MECHATRONIC FISH

Advanced (with coding)

Stimulate problem solving	**Gold**	**Stimulate entrepreneurship**	**Gold**
Stimulate creativity	**Gold**	**Informal learning enviro.**	**Gold**
Stimulate critical thinking	**Bronze**	**Technology use**	**Gold**
Stimulate group work	**Platinum**		

Practicalities

Preparation: 3-6 hours

Duration: 6 - 8 hours

Material needs:

- Styrofoam 40x600x1200mm
- Styrofoam Cutter (see below)
- 1mm wire
- Jute cloth
- Styrofoam chips
- Spray Paint (10 colors)
- Laptop
- Projector
- Whiteboard marker
- Plywood (4mm - 6mm)
- Arduino Uno (with USB-cord)
- Popsicle sticks
- 2mm and 4mm drills
- Drilling Machine
- Knife
- Styrofoam cutter (DIY link in Content links)

Group size range: 3 - 24
Ideal sub-group size: 3-4

Workshop made for: 16+
Easily transferable to workshops for ages between: 12 - 16

Environment FabLab necessary: No / But some facilities

Educational area:
* Biology
* Computer science
* Mechanics
* (Visual) Arts

Precognition

Fish come in many different shapes and sizes. This workshop is about making a mechatronic fish and adding one (or two) mechanical movements. When deciding what fish they choose to make, the participants also decide what movement is most interesting about that fish. Is it a fish that has a very strong jaw, is it a fish with an unusual tail fin movement or is it a fish with some other interesting property?

The mechanical movements can be accomplished by different mechanisms. The movement for a fin or the jaw can be simplified as angular movement around an axis. This movement is also used for example for steering a RC-car. (see box ‘content links’ below)

Student Preparation:

Before the day of the workshop there are some necessary preparations that need to be done:

Divide the pupils into a number of small groups, approximately 3-4 pupils in each group.

Ask the groups to choose what kind of fish they want to create. Regarding the task itself, it doesn't matter which fish they choose. It's up to you as a teacher if you for some reason want the pupils to work with a specific kind of fish.

Now let the pupils study the fish of their choice based on different aspects: colour, thickness and movement patterns.

The fish the pupils are making will be able to move when done. Therefore the groups must pick two different movements they want their fish to perform, for instance movement of the eye, of the mouth, of one or two fins, or even simple body movements (note that it has to be a one joint movement).

Finally the groups must find a good, high resolution image of their fish. They should then bring the digital image, or a link to the image, on the day of the workshop. The arranging institution might also want you to e-mail the images in advance.

If you so wish, you may want to instruct the pupils to write a short factual text, to be used as a sign next to the completed fish. Of course, if you haven't got the time to do this in advance you let the pupils do it later on.

Workshop Preparation:

Prepare the fish brain (Arduino)

Upload the code from the file fishy.ino to the Arduino board

https://www.karlstadmakers.se/fishy.ino

Prepare the fish muscles (Servo board)

3D-print two servo holders

http://karlstadmakers.se/servoHolder.stl

Make/Test the foam cutter(s)

If you are using a DIY cutter with a power supply: Turn up the effect until the thread turns red. Then turn it down slightly so the thread stops glowing.

