PRACTICAL 9

AIM: Raspberry Pi: Remote Access Setup and LED Control via Python Programming.

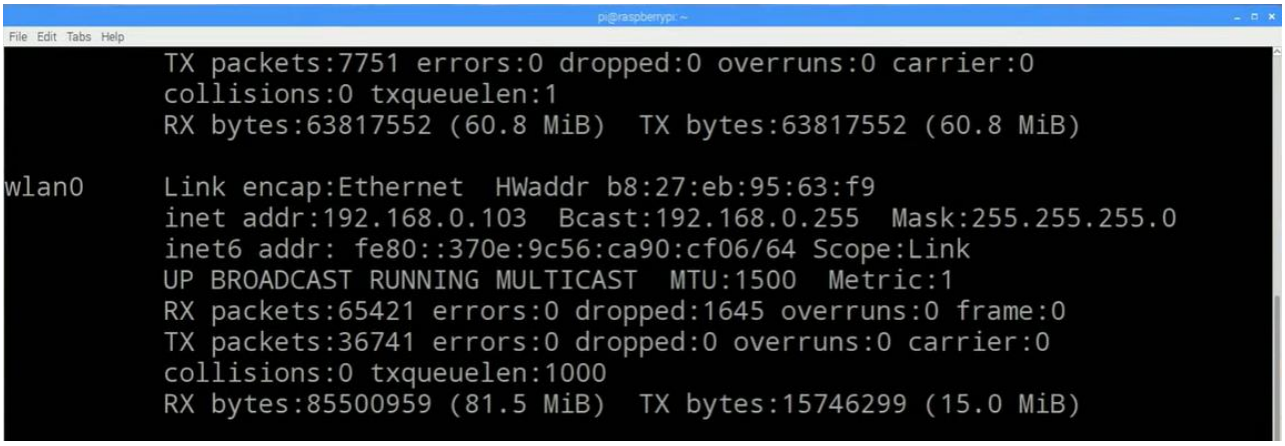
PREREQUISITE: Basics of programming, microcontrollers and basic electronics.

OUTCOME: Access the Raspberry Pi remotely using SSH and VNC, and demonstrate control of its GPIO to blink an LED using Python programming.

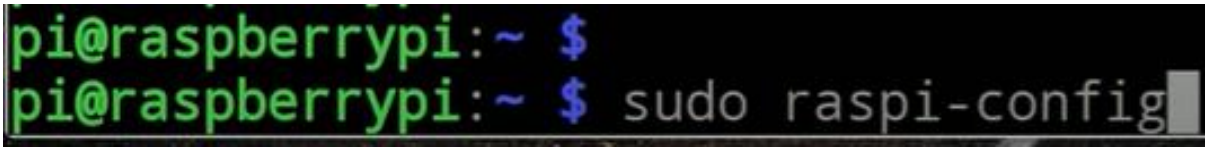
PROCEDURE:

A) For Remote Access:

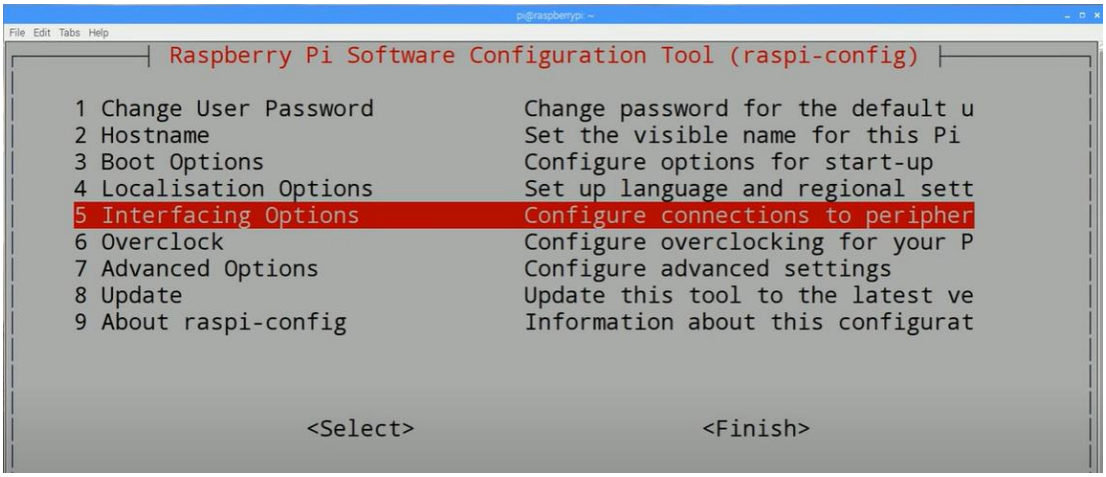
Step 1: Connect the Raspberry Pi to a Wi-Fi network and note its IP address.



