Interdisciplinary course of

Design and Robotics

6° edition, 2018

Professors:
Andrea Bonarini, DEIB department, Politecnico di Milano
Maximiliano Romero , DPAC department, IUAV, Venice University

Tutors:
Rohan Vangal, Goutham Gopal Ksheerasagar

Students:
Matteo Sinico, School of Engineering
Martin Silvio Di Pietro, School of Design
Tommaso Da Col, School of Engineering
Ioannis Koutoulogenis, School of Engineering
Ruilin Zhang, School of Engineering
Martina Kiselichka, School of Design

Index

Abstract

This report is the detailed guidance for the robotic and design course project of TEAM II. The report contains the detailed description of the research field, development process and interaction guidance regarding the robot. Our goal in this project was to make an interactive robot which has the ability to demonstrate five different emotions through the interaction with its surrounding. The five different emotions which are joy, fear, disgust, sadness and anger they will be displayed through the movement of the robot itself and various sound and light effect.

As a team which composed of six people with different cultural and academic background a various focus and interests, we as a group decided to divide the work into different sections. Each different section of the report is the detailed step of how our group managed to combine the design, mechanics, informatics and electronics all together in this robot. The reader can also go to the specific section page through the page number indicated on the index page.

In the first three sections of this report which are research/inspiration resources, concept and development process, the reader can have a brief idea of where we got our inspiration from and how we decided to design the robot based on every team member's background knowledge. In the development section, we have seperated the process into interaction, shape, mechanics and informatics. In the design and making of the robot, we have stuck to the rule of simplicity.

Throughout the process of the development of robot, our group has also encountered some difficulties, and we managed to solve them all. Regardless of the complexity of design, and how to combine electricity, mechanism and design all inside the limited space of robot, we also had a hard time with the material choosing and situation handling. In the conclusion section, you will see the detailed process and reflection that our team has learnt from the building of robot.

It is hoped that this report will inform the readers for every step of our design process, and how our team managed to design this interactive robot from scratch to prototyping and eventually to the final result.

Description

The project is about an interactive robot which can express five different emotions, according to the interaction with the user. The robot provides rich expressive capability which are joy, disgust, fear, anger and sadness.

This robot is based on the shape of a tetrahedron which consist of four surfaces. We have decided on following the simplicity rule to create a symmetric polyhedron based robot. The surface on the bottom is the robot's basis. The two surfaces on the side will be the "wings" of the robot, and the surface at the front will be its front face.

There are three wheels placed at the bottom of the robot and the movement of the wheels is according to the emotion of the robot. The five emotions are also represented by the movement of the wings (up and down) and the robot changes color and sounds according to the current emotion it shows. Details about the interaction for the emotions can be found at the interaction section. The user manual is attached at the last page of the report.

Throughout the whole building process, we have also encountered several challenges regarding the material choosing, the mechanics building and how to create a vivid movement with the aesthetic perception. Every week, we are improving the appearance and mechanic of the robot based on the feedback of the professor. From this report, you can go through our decision making and the whole building process of this interactive robot.

Research

State of the Art

Our goal is to create an interactive robot with delicate appearance. We got inspiration from the following projects which are Luxo, Lou, Cozmo and yeti.
Keywords: Robot, Pixar, Luxo, Lou, Cozmo, Yeti

Luxo

Luxo Jr. is a 1986 American computer-animated short film produced by Pixar and written and directed by John Lasseter. The short animation film contains all the principle of animation such as squash and stretch, pose to pose and staging. The narrative is simple and it is mainly about how the two lamps interact with the ball in a playful way. The interaction between the lamp and the rubber ball is interesting. The robot (lamp) is able to express several emotions including happiness, sadness, surprisingness and madness. The short film involved a large amount of animation principle and it was a perfect example for our team to consider the interaction between the robot and user.

Lou

Lou is a creature made from various unclaimed items. The box for lost items on the playground is the home of Lou. Every day Lou raises from the toys that children leave; the next day he puts them on the site for their owners, so that they can find them.

A hooligan named JJ begins to take away other people's toys, keeping them in his backpack. Lou starts to steal toys from his backpack, but JJ catches him and chases him.
During the chase, Lou discovers on his underwear the name of the hooligan JJ and remembers that he saw this boy in his childhood and how he was deprived of his toy - a teddy bear. This bear was in the box of Lou. JJ saw his bear and tries to take it, but Lou refuses to give it away.

JJ returns all the selected toys, which he took back from all of the kids. In the end, JJ feels joy in reuniting children with his toys and emotionally returns all the toys out of the box, even those that make up Lou. Returning to the box, JJ sees that the last toy was his teddy bear and deservedly takes it.

Cozmo

Cozmo, is an artificial intelligence toy robot with a big brain and even bigger personality. Cozmo is always ready to play and learns more as you hang out.

Yeti

Yeti (Your Emotion Through Interaction), as the name said, is a robot that interact with disabled child in a game based on emotion. The robot could actually express emotion through the 2D movement, the changing on his shape and the colour that it could assume. The game is based on color recognition, a device change his colour and has to be positioned by the player on a coloured base placed in the game space. If the task is doing properly by the player the robot became happy, otherwise, if for example the colour is wrong or nobody plays it become sad. Actually the things we liked about it is the very simple way to express emotion through those features and also the thing that is not humanoid or an animal, even if it maintains some human features like a head with eyes and mouth. We decided to go over it by realizing complete inanimate object that could express however the emotion.

Concept

Inspiration

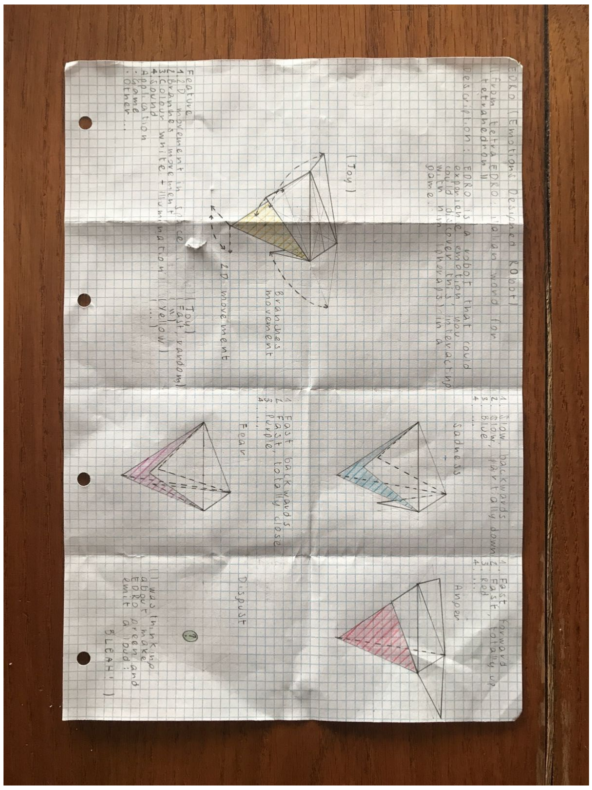
Inspired by the idea of the robots Luxo, Lou, Cozmo and Yeti we have came to the idea of creating robot that has the similar functions, showing emotions as a happiness, sadness, anger, disgust and fear called EDRO. EDRO is a special robot that has very simple shape with which we show a total diversity of human emotions.

