

#### MACHINE VISION PLATFORM FOR FACTORY AUTOMATION

o B4 by Alo Matri







For more information please see: www.aiomatrix.vn

Make Machine Vision Easier

## **Table of Content**

CHAPT	ER 1: TỔNG QUAN VỀ CVAIO B+	.1
1.1	Giới thiệu	. 2
1.2	Giao diện người dùng - GUI	. 2
	Main GUI – Giao diện chính	. 3
	UI Design – Giao diện thiết kế hiển thị	. 4
	Model – Giao diện thiết kế Vision Process	. 5
	Calibration – Giao diện căn chỉnh tự động	. 6
	Setting – Giao diện cài đặt liên quan	. 7
	History – Giao diện lịch sử Log và hình ảnh	. 8
1.3	Platform Structure – Cấu trúc chương trình	.9
1.4	Setup Guide – Các bước để thiết lập chương trình	10
CHAPT	ER 2: GUI LIBRAY – THƯ VIỆN CÁC ĐỐI TƯỢNG HIỂN THỊ	13
2.1	Overview - Giới thiệu chung	14
2.2	Screen Display - Hiển thị đa màn hình	16
2.3	Scheduler Display – Hiển thị luồng xử lý ảnh	16
2.4	Interface Display – Hiển thị trạng thái giao tiếp với PLC/PC	17
2.5	Image Display – Hiển thị hình ảnh	17
2.6	Grid View Display – Hiển thị dữ liệu dạng lưới	18
2.7	List View Display – Hiển thị dữ liệu dạng danh sách	18
2.8	Graph Display – Hiển thị đồ thị	19
	Case Study 1: UI Design	20
	Case Study 2: Multi Screen	21
CHAPT	ER 3: TOOL LIBRARY – THƯ VIỆN CÁC CÔNG CỤ XỬ LÝ ẢNH	22
3.1	Overview - Giới thiệu chung	23
3.2	Common Tool – Acquisition, Image File, Region, Template Match, Light Control	25
3.3	Calibration & Fixturing – Find Chessboard, Find Calib Matrix, Fiture	28
3.4	Deprecated – QR Decode	30
3.5	Geometry Creation – Create Circle, Create Line, Create Rectangle	30
3.6	Geometry Finding & Fitting – Find Line, Find Circle, Find Corner, Find Points	32
3.7	Geometry Measurement – Distance Points, Distance Line, Distance Circle	35
3.8	Geometry Intersection – Circle x Circle, Line x Circle, Line x Line	37

3.9 Image Processing – Add Weighted, Binarization, Blob, Histogram, Morphology, etc	38
3.10 Scheduler – Công cụ điều phối các nhánh xử lý	42
3.11 Vision Algorithm – Thuật toán xử lý ảnh	43
Case Study 3: Circuit Inspection	44
CHAPTER 4: CALIBRATION – CĂN CHỈNH	45
4.1 Threory – Lý thuyết về căn chỉnh	46
4.2 Manual Calibration – Căn chỉnh bằng Chessboard	47
4.3 Auto Calibration – Căn chỉnh tự động	48
CHAPTER 5: HISTORY & USER LOGIN – LỊCH SỬ & QUẢN LÝ TÀI KHOẢN ĐĂNG NHẬP	51
5.1 History – Phân tích lịch sử	52
5.2 User Login – Quản lý tài khoản đăng nhập	52
CHAPTER 6: TOOL BY USER – PHÁT TRIỀN CÔNG CỤ VÀ LUỒNG XỬ LÝ MỚI	53
6.1 Overview - Giới thiệu chung	54
6.2 Tool By User – Phát triển công cụ xử lý ảnh	54
6.3 Scheduler By User – Phát triển luồng xử lý ảnh	55
6.4 Camera By User – Phát triển tích hợp camera	55
6.5 Interface By User – Phát triển module giao tiếp	56
CHAPTER 7: COGNEX VISION PRO & HALCON MVTEC	57
7.1 Cognex Vision Pro	58
7.2 Halcon MVTech	58
APPENDIX	59
Appendix 1	59
Appendix 2	59

# CHAPTER 1: TỔNG QUAN VỀ CVAIO B+

- **1.1 Overview** Giới thiệu chung
- 1.2 GUI Giao diện người dùng
- **1.3 Platform Structure** Cấu trúc chương trình
- **1.4 Setup Guide** Các bước để thiết lập chương trình

#### Các nội dung sẽ được đề cập trong chương này:

- \* Tổng quan về CVAIO B+ Platform
- $\div$ Các màn hình chính của giao diện hiển thị
- Cấu trúc của phần mềm, các luồng xử lý ảnh, giao tiếp với PLC, Camera, Light Controller \*
- Chi tiết các bước để thiết lập một chương trình xử lý ảnh với CVAIO B+ \*

#### 1.1 Giới thiệu



What is CVAIO B+? Với mục tiêu giảm thiểu thời gian phát triển một chương trình Machine Vision cũng như cắt giảm tối đa chi phí phát sinh trong quá trình đưa một thuật toán xử lý ảnh vào áp dụng trong nhà máy, AIO MATRIX cung cấp một nền tảng Machine Vision Platform CVAiO B+ dựa trên mã nguồn mở OpenCV. Thông qua thao tác bốc thả (Drag&Drop), người dùng có thể dễ dàng thêm hoặc bớt đi các luồng xử lý ảnh. Cùng với sự linh hoạt của công cụ điều phối (Scheduler), các thuật toán xử lý ảnh có thể được tiến hành lần lượt (Multi-Shot) trước khi kết quả cuối cùng được tổng hợp và phán định bởi công cụ Algorithm. Thêm vào đó là việc tích hợp tính năng căn chỉnh tự động (Auto Calibration) giúp cho việc điều hướng robot tìm gắp vật thể đơn giản và dễ dàng hơn bao giờ hết

#### 1.2 Giao diện người dùng - GUI

Giao diện của chương trình CVAIO B+ bao gồm các phần chính sau:

- > Main: Hiển thị trạng thái xử lý của toàn bộ chương trình khi chạy tự động
- > UI Design: Thiết kế màn hình hiển thị Main, điều chỉnh các thuộc tính của từng đối tượng
- Model: Thiết kế thuật toán xử lý ảnh, lựa chon IO cho các Scheduler
- Calibration: Cài đặt và tiến hành căn chỉnh tự động
- Setting: Quan sát trạng thái giao tiếp PLC và các cài đặt khác
- > History: Hiển thị lịch sử hoạt động của chương trình và kết quả hình ảnh đã được lưu lại

 $\overline{7}$ 

Main Display<sup>(5)</sup>

1

#### Main GUI – Giao diện chính

MACHINE VISION PLATFORM CVAIO B+	AIO MATRIX 6 #86 Date 2 Time	1022 - 12 - 17 19:43:54	License Activated Version 5.0.1	CPU 5% C Drive RAM 60015.8GB D Drive	937221GB 23/244GB
SchedulerDisplay #0 (SchObjectAlign1P)		WaitStar	tRequest 0 ms 🕧	User Login	
AlgoObjectAlign1P 0			0	2 (2)	(1)
[AlgoObjectAlign1P] OutputImage		•	ModbusTCP		
			Data IF	User ID Admin	MAIN
		0	ModbusTCP	Company A V	_~~
	(11032 25350 2552 250 00 00 00 00		O CTADT 1	Acesss Admin V	
			COMP 7		UI DESIGN
			READY 70		
			OK 76		
			00077		MODEL
20			NG-77	Model Control	000
× ×			chadular Time	ALIGNMENT 1	<u></u> র্০) ।
		S K	(ey Value	NE 3	000
		Co	omplete 343	COPY	CALIB
H 2048 x W 2448 [-110 104]=-	Auto-Et Culine Avie Pivel	× 17% × W	rte OK/NG 290	DELETE	10.5
Scheduler Lee		Ju	dgement 285	Basic IO Communication	
Time Scheduler	Message	Vis	sion Process 254	ModbusTCP	SETTING
19:43:53:631 SchedulerDisplay #0 (SchObjectAlign1P)	Do Judgement Done 19:43:39:452 332 OK	Ca	alc Off 242	System Reset	-
19:43:53:631 SchedulerDisplay #0 (SchObjectAlign1P) 19:43:53:632 SchedulerDisplay #0 (SchObjectAlign1P)	Calculate Judgement: 285 19:43:41:227 346 OK Do Write Result Done 19:43:43:015 362 OK	Tri	igger 1 Off 164	ID State	
19:43:53:634 SchedulerDisplay #0 (SchObjectAlign1P) 19:43:53:636 SchedulerDisplay #0 (SchObjectAlign1P)	Write Data : 288 19:43:44:803 357 OK Write Judgement : 290 19:43:46:565 340 OK	Tri	igger 1 On 27	Reset - 10'	HISTORY
19:43:53:677 SchedulerDisplay #0 (SchObject Align 1P)	Wait End Ack : 331 19:43:48:362 361 OK		1	Alive - '64'	
19:43:53:589 SchedulerDisplay #0 (SchObjectAlign IP) 19:43:54:070 SchedulerDisplay #0 (SchObjectAlign IP)	Finalization Done v 19:43:50:142 346 OK			🧿 Ready - '65'	
	> 19:43:53:689 343 OK	× ( <	>	🔘 ST	
	Object Aligment - Robot Pickup with Single Tool			STO I	
			/	Copyright by AIO	EXIT MATRIX 2022
Manu Bar	Lựa chọn chuyển đổi gii	ửa các n	nàn hình hiểr	n thị	
User Login <sup>(1)</sup>	Quản lý người dùng đăr	ıg nhập			
Recipe Control <sup>(2)</sup>	Quản lý việc tạo mới, th	êm, sửa	, xóa các rec	cipe	
Basic IO Status <sup>(3)</sup>	Hiển thị trạng thái của c	ác tín hiể	ệu giao tiếp c	cơ bản	
Program Control	Quản lý việc chạy và dù	ng các S	Scheduler xů	r lý ảnh	
Title Bar <sup>(4)</sup>	Hiển thị tiêu đề, ID thiết	bị, phiêr	n bản chươn	g trình	

(1) Việc quản lý tài khoản đăng nhập giúp kiểm soát việc thay đổi các thông số cài đặt của chương trình. Trong một số trường hợp, chỉ có các tài khoản admin mới được phép truy cập và thay đổi các parameter. Xem chi tiết tại <u>Section 5.2</u>.

Hiển thị trạng thái, kết quả xử lý, trạng thái giao tiếp với PLC/PC

- (2) Tất cả các thông số cài đặt được quản lý theo model. Việc quản lý này giúp người dùng dễ dàng phân biệt được model nào đang được sử dụng cũng như thuận tiện trong quá trình thay đổi model (Job change). Dữ liệu thông số của tất cả các model được lưu tại địa chỉ: C:\CVAiO.Bplus\Recipe
- (3) Các tín hiệu IO báo cáo trạng thái của chương trình cho PLC. Bit Ready là tín hiệu duy trì ngay sau khi nút START được bấm. Bit Alive là tín hiệu nháy On, Off theo chu kỳ 3 giây. Sau 3 giây mà không thấy Bit Alive thay đổi trạng thái thì xem như chương trình đang bị lỗi và cần tiến hành khởi động lại chương trình. Xem chi tiết tại <u>Section 1.3</u>
- (4) Tên và ID của chương trình có thể thay đổi bằng cách nhấp đúp chuột và thay đổi 1 tên mới hoặc sửa trực tiếp trong system file tại địa chỉ: C:\CVAiO.Bplus\System. Kích đúp chuột vào trạng thái "Actived" để kiểm tra thông tin license của dongle key.
- (5) Tham khảo thiết kế mẫu của chương trình với 3 Schedulers ở Case Study 1

Note: Trong phiên bản dùng thử không có dongle key, chương trình sẽ tự động đóng tất cả các luồng xử lý ảnh và thoát ra sau 30 phút kể từ khi khởi động. Thời gian còn lại của mỗi lần dùng thử được đếm ngược và hiển thị dưới đáy màn hình.

#### UI Design – Giao diện thiết kế hiển thị

MACHINE VISION	CVAIO B+	AIC		No #86	Date 202	22 - 12 - 17	Lice	ense Activated	CPU	3% CI	Drive 93/22
XIIIX		MAII	uA		Time	19:46:35	Ver	sion 5.0.1	FOAM	AND A STOCKED D	Unive 15/24
GUITIBRARY	Schedu	erDisplay #0 (	SchObject	Align 1P)		Wai	it Start Red	quest 0 ms	User Log	in	
E- 🗄 Display Control	AlgoOt	piectAlian 1P.0					0			HELLO	
SchedulerDisplay	/ good	Jeen igni o						(2) <sup>4</sup>		neeco	
InterfaceDisplay	[AlgoObj	ectAlign 1P] Outp	utimage				•				
- Image Display								ModbusTCP		-	MAIN
GraphDisplay								Data IF	UserID	Admin	
GridViewDisplay								ModbusTCP	Company	A ~	
	6.2										; (1)
	- <b>B</b>			-				10	Acesss	Admin	
Screen Control								START-1	Password		
Control Control				2010 M							UI DESI
Screen Display	× -							COMP-7	S	IGN OUT	
	o										
DISPLAY PROPERTIES								READY-70		MODIFY	
				10 10 10 10 10				OK 76	-		
	3.0				7			01470		09:53	
<ul> <li>Information</li> </ul>	A							NG-77			MODE
DisplayName SchedulerDis	play #								Model Co	ontrol	00
Name SchedulerDispla	y #0								ALIGNN	AENT 1 V	
Type CVAiO.Bplus.	GUI.C							CI II T			Q Q
OwnerScheduler		1.00						Scheduler Time		NEW	00
How 0 5								Key 1	4		CALIR
Column 0		1.00						Complete 3	4   (	COPY	CALID
Calump Same 72								Wete OK /NG 2		DELETE	
GUID a767adRo 1fa6	19725					. 16	172/	11110 010 140 2		JELETE	
IsSelected True	+372-0 H 2048	x W 2448 [-2.	20,1631]=-	M Auto-Hit L C-Lin	ne 🗌 Axis M	xel 🗸	1/4 4	Write Data 2	9		
IsEditMode True	Sahad	ulor Log		🖸 🐸 Caurata 50	- C T D	County (50		Judgement 2	8 Basic IO	Communication	600
<ul> <li>2. Scheduler</li> </ul>	Juneo	ulei Log	1995 I I I	G 🗛 Courre. Jo		Court. 50		Vision Process 2	6 🔘 M	odbusTCP	SETTIN
SchedulerType SchObjectAli	n 1P Time	Sched	Messa		Time	Tack	Resull	Cala Off 2	S. C. de	2000	
✓ Scheduler	19:45:2	6 Schedul	Finaliza		19:45:26	. 346	OK	Calc UTT 2	System F	veset	-
SchedulerNar SchedulerDis	play #					and the second s		Calc On 1	8		
M IECommunicat								Trigger 1 Off 1	5 10 State		
+ II Communica								Trigger 1 On 2	4		
ComIO ModbusTCP									1 (• F	(eset '0'	HISTOR
ComIO ModbusTCP	- II										
ComIO ModbusTCP	<u> </u>								- A	live - '64'	
ComIO ModbusTCP	Ť.									Vive - '64'	
ComIO ModbusTCP Name Name									A	Vive '64' eady '65'	
Name					<		>			Vive - '64'	
Name RELOAD					<		>	< >		Vive - '64' sady - '65' START	
Name RELOAD				Object Aligment - Rob	KOT Pickup with Sin	gle Tool	>	<u> </u>		Vive - '64' eady - '65' START	Ľ
Name Name Name				Object Aligment - Rob MAIN	Tot Pickup with Sin	gle Tool	>	3		Vive - '64' eady - '65' START STOP	
Name ReLOAD SAVE 6				Object Aligment - Rob	The second secon	gle Tool	>	3		Vive - '64' eady - '65' START STOP	EXIT

- 1) UI Design<sup>(1)</sup> Lựa chọn hiển thị màn hình thiết kế GUI 2) Main Display<sup>(2)</sup> Danh sách các đối tượng hiển thị **Screen control** Lựa chọn chuyển đổi giữa các màn hình hiển thị 3 GUI Library<sup>(3)</sup> Thư viện các đối tượng hiển thị<sup>(4)</sup> 4) Hiển thị các đặc tính của đối tượng hiển thị được chọn Display Property<sup>(5)</sup> (5) **Reload/Save-**Mở lại / lưu các cài đặt hiện tại của model (6)
- (1) Cần phải đăng nhập với quyền Engineer hoặc Admin để có thể chuy cập cửa sổ này. Cửa sổ sẽ tự động chuyển về màn hình Main sau 10 phút không có tác động chuột nào.
- (2) Ngoại trừ đối tượng ScreenDisplay, tất cả các đối tượng còn lại đều có thể tùy biến vị trí theo yêu cầu của người dùng. Số lượng SchedulerDiplay cũng chính là số lượng luồng xử lý ảnh sẽ được dùng trong chương trình. Xem chi tiết tại <u>Section 2.3</u>
- (3) Việc lựa chọn sử dụng đối tượng hiển thị nào tùy thuộc vào yêu cầu của từng bài toán xử lý ảnh. Luồng xử lý ảnh được cấu trúc tách biệt với luồng hiển thị nên số lượng đối tượng hiển thị không làm ảnh hưởng đến thời gian (tack time) xử lý của chương trình.
- (4) Cài đặt liên quan đến các đối tượng hiển thị có thể xem chi tiết tại Chapter 2.
- (5) Tất cả các thay đổi liên quan đến thuộc tính đều được lưu lại trong log file và không được lưu lại cho đến khi người dùng lựa chọn SAVE toàn bộ model đang được sử dụng.

