

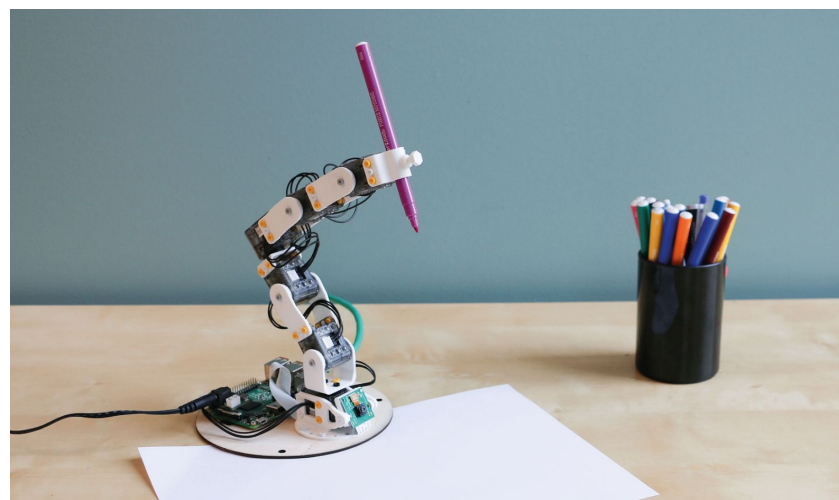
## **EARLY Kennsluáætlun**

**Viðfangsefni:** teikning með Poppy Ergo Jr

**Markmið:**

Nemendur læra að:

- Setja saman Poppy Ergo Jr vélmenni
- Nota grunnskipanir snap!
- Forrita einfaldar hreyfingar Poppy



**Færni nemenda sem þróast á meðan á vinnu stendur.** Aðalnámskrá grunnskóla →

Við lok 7. bekkjar getur nemandi:

- sýnt sjálfstæði í vinnubrögðum undir leiðsögn og í samvinnu með öðrum

- nýtt sér mismunandi tæknibúnað á hagkvæman og fjölbreyttan hátt
- nýtt hugbúnað og forrit við einfalda vinnu
- nýtt hugbúnað/forrit við miðlun þekkingar á skapandi og skýran hátt
- notað hugtök og aðferðir rúmfræðinnar til að útskýra hversdagsleg og fræðileg fyrirbrigði
- áætlað og mælt horn, þyngd, tíma og hitastig með viðeigandi mælikvarða og dregið ályktanir af mælingunum

### **Hæfnimarkmið og þjáfun í þessu verkefni:**

- Að búa til vélmenni úr þeim hlutum sem liggja til grundvallar
- Að forrita vélmenni á myndrænan hátt með notkun blokka (e. blocks) forritunar tungumálsins
- Lausnaleit
- Samvinna

**Markhópur:** Nemendur á unglíngastigi (7. - 8. bekkur)

**Aldur nemenda:** 13 - 14 ára

**Fjöldi nemenda:** hámark 10 í hóp

**Lengd (áætlaður tími/fjöldi kennslustunda):** þrjár kennslustundir x 45 - 90 mínútur hver

**Forsendur verkefnis (nauðsynlegur búnaður og upplýsingar af neti):**

- Poppy Ergo Jr með haldi fyrir blýant/lit
- Tölvur með nægilega mörgum breytum til að keyra Snap!
- Stuðnings- og fræðsluefni á netinu: <https://www.poppy-project.org/en/robots/poppy-ergo-jr>

### **Kynning á kennsluáætluninni (ásamt. mögulegum hugbúnaði, aðrir valkostir og áhættuþættir):**

Poppy Ergo Jr er vél-armur, hannaður með menntun í huga. Poppy samanstendur af sex mótorum sem geta framkvæmt fjölda hreyfinga. Hann inniheldur þrjá meginhluti, sem gegna mismunandi hlutverki; a gripper, a lampshade og pen holder. Vélmanninu er stjórnað með Raspberry Pi board með hjálp myndavélar sem tengir það við umhverfi sitt. Hægt er að forrita Poppy með sjónrænum hætti sem veitri nemendum skemmtilega og skapandi námsreynslu. Þrátt fyrir að framleiðendur haldi því fram að auðvelt sé að byggja/breyta vélmanninu og allt sem því tengist sé mjög aðgengilegt og auðvelt í notkun höfum við komist að því að Poppy getur verið flókinn viðureignar. Þetta getur verið kjörin áskorun fyrir bæði nemendur og kennara og hvetur til gagnrýnnar hugsunar, samvinnu og lausnaleitar. Þessi kennsluáætlun ætti að veita ykkur greiða leið í vinnu með Poppy með nemendum á eldri stigum grunnskóla.

