การควบคุมแขนกลสำหรับติดสติ๊กเกอร์บาร์โค้ดแบบอัตโนมัติ จากเครื่องพิมพ์บาร์โค้ด

AUTO STICK LABEL FROM ZEBRA PRINTER

สิทธิชัย เสียงสนั่น

รายงานสหกิจศึกษานี้เป็นส่วนหนึ่งในหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบการผลิต
วิทยาลัยนวัตกรรมการผลิตขั้นสูง
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมแขนกลสำหรับติดสติ๊กเกอร์บาร์โค้ดแบบอัตโนมัติจากเครื่องพิมพ์บาร์โค้ด
AUTO STICK LABEL FROM ZEBRA PRINTER

T147915

สิทธิชัย เสียงสนั่น

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ระบบ
เลขทะเบียน 147915
วัน.เดือน.ปี 6 ต.ค. 2560

b.12862319
i.

รายงานสหกิจศึกษานี้เป็นส่วนหนึ่งในหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบการผลิต
วิทยาลัยนวัตกรรมการผลิตขั้นสูง
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTO STICK LABEL FROM ZEBRA PRINTER

SITICHAI SEANGSANUN

A CO-OPERATIVE REPORT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MANUFACTURING SYSTEM ENGINEERING
COLLEGE OF ADVANCED MANUFACTURING INNOVATION
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การควบคุมแขนกลสำหรับติดสติ๊กเกอร์บาร์โค้ดแบบอัตโนมัติจากเครื่องพิมพ์บาร์โค้ด Auto Stick Label from Zebra Printer
นักศึกษา	นายสิทธิชัย เสียงสนั่น
รหัสนักศึกษา	56120037
ปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมระบบการผลิต
พ.ศ.	2559
อาจารย์ที่ปรึกษา	ผศ.ดร. วรวุฒิ มรรคเจริญ

วิทยาลัยนวัตกรรมการผลิตขั้นสูง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้สหกิจศึกษาเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต ประจำปีการศึกษา 2559

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.วรวุฒิ มรรคเจริญ	
ผศ.ดร.อนรรฆพล แสนทน	
ดร. ฉัตรพล ภคศิริ	
นางสาวศลิษา เผือกเนียม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การควบคุมแขนกลสำหรับติดบาร์โค้ดแบบอัตโนมัติจากเครื่องพิมพ์บาร์โค้ด Auto Stick Label from Zebra Printer
ชื่อนักศึกษา	นายสิทธิชัย เสียงสนั่น รหัสนักศึกษา 56120037
ปริญญา	วิศวกรรมศาสตรบัณฑิต วิศวกรรมระบบการผลิต
สาขาวิชา	วิศวกรรมระบบการผลิต
ปีการศึกษา	2559
อาจารย์นิเทศ	ผศ.ดร. วรวุฒิ มรรคเจริญ
ผู้นิเทศงาน	นางสาวศลิษา เผือกเนียม
ชื่อสถานประกอบการ	บริษัทแคล-คอมพ์ อิเล็คโทรนิคส์ (ประเทศไทย) จำกัด (มหาชน)

บทคัดย่อ

การแข่งขันทางการค้าของแต่ละอุตสาหกรรม มีการแข่งขันที่สูงขึ้น แต่ละอุตสาหกรรมจึงจำเป็นต้องมีการลดต้นทุนการผลิต และหาแนวทางในการเพิ่มประสิทธิภาพของการผลิตให้ได้มากยิ่งขึ้น เพื่อตอบสนองความต้องการของตลาด ดังนั้น การผลิตสินค้าให้ได้ปริมาณที่มากขึ้น มีคุณภาพและมาตรฐานในการผลิตที่มากขึ้น โรงงานอุตสาหกรรมส่วนใหญ่จึงเลือกที่จะใช้ระบบการผลิตแบบอัตโนมัติในการช่วยเพิ่มประสิทธิภาพในการผลิต และทดแทนการใช้แรงงานมนุษย์ที่มีข้อจำกัดในด้านต่างๆ ทั้งยังเป็นการลดต้นทุนอีกทางหนึ่ง

เครื่องติดสติ๊กเกอร์บาร์โค้ดแบบอัตโนมัตินี้ เป็นหัวข้อที่ผู้วิจัยให้ความสนใจ ศึกษาและจัดทำขึ้น โดยเครื่องนี้ จะประกอบไปด้วยการทำงานร่วมกันแบบอัตโนมัติ ระหว่างแขนกลอัตโนมัติ ระบบนิวเมติกส์ (Pneumatics) และระบบวิทัศน์ (Vision) ซึ่งจะควบคุมการเคลื่อนที่ของฝาครอบฮาร์ดดิสก์ด้วยสายพาน และมีการควบคุมตำแหน่งการหยุดของฝาครอบฮาร์ดดิสก์ด้วยตัวหยุด(Stopper) โดยใช้ระบบนิวเมติกส์ (Pneumatics) และใช้แขนกลในการหยิบสติ๊กเกอร์จากเครื่องพิมพ์สติ๊กเกอร์บาร์โค้ด (Zebra Printer) สำหรับติดบนฝาครอบฮาร์ดดิสก์ ซึ่งมีการใช้ระบบวิทัศน์เข้ามาช่วยในการปรับแก้ตำแหน่งการหยิบสติ๊กเกอร์บาร์โค้ดจากเครื่องพิมพ์ โดยจะเกิดการทำงานต่อเนื่องแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Auto Stick Label from Zebra Printer		
Student	Mr.Sitichai Seangsanun	Student ID	56120037
Degree	Bachelor of Engineering		
Program	Manufacturing System Engineering		
Year	2016		
Advisor	Dr.Worawut Makcharoen		
Mentor	Ms.Salisa Phuakneam		
ชื่อสถานประกอบการ	Cal-Comp Electronics (Thailand) Public Co., Ltd.		

ABSTRACT

The high competition trading in each industry. The industrial need to decrease the cost of manufacturing. And looking for way to increase the manufacture performance according to market demand. Due to the product quantity increase and need to have high quality standard manufacturer. Most of the manufacturers change to use an automated system for the higher performance process, replace the human resources and reduce cost of manufacturing.

Auto stick label from zebra printer machine is the topic that I have interested in. This machine is equipped with collaboration between robots, pneumatic and vision. Therefore this project works about control movement of Hard disk's lower case by conveyor and stopper. Use vision for locating the label from printer, then control robot to pick up sticker from printer and stick at the lower case. In addition, this machine can works continuous.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่ข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษา ณ บริษัทแคล-คอมป์ อีเล็กทรอนิกส์ (ประเทศไทย) จำกัด (มหาชน) ตั้งแต่วันที่ 10 สิงหาคม 2559 ถึง 5 ธันวาคม 2559 ส่งผลให้ข้าพเจ้าได้รับความรู้ และประสบการณ์ต่างๆ อันก่อให้เกิดประโยชน์ และสามารถนำไปประยุกต์ใช้ได้ในอนาคต

สำหรับรายงานวิชาสหกิจศึกษาฉบับนี้ สำเร็จลงได้ด้วยดี เนื่องมาจากความกรุณา และได้รับความร่วมมือในการสนับสนุนจากหลายฝ่าย ทั้งเพื่อนๆ รุ่นพี่จากบริษัทแคล-คอมป์ อีเล็กทรอนิกส์ (ประเทศไทย) จำกัด (มหาชน) ที่ได้ให้คำแนะนำ แนวคิด และช่วยเหลือให้คำปรึกษา อันเป็นประโยชน์ต่อรายงานฉบับนี้ และขอขอบคุณ ผศ.ดร.วรวุฒิ มรรคเจริญ ที่เป็นอาจารย์ที่ปรึกษาช่วยเหลือ ให้คำแนะนำ และแนวคิด รวมถึงยังได้ตรวจทานแก้ไขรายงานฉบับนี้จนสำเร็จบรรลุตามเป้าหมายได้ด้วยดี รวมทั้ง ใคร่ขอขอบคุณ บิดา และมารดา ที่ให้การสนับสนุนในด้านต่างๆ รวมทั้งคอยให้กำลังใจเสมอมา จนประสบความสำเร็จ

ทั้งนี้ ข้าพเจ้าหวังเป็นอย่างยิ่งว่า โครงงานที่จัดทำขึ้นนี้ จะเป็นประโยชน์แก่ผู้ที่สนใจ และใคร่ศึกษา เพื่อนำไปประยุกต์ใช้ และพัฒนาในด้านอื่นๆต่อไป หากมีข้อผิดพลาดประการใด ข้าพเจ้าขอน้อมรับและนำไปแก้ไขต่อไป

สิทธิชัย เสียงสนั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญ(ต่อ) V
สารบัญตาราง VI
สารบัญรูป VII
สารบัญรูป(ต่อ) VIII
สารบัญรูป(ต่อ) IX
บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญ 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 1
1.4 ตารางเวลาที่ใช้ในการดำเนินโครงงาน 1
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2
1.6 คำจำกัดความที่ใช้ในการศึกษา 2
1.7 ประวัติของสถานประกอบการ 3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4
2.1 ระบบควบคุมอัตโนมัติ 4
2.1.1 องค์ประกอบของการควบคุม 4
2.1.2 ประเภทของการควบคุม 5
2.1.3 การเลือกใช้ประเภทของการควบคุม 6
2.1.4 การควบคุมแบบต่างๆในโรงงานอุตสาหกรรม 6
2.2 แขนกลอัตโนมัติ 6
2.2.1 การแบ่งชนิดของหุ่นยนต์ 7
2.2.2 Cartesian (Gantry) Robot 8
2.2.3 Cylindrical Robot 9
2.2.4 Spherical Robot (Polar) 10
2.2.5 SCARA Robot 11
2.2.6 Articulated Arm (Revolute) 12
2.3 การควบคุมระบบนิวเมติกส์ 13
2.3.1 โซลินอยด์วาล์ว 14
2.4 ระบบการสื่อสาร (Communication) 15
2.4.1 มาตรฐานแนะนำรุ่น -232 (Recommended Standard) 15
2.4.2 ระบบเครือข่ายคอมพิวเตอร์ (LAN) 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.4.3 ระบบอินพุต และเอาท์พุต (I/O) 18
2.5 เซนเซอร์ 19
2.5.1 Barcode reader 18
2.5.2 Pressure sensor 20
2.5.3 Reed switch 20
2.5.4 Photo sensor 21
2.6 ระบบควบคุมความเร็วมอเตอร์ 25
2.7 ความรู้ทั่วไปเกี่ยวกับรีเลย์ 26
2.8 พื้นฐานการใช้ภาษา SPEL 28
บทที่ 3 วิธีดำเนินงาน 29
3.1 ศึกษาและรวบรวมข้อมูล 29
3.2 การออกแบบเครื่อง 30
3.3 การติดตั้งอุปกรณ์ 37
3.3.1 ปุ่มฉุกเฉิน (Emergency button) 37
3.3.2 สายมาตรฐานแนะนำรุ่น -232 (Recommended Standard) 38
3.3.3 Hand checking 39
3.3.4 Solenoid valve 40
3.3.5 Wiring I/O 41
3.3.6 Robot setup 43
3.4 เขียนโปรแกรมควบคุม 48
3.4.1 Sequence 48
3.4.2 GUI (Graphic user interface) 57
บทที่ 4 ผลการดำเนินงาน 61
บทที่ 5 สรุปการดำเนินงานและข้อเสนอแนะ 68
5.1 สรุปผลการดำเนินงาน 68
5.2 ปัญหาและอุปสรรค 68
5.3 ข้อเสนอแนะ 68
เอกสารอ้างอิง 69
ภาคผนวก 70
ภาคผนวก ก 71
ภาคผนวก ข 82
ภาคผนวก ค 93
ประวัติผู้เขียน 112

