University of California, Davis
Electrical and Computer Engineering

EEC 1/89D

Introduction to Electrical and Computer Engineering

Part 3: Sensors, Signals, and Systems

Optical Audio Transmitter and Receiver and Amplifiers

Lab 4

Introduction:

a. Purpose/ Scope

We live in the information age but how exactly does one define information? Although the concept of information is abstract, a practical definition would be the energetic representation of a given system. Consider the scenario where you just heated up a cup of coffee to study for your exams. Before taking a big drink, you need to see how hot it is, so you take a quick sip. In that sip, energy is transferred in the form of heat from the coffee into your skin, or more accurately your nerves. Your nerves transport this energy in the form of electro-chemical energy that is interpreted in your brain which tells you to stop drinking the coffee so you do not burn yourself. One of the cornerstones of electrical engineering is the extraction, manipulation, and transmission of information in a very similar manner.

With this in mind, this lab emphasizes a topic that we have seen in previous labs but haven't explored in great detail, namely transducers and their application to electrical engineering. Transducers are devices that convert one form of energy into another, or based on our previous statement, takes one form of information, and converts it into another. This can be the conversion of mechanical energy in the form of pressure or sound into electricity or light into electricity and vice versa. This lab aims to introduce you to different sensors/transducers, some you have seen before and some you have not, and explore their dependencies (operating conditions), limitations, and applicability to engineering devices. Although the systems you will build are simple, they are eloquent in their simplicity and are the basis to a wide variety of devices.

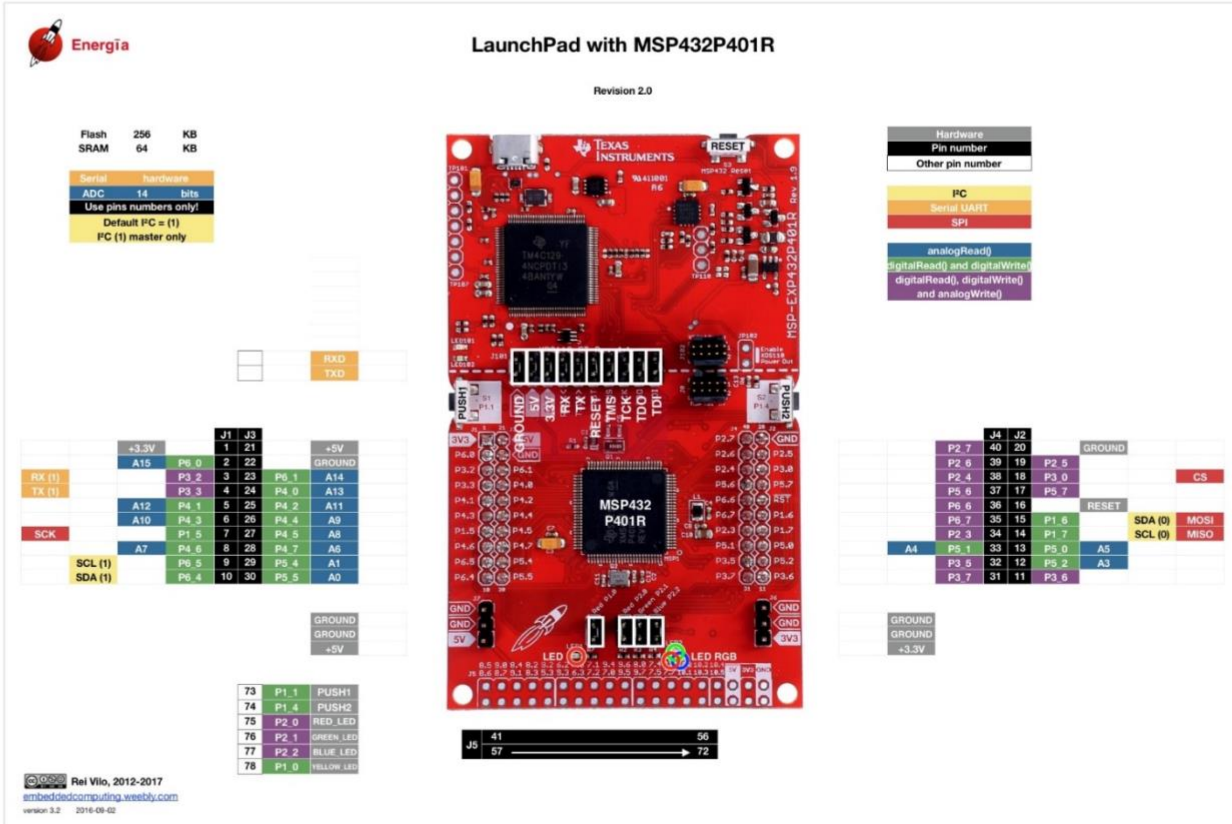
b. Resources/Lab Materials

Lab Materials:

