

DuploCAR Worksheet 1

Walking Robot Car with Duplo® - Claudio Gasparini - www.cad-tutor.com/duplocar 1

Worksheet of DuploCar

A drawing Robot made with Lego Duplo blocka

S1. Uso di Kitronik Servo:Lite Board

In this project I used the **Kitronik Servo:Lite** Board expansion board because it is easy to use, with a very low cost and because it is equipped with many complete control software.

The link is: kitronik.co.uk/5623

It is necessary to load the control software of the *kitronik-servo-lite* expansion board from the **MakeCode Extension** menu.

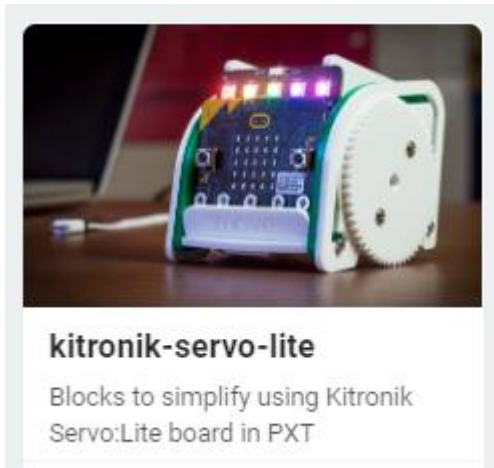


Figure 1 Card to load inExtension of MakeCode

S2. How to calibrate servo motors

To have a continuous rotation of the wheels I used the continuous rotation *servomotors (motors)*, also called "360 degree servo".

Continuous rotation servos are controlled by defining the rotation speed and not the angular distance as with normal 180° servos.



Figure 2 - Servo speed definition diagram

Educational Worksheet

Primary

Secondary



20-35 min

Teacher Support

Key objectives

The students will be able to control movements using principles of geometry

Time: 20-30 min

Working group: the whole class

Materias: Lego Duplo + micro:bit and Kitronik Servo:Lite

Key Competences

Problem solving: identify the appropriate problem-solving strategies

Project site:

www.cad-tutor.com/duplocar



Video on YouTube



Files .hex

DuploCAR Worksheet 1

Walking Robot Car with Duplo® - Claudio Gasparini - www.cad-tutor.com/duplocar 2

In the following figure with **button A** the servos **P1** and **P2** both rotate forward at maximum speed for 1 second and then stop. With **button B** they rotate in reverse.

With **button B** they rotate in reverse. Both after 1 sec. they stop because speed **90** means **STOP**.

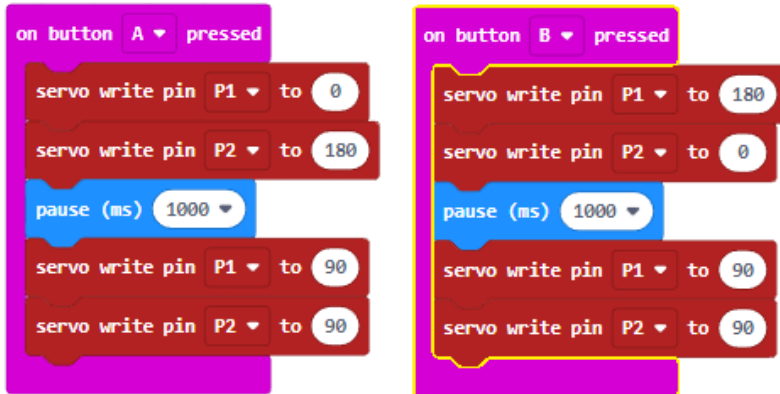


Figure 3 Forward rotation and stop

Note 1: Remember that the servos are mirrored so the values must be reversed to have the same direction.

NOTE 2: you can never have an absolute precision between the angle values **indicated** in the code compared to those obtained in practice by the robot for various reasons such as: battery charge, not always constant friction of the wheels and speed of movement.

