**Báo cáo thực tập**

**Kì 20223**

Đề tài:

**Lập trình gateway tương tác với camera, hỗ trợ điều khiển, lưu trữ và đưa video từ camera trong mạng LAN ra Internet**

|  |  |
| --- | --- |
| Họ và tên | MSSV |
| Hà Duy Long | 20204841 |

**Hà Nội, 2023**

[**Chương 1: Tổng quan đề tài** 3](#_Toc146620584)

[**1.** **Lý do chọn đề tài** 3](#_Toc146620585)

[**2.** **Nhiệm vụ** 3](#_Toc146620586)

[**3.** **Công nghệ sử dụng** 3](#_Toc146620587)

[**Chương 2: Nội dung công việc** 4](#_Toc146620588)

[**1.** **Cơ sở dữ liệu và cấu hình sqlalchemy** 4](#_Toc146620589)

[**1.1** **Table devices** 4](#_Toc146620590)

[**1.2** **Table records** 5](#_Toc146620591)

[**2.** **Tạo api bằng Fastapi** 6](#_Toc146620592)

[**3.** **Quét tìm camera trong mạng LAN** 6](#_Toc146620593)

[**4.** **Chuẩn ONVIF** 8](#_Toc146620594)

[**4.1 Thư viện python OnvifClient** 8](#_Toc146620595)

[**4.2 Chức năng điều khiển camera PTZ** 9](#_Toc146620596)

[**5.** **Chức năng record** 9](#_Toc146620597)

[**6.** **Stream video từ camera đến internet sử dụng webrtc** 10](#_Toc146620598)

[**6.1** **Truyền thông thời gian thực bằng webrtc.** 11](#_Toc146620599)

[**6.2** **Tạo chức năng stream video ra internet sử dụng aiortc và opencv** 12](#_Toc146620600)

[**Kết quả** 14](#_Toc146620601)

[**7.** **Xem lại records** 14](#_Toc146620602)

[**Chương 3: Tổng kết** 15](#_Toc146620603)

[**Kết quả:** 15](#_Toc146620604)

[**Hạn chế:** 15](#_Toc146620605)

[**Khó khăn**: 15](#_Toc146620606)

[**Hướng phát triển**: 15](#_Toc146620607)

[**Tài liệu tham khảo** 16](#_Toc146620608)

# **Chương 1: Tổng quan đề tài**

1. **Lý do chọn đề tài**

Hiện nay tồn tại nhiều sản phẩm camera của các nhà sản xuất khác nhau với các bộ chuẩn và các cấu hình hệ thống camera cũng có sự khác biệt. Thông thường phần mềm giám sát cài trên PC hoặc NVR sẽ phải cùng nhà sản xuất với Camera IP do sự chưa thống nhất trong các chuẩn sản xuất của các nhà cung cấp khác nhau. Từ năm 2008 chuẩn ONVIF ra đời và được xem là một tiêu chuẩn để các sản phẩm IP giám sát hình ảnh và các vùng giám sát an ninh khác có thể giao tiếp với nhau.

Trong đề tài này sẽ xây dựng chương trình và các API đơn giản bằng ngôn ngữ lập trình **Python** và cài đặt trên Gateway hỗ trợ việc điều khiển, lưu trữ, giám sát từ xa đối với camera hỗ trợ ONVIF.

**A diagram of a network

Description automatically generated**

**Hình 1 1Tổng quan hệ thống**

1. **Nhiệm vụ**

Xây dựng chương trình cài đặt trên Gateway với các chức năng:

