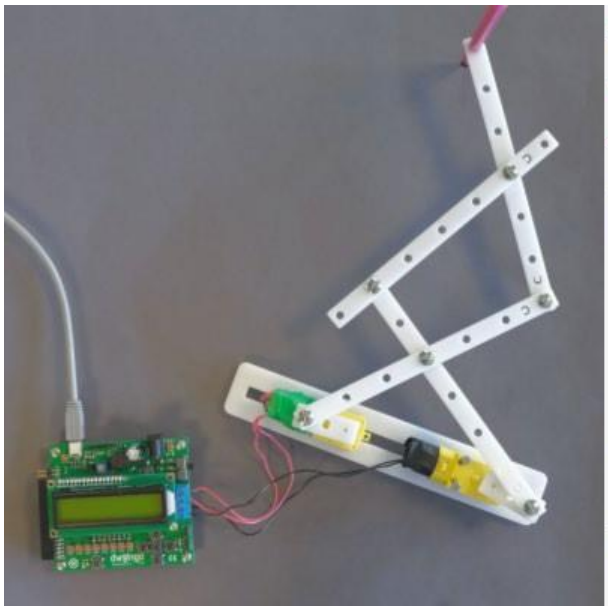
What does a robot consist of?

A robot consists of hardware (its body, with wiring, computing unit and batteries) and software (the program that controls it).



```
#include <Wire.h>

#include <Dwenguino.h>

#include <LiquidCrystal.h>

void setup()
{
  initDwenguino();

  dwenguinoLCD.setCursor(0,0);
  dwenguinoLCD.print(String("Ben je wat"));
  dwenguinoLCD.setCursor(0,1);
  dwenguinoLCD.print(String("Nieuwsgierig?"));
  while(digitalRead(SW_N)){}
}


void loop()
{
    dwenguinoLCD.setCursor(0,0);
```

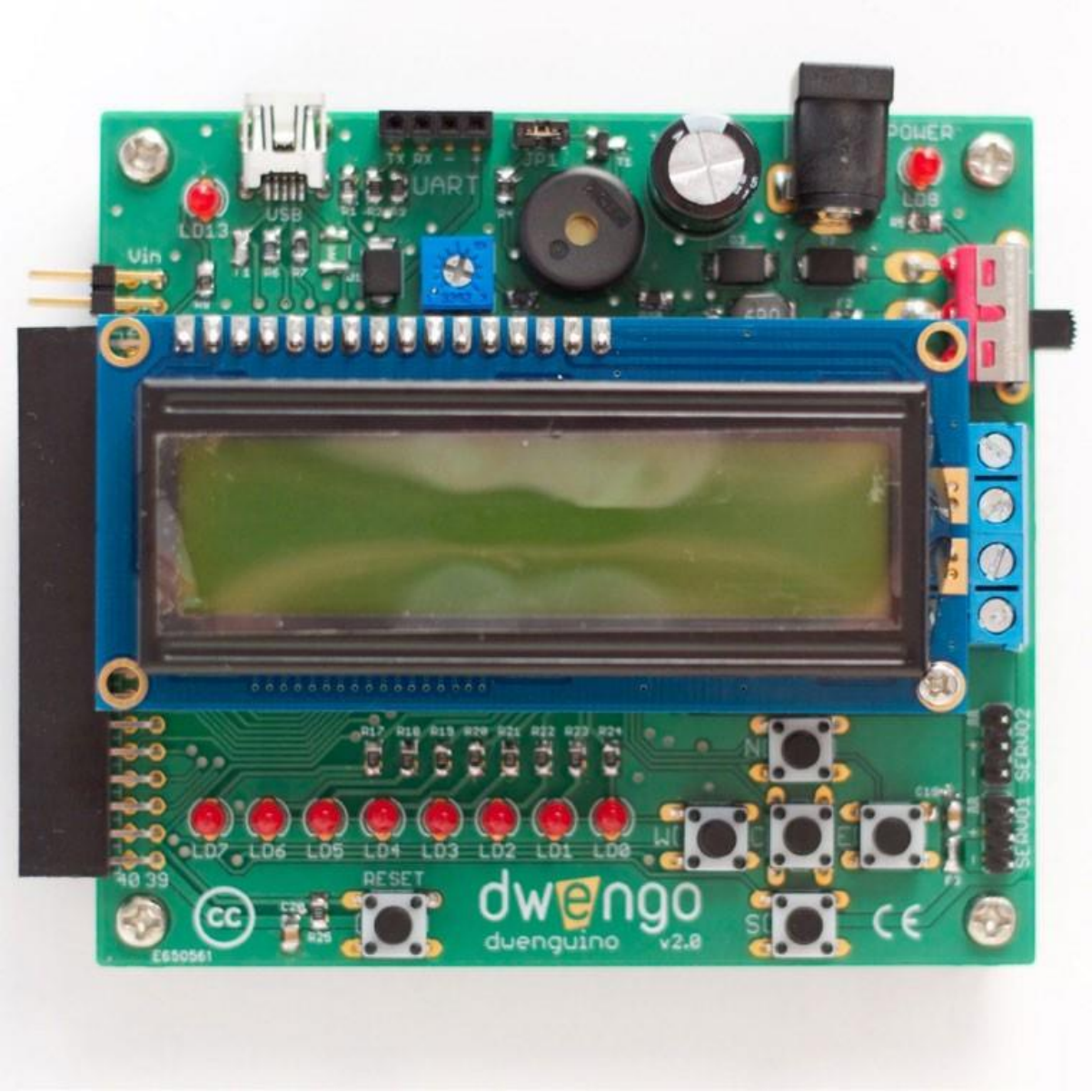
The 'body' of a robot contains mechanical parts that can be made of many materials: wheels, an arm, a head, ...

