IOT LAB REPORT 2024

(CO3038) Phát triển Ứng dụng Internet Of Things

Bài thí nghiệm số 2

GVHD: VŨ TRỌNG THIÊN

Người viết: Tô Hoàng Phong - 2112012

Ho Chi Minh University of Technology (HCMUT)

Khoa khoa học và kỹ thuật máy tính

Mục lục

1 Đường dẫn github và một số đường dẫn khác (nếu có)

- Đường dẫn github của bài thí nghiệm: github
- Đường dẫn dashboard của bài thí nghiệm: dashboard
- Đường dẫn video demo (nếu có)

2 Các yêu cầu

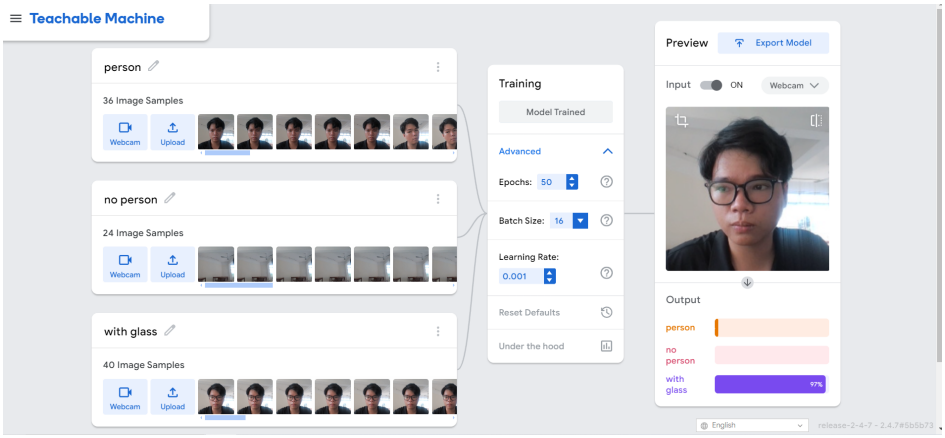
Các yêu cầu đã thực hiện được trong phần thí nghiệm này:

- Tích hợp Google Teachable Machine.
- Đọc dữ liệu từ cảm biến.
- Chạy code trên môi trường Window.
- Chạy code trên môi trường Linux.

3 Tích hợp Google Teachable Machine

Đào tạo mô hình (train model)

Ở trong lab này, máy tính sẽ chạy model, phân loại nhãn (label), rồi từ đó sử dụng camera của máy tính để phát hiện đặc trưng hình ảnh.



Hình 1: Hình ảnh model ở website

Có ba nhãn:

- person: phát hiện có người không đeo kính trong khung hình.
- no person: không có người xuất hiện trong khung hình.
- with glass: phát hiện có người và đeo hình.

Tạo file và class trong python

Trong file *'my_ai.py'*:

```
class DetectPersonModel
```

Listing 1: "class phát hiện người"

với API:

```
def detect_person(self)
```

Listing 2: "Hàm detect_person"

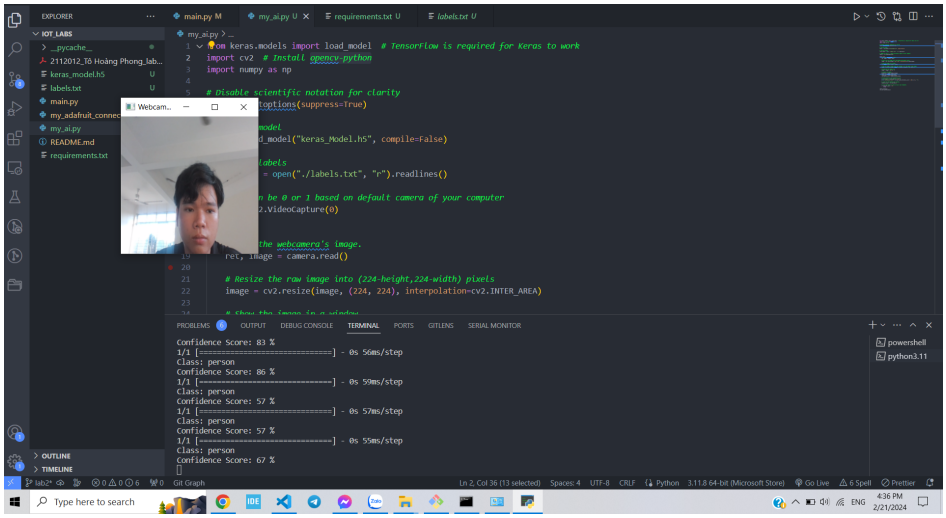
- Hàm mở camera và phát hiện người (chạy trong loop)
- Tham số: không
- Trả về: kết quả và Confidence score 'Mức độ tin cậy'

```
def exit(self)
```

