

## Edge Computing Lab

### Class: TY-AIEC

School of Computing, MIT Art Design Technology University

Academic Year: 2024-25

#### Experiment No. 3

##### Title

DHT11 Sensor and Alert System using Blynk IoT

##### Objective:

The goal of this project is to create a system with a DHT11 sensor interfaced with a Raspberry Pi that monitors humidity levels and sends alerts via the Blynk IoT platform when humidity exceeds 70%.

##### Materials:

- Raspberry Pi (any model with GPIO pins)
- DHT11 Temperature and Humidity Sensor
- Breadboard and jumper wires
- Resistors (typically 10k $\Omega$  for DHT11 pull-up)
- Blynk Mobile App
- Internet connection

##### Procedure:

###### Task 1: Connect a DHT11 to the Raspberry Pi

**1. Initial Setup:** Ensure your Raspberry Pi is set up with the latest version of Raspbian OS and is connected to the internet.

**2. Wiring:** Connect the DHT11 sensor to the Raspberry Pi GPIO pins.

- VCC pin to a 5V pin on the Raspberry Pi.
- Data pin to a GPIO pin (e.g., GPIO4).
- GND pin to a ground pin on the Raspberry Pi.
- Place a 10k $\Omega$  resistor between VCC and the Data pin (this acts as a pull-up resistor).

###### Task 2: Program the Raspberry Pi

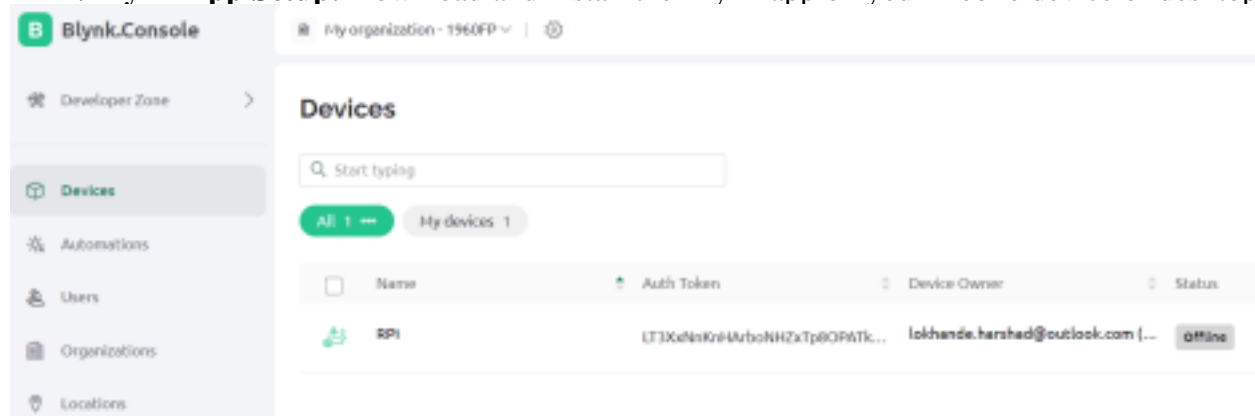
**1. Install Libraries:** Install the DHT11 Python library by running ``sudo pip install dht11`` in the terminal.

**2. Coding:**

- Write a Python script that reads humidity and temperature from the DHT11 sensor.
- Include a conditional statement to check if the humidity is greater than 70%. - If the condition is true, use the Blynk library to send a notification.

