Vivekanand Education Society's

Institute of Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Information Technology

IOE Lab

Lab Assignment - 1

Aim: Explore IoT simulation tools(Any two).

Roll No.	70
Name	MAYURI SHRIDATTA YERANDE
Class	D20B
Subject	Internet of Everything
Grade:	

AIM: Explore IoT simulation tools(Any two).

TO-DO:

1. Description of tool, advantages and disadvantages
2. Screenshots of simulation for a sample case study(Step by step)
3. Results of simulation

THEORY:

Tinkercad is a popular 3D modelling, circuitry-simulating, and block-coding software package that's accessible for free via a web browser. Its popularity is, no doubt, a result of its user-friendly simplicity. It consists of three sections, each of which can be considered its own endeavour and be used for different purposes.

Tinkercad was launched as a free web-based 3D modelling tool by Kai Backman and Mikko Mononen in 2011. One year later, Tinkercad had over 100,000 user-created designs on its website. Recognizing the opportunity, Autodesk acquired Tinkercad in 2013. Then in 2017, Autodesk added circuit modelling and, later on, a coding tool to create the Tinkercad we know today.

Because Tinkercad is free and available via a web browser (accessible via PC or tablet), it's popular with beginners and for classroom use. But this doesn't mean that more experienced users won't also enjoy using it.

Advantages of Tinkercad:

1. User-Friendly Interface
2. Web-Based
3. Built-In Learning Resources
4. Integration with 3D Printing
5. Wide Range of Design Tools
6. Collaboration

Disadvantages of Tinkercad:

1. Limited Complexity
2. Limited Export Formats
3. Not Suitable for Advanced Users:
4. Internet Dependency
5. Limited Materials and Textures
6. Privacy Concerns

Wokwi is an online Electronics simulator. You can use it to simulate Arduino, ESP32, STM32, and many other popular boards, parts and sensors.

Unique Features :-

- **WiFi simulation** - Connect your simulated project to the internet. You can use MQTT, HTTP, NTP, and many other network protocols.
- **Virtual Logic Analyzer** - Capture digital signals in your simulation (e.g. UART, I2C, SPI) and analyze them on your computer.
- **Advanced debugging with GDB** - Powerful Arduino and Raspberry Pi Pico debugger for advanced users.
- **SD card simulation** - Store and retrieve files and directories from your code. Club members can also upload binary files (such as images)
- **Chips API** - Create your own custom chips and parts, and share them with the community.
- **Visual Studio Code integration** - Simulate your embedded projects directly from VS Code.

Wokwi compiles your code into a binary firmware, and then executes the binary firmware one instruction at a time, as a real microcontroller would. If you want to learn about the internals, check out the following resources:
The ESP32 is a popular WiFi and Bluetooth-enabled microcontroller, widely used for IoT Projects. Wokwi simulates the ESP32, ESP32-C3, ESP32-S2, ESP32-S3, ESP32-C6 (beta), and ESP32-H2 (alpha).

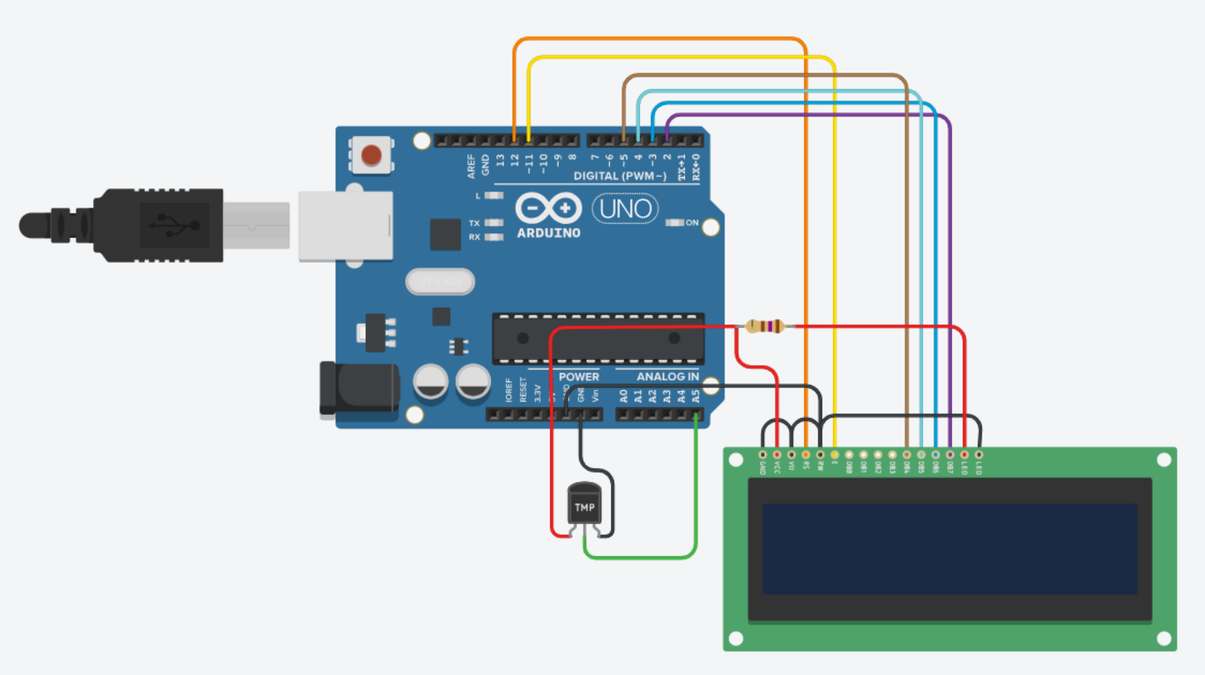
IMPLEMENTATION:

