Team Manual

Output Devices – motors

EV3 Large motor

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

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**
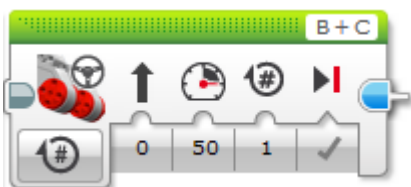
the Duration in rotations.

Figure 1. Large motor block

Figure 2. Move Steering block With this block you can move your robot with two motors.
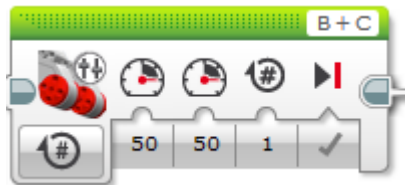

Figure 3. Move Tank block With this block you can give a different power to both motors at the same time.
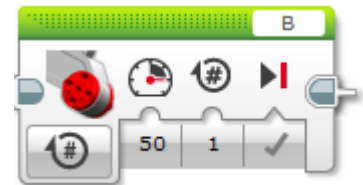

Figure 4. With this block only the motor in port B is moving. This means that the robot is turning over the middle point of
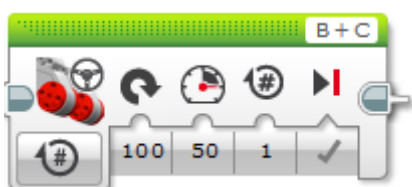

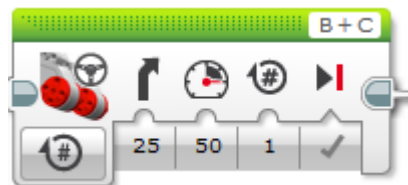*Figure 5.* With this setup the robot is turning on spot.

Figure 6. With this setup the robot is turning with two motors and the turning radius and directions depends on the Steering value.

Loop block

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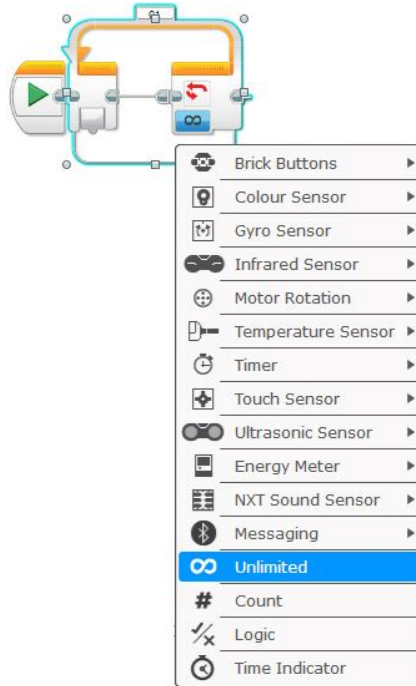
Wait block

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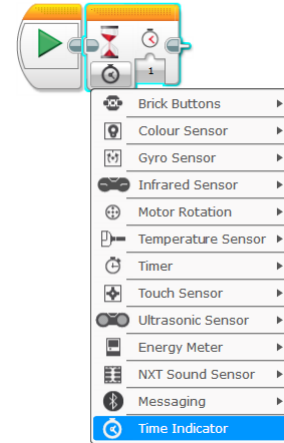
Sensors are they also very often used. This means that the robot is waiting for a sensor to be activated (like touch sensor for example) or a certain sensor value is reached (for example, the program is waiting for the distance of Ultrasonic sensors to reach more than 50 cm and then the program continues.

Driving Base

Materials needed:

- Assembled LEGO EV3 base robot without sensors

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:
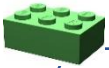- Large Motor and Wheel
- Bevel Gears

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.

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

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

Output devices – feedback

Materials needed:

- Assembled LEGO EV3 base robot without sensors.

Activity no1:

Your LEGO Mindstorms EV3 robot has LED lights in the buttons area. 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).
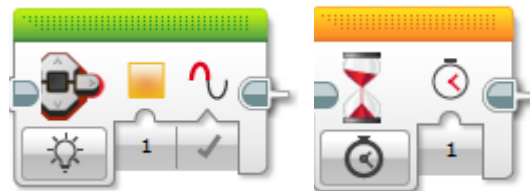
Figure 7. The *Brick Status Light* block on the left and *Wait* block on the right

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.

