

การออกแบบหุ่นยนต์แบบหีบและวางที่ใช้ระบบการมองเห็นของเครื่องจักร



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมการผลิต

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2558

**DESIGN OF PICK AND PLACE ROBOT WITH  
MACHINE VISION SYSTEM**



**A Thesis Submitted in Partial Fulfillment of the Requirements for the  
Degree of Master of Engineering in Manufacturing Engineering**

**Suranaree University of Technology**

**Academic Year 2015**

## การออกแบบหุ่นยนต์แบบหีบและวางที่ใช้ระบบการมองเห็นของเครื่องจักร

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นักวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

---

(อ. ดร.วรรณวนัช บุ่งสุด)

ประธานกรรมการ

---

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

---

(อ. ดร.โสทรฎา แข็งการ)

กรรมการ

---

(อ. ดร.พิจิตรา เอื้องไพโรจน์)

กรรมการ

---

(ศ. ดร.ชูกิจ ลิมปิจำนงค์)

รองอธิการบดีฝ่ายวิชาการและนวัตกรรม

---

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

พันศักดิ์ เนระแก : การออกแบบหุ่นยนต์แบบหยิบและวางที่ใช้ระบบการมองเห็นของ  
เครื่องจักร (DESIGN OF PICK AND PLACE ROBOT WITH MACHINE VISION SYSTEM)  
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์, 119 หน้า.

เนื่องจากการทำงานของหุ่นยนต์แบบหยิบและวางในปัจจุบันจะต้องมีการจัดวางตำแหน่ง  
ของส่วนประกอบต่าง ๆ ทุกส่วนให้ถูกต้องเป็นไปตามที่ได้กำหนดไว้ในตอนแรก เพราะ  
การที่ชิ้นส่วนใดมีตำแหน่งที่ผิดพลาดไปเพียงเล็กน้อยจะทำให้กระบวนการประกอบนั้นล้มเหลวได้  
ทำให้ในกระบวนการผลิตจริงจะต้องมีการหยุดระบบการผลิตเพื่อปรับตำแหน่งของชิ้นส่วนต่าง ๆ  
อย่างละเอียด ทั้งในการบำรุงรักษาปกติและการหยุดสายการผลิตเป็นเวลานาน เพื่อการเปลี่ยนรุ่น  
ผลิตภัณฑ์ที่จะผลิต วัตถุประสงค์งานวิจัยนี้จะเป็นการออกแบบและสร้างเครื่องจักรต้นแบบที่ใช้ใน  
การหยิบและวางในกระบวนการประกอบ โดยเพิ่มความสามารถในการทำงานของหุ่นยนต์ด้วย  
การใช้ระบบการมองเห็นของเครื่องจักร ประกอบการทำงานของหุ่นยนต์ วิธีการออกแบบ  
มีส่วนประกอบสำคัญ 2 ส่วน คือ 1) ระบบลำเลียงชิ้นงานอัตโนมัติ โดยใช้โปรแกรม KV STUDIO  
เขียนระบบการทำงานของชุดลำเลียงชิ้นงาน 2) การใช้โปรแกรม NI Vision Builder ในการกำหนด  
รูปแบบจุดศูนย์กลางของรูปแบบ ความเหมือน สีที่ใช้ในการค้นหา ขอบเขตในการค้นหา และส่วน  
ของโปรแกรม NI LabVIEW ร่วมกับโมดูล NI Vision ใช้ในการออกแบบกระบวนการทำงานของ  
ระบบการมองเห็นของเครื่องจักร เพื่อค้นหาพิกัดและแนวการวางตัวของชิ้นงาน และควบคุมการ  
เคลื่อนที่ของหุ่นยนต์ หุ่นยนต์ที่ใช้ในการศึกษานี้เป็นแบบ Scara Robot โดยการจะสั่งให้ Robot เคลื่อนที่  
ได้นั้น เราจะต้องส่งชุดคำสั่ง Protocol B ซึ่งเป็นชุดคำสั่งเฉพาะในการสั่งการ Robot เพื่อให้ระบบ  
การทำงานทั้งสองส่วนทำงานสอดคล้องประสาน และทำงานอย่างถูกต้องไปพร้อมกัน ผลที่คาดว่าจะ  
ได้รับระบบลำเลียงชิ้นงานสามารถลำเลียงชิ้นงานมาหยุดที่จุดจับยึด และระบบการมองเห็นด้วยภาพ  
ของเครื่องจักรที่ได้ทำการออกแบบสามารถตรวจจับตำแหน่งตามแนวแกน X, Y และแนวการวางตัว  
ตามแนวแกน B ได้อย่างถูกต้องตามรูปแบบของชิ้นงานที่ได้รับ และสามารถสั่งการให้หุ่นยนต์หยิบและ  
วางชิ้นส่วนในการประกอบได้อย่างถูกต้องและแม่นยำ

สาขาวิชา วิศวกรรมการผลิต

ปีการศึกษา 2558

ลายมือชื่อนักศึกษา \_\_\_\_\_

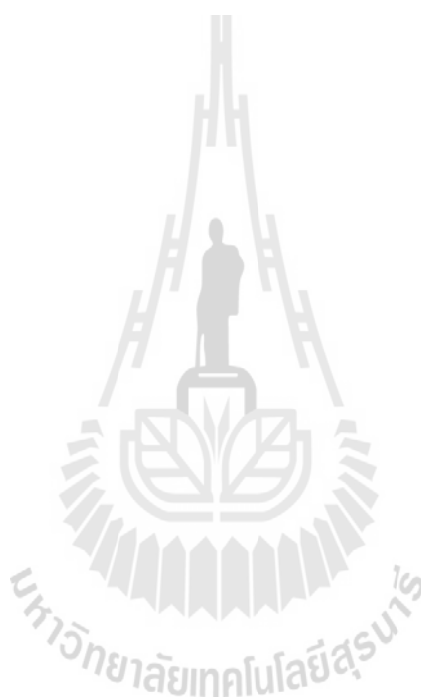
ลายมือชื่ออาจารย์ที่ปรึกษา \_\_\_\_\_

PHANSAK NERAKAE : DESIGN OF PICK AND PLACE ROBOT WITH  
MACHINE VISION SYSTEM. THESIS ADVISOR : ASSOC. PROF. FLT. LT.  
KONTORN CHAMNIPRASART, Ph.D., 119 PP.

## MACHINE VISION/PICK AND PLACE ROBOT/ASSEMBLY PROCESS

In the regular pick and place process of the production line, all parts and equipment must be aligned with in the right position. Since infinitesimal off-alignment can cause difficulty in assembly process. This leads the production line to shut down regularly due to line maintenance and product model change. The objective of this research is to design and build a prototype machine which is used to pick and place in the assembly process. The machine vision system is combined with the robotic systems to enhance the efficiency of pick and place process. The design phase has two integral parts: 1) automated conveyor systems. The KV STUDIO is used to write the conveyor system. 2) Vision system. The NI Vision Builder is used to defined patterns, the center of the model defined, pattern similarity, searching color and the searching scope. The application of NI LabVIEW with NI Vision module is used in the design process of the machine vision system to find the coordinates and orientation of the piece. These coordinates and orientation are used to control the movement of the robot. Scara Robot is used in this study commanded, we must send a set of instructions Protocol B, a set of specific commands to instruct the robot, to make the robot moves to the right coordination and work properly. As the results of the study, the conveyor system transports work pieces can be clamped at the break. The vision systems that have been designed can detect the position of the pattern along the X, Y, and its

orientation of the axis U correctly. And the instructed robot can pick and place parts in the assembly process correctly and accurately.



School of Manufacturing Engineering

Academic Year 2015

Student's Signature \_\_\_\_\_

Advisor's Signature \_\_\_\_\_

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ผู้วิจัยขอขอบพระคุณบุคคลและกลุ่มบุคคลต่าง ๆ ที่ให้คำปรึกษา แนะนำ และช่วยเหลือเป็นอย่างดีมาโดยตลอด ทั้งด้านวิชาการ และการดำเนินงานวิจัย ดังนี้

รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ คณบดีสำนักวิชาวิศวกรรมศาสตร์ และยังเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้ มอบโอกาสทางการศึกษาในระดับบัณฑิตศึกษา อีกทั้งยังให้ความรู้ด้านวิชาการ การดำเนินชีวิต อีกทั้งยังให้คำปรึกษาจนสามารถผ่านอุปสรรคต่าง ๆ ลุล่วงมาด้วยดี นอกจากนี้ยังชี้แนะ และมอบโอกาสที่ดี ๆ เสมอมาที่

อาจารย์ ดร. สมศักดิ์ ศิวคำรองพงส์ หัวหน้าสาขาวิชาวิศวกรรมการผลิต ผู้ซึ่งประสาทวิชาความรู้ ให้คำแนะนำ ทั้งในด้านวิชาการ การดำเนินชีวิต อีกทั้งยังให้กำลังใจจนสามารถผ่านอุปสรรคต่าง ๆ ลุล่วงมาได้ด้วยดี นอกจากนี้ยังชี้แนะแนวทางการทำงาน

ดร. พิจิตรา เอื่องไพโรจน์ ที่ได้ช่วยให้คำแนะนำทางด้านภาษาอังกฤษ ที่ใช้ในการเขียนบทความ และยังให้คำปรึกษาจนสามารถผ่านอุปสรรคต่าง ๆ

นายอริยวัตร วริคุณ, นายณัฐพล อิทชาติ, นายภูวรินทร์ สินมะเร็ง, นายชัยยุทธ คงเชื้อ ที่ได้ช่วยออกแบบระบบการทำงานของเครื่องจักร แนะนำการเขียนโปรแกรม และได้ช่วยสร้างเครื่องจักรจนสำเร็จ

บุคลากรประจำสาขาวิชาวิศวกรรมการผลิต และบุคลากรประจำสำนักวิชาวิศวกรรมศาสตร์ ตลอดจนบุคลากรประจำศูนย์เครื่องมือวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ได้อำนวยความสะดวกในการทำวิจัยทั้งทางด้านเอกสาร ข้อมูล และอุปกรณ์เครื่องมือต่าง ๆ ที่ใช้ในงานวิจัย

บริษัท ฮิตาชิ โกลบอล สตรอเรีย เทคโนโลยีส์ (ประเทศไทย) จำกัด ที่ให้ความอนุเคราะห์วัสดุในการทำวิจัย รวมทั้งข้อมูลที่จำเป็นต่องานวิจัย ตลอดจนคำแนะนำต่าง ๆ

สุดท้ายนี้ คุณงามความดีอันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแต่บิดา มารดา อันเป็นที่เคารพยิ่ง และคณาจารย์ผู้ประสาทวิชาความรู้ ตลอดจนทุก ๆ ท่านที่ให้กำลังใจช่วยเหลือจนกระทั่งวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงด้วยดี

พันศักดิ์ เนระแก

# สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ญ
สารบัญรูป.....	ฎ
<b>บทที่</b>	
<b>1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	3
1.3 ขอบเขตของการวิจัย.....	3
1.4 สถานที่ทำงานวิจัย.....	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
<b>2 ทัศนั้วรรณกรรมและทฤษฎีที่เกี่ยวข้อง.....</b>	<b>5</b>
2.1 กล่าวนำ.....	5
2.2 ความรู้ทั่วไปเกี่ยวกับหุ่นยนต์.....	5
2.2.1 ประวัติความเป็นมาของหุ่นยนต์.....	5
2.2.2 โครงสร้างของหุ่นยนต์.....	6
2.2.3 ส่วนประกอบของหุ่นยนต์.....	6
2.2.4 ชนิดของหุ่นยนต์.....	7
2.2.5 การหา DH Parameter.....	10
2.2.6 การหาเวกเตอร์การเคลื่อนที่ของหุ่นยนต์.....	11



## สารบัญ (ต่อ)

หน้า

2.3	การควบคุมด้วย PLC (Programmable Logic Controller, PLC).....	13
2.3.1	หลักการทํางานพื้นฐานของ PLC.....	15
2.3.2	การโปรแกรม PLC.....	18
2.4	ทฤษฎีเกี่ยวกับแสงและสี.....	20
2.4.1	คุณสมบัติของแสง.....	21
2.5	ความรู้พื้นฐานของภาพดิจิทัล.....	24
2.5.1	ภาพดิจิทัล (Digital Image).....	24
2.5.2	คุณสมบัติของภาพดิจิทัล (Properties of Digital Image).....	25
2.5.3	ชนิดของภาพดิจิทัล (Image Types).....	25
2.5.4	ชนิดของข้อมูลภาพดิจิทัล (Image File Type).....	26
2.5.5	ขอบภาพ (Image Borders).....	26
2.6	การหาขอบภาพแบบโซเบล (Sobel Edge Detector).....	26
2.6.1	Gradient Method.....	27
2.6.2	Laplacian Method.....	27
2.7	การค้นหาวัดถุในภาพโดยวิธี โมดิไฟยเจเนอรัลไลพีฮัฟทรานส์ฟอร์ม (Modified Generalized Hough Transform: MGHT).....	28
2.7.1	การเรียนรู้ (Learning).....	29
2.7.2	การค้นห (Detection).....	30
2.8	การค้นหภาพแบบแคนนี่ (Canny Edge Detection Algorithm).....	31
2.8.1	Smoothing.....	31
2.8.2	Gradient Calculation.....	32
2.8.3	Nonmaxima Suppression.....	32
2.9	ทฤษฎี LabVIEW.....	33
2.9.1	ส่วนประกอบต่าง ๆ ใน LabVIEW.....	33
2.9.1.1	Front Panel.....	34
2.9.1.2	Block Diagram.....	36

## สารบัญ (ต่อ)

หน้า

2.9.1.3 Icon และ Connector.....	38
2.9.2 DAQ, GPIB and Serial Communication.....	41
<b>3 วิธีการดำเนินงานวิจัย.....</b>	<b>44</b>
3.1 อุปกรณ์ที่สำคัญในการดำเนินการ.....	45
3.1.1 การออกแบบชุดลำเลียงชิ้นงานและตัวจับชิ้นงานเพื่อรอการตรวจสอบ.....	45
3.1.2 ระบบการประกอบชิ้นส่วน.....	45
3.1.3 ระบบการตรวจจับรูปแบบและตำแหน่งการวางตัวของชิ้นงาน.....	46
3.2 การจัดวางตำแหน่งของอุปกรณ์ต่างๆ.....	48
3.2.1 ทำการจัดวางระบบลำเลียงชิ้นงานของระบบ.....	48
3.2.2 การติดตั้งตำแหน่งของ Scara Robot.....	49
3.2.3 การติดตั้งหัวจับชิ้นงาน.....	49
3.2.4 การวางตำแหน่งกล้อง CCD Camera.....	50
3.2.5 การออกแบบรูปแบบของชิ้นงาน.....	50
3.2.6 ทำการติดตั้ง Store.....	51
3.2.7 ทำการติดตั้งตัวจับชิ้นงาน.....	51
3.2.8 ทำการติดตั้งชุด Control Scara Robot.....	52
3.2.9 ทำการติดตั้ง PLC KEYNCE KV-700.....	52
3.2.10 ทำการติดตั้ง Power Supply.....	53
3.2.11 ทำการติดตั้ง Main Regulates.....	53
3.2.12 ทำการติดตั้งชุด Compact Valve ของระบบนิวเมติก.....	54
3.2.13 ทำการติดตั้งไฟแจ้งสถานะ (Status Lamp).....	54
3.2.14 ทำการติดตั้งสวิตช์ควบคุมการทำงาน.....	55
3.3 ทำการเขียนโปรแกรม PLC ควบคุมการทำงานของ Conveyer.....	55
3.3.1 การกำหนด Address Input / Output.....	55
3.3.2 ระบบการทำงานของชุดลำเลียงชิ้นงาน.....	58
3.3.3 โปรแกรมที่ทำกรออกแบบการทำงานของระบบลำเลียงชิ้นงาน.....	59

## สารบัญ (ต่อ)

หน้า

3.4	ทำการเขียนโปรแกรม NI Vision Builder.....	65
3.4.1	การเลือกชนิดของกล้องในการตรวจสอบ.....	65
3.4.2	ทำการทดสอบกล้องและปรับความคมชัด.....	66
3.4.3	ทำการปรับสีในการค้นหา.....	67
3.4.4	ทำการกำหนดขอบเขตในการค้นหา.....	67
3.4.5	การกำหนดรูปแบบในการค้นหา.....	68
3.5	Image Calibration.....	68
3.5.1	Calibration Object.....	69
3.5.2	พิกัดจริงจากรูปแบบที่ได้ออกแบบ.....	69
3.5.3	ตำแหน่งจริงที่ได้จากการประมวลผลของกล้อง.....	71
3.5.4	การหาค่า Error ที่จุดต่างๆ.....	72
3.6	การเขียนโปรแกรม NI LabVIEW.....	73
3.6.1	การออกแบบระบบการทำงานของ NI LabVIEW.....	73
3.6.2	ลักษณะ Block Diagram สำหรับเครื่องตรวจสอบ.....	75
4	ผลการออกแบบเครื่องจักรและการทำงานของเครื่องจักร.....	95
4.1	ผลการเขียนโปรแกรมระบบการทำงานและการทดสอบการทำงานของชุดกลไกอัตโนมัติสำหรับส่งชิ้นงาน ชุดลำเลียงชิ้นงาน ชุดจับยึดชิ้นงาน.....	95
4.2	ผลการตรวจสอบระบบการตรวจสอบด้วยภาพ.....	96
4.2.1	ผลการทดสอบเวลาที่ใช้ในการตรวจสอบด้วยภาพ.....	96
4.2.2	ผลการทดสอบโปรแกรม NI LabVIEW.....	97
4.2.2.2	ผลการทดสอบเวลาที่ใช้ในการตรวจสอบด้วยภาพ.....	99
4.2.2.3	ผลจากการตรวจสอบและบอกพิกัดของชิ้นงาน.....	100
4.3	ผลการประกอบชิ้นส่วน.....	101
4.3.1	ตำแหน่งการหยิบชิ้นส่วนเพื่อนำมาประกอบ.....	101
4.3.2	ผลการทดสอบการทำงานของเครื่องจักรต้นแบบทั้งหมด.....	102
4.3.3	Image Calibration.....	103

## สารบัญ (ต่อ)

	หน้า
4.3.4 ค่าที่ได้จากการ Calibration.....	105
4.4 สรุปผลการทดลอง.....	106
<b>5 สรุปผลงานวิจัยและข้อเสนอแนะ.....</b>	<b>107</b>
5.1 สรุปผลงานวิจัย.....	107
5.2 ข้อเสนอแนะ .....	108
รายการอ้างอิง .....	110
ภาคผนวก	
ภาคผนวก ก. บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่.....	111
ประวัติผู้เขียน .....	119



## สารบัญตาราง

ตารางที่		หน้า
2.1	เปรียบเทียบศัพท์ที่ใช้ใน LabVIEW.....	39
3.1	Input Output PLC.....	56
3.2	ระยะพิกัดจริงที่ได้ทำการออกแบบ ( $X_m, Y_m$ ).....	70
3.3	ตำแหน่งตามความเป็นจริงที่ได้จากการถ่ายภาพหรือ ( $X_p, Y_p$ ).....	71
3.4	ตาราง Error ของจุดต่าง ๆ ตามแกน X, Y.....	72



## สารบัญรูป

รูปที่		หน้า
1.1	การทำงานของสายการผลิตที่ใช้หุ่นยนต์แบบหยิบแล้ววาง.....	1
1.2	ส่วนประกอบของเครื่องที่จะออกแบบและจัดสร้างในงานวิจัยนี้.....	3
2.1	ตัวอย่าง The Cartesian Robot.....	7
2.2	ตัวอย่าง The Cylindrical Robot.....	8
2.3	ตัวอย่าง The Spherical Robot.....	8
2.4	ตัวอย่าง The Articulated Robot หรือ The Revolute Robot.....	9
2.5	ตัวอย่าง The SCARA Robot.....	9
2.6	ตัวอย่าง The Parallel Robot.....	10
2.7	รายละเอียดของตำแหน่งตัวแปรต่าง ๆ ของ DH Parameters.....	10
2.8	ตัวอย่างแบบจำลองของหุ่นยนต์ 2 แขน.....	12
2.9	อุปกรณ์ที่เชื่อมต่อกับ PLC เพื่อใช้ในการควบคุม.....	14
2.10	ส่วนประกอบของ PLC.....	15
2.11	รอบการทำงานหนึ่งรอบของ PLC.....	17
2.12	Contact ใน PLC ตามข้อกำหนดของ Mitsubishi.....	19
2.13	สัญลักษณ์ของ Coil ที่นิยมใช้สามแบบตามบริษัทผู้ผลิต.....	19
2.14	สัญลักษณ์ของ Box ในรูปแบบต่าง ๆ.....	20
2.15	แสงขาวและสเปกตรัม.....	20
2.16	การสะท้อนแบบปกติ.....	21
2.17	การสะท้อนแบบกระจาย.....	21
2.18	กฎการสะท้อนของแสง.....	22
2.19	การหักเหของแสงกรณี $n_1$ .....	22
2.20	การหักเหของแสงกรณี $n_1 > n_2$ .....	23
2.21	ตัวอย่างการสะท้อนกลับหมดของแสง.....	23
2.22	ลักษณะของจุดเริ่มต้นของ Spatial coordinate.....	24

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.23	ลักษณะของการเกิดภาพทางดิจิทัล..... 25
2.24	ตัวกระทำของการหาขอบภาพด้วยวิธี Sobel..... 28
2.25	คุณลักษณะของภาพต้นแบบ..... 29
2.26	ข้อมูลภาพต้นแบบในตาราง R..... 29
2.27	ขั้นตอนของ Canny edge detection..... 31
2.28	โลโก้ของ LabVIEW..... 33
2.29	ลักษณะของ Front Diagram..... 34
2.30	Object ที่อยู่บน Front Panel ของ LabVIEW..... 35
2.31	Controls Palette ที่ใช้ในการออกแบบ Front Panel..... 35
2.32	Tools Palette ที่ใช้ในการออกแบบ Front Panel..... 36
2.33	ตัวอย่าง Block Diagram..... 36
2.34	ตัวอย่าง Block Diagram Node..... 37
2.35	เครื่องมือสำหรับ Dam-Data Acquisition..... 38
2.36	ลักษณะ Icon และ Connector..... 38
2.37	ข้อมูลประเภท Numeric..... 40
2.38	ข้อมูลประเภท Boolean..... 40
3.1	ลักษณะทั่วไปของเครื่องที่ได้ออกแบบ..... 44
3.2	รูปแบบระบบชุดลำเลียงชิ้นงาน..... 45
3.3	รูปแบบของ SCARA ROBOT..... 46
3.4	กล้องที่ใช้ในการตรวจสอบ..... 47
3.5	หัว Vacuum..... 47
3.6	การจัดวางระบบลำเลียงชิ้นงาน..... 48
3.7	รูปร่างของตัว Scara robot และตำแหน่งการวางของ Robot..... 49
3.8	การติดตั้งหัว Vacuum..... 49
3.9	รูปตำแหน่งการติดตั้ง Camera..... 50
3.10	รูปแบบของชิ้นงาน..... 50

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 ตำแหน่งการวาง Store.....	51
3.12 ตัวจับชิ้นงานรอกการถ่ายภาพ.....	51
3.13 การติดตั้งการควบคุม Robot.....	52
3.14 PLC.....	52
3.15 การติดตั้ง Power Supply.....	53
3.16 การติดตั้ง Main Regulator.....	53
3.17 การติดตั้งชุด Compact Valve.....	54
3.18 ทำการติดตั้งไฟแจ้งสถานะ.....	54
3.19 การติดตั้งสวิทช์ควบคุมการทำงาน.....	55
3.20 ระบบการทำงานของชุดลำเลียงชิ้นงาน.....	58
3.21 Diagram การทำงานระบบลำเลียงชิ้นงาน.....	59
3.22 Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ).....	60
3.23 Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ).....	61
3.24 Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ).....	62
3.25 Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ).....	63
3.26 Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ).....	64
3.27 ขั้นตอนการกำหนดรูปแบบของ Program NI Vision Builder.....	65
3.28 การเลือกชนิดของกล้องที่ใช้.....	66
3.29 การทดสอบการเชื่อมต่อและการปรับความคมชัดของภาพ.....	66
3.30 การปรับสีเพื่อใช้ในการค้นหา.....	67
3.31 การกำหนดขอบเขตในการค้นหา.....	68
3.32 การกำหนดรูปแบบในการค้นหา.....	68
3.33 รูปการ Calibration Object.....	69
3.34 การออกแบบระบบการทำงานของ Program NI LabVIEW.....	74
3.35 การกำหนด Serial port, กำหนด on, off Servo.....	75
3.36 ขั้นตอนการทำงานหลังกดปุ่ม Home.....	76



## สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.37	ขั้นตอนหลังรับสัญญาณ DAQ P 0.0.....	76
3.38	ขั้นตอนการตรวจสอบรูปแบบตาม Case ต่าง ๆ.....	77
3.39	ขั้นตอนการชดเชยค่าตำแหน่งหลังจาก Calibration.....	78
3.40	ขั้นตอนการปรับค่าตำแหน่งกึ่งกลางของรูปแบบ.....	79
3.41	ขั้นตอนการหาหมุนหามุมรูปสี่เหลี่ยมที่น้อยที่สุด.....	79
3.42	ขั้นตอนการหาหมุนหามุมรูปสามเหลี่ยมที่น้อยที่สุด.....	80
3.43	การบอกไฟสถานะการตรวจพบรูปแบบต่าง ๆ.....	80
3.44	ขั้นตอนการบอกจำนวนที่ได้มีการตรวจพบ.....	81
3.45	ขั้นตอนการบอกจำนวนที่ไม่มีการตรวจพบ.....	81
3.46	ขั้นตอนการบอกการตรวจสอบครบ 5 ครั้ง.....	82
3.47	การบอกตำแหน่งการหยิบชิ้นงานใน Store ตำแหน่งต่าง ๆ.....	82
3.48	การบอกตำแหน่งการหยิบชิ้นงานรูปสี่เหลี่ยมใน Store ตำแหน่งต่าง ๆ.....	83
3.49	การบอกตำแหน่งการหยิบชิ้นงานรูปสามเหลี่ยมใน Store ตำแหน่งต่าง ๆ.....	84
3.50	การบอกตำแหน่งการหยิบชิ้นงานรูปวงกลมใน Store ตำแหน่งต่าง ๆ.....	85
3.51	ขั้นตอนการบอกการจบกระบวนการประมวลผลด้วยภาพ.....	85
3.52	ขั้นตอนการส่งตำแหน่งให้ Robot เคลื่อนที่ก่อนจะจับชิ้นงาน.....	86
3.53	ขั้นตอนการแปลงค่า Robot return X.....	87
3.54	ขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ลงไปจับชิ้นงาน.....	87
3.55	ขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ขึ้นเมื่อจับชิ้นงานแล้ว.....	88
3.56	ขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ไปยังตำแหน่งก่อนประกอบชิ้นงาน.....	88
3.57	ขั้นตอนการส่งสัญญาณให้ Robot หมุนชิ้นงานก่อนประกอบ.....	89
3.50	การบอกตำแหน่งการหยิบชิ้นงานรูปวงกลมใน Store ตำแหน่งต่าง ๆ.....	85
3.51	ขั้นตอนการบอกการจบกระบวนการประมวลผลด้วยภาพ.....	85
3.52	ขั้นตอนการส่งตำแหน่งให้ Robot เคลื่อนที่ก่อนจะจับชิ้นงาน.....	86
3.53	ขั้นตอนการแปลงค่า Robot return X.....	87
3.54	ขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ลงไปจับชิ้นงาน.....	87

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.55	ขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ขึ้นเมื่อจับชิ้นงานแล้ว ..... 88
3.56	ขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ไปยังตำแหน่งก่อนประกอบชิ้นงาน ..... 88
3.57	ขั้นตอนการส่งสัญญาณให้ Robot หมุนชิ้นงานก่อนประกอบ ..... 89
3.58	ขั้นตอนการส่งสัญญาณให้ Robot วางชิ้นงานเพื่อประกอบ ..... 89
3.59	ขั้นตอนการตัดสัญญาณ Vacuum ..... 90
3.60	ขั้นตอนการส่งสัญญาณให้แกน Z เคลื่อนที่ขึ้นหลังจากประกอบเสร็จ ..... 90
3.61	ขั้นตอนการส่งสัญญาณให้แกน U เคลื่อนที่กลับตำแหน่งเดิม ..... 91
3.62	ขั้นตอนการส่งสัญญาณให้ Robot กลับไปยังตำแหน่ง Home ..... 91
3.63	ขั้นตอนการส่งสัญญาณให้ Reset จำนวนที่ตรวจสอบพบชิ้นงาน ..... 92
3.64	การบอก PLC ว่าจบการทำงานในส่วนของ Robot ..... 92
3.65	ขั้นตอนการส่งสัญญาณค่าเริ่มต้นและสัญญาณจบการทำงาน ..... 93

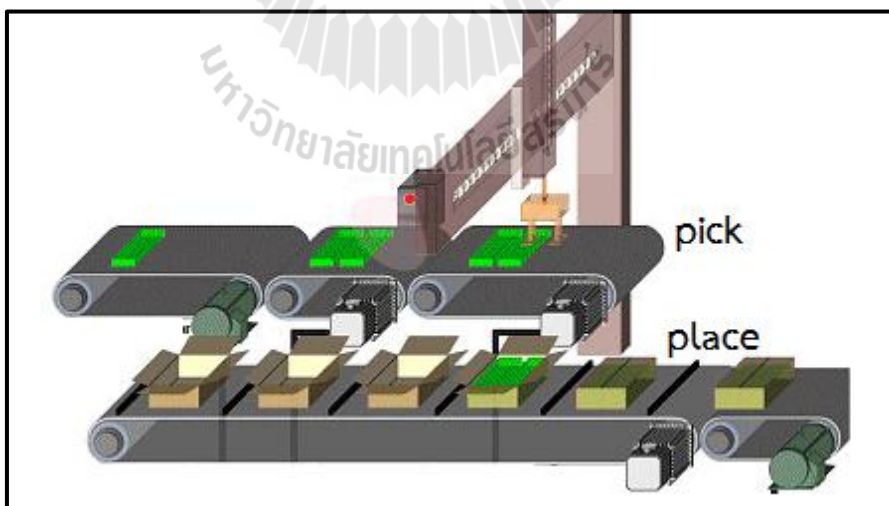


# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในกระบวนการผลิตในปัจจุบันมีการใช้หุ่นยนต์กันอย่างแพร่หลาย และหน้าที่หนึ่งที่นิยมใช้หุ่นยนต์ในการทำงาน คือ การประกอบชิ้นส่วนต่าง ๆ เข้าด้วยกัน โดยในขั้นตอนประกอบนั้น หุ่นยนต์มักจะทำหน้าที่หยิบชิ้นส่วนหนึ่งเข้าไปประกอบอีกชิ้นส่วนหนึ่ง ซึ่งเรามักเรียกหุ่นยนต์ที่ทำหน้าที่ หยิบและวาง (Pick and Place) ซึ่งในขั้นตอนนี้ส่วนที่สำคัญในกระบวนการผลิตที่ต้องจัดเตรียม คือ การกำหนดตำแหน่งและแนวการวางตัว (Position and Orientation) ของชิ้นส่วนทั้งสองต้องเป็นไปตามที่กำหนดไว้เท่านั้น เพราะหุ่นยนต์เองจะได้รับการสอนและโปรแกรม ให้หยิบชิ้นส่วนแรก ที่ตำแหน่งและแนววางตัวที่แน่นอน เพื่อจะนำไปประกอบเข้ากับชิ้นส่วนที่สอง ซึ่งจะมีการกำหนดตำแหน่งและแนววางตัวที่แน่นอนเช่นเดียวกัน ตามที่แสดงในรูปที่ 1.1 ซึ่งจะเห็นว่าชิ้นส่วนแรกเคลื่อนที่มาตามสายพานด้านบน หุ่นยนต์จะทำหน้าที่หยิบชิ้นส่วนแรก แล้วนำไปวางลงในกล่อง ซึ่งเหมือนกับว่ากล่องนั้นเป็นชิ้นส่วนที่สอง



รูปที่ 1.1 แสดงการทำงานของสายการผลิตที่ใช้หุ่นยนต์แบบหยิบแล้ววาง

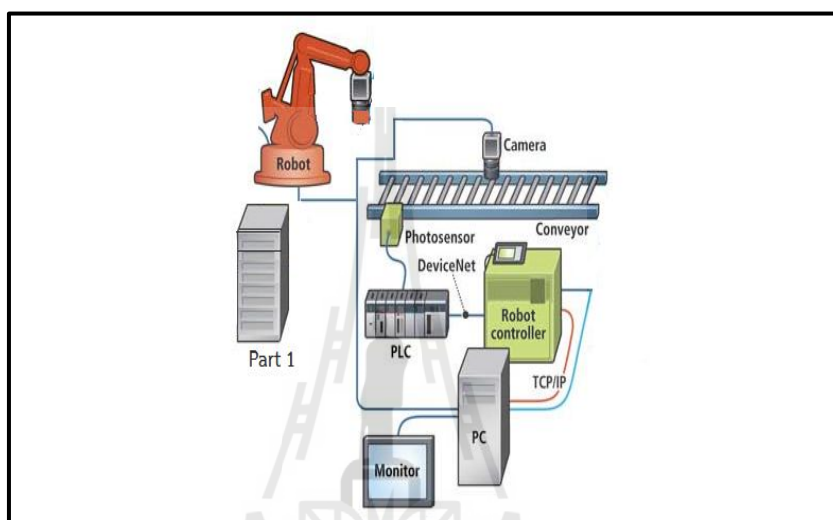
สำหรับสายการผลิตที่ใช้กระบวนการเช่นนี้ วิศวกรและช่างเทคนิคมีความจำเป็นที่จะต้องทำการปรับเทียบชิ้นส่วนทุกชิ้นให้มีตำแหน่งและแนววางตัวที่แน่นอน อีกทั้งหากชิ้นส่วนที่มีการเคลื่อนที่ไปตามสายการผลิต ตำแหน่งที่วัตถุเคลื่อนตัวเข้ามาจะต้องเป็นไปตามที่กำหนด ความคลาดเคลื่อนในตำแหน่งเพียงเล็กน้อย อาจหมายถึงผลผลิตที่ไม่มีคุณภาพตามที่กำหนดได้ หากกล่องมาในตำแหน่งที่เอียงทำมุมมากเกินไป อาจทำให้การวางของลงในกล่องผิดพลาดของอาจตกออกจากกล่อง หรือชิ้นด้านในเอียงทำให้ปิดฝากล่องไม่ได้ ดังนั้นในการออกแบบกระบวนการผลิตและสายการผลิตจะต้องรอบคอบและใช้ในการผลิตของที่มีจำนวนมาก เพื่อที่จะได้ไม่ต้องปรับแต่งบ่อยครั้ง เพราะเมื่อมีการเปลี่ยนผลิตภัณฑ์ที่ใช้ในกระบวนการผลิต จะพบว่าวิศวกรและช่างเทคนิค ต้องใช้เวลามากในการปรับแต่งสายการผลิตเพื่อที่จะรองรับการผลิตที่แตกต่างออกไปได้ เช่น การประกอบชิ้นส่วนอิเล็กทรอนิกส์เข้ากับแผงวงจร ซึ่งชิ้นส่วน เช่น แผงวงจรรวม (Integrate circuit) จะต้องวางในตำแหน่งและแนววางตัวที่แน่นอนบนแผงวงจรด้วยความแม่นยำสูง ดังนั้นหากมีการเปลี่ยนรุ่นของแผงวงจร สายการผลิตจะต้องถูกพักเพื่อปรับปรุงเป็นระยะเวลาาน ซึ่งการที่จะให้บรรลุวัตถุประสงค์ดังกล่าว จำเป็นต้องมีวิศวกรที่ทำหน้าที่วางแผนว่าผลิตภัณฑ์แต่ละรุ่นผลิตภัณฑ์ หรือโมเดลนั้นจะผลิตในปริมาณเท่าใด และใช้เครื่องจักรในสายการผลิตใดบ้าง เพื่อที่จะสามารถวางแผนการหยุดสายการผลิตเพื่อรองรับการเปลี่ยนไปของรุ่นผลิตภัณฑ์ได้อย่างมีประสิทธิภาพ ซึ่งการดำเนินการตามแนวทางนี้จะเหมาะกับโรงงานประกอบที่มีจำนวนรุ่นผลิตภัณฑ์ไม่มากนัก ส่งผลให้ไม่จำเป็นต้องมีการปรับเปลี่ยนสายการผลิตบ่อยมาก

อย่างไรก็ตามในยุคปัจจุบันการผลิตจำนวนมากในสินค้าประเภทเดียว ขนาดเดียว อาจไม่ใช่ทางเลือกที่เหมาะสมของภาคการผลิต เพราะในปัจจุบันผลิตภัณฑ์ที่แต่ละผู้ผลิตทำขึ้นอาจมีหลายโมเดล และรายละเอียดของแต่ละโมเดลย่อยก็มีความแตกต่างกันออกไป เช่นอาจมีการผลิตแผงวงจรของอุปกรณ์เฉพาะทางเพียง 1,000 แผงวงจร แต่มีอยู่ด้วยกัน 10 รุ่น ซึ่งทำให้สายการผลิตต้องหยุดทุกการผลิตแผงวงจรครบ 1,000 แผง ด้วยความต้องการความอ่อนตัวในกระบวนการผลิตที่มากขึ้นทำให้ผู้ผลิตมองหาวิธีการผลิตและประกอบผลิตภัณฑ์ที่สามารถปรับเปลี่ยนหุ่นยนต์แบบหยิบและวางนี้ ให้มีความสามารถมากขึ้น โดยการหยิบชิ้นส่วนนั้นอาจจะหยิบจากตำแหน่งเดิมซ้ำกันไปได้ แต่ตำแหน่งวางนั้นสามารถที่จะปรับเปลี่ยนตำแหน่งและแนววางตัวได้ตามที่ต้องการ

งานวิจัยนี้เป็นการเสนอแนวทางใหม่ในกระบวนการประกอบ โดยจะทำการจัดสร้างหุ่นยนต์ขึ้นมาเพื่อใช้ในกระบวนการประกอบชิ้นงาน โดยหุ่นยนต์นี้จะทำหน้าที่หยิบชิ้นส่วนที่หนึ่งแล้วนำไปวางประกอบในชิ้นส่วนที่สอง ข้อที่แตกต่างจากหุ่นยนต์และวิธีการที่ใช้กันในปัจจุบัน คือ หุ่นยนต์นี้จะติดตั้งระบบการมองเห็นของเครื่องจักร ทั้งนี้เพื่อช่วยกำหนดตำแหน่งว่า ตำแหน่งที่ควร

จะวางชิ้นส่วนแรกเข้ากับชิ้นส่วนที่สองนั้น มีตำแหน่งเท่าใด และมีแนววางตัวอย่างไรจึงจะสามารถประกอบชิ้นส่วนทั้งสองชิ้นเข้าด้วยกันได้อย่างสมบูรณ์

รูปที่ 1.2 แสดงลักษณะหุ่นยนต์ที่จะออกแบบและสร้างขึ้นในงานวิจัยนี้ คือจะมีระบบคอมพิวเตอร์ที่จะควบคุมการเคลื่อนที่ของหุ่นยนต์ และจะมีระบบการมองเห็นของเครื่องจักร เพื่อตรวจสอบตำแหน่งและแนววางตัวของส่วนประกอบที่ชิ้นส่วนจะถูกนำไปวางลงไป



รูปที่ 1.2 ส่วนประกอบของเครื่องที่จะออกแบบและจัดสร้างในงานวิจัยนี้

## 1.2 วัตถุประสงค์การวิจัย

สำหรับงานวิจัยนี้ มีวัตถุประสงค์ของงานวิจัยในด้านการออกแบบและพัฒนาอัลกอริทึมในการใช้การมองเห็นของเครื่องจักร เข้าช่วยการทำงานของหุ่นยนต์แบบหยิบและวางตามเงื่อนไขที่กำหนดให้ ซึ่งวัตถุประสงค์หลักของงานวิจัยมีดังต่อไปนี้

1.2.1 จัดสร้างเครื่องต้นแบบระดับห้องปฏิบัติการ ที่ใช้ควบคุมหุ่นยนต์แบบหยิบและวาง ให้สามารถจับวางชิ้นส่วนให้ได้ในตำแหน่งและแนววางตัวที่ต้องการ

1.2.2 ออกแบบระบบการทำงานร่วมกันระหว่างระบบกลไกและระบบตรวจสอบด้วยกล้องเพื่อให้ทั้งสองส่วนทำงานที่สอดคล้องกัน

## 1.3 ขอบเขตงานวิจัย

งานวิจัยนี้จะกำหนดขอบเขตการวิจัยเพื่อสามารถกำหนดเป้าหมายและวิธีการในการแก้ปัญหาได้อย่างเป็นระบบ ขอบเขตของงานวิจัยนี้ประกอบด้วย

1. หุ่นยนต์เป็นแบบ Scara Robot
2. การมองเห็นของเครื่องจักร ใช้โปรแกรมสำเร็จรูป Program Vision Builder, LabVIEW NI Vision ช่วยในการกำหนดรูปแบบ กำหนดขั้นตอนการทำงาน
3. ชั้นส่วนที่จะใช้ประกอบ ชั้นที่ 1 จะใช้แทนแผงวงจรรวม จะถูกป้อนเข้าราง Conveyer และจะหยุดอยู่ในตำแหน่งที่คงที่ ส่วนตำแหน่งที่จะถูกประกอบนั้นจะปรับเปลี่ยนไปตามแนวการวางตัว ซึ่งจะมีรูปแบบ 3 แบบ คือ สามเหลี่ยม สี่เหลี่ยม วงกลม เราจะเรียกว่า Main Part
4. ชั้นส่วนที่ 2 จะเป็นส่วนของ Store ซึ่งจะมีชั้นส่วนที่ใช้การประกอบเข้ากับแผงวงจร เราจะเรียกว่า Assemble Part
5. ใช้ระบบควบคุมด้วยคอมพิวเตอร์ ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ประสานกับการทำงานของโปรแกรมการมองเห็นของเครื่องจักร

#### 1.4 สถานที่ทำงานวิจัย

- 1.4.1 อาคารเครื่องมือวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีสุรนารี

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ได้แนวคิดใหม่สำหรับกระบวนการประกอบอัตโนมัติ
- 1.5.2 ช่วยให้อุตสาหกรรมวางแผนกระบวนการผลิตได้คล่องตัวขึ้น
- 1.5.3 ลดระยะเวลา down time ของสายการผลิตลง
- 1.5.4 ได้แนวทางการพัฒนาเครื่องจักรเพื่อยกระดับจากเครื่องต้นแบบระดับห้องปฏิบัติการ

## บทที่ 2

### ปรัชญาบรรณกรรมและทฤษฎีที่เกี่ยวข้อง

#### 2.1 กล่าวนำ

ในบทนี้จะเป็นการกล่าวถึงปรัชญาบรรณกรรมที่เกี่ยวข้องกับงานวิจัยฉบับนี้ และกล่าวถึงความจำเป็นมาของการพัฒนาเทคโนโลยีในด้านของความรู้ทั่วไปเกี่ยวกับหุ่นยนต์ และระบบการมองเห็นของเครื่องจักรอัตโนมัติเพื่อการผลิต ในภาคอุตสาหกรรมซึ่งมีการค้นคว้าและวิจัยในด้านนี้มาเป็นระยะเวลาอันยาวนาน และได้เริ่มเข้ามามีบทบาทแพร่หลายในภาคอุตสาหกรรม ประกอบกับอุปกรณ์ด้านการมองเห็นของเครื่องจักรจึงได้รับการวิจัยอย่างต่อเนื่องเสมอมา

#### 2.2 ความรู้ทั่วไปเกี่ยวกับหุ่นยนต์

##### 2.2.1 ประวัติความเป็นมาของหุ่นยนต์

หุ่นยนต์เป็นเทคโนโลยีที่มีใช้กันอย่างแพร่หลายในโรงงานอุตสาหกรรมการผลิต เพราะความสามารถที่ทำงานได้คล้ายกับมนุษย์และไม่รู้จักเหน็ดเหนื่อย ในปัจจุบันนั้นได้มีการพัฒนาหุ่นยนต์กันอย่างต่อเนื่องจนสามารถตอบสนองความต้องการของมนุษย์พร้อมกับแสดงปฏิกิริยาตามโปรแกรมที่ป้อนไว้เพื่อให้มนุษย์เข้าใจได้หุ่นยนต์หรือที่เรียกกันว่าโรบอท (Robot) [1,2] เริ่มต้นในปี ค.ศ. 1760 ได้มีการนำเอาเครื่องจักรไอน้ำเข้ามาทำงานแทนคนงานในโรงงานและผลที่ได้คือ เครื่องจักรทำงานได้ดีกว่ามนุษย์ จากนั้นเรื่องก็เงียบมาจนถึงปี ค.ศ. 1920 ได้มีนักเขียนละครชาวเชคโกสโลวาเกียชื่อ Karel Capek เขียนบทละครเรื่อง Rossum's Universal Robots : R.U.R โดยคำว่า Robot เป็นคำแผลงมาจากคำว่า Robota ในภาษาเชคที่แปลว่า Worker หรือคนงาน ต่อมาในปี ค.ศ. 1942 ได้มีนักประพันธ์ทางด้านวิทยาศาสตร์ Isaac Asimov ได้บัญญัติกฎของหุ่นยนต์ขึ้น 3 ข้อ คือ

- 1 หุ่นยนต์ห้ามทำร้ายมนุษย์ หรือปล่อยให้มนุษย์ได้รับอันตราย
- 2 หุ่นยนต์ต้องเชื่อฟังคำสั่งของมนุษย์เว้นแต่จะขัดแย้งกับกฎข้อหนึ่ง
- 3 หุ่นยนต์ต้องป้องกันตัวเองโดยไม่ขัดแย้งกับกฎข้อหนึ่งและข้อสอง

ถัดมาในปี ค.ศ. 1954 ได้มีการสร้างหุ่นยนต์ตัวแรกของโลกโดย George C. Devol วิศวกรชาวอเมริกาพร้อมกับจดสิทธิบัตร และจากนั้นก็ได้มีการพัฒนาเทคโนโลยีหุ่นยนต์เรื่อย ๆ จนถึงปัจจุบัน

### 2.2.2 โครงสร้างของหุ่นยนต์

โครงสร้างของหุ่นยนต์สามารถแบ่งออกเป็น 2 ส่วนสำคัญคือส่วนลำตัวและแขน (Body-and-Arm) และส่วนข้อมือ (Wrist) โดยมากแล้วส่วนลำตัวและแขนจะมีระดับขั้นความเสรี 3 ระดับและในส่วนข้อมือจะมีระดับขั้นของความเสรีอยู่ 2-3 ระดับที่ปลายของข้อมือจะเป็นวัตถุ ซึ่งมีความสัมพันธ์กับงานที่หุ่นยนต์ต้องทำ ตัวอย่างเช่น วัตถุอาจจะเป็นชิ้นงานที่ต้องการไหลดเข้าเครื่องจักร หรืออาจจะเป็นเครื่องมือที่หุ่นยนต์ต้องใช้ในกระบวนการผลิตบางอย่าง เป็นต้น ส่วนลำตัวและแขนของหุ่นยนต์จะใช้ในการจัดตำแหน่งที่ถูกต้องของวัตถุ และส่วนข้อมือของหุ่นยนต์จะใช้สำหรับจัดทิศทางการวางตัวที่เหมาะสมให้แก่วัตถุเพื่อที่จะวางตำแหน่งของวัตถุให้เหมาะสมส่วนลำตัวและแขนของหุ่นยนต์จะต้องสามารถเคลื่อนวัตถุได้ทิศทางทั้ง 3 ดังต่อไปนี้

