# Edge Computing Lab

## Class: TY-AIEC

### School of Computing, MIT Art Design Technology University

#### Academic Year: 2024-25

**Experiment No. 1**

# Title: “Hello World” to Raspberry Pi

# Raspbian OS Installation and Configuration on Raspberry Pi 4 and familiar with Raspberry Pi 4 GPIO’s

## Introduction

This manual provides a comprehensive guide for undergraduate students to understand the OS installation process using the Raspbian OS Imager, and configure remote access tools like PuTTY and VNC Viewer on Raspberry Pi 4.

## Prerequisites

* Raspberry Pi 4
* MicroSD card (minimum 8 GB)
* Raspbian OS Imager software
* PuTTY for SSH access
* VNC Viewer for remote desktop access
* Internet connection
* Additional peripherals (monitor, keyboard, mouse, etc.)

## Experiment Steps

### Part 1: Installing Raspbian OS

Detailed steps on using the Raspbian OS Imager to select the correct version of the OS and write it to the MicroSD card. Illustrations will include screenshots of the imager interface and selection process.

### Part 2: Setting Up Raspberry Pi

Instructions on inserting the MicroSD card into the Raspberry Pi, connecting all necessary peripherals, and completing the initial boot process. This section will include diagrams showing how to connect the Raspberry Pi to various peripherals.

### Part 3: Configuring PuTTY for SSH Access

Step-by-step guide on installing PuTTY, finding the Raspberry Pi's IP address, and establishing an SSH connection. This section will contain figures illustrating the PuTTY configuration settings.

### Part 4: Setting Up VNC Viewer

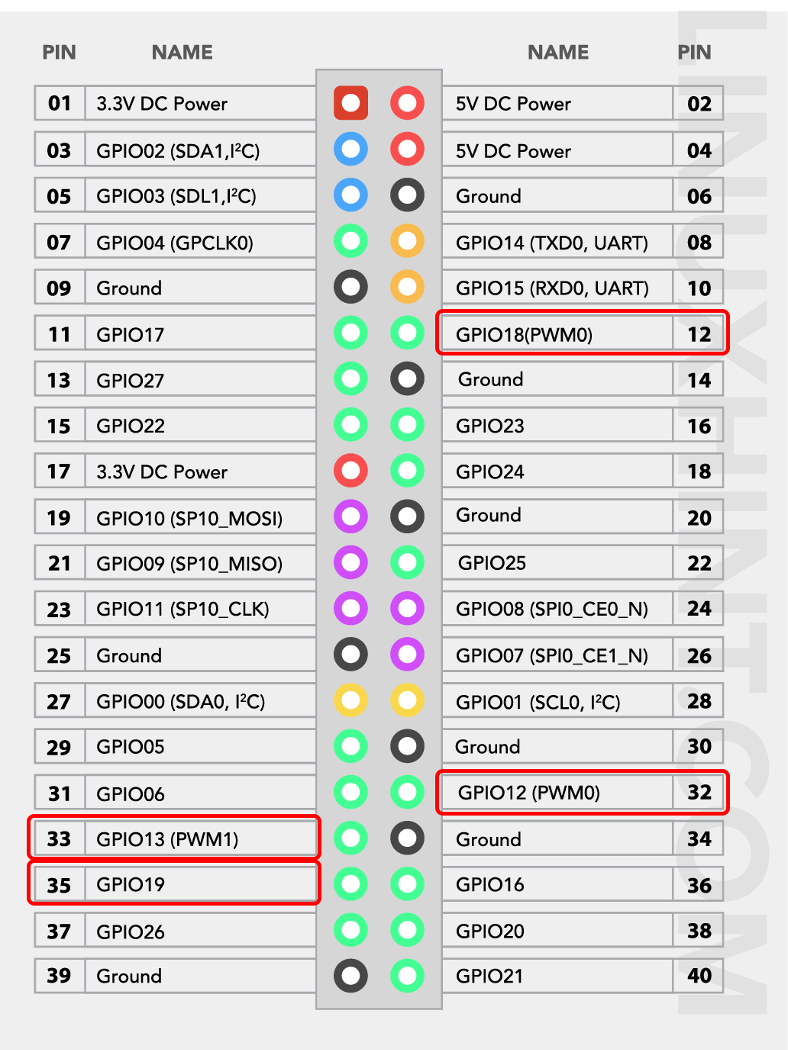
Detailed instructions on enabling VNC on the Raspberry Pi through the terminal or Raspberry Pi configuration settings, and connecting via VNC Viewer. Screenshots will guide the user through the VNC setup and connection process.

