Download header file from: <https://github.com/FastLED/FastLED/blob/master/FastLED.h>

#include<FastLED.h>

#define LED\_PIN 3

#define NUM\_LEDS 1

#define LED\_TYPE WS2812B

#define COLOR\_ORDER GRB

#define BRIGHTNESS 100

CRGB leds[NUM\_LEDS];

#define UPDATES\_PER\_SECOND 100

void setup()

{

Serial.begin(9600);

FastLED.addLeds<LED\_TYPE, LED\_PIN, COLOR\_ORDER>(leds, NUM\_LEDS);

FastLED.setBrightness( BRIGHTNESS );

}

void loop()

{

Serial.print("Start\n");

for(int i=0;i<=3000;i++)

{

static uint8\_t hue = 0;

for(int i = 0; i < NUM\_LEDS; i++)

{

leds[i] = CHSV(hue++, 255, 255);

FastLED.show();

fadeall();

delay(10);

}

}

for(int i=0;i<20;i++)

{

leds[0] = CRGB(0,255,0);

FastLED.show();

delay(60000);

}

while(1)

{

leds[0] = CRGB(255,0,0);

FastLED.show();

}

}

void fadeall()

{

for(int i = 0; i < NUM\_LEDS; i++)

{

leds[i].nscale8(250);

}

}

int vibratePin = 7; //sets the digital pin(d0)of the vibration sensor @pin 7

int ledPin = 2; //sets the LED @pin 2

int value; //defines the value variable

void setup() {

pinMode(vibratePin,INPUT); //sets the vibration sensor as INPUT

pinMode(ledPin,OUTPUT); //sets the LED as OUTPUT

}

void loop() {

value=digitalRead(vibratePin); //reads the sensors status

if(value==1) //if sensor sends a HIGH signal

{

digitalWrite(ledPin,HIGH); //turns LED on

delay(3000); //duration of 3 seconds

}

else

digitalWrite(ledPin,LOW); //LED is turned off

}

/\*//==============================================================================//

\* Vibration Sensor interfacing with Arduino UNO

\*/ //=============================================================================//

#include <Arduino.h>

#include <stdio.h>

#define ON 1

#define OFF 0

/\*

\* Pin Description

\*/

int vibration\_Sensor = A5;

int LED = 13;

/\*

\* Programme flow Description

\*/

int present\_condition = 0;

int previous\_condition = 0;

/\*

\* Pin mode setup

\*/

void setup() {

pinMode(vibration\_Sensor, INPUT);

pinMode(LED, OUTPUT);

}

/\*

\* Led blink

\*/

void led\_blink(void);

/\*

\* main\_loop

\*/

void loop() {

previous\_condition = present\_condition;

present\_condition = digitalRead(A5); // Reading digital data from the A5 Pin of the Arduino.

if (previous\_condition != present\_condition) {

led\_blink();

} else {

digitalWrite(LED, OFF);

}

}

void led\_blink(void) {

digitalWrite(LED, ON);

delay(250);

digitalWrite(LED, OFF);

delay(250);

digitalWrite(LED, ON);

delay(250);

digitalWrite(LED, OFF);

delay(250);

}

Code for Arduino UNO

From <https://forum.arduino.cc/index.php?topic=567400.0>

const byte RelayPin = 13 ; //names the pin to be used in things like digitalprint(RelayPin, HIGH) instead of using a number that you have to change in a bunch of places. 13 is the onboard LED, used for testing. Because later it will be a different pin, controlling a relay that powers an alarm

const byte VibeSenseOut = 3; //continuously attempts to power pin 2 through vibration sensor+resistor

const byte VibeSenseIn = 2; //if this gets power, then it means there is a complete circuit through the vibration sensor+resistor. That means there has been a strong impact, and that resets the clock/turns off LED

unsigned long prevMillis;

unsigned long interval = 10000;

bool prevState;

void setup() {

pinMode(RelayPin, OUTPUT) ; //declares pin as inputs/outputs

pinMode(VibeSenseOut, OUTPUT) ; //declares pin as output

pinMode(VibeSenseIn, INPUT) ; //declares pin as Input

digitalWrite(VibeSenseOut, HIGH) ; //sets initial states

digitalWrite(VibeSenseIn, LOW) ;

}

void loop() {

VibeSense();

Countdown();

}

void VibeSense() {

//First "method," goal is to use two declared pins (input and output with vibration sensor and resistor inbetween) to sense an impact

bool currentState = digitalRead(VibeSenseIn);

if ( currentState == HIGH && prevState == LOW ) {

//When a NEW impact is detected

//Turn off LED (later relay) off and restart the countdown

digitalWrite(RelayPin, LOW);

prevMillis = millis();

}

prevState = currentState;

}

void Countdown() {

//Second "method," uses Millis() as a clock to see if a certain time has elapsed. If it has, an onboard LED lights up (later on, it will close a relay)

if (millis() - prevMillis >= interval) {

//time has expired

digitalWrite(RelayPin, HIGH); //Lights the LED (later a relay closes)

}

}