usitt

td&t

2017 Summer

theatre design & technology

Turning Heads: Automation on a Budget Using Arduino

By Thomas Fagerholm and Andrew Armas
Southern Illinois University

Figure 1

Arduino and similar microcontrollers offer an inexpensive opportunity to access scenic automation without breaking the bank or requiring specialized training. The following outlines resources and approaches to automation using examples from Southern Illinois University's spring 2016 production of *Tartuffe*.

The director focused on gossip as an integral element of the show. To help support this concept, the scenic designer suggested using marble busts of contemporary political figures like Bernie Sanders, Barack Obama, and Donald Trump (Figure. 1). Additionally, it was suggested that these busts interact with the performers. We saw this as an opportunity to incorporate automation using an Arduino microcontroller to make the head movements hands-free and cue-able.

To turn the busts, we chose to use four motors, one for each, so that their positions could be individually adjusted or cued if desired. Knowing that the busts needed to turn their heads from side to side and that the motors would have to carry the weight of the busts themselves, mostly foam, we explored using hobby servo motors. These little gear motors have a wide variety of uses and come in an assortment sizes, torque, and speed ratings. Additionally, they run on only five volts direct current (DC), meaning we could use a simple DC power supply. One benefit of these motors is that they have position feedback information built into them, allowing us to specifically tell them where to turn. We looked into different motors with a higher amount of torque to help compensate for the weight of the busts and found that our best options for servo motors were Futaba S3010 and S3305 (Figure 2).

The hobby servo motors allowed us a wide range of motion that was programmable through the Arduino; however, how far the heads could turn was limited. Hobby servos with position control have a 180-degree range before stripping or damaging a gear. Unfortunately, due to a programming error, we encountered this during a test run. Realizing we had damaged one of the nylon gears of the Futaba S3010 servos, we adjusted our coding to include software limits. This effectively limited the range of motion, from five degrees to 175 degrees, as a safety in case the weight of busts caused the motors to slightly pass their initial position. As an extra precaution, we upgraded to a Futaba 3305 servo, containing metal gears.

In order to control movement, we connected the signal leads from the motors to four of the Arduino's Pulse Width Modulation (PWM) outputs (Figure 3). Coding for the Arduino allowed us to send PWM signals for precise speed and position to the motors. These leads were connected to the Arduino through a terminal block. Power for the motors was separated from the Arduino and connected to an independent power supply preventing the possibility of an overload. This allowed us to avoid using servo motor shields and simplified programming by using Arduino's built in servo program library. The terminal blocks allowed for secure electrical connections and helped with cable management between devices, allowing us to bridge terminals or connect more than one wire to each. Cable runs to the motors were done with three-pin XLR mic cable for quick connections and because of XLR's shielding properties. The shield of the cables was connected to the power supply ground to keep electromagnetic interference (EMI) separate from the Arduino.

