Practical no.

Aim- Starting Raspbian OS, familiarizing with Raspberry Pi Components and interfacing, Connecting to ethernet, Monitor, USB

Requirements -

Hardware Requirements - 1] Raspberry Pi 3B + model
2] MicroSD card
3] Power Supply
4] HDMI cable
5] Monitor
6] USB Keyboard and Mouse
7] Ethernet cable (optional)

Software Requirements - 1] Raspberry Pi Os (Raspbian)
~~2] Raspberry Pi~~
2] NOOBS (New Out of Box Software)

Circuit Diagram

Procedure -

Step 1 - Format the SD card

Step 2 - Download NOOBS from the official Raspberry Pi website

Step 3 - Extract the zip folder to access the files

Step 4 - Copy all the extracted files directly to the root directory of the formatted SD card

Step 5 - Safely Eject the SD card

Step 6 - Insert SD card into Raspberry Pi

Step 7 - Connect Monitor, keyboard and mouse using HDMI and USB cable

Step 8 - Power up Raspberry Pi

Step 9 - In the NOOBS interface, choose Raspbian and Install the OS

Step 10 - On first boot, Raspbian will guide you through an initial setup of language, timezone, etc.

Step 11 - Finish

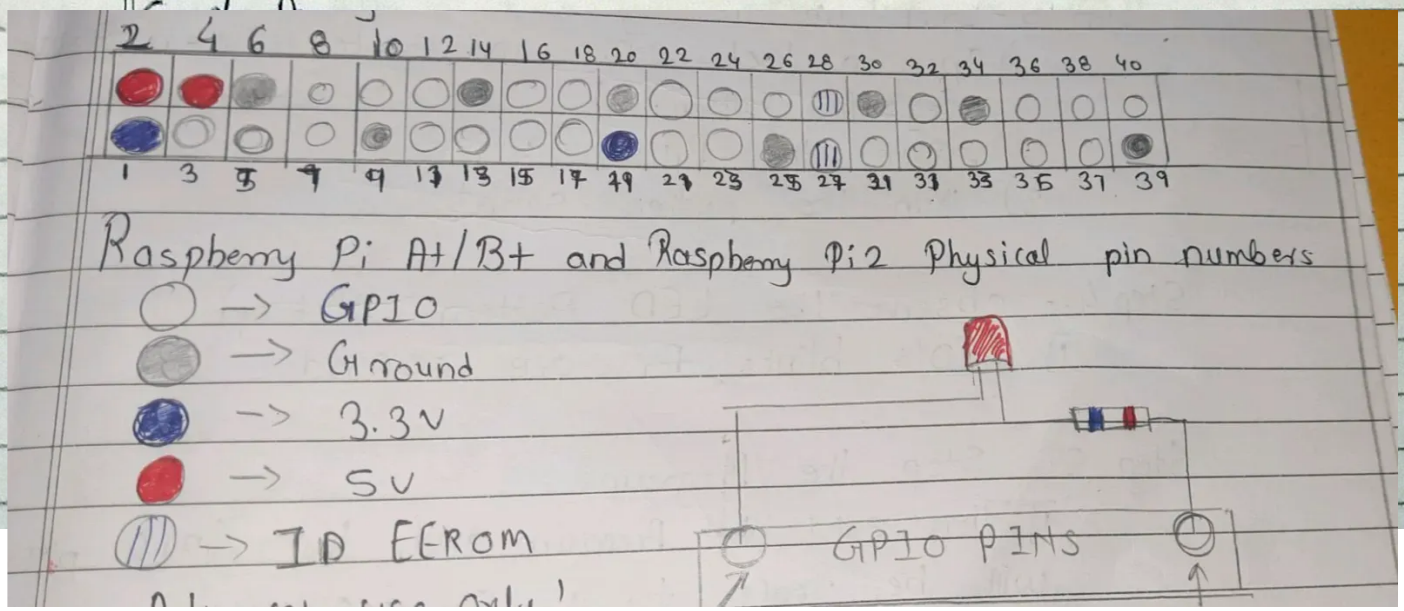
Practical no. 2

Aim- Displaying different LED patterns with Raspberry Pi

Requirements -

Software Requirements - 1] Raspberry Pi OS (Raspbian)
2] Python3
3] RPi.GPIO library for Controlling GPIO Pins