The robot has sensors (inputs) to "feel" and actuators (outputs) to "act". One can equip a robot with distance sensors, ground sensors, touch sensors, light sensors or sound sensors. Examples of actuators include an LCD screen, a buzzer or a servo motor.

Your computer also has inputs and outputs. The keyboard and mouse are examples of inputs; the screen is an output.

The computational unit is the brain of the robot, it is used by your robot to make "decisions. For this, you use a processor (like in a computer) or a microcontroller (like on the dwenguino), for example.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them. Project Number 2022-1-CY01-KA210-SCH-000084731



The wiring ensures that the computing unit, sensors and actuators are connected. How the robot controls its actuators depends on the information it collects from its sensors and how the computing unit is programmed. The batteries in the robot supply the motors with the necessary power.

A robot can perform a certain task only if its computing unit is programmed to do so. What the robot does is not necessarily fixed: you can reprogram the computing unit so that the robot performs a different task. Moreover, that task may depend on the information that the microcontroller receives through its inputs.

Wondering what your drawing robot is capable of doing? Let's find out!

STEAM4AUT

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

S·T·E·M EDUCATION CYPRUS PRIVATE INSTITUTE

NOVATEX SOLUTIONS.EU

GHENT UNIVERSITY

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

Preparation - Building a dwengo social robot.

