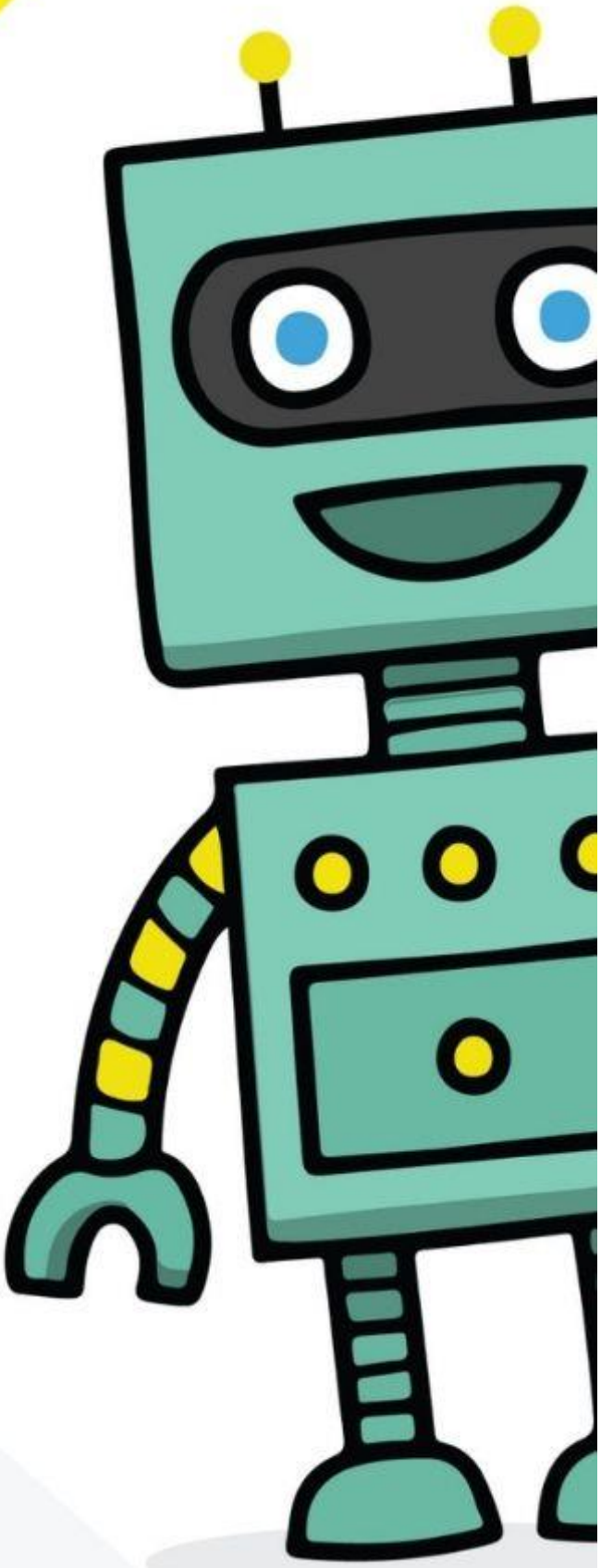


CODESKILLS
4ROBOTICS

DEVELOP BASIC STEM SKILLS AND PROGRAMMING

PRIMARY SCHOOL EDUCATION

Erasmus+

INFORMATION

TITLE OF THE OUTPUT:
EDUCATIONAL BACK PACK

MODULE 1 - DEVELOP BASIC STEM SKILLS AND PROGRAMMING

PROJECT INFORMATION

PROJECT: ERASMUS+

PROJECT TITLE: CODESKILLS4ROBOTICS
PROMOTING CODING & STEM SKILLS THROUGH ROBOTICS: SUPPORTING PRIMARY SCHOOLS TO DEVELOP INCLUSIVE DIGITAL STRATEGIES FOR ALL

PROJECT NO.: 2018-1-EL01-KA201-047823

PROJECT COORDINATOR: N.C.S.R DEMOKRITOS

PROJECT PARTNERS:

DEMOKRITOS

TEI of Crete
Technological Educational Institute of Crete

LIFELONG LEARNING PLATFORM
EUROPEAN CIVIL SOCIETY FOR EDUCATION

Emphasys CENTRE

Hälsinglands Utbildningsförbund

REGIONAL DIRECTORATE OF PRIMARY AND SECONDARY EDUCATION OF CRETE

Table of Contents

Introduction 1

What to Expect 1

Why Lego Boost? 2

What Do You Need? 2

Let's Speak Lego 3

This is the Lego Canvas 7

Section A - Basic Robotics Movements 9

REA - Robotics for Educational Applicability 9

1.1 Let's Build REA 9

Building REA 10

1.2 Moving Instructions of REA 31

Coding REA 31

The Coding Environment 31

The Block Pallets 32

Moving REA 35

Sample Programs of Moving Blocks 36

Example Exercise Sheet 39

1.3 Using Loop Commands with REA 39

The Three Types of Loop Blocks 39

Sample Programs of Loop Blocks 41

Example Exercise Sheet 43

Section B - Robotics Sensors 45

1.4 Using Sensors with REA 45

Sensors 45

Building the Sensor 45

The Lego Boost Color and Distance Sensor 47

Detecting Objects 47

How the Sensor Works 47

The Detecting Objects Blocks 48

Sample Programs of Detecting Obstacles 49

Example Exercise Sheet 50

Detecting Colors 51
The Detecting Colors Blocks: 53
Sample Programs of Detecting Colors 54
The If/Else Blocks: 56
Example Exercise Sheet 1 57
Example Exercise Sheet 2 59
1.5 Following Walls with REA 60
Following Walls 60
Building the Sensor 61
Example Exercise Sheet 65
1.6 Following Lines with REA 66
Follow the Line 66
Building the Sensor 67
Sample Programs for Following the Line 72
Example Exercise Sheet 74
1.7 Detecting Sound with REA 75
React on Sound 75
The Sound Sensor Blocks: 75
Sample Programs for Detecting Sound 76
Example Exercise Sheet 78
1.8 Navigating REA with a Remote Controller 79
Remote Control 79
The Remote Control Blocks: 79
Sample Programs for Remote Control 80
Example Exercise Sheet: 82
Section C - Advanced Robotics 85
1.9 Using Gears with REA 85
Gears 85
Geared Up REA 85
Geared Down REA 94
1.10 Using Variables with REA 96
Mathematics and Calculations 96

The Operator Blocks: 98
The Variable Blocks: 98
Sample Programs for Mathematics and Calculations 99
Example Exercises 101
1.11 Troubleshooting 102
Bibliography 104
Index 105

Table of Figures

Figure 1: The Lego Boost Kit Box 3
Figure 2: The 3 Boost Bricks 4
Figure 3: The Move Hub 4
Figure 4: The Battery compartment 4
Figure 5: The Color and Distance Sensor 4
Figure 6: The Interactive Motor 4
Figure 7: The Main Lobby Screen (Left) 5
Figure 8: The Main Lobby Screen (Right) 6
Figure 9: The Curtain Covering the Creative Canvas 7
Figure 10: The Vortex of the Creative Canvas is Revealed 7
Figure 11: Creating a New Program in Creative Canvas 8
Figure 12: Editing the Program Information 8
Figure 13: Digital Representation of REA 9
Figure 14: Creating the Building Instructions 10
Figure 15: REA the Robot 11
Figure 16: The Creative Canva's Coding Environment 31
Figure 17: The Yellow Block Pallet 32
Figure 18: The Orange Block Pallet 32
Figure 19: The Green Block Pallet 32
Figure 20: The Purple Block Pallet 33
Figure 21: The White Block Pallet 33
Figure 22: The Light Purple Block Pallet 33
Figure 23: Recording a Sound 34
Figure 24: The Group Actions Block 34
Figure 25: Program: Moving REA 36
Figure 26: The Main Lobby Scene of the Lego Boost App 37
Figure 27: The Simple Car Activity 1 37
Figure 28: Building the Simple Car 38
Figure 29: Program - The Basic Blocks for Simple Movements 38
Figure 30: Sample Diagrams for Moving REA 38
Figure 31: Program - Loop Blocks 41
Figure 32: Program - Nested Loop 42
Figure 33: Square Example 1 - Boost Playmat 43
Figure 34: Square Example 2 - Boost Playmat 43
Figure 35: Exercise - Triangle 44
Figure 36: Exercise - 6-Side Polygon 44
Figure 37: REA the Explorer 45
Figure 38: The Color and Distance Sensor 47

Figure 39: Sending/Receiving an Infrared Signal 47
Figure 40: Program - Detecting Obstacles 49
Figure 41: Exercise - The 3-Point Turn 50
Figure 42: Detecting Colors 51
Figure 43: Detecting Colors - Distance 51
Figure 44: The Colors, which the Sensor can see: Black, Blue, Green, Yellow, Red, White 52
Figure 45: Program - Detecting Colors 55
Figure 46: REA with the Distance Sensor 62
Figure 47: Exercise - Tracking 62
Figure 48: Program - Avoiding Obstacles 63
Figure 49: Program 2 - Avoiding Obstacles 64
Figure 50: Follow the Line 69
Figure 51: Exercise - Follow the Line 69
Figure 52: Follow the Line - REAs Position 70
Figure 53: Follow the Line - Take One Measurement 70
Figure 54: Follow the Line - Take 2 Measurements on the Black Line 71
Figure 55: Follow the Line - Take 2 Measurements on the White Background 71
Figure 56: Program - Follow the Line 72
Figure 57: Program 2 - Follow the Line 73
Figure 58: Program - Detect Sound 76
Figure 59: Program 2 - Detect Colors 77
Figure 60: Program - Remote Control 80
Figure 61: Program - Joystick Controller 80
Figure 62: Program 2 - Remote Control 81
Figure 63: Exercise - Race Track 82
Figure 64: Exercise - Sumo 82
Figure 65: Geared Up REA 92
Figure 66: Gears Position 92
Figure 67: Geared Down REA 94
Figure 68: Exercise - Measure the Angle 95
Figure 69: Red and White Lego Boost 96
Figure 70: English Alphabet 97
Figure 71: Symbolic Alphabet 97
Figure 72: Program - Mathematics and Calculations 99
Figure 73: Program 2 - Mathematics and Calculations 100

Introduction

What to Expect

This handbook is a gentle introduction to the amazing world of Robotics and Coding for pupils aged 9 (nine) to 12 (twelve). Its holistic approach will allow teachers and mentors to introduce subjects in mathematics and physics and various STEM related fields with the given examples and exercises.

The first part of the book will guide you in what you need to get started, which includes the equipment and the software being used.

The second part of the handbook dives into creating and programming REA - our model robot, with detailed instructions for building her as well as going through sample code and exercises which introduce the basic principles of coding such as Sequence, Selection and Iteration as well as the use of Variables. This is done in conjunction with the use of sensors, motors and gears.

The third and final part of the handbook tackles troubleshooting issues, extra resources and bibliography.

Why Lego Boost?

Taking into account the cost, the skills required by both teachers and students, the availability and the suitability for the age group (pupils aged 9-12) the ideal kit which is currently available in the market is the Lego Boost kit.

Other kits such as the Lego WeDo were not preferred due to the cost and aging of the product. The Lego Mindstorms platform was also excluded due to its complexity and steep pricing.

What Do You Need?

A Lego Boost Kit is required in order to complete all the builds recommended by this handbook together with the Boost App, which can be found for both IOS devices such as iPads and iPhones with IOS 10.3 and newer, Android phones and tablets with Android 5.3 and newer as well as Windows 10 Devices with Bluetooth functionality.

The full list of compatible devices can be found at:

https://www.lego.com/en-us/service/device-guide/boost

The app can be downloaded for free from the Apple Store, Google Play and Microsoft Store.

Let's Speak Lego

The LEGO BOOST Kit was released by Lego in 2017. The recommended ages are 7-12 and it combines classic LEGO building with intuitive programming.

What does the LEGO® BOOST Creative Toolbox include?

- 3 BOOST bricks:
  - Move Hub
  - Color and Distance Sensor
  - Interactive Motor
- 847 bricks
- A Poster
- A Playmat

Figure 1: The Lego Boost Kit Box

Figure 2: The 3 Boost Bricks

- The Move Hub contains 2 motors with tachometers, 2 input and output ports, a 6-axis tilt sensor and a multicolored light.

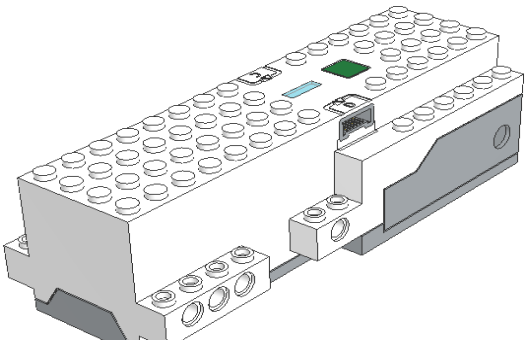


Figure 3: The Move Hub

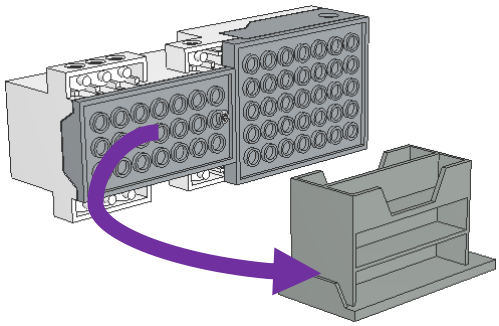
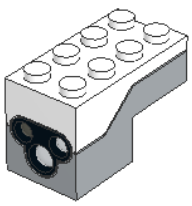


Figure 4: The Battery compartment

- The battery compartment takes 6 AAA batteries and requires a standard Philips screwdriver to open it.



- The color and distance sensor can sense both distance and different colors that are placed in front of it. It can also be used as a motion detector or just as a light.

Figure 5: The Color and Distance Sensor

- The interactive motor has the same functionality as the motor in the Move Hub, but as a detachable unit.

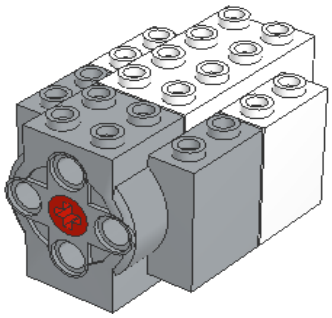


Figure 6: The Interactive Motor

Once the app is installed and launched, the user will be greeted by the Main Lobby Screen where the six main builds are located.

These are (as seen in Figure 7):

- Simple driving car (Lego has not given an official name to it)
- Vernie the Robot
- Guitar4000
- M.T.R. 4
- Frankie the Cat

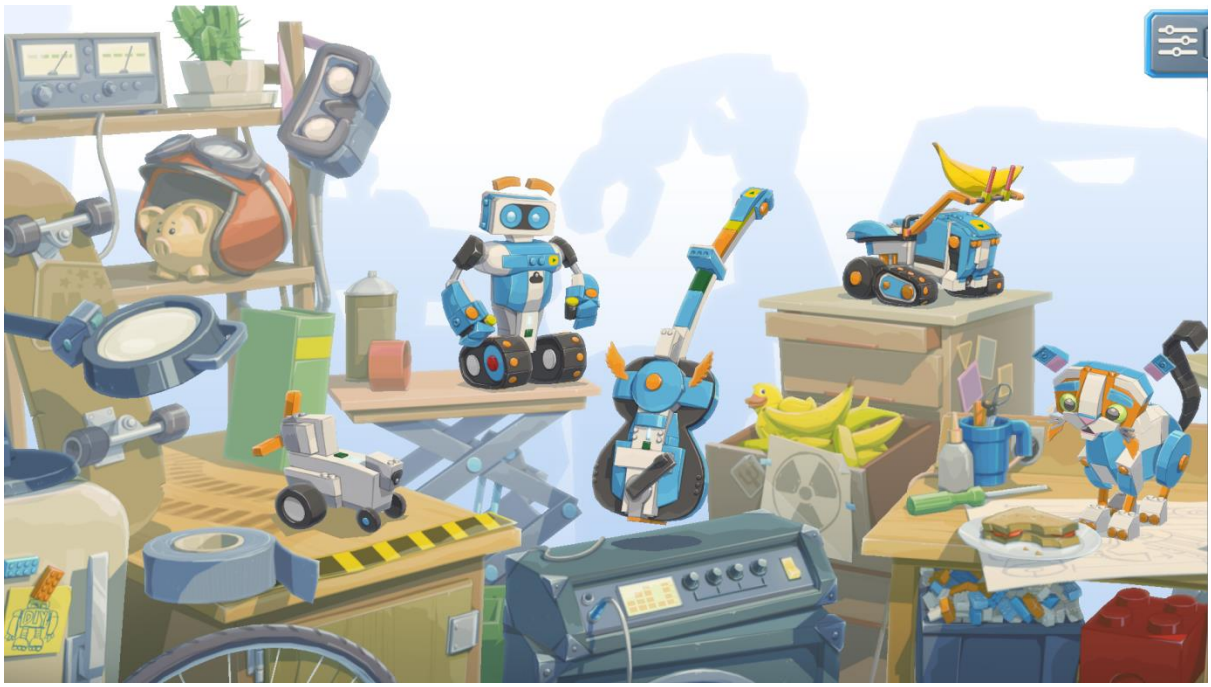


Figure 7: The Main Lobby Screen (Left)

Continuing to the right side of the Main Lobby Screen we have the AutoBuilder (as seen in Figure 8).

Following the AutoBuilder there is the Creative Canvas, which will be explained in the following section.

There are currently two more builds, which combine different Lego Kits one being the NINJAGO Kit and the other being the Lego CITY Kit, which are sold separately.

Bear in mind that this app is regularly being updated by LEGO and more functionalities and additional builds are being added from time to time.

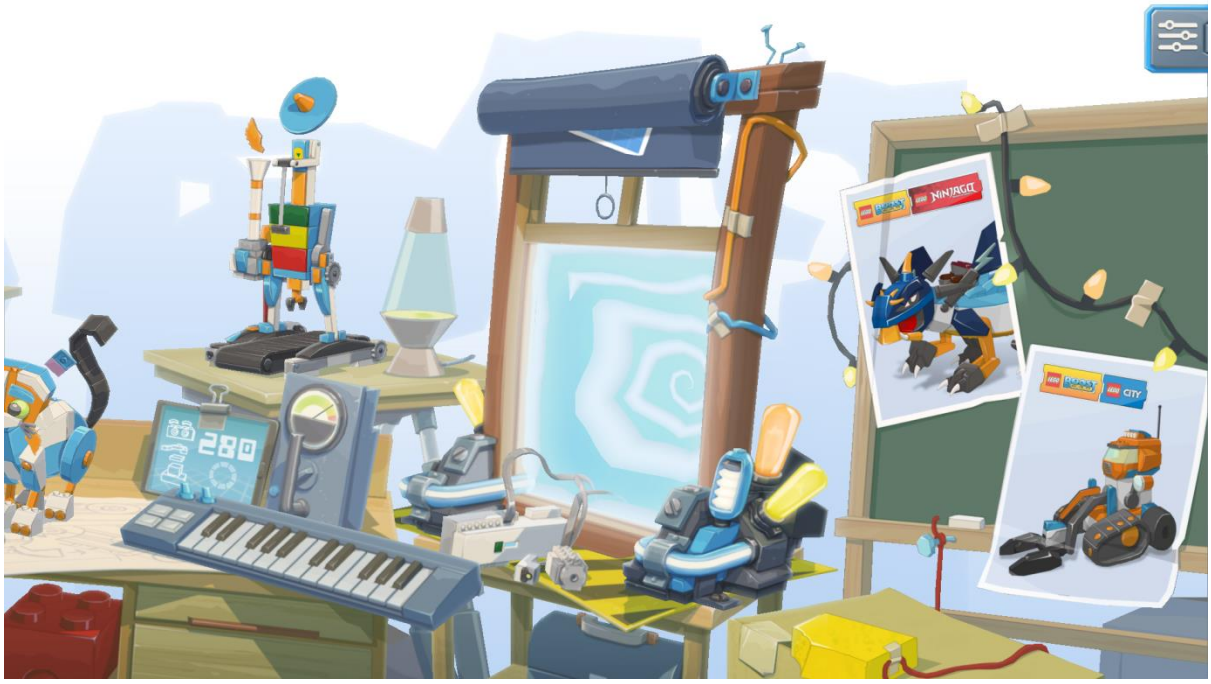


Figure 8: The Main Lobby Screen (Right)

This is the Lego Canvas

When building and programming the recommended Lego builds the programming blocks are specific and sometimes limited to the capabilities of the particular build. In order to have access to the whole range of the programming blocks for the learner to program their own builds the Creative Canvas has to be used.

The first step is to click on the curtain in order for the Creative Canvas to become visible.

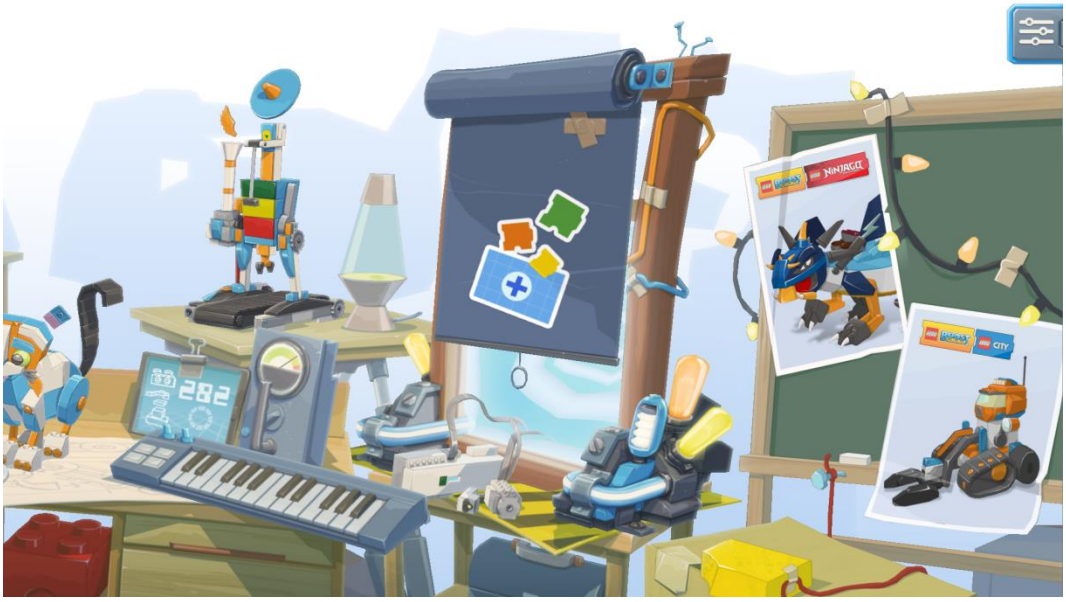


Figure 9: The Curtain Covering the Creative Canvas

Once the curtain is clicked, the Creative Canvas is revealed.

Figure 10: The Vortex of the Creative Canvas is Revealed

By clicking on the Creative Canvas, the following screen appears where the learner can create a program by clicking on the Plus sign.

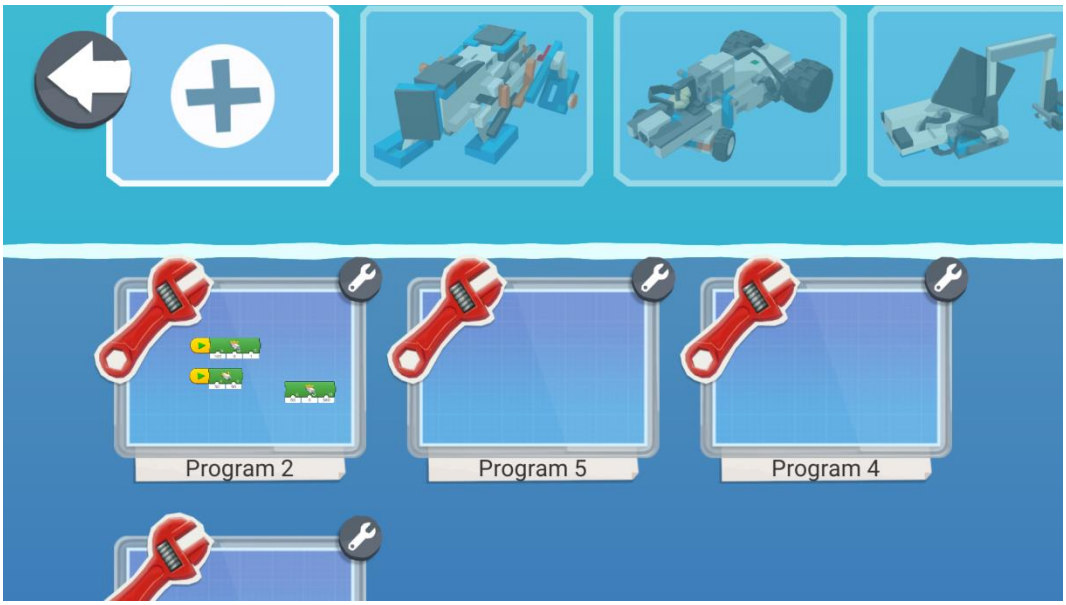


Figure 11: Creating a New Program in Creative Canvas

By clicking on the wrench icon on the right of the newly created program, the user can change the name of the program, delete it or create a duplicated copy of the program and even change the avatar of the program (image on the top left side of the program).

