Laboratory Activity #7

Pulse Width Modulation (PWM) - Tone Generator

Name: **<Eric C. Delos Reyes>**
ID number: **<12212814>**

Suggested Time: **120 minutes**

Individual work will be evaluated manually using the following criteria:

- Circuit Connection (30%)
- Program / Code (40%)
- Discussion & Documentation (30%)

---

1. Materials

a. Arduino Uno Board with USB Cable
b. 3 pcs LEDs
c. 4 pcs 220 ohm resistor (any value 1KΩ or less)
d. 1 pc 10K ohm resistor
e. 1 pc LDR
f. Piezo buzzer or computer speaker
g. Connecting wires
h. Breadboard

2. Procedures

a. Choose any three(3) PWM pins of your Arduino board and connect the three (3) resistors (220 ohms) and LEDs in series with the pins.

b. Choose one PWM pin of your Arduino board and connect a resistor (220 ohms or less) and a piezo buzzer in series with the pin.

c. Connect all the negative terminals of the LEDs and the piezo buzzer to the GND of the Arduino board. See Figure 1.

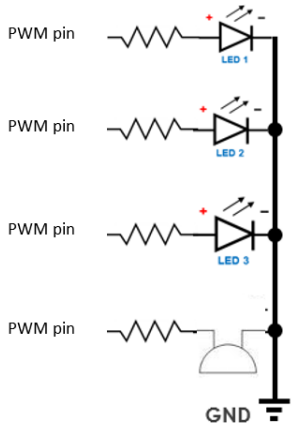


Figure 1.

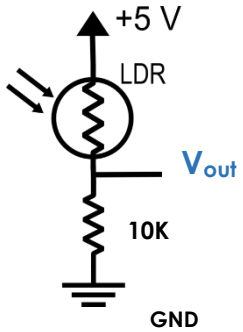


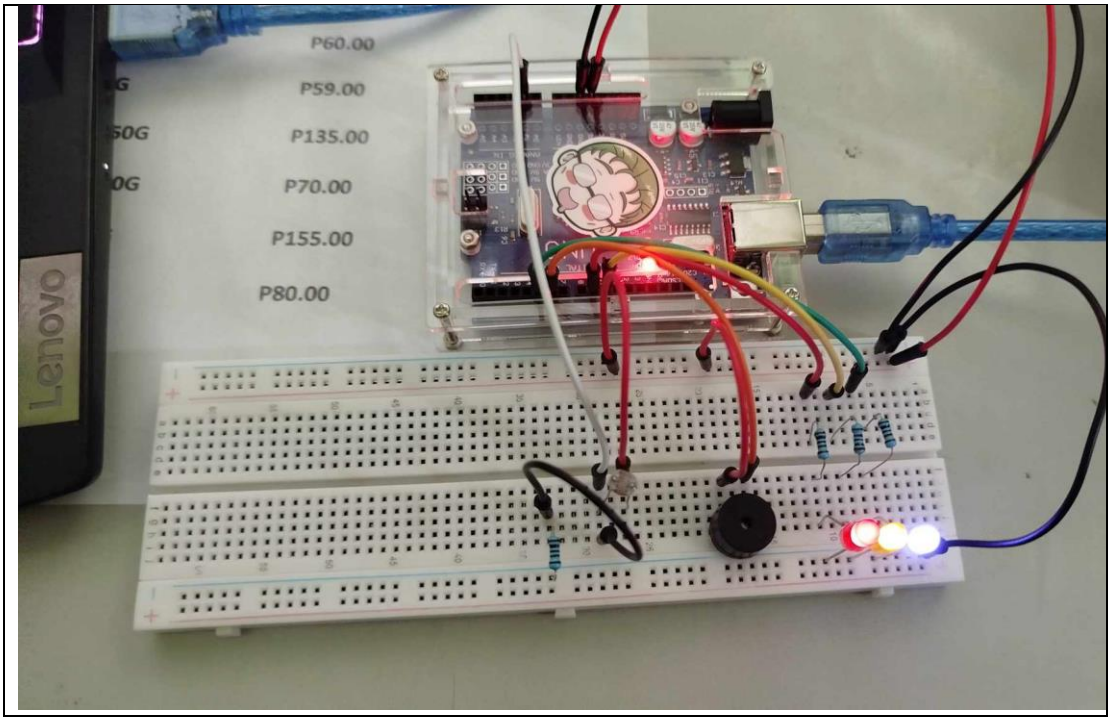
Figure 2 .

d. Choose one analog input pin of your Arduino board and connect the **V_{out}** of the voltage divider circuit of your LDR shown in Figure 2.

e. Create a program that reads the analog signal from the LDR circuit, generates a musical tone (any music), and controls the 4 PWM pins connected to the OUTPUT LEDs and a Piezo Buzzer. The required operations are as follows:

1. The output LEDs glow (gradually increasing brightness) in sync with the tone and tempo of the music. *** LEDs are OFF when there is no tone.
2. The music tempo depends on the LDR sensor: when the LDR senses a lower light intensity (blocked or covered), the tempo (beat) of the music increases and vice versa.
3. Please see the video lecture for the sample output: https://youtu.be/w2Hw4gZW8lg

f. Take a **clear** picture(s) of your circuit implementation and paste it here.



