UNIVERSITY OF NEVADA LAS VEGAS. DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING LABORATORIES.

Class:	CPE301L Digital Systems Architecture and Design 1001		Semester:	Spring 2025
Points		Document author:	Narek Kalikian	
		Author's email:	kalikn1@unlv.nevada.edu	
		Document topic:	Final Project Report - Reaction Time Game	
Instructor's comments:				

1. Introduction / Theory of Operation

The purpose of my project is to use the ATmega328p AVR microcontroller along with the knowledge gained from this lab to create a fun, interactive game that can measure a player's reaction time. Equipment used with the microcontroller includes:

- Onboard LED
- Onboard switches (buttons)
- Onboard buzzer
- Liquid crystal display (LCD)
- Jumper wires for pin connections

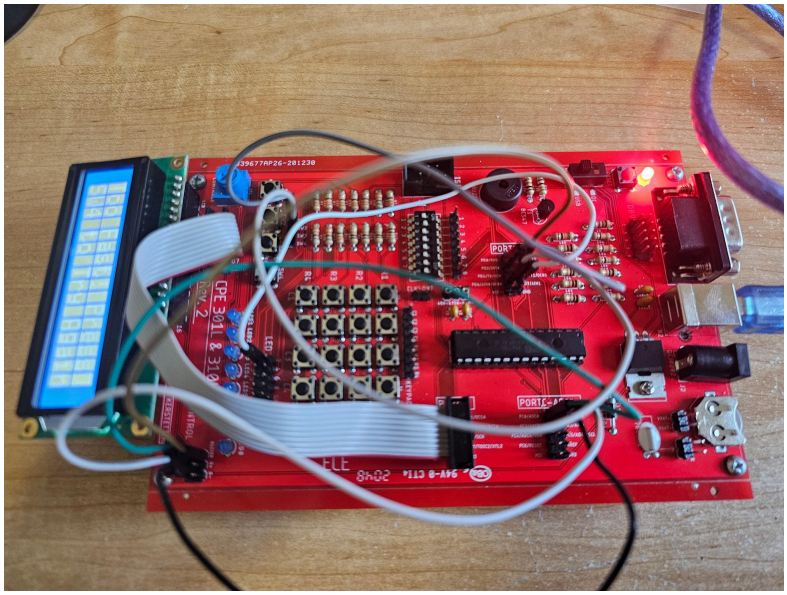
2. Project Description

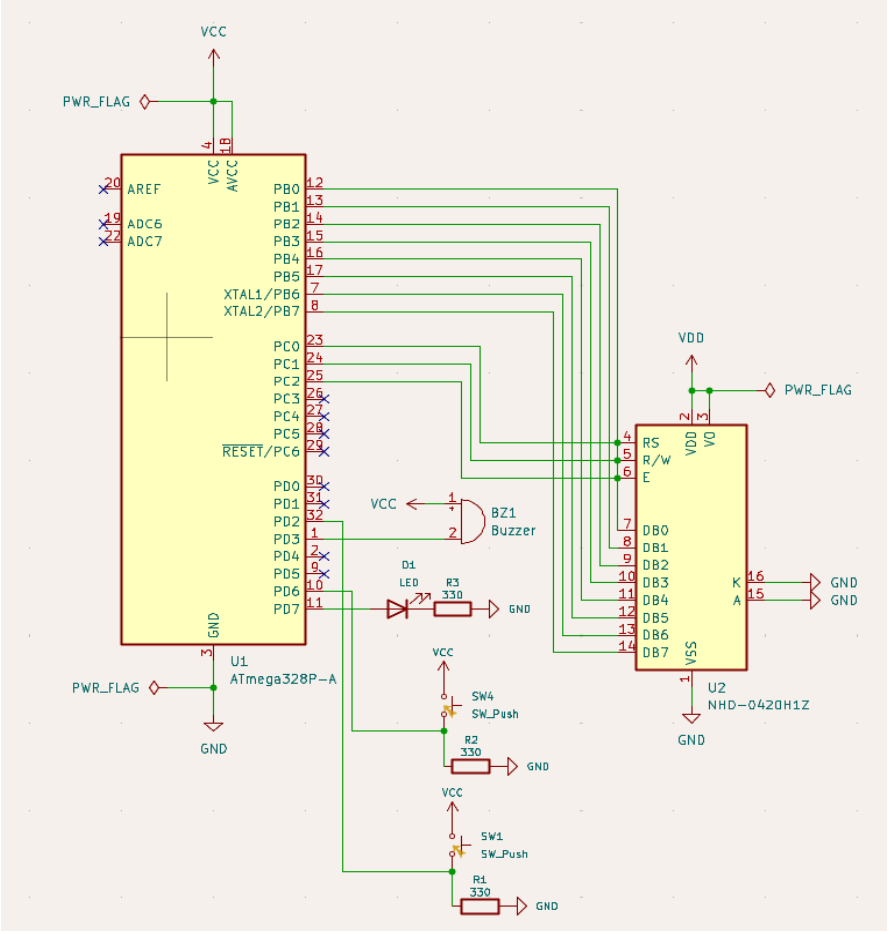
- Inputs:
  - User input button pressed when the LED lights up
  - User input button press to reset the game
  - Random number generator for LED delay (1 to 7 second delay)
  - Clock signal
  - Power
- Outputs:
  - LED (visual cue to react to)
  - Text on LCD
    - Game instructions (Starting state)
    - Reaction time results (Success state)
    - Error messaging (Fail state)
  - Reaction time calculation
  - Error state buzzer