Note: Để đảm bảo mọi thao tác đã được lưu lại, sau khi lựa chon SAVE nên thử RELOAD và kiểm tra lại dữ liệu 1 lần nữa.

#### Model – Giao diện thiết kế Vision Process

	MACHINE VISION	CVAIO B+	AIO MATRIX	No #86	Date Time	2022 - 12 - 17 19:52:15	License Version	Activated 5.0.1	CPU   2%	C Drive 937221GB B D Drive 23/244GB
	TOOL LIBRARY         Acquation         Image Load         Image Load         Image Load         Image Load         Image Load         Image Save         Regin Tool         Compate Match Ex         Uight Control         Geometry - Greation         Geometry - Geometry - Monay & Fig         Geometry - Intersection         InTigger124         InTigger				NAN Nanci Control Cont	rruns Truns Truns tr			Algoridgetting Algoridgetting	
1	Model <sup>(1)</sup>		Lựa ch	iọn hiển tl	hị màr	n hình thiế	ết kế ∖	/ision P	rocess	
2	Process Control		Manu	các điều k	khiển c	chức năng	g của	Vision F	Process	
3	Process Display	(2)	Hiển th	nį Schedu	ler và	các công	cụ xủ	r lý ảnh		
4	Process Status		Hiển th	nị trạng th	ái của	Vision P	roces	S		
5	Scheduler Proce	ess <sup>(3)</sup>	Lựa ch	iọn các pr	rocess	hiện có	của cł	nương ti	rình	
6	Tool Library		Danh s	sách các d	công c	ụ xử lý ả	nh			

- (7) **Tool Properties**<sup>(4)</sup> Hiển thị các đặc tính của công cụ xử lý ảnh được chọn
- (1) Trong lần đầu tiên được lựa chọn, chương trình sẽ mất thời gian để load tất cả các thư viện cần thiết. Đặc biệt là trong trường hợp có sử dụng thư viện của Cognex Vision Pro
- (2) Chỉ có các Input và Output cùng loại mới có thể kết nối được với nhau. Thông tin chi tiết về các loại đầu vào, đầu ra tham khảo <u>Section 3.1</u>. Công cụ Scheduler và Algorithm là bắt buộc để đảm bảo các quá trình xử lý có thể thực hiện được.
- (3) Tất cả các process được hiển thị tại đây. Trong trường hợp Scheduler chưa được chọn, Vision Process được gán cho giá trị Null
- (4) Tất cả sự thay đổi liên quan đến đặc tính của các công cụ đều được lưu lại trong log file.

Note: Sau khi tất cả các tool được thực thi, cần phải ấn Stop để đưa toàn bộ process về trạng thái sẵn sàng

#### Calibration – Giao diện căn chỉnh tự động



- (1) Calibration<sup>(1)</sup> Lựa chọn hiển thị màn hình căn chỉnh tự động 2 Calibration Control<sup>(2)</sup> Manu các điều khiển quá trình căn chỉnh chạy single run 3) **Calibration Display** Hiến thị Calibration Scheduler và các công cụ cho căn chỉnh Auto (4) **Calibration Status** Hiển thị trạng thái của quá trình căn chỉnh (5) Interface Status<sup>(3)</sup> Lựa chọn các process hiện có của chương trình **Tool Properties**<sup>(4)</sup> (6) Hiển thị các đặc tính của công cụ xử lý ảnh được chọn (7) Calibration Auto<sup>(5)</sup> Chạy hoặc hủy quá trình căn chỉnh tự động
- (1) Cần phải đăng nhập với quyền Engineer hoặc Admin để có thể chuy cập cửa sổ này. Đây là màn hình cho chế độ căn chỉnh tự động có liên kết với PLC/PC. Tham khảo thêm chế độ căn chỉnh manual ở <u>Section 4.2</u>
- (2) Nút Run chỉ có tác dụng chạy Single Run để kiểm tra cài đặt của từng công cụ xử lý ảnh. Nút New sẽ tạo ra 1 process căn chỉnh mới.
- (3) Tham khảo Interface Sequence ở Section 4.3. Interface Type và các thông số cài đặt liên quan đến căn chỉnh độc lập so với các thông tin Interface của chương trình chính.
- (4) Tất cả sự thay đổi liên quan đến đặc tính của các công cụ đều được lưu lại trong log file.
- (5) Quá trình căn chỉnh sẽ chia làm 2 giai đoạn: Translation và Rotation và sẽ được thực hiện liên tục

#### Setting - Giao diện cài đặt liên quan

eset 3 Start Trigger 1 Trigger 2 Trigger 3 Trigger 4 Calc	Ready     Ready     Trigger 1 Ack     Trigger 2 Ack     Trigger 3 Ack     Trigger 4 Ack     Calc Ack		This is for user de	esign	
Start Trigger 1 Trigger 2 Trigger 3 Trigger 4 Calc	Ready     Trigger 1 Ack     Trigger 2 Ack     Trigger 3 Ack     Trigger 4 Ack     Calc Ack			-	MAIN
Trigger 1 Trigger 2 Trigger 3 Trigger 4 Calc	<ul> <li>Trigger 1 Ack</li> <li>Trigger 2 Ack</li> <li>Trigger 3 Ack</li> <li>Trigger 4 Ack</li> <li>Calc Ack</li> </ul>				
Trigger 2 Trigger 3 Trigger 4 Calc	Ingger 1 Ack     Trigger 2 Ack     Trigger 3 Ack     Trigger 4 Ack     Calc Ack				
Trigger 3 Trigger 4 Calc	Trigger 2 Ack     Trigger 3 Ack     Trigger 4 Ack     Calc Ack	{			
Trigger 3 Calc	Ingger 3 Ack     Trigger 4 Ack     Calc Ack				ULDESIG
Calc	Ingger 4 Ack     Calc Ack				
Calc	Calc Ack	{ [ ]			
Cohodul	lor 3	-			i 💻
Jonedal		-			MODEL
Start	<ul> <li>Ready</li> </ul>	J			
Trigger 1	Trigger 1 Ack				
Trigger 2	Trigger 2 Ack				CALIB
Trigger 3	Trigger 3 Ack				li 🥱
Trigger 4	Trigger 4 Ack	1 I I I I I I I I I I I I I I I I I I I			!i <b>(1</b> )
Calc	Calc Ack	1 i i			SETTING
Schedul	ler 3				
Start	Ready				
Trigger 1	Trigger 1 Ack	n i i			HISTORY
Trigger 2	Trigger 2 Ack	í !!			
Trigger 3	Trigger 3 Ack	í I I			i ;= 2
	Trigger 4 Ack	1			
Trigger 4	Calc Ack	1 <b>i i</b>			
Trigger 4 Calc					
Trigger 4 Calc					Copylight by AIO MATRIX
	Calc	Calc Calc Ack	Calc Ack	Calc Calc Ack	Celc Celc Ack

1 Setting <sup>(1)</sup>	Lựa chọn hiển thị màn hình cài đặt Setting
2 User Design UI <sup>(1)</sup>	Màn hình mở để người dùng có thể thiết kế thêm tính năng khác
3 Interface Status <sup>(2)</sup>	Trạng thái On/Off của từng IO trong các Interface tương ứng
4 Interface Control	Chạy hoặc dừng quan sát trạng thái của Interface
5 Interface Properties	Hiển thị đặc tính của Interface được chọn
6 Load Save	Tải hoặc lưu lại thông tin Interface

(1) Người dùng có thể thiết kế thêm phần hiển thị trong project "CVAiO.Bplus.SettingGUI"

(2) Tất cả các thông tin về interface của chương trình chính sẽ được liên kết và hiển thị tại đây. Trong trường hợp chương trình chính đang ở chể độ Auto thì người dùng sẽ không thể thay đổi các đặc tính của Interface và người dùng có thể quan sát trạng thái làm việc hiện tại của các IO (Monitor Mode)

### History – Giao diện lịch sử Log và hình ảnh

MACHINE VISION PLATFORM	CVAIO B+	AIO MATRIX	K No	o #86	Date         2022 - 12 - 17           Time         19:55:05	License Activated Version 5.0.1	CPU         2%         C Drive         93/221GB           RAM         11/15.8GB         D Drive         23/244GB
Start Date	No	Date	Time	Туре	Details		
Friday , December 16, 2022	222524	2022-12-17	19:51:34.316	Tool	Tool Selected: Scheduler		
End Data	222525	2022-12-17	19:51:36.389	Tool	Tool Selected: Light Control 1	(2)	
Saturday December 17 2022	222526	2022-12-17	19:51:37.974	Tool	Tool Selected: Light Control 0		
	222527	2022-12-17	19:51:41.570	Tool	LightControl - Parameter chang	ed: Name Light On	MAIN
😑 System 🚺 📀 Schedu	ler 222528	2022-12-17	19:51:43.797	Tool	Tool Selected: Light Control 1		
Procese Calibrati	222529	2022-12-17	19:51:49.651	Tool	LightControl - Parameter chang	ed: Name Light Off	
	222530	2022-12-17	19:51:54.592	Tool	Tool Selected: AlgoObjectAlign	1P 0	
🔵 Tool 🔵 Debug	222531	2022-12-17	19:51:57.472	Tool	Tool Selected: Light Off		UI DESIGN
Error O Exception	on 222532	2022-12-17	19:52:14.518	Tool	Tool Selected: Scheduler		
	222533	2022-12-17	19:52:24.809	Tool	Tool Selected: Template Match	0 ۱	
License	222534	2022-12-17	19:52:26.840	Tool	Tool Selected: Fixture 0		
	222535	2022-12-17	19:52:28.737	Tool	Tool Selected: Light Off		MODEL
REFRESH	222536	2022-12-17	19:52:32.256	Tool	Tool Selected: Scheduler		MODEL
Start Date	222537	2022-12-17	19:52:43.877	Tool	Tool Selected: Acquisition 0		0,00
Friday , December 16, 2022	222538	2022-12-17	19:52:44.222	Tool	Tool Opened: Acquisition 0		1 200
Ford Data	222539	2022-12-17	19:52:46.928	Tool	Tool Selected: Template Match	10	000
Saturday December 17 2022	222540	2022-12-17	19:52:47.235	Tool	Tool Opened: Template Match	0	CALIB
	222541	2022-12-17	19:53:10.442	Tool	Scheduler - Parameter change	d: Name Calibration Scheduler 0	
🔴 OK Display 📄 🔴 OK	222542	2022-12-17	19:53:10.442	Tool	Template Match - Parameter ch	anged: Name Template Match 0	27.15
NG Diselar	222543	2022-12-17	19:53:10.443	Calibration	Auto Calib Tool Created: Temp	late Match 0 CVAiO.Bplus.TemplateMatch	I.TemplateMatch
	222544	2022-12-17	19:53:10.443	System	Openning Calibration Window		SETTING
	222545	2022-12-17	19:53:10.443	Tool	ImageAcquisition - Parameter c	hanged: Name Acquisition 0	<u>Set tille</u>
	222546	2022-12-17	19:53:10.444	Tool	FindCalibMatrix - Parameter cha	anged: Name Find Calib Matrix 0	
	222547	2022-12-17	19:53:10.444	Calibration	Auto Calib Tool Created: Find C	Calib Matrix 0 CVAiO.Bplus.Calibration.Find	Calib Matrix
	222548	2022-12-17	19:53:10.445	Calibration	Auto Calib Tool Created: Calibr	ation Scheduler 0 CVAiO.Bplus.AutoCalibr	ation.Scheduler
	222549	2022-12-17	19:53:10.445	Calibration	Auto Calib Tool Created: Acqui	sition 0 CVAiO.Bplus.Acquisition.ImageAc	uisition
	222550	2022-12-17	19:53:13.732	Calibration	Tool Selected: Find Calib Matri	< 0	
	222551	2022-12-17	19:53:36.384	Calibration	Process load fail: D:\00 CVAiO	\01 Solution Release\CVAiO.Bplus\Debu	g\aaa.vpj
	222552	2022-12-17	19:54:19.987	System	Openning SettingGUI Window		
	222553	2022-12-17	19:54:21.329	System	Openning History Window		
DEEDEO							
REFRESH	!i						
							Copyright by AIO MATRIX 2022

- History Lựa chọn hiển thị màn hình lịch sử Log
   History Display Thông tin lịch sử log hoặc hình ảnh
   Log History Control<sup>(1)</sup> Lựa chọn hiển thị thông tin lịch sử Log
- (4) Image History Control Lựa chọn hiển thị lịch sử hình ảnh (Raw Image & Display Image)





Chương trình được cấu trúc thành các module riêng biệt bao gồm có module giao tiếp PLC, module giao tiếp với camera, module giao tiếp với bộ điều khiển đèn và các module xử lý ảnh. Chương trình chính được thiết kế để có thể thêm bớt các process vision một cách dễ dàng và các process vision được thêm vào đều có thể liên kết với các module còn lại để thiết lập kết nối với phần cứng. Bản thân bên trong các module giao tiếp với phần cứng, các lớp tương ứng với các phần cứng khác nhau cũng được phân chia rõ ràng để có thể lựa chọn cho phù hợp với từng hệ thống thực tế. Mỗi thao tác bốc thả <u>Scheduler Display</u>, một vision process sẽ được thêm vào cung cấp cho người dùng khả năng thiết kế một luồng xử lý ảnh. Việc điều khiển hoạt động của từng process vision tuân theo IO sequence như mô tả ở hình trên đây.Tham khảo thêm vai trò của các IO ở <u>section 2.3</u> và <u>section 3.10</u>.



#### 1.4 Setup Guide – Các bước để thiết lập chương trình

Dưới đây là chi tiết các bước để thiết lập một chương trình xử lý ảnh

#### Step 1: Khởi động chương trình

• Chương trình: CVAiO.Bplus.exe

Name	Date modified	Туре
🗰 CVAiO.Bplus	2/21/2023 11:28 AM	Application
Basler.Pylon.dll	4/3/2018 11:45 AM	Application exten.
CLIDelegate.dll	9/27/2021 9:57 PM	Application exten.

#### Step 2: Đăng nhập người dùng – User login

Nhập user & password sau đó bấm log in để đăng nhập tài khoản người quản trị. Phải đăng nhập với quyền admin hoặc engineer mới có thể thiết lập chương trình mới. Chi tiết tham khảo <u>Section</u> <u>5.2</u>

#### \* Step 3: Khởi tạo Model - Recipe

 Tạo model mới để quản lý toàn bộ các cài đặt liên quan. Trong trường hợp muốn tạo dựa trên 1 model hiện có, người dùng có thể sử dụng tính năng COPY. Dữ liệu liên quan đến model được lưu tại C:\CVAiO.Bplus\Recipe

#### Step 4: Cài đặt giao tiếp với PLC

- Chuyển sang màn hình UI Design để bắt đầu thiết kế GUI
- Chuột trái vào màn hình chính hoặc chuyển lựa chọn properties bên trái thành COMMON PROPERTIES.
- Lựa chọn loại giao tiếp cho dữ liệu IO On/Off tại IOType
- Lựa chọn loại giao tiếp cho dữ liệu data tại DataType
- Tùy theo loại giao tiếp được chọn của IOType và DataType, các lựa chọn tương ứng sẽ mở ra để cài đặt thêm thông tin giao tiếp
- Cài đặt địa chỉ IO cho các tin hiệu Reset, Alive, Ready. Xem chi tiết tại Section 1.3

#### Step 5: Thiết kế giao diện hiển thị 1

- Lựa chọn số màn hình hiện thị bằng cách bốc thả đối tượng <u>ScreenDisplay</u>. Lựa chọn tên hiển thị và chọn ScreenDetach là True nếu muốn hiển thị ra 1 màn hình riêng biệt.
- Lựa chọn số lượng luồng xử lý ảnh bằng cách bốc thả đối tượng <u>SchedulerDisplay</u>.
- Mỗi Scheduler sẽ có 1 GUID không trùng lặp để sau đó các đối tượng hiển thị khác sẽ lựa chọn để kết nối và lấy dữ liệu hiển thị.
- Chọn vào từng Scheduler sau đó thay đổi các thuộc tính liên quan bao gồm SchedulerType và các
   IO liên quan đến logic hoạt động của scheduler. Chi tiết tham khảo <u>section 2.3</u>
- Bấm Save để lưu lại các thiết kế hiện tại.

