

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)

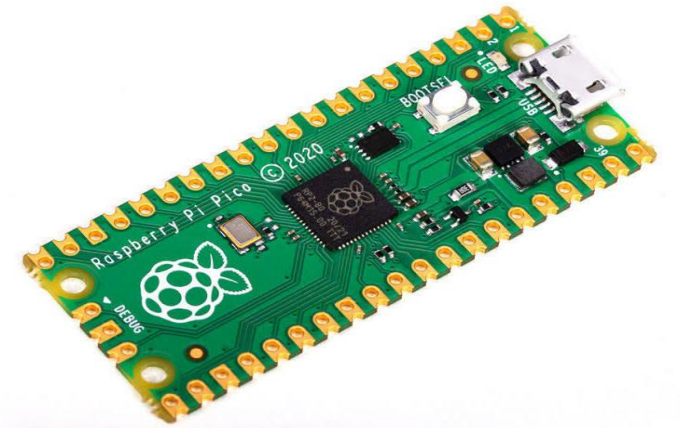
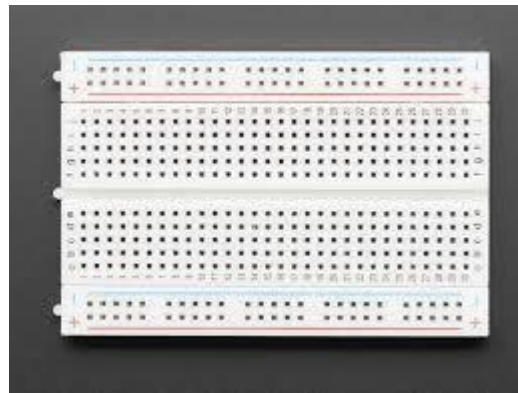
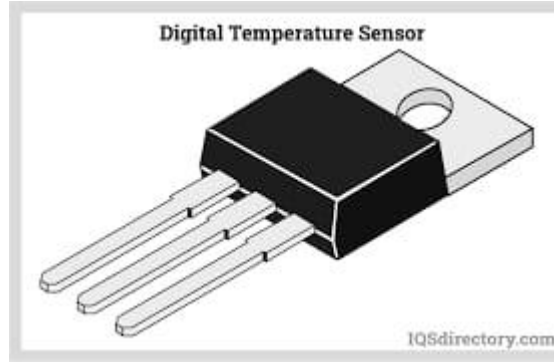
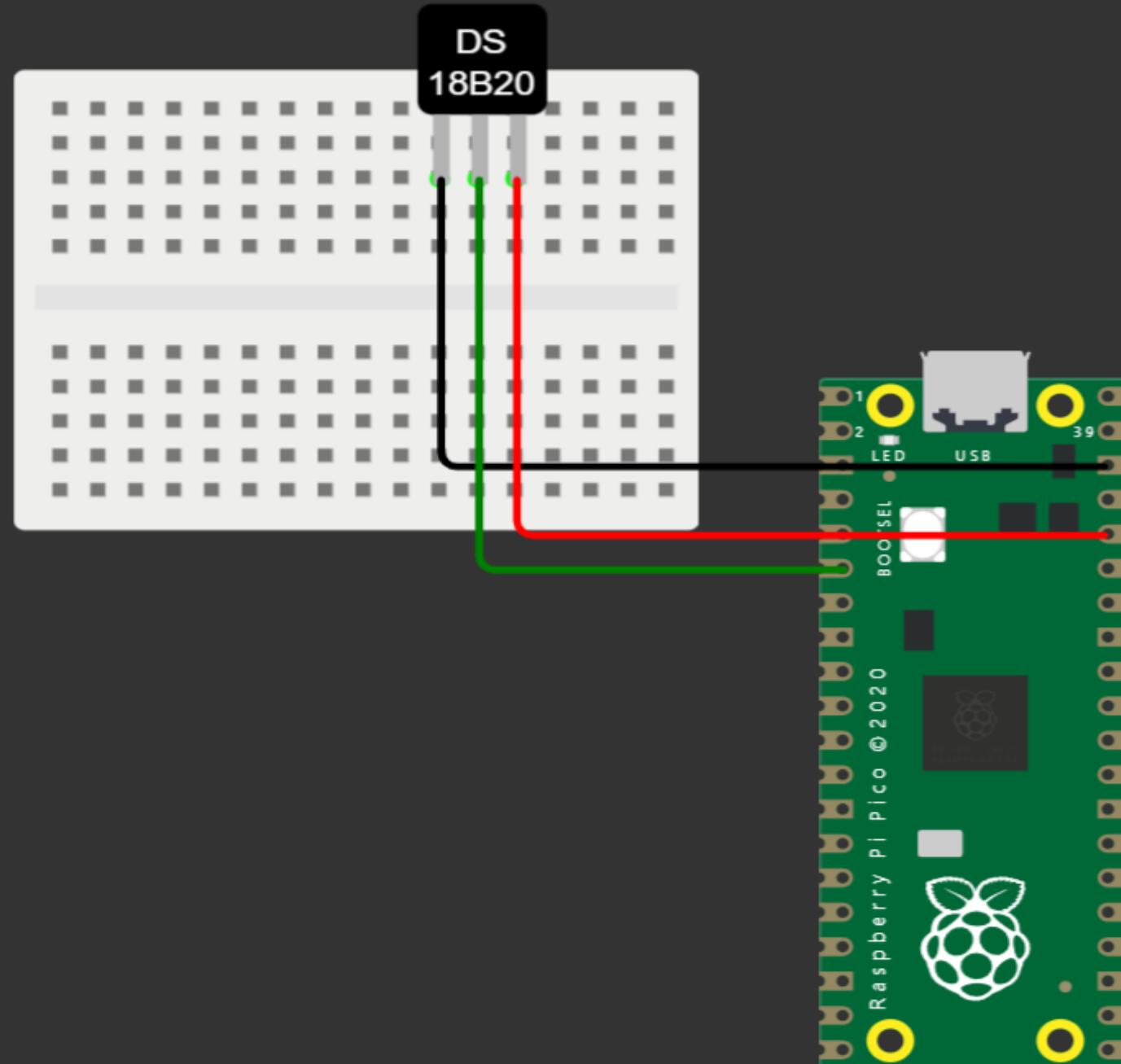