### Part 5: Interface the LED with Raspberry Pi 4 on GPIO14 using BCM

Experiments tis section designed to introduce students to the fundamentals of hardware interfacing using the Raspberry Pi 4. By the end of this lab, students will learn how to control an external LED using the GPIO pins of the Raspberry Pi.

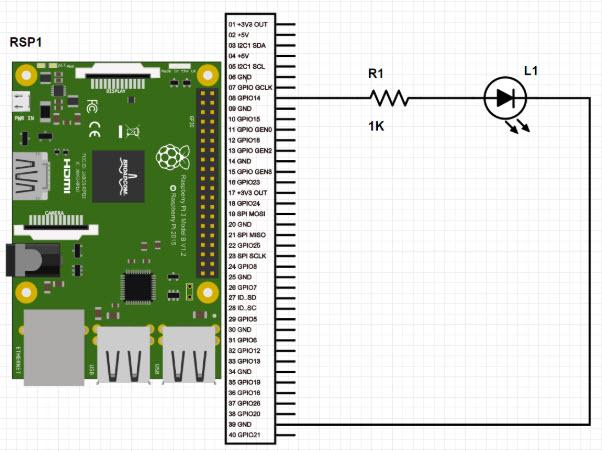
## Theory

GPIO (General Purpose Input/Output) pins on the Raspberry Pi are used for interfacing with other electronic components. BCM numbering refers to the pin numbers in the Broadcom SOC channel, which is a more consistent way to refer to the GPIO pins across different versions of the Raspberry Pi.



## Circuit Diagram

An illustrative diagram showing how to connect the LED to the Raspberry Pi GPIO14 pin through a resistor using a breadboard will be included here.