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1-1 ตารางระยะเวลาที่ใช้ดำเนินโครงงาน	1
2-1 ตารางแสดงการแบ่งรูปแบบของหุ่นยนต์	7
2-2 ตารางแสดงการแบ่งชนิดของหุ่นยนต์	8
2-3 ตารางบอกหน้าที่ของแต่ละพินของ RS-232	16
2-4 ตารางแสดงอัตราการส่งข้อมูล (Baud rate) กับความยาวสายเชื่อมต่อ	16
2-5 Specification of 8 pin speed controller	25
3-1 ตารางแสดงอุปกรณ์ที่ใช้ภายในเครื่องติดสติ๊กเกอร์อัตโนมัติ	33
3-2 ตารางแสดงอินพุทของบอร์ด 1	34
3-3 ตารางแสดงอินพุทของบอร์ด 2	35
3-4 ตารางแสดงเอาท์พุทของบอร์ด 1	36
3-5 ตารางแสดงเอาท์พุทของบอร์ด 2	36
4-1 ตารางแสดงผลการทดสอบการทำงานของแต่ละฟังก์ชั่น	65
4-2 ตารางแสดงผลการทำงานของเครื่องติดสติ๊กเกอร์	67
5-1 ตารางแสดงประสิทธิภาพการทำงานของเครื่องติดสติ๊กเกอร์	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 รูปแสดงองค์ประกอบของการควบคุม	4
2.2 รูปแสดงการควบคุมแบบเปิด (Open Loop)	5
2.3 รูปแสดงการควบคุมแบบปิด (Close Loop)	5
2.4 รูปแสดงการเปรียบเทียบส่วนต่างๆของมนุษย์กับหุ่นยนต์	6
2.5 Cartesian Robot Work Envelop Of Cartesian Robot	8
2.6 Cylindrical Robot Work Envelop Of Cylindrical Robot	9
2.7 Spherical Robot Work Envelop Of Spherical Robot	10
2.8 SCARA Robot Work Envelop Of SCARA Robot	11
2.9 Articulated Arm Robot Work Envelop Of Articulated Robot	12
2.10 รูปแสดงส่วนประกอบของโซลินอยด์วาล์ว	14
2.11 รูปแสดงสายเชื่อมต่อ RS-232	15
2.12 รูปแสดงลักษณะพินของ RS-232	15
2.13 รูปแสดงการแปลงเลขฐาน	17
2.14 รูปแสดงลักษณะของสายแลน(LAN)	18
2.15 รูปแสดงลักษณะของคอนโทรลเลอร์สำหรับใช้ควบคุมแขนกล	19
2.16 รูปแสดงลักษณะของเครื่องอ่านบาร์โค้ด	19
2.17 รูปแสดงหลักการทำงานของเครื่องอ่านบาร์โค้ด	20
2.18 รูปแสดงลักษณะของเครื่องวัดความดัน	20
2.19 รูปแสดงลักษณะของ Reed switch	20
2.20 รูปแสดงส่วนประกอบของ Reed switch	21
2.21 รูปแสดงการทำงานของ Reed switch	21
2.22 รูปแสดงการทำงานของ Thru – Beam Type	22
2.23 รูปแสดงการทำงานของ Diffuse – Reflective Beam Sensor	22
2.24 รูปแสดงการทำงานของ Retro – Reflective Beam Sensor	23
2.25 รูปแสดงชุดควบคุมความเร็วมอเตอร์แบบ 8 พิน	25
2.26 รูปแสดงลักษณะทั่วไปของรีเลย์	26
2.27 รูปแสดงการต่อแหล่งจ่ายไฟเข้ากับ NC และ Common	27
2.28 รูปแสดงการต่อแหล่งจ่ายไฟเข้ากับ NO และ Common	27
3.1 รูปแสดงโครงสร้างโดยรวมของสายการประกอบฮาร์ดดิสก์	30
3.2 รูปแสดงโครงสร้างโดยรวมของสายการประกอบฮาร์ดดิสก์(Top view)	30
3.3 รูปแสดงลักษณะโดยรวมของเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ	31
3.4 รูปแสดงสายพานลำเลียงฮาร์ดดิสก์และฝาครอบ	32
3.5 รูปแสดงการเชื่อมต่อต่างๆภายในเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ	32
3.6 รูปแสดงอุปกรณ์สำหรับทำปุ่มฉุกเฉินของคอนโทรลเลอร์	37
3.7 Emergency Connection DB-25 Diagram	37
3.8 รูปแสดง Emergency connector หลังทำการบัดกรี	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.9 รูปแสดงปุ่ม Emergency สำหรับคอนโทรลเลอร์	38
3.10 รูปแสดงการบัดกรีสายเชื่อมต่อ RS-232	38
3.11 รูปแสดงลักษณะสายเชื่อมต่อ RS-232	39
3.12 Relay connection Hand checking Diagram	39
3.13 รูปแสดงตำแหน่งการติดตั้งรีเลย์ภายในแผงควบคุม	39
3.14 Hand checking connector	40
3.15 รูปแสดงลักษณะการแบ่งลมของโซลินอยด์วาล์ว	40
3.16 รูปแสดงลักษณะการต่อสายลมกับโซลินอยด์วาล์ว	40
3.17 รูปแสดงตำแหน่งการติดตั้งโซลินอยด์วาล์วบนแผงควบคุม	41
3.18 รูปแสดงการต่อสาย I/O จาก Terminal I/O ของคอนโทรลเลอร์	41
3.19 รูปแสดงตำแหน่งการติดตั้งเซนเซอร์ ลีดสวิตช์ และสตอปเปอร์	41
3.20 รูปแสดงสายอินพุทจากเซนเซอร์และลีดสวิตช์	42
3.21 รูปแสดงการต่อสายอินพุทจากเซนเซอร์ และเอาท์พุทต่างๆเข้าสู่ Terminal I/O	42
3.22 รูปแสดงการต่อ I/O ทั้งหมดบนแผงควบคุม	42
3.23 รูปแสดงการทดสอบความถูกต้องของการทำงานของ I/O ต่างๆ	43
3.24 รูปแสดงการเปลี่ยนแบตเตอรี่ของแขนกล	43
3.25 รูปแสดงการเปลี่ยนแบตเตอรี่ของคอนโทรลเลอร์	44
3.26 รูปแสดงตำแหน่งการปรับตั้งค่า Joint#1 ของแขนกล	44
3.27 รูปแสดงตำแหน่งการปรับตั้งค่า Joint#2 ของแขนกล	44
3.28 รูปแสดงตำแหน่งการปรับตั้งค่า Joint#3 ของแขนกล	45
3.29 รูปแสดงตำแหน่งการปรับตั้งค่า Joint#4 ของแขนกล	45
3.30 รูปแสดงการเปิดโปรแกรม EPSON RC+7.0	45
3.31 รูปแสดงการเชื่อมต่อคอนโทรลเลอร์กับคอมพิวเตอร์	46
3.32 รูปแสดงการเลือกการเชื่อมต่อแบบ USB ใน EPSON RC+7.0	46
3.33 รูปแสดงการเปิดหน้าต่าง command windows ใน EPSON RC+7.0	46
3.34 รูปแสดงคำสั่งรีเซต encoder	47
3.35 รูปแสดงการสั่งให้คอนโทรลเลอร์เริ่มต้นการทำงานอีกครั้ง	47
3.36 รูปแสดงคำสั่งในการบันทึกค่าการ calibrate แขนกลทั้ง 4 แกน	47
3.37 รูปแสดงสัญลักษณ์โปรแกรม EPSON RC-7.0	48
3.38 รูปแสดงสัญลักษณ์โปรแกรม Visual Studio 2015	48
3.39 แผนภาพแสดงการทำงานโดยรวมของเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ	49
3.40 แผนภาพแสดงการควบคุมการอ่านค่าบาร์โค้ดของเครื่องอ่านบาร์โค้ด	50
3.41 แผนภาพแสดงการลำเลียงฮาร์ดดิสก์บนคอนเวเยอร์	51
3.42 แผนภาพแสดงการลำเลียงฝาครอบบนคอนเวเยอร์	52
3.43 แผนภาพแสดงการส่งข้อมูลไปยังเครื่องพิมพ์ฉลาก	53
3.44 แผนภาพแสดงการยกลิฟเตอร์สำหรับยึดตำแหน่งของฝาครอบ	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.45 แผนภาพแสดงการยกเทเบิ้ลฟีดสำหรับการหยิบสติ๊กเกอร์ของแขนกล	55
3.46 แผนภาพแสดงการควบคุมการเคลื่อนที่ของแขนกล	56
3.47 รูปแสดงหน้าจอเข้าสู่ระบบของโปรแกรม Robot Auto Labeling	57
3.48 รูปแสดงหน้าจอแสดงผลดารทำงานโดยรวมของเครื่องติดสติ๊กเกอร์อัตโนมัติ	57
3.49 รูปแสดงหน้าจอการควบคุมการตั้งค่าของโปรแกรม robot Auto Labeling	58
3.50 รูปแสดงหน้าจอควบคุมการทำงานของแขนกล	58
3.51 รูปแสดงหน้าจอควบคุมการค้นหาตำแหน่งที่ผิดพลาดของฉลาก	59
3.52 รูปแสดงหน้าจอควบคุมการค้นหาบาร์โค้ดบนฉลากของกล้อง	59
3.53 รูปแสดงหน้าจอควบคุมการปรับค่าการวัดของกล้อง	60
3.54 รูปแสดงหน้าจอควบคุมการสั่งพิมพ์ของเครื่องพิมพ์บาร์โค้ด	60
4.1 รูปแสดงการติดตั้งอุปกรณ์ต่างๆในเครื่องติดสติ๊กเกอร์	61
4.2 รูปแสดงปัญหาตำแหน่งที่ไม่เหมาะสมในการติดตั้งเซนเซอร์	61
4.3 รูปแสดงตำแหน่งที่เหมาะสมในการติดตั้งเซนเซอร์	62
4.4 รูปแสดงการลำเลียง buffer ที่ต้องใช้ในเครื่องพิมพ์ฉลาก	62
4.5 รูปแสดงตำแหน่งที่แบนกลสามารถหยิบสติ๊กเกอร์ได้	62
4.6 รูปแสดงการลำเลียงฮาร์ดดิสก์บนคอนเวเยอร์	63
4.7 รูปแสดงการใช้ระบบวิทัศน์ในการค้นหาตำแหน่งของฉลาก	63
4.8 รูปแสดงผลการติดสติ๊กเกอร์ด้วยแขนกล	64
4.9 รูปแสดงผลการทดสอบการติดสติ๊กเกอร์ที่ตำแหน่งเดิมซ้ำๆ	64
4.10 รูปแสดงการสั่งข้ามฉลากที่ไม่สมบูรณ์	64
4.11 รูปแสดงเครื่องติดสติ๊กเกอร์บาร์โค้ดหลังทดสอบการทำงาน	65
4.12 รูปแสดงเครื่องติดสติ๊กเกอร์ที่ถูกห่อด้วยพลาสติกสำหรับการเคลื่อนย้าย	66
4.13 รูปแสดงตำแหน่งเครื่องติดสติ๊กเกอร์ที่ติดตั้งในสายการผลิต	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบัน การแข่งขันด้านการผลิตในภาคอุตสาหกรรม มีการแข่งขันที่สูงขึ้น แต่ละบริษัทจึงจำเป็นต้องมีการพัฒนากระบนการผลิตของตนเองให้มีประสิทธิภาพ และมาตรฐานที่สูงขึ้น เพื่อที่จะสามารถตอบสนองได้ทันต่อความต้องการของตลาด ซึ่งการพัฒนากระบวนการผลิตนั้น จำเป็นต้องใช้ระบบอัตโนมัติเข้ามาช่วยในการดำเนินงานแทนแรงงานมนุษย์ เพื่อให้ได้ประสิทธิภาพ มีความถูกต้องและความแม่นยำในการผลิตที่สูงขึ้น โดยเป็นการลดค่าใช้จ่ายในการผลิตอีกทางหนึ่งด้วย และเนื่องจากบริษัทแคล - คอมพ์ อีเล็คโทรนิคส์ (ประเทศไทย) จำกัด (มหาชน) เป็นบริษัทรับประกอบงานต่างๆ และมีกระบวนการต่างๆที่เกี่ยวการผลิตมากมาย ซึ่งหนึ่งในนั้นคือ การติดสติ๊กเกอร์บาร์โค้ดบนฝาครอบฮาร์ดดิสก์ให้ตรงกับรหัสบาร์โค้ดบนฮาร์ดดิสก์ ซึ่งเดิมจะใช้แรงงานมนุษย์ในการติดสติ๊กเกอร์ ซึ่งมีขั้นตอนที่ยุ่งยาก ใช้เวลาในการดำเนินการสูง และสิ้นเปลืองแรงงานมนุษย์ โครงงานนี้จึงจัดทำขึ้นเพื่อเปลี่ยนขั้นตอนต่างๆให้เป็นระบบอัตโนมัติ เพื่อลดค่าใช้จ่ายจากการจ้างงาน ลดระยะเวลาที่ใช้ดำเนินการ และเป็นการเพิ่มประสิทธิภาพการผลิตให้สูงขึ้นอีกทางหนึ่ง

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 ลดค่าใช้จ่ายของการจ้างแรงงาน

1.2.2 สามารถควบคุมคุณภาพและมาตรฐานการผลิตชิ้นงานได้มากขึ้น

1.2.3 การใช้ระบบอัตโนมัติทดแทนการใช้แรงงานมนุษย์

1.3 ขอบเขตของโครงงาน

1.3.1 ศึกษาและพัฒนาการควบคุมขั้นตอนการทำงานของแขนกลให้ทำการติดสติ๊กเกอร์แบบอัตโนมัติ

1.3.2 ศึกษาและพัฒนาระบบควบคุมการลำเลียงการเคลื่อนที่ของฮาร์ดดิสก์และฝาครอบบนสายพาน

1.3.3 เครื่องจักรสามารถเชื่อมต่อกับเครื่องจักรก่อนหน้า และเครื่องจักรด้านหลังได้

1.4 ตารางระยะเวลาที่ใช้ในการดำเนินโครงงาน

ตารางที่ 1-1 ตารางระยะเวลาที่ใช้ดำเนินโครงงาน

ลำดับ	หัวข้องาน	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3				เดือนที่ 4			
1	ศึกษาการควบคุมแขนกล																
2	ศึกษากระบวนการผลิตในโรงงาน																
3	ออกแบบเครื่องและสั่งซื้อวัสดุ																
4	ประกอบเครื่อง(wiring)																
5	เขียนโปรแกรม																
6	ทดสอบการทำงาน																
7	สรุปผลการดำเนินงาน																

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้เรียนรู้กระบวนการทำงานในสายการผลิต

1.5.2 สามารถเรียนรู้และสร้างประสบการณ์ในการต่อวงจรไฟฟ้า (Wiring)

1.5.2 สามารถเรียนรู้การควบคุมหุ่นยนต์ในอุตสาหกรรม

1.5.3 สามารถนำความรู้จากการใช้ อินพุต เอาท์พุต ของคอนโทรลเลอร์แขนกล (Robot controller) ไปประยุกต์ใช้ในการควบคุมการทำงานอัตโนมัติแบบอื่นๆได้

1.6 คำจำกัดความที่ใช้ในการศึกษา

1.6.1 แขนกลอัตโนมัติ (Robot arm) เป็นเครื่องจักรกลชนิดหนึ่งที่มีลักษณะโครงสร้างและหน้าที่คล้ายแขนของมนุษย์ โดยมีการควบคุมการเคลื่อนที่ผ่านคอนโทรลเลอร์ (Robot controller) เพื่อให้เคลื่อนที่ไปตามตำแหน่งหรือลักษณะที่ต้องการ

1.6.2 คอนโทรลเลอร์ (Robot controller) เป็นตัวควบคุมการทำงานของแขนกลอัตโนมัติ ซึ่งทำหน้าที่แปลงชุดคำสั่ง หรืออินพุต ที่ได้รับมา ไปควบคุมการเคลื่อนที่ของแขนกลในลักษณะต่างๆ

1.6.3 ระบบนิวเมติกส์ (Pneumatics) เป็นระบบการทำงาน โดยใช้การควบคุมการไหลของอากาศในการขับเคลื่อนอุปกรณ์ต่างๆของเครื่องจักร เช่น กระบอกสูบลม เป็นต้น

1.6.4 สตอปเปอร์ (Stopper) คือ กระบอกสูบที่มีการควบคุมการเคลื่อนที่เข้า-ออกด้วยระบบนิวเมติกส์ และโซลินอยด์วาล์ว ใช้สำหรับการหยุดการเคลื่อนที่ของฮาร์ดดิสก์บนสายพาน

1.6.5 โซลินอยด์วาล์ว (Solenoid valve) วาล์วลมที่มีการควบคุมด้วยไฟฟ้า สำหรับควบคุมการไหลของอากาศให้ไปในทิศทางต่างๆที่ต้องการ

1.6.6 สายพานบัฟเฟอร์ (Conveyor buffer) เป็นสายพานที่ใช้สำหรับลำเลียงฮาร์ดดิสก์และฝาครอบไปยังตำแหน่งที่ต้องการโดยอัตโนมัติ โดยใช้คอนโทรลเลอร์ในการควบคุมการทำงานของสายพาน และมีสตอปเปอร์สำหรับการขวางการเคลื่อนที่ของฮาร์ดดิสก์และฝาครอบให้หยุดในตำแหน่งที่ต้องการ

1.6.7 Hand checking คือ การเชื่อมต่อระหว่างสองสิ่งให้ทำงานสอดคล้องกัน และมีการส่งสัญญาณเอาท์พุตบอกสถานะ และรับสัญญาณอินพุตซึ่งกันและกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 ประวัติของสถานประกอบการ