20 - 30 minutes during class

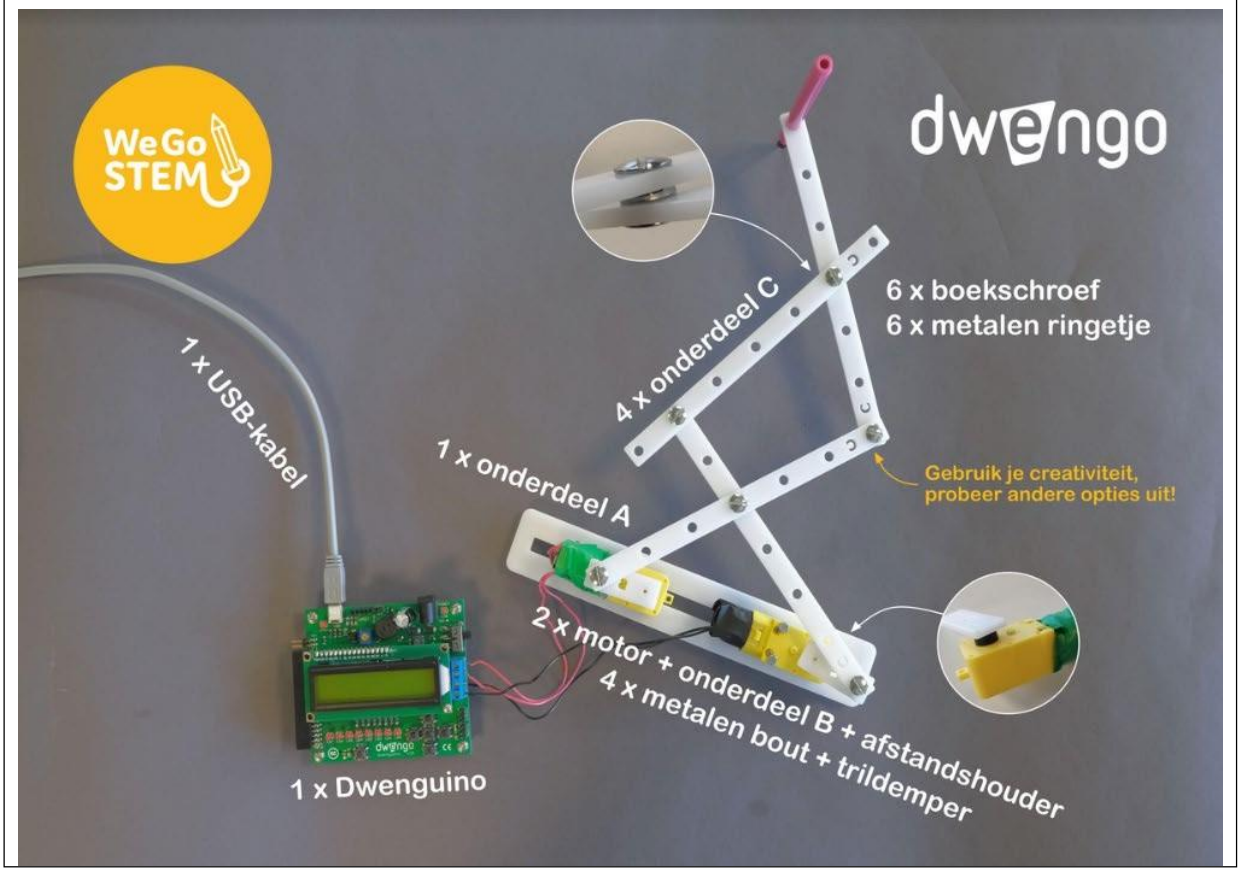
What happens in class?

In this activity, students get to work with physical equipment for the first time. They learn about the different components of a robotic arm and have to assemble them correctly. While doing so they build their drawing robot.
Each group is given a construction diagram. Based on this diagram, the children can usually figure out for themselves how to build the drawing robot. If they do get stuck, you can provide some additional explanation. The steps needed to build the robot are shown in the following video (in Dutch only).

