

# Embodied Speech and Facial Expression Avatar Progress Report 2

Presented to Professor Ricardo Gutierrez-Osuna  
on  
March 29, 2004

by  
Dan Harbin - Evan Zoss - Jaclyn Tech - Brent Sicking

# Table of Contents

---

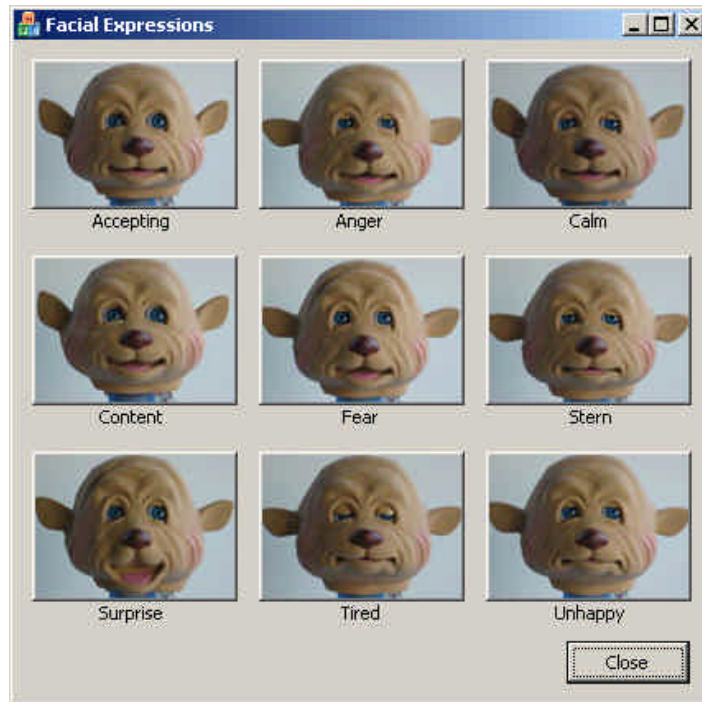
<b>Accomplishments .....</b>	<b>3</b>
Expression Control Dialog.....	3
Figure 1: Expression Control Dialog.....	3
SV203 Input Pins .....	3
Calibration.....	4
Motor Controls .....	4
Figure 2: Motor Control Circuit.....	4
Input circuits .....	5
Figure 3: Input Sensor Circuit.....	5
Eye motor sensors .....	5
Yano's skin .....	6
Figure 4: Yano head with MIT's stand .....	6
<b>Time Schedule .....</b>	<b>7</b>
Figure 5: Gantt chart .....	7
<b>Goals for the Next Two Weeks .....</b>	<b>8</b>
Calibration.....	8
Control .....	8
Streaming speech processing .....	8
Custom Motor Controls .....	8
Eye Motor Sensor.....	9
Circuit for Eye Sensor.....	9
Stuck Motors.....	9
Motor Power .....	9

## Accomplishments

---

### Expression Control Dialog

We have completed our highest level Yano control dialog, the Expression Control Dialog. This window contains pictures of the different preset facial expressions that Yano can make. When one of these buttons is clicked, it will move Yano's motors to a specific set of locations that are hard-coded into the software, to create the expression. The close button will return to the main menu.



**Figure 1: Expression Control Dialog**

### SV203 Input Pins

We can now successfully read the input pins off of the SV203. This took quite a bit of tinkering along with a great deal of communication with the folks at Pontech to figure out the appropriate set of commands to read the pins. First we must send one command that tells the SV203 to read the input pins, and then we need to delay a small amount (we use 10 milliseconds). Next we issue the serial port command to read the data being transmitted from the SV203. We experimented with several of our serial port library's commands for reading and found the one that works the best.

