# Motion Detection using PIR (passive infrared sensor) Sensor.

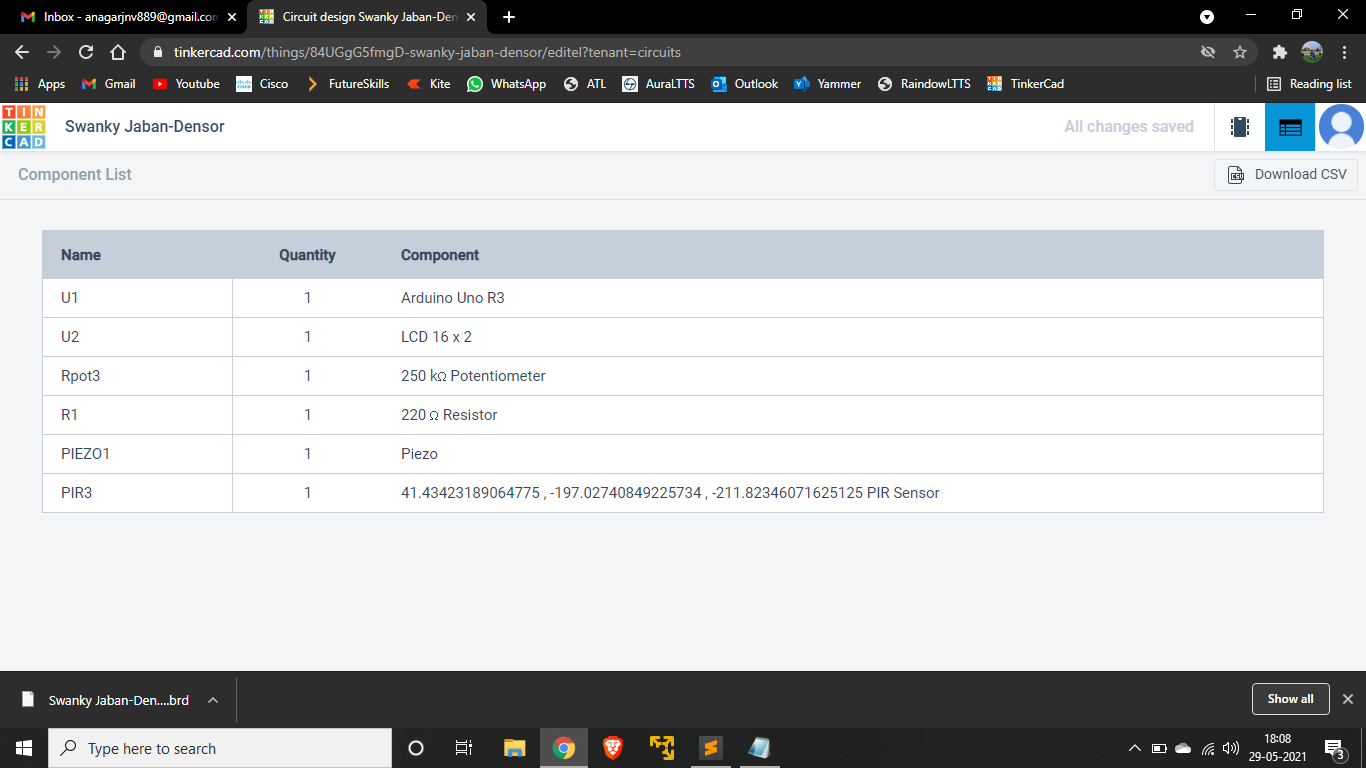
# Introduction:

