

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 Ω 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 Ω resistor between VCC and the Data pin (this acts as a pull-up resistor).

Task 2: Program the Raspberry Pi

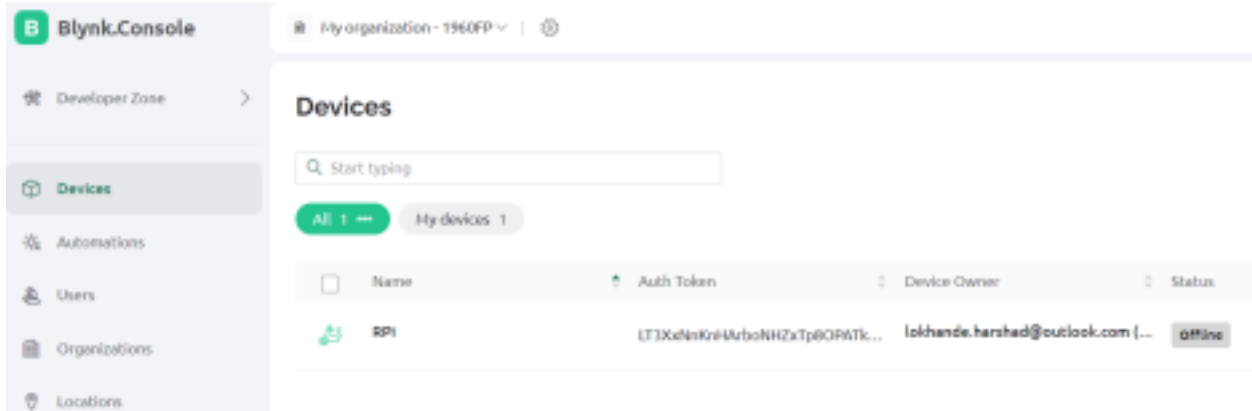
1. Install Libraries: Install the DHT11 Python library by running `sudo pip install dht11` in the terminal.

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.