Preparation

Materials for the classroom

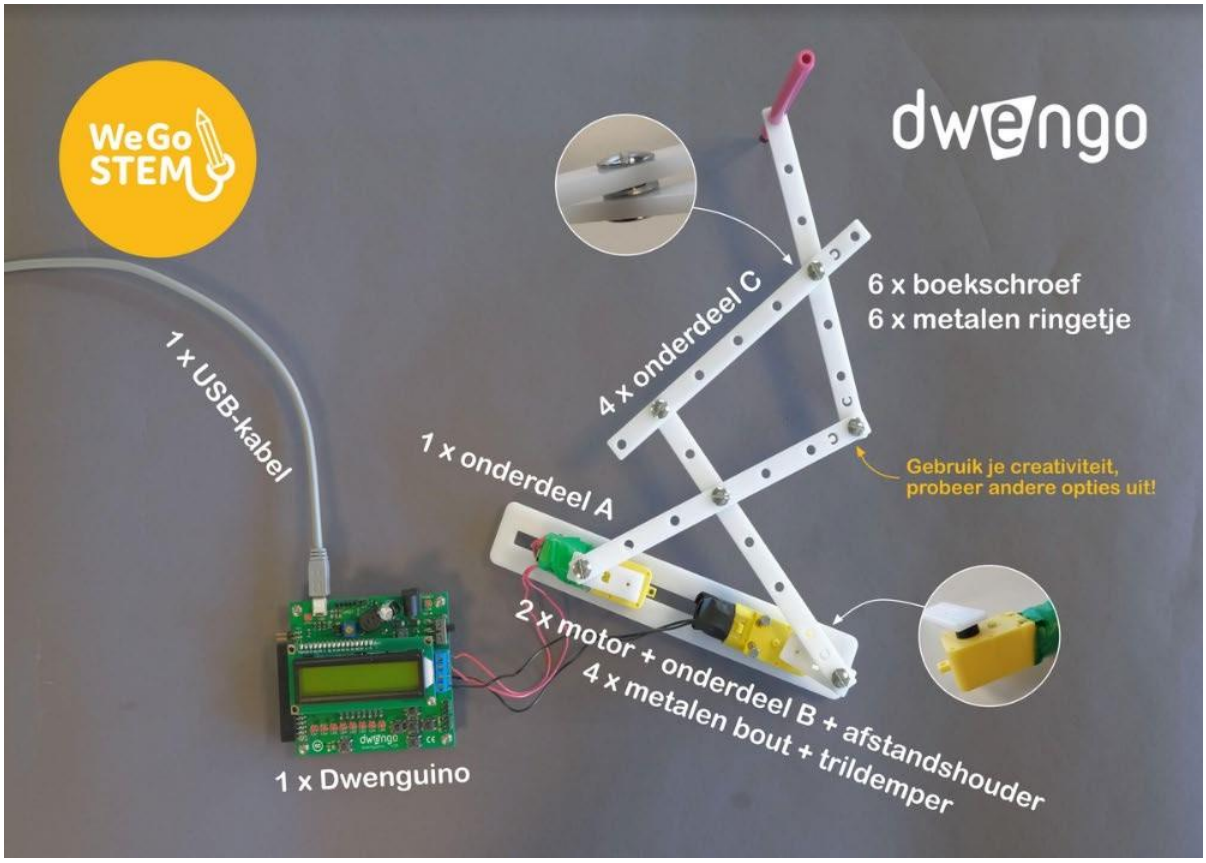
Box of robot components
Photo of a finished robot



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

Building a Drawing Robot

Now that you know how a robot works, you can start building the robot itself! To do this, use the picture and the parts from the box.



Will you manage to build the robot just by looking at the picture? Do you need some extra help? Then watch this video (in Dutch).

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

Preparation - Programming the Drawing Robot

40 min during class

What happens in class?

The students now have a robot and want to activate it. But first, they need to program it. This is the subject of the next steps of this learning pathway.

Step by step

Students will learn to work independently and develop an inquisitive mindset. First in a simulator, then in real life. In several steps, they discover different ways to control their robot. For example, they learn:

1. To see the transition from code to real movment

   Tip: Have students program only when they have finished building their drawing robot. As a teacher, you don't have to spoon-feed your students. Feel free to let them experiment.

Preparation

The children must get started on their own in the learning path. They learn to work with the motors. For each of these functions, students are given information and exercises. These instructions should suffice.
Should the students still have questions, they can approach the teacher. We therefore recommend that you, as a teacher, are familiar with the basic exercises of the course. This way you will be able to guide the students in the best possible way. You don't have to follow the entire track.

Materials for the classroom

Computers with internet connection.
Dwengo social robot

Programming the Drawing Robot

Now that your robot is assembled, we're going to program it. We're going to tell it very precisely what to do. But because the robot doesn't speak human language, we'll first learn programming language. We'll do that in dwenguinoBlockly, a programming environment for kids like you!

dwenguinoBlockly has two big advantages:

1. It is age-appropriate (third-grade primary and first-grade secondary).
2. It is a graphical programming language. Programming is as simple as dragging blocks on the screen.
3. There is a simulator. On the screen, you can already see what the robot will do.

