

**CÔNG TY TNHH CÔNG NGHỆ A.I**

**HƯỚNG DẪN SỬ DỤNG**  
**AI.ACCR SDK**

# MỤC LỤC

<b>1. Giới thiệu.....</b>	<b>1</b>
<b>2. Các lưu ý quan trọng.....</b>	<b>2</b>
2.1 Điểm ảnh (pixel):.....	2
2.2 Đường link RTSP của camera .....	2
2.3 Tài nguyên máy tính khi giải mã video từ camera .....	3
2.4 Mã hóa Base64. ....	3
<b>3. Cài đặt .....</b>	<b>4</b>
<b>4. License.....</b>	<b>5</b>
4.1 License.....	5
4.2 Activate.....	6
<b>5. Tích hợp AI.ACCR SDK vào phần mềm .....</b>	<b>7</b>
5.1 Quy trình khởi tạo.....	8
5.2 Quy trình kết nối camera IP.....	10
5.3 Quy trình xử lý nhận dạng số container.....	11
5.4 Quy trình xử lý hình ảnh.....	12
<b>6. Các “Phương thức” sử dụng .....</b>	<b>12</b>
6.1 initEvent.....	12
6.2 setIndexRtspUrl .....	13
6.3 setIndexControlDisplay .....	13
6.4 setDisplay .....	13
6.5 play .....	13
6.6 setCameraEventMethod.....	14
6.7 cameraCallback .....	14
6.8 getContainerCode .....	14
6.9 checkDigitContainerCode .....	15
6.10 calculationDigitContainerCode .....	15
6.11 saveBitmap .....	15
6.12 saveJpg.....	15
6.13 displayImage.....	16
6.14 loadImageToImgBase64.....	16
6.15 getPlayStatus .....	16
6.16 getImageBase64.....	17
6.17 Dispose .....	17
<b>7. Thông tin công ty TNHH Công nghệ A.I .....</b>	<b>17</b>

## 1. Giới thiệu

AI.ACCR là một SDK về nhận dạng số container, là một công cụ hiệu quả giúp tách ký tự số container từ hình ảnh hoặc từ luồng dữ liệu RTSP (dữ liệu camera), được thiết kế để nhà phát triển có thể tích hợp cho các dự án như: kiểm soát container cảng biển, ...

Sử dụng các giải thuật và mô hình học máy trong trí tuệ nhân tạo, AI.ACCR giúp tối đa hóa tính năng tính toán và cải thiện hiệu suất xử lý. Chức năng chính của Module bao gồm:

- Kết nối camera IP hiển thị tín hiệu video lên control của phần mềm chính.
- Chụp ảnh xe từ camera IP, tải hình ảnh từ đĩa, lưu hình ảnh.
- Tách số container từ hình ảnh thành dạng ký tự.

AI.ACCR được chia thành ba phiên bản chính để phù hợp với các yêu cầu khác nhau của khách hàng:

- Phiên bản đầu tiên chỉ sử dụng CPU
- Phiên bản thứ hai sử dụng GPU Intel từ thế hệ 6 trở lên.
- Phiên bản thứ ba sử dụng GPU Nvidia hỗ trợ "Compute Capability" phiên bản 6.1 trở lên.

Với mỗi phiên bản, chúng tôi đã tối ưu hóa tính năng tính toán để đạt được hiệu suất tối đa.

AI.ACCR chỉ hỗ trợ ứng dụng 64 bit để đảm bảo sự tương thích với các thế hệ CPU mới nhất. Để đảm bảo tương thích máy tính xử lý phải cài sẵn trình điều khiển mới nhất và cài đặt sẵn các thành phần cần thiết như NET framework 4.8 và C++ 2015-2019.

AI.ACCR hỗ trợ đa dạng các ngôn ngữ lập trình như c#, vb.net, c++, powerbuilder, java, python ... giúp dễ dàng tích hợp vào các ứng dụng khác.

AI.ACCR SDK này không được thiết kế để sử dụng cho người dùng cuối, mà chỉ dành cho các nhà phát triển và các doanh nghiệp muốn tích hợp tính năng này vào ứng dụng của mình.

Nếu có yêu cầu đặc biệt khác, như yêu cầu tùy chỉnh hoặc tích hợp vào phần mềm sẵn có, người dùng có thể liên hệ trực tiếp với chúng tôi để được tư vấn và hỗ trợ

## 2. Các lưu ý quan trọng

### 2.1 Điểm ảnh (pixel):

Pixel (viết tắt của picture element) là đơn vị nhỏ nhất của hình ảnh kỹ thuật số, chúng là các điểm ảnh nhỏ nhất được sắp xếp theo một mạng lưới trên màn hình hoặc trên các thiết bị lưu trữ hình ảnh. Pixel có kích thước nhỏ hơn 1/1000 inch, và được sử dụng để tạo ra các hình ảnh số, bao gồm các bức ảnh, biểu đồ, logo, hoặc các hình ảnh khác.

Mỗi pixel được mô tả bằng một số, đại diện cho màu sắc và độ sáng của điểm ảnh đó. Số lượng pixel càng nhiều, hình ảnh sẽ càng chi tiết và sắc nét hơn.

