Rafik Hariri University
جامعة رفيق الحريري

Summative Learning Project

Friendly Robot Interacting with Children

Done by

Mohammad Daoud

Mohammad Al Kaderi

Karim Al Harake

Bassem Bashnak

Rafik Hariri University, Department of Mechanical and Mechatronics Engineering, Mechatronics Engineering

Submitted to

Dr. Hassan Hariri

In partial fulfillment of the requirements of the BE degree in Mechatronics Engineering Major of the College of Engineering at Rafik Hariri University

MECHREF, LEBANON

January, 2021

Mohammad Daoud (2017-0106)

Mohammad Al Kaderi (2017-0262)

Karim Al Harake (2016-0132)

Bassem Bashnak (2015-0304)

ACKNOWLEDGEMENTS

We would like to thank everyone who has guided us and helped us in this project, starting from Doctor Hassan Hariri, who answered all our inquiries, gave us ideas and guided us through every hardship.

We would also like to thank Mr. Anas Al Shaghouri and Mr. Hassan Daoud, who have also helped us solve any problem we encountered.

We would like to express our deepest gratitude to Rafik Hariri University and to its entire staff and faculty.

Finally, we dedicate this project to our guardians, who devoted their lives for us and our well-being and success, and without them, we wouldn’t be who we are today.

ABSTRACT

The objective of this project is to design a friendly robot that interacts and plays with children. To do that, the shape and structure of the robot are to be designed using 3-D software then built, face and emotion detection algorithms and techniques are to be used, and the robot should be programmed in order to perform certain actions such as changing its eye expression and producing sounds and lights. As a result, we should have a stationary robot with an aesthetic shape that detects what a child is feeling, then performs the appropriate reactions to entertain and provide company for the child.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Motivation

Children want attention. They are always in need of someone or something to distract them and fulfill their emotional needs. That's what parents or siblings do. But what if the parents and siblings are busy? Some parents have full time jobs and barely have time to socialize and connect with their child, and siblings have other things to attend to as well and can't always be there to play and interact with their little brother or sister. That's what the robot we are designing is for. It connects with the child and keeps it company whenever his or her family members are not available. Of course, nothing can replace the roles of the parents and family members, but still, this robot provides some added value and some gap filling. In addition to that, studies have shown that interactive robots are promising tools in helping autistic children, under the right circumstances [1]. Children with autism have a deficit in social interaction [2], therefore, presenting an artificial form of interaction may be very helpful.

1.2 Overview

The robot we are going to build is a circular shaped robot, with a circular head that surrounds the body. The robot itself is stationary, but its head moves around the body with the action of 3 servo motors. It is very similar in mechanism to the Mira robot [1]. It has 2 OLED screen eyes that have an oval shape and blink. In addition, they change their shape according to the emotion it is displaying. The objective is to make the robot's head roam freely, detect a face, follow it and detect which emotion-neutral, happy, sad, angry or surprised- the child (or any person) is feeling depending on their facial expressions, using a

Raspberry Pi Camera v2, and imitate the emotion by changing its eye shape and producing sounds and lights accordingly. As for the lights, we are using an RGB LED strip and for the sounds, we are using a speaker that generates sounds and music. Both are programmed using Arduino. For the face and emotion detection, we are using algorithms and methods such as detectnet using Jetson Inference library. Finally, we are going to establish communication between the Jetson Nano and the Arduino in order to achieve the complete functioning of the robot.

1.3 Constraints and Requirements

Now there are some requirements that we are willing to meet and some constraints we should abide by when designing and building our robot. Those requirements and constraints are:

Constraints:

1- Safe: no pointy parts, electrical components are out of reach...

2- Maintainable: All components are reachable.

3- Age: 5+

4- Size: 20 cm diameter circle maximum and must fit all components

5- Price budget: a maximum of 200 dollars considering the difficult economic situation in Lebanon

Requirements:

1- real time emotion recognition and reaction

2- Small camera: raspberry pi camera

3- Cheap: 3d printed, simple

4- Children-friendly: cute and simple design

5- Safe: electrically insulated

6- Plug to power: cheaper and safer

More requirements and constraints may be added later.

The robot should be children friendly, so having no pointy edges, as well as no visible electrical components, is a must. Also, we want our robot to be maintainable, we want to design it in a way that we can easily reach the components and work with them if needed. We also don't want our robot to be too big; we want it to be maneuverable. As for the requirements, we want our robot to have a high performance, to have high accuracy and speed in reacting to the child. And we want it to be a cute aesthetic robot, but we also want it to be simple, not too sophisticated in shape. Last but not least, the robot should be either powered by rechargeable batteries, or plugged to an outlet for power.

1.4 Flow Chart and Gantt Chart

Below is a flow chart that describes in a simplified way the way our robot works, and a Gantt chart of how we are going to proceed with our work: what our tasks are going to be and the time plan for their completion.

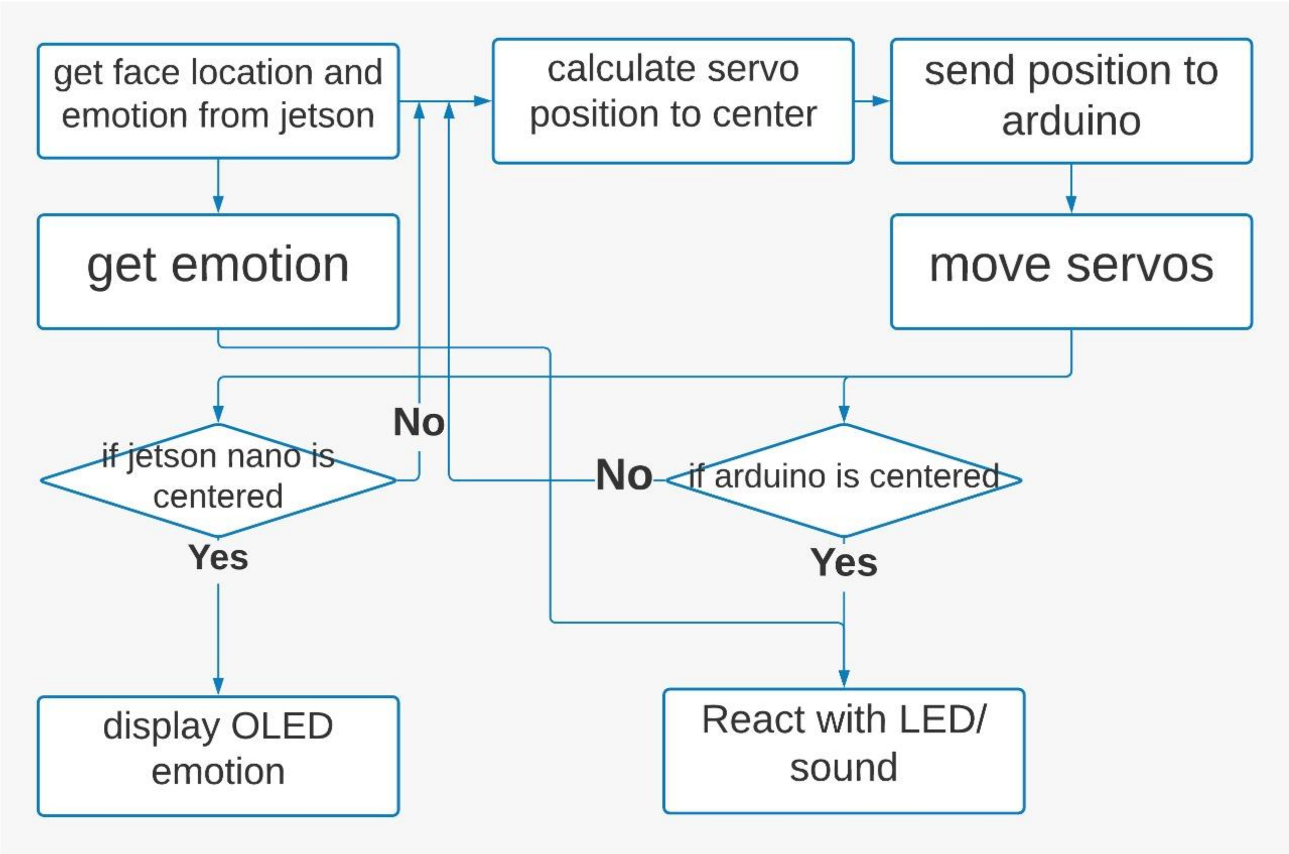


Figure 1: Flow Chart of the Function of the Robot

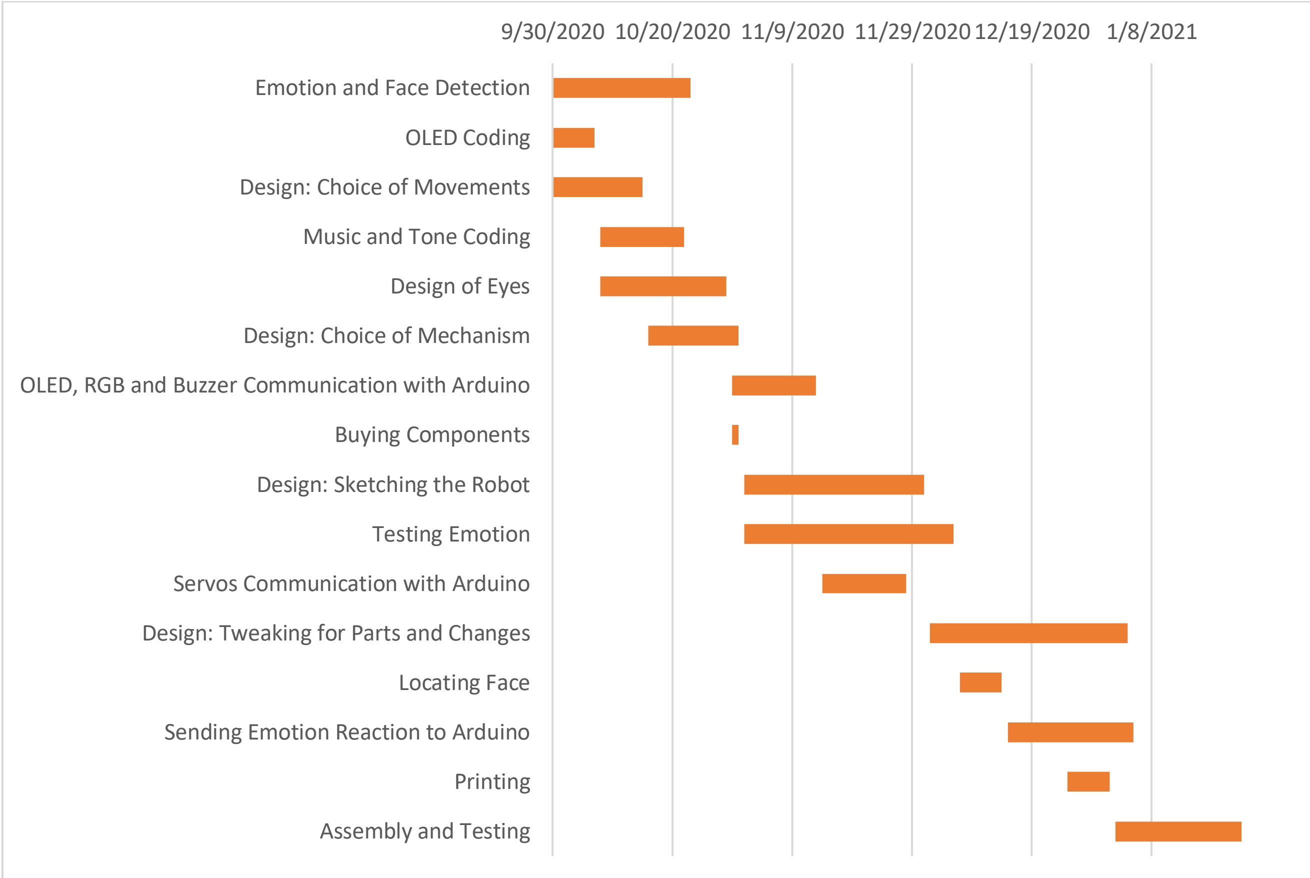


Figure 2: Gantt Chart

We finished our product a bit later than the end date because of the lockdowns.

CHAPTER 2

Literature Review

2.1 Overview on Similar Robots

We did our project based on Mira and Pia robots, which serve the same purpose as ours: interacting with children. Mira robot was designed to communicate with kids using an emotional and not a verbal language [32], and our robot does the same. It was made by the talented Pixar artist Alonso Martinez. He wanted to bring the characters on screen to life so that people could interact with them. Mira robot can blink. It also moves her head to follow a human face, plays games such as peek-a-boo, shows a surprised reaction when you point your finger towards her, and giggles when you kiss her [30]. Our robot will do almost the same things but with modifications and additions. It will follow the human face, and it will show 5 different emotions with facial expressions, lights and sounds, depending on the emotion it detects on the face.

