of 1 meter on a rough surface. What changes would you make to your robot build and why?
Building Ideas:

- Tracks

Learning outcomes

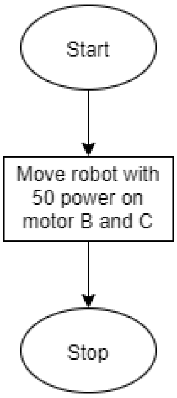
Assess together with your partner, how did you reach following learning goals:

- describes different movement mechanisms and their characteristics.
- display the distance that robot moves.
- select the driving mechanism for their robot depending on the terrain.

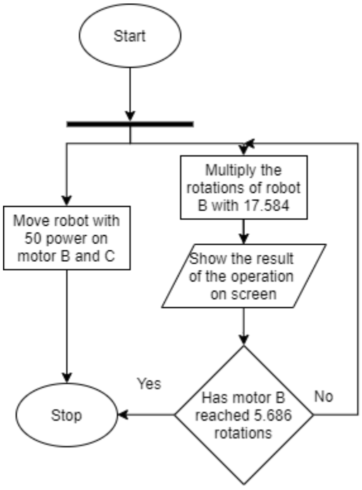
Suggested UML Solutions

Activity no1:

Program 1 , Program 2



Program 3


Start
Move robot with 50 power on motor B and C
Multiply the rotations of robot B with 17.584
Show the result of the operation on screen
Has motor B reached 5.686 rotations
Yes
No
Stop

Driving Base

Lesson 19.-20.

Materials needed:

- LEGO Mindstorms Edu EV3 kit

A self-driving cars moves

A self-driving car, also known as an autonomous car, driverless car, or robotic car, is a vehicle that is capable of sensing its environment and moving safely with little or no human input

Self-driving cars combine a variety of sensors to perceive their surroundings, such as radar, lidar, sonar, GPS, odometry and inertial measurement units.Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles and relevant signage.

Long distance trucks are seen as being in the forefront of adopting and implementing the technology

So how do driverless cars work?

There are several systems that work in conjunction with each other to control a driverless car.

- Radar sensors dotted around the car monitor the position of vehicles nearby.
- Video cameras detect traffic lights, read road signs and keep track of other vehicles, while also looking out for pedestrians and other obstacles.
- Lidar sensors help to detect the edges of roads and identify lane markings by bouncing pulses of light off the car’s surroundings.
- Ultrasonic sensors in the wheels can detect the position of kerbs and other vehicles when parking.
- Finally, a central computer analyses all of the data from the various sensors to manipulate the steering, acceleration and braking.

Programming

For the next activities use the robots made by the students in the previous workshop.

Activity no1:
Program your robot :

- to execute a square path (1mx1m) using different parameters (rotations, degrees, seconds) into programming blocks.

Try to guess the algorithm of the program. Write a description of the program

Activity no2 :
Each team is responsible for programming a robotic car that will move on a specific route on the streets of a city. (Streets of a city are designed from the teacher. Suggest that the route has a rough surface too, so that children can use the robot with tracks.)

Try to guess the algorithm of the program. Write a descripption of steps of the program.

Learning outcomes

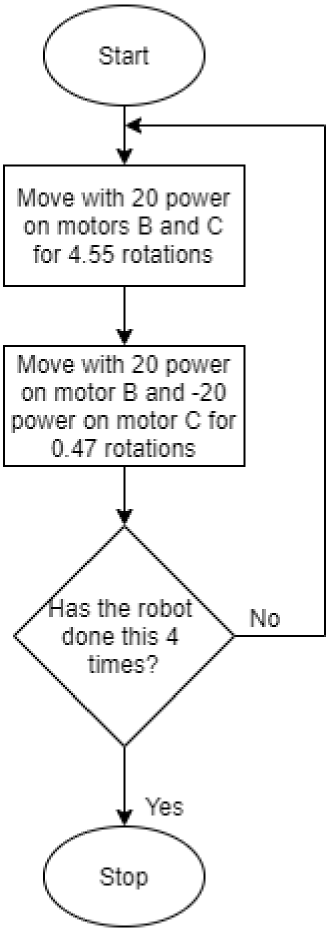
Assess together with your partner, how did you reach following learning goals:

- describe how self-driving cars move.
- program a robot for driving a specific route.

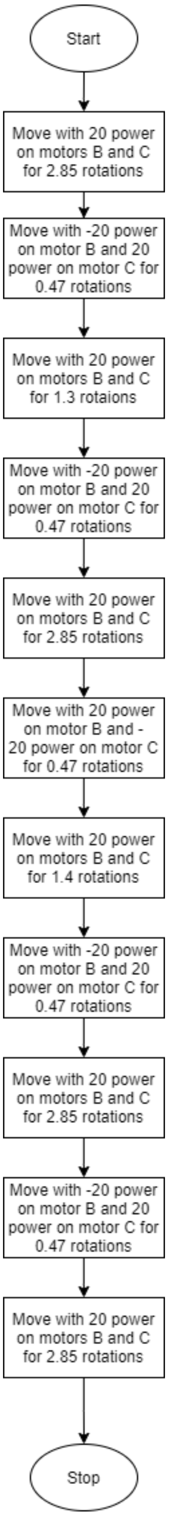
Co-funded by the Erasmus+ Programme of the European Union

Suggested UML Solutions

Activity no1:


Start
Move with 20 power on motors B and C for 4.55 rotations
Move with 20 power on motor B and -20 power on motor C for 0.47 rotations
Has the robot done this 4 times?
No
Yes
Stop

Activity no2:


Start
Move with 20 power on motors B and C for 2.85 rotations
Move with -20 power on motor B and 20 power on motor C for 0.47 rotations
Move with 20 power on motors B and C for 1.3 rotaions
Move with -20 power on motor B and 20 power on motor C for 0.47 rotations
Move with 20 power on motors B and C for 2.85 rotations
Move with 20 power on motor B and -20 power on motor C for 0.47 rotations
Move with 20 power on motors B and C for 1.4 rotations
Move with -20 power on motor B and 20 power on motor C for 0.47 rotations
Move with 20 power on motors B and C for 2.85 rotations
Move with -20 power on motor B and 20 power on motor C for 0.47 rotations
Move with 20 power on motors B and C for 2.85 rotations
Stop

Co-funded by the Erasmus+ Programme of the European Union

Critical thinking on the robot field – choosing missions based on time and points

Lesson 21.-22.

Learning outcomes:

- 21st century skills (in particular critical thinking)
- Creating problem solving algorithms

Materials Needed:

LEGO EV3 base robot assembled with sensors

Training mat (or marked floor space)

Assembled cuboid

Time estimation:	Introduction: 10 mins
	Task: 55 mins
	Challenge: approx. 15 mins (3 minutes per team)
	Reflection: 10 mins

Airport scenario

HeliForce Airport recently bought a new robot to solve some of the routine tasks and some of the dangerous tasks that take place at the airport every day. Before trusting the robot to do its tasks, it is important to test it and be sure it is able to complete them by itself.

Level	Points
1a	200
1b	200
1c	200
2	300
3	500
4	700 + 100

Levels and points

There are four levels with different kinds of tasks to execute. Level 1 contains less tasks than level 2. Each level gives different score points (see table), if the task is executed properly and as expected. If the task is not completed, no points are given.

Time

After doing the programming part, each team are given maximum of 3 minutes to complete all the tasks. The teams have to make a plan for which tasks their robot are going to solve first and last, and the time spent will also give a certain amount of points. If a team fails to program for all four levels during the programming time, the team will still have the chance to test their robot and get points for the tasks they complete.

It is not necessary to complete all tasks on level 1 to get maximum time score.

Team solved all tasks (time)		Team solved some of the tasks (time)	
Under 3 mins	200	Under 3 mins	100
Under 2 mins	400	Under 2 mins	200
Under 1.5 mins	600	Under 1.5 mins	300

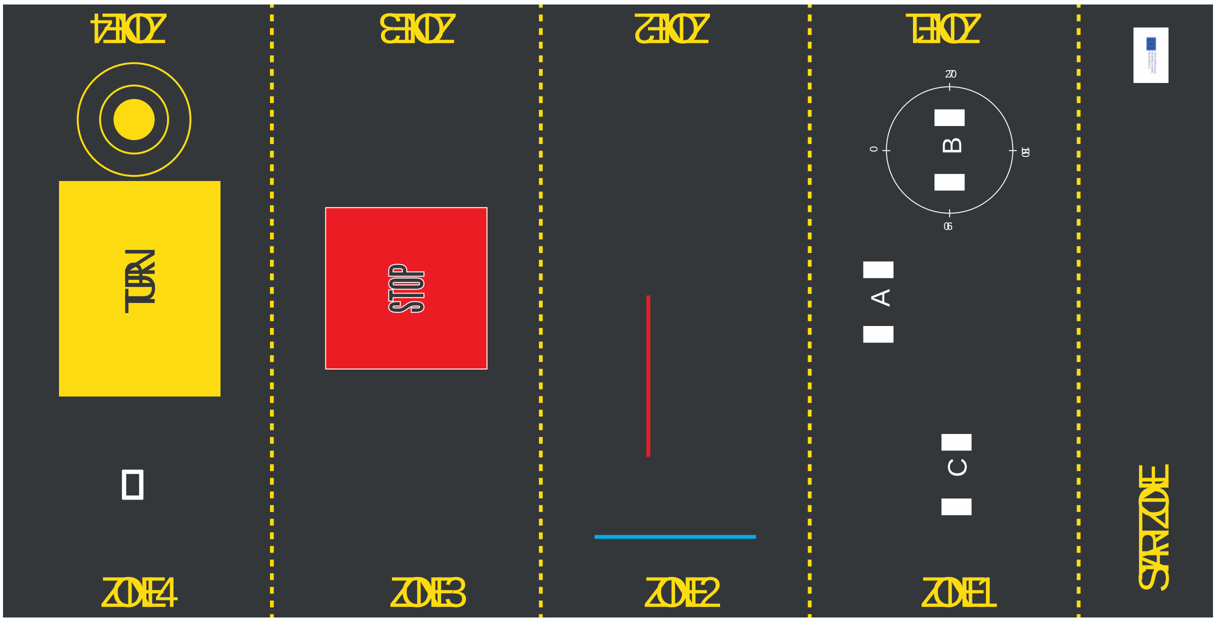


Illustration 1: Airport scenario traning-mat

Level 1: Test precision A

The robot starts in the START ZONE.

a) Drive forward to announced track in ZONE 1.
b) Stop
c) Play sound STOP
d) Return to START ZONE

All tasks have to be completed to get points.

Level 1: Test precision B

The robot starts in the START ZONE.

a) Drive forward to marked circle area in ZONE 1
b) Stop
c) Play sound TURN LEFT
d) Turn 90 degrees
e) Stop
f) Return to START ZONE

All tasks have to be completed to get points.

Level 1: Test precision C

The robot starts in the START ZONE.

a) Drive forward to marked area C in ZONE 1
b) Stop
c) Play sound STOP
d) Return to START ZONE

All tasks have to be completed to get points.

Level 2: Test color sensor

The robot starts in the START ZONE with color sensor facing down.

a) Drive forward until red color appears in zone 2
b) Stop
c) Make alarm sound
d) Turn 180 degrees
e) Return to START ZONE

All tasks have to be completed to get points

Level 3: Test touch sensor

The robot starts in the START ZONE with touch sensor facing forward.

a) Drive forward until reached area marked STOP in zone 3
b) Turn around 90 degrees
c) Drive forward until reached 10 cm tall and stable object (can be LEGO bricks, book etc) and the touch sensor is pressed
d) Play sound “Finished”
e) Show on display: “Touch sensor pressed”
f) Return to START ZONE

All tasks have to be completed to get points.

Level 4: Test medium motor and ultrasonic sensor

The robot starts in the START ZONE with ultrasonic sensor facing forward.

a) Drive forward until turnaround area in zone 4
b) Turn around 90 degrees
c) Use ultrasonic sensor to stop at object CUBOID left on the runway
d) Use medium motor and grabber to take a hold of CUBOID
e) Turn around 180 degrees
f) Drive forward and put object in SAFE ZONE, left in the middle of the circle gives an extra 100 points
g) Show on display text: All tests finished
h) Return to START ZONE

All tasks have to be completed to get points.