บริษัท แคล-คอมพ์ อิเล็คโทรนิคส์ (ประเทศไทย) จำกัด (มหาชน) ("บริษัทฯ") ก่อตั้งด้วยทุนจดทะเบียนเริ่มแรก 125 ล้านบาท เมื่อวันที่ 4 ธันวาคม 2532 เพื่อผลิตและส่งออกผลิตภัณฑ์สำเร็จรูปและอุปกรณ์ประเภทอีเล็กโทรนิกส์ (Electronics Manufacturing Services: EMS) ปัจจุบันบริษัทฯ บริษัทฯ มีทุนจดทะเบียน 4,277,556,192 บาท และ มีทุนชำระแล้วทั้งสิ้น 4,085,064,192 บาท

บริษัทฯ ประกอบธุรกิจผลิต (Original Equipment Manufacturing-OEM) ผลิตภัณฑ์อีเล็กทรอนิกส์ให้กับเจ้าของผลิตภัณฑ์ชั้นนำของโลกหลายบริษัท เช่น Western Digital, Seagate, Hitachi, Advance Digital Broadcast, Technicolor, Pace, Hewlett Packard, Nikon และอื่นๆ เป็นต้น ผลิตภัณฑ์อิเล็กทรอนิกส์ที่บริษัทฯ ผลิต สามารถแบ่งออกได้เป็น 2 ประเภทหลัก ได้แก่

อุปกรณ์คอมพิวเตอร์ ได้แก่ เครื่องพิมพ์ชนิดพ่นหมึก เครื่องพิมพ์ชนิดเลเซอร์ เครื่องพิมพ์ multi-function เครื่องพิมพ์กระดาษต่อเนื่อง เครื่องพิมพ์ขนาดใหญ่ เมนบอร์ด External Hard Disk และแผงวงจร PC สำหรับ hard disk เป็นต้น

อุปกรณ์โทรคมนาคม ได้แก่ รับส่งสัญญาณดาวเทียม เครื่องหูฟังโทรศัพท์มือถือระบบไร้สาย (Bluetooth) เป็นต้น

ณ วันที่ 31 ธันวาคม 2555 บริษัทฯ มีพนักงานทั่วโลกทั้งสิ้น 16,937 คน โดยแบ่งออกเป็นพนักงานในประเทศจีนจำนวน 5,547 คน ในไทยจำนวน 6,994 คน ในไต้หวันจำนวน 1,068 คน ในมาเลเซียจำนวน 678คน ในบราซิลจำนวน1,345 คนในเม็กซิโกจำนวน 890 คน ในสหรัฐอเมริกาจำนวน 258 คน ในสิงคโปร์จำนวน 61 คนและในฟิลิปปินส์จำนวน 96 คน

บริษัทฯ ได้รับการส่งเสริมการลงทุนในอุตสาหกรรมการผลิต ผลิตภัณฑ์อิเล็กทรอนิกส์ และ ผลิตชิ้นส่วนหรืออุปกรณ์ที่ใช้กับผลิตภัณฑ์ อิเล็กทรอนิกส์ จากสำนักงานคณะกรรมการส่งเสริมการลงทุน (BOI) ใช้สิทธิประโยชน์การส่งเสริมการลงทุนจากสำนักงาน จำนวน 8 โครงการ นอกจากนี้ยังมี บริษัท Avaplas (Thailand) Limited ซึ่งได้รับการส่งเสริมการลงทุนในโครงการสำหรับการผลิตชิ้นส่วนพลาสติกสำหรับสินค้าอิเล็กทรอนิกส์ อีก 3 โครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจวัด หมายถึง อุปกรณ์ที่ให้สัญญาณขาออก ซึ่งมีขนาดสัมพันธ์กับขนาดตัวแปรทางฟิสิกส์ของสิ่งที่ต้องการวัดหรือสั่งงาน

การควบคุม หมายถึง สิ่งที่ทำหน้าที่ออกคำสั่งหรือกำหนดสัญญาฯควบคุมตามกฎเกณฑ์การควบคุมที่กำหนดไว้ล่วงหน้า คำสั่งหรือสัญญาณควบคุมนี้อาจจะเป็นฟังก์ชั่นกับเวลาหรือฟังก์ชั่นกับสัญญาณขาเข้าที่ได้รับจากอุปกรณ์ตรวจวัด

การควบคุมสุดท้าย หมายถึง อุปกรณ์ที่ทำหน้าที่ปรับถาวะของกระบวนการด้วยการเปลี่ยนแปลงค่าตัวแปรปรับการทำงาน ในการควบคุมส่วนใหญ่คำสั่งหรือสัญญาณควบคุมจะมีขนาดหรือพลังงานน้อยๆ ส่วนตัวแปรปรับกระบวนการจะมีขนาดหรือพลังงานมาก

กระบวนการ หมายถึง กระบวนการทางฟิสิกส์ที่ต้องการควบคุมให้มีสภาวะการทำงานตามต้องการในขณะที่สภาวะการทำงานหรือสภาพแวดล้อมอาจจะเปลี่ยนแปลงตลอดเวลา

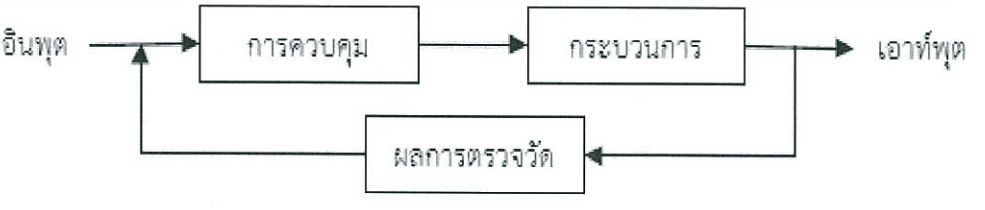
2.1.2 ประเภทของการควบคุม

การแบ่งประเภทของการควบคุมจะแบ่งตามเกณฑ์ของประเภทได้ดังนี้

- การควบคุมแบบเปิด (Open Loop) เป็นระบบควบคุมที่เอาท์พุตของระบบไม่มีผลต่อการควบคุมเอาท์พุตของระบบจะไม่ถูกวัดหรือถูกป้อนกลับมาเพื่อเปรียบเทียบอินพุต ตัวอย่างเช่นการควบคุมการปิดไฟ การควบคุมทิศทางการหมุนของมอเตอร์ การควบคุมมอเตอร์ขับสายพานลำเลียง การควบคุมเครื่องซักผ้า การควบคุมสัญญาณไฟจราจร (พรจิต, 2542: 12-15)

รูปที่ 2.2 รูปแสดงการควบคุมแบบเปิด (Open Loop)

- การควบคุมแบบปิด (Close Loop) เป็นระบบควบคุมซึ่งสัญญาณเอาท์พุตจะมีผลโดยตรงต่อการควบคุม ระบบการควบคุมแบบป้อนกลับนี้อาจจะเป็นสัญญาณเอาท์พุต โดยตรงหรือเป็นสัญญาณที่เป็นฟังก์ชั่นของสัญญาณเอาท์พุตก็ได้ตัวอย่างเช่น การควบคุมอุณหภูมิ ความดัน การไหล ความเร็ว ให้คงที่



รูปที่ 2.3 แสดงการควบคุมแบบปิด (Close Loop)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การเลือกใช้ประเภทของการควบคุม

ระบบควบคุมแบบเปิด (Open Loop) นั้นเหมาะกับระบบที่ทราบว่าอินพุตของระบบจะมีการเปลี่ยนแปลงอย่างไร และแน่ใจว่าไม่มีสัญญาณรบกวนภายนอก สำหรับการควบคุมแบบปิด (Close Loop) จะใช้เมื่อไม่ทราบถึงการเปลี่ยนแปลง ค่าพารามิเตอร์ของอุปกรณ์ในระบบโดยไม่คาดคิดมาก่อน และในบางกรณี อาจจะมีทั้งการควบคุมทั้ง 2 แบบ เพื่อให้ระบบทั้งหมดมีผลตามที่ต้องการ

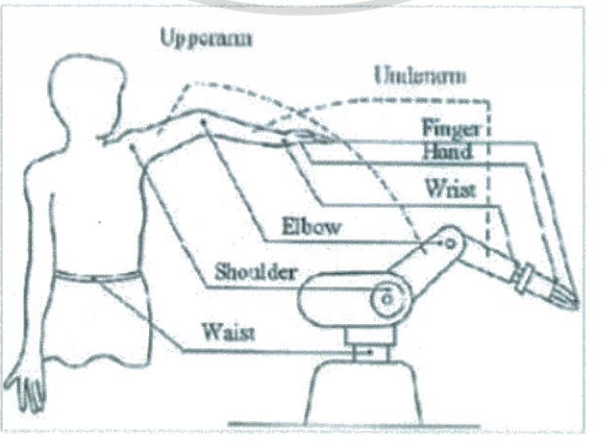
2.1.4 การควบคุมแบบต่างๆ ในโรงงานอุตสาหกรรม

ในโรงงานอุตสาหกรรมโดยทั่วไปส่วนใหญ่แล้ว การควบคุมการทำงาน ของระบบอัตโนมัติในโรงงานจะนิยมเอาอุปกรณ์ต่างๆ เหล่านี้เข้ามาใช้เป็นตัวควบคุมการทำงาน ในระบบอัตโนมัติที่เป็นระบบนิวเมติกส์หรือไฮโดรลิกส์คือ

- ระบบการควบคุมที่ใช้แผงรีเลย์
- ระบบการควบคุมที่ใช้แผงวงจรอิเล็กทรอนิกส์
- ระบบการควบคุมที่ใช้คอมพิวเตอร์
- ระบบการควบคุมที่ใช้ชุดพีแอลซี

2.2 แขนกลอัตโนมัติ

อุตสาหกรรมในประเทศไทยจะเห็นได้ว่ามีการนำเทคโนโลยีระบบอัตโนมัติ (Automation Technology) เข้ามาใช้งานเพื่อให้สินค้าสามารถแข่งขันในตลาดโลกได้ ทั้งในเรื่องราคา และคุณภาพ โดยเฉพาะในเรื่องคุณภาพ อาจเนื่องจากมีการเปลี่ยนรุ่นผลิตภัณฑ์อยู่บ่อยๆ ต้องใช้เวลาในการ Set Up ปัจจุบันจึงมีการนำเทคโนโลยีต่างๆ เข้ามาใช้ หนึ่งในเทคโนโลยีที่มีความยืดหยุ่นสูง ได้แก่ หุ่นยนต์อุตสาหกรรม เนื่องจากการเปลี่ยนการทำงาน สามารถทำได้โดยการเปลี่ยนโปรแกรม นอกจากนี้คุณภาพของผลิตภัณฑ์ที่ได้มี ความสม่ำเสมอเป็นมาตรฐานเดียวกัน การทำงานของหุ่นยนต์อุตสาหกรรมจะเลียนแบบร่างกายของมนุษย์ โดยจะเลียนแบบเฉพาะส่วนของ ร่างกายที่จะนำไปใช้ประโยชน์ในอุตสาหกรรมเท่านั้น นั่นคือช่วงแขนของมนุษย์ ดังนั้น บางคนอาจจะได้ยินคำ ว่า “แขนกล” ซึ่งก็หมายถึงหุ่นยนต์อุตสาหกรรม การทำงานของหุ่นยนต์อุตสาหกรรมเปรียบเทียบกับแขนมนุษย์ แสดงดังรูป



รูปที่ 2.4 รูปแสดงการเปรียบเทียบส่วนต่างๆของมนุษย์กับหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบัน และในอนาคตหุ่นยนต์อุตสาหกรรมจะเข้ามามีบทบาทในอุตสาหกรรมมากขึ้น โดยจะทำงานแทนมนุษย์ในงานต่างๆ เหล่านี้เป็นงานที่อันตราย เช่น งานยกเหล็กเข้าเตาหลอม งานที่เกี่ยวข้องกับสารเคมี งาน ซ้ำซากน่าเบื่อ เช่น งานยกสินค้าจากสายงานการผลิต งานประกอบ งานบรรจุผลิตภัณฑ์ งานที่ต้องการ คุณภาพมาตรฐานเดียวกัน เช่น งานเชื่อม งานตัด งานที่ต้องใช้ทักษะความชำนาญสูง เช่น งานเชื่อมแนว เชื่อมเลเซอร์ งานที่ต้องใช้ความละเอียดประณีต เช่น งานประกอบชิ้นส่วนอิเล็กทรอนิกส์ งานตรวจสอบ (Inspection) ฯลฯ

หุ่นยนต์ คือ เครื่องจักรที่ถูกควบคุมอัตโนมัติ สามารถเขียนโปรแกรมใหม่ได้ ใช้งานเอนกประสงค์ โปรแกรมการเคลื่อนที่จะต้องสามารถโปรแกรมให้เคลื่อนที่ได้อย่างน้อย 3 แกนหรือมากกว่า หุ่นยนต์ อาจจะ ยึดอยู่กับที่หรือย้ายตำแหน่ง (Mobile) เพื่อใช้ในงานอุตสาหกรรม

2.2.1 การแบ่งชนิดของหุ่นยนต์

โดยทั่วไปการแบ่งชนิดของหุ่นยนต์จะแบ่งตามลักษณะรูปทรงของพื้นที่ทำงาน (Envelope Geometric) แต่ก่อนจะอธิบายชนิดของหุ่นยนต์ขออธิบายการท างานของจุดต่อ (Joint) ของหุ่นยนต์อุตสาหกรรมซึ่งในขั้นพื้นฐานมี 2 ชนิดด้วยกัน ดังนี้

ตารางที่ 2-1 ตารางการแบ่งรูปแบบของหุ่นยนต์

ชนิด	สัญญาลักษณ์	หมายเหตุ
Revolute (R)		การหมุนรอบแกน (Rotary)
Prismatic (P)		การเคลื่อนที่เชิงเส้น (Linear motion)

จุดต่อ (Joint) ทั้งสองแบบเมื่อนำมาต่อเข้าด้วยกันอย่างน้อย 3 แกนหลักจะได้พื้นที่ทำงาน (Work envelope) ที่มีลักษณะแตกต่างกันไป ซึ่งสามารถนำมาแบ่งชนิดของหุ่นยนต์ได้ดังต่อไปนี้

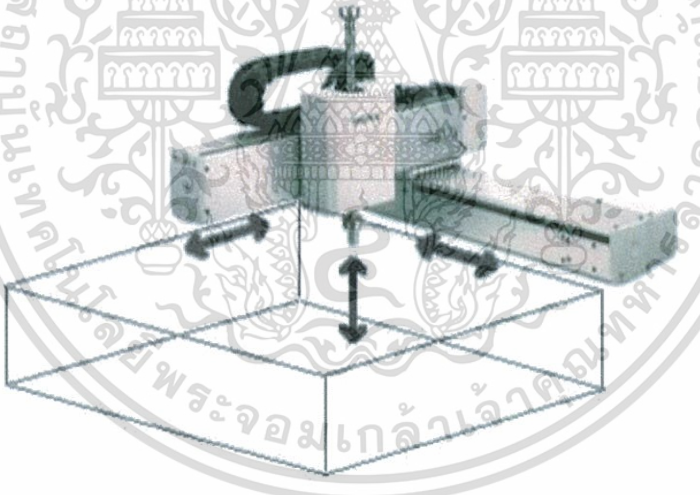
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2-2 ตารางการแบ่งชนิดของหุ่นยนต์