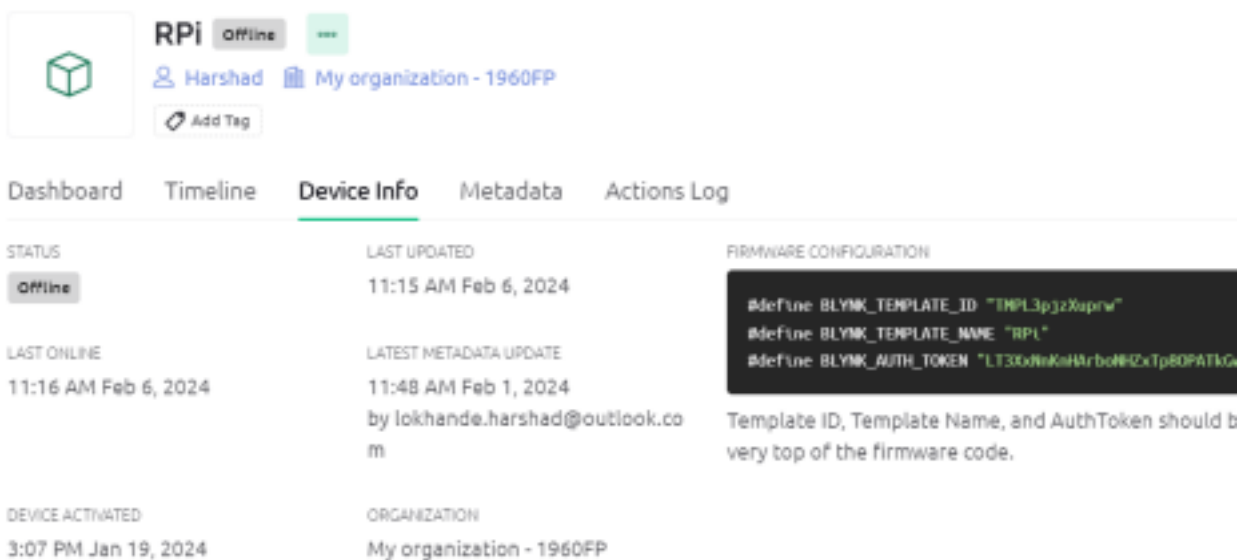
Task 3: Configure the Blynk IoT

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



2. Create a New Project:

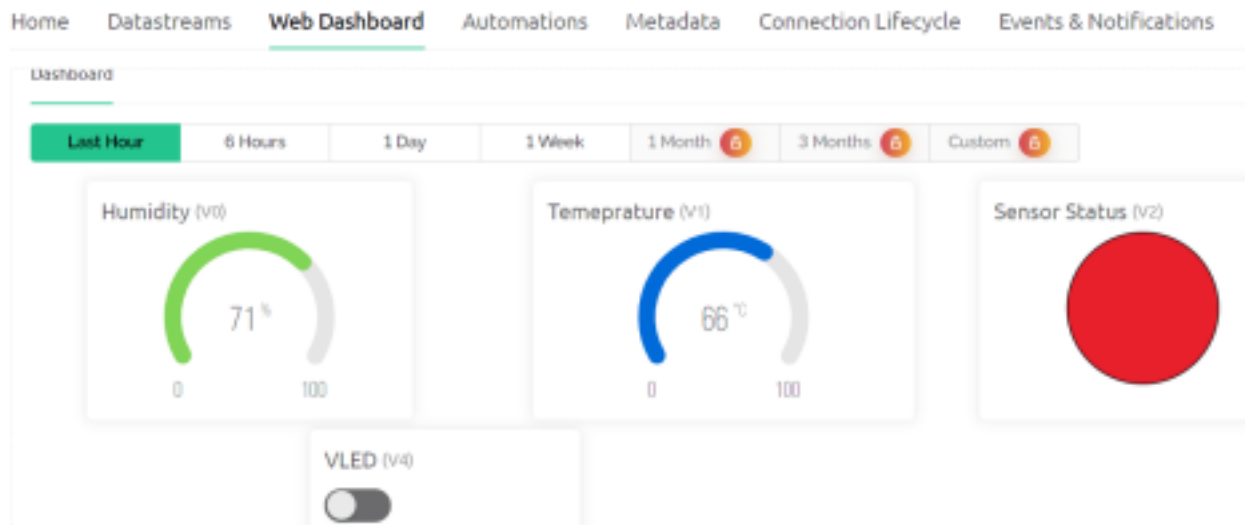
- Open the app and create a new project.
- Select the device as Raspberry Pi and the connection type as Wi-Fi.
- An authentication token will be sent to your email, which will be used in your Python script.



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 is a search bar labeled 'Search datastream'. The main area contains a table with the following data:

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

Task 5: Apply Analytics for Alert Generations in Blynk IoT

1. Script Enhancement:

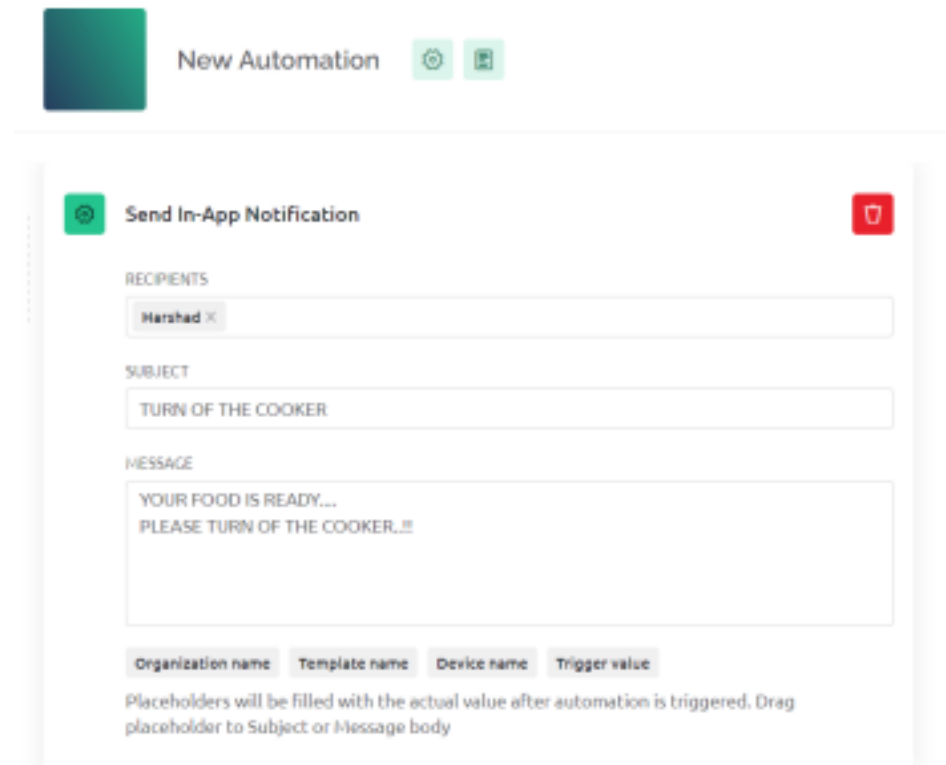
- Modify the Python script to send data to Blynk using the Virtual Pins.
- Use Blynk's 'eventor' feature to set up the logic for alert generation based on the humidity value.

