

CÔNG TY TNHH CÔNG NGHỆ A.I

**HƯỚNG DẪN SỬ DỤNG
MODULE AI.CAMERA**

MỤC LỤC

1. Giới thiệu	2
2. Lưu ý quan trọng	2
2.1 Tài nguyên máy tính khi giải mã video từ camera	2
2.2 Đường link RTSP của camera	3
2.3 Mã hóa Base64	3
3. Cài đặt chương trình	4
4. License.....	6
4.1 Activate	6
4.2 License	8
5. Tích hợp module AI.CAMERA vào phần mềm.....	9
5.1 Khởi tạo.....	9
5.2 “Phương thức” khởi tạo.....	11
5.3 Các “Phương thức” tương tác với camera.....	12
5.4 Các “Phương thức tương tác với hình ảnh”	13
5.5 Các bước thao tác vẽ.	14
6. Các “Phương thức” sử dụng.....	18
6.1 initEvent	18
6.2 setCameraEventMethod	18
6.3 setIndexRtspUrl.....	18
6.4 setIndexControlDisplay.....	18
6.5 play	19
6.6 setDisplay	19
6.7 stop	19
6.8 getPlayStatus	19
6.9 getImageBase64	20
6.10 displayImage	20
6.11 saveBitmap.....	20
6.12 saveJpg	21
6.13 loadImageToImgBase64	21
6.14 dispose.....	21
6.15 eventName.....	21
6.16 addPolygon.....	22
6.17 addPolygonPoint	22
6.18 addFillPolygon	22
6.19 addFillPolygonPoint.....	23
6.20 addString	23
6.21 addLine.....	24
6.22 addFillRectangle.....	24
6.23 addRectangle	25
6.24 addEllipse	25
6.25 addFillEllipse	26
6.26 addImageBase64	26
6.27 setDrawToVideo	27

6.28	drawToImage	27
6.29	getFrameCounter	27
6.30	reloadDrawImageData	27
6.31	reloadDrawVideoData.....	28
6.32	clearDrawData.....	28
6.33	clearString	28
6.34	clearLine.....	28
6.35	clearRectangle	28
6.36	clearFillRectangle.....	29
6.37	clearEllipse	29
6.38	clearFillEllipse	29
6.39	clearPolygon.....	29
6.40	clearFillPolygon	30
6.41	clearImageBase64	30
7.	Thông tin công ty TNHH Công nghệ A.I.....	30

1. Giới thiệu

Module AI.CAMERA là một SDK được phát triển dưới dạng API COM (Component Object Model) để cung cấp các giao diện và phương thức cho người dùng sử dụng, dùng để kết nối và lấy dữ liệu từ các camera stream RTSP.

Module hỗ trợ các ngôn ngữ lập trình như C++, C#, Visual Basic, Python, PowerBuilder, Java... người dùng có thể sử dụng cho các ứng dụng Windows để thực hiện các thao tác:

- Kết nối camera IP hiển thị tín hiệu video lên control.
- Chụp hình ảnh từ camera IP, tải hình ảnh từ đĩa, lưu hình ảnh.

Ưu điểm của module:

- Kết nối camera tốc độ cao, không có độ trễ.
- Tự động kết nối lại khi mất kết nối.
- Sử dụng tài nguyên máy tính thấp.

Module 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.

2. Lưu ý quan trọng

2.1 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

2.2 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 Mã hóa Base64.

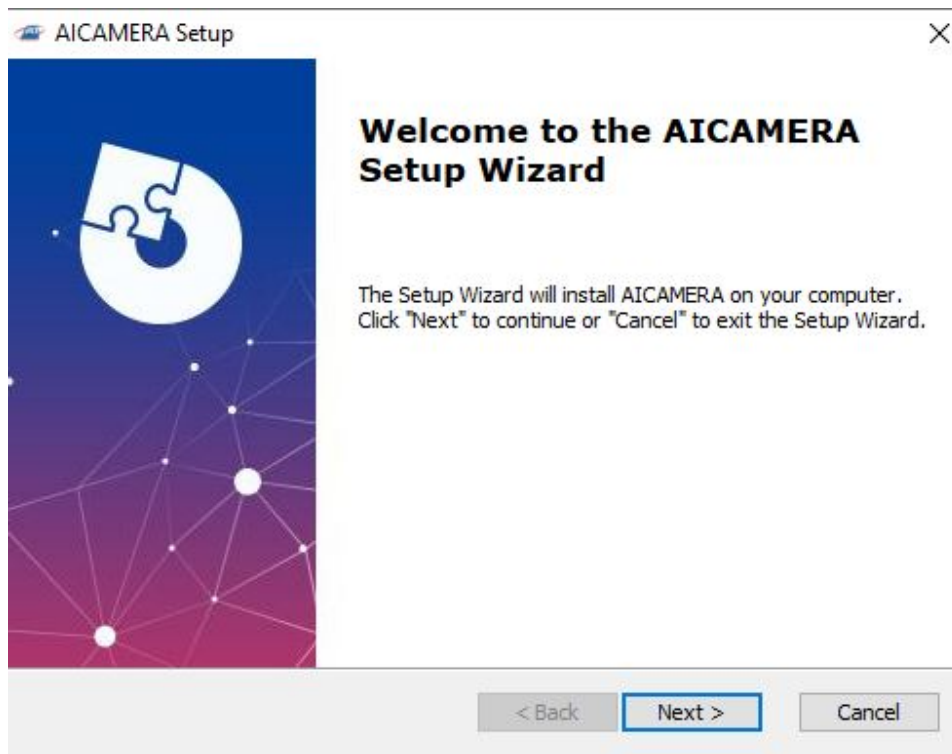
Base64 là một phương thức mã hóa 2 chiều từ tập tin nhị phân (binary) sang chuỗi ký tự (string) và ngược lại, nó thường được sử dụng để mã hóa các tập tin đa phương tiện (hình ảnh, âm thanh, video,...). Ký tự 64 trong Base64 là đại diện cho 64 ký tự trong bảng mã ASCII. Base64 thường được sử dụng trong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

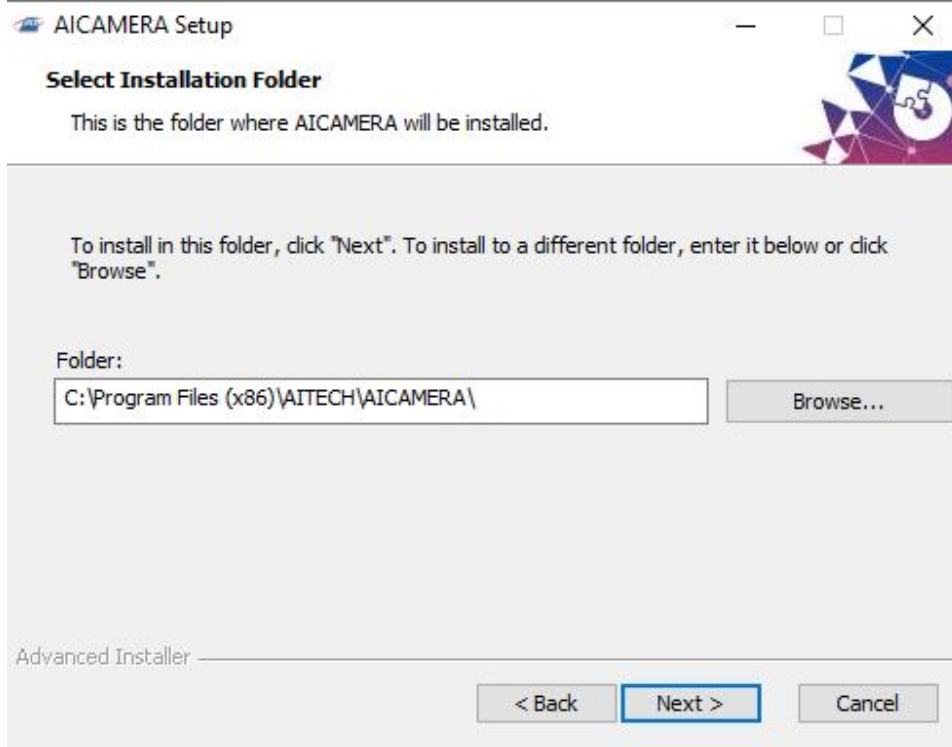
Chọn tệp AICAMERA nhấn ENTER hoặc nhấn đúp vào file để chạy cài đặt.