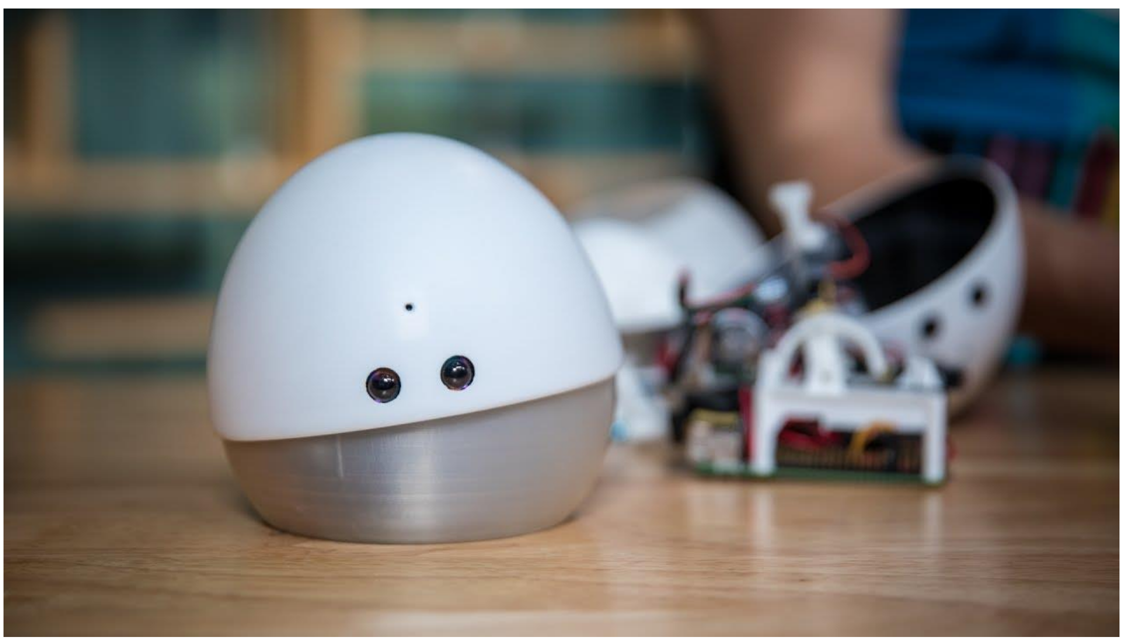


Figure 3: Mira Robot

Mira is an egg-shaped robot that can roll, pitch and yaw its head in 3 dimensions. This mechanism is controlled by a microservo-powered floating-joint mechanism, which is inspired by a video game controller! And as Mira moves its head around until it detects a face, it blinks using 2

OLED screens with hemispherical orbs glued behind them. And it recognizes faces from a small camera between its eyes, and the data is processed using OpenCV on a Raspberry Pi. Finally, it responds to actions with sounds from a speaker, as well as change its tint using a bright RGB LED [32]. Pia Robot, however, is made of two 3D printed concentric spheres, 3 servos, a gimble system, a Raspberry Pi 3 and a camera, an Adafruit 12-bit servo driver, and Adafruit's Circuit Playground Express which contains a variety of sensors [33]. In our project, we used 3 servo motors like Pia robot, talked about later, we used an RGB strip like Mira, and the same lighting technique (tint) as Mira. However, we didn't use OpenCV and Raspberry Pi for face detection; we used deep learning techniques instead talked about later.

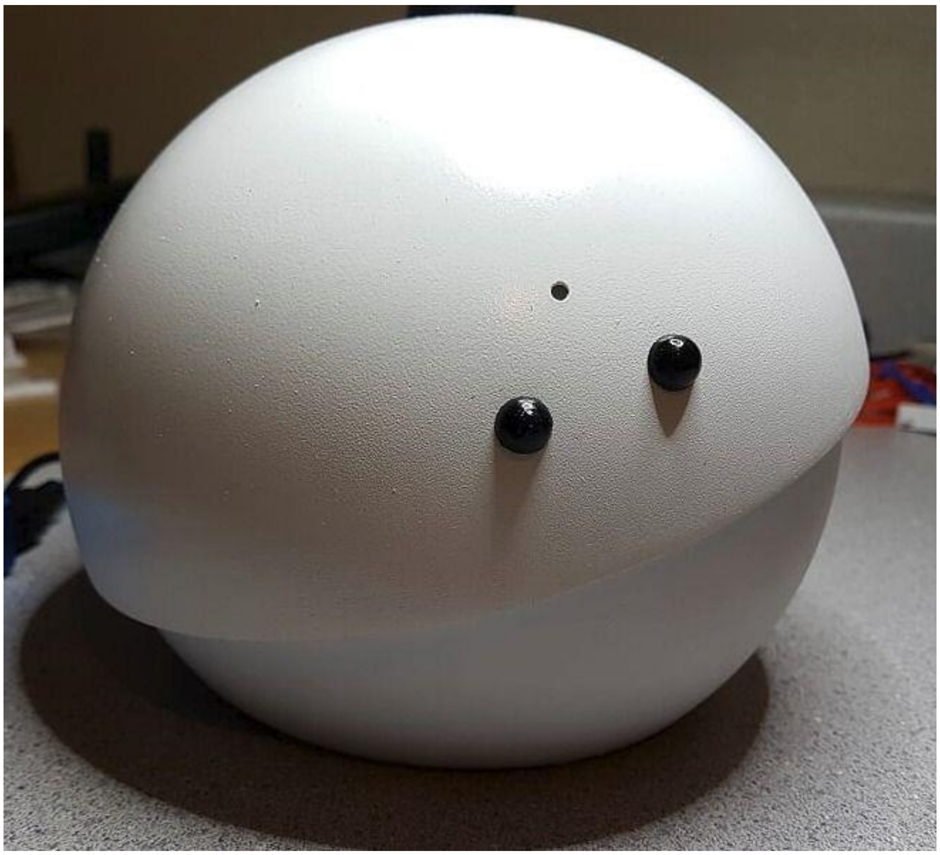


Figure 4: Pia Robot

As we can see, Mira and Pia robots are very similar in shape, mechanism and functionality with just minor differences. Our robot will also be similar but also with some differences and additions.

2.2 Robot's Mechanism

Some of the mechanisms we looked at are:

We first had a look at a mechanism that uses two servos mounted on top of each other. These servos are mounted into brackets that hold them in position. The lower servo performs rotation about the vertical axis, and the top servo tilts the mechanism forward and backward, as it is mounted sideways. The figure below shows a model of the mechanism.

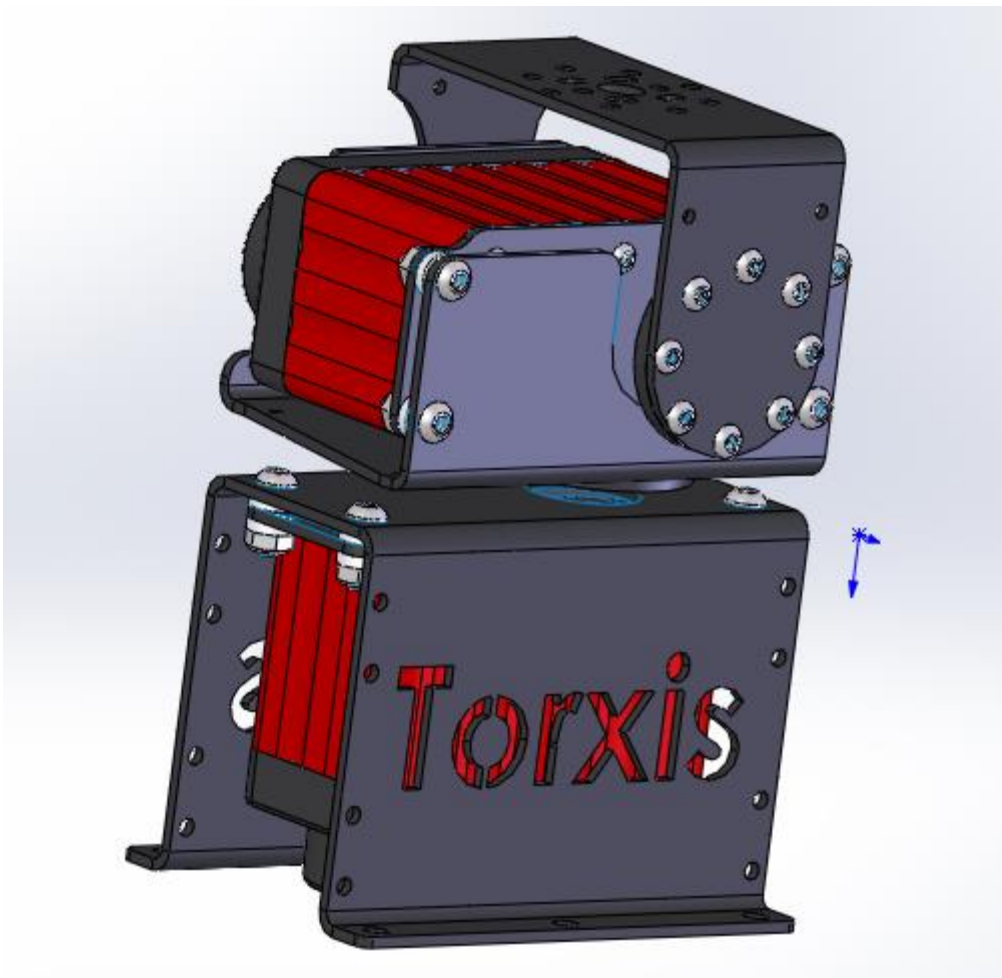


Figure 5: Torxis Pan Tilt

We discarded this mechanism for two main reasons. First is the fact that this mechanism does not accommodate tilting sideways, which we need for our robot, and second is the fact that this mechanism mounts several servos above each other, which leads to jerky operation upon assembly, especially if the power delivered to the servos varies.

The second alternative we looked at is a pan-tilt mechanism we saw in a website. It is comprised of 3D printed brackets that is similar in functionality to alternative 1, but differs in shape. For the same reasons we discarded the first alternative, we discarded this one too. Below is a figure that demonstrates what the mechanism looks like.

Figure 6: Pan Tilt Alternative

These mechanisms or similar is what we found others working with, so we had to improvise.

The mechanism we deployed in the robot is the joystick mechanism, with extras. The joystick mechanism is used in gaming console controllers and other applications. It can pan and tilt (perform incomplete rotations about the axes in its plane. We add to that an element that can rotate the head about the axis perpendicular to the mechanism plane. , the result is the ability to tilt the head forward and backward, as well as left and right. The head is also able to rotate completely around the vertical axis.

We started with a basic joystick mechanism. This mechanism can be found in gaming consoles' controllers (like play station and Xbox). It is comprised of two semi-circular arcs, which rotate about two axes perpendicular to each other. The axes of rotation intersect at a point that is the pivot point of the arm that will tilt forward/backward and right/left.

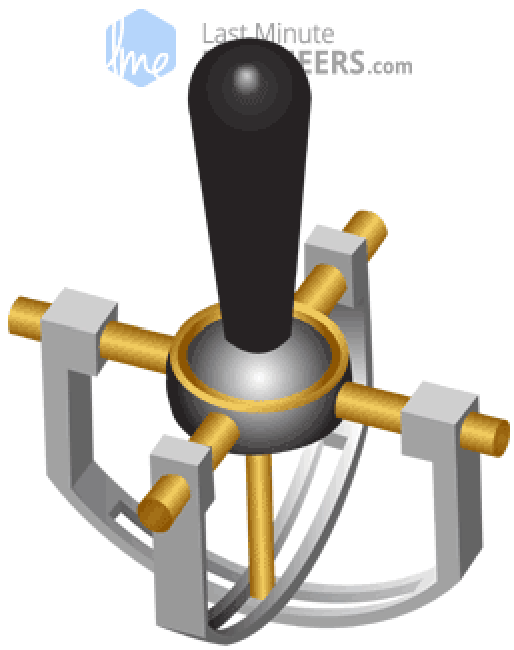


Figure 7: Joystick Mechanism

We add to the top of the moving shaft a mount for one extra servo. This will rotate the head about the vertical axis and our mechanism is done. The inspiration for this mechanism is the PIA robot.

Figure 8: Pia Mechanism

The design of the robot parts and body will be done on DS SolidWorks software. This software allows us to model the robot parts, assemble them, and animate the moving parts in what is similar to real life.

After the design is done, the drawn parts will move into the 3D printing phase.

2.3 Actuation of Mechanism:

The actuation of the mechanism will be done using MG90s metal-gear micro servomotors. These servos provide enough torque without consuming high power, and do not need extra circuitry to connect and make them work. We looked at the datasheet of the servo motor, and it turns out it can accommodate up to 1.8 kgf.cm at 4.8 volts which is plenty for this application [7].

A possible alternative is the Microservo 9G which is similar to the servo we are using, but is made with plastic gears, and can accommodate 2.5 kg.cm3 [9]. However, on the long term, it might become prone to wear and fatigue, so a safer and more reliable option is the way to go.

The stepper motor is a good and strong alternative for the servo, but we discarded it as an option due to the fact that it requires extra circuitry and it is big in size, which is not in our favor in this case.

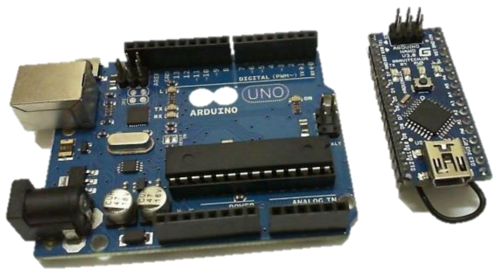
Figure 9: MG90s Servo Motor

2.4 Microcontrollers:

Reacting to the 5 (happy, sad, angry, neutral, surprised) emotions is easy to human but to machines it is kind challenging. The reacting to the emotions needs high quality controller that communicate between the components used and the output of the Jeston Nano. The communication must be smooth and fast to get the best reaction on the robot. Therefore, we will used microcontrollers to accomplish the job. Microcontrollers are belonging to group of embedded systems, which are applicable for control of processes and product functions. This purpose needs to capture information about the product and about surround. These information's have to be processed inside the microcontroller and after processing, the microcontroller makes decision about next steps if it is necessary. Data capturing and processing is the main role of microcontroller [10]. There are many types of microcontrollers: Arduino Uno/nano Microcontroller Board, nodemcu esp8266, Teensy 4.0 and ESP32 Microcontroller Board.