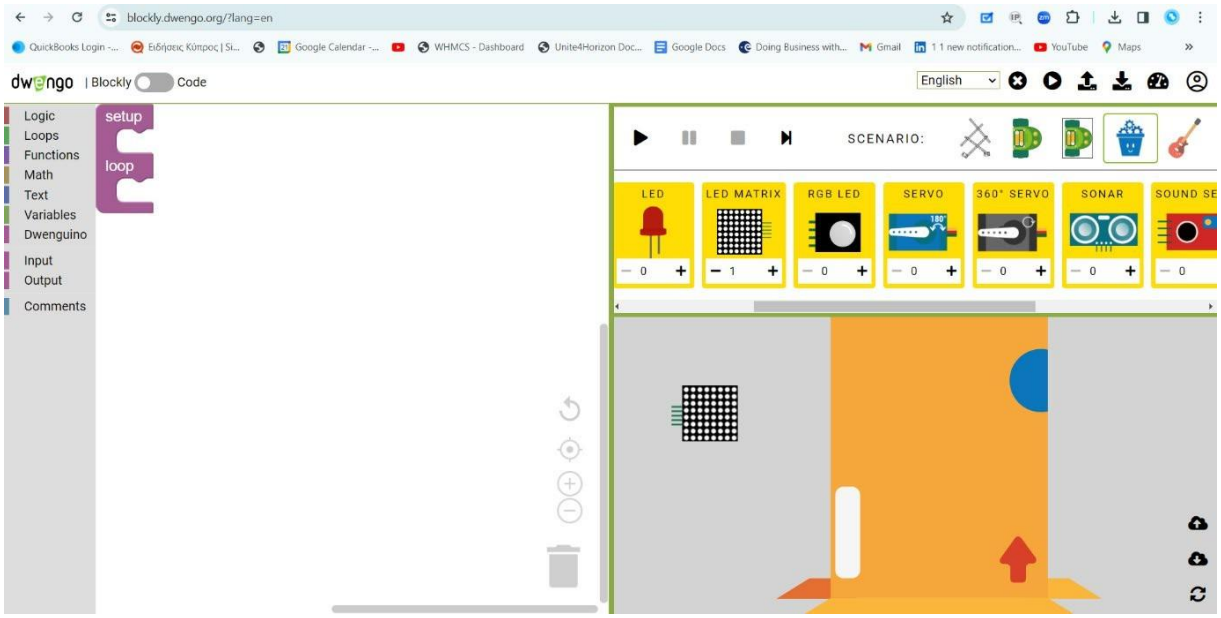
In the next steps of this learning pathway, you will learn to work with dwenguinoBlockly. First in the simulator, then with your robot.

Good luck!

dwenguinoBlockly: A programming environment

The programming environment with the simulator is available online at https://www.dwengo.org/dwenguinoblockly .

Below is a screenshot of the environment with a description of the various components.



1. The toolbox: In this menu, you will find the different code blocks. The menu is divided according to categories, each containing a specific kind of block. For example, you can find all the blocks to control the social robot.
2. The code field: Here you can find the program you are creating. The 'get ready/repeat' block is already there.

Only code placed in the get ready' and 'repeat' sections of this block is executed. Code in any other place will not be executed. So to program, drag and drop blocks from the toolbox into the code field and snap them into the 'get ready/ repeat' block.

3. Main menu: This menu allows you to perform actions such as saving your code (using), loading it back in (using), or opening and closing the simulation environment (using).
4. The simulator menu: Here you will find the buttons to start and stop the simulation with the buttons ▶ and ■. It also allows you to choose a specific scenario in which to run your code.
5. The simulation window: In this window you see a virtual robot and often a virtual microcontroller board, the dwenguino, on which you can run your code. In the picture, the drawing robot scenario is selected. At the top you see a virtual dwenguino board, at the bottom a virtual drawing robot that you can program.

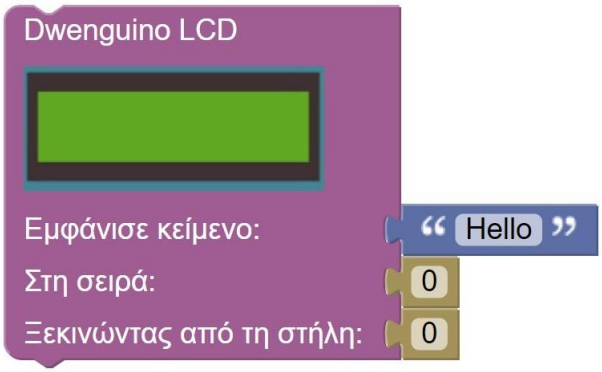
So in the toolbox you can find the blocks you need to create programs. You have to drag these blocks from here and then place them in the desired order.