## Calibration

We completed the calibration routines for the mouth and the cheeks. Our method is the following: For each motor, pulse it in one direction until that direction's limit switch is triggered, then pulse it all the way back until the other limit switch is triggered, and count the number of pulses it takes. By doing this, we accomplish two things. One, the mouth and cheeks will always start out at the same position after being calibrated, which is necessary for the rest of our software. And two, we can count the maximum number of pulses it takes for the motor to travel its full distance which also allows our software to work as designed.

## Motor Controls

All motor controllers are now working simultaneously on one circuit. We can run all the motors at the same time, each individually, or a combination if we like. The three circuits are identical and conveniently run off of the same power source.

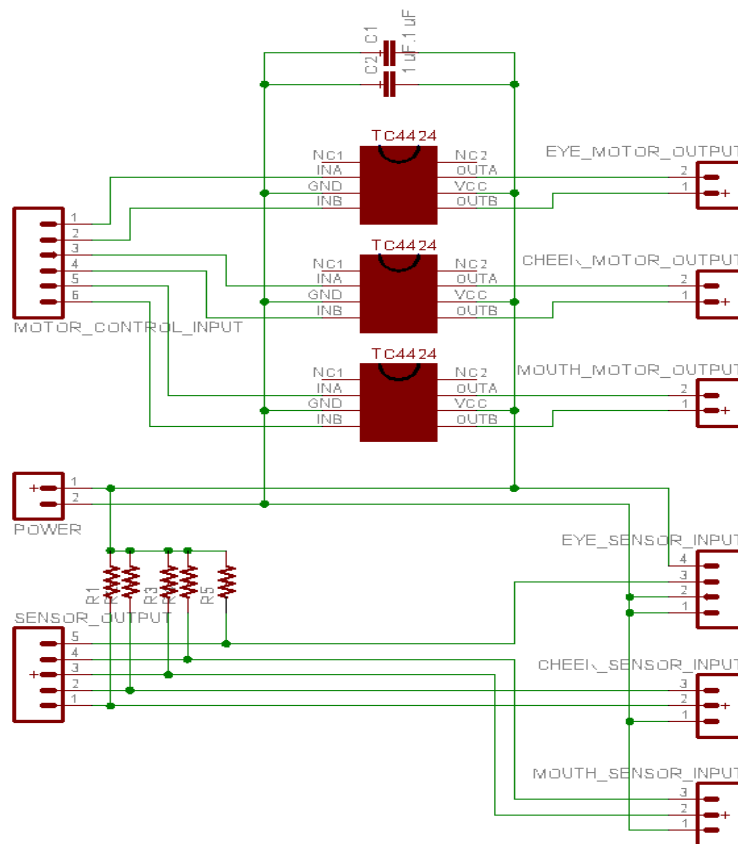
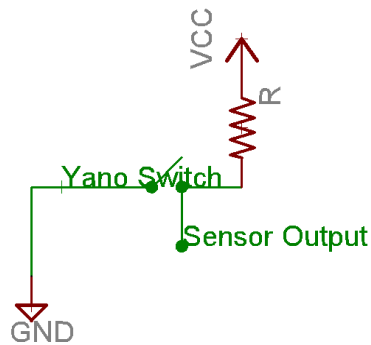


Figure 2: Motor Control Circuit

## Input circuits

The input circuits are built for the mouth and cheek sensors. These circuits are made to read the state of the motor limit switches, open or closed. The circuit is quite simple and creates a 6V, high, value if the switch is open, and a 0V, low, value if it is closed. The SV203 can then read these voltages and the software uses them to detect the status of the limit switches.



**Figure 3: Input Sensor Circuit**

## Eye motor sensors

We have experimented some with the eye motor sensors to try and figure them out further. At this point what we know is that there are 2 sensors, each of which has an IR LED that shines at some type of receiver through a gear. The receiver will only detect the light when a hole in the gear lines up with the LED. This allows the position of the gear to be determined. However, all of this is in the deep recesses of Yano's head. Investigating these sensors further would pretty much involve taking every single gear and switch out of Yano's head, and we would like to avoid this if possible. So, we've emailed the guys at MIT, because they did in fact take apart the entire head, and could perhaps offer us some insight into how the eye switches work. In the meantime, we're testing the circuit from the outside to try and figure out how we can use it. One thing interesting that we have found out is that it looks like the sensors require power from the eye motor to be supplied, which means the eye motor must be on in order for the eye motor sensor to work. This would require different methods of calibration for the eye motor.