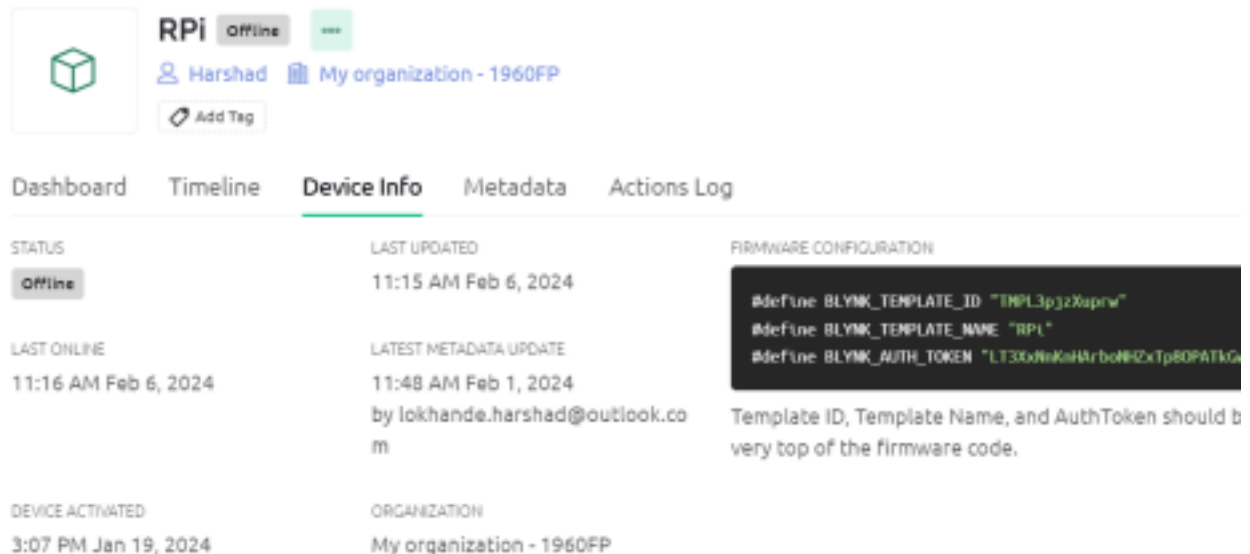
### Task 3: Configure the Blynk IoT

**1. Blynk App Setup:** Download and install the Blynk app on your mobile device or desktop.



### **2. Create a New Project:**

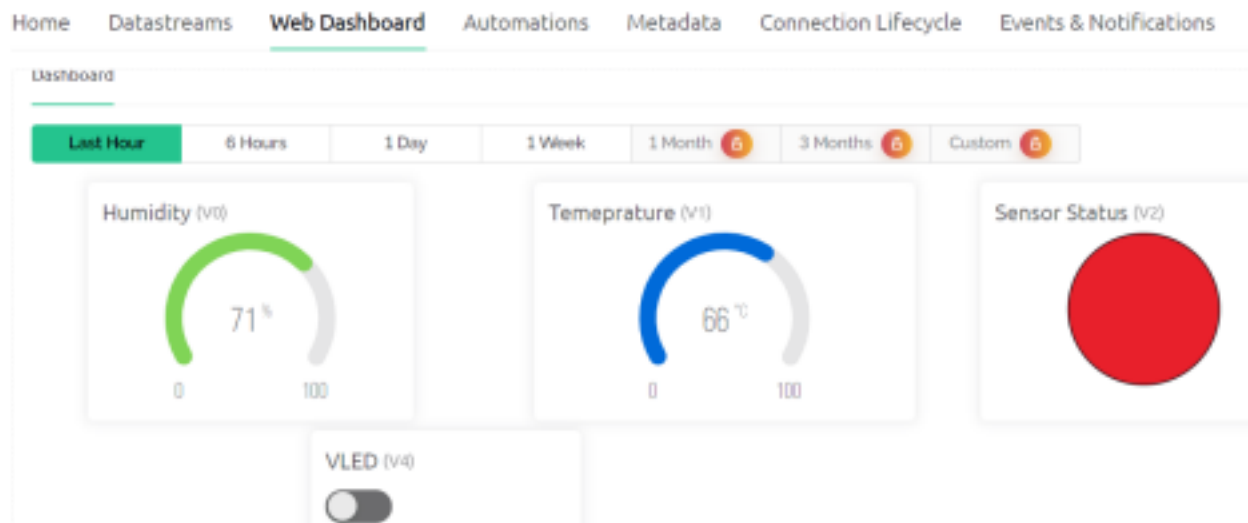
- Open the app and create a new project.
- Select the device as Raspberry Pi and the connection type as Wi-Fi.
- An authentication token will be sent to your email, which will be used in your Python script.



### Task 4: Generate the GUI on Mobile / Desktop

#### **1. Adding Widgets:**

- In the Blynk app project, add a Gauge widget for displaying humidity.



- Add a Notification widget that will be used to send alerts.
- Add the DataStream

The screenshot shows the Blynk Datastreams page. At the top, there are navigation tabs: Home, Datastreams (selected), Web Dashboard, Automations, Metadata, Connection Lifecycle, and Events & Notifications. Below the tabs, there's a search bar labeled 'Search datastream'. Below the search bar, there's a table listing data streams.

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min
1	Humidity	Humidity		V0	Double	%	false	0
2	Temperature	Temperature		V1	Double	°C	false	0
3	SensorStatus	SensorStatus		V2	Integer		false	0

### Task 5: Apply Analytics for Alert Generations in Blynk IoT

#### **1. Script Enhancement:**

- Modify the Python script to send data to Blynk using the Virtual Pins.
- Use Blynk's `eventor` feature to set up the logic for alert generation based on the humidity value.