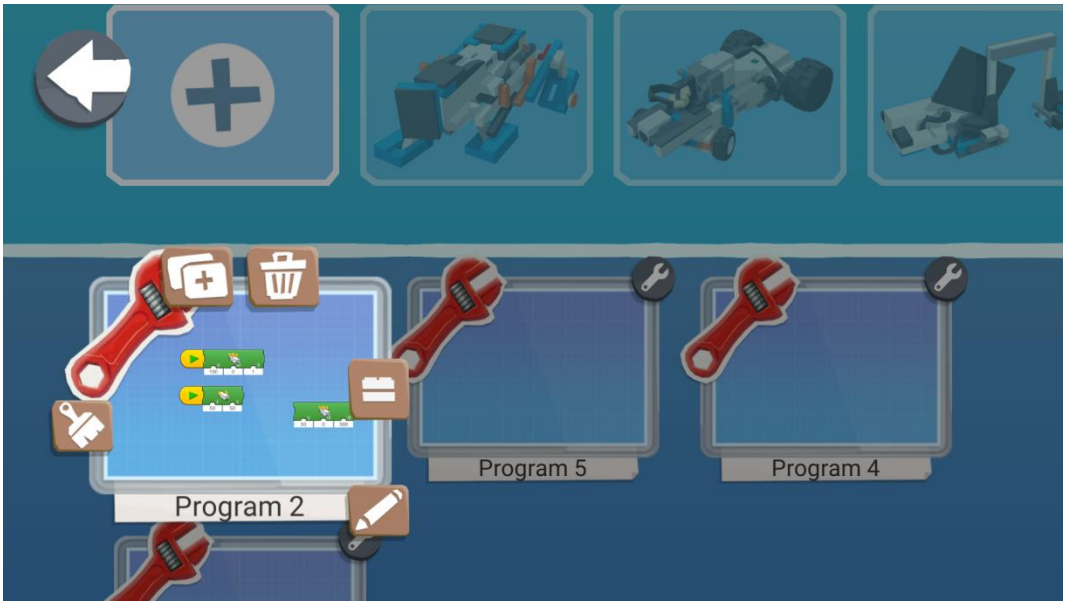


Figure 12: Editing the Program Information

Section A - Basic Robotics Movements

The aim of this section is to introduce students to the basic movements of robots. Students will build the robot REA and learn to program it to perform basic movements. They will also learn how to use loops to program it to make repetitive movements.

REA - Robotics for Educational Applicability

1.1 Let's Build REA

REA went through different stages of development and each built was tested and improved until it reached its current structure. The final form was digitally designed using the LeoCAD version 19.01 software. Students and teachers can design their own robot digitally without using any actual Lego blocks. Instead they can have at their disposal the full library of digital representations of all the blocks.

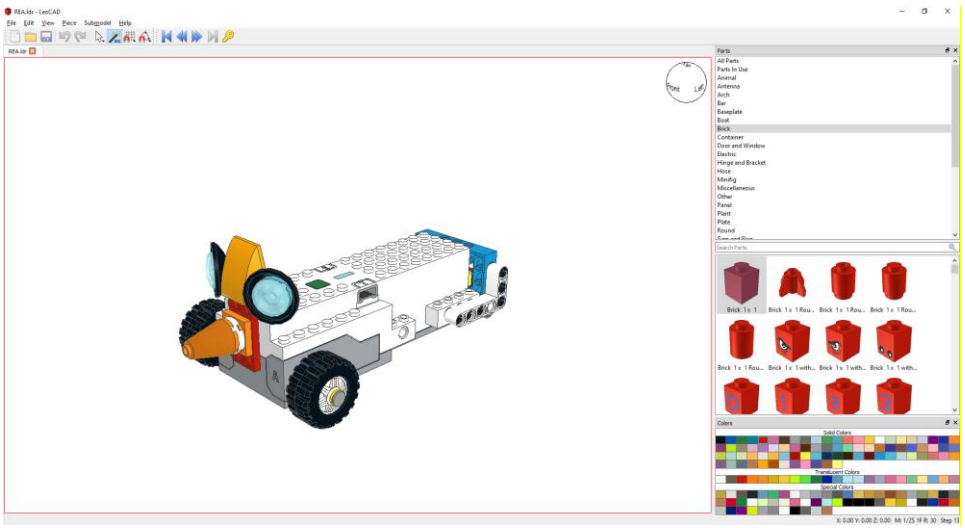


Figure 13: Digital Representation of REA

The instructions for building REA and the various add-ons which are shown in this handbook where created using the LPub3D version 2.3.12.0.1356 software. Teachers can create instructions of their own builds and share them with their students in order to recreate them.

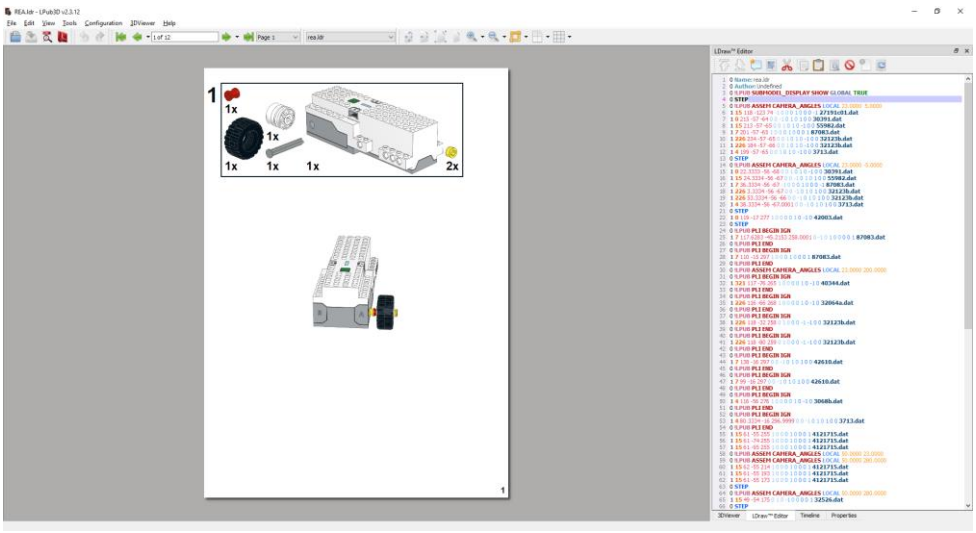


Figure 14: Creating the Building Instructions

Building REA

Prerequisites: A complete Lego Kit is required in order to complete the built of REA.

This section presents a step-by-step guide including figures on how to build the REA robot using the available Lego bricks of the Lego BOOST kit.

REA Robot is the robot to be used in order to learn how to use the three electronic components of LEGO BOOST Kit, which are:

1. the Move Hub,
2. the Color/Distance/Motion Sensor and
3. the External Motor

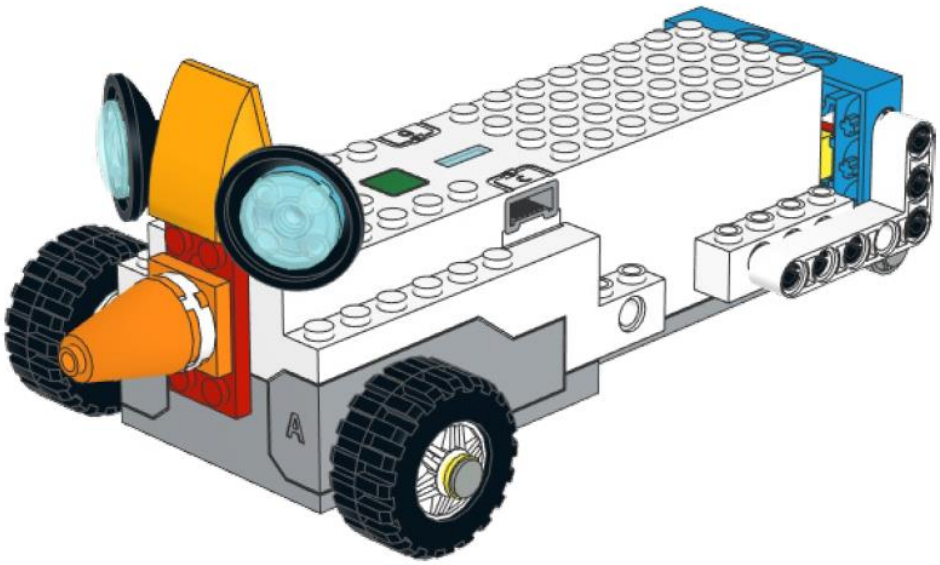
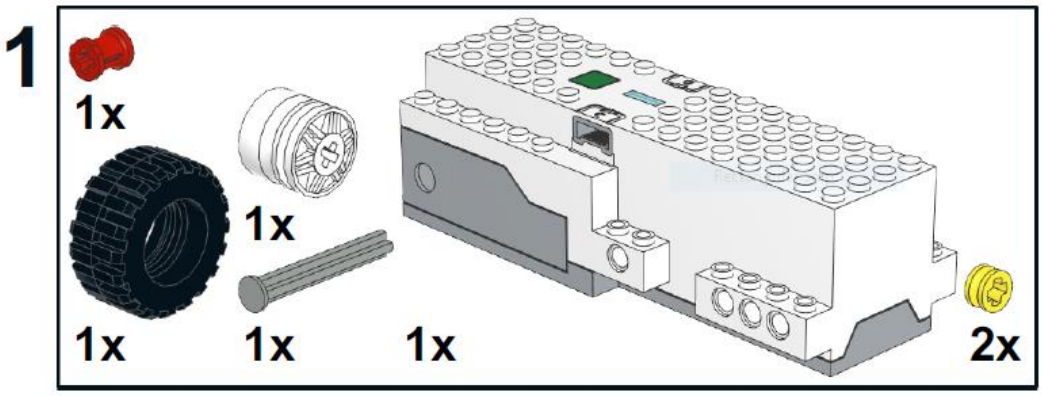
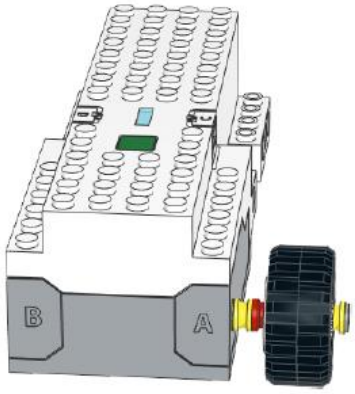
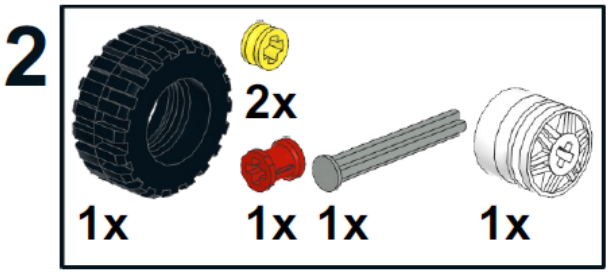
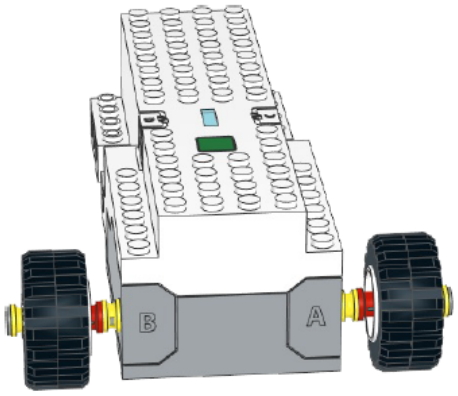


Figure 15: REA the Robot

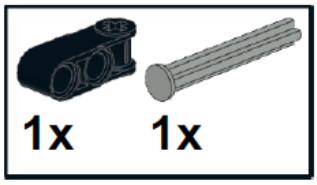

1
1x
1x
1x
1x
1x
2x

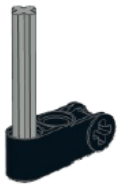

B
A

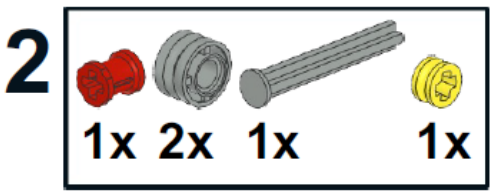

2
2x
1x
1x
1x
1x

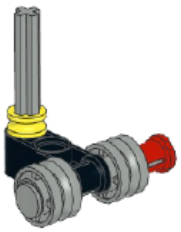

B
A

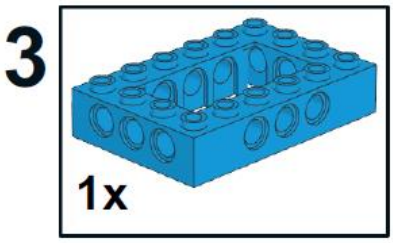
1

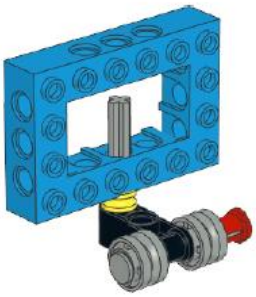


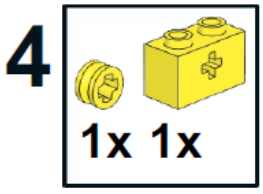


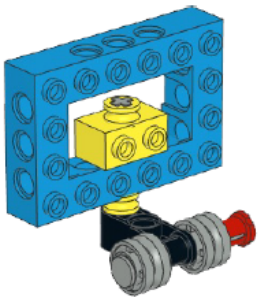

2
1x 2x 1x 1x

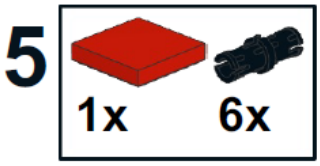


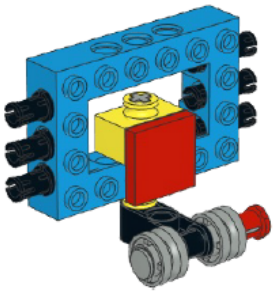

3
1x



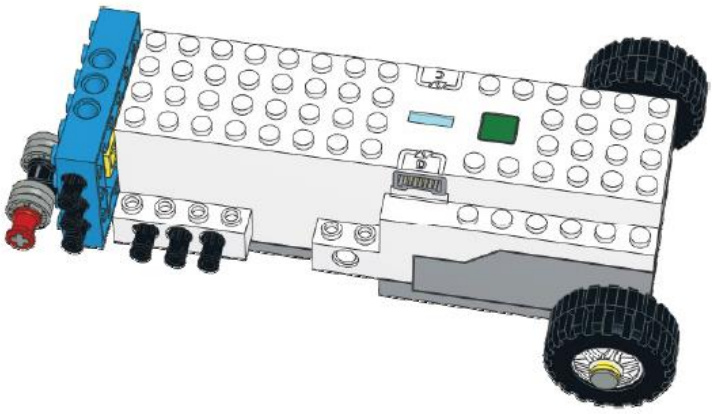

4
1x 1x

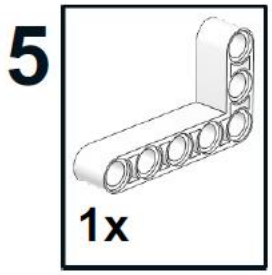


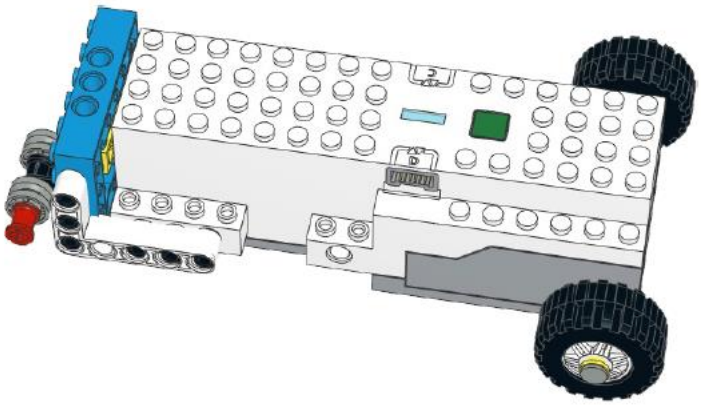

5
1x
6x

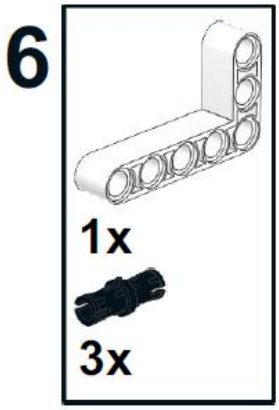


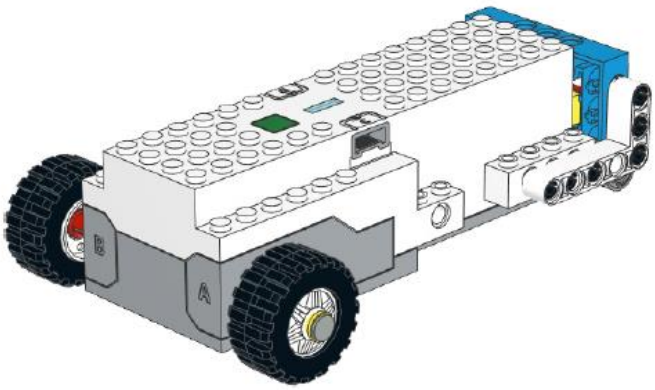
4
6x

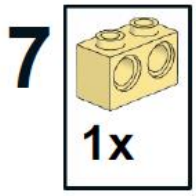


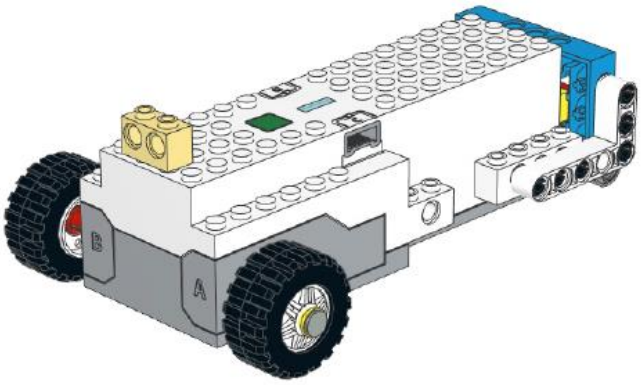

5
1x



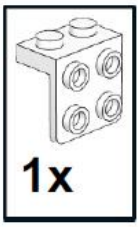

6
1x
3x

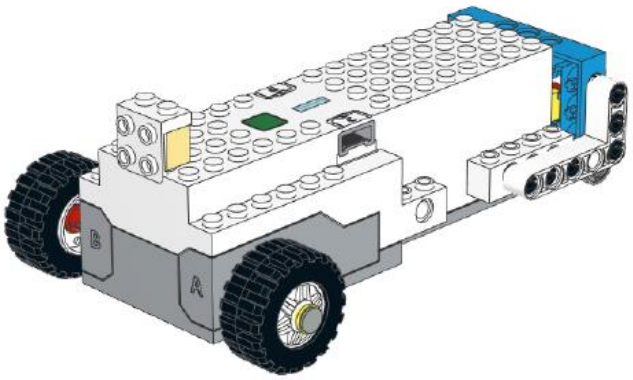


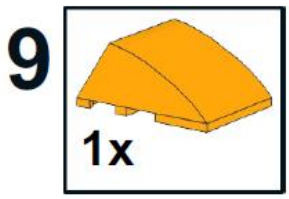

7
1x

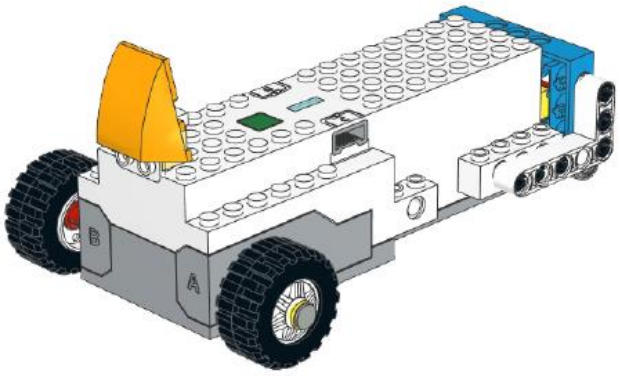


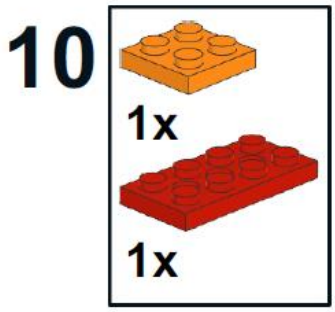
8

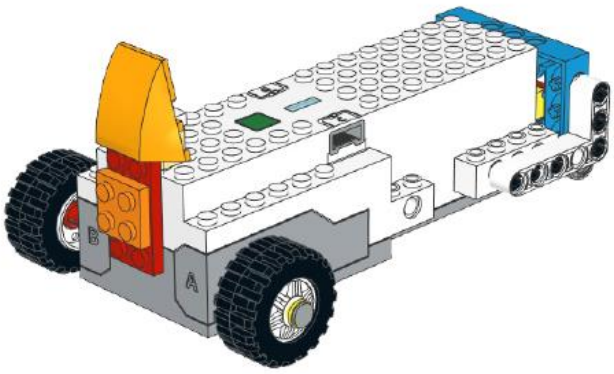




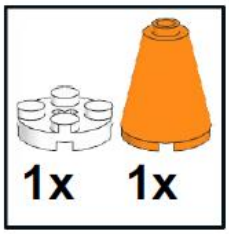

9
1x

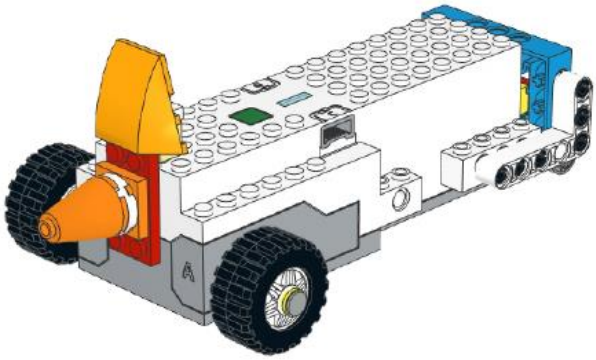

B
A


10
1x
1x

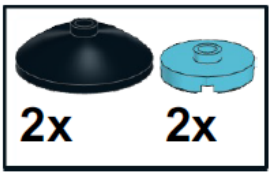


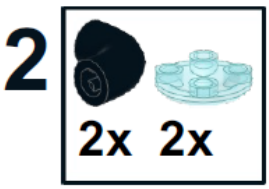
11



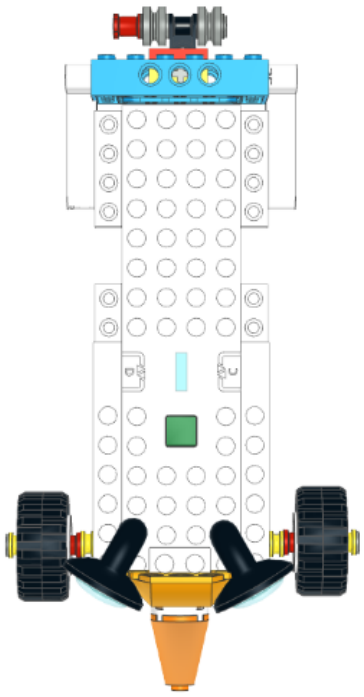


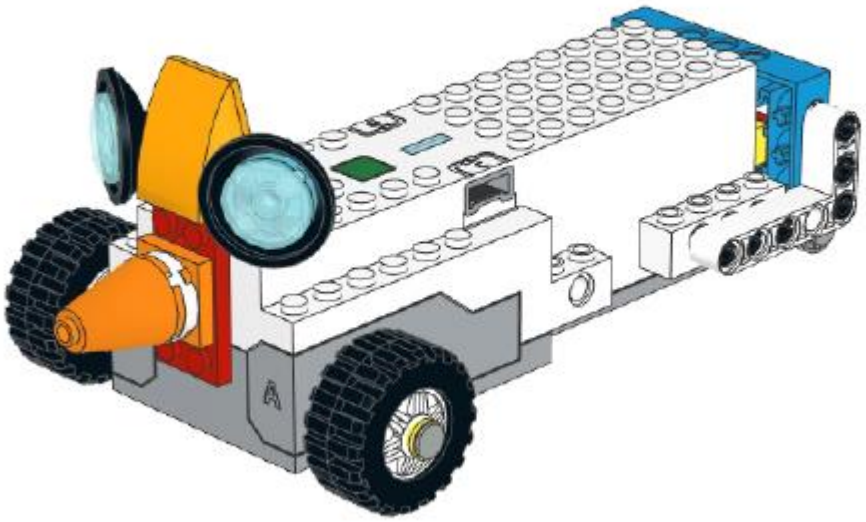
1




2
2x 2x

12




A

1.2 Moving Instructions of REA

Prerequisites: A constructed model of the REA Robot.

Coding REA

The Coding Environment

This section presents the LEGO BOOST Creative Canvas coding environment and the coding blocks of canvas in order to learn how to use each set of blocks for developing a programme using the LEGO BOOST app.