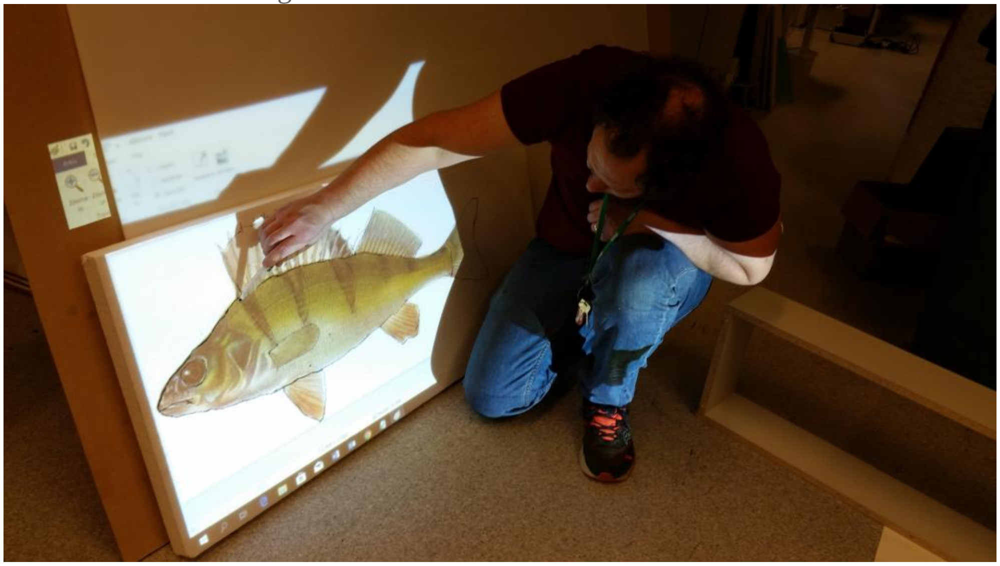
Workshop Guidelines

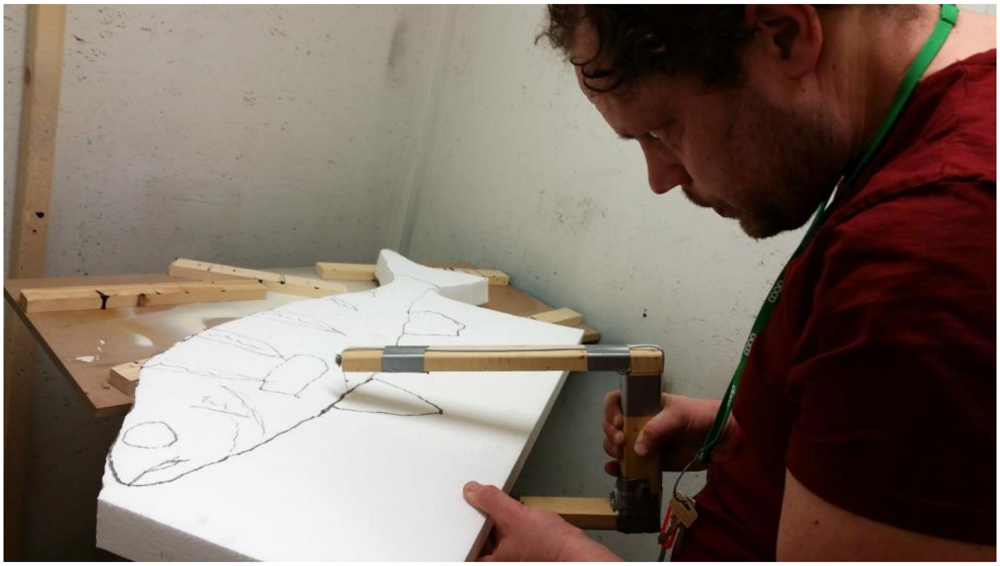
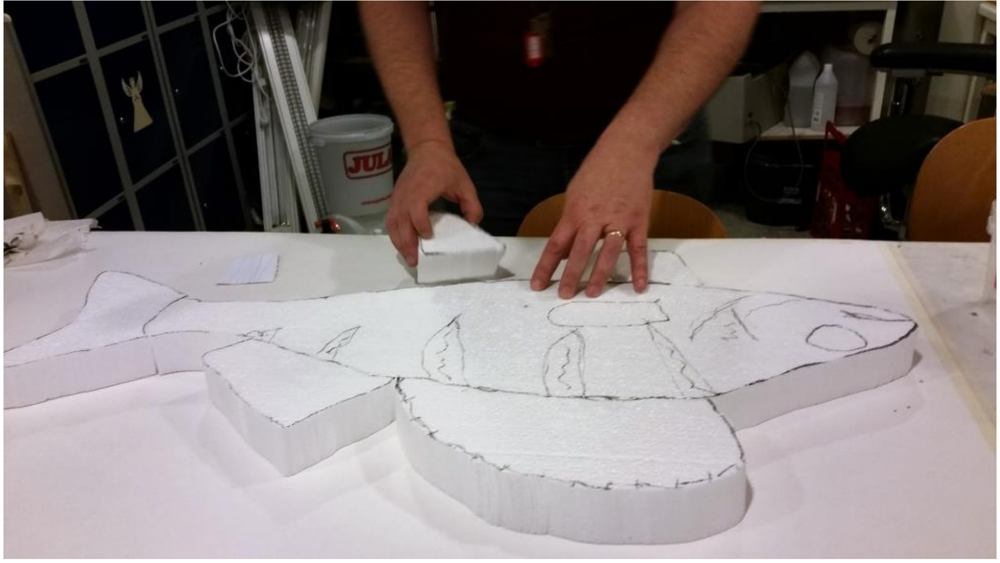
Projecting, drawing and cutting

Material needs:
Essential: *Digital image of fish , computer, projector, styrofoam, styrofoam cutter, whiteboard marker*

Goals:
Skill Goals (**Blue**)
(S1) Projection method of image copying
(S2) Melting point of plastic
Content Goals (**Green**)
(C1) Using projection to paint.
(C2) Using a foam cutter.

Background story:

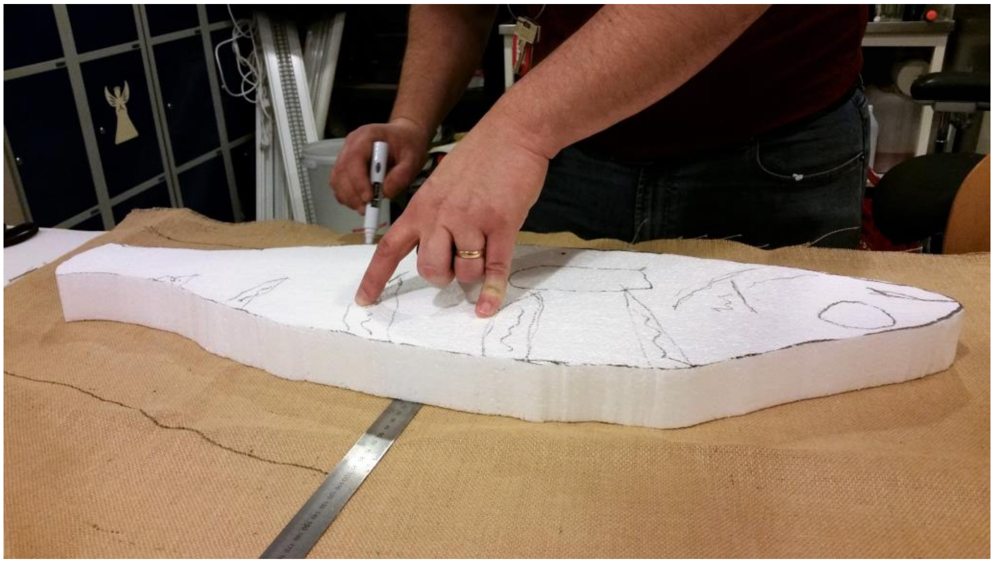
Goals	Activities	Duration
S1	Project the image of the fish onto the styrofoam.	10
S1, C1	Draw the contours using a whiteboard marker 	20
S2, C2	Cut out the fish and the fins with a styrofoam cutter. Separate the fins from the body	20