ชนิดของหุ่นยนต์	แกนที่ 1	แกนที่ 2	แกนที่ 3
Cartesian (gantry)	P	P	P
Cylindrical	R	P	P
Spherical (Polar)	R	R	P
SCARA	R	R	R
Articulated	R	R	R

2.2.2 Cartesian (Gantry) Robot

แกนทั้ง 3 ของหุ่นยนต์จะเคลื่อนที่เป็นแบบเชิงเส้น (Prismatic) ถ้าโครงสร้างมีลักษณะคล้าย Overhead Crane จะเรียกว่าเป็นหุ่นยนต์ชนิด Gantry แต่ถ้าหุ่นยนต์ไม่มีขาตั้งหรือขาเป็นแบบอื่น เรียกว่า ชนิด Cartesian



รูปที่ 2.5 Cartesian Robot Work Envelop Of Cartesian Robot

ข้อดี

1. เคลื่อนที่เป็นแนวเส้นตรงทั้ง 3 มิติ
2. การเคลื่อนที่สามารถทำความเข้าใจง่าย
3. มีส่วนประกอบง่าย
4. โครงสร้างแข็งแรงตลอดการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสีย

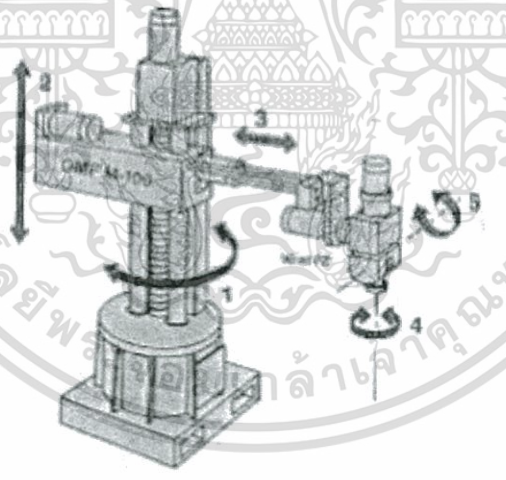
1. ต้องการพื้นที่ติดตั้งมาก
2. บริเวณที่หุ่นยนต์เข้าไปทำงานได้ จะเล็กกว่าขนาดของตัวหุ่นยนต์
3. ไม่สามารถเข้าถึงวัตถุจากทิศทางข้างใต้ได้
4. แกนแบบเชิงเส้นจะ Seal เพื่อป้องกันฝุ่น และของเหลวได้ยาก

การประยุกต์ใช้งาน

เนื่องจากโครงสร้างมีความแข็งแรงตลอดแนวการเคลื่อนที่ ดังนั้นจึงเหมาะกับงานเคลื่อนย้าย ของหนักๆ หรือเรียกว่างาน Pick-and-Place เช่น ใช้โหลดชิ้นงานเข้าเครื่องจักร (Machine loading) ใช้จัดเก็บชิ้นงาน (Stacking) นอกจากนี้ยังสามารถใช้ในงานประกอบ (Assembly) ที่ไม่ต้องการเข้าถึง ในลักษณะที่มีมุมหมุน เช่น ประกอบอุปกรณ์อิเล็กทรอนิกส์ และงาน Test ต่างๆ

2.2.3 Cylindrical Robot

หุ่นยนต์ประเภทนี้จะมีแกนที่ 2 (ไหล่) และแกนที่ 3 (ข้อศอก) เป็นแบบ Prismatic ส่วนแกน ที่ 1 (เอว) จะเป็นแบบหมุน (Revolute) ทำให้การเคลื่อนที่ได้พื้นที่การท างานเป็นรูปทรงกระบอก ดังรูป



รูปที่ 2.6 Cylindrical Robot Work Envelop Of Cylindrical Robot

ข้อดี

1. มีส่วนประกอบไม่ซับซ้อน
2. การเคลื่อนที่สามารถเข้าใจได้ง่าย
3. สามารถเข้าถึงเครื่องจักรที่มีการเปิด - ปิด หรือเข้าไปในบริเวณที่เป็นช่องหรือโพรงได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสีย

1. มีพื้นที่ทำงานจำกัด

2. แกนที่เป็นเชิงเส้นมีความยุ่งยากในการ Seal เพื่อป้องกันฝุ่น และของเหลว

การประยุกต์ใช้งาน

โดยทั่วไปจะใช้ในการหยิบยกชิ้นงาน (Pick-and-Place) หรือป้อนชิ้นงานเข้าเครื่องจักร เพราะสามารถเคลื่อนที่เข้าออกบริเวณที่เป็นช่องโพรงเล็กๆ ได้สะดวก

2.2.4 Spherical Robot (Polar)

มีสองแกนที่เคลื่อนในลักษณะการหมุน (Revolute Joint) คือแกนที่ 1 (เอว) และ แกนที่ 2 (ไหล่) ส่วนแกนที่ 3 (ข้อศอก) จะเป็นลักษณะของการเคลื่อนที่แนวเส้นตรง ดังรูป

รูปที่ 2.7 Spherical Robot Work Envelop Of Spherical Robot

ข้อดี

1. มีปริมาตรการทำงานมากขึ้นเนื่องจากการหมุนของแกนที่ 2 (ไหล่)
2. สามารถที่จะก้มลงมาจับชิ้นงานบนพื้นได้สะดวก

ข้อเสีย

1. มีระบบพิกัด (Coordinate) และส่วนประกอบ ที่ซับซ้อน
2. การเคลื่อนที่ และระบบควบคุมมีความซับซ้อนขึ้น

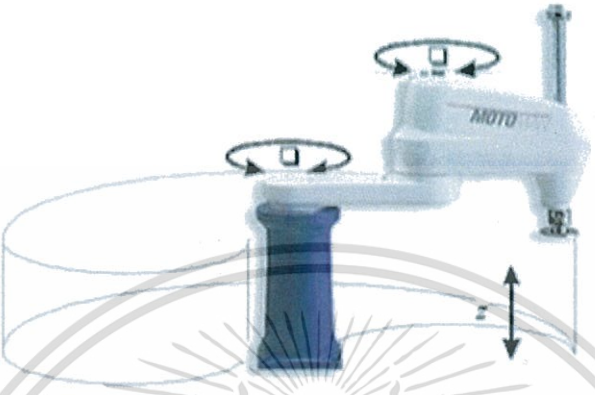
การประยุกต์ใช้งาน

ใช้ในงานที่มีการเคลื่อนที่ในแนวตั้ง (Vertical) เพียงเล็กน้อย เช่น การโหลดชิ้นงาน เข้าออกจากเครื่องปั้ม (Press) หรืออาจจะใช้งานเชื่อมจุด (Spot Welding)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 SCARA Robot

หุ่นยนต์ SCARA (Selective Compliance Assembly Robot Arm) จะมีลักษณะ แกนที่ 1 (เอว) และแกนที่ 3 (ข้อศอก) หมุนรอบแกนแนวตั้ง และแกนที่ 2 จะเป็นลักษณะ การเคลื่อนที่ขึ้นลง (Prismatic) ดังรูป หุ่นยนต์ SCARA จะเคลื่อนที่ได้รวดเร็วในแนวระนาบ และมีความแม่นยำสูง



รูปที่ 2.8 SCARA Robot Work Envelop Of SCARA Robot

ข้อดี

1. สามารถเคลื่อนที่ในแนวระนาบ และขึ้นลงได้รวดเร็ว
2. มีความแม่นยำสูง

ข้อเสีย

1. มีพื้นที่ทำงานจำกัด
2. ไม่สามารถหมุน (rotation)ในลักษณะมุมต่างๆได้
3. สามารถยกน้ำหนัก (Payload) ได้ไม่มากนัก

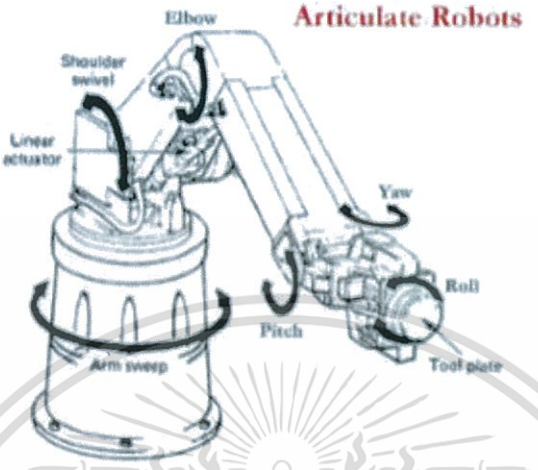
การประยุกต์ใช้งาน

เนื่องจากการเคลื่อนที่ในแนวระนาบ และขึ้นลงได้รวดเร็วจึงเหมาะกับงานประกอบ ชิ้นส่วนทางอิเล็กทรอนิกส์ ซึ่งต้องการความรวดเร็ว และการเคลื่อนที่ก็ไม่ต้องการการหมุนมาก นัก แต่จะไม่เหมาะกับงานประกอบชิ้นส่วนทางกล (Mechanical Part) ซึ่งส่วนใหญ่การ ประกอบจะอาศัยการหมุน (Rotation) ในลักษณะมุมต่างๆ นอกจากนี้ SCARA Robot ยัง เหมาะกับงานตรวจสอบ (Inspection) งานบรรจุภัณฑ์ (Packaging)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 Articulated Arm (Revolute)

ทุกแกนการเคลื่อนที่จะเป็นแบบหมุน (Revolute) รูปแบบการเคลื่อนที่จะคล้ายกับ แขนคน ซึ่งจะประกอบด้วยช่วงเอว ท่อนแขนบน ท่อนแขนล่าง ข้อมือ การเคลื่อนที่ทำให้ได้ พื้นที่การทำงาน ดังรูป



รูปที่ 2.9 Articulated Arm Robot Work Envelop Of Articulated Robot

ข้อดี

1. เนื่องจากทุกแกนจะเคลื่อนที่ในลักษณะของการหมุน ทำให้มีความยืดหยุ่นสูงในการเข้าไปยังตำแหน่งต่างๆ
2. บริเวณข้อต่อ (Joint) สามารถ Seal เพื่อป้องกันฝุ่น ความชื้น หรือน้ำ ได้ง่าย
3. มีพื้นที่การทำงานมาก
4. สามารถเข้าถึงชิ้นงานทั้งจากด้านบน ด้านล่าง

ข้อเสีย

1. มีระบบพิกัด (Coordinate) ที่ซับซ้อน
2. การเคลื่อนที่ และระบบควบคุมทำความ เข้าใจได้ยากขึ้น
3. ควบคุมให้เคลื่อนที่ในแนวเส้นตรง (Linear) ได้ยาก
4. โครงสร้างไม่มั่นคงในการเคลื่อนที่ เพราะบริเวณขอบ Work Envelope ปลายแขนจะมีการสั่น ทำให้ความแม่ยำลดลง

การประยุกต์ใช้งาน

หุ่นยนต์ชนิดนี้สามารถใช้งานได้กว้างขวางเพราะสามารถเข้าถึงตำแหน่งต่างๆ ได้ดี เช่น งานเชื่อม Spot Welding, Path Welding, งานยกของ, งานตัด, งานทากาว, งานที่มี การเคลื่อนที่ยากๆ เช่น งานพ่นสี งาน Sealing ฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ระบบนิวเมติกส์

ระบบนิวเมติกส์ หมายถึง ระบบการส่งถ่ายกำลังโดยอาศัยความดันลมเป็นตัวกลางในการส่งถ่ายกำลัง ทำให้เกิดการทำงาน หรือการเคลื่อนที่ โดยมีอุปกรณ์ เช่น กระบอกสูบ ทำหน้าที่เปลี่ยนพลังงานลมให้เป็นพลังงานกล

ระบบนิวเมติกส์มักถูกนำมาใช้ในงานอุตสาหกรรม ด้วยเหตุผลของการทำงานอย่างอัตโนมัติ และการประหยัดแรงงาน ทั้งนี้ การใช้ลมอัดนั้น ถ้ามีการประกอบร่วมกับกำลังไฟฟ้า สามารถดัดแปลงเป็นการควบคุมอัตโนมัติแบบไร้สายได้ อีกทั้งลมอัด และระบบนิวเมติกส์ยังมีข้อดีอีกหลายประการที่ส่งผลประโยชน์ในด้านอุตสาหกรรม เช่น มีค่าใช้จ่ายต่ำ มีโครงสร้างอย่างง่าย มีความสะดวกในการบำรุงรักษา เป็นต้น

ข้อดีของระบบนิวเมติกส์

- ลมอัดมีปริมาณไม่จำกัดในทุกๆแห่ง เพราะมีอยู่ทั่วไป
- ลมอัดสามารถส่งผ่านไปตามท่อที่มีระยะทางไกลๆได้สะดวก และไม่ต้องส่งกลับมา สามารถปล่อยทิ้งในบรรยากาศได้หลังจากใช้งานแล้ว
- สามารถกักเก็บลมอัดไว้ในถังได้ ทำให้สามารถนำไปใช้งานได้ตามต้องการ
- ลมอัดไม่เกิดการระเบิดหรือติดไฟง่ายเมื่อมีการรั่วซึม ดังนั้น จึงไม่มีความจำเป็นที่ต้องมีอุปกรณ์พิเศษราคาแพงเพื่อใช้ในการป้องกันการระเบิด
- ลมอัดไม่มีความไวต่อการเปลี่ยงเบนของอุณหภูมิ มีความแน่นอนในการทำงานสูง แม้จะอยู่ในสภาวะอุณหภูมิสูงมากๆก็ตาม
- เครื่องมือ และอุปกรณ์ในระบบนิวเมติกส์มีโครงสร้างแบบง่ายๆ ทำให้มีราคาถูก ทนทาน และซ่อมบำรุงรักษาได้ง่าย
- ลมอัดมีความเร็วสูง ดังนั้นอัตราความเร็วในการทำงานก็จะสูงด้วย
- สามารถควบคุมความเร็ว ความดัน และแรงของลมอัดในระบบนิวเมติกส์ได้ตามต้องการ
- เครื่องมือ และอุปกรณ์ของระบบนิวเมติกส์ สามารถใช้งานเกินกำลังได้โดยไม่เกิดการเสียหาย
- การเคลื่อนที่ในทางตรงสามารถทำได้โดยตรง

