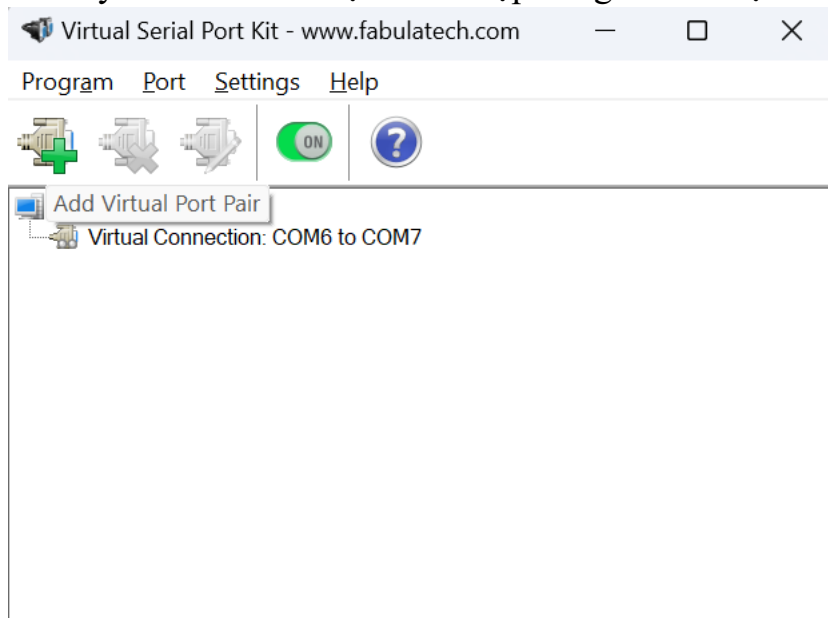
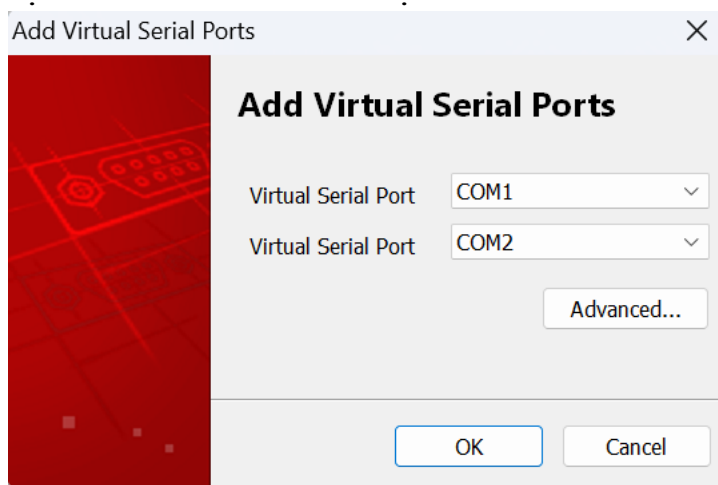


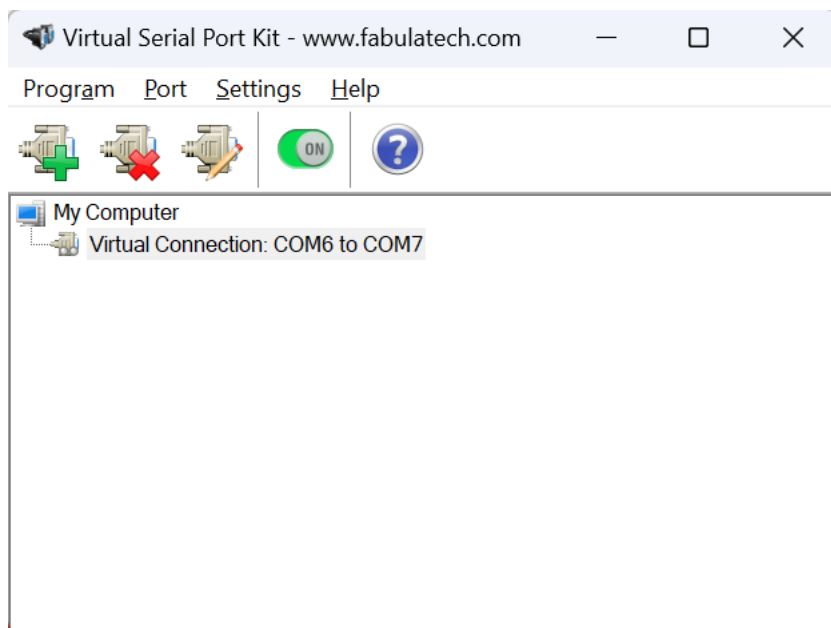
I. Tạo các thiết bị điều khiển ảo

- Bước 1: Mở phần mềm Virtual Serial Port Kit, nhấn vào biểu tượng đầu tiên bên tay trái để chuẩn bị thêm 1 cặp cổng ảo để nhận 2 thiết bị ảo

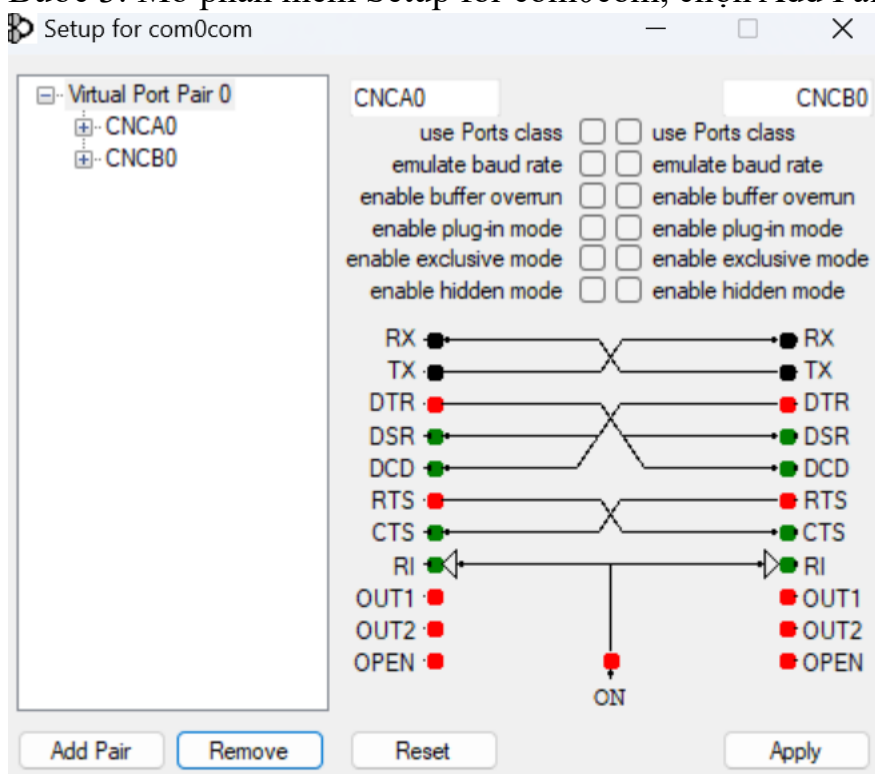


- Bước 2: Tại giao diện thêm cổng ta chọn 2 cổng mình muốn để nhận 2 thiết bị ảo sau đó nhấn OK để tạo

