Lab 4 – LCDs and Accelerometers

Introduction:

In this lab, will learn how to use a LCD (liquid crystal display) screen for text display and other interesting applications. You will first create a timer that shows a countdown and sounds a buzzer after a certain amount of time has elapsed. You will then change the pitch of a buzzer by tilting an accelerometer, while displaying the frequency of the pitch on the LCD screen. Finally, you will use the accelerometer and LCD screen to simulate a spirit-level indicator.

Parts Required:

1. Arduino Board
2. USB Cable
3. Sound Buzzer
4. 16X2 LCD Display
5. ADXL335, 3-axis Accelerometer
6. Wires

Self-Timer using the Arduino

Procedure:

a. Building the circuit

Attach the LCD screen shield to the Arduino, and place the breadboard shield on top, as shown in Figure 1. Build the sound buzzer circuit as described in Figure 2.
b. Compile and upload the following code to the Arduino Board using the Arduino IDE.

```c
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 13, 10, 9, 8);
int runTimer = 1; // true condition for timer
int serialData = 0; // false condition for serial communication
int speakerPin = 6;
int data = 0; // default condition

void setup() {
  pinMode(speakerPin, OUTPUT);
  // set up the LCD's number of rows and columns:
  lcd.begin(16, 2);
}

void loop() {
  // To execute timer only once
  if(runTimer == 1){
    // Print a message to the LCD.
    lcd.clear();
    lcd.print("Timer: ");
    //Start timer
    timer(); // runs the timer code below, under void timer()
  }
}
runTimer = 0;
lcd.noDisplay();
delay(250);

// Sound Buzzer
for(int duration = 0; duration < 100; duration ++)
{
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(2000);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(2000);
}
lcd.display();
delay(250);
}

void timer(){
    // For loop to run the COUNT-DOWN in Seconds
    for(int timer = 10; timer > 0; --timer)
    {
        // Set the Cursor to the space after the display "TIMER: ">
        if(timer >= 10)
        {
            lcd.setCursor(6,0);
        }
        else
        {
            lcd.setCursor(6,0);
            lcd.print('0');
            lcd.setCursor(7,0);
        }
        // Display the COUNT-DOWN Seconds
        lcd.print(timer);
        lcd.print("s");
        delay(1000);
    }
}
c. Self-Timer and Sound Buzzer

Press the RESET Button of the Arduino board, the timer will countdown from 10 seconds, as programmed. Once the timer countdown reaches 0s, the buzzer will go on and the LCD display will blink “Buzzer!”

The program is reset every time you press the RESET Button of the Arduino board and the timer countdown begins again.

d. Questions

i. Can you make the timer countdown from 100 seconds?

ii. Can you make the buzzer buzz only for a fixed amount of time (e.g. 5 seconds) after the timer countdown reaches 0s? **Show a TA!**

Possibly helpful reference links:


**Pitch-Control Using an Accelerometer and the Arduino**

Procedure:

a. 3-Axis Accelerometer

An accelerometer is used to measure the acceleration experienced by an object. The ADXL335 (two different versions shown in Figure 3) is adopted to measure the acceleration experienced by the object in motion with respect to the X or Y or Z axis. We will only be measuring movements to the Y axis in this lab.
b. Interface the 3-Axis Accelerometer

Connect the 3-Axis accelerometer to the Arduino shield as shown in Figure 4. Keep your buzzer circuit from the previous part connected.

c. Copy the following code into the Arduino IDE.

The code below will work correctly for the red Sparkfun accelerometers connected as shown in Figure 4. If you are using a blue or green ADXL335, you must make sure the groundpin, powerpin, and ypin values match the corresponding pins on the accelerometer. Note that analog I/O pins 0-5 are referred to as pins 14-19 (to distinguish them from the digital I/O pins).

