



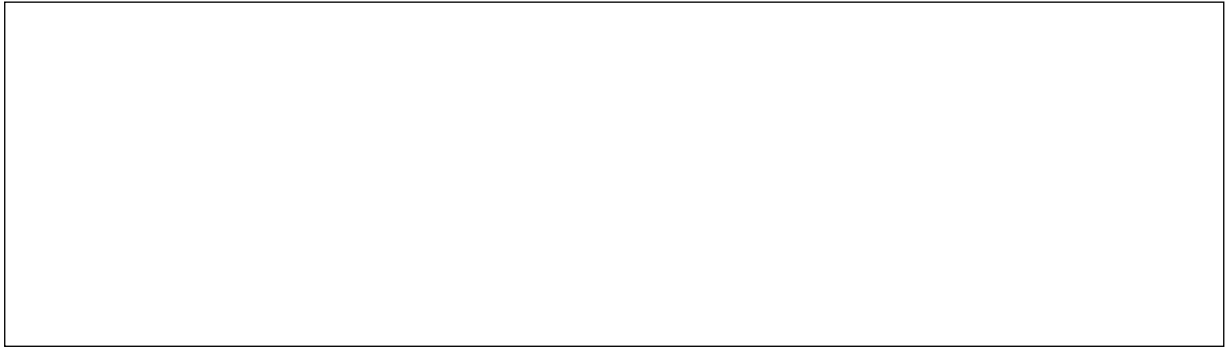
The Sunflower project

Worksheet for students

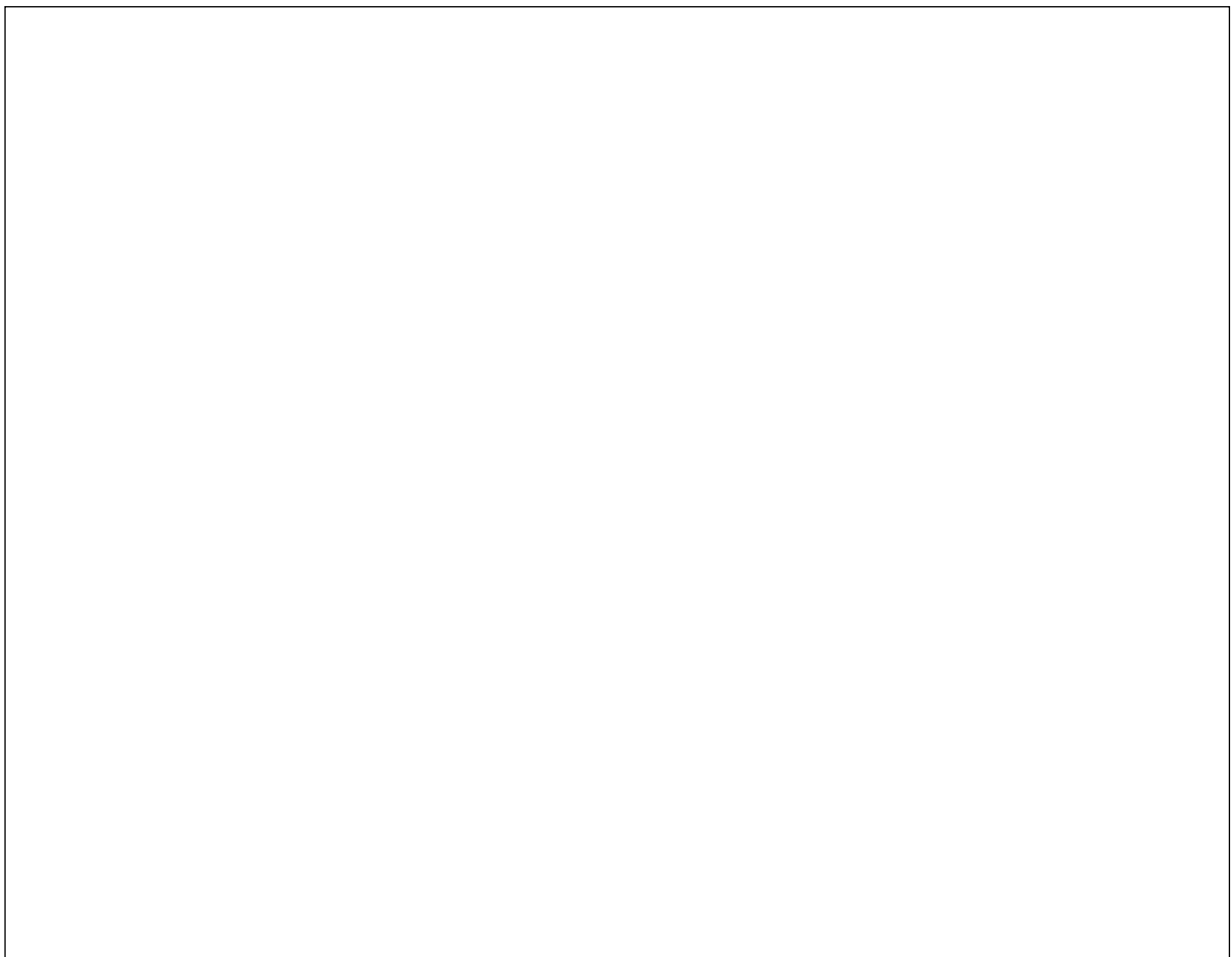
Team:.....

