PROJECT REPORT

TEAM NDNS

APPROACH:

1. Firstly, we understood the circuit and the concepts regarding it by studying the existing works on Colpitts oscillators and adaptive circuits for insights.
2. We tested the stability of the circuit at different frequencies by varying the inductor and capacitor values and finally chose 100kHz as the output frequency.
3. Used Microcap to simulate the circuit and analyze the theoretical frequency response.
4. Then we built the circuit and checked the output frequency by connecting to an oscilloscope.
5. Collected data at finer temperature intervals using Microcap by adjusting the temperature coefficient to capture subtler trends. Also verified the closeness to the real data.
6. Then based on the datasets we trained a linear regression model in python to predict the capacitor and inductor values when temperature is varied. Plotted the voltage vs frequency and the temperature vs frequency graphs.
7. Then integrated the python code to Arduino and verified the output of the code.
8. Then we verified the model results with practical experiments using temperature and voltage sensor.

PROBLEMS FACED:

1. Because of the faulty components, noise in wires and breadboard we faced problems in building the oscillator to achieve the desired frequency.
2. Understanding the concepts like BJT biasing, blocking the DC current through capacitors and amplifying the feedback was difficult and time taking.
3. Finding the temperature coefficient values of inductors, capacitors, BJT was tiresome.
4. Due to overloading of the DC power supply Arduino got overheated and the connected laptop got shut down.
5. The model's accuracy is not 100% because highly correlated features can make it difficult to determine the individual impact of each predictor, leading to unstable coefficient estimates.
6. Model's predictions are not matching accurately to the real experiments sometimes because of the unavailability of exact values of predicted components at different temperatures.

CONTRIBUTIONS:

1. SAI CHARAN: -
   BUILDING AND SETTING UP THE COLPITTS OSCILLATOR CIRCUIT TO GET THE RESULTING FREQUENCY.
2. DEEPAK: -
   DATA COLLECTION TO TRAIN ML MODEL USING REAL DATA OBTAINED AND THROUGH THE SIMULATION CIRCUIT BUILT IN MICRO CAP. ALSO VERIFIED THE CLOSENESS OF REAL AND MICROCAP DATA.
3. NIKHILESH:
   TRAINING THE ML MODEL WITH THE COLLECTED DATA USING LINEAR REGRESSION APPROACH.
4. NIKHIL: -
   DEALT WITH THE TEMPERATURE SENSOR, VOLTAGE SENSOR, HARDWARE INTEGRATION OF ML MODEL ON ARDUINO.

ABOUT OUR EXPERIMENT:

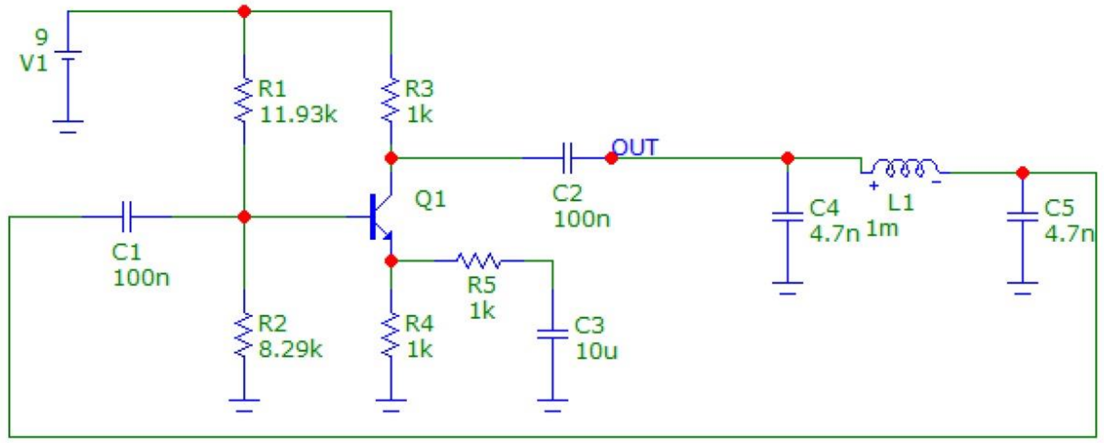
➔ A Colpitts oscillator is an electronic oscillator that generates high-frequency sinusoidal signals

➔ It uses a combination of inductors and capacitors in a resonant LC circuit to determine the oscillation frequency.

➔ The capacitors in the circuit form a capacitive divider, which provides the necessary feedback to sustain oscillations.

➔ It's widely used in RF circuits due to its stability and ease of tuning.

