# CHAPTER 13

■ ■ ■

# Creating a Realistic Robotic Face

Expressing emotions through our facial expressions is a natural task for humans, but it turns out it is very difficult to emulate on a robot.

The human face expresses emotions using a series of muscles controlled by the facial nerve that comes from the cerebral cortex and connects to several muscles that split in front of the ears. A simple smile uses 17 muscles and more than 40 are used to frown.

At the Science University of Tokyo, professor Fumio Hara, PhD, demonstrated in the documentary *Most Advanced Humanoid Robots: Rise of Machine,* a robot face representing two emotions—surprise and sadness—and received $3 million in funding to improve human acceptance when dealing with robots.

If each muscle represents a certain DOF (degrees of freedom)—a concept explained in Chapter 11—then to represent a facial expression in a robotic face, several servos motors would be required. It is estimated that the human face has around 59 DOF.

Robotic faces can be manipulated by simple DC motors when precision is not required. In 2001, the company named The Original San Francisco Toymakers Ltd. created a toy called Yano Storyteller, which is a little robot in form of a pal that tells stories. Kids can interact with Yano by choosing different ends but Yano's best features are his facial expressions, which are changed by six DC motors in contact with a latex mask. The product was discontinued. I believe the reason were the Internet posts saying that kids were scared of Yano's expressions, which were in fact are very expressive. If you have opportunity to find an Yano on eBay or any other site, you can integrate him using Intel Galileo. You'll need to make some changes to the methods presented in the paper "Robotic F.A.C.E.: Facial Alerting in a Communication Environment" from MIT. It can be found at http://web.media.mit.edu/~stefanm/yano/.

The idea in this chapter is to create a robotic face that can express three emotional expressions—happiness, sadness, and seriousness—with lowest budget possible and using regular tools instead of a laser cutter or a 3D printing.

In other to create this robotic face, some animatronic techniques will be discussed and some work with wood, wires, and servo motors will be required.

1

# Project Details

You can build the robotic face for this project using rice, Styrofoam balls, balsa wood, glue gun, nails, music wire, and some screws.

The face is divided in two parts; the animatronic eyes and the mouth. Each part is controlled by two mini servos and can be built separately. This means that if you are interested only in the animatronic eyes, not the mouth or vice versa, you can build each part individually.

The cost to build the robotic face is around $30.

If you have a laser cutter, the DXF (Drawing Interchange Format) files are also provided for the mouth and to the animatronic eyes. The robotic face has a very simple design composed specially for orthogonal lines to make it easier to cut the balsa wood manually, so do not expect a complex design like the T-800 robot from the *Terminator* movie series.

The robotic face is controlled based on the servo control board developed in Chapter 11, in the section called "**Building a Servos Control Board**"; however, only one thumbstick is used instead of the three present on this board. That means you can build a simplified version of the board, eliminating the other two thumbsticks. More details will be discussed later.

The next section discusses the materials list and tools needed to build the animatronic eyes and mouth.

# Materials List

As explained in the previous section, the animatronic eye and mouth can be built separately, so two different materials lists are presented.

The animatronic eyes require the materials listed in Table 13-1.

*Table 13-1.* *Required Materials for the Animatronic Eyes*

| Quantity | Components |
| --- | --- |
| 1 | Music wire package with four wires; code 5497 0.039 K&S |
| 1 | 1.5 inch Styrofoam ball |
| 2 | Balsa wood sheet, 3.93 x 6.5 x 0.11 inches (10 x 16.5 x 0.3 cm) |
| 2 | Servo motors SG-90 with accessories |
| 2 | Sets of universal joints (U-joints) Mini V, Mini C ProBoat |
| 4 | Screw eyes 3 x 12 mm |
| 6 | Nails 1-1/4'' (3.18 cm) |

2

The mouth requires the materials listed in Table 13-2.

*Table 13-2.* *Required Materials for the Mouth*

| Quantity | Components |
| --- | --- |
| 2 | Balsa wood sheet, 3.93 x 6.5 x 0.11 inches (10 x 16.5 x 0.3 cm) |
| 2 | Cuts of balsa wood, 3.93 x 2.55 x 0.2 inches (10 x 2.5 x 0.5 cm) |
| 1 | *Cut of balsa wood, 3.7 x 0.47 x 0.47 inches (9.4 x 1.2 x 1.2 cm) |
| 2 | Servo motors SG-90 with accessories |
| 1 | 3.14 inches (8 cm) Natural Tubular Latex Rubber Band 6 x 9 mm |

*(\*) Only if you mounted the animatronic eyes*

The tools required for this project are listed in Table 13-3.

*Table 13-3.* *Required Tools*

| Quantity | Components |
| --- | --- |
| 1 | Wire cutter (preferable 8 1/4") |
| 1 | 20g of rice or similar granular material |
| 1 | Hot glue gun |
| 1 | Needle nose plier |
| 1 | Regular plier |
| 1 | Cut knife number 30 (recommended part number 16030) |
| 1 | Mini screwdriver for servo accessories screws |
| 1 | Inkjet or laser printer |
| 1 | Wood ruler with metal edge |

If you do not have balsa wood sheets in the dimensions specified by the materials lists, I recommend getting the package sold by Hobby Lobby named "Balsa Wood Assortment - Random Sizes and Shapes," part number #696039. It costs around $9 but comes with all shapes needed for this project. With the assortments provided in this package, you need to cut the balsa wood using the cutter knife. Check this link for more details: http://shop.hobbylobby.com/products/balsa-wood-economy-bag-696039/.

You can get a package with four music wires for around $1 and I recommend using the K & S Engineering sku# 602201. Check this link: http://shop.hobbylobby.com/products/-039-music-wire-602201/. These music wires will be used to build the rods that connect the servo to the eyes. They are a good alternative to the rods used with RC models.

The screw eyes can also be found at Hobby Lobby. Part number 14636 costs $1.99. Check this link: http://shop.hobbylobby.com/products/12mm-screw-eyes-146365/. These screw eyes are used to connect the eyes to the custom rods made with music wires and are an affordable solution to the ball links and sockets normally used in RC models.

The servo motors SG-90 are mini servos. They can be found on Amazon and eBay for less than $15.

The rice fills the tubular latex band so the mouth doesn't fold on its edges during its manipulation. This tubular latex is the same used in the slingshots and can be found online or in pharmacies. Alternatively you can use a piece of house, but it must be very flexible to create a good effect when the mouth moves. If you decide to use a piece of house, I recommend using a micro fuel line 1/4'' OD and 1/8'' ID.

Regarding the tools, the cut knife can be found for around $2.50.

The nails are used in order to fix the eyes with universal joints. The nails are a cheap alternative to screws that might increase the cost of this project. Due to the proportion of the accessories and the holes in the universal joins, you would need "mini machine screws," which are not cheap. You will adjust the size of the nails by cutting them and fixing them with the help of a hot glue gun.

The inkjet or laser printer is used to print the drawing and use it as template to make the cuts in the balsa wood. The next sections explain how to build the animatronic eyes and mouth.

---

■ **Warning**   The tools used on this chapter are not toys and misusage can inflict serious injuries. It is your responsibility to manage all tools and materials listed in this chapter. If you are underage, be sure to get help from an adult.

---

This book provides templates for cutting the balsa wood in three formats: **DXF (Drawing Exchange Format)**, **DWG (DraWinG)**, and **PDF**. You can find these in the plans folder of this chapter. The DXF and DWG files must be opened using AutoCAD software, while the PDF uses Acrobat. The idea around providing AutoCAD software is to allow you to use the files in case you have a laser cutter, while PDF is better if you are mounting the robot face manually.

If you are interested in opening the files using AutoCAD, I recommend installing **DraftSight AutoCAD**, because it is free for hobbyists, easy to use, and runs in Windows, Linux, and Mac OSX. You can download this program from http://www.3ds.com/products-services/draftsight/overview/.

The Acrobat reader used for PDF files can be downloaded from http://get.adobe.com/reader/.

The required software is listed in Table 13-4.

*Table 13-4.* *Required Software*

| Quantity | Components |
|----------|-----------|
| 1 | Acrobat Reader (PDF format). |
| 1 | Any AutoCAD (DXF and DGW), although DraftSight AutoCAD is recommended. |

The next section discusses how to assemble the animatronic eyes and mouth.

# Mounting the Robot's Face

The process for mounting the robot face is split in two steps: mounting the animatronic eyes and mounting the mouth. Both are independent and you can mount only the animatronic eyes, only the mouth, or both to create the full robot face. The next sections explain how to mount each individually.

## Mounting the Animatronic Eyes

To mount the animatronic eyes, you need patience and care. You will be using a knife that could seriously hurt you. One of the reasons it's best to use a metal ruler is that it protects your fingers from any slips of the knife. Always use the metal ruler side while using the knife.