Co-funded by the Erasmus+ Programme of the European Union

Critical points before starting programming

Here are some questions to ask yourself while planning what missions to choose when you start working with the Airport challenge:

- Is it possible to distribute tasks between team members to get the tasks done in an efficient way?
- What tasks give the most points compared to the time it takes to solve them?
- How much time can you use to program for each task given the time available, will it be enough?
- Should you program your robot to do several missions in one run, or lower the risk by doing one or two in each run?

Score sheet for airport scenario

Team:	**Points**	**Points given**	**Time spent tasks all levels /points**	**Time spent some tasks / points**
Level 1a	200		Under 3 = 200 Under 2 = 400 Under 1,5 = 600	Under 3 = 100 Under 2 = 200 Under 1,5 = 300
Level 1b	200			
Level 1c	200			
Level 2	300			
Level 3	500			
Level 4	700 + 100			
TOTAL:				

Learning outcomes

Reflect and think about how well you met the learning goals and what points of reflection that is worth talking about after this task?

Evaluate what to keep, what to discard and what to improve if you were to be given a similar challenge?

Notes:

Data Communication

Lesson 23.-24.

Materials needed:

- LEGO Mindstorms Edu EV3 kits (at least 3)
- Pairs of LEGO Mindstorms Edu EV3 base robot assembled without sensors

Data Communication

Data communication is the active process of transporting data from one point to another. Data communication is said to be local if communicating devices are in the same building or a similarly restricted geographical area.

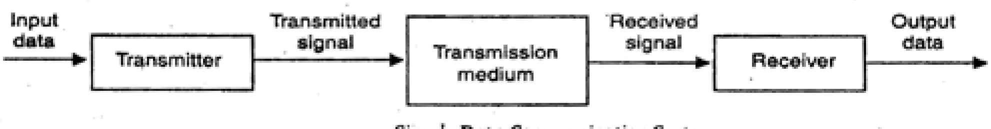
The meanings of source and receiver are very simple. The device that transmits the data is known as source and the device that receives the transmitted data is known as receiver. Data communication aims at the transfer of data and maintenance of the data during the process but not the actual generation of the information at the source and receiver.

Principles and Charateristics

The effectiveness of a data communication system depends on three fundamental characteristics:

- **Delivery**: The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.
- **Accuracy**: The system must deliver data accurately. Data that have been altered in transmission and left uncorrected are unusable.
- **Timeliness**: The system must deliver data in a timely manner. Data delivered late are useless. In the case of video, audio and voice data, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called real-time transmission.
- **Jitter**: It is the uneven delay in the packet arrival time that cause uneven quality.

Data can exist in a variety of forms such as numbers, text, bits and bytes. The Figure is an illustration of a simple data communication system.



Simple Data Communication System

The term data used to describe information, under whatever form of words you will be using.

Components of data communication system

A Communication system has following components:

1. **Message**: It is the information or data to be communicated. It can consist of text, numbers, pictures, sound or video or any combination of these.

2. **Sender**: It is the device/computer that generates and sends that message.

3. **Receiver**: It is the device or computer that receives the message. The location of receiver computer is generally different from the sender computer. The distance between sender and receiver depends upon the types of network used in between.

4. **Medium**: It is the channel or physical path through which the message is carried from sender to the receiver. The medium can be wired like twisted pair wire, coaxial cable, fiber-optic cable or wireless like laser, radio waves, and microwaves.

5. **Protocol**: It is a set of rules that govern the communication between the devices. Both sender and receiver follow same protocols to communicate with each other.

Most common basic data communication methods

Whether you want to transfer extremely important files, like funny memes or the latest movies, to your friend’s phone, or change the channel on your TV without having to pull yourself off the comfy couch, technology is always there to make things easier for you.

Bluetooth

In today’s world of smartphones and tablets, Bluetooth has become synonymous with the wireless sharing of data. It’s also used for a wireless communication between a phone and a headset, streaming audio in Bluetooth-enabled speakers, local connectivity between computers, and plenty of other applications that require wireless data transmission.

Bluetooth is a wireless technology standard that uses short-wavelength UHF radio waves (2.4 GHz to 2.485 GHz) to transmit and receive data over short distances (typically within the range of 10 meters).

If you have a smartphone, then you likely know about Bluetooth and what it does. It helps you wirelessly connect with other devices in the vicinity, such as your headphones, headset or someone else's phone, and also exchange data.

The advantages of Bluetooth include less power consumption, no requirement for additional hardware (other than the devices that need to be connected) and simplicity of use.

Bluetooth does not come with any such 'positional' constraints. You can be carrying the phone anywhere within the transmission range and you should be able to transfer photos to your computer without any problems. In fact, it works even when you're in a different room altogether – separated by walls!

Infrared Transmission

Infrared transmission derives its name from infrared light rays. Infrared rays have a wavelength that is greater than visible light, extending from the nominal red edge of the visible spectrum at **700 nanometer to 1 mm**.

The infrared rays that allow you to change the channel from a considerable distance. Infrared signals use pulses of infrared rays to connect two devices locally and exchange information between them. These are most used for short-range or medium-range communications between two devices. The most common example of infrared communication is your TV remote control; infrared rays are what allow you to change channels, increase/decrease volume, and perform plenty of other 'remote' operations without having to leave the couch.

Daisy chain

In electrical and electronic engineering a **daisy chain** is a wiring scheme in which multiple devices are wired together in sequence or in a ring. Other examples of devices which can be used to form **daisy chains** are those based on USB, FireWire, Thunderbolt and Ethernet **cables**.

How an EV3 robot communicate with other devices?

Your EV3 provides three types of communication, bluetooth, wifi and USB. Bluetooth and USB can be used without any additional equipment, the communication via wifi needs a dongle.

Brick to Brick Communication

EV3 Bricks can communicate with each othea in two ways. The first is by USB cable called Daisy- Chaining, and the secon is via Bluetooth. Both methods have their own characteristics.

Daisy – Chaining

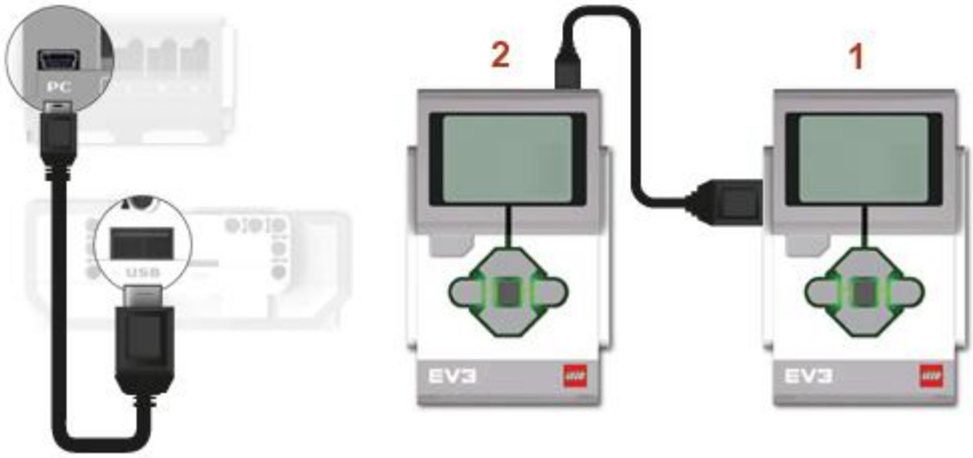
Daisy – Chaining is used to connect up to four EV3 Bricks together usin USB cables. What is unique about this system is one program controls up to four bricks. This means that complicated systems can e developed with just one program. If you want a design that has sixteen motors and sixteen sensors, you can! You can then use the first EV3 Brick in the chain to control the motors and read the sensors of the daisy chained EV3 Bricks.

Activities

Select the Daisy Chaining check box on the Project Properties Page to enable Daisy Chaining.

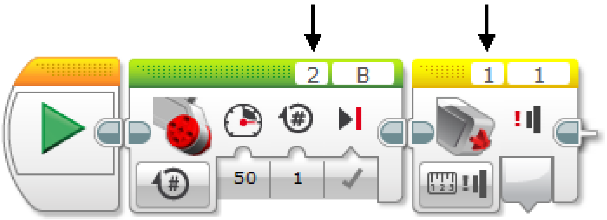
DAISY CHAINING THE EV3 BRICKS

The USB Port on the side of the first EV3 is connected to the mini USB Port of the next EV3 Brick using an appropriate USB cable. The PC Port of the next EV3 Brick in the chain is connected to the USB Port of the previous EV3 Brick, using an appropriate USB cable.



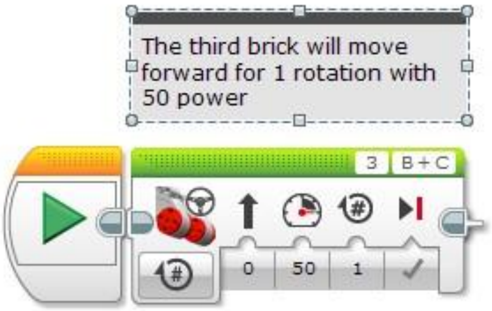
USING THE LAYER SELECTOR

1. When Daisy Chaining is enabled, each Motor Block and Sensor Block is modified to include a Layer Selector.

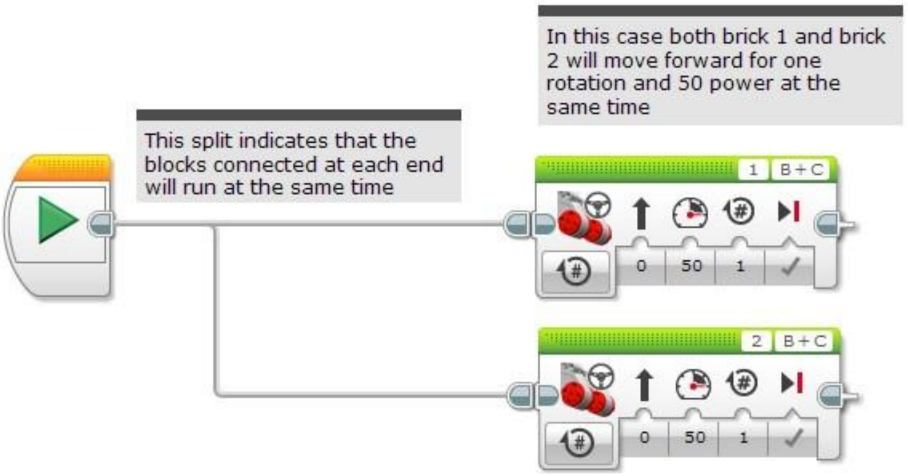


Use the Layer Selector to choose which EV3 Brick the programming block will run on.

2. Program the third EV3 Brick drive forward.


The third brick will move forward for 1 rotation with 50 power

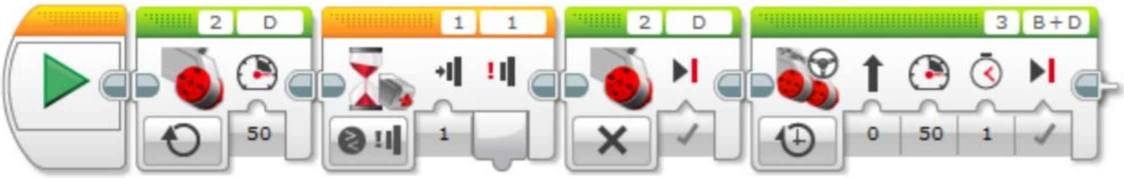
3. Program the first EV3 robot and the second EV3 robot drive forward at the same time.



In this case both brick 1 and brick 2 will move forward for one rotation and 50 power at the same time

This split indicates that the blocks connected at each end will run at the same time

4. Transfer the program to your computer. Connect the first brick to a touch sensor, to the second engine and to the third two engines.