CIRCUIT IMAGE:



COMPONENETS USED:

1. Inductor (L) – For the resonant LC circuit

2. Capacitors (C1 and C2) – Form the capacitive divider for feedback.

3. Transistor (e.g., BJT or FET) – Acts as the active amplifying device.

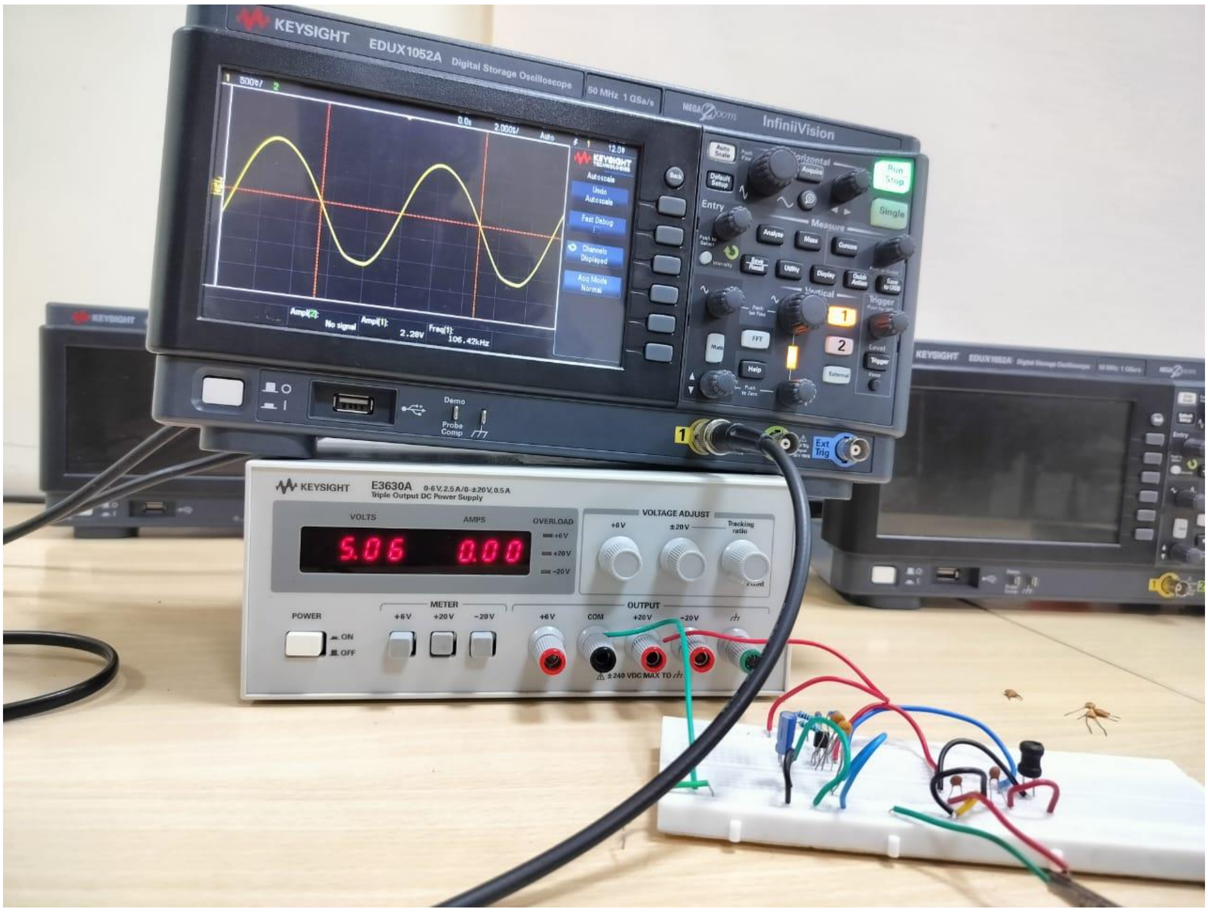
4. Resistors – For biasing the transistor and stabilizing the circuit.

5.Power supply – To provide the necessary operating voltage.

6. Optional bypass capacitor – To stabilize the power supply.

These components are connected to form the LC tank circuit and feedback loop essential for oscillation.

HARWARE IMAGE:



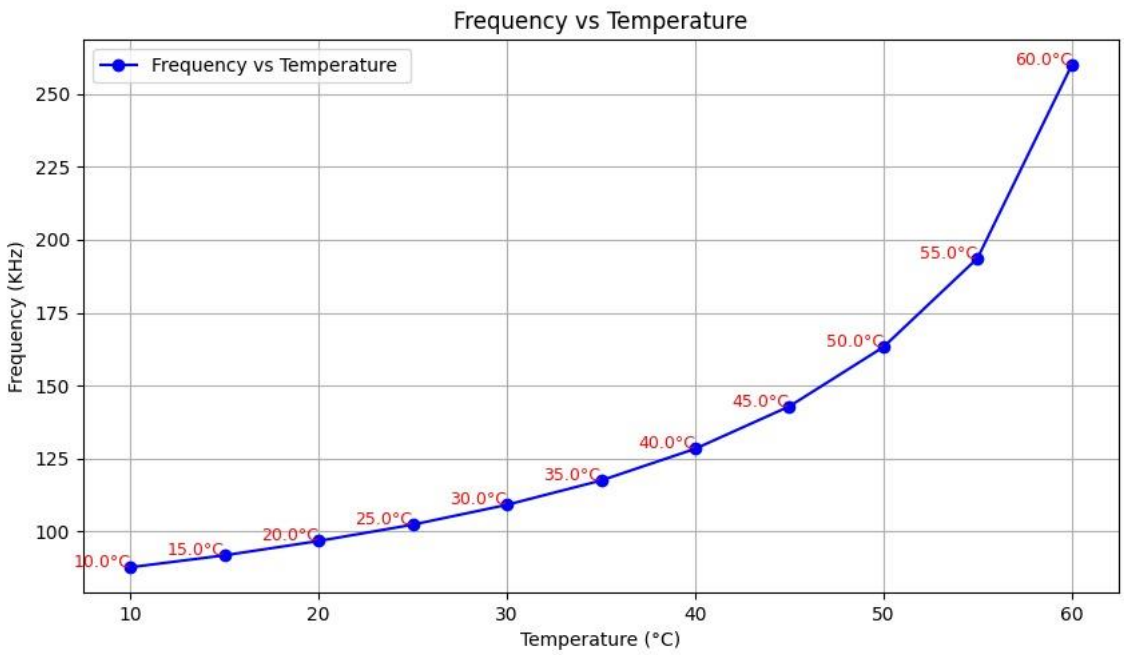
USES AND ITS APPLICATIONS:

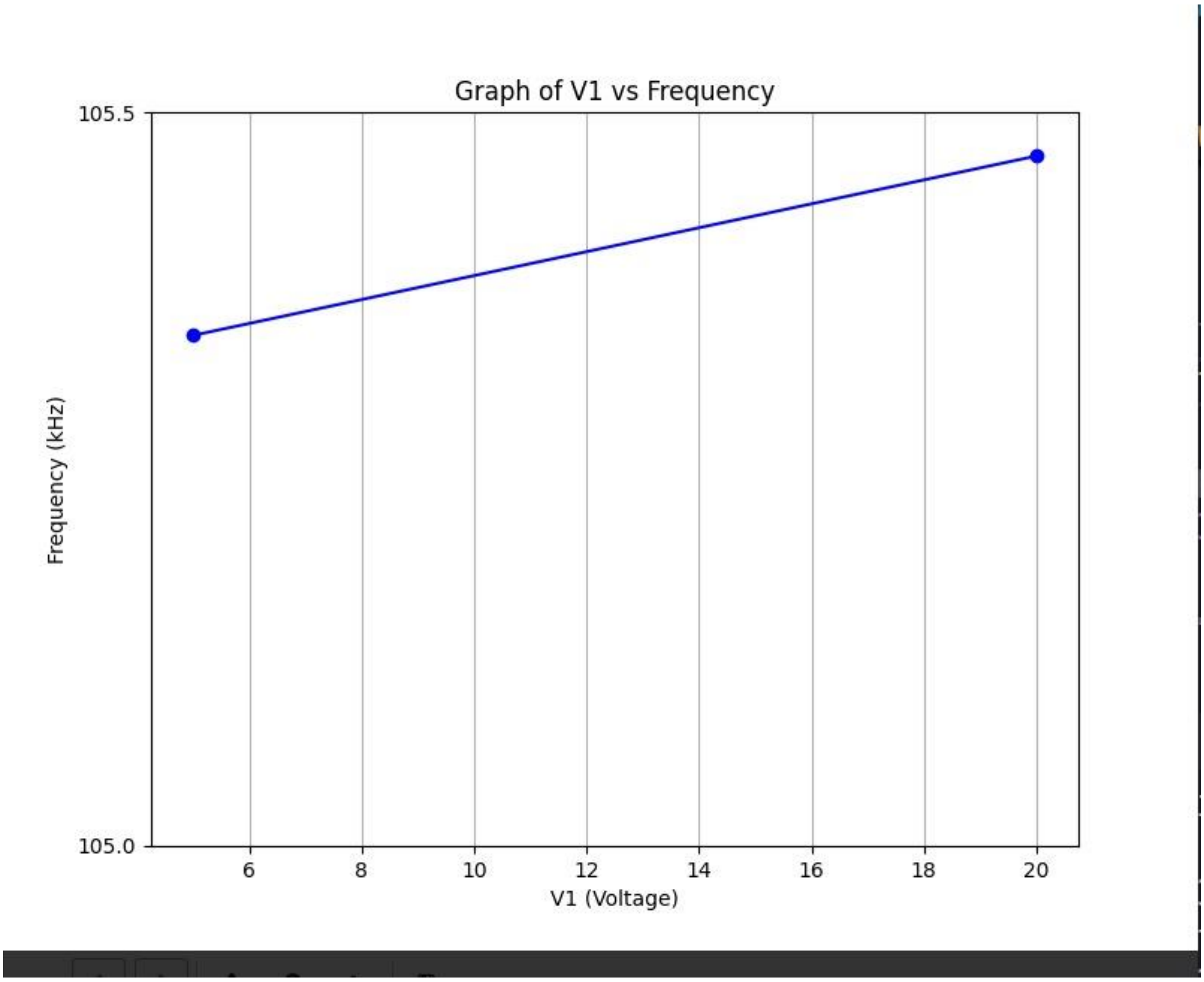
- Signal Generators: Acts as a source of high-frequency sinusoidal signals in testing and instrumentation.
- Frequency Modulation (FM) Circuits: Provides the carrier signal in FM transmitters.

