Earthquake Detector

**Aim:** Aim: WAP in Tinkercad to develop earthquake detector using tilt sensor, LCD, Buzzer and Arduino Uno.

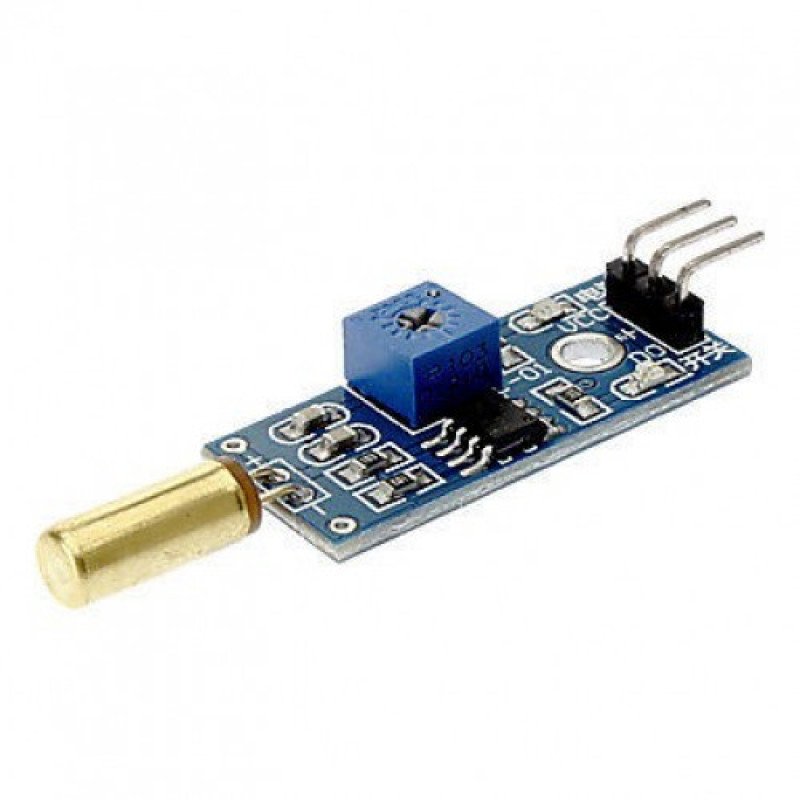
**Apparatus:**

* Tilt Sensor
* Arduino Uno
* Buzzer

**Theory:**

**Tilt Sensor**

A **tilt sensor** is an instrument that is used for measuring the **tilt** in multiple axes of a reference plane. **Tilt sensors** measure the **tilting** position with reference to gravity and are used in numerous applications. They enable the easy detection of orientation or **inclination.**



**Arduino UNO**

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino Uno differs from all preceding boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Arduino Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.