### **Áður en vinna hefst (undirbúningur kennara):**

- Undirbúið góða aðstöðu (t.d. tölvustofu skólans) með nægu vinnustöðvum fyrir nemendur.
- Poppy reiðir sig á Snap! (<https://snap.berkeley.edu/>), afbrigði af Scratch. Gakktu úr skugga um að allar tölvur hafi stöðuga nettengingu og að þær geti keyrt Snap (opnaðu nýjan flipa og sláðu inn <http://snap.berkeley.edu/run>).
- Sá fjöldi sem skólinn hefur til umráða af Poppy (vél-armi) hefur áhrif á hversu margir nemendur geta unnið á sama tíma. Hlutar Poppy eru fjölmargir og sumir pínulitlir þannig að ekki ættu fleiri en 2-3 nemendur að vinna að sama vélmanninu. Flesta hluti má 3D prenta til að draga úr kostnaði við smíði vélmannana, frekar ætti að leggja áherslu á að kaupa vélræna parta sem ekki má prenta. Kennarar ættu fyrst sjálfir að fara í gegnum allt ferlið, til að gæta að því að þeir skilji þetta allt rétt, áður en þeir leggja verkefnið fyrir nemendur.

### **Meginhluti kennsluáætlunnar (þrjár kennslustundir):**

Hver eftirtalin kennslustund er einstök og má því kenna eina og sér. Helst þurfa nemendur að komast yfir allar kennslustundirnar en það fer þó eftir getu þeirra og fjölda tölva og vélmanna sem skólinn hefur til umráða. Ef til vill er hægt að útfæra verkefnið þannig að aðeins hluti nemenda fari í það að setja vélmannið saman og restin svo kennd með öllum bekknum.

## Fyrsta kennslustund: Setjið saman Poppy Ergo Jr

Framleiðendur Poppy gáfu út stuttar leiðbeiningar um hvernig hægt er að setja vélmennið saman. Þær ná yfir stillingu mótorá, rafræna samsetningu og uppbyggingu vélbúnaðar:

<https://docs.poppy-project.org/en/assembly-guides/ergo-jr/index.html>

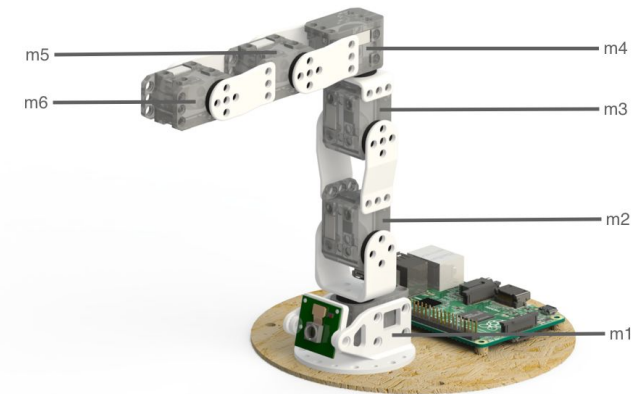
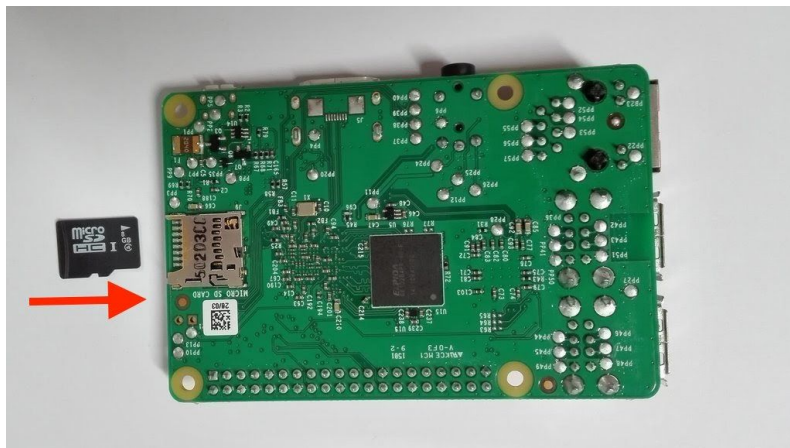
Við mælum ekki með að nemendur lesi þessa vefsíðu sjálfir heldur fái aðstoð kennara til þess vilji þeir kafa dýpra í efnið. Við mælum hins vegar með þessu einfalda myndbandi þar sem möguleikar vélmennisins eru sýndir

<https://edurobots.eu/assembly-poppy-ergo-jr/>

Gott er að horfa fyrst á myndbandið í heild án þess að gera nokkuð. Horfið síðan aftur á myndbandið og byrjið að setjað saman.