Figure 13-1 shows the wood cuts, marked as BASE, EYE SUPPORT, and BASE FOR SERVO ACCESSORIES. This figure also shows the universal joins and the eyes.
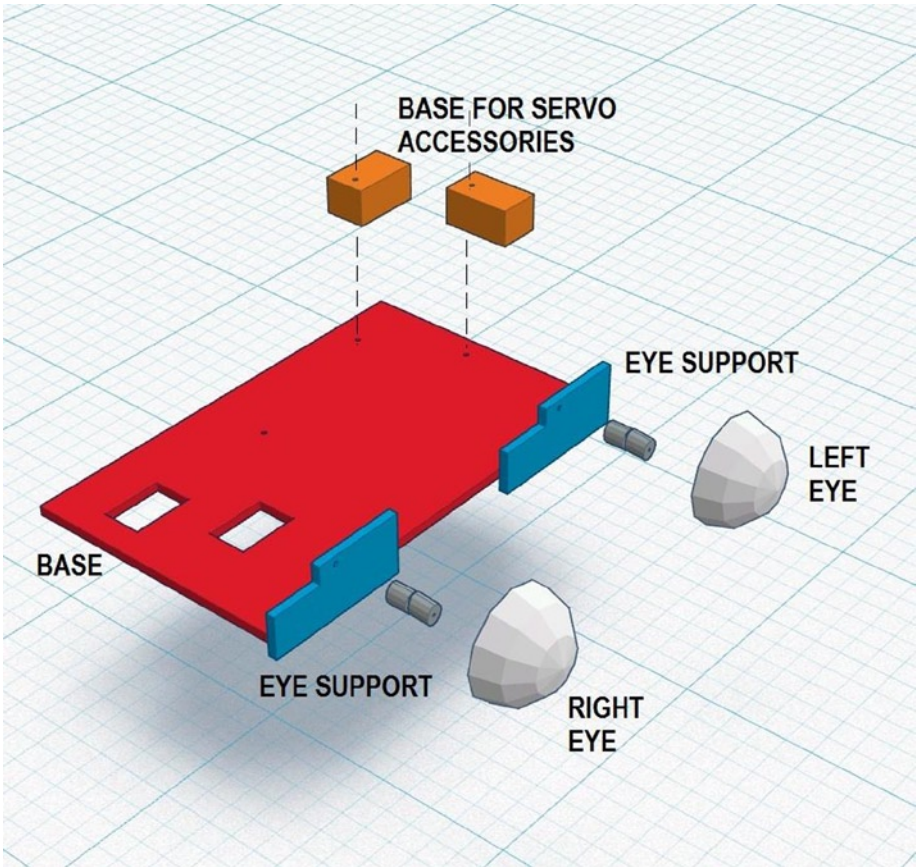
*Figure 13-1.*  *The animatronic eyes assembly*

The following steps are required to mount the animatronic eyes.

## Step 1: Preparing the Styrofoam Eyes

The eyes will have screw eyes attached to its structure and these screw eyes are connected to rods moved by the servo motors. The music wires are used to create the rods but if you check the thickness of the music wires and compare that with the diameter of the screw eye's holes, there is a certain looseness that might affect the precision of the eye movement. In order to eliminate this problem, insert a little bit of hot glue in the screw eyes' hole, forming a little ball. When the glue is almost dry, use the wire to create a small hole in the glue to reduce the looseness.

This process must be done four times, because each eye needs two applications, one for horizontal movement and another one for vertical.

Hold a screw eye using pliers and apply a small amount of hot glue, thereby forming a little ball on its hole. Wait about 10 seconds and then insert the music wire in the middle of the glue ball you created, as shown in Figure 13-2.
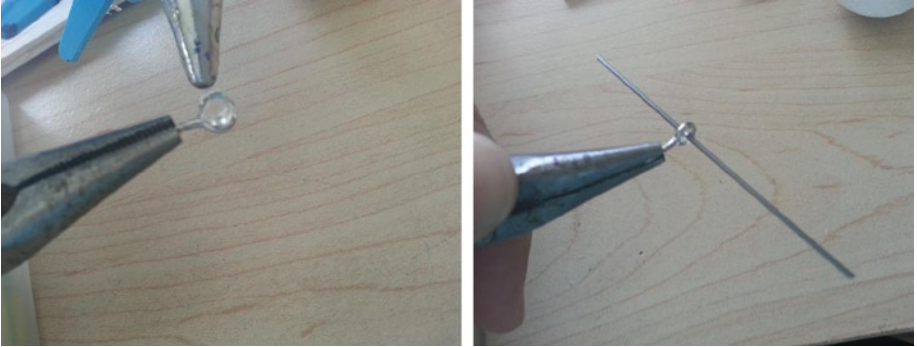


***Figure 13-2.*** *Preparing the screw eyes*

Cut the Styrofoam balls in two halves and screw the screws on each half, forming 90 degrees in the horizontal position, as shown in Figure 13-3. Do not insert the screw eyes by pushing; instead, screw them in. This will cause them to be tighten around the Styrofoam, preserving the material.
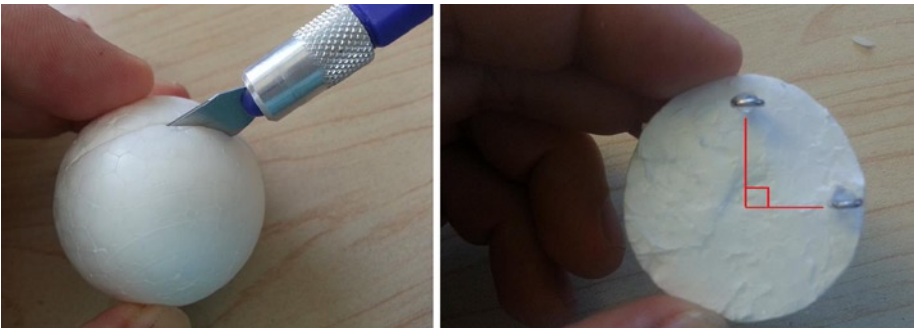


***Figure 13-3.*** *Balls cut in halves (left) with screw eyes introduced (right)*

Then, using the wire cutter, cut two nails in 0.86 inches (2.5 cm) and insert the nails completely in the central concave part (the iris position), as shown in Figure 13-4, left. In the remaining nail part of the flat area of the ball, apply a little bit of glue, as shown in the bubble numbered "1" in Figure 13-4. Insert the universal joint, but make sure the "eye" will be able to move horizontally and the universal joint is perpendicular to the flat area of the eye.

**Figure 13-4.** *Introducing the nails and the universal joint*

Repeat this procedure with the other half of the Styrofoam ball. The next section explains how to assemble the base.

## Step 2: Mounting the Animatronic Eye Frame

Open the file named base_eyes located in the code/plans folder using your preferable extension and then print the drawing using a regular laser or inkjet printer.

This drawing contains the base, two supports for the universal joins, and two supports for the servo accessories, as shown in Figure 13-5.
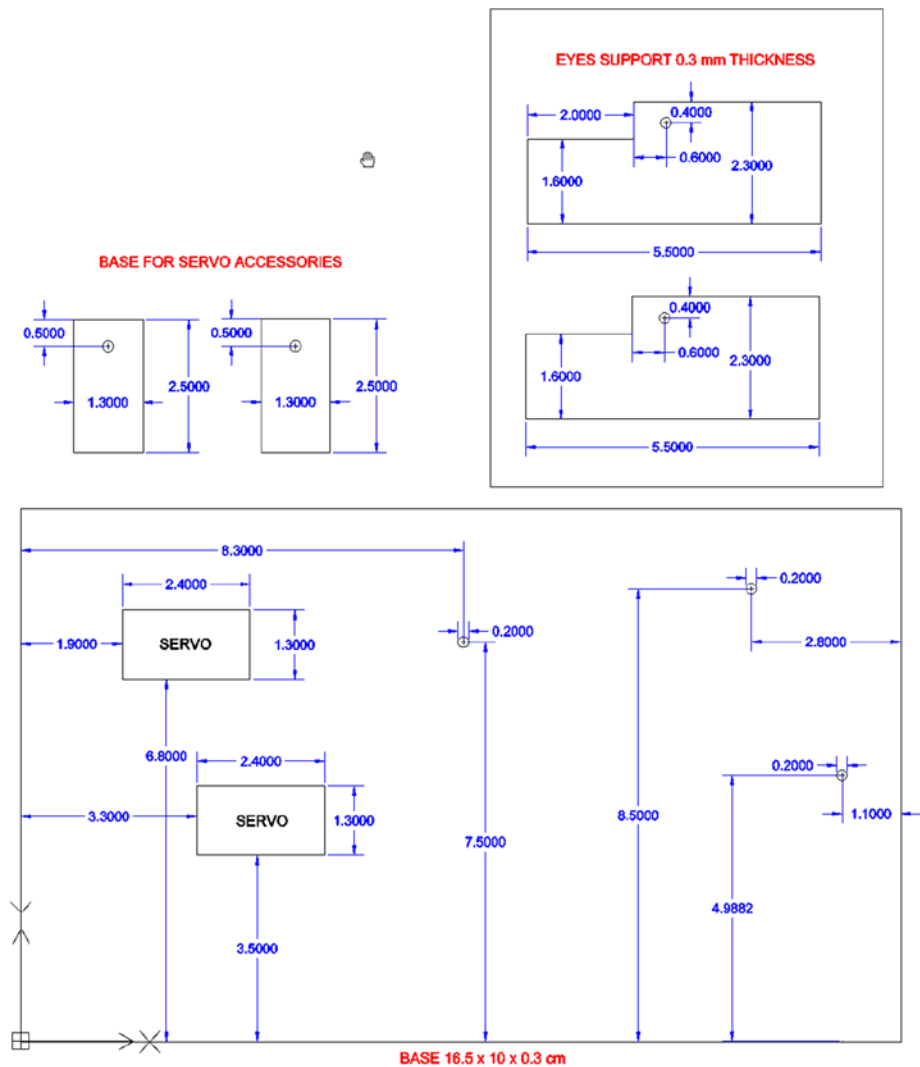
**Figure 13-5.** *The animatronic eyes base drawing*