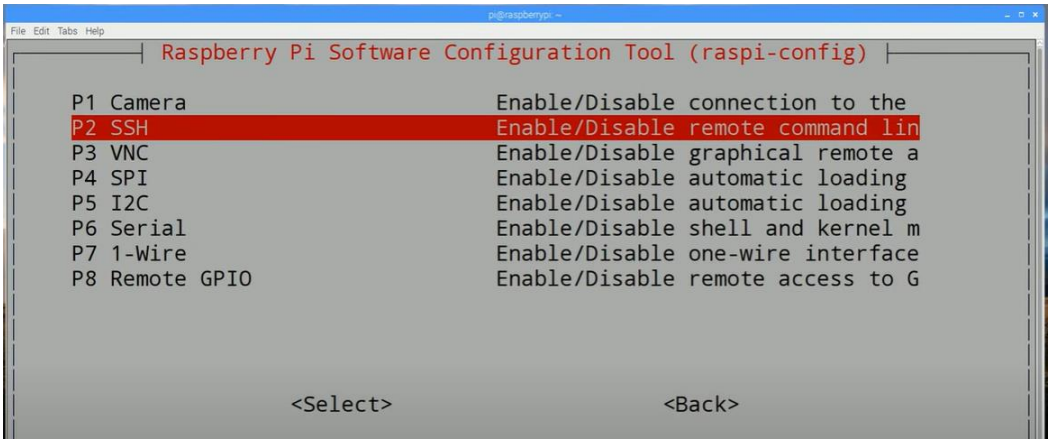
Step 2: Write command: sudo raspi-config