# The objective of this project is to develop a motion sensor alarm based on a Passive Infra-Red (PIR) sensor module. There are many vendors that manufacture the PIR sensor modules and almost all of them are pretty much the same in function. PIR sensor detects the changes in the IR levels emitted by humans. PIR sensors can detect levels of infrared radiation. In this project, microcontroller continuously monitors the output from the sensor module and turns a buzzer on when it goes active. The application areas of this project are: All outdoor lights, Lift lobby, Multi apartment complexes, common staircases, for basement or covered parking area, shopping malls, for garden lights. Certain crystalline materials have the property to generate a surface electric charge when exposed to thermal infrared radiation. This phenomenon is known as pyroelectricity. The presence of human body creates a sudden change in the IR profile of the surrounding that is sensed by the pyroelectric sensor. When the microcontroller detects the sensor is triggered, it drives the buzzer. The sensor is in retriggering mode, the buzzer remains on as long as the motion is continuously sensed.

# Functional Requirements

1. Power supply (+5v) to the hardware circuit.
2. An interrupt is given to microcontroller by PIR sensor when it detects the motion.
3. A buzzer which is used to produce sound.

**Non-Functional Requirements**

1. Arduino Uno R3
2. 41.43423189064775, -197.02740849225734, -211.82346071625125 PIR Sensor
3. LCD 16x2
4. 250K ohm Potentiometer
5. 220-ohm Resistor
6. Piezo
7. Printed Circuit Board
8. 

Description

1. **Arduino Uno R3**

The Arduino Uno R3 is a microcontroller board based on a removable, dual-inline-package (DIP) ATmega328 AVR microcontroller. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs). Programs can be loaded on to it from the easy-to-use Arduino computer program. The Arduino has an extensive support community, which makes it a very easy way to get started working with embedded electronics. The R3 is the third, and latest, revision of the Arduino Uno.

1. **PIR sensors**

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason, they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

1. **LCD 16x2**

The term [LCD stands for liquid crystal display](https://www.elprocus.com/difference-alphanumeric-display-and-customized-lcd/). It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment [light-emitting diodes](https://www.elprocus.com/light-emitting-diode-led-working-application/) and seven segments.

# 250K ohm Potentiometer

# A potentiometer is a three-[terminal](https://en.wikipedia.org/wiki/Terminal_(electronics)) [resistor](https://en.wikipedia.org/wiki/Resistor) with a sliding or rotating contact that forms an adjustable [voltage divider](https://en.wikipedia.org/wiki/Voltage_divider).[[1]](https://en.wikipedia.org/wiki/Potentiometer#cite_note-1) If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat. The measuring instrument called a [potentiometer](https://en.wikipedia.org/wiki/Potentiometer_(measuring_instrument)) is essentially a [voltage divider](https://en.wikipedia.org/wiki/Voltage_divider) used for measuring [electric potential](https://en.wikipedia.org/wiki/Electric_potential) (voltage); the component is an implementation of the same principle, hence its name.

1. **220-ohm Resistor**

The resistor is a passive electrical component to create resistance in the flow of electric current. In almost all electrical networks and electronic circuits they can be found. The resistance is measured in ohms.