1. การเคลื่อนที่ในแนวตั้ง (การเคลื่อนที่ในแนวแกน Z)
2. การเคลื่อนที่ในแนวรัศมี (เข้าหรือออกหรือการเคลื่อนที่ในแนวแกน Y)
3. การเคลื่อนที่จากซ้ายไปขวา (การเคลื่อนที่ในแนวแกน X)

การที่จะทำให้หุ่นยนต์สามารถเคลื่อนที่ในลักษณะข้างต้นได้มีหลายวิธี ทั้งนี้ขึ้นอยู่กับชนิดของข้อต่อที่ใช้ในการสร้างลำตัวและแขนของหุ่นยนต์ ซึ่งจะกล่าวถึงรายละเอียดต่อไป เพื่อที่จะทำให้เกิดการจัดทิศทางเหมาะสมของวัตถุ เราสามารถกำหนดระดับขั้นความเสรี 3 ระดับสำหรับข้อมือของหุ่นยนต์ตามตัวอย่างต่อไปนี้เป็นหนึ่งในรูปแบบในการประกอบข้อมือของหุ่นยนต์เพื่อทำให้เกิดเป็น 3 ลำดับขั้นความเสรี (3 Degree of Freedom)

1. Roll ลำดับขั้นความเสรีลักษณะนี้ทำได้โดยใช้ข้อต่อแบบ T เพื่อที่จะหมุนวัตถุรอบแกนของแขน
2. Pitch เกี่ยวกับการหมุนขึ้น-ลงของวัตถุซึ่งทำได้โดยข้อต่อแบบ R (Rotational Joint)
3. Yaw เกี่ยวกับการหมุนซ้าย-ขวาของวัตถุซึ่งทำได้โดยใช้ข้อต่อแบบ R (Rotational Joint)

### 2.2.3 ส่วนประกอบของหุ่นยนต์

- 1) Actuator เป็นอุปกรณ์หรือชุดขับเคลื่อนเพื่อให้แขนกลหรือหุ่นยนต์เคลื่อนไหวได้แก่ มอเตอร์กระแสตรง เป็นต้น
- 2) Manipulators เป็นส่วนประกอบเพื่อช่วยเรื่องของการเคลื่อนที่ซึ่งประกอบด้วย แขน (Links) และข้อต่อ (Joints)
- 3) End Effectors เป็นส่วนท้ายสุดของแขนกลไว้ใช้ในการทำงานต่าง ๆ เช่น มือจับ สิ่งของอุปกรณ์ใช้ในทางเทคนิคประเภทงานกัด, กลึง, เชื่อม เป็นต้น
- 4) Sensor เป็นอุปกรณ์ที่ใช้ตรวจสอบสถานะตำแหน่งหรือข้อมูลที่ต้องการว่าได้อะไรหรือมีค่าเท่าไรเทียบกับประสาทสัมผัสของมนุษย์



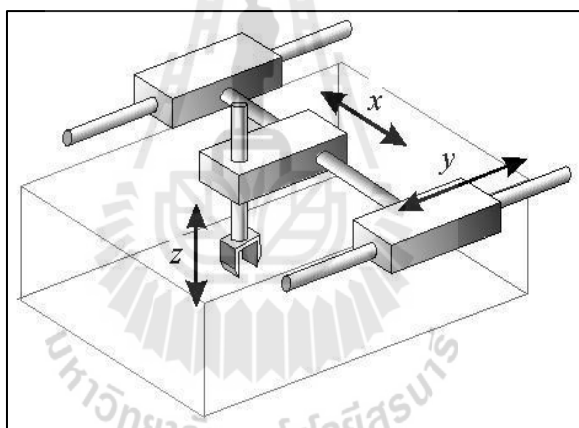
5) Power Supply เป็นแหล่งจ่ายพลังงานให้กับอุปกรณ์ต่าง ๆ ของแขนกล ไม่ว่าจะ เป็นกระแสไฟฟ้าหรือวาล์วที่ใช้ควบคุมกระบอกสูบ

6) Controller เปรียบเสมือนสมองของมนุษย์ที่คอยควบคุมขั้นตอนต่าง ๆ ในการทำงานของระบบแขนกล โดยสั่งการไปยังชุดขับเคลื่อนเพื่อเป็นไปตามโปรแกรมที่ตั้งไว้

7) Program เป็นการกำหนดขั้นตอนต่าง ๆ ของการทำงานเพื่อให้ได้งานตามที่ต้องการ

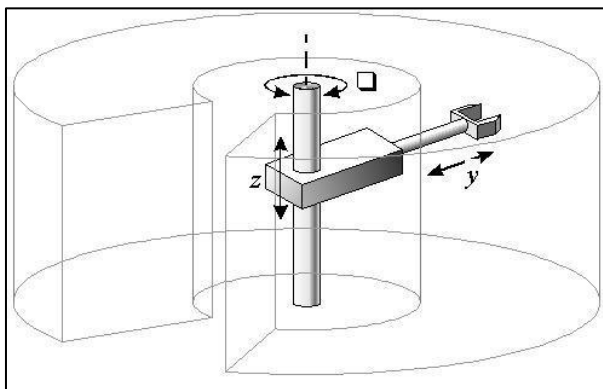
#### 2.2.4 ชนิดของหุ่นยนต์

1. The Cartesian Robot เป็นหุ่นยนต์ที่มีการเคลื่อนที่เป็นแบบเชิงเส้นทั้ง 3 แกน มีลักษณะการทำงานที่ไม่ซับซ้อนสามารถทำความเข้าใจได้ง่าย แต่หุ่นยนต์ชนิดนี้มีขีดจำกัดเรื่องของ พื้นที่ที่สามารถทำงานได้เฉพาะอาณาเขตการเคลื่อนที่เป็นแบบกล่องสี่เหลี่ยม ส่วนมากหุ่นยนต์ประเภทนี้จะนำไปใช้ในลักษณะเคลื่อนย้ายของหนัก ซึ่งหุ่นยนต์ประเภทนี้จะมีลักษณะดัง ภาพประกอบ 2.1



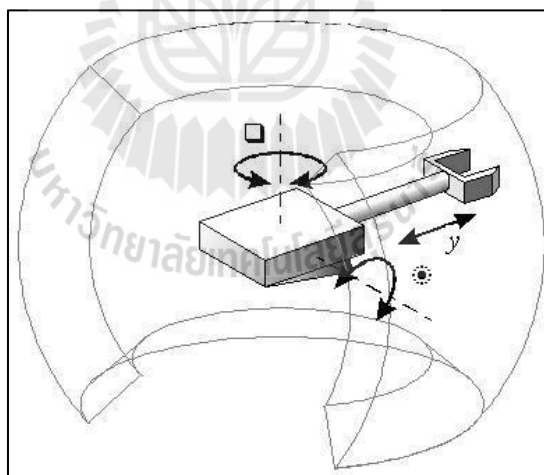
รูปที่ 2.1 แสดงตัวอย่าง The Cartesian Robot [3]

2 The Cylindrical Robot เป็นหุ่นยนต์ที่มีลักษณะของการทำงานแบบทรงกระบอก โดยการเคลื่อนที่นั้นจะเป็นเชิงเส้น 2 แกน และเคลื่อนที่เชิงมุมอีกแกน สามารถนำหุ่นยนต์ชนิดนี้ไปใช้ในลักษณะงานจับวางได้ ซึ่งหุ่นยนต์ประเภทนี้จะมีลักษณะดังรูปที่ 2.2



รูปที่ 2.2 แสดงตัวอย่าง The Cylindrical Robot [3]

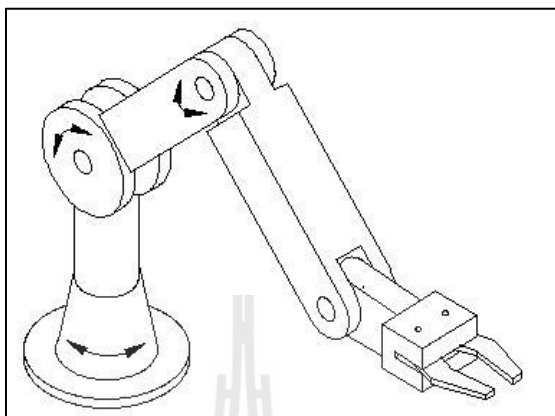
3 The Spherical Robot หุ่นยนต์ประเภทนี้เป็นหุ่นยนต์ที่มีลักษณะการเคลื่อนที่แบบเชิงเส้น 1 แกน กับการเคลื่อนที่เชิงมุมอีก 2 แกน หุ่นยนต์ประเภทนี้จะมีความสามารถในการเข้าถึงพื้นที่ที่มีความซับซ้อนได้ในระดับหนึ่ง แต่หุ่นยนต์ชนิดนี้จะไม่สามารถรับวัตถุหรือสิ่งของที่มีน้ำหนักมากได้ ซึ่งหุ่นยนต์ประเภทนี้จะมีลักษณะดังรูปที่ 2.3



รูปที่ 2.3 แสดงตัวอย่าง The Spherical Robot [3]

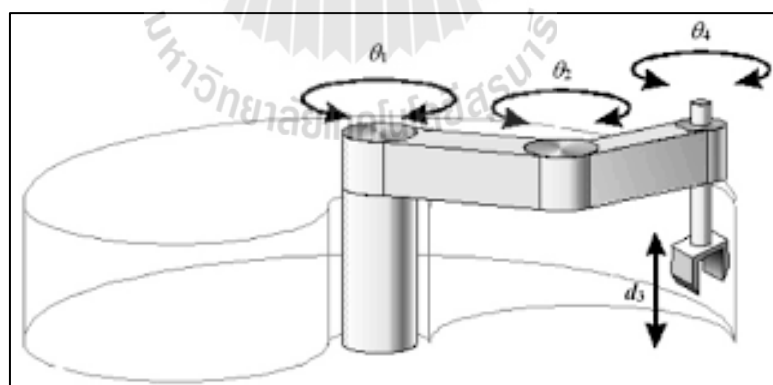
4 The Articulated Robot หรือ The Revolute Robot เป็นหุ่นยนต์ที่มีลักษณะการเคลื่อนที่แบบเชิงมุมทั้ง 3 แกน และมีพื้นที่การทำงานรอบตัว โดยหุ่นยนต์ชนิดนี้จะสามารถเข้าถึงพื้นที่ที่ยากต่อการเข้าถึงได้ โดยทั่วไปจะมี 2 แขน กับอีก 1 ฐาน ซึ่งหุ่นยนต์ชนิดนี้สามารถ

ประยุกต์ใช้งานได้อีกมากมาย เช่น การเพิ่มแขนและข้อต่อเข้าไปอีก ก็จะสามารถทำงานในจุดที่มนุษย์ไม่สามารถเข้าได้ โดยหุ่นยนต์ชนิดนี้มีลักษณะดังรูปที่ 2.4



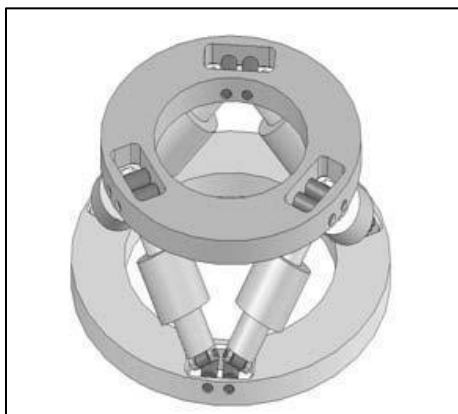
รูปที่ 2.4 แสดงตัวอย่าง The Articulated Robot หรือ The Revolute Robot [3]

5. The SCARA Robot เป็นหุ่นยนต์ที่มีลักษณะการเคลื่อนที่แบบเชิงเส้น 1 แกน และแบบเชิงมุมอีก 2 แกน โดยส่วนมากจะใช้ในการป้อนชิ้นงานตามรางขนส่ง (Conveyer) หรือประกอบชิ้นงานที่มีขนาดไม่ใหญ่นัก ซึ่งหุ่นยนต์ชนิดนี้มีลักษณะดังรูปที่ 2.5



รูปที่ 2.5 แสดงตัวอย่าง The SCARA Robot [3]

6. The Parallel Robot เป็นหุ่นยนต์อีกแบบหนึ่งที่มีลักษณะเป็นระนาบ 2 ระนาบควบคุมโดยขาแต่ละขา ซึ่งสามารถนำไปประยุกต์ใช้งานได้อีกมาก ดังรูปที่ 2.6

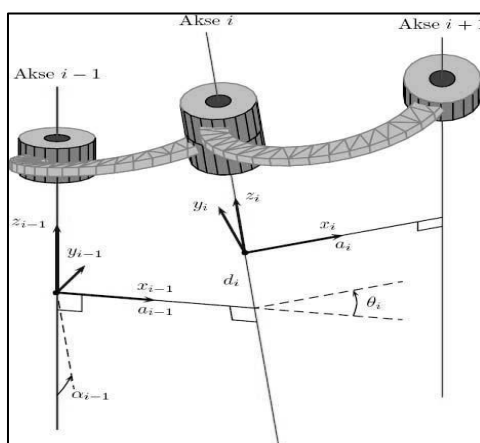


รูปที่ 2.6 แสดงตัวอย่าง The Parallel Robot [4]

### 2.2.5 การหา DH Parameters

DH Parameters ย่อมาจากคำว่า Denavit-Hartenberg Parameters เป็นการสร้างตัวแปรเพื่อนำมาใช้ในการหาสมการการเคลื่อนที่ของหุ่นยนต์ซึ่งมีตัวแปรทั้งหมด 4 ตัวได้แก่

- 1) Link Length ( $a_i$ ) คือ ระยะทางของ  $X_i$  ที่ตั้งฉากระหว่างแกน  $Z_i$  ถึงแกน  $Z_{i+1}$
  - 2) Link Twist ( $\alpha_i$ ) คือ มุมระหว่าง  $Z_{i-1}$  และ  $Z_i$  ที่มองจากหัวลูกศร  $X_{i-1}$  เข้าไป
  - 3) Link Offset ( $d_i$ ) คือ ระยะทางระหว่างแกน  $A_{i-1}$  ที่ตัดบนแกน  $Z_i$  กับแกน  $A_i$  ที่ตัดบนแกน  $Z_i$
  - 4) Joint Angel ( $\theta_i$ ) คือมุมระหว่างแกน  $X_{i-1}$  กับแกน  $X_i$  โดยพิจารณาบนแกน  $Z_i$
- จากคำอธิบายตัวแปรทั้ง 4 ตัวนั้นสามารถแสดงได้ ดังรูปที่ 2.7



รูปที่ 2.7 แสดงรายละเอียดของตำแหน่งตัวแปรต่าง ๆ ของ DH Parameters [5]

## 2.2.6 การหาเวกเตอร์การเคลื่อนที่ของหุ่นยนต์

หลังจากที่ได้ค่า DH Parameters แล้วต่อไปทำการหาค่าทรานฟอร์มเมชันเมตริกซ์ (Transformation Matrix: T) ซึ่งเป็นเมตริกซ์ขนาด 4x4 ซึ่งภายในเมตริกซ์นี้ประกอบไปด้วยเมตริกซ์การหมุนและเวกเตอร์การเคลื่อนที่ โดยการระบุค่าต่าง ๆ ของ DH Parameters ลงในสมการที่ 2.1

$$T_{i-1}^i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

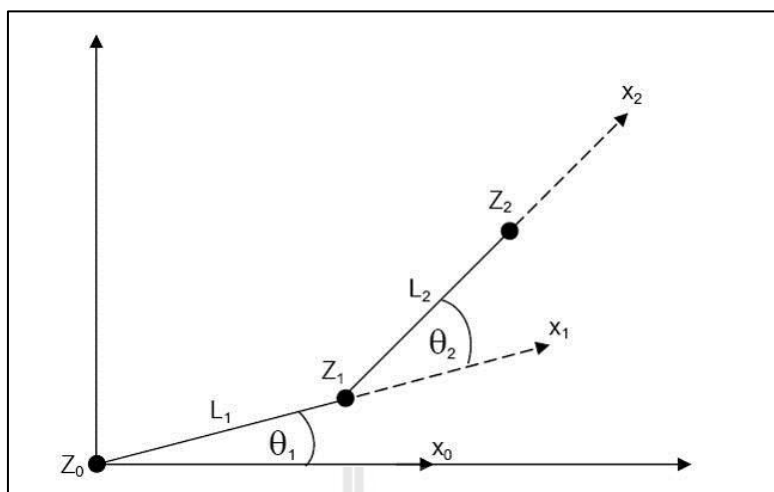
โดย  $T_{i-1}^i$  คือ ทรานฟอร์มเมชันเมตริกซ์ (Transformation Matrix)  
 $i$  คือ จำนวนแขน (Link)  
 $a_i$  คือ Link Length  
 $\theta_i$  คือ Link Twist  
 $d_i$  คือ Link Offset  
 $\alpha_i$  คือ Joint Angel

จากสมการที่ 2.1 จะได้เมตริกซ์การหมุนและเวกเตอร์การเคลื่อนที่ดังสมการที่ 2.2 และ 2.3

$$\text{สมการการหมุน} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \quad (2.2)$$

$$\text{เวกเตอร์การเคลื่อนที่} = \begin{bmatrix} a_i \cos \theta_i \\ a_i \sin \theta_i \\ d_i \\ 1 \end{bmatrix} \quad (2.3)$$

ตามลำดับหากหุ่นยนต์นั้นมีจำนวนแขน  $i$  แขน (Link) จะต้องนำค่าทรานฟอร์มเมชันเมตริกซ์ (T) มาคูณกันทางเมตริกซ์เช่นหุ่นยนต์มีแขน 2 แขน ดังรูปที่ 2.8



รูปที่ 2.8 แสดงตัวอย่างแบบจำลองของหุ่นยนต์ 2 แขน

จากรูปที่ 2.8 สามารถหาค่า DH Parameters โดยพิจารณาทีละแขนได้แขน  $L_1$  มี DH Parameters 4 ตัวดังนี้

$a_i$  มีค่า  $L_1$

$\theta_i$  มีค่า  $\theta_{1(r)}$

$d_i$  มีค่า 0

$\alpha_i$  มีค่า 0

แขนของ  $L_2$  มี DH Parameter 4 ตัว ดังนี้

$a_i$  มีค่า  $L_2$

$\theta_i$  มีค่า  $\theta_{2(r)}$

$d_i$  มีค่า 0

$\alpha_i$  มีค่า 0

หลังจากหาค่า DH Parameters เสร็จแล้วต่อไปเป็นการค่าหาทรานฟอร์มเมชันเมตริกซ์ (Transformation Matrix) ได้ดังนี้

$$T_0^i = \begin{vmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & L_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & L_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.4)$$

$$T_1^2 = \begin{vmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & L_2 \cos \theta_2 \\ \sin \theta_2 & 2 & 0 & L_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.5)$$

$$T_0^2 = \begin{vmatrix} \cos(\theta_1 - \theta_2) & -\sin(\theta_1 - \theta_2) & 0 & L_2 \cos(\theta_1 - \theta_2) \\ \sin(\theta_1 - \theta_2) & \cos(\theta_1 - \theta_2) & 0 & L_2 \sin(\theta_1 - \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.6)$$

จากสมการที่ 2.6 จะได้เวกเตอร์การเคลื่อนที่ (Translation vector) ของหุ่นยนต์ตามแนวพิกัด xyz ได้ดังนี้

- 1). แกน x มีเวกเตอร์การเคลื่อนที่คือ  $L_1 \cos \theta + L_2 \cos(\theta_1 + \theta_2)$
- 2). แกน y มีเวกเตอร์การเคลื่อนที่คือ  $L_1 \sin \theta + L_2 \sin(\theta_1 + \theta_2)$
- 3). แกน z มีเวกเตอร์การเคลื่อนที่คือ 0 (ไม่มีการเคลื่อนที่ในแนวแกน z)

เท่านี้จะได้เวกเตอร์การเคลื่อนที่ของแกนต่าง ๆ ของหุ่นยนต์ในหัวข้อต่อไป จะกล่าวถึงทฤษฎีเกี่ยวกับแรงเสียดทาน เพื่อใช้ในการวิเคราะห์ถึงการเคลื่อนไถลของวัตถุ และการทำให้วัตถุไม่เกิดการลื่นไถล

### 2.3 การควบคุมด้วยพีแอลซี (Programmable Logic Controller, PLC)

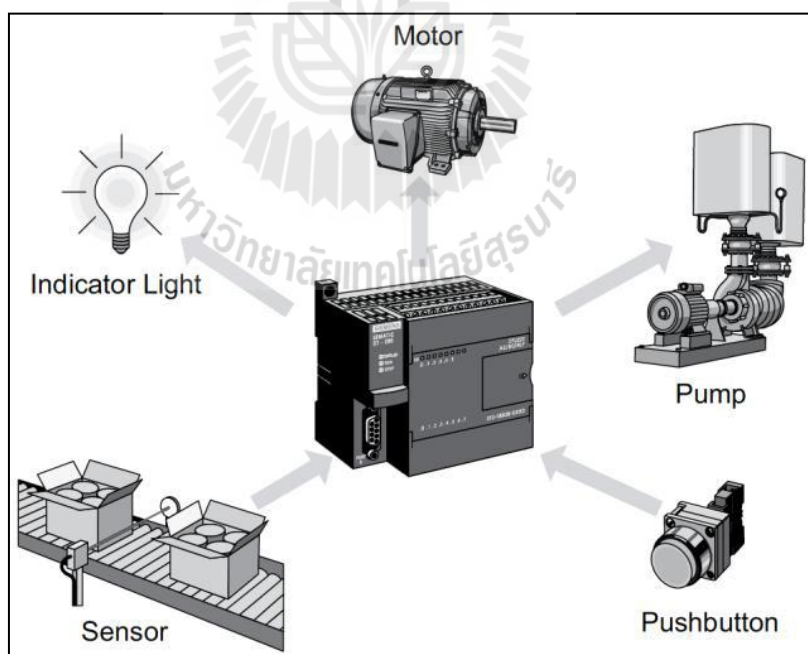
เนื่องจากงานวิจัยนี้ได้ใช้ PLC เป็นอุปกรณ์ควบคุมการเคลื่อนที่ของกลไกที่ใช้ในการจับยึดฮาร์ดดิสก์ไดรฟ์ ดังนั้นในหัวข้อนี้จะเป็นการกล่าวถึงพื้นฐานการทำงานของ PLC ในเบื้องต้น PLC เริ่มพัฒนาครั้งแรกในช่วงปี ค.ศ. 1968 เพื่อตอบสนองความต้องการของอุตสาหกรรมยานยนต์ในประเทศสหรัฐอเมริกา และเริ่มติดตั้งเพื่อใช้งานจริงในภาคอุตสาหกรรมการผลิตในปี ค.ศ. 1969 ซึ่งในช่วงแรกนี้ตรรกะ หรือ Logic ที่ใช้ยังไม่ซับซ้อนมาก และจะเน้นไปที่ On-Off Control

ในเวลาต่อมา PLC ก็ได้มีการพัฒนาขึ้น สามารถที่จะใช้ควบคุมอุปกรณ์ได้หลากหลายมากขึ้น สามารถที่จะส่งและรับสัญญาณที่มีค่าแรงเคลื่อนที่แตกต่างกันได้ ซึ่งทำให้ PLC ก้าวเข้าสู่ช่วงที่สามารถรับและจ่ายสัญญาณอนาล็อก (Analog Input/Output) ได้ ส่วนในช่วงยุคปี ค.ศ. 1980 ได้มีการกำหนดมาตรฐานสำหรับ PLC ที่รู้จักกันในชื่อ Manufacturing Automation Protocol (MAP) และมีการลดขนาดของ พีแอลซีให้เล็กลง นอกเหนือจากนั้นยังได้เริ่มมีการใช้โปรแกรมช่วยใน

การเขียนคำสั่งบน PC ในลักษณะของสัญลักษณ์ หรือ Symbolic Programming ทำให้สามารถที่จะทำการโปรแกรมได้โดยไม่ต้องใช้อุปกรณ์พ่วงต่อที่เขียนชุดคำสั่งลงไปบน PLC

ในยุคปี ค.ศ. 1990 เราได้พบเห็นการนำเสนอลักษณะการทำงาน กระบวนการทำงานและฟังก์ชันใหม่ ๆ ที่ใช้งานกับ PLC ซึ่งการพัฒนาเหล่านี้เกิดจากการแข่งขันของบริษัทผู้ผลิต ที่พยายามที่จะให้อุปกรณ์ของตนเองใช้งานได้ตรงตามความต้องการของลูกค้าให้มากที่สุด แต่เพื่อให้การพัฒนา PLC ยังคงอยู่ในมาตรฐานที่ยอมรับกันได้ทั่วโลก จึงได้มีการกำหนดมาตรฐานขึ้น โดยล่าสุดได้มีการกำหนดมาตรฐาน “IEC 1131-3” ซึ่งพยายามรวมภาษาการเขียนโปรแกรม PLC ให้เป็นมาตรฐาน ซึ่งในปัจจุบันนี้ ภาษามาตรฐานของ PLC ได้แก่ Function Block Diagrams, Instruction Lists C และ Structured Text ซึ่งการที่ผู้ใช้จะเลือกใช้วิธีการเขียนอย่างไรนั้นก็ขึ้นอยู่กับความเหมาะสมและความถนัดของผู้ใช้เป็นสำคัญ

สิ่งที่ทำให้ PLC เป็นที่นิยมอย่างกว้างขวางในแวดวงอุตสาหกรรมอัตโนมัติและได้นำมาใช้ในงานในงานวิจัยนี้ เพราะอุปกรณ์นี้มีความทนทาน ทนต่อสภาพอุณหภูมิที่เปลี่ยนแปลงไปได้ดี มีชุดอินพุตและเอาต์พุต บรรจุรวมเข้าอยู่กับชุดควบคุม และการเขียนชุดคำสั่งหรือภาษาที่ใช้ นั้นง่ายต่อความเข้าใจ อุปกรณ์ต่างที่เชื่อมต่อกับ PLC ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 แสดงอุปกรณ์ที่เชื่อมต่อกับ PLC เพื่อใช้ในการควบคุม

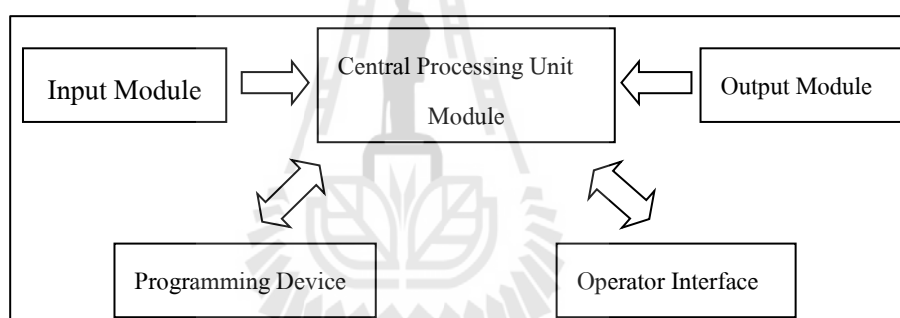


### 2.3.1 หลักการทำงานพื้นฐานของ PLC

ในการทำงานของ PLC จะมีส่วนประกอบอยู่หลายส่วนที่ทำงานสอดคล้องและประสานกัน สำหรับส่วนประกอบหลักของ PLC จะประกอบด้วยส่วนหรือ Module ต่าง ๆ ดังต่อไปนี้

- โมดูลอินพุท (Input Modules)
- หน่วยประมวลผลกลาง หรือ Central Processing Unit (CPU)
- โมดูลเอาต์พุท (Output Modules)
- อุปกรณ์ใช้ในการโปรแกรม (Programming Device)

ซึ่งการเชื่อมโยงการทำงานของอุปกรณ์ต่าง ๆ นี้ ดังแสดงในรูปที่ 2.10 และนอกเหนือจากส่วนประกอบหลักนี้แล้ว PLC ยังมีส่วนประกอบที่ทำหน้าที่เชื่อมโยงหรือประสานการทำงานของเครื่องร่วมกับผู้ใช้ หรือ Operator Interface ร่วมอยู่ด้วย สำหรับรายละเอียดและหน้าที่หลักของส่วนประกอบต่าง ๆ ที่กล่าวมาแล้วจะมีดังต่อไปนี้



รูปที่ 2.10 แสดงส่วนประกอบของ PLC

#### 1) โมดูลอินพุท (Input Module)

ลักษณะและประเภทของโมดูลอินพุทนี้จะแตกต่างกันออกไปตามชนิดของข้อมูลที่จะส่งให้กับ PLC ที่พบเห็นมากที่สุดจะเป็น DC voltage Digital Input หรือ Discrete Input คือ จะเป็นการจ่ายไฟที่แรงเคลื่อนไฟฟ้าค่าหนึ่ง อาจจะเป็น 5 หรือ 24 volt หรือไม่มีการจ่ายไฟให้ ซึ่งก็หมายความว่าเราจะได้ตรรกะของอินพุท (Logical Input) เป็นเปิด (ON) หรือปิด (OFF) จากโมดูลอินพุทนี้ อย่างไรก็ตามในปัจจุบันเริ่มพบเห็น PLC ที่รองรับอินพุทที่เป็นอนาลอกมากขึ้น คือสามารถที่จะตรวจจับสนิทไฟที่เข้ามาที่ช่องสัญญาณว่ามีปริมาณเท่าใดแทนที่จะตรวจจับสนิทไฟว่ามีหรือไม่มีไฟมาที่ช่องสัญญาณ ซึ่งจะทำให้สามารถตรวจจับสนิทไฟ ความดัน หรือปริมาณอื่น ๆ ได้ ซึ่งทำให้ระบบการควบคุมจะสมบูรณ์และซับซ้อนมากขึ้นได้

## 2) หน่วยประมวลผลกลาง

หน่วยประมวลผลกลางนี้ มีหน้าที่ประมวลผลที่ได้รับ โดยจะเริ่มจากการที่รับเอาข้อมูลจากโมดูลอินพุตทั้งหมดเข้ามาพิจารณา จากนั้นก็จะมีการประมวลผลตามตรรกะที่กำหนดไว้โดยโปรแกรม จากนั้นหน่วยประมวลผลกลางนี้จะทำหน้าที่ตัดสินใจ ทำการสั่งการ โมดูลเอาต์พุตให้เปิดหรือปิดช่องสัญญาณต่าง ๆ

สำหรับในหน่วยประมวลผลกลางนี้ จะทำหน้าที่เหมือนคอมพิวเตอร์ทั่วไป คือ จะมีหน่วยย่อย ๆ ประกอบอยู่ด้วย เช่น หน่วยความจำทั้ง ROM และ RAM นอกเหนือจากนั้นยังมีหน้าสัมผัสแบบต่าง ๆ เช่น ชุด Control Relay Special Relay Contacts และอื่น ๆ อีกมาก แต่เราจะไม่ขอกล่าวถึงในรายละเอียดในที่นี้

## 3) โมดูลเอาต์พุต

หน้าที่หลักของส่วนโมดูลเอาต์พุตนี้ จะทำหน้าที่เปลี่ยนสัญญาณที่ได้รับในการตัดสินใจของหน่วยประมวลผลกลาง ให้กลายเป็นสัญญาณควบคุมที่จะส่งออกไปทางช่องสัญญาณของโมดูลเอาต์พุตนี้ สำหรับช่องสัญญาณนี้อาจจะเป็นช่องสัญญาณแบบดิจิทัล(Digital) คือ มีเฉพาะสัญญาณเปิดหรือปิดเท่านั้น หรือจะเป็นสัญญาณแบบอนาลอกที่สามารถปรับค่าต่าง ๆ ให้เข้าหรือออกจากอุปกรณ์นี้ได้ตามความต้องการ

## 4) อุปกรณ์ใช้ในการโปรแกรม

อุปกรณ์เขียนและแก้ไขโปรแกรม หรืออุปกรณ์ในการเขียนโปรแกรมนี้นี้เป็นอุปกรณ์ที่มีหน้าที่ในการเขียนชุดคำสั่งที่ผู้ใช้ต้องการเข้าสู่เครื่อง PLC หรือทำหน้าที่ตั้งโปรแกรมที่บรรจุอยู่ในหน่วยประมวลผลกลาง CPU ของ PLC ออกมาสู่เครื่องนี้ นอกเหนือจากนี้ในบางรุ่นยังสามารถที่จะทำหน้าที่ตรวจสอบสถานะการทำงานของ PLC ว่าสามารถที่จะทำงานได้ตามที่ผู้เขียนโปรแกรมต้องการหรือไม่

## 5) การประมวลผลของ PLC (Processing)

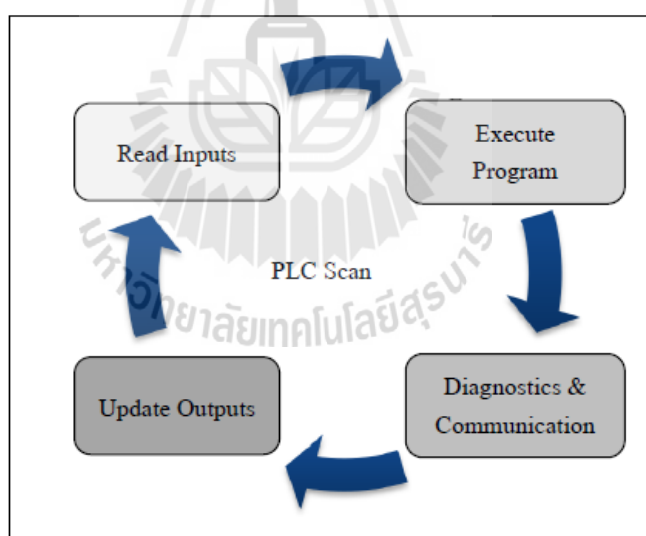
เมื่อมีส่วนประกอบหลายส่วนใน PLC จึงต้องมีการกำหนดกระบวนการ การทำงานของ PLC เอาไว้ ขั้นตอนการทำงานของ PLC นั้นจะมีอยู่ 2 โหมด (Mode) คือ

Edit Mode คือโหมดที่ผู้ใช้จะทำการเขียนโปรแกรมหรือแก้ไขโปรแกรม ซึ่งในระหว่างที่อยู่ในโหมดนี้ เครื่องจะไม่มีกรับข้อมูลจากอินพุตมาทำการวิเคราะห์ หรือส่งข้อมูลใด ๆ ออกไปทางช่องเอาต์พุต

Run Mode คือโหมดที่เครื่องจะทำงานตามที่ได้โปรแกรมไว้ โดยระหว่างที่อยู่ในโหมดนี้ จะไม่สามารถที่จะแก้ไขโปรแกรมได้ และสำหรับในบางผู้ผลิตจะมีโหมดการเฝ้าตรวจ (Monitor Mode) เพิ่มขึ้นมา โหมดนี้จะเป็นส่วนพิเศษเพิ่มเติมของ Run Mode โดยจะเพิ่มการ

แสดงผลสถานะต่าง ๆ ของ PLC ในขณะนั้นให้เราทราบไปพร้อมกับการทำงานของเครื่องไปด้วย เมื่ออยู่ใน Run Mode การทำงานของ PLC เริ่มจากเปิดเครื่องมีขั้นตอนดังนี้และลักษณะการทำงานของ PLC

- ลบคำสั่งเดิมที่โมดูลเอาต์พุต
- ดึงข้อมูลจากโมดูลอินพุตเข้ามา และประมวลสัญญาณทั้งหมดส่งให้หน่วยประมวลผลกลาง
  - หน่วยประมวลผลกลางรับข้อมูลอินพุต แล้วเปรียบเทียบกับตรรกะของโปรแกรม แล้ววิเคราะห์ว่าจะต้องดำเนินการเช่นไรบ้าง จากนั้นก็จะส่งสัญญาณตามข้อกำหนดที่วิเคราะห์ให้กับโปรแกรม ส่งไปให้กับโมดูลเอาต์พุต
  - โมดูลเอาต์พุตรับข้อมูลจากหน่วยประมวลผลกลางจากนั้นจะแปลงสัญญาณคำสั่งไปเป็นสัญญาณควบคุมการทำงานของชุดเอาต์พุตตามที่กำหนด
  - เมื่อครบรอบการทำงานนี้จะถือว่า PLC ทำงานครบหนึ่งรอบ หรือหนึ่ง Scan จากนั้นก็จะเริ่มรอบทำงานต่อไป โดยกลับเข้าไปอ่านสถานะของอินพุตว่ามีสถานะเป็นอย่างไร



รูปที่ 2.11 แสดงรอบการทำงานหนึ่งรอบของ PLC

#### 6) การกำหนดตำแหน่ง

ช่องสัญญาณอินพุตหรือเอาต์พุต แต่ละช่องจะต้องมีชื่อ หรือหมายเลขกำกับ เพื่อที่จะได้ทราบถึงความแตกต่างรวมทั้งอุปกรณ์ภายในที่ประเภทรีเลย์ (Relay) ต่าง ๆ ที่อยู่ภายในหน่วยประมวลผลกลาง ล้วนแต่จะต้องมีหมายเลขหรือชื่อกำกับด้วยกันทั้งสิ้น ซึ่งเราจะเรียกการ

กำหนดตัวเลขหรือตัวอักษรนั้นว่า Addressing ซึ่งการกำหนดนี้จะมีลักษณะของการกำหนด เป็นหมายเลขหรือตัวอักษรที่แสดงถึงช่องสัญญาณ หรือแสดงช่องสัญญาณ จากนั้นจะตามด้วย หมายเลขบิตหรือหมายเลขอุปกรณ์ที่อยู่ในช่วงสัญญาณนั้น ยกตัวอย่างเช่นอาจกำหนดหมายเลข อุปกรณ์แบบอินพุต 007 ก็หมายถึงว่าเป็นช่องสัญญาณในด้านอินพุตและเป็นอุปกรณ์หมายเลข 007 ในช่องสัญญาณนั้น

เนื่องจากมีผู้ผลิต PLC หลายบริษัทที่ใช้อยู่ในอุตสาหกรรมในประเทศของเรา และ ผู้ผลิตแต่ละรายจะมีการกำหนดรายละเอียดของการกำหนดตำแหน่งนี้จะแตกต่างกันออกไป นอกเหนือจากนี้จำนวนตำแหน่งของผู้ผลิตรายเดียวกันแต่รุ่น PLC ที่แตกต่างกันออกไปก็อาจจะ แตกต่างกัน หรือมีจำนวนไม่เท่ากัน ซึ่งจะไม่ขอกว่าถึงในรายละเอียดในที่นี่

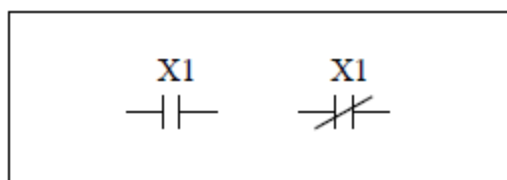
### 2.3.2 การโปรแกรม PLC

การจะโปรแกรม PLC นั้นในยุคแรกอาจมีวิธีการไม่มากนัก แต่ในยุคหลังนี้มี วิธีการให้เลือกเขียนได้หลายวิธีตามที่เราได้กล่าวไว้ก่อนหน้านี้แล้ว แต่ในเอกสารที่เราจะกล่าวถึง ต่อไปนี้จะให้ความสำคัญเพียงสองประเภท คือ การเขียนโปรแกรมขั้นบันไดหรือ Ladder Diagram ซึ่งสำหรับผู้ที่คุ้นเคยกับการเขียนวงจรไฟฟ้าจะเรียนรู้การเขียนขั้นบันไดมากกว่า และสำหรับผู้ผลิต แต่ละรายถึงแม้จะมีการกำหนดตำแหน่งที่แตกต่างกัน แต่ในการเขียนโปรแกรมขั้นบันไดนี้จะมี ความคล้ายคลึงกันมาก สำหรับแบบที่ 2 จะเป็นการเขียนแบบชุดคำสั่ง ซึ่งวิธีการนี้เป็นการใช้ ตัวหนังสือทั้งหมด ด้วยวิธีการนี้ผู้เขียนจะสามารถเขียนโปรแกรมโดยใช้อุปกรณ์ฟังก์ชัน เช่น Programming Console แก้ไขได้ง่าย สะดวกและรวดเร็ว ข้อเสีย คือ ในโปรแกรมที่ซับซ้อนนี้ต้อง อาศัยผู้ที่มีความเชี่ยวชาญในการแก้ไข และแต่ละผู้ผลิตจะมีการกำหนดวิธีการเขียนแบบชุดคำสั่ง (Instruction Code) ที่แตกต่างกันออกไปโดยสิ้นเชิง ลักษณะการเขียนโปรแกรมทั้งสองแบบเราจะ กล่าวถึงต่อไป

การเขียนโปรแกรมขั้นบันได เป็นการเขียนแผนผังกำหนดการทำงานของ PLC ซึ่ง ถ้าพิจารณาแล้วก็เป็นการพัฒนาจากการเขียนแผนภาพวงจรทางไฟฟ้านั่นเอง เพียงแต่การเขียนจะ มีการจัดระเบียบมากกว่า ทำให้มีรูปแบบเป็นขั้นบันได เราจึงเรียกมันว่า Ladder Diagram สัญลักษณ์ ที่เราจะใช้กับ Ladder Diagram จะประกอบด้วย Contact Coil และ Box เป็นหลัก ซึ่งรายละเอียดของ แต่ละตัวเป็นดังนี้คือ

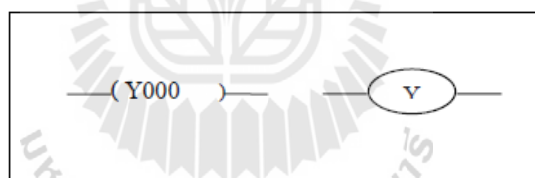
1) Contact จะแสดงเป็นหน้าสัมผัส เหมือนกับการเขียนวงจรไฟฟ้าทั่วไป ซึ่งจะ ประกอบด้วยหน้าสัมผัส 2 แบบคือ Normally Closed และ Normally Open ซึ่งจะมีสัญลักษณ์ ดัง แสดงในรูปที่ 2.12 ซึ่ง Relay แต่ละตัวที่เราเรียกใช้เราอาจจะเลือก ที่เป็นแบบ NO หรือ NC หรือจะ เรียกใช้ทั้งสองแบบในวงจรเดียวกันก็ได้ สำหรับในการเขียนโปรแกรมนั้น หน้าสัมผัสจะต้องมี

หมายเลข Address ของหน้าสัมผัสนั้นประกอบอยู่ด้วย ซึ่งจากรูปที่ 2.12 จะเป็นการกำหนดว่าเป็นหน้าสัมผัสของ Input Relay แบบปกติเปิด X0 และหน้าสัมผัส Input Relay แบบปกติปิด X1 ซึ่งเป็นการกำหนดที่ใช้กับเครื่องของบริษัท Mitsubishi



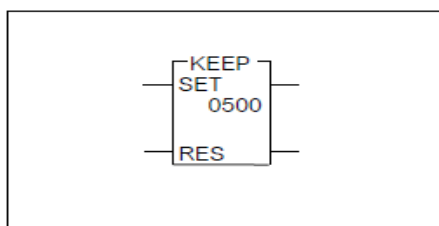
รูปที่ 2.12 แสดง Contact ใน PLC ตามข้อกำหนดของ Mitsubishi

2) Coil เป็นลักษณะของ Relay ที่อาจเป็น Output หรือ Relay พิเศษอื่น ๆ ก็ได้ ซึ่ง Coil แต่ละอัน ก็จะมี Contact ที่เป็นของ Coil นั้นด้วยและ Contact ก็จะสามารถเป็นได้ทั้งแบบ NO และ NC สำหรับสัญลักษณ์ของ Coil ที่ใช้กันอยู่มีสามแบบ ดังแสดงในรูปที่ 2.13 ซึ่งก็จะเหมือนกับหน้าสัมผัส ก็จะต้องมีหมายเลขหรือตัวอักษรกำกับ Address ของ Coil และ Relay นั้นด้วย



รูปที่ 2.13 แสดงสัญลักษณ์ของ Coil ที่นิยมใช้สามแบบตามบริษัทผู้ผลิต

3) Box กล่องหรือ Box นี้เป็นสัญลักษณ์แทน Relay ประเภทพิเศษแบบต่าง ๆ ที่เราจะต้องมีการกำหนดค่าเพิ่มเติมให้กับ Relay เหล่านั้น ซึ่งอาจจะเป็น Timer Relay ซึ่งต้องมีการกำหนดค่าเวลา หรือ Counter Relay ที่ต้องมีการกำหนดค่าจำนวนเหตุการณ์ที่จะนับลงไปด้วย สำหรับ Box นี้จะมีลักษณะแตกต่างกันออกไป ตามแต่ลักษณะของผู้ผลิต สำหรับกรณีทั่วไปแล้ว ดังแสดงในรูปที่ 2.33 ซึ่งใน Box นี้จะมีทั้งคำสั่งหรือ Instruction เช่น บอกว่าเป็น Timer และมี Operant ซึ่งจะบอกตำแหน่งของ Relay ระยะเวลาที่จะให้หน่วยเวลา เป็นต้น

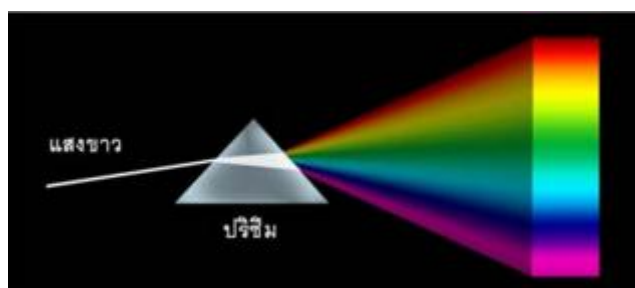


รูปที่ 2.14 แสดงสัญลักษณ์ของ Box ในรูปแบบต่าง ๆ

4) Internal and Special Relays นอกเหนือจากหน้าสัมผัสที่เป็น Input Relay และ Output Coil แล้วใน PLC ยังมี Relay และ Coil อื่นอีก โดยอุปกรณ์เหล่านั้นจะเป็นอุปกรณ์ที่อยู่ใน CPU ไม่สามารถที่จะต่อออกสู่ภายนอกได้จริง อุปกรณ์เหล่านั้นก็เช่น Timer Counter และ Internal Relay อื่น ๆ ซึ่งการกำหนด Address ของอุปกรณ์เหล่านี้จะขึ้นอยู่กับผู้ผลิตแต่ละราย เช่น ของบริษัท OMRON เราจะให้ Relay หมายเลข 008.00 008.01 ... หรือสำหรับของ บริษัท Mitsubishi จะใช้ M100 M101 ... แทน Internal Relay เหล่านี้ สำหรับจำนวนและหมายเลขของ Internal Relay แต่ละรุ่นนั้นขอให้ผู้ใช้ตรวจสอบกับคู่มือการใช้งานก่อนที่จะทำการเขียนโปรแกรม

## 2.4 ทฤษฎีเกี่ยวกับแสงและสี

สีตามความหมายของพจนานุกรมฉบับราชบัณฑิตยสถาน คือ ลักษณะของแสงที่ปรากฏแก่สายตาให้เห็นเป็นสี สำหรับทางวิทยาศาสตร์นั้นได้ให้คำจำกัดความของสีว่าเป็นคลื่นแสง หรือความเข้มของแสงที่สายตาสามารถมองเห็น ตัวอย่างเช่น เมื่อฝนหยุดตกแล้วมีแดดออก จะเห็นรุ้งกินน้ำซึ่งมีสีต่าง ๆ กัน หรือถ้าเราส่องแสงผ่านปริซึม แสงจะแตกออกมาเป็นสเปกตรัมของแสงสี เหล่านี้คือ สีแดง สีแสด สีเหลือง สีเขียว สีน้ำเงิน สีครามและสีม่วง ซึ่งสีเหล่านี้ คือ สเปกตรัมของแสงสีที่มองเห็นได้ ดังแสดงไว้ในรูปที่ 2.15