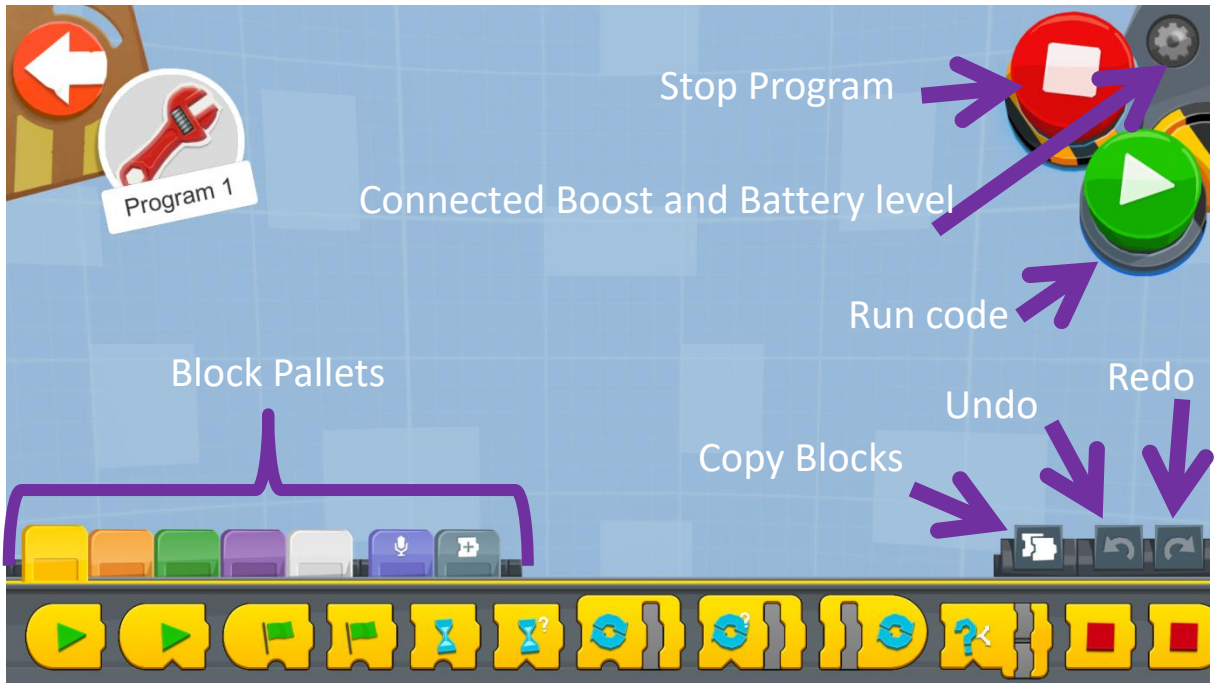


Figure 16: The Creative Canva's Coding Environment

The Block Pallets

- **Yellow** blocks control the flow of the commands
- They can be used to start a program, stop a program, pause a program or even loop a group of commands

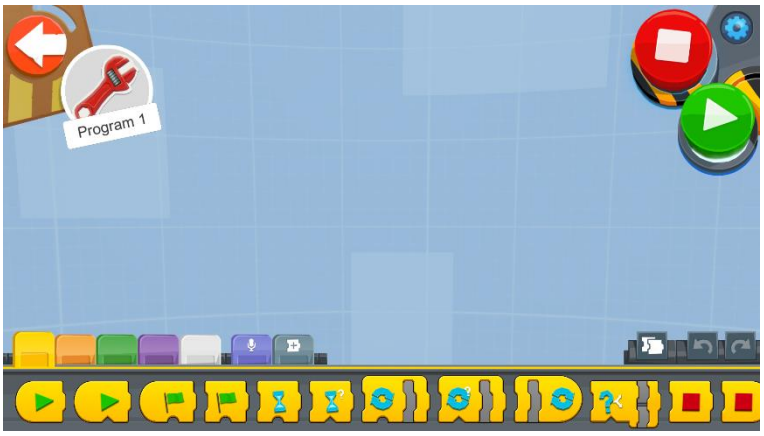


Figure 17: The Yellow Block Pallet

- **Orange** blocks work with the Color and Distance Sensor and the Move Hub's Tilt Sensor
- These blocks will prompt an action when a sensor gets triggered

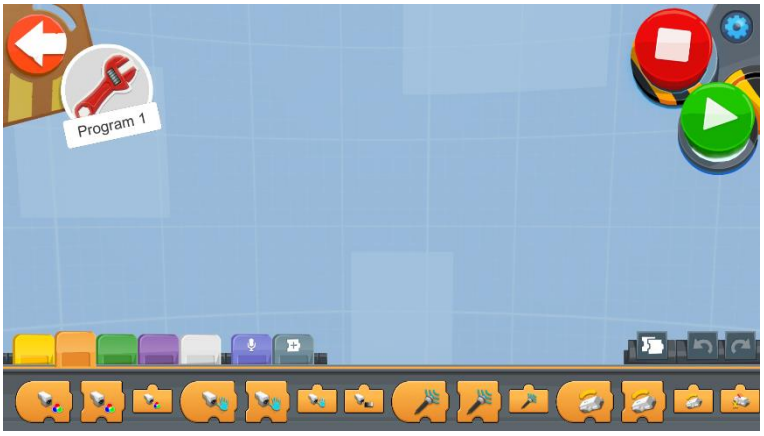


Figure 18: The Orange Block Pallet

- **Green** blocks are used for movement
- These blocks can be used in order to control the **Speed, direction and duration** of the Move Hub motors and the single interactive motor

Figure 19: The Green Block Pallet

- **Purple** blocks can **play sounds** through the user's device speakers
- They can also change the **color of the lights** on the Move Hub & Color and Distance Sensor

Figure 20: The Purple Block Pallet

- **White** blocks let you do some complex programming using 'variables' and 'constants'
- They can also be used when performing various **mathematical calculations**, creating **logic expressions** and generating **random numbers**

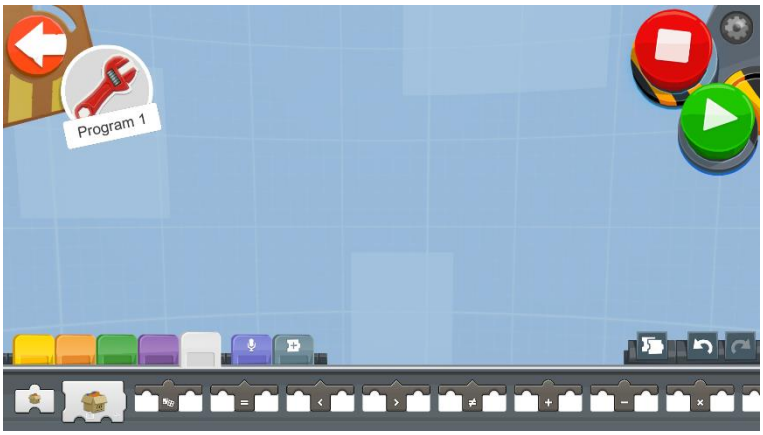


Figure 21: The White Block Pallet

- **Light Purple** blocks let you record a sound for the robot to repeat

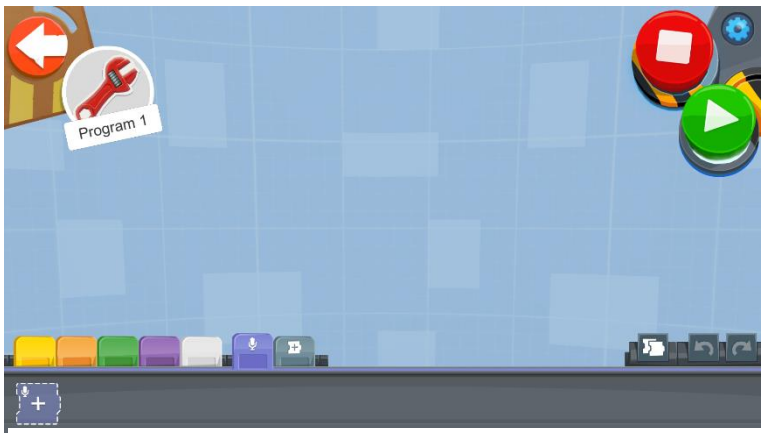


Figure 22: The Light Purple Block Pallet

- You can also add some cool sound voice filters

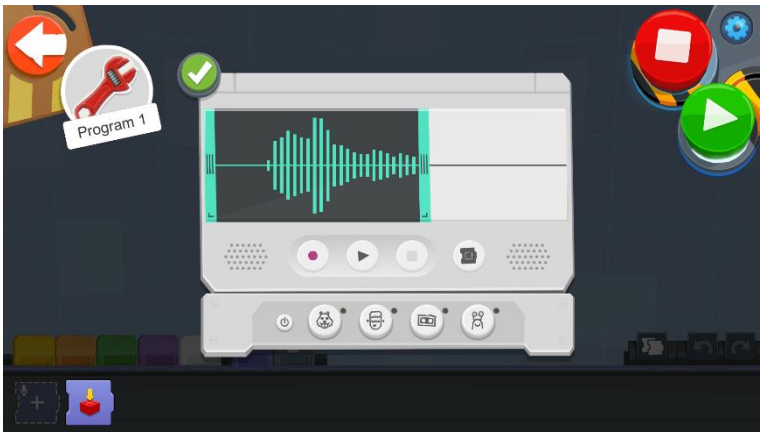


Figure 23: Recording a Sound

- **Group Actions** block let you group as one block button several actions
- This is the equivalent of creating sub-programs

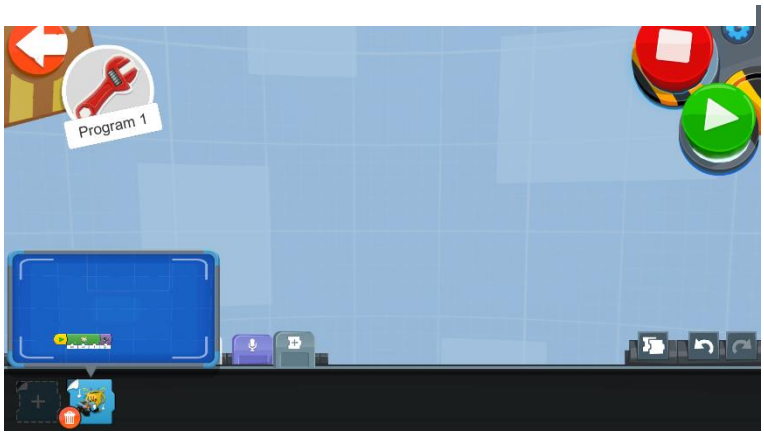


Figure 24: The Group Actions Block

Moving REA

In order to begin coding REA, the first block, which will be used is the **Start Block** located in the Yellow Pallet

There are various blocks which control the movement of the Move Hub. Here are the most common ones, which can be found in the Green Pallet.

Drivebase Move Tank for Duration - Motor A speed (-100..100) and Motor B speed (-100..100) for duration (in seconds)

Drivebase Move Tank for Distance - Motor A speed (-100..100) and Motor B speed (-100..100) for distance (in degrees)

Drivebase Move Steering for Duration - Both motors speed (-100..100) and steering direction (-100..100) for duration (in seconds)

Drivebase Move Steering for Distance - Both motors speed (-100..100) and steering direction (-100..100) for distance (in degrees)

Drivebase Move Tank - Motor A speed (-100..100) and Motor B speed (-100..100)

Drivebase Move Steering for Distance - Both motors speed(-100..100) and steering direction (-100..100) and steering direction (-100..100)

Sample Programs of Moving Blocks

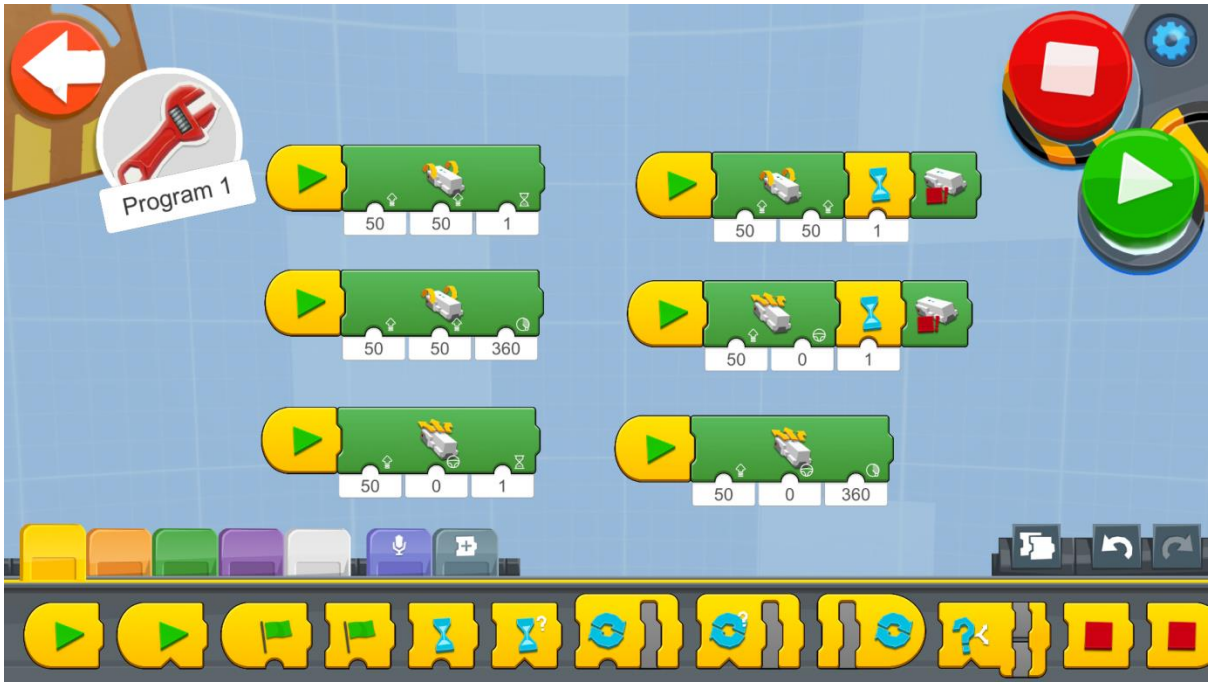


Figure 25: Program: Moving REA

Try using the different blocks for moving like the example program above.

Create a new project on the Creative Canvas and run each block of code and notice the differences and similarities.

Note that the moving blocks which do not set the time interval or the amount of degrees (rotations) of the movement will continue to turn the motors indefinitely unless a wait block is added, which sets the time interval for the motors to work followed by the stop motors block.

Note: The learner can use a more simplified version of the movement blocks if they have successfully completed/unlocked the first simple drive car Code activity or the first of Vernie's Code activities.

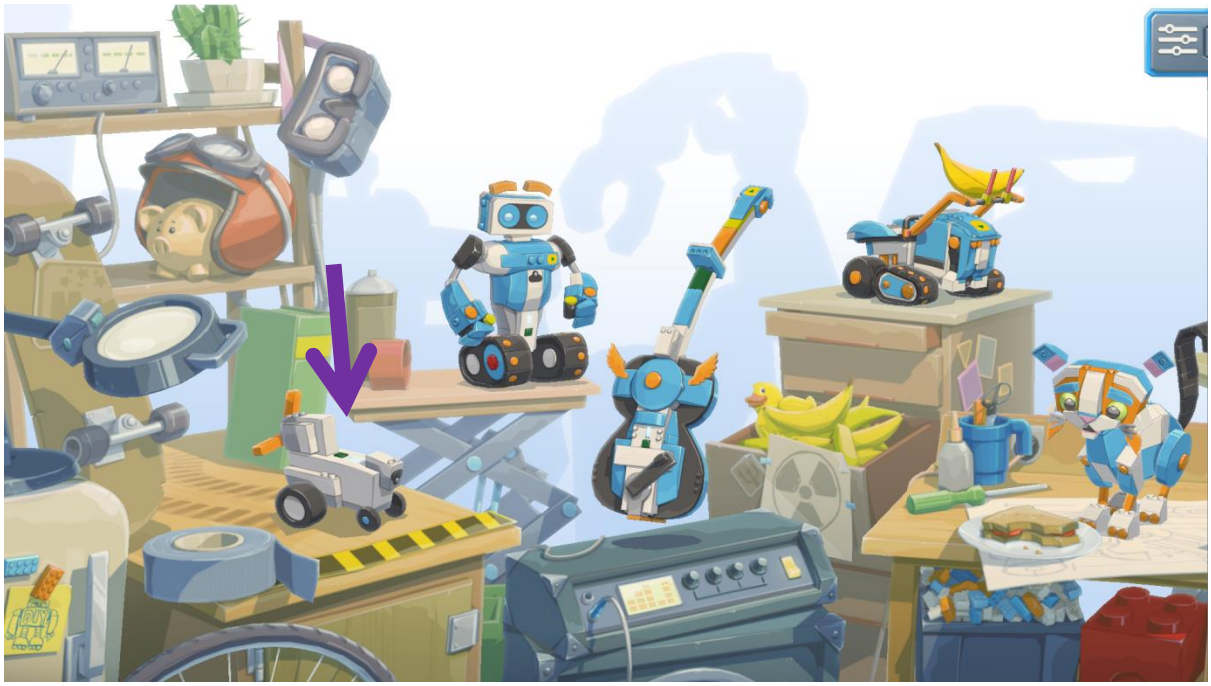


Figure 26: The Main Lobby Scene of the Lego Boost App

Step 1: Navigate to the Main Lobby Scene and click on the Simple Drive Car Icon

Figure 27: The Simple Car Activity 1

Step 2: Click on the First Activity

HINT for the teacher: The programs can be unlocked by pressing a specific Lego Boost Code activity for five seconds.

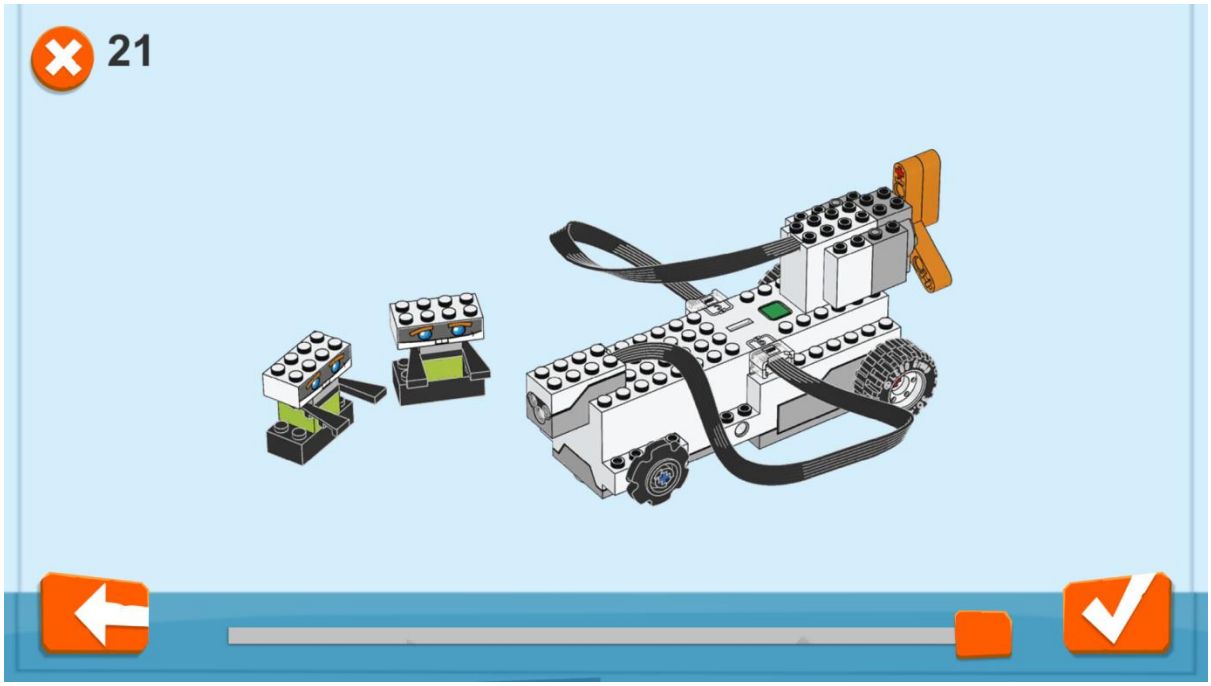


Figure 28: Building the Simple Car

Step 3: if it is the first time that you open this specific activity, the schematics for building the car appear. You can just scroll to the last page and click on the tick icon.

Figure 29: Program - The Basic Blocks for Simple Movements

Step 4: Use the simple move blocks to move REA. How far does she move when one straight arrow block is used? Does it move in the expected direction?

Note for the teacher: The driving motors for REA are in the front of the robot, whereas on the Simple Drive Car they are situated in the back of the robot.

Example Exercise Sheet

The race for Planet Mars is on! You are required to build and test REA, a robot that is capable of following a set of commands to explore the surface of the red planet.

Before we send REA into space, we must first test her thoroughly here on planet earth. Run the following experiments and observe how REA behaves. For these experiments, a ruler will be needed for measuring distances. Do not move to the next experiment until your teacher has seen your current experiment.

Level 1 Exercises:

1. Drive REA forward rotating the wheels by 2 degrees. How far did your robot travel?

2. Drive REA forward for 2 seconds. How far did your robot travel?

3. How many degrees will be needed for REA to drive its wheels for one full rotation?

4. Drive forward for 2 rotations of the wheels. How far did your robot travel?

Level 2 Exercises:

1. What is the circumference of the robots wheel? (Hint: You will need to measure the diameter of the wheel)

2. How far will REA drive if the wheels turn 3 rotations? Do the calculations!

3. Now program REA to move 3 rotations and measure how far it goes. Does it go as far as you expected?

4. Program REA to drive forward for 5 rotations slowly and then 1800 degrees backwards as fast as possible.

5. Make REA turn around a complete circle (360 degrees). What happened? How far did REA turn if you type in 360°?

6. How many degrees of the wheel does REA need to turn a complete circle? (Hint: Keep experimenting until it is perfect!)

Level 3 Exercises:

1. Drive REA forward for 50 cm, turn around 180° and drive back to where you started.

   What should the duration be to go forward 50 cm?

   (Hint: Have a look at the circumference of your wheel. This will tell you how far your robot goes in 1 rotation).

2. Make your robot drive in a 'figure of 8'.

   (Hint: Create a diagram first like the examples below before you start programming. Don't forget to mark your starting point so each effort starts exactly at the same point!)

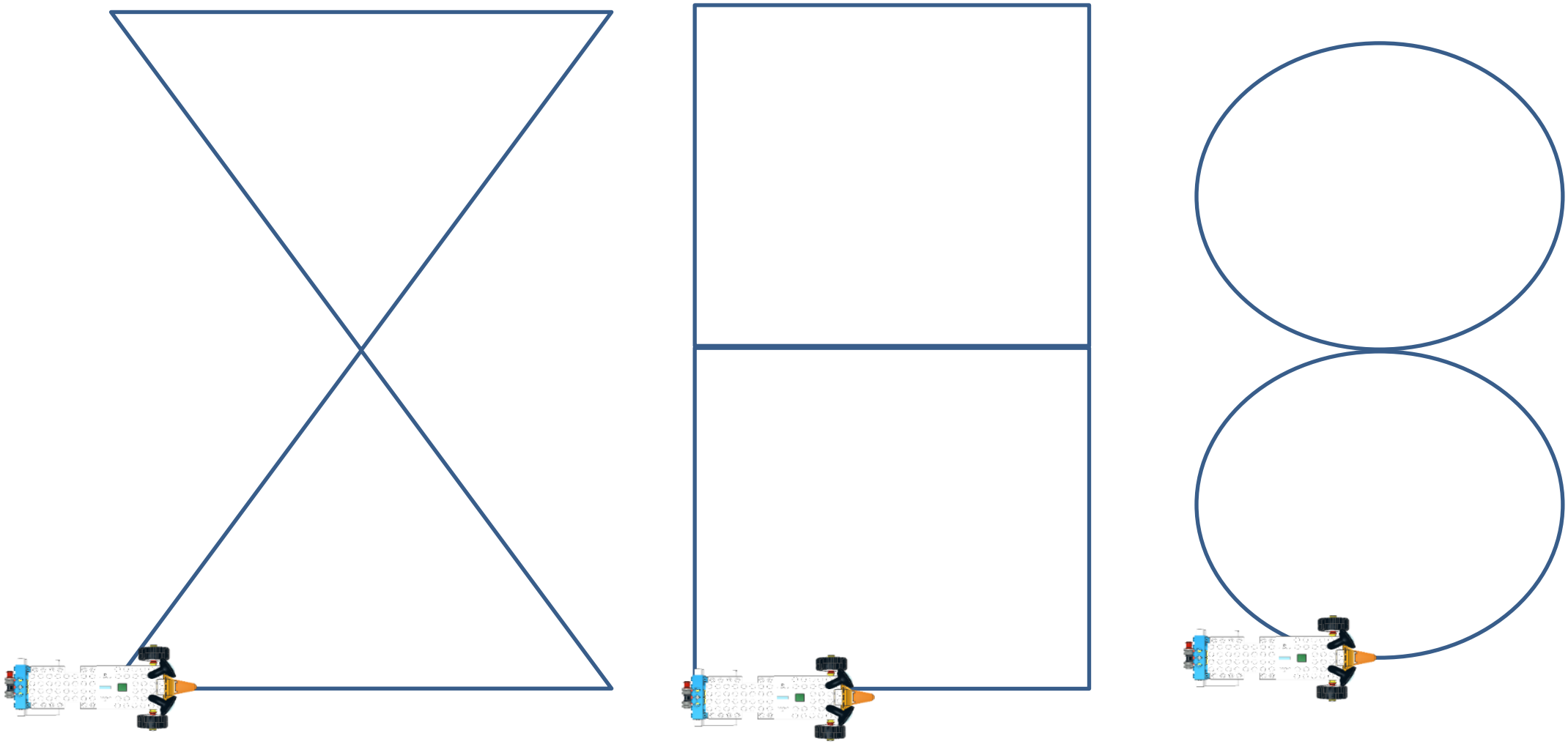


Figure 30: Sample Diagrams for Moving REA

1.3 Using Loop Commands with REA

Prerequisites: A basic understanding of the coding blocks used for movement.

The Three Types of Loop Blocks

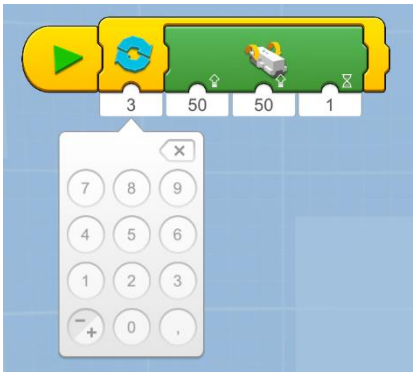
Loops allow REA to easily repeat blocks of code without having to put blocks of code again and again. There are different types of loops in coding.

"**For Loops**" let programmers repeat code a specific number of times.

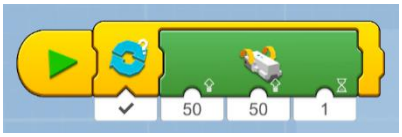
"**While Loops**" require a condition to be true in order to repeat code.

