SAMSUNG

IoT Course

# Exercise

# SIC HCMUTE IOT COURSE

**26/07/2025**


## Group 10

Trần Minh Nhật-UTE_IOT01

Đỗ Trần Quang Hà-UTE_IOT01

Trịnh Minh Trí-UTE_IOT01

Thượng Trí Tín-UTE_IOT01

# Figure Index

# Content

# Chapter 2

**Exercise 1.** **(Conduct the Exercise after Unit 5)**

Write a program that turns on 3 LEDs of different colors one by one each time a button is pressed by using GPIO Zero. Suppose that there are Green, Amber, and Red LEDs, and the initial state is Green. Whenever the button is pressed, the next LED is turned o n, and the existing LED is turned off.

EX6_SIC.py | EX2_SIC_LED.py *

```
1  from gpiozero import LED, Button
2  from time import sleep
3
4  Leds =[LED(20), LED(21), LED(16)]
5  button  = Button(12)
6  current_led = 0
7
8  def next_led():
9      global current_led
10     for led in Leds:
11         led.off()
12     current_led = (current_led + 1 ) % len(Leds)
13     Leds[current_led].on()
14     print(f"LED {current_led} (GPIO {Leds[current_led].pin}) dang sang")
15 try:
16     Leds[0].on()
17     button.when_pressed = next_led
18     while True:
19         sleep(1)
20 except KeyboardInterrupt:
21     print("\n Het")
22 finally:
23     for led in Leds:
24         led.off()
25         led.close()
26     button.close()
```

*Figure 1: The program cycles through LEDs (light-emitting diodes) using a button on a Raspberry Pi.*

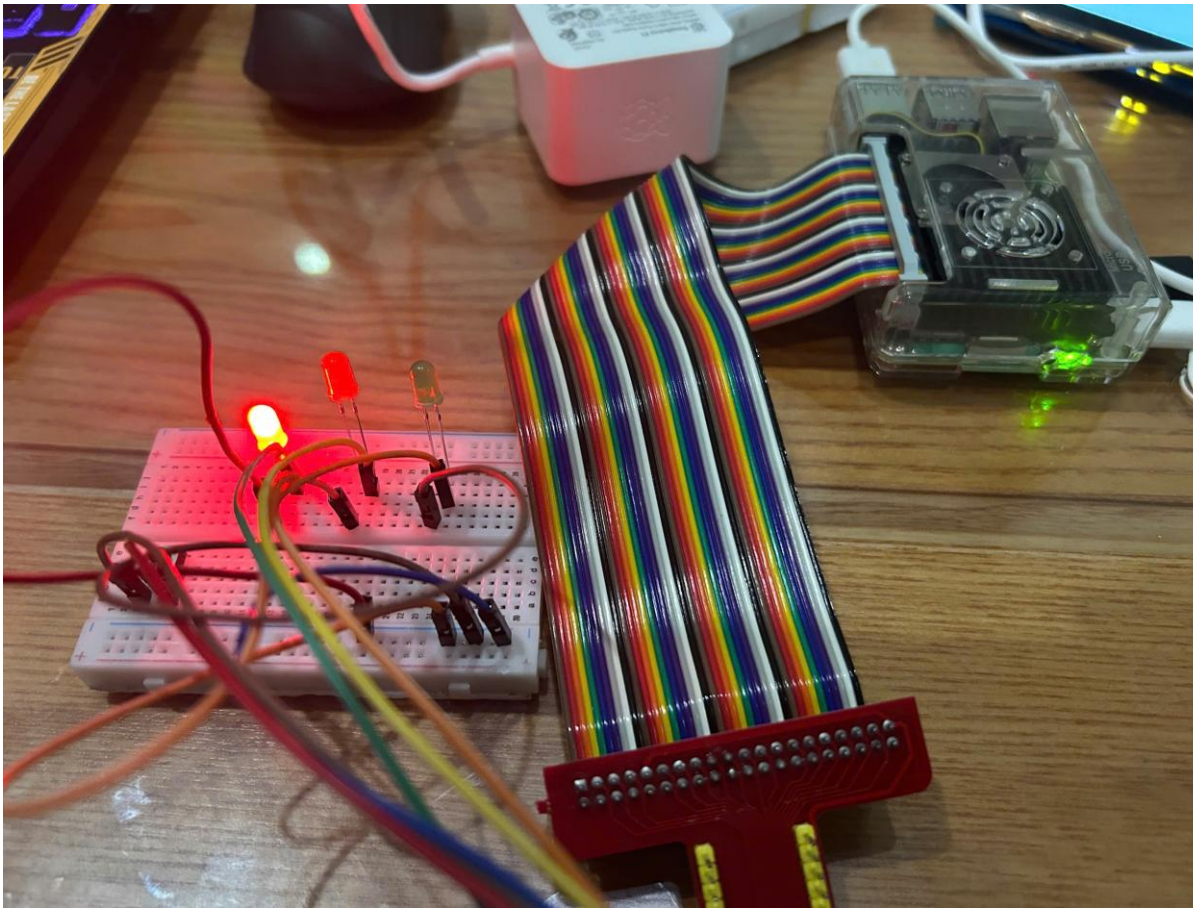

*Figure 2: When the first button is pressed to start the program, the red LED lights up.*

