PRACTICAL FILE

Internet Of Things Labs

Paper code – AIML357

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY, SECTOR – 22, ROHINI, NEW DELHI

Submitted to:

Dr. Tripti Lamba

Associate Professor

Submitted by:

Kushagra Gupta

01114811621

I N D E X

S.no	Name of the Experiment	Total Marks					Date	Signature
		R1	R2	R3	R4	R5		

Experiment I

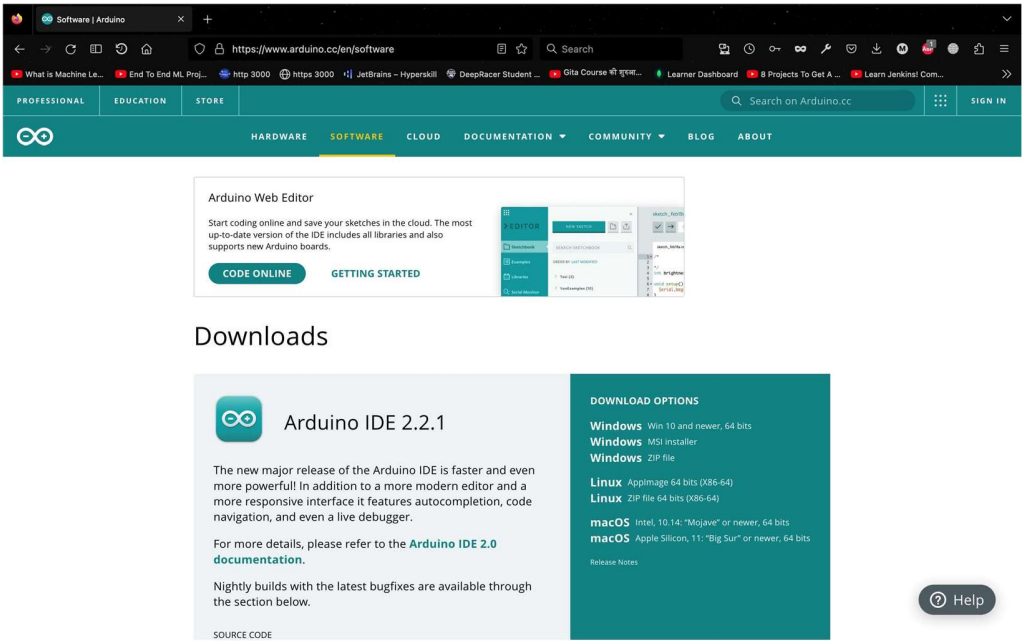
Aim -

To get familiarised with Arduino and the installation of Arduino IDE.

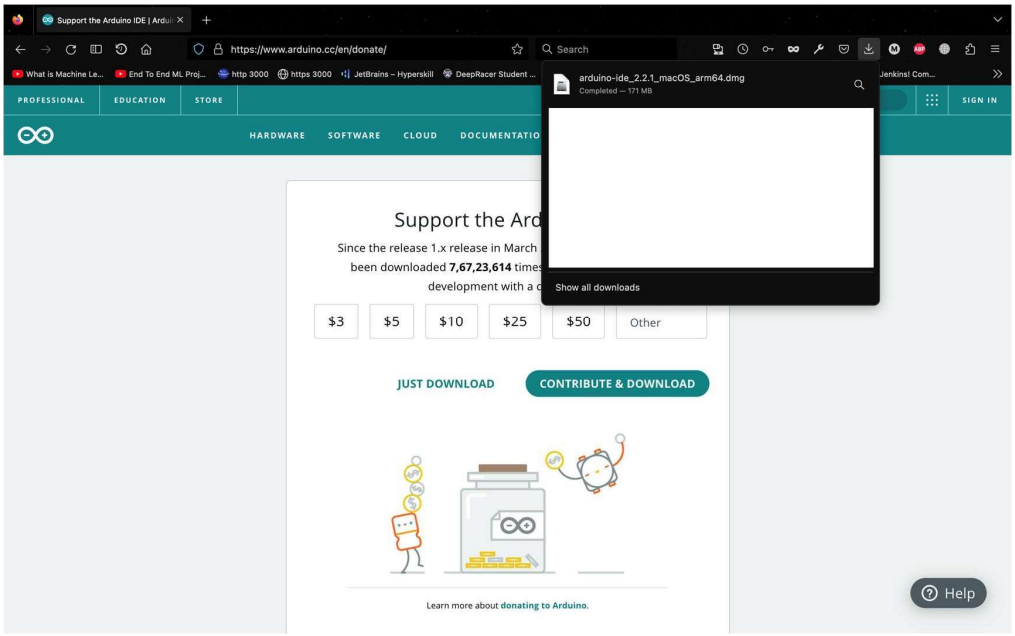
Theory -

- Arduino is an open-source electronics platform popular for its versatility in creating interactive and programmable projects. The Arduino Integrated Development Environment (IDE) is a software tool that plays a crucial role in developing projects with Arduino boards.
- Arduino is a versatile, open-source platform for creating electronics projects, and its IDE is a user-friendly software tool that simplifies the development process, making it accessible to beginners and experts alike.
- Arduino IDE is
- Cross platform.
- Open source.
- User-friendly.
- Provides a vast variety of code library.
- Have a huge community support.

- Steps to install Arduino IDE
- Go to the official website of Arduino, that is, www.arduino.cc .
- Look for the 'Software' toggle or sub-heading. Click it.
- Or, just go to this direct url www.arduino.cc/en/software .

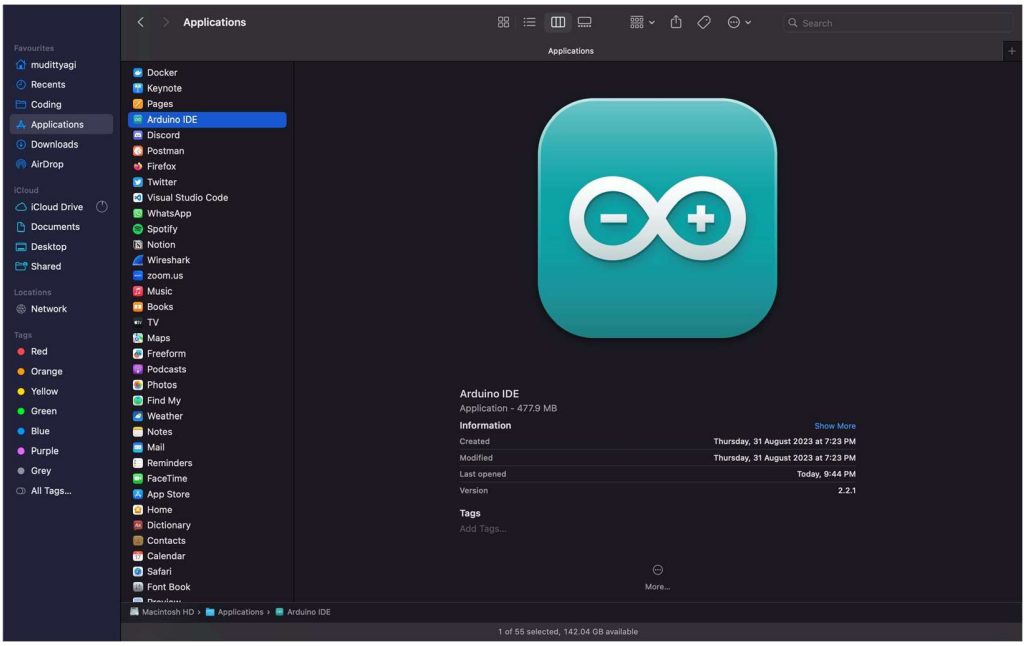


- Choose your Operating system, for me it was 'macOS Apple Silicon'.

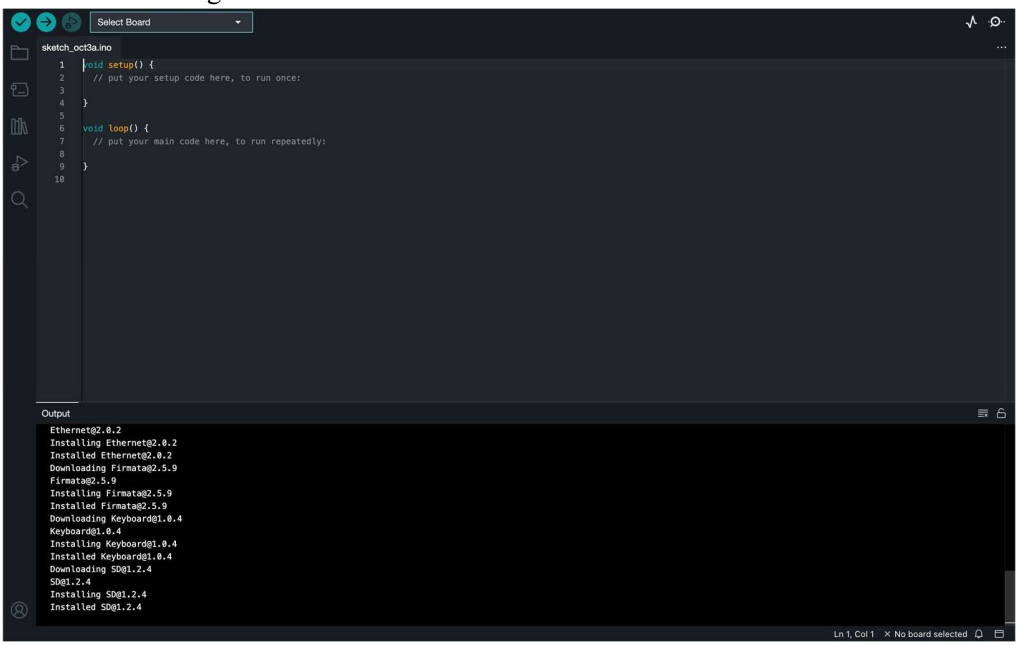


- Once you click the download button, it will start downloading as shown in the above screenshot.
- Open your downloads and double click on the downloaded file.

- Drag the icon to the Applications folder.



- Now you can see the Arduino IDE installed in your system.
- Eject the Arduino installer which is saved on the desktop.
- Double click the application icon it to open it.
- It will look something like this



- It will automatically start downloading some required packages for it to run. Give it the access to the documents folder, etc. Select the port number in which you're inserting Arduino hardware.
- Arduino successfully installed.

Viva Questions I

Q1 - What is Arduino, and what is its primary use?
Ans1 -
Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is primarily used for creating interactive and programmable electronic projects. Arduino boards can be used to sense and control objects in the physical world.

Q2 - What are the essential components required for a basic Arduino setup?
Ans2 -
The essential components for a basic Arduino setup include:
- Arduino Board
- Computer with Arduino IDE installed
- Power source (usually through USB or an external power supply)
- USB cable for connecting the Arduino to the computer

Q3 - What is the Arduino IDE, and why is it important?
Ans3 -
The Arduino IDE (Integrated Development Environment) is a software application used to write, compile, and upload code to Arduino boards. It is essential because it provides a user-friendly interface for programming Arduino, making it accessible for beginners and experienced users to develop projects.

Q4 - How do you install the Arduino IDE on a computer?
Ans4 -
To install the Arduino IDE on a computer:
- Visit the Arduino website (https://www.arduino.cc/en/software)
- Download the Arduino IDE for your operating system (Windows, macOS, or Linux).
- Install the IDE by following the installation instructions provided for your specific OS.