"**Forever Loops**" repeat code forever and should be avoided when not necessary because they can consume all the resources of our computer and run out the batteries of our robot.

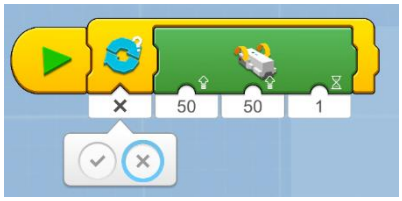
The **Loop For Count**: Loops the enclosed blocks of code for a specific number of times which can be set by the user



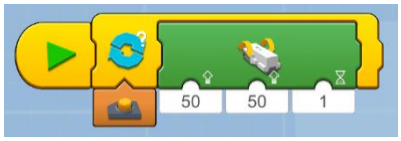
The **Loop While True**: Loops the enclosed blocks of code while a condition is true.



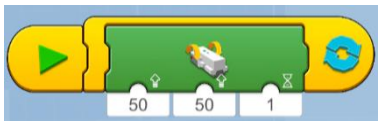
The user can set whether the loop will be executed if the condition is true or false.



Alternatively, the loop can be triggered when an action is performed, like the push of a button.



The **Loop Forever**: Loops the enclosed blocks of code forever (until the batteries run out).



Sample Programs of Loop Blocks

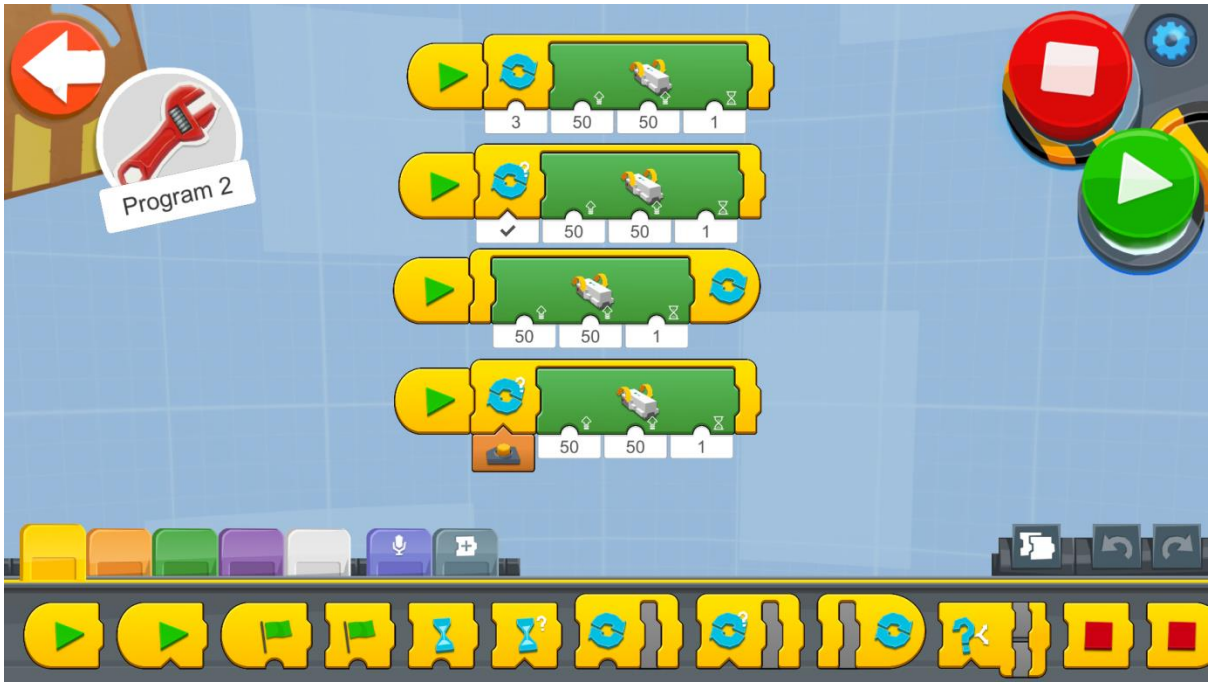


Figure 31: Program - Loop Blocks

Try using the different blocks for Looping commands like the example program above.

Create a new project on the Creative Canvas and run each block of code and notice the differences and similarities.

Notice that the last block of code is not working properly. Try and make it work so when a button is pressed the Loop is activated (the loop condition becomes true).

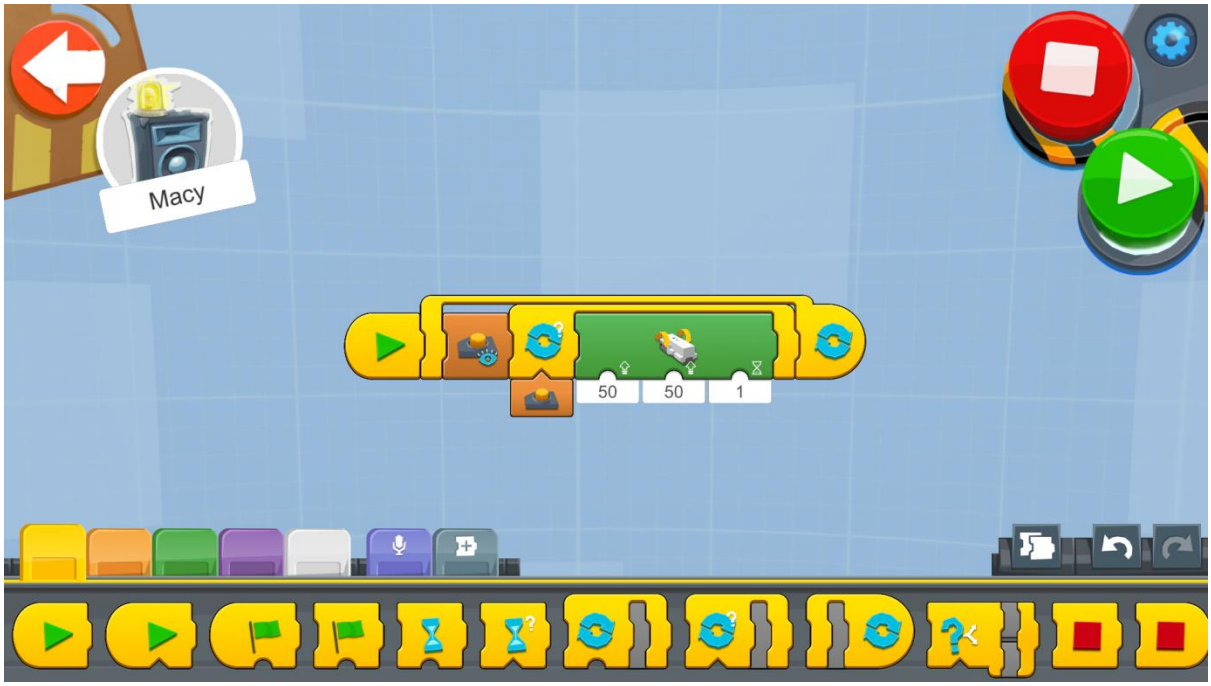


Figure 32: Program - Nested Loop

The solution is to insert another Loop Forever loop and add the button appearance block outside the Loop While True block. This is called a nested loop (a loop inside a loop).

Example Exercise Sheet

Level 1 Exercises:

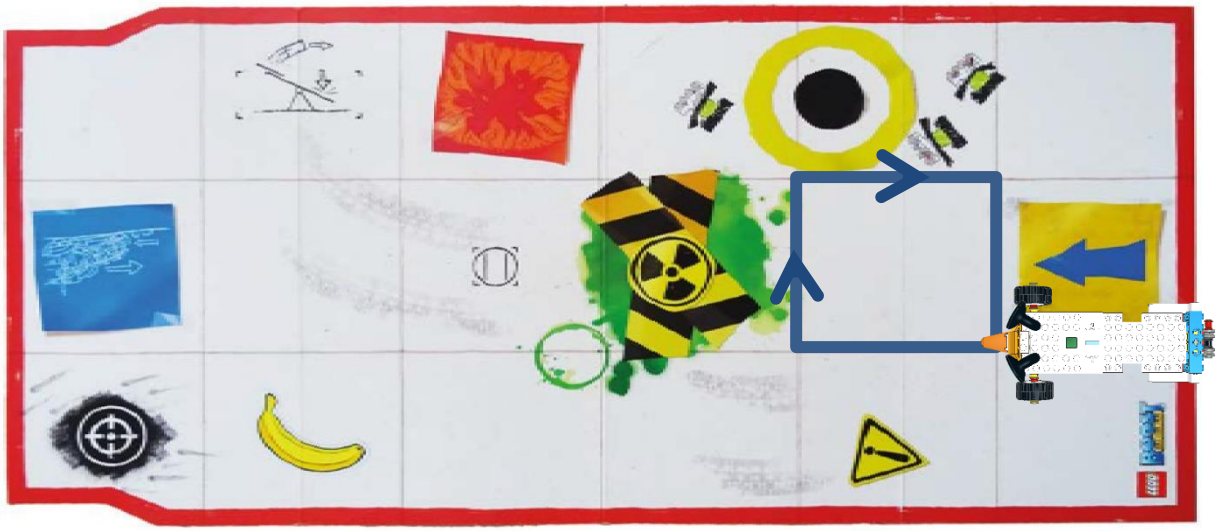


Figure 33: Square Example 1 - Boost Playmat

1. Using the Boost playmat drive in a square (Figure 33). Do not forget to use a loop command!

How many degrees are needed to drive one side of the square?

How many degrees are needed to make the 90° angle turn?

2. Using the Boost playmat to drive in a larger square 2x2 (Figure 34). Do not forget to use a loop command!

Figure 34: Square Example 2 - Boost Playmat

How many degrees are needed to drive one side of the square?

How many degrees are needed to make the 90° angle turn?

Level 2 Exercises:

1. Take a large piece of paper and a marker and draw an equilateral (a triangle with all sides the same length). Using a ruler make each side of the triangle 16cm long (Figure 35). How many degrees will be needed for the Move block to make REA turn correctly, and how many times should the loop repeat?

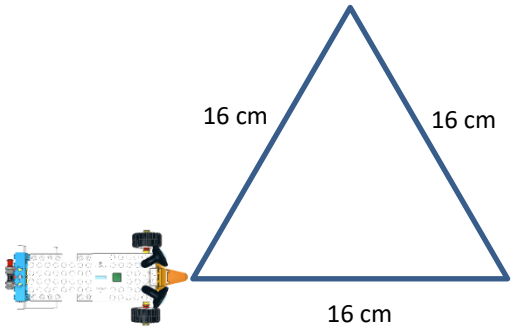


Figure 35: Exercise - Triangle

Level 3 Exercises:

1. Take a large piece of paper and a marker and draw a hexagon (a six-sided polygon). Using a ruler make each side of the hexagon 16cm long. How many degrees will be needed for the Move block to make REA turn correctly, and how many times should the loop repeat?

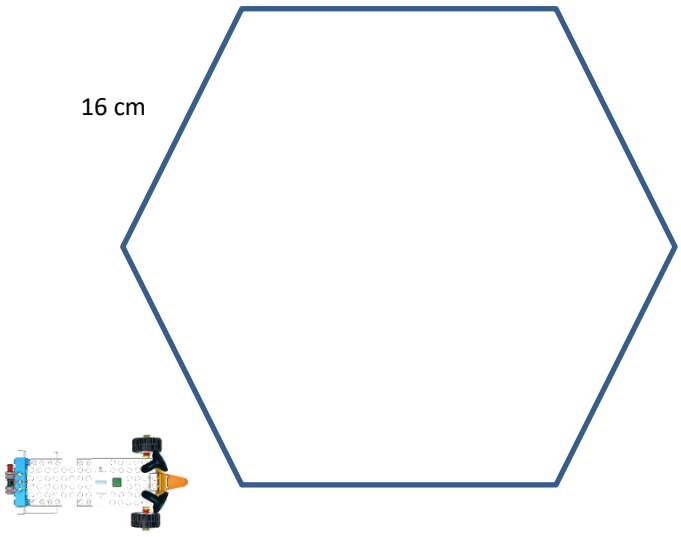


Figure 36: Exercise - 6-Side Polygon

Section B - Robotics Sensors

In this section, students will learn how to use environmental data (via sensors) to program the robot. They will learn how the sensors that the robot is equipped with work, while they will also program it to perform tasks using the sensors.

1.4 Using Sensors with REA

Prerequisites: A basic understanding of the coding blocks used for iteration - loops.

Sensors

In this section, we will see how we can incorporate a sensor to REA and how we can utilise this sensor in order to detect obstacles, various colors, the intensity of light as well as the distance from an object.

Building the Sensor

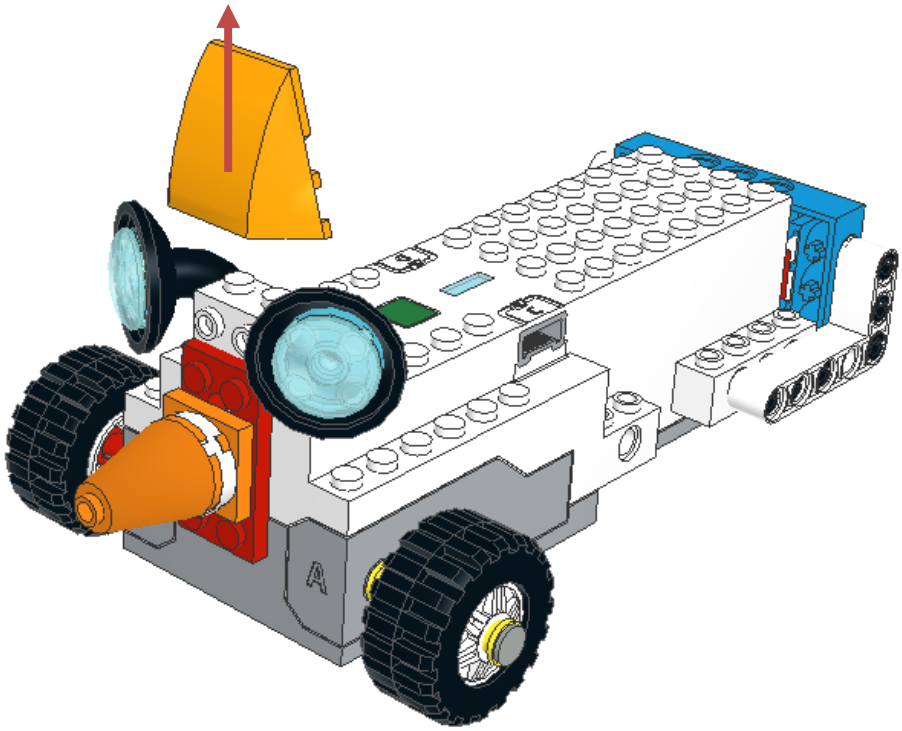
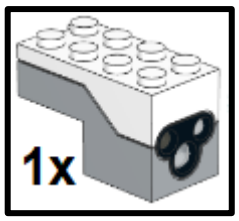
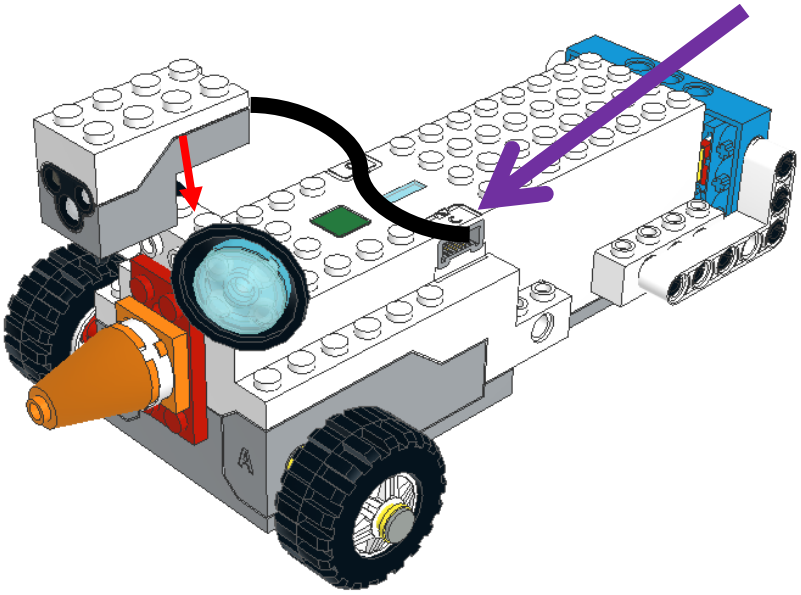
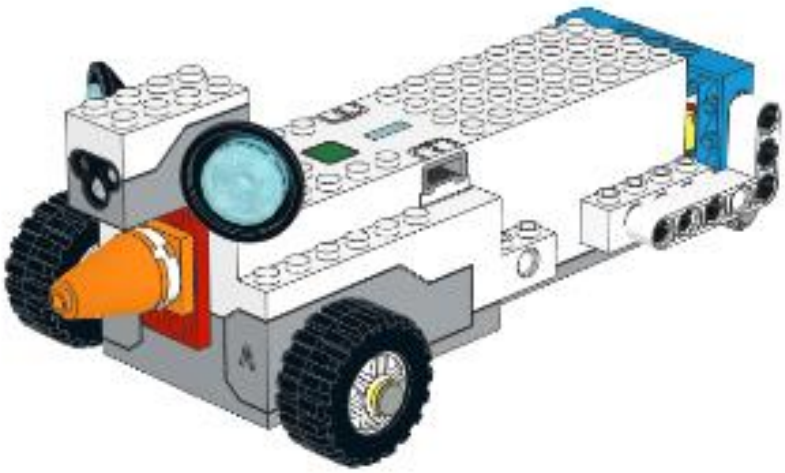


Figure 37: REA the Explorer



Connect Sensor to Port C





The Lego Boost Color and Distance Sensor

A sensor is a device, which detects events or changes in the surrounding environment and sends the information to a computing device in order for it to process it and take actions. There are many examples of sensor usage in our everyday life. From a motion sensor, which detects movement and turns on the lights or opens a sliding door to temperature sensors, which detect the temperature and switch on heaters and air conditions. Sensors are also used in most types of remote-controlled devices such as Televisions and Air Conditions.

The LEGO BOOST Color and Distance sensor as shown in Figure 38 is used for:

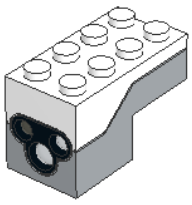
- Sensing distance - How far an object-obstacle is situated from the sensor.
- Sensing color - it can detect specific colors (Black, Blue, Green, Yellow, Red, White) as well as the absence of a color.
- Detecting movement - it can be used as a motion detector
- Detecting light - It can detect the light level reflected on the sensor measuring from 0 - darkest to 10 - lightest.

Figure 38: The Color and Distance Sensor

Additionally, the Color and Distance sensor can emit different colors (Red, Green, Blue, and a combination of the three).

Detecting Objects

How the Sensor Works



The motion sensor part of the Lego BOOST set is an active infrared - IR sensor. This means that one of the eyes of the sensor transmits an infrared signal, which bounces off an object/obstacle and is detected by the sensor (See Figure 39). This infrared technology is used in a wide variety of devices such as in most remote-controlled devices like Televisions, Air Conditions, and even surveillance systems.

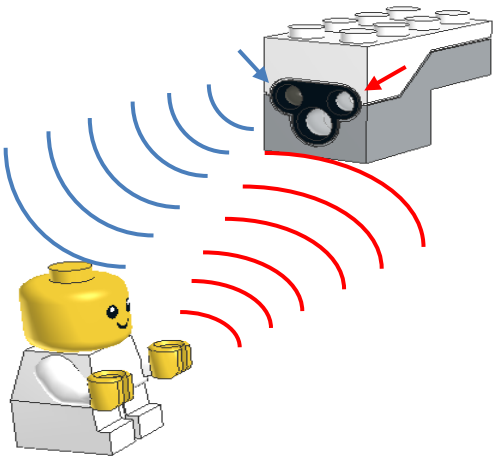


Figure 39: Sending/Receiving an Infrared Signal

The Detecting Objects Blocks

Trigger on Distance Block - Triggers when the distance measured by the sensor is less than the distance indicated by the value under it. When triggered it executes the sequence of code which follows. It can take values from 0 to 10.

Sensor Distance Reporter - Displays in real time the current distance measured by the sensor. In order to be used in a program it needs to be attached to the bottom of other blocks. It can take values from 0 to 10.

Wait for Distance - Waits for the distance measured by the sensor to be less than the distance indicated by the value under it. When an object is not closer than the value indicated, the program stays paused and when the condition is met the program continues to the following sequence of instructions. It can take values from 0 to 10.

Sample Programs of Detecting Obstacles

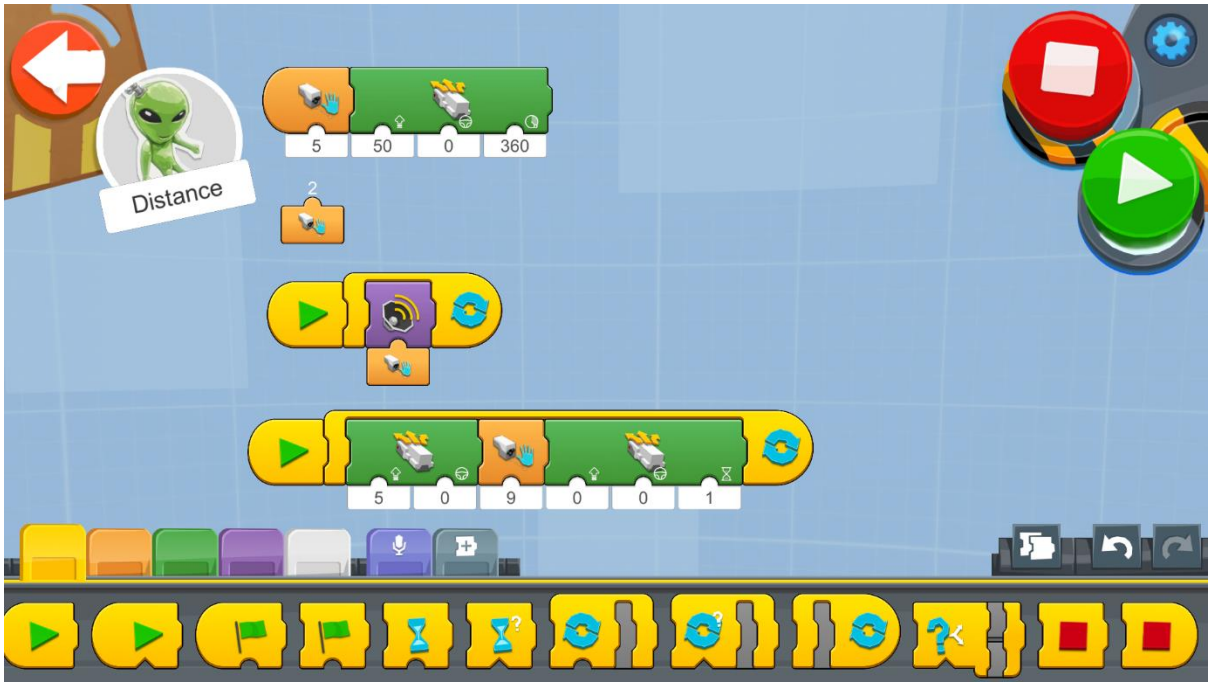


Figure 40: Program - Detecting Obstacles

Try using the different blocks for Detecting objects like the example program above.

Create a new project on the Creative Canvas and run each block of code (For Android and iOS app you can find the blocks on the Creative Canvas of Vernie).

1. The first program activates when an object is less than distance 5 in front of REA. REA will then move 360 degrees forward at speed 50.
2. The second block just indicates in real time the distance from an object in front of the sensor.
3. The third program takes the distance and uses it as an input for selecting and playing different sounds. Notice that when the distance from an object changes the sound also changes.
4. In the fourth program REA starts moving at speed 5 forward and when it detects an object in distance less than 9 it stops for a second. It then tries to move again.

Example Exercise Sheet

Level 1 Exercises:

1. Take a ruler and measure the distance of an object from the LEGO BOOST Color and Distance sensor. Use the **Sensor Distance Reporter** block and note down the corresponding distance. Try and find what the distance 1 to 10 means in the real world in centimetres.
2. Move the sensor at the back of REA. Try and do a 3-point turn with an obstacle at the back. The following schematic will give you an idea on how this problem can be solved.

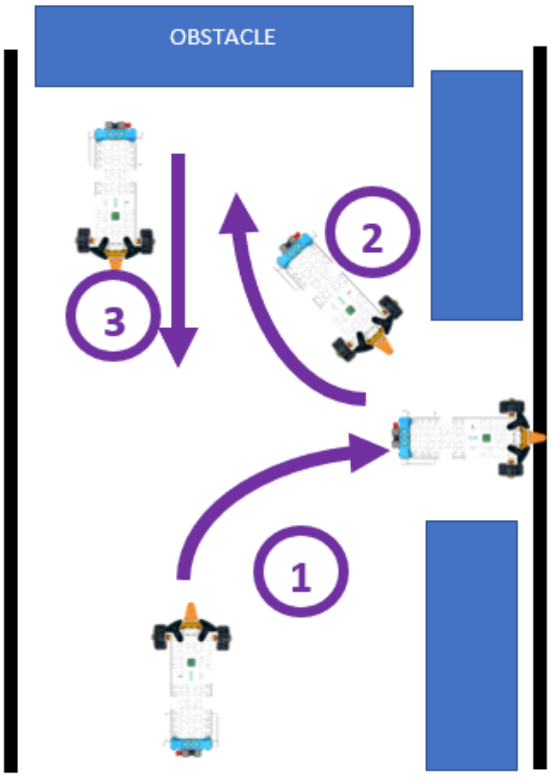


Figure 41: Exercise - The 3-Point Turn

Level 2 Exercises:

1. Create a straight track with obstacles. You can use objects found in the classroom. REA should drive forward and when an obstacle is found try and avoid it and then return to the track.

Level 3 Exercises:

1. Move the sensor so it will face downwards. Try and create a program which will stop REA from falling from the edge of a desk.

