RSA Lab #4: OLED Screen and Ultrasonic Distance Sensor

Part A: Off to the Races Again

For this part of the lab assignment I was able to measure the frequency and period of an LED blinking as fast as possible (while being able to see it blink).

1. Now, make a new program to use the Arduino's LED and resistor which are built-in, internally connected to digital pin 13. Print your program, commented, and a scope plot; annotate the estimated period, calculate the frequency from this period, and compare it with the measured frequency.

   See attached code and scope. The frequency from the oscilloscope was measured as $f = 20.9\ Hz$. Thus, the period for this waveform would be the inverse, or $T = 47.8\ ms$. This is pretty close to the period specified in the program, which was 50 ms.

lab4a

```
const int pinNum = 13;  // Digital pin where output is being measured
const int period = 50;  // Period
unsigned long time;  // Time placeholder

// The setup routine runs once when you press reset:
void setup() {
  Serial.begin(9600);  // Setup serial monitor
  pinMode(pinNum, OUTPUT);  // Set up LED
}


void loop() {
  digitalWrite(pinNum, HIGH);  // LED turns on
  time = millis();
  while (millis() < time + 0.5*period) {}  // Wait half a period for the next state change
  digitalWrite(pinNum, LOW);  // LED turns off
  time = millis();
  while (millis() < time + 0.5*period) {}  // One full period has been completed
}
```

Part B: Getting Started with an OLED Display via SPI Serial Communication

For this section of the lab, I looked into the various libraries related to the OLED screen and was able to use syntax from those libraries to print a simple message: "howdy".

2. Take a look at the Adafruit_SSD1306.h library: use TextEdit (Mac OS) or WordPad (Windows) or similar to open the file called Adafruit_SSD1306.h, which is in the Documents/Arduino/libraries/Adafruit_SSD1306 folder. What is the name of the author of this library?

According to the comments in the *.h* file, this library was Written by Limor Fried/Ladyada for Adafruit Industries.

Part C: Getting Started with Acoustic Range Measurement

For the last section of the lab, I was able to derive the distance of an object in front of an ultrasonic distance sensor taken from several points. This was done by integrating the sensor with the Arduino and measuring the round-trip period of the signal, which I used (along with the speed of sound in air) to find the approximate distance.

3. Find the speed of sound in air. Cite your source.

   According to NASA, the speed of sound in air on a standard day is approximately 761 mph (1).
   (1): https://www.grc.nasa.gov/www/k-12/airplane/sound.html#:~:text=If%20we%20consider%20the%20atmosphere,%2C%20or%201100%20feet%2Fsecond.

4.
   a. Find the analytical relationship you expect between range (in m) and two-way time-of-flight (in s), in terms of the local speed of sound in air (c).

      First, I converted the speed of sound from mph to m/s (340.197 m/s). Then I used the relationship between distance and velocity to derive the following equation:

      $$\Delta x = 340\left(t_{single\ trip}\right) = 170(t_{round\ trip})$$

   b. Now rewrite your equation in this form: $\Delta x\ (in\ cm)\ =\ m\ \Delta t(roundtrip, in\ \mu s)$, where Δx in cm is the distance in cm, ΔtROUNDTRIP in µs is the time in microseconds, and m is the constant of proportionality.

      This task is relatively simple to implement, as we just need to change meters into centimeters and seconds into microseconds:

      $$\Delta x\ [m] = 170(t_{round\ trip}\ [s])$$
      $$\frac{1}{100} \times \Delta x\ [cm] = 170(1000000 \times t_{round\ trip}\ [\mu s])$$
      $$\Delta x\ [cm] = (1.7 \times 10^{10})(t_{round\ trip}\ [\mu s])$$

5. Connect the HC-SR04 ultrasonic sensor to the Arduino (unplug your Arduino first!). Include a circuit schematic in your lab writeup.

   See attached circuit. I simply copied the breadboard arrangement I had made along with some tweaks to the layout to make it more readable.

6. Write a program to activate the ultrasonic sensor once per second. Print and annotate one scope plot; label and estimate: t(OUT) (the duration of the pulse sent from the Arduino to the sensor), t(HOLDOFF) (the delay between the two pulses), and Δt (ROUNDTRIP).

   See attached scope plot.

7. Now, extend your program to measure the range using the ultrasonic sensor, and display the distance in centimeters. Make sure your program is commented clearly, and submit a copy of your program

   See attached commented code.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_MOSI 9
#define OLED_CLK 10
#define OLED_DC 11
#define OLED_CS 12
#define OLED_RESET 13
Adafruit_SSD1306 display(OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);

/*
const int pinNum = 13;  // Digital pin where output is being measured
const int period = 50;  // Period
unsigned long time;  // Time placeholder
*/
const int trig = 2;
const int ech = 3;
float time1, time2;
float dist;
unsigned long time;  // Time placeholder
// The setup routine runs once when you press reset:
void setup() {
  pinMode(trig, OUTPUT);  // Set up trigger
  pinMode(ech, INPUT);  // Set up echo
  display.begin(SSD1306_SWITCHCAPVCC);
  display.display();
  delay(2000);
  display.clearDisplay();
  display.setTextSize(2); // sets text size; 1 is small, 2 is medium, and 3 is large.
  display.setTextColor(WHITE);
  display.print("howdy");
  display.display();
  delay(2000); // Pause for 2 seconds
  display.clearDisplay();
}
```

```
void loop() {
  digitalWrite(trig, LOW); // Sets trigger low
  delayMicroseconds(10);
  digitalWrite(trig, HIGH); // Sets trigger high after a delay
  delayMicroseconds(10);
  digitalWrite(trig, LOW); // Sets trigger low
  while (digitalRead(ech) == LOW) {} // Wait until echo receives a signal
  time1 = micros();
  while (digitalRead(ech) == HIGH) { // Wait until signal is finished
   if ((micros() - time1) > 24000) { // If the signal time is greater than 24,000us, end the loop
    break;
   }
  }
  time2 = micros();
  dist = 1.7*1e10*(time2-time1); // Calculate the distance from the reflective material
                                 // based on the time of the signal's round trip and the
                                 // speed of sound in air.
  display.clearDisplay(); // Clear the screen
  display.print(String(dist, 2)); // Print the distance
  display.display(); // Display the printed statement on the OLED screen
}
```