2.4.1 Arduino Uno/Nano Microcontroller Board:

It is based on the chip named ATmega328P (open-source board). The board has various I/O pins with which you can interface it to other boards and circuits. There are various ports connected including a USB connection port, fourteen number of I/O pins, an ICSP header, a power supply connection, and a reset press. It can be easily connected directly with your personal computer or laptop via a USB cable. Based on IoT, the Arduino board is the cheapest, ready to connect the board with various available online libraries and resources [11].



2.4.2 Nodemcu ESP8266:

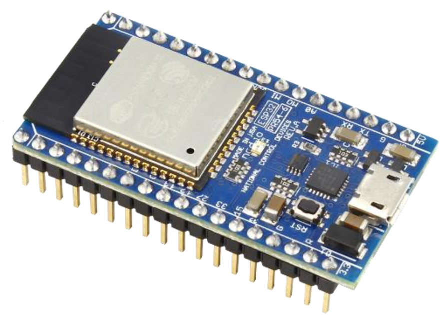
Well Nodemcu esp8266 board is actually the same as Arduino board but the main difference is that esp8266 has bigger storage and high performance but with less I/O pins. In addition, esp8266 has esp 12 chip for wifi connections.

2.4.3 Teensy 4.0:

Teensy 4.0 microcontroller board (600MHz processor) is the latest and fastest board available today. It is of small size as compared to other boards and can be used to make various types of DIY projects. All the commands are given via two USB ports to the board. Arduino IDE with a little Add-on can program teensy 4.0. The microcontroller can be connected to a PC or laptop with a USB cable. It has 1024k RAM compared to 16k RAM of Arduino Uno for more cutting-edge applications [11].

2.4.4 ESP32 Microcontroller Board:

The ESP32 microcontroller board is Bluetooth and Wi-Fi duo combo on a single-chip board (2.4 Giga-Hertz) with ultra-low power consumption. The board is considered the best choice for applications where the best RF performance is required. The board is a bit costly but its power features pay the price. The ESP32 microcontroller board is used for DIY projects like smart home and IoT based projects [11].



After checking all the possible choices that we can use in the project, Arduino Nano board is the best with respect to storage, performance, size and price. The microcontroller that we will choose is only responsible for communicating and controlling the components with respect to the input from the Jetson Nano. Nodemcu esp8266 and Esp32 microcontroller boards having additional features as WIFI and Bluetooth modules, but these features are useless in our case because all connections are wiring and not wireless. Teensy 4.0 microcontroller board is used by MIRA robot, but this board is a bit buffer and has a very high performance higher than what we need in the project. Moreover, Teensy 4.0 is a very high cost board. Arduino with flash storage is 32 KB and clock speed 16 MHz is the best to use in this project having the best size, price and very fast performance.

2.5 Eye Components

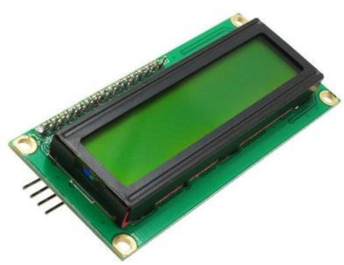
The eyes are very important part in the face that shows half the reaction of anything (Human, animals, paintings…). Moreover, since the robot has only a face, we innovated eye reactions and impressions to show a perfect and smooth reaction of the robot. There are many components that fits in this place: Oled screen, LCD screen, 5110 Display and TFT LCD.

2.5.1 OLED screen:

These screens look similar to the 5110 screens, but they are a significant upgrade. The standard 0.96-Inch screens are 128 x 64 monochrome, and come with a backlight as standard. They connect to your Arduino using I2C, meaning that alongside the V+ and GND pins, only two further pins are required to communicate with the screen. With various sizes and full color options available, these displays are incredibly versatile [12].

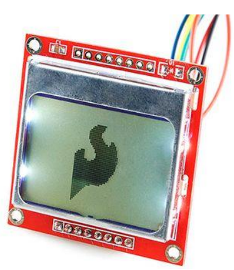
2.5.2 LCD screen:

This type of display can vary in design. Some are larger, with more character spaces and rows, some come with a backlight. Most attach directly to the board through 8 or 12 connections to the Arduino pins, making them incompatible with boards with fewer pins available. In this instance, buy a screen with an I2C adapter, allowing control using only 4 pins [12].



2.5.3 5110 Display:

These tiny LCD screens are monochrome and have a screen size of 84 x 48 pixels, but don't let that fool you. These displays are incredibly cheap and usually come with a backlight as standard. Depending on which library you use, the screen can display multiple lines of text in various fonts. It's also capable of displaying images, and there is free software designed to help get your creations on screen. While the



refresh rate is too slow for detailed animations, these screens are hardy enough to be included in long-term, always-on projects [12].

2.5.4 TFT LCD

Thin-film-transistor liquid-crystal displays (TFT LCDs) are in many ways another step up in quality when it comes to options for adding a screen to your Arduino. Available with or without touchscreen functionality, they also add the ability to load bitmap files from an on-board micro SD card slot [12].

After checking the above components, the best component to use is the Oled screen. Simply because Oled can read frames and print then in a very smooth and attracting way. Therefore, we can easily upload many frames and run them in a specific order to get the output needed. In addition, to make the output of the Oled screen more clear and interactive, we will add dome glass to screen. Therefore, eyes reaction will be very clear, interactive and smooth specially the blinking eye. Comparing to other screens Oled is best in size, price and output. LCD screen is small and draws rectangular shapes as output. The Thin-film-transistor LCD has a big size and high cost. Mira robot uses also Oled screens but different than we will use, these screens boasting acrylic domes are very thin and very expensive. On the other hand, PIA robot uses only two parts of plastic but they show no reaction.

2.6 Sounds, Movement and Lighting

Another way to express the reaction is by sounds, and speaker is best way to make the robot express its reaction. Therefore, we will use a phone speaker with amplification circuit. The speaker will output perfect reaction sounds (Example: laughing sound in happy reaction, crying sound in sad reaction…). In addition, we might add music or songs to make the robot more interactive with the surrounding environment (Example: night songs at night…).

To move the head as a part of expressing the reaction we will use MG90s metal servomotors and since we will use the same design of MIRA robot, the position of the motors will be same.

As for the lighting system, we will use RGB led strip that will be added in the body of the design. The lights will show the reaction of the robot according to the commands for the microcontroller (Example: Yellow for happy, Blue for sad…). On the other hand, MIRA and PIA robots used Neopixel RGB led strip but it is very expensive.

2.7 Deep learning:

One of our project goals is emotion recognition. Differentiating between 5 emotions (happy, sad, angry, neutral, surprised) is easy for humans, but for machines it's quite hard to accomplish, and it can be based on complex lines of code. But that will make the robot "dumb" and not efficient. Thus, we came to a solution of using deep learning for emotion recognition. Deep learning is a division of machine learning. What is machine learning? Machine learning is a branch of A.I that allows the machine to learn and gain experience by itself without being hard-coded. It is a computer program that allows a machine to access data and learn from it [13]. What is deep learning? Deep learning is a branch of machine learning that trains a system using neural networks by using a lot of data to make its own decisions. [14]. What are neural networks? A neural network is a collection of connected layers like a human neural network that takes an input, manipulates it in some ways and produces an output [14]. So deep learning is a branch of machine learning that uses a huge amount of data to learn from using neural networks. In our case, this huge data is pictures of human emotions. There are a lot of workframes in deep learning for emotion recognition, using Tensorflow, Tensorflow lite, Yolov3, TensorRT etc... In addition to that, there are a lot of architectures to use in deep learning for emotion recognition like Detectnet, Haarcascade, Darknet etc... In other robots like PIA and MIRA robots they used Haarcascade and OpenCV, but OpenCV is not for deep learning; OpenCV is a library for computer vision, and these robots were only detecting faces. In our case, we are recognizing emotions in faces which requires more than OpenCV, so we went with TensorRT as a Workframe and Detectnet as an architecture which is developed by NVIDIA, and we end up with this choice because of the microprocessor we are using in our robot.

2.8 Microprocessor:

To enable emotion recognition in our project and deploy a deep learning model, we need a microprocessor to handle computer vision and deep learning model outputs. Deep learning and computer vision require a GPU to run smoothly and in real time. The topic now is to pick something suitable for our case. Other robots like MIRA and PIA robots used raspberry pi 3, taking into consideration that the robots can only detect faces and run computer vision only, which is the case where a raspberry pi can be used, but raspberry pi doesn’t have a GPU so in our case it can’t be used since running deep learning and computer vision only on CPU will slow down the response a lot and might crash and overheat, so an alternative is to add a google coral USB accelerator TPU to a raspberry pi. The coral USB accelerator is an external add-on that can be connected via USB which adds a TPU (tensor processing unit) coprocessor to enable high machine and deep learning for a wide range of systems [15]**.** This alternative can work with our robot, but since we have a price constraint, the raspberry pi 4 8GB costs 75$ [16], and Google coral USB accelerator costs 75$ [15], thus a total of 150$ is expensive for our price constraint. A second alternative can be used which is NVIDIA Jetson nano. Jetson nano is one of the Jetson family microprocessors developed by NVIDIA for A.I projects [17]. Jetson nano comes with an 8 GB RAM Quad-core ARM A57 @ 1.43 GHz, 128-core NVIDIA Maxwell™ GPU, which covers our needs for deep learning and image processing. Moreover, Jetson Nano comes with a customized library for training deep learning models using TensorRT and DetectNet. Thus, we went with Jetson Nano as a microprocessor.

Figure 10: Nvidia's Jetson Nano

2.9 Camera:

Computer vision is done by taking live images of the user's face from a camera and processing them to come up with a decision on the user's current emotion. Thus, the robot should have a camera mounted on it to deal with image capturing. We had 2 options, either choosing a USB camera or a CSI camera. USB cameras are known for having a big size, and we have a size constraint in our project. There are small USB cameras but they are expensive, so USB cameras can't be used, as for CSI cameras, we had to choose between raspberry pi camera V1 and V2, and since we are using Jetson Nano as a microprocessor we had to choose raspberry pi camera V2 since raspberry pi camera V1 is not compatible with it, and raspberry pi camera V2 is used in both PIA and MIRA robots. Also, it's not a bad camera; it has 8MP, 1080p30, 720p60 and 640 × 480p60/90 video mode, and 3280 × 2464 pixels resolution [18].

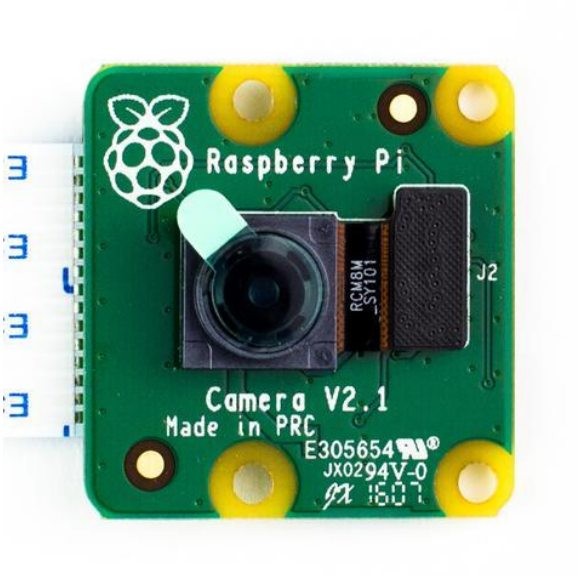


Figure 11: Raspberry Pi Camera V2

Other components used were DfPlayerMini for the sounds and a 32 GB SD Card.

Chapter 3

Design Approach and Implementation

3.1 Mechanism design

The design of the mechanism is based on the mechanism implemented in the Pia robot. However, a lot of add-ons and tweaks were required. We made the base similar in size to the lower base, which needs to support the boards as well as the mechanism on top.

The mechanism is comprised of 2 arcs that mount such that their axes are perpendicular, thus allowing the head to tilt forward and backward, as well as left and right. Now the head of the robot can mimic the neck of human's motion, where the whole head can rock in all direction. In the figure below, each arrow corresponds to how each part moves the mechanism, coded by similar colors.

In addition, we needed the head to rotate to look around while following the face, which made us mount the head on an extra servo motor that rests on top of the mechanism, thus allowing the head to perform a 180 degrees rotation. Below is the preliminary design of the mechanism. This has been changed after the first attempt to 3D print it, because we had some problems with dimensions and tolerances. Section three of this chapter elaborates more details on the issue.