Detecting Colors

How the sensor works:

The color sensor works by shining a white light at an object and then recording the reflected color. It can also record the intensity of the reflection (brightness). Through red, green and blue color filters the photodiode converts the amount of light to current. The converter then converts the current to voltage, which our Lego Hub can read and presents it on our screen.

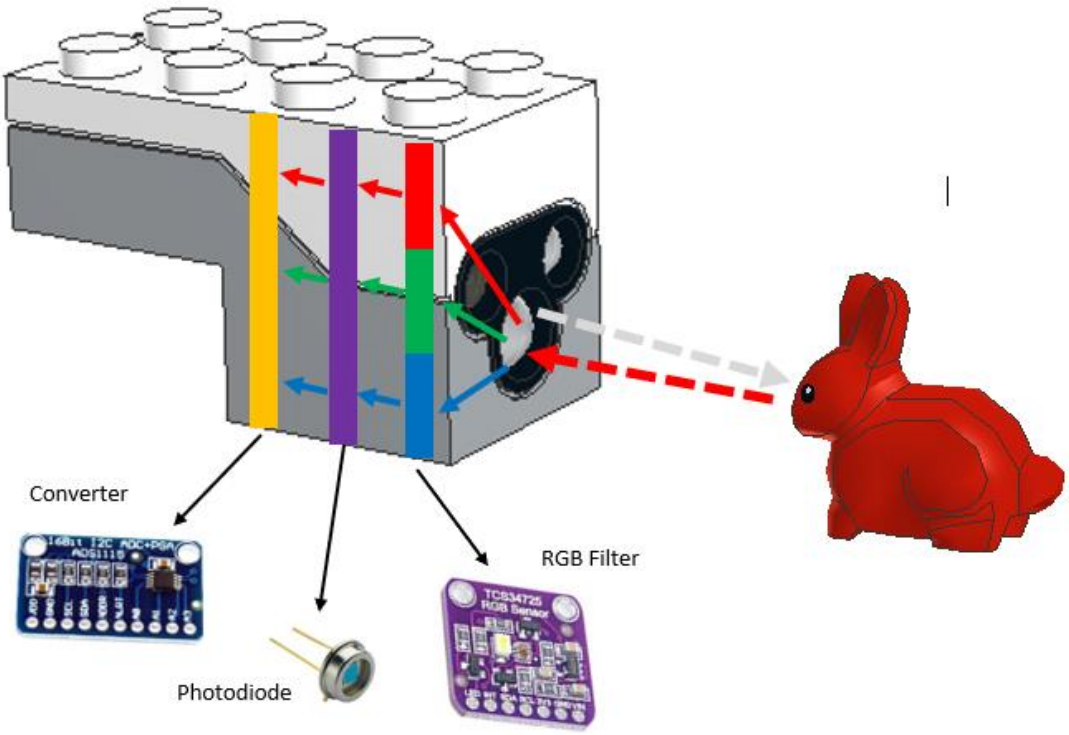


Figure 42: Detecting Colors

The Lego BOOST sensor has to be at a distance of 0.3 cm to 2 cm and at a 90° angle from an object in order to correctly detect the color.

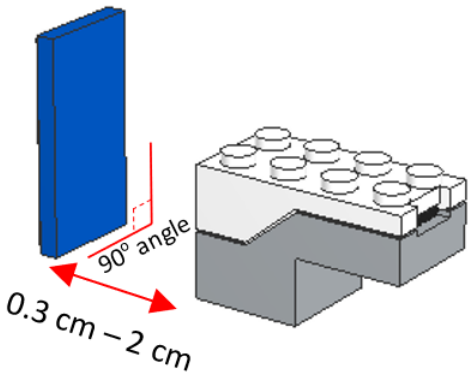


Figure 43: Detecting Colors - Distance

The LEGO BOOST sensor can detect six colors (Black, Blue, Green, Yellow, Red, White, and the absence of color - no color, which means no object is being detected by the sensor).

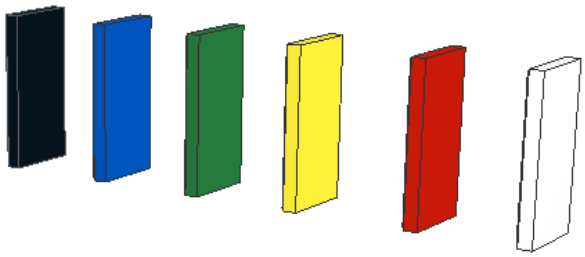


Figure 44: The Colors, which the Sensor can see: Black, Blue, Green, Yellow, Red, White

Notes:

If an object is not in the proposed range of the sensor, the color might not be detected correctly. For example, a green object might be detected as blue.

Also, if the color of the object is not one of the six detectable colors the application will present it as the closest color possible. For example, orange will be presented as red.

The Detecting Colors Blocks:

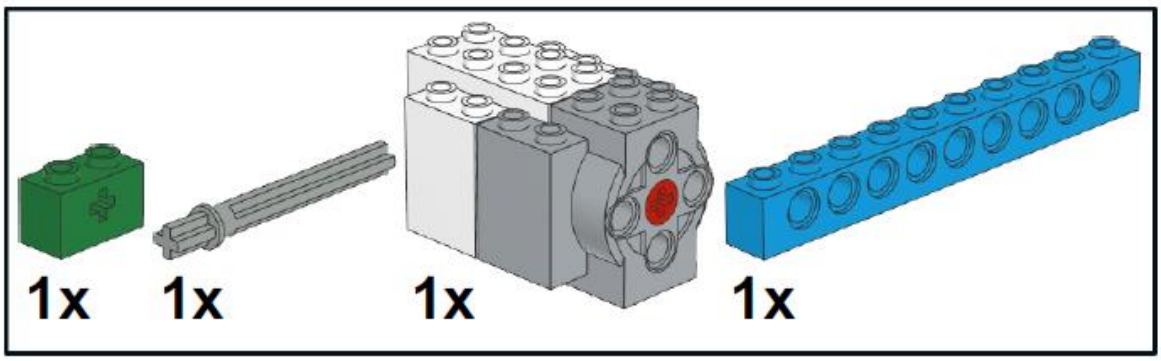
Trigger on Color - Triggers when the color measured by the sensor is equal to the color indicated by the value under it. When triggered, it executes the sequence of code, which follows. It can take seven values: No color, Black, Blue, Green, Yellow, Red and White.

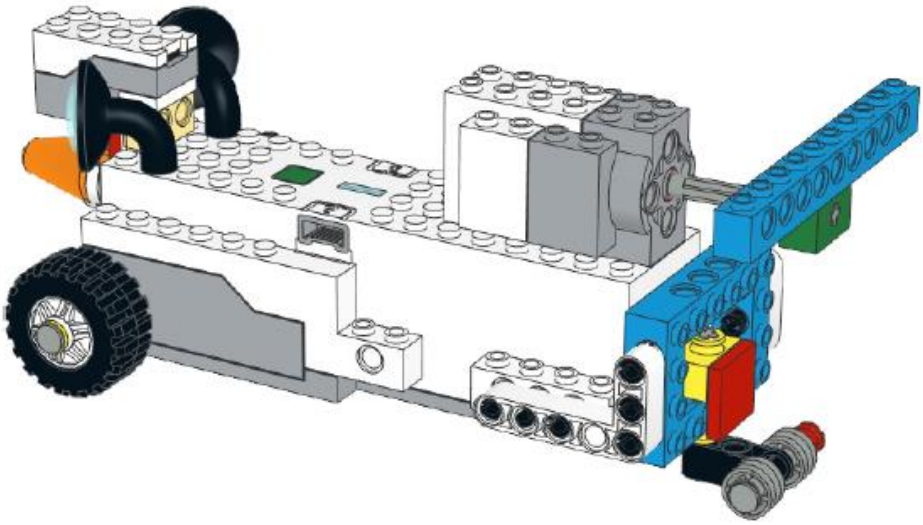
Wait for Color - Waits for the color measured by the sensor to be equal to the color indicated by the value under it. When the color detected is not equal to the value indicated, the program stays paused and when the condition is met the program continues to the following sequence of instructions. It can take seven values: No color, Black, Blue, Green, Yellow, Red and White.

Sensor Color Reporter - Displays in real time the current color measured by the sensor. In order to be used in a program it needs to be attached to the bottom of other blocks. It can show seven values: No color, Black, Blue, Green, Yellow, Red and White.

Sample Programs of Detecting Colors

We will first need to install the interactive motor before we see the detecting colors programs.





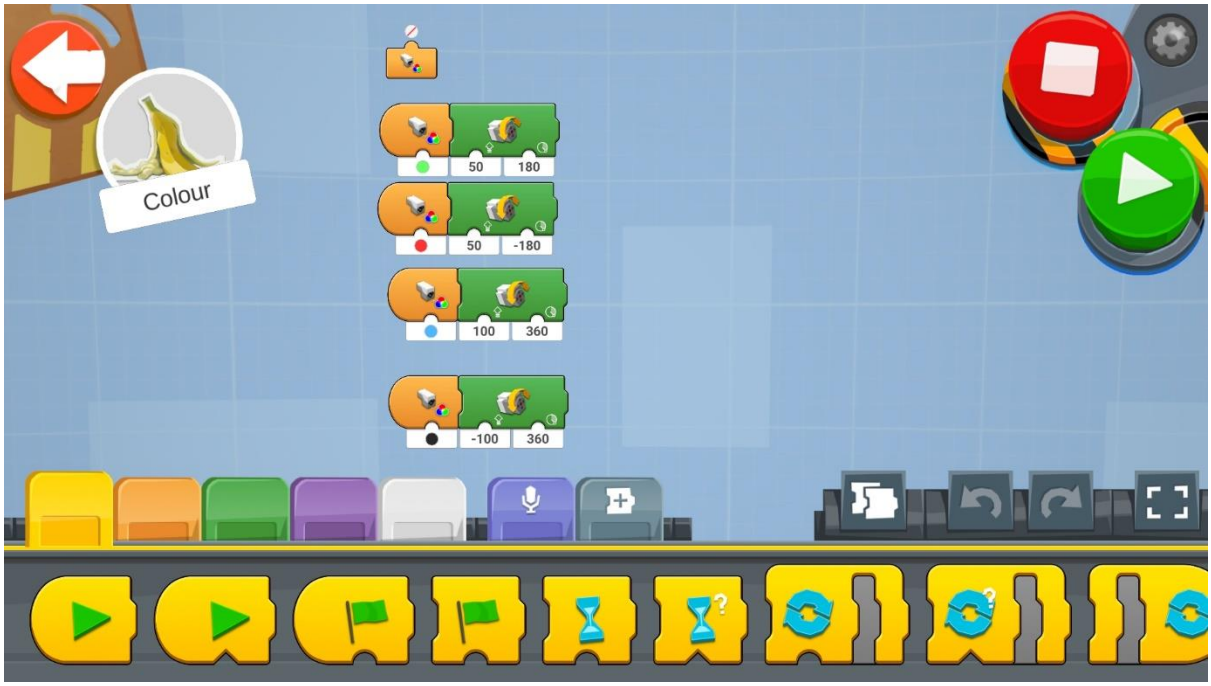


Figure 45: Program - Detecting Colors

Try using the different blocks for Detecting Colors like the example program above.

Create a new project on the Creative Canvas and run each block of code (For Android and iOS app you can find the blocks on the Creative Canvas of Vernie).

1. The first block just indicates in real time the color of the object in front of the sensor.
2. The second program activates when an object of green color is in front of REA's sensor. REA will then move the back propeler 180 degrees to the right at speed 50.
3. The third program activates when an object of red color is in front of REA's sensor. REA will then move the back propeler 180 degrees to the left at speed 50.
4. The fourth program activates when an object of blue color is in front of REA's sensor. REA will then move the back propeler 360 degrees to the right at speed 100.
5. The fifth program activates when an object of green color is in front of REA's sensor. REA will then move the back propeler 360 degrees to the left at speed 100.

The If/Else Blocks:

The **If/Else** statement allows REA to easily make decisions based on the inputs from the sensor. **If** a condition is met (meaning it is **TRUE**), REA will execute a specific block of code. **Else** if the condition is not met (meaning it is **FALSE**), REA will execute another block of code.

If/Else - If a condition is True then execute the top sequence, else if it is False, execute the bottom sequence.

Equal Operator - Returns True when an input from a sensor (color/distance/ambient light) is equal to a value.

Less Than Operator - Returns True when an input from a sensor (color/distance/ambient light) is less than a value.

Greater Than Operator - Returns True when an input from a sensor (color/distance/ambient light) is greater than a value.

Not Equal Operator - Returns True when an input from a sensor (color/distance/ambient light) is not equal to a value.

Example Exercise Sheet 1

Level 1 Exercises:

1. Using the Boost playmat set a square which REA will aim to reach.
   When REA detects:
   a. Green, she should move one square block forward.
   b. Blue, she should make a 90° turn to the right.
   c. Yellow, she should make a 90° turn to the left.

Level 2 Exercises:

1. Try and simulate traffic lights by detecting:
   a. Green - Which will move REA forward
   b. Orange - Which will slow down the speed of REA
   c. Red - Which will stop REA

Level 3 Exercises:

1. Try using the if blocks this time and recreate the first 2 exercises.

   Note to teacher: Nested if will be needed

Measuring the Amount of Light Reflection: The Measuring Light Reflection Blocks:

Sensor Light Level Reporter – Displays in real time the **current ambient light level measured by the sensor**. In order to be used in a program it needs to be attached to the bottom of other blocks. It can indicate values from one (1) to ten (10). One being the darkest and ten being the brightest.

Example Exercise Sheet 2

Level 1 Exercises:

1. Cover REA with a box so no light goes to REA's sensor. When the box is removed REA should move forward for 1 second.

Level 2 Exercises:

1. Take a cardboard, which can fit REA in and allow REA to move around herself 360° on the spot. Punch a hole or create a door from where light can get to the sensor. REA should start rotating and when the sensor senses light REA should stop moving.

Level 3 Exercises:

1. Create a program where when REA detects light intensity less than 5 to move backwards and when it detects light intensity greater or equal to 5 to move forward.

1.5 Following Walls with REA

Prerequisites: A basic understanding of the coding blocks used for recording the reflected light intensity as well as selection - if/else and iteration - loop blocks.

Following Walls

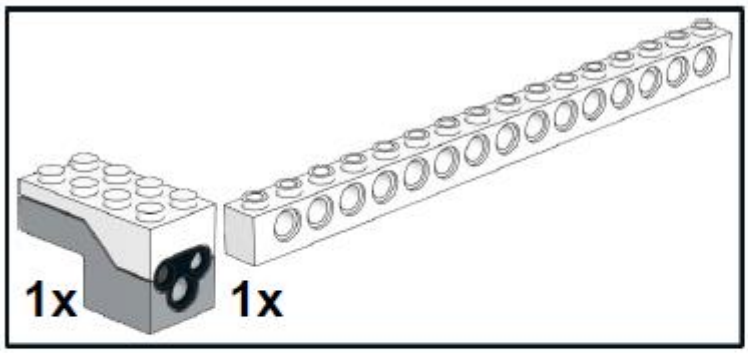
In this section we will see how REA can travel in an area by following one side of a wall or one side of an object.

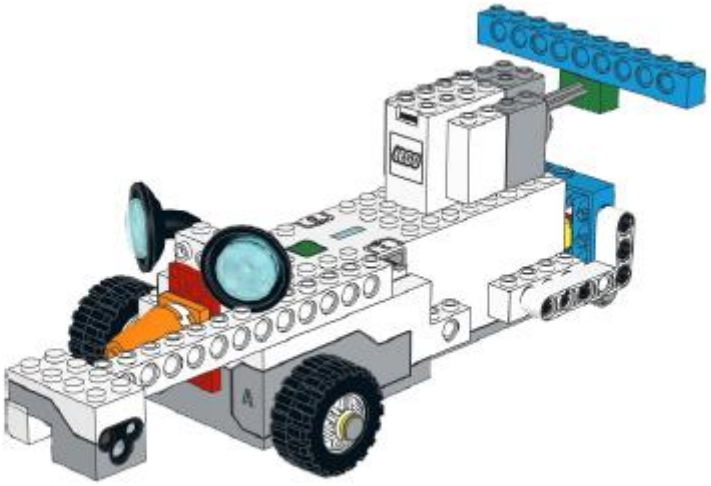
These programs, which react from "inputs" taken from their surrounding environment, are called "Feedback Control" systems. They are called like this because through sensors they monitor changes or situations and respond with changes – "output" in the system, and in our case the system is our robot - REA.

In order to have REA travelling at a constant distance from the wall, the input, which will be used, is the reading from the distance sensor. The output will be the motors of REA, which will adjust the driving direction.

The first thing we will have to do will be to attach an extension in the front part of REA where we will place the color and distance sensor.

Building the Sensor





After we have prepared REA with the distance sensor, we will then need to create a track with walls where REA will be moving in.

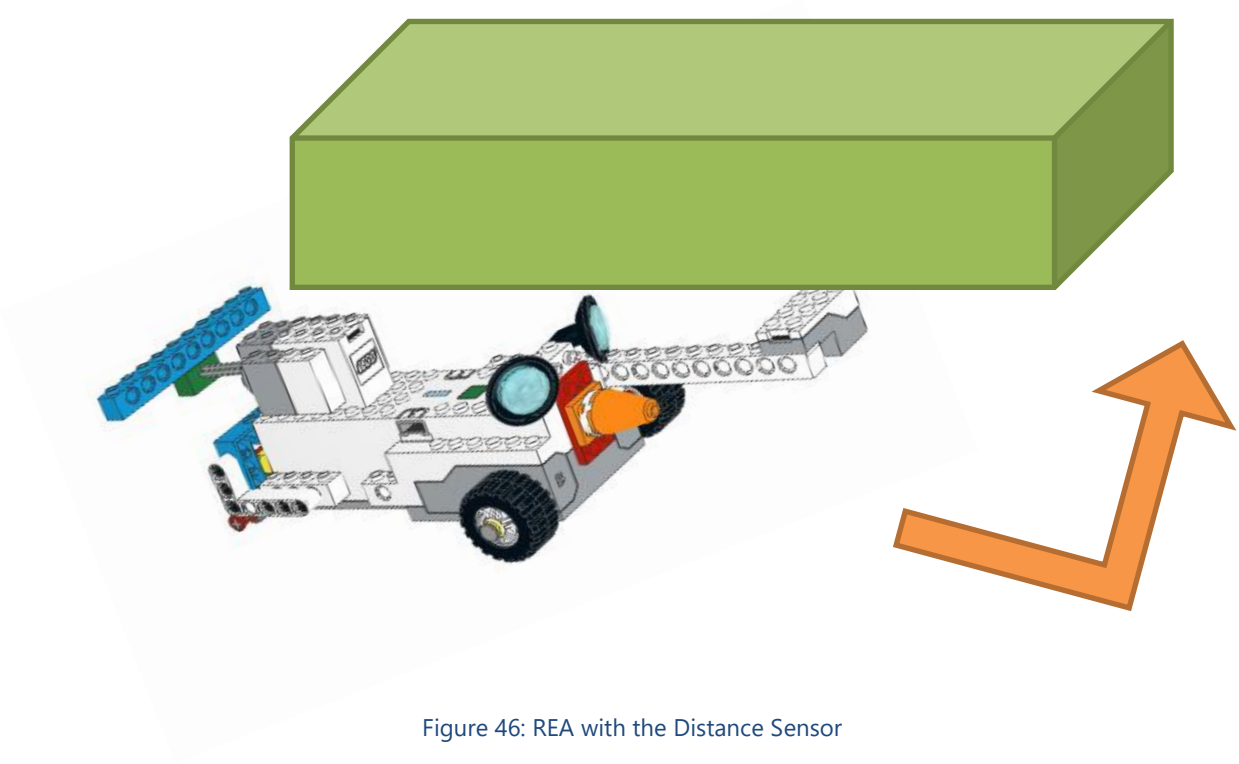


Figure 46: REA with the Distance Sensor

When creating the track, bear in mind that the way the sensor is facing it will be that specific side of the wall, which will be monitored. You can use whatever objects you can find in your classroom to setup the track. It can be boxes or books but just make sure the surface, which the sensor will monitor, will be flat.

An example of a track can be seen here:

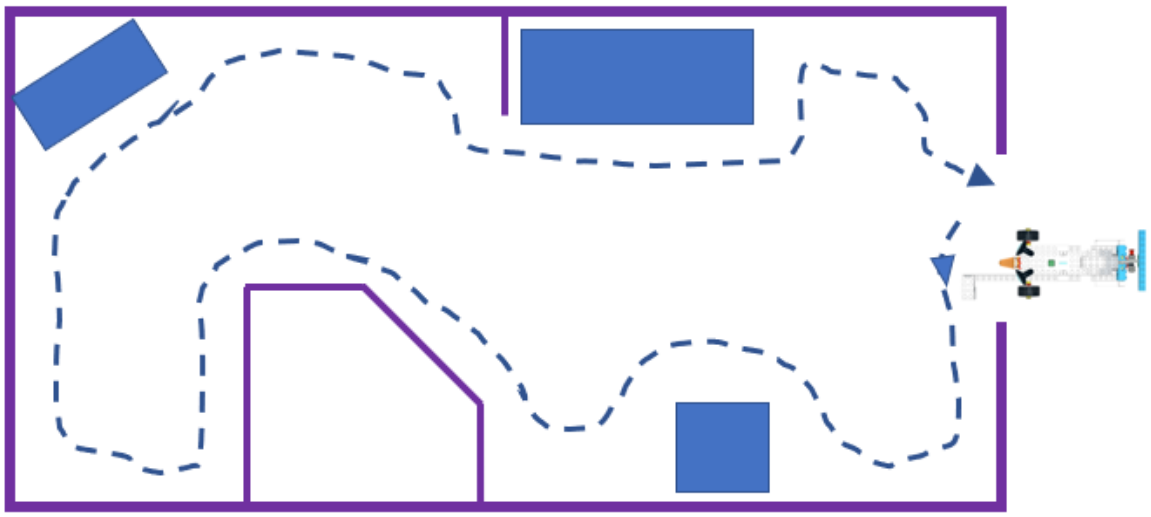


Figure 47: Exercise - Tracking

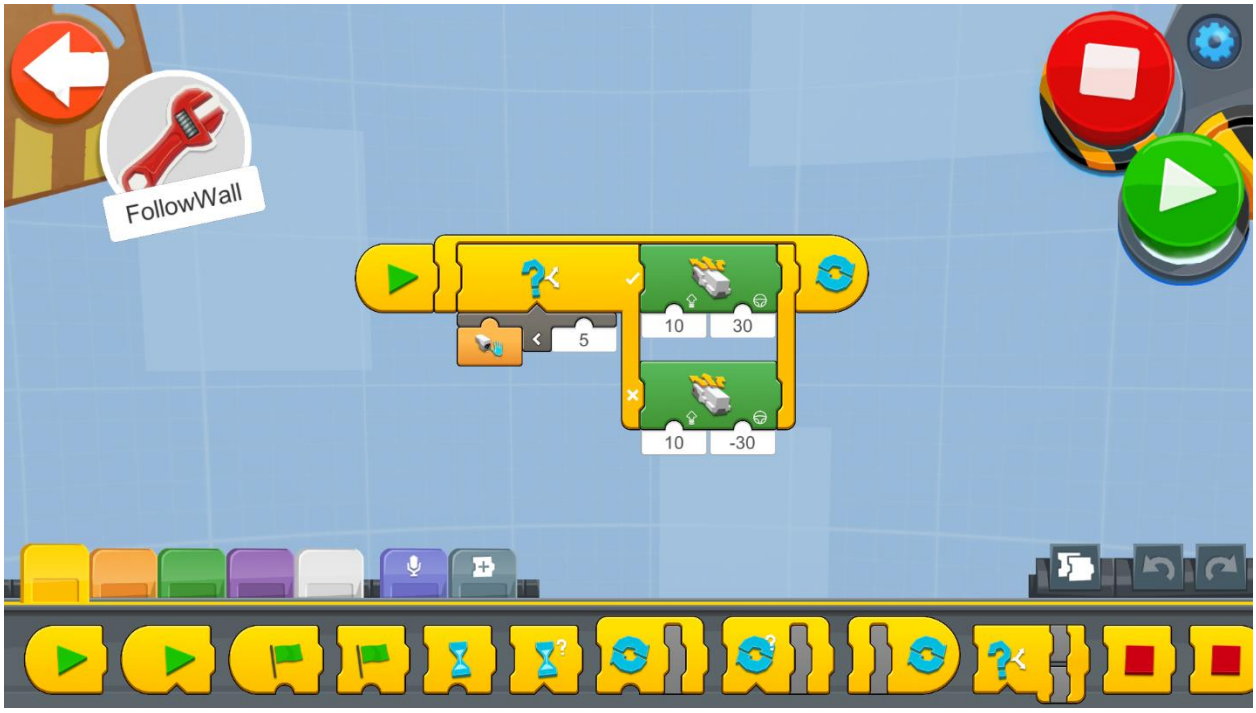


Figure 48: Program - Avoiding Obstacles

Create a new project on the Creative Canvas and run the block of code (for Android and iOS app you can find the blocks on the Creative Canvas of Vernie).

1. An **Infinite Loop** will run the code repeatedly so that REA checks all the time for an object on her left-hand side and act accordingly.
2. Inside the loop there is an **If/Else Block**, which compares the input of the sensor, and if the result of the comparison is True it instructs REA to move accordingly.
3. Under the **If/Else Block** there is a **Less Than Operator** which returns True when an input from a sensor and in this case it's the distance, is less than the value 5 and false if it is equal or greater.
4. If the result of the **If/Else** condition is True, then REA moves 30 degrees towards the right using the **Drivebase Move Steering block** and if the condition is False then REA moves 30 degrees towards the left.