Coding

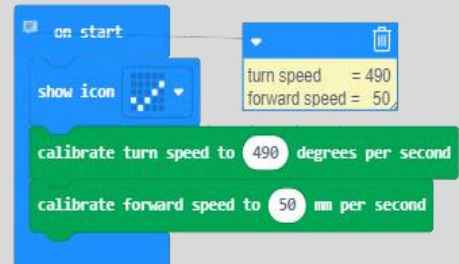


Figure 4 How to calibrate servos

With continuous rotation servo motors it is necessary to calibrate the motors for the **speed** of linear and **angular displacements**.

In my configuration with the type of servo I used, these are the **turn speed (angle)** and **forward speed (forward)** values to have a good match of the movements.

These two simple "calibrated" instructions will be included in each program in "**On start**".

DuploCAR Worksheet 1

Walking Robot Car with Duplo® - Claudio Gasparini - www.cad-tutor.com/duplocar 3

S 3. Draw a square

Drawing a square is a very simple algorithm: go ahead of **L** and spin **90°**

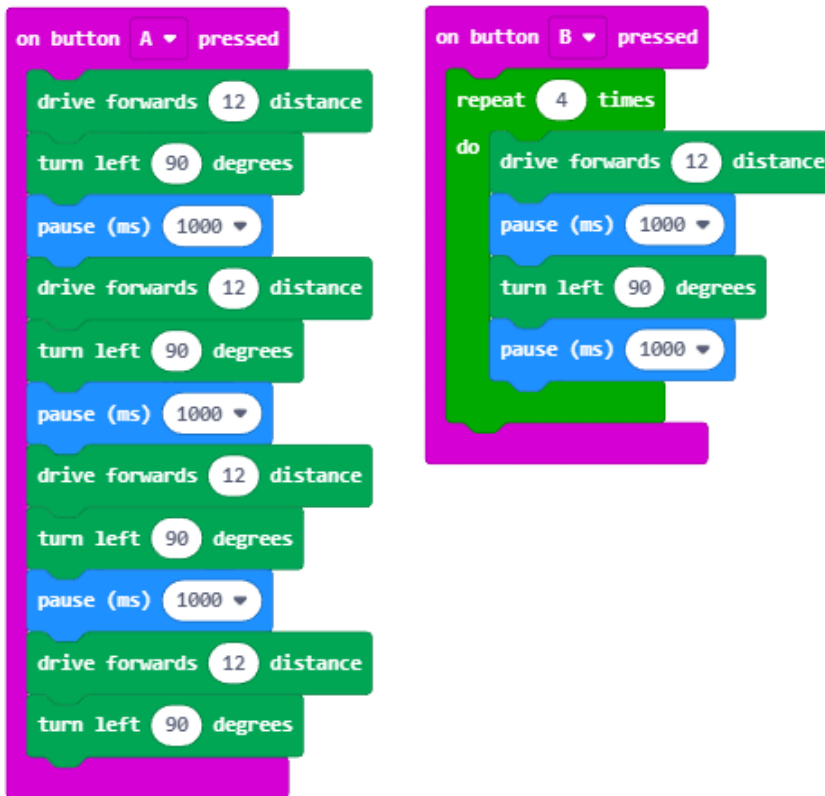


Figure 5 Left extended sequence and right in a repeat loop

In the figure on the left, **button A**, the drawing of the square is made with all the individual instructions while on the right with the **B button** the 2 main instructions (*ahead of 12, turns left 90°*) are inserted in a loop of **repeat** 4 times.

Note that the **instruction pause (ms) 1000** (*pause 1000 milliseconds i.e. 1 sec*) is necessary because it allows the engine to position itself. The pause of 1 sec. is the minimum to have an adequate movement.

To document these tutorial I have included in **YouTube** and **GitHub** the examples of movements and software.