### **Yano's skin**

The latest thing we have done is put the skin back on Yano's face to see what happens. We've only tried this briefly but we have noticed a few things. First, the movement of the mouth and cheeks is not nearly as jerky as it was without the skin; we suspect that the additional resistance provided by the skin acts to smooth out the pulsed movement and make it more natural. Also, it took more power, in the form of a longer pulse to actually make the motors move. This is good because it brings the pulse width required for the mouth and cheeks up closer to the long ones that we need for the eyes, so all three motors should move more similarly. Lastly, the motors seemed to get stuck a little bit more. But we really need to find a better stand to hold Yano's head and investigate this more.



**Figure 4: Yano head with MIT's stand**

# Time Schedule

According to our Gantt chart (Figure 5), we appear to be mostly on schedule. Since the critical design review, we have created a GUI controlling emotions and expressions, began testing the usability and functionality of our software controlling the motors, and progressed with motor calibration. We have successfully calibrated two of Yano's three head motors. As mentioned above, calibrating the eye motor is proving to take longer than expected, so we need to allow another week or two to complete calibration.

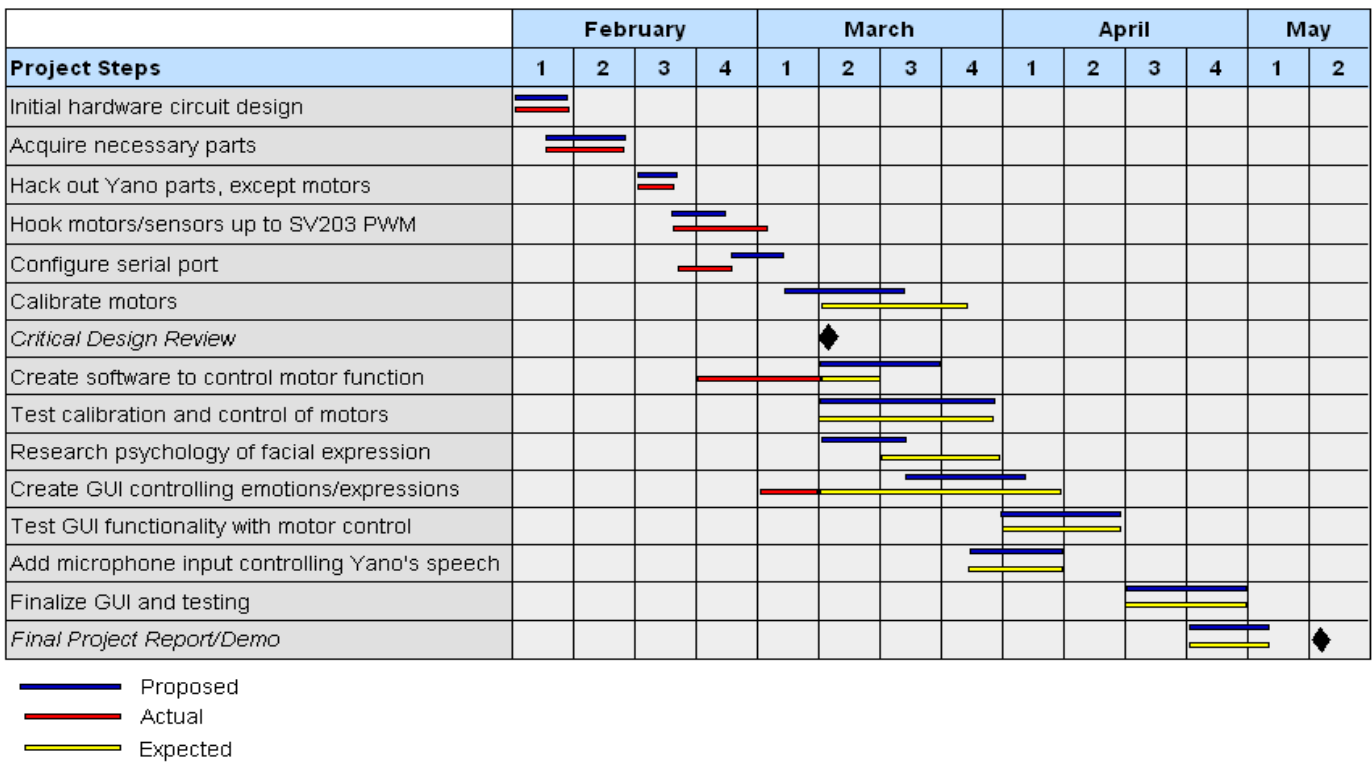


Figure 5: Gantt chart

## Goals for the Next Two Weeks

---

### **Calibration**

Although we have not completely figured out the eye motor switches, it is looking like they will need power to work. So, we will need to think about alternative ways to calibrate, and probably even control, the eye motor. This should be fine, because the eye motor is not as important as the other two for neither the expressions nor the simulated speech aspects of our software.

### **Control**

There are a few problems we're finding with our pulsed method of control. One is that need to send bigger pulses to the motors than we thought just to get them to move at all. Also, sometimes the larger pulse is too much and the motors get pushed too far and get stuck, or at least they get out of calibration. We still have a lot of experimenting to do with this and investigation into why these things are happening, but we might want to think about alternative methods of control that we could test and see if they work better.

