Assignment 1

To write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino.

1. Designing the Circuit in Tinkercad

components:

- Arduino Uno R3
- Ultrasonic Sensor (HC-SR04)
- Breadboard (optional)
- Jumper wires
- 1 LED
- 1 Resistor (220Ω or 330Ω)

Connections:

- VCC (Ultrasonic) → 5V (Arduino)
- GND (Ultrasonic) → GND (Arduino)
- TRIG (Ultrasonic) → Digital Pin 9 (Arduino)
- ECHO (Ultrasonic) → Digital Pin 10 (Arduino)
- **LED Anode (long leg) → Arduino Digital Pin 7** (through resistor).
- Connect the anode to **Pin 7**.
- Place a **220Ω resistor** between Pin 7 and LED anode (to limit current).
- **LED Cathode (short leg) → GND (Arduino)**.

2. Arduino Code for Ultrasonic Distance

```
// Ultrasonic Distance Measurement with LED Indicator

const int trigPin = 9;    // Ultrasonic TRIG pin
const int echoPin = 10;   // Ultrasonic ECHO pin
const int ledPin  = 7;    // LED pin
long duration;
int distance;
void setup () {
  Serial.begin (9600);
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);
  pinMode (ledPin, OUTPUT);   // Set LED pin as output
}
void loop () {
  // Clear the trigPin
  digitalWrite (trigPin, LOW);
  delayMicroseconds (2);
  // Trigger the ultrasonic pulse
```

```
  digitalWrite (trigPin, HIGH);
  delayMicroseconds (10);
  digitalWrite (trigPin, LOW);
  // Measure echo response time
  duration = pulseIn (echoPin, HIGH);
  // Convert time to distance
  distance = duration * 0.034 / 2;
  // Print distance to Serial Monitor
  Serial.print ("Distance: ");
  Serial.print (distance);
  Serial.println (" cm");
  // LED control: Turn ON if distance < 200 cm
  if (distance < 200) {
    digitalWrite (ledPin, HIGH);  // LED ON
  } else {
    digitalWrite (ledPin, LOW);   // LED OFF
  }
  delay(500);
}
```

3. How to Run in Tinkercad

1. Add the **LED and resistor** as described.
2. Upload the modified code in **Code → Text Editor**.
3. Click **Start Simulation**.
4. Move the **object in front of the ultrasonic sensor** (in simulation you can drag the distance slider).
5. Watch the LED:
    - ● ON when object is closer than **200 cm**
    - ● OFF otherwise

Assignment 2

To write a program to sense the available networks using Arduino

1. Designing the Circuit in Tinkercad

components:

- Arduino Uno R3
- LCD 16×2 (parallel)
- 10k potentiometer (for LCD contrast)
- 220 Ω resistor (for backlight)
- Breadboard (optional)
- Jumper wires

Wiring (LCD in 4-bit mode)

Connect the LCD pins exactly as follows:

LCD pin	Connect to (Arduino / others)
VSS	GND
VDD	5V
V0 (contrast)	**middle pin of 10k pot** (other two pot pins → 5V and GND)
RS	D12
RW	GND
E	D11
D4	D5
D5	D4
D6	D3
D7	D2
A (LED+)	5V **through 220Ω resistor**
K (LED-)	GND

Notes:

- If characters are invisible, slowly turn the potentiometer to adjust contrast.
- You can place the pot and resistor on the breadboard for tidy wiring.

```
#include <LiquidCrystal.h>

// LCD pins: RS=12, E=11, D4=5, D5=4, D6=3, D7=2

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

// ----- Mock network data -----

const char* SSIDS[] = {

  "MyWiFi",

  "CollegeNet",

  "Guest123",

  "Lab_AP",
```

```
  "Cafe_Free"
};
int RSSI[] = { -45, -70, -82, -55, -60 }; // pretend signal strengths (dBm)
bool LOCKED[] = { true, true, false, true, false }; // true=Encrypted, false=Open
const int N = sizeof(SSIDS) / sizeof(SSIDS[0]);
// Timing
unsigned long lastSwitch = 0;
unsigned long lastScan = 0;
int idx = 0;
// Map RSSI to 0..5 bars and print using '|' characters
void printBars(int rssi, int col) {
  int bars;
  if (rssi >= -55) bars = 5;
  else if (rssi >= -65) bars = 4;
  else if (rssi >= -75) bars = 3;
  else if (rssi >= -85) bars = 2;
  else if (rssi >= -95) bars = 1;
  else bars = 0;
  lcd.setCursor(col, 1);
  for (int i = 0; i < 5; i++) {
    if (i < bars) lcd.print('|'); // visible bar char
    else lcd.print(' ');
  }
}
void showNetwork(int i) {
  lcd.clear();
  lcd.setCursor(0, 0);
  String name = SSIDS[i];
  if (name.length() > 16) name = name.substring(0, 16);
  lcd.print(name);
  lcd.setCursor(0, 1);
  lcd.print(RSSI[i]); lcd.print("dBm ");
  if (LOCKED[i]) lcd.print("Lock ");
  else           lcd.print("Open ");
  // Bars at columns 11..15
  printBars(RSSI[i], 11);
}
void scanningAnimation(unsigned long ms) {
```

```
  unsigned long start = millis();
  int dot = 0;
  while (millis() - start < ms) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Scanning");
    // animate small dots
    lcd.setCursor(9, 0);
    for (int i = 0; i < (dot % 4); i++) lcd.print('.');
    dot++;
    delay(250);
  }
}
void randomizeRSSI() {
  for (int i = 0; i < N; i++) {
    int jitter = random(-5, 6);
    RSSI[i] += jitter;
    if (RSSI[i] < -90) RSSI[i] = -90;
    if (RSSI[i] > -35) RSSI[i] = -35;
  }
}
void setup() {
  lcd.begin(16, 2);
  randomSeed(analogRead(A0));
  lcd.print("Mock WiFi Scan");
  delay(1000);
  showNetwork(idx);
}
void loop() {
  unsigned long now = millis();
  // Fake rescan every 10 seconds
  if (now - lastScan >= 10000UL) {
    scanningAnimation(2000); // 2 seconds animation
    randomizeRSSI();
    lastScan = now;
    showNetwork(idx);
  }
  // Rotate shown network every 2 seconds
```

```
  if (now - lastSwitch >= 2000UL) {
    idx = (idx + 1) % N;
    showNetwork(idx);
    lastSwitch = now;
  }
}
```

2 How to set up & run the simulation in Tinkercad

1. Open **Tinkercad → Circuits → Create new Circuit**.
2. Drag **Arduino Uno**, **LCD 16×2**, **10k pot**, **220Ω resistor**, and optionally a **breadboard** onto the workspace.
3. Wire the parts exactly per the table in Section 3. (Double check RS, E, D4–D7 pins.)
4. Click **Code → Text** and **paste** the sketch above (replace any blocks).
5. Click **Start Simulation**.
6. If LCD is blank or shows blocks, turn the potentiometer knob to adjust contrast until text appears.
7. Watch: it will show “Scanning…”, then cycle through mock SSIDs, dBm, Open/Lock, and bars. It will “rescan” every 10 seconds (jittering RSSI to simulate changes).

3 Quick explanation of the important code parts

- LiquidCrystal lcd(...) — tells library which pins are used.
- SSIDS[], RSSI[], LOCKED[] — arrays of fake networks. Change/add names here.
- scanningAnimation() — simple visual “Scanning…” animation.
- randomizeRSSI() — slightly changes RSSI values so numbers look dynamic.
- showNetwork(i) — writes SSID on line 1, and dBm/Open/Lock + bars on line 2.
- millis() timing — used so animation, scanning and rotation run without blocking the whole sketch.

4 Troubleshooting tips

- **Blank LCD**: pot not set or RW not to GND; check power (VDD/VSS) and V0 wiring.
- **Garbled characters**: wrong D4–D7 or RS/E pin order. Double-check mapping.
- **Backlight dim**: ensure A → 5V through 220Ω and K → GND.
- **Nothing changes**: ensure you uploaded/pasted the sketch and clicked **Start Simulation**.