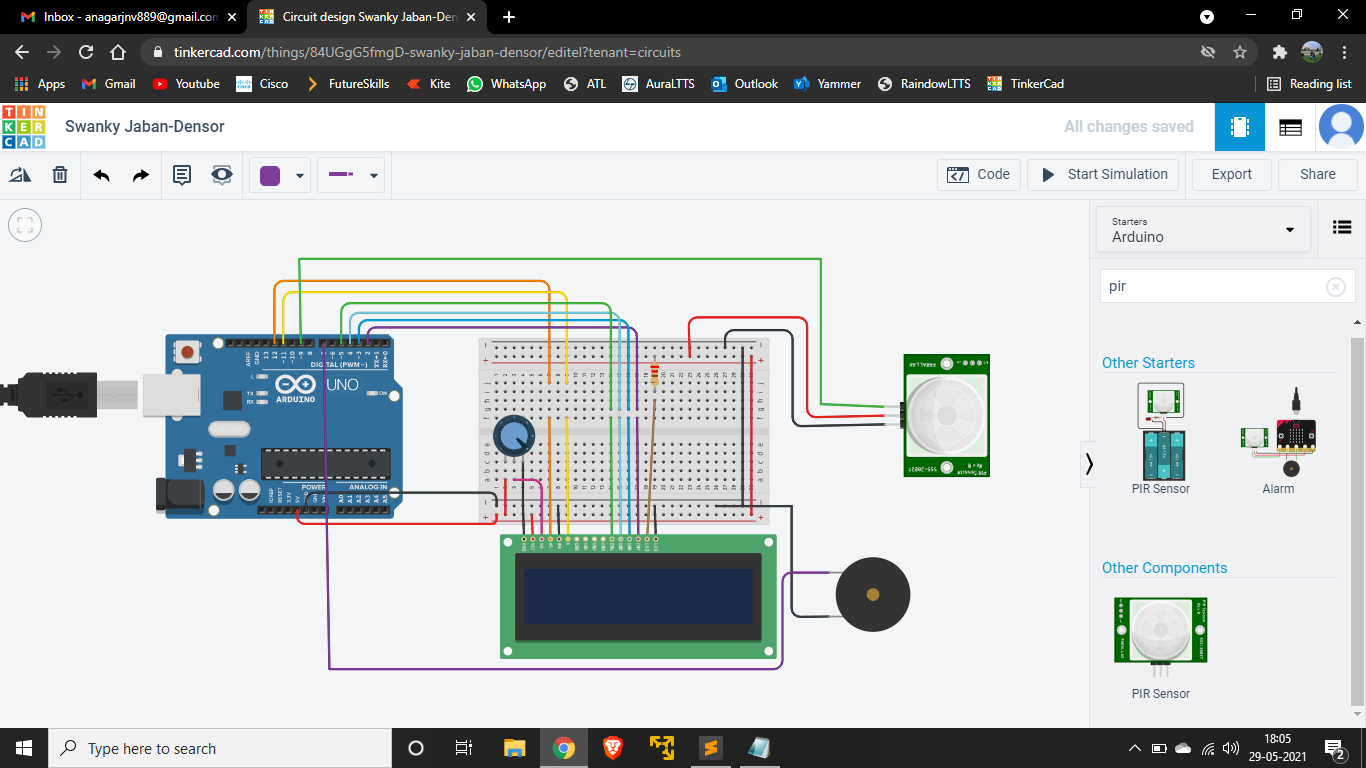
1. **Piezo**

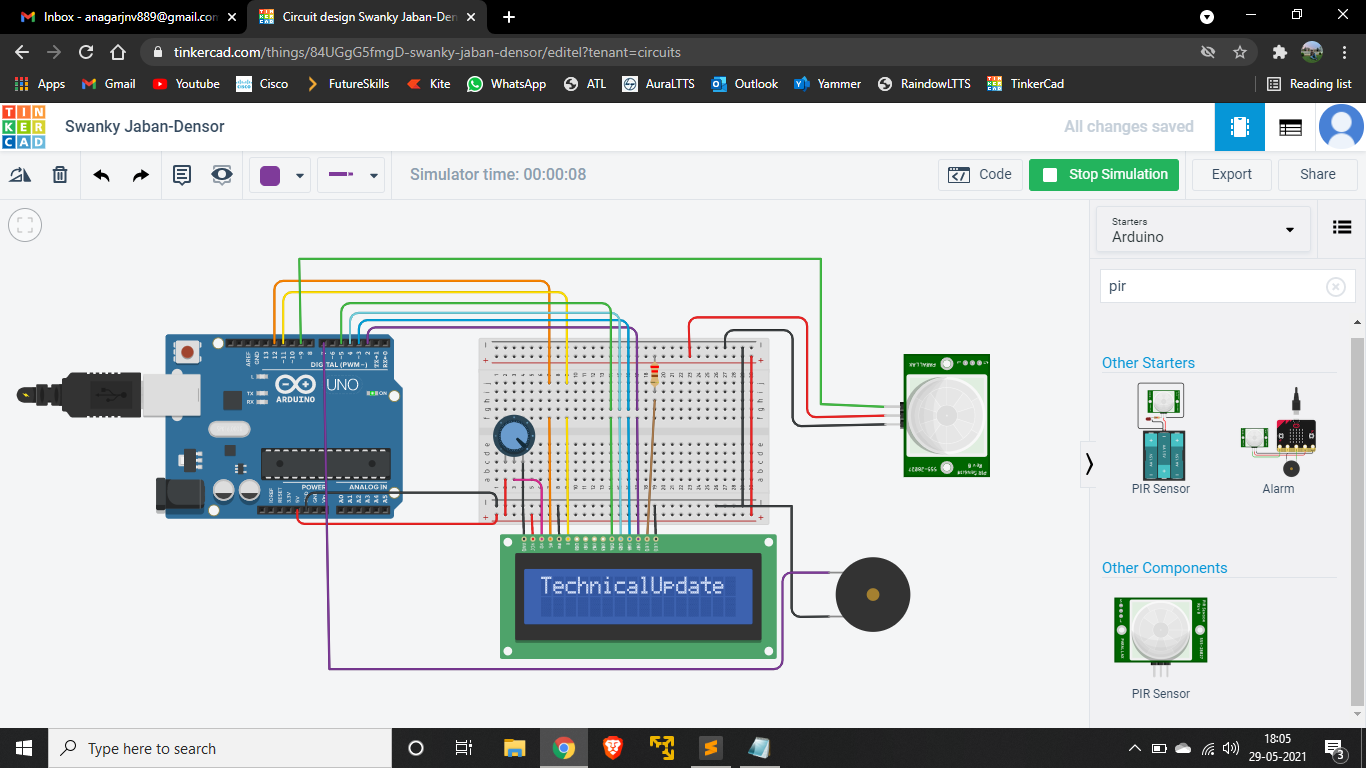
piezo materials can transform an electrical voltage into a displacement (making the material an actuator) or can transform a displacement into an electrical voltage (making the material a load or displacement sensor). The mathematical connection between both parameters is rather complex and can only be described in tensorial form as it is related to the crystal structure of the piezo material. However, for the case of a composite, the strain response as a function of the applied voltage can simply be expressed as a scalar, the so-called [piezoelectric](https://www.sciencedirect.com/topics/materials-science/piezoelectricity) charge constant.