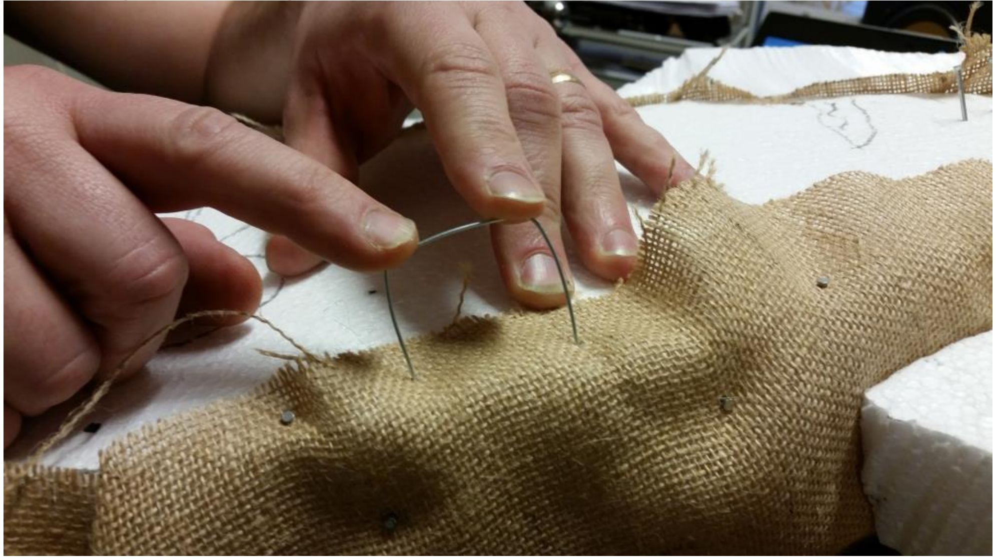
		
	Put the fins away for now 	

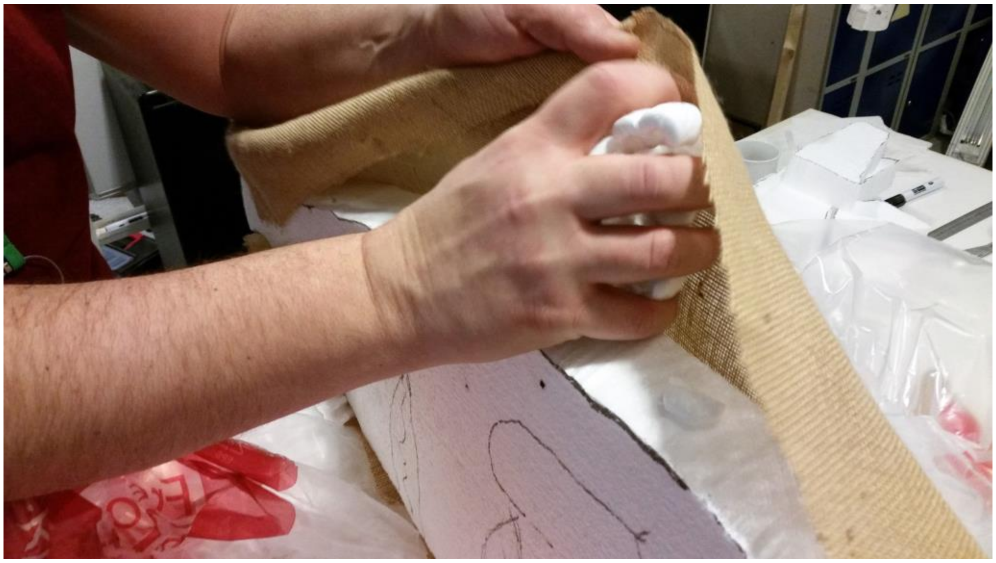
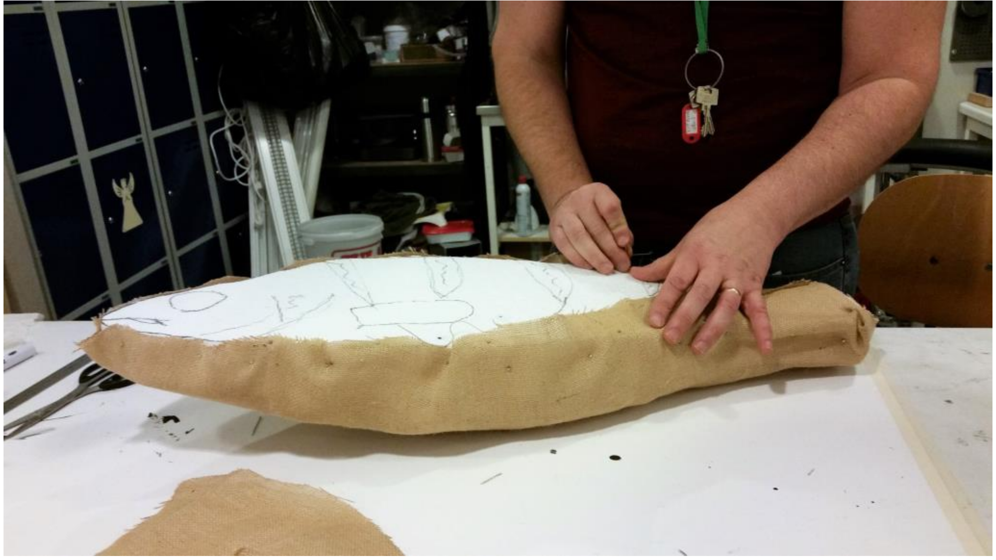
Skin and flesh

