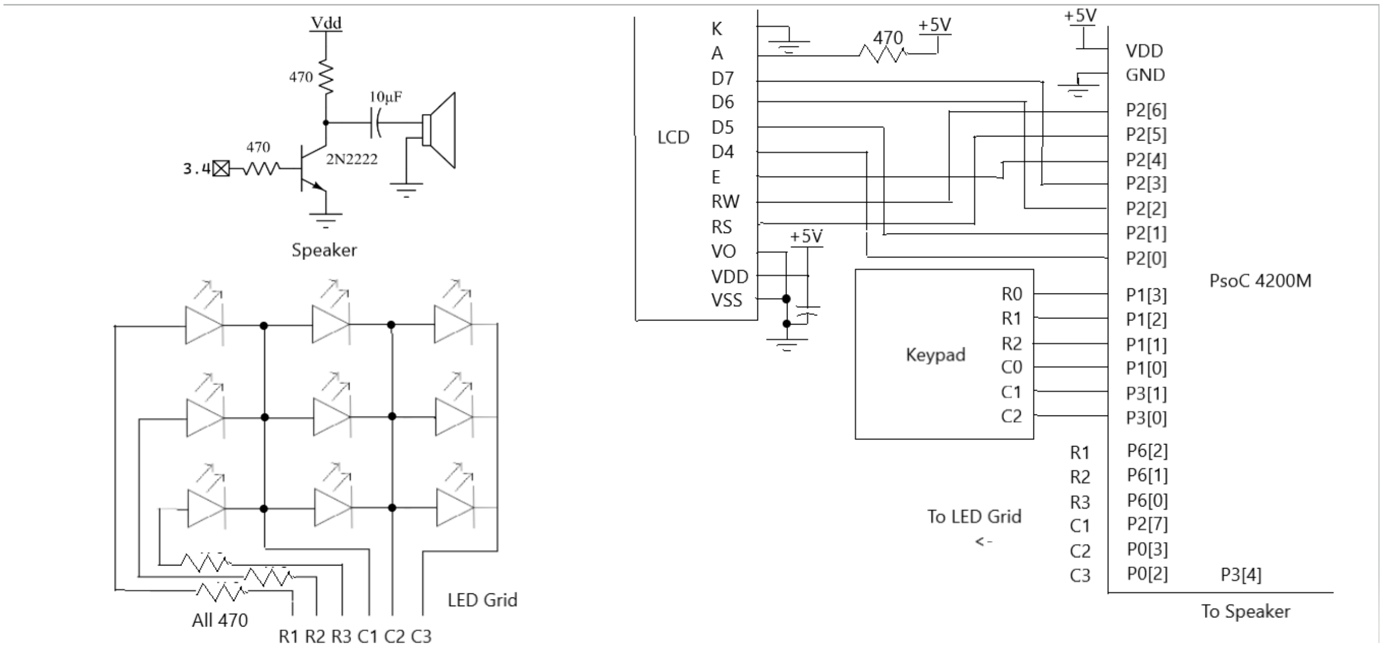
ECE381 Final Project: Whack-A-Mole

Group: Owen Helfrich, Nita Oktavia, Xuewei Bai

Executive Summary:

In our project, we need to design a game like Whack-a-Mole which using the PRS and a PWM to control a grid of LEDs, and the user will press the corresponding button on the keypad when one of the LEDs lights up. The speaker will make a tone when the user presses the correct button, and a different tone when the user presses the incorrect button. When the game is over, the LCD will display the user's score and the highest score in history. Users can play a game for 20 times after the game starts, that is, press the button 20 times according to the LED light, and the game ends after pressing the button 20 times. At this time, users need to restart the game to start playing.

Hardware Description:



Hardware Diagram

Module List:

- Character LCD
- 3 Digital Input Pins (for keypad)
- 9 Digital Output Pins (6 for LEDs, 3 for keypad)
- PRS
- PWM (Timer, LEDs, Speaker)
- Clock
- 3 Interrupts (for each keypad row)

Parts List:

- PSoC 4200M (CY8KIT-044) - $25
- 4x4 Keypad - $1.95
- 9 LEDs - $2.50 for 10
- 2x16 LCD - $34.95
- Speaker - $1.49
- 6 470Ω Resistors - $4.29 for 10
- Breadboard - $7.95
- 22-Gauge Single-Strand Coated Copper Wire - $8.85 for 25 ft.

Algorithm Outline

First, the user presses a button to start the game. The PRS generates 60 random numbers between 0 and 99 which are stored into an array.

Then, the array is read three times – one number is modulated by 9 and used to determine which LED turns on, the next is used to determine the brightness of the LED via the PWM, and the last is multiplied by 50 and used to determine how long the LED is on for, in milliseconds.

The PSoC converts the number corresponding to the LED into the appropriate rows and columns, writing the row pin to 5V and the column pin to ground, while the other row pins are set to ground and the other column pins are set to 5V. This way, the intended LED will be the only one to have 5V on the anode and 0V on the cathode; current will flow through this LED and not the others.

While an LED is on, the PSoC writes the corresponding keypad row pin to VDD and waits for an interrupt from the button in the correct column. If the user presses the correct button in the given timeframe, a score counter is incremented and the speaker plays a success tone. If the user presses any other button, the speaker plays a failure tone. This process repeats 20 times.

Finally, the user's score and the high score are displayed on the LCD screen. The user is given the opportunity to play again by pressing a button.

Timeline:

Week 13 - Top Design completed

Components set up completed

Starting the ISR code

Thursday – update with Dr. York

Week 14 - ISR code completed

Starting the PWM code

Thursday – update with Dr. York

Codes for each module completed

Starting the main.c code

Week 15 - Code in main.c completed

Thursday – update with Dr. York

Trial test for the final project

Preparation for the final project report

Week 16 - Final Projects Demo **(Possibly Tuesday at 12 P.M - 1:40 P.M)**

Final Projects Report **(Tuesday Midnight)**

Final Requirements:

- ❑ The game only starts when the user decides to start playing
- ❑ The game generates 20 random LED positions, each with a random brightness and duration
- ❑ The speaker plays a success or fail tone, depending on whether the user presses the right button at the right time
- ❑ The game keeps track of the user's score, and displays it after the game ends
- ❑ The game allows the user to restart the game without resetting the PSoC
- ❑ The game keeps track of the high score across multiple games