

Edge Computing Laboratory
Lab Assignment 3

Name: HARSHIT PRAKASH SHETTY
Class: TY-AIEC-B
Enrollment No: MITU22BTCS0321
Roll No: 2223617

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 10kΩ for DHT11 pull-up)
- Blynk Mobile App
- Internet connection

Procedure:

Task 1: Connect a DHT11 to the Raspberry Pi

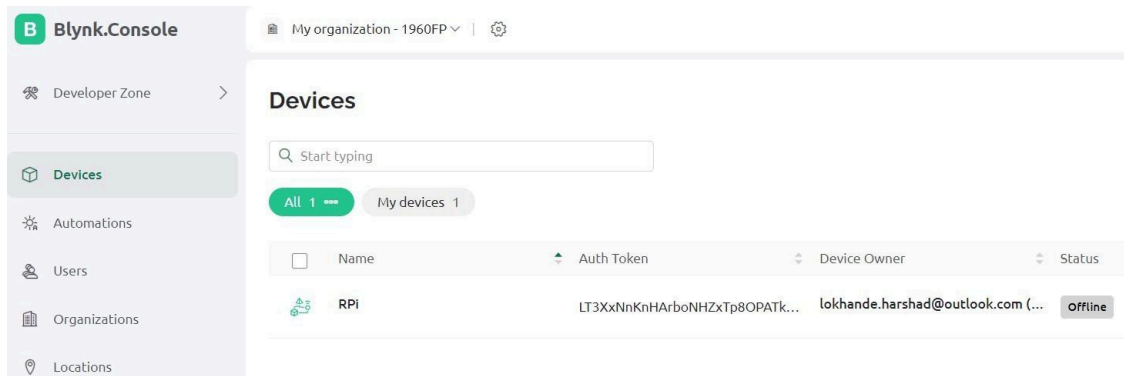
1. Initial Setup: Ensure your Raspberry Pi is set up with the latest version of Raspbian OS and is connected to the internet.
2. Wiring: Connect the DHT11 sensor to the Raspberry Pi GPIO pins.
 - VCC pin to a 5V pin on the Raspberry Pi.
 - Data pin to a GPIO pin (e.g., GPIO4).
 - GND pin to a ground pin on the Raspberry Pi.
 - Place a 10kΩ 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.

