ELEC1601 Exercise 2 Lab Report

SID: 530157791, Lab Group: F12 19

***Abstract*—The purpose of this lab was to delve into digital logic and control systems using an Arduino, a 7-segment display, and servo motors. We executed the simulation and coding on Tinkercad based on the provided tutorial and successfully transitioned the project to a physical setup in our week 5 lab session. This experience equipped me with a foundational understanding of 7-segment displays and servo motors, as well as the skills to manipulate them via Arduino programming.**

I. Introduction

THE purpose of this lab session was to have a deep understanding of basic electronics and microcontroller programming, particularly focusing on digital logic and hardware-software interaction. Key skills targeted for development included:

In this lab, our focus was on teamwork, understanding digital logic, Arduino programming, circuit setup, and control logic. We utilized both software (Arduino IDE and TinkerCAD) and hardware (7-segment display and servo motors) tools to meet these objectives.

- **Software**: Used Arduino IDE for programming and TinkerCAD for virtual prototyping.
- **Hardware**: Employed a 7-segment display and servo motors, connected to an Arduino Uno.
- **Teamwork**: Maintained weekly meetings and code reviews for efficient collaboration.

Our efforts improved our skills in teamwork and circuit designresulting in a successful multi-device setup. Key implementations include button-controlled 7-segment display transitions and servo motor positioning.

II. Background

Key components and their roles and real-world applications are outlined in the table below.

Component	**Lab Use**	**Real-world Applications**
Arduino Uno [1]	Brain of system. Controls display, servos, button.	DIY, automation.
Wires	Connection medium. Connects components.	Circuits, power.
Computer with Arduino IDE [2]	Enables Programming. Codes Arduino.	Software dev.
Resistors	Limits current. Current control.	Circuits.
7-Segment Display	Displays numbers. Shows numbers.	Clocks, signage.
Servo Motors (x2)	Controls position. Shaft positioning.	Robotics.
Button	Switch mechanism. User control.	Appliances.
TinkerCAD	Online testing tool. Prototyping.	Education.

III. Method

A. Part I

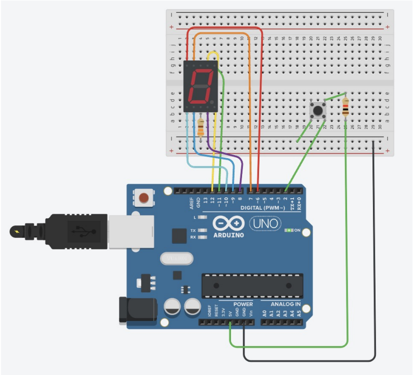

Fig. 1: Circuit diagram for Part I

The circuit we created for Part I, as shown in Figure 1, includes a 7-segment display contains 7 segments. Each segment is controlled independently by pin6 - pin12 on Arduino Uno, corresponding to segments A to G of the display. A resistor of 330-ohms is connected in series with each segment. In addition, a button is connected to pin 2 (interrupt 0). When the button is pressed, the voltage on pin 2 changes from 1 to 0 (FALLING).

The pseudocode in Algorithm 1 controls a 7-segment display with an Arduino Uno.

Algorithm 1 Shortened Pseudocode for 7-Segment Counter

```
Init Booleans $A, B, C, D, E, F, G$, count $= 0$
Configure pins; Attach FALLING interrupt to pin 2 with handler $x$
procedure SETUP
    Init serial at 9600 bps
end procedure
procedure LOOP
    Print pin 2 state; Update $A, B, C, D, E, F, G$ based on count
    Write to pins; Delay 500 ms
end procedure
procedure X
    count $= (\text{count} + 1) \mod 10$
end procedure
```

- **Init:** Boolean variables A, B, C, D, E, F, G and count initialized.
- **Pins:** Pin 2 as INPUT for button; pins 6-12 as OUTPUT for segments $A - G$.
- **Interrupt:** Attached to pin 2 to trigger function x on button press.
- **Loop:**
  - Debugs pin 2 state.
  - Updates $A - G$ based on count.
  - Writes to Arduino pins.
  - 500-ms delay.
- **Function** x**:** Increments count mod 10, updates display.

Design Decisions:

- **Pin Selection:** Pins 6 to 12 were chosen as OUTPUT pins for segments G to A due to their proximity on the Arduino board, facilitating easier wiring. Pin 2 was chosen as the INPUT pin for the button because it supports interrupts, allowing for more efficient button press detection.
- **Use of Interrupts:** Interrupts were used to detect button presses, allowing the program to avoid constantly polling the button's state.
- **Modular Counting:** The variable count was implemented with modular to reset after reaching 9, allowing the display to cycle back to 0.

Observations: The design successfully updates the 7-segment display on button press, utilizing modular arithmetic to back to 0 after 10 presses.

B. Part II

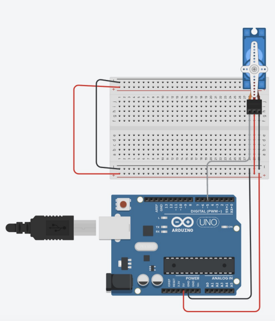

Fig. 2: Circuit diagram for Part II

The circuit we created for Part II, as shown in Figure 2, includes a servo motor connected with Arduino uno.

Algorithm 2 Shortened Pseudocode for Servo Angle Manipulation

```
Include Servo library; Init servo, angle = 0
function SETUP
    Attach servo to pin 5
end function
function LOOP
    for angle from 0 to 150; step 30; wait 300 ms do
        Write angle to servo
    end for
    for angle from 150 to 0; step -30; wait 300 ms do
        Write angle to servo
    end for
end function
```

The pseudocode outlined in Algorithm 2 controls a servo motor connected to pin 5 of the Arduino. The servo's angle increments from 0 to 150 degrees and decrements back to 0, each step separated by a 300-ms delay.