Describe what you think the program below will do

Activity
Pair your base EV3 robots of another team’s brick. Choose your favourite song. Create a small dancefloor with simple movements so that the two robots move in sync.

Draw the moves with arrows below.

Present the dance to other teams.

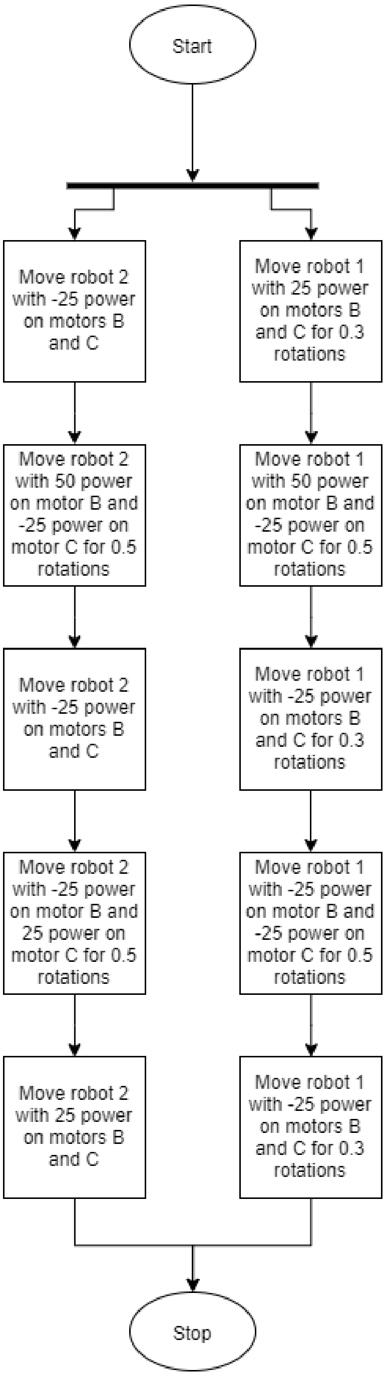
Learning outcomes

Assess together with your partner, how did you reach following learning goals:

- Describe digital communication, its principles and characteristics.
- Draw diagram of the principle of data communication.
- After the activity students will be able to synchronize 2 ev3 bricks for a common goal.

Suggested UML Solutions

Activity


Start
Move robot 2 with -25 power on motors B and C
Move robot 1 with 25 power on motors B and C for 0.3 rotations
Move robot 2 with 50 power on motor B and -25 power on motor C for 0.5 rotations
Move robot 1 with 50 power on motor B and -25 power on motor C for 0.5 rotations
Move robot 2 with -25 power on motors B and C
Move robot 1 with -25 power on motors B and C for 0.3 rotations
Move robot 2 with -25 power on motor B and 25 power on motor C for 0.5 rotations
Move robot 1 with -25 power on motor B and -25 power on motor C for 0.5 rotations
Move robot 2 with 25 power on motors B and C
Move robot 1 with -25 power on motors B and C for 0.3 rotations
Stop

Data Communication

Lesson 25.-26.

Materials needed:

- Pairs of LEGO Mindstorms Edu EV3 base robot assembled without sensors

How an EV3 robot communicate with other devices?

Your EV3 provides three types of communication, bluetooth, wifi and USB. Bluetooth and USB can be used without any additional equipment, the communication via wifi needs a dongle.

The EV3 uses Bluetooth:

- to download programs wirelessly
- to connect to another EV3
- to controlling a large single robot with multiple bricks
- for an alternative to daisy chaining: Requires less wiring but needs separate programs
- to coordinating multiple robots : Have them perform similar actions - Synchronize their actions

Brick to Brick Communication

EV3 Bricks can communicate with each other in two ways. The first is by USB cable called Daisy- Chaining, and the second is via Bluetooth. Both methods have their own characteristics.

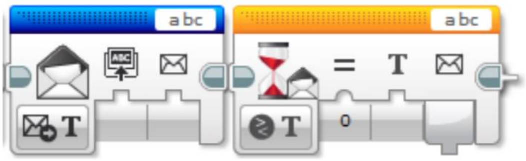
Via Bluetooth

Up to seven EV3 Bricks can communicate with each other. They each need to be programmed separately with the EV3 Software.

There are two big differences between Daisy-Chaining and Bluetooth Communication. The first is the connection: Daisy-Chaining is physical and Bluetooth is not. The second is that Bluetooth communication requires EV3 bricks to be paired and individual programs to be run on them.

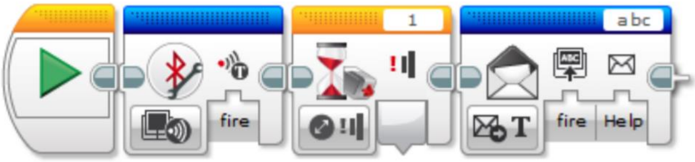
Co-funded by the Erasmus+ Programme of the European Union

Below shows how the first block sends the message on EV3 Brick no1 and the second block shows how the block waits for the message on EV3 Brick no2.

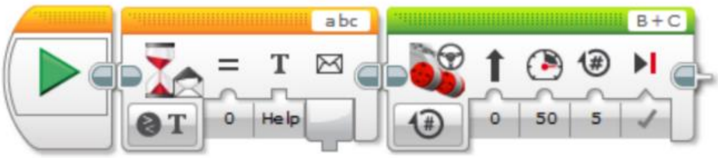


A great example of using Bluetooth cold be an alarm system. One EV3 Brick acts as the panic switch sending Bluetooth messages to the police, ambulance, or fire engines separately. See the example below:

The program block below is the Master EV3 Brick, which initializes itself with the other brick. In this case, the brick is called „fire" as in fire station. When the Touch Sensor is pressed, a „help" message is sent to the „fire" brick.



The second program is the EV3 Brick called „fire". It is waiting for a message called „help". When this is received, the fire engine drives out to attend the fire.



Remember, with Bluetooth communication you need a program for each EV3 Brick you use!

Activities

Tips of success:

Each team needs to:

- Give each brick an unique name (Sender and Receiver for this lesson)
- Use an USB cable for downloading code to the bricks (rather than Bluetooth). You cannot connect between EV3s and a PC at the same time
- Turn on the receiver robot first then the sender in the activities in this lesson. This is because the activity solution only sends the message once in Activity 1.

Activate Bluetooth on the EV3:

Turning on Bluetooth
On the EV3 menu, go to the symbol that looks like a wrench
Go down to the word "Bluetooth" and activate it
To connect to another brick go back into the Bluetooth menu
Go to "Connections"
Select "Search" and find the other brick's name

Activity no1 Messaging:

Send a "Hello World" message from one brick to another. The second brick should receive and display the message on its screen for 5 seconds.

Tips:

You will need to pair the two robots ahead of time

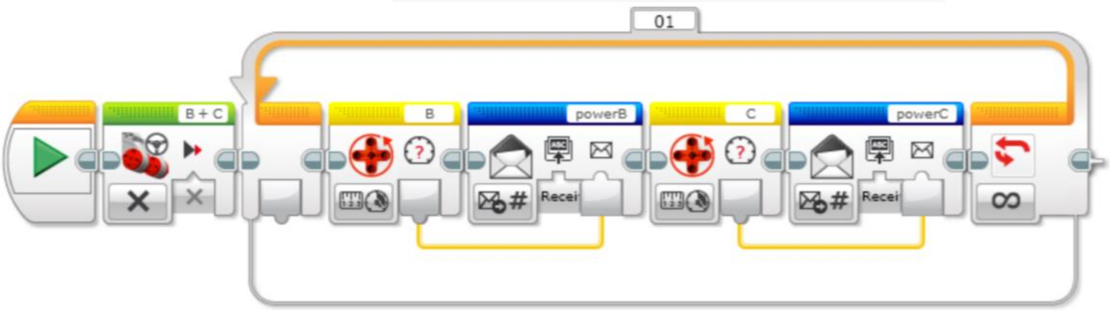
Name one robot Sender and the other Receiver for this challenge

For this challenge, you will use the Messaging block -> Send -> Text

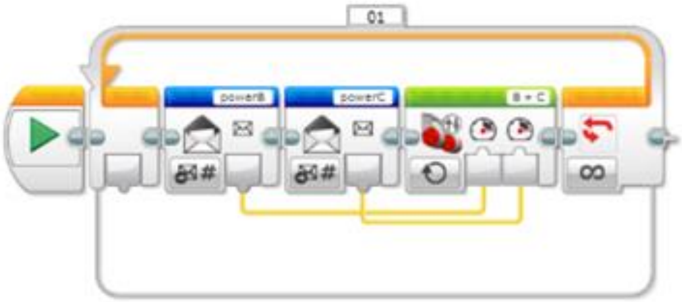
You will use the Wait For block in the Messaging Mode -> Change -> Text

Activity no2: Synchronized Swimmers

Code on the Sender Robot



Code on the Receiver Robot



Make the above programs on your computer.

Move the first robot by hand and send the motor’s power to the other robot.

What do you notice? Why is this happening;

Activity no3: Door Bell

Build a robotic system with two Bricks to use for a door bell. One of the Bricks will have a touch sensor that when pressed will sound the bell from the other Bricks.

Learning outcomes

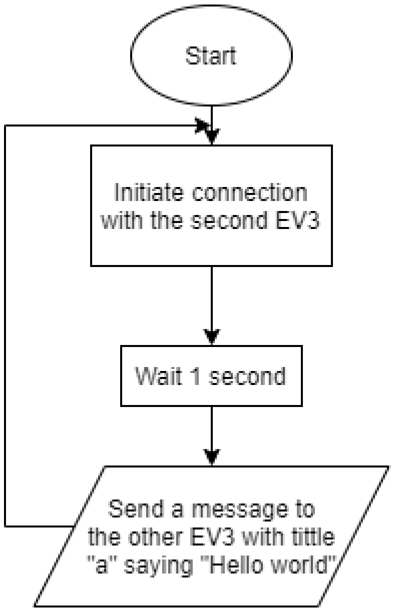
Assess together with your partner, how did you reach following learning goals:

- Select the appropriate data communication solution for your robotics system
- Send messages from one brick to another
- Use Bluetooth connection for connect one brick to another

Suggested UML Solutions

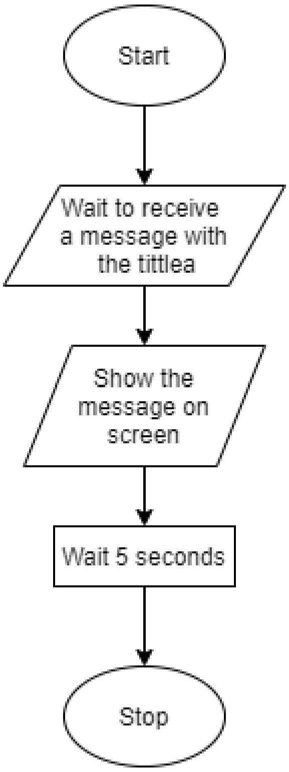
Activity no1 Messaging:

Sender



Start
Initiate connection with the second EV3
Wait 1 second
Send a message to the other EV3 with tittle "a" saying "Hello world"