Q5 - What is the purpose of the Arduino sketch or program?
Ans5 -
The Arduino sketch is a program written in the Arduino IDE that defines the behaviour of the Arduino board. It specifies what the board should do, how it should respond to inputs, and how it should control outputs. In essence, it is the code that runs on the Arduino.

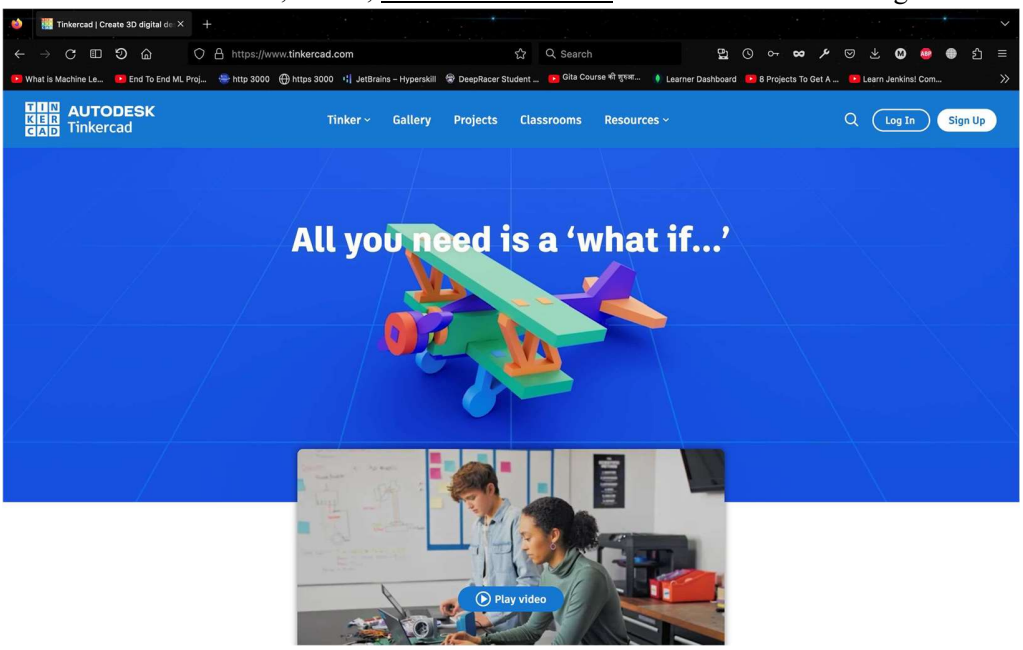
Q6 - How do you upload a sketch to an Arduino board using the Arduino IDE?
Ans6 -
To upload a sketch to an Arduino board using the Arduino IDE:
- Connect your Arduino board to the computer via a USB cable.
- Open the Arduino IDE.
- Write or open your desired sketch.
- Select the correct board and port from the "Tools" menu.
- Click the "Upload" button (an arrow icon) to compile and upload the sketch to the board.

Experiment II

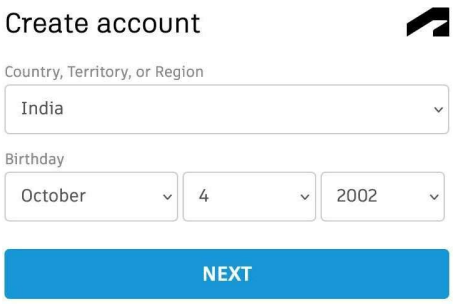
Aim -
Demonstration of AutoCAD and TinkerCAD.

Theory -

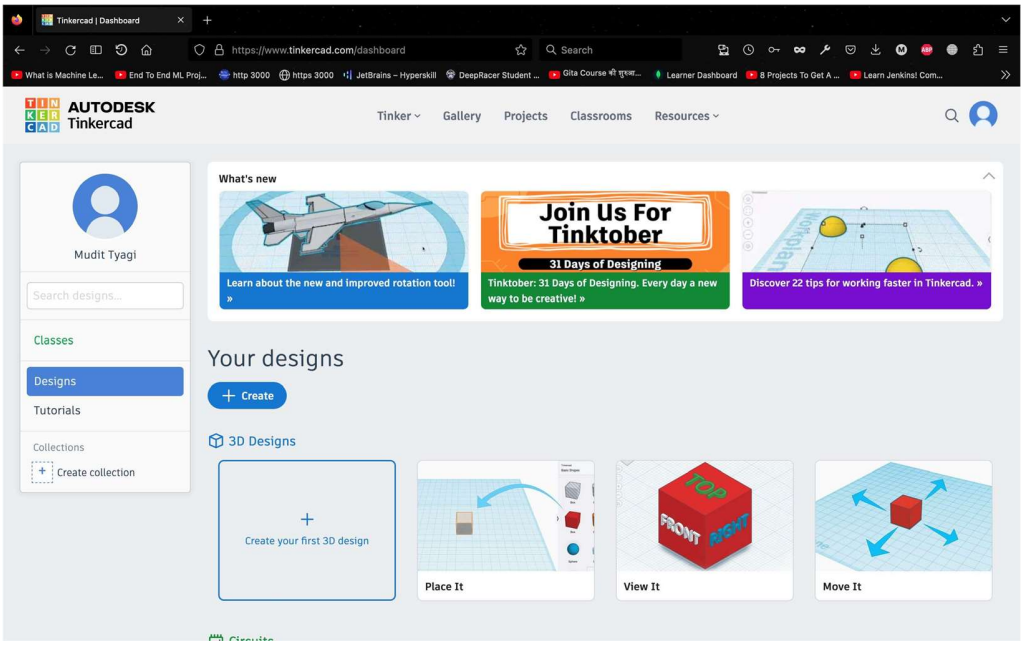
- TinkerCAD is a free-of-charge, online 3D modelling program that runs in a web browser.[1] Since it became available in 2011 it has become a popular platform for creating models for 3D printing as well as an entry-level introduction to constructive solid geometry in schools and colleges.
- Following are the steps to use TinkerCAD
- Go to there official website, that is, www.tinkercad.com . You will see something like this.



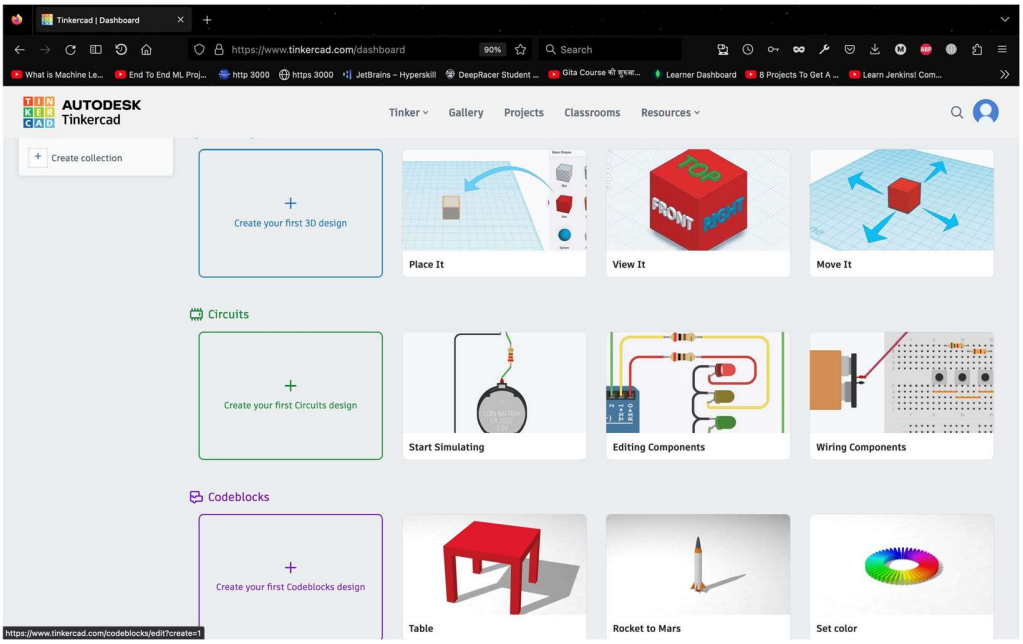
- Create an account. Or sign in using Google OAuth. Fill in your details.



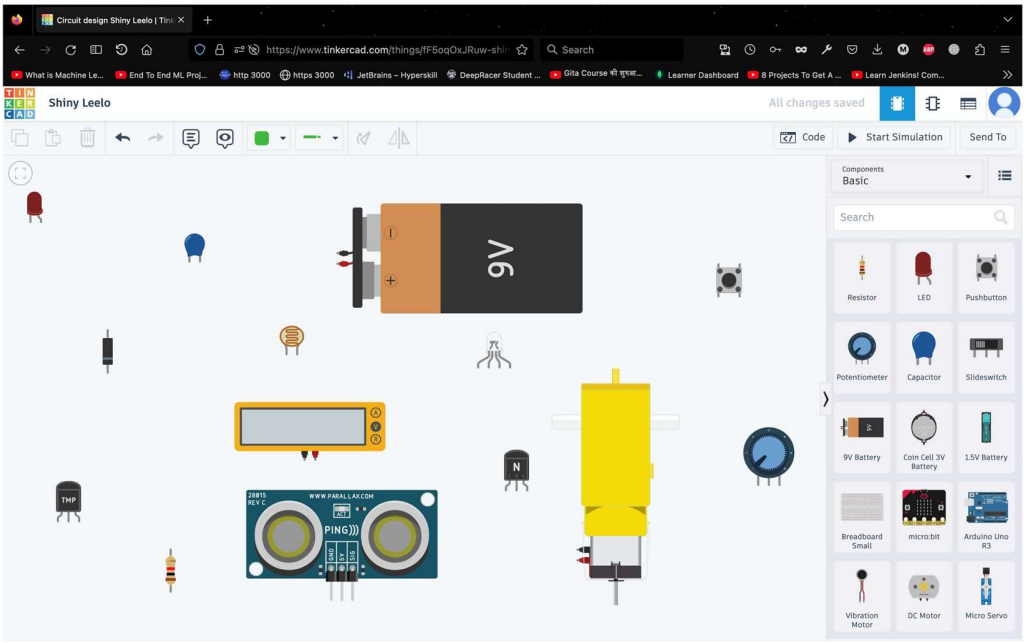
- If you already have a TinkerCAD account, you don't need to sign up for a free account.
- Once you are done with all the necessary signup work. You will see something like this.



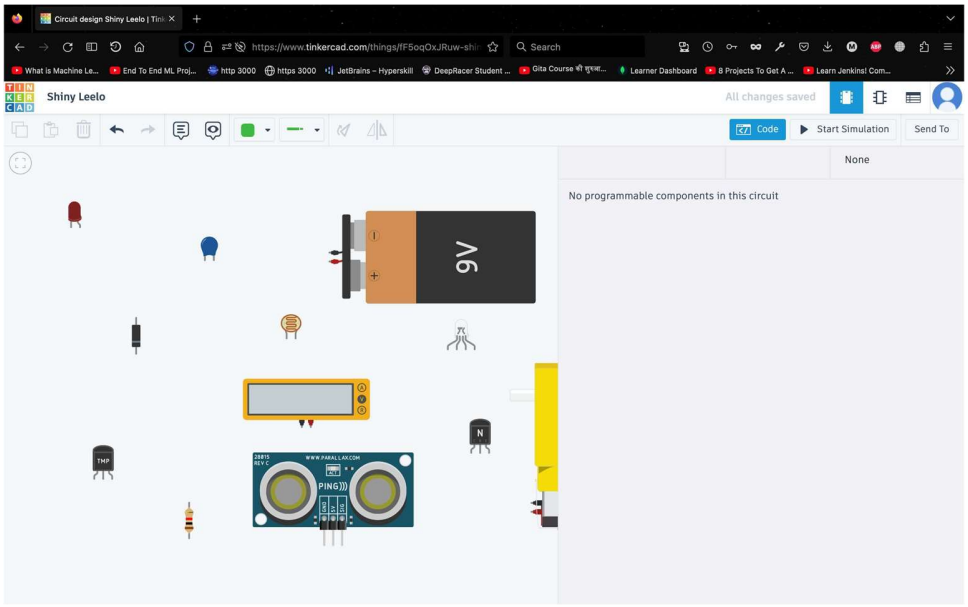
- TinkerCAD offers various project types, including 3D design, electronics, and coding.
- Scroll down, and search for ‘Circuits’. Circuits will be our main project type in this lab.