To make sure you are printing correctly, take a balsa wood sheet cut to 16.5 x 10 x 0.3 cm and place the page over the balsa wood sheet, as shown in Figure 13-6.
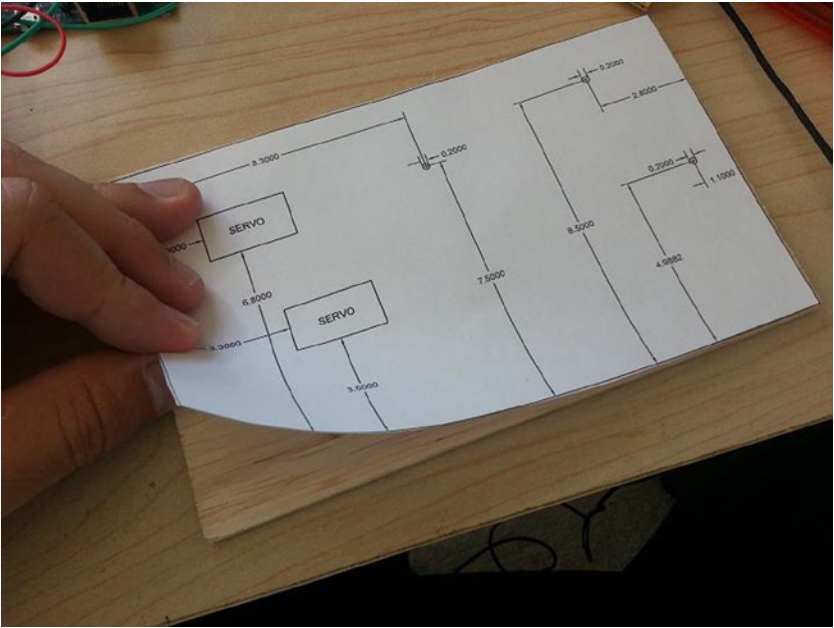
*Figure 13-6.* *Checking if the printing matches the balsa wood base*

If the dimensions match, you are printing correctly. Otherwise, you are doing something wrong. One of the most common mistakes happens when printing the PDF files. Make sure the **Actual Size** option is selected, as shown in Figure 13-7.
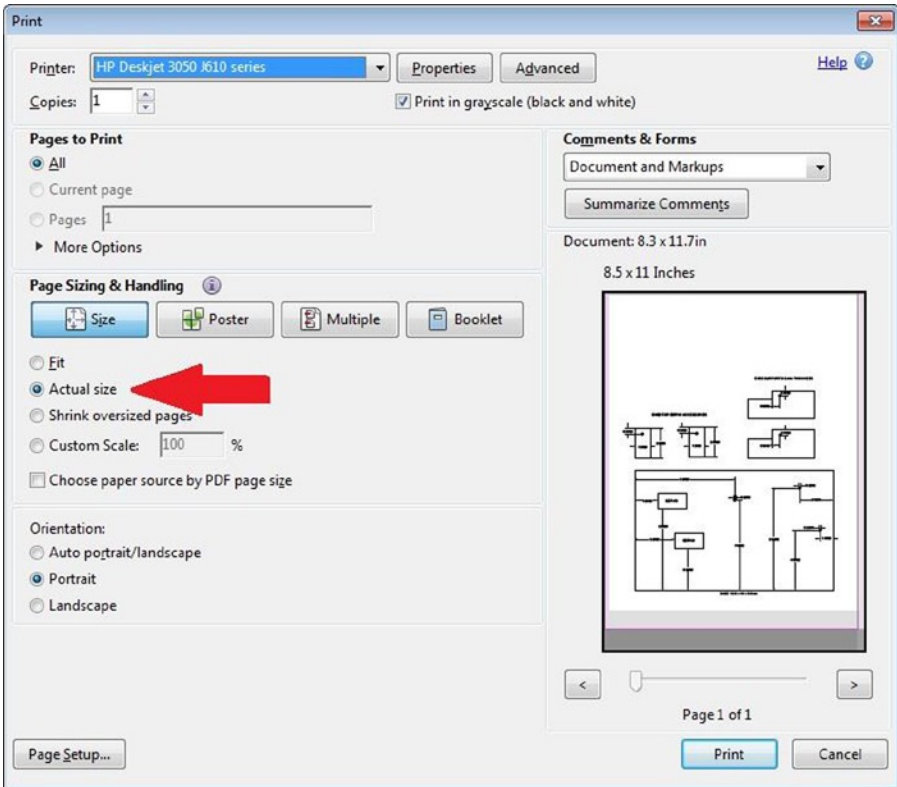
**Figure 13-7.** *Checking if the PDF is being printed with the Actual Size option*

If everything is right, keep the paper over the wood sheet and use the hobby knife and metal side of the ruler to cut the rectangles marked as SERVO, as shown in Figure 13-8.
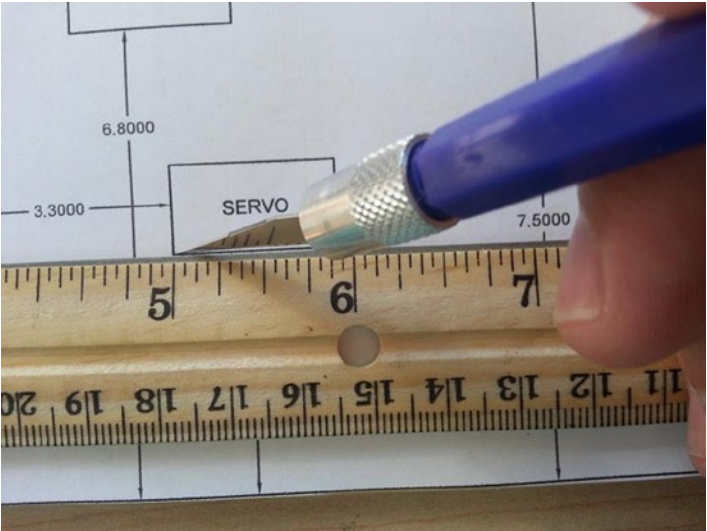
*Figure 13-8.* *Cutting the servos drawings*

Keep the paper over the balsa wood sheet and then make the holes. You can use a mini drill if you want, but the balsa wood is so soft that you can typically make these holes using a mini screwdriver, as shown in Figure 13-9.
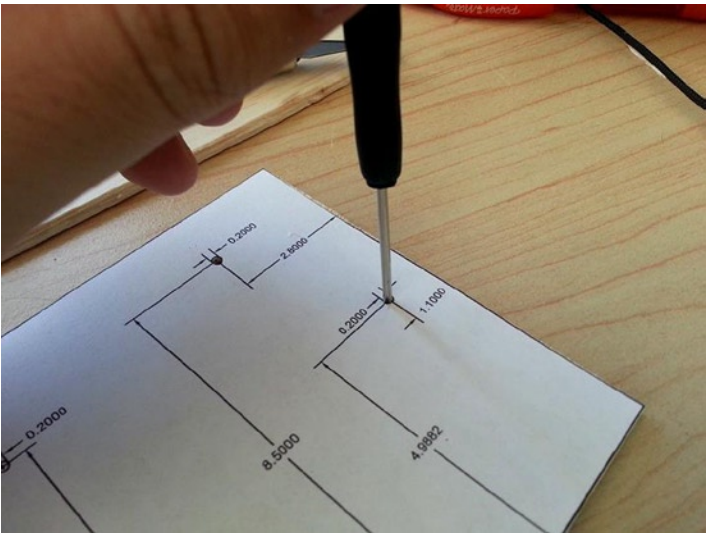