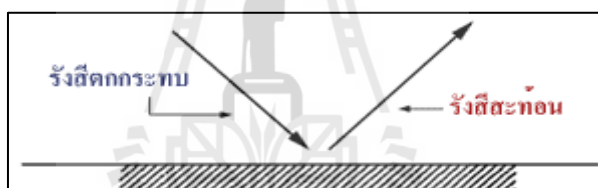


รูปที่ 2.15 แสดงแสงขาวและสเปกตรัม

แสงที่ส่งออกมาจากต้นกำเนิดอาจประกอบไปด้วยสีขาวย สีมixed หรืออาจเป็นสีใดสีหนึ่ง แสงจะเดินทางเป็นเส้นตรงจากต้นกำเนิดแสงมายังตาเรา โดยยังคงภาพของสีที่ถูกสร้างไว้พอกระทบกับวัตถุใด ๆ ก็ตาม วัตถุนั้นจะดูดกลืนแสงบางส่วนไว้ แล้วสะท้อนแสงบางส่วนออกมา สีที่เราเห็นนั้นเป็นสีที่เกิดจากการสะท้อนของแสงนั่นเอง การที่เราสามารถมองเห็นวัตถุ เพราะว่าวัตถุนั้นสะท้อนแสงออกมา สีที่ตามองเห็นถูกกำหนดโดยสีของแสงที่ชนกับวัตถุและสีของวัตถุที่สะท้อน การที่เราเห็นแผ่นกระดาษสีขาวเพราะว่า กระดาษสะท้อนทุกสีของแสงสีขาว ไม่ดูดกลืนสีใดเลย สมมุติว่าเราขายแสงสีน้ำเงินไปบนกระดาษสีขาว กระดาษจะกลายเป็นสีน้ำเงิน เพราะว่ากระดาษสีขาวจะสะท้อนแสงสีน้ำเงินออกมา

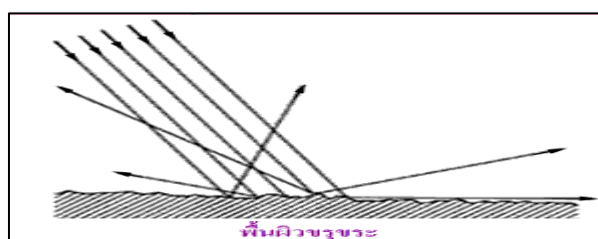
#### 2.4.1 คุณสมบัติของแสง

แสงจะมีคุณสมบัติที่สำคัญได้แก่ การเดินทางเป็นเส้นตรง (Rectilinear propagation), การหักเห (Refraction), การสะท้อน (Reflection) และการกระจาย (Dispersion) การสะท้อนแบบปกติ (Regular reflection) จะเกิดขึ้นเมื่อแสงตกกระทบกับวัตถุที่มีผิวเรียบมันวาวดังรูปที่ 2.16



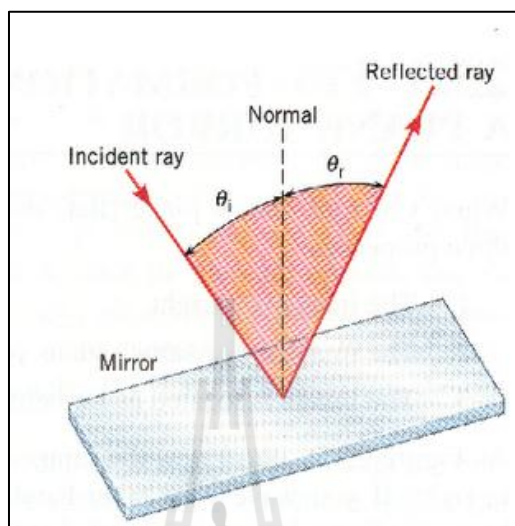
รูปที่ 2.16 แสดงการสะท้อนแบบปกติ

การสะท้อนแบบกระจาย (Diffuse reflection) จะเกิดขึ้นเมื่อแสงตกกระทบวัตถุที่มีผิวขรุขระดังรูปที่ 2.17



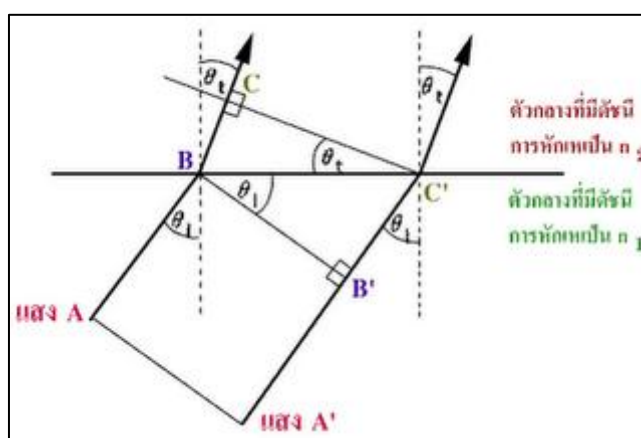
รูปที่ 2.17 แสดงการสะท้อนแบบกระจาย

โดยการสะท้อนของแสงไม่ว่าจะเป็นแบบใดก็ตามจะต้องเป็นไปตามกฎการสะท้อนของแสงที่ว่า "มุมสะท้อนเท่ากับมุมตกกระทบ" ซึ่งแสดงให้เห็นในรูปที่ 2.18



รูปที่ 2.18 แสดงกฎการสะท้อนของแสง

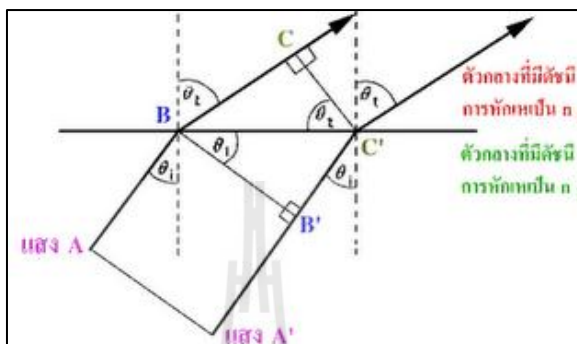
การหักเหการหักเหของแสง (Refraction) จะเกิดขึ้นเมื่อแสงเดินทางผ่านตัวกลางที่มีค่าดัชนีการหักเหไม่เท่ากัน โดยลำแสงที่ตกกระทบจะต้องไม่ทำมุมฉากกับรอยต่อระหว่างตัวกลางทั้งสอง และมุมตกกระทบต้องมีค่าไม่เกินมุมวิกฤต (Critical angle ; ) โดยการหักเหของแสงสามารถแบ่งออกได้เป็น 3 กรณี คือ  $n_1$  ดังแสดงในรูปที่ 2.19



รูปที่ 2.19 แสดงการหักเหของแสงกรณี  $n_1$



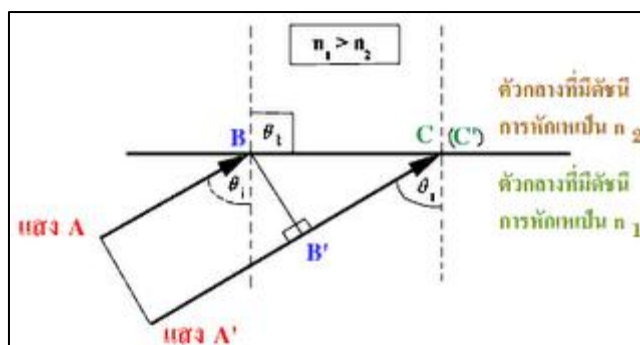
จากรูปที่ 2.19 ระยะเวลาที่แสงใช้ในการเดินทางในช่วง BC จะเท่ากับระยะเวลาที่แสงใช้ในการเดินทางในช่วง B'C' ซึ่งสามารถเขียนเป็นสมการได้ดังสมการ เมื่อพิจารณารูปสามเหลี่ยม BCC' และ BB'C' จะได้ความสัมพันธ์ทางตรีโกณ  $n_1 > n_2$  แสงจะหักเหออกจากเส้นปกติ ดังแสดงรูปที่ 2.20



รูป 2.20 แสดงการหักเหของแสงกรณี  $n_1 > n_2$

จากรูปที่ 2.20 จะเห็นว่าระยะทาง BC มีค่ามากกว่า B'C' เนื่องจากระยะทาง BC เป็นการเดินทางของแสงในตัวกลางที่มีค่าดัชนีการหักเหต่ำกว่า ดังนั้นในระยะเวลาเท่ากันแสงจะสามารถเดินทางได้มากกว่า

การเกิดการสะท้อนกลับหมดของแสง (Total Internal Reflection) จะเกิดขึ้นได้ก็ต่อเมื่อค่าดัชนีการหักเหของตัวกลางที่ 1 มีค่ามากกว่าดัชนีการหักเหของตัวกลางที่ 2  $n_1 > n_2$  และ ซึ่งจะให้ผลให้ มีค่าเท่ากับ หรือมากกว่าโดยเราสามารถหาค่าได้จาก Snell's Law เมื่อ จะเกิดการสะท้อนกลับหมดของแสงซึ่งจะได้ ดังนั้น ดังนั้นจะได้ ดังแสดงในตารางที่ 2.21

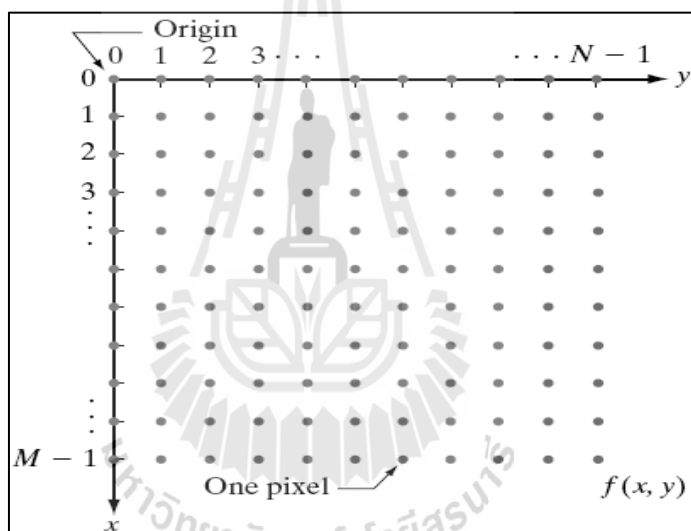


รูปที่ 2.21 แสดงตัวอย่างการสะท้อนกลับหมดของแสง

## 2.5 ความรู้พื้นฐานของภาพดิจิทัล

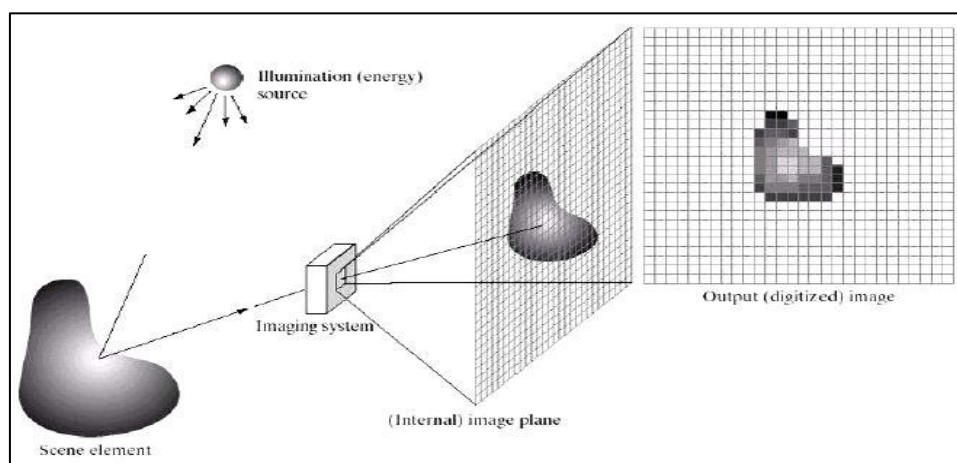
### 2.5.1 ภาพดิจิทัล (Digital Image)

ในทางการทฤษฎีนั้นภาพหนึ่งภาพสามารถที่จะกำหนดได้ด้วย ฟังก์ชัน 2 มิติ  $f(x, y)$  ซึ่งเป็นฟังก์ชันที่ใช้แสดงระดับของปริมาณความหนาแน่นของแสง (Light intensity) และเมื่อ  $f$  คือฟังก์ชันของความสว่างแล้วนั้นค่า  $x$  และ  $y$  นั้นจะแสดงถึงพิกัดที่ระบุตำแหน่งทางแนวนอนและแนวตั้งตามลำดับซึ่งอยู่ในพิกัดแกนของระบบภาพหรือเรียกว่า Spatial coordinate หรือเรียกพิกัด  $x$ - $y$  อีกอย่างว่า จุดภาพ (Pixel) โดยปกติแล้วนั้น Spatial coordinate นั้นจะเริ่มต้นด้วยพิกัดและจะเริ่มที่ตำแหน่งบนสุดและซ้ายสุดของภาพ โดยในแกนจะมีค่าเพิ่มขึ้นจากซ้ายไปขวาและในแกนจะมีค่าเพิ่มขึ้นจากบนลงล่างซึ่งแสดงได้ดังรูปที่ 2.22



รูปที่ 2.22 แสดงลักษณะของจุดเริ่มต้นของ Spatial coordinate

การกำเนิดภาพนั้นเกิดขึ้นจากอุปกรณ์สำหรับถ่ายภาพได้แปลงสัญญาณทางอิเล็กทรอนิกส์มาเป็นปริมาณทางดิจิทัล ซึ่งปริมาณนี้จะขึ้นอยู่กับระดับความละเอียดทางดิจิทัลว่าจะมีกี่บิต โดยในส่วนใหญ่แล้วจะได้ภาพที่มีระดับ 8 บิต โดยระดับความแตกต่างที่ได้จะสามารถหาได้จากปริมาณการแยกแยะของจำนวนบิตมาจาก  $2^n$  เมื่อคือจำนวนบิตที่ใช้ดังนั้นในที่นี้จะสามารถแยกแยะระดับได้  $2^8 = 256$  ระดับซึ่งในการกำหนดนั้นจะเริ่มที่ 0-255 ระดับซึ่งเรียกระดับที่แบ่งแยกได้นี้ว่าระดับเทา level โดยลักษณะการเกิดภาพดิจิทัลสามารถแสดงได้ดังรูปที่ 2.23



รูปที่ 2.23 แสดงลักษณะของการเกิดภาพทางดิจิทัล

### 2.5.2 คุณสมบัติของภาพดิจิทัล (Properties of Digitized Image)

- ความละเอียดของภาพ (Image resolution) คือ จำนวนของจุดภาพในแนวแกน x-y ซึ่งขึ้นอยู่กับความสามารถของอุปกรณ์ถ่ายภาพโดยยิ่งมีค่ามากภาพยิ่งมีความละเอียดมาก
- ความชัดของภาพ (Image Definition) คือ ระดับความสามารถที่ใช้แยกแยะปริมาณความแตกต่างของค่าที่ได้ในจุดภาพ เช่น ภาพ 8 บิตนั้นก็จะมี 256 ระดับแตกต่าง
- จำนวนของระนาบของภาพ (Number of Planes) เป็นจำนวนที่บอกให้ทราบถึงลำดับชั้นของจุดภาพ เช่น ในระบบภาพขาวดำหรือสองสีนั้นเรียกว่าเป็นภาพ 1 ระนาบและถ้าในกรณีภาพสีนั้นจะประกอบไปด้วยแม่สีแดง (Red) เขียว (Green) และน้ำเงิน (Blue) จะเรียกว่า 3 ระนาบ

### 2.5.3 ชนิดของภาพดิจิทัล (Image Types)

- ภาพระดับเทา (Gray scale Image) คือ ภาพที่แต่ละจุดภาพนั้นมี 1 ระนาบ โดยลำดับชั้นของระดับเทานั้นจะขึ้นอยู่กับปริมาณบิตที่ใช้ เช่น 8 บิต แบบจำนวนเต็มไม่กำหนดเครื่องหมายนั้นจะได้ลำดับชั้นของระดับเทาเป็น 256 ระดับ 0-255 และถ้าเป็น 16 บิต แบบกำหนดเครื่องหมายจะสามารถแบ่งลำดับชั้นของระดับเทาได้เป็น -32,768 ถึง +32,767 ระดับ ในแต่ละจุดภาพ
- ภาพสี (Color Image) คือ ภาพที่แต่ละจุดภาพนั้นมี 4 ระนาบเช่น RGB (Red, Green, Blue) และ Alpha หรือ HSL (Hue, Saturation, Luminance) และ Alpha ซึ่งระนาบ Alpha นั้นจะไม่ได้ใช้
- ภาพเชิงซ้อน (Complex Image) คือ ภาพที่มีข้อมูลเชิงความถี่ (Frequency Domain) ประกอบอยู่ด้วยซึ่งเกิดจากการทำ Fast Fourier Transform (FFT) เพื่อใช้ในการวิเคราะห์

### 2.5.4 ชนิดของข้อมูลภาพดิจิทัล (Image File Types)

ลักษณะของไฟล์ภาพนั้นเป็นส่วน of ข้อมูลที่จะบอกถึงคุณสมบัติของภาพนั้น ๆ ว่ามีคุณสมบัติเฉพาะของภาพเป็นเช่นใด เช่น ความละเอียดของภาพ ความชัดของภาพ และข้อมูลในส่วนการวิเคราะห์ด้านอื่น ๆ เช่น การวัดขนาดการเทียบเคียงรูปแบบและการซ้อนทับภาพ โดยชื่อของข้อมูลที่ใช้โดยทั่วไปนั้นแสดงได้ดังนี้

- Bitmap (.BMP)
- Tagged Image File Format (.TIFF)
- Portable Network Graphics (.PNG)
- Joint Photographic Experts Group (.JPEG)
- National Instruments Internal Image File Format (.AIPD)

### 2.5.5 ขอบภาพ (Image Borders)

ในการประมวลผลทางภาพนั้น บ่อยครั้งที่ต้องใช้ค่าที่ได้จากจุดภาพที่อยู่ข้างเคียงในการคำนวณ และในกรณีที่จุดภาพที่คำนวณนั้นอยู่ที่ตำแหน่งขอบของภาพ จะทำให้การคำนวณมีความผิดพลาดได้ เนื่องจากไม่มีค่าของจุดภาพข้างเคียงมาแทนค่าในการคำนวณ ซึ่งด้วยเหตุนี้เองขอบของภาพที่แท้จริง จึงต้องถูกกำหนดขึ้นเพื่อรองรับผลของการประมวลผลทางคณิตศาสตร์ ซึ่งเป็นขอบที่ถูกซ่อนอยู่ไม่ได้แสดงให้เห็นเรียกขอบของภาพที่เกิดขึ้นว่า ขอบภาพ (Image Borders) และขนาดของขอบภาพนั้นจะขึ้นอยู่กับลักษณะและวิธีการที่ใช้ในการประมวลผลทางภาพและค่าที่กำหนดในขอบภาพนั้นจะต้องให้เหมาะสมกับวิธีการที่ใช้ในการประมวลผลทางภาพเพื่อไม่ให้เกิดความผิดพลาดจากการนำขอบภาพมาคำนวณอีกด้วย

## 2.6 การหาขอบภาพแบบโซเบล (Sobel edge detector)

การหาขอบภาพ คือ การตรวจสอบว่าเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใด โดยวัดจากการเปลี่ยนแปลงของความเข้มในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว ซึ่งวิธีการหาขอบนั้นมีด้วยหลายวิธีสามารถแบ่งได้เป็น 2 กลุ่มหลัก คือ Gradient Method และ Laplacian Method สำหรับ Gradient Method จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพเช่น Sobel, Roberts, Prewitt, Canny เป็นต้น ส่วน Laplacian Method จะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า  $y$  เป็น 0 (Zerocrossing)

### 2.6.1 Gradient Method

วิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพโดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า Threshold จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Roberts, Prewitt, Sobel และ Canny เป็นต้น

### 2.6.2 Laplacian Method

จะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า  $y$  เป็น 0 (Zero crossing) ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่า Gradient Method ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น

ขั้นตอนการทำงานวิธี Sobel จะใช้ Gradient Operators โดยใช้หลักการหาค่าอนุพันธ์อันดับหนึ่งกับภาพที่มีลักษณะเป็นภาพ 2 มิติค่า Gradient ของภาพแทนด้วย  $f(x,y)$  ที่ตำแหน่ง  $(x,y)$  ซึ่งถูกกำหนดด้วย Vector ดังสมการต่อไปนี้

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.7)$$

สิ่งที่ได้มาจากวิเคราะห์ Vector ทำให้ได้ค่า  $f$  ที่เกิดจากการเปลี่ยนแปลงในระดับมากที่สุดของ Gradient Vector ในทิศทางต่าง ๆ ที่ตำแหน่ง  $(x,y)$  และที่สำคัญที่สุดในเชิงปริมาณในการหาเส้นขอบก็คือค่า  $\nabla f$  เมื่อ

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} \quad (2.8)$$

ทิศทางของ Gradient vector สามารถคำนวณโดยให้  $(x,y)$  แทนด้วยทิศทางมุมของ Vector  $\nabla f$  ที่  $(x,y)$  โดยใช้สมการ

$$\alpha(x, y) = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad (2.9)$$

เมื่อมุม คือ ระยะทางของแนวแกน x ทิศทางของขอบที่ตำแหน่ง (x,y) คือ เส้นตั้งฉากไปยังทิศทางของ Gradient vector ที่จุดนั้น ๆ การทำงานของ Sobel อาศัยวิธีการหาค่า Gradient Vector และใช้ Mask ที่มีขนาด 3 x 3 สำหรับเป็นตัวกระทำการในขั้นตอนวิธี

$z_1$	$z_2$	$z_3$	-1	-2	-1	-1	0	1
$z_4$	$z_5$	$z_6$	0	0	0	-2	0	2
$z_7$	$z_8$	$z_9$	1	2	1	-1	0	1

ก. การแทนที่ตำแหน่งต่างๆ ของตัวกระทำ

ข. ตัวกระทำของวิธีหา ขอบภาพแบบ Sobel ในแนว  $G_x$

ค. ตัวกระทำของวิธีหา ขอบภาพแบบ Sobel ในแนว  $G_y$

รูปที่ 2.24 แสดงตัวกระทำของการหาขอบภาพด้วยวิธี Sobel

เมื่อได้ Mask ที่มีขนาด 3 x 3 เรียบร้อยแล้วก็จะนำไปแทนที่ในสมการโดยค่า  $G_x$  และ  $G_y$  สามารถคำนวณได้จากสมการต่อไปนี้

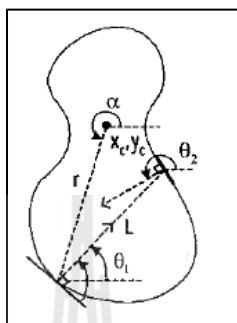
$$G_x = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3) \quad (2.10)$$

$$G_y = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7) \quad (2.11)$$

## 2.7 การค้นหาวัตถุในภาพโดยวิธีโมดิฟายเจเนอรัลไลฟัฮ์ฟทรานส์ฟอร์ม (Modified Generalized Hough Transform: MGHT)

(Tipwai and Madarasmi 2002 : 437-440; Nimkerphol et al. 2000 : 108-113; Nimkerdphol and Madarasmi 2001) ได้พัฒนาเทคนิค MGHT ซึ่งเทคนิคดังกล่าวเป็นขั้นตอนวิธีที่เกิดมาจากการพัฒนา Generalized Hough Transform (GHT) ทำให้มีการลดขั้นตอนการทำงานลงเป็น  $O(N^2)$  จาก  $O(RSN^2)$  โดยสามารถค้นหาวัตถุในภาพจากวัตถุต้นแบบได้แม้กระทั่งวัตถุที่มีขนาด หรือรูปร่างที่ผิดไปจากเดิมซึ่งขั้นตอนวิธีนี้จะใช้เมื่อรูปร่างวัตถุที่ต้องการไม่สามารถหาได้ด้วยสมการ

ทางคณิตศาสตร์ดังนั้นแทนที่จะใช้สมการเส้นโค้งแต่จะใช้เป็นค่าที่อยู่ในรูปแบบของตารางค่า เรียกว่า Look-up Table ซึ่งระบุถึงรายละเอียดของวัตถุซึ่งในที่นี้เรียก Look-up Table ว่าตาราง R (R-Table) เทคนิค MGHT จะมีวิธีการ 2 ขั้นตอน คือ ขั้นตอนการเรียนรู้ (Learning) และขั้นตอนการค้นหาค้นหา (Detecting)



รูปที่ 2.25 แสดงคุณลักษณะของภาพต้นแบบ

### 2.7.1 การเรียนรู้ (Learning)

การเรียนรู้ คือ การเก็บข้อมูลของวัตถุที่ต้องการค้นหาจากภาพโดยจะนำข้อมูลของภาพที่ได้เก็บไว้ในรูปค่าความสัมพันธ์ในตาราง R (R-Table) ดังรูปที่ 2.9 โดยมีลำดับขั้นตอนการทำงานดังนี้

- หาจุดอ้างอิง  $(x_c, y_c)$
- ลากเส้นจากจุดอ้างอิง  $(x_c, y_c)$  ไปยังขอบภาพโดยให้  $r$  คือความยาวจากจุดอ้างอิง  $(x_c, y_c)$  ไปยังขอบภาพจะทำให้ได้เกรเดียนท์  $\theta$  และ  $\alpha$  คือมุมที่เส้นตรง  $r$  ทำกับแกน  $x$  จากสมการ
- จากขอบภาพเดิมเกรเดียนท์  $\theta$  ลากเส้นในแนวตั้งฉากภายในจนไปยังเกรเดียนท์  $\theta'$  โดยมี  $L$  คือ ความยาว
- นำค่าไปใส่ในตาราง R ประกอบด้วย  $(\theta, \theta', L, \alpha, r)$  เมื่อ  $\forall \theta = (\theta, \theta')$  ดังรูปที่ 2.10
- จากจุดอ้างอิงเดิม  $(x_c, y_c)$  ทำซ้ำจนครบขอบภาพ

$$\begin{aligned} & \theta_1 (\theta_1^1, \theta_1^1, L_1^1, \alpha_1^1, r_1^1), (\theta_2^1, \theta_2^1, L_2^1, \alpha_2^1, r_2^1), \dots \\ & \theta_1 (\theta_1^2, \theta_1^2, L_1^2, \alpha_1^2, r_1^2), (\theta_2^2, \theta_2^2, L_2^2, \alpha_2^2, r_2^2), \dots \\ & \theta_1 (\theta_1^n, \theta_1^n, L_1^n, \alpha_1^n, r_1^n), (\theta_2^n, \theta_2^n, L_2^n, \alpha_2^n, r_2^n), \dots \end{aligned}$$

รูปที่ 2.26 แสดงข้อมูลภาพต้นแบบในตาราง R

สำหรับ R-Table สามารถสร้างได้หลาย ๆ ตารางตามความแตกต่างของวัตถุแต่ละภาพ

## 2.7.2 การค้นหา (Detection)

การค้นหา คือ การหาตำแหน่งที่อยู่ของข้อมูลที่ได้จากการเรียนรู้ในอีกภาพหนึ่ง ซึ่งอาจจะมีวัตถุที่เหมือนกับวัตถุต้นแบบ ซึ่งได้จากการเรียนรู้มากกว่า 1 วัตถุในรูปภาพนั้น

1. สำหรับทุกจุดในขอบของภาพ (x,y)

- ลากเส้นในแนวเต้าภายในวัตถุจนไปชนกับจุดที่ขอบภาพหรือถึงจุดความยาวมากที่สุด (scalemax : L) โดย LC คือ ความยาวจากจุดเดิมที่มีเกรเดียนท์  $\theta_c$  ไปยังเกรเดียนท์  $\theta'_c$  (จุดที่ลากไปชน)

- นำค่า  $\nabla\theta = (\theta_c, \theta'_c)$  ไปเปรียบเทียบกับค่าในตาราง R โดยที่  $\nabla\theta \approx \theta$

- นำค่าออกมาซึ่งก็  $(\theta, \theta', L, \alpha, r)$

$$X_c = x + S \cdot r \cdot \cos(\alpha + \beta) \quad (2.12)$$

$$Y_c = y + S \cdot r \cdot \sin(\alpha + \beta) \quad (2.13)$$

เมื่อ S คือ Scale Factor, r คือ ความยาวจากจุดอ้างอิง  $(x_c, y_c)$  ไปยังขอบภาพจะทำให้ได้เกรเดียนท์  $\theta$  และ  $\beta$  คือ Rotation Angle จากสมการดังต่อไปนี้

$$s = L_c / L \quad (2.14)$$

$$\beta = \theta_c / \theta \quad (2.15)$$

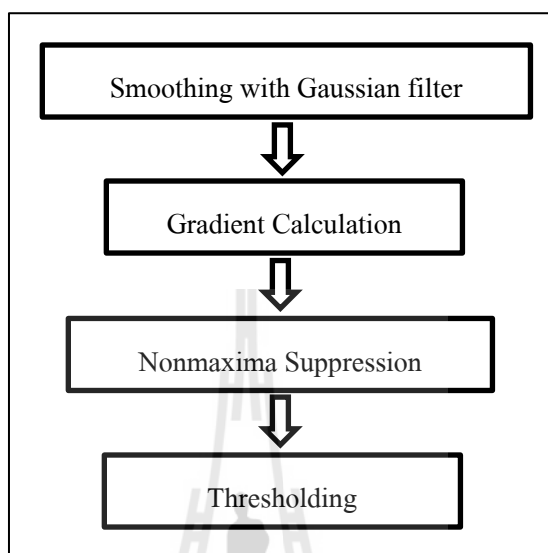
เมื่อ  $L_c$  คือ ความยาวระหว่างจุดเดิมที่มีเกรเดียนท์  $\theta_c$  ไปยังเกรเดียนท์  $\theta'_c$  ในภาพปลายทางและ L คือ ความยาวระหว่างจุดเดิมที่มีเกรเดียนท์  $\theta$  ไปยังเกรเดียนท์  $\theta'$  ในตาราง R  $\theta_c$  คือ ค่าเกรเดียนท์ในภาพปลายทางและ  $\beta$  คือ ค่าเกรเดียนท์ในตาราง R

- คำนวณหา  $x_c, y_c$  จากสมการดังต่อไปนี้
- นำค่า  $x_c, y_c$  ที่ได้ไปหาจุดที่มีค่าสูงสุดที่ผ่านเกณฑ์โดยถ้าหาจุด  $x_c, y_c$  ของจุดบนขอบได้แล้วให้ไปเพิ่มค่าในตัวแปรสะสมค่า A (Accumulator) เพื่อหาจุดที่มีค่าสูงสุด
- ตำแหน่งที่หาได้จะเป็นจุดที่มีค่าสูงสุดผ่านเกณฑ์ที่กำหนด



## 2.8 การหาขอบภาพแบบแคนนี่ (Canny Edge Detection Algorithm)

ขั้นตอนการหาขอบโดยวิธีของ Canny ประกอบด้วย 4 ขั้นตอนดังรูปที่ 2.27



รูปที่ 2.27 แสดงขั้นตอนของ Canny edge detection

การทำงานของ Canny edge detection นั้นเริ่มต้นจากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาส์เซียน (Gaussian filter) เพื่อกำจัดสัญญาณรบกวนหลังจากนั้นคำนวณค่าขนาด (Magnitude) และทิศทาง (Orientation) ของ Gradient โดยใช้การหาอนุพันธ์อันดับหนึ่งในถัดมาจึงใช้ Nonmaxima Suppression กับ Gradient Magnitude เพื่อทำให้ได้ขอบที่บางลงและในขั้นตอนสุดท้ายใช้ Double Thresholding Algorithm เพื่อระบุพิกเซลที่เป็นขอบและช่วยเชื่อมต่อขอบ (Green, Bill 2002; ION528-Image processing algorithms 2005; Canny Operator Links 2005; Rubino, Matthew 2005) โดยในแต่ละขั้นตอนมีรายละเอียดดังต่อไปนี้

### 2.8.1 Smoothing

ในขั้นตอนแรกของการหาขอบโดยอัลกอริทึมนี้จะต้องกำจัดสัญญาณรบกวนออกก่อนโดยใช้ Gaussian Filter ซึ่งสามารถคำนวณได้จากการใช้กรอบ (Mask) ขนาดเล็กขนาดของ Gaussian Mask นี้หากมีขนาดกว้างจะมีผลทำให้ลดสัญญาณรบกวนได้มากแต่ถ้ากว้างมากเกินไปจะมีผลทำให้ขอบย่อย ๆ ที่เป็นส่วนรายละเอียดนั้นหายไป สำหรับการคำนวณหาภาพที่ได้จากการใช้ Gaussian Filter เป็นดังสมการที่ 2.16

$$S[i, j] = G[i, j, \sigma] * I[i, j] \quad (2.16)$$

กำหนดให้

$I[i, j]$	เป็นภาพที่ต้องการหาขอบ
$G[i, j, \sigma]$	เป็น Gaussian smoothing filter
$\sigma$	เป็น Spread of the Gaussian (ควบคุมระดับของการ Smoothing)

### 2.8.2 Gradient Calculation

ในขั้นแรกนำ Smoothing Image  $S[i, j]$  มาสร้าง  $x, y$  Partial Derivatives  $P[i, j]$  และ  $Q[i, j]$  ตามลำดับดังสมการที่ 2.17 และ 2.18

$$P[i, j] \approx (S[i, j+1] - S[i, j] + S[i+1, j+1] - S[i+1, j]) / 2 \quad (2.17)$$

$$Q[i, j] \approx (S[i, j] - S[i+1, j] + S[i, j+1] - S[i+1, j+1]) / 2 \quad (2.18)$$

หลังจากนั้นนำค่า  $x, y$  Partial Derivatives มาคำนวณด้วยสูตรมาตรฐานสำหรับการแปลงรูปแบบจาก Rectangular ไปเป็น Polar (Rectangular-to-Polar Conversion) เพื่อหาขนาดและทิศทางของ Gradient ตามสมการที่ 2.19

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad (2.19)$$

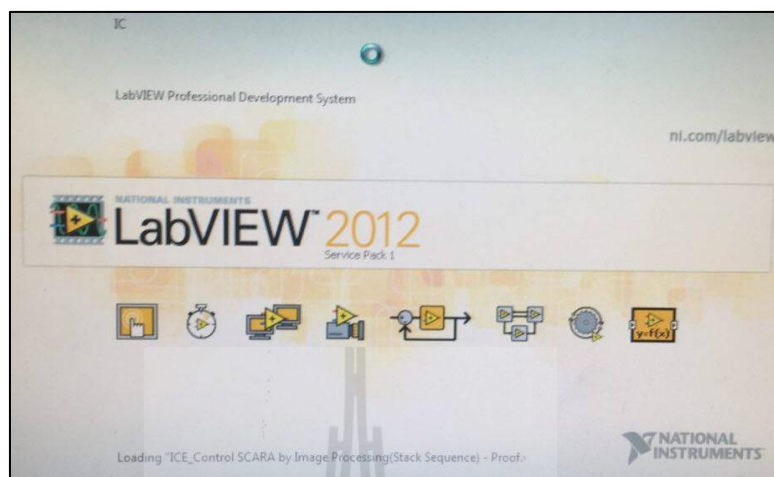
$$\theta[i, j] = \arctan(Q[i, j] / P[i, j]) \quad (2.20)$$

จากสมการข้างต้นจะสามารถหาค่ามุม  $\theta$  ออกมาได้เมื่อแทนค่าตัวแปรในฟังก์ชัน  $\arctan(x, y)$

### 2.8.3 Nonmaxima Suppression

สำหรับการหาขอบโดย Canny Method จุดที่ถือเป็นเส้นขอบได้นั้น ต้องเป็นจุดที่ให้ค่าสูงสุดเฉพาะที่และเป็นทิศทางเดียวกับ Gradient ด้วยซึ่งด้วยวิธีดังกล่าวนี้ทำให้ได้ขอบที่บางเพียง 1 พิกเซลภาพที่ได้หลังการทำ Nonmaxima Suppression จะให้ค่าเป็นศูนย์ในทุกจุดยกเว้นจุดที่เป็น Local Maxima Points ซึ่งจะยังคงค่าเดิมไว้

## 2.9 ทฤษฎี LabVIEW



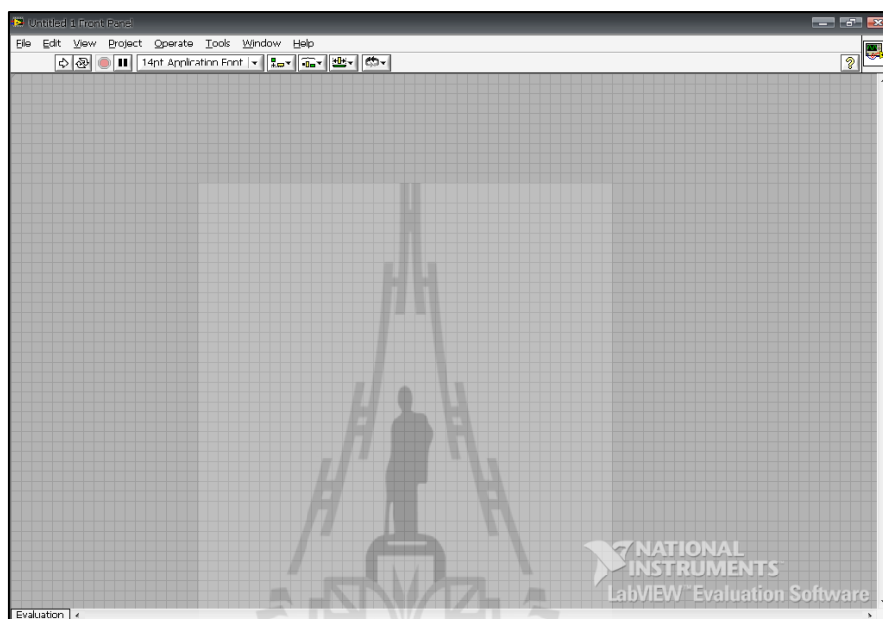
รูปที่ 2.28 แสดงโลโก้ของ LabVIEW

LabVIEW ย่อมาจาก Laboratory Virtual Instrument Engineering Workbench ซึ่งเป็นบริษัทผู้ผลิตผู้พัฒนาอุปกรณ์และซอฟต์แวร์ที่ใช้ในการวัดและงานควบคุมอัตโนมัติ โดย LabVIEW เป็นซอฟต์แวร์ที่ได้รับการพัฒนามาตั้งแต่ปี ค.ศ. 1893 และเปิดตัว LabVIEW 1.0 LabVIEW จัดเป็นซอฟต์แวร์ประเภท Programming Language จะคล้ายกับโปรแกรม Visual Basic แตกต่างกันตรง LabVIEW สามารถโปรแกรมได้เชิงกราฟิก โดยซอฟต์แวร์จะสร้างไบลลารีหรือผู้ใช้งานสร้างขึ้นเอง จะคล้ายเครื่องมือเสมือน VI (Virtual Instrument) ซึ่งสามารถเชื่อมต่อกันด้วยไอคอน (Icon) แทนการเขียนโปรแกรมย่อย (Subroutine) และใช้เส้นเชื่อมต่อระหว่างบล็อกฟังก์ชันแทนการไหลของข้อมูลระหว่างโปรแกรมย่อยนั้น ๆ คล้ายการเขียนบล็อกไดอะแกรม (Block Diagram)

### 2.9.1 ส่วนประกอบต่างๆใน LabVIEW

โปรแกรมที่เขียนขึ้นมาโดย LabVIEW เราจะเรียกว่า Virtual Instrument (VI) เพราะลักษณะที่ปรากฏทางจอภาพเมื่อผู้ใช้งานจะเหมือนกับเครื่องมือหรืออุปกรณ์ทางวิศวกรรม ในขณะเดียวกันหลังจากของอุปกรณ์เสมือนจริงเหล่านั้นจะเป็นการทำงานของ ฟังก์ชัน, Subroutines และโปรแกรมหลักเหมือนกับภาษาทั่วไป สำหรับ VI หนึ่ง ๆ จะประกอบด้วยส่วนประกอบ 3 ส่วน คือ 1. Front Panel 2. Block Diagram 3. Icon และ Connector ทั้งหมดนี้จะประกอบกันขึ้นมาเป็นอุปกรณ์เสมือนจริง ลักษณะและหน้าที่ของส่วนประกอบทั้ง 3 มีดังต่อไปนี้

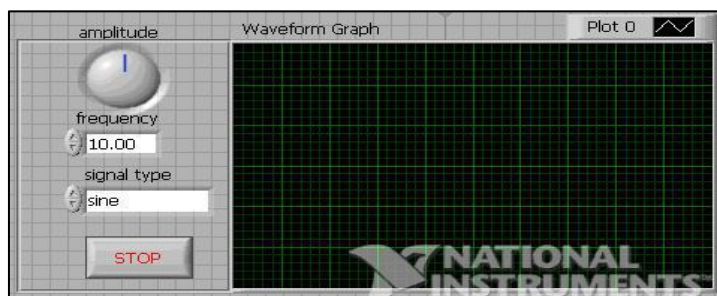
2.9.1.1 Front Panel จะเป็นส่วนที่ใช้สื่อความกันระหว่างผู้ใช้กับ โปรแกรม (หรือที่นิยมเรียก User Interface) ซึ่ง Front Panel จะประกอบด้วยส่วนรับคำสั่งควบคุม เช่น ปุ่ม, สวิตช์, เป็นต้น ส่วนต่าง ๆ เหล่านี้สามารถเลือกได้จากไอคอนใน Control Palette แล้วนำมาวางประกอบกันเป็น Front Panel โดย ดังแสดงในรูปที่ 2.29



รูปที่ 2.29 แสดงลักษณะของ Front Diagram

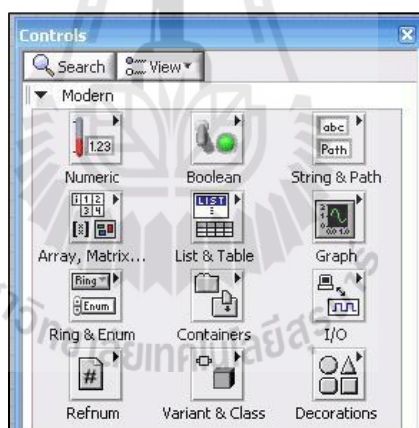
Object ที่อยู่บน Front Panel จะมีอยู่ 3 ประเภท คือ

- 1) Control คือ ประเภทที่รับค่าจากผู้ใช้ (Input) ซึ่งผู้ใช้สามารถพิมพ์ค่าลงไป หรือใช้เมาส์คลิกเพื่อเปลี่ยนแปลงค่าได้ เช่น ปุ่มหมุน ปุ่มเลื่อน สวิตช์ เป็นต้น
- 2) Indicators คือ ประเภทที่ใช้แสดงค่าต่าง ๆ เท่านั้น (Output) ผู้ใช้ไม่สามารถแก้ไขได้ เช่น กราฟ, มิเตอร์, LED
- 3) Decorations เป็น Object ที่ไม่เกี่ยวข้องกับโปรแกรมและ Code บน Block Diagram เลยแต่มีไว้เพื่อความสวยงามเป็นระเบียบของ Front Panel เท่านั้นนั่นเอง ดังแสดงรูปที่ 2.30



รูปที่ 2.30 แสดง Object ที่อยู่บน Front Panel ของ LabVIEW

เครื่องมือที่ใช้ในการออกแบบ Front Panel จะประกอบไปด้วย Control Palette และ Tools Palette ซึ่ง LabVIEW มี Controls Palette ที่ใช้ในการออกแบบ Front Panel แสดงดังรูป 2.8 ซึ่งเป็นส่วนที่ติดต่อกับผู้ใช้งาน (User Interface) โดยจะจัดเป็นกลุ่มต่าง ๆ เช่น กลุ่มของตัวเลข (Numeric) ซึ่งภายในกลุ่มจะมี Control และ Indicator ต่าง ๆ ที่เกี่ยวกับตัวเลข ดังแสดงในรูปที่ 2.31



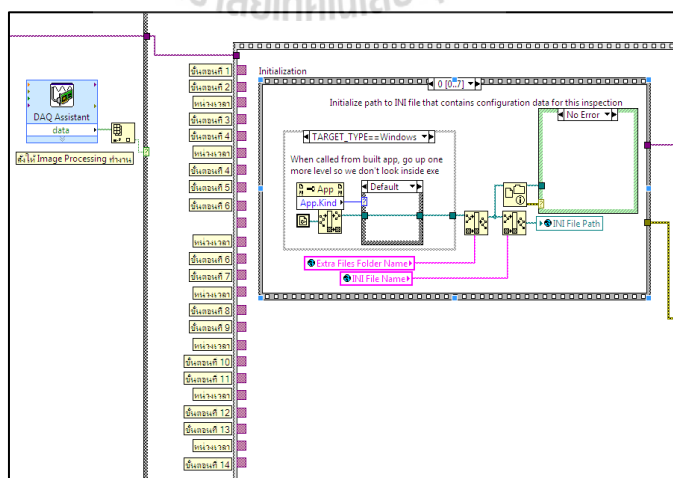
รูปที่ 2.31 แสดง Controls Palette ที่ใช้ในการออกแบบ Front Panel

Tools Palette คือ เครื่องมือที่ใช้ในการพัฒนาโปรแกรม ซึ่งจะใช้ทั้งการออกแบบ Front Panel และ Block Diagram ในส่วนนี้จะกล่าวถึง Tools Palette สำหรับออกแบบ Front Panel ดังแสดงในรูปที่ 2.32



รูปที่ 2.32 แสดง Tools Palette ที่ใช้ในการออกแบบ Front Panel

2.9.1.2 Block Diagram เพื่อให้เกิดความเข้าใจง่ายขึ้น เราอาจมอง Block Diagram นี้ เป็นเสมือนกับ Source Code หรือโปรแกรมของ LabVIEW ซึ่งปรากฏอยู่ในรูปของภาษา G ซึ่ง Block Diagram นี้ ถือว่าเป็น Executable Program คือ สามารถที่จะทำงานได้ทันที และข้อดีอีกประการหนึ่งก็คือ LabVIEW จะมีการตรวจสอบความผิดพลาดของโปรแกรมตลอดเวลา ทำให้โปรแกรมจะทำงานได้ทันทีเมื่อไม่มีข้อผิดพลาดในโปรแกรมเท่านั้น โดยผู้ใช้สามารถที่จะดูรายละเอียดของความผิดพลาดแสดงให้เห็นได้ตลอดเวลาทำให้การเขียนโปรแกรมนั้นง่ายขึ้นมาก ส่วนประกอบภายใน Block Diagram นี้จะประกอบด้วย ฟังก์ชัน ค่าคงที่ โปรแกรมควบคุมการทำงานหรือโครงสร้าง จากนั้นในแต่ละส่วนเหล่านี้ซึ่งจะปรากฏในรูปของ Block เราจะได้รับบริการต่อสาย (Wire) สำหรับ Block ที่เหมาะสมเข้าด้วยกัน เพื่อกำหนดลักษณะการไหลของข้อมูลระหว่าง Block เหล่านี้ ทำให้ข้อมูลได้รับการประมวลผลตามที่ต้องการและแสดงผลออกมาให้แก่ผู้ใช้ต่อไป ดังแสดงในรูปที่ 2.33