Material needs:
Essential: *Fish, jute fabric, whiteboard marker, wire, nippers, foam chips*

Goals:
Skill Goals (**Blue**)
(S1) Drawing and Cutting
(S2) Surface area / volume relation
Content Goals (**Green**)
(C1) Draw the fish
(C2) Fill and mount the fish

Goals	Activities	Duration
	Place the fish, without the fins, on a piece of jute fabric	2
S1, S2, C1	Draw a line on the cloth 10 - 15 cm outside the fish. 	10
S1, S2, C1	Cut the fabric = the skin of the fish	10

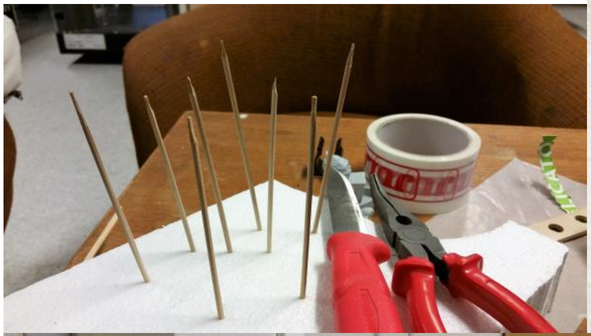
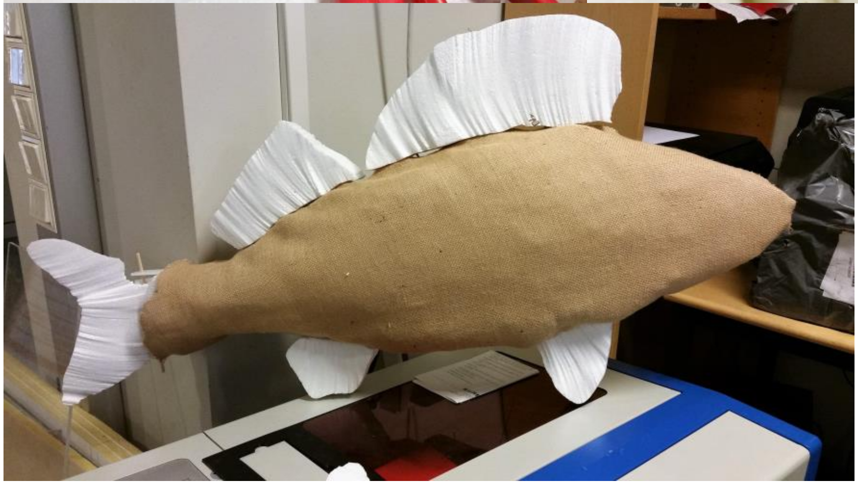
S2, C2	Attach the fabric to the styrofoam using pieces of wire, bent to U end loops. Leave an opening for the filling 	20
S2, C2	Fill the fish to desired thickness by putting styrofoam chips in between the body and the skin	10