Ví dụ, một bức ảnh với độ phân giải 1024x768 có nghĩa là chiều rộng của nó là 1024 pixel và chiều cao là 768 pixel. Tổng số pixel của ảnh này là  $1024 \times 768 = 786,432$  pixel. Mỗi pixel trong bức ảnh sẽ được mô tả bằng một giá trị RGB (Red-Green-Blue) cho biết màu sắc của điểm ảnh đó..

Cùng kích thước vật lý, độ phân giải cao tức là số lượng pixel nhiều và kích thước từng pixel sẽ nhỏ hơn là độ phân giải thấp tức là có ít pixel hơn và chúng lại có kích thước lớn hơn.

### 2.2 Đường link RTSP của camera

RTSP là viết tắt của Real-Time Streaming Protocol, một giao thức truyền phát video trực tiếp trên mạng. RTSP được sử dụng trong nhiều ứng dụng, bao gồm cả trong camera giám sát, phần lớn camera IP trên thị trường đều hỗ trợ giao thức RTSP.

Lấy đường link RTSP của camera bằng cách đọc tài liệu hướng dẫn sử dụng của nhà sản xuất hoặc sử dụng phần mềm ONVIF Device Manager.

Đường link RTSP tiêu chuẩn thường có định dạng như sau:

**rtsp://<username>:<password>@<camera\_IP\_address>:<port>/<stream\_name>**

Trong đó:

**<username>**: Tên đăng nhập vào camera.

**<password>**: Mật khẩu để đăng nhập vào camera.

**<camera\_IP>:\_address>**: Địa chỉ IP của camera trên mạng.

<port>: Cổng được sử dụng để truyền tải video.

<stream\_name>: Tên của luồng video được truyền tải từ camera.

Ví dụ, nếu tên đăng nhập của camera là "admin", mật khẩu là "12345", địa chỉ IP của camera là "192.168.1.10", cổng RTSP là 554 và tên của luồng video là "live", thì đường link RTSP sẽ có dạng như sau:

**rtsp://admin:12345@192.168.1.10:554/live**

### 2.3 Tài nguyên máy tính khi giải mã video từ camera

Các thông số của camera IP có thể ảnh hưởng đến tài nguyên (CPU, RAM, Băng thông,...) của máy tính khi giải mã video của camera, cụ thể như sau:

- Độ phân giải: Độ phân giải của video càng cao thì càng cần nhiều tài nguyên của máy tính để giải mã và xử lý. Nếu máy tính không đủ mạnh, thì video có thể bị giật hoặc chậm.
- Tốc độ khung hình (Frame rate): Nếu tốc độ khung hình cao, thì cần nhiều tài nguyên để xử lý.
- Chuẩn nén: Các định dạng nén video khác nhau có độ phức tạp khác nhau và yêu cầu nhiều hoặc ít tài nguyên để giải mã. Các chuẩn nén từ thấp đến cao cần tài nguyên của máy tính: MJPEG -> H264 -> H265
- Độ trễ: Nếu độ trễ cao, thì máy tính cần nhiều tài nguyên để giải mã và hiển thị video một cách liền mạch.

Các thông số của camera IP có thể ảnh hưởng đến tài nguyên của máy tính khi giải mã video của camera, do đó cần đánh giá kỹ các thông số này để chọn một máy tính có đủ tài nguyên để xử lý video của camera một cách mượt mà và ổn định.

Trong trường hợp không cần tốc độ cao thì có thể sử dụng cấu hình camera tiêu chuẩn sau:

- Độ phân giải: 1280x720
- Tốc độ khung hình: 6 - 12 frame.
- Chuẩn nén: H264

### 2.4 Mã hóa Base64.

Base64 là một phương thức mã hóa 2 chiều từ tập tin nhị phân (binary) sang chuỗi ký tự (string) và ngược lại, nó thường được sử dụng để mã hóa các tập tin đa phương tiện (hình ảnh, âm thanh, video,...). Ký tự 64 trong Base64 là đại diện cho 64 ký tự trong bảng mã ASCII. Base64 thường được sử dụng trong việc truyền tải email. Tuy nhiên, chúng tôi dùng

Base64 để mã hóa hình ảnh phục vụ mục đích sử dụng dữ liệu hình ảnh giữa các ngôn ngữ lập trình được đơn giản nhất.

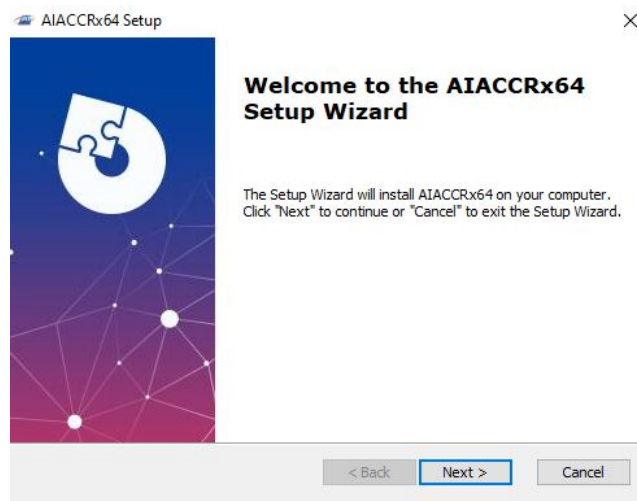
### 3. Cài đặt

Chọn tệp AIACCRx64.exe nhấn ENTER hoặc đúp vào file để chạy cài đặt.