*simple shape /minimal aesthetic design/ use its movements to show human expression

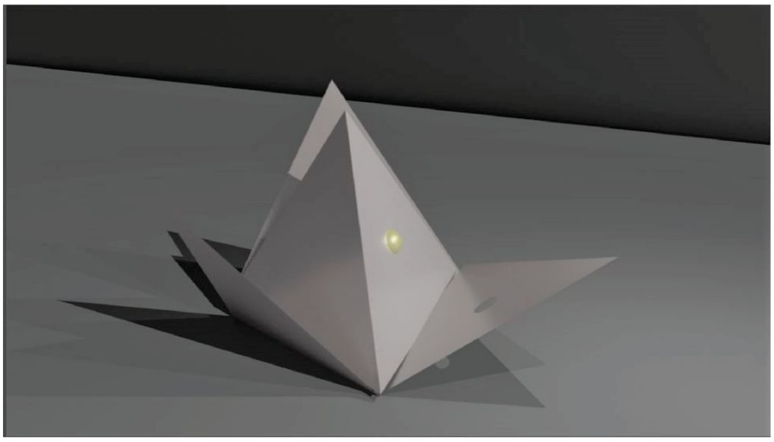
We are aware of the weight of our task that we have chosen to show emotions from such a simple shape.

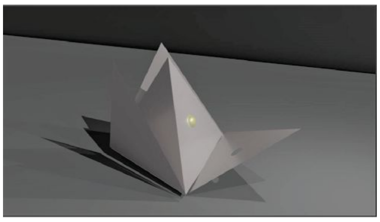
The five emotions are represented by the movement of the wings (up and down) and by the movement of the robot itself with the help of the wheels. The robot changes color in terms of the emotion it shows.

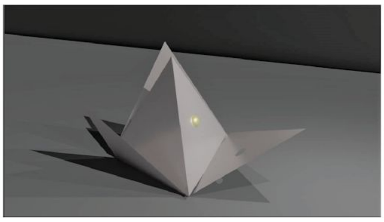
*Sketch from the first draft

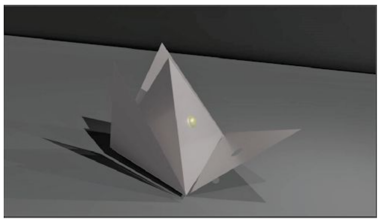


1. Joy: Fast indipendent movements of the wings and moderate shaking of the body.
   Color - yellow or multi color

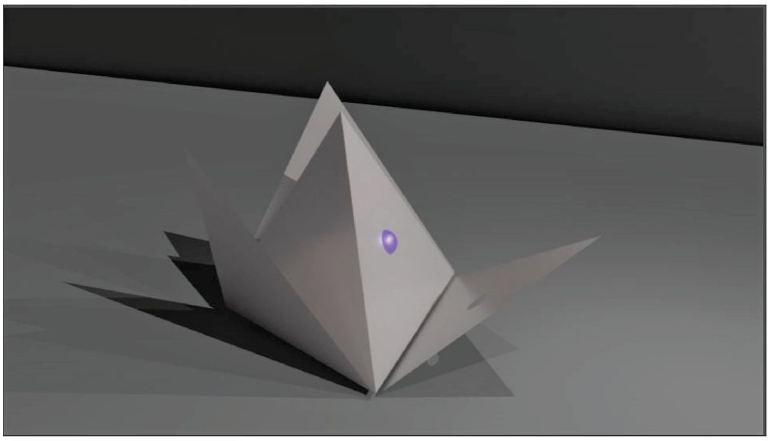


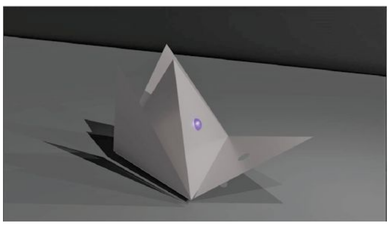


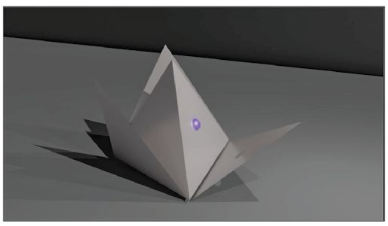


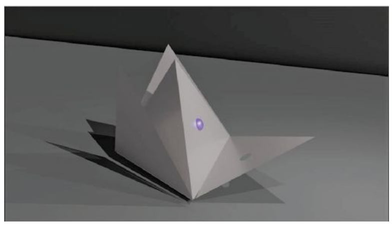


2. Fear: wings close fast and the body is shaking rapidly, like it is vibrating.
   Color - violet

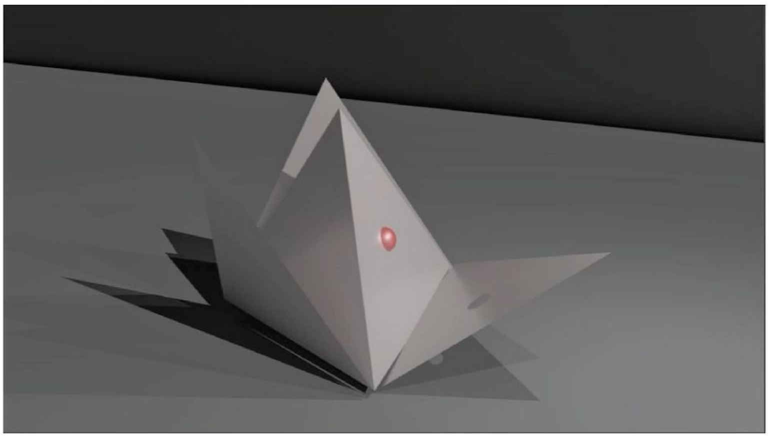


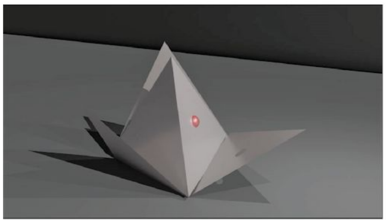
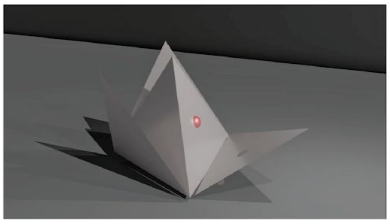
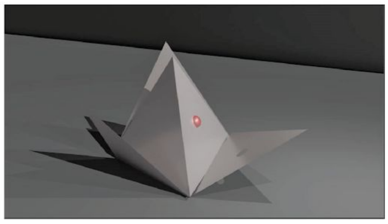




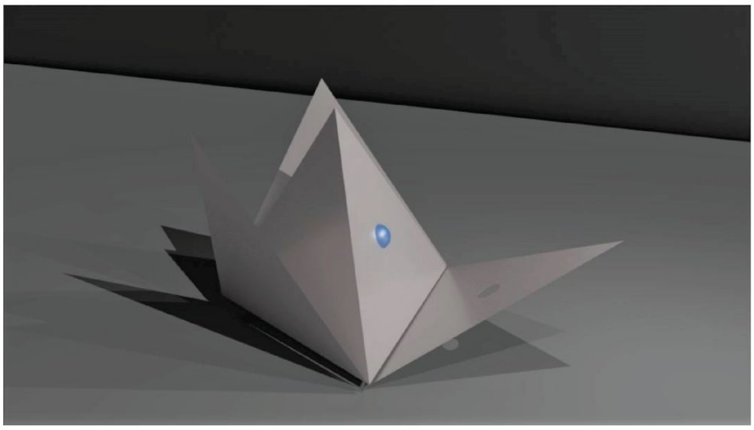


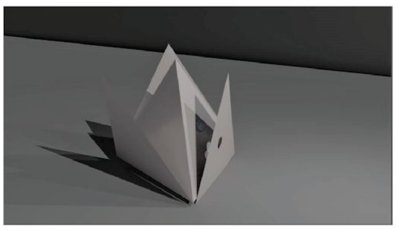
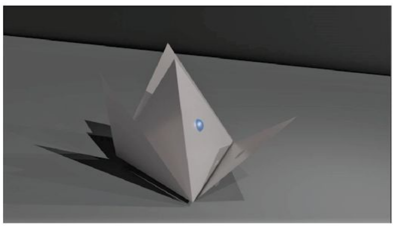
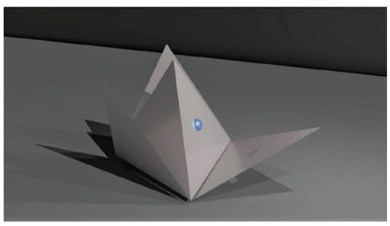
3. Anger: Lateral wings closed and the front one open in a rigid way and the body is moving fast straight in the front (like attacking).
   Color - red.