*Figure 13-9.* *Adding holes with mini screwdriver*

Now you need to prepare and add the accessories support eyes. Cut the rectangle limiting the **EYES SUPPORT 0.3 mm THICKNESS**. Use any regular balsa wood sheet with 0.3 mm thickness to cut in same manner as with the base, as shown in Figure 13-10. Keeping the paper over the wood, make the 2mm holes using a mini screwdriver or a mini drill.
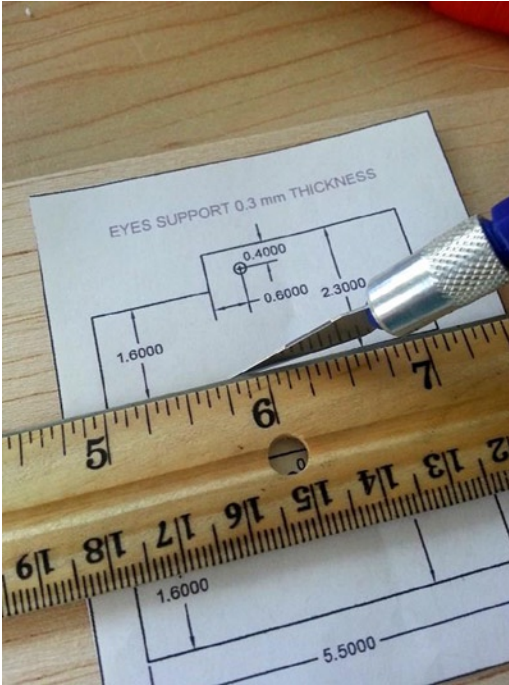


***Figure 13-10.** Cutting the eye supports*

Take the two eye supports and glue them on perpendicular to the base. Leave a uniform space of about 0.5mm, as shown in Figure 13-11.
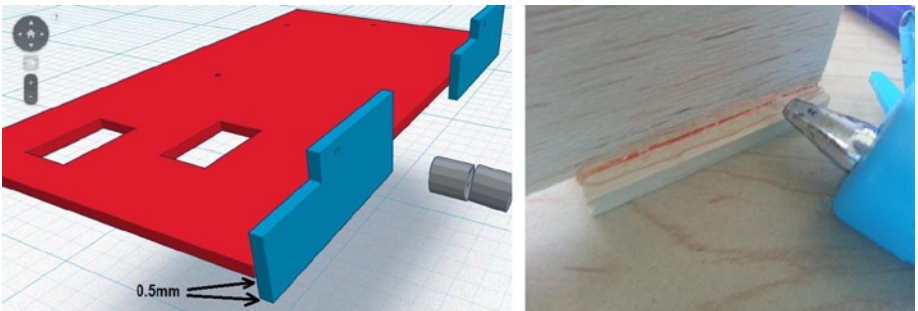


***Figure 13-11.** Placing the eyes supports*

Next, cut two pieces of 2.5 x 1.3 x 1.3 cm balsa wood to create the **BASE FOR SERVO ACCESSORIES** and make the holes as designed in the base_eyes file. Align the holes of these pieces with the holes created in Figure 13-9 and then glue these parts to the main base, as indicated by the arrows in Figure 13-12, right. Figure 13-12, left, shows a tip for aligning the holes using the mini screwdriver. Glue the base on all sides, as represented by the red arrows in Figure 13-12, right.
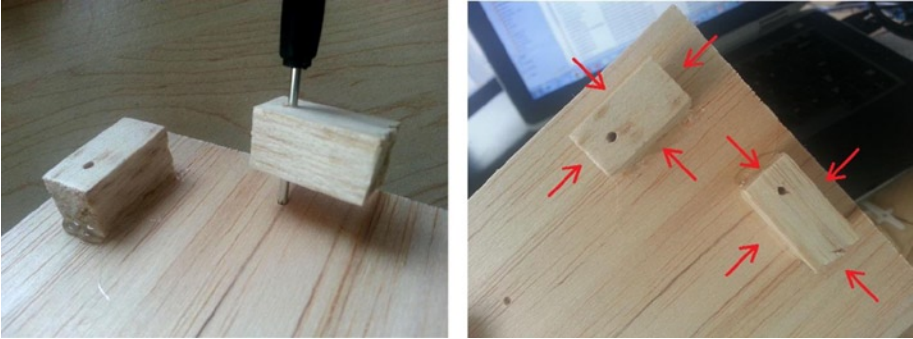


***Figure 13-12.*** *Placing the base for the servo accessories*

At this point, the wood cuts should appear as shown in Figure 13-13.
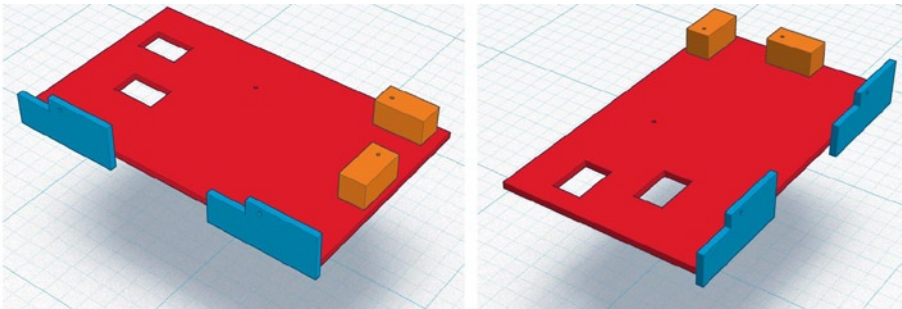


***Figure 13-13.*** *The animatronic eyes' frame with all pieces in place*

## Step 3: Gluing the Eyes to the Frame

Cut the nails 1 cm and push from the rear of **EYES SUPPORT**. Place a little bit of hot glue on the universal joint base and push the joints against the nail so the eyes can move vertically and perpendicular to the support, as shown in Figure 13-14, left. Next, cover the nail head with glue, as shown in Figure 13-14, right.
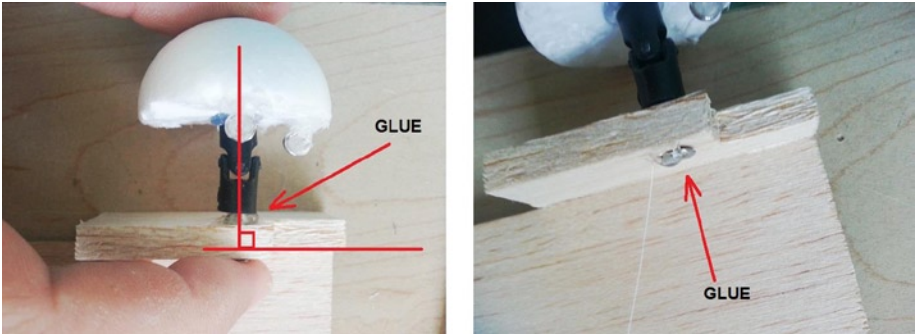
***Figure 13-14.*** *Connecting the eyes to the eyes' support*

Certify that both eyes have perpendicular vertical and horizontal movement. It's important to guarantee that the eyes have uniform and synchronized movement.

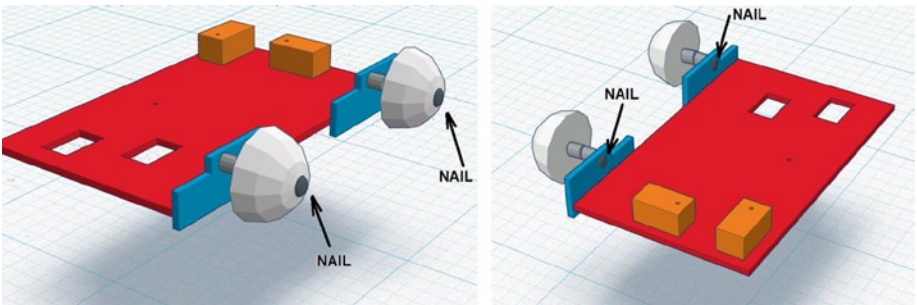Figure 13-15 shows the assembly expected at the end of Step 3.



***Figure 13-15.*** *How things should look at this point*

## Step 4: Connecting the Servos and Accessories

Before you connect the servos it is necessary to make sure they are at 90 degrees. Use the procedure called "Preparing the Servos" in the section entitled "Assembling the Robotic Arm" in Chapter 11 for all servos.