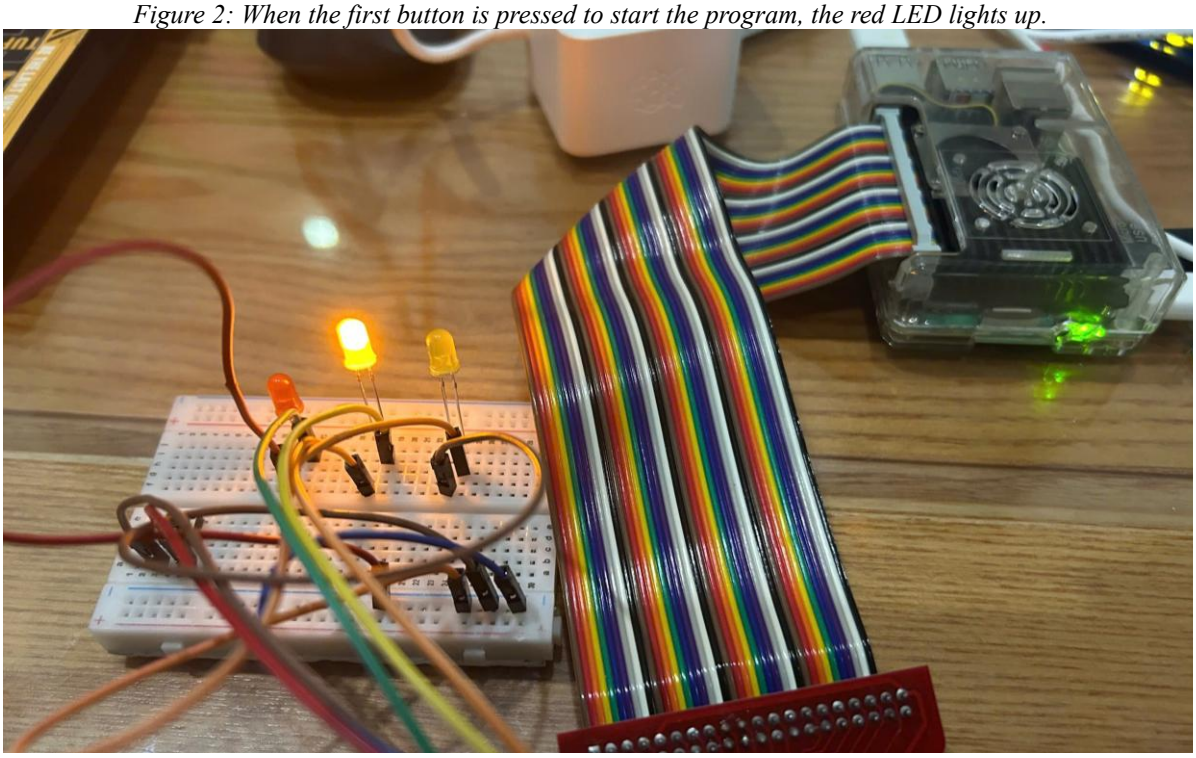

*Figure 3: When the second button is pressed after the program starts, the yellow LED lights up.*

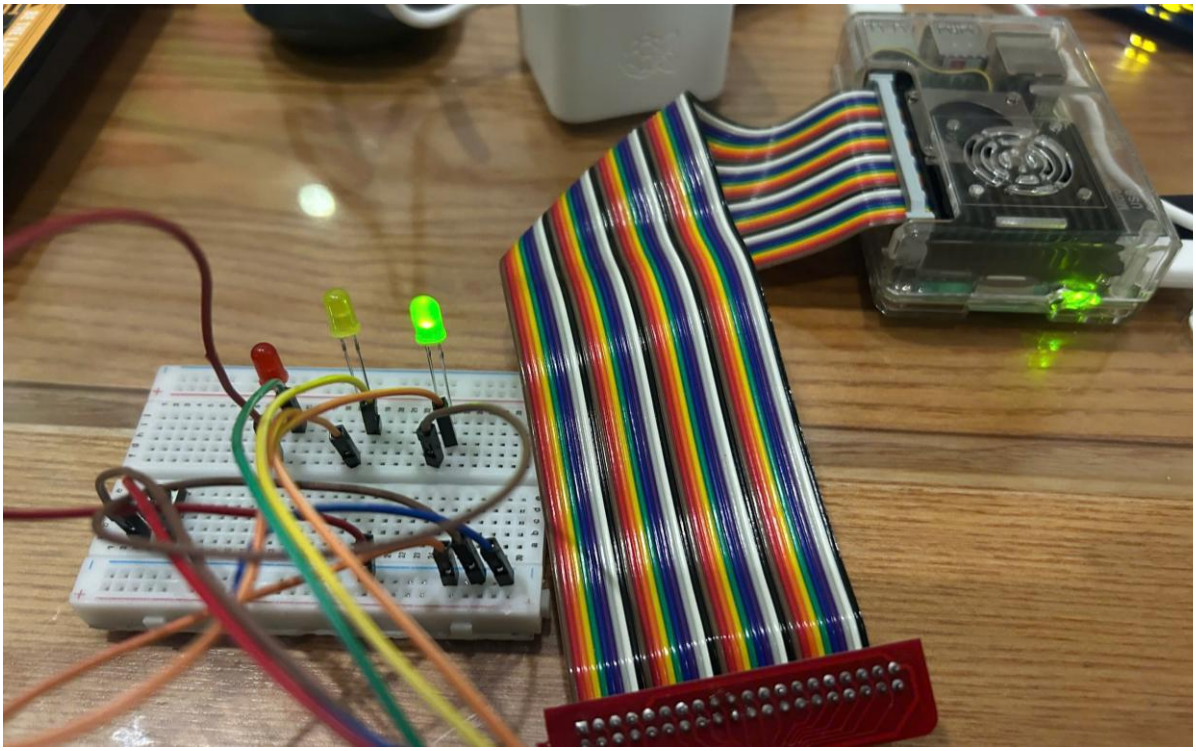

*Figure 4: When the third button is pressed after the program starts, the green LED lights up.*

**Exercise 2. (Conduct the Exercise after Unit 5)**

Write a program that turns on 3 LEDs of different colors one by one each time a button is pressed by using GPIO Zero. Suppose that there are Green, Amber, and Red LEDs, and the initial state is Green. Whenever the button is pressed, the next LED is turned o n, and the existing LED is turned off.

EX4_SIC_2.py | EX2_SIC_LED.py | EX2_SIC_LED_2.py *

```python
from gpiozero import LED, DistanceSensor
from time import sleep


sensor= DistanceSensor(echo=24, trigger=23)
Leds = [LED(21), LED(20), LED(16)]
Color = ["Green", "Yellow", "Red"]
def distance():
    distance= sensor.distance*100
    return distance

def Led_off():
    for led in Leds:
        led.off()
try:
    print("Start the project")

    while True:
        kc= distance()
        Led_off()
        if (kc > 80):
            Led_off()
            Leds[0].on()
            print(f"Distance: {kc} - Led {Color[0]} dang sang")
        elif (kc < 20):
            Led_off()
            Leds[2].on()
            print(f"Distance: {kc} - Led {Color[2]} dang sang")
        else:
            Led_off()
            Leds[1].on()
            print(f"Distance: {kc} - Led {Color[1]} dang sang")
        sleep(1)

except KeyboardInterrupt:
    print("\n End projet")
finally:
    for led in Leds:
        led.off()
        led.close()
    sensor.close()
```

*Figure 5: The LED control program based on distance measured by an ultrasonic sensor.*