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.



The screenshot shows the 'New Automation' interface in the Blynk app. At the top, there's a 'New Automation' button with a gear icon and a document icon. Below this, a card titled 'Send In-App Notification' is displayed. The card has a green header with a gear icon and a red header with a shield icon. The card contains the following fields:

- RECIPIENTS:** A text input field containing 'Harshad X'.
- SUBJECT:** A text input field containing 'TURN OF THE COOKER'.
- MESSAGE:** A text input field containing 'YOUR FOOD IS READY... PLEASE TURN OF THE COOKER..!!'.

At the bottom of the card, there are four placeholder buttons: 'Organization name', 'Template name', 'Device name', and 'Trigger value'. Below these buttons, a note states: 'Placeholders will be filled with the actual value after automation is triggered. Drag placeholder to Subject or Message body'.

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

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.

FAQ's:

Name : Deepika Nodgi

Batch : TY AEEC - C.

Roll no : 2223928

Subject : EC lab.

Experiment No 3: DHT11 on Raspberry Pi and displaying values on blynk.

Questions:

Q1] What is Blynk IoT and how does it facilitate IoT application development?

Ans: Blynk IoT is a platform that allows developers to build and control IoT operations using a mobile app and cloud base server.

• Features:

i] Drag and drop widgets for dashboards.

ii] Supports Raspberry Pi, ESP8266, Arduino, etc.

iii] Works with WiFi, Ethernet, GSM or bluetooth.

iii] Real time data monitoring and remote device control.

Adv: i] No need for complex web servers.

ii] Quicker Prototyping with minimal coding.

iii] cloud storage for sensor data

iv] User friendly.

Q2] Explain the steps to set up a blynk project and generate an authentication token for Raspberry Pi

Ans: i] install Blynk App

ii] ~~Signa~~ Sign up.

iii] create New Project:

- Select Raspberry Pi as device.

- Choose wifi / Ethernet as connection type.

iv] Get the Authentication Token:

Blynk will email you the Auth Token.

v] Install Blynk Python Library on Raspberry Pi.

```
pip install blynk-library-python
```

Q3] Write a Python script to read data from DHT and send it to Blynk IoT.

```
Ans: import BlynkLib
```

```
import Adafruit_DHT
```

```
import time
```

```
Blynk_AUTH = "your Auth token"
```

```
blynk = BlynkLib.Blynk(BLYNK_AUTH).
```

```
DHT_SENSOR = Adafruit_DHT.DHT11
```

```
DHT_PIN = 7.
```

```
def read_sensor():
```

```
    humidity, temperature = Adafruit_DHT.
```

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

```
    if humidity is not None and temperature  
        is not None:
```

```
        blynk.virtual_write(1, temperature)
```

```
        blynk.virtual_write(2, humidity)
```

```
while True :  
    read_sensor()  
    blynk.run()  
    time.sleep(5)
```

Q4] How can you configure a Blynk widget to display temperature and humidity values in real-time?

Ans: i] Open Blynk app → Select Your Project

ii] Add 'Label Value' Widgets:

- One for temperature (V1)
- One for humidity (V2)

iii] Configure widget Settings:

- Set the Input Pin to V1 (Temperature)
- Set the Input Pin V2 (Humidity)

iv] Run Python Script on Raspberry Pi

v] See live updates on the Blynk.

Q5] What are the four advantages of using Blynk IoT over a traditional web server for data visualization?

Ans: Advantages are:

i] Easy Drag and Drop setup.

ii] Blynk Cloud for sensor data.

iii] Mobile App support.

iv] Real time Monitoring.

v] No Port Forwarding.