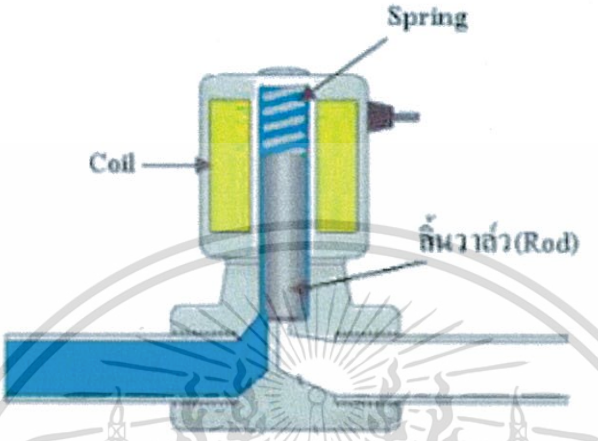
ข้อเสียของระบบนิวเมติกส์

- ลมอัดมีความชื้น และฝุ่นละออง ดังนั้นจึงต้องมีอุปกรณ์กรองความชื้น และฝุ่นละอองก่อนนำไปใช้
- ลมอัดมีเสียงดังเมื่อมีการระบายทิ้งออกสู่บรรยากาศ เพราะฉะนั้นจึงต้องมีอุปกรณ์เก็บเสียง
- ลมอัดจะประหยัดเฉพาะที่ใช้แรงขยายถึงจุดหนึ่งเท่านั้น โดยปกติแล้วใช้ความดันที่ 600 kpa (6 bar) ข้อจำกัดของแรงอยู่ที่ 20,000 –30,000 นิวตัน ขึ้นอยู่กับความเร็ว และระยะทางที่ใช้งาน
- ระบบนิวเมติกส์ จะมีความดันที่ใช้งานเพียง 4/7 bar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ลมอัดเป็นตัวกลางที่ค่อนข้างแพง เมื่อเทียบกับระบบการเปลี่ยนแปลงพลังงานอื่นๆ (อย่างไรก็ตาม จะถูกชดเชยจากอุปกรณ์บางชิ้นส่วนที่มีราคาถูก และมีสมรรถนะสูงกว่า)

2.3.1 โซลินอยด์วาล์ว



รูปที่ 2.10 รูปแสดงส่วนประกอบโซลินอยด์วาล์ว

โซลินอยด์ (Solenoid) เป็นอุปกรณ์แม่เหล็กไฟฟ้าชนิดหนึ่ง ที่มีหลักการทำงานคล้ายกับรีเลย์ (Relay) ภายในโครงสร้างของโซลินอยด์จะประกอบด้วยขดลวดที่พันอยู่รอบแท่งเหล็กที่ภายในประกอบด้วยแม่เหล็กชุดบนกับชุดล่าง เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดที่พันรอบแท่งเหล็ก ทำให้แท่งเหล็กชุดล่างมีอำนาจแม่เหล็กดึงแท่งเหล็กชุดบนลงมาสัมผัสกันทำให้ครบวงจรทำงาน เมื่อวงจรถูกตัดกระแสไฟฟ้าทำให้แท่งเหล็กส่วนล่างหมดอำนาจแม่เหล็ก สปริงก็จะดันแท่งเหล็กส่วนบนกลับสู่ตำแหน่งปกติ จากหลักการดังกล่าวของโซลินอยด์ก็จะนำมาใช้ในการเลื่อนลิ้นวาล์วของระบบนิวเมติกส์ การปิด-เปิดการจ่ายน้ำหรือของเหลวอื่นๆ โครงสร้างของ Solenoid โดยทั่วไปแบ่งออกเป็น 2 ชนิดคือ เลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยสปริง (Single Solenoid Valve) และเลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยโซลินอยด์วาล์ว (Double Solenoid Valve) ในที่นี้ใช้แบบ เลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยสปริง (Single Solenoid Valve)

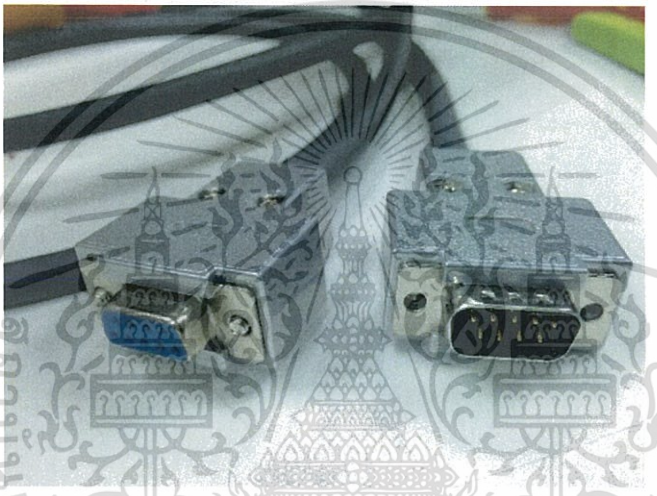
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ระบบการสื่อสาร (Communication)

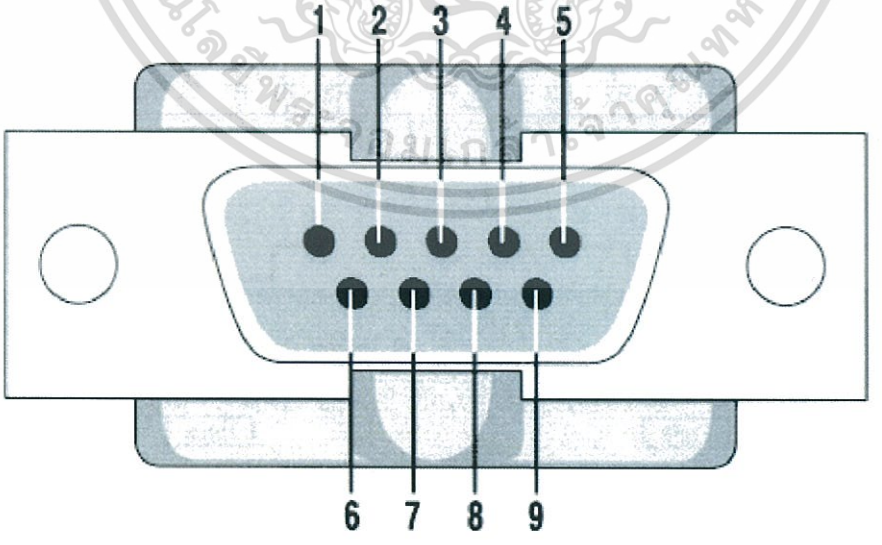
เนื่องจากการทำงานของเครื่องติดสติ๊กเกอร์บาร์โค้ดนี้ มีการทำงานร่วมกันระหว่างคอนโทรลเลอร์ของแขนกล และคอมพิวเตอร์ จึงจำเป็นต้องมีการเชื่อมต่อซึ่งกัน และกันผ่านการเชื่อมต่อสายในรูปแบบต่างๆ เพื่อรับส่งข้อมูลซึ่งกัน และกัน เช่น RS-232, LAN และ I/O terminal

2.4.1 มาตรฐานแนะนำรุ่น -232 (Recommended Standard : RS-232)

เป็นมาตรฐานการเชื่อมต่อสำหรับการส่งข้อมูลแบบอนุกรม (Serial Port) ที่กำหนดโดย EIA (Electrical Industries Association) สำหรับใช้ส่งข้อมูลเชื่อมต่อระหว่างคอมพิวเตอร์ และโมเด็มต่างๆ



รูปที่ 2.11 รูปแสดงสายเชื่อมต่อ RS-232



รูปที่ 2.12 รูปแสดงลักษณะพินของ RS-232

จากรูปที่ 2.12 แต่ละพิน (pin) จะมีหน้าที่แตกต่างกันซึ่งจะบอกรายละเอียด ดังตารางที่ 2-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2-3 ตารางบอกหน้าที่ของแต่ละพินของ RS-232

Pin	สัญญาณ	Pin	สัญญาณ
1	Date Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring indicator
5	Signal Ground		

หลักการรับ-ส่งข้อมูลของ RS232

สัญญาณรบกวนที่เกิดขึ้น ในสายนำสัญญาณ มักจะมีแรงดันเป็นบวก เมื่อเปรียบเทียบกับ กราวด์ จึงได้ทำการป้องกันสัญญาณรบกวน โดยออกแบบแรงดัน ของโลจิก "1" เป็นลบ คืออยู่ในช่วง -3V ถึง -15V ส่วนแรงดัน ของโลจิก "0" อยู่ในช่วง +3V ถึง +15V อัตราการส่งข้อมูล (Baud rate) คือความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิดต่อวินาที มันค่า Baud rate แปรผันตามความยาวสายสัญญาณดังนี้

ตารางที่ 2-4 ตารางแสดงอัตราการส่งข้อมูล (Baud rate) กับความยาวสายเชื่อมต่อ

Baud rate	ระยะสายไฟที่สามารถใช้ได้ (ฟุต)
19200	50
9600	500
4800	1000
2400	3000

การเลือกอัตราการส่งข้อมูลจึงขึ้นอยู่กับ ชนิดของสายสัญญาณ, ระยะทาง, และปริมาณสัญญาณรบกวน ในทางการใช้งานโดยทั่วไปจะใช้ 9600 บิดต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รูปแบบข้อมูลมาตรฐาน

การสื่อสารข้อมูลของ RS232 โดยมาตรฐาน จะใช้เป็นอักขระแบบ ASCII (The American Standard Code for Information Interchange)

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	Del

รูปที่ 2.13 รูปแสดงการแปลงเลขฐาน

2.4.2 ระบบเครือข่ายคอมพิวเตอร์ (Local Area Network : LAN)

แลน (อังกฤษ: Local Area Network หรือ LAN) หรือ ข่ายงานบริเวณเฉพาะที่ เป็นการเชื่อมโยงเครือข่ายคอมพิวเตอร์ถึงกันทั้งหมดโดยอาศัยสื่อกลาง มีการแบ่งแยกเครือข่ายออกเป็น 2 รูปแบบการเชื่อมโยงคือ การเชื่อมโยงภายในพื้นที่ระยะใกล้หรือ แลน (LAN) และการเชื่อมโยง ระยะไกล หรือแวน (WAN) โดยการเชื่อมโยงเครือข่ายแบบแลน มี 3 รูปแบบ คือ

- Bus มีการรับส่งข้อมูลด้วยความเร็ว 10-100 MB/sจะเชื่อมต่อกันบนสายสัญญาณเส้นเดียวกัน โดยจะมีอุปกรณ์ที่เรียกว่า T-Connector เป็นตัวแปลงสัญญาณข้อมูลเพื่อนำเข้าสู่ระบบ คอมพิวเตอร์ และ Terminator ในการปิดหัวท้ายของสายในระบบเครือข่ายเพื่อดูดซับข้อมูลไม่ให้ เกิดการสะท้อนกลับของสัญญาณ
- Star เป็นระบบที่มีเป็นการต่อแบบรวมศูนย์ โดยเครื่องคอมพิวเตอร์ทุกเครื่องจะต่อสายเข้าไปที่ อุปกรณ์ที่เรียกว่า Hub หรือ Switch โดยอุปกรณ์ที่เรียกว่า Hub หรือ Switch จะทำหน้าที่ เปรียบศูนย์กลางที่ทำหน้าที่กระจายข้อมูล โดยข้อดีของการต่อในรูปแบบนี้คือ หากสายสัญญาณ เกิดขาดในคอมพิวเตอร์เครื่องใดเครื่องหนึ่ง เครื่องคอมพิวเตอร์อื่นๆจะสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
147915

ใช้งานได้ปกติ แต่หากศูนย์กลางคือ Hub หรือ Switch เกิดเสียจะทำให้ระบบทั้งระบบไม่สามารถทำงานได้ทั้ง ระบบ

- Ring เป็นระบบที่มีการส่งข้อมูลไปในทิศทางเดียวกัน โดยจะมีเครื่อง Server หรือ Switch ในการ ปล่อย Token เพื่อตรวจสอบว่ามีเครื่องคอมพิวเตอร์ใดต้องการส่งข้อมูลหรือไม่ และระหว่างการ ส่งข้อมูลเครื่องคอมพิวเตอร์อื่นๆที่ต้องการส่งข้อมูลจะต้องทำการรอให้ข้อมูลก่อนหน้านั้นถูกส่ง ให้สำเร็จเสียก่อน

รูปที่ 2.14 รูปแสดงลักษณะของสายแลน(LAN)

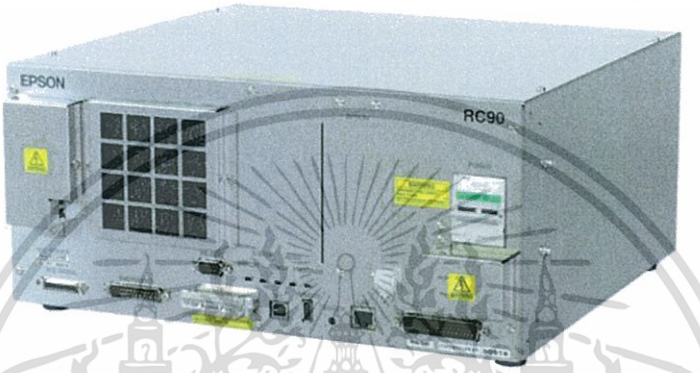
2.4.3 ระบบอินพุต และเอาท์พุต (Input / Output : I/O)

I/O ย่อมาจาก Input /Output คือ การนำเข้า และการแสดงผล ในวงการคอมพิวเตอร์คำศัพท์นี้ถูกใช้เรียก ระบบรับ และแสดงผล ของคอมพิวเตอร์ ดังนี้

- อุปกรณ์ที่เป็นตัวช่วยในการนำข้อมูลเข้าคอมพิวเตอร์ (Input device) ซึ่งทำหน้าที่รับข้อมูลจากคนหรือเซนเซอร์ และส่งต่อข้อมูลไปยัง หน่วยประมวลผลกลาง (CPU : Central Process Unit) ของ คอมพิวเตอร์ เพื่อทำการประมวลผลต่อไป ตัวอย่างเช่น คีย์บอร์ด (keyboard),เมาส์ (mouse),สแกนเนอร์ (scanner),ไมโครโฟน (microphone),กล้องเว็บแคม (webcam) และเซนเซอร์(Sensor) เป็นต้น
- อุปกรณ์ที่เป็นตัวช่วยในการแสดงผลจากคอมพิวเตอร์สู่คน เพื่อให้คนสามารถอ่าน หรือรับรู้ข้อมูลจากคอมพิวเตอร์ได้ (Output device) ซึ่งจะทำหน้าที่แสดงผลลัพธ์ที่มาจากการประมวลผลข้อมูล ของส่วนประมวลผลข้อมูลภายในคอมพิวเตอร์ ตัวอย่างเช่น เครื่องพิมพ์ (Printer), เครื่องวาด (Plotter), จอภาพ (Monitor), เครื่องฉายภาพ (LCD Projector), ลำโพง หรือการเคลื่อนที่ของแขนกล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นระบบเครือข่ายแบบเชื่อมต่อคอมพิวเตอร์เข้าด้วยกันในระยะที่จำกัด เช่น ในอาคารเดียวกัน หรือบริเวณเดียวกันที่สามารถลากสายได้ถึงกันโดยตรง เราสามารถนำมาใช้ประโยชน์ในรูปแบบของการสื่อสารระหว่างคอมพิวเตอร์ (Computer) กับคอนโทรลเลอร์ (Controller) สามารถควบคุมการเคลื่อนที่ของหุ่นยนต์ได้ และยังสามารถส่งข้อมูลในรูปแบบข้อความ (string) หรือตัวเลข (integer) จากคอมพิวเตอร์ (Computer) ไปยังคอนโทรลเลอร์ (Controller) เพื่อให้สามารถทำงาน ได้ในภาวะหรือสถานะที่เปลี่ยนแปลงไป