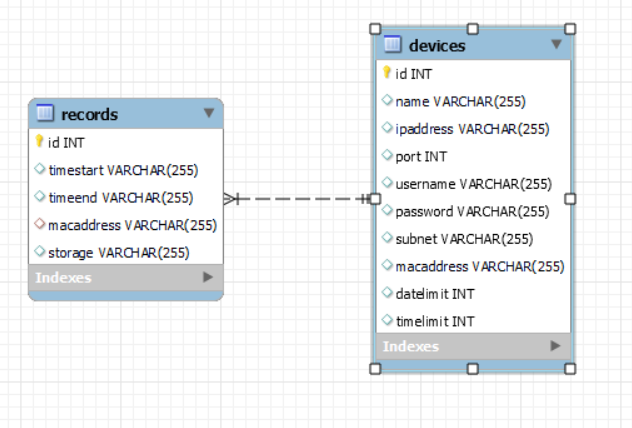
* Quét tìm kiếm các camera có trong mạng LAN
* Điều khiển camera (trái, phải, trên ……) với chuẩn onvif
* Tổ chức lưu trữ video của các camera trên Gateway
* Stream video nhận được từ camera tới **gateway** ra ngoài internet bằng **webrtc**

1. **Công nghệ sử dụng**

* Ngôn ngữ lập trình python, Fastapi
* Các thư viện python:
* Hỗ trợ ONVIF: WsDiscoveryClient, OnvifClient
* Hỗ trợ WebRTC: aiortc
* Object-Relational Mapping: sqlalchemy
* Html/javascript
* Hệ quản trị cơ sở dữ liệu Mysql

# **Chương 2: Nội dung công việc**

1. **Cơ sở dữ liệu và cấu hình sqlalchemy**



**Tổng quan cơ sở dữ liệu 1**

**SQLAlchemy**[[6]](https://www.sqlalchemy.org/) là một **thư viện Python** cung cấp một cơ sở dữ liệu đối tượng (**Object-Relational Mapping - ORM**) để tương tác với cơ sở dữ liệu quan hệ. ORM cho phép bạn làm việc với cơ sở dữ liệu bằng cách sử dụng các đối tượng Python thay vì viết trực tiếp các câu lệnh SQL.

Cài đặt: **pip install sqlalchemy**

Cấu hình

A screen shot of a computer program

Description automatically generated

* 1. **Table devices**

Bảng **devices** chứa các trường dữ liệu gồm:

* + name: tên camera do người dùng chọn
  + ipaddress: địa chỉ ip của camera
  + port: cổng kết nối
  + username
  + password
  + subnet: mặt nạ mạng
  + datelimit: số ngày lưu trữ tối đa
  + timelimit: thời gian lưu của một file (phút)

Đối tượng

A computer screen shot of text

Description automatically generated

* 1. **Table records**

Bảng **records** chứa các trường dữ liệu gồm:

* + timestart: thời gian bắt đầu của file ghi
  + timeend: thời gian kết thúc của file ghi
  + macaddress: địa chỉ mac của camera
  + storage: đường dẫn đến file lưu trữ

Đối tượng

A screen shot of a computer code

Description automatically generated



1. **Tạo api bằng Fastapi**

FastApi là 1 web framework dùng để build API có hiệu năng cao

Cài đặt: pip install fastapi

ASGI server khi deploy sản phẩm: pip install uvicorn

Run chương trình trên terminal sử dụng lệnh: uvicorn main:app –reload

A screen shot of a computer

Description automatically generated

Do based trên OpenAI mà trước đó có tên là Swagger nên FastAPI cung cấp doc có giao diện dễ nhìn, dễ sử dụng. Ví dụ minh họa:

Khi **bật doc** bằng local url [**http://0.0.0.0:8000/docs**](http://0.0.0.0:8000/docs).

1. **Quét tìm camera trong mạng LAN**

Các thiết bị onvif hỗ trợ WS-Discovery để tìm kiếm thiết bị onvif. Để xác định thiết bị onvif có tron mạng, người gửi tin nhắn thăm dò Probe. Các thiết bị phù hợp với các **types** và **scopes** trong tin nhắn tham dò sẽ phản hồi bằng cách gửi ProbeMatch. Sau đó có thể xác nhận được các thiết bị có trong mạng

A diagram of a problem

Description automatically generated

**tìm kiếm thiết bị onvif trong mạng LAN 1**

Cài đặt thư viện WsDiscoveryClient: pip install WSDiscovery

**Api: /device/scan**



Khi nhặn yêu cầu từ client, server thực hiện tìm kiếm các camera hỗ trợ onvif trong mạng LAN sau đó trả về danh sách thông tin các camera trong mạng LAN gồm địa chỉ ip, địa chỉ mac và port của từng camera.

