**PRACTICAL –8**

**Aim: - Raspberry pi basic programming.**

**Practical 8: Raspberry Pi Programming Fundamentals**

This practical session delves into the captivating realm of Raspberry Pi programming, equipping participants with the foundational knowledge to control electronic components using the GPIO pins. The session outlines essential experiments to illuminate core programming concepts and establish a solid understanding of GPIO interaction. SSH and VNC connection procedures are also presented, facilitating remote access to the Raspberry Pi for enhanced project development.

**1. Raspberry Pi 3 GPIO Header: Demystifying the I/O Landscape**

The Raspberry Pi 3 features a General Purpose Input/Output (GPIO) header, a versatile interface that grants the ability to connect and control various electronic components. These pins can be configured as either inputs (receiving signals from sensors) or outputs (sending signals to control actuators like LEDs). The GPIO header on the Raspberry Pi 3 consists of 40 pins, each with a unique identifier. Understanding pin numbering schemes is crucial for proper GPIO programming.

**2. Experiments in Exploration: Mastering LED Control**

The practical session commences with a series of experiments designed to solidify fundamental programming concepts and GPIO interaction. These experiments all focus on controlling an LED connected to the Raspberry Pi:

* **LED ON:** This introductory experiment establishes a basic understanding of how to write code to turn on an LED connected to a specific GPIO pin.
* **LED Blinking:** Building upon the previous experiment, participants will explore techniques to create a blinking LED effect using code that repeatedly switches the LED on and off at controlled intervals. This introduces concepts like loops and timing functions within the programming language.
* **Brightness Control:** This experiment delves into a more advanced concept - Pulse Width Modulation (PWM). By rapidly switching the LED on and off at varying duty cycles, participants will be able to control the perceived brightness of the LED using code. This demonstrates the ability to manipulate output signals for nuanced control.

**3. Establishing Remote Access: SSH and VNC Connections**

For convenient project development and program execution on the Raspberry Pi, the session incorporates instructions for setting up remote access methods:

* **SSH Connection:** Secure Shell (SSH) offers a secure command-line interface for interacting with the Raspberry Pi from a remote computer. The guide outlines the installation of TeraTerm, a terminal emulator, and the steps involved in establishing an SSH connection using the Raspberry Pi's IP address, username (pi), and password (raspberry by default).
* **VNC Connection:** Virtual Network Computing (VNC) provides a graphical desktop interface for interacting with the Raspberry Pi remotely. The guide covers the installation of a TightVNC viewer and the process of connecting to the Raspberry Pi's graphical desktop using its IP address and display number (usually 1). The default password for VNC access is also provided ("password").

**4. Raspbian OS and the Pixel Desktop Environment**

The Raspberry Pi typically comes pre-installed with Raspbian OS, a Debian-based operating system optimized for the Raspberry Pi's hardware. Raspbian provides a user-friendly environment for development and experimentation. The default desktop environment on Raspbian is PIXEL, offering a graphical interface with familiar elements like menus, windows, and a taskbar.

By successfully completing these experiments and mastering remote access methods, participants will establish a strong foundation for further exploration in Raspberry Pi programming and Internet of Things (IoT) development. The ability to control electronic components using GPIO pins opens a door to creating interactive projects and innovative applications.