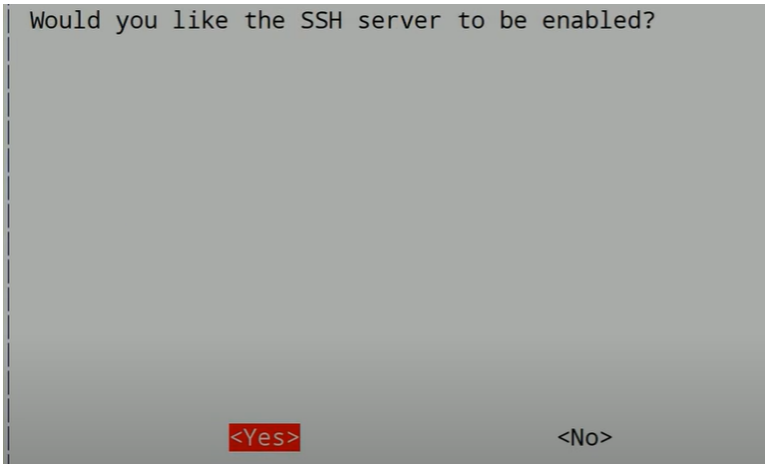


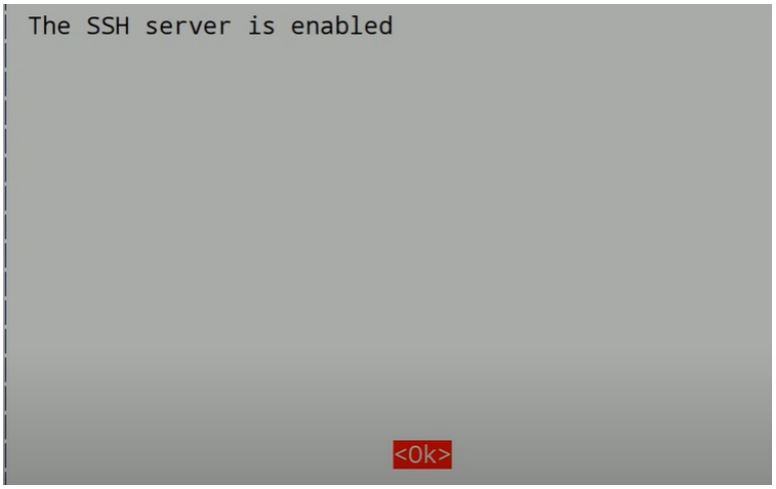
Step 3: Click on Interfacing Options.