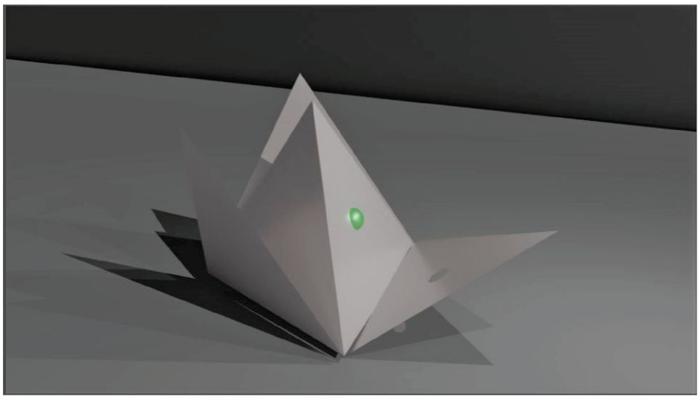




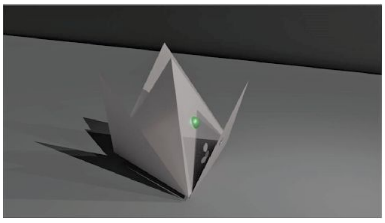
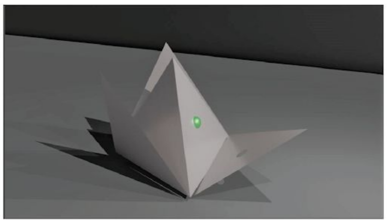
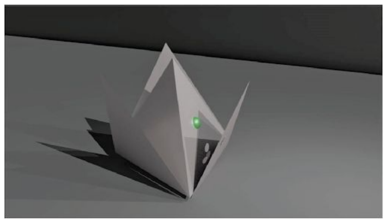
4. Sadness: Wings open in a slow way and go down, going backward slowly.
   Color - blue (blinking smoothly).



5. Disgust: only the front wings open down and the other two wings are closing , going backward and shaking the body, like it’s shaking its head.
   Color - green.



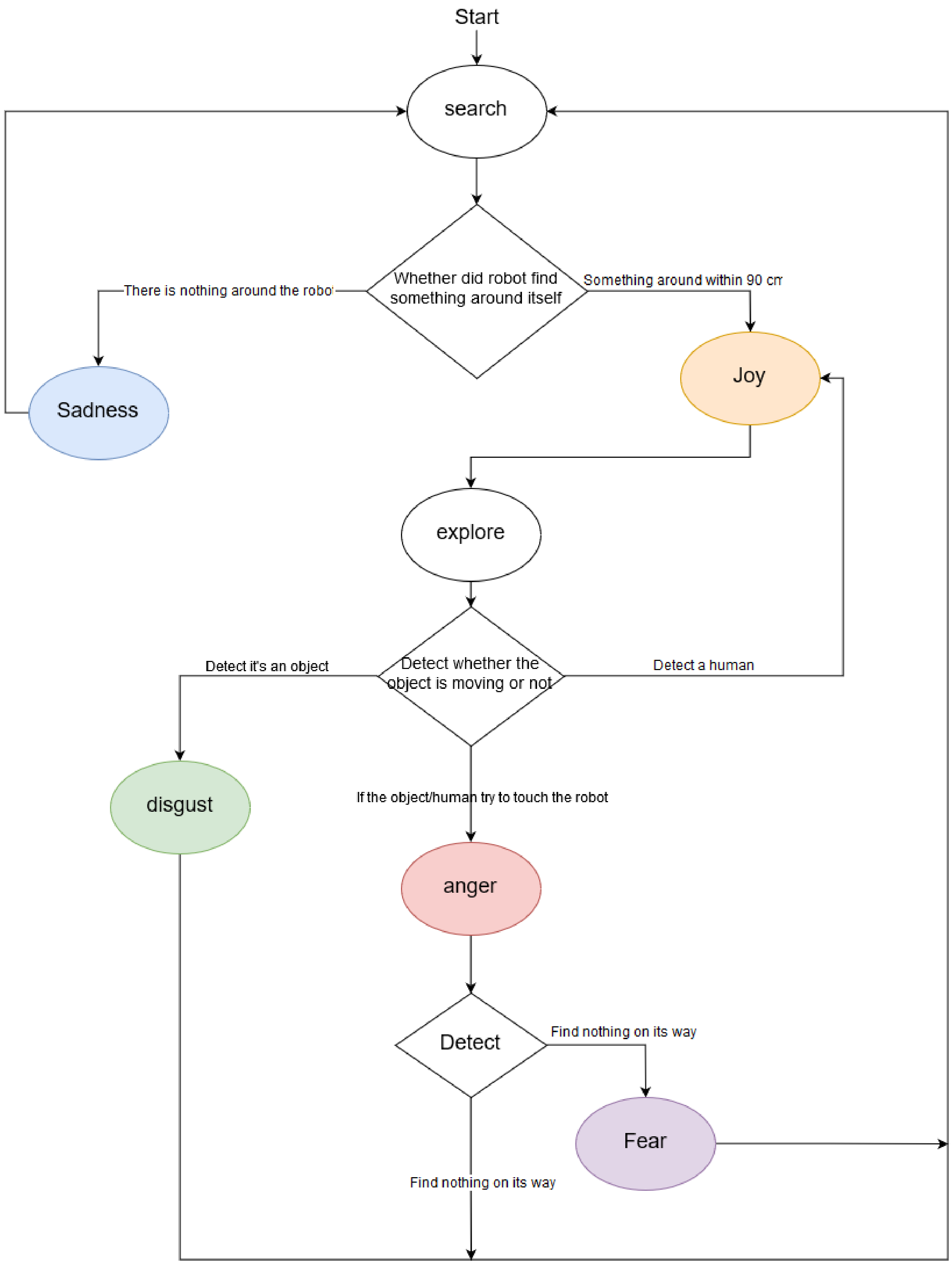
Development

Interaction

<u>First approach:</u>
Sensor: distance sensor

The starting state of the robot is searching. If the robot’s distance sensor detects any object within 90 cm, the robot will express joy. Otherwise, the robot will express sadness and come back to the first state which is searching again. After the robot express joy emotion, it will start to detect the movement of the object that it found, if the distance between the object and robot itself is changing(decreasing), robot will detect the object as human. Otherwise, the object will be detected as a still object.

In the third state, if the object/human try to touch the robot, the robot itself will express anger. And if the distance between the robot and human are still decreasing, the robot will express fear and then come back to the first state which is searching again. The detailed interaction flowchart is shown below.


