Lab Report: Project 8 – Digital Hourglass

Lucas Porter

Spring 2025 TESY 3300

Trevor P. Robinson, PhD

Lab 8 – Digital Hourglass

03/02/2025

Abstract

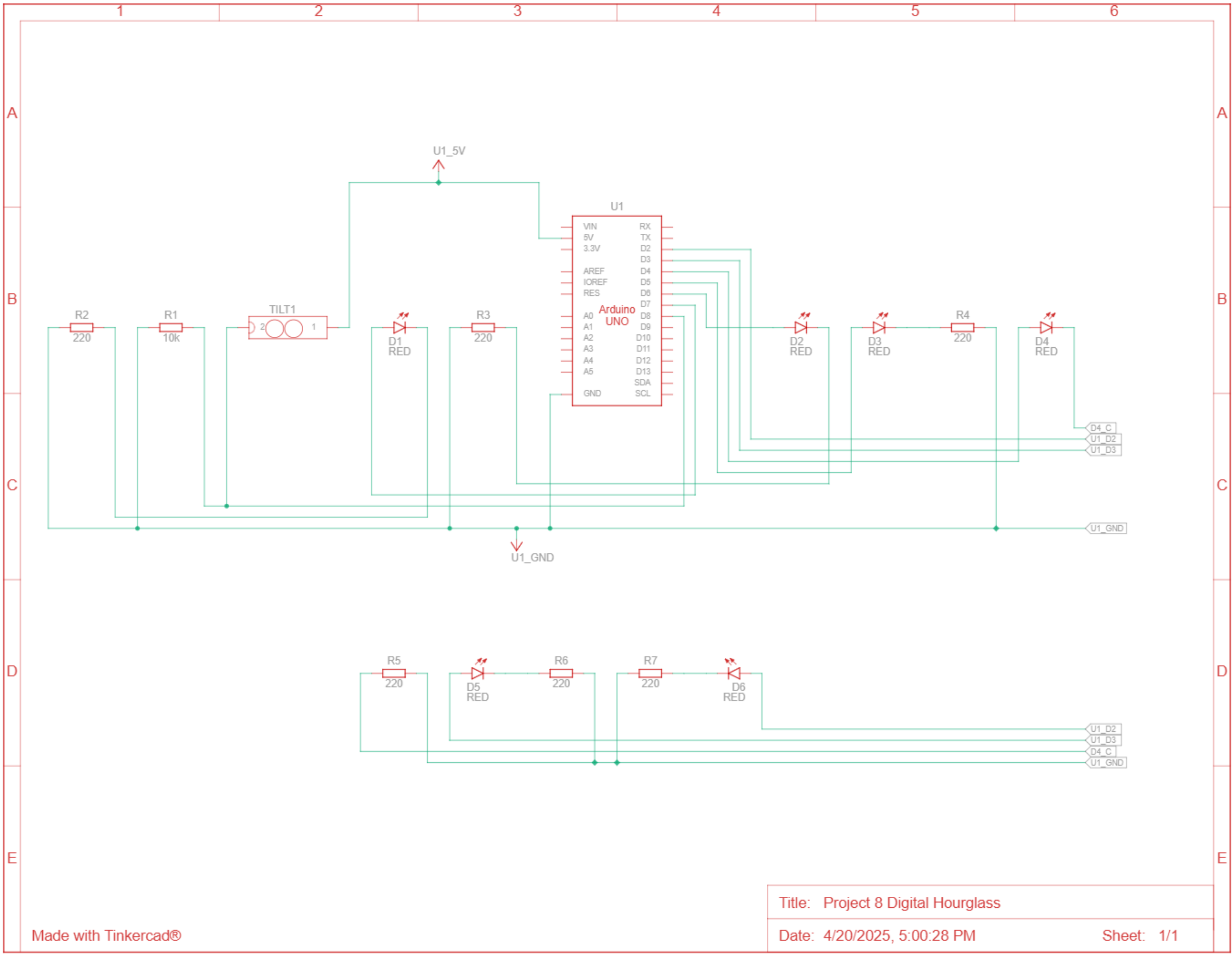
The goal of this lab was to build a digital hourglass using an Arduino Uno, a tilt switch, and a series of LEDs. As the timer progressed, LEDs lit up one by one until all were illuminated. The tilt sensor allowed the user to reset the hourglass by flipping the device over. This project demonstrated the use of time-based programming, digital input handling, and sequential control of outputs. It also reinforced the concepts of using timers and data types suited for timekeeping.

Materials

- Arduino Uno Board
- Breadboard
- Tilt Sensor
- 6 LEDs
- 6 220Ω Resistors
- Jumper Wires
- USB Cable
- Computer with Arduino IDE

Procedure

Circuit Diagram



Steps

1. Connected the Arduino's 5V and GND pins to the power and ground rails of the breadboard.
2. Connected each LED to a digital output pin with a 220Ω resistor to ground.
3. Connected the tilt sensor between ground and a digital input pin with proper pull-up settings.
4. Uploaded the Arduino sketch and tested the circuit.
5. Adjusted the delay time in the code for quicker testing.

Code

```
const int switchPin = 8;

unsigned long previousTime = 0;  // store the last time an LED was updated
int switchState = 0;             // the current switch state
int prevSwitchState = 0;         // the previous switch state
int led = 2;                     // a variable to refer to the LEDs


long interval = 1000;

void setup() {

  for (int x = 2; x < 8; x++) {
    pinMode(x, OUTPUT);
  }

  pinMode(switchPin, INPUT);
}

void loop() {

  unsigned long currentTime = millis();


  if (currentTime - previousTime > interval) {

    previousTime = currentTime;

    digitalWrite(led, HIGH);

    led++;

    if (led == 7) {

    }
  }

  switchState = digitalRead(switchPin);

  if (switchState != prevSwitchState) {
    for (int x = 2; x < 8; x++) {
      digitalWrite(x, LOW);
    }

    led = 2;

    previousTime = currentTime;
  }

  prevSwitchState = switchState;
}
```

Discussion

1. **Describe how long data type is different than the int data type.**
   The long data type can store much larger numbers than the int data type. This is useful for timing functions where large numbers of milliseconds must be tracked.

2. **How long is 480,000 millis?**
   480,000 milliseconds is equal to 8 minutes.

3. **How would you change the program so that you are waiting a shorter amount of time between LEDs turning on, say, make a minute glass instead of an hourglass?**
   To make the hourglass faster, you would reduce the interval time in the code, such as changing the delay to a much smaller number to reflect a 1-minute total instead of 1 hour.

4. **How could you modify the circuit and the program to create a buzzing sound once the hour/minute glass reaches the final time (or all of the LEDs are on)?**
   A piezo buzzer could be added to a digital output pin, and after the final LED lights up, the code could trigger a tone() command to make a buzzing sound.

Troubleshooting

1. **Issue:** The tilt sensor kept popping out of the breadboard due to its short leads, and the directionality was confusing because of its four connections, even though only two were used.
   Solution: Double-checked the tilt sensor wiring against the project instructions and secured the sensor better with additional jumper wires. Shortened the overall delay to 10 seconds for faster testing and debugging.

Conclusion

The project behaved correctly but was initially very slow because it was set up as a one-hour timer. The LEDs lit up in sequence as expected. To speed up testing, I tweaked the code to reduce the time between LEDs turning on to just a few seconds. Overall, the digital hourglass demonstrated how to use timers and event-driven programming to control a sequence of outputs. Flipping the tilt sensor successfully reset the system each time.