[Motors configuration](#): Til þess að mótorarnir séu allir tengdir og auðkenndir á sömu línu verða þeir að hafa sérstakt auðkenni. Það verður að stilla þá alla á sama auðkennið og setja nýtt og sérstakt auðkenni fyrir hvern mótör.



### [Electronic](#)

[assembly](#): Settu micro-SD kortið inn í Raspberry Pi - ýttu micro-SD inn í tengiraufina þar til þú heyrir "click" hljóð.

Mechanical assembly(vélræn samsetning): Hana þarf að framkvæma með sérstökum skrúfum til að tengja/festa saman ólíka hluta vélmennisins. Mikilvægt er að fylgja leiðbeiningunum nákvæmlega.

Sniðugt er að vera búin að láta nemendur prófa vélmenni sem þegar hefur verið sett saman til hvatningar fyrir sig. Það gerir þá áhugasamari og þolinmóðari við að setja saman sitt eigið.

Áður en kennslustundinni lýkur þurfa nemendur að prófa vélmennin sín, eða thuga hvort þeir hafi ekki gert allt rétt. Farið á heimasíðu vélmennisins <http://poppy.local>. Click on *Reboot the robot* button to be sure that the robot software was started with all motors wired in. You can go to the monitor, click on *Monitor and Control* button. If the robot software is correctly started, you should see the green connection logo, otherwise it will be red. If the connection logo is red, you can see what is wrong by looking at messages in *What happened?* page. Most of the time, it's because a wire is unplugged or because you forgot to configure a motor.



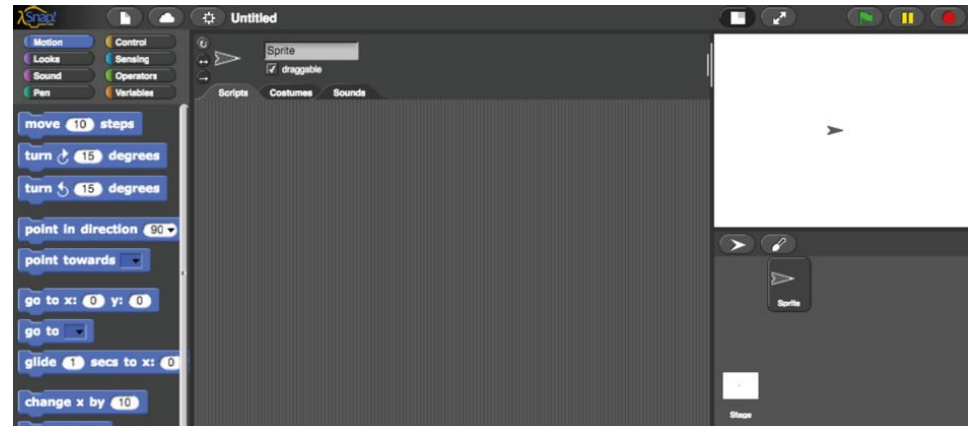
### **Önnur kennslustund: Connecting Poppy Ergo Jr to Snap!**

Snap! is a blocks-based graphical programming language that allows users to create interactive animations, games, and more, while learning about mathematical and computational ideas. Snap! is open-source and it is entirely written in javascript, you can use it from the [official website](#) but you can also use a [copy of the website](#) in your personal computer and open the snap.html file in your browser. The students will need to know the basic functions of Snap! to be able to programme the Poppy robot. This lesson has this purpose.

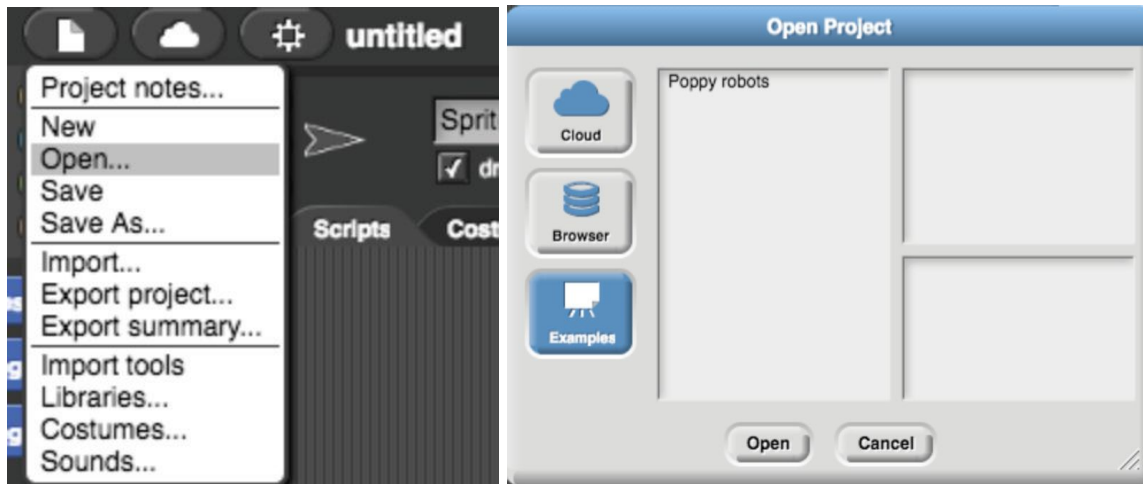
First students have to choose Snap! from the main menu.