Màn hình xuất hiện :

Nhấn “Next” để tiếp tục cài đặt hoặc nhấn “Cancel” để hủy thao tác cài đặt.

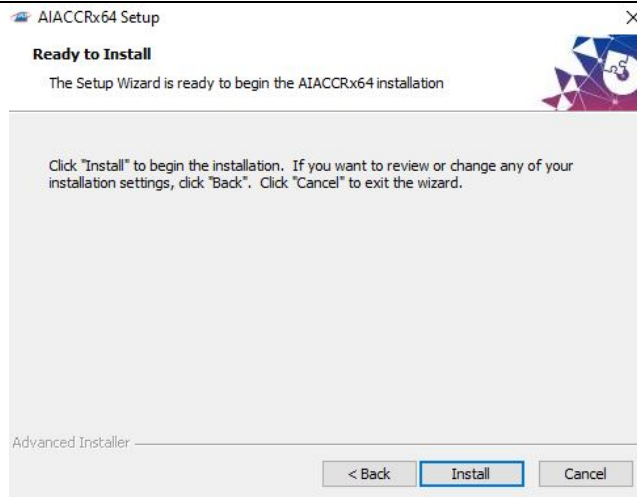
Tiếp đến sẽ hiển thị 1 màn hình mới:



Folder: Chương trình mặc định cài trên thư mục “C:\Program Files (x86)\AITECH\AIACCRx64” nếu bạn muốn thay đổi thư mục khác thì nhấn “Browse” chọn đến thư mục cần cài đặt.

Nhấn nút “Next” để tiếp tục cài đặt.

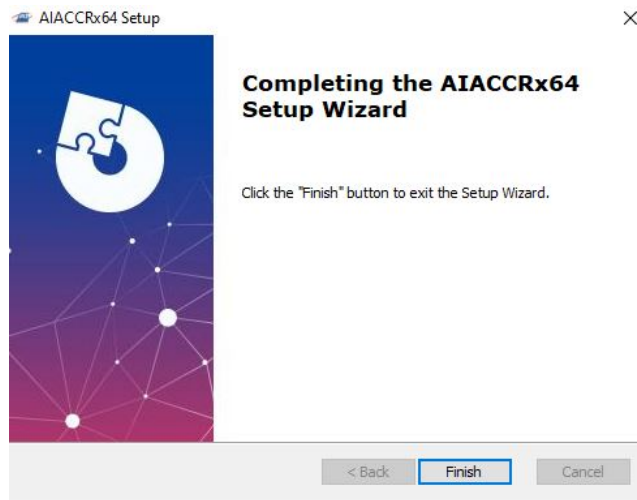
Tiếp đến sẽ hiển thị màn hình chấp nhận cài đặt:



Nhấn “Install” để cài đặt.

Chương trình đang chạy tiến trình cài đặt.

Khi chạy xong nó sẽ hiện thông báo tiến trình cài đặt thành công như hình.



Nhấn “Finish” kết thúc tiến trình cài đặt.

## 4. License.

### 4.1 License

License chúng tôi cung cấp là: License phần mềm theo máy tính (còn gọi là "perpetual license") là một loại giấy phép mà khi người sử dụng mua một phần mềm thì sẽ được sử dụng phần mềm đó vĩnh viễn trên một máy tính cụ thể mà họ đã mua giấy phép.

Nói cách khác, người sử dụng không cần phải trả thêm phí để sử dụng phần mềm trên máy tính đó trong tương lai, trừ khi họ muốn nâng cấp phiên bản phần mềm hoặc mua một giấy phép khác để sử dụng trên một máy tính khác.

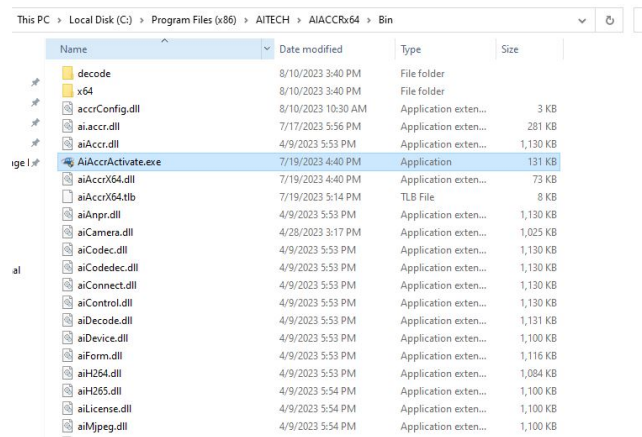
Hiện tại, chúng tôi cấp license trên website <https://aitech.com.vn/> hoặc có thể liên hệ trực tiếp để được tư vấn cụ thể hơn.