Listing 3: "Hàm exit"

- Khi nhấn ESC sẽ tắt camera, đóng model, thoát chương trình
- Tham số: không
- Trả về: 0 - Không nhấn ESC
  1 - Nhấn ESC

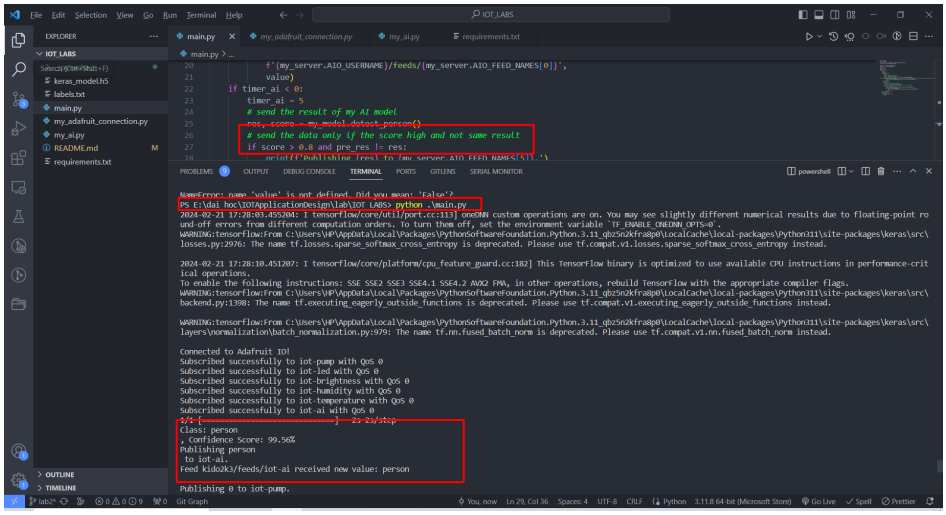
Kết quả kiểm tra



Hình 2: Kết quả kiểm tra

Khi camera phát hiện có người không đeo kính, kết quả sẽ là 'person'