A black screen with green dots

Description automatically generated

1. **Chuẩn ONVIF**

ONVIF (Open Network Video Interface Forum) là một tiêu chuẩn để các sản phẩm IP giám sát hình ảnh và các vùng giám sát an ninh khác có thể giao tiếp với nhau. Được thành lập năm 2008 bởi các thương hiệu lớn như Axis, Bosch, Sony,…Tiêu chuẩn giao tiếp giữa các thiết bị an ninh trên nền tảng IP.

Onvif định nghĩa các giao thức phù hợp với thiết bị bao gồm các dịch vụ

<https://www.onvif.org/profiles/specifications/>

### **4.1 Thư viện python OnvifClient**

Cài đặt yêu cầu:

* Python >= 3.9
* WSDiscovery >= 2.0.0
* onvif2-zeep >= 0.3.4

[pip install onvif-client](https://pypi.org/project/onvif-client/)

Lớp OnvifClient chứa một số phương thức đã được xây dựng sẵn để gọi đến các service của onvif. Ví dụ

A computer code with numbers and letters

Description automatically generated with medium confidence

Ngoài ra cách để tạo một service là sử dụng thuộc tính **\_onvif\_camera** của OnvifClient.

onvif\_client = OnvifClient('192.168.0.10', 80, 'user', 'password')

onvif\_camera = onvifclient.\_onvif\_camera

Với **devicemgmt** là một service chứa phương thức [**SetHostName**](https://www.onvif.org/onvif/ver10/device/wsdl/devicemgmt.wsdl#op.SetHostname)

Cách 1: Dict

**params = {‘Name’ : ‘NewHostName’}**

**onvif\_camera.SetHostName(params)**

Cách 2: tạo instance

**params = onvif\_camera.devicemgmt.createtype(‘SetHostName’)**

**params.Hostname = ‘NewHostName’**

**onvif\_camera.devicemgmt.SetHostName(params)**

### **4.2 Chức năng điều khiển camera** [**PTZ**](https://www.onvif.org/onvif/ver20/ptz/wsdl/ptz.wsdl)

Điều khiển camera xoay theo các hướng trái, phải, lên, xuống, zoom in, zoom out [[5]](https://www.onvif.org/wp-content/uploads/2016/12/ONVIF_WG-APG-Application_Programmers_Guide-1.pdf)

**Api: /devices/ptz/{ipaddress}/{move\_action}**

A screen shot of a computer code

Description automatically generated

Sử dụng hàm **moveAction** trong **threading** để không ảnh hưởng đến các luồng khác khi chay chương trình. Thực hiện kết nối đến camera sau đó thực thi các lệnh như up, down ….

1. **Chức năng record**

Thực hiện lưu video của camera vào Gateway, mỗi camera có thư mục và file video riêng cùng thời gian lưu trữ được xác định.

Cấu trúc thư mục: **../records/ten\_camera/ngay\_luu/timestart timeend.mp4**

Bắt đầu record: **/devices/record-thread/start/{ipaddress}**

Gửi yêu cầu record đến server, khởi tạo một **threading riêng** phục vụ **record** cho mỗi camera. Biến **threads** và **finish** là hai dictionaty lần lượt **đại diện** cho từng **theading record** và **trạng thái** của các threading

Hàm **record\_thread** được gọi đến và chạy trong một threading riêng thực hiện lần lượt các công việc:

* Tìm kiếm thông tin camera trong database, kết nối đến camera và lấy rtsp link
* Tạo thư mục lưu trữ cho camera, sử dụng thư viện **opencv** và rtsp url để tạo đối tượng cho việc lưu video
* Sử dụng ba vòng lặp lồng nhau.
  + Vòng lặp đầu tiên xác định ngày lưu và kiểm tra số ngày lưu liệu có vượt quá cấu hình hay không
  + Vòng lặp thứ hai thực hiện tạo file, lưu thông tin file vào database, kiểm tra xác định ngày mới
  + Vòng lặp thứ 3 thực hiện ghi video vào file

Kết quả khi record trên ba camera là dv0, dv1 và yootek03

A screenshot of a computer

Description automatically generated

1. **Stream video từ camera đến internet sử dụng webrtc** 
   1. **Truyền thông thời gian thực bằng webrtc.**

**WebRTC** (Web Real-Time Communication) là một dự án nguồn mở và miễn phí cung cấp trình duyệt web và ứng dụng di động với giao tiếp thời gian thực (RTC) thông qua giao diện lập trình ứng dụng (API). Nó cho phép giao tiếp âm thanh và video hoạt động bên trong các trang web bằng cách cho phép giao tiếp ngang hàng trực tiếp (**peer to peer**), loại bỏ sự cần thiết phải cài đặt các plugin hoặc tải xuống các ứng dụng gốc (native app).

Các **thành phần quan trọng** trong webrtc gồm:

* **MediaStream:** tream dữ liệu âm thanh, hình ảnh. Phát huy vai trò truy cập vào stream của thiết bị, xảy ra khi một kết nối webrtc được thiết lập với một thiết bị khác
  + Input: lấy dữ liệu hình ảnh âm thanh của thiết bị
  + Output: hiển thị dữ liệu lên view hoặc được RTCPeerConnection sử dụng
* **RTCDataChanel**: là một kênh hai chiều chịu trách nhiệm trao đổi dữ liệu thời gian thực. Trong đó sẽ tiến hành chuyển đổi dữ liệu dạng text, chia sẻ tệp P2P và các loại khác. Đây đều là những đối tượng không chứa dữ liệu nghe nhìn.
* **RTCPeerConnection:** đại diện cho kết nối webrtc giữa máy local và remote (peer to peer)

A computer chat box with blue text

Description automatically generated with medium confidence

Để có thể truyền dữ liệu, đầu tiên các peer cần:

* Khởi tạo đối tượng **PeerConnection** của mình.
* Các peer sau đó sẽ tạo các **local** **description** của riêng mình qua offer hoặc answer và gửi cho peer khác.
* Các peer nhận được offer hoặc answer sẽ sử dụng nó để tạo **remote description**. Sau đó tương tự cũng tạo **local description** cho chính nó và gửi lại cho peer khác.
* Các peer nhận lại được **description** của peer khác sẽ sử dụng nó để tạo **remote description**. Sau bước này tất cả cá peer đều có được description của nhau.
* Cuối cùng kệnh truyền được thiết lập và hai bên bắt đầu truyền dữ liệu.
  1. **Tạo chức năng stream video ra internet sử dụng aiortc và opencv**

[**Aiortc**](https://aiortc.readthedocs.io/en/latest/api.html) là thư viện WebRTC nguồn mở cho python với các thành phần quan trọng **RTCPeerConnection, MediaStreamTrack** giúp truyền video/audio thời gian thực. Cài đặt: pip install aiortc.

**Opencv** là một thư viện xử lý ảnh của python. Nó cung cấp phương thức cv2.VideoCapture(rtsp\_url) sẽ sử dụng để lấy stream từ camera:

A screen shot of a computer

Description automatically generated

Sau khi chạy chương trình và truy cập [**http://127.0.0.1:8000**](http://127.0.0.1:8000)brower sẽ load **scripts.js** và **index.js** hiện lên giao diện.