#### **2. Data Logging:** Use Blynk's Super-Chart widget to log and display humidity data over time.

#### **Execution:**

1. Run the Python script on the Raspberry Pi.
2. Ensure that the script is reading the DHT11 sensor data correctly.
3. Monitor the Blynk app dashboard for real-time data.

#### **Code:**

```

import time

import Adafruit_DHT

import BlynkLib

# Replace with your Blynk Auth Token

BLYNK_AUTH_TOKEN = '_XnsCg95jOshh7O2jiBuMq7oWbVA6iCr'


# Set the sensor type and the GPIO pin

DHT_SENSOR = Adafruit_DHT.DHT11

DHT_PIN = 4 # GPIO pin number where the sensor's data pin is connected


# Initialize Blynk

blynk = BlynkLib.Blynk(BLYNK_AUTH_TOKEN)


# Function to read DHT11 data and send it to Blynk

@blynk.on("connected")

def read_and_send_data():

    humidity, temperature = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)

    if humidity is not None and temperature is not None:

        print(f'Temperature: {temperature}C, Humidity: {humidity}%')

        blynk.virtual_write(1, temperature) # Send temperature to virtual pin
V1

        blynk.virtual_write(0, humidity)      # Send humidity to virtual pin
V2

    else:

        print('Failed to retrieve data from sensor')

```

```
# Main loop

while True:

    read_and_send_data() # Read and send data

    blynk.run() # Keep Blynk connection alive

    time.sleep(1) # Wait for 10 seconds before reading again
```

## Results:

- The system should accurately read the humidity levels from the DHT11 sensor.
- The Blynk app should display real-time humidity data.
- Upon reaching the 70% humidity threshold, the system should send a notification alert.

## Discussion:

- Potential issues could include inaccurate readings from the DHT11 sensor, which may require calibration.
- Network instability could affect the performance of the Blynk app notifications.

## Conclusion:

The system successfully integrates a DHT11 sensor with a Raspberry Pi to monitor humidity levels and uses the Blynk IoT platform to send alerts when thresholds are exceeded, demonstrating the viability of IoT for home automation and monitoring tasks.

## WriteUp:

Name : Sudip Satish Konde Class : Tr(AIEE)  
Roll No : 2223118 Sub : ECI

### Experiment No. 3

DHT11 ON Raspberry Pi and displaying  
values on Blynk IoT.

What is Blynk IoT & how does it facilitate  
IoT application development?

Blynk IoT is a cloud-based IoT platform  
that enable developers to remotely monitor  
& control devices via a mobile app.

Key Features :

- No Complex Coding
- Cross platform
- Real-time data
- Cloud Integration.

② i) Install Blynk app

ii) Create a new project

- Select "Rasp Pi" as the device
- Connection type: WiFi or Ethernet.

iii) Get the Authentication token.

- After creating the project blynk emails  
you an auth token.



• save this token for use in your python script.

③ Install dependencies :-

```
pip install blynklib Adafruit-DHT
```

Python Script.

```
import blynklib.  
import Adafruit-DHT.
```

```
Blynk_AUTH = 'Auth-Token'
```

```
DHT_SENSOR = Adafruit-DHT.DHT11
```

```
DHT_PIN = 4
```

```
blynk = blynklib.Blynk(Blynk_AUTH)
```

```
@blynk.handle_event('readV1')
```

```
def read_temp():
```

```
    humidity, temperature = Adafruit-DHT.
```

```
    read(DHT_SENSOR, DHT_PIN).
```

```
    if temperature is not None:
```

```
        blynk.virtual_ write(1, temperature)
```

```
@Blynk.handle_event('readV2')
```

```
def read_humidity():
```

if humidity is not None:  
blynk.virtual-write(2, humidity)

while True:  
blynk.run()

Run the Script:  
python blynk.DHT11.py

- ④
- i) Open the Blynk App & go to your project.
  - ii) Add widgets:
    - Gauge for temperature
    - Gauge for Humidity.
  - iii) Configure virtual pins.
    - Set Temp widget to V1.
    - Set Humidity widget to V2.
  - iv) Save and Run.

- ⑤ Advantages of Blynk IoT over a Traditional web server.
- No Flask or hosting needed - cloud based solution
  - Easy mobile integration - Ready made UI
  - Real-time & Remote Access - No port forwarding.
  - Fast Development - No manual API Setup.
  - Secure Cloud storage - No need for local storage.