Kết quả khi áp dụng vào hệ thống

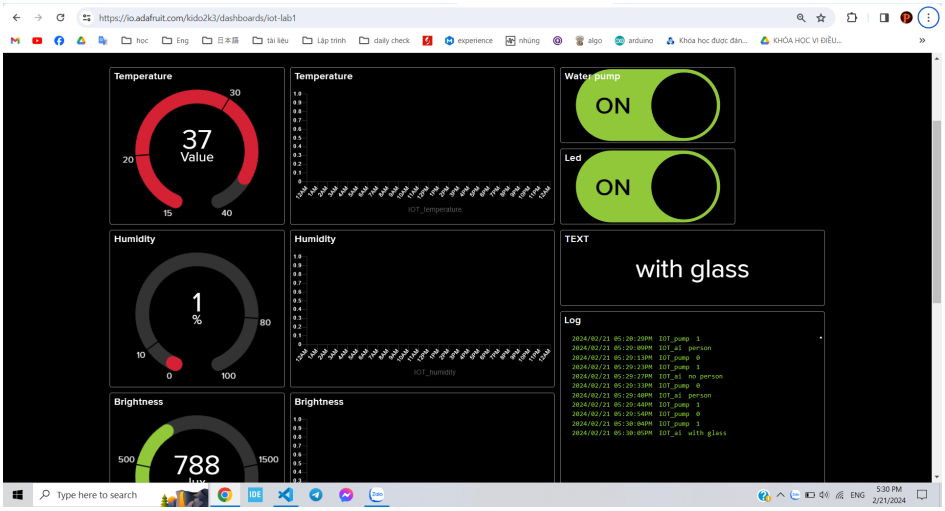


Hình 3: Kết quả khi áp dụng vào hệ thống

Lưu ý: Để giảm tải dữ liệu cần truyền lên server, ta sẽ gửi kết quả mỗi 5s, khác với dữ liệu cũ và confidence score cao

```
if timer_ai < 0:
    code...
    if score > 0.8 and pre_res != res:
        my_server.client.publish(
                f'{my_server.AIO_USERNAME}/feeds/{my_server.AIO_FEED_NAMES[5]}',
                res)
        code...
```

Listing 4: "Điều kiện gửi"

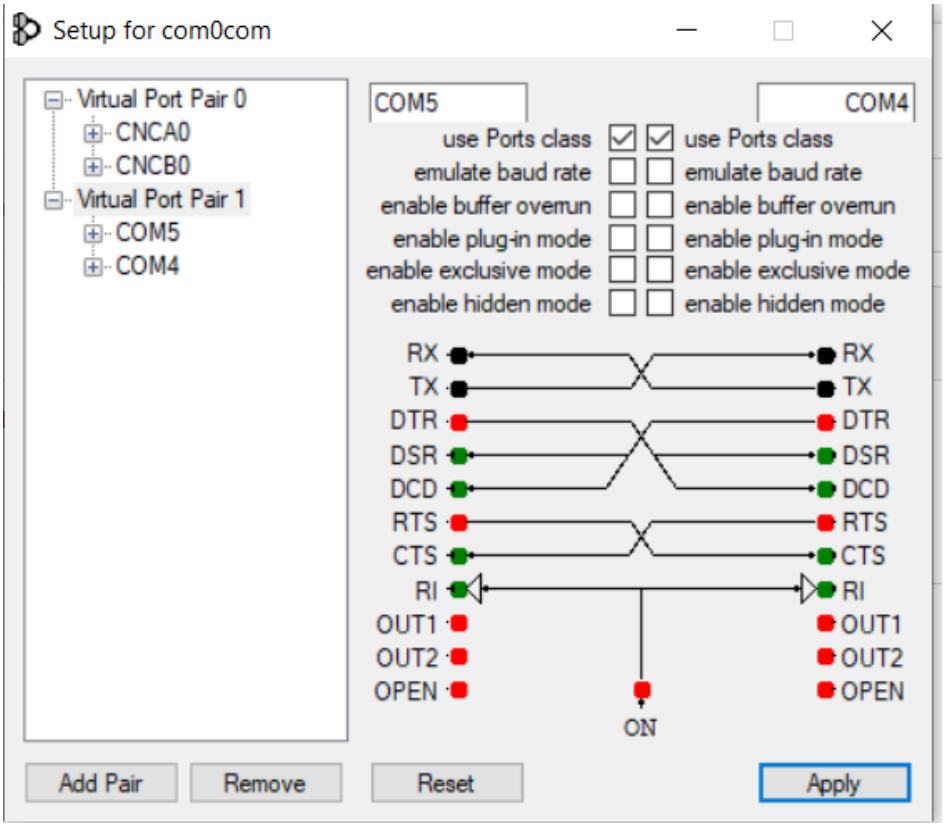


Hình 4: Kết quả trên dashboard

4 Đọc dữ liệu từ cảm biến

Trong phần này, ta cần phải xử lý giao tiếp giữa IOT gateway với thiết bị (cảm biến, vi điều khiển. Ta sẽ vừa **tạo 1 pair ảo giữa hai ports để giao tiếp**, vừa **sử dụng arduino để giao tiếp với PC**

Tạo 1 pair ảo giữa hai ports để giao tiếp



Hình 5: com0com

Trong hình có thể thấy, COM4 và COM5 được kết nối với nhau. COM4 sẽ

kết nối với máy tính, trong khi COM5 sẽ đóng vai trò vi điều khiển.