Cửa sổ thiết lập cài đặt chương trình

258A 6071	CVAIO B+ 5.0.1	Date	2023 - 02 - 21
2 1 0 0 HATRIX	Copyright by AIO MATRIX	Time	11:50:14



×	1. Display		
	Rows	60	
	Columns	72	
	ShowGrid	True	
	EnableMove	True	
~	2. IF Communi	cation	
	IOType	ModbusRTU	
>	ComIO	ModbusRTU	
	DataType	MelsecTCP	
>	ComData	MelsecTCP	
	InReset	0	
	OutAlive	64	
	OutReady	65	
~	3. Log File		
	MaxLogDate	5	
	SavelmageOk	False	
	SavelmageNg	True	
	SaveDisplayImag	E False	
	0 0 1 1	<b>T</b>	

GUI LIBRARY	SchedulerDisplay #1	WaitError	ResetAck 1000 ms 🚺
Bisplay Control     SchedulerDisplay     InterfaceDisplay     GridViewDisplay     GridViewDisplay     GridViewDisplay     GraphDisplay     ListViewDisplay     Screen Control     Screen Display	Lin Lin	spection Scheduler	
DISPLAY PR	operties	Alignment Scheduler	]
▼     2. Scheduler SchedulerType     SchedulerInspectio       > Scheduler     SchedulerInspectio       > SchedulerNar     SchedulerDisplay #       > ComPot ModbusRTU     InStat 1       InComple 7     OutRead 70       OutRead 70     OutOK 76       OutNK 76     OutNG 77       > ComData Melsec TCP     DataAdd 200       SchedulerTar 0     DicDisplay       CVARead 70.     CVA0. Bplus.GUI.Core       ListWewDspl     ✓	SchedulerDisplay #2 (Inspection)	WaitError	ResetAck 1000 ms 👔
IsSelected	Màn hình chír	nh Màn hình	phụ
SAVE	INSPECTION GUI	ALIGNMEN	T DISPLAY

Step 6: Thiết kế thuật toán xử lý ảnh

- Chuyển sang màn hình Model để bắt đầu thiết kế process
- Bấm vào từng Scheduler phía dưới để hiển thị từng process



- Chuột phải vào công cụ Scheduler để thêm hoặc bớt số lượng trigger. Tham khảo thêm về Scheduler tại <u>Section 3.10</u>
- Lựa chọn công cụ Algorithm rồi kéo thả vào màn hình. Mỗi process chỉ được sử dụng 1 công cụ Algorithm. Tham khảo thêm về Algorithm tại <u>Section 3.11</u>

- Kéo thả các công cụ xử lý ảnh khác tùy theo yêu cầu của bài toán
- Click đúp vào công cụ Scheduler để bật-tắt các trigger trước khi bấm Start để chạy, Stop để reset.

#### Step 7: Thiết kế giao diện hiển thị 2

- Chuyển sang màn hình UI Design để bắt đầu thiết kế các thành phần còn lại của GUI
- Bốc thả các đối tượng tùy theo yêu cầu của bài toán vision.
- Lựa chọn OwnerScheduler (Scheduler mà từng đối tượng hiển thị sẽ kết nối vào để lấy thông tin) cho từng đối tượng hiển thị.
- Điều chỉnh thuộc tính của các đối tượng hiển thị theo hướng dẫn ở Chapter 2
- Bấm Save để lưu lại các thiết kế hiện tại.

#### Step 8: Hoàn thiện và chạy thử với chương trình mô phỏng

- Trong trường hợp không có PLC để kết nối trực tiếp, người dùng có thể sử dụng bộ mô phỏng PLC với module interface Modbus TCP/IP.
- Quay lại Step 4 và điều chỉnh toàn bộ lựa chọn IOType và DataType thành ModbusTCP (127.0.0.1).
- Khởi động chương trình mô phỏng PLC
- Bấm START trên chương trình Vision để chạy chương trình
- Bấm START bên chương trình mô phỏng để chạy mô phỏng và kết nối với chương trình vision qua Modbus TCP.

# CHAPTER 2: GUI LIBRAY – THƯ VIỆN CÁC ĐỐI TƯỢNG HIỂN THỊ

- 2.1 Overview Giới thiệu chung
- 2.2 Screen Display Hiển thị đa màn hình
- 2.3 Scheduler Display Hiển thị luồng xử lý ảnh
- 2.4 Interface Display Hiển thi trạng thái giao tiếp với PLC/PC
- 2.5 Image Display Hiển thị hình ảnh
- 2.6 Grid View Display Hiển thị dữ liệu dạng lưới
- 2.7 List View Display Hiển thị dữ liệu dạng danh sách
- 2.8 Graph Display Hiển thị đồ thị
- 2.9 Label Display Hiển thị nhãn

#### Các nội dung sẽ được đề cập trong chương này:

- Tổng quan về ý tưởng thiết kế màn hình hiển thị
- Các chức năng, thuộc tính chung của tất cả đối tượng hiển thị
- Chức năng, thuộc tính và cách sử dụng của từng đối tượng hiển thị
- Case Study 1: Thiết kế màn hình hiển thị cho 1 Task có 3 luồng xử lý ảnh
- Case Study 2: Chương trình hiển thị trên 2 màn hình độc lập

#### 2.1 Overview - Giới thiệu chung

Động lực cho ý tưởng thiết kế màn hình hiển thị đến từ yêu cầu thực tế của các bài toán vision trong hệ thống tự động hóa. Mỗi bài toán khác nhau sẽ yêu cầu thêm hoặc bớt các đối tượng hiển thị. Việc cố định cứng các đối tượng hiển thị như các chương trình hiện nay sẽ làm giảm đi sự linh hoạt trong trải nghiệm của người dùng. Để loại bỏ các trở ngại đó, CVAIO B+ cung cấp sự tùy biến trong giao diện hiển thị từ việc lựa chọn số lượng luồn xử lý cũng như việc tăng giảm màn hình quản lý hiển thị. Bằng các thao tác bốc thả (drag & drop) đơn giản, người dùng có thể tự lựa chọn các đối tượng hiển thị và sắp xếp các đối tượng theo yêu cầu của từng bài toán. Với các thao tác bấm chuột đơn giản, thuộc tính của từng đối tượng hiển thị có thể được thay đổi mà không tốn quá nhiều thời gian.

Hướng dẫn thao tác thiết kế màn hình hiển thị:

- Step 1: Lựa chọn màn hình hiển thị sẽ thiết kế
- \* Step 2: Lựa chọn đối tượng hiển thị và bốc thả vào vị trí bất kỳ
- Step 3: Điều chỉnh kích thước, vị trí của đối tượng hiển thị
- \* Step 4: Lựa chọn Owner Scheduler mà đối tượng sẽ kết nối đến
- Step 5: Điều chỉnh thuộc tính của đối tượng
- Step 6: Kiểm tra thông tin và lưu lại kết quả thực hiện



#### Common Properties:

- (1) Rows: Số hàng chia của lưới trên màn hình thiết kế GUI
- (2) Colums: Số cột chia của lưới trên màn hình thiết kế GUI
- (3) ShowGrid: Ấn hiện lưới trên màn hình thiết kế GUI
- (4) EnableMove: Cho phép chỉnh sửa vị trí của các đối tượng hiển thị

w | 2

- (5) IOType: Loại giao tiếp với PLC để truyền nhận trạng thái IO
- (6) ComIO: Thuộc tính cài đặt của giao tiếp với PLC để truyền nhận trạng thái IO
- (7) DataType: Loại giao tiếp với PLC để truyền nhận Data
- (8) ComData: Thuộc tính cài đặt của giao tiếp với PLC để truyền nhận dữ liệu data
- (9) InReset: Tín hiệu IO để reset lại toàn bộ các luồng xử lý ảnh
- (10)OutAlive: Tín hiệu báo trạng thái đang hoạt động của chương trình (nháy on/off 3-5s)
- (11)OutReady: Tính hiệu báo trạng thái sẵn sàng của chương trình
- (12)MaxLogDate: Giới hạn số ngày lưu giữ lịch sử
- (13)SavelmageOk: Lựa chọn lưu hình ành raw khi kết quả xử lý OK
- (14)SavelmageNg: Lựa chon lưu hình ảnh raw khi kết quả xử lý NG
- (15)SaveDisplayOk: Lựa chọn lưu hình ảnh hiển thị trên GUI khi kết quả xử lý OK
- (16)SaveDisplayNg: Lựa chọn lưu hình ảnh hiển thị trên GUI khi kết quả xử lý NG

#### Display Properties: (Chung cho tất cả đối tượng hiển thi)

- (1) DisplayName: Tên của đối tượng hiển thị sẽ xuất hiện trên màn hình chính
- (2) Name: Tên của đối tượng hiện thị dùng để quản lý nội bộ (read only)
- (3) Type: Phân loại của đối tượng hiện thị (read only)
- (4) OwnerScheduler: Scheduler mà đối tượng hiển thị sẽ được đính vào để lấy dữ liệu
- (5) Row: Vị trí tính theo hàng
- (6) Column: Vị trí tính theo cột
- (7) RowSpan: Độ rộng tính theo số hàng
- (8) ColumnSpan: Độ rộng tính theo số cột
- (9) GUID: ID định danh của đối tượng (read only)
- (10)IsSelected: Trạng thái lựa chọn (read only)
- (11)IsEditMode: Trạng thái chỉnh sửa (read only)

#### \* Note:

Các trạng thái read only là các thuộc tính được sử dụng trong nội bộ chương trình, người dùng không thể điểu chỉnh các thuộc tính này.

#### 2.2 Screen Display - Hiển thị đa màn hình

DIS	PLAY PROPER	TIES ~			
	<b>≵</b> ↓ 🖻				
~	1. Information				
	DisplayName	ALIGNMENT DISPLAY			
	Name	ScreenDisplay #1			
	Туре	CVAiO.Bplus.GUI.Core			
	OwnerScheduler				
	Row	58			
	Column	36			
	RowSpan	2			
	ColumnSpan	36			
	GUID	faa91f6d-26b1-4cc0-b852-			
	IsSelected	True			
	IsEditMode	True			
~	2. Screen Disp	lay			
	ScreenDetach	True			
Na	Name				
Nar	ne				

Mục đích: Tăng giảm số lượng màn hình hiển thị để đáp ứng trong trường hợp người dùng sử dụng nhiều luồng xử lý ảnh, đặc biệt là trong trường hợp muốn hiển thị trên nhiều pc monitor.

#### Display Properties:

(1) ScreenDetach: Tách màn hình ra hiển thị ở 1 cửa sổ riêng biệt

#### 2.3 Scheduler Display – Hiển thị luồng xử lý ảnh

#### SchedulerDisplay #0 (Std)

DISPLAY PROPERTIES ~						
•	Az↓	1-1				
~	2. S	cheduler		~		
	Sche	dulerType	StdScheduler			
~	Sche	duler				
	5	SchedulerNar	SchedulerDisplay #			
	$\sim$ 1	FCommunicat				
		ComIO	ModbusRTU			
		InStart	1			
		InComple	7			
		OutRead	70			
		OutOK	76			
		OutNG	77			
		ComData	MelsecTCP			
		Data Addı	200			
	5	SchedulerSta	Ready			
	5	SchedulerTac	0			
	[	DicDisplay	CVAiO.Bplus.GUI.Core			
	L	.istViewDispla				
	0	Grid View Displ		5		
		D: I		÷		
Na	me					
Nar	ne					

WaitErrorResetAck 1000 ms 🕧

Mục đích: Tăng giảm số lượng luồng xử lý ảnh, lựa chọn loại luồng xử lý (bao gồm các bước trước khi xử lý ảnh, truyền nhận dữ liệu với PLC). Người dùng có thể tự thiết kế 1 logic phù hợp với yêu cầu của từng bài toán. Chi tiết về việc thiết kế tham khảo tại Section 6.3.

#### **Display Properties:**

- (1) SchedulerType: Loại luồng xử lý ảnh,
- (2) ComIO: Loại giao tiếp cho dữ liệu IO On/Off (Melsec TCP, Modbus, Serial Port)
- (3) InStart: Input báo hiệu bắt đầu quá trình xử lý
- (4) InComplete: Input báo hiệu kết thúc quá trình xử lý
- (5) OutReady: Output báo luồng xử lý ảnh đang ở trạng thái sẵn sàng
- (6) OutOk: Output báo kết quả xử lý và phán định OK
- (7) OutNg: Output báo kết quả xử lý và phán định NG
- (8) ComData: Loại giao tiếp cho dữ liệu Data
- (9) DataAddr: Vị trí bắt đầu ghi dữ liệu Data

• Muc đích: Hiển thi trang thái hiên tai của

các IO giao tiếp với PLC của luồng xử lý ảnh

được đính vào. Người dùng cần lựa chon luồng xử lý ảnh tương ứng ở thuộc tính Owner 2

#### 2.4 Interface Display – Hiển thị trạng thái giao tiếp với PLC/PC

DI	SPLAY PROPER	RTIES		• Mục dich: Hiến thị trạng thai hiện tại c
•	<b>2</b> ↓ 🖻		IO IF ModbusRTU	các IO giao tiếp với PLC của luồng xử lý ả được đính vào. Người dùng cần lựa chon luồ
>	1. Information		Data IF	
$\sim$	2. Interface D	isplay	MelsecTCP	xử lý ann tương ứng ở thuộc tinh Owr
~	Interface		- 10	Scherduler.
	ComIO	ModbusRTU	Start - 1	
	InStart	1		
	InComplete	7	Comp - 7	
	OutReady	70	Ready - 70	
	OutOK	76		
	OutNG	77	UK - 76	
	ComData	MelsecTCP	Ng - 77	
	DataAddr	200		1
			• <u>Display Prop</u> (1) Cor <i>TCP, Modbus, S</i>	erties: nIO: Loại giao tiếp cho dữ liệu IO On/Off (Melsec erial Port)
			(2) INST	tart: Input bao niệu bat dau qua trinn xử lý
			(3) INC	omplete: Input bao nieu ket thuc qua trinn xu iy
			(4) Out	Ready: Output báo luông xứ lý ánh đang ở trạng
			thái săn sàng	,
1.	Information		(5) Out	<b>Ok:</b> Output báo kêt quả xử lý và phán định OK
			(6) Out	<b>Ng:</b> Output báo kết quả xử lý và phán định NG
			(7) Cor	nData: Loại giao tiếp cho dữ liệu Data
_			🚽 (8) Data	aAddr: Vị trí bắt đầu ghi dữ liệu Data

#### 2.5 Image Display – Hiển thị hình ảnh

DIS	SPLAY PROPER	TIES	$\sim$
	<b>≵</b> ↓ 🖻		
	OwnerScheduler		۸
DISPL OW Ro Co Ro Co GL IsS IsE V 2. To Se Co Do Sta Au Ce Sh Co	Row	12	
	Column	36	
DISPLA OW Row Colu GU ISS ISE V 2. Too Sel Cor Doo Stal Aut Cer Sho Coo Name Name	RowSpan	22	
	ColumnSpan	23	
DISP C F C C F C C C C C C C C C C C C C	GUID	4001aab0-50c0-4f93-b	
	IsSelected	True	
	IsEditMode	True	
$\sim$	2. Image Displa	ay	
~	ToolName		
	SelectedImage	10	
	ControlVisible	False	
	Dock	Left	
	StatusVisible	False	
	AutoFit	True	
	CenterLine	False	
	Show Axis	False	
	OwnerScheduler         Row       12         Column       36         Row Span       22         Column Span       23         GUID       4001         IsSelected       True         IsEditMode       True         Selected       True         SelectedImage       10         ControlVisible       False         Dock       Left         StatusVisible       False         AutoFit       True         ShowAxis       False         Coordinate       Pixe	Pixel	v
Na Na	me me		



Mục đích: Hiển thị hình ảnh của các giai đoạn trong quá trình xử lý. Người dùng cần lựa chọn luồng xử lý ảnh tương ứng ở thuộc tính Owner Scherduler sau đó lựa chọn công cụ xử lý ảnh trong luồng đó ở thuộc tính ToolName.