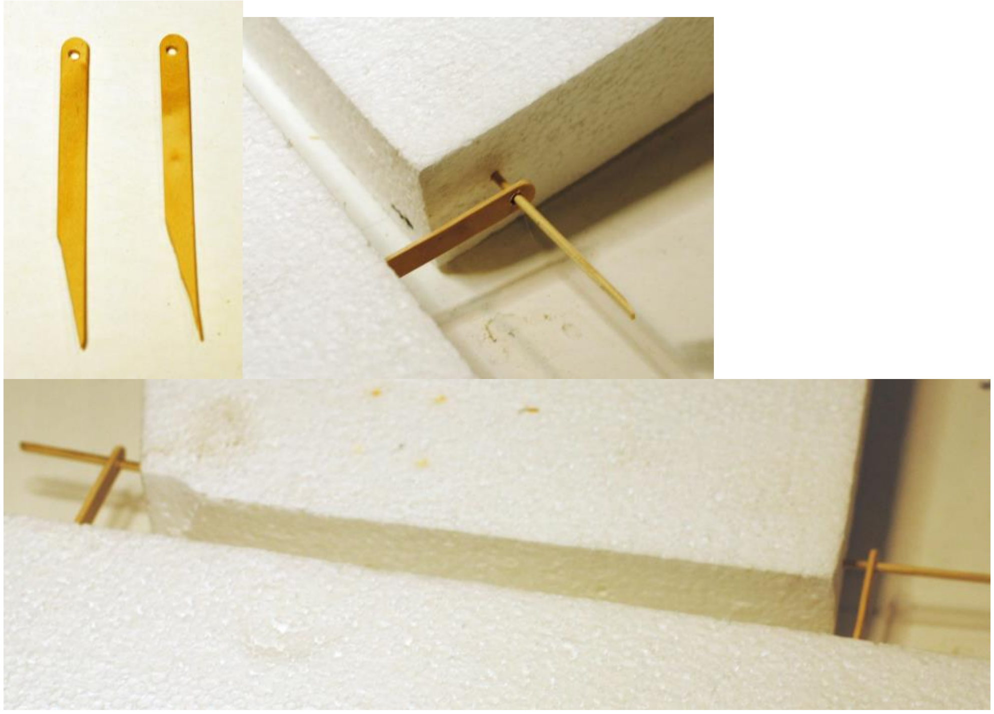
		
C2	Attach the remaining opening with wire loops 	5

Mounting the Fish body

Material needs:
Essential: *fish, fins, skewers, popsicle sticks, 4mm drill*

Goals:
Skill Goals (**Blue**)
(S1) Mechanics Static attachment
(S2) Mechanics Angular movement
Content Goals (**Green**)
(C1) Attach fins
(C2) Make a functional hinge

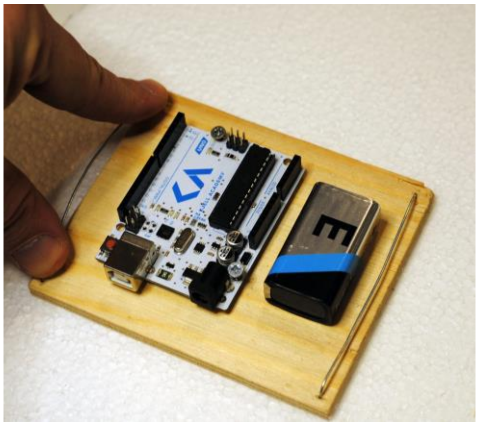
Goals	Activities	Duration
S1, C1	Mount the static fins on the fish body using skewers sharpened i both ends.  	5
S1.S2, C2	Mount the moving parts using hinges. Mage a hinge by: Cutting a popsicle stick in half. Sharpening the strait side of a popsicle stick Drill a hole in the round side of the part.	10

Mounting the brain and muscles

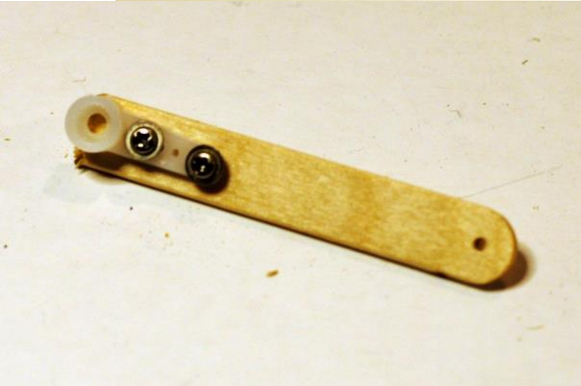
Material needs:
Essential: *Arduino uno, plywood board, 2mm drill, wire U end loops, microservo with arm , servo holders, cables*

Goals:
Skill Goals (**Blue**)
(S1) Micro computer layout
(S2) The functionality of a servo
(S3) Transmission of a movement
Content Goals (**Green**)
(C1) Rigid Mount of the boards
(C2) Build a functional movement

Goals	Activities	Duration
S1, C1	Mount the brain (Arduino board) Download the arduino file Fishy.ino from https:// www.karlstadmakers.se/fishy.ino Upload the file to the arduino (If you don't know how to upload a file to the arduino: google it. There are lots of tutorials online) Make a 120mm x 100mm plywood board (4 or 6 mm plywood) drill 2mm holes in all 4 corners Mount the arduino onto the Brain board with screws. Attach the battery to the brain board using a rubber- band. 	15
S2, C1	Mount the muscles (servo board) Cut a popsicle stick in half. Drill a 2mm hole in the rounded end of the stick.	15

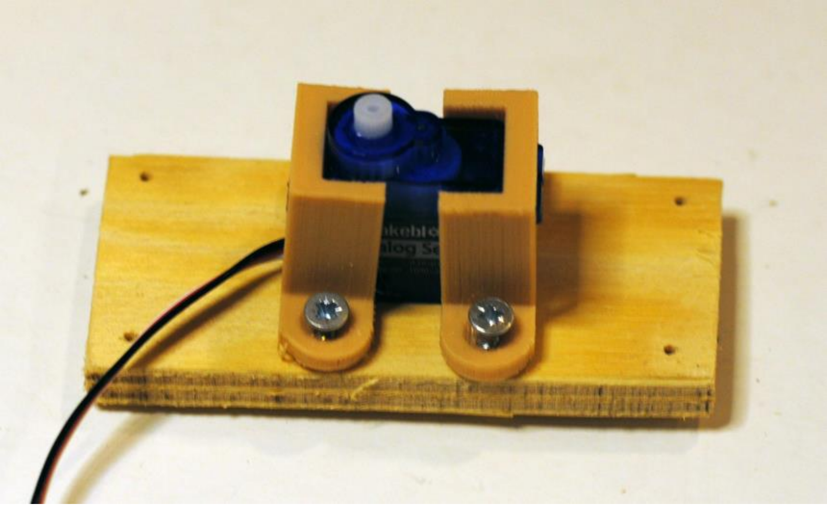
Sharpen t

Glue och o servo arm.

