Practical 2

**Aim:-** To control an LED using Raspberry Pi’s GPIO pins and Python programming.

**Components Required:**

1.Raspberry Pi board (any model with GPIO support)

2.LED (Light Emitting Diode)

3. 330Ω Resistor

4.Breadboard

5.Jumper wires

**Steps for Connection:**

1. **Identify GPIO Pins:** Find the GPIO pin layout of your Raspberry Pi (e.g., GPIO17).
2. **Connect the LED:**
   * Connect the **long leg (anode, +)** of the LED to a **GPIO pin** (e.g., GPIO17) via a **330Ω resistor**.
   * Connect the **short leg (cathode, -)** to the **GND (Ground) pin** of Raspberry Pi.
3. **Check Connections:** Ensure the LED and resistor are securely placed on the breadboard.

**Code:**

#Blink LED Program

#Connect the LED to GPIO 22 Pin

#LED Blink Progarm

#Connect the LED to GPIO22 (i.e. Physical Pin15)

#import GPIO and time library

import RPi.GPIO as GPIO

from time import sleep

GPIO.setmode(GPIO.BCM)

ledPin = 22

GPIO.setup(ledPin, GPIO.OUT)

GPIO.output(ledPin, False)

try:

while True:

GPIO.output(ledPin, True)

#Set the LED Pin to HIGH

print("LED ON")

sleep(1)

GPIO.output(ledPin, False) #Set the LED Pin to LOW

print("LED OFF")

sleep(1)

finally:

#reset the GPIO Pins

GPIO.output(ledPin, False)

GPIO.cleanup()

Practical 3

**Aim:**To connect a Raspberry Pi Camera Module and capture images using Python.

**Components Required:**

1.Raspberry Pi board (with camera support)

2.Raspberry Pi Camera Module (or USB webcam)

3.Camera ribbon cable (for Pi Camera)

4.Jumper wires (if using an external camera)

5.Monitor, keyboard, and mouse (for setup)

**Steps for Connection:**

**1. Connect the Camera to Raspberry Pi:**

* If using the **Raspberry Pi Camera Module**:
  + Locate the **CSI (Camera Serial Interface) port** on Raspberry Pi.
  + Insert the **ribbon cable** (blue side facing the Ethernet port).
* If using a **USB Webcam**:
  + Simply plug it into a **USB port** on the Raspberry Pi.

**2. Enable the Camera (for Pi Camera Module):**

* Open the terminal and run:

sudo raspi-config

* Navigate to **Interfacing Options > Camera > Enable**.
* Reboot the Raspberry Pi:

sudo reboot

**Code:**

from time import sleep

from picamera import PiCamera

camera = PiCamera()

camera.resolution=(1280,720)

camera.start\_preview()

#camera warm-up time

sleep(2)

camera.capture('/home/pi/Desktop/image.jpg')

camera.stop\_preview()

Practical 4

**Aim:**

To display the current time (HH:MM) on a **4-digit 7-segment display** using Raspberry Pi and Python.

**Components Required:**

1. Raspberry Pi (any model with GPIO)
2. 4-digit 7-segment display (Common Cathode/Anode)
3. 4x 220Ω resistors (for current limiting)
4. Jumper wires
5. Breadboard
6. TM1637-based 4-digit 7-segment display

**Steps for Connection:**

Hook up circuit as follow

1. Connect the pin2(5v) of Rpi to vcc pin of module
2. Connect the pin 6 (ground) of Rpi to GND of module
3. Connect the pin 38(GPIO20) of Rpi to DIO(data) of module
4. Lastly connect the pin40(GPIO22) of Rpi to CLK(clock) of module

**Code:**

from time import sleep

import tm1637

try:

import thread

except ImportError:

import\_thread as thread

Display = tm1637.TM1637(CLK=21, DIO-20, brightness=1.0)

try:

print ("Starting clock in the background (press CTRL + C to stop):")

Display StartClock(military\_time=True)

Display SetBrightness(1.0)

while True:

Display ShowDoublepoint(True)

sleep(1)

Display ShowDoublepoint(False)

sleep(1)

Display. StopClock()

thread.interrupt\_main()

except KeyboardInterrupt:

print ("Properly closing the clock and open GPIO pins")

Display.cleanup()