Start
search
Whether did robot find something around itself
There is nothing around the robot
Something around within 90 cm
Sadness
Joy
explore
Detect whether the object is moving or not
Detect it's an object
Detect a human
disgust
If the object/human try to touch the robot
anger
Detect
Find nothing on its way
Fear
Find nothing on its way

Joy:
When the robot detected something within 90 cm distance.
Sound: joy sound

Sadness:
When the robot did not detect something within 90 cm distance.
Sound: sad sound

Disgustness:
When the robot detected some object that is within 90 cm distance and is not moving.
Sound: disgust sound

Anger:
When the user gets too close to the robot, the robot will start acting anger. (The distance between the user and distance sensor underneath certain range.)
Sound: anger sound

Fear:
After the robot express anger emotion, if the object is still getting closer to the robot, it will express fear.
Sound: fear sound

Shape

Aesthetics:
white color, basic, clean.

Light:
change adapt to the emotions


Materials used:

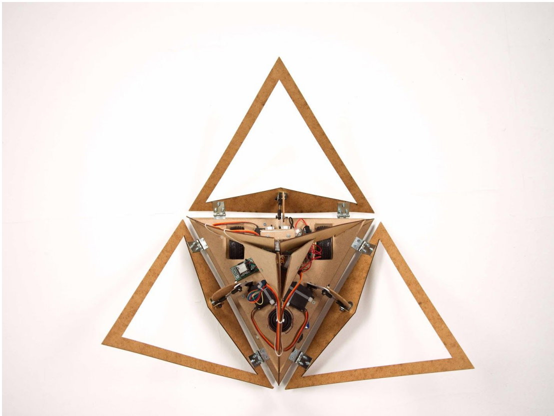
- base and structure: laser cutted MDF (medium density fiberboard) 2 mm of thickness
- cover: Spandex (LYCRA®)

Process:

1. Sketch the idea of shape and structure
2. Cad design of the shapes of the structure
3. Check and choose the best material keeping in mind that post-works
4. Laser cut the material
5. Mount every shape to build the final structure and the Spandex cover to finish the aesthetics.

Structure:

It is a tetrahedron composed by a laser-cutted modular structure, that takes shape through joints solutions.



Mechanics

Our final structure is composed by a tetrahedron, made of MDF wood, as well as the main body and the chassis (two bases), on which we are placing our components. Also, our robot has three wings attached at each side, which are made of MDF wood, too. The final material that we used it was designed using CAD software and produced by laser-cutter.

Inside the structure there are three servo motors, each one dedicated for the side-wings movement. The servo motors are connected to the wings through a mechanism, described below.

One of our first main problems was how to implement mechanically the movements of the three triangular side wings.

<u>First Approach:</u>

The first and simplest thought was to connect each wing with a servo motor, inside the internal part of the robot, by using a rope. The rope would connect the top of the wing with the servo. This solution, though, had many drawbacks and the most important was that the rope would be visible, because it would come out of the robot body, and that was something that would destroy our design. Then we were thinking to somehow hide the rope and connect it at the lowest part of the wing, but this solutions also came with several implementation issues.

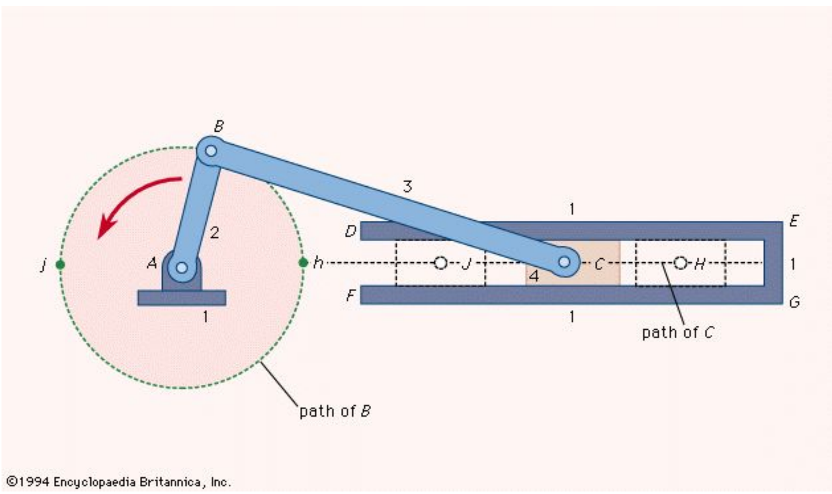
<u>Second Approach:</u>

Our next thought was to use gears in the lowest part of wings connected with the base. This was a much better solution, but then we realized that making the gears totally synchronized and connected with each other was a very demanding task and we didn't have neither the experience nor the time to make such a mechanism.

Before arriving to our final solution, we were thinking again about using a smaller rope connected to the wings, but also use springs connected to the external part of wings with the base, in order to control the movement. As expected, this solution was also a "bad" implementation and we would like to avoid it.

<u>Third (Final) Approach:</u>

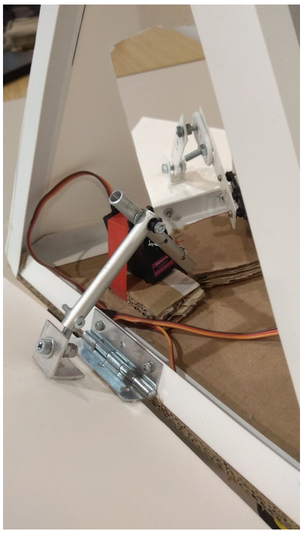
After all these attempts, we finally came out with our wing mechanism, which was much simpler and more robust than our previous solutions. We inspired by the way spiders move their legs and especially the spider robots. Our solutions consists of two bars connected to each other at one single point that can rotate. Then we attach the other two edges of bars, one at the wing and the other at the servo motor.



Working principle of the final mechanism

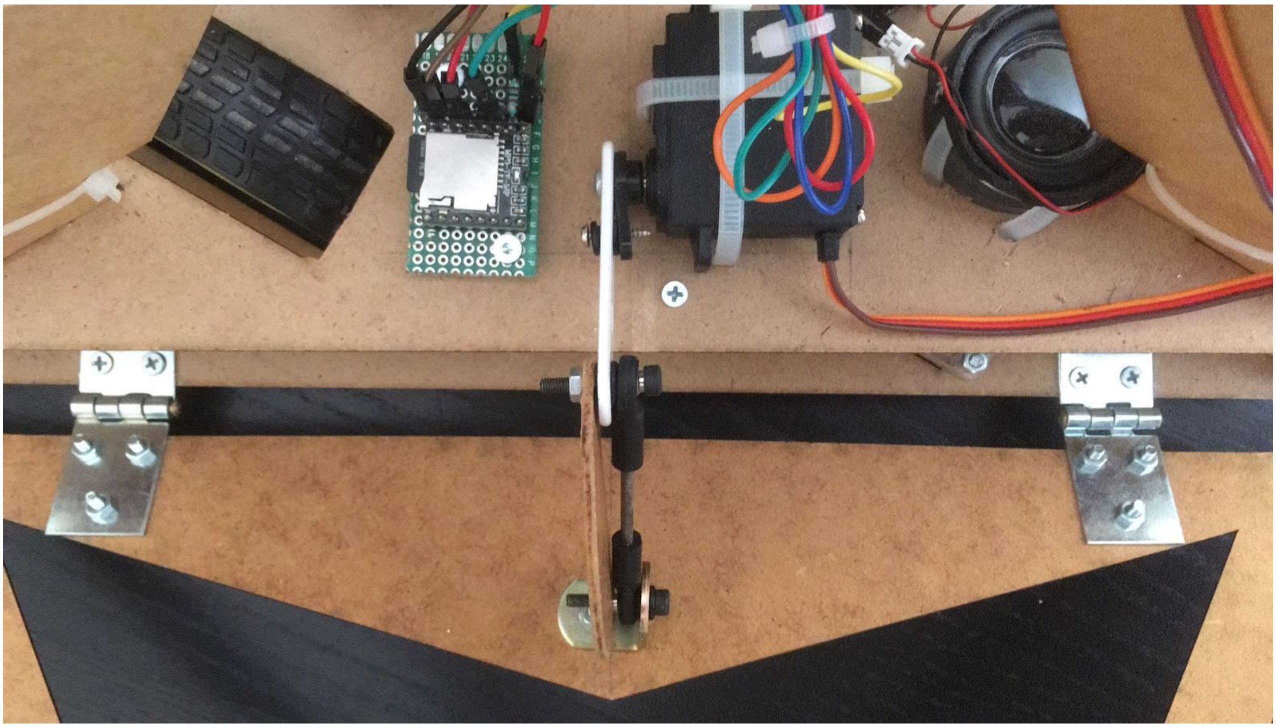
<u>1st implementation:</u>
Our first implementation was made by simple materials, like plastic, that were found in the lab. It was neither stable nor robust, but it was sufficient as a first prototype. It looked like this:

<u>2nd implementation:</u>
We wanted to make the wing movements as smooth as possible and so, we decided to add a small spring at the end of the mechanism for this reason. We also made it a bit more robust:



Final implementation:

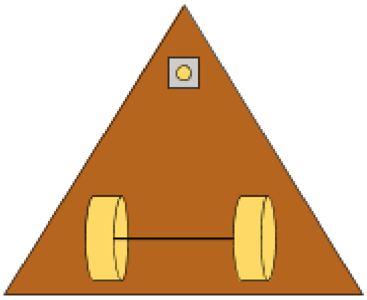
Our final implementation followed the principles of the 2nd one. but we removed the spring and also we used a more appropriate material, which is perfect for rotational movements. It is a specific type of ball-bearing mechanism. We also covered the mechanism with some semi-circular part, in order to make it smoother.



As you can notice from the pictures above, for attaching the wings to the base we used a mechanism called “hinge”, which is often met in door systems.

Wheels

We decided to use two wheels, at the front part, for the movements and the rotation of our robot and one supporting wheel at the back, without contributing to the movement, just for supporting the robot.

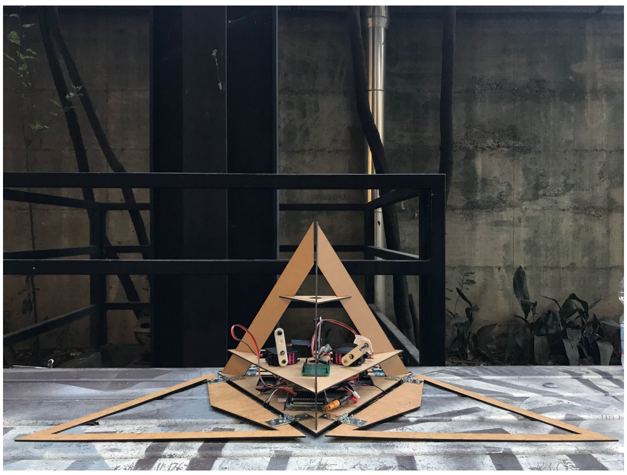
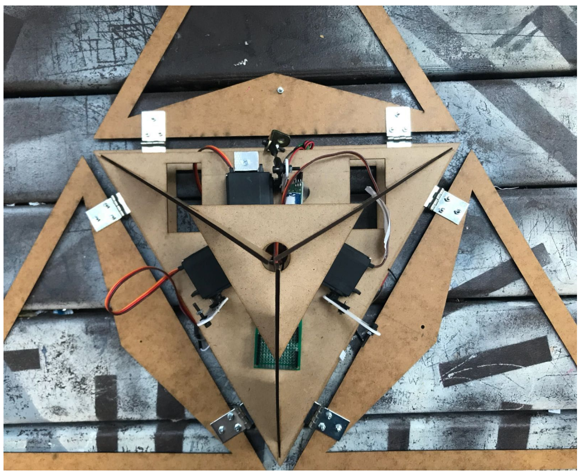


Base

At the beginning we had designed our robot with one main base, on which we would place all the electronic components. After the first prototype and the first attempts to place the servo motors along with the other components, it turned out that the space was very limited. In addition, we wanted also to hide the wheels and if possible to make them enter half inside the base.

So, we decided to modify our model by creating three bases. One bigger at the bottom that would help us to hide the wheels, carrying also the arduino, the DC motors, the DC motors' driver and the battery, and one second smaller placed a bit higher than the first one, that would carry the three servo motors and the speaker. The third base is the smallest one and it is placed above the second one. It carries the ultrasonic sensor and the led lights.

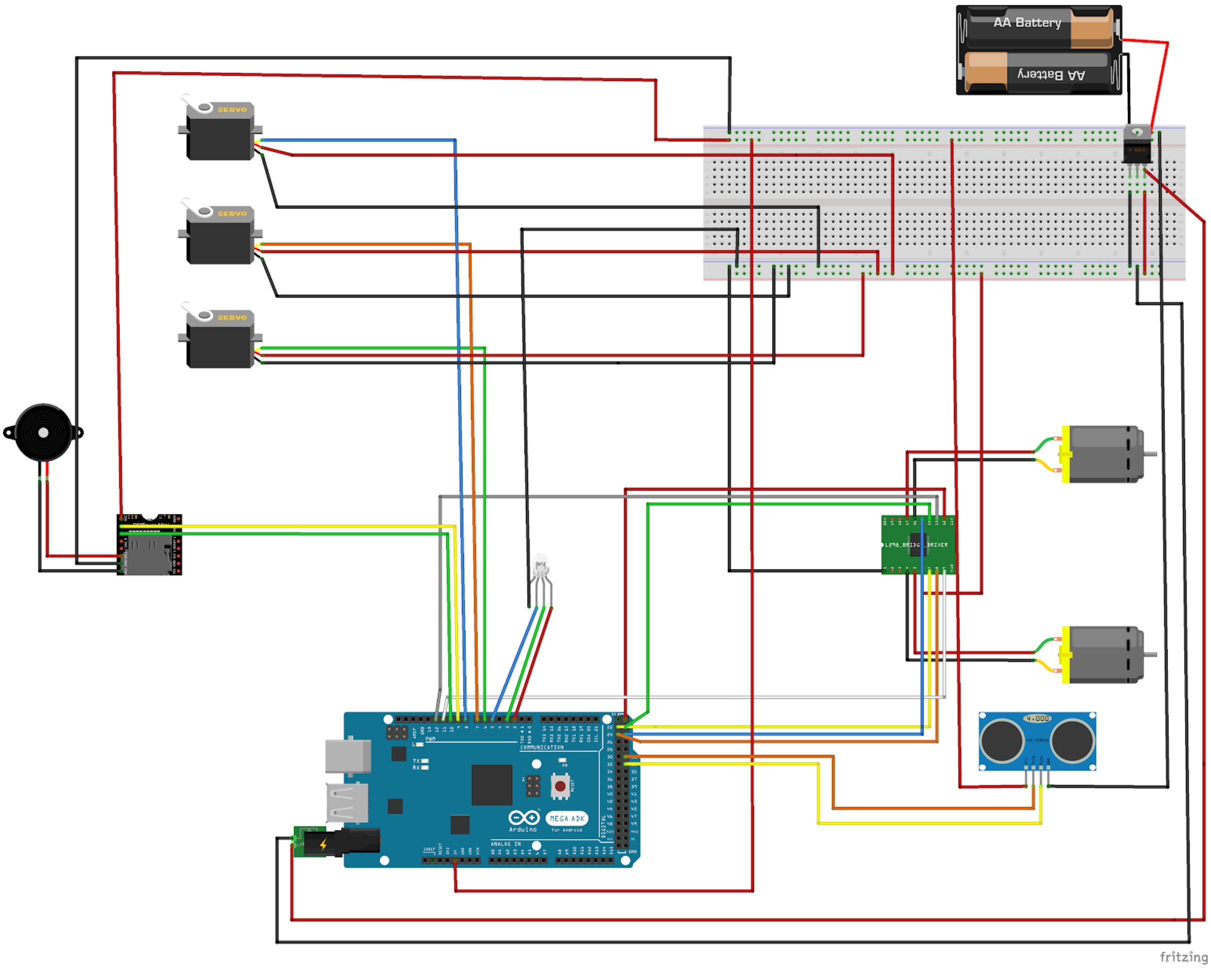
With this way, we managed to create more space and also make our structure better organised.