A screenshot of a computer

Description automatically generated

Sau khi nhập địa chỉ ip và **start**, hàm start tương ứng sẽ được thực thi [[3]](https://dev.to/whitphx/python-webrtc-basics-with-aiortc-48id). Hàm này sẽ tạo một **RTCPeerConnection** mới và đính kèm event listeners.

Tiếp theo hàm **negotiate()** được gọi đến. Đầu tiên pc.createOffer() để tạo offer. Tiếp theo tạo **local description** với **pc.setLocalDescription().** Sau đó tạo **request** tới **server main.py.**

**Phía server** sau khi nhận được request, thực hiện tìm thông tin của camera ip. Sau đó kết nối đến camera để lấy rtsp\_url.

Server tạo PeerConnection. Sử dụng lớp **CameraStreamTrack** kế thừa từ lớp **MediaStreamTrack** giúp tạo **mediastream** từ camera ip. Hàm **force\_codec** tạo codec

Tiếp theo tạo **remote description** bằng cách truyền offer đã nhận được vào **pc.setRemoteDescription()**. Sau đó tạo **answer** bằng **pc.createAnswer()** và tạo **local description** bằng **pc.setLocalDescription()**.

Cuối cùng answer được trả về cho client.js dưới dạng JSON của HTTP response.

Quay lại với **client** hàm **negotiate()** khi nhận lại được response từ server sẽ phân tích response và nhận được **answer**. Sau đó để tạo **remote description**, **answer** được truyền vào **pc.setRemoteDescription().** Kênh truyền được thiết lập giữ client và server.

Khi kết nối bị ngắt hàm **stop** ở client và **@pc.on("connectionstatechange")** ở server sẽ được gọi đến để đóng kết nối.

### **Kết quả**

A screenshot of a video

Description automatically generated

1. **Xem lại records**

Api: **/records/live/{macaddress}/{timestart}/{timeend}**

Tương tự như stream video từ camera, đầu tiên lấy từ database đường dẫn đến thư mục lưu trữ file. Lấy mediastream từ file bằng lớp **MediaPlayer** sau đó sử dụng **pc.addTrack()**

**Kết quả**

A group of people sitting at computers

Description automatically generated

# **Chương 3: Tổng kết**

## **Kết quả:**

* Các chức năng đề cập trong phần nhiệm vụ đều được hoàn thành.
* Các Api hoạt động và không phát sinh lỗi

## **Hạn chế:**

* Một số api chưa được tối ưu.
* Phần stream video khi kết nối thêm một camera dẫn đến phần stream của các camera khác bị delay.
* Hình ảnh hiển thị thỉnh thoảng bị mờ.
* Phần video khi xem lại record không tua được

## **Khó khăn**:

* Một số model camera không hỗ trợ hoàn toàn các service của onvif
* Aiortc có thể không hỗ trợ them nhiều media stream vào một peer connection.
* Không sử dụng đa luồng trong việc khởi tạo nhiều kênh truyền một lúc

## **Hướng phát triển**:

* Khắc phục hạn chế, cải tiến và tìm kiếm giải pháp cho việc tiếp nhận nhiều kết nối
* Tích hợp các công nghệ AI như nhận diện gương mặt, phát hiện chuyển động…
* Phát triển trên nền tảng web và mobile

# **Tài liệu tham khảo**

Source code <https://github.com/haduylong/thuc_tap_20223>

[1] Onvif <https://www.onvif.org/profiles/specifications/>

[2] Aiortc <https://aiortc.readthedocs.io/en/latest/api.html>

[3] Basic aiortc example <https://dev.to/whitphx/python-webrtc-basics-with-aiortc-48id>

[4] Webrtc <https://webrtc.org/>

[5] Lập trình onvif <https://www.onvif.org/wp-content/uploads/2016/12/ONVIF_WG-APG-Application_Programmers_Guide-1.pdf>

[6] sqlalchemy <https://www.sqlalchemy.org/>