1. Tinkercad

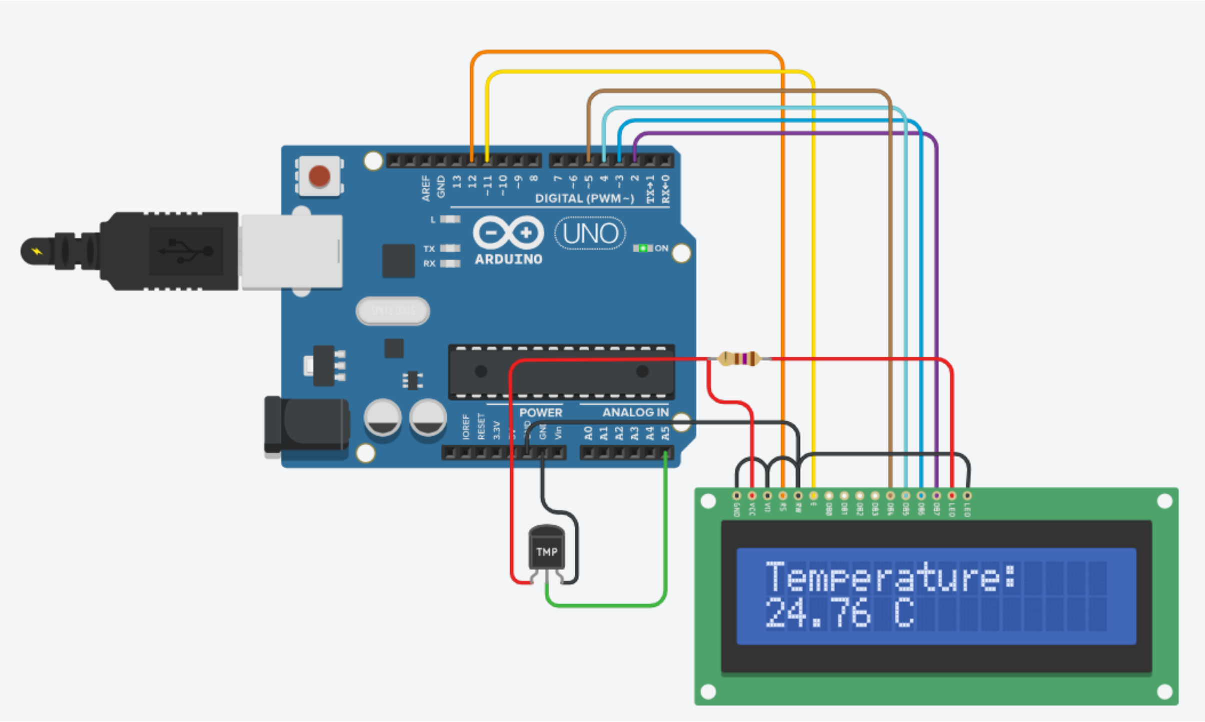
Arduino with LM35 And LCD Screen

Connection:

- The VCC pin of the LM35 is connected to the 5V pin on the Arduino.
- The GND pin of the LM35 is connected to the GND pin on the Arduino.
- The Output pin of the LM35 is connected to the A0 pin on the Arduino.
- The RS pin of the LCD is connected to digital pin 12 on the Arduino.
- The Enable pin (E) of the LCD is connected to digital pin 11 on the Arduino.
- The D4 pin of the LCD is connected to digital pin 5 on the Arduino.
- The D5 pin of the LCD is connected to digital pin 4 on the Arduino.
- The D6 pin of the LCD is connected to digital pin 3 on the Arduino.
- The D7 pin of the LCD is connected to digital pin 2 on the Arduino.
- The Anode pin of the LCD backlight is connected to the 5V pin on the Arduino.
- The Cathode pin of the LCD backlight is connected to a 100 ohm resistor, which is then connected to the GND pin on the Arduino.



Output:



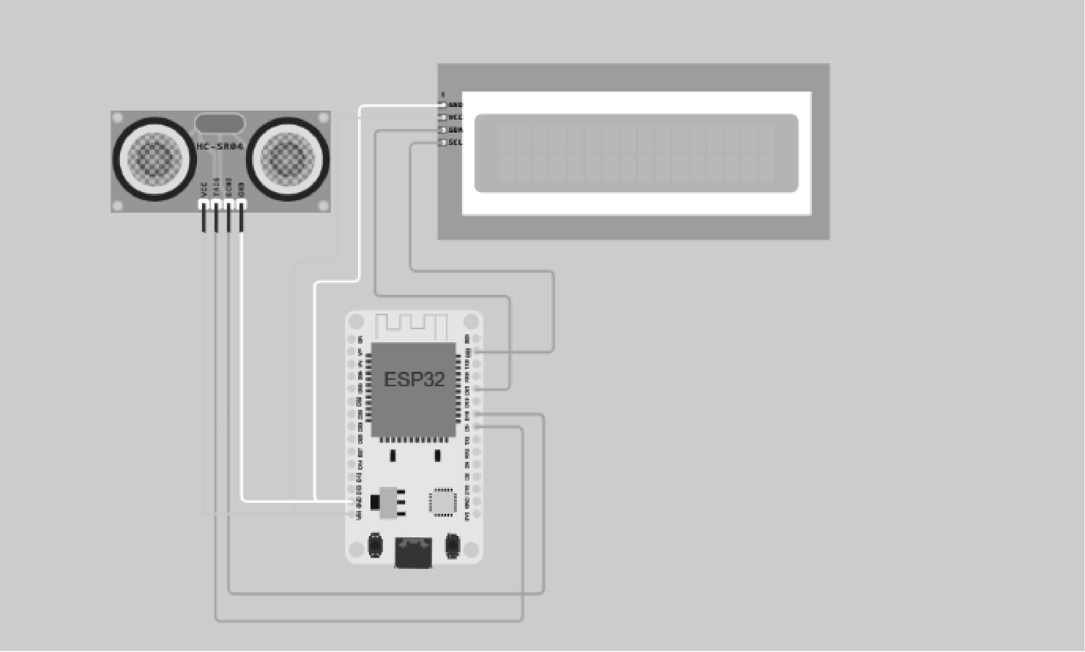
The LCD will display the text "Temperature: " on the first line, followed by the current temperature reading in degrees Celsius on the second line. If the temperature changes, the previous reading will be cleared from the second line before displaying the new reading. This provides a continuously updated temperature display on the LCD.

Code:

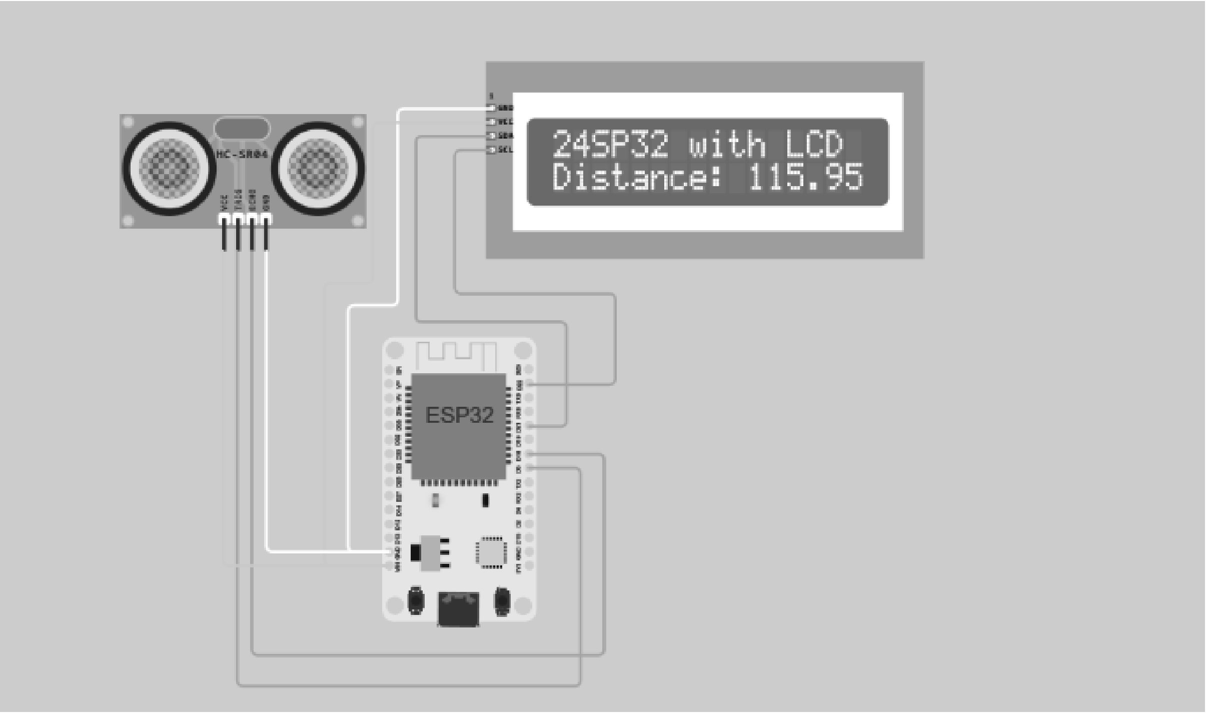
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

#define temp A5
#define led 13

void setup()
{
  lcd.begin(16, 2);
  pinMode(led, OUTPUT);
  pinMode(temp, INPUT);
  Serial.begin(9600);
  lcd.clear();
  lcd.print("Temperature: ");
}
float pre_temp = 0;
void loop() {
  float temperature = 0;
  temperature = (analogRead(temp) * 0.48828125) - 49.95;
  if(pre_temp != temperature)
  {
    lcd.setCursor(0,1);
    lcd.print("                ");
  }
  lcd.setCursor(0,1);
  lcd.print(temperature);
  lcd.print(" C");
  pre_temp = temperature;
}
```

2. WOKWI

ESP32 With Ultrasonic and LCD Screen



Output:



```
24SP32 with LCD
Distance: 115.95
```

- The outcome of the provided code is a continuous display of distance measurements in centimetres on an I2C LCD screen and in the console output.
- The ESP32 microcontroller is set up to work with an ultrasonic sensor (HCSR04) and an I2C LCD screen.
- The LCD initially displays "ESP32 with LCD" on its first line when the code starts running.
- In an infinite loop, the code repeatedly does the following:
  - Measures the distance using the ultrasonic sensor.
  - Prints the distance value in centimetres to the console.
  - Updates the LCD screen with the current distance value on the second line.
- This loop continues indefinitely, so the ESP32 continuously monitors the distance to an object and provides real-time feedback on both the LCD screen and in the console output.

Code:

```
import machine
from machine import SoftI2C, Pin
from lcd_api import LcdApi
from i2c_lcd import I2cLcd
from hcsr04 import HCSR04
from time import sleep
sensor = HCSR04(trigger_pin=5, echo_pin=18, echo_timeout_us=10000)
I2C_ADDR_LCD = 0x27
lcd_Rows = 2
lcd_Columns = 16
i2c = SoftI2C(scl=Pin(22), sda=Pin(21), freq=10000)
lcd = I2cLcd(i2c, I2C_ADDR_LCD, lcd_Rows, lcd_Columns)
lcd.move_to(1, 0)
lcd.putstr("ESP32 with LCD")
while True:
    distance = sensor.distance_cm()
    print('Distance:', distance, 'cm')
    lcd.move_to(0, 1)
    lcd.putstr('Distance: ' + str(distance))
    sleep(1)
```

CONCLUSION:- Exploring these IoT simulation tools reveals valuable resources for IoT development. Tinkercad's simplicity and educational focus make it an excellent starting point for beginners. On the other hand, Wokwi's advanced features, such as internet connectivity and debugging, cater to more complex projects and experienced users. Both tools contribute significantly to the IoT ecosystem, offering versatile options for prototyping and experimentation.