**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**

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:

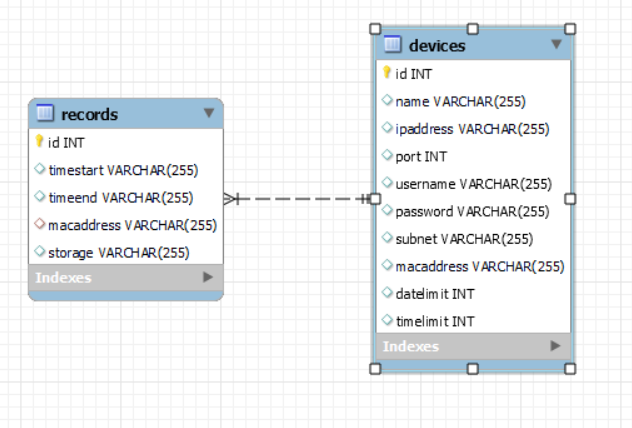
* Quét tìm kiếm các camera có trong mạng LAN
* Điều khiển camera (trái, phải, trên ……) với chuẩn onvif
* Tổ chức lưu trữ video của các camera trên Gateway
* Stream video nhận được từ camera tới **gateway** ra ngoài internet bằng **webrtc**

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** 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**

Cài đặt thư viện: WsDiscoveryClient

**Api: /device/scan**

A screen shot of a computer code

Description automatically generated

Hàm **scanDevice()** tìm ra địa chỉ **ip** và **port** sau đó gọi đến hàm **get\_mac\_address** để tìm địa chỉa **mac** tương ứng với địa chỉ ip

A screen shot of a computer program

Description automatically generated



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

**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 ….

A screen shot of a computer program

Description automatically generated

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

A screen shot of a computer program

Description automatically generated

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

A screen shot of a computer code

Description automatically generated

* Tạo thư mục lưu trữ cho camera, sử dụng thư viện **opencv** và rtsp url để tạo đối tượng cho việc lưu video

A screen shot of a computer program

Description automatically generated

* 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

A screen shot of a computer screen

Description automatically generated

* Vòng lặp thứ hai thực hiện tạo file, lưu thông tin file vào database

A screen shot of a computer code

Description automatically generated

* Vòng lặp thứ 3 thực hiện ghi video vào file

A screen shot of a computer screen

Description automatically generated

1. **Strem video từ camera đến internet sử dụng webrtc**

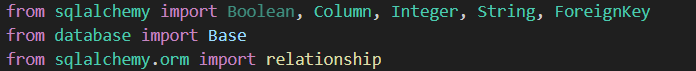
Tổng quan

Tạo cơ sở dữ liệu với Mysql và sqlalchemy

Tạo các entity với sqlalchemy

Cài đặt: pip install sqlalchemy

Trong file model.py tạo các entity

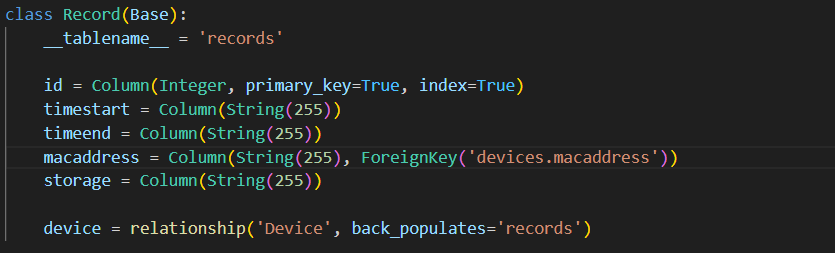


Entity device chứa thông tin camera

A computer screen shot of text

Description automatically generated

Entity record chứa thông tin các file video của camera



Tạo api với python sử dụng FastApi, uvicorn

Cài đặt: pip install fastapi, uvicorn

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

A screen shot of a computer

Description automatically generated

Test api: **http://127.0.0.1:8000/docs**

Tạo api chức năng quét tìm các camera trong mạng lan

Cài đặt thư viện: WsDiscoveryClient

A screen shot of a computer

Description automatically generated

Hàm scanDevice trả về danh sách các camera gồm thông tin về địa chỉ ip, địa chỉ mac, port.

A black screen with green dots

Description automatically generated

Tạo các api thao tác với camera hỗ trợ chuẩn onvif

Cài đặt thư viện python-onvif

Điều khiển camera

A screen shot of a computer code

Description automatically generated

Hàm moveAction thực hiện điều khiển camera lên xuống, xoay trái phải và zoom

A screen shot of a computer code

Description automatically generated

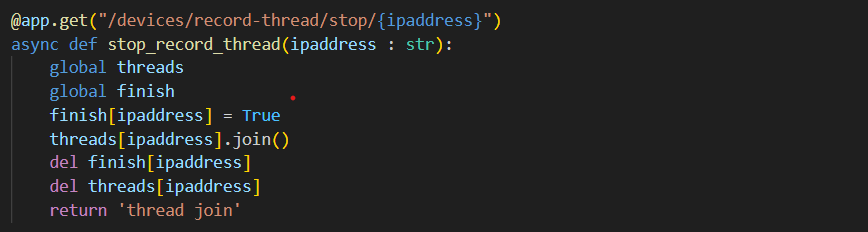
Tạo api thực hiện lưu video từ camera

A screen shot of a computer program

Description automatically generated

Thực hiện hàm **recoding\_thread** với threading

Kết thúc record video



Tạo api hỗ trợ stream video từ mạng LAN đến Internet

Sử dụng webrtc và opencv

Cài đặt thư viện **aiortc** là một thư viện python hỗ trợ webrtc

Lấy thông tin camera và tạo rtsp url giúp thiết lập kết nối tới camera

A computer screen with many colorful text

Description automatically generated

Từ client.js gửi yêu cầu thiết lập kết nối A screen shot of a computer screen

Description automatically generated

Ở trên client tạo offer chứa thông tin peer của client gửi đến server bằng request và sau khi nhận được server phân tích tạo answer đến client. Sau đó kênh truyền giữa server và client được thiết lập

A screen shot of a computer program

Description automatically generated

Tạo lớp **CameraVideoTrack** kế thừa từ lớp **VideoStreamTrack**. Lớp này sử dụng opencv với rtsp url đã lấy được để nhận video từ camera và đưa đến kênh tuyền dữ liệu

A screen shot of a computer program

Description automatically generated

Test

A screenshot of a video

Description automatically generated