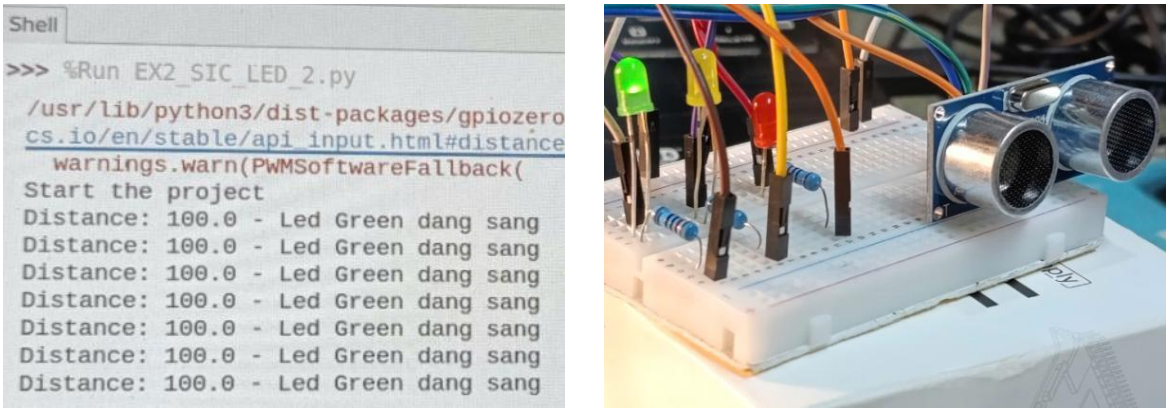


*Figure 6: When the distance is over 80, the green LED will light up.*

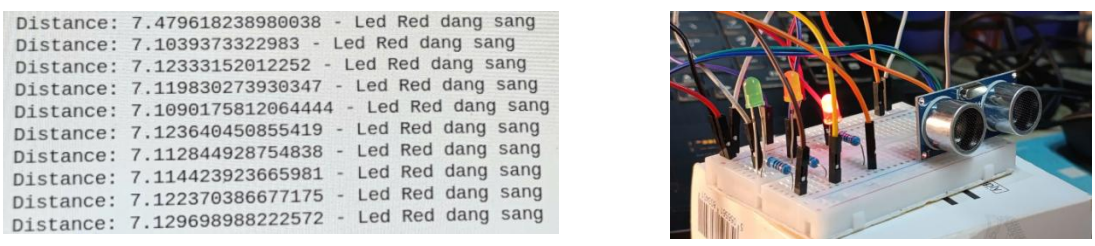


*Figure 7: When the distance is under 20, the red LED will light up.*

```
Distance: 37.66317372067488 - Led Yellow dang sang
Distance: 21.516188993658943 - Led Yellow dang sang
Distance: 75.89051241096831 - Led Yellow dang sang
Distance: 62.41874166580191 - Led Yellow dang sang
Distance: 22.911798338462177 - Led Yellow dang sang
Distance: 22.911798338462177 - Led Yellow dang sang
Distance: 22.912742309089026 - Led Yellow dang sang
Distance: 22.06380883454267 - Led Yellow dang sang
Distance: 22.498599112505417 - Led Yellow dang sang
Distance: 22.496711186861376 - Led Yellow dang sang
Distance: 22.07366039382123 - Led Yellow dang sang
Distance: 22.07304253235543 - Led Yellow dang sang
```

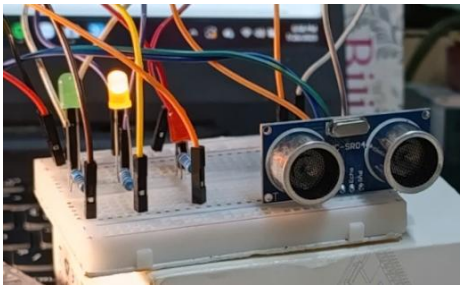

*Figure 8: When the distance is over 20 and under 80, the yellow LED will light up.*

*Figure 9: Our group picture when doing exercise 2*

# Chapter 4

**Exercise 1.** **(Conduct the Exercise after Unit 1)**

Write a program that monitors the total storage space usage of the Raspberry Pi. When the disk usage is over 60%, the red LED is turned on. When the disk usage is over 30 %, the yellow LED is turned on, and the disk usage is saved as a log in /home/pi/Unit_ Practice/disk_usage_log.txt. For reference, the disk usage rate can be obtained by usin g psutil.disk_usage('/').percent.

EX6_SIC.py | EX2_SIC_LED.py * | EX4_SIC_1.py *

```
from gpiozero import LED
import psutil
from datetime import datetime
import os
from time import sleep

led_r = LED(20)
led_y = LED(21)
log_file= "/home/anhem/Unit_Practice/disk_usage_log.txt"
disk_usage_percent = psutil.disk_usage('/').percent

def over_60():
    led_r.on()
    led_y.off()
def over_30():
    led_r.off()
    led_y.on()
def under_30():
    led_r.off()
    led_y.off()
try:
    while True:
        if (disk_usage_percent>60):
            over_60()
        elif (disk_usage_percent>30):
            over_30()
        else:
            under_30()

        os.makedirs(os.path.dirname(log_file), exist_ok = True)
        with open(log_file, "a")as f:
            timestamp= datetime.now().strftime('%Y-%m-%d %H:%M:%S')
            f.write(f"{timestamp} - Disk usage: {disk_usage_percent}")
        print(f"{timestamp} - Disk usage: {disk_usage_percent}")
        sleep(10)
        disk_usage_percent = disk_usage_percent + 10

except KeyboardInterrupt:
    print("Het roi nghi di")
finally:
    led_y.off()
    led_r.off()
    led_r.close()
    led_y.close()
```