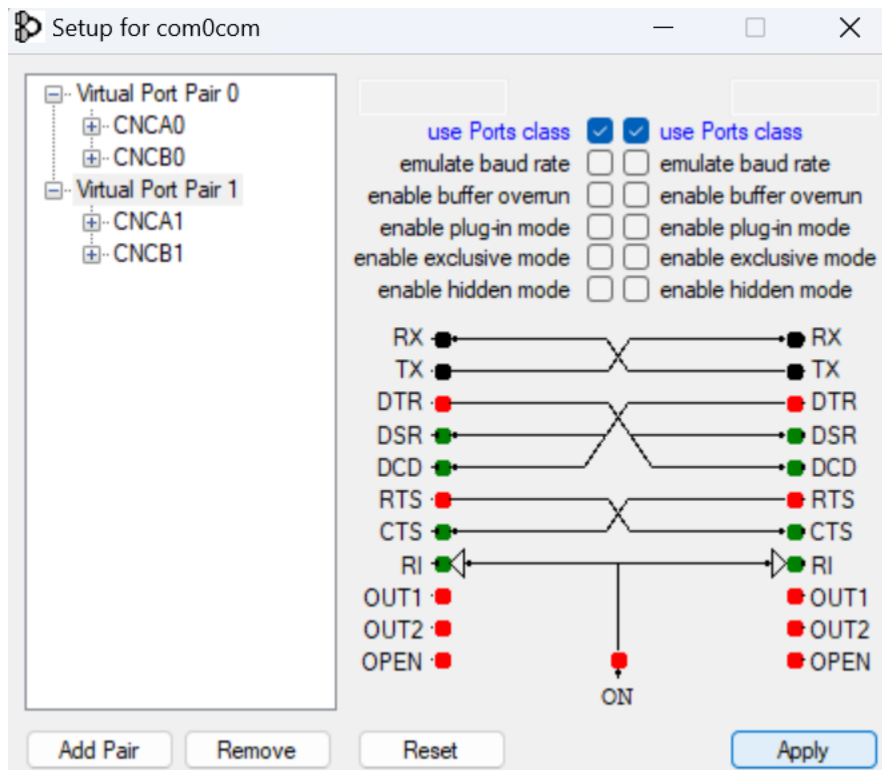




- Bước 3: Mở phần mềm Setup for com0com, chọn Add Pair:



- Bước 4: Tại port pair ảo ta vừa tạo, tích vào use Ports class cho cả 2 Port ảo đó. Cuối cùng nhấn Apply để áp dụng các thay đổi



II. Gửi tín hiệu từ thiết bị ảo đến Adafruit

- Bước 1: Ta thêm 1 file python vào folder mà khi trước đã tạo để gửi tín hiệu AI nhận diện mặt lên Adafruit. Sau đó tải pyserial bằng lệnh **pip install pyserial**.

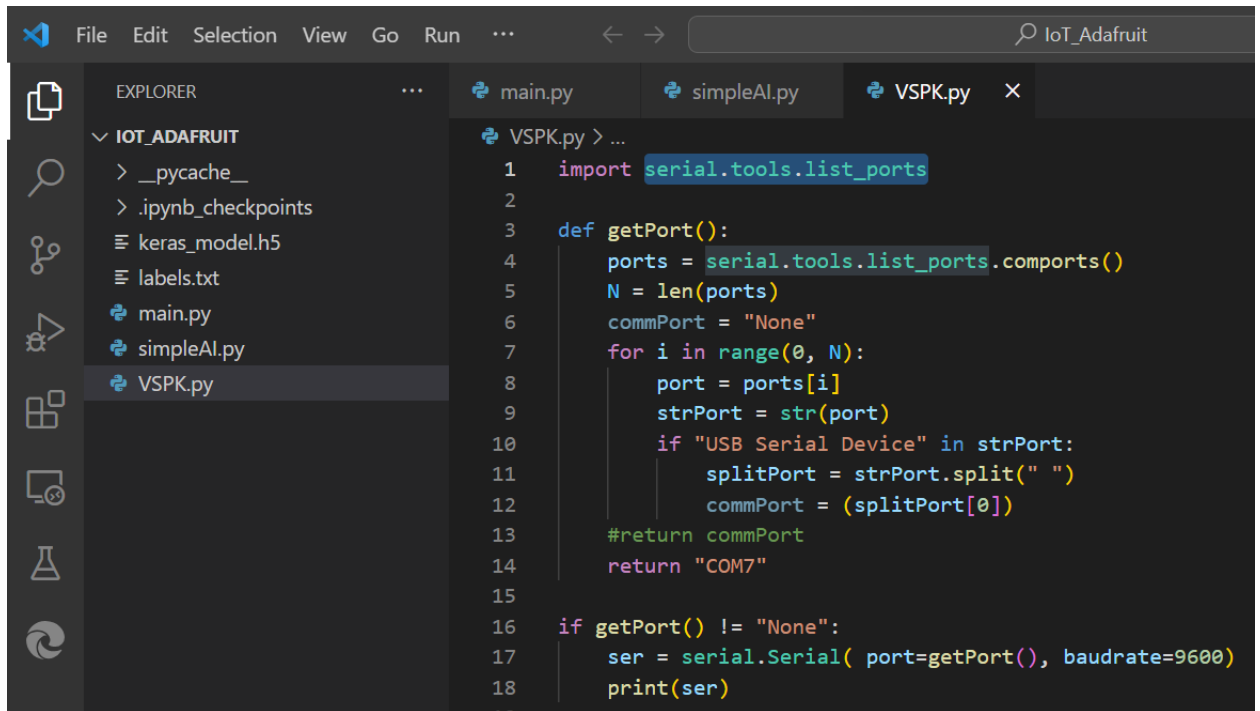
```

C:\Windows\system32\cmd.e
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\tuana>pip install pyserial

```

- Bước 2: Vào file vừa tạo import thư viện *serial.tools.list_ports* tạo hàm getPort để nhận các port đang chạy trên máy. Lệnh `serial.Serial(port, baudrate)` dùng khởi chạy port tương ứng.



```
1 import serial.tools.list_ports
2
3 def getPort():
4     ports = serial.tools.list_ports.comports()
5     N = len(ports)
6     commPort = "None"
7     for i in range(0, N):
8         port = ports[i]
9         strPort = str(port)
10        if "USB Serial Device" in strPort:
11            splitPort = strPort.split(" ")
12            commPort = (splitPort[0])
13        #return commPort
14        return "COM7"
15
16 if getPort() != "None":
17     ser = serial.Serial( port=getPort(), baudrate=9600)
18     print(ser)
```