- breadboard x2
- MSP432 MCU
- Ruler
- 555 timer x1
- LM35DZ temperature sensor x1
- photoresistor x1
- red LED x1
- green LED x1
- passive buzzer x1
- pushbuttons x4
- resistors (100 x2, 330 x2, 1k x3, 10k x2)
- capacitors (0.01 μF x1, 0.1 μF x1, 10 μF x1, 100 μF x1)
- wires

Energia Files:

- Lab4_buzzer_demo.ino



Pin layout of MSP432

Lab Exercises:

Part A: Sensors for process control

Exercise 1: Temperature sensor

Description: In this exercise you will be using your microcontroller and a temperature sensor in a very simple circuit to toggle an LED on after the temperature exceeds a certain threshold, and toggle it off once it falls beneath that threshold. While turning on a LED is not very exciting, the LED can be easily replaced with some other process like turning on a sprinkler or sounding an alarm because there is a tremendous amount of heat from a fire. This exercise will provide an introduction to sensors and how they can control processes and provide information from the environment to be used when designing problem solutions.

Procedure:

1. First, construct the circuit shown below in Figure 1. The three terminal device shown is a LM35DZ temperature sensor and its pin layout is shown in Figure 1B. **WARNING:** the pin layout is if you are looking at the device from the bottom with the pins pointing at you. If you connect it backwards it will not work. You will know quickly because it will smell like something is burning and the device will get very hot quickly.

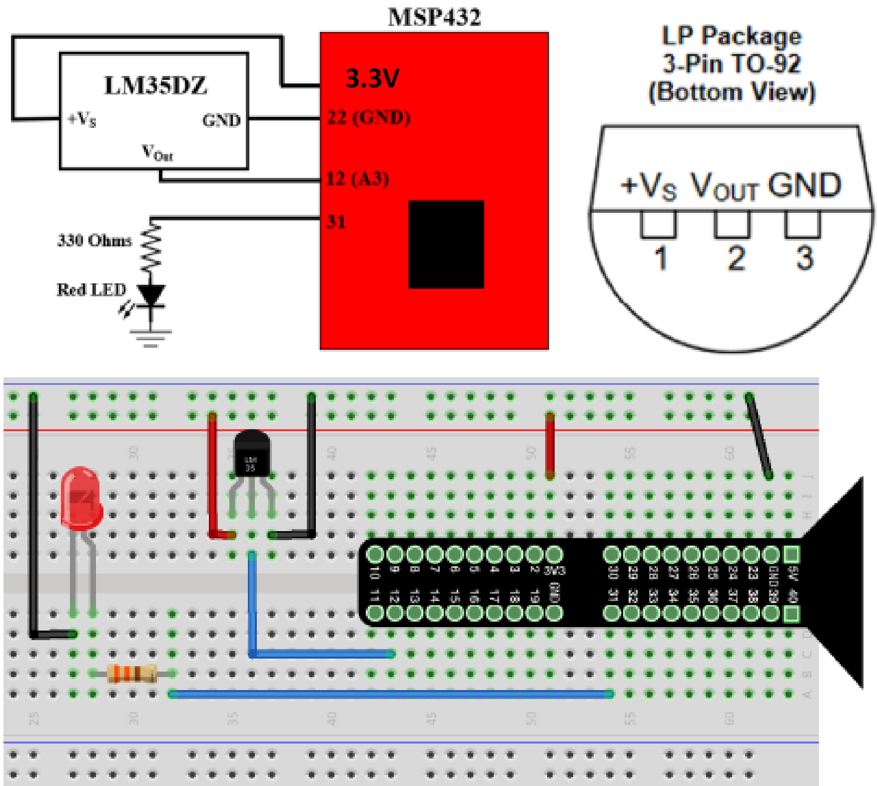


Figure 1. (A) Schematic of temperature sensor circuit.
(B) Pin layout for LM35DZ temperature sensor. **(C)** Diagram of circuit.