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 8. 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).
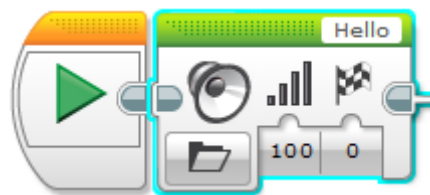
Figure 9. Sound block in EV3 program

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.

Sensors

Materials needed:

- LEGO EV3 base robot assembled with all sensors

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.
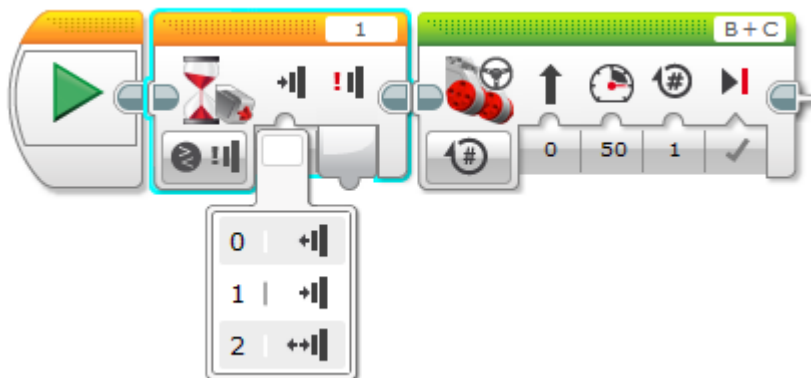
See that touch sensor has three states (figure above):
- 0 – react only when button is released
- 1 – react only when button is pressed
- 2 – react only when button is pressed and released

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

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.
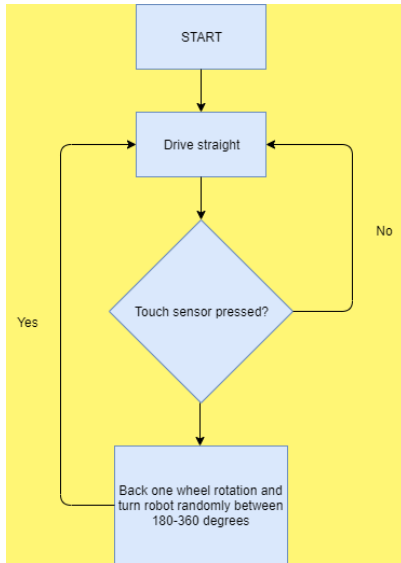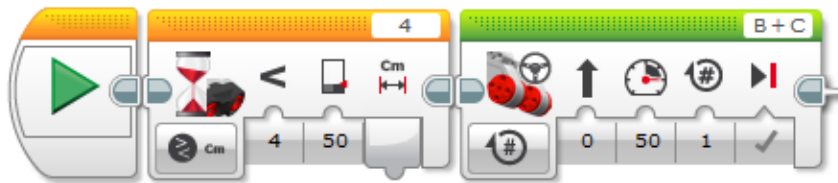

Can you remake the program so robot drives forward until it detects something closer than 50 cm?

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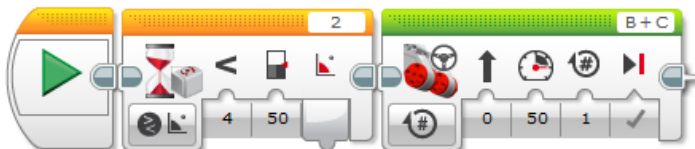
Gyro sensor can also detect speed of angle change degrees/sec.