- Bước 3: Tạo hàm processData để đưa dữ liệu lên client và hàm readSerial đọc và chuyển hóa các dữ liệu được mã hóa.

```

def processData(client, data):
    data = data.replace("!", "")
    data = data.replace("#", "")
    splitData = data.split(":")
    print(splitData)
    if splitData[1] == "T":
        client.publish("nhiet-do", splitData[2])
    elif splitData[1] == "H":
        client.publish("do-am", splitData[2])

mess = ""
def readSerial(client):
    bytesToRead = ser.inWaiting()
    if (bytesToRead > 0):
        global mess
        mess = mess + ser.read(bytesToRead).decode("UTF-8")
        while ("#" in mess) and ("!" in mess):
            start = mess.find("!")
            end = mess.find("#")
            processData(client, mess[start:end + 1])
            if (end == len(mess)):
                mess = ""
            else:
                mess = mess[end+1:]

```

- Bước 4: Quay lại file main.py để import file mới tạo vào, sau đó gọi hàm readSerial và truyền biến client vào để đưa dữ liệu lên Adafruit

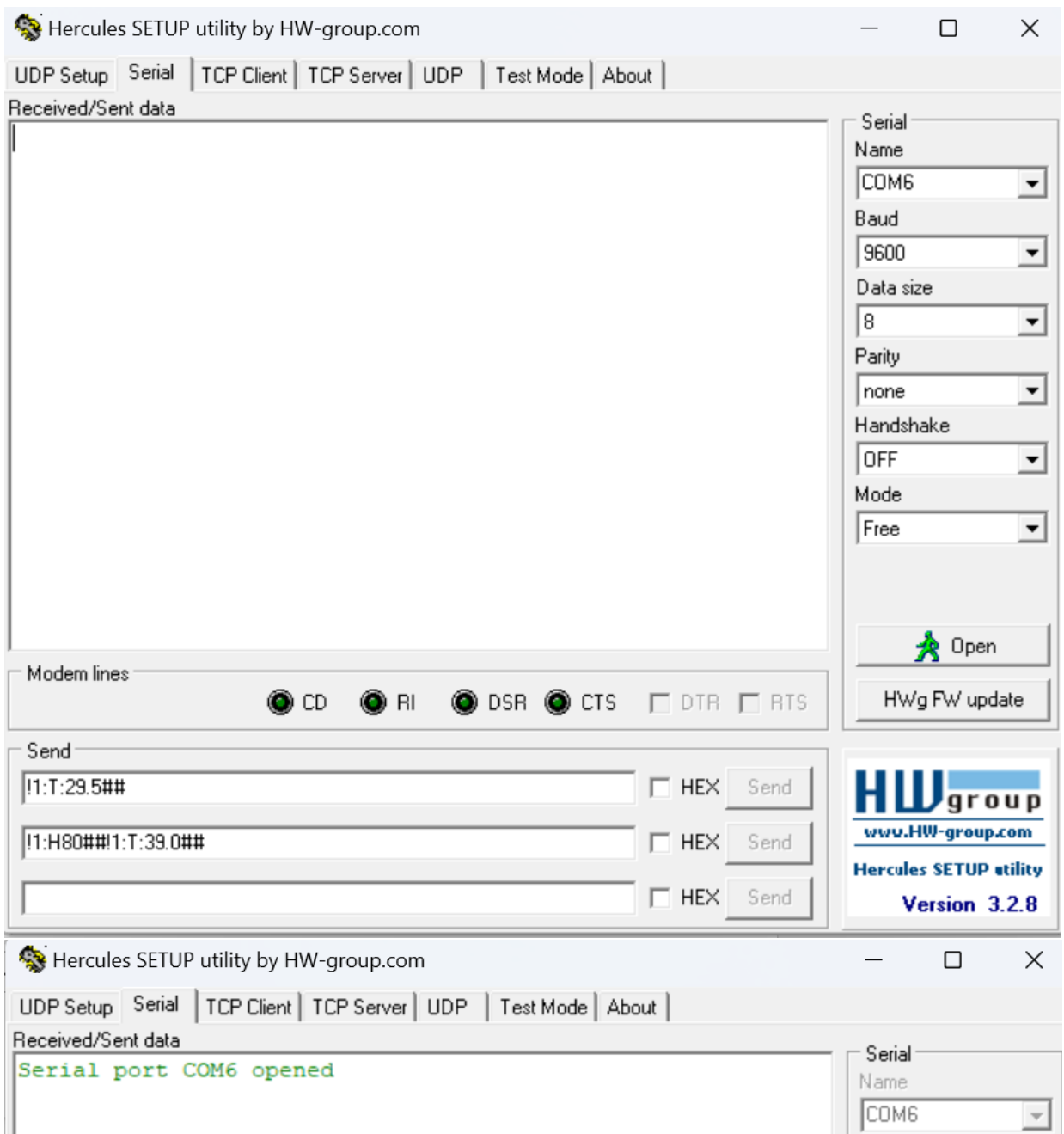
```

from VSPK import *

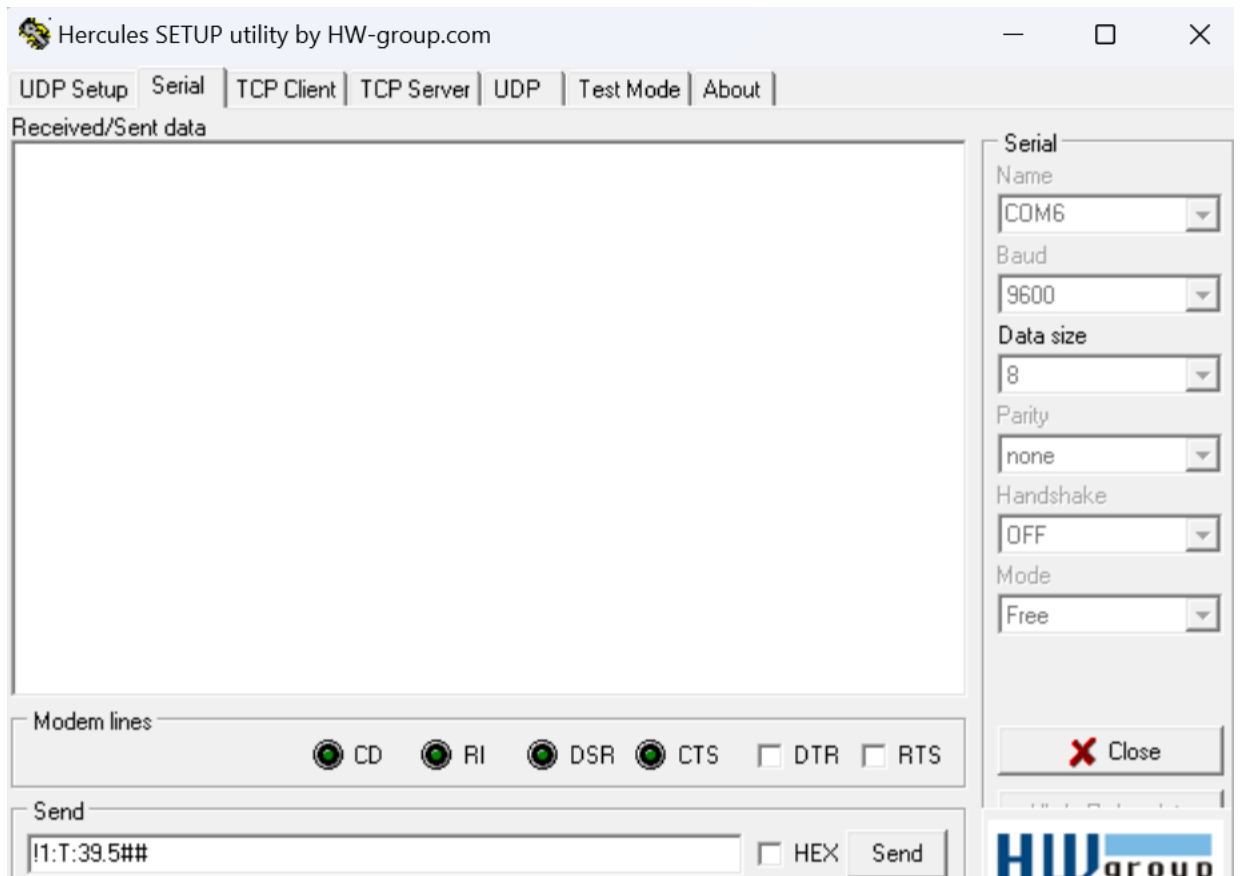
readSerial(client)