Graphics with Scratch

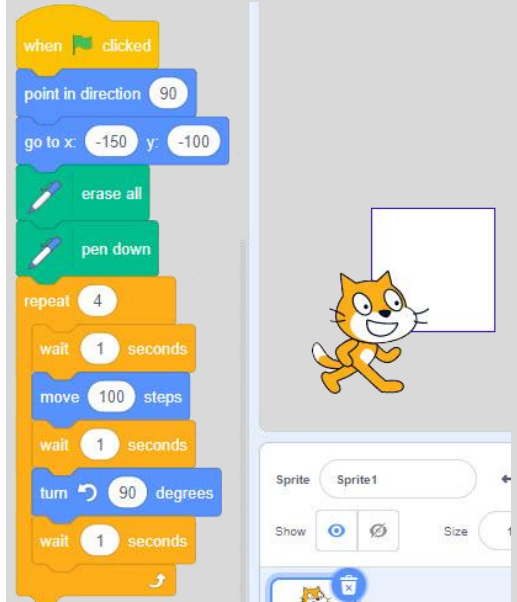


Figure 6 Square with Scratch

Even with **Scratch** you can use the **Repeat** function to draw a square in a reduced way, with the following commands:

- **forward** 100
 - **turns** 90°
- and repeated **4 times**.

DuploCAR Worksheet 1

Walking Robot Car with Duplo® - Claudio Gasparini - www.cad-tutor.com/duplocar 4

S 4. Drawing a polygon

To draw a polygon you use an **Function** that allows you to receive two parameters input: **sides** (*number of sides*) and **length** (*length of sides*).

In this way it is possible to draw, with a single function, many types of polygons indicating in input as variables the numerical values of **sides** and **lengths**.

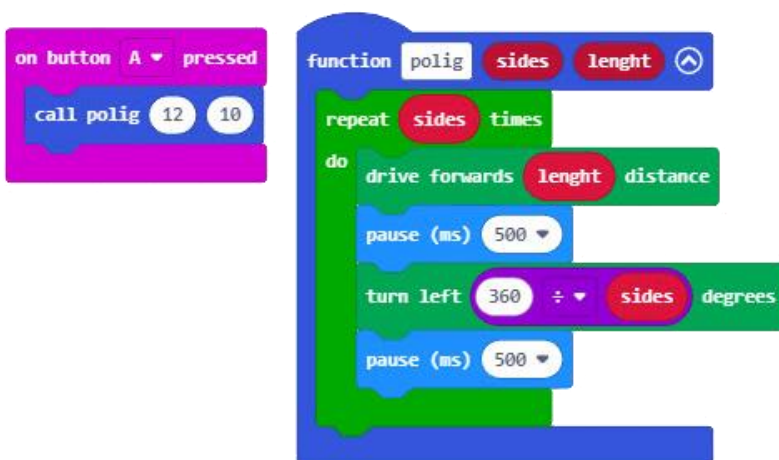


Figure 7 Polig function

The **Polig function** is called with the **A button** where you can specify n. 12 sides with **length 10**.

Graphics with Scratch

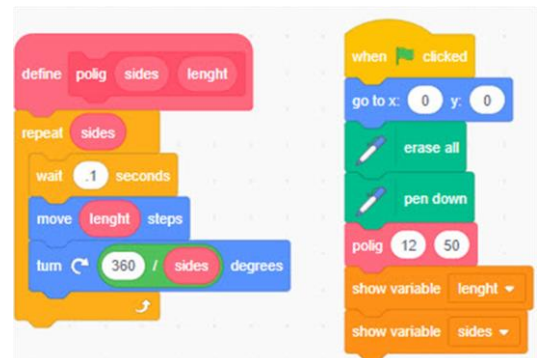


Figure 8 Drawing a Polygon

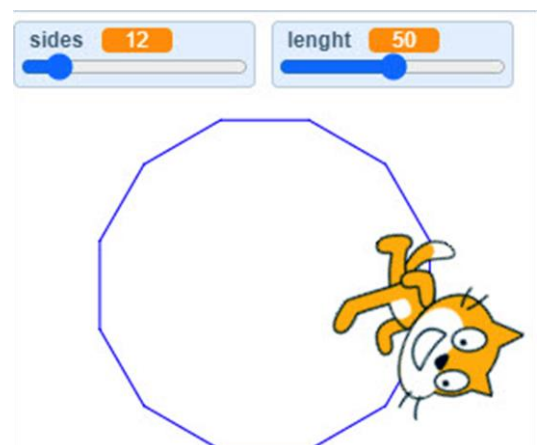


Figure 9 The 12-sided Polygon

The **wait** in **Scratch** has the function of slowing down the drawing to allow you to control the individual movement.