รูปที่ 2.15 รูปแสดงลักษณะของคอนโทรลเลอร์ (Controller) สำหรับใช้ควบคุมแขนกล (Robot)

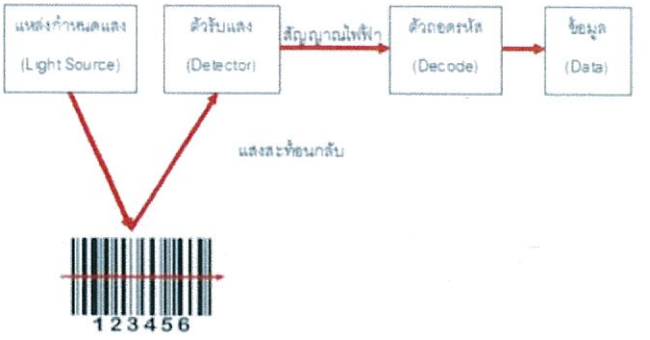
2.5 เซนเซอร์ (Sensor)

2.5.1 Barcode reader

รูปที่ 2.16 รูปแสดงลักษณะของเครื่องอ่านบาร์โค้ด

เครื่องอ่านบาร์โค้ด เป็นอุปกรณ์ที่ทำการแปลงข้อมูลจากรหัสบาร์โค้ด(Barcode) เป็นสัญญาณดิจิตอลผ่านคอมพิวเตอร์ หลังจากนั้นนำสัญญาณดิจิตอลที่ได้นั้นมาแปลเป็นข้อมูลด้วยการถอดรหัส ให้เป็นตัวเลข ตัวอักษร หรือสัญลักษณ์ที่ถูกบรรจุในข้อมูลนั้น ๆ โดยเครื่องอ่านบาร์โค้ด จะทำหน้าที่ในการผลิตลำแสงซึ่งดูดซึมส่วนที่เป็นแท่งดำทึบ และสะท้อนส่วนที่เป็นช่องว่างระหว่างแท่งสัญญาณที่สะท้อนขึ้น โดยมีหลักการทำงานเริ่มต้นจาก เครื่องอ่านบาร์โค้ดจะฉายแสงลงบนแท่งบาร์โค้ด แล้วรับแสงที่สะท้อนกลับจากแท่งบาร์โค้ด จากนั้นปริมาณแสงสะท้อนจะถูกเปลี่ยนไปเป็นสัญญาณทางไฟฟ้า แล้วถูกส่งต่อไปแปลงให้เป็นข้อมูลที่สามารถนำไปใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 รูปแสดงหลักการทำงานของเครื่องอ่านบาร์โค้ด

2.5.2 Pressure sensor

รูปที่ 2.18 รูปแสดงลักษณะของเครื่องวัดความดัน

อุปกรณ์ตรวจวัดความดันของก๊าซหรือเหลว เซ็นเซอร์จะส่งสัญญาณทางไฟฟ้าที่มีความสัมพันธ์กับความดัน เซนเซอร์ถูกสร้างให้มีเยื่อบางที่สามารถโค้งงอตามความดัน ซึ่งระดับความโค้งงอ สามารถวัดได้ด้วยการเปลี่ยนแปลงความต้านทาน หรือ การเปลี่ยนแปลงของการเก็บประจุ เซ็นเซอร์แรกเริ่มทำด้วยวิธีการประดิษฐ์โครงสร้างจุลภาคบนพื้นผิว (Surface Micromachining) สำหรับใช้งานในช่วงความดันต่างๆ เพื่อประยุกต์ใช้ในอุตสาหกรรมยานยนต์ เป็นต้น

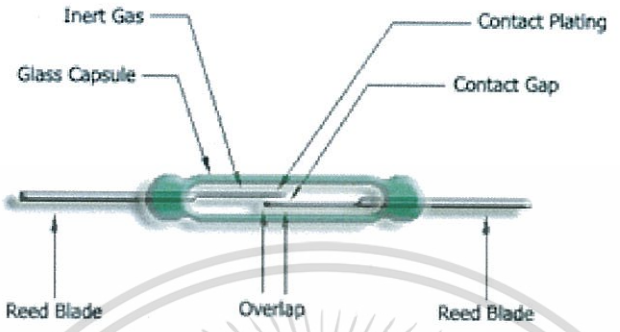
2.5.3 Reed switch

รีดสวิตช์ คือสวิตช์ที่ควบคุมการทำงานโดยใช้แม่เหล็ก ในการใช้งาน จะยึดรีดสวิตช์ไว้ที่ตัวกระบอกสูบดังรูปที่ 2.19 โดยตัวกระบอกสูบต้องทำจากอลูมิเนียม ลูกสูบต้องมีคุณสมบัติเป็นแม่เหล็กถาวร ซึ่งการใช้รีดสวิตช์มีความสะดวกในเรื่องของการติดตั้งที่ง่ายกว่าลิมิตสวิตช์ทั่วไป

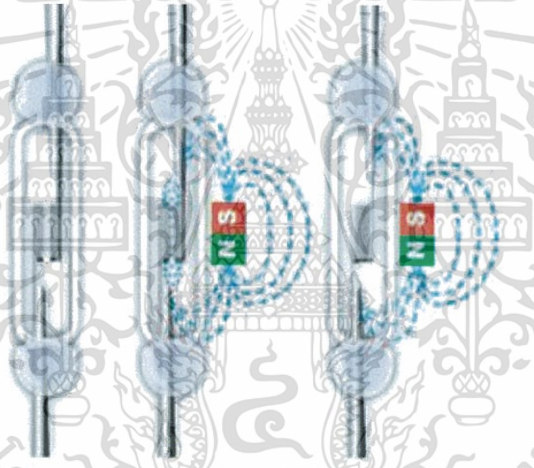
รูปที่ 2.19 รูปแสดงลักษณะของ Reed switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงาน เมื่อลูกสูบเคลื่อนที่เข้าสุด อำนาจแม่เหล็กที่ตัวลูกสูบจะไปดึงดูดให้หน้าคอนแทคของรีดสวิตซ์ให้เชื่อมต่อกัน ซึ่งปกติหน้าคอนแทคจะเป็นแบบปกติเปิด(Normally Open : NO) เมื่อลูกสูบเคลื่อนที่มาตรงกับตำแหน่งของรีดสวิตซ์ รีดสวิตซ์ก็จะปิดวงจร และเมื่อลูกสูบเคลื่อนที่ออกไปตรงกับตำแหน่งของรีดสวิตซ์ตัวนอก อำนาจแม่เหล็กของลูกสูบก็จะดึงดูดให้รีดสวิตซ์ปิดวงจรเช่นกัน



รูปที่ 2.20 รูปแสดงส่วนประกอบของ Reed switch



รูปที่ 2.21 รูปแสดงการทำงานของ Reed switch

2.5.4 Photo sensor

เป็นอุปกรณ์เซนเซอร์ที่ใช้แสงในการตรวจจับวัตถุที่ต้องการตรวจจับ ซึ่งสวิตซ์ลำแสงยังสามารถแบ่งประเภทออกได้ดังนี้

2.5.4.1 แบบตัวรับส่ง (Thru-Beam Sensor)

เป็นแบบที่ตัวรับตัวส่งแสง แยกกันอยู่คนละฝั่ง และให้วัตถุที่ตรวจจับเคลื่อนที่ผ่านระหว่างกลางเพื่อขวางการส่ง และรับสัญญาณ

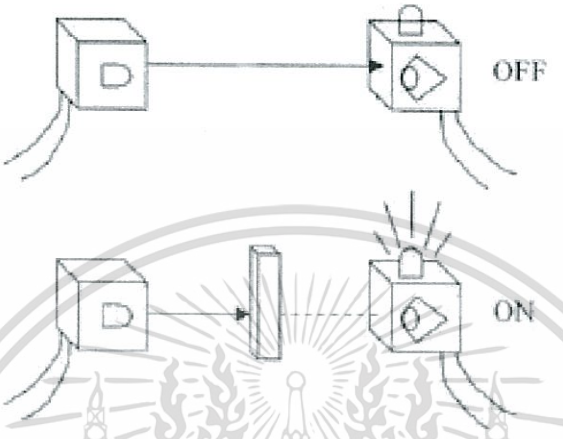
ข้อดีของการตรวจจับแบบ Thru-Beam Sensor

- ระยะการตรวจจับไกลมีความแม่นยำในการตรวจจับสูง
- สามารถตรวจจับวัตถุที่มีขนาดเล็กได้
- สี และพื้นผิวของวัตถุไม่มีผลกระทบต่อการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสียของการตรวจจับแบบ Thru-Beam Sensor

- ใช้เนื้อที่การติดตั้งมาก
- เสียค่าใช้จ่ายในการเดินสายไฟสูง
- การปรับแต่งค่อนข้างยาก
- ไม่สามารถตรวจจับวัตถุโปร่งใสได้



รูปที่ 2.22 รูปแสดงการทำงานของ Thru - Beam Type

2.5.4.2 Diffuse – Reflective Beam Sensor

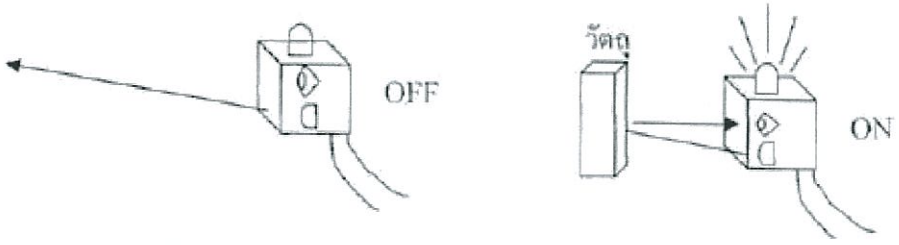
เป็นแบบที่ตัวรับตัวส่งแสงรวมภายในตัวเดียวกัน เมื่อลำแสงไปกระทบกับผิวของวัตถุแล้วสะท้อนกลับเข้ามายังตัวรับแสงที่อยู่เดียวกัน ทำให้สวิตช์แสงมีสัญญาณเอาท์พุตเปลี่ยนแปลงไป แต่ถ้าหากไม่มีวัตถุมาบังแสงภายในระยะตรวจจับ สวิตช์ลำแสงก็จะไม่ทำงาน

ข้อดีของการตรวจจับแบบ Diffuse – Reflective Beam Sensor

- ไม่ต้องทำการจัดตำแหน่งของตัวรับตัวส่งให้ตรงกัน
- สามารถตรวจจับวัตถุโปร่งใสได้
- ใช้พื้นที่ในการติดตั้งน้อย

ข้อเสียของการตรวจจับแบบ Diffuse – Reflective Beam Sensor

- ระยะการตรวจจับสั้น
- การตรวจจับขึ้นอยู่กับสี และการสะท้อนที่ผิวของวัตถุ



รูปที่ 2.23 แสดงการทำงานของ Diffuse – Reflective Beam Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4.3 Retro – Reflective Beam Sensor

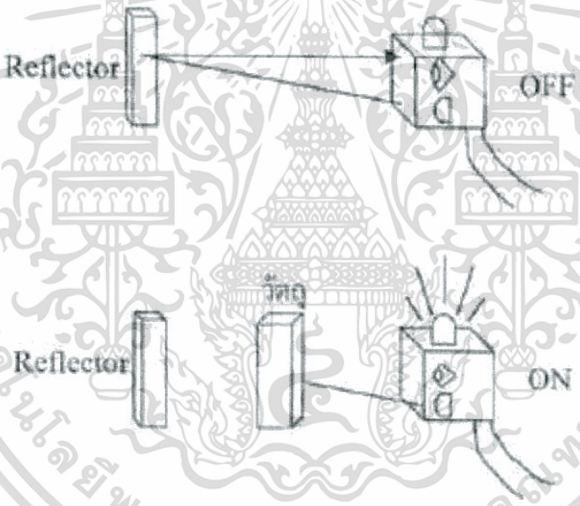
สวิตช์ลำแสงจะตรวจจับวัตถุที่มีการสะท้อนแสงน้อยกว่าแผ่นสะท้อนซึ่งวัตถุจะต้องอยู่ระหว่างสวิตช์ลำแสง และแผ่นสะท้อน

ข้อดีของการตรวจจับแบบ Retro – Reflective Beam Sensor

- ใช้เนื้อที่ในการติดตั้งน้อย
- ค่าใช้จ่ายในการเดินสายไฟต่ำ
- การปรับแต่งทำได้ง่าย

ข้อเสียของการตรวจจับแบบ Retro – Reflective Beam Sensor

- จำเป็นต้องใช้แผ่นสะท้อนแสงแบบพิเศษ
- ความแม่นยำในการตรวจจับต่ำกว่าแบบ Thru-Beam
- ไม่สามารถตรวจจับวัตถุที่มีการสะท้อนแสงที่ดีกว่าแผ่นสะท้อน



รูปที่ 2.24 รูปแสดงการทำงานของ Retro – Reflective Beam Sensor

ซึ่งสวิตช์ลำแสงทั้ง 3 ชนิดมีลักษณะสำคัญดังนี้

1. ตรวจจับได้โดยไม่ต้องสัมผัส

   ซึ่งส่งผลให้อุปกรณ์ตรวจจับมีอายุการใช้งานนานขึ้นเนื่องจากไม่มีกลไกในการเคลื่อนไหวจึงทำให้ไม่มีการสึกหรอเกิดขึ้น และวัตถุบางชนิดไม่สามารถที่จะสัมผัสได้ เพราะเมื่อการสัมผัสระหว่างวัตถุกับอุปกรณ์ตรวจจับอาจทำให้วัตถุเสียหายได้