- Click on, ‘Create a new Circuit Design’. An white page will be opened, and some devices on the right side, drag and drop them one by one on to the white black page in the middle. You should see something like this.



- There is also a ‘code’ toggle on the top right corner of the screen, Though it will only get active when we have a code-able device on the white page, for example an Arduino.



Viva Questions II

Q1 - What is TinkerCAD?
Ans1 -
TinkerCAD is a free online 3D design and printing platform. It is easy to use and suitable for beginners and experts alike. TinkerCAD can be used to design simple objects, such as toys and figurines, or complex objects, such as machines and buildings.

Q2 - How does TinkerCAD work?
Ans2 -
TinkerCAD uses a block-based design system. To design an object, users simply drag and drop blocks onto the canvas. Blocks can be combined to create more complex objects.

Q3 - How to use the TinkerCAD simulator?
Ans3 -
To use the TinkerCAD simulator, simply go to the TinkerCAD website and create an account. Once you have created an account, you can start designing objects by dragging and dropping blocks onto the canvas.

Q4 - What are the features of the TinkerCAD simulator?
Ans4 -
The TinkerCAD simulator has a wide range of features, including:
- A block-based design system that is easy to use.
- A library of pre-made shapes and objects.
- The ability to import and export 3D models.
- The ability to generate code for 3D printing.

Q5 - What are the benefits of using the TinkerCAD simulator?
Ans5 -
The TinkerCAD simulator has a number of benefits, including:
- It is free to use.
- It is easy to use, even for beginners.
- It has a wide range of features.
- It can be used to design objects for 3D printing.

Experiment III

Aim -

To interface an Arduino with an LED and create a blinking application.

Requirements -

- Arduino board (Arduino Uno)
- LED
- Resistor
- Jumper and Connecting wires
- A laptop (just a laptop would also be enough - TinkerCAD).

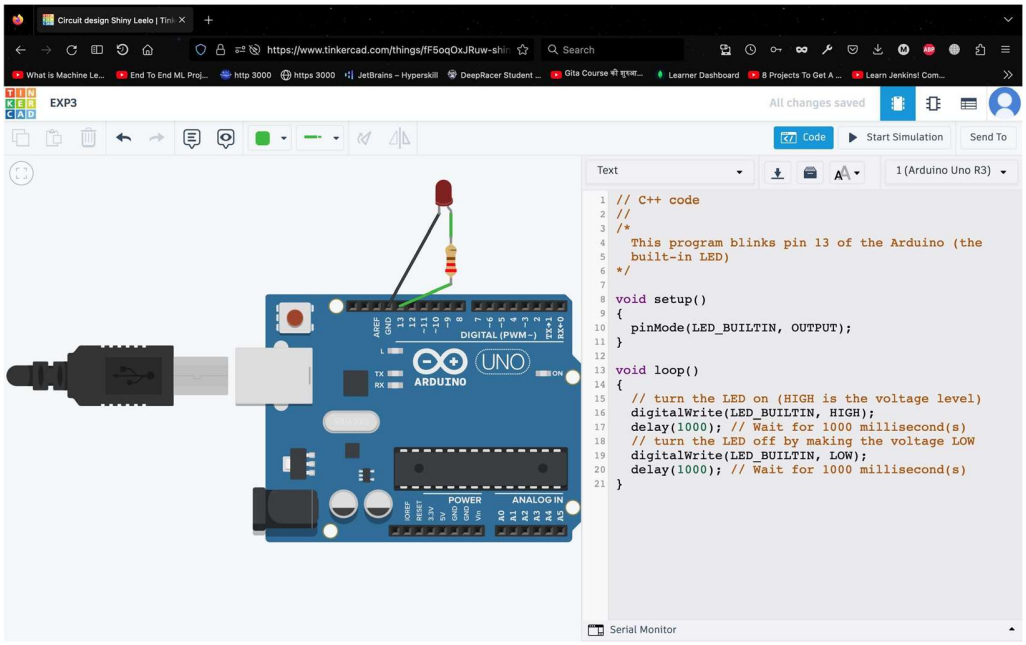
Theory -

- Open TinkerCAD, just as shown in the previous experiment.
- Create a new circuit.
- Put an LED from the right panel onto the white board in the centre.
- Now, search for an 'Arduino Uno' in the right panel. Put that onto the white board also.
- Make connections-
- LED Connections: Connect the LED to the breadboard. The LED has two legs, the longer of which is the anode (positive) and the shorter of which is the cathode (negative).
- Resistor Connection: Insert one end of the resistor into the same row of the breadboard as the LED's Anode. The resistor's other end should be connected to the Arduino's digital output pin.
- Ground (GND) Connection: Connect a jumper wire from the same row as the LED's cathode to any Arduino board GND (Ground) pin. This connects the circuit to the ground of the Arduino.
- Working Procedure-
- setup() and loop() are two fundamental Arduino functions for controlling the behaviour of your board.
- The setup() function is only called once when the Arduino board boots up or is reset. Its goal is to set pin modes, initialise variables, and execute any other necessary setup tasks before the main loop begins. This function can be used to configure settings that should only be changed once over the board's lifespan.
- The loop() function is the heart of an Arduino program. After the setup() function is executed, the loop() function starts running repeatedly until the Arduino is powered off or reset. It contains the main code that performs the desired tasks, controls the board, user input. Whatever is included in the loop() function will be executed in a continuous loop, allowing the Arduino to perform its intended functions continuously.

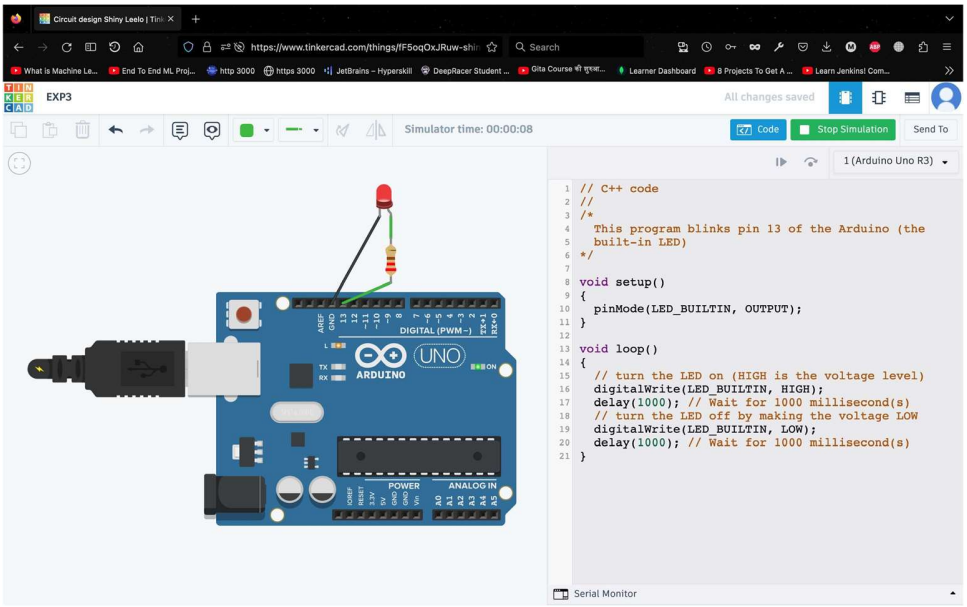
Code - `void`

```
setup()
{
  pinMode(LED_BUILTIN,  OUTPUT);
}
void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);// turn the LED on
  delay(1000);                    // delay of 1 second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off
  delay(1000);                    // delay of 1 second
```

}



- Toggle the code section, which is on the top right corner of the screen as shown in the above photo.
- Write the necessary code, which decides the delay and setup of the Arduino and LED.
- Click on the ‘Start simulation’ button, which is just next to code button on the top right corner.
- Once you click it, The LED will start blinking on and off after 1 second delay.



Viva Questions III

Q1 - What does the int ledPin = 13; line do?
Ans1 -
It defines a variable named ledPin and assigns it the value 13, which corresponds to the digital pin connected to the built-in LED on most Arduino boards.

Q2 - Why is pinMode(ledPin, OUTPUT); used in the setup() function?
Ans2 -
It sets the ledPin as an output, indicating that it will be used to send signals to the LED.

Q3 - What does digitalWrite(ledPin, HIGH); do in the loop() function?
Ans3 -
It turns the LED on by setting the voltage on ledPin to HIGH, which supplies power to the LED.

Q4 - How long does the LED stay on and off in this program?
Ans4 -
The LED stays on for 1 second (1000 milliseconds) and off for 1 second in an alternating pattern.

Q5 - Can this program be used with a Raspberry Pi?
Ans5 -
No, this program is written for Arduino and is not compatible with a Raspberry Pi. Raspberry Pi uses different programming languages and libraries for GPIO control.

Experiment IV

Aim -
To print “Hello World!” using Arduino and the Serial monitor.

Requirements - •
Arduino board.
• Arduino’s USB connector.

Theory -

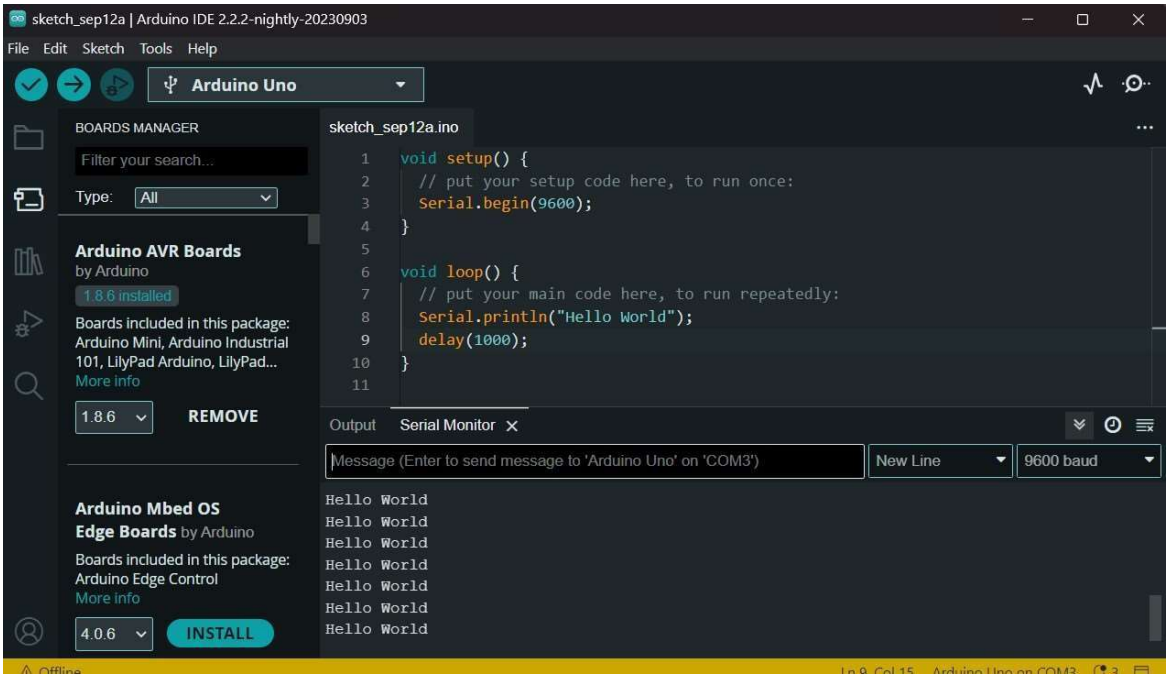
- Baud rate is the rate at which the number of signal elements or changes to the signal occurs per second when it passes through a transmission medium. The higher a baud rate is the faster the data is sent/received.
- **Serial** is used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART), and some have several.
- **Serial.begin()** - Sets the data rate in bits per second (baud) for serial data transmission.
- **Serial.println()** - Prints data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n').

