

Name : Tanmay Khedekar
Class : TY-15 (AIEC)
Roll No : 2223122
Sub : EC - Lab

Assignment - 1

Experiment 1:- Introduce to Raspberry Pi,
OS Installation, Putty and VNC

Q1. What are the key hardware component of Raspberry Pi include how does it differ from traditional computers.

Raspberry Pi	Traditional computer
• size of Rasp Pi is smaller	size is larger
• Power consumption is Higher	Power consumption is Lower
• storage: microSD card slot for OS and data storage	• storage: use microSD instead of HDD/SSD.
• Performance is High	• Performance Limited processing

Q2. Describe the step-by-step process for installing an OS on a Raspberry Pi using Raspberry Pi imager

1. Download Raspberry Pi Imager from website
2. Insert a microSD card (minimum of 8GB) into your computer using an SD card reader.
3. open Raspberry Pi Imager and select the OS:- click choose OS and pick official OS like Raspberry Pi OS.
5. select the microSD card by clicking 'write' and wait for the process to complete
4. select storage
6. eject the microSD card safely and insert it into the Raspberry Pi.
- 7). connect Rasbe Pi by connecting it to a power source
- 8). follow the on-screen setup (wifi, username, password), etc.

Q3. How can putty be used to establish an SSH connection with Raspberry Pi? what are default login credentials?

→ Enable SSH on Raspberry Pi:-

- if using Raspberry Pi OS with a desktop, enable SSH via Raspberry Pi configuration

- If in headline mode, create a blank file named ssh in the boot partition of micro SD card.

email you can find

2) find Rasp Pi IP Address.

- Run `hostname -I` in Rasp Pi terminal
- use a router interface or an IP scanner tool to find it.

3) use Putty to connect :-

- open putty on your device
- enter Pi IP address in Hostname
- set port to 22 and select SSH
- click on.

4) Log in with default credentials.

- username : Pi
- Password : raspberry

Q4. explain the purpose of VNC in Raspberry Pi
How can you enable and access the
Raspberry Pi remotely using VNC?

→ purpose of VNC :-

VNC (Virtual Network Computing) allow remote access to Raspberry Pi graphical interface from another device over network.

It is useful when a monitor is not available.

• How to enable VNC on Raspberry Pi's.

1. open Raspberry Pi configuration
(`sudo raspi-config`)
2. Go to interface 'select VNC and enable it
3. Install Real VNC Viewer on your computer.

Q5. What are the advantage of using headless mode for Raspberry Pi, how can it be configured?

→

Advantage :-

- eliminates the need for an external monitor, key board and mouse
- Save space and power
- Allow remote access via ssh and VNC
- enable automation and IoT applications.

• How to configure headless mode :-

1. Prepare the microSD card with Raspberry Pi OS using Raspberry Pi Imager
2. Enable ssh :-
 - create blank file name ssh in boot partition
3. Enable Wifi
 - create a file name wpa-supplicant.conf in the boot partition.
4. Insert the microSD card into the Raspberry Pi, power it on
5. find Raspberry Pi IP address and connect via ssh or VNC.

MIT SCHOOL OF COMPUTING

Rajbaug, Loni-Kalbhor, Pune

Name : Tanmay Khedekar

class : TY-15 (AIIEC)

Roll No : 2223122

Sub : ECL

ECL Experiment No : 2

DHT11 with Raspberry Pi and web server

Q1. what is the function of the DHT11 Sensor and how does it measure temp and humidity?

→ The DHT11 sensor is used to measure Temperature and Humidity.

- Temperature Measurement - use a thermistor

- Humidity Measurement : use a capacitive humidity sensor.

- Data communication : output digital data.

Via a single-wire communication protocol

Q2. How do you interface the DHT11 sensor with Raspberry Pi using GPIO Pins?

Provide a circuit diagram

→ Interfacing DHT11 with Raspberry Pi using GPIO. To connect DHT11 to Raspi, follow these steps:-

Connections:-

- VCC - 3.3 or 5V (Raspi)

- GND - Ground (Raspi)

- Data - Any GPIO.

circuit Diagram:

Rasp Pi

3.3

GND

GP204

→

→

→

DHT 11

VCC

GND

Data

Q3. What is flask? and how can it be used to create web server on Rasp Pi?

→ Flask is a Lightweight Python web framework that allow you to create web application easily

use on Raspberry Pi -

- Display sensor data on a web page
- Create IoT dashboard for remote monitoring
- Control GPIO pins from browser

Q4 write a basic Python script to read data from DHT11 sensor and display on flask

→ Python script to read DHT11 Data and display on flask web page

Install required Libraries

pip install Adafruit-DHT flask

Python script (app.py)

from flask import flask, render_template

import adafruit_DHT

app = flask(__name__)

DHT_sensor = Adafruit-DHT.DHT11

DHT_PIN = 4

This setup allow real-time monitoring of DHT sensor data from any device on the same


```
def index():  
    humidity, temperature = Adafruit_DHT  
    read(DHT_SENSOR, DHT_PIN)  
    if humidity is not None and temperature is not None:  
        return f"Temperature: {temperature:  
            'C <br> Humidity: {humidity:  
    else: "sensor"
```

```
if __name__ == '__main__':  
    app.run(host='0.0.0.0', port=5000,  
            debug=True)
```

Run the script

Python app.py

Q5. How can you make the Flask Server run accessible from another device in the same network?