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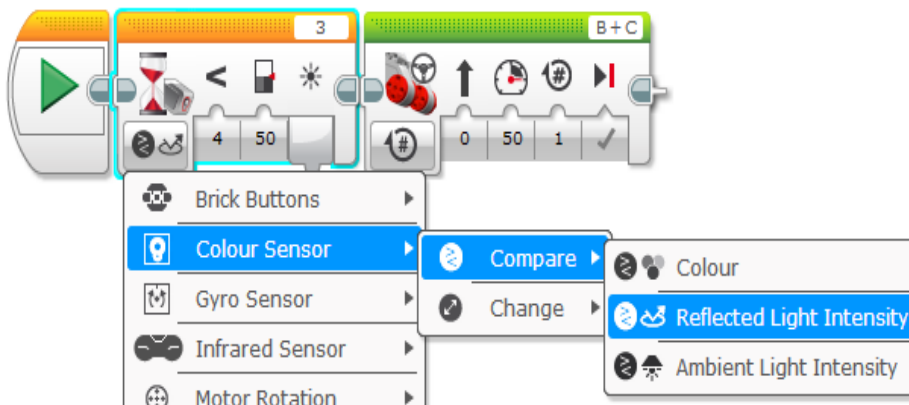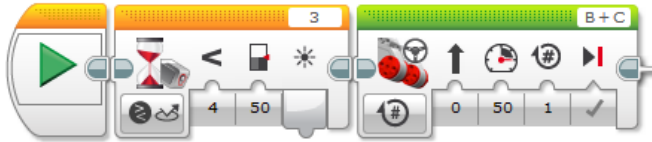
Light sensor has ability to detect colors and ambient light. See from the picture on the left.

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.

```
START            START            START
  |                |                |
  v                v                v
Gyro value      Light value      Distance value
changed?        changed?         changed?
  |                |                |
 No               No               No
  |                |                |
 Yes             Yes             Yes
        \          |          /
         \         |         /
          v        v        v
           ALARM goes
              ON
```
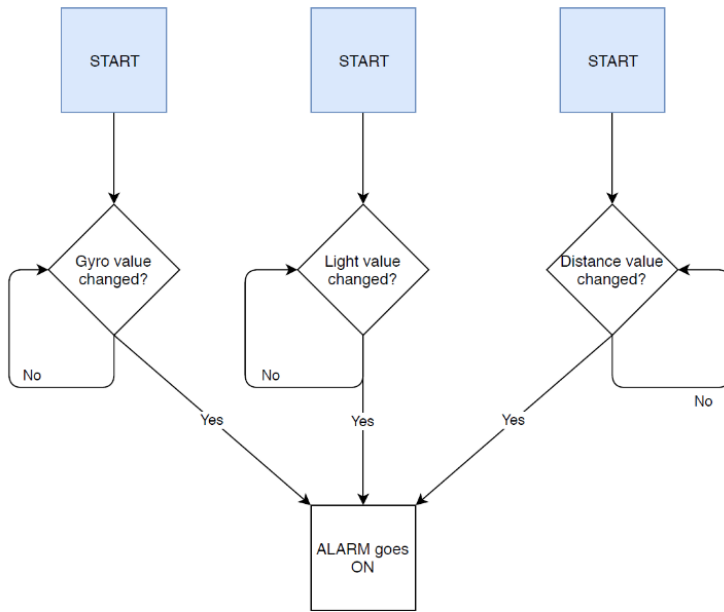
See the example program (from teacher) and make it to work the way you want.

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.

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

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.

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.
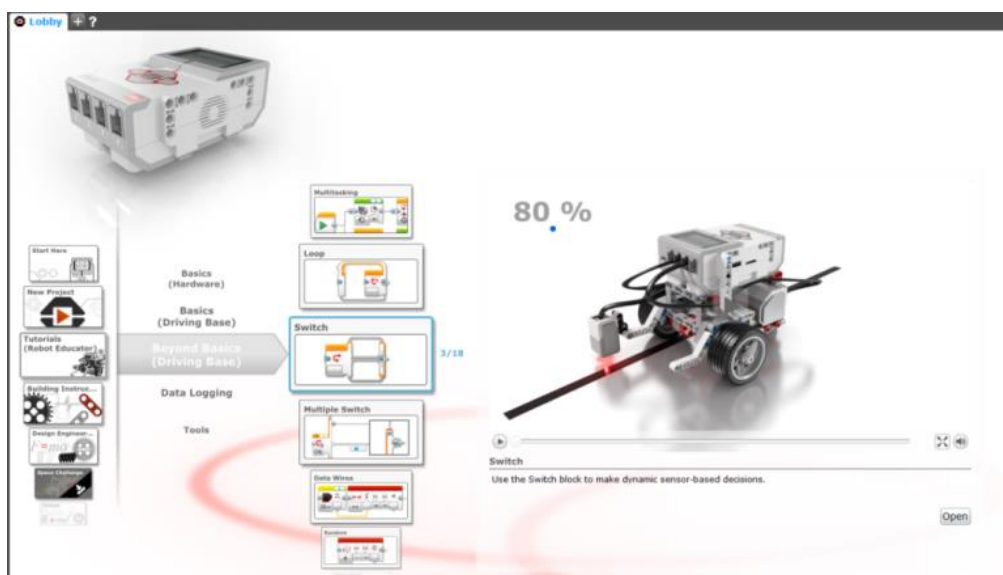
Safe robots

