Lab Report: Project 7 - Keyboard Instrument

Lucas Porter

Spring 2025 TESY 3300

Trevor P. Robinson, PhD

Lab 5 – Mood Cue

2/23/2025

Abstract

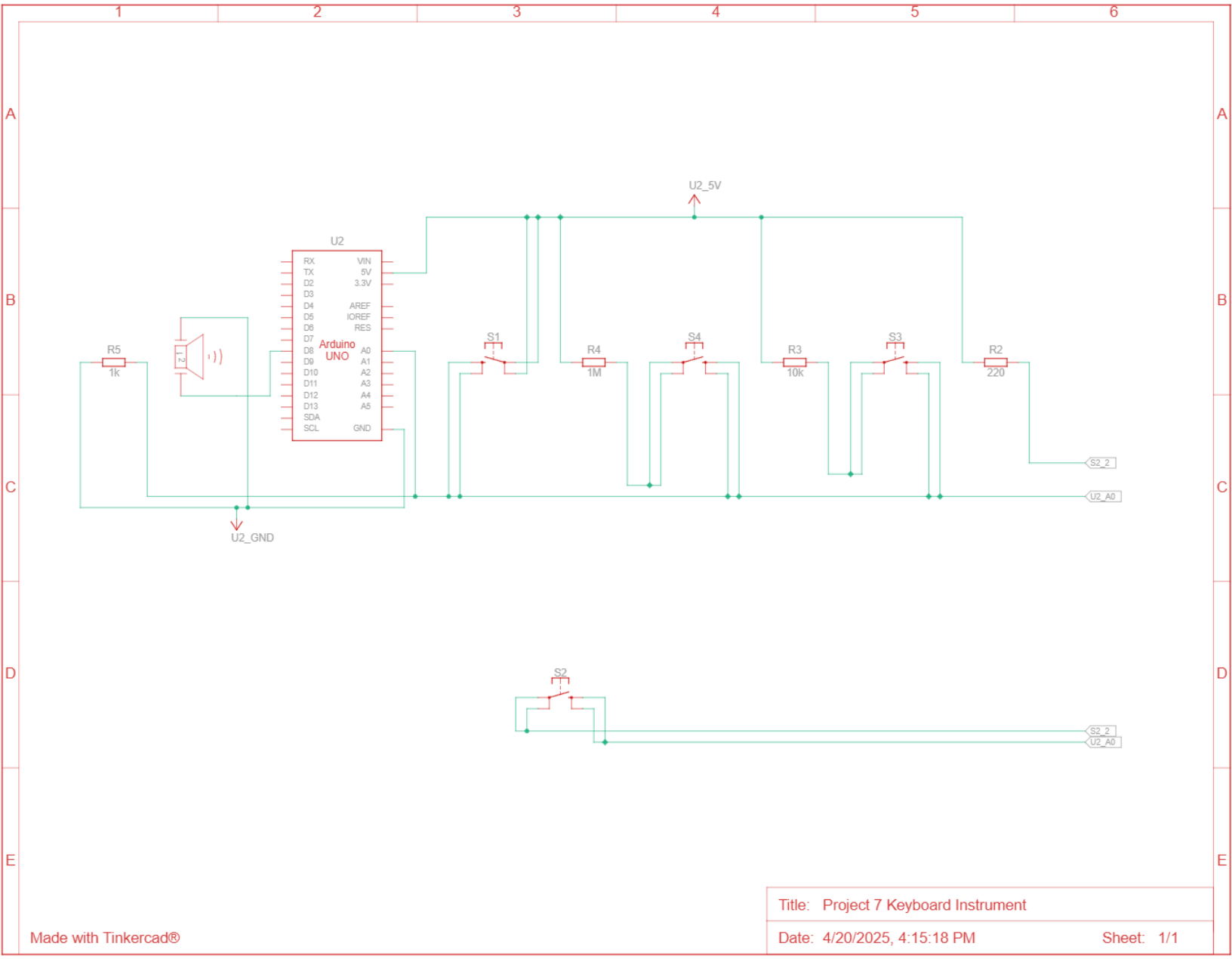
The purpose of this lab was to create a simple keyboard instrument using an Arduino Uno, pushbuttons, and a piezo buzzer. By pressing different buttons, various musical notes were generated through the piezo speaker. This lab introduced concepts of tone generation, timing functions, arrays, and digital input handling. Adjustments to button input ranges allowed for smoother note selection and demonstrated how physical interaction can be translated into digital sound output.

Materials *(15 points)*

- Arduino Uno Board
- Breadboard
- Piezo Buzzer
- 4 Pushbuttons
- 4 10kΩ Resistors
- Jumper Wires
- USB Cable
- Computer with Arduino IDE

Procedure *(25 points)*

Circuit Diagram



Steps

1. Connected the Arduino's 5V and GND pins to the power and ground rails of the breadboard.
2. Placed four pushbuttons on the breadboard, each connected between 5V and a digital input pin.
3. Used 10kΩ pull-down resistors between each button's input pin and ground.
4. Connected the piezo buzzer to a digital output pin and ground.
5. Uploaded the provided Arduino sketch and tested the circuit.
6. Adjusted the code to properly map button presses to specific notes.

Code

p07_Keyboard_Instrument.ino

```
int notes[] = { 262, 294, 330, 349 };

void setup() {

  Serial.begin(9600);
}

void loop() {

  int keyVal = analogRead(A0);

  Serial.println(keyVal);

  if (keyVal == 1023) {

    tone(8, notes[0]);
  } else if (keyVal >= 990 && keyVal <= 1010) {

    tone(8, notes[1]);
  } else if (keyVal >= 505 && keyVal <= 515) {

    tone(8, notes[2]);
  } else if (keyVal >= 5 && keyVal <= 10) {

    tone(8, notes[3]);
  } else {

    noTone(8);
  }
}

```

Discussion (20 points)

1. **Describe how the piezo device works as a speaker.**
   A piezo device vibrates when an electrical signal is applied, producing sound. The frequency of the electrical pulses determines the pitch of the sound.

2. **Compare the analogWrite() and tone() functions.**
   analogWrite() generates a PWM signal to simulate an analog output, useful for dimming LEDs or controlling motor speeds. tone() specifically generates a square wave at a set frequency to drive a piezo speaker and create sound.

3. **Which function returns the number of milliseconds passed since the board was powered on or reset?**
   The millis() function returns the number of milliseconds since the Arduino board was powered on or reset.

4. **How is an unsigned long variable different from an int variable?**
   An unsigned long can store much larger positive numbers (up to about 4 billion) compared to an int, which has a smaller range and can store negative values.

5. **What is an array in Arduino programming? How was it used in this lab?**
   An array is a collection of variables stored under one name and accessed by index. In this lab, arrays were used to store button pin numbers and their corresponding tone frequencies.

6. **For what could a resistor ladder be used?**
   A resistor ladder can divide voltage into different levels, allowing multiple inputs to be read from a single analog pin, useful in applications like multi-button keypads.

Troubleshooting

- **Issue:** Combining key inputs into a new note was tricky.
  Solution: Adjusted the input detection ranges in the code to better handle multiple button presses, ensuring each button triggered the correct note.

Conclusion

The keyboard instrument was very responsive and behaved as desired. Initially, the notes sounded a bit scratchy and jittery, but after making minor adjustments to the input ranges in the code, the notes played much more smoothly. The project demonstrated how button inputs can be effectively used to control a piezo speaker, and how small code tweaks can greatly improve the responsiveness and clarity of the output. Overall, the lab was successful, and no major circuit or coding changes were needed beyond tuning input handling for cleaner sound.