- Display Properties:
  - (1) ToolName: Công cụ xử lý ảnh mà hình ảnh sẽ được lấy
  - (2) SelectedImage: vi trí trong danh sách anh của tool
  - (3) ControlVisible: Ån hiện thanh công cụ điều khiển
  - (4) Dock: Vị trí của thanh công cụ điều khiển
  - (5) StatusVisible: Ån hiện thanh công cụ trạng thái
  - (6) AutoFit: Tự động điền đầy của hình ảnh
  - (7) CenterLine: Ấn hiện đường tâm
  - (8) ShowAxis: Ấn hiện trục tọa độ
  - (9) Coordinate: Lựa chọn tọa độ hiển thị

#### 2.6 Grid View Display – Hiển thị dữ liệu dạng lưới

DIS	SPLAY PROPER	TIES ~
	<b>≵</b> ↓ 🖻	
~	1. Information	
	DisplayName	Scheduler Time
	Name	GridViewDisplay #0
	Туре	CVAiO.Bplus.GUI.Core
	OwnerScheduler	cf33f782-aebc-472b-b
	Row	10
	Column	11
	RowSpan	12
	ColumnSpan	36
	GUID	7295ead0-579c-4d57-853
	IsSelected	True
	IsEditMode	True
~	2. Grid View Di	splay
	GridName	Scheduler Time
	GridType	System.Collections.Generic
Na	me	
INA	me	

GridViewDisplay	#1
Кеу	Value
Complete	466
Write OK/NG	373
Write Data	373
Judgement	373
Vision Process	348
Calc Off	343
Calc On	249
Trigger 1 Off	218
Trigger 1 On	16

Mục đích: Hiển thị các dữ liệu dạng lưới có trong luồng xử lý ảnh tương ứng ở thuộc tính Owner Scherduler. Người dùng có khả năng thêm bớt loại dữ liệu này bằng cách sửa đổi trong mã nguồn của Scheduler tương ứng. Tham khảo <u>Section 6.3</u>

#### Display Properties:

- (1) GridName: Tên của dữ liệu dạng lưới trong Scheduler
- (2) GridType: Loại dữ liệu dạng bảng

i

#### 2.7 List View Display – Hiển thị dữ liệu dạng danh sách

DIS	SPLAY PROPER	TIES	$\sim$
•	<b>2</b> ↓ □		
	Name	ListViewDisplay #0	^
	Туре	CVAiO.Bplus.GUI.C	
Name Name	OwnerScheduler	cf33f782-aebc-472	
	Row	13	
	Column	19	
	RowSpan	14	
	ColumnSpan	41	
	GUID	225d8d52-1955-4745-{	
	IsSelected	True	
	IsEditMode	True	
$\sim$	2. List View Dis	splay	
	ListName	Scheduler Log	
	ListColumn	3	
	HeaderNames	(Collection)	
	MaxLines	50	
			۷
Na Na	me me		

Scheduler	Log	🔁 💢 Count: 50 🖨 🚺
Time	Scheduler	Message
<		>

Mục đích: Hiển thị các dữ liệu dạng danh sách có trong luồng xử lý ảnh tương ứng ở thuộc tính Owner Scherduler. Người dùng có khả năng thêm bớt loại dữ liệu này bằng cách sửa đổi trong mã nguồn của Scheduler tương ứng. Tham khảo <u>Section 6.3</u>

#### • Display Properties:

- (1) ListName: Tên của dữ liệu dạng danh sách
- (2) ListColumn: Số cột hiển thị
- (3) HeaderNames: Tên của các cột hiển thị
- (4) MaxLines: Số dòng hiển thị tối đa

#### 2.8 Graph Display – Hiển thị đồ thị

DISPLAY PROPERTIES ~						
	<b>A</b> ↓ □					
	GUID	694acdbd-217b-40c9-8	~			
	IsSelected	True				
	IsEditMode	True				
~	2. Graph Displa	ay				
	GraphName					
	CurveCount	3				
	MaxData	50				
	GraphPane	CVAiO.Bplus.ZedG				
	MainAxisTitle					
	Sub Axis Title					
	XGridVisible	False				
	YGridVisible	False				
	Zero Line Visible	True				
×	Misc					
	CurveList	(Collection)				
			$\mathbf{x}$			
MainAxisTitle Main Axis Title						

GraphDisp	lay #0						8	ដ ដ	ount: 50	+
1										
0.9 -										
0.8 -										
0.7 -										
0.6 -										
0.5 -										
0.4 -										
0.3 -										
0.2 -										
0.1 -										
0										
0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	8.0	0.9	

Mục đích: Hiển thị các dữ liệu dạng đồ thị có trong luồng xử lý ảnh tương ứng ở thuộc tính Owner Scherduler. Người dùng có khả năng thêm bớt loại dữ liệu này bằng cách sửa đổi trong mã nguồn của Scheduler tương ứng. Tham khảo <u>Section 6.3</u>

#### Display Properties:

- (1) GraphName: Tên của dữ liệu dạng đồ thị
- (2) CurveCount: Số lượng dữ liệu
- (3) MaxData: Số lượng điểm tối đa của dữ liệu
- (4) MainAxisTitle: Tiêu đề của trục chính
- (5) SubAxisTitle: Tiêu đề của trục phụ
- (6) XGridVisible: Hiển thị lưới của trục X
- (7) YGridVisible: Hiển thị lưới của trục Y
- (8) ZeroLineVisible: Hiển thị được tọa độ 0

#### Case Study 1: UI Design

Giao diện của một chương trình có 3 luồng xử lý ảnh sử dụng truyền thông modbus TCP/IP.

MACHINE VISION CVAI	O B+ AIO MATRIX - INSPECTION AND ALIGNM	Date         2023 - 02 - 07         License         Activated         CPU         2%         C Drive         221212           IENT         No #86         Time         22:54:32         Version         5.0.1         RAM         585         S.0B         Drive         1/2446B
GUI LIBRARY  GUI LIBRARY  GUI LIBRARY  GulderDaplay  Construction  GulderDaplay  GulderDaplay  GulderDaplay  GulderDaplay  CompOlliplay  CompOlliplay  DISPLAY PROPERTIES  1. Information  DisplayName GridViewDisplay #1  Name GridViewDisplay #1  Name GulderDaplay  CVMO Bplus GUI Comp  Compoledate  CVMO Bplus GUI Comp  Compoledate  Compoledate  Compoledate  Compoledate  Compoledate  Compoledate  Compoledate  GulderDaplay  GulderDaplay  GulderDaplay  GulderDaplay  GulderDaplay  CVMO Bplus  GulderDaplay  GulderDaplay  GulderDaplay  GulderDaplay  CVMO Bplus  GulderDaplay  GulderDaplay  GulderDaplay  GulderDaplay  GulderDaplay  GulderDaplay  CVMO  BullerDaplay  GulderDaplay  GulderDapla	Image Load 0         WaitErrorResetAck         1000 ms           ModbusTCP         Image Load 0         <	Image Load 0       Image Load 0 <td< td=""></td<>
Row 29 Column 12 RowSpan 17 ColumSpan 17 ColumSpan 12 GUID 9825679-007-4342-9a30 IsSelected True IsEdtMode True Selected True Critic Ver Display Grid Name Scheduler Time Grid Type System Collections Generic	Header0         Header1         Header2           20:59:49         Schedul         Calc Active state : 1195         OK           C         Schedul         Calc Active state : 1195         OK           Calc On         1195         Tingger 1 Off         Calc On         Schedul           Tingger 1 Off         218         Tingger 1 Off         Schedul         Schedul           Viato Process         343         Calc Off         343         Calc Off         243           Calc On         249         Tingger 1 Off         218         Tingger 1 Off         220         Tingger 1 Off         220           Tingger 1 On         16         Tingger 1 Off         220         Tingger 1 Off         220         Tingger 1 Off         220	2       Header0       Header1       Header1       Header2       COPY         205947       Schedul       Finalization Done       COPY       CAUB         2       205947       466       OK       DELETE       Basic 10 Communication         3       C       SchedulerDisplay #2 (Std)       WaitErrorResetAck       1000 ms ()       System Reset       System Reset         10       If       Image Load 0       ()       Algoinspecified 0       ()       System Reset       I)       System Reset       I)       System Reset       I)       IIII System Reset       IIIII System Reset       IIIII System Reset       IIIII System Reset       IIIII System Reset       IIIIIIIIII System Reset       IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
Name Name RELOAD SAVE	LabelDisplay #0	List ViewDisplay #4 Court: 50 C Court: 50
	A	EXIT Copyright by AIO MATRIX 202

#### Case Study 2: Multi Screen



# CHAPTER 3: TOOL LIBRARY – THƯ VIỆN CÁC CÔNG CỤ XỬ LÝ ẢNH

#### <u>3.1 Overview – Giới thiệu chung</u>

- 3.2 Common Tool Acquisition, Imag File, Region, Template Match, Light Control
- 3.3 Calibration & Fixturing Find Chessboard, Find Calib Matrix, Fixture
- <u>3.4 Deprecated QR Decode</u>
- 3.5 Geometry Creation Create Circle, Create Line, Create Rectangle
- 3.6 Geometry Finding & Fitting Find Line, Find Circle, Find Corner, Find Points
- 3.7 Geometry Measurement Distance Points, Distance Line, Distance Circle
- <u>3.8 Geometry Intersection Circle x Circle, Line x Circle, Line x Line</u>
- 3.9 Image Processing Add Weighted, Binarization, Blob, Histogram, Morphology
- 3.10 Scheduler Công cụ điều phối các nhánh xử lý
- 3.11 Vision Algorithm Thuật toán xử lý ảnh

#### Các nội dung sẽ được đề cập trong chương này:

- Tổng quan về các công cụ xử lý ảnh
- Giải thích chi tiết các công cụ xử lý ảnh, đầu vào, đầu ra, thông số cài đặt
- Case Study 1: Kiểm tra bản mạch sử dụng histogram
- Case Study 2: Chương trình hiển thị trên 2 màn hình độc lập

#### 3.1 Overview - Giới thiệu chung

Toàn bộ các công cụ xử lý ảnh sẽ được điều khiển bởi công cụ Scheduler. Đây là công cụ mặc định được tạo ra trong các process xử lý ảnh. Trong công cụ này sẽ có các tín hiệu IO để giao tiếp với PLC thông qua module giao tiếp đã được chon trong phần thiết kế GUI. Mỗi khi nhận được tín hiệu On từ PLC, Scheduler sẽ kích hoạt các Trigger thành active và từ đó tất cả các công cụ xử lý ảnh được nối với Trigger đó sẽ được thực thi theo concept lan truyền dọc theo các đường nối. Có tổng cộng 4 Trigger trong mỗi Scheduler. Trong trường hợp không sử dụng hết cả 4 triggers, người dùng có thể chọn chuột phải và remove. Sau khi tất cả các triggers cần thiết đã được thực thi, PLC sẽ On 1 tín hiệu Calc (nối từ Scheduler đến Algorithm) để báo cho công cụ Algorithm biết toàn bộ các tính toán đã thực hiện xong, tiếp đến công cụ Algorithm sẽ tổng hợp và phát định kết quả trước khi toàn tất toàn bộ các tính toán của 1 bài toán xử lý ảnh.

Hình dưới đây mô tả 1 process vision dùng để xác định vị trí của vật thể sử dụng 1 camera và 2 lần chụp riêng biệt sử dụng trigger 1,2.

- \* Step 1: Khởi tạo toàn bộ các params của tất cả các công cụ xử lý ảnh
- Step 2: Robot mang camera di chuyển đến vị trí chụp thứ nhất. Kích hoạt trigger 1 để thực hiện việc thu thập ảnh, truyền hình ảnh sang công cụ Template Matching 1 để tìm kiếm vị trí điểm mark. Kết quả tìm được sẽ được truyền sang công cụ thuật toán và chờ xử lý tiếp.
- Step 3: Sau khi robot di chuyển sang vị trí chụp tiếp theo. Kích hoạt trigger 2 để 1 lần nữa thực hiện việc thu thập hình ảnh và tìm kiếm vị trí điểm mark bằng công cụ Template Matching 2. Kết quả tìm được sẽ vẫn được truyền sang công cụ thuật toán và chờ xử lý tiếp.
- Step 4: Sau khi quá trình thu thập kết quả kết thúc. Kích hoạt tín hiệu Calc để truyền tín hiệu yêu cầu thực hiện tính toán kết quá và đưa ra phát định về cho các luồng xử lý để truyền dữ liệu sang cho PLC.



Hướng dẫn thao tác thiết kế thuật toán xử lý ảnh của từng Scheduler:

- Step 1: Lựa chọn Vision Process để bắt đầu thiết kế
- Step 2: Điều chỉnh số lượng trigger của Scheduler. Tham khảo Section 3.10
- Step 3: Bốc thả thuật toán Algo-Tool xử lý ảnh tương ứng. Tham khảo Section 3.11
- Step 4: Bốc thả các công cụ xử lý ảnh theo yêu cầu của từng bài toán
- Step 5: Kết nối các đầu ra và đầu vào của các công cụ theo yêu cầu của bài toán
- Step 6: Nháy đúp chuột vào từng công cụ xử lý ảnh để mở cửa sổ chỉnh sửa chi tiết
- Step 7: Nháy đúp chuốt vào Scheduler để mở cửa sổ mô phỏng các tín hiệu IO
- \* Step 8: Bấm Start để chạy toàn bộ process, Stop để khởi tạo lại toàn bộ các công cụ
- Step 9: Điều chỉnh thuộc tính IO giao tiếp PLC của các Scheduler



\* Note:

- ✓ Sau khi kết nối đầu vào và đầu ra của các công cụ, cần chạy lại Step 8 để cập nhật kết quả của toàn bộ process.
- ✓ Module giao tiếp PLC được lấy từ các cài đặt khi thiết kết UI Design. Tham khảo tại Section 2.1

#### 3.2 Common Tool – Acquisition, Image File, Region, Template Match, Light Control



#### TOOL PROPERTIES a gi 98 ^ 3. Tool Params **RunParams** CameraType HikVision CameraNumb -1 CameraSerial ToGray True GravTvpe BGR2Grav HorizontalFlip False Vertical Flip False Rotation R0 Exposure Time 0 4. Input **CalibrationFile** Name Name

#### Acquisition: Cài đặt camera và thu thập hình ảnh

• Input:

- (1) Trigger: Tín hiệu bắt đầu chụp từ scheduler
- Output:
  - (1) OutImage: Hình ảnh nhận được
  - (2) OutTrigger: Trigger để tắt đèn nếu cần thiết

#### Properties:

- (1) CameraType: Loại camera cần được kết nối
- (2) CameraNumber: Số thứ tự của camera
- (3) CameraSerial: Serial number cua camera
- (4) ToGray: Lựa chọn biến đổi thành ảnh đen trắng
- (5) GrayType: Lựa chọn loại ảnh đen trắng
- (6) HorizontalFlip: Đối xứng hình qua trục X
- (7) VerticalFlip: Đối xứng hình qua trục Y
- (8) Rotation: Xoay hình đi 90, 180, 279 độ
- (9) ExposureTime: Thời gian phơi sáng của hình
- (10)CalibrationFile: Load file căn chỉnh px-mm

#### \* Note:

✓ Sau khi được load vào, file căn chỉnh sẽ được đính cùng hình ảnh OutImage và có thể được sử dụng ở bất kỳ Tool nào.

✓ Hiện tại chương trình hỗ trợ kết nối với 4 loại camera thông dụng trên thị trường: HikVision, Basler, IMI, Dahua. Tham khảo <u>section 6.4</u> nếu HW camera không nằm trong danh sách trên.

Image Load 0

Trigger OutImage

Waiting

• Input:

- (1) Trigger: Tín hiệu bắt đầu load hình ảnh từ scheduler
- Output:

(1) OutImage: Hình ảnh được load lên từ PC

▶ Image Load: Lấy hình ảnh được lưu trong máy tính ◄

#### TOOL PROPERTIES

0003

•	A⊈↓ 🖻		
	Х	340 /	
	Y	279	
~	3. Tool Params		
~	RunParams		
	CurrentImage	0	
	InputImageLis	(Collection)	
	ImageFileNarr		
	ToGray	True	
	GrayType	BGR2Gray	
	HorizontalFlip	False	
	VerticalFlip	False	
	Rotation	R0	
~	4. Input		
	Calibration File		
		· · · · · · · · · · · · · · · · · · ·	1
Na Nar	<b>me</b> ne		

#### • Properties:

- (1) CurrentimageIndex: Thứ tự ảnh trong danh sách
- (2) InputImageList: Danh sách các ảnh sẽ được load
- (3) ImageFileName: Tên của ảnh hiện tại được load
- (4) ToGray: Lựa chọn biến đổi thành ảnh đen trắng
- (5) GrayType: Lựa chọn loại ảnh đen trắng
- (6) HorizontalFlip: Đối xứng hình qua trục X
- (7) VerticalFlip: Đối xứng hình qua trục Y
- (8) Rotation: Xoay hình đi 90, 180, 279 độ
- (9) CalibrationFile: Load file căn chỉnh px-mm

#### \* Note:

✓ Sau khi được load vào, file căn chỉnh sẽ được đính cùng hình ảnh OutImage và có thể được sử dụng ở bất kỳ Tool nào.