With all servos set to 90 degrees, it is time to prepare the accessories. The servo SG-90 comes with three accessories represented in Figure 13-16, left. You see a short with one single leg, another one that looks like a cross, and a third one with two legs. However, this project requires you to change the two leg accessories in a single leg by cutting it, as shown in Figure 13-16, right.
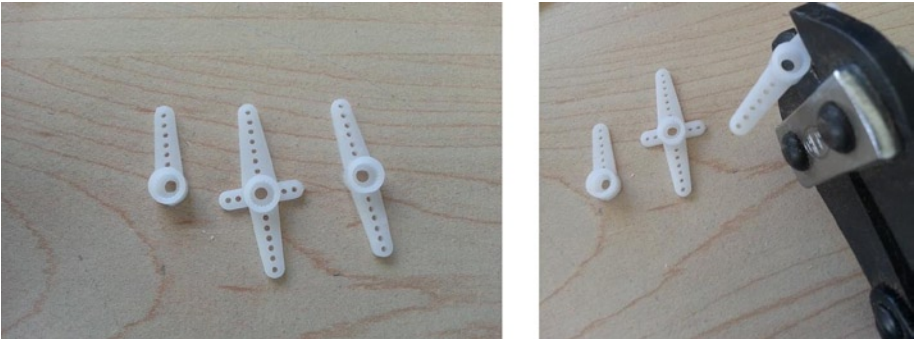
***Figure 13-16.*** *The servo accessories (left) and cutting the accessory (right)*

Then place the servos on the base using the screws. The shaft and the cross accessory with seven holes should both toward the left, as shown in Figure 13-17.
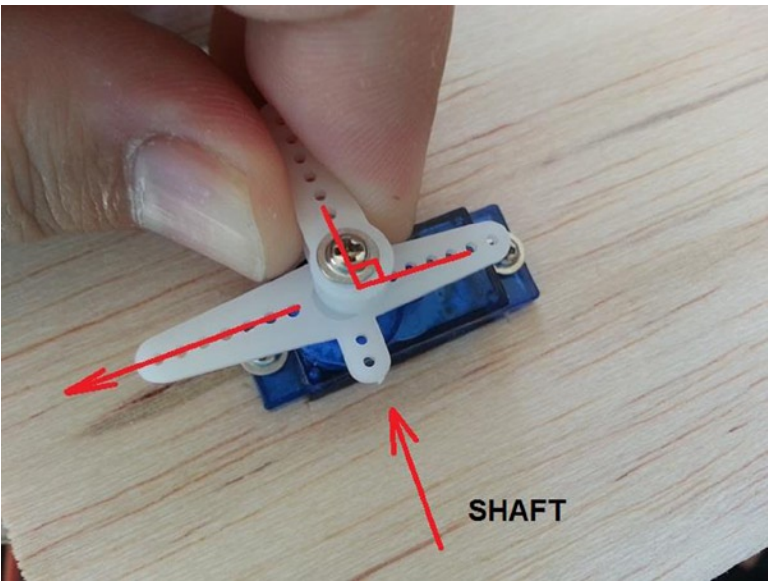


***Figure 13-17.*** *Placing the servos with accessories*

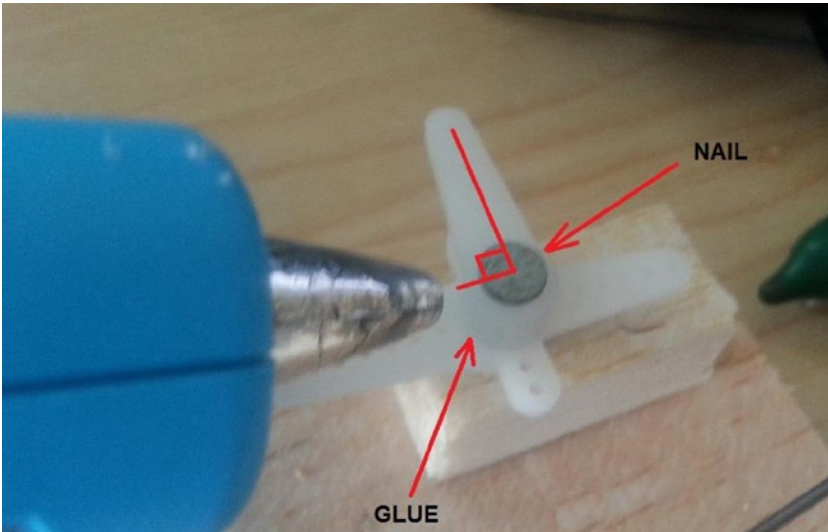Finally, glue other accessories at 90 degrees, as shown in Figure 13-18.



***Figure 13-18.*** *Placing the servos with accessories*

Alternatively, you can replace these nails by mini machine screws and nuts because it's better to glue the salient of the nail to the bottom of the wood base to avoid any release.

## Step 5: Preparing and Connecting the Rod

Using the pliers, cut and fold the music wires and create the six rods as shown in Figure 13-19. The code/plans folder contains a document entitled rods, which contains these rods available in AutoCAD and PDF.
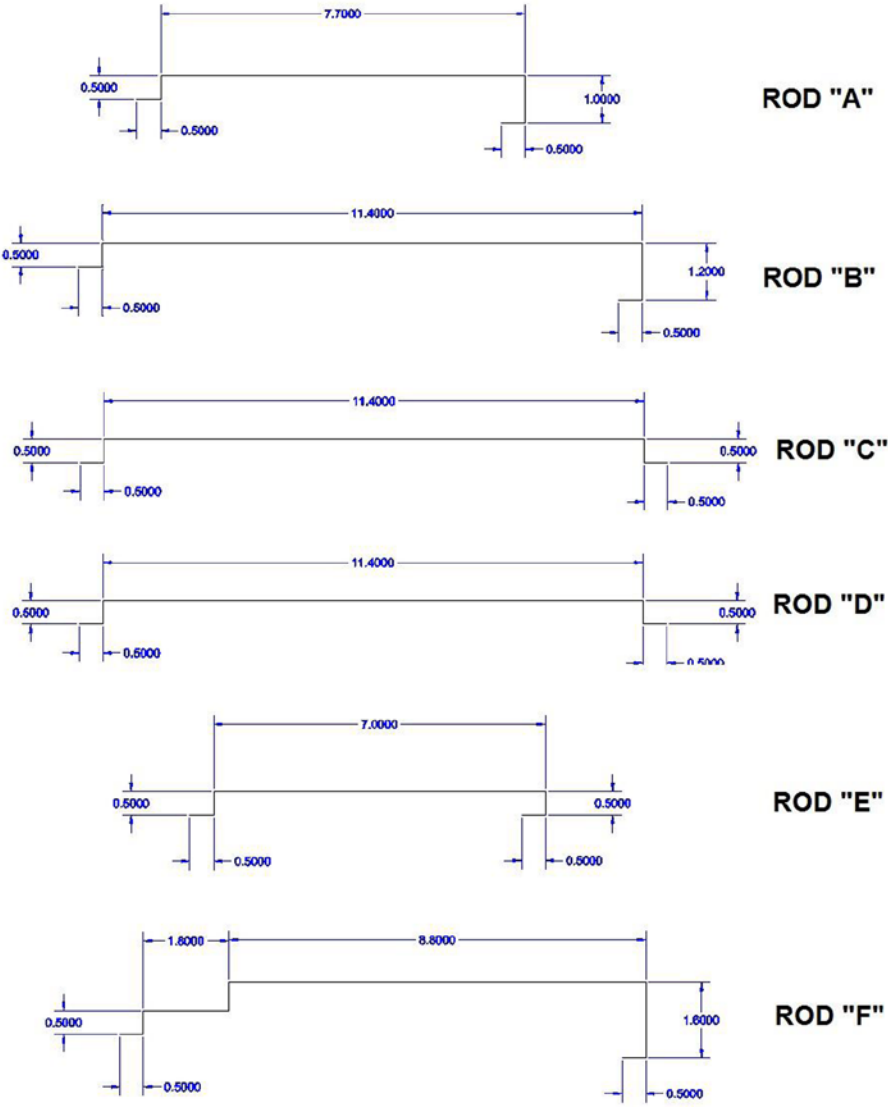
*Figure 13-19.* *The rods created using music wires*

Then connect the rods to the eyes and accessories, as shown in Figure 13-20.

**Figure 13-20.** *The rods have been connected*

At this point, your animatronic eyes are ready to be tested. This is explained in the next section.

## Mounting the Animatronic Mouth

To assemble the mouth you need a few pieces of wood and two servos, as presented in Table 13-2. The placement of the wood to have the mouth functional is shown in Figure 13-21.
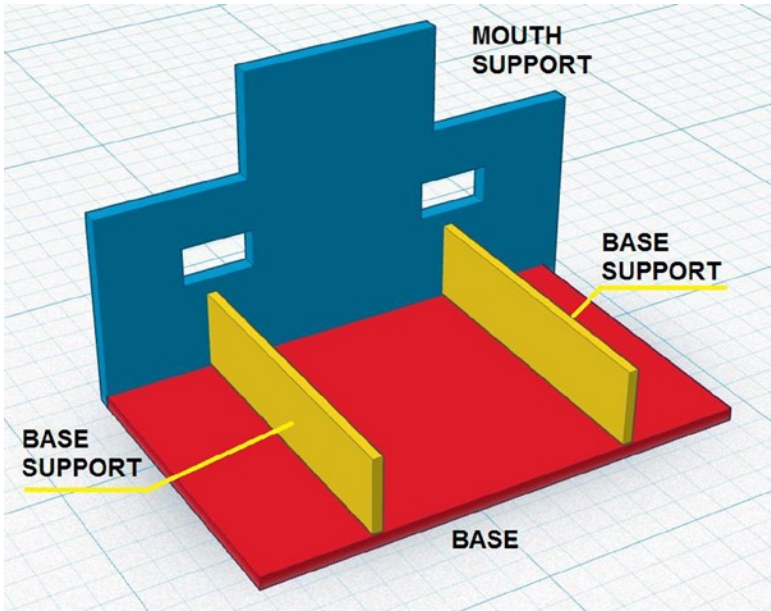
**Figure 13-21.** *The mounting to support the mouth*