## 4.2 Activate.

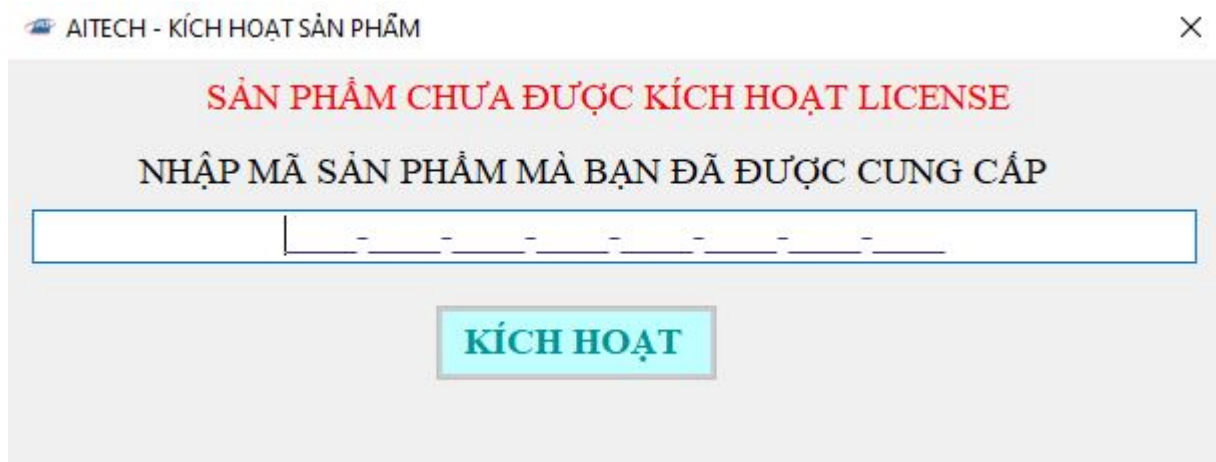
### - Bước 1:

Chạy chương trình “**AiAccrActivate.exe**” trong thư mục cài đặt

**C:\Program Files (x86)\AITECH\AIACCRx64\Bin\AiAccrActivate.exe**

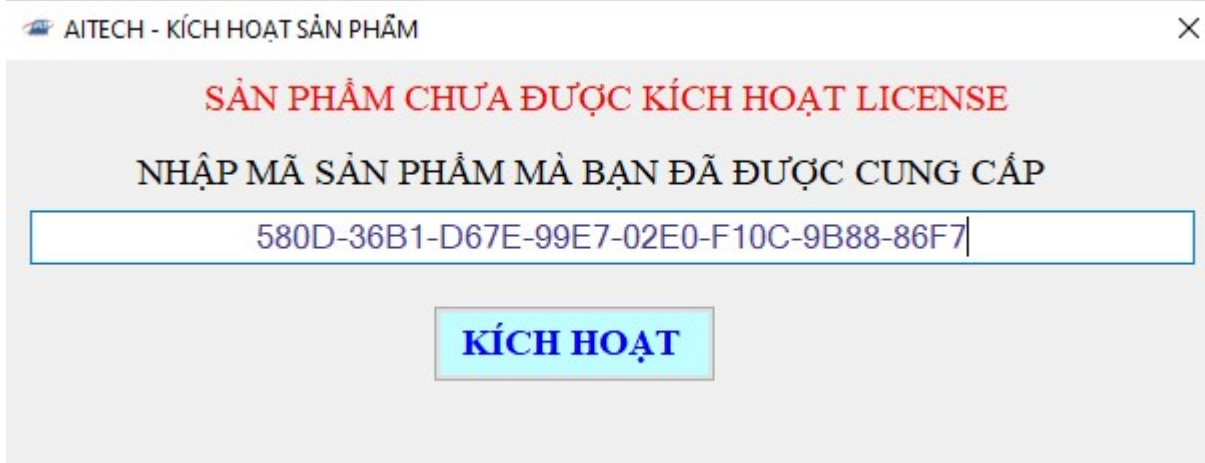


### - Bước 2:



Nhập mã sản phẩm mà bạn được cung cấp.





- **Bước 3:**

“KÍCH HOẠT”



Chúng tôi có cung cấp license dùng thử trên website <https://aitech.com.vn/> , bạn có thể đăng ký dùng thử.

### 5. Tích hợp AI.ACCR SDK vào phần mềm

AI.ACCR SDK sau khi cài đặt là một tập hợp các giao diện lập trình ứng dụng API COM cho phép các ứng dụng Windows truy cập vào các tính năng của module để sử dụng.