→ Access Flask web-server from Another Device in the same network

1. Find Rasp PI's IP Address :-

Hostname -I

output = 192.168.1.100

2. Run Flask with host = '0.0.0.0'

3. Access from another device :-

• open a web browser and get to :-

http://192.168.1.100:5000

• You should see the temperature and humidity data displayed

This setup allow real-time monitoring of DHT sensor data from any device on the same

Name: Tanmay Sandip Rhodekar

Days: 7-15-2025

Roll No: 2223122

EC2 Experiment NO. 3

DH11 on Raspberry Pi and displaying
value on Blynk IOT.

Q1. What is Blynk IOT and How does it facilitate
IoT application development?

→ Blynk IOT is a cloud-based IOT platform
that enables developer to remotely
Monitor and control device via mobile app
Key features:-

- No complex coding
- cross platform
- Real time data
- cloud integration

Q2. Explain the step to set-up a Blynk
project and generate an authentication
token from Rasp Pi?

→ 1. Install Blynk APP

2. Create a new project

- select 'Rasp Pi' as the device
- connection type 'wifi' or ethernet

3. Get the Authentication token:-

- After creating the project, Blynk
email you an auth token

- save this token for use in your Python scripts.

03.

→ Install dependencies.?

- pip install blynklib Adafruit_DHT
Python script

```
import blynklib
import Adafruit_DHT
Blynk_Auth = 'Auth-DHT'
DHT_Sensor = 'Adafruit_DHT.DHT11'
DHT_PIN = 4
blynk = Blynk.Blynk(Blynk_Auth)
@blynk.handle_event('readv1')
def read_temp():
    Humidity, temp = Adafruit_DHT
    read(DHT_Sensor, DHT_PIN)
    if temperature is not None:-
        Blynk.virtual_write(1, temp)

@Blynk.handle_event('readv2')
def read_humidity():
    if humidity is not None:-
        Blynk.virtual_write(2, humidity)

while True:-
    Blynk.run()
```


Q4. How can you configure a Blynk widget to display temp and humidity values to real time

-
- ① open the Blynk App and go to your Project :-
 - ② Add widgets :-
 - Gauge from temperature
 - Gauge for Humidity.
 - ③ configure virtual Pins
 - set Temp widget to V1
 - ④, Save and Run

Q5. What are the advantages of using Rasp Pi over Traditional web server for data visualization?

- Advantage 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

Name: Tarmay Khedekar

class: TY-15 (A3EC)

Sub: EC1

Roll.No: 2223122

EC1 Experiment - 4

Q1. What is MQTT and why is it commonly used in IoT application?

→ MQTT (Messaging queuing Telemetry Transport)

is a lightweight messaging protocol designed for low bandwidth high latency or unreliable network, making it ideal for IoT application like smart homes, industrial automation and remote monitoring.

Q2. Explain the roles of MQTT Broker publisher and subscriber in an IoT system

→ MQTT components:

- Broker: central server that manages message transmission

- Publisher: send data to a topic

- ex a sensor publishing temperature

- Subscriber: Receives message from a topic

Q3. How can you install and set up the Mosquitto MQTT Broker on Rasp Pi?

→ Installing Mosquitto on Raspberry Pi

Sudo apt update

Sudo apt install -y mosquitto mosquitto-clients

Sudo systemctl enable mosquitto

Sudo systemctl start mosquitto

Q4. Write a python script to publish temp data from a DHT11 sensor to a MQTT topic?

→ `import paho.mqtt.client as mqtt`

`import Adafruit_DHT`

`sensor = Adafruit_DHT`

`pin = 4`

`client = mqtt.Client()`

`client.connect("broker.hivemq.com", 1883, 60)`

`while True:-`

`humidity, temp = Adafruit_DHT.read`

`-retry(sensor, pin)`

`if temperature:-`

`client.publish("temp/temp", temp)`

Q5. How can you subscribe to an MQTT topic from another device to receive real-time sensor updates?

→

`import paho.mqtt.client as mqtt.`


```
def on_message(client, userdata, msg):  
    print(f"Received: {msg.topic} → {msg.payload.decode()}")  
client = mqtt.client()  
client.on_message = on_message  
client.connect("broker.hivemq.com",  
              1883, 60)  
client.subscribe('iot/temperature')  
client.loop_forever()
```