Assignment 3

To write a program to detects the vibration of an object with sensor using Arduino

1. Components Needed (in Tinkercad)

- Arduino Uno R3
- Tilt switch sensor (SW-200D) (or pushbutton → simulating vibration sensor)
- 10kΩ resistor (for pull-down if using button; tilt switch has internal)
- 1 LED + 220Ω resistor (indicator for vibration)
- Jumper wires
- Breadboard (optional for neat wiring)

2. Circuit Design (Connections)

If using tilt switch in Tinkercad (SW-200D):

- One leg of tilt switch → **5V**
- Other leg → **Arduino digital pin 2**
- Also connect that leg → **10kΩ resistor → GND** (pull-down resistor so it reads LOW when no vibration)

LED connections:

- LED **Anode (long leg)** → Arduino **D13** (through 220Ω resistor)
- LED **Cathode (short leg)** → GND

So the Arduino can read HIGH when vibration is detected and turn ON the LED.

3. Arduino Program

```
const int vibrationPin = 2;   // Tilt switch / vibration sensor input
const int ledPin = 13;        // LED output
int vibrationState = 0;
void setup () {
  pinMode(vibrationPin, INPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
void loop () {
  // Read sensor
  vibrationState = digitalRead(vibrationPin);
  if (vibrationState == HIGH) {
    digitalWrite(ledPin, HIGH);      // LED ON
    Serial.println("Vibration Detected!");
  } else {
```

```
    digitalWrite(ledPin, LOW);       // LED OFF
    Serial.println("No Vibration");
  }
  delay(200); // small delay to stabilize readings
}
```

4. How to Run in Tinkercad

1. Go to **Tinkercad → Circuits → Create New Circuit**.
2. Drag and place: **Arduino Uno**, **tilt switch sensor (or pushbutton)**, **LED**, **resistors**.
3. Wire exactly as explained above.
4. Go to **Code → Text** and paste the Arduino sketch.
5. Click **Start Simulation**.
6. To simulate vibration:
    - If using tilt switch: click the component → toggle its state → Arduino reads it as vibration.
    - If using pushbutton: press/release → Arduino simulates vibration.
7. Open the **Serial Monitor** to see `"Vibration Detected!"` or `"No Vibration"`.
8. LED on Pin 13 will also turn ON when vibration is detected.

Working Principle

- The tilt switch (or pushbutton) works like a vibration sensor: it closes (HIGH) when shaken or tilted.
- Arduino checks the sensor state:
    - **HIGH → vibration detected** → LED ON + Serial message.
    - **LOW → no vibration** → LED OFF.

Assignment 4

To write a program to get temperature notification using Arduino.

We'll use an **Arduino Uno** with a **temperature sensor** (e.g., LM35/ TMP36) and an **LED or Serial Monitor** for notification.

1. Components Needed (in Tinkercad)

Components:

- Arduino Uno R3
- LM35/ TMP36 (temperature sensor)
- 1 LED (for visual notification)
- 1 Resistor (220Ω for LED)
- Breadboard & wires

2. Circuit Design (Connections)

Connections:

- LM35/ TMP36:
  - VCC → 5V
  - GND → GND
  - OUT → A0 (Analog Pin)
- LED:
  - Anode (+) → Pin 13 (or any digital pin) via 220Ω resistor
  - Cathode (–) → GND