The following steps are required to mount the animatronic eyes.
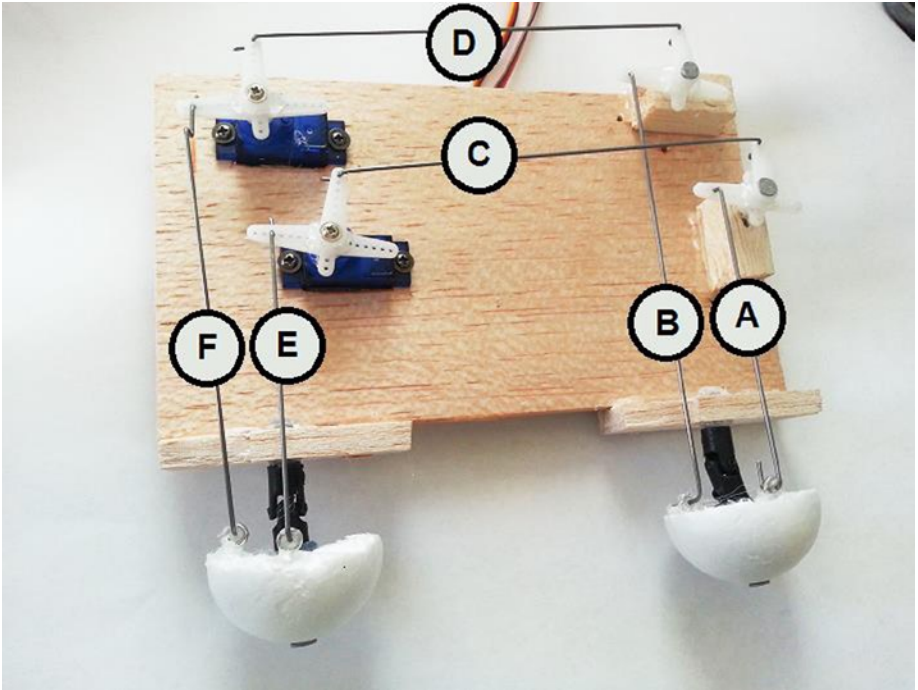
## Step 1: Mounting the Mouth Frame

Open the file named mouth located in the code/plans folder using your preferable extension and print the drawing using a regular laser or inkjet printer.

This drawing contains the mouth support, two BASE SUPPORTS, and the pillar to provide the support between the animatronic eyes and the mouth assembly, as shown in Figure 13-22.

**Figure 13-22.** *The mouth frame plans*

Using the same technique explained in the Step 2 in the previous section, cut the balsa wood and place the printing over the wood. Be careful not to cut your hands.

The base is not drawn because it is a simple piece of 10 x 16 cm wood and if you bought the assorted balsa wood package from Hobby Lobby, you already have a piece of wood cut to these dimensions.
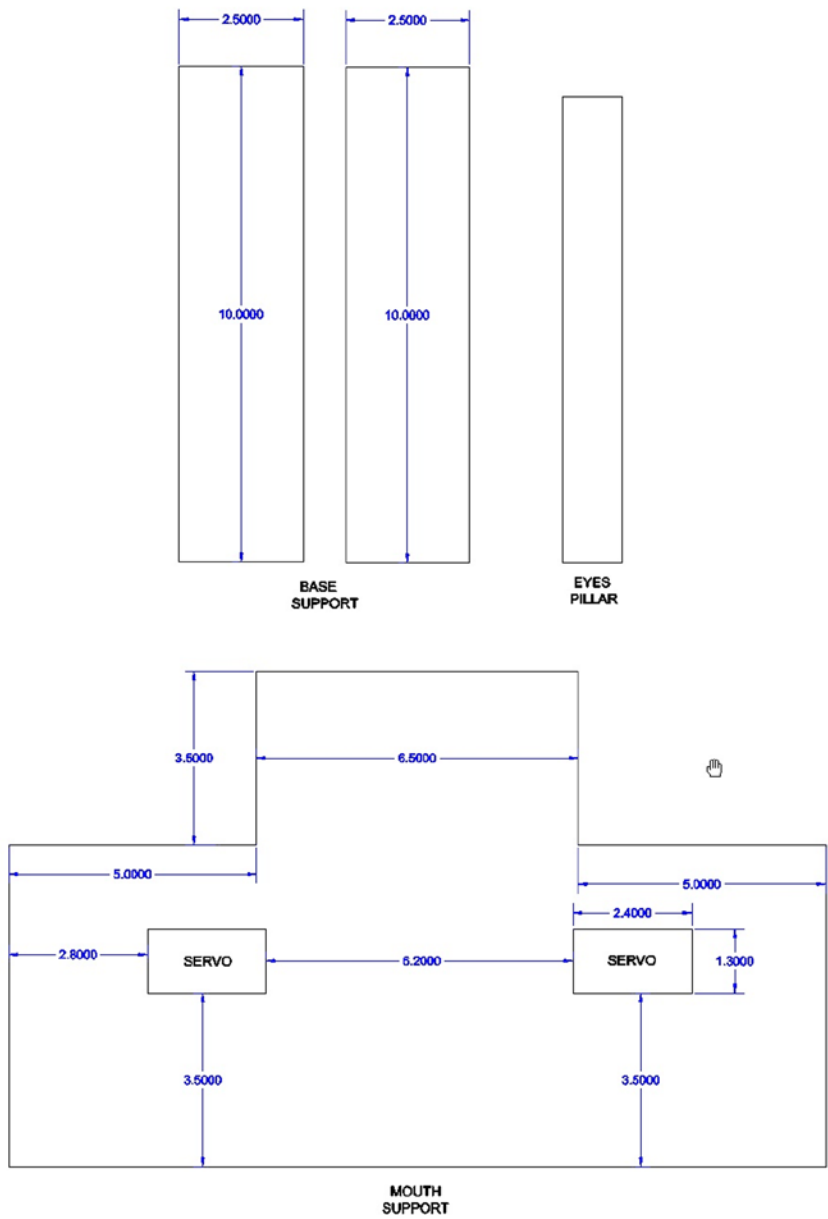
## Step 2: Gluing the Mouth Frame

Simply hot glue the frame and form the structure shown in Figure 13-21. Note that it's best to keep the base supports at least 2.5 cm (around 1 inch) from the base's edges.

## Step 3: Placing the Pillar

In the animatronic eyes, connect the pillar as shown in Figure 13-23. Screw the screw in the hole located in the central part of the animatronic eyes' base.
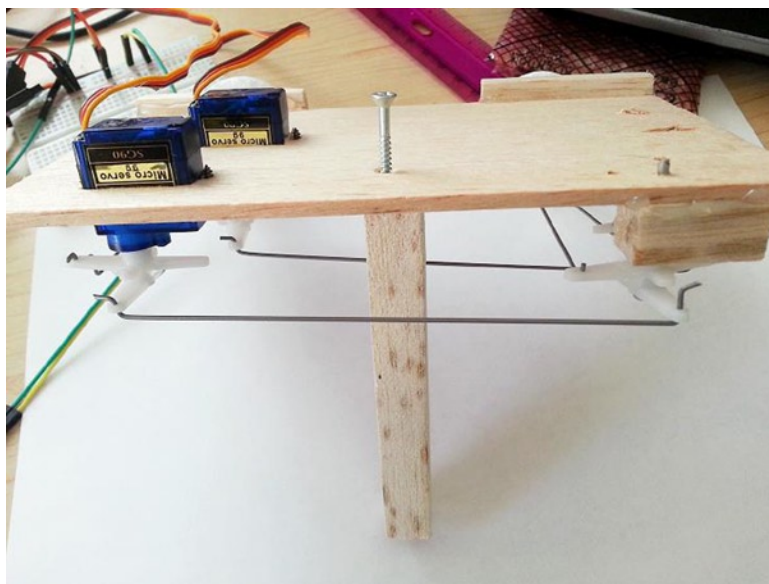


*Figure 13-23.* *Connecting the pillar wood to the animatronic eyes' base*

## Step 4: Placing the Servos

Before you place the servos, make sure they are calibrated to 90 degrees using the same procedure in Step 4 of the previous section.