```

- Bước 5: Mở ứng dụng hercules vào tag Serial. Tại thông tin Serial bên phải cửa sổ chọn tên port còn lại (port chưa được nhận trong hàm getPort), trong phần Baud ta chọn 9600. Cuối cùng nhấn Open để mở port.

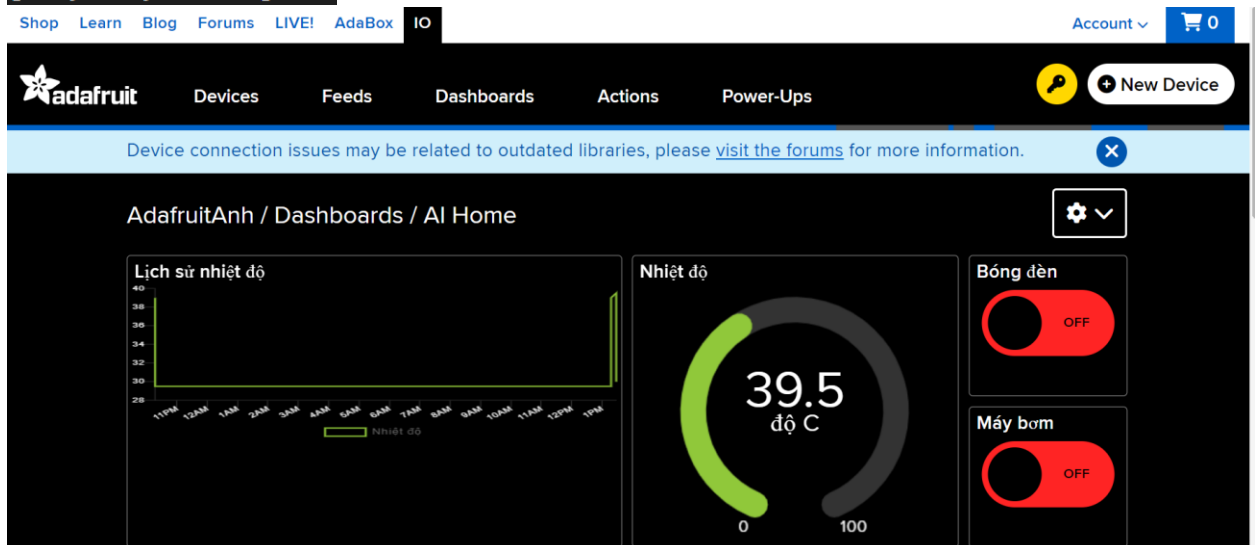


- Bước 6: Tại phần Send nằm ở phía dưới cùng của cửa sổ. Ta thêm dòng !1:T:39.5## vào textbox.

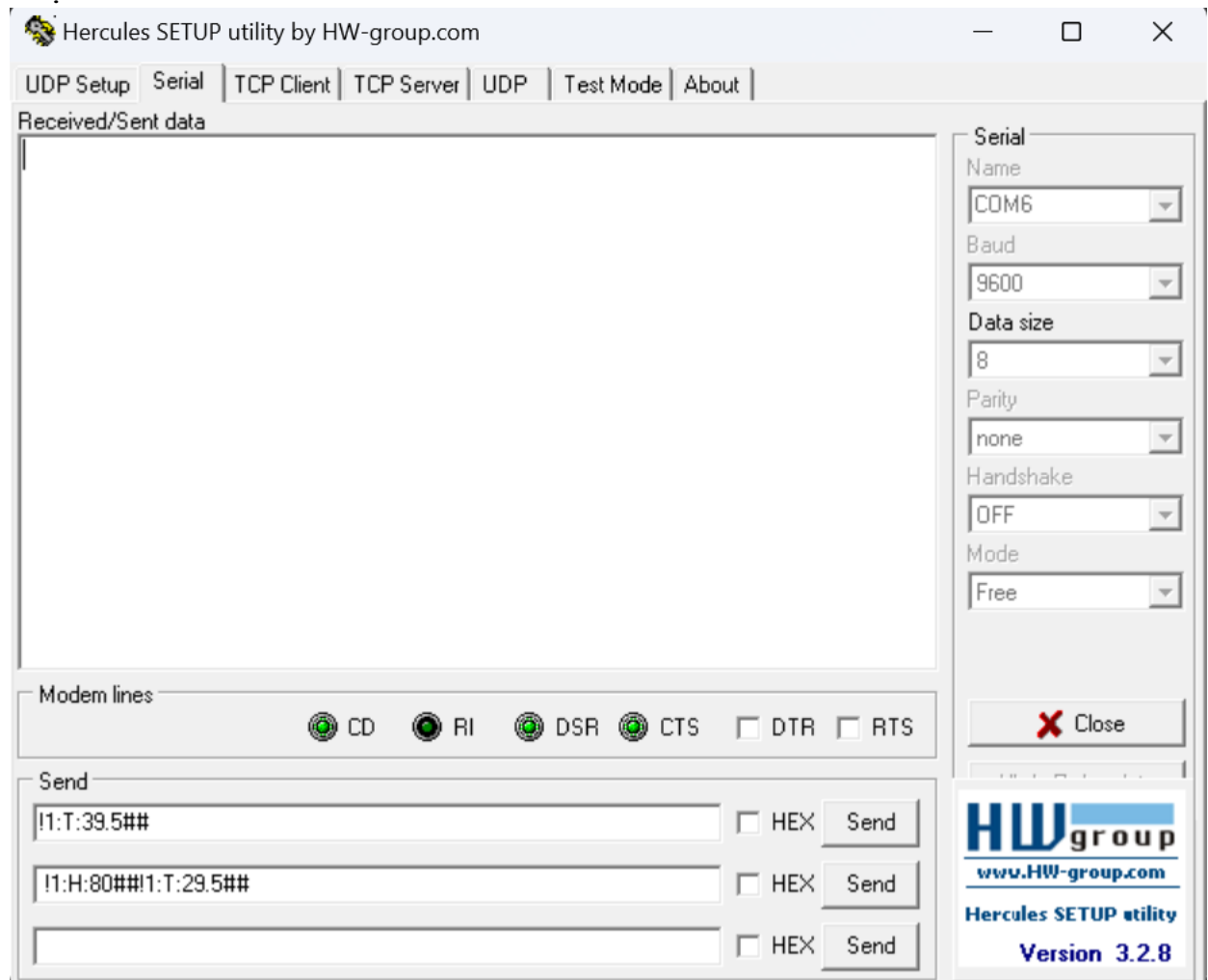


- Bước 7: Chạy file main.py sau đó quay lại ứng dụng hercules nhấn nút Send cạnh textbox mà ta vừa thêm !1:T:39.5## vào. Kiểm tra xem kết quả đã được gửi chưa bằng cách xem kết quả như sau:

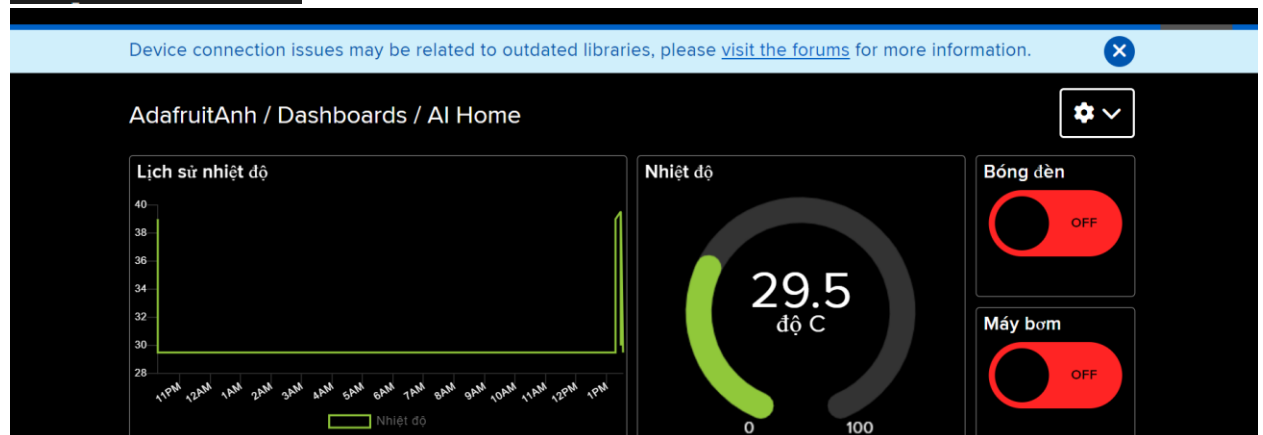
```
['1', 'T', '39.5']
```

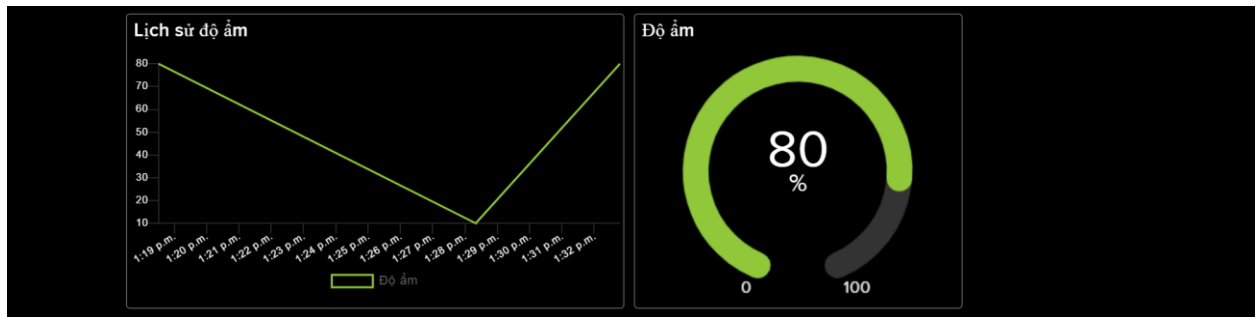


Gửi tín hiệu độ ẩm là 80 và nhiệt độ là 29.5 bằng cách thêm dòng
!1:H:80##!1:T:29.5## vào textbox như trong bước 6, sau đó lặp lại bước 7 ta
được:



```
['1', 'H', '80']  
['1', 'T', '29.5']
```





III. Nhận tín hiệu từ Adaruit

- Bước 1: Trong file đã tạo để truyền dữ liệu từ thiết bị ảo đến Adafruit ta thêm hàm `writeData(data)` để nhận dữ liệu data từ Adafruit truyền về và in nó ra.

```
def writeData(data):
    ser.write(str(data).encode())
```

- Bước 2: Quay lại file `main.py` ta gọi hàm `writeData` trong hàm `message` đã tạo trước đó. Nếu là tín hiệu từ nút bóng đèn thì ta sẽ in ra 1 khi nút đó bằng 0 và in ra 2 nếu nút đó bằng 1. Tương tự với nút máy bơm, ta in ra 3 khi nút đó bằng 0 và in ra 4 nếu nút đó bằng 1.