Tạo file và class trong python

Trong file *'my_serial.py'*:

```
class UART
```

Listing 5: "class uart"

với API:

```
def ProcessData(self, data, my_server)
```

Listing 6: "Hàm ProcessData"

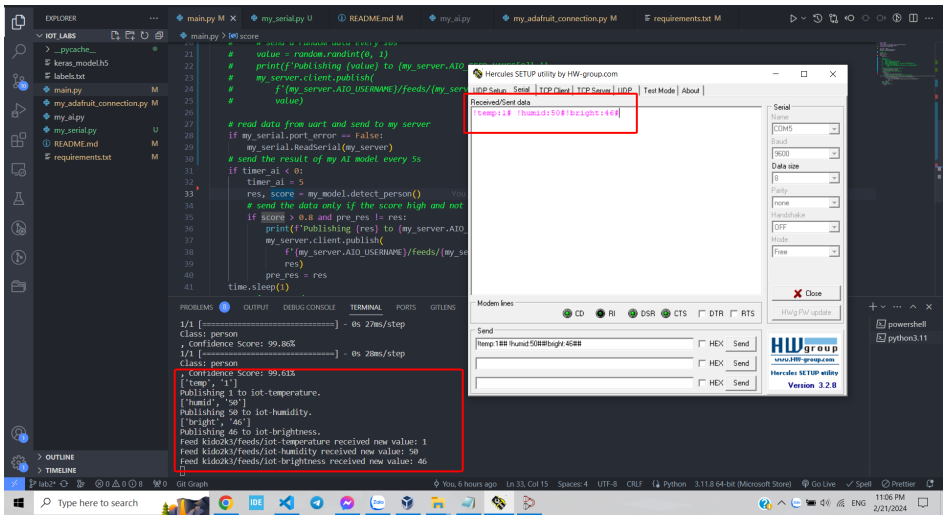
- Hàm xử lý data nhận được từ port, gửi lên server
- Tham số: data - dữ liệu từ port
  my_server - class AdafruitConnection trong *my_adafruit_connection.py*
- Trả về: không

```
def ReadSerial(self, my_server)
```

Listing 7: "Hàm ReadSerial"

- Hàm đọc data (chạy trong loop)
- Tham số: my_server - class AdafruitConnection trong *my_adafruit_connection.py*
- Trả về: không

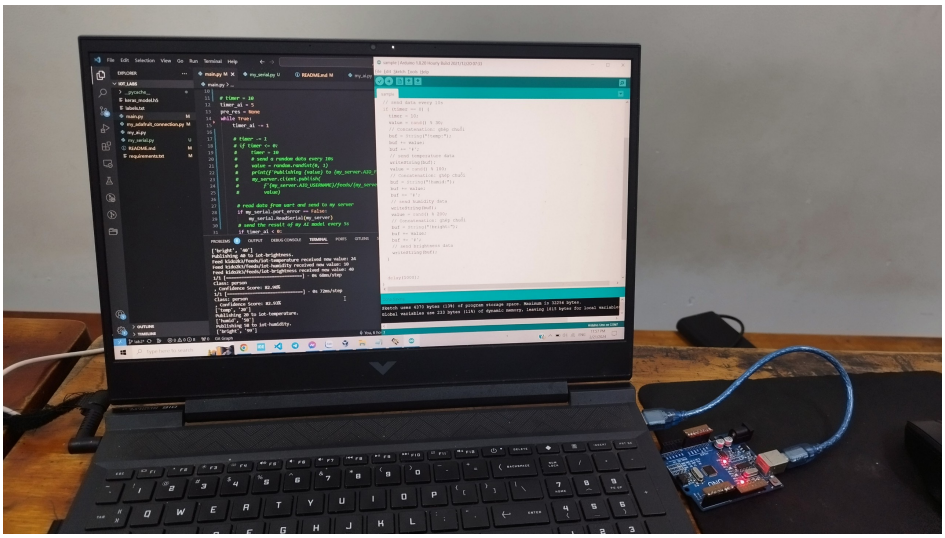
Kết quả khi áp dụng vào hệ thống, chạy COM ảo