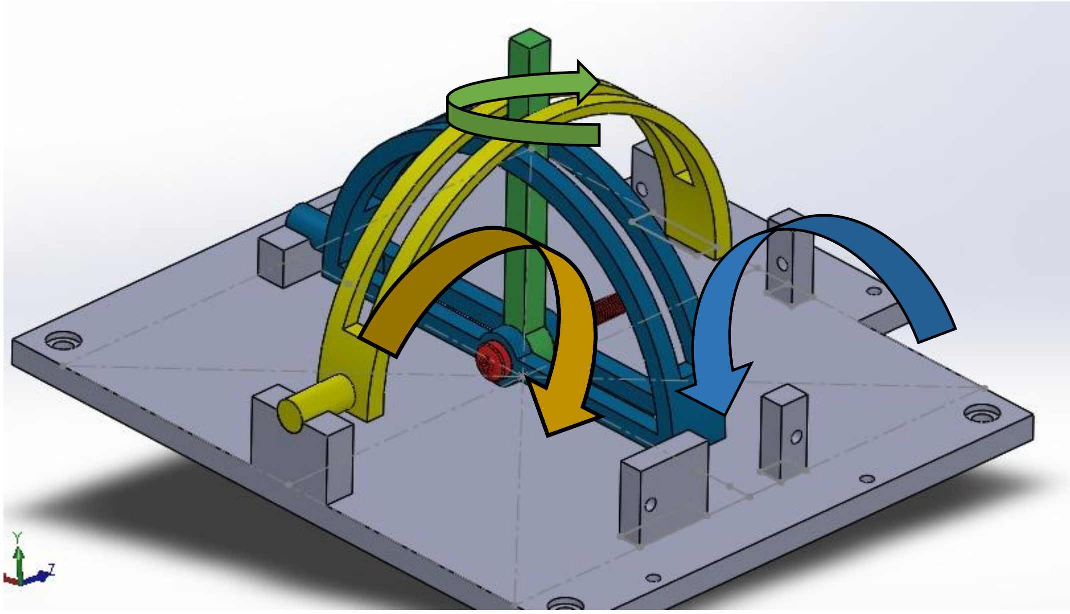


Figure 12: Preliminary Design

3.2 Body design

The body design was initially planned to be egg-shaped, and if time permitted, we might be able to be able to add more features to our robot. However, time was not in our favor and we were left with what we had initially planned for the body.

The eyes being OLED screens, we needed to flatten the place where these mount, so we changed those to fit the screens perfectly, keeping a flat space between the eyes that will accommodate the camera module, and looks like the nose.

The reason we had to switch these out was to fix a few issues that came out while designing the head. First of all, we planned on making the camera module hidden within the head. If the camera is placed in the upright position, the curvature of the head would make the scope seen by the camera smaller, and would restrict the view for the camera. If it is placed parallel to the curvature of the head, the camera would point up, while the eyes are directed straightforward. This would result in the camera following with the eyes not looking at the face the robot is tracking, which does not work for us.

Regarding the body, we started by creating a cylinder with bulgy sides, but that would lead to a bigger head for the robot. The resulting dimension would not fit in the printer we have at hand, and we therefore decided to change this design. We settled on a semi-sphere body with a flattened bottom. This would help us in determining the outer diameter with ease, and based on the widest dimension we can reach, for the head, and then base the body upon it.

Below is a figure of the final design before the first printing attempt, rendered in Blender (Blender is the free and open source 3D creation software for modeling, rigging, animation, simulation, rendering, compositing and motion tracking, video editing and 2D animation pipeline)

Figure 13: Blender Animation of the Robot

3.3 Printing

After finishing the design, we started with the printing of the mechanism. The first print let us discover a problem with the main arc, the one represented in figure 25 in blue color. The bottom of it was making contact with the plate beneath it, and we had to review the design to add fillets to the bottom side of it, and the servo holders seemed a little weak and we had to add triangular supports in order for these not to bend. We also discovered that the length of the green shaft that holds the servo was not enough, but we managed to connect it with a 3 millimeter threaded rod, which helped handle the weight of the head better rather than having the weight on 3D printed parts, and improved the rigidity of the mechanism as a whole. Below are some pictures of the design flaws that were corrected for the second print.

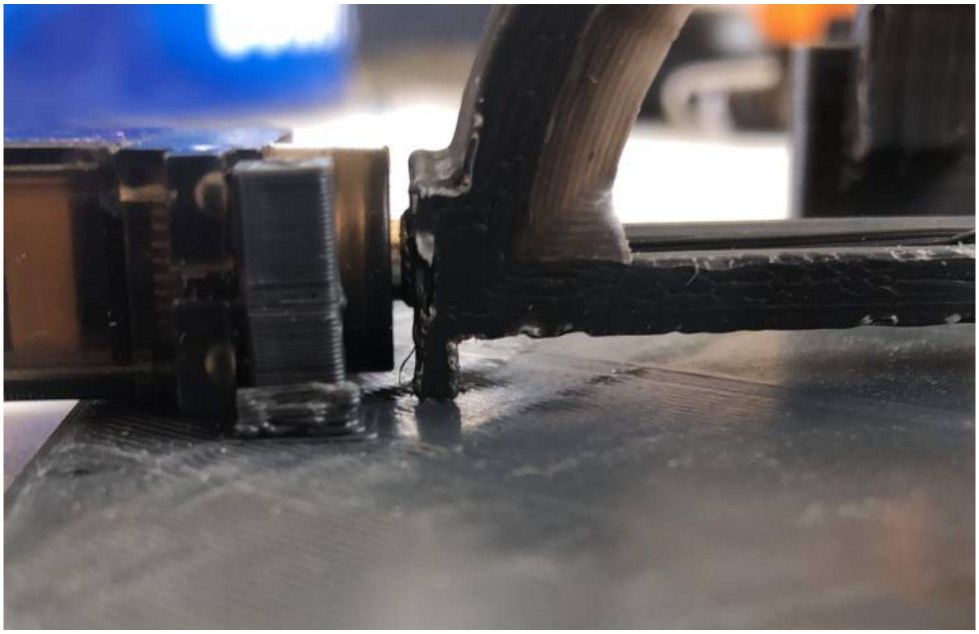


Figure 14: The Touching Area of the Arc and the Plate Beneath it

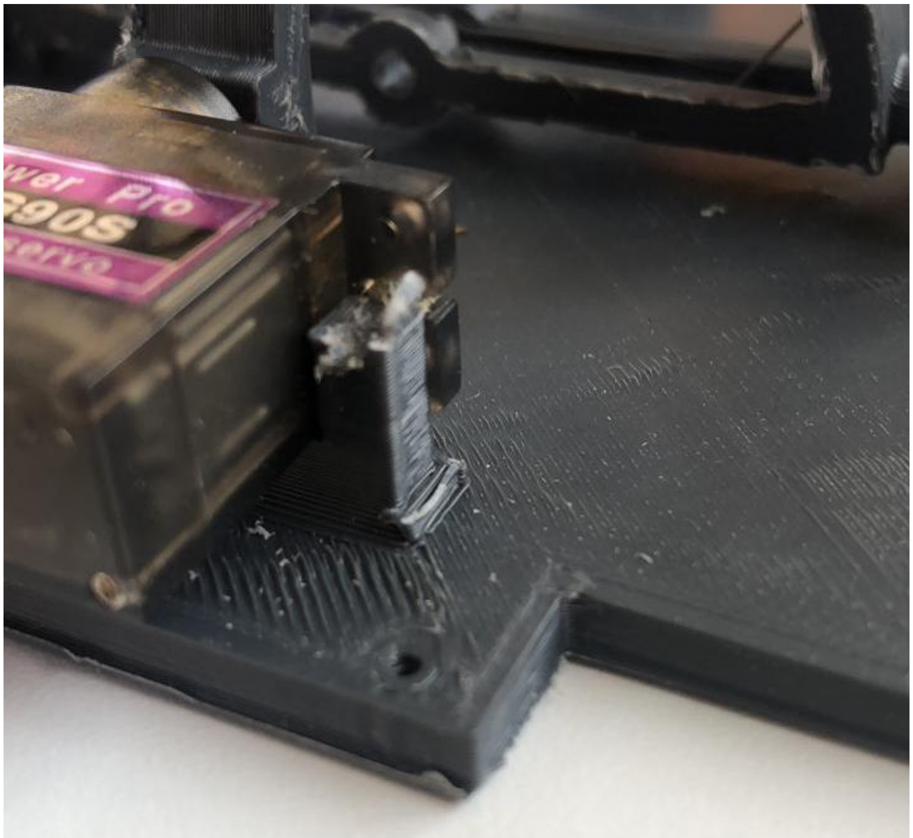


Figure 15: The Break of the Servo Mount Due to the Thin Material

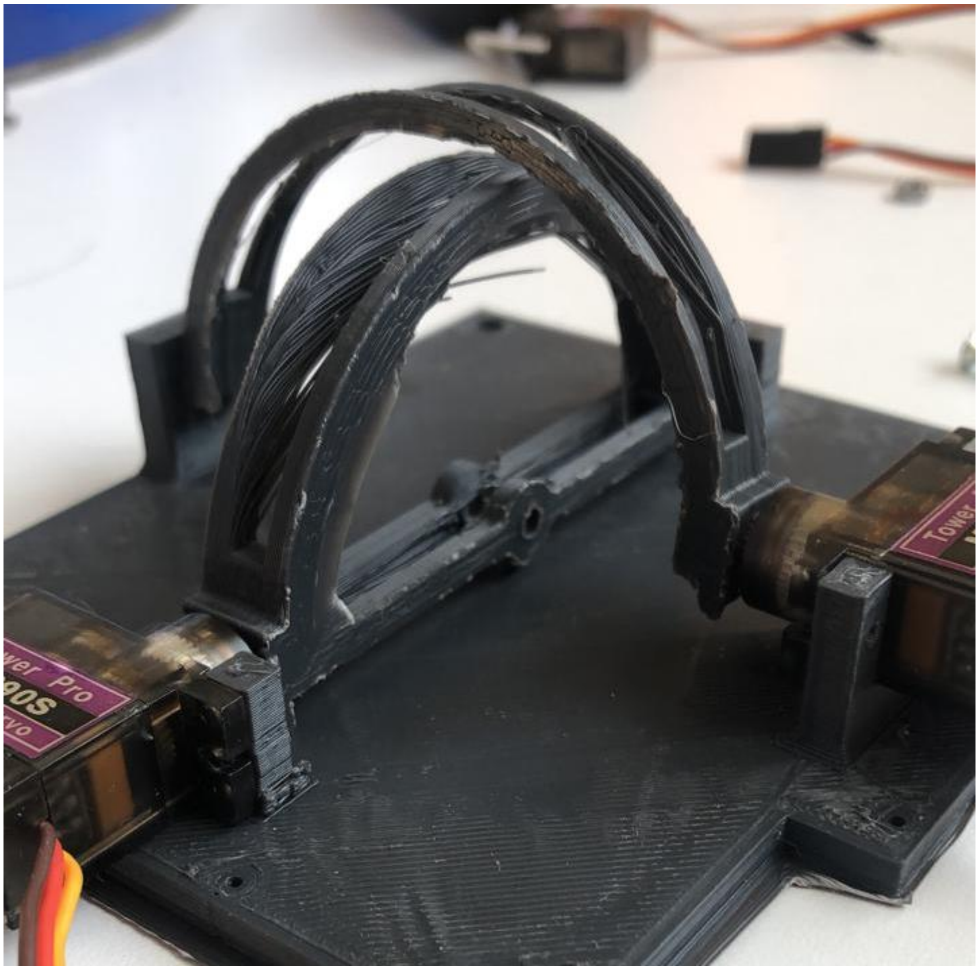


Figure 16: Excess Space between the 2 Arcs

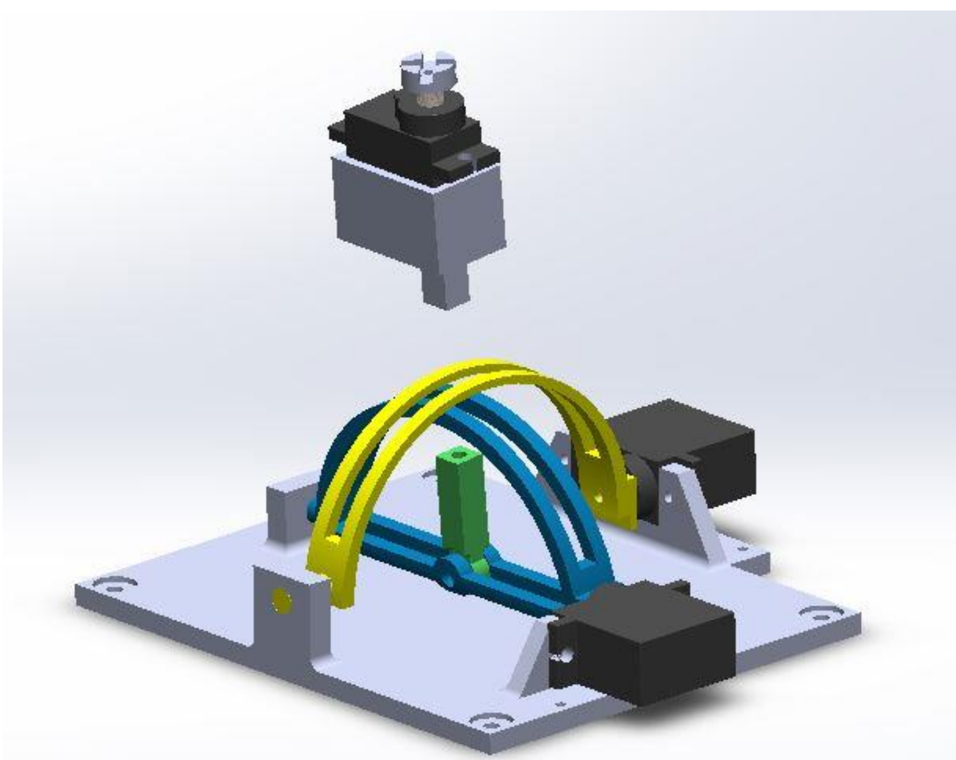


Figure 17: The Fixed Design Following the First Printing Trial

Chapter 4

Deep Learning Algorithm and Microprocessor

The 3 basic steps followed were:

-collecting the dataset and labeling it manually

-choosing the suitable microprocessor

-choosing the algorithm

One of the project goals is emotion recognition. Recognizing and differentiating between 5 emotions (happy, sad, angry, neutral, and surprised), which easy for humans but complex for machines to do. The only efficient path to take is to use a deep learning algorithm for emotion recognition.

