

321 GO!

Servo Motor Stopwatch & Clicker Game

Brandon Maruri

Tolga Kaya

ENGR 200 - Computational Methods in Engineering



Abstract

In this project, I created a stopwatch using a servo motor and implemented a clicker game that counts how many times the player is able to click a pushbutton in 30 seconds. The servo motor stopwatch uses a for loop that positions the servo motor at 180 degrees (30 second mark), and ticks down every second until the position is 0 degrees (0 seconds) There was a conversion code used to maintain the accuracy of each tick. Within this for loop, there is a while loop that counts how many times the pushbutton is pressed each second and at the end, the amount for each second was summed up and displayed to show the user their final score. Prior to each game, there is a “3, 2, 1, Go!” sound effect that is created by a buzzer to indicate the start of the Clicker Game. There is also an alarm sound that indicates the end of the game.

Code

The code for the clicker game was created in the MATLAB program using elements of Arduino. In order to use the Arduino Uno within MATLAB, the first step was to call the Arduino functions and including the port, the type of Arduino board, the libraries, and the servo motor functions. After this, the servo motor needed to be identified by specifying the port that it is used in. In this case, the servo motor is connected to pin D4. The starting position also need to be identified at 1, which is equal to the 180° point on the servo motor (since the servo motor only reads values in between 0 and 1). A conversion is then used to change seconds to its corresponding position. The position is then sent to the servo motor and allows it to be ready to countdown. After a 1.5 second pause, a code is sent to the buzzer that creates a “3, 2, 1, Go!” noise to start the Clicker Game.

From here, an array made up of 30 zeros (for each second) is created to keep track of the player’s score each second. The main for loop is then initiated starting with the increments that the servo motor will move each time it loops. A variable is then set that will keep track of the amount of clicks each second, which will be added to the array after each loop. A while loop is included within the for loop to add a point to the player’s score after each time the push button is pressed. Since a tic toc function is also implemented within this for loop, there is a one second pause between each loop. After the for loop is completed, an alarm sound is played by the buzzer to indicate the end of the game. The player’s final score is then summed up using the array and then displayed on the MATLAB program.



Buzzer

4mm Push Button

Methods and Materials

The materials that were used in the Clicker Game were an SG90 Servo Motor, a 4mm push button, a buzzer, jumper wires, and the Arduino Uno. The first steps were to connect the Servo Motor to the Arduino Uno’s 5V, GND, and Digital Pin 4. From here, the code was then made to allow the Servo Motor to position itself at 180° and tick down for exactly 30 seconds. From here, the buzzer was connected to Digital Pin 5 and GND to create countdown indicating the start of the game and a 2 second alarm indicating the end of the game