Electronics

For this project we've used a microcontroller and a set of actuators and sensors.
In this chapter, starting from the schematic of the circuit, all the single components will be analyzed separately, and at the end it will be also present a section regarding some notes about issues we have encountered during the development of the project. Datasheet and user manual of the component are present in bio.
The components we've used are:

1. Arduino
2. Power
3. Ubec
4. Driver and DC Motor
5. Servo
6. DF player and Speaker
7. Led
8. Sonar



<u>Arduino MEGA 2560 rev3</u>
The Arduino MEGA 2560 is designed for complex projects. With 54 digital I/O pins, 16 analog inputs and a larger memory it is the recommended board for robotics projects. This gives your projects plenty of room and opportunities.

The Mega is a microcontroller board based on the ATMega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Mega 2560 board can be programmed with the Arduino Software IDE.
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
The ATmega2560 on the Mega 2560 comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.
Each of the 54 digital pins on the Mega can be used as an input or output. They operate at 5 volts. In addition, some pins have specialized functions:
Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.
External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level.
PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output.
SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication. The SPI pins are also broken out on the ICSP header.
LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
TWI: 20 (SDA) and 21 (SCL). Support TWI communication.

The Mega 2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin.

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM.

The Mega 2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

<u>Power (Turnigy LIPO Battery 7.4V 2200mAh, Turnigy Ubec Buzzer 6.1V 3A)</u>

The system is powered by a LIPO of 7.4V and 2200mAh that could be turned on or off through a switch positioned on the base of the robot.

Actually there are 3 ways in which each components could be powered, depending on the recommended voltage they need in order to work.

Directly by the LIPO:

The only component powered in this way is the Arduino mega. In fact The Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector. Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

Through a Turnigy Ubec Buzzer of 6.1V and 3A:

The Ubec can low down an input voltage in the range of 6V to 23V to a stable DC voltage of 5.1V or 6.1V. The components that needs a lower voltage to the 7.4 of the LIPO but a larger power than the one that could be given by the Mega, are powered through this component. It also have a practical buzzer that is activated when the input voltage goes lower than 6V. It’s mandatory to have such alarm working with a LIPO since it could be damaged In case of excessive discharge.

The Ubec powered:

1. DC Motor
2. 3x Servo motor
3. Speaker

Through the Mega:

Also the Arduino could power the components, for this we’ve used the 5V pin. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the

board (7-12V). Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller
Also a A 3.3 volt could be supplied generated by the on-board regulator. Maximum current draw is 50 mA.
The Mega powered:
1. The Leds (Via PWM)
2. The Sonar
3. Control unit of driver and DF player

Of course all the ground are coupled.

Driver (L298N Dual H Bridge Driver) and DC Motor (Cylewet CYT1037)

In order to allow the movement in the space of the base of the robot, we've used a kit of two wheels each one included his own motor. They are controlled by a L298 Dual H Bridge driver, that is connected to Arduino with 3 pin for each wheel: Two are digital and allows to rotate the motor in one direction or another if they are maintained at a voltage one high and the other low or viceversa; The third is a PWM that modulates the velocity of each one and also works as an enable for the motor. The driver transmit those modulated information at the two motors thanks to two pins each. The driver allows also to power the motor with a voltage different than the one of the control unit, avoiding in this way possible damage at the microcontroller, that isn't capable of give a voltage and a current sufficient in order to make motors work. Those in fact need a voltage of 6V and a major power. The wheels are mounted on the frontal part of the base and a support wheel is mounted on the back, for warranty stability and fluidity of movement.

Servo (Mapei MG996R)

They are actuator, powered at 6V, that can convert electric signal given by the pin of Arduino in rotation of the crankshaft between 0-180°. In this project we've used three of them in order to have a independent movement of each wing, to obtain a wider range of movement for the emotions. They're joint to a mechanism of two bars between the middle base and each wings, that allows the complete excursion of the wing from the Body of EDRo to the floor.

Led (RGB LED Common Cathode)
They are led containing the colored diode (1 red, 1 green, 1 blue) that allows to obtain a full range of colors. The digital modulated pulses sent to the anodes change the light intensity of each diode. The common cathode has to be connected to ground.
In this project we've used 10 led connected in parallel in order to light the eye of different colors depending on the emotion.
Considering that we've powered them with a PWM no resistor were used to lower the voltage but consideration on the optimal duty cycle in order to obtain the optimal voltage for each diode (2V red, 3.5V blue and green) and on the current to obtain a good illumination of the led (20 mA) were made.

Sonar (HC electronic HC-SR04)

They’re ultrasound sensors, powered at 5V, that were used for the obstacle avoidance and the interaction. They use ultrasound and we choose them because the have a minor cost than electromagnetic radar or infrared sensors. They measure the time of flight (ms) of ultrasound emitted by a source to come back if they encountered an object. They have 4 pins: V+ (5V), trigger, echo and GND. It’s functioning is the following: After a pulse on the trigger pin the sonar emits a signal of high acoustic energy (ping), and wait for the return of reflected waves. At this point it respond on the eco pin with a high pulse with the duration that corresponds at the time of flight of the ultrasound. After 30 ms, if no ultrasound waves return, it’s considered like no obstacle has been encountered. For safety 50-60ms are waited in order to not have interference with the sequent measure. The have a good accuracy between 2-400cm. In this project only a sonar has been utilized, but future development of the robot could consider the use of more of them to improve the interaction and the obstacle avoidance.

Notes

Regarding the electronics our main goals were:

1. Realize compact circuitry, in order to make everything fit inside the robot. So we have choose the optimal internal design (based on different layer) for placing the components and we’ve used stripboards and soldering when possible.
2. Choose the right component in order to optimize resources., that are mainly costs but also power consumption. So the solution was to realize the structure with a very light weight in order to permit the use of low power Servo and DC motor, and also find solutions in order to use the smaller amount possible of leds.

Fire

After a first try, in which we’ve powered the servo and the DC motor at 5V, we’ve observed that the movement of the robot were not the one that we expected. So, to obtain a movement as fast as possible, in order to make emotion more expressive, we’ve pushed up a little such component connecting directly to the LIPO. After a first day of perfect working, in which we’ve obtained exactly what we want, by the time that we had to recharge the battery and it returned to operate at 8.1V we burned not one but two servos. At the end we’ve decided to use a Ubec for stabilizing the voltage at 6V in order to have the right compromises between performance and a not exploding robot. So a suggestion, feel free to experience, but do not joke around with fire (especially if your robot is made of wood).