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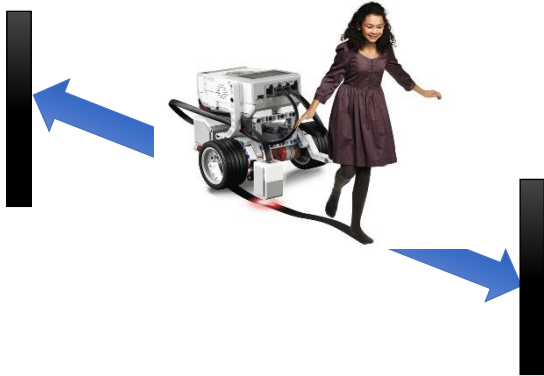
Here you can find sample programm below. But it does not work perfectly. Lower block is called loop interrupt and it stops upper loop. Can you make it work?
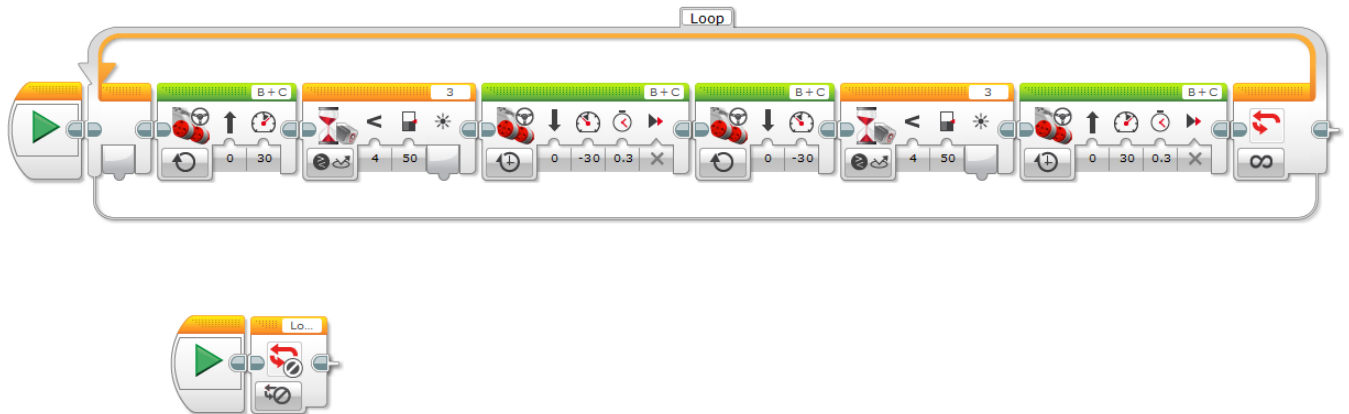
Developing STEM competence with robotics

Teambuilding – Algorithmic thinking
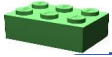
> **Materials needed:**
>
> Tape or chalk, Paper and pencil, Codes with example

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.

> **Algorithmic thinking involves**
>
> - the detailed steps that is needed to solve a problem
> - use programming skills to make the computer solve the problem

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

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

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

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.

Activity 2: Teambuilding – Building the Cuboid

Materials needed:

LEGO Mindstorms edu EV3 kit

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

The students will have one of the two roles "*driver*" and "*navigator*" while programming.