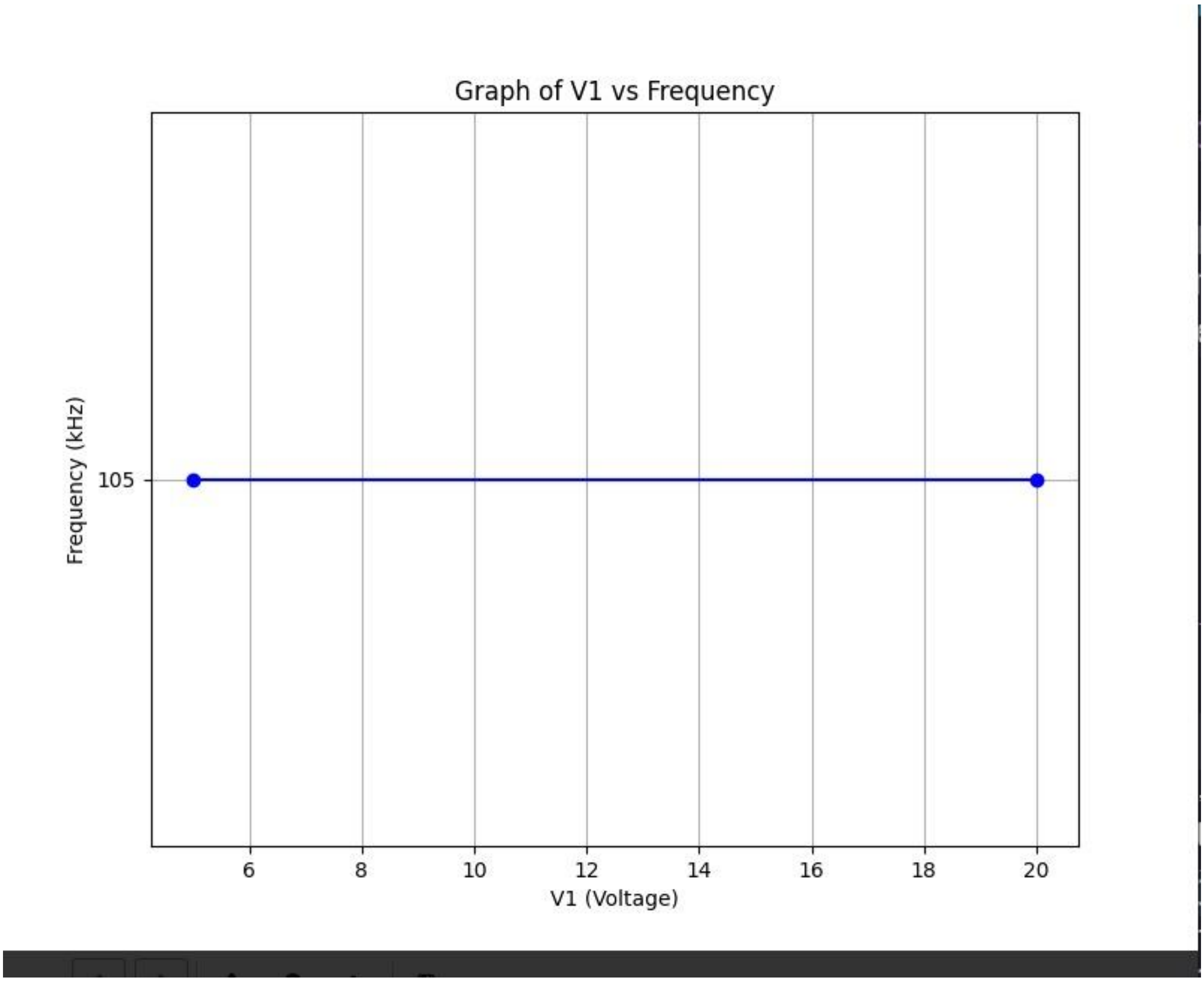
- Clock Generation: Used in digital circuits to produce precise timing signals.
- Wireless Communication: Forms the basis of oscillators in Wi-Fi, Bluetooth, and other wireless technologies.
- Radio Transmitters and Receivers: Generates stable RF signals for communication systems.