*Figure 10: Measuring total storage space and informed to users by Led.*

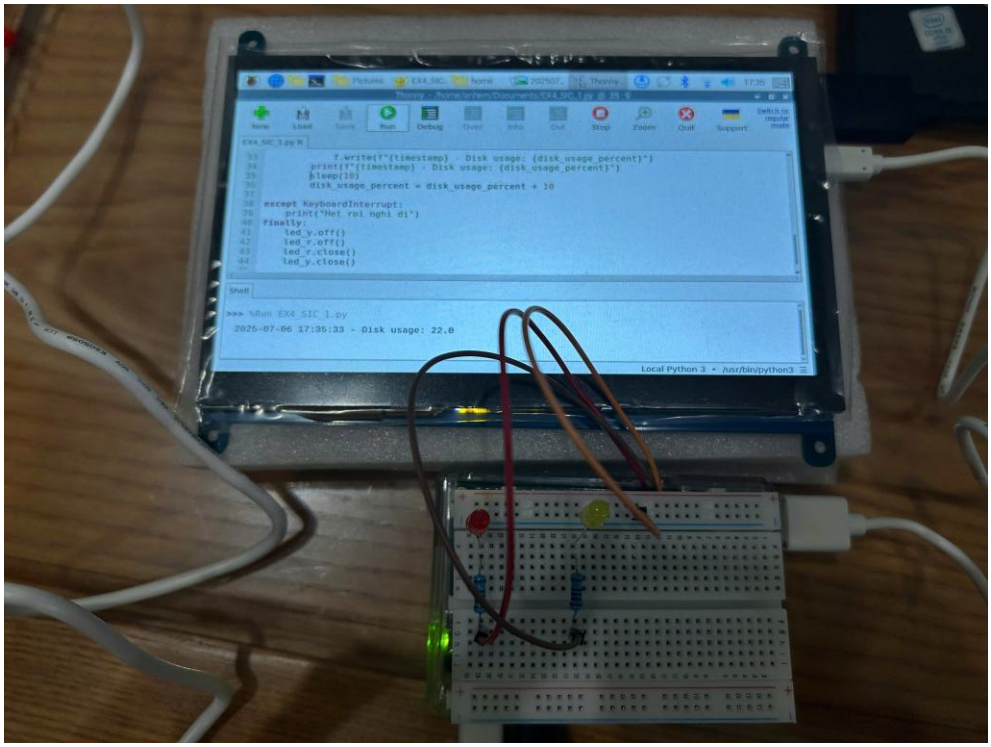


*Figure 11: When disk usage is below 30%, both LEDs will not light up.*

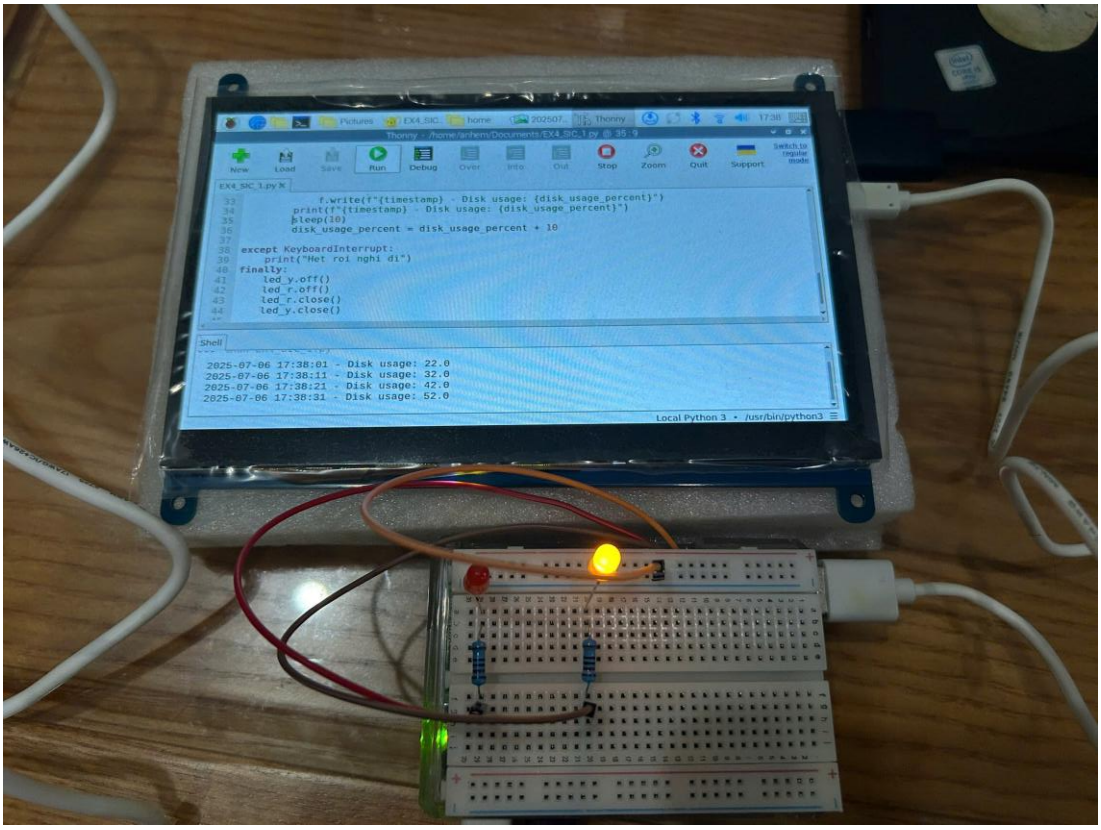


*Figure 12: When disk usage is over 30% and under 60%, the yellow LED will light up.*

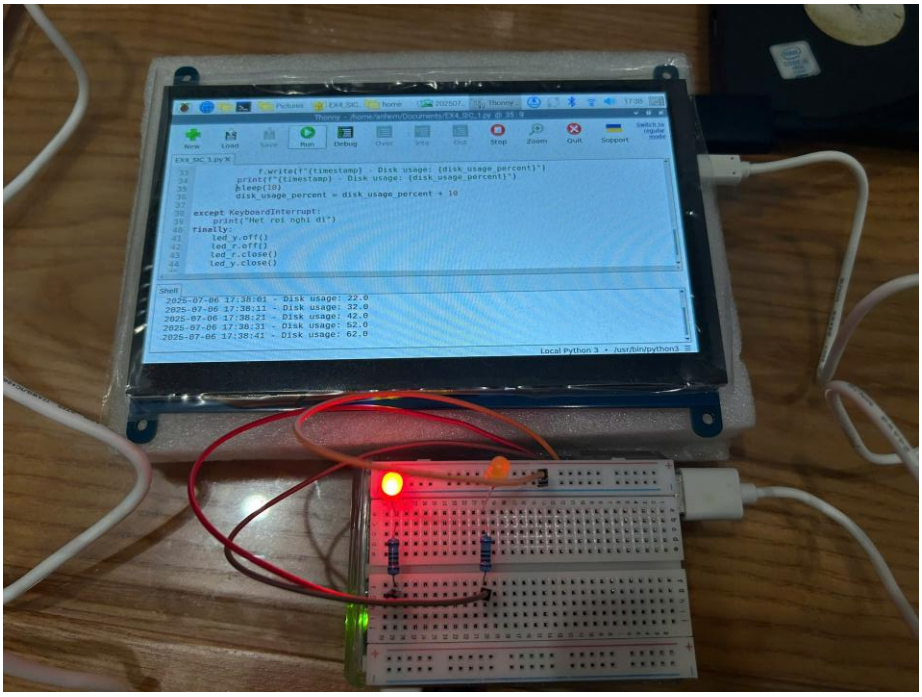

*Figure 13: When disk usage is over 60%, the red LED will light up.*

**Exercise 2.** **(Conduct the Exercise after Unit 2)**

We had a toy project that measured and logged the span of the ultrasonic sensor. This t ime, write a program that records a log after measuring only when a button is pressed. Also, it turns on the LED only when the distance is smaller than the span during measu rement. Save the measured value as a log in /home/pi/Unit_Practice/distance_log.txt.

