

```

// These constants won't change. They're used to give names
// to the pins used:
const int analogInPin = A0; // Analog input pin that the potentiometer is
attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to

int sensorValue = 0; // value read from the pot
int outputValue = 0; // value output to the PWM (analog out)
int fuckFade = 1;
int fadeHolder;

unsigned long int currentMillis;
int thresholdMillis = 4000;
boolean started = false;

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);

  // set variables to current time
}

void loop() {
  sensorValue = analogRead(analogInPin);

  if (fadeHolder < sensorValue) {
    fadeHolder = sensorValue;
  }

  fadeHolder = fadeHolder - fuckFade;

  if (fadeHolder > 400) {
    if (started == false) {
      started = true;
      currentMillis = millis();
    }

    if ((millis() - currentMillis) >= thresholdMillis && started = true) {
      analogWrite(analogOutPin, outputValue);
      delay(100);
    }
  }
}

```

```

    analogWrite(analogOutPin, 0);
    delay(100);
    analogWrite(analogOutPin, outputValue);
    delay(100);
    analogWrite(analogOutPin, 0);
    delay(100);
    analogWrite(analogOutPin, outputValue);
    delay(100);
    analogWrite(analogOutPin, 0);
    delay(100);
    analogWrite(analogOutPin, outputValue);
    delay(100);
    analogWrite(analogOutPin, 0);
    delay(100);
    started = false;
    currentMillis = millis();
    //Serial.println((millis() - currentMillis));
} else {

```

```

    outputValue = map(fadeHolder, 400, 1023, 0, 255);
    analogWrite(analogOutPin, outputValue);

```

```

}

```

```

if (fadeHolder < 400) {
    analogWrite(analogOutPin, 0);

    currentMillis = millis();
}

```

```

/*
// print the results to the serial monitor:
Serial.print("sensor = " );
Serial.print(sensorValue);
Serial.print("\t output = ");
Serial.println(outputValue);

// wait 2 milliseconds before the next loop
// for the analog-to-digital converter to settle
// after the last reading:
*/
delay(20);
}
}

```