- Steps
- Open the Arduino IDE, which we installed in experiment 1.
- Connect the Arduino Uno (Hardware) to your laptop using a USB cable.
- Select the port in which you just inserted the Arduino hardware, so that the IDE can successfully interact with the Arduino hardware inserted in that port.
- Write the code/sketch in your IDE, and upload it once completed.
- After uploading the code/sketch, Look for ‘Tools’ toggle on the top left corner of the IDE interface.
- After clicking Tools, you should see a button saying ‘Serial Monitor’. Click it.
- After clicking, a type of interface should pop up from bottom of the screen which should look similar command line interface of Visual studio code.
- On the top right corner of this interface, set the baud rate to ‘9600’, which is similar to the baud rate written in our code/sketch.
- After that, we can see "Hello, World!" messages printed to the Serial Monitor with a 1-second interval between each message.

Code -

```
void setup()
{
  // Initialise the serial communication at a baud rate of 9600
  Serial.begin(9600);
}
void loop()
{

  // Send "Hello, World!" to the Serial Monitor
  Serial.println("Hello, World!");

  // Delay for 1 second (1000 milliseconds)
```

```
  delay(1000);
}
```


sketch_sep12a | Arduino IDE 2.2.2-nightly-20230903
File Edit Sketch Tools Help
Arduino Uno
BOARDS MANAGER
Filter your search...
Type: All
Arduino AVR Boards
by Arduino
1.8.6 installed
Boards included in this package: Arduino Mini, Arduino Industrial 101, LilyPad Arduino, LilyPad...
More info
1.8.6
REMOVE
Arduino Mbed OS Edge Boards by Arduino
Boards included in this package: Arduino Edge Control
More info
4.0.6
INSTALL
sketch_sep12a.ino
1 void setup() {
2 // put your setup code here, to run once:
3 Serial.begin(9600);
4 }
5
6 void loop() {
7 // put your main code here, to run repeatedly:
8 Serial.println("Hello World");
9 delay(1000);
10 }
11
Output Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM3')
New Line
9600 baud
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Offline
Ln 9, Col 15 Arduino Uno on COM3

Viva Questions IV

Q1 - What is the Internet of Things (IoT)?
Ans1 -
IoT refers to the network of interconnected physical devices or "things" that can collect and exchange data over the internet.

Q2 - What are some examples of IoT applications?
Ans2 -
Examples include smart home devices (thermostats, lights), industrial sensors, wearable fitness trackers, and smart city infrastructure (traffic monitoring).

Q3 - What is the role of sensors in IoT devices?
Ans3 -
Sensors in IoT devices capture data from the physical world, such as temperature, humidity, or motion, and transmit it for analysis and decision-making.

Q4 - What is MQTT in IoT communication?
Ans4 -
MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol used for efficient communication between IoT devices and servers.

Q5 - How does IoT enhance efficiency in industries?
Ans5 - IoT can improve efficiency by providing real-time data monitoring, predictive maintenance, and automation, leading to reduced downtime and cost savings.

EXPERIMENT - 5

AIM:

Identification of different sensors used in IoT applications.

OBJECTIVE:

1. To study hardware related to IoT.
2. To understand and identify the different sensors used in IoT.

THEORY:

TABLE I. IOT APPLICATIONS AND TYPES OF SENSORS USED IN IT

IoT Applications	Type of Sensors
Smart City	Velocity, Light, Accelerometer, Position, Temperature, Proximity, Humidity, Pressure, Infrared
Smart Environment	Light, Temperature, Humidity, Chemical, Gyroscope, Bio Sensors, Chemicals, Accelerometer, Optical
Smart Water	Temperature, Humidity, Occupancy, Water Quality
Smart Building	Light, Accelerometer, Chemical, Gyroscope, Magneto
Smart Health	Light, Gyroscope, Biosensors, Chemicals, Magneto, Accelerometer, Pressure
Smart Home	Light, Gyroscope, Biosensors, Chemicals, Magneto, Accelerometer, Temperature, Proximity, Position, Infrared
Smart Transport	Gyroscope, Pressure, chemicals, Magneto, Accelerometer, Temperature, Motion, Infrared
Smart Security	Light, Gyroscope, Chemical, Magneto, Accelerometer, Temperature, Infrared
Smart Agriculture	Temperature, Humidity, Water Quality, Chemical, Proximity, Position

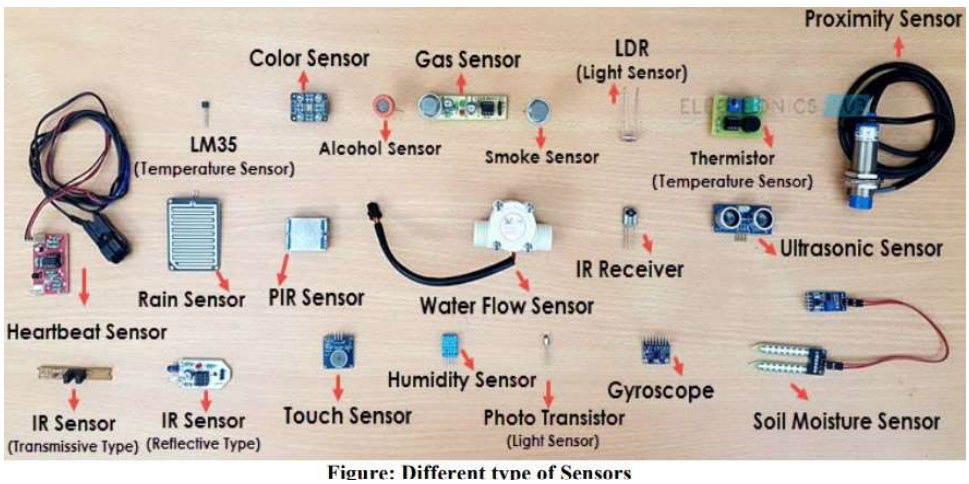


Figure: Different type of Sensors

1. **Temperature Sensors:**
   This technology represents a big role in the IoT sensing market. Temperature sensors provide real-time readings of temperatures, they provide an output signal that is proportional to the temperature it measures.

2. **Proximity Sensor:**
   A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact. A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance), and looks for changes in the return signal. The object being sensed is often referred to as the proximity sensor's target.

3. **Pressure sensor:**
   An IoT pressure sensor is any device that senses pressure and converts it into an electric signal. The level of voltage given out by the sensor depends on the level of pressure applied. These sensors enable IoT systems that monitor systems and devices that are pressure propelled. If there's any deviation from standard pressure ranges, the device notifies the administrator of the problem.

4. **Water Quality Sensor:**
   IoT sensors are used to measure various parameters of water quality, such as pH, temperature, dissolved oxygen, and the presence of chemicals and microorganisms. These sensors can be placed in rivers, lakes, and other bodies of water, and they can transmit data in real-time to a central monitoring system.

5. **Chemical Sensor:** These are devices that detect and measure the presence or concentration of specific chemicals in a gaseous or liquid environment. They are used in a wide variety of applications, including environmental monitoring, industrial process control, and medical diagnostics.

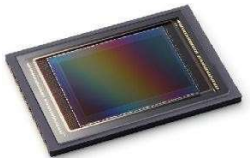
6. **Gas Sensor**: It detect the presence and concentration of various hazardous gasses such as explosive gasses, Volatile Organic Compounds (VOCs), odor, humidity and so on.

7. **Smoke sensor**:
   A smoke detector is an electronic fire-protection device that automatically senses the presence of smoke, as a key indication of fire, and sounds a warning to building occupants.

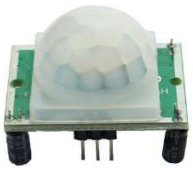
8. **IR Sensors:**
   IR sensors are now widely used in motion detectors, which are used in building services to switch on lamps or in alarm systems to detect unwelcome guests. In a defined angle range, the sensor elements detect the heat radiation (infrared radiation) that changes over time and space due to the movement of people.

9. **Level Sensor**:
   These are used to measure the position of the surface within some form of container of both liquids and solid materials in the form of powders.

10. **Image Sensor**:
    An image sensor or imager is a sensor that detects and conveys information used to form an image. It does so by converting the variable attenuation of light waves (as they pass through or reflect off objects) into signals, small bursts of current that convey the information.



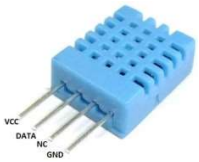
11. **Motion Detection Sensors**:
    A motion detector is an electrical device that utilizes a sensor to detect nearby motion. Such a device is often integrated as a component of a system that automatically performs a task or alerts a user of motion in an area. They form a vital component of security, automated lighting control, home control, energy efficiency, and other useful systems.



12. **Accelerometer Sensor:**
    Accelerometer sensors are ICs that measure acceleration, which is the change in speed (velocity) per unit time. Measuring acceleration makes it possible to obtain information such as object inclination and vibration.

13. **Gyroscope Sensor:**

A gyroscope is a device that uses Earth's gravity to help determine orientation. Gyro sensors are devices that sense angular velocity which is the change in rotational angle per unit of time. Angular velocity is generally expressed in deg/s (degrees per second).



14. **Humidity Sensor**:
    Humidity sensors are electronic devices that measure and report the moisture and air temperature of the surrounding environment where they are deployed e.g., in air, soil, or confined spaces. Humidity measurements indicate the concentration of water vapor presented in the air.

15. **Optical Sensors**:
    An optical sensor converts light rays into electronic signals. It measures the physical quantity of light and then translates it into a form that is readable by an instrument. An optical sensor is generally part of a larger system that integrates a source of light, a measuring device and the optical sensor.

EXPERIMENT - 6

AIM:

To demonstrate LED blinking using a push button on Arduino UNO.

THEORY:

Components Required:

1 * Breadboard

1* Arduino Uno R3

1 * LED

2* 330 ohm Resistor

6 * Jumper

1 * Push Button

PROCEDURE :

1.) Set up the Hardware:

- Connect one end of the LED to digital pin 8 on the Arduino Uno (via a current-limiting resistor).
- Connect the other end of the LED to the ground (GND) on the Arduino.
- Connect one leg of the push button to digital pin 7 on the Arduino.
- Connect the other leg of the push button to the ground (GND) on the Arduino.

2.) Write the Arduino Sketch:

- Open the Arduino IDE on your computer.
- Write the Arduino sketch (code) to control the LED based on the state of the push button. You've already provided the code in your previous message.

3.) Upload the Code:

- Connect your Arduino Uno to your computer using a USB cable.
- Select the correct board (Arduino Uno) and COM port in the Arduino IDE.
- Click the "Upload" button to upload the code to the Arduino.

4.) Test the Circuit:

- Once the code is uploaded, press and release the push button to see the LED at pin 8 turn on when the button is pressed and turn off when it's released.

CODE:

```
void setup()
{
 pinMode(13, OUTPUT);
 pinMode(2, INPUT);
}

void loop()
{
 if(digitalRead(2)==HIGH)
 {
  digitalWrite(13, HIGH);
 }
 else{
 digitalWrite(13, LOW);
 }
}
```

Circuit Diagram:

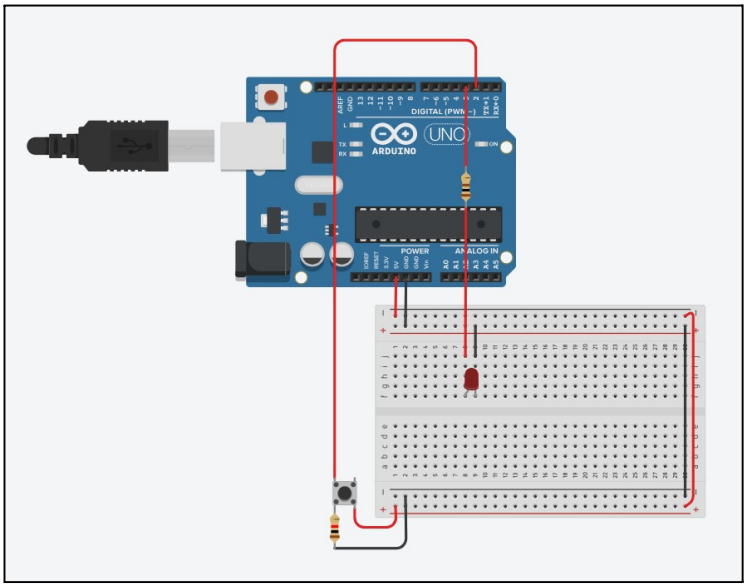
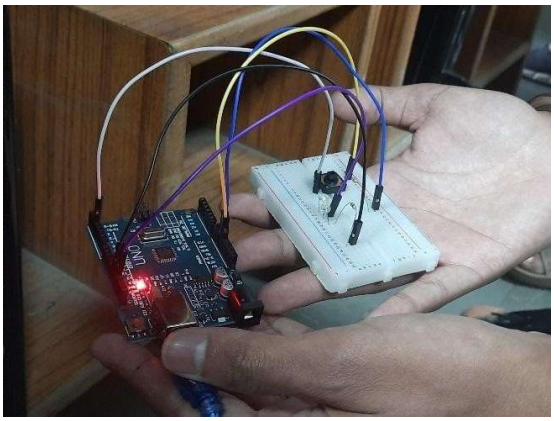


fig: Circuit Diagram

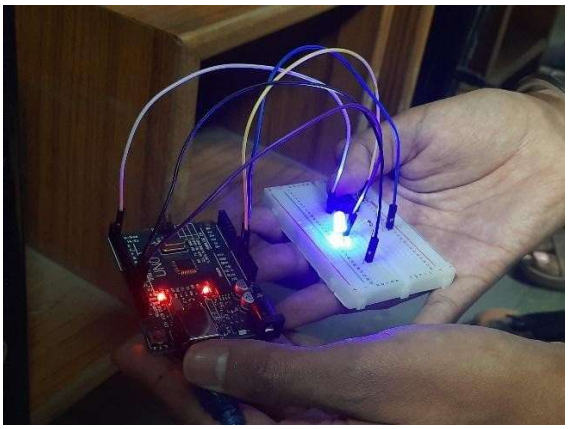
OUTPUT

Here, we can see the LED light blinking using the pushbutton on Arduino UNO.

The first figure(A) shows the LED light is turned off and the second figure(B) shows that, when we click on the pushbutton the LED light glows up.



Figure(A)



Figure(B)

VIVA QUESTIONS

1. **What are some security challenges associated with IoT devices?**
   IoT security challenges include data privacy, device authentication, and protection against cyberattacks.
2. **Explain the concept of IoT edge computing.**
   IoT edge computing involves processing data locally on IoT devices or at the edge of the network, reducing latency and bandwidth usage.
3. **How do IoT devices typically connect to the internet?**
   IoT devices can connect via Wi-Fi, cellular networks, Ethernet, Bluetooth, or Low- Power Wide Area Networks (LPWANs).
4. **What is the significance of data analytics in IoT applications?**
   Data analytics in IoT helps extract valuable insights from large volumes of data, enabling better decision-making and predictive analysis.
5. **Name an example of a smart city application using IoT technology.**
   Smart traffic management systems that use IoT sensors to monitor and optimize traffic flow in real-time.

Experiment - 7

Aim: Interfacing of Arduino/Raspberry Pi with temperature and humidity sensor with real time application.

Components Required:

1 * Breadboard

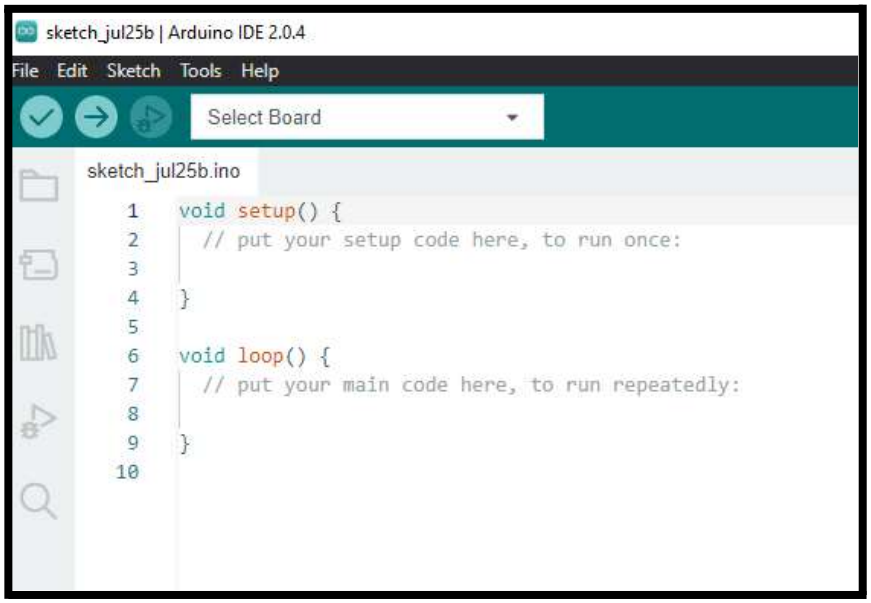
1 * Arduino Uno R3

1 * DH11 Sensor

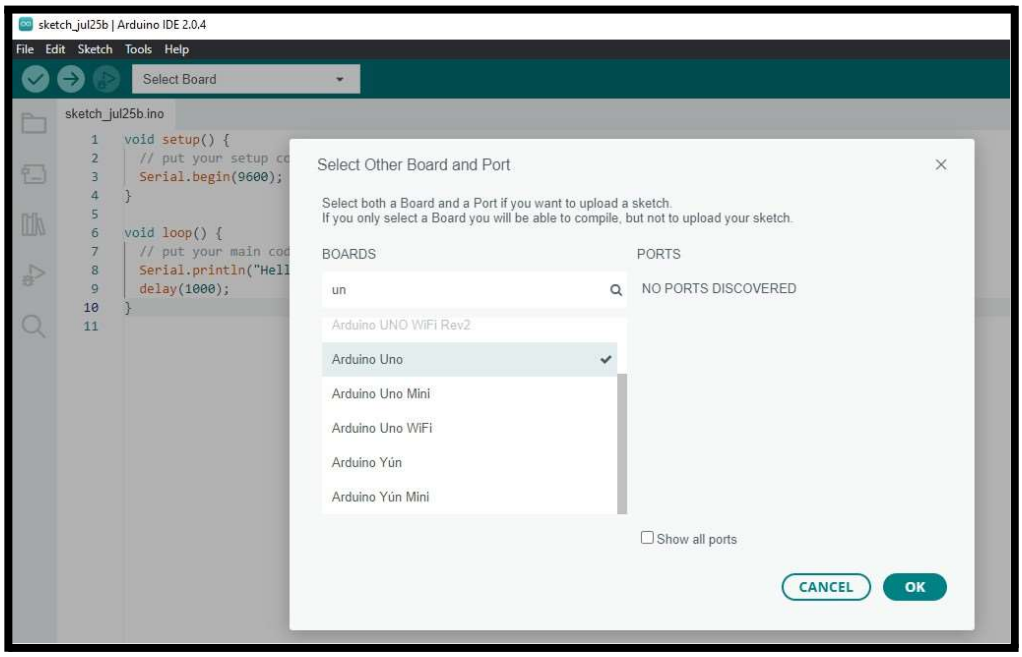
4 * Jumper Wires

Procedure:

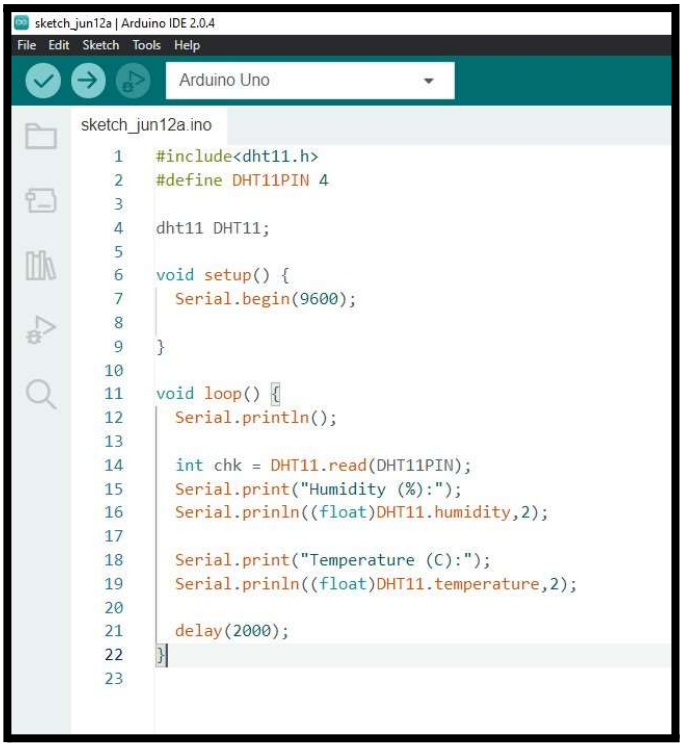
1) Open Arduino IDE



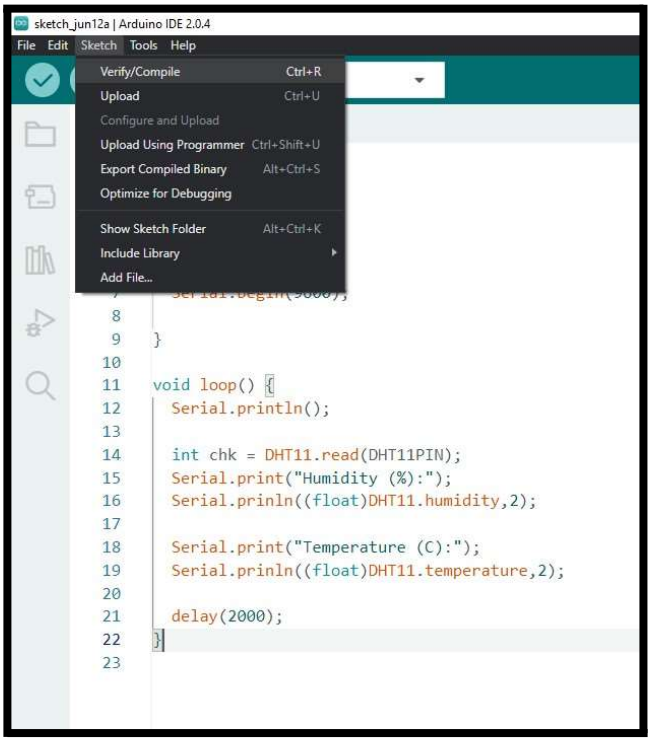
2) Select board and port to which the Arduino Board is connected .



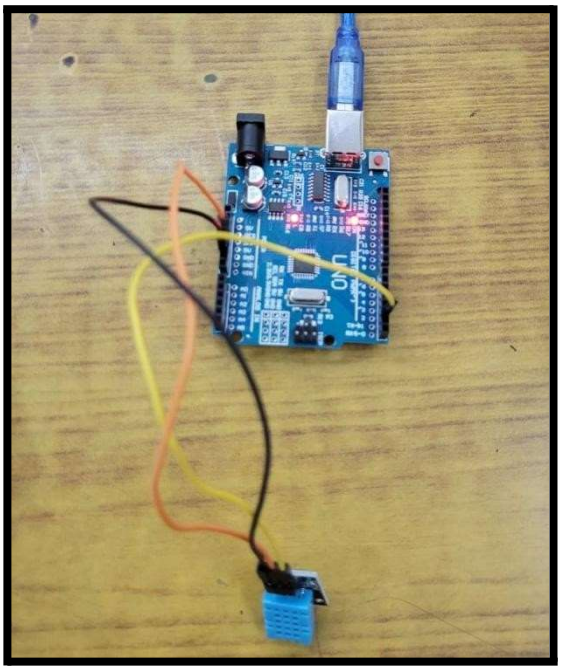
3) Write the code for temperature and humidity sensor



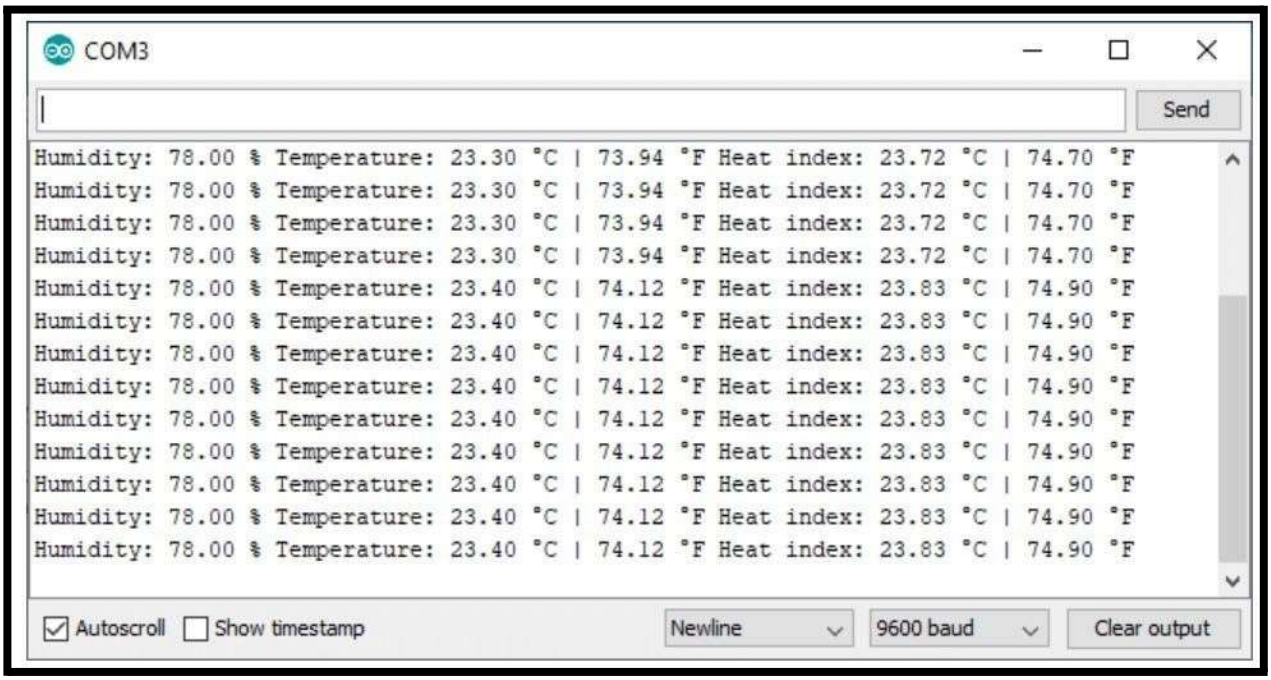
4) Click on sketch tab then select verify/compile or press Ctrl + R



5) The connection with Arduino should be as follows.



6) The output will be:



Result: We can get the humidity and temperature with the help of DHT11.

Viva Questions:

Q1) What does DHT11 stand for?

Answer: DHT11 stands for "Digital Humidity and Temperature Sensor."

Q2) How do you connect the DHT11 sensor to an Arduino?

Answer: You connect the DHT11's data pin to a digital input/output pin on the Arduino, supply it with 5V power, and connect its ground (GND) pin.

Q3) What library do you typically use to interface with the DHT11 sensor in Arduino?

Answer: The "DHT" library, often used with "Adafruit_DHT" library, is commonly used to interface with DHT11 and similar sensors in Arduino.

Q4) How can you read humidity and temperature values from a DHT11 sensor in Arduino code?

Answer: You can use the DHT.readHumidity() and DHT.readTemperature() functions from the library to obtain humidity and temperature values, respectively.

Q5) What type of data does the DHT11 sensor provide, analog or digital?

Answer: The DHT11 sensor provides digital data, making it easy to interface with digital pins on the Arduino.

Experiment - 8

Aim: To measure the distance of an object using an ultrasonic sensor.

Components Required:

1 * Breadboard

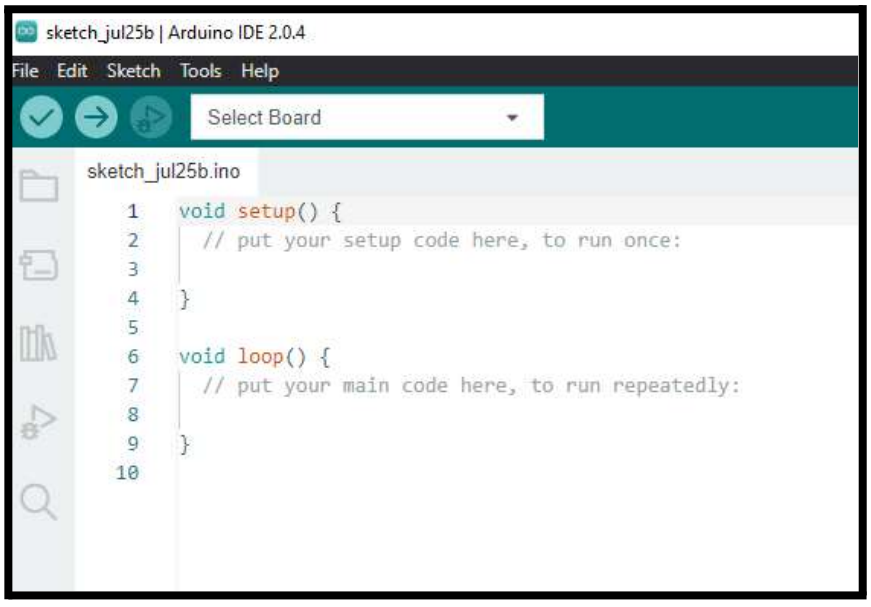
1 * Arduino Uno R3

1 * Ultrasonic Sensor

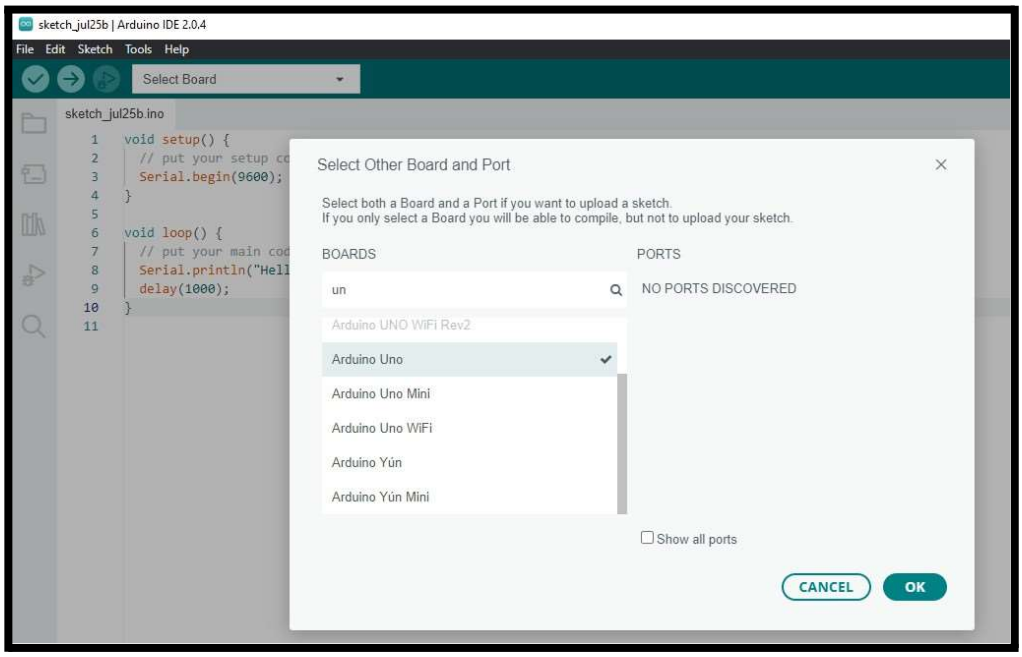
8 * Jumper Wires

Procedure:

1) Open Arduino IDE



2) Select board and port to which the Arduino Board is connected .

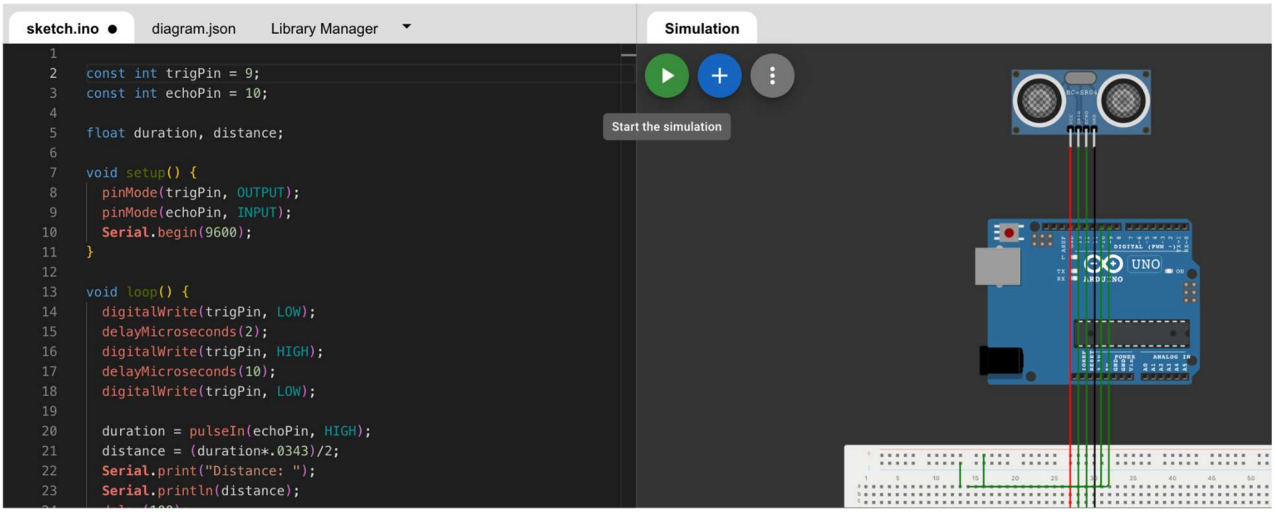


3) Write the code for the Ultrasonic sensor HC-SR04.

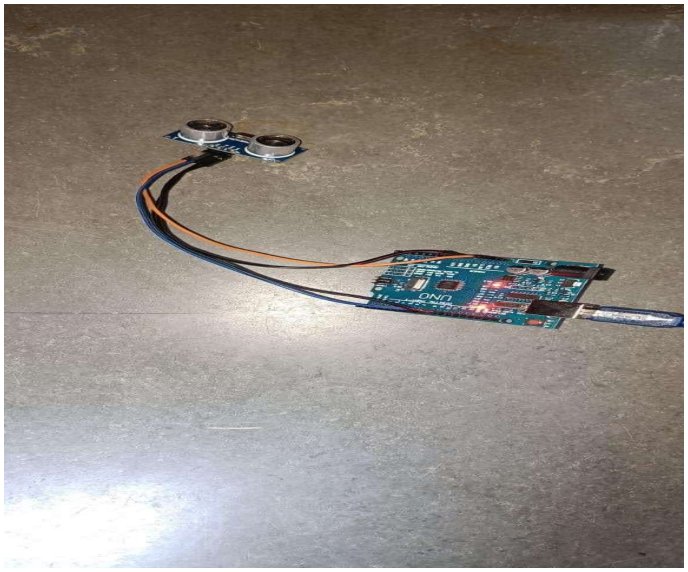
sketch.ino ● diagram.json Library Manager

```
const int trigPin = 9;
const int echoPin = 10;

float duration, distance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration*.0343)/2;
  Serial.print("Distance: ");
  Serial.println(distance);
  delay(100);
}
```

4) Click on the simulation button to compile and verify the code.



5) The connection with Arduino should be as follows.



6) The output will be:

```
Distance: 53.56
Distance: 53.13
Distance: 53.99
Distance: 53.15
Distance: 53.06
Distance: 53.56
Distance: 53.58
Distance: 54.33
```

Viva Questions:

Q1) What is the basic principle of operation for an ultrasonic sensor in distance measurement?

Answer: Ultrasonic sensors work on the principle of sending high-frequency sound waves and measuring the time it takes for the waves to bounce back after hitting an object. Distance is calculated based on the time delay.

Q2) What is the typical range of measurement for an ultrasonic sensor?

Answer: The typical range for ultrasonic sensors can vary, but it is often in the range of 2 cm to 5 meters, depending on the specific sensor model.

Q3) How does an ultrasonic sensor handle distance measurement in different environmental conditions, such as temperature variations or humidity?

Answer: Ultrasonic sensors may be affected by temperature and humidity changes, which can impact the speed of sound. Many sensors are equipped with compensation mechanisms to adjust for these environmental factors and maintain accuracy.

Q4) What is the 'dead zone' in an ultrasonic sensor, and why is it important to consider in applications?

Answer: The 'dead zone' refers to the minimum distance within which an ultrasonic sensor cannot accurately measure objects. It occurs due to the time it takes for the sensor to switch from transmitting to receiving mode. It's important to consider in applications because objects within the dead zone will not be detected or measured accurately.

Q5) How can interference and multi-path reflections be minimized in ultrasonic distance measurement applications?

Answer: Interference and multi-path reflections can be reduced by using features like beam shaping, damping materials, or using multiple sensors with synchronized firing to distinguish between direct and reflected signals. Additionally, filtering and signal processing techniques can help mitigate interference.

Experiment - 9

Aim: To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when '1'/'0' is received from a smartphone using Bluetooth.

Components Required:

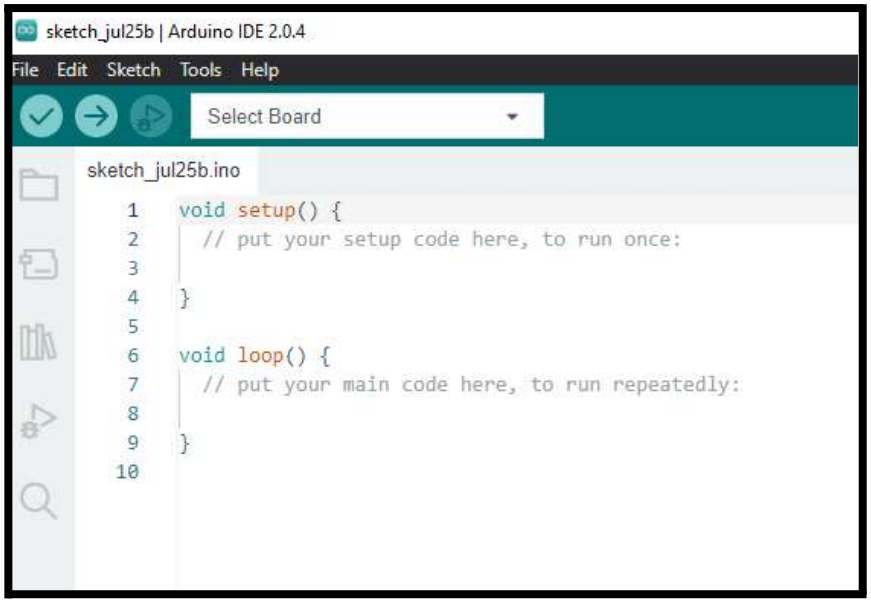
1 * Arduino Uno R3

1 * Bluetooth Sensor

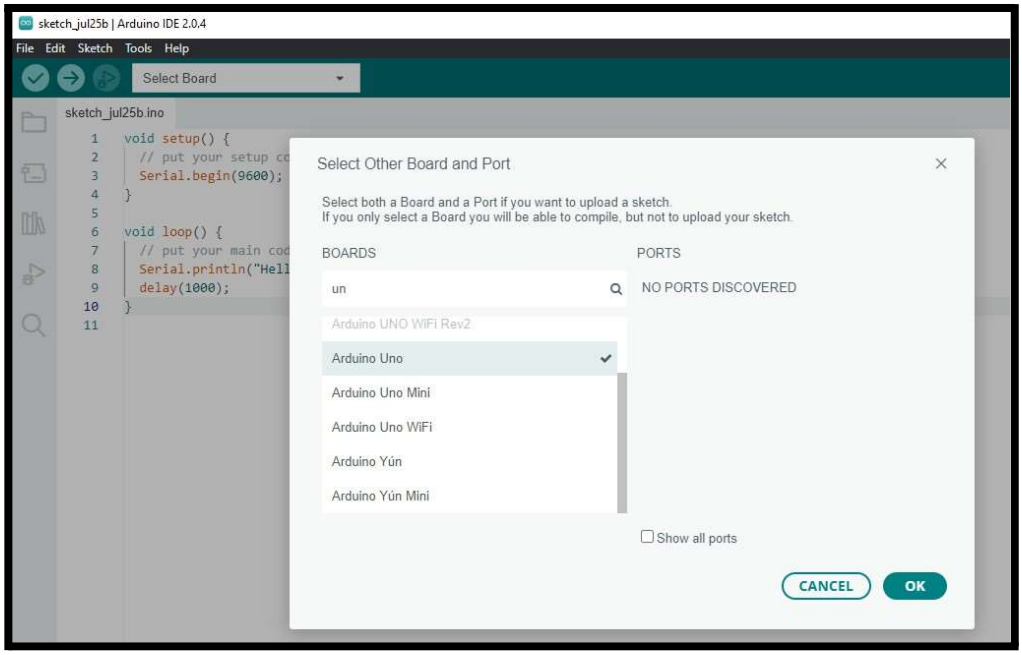
4 * Jumper Wires

Procedure:

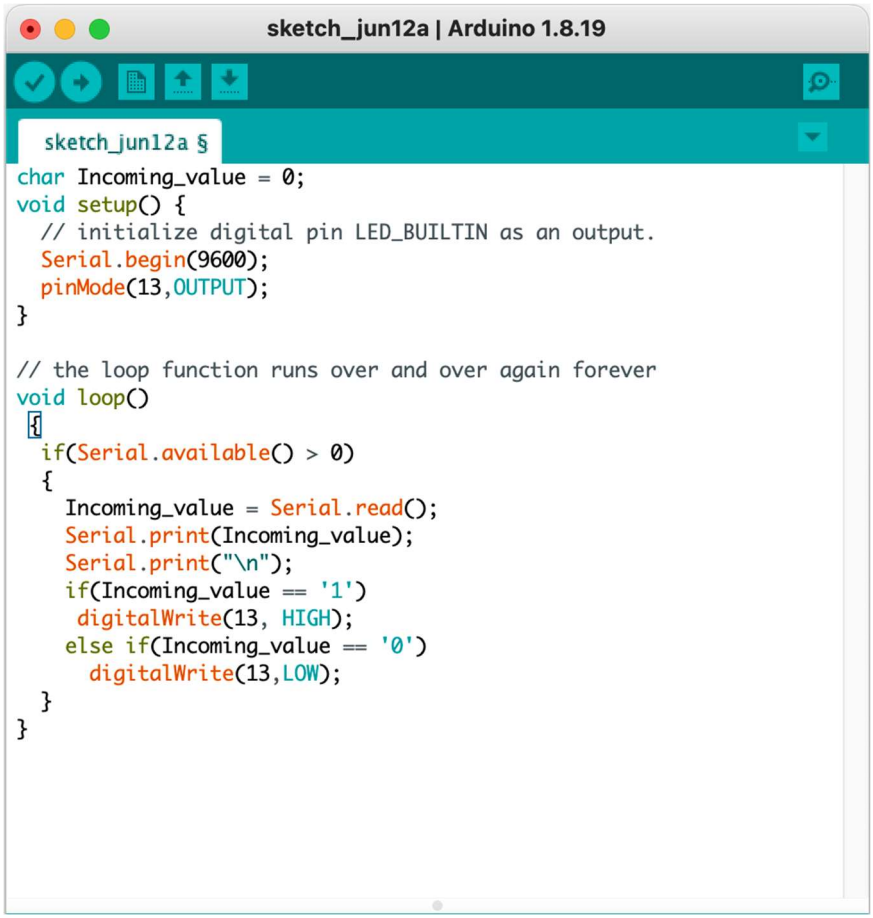
1) Open Arduino IDE



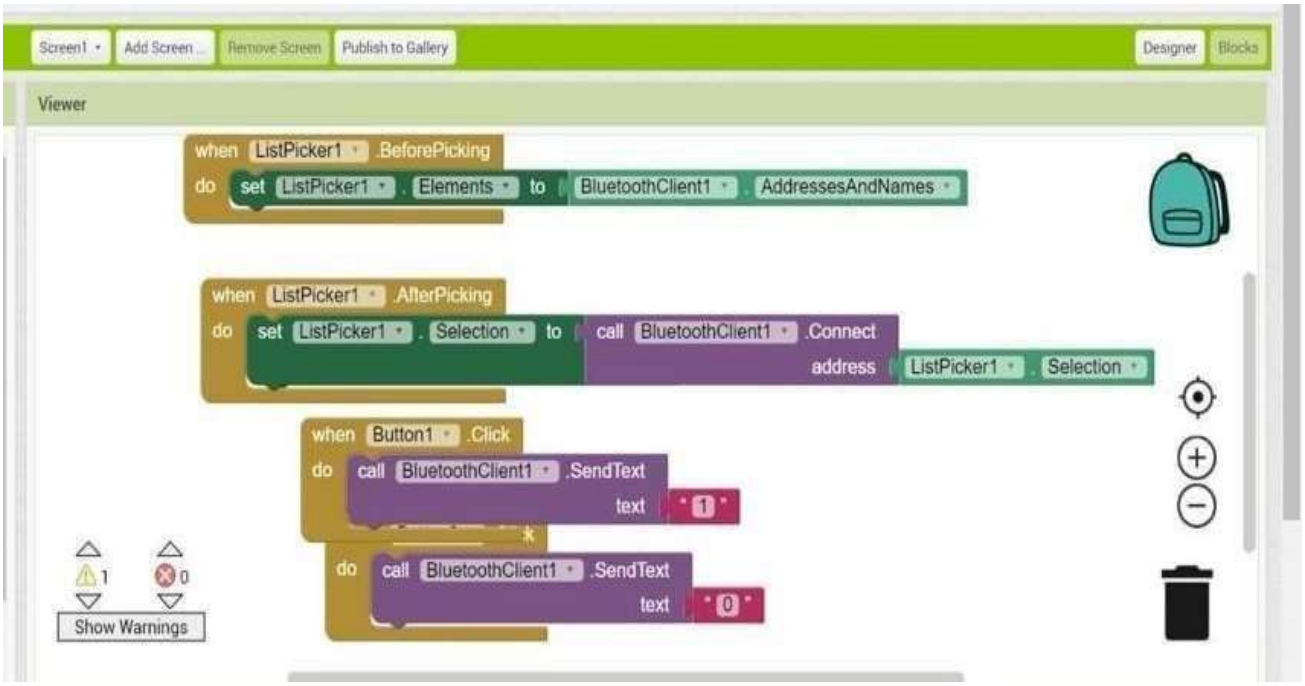
2) Select board and port to which the Arduino Board is connected .

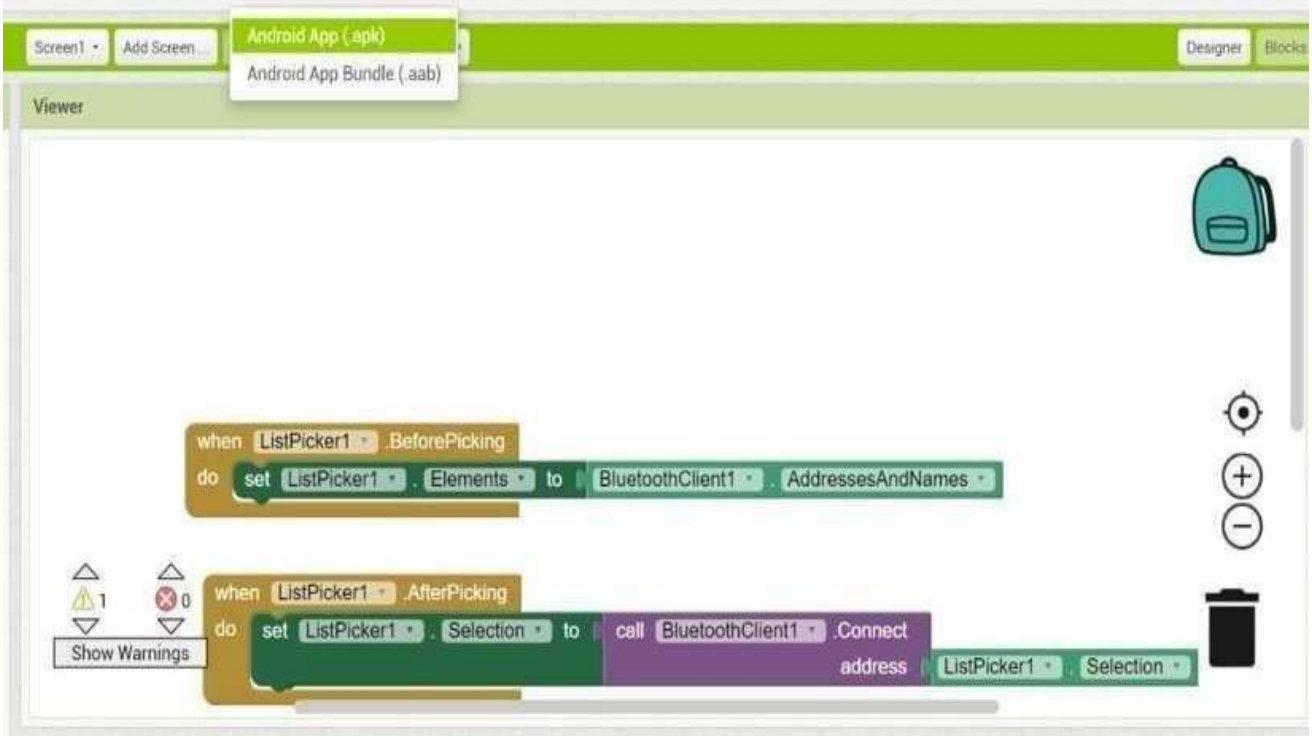


3) Write the code for connecting the Bluetooth module to Arduino.

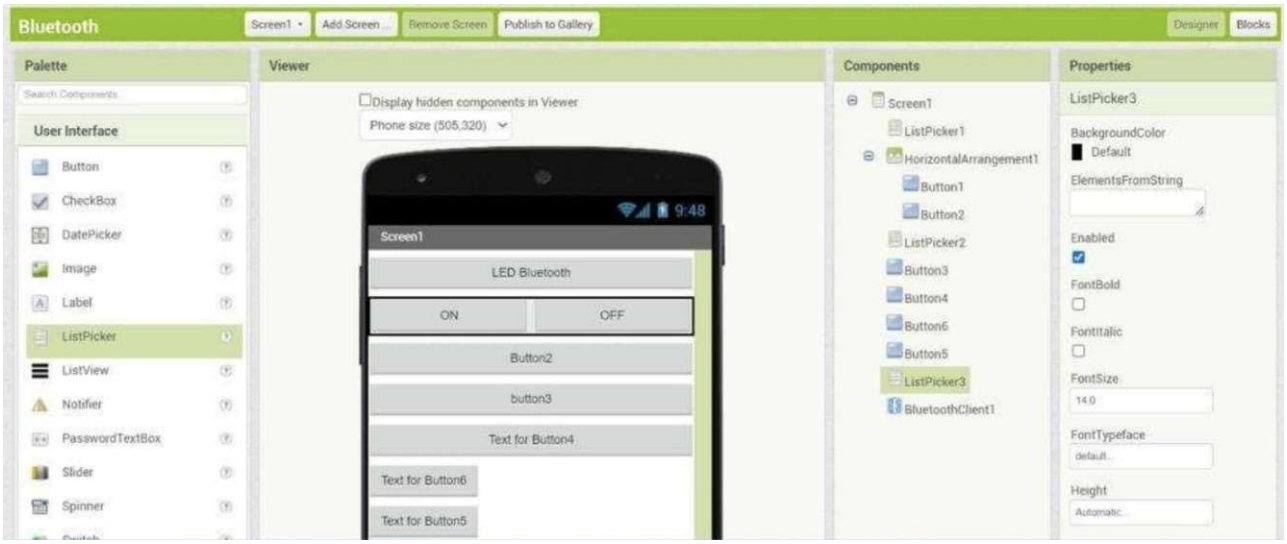


4) We created an app using the MIT app inventor. We can create an app using just drag and drop.



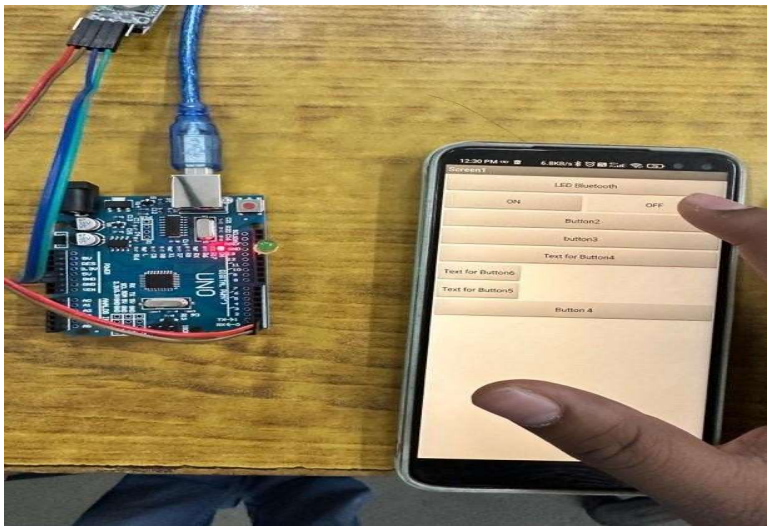


5) After that we can preview our app.



6) We can simply download the app in our mobile by generating QR code. Scan the QR code in android mobile and download the app.

7) Now we have to connect our mobile to Arduino board and we can now control the led.



Viva Questions:

Q1) How can Bluetooth be interfaced with Arduino or Raspberry Pi for communication?
Ans) One can use Bluetooth modules such as HC-05 for Arduino or Bluetooth dongles for Raspberry Pi to establish a wireless communication link.

Q2) What programming language is commonly used to write a program for Arduino or Raspberry Pi in this scenario?
Ans) Arduino typically uses the Arduino programming language, while Raspberry Pi commonly employs Python for such applications.

Q3) How would you configure the Bluetooth module to establish a connection between the smartphone and the Arduino/Raspberry Pi?
Ans) Set the Bluetooth module in a discoverable mode and pair it with the smartphone through the Bluetooth settings on the phone.

Q4) What is the purpose of the program you would write for the Arduino/Raspberry Pi in this project?
Ans) The program would involve listening for incoming Bluetooth signals and, upon receiving '1', turn the LED on, and upon receiving '0', turn the LED off.

Q5) Can you briefly explain the steps involved in the program to control the LED based on Bluetooth signals?
Ans) The program would include initializing the Bluetooth module, establishing a connection, continuously checking for incoming data, and based on the received '1' or '0', controlling the state of the LED accordingly.