EX6_SIC.py | EX2_SIC_LED.py * | EX4_SIC_1.py * | EX4_SIC_2.py

```
from gpiozero import LED, Button, DistanceSensor
from time import sleep
import os
from datetime import datetime

led=LED(21)
button= Button(20)
sensor= DistanceSensor(echo=12, trigger=16)
ld= 30
log_file= "/home/anhem/Unit_Practice/distance_log.txt"

def distance():
    kc= sensor.distance*100
    return kc
def exit_do():
    raise KeyboardInterrupt


try:
    print(" An nut di cu")
    button.wait_for_press()
    button.when_pressed = exit_do
    while True:
        kc= distance()
        os.makedirs(os.path.dirname(log_file), exist_ok = True)
        with open(log_file, "a")as f:
            timestamp= datetime.now().strftime('%Y-%m-%d %H:%M:%S')
            f.write(f"{timestamp} - Distance: {kc}")
        if kc<ld:
            led.on()
            print(f"Dangerous Zone {kc}")
        else:
            led.off()
            print(f"{timestamp} - Distance: {kc}")
        sleep(0.07)


except KeyboardInterrupt:
    print("Het roi nghi di")
finally:
    led.off()
    led.close()
    sensor.close()
    button.close
```

*Figure 14: The program that measures distance using an ultrasonic sensor, controls an LED based on the dist ance, logs distance data to a file, and has a button to exit the program.*

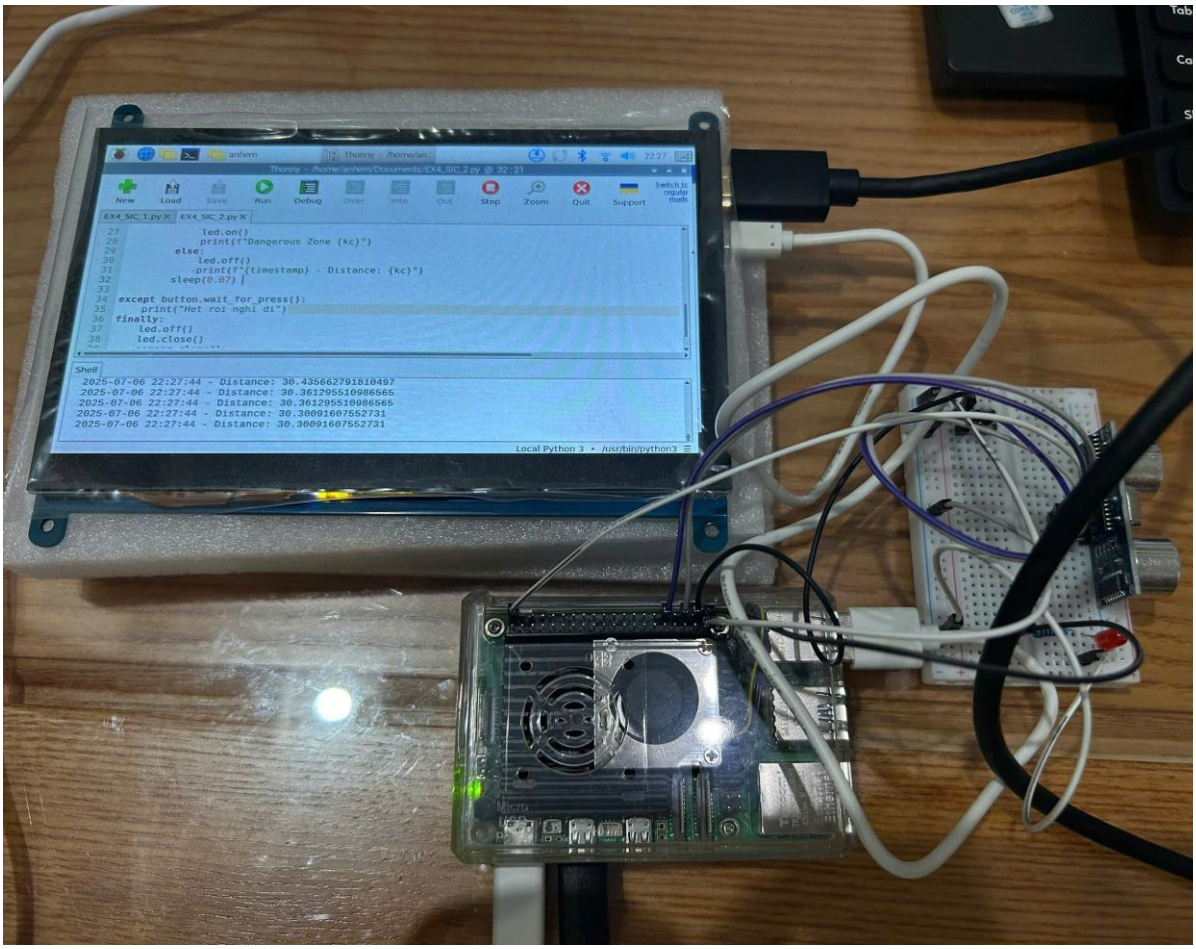


*Figure 15: When the distance is greater than 30, it's a safe zone, so the red LED does not light up.*

