The DIY automobile project (Level 3)

Worksheet for students

Team:……

Aim: Creation of a DIY automobile that moves on different angles and/or geometrical shapes

There is no need to upgrade your DIY automobile in terms of electrical wiring. You simply need to focus on the programming part. Good luck and enjoy!

Time for programming!

Open mBlock and connect your Arduino.

Let's further explore the turning mode/functionality/process that was introduced in the previous level. Try the script below. Change the value of the time argument between **turn right** and **Motors Forward** blocks, and document your observations in the table below.

Time (sec)	0.3	0.5	0.8	1.0	Other……
Estimated angle (degrees)					

Tip 1: In case you cannot estimate the angle, you can sketch the trail of the DIY on a paper, or you can use a tape to recreate/represent the trail on the floor.

Tip 2: Bear in mind that different conditions (i.e. friction of the surface) may affect the movement. Experimentation, trial and error are encouraged. You may note down your observations to a new row on the table above.

Which time value is ideal for making the automobile perform 90 degrees turn (i.e., vertical angle)?

Check the values that your classmates identified as optimal. How do you explain possible differences?

The DIY automobile moves on square:

Let's try to program the DIY automobile to move on specific geometrical shapes (i.e. square, triangle).

Based on the aforementioned script and your observations, try to think how a new script, that instructs the DIY to move on a square (meaning a shape with four equal sides and four equal angles of 90 degrees each), will look like. Discuss with the members of your team the possible solutions and note your thoughts below. *Keep in mind that the DC motors do not support the parameter of angular turn, therefore you need to use of the parameter of time together with the block commands for doing a turn.*

The following script (see below) is semi-structured. Find the blocks and place them in the correct order into the scripting area, in order to create a script that allows the DIY automobile to move as close as possible on a square.

Notes:

- To do that, the automobile should move on a constant speed (i.e. 200) for a fixed amount of time (i.e. 1 sec) and make a pivotal turn of 90^0 degrees. This procedure should be repeated for four times and then the motors should be stopped.
- When Arduino starts up, the automobile waits for 2 sec (in order to prepare the status of its position).

Think of other shapes that you would like to recreate (triangle, hexagon etc.). In your opinion, what parts of the scripts should be modified? Write your answer below:

The DIY automobile moves on triangle:

Now, let's try to guide our automobile to move in such a way that will abstractly draw an equilateral triangle shape (meaning a shape with three equal sides and three equal angles of 120 degrees each).

The following script (see below) is semi-structured. Find the blocks and place them in the correct order into the scripting area, in order to create a script that allows the DIY automobile to move as close as possible on triangle.

Notes:

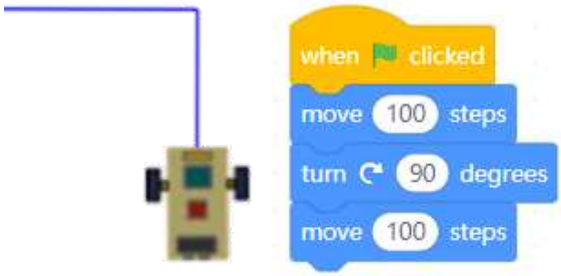
- The automobile should move on a constant speed (i.e. 200) for a fixed amount of time (i.e. 1 sec) and make a pivotal turn of 120^0 degrees. This procedure should be repeated for three times and then the motors should be stopped.
- When Arduino starts up, the automobile waits for 2 sec (in order to prepare the status of its position).

Tips zone

Using the mBlock sprite

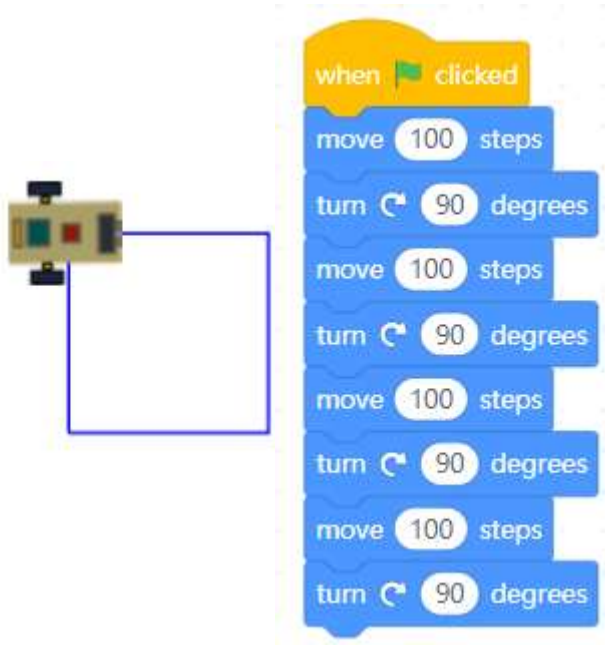
You can use the mBlock sprite as a familiarization (intermediate) stage towards the concept of angular turning.