The screenshot shows a Python IDE with a file named `main.py` open. The code defines a `message` function that handles incoming data from Adafruit. It checks the `feed_id` and `payload` to determine which action to take. The terminal output shows the program running successfully, with messages indicating the receipt of data and the execution of the `writeData` function.

```
def subscribe(client, userdata, mid, granted_qos):
    print("Subscribe thanh cong ...")

def disconnected(client):
    print("Ngat ket noi ...")
    sys.exit(0)

def message(client, feed_id, payload):
    print("Nhan du lieu: " + payload + ", feed id: " + feed_id)
    if feed_id == "bong-den":
        if payload == "0":
            writeData("1")
        else:
            writeData("2")
    if feed_id == "may-bom":
        if payload == "0":
            writeData("3")
        else:
            writeData("4")

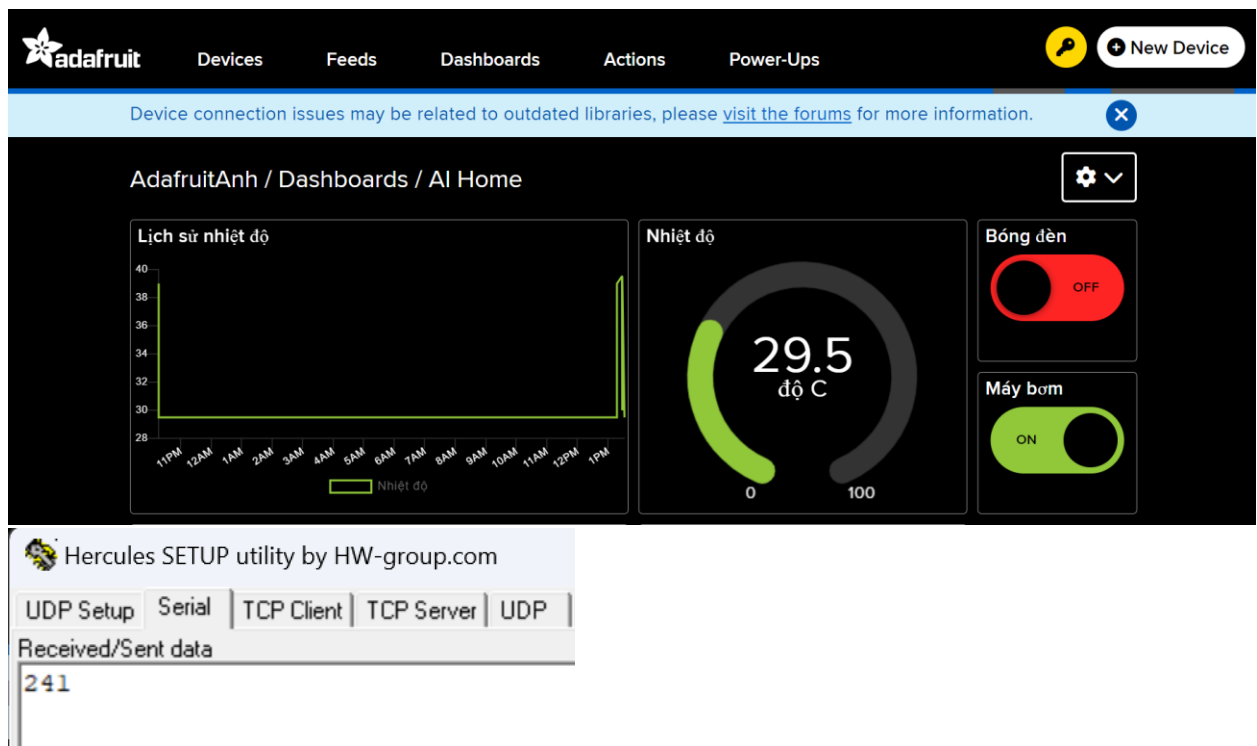
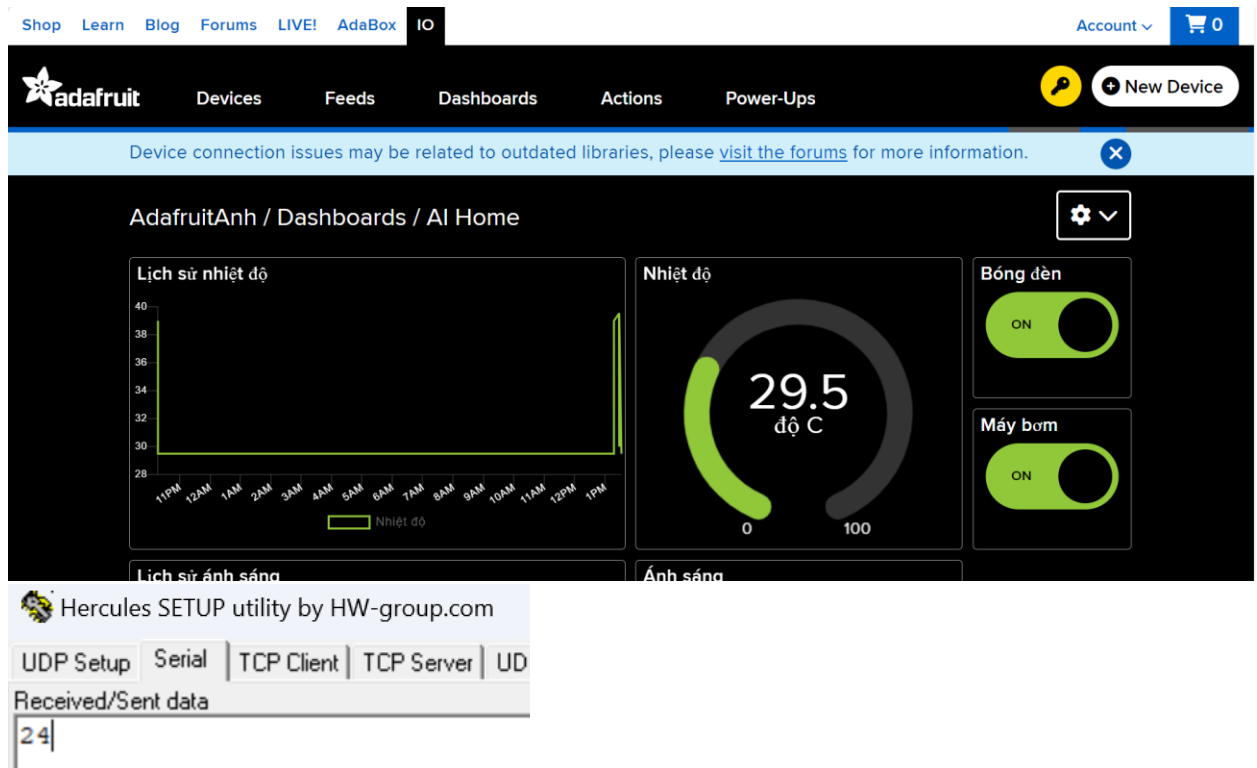
client = MQTTClient(AIO_USERNAME, AIO_KEY)
client.on_connect = connected
client.on_disconnect = disconnected

client = MQTTClient(AIO_USERNAME, AIO_KEY)
client.on_connect = connected
client.on_disconnect = disconnected
```

Terminal Output:

```
1/1 [=====] - 0s 74ms/step
Class: Không đeo khẩu trang
Confidence Score: 95 %
1/1 [=====] - 0s 113ms/step
Class: Không đeo khẩu trang
Confidence Score: 99 %
PS E:\Python\Anaconda\IoT_Adafruit>
```