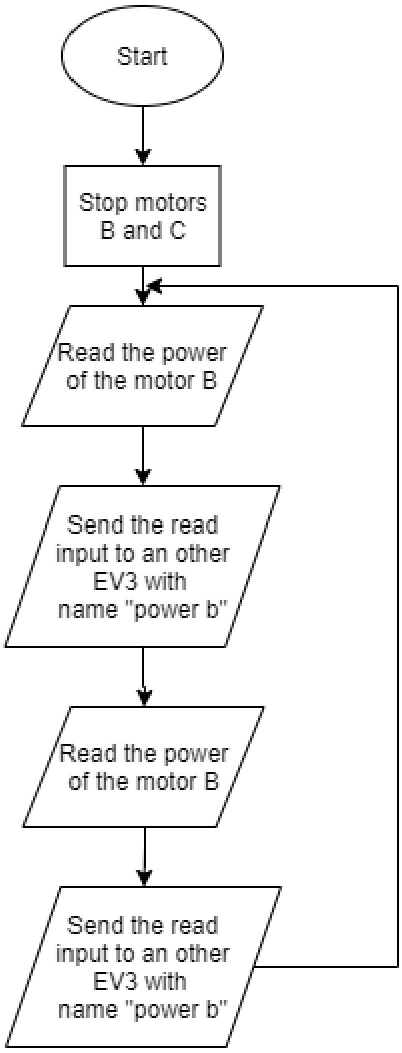
Receiver



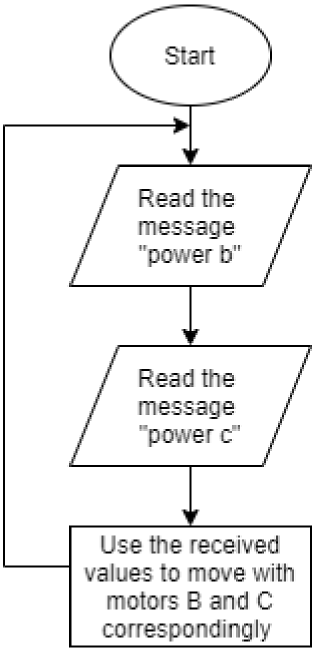
Start
Wait to receive a message with the tittlea
Show the message on screen
Wait 5 seconds
Stop

Activity no2: Synchronized Swimmers

Sender

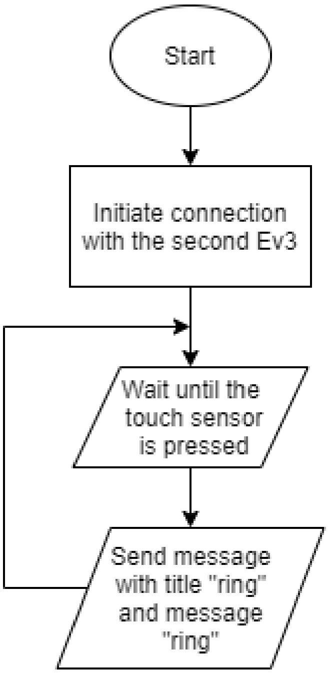

Start
Stop motors B and C
Read the power of the motor B
Send the read input to an other EV3 with name "power b"
Read the power of the motor B
Send the read input to an other EV3 with name "power b"

Receiver

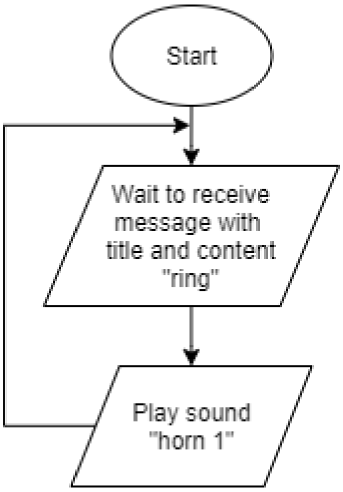

Start
Read the message "power b"
Read the message "power c"
Use the received values to move with motors B and C correspondingly

Activity no3: Door Bell

Sender


Start
Initiate connection with the second Ev3
Wait until the touch sensor is pressed
Send message with title "ring" and message "ring"

Receiver



Data logging and processing

Lesson 27.-28.

Materials needed:

- LEGO Mindstorms Edu EV3 base robot assembled without sensors
- LEGO Mindstorms Edu EV3 Velocity (instuctions included)

Data Logging

What does *Data Logging* mean?

Data logging is the process of collecting and storing data over a period of time in order to analyze specific trends or record the data-based events/actions of a system, network or IT environment. It enables the tracking of all interactions through which data, files or applications are stored, accessed or modified on a storage device or application.

Data logging enables the recording of activity performed on one or more data/file objects or sets. Typically data logging records events/actions, such as the data's size, most recent modification and username/name of the individual that modified the data.

Data logging also facilitates the storage and collection of computer or device information. For example, data logging can store processor temperature and memory utilization over time and network bandwidth usage. System/network administrators use this data to analyze system or network performance during a specific period.

Data logging also allows information security and auditing staff to analyze system access information and assess audit trails to trace viruses and identify suspicious activities.

Do we collect data in life?

Here are five ways it's used in your daily life.

1. Mobile maps and GPS. Remember reading maps to take a road trip?
2. Online Shopping.
3. Urban Planning.

4. Energy Consumption.
5. Wearables.

How is weather data collected?

The weather of an area is due to four factors. They are heat energy, air pressure, winds, and moisture. To make weather maps and forecast weather, weather data must be collected from Earth's surface and atmosphere. Scientists analyze the weather maps and the data in them to forecast the weather.

They are used to measure the size and motion of clouds and precipitation. Satellites designed for weather data collection typically contain cameras and radar systems. They're made of latex and are most often used to measure pressure, temperature, humidity, and wind speed.

A weather station is a device that collects data related to the weather and environment using many different sensors. Weather stations sensors may include a thermometer to take temperature readings, a barometer to measure the pressure in the atmosphere, as well as other sensors to measure rain, wind, humidity and more.

What is EV3 Data Logging?

The EV3 software provides a simple way to continuously record sensor readings to a file and to plot the values later. This is called Data Logging.

Why use Data Logging:

- Great for science experiments. You can record values from a sensor for a science project.
- Great for understanding robot programming blocks. You can use data logging to measure the difference between turns.
- Great for understanding sensor behavior. You can use data logging to understand the details of sensors such as the gyro sensor.

Most common basic data logging methods

There are 4 ways to data log using the EV3 MINDSTORMS:

1.Live Data Logging: Real time data collected directly in the EV3 software

2. Remote Data Logging: Use the the brick to collect data, and transfer the data to the computer for analysis

3. Brick Data Logging: Run the experiment directly from the brick

4. Autonomous. Collect data with the Data Logging block. The data is stored on the brick.

Learning outcomes

Student is able to understand basic principles about logging and collecting data, understand graphs, analyze these and make conclusions.

Activities

Activity no1 Comparing Turns

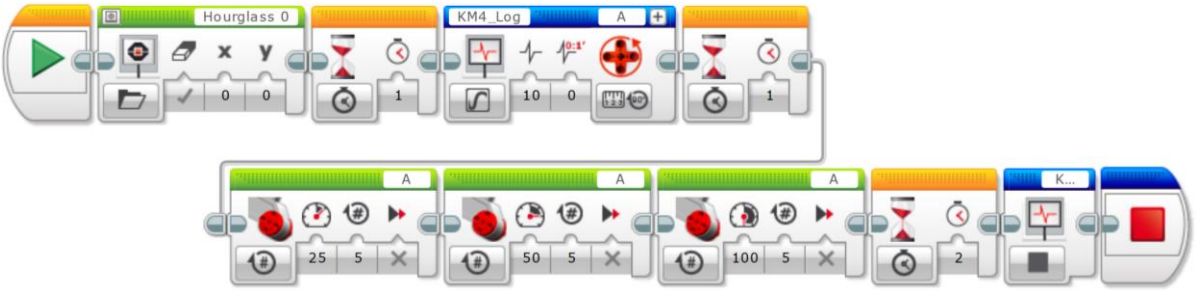
Make a program that measure the different types of turns the robots can do and compare the data from the rotation sensor. Ask for teacher's help if needed.

Follow this steps.

- In the Data Logging My Block, select the sensor you are reading, the ports they are in.
- Select the duration and rate
- Remember to stop data logging at the end of your code
- Import your data file and compare the graphs.
  Which type of turn is the most reliable?

Activity no2 Velocity

The following experiment involves the creation of motion charts indication various values such as time, distance , velocity, and acceleration. Relationships can be clearly illustrated and evaluated using logged data.



Make the above program to the software.Download the program to the EV3 Brick. What do you think the robot will do?

- Start Program 2.
- Measured values are recorded automatically for subsequent evaluation.
- Connect the EV3 Brick. Upload the data from the EV3 to the software.
- Create and interpret a graph illustrating the distance travelled over time. Use the datasets from the motor to create a graph. The dataset is called „Rotation_pA_01“
- Create a graph using the calculated data: Velocity, slope of the distance: Time. This graph is based on the Distance dataset crated previously. Name the new dataset Velocity.

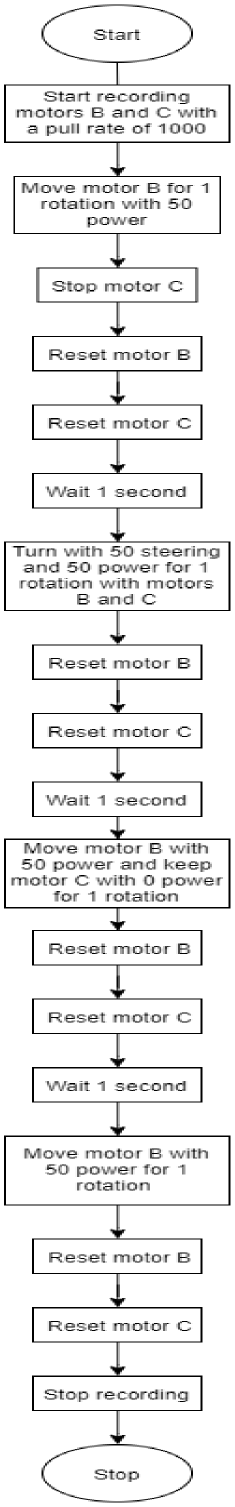
Learning outcomes

Assess together with your partner, how did you reach following learning goals:

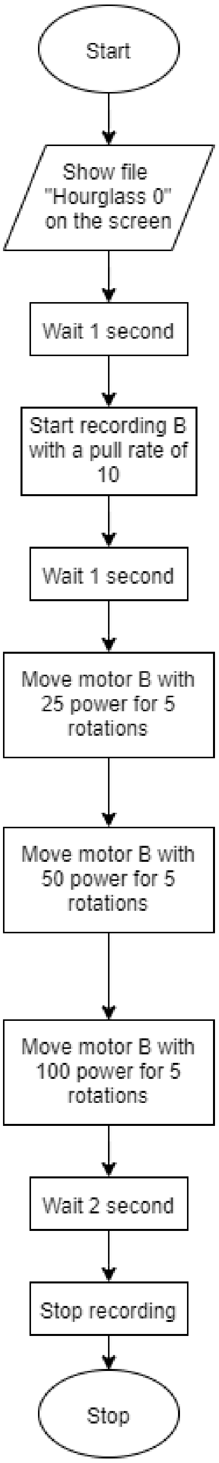
- Understand basic principles about logging and collecting data, understand graphs, analyse these and make conclusions.