DIAGRAM:



Program :

```
import machine
import onewire
import ds18x20
import time
import network
import urequests

# Wi-Fi credentials
SSID = "YourWiFiSSID"
PASSWORD = "YourWiFiPassword"

# DS18B20 setup
data_pin = machine.Pin(4)
ds_sensor = ds18x20.DS18X20(onewire.OneWire(data_pin))
roms = ds_sensor.scan()

# Connect to Wi-Fi
wifi = network.WLAN(network.STA_IF)
wifi.active(True)
wifi.connect(SSID, PASSWORD)

while not wifi.isconnected():
    pass

print("Connected to Wi-Fi")
```

```
# Beeceptor endpoint
BEECEPTOR_URL = "https://monitoringplatform.free.beeceptor.com"

while True:
    ds_sensor.convert_temp()
    time.sleep_ms(750)
    for rom in roms:
        temperature = ds_sensor.read_temp(rom)

        # Send temperature data to Beeceptor
        data = {"temperature": temperature}
        response = urequests.post(BEECEPTOR_URL, json=data)

        # Print the response from Beeceptor
        print("Temperature sent to Beeceptor. Response:",
response.content)

    time.sleep(300) # Wait for 5 minutes before sending the next data
                    (adjust as needed)
```

HTTP Mocking Rules on End User

Method - POST

Response Headers :

```
{
  "method": "POST",
  "path": "/data",
  "response": {
    "statusCode": 200,
    "headers": {
      "Content-Type": "application/json"
    },
    "body": {
      "message": "Received POST data",
      "temperature": "{{temperature}}"
    }
  }
}
```

Method - GET

Response Headers

```
{
  "method": "GET",
  "path": "/data",
  "response": {
    "statusCode": 200,
    "headers": {
      "Content-Type": "application/json"
    },
    "body": {
      "message": "Received POST data",
      "temperature": "{{query.temperature}}"
    }
  }
}
```

Method - GET

Response Headers

```
{"Temperature:body.temperature"}
```

Usage Scenario :

- ☐ You can expand the environmental monitoring system by adding more sensors to measure other parameters, such as humidity, light, or soil moisture, depending on the specific needs of your greenhouse and the plants you are growing.
- ☐ Implement data logging and visualization tools to view historical temperature data and make informed decisions for greenhouse management
- ☐ Configure the monitoring system to be accessible remotely through a web or mobile app so you can check temperature data and receive alerts from anywhere.
- ☐ Integrate other environmental control mechanisms, like cooling or heating systems, based on the temperature data collected.

Team Members Details

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