In Scratch, the **Polig Function** is created with the **Make a block** command

Exercises 1.2

1. Enter a third parameter corresponding to the pause time
2. Is it possible to insert the angle instead of the number of sides?

DuploCAR Worksheet 1

S 5. Draw a circle

The formula for calculating a circumference is: $2 \pi r$
In figure 10 the circle that is drawn has values:
radius = 20 and circumference divided into **30 segments**.

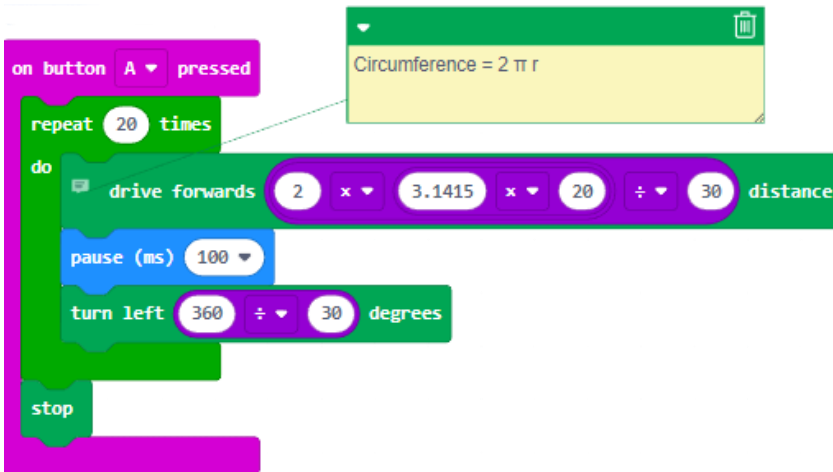


Figure 10 Circumference divided into 20 segments

No **function** is used here and this means having to change the values individually.

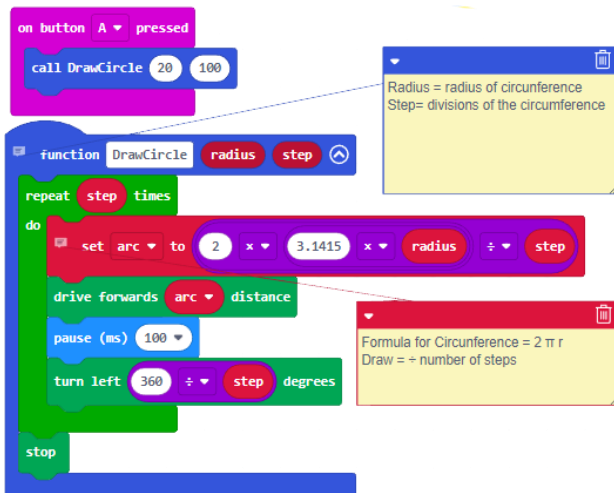


Figure 11 Circumference Function

I used a **Function** with two parameters: **Radius** and **Step** where *Radius* is the radius and *Step* is the number of segments of the circumference. The higher the **Step** value, the higher the tracking accuracy.

Note: The dimensions of the drawing cannot be as accurate as the plotter of course.

Graphics with Scratch

The same algorithm is also used in **Scratch** with a function with only one variable of the radius of 100.

The circumference is divided into 360 segments to have greater accuracy of the drawing.