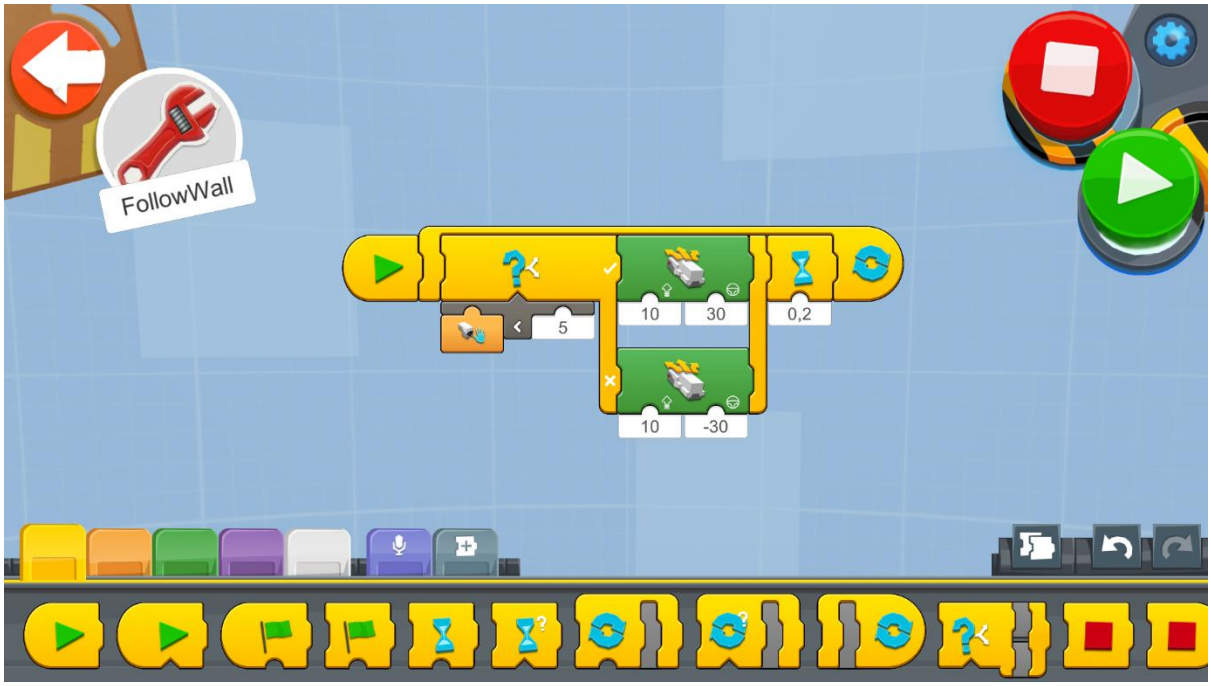


Figure 49: Program 2 - Avoiding Obstacles

Note to teacher: We can add a **Wait for Time block** instruction in order to pause the execution by 0.2 seconds in order to slow down how fast the next reading from the sensor gets processed.

This pause makes the robot drive for 0.2 seconds before checking for the distance from the object. Without the 0.2 seconds pause, the loop would run at full speed, and in some cases, it was observed the Move Hub would receive commands so often that REA would be unable to respond, causing her to move randomly.

Example Exercise Sheet

Level 1 Exercises:

1. Try to modify the above code so that REA can finish the track you created faster!

    a. You can experiment with increasing and decreasing the amount for comparison in the **Less than Operator**.
        i. What happens when 5 is decreased?
        ii. What happens when 5 is increased?

    b. Experiment with increasing and decreasing the **speed** of REA
        i. What happens when the speed is decreased?
        ii. What happens when the speed is increased?

    c. Experiment with increasing and decreasing the **angle** which REA turns
        i. What happens when the angle is decreased?
        ii. What happens when the angle is increased?

    Record the settings you have made changes to. Try and figure out the best settings for your robot to reach the end of the track at the least amount of time.

Level 2 Exercises:

1. Modify REA's Sensor location so it will follow the edge of a table. You will need to make changes on REA's program also.

Level 3 Exercises:

1. Modify REA's Sensor location so it will follow your hand. You will need to make changes on REA's program also.

1.6 Following Lines with REA

Prerequisites: A basic understanding of the coding blocks used for recording the reflected light intensity as well as selection - if/else and iteration - loop blocks.

Follow the Line

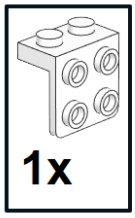
In this section we will see how REA can travel in a predifined path marked with a black line on a white background. This task is called line-following and it is one of the most common tasks in robotics. There are also many competitions which include this task and the tracks can vary in complexity, with lines which have sharp turns and intersections!

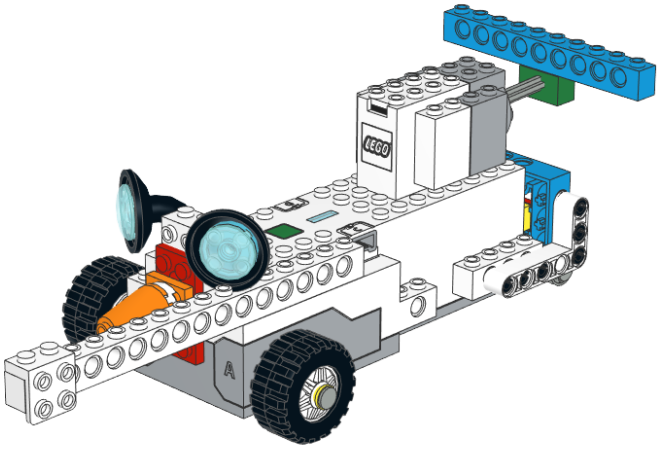
This type of travelling for robots is also used commercially in many cases. One example can be Amazon warehouses where products have to be collected from different locations and brought together in a box so they can be easily packed and shipped. Lego is also using a similar system when producing new parts in its factory. Line-following robots can either follow a clearly marked line on the floor or they can follow metalic wires which they can detect by using magnetic sensors.

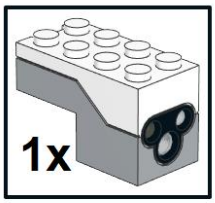
REA will be following a line on the ground of a track. In order for this to happen the LEGO BOOST Color and Distance sensor will need to face downwards on the floor. The line and the background have to be colors of high contrast. We can have for example a black line with a white background or a white line with a black background.

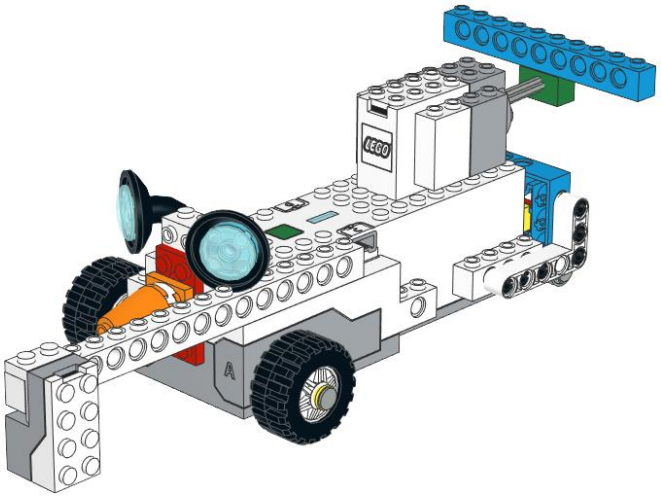
The first thing we will have to do will be to attach in the front part of REA an extension where we will place the color and distance sensor to face downwards.

Building the Sensor






1x



The next thing we will need to do is to create a track where REA will be moving in.

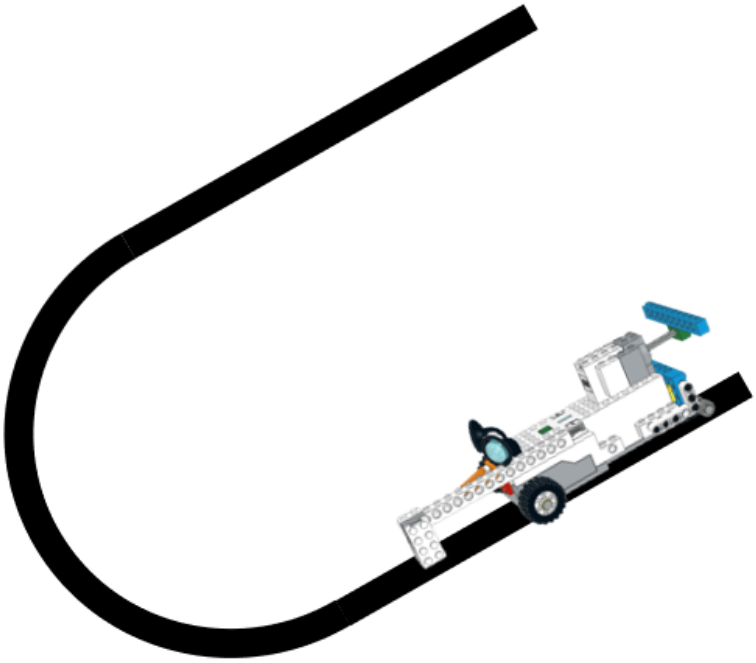


Figure 50: Follow the Line

One simple way to create paths is to attach some black tape to a light floor or to print thick black paths on white paper.

Here is one example of a simple follow the line track. You can use this one in order for REA to practice or create one of your own.

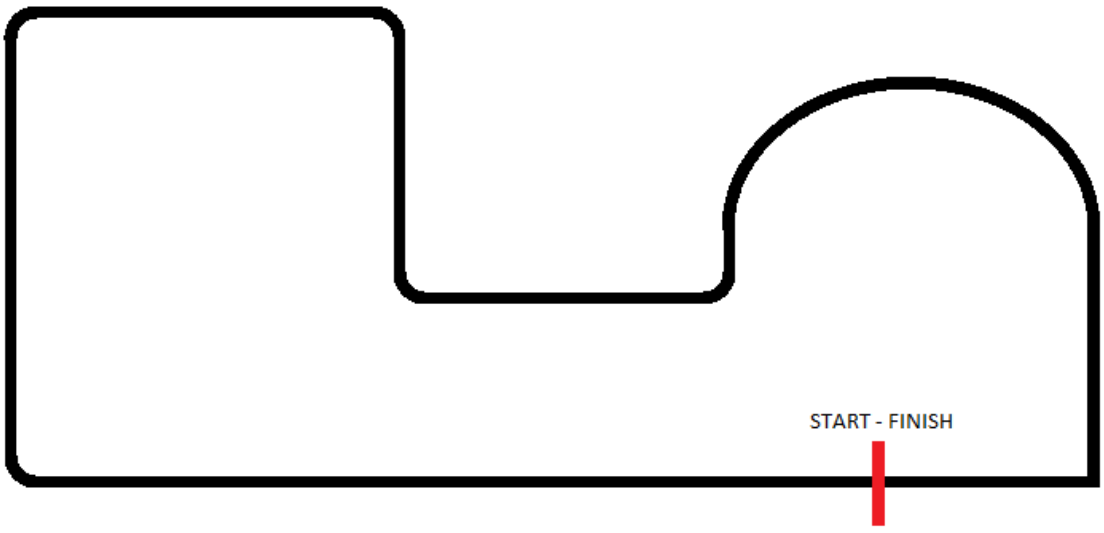


Figure 51: Exercise - Follow the Line

Ideally, REA should aim for the sensor to be between the black line and the white background as shown in Figure 52 below. We do not want the sensor to be only on the black line or only on the white background.

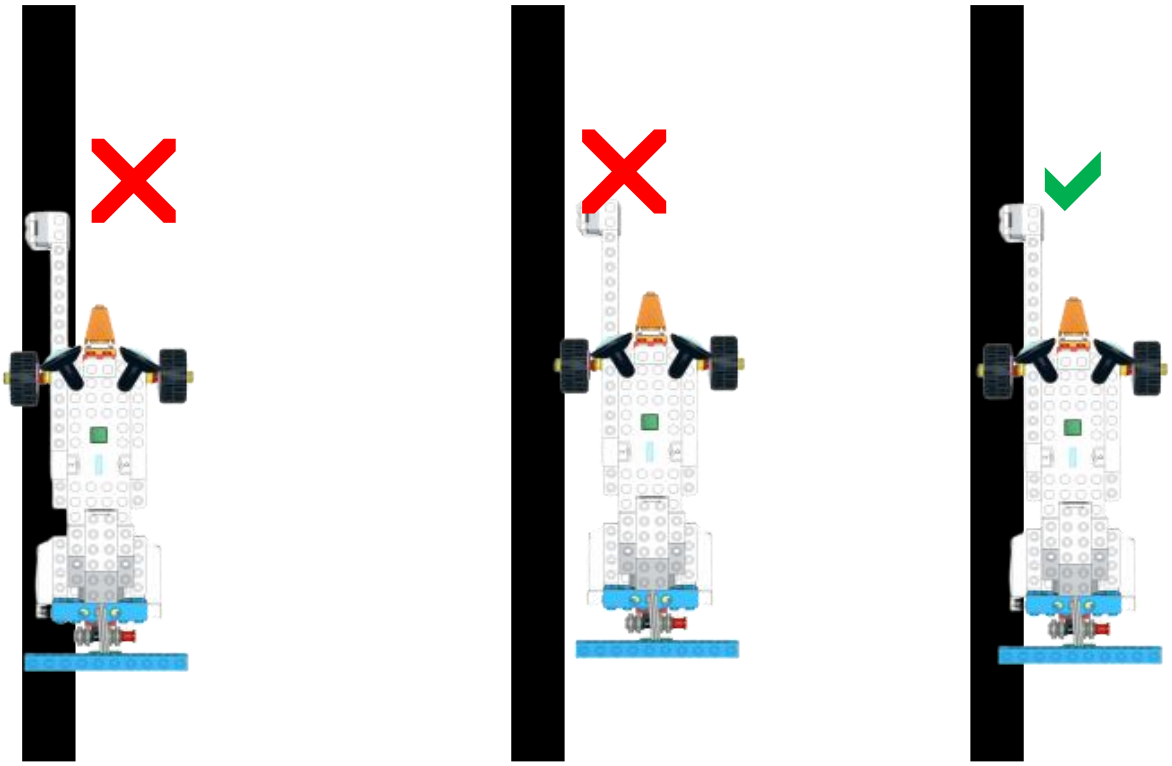


Figure 52: Follow the Line - REAs Position

We will then have to take some measurements from the sensor for our track we have created previously in class.

We will need to use the **Sensor Light Level Reporter** block in order to do so.

There are two ways we can take measurements:

a. By taking one measurement when the sensor is placed in between the black line and the white background as shown in Figure 53.

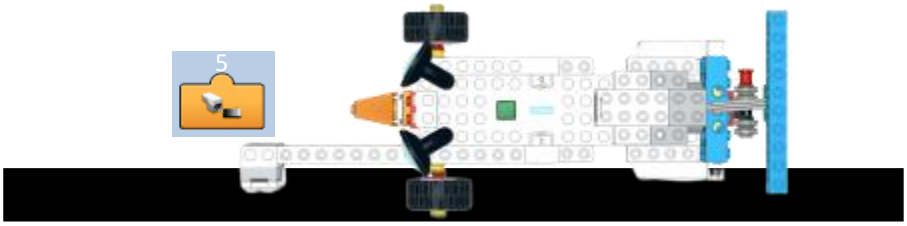


Figure 53: Follow the Line - Take One Measurement

b. By taking two measurements when the sensor is placed:

i) only on the black line

Figure 54: Follow the Line - Take 2 Measurements on the Black Line

ii) only on the white background

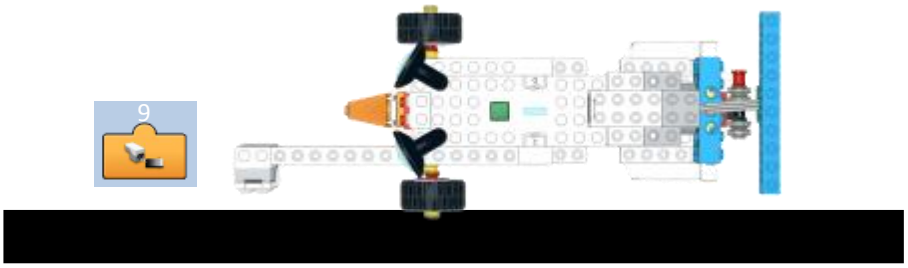


Figure 55: Follow the Line - Take 2 Measurements on the White Background

And then find the average of the two measurements taken:

$$\frac{(\boldsymbol{Black\ Line\ Measurement\ + White\ Background\ Measurement})}{\mathbf{2}} = \boldsymbol{Desired\ Reading\ to\ stay\ on\ correct\ path}$$

Example calculation: $\frac{(1+9)}{2}=5$

Notes:

When taking these readings students should take under consideration some external factors which affect the readings of the sensor. These can be the type of paper used for the track (try and avoid glossy paper), the shadows from various objects such as walls, books, or even the shadows of students reflected on the track. Try and place your track away from the sunlight of your classrooms' window and place it directly under a lamp preferably LED due to the fact that the light is spread more evenly. As a general rule, you should try and be consistent and place REA and the follow the line track at the same exact area as possible.

Another important issue is the color of the line and background. There can be various tones of black and white color so the above measurements are the one's suitable for a specific case. You should take your own measurements which will be the one's suitable for your specific track you have created.

Sample Programs for Following the Line

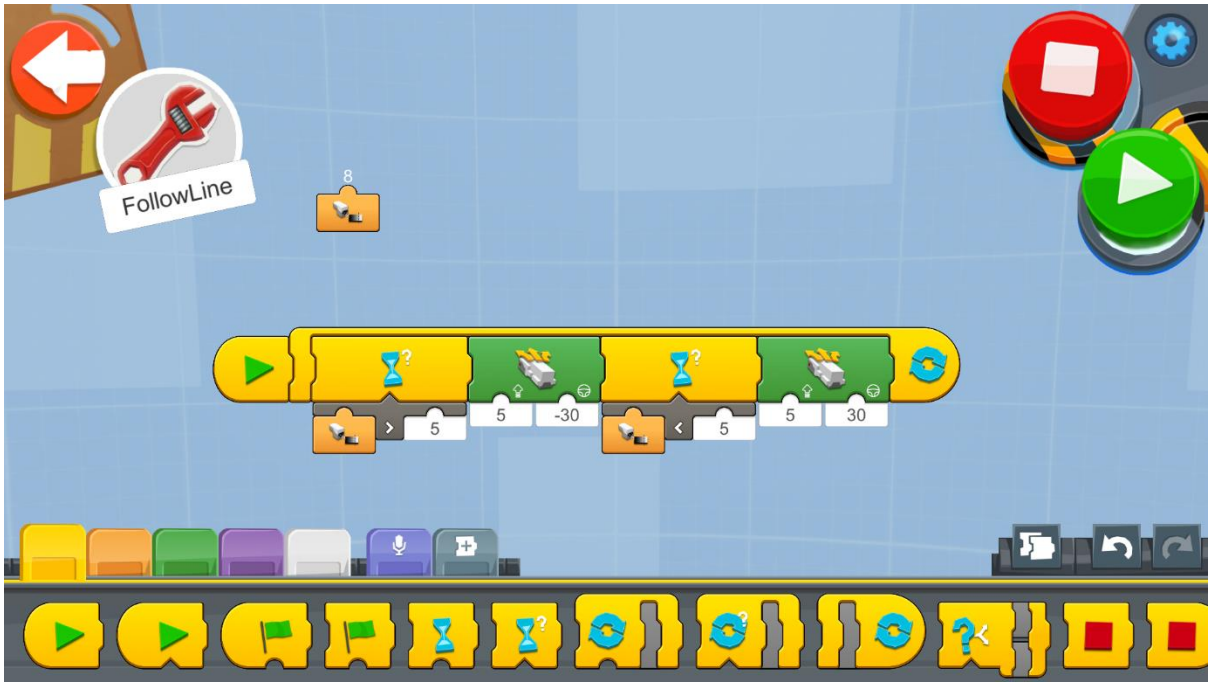


Figure 56: Program - Follow the Line

Create a new project on the Creative Canvas and run the block of code:

1. An **Infinite Loop** will run the code repeatedly so that REA checks all the time for the reading from the sensor and act accordingly.
2. Inside the loop there is a **Wait for True Block,** which waits for the condition to become true in order to proceed to the next instruction block.
3. Under the **Wait for True Block** there is a **Greater Than Operator** which returns True when an input from a sensor and in this case it's the distance and color sensor, is bigger than the value 5 and false if it is equal or less. This will return true if REA is drifting away from the black line and the sensor is facing only the white background.
4. If the result of the **Wait for True Block** condition is True then REA moves 30 degrees towards the left (tries to move closer to the black line) using the **Drivebase Move Steering block at speed 5** (we try and keep a low speed).
5. In the second **Wait for True Block** there is a **Less Than Operator** which returns True when an input from a sensor and in this case it's the distance and color sensor, is less than the value 5 and false if it is equal or bigger. This will return true if REA is getting closer to the black line and the sensor is facing only to the black color of the line.
6. If the result of the second **Wait for True Block** condition is True, then REA moves 30 degrees towards the right (tries to move away from the black line) using the **Drivebase Move Steering block at speed 5**.
7. Finally, it is always good to have an indication in real time of the values the sensor is sending to the App. For this case, the **Sensor Light Level Reporter** block is used.

Note: This program will work well only if the **sensor is placed on the right side** of the line and not on the left!

A second follow the line program:

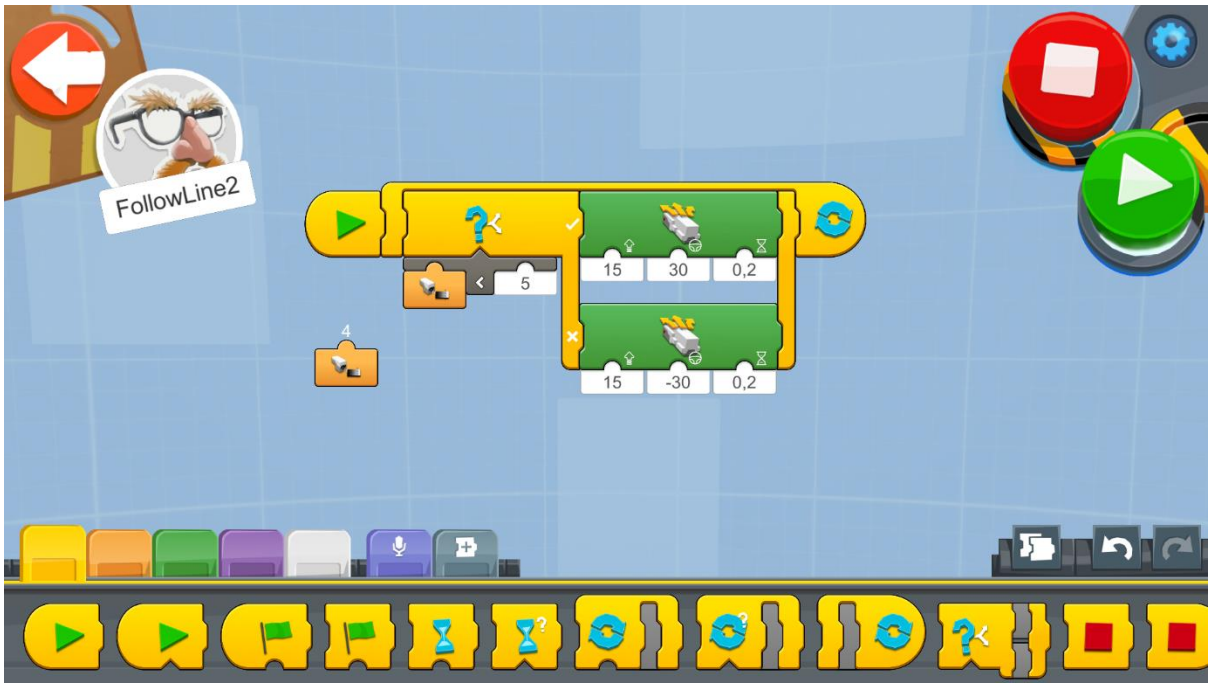


Figure 57: Program 2 - Follow the Line

Create a new project on the Creative Canvas and run the block of code:

1. An **Infinite Loop** will run the code repeatedly so that REA checks all the time for the reading from the sensor and act accordingly.
2. Inside the Loop there is an **If/Else** block which if it is TRUE, REA will move towards the right (tries to move away from the black line) and if it is FALSE, REA will move towards the left (tries to move closer to the black line).
3. The **Compare Less Than** block sets the condition for the Switch block. The **Compare Less Than** block compares the output of the **Sensor Light Level Reporter** block to the value 5. It will return true when the sensor is closer to the black line and false when it moves away.
4. If the output of the **If/Else** block is true, then the **Drivebase Move Steering for Duration block** will move REA at speed 15, at an angle of 30 degrees to the right and will keep on doing this for 0.2 seconds. If it is false, the second **Drivebase Move Steering for Duration block** will move REA at speed 15, at an angle of 30 degrees to the left and will keep on doing this for 0.2 seconds.

Note:
The reason for adding a 0.2 second duration is because of the **delay** from the time between the readings sent to REA until REA decides where to move. Therefore, when REA has moved close to the black line the decision to move away will need to compensate for the time lost making calculations and transmitting the instruction while REA was still moving towards the black line.

Example Exercise Sheet

Level 1 Exercises:

1. Create a follow the line track and try to make REA to complete it. The black line thickness should be at least 2.5 cm.

Level 2 Exercises:

1. Try to make REA to follow the left side of the line instead of the right side, which was indicated in the example programs.

Level 3 Exercises:

1. Based on the "follow the line" program make changes in order for REA to bounce in black line borders.

1.7 Detecting Sound with REA

Prerequisites: A basic understanding of the coding blocks used for selection - if/else and iteration - loop.

React on Sound

A voice-activated robot

In this section we will see how we can control REA with our voice, or by different sounds such as clapping our hands, and by using different sound intensities, with the use of the Sound Sensor blocks.