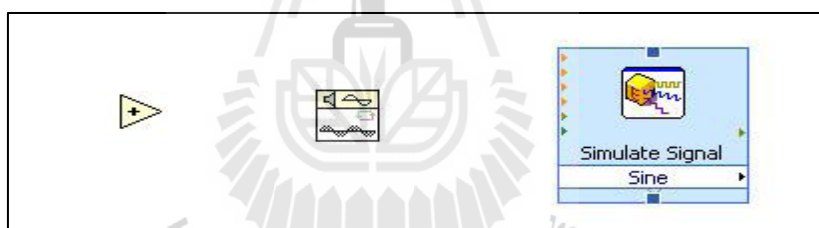
รูปที่ 2.33 แสดงตัวอย่าง Block Diagram

Node คือรูป Icon ที่อยู่บน Block Diagram ซึ่งมี Input และ/หรือ Output และจะทำงานตามหน้าที่เมื่อมีการรันโปรแกรม โคนแบ่งเป็น 3 ชนิดหลัก

1) Function คือ Node ที่มีหน้าที่พื้นฐานของคอมพิวเตอร์ ซึ่งเราไม่สามารถที่จะเจาะเข้าไปดูรายละเอียดภายในได้อีก เช่น การบวก การคูณ

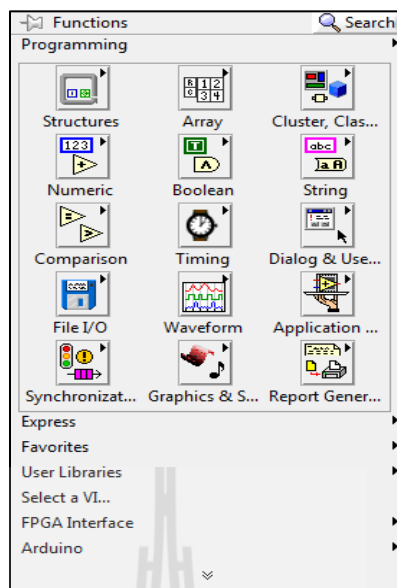
2) SubVIs หรือในภาษาทางซอฟต์แวร์อาจจะเรียกว่า Subroutine หรือ Subprogram คือ โปรแกรมย่อยที่ถูกเขียนขึ้นมาเพื่อถูกนำมาเรียกใช้ในอีกโปรแกรมหนึ่ง เราสามารถเปิดเข้าไปดู Front Panel และ Block Diagram ได้เมื่อ Double Click ที่ Icon ของมัน

3) Express VIs เป็น SubVIs ประเภทพิเศษ คือ เมื่อเราเลือก Express VI มาวางบน Block Diagram มันจะปรากฏหน้าต่าง Configuration ขึ้นมาเพื่อนให้เราเข้าไปป้อนค่า Parameters ต่าง ๆ ตามต้องการและเมื่อเราป้อนค่าเสร็จ มันก็จะสร้างโค้ดไว้ภายในอัตโนมัติตามที่เราได้ตั้งค่าไว้ ซึ่งความสามารถของ Express VI นี้ทำให้เราแทบไม่จำเป็นต้องต่อสาย Input เลยเพราะ Parameter ทั้งหมดได้ถูกสร้างขึ้นมาแล้วถูกเก็บไว้ภายในเรียบร้อยแล้ว จึงทำให้การเขียน LabVIEW ง่ายและเร็วขึ้นมาก สังเกตง่าย ๆ Express VI จะมี Icon ขนาดใหญ่ที่มีพื้นหลังเป็นสีฟ้า



รูปที่ 2.34 แสดงตัวอย่าง Block Diagram Node

เครื่องมือที่ใช้ในการเขียนโปรแกรมบน Block Diagram ซึ่ง LabVIEW ใช้ Functions Palette ซึ่งจะมี Function และ SubVI ต่าง ๆ ที่มีอยู่แล้วให้ผู้ใช้เลือกใช้ โดย Function และ SubVI จัดเป็นกลุ่ม ๆ เช่น Numeric Function จะมี Function ต่าง ๆ เกี่ยวกับตัวเลข เช่น บวก ลบ คูณหาร แสดงดังรูป 2.35



รูปที่ 2.35 แสดงเครื่องมือสำหรับ Dam-Data Acquisition

2.9.1.3 Icon และ Connector เปรียบเสมือน โปรแกรมย่อย Subroutine ในโปรแกรมปกติทั่ว ๆ ไป โดย Icon จะหมายถึง Block Diagram ตัวหนึ่งที่มีการส่งข้อมูลเข้าและออกผ่านทาง Connector ซึ่งใน LabVIEW เราจะเรียก Subroutine นี้ว่า subVI ข้อดีของการเขียนโปรแกรมด้วยภาษา G นี้ก็คือ เราสามารถสร้าง VI ที่ละส่วนขึ้นมาให้ทำงานด้วยตัวเองได้อย่างอิสระ จากนั้นในภายหลังหากเราต้องการเราก็สามารถเขียนโปรแกรมอื่นขึ้นมาเพื่อเรียกใช้งาน VI ที่เราเคยสร้างขึ้นก่อนหน้านี้ทีละตัว ซึ่งทำให้ VI ที่เราเขียนขึ้นก่อนกลายเป็น subVI ไป การเขียนในลักษณะนี้เรียกว่าเขียนเป็น Module

สำหรับลักษณะทั่วไปของ Icon และ Connector จะแสดงในรูปต่อไปนี้ เราจะเห็นว่าเมื่อเราแสดงในรูปของ Connector เราจะพบว่า มีช่องต่อข้อมูลหรือที่เรียกว่า Terminal ปรากฏให้เห็น



รูปที่ 2.36 แสดงลักษณะ Icon และ Connector



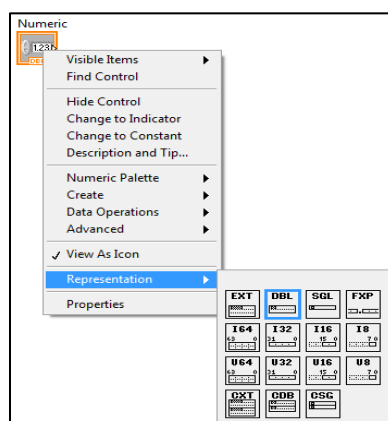
คำศัพท์ต่าง ๆ ที่ใช้กันใน LabVIEW นี้มักจะแตกต่างจากที่ใช้กันในภาษาการเขียนโปรแกรมตัวหนังสือทั่ว ๆ ไปในหลาย ๆ ด้าน ดังนั้นเพื่อให้ผู้ที่เริ่มใช้ LabVIEW เข้าใจถึงศัพท์ต่าง ๆ ที่ใช้ในโปรแกรม เราจึงขอเปรียบเทียบศัพท์ที่ใช้ใน LabVIEW กับโปรแกรมพื้นฐานทั่ว ๆ ไปตามตารางที่ 2.1

ตารางที่ 2.1 เปรียบเทียบศัพท์ที่ใช้ใน LabVIEW

LabVIEW	โปรแกรมพื้นฐาน	หน้าที่
VI	Program	ตัวโปรแกรมหลัก
Function	Function	ฟังก์ชันสำเร็จรูปที่สร้างขึ้นมากับโปรแกรมนั้นเช่น sin, log เป็นต้น
SubVI	Subroutine	โปรแกรมย่อยที่ถูกเรียกใช้โดยโปรแกรมหลัก
Front Panel	user interface	ส่วนที่ติดต่อกับผู้ใช้
Block Diagram	Program code	การเขียนตามขั้นตอนของทีแต่ละโปรแกรมกำหนดขึ้น

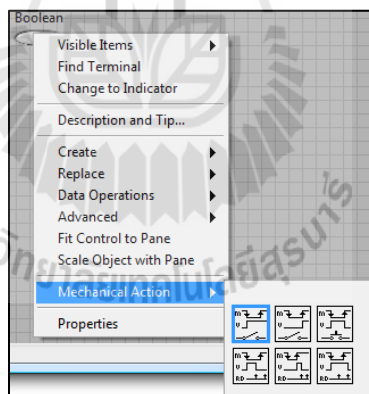
ในการเขียนโปรแกรมทั่ว ๆ ไปจะต้องมีการประกาศตัวแปรก่อนที่จะใช้ตัวแปรนั้น แต่สำหรับโปรแกรม LabVIEW มันจะจัดการให้เองหมดโดยผู้ใช้ไม่จำเป็นต้องทำเอง เพียงแค่เลือกประเภทของข้อมูลที่มาวางบนโค้ดให้ถูกต้องเท่านั้น ประเภทของข้อมูลภายใน LabVIEW ก็มีหลายอย่างที่เหมือนกับโปรแกรมในภาษาอื่น ๆ และยังมีอีกบางประเภทที่ใช้ใน LabVIEW เท่านั้น โปรแกรม LabVIEW แบ่งข้อมูลเป็น 6 ชนิดดังนี้คือ

1) Numeric คือ ข้อมูลประเภทตัวเลข มีทั้งจำนวนเต็มซึ่งใน Block Diagram จะเห็นเป็นสีน้ำเงิน และจำนวนทศนิยมจะเห็นเป็นสีส้ม และสามารถเปลี่ยนไปมาได้โดยการคลิกขวาที่ตัวเลขนั้น แล้วเลือก Representation และเลือกประเภทตัวเลขได้เลย แสดงดังรูป 2.37



รูปที่ 2.37 แสดงข้อมูลประเภท Numeric

2) Boolean คือข้อมูลประเภทที่มีสองค่า คือ True และ false บน Block Diagram จะแสดงข้อมูลเป็นสีเขียว และสำหรับ Front Panel ตัว Boolean จะมีลักษณะเป็น ตัว Control หรือ สวิตช์ ถ้าเป็น Output ก็จะเป็น LED หรือหลอดไฟประเภทต่าง ๆ แสดงดังรูป 2.37



รูปที่ 2.38 แสดงข้อมูลประเภท Boolean

3) String คือ ข้อมูลประเภทที่เป็นตัวอักษร Icon จะแสดงเป็นสีชมพู สำหรับการแสดงผล จะมีอยู่ 4 แบบ คือ

- Normal Display คือ การแสดงปกติ
- Code Display คือ การแสดงแบบ โค้ดมีประโยชน์สำหรับแสดงตัวอักษรที่ตาเปล่ามองไม่เห็น การเว้นวรรค แทบ หรือการขึ้นบรรทัดใหม่
- Password Display คือ การแทนตัวอักษรด้วย \*

- Hex Display แสดงผลเป็นรหัสเลขฐานสิบหก

4) Enum คือ ข้อมูลประเภทแสดงให้ผู้ใช้เห็นเป็นตัวหนังสือ แต่ค่าจริงของมัน คือ ตัวเลข ดังนั้นบน Block Diagram เราจึงมองเห็นข้อมูลประเภทนี้เป็นสีน้ำเงิน ซึ่งเหมือนกับจำนวนเต็ม

5) Dynamic เป็นข้อมูลที่อยู่ในรูปของ Waveform บน Block Diagram ถูกแสดงด้วยสีน้ำเงินเข้มซึ่งภายในจะประกอบด้วย Array ของเวฟฟอร์ม Time Stamp ชื่อของสัญญาณ ข้อมูลประเภท Dynamic นี้ส่วนใหญ่ใช้ใน Express VI จากพวกการอ่านกำเนิด และวิเคราะห์สัญญาณ

6) Time Stamp เป็นข้อมูลที่ประกอบด้วยวันที่และเวลาที่มีความละเอียดถึงมิลลิวินาที Time Stamp บน Block diagram จะมีหน้าต่างที่เป็นสีน้ำตาลเส้นหน้า สามารถนำมาแปลงให้เป็นวันที่ เวลา แบบ String ได้

## 2.9.2 DAQ, GPIB and Serial Communication

ในงานด้านการวัดและเครื่องมือวัดทางวิศวกรรม จะประกอบด้วยขั้นตอนหรือกระบวนการวัดหลัก ๆ ตามลำดับดังนี้

1) Sensor-Transducer ทำหน้าที่เปลี่ยนปรากฏการณ์ทางธรรมชาติหรือปริมาณต่าง ๆ ทางฟิสิกส์ให้เป็นปริมาณทางไฟฟ้าที่สามารถตรวจจับได้ ไม่ว่าจะเป็น กระแส ความต่างศักย์ แรงเคลื่อนไฟฟ้า หรือความต้านทานไฟฟ้า

2) Signal Conditioner ทำหน้าที่ปรับแต่งปริมาณสัญญาณที่ได้จากขั้นที่ 1 ให้มีขนาดปริมาณหรือลักษณะที่เหมาะสม เพราะสัญญาณที่ได้จากขั้นตอนที่ 1 นั้น อาจมีขนาดไม่เหมาะสมหรือมีสัญญาณรบกวนมากเกินไปที่จะนำไปวิเคราะห์ในทันทีได้

3) Data Acquisition ทำหน้าที่ประมวลผลความหมายหรือเปลี่ยนสัญญาณในลักษณะ Analog ให้มาอยู่ในรูปของ Digital Signal เพื่อประโยชน์ในการตีความหมายและใช้ในการควบคุมหน้าที่ของ DAQ Boards อาจจะเป็นการอ่านสัญญาณ Analog (A/D Conversion) การสร้างสัญญาณแอนาล็อก (D/A Conversion) เขียนและอ่านสัญญาณ เพื่อเชื่อมต่อกับ Transducer ในเอกสารบางเล่มอาจเรียกขั้นตอนที่ 2 และ 3 อาจเรียกรวมกันว่า Signal Processing และในบางกรณีที่มีสัญญาณที่ได้มาจากการวัดนั้น หากมีขนาดที่เหมาะสมเราอาจไม่ต้องการขั้นตอนการปรับสภาพสัญญาณหรือ Signal Conditioner ก็เป็นไปได้ การติดต่อระหว่างคอมพิวเตอร์และ Transducer จึงเป็นเรื่องสำคัญสำหรับคอมพิวเตอร์ โดยปกติแล้วสามารถที่จะติดต่อสื่อสารกับอุปกรณ์ภายนอกได้โดยการผ่าน Input/Output Board (I/O Board) ซึ่ง I/O Board นี้จะมีหลายแบบแต่แบบที่สำคัญและสามารถเชื่อมต่อโดยผ่านคำสั่งของ LabVIEW ได้ทันทีจะประกอบด้วยอุปกรณ์ดังต่อไปนี้

4) DAQ Board โดยปกติแล้ว DAQ Board เป็นอุปกรณ์พิเศษ คือไม่ได้เป็นมาตรฐานที่ติดตั้งมากับคอมพิวเตอร์ สำหรับบอร์ดประเภทนี้จะมีผู้ผลิตหลายบริษัท ซึ่งลักษณะการ

ทำงานก็อาจจะแตกต่างกันบ้างตามแต่ผู้ผลิตแต่ละแห่งออกแบบมา ข้อสำคัญที่เราต้องเข้าใจในเบื้องต้นก็คือ ไม่ใช่ว่า DAQ Board ทุกแบบ จะเหมาะสมกับการทำงานร่วมกับ LabVIEW ทั้งหมด

สำหรับ DAQ Board ที่ผลิตโดยผู้ผลิตไม่ใช่ว่าจะไม่สามารถทำงานร่วมกับ LabVIEW ได้ แต่ว่าการที่ DAQ Board ที่ผลิตโดยผู้ผลิตรายอื่นนั้นจะสามารถทำงานร่วมกับ LabVIEW ได้ก็ต่อเมื่อมี Driver ของอุปกรณ์นั้น เพื่อให้ใช้งานร่วมกับ LabVIEW ได้เท่านั้น ดังนั้นหากว่าทางบริษัทผู้ผลิต DAQ Board นั้นไม่มี Driver ที่ใช้งานร่วมกับ LabVIEW มาให้ เราอาจจำเป็นต้องเขียน Driver ขึ้นมาเอง ซึ่งวิธีการนี้ค่อนข้างจะยุ่งยากและเสียเวลามาก

การเลือกใช้ DAQ Board ที่เหมาะสมกับคนที่จะใช้ถือว่าเป็นเรื่องที่สำคัญมากประการหนึ่ง เพราะราคาของเครื่องมือนี้จะเปลี่ยนแปลงไปอย่างมากหากเรากำหนดข้อกำหนดแตกต่างกันออกไป เช่น อัตราการเรียกสุ่มตัวอย่าง (Sampling Rate) จำนวนช่องรับ-ส่งสัญญาณ (I/O Channel) หรืออื่น ๆ โดยปกติราคาของ DAQ Board ที่ผลิตโดยบริษัท National Instrument จะมีราคาอยู่ระหว่าง \$700-2000 ขึ้นอยู่กับความต้องการ ซึ่งเป็นราคาที่นับว่าสูงมาก เราอาจจะได้ Board ที่ราคาถูกลงกว่าจากผู้ผลิตรายอื่น แต่อาจจะมีปัญหาเรื่องการใช้งานร่วมกับ LabVIEW

หน้าที่ของ DAQ Board นี้จะทำการติดต่อโดยตรงกับ Transducer ซึ่งอาจจะผ่านอุปกรณ์ Signal Conditioner หรือไม่ได้ก็ตามความจำเป็น นอกเหนือจากนั้นแล้วเราไม่จำเป็นต้องมีอุปกรณ์อื่นเข้ามาร่วมในการสร้างเครื่องมือวัดเสมือนจริง ซึ่งจุดนี้เป็นจุดได้เปรียบของ DAQ Board นั้นคือแม้ว่าจะมีราคาแพง แต่สามารถจะทดแทนการซื้อเครื่องมือวัดอื่น ๆ ได้มากมาย เราจะกล่าวถึง DAQ Board อย่างละเอียดอีกครั้งหนึ่งในบทที่กล่าวถึง Data Acquisition Hardware

5) GPIB General Purpose Interface Bus เป็นการขนถ่ายข้อมูลระบบใหม่ที่เพิ่งได้รับความนิยม แม้ว่าจะได้รับการพัฒนาโดยบริษัท Hewlett-Packard ตั้งแต่ปลายทศวรรษ 1960 และได้พัฒนาจนกระทั่งได้รับมาตรฐานจาก Institute of Electrical and Electronic Engineer (IEEE) ในปี 1975 ซึ่งต่อมารู้จักกันในชื่อ IEEE 488 Standard

จุดประสงค์แรกของ GPIB คือ ใช้ในการควบคุมเครื่องมือวัดโดยคอมพิวเตอร์ อย่างไรก็ตามจุดประสงค์ก็ได้เปลี่ยนแปลงไปบ้างแล้ว โดยการนำ GPIB มาใช้ควบคุมและติดต่อระหว่างคอมพิวเตอร์ด้วยกันหรือระหว่างคอมพิวเตอร์ กับ Scanner หรือเครื่องมือวัดอื่น ๆ ในระยะ 1-2 ปี ที่ผ่านมาระยะพบว่าเครื่องคอมพิวเตอร์บางรุ่นเริ่มมี GPIB ติดมาเป็นอุปกรณ์มาตรฐานของเครื่องแล้ว

GPIB เป็นการติดต่อแบบใหม่ที่สามารถเชื่อมต่ออุปกรณ์หลายชิ้นเข้ากับ GPIB Port ตัวเดียวได้ โดยสามารถต่ออุปกรณ์ได้สูงถึง 15 ชิ้น โดยใช้ Bus เพียงตัวเดียว ทำให้ประหยัด (และป้องกันเรื่องปวดหัวในการติดตั้งอุปกรณ์) ส่วนข้อดีอื่น ๆ มีดังนี้

- ส่งผ่านข้อมูลด้วยวิธีแบบขนาน ครั้งละ 1 byte (8-bits)

- Hardware จะเป็นผู้จัดการเรื่อง Handshaking, timing และอื่น ๆ
- อัตราการส่งผ่านข้อมูล 800 Kbytes/sec หรือมากกว่า ซึ่งนับว่าเร็วมากเมื่อเทียบกับ Port แบบเก่า

- ใช้คำสั่ง ASCII ในการติดต่อและอื่น ๆ อีกหลายประการ

ข้อดี GPIB Port ซึ่งมีประสิทธิภาพตามที่กล่าวมาแล้วและมีราคาถูกมาก หรืออาจติดตั้งมาพร้อมกับคอมพิวเตอร์แล้ว ทำไมเราจะต้องซื้อ DAQ Card มาใช้อีก สิ่งที่เราต้องเข้าใจในอันดับแรกคือ GPIB Port ไม่สามารถทำให้คอมพิวเตอร์ติดต่อโดยตรงกับ Transducer ได้ GPIB Port จะเป็นการติดต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ที่มี GPIB Port เท่านั้น ซึ่งอุปกรณ์ที่มี GPIB Port ติดตั้งอยู่ก็มักจะเป็นอุปกรณ์ประเภทเครื่องมือวัด เช่น Oscilloscope, Multimeter ซึ่งเป็น Actual Instrument อยู่แล้ว ดังนั้นการส่งผ่านข้อมูลจึงเป็นข้อมูลที่อ่านค่าได้เรียบร้อยแล้ว เพราะบนเครื่องมือเหล่านั้นจะมีระบบ Signal Processing อยู่ในตัวเองเรียบร้อยแล้ว

อย่างไรก็ตามการที่ LabVIEW ติดต่อกับอุปกรณ์อื่นโดยผ่านทาง GPIB Port ได้สามารถทำให้เราสามารถเพิ่มศักยภาพของเครื่องมือจริง และสามารถนำค่าที่ได้จากเครื่องมือไปใช้ควบคุมอุปกรณ์อื่น ๆ ด้วยคอมพิวเตอร์ได้อีกด้วย

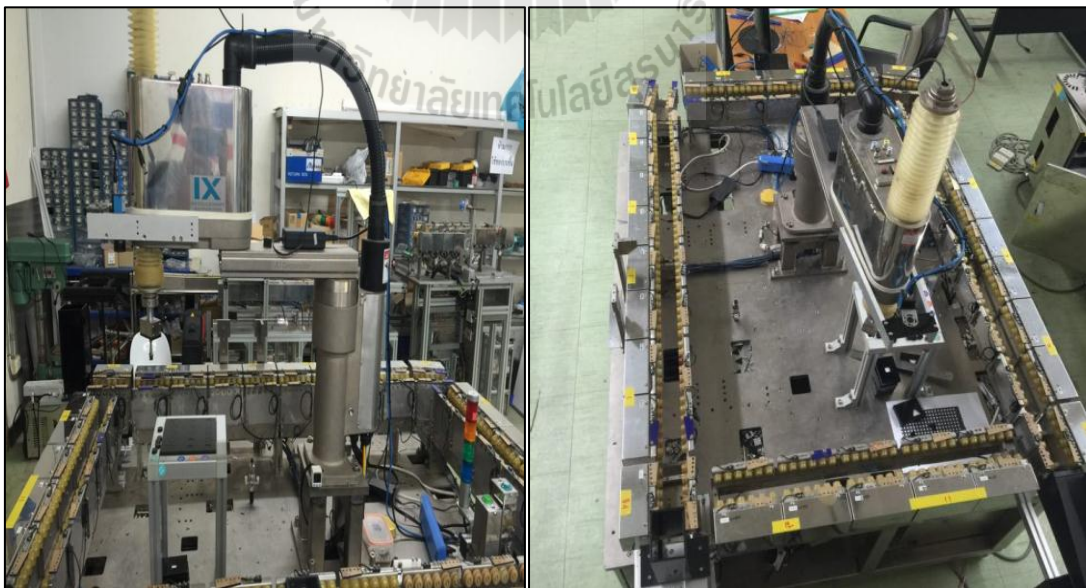
6) Serial Communication ข้อดีของ Serial communication ผ่านทาง Serial Port หรือพอร์ตอนุกรม คือ ราคาถูกและเครื่องมือมากมายได้ใช้อุปกรณ์นี้เป็นมาตรฐาน ลักษณะการทำงานของพอร์ตก็จะคล้ายกับ GPIB Port เพียงแต่ส่งผ่านข้อมูลแบบอนุกรม และมีอัตราการส่งข้อมูลที่ช้ากว่า ซึ่งพอร์ตนี้จะใช้มาตรฐาน RS-232 หรือ RS-485 และการติดต่อข้อมูลจะต้องทำระหว่างคอมพิวเตอร์กับอุปกรณ์ที่มีพอร์ตนี้อยู่ด้วย LabVIEW มี SubVI ที่ใช้ในการติดต่อระหว่างอุปกรณ์ต่าง ๆ กับคอมพิวเตอร์ที่สามารถเรียกมาใช้งานได้ทันทีและการ Set Up เครื่องมือจะมีความยุ่งยากน้อยกว่าอุปกรณ์แบบอื่น ๆ ข้อสำคัญเราต้องมีความเข้าใจกับ Baud Speed, Parity, Stop Bits และอื่น ๆ เป็นต้น อย่างไรก็ตามมีแนวโน้มว่า GPIB Port จะเข้ามาเป็นมาตรฐานและเข้าแทนที่พอร์ตขนานได้ในอีกไม่นานนัก

กล่าวโดยรวมแล้วการทำงานของ LabVIEW จะมีประสิทธิภาพสูงสุดหากว่าเราใช้เพื่อการติดต่อกับเครื่องมือวัดโดยตรง แม้ว่าการเขียนโปรแกรมขั้นพื้นฐานเราจะไม่ได้กล่าวถึงการร่วมใช้กับอุปกรณ์อื่น ๆ ก็ตาม การเขียน VI ในเบื้องต้นผู้ใช้อาจมีความรู้สึว่ายุ่งยาก ต้องจดจำ icon และระบบควบคุมต่าง ๆ มากมายแต่ก็จะเหมือนกับโปรแกรมหลาย ๆ แบบ หลังจากที่เรารู้สึกว่าความคุ้นเคยกับตัวโปรแกรมและการเขียนโปรแกรมพื้นฐานแล้ว เราจะพบว่าการนำ LabVIEW ไปใช้งานไม่ได้มีความยุ่งยากอย่างที่เราคิดเลย

### บทที่ 3

#### วิธีการดำเนินการวิจัย

งานวิจัยนี้จะเป็นการออกแบบและสร้างหุ่นยนต์ที่ใช้ในการหยิบและวางในกระบวนการประกอบ โดยเพิ่มความสามารถในการทำงานของหุ่นยนต์ด้วยการใช้ระบบการมองเห็นของเครื่องจักร ประกอบการทำงานของหุ่นยนต์ ทั้งนี้เนื่องจากการทำงานของหุ่นยนต์แบบหยิบและวางในปัจจุบันจะต้องมีการจัดวางตำแหน่งของส่วนประกอบต่าง ๆ ทุกส่วนให้ถูกต้องเป็นไปตามที่ได้กำหนดไว้ในตอนแรก เพราะการที่ชิ้นส่วนใดมีตำแหน่งที่ผิดพลาดไปเพียงเล็กน้อยก็จะทำให้กระบวนการประกอบนั้นล้มเหลวได้ ทำให้ในกระบวนการผลิตจริงจะต้องมีการหยุดระบบการผลิตเพื่อปรับตำแหน่งของชิ้นส่วนต่าง ๆ อย่างละเอียด ทั้งในการบำรุงรักษาปกติและการหยุดสายการผลิตเป็นเวลานานเพื่อการเปลี่ยนรุ่นผลิตภัณฑ์ที่จะผลิต งานวิจัยนี้คาดว่าจะทำให้การออกแบบกระบวนการผลิตมีความคล่องตัวมากขึ้น เพราะตำแหน่งและแนวการวางตัวของส่วนต่าง ๆ ในสายการผลิต ไม่จำเป็นต้องคงที่ แต่อาจปรับเปลี่ยนไปได้ เพราะการมองเห็นของเครื่องจักรจะสามารถตรวจจับตำแหน่งและแนววางตัวที่เหมาะสมและสั่งการให้หุ่นยนต์หยิบและวางได้อย่างถูกต้องและแม่นยำ ดังรูปที่ 3.1



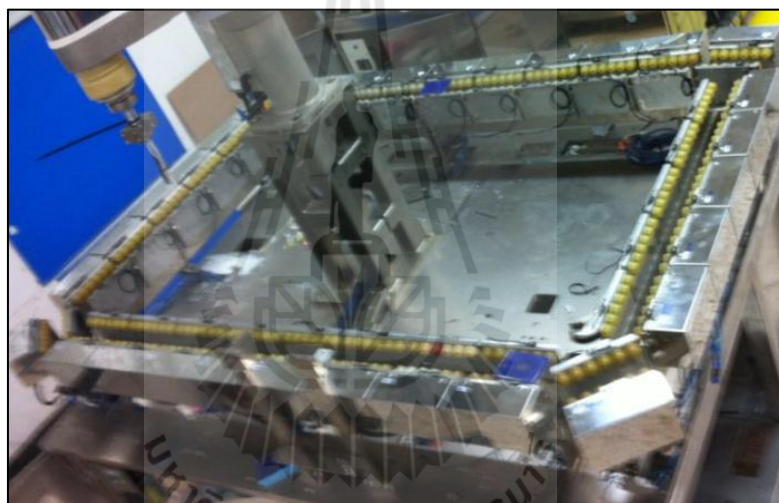
รูปที่ 3.1 แสดงลักษณะทั่วไปของเครื่องที่ได้ออกแบบ

### 3.1 อุปกรณ์ที่สำคัญในการดำเนินงาน

มีอุปกรณ์หลักที่ใช้ในการออกแบบดังต่อไปนี้

#### 3.1.1 การออกแบบชุดลำเลียงชิ้นงานและตัวจับชิ้นงานเพื่อรอการตรวจสอบ

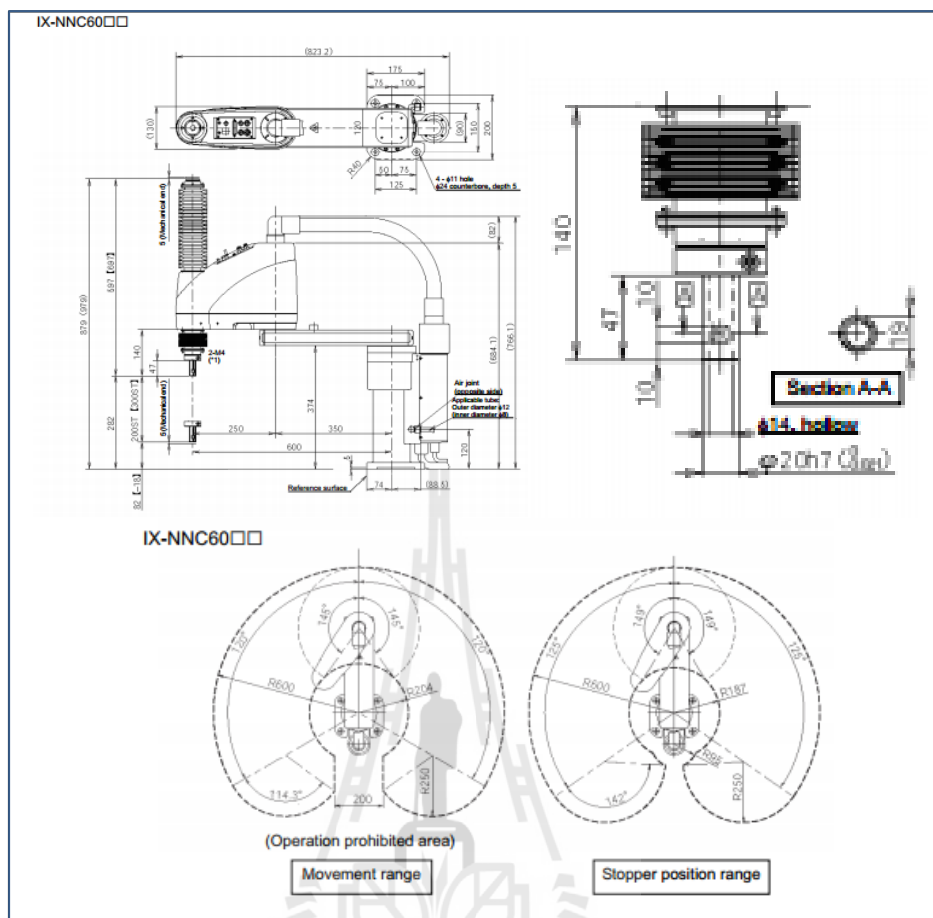
เนื่องจากในงานวิจัยนี้ต้องการการทำงานแบบอัตโนมัติ จึงได้ทำการออกแบบระบบการลำเลียงชิ้นงานให้มีรูปแบบเป็นรูปเกือกม้า โดยด้านบนของเกือกมาให้เป็น Store เพื่อทำการส่งชิ้นงานเข้ามายัง Conveyer เมื่อชิ้นงานไหลมาตามราง Conveyer สักพักก็จะมาถึงส่วนที่เป็นตัวจับชิ้นงานเพื่อรอการตรวจสอบ เมื่อทำการตรวจสอบเสร็จและประกอบเสร็จแล้วจะทำการส่งให้ไหลไปตามราง Conveyer ต่อไปจนถึงส่วนด้านล่างของเกือกมาซึ่งจะเป็นกล่องรับชิ้นงานที่ได้ทำการประกอบแล้ว ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 แสดงรูปแบบระบบชุดลำเลียงชิ้นงาน

#### 3.1.2 ระบบการประกอบชิ้นส่วน

ส่วนของระบบการประกอบชิ้นส่วนนั้น ผู้วิจัยได้นำ Sacara Robot เข้ามาใช้ในการหยิบชิ้นงานมาประกอบ โดยจะใช้ Sacara Robot รุ่น IX-NNC6020 -5L-T1-SP ซึ่งมีลักษณะและพื้นที่ในการทำงานดังรูปที่ 3.3 โดยจะวาง Robot ไว้ยังกึ่งกลางของเครื่องที่ได้ทำการออกแบบดังแสดงในรูปที่ 3.4 เพื่อให้ Robot สามารถเคลื่อนที่ไปหยิบชิ้นส่วนที่ต้องการที่จะประกอบเข้ากับชิ้นงานมาประกอบเข้าด้วยกันได้



รูปที่ 3.3 แสดงรูปแบบของ SCARA ROBOT

### 3.1.3 ระบบการตรวจจับรูปแบบและตำแหน่งการวางตัวของชิ้นงาน

ส่วนของระบบการตรวจจับรูปแบบและตำแหน่งการวางตัวของชิ้นงานนั้น ผู้วิจัยได้ใช้ Camera Image Source DFK23G618 โดยทำการติดตั้งกล้องไปยังปลายของตัว Robot ดังแสดงในรูปที่ 3.2 เพื่อเป็นส่วนที่เก็บภาพส่งสัญญาณภาพไปยังโปรแกรม ส่วนของโปรแกรมที่ใช้ในการวิเคราะห์ผู้วิจัยได้ใช้โปรแกรม NI Vision Builder ส่วนในส่วนของการประมวลผลของภาพแบ่งแยกรูปแบบของชิ้นงานและตำแหน่งการวางตัวของชิ้นงาน ส่วนโปรแกรม LabVIEW ผู้วิจัยได้นำมาใช้ร่วมกับโปรแกรม NI Vision Builder โดยเมื่อมีการประมวลผลรูปแบบและตำแหน่งการวางตัวแล้ว เราจะต้องเขียนโปรแกรมเพื่อให้เกิดการทำงานของกระบวนการประกอบชิ้นงาน การสั่งการในส่วนอื่น ๆ ตัวอย่างเช่น การสั่งให้ Robot เคลื่อนที่ไปหยิบชิ้นส่วนเพื่อนำมาประกอบยังชิ้นงานได้อย่างถูกต้อง





รูปที่ 3.4 แสดงกล้องที่ใช้ในการตรวจสอบ

เมื่อทำการตรวจสอบชิ้นงานโดยโปรแกรมเสร็จแล้ว โปรแกรมจะสั่งให้ Robot เคลื่อนที่ไปจับชิ้นส่วนเพื่อนำมาประกอบ โดยส่วนที่เป็นตัวจับชิ้นงานส่วนนี้ได้ใช้เป็นหัว Vacuum ที่มีหัวขนาดเส้นผ่านศูนย์กลาง 10 mm. เพื่อดูดจับชิ้นส่วนเข้าไปประกอบ



รูปที่ 3.5 แสดงหัว Vacuum

### 3.2 การจัดวางตำแหน่งของอุปกรณ์ต่าง ๆ

การจัดวางตำแหน่งอุปกรณ์ต่าง ๆ มีความสำคัญมาก เพราะตำแหน่งต่าง ๆ ในการการจัดวางมีส่วนทำให้ระบบทำงานได้อย่างถูกต้อง ไม่ว่าจะเป็นการจัดวางตำแหน่งของตัว Robot การจัดวางตำแหน่งของ Store เพราะถ้าเราวางตำแหน่งของ Store ไกลจากตัว Robot มากเกินไปก็จะทำให้ Robot ไม่สามารถเคลื่อนที่ไปจับชิ้นส่วนใน Store ได้ และส่วนของจุดที่จะหยุดชิ้นงาน ตำแหน่งการวางกล้อง ถ้ากล้องวางในตำแหน่งที่ไม่ดีก็จะมีผลต่อการจับภาพ ถ้าการจับยึดไม่ดีก็จะทำให้ชิ้นงานไม่นิ่งทำให้พิกัดต่าง ๆ ผิดไปจากความเป็นจริงก็เป็นได้

#### 3.2.1 ทำการจัดวางระบบลำเลียงชิ้นงานของระบบ

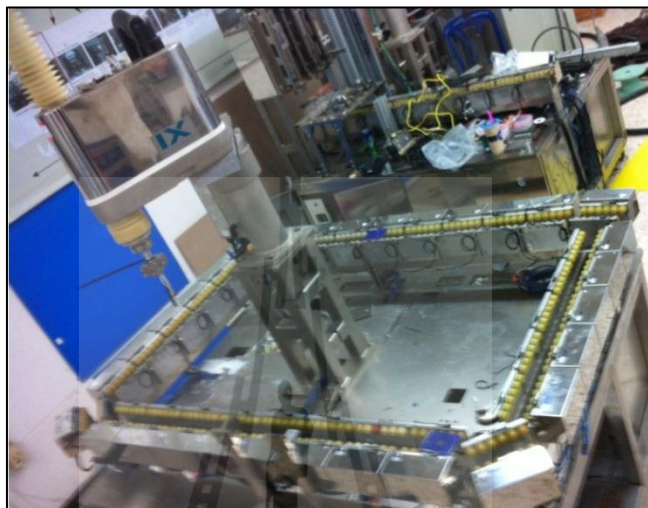
ทำการติดตั้งระบบลำเลียงชิ้นงานของระบบ โดยการจัดรูปแบบให้เป็นรูปสี่เหลี่ยม เพื่อให้ชิ้นงานสามารถไหลวนได้



รูปที่ 3.6 แสดงการจัดวางระบบลำเลียงชิ้นงาน

### 3.2.2 การติดตั้งตำแหน่งของ Scara robot

การกำหนดตำแหน่งของตัว Robot เพื่อการทำงานของตัว Robot ในการจับชิ้นงาน เพื่อนำมาประกอบ ซึ่งได้กำหนดให้ติดตั้งตัว Robot ไว้ที่กึ่งกลางของเครื่อง เพื่อสามารถให้ Robot เคลื่อนที่หยิบและวางชิ้นงานได้อย่างง่าย



รูปที่ 3.7 แสดงรูปร่างของตัว Scara robot และตำแหน่งการวางของ Robot

### 3.2.3 การติดตั้งหัวจับชิ้นงาน

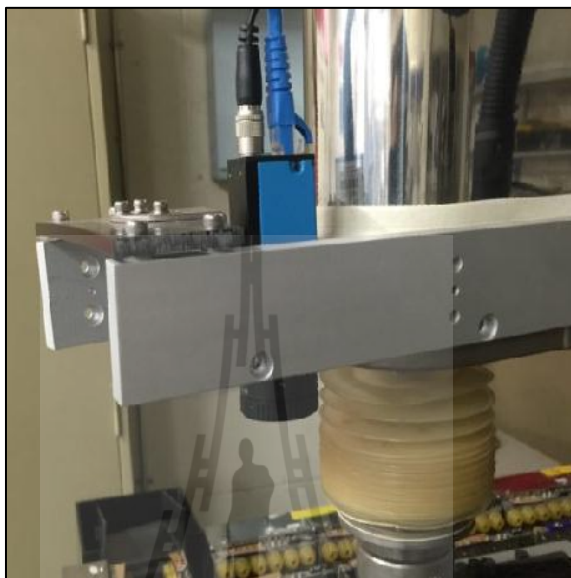
ตัวจับชิ้นงานได้เลือกใช้เป็นตัว Vacuum ที่มีขนาดเส้นผ่านศูนย์กลาง 10 mm. โดยจะนำไปติดตั้งที่ปลายของตัว Robot



รูปที่ 3.8 แสดงการติดตั้งหัว Vacuum

### 3.2.4 การวางตำแหน่งกล้อง CCD Camera

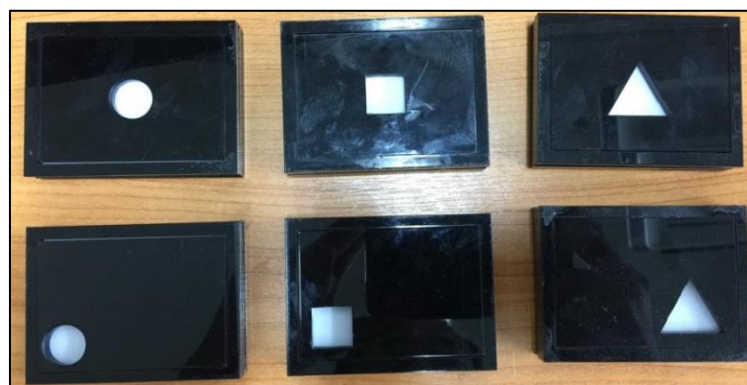
ได้ทำการติดตั้งกล้องไว้ที่ส่วนปลายสุดของ Robot เพื่อที่กล้องจะสามารถทำการจับภาพชิ้นงานได้ทั่วถึงจากข้างบนของชิ้นงาน



รูปที่ 3.9 แสดงรูปตำแหน่งการติดตั้ง Camera

### 3.2.5 การออกแบบรูปแบบของชิ้นงาน

รูปแบบของชิ้นงานได้ทำการออกแบบมา 3 รูปแบบโดยมีรูปแบบของ สามเหลี่ยม สีเหลี่ยม วงกลม ซึ่งจะมีตำแหน่งและแนวการวางตัวที่แตกต่างกัน

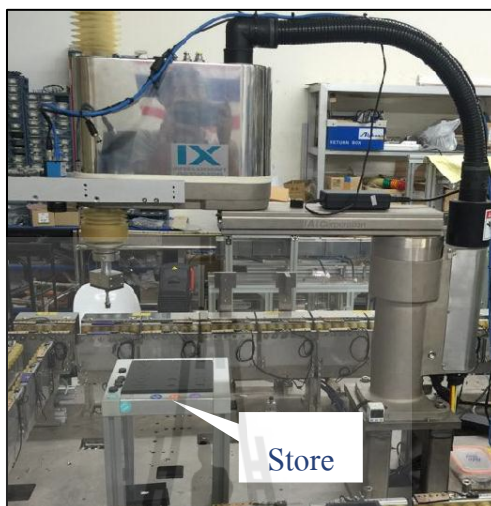


รูปที่ 3.10 แสดงรูปแบบของชิ้นงาน



### 3.2.6 ทำการติดตั้ง Store

การเลือกตำแหน่งการวาง Store เพื่อให้ Robot เคลื่อนที่มาจับชิ้นงานเพื่อนำไปประกอบ โดย Store จะมีส่วนประกอบที่เป็น สามเหลี่ยม สี่เหลี่ยม วงกลม อยู่อย่างละ 5 ชิ้น



รูปที่ 3.11 แสดงตำแหน่งการวาง Store

### 3.2.7 ทำการติดตั้งตัวหยุดชิ้นงาน

การติดตั้งตัวหยุดชิ้นงานเพื่อหยุดชิ้นงานให้อยู่นิ่ง ๆ เพื่อทำการจับภาพ โดยจะมี 2 ส่วน ส่วนแรกจะเป็นตัวยกชิ้นงานขึ้นจากราง Conveyer ส่วนที่สองจะเป็นส่วนของการจับชิ้นงานให้อยู่นิ่ง เพื่อรอการถ่ายภาพ



รูปที่ 3.12 แสดงตัวจับชิ้นงานรอการถ่ายภาพ

### 3.2.8 ทำการติดตั้งชุด Control Scara Robot

ได้ทำการติดตั้งชุดควบคุม Robot ไว้ได้เครื่องเพื่อไม่ให้กีดขวางการทำงาน



รูปที่ 3.13 แสดงการติดตั้งการควบคุม Robot

### 3.2.9 ทำการติดตั้ง PLC KEYNCE KV-700

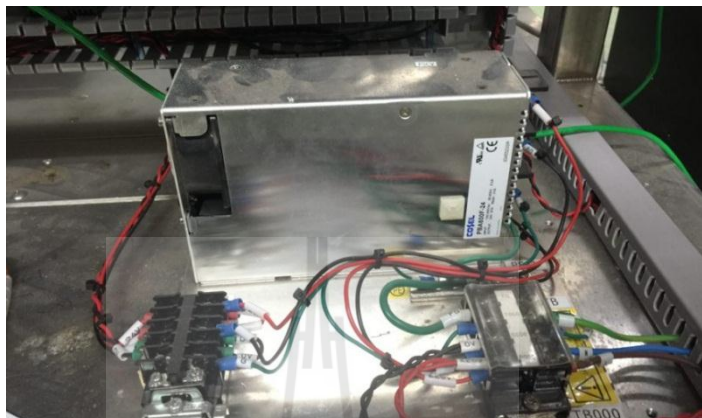
ได้เลือกใช้ PLC KEYNCE KV-700 ซึ่งมี Module Input KV-B16XA และ Module Output KV-B16TA เพื่อใช้เป็นตัวสั่งการให้ Conveyor ทำงานและยังควบคุมอีกหลายส่วน



รูปที่ 3.14 แสดง PLC

### 3.2.10 ทำการติดตั้ง Power Supply

ทำการติดตั้ง Supply เพื่อแปลงไฟจาก 220V เป็น 24V เพื่อจ่ายไป 24V ให้กับอุปกรณ์



รูปที่ 3.15 แสดงการติดตั้ง Power Supply

### 3.2.11 ทำการติดตั้ง Main Regulator

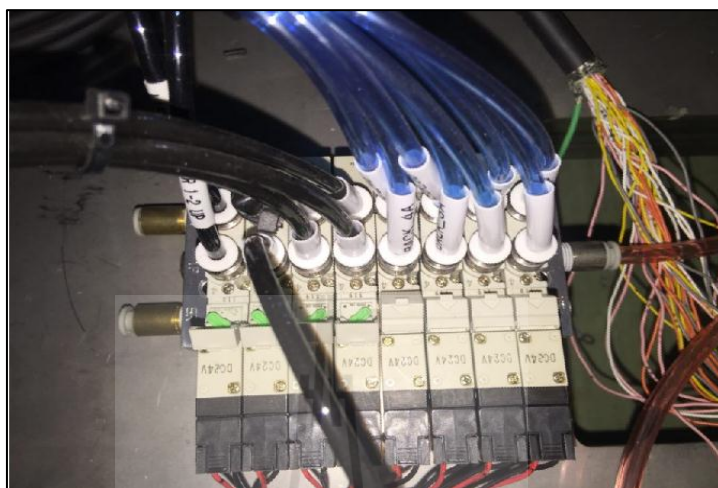
เพื่อควบคุมลมของเครื่องให้ไหลเข้ามามีความสม่ำเสมอ และมีความละเอียดเพิ่มขึ้น