Màn hình xuất hiện :



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

Nhấn Next sẽ hiển thị màn hình mới như bên dưới:

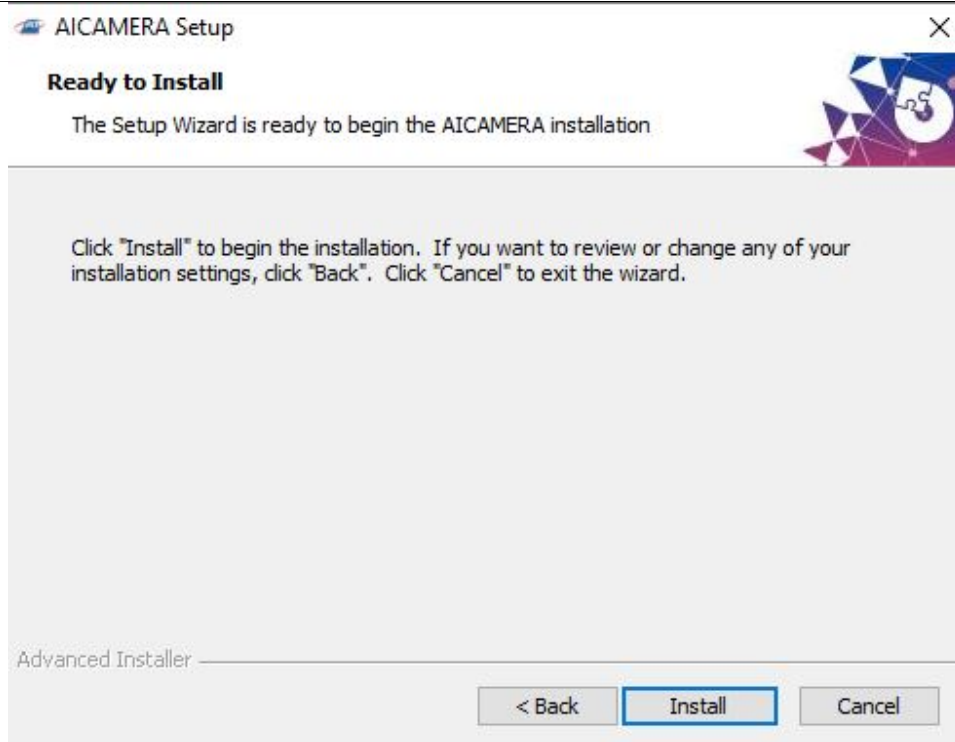


Folder:

Chương trình mặc định cài trên thư mục “C:\Program Files (x86)\AITECH\AICAMERA” 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 hiển thị màn hình chấp nhận cài đặt:



Nhấn “Install” để cài đặt.

Khi chạy xong, sẽ hiện thông báo tiến trình cài đặt thành công.

Nhấn “Finish” kết thúc tiến trình cài đặt.

4. License

4.1 Activate.

- **Bước 1:**

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

C:\Program Files (x86)\AITECH\AICAMERA\Bin\aiCameraActivate.exe

C > Local Disk (C:) > Program Files (x86) > AITECH > AICAMERA > Bin

Name	Date modified	Type	Size
decode	4/15/2023 3:16 PM	File folder	
aiAccr.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiAnpr.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiCamera.dll	4/15/2023 9:27 AM	Application exten...	1,069 KB
aiCameraActivate.exe	4/20/2023 9:05 AM	Application	113 KB
aiCodec.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiCodecdec.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiConnect.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiControl.dll	4/9/2023 5:53 PM	Application exten...	1,130 KB
aiDecode.dll	4/9/2023 5:53 PM	Application exten...	1,131 KB
aiDevice.dll	4/9/2023 5:53 PM	Application exten...	1,100 KB
aiForm.dll	4/9/2023 5:53 PM	Application exten...	1,116 KB
aiH264.dll	4/9/2023 5:53 PM	Application exten...	1,084 KB
aiH265.dll	4/9/2023 5:53 PM	Application exten...	1,100 KB
aiLicense.dll	4/9/2023 5:54 PM	Application exten...	1,100 KB
aiMjpeg.dll	4/9/2023 5:54 PM	Application exten...	1,100 KB
aiOnvif.dll	4/9/2023 5:54 PM	Application exten...	1,100 KB
aiPlayer.dll	4/15/2023 9:27 AM	Application exten...	64 KB
aiRtsp.dll	4/9/2023 5:54 PM	Application exten...	1,099 KB
aiStream.dll	4/15/2023 9:27 AM	Application exten...	64 KB
Newtonsoft.Json.dll	3/8/2023 2:09 PM	Application exten...	696 KB