Sound triggered reactions play a key role in the evolution of the Robotics field. Nowadays there is a continuous research in the field of voice recognition with devices such as Amazon's Alexa, Google's Assistant and Apple's Siri emerging and trying to make our lives easier.

In the case of REA, the sound sensor is actually a built in microphone, which is not located in the Move Hub. Instead, it relies on the microphone, which is located inside of the device, which REA is connected to such as the tablet, laptop or a smartphone.

The Lego Boost App does not actually recognize words but it can recognize how loud a sound is. Hence, we can program it to react differently depending on the sound intensity it receives.

The Sound Sensor Blocks:

Trigger on Sound Level - Triggers when the sound level measured by the sensor is greater than the sound level indicated by the value under it. When triggered, it executes the sequence of code, which follows. It can take eleven values from 0 - 10.

Wait for Sound Level - Waits for the sound level measured by the sensor to be greater than the sound level indicated by the value under it. When the sound level detected is not greater than the value indicated, the program stays paused and when the condition is met the program continues to the following sequence of instructions. It can take eleven values from 0-10.

Sound Level Reporter - Displays in real time the current sound level measured by the sensor. In order to be used in a program it needs to be attached to the bottom of other blocks. It can show values from 0-10 including one decimal value, for example 7,8.

Sample Programs for Detecting Sound

Program 1

Figure 58: Program - Detect Sound

Create a new project on the Creative Canvas and run the block of code:

1. The code will execute when the **Trigger on Sound Level block** triggers. This will happen when **a noise of level greater than 5** is detected.
2. If the sound is detected, REA moves forward using the **Drivebase Move Steering block at speed 50**.
3. The **Wait for Time** block is set for 1 second. The reason for this block is for REA not to be confused with the same sound used for activating the program to move forward and to trigger the rest of the code.
4. The **Wait for Sound Level block** waits for **a noise of level greater than 5** in order to trigger the execution of the rest of the code.
5. Finally, when the **Wait for Sound** Level block is triggered, the **Drivebase Stop** activates and stops REA from moving.

Program 2

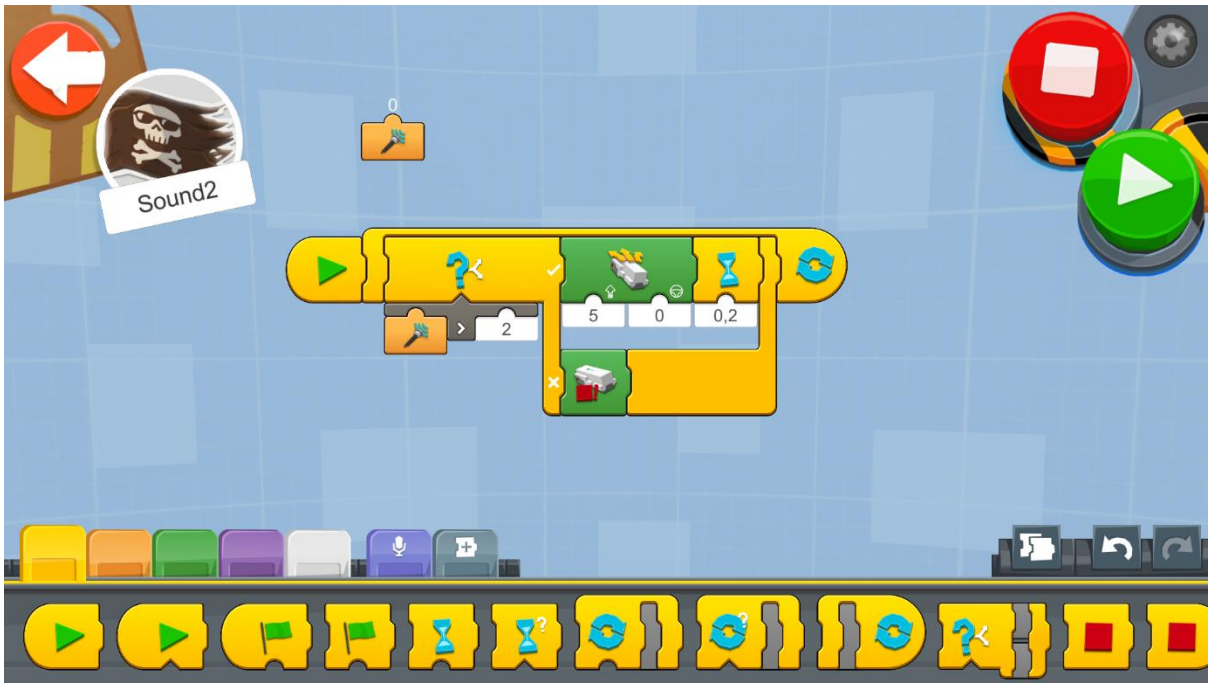


Figure 59: Program 2 - Detect Colors

Create a new project on the Creative Canvas and run the block of code:

1. An **Infinite Loop** will run the code repeatedly so that REA checks the reading from the sensor all the time and act accordingly.
2. Inside the Loop, there is an **If/Else** block which if TRUE, REA will move forward and if FALSE, REA will come to stop.
3. The **Compare Greater Than** block sets the condition for the Switch block. The **Compare Greater Than** block compares the output of the **Sensor Sound Level Reporter** block to the value 2. It will return true, when the sensor detects sound levels greater than 2 and false, when it is equal or less than 2.
4. If the output of the **If/Else** block is true then the **Drivebase Move Steering** block will move REA at speed 55, at an angle of 0 degrees and will keep on doing this for 0.2 seconds. If it is false, the **Drivebase Stop** block will stop REA from moving.

Example Exercise Sheet

Level 1 Exercises:

1. Create a program where, when you whisper REA will move forward and when you shout to move backwards.

Level 2 Exercises:

1. Create a program where each time REA detects a clapping sound to turn to the right and continue straight and when a second clapping sound is detected to turn to the left and continue straight.

1.8 Navigating REA with a Remote Controller

Prerequisites: A basic understanding of the coding blocks used for iteration - loop and basic mathematical calculations (division).

Remote Control

In this section, we will see how REA can move with the use of a remote control.

There are many different ways to communicate with a robot. Robots are most often controlled using either a wired or a wireless controller or they can run autonomously by getting instructions from their control program (which is what we did with REA in the previous chapters).

Remote controlled robots can have various scientific uses including hazardous environments such as extreme heat or cold and radioactive locations, working in the deep ocean to place pipes and cables, as well as exploring the outer space and planets.

The Remote Control Blocks:

Joystick Widget Show - Presents the Joystick Widget in the Lego BOOST App when this block is activated.

Joystick Widget Hide - Hides the Joystick Widget from the Lego BOOST App when this block is activated.

Joystick Widget Speed Reporter - Displays in real time the current Joystick Widget speed (-100..100). In order to be used in a program it needs to be attached to the bottom of other blocks. In most cases it is used as the steering input to a Drivebase Move Steering block. It can show values from -100 to 100 including two decimal values, for example 78,89.

Joystick Widget Steering Reporter - Displays in real time the current Joystick Widget steering (-100..100). In order to be used in a program it needs to be attached to the bottom of other blocks. In most cases it is used as the speed input to a Drivebase Move Steering block. It can show values from -100 to 100 including two decimal values, for example 78,89.

Sample Programs for Remote Control

Program 1

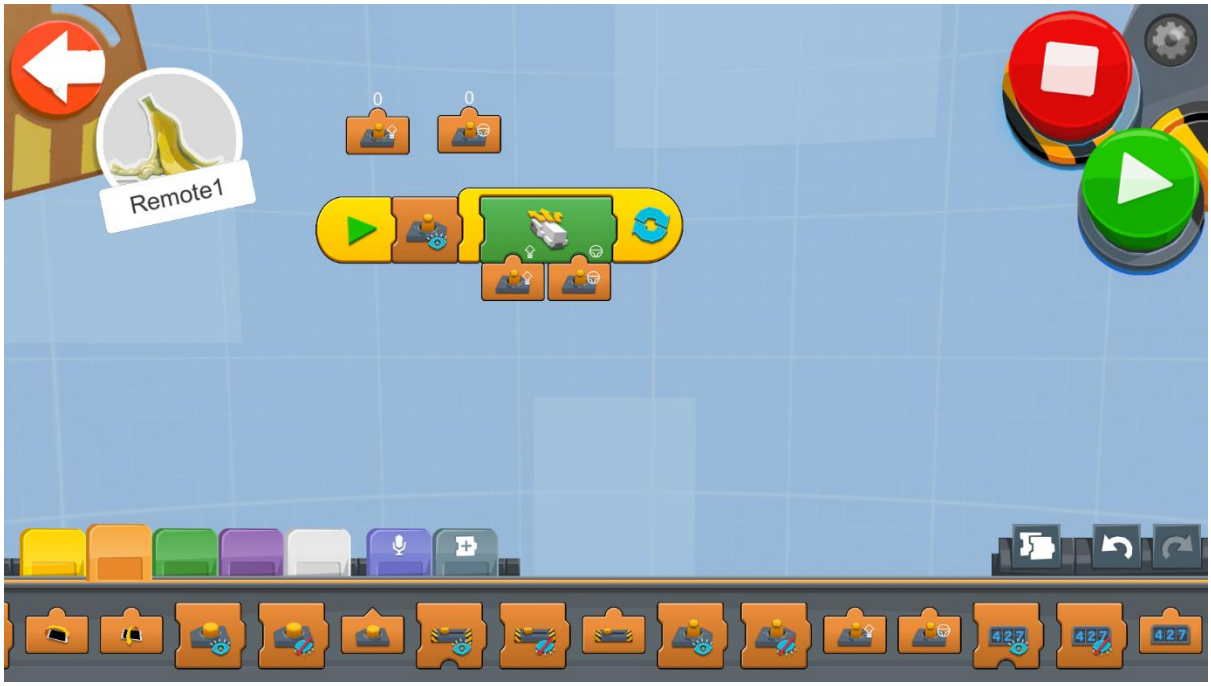


Figure 60: Program - Remote Control

Create a new project on the Creative Canvas and run the block of code:

1. The **Joystick Widget Show** will show the joystick controller on the app (See Figure 61).
2. An **Infinite Loop** will run the code repeatedly so that REA checks all the time for the reading from the joystick and act accordingly.
3. Inside the Loop there is a **Drivebase Move Steering** block which will take as speed input the **Joystick Widget Speed Reporter** block reading and as direction input the **Joystick Widget Steering Reporter** block reading.

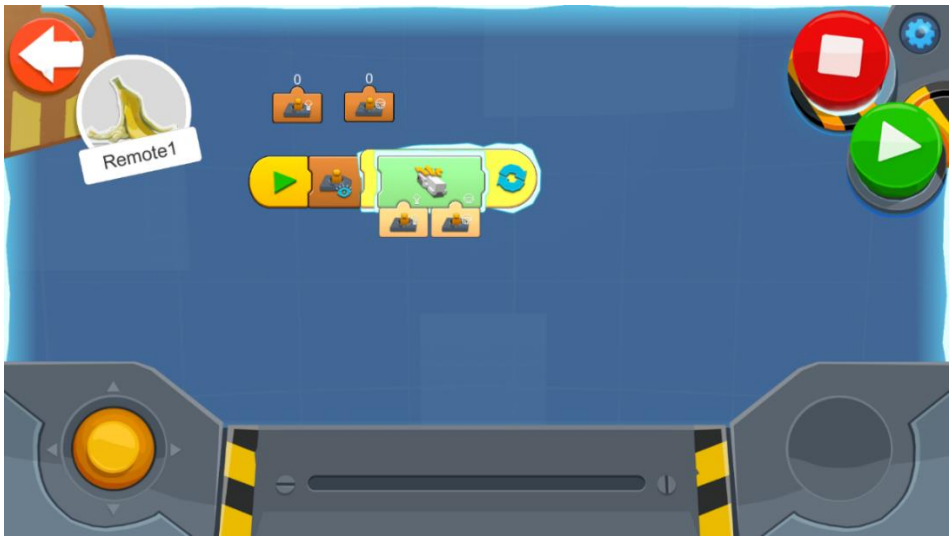


Figure 61: Program - Joystick Controller

Program 2

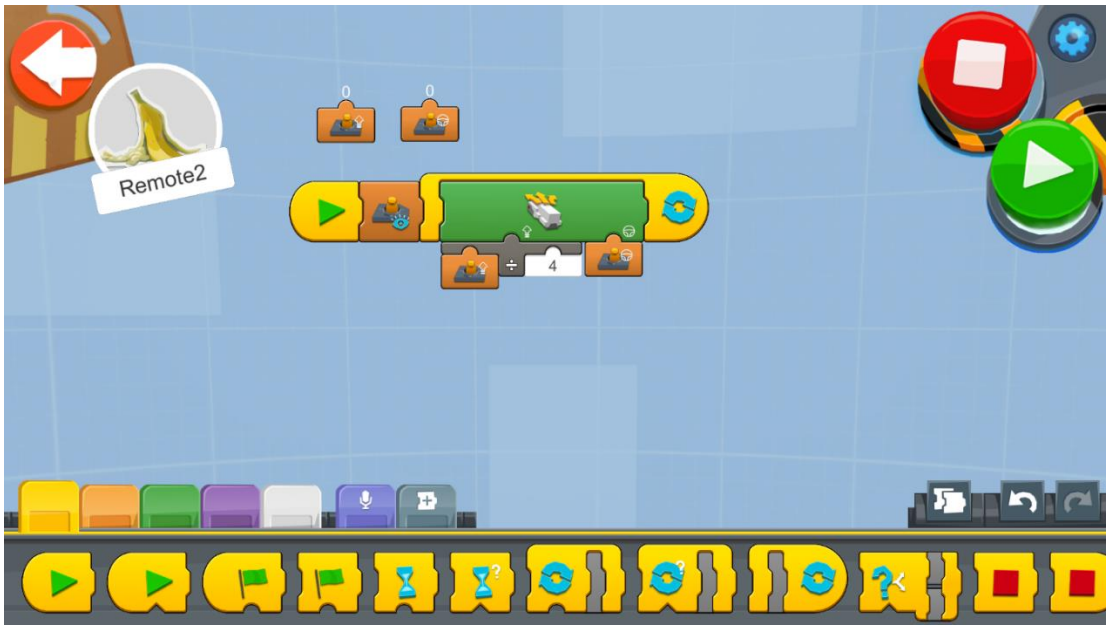


Figure 62: Program 2 - Remote Control

Sometimes we need more accurate control of REA and the input speed is too fast for this. For this we can do a trick in order to reduce the input speed which is sent by the **Joystick Widget Speed Reporter** block. A Division Operator is used for the speed input of the **Drivebase Move** block. The input of the **Joystick Widget Speed Reporter** block is first divided by 4 and then the value is sent as the speed value. For example, if the value reading from the joystick is 100 the speed will be:

$$100 \div 4 = 25$$

So, the real speed input would be 25.

Example Exercise Sheet:

Level 1 Exercises:

1. Race on a track! Prepare for a race with your classmates. Create a race track in your classroom with different objects available to you. Take control of REA and navigate through the curves and straight lines. Here is an example of a track but you can always use your imagination and create a different one!

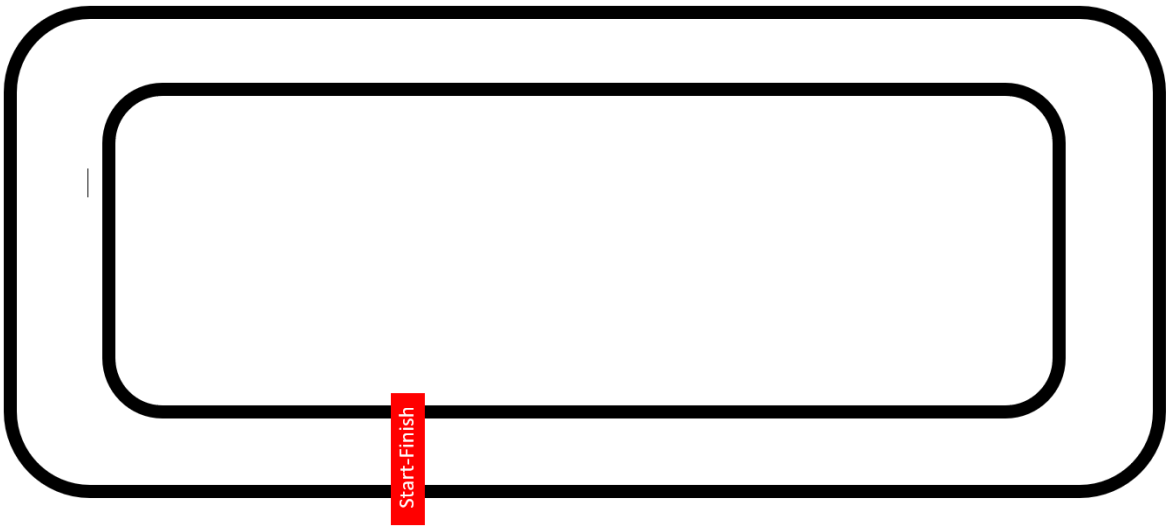


Figure 63: Exercise - Race Track

Level 2 Exercises:

1. Have a Sumo competition! Create a Sumo track and try to throw outside the track the opposing robot. You can modify REA to be heavier or lighter depending on your strategy. The Sumo track should have a diameter of 77cm similar to the figure below.

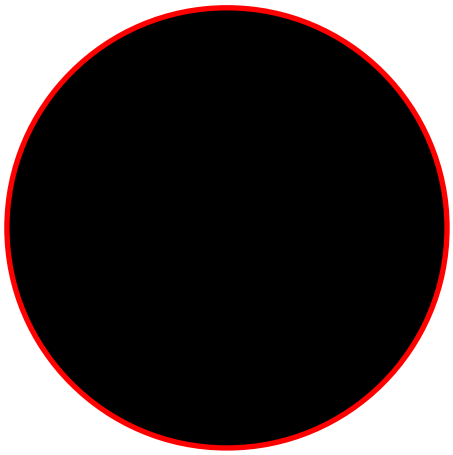
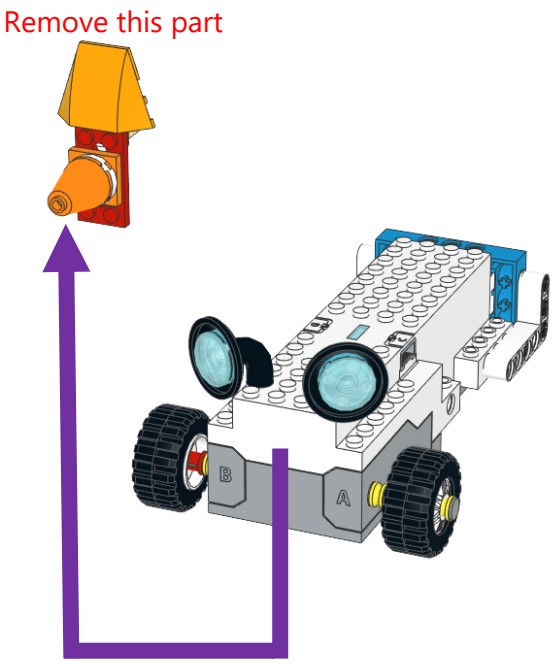


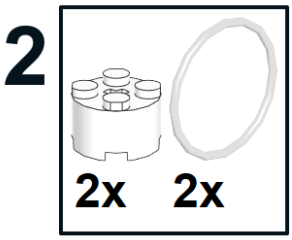
Figure 64: Exercise - Sumo

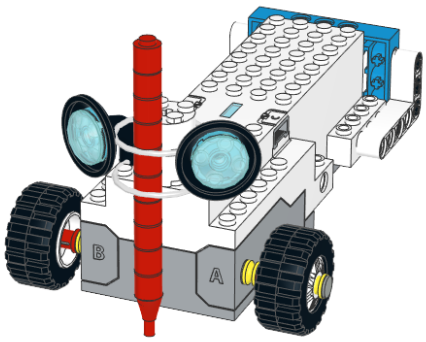
Level 3 Exercises:

1. Let us modify REA and add a pen/marker to her. Then, with the help of the joystick of the Boost App try to write your name initials on a piece of paper. You can follow the instructions provided or you can also think of other ways to add the marker.

1







Note: You can make the pen hold more firmly by adding a sellotape and sticking together the pen and the move hub.

Section C - Advanced Robotics

In this section, students will learn about specific and specialized aspects of robots and programming such as the use of gears and the concept of variables.

1.9 Using Gears with REA

Prerequisites: A basic understanding of the coding blocks used for movement and basic mathematical calculations.

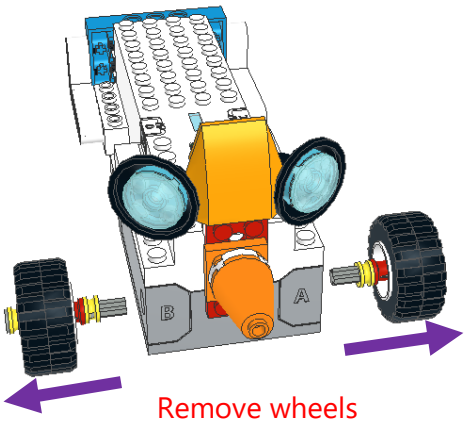
Gears

Geared Up REA

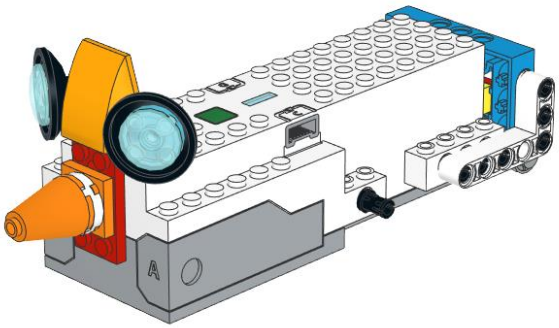
In this section, we will see how gears can be used in order to change the speed and torque of REA. Gears are wheels with "teeth" which connect with other gears in order to create rotation. The simplest example could be for a motor to transmit rotation to a wheel.

The first thing we have to do in order to be able to investigate how gears work, will be to make some changes in REA's wheels as shown in the instructions below.

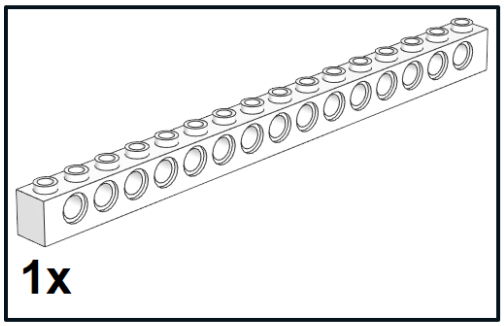
1

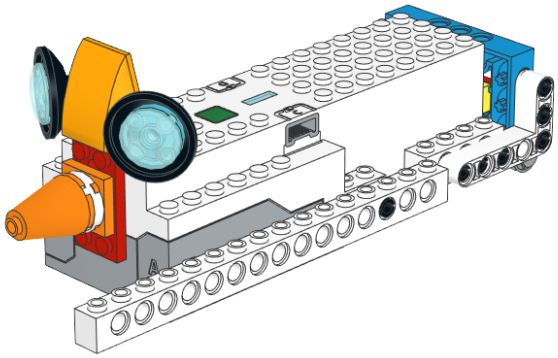


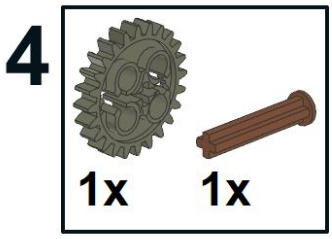
2
1x

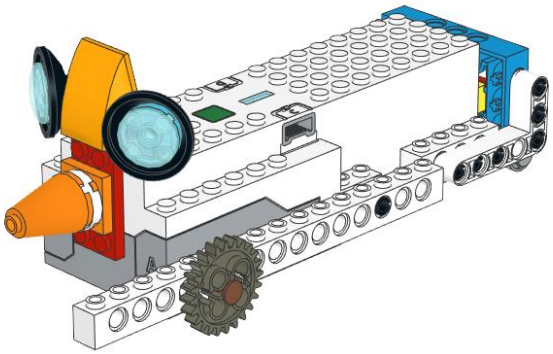

A

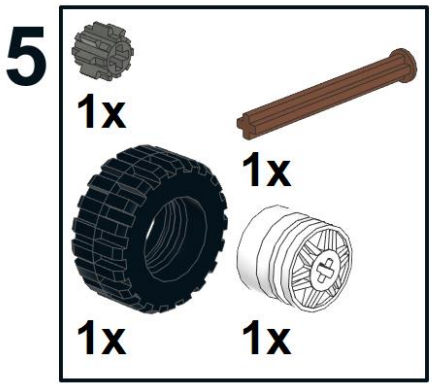
3

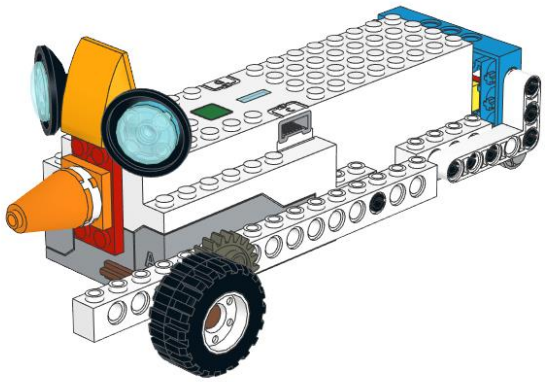




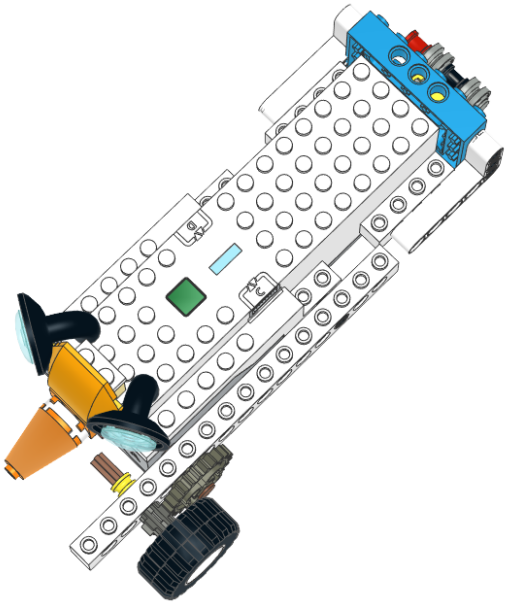

4
1x
1x



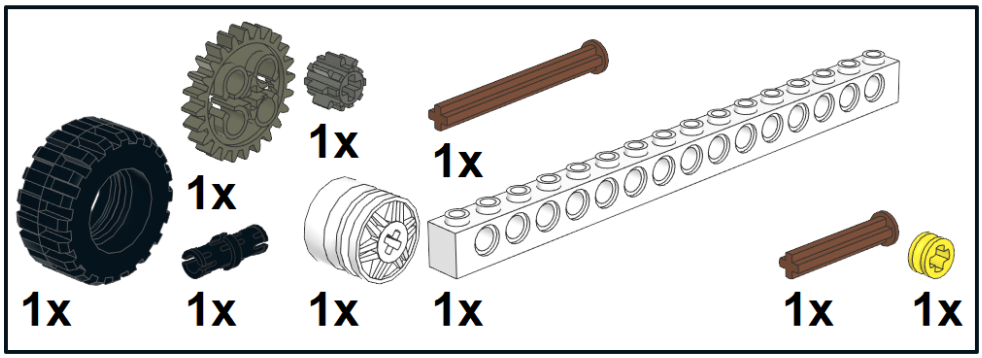

5
1x
1x
1x
1x

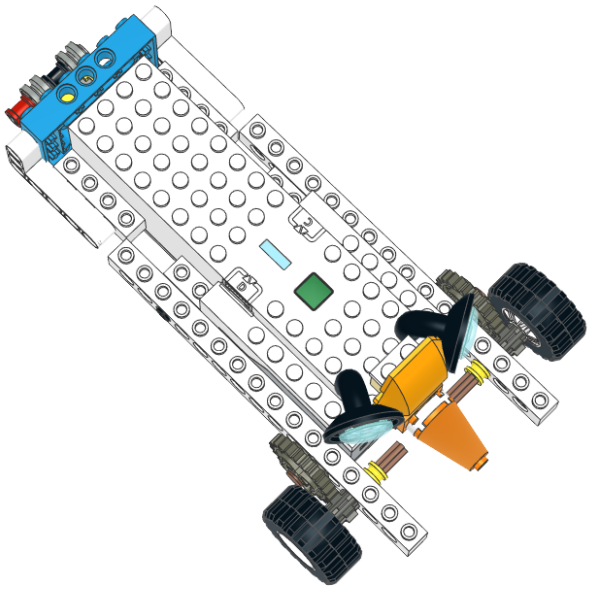


6
1x



7





Now that we have constructed "Geared Up REA", let us see in more detail the gears used.