4.1 Collecting Dataset

To train any supervised A.I system, a dataset is needed. In our case, the dataset is a group of images of people's facial expressions. For the A.I system to be robust and effective, the collected images should be diverse like collecting images of people with different nationalities, ages, gender, amount of expressed facial expression, and face position.

Figure 18: Sample Image from the Dataset Showing Diversity in Nationality, Gender, and Age

The dataset was manually collected using Google's search engine. It consists of 1000 RGB images with a .jpg or .jpeg format, some of the collected images are collage images containing multiple faces in one image which helped in increasing the number of facial expression and decreasing the image collection time. These 1000 images are divided into 5 emotions (happy, sad, angry, neutral, and surprised), each emotion have 200 images making 1000 images as total to be used for the A.I training.

4.2 Labeling Dataset

For the A.I system to understand the given images in the dataset, the images should be labeled, this means each image containing a facial expression should have a bounding box (a square) and labeled to the specified emotion, each image will have a file with the same name of the image, containing details about the bounding box the location of the face with respect to the image, the label which is one of the 5 emotions, knowing that the same image could have multiple bounding boxes and labels.

The labelling was done using a program named labelimg, the program helps by taking the images giving the user tools to create the bounding boxes and labels, and it automatically creates the necessary file that contains the labeling details.

There are 2 types of labeling methods, either YOLO or PASCAL (VOC) method, they both do the same functionality which is creating a file with details about the location, label etc. the difference between them are that YOLO creates .txt files containing first the label of the bounding box, location by writing the starting X, Y coordinates and ending X, Y coordinates of the bounding box. However, PASCAL (VOC) creates .xml file containing more details like folder name, filename, path to file, size of the image, label, and location of the bounding box.


```
<annotation>
	<folder>surprised</folder>
	<filename>Training_1009179.jpg</filename>
	<path>C:\Users\PC\Desktop\BE1 ppt-D\dataset-new\surprised\Training_1009179.jpg</path>
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>48</width>
		<height>48</height>
		<depth>1</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>surprised</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>7</xmin>
			<ymin>12</ymin>
			<xmax>42</xmax>
			<ymax>47</ymax>
		</bndbox>
	</object>
</annotation>
```


Figure 19: PASCAL (VOC) Annotations

```
0 0.904444 0.095556 0.057778 0.066667
0 0.702222 0.094444 0.062222 0.068889
0 0.503333 0.093333 0.060000 0.071111
0 0.297778 0.090000 0.057778 0.073333
0 0.096667 0.082222 0.060000 0.071111
0 0.095556 0.300000 0.062222 0.062222
0 0.297778 0.291111 0.053333 0.071111
0 0.497778 0.290000 0.057778 0.073333
0 0.706667 0.293333 0.057778 0.066667
0 0.901111 0.290000 0.060000 0.068889
0 0.900000 0.486667 0.057778 0.057778
0 0.698889 0.506667 0.064444 0.066667
0 0.494444 0.485556 0.060000 0.060000
0 0.304444 0.485556 0.062222 0.068889
0 0.097778 0.487778 0.066667 0.073333
0 0.097778 0.680000 0.057778 0.062222
0 0.297778 0.707778 0.057778 0.064444
0 0.494444 0.694444 0.051111 0.068889
0 0.700000 0.692222 0.057778 0.068889
0 0.893333 0.692222 0.057778 0.068889
0 0.904444 0.876667 0.057778 0.073333
0 0.698889 0.886667 0.064444 0.071111
0 0.508889 0.896667 0.071111 0.073333
0 0.302222 0.890000 0.066667 0.073333
0 0.091111 0.893333 0.053333 0.066667
```

Figure 20: YOLO Annotations

Choosing the annotation type depends on the algorithm used to train the A.I system, which will be discussed later.

4.3 Choosing Microprocessor and Algorithm:

When choosing an algorithm for the A.I system, we should take into consideration the available microprocessors in the market, since some algorithms can’t run on microprocessor due to their high need of a good GPU and CPU. In addition, when choosing a microprocessor to run the A.I system we should take into consideration the size constrain which is the microprocessor should fit in 20cm diameter circle and the price constrain. So, first we started with the cheapest and smaller size solution which is a raspberry pi4 microprocessor which can run openCV easily, the first issue we faced is that openCV doesn’t have good community support when it comes to training an A.I system using its Haarcascade algorithm, so to cut time and effort we didn’t use openCV as an algorithm. Our second solution was using YOLOv3 as an algorithm, since it has a lot of support from the community and is heavily used by a lot of users we tried it. YOLOv3 uses YOLO annotation method, so the labelling was done using YOLO method, the training was done using google collaborator, which is an online website that uses Jupyter notebook to allow users to write python scripts and Linux commands. In addition, it gives users a free high performance GPU which helps in decreasing the training time for the A.I system, rather than using slow performance PCs. So, after uploading the dataset and training the A.I system the final result had an A.I system with an accuracy of 97% and after testing a number of images and live detection it was good to be used, however it couldn't run on the raspberry pi4 since it doesn't have a GPU, so another solution for the microprocessor was using Nvidia Jetson nano, which is one of Nvidia's Jetson family, which also couldn't run on the Jetson nano, another solution was to use a higher performance microprocessor from the Jetson family but they are either bigger in size which violates our size constrain or expensive which also violates our price constrain. The last solution was using Nvidia's Detectnet algorithm which was tailored to work on all the Jetson family microprocessors including the Jetson nano. In addition to having online documentation and tutorial to use the algorithm to train an A.I system.

Detectnet uses PASCAL (VOC) annotation method, and we did the labelling before manually using YOLO method, so to reduce time, a python code was written and executed to transfer all the .txt files created to .xml files to be used in training. After transferring the file types, on the Jetson nano a GitHub repository created by Nvidia which is jetson inference, which helps in training and creating a model that can be deployed on the Jetson nano, after following the tutorial steps and training the A.I system for 100 epochs, the A.I model was created with an accuracy of 97%, after testing on live emotion detection it was ready to be deployed, the model had a detection speed of 40 to 50 fps which can be considered as real-time detection.

The real-time detection was done using a raspberry pi v2 camera which meets our size constrain for the camera, knowing that we could go with the cheaper solution which is the raspberry pi v1 camera but the sensor in the v1 camera is not compatible with the Jetson nano so we went with V2 camera.

Chapter 5

Reactions of the Robot

5.1 Arduino Reactions

Reacting to the 5 (happy, sad, angry, neutral, surprised) emotions is easy to human but to machines it is kind challenging. The reacting to the emotions needs high quality controller that communicate between the components used and the output of the Jetson Nano. The communication must be smooth and fast to get the best reaction on the robot. Therefore, we will used microcontrollers to accomplish the job. Microcontrollers are belonging to group of embedded systems, which are applicable for control of processes and product functions. This purpose needs to capture information about the product and about surround. These information's have to be processed inside the microcontroller and after processing, the microcontroller makes decision about next steps if it is necessary. Data capturing and processing is the main role of microcontroller**.**

Arduino Nano board is the best with respect to storage, performance, size and price. The microcontroller that we will choose is only responsible for communicating and controlling the components with respect to the input from the Jetson Nano. NodeMCU, Esp8266 and Esp32 microcontroller boards having additional features as WIFI and Bluetooth modules, but these features are useless in our case because all connections are wiring and not wireless. Teensy 4.0 microcontroller board is used by MIRA robot, but this board is a bit buffer and has a very high performance higher than what we need in the project. Moreover, Teensy 4.0 is a very high cost board. Arduino with flash storage is 32 KB and clock speed 16 MHz is the best to use in this project having the best size, price and very fast performance.

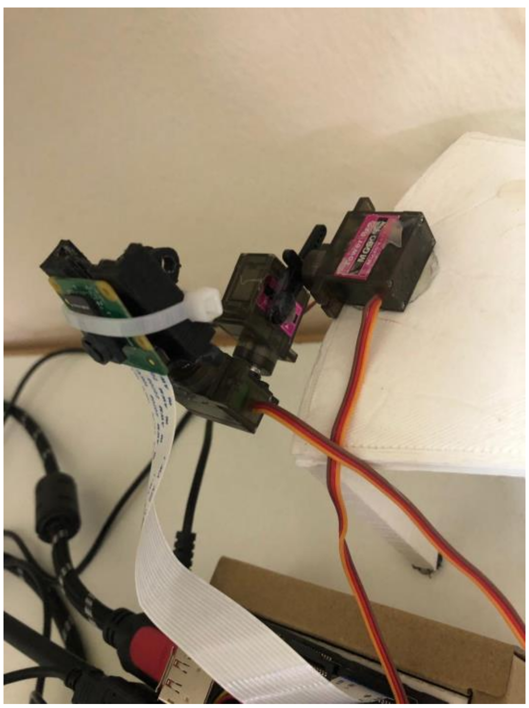
5.1.1 Face Tracking

After the A.I. detects a face, the robot's head should centers with the face and track it as it moves. Here comes the use of the 3 servo motors which moves the robot's head in 3DOF (X, Y, and Z axis). The A.I. system outputs 3 variables: index of the detected face (happy, sad. Angry, neutral, and surprised), percentage of confidence of the detected face, and the location of the face in the X, Y plane of the camera in pixels. The origin of the X, Y plane is in the bottom left corner of the camera image, to get the center of the image we divide the display width over 2 and the display height over 2 which we called the centerx and centery, so the center of the detected face should be in the X and Y plane should be equal to the centerx and centery. Moreover, we gave a range of error of 150 pixels for x and y divided left and right of the center location for both X and Y axis.

So, the robot first centers its head with the detected face first with respect to the x axis, if the value of the face center x value is less than the centerx the servo which makes rotation in the z axis will move to the left and vice versa. After centering in the x axis, the robot centers the head with the detected face in the y axis in the same way as in x axis.

Since the robots reaction is fast and the servo movement is also fast, the user will not notice the head is moving to center itself with the face in x axis then y axis, it will look like the head is moving in the x and y axis simultaneously.

The implementation of the face centering was done first with gluing the 3 servos and the camera together, making a look alike mechanism of the final design mechanism to test the face centering code.



After building the first experiment, we encountered an issue. The mechanism was getting stuck with the white base, so a small improvement was made, the whole mechanism was extended and placed on a wood plate to prevent it from getting stuck. And some counter weights were added to prevent the mechanism from falling. As shown in the figure below.

Figure 21: Showing the First Experiment to Test Face Centering Code

After testing the code on the experiment and making sure it works well. The code was then tested on the real mechanism with the head attached.

Figure 22: Showing the Improvements to the First Experiment

5.1.2 Eye Reactions

As for the eyes reaction, the best component to use is the OLED screen. Simply because OLED can read frames and print then in a very smooth and attracting way. Therefore, we can easily upload many frames and run them in a specific order to get the output needed. Therefore, eyes reaction will be very clear, interactive and smooth specially the blinking eye. Comparing to other screens OLED is best in size, price and output. LCD screen is small and draws rectangular shapes as output. The Thin-film-transistor LCD has a big size and high cost. Mira robot uses also OLED screens but different than we will use, these screens boasting acrylic domes are very thin and very expensive. On the other hand, PIA robot uses only two parts of plastic but they show no reaction.

Knowing that the OLEDs work fine on the Arduino, we faced a problem which is that the 2 OLEDs have the same i2c address, and the Arduino has only one i2c bus, and we need to display different animations on the OLEDs, so we can't be connected to the Arduino.

One solution for this problem is to use another OLED with a different i2c buss, but the available i2c OLEDs in the market all have the same i2c address. Another solution is to use an i2c multiplexer, but also it was not available in the market and we don't want to add more hardware.

The final solution that didn't require adding any hardware, was to attach the 2 OLEDs to the Jetson Nano which has a 2 i2c busses, and have a library for the used i2c OLEDs. Adding the OLEDs to Jetson Nano was not only a solution, it was helpful to make the robot faster and more responsive. Since the Jetson Nano has multiple cores, we can apply multiprocessing to run both OLEDs in parallel with the A.I. to detect faces and emotion, and at the same time display emotions on the OLEDs. In addition, in the Arduino we had to convert the images to bitmaps and use them as a matrix to display them, which took a lot of time to do and a lot of memory from the Arduino, and it slows the code. In the Jetson Nano, we had to use only a .PNG or .JPG image to display it on the OLED and it didn't affect the memory of the Jetson Nano since we are using a 32GB SD card and gives a higher resolution on the OLEDs which is better than the resolution on the OLEDs when they are connected to the Arduino.