The Snap! is divided into three parts: To program Poppy Ergo Jr, you will mainly use the zones left and center.

- Left: the list of available blocks (instructions), arranged in different categories.
- In the center: the script area, where we assemble the blocks.
- Right: above a display area that can be controlled by scripts, below programmable objects that can evolve in the display area.



Open the Poppy robots project to be able to access the specific blocks for the Poppy robot: to do this click on: File> Open> Examples> Poppy robots> Open

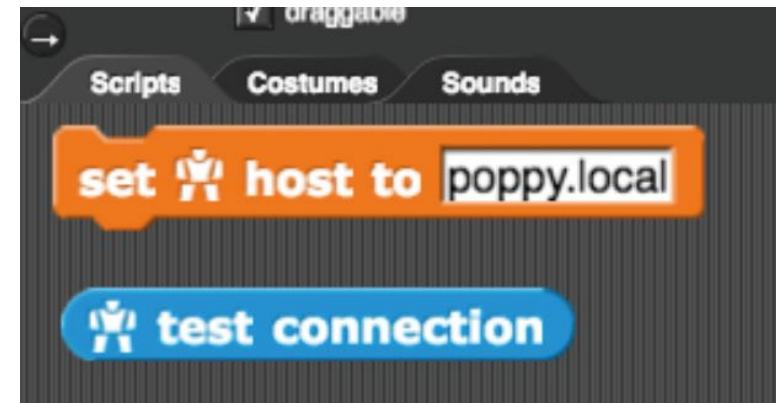


Tip: For information on a Snap! right-click on the block then select help.

You should see two blocks appear in the script area.

Check that you are connected to the robot:

- Write the name of the robot or the IP address in the block (Here the name of the robot is poppy)
- Click on the block TEST CONNECTION to verify that you are well connected to the robot:
- If an error message appears, check that you are connected correctly (see the robot documentation, if necessary).





Save your program: you can save it on your computer's hard drive by clicking on: Files> Export> right click> Save as or save it to the cloud, which requires having a user account:

- Click on the cloud menu in the toolbar
- Select the sign up option from the menu, and follow the instructions
- Check your email to get your initial password

Before activating Ergo Jr, check the correct position of the robot each time. You are ready to start the activities!

### **Priðja kennslustund: How to draw with Poppy Ergo Jr**

In this lesson the students will learn how to make the robot draw using the Snap! - a language of programming with which we assemble instruction blocks. An assembly of blocks is called a script.

To find blocks in Snap ! you can search for:

- By color / category (each one color category)
- By keywords (> Right click on the left side> find blocks) and enter the keyword robot in order to select only the specific blocks for Ergo Jr robot



Create the two scripts below in order to be able to put Ergo Jr in specific positions:



In order to do this:

1. Select and drop the three blocks which you need on the central workspace:



2. Click on the lists and choose the proper values:

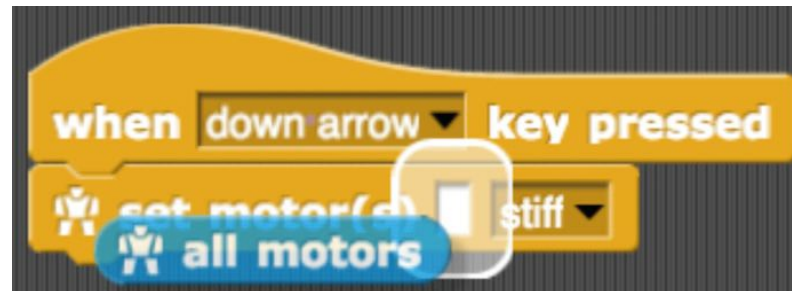


3. Assemble the blocks together:



4. Do the same for the second script.

To nest blocks: select the block and drop it in the desired location with the mouse. The white border indicates that the blocks are going to fit together.