2. Next, copy the code segment below into Energia:

```
const int ledPin = 31;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  int sensorValue = analogRead(A3);
  float temp_C = ((3.3*sensorValue/1023.0))/0.01;
  float temp_F = temp_C*(9.0/5.0)+32.0;
  Serial.println(temp_F);
  delay(1000);
}
```

Take a moment to look at the code an understand it as there will be some brief questions on it for this exercise. Run the code and open the serial monitor. You should see the ambient temperature printing to screen. **Note**: this is the first time we have seen the analogRead() function, it reads an analog voltage from a pin on the Launchpad and returns a number from 0 to 1023 that must be converted to represent a voltage using some math.

3. Now modify the Energia script to include the following lines and upload the program to the MCU:

```
const int ledPin = 31;
const float threshold = 80.0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  int sensorValue = analogRead(A3);
  float temp_C = ((3.3*sensorValue/1023.0))/0.01;
  float temp_F = temp_C*(9.0/5.0)+32.0;
  Serial.println(temp_F);
  delay(1000);
  if (temp_F >= threshold){
    digitalWrite(ledPin,HIGH);
  }
  else{
    digitalWrite(ledPin,LOW);
  }
}
```

Fig. 2. Segment of code to edit in Energia

4. Now place your hand near or gently on the temperature sensor. After a short time the LED should turn on. Once you have verified that the circuit is working correctly, answer the following questions.

Questions: Submit answers to the following questions on Canvas.

1. Refer to the variable named temp_C in the Energia code. At this point you should be familiar with this line of code, however, this time there is an added factor of 0.01 in the expression. Explain why this factor is included.

2. In this code, we defined a variable threshold. What is the purpose of this value and what would happen if we increased or decreased it?

<u>Part B:</u> Transducers and Signals

Exercise 2: Light to electricity

Description: In the last lab, you worked with a combination of an LED and a photoresistor. In this exercise you will do the same except you will systematically analyze the dependencies of the photoresistor on separation distance and color of the light source. These aspects are common considerations when it comes to system design.

Procedure:

For this exercise you will need both of your bread boards and a ruler.

1. Construct the circuit shown below using two separate bread boards. Note: the photoresistor will be operated using the 3.3 V supply from the MCU while the LED will be using the 5 V from the MCU using jumper wires between the boards. The wires should be long as you will be moving the bread boards apart for your measurements. The long leg of the LED is connected to the power supply and the short leg is connected to the resistor.

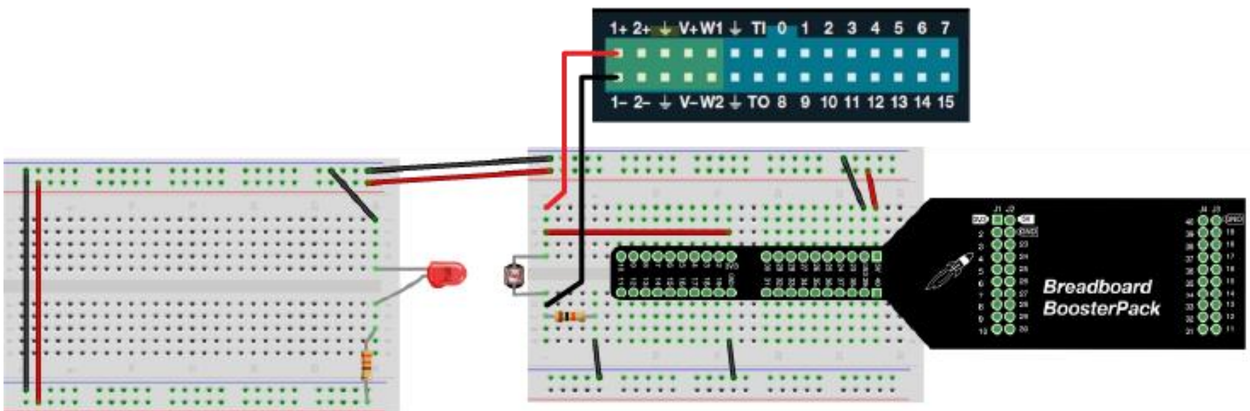


Figure 3. Diagram of circuit for LED and photoresistor, measuring the voltage across the photoresistor with the ADALM voltmeter.