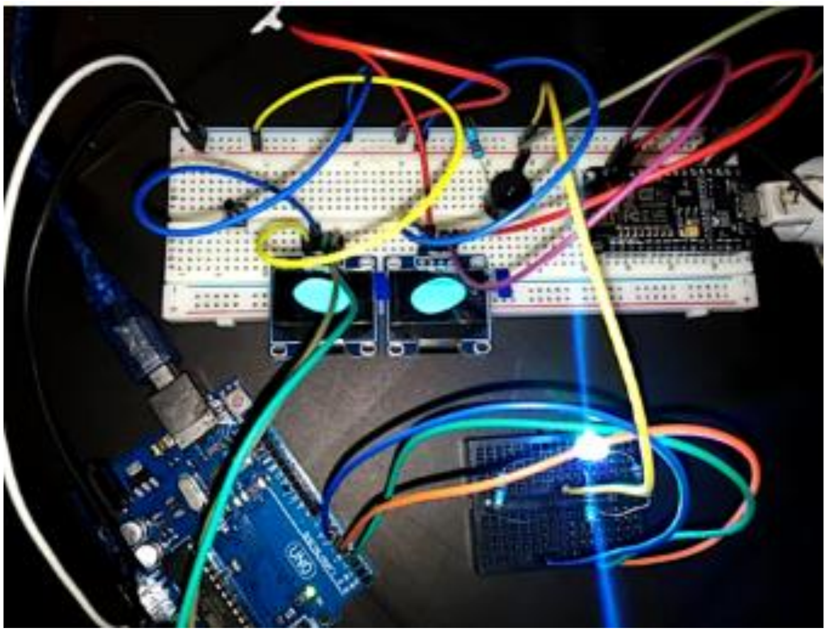


Figure 23: Neutral Eye Reaction 1st Trial with Arduino

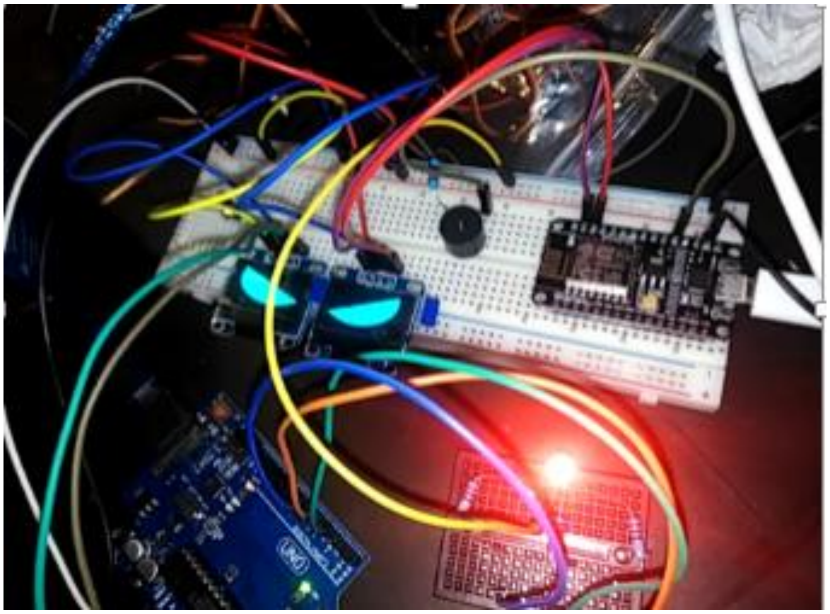


Figure 24: Angry Eye Reaction 1st trial with Arduino

5.1.3 Sounds

Another way to express the reaction is by sounds, we first used small Piezo buzzer to display small tones on it and then to make it more attractive we used small Nokia phone speakers, but the speakers' volumes were very low to hear then we used 2 0.5 W and 8 ohm speakers connected to a mp3 player. The speaker will output perfect reaction sounds (Example: laughing sound in happy reaction, crying sound in sad reaction…). In addition, we might add music or songs to make the robot more interactive with the surrounding environment (Example: night songs at night…).

5.1.4 Lights

As for the lighting system, we will use RGB led strip that is added in the body of the design. The lights will show the reaction of the robot according to the commands for the microcontroller (Example: Yellow for happy, Blue for sad…). On the other hand, MIRA and PIA robots used Neopixel RGB led strip but it is very expensive.

Finally for the robot’s head position we used a Mg90s servo motor metal gear and connected in the mechanism that will be shown in the robot deign to move the head to its position.

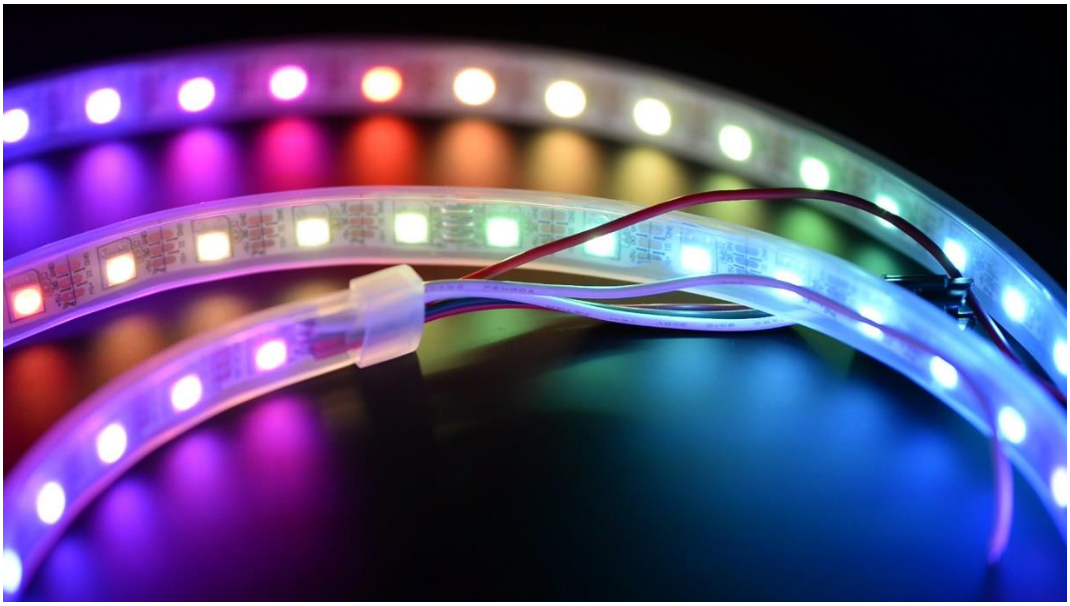


Figure 25: RGB LED Strip

5.2 Communication between Jetson Nano and Arduino

Communicating between the two boards is the most important step; Jetson Nano will give the emotion feedback to the Arduino to react on. We had many ways to complete the communication, but the faster and the easier way is by bits. Six pins are connected between the Jetson Nano, in this case it's the master, and the Arduino; in this case it's the slave. The Jetson Nano gives two outputs 1 and 0, the six pins are divided to two groups, first for the emotion feedback, and the second is for the head position. For example when the robot recognize a happy emotion it sends (0 0 1) to the Arduino to react, and another bits (0 1 0) to move the head.

5.2.1 Components connected to the Jetson Nano board:

There are only two components connected to the Jetson Nano board and they are: Raspberry Pi V2 RGB camera and two I2C OLEDS. To display the frames on the OLED screen you have to follow the following steps. First draw the needed frame or frames, second convert these frames from .PNG to bitmap images. The bitmap image is a matrix of black and white dots that is used for creating a perfect image, in our case we will convert the dot matrix into a 1x1024 array in order to use it in an Arduino code to display the frames. We also used small delays between multiple frames in order to form a GIF on the OLED. In that way OLEDs express the robot feelings by displaying eyes frames designed to look friendly. For example when the AI system recognize a happy emotion, the OLEDs display happy eyes. When the camera doesn't recognize a face, the OLEDs will display blinking eyes.

5.2.2 Components connected to the Arduino Board

The components programmed with Arduino are the RGB LED strip, three Servo motors MG90s and two speakers. Since the robot is especially for kids, expressing the feelings by colors is very important. Kids will learn how to express their feelings by choosing the best colors, for example red for angry… Using colors to learn and express emotions promotes self-regulation and reflection, planning and decision-making skills, mobility and fine-motor skills, focus, language development and more. Additionally, creating art can be used as a calming and grounding activity for many children [1].

MG90s Servo motors are used for the positioning of the head. The camera is placed in the middle of the head, so the head will move around to navigate in order to recognize a face. Each emotion has a unique face position. Each servo motor moves the face in a specific plane (X, Y or Z). When the robot recognize an emotion, the Jetson Nano board sends three bits (each pattern of the three bits is specified to an emotion) to Arduino, then Arduino gives order to the servo motors to move the head to the specified position.

Sounds are important to express feelings as well, so we used speakers as another way to express feelings. Recognizing emotions through sounds can help develop emotional intelligence. Sounds can help children to hear what certain feelings sound like, and they can learn to tell what emotion is that sound is about. Sounds can help a child to get in touch with and/or express a feeling he or she may be experiencing at the time; whether that may be happy, sad, surprised, or mad. The robot's sound can be a gate for their emotions and imitating them allows their emotions to come through. For example, when the robot makes a laughing sound, the kid will know that laughing is for happiness. Same as for crying sound, the child will know that this sound is for sadness.

Chapter 6

Results

As a result, our robot is now fully functional as originally desired. It tracks the face, detects it, recognizes the emotion of the person based on their facial features, and reacts accordingly with lights, sounds and eye expressions. Below are some pictures of the final prototype and its functioning.

Figure 26: Emotion Recognition Results

Figure 27: Sad Robot Reaction

Videos are available upon request.

Chapter 7

Conclusion and Recommendations

With creativity and knowledge skills, even complex systems that use AI can be made with inexpensive components, and because of the economic situation in Lebanon, we sought the best components for prices that aren't very high. However, some products are unique and cannot be replaced. For example, Jetson Nano, though expensive, was vital for our project.

This project has taught us skills additional to the skills gained from our courses, such as working with advanced and new technology like deep learning, new programming languages, turning ideas from theoretical to practical, and troubleshooting and problem solving, either in the mechanical and design, or in the electronics and programming subsystems.

The search for extra knowledge and skillsets is a never-ending path for an engineer. Technology is always updating, and so is a successful engineer.

Were it not for the lockdowns and unintended delays, many added features could have been added to this project to make it more interesting, advanced and attractive. More features and sophistication may be integrated in the future.

Of course, safety comes first, and that is why we abided by general mechanical and mechatronics standards and engineering ethics.

The team is not planning to stop with these results, in fact this is only prototype 1; more prototypes, experiments, features, and ideas will be done and implemented to improve the results and reach a goal where the robot should be a valid product and solution for the problem we are tackling, and not only an idea or a prototype. We plan to try this project with children, especially children with autism, and we hope it gives great results in terms of their social interaction and emotional well-being. After all, engineering's primary goal is to serve the community.

Finally, choosing the right components could prove difficult but is one of the most important steps in every project. Moreover, when designing a mechatronics system, the teammates should work in parallel, and not work on every subsystem alone.

Future improvements include:

- Making a more robust and accurate model for emotion recognition.

- adding all electronics hardware to one PCB

- adding speech recognition and commands

- advanced reactions to human emotions, not only mimicking the detected emotion

Our future goals:

- building more prototypes and implementing new ideas

- turning prototypes and ideas into a well-developed high end product that serves our end goals and targets

References

[1] https://www.frontiersin.org/articles/10.3389/frobt.2019.00107/full

[2] https://childmind.org/guide/guide-to-autism-spectrum-disorder/

[3] https://www.3dcontentcentral.com/secure/download-model.aspx?catalogid=10434&id=807822

[4] https://veerobot.com/store/ROB-MAA-MTSA-1011

[5] https://lastminuteengineers.com/joystick-interfacing-arduino-processing/

[6] https://www.thingiverse.com/thing:2929670

[7] https://datasheetspdf.com/pdf/1106582/ETC/MG90S/1

[8] https://shopee.com.my/MG90S-Micro-Metal-Gear-9g-Servo-for-RC-Plane-Helicopter-Boat-Car-4.8V-6V-i.133107381.2141487932

[9]http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

[10]http://actamechatronica.eu/issues/2017/III_2017_02_Virgala_Mikova_Kelemen_Liptak_Hroncova.pdf

[11] https://www.engineeringpassion.com/10-best-microcontroller-boards-for-engineers-and-geeks/

[12] https://www.makeuseof.com/tag/arduino-displays-options/

[13]: https://www.expert.ai/blog/machine-learning-definition/

[14]: https://towardsdatascience.com/deep-learning-with-tensorflow-part-1-b19ce7803428

[15]: https://coral.ai/products/accelerator/

[16]: https://www.raspberrypi.org/blog/8gb-raspberry-pi-4-on-sale-now-at-75/

[17]: https://www.nvidia.com/en-us/autonomous-machines/jetson-store/

[18]: https://www.raspberrypi.org/documentation/hardware/camera/

[19] https://www.circuitstoday.com/wp-content/uploads/2018/02/Arduino-Nano-Pinout.jpg

[20] https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano

[21]https://developer.nvidia.com/sites/default/files/akamai/embedded/images/jetsonNano/JetsonNano-DevKit_Front-Top_Right_trimmed.jpg

[22] https://www.newegg.com/insider/nvidia-jetson-nano-dev-kit-makers-overview-what-can-it-do/

[23] https://www.factoryforward.com/wp-content/uploads/2019/06/OLED-I2C-factoryforward-1-458x458.png

[24] https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/

[25] https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[28]https://cdn.shopify.com/s/files/1/0174/1800/products/Camera_Module_v2.1_2_of_3_large.JPG?v=1521201261

[29] https://www.raspberrypi.org/documentation/hardware/camera/

[30] https://www.electronicscomp.com/image/cache/catalog/towerpro-mg90s-servo-motor-800x800.jpg

[31] https://components101.com/motors/mg90s-metal-gear-servo-motor

[32] https://shop.westerndigital.com/content/dam/store/en-us/assets/products/memory-cards/ultra-uhs-i-microsd/gallery/ultra-uhs-i-microsd-32gb.png

[33] https://www.amazon.com/SanDisk-Ultra-microSDXC-Memory-Adapter/dp/B073JWXGNT?th=1

[34] https://nvidianews.nvidia.com/news/nvidia-announces-jetson-nano-99-tiny-yet-mighty-nvidia-cuda-x-ai-computer-that-runs-all-ai-models