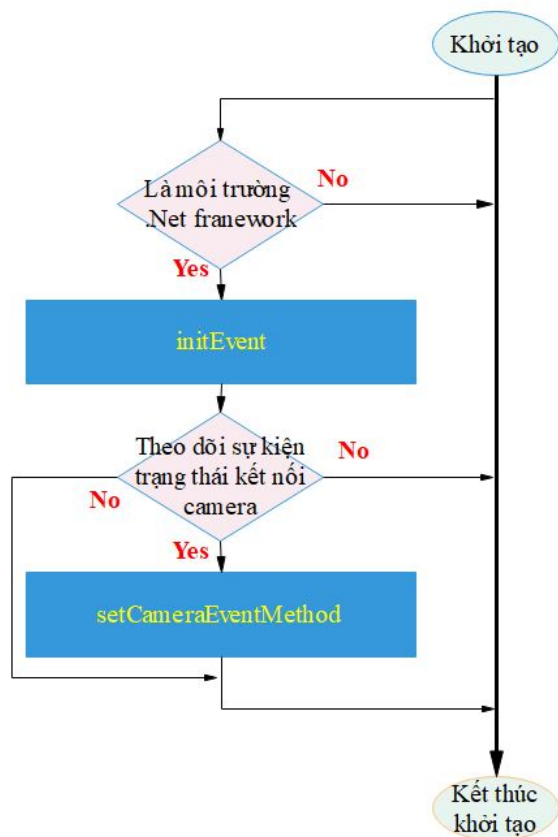
Sau khi cài đặt và kích hoạt license, chúng ta có thể dùng thử bằng phần mềm chạy thử với hình ảnh và đường link RTSP cung cấp sẵn để sử dụng trong quá trình tích hợp.

Module được cài đặt kèm theo mã nguồn tích hợp của một số ngôn ngữ lập trình thông dụng tại thư mục cài đặt: **C:\Program Files (x86)\AITECH\AIACCRx64\DemoSourceCode**

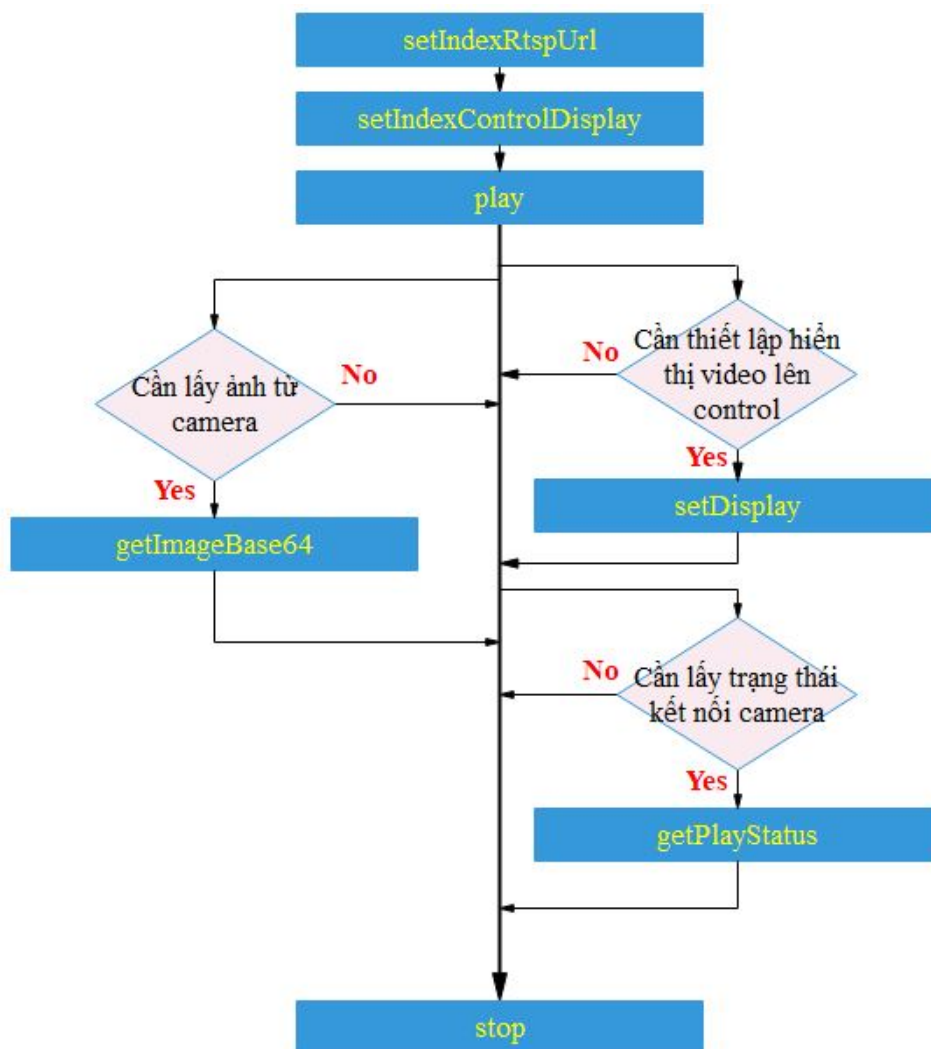
## 5.1 Quy trình khởi tạo

Đối với ngôn ngữ lập trình khác nhau thì cách khởi tạo kết nối AP COM cũng khác nhau, các khởi tạo ngôn ngữ lập trình thông dụng.

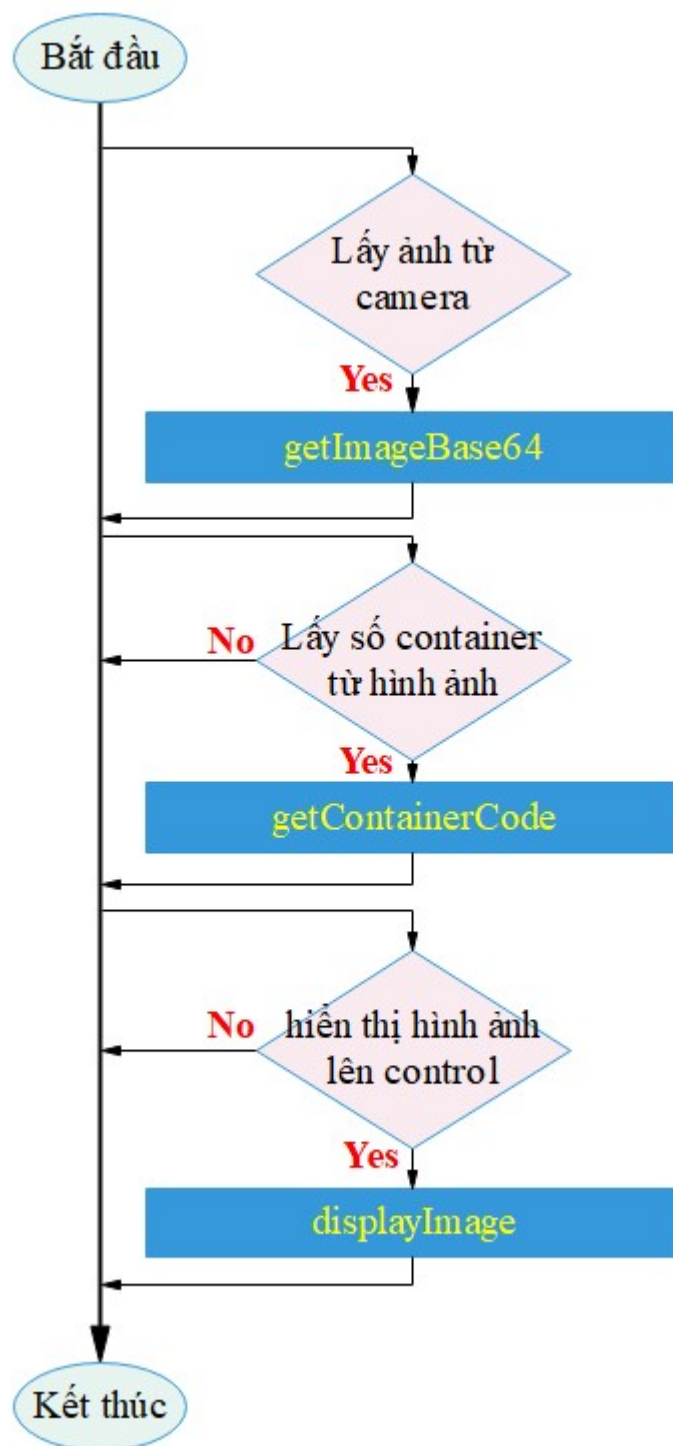
Ngôn ngữ lập trình	Khởi tạo API COM
C#	<pre>Type AccrCom dynamic aiAccr AccrCom = Type.GetTypeFromProgID("Aitech.ClsAiAccrX64") aiAccr = Activator.CreateInstance(AccrCom, false)</pre>
VB.Net	<pre>Dim AccrCom As Type Dim aiAccr As Object AccrCom = Type.GetTypeFromProgID("Aitech.ClsAiAccrX64") aiAccr = Activator.CreateInstance(AccrCom, False)</pre>
C++	<pre>#import " C:\Program Files (x86)\AITECH\AIACCRx64\Bin\aiAccrX64.tlb " ClsAiAccrX64::IClsAiAccrX64Ptr aiaccr = nullptr SUCCEEDED(CoInitialize(0)) SUCCEEDED(aiAccr.CreateInstance("Aitech.ClsAiAccrX64"))</pre>
Python	<pre>import win32com.client aiAccr = win32com.client.Dispatch("Aitech.ClsAiAccrX64")</pre>