g. Copy-paste your working source code here.

```
int LightPins[3] = {9, 10, 5};
int duration;
double conv_light;
int delayTime;
double Melody[46] = {
  392, 262, 294, 330, 330, 0,
  330, 294, 330, 262, 262, 0,
```

```
  262, 294, 330, 350, 440, 0,
  440, 392, 350, 330, 0,
  262, 294, 330, 350, 440, 0,
  440, 392, 350, 330, 262, 0,
  392, 262, 294, 330, 350, 0,
  294, 294, 330, 262, 0
  };

//https://gist.github.com/mhermans/b5c39250ce74c8974324 - You
Are My Sunshine
unsigned long Durations[46] = {
  4, 4, 4, 4, 4, 8,
  4, 4, 4, 4, 4, 8,
  4, 4, 4, 4, 4, 8,
  4, 4, 4, 4, 8,
  4, 4, 4, 4, 4, 8,
  4, 4, 4, 4, 4, 8,
  4, 4, 4, 4, 4, 8,
  4, 4, 4, 4, 8
};

void setup() {
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);

  pinMode(A1, INPUT);

  pinMode(6, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  delay(400);


  for (int j = 0; j < 46; j++){
    delayTime = analogRead(A1);
    Serial.println(delayTime);
    duration = ((delayTime * 4) / Durations[j]);

    // //Using (calculated map - get percent and multiplied to
max number)
    // for (int i = 0; i < 3; i++){
    //   conv_light = (Melody[j]/450)*255;
```

```
          //   analogWrite(LightPins[i], conv_light);
          //   Serial.println(conv_light);
          // }


          for (int i = 0; i < 3; i++){
            conv_light = (Melody[j]/450)*255;
            analogWrite(LightPins[i], Melody[j]);
          }

          tone(6, Melody[j], duration);
          int pauseBetweenNotes = duration * .950;
          delay(pauseBetweenNotes);
        }

      }
```

h. Write a short self-evaluation on what you have learned in the Week 08 Activity.

I learned how to create an analog-like digital system using the PWM by changing the clock speed which is pragmatically done using the AnalogWrite and also using tone for the Piezo buzzer. I learned and especially was able to appreciate how it works which mainly based on making the lights blink faster in order to create varying light intensity. I see its importance if the system needs extra slots for its Analog input. I learned how to implement it in board especially using the buzzer and LEDs to notice the changes according to different clock speed.

i. Desired Score: 100/100

j. Create a YouTube channel (*if you don't have one yet*). Take a **short video** (average of 30 seconds) of your **running prototype**.

https://youtu.be/Fb2lR-jmrzo

Required to Submit:

Modify this document with the following content:

- ✓ **Name** and **Student number**
- ✓ Clickable Link of your 15-30 seconds **video clip** via YouTube
- ✓ Save this document as **PDF** before submitting to Dropbox

Your work will be graded based on the following Rubric:

- Circuit Connection (40%)
- Code Accuracy and Efficiency (40%)
- Document (20%)

Criteria	Level 5 **Excellent** 40 points	Level 4 **Good** 35 points	Level 3 **Satisfactory** 30 points	Level 2 **Poor** 20 points	Level 1 **Unacceptable** 0 point
Circuit Connection (40%)	The circuit is working perfectly based on the given circuit diagram; the connection is neat and understandable; Exceptional circuit connection;	The circuit is working based on the given circuit diagram; the connection is understandable; Good circuit connection;	The circuit is working based on the given circuit diagram, but the connection is a bit messy with extra unnecessary routings; Acceptable circuit connection;	The circuit is barely working; the connection is hard to trace and not in the right place where it should be. Needs improvement;	The circuit is not working;
Criteria	Level 5 **Excellent** 40 points	Level 4 **Good** 35 points	Level 3 **Satisfactory** 30 points	Level 2 **Poor** 20 points	Level 1 **Unacceptable** 0 point
Code Accuracy and Efficiency (40 %)	The computation and logic formulation are precise to produce the expected output for all the required functionalities; It effectively uses the appropriate programming constructs;	The computation and logic formulation produce the expected output for all the required functionalities; but are unable to employ more appropriate programming constructs to reduce the amount of program resources consumed;	The computation and logic formulation occasionally produce the expected output; Unable to use more appropriate programming constructs to reduce the amount of program resources consumed;	The computation and logic formulation is incorrect for most of the functionalities; the program does not produce the correct result; The program logic is poorly written and does not use the appropriate programming constructs;	The code is not working
Criteria	Level 5 **Excellent** 20 points	Level 4 **Good** 15 points	Level 3 **Satisfactory** 10 points	Level 2 **Poor** 5 points	Level 1 **Unacceptable** 0 point

Discussion and Documentation (20 %)	The discussion and document are complete with excellent discussion of results.	Documentation is good but either the pictures of the circuit connection, discussion, or a clickable link of video clip in YouTube is missing.	The discussion and documentation are fairly acceptable; with one of the required contents missing.	Documentation is incomplete; low-quality pictures of the circuit connection, poor discussion, and/or no clickable YouTube video link.	No Document