Design Decisions

Angle steps are set at 30 degrees and a 300-ms delay is used for a balance between speed and observation.

Limitations, Successes

Limitation: Servo's 150-degree range despite 180-degree specs.Success: Effective servo angle control.

C. Part III

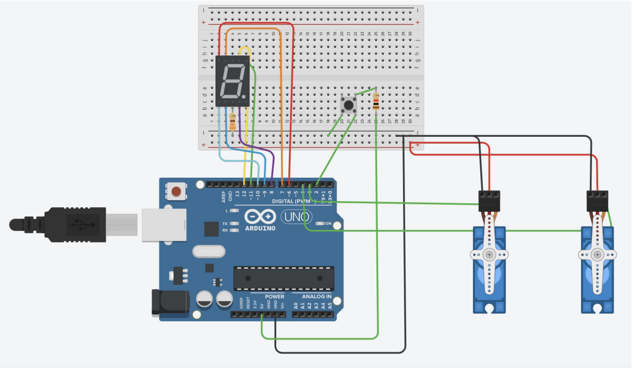

Fig. 3: Circuit diagram for Part III

The circuit we created for Part III, as shown in Figure 3, includes a servo motor connected with Arduino uno.

Algorithm 3 Simplified Pseudocode for Part 3

```
Import Servo; Init servo1, servo2, pos1=0, pos2=180, count=0, A-G
procedure SETUP
    Config pins; Attach interrupts; Attach servos; Init positions and Serial
end procedure
procedure LOOP
    Serial print pin 2; UPDATEDISPLAY; Update servo positions; Delay 500 ms
end procedure
procedure ISR
    $\text{count} = (\text{count} + 1) \mod 10$; Update servos
end procedure
procedure UPDATEDISPLAY
    Update A-G; Write to 7-segment
end procedure
```

The pseudocode outlined in Algorithm 3 controls a servo motor connected to pin 5 of the Arduino. The servo's angle increments from 0 to 150 degrees and decrements back to 0, each step separated by a 300-ms delay.

Design Decisions

Pin Choices: I choose pins 6-12 connect to the 7-segment display for compactness. Pin 2, with interrupt capability, connects to the button for press detection.

Servo Control: Servos connect to pins 3 and 4, which and initialize at 0 and 180 degrees .

Limitations, Successes, and Unexpected Results

Limitations: The 500 ms delay in the main loop makes the system less responsive. Events happening during this half-second might be missed or delayed, causing potential error.

Successes: I successfully implemented the functionality to rotate the servo motor to a angle by pressing a button, and it can also cycle back to the original position.

This part of the lab builds upon part1 of this lab session [3].

IV. Results

A. Successes

- **Functionality**: Circuit and code executed as planned [3].
- **Learning**: Acquired fundamental knowledge in digital logic and Arduino.
- **Teamwork**: Effective communication led to smooth task completion despite initial disagreements.

B. Failures

- **Code Issues**: Undeclared variables and case-sensitive typos hindered progress.
- **Complexity**: Initial circuit design was overly intricate, complicating subsequent coding.

C. Limitations

- **Hardware**: It's difficult to observe the servo motor when it rotates at very small angles.The jitter when pressing the button may lead to unexpected results.
- **Code**: The delay method in loop may cause less responsive of system .

V. Relation to Real-World Electronics

- **Seven-Segment Display**:
  - **Digital Clocks**: Utilized for their low power consumption and clear visibility, making them ideal for battery-operated digital clocks [3].
  - **Elevators**: Efficiently display floor numbers and are easily readable from a distance.
- **Servo Motor**:
  - **Robotic Arms**: Employed in precision tasks such as assembly lines and surgical robots where exact movements are crucial.
  - **Remote-Controlled Cars**: Used to control steering mechanisms, offering a high degree of control over the car's direction.
- **Button**:
  - **Interactive Kiosks**: Often used in public information stations, ATMs, and ticket vending machines to enable user interaction [4].
- **Combined Applications**:
  - **Smart Thermostats**: Incorporate a seven-segment display for temperature readouts, buttons for user control, and servo motors to actuate air vents, offering an integrated climate control solution.

In summary, seven-segment displays, servo motors, and buttons are fundamental technologies with broad applications. From digital clocks to smart thermostats, these components are the building blocks of many devices that are integral to our daily lives.

VI. Conclusion

Our lab aimed to master Arduino-based device control, focusing on seven-segment displays and motors. The session enriched our understanding of hardware-software integration and signal manipulation for motor control. Teamwork was pivotal, enhancing our communication and problem-solving skills. Overall, we met our learning goals.

References

[1] B_E_N, "What is an arduino?." https://learn.sparkfun.com/tutorials/what-is-an-arduino/all. Accessed: 2023-09-05.

[2] K. Söderby and J. Hylén, "Getting started with arduino ide 2.0." https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2, 2023. Last revised: 2023-09-06.

[3] "Elec1601 canvas homepage." https://canvas.sydney.edu.au/courses/52848/pages/weeks-2-and-3-lab-exercise-1-intro-leds-interrupts?module_item_id=2053576n. Accessed: 2023-09-05.

[4] G. Kouroupetroglou, *Universal Access in Public Terminals: Information Kiosks and ATMs*, pp. 1–19. 01 2009.