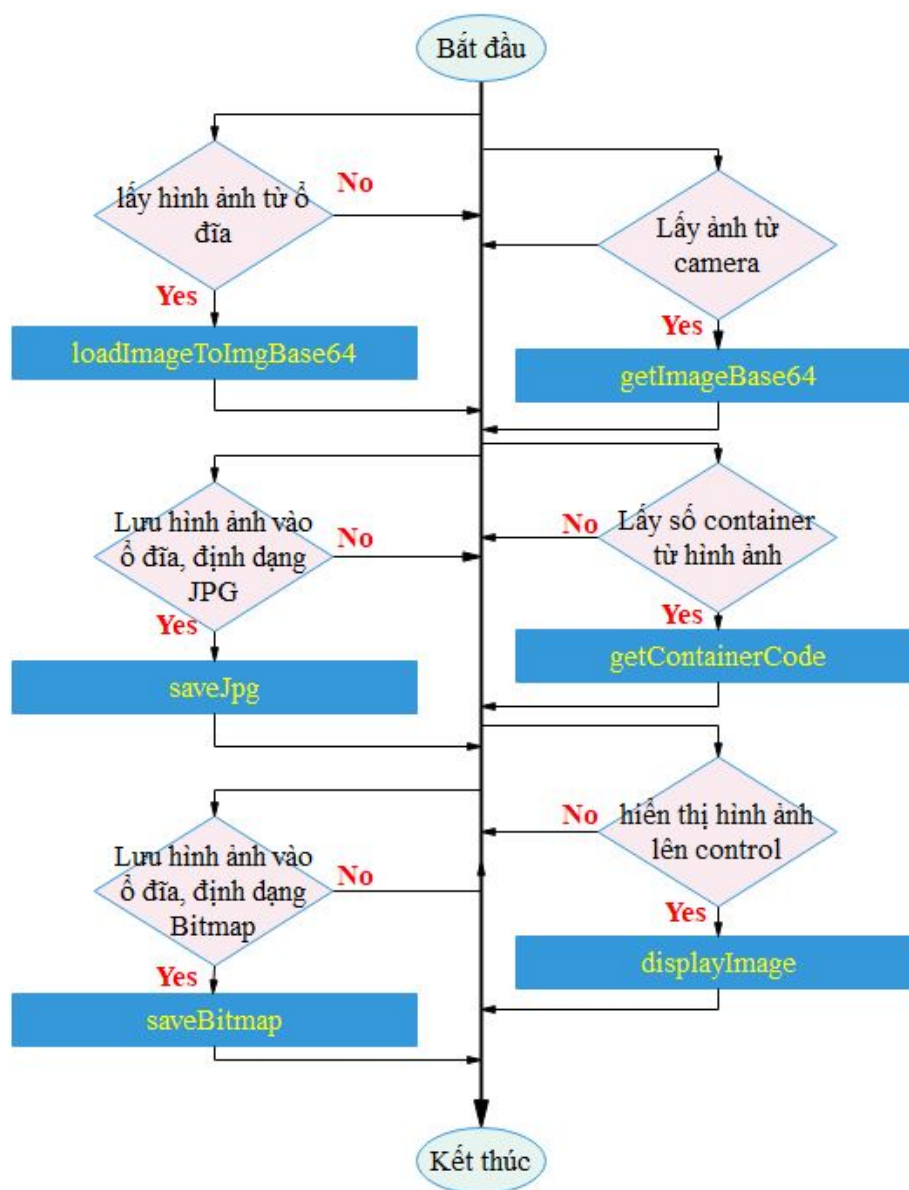
## 5.2 Quy trình kết nối camera IP



### 5.3 Quy trình xử lý nhận dạng số container



## 5.4 Quy trình xử lý hình ảnh.



## 6. Các “Phương thức” sử dụng

### 6.1 initEvent

- Cấu trúc:

`void initEvent()`

- Tham số: Không

- Trả về: Không

- Kết quả: Khởi tạo môi trường nhận trạng thái kể nối camera (online, offline).

## 6.2 setIndexRtspUrl

- Cấu trúc:

```
void setIndexRtspUrl(int cameraIndex, string rtspUrl)
```

- Tham số:
  - o cameraIndex: Số thứ tự camera
  - o rtspUrl: Đường link RTSP của camera
- Trả về: Không
- Kết quả: đường link RTSP **rtspUrl** của camera vào số thứ tự **CameraIndex**

## 6.3 setIndexControlDisplay

- Cấu trúc:

```
void setIndexControlDisplay(int cameraIndex, int displayControlHandle)
```

- Tham số:
  - o cameraIndex: Số thứ tự camera
  - o displayControlHandle: Handle của control hiển thị video
- Trả về: Không
- Kết quả: Video camera sẽ hiển thị lên control

## 6.4 setDisplay

- Cấu trúc: `void setDisplay(int cameraIndex, Boolean displayValue)`
- Tham số:
  - o cameraIndex: Số thứ tự camera
  - o displayValue: Trạng thái muốn hiển thị hay không (true, false)
- Trả về: Không
- Kết quả: Hiển thị hoặc không hiển thị video lên control

## 6.5 play

- Cấu trúc:

```
void play(int cameraIndex)
```

- Tham số:
  - o cameraIndex: Số thứ tự camera

- Trả về: Không
- Kết quả: Bắt đầu giải mã tín hiệu video tại vị trí **CameraIndex**

## 6.6 setCameraEventMethod

- Cấu trúc:

**void setCameraEventMethod**(**object** consumer, **string** eventName)

- Tham số:
  - Consumer: Đối tượng chứa “phương thức” eventName( This (c#), Me (vb.net)..)
  - EventName: Tên “phương thức” (callback ) nhận kết quả. Cấu trúc của “Phương thức” nhận kết quả: **void cameraCallback** (**int** cameraIndex, **Boolean** playStatus)
- Trả về: Không
- Kết quả: Đăng ký “phương thức” nhận kết quả trả về trạng thái kết nối camera.