Suggested UML Solutions

Activity no1 Comparing Turns


Start
Start recording motors B and C with a pull rate of 1000
Move motor B for 1 rotation with 50 power
Stop motor C
Reset motor B
Reset motor C
Wait 1 second
Turn with 50 steering and 50 power for 1 rotation with motors B and C
Reset motor B
Reset motor C
Wait 1 second
Move motor B with 50 power and keep motor C with 0 power for 1 rotation
Reset motor B
Reset motor C
Wait 1 second
Move motor B with 50 power for 1 rotation
Reset motor B
Reset motor C
Stop recording
Stop

Activity no2 Velocity


Start
Show file "Hourglass 0" on the screen
Wait 1 second
Start recording B with a pull rate of 10
Wait 1 second
Move motor B with 25 power for 5 rotations
Move motor B with 50 power for 5 rotations
Move motor B with 100 power for 5 rotations
Wait 2 second
Stop recording
Stop

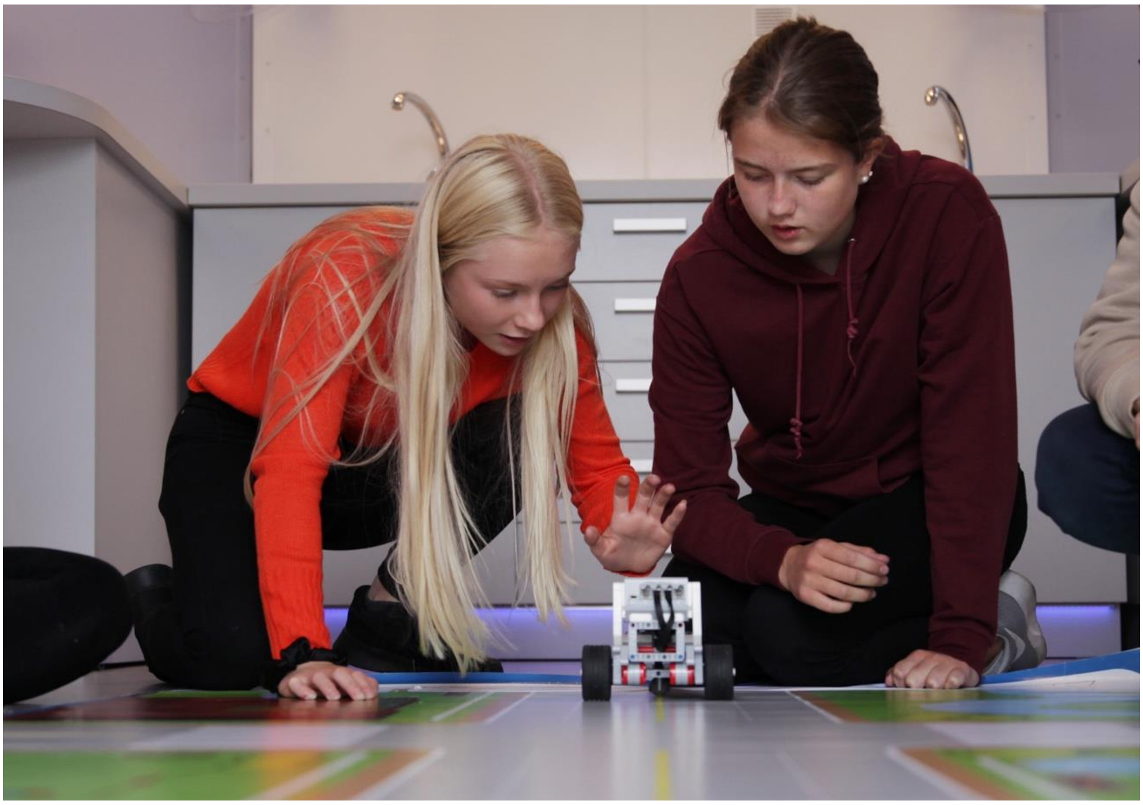
Innovation project

Lesson 29.-30.

Introduction

In lessons 29-32 you will:

- Identify a problem within your own community, that is possible to solve using robotics
- Break the problem down to smaller pieces and make individual algorithms for some of the smaller problems
- Design an innovative solution for the identified problem, and make sure you explain what parts of the problem that a robot can help solve, and which parts humans must or can solve themselves.
- Share your solution with others in a presentation



Co-funded by the Erasmus+ Programme of the European Union

Lesson 29-30: Problem identification using the brainstorming Post-It process & Sketching a solution with algorithms

Learning outcomes:

- 21st century skills (esp. creativity and collaboration)
- Practical use of algorithms and algorithmic thinking

Materials needed:

Post It-notes

Notebooks

Pens/pencils

Computer for presentation

Lesson 29: Problem identification - Post It-process // Brainstorming session

In this part of the project, you are going to identify problems and solutions in your community or everyday life and find one you want to focus on. **Step 1** and **Step 2** of this activity is preferably done together as a group of eight. From **Step 3**, you will divide into smaller groups or pairs.

Step 1: Use a stack of Post-it notes and write down a problem and a possible solution on each note.

Ex: *Problem: Escalators use a lot of power, always running during the day in shopping malls, office building and airports. This contributes to energy waste and is not very sustainable. Solution: Make the escalator run for x seconds using a motion sensor that recognizes a human.*

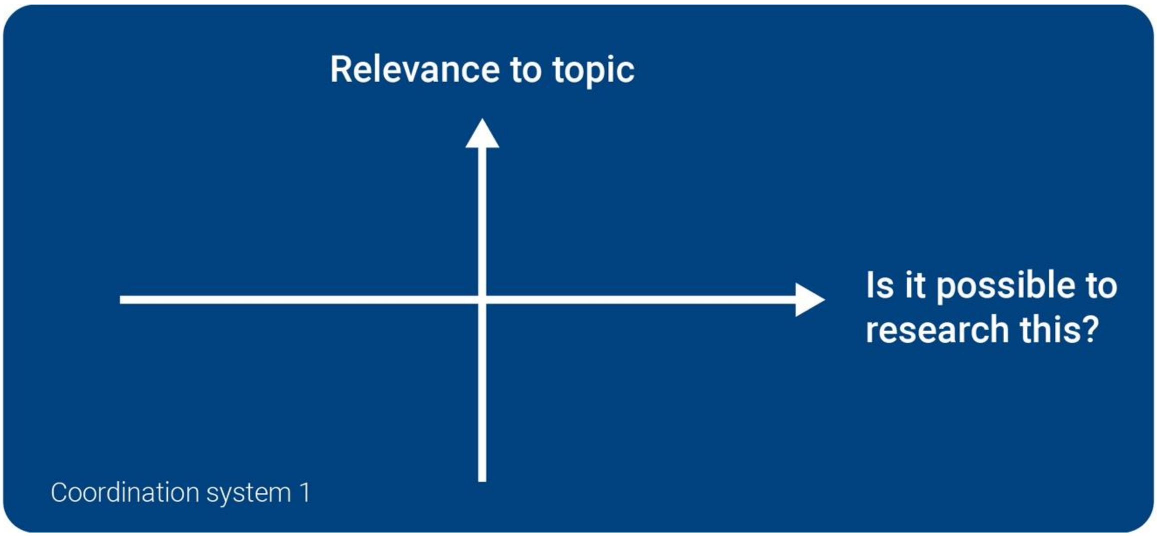
Hand out post-it notes to each team member. Have them write down one problem and a possible solution on each note and place them on a board. Focus on finding at least two ideas each.

Focus on:

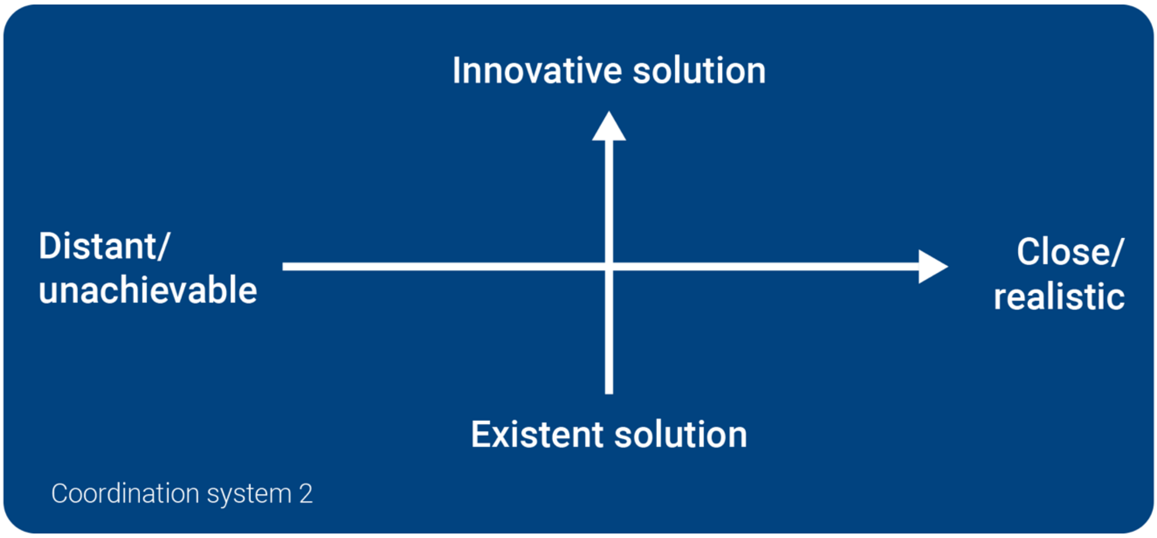
- At this stage, all ideas are good, so let your imagination float freely.
- It is not allowed to criticize any of the team member´s ideas, all ideas are welcome and appreciated.
- Quantity can lead to quality!
- Think of your community and everyday life, what tasks could be made easier or what problems could be solved making life better for you and/or society.
- The aim is to find an idea for a solution (using robotics in some way) that is either innovative or improve upon an already existing idea.

Step 2: Use a black/whiteboard to copy the coordinating system below. Add your ideas on to it to be able to choose which ones to continue to work on.

- Place each idea/post-it notes where it belongs in the coordination system (see picture below)



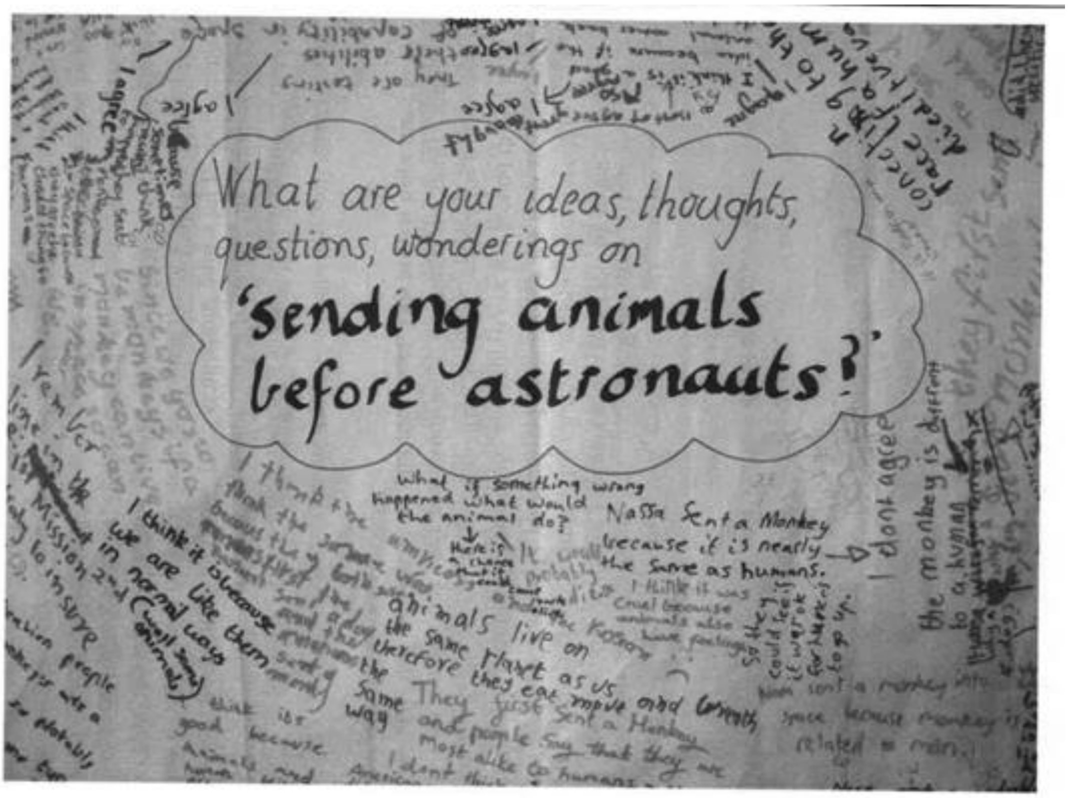
- Once all the ideas are placed in coordination system 1, save all the ideas placed in the upper right-hand square, and discard the others as they are no longer eligible for this project.
- Replace the parameters, and place the post-it notes you saved in the coordination system 2. See picture below.



Co-funded by the Erasmus+ Programme of the European Union

Step 3: Choose the ideas that were placed in the upper right-hand square and write each of them down on a separate sheet of paper. You should have approximately 4 ideas, if you have more, choose 4 that you feel have the most potential. An alternative is to let team members vote on which to continue working on.