Light

We spent a lot of time to decide how to light the robot. We pass through different option like, lightened all the internal in order to show all the cables and inner structure as shadow on the tissue that cover the robot, realize some peculiar shape to drive the light in order to have a definitive form on the surface and so on. Also technically we’ve many doubt about the types of led to use, simple RGB Led at common cathode, addressable Led, Led stripes…

At the end we’ve decided to optimize cost and consumption using simple RGB Led, but to obtain however a good results we’ve designed a structure in the upper part of the robot in order to obtain lightened circle, something that looks like an eye.

Informatics

The robot express the five emotions using the following four main component:

1. Wings
2. Wheels
3. Sound
4. Light

while the interaction it's given by the use of an ultrasonic sensor to recognize the presence of someone. The code has to control all this components.

First Approach

One sketch for each one of this component, where are coded the functions they have to furnish in order to express emotion (servo_joy, servo_sadness …), which will be called from the main sketch which is responsible of decide which emotion to express based on the input he is detecting.
This was the first idea, where all the emotion function were synchronized using the delay() function. But we soon realize that this approach doesn't allow us to control multiple task at the same time. So we have changed the structure of the code in order to substitute the use of the delay() function with the millis() function that allow us to use timer in order to synchronize all the emotion, as it is described below.

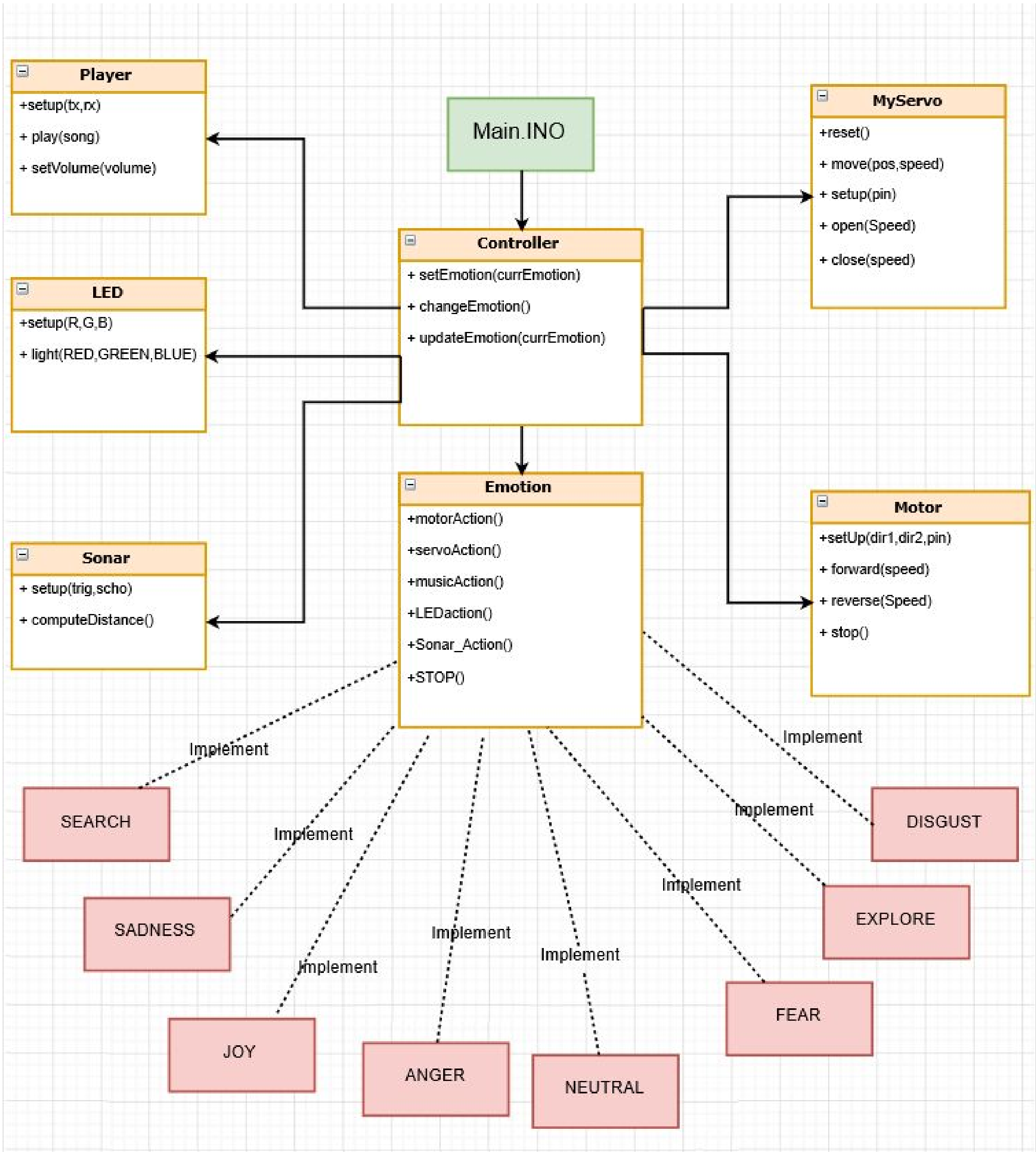
Second approach:
Here is a high level description of the code:
As you can see from the class diagram there are 7 main class, 8 derived class and one sketch which is the file which arduino it's used to work with.

Main

This is a basic arduino sketch with only the setup() and the loop() functions, plus the declaration of all the hw component's class (Servo, Sonar, …) and of the controller class, which is the real core of the robot.
In the setup function there is the pin assignment of the hardware component.
In the loop function is called the method updateEmotion of the controller passing to it the current millisecond given by the millis() function of the Arduino library.

Controller

The real core of the robot is this class, which is responsible of changing the emotion based on the input received from the outside and also give the ability to the robot to accomplish multiple action at the same time, simply scheduling them.
Basically, when the loop function in the main sketch calls the updateEmotion method, the controller, based on the emotion in which the robot it is, checks if a timer is expired and if so calls the right method of the current emotion in order to perform the action required, that could be one from the actions listed in the emotion class.



Emotion
The emotion class is just an interface from where all the other emotion class derive the main methods and variables which are common for all the emotions. Here follow a list of all the variables and methods common to all the emotion.

Times, which are used by the controller to know if an emotion is ended or not:
- emotion_started;
- emotion_duration;

Intervals, which are used by the controller to know how frequent an action has to be performed:
- motor_interval;
- servo_interval;
- led_interval;
- music_interval;
- sonar_interval;

Last call, which are used by the controller to know how much time is elapsed since the last time the action was performed:
- motor_last_millis
- servo_last_millis
- led_last_millis
- music_last_millis
- sonar_last_millis

Actions, which implement the action that the emotion has to perform
- motor_action();
- servo_action();
- led_action();
- music_action();
- sonar_action()

This interface allow us to implement a state design pattern in order to reduce at minimum the code replication and make the code more readable.

MyServo
This class is a wrapper of the library VarSpeedServo.h, which we have done in order to better integrate with all the other code. It's simply allow us to move the servo motors at different speeds in a simply and clean way.

Motor
Given the fact we have a hardware driver which control the wheels, in the code we just need to drive HIGH or LOW digital pins in order to move the wheels forward or backward, and use

pwm pins in order to control the speed of the wheels. So this class it's just a wrapper which allow us to abstract from the pin configuration and furnish easy and human readable method to control the wheels.

Led

This class expose the methods to allow us to control the RGB leds, in order to light it up in different color and to switch it off when we don't need them.

Player

The player class is the class that control the MP3 DFPlayer which allow us to reproduce mp3 sound on our robot. The class expose all the methods that we need in order to play different songs and set the volume.

Sonar

The Sonar class is a wrapper of the NewPing library which simply compute the distance of the nearest object when it is requested to do so, and allow us to do obstacle recognition. The maximum distance that can detect is set to 2 meter.