You can copy / paste the blocks and scripts: Right click then duplicate



Activate both scripts (a white border appears around a script activated): Press  $\downarrow$  or  $\uparrow$  on your keyboard and manipulate the robot to put it in different positions.

Get Ergo Jr moving with his motors

We use the following block to move Ergo Jr, motor by motor:



Before you can get moving with Snap! make sure that all the motors are activated (stiff mode) and put all the motors in the basic position (which corresponds to the position where each motor is at 0 degrees: aligned with the notch) in clicking on the following block to execute it:






Put the motor m1 in the 90 degree position in 2 seconds.



Look for the blocks opposite and execute them in turn:



The blocks have different shapes, each shape corresponds to a specific category:

- The oval shaped blocks (like ) are called reporters: when executed, they return a value.
- At the top of the script can be a Hat block, which indicates when the scenario should be executed. The names of Hat blocks usually start with the word When (example: ) ; a script does not necessarily have a block Hat, but without this block, the script will be executed only if the user clicks on the script itself.
- Command blocks (like ) correspond to one action.

Modify the set position block to set the motor m1 and the motor m6 in the -30 degree position in 2 seconds.

With the help of the blocks we have just discovered, build two programs corresponding to the instructions below:

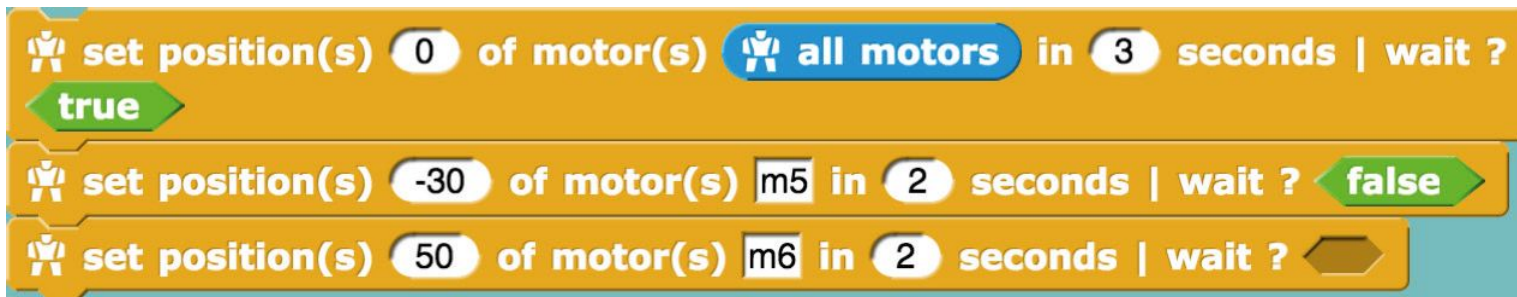
- Press ⇨ on your keyboard, then put all the motors in 0 degree position in 3 seconds.
- Press ⇩ on your keyboard, then put the m1 motors and m4 in position 60 degrees in 2 seconds.

Now let's use the motors to create movements.

Run the script below and watch what happens:




Replace the second block  with the block 



When there are two (or more) nested blocks, in what order do they work the actions ? What happens when WAIT is set to true? What happens when WAIT is equal to false?

The lines of code execute almost instantaneously; and sometimes even if the position requested in the previous line has not been reached. The WAIT part allows you to wait until the engine has reached the desired position before executing the following command.

With the blocks you know now, play a movement to Ergo Jr meaning “hello” when you press the b key.

- Start with a simple movement then enrich it as you go.
- Choose the engines you want to use for the creation of the movement.
- Make the robot play the chosen movement (in compliant mode) and observe the actions of each engine.
- You can help yourself with the block  to know the position of a target engine, and thus note the value to use it again.
- Program the movement motor by motor and test each time the result of your program.

Do not hesitate to create and test other movements!

### Hæfniviðmið

#### Nemendur hafa öðlast færni í að:

- Work independently, trusting themselves
- Follow written instructions and video tutorials
- Assemble a small robotic arm with provided components
- Use a block-based programming language to make the robot execute movements
- Collaborate with peers in on the workshop tasks

### Sources

This scenario is based on the materials available at <https://www.poppy-project.org/en/robots/poppy-ergo-jr>:

- Apprendre à programmer Ergo Jr en Snap! Livret d'accompagnement du robot Poppy Ergo Jr (Conception et réalisation : Équipe Flowers Inria et Ensta ParisTech et Poppy Project; Design graphique : Antonin+Margaux)
- A day to build and program your robot! (<https://www.poppy-project.org/en/posts/poppy-ergo-jr-workshop-at-cern>)