[36] https://www.blender.org/

[37] https://www.actionforhealthykids.org/activity/feeling-through-colors/

Appendix A

Student Outcomes (KPI's)

	How was it addressed in your SLP?	***Where was it addressed in your SLP?***
1. An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics		
1.1 An ability to apply knowledge of mathematics	***Calculating the angle of freedom of each axis for the head***	***Chapter 3***
1.2 An ability to apply knowledge of Science	***Choosig PLA filament for the robot's structure because it is light weight and robust***	***Section 2.2 and Chapter 3***
1.3 An ability to apply knowledge of Engineering	***MECH: The design of the robot using Soldiworks, choosing the material and 3D printing it*** ***MECA: The use of microcontrollers and microprocessors in order to automate the robot and***	***Chapters 2, 3, 4, and 5***

	make it intelligent	
2. An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors		
2.1 Design a system/component of a system or a process to meet specific needs while respecting safety, health and welfare of the public and adhering to cultural, social, environmental and economic factors.	***Safety: The embedding of all the electrical components inside the robot in a way that they can be reached only for maintenance*** ***Economy: The materials chosen were of high quality and not very expensive*** ***Environmental: The robot doesn't harm the environment in any way*** ***Social: The robot's primary goal is to provide social and emotional support for children and especially autistic children***	***Sections 1.1 and 1.3***

2.2 Modify a system/component of a system or a process to meet specific needs while respecting safety, health and welfare of the public and adhering to cultural, social, environmental and economic factors.	***Safety: The first design was meant to be battery powered, but the battery presents certain safety risks if some safety measures weren't taken.*** ***Environment: Less waste*** ***Economy: Less expensive components were later used***	***Section 1.3 and Chapter 2***
3. An ability to communicate effectively with a range of audiences		
3.1 Ability to write a well-structured formal report/technical document that addresses an audience with diverse educational-background	***Throughout the university years, we were taught by our various professors on how to write and structure professional and clear reports that adhere to all audiences, as well as what sections to include and not to include.***	***Reports of various courses/ BE1 Report***

3.2 Ability to deliver a well-structured formal presentation that addresses an audience with diverse educational-background[1]	***We were also taught by our professors on how to design Power Point presentations in various fields and for all kinds of audiences, and how to present clearly and professionally.***	***Presentations in different courses/ BE1 Presentation/ Final BE Presentation***
4. An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts		
4.1 Identify global, economic, environmental, and societal impact of implementing engineering solutions using applicable engineering code of ethics to differentiate between ethical/unethical behaviors	***The Engineering code of ethics was generally used in implementing this project based on its principles and canons.***	***Section 1.3***
4.2 Identify global, economic, environmental, and societal impact of implementing engineering solutions using applicable engineering standards and codes to differentiate between professional/unprofessional behaviors	***ASME and IEEE standards for components were generally used in order to ensure maximum safety for the users as well as the least environmental harm.***	***Section 1.3 and Chapter 2***

5. An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives		
5.1 Ability to plan and organize team tasks collectively to meet established goals	***The team members were always in contact with one another almost at all times for the exchange of ideas and solutions and collaborative work on the project. Work was divided, but we helped each other with all the tasks, did a time plan etc…***	***Appendix B and Section 1.4***
5.2 Ability to carry out tasks assigned by the team to attain set objectives.	***Team members agreed among each other on how to divide the task and when to meet in order to finish certain tasks together. Time planning was a major pillar of our work.***	***Appendix B and Section 1.4***
6. An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions		
6.1 An ability to design experiments.	***Flowcharts, time plans, and hours of planning were***	***Sections 1.3 and 1.4***

	conducted before starting the implementatio n.	
6.2 An ability to conduct experiments.	***Ex: -OLED reaction FPS*** ***-Testing servo motor reaction speed in tracking the face*** ***-Testing real time emotion recognition speed*** ***-Testing different types of speakers that best fit the robot***	***Chapters 3, 4 and 5***
6.3 An ability to draw apt evidence-based conclusions by analyzing and interpreting data	***Ex: Testing the effectiveness of the AI algorithm in order to make the next improvements in the dataset, training time etc…***	***Sections 4.3 and 5.1***
7. An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.		

7.1 Identify necessary skills and tools of contemporary engineering practice to solve a problem at hand.	***Ex: -Arduino: Full running code without using any delays*** ***-Multiprocessing in Python*** ***-Improving the results of AI by analyzing the output***	***Appen dices D and E***
7.2 Apply self-learned skills and tools of contemporary engineering practice to solve a problem at hand.	***Ex: -We simulated the head mechanism to identify the degrees of freedom.*** ***-Knowledge in adobe illustrator to draw the robot's eyes*** ***-Advanced knowledge in Solidworks*** ***-Strong knowledge of Python Coding*** ***-Basic knowledge of Linux commands*** ***-Advanced 3D printing experience and knowledge***	***Figure s 12, 13, 16, 17, 19, 20, 21, 22 etc…***

Appendix B

Minutes of Meeting 1

Rafik Hariri University
جامعة رفيق الحريري

Higher Education as it

should be

MINUTES OF MECA 595A MEETING (1)

COLLEGE OF ENGINEERING – MME Department – RHU

ON October 7th, 2020 AT 5 PM

Present: Dr. Hassan Hariri (Advisor), Mohammad Al Kaderi, Karim Al Harake, Bassem Bashnak, Mohammad Daoud (Students)

Absent: NONE

This meeting came to order at 5:00 pm.

1. Updates

- We had worked a bit on the exterior design of the robot and decided on how the robot will move.

- We had also started working on emotion and face detection and the coding for the OLED screens.
- We came up with a preliminary plan for our work.

2. Advisor Comments and Recommendations

-

- Dr. Hariri explained that we need to include timing to our plan.
- Dr. Hariri pointed out that our preliminary design isn't very efficient as a body for the robot. And that it could only allow for rotation about 1 axis.
- Dr. Hariri explained to us that we need to include the constraints and the requirements of our project in our report.
- Dr. Hariri talked a bit about what to write in our report.

3. Expected Deliverables for Next Meeting

Dr. Hariri expected us to come up with a time plan, to better our design, and to start writing our report.

4. Assessment

Dr. Hariri was satisfied overall of how we are thinking, our planning and our work so far, and expected us to follow his recommendations in order to be on the right track.

The meeting was adjourned at 5:40 pm.

Minutes taken by: Karim Al Harake

Minutes of Meeting 2

Rafik Hariri University
جامعة رفيق الحريري

Higher Education as it

should be

MINUTES OF MECA 595A MEETING (2)

COLLEGE OF ENGINEERING – MME Department – RHU

ON November 11th, 2020 AT 5 pm

Present: Dr. Hassan Hariri (Advisor), Mohammad Daoud, Bassem Bashnak, Karim Al Harake, Mohammad Al Kaderi.

Absent: NONE

The meeting came to order at 5:00 pm.

5. Updates

- We showed Doctor Hassan the results of our emotion detection system using Jetson Nano with a video of Mohammad Daoud trying it on his face.
- We showed Doctor Hassan the state of the OLED display (The eye of the robot), the RGB strip blue color and the buzzer music when the robot is in the neutral mode.

6. Advisor Comments and Recommendations

- Doctor Hassan said that those results were very nice, and he encouraged us to keep making them even better.
- Doctor Hassan wanted us to start integrating our work.

- Doctor Hassan explained what to write in our literature review

7. Expected Deliverables for Next Meeting

- Training for emotion detection with a bigger dataset to improve accuracy etc…
- Finalizing communication between OLED, buzzer and RGB
- Integrating our work

8. Assessment

Our work was very good, but we needed to start working on our report.

The meeting was adjourned at 5:30 pm.

Minutes taken by: Karim Al Harake

Minutes of Meeting 3

Rafik Hariri University
جامعة رفيق الحريري

Higher Education as it

should be

MINUTES OF MECA 595A MEETING (3)

COLLEGE OF ENGINEERING – MME Department – RHU

Group I

ON November 14th, 2018 AT 11:30 AM

Present: Dr. Hassan Hariri (Advisor), Mohammad Daoud, Bassem Bashnak, Karim Al Harake, Mohammad Al Kaderi.

Absent: NONE

The meeting came to order at 5:00 pm.

9. *Updates*

- We had done a preliminary Gantt chart.
- We had done the abstract, some of the introduction, and the components and software used in the report.
- We told the doctor about some of the challenges we faced in finalizing the design of the mechanism.
- We explained how we collaborated in order to design the eyes on the OLED screen

- We showed the Doctor the eyes blinking with music.

10. Advisor Comments and Recommendations

- Doctor Hassan said that the Gantt chart could be improved by adding today’s date, and differentiating between what we have finished and what is yet to be done, but it is not a major issue. Some people do it the way we did it.
- Dr. Hassan said that we need to collaborate in writing the literature review. One person cannot do it alone.
- Dr. Hassan explained more about how we need to write our report, for example, what our motivation behind doing the project is, what the objective is, why we chose our components and not other ones, the difference between our work and similar work done by others etc…

11. Expected Deliverables for Next Meeting

- Improving our report because some of the things were unclear and some things can be added
- Writing our literature review in the proper way explained by the Doctor
- Collaborating to finalize our report

12. Assessment

Doctor Hassan is satisfied with our work. We just needed to improve our report and finalize the design.

The meeting was adjourned at 6:00 pm.

Minutes taken by: Karim Al Harake

Appendix C

Bill of Materials

Components	Quantity	Price
PLA Filament Spool	1	60$
Nvidia Jetson Nano	1	100$
Raspberry Pi Camera V2	1	25$
0.96" OLED Display	2	5$
RGB Led Strip	1	5$
Arduino Nano	1	10$
Servos	3	10$
32 GB SD Card	1	6$
Dfplayer Mini	1	15$
Speaker	1	1$
TOTAL	-	242$