3. Function Breakdown and Circuit Diagram

- An onboard LED lights up at a random interval

- When the LED lights up, the player is to press an onboard button
- The program calculates the time (in ms) between the LED light up and the button press
- This value is displayed on the LCD as the player's score
- The player can then restart the game and try for a higher score
- Error State: If the player presses the button prematurely (before the LED lights up)
    - A message with a restart prompt will be displayed
    - The onboard buzzer will also sound for 1 second to indicate the error state





4. Mathematical Models

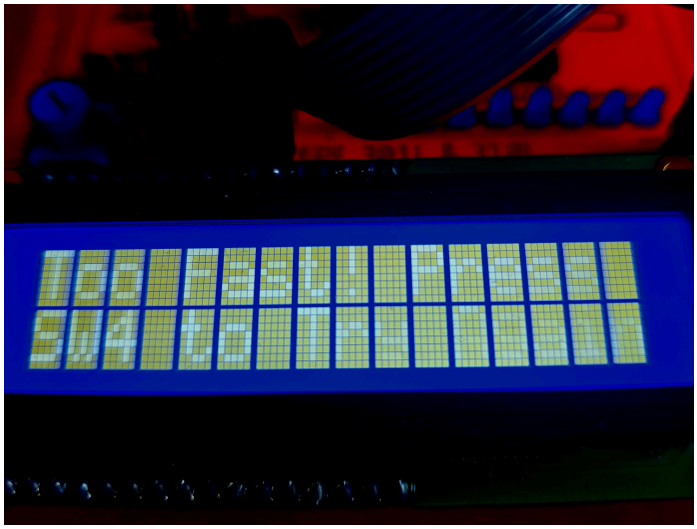
- Delays using 16MHz clock: Each iteration of the game lights the LED up randomly at some time between 1 & 7 seconds
    - 1 second: 1/16 = 0.063 → 63.0 ms delay
    - 7 second: 7/16 = 0.438 → 438 ms delay
- Reaction time calculation using Timer1:
    - start_time variable = TCNT1 immediately after the LED turns on
    - end_time variable = TCNT1 immediately after the button press is detected
    - start_time - end_time = Player reaction time for each game iteration

5. Results - LCD States

Game Instructions:

Reaction Time Results:

Error State Message:



6. Implementation on the Board

Demonstration Video: https://youtu.be/TXO1ly2-fWw

7. Conclusions

The only real problem I encountered was some issues with consistent button press detection. This was due to mechanical button bouncing. I resolved this by adding debouncing using short delays after a button press was detected. This fixed the issue by allowing time for input signals to stabilize, leading to consistent button press detection. My reaction time game project demonstrates successful implementation and use of functions from several of our labs from this semester to create an interactive system:

- Use of timers to measure reaction time
- Use of pulse-width modulation for buzzer control
- Use of LCD to display instructions, reaction time results, and error messages

My project highlights the importance and practical use of input handling, timing, and UI tools for interfacing microcontrollers.

8. References

[1] UNLV, http://eelabs.faculty.unlv.edu/docs/guides/ATMEGA328P_on_breadboard.pdf (accessed Apr. 29, 2025).
[2] NHD-0216K1Z-NSW-BBW-L Newhaven Display | Mouser, https://www.mouser.com/ProductDetail/Newhaven-Display/NHD-0216K1Z-NSW-BBW-L?qs=3vk7fz9CmNwNQ%252BWMZg4kCg%3D%3D (accessed Apr. 29, 2025).