Aim: Simulate the movement of a sunflower towards the light source

Helianthus or sunflower tilts during the day to face the sun. This tracking of the sun is called heliotropism: How do you explain this behaviour? Search for information online and write your answers/thoughts below.



Can you create a flower (i.e. a sunflower) and somehow make it follow the sun? What kind of material will you need? What can be used to play the role of the sun (light source)? Discuss in your team, write your ideas and sketch your design below.



List of crafting material:

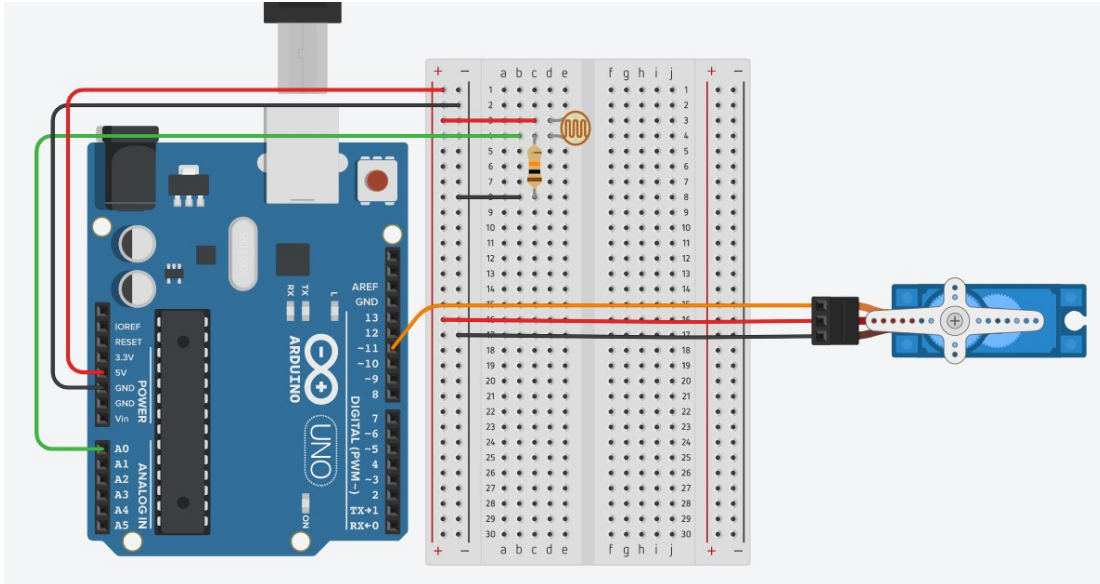
Time for crafting! Start working on the design of the sunflower model using the available (in your class) crafting materials; bear in mind that your design might be revised several times and adjustments/improvements can be introduced.