Figure 2

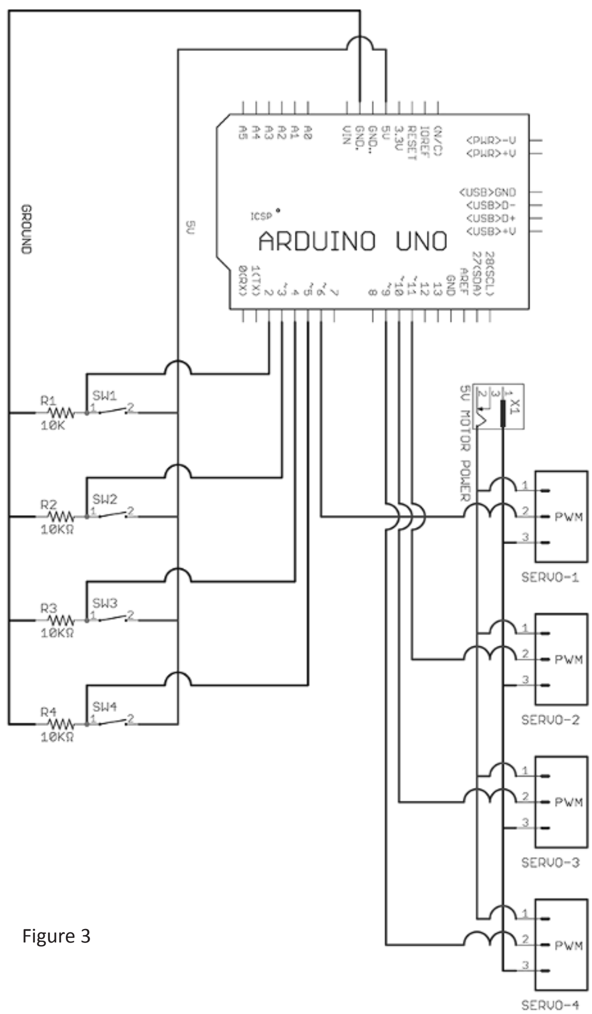

Figure 3

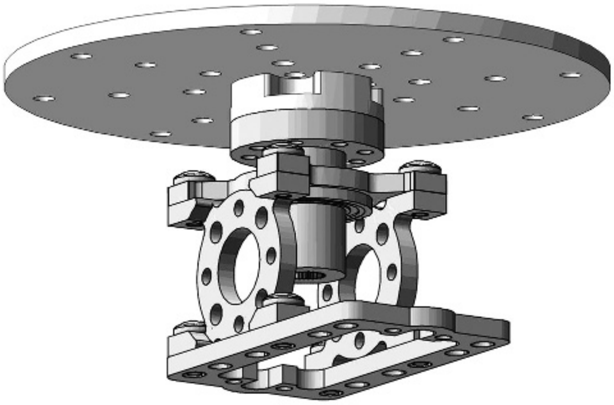

Figure 4

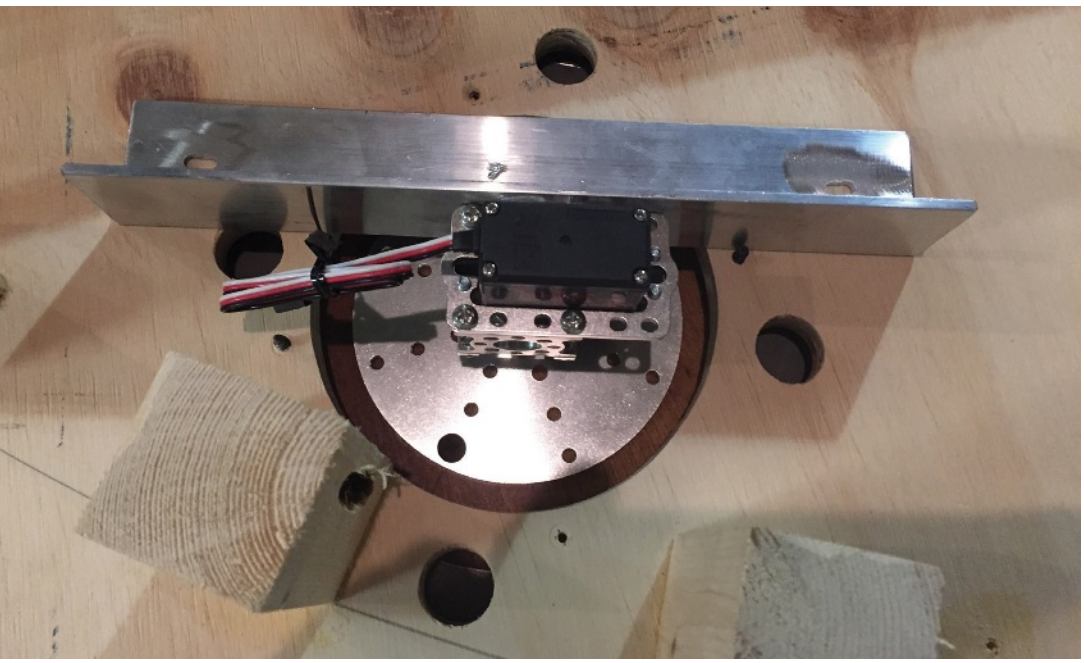

Figure 5

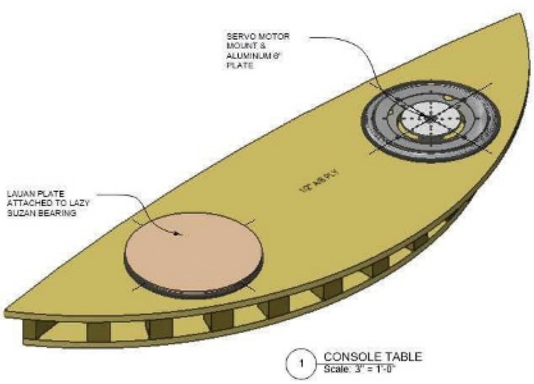

Figure 6

Ferrite collars were added to the unshielded leads from the motors to help further eliminate EMI.

For cuing the busts, we used a combination of pre-programmed movements in the Arduino triggered by switches connected to digital pins on the Arduino. For example: if button No. 1 is closed, head No. 1 will rotate to 90 degrees or if buttons No. 1 and No. 2 are closed, head No. 1 will rotate to 175 degrees. In essence, the buttons functioned as a four-bit binary code that correlated to the Arduino and the programmed movements. For testing, we used two push buttons to send various binary codes, but for production, we used a Creative Conners Stagehand FX unit with Spikemark software. The Stagehand FX has four controllable solenoid switches that could be precisely programmed. Using this combination eliminated concerns about consistency when manually pressing a series of buttons to complete one command. Note that this same effect could be achieved without the Conners equipment by programming within the Arduino to cycle between preset positions. This was used on the same project to control LEDs in the eyes; however, the programming was more complex.

To mount the servo motors to the busts we used ServoBlocks from ServoCity.com (Figure 4). These allowed us to mount each motor to an aluminum angle that was screwed onto the wall-mounted console table (Figure 5). An extension from each motor connected to a 6" radius plate, mounted to the bottom of the busts. Each bust sat on a Lazy Susan bearing that was also mounted to the table for additional support (Figure 6).

The movements of the busts added to the overall comedic effect and achieved what the director sought. This project served as a learning opportunity for our students and staff to study a new piece of technology that has endless possibilities in theatrical application and is steadily being incorporated to special effects in theater. This project also became a collaboration not only between students and faculty, but between departments and disciplines. Consulting with the computer-science and industrial design departments not only aided in trouble-shooting and programming, but also allowed their students to be a part of an art form that they may not have experienced otherwise. These statues were our stepping stone into a new world of technology and collaborations and were well worth the headaches and broken gears.