These are the libraries used:
- VarSpeedServo.h, to control the servo-motors which moves the wings;
- NewPing.h, to control the sonar sensor;
-
- Arduino.h, which has all the main functions (like Serial, millis() and so on…)

Conclusion

For this interactive robot project, it was absolutely a fun experience building this robot. Every step from scratching to the final building involves a lot of group work. And from designing the robot appearance to build the final mechanic and interaction inside, every step involves careful decision making. And throughout the whole period which is roughly around two months, we are constantly improving our design according to the suggestion and weekly feedback from the professor.

As a group, we have learnt the importance of cooperation and being organized at all time. And each team member individually is also working towards the best result of the robot. Everyone has a different cultural and academic background, so we divide the work wisely according to the role of the member. Everyone has their own focus area and is also work on the whole structure of the project at the meantime. Throughout the building process, we have also encountered several difficulties. First challenges was the mechanic, it is easy to make a movement, but it is hard to have a vivid movement with aesthetic perception. So after several versions of mechanism testing, we have finally decide on the right one.

During the making of the robot, the other challenge was related to the distance sensor. As we planned the layer of the wings should be two. While the thickness of the material(fabric) is affecting the performance of distance sensor, so after several adjustment, we decided to keep only one layer. What’s more, when we were at the final stage of building the wheels of the robot, the servo motor burnt in an incidence. And there is countless times that we could not find the perfect material at the right time, or we run out of the screws or some other stuff. But none of these problem or difficulties has affected us, in the final stage, we have finally accomplished the ultimate edition of our interactive robot.

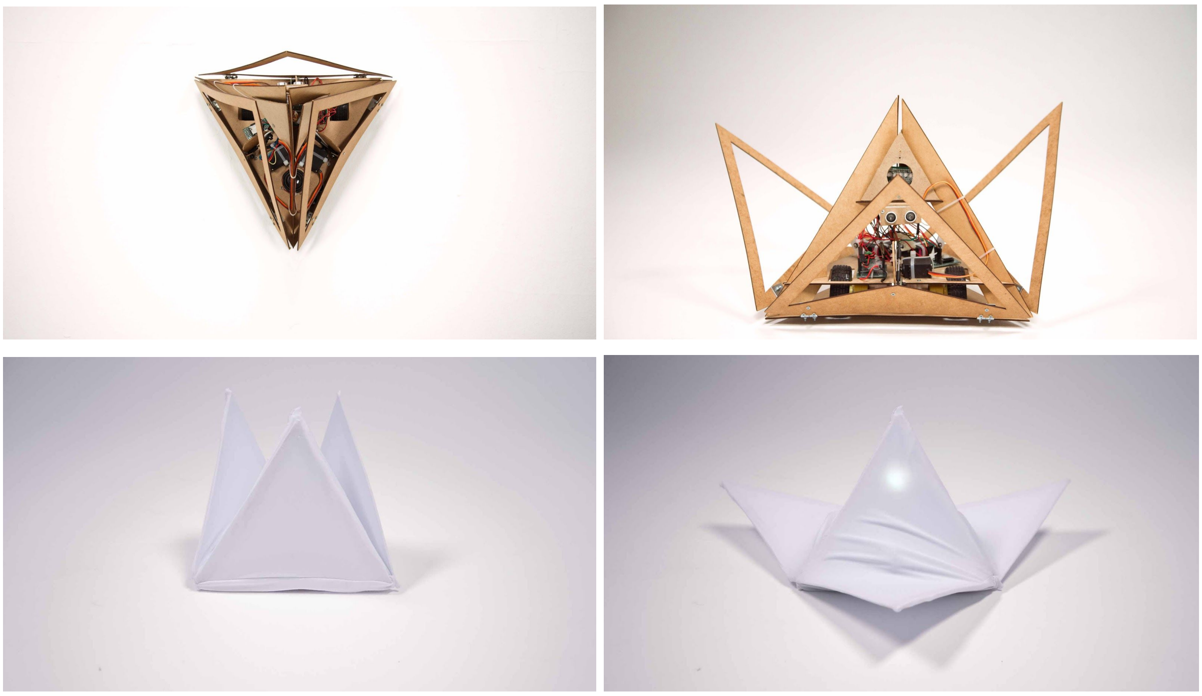
Annex

Bill of materials

- 12,99€ Arduino mega
- 19€ 7.4 V Lipo battery
- 20€ 3 x Servo motors
- 7,30€ DC motors’ driver
- 4,69€ 2 DC motor with wheels
- 6,99€ Cables
- 1,50€ Ultrasonic sensors
- 6,89€ 6V Regulator
- 7,99€ Mini MP3 DFPlayer
- 8,30€ 3W speaker
- 8,28€ Micro SD Card
- 2,29€ LEDs
- 0,5€ Switch button
- 30€ Laser-cut
- 24,5€ MDF wood (structure)
- 15€ Fabric
- 15€ Ball-bearing mechanisms
- 15€ Screws etc.

TOTAL: 206,22€

Shape



Electronics

Link to datasheet:

- Servo: https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf
- Dc motor Driver: https://www.st.com/resource/en/datasheet/l298.pdf
- Sonar: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
- DFPlayer: http://www.picaxe.com/docs/spe033.pdf

Informatics

Link to the public github repository: https://github.com/uino95/EDRO
Link for library download:
VarSpeedServo.h: https://github.com/netlabtoolkit/VarSpeedServo
DFRobotDFPlayerMini.h: https://github.com/DFRobot/DFRobotDFPlayerMini
For what it regards the NewPing library is already included in the code of the robot because we need to modify it a bit in order to better integrate with all the other code.

User Manual

Switch on the robot:
In order to switch on the robot you have to push the button that you can find on the bottom of the robot near the third wheels.

Switch off the robot:
In order to switch off the robot you just have to re-push the same button.

Remove the arduino:
In order to remove the arduino you have to follow this steps:

1. Remove the cover;
2. On the right part of the robot (if the robot is facing you), you could see the arduino, and on the upper part there is a small nut that you can unscrew with your hands;
3. Turn the robot upside down;
4. Turn the small metal bar of 90 degrees;
5. Push the arduino inside of the robot in order to free it from the screw in the upper part;
6. Pull the arduino off;
7. Unplug all the cable if you need to remove it.

Change the battery:
In order to change the battery you have to follow this steps:

1. Remove the cover;
2. On the left part of the robot (if the robot is facing you) you can see the connections between the switch and Lipo battery, disconnect it;
3. The battery is attached to the robot with a velcro strip, so you have just to pull it off;

Interaction:
The Robot will start to move in the space, in its neutral configuration (White light on the front face, wings partially open), until it encounters an object (detected with sonar) and it becomes happy, otherwise if it doesn't encounters an object in 10 seconds he becomes sad.
If he has encountered the object he starts investigating the object, types of interaction:
1. If the object in front of him moves: it become happy (Yellow light on the front face, small rotation in alternated direction on itself, wings open and close simultaneously, happy sound)
2. If the object in front of him doesn't moves: it become disgusted (Green light on the front face, backwards translation, front wings open, the other two closed, disgusted sound)
3. If the object it's too near (goes through him) for a first time it becomes angry (Red light on the front face, complete rotation on itself with wings closed, translation upwards with the front wing open and the other two open and close simultaneously and fast, angry sound)

4. If the object it’s too near (goes through him) for a second time it becomes scared (Purple light on the front face, vibration with wings closed, fear sound)

After each types of interaction the robot returns in it’s neutral exploring state, but for the happiness state, after the happiness state he starts investigating the object in front of him.