Driver is looking out of the circle

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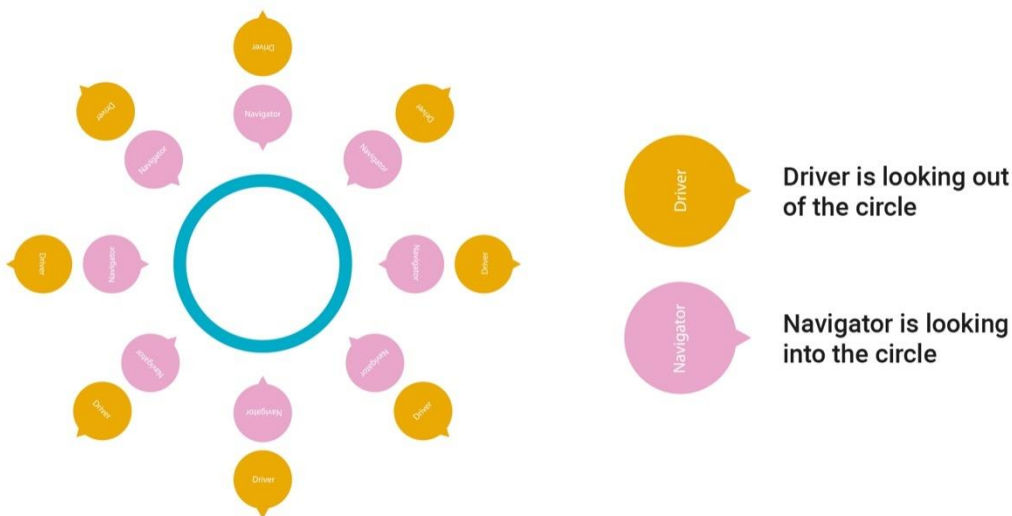
To minimize disturbance, each group should be placed sitting back to back in a big circle and with some distance to each other.

- 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

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

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.

Levels and points

There are four levels with different kinds of tasks to execute. 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.

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

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
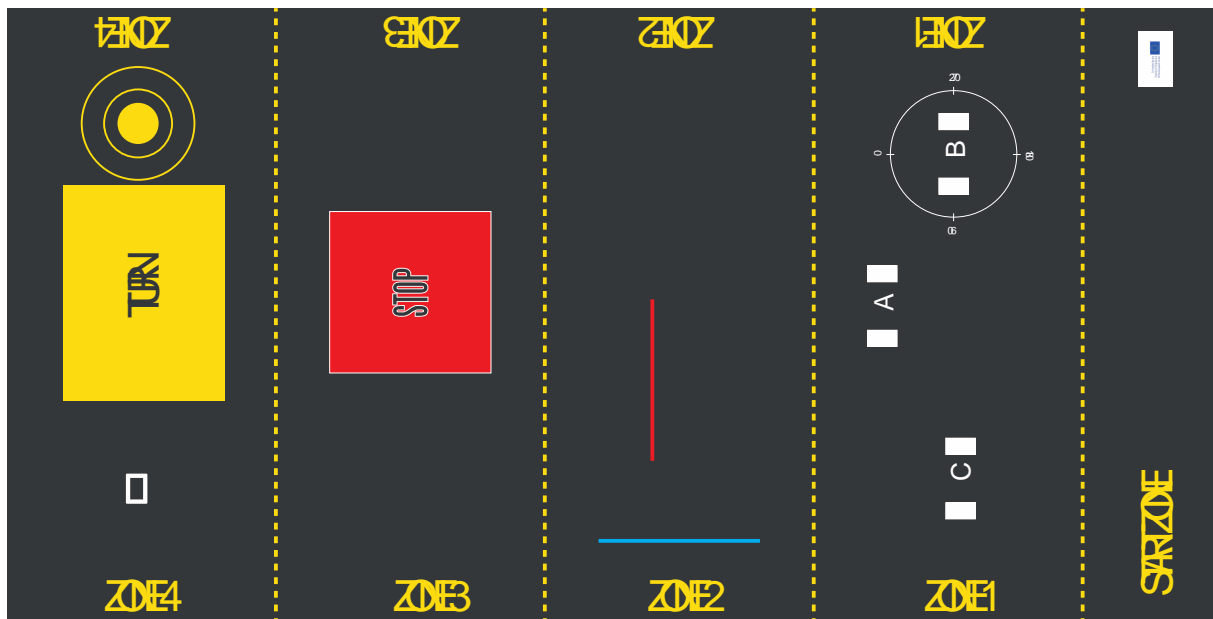
Illustration 1: Airport scenario traning-mat

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.