Throughout the exercises you will need new blocks, you find these blocks in the toolbox. The Dwenguino toolbox for example will contain blocks that you will need for your robot.

Using the simulator

Now that you know what's where, you can start programming!

- In the simulation environment, choose the drawing robot scenario (spirograph).
- In the Dwenguino category, look for this block:



- Drag this block into the code field and snap it into the "get ready" section of the "get ready/ repeat" block.
- You just wrote your first program!
- Run this program with the simulator by clicking on the play button ▶ in the simulator menu.

After this exercise you will know the basics. You can take blocks from the toolbox and add them to a program in the code field. You know how to execute that code in the simulator and you see what the code does using the simulator.

Once a program works in the simulator, you can also try it out on a real dwenguino! The following section describes how to upload a program from the simulator to the dwenguino.

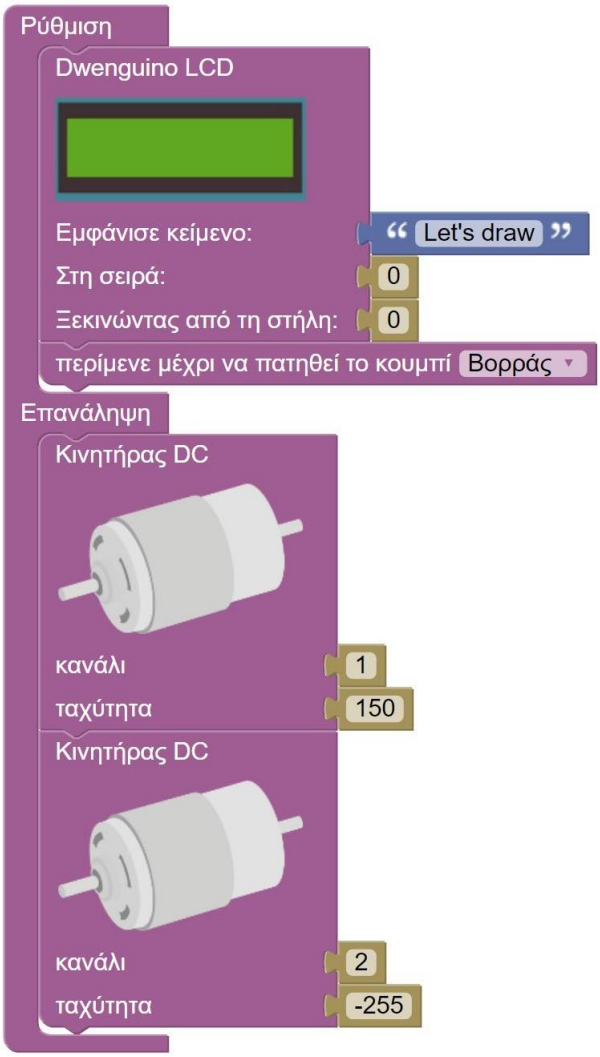
Uploading code to the dwenguino

The video (in Dutch) shows you how to upload a program from any computer and browser to your dwenguino. You must go through all the steps as shown in the video!

- Write the program.
- Download the program.
- Open downloads, there is now a .dw-file there
- Connect the dwenguino and press RESET + SOUTH, then release the reset button.
- The dwenguino is between your folders as a USB drive;
- Copy the .dw file to the dwenguino;
- Press RESET

Task 1: Motors

Build the following program and see what it does.



Task 2: Testing

Let the drawing robot draw faster.

Have the drawing robot draw slower.

Run the motors of the drawing robot to the other side.

Task 3: Extension

Run one of the motors to the right and the other to the left.

Run one of the motors and not the other.

Experiment with the speeds of the motors.

After this exercise, you will be able to control the drawing robot. You now understand that the drawing obtained depends on the rotation sense and the speed of the motors.

If anything is still unclear, this video summarizes everything again.

Packing up

Rest assured, you don't have to start packing up immediately now that you've finished the exercises! Just keep trying different programs until your teacher lets you know it's time to finish.

First, reset the dwenguino by clicking the cross  in dwenguinoBlockly. Here you follow the same process as when you upload a program. Upon a successful reset, 'dwenguino ;)' will appear on the LCD screen.

Next, take everything apart and tidily put everything back in the box. This way, the next class can have a smooth start.