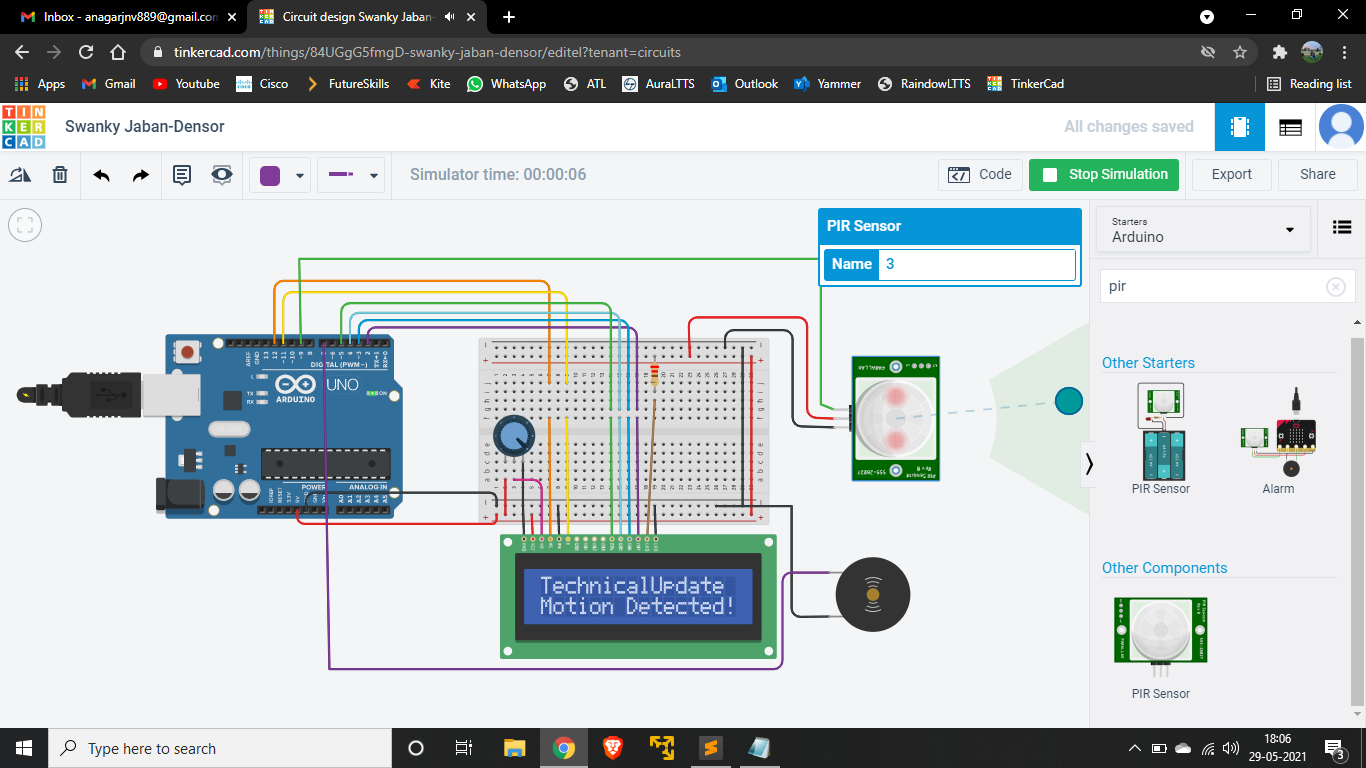
1. **Printed Circuit Board**

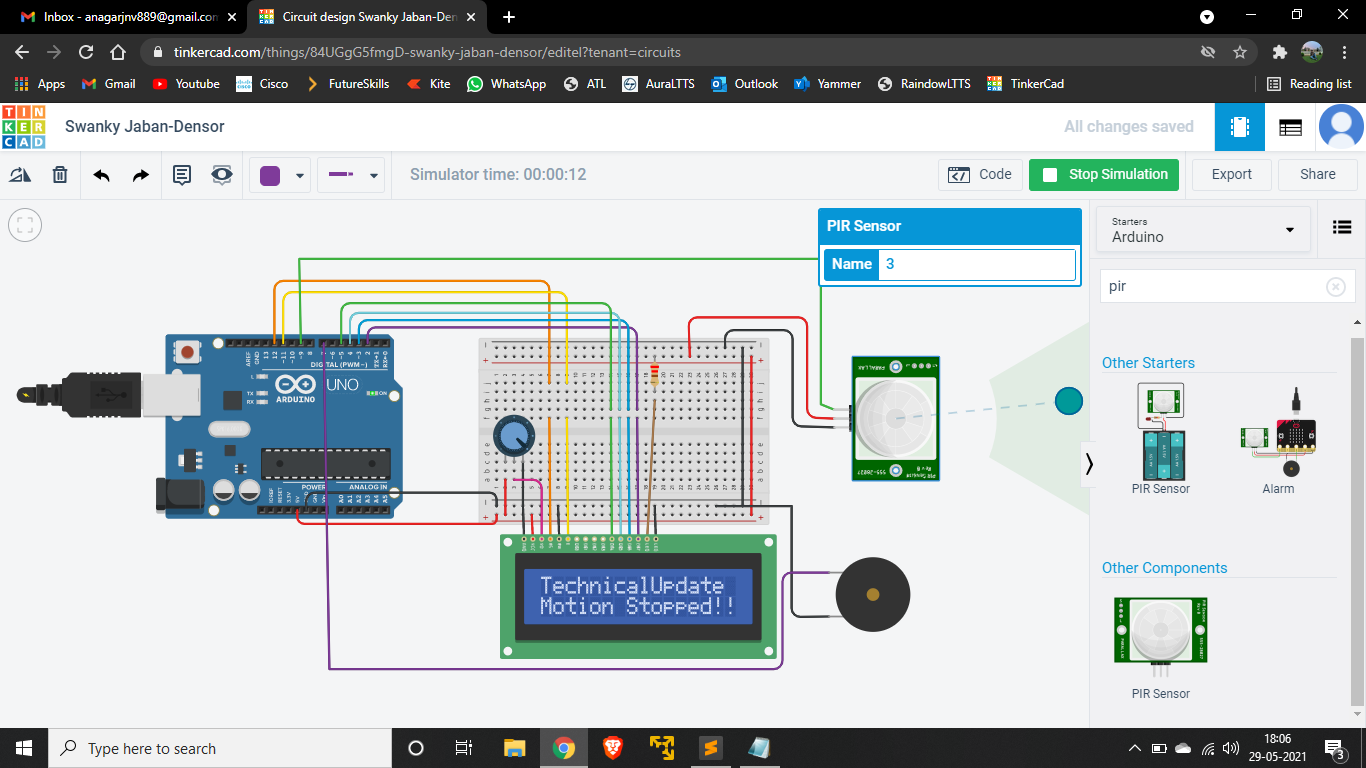
A printed circuit board (PCB) mechanically supports and electrically connects [electrical](https://en.wikipedia.org/wiki/Electrical) or [electronic components](https://en.wikipedia.org/wiki/Electronic_components) using [conductive](https://en.wikipedia.org/wiki/Electrical_conductor) tracks, pads and other features [etched](https://en.wikipedia.org/wiki/Industrial_etching) from one or more sheet layers of copper [laminated](https://en.wikipedia.org/wiki/Laminated) onto and/or between sheet layers of a [non-conductive](https://en.wikipedia.org/wiki/Insulator_(electricity)) substrate. Components are generally [soldered](https://en.wikipedia.org/wiki/Soldering) onto the PCB to both electrically connect and mechanically fasten them to it. Printed circuit boards are used in all but the simplest electronic products. They are also used in some electrical products, such as passive switch boxes.