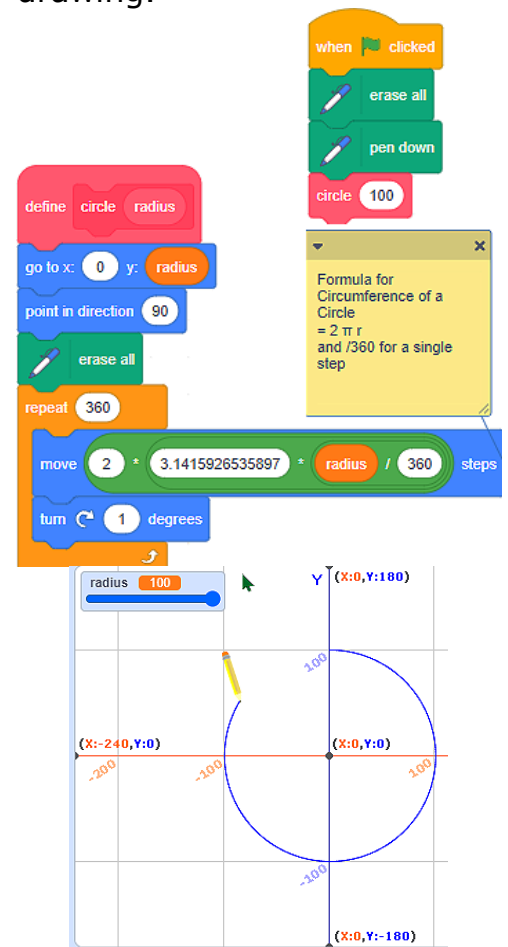


Figure 12 Circle Drawing

DuploCAR Worksheet 1

Walking Robot Car with Duplo® - Claudio Gasparini - www.cad-tutor.com/duplocar 6

S 6. Drawing a fractal with MakeCode

It's a lot of fun to have the **DuploCAR** car robot draw a fractal because the design that comes out of it is assumed random and random in all its aspects.

Since it is not possible to obtain a precision of the drawing for the reasons mentioned above, the drawing that is obtained is more like an "art drawing" than a fractal graphic.

Have fun!

On the first page I have attached the link to the software in **GitHub** and to some videos posted on **YouTube**.

Graphics with Scratch

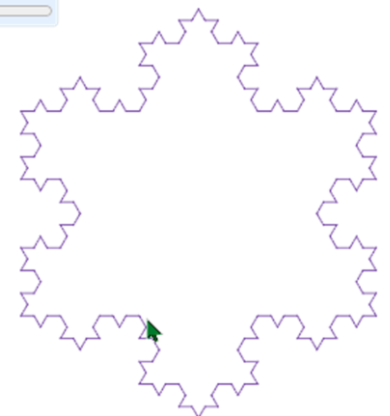


Figure 13 Koch curve

Esercizi 1.4

1. Insert a comment in a block
2. What changes the "External inputs" function?
 - 1.

DuploCAR Worksheet 1

Walking Robot Car with Duplo® - Claudio Gasparini - www.cad-tutor.com/duplocar 7

S 7. Drawing a fractal with Scratch

With *Scratch* the fractal drawing is much more precise and clear by being able to control the direct effects of the variation of the variables.

A mandala is a series of geometric designs that are generated by the concentric rotation of simple flat figures.

In **Scratch** you can find many examples of **Koch Snowflake** of the most varied shapes and colors.

A fractal design can also be reproduced with the 3D printer with a varied depth and thickness.