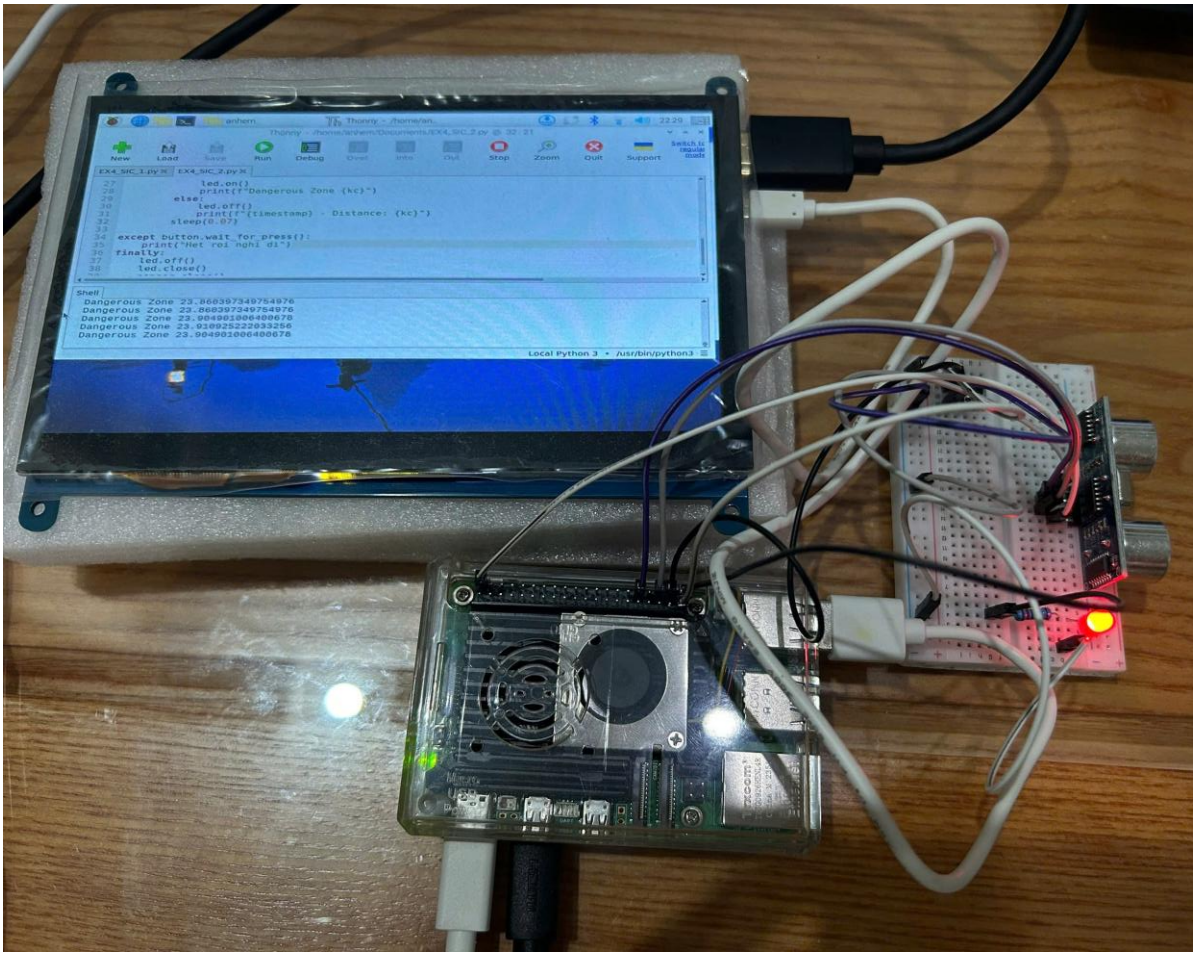


*Figure 16: When the distance is under 30, it's a dangerous zone, so the red LED will light up.*

*Figure 17: Our group picture when finishing exercise 4.*

# Chapter 5

1. What is the command to go directly to the root directory?

-This command takes you directly to the root directory of the file system.
**Cd/**

2. What is the command to show the inode number of the home directory, regardless of the current location (d irectory)? (2 lines)

-To show the inode number of the home directory from any location, use: **ls -id $HOME**
-This command lists the inode number of the home directory ($HOME) using the -i flag with ls.

3. If a file called a.txt exists in the current directory, what is the command to output a string in the contents of a.txt that starts with a number and the second letter is lowercase in English along with the line number?

- To find and display lines in a.txt that start with a number followed by a lowercase English letter, along with the ir line numbers, use: **grep -n '^[0-9][a-z]' a.txt**
-Explanation:

- **grep** searches for patterns in the file.
- **-n** displays the line number for each match.
- **^[0-9]** matches a line starting with a digit (0-9).
- **[a-z]** matches a lowercase English letter as the second character.
- **a.txt** is the file to search in.

4. What does the tar cf filename.tar a.txt b command do?

-The command **tar cf filename.tar a.txt b** does the following:

- Creates a new tar archive named **filename.tar.**
- Includes the file **a.txt** and the file or directory **b** in the archive.
- **c** means create a new archive.
- **f** specifies the archive file name (**filename.tar**).

If **b** is a file, it's added to the archive along with **a.txt.** If **b** is a directory, the entire directory and its contents are included. The command does not compress the archive; it only bundles the files.

5. What is the command to check the information of the currently logged-in user and switch to the user whose user ID is daniel? (2 lines)

-To check the information of the currently logged-in user and switch to the user with the user ID "daniel", use:
**Whoami**
**su - daniel**
-Explanation:

- **whoami** displays the username of the currently logged-in user.
- **su - daniel** switches to the user "daniel" with their environment (assuming "daniel" is a valid user ID on the system).

6. After checking the IP address of the Raspberry Pi, save the IP address in a variable called MYIP. What is th e command to check whether the external host server has network access and saves the result to a file calle d test in the current directory? (Assume the IP address is 192.168.0.0) (3 lines)

-To check the IP address of the Raspberry Pi, save it to a variable called MYIP, and then test network access to a n external host server (using the provided IP address 192.168.0.0) while saving the result to a file called test in th e current directory, use the following commands:
**MYIP=$(hostname -I | awk '{print $1}')**
**ping -c 4 192.168.0.0 > test**
-Explanation:

- **MYIP=$(hostname -I | awk '{print $1}')** retrieves the Raspberry Pi's IP address using **hostname -I** an d stores the first IP address in the **MYIP** variable.
- **ping -c 4 192.168.0.0** sends 4 ping requests to the external host (192.168.0.0) to check network access.
- **> test** redirects the ping output to a file named **test** in the current directory.

Note: The IP address 192.168.0.0 is typically a network address, not a host address, so pinging it may not work a s expected. A valid host IP (e.g., 192.168.0.1) might be more appropriate for testing connectivity.

7. A user named abc tried to read a.txt, but could not read it because it was neither a user nor a group. In this c ase, what is the command for abc to read a.txt?

To allow the user abc to read the file a.txt when they are neither the owner nor part of the group with read permissions, you need to grant read permission to "others" (users not in the owner or group categories). The command is: **chmod o+r a.txt**
Explanation:

- **chmod** changes the file's permissions.
- **o+r** adds read (**r**) permission for "others" (users like **abc** who are not the owner or in the group).
- **a.txt** is the target file.

This command ensures **abc** can read **a.txt** by granting read access to all users in the "others" category.

# Chapter 6

**Exercise 1. (Conduct the Exercise after Unit 4)**

When the input string is given as ‘1 ON’ or ‘2 OFF’, write a program that controls the two connected LEDs according to the meaning of the string. Modify the Bluetooth_led.py file to write it.

Car_checked.py | Bluetooth_led.py | Test_bluetooth.py | EX6_SIC.py *

```
import serial
import RPi.GPIO as GPIO

Led_1 = 20
Led_2 = 21

GPIO.setmode(GPIO.BCM)
GPIO.setup(Led_1,GPIO.OUT)
GPIO.setup(Led_2,GPIO.OUT)

ser = serial.Serial('/home/anhem/Documents/', 115200)
ser.close()
ser.open()

welcome = b'Bluetooth LED Control\r\n'
n = ser.write(welcome)

try:
    while True:
        if ser.readable():
            respone =ser.readline()
            if respone == b'1 ON\n':
               GPIO.output (Led_1 ,True )
            elif respone ==b'2 ON\n':
                GPIO.output (Led_2, True)
            elif respone ==b'1 OFF\n':
                GPIO.output (Led_1 ,False)
            elif respone ==b'2 OFF\n':
                GPIO.output (Led_2, False)

            print(respone)
except KeyboardInterupt:
    pass
finally:
    ser.close()
```

*Figure 18: LED control program via Bluetooth Serial communication.*

This program communicates with a Bluetooth device. When it receives the commands '1 ON', '1 OFF', '2 ON', or '2 OFF', it will turn on or off the corresponding LEDs conne cted to GPIO 20 and 21. This is a basic program for controlling LEDs via Bluetooth.