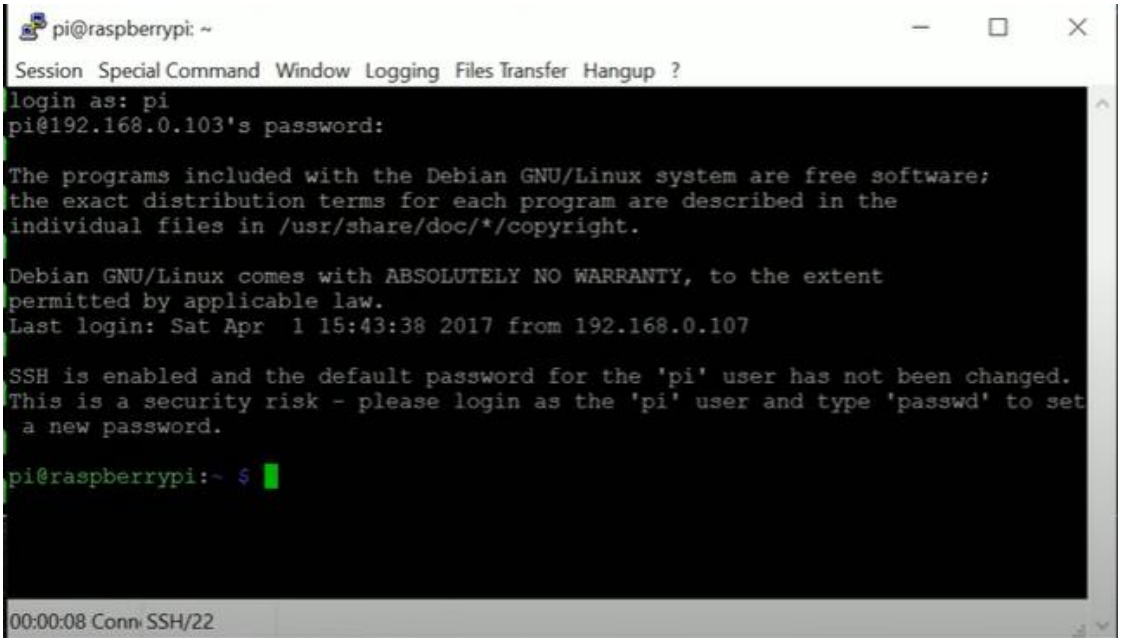
Step 4: Click on P2 SSH



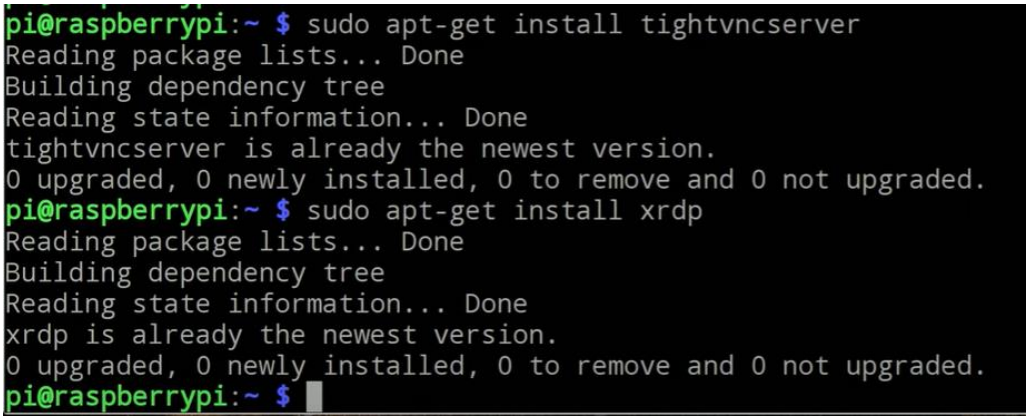




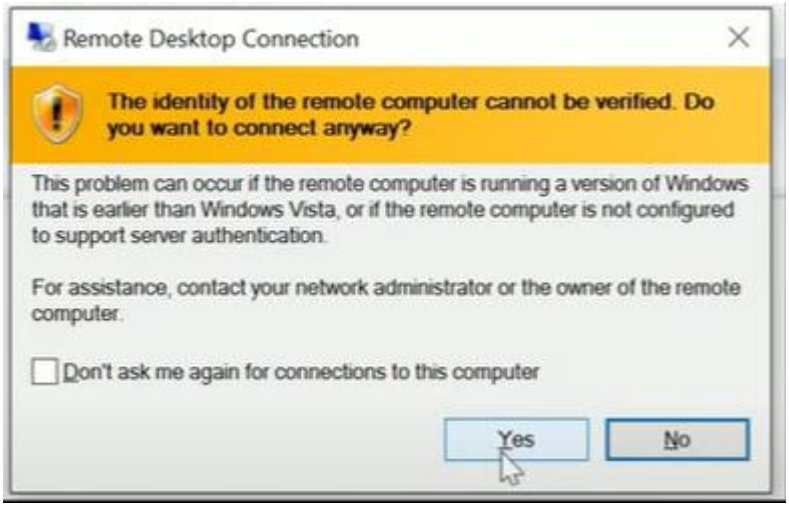
Step 5: Use the SSH client to remotely access the Raspberry Pi by entering its IP address.



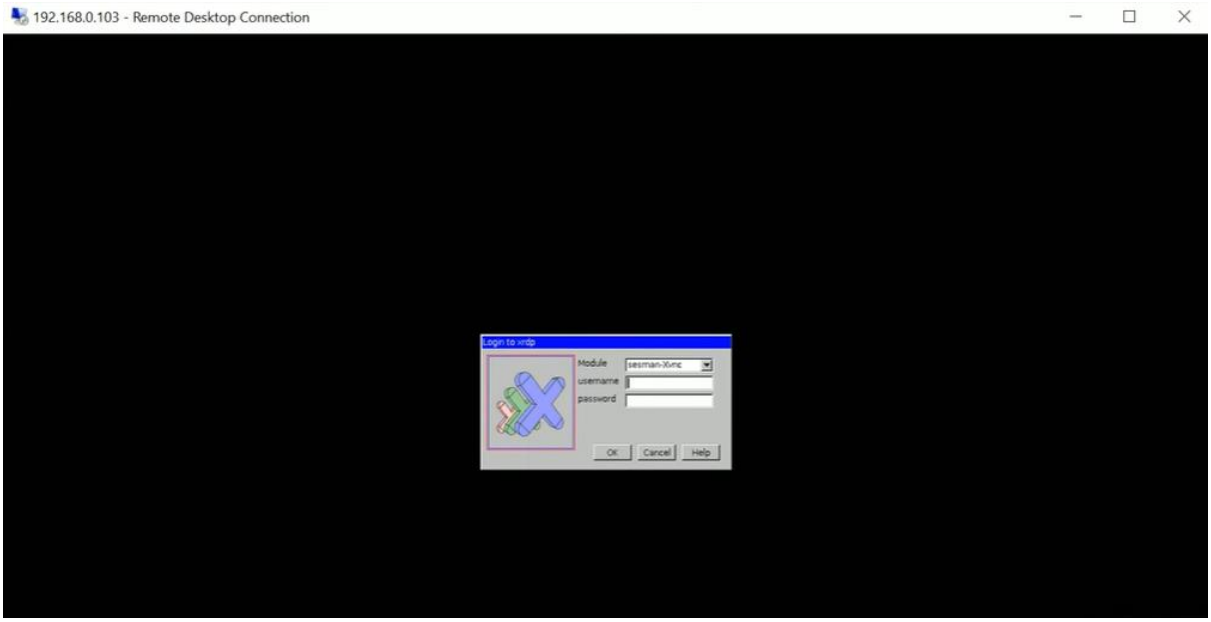
Step 6: Install an SSH client (e.g., PuTTY) and a VNC viewer (e.g., tightvnc Server).