viac	chine Vision Pla	tform	AIO MATRIX	3.2 Common To
	Image Sa	ve0 ► 1	Image Save: Lưu hình ảnh ra 1 folder được lưa chor	<u>n trước</u> ◀
	Image Sa	ive 🔊		
	💿 Inimage		• IIIpul. (1) Inimage: Hinh sã được lược ra file	
	0003	Waiting	• Output:	
			· <u>oupu.</u>	
то		s		
•=				
•	Ž 🕴 📄		Properties:	
~	1. Tool Info		(1) FileName: Tên file sẽ được lưu	ı ra
	Name	Image Save 0	(2) FilePath: Đường dẫn đến folde	er lưu file
	ToolGroup	Common	(3) ByPass: Lưa chon bỏ qua việc	lưu file
	ToolName	Image Save	(4) AddDateTime: Thêm ngày thá	na vào tên file
	GUID	e4377023-7032-41	19d-a8ec (1) riddbato rinior rhon rigdy that	
~	2. Tool Display	200		
	X	300		
	ř а.т. на	300		
Ľ	3. Tool Params	5		
~	RunParams			
	FileName			
	FilePath	<b>F</b> 1		
	ByPass	False		
	/ ddbdto fillit			
Fil	eName			
Fil	Region T Region T Region 1 InImage Ou	Tool 0 Tool utImage Waiting	Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy các • <u>Input:</u> (1) InImage: Ảnh đầu vào • <u>Output:</u> (1) OutImago: Ảnh đầu ra	<u>c vùng bên ngoài</u> ◀
Fil	Region T Region T Region 1 InImage Ou InImage Ou	Tool 0 Tool @ utImage Waiting S	Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy các • <u>Input:</u> (1) InImage: Ảnh đầu vào • <u>Output:</u> (1) OutImage: Ảnh đầu ra	<u>c vùng bên ngoài</u> ◀
	Region T Region T Region 1 InImage Ou DL PROPERTIE	Tool 0 Fool ② utImage ③ Waiting S	Region Tool: Lưa chon vùng xử lý ảnh, điền đầy cáu • Input: (1) InImage: Ảnh đầu vào • <u>Output:</u> (1) OutImage: Ảnh đầu ra	<u>c vùng bên ngoài</u> ◀
Fil	Region T Region T Region T InImage Ou DL PROPERTIE	Tool 0 Tool © utImage ● Waiting S	Region Tool: Lựa chọn vùng xử lý ảnh, điền đầy cáu • Input: (1) InImage: Ảnh đầu vào • Output: (1) OutImage: Ảnh đầu ra • Properties: (1) RoiType: Thứ tự ảnh trong dan	<u>c_vùng bên ngoài</u> .◀ h sách
Fil	Region T Region T Region T InImage Ou InImage Ou DL PROPERTIE GUID	Tool 0 Tool © utImage ● Waiting S ec01231b-ed30-464	Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy các         • Input:         (1) InImage: Ảnh đầu vào         • Output:         (1) OutImage: Ảnh đầu ra         • Properties:         (1) RoiType: Thứ tự ảnh trong dan         (2) RegionOfInterest: Kích thước vào	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi
	Region T Region T Region 1 InImage Ou InImage Ou DL PROPERTIE A GUID 2. Tool Display	Tool 0     ▶       Tool @       utImage ●       Waiting       S       ec01231b-ed30-464	Region Tool: Lựa chọn vùng xử lý ảnh, điền đầy cáu         • Input:         (1) InImage: Ảnh đầu vào         • Output:         (1) OutImage: Ảnh đầu ra         • Properties:         (1) RoiType: Thứ tự ảnh trong dan         (2) RegionOfInterest: Kích thước v         (3) FillOption: Điền đầy bên ngoài	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi
	Region T Region T Region 1 InImage Ou InImage Ou DL PROPERTIE A GUID 2. Tool Display X	cool 0     ▶ 1       Cool (Q)     ↓       ItImage (P)     ↓       Waiting     ↓       S     ↓       ec01231b-ed30-464       286	Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy cáu         • Input:         (1) InImage: Ảnh đầu vào         • Output:         (1) OutImage: Ảnh đầu ra         • Properties:         (1) RoiType: Thứ tự ảnh trong dan         (2) RegionOfInterest: Kích thước v         (3) FillOption: Điền đầy bên ngoài         (4) FillColor: Màu của vùng sẽ được	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi pc điền đầy
	Region T Region T Region 1 InImage Ou InImage Ou DL PROPERTIE A GUID 2. Tool Display X Y	cool 0       ▶ 1         fool       Image         utImage       Image         Waiting       S         ec01231b-ed30-464         286         299	Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy cáu         • Input:         (1) InImage: Ảnh đầu vào         • Output:         (1) OutImage: Ảnh đầu ra         • Properties:         (1) RoiType: Thứ tự ảnh trong dan         (2) RegionOfInterest: Kích thước v         (3) FillOption: Điền đầy bên ngoài         (4) FillColor: Màu của vùng sẽ đượ	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi yừng Roi ợc điền đầy
Fil TOX	Region T Region T Region T InImage Ou InImage Ou DL PROPERTIE 2 I IIII GUID 2. Tool Display X Y 3. Tool Params	ool 0       ▷         fool ②       ○         utImage ③       ○         Waiting       ○         S       ○         ec01231b-ed30-464       ○         286       ○         299       ○	Region Tool: Lưa chon vùng xử lý ảnh, điền đầy cáu         • Input:         (1) InImage: Ảnh đầu vào         • Output:         (1) OutImage: Ảnh đầu ra         • Properties:         (1) RoiType: Thứ tự ảnh trong dan         (2) RegionOfInterest: Kích thước v         (3) FillOption: Điền đầy bên ngoài         (4) FillColor: Màu của vùng sẽ đượ         * Note:	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi pc điền đầy
	Region T Region T Region T Region T InImage Ou DL PROPERTIE DL PROPERTIE CUID 2. Tool Display X Y 3. Tool Params RunParams	iool 0       Image         iutImage       Image         Waiting       Image         s       Image         ec01231b-ed30-464         286       299	Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy cáu         • Input:         (1) InImage: Ảnh đầu vào         • Output:         (1) OutImage: Ảnh đầu ra         • Properties:         (1) RoiType: Thứ tự ảnh trong dan         (2) RegionOfInterest: Kích thước v         (3) FillOption: Điền đầy bên ngoài         (4) FillColor: Màu của vùng sẽ đượ         × Note:         ✓ Có 4 loại vùng Roi: Rectangle - hình chữ r	<u>c_vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi pc điền đầy
	Region T Region T Region 1 InImage Ou InImage Ou DL PROPERTIE A CUID 2. Tool Display X Y 3. Tool Params RunParams RoiType	Tool 0       Image         Tool 0       Image <td>Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy cáu         • Input:         (1) InImage: Ảnh đầu vào         • Output:         (1) OutImage: Ảnh đầu ra         • Properties:         (1) RoiType: Thứ tự ảnh trong dan         (2) RegionOfInterest: Kích thước v         (3) FillOption: Điền đầy bên ngoài         (4) FillColor: Màu của vùng sẽ đượ         * Note:         ✓ Có 4 loại vùng Roi: Rectangle - hình chữ r         chữ nhật xoay, Ellipse – hình Elip;.Polyline – hình đa gu</td> <td><u>c_vùng_bên ngoài</u> h sách vùng Roi vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình iác</td>	Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy cáu         • Input:         (1) InImage: Ảnh đầu vào         • Output:         (1) OutImage: Ảnh đầu ra         • Properties:         (1) RoiType: Thứ tự ảnh trong dan         (2) RegionOfInterest: Kích thước v         (3) FillOption: Điền đầy bên ngoài         (4) FillColor: Màu của vùng sẽ đượ         * Note:         ✓ Có 4 loại vùng Roi: Rectangle - hình chữ r         chữ nhật xoay, Ellipse – hình Elip;.Polyline – hình đa gu	<u>c_vùng_bên ngoài</u> h sách vùng Roi vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình iác
	Region T         Region 1         Region 1         InImage 0	cool 0       Image       <	<ul> <li>Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy cáu</li> <li>Input:         <ul> <li>(1) InImage: Ảnh đầu vào</li> <li>Output:                 <ul></ul></li></ul></li></ul>	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình iác
	Region T Region T Region 1 Region 1 InImage Ou InImage	iool 0       Image       <	<ul> <li>Region Tool: Lưa chon vùng xử lý ảnh, điền đầy cáu</li> <li>Input:         <ul> <li>(1) InImage: Ảnh đầu vào</li> <li>Output:                 <ul></ul></li></ul></li></ul>	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình iác
	Region T Region T Region T Region T InImage Ou DI PROPERTIE 2 DL PROPERTIE 2 DL PROPERTIE 2 DL PROPERTIE 2 DL PROPERTIE 2 DL PROPERTIE 3 COUD 2. Tool Display X Y 3. Tool Params RoiType RegionOfInter X Y	iool 0       Image       <	<ul> <li>Region Tool: Lưa chon vùng xử lý ảnh, điền đầy cáu</li> <li>Input:         <ul> <li>(1) InImage: Ảnh đầu vào</li> <li>Output:                 <ul></ul></li></ul></li></ul>	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình iác
	Region T Region T Region T Region T InImage Ou DL PROPERTIE DL PROPERTIE DL PROPERTIE COUD 2. Tool Display X Y 3. Tool Params RoiType RegionOfInter X Y Width	iool 0       Image       <	<ul> <li>Region Tool: Lưa chọn vùng xử lý ảnh, điền đầy cáu</li> <li>Input:         <ul> <li>(1) InImage: Ảnh đầu vào</li> <li>Output:                 <ul></ul></li></ul></li></ul>	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình iác
	Region T Region T Region T Region T InImage Ou DL PROPERTIE 2 DL PROPERTIE CUID 2. Tool Display X Y 3. Tool Params RoiType RegionOfInter X Y Width Height	Tool 0       Image       <	<ul> <li>Region Tool: Lựa chọn vùng xử lý ảnh, điền đầy cáu</li> <li>Input: <ul> <li>(1) InImage: Ảnh đầu vào</li> </ul> </li> <li>Output: <ul> <li>(1) OutImage: Ảnh đầu ra</li> </ul> </li> <li>Properties: <ul> <li>(1) RoiType: Thứ tự ảnh trong dan</li> <li>(2) RegionOfInterest: Kích thước v</li> <li>(3) FillOption: Điền đầy bên ngoài</li> <li>(4) FillColor: Màu của vùng sẽ đượ</li> <li>× Note: <ul> <li>Có 4 loại vùng Roi: Rectangle - hình chữ r chữ nhật xoay, Ellipse – hình Elip; Polyline – hình đa gu</li> </ul> </li> </ul></li></ul>	<u>c_vùng_bên ngoài</u> h sách vùng Roi vùng Roi pc điền đầy nhật; Rotated Rectangle – hình lác
	Region T Region T Region 1 Region 1 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 3 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 3 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 2 InImage Ou DI PROPERTIE 3 InImage Ou DI PROPERTIE 3 InImage Ou DI PROPERTIE 3 InImage Ou DI PROPERTIE 3 InImage Ou DI DI PROPERTIE 3 InImage Ou DI DI PROPERTIE 3 InImage Ou DI DI DI PROPERTIE 3 InImage Ou DI DI DI PROPERTIE 3 InImage Ou DI DI DI PROPERTIE 3 INIMA IN	Tool 0       Image       <	<ul> <li>Region Tool: Lựa chọn vùng xử lý ảnh, điền đầy cáu</li> <li>Input: <ul> <li>(1) InImage: Ảnh đầu vào</li> </ul> </li> <li>Output: <ul> <li>(1) OutImage: Ảnh đầu ra</li> </ul> </li> <li>Properties: <ul> <li>(1) RoiType: Thứ tự ảnh trong dan</li> <li>(2) RegionOfInterest: Kích thước v</li> <li>(3) FillOption: Điền đầy bên ngoài</li> <li>(4) FillColor: Màu của vùng sẽ đượ</li> <li>× Note: <ul> <li>Có 4 loại vùng Roi: Rectangle - hình chữ r chữ nhật xoay, Ellipse – hình Elip: Polyline – hình đa gu</li> </ul> </li> </ul></li></ul>	<u>c_vùng bên ngoài</u> .◀ h sách vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình iác
	Region T         Region T         Region T         InImage Out         InImage Out <td< td=""><td>cool 0       Image       &lt;</td><td><ul> <li>Region Tool: Lưa chon vùng xử lý ảnh, điền đầy cáu</li> <li>Input: <ul> <li>(1) InImage: Ảnh đầu vào</li> </ul> </li> <li>Output: <ul> <li>(1) OutImage: Ảnh đầu ra</li> </ul> </li> <li>Properties: <ul> <li>(1) RoiType: Thứ tự ảnh trong dan</li> <li>(2) RegionOfInterest: Kích thước v</li> <li>(3) FillOption: Điền đầy bên ngoài</li> <li>(4) FillColor: Màu của vùng sẽ đượ</li> <li>Note: <ul> <li>Có 4 loại vùng Roi: Rectangle - hình chữ r chữ nhật xoay, Ellipse – hình Elip;.Polyline – hình đa gu</li> </ul> </li> </ul></li></ul></td><td><u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình lác</td></td<>	cool 0       Image       <	<ul> <li>Region Tool: Lưa chon vùng xử lý ảnh, điền đầy cáu</li> <li>Input: <ul> <li>(1) InImage: Ảnh đầu vào</li> </ul> </li> <li>Output: <ul> <li>(1) OutImage: Ảnh đầu ra</li> </ul> </li> <li>Properties: <ul> <li>(1) RoiType: Thứ tự ảnh trong dan</li> <li>(2) RegionOfInterest: Kích thước v</li> <li>(3) FillOption: Điền đầy bên ngoài</li> <li>(4) FillColor: Màu của vùng sẽ đượ</li> <li>Note: <ul> <li>Có 4 loại vùng Roi: Rectangle - hình chữ r chữ nhật xoay, Ellipse – hình Elip;.Polyline – hình đa gu</li> </ul> </li> </ul></li></ul>	<u>c vùng bên ngoài</u> ◀ h sách vùng Roi vùng Roi ợc điền đầy nhật; Rotated Rectangle – hình lác

Name Name

	Template Match 0		<u>npla</u>	te Match: Tìm kiếm vi trí của 1 hình được đăng ký trước (Single) ◄		
<ul> <li>Template Match</li> <li>InImage OutImage</li> <li>Point</li> <li>Out</li> <li>Out</li> </ul>			• <u>Inr</u> • <u>Ou</u>	<u>ut:</u> (1) InImage: Ảnh đầu vào t <u>put:</u> (1) OutImage: Ảnh đầu ra		
TO	OL PROPERTIE	S		(2) Point: Vị trì hình giông template da tìm được		
	A Z↓ □ 3. Tool Params		^	Properties:     (1) Algorithm: Thuật toán tìm kiếm		
~	RunParams			(2) Mode: Kiểu tìm kiếm nếu sử dụng multi-mark		
	Algorithm	MaxAccuracy		(3) ScoreLimit: Giới hạn Score của quá trình tìm kiêm		
	Mode	None		(4) SearchRegion: Vùng tìm kiêm		
	ScoreLimit	0.7	- 10	(5) TrainRegion: Vùng đăng ký Pattern		
	> SearchRegior	Rectangle		(6) Origin: Tọa độ gốc của Pattern		
	> TrainRegion	Rectangle		(7) PatternData: Các pattern đã được đăng ký		
	> Origin	Coordinate		(8) AngleNeg: Giới hạn dưới của góc tìm kiếm		
	PatternDatas	(Collection)		(9) AnglePos: Giới hạn trên của góc tìm kiếm		
	AngleNeg	-20		(10) First Ston: Bước nhất của gia tìm kiếm ban đầu		
	AnglePos	20				
	FirstStep	5		* Note:		
<b> </b> ~	4. Output			✓ Các Output khác: Score, SelectedPattern,MatchingResult		
	SelectedPatternIn	0		✓ Multi-Mark: Sử dụng trong trường hợp điểm mark bị mờ, làm cho giá trị		
<b>_</b>	✓ MatchingResult Matching Result ✓		~	Score nhỏ hơn ScoreLimit		
So	core					



то	TOOL PROPERTIES						
	<b>A</b> ↓ □						
~	3. Tool Params						
~	RunParams						
	Algorithm	MaxAccuracy					
	Mode	None					
	ScoreLimit	0.7					
	> SearchRegior	Rectangle					
	> TrainRegion	Rectangle					
	> Origin	Coordinate					
	PattemDatas	(Collection)					
	AngleNeg	-20					
	AnglePos	20					
	First Step	5					
	MaxPose	10					
	FillColor	0					
~	✓ 4. Output						
Ca	Capacity						

Template Match Ex: Tìm kiếm vị trí của 1 hình được đăng ký trước (Multi)

#### • Input:

(1) InImage: Tín hiệu bắt đầu load hình ảnh từ scheduler

• Output:

- (1) OutImage: Hình ảnh được load lên từ PC
- (2) MatchResults: Danh sách kết quả tìm kiếm

#### • Properties:

- (1) Algorithm: Thuật toán tìm kiếm
- (2) Mode: Kiểu tìm kiếm nếu sử dụng multi-mark
- (3) ScoreLimit: Giới hạn Score của quá trình tìm kiếm
- (4) SearchRegion: Vùng tìm kiếm
- (5) TrainRegion: Vùng đăng ký Pattern
- (6) Origin: Tọa độ gốc của Pattern
- (7) PatternData: Các pattern đã được đăng ký
- (8) AngleNeg: Giới hạn dưới của góc tìm kiếm
- (9) AnglePos: Giới hạn trên của góc tìm kiếm
- (10)FirstStep: Bước nhẩy của góc tìm kiếm ban đầu
- (11)MaxPose: Giới hạn số lượng tìm kiếm
- (12)FillColor: Mầu điền đầy để tìm kiếm tiếp tục

#### \* Note:

✓ Đây là công cụ mở rộng của Template Match. Việc tìm kiếm sẽ được thực hiện giống như Template Match sau đó sẽ tiến hành xóa bỏ hình đã tìm kiêm được bằng cách điển đầy vùng tìm được với Fill Color

	0	Light Cont Light Cont Trigger	rol 0 rol P Waiting	► <u>Light C</u> • <u>In</u> • <u>O</u>	Control: Điều khiển bật tắt đèn thông qua cổng serial port ◀ p <u>put:</u> (1) InImage: Ảnh đầu vào utput: •
то 	OL PROPERTIES A ↓ B 3. Tool Params RunParams COLLINE		^	<ul> <li>Properties:         <ul> <li>(1) LightPort: Lựa chọn cổng kết nối với Controller</li> <li>(2) LightType: Loại controller của đèn</li> <li>(3) LighDevice: Các thông số cài đặt Interface</li> <li>(4) LightStatus: Trang thái sẽ được cài đặt</li> </ul> </li> </ul>	
	~	LightPort LightType Controller PortName MaxChan MaxVolur BaudRate Parity Databits Stopbits	ALP_8CH ALP_8CH COM1 8 255 B9600 Even 8 None		<ul> <li>(5) Channel: Kênh điều khiển tương ứng của đèn</li> <li>(6) Volume: Mức độ sáng của đèn</li> <li>* Note:         <ul> <li>Volume = 0 tương ứng với việc điều khiển tắt đèn</li> <li>Thêm bớt các loại đèn có thể tìm thấy trong project:CVAiO.Bplus.</li> </ul> </li> <li>LightingDevice</li> </ul>
		LightStatus Channel	OFF CH1	~	
3.3	me	Calibratio Find Chess	n & Fixtu board 0 board @	ring – Fine ► <u>Find Ct</u> • In	d Chessboard, Find Calib Matrix, Fiture hessboard: <i>Tìm kiếm vị trí các góc của ô bàn cờ</i> ◀ put:

- (1) InImage: Ảnh đầu vào
- Output:

٨

- (1) OutImage: Ånh đầu ra
- (2) VisionPoints: Tọa độ các góc tính theo px
- (3) RobotPoints: Tọa độ các góc tính theo mm

#### • Properties:

- (1) ChessLx: Độ rộng ô bàn cờ theo trục x
- (2) ChessLy: Độ rộng ô bàn cờ theo trục y
- (3) FlipX: Lấy đối xứng vị trí theo phương x
- (4) FlipY: Lấy đối xứng vị trí theo phương y
- (5) OriginPosition: Gốc tọa độ của hệ truc oxy
- (6) WinSize: Kích thước vùng tìm kiếm
- (7) ZeroZone: Kích thước vùng Dead Region

\* Note:

✓ Đây là công cụ hỗ trợ việc căn chỉnh manual. Chi tiết về cách sử dụng tham khảo <u>Section 4.2</u>

<u>Thuật toán trong OpenCV</u>



28	Р	а	a	е

Name Name

Find Calib Matri	x 0
Find Calib Matri	x 🖻
🔵 InImage	
VisionPoints	
RobotPoints	

COOS Waiting

- Find Calib Matrix: Tính toán ma trận căn trình dựa trên dữ liệu đầu vào
  - Input:
- (1) InImage: Ảnh đầu vào
- (2) VisionPoints: Dữ liệu trong hệ tọa độ px
- (3) RobotPoints: Dữ liệu trong hệ tọa độ mm
- Output:

TOOL PROPERTIES					
•	â↓ 📼				
	GUID	92e3614b-7fea-4c6b-9	٨		
~	2. Tool Display				
	X	292			
	Y	280			
~	3. Tool Params				
~	RunParams				
	TransShiftX	0			
	TransShiftY	0			
	RotShiftX	0			
	RotShiftY	0			
~	4. Calibration M	latrix			
	Overwrite	True			
	Folder	C:\CVAiO.Bplus\Ca			
	File	CalibMatrix			
			۷		
Na	me				
Na	me				

- Properties:
  - (1) TransShiftX: Offset giá trị căn chỉnh oxy theo trục X
  - (2) TransShiftY: Offset giá trị căn chỉnh oxy theo trục Y
  - (3) RotShiftX: Offset tâm xoay tìm được theo trục X
  - (4) RotShiftY: Offset tâm xoay tìm được theo trục Y
  - (5) Overwrite: Lưu đè nếu có file căn chỉnh cùng tên
  - (6) Folder: Vị trí lưu file căn chỉnh
  - (7) File: Tên file căn chỉnh

#### \* Note:

Auto

- ✓ Công cụ này được sử dụng trong cả 2 trường hợp căn chỉnh manual và
- ✓ Chi tiết về cách sử dụng có thể tham khảo tại Chapter 4

- Fixture 0 Fixture © Inlmage OutImage () InPoint Waiting
- Fixture: Định nghĩa 1 hệ tọa độ mới và gắn vào hình ảnh đầu ra

#### • Input:

- (1) InImage: Ảnh đầu vào
- (2) InPoint: Tọa độ của hệ trục tọa độ mới

#### • Output:

(1) OutImage: Ảnh đầu ra có đính kèm hệ tọa độ Fixture

#### TOOL PROPERTIES



#### • Properties:

#### Note:

✓ Đây thường là công cụ nối tiếp sau Template Match

✓ Hình ảnh sau khi đính kèm hệ tọa độ Fixute thường được sử dụng với công cụ <u>FindLine</u>, <u>FindCircle</u> do các tool caliper sẽ chạy theo hệ tọa độ Fixture giúp giảm độ rộng của các vị trí caliper.

#### 3.4 Deprecated – QR Decode

	QR Dec QR Dec Inimage O D	ode 0 code © utImage () ataCode () Waiting	► <u>QR De</u> • <u>In</u> • <u>O</u>	<u>code: <i>Giå</i> nput:</u> (1) <u>putput:</u> (1) 1 (2)	i <u>mã QR Code</u> ◀ InImage: Ảnh đầu vào OutImage: Ảnh đầu ra DataCode: Kết quả giải mã QR Code
	2↓ ©	IES		• Prope	erties:
* * *	1. Tool Info Name ToolGroup ToolName GUID 2. Tool Displ X Y 3. Tool Para	QR Decode Deprecated QR Decode dab 8fb 7a-5 ay 262 78 ms	<b>de 0</b> ∄ ⊧ i92a-4392-b11c	∻ Note:	
✓ Na Na	RunParams 4. Output DataCode				

3.5 Geometry Creation – Create Circle, Create Line, Create Rectangle



TO	ol propertie	S	
	<b>A</b> ↓		
	Y	218	^
~	3. Tool Params		
	RunParams		
$\sim$	4. Input		
	Point1		
	Point2		
	Point3		
$\sim$	5. Output		
	Circle		
	Center		
	Radius	0	
	LineColor	Blue	
	LineWidth	0	
	LineStyle	Solid	
			۷
Na Nar	<b>me</b> ne		

- Create Circle: Tao đường tròn đi qua 3 điểm đầu vào
  - Input:
- (1) InImage: Ảnh đầu vào
- (2) Points1,2,3: Điểm để tạo đường tròn
- Output:

\*Note:

- (1) OutImage: Ånh đầu ra
- (2) Circle: Đường tròn đi qua 2 điểm đầu vào
- Properties:

Name Name

_
- 5
<b>1</b>
-

lac	hine Vision Pla	atform		CVAIO B+	3.5 Geometry Creati
	Crostal	ino ()			
	Create L	ine 🕡 🕨	Create C	r <b>cle:</b> Tạo đường tròn đi qua 3 điể	<u>m đầu vào</u> ◀
	🔵 InImage O	utlmage 🔵	e Inni		
	Point1	Line 🔘	• inpu	(1) Inimagai Ánh đầu vào	
	Point2			(1) Inimage. Ann dau vao (2) Pointol 2: Diâm đổ too đ	vàng thẳng
		14.15	•	(2) Points1,2: Diem de tạo di	rong thang
		Waiting	• Out	out:	
то	OL PROPERTI	ES		(1) OutImage: A	nh đâu ra thắng đi qua 2 điểm
•	AL E				
•=	2 T D	_		Properties:	
Ľ	3. Tool Param	S	<u>^</u>	(1) CreateLineType: Phu	rơng pháp tao đường thẳng
*	CreatellineT	Deinte			
~				> Note:	
*	4. Input Point1			✓ Other Input: Inline, InCircle	
	Point2			✓ Ngoài phương pháp tạo ra đư nhán kháo nhự LinoPorrolo LinoPorrondia	ờng tháng qua 2 điểm còn có các phường
	In Line			οπαρ κπας πηια Επιεγατιαίε, Επιεγετρεπαίς Γικορα ίκρα	una, circle rangent sử dụng các dấu vào
	InCircle			aong ang	
~	5 Output				
Ĵ	Line	Line			
	SPX	0 000			
	SPY	0.000			
	EPX	0.000			
	EPY	0.000			
	ThetaAngle	0.000	~		
	<ul> <li>Create Rec</li> <li>InImage Ou</li> <li>Line0OutRe</li> <li>Line1</li> </ul>	tangle 🕑 itlmage 🌘 ctangle 🌒	• <u>Inp</u>	I <u>t:</u> (1) InImage: Ảnh đầu vào (2) Line1,2,3,4: Các đường ti	hẳng để tạo hình tứ giác
	Line2		• Out	put:	
	Line3			(1) Outlmage: Ánh đầu ra	
	6000	Waiting		(2) Rectangle: Hình tứ giác ở	lầu ra
_				Proportion	
10		ES		• <u>Floperties.</u> (1) AngleLimit: Giới han	góc khi tạo hình tứ giác
•	Ż↓			(2) LengthLimit: Giới hạ	n dưới độ dài cạnh tứ giác
~	3. Tool Param	s	^		
~	RunParams		_	≻ Note:	
	AngleLimit	2	_	✓ Đay la cong cụ thường được s thông tin liên quan đến warning tại soction	sự dụng đe tạo diem warping. Tim hieu ther
	LengthLimit	15		ulong un lien quan den warping ta <u>section</u> ✓ I ý thuyết về warping Perspec	5.9 tive ở OpenCV
~	4. Output			Ly and for the <u>marging receptor</u>	
	OutRectangle	5.1			
	IsRectangle	False			
	IsAngle	False			
	IsDistance	False			
		Blue			
		U Solid			
~	Misc	3010			
-	Point()				
	Detect		~		
	Point				

#### AIO MATRIX

3

#### 3.6 Geometry Finding & Fitting – Find Line, Find Circle, Find Corner, Find Points





Find Point Line 0
🥥 Find Point Line 🔞
🌒 InImage 🔵
Points 🔵
COC & Waiting

#### TOOL PROPERTIES

	Az↓		2		
~	3. 1	Гоо	l Params		^
<b>~</b>	Run	Par	ams		
	$\mathbf{v}$	Line	eCaliper	LineCaliper	
			NumberC	20	
			SearchLe	100	
			Projectior	20	
			Polarity	Dark2Light	
			Priority	Peak	
			FilterHalf:	1	
			MinContri	20	
		$\sim$	CaliperLir	Line	
			SPX	0.000	
			SPY	0.000	
			EPX	0.000	
			EPY	0.000	4
Na Nar	me ne				

Find Point Line: Tìm kiếm dành sách các điểm thông qua CaliperLine

#### • Input:

- (1) InImage: Ảnh đầu vào
- Output:
  - (1) OutImage: Ånh đầu ra
  - (2) Points: Danh sách các điểm tìm được của mỗi Caliper
  - Properties:
    - (1) LineCaliper: Đường thẳng chứa các caliper
    - (2) NumberCaliper: Số lượng caliper sẽ tìm kiếm
    - (3) SearchLength: Độ dài của 1 caliper
    - (4) ProjectionLength: Độ rộng của 1 caliper
    - (5) Polarity: Loại phân cực khi tìm kiếm
    - (6) Priority: Mức độ ưu tiên khi tìm kiếm
    - (7) FilterHalfPixel: Kich thước của ½ pixel lọc
    - (8) MinContrast: Giới hạn tương phản khi tìm kiếm

> Note:

LineCaliper sẽ chạy theo hệ tọa độ của <u>Fixture</u>

 ✓ Find Line là công cụ mở rộng của Find Point Line sau khi thực hiện xấp sỉ các điểm tìm được bằng 1 đường thẳng

Find Point Circle (2) Find Point Circle (2) InImage OutImage (2) Points (2) Waiting	<ul> <li>Find Point Circle: 1 im kiem danh sach cac diem thong gua CaliperCircle</li> <li>Input:         <ul> <li>(1) InImage: Ånh đầu vào</li> <li>Output:</li></ul></li></ul>
TOOL PROPERTIES	<ul> <li>(2) Points: Danh sách các điểm tìm được của mỗi Caliper</li> <li>Properties:</li> </ul>
<ul> <li>S. Tool Params</li> <li>RunParams</li> <li>CircleCaliper</li> <li>NumberC</li> <li>SearchLe</li> <li>Projectior</li> <li>Projectior</li> <li>Projectior</li> <li>Projectior</li> <li>Projectior</li> <li>Projectior</li> <li>Projectior</li> <li>Projection</li> <li>Priority</li> <li>Peak</li> <li>FilterHalf:</li> <li>MinContri</li> <li>Direction</li> <li>Inner</li> <li>CaliperCir</li> <li>Ellipse</li> <li>Cent</li> <li>100.000</li> <li>Teiq</li> <li>300.000</li> </ul>	<ul> <li>(1) CircleCaliper: Đường tròn chứa các caliper</li> <li>(2) NumberCaliper: Số lượng caliper sẽ tìm kiếm</li> <li>(3) SearchLength: Độ dài của 1 caliper</li> <li>(4) ProjectionLength: Độ rộng của 1 caliper</li> <li>(5) Polarity: Loại phân cực khi tìm kiếm</li> <li>(6) Priority: Mức độ ưu tiên khi tìm kiếm</li> <li>(7) FilterHalfPixel: Kích thước của ½ pixel lọc</li> <li>(8) MinContrast: Giới hạn tương phản khi tìm kiếm</li> <li>&gt; Note:         <ul> <li>CircleCaliper sẽ chạy theo hệ tọa độ của Fixture</li> <li>Find Circle là công cụ mở rộng của Find Point Circle sau khi thực hiện xấp si các điểm tìm được bằng 1 đường tròn</li> </ul> </li> </ul>



#### COO3 Waiting

#### TOOL PROPERTIES

•	2↓ 📼	
	Y	357
~	3. Tool Params	
$\sim$	RunParams	
	<ul> <li>LineCaliper</li> </ul>	LineCaliperEx
	SearchLe	10
	Polarity	Dark2Light
	FilterHalf:	1
	MinContra	20
	✓ CaliperLir	Line
	SPX	0.000
	SPY	0.000
	EPX	0.000
	EPY	0.000
	Thet	0.000
Na	me	
Nar	ne	

#### ► Find Point 1D: Tìm kiếm các điểm nằm trên 1 đường thẳng ◄

#### • Input:

- (1) InImage: Ảnh đầu vào
- Output:

~

- (1) OutImage: Ånh đầu ra
- (2) Points: Danh sách các điểm tìm được dọc theo đường caliper

#### • Properties:

- (1) LineCaliper: Đường thẳng chứa các caliper
- (2) SearchLength: Độ dài của 1 caliper
- (3) Polarity: Loại phân cực khi tìm kiếm
- (4) FilterHalfPixel: Kích thước của ½ pixel
- (5) MinContrast: Giới hạn tương phản khi tìm kiếm

#### ≻ Note:

LineCaliper sẽ chạy theo hệ tọa độ của <u>Fixture</u>

#### 3.7 Geometry Measurement – Distance Points, Distance Line, Distance Circle

Distance2P0				
	Distan	ce2P	Ø	
🔵 In	lmage	Lengt	h 🔵	
Pe	oint1			
🔵 Po	pint2			
COCS Waiting				

Distance3P 0 Distance3P

🔵 InImage

0002

🔵 InLine

Point

Length (

Waiting

- ▶ Distance2P: Tính khoảng cách giữa 2 điểm ◀
  - Input:
- (1) InImage: Ảnh đầu vào
- (2) Point1,2: Điểm đầu vào
- Output:
  - (1) Length: Khoảng cách giữa 2 điểm

- Input:
- (1) InImage: Ảnh đầu vào
- (2) InLine: Đường thẳng đầu vào
- (3) Point: Điểm đầu vào
- Output:
  - (1) Length: Khoảng cách từ điểm đến đường thẳng caliper

Distance Points 0			
Distance Points			
🔵 InImage 🛛 Lengths 🔵			
Points1			
Points2			
COC Waiting			

- Distance Points: Tính toán khoảng cách giữa 2 danh sách các điểm
  - Input:
- (1) InImage: Ảnh đầu vào
- (2) Points1,2: Danh sách các điểm đầu vào
- Output:
  - (1) Lengths: Các khoảng cách tính toán được



#### TOOL PROPERTIES

~	2. Tool Display		^
	X	348	
	Y	279	
~	3. Tool Params		
×	RunParams		
	<ul> <li>LineCaliper</li> </ul>	LineCaliper	
	NumberO	20	
	SearchLe	100	
	Projectior	20	
	Polarity	Dark2Light	
	Priority	First	
	FilterHalf:	1	
	MinContra	20	
	> CaliperLir	Line	
			<b>×</b>
1.	Tool Info		

- ► Distance Line: Khoảng cách giữa các điểm Black↔White theo đường thẳng ◄ • Input:
  - (1) InImage: Ảnh đầu vào

• Output:

- (1) Lengths: Danh sách các khoảng cách đầu ra
- (2) Points1,2: Danh sách các điểm tìm được của Caliper
- Properties:
  - (1) CircleCaliper: Đường tròn chứa các caliper
  - (2) NumberCaliper: Số lượng caliper sẽ tìm kiếm
  - (3) SearchLength: Độ dài của 1 caliper
  - (4) ProjectionLength: Độ rộng của 1 caliper
  - (5) Polarity: Loại phân cực khi tìm kiếm
  - (6) Priority: Mức độ ưu tiên khi tìm kiếm
  - (7) FilterHalfPixel: Kích thước của ½ pixel lọc
  - (8) MinContrast: Giới hạn tương phản khi tìm kiếm

#### > Note:

✓ CircleCaliper sẽ chạy theo hệ tọa độ của <u>Fixture</u>

✓ Find Circle là công cụ mở rộng của Find Point Circle sau khi thực hiện xấp sỉ các điểm tìm được bằng 1 đường tròn