- **Bước 2:**

AITECH - KÍCH HOẠT SẢN PHẨM ×

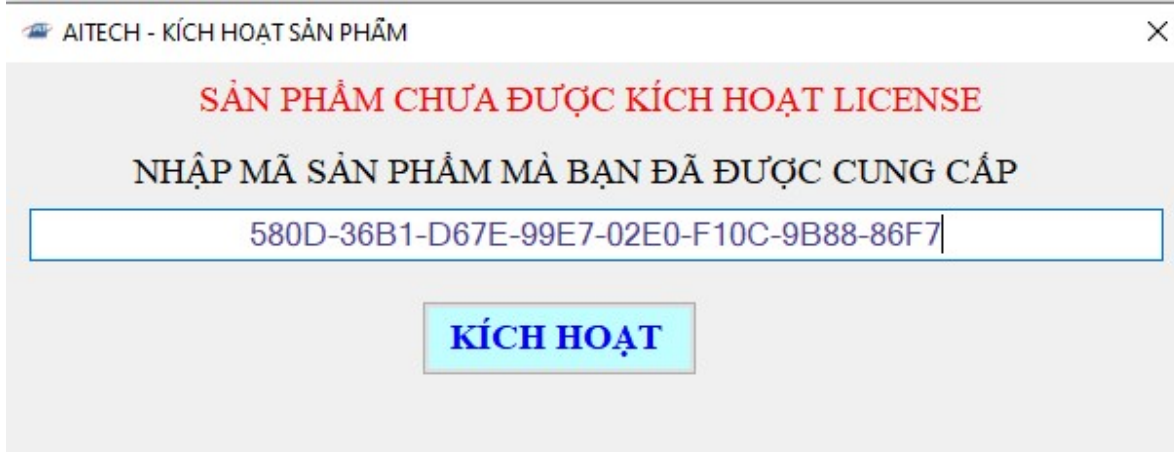
SẢN PHẨM CHƯA ĐƯỢC KÍCH HOẠT LICENSE

NHẬP MÃ SẢN PHẨM MÀ BẠN ĐÃ ĐƯỢC CUNG CẤP

- - - - -

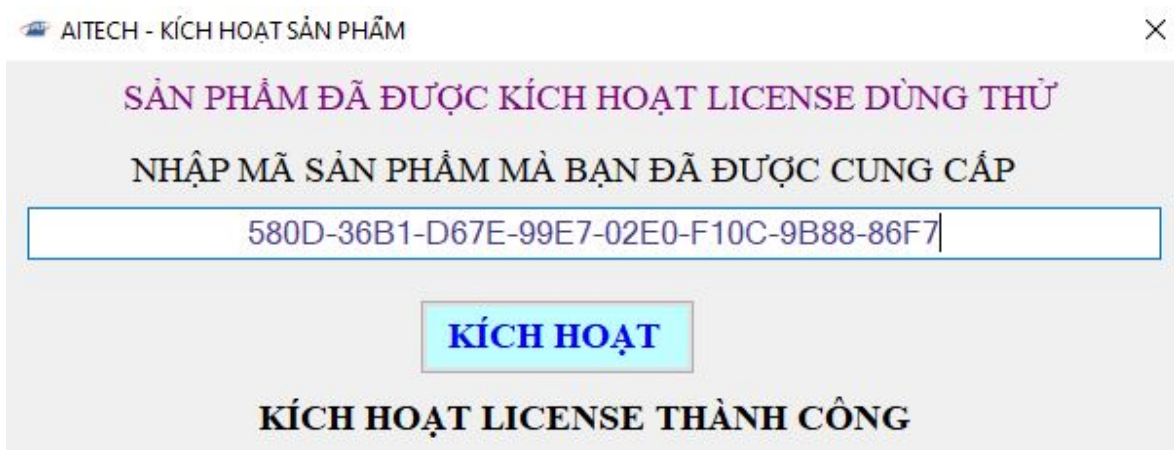
KÍCH HOẠT

Nhập mã sản phẩm mà bạn được cung cấp.



- **Bước 3:**

“KÍCH HOẠT”



4.2 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.

5. Tích hợp module AI.CAMERA vào phần mềm

Module phần mềm 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\AICAMERA\DemoSourceCode**

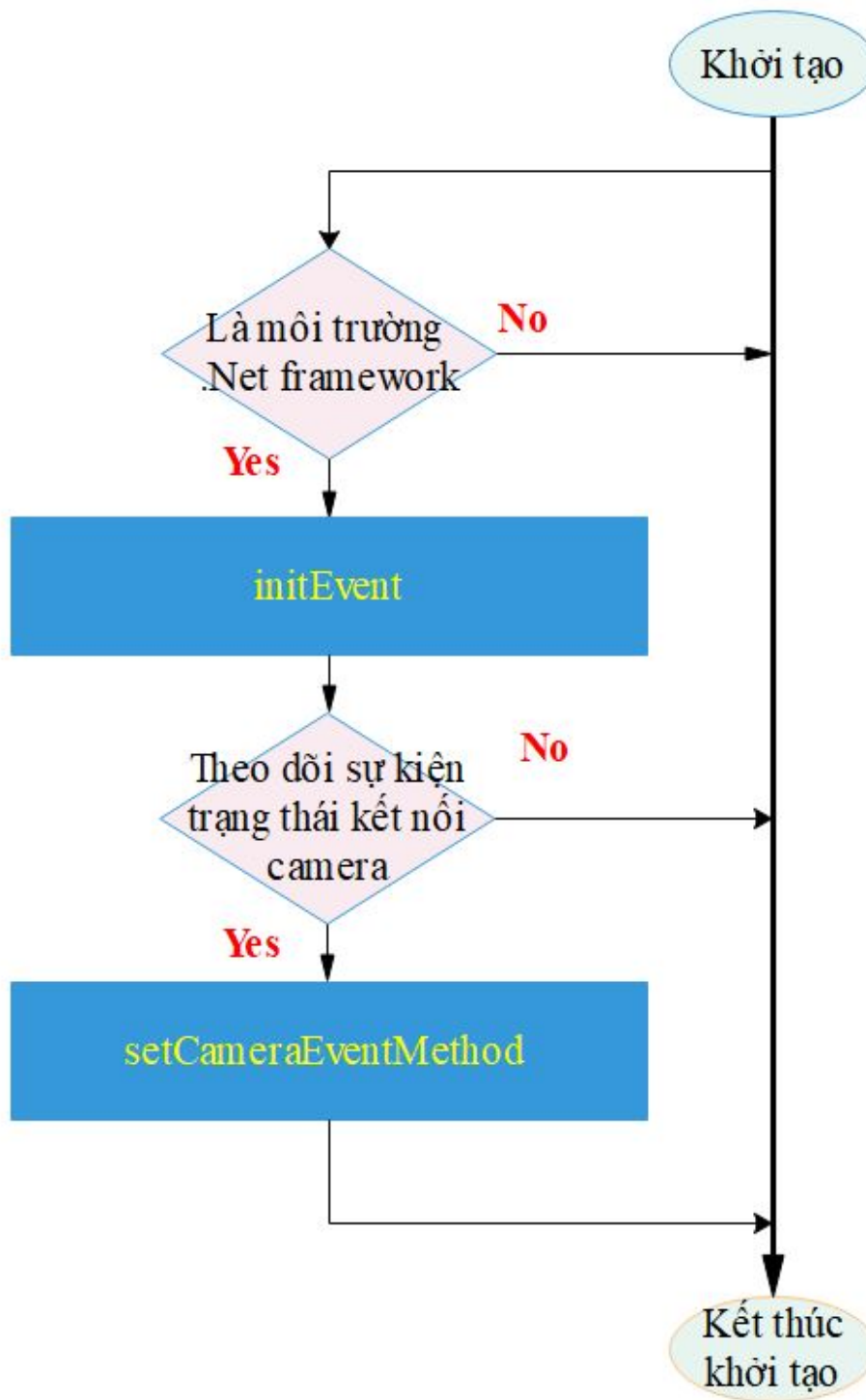
5.1 Khởi tạo

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.