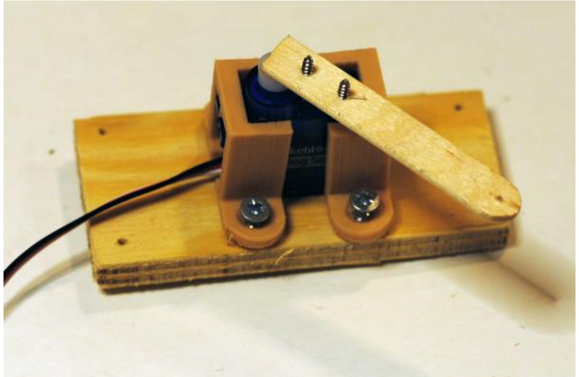
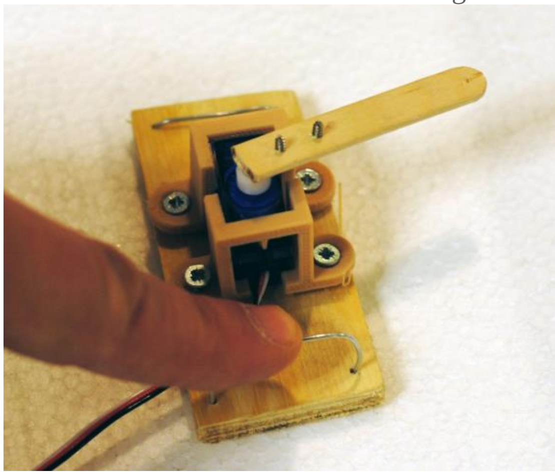
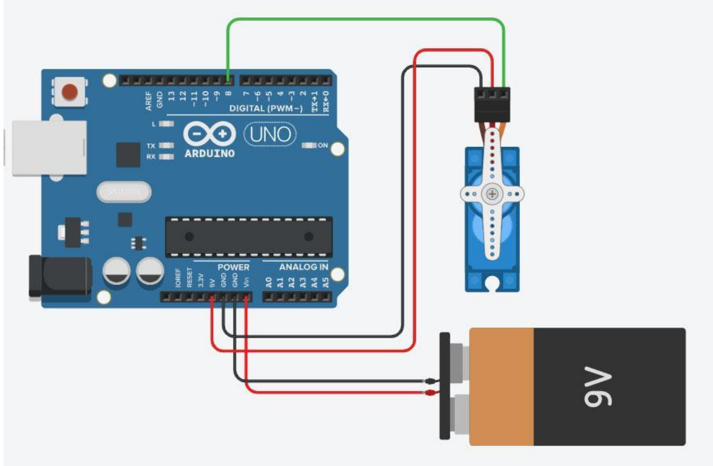


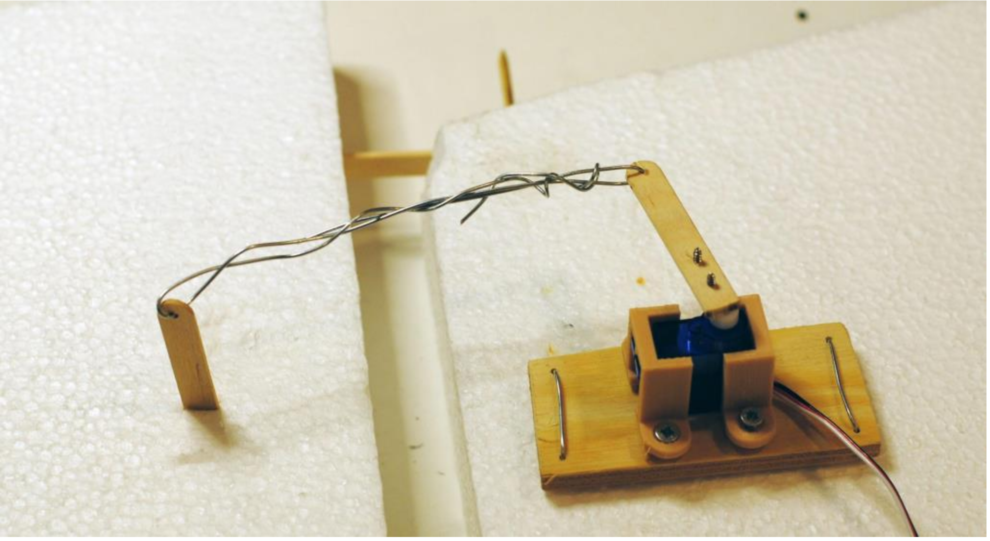
Cut a 40mm x 80mm plywood board.
Drill 2mm holes in the corners of the board.

3D-print/Make/Buy Servo holders and attach the servo to the board.



Attach the modified servo arm with the popsicle stick to the servo.