Time for circuit making!

The following image depicts a circuit in which a servo motor, a photoresistor and a resistor are already connected. You will need to connect one more photoresistor.

Will you need to connect any other component?

Try to draw the circuit below or create the simulation on TinkerCAD.



What does the servo motor do and how does it support the simulation of the phenomenon of phototropism/heliotropism?

Why do we need two photoresistors? Explain briefly below.

Time for hands-on practice!

Let's create the circuit using your Arduino board and the corresponding electrical components.

Time for programming!

Open Snap4Arduino and connect your Arduino to Snap4Arduino.

Now it's time to breath some life to your construction. To do that you need to assemble a script in Snap4Arduino that will simulate the behaviour of the sunflower.

A script in Snap4Arduino (and in most of the block-based programming environments) is assembled by dragging blocks from a palette into the scripting area in the middle part of the window in Snap4Arduino.

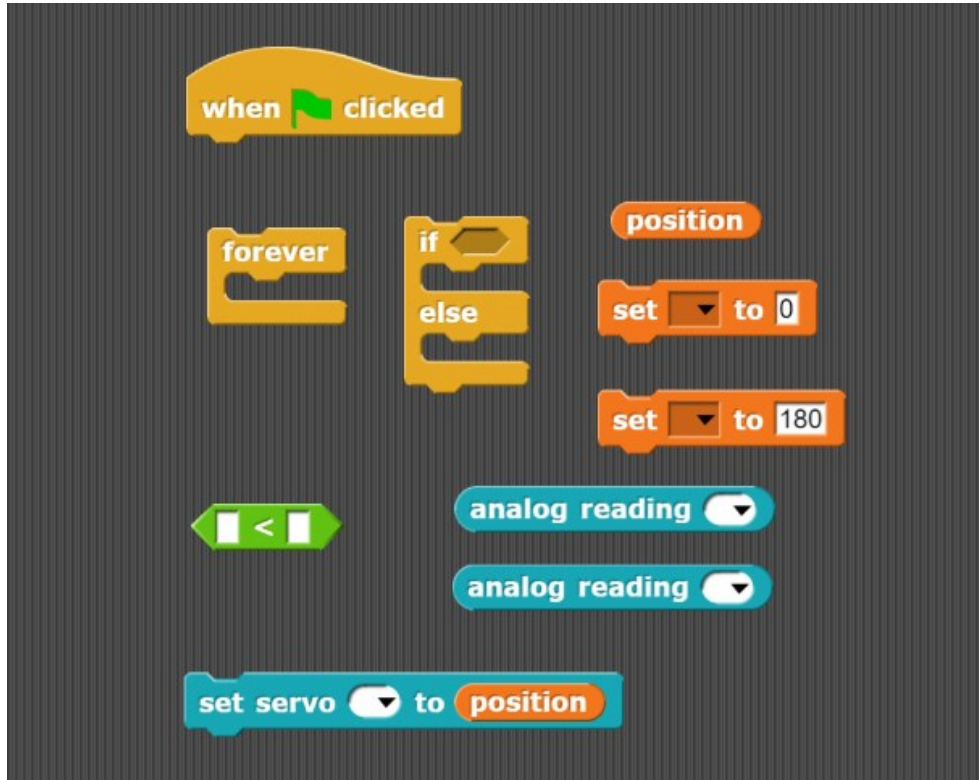
Before assembling the script, it is important to observe the analog readings that are retrieved by the photoresistors. For this purpose, use the following script.



What are the values of the light values that the two light sensors will be reading in the following cases? Complete the table below.