รูปที่ 3.16 แสดงการติดตั้ง Main Regulator

### 3.2.12 ทำการติดตั้งชุด Compact valve ของระบบนิวแมติก

โดยใช้วาล์วของ CKD เพื่อควบคุมการทำงานของกระบอกลูกสูบนิวแมติก



รูปที่ 3.17 แสดงการติดตั้งชุด Compact Valve

### 3.2.13 ทำการติดตั้งไฟแจ้งสถานะ (Status Lamp)

เพื่อบอกสถานะการทำงานของเครื่อง



รูปที่ 3.18 แสดงทำการติดตั้งไฟแจ้งสถานะ



### 3.2.14 ทำการติดตั้งสวิตช์ควบคุมการทำงาน

ซึ่งเป็นสวิตช์แบบปุ่มกด (Pushbutton Switch) เพื่อควบคุมการทำงานของเครื่องจักร



รูปที่ 3.19 แสดงการติดตั้งสวิตช์ควบคุมการทำงาน

## 3.3 ทำการเขียนโปรแกรม PLC สำหรับควบคุมการทำงานของ Conveyer

การเขียนโปรแกรมควบคุม PLC จะใช้โปรแกรม Program KV-STUDIO เพื่อเขียนชุดคำสั่งลงไป PLC โดยอันดับแรกจะต้องทำการเชื่อมต่อระหว่างคอมพิวเตอร์กับ PLC โดยจะเชื่อมต่อผ่านทาง RS-232

### 3.3.1 การกำหนด Address Input / Output

ก่อนทำการเขียน โปรแกรมการทำงานของระบบ เราต้องทำการกำหนด Address Input/Output เพื่อกำหนดว่า Input/Output สิ่งการทำงานของชิ้นส่วนใด ดังแสดงในตารางที่ 3.1

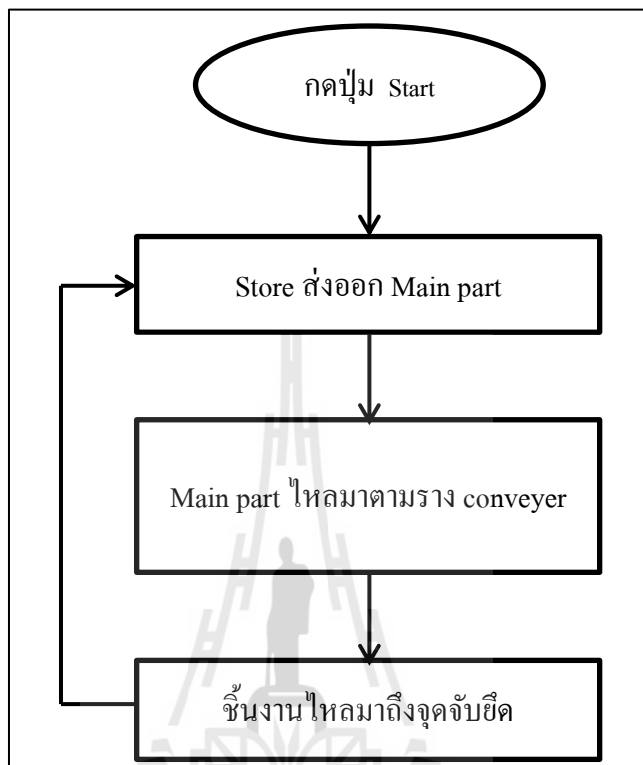
ตารางที่ 3.1 Input Output PLC

INPUT		OUTPUT	
Conveyer_1	1000	Moter_Run_1	5000
		Moter_Stop_1	5001
Conveyer_2	1001	Moter_Run_2	5002
		Moter_Stop_2	5003
Conveyer_3	1002	Moter_Run_3	5004
		Moter_Stop_3	5005
Conveyer_4	1003	Moter_Run_4	5006
		Moter_Stop_4	5007
Conveyer_5	1004	Moter_Run_5	5008
		Moter_Stop_5	5009
Conveyer_6	1005	Moter_Run_6	5010
		Moter_Stop_6	5011
Conveyer_7	1006	Moter_Run_7	5012
		Moter_Stop_7	5013
Conveyer_8	1007	Moter_Run_8	5014
		Moter_Stop_8	5015
Conveyer_9	1008	Moter_Run_9	6000
		Moter_Stop_9	6001
Conveyer_10	1009	Moter_Run_10	6002
		Moter_Stop_10	6003
Conveyer_11	1010	Moter_Run_11	6004
		Moter_Stop_11	6005
Conveyer_12	1011	Moter_Run_12	6006
		Moter_Stop_12	6007
Conveyer_B1	10013	Moter_Run_B1	6008
		Moter_Stop_B1	6009
Conveyer_B2	1014	Moter_Run_B2	6010
		Moter_Stop_B2	6011

ตารางที่ 3.1 Input Output PLC (ต่อ)

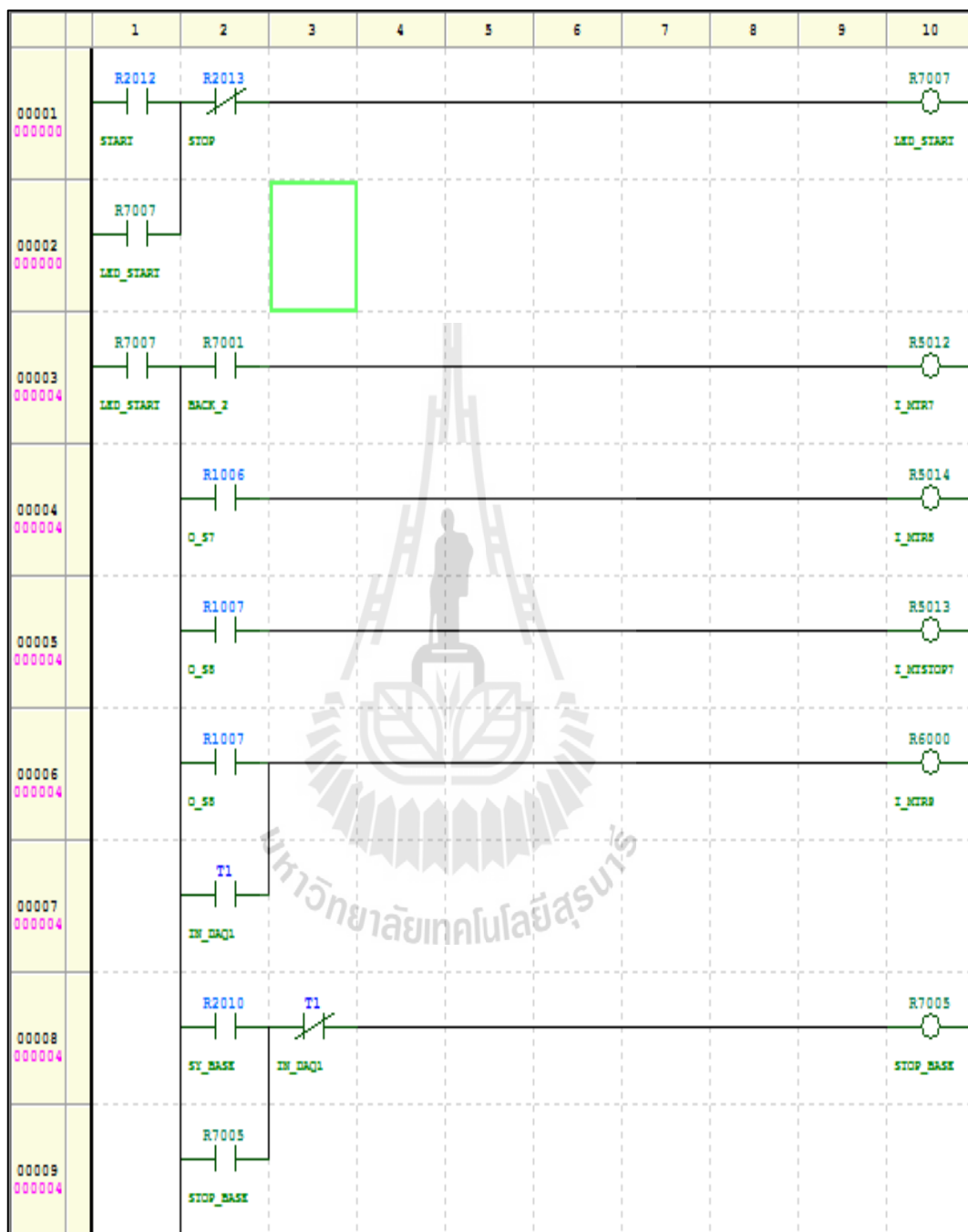
INPUT		OUTPUT	
Conveyer_B3	1015	Moter_Run_B3	6012
		Moter_Stop_B3	6013
Conveyer_B4	1012	Moter_Run_B4	6014
		Moter_Stop_B4	6015
Lead switch B1A	2000	Soloid_valve_B1	7000
Lead switch B1B	2001	Soloid_valve_B2	7001
Lead switch B2A	2002	Soloid_valve_B3	7002
Lead switch B1B	2003	Soloid_valve_B4	7003
Lead switch B3A	2004	Jig	7004
Lead switch B3B	2005	Stop work	7005
Lead switch B4A	2006	Vacuum	7006
Lead switch B4B	2007	Led Start	7007
Lead switch up	2008	Led Stop	7008
Lead switch down	2009	Image Processing	7009
Lead switch jig up	2010	Solenoid valve Store	7010
Lead switch jig down	2011	Solenoid valve Jig	7011
Switch start	2013	Led Red	7012
Switch stop	2014	Led Orange	7013
Final robot		Led Green	7014
On Off Vacuum	2015	Led Blue	7015
Lead switch jig in	3000		
Lead switch jig out	3001		

### 3.3.2 ระบบการทำงานของชุดลำเลียงชิ้นงาน

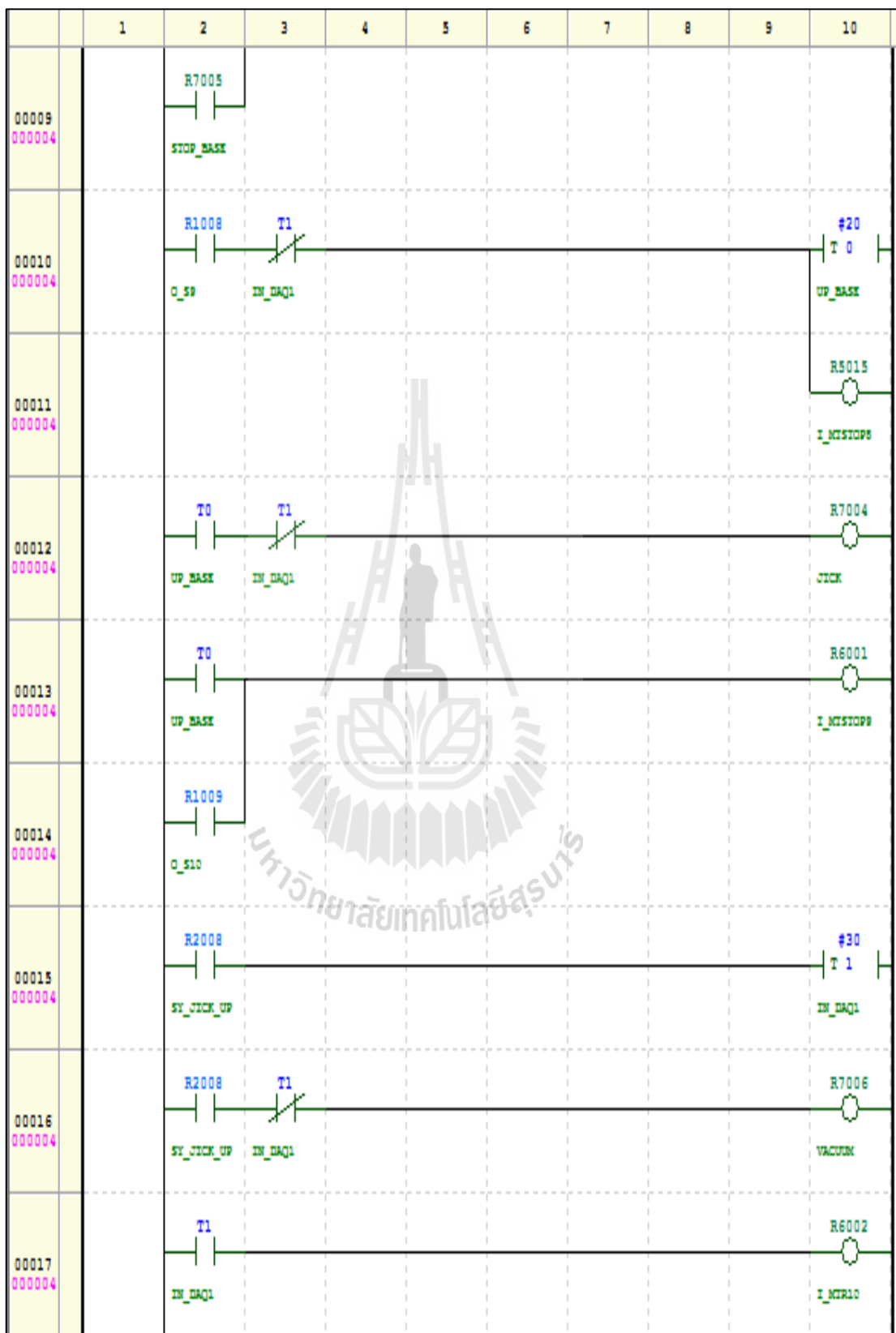


รูปที่ 3.20 แสดงระบบการทำงานของชุดลำเลียงชิ้นงาน

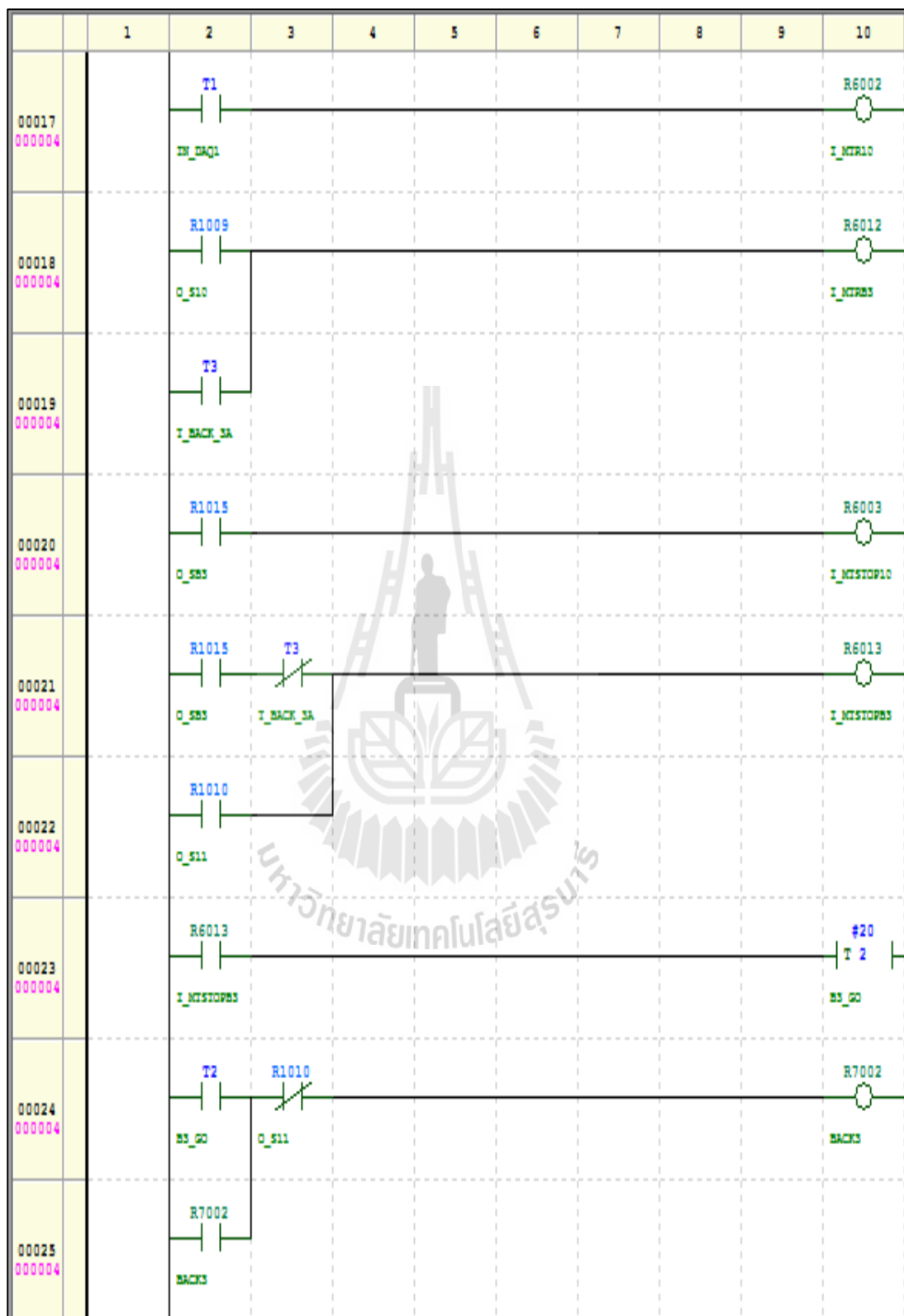
### 3.3.3 โปรแกรมที่ทำการออกแบบการทำงานของระบบลำเลียงชิ้นงาน



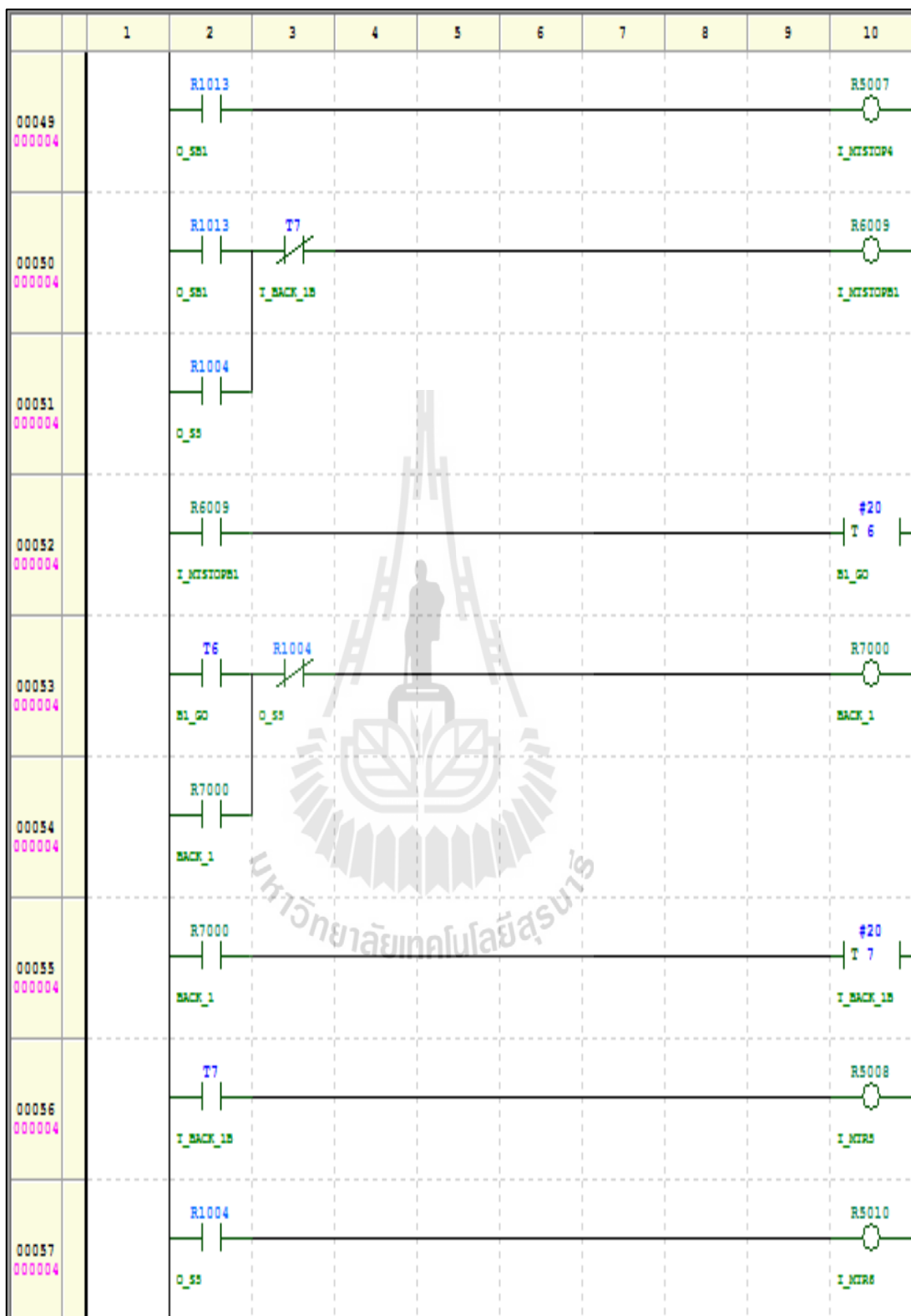
รูปที่ 3.21 แสดง Diagram การทำงานระบบลำเลียงชิ้นงาน



รูปที่ 3.22 แสดง Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ)

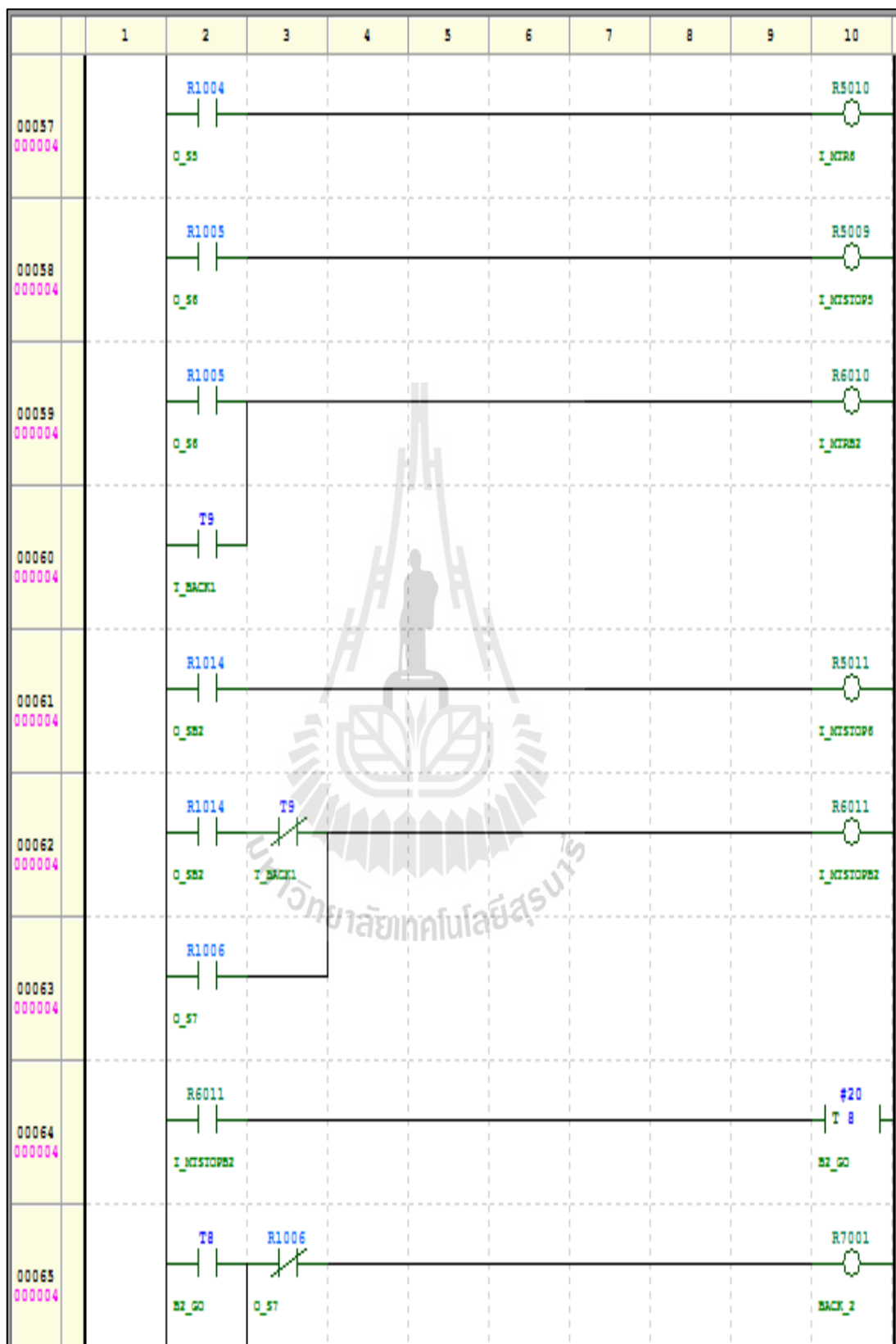


รูปที่ 3.23 แสดง Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ)

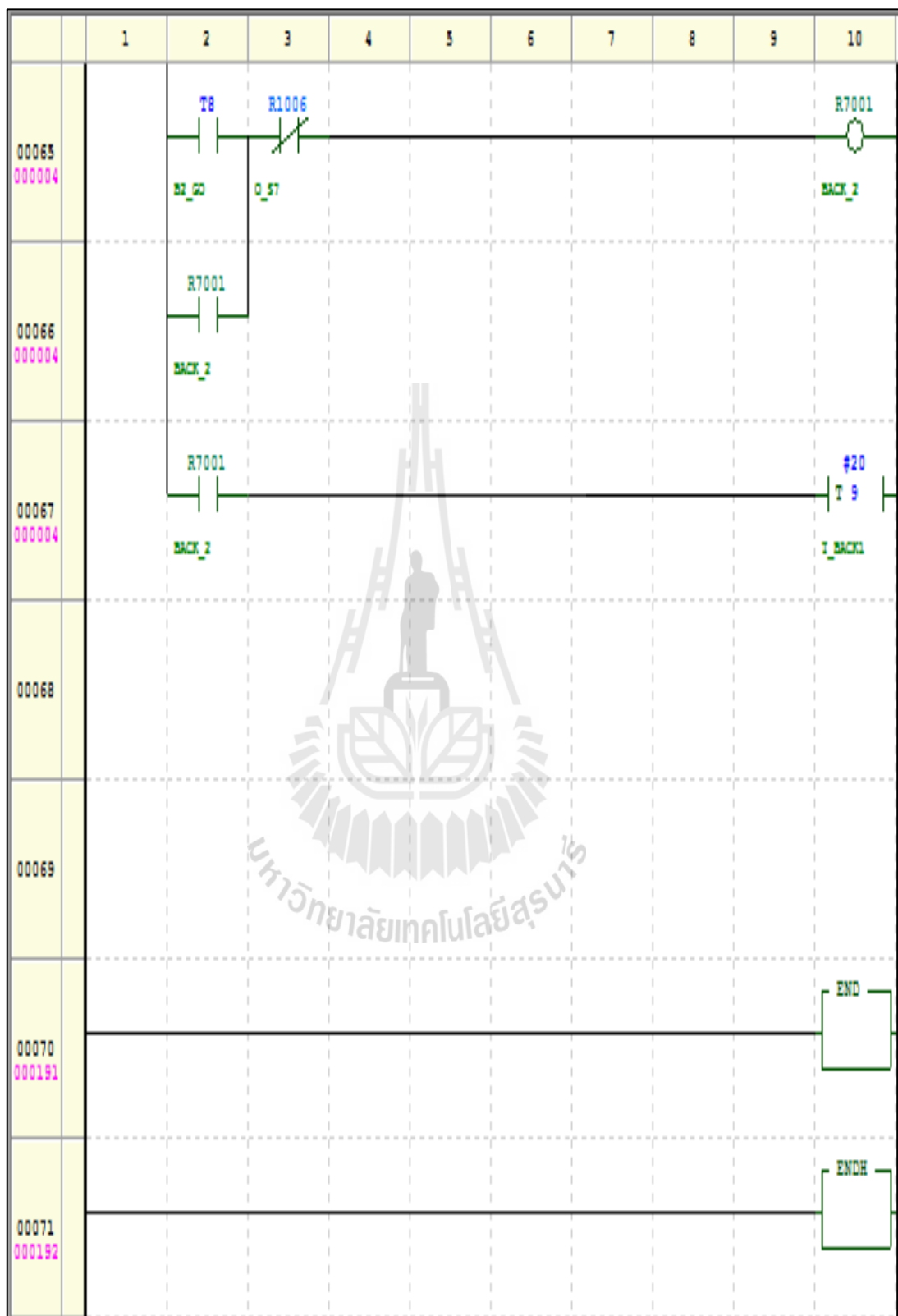


รูปที่ 3.24 แสดง Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ)





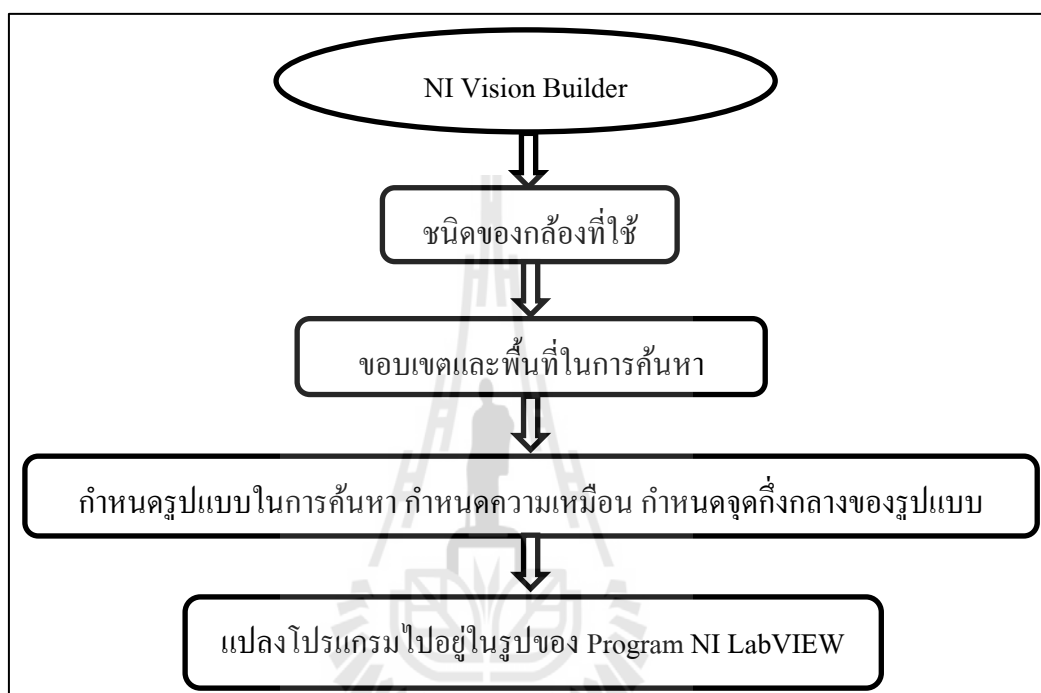
รูปที่ 3.25 แสดง Diagram การทำงานระบบลำเลียงชิ้นงาน(ต่อ)



รูปที่ 3.26 แสดง Diagram การทำงานระบบลำเลียงชิ้นงาน (ต่อ)

### 3.4 ทำการเขียนโปรแกรม NI Vision Builder

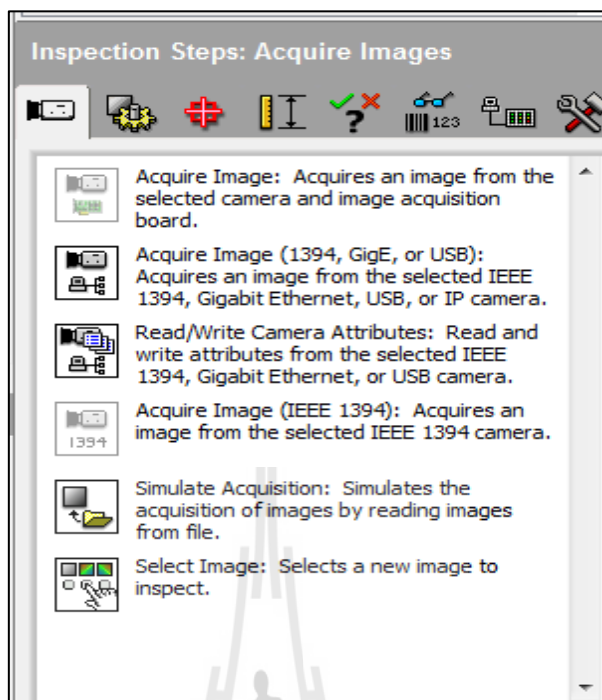
เมื่อทำการเขียนฮาร์ดแวร์เพื่อทำการควบคุมระบบการลำเลียงชิ้นงานเสร็จแล้ว จากนั้นได้ทำการกำหนดรูปแบบของชิ้นงาน กำหนดขอบเขตของการค้นหา กำหนดค่าความเหมือนของการค้นหา โดยใช้โปรแกรม NI Vision Builder



รูปที่ 3.27 แสดงขั้นตอนการกำหนดรูปแบบของ Program NI Vision Builder

#### 3.4.1 การเลือกชนิดของกล้องในการตรวจสอบ

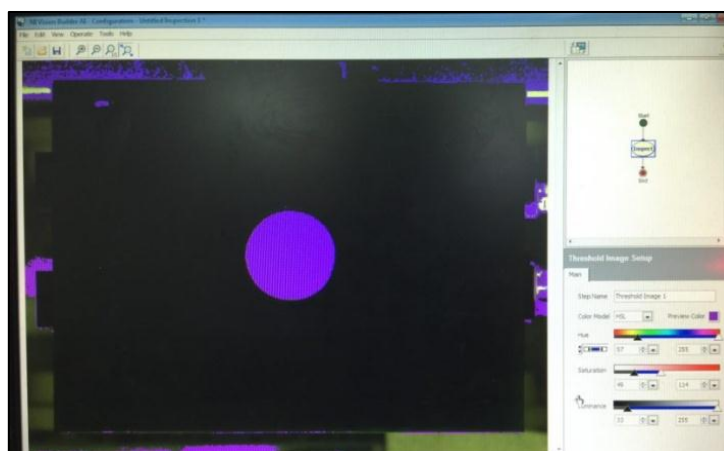
เราต้องมีการกำหนดชนิดของกล้องที่ใช้ในการทดสอบ ซึ่งจะมีหลายแบบหลายชนิด แต่ในงานวิจัยนี้ได้ใช้เป็นแบบ GigE เพราะมีความเร็วในการส่งสัญญาณ สะดวกในการเชื่อมต่อกับคอมพิวเตอร์ ไม่ต้องตั้งค่ากล้องใหม่เมื่อเกิดการ Error หรือ สัญญาณขาดหาย



รูปที่ 3.28 แสดงการเลือกชนิดของกล้องที่ใช้

### 3.4.2 ทำการทดสอบกล้องและปรับความคมชัด

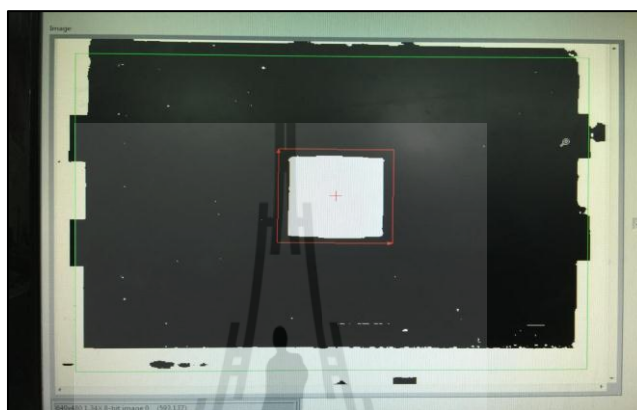
เมื่อได้เลือกชนิดของกล้องแล้วเราต้องทำการทดสอบการเชื่อมต่อระหว่างกล้องและโปรแกรม เมื่อเชื่อมต่อได้แล้วเราจะต้องทำการปรับความคมชัดของภาพที่ได้ให้มีความชัดเจน ดังแสดงในรูปที่ 3.29



รูปที่ 3.29 แสดงการทดสอบการเชื่อมต่อและการปรับความคมชัดของภาพ

### 3.4.3 ทำการปรับสีในการค้นหา

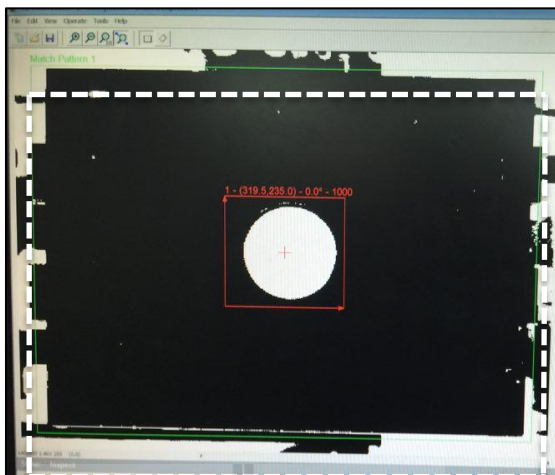
จากสีที่ได้จากการถ่ายภาพเราจะได้ภาพที่ได้มาเป็นสีหลายสี เราจึงทำการปรับสีในการค้นหาในงานวิจัยนี้ได้ใช้เป็นสีขาวและสีดำในการค้นหา โดยจะกำหนดให้วัตถุในการค้นหาเป็นสีขาว และสีที่ไม่ต้องการให้มีสีดำ เพื่อให้แยกแยะส่วนที่เราต้องการค้นหาและส่วนที่เราไม่ต้องการให้มีความแตกต่างกันอย่างเห็นได้ชัดเจน ดังแสดงในรูปที่ 3.30



รูปที่ 3.30 แสดงการปรับสีเพื่อใช้ในการค้นหา

### 3.4.4 ทำการกำหนดขอบเขตในการค้นหา

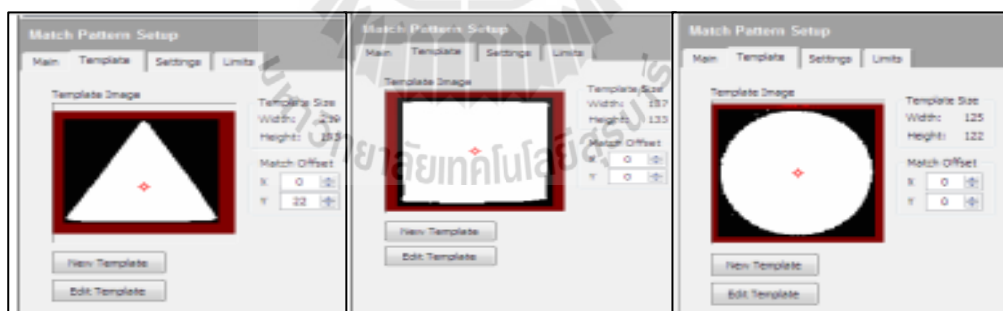
โดยหลักการกำหนดขอบเขตในการค้นหานั้น โปรแกรมจะทำการค้นหาแค่ในส่วนที่เรากำหนด เราจึงต้องทำการกำหนดส่วนที่เราต้องการค้นหา เพื่อกำหนดพื้นที่ให้อยู่แค่ภายในขอบเขตที่กำหนดไว้ ส่วนนอกขอบเขตที่กำหนดไว้ถ้ารูปแบบของเราไปอยู่ในส่วนนั้นก็จะไม่มีการค้นหาหรือค้นหาไม่พบ ดังแสดงในรูปที่ 3.31



รูปที่ 3.31 แสดงการกำหนดขอบเขตในการค้นหา

### 3.4.5 การกำหนดรูปแบบในการค้นหา

การกำหนดรูปแบบในการค้นหาเพื่อให้ง่ายต่อการค้นหา เราจะกำหนดได้หลายรูปแบบแล้วแต่เรากำหนด แต่ในงานวิจัยนี้ได้กำหนด 3 แบบ คือ วงกลม สี่เหลี่ยม สามเหลี่ยม เรายังต้องกำหนดความเหมือนของรูปแบบที่เรากำหนดได้อีกด้วย ดังแสดงในรูปที่ 3.32



รูปที่ 3.32 แสดงการกำหนดรูปแบบในการค้นหา

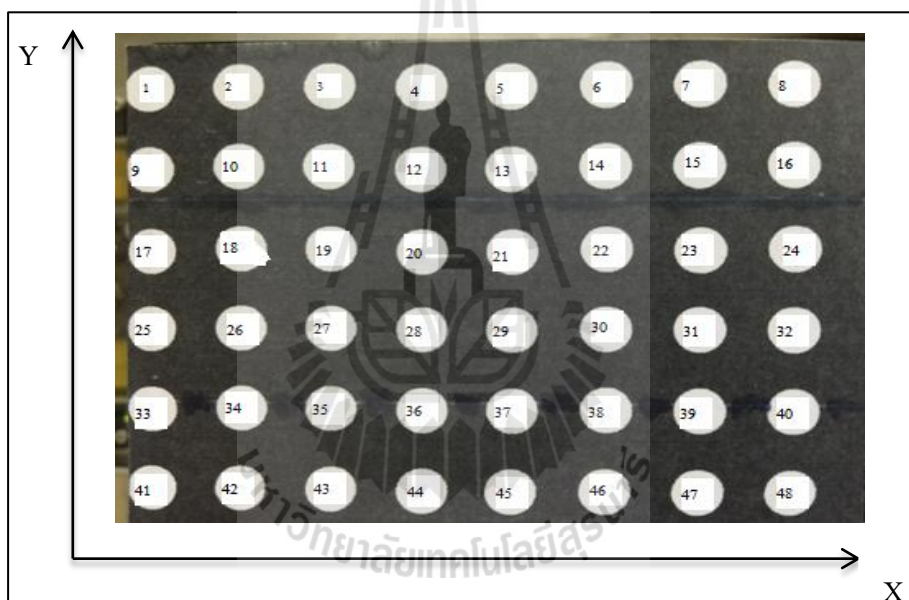
## 3.5 Image Calibration

จากการทดสอบการกำหนดตำแหน่งของวัตถุด้วยภาพที่ถ่ายมาได้ เราพบว่าเมื่อกำหนด Coordinate ให้กับ Robot ของเราแล้ว ตำแหน่งที่ Robot ไปนั้น ไม่ตรงกับตำแหน่งของวัตถุจริง จากการวิเคราะห์พบว่าปัญหาน่าจะเกิดจากรูปถ่ายที่ได้จากกล้อง เนื่องจากกล้องไม่ได้วางที่จุดกึ่งกลาง

ภาพและมีแนวตั้งฉากกับภาพ ทำให้รูปที่ได้เกิด distortion และเมื่อคำนวณหาตำแหน่งจะเกิดการผิดพลาดขึ้นได้ ดังนั้นเราจึงทำการ calibrate ภาพถ่ายที่ได้จากกล้องเสียก่อน

### 3.5.1 Calibration Object

วัตถุที่เราใช้ในการ calibration จะเป็นวัตถุขนาดเท่าชิ้นงานที่ใช้ และได้สร้างวงกลมขึ้นจำนวน 6x8 แถว ซึ่งเราได้กำหนดตำแหน่งที่แน่นอนของวงกลมทั้ง 48 วงนี้ไว้ โดยวงกลมมีเส้นผ่านศูนย์กลาง 10 mm และจุดศูนย์กลางของแต่ละวงจะอยู่ห่างกันเป็นระยะ 10 mm โดยการกำหนดหมายเลขและตำแหน่งของวงกลมจะเป็นไปตามรูป ซึ่งตำแหน่งของจุดศูนย์กลางวงกลมหมายเลข 1 จะเป็น (5, 10) mm หมายเลข 2 จะเป็น (15, 10) mm หมายเลข 10 จะเป็น (15, 20) เป็นต้น ดังแสดงในรูปที่ 3.33



รูปที่ 3.33 แสดงรูปการ Calibration Object

จากนั้นเราจะนำวัตถุนี้เข้าไปในเครื่อง เพื่อให้ระบบตรวจจับตำแหน่ง ทำการกำหนดตำแหน่งของจุดศูนย์กลางวงกลมซึ่งจะเป็นระยะพิกัดที่วัดจากเครื่องหรือ  $(x_m, y_m)$  จากนั้นนำมาเปรียบเทียบกับตำแหน่งตามความเป็นจริงทางกายภาพหรือ  $(x_r, y_r)$  ระยะพิกัดที่วัดจากเครื่อง  $(x_m, y_m)$

### 3.5.2 พิกัดจริงจากรูปแบบที่ได้ออกแบบ

พิกัดจริงของจุดศูนย์กลางของวงกลมวงต่าง ๆ ที่เราได้ทำการออกแบบ เพื่อทำการ Calibration โดยเราจะกำหนดให้เป็น  $(x_m, y_m)$  ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 ระยะพิกัดจริงที่ได้ทำการออกแบบ ( $x_m, y_m$ )

ตำแหน่งจริงของวงกลมบนชิ้นงาน ที่เรานำมา Calibration (mm.)			ตำแหน่งจริงของวงกลมบนชิ้นงาน ที่เรานำมา Calibration (mm.)		
วงกลมที่	X	Y	วงกลมที่	X	Y
1	5	10	25	5	40
2	15	10	26	15	40
3	25	10	27	25	40
4	35	10	28	35	40
5	45	10	29	45	40
6	55	10	30	55	40
7	65	10	31	65	40
8	75	10	32	75	40
9	5	20	33	5	50
10	15	20	34	15	50
11	25	20	35	25	50
12	35	20	36	35	50
13	45	20	37	45	50
14	55	20	38	55	50
15	65	20	39	65	50
16	75	20	40	75	50
17	5	30	41	5	60
18	15	30	42	15	60
19	25	30	43	25	60
20	35	30	44	35	60
21	45	30	45	45	60
22	55	30	46	55	60
23	65	30	47	65	60
24	75	30	48	75	60



จากตารางแสดงพิกัดค่าที่เราได้กำหนดความห่างของจุดศูนย์กลางของวงกลมแต่ละวงตามที่เรากำหนดไว้ เพื่อนำค่าที่ได้ไปเปรียบเทียบกับค่าที่ได้จากการถ่ายภาพ

### 3.5.3 ตำแหน่งจริงที่ได้จากการประมวลผลของกล้อง

ตำแหน่งตามความเป็นจริงทางกายภาพหรือ พิกัดที่ได้จากการถ่ายภาพชิ้นงานที่เราจะนำมา Calibration โดยเราจะกำหนดพิกัดที่ได้จากการถ่ายภาพเป็น  $(x_r, y_r)$  ดังแสดงในตารางที่ 3.3

ตารางที่ 3.3 ตำแหน่งตามความเป็นจริงที่ได้จากการถ่ายภาพหรือ  $(x_r, y_r)$

ตำแหน่งที่ได้จากการถ่ายภาพ (mm.)			ตำแหน่งที่ได้จากการถ่ายภาพ (mm.)		
วงที่	X	Y	วงที่	X	Y
1	4.791788	9.33270705	25	4.731289	39.4035365
2	14.80573	9.28578822	26	14.80848	39.2724397
3	24.81005	9.30924764	27	24.69593	39.3234985
4	34.85012	9.55212158	28	34.71262	39.4973741
5	44.50794	9.61008013	29	44.56157	39.5401531
6	54.68826	9.48588323	30	54.77626	39.4131962
7	64.63896	9.57144109	31	64.79158	39.4573551
8	74.5814	9.63353954	32	74.73952	39.4877144
9	4.716165	19.4340549	33	4.696915	49.5683628
10	14.75348	19.1304625	34	14.78648	49.3724077
11	24.61755	19.1691015	35	24.68218	49.3834474
12	34.63975	19.5679116	36	34.71125	49.6221814
13	44.49969	19.6244902	37	44.56432	49.6415009
14	54.66626	19.3333174	38	54.78038	49.4634854
15	64.61558	19.352637	39	64.80395	49.4993645
16	74.55115	19.4285351	40	74.7519	49.5435234
17	4.860537	29.7175582	41	4.470044	59.5372339
18	14.86898	29.3725669	42	14.55824	59.5275742
19	24.75643	29.4277655	43	24.64505	59.5386139
20	34.84049	29.767237	44	34.64937	59.6904101
21	44.62482	29.8265755	45	44.3402	59.7249093
22	54.83401	29.5367828	46	54.58926	59.574493
23	64.8452	29.7520574	47	64.59358	59.5910526
24	74.75602	29.6154408	48	74.54015	59.5924326

จากค่าในตารางที่ได้เป็นค่าที่ได้จากการถ่ายภาพ เราจะนำค่าที่ได้เปรียบเทียบกับค่าจริงที่ได้ได้ออกแบบไว้เพื่อนำหาค่า Error

### 3.5.4 การหาค่า Error ที่จุดต่าง ๆ

จากนั้นนำตำแหน่งจริงและตำแหน่งที่ภาพคำนวณได้มาหาค่าความผิดพลาดที่เกิดขึ้น ตามสมการ  $\varepsilon_x = \frac{x_r - x_m}{x_r} \times 100$  และทำนองเดียวกันกับค่าความผิดพลาดตามแกน y

ดังแสดงในตารางที่ 3.4

ตารางที่ 3.4 ตาราง Error ของจุดต่าง ๆ ตามแกน X,Y

Error ใน X, Y (%)			Error ใน X, Y (%)		
วงกลมที่	X	Y	วงกลมที่	X	Y
1	4.164234	6.672929	20	0.455729	0.775877
2	1.295119	7.142118	21	0.833738	0.578082
3	0.759795	6.907524	22	0.301804	1.544057
4	0.42823	4.478784	23	0.238151	0.826475
5	1.093456	3.899199	24	0.325303	1.281864
6	0.566799	5.141168	25	5.374212	1.491159
7	0.555453	4.285589	26	1.276786	1.818901
8	0.558132	3.664605	27	1.216287	1.691254
9	5.676706	2.829725	28	0.82108	1.256565
10	1.643446	4.347688	29	0.974291	1.149617
11	1.529781	4.154492	30	0.406802	1.467009
12	1.02929	2.160442	31	0.320649	1.356612
13	1.111789	1.877549	32	0.347303	1.280714
14	0.606798	3.333413	33	6.061699	0.863274
15	0.591414	3.236815	34	1.42345	1.255185
16	0.598465	2.857325	35	1.271286	1.233105
17	2.789259	0.941473	36	0.825008	0.755637
18	0.87346	2.091444	37	0.96818	0.716998
19	0.974291	1.907448	38	0.399302	1.073029

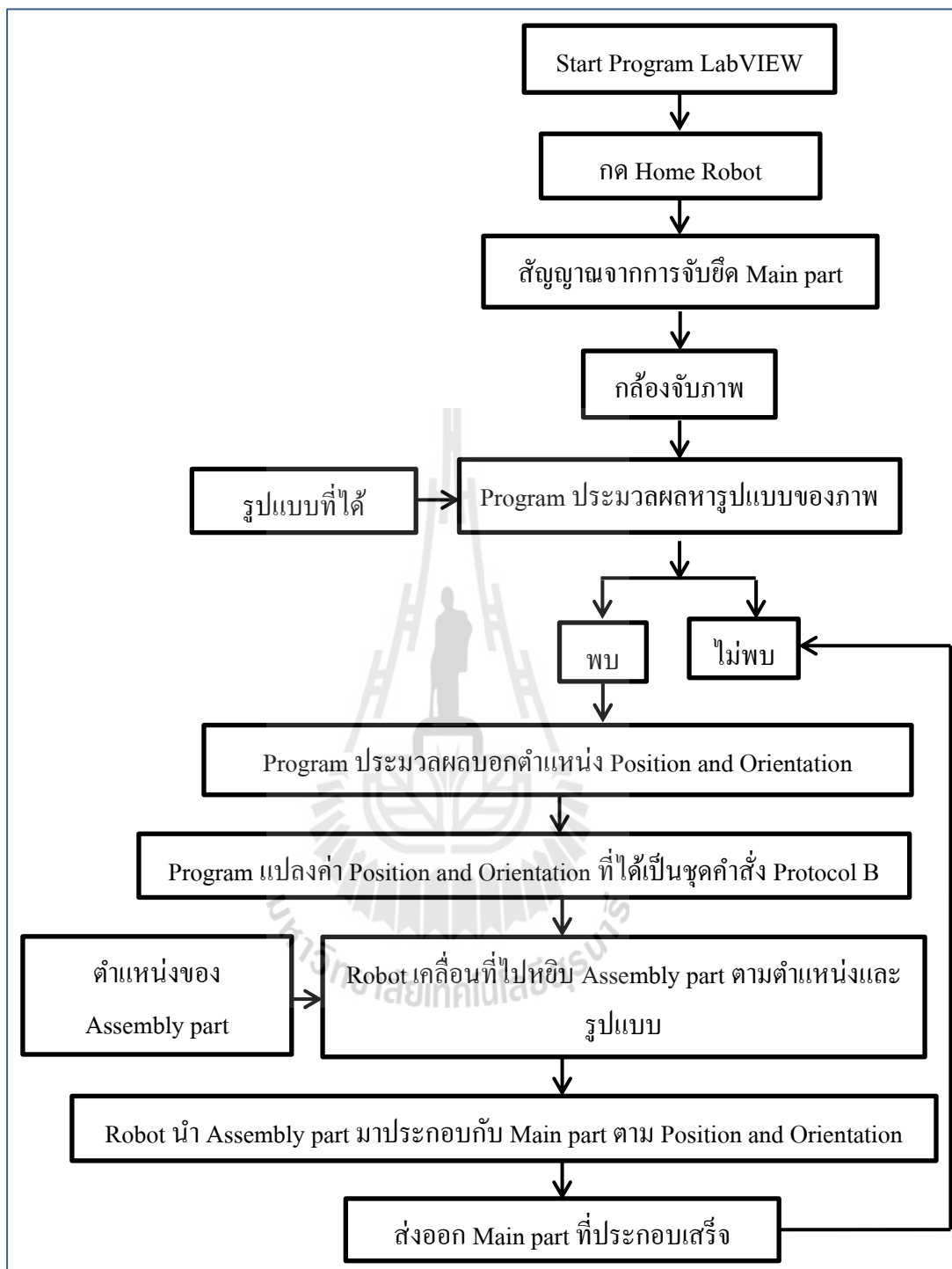
ตารางที่ 3.4 ตาราง Error ของจุดต่าง ๆ ตามแกน X,Y (ต่อ)

Error ใน X, Y (%)			Error ใน X, Y (%)		
วงกลมที่	X	Y	วงกลมที่	X	Y
39	0.301611	1.001271	44	1.001791	0.515983
40	0.330803	0.912953	45	1.466227	0.458485
41	10.59912	0.771277	46	0.746796	0.709178
42	2.945089	0.787376	47	0.625259	0.681579
43	1.419783	0.768977	48	0.613131	0.679279

### 3.6 ทำการเขียนโปรแกรม NI LabVIEW

#### 3.6.1 การออกแบบระบบการทำงานของ NI LabVIEW

เราจะใช้ในการกำหนดขั้นตอนการทำงานของโปรแกรม จากรูปที่ 3.28 โดยการ  
ทำงาน Program NI LabVIEW นั้นใช้ในการแปลงค่าตัวเลขให้เป็นชุดคำสั่งต่าง ๆ เช่นการแปลงค่า  
Pixel จากภาพที่ถ่ายได้ให้อยู่ในค่าจริงเป็นมิลลิเมตร เพื่อนำไปสั่งการให้ Robot เคลื่อนที่ไปยัง  
ตำแหน่งจริง และยังแปลงจากค่ามิลลิเมตรที่ได้ให้ไปอยู่ในชุดคำสั่งของ Robot ยังกำหนด Front  
Panel ของการทำงานเพื่อให้เข้าใจว่าตำแหน่งไหนบอกค่าเกี่ยวกับอะไรอีกด้วย ดังแสดงในรูปที่ 3.34

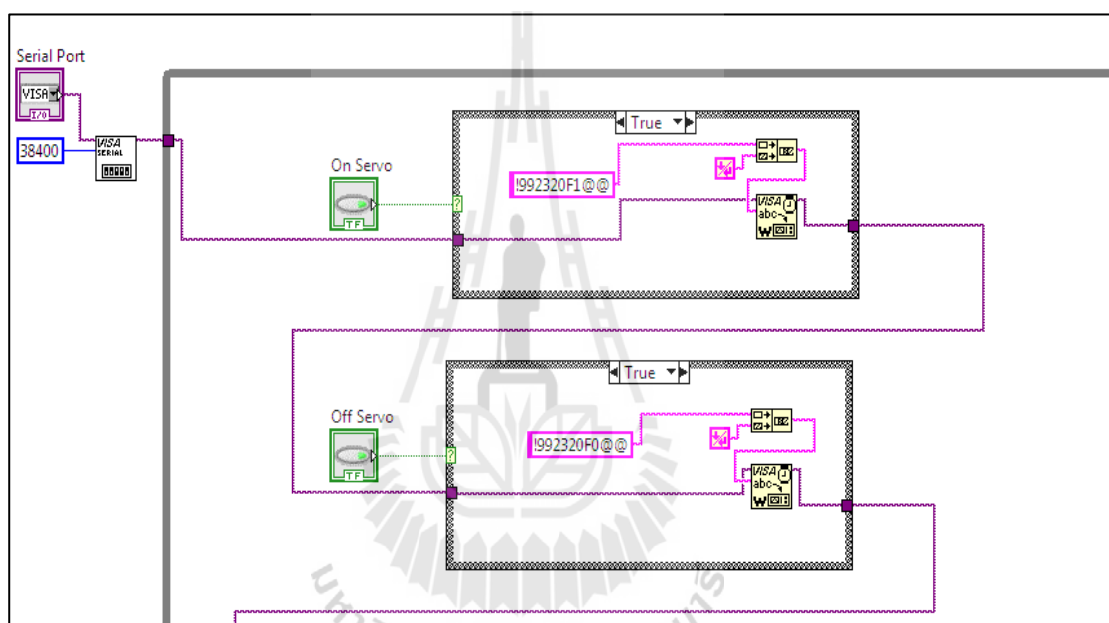


รูปที่ 3.34 แสดงการออกแบบระบบการทำงานของ Program NI LabVIEW

### 3.6.2 ลักษณะ Block diagram สำหรับเครื่องตรวจสอบฯ

Block diagram กางทำงานของโปรแกรมที่ได้ทำการออกแบบ เพื่อตรวจสอบรูปแบบของชิ้นงาน และแนวการวางตัวของชิ้นงาน

- Serial port ตั้งค่า Baudrate 38400 เลือก Port ที่ Front panel Button เป็นแบบ Latch when release หลังจากการ click แล้วจึงจะส่งสัญญาณ ถ้ามีสัญญาณ (true) จึงจะมีการส่งคำสั่งออกไปที่ Serial port เพื่อ On หรือ Off Servo หากไม่มีสัญญาณ (false) จะไม่มีการส่งคำสั่งออกไป ดังแสดงในรูปที่ 3.35

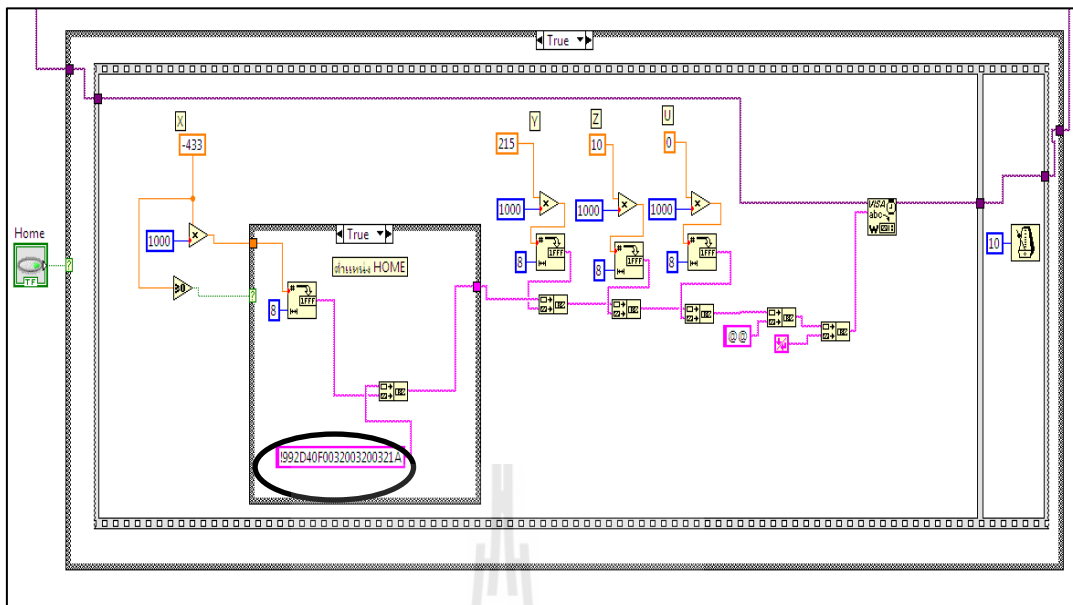


รูปที่ 3.35 แสดงการกำหนด Serial port, กำหนด on, off Servo

- ปุ่ม Home เป็นแบบ latch when release หลังจากการ click แล้วจึงจะส่งสัญญาณถ้ามีสัญญาณ (true) จะสั่งการให้ robot เคลื่อนที่ไปยังตำแหน่งที่ต้องการ หากไม่มีสัญญาณ (false) จะไม่มีการส่งคำสั่งออกไป ดังแสดงในรูปที่ 3.36

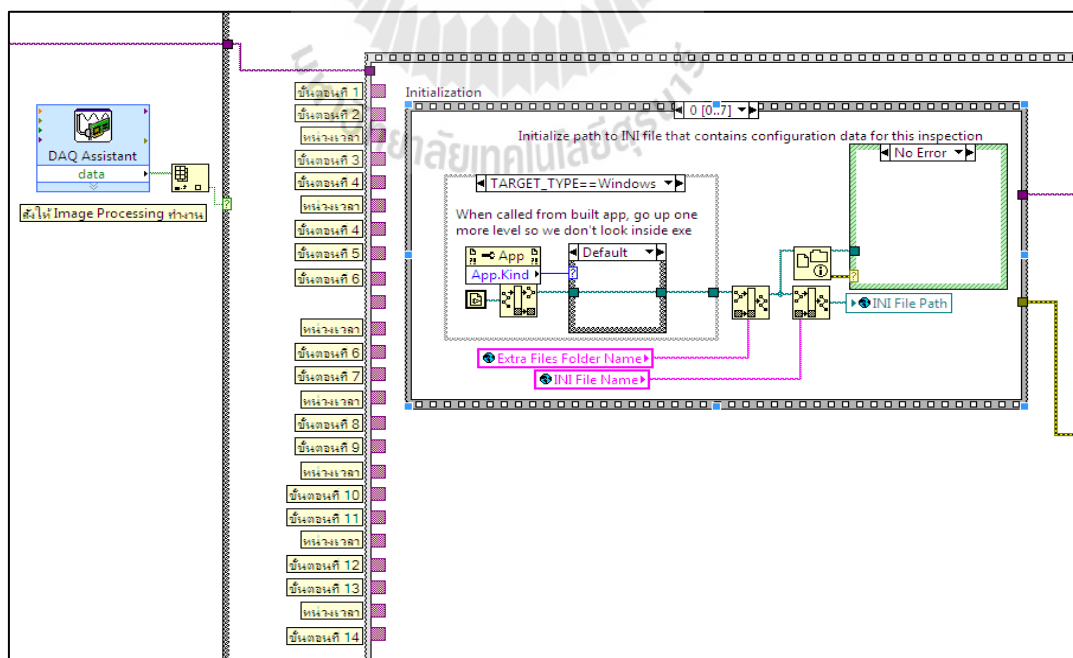
ในวงกลม คือ คำสั่งในการเลือกระบบแขนที่ใช้ในการทำงาน

- ถ้า  $x$  มากกว่า 0 ใช้ระบบแขนซ้าย
- ถ้า  $x$  น้อยกว่า 0 ใช้ระบบแขนขวา



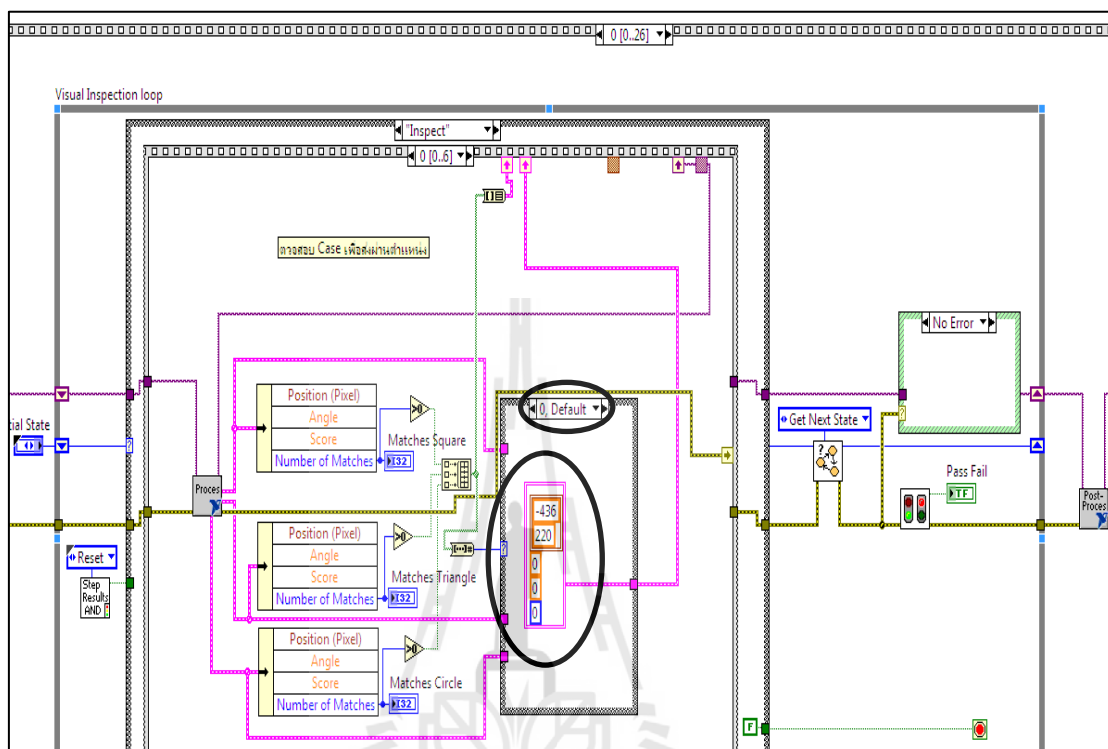
รูปที่ 3.36 แสดงขั้นตอนการทำงานหลังกดปุ่ม Home

- DAQ P0.0 สั่งการให้ Image Processing ทำงาน (สัญญาณประมวลผลภาพสั่งให้เริ่มทำงาน) ดังแสดงในรูปที่ 3.37



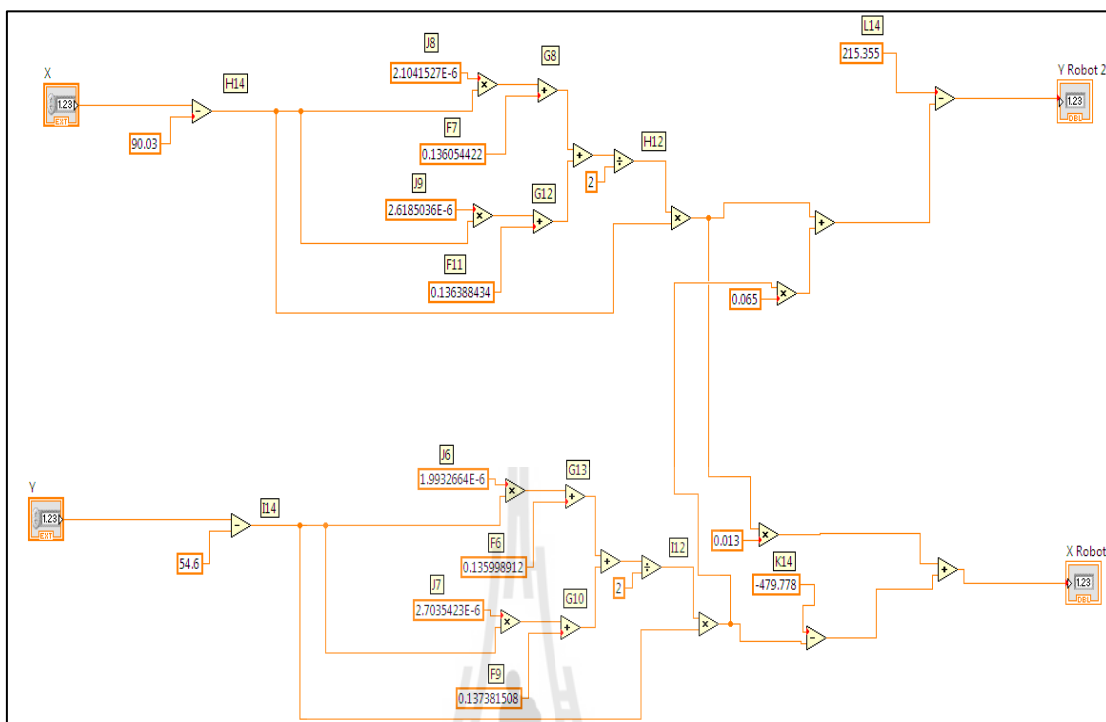
รูปที่ 3.37 แสดงขั้นตอนหลังรับสัญญาณ DAQ P 0.0

- เมื่อมีการส่งข้อมูลภาพเข้ามาแล้ว ต่อไปก็จะนำตรวจสอบ Case เพื่อส่งผ่านตำแหน่งของรูปแบบ และค่าที่พบเพื่อใช้ในการคำนวณ ดังแสดงในรูปที่ 3.38



รูปที่ 3.38 แสดงขั้นตอนการตรวจสอบรูปแบบตาม Case ต่าง ๆ

- เครื่องมือที่ใช้ในการคำนวณตำแหน่งการวางชิ้นงานจากตำแหน่งที่ได้จากการประมวลผลภาพ ที่ได้หลังจากการ Calibration ผลที่ได้จากการคำนวณค่า Error เราได้นำมาออกแบบค่าชดเชยในการคำนวณ พิกัดการวางชิ้นงาน ดังแสดงในรูปที่ 3.39

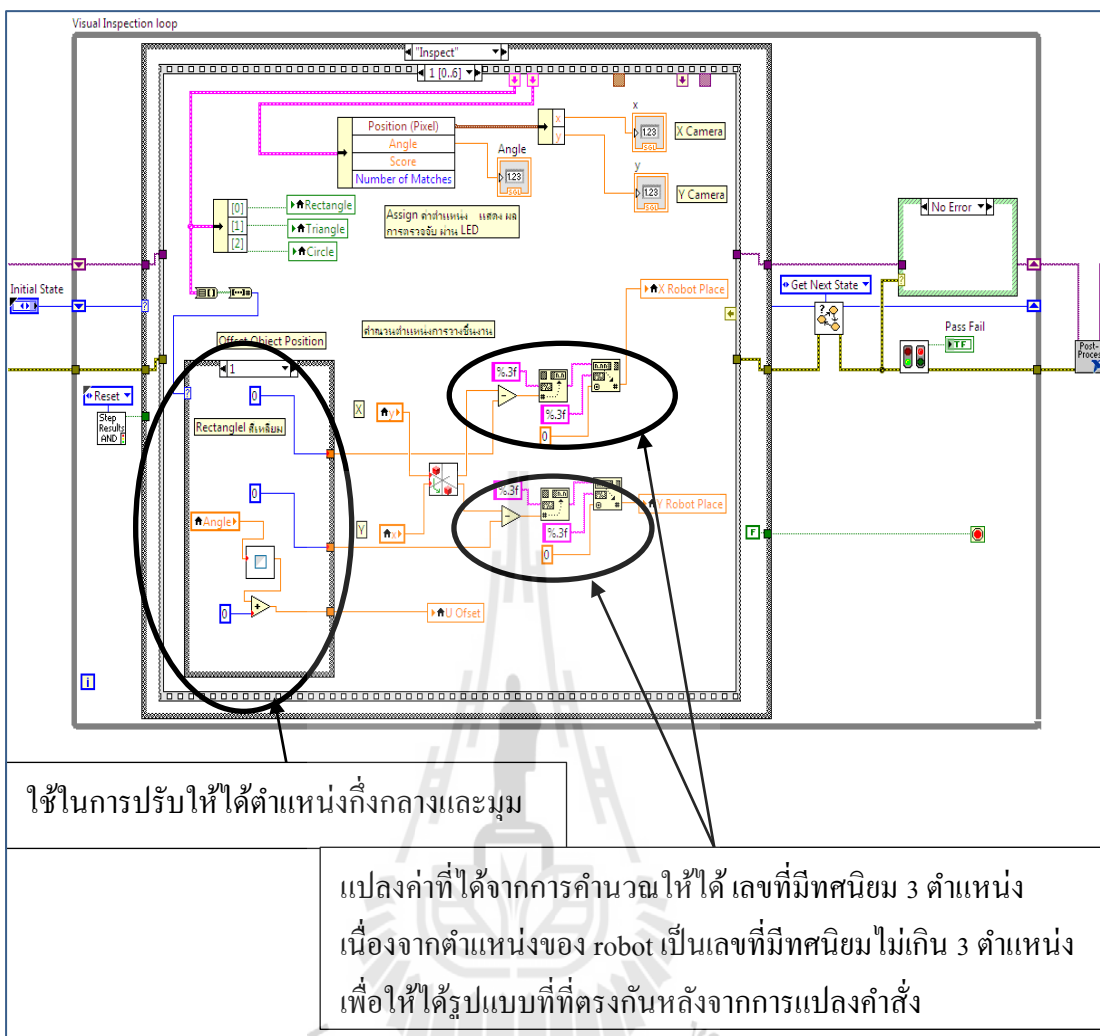


รูปที่ 3.39 แสดงขั้นตอนการชดเชยค่าตำแหน่งหลังจาก Calibration

- การปรับให้ได้ตำแหน่งกึ่งกลาง และมุมของรูป ด้วยการแปลงค่าที่ได้จากการคำนวณให้ได้ เลขที่มีทศนิยม 3 ตำแหน่ง ดังแสดงในรูปที่ 3.40

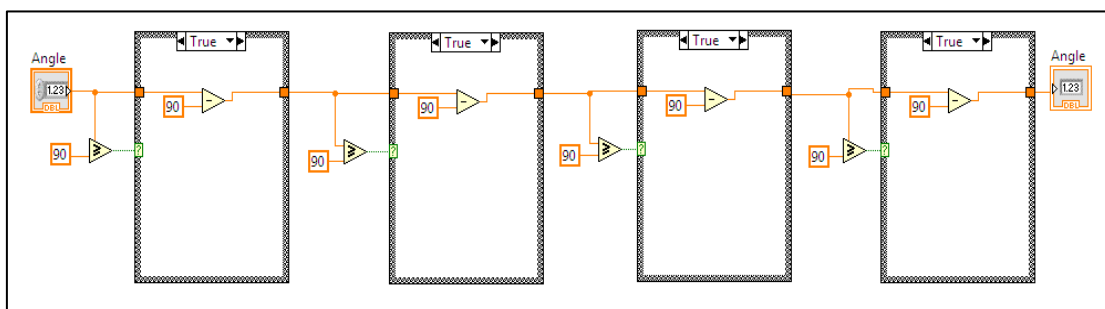
มหาวิทยาลัยเทคโนโลยีสุรนารี





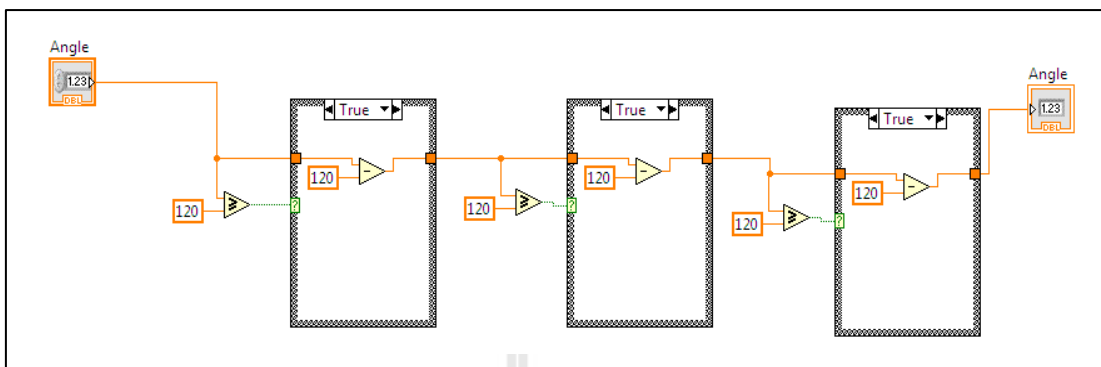
รูปที่ 3.40 แสดงขั้นตอนการปรับค่าตำแหน่งกึ่งกลางของรูปแบบ

- การคำนวณค่าของการหมุนของมุมไม่ให้น้อยที่สุดของรูปสี่เหลี่ยมดังแสดงในรูปที่ 3.41



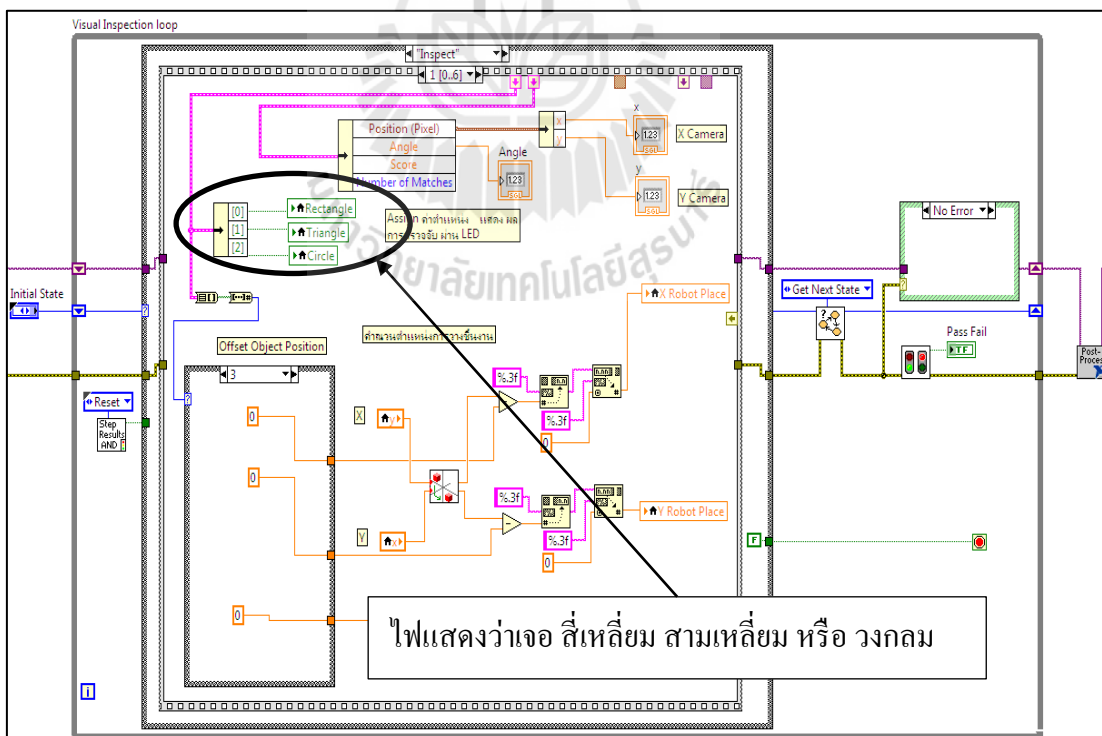
รูปที่ 3.41 แสดงขั้นตอนการหาหมุนหามุมรูปสี่เหลี่ยมที่น้อยที่สุด

- การคำนวณค่ามุมของรูปสามเหลี่ยมให้มุมน้อยที่สุด ดังแสดงในรูปที่ 3.42



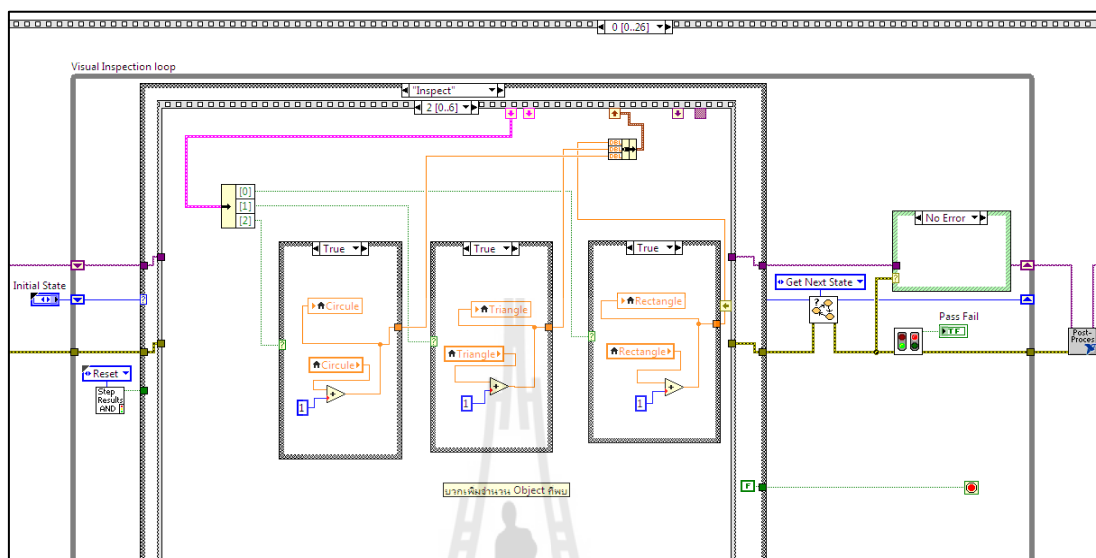
รูปที่ 3.42 แสดงขั้นตอนการหามุมสามเหลี่ยมที่น้อยที่สุด

- ไฟแสดงสถานะว่าตรวจพบรูปแบบของชิ้นงานเป็น สี่เหลี่ยม, สามเหลี่ยม หรือ วงกลม ดังแสดงในรูปที่ 3.43



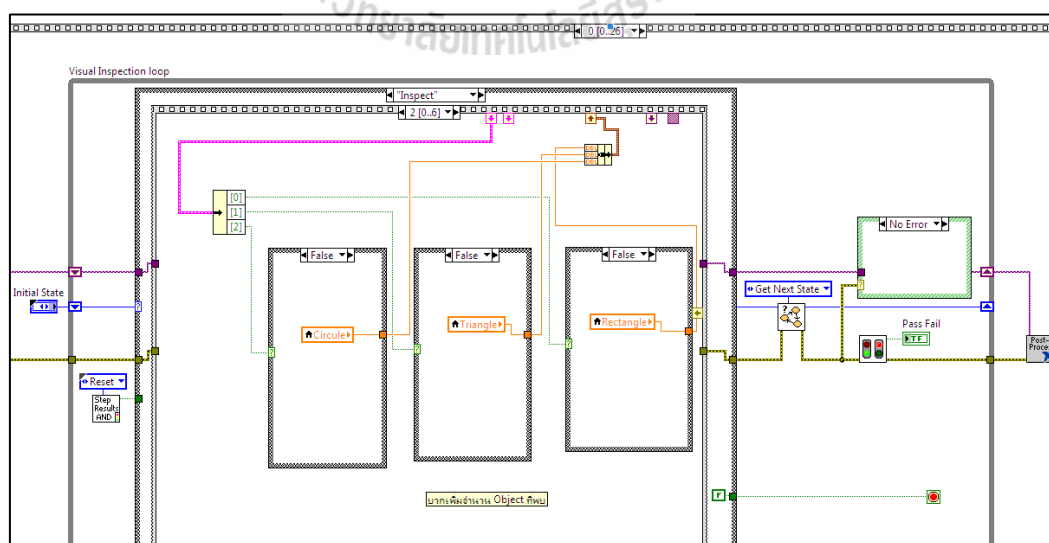
รูปที่ 3.43 แสดงการบอกไฟสถานะการตรวจพบรูปแบบต่าง ๆ

- หากตรวจเจอ สีเหลือง, สามเหลี่ยม หรือวงกลม จะเพิ่มค่าเพื่อนับจำนวนที่ตรวจพบ  
 ดังแสดงในรูปที่ 3.44



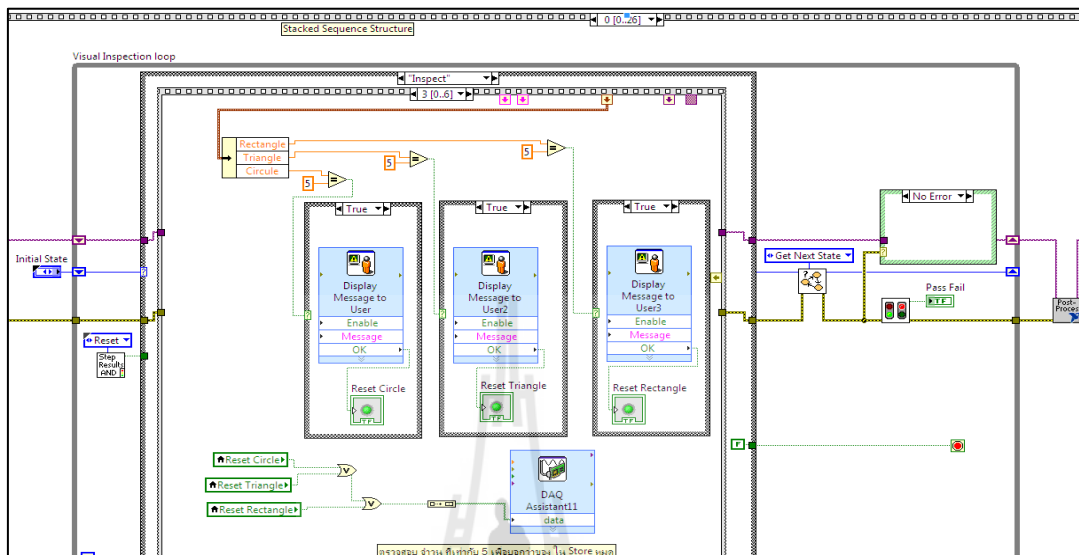
รูปที่ 3.44 แสดงขั้นตอนการบอกจำนวนที่ได้มีการตรวจพบ

- หากตรวจไม่เจอ สีเหลือง สามเหลี่ยม หรือวงกลม จะไม่มีการเพิ่มค่าเพื่อนับ  
 จำนวนดังแสดงในรูปที่ 3.45



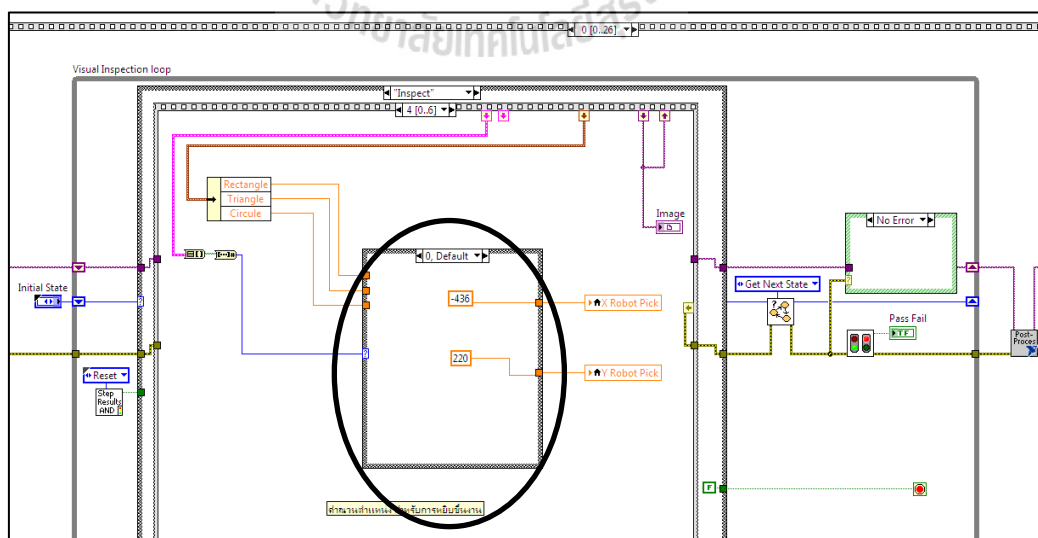
รูปที่ 3.45 แสดงขั้นตอนการบอกจำนวนที่ไม่มีการตรวจพบ

- หากมีการตรวจพบรูปแบบใดรูปแบบหนึ่งครบ 5 ครั้ง จะมีการแจ้งเตือนผู้ใช้ และส่งสัญญาณออกไปที่ DAQ ดังแสดงในรูปที่ 3.46



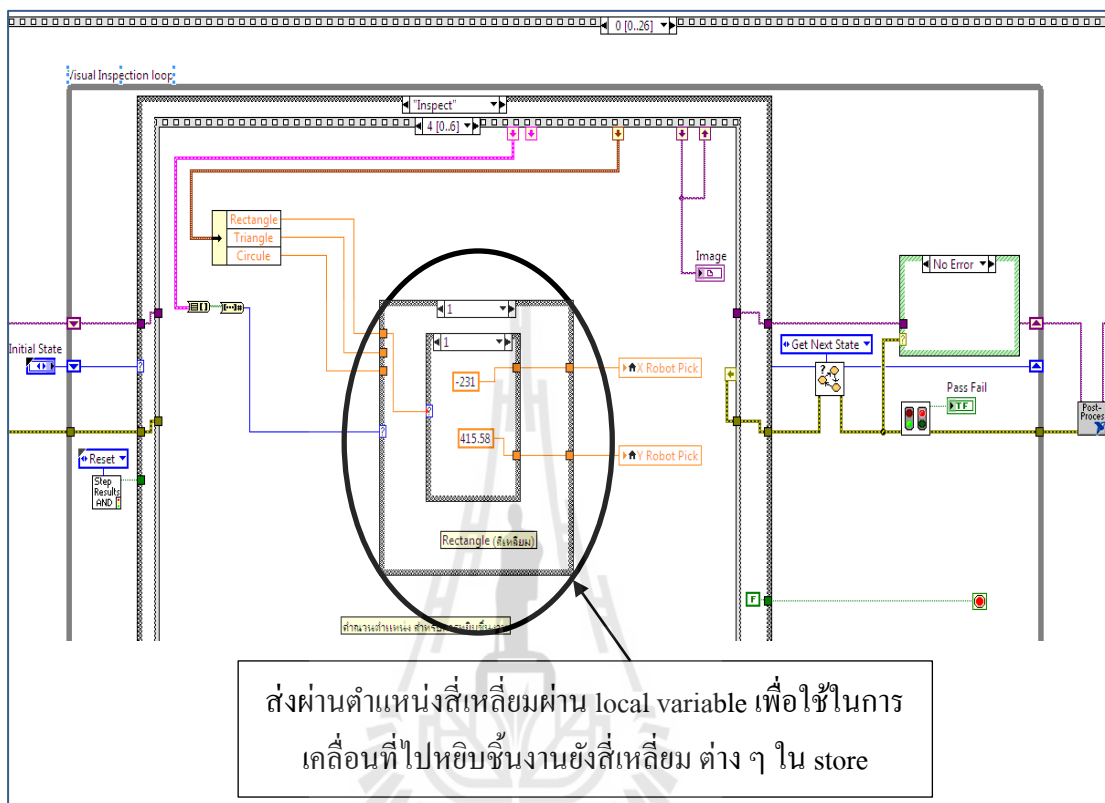
รูปที่ 3.46 แสดงขั้นตอนการบอกการตรวจสอบครบ 5 ครั้ง

- การส่งผ่านตำแหน่งการหยิบชิ้นงานใน Store ตามรูปแบบที่จับภาพได้และตามลำดับดังแสดงในรูปที่ 3.47



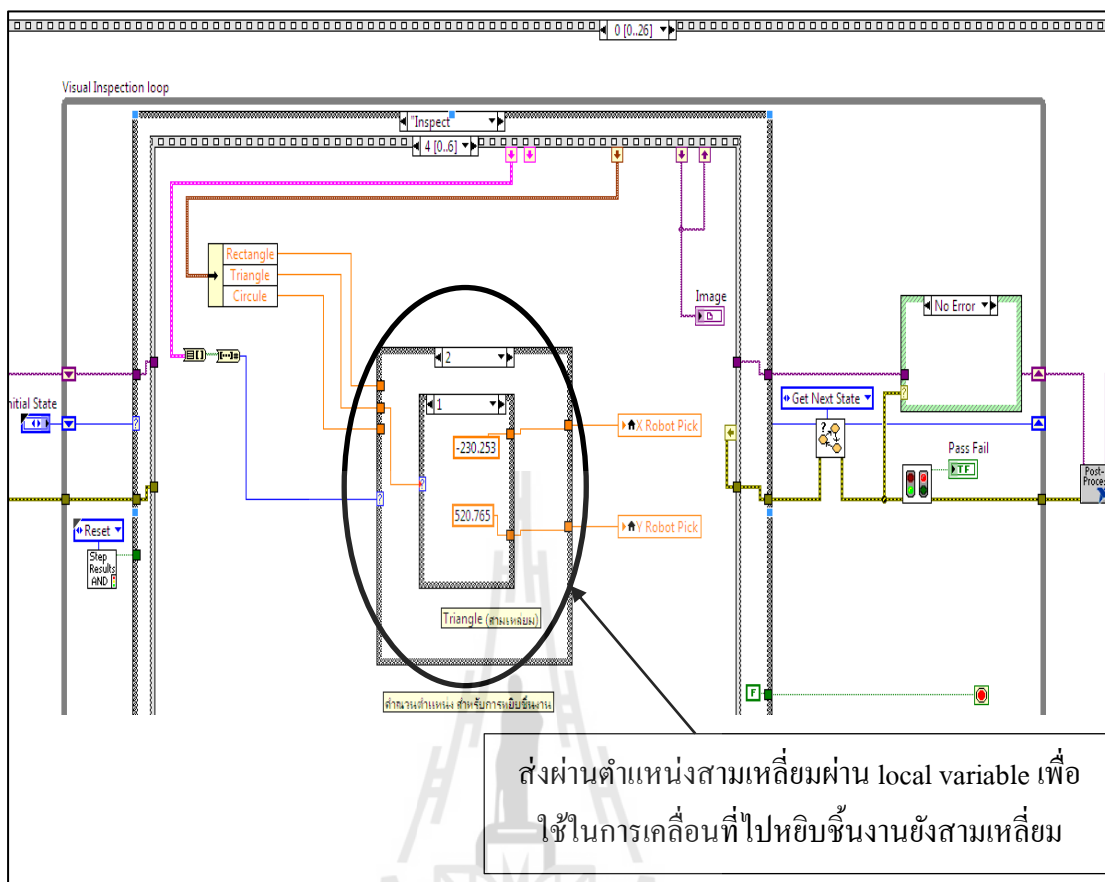
รูปที่ 3.47 แสดงการบอกตำแหน่งการหยิบชิ้นงานใน Store ตำแหน่งต่าง ๆ

- ขั้นตอนการส่งผ่านพิกัดชิ้นงานรูปสี่เหลี่ยมใน Store ในตำแหน่งต่าง ๆ ดังแสดง  
ในรูปที่ 3.48