RESULTS:

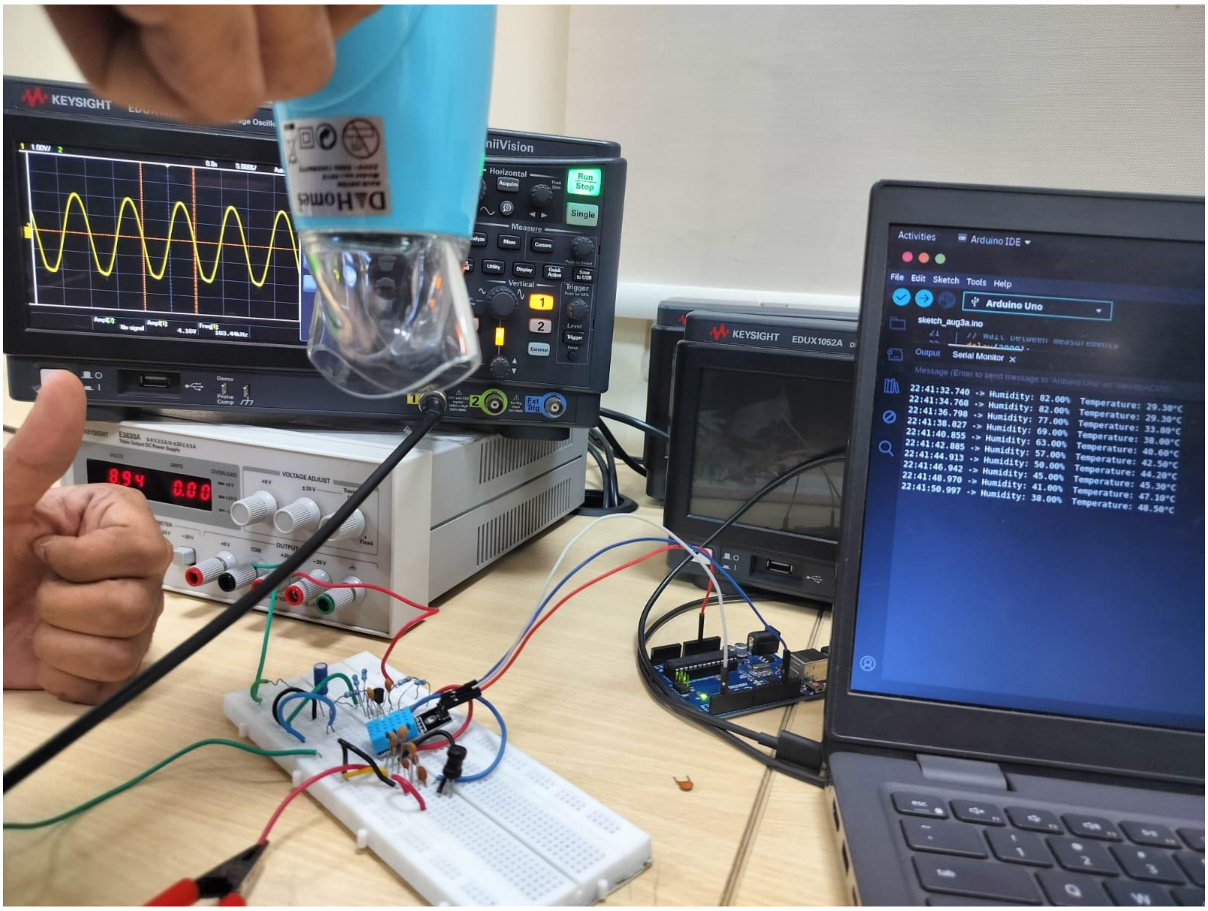
INDUCTANCE	CAPACITOR	TEMPERATUREin C	FREQUENCY in KHZ(error:2kHz)
1mh	4.7nf	27	104
1mh	4.7nf	39	120
1mh	5.7nf	39	105
1mh	4.7nf	49	141
1mh	11.4nf	49	104

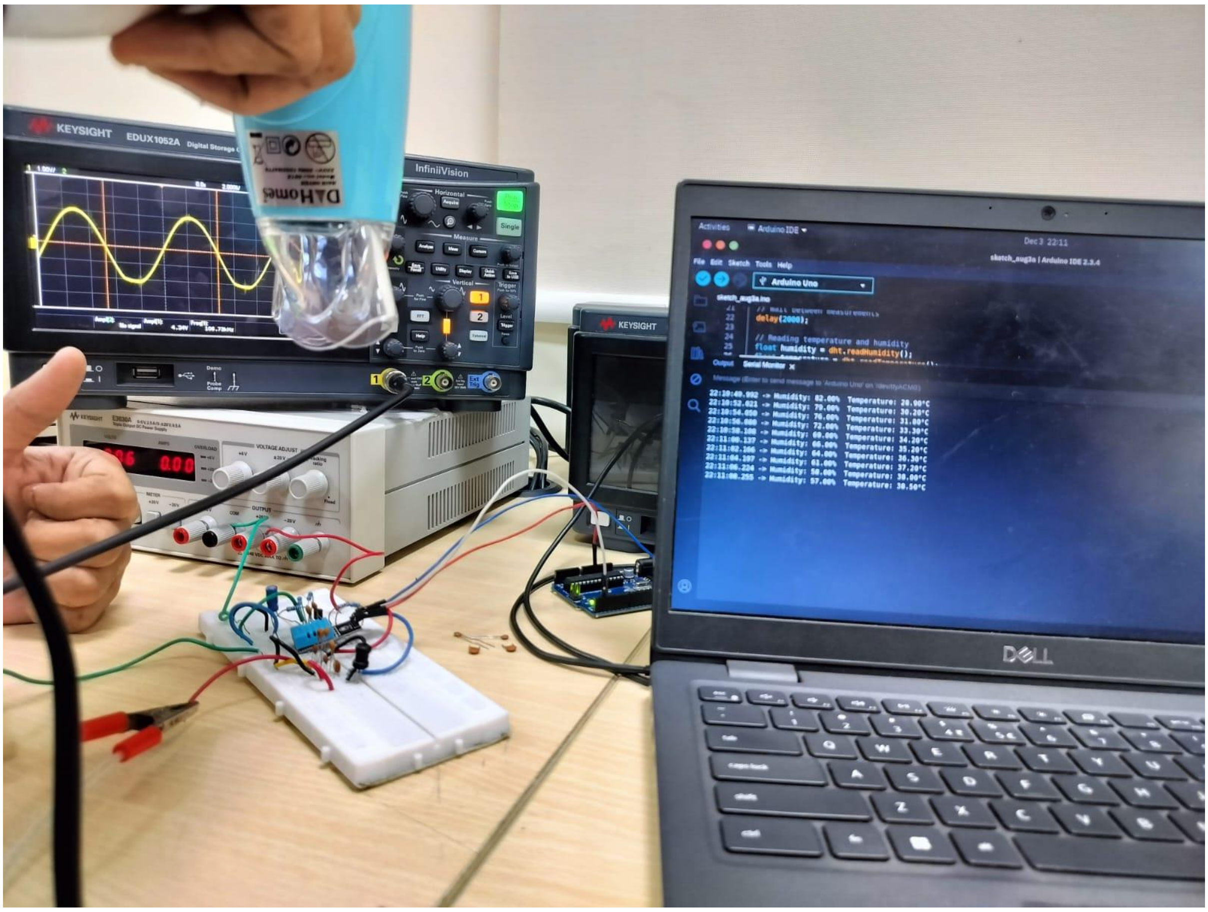


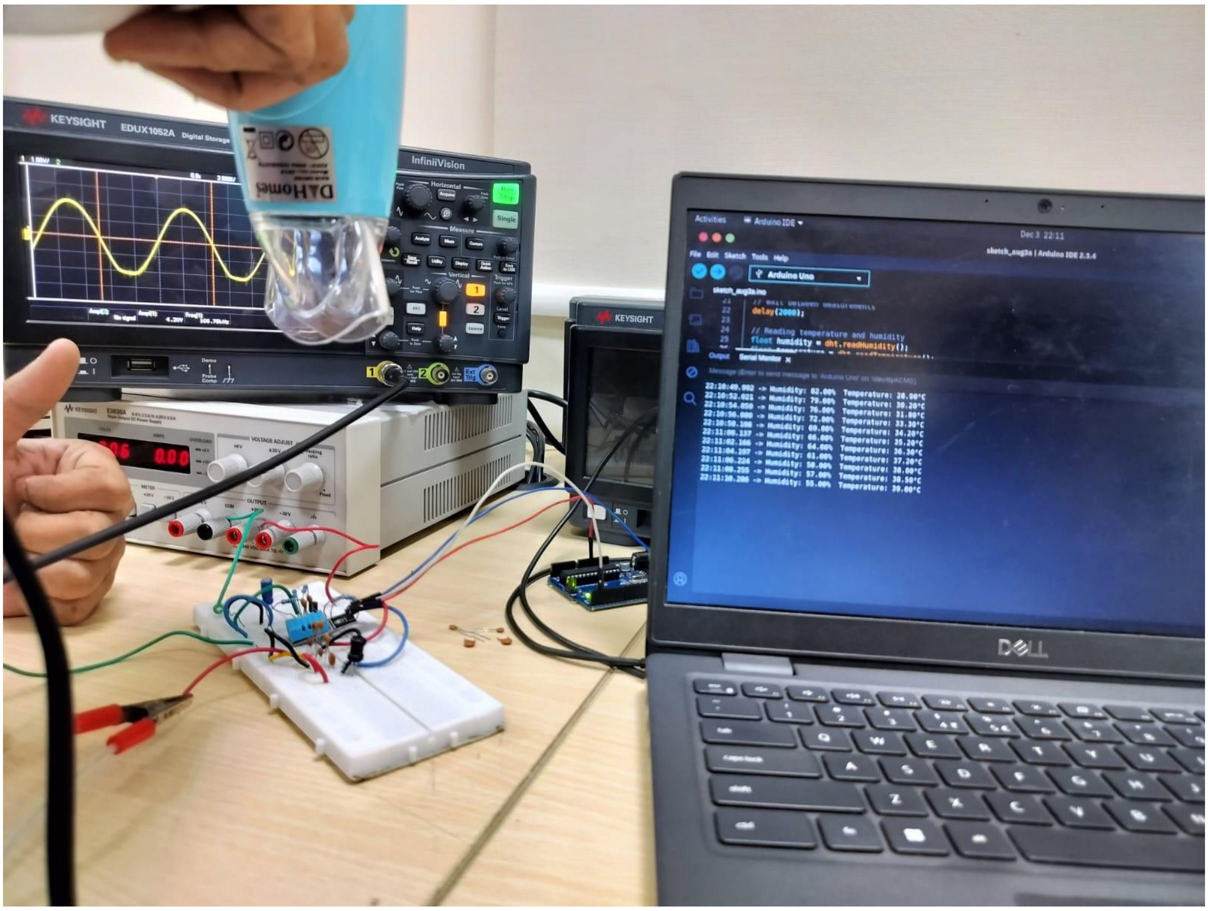

Graph of V1 vs Frequency
105.5
105.0
Frequency (kHz)
6
8
10
12
14
16
18
20
V1 (Voltage)

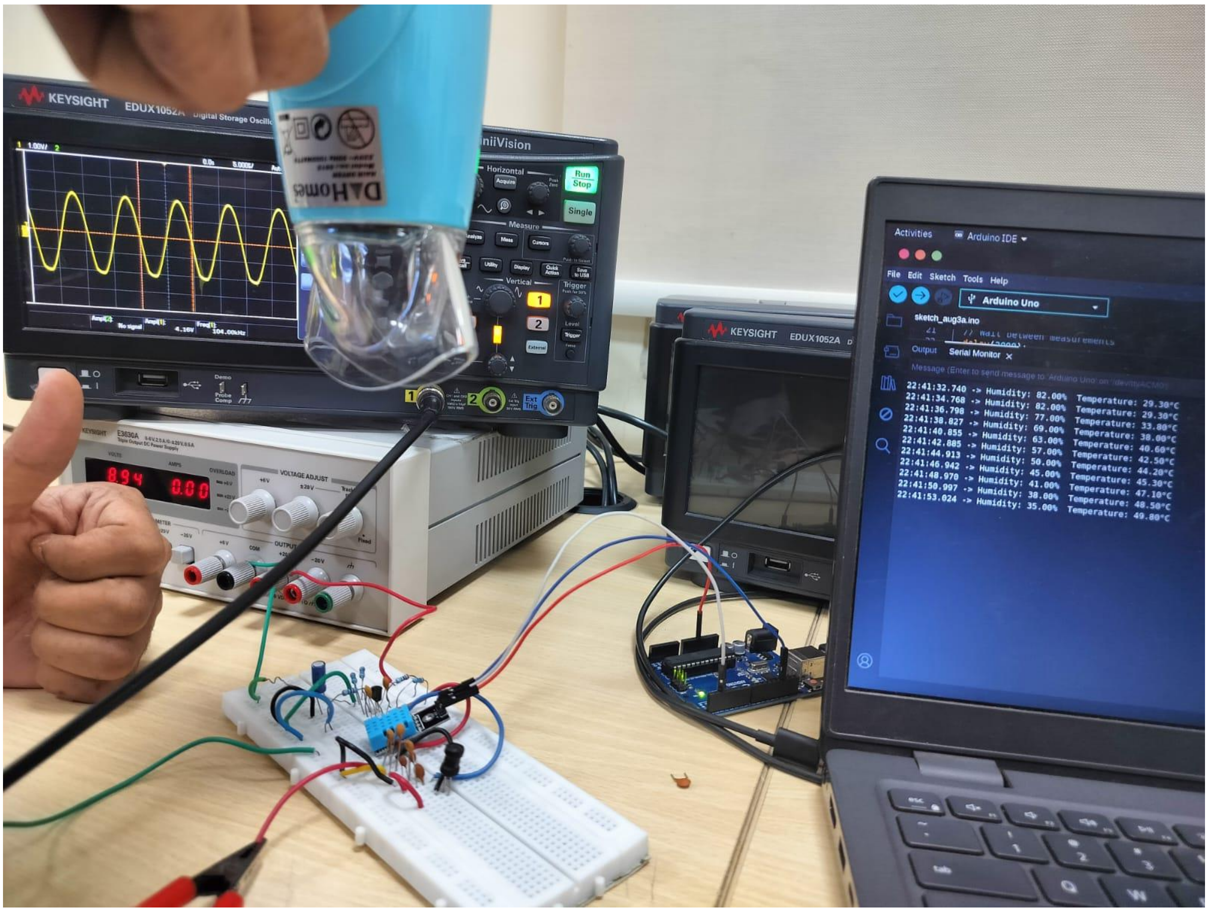


Graph of V1 vs Frequency
Frequency (kHz)
105
6
8
10
12
14
16
18
20
V1 (Voltage)


KEYSIGHT
InfiniiVision
Activities
Arduino IDE
File Edit Sketch Tools Help
Arduino Uno
sketch_aug3a.ino
Output
Serial Monitor
22:41:32.740 -> Humidity: 82.00% Temperature: 29.30°C
22:41:34.768 -> Humidity: 82.00% Temperature: 29.30°C
22:41:36.798 -> Humidity: 77.00% Temperature: 33.80°C
22:41:38.827 -> Humidity: 69.00% Temperature: 38.00°C
22:41:40.855 -> Humidity: 63.00% Temperature: 40.60°C
22:41:42.885 -> Humidity: 57.00% Temperature: 42.50°C
22:41:44.913 -> Humidity: 50.00% Temperature: 44.20°C
22:41:46.942 -> Humidity: 45.00% Temperature: 45.30°C
22:41:48.970 -> Humidity: 41.00% Temperature: 47.10°C
22:41:50.997 -> Humidity: 38.00% Temperature: 48.50°C


KEYSIGHT
EDUX1052A
InfiniiVision
Activities
Arduino IDE
Dec 3 22:11
sketch_aug3a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Arduino Uno
sketch_aug3a.ino
delay(2000);
// Reading temperature and humidity
float humidity = dht.readHumidity();
Output
Serial Monitor
DELL


KEYSIGHT
EDUX1052A
InfiniiVision
Activities
Arduino IDE
Dec 3 22:11
sketch_aug3a | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Arduino Uno
delay(2000);
// Reading temperature and humidity
float humidity = dht.readHumidity();
Output
Serial Monitor
DELL


KEYSIGHT
EDUX1052A
Activities
Arduino IDE
File Edit Sketch Tools Help
Arduino Uno
sketch_aug3a.ino
Output
Serial Monitor
22:41:32.740 -> Humidity: 82.00%
22:41:34.768 -> Humidity: 82.00% Temperature: 29.30°C
22:41:36.798 -> Humidity: 77.00% Temperature: 29.30°C
22:41:38.827 -> Humidity: 69.00% Temperature: 33.80°C
22:41:40.855 -> Humidity: 63.00% Temperature: 38.00°C
22:41:42.885 -> Humidity: 57.00% Temperature: 40.60°C
22:41:44.913 -> Humidity: 50.00% Temperature: 42.50°C
22:41:46.942 -> Humidity: 45.00% Temperature: 44.20°C
22:41:48.970 -> Humidity: 41.00% Temperature: 45.30°C
22:41:50.997 -> Humidity: 38.00% Temperature: 47.10°C
22:41:53.024 -> Humidity: 35.00% Temperature: 48.50°C
Temperature: 49.80°C

```
nikhilesht@pop-os:~/Desktop/ESW/check$ python3 predict.py
Enter the temperature (°C): 49
Predicted frequency at 49.0°C: 160.45 kHz
Enter the target frequency (kHz): 104
Target frequency: 104.00 kHz
Change factor: 0.6482
Temperature-adjusted capacitance: 6.674 nF
Required inductance (L) for target frequency: 0.420 mH
Required capacitance (C) for target frequency: 11.187 nF
Change in inductance: -0.580 mH
Change in capacitance: 6.487 nF
nikhilesht@pop-os:~/Desktop/ESW/check$
```

```
nikhilesht@pop-os:~/Desktop/ESW/check$ python3 predict.py
Enter the temperature (°C): 38
Predicted frequency at 38.0°C: 120.74 kHz
Enter the target frequency (kHz): 104
Target frequency: 104.00 kHz
Change factor: 0.8613
Temperature-adjusted capacitance: 5.640 nF
Required inductance (L) for target frequency: 0.742 mH
Required capacitance (C) for target frequency: 6.335 nF
Change in inductance: -0.258 mH
Change in capacitance: 1.635 nF
```

CONCLUSIONS:

- It is a simple and reliable high-frequency oscillator based on an LC tank circuit.
- The capacitive feedback ensures stable and sustained oscillations.
- It is widely used in RF communication, signal generation, and timing applications.
- The oscillator's frequency can be easily adjusted by tuning the inductor or capacitors.