- Divide into teams of two. Give each pair 5 minutes to work on one of the ideas each, before they hand the paper on to the next group to get more input, questions and additional ideas.
- Each pair are to write questions to explore and develop the idea, tips, and suggestions on how to further develop the idea, or other relevant comments. Focus on being constructive in this part of the process. Look at the following picture for example.
- Once all the ideas have been through this process, each pair is to choose one idea to bring to the next step.


What are your ideas, thoughts, questions, wonderings on 'sending animals before astronauts?'
What if something wrong happened what would the animal do?
Nasa Sent a Monkey because it is nearly the Same as humans.
I dont agree
the monkey is different to a human
I think it was cruel because animals also have feelings
animals live on the same planet as us
I think it is because they are like them in normal ways

Step 4: Exploring and developing the idea

It is time for each pair to explore and develop the idea further.

Use the idea you chose as your starting point, and answer the following questions:

- Who is the target group for this idea/solution?
- What information sources are available?
- Is our solution innovative, or does it already exist? If it already exists, how can we improve the solution?
- What part of the problem can be solved using robotics?

Lesson 30: **Sketching a solution and its algorithms**

Depending on the idea you chose, you are going to make a draft of your solution and one or several algorithms that is needed to solve parts of or the whole idea.

Some ideas are easy to sketch, while others will demand a lot of time and creative thinking.

Example: Make a cup of tea algorithm

The procedure for making a cup of tea could be:

1) Find a suitable cup, 2 dl size big
2) Find kettle to boil water in
3) Find water and fill in kettle
4) Turn on kettle with desired temperature
5) When kettle turns off, water is hot enough for tea
6) Find teabag
7) Put teabag in cup
8) Fill hot water in cup
9) Find sugar, honey, milk or other
10) Add sugar, honey, milk or other in tea
11) Find spoon
12) Use spoon to stir
13) Wait to temperature is good
14) Drink tea

Now, which elements of this procedure could be fixed by a programmed robot? A really advanced robot could be able to do every point of the list, except from the last point. But, if we were going to make and program a robot with LEGO for instance, maybe just a few points would be interesting. Depending on how the robot was built, it would be able to do some of or most of the procedure.

Questions to consider:

- How will we display the finished product (as a model, on a computer, as a prototype the viewers can try)?
- What parts of the solution can robotics assist with and what parts will require human contributions to complete?
- When you break down the solution, what is the logical first steps?
- Can you implement some of the sensors you have learned about?
- Can you use parallel programming to solve some of the algorithms?
- What advanced programming can you use to make good algorithms? (if-then-else, loop, variables)

Learning outcomes

Discuss and reflect upon how you met the learning goals, and what learning outcomes you can bring with you from this task that will be useful in your future problem solving.

Notes:

Build a presentation

Share your solution

Lesson 31.-32.

Learning outcomes:

- Working with 21st century skills (particularly communication)
- Presentation skills (digital presentation optional)

Materials needed:

Dependant on presentation format agreed upon in class

Lesson 31: Prepare a presentation and share a solution

If you are an innovator who wants to sell or share your ideas, you will at one point need to reach out to investors and companies to convince them to invest in your idea. The best way to do this is to prepare a good pitch or presentation. You will now go on to make a presentation to share your idea and solution! A great presentation calls for clear communication, innovative thinking, good planning and a lot of practice.

Questions to consider:

- How can the idea be explained so clearly that everyone understands the problem, even without prior knowledge?
- How can we convince the listeners that the problem we found, is an important one to solve?
- How will we convey that this is a realistic idea, that we can implement in real life society?
- How can the algorithms used, be presented in a good and understandable way?
- How can we convince the listeners that we are inspired and engaged in our work, the result and its potential?

Task: Use the resources you have to create a presentation. Remember to be creative, thorough and expressive, to really convey your message to the listeners. A good pitch does not always need an extravagant presentation, but the team must be able to communicate their vision and solution. Creativity does not demand a lot of equipment, but it is your pitch, and you decide. Good luck!

Lesson 32: Presentation time – Share your solution with the class

Presentation time and communication of feedback

Now that you have prepared your presentation, you are going to present it. The rest of your class and your supervisor is the audience. While only one team can present at a time, the ones who listen will also have an assignment: Give the presenting group a feedback-sandwich for their effort; two (slices) of positive comments filled with one constructive comment on what could be improved.

Ex.
Positive 1: You were very engaged and inspired by your idea, the enthusiasm made me want to see it realised!
Constructive advice: Your connection with the audience could possibly be even better with more focus on eye contact being less reliant on your manuscript?

Positive 2: You had a detailed algorithm for how to use robotics in solving this real life problem, it made me believe that it really can be done and that it will make a real difference for those who need it!

When everyone has held their presentations, be sure to applaud each other for your hard work and engagement! Well done, you are now officially programmers!

Possible criteria for evaluation:

Clarity of problem statement and solution

Talking freely or using a manuscript?

Eye-contact with audience?

Engagement in the problem and presentation

Logical design of algorithm

Did you know?

The word ''critical'' derives etymologically from two Greek roots: "kriticos" (meaning discerning judgment) and "kriterion" (meaning standards). Etymologically, then, the word implies the development of "discerning judgment based on standards."

Source: criticalthinking.org

Innovative use of robotics in solving the problem

Level of impact on life quality if successful

Realistic

The teams can also give suggestions for other relevant evaluation criteria!

Notes:

Learning outcomes

Assess and reflect together with your teammates, how did you meet the learning outcomes and the different evaluation criteria? If you were given a similar task in the future, what would do similarly and what would you do differently?

Programming

Lesson 33.-34.

Materials needed:

- LEGO Mindstorms Edu EV3 base robot with touch and colour sensor
- base set LEGO pieces for building

Variables and Constants

The most basic thing to do within a program is to store information which can be reused or assessed later in the program's logic.

What is a programming variable?

A variable is a storage location for data which we give a name within a program. The name provides a way of labelling the data so that we can better understand its purpose within the program's logic. Think of it like a notepad or a box that holds a value for you. You can name the variable whatever you want.

You can define the type of variable:

◦ Numeric (Holds a number)

◦ Logic (Holds True/False)

◦ Text (Holds lines of text … "Hello World")

◦ Numeric Array (Holds a set of numbers … 1,2,3,10,55)

◦ Logic Array (Holds a set of logic … True, True, False)

They can be used as either Inputs or Outputs so you can either….

◦ Write – put a value into the variable

◦ Read – retrieve the last value written to the variable

To store and access variable information, use a Variable block, which you'll recognize by the suitcase icon on the block. Write (Inputs) have a bump up Read (Outputs) have a bump down.

What is a programming constant?

In computer **programming**, a **constant** is a value that cannot be altered by the program during normal execution, i.e., the value is **constant**.This is contrasted with a variable, which is an identifier with a value that can be changed during normal execution, i.e., the value is variable.A **constant** is a named memory location which temporarily stores data that remains the same throughout the execution of the program.

The Constant block provides a starting point for a data wire whose value you can choose manually. You choose the data wire type by selecting a mode (Text, Numeric, or Logic), and you enter its value in the Value field. The Constant block is useful if you want to configure the settings of multiple blocks with the same value.

Activities

After every activity pupils have to present their developed projects to the other teams for discussion and changes.

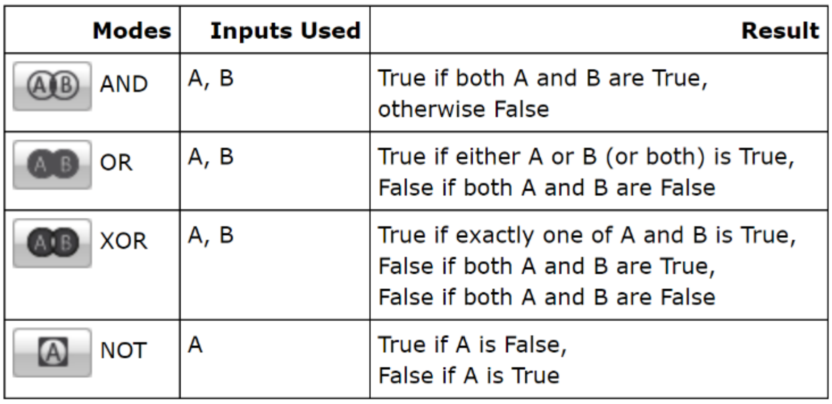
Activity no1
Can you make a program that displays the number of times that you have clicked the up button?

Activity no2
Can you write a program that counts the number of black lines you have crossed?

Logic

The Logic Block does a Logic operation on its inputs, and outputs the result.A Logic Block takes inputs that are True or False, and produces a True or False output Logic values can be used as inputs into loop exists and switch conditions. It is found in the Red Programming Pallet tab.

Different Modes in the Logic Block



Modes	Inputs Used	Result
AND	A, B	True if both A and B are True, otherwise False
OR	A, B	True if either A or B (or both) is True, False if both A and B are False
XOR	A, B	True if exactly one of A and B is True, False if both A and B are True, False if both A and B are False
NOT	A	True if A is False, False if A is True

Activity no3

Make your robot drive forward until EITHER the Touch Sensor is pressed or the Color Sensor detects black. Follow the steps.

- Turn the motors on
- Add the Logic and Sensor Blocks
  ◦ A. Use a Logic Block in the OR mode
  ◦ B. Add the inputs: Take a color sensor and a touch sensor blocks and wire them into the Logic Block as inputs
- Add a Loop and loop exit condition:
  ◦ Place the Sensor and Logic Blocks in a loop
  ◦ For the exit condition of the loop, select logic. Wire the result of the Logic Block into the exit condition
  ◦ If the result of STEP 2 is True, you should exit the loop and stop the robot

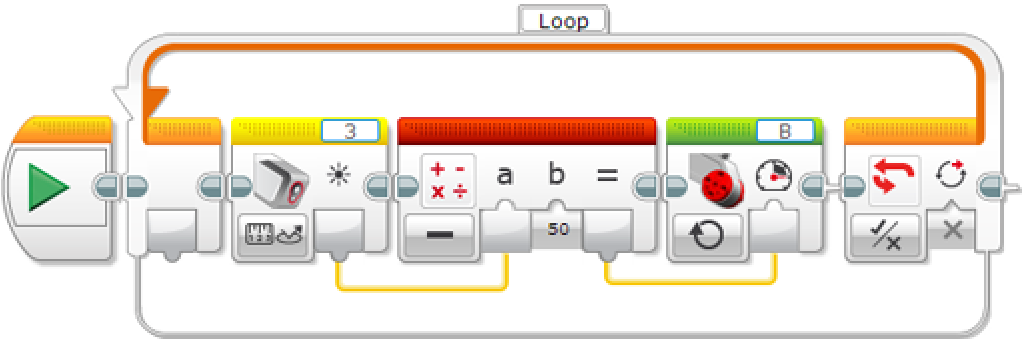
Activity no4

Each team is required to build a system that will read the water level (blue bricks). When the water level pass the limit then the engine will open the safety and water will continue to flow. Open the programs **Activity 4 tubes** and **Activity 4 overflow** to help you.

Math

The Mathematics block does a mathematics calculation on its inputs, and outputs the result. You can do a simple mathematics operation with one or two inputs, or enter a formula with up to four inputs. Choose the mathematics operation you want to use by selecting a mode with the Mode Selector. After selecting the mode, you can choose values for the inputs. The inputs available will change depending on the mode.

Activity no5



What do you think the above program do?

