ECE 303 – Week 2 Lab: Codebreaker Project

Introduction:

Imagine you are trying to crack a safe with some 4-digit combination. You have a limited number of attempts to find the code. As you enter a possible combination, you gain information about whether or not any of the numbers input match the safe's code. After you reach the maximum number of guesses, the safe will lock. This week, you will attempt to replicate this process using the Arduino Mega 2560. This lab provides experience in programming Arduino, understanding timers and interrupts, and using input from the serial monitor.

Additional Resources:

- If not familiar with programming an Arduino, review tutorial site (Web page). Start with the **Program Structure** part and continue until you are comfortable.
- Familiarize yourself with Arduino commands (Web page)
- On attaching interrupts (Web page)
- On timer interrupts (Web page)
- On the Serial command (Web page)
- How to write a technical memo (Web page)

Functionality:

The general functionality of the code breaking project should be as follows:

- The program should generate a number between 0000 and 9999.
- Each position of the generated number corresponds to an LED. When the program begins, each LED should be blinking. The LED blinking is controlled by a timer, and should initially be blinking slowly.
- The user enters a four-digit number. That number is broken down into its 4 integers and is compared to the respective number in the same position as the computer-generated number.
- At each iteration (for each position):
    - If the input number is correct, the corresponding LED should turn off.
    - If the input number is incorrect, the corresponding LED should speed up.
    - After 5 tries, if the user has failed to guess the code, all integers of the number which have not been guessed stay solid.

As a simple example, here is how the program should respond if the user has 3 attempts to guess the code. Assume between subsequent iterations, LED blink frequency will increase by 100 Hz for each time that position is guessed incorrectly. HIGH will indicate the LED is constantly on while LOW means it is constantly off.

- Generated Code (what you are trying to guess): 1000-

LED Behavior: Input Guess Attempt 0				
Integer Position	1	2	3	4
User Input Position Value	-	-	-	-
LED Blink Frequency (Hz)	100	100	100	100

- User Input (Attempt 1): 1234-

LED Behavior: Input Guess Attempt 0				
Integer Position	1	2	3	4
User Input Position Value	-	-	-	-
LED Blink Frequency (Hz)	LOW	200	200	200

- User Input (Attempt 2): 1709-

LED Behavior: Input Guess Attempt 0				
Integer Position	1	2	3	4
User Input Position Value	-	-	-	-
LED Blink Frequency (Hz)	LOW	300	LOW	300

- User Input (Attempt 3) (Final Attempt): 1500-

LED Behavior: Input Guess Attempt 0				
Integer Position	1	2	3	4
User Input Position Value	-	-	-	-
LED Blink Frequency (Hz)	LOW	HIGH	LOW	LOW

Breadboarded Circuit:

Four LEDs are needed for each integer position. Each LED is connected in series with a resistor to limit the LED current (R=1kΩ will work). Each LED is connected to a digital output pin corresponding to a different timer. Keep track of which digital pins are being used. Each timer is associated with 2 or 3 pins. You do not want multiple LEDs connected to the same timer.

Sketch Requirements:

Below are the main components you need to include in your sketch. There are a number of ways of programming this; do whichever you find best.

- Method of generating an initial combination (the number you are trying to guess between 0000 and 9999)
- Read input from the Serial Monitor to be used in your program
    - Remember to properly set the baud rate.
- Four timers, one corresponding to each LED
    - You have both 8-bit and 16-bit timers available. We would like you to use at least one of each.
- Four interrupts (output compare)
    - You can use the output compare interrupt to toggle the LED status. By changing the OCR value of each timer, you can change the frequency of the signal entering the timer counter register. This will allow you to change the blink frequency of each LED.
- Method of comparing each position of the input value with the computer-generated number. This will be used to determine what change will be made to each LED.

Deliverables:

Write a technical memo summarizing the lab. Given this lab is not having you collect data, to verify you were able to complete the lab, we ask that you submit the following as well:

- Append your Arduino sketch to the technical memo
- Provide a brief video of your functional lab. You should input guesses in the serial monitor, show your LED response, and show what happens when you reach the maximum number of attempts.