a. Creating script for make a pivotal turn of 90 degrees



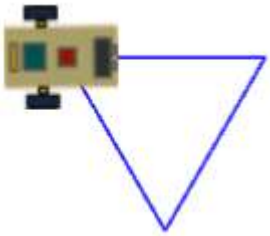
Use the move and the turn blocks from **Motion blocks (blue blocks)** to assemble a script that will make the sprite (an image of the DIY in the example) to move for 100 steps, make a pivotal turn of 90 degrees and move again for 100 steps. The present script can be the first step for the construction of a squared shape.

b. creating a script for inscribing a square



This script helps to understand the concept of a squared shape construction through the implementation of mBlock sprites block commands. The sprite is scripted to *move for 100 and* make a pivotal *turn of* 90 (degrees), for four consecutive times. You can either choose of repeating the aforementioned block for four times, creating an 8-line code, or simplify the procedure by using the **Control block "Repeat"**.

c. creating a script for inscribing an equilateral triangle



This script helps to understand the triangle construction. For this purpose, the instructions *move 100 and turn* 120 degrees are repeated three (3) times.

Tips zone

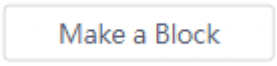
Programming blocks

This is an Arduino extension Event block that executes the subsequent script when Arduino board starts up.

This block sets the output of the selected PWM pin to the specified value.

PWM signals can be used to control the speed of DC motors. Pins 3, 5, 6, 9, 10, and 11 of Arduino Uno can be used as PWM output. The range of values varies from 0 to 255, where 0 indicates the duty cycle of 0%, and 255 the duty cycle of 100%

Sets the output of the selected digital pin to low (false) or high (true) level.

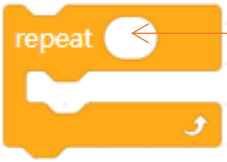


Click on **Make a Block** command to create a procedure that contains a number of consecutive commands (i.e. Move Forward).

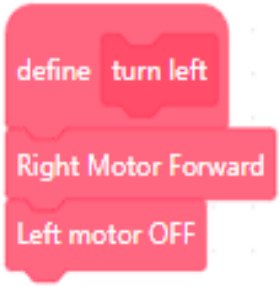
Drag the needed function blocks and assemble them under the hat block "**define()"** to set a new procedure (i.e. all the needed functions to make your Automobile move forward).

Use the created procedure (i.e Move Forward) into the main code, under the Event hat block. When the procedure runs, mBlock will run the blocks below the corresponding Define block.

Click on **Make a Variable** command to create



This is a repeat loop from the Control menu. The commands/blocks that will be placed in the "repeat construct" are repeated based on a defined value of times.



Through this block, a number of consecutive commands are assigned to the "turn left" procedure, instructing the DIY automobile to turn left (pivot).

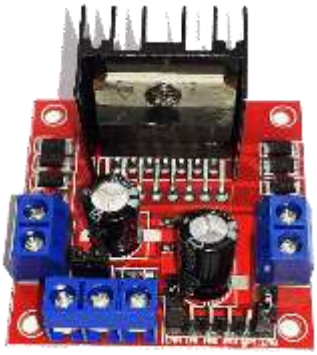
Through this block, a number of consecutive commands are assigned to the "turn right" procedure, instructing the DIY automobile to turn right (pivot).

A number of consecutive commands are assigned to “spin left” procedure, instructing the DIY automobile to make a rather quick turn on the left, by setting the right motor to move forward and the left to move backwards.

A number of consecutive commands are assigned to “spin right” procedure, instructing the DIY automobile to make a rather quick turn on the right, by setting the left motor to move forwards and the right to move backwards.

Electrical components

The following table is an index containing the main electrical components that need to be implemented for accomplishing the present activity.

	**DC Motor**
	**L298n driver**
	**Arduino Sensor Shield**

ROBOSCIENTISTS PROJECT

Motivating secondary school students towards STEM careers through robotic artefact making

Erasmus+ KA2 2018-1PL01-KA201-051129

Creators

Chrysanthi Papasarantou (EDUMOTIVA), Konstantinos Salpasaranis (EDUMOTIVA), Rene AlimisI (EDUMOTIVA)

Declaration

This report has been prepared in the context of the ROBOSCIENTISTS project. Where other published and unpublished source materials have been used, these have been acknowledged.

Copyright

Funding Disclaimer

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.