Learning outcomes

Assess together with your partner, how did you reach following learning goals:

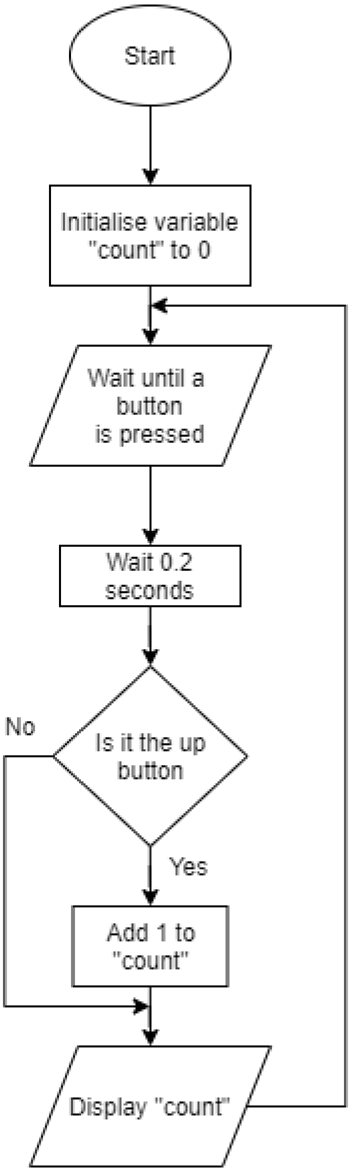
Use programming concepts in algorithms such as

- Constant
- Variable
- Logical operations
- Math

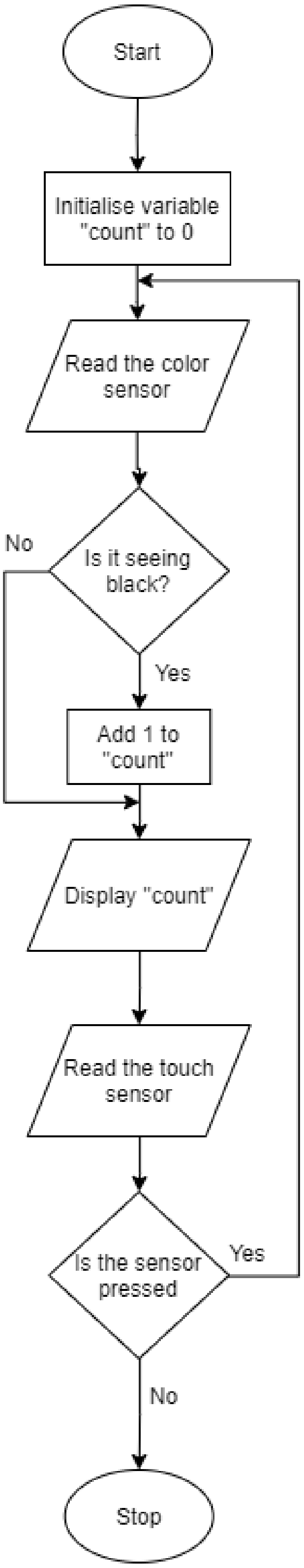
Suggested UML Solutions

Co-funded by the Erasmus+ Programme of the European Union

Activity no2

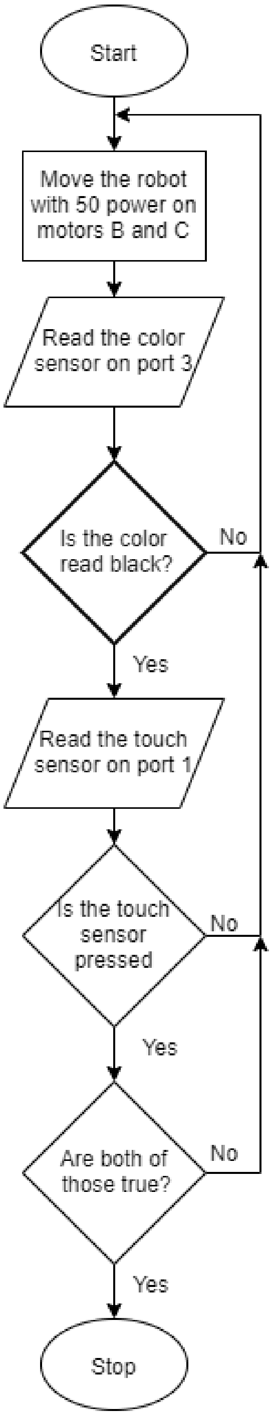


Activity no1

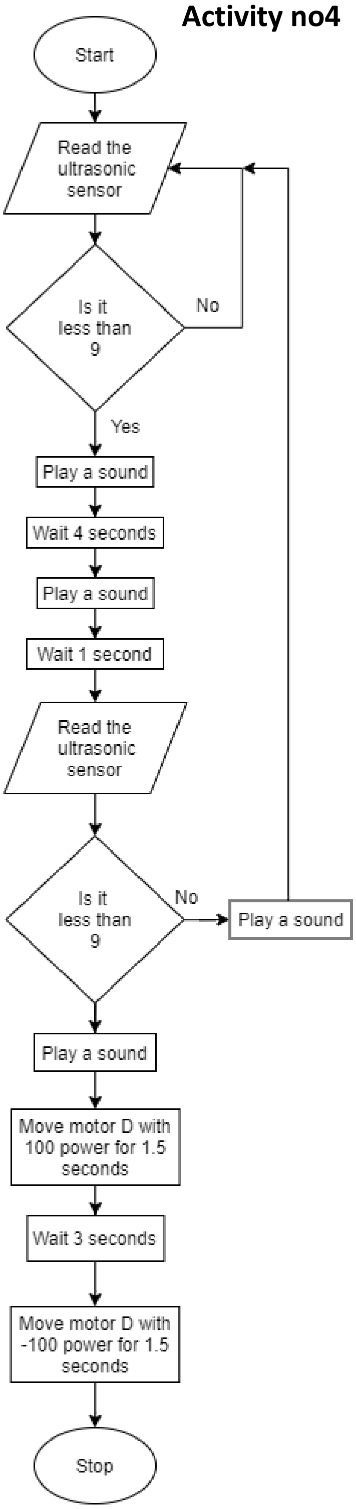


Co-funded by the Erasmus+ Programme of the European Union

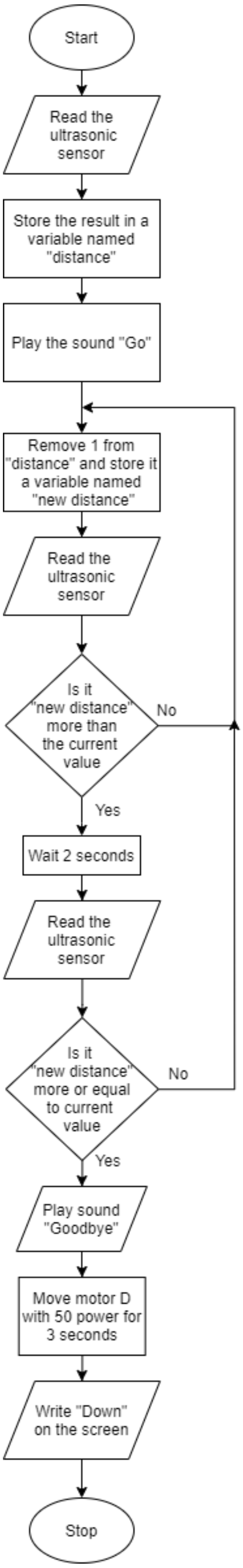
no3


Start
Move the robot with 50 power on motors B and C
Read the color sensor on port 3
Is the color read black?
No
Yes
Read the touch sensor on port 1
Is the touch sensor pressed
No
Yes
Are both of those true?
No
Yes
Stop

Activity no4

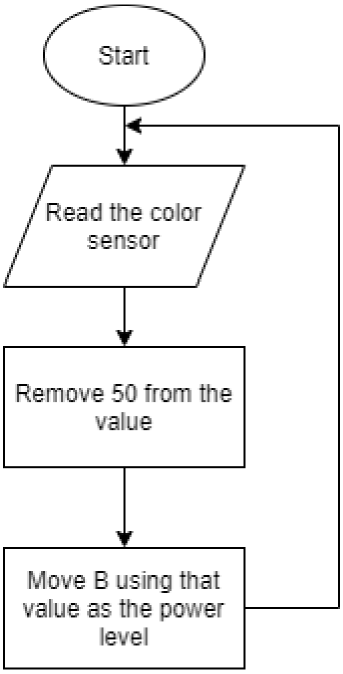

Start
Read the ultrasonic sensor
Is it less than 9
No
Yes
Play a sound
Wait 4 seconds
Play a sound
Wait 1 second
Read the ultrasonic sensor
Is it less than 9
No
Play a sound
Play a sound
Move motor D with 100 power for 1.5 seconds
Wait 3 seconds
Move motor D with -100 power for 1.5 seconds
Stop

Activity


Start
Read the ultrasonic sensor
Store the result in a variable named "distance"
Play the sound "Go"
Remove 1 from "distance" and store it a variable named "new distance"
Read the ultrasonic sensor
Is it "new distance" more than the current value
No
Yes
Wait 2 seconds
Read the ultrasonic sensor
Is it "new distance" more or equal to current value
No
Yes
Play sound "Goodbye"
Move motor D with 50 power for 3 seconds
Write "Down" on the screen
Stop

Co-funded by the Erasmus+ Programme of the European Union

Activity no5



Programming

Lesson 35.-36.

Materials needed:

- LEGO Mindstorms Edu EV3 kit
- base set LEGO pieces for building

Comparison

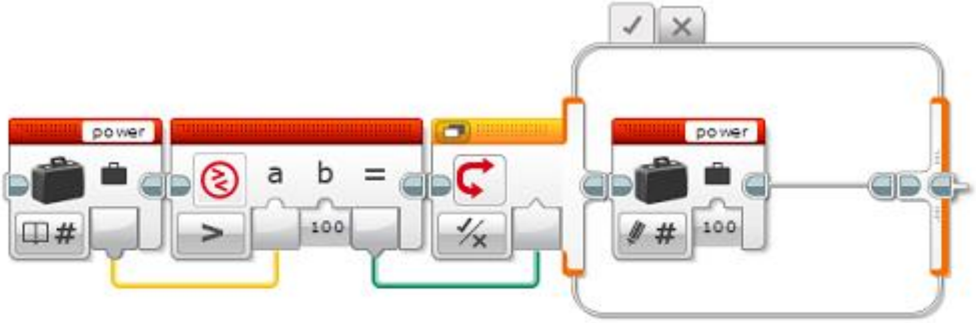
The Compare block compares two numbers to find out whether they are equal, or which number is greater. You can choose one of six different comparisons. The output result is True or False. Choose the type of comparison you want to use by selecting a mode with the Mode Selector. The block will calculate the Result output by comparing the two inputs A and B as shown in the table below.

Mode	Inputs Used	Output Result
= Equal To	A, B	True if A = B, otherwise False
≠ Not Equal To	A, B	True if A ≠ B, otherwise False
> Greater Than	A, B	True if A > B, otherwise False
< Less Than	A, B	True if A < B, otherwise False
≥ Greater Than or Equal To	A, B	True if A ≥ B, otherwise False

Co-funded by the Erasmus+ Programme of the European Union

≤ Less Than or Equal To	A, B	True if A ≤ B, otherwise False

Example



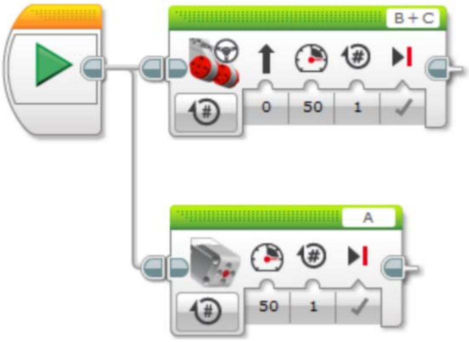
This block sequence tests to see if the value of the "power" variable is greater than 100, and if so, it sets it to 100. The Compare block compares the value of the variable to 100, and the Logic result is used by the Switch to choose whether to change the value of the variable.

Parallel and Serial Programming

Serial programming is when each command is executed one after the other. Parallel programming is when more than one command is executed at the same time.

What are Parallel Beams?

Parallel beams allow you to run two or more blocks at the same time. What if you have one or more attachment arms connected to motors and you want to turn these arms while the robot is moving to complete a mission.



To create a parallel beam click and drag on the bump on the right center of any block and release once you hover over the inverted bump on the left center side on a block.

Note: Blocks before the split will run one at a time. After the split blocks on the two "beams" will run at the same time.

What Real life

Do you recognize where these concepts are used around you? Can you point and name machines/robots next to you where you see these programming concepts used in the behavior? Like how the logic of the elevator is built?

What is this robot doing?