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 3 = 100
Level 1b	200		Under 2 = 400	Under 2 = 200
Level 1c	200		Under 1,5 = 600	Under 1,5 = 300
Level 2	300			
Level 3	500			
Level 4	700 + 100			
TOTAL:				

Presentation real world problem solving

Materials needed:

Dependant on presentation format agreed upon in class

Prepare a presentation and share a solution

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. Creativity does not demand a lot of equipment, but it is your pitch, and you decide. Good luck!

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. 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.

When everyone has held their presentations, be sure to applaud each other for your hard work and engagement!

Advanced Programming

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.

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.

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.

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
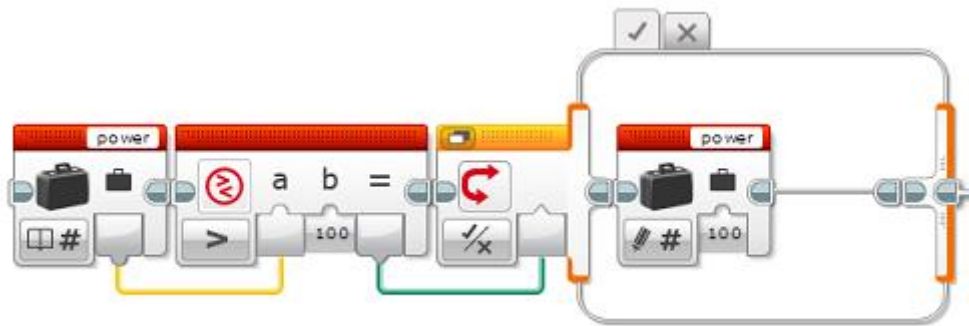
Activity no5
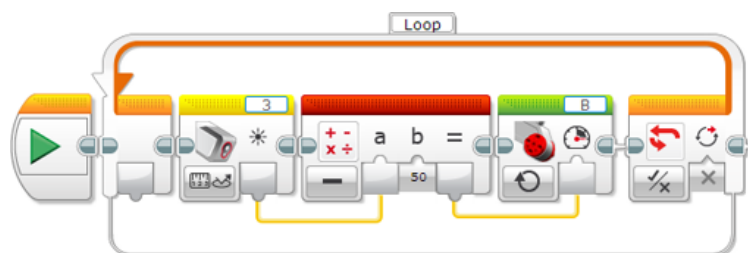
What do you think the above program do?

Comparison

The Compare block compares two numbers to find out whether they are equal, or which number is greater. The output result is True or 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
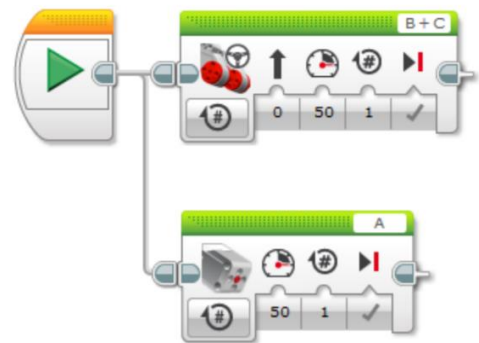
Activities

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?
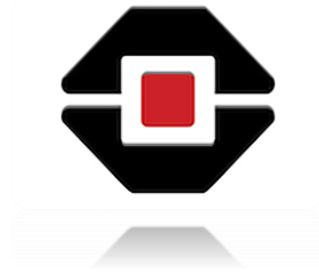
Building a competition robot

Materials needed:

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

LEGO Sumo

LEGO Sumo is a competition where two LEGO robots are competing against each other.

Figure 10. 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.

Figure 11. 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.

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).
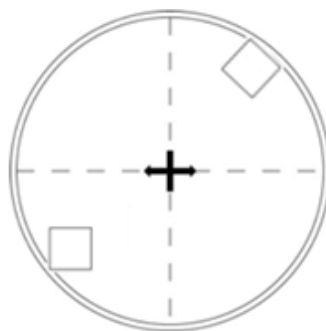

Figure 12. Starting position for the robots.

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.

Sumo strategies

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.

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.