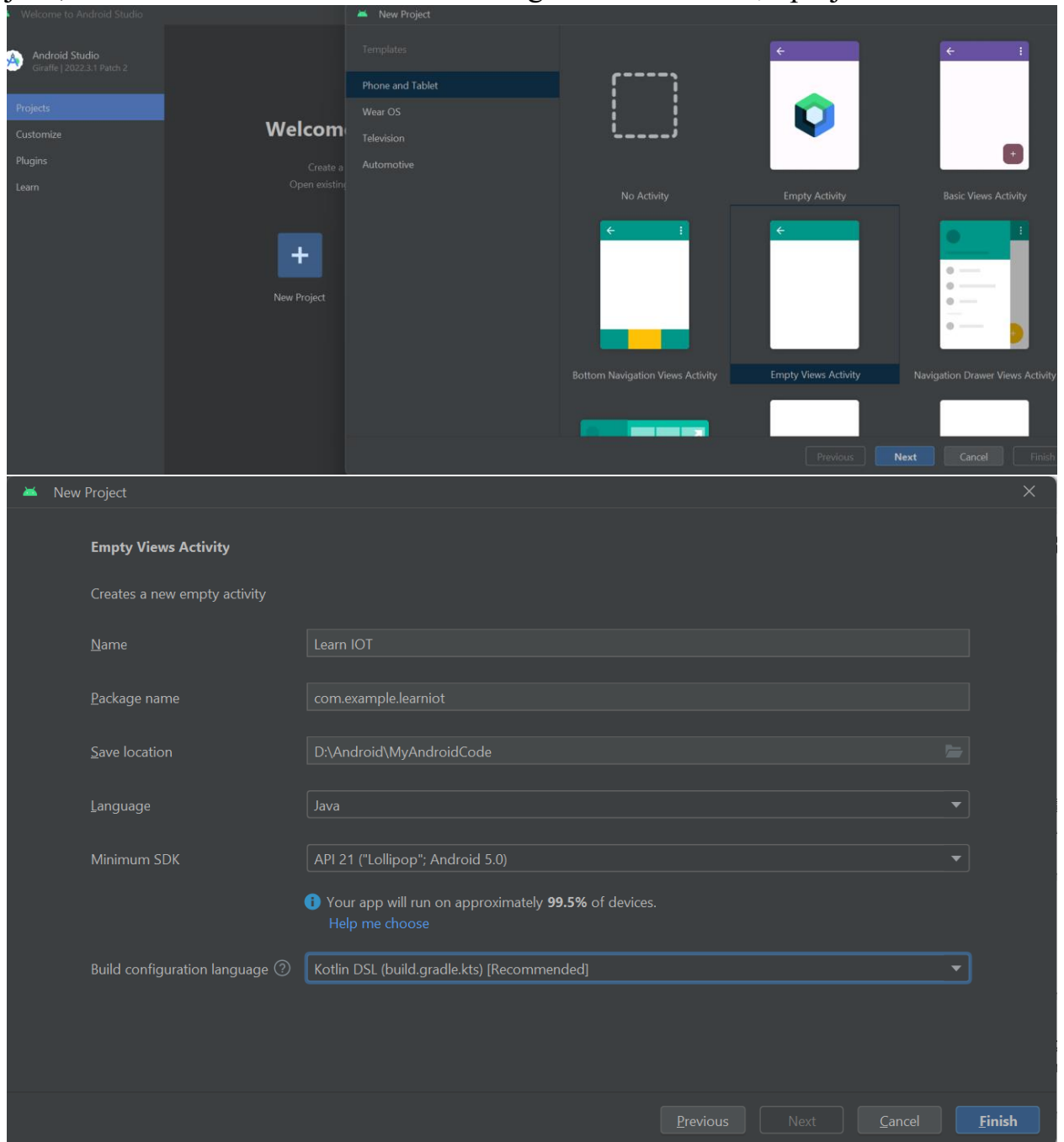
- Bước 3: Ta chạy chương trình và kiểm tra kết quả. Kết quả sẽ được hiển thị trong ứng dụng hercules giống như sau:



IV. Android Studio

- Bước 1: Mở ứng dụng Android Studio, chọn **New Project**, Chọn tag **Phone and Tablet**, chọn **Empty Views Activity**, tiếp đó nhấn **Next**. Sau đó điền tên

cho project, đường dẫn lưu project chọn ngôn ngữ sử dụng cho project là java, Minimum SDK là API 21. Cuối cùng nhấn Finish để tạo project.



- Bước 2: Mở điện thoại Android. Vào mục cài đặt → chọn Giới thiệu điện thoại → Nhấn 7 lần vào mục phiên bản MIUI để trở thành nhà phát triển



Giới thiệu điện thoại



Phiên bản MIUI

MIUI Global
13.0.2
Ổn định

Tên thiết bị

Redmi 9

Bộ nhớ

Đã chiếm dụng
54.7 GB/64 GB

Phiên bản MIUI

MIUI Global 13.0.2
Ổn định
13.0.2.0(SJCMIXM
)

Phiên bản Android

12
SP1A.210812.016

Bản cập nhật Android

Bạn đã là nhà phát triển!
22-10-01

Tất cả thông số



- Bước 3: Vào cài đặt bổ sung, chọn chức năng Tùy chọn cho nhà phát triển, kéo xuống và mở chức năng Gỡ lỗi USB, mở Cài đặt qua USB và Gỡ lỗi USB(Cài đặt bảo mật).



Cài đặt bổ sung

Làm sạch loa



Hỗ trợ tiếp cận



Mở rộng bộ nhớ



Chế độ doanh nghiệp



Tùy chọn nhà phát triển



Cần cài đặt khác?

Sao lưu và khôi phục

Hiệu ứng âm thanh

Màn hình tràn cạnh

Đặt lại dữ liệu gốc

← Tùy chọn nhà phát triển

Ô cài đặt nhanh dành cho nhà phát triển >

GỠ LỖI

Gỡ lỗi USB

Chuyển sang chế độ gỡ lỗi khi USB được kết nối



Hủy ủy quyền gỡ lỗi USB >

Gỡ lỗi không dây

Bật chế độ gỡ lỗi khi Wi-Fi được kết nối >

Cài đặt qua USB

Cho phép cài đặt ứng dụng qua USB



Gỡ lỗi USB (Cài đặt bảo mật)

Cho phép quyền và nhập mô phỏng qua gỡ lỗi USB

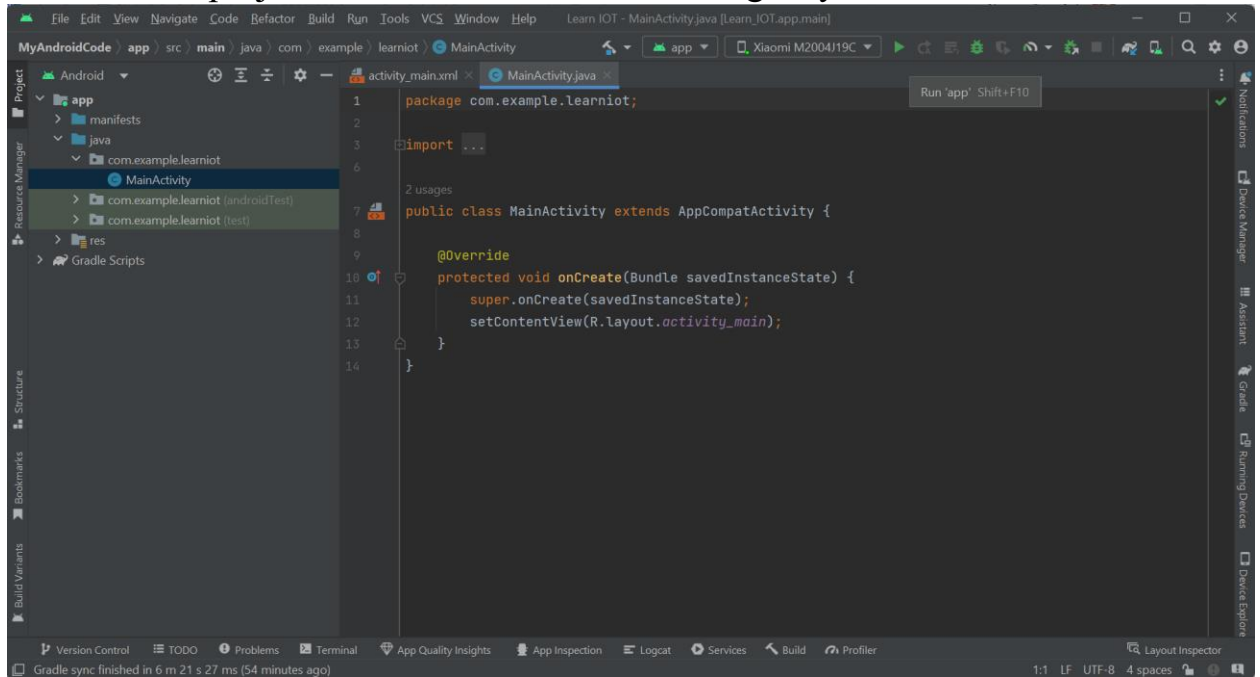


Tắt thời gian chờ ủy quyền adb

Tắt tính năng tự động thu hồi các lượt ủy quyền adb đối với những hệ



- Bước 4: Quay lại ứng dụng Android Studio và chạy file MainActivity.java để chạy và gửi chương trình qua điện thoại. Trên điện thoại sẽ hỏi bạn có muốn cài đặt project của bạn trên điện thoại không, hãy nhấn Cài đặt.