1 st photoresistor	2 nd photoresistor	Light value of the 1 st photoresistor	Light value of the 2 nd photoresistor
covered	Not covered (in physical light)		
covered	covered		
not covered (in physical light)	not covered (in physical light)		
A torch is pointing directly on the 1 st photoresistor	Not covered		

The following script is semi-structured. Place the blocks in the correct order into the scripting area in the middle part of the window in Snap4Arduino in order to make your sunflower move toward the light source.



What will happen if you change the value of the variable 'position' from 0 to 60? Write your answer below

What changes do you think that should be introduced in the script in order to make the sunflower move smoother? Write your ideas below

The following script was introduced by a team of students in order to achieve a smoother transition of the sunflower towards the lighter source. Apply the following script and discuss with your team if they reach their goal.

```
when clicked
  set position to 90
  set servo 11 to position
  wait 2 secs
  forever
    if analog reading 0 < analog reading 1
      change position by -5
    else
      change position by 5
    set servo 11 to position
```

What role do the blocks `change position by -5` and `change position by 5` play?

(Optional) Experiment with additional ideas in order to provide more optimal solutions!

Present the current status of your project in the class.

Tips zone

Programming blocks



This is a *hat* block that indicates that the script should be carried out when the green flag will be clicked.



This is a *C*-block. The slot inside the C shape is a special kind of input slot that accepts a *script* as the input. Any script placed *in there* will be carried out forever in a circle.



The if-else block runs its *if-input* script if (and only if) the expression in its hexagonal input reports true.

If it reports false, the *else-input* is executed.



Click on the Variables Palette and then *here* to make a variable



This block from the *Variables** Palette sets the value of a specific variable (in this case: position) to 180 (subject to change).



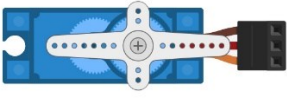


This block from the *Variables** Palette changes the value of a specific variable (in this case: position) by -5 units (subject to change).



This block from the Arduino Palette instructs the *servo** to move to the value indicated by the variable position.

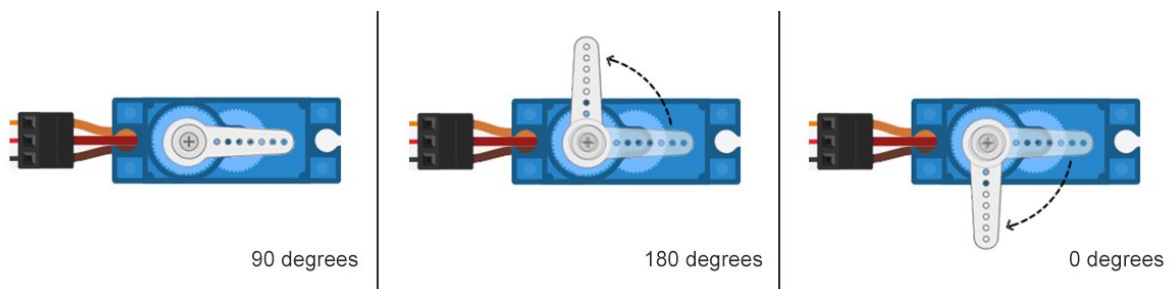
Electrical components

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

	(Angle) Servo motor *
	10 KΩ resistor
	photoresistor

* Servo motor:

A servo motor is a type of geared motor that can only rotate 180 degrees. The following diagram presents the way that servo rotates and the crucial positions that it can occupy, namely 0, 90 and 180 degrees.



ROBOSCIENTISTS PROJECT

Motivating secondary school students towards STEM careers through robotic artefact making

Erasmus+ KA2 2018-1PL01-KA201-051129

Creators

Rene Alimisi, Chrysanthi Papasarantou, Konstantinos Salpasaranis (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

© Copyright 2018 - 2021 the Roboscientists Consortium

All rights reserved.



This document is licensed to the public under a Creative Commons Attribution- NonCommercial-ShareAlike 4.0 International License.

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.