```
#define TRIG_PIN  16  // Trigger pin of HC-SR04 (use GPIO 16)
#define ECHO_PIN  17  // Echo pin of HC-SR04 (use GPIO 17)
#define LED_PIN   2   // LED pin
#define MAX_DISTANCE 150  // Maximum distance (in cm) where LED is active
void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  Serial.begin(9600);  // Start serial communication for debugging
}
void loop() {
  // Send a pulse to trigger the HC-SR04 sensor
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  // Read the duration of the pulse from the echo pin
  long duration = pulseIn(ECHO_PIN, HIGH);
  // Calculate the distance in cm
  long distance = duration * 0.0344 / 2;
  // Debugging: Print the distance
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  // Only blink LED if object is within the MAX_DISTANCE
  if (distance > 0 && distance <= MAX_DISTANCE) {
    // Map the distance to a delay value (for example, between 100ms and 1000ms)
    long blinkDelay = map(distance, 0, 200, 100, 1000);  // Adjust the distance range if needed

    Serial.print("Blinking Delay: ");
    Serial.println(blinkDelay);
    digitalWrite(LED_PIN, HIGH);
    delay(blinkDelay);
    digitalWrite(LED_PIN, LOW);
    delay(blinkDelay);
  } else {
    Serial.println("Object too far, LED off.");
    digitalWrite(LED_PIN, LOW);  // Ensure LED stays off if object is too far
  }
  delay(100); // Small delay to prevent excessive looping
}
```

This is meant to fix the range for the LED to reduce the amount of length it takes for the echo pin to send out a signal if within the range. This helps our range with the guitar so that nothing else but the hand is picked up when over the sensor.