After these two aspects of the Clicker Game were fully functional, the button was added to the breadboard to create the clicker part of the game. The push button was connected to Digital Pin 2 and GND in order to let the Arduino know the status of the button when needed. During the game, everytime the pushbutton reads “1” the score goes up by one adds it to the summed final score.

```
clear
a = arduino('/dev/cu.usbmodem14201', 'Uno', 'Libraries', {'Servo'}); % call the arduino
servo_motor = servo(a, 'D4'); % call the servo motor with its corresponding pin

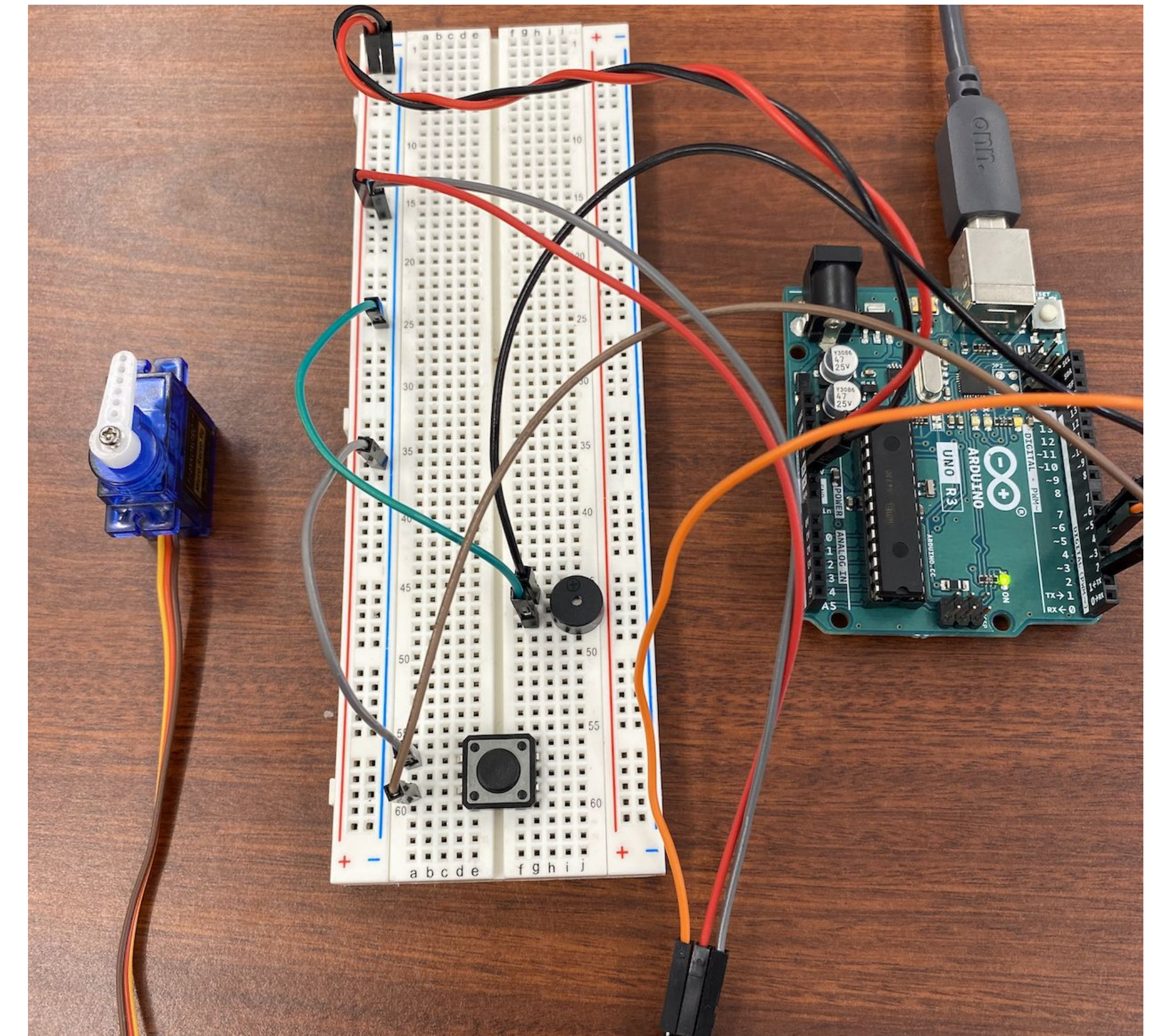
start = 60/60; % define the position of zero seconds
playTone(a, 'D5', 1000, 0.05); %Indicate that the game is about to start
x = 60;
pos = (60-x)/60; % conversion for the seconds to position on servo motor
writePosition(servo_motor, pos); % position the dial to the 30 second position
pause(1.5)

% 3, 2, 1, GO!
pause(1)
playTone(a, 'D5', 1000, 0.15);
pause(1)
playTone(a, 'D5', 1000, 0.15);
pause(1)
playTone(a, 'D5', 1000, 0.15);
pause(1)
playTone(a, 'D5', 1500, 0.4);
pause(1)

clicks= zeros(30,1); % Set a array of 30 zeros
for i = 60:-2:1 % Start the for loop that begins the countdown
    counter = 0; % Create the counter for the amount of clicks
    tic
        writePosition(servo_motor, (60-i)/60); % Position for each second
        while(toc<1)
            y = readDigitalPin(a, "D2"); % Read the pushbutton
            if y == 1
                counter = counter + 1; % Add to the counter when pressed
            end
        end
        toc
        clicks(i) = counter;
    end

playTone(a, 'D5', 1500, 1.5); % Alarm to indicate the end of the game

FinalScore = sum(clicks) % Display the Final Score
```



Results

After the game is completed, the final score is displayed on the command line. Using the information provided by the clicks each second, a graph will be displayed analyzing the average clicks per second throughout the entire game. This will allow for the player to read his/her stats for the game they just played.

Conclusions

In conclusion, this project has allowed me to apply almost every skill that I learned in the ENGR-200 Computational Methods class. Learning how to use MATLAB was an exciting experience and increased my interests in coding overall. The Servo Motor Clicker Game required for me to code a servo motor, buzzer, and push button all together using the knowledge that I gained throughout the school year. Hand- on projects like these are the reason why I decided to study engineering.

Every week was a different challenge and I was always ready to face the challenge head-on. One of the most difficult challenges that I faced this semester was implementing the clicker game at the same time as the stopwatch. The individual functions would not be very difficult to code, but both of them at the same time was quite the struggle. However, as engineer, I was determined to overcome this challenge.

Contact

Brandon Maruri
marurib@mail.sacredheart.edu
Welch College of Business and Technology
Sacred Heart University



Sacred Heart
UNIVERSITY

ENGINEERING