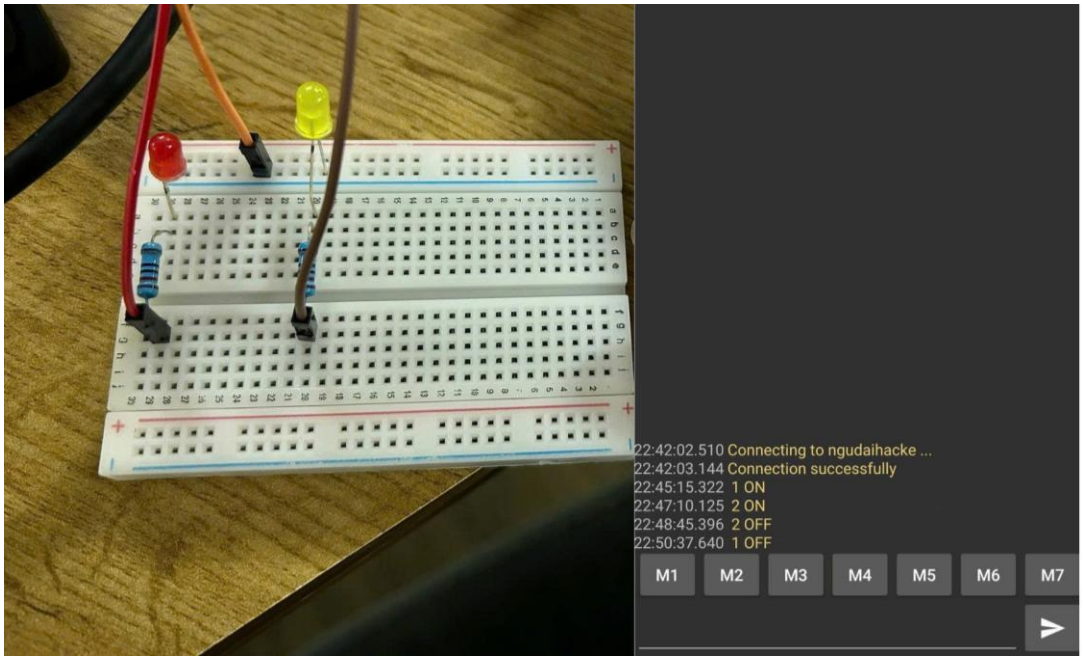


*Figure 19: Bluetooth connected success.*

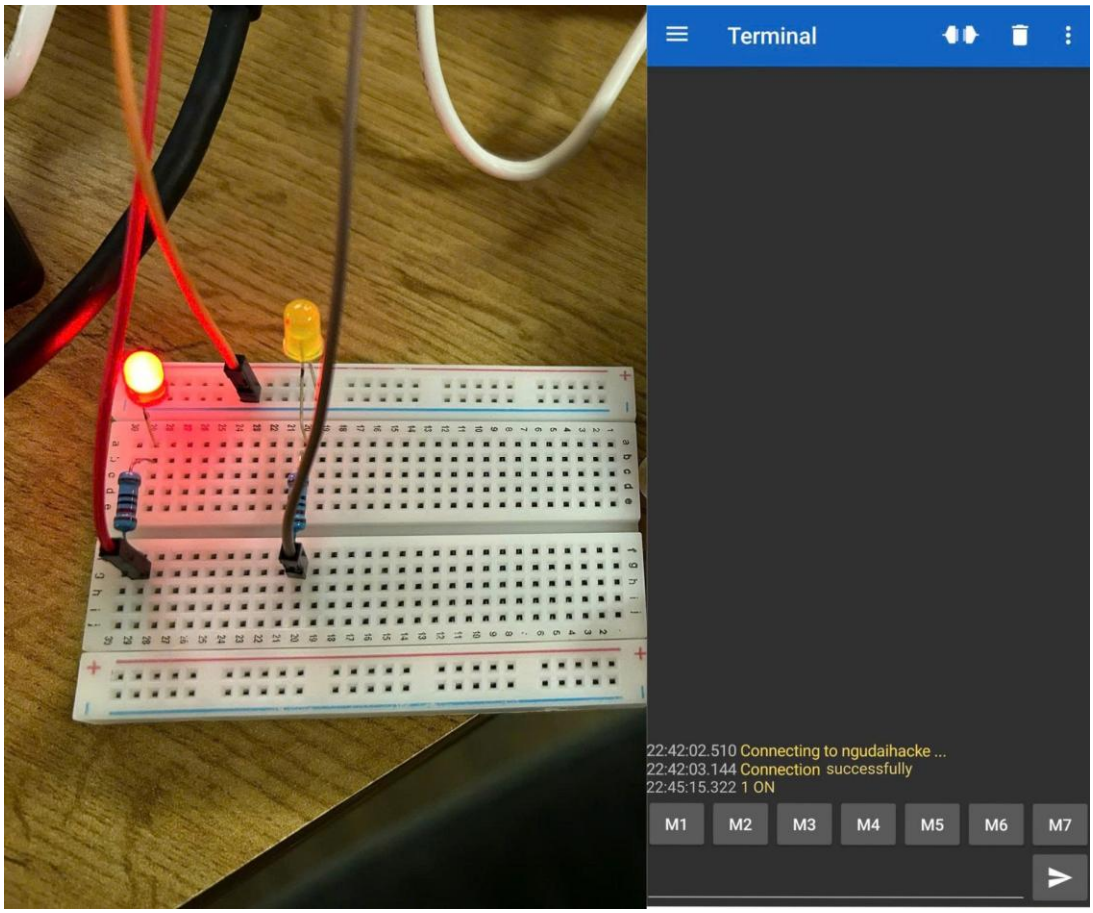


*Figure 20: When the red LED is in mode 1*

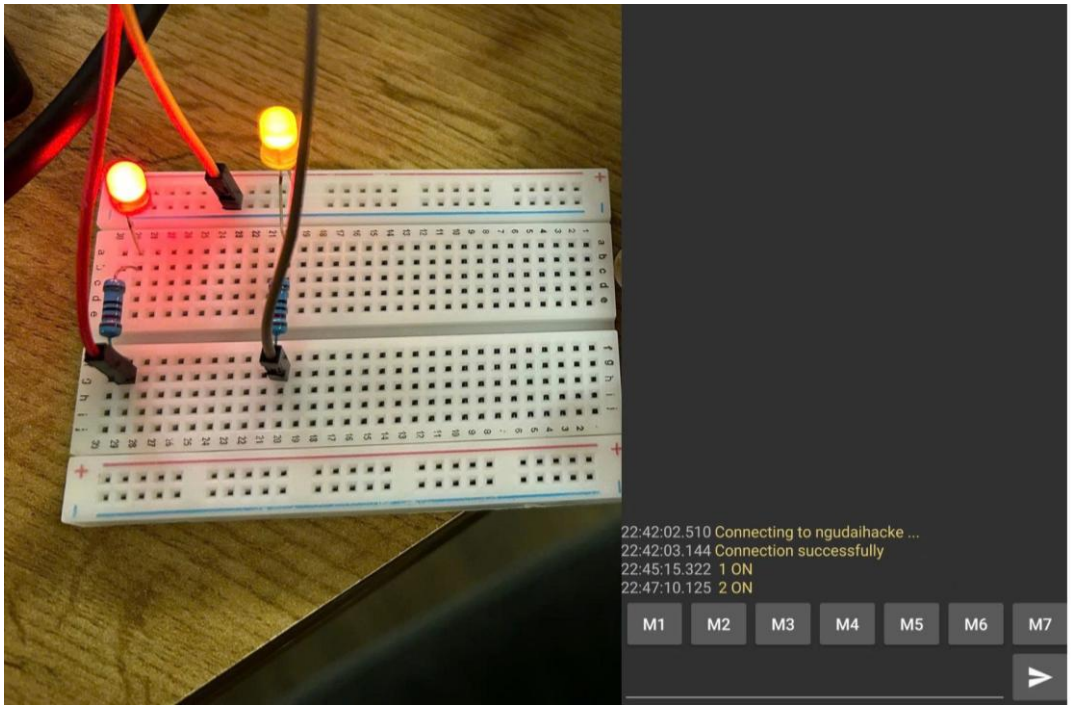


*Figure 21: When both LEDs are in mode 1, it's the on mode.*