Hardware Requirements - 1] Raspberry Pi 3B+ model
2] LED
3] Resistor - 220Ω to 330Ω
4] Breadboard
5] Jumper wires
6] Power Supply



Procedure -

Step 1- Hardware Setup
1] Place the LED on the Breadboard
2] Connect anode to a separate GPIO pin on the Raspberry Pi using a Resistor.
3] Connect cathode of LED to a ground pin on the Raspberry Pi

Step 2- Install Required Software
1] Boot your Raspberry Pi and open a terminal
2] Update the system and install the necessary libraries:
- sudo apt update
- sudo apt install python3-pip
- sudo pip3 install RPi.GPIO

Step 3- Set up the Python Script
1] Open a text editor and create a Python script (eg- led_pattern.py)
2] Save the Python script
3] Run the Python script

Step 4- Observe the LED Pattern (Blinking)
1] LED's blinks for one second

Step 5- Stop the Program
1] Then exist the Program and the GPIO pins will be reset to a safe state.

Code -

led_pattern.py

```
import RPi.GPIO as IO
import time

IO.setup(IO.BOARD)
IO.setup(40, IO.OUT)
IO.output(40, 1)
time.sleep(1)
IO.cleanup()
time.sleep(1)

# loop is executed second time
IO.setmode(IO.BOARD)
IO.setup(40, IO.OUT)
IO.output(40, 1)
time.sleep(1)
IO.cleanup()
time.sleep(1)

# loop is executed third time

IO.setmode(IO.BOARD)
IO.setup(40, IO.OUT)
IO.output(40,1)
time.sleep(1)
```

Practical no. 3

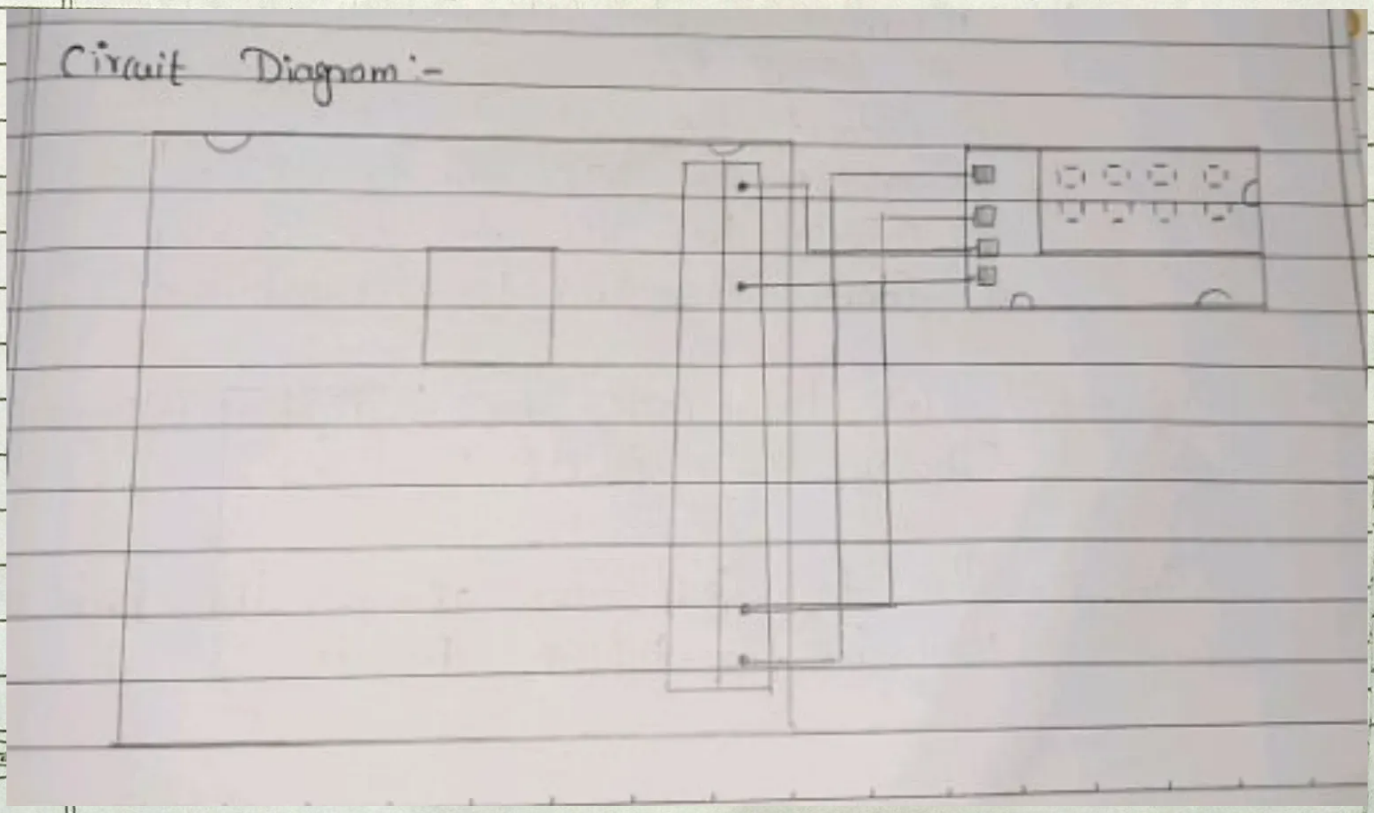
Aim - Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi

Requirements -

Software Requirements -
1] Raspberry Pi OS (Raspbian)
2] Python3
3] RPi.GPIO library
4] tm1637 library
5] datetime module

Hardware Requirements -
1] Raspberry Pi 3B+ model
2] 4digit 7-Segment Display
3] Jumper wires
4] Power Supply

Circuit Diagram :-



Procedure -

Step 1 - Update your Raspbian
$ sudo apt update
$ sudo apt upgrade

Step 2 - Config the Raspberry Pi
$ sudo raspi-config

Step 3 - Change To your Local TimeZone.
Tab to Finish

Step 4 - Reboot the OS

Step 5 - Hardware setup
1] Connect 4-digit 7-segment display to Raspberry Pi GPIO Pins
a] Connect the CLK to GPIO 21
b] Connect the DTO pin to GPIO 20
c] Connect the VCC pin to 3.3 V
d] Connect the gnd to ground pin

Step 6 - Write the code for 4-digit 7Segment in Thonny Python IDE

Step 7 - Run the Code and observe the time which will be in HH:MM format

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 Background (press Ctrl + C
                                    to Stop):")
    Display.StartClock(military_time=False)
    print("Continue Python-Script and Tweak Display!")
    sleep(5)
    Display.showDoublepoint(False)
    sleep(5)
    loops = 3
    while loops > 0:
        for i in range(0,10):
            Display.SetBrightness(i/100)
            sleep(0.5)
        loops -= 1
    Display.StopClocks()
    thread.interrupt_main()
```