Appendix D

Python Code

```
#!/usr/bin/python3

#

# Copyright (c) 2019, NVIDIA CORPORATION. All rights reserved.

#

# Permission is hereby granted, free of charge, to any person obtaining a

# copy of this software and associated documentation files (the "Software"),

# to deal in the Software without restriction, including without limitation

# the rights to use, copy, modify, merge, publish, distribute, sublicense,

# and/or sell copies of the Software, and to permit persons to whom the

# Software is furnished to do so, subject to the following conditions:

#

# The above copyright notice and this permission notice shall be included in

# all copies or substantial portions of the Software.

#

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR

# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,

# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT
SHALL

# THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER
```

```
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING
# FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER
# DEALINGS IN THE SOFTWARE.
#

import multiprocessing

import Jetson.GPIO as GPIO
import jetson.inference
import jetson.utils

import time
import sys

import Adafruit_SSD1306   # This is the driver chip for the Adafruit PiOLED

from PIL import Image
from PIL import ImageDraw
from PIL import ImageFont
import PIL.ImageOps

disp2 = Adafruit_SSD1306.SSD1306_128_32(rst=None, i2c_bus=0, gpio=1)
disp = Adafruit_SSD1306.SSD1306_128_32(rst=None, i2c_bus=1, gpio=1)
```

```
# Clear display.
disp.begin()
disp.display()
disp2.begin()
disp2.display()




net = jetson.inference.detectNet(argv=['--model=ssd-mobilenet.onnx', '--labels=labels.txt', '--
input-blob=input_0', '--output-cvg=scores', '--output-bbox=boxes', '--overlay=box,labels,conf'],
threshold=80)
camera = jetson.utils.videoSource("csi://0")      # '/dev/video0' for V4L2
display = jetson.utils.glDisplay() # jetson.utils.videoOutput("display://0") # 'my_video.mp4' for
file

global display_width
global display_height

display_width = jetson.utils.glDisplay().GetWidth()
display_height = jetson.utils.glDisplay().GetHeight()
print("display-width: " + str(display_width))
print("display-height: " + str(display_height))

global index
global width
```

```
global height

global centerx

global centery

global confidence

global flag

global x

global y

global finish




GPIO.setmode(GPIO.BOARD) #Jetson nano pin numbering


GPIO.setup(11, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(12, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(13, GPIO.OUT, initial=GPIO.LOW)


GPIO.setup(32, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(31, GPIO.OUT, initial=GPIO.LOW)

GPIO.setup(33, GPIO.OUT, initial=GPIO.LOW)


GPIO.output(32, GPIO.HIGH)

GPIO.output(31, GPIO.HIGH)

GPIO.output(33, GPIO.HIGH)


def blink(path):
```

```
        finish = False

        if path == "/home/daoud/Desktop/eyes-2/angry/angry-":
                Range1 = 4
                Range2 = 0
        else:
                Range1 = 11
                Range2 = 11

        image = (Image.open(str(path)+"0.png").resize((disp.width,disp.height),
Image.BICUBIC).convert("1"))
        im = (Image.open(str(path)+"0.png").resize((disp.width,disp.height),
Image.BICUBIC).convert("1"))
        disp.image(image)
        disp.display()
        disp2.image(im)
        disp2.display()

        for i in range(Range1):
                image=Image.open(str(path)+str(i)+".png").resize((disp.width,disp.height),
Image.BICUBIC).convert("1")
                im=Image.open(str(path)+str(i)+".png").resize((disp.width,disp.height),
Image.BICUBIC).convert("1")
                disp.image(im)
                disp.display()
                disp2.image(image)
```

```
                disp2.display()
                time.sleep(0.03)
        print("closing")
        for i in range(Range2-1, -1, -1):
                image=Image.open(str(path)+str(i)+".png").resize((disp.width,disp.height),
Image.BICUBIC).convert("1")
                im=Image.open(str(path)+str(i)+".png").resize((disp.width,disp.height),
Image.BICUBIC).convert("1")
                disp.image(im)
                disp.display()
                disp2.image(image)
                disp2.display()
                time.sleep(0.03)
        print("oppening")
        finish = True

path1 = "/home/daoud/Desktop/eyes-2/blink/blink-"
path2 = "/home/daoud/Desktop/eyes-2/angry/angry-"
tangry = multiprocessing.Process(target = blink, args =[path1])
thappy = multiprocessing.Process(target = blink, args =[path2])

index = 0
width = 0
height = 0
confidence=0
flag=0
```

```
flaghappy=0

flagangry=0

x=0

y=0

centerx = 0

centery = 0

finish = True


while display.IsOpen():

    img = camera.Capture()

    detections = net.Detect(img)

    display.RenderOnce(img)

    display.SetTitle("Object Detection | Network {:.0f} FPS".format(net.GetNetworkFPS()))


    flag = 0


    for detection in detections:

        index = detections[0].ClassID

        width = (detections[0].Width)

        height = (detections[0].Height)

        centerx = (detections[0].Center[0])

        centery = (detections[0].Center[1])

        confidence = (detections[0].Confidence)

        flag = 1
```

```
print("detection: ")
print(index)
print("confidence: ", confidence)

if (flag == 1):
    if (confidence >= 0.5):

        if (index == 1): #happy
            print("happy")
            GPIO.output(11, GPIO.LOW)
            GPIO.output(12, GPIO.LOW)
            GPIO.output(13, GPIO.HIGH)

            flagangry = 0

            thappy = multiprocessing.Process(target = blink, args =[path1])
            if flaghappy == 0:
                thappy.start()
                print("happy PROCESS")
                flaghappy = 1



        elif (index == 2): #sad
            print("sad")
            GPIO.output(11, GPIO.LOW)
            GPIO.output(12, GPIO.HIGH)
```

```
        GPIO.output(13, GPIO.LOW)
        flaghappy = 0
        flagangry = 0

    elif (index == 3): #angry
        print("angry")
        GPIO.output(11, GPIO.LOW)
        GPIO.output(12, GPIO.HIGH)
        GPIO.output(13, GPIO.HIGH)
        flaghappy = 0

        tangry = multiprocessing.Process(target = blink, args =[path2])
        if flagangry == 0:
            tangry.start()
            flagangry = 1

    elif (index == 4): #neutral
        print("neutral")
        GPIO.output(11, GPIO.HIGH)
        GPIO.output(12, GPIO.LOW)
        GPIO.output(13, GPIO.LOW)
        flaghappy = 0
        flagangry = 0

    elif (index == 5): #surprised
        print("surprised")
```

```
                GPIO.output(11, GPIO.HIGH)
                GPIO.output(12, GPIO.LOW)
                GPIO.output(13, GPIO.HIGH)
                flaghappy = 0
                flagangry = 0



        else :
            print("confidence less than 70")
            GPIO.output(11, GPIO.LOW)
            GPIO.output(12, GPIO.LOW)
            GPIO.output(13, GPIO.LOW)
            flaghappy = 0
            flagangry = 0



        x = display_width/2
        y = display_height/2

        #GPIO.output(35, GPIO.HIGH)

        if (centerx > x-75 and centerx < x+75):
            #center
            print("CENTERX")
            print(centerx)
            #GPIO.output(35, GPIO.LOW)
```

```
        #GPIO.output(32, GPIO.LOW)

        if(centery > y-75 and centery < y+75):
            print("CENTERY")
            print(centery)
            GPIO.output(32, GPIO.HIGH)
            GPIO.output(31, GPIO.LOW)
            GPIO.output(33, GPIO.HIGH)

        else:

            if (centery < y-75):
                #up
                print("UP")
                GPIO.output(32, GPIO.LOW)
                GPIO.output(31, GPIO.HIGH)
                GPIO.output(33, GPIO.HIGH)

            elif (centery > y+75):
                #down
                print("DOWN")
                GPIO.output(32, GPIO.HIGH)
                GPIO.output(31, GPIO.LOW)
                GPIO.output(33, GPIO.LOW)
    else:
        if (centerx < x-75):
```

```
                    #left
                    print("LEFT")
                    GPIO.output(32, GPIO.LOW)
                    GPIO.output(31, GPIO.LOW)
                    GPIO.output(33, GPIO.HIGH)

                elif (centerx > x+75):
                    #right
                    print("RIGHT")
                    GPIO.output(32, GPIO.LOW)
                    GPIO.output(31, GPIO.HIGH)
                    GPIO.output(33, GPIO.LOW)

        else:
            print("no face-free rom")
            GPIO.output(11, GPIO.LOW)
            GPIO.output(12, GPIO.LOW)
            GPIO.output(13, GPIO.LOW)

            GPIO.output(32, GPIO.LOW)
            GPIO.output(31, GPIO.LOW)
            GPIO.output(33, GPIO.LOW)

    print("going home")
    GPIO.output(32, GPIO.HIGH)
    GPIO.output(31, GPIO.HIGH)
```

```
GPIO.output(33, GPIO.HIGH)
```

Appendix E

Arduino Code

```
#include <Servo.h>
#include <Wire.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <DFRobotDFPlayerMini.h>

SoftwareSerial mySoftwareSerial(11, 10); // RX, TX
DFRobotDFPlayerMini myDFPlayer;

Servo xservo;
Servo yservo;
Servo zservo;

int R = 3;
int G = 6;
int B = 5;

int  a = 0;
int  b = 0;
int  c = 0;

int bit1 = 1;
int bit2 = 0;
```

```
int bit3 = 1;

int flaghappy = false;
int flagsad = false;
int flagangry = false;
int flagneutral = false;
int flagsurprised = false;

int flagcenter = false;

int x = 95; //(2): 50-90-140
int y = 110; //(1): 40-90-150
int z = 90;

int previousx = 0;
int previousy = 0;
int previousz = 0;

int newx = 0;
int newy = 0;
int newz = 0;

unsigned long previousMillis = 0;
unsigned long currentMillis = 0;
int intervalfree = 50;
int interval1 = 1000;
```

```
int interval2 = 2000;
int interval3 = 3000;


void setup() {
  // put your setup code here, to run once:
  xservo.attach(12);
  yservo.attach(8);
  zservo.attach(13);


  //a-b-c pins
  pinMode(2, INPUT);
  pinMode(4, INPUT);
  pinMode(7, INPUT);


  //bit1-bit2-bit3 pins
  pinMode(14, INPUT);
  pinMode(15, INPUT);
  pinMode(16, INPUT);


  xservo.write(x);
  yservo.write(y);
  zservo.write(z);


  newx = random(70, 130);
  newy = random(70, 130);
  newz = random(55, 105);
```

```
  pinMode(R, OUTPUT);// Red from RGB led
  pinMode(G, OUTPUT);// Green from RGB led
  pinMode(B, OUTPUT);// Blue from RGB led

  mySoftwareSerial.begin(9600);
  Serial.begin(115200);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if (!myDFPlayer.begin(mySoftwareSerial)) {  //Use softwareSerial to communicate with
mp3.
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while (true) {
    }
  }
  Serial.println(F("DFPlayer Mini online."));

  myDFPlayer.volume(25);  //Set volume value. From 0 to 30
  myDFPlayer.playFolder(16, 4); //neutral
  delay(2000);
```

```
  randomSeed(analogRead(A3));
}

void loop() {
  // put your main code here, to run repeatedly:
  currentMillis = millis();

  previousx = x;
  previousy = y;
  previousz = z;

  a = digitalRead(2);
  b = digitalRead(4);
  c = digitalRead(7);

  bit1 = digitalRead(15);
  bit2 = digitalRead(14);
  bit3 = digitalRead(16);
  Serial.print("bit1: ");
  Serial.print(bit1);
  Serial.print(" / ");
  Serial.print("bit2: ");
  Serial.print(bit2);
  Serial.print(" / ");
  Serial.print("bit3: ");
  Serial.print(bit3);
```

```
Serial.println("");

if (bit1 == 1 && bit2 == 0 && bit3 == 1) {
  Serial.println("Stop");
  flagcenter = true;
}
else if (bit1 == 0 && bit2 == 0 && bit3 == 1) {
  //left
  Serial.println("left");
  flagcenter = false;
  z++;
}
else if (bit1 == 0 && bit2 == 1 && bit3 == 0) {
  //right
  Serial.println("right");
  flagcenter = false;
  z--;
}
else if (bit1 == 0 && bit2 == 1 && bit3 == 1) {
  //up
  Serial.println("up");
  flagcenter = false;
  y++;
}
else if (bit1 == 1 && bit2 == 0 && bit3 == 0) {
  //down
```

```
  Serial.println("down");
  flagcenter = false;
  y--;
}

else if (bit1 == 0 && bit2 == 0 && bit3 == 0) {
  //free-roam
  Serial.println("free-roam");
  flagcenter = false;

  if (currentMillis - previousMillis >= intervalfree) {
    if (previousx != newx) {
      if (previousx < newx) {
        x++;
      }
      else if (previousx > newx) {
        x--;
      }
    }
    else {
      newx = random(70, 130);
    }

    if (previousy != newy) {
      if (previousy < newy) {
        y++;
```

```
  }
  else if (previousy > newy) {
    y--;
  }
}
else {
  newy = random(70, 130);
}

if (previousz != newz) {
  if (previousz < newz) {
    z++;
  }
  else if (previousz > newz) {
    z--;
  }
}
else {
  newz = random(55, 105);
}

previousx = x;
x = 95;
previousy = y;
previousz = z;
previousMillis = currentMillis;
```

```
  }
}
else if (bit1 == 1 && bit2 == 1 && bit3 == 1) {
  //home
  Serial.println("home");
  flagcenter = false;
  x = 95;
  y = 110;
  z = 90;
}

if (x >= 140) {
  x = 140;
}
else if (x <= 50) {
  x = 50;
}

if (y >= 150) {
  y = 150;
}
else if (y <= 40) {
  y = 40;
}
if (z >= 179) {
  z = 179;
```

```
}
else if (z <= 0) {
  z = 0;
}
Serial.print("X: ");
Serial.print(x);
Serial.print(" / ");
Serial.print("Y: ");
Serial.print(y);
Serial.print(" / ");
Serial.print("Z: ");
Serial.println(z);
delay(10);
xservo.write(x);
delay(10);
yservo.write(y);
delay(10);
zservo.write(z);
delay(10);

if (flagcenter == true) {
  if (a == 0 && b == 0 && c == 0) // Face detected but low confidence
  {
    flaghappy = false;
    flagangry = false;
    flagsad = false;
```

```
  flagsurprised = false;

  digitalWrite(R, 255);
  digitalWrite(G, 255);
  digitalWrite(B, 255);

  if (flagneutral == false) {
    if (currentMillis - previousMillis >= interval2) {
      myDFPlayer.playFolder(16, 1); //neutral
      previousMillis = currentMillis;
      flagneutral = true;
    }
  }
}

if (a == 0 && b == 0 && c == 1) //HAPPY FACE
{
  digitalWrite(R, 255);
  digitalWrite(G, 255);
  digitalWrite(B, 0);
  Serial.println("HAPPY");
  // YELLOW COLOR//
  flagangry = false;
  flagneutral = false;
  flagsad = false;
  flagsurprised = false;
```

```
  if (flaghappy == false) {
    if (currentMillis - previousMillis >= interval3) {
      myDFPlayer.playFolder(16, 3); //happy
      previousMillis = currentMillis;
      flaghappy = true;
    }
  }
}

if (a == 0 && b == 1 && c == 0) //SAD FACE
{
  // BLUE COLOR//
  digitalWrite(R, 0);
  digitalWrite(G, 0);
  digitalWrite(B, 255);
  Serial.println("SAD");
  flaghappy = false;
  flagangry = false;
  flagneutral = false;
  flagsurprised = false;

  if (flagsad == false) {
    if (currentMillis - previousMillis >= interval1) {
      myDFPlayer.playFolder(16, 5); //sad
      previousMillis = currentMillis;
```

```
      flagsad = true;
    }
  }
}

if (a == 0 && b == 1 && c == 1) //ANGRY FACE
{
  // RED COLOR//
  digitalWrite(R, 255);
  digitalWrite(G, 0);
  digitalWrite(B, 0);
  Serial.println("ANGRY");
  flaghappy = false;
  flagneutral = false;
  flagsad = false;
  flagsurprised = false;

  if (flagangry == false) {
    if (currentMillis - previousMillis >= interval1) {
      myDFPlayer.playFolder(16, 2); //angry
      previousMillis = currentMillis;
      flagangry = true;
    }
  }
}
```

```
if (a == 1 && b == 0 && c == 0) //NEUTRAL FACE
{
  // WHITE COLOR//
  digitalWrite(R, 255);
  digitalWrite(G, 255);
  digitalWrite(B, 255);
  Serial.println("NEUTRAL");
  flaghappy = false;
  flagangry = false;
  flagsad = false;
  flagsurprised = false;

}

if (a == 1 && b == 0 && c == 1) //SURPRISED FACE
{
  // YELLOW COLOR//
  digitalWrite(R, 255);
  digitalWrite(G, 0);
  digitalWrite(B, 255);
  Serial.println("SUPRISED");
  flaghappy = false;
  flagangry = false;
  flagneutral = false;
  flagsad = false;
```

```
      if (flagsurprised == false) {
        if (currentMillis - previousMillis >= interval1) {
          myDFPlayer.playFolder(16, 6); //surprised
          previousMillis = currentMillis;
          flagsurprised = true;
        }
      }
    }
  }
}
```