After they have been adjusted to 90 degrees, place the servos so that the shafts are toward the centers and connect the accessories, as shown in Figure 13-24.
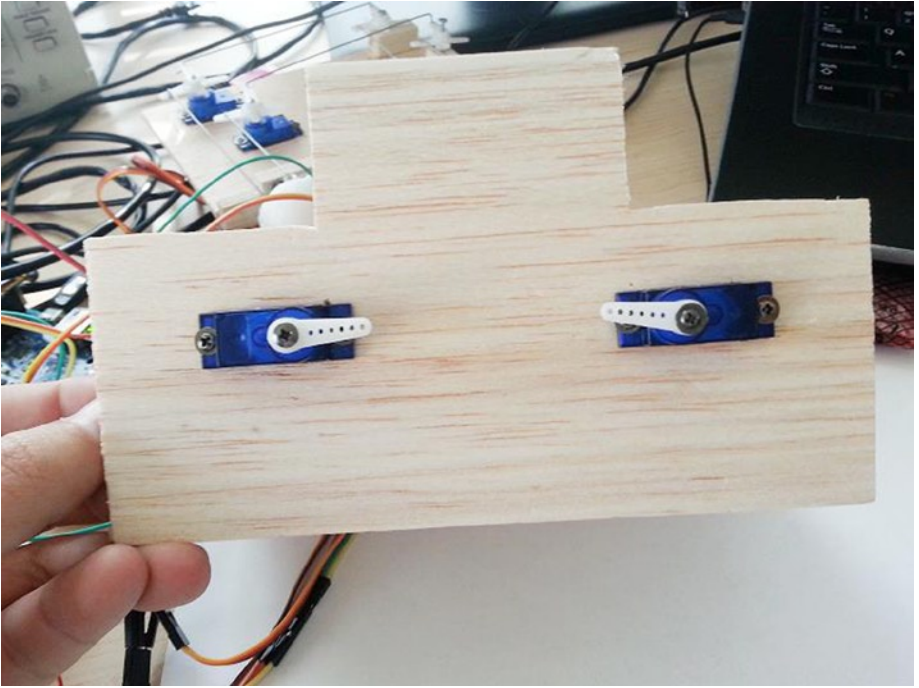
**Figure 13-24.** *Placing the servos in the mouth frame*

## Step 5: Preparing the Mouth

Using the tubular latex rubber, insert the rice grains into the rubber. The idea of rice is to prevent the rubber from folding and making corners in the material. With the rice inside, the rubber will bend properly and simulate a real lip, thereby providing a better visualization of the emotion.

It might sound difficult, but I recommend you use a small plastic bag for the rice and insert the rubber in order to force the rice into the rubber tube, as shown in Figure 13-25.

*Figure 13-25.* *Inserting rice into the tubular rubber*

## Step 6: Connecting the Mouth

Using small piece of wire, connect the mouth to the servo's accessories. Note that the servos at this point are at 90 degrees so the robot should have a serious expression. This ensures that the rubber is in line if you look directly at the mouth. If the rubber is bending, try to reduce and adjust the latex; otherwise you will not achieve good expressions when the rubber is manipulated.

## Step 7: Joining Everything

Place the animatronic eyes at top of the mouth frame and hot glue and you are done. Figure 13-26 shows the final mounting with all the expressions.
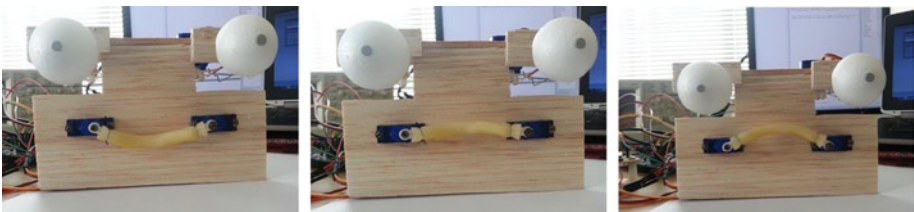


*Figure 13-26.* *The final mounting with all three emotional expressions*

# Controlling the Animatronic Eyes and Mouth

This section discusses how to assemble simple hardware or how to re-use the board servo control built in Chapter 11 to control the animatronic eyes and mouth. A simple sketch is also discussed that allows you to control the face by re-using the Thumbstick library created in Chapter 11.

## The Hardware

To control the animatronic eyes, the only thing you need is an analog thumbstick. The board used to control the animatronic eyes is the same one used to control the robotic arm in Chapter 11. It contains three thumbsticks, so two of them will be useless here.

However, if you did not build the servo board control used in Chapter 11 or want to use a new circuit, you might build a very simple circuit with only one thumbstick using the material listed in Table 13-5.

***Table 13-5.*** *Materials List for the Servos Controller*

| Number | Description |
| --- | --- |
| 1 | Thumbsticks SKU 121340 from dealxtream.com or equivalent |
| 1 | 1K Ohm 1/4W resistors (only if you ordered the previous thumbstick) |
| 5 | DuPont male to female cables |
| Some | AWG23 0.5mm2 or breadboard cables with at least two different colors (30 inches is enough) |

For a full understanding of how the thumbsticks work, read the section entitled "The Thumbsticks" in Chapter 11. If you ordered the thumbstick SKU 121340, you should prepare the thumbstick by soldering the 1K Ohm resistor as shown in Figure 11-40 of Chapter 11.

The connection with Intel Galileo is quite simple, as shown in Figure 13-27.
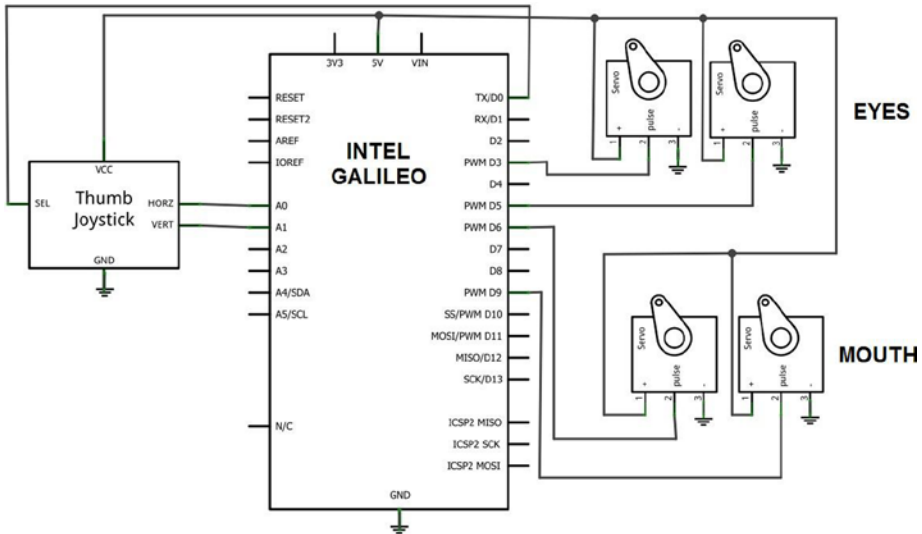
**Figure 13-27.** *Robotic face control schematics*

The two servos responsible for the eyes are connected to the PWM headers 3 and 5, while the other two servos responsible for the mouth are connected to PWM headers 6 and 9, respectively.

The next section discusses how to create the software to control the eyes and mouth.

## The Software

The idea is to create software that can control the animatronic eyes and the mouth independently using a single thumbstick. Consider the following explanations:

- **The eyes:** Two servos are used to control the eyes. One of the servos controls the horizontal movement and the other one controls the vertical movement, converting the thumbsticks movements. The mechanical characteristic of this project limits the servos from moving between 40 to 140 degrees. If the servos moved more than that, it could damage the mounting or simply disconnect the rods. When the eyes are looking straight ahead, both servos are at 90 degrees as calibrated in Step 4 of section entitled "Mounting the Animatronic Eyes." To control the servo, the API Thumbsticks created in Chapter 11 is re-used, making the sketch very simple. Note that it is necessary to install the API, so you need to follow the instructions in the section entitled "**Installing the API**" in Chapter 11.

- **The mouth**: Two servos are used to control the mouth in a very simple way. When the servos are at 90 degrees, the expression is serious. When they raise 30 degrees, the mouth forms a sad expression. If they move down 30 degrees, the mouth looks happy. The software is very simple; you just attach the servos and move them as you learned in Chapter 4. The only trick is that when one servo moves, the other one must have the inverse movement in order to synchronize the lips, as shown in Figure 13-26. This is due to the position of servos having the shafts pointing to each other, as shown in Figure 13-24.

The sketch is shown in Listing 13-1 and is located in the code/robotic_head folder.