2. ระยะในการตรวจจับวัตถุไกล

   เนื่องจากไม่มีกลไกหรือการเคลื่อนไหวทางกล จึงทำให้สวิตช์ลำแสงมีระยะการตรวจจับไกลขึ้นเพราะใช้แสงเป็นตัวกลางในการตรวจจับ ระยะในการตรวจจับของสวิตช์ลำแสง ถ้าเป็นชนิดที่มีตัวรับส่งจะมีระยะไกลสุด 500 เมตร ส่วนชนิดที่ใช้กับการสะท้อนกลับระยะการตรวจจับไกลสุดจะได้ 3 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สามารถตรวจจับวัตถุได้แทบทุกชนิด

สวิตช์ลำแสงไม่คำนึงถึงชนิดของวัตถุนั้นๆ ว่าเป็นโลหะหรืออโลหะ เพียงแต่วัตถุที่ตรวจจับสามารถ ทำให้แสงของสวิตช์ลำแสงเปลี่ยนแปลงไปได้ก็จะสามารถตรวจจับได้

4. มีการตรวจจับที่รวดเร็ว

สวิตช์ลำแสงใช้แสงเป็นตัวกลางในการตรวจจับวัตถุร่วมกับอุปกรณ์อิเล็กทรอนิกส์เป็นที่ทราบกันดีแล้วว่าแสงมีความเร็วในการเดินทางสูงมาก จึงทำให้สวิตช์ชนิดนี้มีระยะเวลาในการตอบสนองต่อการตรวจจับสั้นมาก

5. สามารถตรวจแยกสีได้

สวิตช์ลำแสง ใช้แสงเป็นตัวกลางในการแยกสีซึ่งจะตรวจจับจากอัตราส่วนระหว่างการสะท้อนหรือดูดกลืนสีไม่เท่ากัน จึงทำให้สวิตช์ลำแสงสามารถตรวจแยกสีได้

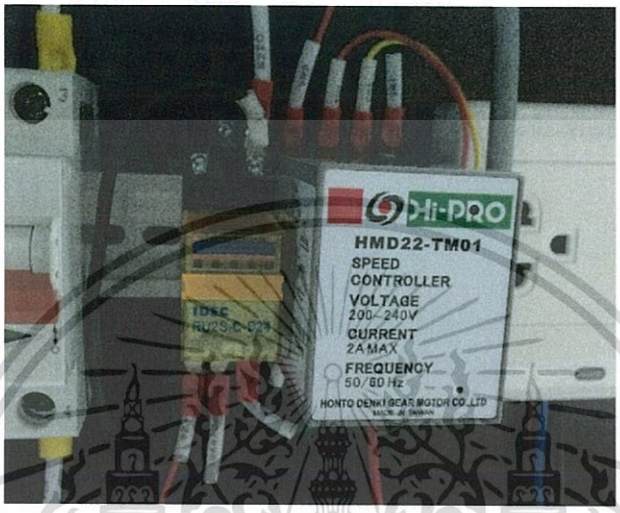
6. มีคามแม่นยำในการตรวจจับสูง

เนื่องจากเป็นระบบที่ใช้แสงร่วมกับวงจรอิเล็กทรอนิกส์ที่ผลิตโดยใช้เทคโนโลยีชั้นสูง จึงทำให้สวิตช์ลำแสงมีความแม่นยำในการตรวจจับสูงมาก แต่สวิตช์ลำแสงจะมีข้อเสียอยู่คือ ถ้าหากบริเวณหน้าเลนส์ของสวิตช์ลำแสงสกปรก หรือบางครั้งมีการรบกวนของแสงที่ส่องไปเพื่อตรวจจับวัตถุ สวิตช์ลำแสงอาจทำงานผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ชุดควบคุมความเร็ว DC มอเตอร์

ใช้สำหรับควบคุมความเร็วในการหมุนของมอเตอร์ที่ติดตั้งบนคอนเวเยอร์(Conveyor) โดยควบคุมระดับความเร็วผ่านตัวต้านทานชนิดปรับค่าได้ และสั่งการควบคุมไปยังชุดควบคุม 8 pin speed controller เพื่อควบคุมความเร็วในการหมุนของมอเตอร์ได้ตามต้องการ



รูปที่ 2.25 รูปแสดงชุดควบคุมความเร็วมอเตอร์แบบ 8 พิน

ตารางที่ 2-5 Specification of 8 pin speed controller

Type	HMD21	HMD22
Voltage	Single Phase AC110V+-10% 50/60Hz	Single Phase AC220V+-10% 50/60Hz
Current	2A Below	
Gentry Start and Stop Time Frame	Approximately 0-5 Seconds Above	
Range of Speed Control	50Hz:90~1400r/min 60Hz:90~1600r/min	
Temperature Ability	0°C～40°C (No Frozen)	
Humid Ability	Below 85% (No frosting happen)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 รีเลย์(Relay)

รีเลย์ (Relay) เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย

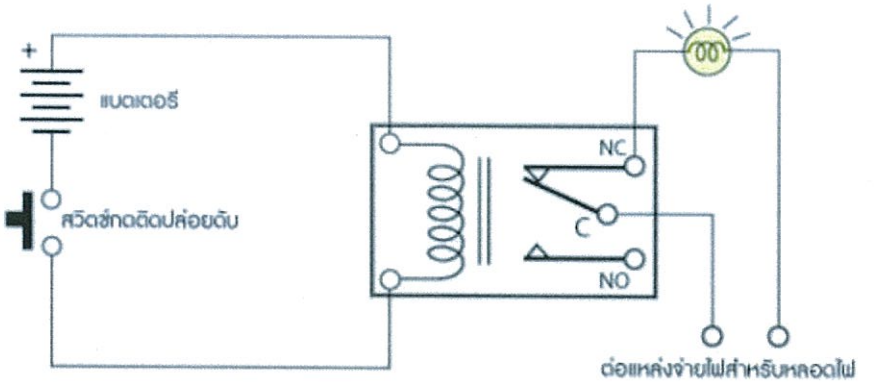
รูปที่ 2.26 รูปแสดงลักษณะทั่วไปของรีเลย์

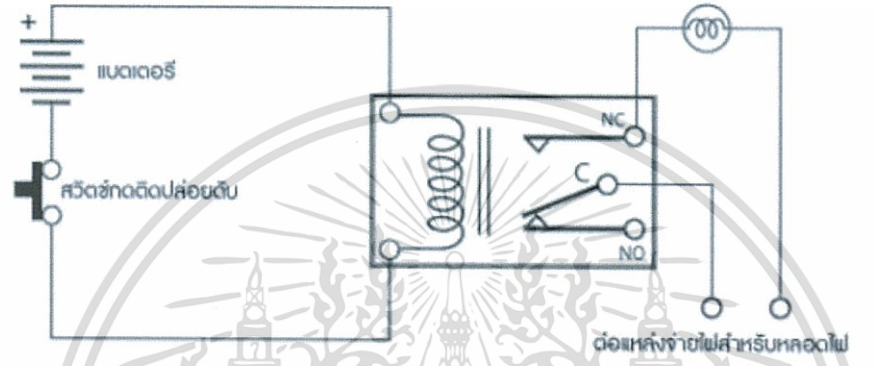
ประกอบด้วยส่วนสำคัญ 2 ส่วนหลักก็คือ

- ส่วนของขดลวด (coil) เหนี่ยวนำกระแสต่ำ ทำหน้าที่สร้างสนามแม่เหล็กไฟฟ้าให้แกนโลหะไปกระทุ้งให้หน้าสัมผัสต่อกัน ทำงานโดยการรับแรงดันจากภายนอกต่อคร่อมที่ขดลวดเหนี่ยวนำนี้ เมื่อขดลวดได้รับแรงดัน จะเกิดสนามแม่เหล็กไฟฟ้าทำให้แกนโลหะด้านในไปกระทุ้งให้แผ่นหน้าสัมผัสต่อกัน โดยค่าแรงดันที่รีเลย์ต้องการขึ้นกับชนิด และรุ่นตามที่ผู้ผลิตกำหนด
- ส่วนของหน้าสัมผัส (contact) ทำหน้าที่เหมือนสวิตช์จ่ายกระแสไฟให้กับอุปกรณ์ที่ต้องการจุดต่อของรีเลย์จะประกอบไปด้วย จุดต่อNC จุดต่อNO และจุดต่อC ซึ่งจุดต่อ NC ย่อมาจาก **normal close** หมายความว่าปกติปิด หรือ หากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา จุดต่อ NO ย่อมาจาก **normal open** หมายความว่าปกติเปิด หรือหากยังไม่จ่ายไฟให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิดปิดเช่นโคมไฟ จุดต่อ C ย่อมากจาก **common** คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ

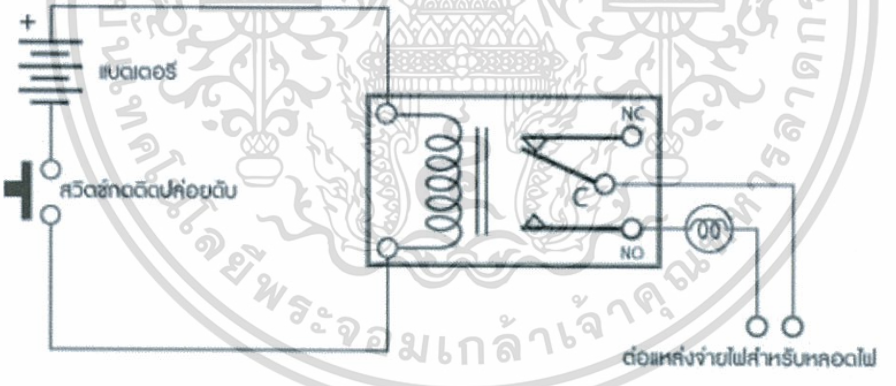
การนำรีเลย์ไปใช้งาน สามารถนำไปใช้ได้ในหลายรูปแบบ ขึ้นอยู่กับลักษณะการต่อแหล่งจ่ายไฟ เช่น ในขณะที่รีเลย์ไม่ทำงาน ถ้าต่อ แหล่งจ่ายไฟไปยัง NC และ Common และต่ออนุกรมกับหลอดไฟ จะทำให้หลอดไฟติด แต่หากรีเลย์ทำงาน หลอดไฟนั้นจะดับลง ในทางตรงกันข้าม หากต่อแหล่งจ่ายไฟไปยัง NO และ Common และต่ออนุกรมกับหลอดไฟ ในขณะที่รีเลย์ไม่ทำงาน หลอดไฟจะไม่ติด จนกว่ารีเลย์จะทำงาน แสดงดังรูปที่ 2.27 และ รูปที่ 2.28

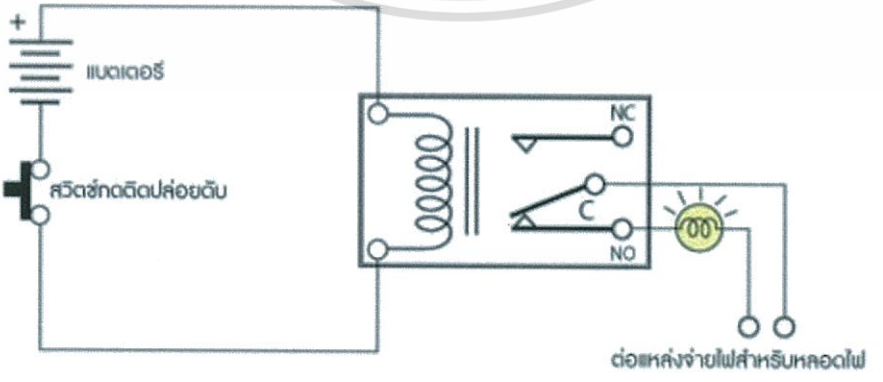
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 2.27 การต่อแหล่งจ่ายไฟเข้ากับ NC และ Common





รูปที่ 2.28 การต่อแหล่งจ่ายไฟเข้ากับ NO และ Common

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 พื้นฐานภาษา SPEL

ภาษา SPEL เป็นภาษาที่ใช้สำหรับควบคุมการทำงานของคอนโทรลเลอร์(Controller) สำหรับสั่งการแขนกล(Epson robot) และเอาท์พุต เพื่อให้เกิดการดำเนินงานตามที่ต้องการ ซึ่งภาษา SPEL ที่ใช้ควบคุมคอนโทรลเลอร์นั้น มีความสะดวกในการใช้งาน โดยผ่าน โปรแกรม EPSON RC+ 5.0 หรือ EPSON RC+ 7.0 (การใช้ขึ้นอยู่กับรุ่นของคอนโทรลเลอร์)

คำสั่งที่ใช้ควบคุมการทำงานของคอนโทรลเลอร์นั้น สามารถแบ่งได้เป็นหัวข้อสำคัญดังนี้

- System management commands
- Robot control commands
- Input/output commands
- Program control commands
- Program execution commands
- Logical operations
- Variable commands

โดยสามารถศึกษาการใช้งานคำสั่งต่างๆได้จากภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีดำเนินงาน

การศึกษาเกี่ยวกับ การควบคุมแขนกลสำหรับติดสติ๊กเกอร์บาร์โค้ดแบบอัตโนมัติ ในวิชาโครงงานสหกิจศึกษา หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมระบบการผลิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง มีขั้นตอนการศึกษาและดำเนินงาน ของเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ ซึ่งสามารถแบ่งได้เป็น 4 ขั้นตอน ดังนี้

3.1 ศึกษาและรวบรวมข้อมูล
3.2 การออกแบบเครื่อง
3.3 การติดตั้งอุปกรณ์ทางไฟฟ้าและต่อสายไฟภายในแผงควบคุม
3.4 เขียนโปรแกรมควบคุม