	 Attach the servo board to the fish using wire U end loops 	
S1,C2	Connect battery and servo to the Arduino using cables.  DIGITAL (PWM~) UNO ARDUINO POWER ANALOG IN 9V	15
S1, S3, C1, C2	Attach the muscles Use the other part of the halved popsicle stick and:	15 - 45

	Drill a 2mm hole in the rounded end of the stick. Sharpen the strait end of the stick. Press the sharpened servo arm into the moving part (fin) Make a linkage arm with twisted wire and put it in the holes of the servo arms. Think about the length of the servo arm. You might need to adjust it later. 	
S2, C2	Customize the movement The fishy.ino sketch allows you to connect a servo to three different pins with preprogrammed movements pin 8 = slow pin 10 = fast pin 12 = fast and wait	

Programming the fish

Material needs:
Essential: Computer with Arduino software installed, USB- cable, LED

Goals:
Skill Goals (**Blue**)
(S1) Basic syntax of C
Content Goals (**Green**)
(C1) Make a custom movement

Goals	Activities	Duration
S1	Read “Fishy code” below.	60 - 90
S1, C1	Tweak the timing in the code to make the fish move the way you want	15 - 30
S1, C1	Make your own code Upload your own code to the arduino	30 - 120

FISHY CODE

To program the movements of your fish you will need some basic programming skills.

After you have made a program you can upload the program to the Arduino to test it.

If you want to make a program without the real Arduino you can use the free Arduino/electronics simulator Tinkercad. (www.tinkercad.com)

The programming language for Arduino is a version of the language C.
In C there are a few very important syntax that you will have to understand.

CHARACTERS

The curly brackets { and }

These brackets make up the structure of the code. What stands between them makes up a block of code. Blocks of code contain other blocks of code and it is important to keep track of your curly brackets. One bracket in the wrong place and the code won’t function as intended.

The parenthesis (and)

The parentheses are used to define parameters of function.

The parentheses are also used to define a statement that can be true or false in a if...then-statement

The semicolon ;

Semicolon is used to close a line of code.

CAPITAL and lower-case letters

are very important. Be sure to use the right kind.

Double slash //

The text after the double slash is not used as code, it's a comment.

BASIC COMMANDS

Pinmode

sets one of the pins to an output or a input pin

digitalWrite

sets the value of an output pin

for loop

repeats a block of code over and over while increasing or decreasing a variable

LIBRARIES

Alot of programming can be simplified by using libraries. When a library is used you get access to new commands.

BASIC STRUCTURE

All arduino programs are made up of two main blocks of code The setup and the loop.

Variables that are used in the whole program are declared before the setup

The setup is performed every time the arduino is turned on of when the reset button is pressed.

The loop is performed after the setup and as the name suggests it is performed over and over.

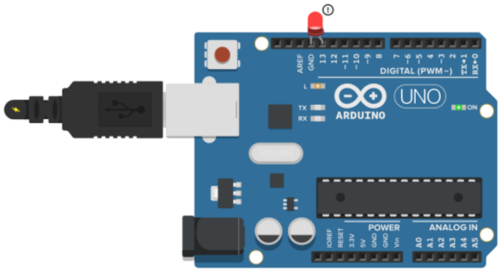
BLINKING LED

Let’s start with a very simple program to test how to use the Arduino

Connect a LED on pin 13 and GND. A LED will only allow current to flow in one direction so be sure to put the longer leg (anod) in pin 13 and the shorter leg (cathode) in GND.

(connecting a LED without a resistor will shorten the lifetime of the LED, but that is not important for now)

```
void setup()

{

pinMode(13, OUTPUT); // sets 13 to be an output pin

}

void loop()

{

digitalWrite(13, HIGH);

delay(1000);

digitalWrite(13, LOW);

delay(1000);

}
```



void setup() defines what should be executed every time the Arduino starts up

The code between the two curly brackets is the sepup block

There is only one line of code in the setup block:

pinMode(13, OUTPUT);

This sets pin 13 to an output pin and enables sending a current from pin 13

void loop() defines the loop, which is executed after the setup and repeats over and over.

The code between the two curly brackets is the loop block

digitalWrite(13, HIGH): //turns on pin 13

```
delay(1000); //waits for 1000 milliseconds (one second)

digitalWrite(13, LOW): //turns off pin 13

delay(1000); //waits for 1000 milliseconds (one second)
```

SERVOS

Connect the Ground(GND), Power(5V) and Signal cables from the servo to GND, 5V and pin 3 on the arduino.

Servos are controlled by Pulse width modulation (PWM). Fortunately we don't have to get so theoretical about that. Arduino pins with tilde sign (~) support PWM.

We can just import a library to enable us to use some new commands.

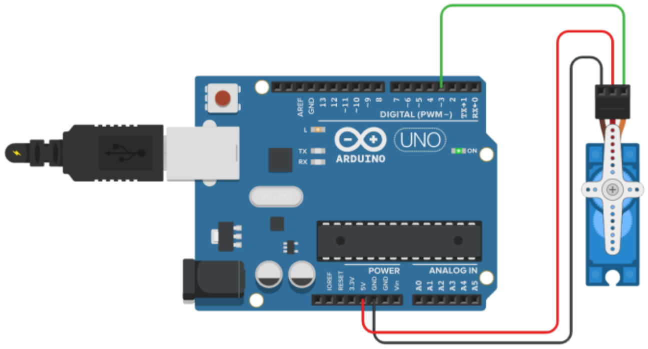
Put #include <Servo.h> at the beginning of your code to enable servo commands

Define a servo by putting Servo name_of_your_servo at the top of your code

Use name_of_your_servo.attach(your_servo_pin) in the setup() to enable the servo.