**Demonstration with the help of Diagram**



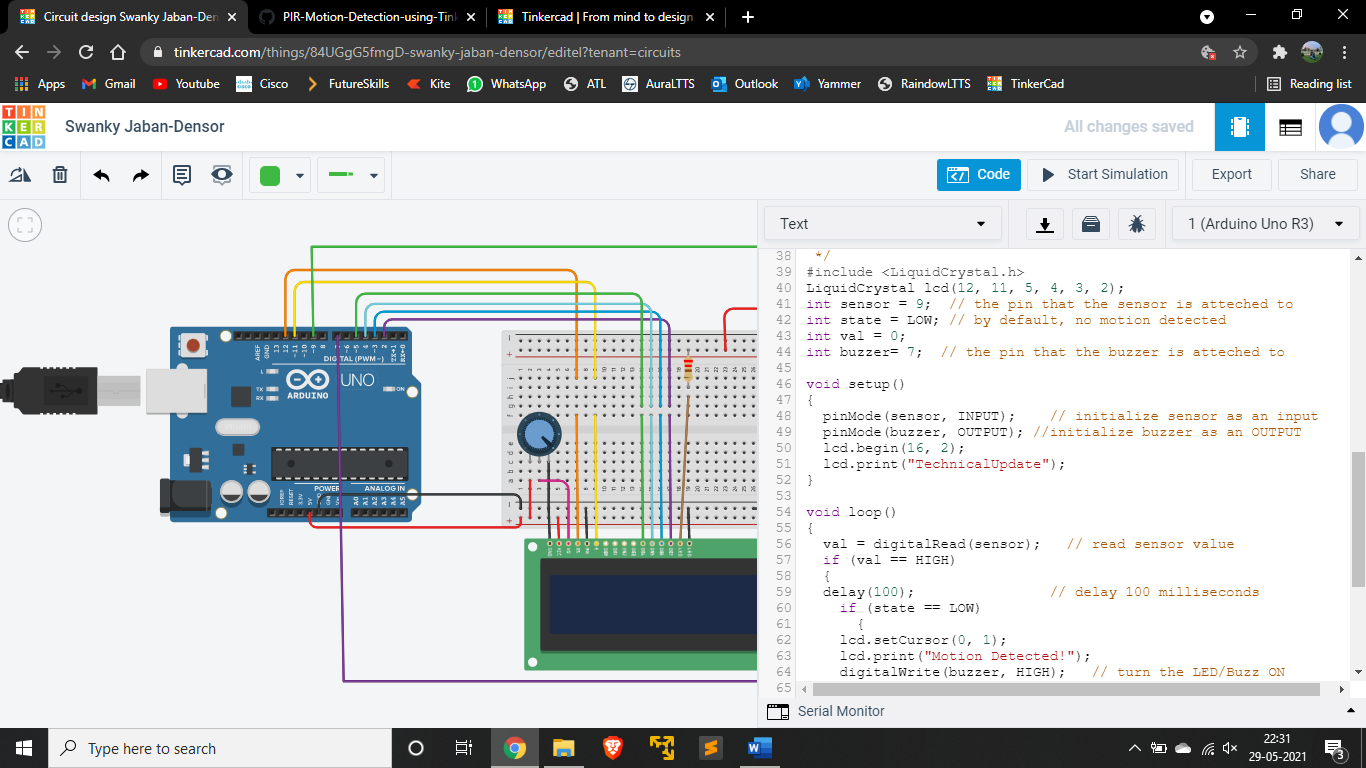






**Programming of Arduino**

|  |
| --- |
|  |
|  | //This code will only work using the tinkercad.com starter circuits of LCD |
|  |  |
|  | #include <LiquidCrystal.h> |
|  | LiquidCrystal lcd(12, 11, 5, 4, 3, 2); |
|  | int sensor = 9; // the pin that the sensor is atteched to |
|  | int state = LOW; // by default, no motion detected |
|  | int val = 0; |
|  | int buzzer= 7; // the pin that the buzzer is atteched to |
|  |  |
|  | void setup() |
|  | { |
|  | pinMode(sensor, INPUT); // initialize sensor as an input |
|  | pinMode(buzzer, OUTPUT); //initialize buzzer as an OUTPUT |
|  | lcd.begin(16, 2); |
|  | lcd.print("TechnicalUpdate"); |
|  | } |
|  |  |
|  | void loop() |
|  | { |
|  | val = digitalRead(sensor); // read sensor value |
|  | if (val == HIGH) |
|  | { |
|  | delay(100); // delay 100 milliseconds |
|  | if (state == LOW) |
|  | { |
|  | lcd.setCursor(0, 1); |
|  | lcd.print("Motion Detected!"); |
|  | digitalWrite(buzzer, HIGH); // turn the LED/Buzz ON |
|  | state = HIGH; // update variable state to HIGH |
|  | } |
|  | } |
|  | else |
|  | { |
|  | delay(200); // delay 200 milliseconds |
|  | if (state == HIGH) |
|  | { |
|  | lcd.setCursor(0, 1); |
|  | lcd.print("Motion Stopped!"); |
|  | digitalWrite(buzzer, LOW); // turn the Buzzer ON |
|  | state = LOW; // update variable state to LOW |
|  | } |
|  | } |
|  | }  **Diagrams With Programming** |

****