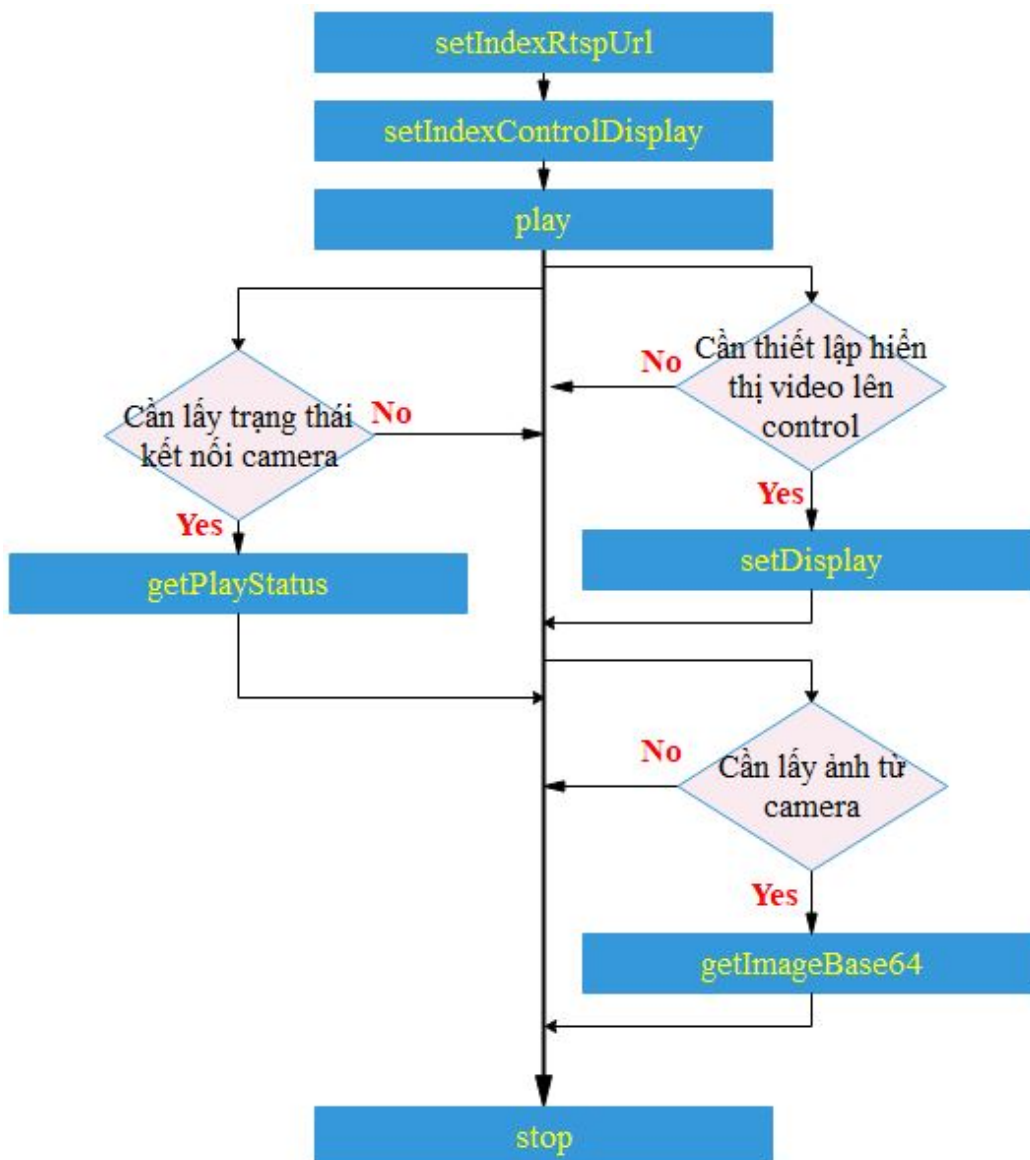
Ngôn ngữ lập trình	Khởi tạo API COM
C#	<pre>Type cameraCom; dynamic aicamera; cameraCom = Type.GetTypeFromProgID("Aitech.ClsAiRtspPlayerX64"); aicamera = Activator.CreateInstance(cameraCom, false);</pre>
VB.Net	<pre>Dim cameraCom As Type Dim aicamera As Object cameraCom = Type.GetTypeFromProgID("Aitech.ClsAiRtspPlayerX64") aicamera = Activator.CreateInstance(cameraCom, False)</pre>
C++	<pre>#import "C:\Program Files (x86)\AITECH\AICAMERA\aiCameraX64.tlb"; aiCameraX86::IClsAiRtspPlayerX86Ptr aicamera = nullptr; SUCCEEDED(CoInitialize(0)); SUCCEEDED(aicamera.CreateInstance("Aitech.ClsAiRtspPlayerX64"));</pre>
PowerBuilder10	<pre>OLEObject aiCamera aiCamera = CREATE oleobject aiCamera.ConnectToNewObject("Aitech.ClsAiRtspPlayerX86")</pre>

VB6	<code>Dim aicamera As Object</code> <code>Set aicamera = CreateObject("Aitech.ClsAiRtspPlayerX86")</code>
Python	<code>import win32com.client</code> <code>aicamera = win32com.client.Dispatch("Aitech.ClsAiRtspPlayerX64")</code>