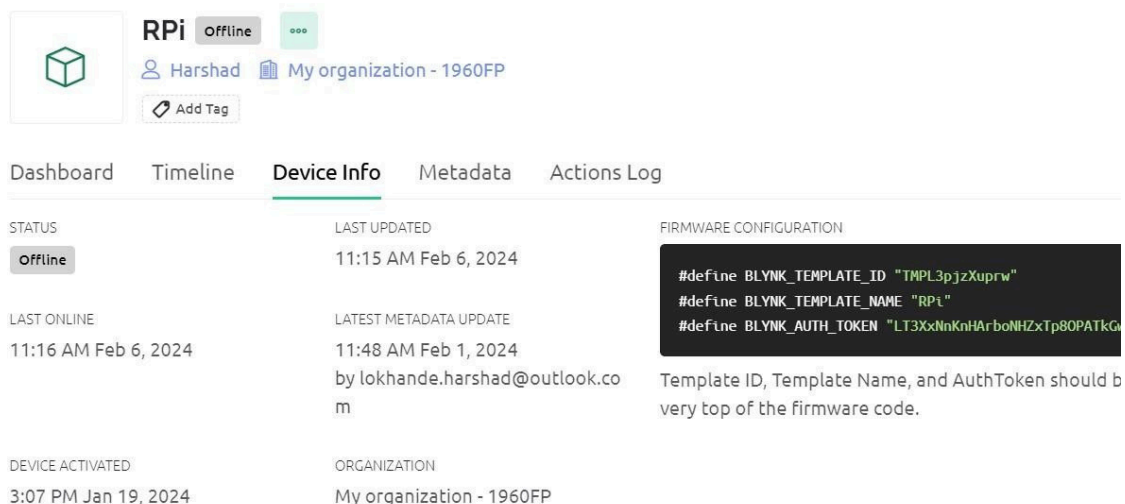
Task 3: Configure the Blynk IoT

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



2. Create a New Project:

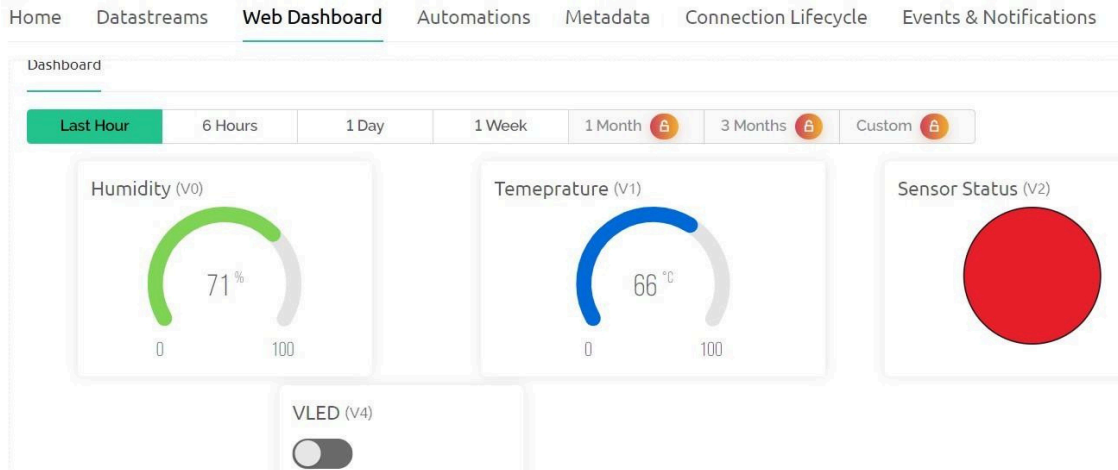
- Open the app and create a new project.
- Select the device as Raspberry Pi and the connection type as Wi-Fi.
- An authentication token will be sent to your email, which will be used in your Python script.



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

Home Datastreams Web Dashboard Automations Metadata Connection Lifecycle Events & Notifications									
<input type="text" value="Search datastream"/>									
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.
4. Test the system by artificially increasing the humidity to trigger the alarm.


New Automation




Send In-App Notification


RECIPIENTS

Harshad X

SUBJECT

TURN OF THE COOKER

MESSAGE

YOUR FOOD IS READY....
PLEASE TURN OF THE COOKER...!!

Organization name

Template name

Device name

Trigger value

Placeholders will be filled with the actual value after automation is triggered. Drag placeholder to Subject or Message body

Python Code:

```

import Adafruit_DHT
import RPi.GPIO as GPIO
from BlynkLib import Blynk
import time

# DHT11 Sensor Setup
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 4

# Blynk IoT Setup
BLYNK_AUTH_TOKEN = "_t3Bu6MIWbPE7DCifMI87D-aBvIN5wwq"

HUMIDITY_THRESHOLD = 70 # Alert if humidity > 70%

def read_sensor():
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
    return humidity, temperature

def send_blynk_notification(message):
    blynk.log_event("high_humidity", message)
    print(f"ALERT: {message}")
    try:
        while True:

```

```

humidity, temperature = read_sensor()

if humidity is not None and temperature is not None:
    print(f"Temp: {temperature:.1f}°C | Humidity: {humidity:.1f}%")

    # Send data to Blynk
    blynk.virtual_write(0, temperature) # Virtual Pin V0 (Temp)
    blynk.virtual_write(1, humidity) # Virtual Pin V1 (Humidity)

    if humidity > HUMIDITY_THRESHOLD:
        alert_msg = f"High Humidity Detected: {humidity}%"
        send_blynk_notification(alert_msg)

    else:
        print("Failed to read sensor data!")

    time.sleep(2) # Read every 2 seconds
    blynk.run()

except KeyboardInterrupt:
    print("\nExiting...") finally:
        GPIO.cleanup() Output:

```

```

Temp: 25.0°C | Humidity: 65.0%
Temp: 25.1°C | Humidity: 68.0%
Temp: 25.2°C | Humidity: 72.0%
ALERT: High Humidity Detected: 72.0%
Temp: 25.1°C | Humidity: 71.0%
ALERT: High Humidity Detected: 71.0%
Exiting...

```