## 6.7 cameraCallback

- Cấu trúc:

**void cameraCallback** (**int** cameraIndex, **Boolean** playStatus)

- Tham số:
  - cameraIndex: Thứ tự camera
  - playStatus: Trạng thái camera
- Trả về: Không
- Kết quả: Sự kiện trả về trạng thái camera. **cameraCallback** ( Có thể đổi tên phương thức).

## 6.8 getContainerCode

- Cấu trúc:

**string** getContainerCode(**string** imageBase64)

- Tham số:
  - imageBase64: Hình ảnh dạng base64 đưa vào để nhận dạng số container.
- Trả về: Kiểu **string** là số container trả về
- Kết quả: Lấy được số container.



## 6.9 checkDigitContainerCode

- Cấu trúc:

**Boolean** checkDigitContainerCode(**string** containercode)

- Tham số:
  - o containercode: Số container cần kiểm tra.
- Trả về: Kiểu **Boolean** là số container có theo tiêu chuẩn ISO 6346 hay không?
- Kết quả: Cấu trúc số container đã nhận dạng được có theo tiêu chuẩn ISO 6346 hay không.

## 6.10 calculationDigitContainerCode

- Cấu trúc:

**string** calculationDigitContainerCode(**string** containercode)

- Tham số:
  - o containercode: Số container cần kiểm tra.
- Trả về: Kiểu **string** là Chữ số kiểm tra (Check digit) của số container?
- Kết quả: Lấy Chữ số kiểm tra (Check digit), được sử dụng để kiểm tra tính hợp lệ của toàn bộ mã container. Đây là một chữ số, thường là chữ số cuối cùng của số container.

## 6.11 saveBitmap

- Cấu trúc:

**Boolean** saveBitmap(**string** imageBase64, **string** fullFileName)

- Tham số:
  - o imageBase64: Hình ảnh dạng base 64
  - o fullFileName: đường dẫn muốn lưu hình
- Trả về: kiểu **Boolean** lưu hình thành công hay thất bại
- Kết quả: Lưu hình ảnh dạng **Bitmap** vào ổ đĩa cứng, nếu tên hình bị trùng sẽ xóa file cũ và lưu hình mới.

## 6.12 saveJpg

- Cấu trúc:

**Boolean** saveJpg(**string** imageBase64, **string** fullFileName)

- Tham số:
  - o imageBase64: Hình ảnh dạng base64
  - o fullFileName: đường dẫn lưu hình
- Trả về: kiểu **Boolean** lưu hình thành công hay thất bại
- Kết quả: Lưu hình ảnh dạng **JPG** vào ổ đĩa cứng, nếu tên hình bị trùng sẽ xóa file cũ và lưu hình mới.

### 6.13 displayImage

- Cấu trúc:  
**Boolean saveBitmap(string imageBase64, string fullFileName)**
- Tham số:
  - o imageBase64: Hình ảnh dạng base 64
  - o fullFileName: đường dẫn muốn lưu hình
- Trả về: Kiểu **Boolean** lưu hình thành công hay thất bại
- Kết quả: Lưu hình ảnh dạng **Bitmap** vào ổ đĩa cứng, nếu tên hình bị trùng sẽ xóa file cũ và lưu hình mới.

### 6.14 loadImageToImgBase64

- Cấu trúc:  
**string loadImageToImgBase64(string fullFileName)**
- Tham số:
  - o fullFileName: Đường dẫn hình ảnh trong có trong ổ cứng
- Trả về: Kết quả kiểu **string** là hình ảnh dạng base64
- Kết quả: Tải hình ảnh lưu từ ổ đĩa thành hình ảnh dạng Base64

### 6.15 getPlayStatus

- Cấu trúc:  
**Boolean getPlayStatus(int cameraIndex)**
- Tham số:
  - o CameraIndex: Số thứ tự camera
- Trả về: Kiểu **Boolean** là trạng thái của camera

- Kết quả: Trả về trạng thái kết nối camera

### 6.16 getImageBase64

- Cấu trúc:

`string getImageBase64(int cameraIndex)`

- Tham số:

- o cameraIndex: Số thứ tự camera

- Trả về: Kiểu `string` là hình ảnh dạng base64

- Kết quả: Lấy hình ảnh dạng Base64 từ camera có vị trí CameraIndex

### 6.17 Dispose

- Cấu trúc:

`void dispose()`

- Tham số: Không

- Trả về: Không

- Kết quả: Dừng tất cả camera đang hoạt động

## 7. Thông tin công ty TNHH Công nghệ A.I

Công ty TNHH CÔNG NGHỆ A.I

Địa chỉ: Số 425 Đường TCH 21, Phường Tân Chánh Hiệp, Quận 12, Tp. HCM

Điện thoại: 0286 250 7888

Hotline: 0914452090

Email: [info@aitech.com.vn](mailto:info@aitech.com.vn)

Website: <https://aitech.com.vn>

Fanpage: <https://www.facebook.com/SoftwareComputervision>