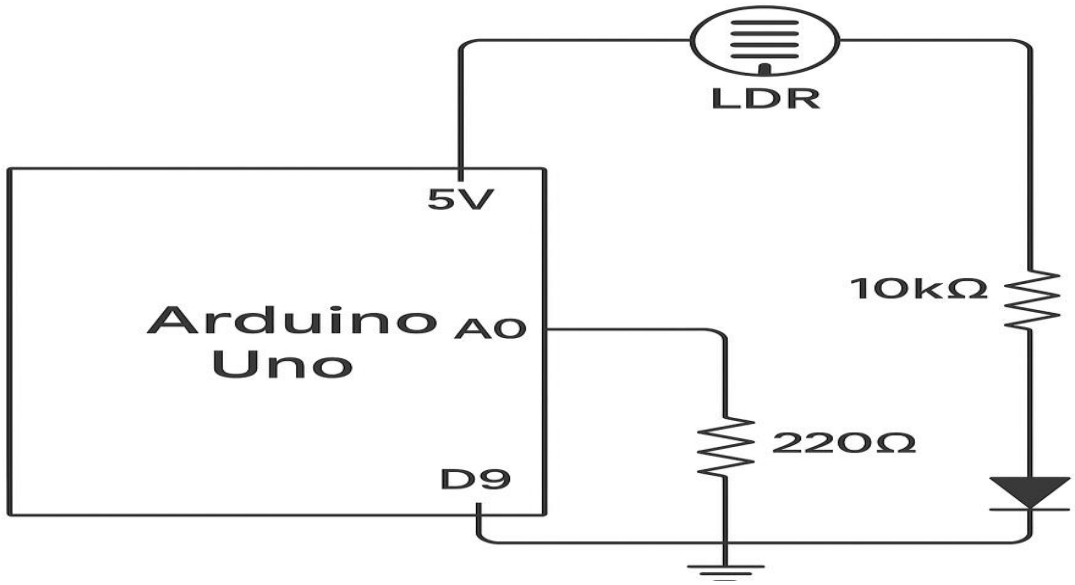
3. Arduino Code

```
// Temperature Notification using LM35


int sensorPin = A0;   // LM35 output connected to A0 (TMP36)
int ledPin = 13;      // LED connected to pin 13
float temperature;    // variable to store temperature
void setup() {
  Serial.begin (9600);   // Start Serial Monitor
  pinMode(ledPin, OUTPUT);
}
void loop () {
  int reading = analogRead(sensorPin);         // read sensor value
  temperature = (reading * 5.0 * 100.0) / 1024; // convert to Celsius
  Serial.print("Temperature: ");
  Serial.print(temperature);
```

```
  Serial.println(" *C");

  // Notification if temperature > 30°C
  if (temperature > 30) {
    digitalWrite(ledPin, HIGH);  // Turn ON LED
    Serial.println("⚠ High Temperature!");
  } else {
    digitalWrite(ledPin, LOW);   // Turn OFF LED
  }
  delay(1000); // read every second
}
```

4. Run the Simulation

1. Paste the code into the **Code Editor** in Tinkercad.
2. Click **Start Simulation**.
3. Open the **Serial Monitor** (top-right) to see temperature readings.
4. Increase TMP36 temperature (click the sensor in Tinkercad → adjust value).
    - If >30°C → LED lights up + “High Temperature!” message appears.



Assignment 5

To write a program for LDR to vary the light intensity of LED using Arduino.

1. Components Needed in Tinkercad

- Arduino Uno board
- Breadboard
- 1 × LED
- 1 × 220Ω resistor (for LED)
- 1 × LDR (Light Dependent Resistor)
- 1 × 10kΩ resistor (for LDR voltage divider)
- Jumper wires

2. Circuit Connections

1. **LED Circuit**
   - Connect LED **anode (long leg)** → Arduino pin **9** (PWM pin).
   - Connect LED **cathode (short leg)** → 220Ω resistor → **GND**.
2. **LDR Circuit (Voltage Divider)**
   - One leg of **LDR → 5V**.
   - Other leg of **LDR → A0** and one side of **10kΩ resistor**.
   - Other side of **10kΩ resistor → GND**.

This creates a voltage divider so Arduino can read varying light values on **A0**.

3. Arduino Code

Here's the program you can wright into Tinkercad's Code Editor → Text Mode:

```
int LDR_Pin = A0;    // LDR connected to A0
int LED_Pin = 9;     // LED connected to digital pin 9 (PWM)
void setup() {
  pinMode(LED_Pin, OUTPUT);
  Serial.begin(9600);   // for debugging
}
void loop () {
  int ldrValue = analogRead (LDR_Pin);   // Read LDR (0–1023)

  // Map LDR value to LED brightness (0–255 for PWM)
```

```
  int ledBrightness = map (ldrValue, 0, 1023, 255, 0);
  // Inverted mapping: more light → dim LED, less light → brighter LED
  analogWrite (LED_Pin, ledBrightness);  // Adjust LED brightness
  Serial.print ("LDR Value: ");
  Serial.print (ldrValue);
  Serial.print ("  LED Brightness: ");
  Serial.println (ledBrightness);
  Delay (100);
}
```

4. How to Run in Tinkercad

1. Go to Tinkercad Circuits.
2. Create a **new circuit**.
3. Drag and drop **Arduino Uno, breadboard, LED, resistors, and LDR**.
4. Connect components as per wiring above.
5. Click **Code → Switch to Text → Paste code**.
6. Press **Start Simulation**.
7. Change the **light level on LDR** (Tinkercad lets you drag a small "sun" slider on the LDR).
8. Watch the LED brightness change accordingly.

Now your LED will glow **bright in darkness** and **dim in brightness** depending on LDR input.

Schematic View: