Environmental Monitoring

Using Internet of Things

INTRODUCTION TO ENVIRONMENTAL MONITOR:

An environmental monitor is a valuable device used to collect and assess data related to the surrounding environment. It plays a crucial role in various applications, including weather forecasting, climate research, industrial safety, agriculture, and smart home systems. Environmental monitors are designed to measure a wide range of parameters that provide insights into the state of the environment, helping individuals and organizations make informed decisions.

Working Principle :

- **Sensor Array**: Environmental monitors are equipped with an array of sensors, each designed to measure specific environmental parameters. These sensors may include but are not limited to temperature, humidity, air quality, atmospheric pressure, wind speed, and precipitation.

- **Data Acquisition**: The sensors continuously collect data from their respective parameters. Some sensors use analog signals, while others provide digital data directly. In some cases, analog signals are converted to digital format for easier processing.

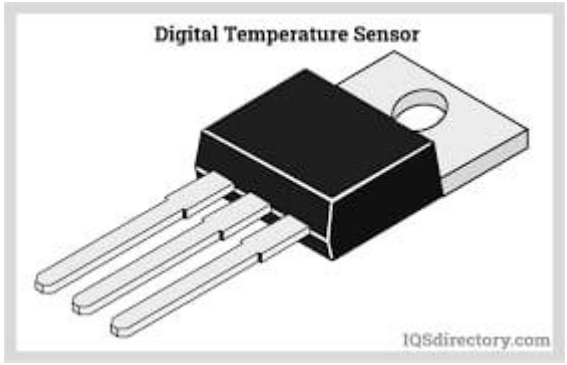
- **Data Processing**: The collected data is processed within the monitor's internal electronics. This may involve data filtering, calibration, or data fusion to provide accurate and meaningful information.

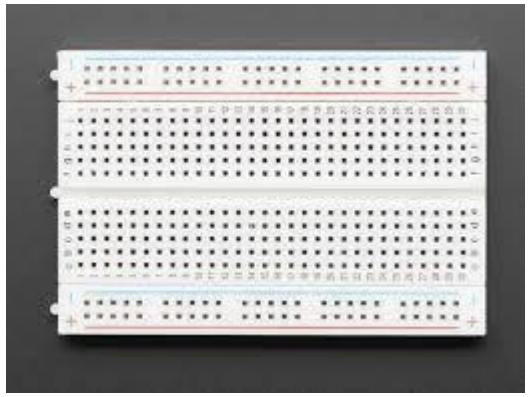
- **Data Storage**: Environmental monitors typically have storage capabilities, such as internal memory or external storage options. Data is stored for later retrieval and analysis.

- **Connectivity**: Many modern environmental monitors are equipped with connectivity options, such as Wi-Fi, Ethernet, or cellular connections. This allows the monitor to transmit real-time or periodic data to remote servers, databases, or cloud platforms.

Components Required :

- ❑ Temperature Sensor (DS 18B20)
- ❑ Jumper Cable
- ❑ Bread Board
- ❑ Raspberry Pi Pico
- ❑ Wifi Module (ESP 32)





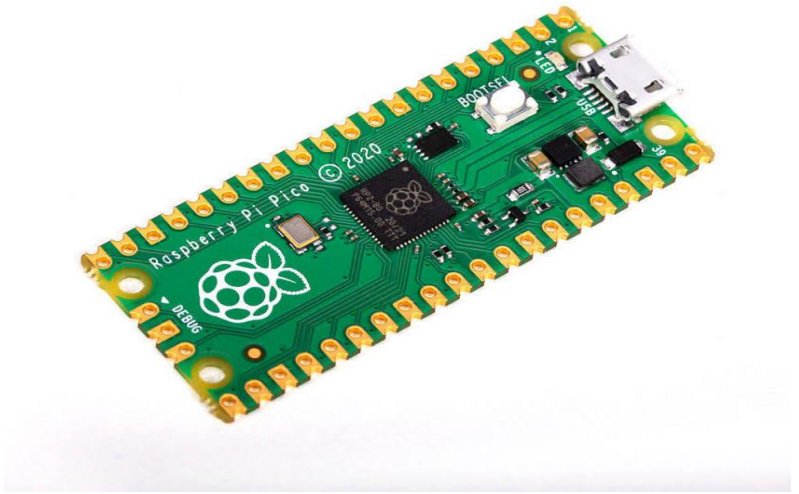
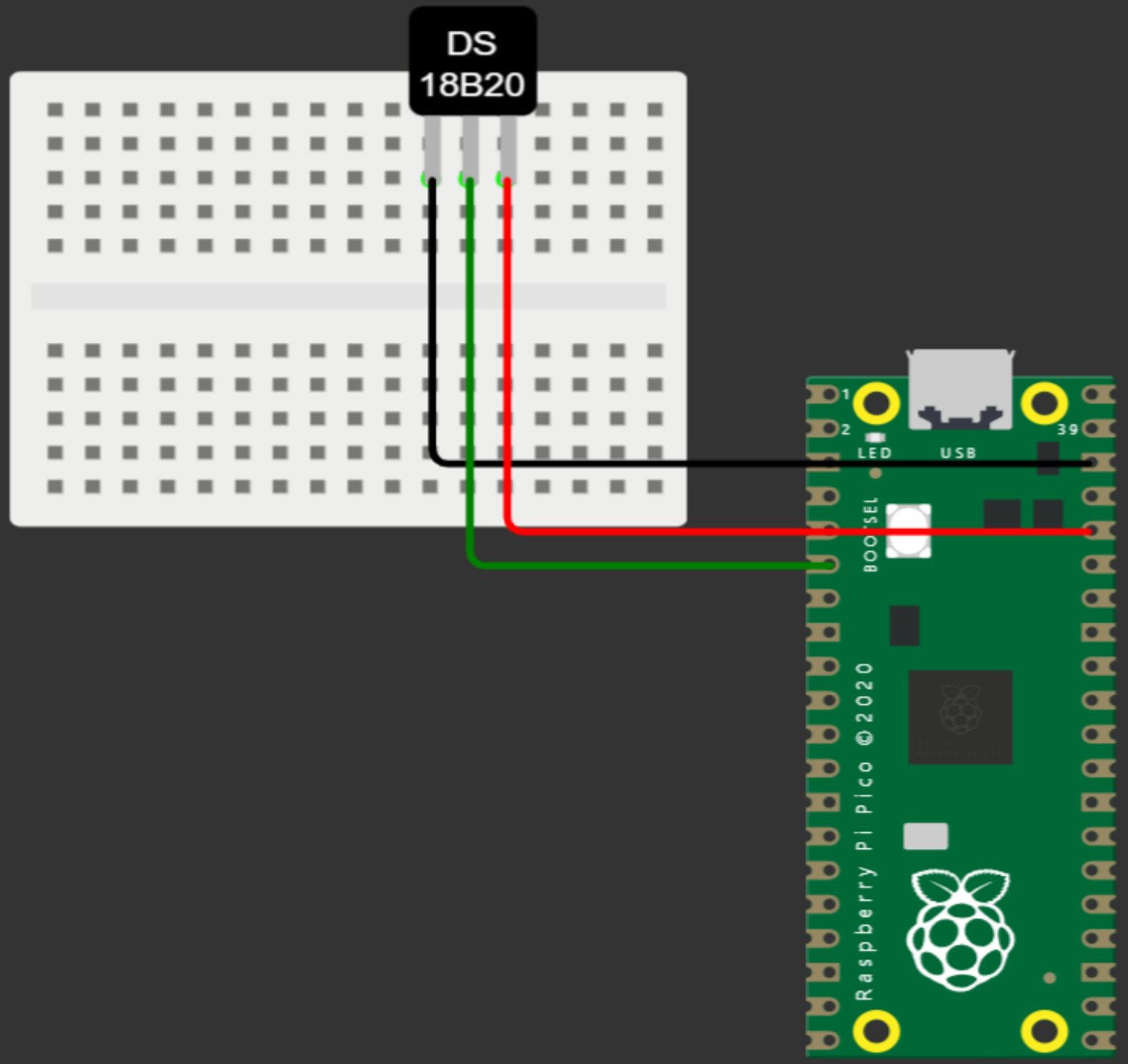


DIAGRAM :



PYTHON CODE :

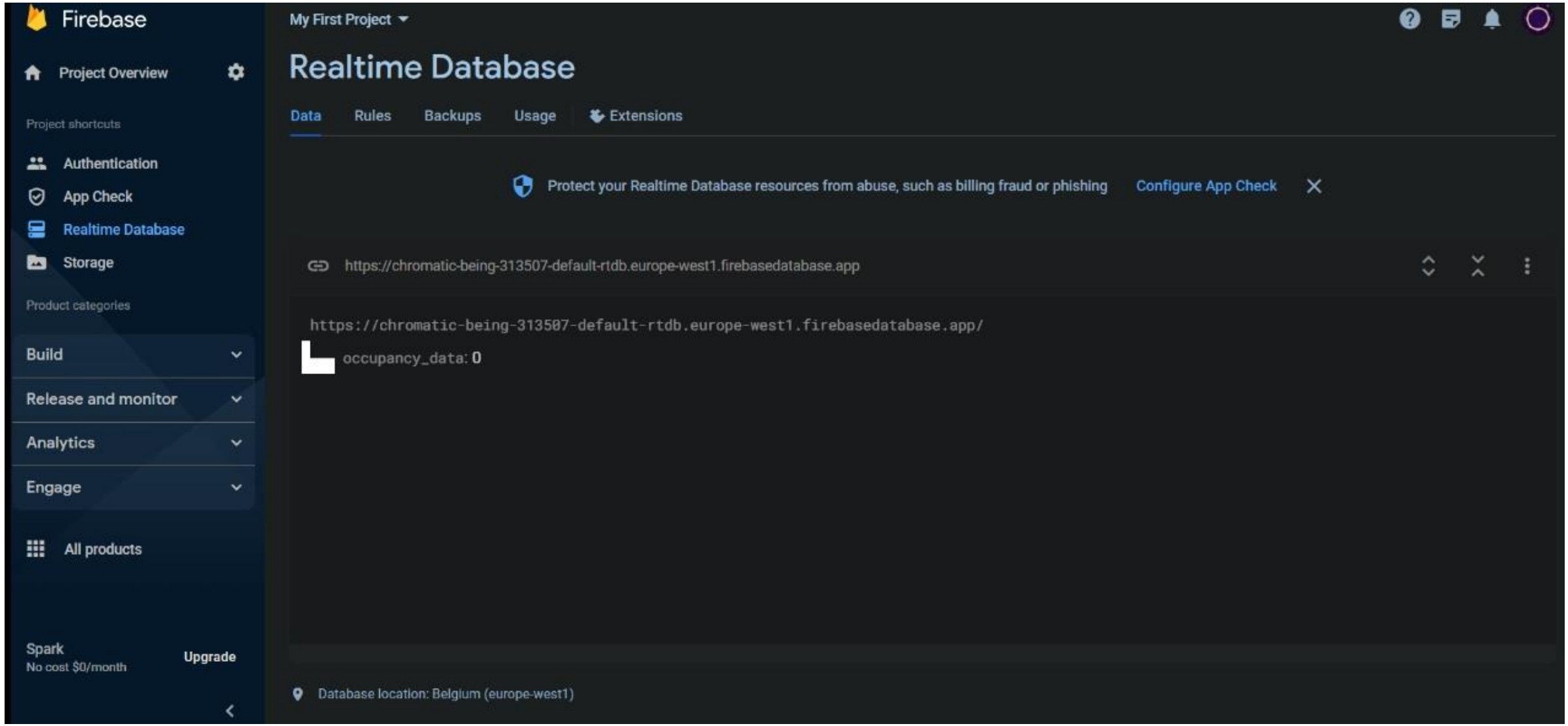
```
import network
import urequests
 as requestsimport dht  # Replace with the appropriate sensor
library
# Wi-Fi credentialsWIFI_SSID =
"Your_WiFi_SSID"WIFI_PASSWORD = "Your_WiFi_Password"
# Your Firebase Realtime Database URLFIREBASE_URL =
https://your-firebase-database-url.firebaseio.com
# Initialize Wi-Fi connectionwifi =
network.WLAN(network.STA_IF)wifi.active(True)wifi.connect(
WIFI_SSID, WIFI_PASSWORD)
# Initialize the DHT sensor (replace with your
sensor)dht_sensor = dht.DHT22(Pin(4))
  # Use the appropriate pin for your sensor
# Function to send data to Firebasedef
send_data_to_firebase(temperature, humidity):    data = {
"temperature": temperature,       "humidity": humidity    }
response = requests.post(FIREBASE_URL +
"/sensor_data.json", json=data)
 if response.status_code == 200:
      print("Data sent to Firebase successfully.")
  else:
    print("Failed to send data to Firebase.")
```

```
# Main loop to read sensor data and send to Firebasewhile
True:
  try:
  dht_sensor.measure()      temperature =
dht_sensor.temperature()      humidity =
dht_sensor.humidity()
send_data_to_firebase(temperature, humidity)
  except Exception as e:
 print("Error reading sensor data:", e)
  # Adjust the sleep time as needed (e.g., every 5 minutes)
time.sleep(300)
```

FIRE BASE CODE :

```
import firebase_admin
from firebase_admin import credentials, db
import time
# Replace with your Firebase service account credentials JSON file
cred = credentials.Certificate("path/to/your-service-account-key.json")
# Initialize Firebase appfirebase_admin.initialize_app(cred,
 {
  'databaseURL': 'https://your-firebase-database-url.firebaseio.com'})
```

OUTPUT :


Firebase
My First Project
Project Overview
Realtime Database
Project shortcuts
Data
Rules
Backups
Usage
Extensions
Authentication
App Check
Realtime Database
Storage
Protect your Realtime Database resources from abuse, such as billing fraud or phishing
Configure App Check
https://chromatic-being-313507-default-rtdb.europe-west1.firebasedatabase.app
Product categories
https://chromatic-being-313507-default-rtdb.europe-west1.firebasedatabase.app/
Build
occupancy_data: 0
Release and monitor
Analytics
Engage
All products
Spark
No cost $0/month
Upgrade
Database location: Belgium (europe-west1)

JAVA SCRIPT CODE :

```
<!DOCTYPE html>
<html>
<head>
   <title>Temperature and Humidity Data</title>
   <!-- Include the Firebase JavaScript SDK -->
   <script src="https://www.gstatic.com/firebasejs/9.6.0/firebase-app.js"></script>
   <script src="https://www.gstatic.com/firebasejs/9.6.0/firebase-database.js"></script>
</head>
<body>
   <h1>Temperature and Humidity Data</h1>

   <!-- Input fields for temperature and humidity -->
   <label for="temperature">Temperature:</label>
   <input type="text" id="temperature" placeholder="Enter temperature">

   <label for="humidity">Humidity:</label>
   <input type="text" id="humidity" placeholder="Enter humidity">

   <button onclick="sendDataToFirebase()">Send Data</button>
```

```
<script>
  // Your web app's Firebase configuration
  const firebaseConfig = {
    apiKey: "YOUR_API_KEY",
    authDomain: "YOUR_AUTH_DOMAIN",
    projectId: "YOUR_PROJECT_ID",
    storageBucket: "YOUR_STORAGE_BUCKET",
    messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
    appId: "YOUR_APP_ID"
  };

  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);

  // Function to send data to Firebase
  function sendDataToFirebase() {
    const temperature =
document.getElementById("temperature").value;
    const humidity =
document.getElementById("humidity").value;

    const db = firebase.database();
    const dataRef = db.ref("sensor_data");
```

```
 dataRef.push({
            temperature: temperature,
            humidity: humidity
          });

          console.log("Data sent to Firebase successfully.");
      }
    </script>
</body>
</html>
```

Team Members Details

Role in Team	Name	Branch Name	Year
Team Leader	Vimalanathan.G	CSE	3rd Year
Team Member 1	Harivignesh.M	CSE	3rd Year
Team Member 2	Abinesh.M	CSE	3rd Year
Team Member 3	Sivaraman	CSE	3rd Year