Step 7: Open the VNC viewer, input the IP address of the Raspberry Pi, and log in to access the desktop remotely.



Step 8: Raspberry Pi remote access successful



B) Blink an LED using Python programming.

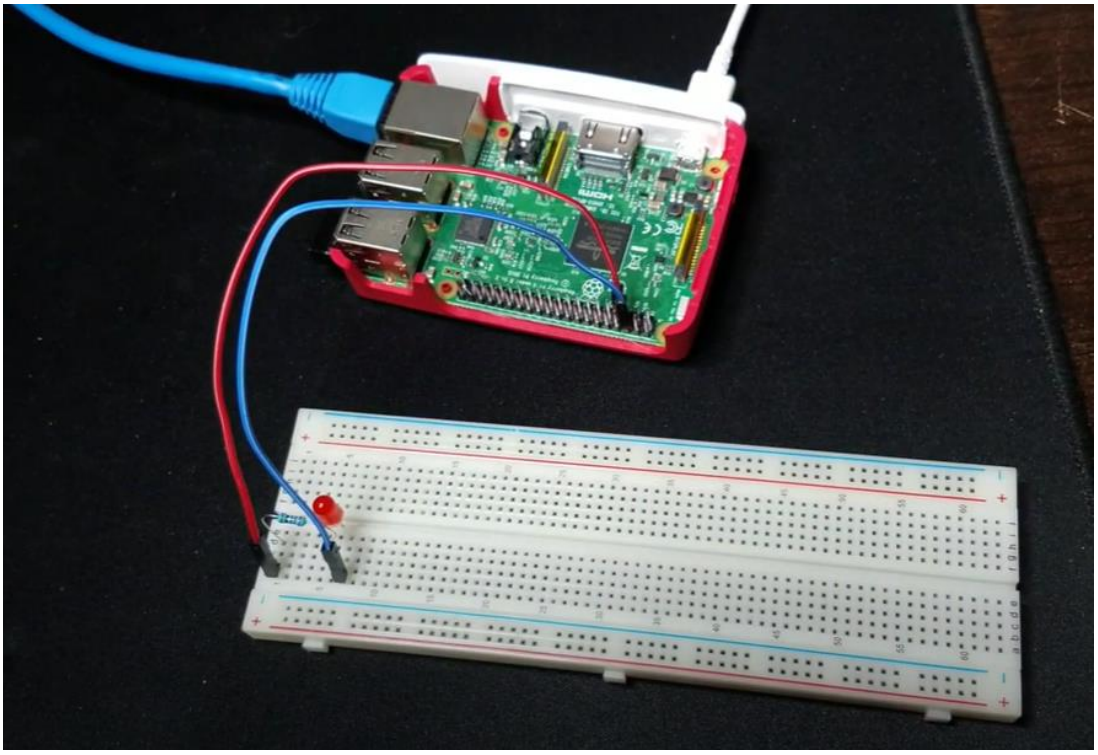
Code:

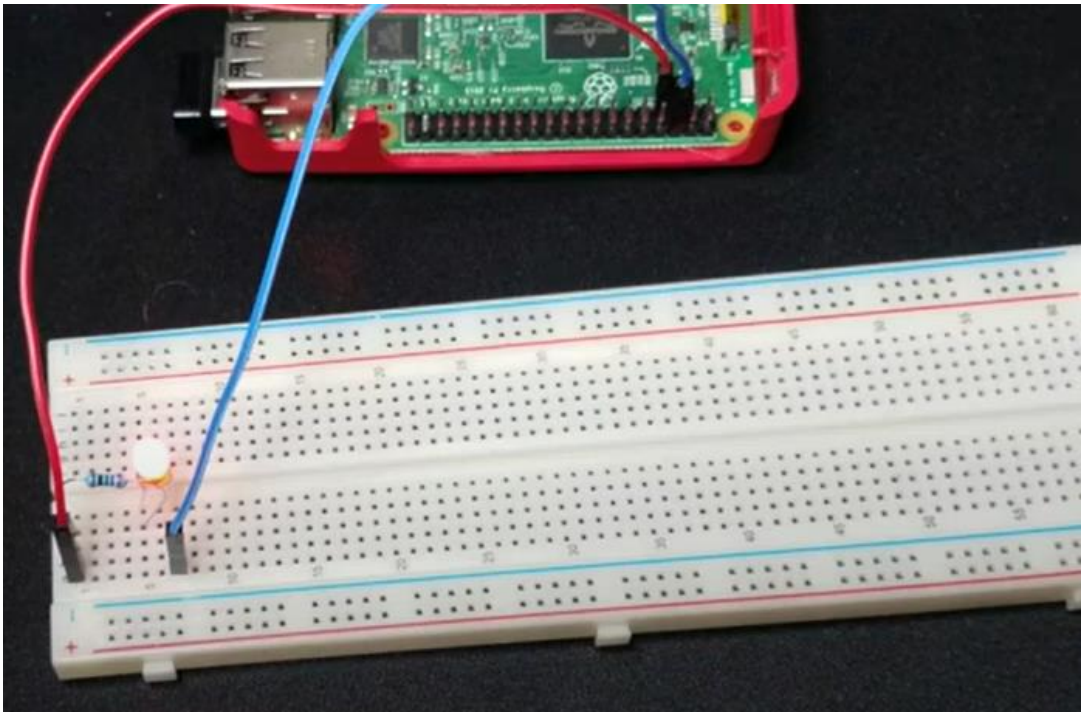
```
import RPi.GPIO as GPIO
import time

# Set up GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
```

```
# Blink LED
try:
    while True:
        GPIO.output(17, GPIO.HIGH)  # LED on
        time.sleep(1)
        GPIO.output(17, GPIO.LOW)   # LED off
        time.sleep(1)
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```

OUTPUT:





Observation

During the experiment, we successfully accessed the Raspberry Pi remotely through SSH and VNC, allowing us to manage it without a physical display. We learned to control GPIO pins using Python programming to blink an LED, gaining hands-on experience with basic circuit connections and Raspberry Pi's GPIO setup. This experiment reinforced our understanding of remote management of IoT devices and the flexibility of Python in hardware control.

Conclusion

In conclusion, remote access to Raspberry Pi through SSH and VNC is highly effective for managing and programming the device without a monitor. This experiment demonstrated how Raspberry Pi GPIO pins could be controlled using Python to interact with hardware components like LEDs. Such setups can be expanded to control more complex hardware, showcasing the Raspberry Pi's suitability for various IoT and automation projects.