```
except KeyboardInterrupt:
    print("Properly closing the clock and open
          GPIO pins:")
    Display.cleanup()
```

Practical no.

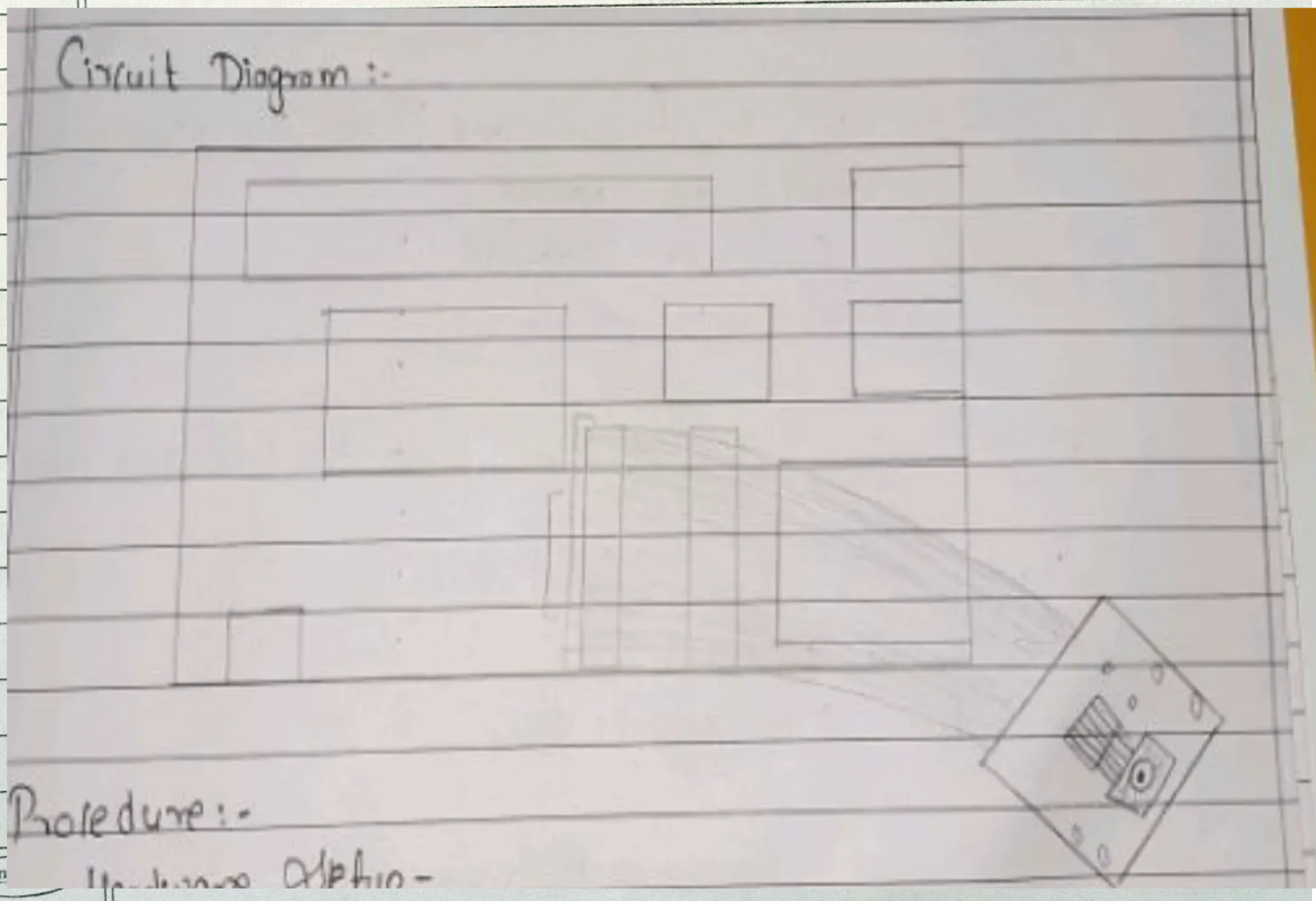
Aim- Pi Camera Module Interface with Raspberry Pi using Python

Requirements-

Hardware Requirements - 1] Raspberry Pi 3B+ model
2] Pi Camera module
3] Ribbon cable (for camera connection)

Software Requirements - 1] Raspberry Pi OS (Raspbian)
2] Python 3
3] picamera library for camera control

Circuit Diagram :-



Procedure :-

Hardware Setup -

Procedure-

Hardware Setup-

Step 1- Locate the camera port on the Raspberry Pi board

Step 2- Insert the ribbon cable from the camera module into the camera port

Software setup-

Step 1- Open the terminal on the Raspberry Pi

Step 2- Run the command -
sudo raspi-config

Step 3- Select Interfacing options > Camera > Enable

Step 4- Restart the Raspberry Pi to apply changes

Step 5- Open the terminal and install the picamera library-
sudo apt update
sudo apt upgrade
python3-picamera

Step 6- Open the text-editor, write the code and run it.

Code-

```
import picamera
from time import sleep

camera = picamera.PiCamera()

camera.resolution = (1024, 768)
camera.brightness = 60
camera.start_preview()

camera.annotate_text = 'Hi Pi User'
sleep(5)