- Distance Circle 0 Distance Circle @ InImage Lengths Points1 Points2 Waiting
- ▶ Distance Circle: Khoảng cách giữa các điểm Black ↔ White theo đường tròn ◄
   Input:
  - (1) InImage: Ảnh đầu vào
  - Output:
    - (1) OutImage: Danh sách các khoảng cách đầu ra
    - (2) Points1,2: Danh sách các điểm tìm được của Caliper
- TOOL PROPERTIES ₿₽ Х 344 ~ 280 3. Tool Params RunParams CircleCaliper CircleCaliper NumberC 20 SearchLe 100 Projectior 4 Polarity Dark2Light Priority First FilterHalf! 1 MinContr. 20 Direction Inner > CaliperCir Ellipse 1. Tool Info
- Properties:
  - (1) CircleCaliper: Đường tròn chứa các caliper
  - (2) NumberCaliper: Số lượng caliper sẽ tìm kiếm
  - (3) SearchLength: Độ dài của 1 caliper
  - (4) ProjectionLength: Độ rộng của 1 caliper
  - (5) Polarity: Loại phân cực khi tìm kiếm
  - (6) FilterHalfPixel: Kích thước của 1/2 pixel
  - (7) MinContrast: Giới hạn tương phản khi tìm kiếm
  - (8) Direction: Hướng bắt đầu tìm kiếm

> Note:

✓ CircleCaliper sẽ chạy theo hệ tọa độ của <u>Fixture</u>

Distance Points 0	
Distance Points	► <u>Distance Line: <i>Khoảng cách giữa các điểm đầu vào</i> ◄</u>
🔵 InImage 🛛 Lengths 🔵	• Input:
Points1	(1) InImage: Ảnh đầu vào
Points2	(2) Points1,2: Danh sách các điểm đầu vào
DOOS Waiting	• Output:
	(1) Lengths: Danh sách các khoảng cách đầu ra

> Note:

✓ Points 1,2 thường là kết quả của công cụ tìm kiếm các điểm Find Point Line hoặc Find Point Circle

#### 3.8 Geometry Intersection – Circle x Circle, Line x Circle, Line x Line

Intersect Circle Circle 0 Circle X Circle 2 Inlmage Point1	<ul> <li>Intersect Circle Circle: Tìm điểm cắt nhau của 2 đường tròn</li> <li>Input:         <ul> <li>Input:</li> <li>Inlmage: Ảnh đầu vào</li> <li>Inimage: Ảnh đầu vào</li> </ul> </li> </ul>
CircleA Point2	(2) Circle A,B: Các đường trồn đầu vào • <u>Output:</u>
Circles Waiting	(1) Point1,2: Điểm cắt nhau của 2 đường tròn
Intersect Line Circle 0	▶ Intersect Line Circle: Tìm điểm cắt nhau của 2 đường tròn ◀
🔎 Line X Circle 🗊	• Input:
Inimage Point1	(1) InImage: Ảnh đầu vào
Line Point?	(2) Line: Đường thẳng đầu vào
Circle	(3) Circle: Đường tròn đầu vào
	Output:

Intersect Line Line 0		
🥥 Line X l	Line 🖻	
🔵 Inimage	Point 🔵	
🔵 LineA		
LineB		
0002	Waiting	

- Intersect Line Circle: Tìm điểm cắt nhau của 2 đường tròn
  - Input:
    - (1) InImage: Ảnh đầu vào
    - (2) LineA,B: Đường thẳng đầu vào

Output:

(1) Point: Điểm cắt nhau của 2 đường nếu có

#### 3.9 Image Processing – Add Weighted, Binarization, Blob, Histogram, Morphology, etc.



Add Weighted 0

b44894b5-70d1-411d-b77

Image Processing Add Weighted

291

291

0.500

0.500

TOOL PROPERTIES

1. Tool Info

Name

GUID

х

~

Name Name

ToolGroup

ToolName

2. Tool Display

3. Tool Params

RunParams

Alpha Beta

Gama

AutoSubtract False

8∎ **A**↓

#### • AddWeighted: Pha trộn 2 hình theo công thức $O = (\alpha)Img_1 + (\beta)Img_2 + \gamma \blacktriangleleft$

- Input:
  - (1) InImage1,2: Ảnh đầu vào

Output:

- (1) OutImage: Ảnh đầu ra sau khi pha trộn
- Properties:
  - (1) Alpha: Hệ số α
  - (2) Beta: Hệ số β
  - (3) Gama: Hệ số γ
  - (4) AutoSubtract: Tự động tính toán và thực hiện pha trộn

#### > Note:

✓ Lý thuyết về pha trộn 2 hình ảnh đầu vào trên <u>OpenCV</u>

✓ Có thể sử dụng công cụ này như 1 phương pháp để kiểm tra (Inspection) bằng cách lấy ảnh nhận được trừ đi ảnh mẫu

Binarization 0 Binarization 2 InImage OutBinary AreaSum AreaSum Waiting

#### TOOL PROPERTIES



- Binarization: Chuyển đổi ảnh từ Gray Scale thành ảnh đen trắng (Binary)
   Input:
  - (1) InImage1: Ảnh đầu vào
  - Output:
    - (1) OutBinary: Ảnh đầu ra đã nhị phân hóa
    - (2) AreaSum: Tổng điện tích có pixel khác 0
    - Properties:
      - (1) Search Region: Vùng sẽ thực hiện biến đổi nhị phân
    - (2) KernelSize: Kich thước Kernel dùng cho Adaptive threshold
      - (2) Dong
        - (3) RangeLow: Giới hạn dưới
        - (4) RangeHigh: Giới hạn trên
        - (5) Mode: Kiểu chuyển đổi Binary
        - (6) ThresholdType: Loại chuyển đổi Binary

(7) AdaptiveThresholdType: Loại chuyển đổi Binay khi sử dụng AdaptiveMode

(8) Constant: Constant subtracted from the mean or weighted mean



Histogram 0 Histogram (2) InImage OutImage (2) HistogramResults (2) Mean (2) Waiting

FilterCenterMa FilterWidth FilterHeight

✓ 4. Output
 3. Tool Params

TOOL PROPERTIES			
2 2↓ □			
~	3. Tool Params		^
	RoiCount	0	
~	Roi	Rectangle	
	×	0.000	
	Y	0.000	
	Width	200.000	
	Height	200.000	
~	RunParams		
	NumberOfRoi	1	
	✓ RoiList	System.Collections	
	Capacity	4	
	Count	1	
	RangeLow	0	
	RangeHigh	255	
~	✓ 4. Output		
2. Tool Display			

- Histogram: Chuyển đổi ảnh từ Gray Scale thành ảnh đen trắng (Binary)
   Input:
  - (1) InImage1: Ảnh đầu vào
  - Output:
    - (1) OutBinary: Ảnh đầu ra đã nhị phân hóa
    - (2) AreaSum: Tổng điện tích có pixel khác 0
    - Properties:
      - (1) SearchRegion: Vùng sẽ thực hiện biến đổi nhị phân
    - (2) KernelSize: Kích thước Kernel dùng cho Adaptive threshold
      - (3) RangeLow: Giới hạn dưới
      - (4) RangeHigh: Giới hạn trên
      - (5) Mode: Kiểu chuyển đổi Binary
      - (6) ThresholdType: Loại chuyển đổi Binary
    - (7) AdaptiveThresholdType: Loại chuyển đổi Binay khi sử dụng AdaptiveMode

(8) Constant: Constant subtracted from the mean or weighted mean



#### TOOL PROPERTIES

	<b>≵</b> ↓ 🖻		
~	✓ 1. Tool Info		
	Name	Convert 0	
	ToolGroup	Image Processing	
	ToolName	Convert	
	GUID	628136f8-bb72-4599-a462	
~	2. Tool Display		
	х	287	
	Y	293	
~	3. Tool Params		
~	RunParams		
	Contrast	1.000	
	Brightness	0.000	
Name			
Name			

Convert: Điều chỉnh brightness và contrast

Input:

(1) InImage: Ảnh đầu vào

Output:

(1) OutImage: Ảnh đầu ra sau khi điều chỉnh độ tương phản và độ sáng

- Properties:
  - (1) Constrast: Độ tương phản
  - (2) Brightness: Độ sáng

Morphology 0
Morphology
🔵 Inlmage 🛛 Outlmage 🔵
COC Waiting

TOOL PROPERTIES

8∎ **2**↓

~

Name

GUID

х

Name Name

ToolGroup

ToolName

2. Tool Display

Tool Params RunParams

Туре

Shape

SizeX SizeY

Iteration

Morphology: Biến đổi hình thái học, các phép toán liên quan tới cấu trúc	
hình học. Giãn nở (dialation) phép co (erosion), phép mở	
(opening), phép đóng (closing). <	

Input:

٨

v

Morphology 0

Image Processing

d8f23748-5832-4477-a

Morphology

287

299

Dilate

Rect 1

3

3

(1) Inimage1: Ảnh đầu vào

- Output:
  - (1) OutBinary: Ảnh đầu ra đã nhị phân hóa
  - (2) AreaSum: Tổng điện tích có pixel khác 0
- Properties:
  - (1) Type: Loại biến đổi hình thái học
  - (2) Shape: Hình dạng của cửa sổ lọc
  - (3) Iteration: Số vòng lặp biến đổi
  - (4) SizeX: Kích thước cửa sổ lọc theo trục X
  - (5) SizeY: Kích thước cửa số lọc theo trục Y

> Note:

✓ Lý thuyết về biển đổi hình thái học trên <u>OpenCV</u>

	Rotate 0
	Rotate 😥
ا (	nlmage Outlmage 🧿
🔘 P	Point
	003 Waiting

Š

Š ~

~

Name

►	Rotation:	Xoay h	ình ảnh	quanh	1 điểm	◀
	Input:					

- (1) Inimage: Ảnh đầu vào
- Output:
  - (1) OutImage: Ảnh đầu ra sau khi xoay
- TOOL PROPERTIES ₿₽₽ Image Processing ToolGroup ٨ ToolName Rotate GUID 66d4a361-d563-4e0a-{ 2. Tool Display 294 х 293 3. Tool Params **RunParams** Reverse False XOffset 0 0 YOffset 0 TOffset 4. Input Point Name
- Properties:
  - (1) Reverse: Đảo chiều góc xoay
  - (2) XOffset: Offset theo chiều X
  - (3) YOffset: Offset theo chiều Y
  - (4) TOffset: Offset theo goc Theta

Warpping 0
🦳 Warpping 😥
🔵 InImage 🔘
DstLines
SrcLines
COC Waiting

то	TOOL PROPERTIES		
	<u>₽</u> ↓ 🖻		
~	3. Tool Params	]	,
~	RunParams		
	WarppingTyp	Perspective	
~	4. Input		
~	DstPoints	CVAiO.Bplus.Core.l	
	Length	4	
	LongLength	4	
	Rank	1	
	SyncRoot	CVAiO.Bplus.Core.Inter	
	IsReadOnly	False	
	IsFixedSize	True	
	IsSynchronize	False	
~	SrcPoints	CVAiO.Bplus.Core.l	
	Length	4	
	LongLength	4	1
3.	Tool Params		

- ▶ <u>Warpping</u>: Biến đổi hình ảnh theo thuật toán warping perspective. ◄ • Input:
  - (1) InImage: Ảnh đầu vào
  - (2) DstLines: Hình chữ nhật địch của quá trình warping
  - (3) SrcLines: Hình chữ nhật nguồn của quá trình warping
  - Output:
    - (1) OutBinary: Ảnh đầu ra sau khi đã warping
    - Properties:
      - (1) WarpingType: Loại thuật toán warping
    - > Note:
      - ✓ Lý thuyết về warping trên <u>OpenCV</u>

#### 3.10 Scheduler – Công cụ điều phối các nhánh xử lý

<ul> <li>Trigger1 • Output: Trigger2 • Trigger3 • Calc • Waiting</li> <li>• Output: (1) Trigger1,2,3,4: Tín hiệu điều khiển cho các cả (2) Calc: Tín hiệu điều khiển cho các cả Waiting</li> </ul>	các process vision ông cụ Algorithm
• Properties:	
(1) ComIO: Loại giao tiếp với PLC	để truyền nhận IO
(2) InTrigger1,2,3,4: Vị trí tín hiệu t	trigger
✓ 3. IO Setting ∧ (3) OutTriggerAck1,2,3,4: Vi trí tín	n hiệu trigger ack
ComIO ModbusRTU (4) InCal: Vị trí tín hiệu Calculation	
In Trigger 1 2 (5) In CalAck: Vị trí tín hiệu Calcula	ntion Ack
In Trigger 2 3	
In Trigger3 4 > Note:	
In Trigger 4 5 ✓ Loại ComIO được lấy từ cài đặt trong phần	n interface của <u>UI Design</u>
InCalc 6	
OutTrigger1Ack 71	
Out Ingger2Ack 72	
Out Ingger3Ack 73	
Out Ingger4Ack 74	
✓ 4. IO State Trianers <sup>1</sup> State	
Trigger State False	
Trigger2State False	
Trigger/State False	
Trigger1Ack State False	
Name Name	
• Scheduler Simulator: Để thuật tiện trong vi	ệc thiết kế và điểu chỉnh

 TRIGGER 1 (Disactive)
 các thông số của process vision, simulator cho phép người dùng mô phóng việc bật tắt các tín hiệu Trigger và Calc bằng cách lựa chọn Active/Disactive.

 TRIGGER 2 (Disactive)

TRIGGER 3 (Disactive)

TRIGGER 4 (Disactive)

CALC (Disactive)

#### 3.11 Vision Algorithm – Thuật toán xử lý ảnh

#### **Case Study 3: Circuit Inspection**

Kiểm tra thiếu linh kiện trên bản mạch sau khi lắp giáp bằng cách sử dụng công cụ Histogram



# **CHAPTER 4: CALIBRATION – CĂN CHỈNH**

4.1 Threory – Lý thuyết về căn chỉnh

4.2 Manual Calibration – Căn chỉnh bằng Chessboard

4.3 Auto Calibration - Căn chỉnh tự động

#### Các nội dung sẽ được đề cập trong chương này:

- Lý thuyết căn chỉnh
- Các bước để thực hiện căn chỉnh manual
- Các bước để thực hiện căn chỉnh tự động

#### 4.1 Threory – Lý thuyết về căn chỉnh

Trong hệ Machine Vision, căn chỉnh là quá trình tính toán ma trận chuyển đổi từ hệ tọa độ pixel của camera sensor sang hệ tọa tộ thực tế - world coordinate system. Ma trận chuyển đổi định nghĩa mối quan hệ giữa khoảng cách đo được theo pixel trong hệ tọa độ camera và khoảng cách theo miliment trong hệ tọa độ của đối tượng được chụp. Cảm biến của camera được chia thành các pixel riêng biệt. Hình ảnh được lưu lại trọng bộ nhớ của hệ thống vision là 1 mảng ghi nhận sự thay đổi được tạo ra tại các vị trí pixel sensor. Thông thường, hệ thống machine vision và xử lý ảnh sẽ sử dụng vị trí trên cùng bên phải như là điểm gốc của ảnh và từ đó tạo ra hệ trục tọa độ ảnh với trục X chạy theo hàng, trục Y chạy theo cột của các sensor.

Tất cả các đặc trưng trong ảnh như là điểm, đường, góc, cạnh sẽ được biểu diễn và tính tỉ lệ theo hệ tọa độ ảnh. Điều này có nghĩa là, nếu thiếu đi quá trình căn chỉnh, tất cả các vị trí sẽ là trong hệ tọa độ pixel tính tương đối với góc ngoài cùng bên phải và khoảng cách sẽ tính bằng pixels. Quá trình căn chỉnh sẽ cho phép chúng ta xác định 1 hệ tọa độ khác có ý nghĩa thực tế hơn. Điều đó dẫn tới kết quả của quá trình tính toán vision sẽ được biểu diễn trong hệ tọa độ mới này.