รูปที่ 3.48 แสดงการบอกตำแหน่งการหยิบชิ้นงานรูปสี่เหลี่ยมใน Store ตำแหน่งต่าง ๆ

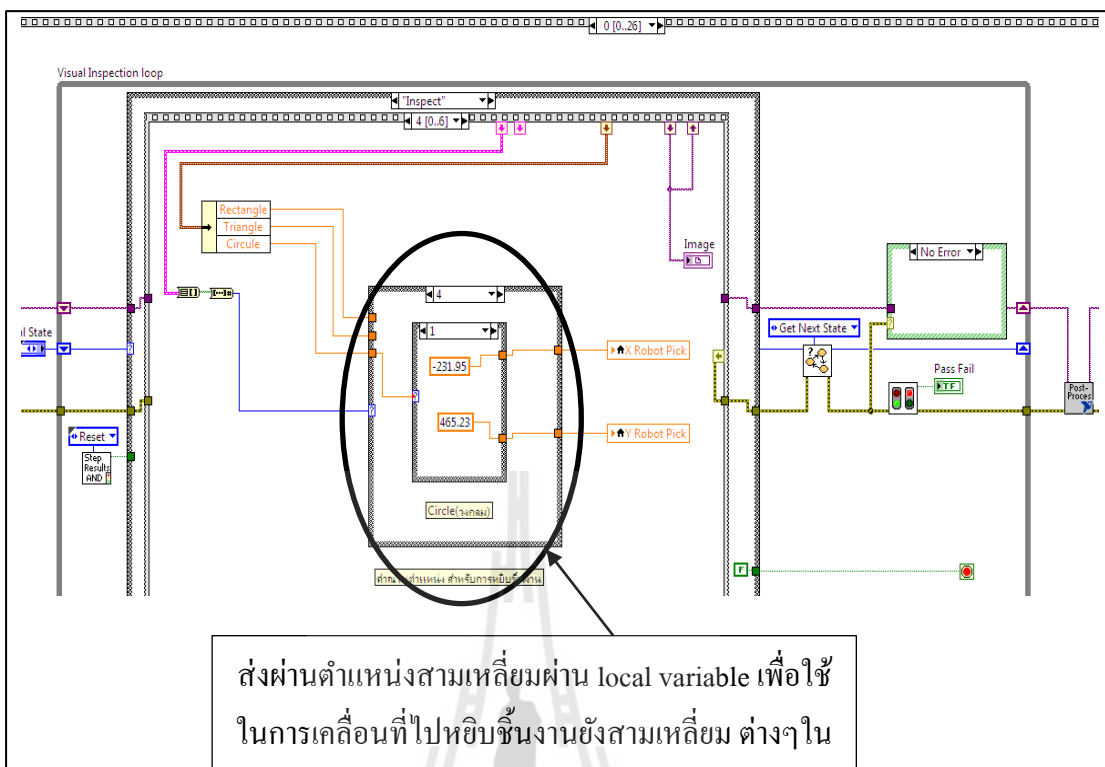
- ขั้นตอนการส่งผ่านพิกัดชิ้นงานรูปสามเหลี่ยมใน Store ในตำแหน่งต่าง ๆ  
ดังแสดงในรูปที่ 3.49



รูปที่ 3.49 แสดงการบอกตำแหน่งการหยิบชิ้นงานรูปสามเหลี่ยมใน Store ตำแหน่งต่าง ๆ

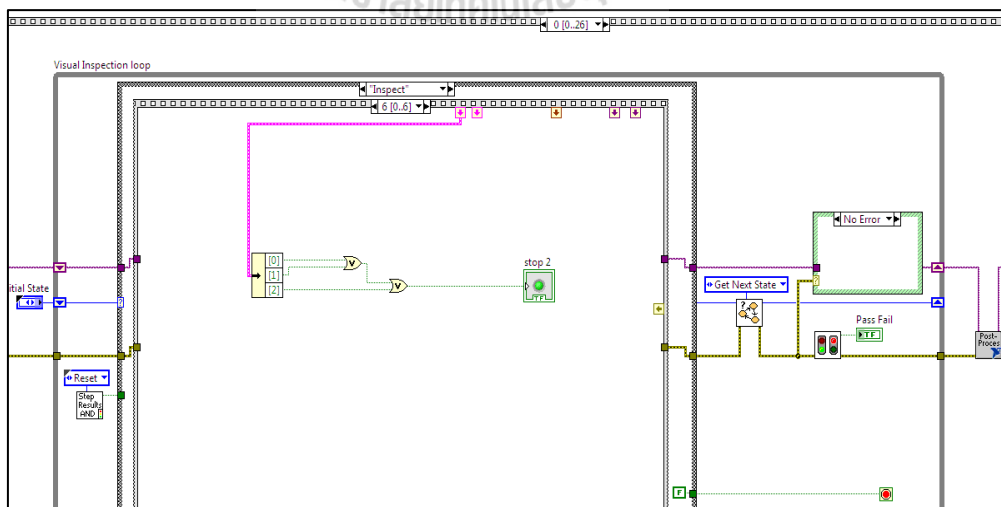
- ขั้นตอนการส่งผ่านพิกัดชิ้นงานรูปวงกลมใน Store ในตำแหน่งต่าง ๆ ดังแสดงใน

รูปที่ 3.50



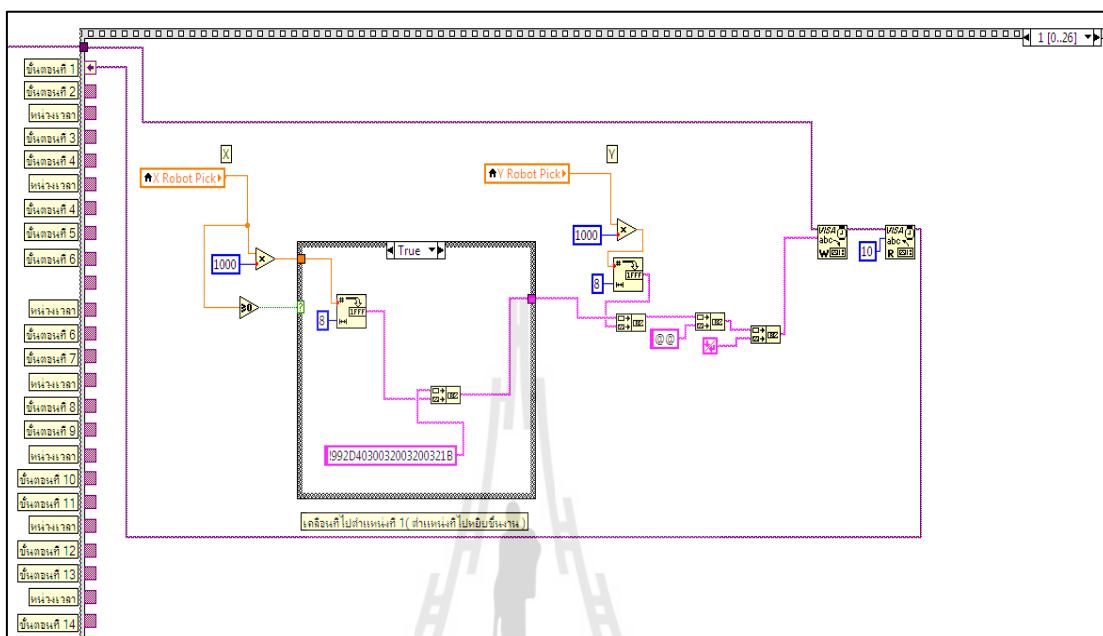
รูปที่ 3.50 แสดงการบอกตำแหน่งการหยิบชิ้นงานรูปร่างกลมใน Store ตำแหน่งต่าง ๆ

- หากมีการพบรูปแบบใดรูปแบบหนึ่ง จะจบการทำงานในส่วนของการประมวลผลภาพดังแสดงในรูปที่ 3.51



รูปที่ 3.51 แสดงขั้นตอนการบอกการจบกระบวนการประมวลผลด้วยภาพ

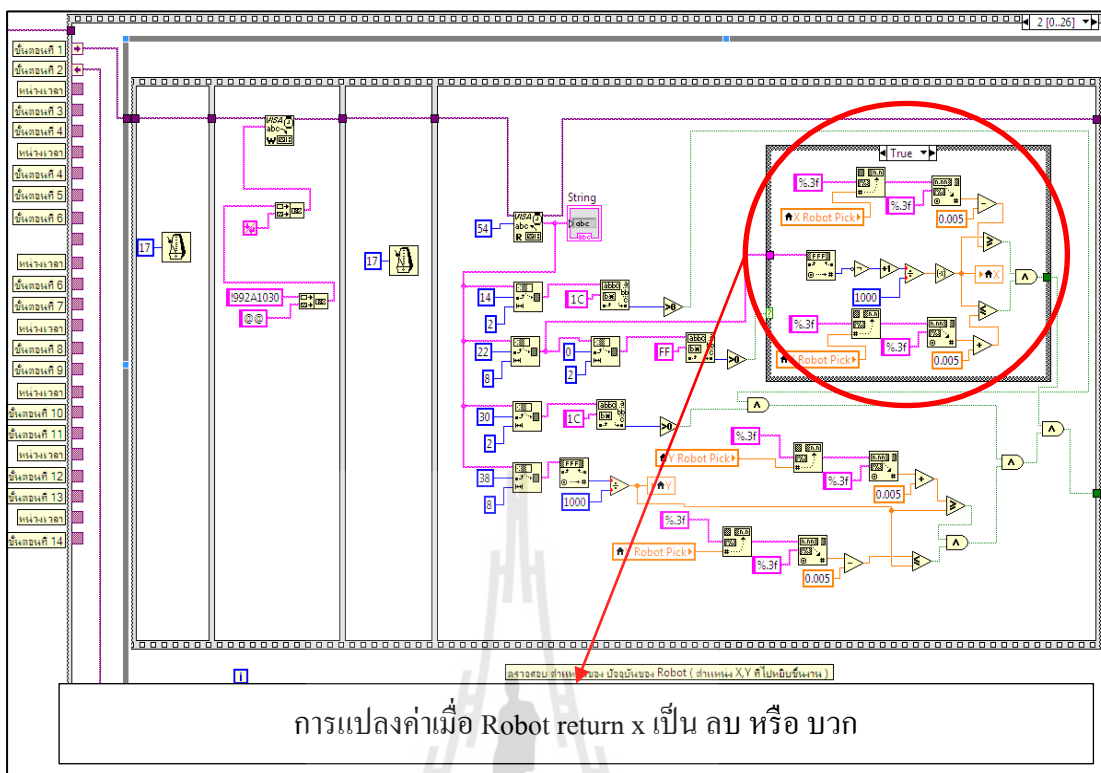
- ส่งคำสั่งให้ Robot เคลื่อนที่ไปยังตำแหน่ง X, Y ก่อนจับชิ้นงาน ดังแสดงในรูปที่ 3.52



รูปที่ 3.52 แสดงขั้นตอนการส่งตำแหน่งให้ Robot เคลื่อนที่ก่อนจะจับชิ้นงาน

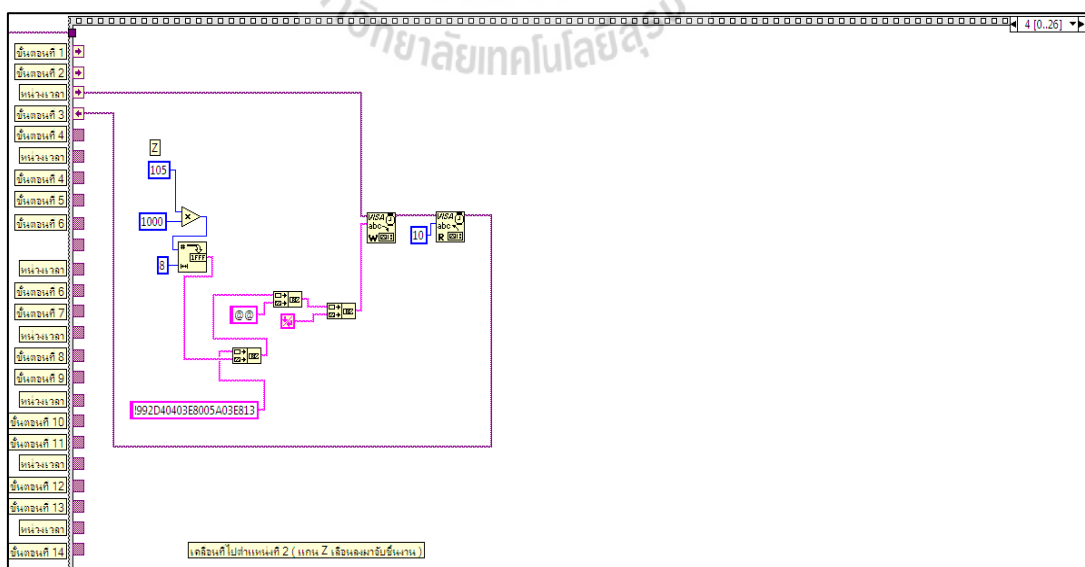
- ส่งคำสั่งให้ Robot เพื่อตรวจสอบตำแหน่งปัจจุบันของ Robot ว่าไปถึงตำแหน่งที่เราต้องการหรือยัง และการแปลงค่าตำแหน่งเป็น ลบ หรือ บวก ดังแสดงในรูปที่ 3.53





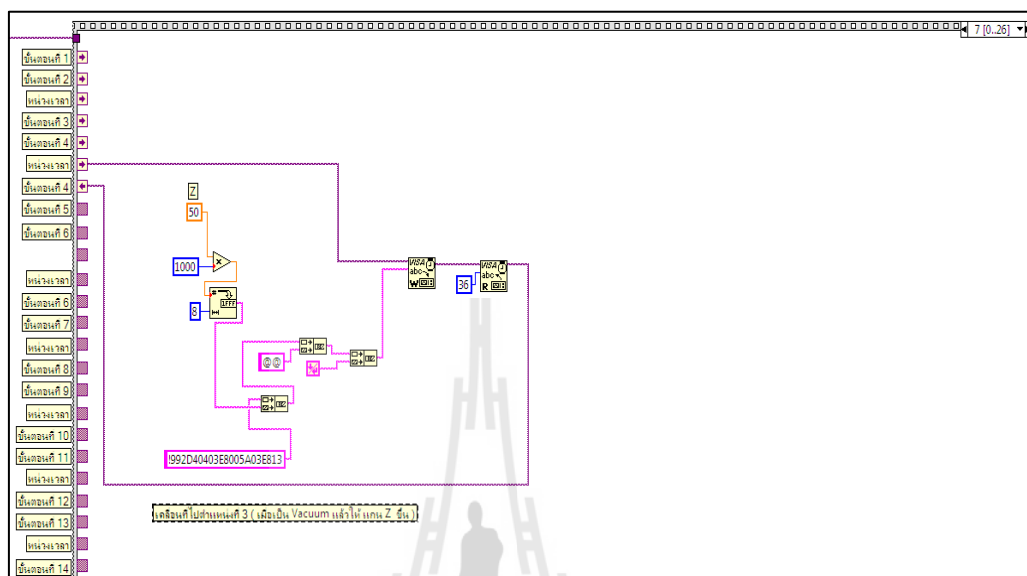
รูปที่ 3.53 แสดงขั้นตอนการแปลงค่า Robot return X

- ส่งคำสั่งให้ Robot เคลื่อนที่ลงไปจับชิ้นงาน ดังแสดงในรูปที่ 3.54



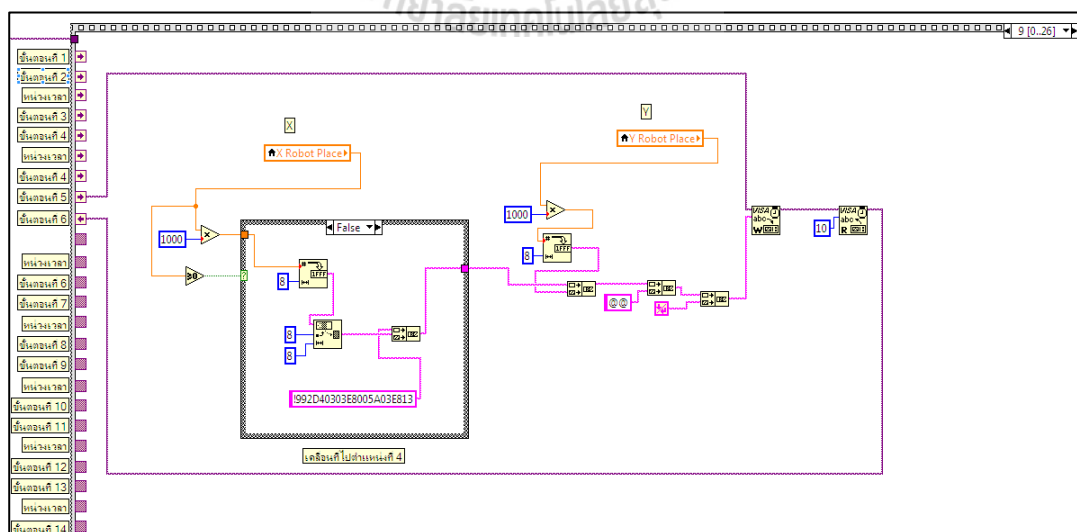
รูปที่ 3.54 แสดงขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ลงไปจับชิ้นงาน

- ส่ง Output ออกไปยัง DAQ เพื่อสั่งให้ Vacuum ทำงาน และรอจนกว่าจะมีสัญญาณ Vacuum และส่งคำสั่งให้ Robot ค้างขึ้นงานขึ้นเมื่อจับขึ้นงานแล้ว ดังแสดงในรูปที่ 3.55



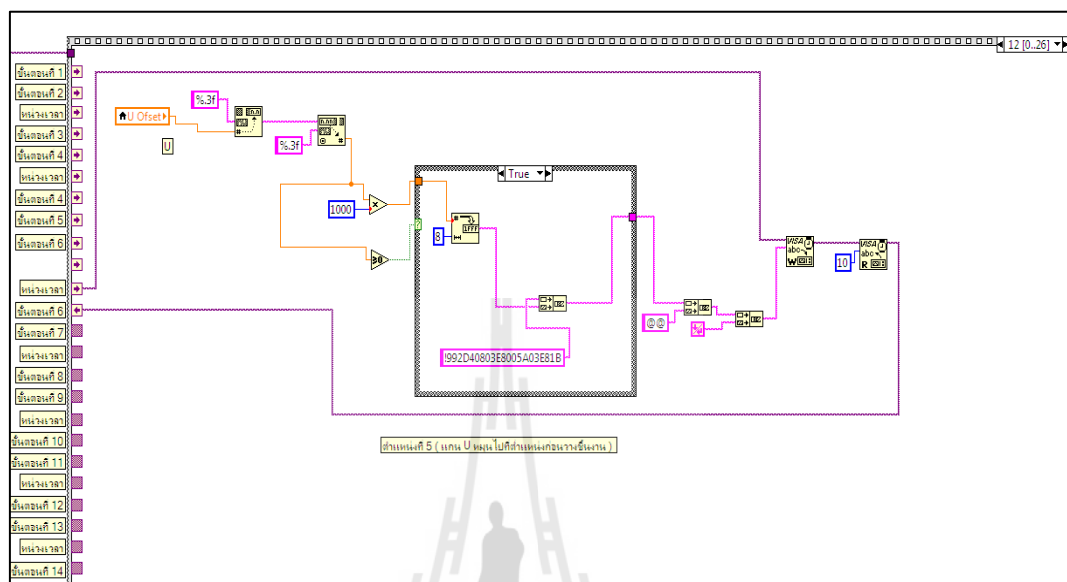
รูปที่ 3.55 แสดงขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ขึ้นเมื่อจับขึ้นงานแล้ว

- เมื่อ Robot จับขึ้นงาน ได้แล้วจะส่งคำสั่งให้ Robot เคลื่อนที่ไปยังตำแหน่ง X, Y ก่อนที่จะประกอบชิ้นงาน ดังแสดงในรูปที่ 3.56



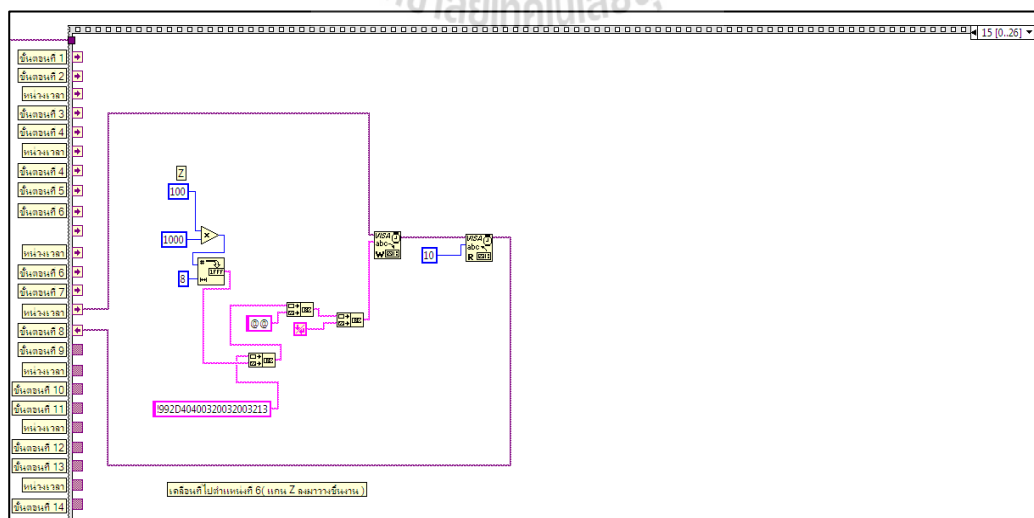
รูปที่ 3.56 แสดงขั้นตอนการส่งสัญญาณให้ Robot เคลื่อนที่ไปยังตำแหน่งก่อนประกอบชิ้นงาน

- เมื่อ Robot เคลื่อนที่มายังตำแหน่งแกน X, Y ในการประกอบ จะส่งคำสั่งให้ Robot หมุนชิ้นงานสำหรับการวางชิ้นงานให้ตรง ดังแสดงในรูปที่ 3.57



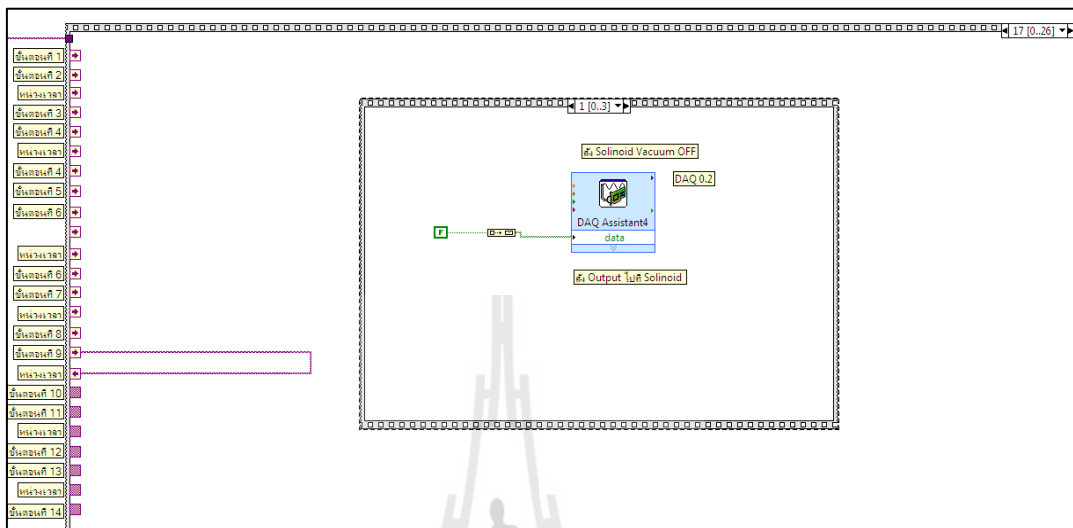
รูปที่ 3.57 แสดงขั้นตอนการส่งสัญญาณให้ Robot หมุนชิ้นงานก่อนประกอบ

- เมื่อ Robot หมุนตามตำแหน่งในแนวแกน U เสร็จแล้ว จะส่งคำสั่งให้ Robot วางชิ้นงาน ดังแสดงในรูปที่ 3.58



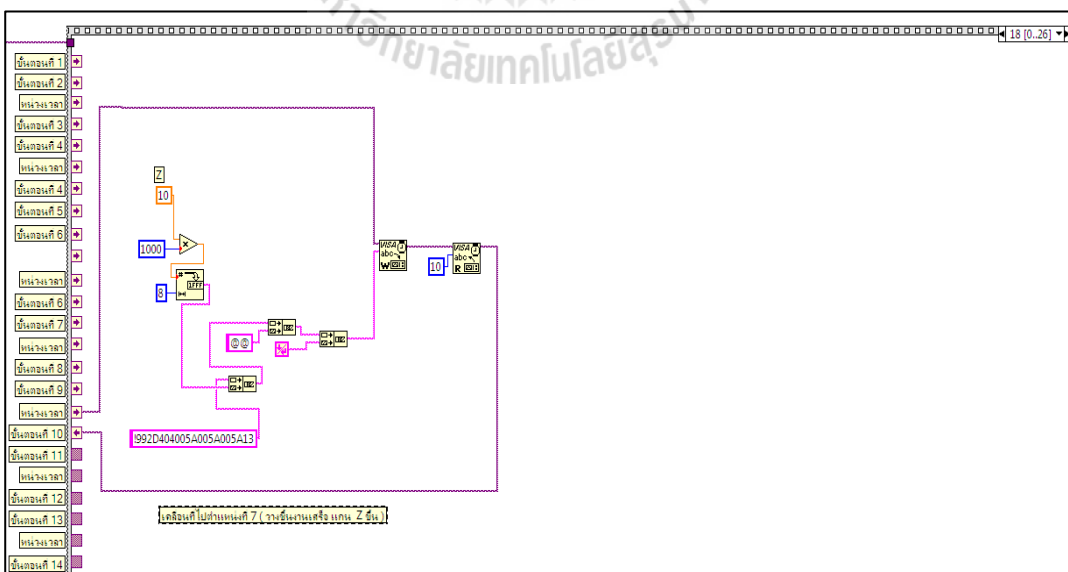
รูปที่ 3.58 แสดงขั้นตอนการส่งสัญญาณให้ Robot วางชิ้นงานเพื่อประกอบ

- เมื่อแกน Z เคลื่อนที่ลงมาถึงส่วนที่จะวางชิ้นงานในการประกอบ เราจะส่ง Output ออกไปที่ DAQ เพื่อสั่งให้ Vacuum หยุดทำงาน เพื่อปล่อยชิ้นส่วนในการประกอบ ดังแสดงในรูปที่ 3.59



รูปที่ 3.59 แสดงขั้นตอนการตัดสัญญาณ Vacuum

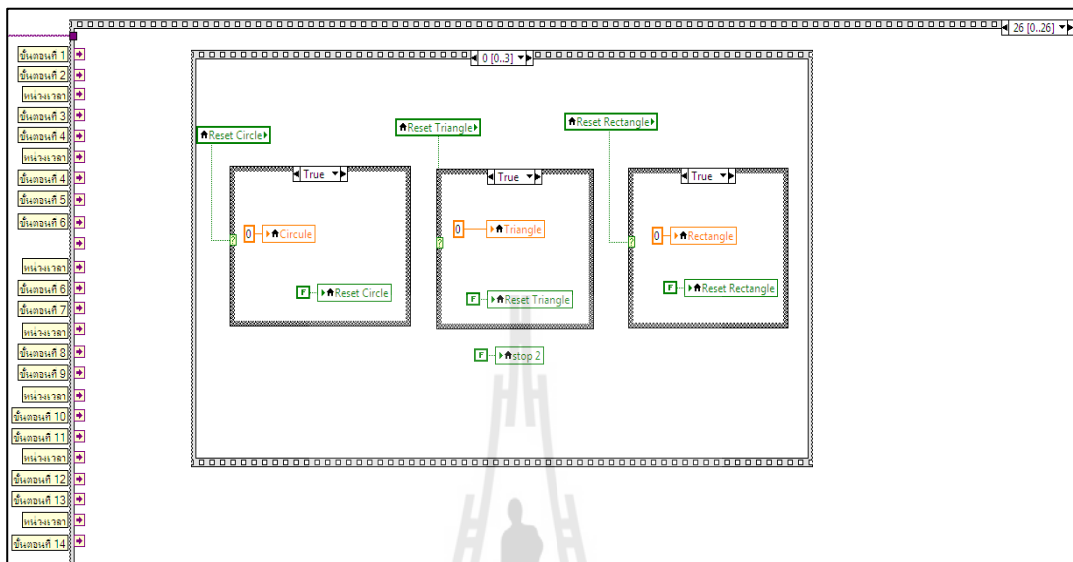
- เมื่อวางชิ้นส่วนในการประกอบเสร็จสิ้นแล้ว จะส่งคำสั่งให้แกน Z ของ Robot เคลื่อนที่ขึ้น ดังแสดงในรูปที่ 3.60



รูปที่ 3.60 แสดงขั้นตอนการส่งสัญญาณให้แกน Z เคลื่อนที่ขึ้นหลังจากประกอบเสร็จ

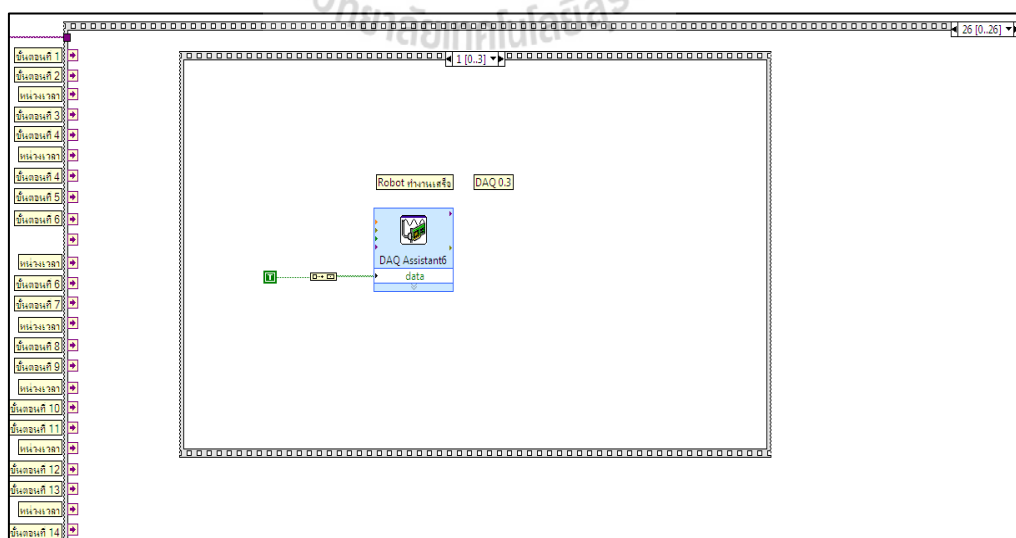


- หากมีการส่งสัญญาณ Reset (จำนวนครบ 5 ) ผ่าน local variable จะต้องทำการ reset ตัวนับ และคืนค่าสัญญาณ Reset กลับสู่สถานะเริ่มต้น ดังแสดงในรูปที่ 3.63



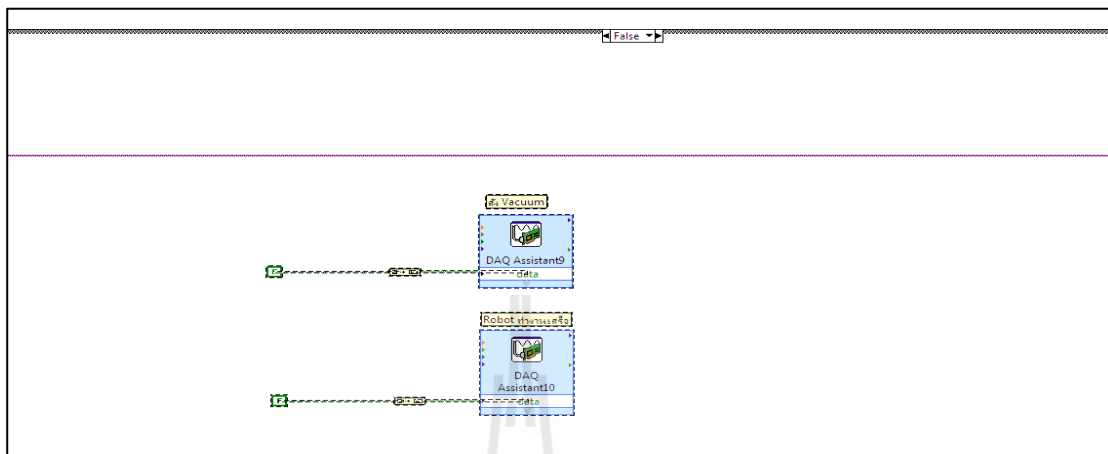
รูปที่ 3.63 แสดงขั้นตอนการส่งสัญญาณให้ Reset จำนวนที่ตรวจสอบพบชิ้นงาน

- ส่งสัญญาณ Output ไปที่ DAQ เพื่อบอก PLC ว่า จบการทำงานของ Robot ดังแสดงในรูปที่ 3.64



รูปที่ 3.64 แสดงการบอก PLC ว่าจบการทำงานในส่วนของ Robot

- ค่าเริ่มต้นสำหรับสัญญาณ Vacuum และ สัญญาณจบการทำงานของ Robot ดัง  
แสดงในรูปที่ 3.65



รูปที่ 3.65 แสดงขั้นตอนการส่งสัญญาณค่าเริ่มต้นและสัญญาณจบการทำงาน



## บทที่ 4

### ผลการออกแบบเครื่องจักรและการทำงานของเครื่องจักร

สำหรับบทนี้จะนำเสนอผลการดำเนินงานสร้างเครื่องจักรต้นแบบที่ใช้ในการหยิบและวางในกระบวนการประกอบแบบอัตโนมัติ ดังแสดงในรูปที่ 4.1 โดยกระบวนการวิจัยนี้ได้แบ่งการทำงานออกเป็น 2 ระบบ คือระบบการทำงานของชุดกลไกอัตโนมัติสำหรับลำเลียงชิ้นงานและจับยึดชิ้นงาน และระบบการทำงานของตรวจสอบด้วยภาพ จึงต้องทำการทดสอบระบบทั้งสองระบบให้สมบูรณ์ก่อนที่จะนำมาทำงานร่วมกัน เพื่อให้จะได้ระบบการทำงานที่ถูกต้องและแม่นยำ



รูปที่ 4.1 การออกแบบและสร้างหุ่นยนต์ที่ใช้ในการหยิบและวางในกระบวนการประกอบ

#### 4.1 ผลของการเขียนโปรแกรมระบบการทำงานและการทดสอบการทำงานของชุดกลไกอัตโนมัติสำหรับส่งชิ้นงาน ชุดลำเลียงชิ้นงาน ชุดจับยึดชิ้นงาน

จากการกำหนด Address ของ PLC ได้นำไปเขียนโปรแกรมเพื่อควบคุมการทำงานของระบบลำเลียงชิ้นงาน โดยการลำเลียงชิ้นงานจะเริ่มตั้งแต่ชิ้นงานถูกส่งออกจาก Store และไหลไปจนถึงจุดจับยึดชิ้นงานและส่งออกชิ้นงานที่ประกอบเสร็จสมบูรณ์ ซึ่งการทำงานของกลไกชุดส่งชิ้นงาน ชุดลำเลียงชิ้นงาน ชุดจับยึดชิ้นงานที่ต้องการทำการตรวจสอบเข้าหากำลัง โดยการออกแบบงานนี้จะทำการตรวจสอบชิ้นที่ได้ทำการออกแบบ แทนแผ่นวงจรอิเล็กทรอนิกส์ โดยจะ



ออกแบบมา 3 รูปแบบ วงกลม สามเหลี่ยม สี่เหลี่ยม การทำงานของเครื่องจักรต้นแบบนี้ จะทำการเก็บข้อมูลเวลาของการทำงานของเครื่องจักรต้นแบบจึงรวมเวลาในการส่งชิ้นงานเข้าไปในระบบราง Conveyer และการส่งออกชิ้นงานที่ถูกประกอบเสร็จขึ้นอีกด้วย โดยเราจะทดสอบแต่ละรูปแบบจำนวน 10 ครั้ง/รูปแบบ ซึ่งการเก็บข้อมูลนี้เป็นการทดสอบการทำงานของเครื่องจักรต้นแบบโดยยังไม่ได้ทำการวิเคราะห์ผลของการจับภาพและการประกอบชิ้นงาน ได้ผลการทดลอง ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 แสดงเวลาการทำงานของชุดกลไกอัตโนมัติสำหรับลำเลียงชิ้นงาน

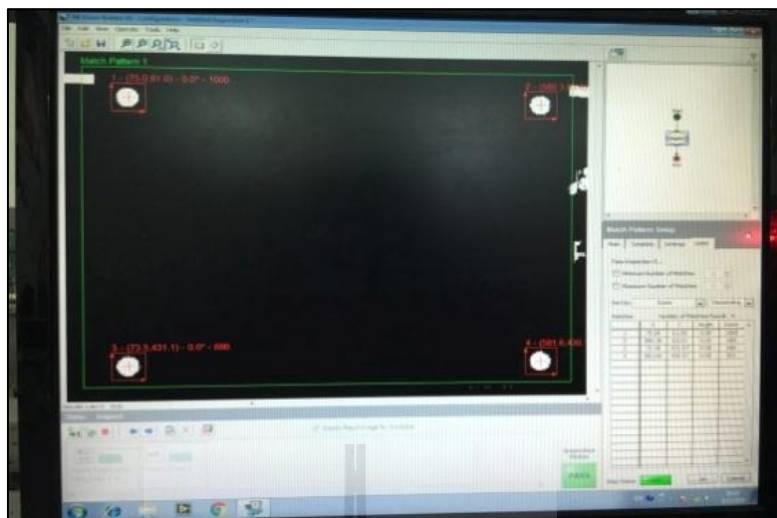
ลำดับที่	ขั้นตอน	เวลาเฉลี่ยทั้ง 30 ครั้ง ( วินาที )
1	ลำเลียงชิ้นงาน	20
2	ตรวจสอบและประกอบ	10
	รวม	30

จากผลของการทดสอบเวลาที่ใช้ในการทำงานของชุดกลไกอัตโนมัติสำหรับการลำเลียงชิ้นงานและจับยึดใช้เวลาเฉลี่ย 30 วินาทีต่อการทำงาน 1 รอบ นับจากชิ้นงานถูกปล่อยออกมาจาก Store ถึงกระบวนการตรวจสอบชิ้นงานและประกอบชิ้นส่วน จนถึงสิ้นสุดกระบวนการการทำงาน

## 4.2 ผลการทดสอบระบบตรวจสอบด้วยภาพ

### 4.2.1 ผลของการกำหนดรูปแบบต่างๆในโปรแกรม NI Vision Builder

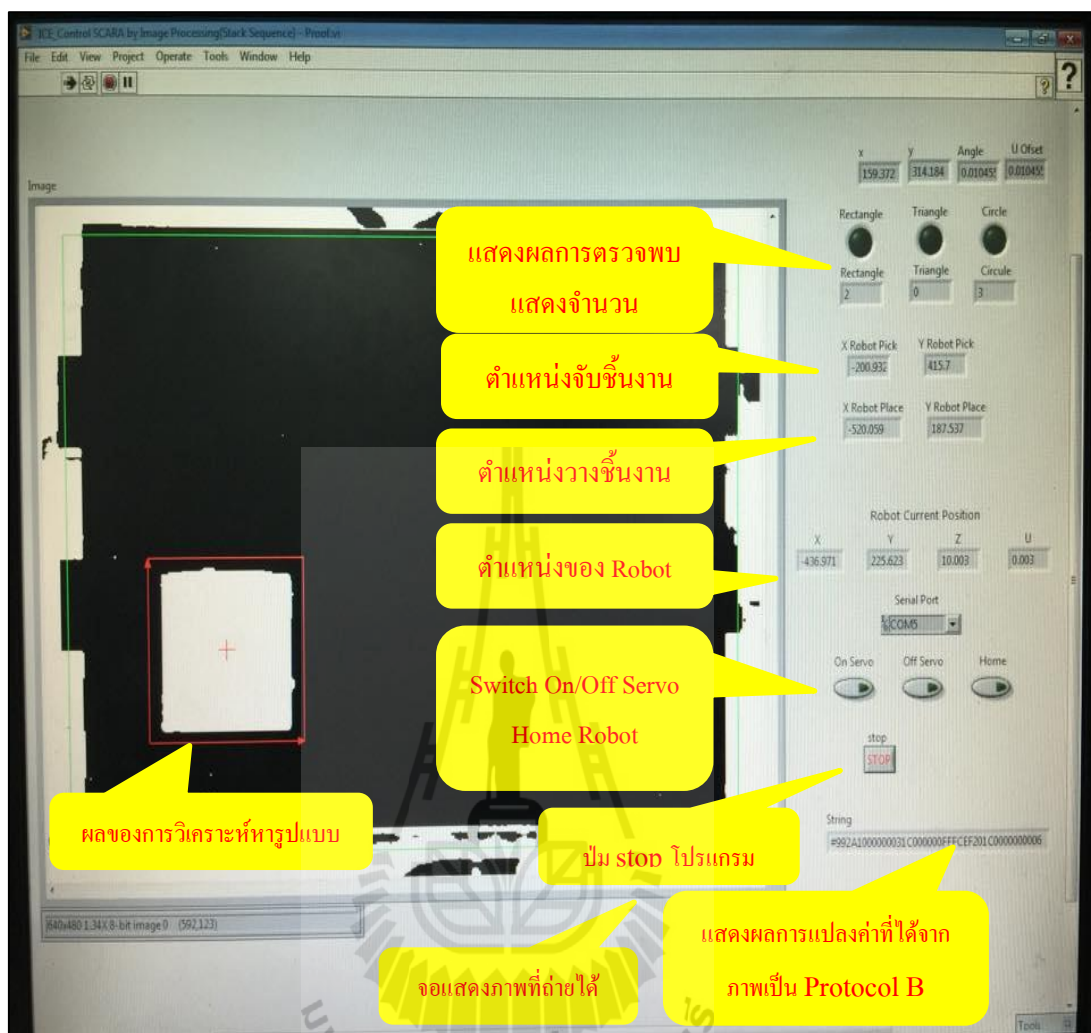
จากการกำหนดค่าในโปรแกรม NI Vision Builder โดยการกำหนดชนิดของกล้องที่ใช้กำหนดขอบเขตและพื้นที่ในการค้นหา กำหนดรูปแบบในการค้นหา กำหนดจุดกึ่งกลางของรูปแบบ และกำหนดความคมชัดค่าความเหมือนของรูปภาพในการค้นหา โปรแกรมก็จะให้ค่าต่าง ๆ ของเรามาตามที่เรากำหนดไว้ ดังรูปที่ 4.2



รูปที่ 4.2 ผลของการกำหนดค่า NI Vision Builder

#### 4.2.2 ผลการทดสอบโปรแกรม NI LabVIEW ที่ได้ออกแบบมา

จากการทำการเขียน โปรแกรมขึ้นมาเพื่อควบคุมให้เครื่องจักรอัตโนมัติทำงานตามขั้นตอนที่วางไว้ ซึ่งได้ใช้โปรแกรมในการประมวลผลทาง Vision ของโปรแกรม LabVIEW ในการประมวลผล สามารถแสดงลักษณะของโปรแกรมใน Front Panel โดย Front Panel จะแสดงผลส่วนที่เป็นพิกัดของหุ่นยนต์ พิกัดของการหยิบและวางชิ้นงาน ปุ่มควบคุมการทำงานต่าง ๆ



รูปที่ 4.3 Front Panel ของเครื่องตรวจสอบฯ

จากการเขียนโปรแกรมเราสามารถที่จะกำหนดหน้า Front Panel ได้โดยที่เราจะกำหนดให้หน้า Front Panel แสดงผลของสิ่งที่เราสนใจ อย่างเช่น สวิตช์ควบคุมการทำงาน ตำแหน่งของตัว Robot ตำแหน่งของการจับชิ้นส่วน ตำแหน่งของการวางชิ้นงาน และเราสามารถเลือกสิ่งที่เราต้องการจะให้แสดงบน Front Panel เพื่อโชว์สิ่งที่เราต้องการอยากให้เห็นได้อีกหลากหลาย

จากการเขียนโปรแกรม NI LabVIEW ที่ใช้ในการควบคุมการทำงานของระบบ เราต้องทำการทดสอบการทำงานต่าง ๆ เพื่อให้ได้ค่าที่เหมาะสมต่อการทำงาน ไม่ให้มากหรือน้อยจนเกินไป เราจึงต้องนำมาทดสอบในโปรแกรมที่จะใช้ในการทำงานจริงอีกครั้ง โดยจะทดสอบ

#### 4.2.2.1 การทดสอบหาค่าพารามิเตอร์ต่างๆที่เหมาะสม

ในการประมวลผลด้วยภาพสิ่งที่สำคัญและต้องควบคุมให้คงที่ก็คือระยะโฟกัส ค่าความคมชัด (Minimum Contrast) และค่าความเหมือน (Minimum Match Score) ซึ่งในการวิจัยนี้มีข้อจำกัดของกล้อง จึงทำให้ระยะของการโฟกัส จึงจำเป็นจะต้องกำหนดตามความสามารถของกล้อง ที่จะสามารถถ่ายภาพให้ครอบคลุมบริเวณที่ต้องการจะทดสอบได้ ในงานวิจัยนี้จึงควบคุมระยะโฟกัสระหว่างกล้องและตัวชิ้นงานคงที่อยู่ที่ 30 เซนติเมตร เพื่อใช้ในการตรวจสอบชิ้นงานหาค่า Minimum Contrast และค่า Minimum Match Score ไม่เหมาะสมก็อาจจะทำให้การตรวจสอบเกิดความคลาดเคลื่อนได้ จึงต้องทำการทดสอบเพื่อหาค่า Minimum Contrast และค่า Minimum Match Score ที่เหมาะสมที่จะใช้ในการทดสอบชิ้นงานที่ได้ออกแบบมา ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 แสดงผลการทดสอบหาค่าพารามิเตอร์ต่าง ๆ ที่เหมาะสมกับการประมวลผลด้วยภาพ

ส่วนที่ ต้องการ ตรวจสอบ	ค่า Minimum Match Score	ความถูกต้องในการตรวจสอบ (%) ที่ค่า Minimum Contrast			
		20	25	30	35
วงกลม	550	60	70	100	100
	600	70	80	100	100
	650	30	40	50	40
สามเหลี่ยม	550	70	80	100	80
	600	60	60	80	80
	650	60	60	80	70
สี่เหลี่ยม	550	80	80	100	100
	600	70	80	90	100
	650	70	70	70	80

จากผลการทดสอบหาค่าพารามิเตอร์ต่าง ๆ สามารถเลือกค่า Minimum Match Score และ Minimum Contrast เราได้กำหนดค่า Minimum Match Score จากความเหมือนมากที่สุดที่ 1000 ปรากฏว่ามีการตรวจสอบพบรูปแบบน้อยมาก เราจึงลดค่าความเหมือนลงมาไม่ให้น้อยเกินไป ซึ่งจะอยู่ในช่วงระหว่าง 550-650 และค่า Minimum Contrast ที่เหมาะสมสำหรับการตรวจสอบการตรวจจับรูปแบบต่างๆ ได้ดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 ค่า Minimum Contrast และค่า Minimum Match Score

พารามิเตอร์ที่ใช้ควบคุม	ส่วนที่ต้องการตรวจสอบ		
	วงกลม	สามเหลี่ยม	สี่เหลี่ยม
Minimum Match Score	600	550	550
Minimum Contrast	30	30	30