2. Just like in the previous lab when we worked with photodiodes, we want the LED to be focused on the transducer. To do this tilt the LED and photoresistor so that they are pointing at each other and use the voltmeter in Scopy to record the voltage across the photoresistor. A depiction of the LED, photoresistor configuration can be seen below in Figure 4.

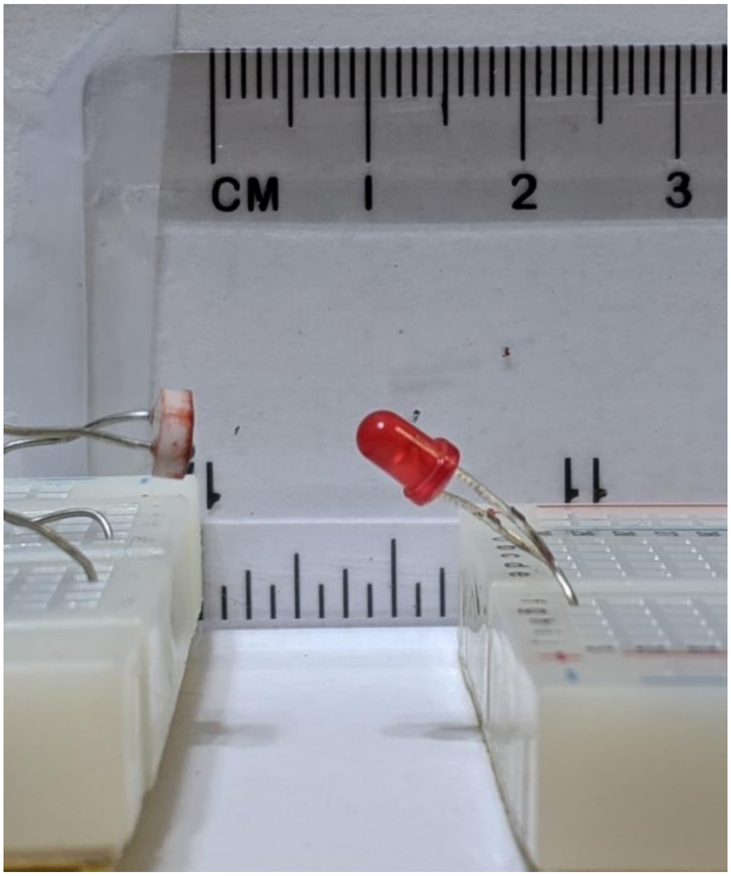


Fig. 4. Image of LED and photoresistor configured to point at each other and maximize the signal. The separation distance in this image is estimated to be 1 cm.

Use the ruler to measure the separation distance between the tip of the LED to the face of the photoresistor. Start with the LED resting on the face of the photoresistor and then slowly move the breadboards away from each other and record the voltage and distance at each separation. Start with the LED resting on the photoresistor and record the voltage every 0.5 cm until the LED and photoresistor are separated by 5 cm. **IMPORTANT:** the ambient light will noticeably affect the photoresistor so you should do this step in the dark or place a dark cloth or box over the circuit to decrease the ambient light.

3. Repeat this process over the same range but this time with a green LED and answer the questions that follow.

Questions and Submission: Answer the following questions on Canvas and submit the required graph as a .pdf file. **ONLY .pdf files will be accepted.**

1. Prepare a graph in excel or any other software you are comfortable with, showing distance on the x-axis and voltage on the y-axis for the both LEDs. Both curves should be on the same graph for comparison. The units for the x-axis should be **cm**, while the units of the y-axis will be volts. Make sure the axes are labeled and units are specified.

2. How does the response between the red LED and green LED compare and why do think this behavior is observed?

3. What do you think the behavior of the curve would be if a blue LED were used (shorter wavelength)? What about a longer wavelength LED (infrared)?

<u>Exercise 3:</u> The 555 timer and piezo buzzer

Description: In previous labs we used PWM from the MCU to generate analog signals. Another way to do this is with an IC known as a 555 timer. This IC is useful if someone wanted to repeatedly toggle something like an LED on or off, which is exactly what we intend to do for this exercise.

<u>Procedure:</u>

<u>Part 3A:</u> Buzzer Demo

To start this lab exercise, we begin with a simple, yet fun, example of how time varying electrical signals can be converted into mechanical energy in the form of sound. Build the following circuit using the passive buzzer component and pushbuttons illustrated. When connecting the buzzer, the longer leg will go to the MCU while the shorter leg will go to ground. This is also indicated by a plus sign on the top of the buzzer. Next, run the Energia program Lab4_buzzer_demo.ino . If done properly, the buzzer will play different songs depending on which button you press.