Hình 6: kết quả chạy COM ảo

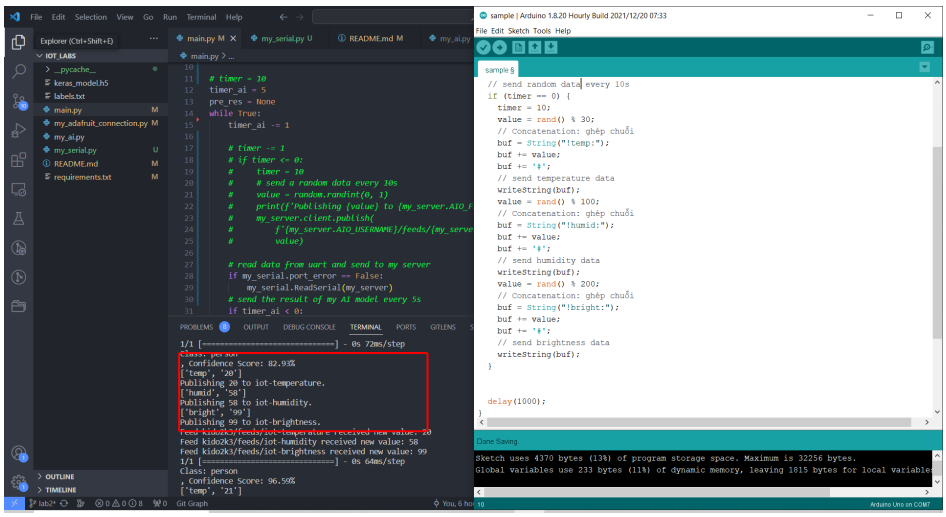
Hình 7: kết quả chạy COM ảo trên dashboard