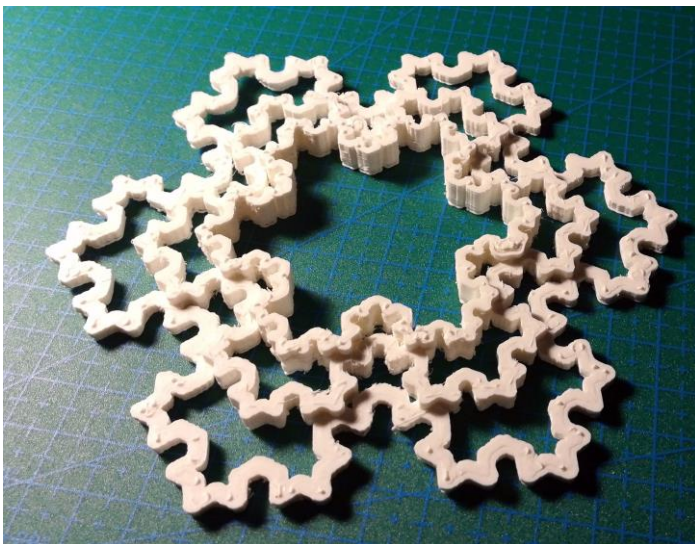


Figure 15 Koch curve printed with 3D printer

Graphics with Scratch

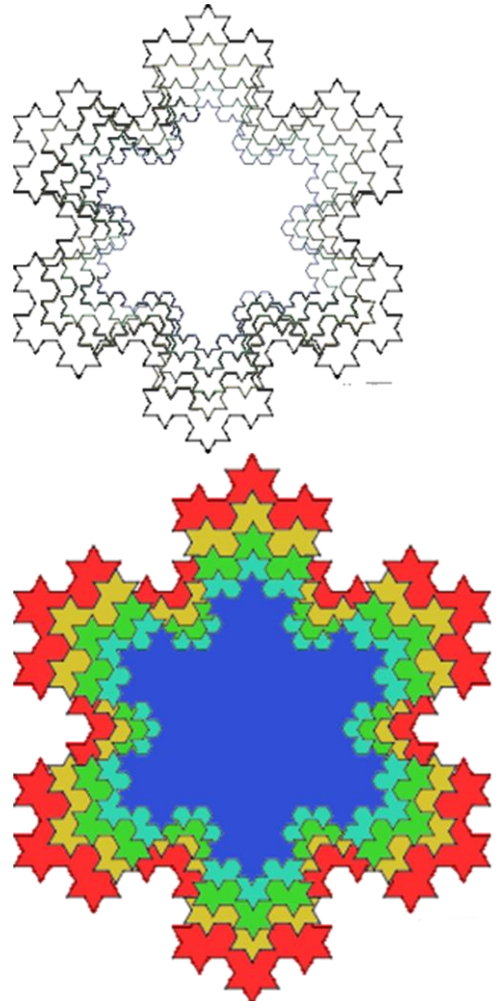


Figure 14 Concentric repeated Koch curve

Esercizi 1.5

1. Create fractal copies that you can then color with a painting program
2. How about reproducing the same design with 3D printed?

DuploCAR Worksheet 1

S 8. Draw a mandala with Scratch

A **mandala** is a concentric design decorated with A mandala is a design composed of the repetition of basic geometric figures repeated according to a circular pattern.

With Scratch the construction is very simple because it is based on two **loops (repeat)** inserted one inside the other with the geometric figure continuously rotated by a limited angle.

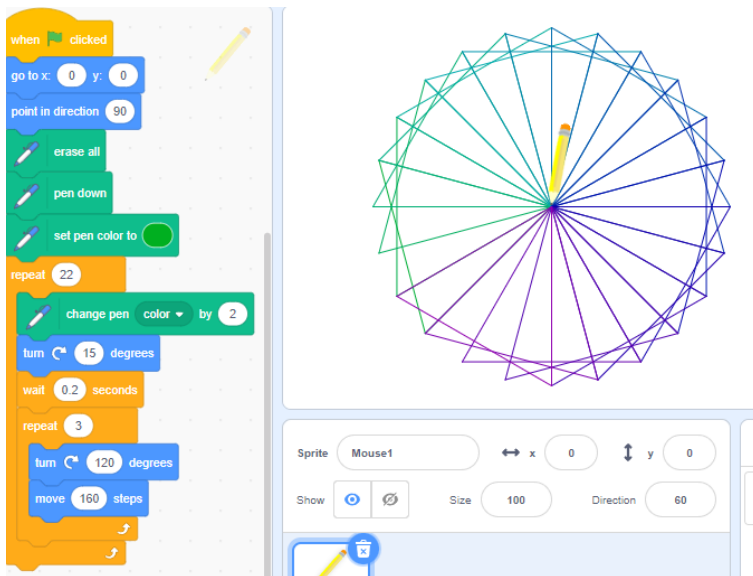


Figure 16 Simple Mandala