### **Streaming speech processing**

So far, we are not having any luck finding working, sample Visual C++ code that analyzes volume of input audio. Plenty of code is available; however, none of it appears to work with the computers at the lab. Over the next two weeks, we hope to overcome this dilemma by either finding working sample code, or attempting to write simple code to analyze the volume of input audio ourselves. If we successfully get code working to analyze the volume, our next step will be to create an interface to record our own audio, and then analyze it.

### **Custom Motor Controls**

At this point, we treat all three motors basically the same. The only difference is what pins we set to move them and what pins we read to check their sensors. However, it is becoming apparent that due to mechanical lodes on the motors and the different ways in which we use them, the motors will not be able to be treated the exact same. So we will need to begin incorporating the ability to customize each motor's class a little bit, and possibly even some of the controls we have set up.

### **Eye Motor Sensor**

This is the biggest roadblock in the way of the hardware right now. We really need to get these sensors figured out one way or another. We emailed MIT early this week and have yet to get a reply, but they may be on Spring Break. We will give them to mid next week probably to reply, and then we'll have to go ahead and take Yano completely apart to get a look at this circuit and see what we need to do. The reason we are waiting is because the eye motor is not really important at all to calibrate or sense what position it's at. So it is tough to justify taking Yano's head completely apart which will run the risk of messing up the parts that area already working.

### **Circuit for Eye Sensor**

Once we figure out how the sensor works, we will need to construct a circuit to read it. This will also need to take into consideration what we want to do in terms of software calibration and control of the eye motor.

### **Stuck Motors**

We also need to figure out why the motors seem so prone to getting stuck at the extremities of their movement. Also, we will want to figure out possible solutions to the problems, once they are identified. This will mostly involve a lot of testing, and it will be good to do this with Yano's skin on.

### **Motor Power**

Although nothing seems like it is running particularly wrong when all the motors are running, we think it would be worthwhile to do some investigation into the current being drawn by the motors and if our power supply is providing enough. This may be something that we would like to look into increasing, to make sure the motors have the maximum power they can, even when all are running simultaneously.