```c
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 13, 10, 9, 8);
const int groundpin = 18;       // analog input pin4 -- ground (GND)
const int powerpin = 19;        // analog input pin5 -- voltage (Vcc)
const int ypin = 16;            // analog input pin2 -- y-axis pin
const int buzzerPin = 6;        // digital input pin6 -- buzzer
```
int yvalue;
int freq;
int ymin = 0;   // Replace with measured minimum yvalue
int ymax = 1;   // Replace with measured maximum yvalue

void setup() {
  pinMode(ypin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(groundpin, OUTPUT);
  pinMode(powerpin, OUTPUT);
  digitalWrite(groundpin, LOW); // make analog pin (4) equivalent to GND (ground)
  digitalWrite(powerpin, HIGH); // make analog pin (5) equivalent to Vcc (voltage source)
  // set up the LCD's number of rows and columns:
  lcd.begin(16, 2);
}

void loop() {
  yvalue = analogRead(ypin);
  freq = map(yvalue, ymin, ymax, 100, 10000);   // maps yvalue into frequency range
  tone(buzzerPin, freq);                  // sounds buzzer at given frequency
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Freq: ");
  lcd.print(freq);
  lcd.print(" Hz");
  lcd.setCursor(0,1);
  lcd.print("yvalue: ");
  lcd.print(yvalue);
  delay(150);
}
**d. Find the range of the y-axis acceleration**

Find the minimum and maximum values of the y-axis acceleration \((y_{value})\) experimentally by tilting the board about the y-axis and looking at the values on the LCD. Replace the values of \(y_{min}\) and \(y_{max}\) with the values you found. Upload the code to the Arduino and tilt the board along the y-axis to observe its effect.

The map function \(\text{map}(\text{value}, \text{fromLow}, \text{fromHigh}, \text{toLow}, \text{toHigh})\) maps a value of \(\text{fromLow}\) to \(\text{toLow}\), a value of \(\text{fromHigh}\) to \(\text{toHigh}\), values in-between to values in-between, etc. In this case, \(y_{value}\) is converted from an acceleration value to a frequency value. To get more information about any function, right-click the function name and click “Find in Reference.”

**e. Questions**

1. If \(y_{value}\) is 450, then what is the value of \(freq\)?
2. How would you change the range of frequencies that is swept through?
3. Can you modify the code so that change of pitch sounds more continuous?
4. Change the code so that the change in frequency correlates with acceleration on the x-axis instead of the y-axis? **Show a TA!**

---

**Spirit-Level Indicator using the Arduino**

**Procedure:**

a. Compile and download the working code to the Arduino Board.

```c
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 13, 10, 9, 8);

const int groundpin = 18;   // analog input pin0 -- ground
const int powerpin = 19;    // analog input pin4 -- voltage
const int ypin = 16;        // y-axis of the accelerometer

void setup() {

  lcd.begin(16, 2);
  Serial.begin(9600);
  pinMode(ypin, INPUT);
  pinMode(groundpin, OUTPUT);
  pinMode(powerpin, OUTPUT);
  digitalWrite(groundpin, LOW);
  digitalWrite(powerpin, HIGH);
```
void loop() {
    int avalue = 0;
    int lcd_Cursor_Position = 0;
    lcd.clear();
    avalue = analogRead(ypin); // read value of the X-axis acceleration
    lcd_Cursor_Position = 46 - avalue/13; // calculation to position the lcd cursor
    Serial.print(avalue); // prints x-axis acceleration in serial monitor
    lcd.setCursor((15 - lcd_Cursor_Position), 1);
    lcd.print('.');
    lcd.setCursor((15 - lcd_Cursor_Position), 0);
    lcd.print('.');
    delay(100);
}

b. Questions:

  v. Can you display the spirit level using other characters apart from '.'?
  vi. Can you make the spirit level indicator '.' move in the direction of the acceleration?
  vii. Can you make one spirit level indicator '.' move in opposite direction to the other?
  viii. Extra Credit: Using a sound buzzer, play any musical note, if the level indicator is stationary in a particular position for more than 10 seconds. Show a TA!