*Listing 13-1.* robotic_head.ino

```
#include <Thumbstick.h>
#include <Servo.h>

#define MOUTH_IS_MOUNTED 1 // use 1 if yes, or any other thing if it is not

enum {
  MOUTH_HAPPY,
  MOUTH_SERIOUS,
  MOUTH_SAD,
  MOUTH_UNKNOW_STATE
} typedef MOUTH_T;

// which PWM channels the eyes are mounted?
#define PIN_SERVO_EYES_H      3
#define PIN_SERVO_EYES_V      5

// Declaring the servos in the arm
Servo eyes_h;
Servo eyes_v;

//Mouth servos
Servo mouth_right;
Servo mouth_left;

// Declaring the sticks with
// 8 is the joystick selection or you can connect directly to 5V
// A0 - is the X axis
// A1 - is the Y axis
// 0  - is the thumbstick SELECT (button)
// 40 and 140 - are the angle limits for X axis
// 40 and 140 - are the angle limits for Y axis
Thumbstick joystick1(8, A0, A1, 0, 40, 140, 40, 140);
```

```
// Variable used to store the result
Thumbstick::JOYSTICK_XY_T res1;

int mouth_state = MOUTH_HAPPY;

void setup()
{

    // for debug purposes let's use the serial terminal
    Serial.begin(115200);

    // init the joystick pins
    joystick1.initPins();

    eyes_h.attach(PIN_SERVO_EYES_H);   // joystick 1 - axis -> X
    eyes_v.attach(PIN_SERVO_EYES_V);   // joystick 1 - axis -> y

    // eyes in initial position
    eyes_h.write(90);
    eyes_v.write(90);

    // a small delay to allow the user to open the serial terminal
    delay(3000);

    Serial.println("*** Calibrating Joysticks **** ");
    Serial.println();
    Serial.println("Reading the CENTERS of each control stick");

    // let's run the calibration process for 15 seconds (15000 ms)
    boolean msgShown = false;
    long t = millis();
    while (millis()-t < 15000) {
        joystick1.calibrate();

        if (joystick1.isCenterCalibrated() &&
            msgShown == false)
            {
                msgShown = true; // only to show the message one single time
                Serial.println("Please, start spinning the thumbstick !!!!");
            }
    }
```

```
    if (DEBUG)
    {
        joystick1.dumpCalibration();
    }

    Serial.println("*** Calibration process completed.");

    if (MOUTH_IS_MOUNTED)
    {
        mouth_right.attach(6);
        mouth_left.attach(9);

    }

}

void loop()
{

    static boolean isGripperOpened = true;

    // checking the joysticks calibration
    if (joystick1.isCalibrated() == false)
    {
        Serial.println("ERROR: Joystick 1 not calibrated");
        // aborting
        Serial.println("You need to calibrate the joysticks !!! RESTART the
sketch pressing RESET button!! ");
        delay(1000);
        return;
    }

    // Note: this project uses absolute position, which means when you release
    // your hands from the thumbstick, the eyes will move to 90 degrees,
    // following the central position of the thumbstick.
    Thumbstick::MOVE_T move_type = Thumbstick::ABSOLUTE_POSITION;

    // reading the thumbstick
    res1 = joystick1.read(move_type);

    // moving the servos according to the responses of each thumbstick
    eyes_h.write(res1.X);
    eyes_v.write(res1.Y);
```

```
if (MOUTH_IS_MOUNTED)
{
    int angle = 0;
    // the mouth expression will change according to the
    if (res1.buttonPressed)
    {
        mouth_state++;
        if (mouth_state == MOUTH_UNKNOW_STATE) mouth_state = MOUTH_HAPPY;

        switch (mouth_state) {
            case MOUTH_HAPPY:
              angle = -30;
            break;

            case MOUTH_SERIOUS:
              angle = 0;
            break;

            case MOUTH_SAD:
              angle = 30;
            break;

            case MOUTH_UNKNOW_STATE:
            default:
            break;
        }

        mouth_right.write(90 - angle);
        mouth_left.write(90 + angle);

        if (DEBUG)
        {
            Serial.print("MOUTH STATE:");
            Serial.println(mouth_state);
        }

    }

}

delay(200); // adjust this delay to control the debounce
}
```

# Reviewing the Code

The code uses the Thumbstick API created in Chapter 11. The first line includes the API header file.

```
#include <Thumbstick.h>
```

If you are not using the mouth assembly, you can disable the code responsible for managing the mouth  by changing the following line:

```
#define MOUTH_IS_MOUNTED
        1 // use 1 if yes, or any other thing if it is not
```

The mouth assumes three states defined by the MOUTH_T type.

```
enum {
  MOUTH_HAPPY,
  MOUTH_SERIOUS,
  MOUTH_SAD,
  MOUTH_UNKNOW_STATE
} typedef MOUTH_T;
```

Then the servos are declared and the PWM headers that the servos are connected are set. The following code is according to the schematics of Figure 13-27.

```
// which PWM channels the eyes are mounted?
#define PIN_SERVO_EYES_H       3
#define PIN_SERVO_EYES_V       5

// Declaring the servos in the arm
Servo eyes_h;
Servo eyes_v;

//Mouth servos
Servo mouth_right;
Servo mouth_left;
```

The joystick1 object is created to control the eyes. The comment in the code explains which value is used in the constructor method.

```
// Declaring the sticks with
// 8 is the joystick selection or you can connect directly to 5V
// A0 - is the X axis
// A1 - is the Y axis
// 0  - is the thumbstick SELECT (button)
// 40 and 140 - are the angle limits for X axis
// 40 and 140 - are the angle limits for Y axis
Thumbstick joystick1(8,  A0, A1, 0, 40, 140, 40, 140);
```

In the setup() function, the joystick1 pins are initiated and the servos that manage the eyes are attached to their respective PWM channels. The initial position of 90 degrees is established so the eyes are looking straight ahead.

```
// init the joystick pins
joystick1.initPins();

eyes_h.attach(PIN_SERVO_EYES_H);   // joystick 1 - axis -> X
eyes_v.attach(PIN_SERVO_EYES_V);   // joystick 1 - axis -> y

// eyes in initial position
eyes_h.write(90);
eyes_v.write(90);
```

Next, the thumbstick must be calibrated in the same manner explained in Chapter 11. In order words, you need to spin the thumbstick in all directions so the software will recognize the maximum and minimum analog readings. The current sketch waits 15 seconds for the user to perform the calibration, as shown in the following code:

```
while (millis()-t < 15000) {
   joystick1.calibrate();

   if (joystick1.isCenterCalibrated() &&
      msgShown == false)
      {
         msgShown = true; // only to show the message one time
         Serial.println("Please, start spinning the thumbstick !!!!");
      }
}
```

Still in the setup() function, the servos that manage the mouth are set to the 9 and 6 pins according to the schematics shown in Figure 13-27.

```
if (MOUTH_IS_MOUNTED)
{
   mouth_right.attach(6);
   mouth_left.attach(9);

}
```

The loop() function is where the magic happens, as it's where the joystick is read and the values are passed to the servos. Note that the Thumbstick API now uses the ABSOLUTE_POSITION movement, which makes the eyes follow the position of the thumbstick exactly. This is different from Chapter 11. This means when you release the thumbstick, the eyes will assume the central position, looking straight ahead.

```
// Note: this project uses absolute position, which means
// when you release your hands from the thumbstick,
// the eyes will move o 90 degrees, following the central
// position of the thumbstick.
Thumbstick::MOVE_T move_type = Thumbstick::ABSOLUTE_POSITION;

// reading the thumbstick
res1 = joystick1.read(move_type);

// moving the servos according to the responses of each thumbstick
eyes_h.write(res1.X);
eyes_v.write(res1.Y);
```

Finally, if the mouth was assembled:

```
if (MOUTH_IS_MOUNTED)
{
    int angle = 0;
    // the mouth expression will change according to the
    if (res1.buttonPressed)
    {
        mouth_state++;
        if (mouth_state == MOUTH_UNKNOW_STATE) mouth_state = MOUTH_HAPPY;

        switch (mouth_state) {
            case MOUTH_HAPPY:
              angle = -30;
            break;

            case MOUTH_SERIOUS:
              angle = 0;
            break;

            case MOUTH_SAD:
              angle = 30;
            break;

            case MOUTH_UNKNOW_STATE:
            default:
            break;
        }

        mouth_right.write(90 - angle);
        mouth_left.write(90 + angle);
```

The next section discusses how to run the code.

## Running the Code

Download the sketch and then open the Serial Monitor by selecting Tools ➤ Serial Monitor or pressing Ctrl+Shift+M.

Then spin your thumbstick in all positions in order to calibrate it. If you calibrated it correctly, you should see a message like this:

```
*** Calibrating Joysticks ****

Reading the CENTERS of each control stick
centerX:510
centerY:517
Please, start spinning the thumbstick!!!!
Xmax:1023 Xmin:0 Ymax:1023 Ymin:1 centerX:510 Angle centerX:89 centerY:517
Angle centerY:90
*** Calibration process completed.
```

After the calibration is successful, you are ready to have fun. Move the eyes around in all positions and click the thumbstick to see the mouth change expression.

## A Video Sample

In the `code/video` folder, there is a file named `anim_eyes.mp4` that shows the animatronic eyes working with this sketch.

# Ideas for Improving this Project

You might want to consider improving this project with some of the ideas described next.

## Integrating it with OpenCV

Note that there is a flat area on top of the robot's head. That's left there so you can install a webcam and (using the same technique explored in Chapter 7) make the robotic "look" at your face through the camera. With OpenCV, it is possible to have the eyes track your face and make the robot to replicate your emotions.

In Chapter 7, you learned that the software can recognize three emotional states—happiness, sadness, and seriousness—which are the same emotions used in this project.

You could invest time merging the concepts in Chapter 7 with this chapter to make a very interesting robot.

## Adding Irises to the Eyes

This is just a cosmetic suggestion. Maybe you want something more realistic than nails for the animatronic eyes. If so, check out the `http://homepage.ntlworld.com/toneth/irisfullsheet.jpg` website, which includes several irises that you can print and glue to the robot's eyes.

# Summary

This chapter provides a very affordable solution for mounting a robotic face with animatronic eyes and a simple mouth to express three emotions. The project is also easy enough for anyone to assemble, using regular material found in office and electronic stores, which helps you avoid using expensive 3D printers or laser cutters.

In terms of software, the Thumbstick API created in Chapter 11 and re-used in this chapter makes the sketch very simple and easy to understand.