

## Assignment : Day 2

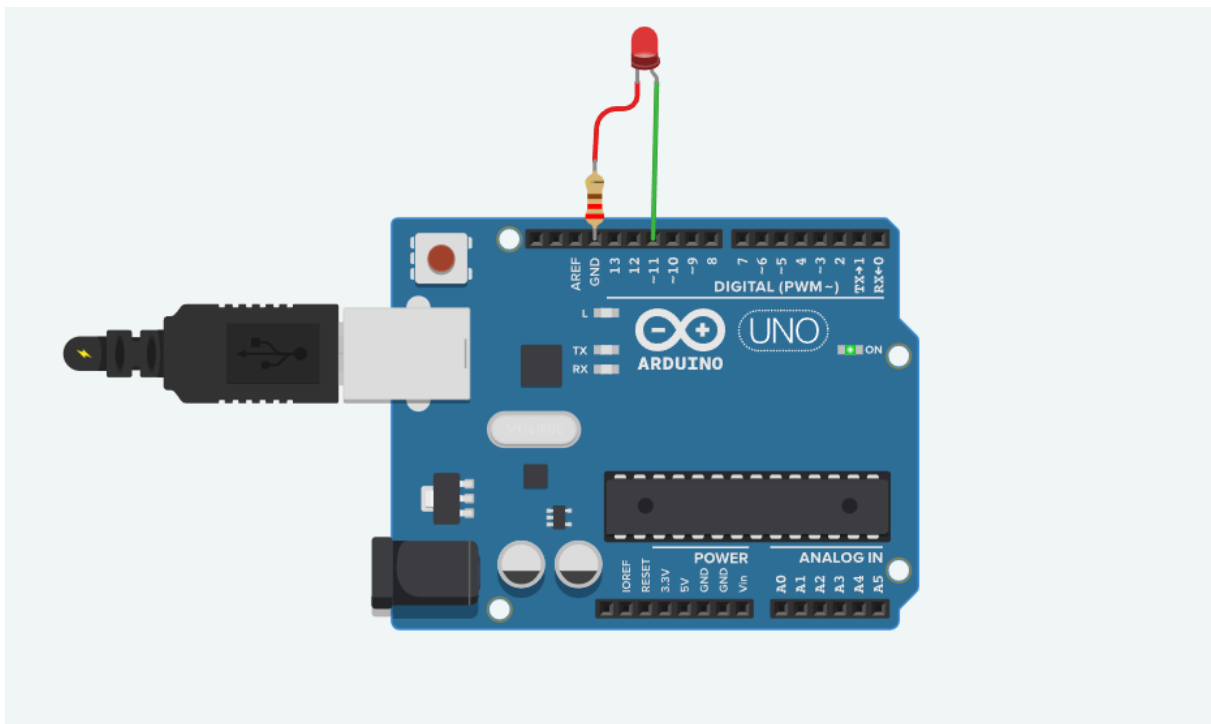
**Is it possible to create a simulation in Tinkercad where an LED is dimmed and brightened using PWM with an Arduino? If yes, explain how you would implement this simulation step by step, including the circuit setup and example code. If not, explain why it is not possible.**

---

Yes, it is possible to create such a simulation in Tinkercad. Here's how we can do it step by step:

1. Create a new Tinkercad circuit project.
  2. Drag and drop an Arduino board, an LED and a resistor onto the workspace.
  3. Connect anode of LED to any PWM supported pin of the microcontroller. I am using 11 in this case.
  4. Connect cathode of LED to ground of the microcontroller through a resistor. I am using 220 ohm resistor in this case.
  5. Add code logic for operating the LED using PWM. The code will gradually increase the brightness of the LED from 0 to 255 and then decrease it back to 0, creating a dimming effect.
  6. Start simulation.
- 

**Here is the circuit diagram for the simulation:**



Here is the code I am using along with description of major parts:

```
int ledPin = 11; // PWM pin
int brightness = 0; // LED brightness
int fadeAmount = 5; // Amount to change brightness in each loop

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
  analogWrite(ledPin, brightness); // Set PWM duty cycle
  brightness += fadeAmount;

  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount; // Reverse fade direction
  }

  delay(10); // Wait for visible fade
}
```

---

Let us understanding the code step by step:

```
int ledPin = 11;
int brightness = 0;
int fadeAmount = 5;
```

- ledPin is the PWM pin used.
- brightness is the for tracking current brightness of LED.
- fadeAmount is the amount by which brightness will be increased or decreased in each loop iteration.

```
void setup() {
  pinMode(ledPin, OUTPUT);
}
```

- In setup function, we set the ledPin as OUTPUT so that we can control the LED.

```
void loop() {
  analogWrite(ledPin, brightness);
  brightness += fadeAmount;

  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }

  delay(10);
}
```

- In loop function, we use analogWrite to set the PWM duty cycle to current brightness value.
- We increase the brightness by fadeAmount.
- If brightness reaches 0 or 255, we reverse the fade direction by changing the sign of fadeAmount, so from next loop, the brightness will decrease by fadeAmount.
- Finally, we add a small delay to make the fade effect visible.