DEMONSTRATION: As you have done previously, demonstrate that the circuit is working correctly to the TA or record a short video showing each button working correctly and submit it to canvas.

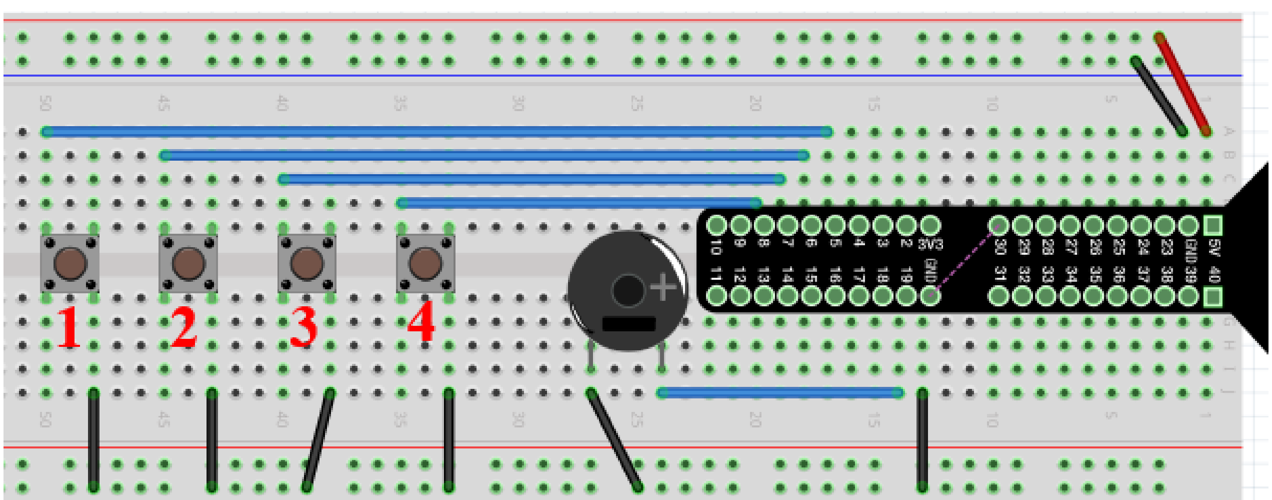


Fig. 5. Diagram of circuit for buzzer demo

<u>Part 3B:</u> 555 timer with LED and buzzer

The previous example used the MCU to generate the time varying signal necessary to make the buzzer play a set of notes, but what would you do if you did not have an MCU? One option is the 555 timer. At this step you will use a 555 timer to turn an LED on and off. Build the circuit shown below in Figure 6. Figure 6A displays a schematic representation of circuit, labeling the different pins that are illustrated in Figure 6D. The diagram of the constructed circuit can be seen in Figure 6B. Notice the red circles on the ICs in figures 6B and 6D. This is will let you know how to place the IC in the circuit correctly when putting the circuit together.

You will notice two additional components you have not used before. These parts are capacitors. The large "can-shaped" capacitor is an electrolytic capacitor which is a polar device, meaning it matters which way you plug it in. If you plug it in backwards it can easily pop and leak chemicals onto the board. To use it correctly, you will notice the shorter leg has minus signs drawn by it telling you that this side should go to the lower potential i.e. ground. The other capacitors are ceramic capacitors which are non-polar. The capacitors you will be using are illustrated in Figure 6C.