After every activity pupils have to present their developed projects to the other teams for discussion and changes.

Activity no1:

Step 1: Build a robot that can move and pick up objects. Instruction in Lego Education EV3 project will help you. Use the ultrasonic sensor too.
Step 2: Can you write a program that uses parallel beams that have to move and pick up an object at the same time?

Activity no2 :

Step 1 : Use the previous construct of the robot with two motor and an ultrasonic sensor. This robot is a puppy and ultrasonic sensor is puppy's eyes.
Step 2: Program your puppy to follow your hand. When your hand is between 10 – 15 cm then the puppy will go forward and when your hand is between 0 – 15 cm the puppy will go backward.
Step 3: Do you use any variables in your program and why?

Learning outcomes

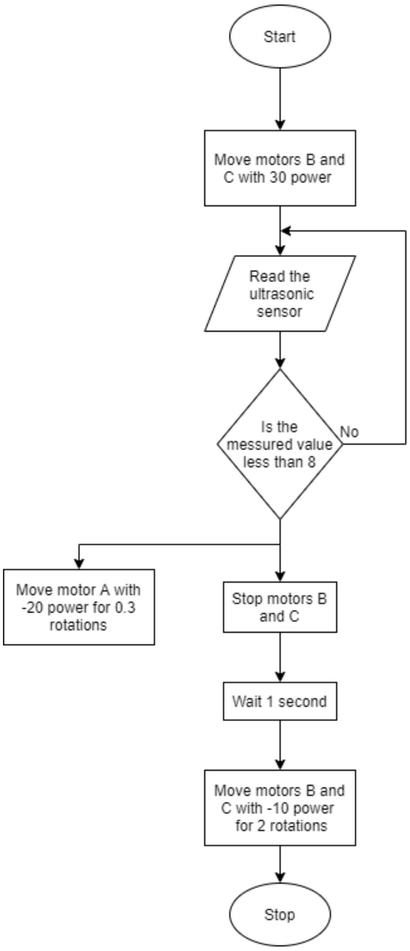
Assess together with your partner, how did you reach following learning goals:

- Student knows what the difference between serial and parallel programs is and understands when to use one or another.

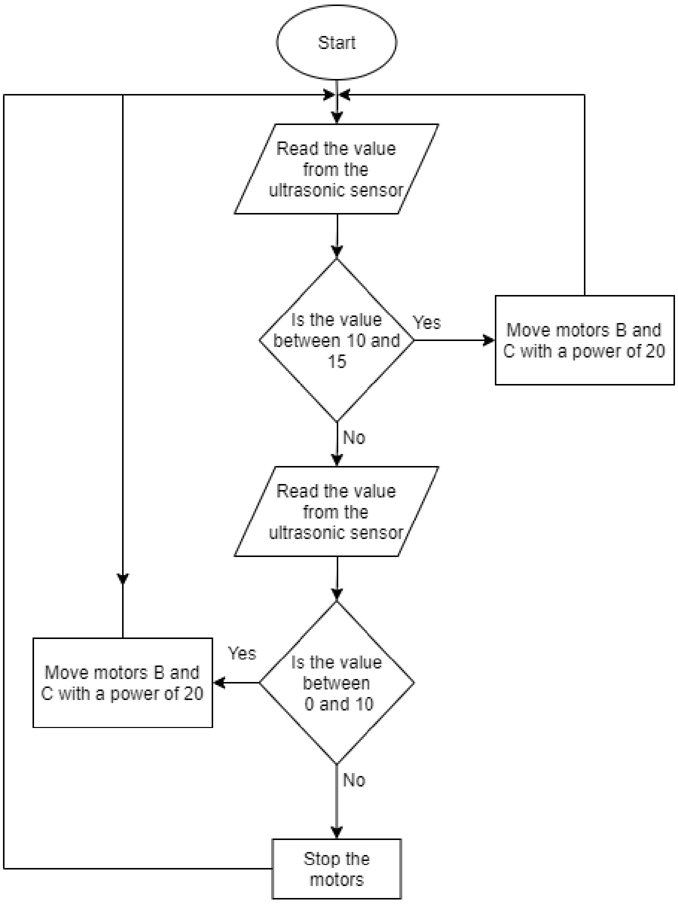
- Use programming concepts in algorithms such as
  - Comparison

Suggested UML Solutions

Activity no1:


Start
Move motors B and C with 30 power
Read the ultrasonic sensor
Is the messured value less than 8
No
Move motor A with -20 power for 0.3 rotations
Stop motors B and C
Wait 1 second
Move motors B and C with -10 power for 2 rotations
Stop

Activity no2:


Start
Read the value from the ultrasonic sensor
Is the value between 10 and 15
Yes
Move motors B and C with a power of 20
No
Read the value from the ultrasonic sensor
Is the value between 0 and 10
Yes
Move motors B and C with a power of 20
No
Stop the motors

Building a competition robot

LEGO Sumo challenge

Lesson 37. - 40.

Materials needed:

- Assembled LEGO EV3 base robot with color sensor
- Sumo ring

One of the most popular robotics competitions among students is the LEGO Sumo. Sumo itself is a very popular sports in Japan where male and female Sumo wrestlers are competing against each other. (See Figure 1.) The main goal of Sumo competition is to push the opponent out of the SUMO ring.

Figure 1. Two Japanese Sumo wrestlers pushing each other.

Co-funded by the Erasmus+ Programme of the European Union

LEGO Sumo

LEGO Sumo is a competition where two LEGO robots are competing against each other. (See Figure 2). The main requirement is that the robot must be made from LEGO bricks only and it must use the controllers and sensors made by LEGO. The main platforms that the students are using are LEGO Mindstorms NXT and LEGO Mindstorms EV3. In 2020 also the newest LEGO SPIKE Prime can be used.

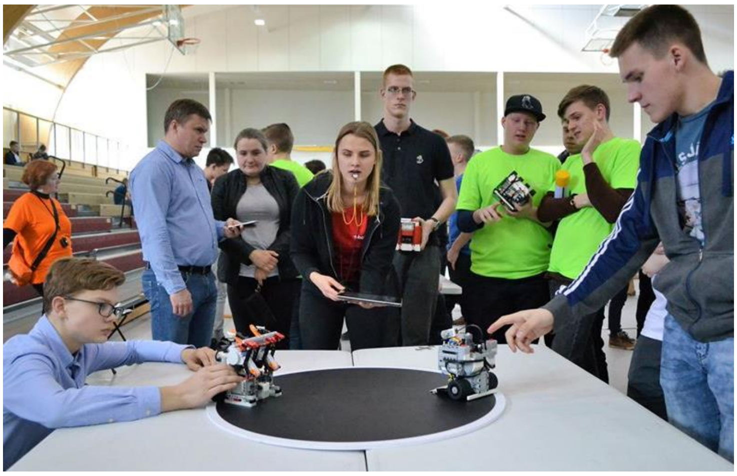


Figure 2. Two LEGO robots waiting for the referee's start sign

Basic rules

The field

The field where the robots are competing is a black ring with a white border. In LEGO Sumo the black ring is with a diameter of 77 cm and the white border is 2,5 cm wide and the ring is normally made of plastic or wood. (See Figure 3).

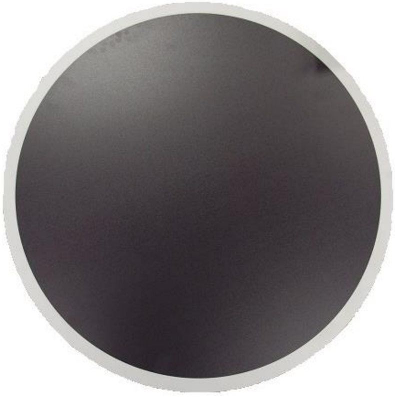


Figure 3. Black Sumo ring with white border

The Robot

LEGO Sumo robot is made of LEGO bricks. Normally a LEGO Technics type bricks are used. The measurements for a LEGO Sumo robot are 15 cm x 15 cm (with the tolerance of 2mm). The maximum weight is 1 kg.

NB! Starting height is not limited.

In the competition a 15,2cm x 15,2 cm box is used to measure the robot and a scale is used to weight the robot in the inspection area.

When starting the robot (after Referee's permissive sign) it must wait for 5 seconds before it is starting to move.

The Match

Sumo match generally contains three rounds and lasts up to three minutes. The robot who will be first to earn two Yuko points (effective points) during the time of the match, will be the winner. Match time is measured during rounds, not between them.

If only one Yuko point has been earned by the end of the match time, the winner is the robot that earned it.

So, the match ends with **2:0** or **2:1** or **1:0** (if the time ends).

Starting the match

Before each round and with the signal from the referee, the contestants place their robots simultaneously on the Dohyo (on the field). The robots must be placed in reciprocal sectors and at least some part of the robot must stay on the white line (see Figure 4 Starting cross). The robots are not allowed to move after they have been placed on the Dohyo.

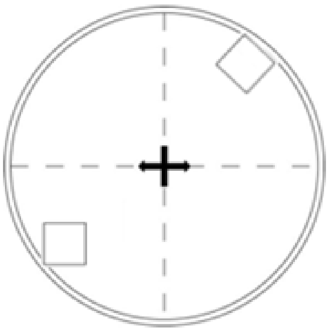


Figure 4. Starting position for the robots.

After the signal of the referee, the operators start the 5-second timer in the robot and immediately leave the area of Dohyo Jyonai. The robots may start moving in 5 seconds after they have received the start command.

Ending the match

The referee gives a signal to stop the match and operators stop their robots. You have maximum of 30 seconds to repair your robot between the rounds.

Competition rules

Your teacher should have the official LEGO Sumo competition rules. These rules are used in big International competitions like Robotex International, Baltic Robot Sumo etc. These rules use the official Japanese Sumo principles and names and are because of that sometimes hard to read. Do not let these Japanese names to scare you, the principles are quite logical.

Sumo strategies

There are numerous strategies in LEGO Sumo. Some expert teams have more than 30 different strategies (programs inside the robot) that they can choose. You can just go forward and hope that the opponent is there, or you can use different sensors to locate the opponent and then attack. Some robots will first peek left or right and then attack.

When building a Sumo robot, you should think about some aspects:

- Making your robots face as low as possible to go under the opponent robot to lift it and push it out. You can use a LEGO plate etc to build a plow.
- Some robots have a motor in front of the robot to lift other robot up.
- You have to think about the friction of the wheels and how to make it better. This means that more friction gives you more power to push.
- Think about the defense mechanisms for not letting the opponent getting under your robot.
- Using a color or light sensor to detect the white border line (for not driving unintentionally out of the field).
- Using an ultrasonic or infrared sensor to detect another robot.
- Etc…

OK. It’s time to start building the robot now. You can find the task in next page in the Activities section.

Activities

Activity no1:

Your first task is to modify your EV3 base robot in a way that the color sensor is in front middle of the robot. So, it can be used to detect the white line.

Now make a program using an EV3 programming environment:

1. Robot is waiting 5 second
2. Then it starts driving forward until it detects the white line (using a color sensor) and
3. Then stops and drives backward for 1 rotation and then turns left or right and
4. Then goes back to 2. point (using Loop).

The measurements of the robot are not fixed. This is just training.

Activity no2 – breaking the rules

Build a Sumo robot that weights up to 1 kg and the robot must fit on the A4 paper. This task is to think about building a Sumo robot with limited constraints (wight and size) and to have fun.

You have round 30 minutes to build the robot so think fast and add the pieces. After 30 minutes you can have a small competition to see who had a better strategy.

As there is so less time to build your robot the teacher will give everybody the same program. So, it will be only building!

Activity no3 – real deal

Now it is time to go serious. Your last task is to build a competition ready LEGO Sumo robot. You must follow all the requirements there is for competition robot:

- Size 15 cm x 15 cm
- Weight 1 kg
- Starting height is not limited

If you have time, make at least 3 different programs that have different strategies. Make a small competition with your classmates.

Learning Outcomes

In these lessons you learned how to make a competition ready Sumo robot. You also looked through the building principles, moving and attacking strategies and features to make a competitive robot.