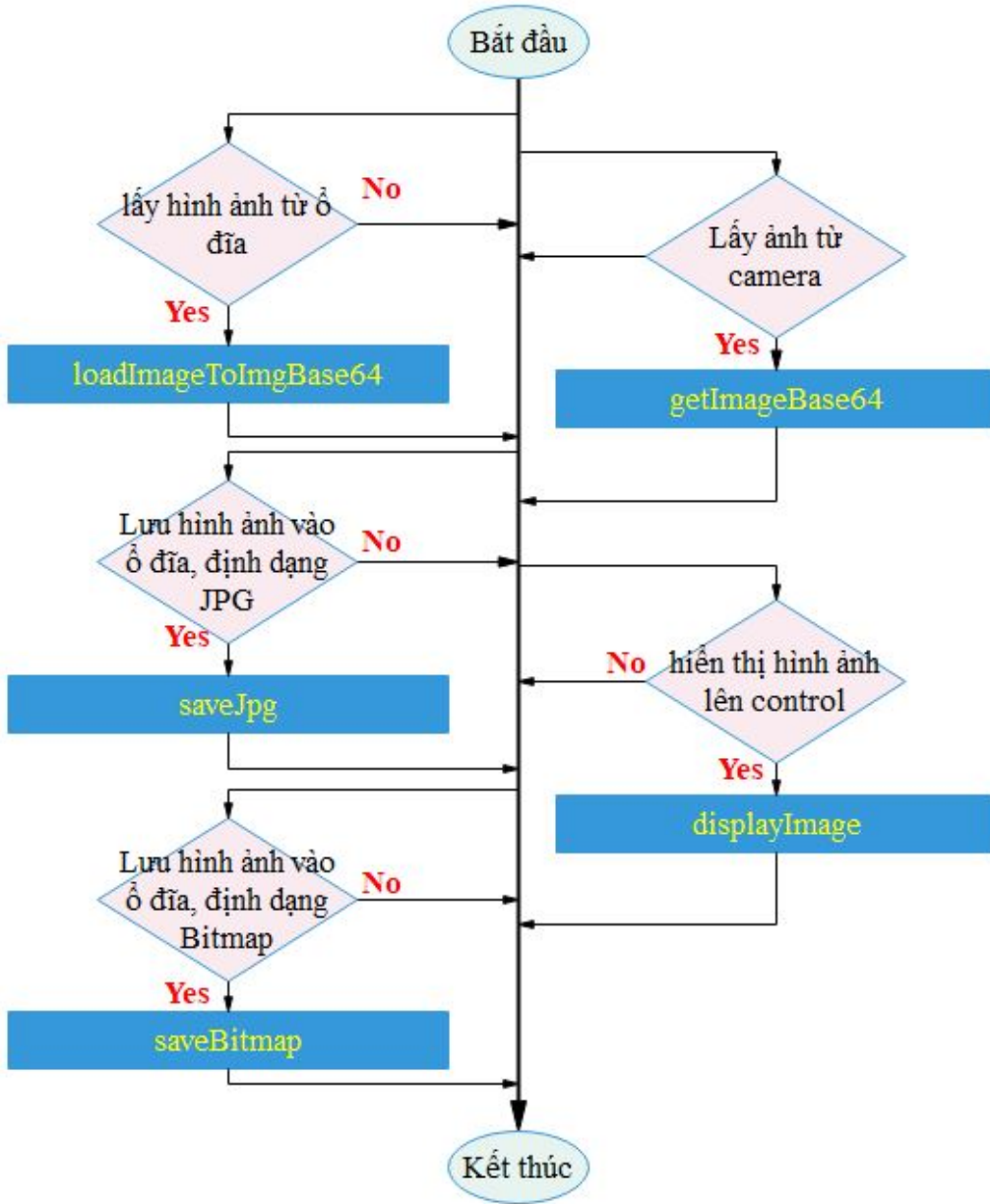
5.2 “Phương thức” khởi tạo



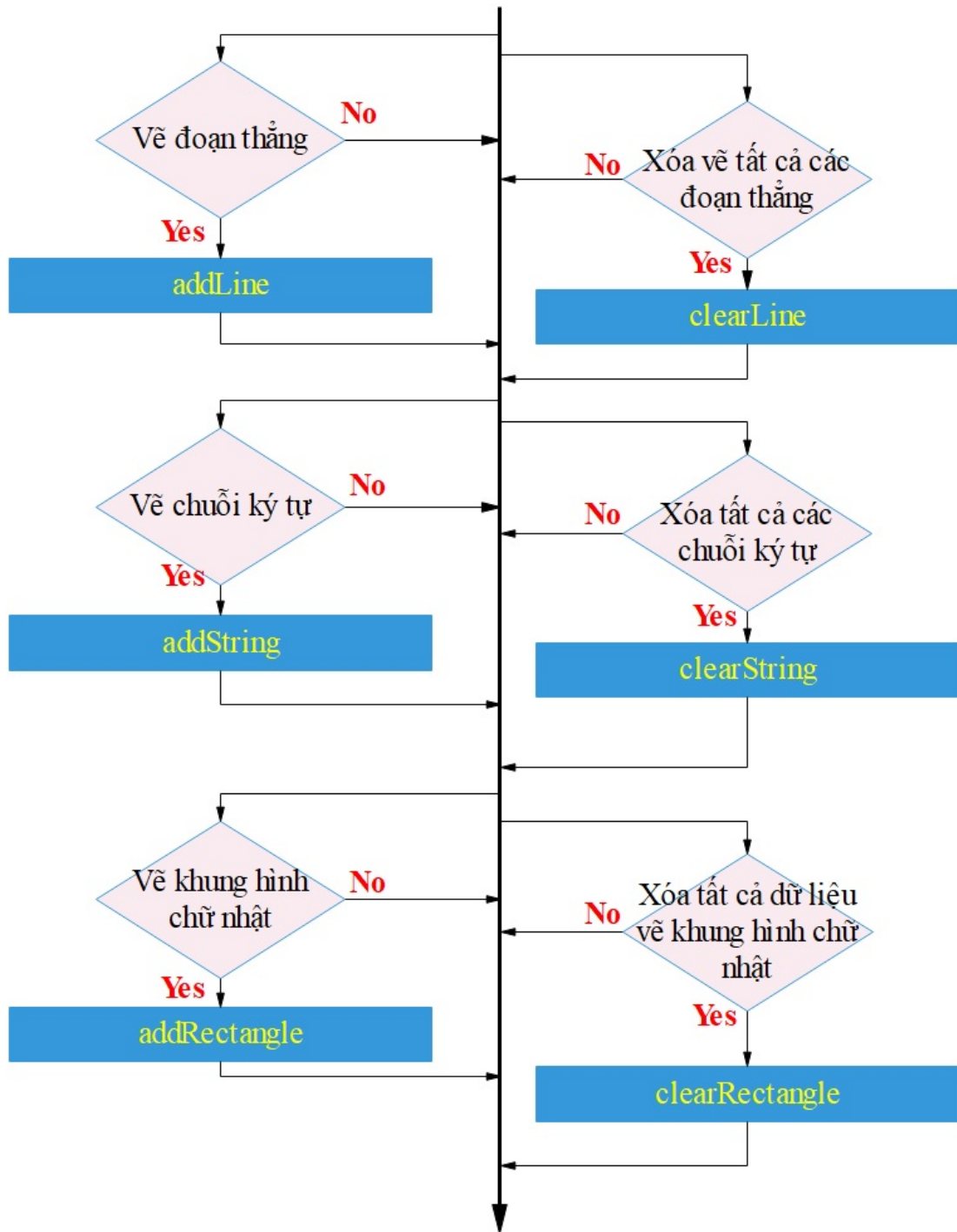
5.3 Các “Phương thức” tương tác với camera