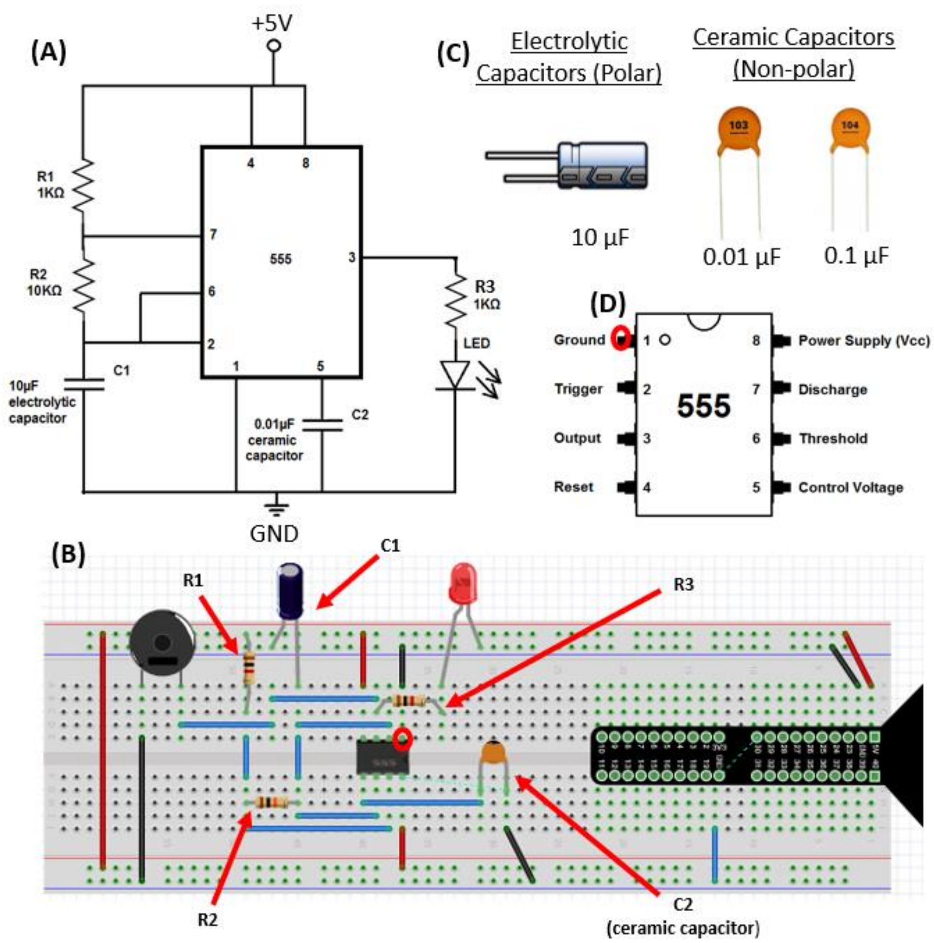


Fig. 6. (A) Schematic for a 555 timer circuit driving an LED and buzzer. **(B)** Diagram of circuit. **(C)** Illustration of capacitors used in the circuit. **(D)** Pin layout of the 555 timer IC. The red circle on pin 1 corresponds to the red circle on the IC of figure **(B)**.

If you have constructed the circuit above correctly, the LED will be blinking approximately 5 times per second. When an event occurs at a regular interval we can define it's frequency as the number of events that occur per second. Frequency has units of Hertz (Hz), so for this scenario, the light will blink at a frequency of approximately 5 times per second or 5 Hz.

Next, we wish to see how we can change the frequency at which the LED blinks. The frequency of a 555 timer signal depends on the values of R1, R2, and C1 and is given by the following formula:

$$Frequency = \frac{1.44}{(R_1 + 2R_2)C_1}$$

This formula will be used in the following questions. Complete these questions in Canvas before moving on to part c.

Questions: Submit answers to the following questions on Canvas.

1. Now we want to see what happens when we change the values of the resistors and capacitor to control the frequency. Below you will see a table containing different combinations of resistors and capacitors that can be used in the circuit above. Make the changes and complete the table on Canvas.

R1 (Ohms)	**R2 (Ohms)**	**C1 (µF)**	**f (Hz)**
10k	1k	10	
1k	1k	10	
330	330	10	
330	100	10	
330	100	100	
1k	1k	100	
10k	1k	0.1	
1k	1k	0.1	
1k	10k	0.1	
330	100	0.1	

NOTE: When changing components, make sure you unplug the MCU from your computer before you make any changes. Once you exchange the components, plug the board back in and observe the LED blinking.

2. At what combination of R1, R2, and C1 were you unable to tell if the LED was blinking or not? What frequency does this correspond to? When did it become hard to hear the buzzer?

3. The eye works similar to a camera in the sense that it has an ideal framerate for visualizing motion. For instance, the monitor on your laptop has a refresh rate of 60 Hz so that image transitions look smooth to the human eye. Given the formula for the frequency output of a 555 timer and using only the values you have available to you, what is a possible combination of resistors and capacitors that will get you as close as possible to this value (Note: You may use resistor values that could be made if you configure a set of resistors in series or parallel)?