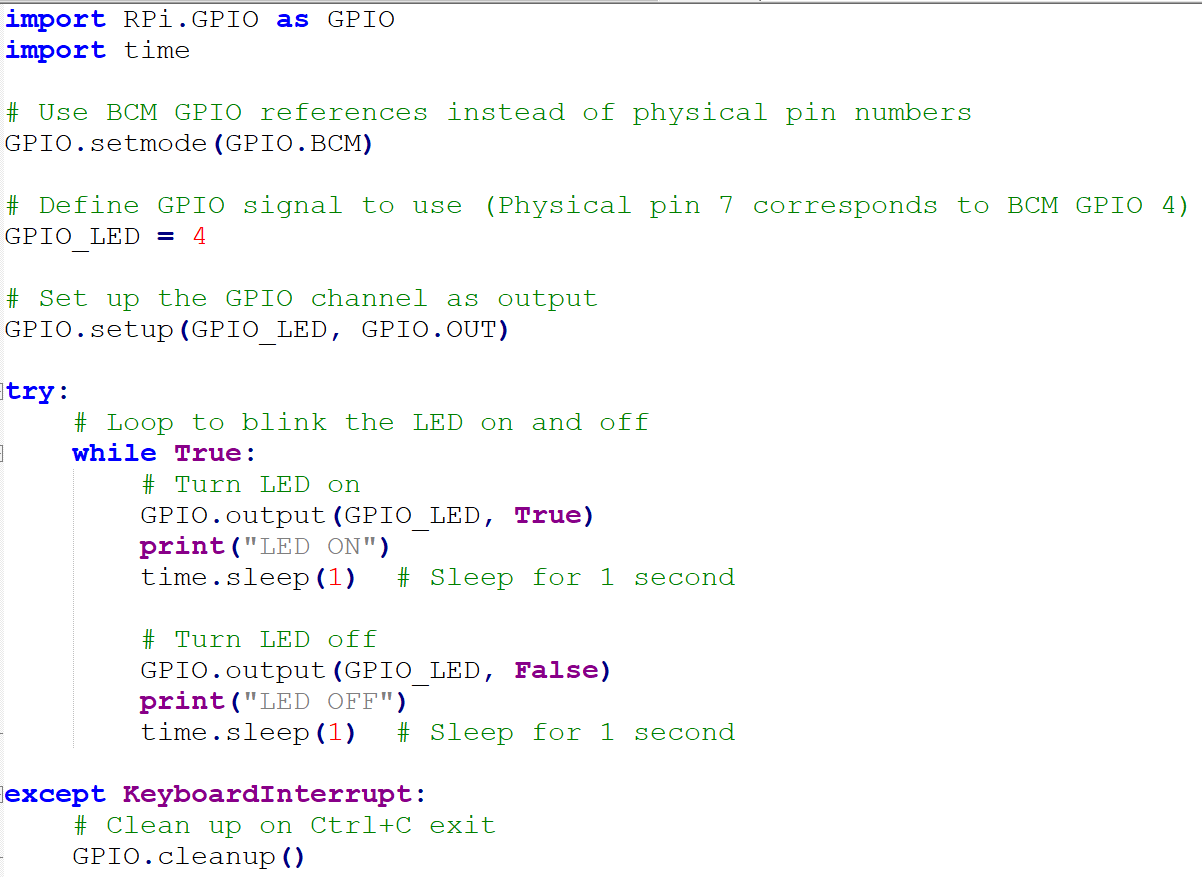


## Experiment Procedure

Detailed steps for setting up the Raspberry Pi, connecting the LED to the GPIO14 pin through the resistor, and securing all connections on the breadboard.

## Python Code

A simple Python script to control the LED by turning it on and off will be provided, demonstrating the use of GPIO library and BCM pin numbering.



## Python Code to display ‘Hello World’ on LCD

import smbus

import time

# Define the I2C address of the LCD

LCD\_ADDRESS = 0x3E

RGB\_ADDRESS = 0x62

# Initialize the I2C bus

bus = smbus.SMBus(1)

def send\_command(cmd):

"""Send a command to the LCD."""

bus.write\_byte\_data(LCD\_ADDRESS, 0x00, cmd)

def send\_data(data):

"""Send data to the LCD."""

bus.write\_byte\_data(LCD\_ADDRESS, 0x40, data)

def set\_rgb(r, g, b):

"""Set the backlight color of the LCD."""

bus.write\_byte\_data(RGB\_ADDRESS, 0x00, 0x00)

bus.write\_byte\_data(RGB\_ADDRESS, 0x01, 0x00)

bus.write\_byte\_data(RGB\_ADDRESS, 0x08, 0xAA)

bus.write\_byte\_data(RGB\_ADDRESS, 0x04, r)

bus.write\_byte\_data(RGB\_ADDRESS, 0x03, g)

bus.write\_byte\_data(RGB\_ADDRESS, 0x02, b)

def initialize\_lcd():

"""Initialize the LCD."""

send\_command(0x01) # Clear display

time.sleep(0.05)

send\_command(0x38) # Function set: 8-bit, 2 lines

send\_command(0x0C) # Display ON, Cursor OFF, Blink OFF

send\_command(0x06) # Entry mode set: increment, no shift

time.sleep(0.05)

def write\_message(line1, line2):

"""Write a message to the LCD."""

send\_command(0x80) # Move cursor to the beginning of the first line

for char in line1:

send\_data(ord(char)) # Send each character as data to the LCD

send\_command(0xC0) # Move cursor to the beginning of the second line

for char in line2:

send\_data(ord(char)) # Send each character as data to the LCD

if \_\_name\_\_ == "\_\_main\_\_":

try:

initialize\_lcd()

set\_rgb(0, 255, 255) # Set backlight to greenish color

write\_message("Hello", "World") # Display message

while True:

time.sleep(1)

except KeyboardInterrupt:

send\_command(0x01) # Clear display

set\_rgb(0, 0, 0) # Turn off backlight

print("Exiting...")

## Verification and Testing

Steps to verify the successful installation of the Raspbian OS, and the functionality of SSH and VNC connections, including sample commands and expected outputs.

## Troubleshooting Tips

Common issues and their solutions related to OS installation, SSH, and VNC connections, supported by troubleshooting flowcharts and diagrams.

## Observation

Students are expected to observe the LED turning on and off in response to the Python script, demonstrating the successful interfacing and control of external hardware with the Raspberry Pi

## Conclusion

A summary of the key learning points from the manual and encouragement for students to explore further applications and configurations of the Raspberry Pi 4.