#### 4.2.2.2 ผลการทดสอบเวลาที่ใช้ในการตรวจสอบด้วยภาพ

ในการตรวจสอบด้วยภาพนี้หากกล้องสามารถถ่ายภาพและส่งข้อมูลเข้าไปยังโปรแกรมสำหรับการประมวลผล จะใช้เวลาในการตรวจสอบน้อยมาก แต่เนื่องจากรูปแบบชิ้นงานมีการเคลื่อน ที่กล้องจะไม่สามารถถ่ายภาพได้ทันที จึงจะต้องให้ชิ้นงานหยุดนิ่งเสียก่อน จึงต้องมีการทดสอบเพื่อหาเวลาในการถ่ายภาพที่เหมาะสม โดยการกำหนดระยะเวลาในการค้นหาที่ 0.5, 1, 1.5, 2, 2.5 วินาที ซึ่งเวลาที่เหมาะสมนั้นก็ไม่ว่าจะมากหรือน้อยจนเกินไป เพราะถ้าน้อยไปก็อาจจะจับภาพได้ไม่ทัน ถึงแม้จะใช้เวลามากแล้วจะจับภาพได้ตลอดทุกครั้งแต่มันก็จะทำให้ใช้เวลานาน ในการทำงานของเครื่องจักรดังแสดงในตารางที่ 4.4 โดยคำนวณจาก

$$\frac{\text{จำนวนการทดสอบทั้งหมด} - \text{จำนวนการทดสอบที่ไม่พบ}}{\text{จำนวนการทดสอบทั้งหมด}} \times 100 \quad (4.1)$$

ตารางที่ 4.4 แสดงผลการทดสอบเวลาที่เหมาะสมสำหรับการถ่ายภาพ

การวิเคราะห์	เวลาที่ใช้ในการถ่ายภาพ (วินาที)				
	0.5	1	1.5	2	2.5
ความถูกต้องในการประมวลผล (%)	6.666	30	50	93.333	93.333

จากผลของการทดสอบหาค่าเวลาที่เหมาะสมในการถ่ายภาพและตรวจจ็บบรูปแบบ จากค่าของการทดสอบทั้งหมดในตารางนั้น ณ เวลา 2 วินาที และ 2.5 วินาที จะมีความถูกต้องและแม่นยำที่เท่าเทียมกันจึงเลือกใช้เวลาที่จะใช้ในการรอผลการประมวลผลภาพที่มีเวลาที่น้อยกว่า เพื่อที่จะถ่ายภาพหลังจากที่ชิ้นงานได้หยุดอยู่นิ่งอยู่ที่ในตำแหน่งที่พร้อมจะถ่ายภาพแล้วเป็นเวลา 2 วินาที

#### 4.2.2.3 ผลจากการตรวจสอบและบอกพิคต์ของชิ้นงาน

ผลในการตรวจสอบภาพชิ้นงานจะต้องใช้เวลาในการตรวจสอบการหารูปแบบของชิ้นงาน ซึ่งรูปแบบนั้นก็ยังคงมีความแตกต่างกันของรูปแบบและลักษณะของรูปแบบ ไม่ว่าจะเป็นตัวรูปแบบ วงกลม สี่เหลี่ยม สามเหลี่ยม และยังมีแนวการวางตัว มุมหรือความเอียงของรูปแบบต่างๆ และยังต้องใช้เวลาในการหาจุดกึ่งกลางของรูปแบบ จึงทำให้เสียเวลาในการระบุตำแหน่งจุดกึ่งกลางของรูปแบบที่จับภาพได้ เพื่อทำการบอกพิคต์ไปยังโปรแกรมส่วนต่อไปให้นำไปใช้ในการประกอบชิ้นงาน ดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 แสดงเวลาในการตรวจสอบด้วยภาพเพื่อบอกพิคต์

ชิ้นงาน ทดสอบ	จำนวนครั้งที่ ทดสอบ	เวลาเฉลี่ยในการตรวจสอบและบอกพิคต์ถูกต้อง ( วินาที )		
		วงกลม	สามเหลี่ยม	สี่เหลี่ยม
1	30	2.766	2.814	2.852
2	30	2.698	2.851	2.765
3	30	2.752	2.740	2.716
เฉลี่ย		2.738	2.801	2.777

จากผลการทดสอบการตรวจจ็บบชิ้นงานด้วยภาพ จากชิ้นงานทั้ง 3 รูปแบบ เวลาที่ใช้การตรวจสอบแต่ละรูปแบบก็มีเวลาที่แตกต่างกันด้วย จะเห็นได้ว่าการตรวจสอบรูปภาพก็ยังคงใช้เวลาในการตรวจสอบ เพื่อให้ได้มาของผลที่ถูกต้องและแม่นยำ โดยการตรวจสอบรูปวงกลมใช้เวลาเฉลี่ยประมาณ 2.738 วินาที รูปสามเหลี่ยมใช้เวลาเฉลี่ยประมาณ 2.777 วินาที รูปสี่เหลี่ยมใช้เวลาเฉลี่ยประมาณ 2.801 วินาที จึงทำให้เวลาที่ใช้ในการวิเคราะห์จุดกึ่งกลางของรูปแบบและส่งข้อมูลไปยังโปรแกรมส่วนต่าง ๆ ต่อไป ก็จะมีควมล่าช้ามากขึ้น

### 4.3 ผลการประกอบชิ้นส่วน

การประกอบชิ้นส่วนนั้นจะต้องมีเวลาที่ใช้ในหยิบชิ้นส่วนจาก Store เพื่อใช้ในการประกอบ และยังมีพิกัดที่จะหยิบชิ้นส่วนในการประกอบที่แตกต่างกัน จะทำให้ใช้เวลาที่จะไปหยิบแตกต่างกันด้วย ซึ่งจุดที่ใช้ในการหยิบชิ้นงานในการประกอบจะมีชิ้นส่วนในการประกอบ ของทั้ง 3 รูปแบบ โดยจะมีพิกัดในการไปหยิบชิ้นส่วนที่แตกต่างกัน และยังรวมไปถึงพิกัดในการนำชิ้นส่วนเข้าไป ประกอบกับชิ้นงานที่ตรวจพบ ก็ยังมีพิกัดและมุมที่แตกต่างออกไปอีกด้วย จึงทำให้เสียเวลาในการ ประกอบมากขึ้น จึงทำให้เวลาในการประกอบแต่ละครั้งมีความแตกต่างกัน

#### 4.3.1 ตำแหน่งของการหยิบชิ้นส่วนเพื่อนำมาประกอบ

จากการออกแบบส่วนของ Store จะมีตำแหน่งของชิ้นส่วนที่จะนำมาประกอบที่มีความแตกต่างกันของชิ้นส่วน เพื่อนำมาประกอบแต่ละรูปแบบจะมีผลต่อเวลาในการหยิบชิ้นส่วน เพื่อนำมาประกอบ เราจึงได้ทำการทดลองการหยิบชิ้นส่วนแต่ละตำแหน่งเพื่อมาวางยังตำแหน่งที่เรา กำหนดไว้ก่อนหน้านี้

การเคลื่อนที่ไปหยิบชิ้นงานเพื่อนำมาประกอบ Robot จะเคลื่อนที่จากจุดเริ่มต้น เพื่อไปหยิบชิ้นส่วนจาก Store ซึ่งมีตำแหน่งที่แตกต่างกัน จะทำให้ใช้เวลาในการเคลื่อนที่ไปหยิบที่ แตกต่างกันด้วยดังนั้นจะทำให้ใช้เวลาในการหยิบแตกต่างกัน ดังแสดงในตารางที่ 4.6

ตารางที่ 4.6 ตารางแสดงเวลาการหยิบชิ้นงาน

วงกลม	รอบ	ตำแหน่งในการหยิบ				
		1	2	3	4	5
	1	4.55	5.18	5.25	5.47	5.61
	2	4.41	4.98	5.17	5.51	5.57
	3	4.64	5.23	5.23	5.4	5.65
	4	4.58	5.23	5.27	5.58	5.71
เฉลี่ย		4.545	5.155	5.23	5.49	5.635
สามเหลี่ยม	รอบ	ตำแหน่งในการหยิบ				
		1	2	3	4	5
	1	4.9	5.58	5.62	5.45	5.51
	2	4.78	5.51	5.58	5.48	5.61
	3	4.61	5.49	5.68	5.5	5.59
	4	4.84	5.55	5.61	5.58	5.55
เฉลี่ย		4.692	5.532	5.622	5.502	5.565

ตารางที่ 4.6 ตารางแสดงเวลาการหยิบชิ้นงาน (ต่อ)

สีเหลี่ยม	รอบ	ตำแหน่งในการหยิบ				
		1	2	3	4	5
	1	4.49	4.9	5.1	5.55	5.71
	2	4.51	5.1	4.98	5.46	5.69
	3	4.35	4.83	5.21	5.6	5.75
	4	4.5	4.89	5.15	5.58	5.71
เฉลี่ย		4.462	4.93	5.11	5.547	5.715

จากตำแหน่งในการหยิบชิ้นงานเพื่อมาประกอบนั้น จะเห็นได้ว่ามีผลต่อเวลาที่ใช้ในการประกอบ เพราะแต่ละจุดในการหยิบชิ้นส่วนนั้นจะมีพิกัดที่แตกต่างกัน แต่ก็มีผลไม่มากสักเท่าไร เพราะจุดแต่ละจุดมีพิกัดของจุดห่างกันไม่มากเท่าไร จึงทำให้มีผลต่อเวลาในการทำงานไม่มากนัก

#### 4.3.2 ผลการทดสอบการทำงานของเครื่องจักรต้นแบบทั้งหมด

ผลการทดสอบเครื่องต้นแบบโดยเวลาที่ใช้ในการประกอบชิ้นส่วนทั้งหมด เริ่มต้นจากการส่งชิ้นงานออกจาก Store เข้าไปยัง Conveyer ไปจนถึงจุดจับชิ้นงาน เพื่อรอการประมวลผลของกล้องส่งคำสั่งไปยังโปรแกรม เพื่อคำนวณพิกัดของรูปแบบที่จับภาพได้ เพื่อสั่งให้ Robot เคลื่อนที่ไปหยิบชิ้นส่วนเพื่อนำมาประกอบเข้ากับชิ้นงาน การประกอบชิ้นงานแต่ละชิ้นส่วนก็ยังมีเวลาที่แตกต่างกันไปตามรูปแบบและลักษณะการวางตัวของชิ้นงานอีกด้วย เราจึงได้ทำการยกตัวอย่างรูปแบบของชิ้นงานมา 7 แบบ แต่ละรูปแบบของชิ้นงานจะมีแนวการวางตัวที่ต่างกันไปอย่างเช่น กิ่งกลางชิ้นงาน ด้านซ้ายบนของชิ้นงาน ซ้ายล่างของชิ้นงาน ขวาบน ขวาล่าง และรูปแบบที่เอียง เราจะนำเอาเวลาของรูปแบบของรูปสี่เหลี่ยมมายกตัวอย่าง เพราะรูปสี่เหลี่ยมนี้จะมีการทำมุมของรูปแบบด้วย เพื่อให้ได้เวลาที่ใกล้เคียงกับความจริงมากที่สุด ดังแสดงในตารางที่ 4.7



ตารางที่ 4.7 แสดงเวลาการทำงานการหยิบและประกอบชิ้นงาน

ตำแหน่ง Assembly Parts	ครั้ง	รูปแบบของชิ้นงาน					
		กึ่งกลาง	ซ้ายบน	ซ้ายล่าง	ขวาบน	ขวาล่าง	เอียง
1	20	8.96	9.08	8.66	8.78	8.98	9.14
2	20	9.13	9.12	8.85	9.05	9.12	9.25
3	20	9.17	9.23	9.13	9.08	9.23	9.32
4	20	9.25	9.31	9.6	9.18	9.27	9.54
5	20	9.37	9.35	10.1	9.23	9.32	9.86
เฉลี่ย		9.176	9.218	9.268	9.064	9.184	9.422

จากผลการทดสอบระบบการทำงานของเครื่องจักรต้นแบบที่ได้ทำการออกแบบ จากผลการตรวจสอบด้วยภาพ โดยการค้นหารูปแบบที่กำหนดไว้มีความถูกต้อง ก่อ้งสามารถ ค้นพบรูปแบบ และโปรแกรมสามารถบอกพิกัดในการประกอบชิ้นส่วนได้อย่างถูกต้อง จึงจะทำให้ การประกอบชิ้นงานได้อย่างถูกต้องตามแนวการวางตัวของชิ้นส่วนที่ตรวจสอบพบ

#### 4.3.3 Image Calibration

จากการทดสอบการทำงานของเครื่อง การกำหนดตำแหน่งของวัตถุด้วยภาพที่ถ่าย มาได้ เราพบว่าเมื่อกำหนด Coordinate ให้กับ Robot ของเราแล้ว ตำแหน่งที่ Robot ไปนั้น ไม่ตรงกับตำแหน่งของวัตถุจริง จากการวิเคราะห์พบว่าปัญหานี้จะเกิดจากรูปถ่ายที่ได้จากกล้อง เนื่องจาก กล้องไม่ได้วางที่จุดกึ่งกลางภาพและมีแนวตั้งฉากกับภาพ ทำให้รูปที่ได้เกิด Distortion และเมื่อ คำนวณหาตำแหน่งจะเกิดการผิดพลาดขึ้นได้ ดังนั้นเราจึงทำการ calibrate ภาพถ่ายที่ได้จากกล้อง เสียก่อน จะเห็นได้ว่าจุดใดที่มีการ Error มากที่สุด ดังแสดงในตารางที่ 4.8

จากการทดลอง เราพบว่าค่าความผิดพลาดทางแกน x จะสูงที่สุดที่  $\varepsilon_x = 10.59\%$  สำหรับวงกลมหมายเลข 41 และค่าทางแกน y จะผิดพลาดมากที่สุดที่  $\varepsilon_y = 7.14\%$  ที่วงกลม หมายเลข 2 แต่ถ้าหากค่าความผิดพลาดรวม จากสมการ

$$\varepsilon_T = \sqrt{\varepsilon_x^2 + \varepsilon_y^2} \quad (4.2)$$

จะได้ว่าความผิดพลาดเกิดขึ้นมากที่สุดเท่ากับ 10.621% ที่วงกลมหมายเลข 41 สำหรับตารางต่อไปนี้ แสดงวงกลมที่เกิดความผิดพลาดรวมมากที่สุด 10 อันดับแรก ดังแสดงในตารางที่ 4.8

ตารางที่ 4.8 แสดงจุดที่มี Error มากที่สุด

Number	X error ( $\epsilon_x\%$ )	Y error ( $\epsilon_y\%$ )	Total error ( $\epsilon_T\%$ )
41	10.599	0.771	10.627
1	4.164	6.672	7.865
2	1.295	7.142	7.258
3	0.759	6.907	6.949
9	5.676	2.829	6.342
33	6.061	0.863	6.122
25	5.374	1.491	5.577
6	0.566	5.141	5.172
10	1.643	4.347	4.647
4	0.428	4.478	4.499

จากตารางจะเห็นว่าค่าความผิดพลาดจะเกิดขึ้นมากบริเวณขอบของรูป ทั้งนี้เนื่องจากเกิด Distortion ของภาพทำให้ภาพที่ได้มีลักษณะเป็นภาพ Perspective ไม่ได้เป็นภาพที่มองแบบ Perpendicular

#### 4.3.4 ค่าที่ได้จากการ Calibration

จากการทดสอบเครื่องจักรในครั้งแรกเราได้พบการวางตำแหน่งในการประกอบของชิ้นงาน ยังไม่สามารถที่จะประกอบได้อย่างถูกต้อง เราจึงจะนำค่าที่ได้จากการ Calibration ที่มีค่า Error มาชดเชยค่า เพื่อประมวลผลหาค่าพิกัดในการประกอบชิ้นงาน เพื่อให้ได้พิกัดในการประกอบที่ถูกต้องและแม่นยำมากขึ้น ซึ่งผลที่ได้จากการชดเชยค่าเราก็จะได้ผลที่นำมาเปรียบเทียบระหว่างค่าที่ได้ก่อนทำการชดเชยและค่าที่ได้หลังการชดเชย เราจะยกตัวอย่างลักษณะที่เป็นรูปแบบวงกลมมา ดังแสดงในตารางที่ 4.9

ตารางที่ 4.9 แสดงค่าพิกัดก่อนชดเชยและหลังชดเชย

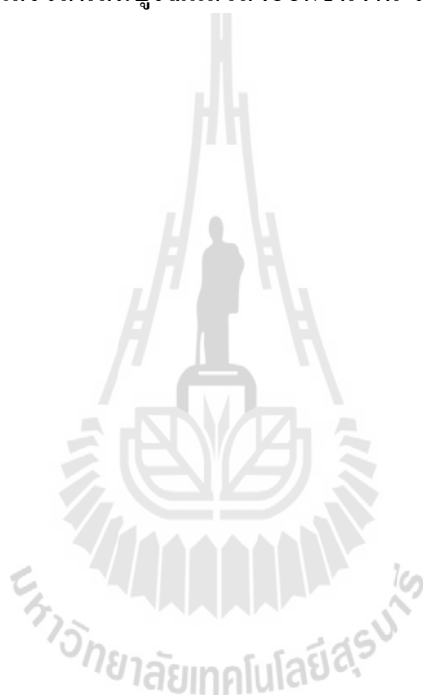
วงกลมกึ่งกลางชิ้นงาน					วงกลมมุมขวาบนของชิ้นงาน				
	ก่อน Calibration		หลัง Calibration			ก่อน Calibration		หลัง Calibration	
ครั้ง	X	Y	X	Y	ครั้ง	X	Y	X	Y
1	-504.33	184.07	-504.08	182.45	1	-489.569	159.02	-488.88	158.11
2	-504.36	183.97	-504.11	182.34	2	-489.64	159.03	-488.951	158.18
3	-504.29	184.09	-504.05	182.47	3	-489.751	159.02	-489.063	158.16
4	-504.62	183.45	-503.36	181.87	4	-489.959	159.02	-489.272	158.14
5	-504.32	184.02	-504.07	182.40	5	-489.464	158.97	-488.773	158.13
6	-503.91	183.47	-503.65	181.87	6	-489.481	159.08	-488.792	158.24
7	-504.72	184.08	-504.48	182.43	7	-489.434	159.08	-488.745	158.24
8	-504.29	183.98	-504.04	182.36	8	-489.62	159.01	-488.931	158.16
9	-503.68	183.51	-503.42	181.93	9	-489.789	158.98	-489.1	158.12
10	-503.68	183.51	-503.42	181.93	10	-489.424	159.03	-489.731	158.19

จากข้อมูลที่ได้อจากการ Calibration นี้จะทำให้เราสามารถที่จะหาค่า Parameter ที่ใช้ในการเปลี่ยนแปลงตำแหน่งในรูปภาพ ให้มาเป็นตำแหน่งจริงทางกายภาพได้แม่นยำมากขึ้น และหลังจากที่เราได้ทำการ Calibrated โปรแกรมด้วย Parameter ดังกล่าว เราพบว่าหุ่นยนต์ของเราสามารถเคลื่อนที่ไปยังที่ตำแหน่งของวัตถุได้อย่างถูกต้องและแม่นยำ และสามารถที่จะวาง (Place) ชิ้นส่วนของชิ้นงานได้อย่างถูกต้องและแม่นยำมากขึ้น จึงจะทำให้เครื่องจักรต้นแบบที่ได้ออกแบบมาสามารถที่จะทำงานได้อย่างถูกต้อง และสมบูรณ์แบบมากขึ้นกว่าในตอนแรกที่ยังไม่มีการ Calibration จึงทำให้การ Calibration มีส่วนที่ส่งผลต่อการทำงานของระบบในการบอกพิกัดในการประกอบชิ้นงานอีกด้วย

#### 4.4 สรุปผลการทดสอบ

จากการทดสอบทั้งสองระบบของเครื่องจักรต้นแบบนั้น ระบบการทำงานของชุดกลไก ลำเลียงชิ้นงานอัตโนมัติสำหรับการลำเลียงชิ้นงาน และชุดจับยึดชิ้นงานเพื่อรอการตรวจสอบนั้น ชิ้นงานชิ้นแรกจะใช้เวลาประมาณ 20 วินาที แต่ชิ้นต่อ ๆ ไปจะใช้เวลาในการไหลไปจนถึงจุดที่จะจับชิ้นงานเวลาประมาณ 5 วินาที เพราะชิ้นงานชิ้นแรกเป็นชิ้นเริ่มต้นของการทำงาน ชิ้นงานจะไหล

ออกจาก Store จะมีระยะทางที่ไกลกว่าจะถึงจุดจับชิ้นงาน แต่ชิ้นต่อ ๆ ไปจะไหลออกมาจาก Store ต่อเมื่อมีการจับชิ้นงานเพื่อรอตรวจสอบของชิ้นงานแรก จึงจะทำให้มีระยะเวลาที่ลดลง เวลาที่ใช้ในการตรวจสอบด้วยภาพเพื่อหารูปแบบของชิ้นงานจะใช้เวลาประมาณ 2 วินาที/ชิ้น เวลาที่ใช้ในการเคลื่อนที่ไปหยิบชิ้นส่วนในการประกอบจะมีเวลาที่แตกต่างกันตามแต่ละรูปแบบของชิ้นงาน แต่ละพิกัดของชิ้นงานที่จะนำไปประกอบ ซึ่งที่จุดต่าง ๆ ใช้เวลาเฉลี่ยประมาณ 5 วินาที/ชิ้น ส่วนเวลาการทำงานของเครื่องทั้งหมดนับจากเสร็จสิ้นการทำงานในชิ้นแรกโดยเฉลี่ยจะใช้เวลาประมาณ 6 วินาที/ชิ้น เพราะฉะนั้นการทำงานของระบบจากชิ้นงานไหลออกมาจาก Store จนถึงสิ้นสุดการทำงานจากระบบโดยที่มีการประกอบเสร็จสิ้นสมบูรณ์แล้วส่งออกชิ้นงาน จะใช้เวลาเฉลี่ยแล้ว 30 วินาที/ชิ้น



## บทที่ 5

### สรุปงานวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

งานวิจัยนี้เป็นการออกแบบและสร้างเครื่องจักรต้นแบบ เพื่อการออกแบบหุ่นยนต์แบบหยิบและวางที่ใช้ระบบการมองเห็นของเครื่องจักร ซึ่งเครื่องจักรต้นแบบนี้ได้ออกแบบให้สามารถบอกรูปแบบของชิ้นงานและพิกัดในการประกอบชิ้นส่วนต่าง ๆ โดยรูปแบบที่ได้กำหนด คือ วงกลม สามเหลี่ยม สี่เหลี่ยม ซึ่งเป็นรูปแบบที่แตกต่างกันและยังมีแนวการวางตัวที่หลากหลายอีกด้วย เพื่อใช้เป็นแนวคิดหรือใช้ปรับปรุงการทำงานของเครื่องจักรต้นแบบให้มีประสิทธิภาพมากขึ้นจากการออกแบบเครื่องจักรได้แบบเป็น 2 ระบบใหญ่ ๆ

1. ระบบกลไกการลำเลียงชิ้นงานอัตโนมัติ โดยส่วนของการลำเลียงชิ้นงานอัตโนมัติ นั้น จะใช้ PLC KEYENCE KV-700 โดยมี Module Input KV-B16XA และ Module Output KV-B16TA เป็นอุปกรณ์สั่งการ โดยสั่งการผ่าน Program KV-STUDIO เพื่อใช้ควบคุมการทำงานของชุดการลำเลียงชิ้นงานจาก Store ไปจนถึงจุดจับชิ้นงานเพื่อรอการตรวจสอบประมาณ 20 วินาที

2. การออกแบบระบบการมองเห็นของเครื่องจักร เราจะต้องแบ่งโปรแกรมออกเป็น 2 ส่วนเพื่อช่วยในการทำงานของเครื่องจักรต้นแบบ โดยจะมีส่วนของ Program NI Vision Builder, National Instrument LabVIEW

- 2.1 Program NI Vision Builder ใช้ในการกำหนด ชนิดของกล้องที่ใช้ในการค้นหา กำหนดรูปแบบในการค้นหา ซึ่งในงานวิจัยนี้เราได้กำหนด 3 แบบ ได้แก่ วงกลม สามเหลี่ยม สี่เหลี่ยม กำหนดความเหมือนของภาพที่ค้นหา กำหนดสีของภาพที่กล้องถ่ายได้เพื่อใช้ในการค้นหา กำหนดพื้นที่และขอบเขตในการค้นหา กำหนดตำแหน่งจุดกึ่งกลางของรูปแบบที่จะค้นหา ซึ่งโปรแกรมจะทำการตรวจสอบลักษณะของรูปแบบชิ้นงานที่ได้จากการถ่ายภาพ เพื่อทำการเปรียบเทียบกับรูปแบบที่มีการกำหนดไว้ก่อนหน้า และบอกตำแหน่งจุดกึ่งกลางของรูปแบบโดยการตรวจสอบพบว่าโปรแกรมได้มีการตรวจพบรูปแบบที่กำหนดไว้ 98 เปอร์เซ็นต์ โดยใช้เวลาเฉลี่ยในการตรวจสอบ 2 วินาที ของจำนวนการทดสอบทั้งหมด และยังสามารถบอกถึงตำแหน่งการประกอบชิ้นส่วนได้อย่างถูกต้องอีกด้วย โดยเราจะนำตำแหน่งที่ได้จากการตรวจสอบส่งเข้าไปยังโปรแกรมส่วนต่อไปเพื่อจะได้สั่งการให้ Robot เคลื่อนที่ไปหยิบชิ้นงานเพื่อนำมาประกอบ

2.2 National Instrument LabVIEW ใช้รูปแบบระบบการทำงานของเครื่องจักรต้นแบบ จะเริ่มทำงานเมื่อชิ้นงานถูกจับยึดที่ตำแหน่งที่ได้กำหนดไว้ โดยเริ่มสั่งการให้กล้องได้ทำการถ่ายภาพเพื่อให้ได้รูปแบบและตำแหน่งจุดกึ่งกลางของชิ้นงานที่ถ่ายภาพได้ เมื่อได้ตำแหน่งจุดกึ่งกลางของชิ้นงานมาแล้ว จะนำไปทำการคำนวณแปลงค่าตำแหน่งที่ได้ให้เป็นค่าจริงเพื่อการหยิบและการวางชิ้นงาน เมื่อได้ค่าที่เป็นจริงแล้ว โปรแกรมยังจะต้องแปลงค่าอีกครั้งให้อยู่ในชุดคำสั่งเพื่อสั่งการให้หุ่นยนต์เคลื่อนที่ไปหยิบชิ้นส่วนมาประกอบได้อย่างถูกต้องและแม่นยำตามแนวการวางตัวของชิ้นงาน โดยเฉลี่ยแล้วใช้เวลาในการทำงานของกระบวนการหนึ่งรอบการทำงาน 25 วินาที โดยเริ่มต้นจากชิ้นงานถูกส่งออกจาก Store จนถึงสิ้นสุดการทำงาน โดยที่มีการส่งออกชิ้นงานเมื่อมีการประกอบเสร็จสิ้นแล้ว

จากการออกแบบระบบการทำงานของเครื่องจักรต้นแบบนี้ทั้ง 2 ระบบ ระบบสามารถทำงานได้อย่างถูกต้องและแม่นยำ ไม่ว่าจะเป็นการส่งชิ้นงานออกจาก Store เข้าไปยังชุดลำเลียงชิ้นงาน (Conveyer) ซึ่งส่งออกชิ้นงานได้อย่างถูกต้องตามกระบวนการทำงานที่ได้กำหนดไว้ ส่วนของระบบการตรวจสอบการค้นหารูปแบบและบอกตำแหน่งจุดกึ่งกลางของรูปแบบ ระบบทำงานได้อย่างถูกต้อง ไม่ว่าจะเป็นการค้นหาตามรูปแบบที่ได้กำหนดไว้ก่อนหน้าหรือ การบอกตำแหน่งก็สามารถบอกตำแหน่งจุดศูนย์กลางของรูปแบบจากภาพถ่ายได้อย่างถูกต้อง โปรแกรมยังแปลงเป็นชุดคำสั่งเฉพาะเพื่อส่งคำสั่งไปยังหุ่นยนต์ เพื่อให้หุ่นยนต์เคลื่อนที่ไปหยิบชิ้นงานนำมาประกอบได้อย่างถูกต้องแม่นยำตามรูปแบบที่โปรแกรมสามารถจับภาพได้

## 5.2 ข้อเสนอแนะ

5.2.1 เนื่องจากเครื่องจักรต้นแบบเพื่อการออกแบบหุ่นยนต์แบบหยิบและวางที่ใช้ระบบการมองเห็นของเครื่องจักร มีความจำเป็นที่จะต้องใช้ภาพสำหรับการประมวลผล หากต้องการตรวจสอบเป็นระบบเวลาจริง (Real Time System) ระยะไฟกัสและความเข้มของแสงในการทำงานจึงมีผลต่อการประมวลผลด้วยภาพ เพื่อความถูกต้องและแม่นยำจะต้องควบคุมค่าพารามิเตอร์ต่าง ๆ เหล่านี้ให้คงที่อยู่เสมอ

5.2.2 ความละเอียดของกล้องที่นำมาใช้ ส่งผลต่อความแม่นยำในการตรวจสอบ หากนำกล้องที่มีความละเอียดสูงมาใช้งานความถูกต้องในการประมวลผลก็จะมากขึ้นด้วย และในงานที่ต้องการความแม่นยำสูงควรจะต้องเลือกใช้กล้องที่มีความละเอียดสูง แต่ราคาก็จะสูงขึ้นตามไปด้วย

5.2.3 การจัดวางตำแหน่งของกล้องเพื่อการตรวจสอบ ก็จะมีผลต่อการตรวจสอบและการบอกตำแหน่งในการประกอบด้วย จึงต้องจัดวางตำแหน่งให้เหมาะสม ควบคุมพื้นที่ในการตรวจค้นหา

5.2.4 ในการบำรุงรักษาอุปกรณ์ที่ใช้ในระบบนิวเมติก เพื่อให้ระบบทำงานอย่างมีประสิทธิภาพ ซึ่งปัญหาที่พบ เช่น มีน้ำที่จะเข้าไปค้างอยู่ในระบบ ซึ่งมีผลเสียโดยตรง ดังนั้นควรตรวจสอบระบบอย่างสม่ำเสมอ เพื่อไม่ให้มีน้ำเข้าไปค้างอยู่

5.2.5 เนื่องจากอุปกรณ์ต่าง ๆ ภายในเครื่องตรวจสอบนั้น ใช้ไฟฟ้ากระแสตรงที่ 24 v. และทำการแปลงไฟมาจากหม้อแปลง 220 v. ดังนั้นควรระมัดระวังในเรื่องของการเสียบปลั๊กจากแหล่งจ่ายไฟเข้าหม้อแปลง

5.2.6 สำหรับแนวคิดการประยุกต์ใช้ในการเรียนการสอนระดับห้องปฏิบัติการ อาจจะเพิ่มระบบกลไกในการนำชิ้นงานส่งออกมาตามสายพานลำเลียงมาใส่เข้าสู่ระบบ และระบบการจัดเก็บชิ้นงานที่ทำการประกอบเสร็จสิ้นแล้ว เพื่อการพัฒนาสู่ระบบการทำงานที่เป็นแบบอัตโนมัติที่สมบูรณ์มากขึ้นต่อไป



## รายการอ้างอิง

- กนต์ธร ธานีประศาสน์ การวัดเชิงกลด้วย LabVIEW. สาขาวิชาวิศวกรรมเครื่องกล สำนักวิชา วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
- Chen, T.Q., Murphey, Y.L., Zhang, J.X., and Zhou, Y.N. (2001) **A Smart Machine Vision System for PCB Inspection.** *IEA/AIE*.
- Guerra, E., and Villalobos, J.R. (2001) **A three-dimensional automated visual inspection system for SMT assembly.** *Computers & Industrial Engineering* 40. (pp.175-190).
- Kim, S.M., Lee, S.C., and Lee, Y.C. (2006) **Vision Based Automatic Inspection System for Nuts Welded on the Support Hinge.** *SICE-ICASE International Joint Conference 2006*.
- National Instruments. (2005) **NI-IMAQ for USB Cameras (online).** Available : [http://www.csun.edu/~rd436460/Labview/NIIMAQ\\_for\\_USB\\_Cameras\\_User\\_Guide.pdf](http://www.csun.edu/~rd436460/Labview/NIIMAQ_for_USB_Cameras_User_Guide.pdf)
- National Instruments. (2007) **NI Vision Assistant Tutorial (online).** Available : <http://www.ni.com/pdf/manuals/372228h.pdf>
- National Instruments. (2005) **NI Vision Concepts Manual (online).** Available : <http://www.ni.com/pdf/manuals/372916e.pdf>
- Prommarak, N., and Chamniprasart, K. (2010) **Damage Screw Inspection Computer Vision Prototype Machine.** *Proceedings of International Conference on Data Storage Technology (DST-CON) Bangkok, Thailand.*
- Wang, Z., Huan, Y., Ji, S., Yang, G., and Zhang, L. (2006) **Detecting assembling quality of Razor Based on the machine vision system.** *5th WSEAS*, pp. 49-52
- Watanabe, T., Fujiwara, T., Kusano, A., and Koshimizu, H. (2007) **3D Precise Inspection of Electronic Devices by Single Stereo Vision.** *MVA2007 IAPR Conference on Machine Vision Applications.*
- Youssef, S.M., and Salem, R.M. (2007) **Automated barcode recognition for smart identification and inspection automation.** *Expert Systems with Applications* 33, pp. 968-977.

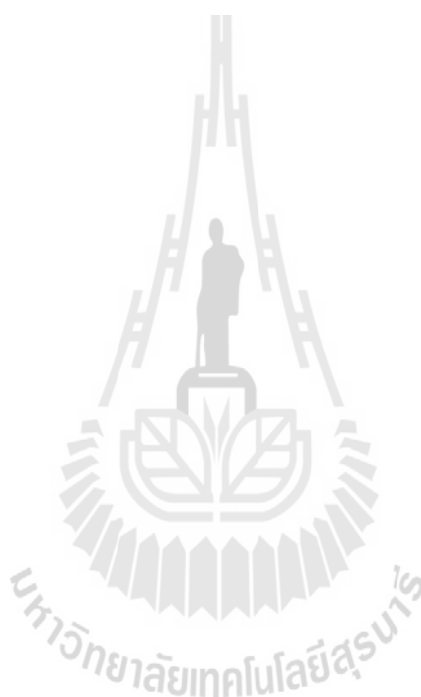


ภาคผนวก ก  
บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่



## รายชื่อบทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่

Phansak Nerakae, Pichitra Uangpairoj, Kontorn Chamniprasart. **Prototyping of Flexible Manufacturing System with Machine Vision.** The 2<sup>nd</sup> International Conference on Engineering Science and Innovative Technology (ESIT 2016), Phuket, Thailand, April 21 – 23 2016.



## Prototyping of Flexible Manufacturing System with Machine Vision

**Phansak Nerakae<sup>1</sup>, Pichitra Uangpairoj<sup>2</sup>, Kontorn Chamniprasart<sup>2,\*</sup>**

### Abstract

Automated manufacturing system using industrial robots are now widely used for manufacturing process in industry. Robots are generally used for pick and place process in an assembly process. To pick up and place things in the right place, the robot must be programmed to move to the precise position. However, when the production line is changed such as minor model changes, the robot must be reprogrammed in order to work with a new model correctly. This process is time consuming and the production line must be stopped. To overcome this problem, the prototype of flexible automated manufacturing process is introduced in this paper. A robot is integrated with machine vision systems. In this system, the machine vision system identifies the model at current location in the assembly line and then the machine vision system identifies the exact location of pick and place location for part assembly. LabVIEW is used as a machine vision engine that sends information of coordinates of pick and place location to robot controller. The robot then performs by moving end-effector to the right position for assembly process. The machine vision of flexible manufacturing system is utilized not only for model changes in the assembly process, but also for the cost reduction of jig and fixture of part holding. The robot can identify the exact location via machine vision system every time in assembly process, consequently the precision of jig and fixture is not required. The results show that the system works very well. The speed of production line significantly decrease compared with the old design system. The accuracy and precision of robot location are very good.

**Keywords:** flexible manufacturing system, machine vision systems, pick and place process, LabVIEW

---

<sup>1</sup> School of Manufacturing Engineering, Institute of Engineering, Suranaree University of Technology

<sup>2</sup> School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology

\* Corresponding author, E-mail: kontorn@sut.ac.th

## 1. Introduction

Currently, the factory automation industry has brought in to assist in the production process. To get enough yields serving the demands of the market, one production line has a lot of machines that are used to produce a single type of products. The important part of the process is assembly process [1]. The pick and place process, a sub process of assembly process, requires a precise location for assembled parts. The change in model leads to the change of part location. Each model is different in its size, position, and orientation, while it must be compatible with a number of different products [2]. When the product models are changed, the delay in production process is caused by machine adjustment, conveyor adjustment, and software modification. Consequently, the production costs are raised by the inverse in labor cost, machine and equipment cost [3].

This paper proposes the devices and method that are used to develop an effective inspection system. The human vision is simulated. The computer vision system is integrated with the division's work of a single production line to locate the assembly space of the work piece and identify the geometry of assembly space. Thus, a single production line can be performed effectively. This system can save time and cost for adjusting production lines. The system works through the following steps [4], as shown in Fig. 1.

1. Object Inspection – a main part flows along the conveyor belt and stops at the specific position for checking the geometry of assembling space.

2. Camera – the image of the part is captured using camera.

3. Image processor – in this step, the captured image is compared with the reference image in order to specify the geometry of the matched assembly spare location.

4. Controller – robot controller commands the robot to move and pick up the correct assembly part, then finally place the assembly part on the correct assembly space.

5. Scara robot – the robot performs all instruction controlled by the control system.

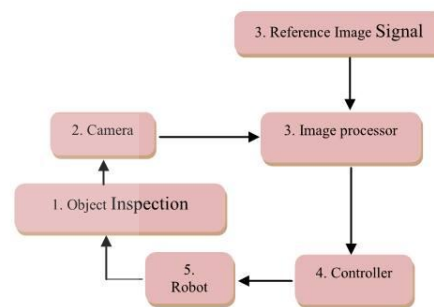


Fig. 1. Schematic Block diagram of Object Inspection System.

## 2. Experiment

Vision systems are designed to identify model component model and the orientation of components. In this experiment, model components consist of (1) A main part which flow on the conveyor belt, and (2) An assembly part which is stored on the storage station. The assembly process must be done correctly in order to complete the production. The design stage is divided into two stages which are (1) Hardware design and (2) software.

The system composed of hardware design composes of Camera, robot, and end effector of robot.

Vision Module uses the USB CCD Camera with the resolution of 640 X 840 pixels to capture the main part on the conveyor belt. The brightness of the backlight at 700 Lux makes it is easier to create image processing software of LabVIEW.

We use Scara Robot model IX-NNC6020-5L-T1-SP to receive processing information. Then it is programmed to move to the desired location.

The end effector of the robot, a 10 mm diameter vacuum suction head, is used to hold the parts in assembly step.

The software consists of LabVIEW NI Vision, and NI Vision Builder.

The system uses the LabVIEW NI vision software to develop computer programs for detection. The program can tell the position and orientation of the parts. LabVIEW is also used for communication between the robot controllers by sending part location to the robot.

NI Vision Builder program is brought in to define the format of the program such as the configuration of the objects that needs to be searched, color, brightness, area search, etc. This vision builder helps the program work easily.

**2.1 The positioning of the device**

Robot is placed at the center of the base. A camera mounted on the tip of the robot. The work pieces the target location, flow along the tracks conveyer. When they reaches the jig is assigned to hold a workpiece. The conveyer is stoped to capture images using camera, as shown in Fig. 2.

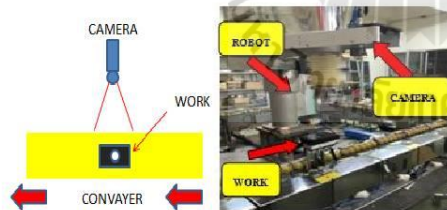


Fig. 2. The position of the workpiece stop and camera.

**2.2 Algorithm**

The first pictures of the parts are taken for processing the images using high-resolution cameras. The vision system uses a compact storage system to import images from the camera to the LabVIEW software to start monitoring. To create the application, we have to add and confit process monitoring. In the process of checking, the image with the image acquisition will be simulated by loading images from a file. Data from file, will there be a model, framework of the model and the mid-point of the model, there is also a selection process match the captured image with the reference image, as shown in Fig. 3.

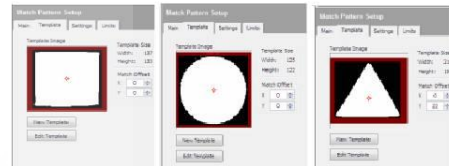


Fig. 3. Models assembly space and assembly part.

The algorithm comprises the various analyzing and processing functions illustrated in Fig. 4.

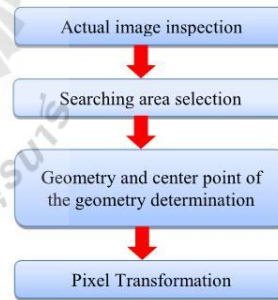


Fig. 4. Algorithm chart for object inspection.

### 2.2.1 Actual image to inspection

Fig. 5 shows the image of the main part is captured by visual inspection. The resolution size of the image is 640 X 480 pixels.

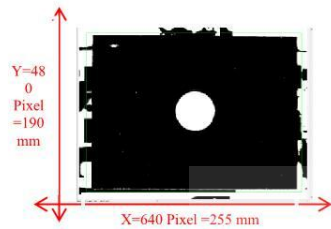


Fig. 5. Example of the captured image.

### 2.2.2 Searching area selection

The searching area is important. It defines the boundaries of the search. The position and orientation of the object that we are looking for must be in the searching area, otherwise we cannot identify the right position for the robot end-effector, as shown in Fig. 6.

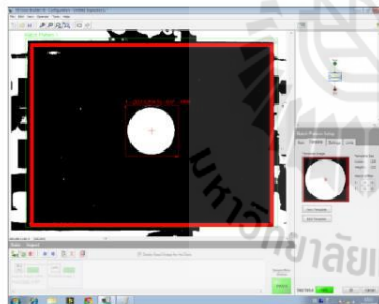


Fig. 6. Searching area determination.

### 2.2.3 Geometry and center point of the geometry determination.

In this step LabVIEW is commanded to search for geometry of assembly. Moreover, we have to set up the form and the geometry, framework of geometry, mid-point of geometry, of the assembly that we need to get from the storage station next, the program tells the halfway point of what we are looking for. The program inform the pixel along the X axis and Y axis of the photo, shown in Fig. 7.

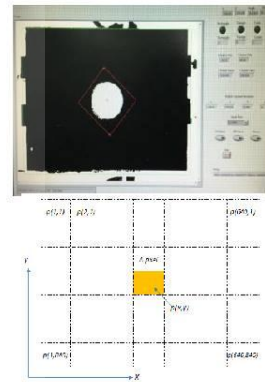


Fig. 7. The captured images format and a specific pixel of the image.

### 2.2.4 Pixel Transformation.

When we have the pixel coordinates from the previous step, we need to be convert the pixel coordinates into the coordinates in millimeter using conversion factor of 0.138425 mm/pixel in x-direction and 0.1351095 mm/pixel in y-direction.

After that, we have to convert the coordinates in millimeter into a series of commands of IAI Protocol B that is used to control the robot movement in the pick and place process.



### 3. Results and analysis

The combination system of vision system, LabVIEW, and scara robot can work very well. LabVIEW told the nature of the geometry, mid-point of the geometry, on the assembly space on the main part. Is detected and is matched with the reference geometry correctly, and can be specified the position of assembly space in the form of coordinates (x, y) correctly as presented in Fig. 8. Finally, using the command set Protocol B, the robot can pick up the correct assembly part and place the assembly part on the main part perfectly.

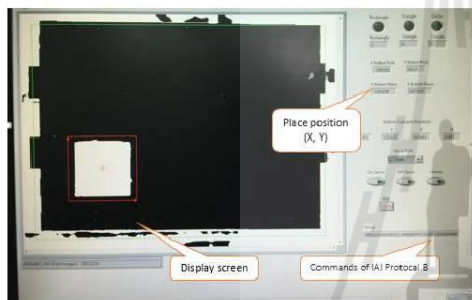


Fig. 8 Processing of the LabVIEW

### 4. Conclusion

The prototype design in this research shows that the integration of machine vision system with industrial robot can introduce the new flexible manufacturing systems. This machine vision for flexible manufacturing system not only helping when model change in assembly process but also help in mixed manufacturing system where variety of model can be assembly in a single production line. This research shows that this concept can be used in the industrial manufacturing system in the future.

The program can capture the form that has defined. When the image is detected in a file, we will have position midpoint of the axis (x, y) shown in Fig. 9. Then the

program will convert the midpoint to be. To command the robot to pick and place components correctly according to the position.

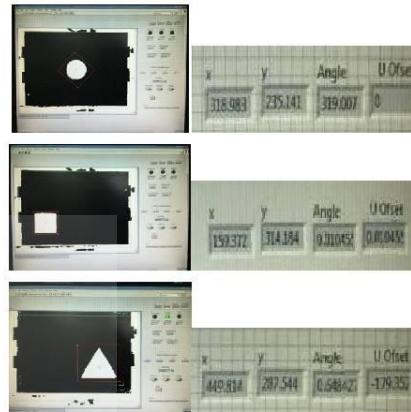


Fig. 9 The captured images format (left) and the coordinates X, Y at the center of the assembly space (right).

### 5. Reference

- [1] L. W. Teck, M. Sulaiman, H. N. M. Shah and R. Omar, "Implementation of Shape - Based Matching Vision System in Flexible Manufacturing System", Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Karung Berkunci 1752, Pejabat Pos Durian Tunggal, 76109. Durian Tunggal, Melaka.
- [2] Rakesh.N, Pradeep Kumar.A, Ajay.S, "Design and Manufacturing of Low Cost Pneumatic Pick and Place Robot", International journal of scientific & technology research volume 2, issue 8, august 2013.
- [3] U. SaburaBanu, M. Padma Usha, M. Ramamoorthy b, "LabVIEW based dimension detection and segregation of good and faulty crank pins", International Journal of Emerging Technology and

Advanced Engineering, Volume 3, Special Issue 1,  
January 2013).

- [4] Santosh Kumar Sahoo, B. B. Choudhury, "A Robotic Assistance Machine Vision Technique for an Effective Inspection and Analysis", International Journal of Electrical and Computer Engineering (IJECE) Vol. 5, No. 1, February 2015, pp. 46~54.





## ประวัติผู้เขียน

นายพันศักดิ์ เนระแก เกิดเมื่อวันที่ 21 สิงหาคม พ.ศ. 2529 ศึกษาชั้นประถมศึกษาที่โรงเรียนมารีย์วิทยา ชั้นมัธยมศึกษาตอนต้นที่โรงเรียนมารีย์วิทยา ชั้นมัธยมศึกษาตอนปลายที่ศูนย์การศึกษานอกโรงเรียน จังหวัดนครราชสีมา ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) วิทยาลัยเทคนิคนครราชสีมา และสำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมแมคคาทรอนิกส์ จากมหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อปี พ.ศ.2554

ปี พ.ศ. 2555 เข้าศึกษาต่อในระดับปริญญาโท วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมการผลิต สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา

ขณะที่ศึกษาต่อในระดับมหาบัณฑิตนั้น ได้มีประสบการณ์โดยเป็นผู้สอนในรายวิชาดังนี้

- 1) ปฏิบัติการวิศวกรรมเครื่องกล 1
- 2) ปฏิบัติการวิศวกรรมเครื่องกล 2
- 3) ปฏิบัติการวิศวกรรมการผลิต 1
- 4) ปฏิบัติการวิศวกรรมการผลิต 2

ในระหว่างการศึกษาปริญญาโท ได้นำเสนอผลงานทางวิชาการ ตามที่ได้นำเสนอแสดงในภาคผนวก ก