Quá trình căn chỉnh được hỗ trợ bởi CVAIO B+ sẽ cho phép chúng ta xác dịnh các đặc trưng quan trọng của hệ tọa độ căn chỉnh bao gồm các thành phần sau:



(1) Vi trí của điểm gốc – Origin của hệ tọa độ căn chỉnh: Đây là vị trí (0,0) của hệ tọa độ căn chỉnh trong mối liên hệ với camera sensor (0,0)

(2) Hướng các trục chỉnh của hệ tọa độ căn chỉnh.
 Đây là chiều dương của trục X, Y trong hệ tọa độ căn chỉnh

(3) Góc giữa hệ tọa độ căn chỉnh và hệ tọa độ camera sensor. Đây là tổng lượng góc xoay giũa hệ tọa độ căn chỉnh và hệ tọa độ sensor

(4) Tỉ lệ thu phóng của hệ tọa độ căn chỉnh. Đây là mỗi quan hệ giữa 1 đơn vị độ dài trong hệ tọa độ căn chỉnh và độ lớn 1 pixel trong hệ tọa độ camera sensor

Việc tính toán ma trận căn chỉnh được CVAIO B+ hỗ trợ thông qua việc sử dụng công cụ Find Calib Matrix với giá trị đầu vào là các giá trị tọa độ trong hệ tọa độ vision và hệ tọa độ thực tế như trình bày ở <u>Section 3.3</u>

#### Các cách sử dụng ma trận căn chỉnh trong CVAIO B+:

Cách 1: Gắn ma trận căn chỉnh vào hình ảnh ở công cụ Acquisition. Ma trận căn chỉnh sau khi được load lên sẽ đi cùng với hình ảnh trong suốt quá trình xử lý



Cách 2: Load vào trong công cụ Algorithm hoặc các công cụ ByUser

[Description("Calibration File"), Category("4. Input"), PropertyOrder(11)]
[Editor(typeof(CalibrationFileSelectionEditor), typeof(System.Drawing.Design.UITypeEditor))]
Orderences
public string CalibrationFile
{
 get => calibrationFile;
 set
 {
 CalibMatrix calibMatrix = new Serializer().Deserializing(value) as CalibMatrix;
 if (calibMatrix = null) return;
 this.calibMatrix = calibMatrix;
 calibrationFile = value;
 }
 [Browsable(false)]

public CalibMatrix CalibMatrix { get { if (calibMatrix == null) calibMatrix = new CalibMatrix(); return calibMatrix; } set => calibMatrix = value; }

#### Chuyển đổi giữa các hệ trục tọa độ:

♦ Vision Coordinate → World Coordinate:

Step 1: Thêm thư viện: Project  $\rightarrow$  References  $\rightarrow$ Add Reference  $\rightarrow$  Browse  $\rightarrow$  CVAiO.Bplus.Core Step2: Goi function: World Coordinate = **AIO.VtoR**(Ma trận căn chỉnh, Vision Coordinate)

✤ World Coordinate → Vision Coordinate:

Step 1: Thêm thư viện: Project  $\rightarrow$  References  $\rightarrow$ Add Reference  $\rightarrow$  Browse  $\rightarrow$  CVAiO.Bplus.Core Step 2: Gọi function: Vision Coordinate = **AIO.RtoV**(Ma trận căn chỉnh, World Coordinate)

#### 4.2 Manual Calibration – Căn chỉnh bằng Chessboard

Căn chỉnh manual được sử dụng chủ yếu để tìm hệ số chuyển đổi từ pixel sang mm. Camera sẽ chụp hình 1 tấm hình ô bàn cờ được in ra với một kích thước nhất định tính theo mm. Công cụ Find Chessboard sẽ giúp tìm ra vị trí các góc của ô bàn cờ, sau đó tính toán vị trí góc theo mm dựa theo kích thước thực tế nhập vào. Tổng hợp kết quả sẽ được đưa vào công cụ Find Calib Matrix để tính toán ma trận căn chỉnh.



Các bước để thực hiện căn chỉnh manual.

- Step 1: Chuẩn bị tấm ô bàn: Số lượng các ô bàn cờ theo cả 2 chiều phải là số lẻ. Có thể download hoặc tự tạo các ô bàn có theo hướng dẫn ở <u>OpenCV</u> hoặc <u>CVAiO</u>
- Step 2: Thiết lập process căn chỉnh: Các công cụ được chọn và ghép lại giống như hình bên trên. Điều chỉnh lại thông số của các công cụ Find Chessboard Conner và Find Calib Matrix

- Step 3: Bấm nút run để chạy process căn chỉnh và lưu lại file căn chỉnh.
- Step 4: Kiểm tra kết quả sau căn chỉnh: Mở công cụ Find Calib Matrix và bấm Run để hiển thị kết quả căn chỉnh. Các thông tin cần kiểm tra bao gồm giá trị sái số nhỏ nhất *min(mm)*, giá trí sai số lớn nhất *max(mm)*, độ rộng vùng 3 sigma *m\*std* của từng trục X,Y. Nếu các giá trị trên nhỏ tức là kết quả căn chỉnh có độ chính xác cao.

#### 4.3 Auto Calibration – Căn chỉnh tự động

Căn chỉnh tự động được sử dụng để đồng bộ hóa hệ tọa độ vision và hệ tọa độ của robot hoặc hệ servo XYT. Ma trận căn chỉnh sẽ giúp tìm được vị trí của vật thể trong hệ tọa độ robot sau khi xác định được trong hệ tọa độ vision. Quá trình căn chỉnh được chia làm 2 phần riêng biệt: Xác định chiều dương của hệ trục OX, OY và xác định vị trí tâm xoay O. Để xác định chiều của hệ trục tọa độ OXY, chương trình vision sẽ gửi lần lượt thông tin 9 vị trí (*RobotPoints*) tính theo mm trong hệ tọa tộ oxy cho robot sau đó chụp hình để xác định lại 9 vị trí trong hệ tọa độ vision (*VisionPoints*) tính theo pixel. Để xác định vị trí của điểm gốc không O, chương trình vision sẽ gửi lần lượt thông tin 5 góc xoay tính theo độ trong hệ tọa độ oxy cho robot rồi sau đó chụp hình đề xác định 5 vị trí góc xoay trong hệ tọa độ vision tính theo pixel. Hình dưới đây mô tả lại toàn bộ quá trình căn chỉnh auto với robot 3 trục XYT:





Trong toàn bộ quá trình căn chỉnh, chương trình vision sẽ giữ vai trò chủ động và robot (PLC) sẽ giữ vai trò bị động. Chương trình vision sẽ tính toán vị trí mà robot cần phải dịch chuyển sau đó gửi dữ liệu vị trí trước khi gửi tín hiệu IO yêu cầu robot dịch chuyển. Phía dưới mô tả chi tiết process căn chỉnh giữa chương trình vision và robot. Trong đó quá trình 3 steps

- (1) Gửi tọa độ vị trí robot cần phải dịch chuyển
- (2) Robot dịch chuyển tới vị trí yêu cầu;
- (3) Vision tiến hành xử lý tìm vị trí điểm mark;

sẽ được lặp lại nhiều lần cho đến khi tổng số vị trì cần cho quá trình căn chỉnh hoàn tất. Trong lúc quá trình căn chỉnh tự động được thực hiện, người dùng có thể huy bỏ bằng cách bấm nút Cancel góc dưới bên trái. Việc giao tiếp giữa vision và robot phải tuân thủ chặt chẽ theo đúng process đưa ra ở hình bên dưới để đảm bảo quá trình căn chỉnh được chính xác nhất.



Các bước để thực hiện căn chỉnh tự động:

INTERFACE STATUS		
IO IF Data IF		
	Acquisition 0 Template Match 0 Find Calil	b Matrix 0
had Ottal	🦲 Aca 💦 🙉 🦲 Template Match 🙉 🦲 Find Cali	b Matrix 😰
input Odiput	2 mage  Inima 2 timage  Inimage	8
READY     READY     READY	DOOR Waiting VisionPo	
START ACK	RobotPoir     OroShift     OroShift	nts
MOVE ACK 6 MOVE CMD	Calibration Scheduler 0	Tool Name Calibration Scheduler 0 - Calibration
DONE CMD     DONE ACK	Calibra 4 heduler @	
	<u>OraShift</u>	
	TOOL PROPERTIES	
TOOL PROPERTIES		× 3. Tool Params
	✓ 1. IF Communication ∧	✓ RunParams
	IOType ModbusRTU	TransAreaX 40
✓ 1. Tool Info	> ComIO ModbusRTU	TransAreaY 40
Name Calibration Schedul	Data Type Modbus TCP	TransCountX 3
ToolName Calibration Scheduler	ComData Modbus ICP	3. Tool Params
GUID bfca151a.9d58.4c6b.a	Infredoy 50	Translation
× 2 Tool Display		
× 98	InDone 53	
Y 142	InEndAck 54	
✓ 3. Tool Params	OutReady 100	U U U
✓ RunParams	OutStart 101	
TransAreaX 40	OutMove 102	
TransAreaY 40	OutDoneAck 103	
TransCountX 3	OutEnd 104	
TransCountY 3	DataAddr 200	
TransShiftX 0	V 2. How Chart	
TransShiftY 0	ComData	
RotAngleFror -10	IF Comunication type	
Name		Rotation
Name		
CANOCI		Status: 🛐 Time: 0.00ms
CANCEL	Status: Time: 0.000 3/28/202 33 PM New Process have been cr	Juitus V IIIIe: 0.00IIIs

- Step 1: Cài đặt giao tiếp với PLC: Chuột phải vào màn hình chính để hiện thị các thuộc tính giao tiếp với PLC ở bên trái màn hình. Lựa chọn IOType, DataType sẽ được sử dụng để truyền nhận IO và dữ liệu vị trí robot sẽ dịch chuyển.
- Step 2: Lựa chọn camera sử dụng trong quá trình căn chỉnh
- Step 3: Setup công cụ template matching để tìm bắt điểm mark căn chỉnh, chỉ sử dụng main mark và sử dụng Max Accuracy algorithm để đạt được độ chính xác cao nhất trong quá trình tìm kiếm.
- Step 4: Lựa chọn tổng số điểm dịch chuyển trong quá trình căn chỉnh. Thông thường sẽ để giá trị mặc định là 9 điểm cho căn chỉnh hướng X,Y và 5 điểm cho căn chỉnh góc T.
- Step 5: Bấm Start để thiết lập kết nối với PLC và bắt đầu quá trình căn chỉnh.
- Step 6: Kiểm tra trạng thái giao tiếp của Vision và PLC
- Step 7: Kiểm tra các bước hiện tại của quá trình căn chỉnh
- Step 8: Kiểm tra kết quả sau căn chỉnh: Mở công cụ Find Calib Matrix và bấm Run để hiển thị kết quả căn chỉnh. Các thông tin cần kiểm tra bao gồm giá trị sái số nhỏ nhất *min(mm)*, giá trí sai số lớn nhất *max(mm)*, độ rộng vùng 3 sigma *m\*std* của từng trục X,Y. Nếu các giá trị trên nhỏ tức là kết quả căn chỉnh có độ chính xác cao.

# CHAPTER 5: HISTORY & USER LOGIN – LỊCH SỬ & QUẢN LÝ TÀI KHOẢN ĐĂNG NHẬP

5.1 History – Phân tích lich sử

5.2 User Login – Quản lý tài khoản đăng nhập

#### Các nội dung sẽ được đề cập trong chương này:

- Cấu trúc của log file và cách phân tích
- Quản lý tài khoản đăng nhập của người dùng

## 5.1 History – Phân tích lịch sử

5.2 User Login – Quản lý tài khoản đăng nhập

# CHAPTER 6: TOOL BY USER – PHÁT TRIỀN CÔNG CỤ VÀ LUỒNG XỬ LÝ MỚI

- 6.1 Overview Giới thiệu chung
- 6.2 Tool By User Phát triển công cụ xử lý ảnh
- 6.3 Scheduler By User Phát triển luồng xử lý ảnh
- 6.4 Camera By User Phát triển tích hợp camera
- 6.5 Interface By User Phát triển module giao tiếp

#### Các nội dung sẽ được đề cập trong chương này:

- Giới thiệu về ý tưởng thiết kế mở cho người dùng
- Các bước để phát triển 1 công cụ xử lý ảnh mới
- Các bước để phát triển 1 luồng xử lý mới
- Các bước để tích hợp thêm camera mới
- Các bước để tích hợp thêm module interface mới

#### 6.1 Overview - Giới thiệu chung

Các bài toán trong hệ machine vision rất da dạng nên không thể thiết kế một platform như một lời giải chung được. Do đo CVAIO B+ được đưa đến cho người dùng dưới dạng môt solution của visual studio c# để người dùng có thể dễ dàng cấu trúc lại cho phù hợp với yêu cầu của từng bài toán. Trong đó có 2 projects sử



dụng để phát triển các công cụ xử lý ảnh, 2 projects sử dụng để phát triển công cụ algorithms, 1 project để phát triển module tích hợp camera, 1 project để phát triển module giao tiếp PLC, 1 project để phát triển module điều khiển ảnh sáng thông qua RS232, 1 project để phát triển luồng xử lý ảnh. Chi tiết về các projects này sẽ được trình bày chi tiết từ section 6.2 đến section 6.5. Ngoài ra còn có 2 projects liên quan đến việc tích hợp Cognex Vision Pro và 2 projects liên quan đến tích hợp halcon sẽ được trình bày trong chapter 7.

#### 6.2 Tool By User – Phát triển công cụ xử lý ảnh

**Yêu cầu:** Phát triển 1 công cụ xử lý ảnh để thực hiện việc phân đoạn hình ảnh sử dụng thuật toán Watershed như trình bày trong OpenCV ở <u>đường link</u> này.



Giải pháp: Phát triển thêm công cụ xử lý ảnh sử dụng project ToolByUser

- Step 1: Xác định đầu vào: Công cụ sẽ sử dụng 2 hình ảnh đầu vào 1 Hình ảnh Gray Scale lấy trực tiếp từ camera, 2 Hình ảnh Binary sau khi đã xử lý thông qua các công cụ Binarization và Morphology.
- Step 2: Code các giá trị đầu vào
   Step 3: Xác đinh đầu ra
- ✤ Step 3: Code các giá trị đầu ra .
- Step 4: Xác định các giá trị Parameter setting
- ✤ Step 5: Code các giá trị đầu ra

#### Đánh giá: Đánh giá kết quả thực hiện được



#### 6.3 Scheduler By User – Phát triển luồng xử lý ảnh

Yêu cầu: Đưa ra yêu cầu của bài toán

Giải pháp: Giái pháp là gì và các bước thực hiện

- \* Step 1: Chuẩn bị tấm căn chỉnh
- Step 2: Thiết lập process căn chỉnh.
- \* Step 3:
- Step 3: Chạy căn chỉnh .
- Step 4: Sau khi quá trình thu.

Đánh giá: Đánh giá kết quả thực hiện được

#### 6.4 Camera By User – Phát triển tích hợp camera

Yêu cầu: Phát triển module camera để có thể tích hợp camera basler Gige vào công cụ Aquisition

Giải pháp: Sử dụng khả năng tích hợp loại camera mới ở project CVAiO.Bplus.Camera.ByUser

Step 1: Cài đặt phần mềm Pylon 5 của nhà cung cấp camera basler tại <u>Basler Web</u> hoặc <u>CVAIO B+</u>. Lưu ý cài đặt sử dụng đường dẫn mặc định của nhà cung cấp và lựa chon <u>Developer Mode</u> để cài đặt toàn bộ các file thư viện cần thiết cho người phát triển.



Copyright © 2007-2018 Basler | All Rights Reserved | www.baslerweb.co

Step 2: Thêm thư viện được cung cấp bởi nhà cung cấp vào trong project bằng cách chọn: project → Reference →Add Reference→Browse→Basler.Pylon.dll

- C# CVAiO.Bplus.Camera.ByUser
  - Properties
    - References
      - Analyzers
      - Basler.Pylon
      - CVAiO.Bplus.Core
      - CVAiO.Bplus.OpenCV
      - Microsoft.CSharp
      - System
      - System.Core
      - System.Data
      - System.Data.DataSetExtensions
      - System.Net.Http
      - System.Xml
      - System.Xml.Linq
  - C# Camera\_ByUser.cs
- Step 3: Thêm class và lựa chọn kế thừa từ interface Camera: Camera\_ByUser : ICamDevice, Idisposable. Các lớp cần phải implement gồm có:

```
public interface ICamDevice : IDisposable
{
    int GetConnectedCamCount(); // Thu thập tổng số cam đang được kết nối đến
    string GetCameraSeiralNo(); // Lấy serial number của camera
    string GetCameraSeiralNo(int camNumber); // Lấy serial number của camera
    void CameraOpen(int camNumber); // Mở kế nối với camNumber
    Mat GrabImage(); // lấy hình và trả ra dưới định dạng Mat của OpenCV
    void CameraClose(); // Đóng kết nối với camera hiện tại
}
```

- Step 4: Các hàm sẽ được gọi trong công cụ Acquision gồm có CameraOpen(int CamNumber): Thiết lập kết nối với camera, GrapImage(): Thu thập hình ảnh từ camera và trả ra dưới định dạng Mat của OpenCV, CameraClose(): Đóng kết nối với camera và giải phóng tài nguyên đã sử dụng,
- Lưu ý: Người dùng nên tham khảo các <u>code mẫu</u> của nhà cung cấp để có thể hoàn thiện chương trình 1 cách chính xác nhất hoặc liên hệ với AIO MATRIX để được hỗ trợ.

**Đánh giá:** Kiểm tra kết quả bằng cách chuyển đổi camera trong công cụ Acqusition thành ByUser, thực hiện kết nối và thu thập hình ảnh từ camera

#### 6.5 Interface By User – Phát triển module giao tiếp

Yêu cầu: Phát triển module giao tiếp với PLC sử dụng chuẩn SLMP

#### Giải pháp:

- Step 1: Tìm hiểu về SLMP protocal
- Step 2:

Đánh giá: Đánh giá kết quả thực hiện được

# CHAPTER 7: COGNEX VISION PRO & HALCON MVTEC

#### 7.1 Cognex Vision Pro

#### 7.2 Halcon MVTec

#### Các nội dung sẽ được đề cập trong chương này:

- Tích hợp thêm công cụ xử lý ảnh của Cognex Vision Pro
- Tích hợp thêm công cụ xử lý ảnh của Halcon MVTec

#### 7.1 Cognex Vision Pro

#### 7.2 Halcon MVTech



#### Appendix 1

Appendix 2