Name_of_your_servo.write sets the servo to a position (in degrees)

```
#include <Servo.h>
Servo servo1;

void setup()
{
servo1.attach(3);
}

void loop()
{
servo1.write(120);
delay(1000);
servo1.write(10);
delay(1000);
}
```



This code turns the servo to position 120 degrees, waits a second, turns it to position 10 degrees, waits a second and loops

VARIABLES AND LOOPS

Often it is convenient (or necessary) to use a variable instead of a constant number.

If want the servo to move in small steps from 0 to 180 degrees we could write

```
servo1.write(0);
delay(50);
servo1.write(1);
delay(50);
servo1.write(2);
delay(50);

...

servo1.write(179);
delay(50);
servo1.write(180);
servo1.write(0);
delay(1000);
```

This is not a good solution. Besides taking a lot of time it will consume a lot of the arduinos limited memory.

To do it more efficiently we can use a for-loop

```
#include <Servo.h>
Servo servo1;

void setup()
{
  servo1.attach(3);
}

void loop()
{

  for (int i = 0; i <= 180; i=i+1) {
    servo1.write(i);
    delay(10);
}

  servo1.write(0);
  delay(1000);

}
```

This will define a variable I and initially set it to zero.

The code between the curly brackets will be performed over and over while I is 180 or less.

i is in increased by 1 every time the loop is performed

After the servo has moved to 180 degrees it moves back to 0 degrees and stops for one second and then starts over since the code is in the loop() function

You should be able to make a lot of different movements just with these commands.

To be able to use sensors, buttons or other interactive input, just google it + arduino

Here are some keywords:

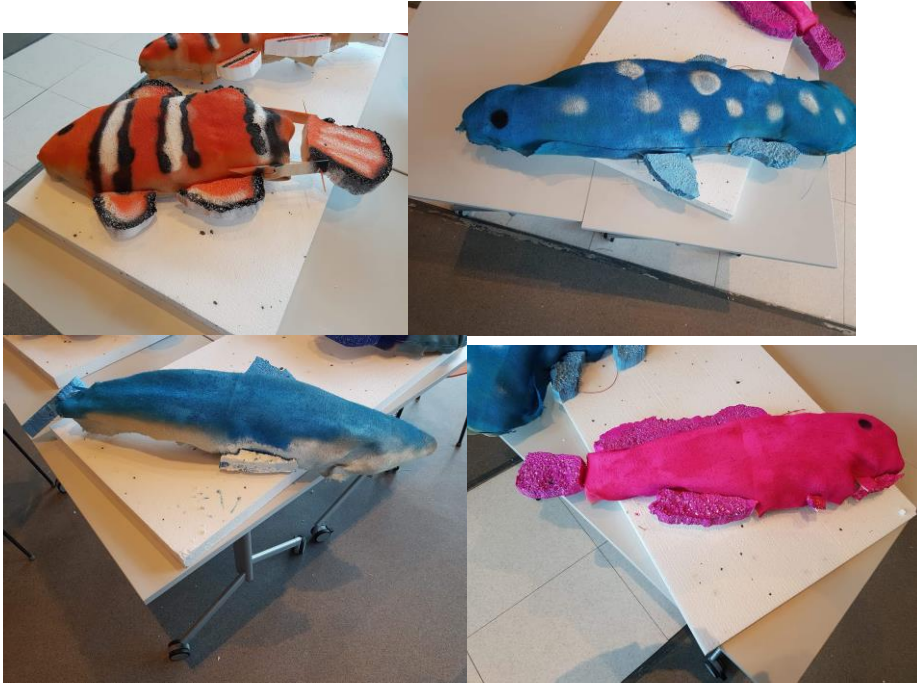
arduino button

arduino motion detector

arduino light sensor

Painting the fish

Material needs:
Essential: Fish, fins, Spray paint
Optional: Acrylic paint, brushes
Goals:
Skill Goals (**Blue**)
(S1) Copy a pattern onto a 3d-object
Content Goals (**Green**)
(C1) Have a 3d-copy of your 2d-image

Goals	Activities	Duration
S1, C1	Paint your fish, if you use paint that smells strong maybe you need to Consider good ventilation. Use an image of your fish to guide you. 	30

Pedagogical tips

Let the participants make mistakes. The equipment is cheap and can easily be modified if things don't work at first.

The exception from this is connecting the power to the arduino. Check that everything is connected properly before connecting both poles to the battery.

How to transfer to non-Fablab environment

Some of the processes require an extractor fan but can also be performed outside.

Do not cut styrofoam with a heated thread or use spray paint without proper ventilation.

Evaluation of achievements

The work with the fish is very much about the process. Participant groups can run into different problems depending on what movement they want to apply.

Try to listen and ask questions during the process to make a good evaluation of their achievements.

Content links

About servos: https://learn.sparkfun.com/tutorials/hobby-servo-tutorial/all
About fish: https://tpwd.texas.gov/kids/wild_things/fish/
Mechanical movements: http://507movements.com/
Arduino: https://www.arduino.cc/en/Tutorial
Software for Arduino IDE: https://www.arduino.cc/en/Main/Software
Foam cutter: https://www.norwegiancreations.com/2017/06/building-a-quick-dirty-foam-cutter/