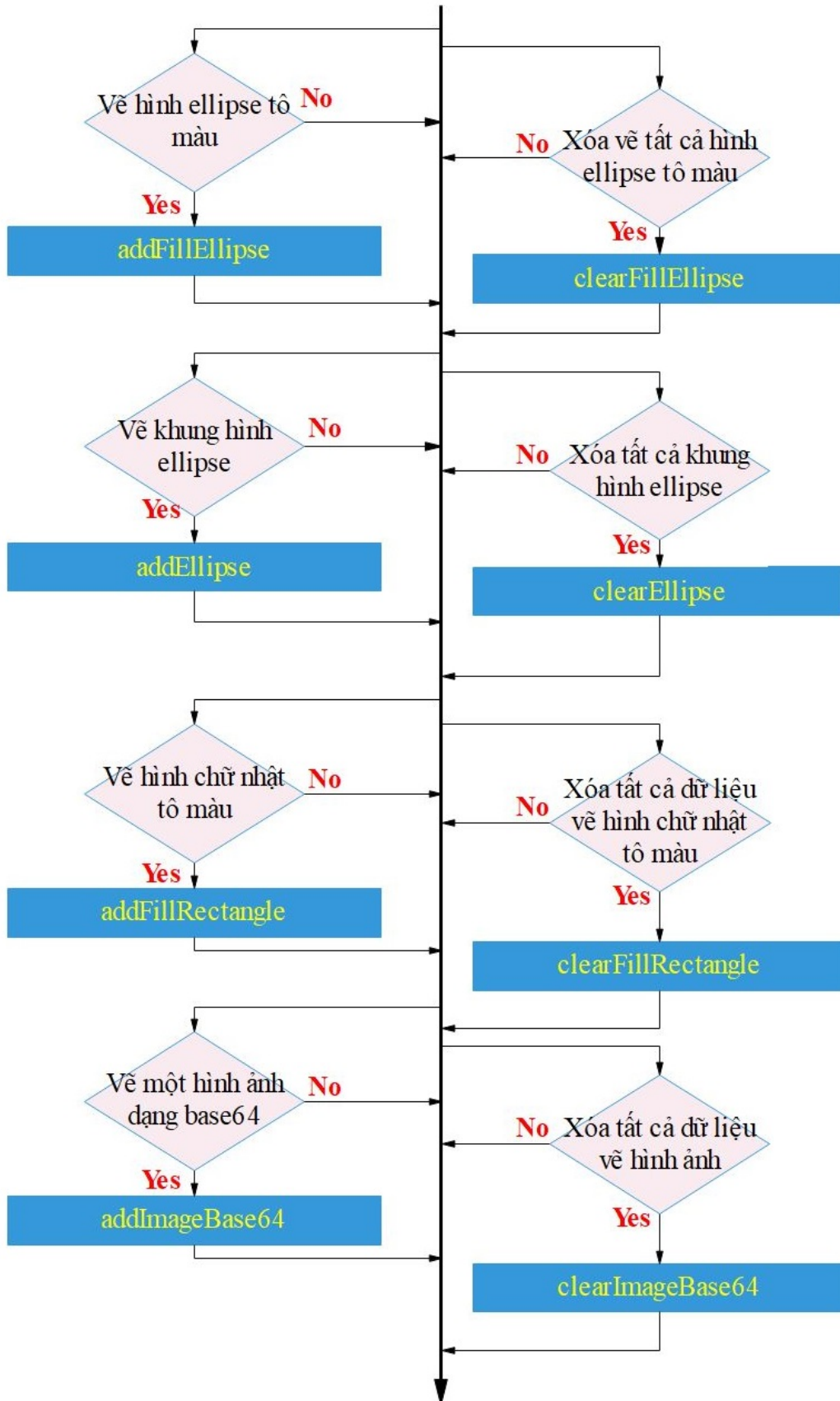


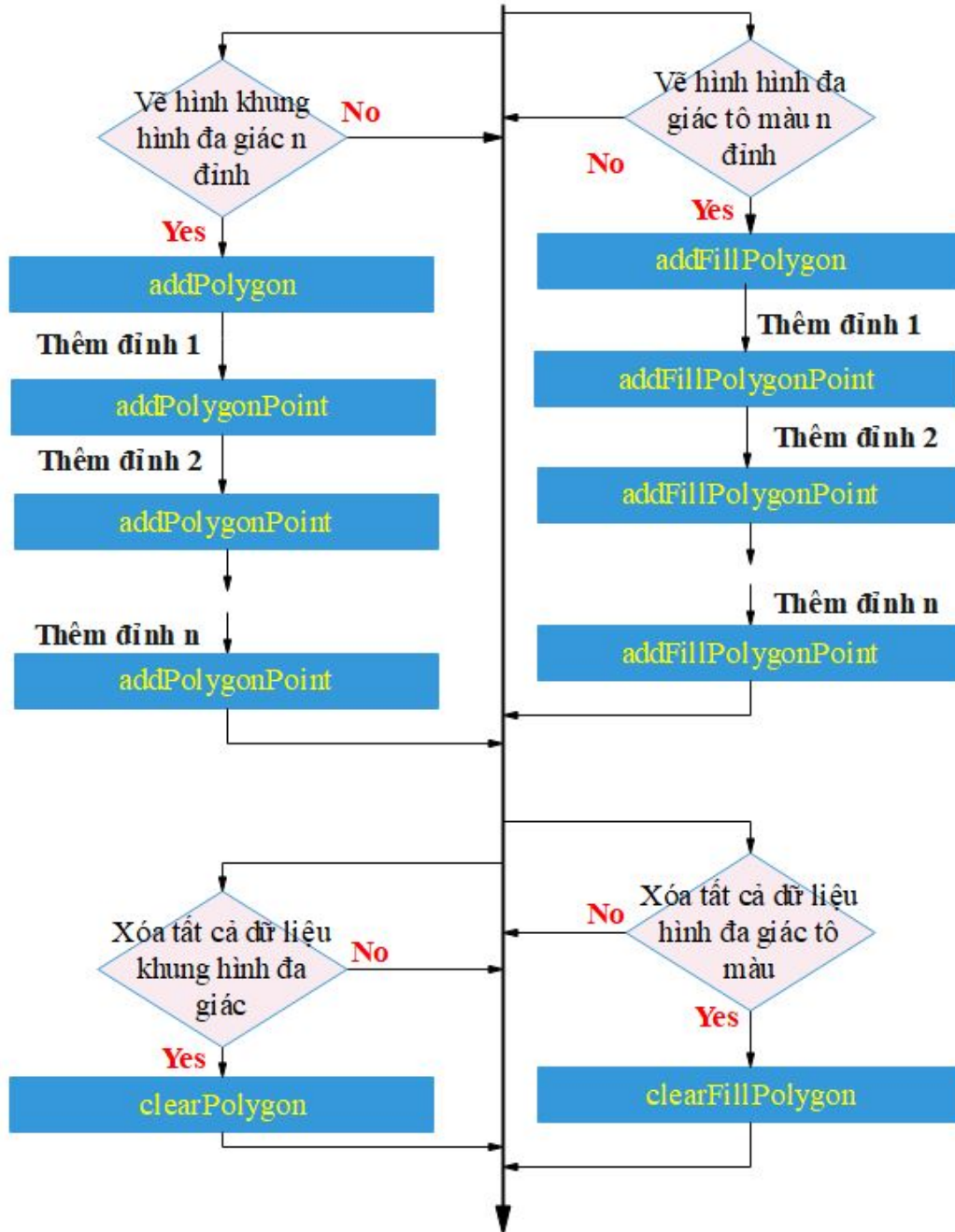
5.4 Các “Phương thức tương tác với hình ảnh”