camera.capture('image1.jpeg')
camera.stop_preview()
```

Practical no.

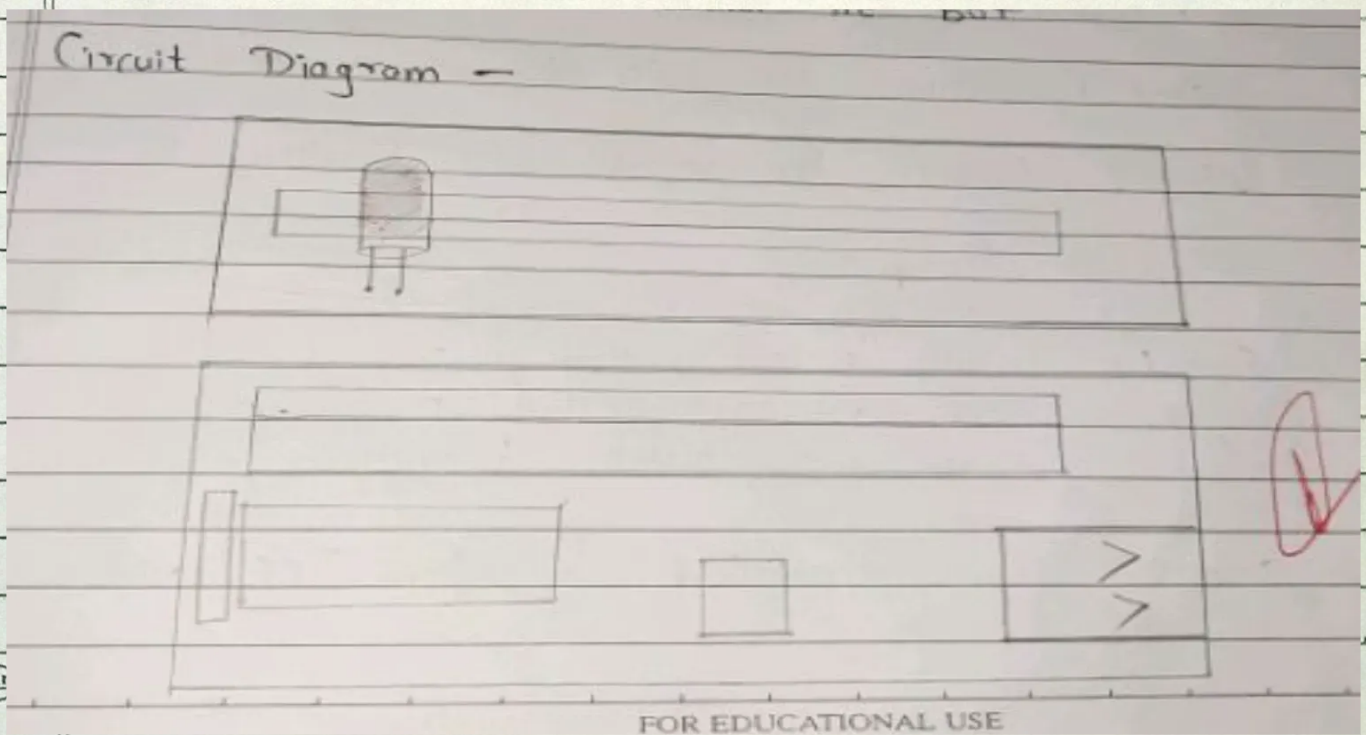
Aim- Controlling Raspberry Pi with telegram

Requirements -

Hardware Requirements - 1] Raspberry Pi 3B+ model
2] LED
3] Resistor (220Ω).
4] Jumper wires
5] Breadboard (if you want use)

Software Requirements - 1] Raspberry Pi OS (Raspbian)
2] Python3
3] Telegram app
4] python-telegram-bot libraries to control the bot.

Circuit Diagram -



Procedure -

Set up a Telegram Bot

Step 1 - Open Telegram app

Step 2 - Search for BotFather

Step 3 - create a new Bot by sending the /newBot command

Step 4 - Name your bot and create username for it

Step 5 - After creation a token will be generated. Save that token.

Hardware set up -

Step 1 - Connect the LED to a resistor

Step 2 - Connect the resistor to Raspberry Pi

Software setup -

Step 1 - Install necessary packages -
sudo apt-get install python-pip
sudo pip install telepot

Step 2 - Write the code in Python IDE and run it

Step 3 - Send the commands from your telegram and observe the LED.

Code -

```
import sys
import time
import telepot
import RPi.GPIO as GPIO

def on(pin):
    GPIO.output(pin, GPIO.HIGH)
    return

def off(pin):
    GPIO.output(pin, GPIO.LOW)
    return

GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)

def handle(msg):
    chat_id = msg['chat']['id']
    command = msg['text']
    print('Got command: %s' % command)
    if command == 'On':
```

```
            bot.sendMessage(chat_id, on(11))
        elif command == 'off':
            bot.sendMessage(chat_id, off(11))

bot = telepot.Bot('your bot token')

bot.message_loop(handle)
print('I am listening')

while 1:
    try:
        time.sleep(10)
    except KeyboardInterrupt:
        print('\n Program interrupted')
        GPIO.cleanup()
        exit()
    except:
        print('other error or exception occured')
        GPIO.cleanup()
```