The gear which is connected to the motor of the Move Hub is a 12-tooth (12z) double-bevel gear and the gear which is connected to the wheel is a 36tooth (36z) double-bevel gear.

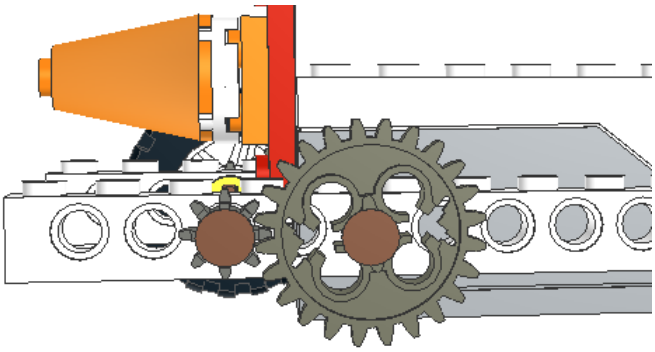


Figure 65: Geared Up REA

Create and run a simple program for REA to drive forward for 1 second at full speed. What do you observe?

1. REA travels in the opposite direction.

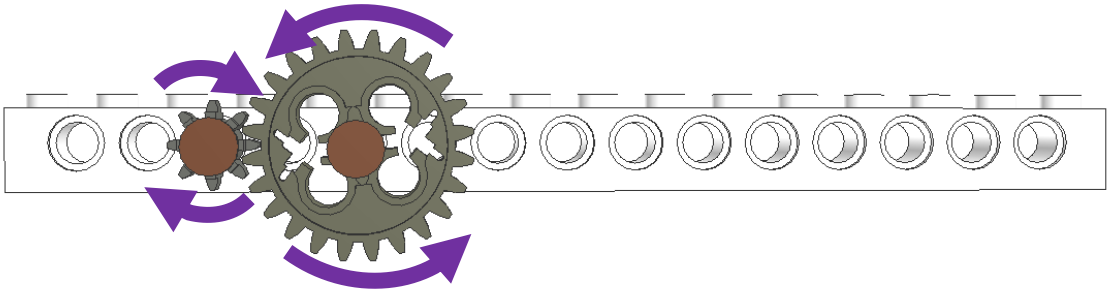


Figure 66: Gears Position

Let us have a closer look on the gears of the left side of REA. The reason for travelling backwards instead of forwards is that the gear, which is connected in the Move Hub motor when asked to move forward, it will move the big gear anti-clockwise. The small gear will then move in a clockwise direction thus the wheel will move backwards.

The opposite happens on the right side of REA, where the big gear moves clockwise and the connected small gear moves in an anti-clockwise direction.

2. REA travels in greater speed.

The reason for this is due to the fact that when the big gear is rotated once, the small gear is rotated three times. Therefore, when the drive motor is rotated once the actual wheel which is connected to the small gear turns three times.

This relationship between input gear, which is connected to the motor, and output gear, which is connected to the wheel, is expressed in terms of the ratio between the numbers of their teeth, which is commonly known as the gear ratio.

The following equation shows how we can find the gear ratio:

$$\textit{Gear Ratio} = \frac{\boldsymbol{Number\ of\ teeth\ of\ output\ gear}}{\boldsymbol{Number\ of\ teeth\ of\ input\ gear}}$$

In our case the number of teeth for the output gear (the one connected to the wheel) is 12 and the number of teeth for the input gear (the one connected to the motor) is 36.

Therefore, the Gear Ratio equation will be as follows:

$$\textit{Gear Ratio} = \frac{\mathbf{12}}{\mathbf{36}}$$

Exercise:

With a timer and a ruler measure the time taken for REA and Geared Up REA to complete the distances indicated and fill in the table

	Distance	**REA time**	**Geared Up REA Time**
1.	30 cm		
2.	40 cm		
3.	50 cm		
4.	60 cm		
5.	70 cm		

What do you observe?

How much faster is Geared Up REA in comparison with normal REA?

Geared Down REA

In this section, we will see how gears can be used in order to increase the torque of REA. Torque is a twisting force, which allows an object to lift more weight.

The first thing we have to do in order to investigate how torque works, will be to make some changes in Geared Up REA and transform her into Geared Down REA as shown in the instructions below.

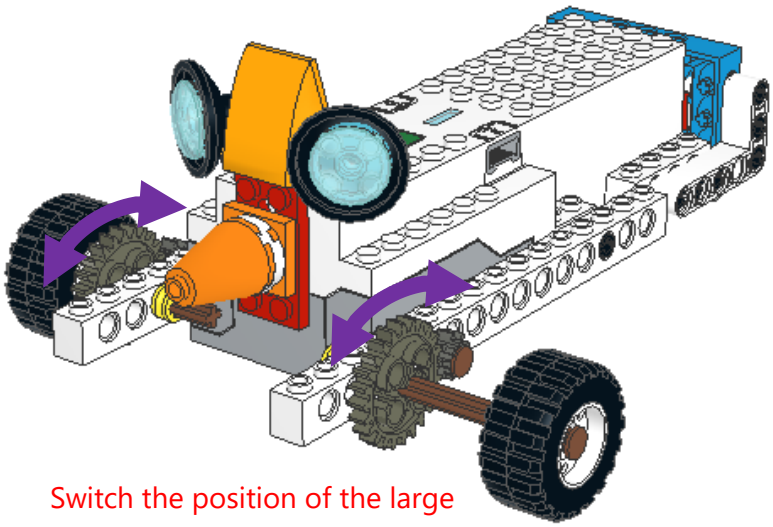


Figure 67: Geared Down REA

When the small (12z) input gear, which is connected directly to the motor moves, the adjacent large (36z) output gear will have three times more torque than the input gear.

Torque is the twisting force which, when applied to the axle which a wheel is connected to, makes it rotate.

This increased level of torque lets us lift a load with three times less energy than you would have to use if you were lifting the load by turning the axle directly.

Torque helps cranes lift objects, which without gears would be impossible. Also, cars use gears in order to move and to use roads which have steep angles.

Exercise:

Take a piece of flat wood big enough for REA to move on and add some books in order to create an angled incline.

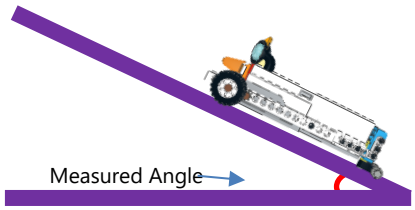


Figure 68: Exercise - Measure the Angle

Use a protractor to measure the angle and compare the angles, which REA and Geared Down REA can climb. Complete the following table.

	Angle in Degrees	REA time	Geared Down REA Time
1.	10		
2.	15		
3.	20		
4.	25		
5.	30		
6.	35		
7.	40		
8.	45		
9.	50		
10.	55		
11.	60		
12.	65		

What do you observe?

What is the maximum angle Geared Down REA can climb and what is the maximum for normal REA?

1.10 Using Variables with REA

Prerequisites: A basic understanding of the coding blocks used for selection - if/else and iteration - loop blocks as well as basic mathematical calculations.

Mathematics and Calculations

In this section, we will see how the different mathematical operations can be used in conjunction with variables.

But what are variables? The code we have created until now did not need to remember any values because the numbers we used such as the reflected light and the level of sound came from the sensor.

There are cases though where we will want REA to store and remember a value such as a number in order to use it later in her program.

In order to store numbers in REA's memory, you'll need to first assign a name to each memory location to avoid mix-ups.

In the example below, we have two memory locations - containers.

The first container has four Red Lego blocks so we can say that memory location "RED" has the number 4 assigned to it. RED=4.

The second container has three White Lego blocks so we can say that memory location "White" has the number 3 assigned to it. WHITE=3.

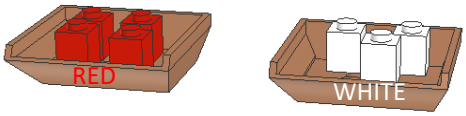


Figure 69: Red and White Lego Boost

The Lego Boost App gives you the option to name the container-memory with the use of only one letter of the English alphabet (Figure 70) or a symbol of a different symbolic alphabet (Figure 71).

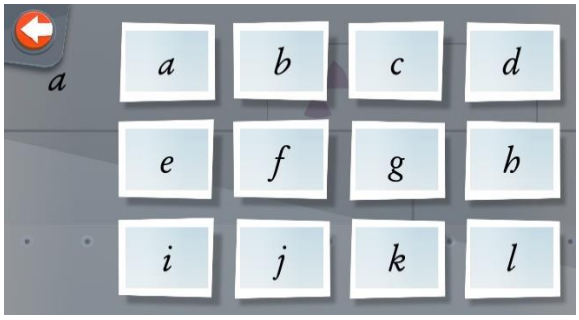


Figure 70: English Alphabet

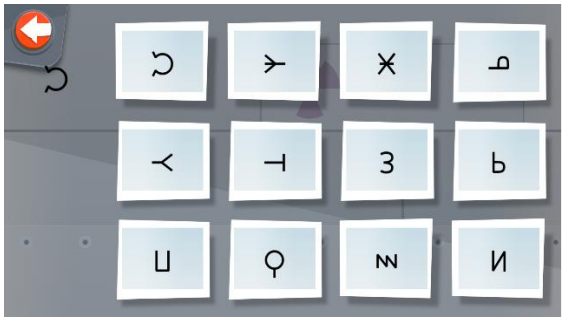


Figure 71: Symbolic Alphabet

The Operator Blocks:

Addition Operator - Returns the result of adding a number to another number.

Subtraction Operator - Returns the result of a number subtracted with another number.

Multiply Operator - Returns the result of a number multiplied with another number.

Division Operator - Returns the result of a number divided by another number.

Equal Operator - Returns True when a number is equal to a value.

The Variable Blocks:

Variable Read Local - Displays in real time the number stored in the local variable.

Variable Write Local - Updates the local variable to store the number indicated.

Sample Programs for Mathematics and Calculations

Program 1

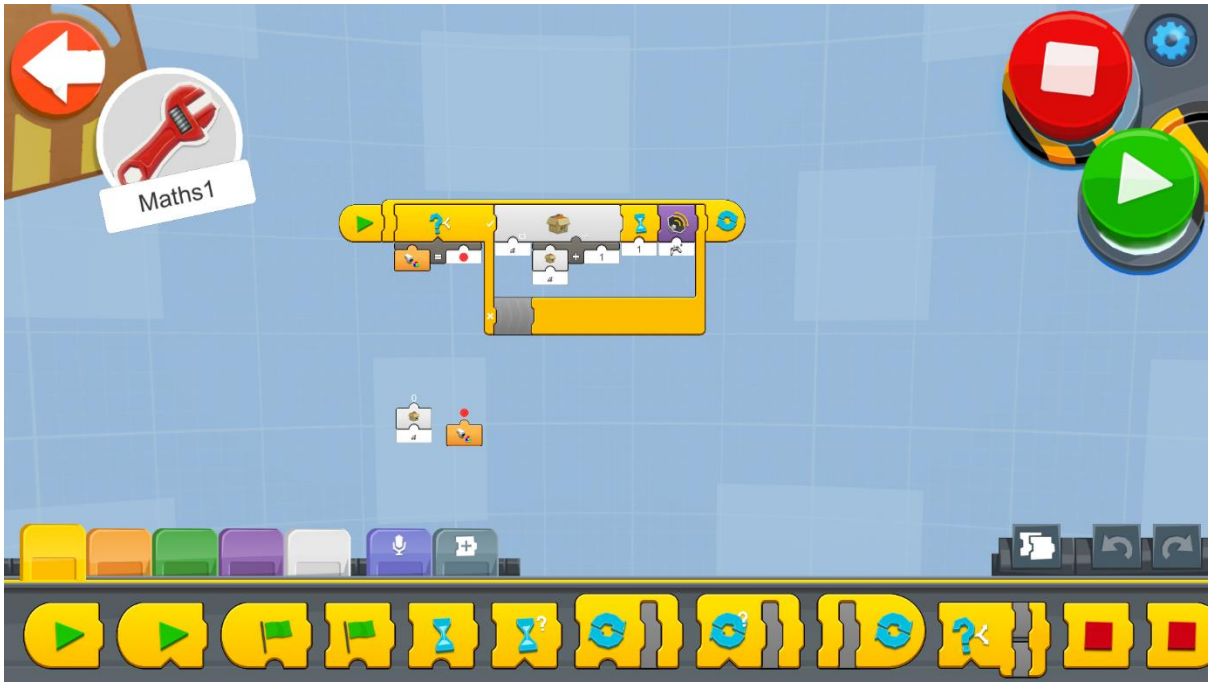


Figure 72: Program - Mathematics and Calculations

Create a new project on the Creative Canvas and run the block of code:

1. An Infinite Loop will run the code repeatedly so that REA checks all the time for the reading from the sensor and act accordingly.
2. Inside the Loop there is an If/Else block, which if the sensor detects a RED color, the contents of variable “a”, will increase by 1.
3. After each increase of the variable by 1 the wait block is activated for 1 second. The reason for this is for the variable not to increase more than once when the red color is detected.
4. Finally, the sound block triggers right after each increase for the user to know that the variable was increased.

Program 2

Figure 73: Program 2 - Mathematics and Calculations

From the previous created program, complete the additional code as shown above.

1. The second If/Else block returns TRUE if the sensor detects a GREEN color.

2. If the result of the If/Else block is TRUE, the medium motor will turn at speed 50 for time (in seconds) equal to the content of variable “a”.

Example Exercises

Level 1 Exercises:

1. Having in mind the previous example programs create a program which when:

    a) RED color is detected to add 1 to variable “a”
    b) GREEN color is detected to subtract 1 from variable “a”
    c) WHITE color is detected to multiply by 2 the contents of variable “a”
    d) BLUE color is detected to divide by 3 the contents of variable “a”

Level 2 Exercises:

1. Create a program, which stores how many squares in the playmat REA can move forward. For example, if the content of the variable used is 3, REA has to move 3 squares forward.

Level 3 Exercises:

1. Create a program, which will slowly increase the speed of REA.

Tip: You can use a loop, add 1 to a variable and use the variable as the speed indicator.

1.11 Troubleshooting

Update firmware:

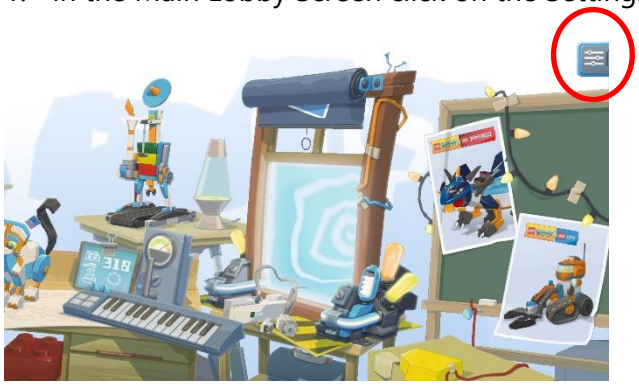
In order to receive updates for the Lego Boost platform, the connected Application must be updated first from your mobile device or computer. When there is a firmware update, the App will first ask you to update as soon as you connect the Lego Boost MoveHub.

There are cases though that a hard reset of the firmware might be necessary. Situations where the MoveHub does not move accordingly to the program. For instance, from our experience the external motor was confused with the right or left motor of the MoveHub. In order to trigger a reset of the firmware you just need to connect the MoveHub to the app and then hold down the Green button for 10 seconds and the firmware reset will start.

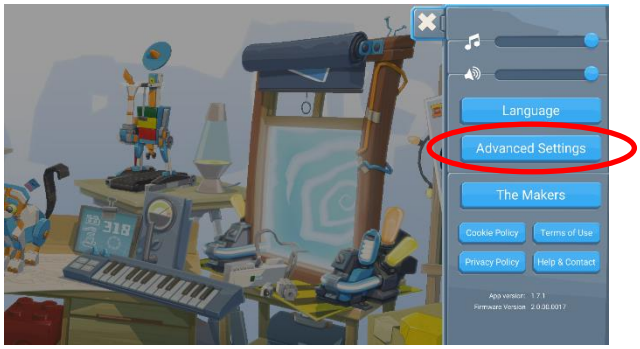
Reset App:

In order to reset the progress for all the builds, you will need to follow these steps:

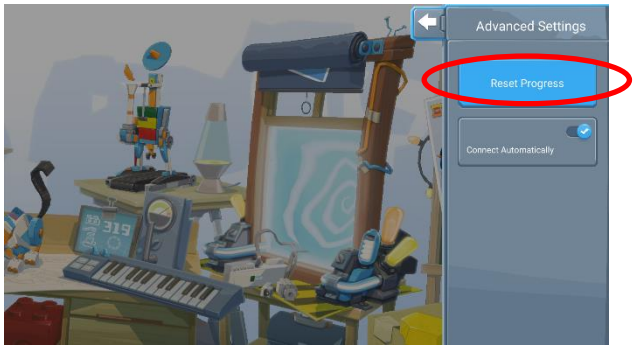
1. In the Main Lobby Screen click on the Settings icon on the top right corner.



2. Click on the Advanced Settings Option.



3. Click on Reset Progress (please note that a warning message will NOT be given).



Replacement parts:

Lego parts can be easily lost or misplaced. There are various options when searching for loose parts. You can order from the official Lego website at:

https://www.lego.com/en-gb/page/static/pick-a-brick

Alternatively, there is a large thriving community for selling new and used Lego parts, two of the most known sites are:

1. https://www.bricklink.com/v2/main.page
2. https://www.brickowl.com/

Software Used for the creation of this content:

The software used in order to construct the digital model of REA is **LeoCAD version 19.01**.

The software used in order to create the instructions for building REA and its modifications is **LPub3D version 2.3.12.0.1356**.

Both Software can be freely downloaded from:

https://www.ldraw.org/

Bibliography

1. LDraw.org (n.d.). What is LDraw?.
   Retrieved from: https://www.ldraw.org/

2. Lego (n.d.).
   Retrieved from: https://www.lego.com/en-gb/themes/boost/about

3. Instructables (n.d.). Everything You Need to Know About Color Sensor.
   Retrieved from:
   https://www.instructables.com/id/Everything-you-need-to-know-about-color-sensors/

4. Robotsquare (n.d.). Customizable Line Following Tracks that you can print yourself.
   Retrieved from: http://robotsquare.com/wp-content/uploads/2012/11/linefollowtiles.pdf

5. Roboticlab (n.d.). Color sensor.
   Retrieved from: http://home.roboticlab.eu/en/examples/sensor/color

6. Remote control vehicle (n.d.).
   Retrieved from: https://en.wikipedia.org/wiki/Remote_control_vehicle

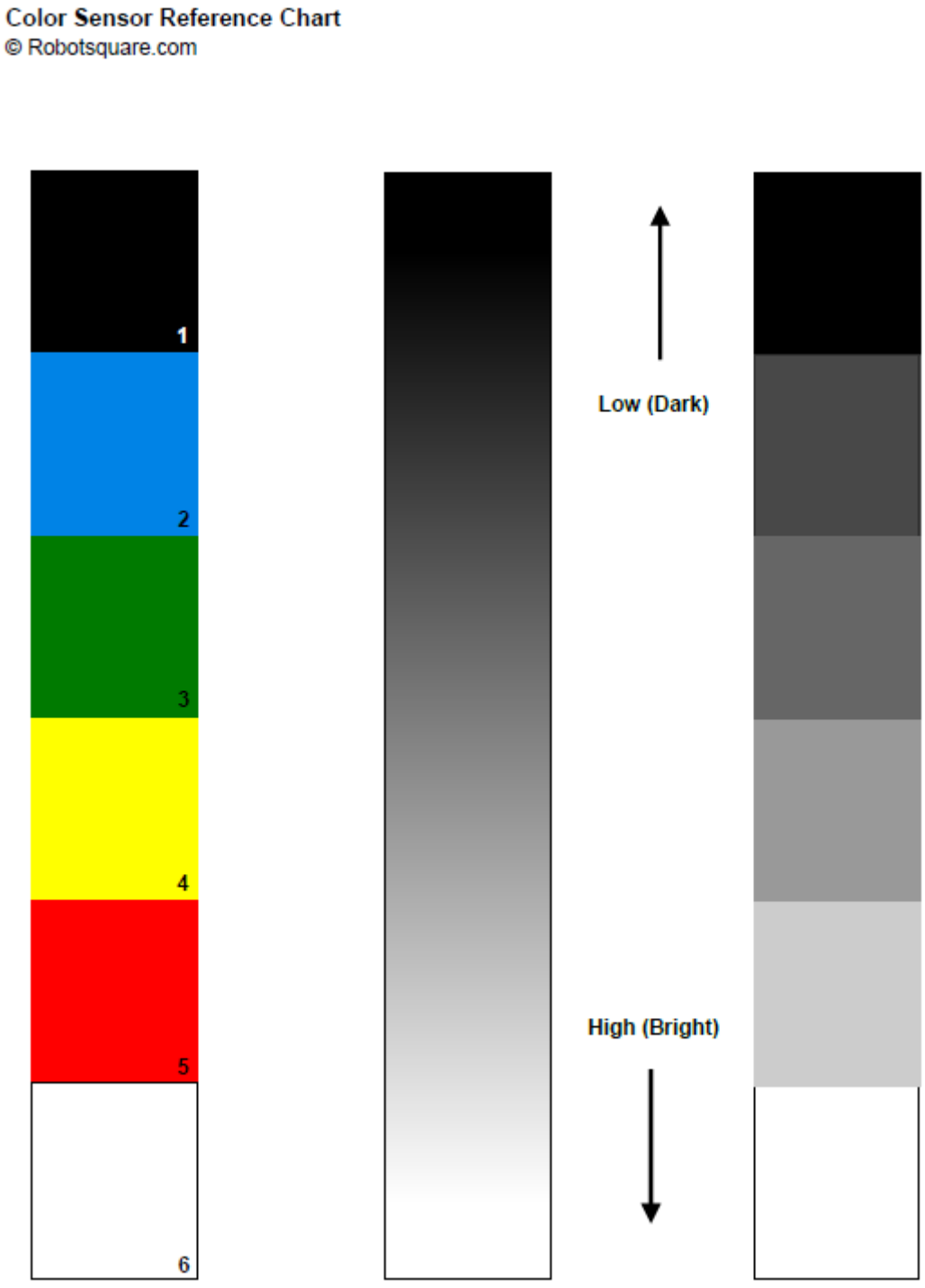
7. Robotshop community (2018). Basics: How Do I Control My Robot?.
   Retrieved from:
   https://www.robotshop.com/community/tutorials/show/basics-how-do-i-control-my-robot

8. Sensor (n.d.).
   Retrieved from: https://en.wikipedia.org/wiki/Sensor

9. Sensor Simple (n.d.).
   Retrieved from: https://simple.wikipedia.org/wiki/Sensor

10. Tynker coding for kids (n.d.). How to Teach Variables to Kids.
    Retrieved from:
    https://www.tynker.com/blog/articles/ideas-and-tips/how-to-teach-variables-to-kids/

Index

http://ev3.robotsquare.com/color.pdf

The following Chart can be printed and tested with the Lego Boost Color Sensor.