Kết quả khi áp dụng vào hệ thống, sử dụng arduino

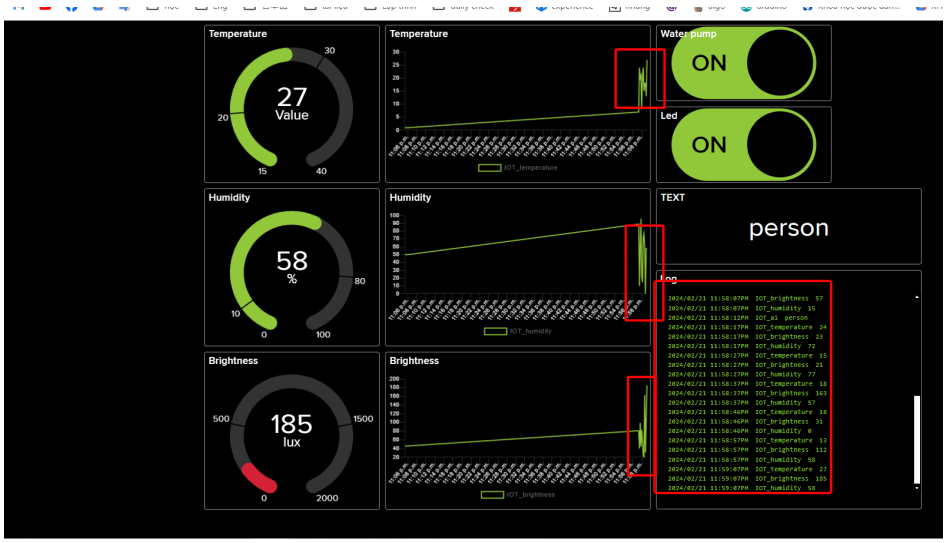


Hình 8: kết nối phần cứng

Arduino sẽ gửi ngẫu nhiên data mỗi 5s

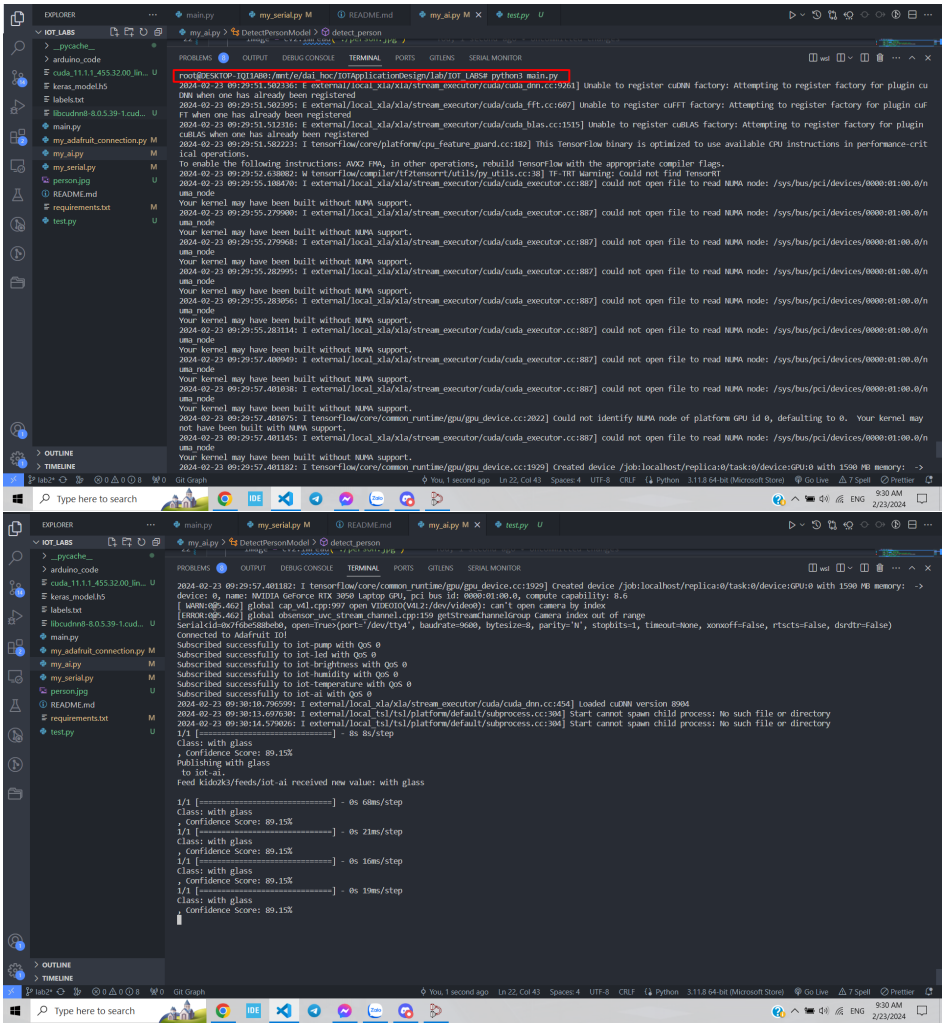


Hình 9: kết quả sử dụng arduino

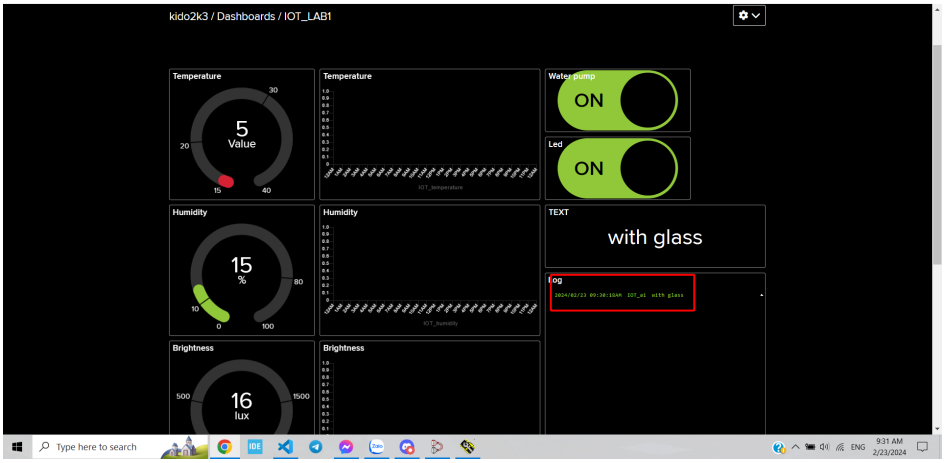


Hình 10: kết quả sử dụng arduino trên dashboard

5 Kết quả chạy trên linux



Hình 11: kết quả chạy trên linux



Hình 12: kết quả chạy trên linux (dashboard)