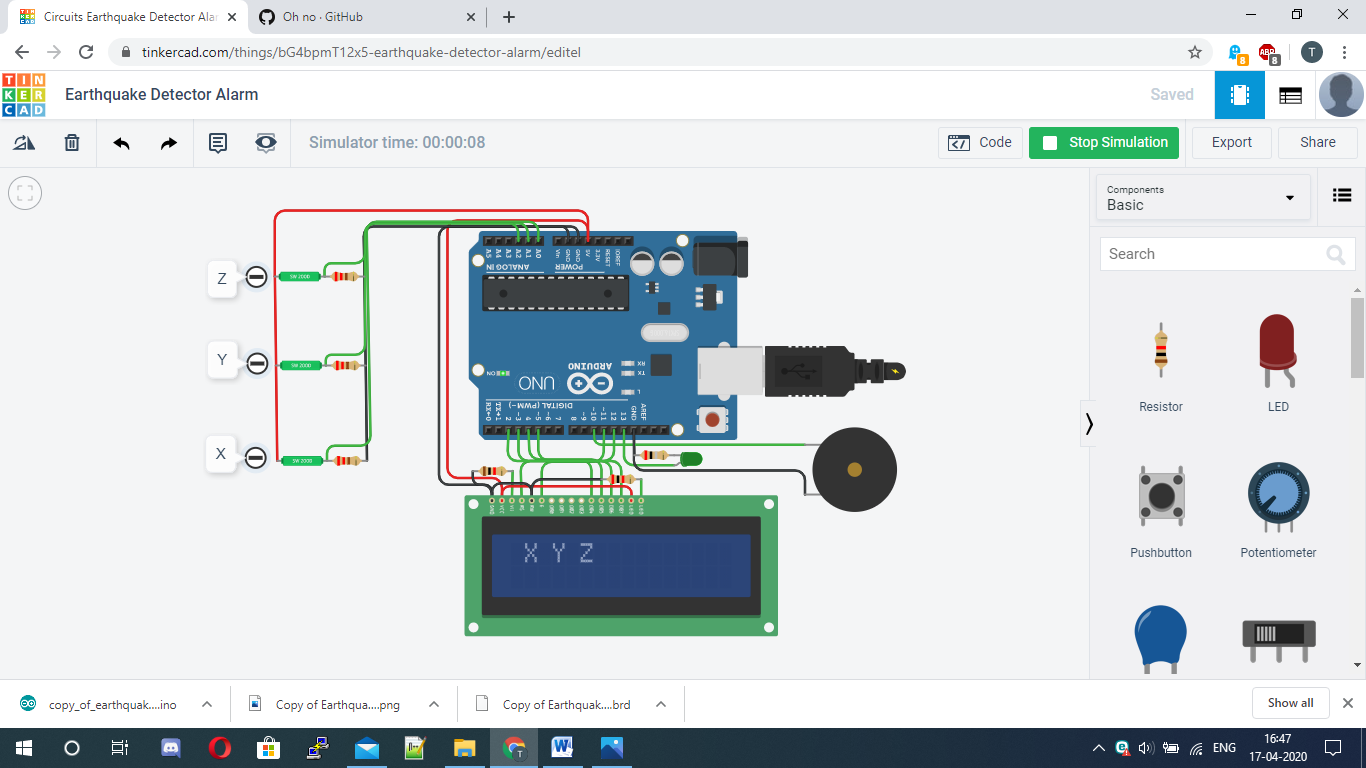
|  |  |
| --- | --- |
| **Features** |  |
| * 14 digital I/O pins (six of which provide PWM output) * 3.3 V supply generated by an on-board regulator * Six analog input pins * 32 KB of flash memory | * Can supply 40 mA of DC current per pin * 16 MHz clock speed * Code example from Arduino website to help get started |

**Buzzer**

A **buzzer** or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of **buzzers** and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.



**Circuit:**



**Code:**

#include<LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2);

#define buzzer 10

#define led 13

#define x A0

#define y A1

#define z A2

int xsample=0;

int ysample=0;

int zsample=0;

long start;

int buz=0;

#define samples 50

#define maxVal 20

#define minVal -20

#define buzTime 5000

void setup(){

lcd.begin(16,2);

Serial.begin(9600);

delay(1000);

lcd.print("EarthQuake ");

lcd.setCursor(0,1);

lcd.print("Detector ");

delay(2000);

lcd.clear();

lcd.print("Calibrating.....");

lcd.setCursor(0,1);

lcd.print("Please wait...");

pinMode(buzzer, OUTPUT);

pinMode(led, OUTPUT);

buz=0;

digitalWrite(buzzer, buz);

digitalWrite(led, buz);

for(int i=0;i<samples;i++){

xsample+=analogRead(x);

ysample+=analogRead(y);

zsample+=analogRead(z);}

xsample/=samples;

ysample/=samples;

zsample/=samples;

delay(3000);

lcd.clear();

lcd.print("Calibrated");

delay(1000);

lcd.clear();

lcd.print("Device Ready");

delay(1000);

lcd.clear();

lcd.print(" X Y Z ");}

void loop(){

int value1=analogRead(x);

int value2=analogRead(y);

int value3=analogRead(z);

int xValue=xsample-value1;

int yValue=ysample-value2;

int zValue=zsample-value3;

lcd.setCursor(0,1);

lcd.print(xValue);

lcd.setCursor(6,1);

lcd.print(yValue);

lcd.setCursor(12,1);

lcd.print(zValue);

delay(100);

if(xValue < minVal || xValue > maxVal || yValue < minVal || yValue > maxVal || zValue < minVal || zValue > maxVal){

if(buz == 0)

start=millis();

buz=1; }

else if(buz == 1) {

lcd.setCursor(0,0);

lcd.print("Earthquake Alert ");

if(millis()>= start+buzTime)

buz=0;}

else{

lcd.clear();

lcd.print(" X Y Z ");}

digitalWrite(buzzer, buz);

digitalWrite(led, buz);

Serial.print("x=");

Serial.println(xValue);

Serial.print("y=");

Serial.println(yValue);

Serial.print("z=");

Serial.println(zValue);

Serial.println(" $");}

**Result:**