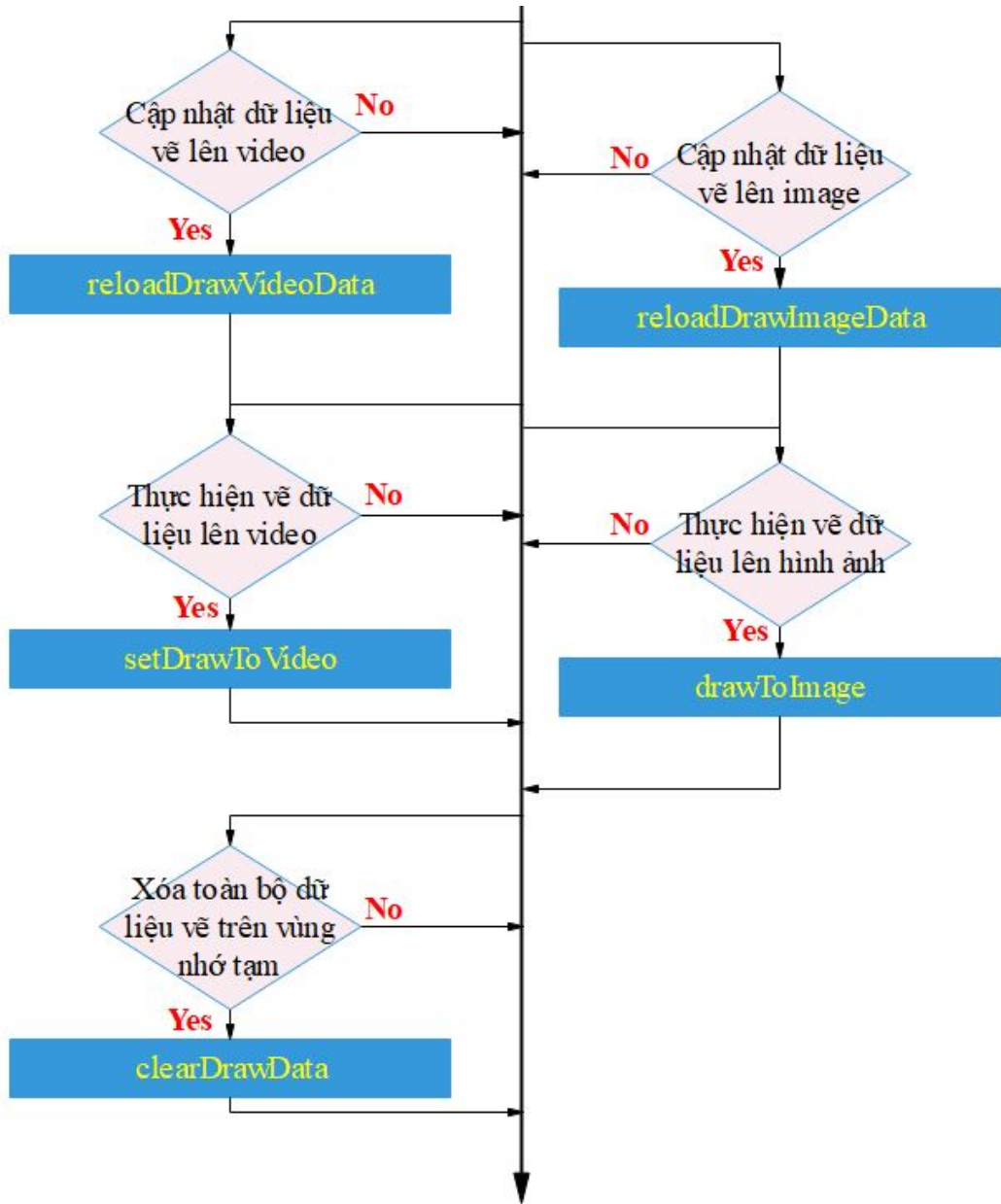


5.5 Các bước thao tác vẽ.









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 kết nối camera (online, offline).

6.2 `setCameraEventMethod`

- Cấu trúc:

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

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

6.3 `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`

6.4 `setIndexControlDisplay`

Tín hiệu camera tại vị trí `CameraIndex` sẽ hiển thị lên `displayControlHandle`.

- Cấu trúc:

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

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

6.5 play

- Cấu trúc:
`void play(int cameraIndex)`
- Tham số:
 - cameraIndex: Số thứ tự camera
- Trả về: Không
- Kết quả: Bắt đầu giải mã tín hiệu video tại vị trí **CameraIndex**

6.6 setDisplay

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

6.7 stop

- Cấu trúc:
`void stop(int cameraIndex)`
 - Tham số: cameraIndex: Số thứ tự camera
- Trả về: không
- Kết quả: Dừng giải mã tín hiệu video camera có vị trí **CameraIndex**

6.8 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

6.9 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

6.10 displayImage

- Cấu trúc:
`void displayImage(int handleControl, string imageBase64)`
- Tham số:
 - o handleControl: Handle của control hiển thị hình ảnh
 - o imageBase64: hình ảnh dạng base64 muốn hiển thị lên control
- Trả về: Không
- Kết quả: Hiển thị hình ảnh **imageBase64** lên control có handle là **handleControl**

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:
 - imageBase64: Hình ảnh dạng base64
 - 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 loadImageToImgBase64

- Cấu trúc:
`string loadImageToImgBase64(string fullFileName)`
- Tham số:
 - fullFileName: đường dẫn hình ảnh 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

6.14 dispose

- Cấu trúc:
`void dispose()`
- Tham số: Không
- Trả về: Không
- Kết quả: Dừng tất cả camera đang hoạt động

6.15 eventName

- Cấu trúc:
`void eventName(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. **eventName (callbackName)**: Có thể đổi tên phương thức.

6.16 addPolygon

- Cấu trúc:
`void addPolygon(int cameraIndex, string penColor, int penSize, int polygonIndex, int numberOfPoint)`
- Tham số:
 - o cameraIndex: Thứ tự camera
 - o penColor: màu của bút vẽ
 - o penSize: kích thước bút vẽ
 - o polygonIndex: số thứ tự của hình đa giác (có thể khai báo vẽ nhiều hình đa giác)
 - o numberOfPoint: số góc của hình đa giác
- Trả về: không
- Kết quả: Khai báo vẽ một khung hình đa giác lên video hoặc hình với số đỉnh là numberOfPoint, vị trí các đỉnh sẽ được khai báo trong “phương thức” **addPolygonPoint**.

6.17 addPolygonPoint

- Cấu trúc:
`void addPolygonPoint(int cameraIndex, int polygonIndex, int x, int y)`
- Tham số:
 - o cameraIndex: Thứ tự camera
 - o polygonIndex: Số thứ tự của đa giác
 - o x: Tọa độ trục X của một đỉnh đa giác
 - o y: Tọa độ trục Y của một đỉnh đa giác
- Trả về: Không
- Kết quả: Thêm một đỉnh của khung đa giác có số thứ tự là polygonIndex đã được khai báo trong hàm **addPolygon**. Nếu khung đa giác khai báo n đỉnh thì thêm n đỉnh bằng “phương thức ” addPolygonPoint.

6.18 addFillPolygon

- Cấu trúc:

```
void addFillPolygon(int cameraIndex, string penColor, int solidBrush, int polygonIndex, int numberOfPoint)
```

- Tham số:
 - cameraIndex: Thứ tự camera
 - penColor: Màu của bút vẽ
 - solidBrush: Độ trong suốt, có giá trị từ nhỏ đến lớn: 0 đến 255
 - polygonIndex: Số thứ tự của đa giác.
 - numberOfPoint: Số đỉnh của đa giác
- Trả về: Không
- Kết quả: Khai báo vẽ một đa giác và tô màu đa giác đó lên video hoặc hình với màu penColor, số đỉnh là numberOfPoint, vị trí các đỉnh sẽ được khai báo trong “phương thức” **addFillPolygonPoint**

6.19 addFillPolygonPoint

- Cấu trúc:

```
void addFillPolygonPoint(int cameraIndex, int polygonIndex, int x, int y)
```
- Tham số:
 - cameraIndex: Thứ tự camera
 - polygonIndex: Số thứ tự của đa giác
 - x: Tọa độ trục X của một đỉnh đa giác
 - y: Tọa độ trục Y của một đỉnh đa giác
- Trả về: Không
- Kết quả: Thêm một đỉnh của đa giác tô màu có số thứ tự là polygonIndex đã được khai báo trong hàm **addFillPolygon**. Nếu đa giác tô màu khai báo n đỉnh thì thêm n đỉnh bằng “phương thức ” **addFillPolygonPoint**.

6.20 addString

- Cấu trúc:

```
void addString(int cameraIndex, string data, string penColor, string fontName, int fontSize, bool fontBold, int x, int y)
```
- Tham số:
 - cameraIndex: Thứ tự camera
 - data: chuỗi ký tự muốn thêm vào.
 - fontName: Tên font (red, blue, green...)

- `fontSize`: Kích thước font.
- `fontBold`: Tô đậm dữ liệu hay không? (true, false)
- `x`: Tọa độ trục X của chuỗi ký tự muốn vẽ lên video hoặc hình.
- `y`: Tọa độ trục Y của chuỗi ký tự muốn vẽ.
- Trả về: Không
- Kết quả: Khai báo một chuỗi ký tự muốn vẽ lên video hoặc hình ảnh.

6.21 `addLine`

- Cấu trúc:
`void addLine(int cameraIndex, string penColor, int penSize, int x1, int y1, int x2, int y2)`
- Tham số:
 - `cameraIndex`: Thứ tự camera
 - `penColor`: Màu của bút vẽ.
 - `penSize`: Kích thước bút vẽ.
 - `x1`: Tọa độ trục X của điểm đầu tiên đoạn thẳng.
 - `y1`: Tọa độ trục Y của điểm đầu tiên đoạn thẳng.
 - `x2`: Tọa độ trục X của điểm thứ hai đoạn thẳng.
 - `y2`: Tọa độ trục Y của điểm thứ hai đoạn thẳng
- Trả về: Không
- Kết quả: Khai báo vẽ một đường thẳng lên video hoặc hình ảnh.

6.22 `addFillRectangle`

- Cấu trúc:
`void addFillRectangle(int cameraIndex, string penColor, int solidBrush, int x, int y, int width, int height)`
- Tham số:
 - `cameraIndex`: Thứ tự camera
 - `penColor`: Màu của bút vẽ
 - `solidBrush`: Độ trong suốt của bút vẽ.
 - `x`: Tọa độ trục X của đỉnh đầu tiên.
 - `y`: Tọa độ trục Y của đỉnh đầu tiên
 - `width`: Chiều dài của hình chữ nhật

- height: Chiều cao của hình chữ nhật
- Trả về: Không
- Kết quả: Khai báo vẽ một hình chữ nhật có tô màu lên video hoặc hình ảnh.

6.23 addRectangle

- Cấu trúc:
`void addRectangle(int cameraIndex, string penColor, int penSize, int x, int y, int width, int height)`
- Tham số:
 - cameraIndex: Thứ tự camera
 - penColor: Màu của bút vẽ
 - penSize: Kích thước bút vẽ
 - x: Tọa độ trục X của đỉnh đầu tiên.
 - y: Tọa độ trục Y của đỉnh đầu tiên.
 - width: Chiều dài của hình chữ nhật
 - height: Chiều cao của hình chữ nhật
- Trả về: Không
- Kết quả: Vẽ một khung hình chữ nhật lên video hoặc hình ảnh.

6.24 addEllipse

- Cấu trúc:
`void addEllipse(int cameraIndex, string penColor, int penSize, int x, int y, int width, int height)`
- Tham số:
 - cameraIndex: Thứ tự camera
 - pen Color: Màu bút vẽ
 - penSize: kích thước bút vẽ
 - x: Tọa độ trục X của khung hình chữ nhật
 - y: Tọa độ trục Y của khung hình chữ nhật
 - width: Chiều dài của khung hình chữ nhật
 - height: Chiều cao của khung hình chữ nhật
- Trả về: Không

- Kết quả: Khai báo **vẽ một khung hình ellipse** bên trong một khung hình chữ nhật (không vẽ khung hình chữ nhật, chỉ vẽ ellipse, đối với hình tròn thì vẽ khung hình chữ nhật có 2 cạnh bằng nhau).

6.25 addFillEllipse

- Cấu trúc:
`void addFillEllipse(int cameraIndex, string penColor, int solidBrush, int x, int y, int width, int height)`
- Tham số:
 - o cameraIndex: Thứ tự camera
 - o penColor: Màu bút vẽ
 - o solidBrush: Độ trong suốt của bút vẽ (từ 0 -> 255)
 - o x: Tọa độ trục X của điểm đầu khung hình chữ nhật
 - o y: Tọa độ trục Y của điểm đầu khung hình chữ nhật
 - o width: Chiều dài khung hình chữ nhật
 - o height: Chiều cao khung hình chữ nhật
- Trả về: Không
- Kết quả: Khai báo **Vẽ một hình ellipse có tô màu** bên trong một khung hình chữ nhật (không vẽ khung hình chữ nhật, chỉ vẽ ellipse, đối với hình tròn thì vẽ khung hình chữ nhật có 2 cạnh bằng nhau).

6.26 addImageBase64

- Cấu trúc:
`void addImageBase64(int cameraIndex, string imageBase64, int x, int y, int width, int height)`
- Tham số:
 - o cameraIndex: Thứ tự camera
 - o imageBase64: Hình ảnh dạng base64
 - o x: Tọa độ trục X của khung hình chữ nhật
 - o y: Tọa độ trục Y của khung hình chữ nhật
 - o width: Chiều dài của khung hình chữ nhật
 - o height: Chiều cao của khung hình chữ nhật
- Trả về: Không

- Kết quả: Vẽ một hình ảnh trong một khung hình chữ nhật (chỉ vẽ hình ảnh, không vẽ khung hình) lên video hoặc hình ảnh.

6.27 **setDrawToVideo**

- Cấu trúc:
`void setDrawToVideo(int cameraIndex, Boolean drawStatus)`
- Tham số:
 - o cameraIndex: Thứ tự camera
 - o drawStatus: (true, false) cài đặt trạng thái vẽ dữ liệu lên video
- Trả về: Không
- Kết quả: Vẽ hoặc dừng vẽ dữ liệu đã khai báo lên video

6.28 **drawToImage**

- Cấu trúc:
`string drawToImage(int cameraIndex, string imageBase64)`
- Tham số:
 - o cameraIndex: Thứ tự camera
 - o imageBase64: Hình ảnh dạng base64 muốn vẽ lên.
- Trả về: kết quả cấu trúc `string` là hình ảnh base64 sau khi vẽ dữ liệu lên
- Kết quả:

6.29 **getFrameCounter**

- Cấu trúc:
`int getFrameCounter(int cameraIndex)`
- Tham số:
 - o cameraIndex: Thứ tự camera
- Trả về: Kiểu `int` : là số khung hình đã giải mã từ stream video
- Kết quả: Lấy số khung đã giải mã từ stream video

6.30 **reloadDrawImageData**

- Cấu trúc:
`void reloadDrawImageData(int cameraIndex)`
- Tham số:
 - o cameraIndex: Thứ tự camera
- Trả về: Không

-
- Kết quả: Tải lại dữ liệu vẽ cho hình ảnh.

6.31 reloadDrawVideoData

- Cấu trúc:
`void reloadDrawVideoData(int cameraIndex)`
- Tham số:
 - o cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Tải lại dữ liệu vẽ cho video.

6.32 clearDrawData

- Cấu trúc:
`void clearDrawData(int cameraIndex)`
- Tham số:
 - o cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa toàn bộ dữ liệu vẽ đã khai báo.

6.33 clearString

- Cấu trúc:
`void clearString(int cameraIndex)`
- Tham số:
 - o cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ dạng `string` đã khai báo

6.34 clearLine

- Cấu trúc:
`void clearLine(int cameraIndex)`
- Tham số:
 - o cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ đoạn thẳng đã khai báo

6.35 clearRectangle

- Cấu trúc:

void clearRectangle(int cameraIndex)

- Tham số:
 - cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ hình chữ nhật đã khai báo

6.36 clearFillRectangle

- Cấu trúc:

void clearFillRectangle(int cameraIndex)
- Tham số:
 - cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ hình chữ nhật có tô màu đã khai báo

6.37 clearEllipse

- Cấu trúc:

void clearEllipse(int cameraIndex)
- Tham số:
 - cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ hình ellipse đã khai báo

6.38 clearFillEllipse

- Cấu trúc:

void clearFillEllipse(int cameraIndex)
- Tham số:
 - cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ hình ellipse có tô màu đã khai báo

6.39 clearPolygon

- Cấu trúc:

void clearPolygon(int cameraIndex)
- Tham số:
 - cameraIndex: Thứ tự camera

- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ khung hình đa giác đã khai báo

6.40 clearFillPolygon

- Cấu trúc:
`void clearFillPolygon(int cameraIndex)`
- Tham số:
 - o cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ hình đa giác có tô màu đã khai báo

6.41 clearImageBase64

- Cấu trúc:
`void clearImageBase64(int cameraIndex)`
- Tham số:
 - o cameraIndex: Thứ tự camera
- Trả về: Không
- Kết quả: Xóa dữ liệu vẽ hình ảnh đã khai báo

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

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

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