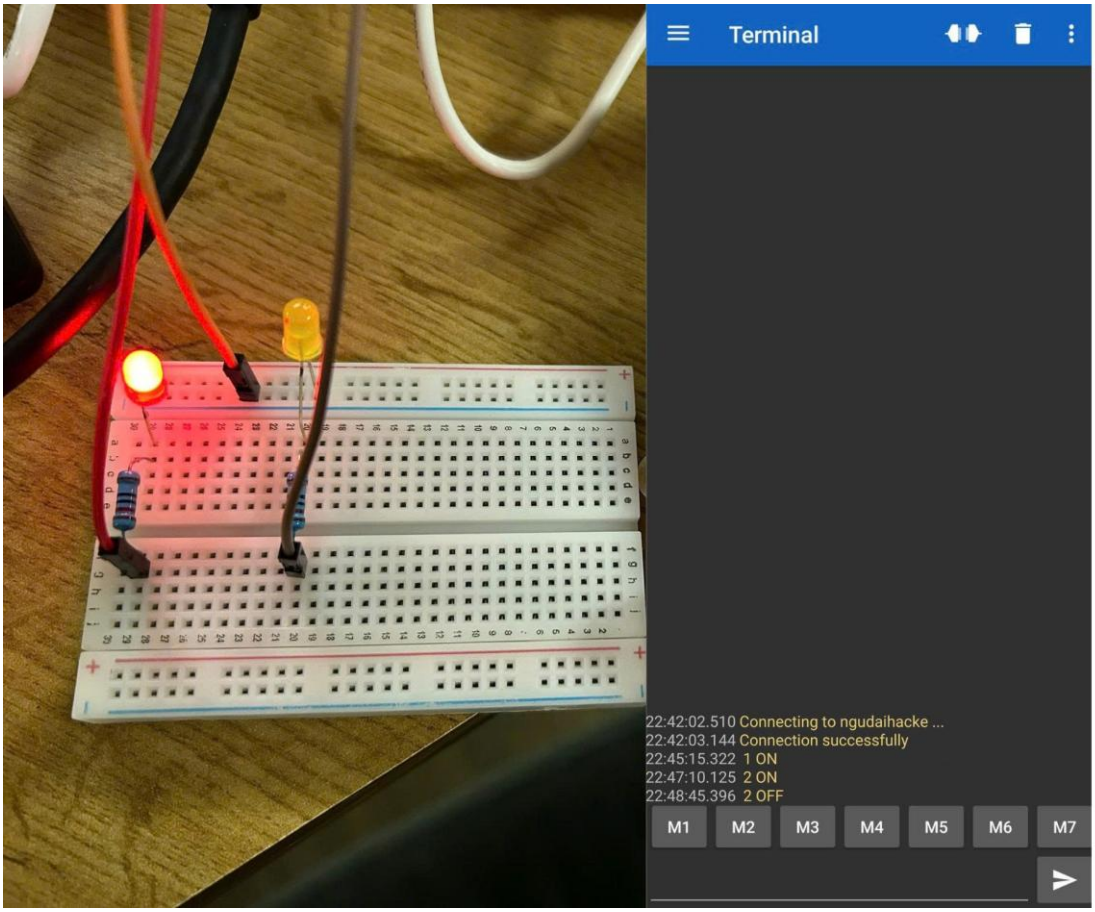


*Figure 22: When the red LED is in mode 1 and the yellow LED is in mode 2.*

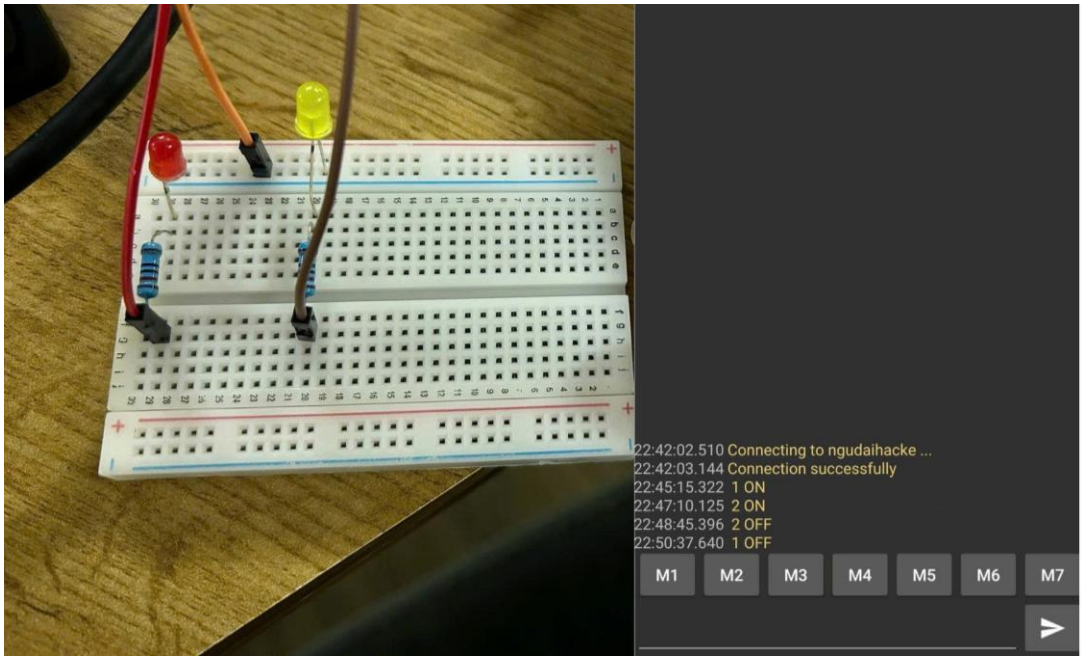


*Figure 23: When both LEDs are in mode 2, it's the off mode.*

*Figure 24: Students carrying out the assignment.*