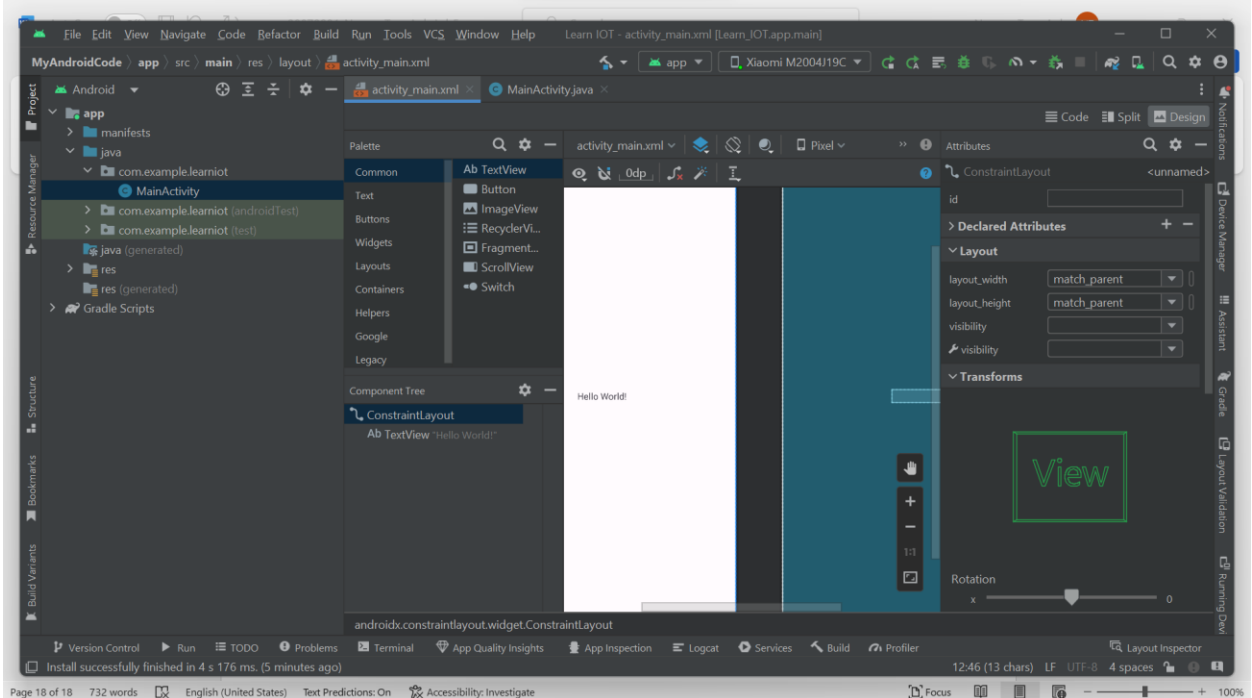
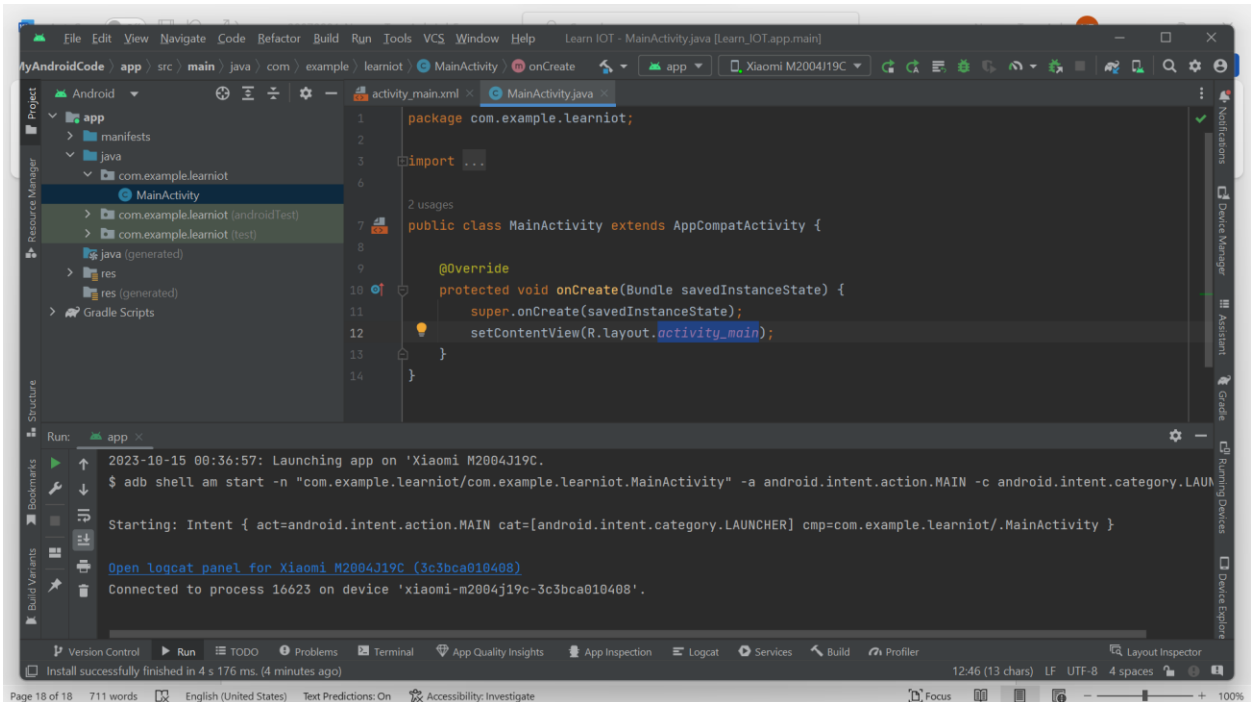
- Bước 5: Kiểm tra xem màn hình có hiển thị như thế này không.

0:37 | 4,9KB/s



Hello World!

- Bước 6: Trong file MainActivity.java nhấn giữ phím Ctrl đồng thời đưa con trỏ chuột tới activity_main và nhấn vào nó. Ta sẽ được giao diện chỉnh sửa cả dữ liệu sẽ hiển thị trên điện thoại.



- Bước 7: Chọn vào mục Split, tìm đến phần android:text = "Hello World!" đổi tên thành "Learn IOT Lab5", sau đó chạy lại chương trình ta sẽ thu được kết quả như sau:

0:48 | 4,9KB/s



Learn IOT Lab5

