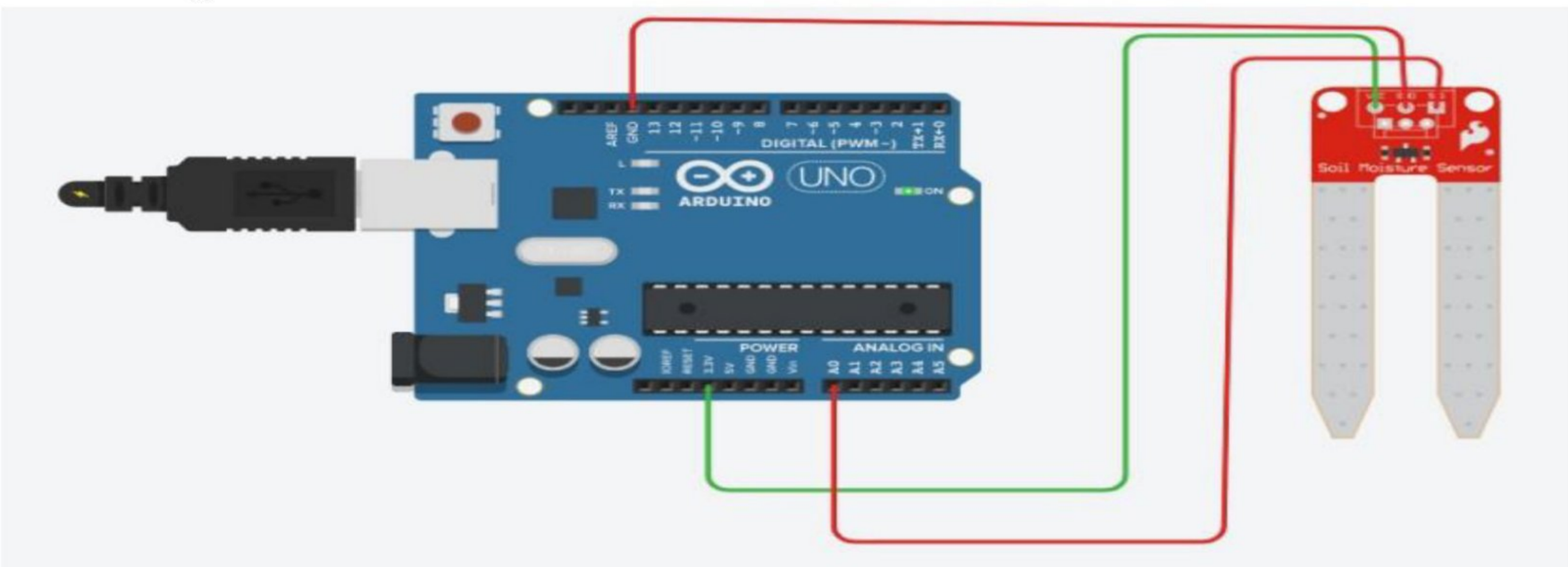
1.Monitor the soil moisture level using Arduino

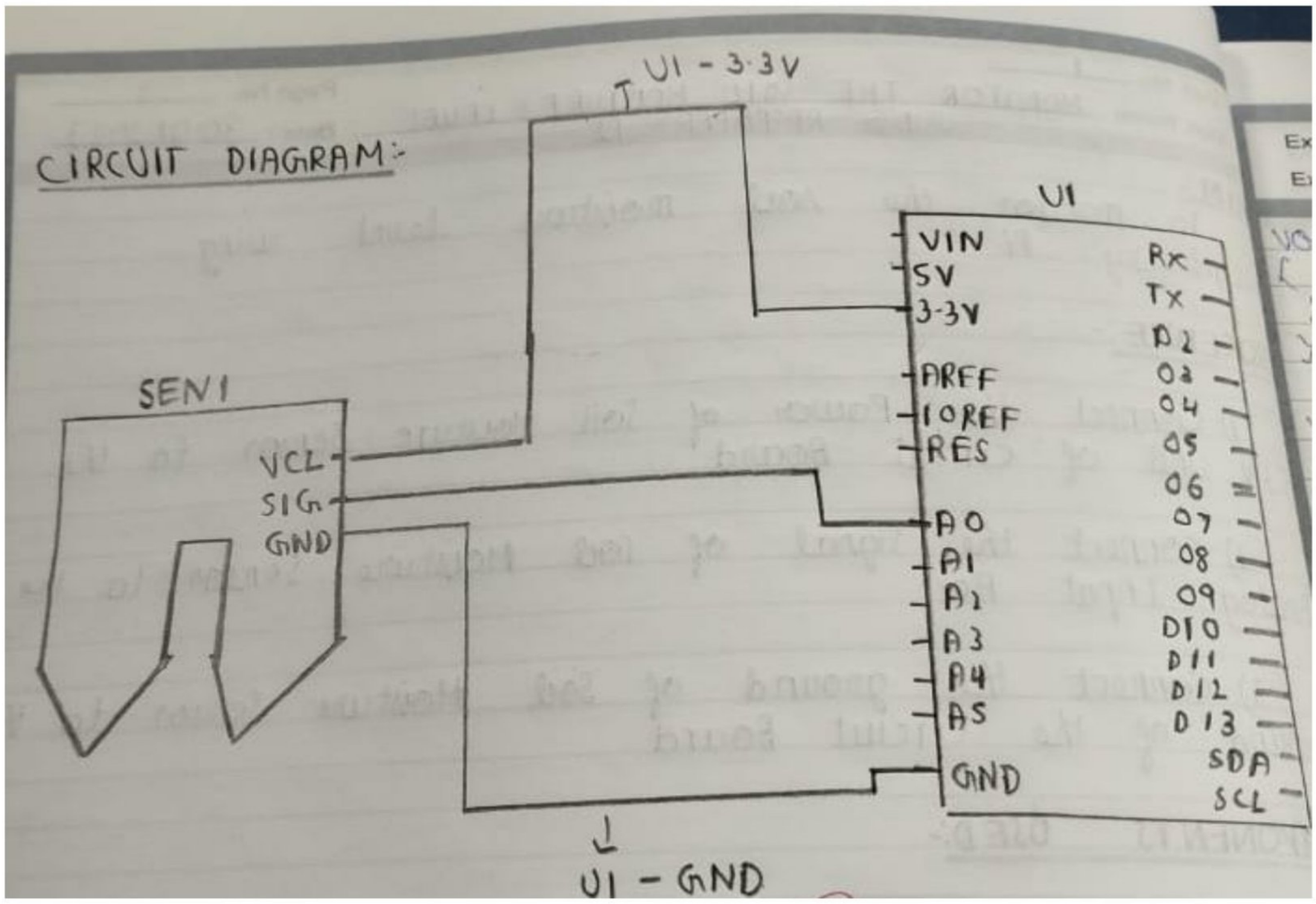
List of components:

Name	Quantity	Component
U1	1	Arduino Uno R3
SEN1	1	Soil Moisture Sensor

Circuit design:



Circuit design for exam:



Scanned by TapScanner

Scanned by TapScanner

```
// C++ code //
int moistureValue;
float moisturePercentage;
void setup()
{
    Serial.begin(9600);
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
}
void loop()
{
    moistureValue = analogRead(A0);
    moisturePercentage = ((moistureValue/539.00) * 100);
    Serial.print("Moisture Value: ");
    Serial.print(moistureValue);
    Serial.print("\nMoisture Percentage: ");
    Serial.print(moisturePercentage);

    if (moisturePercentage < 15.00)
            {
              digitalWrite(13, HIGH);
              delay(1000);
              digitalWrite(13, LOW);
              delay(1000);
            }
    if ((moisturePercentage < 70.00) && (moisturePercentage >= 15.00))
            {
              digitalWrite(12, HIGH);
              delay(1000);
              digitalWrite(12, LOW);
              delay(1000);
            }
if ((moisturePercentage <= 100.00) && (moisturePercentage >= 70.00))
            {
              digitalWrite(11, HIGH);
              delay(1000);
              digitalWrite(11, LOW);
              delay(1000);
             }
Serial.print("\n*\n");
delay(1000);
}
```

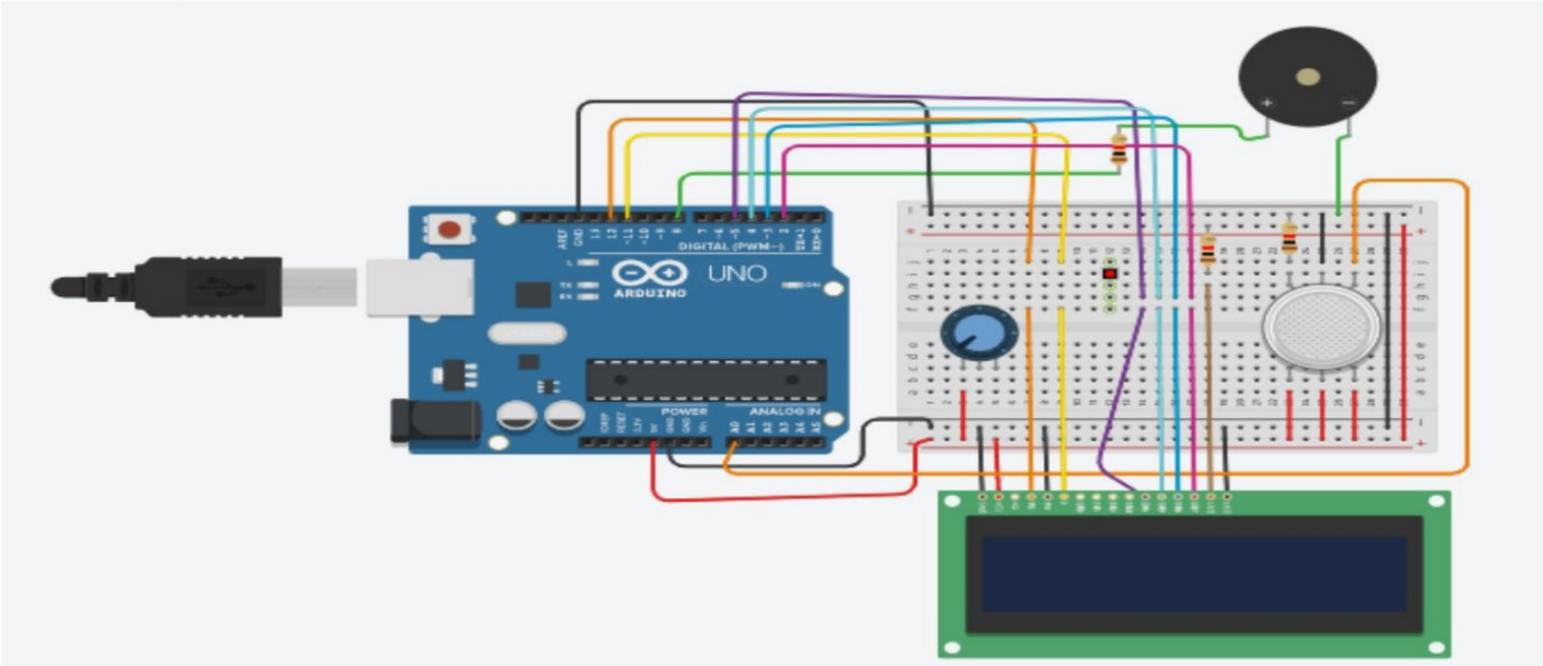
Scanned by TapScanner
Scanned by TapScanner

2.Monitor the surrounding air's quality using Arduino & sensors

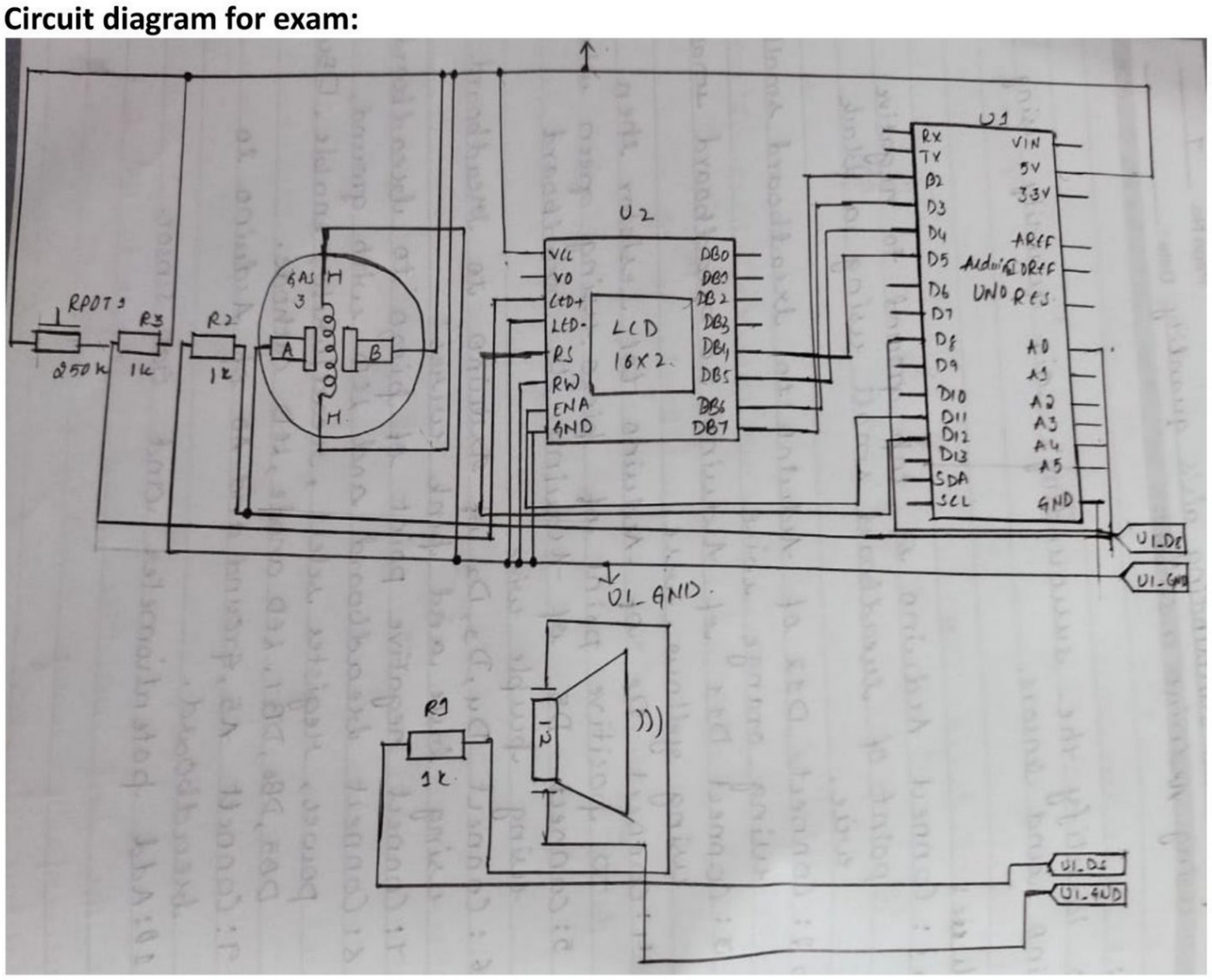
Component required:

Name	Quantity	Component
U1	1	Arduino Uno R3
U2	1	LCD 16 x 2
PIEZO1	1	Piezo
GAS1	1	Gas Sensor
Rpot1	1	250 kΩ Potentiometer
R1 R2 R3	3	1 kΩ Resistor

circuit diagram for connection:



Circuit diagram for exam:



Scanned by TapScanner

Scanned by TapScanner

CODE:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int pin = 8;
int analogPin = A0;
int sensorValue = 0;
void setup()
{
  pinMode(analogPin, INPUT);
  pinMode(pin8, OUTPUT);
  lcd.begin(16, 2);
  lcd.print("what is the air");
  lcd.print("Quality Today");
  Serial.begin(9600);
  lcd.display();
}


void loop()
    {
        delay(100);
        sensorValue = analogRead(analogPin);
        Serial.print("air Quality in RPM=");
        Serial.println(sensorValue);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Air Quality:");
        lcd.print(sensorValue);

         if (sensorValue<=500)
           {
              Serial.print("Fresh air");
              Serial.print("\r\n");
              lcd.setCursor(0,1);
              lcd.print("Fresh Air");
           }
        else if(sensorValue>=500 && sensorValue<=600)
           {
              Serial.print("Poor Air");
              Serial.print("\r\n");
              lcd.setCursor(0,1);
              lcd.print("Poor Air");
            }
        else if (sensorValue>=650)
            {
              Serial.print("Very Poor Air");
              Serial.print("\r\n");
              lcd.setCursor(0,1);
```

Scanned by TapScanner
Scanned by TapScanner

```
            lcd.print("Very Poor Air");
        }
  }
if(sensorValue>650)
    {
      digitalWrite(pin8, HIGH);
    }
else
   {
     digitalWrite(pin8, LOW);
   }
}
```

Scanned by TapScanner
Scanned by TapScanner

Expt. No. 3

Page No. 23

Expt. Name Implementing Raspberrypi as local server by installing Apache, PHP and mySQL

Date :

Aim :

To implement Raspberry pi as local server by installing Apache, PHP and mySQL

Process :

Step 1 - Install Raspberrypi in the system.
Get Raspberrypi and an SD card with atleast 8GB capacity and class 10 speed.

Step 2 - Download the operating system you want to use from the official Raspberrypi website

Step 3 - Use Raspberry pi imager to flash the operating system image to the SD card.

Step 4 - Insert the SD card into Raspberrypi's SD card slot.

Step 5 - Connect keyboard, mouse and monitor to the Raspberry pi

Step 6 - Connect the power supply to the Raspberry pi and power it up and save IP address of pi's

Step 7 - Follow the instructions shown on the screen to setup the operating system.

After installation update and Restart your Raspberry pi
Sudo apt-get update (To get new versions of packages)
Sudo apt-get upgrade

Scanned by TapScanner

Scanned by TapScanner

Expt. No. _____ Page No. 25

Expt. Name _____ Date : _____

Installation of Apache :

Sudo apt-get install apache2

Now check Apache is working or not. For that open web browser and enter your raspberrypi's IP address in the address bar.
You will see the Apache default page.

To get the IP address of Raspberrypi, use the following command :

Sudo raspi_config

Installation of PHP :

Sudo apt-get install php

Check PHP working or not, For that create a PHP file in the Apache root Directory

```
Sudo nano / var / www / html / info.php
<? php
php info();
? >
```

Save and exit the file
Now restart apache service by using following command:

Sudo /etc/ init.d/ apache2 restart

Scanned by TapScanner
Scanned by TapScanner

Installation of mysQL :

→ To install mysQL use the following command
Sudo apt-get install mysql-server
→ To secure mysQL install mysQL-secure
sudo mysql-secure-installation

Now create a mysQL user and Database

```
mysql -u root -p
CREATE DATABASE mydatabase ;
CREATE USER 'myuser"@"localhost' IDENTIFIED BY
                    "mypassword" ;
GRANT ALL PRIVILEGES ON mydatabase.* To
          'myuser"@"localhost' ;
FLUSH PRIVILEGES ;
```

Installation of phpmyAdmin :

We install phpmyAdmin to manage our mysQL database
Use the following command :

Sudo apt-get install phpmyadmin-y

Scanned by TapScanner
Scanned by TapScanner

Expt. No. _____ Page No. 29

Expt. Name _____ Date : _____

NOTE :

During the installation, you will be prompted to configure phpmyAdmin. Then choose Apache. As the web server and enter your mysQL root password when prompted.

After installation you can access phpmyAdmin by visiting 'https://localhost/phpmyAdmin'

Results :

Hence Raspberry Pi is now set up as a local server running Apache, pHp and mysQL. The implementation is successful and is ready for blogging.

Scanned by TapScanner

Scanned by TapScanner

Output -

Default Gateway 192.108.1.254

Output -

[OK] restarting apache2

Scanned by TapScanner

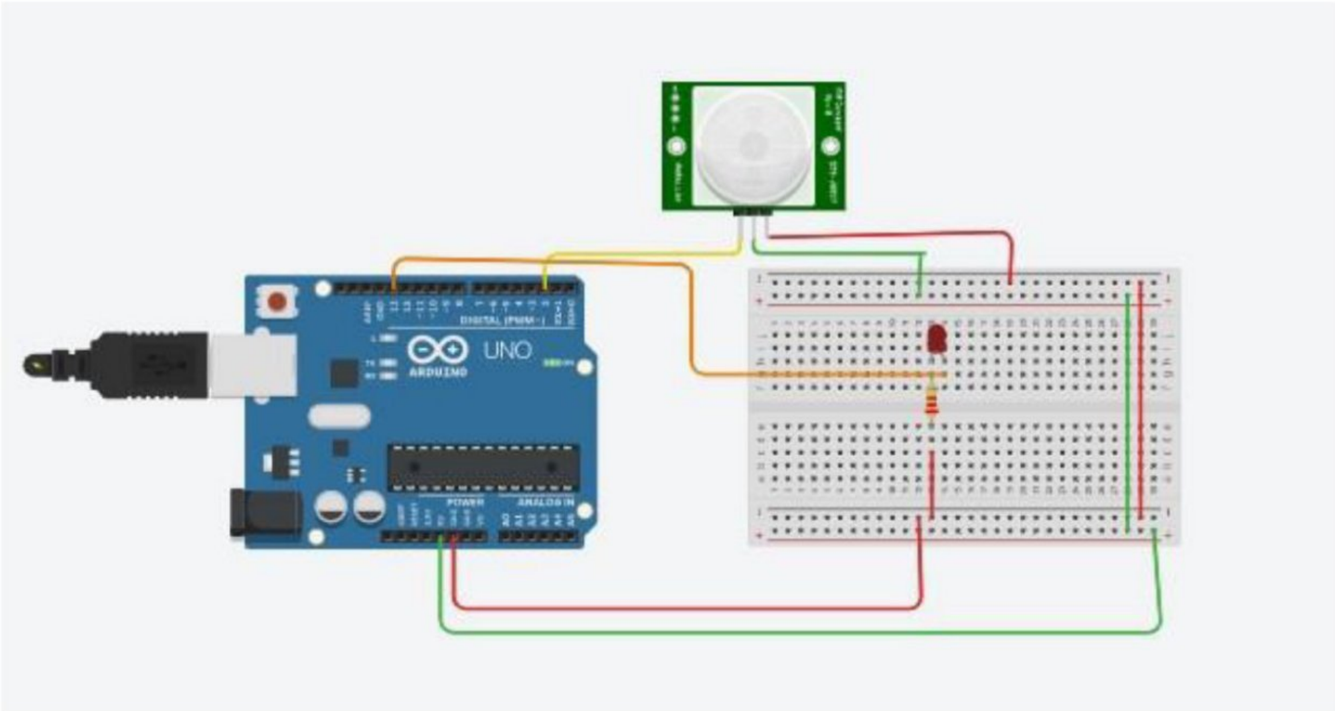
Scanned by TapScanner

4. Interface PIR sensor with ESP8266 WIFI module on Arduino platform.

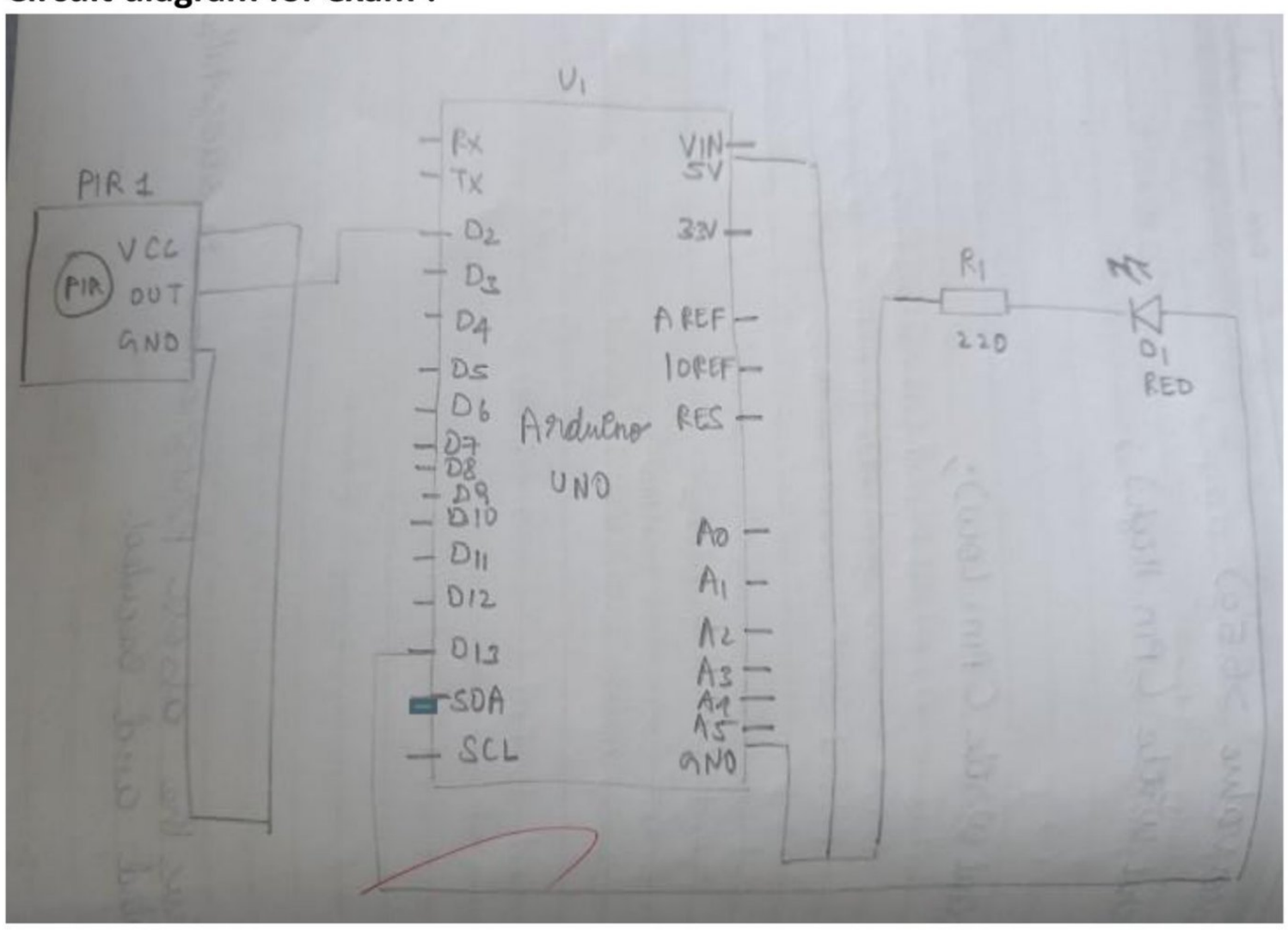
COMPONENTS USED:

Name	Quantity	Component
U1	1	Arduino Uno R3
PIR1	1	-111.27214974329218 , -169.60663633514048 , -169.60663633514048 PIR Sensor
D1	1	Red LED
R1	1	220 Ω Resistor

CIRCUIT DIAGRAM for connection:



Circuit diagram for exam :



Scanned by TapScanner

Scanned by TapScanner

CODE:

```
int pirsensor = 0;
void setup()
    {
      pinMode(13, OUTPUT);
      pinMode(2, INPUT);
    }
void loop()
    {
      pirsensor = digitalRead(2);
    if(pirsensor==HIGH)
          {
            digitalWrite(13, HIGH);
          }
   else
          {
             digitalWrite(13, LOW);
          }
          delay(10);
}
```

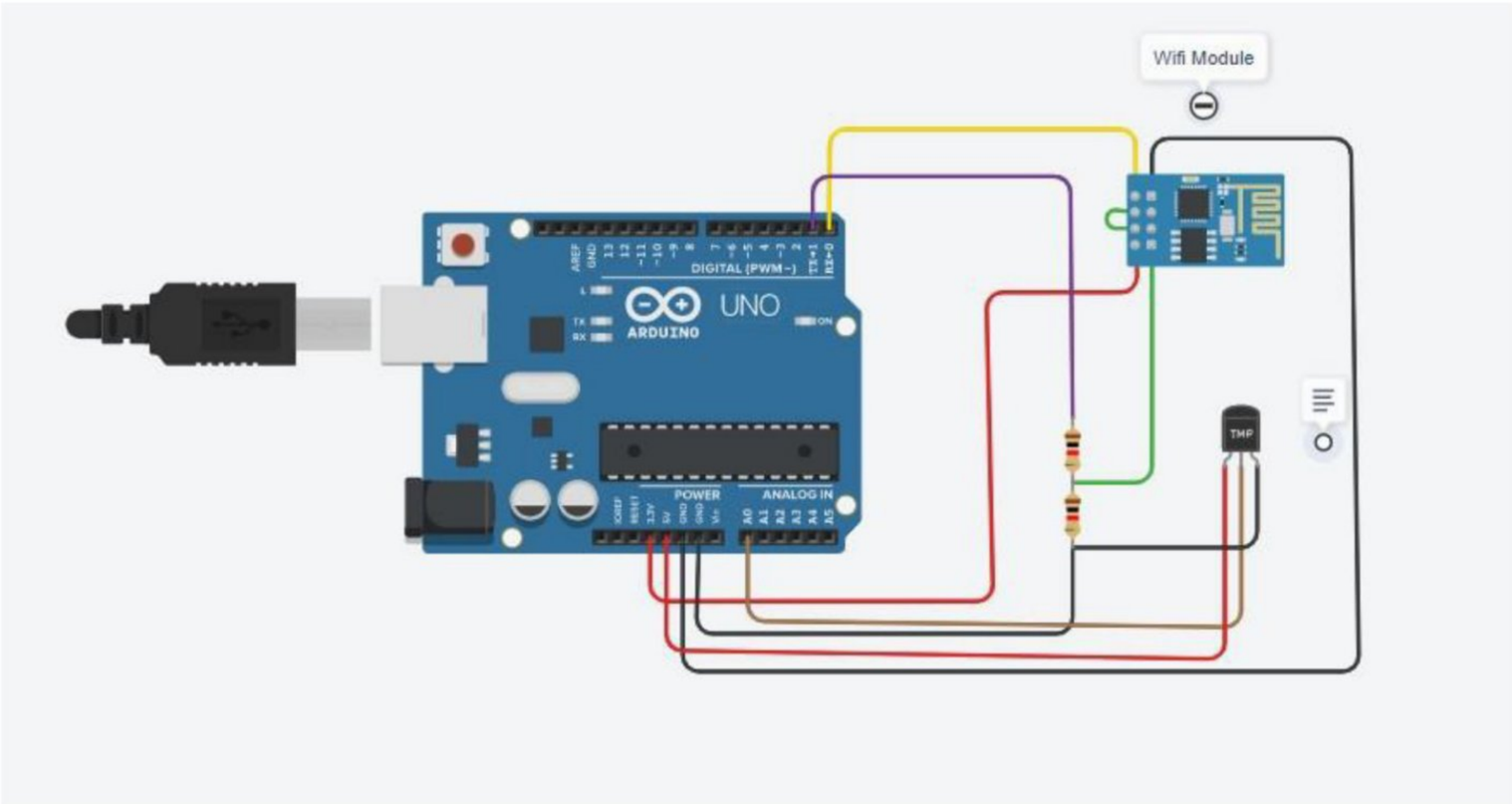
Scanned by TapScanner
Scanned by TapScanner

5. Posting temperature and humidity of the environment to Thing speak cloudservice using NodeMCU.

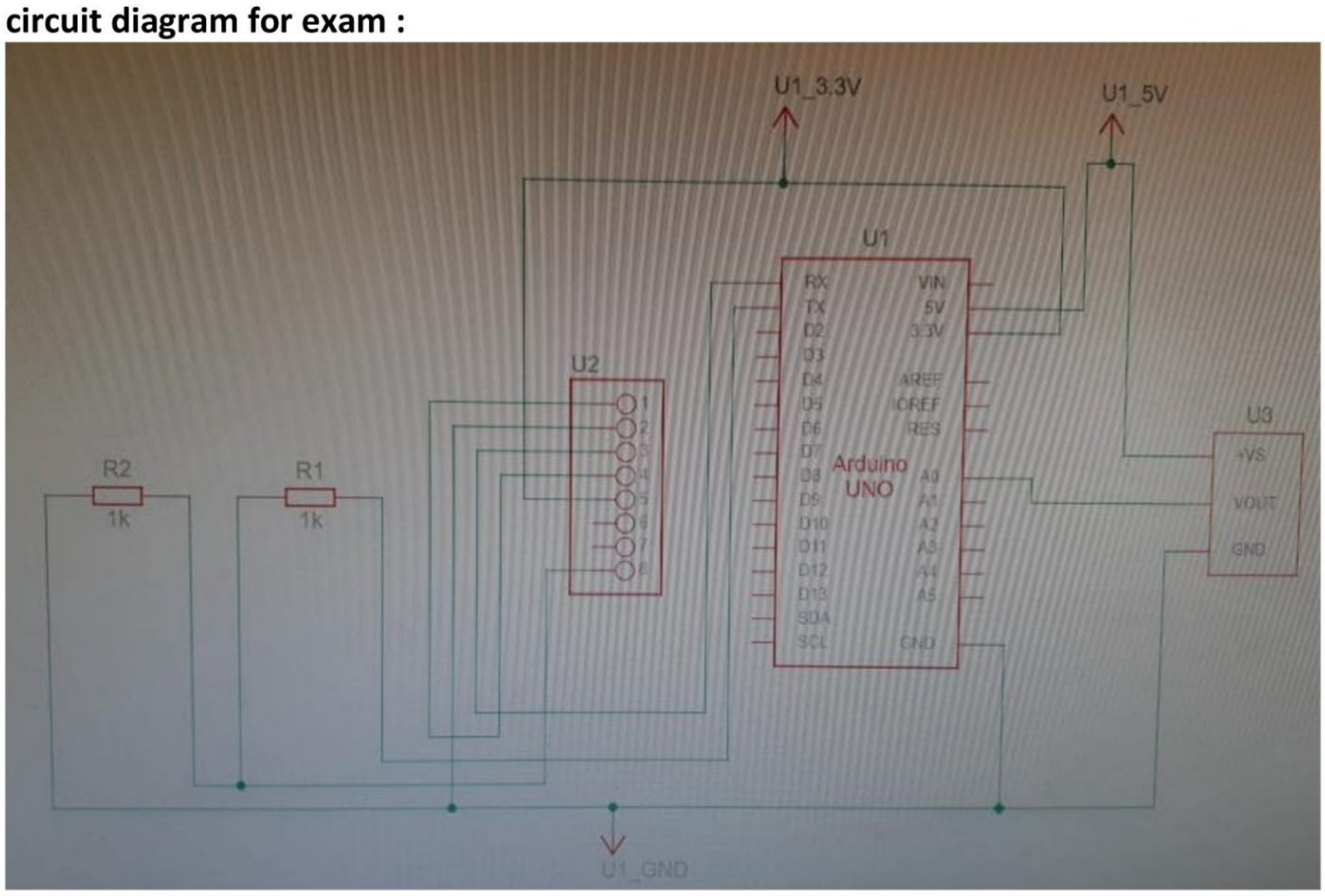
COMPONENTS USED:

Name	Quantity	Component
U1	1	Arduino Uno R3
U2	1	Temperature Sensor [TMP36]
U3	1	Wifi Module (ESP8266)
R1 R2	2	1 kΩ Resistor

circuit diagram for connection:



circuit diagram for exam :



Scanned by TapScanner

Scanned by TapScanner

CODE:

```
float val, voltage, temp;
String ssid      = "Simulator Wifi";String password = "";
String host      = "api.thingspeak.com";
const int httpPort   = 80;
String url       = "/update?api_key=4J2ET3G2MFQRYELM&field1=";void setupESP8266(void)
{
  Serial.begin(115200);Serial.println("AT"); delay(10);
  if (Serial.find("OK")) Serial.println("ESP8266 OK!!!");
  Serial.println("AT+CWJAP=\"" + ssid + "\",\"" + password + "\"");delay(10);
  if (Serial.find("OK")) Serial.println("Connected to WiFi!!!");
  Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);delay(50);
  if (Serial.find("OK"))
 Serial.println("ESP8266 Connected to server!!!");
}
void anydata(void)
   {
        val=analogRead(A0); voltage=val*0.0048828125; temp = (voltage - 0.5) * 100.0;
        String httpPacket = "GET " + url + String(temp) + " HTTP/1.1\r\nHost: " + host +
"\r\n\r\n";int length = httpPacket.length();
         Serial.print("AT+CIPSEND=");Serial.println(length); delay(10); Serial.print(httpPacket);
         delay(10);
        if (Serial.find("SEND OK\r\n")) Serial.println("ESP8266 sends data to the server");
    }
 void setup()
      {
          pinMode(A0, INPUT);setupESP8266();
      }
void loop()
      {
         anydata(); delay(4000);
       }
```

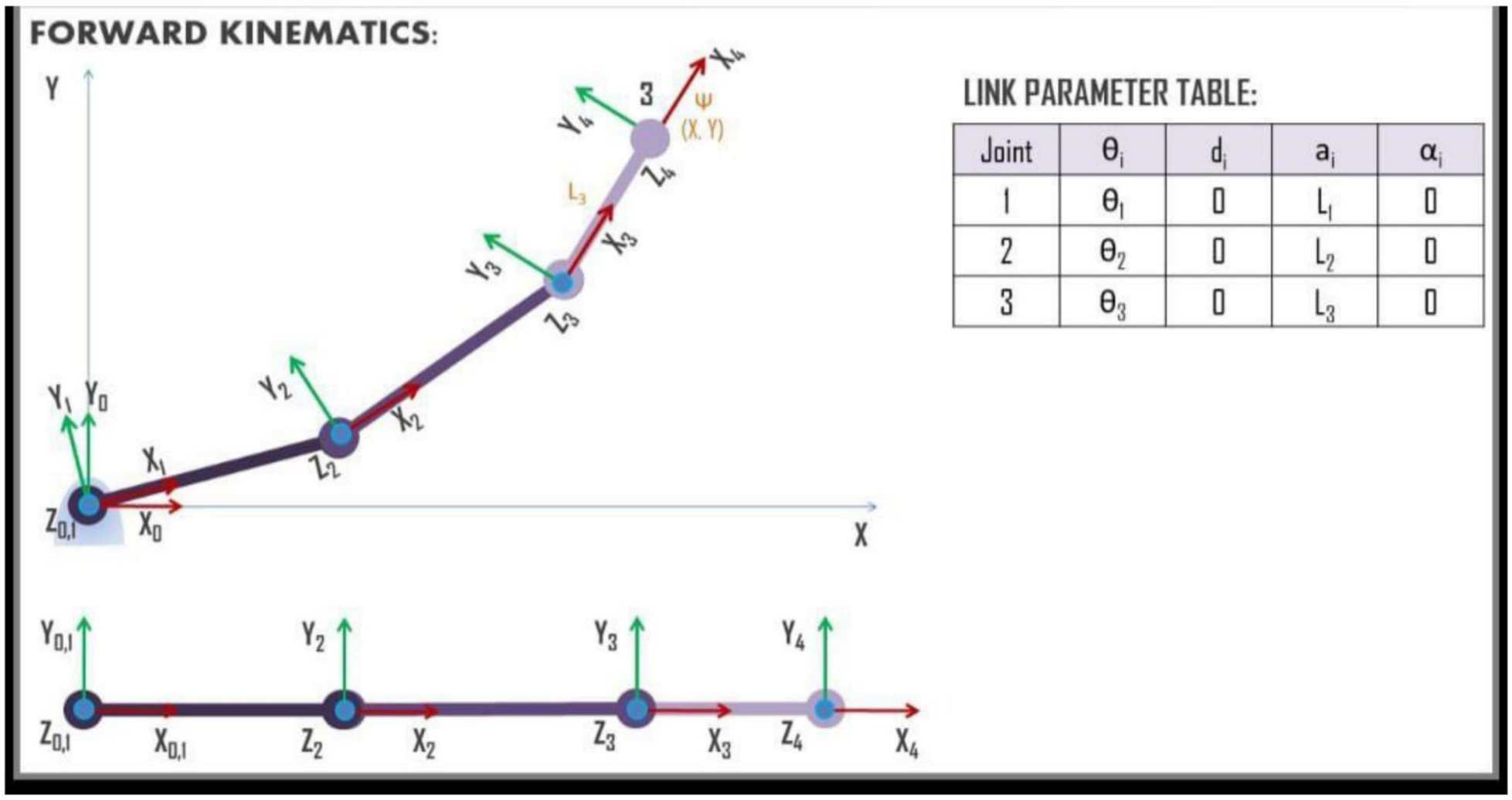
Scanned by TapScanner
Scanned by TapScanner

6. Study components of a real robot and its DH parameters.

Aim:

Study components of a real robot and its DH parameters.

Parameters:
Link length,Link twist, Joint angle, Link offset
Theory:
DH parameters, or Denavit-Hartenberg parameters, are a common method used to describe the kinematics of robotic manipulators. The DH parameter method is a mathematical technique for modeling the relative position and orientation of each joint in a robot arm. There are four DH parameters that are used to describe the kinematics of each joint in a robot arm. These parameters are:
Link length: This is the distance between two consecutive joint axes along the common normal. It is denoted by the symbol "a".
Link twist: This is the angle between two consecutive joint axes about their common normal. It is denoted by the symbol "α".
Joint angle : This is the angle between two consecutive links about their joint axis. It is denoted by the symbol "θ".
Link offset: This is the distance between two consecutive joint axes along the common tangent. It is denoted by the symbol "d".
By defining these four parameters for each joint, it is possible to model the position and orientation of the robot's end effector. The DH parameters are typically used to define the transformation matrix between two consecutive links in a robot arm, which enables the computation of the position and orientation of the end effector based on the joint angles.
In addition to the DH parameters, other factors such as joint limits, joint velocities, and joint torques also need to be considered when designing a robotic manipulator. These factors are important for ensuring that the robot is able to perform its intended tasks safely and efficiently.



Scanned by TapScanner
Scanned by TapScanner

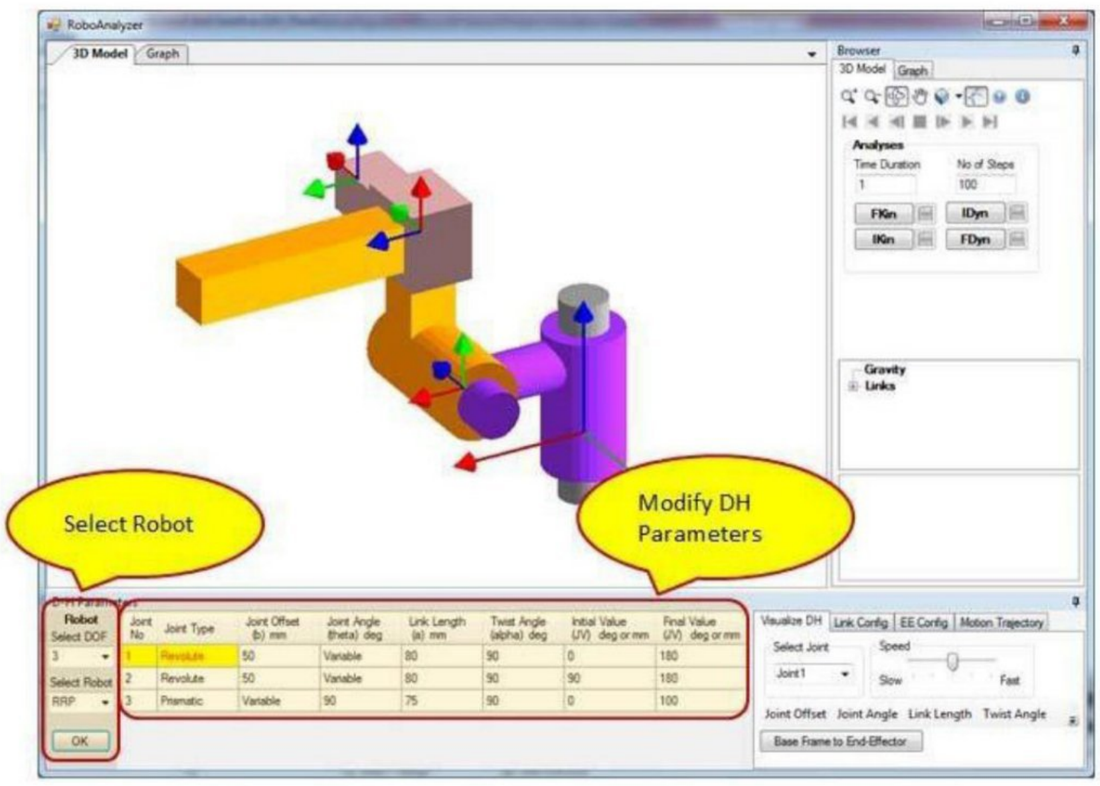
Experiment:7
Forward Kinematics and validate using Robo Analyzer

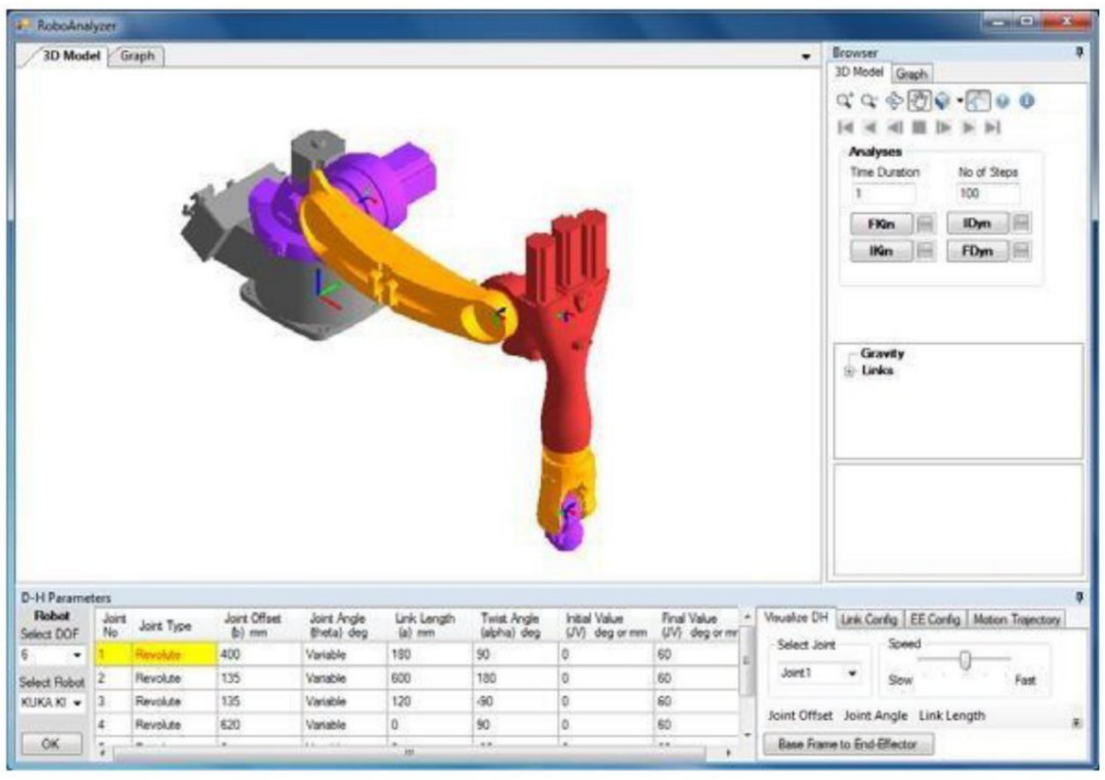
MINIMUM SYSTEM REQUIREMENT

Processor: Atleast 1.5 GHz
RAM: Atleast 512 MB
Operating System: Windows XP, Windows Vista, Windows 7
Dependencies: Microsoft .Net 2.0 framework

Procedure

Step 1: Visit http://www.roboanalyzer.com
Step 2: Click on **Downloads** tab
Step 3: Click on **RoboAnalyzer V5** (or latest version) to download a .zip file
Step 4: A popup window will appear. Select the folder where the file has to be saved and click on **Save**
Step 5: After downloading is complete, unzip RoboAnalyzer5.zip to any folder on your computer. Open the folder RoboAnalyzer5
Step 6: Double-click on RoboAnalyzer5.exe to start RoboAnalyzer



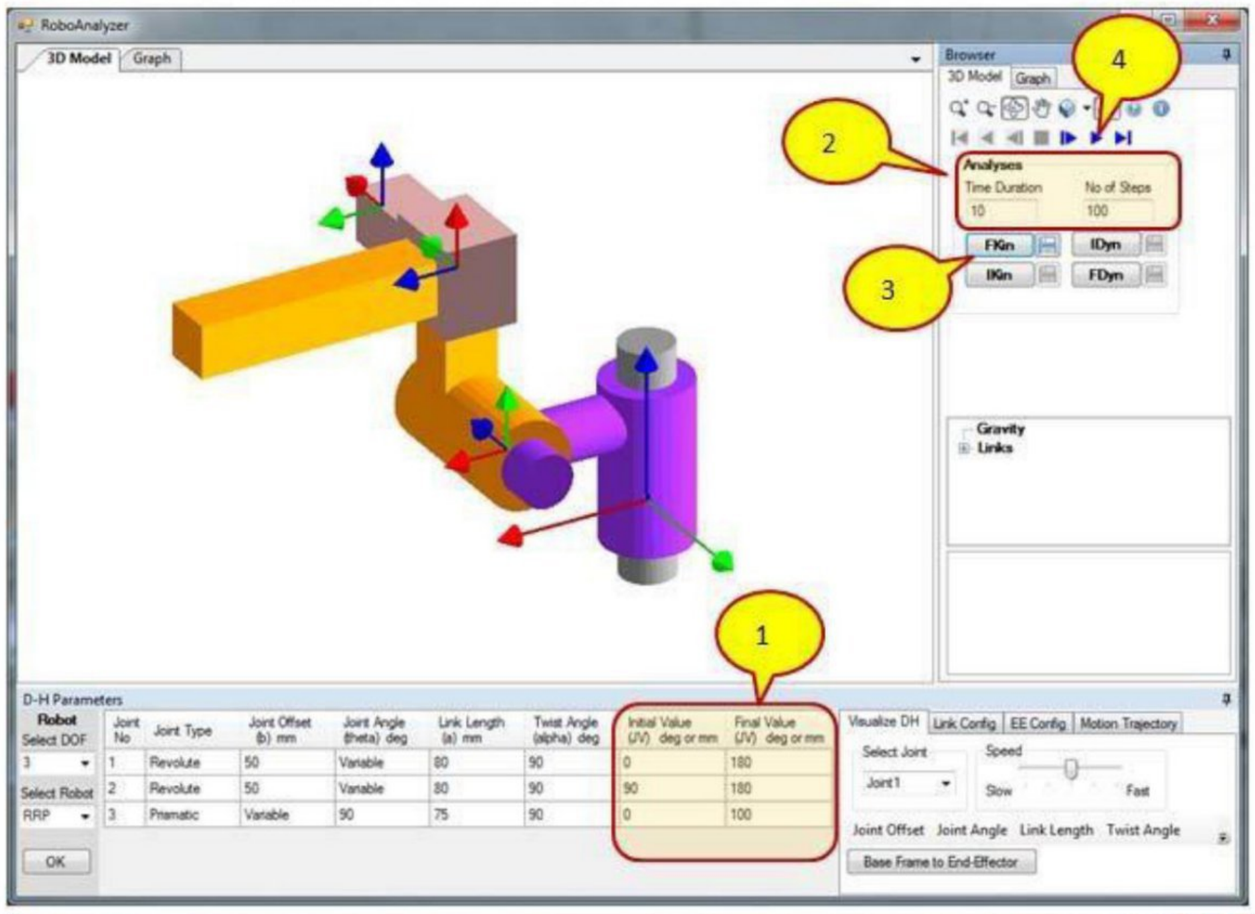


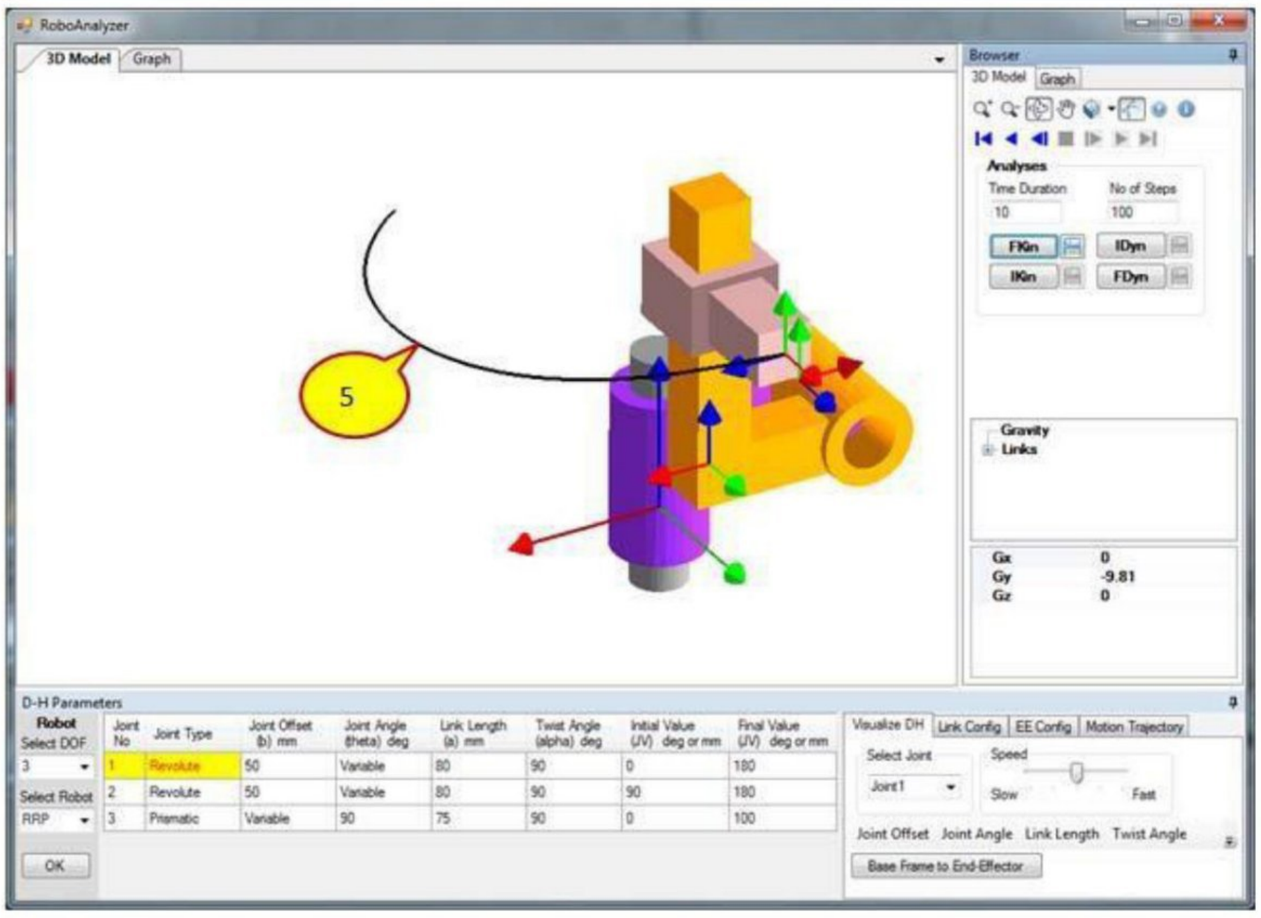
Scanned by TapScanner
Scanned by TapScanner

ANIMATION OF FKIN

1. Set the **initial** and **final** values of joint variables
2. Set **Time Duration** and **Number of Steps**
3. Click on **FKin** button
4. Click on **Play** button to see the animation

The end-effector trace can be viewed





GRAPH PLOTS OF FKIN

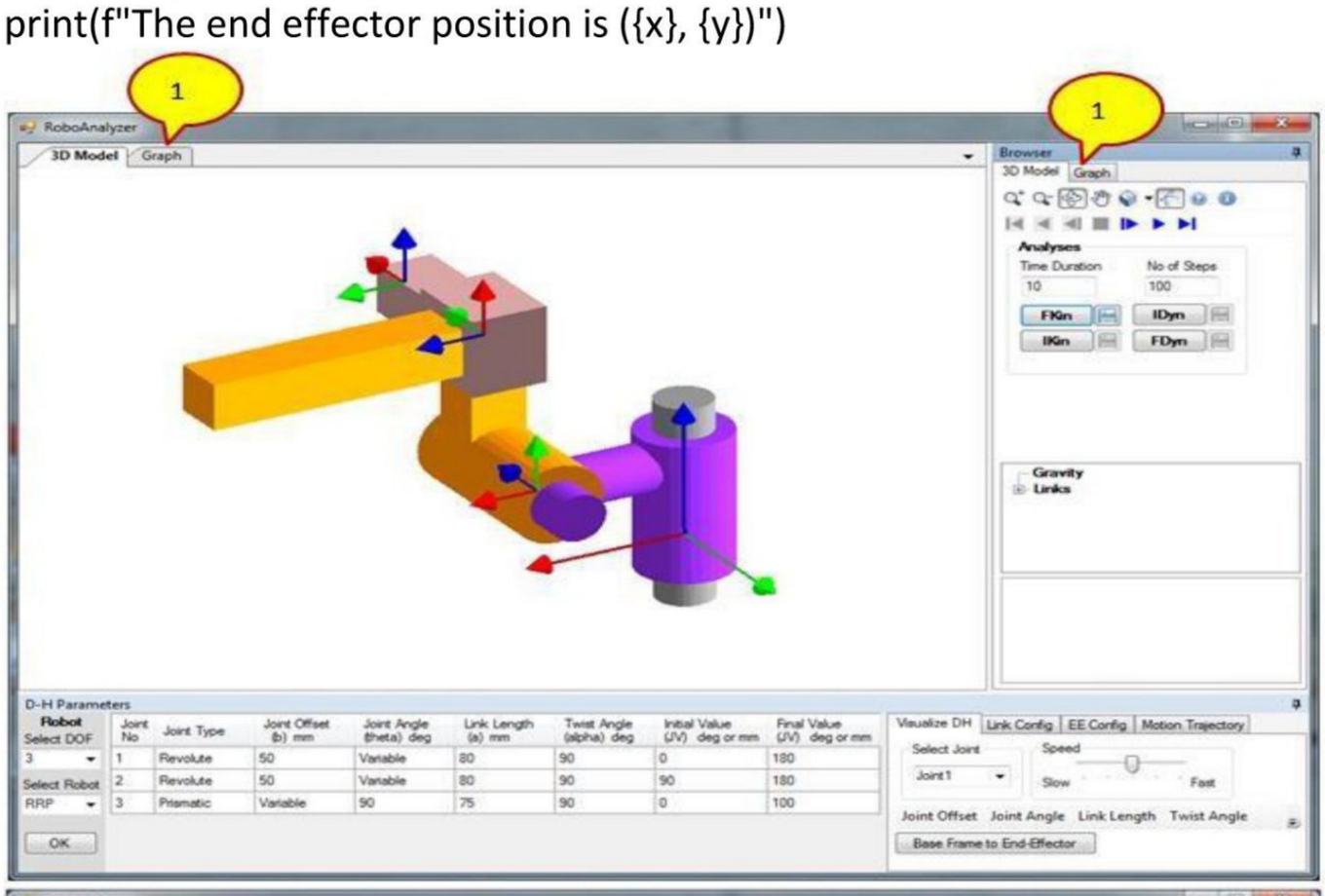
To view the graph plots of a forward kinematics (animation) analysis, the following are the steps are

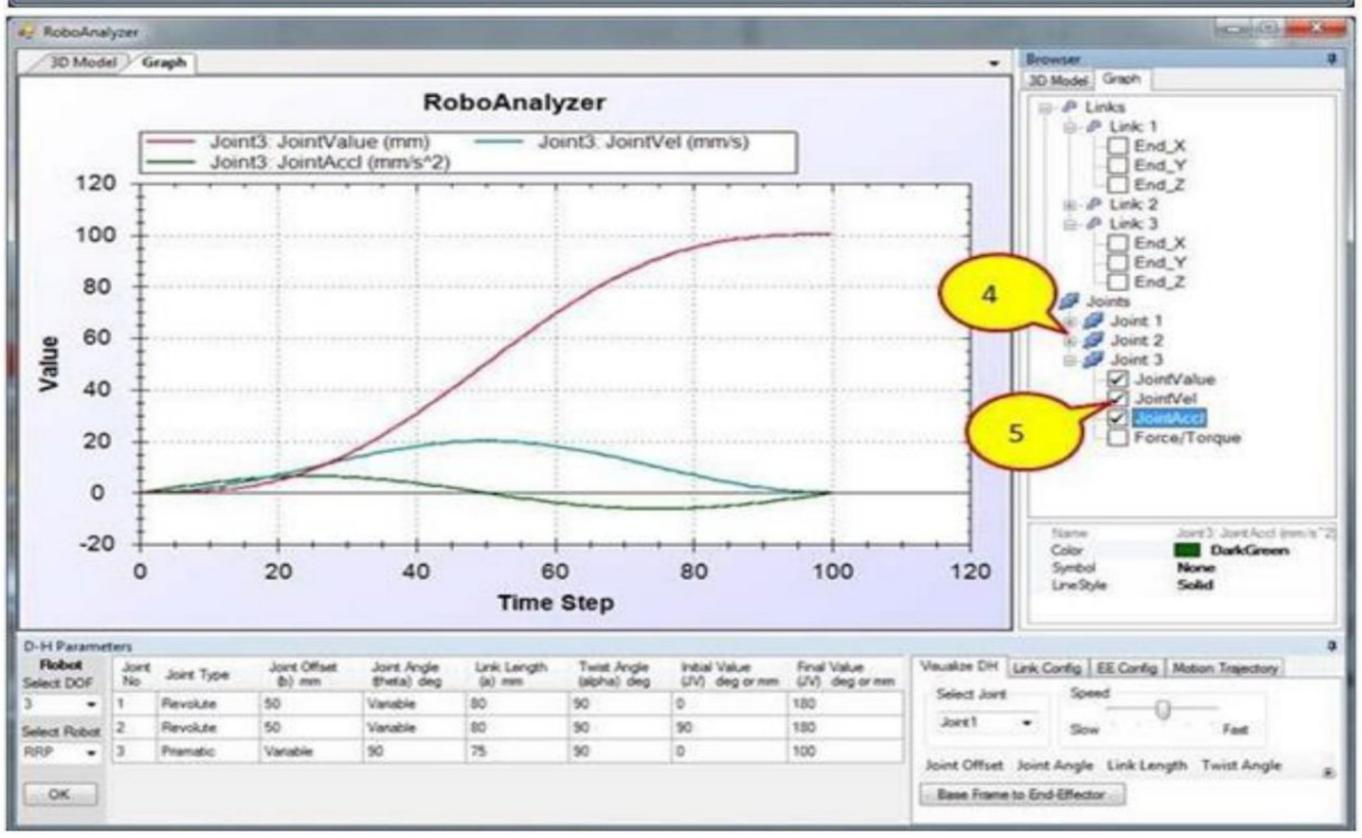
1. Click on **Graph** tab
2. Click on **+** next to the link of which the plots are to be viewed
3. Click on **box** to plot graph of a particular node to see X, Y and Z plots
4. Click on **+** next to the joint of which the plots are to be viewed
5. Click on **box** to plot graph of a particular node to see joint value (joint angle for revolute joint and joint offset for prismatic joints), joint velocity and joint acceleration

Scanned by TapScanner
Scanned by TapScanner

Code:

```
import numpy as np
# Define the joint angles in radians
theta1 = np.pi/6
theta2 = np.pi/4
theta3 = np.pi/3
theta4 = np.pi/2
# Define the lengths of the links
L1 = 2
L2 = 1.5
L3 = 1
L4 = 0.5
# Calculate the end effector position
x = L1*np.cos(theta1) + L2*np.cos(theta1+theta2) + L3*np.cos(theta1+theta2+theta3) +
L4*np.cos(theta1+theta2+theta3+theta4)
y = L1*np.sin(theta1) + L2*np.sin(theta1+theta2) + L3*np.sin(theta1+theta2+theta3) +
L4*np.sin(theta1+theta2+theta3+theta4)
print(f"The end effector position is ({x}, {y})")
```





Scanned by TapScanner

Scanned by TapScanner

Experiment:8 Inverse Kinematics of the real robot and validate using Robo Analyzer

Aim:
Inverse Kinematics of the real robot and validate using Robo Analyzer

MINIMUM SYSTEM REQUIREMENT

Processor: Atleast 1.5 GHz
RAM: Atleast 512 MB
Operating System: Windows XP, Windows Vista, Windows 7
Dependencies: Microsoft .Net 2.0 framework

Procedure

Step 1: Visit http://www.roboanalyzer.com
Step 2: Click on **Downloads** tab
Step 3: Click on **RoboAnalyzer V5** (or latest version) to download a .zip file
Step 4: A popup window will appear. Select the folder where the file has to be saved and click on **Save**
Step 5: After downloading is complete, unzip RoboAnalyzer5.zip to any folder on your computer. Open the folder RoboAnalyzer5
Step 6: Double-click on RoboAnalyzer5.exe to start RoboAnalyzer

SOLUTIONS OF IKIN

1.Click on **IKin** button. It shows a separate window (Figure 16)
2.Select a Robot
3.Enter Input parameters
4.Click on **IKin** button
5. View the possible solutions
6. Click on **Show** button. It shows the selected solution in 3D Model window. To see this go back to main window by minimizing IKin window
7. Select any of the obtained solution as initial and final solution
Click on **OK.** This step replaces the initial and final joint values in DH Parameter table

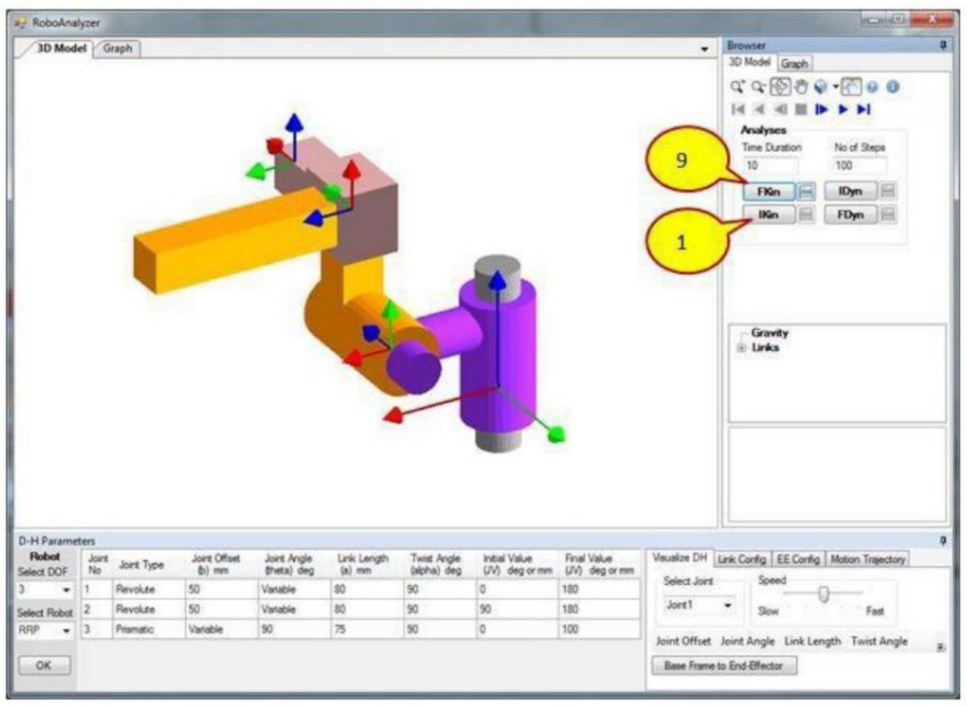
(Main window) by values selected in step 7

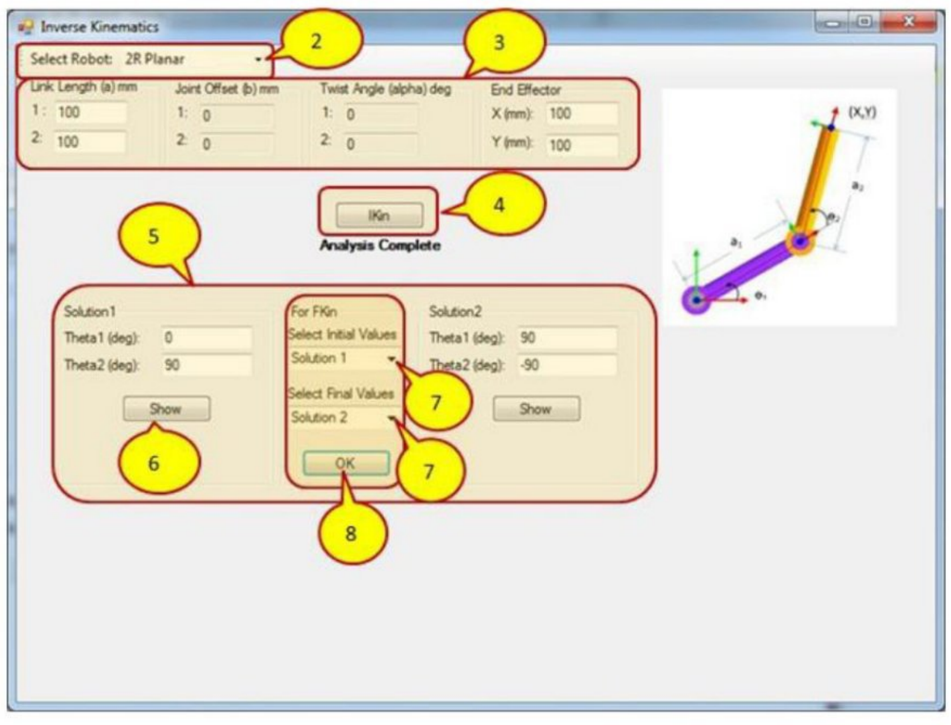
8. Click on **FKin** button to view animation i.e. how robot moves from one solution to another solution selected in step 7

Code:

```
import numpy as np
# Define the end effector position
x = 1.5
y = 0.5
# Define the link lengths
L1 = 2
L2 = 1.5
# Calculate the joint angles using inverse kinematics
theta2 = np.arccos((x*2 + y2 - L12 - L2*2) / (2 * L1 * L2))
theta1 = np.arctan2(y, x) - np.arctan2(L2 * np.sin(theta2), L1 + L2 * np.cos(theta2))
print(f"The joint angles are ({theta1}, {theta2}) in radians")
```

Scanned by TapScanner
Scanned by TapScanner





Scanned by TapScanner
Scanned by TapScanner

Experiment 9 study the use of open source computer vision programming tool open cv

Aim:
study the use of open source computer vision programming tool open cv

image and video processing functions, such as image filtering, edge detection, and object detection. These functions can be used to perform various image and video processing tasks, such as image enhancement, motion detection, and face recognition.

Robotics: OpenCV can be used to develop computer vision algorithms for robots, such as object detection, tracking, and recognition. These algorithms can be used to help robots navigate in a complex environment, identify objects, and interact with the world around them.

Augmented reality: OpenCV can be used to create augmented reality applications that superimpose digital content over real-world images or videos. This can be used for a variety of applications, such as games, advertising, and education.

Machine learning: OpenCV provides support for machine learning algorithms, such as support vector machines (SVMs), neural networks, and decision trees. These algorithms can be used to develop computer vision applications that can learn and improve over time.

Medical imaging: OpenCV can be used to develop medical imaging applications, such as X-ray image analysis and medical diagnosis. It provides developers with a range of image processing functions that can be used to enhance medical images and extract useful information from them.

Overall, OpenCV is a versatile and powerful computer vision programming tool that can be used for a wide range of applications. Its open-source nature makes it accessible to developers all over the world, and its extensive documentation and community support make it easy to learn and use.

Result:
Sucessfully studied the use of open source computervision programming tool open cv

Experiment 10 Image processing using Cv

Scanned by TapScanner
Scanned by TapScanner

Aim:
To implement image processing using open cv

Procedure:

Step-1 install open cv

Step-2 resize an image after downloading the image

Step-3 convert image to grayscale(Black and white)

Step-4 Make an image blur using gaussian blur

Step-5detect edges and dilate the images

Step 6-print all the different types of image

Programme:

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
img = cv2.imread('img name .jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray, (5, 5), 0)
edges = cv2.Canny(blur, 50, 150)
kernel = np.ones((5,5),np.uint8)
dilated = cv2.dilate(edges, kernel, iterations = 1)
cv2_imshow(img)
cv2_imshow(gray)
cv2_imshow(blur)
cv2_imshow(edges)
cv2_imshow(dilated)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Result: The above Programme is Successfully Executed

Scanned by TapScanner
Scanned by TapScanner