3.1 ศึกษาและรวบรวมข้อมูล

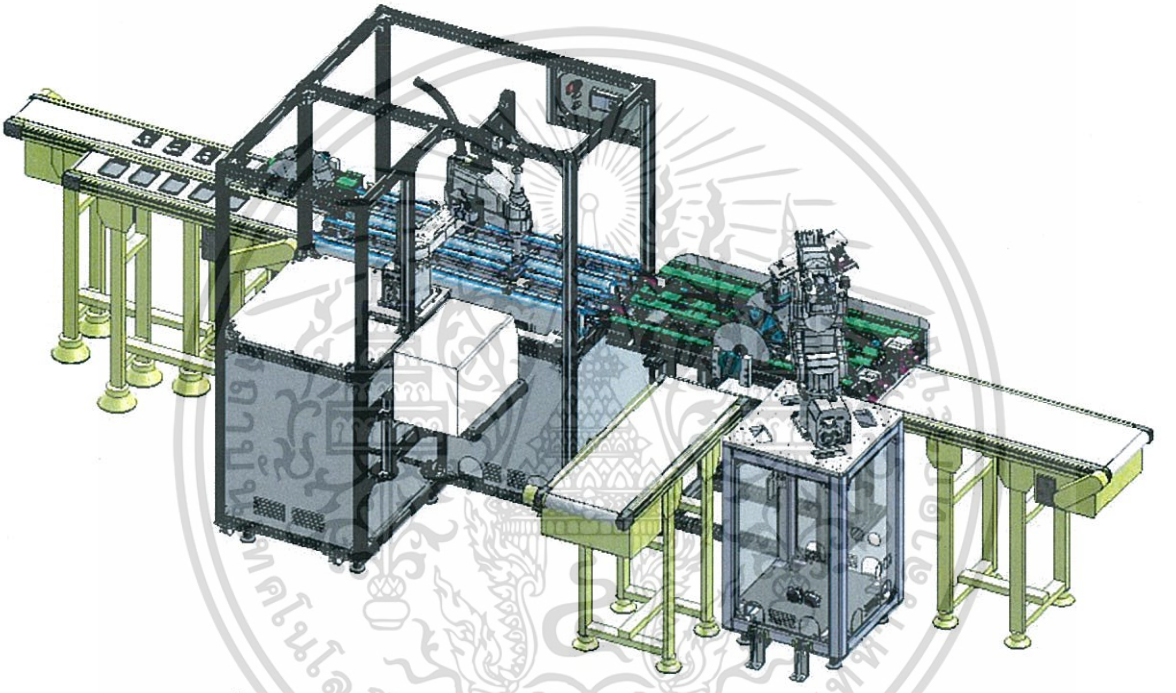
ในสายการผลิตของโรงงาน (CELL 2) ได้มีการดำเนินการสำหรับการประกอบฮาร์ดดิสก์ (Hard disk) เข้ากับฝาครอบ (Lower case) ซึ่งจำเป็นต้องมีการติดสติ๊กเกอร์บาร์โค้ดบนฝาครอบ เพื่อระบุรหัสของฮาร์ดดิสก์ (Serial number) แต่ละตัวที่อยู่ภายในฝาครอบ และต้องมีความถูกต้องและตำแหน่งที่แม่นยำในการติดสติ๊กเกอร์ ซึ่งมีขั้นตอนในการดำเนินการเริ่มต้นจาก ใช้ตัวอ่านบาร์โค้ด (Barcode reader) ในการอ่านบาร์โค้ดบนฮาร์ดดิสก์ (Serial number) และสั่งให้พิมพ์บาร์โค้ดลงบนฉลาก (Label) แล้วทำการติดสติ๊กเกอร์ลงบนฝาครอบ

ในการใช้แรงงานมนุษย์ในการดำเนินการนั้น ใช้ระยะเวลา 6 วินาทีต่อการดำเนินการ 1 ชิ้น และยังมีข้อจำกัดในด้านระยะเวลาที่สามารถทำงานได้ต่อวัน แต่เนื่องจากในปัจจุบัน โรงงานต้องการความเร็วในการผลิตที่มากขึ้น เพื่อประสิทธิภาพในการรองรับการผลิตสินค้าให้ได้จำนวนมากขึ้นในอนาคต และเพื่อเป็นการสร้างความน่าเชื่อถือ และสร้างมาตรฐานให้เกิดความพึงพอใจต่อลูกค้า อีกทั้งยังเป็นการลดต้นทุนในระยะยาว โรงงานจึงจำเป็นต้องมีการเปลี่ยนมาใช้ระบบอัตโนมัติ โดยจะมีการทำงานเป็นขั้นตอน ดังนี้ เริ่มต้นด้วยการสแกนบาร์โค้ดบนฮาร์ดดิสก์ และสั่งพิมพ์บาร์โค้ด และดำเนินการลำเลียงฮาร์ดดิสก์และฝาครอบมายังตำแหน่งที่ต้องการติดสติ๊กเกอร์ด้วยการควบคุมระบบนิวเมติกส์ (Pneumatics) ผ่านคอนโทรลเลอร์โรบอท (Robot controller) โดยจำนวนบัฟเฟอร์ (Buffer) ที่ต้องพิมพ์ก่อนฉลากตัวแรกจะมาถึงตำแหน่งที่แขนกลสามารถหยิบได้นั้นมีทั้งสิ้น 7 ตัว หลังจากฉลากตัวแรกมาถึงตำแหน่งที่แขนกลสามารถหยิบได้แล้ว จึงใช้ระบบวิทัศน์ (Vision) ในการค้นหาตำแหน่งความคลาดเคลื่อนของฉลากและความสมบูรณ์ของฉลากก่อน แล้วจึงสั่งให้แขนกลหยิบสติ๊กเกอร์ไปติดบนฝาครอบฮาร์ดดิสก์

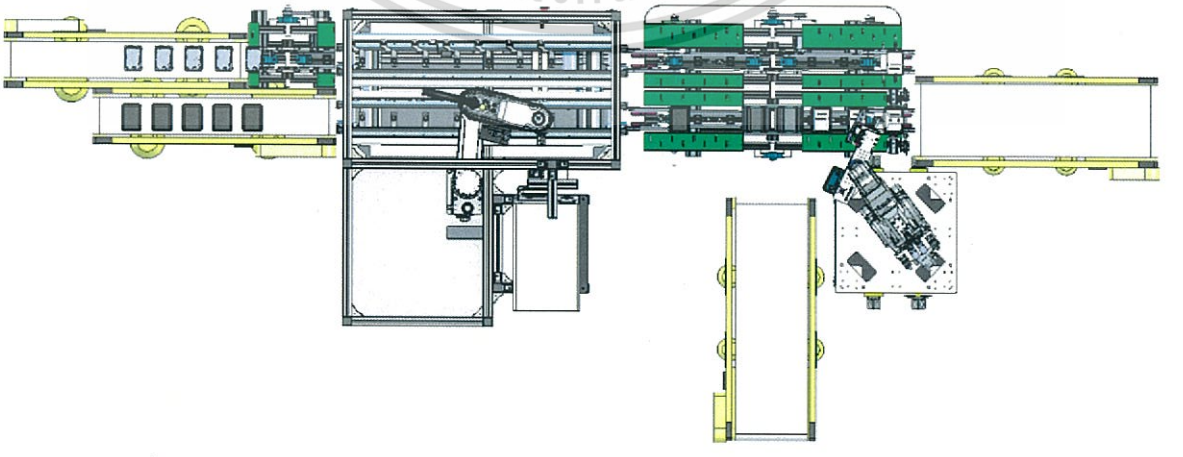
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบเครื่อง

สายการประกอบฮาร์ดดิสก์ (CELL 2) จะเริ่มต้นด้วยสถานีสำหรับการโหลดโปรแกรม (Firmware) ใส่ในฮาร์ดดิสก์ และส่งฮาร์ดดิสก์ผ่านเครื่องหมุน (Rotator) สำหรับการพลิกด้านของฮาร์ดดิสก์เพื่อความสะดวกในการสแกนบาร์โค้ดที่อยู่บนฮาร์ดดิสก์ และส่งต่อไปยังสถานีติดสติ๊กเกอร์ ซึ่งในขณะเดียวกันก็ทำการลำเลียงฝาครอบ (Lower case) ไปยังสถานีติดสติ๊กเกอร์ด้วย แล้วจึงทำการติดบาร์โค้ดที่ฝาครอบ (Lower case) และส่งต่อไปที่สถานีประกอบฮาร์ดดิสก์กับฝาครอบ โดยเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัตินี้ จะเป็นเครื่องที่รับตัวฮาร์ดดิสก์และฝาครอบมาจากสถานีก่อนหน้า และดำเนินการติดสติ๊กเกอร์บาร์โค้ดบนฝาครอบแล้วทำการส่งต่อไปยังสถานีถัดไป ดังรูปที่ 3.1



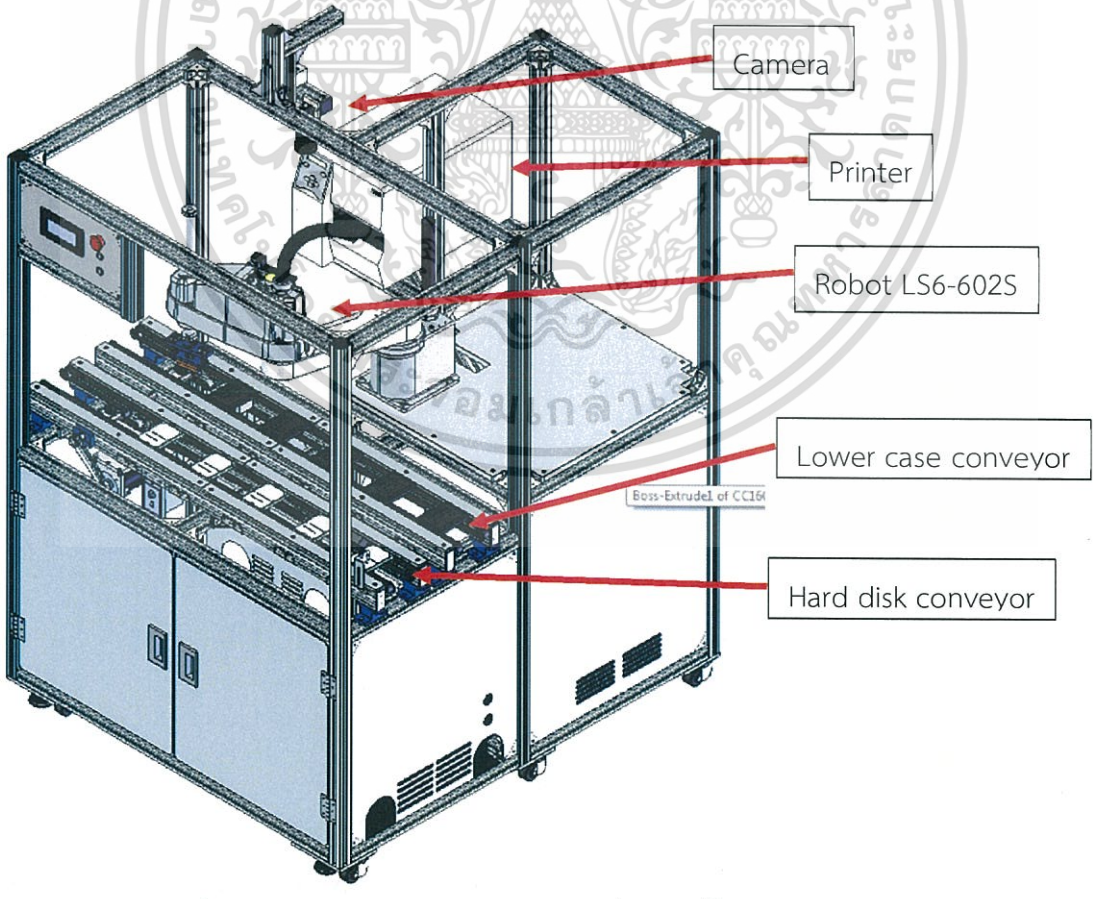
รูปที่ 3.1 รูปแสดงโครงสร้างโดยรวมของสายการประกอบฮาร์ดดิสก์



รูปที่ 3.2 รูปแสดงโครงสร้างโดยรวมของสายการประกอบฮาร์ดดิสก์ (Top view)

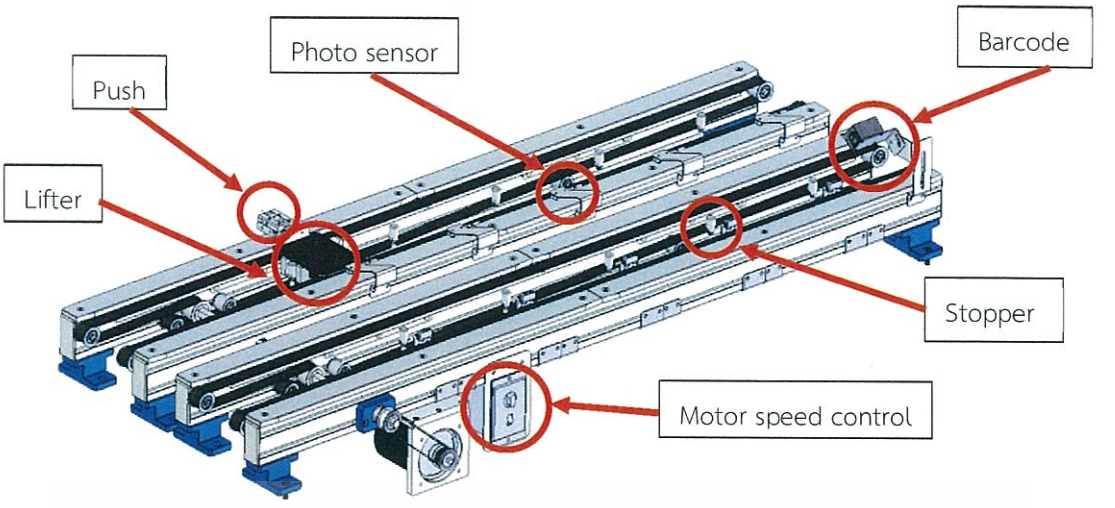
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัตินั้น มีขั้นตอนการทำงานเริ่มต้นจาก การลำเลียงฮาร์ดดิสก์ (Hard disk) และลำเลียงฝาครอบ (Lower case) ด้วยสายพาน (Conveyor) จากเครื่องก่อนหน้า โดยเมื่อฮาร์ดดิสก์เข้ามาถึงสถานีนี้ จะเริ่มต้นด้วยการอ่านค่าบาร์โค้ดบนฮาร์ดดิสก์โดยใช้ตัวสแกนบาร์โค้ด (Barcode reader) แล้วทำการสั่งพิมพ์ฉลาก (Label) และทำการลำเลียงผ่านบัฟเฟอร์ (Buffer) ทั้งหมด 7 ตัว เนื่องจากการสั่งพิมพ์ฉลาก (Label) แต่ละชิ้นนั้น จะมีระยะห่างหลังพิมพ์เสร็จในเครื่องพิมพ์กับตำแหน่งที่แขนกลสามารถหยิบสติ๊กเกอร์ได้เป็นระยะเท่ากับฉลาก 7 ชิ้น จึงจำเป็นต้องมีบัฟเฟอร์เพื่อการพิมพ์ที่ต่อเนื่องของเครื่องพิมพ์บาร์โค้ด ซึ่งการเคลื่อนที่ของฮาร์ดดิสก์ผ่านบัฟเฟอร์แต่ละตำแหน่งนั้นจะถูกควบคุมการเคลื่อนที่ด้วยตัวหยุด (Stopper) และเคลื่อนที่ไปยังตำแหน่งบัฟเฟอร์สุดท้ายสำหรับรอปล่อยสู่สถานีถัดไป โดยในส่วนของฝาครอบ (Lower case) นั้นจะถูกลำเลียงไปยังตำแหน่งสำหรับรอการติดสติ๊กเกอร์ ซึ่งจะถูกยกด้วยแท่นยก (Lifter) และตัวผลัก (Push) เพื่อจำกัดตำแหน่งการเคลื่อนที่ของฝาครอบ และเมื่อทั้งฝาครอบและฮาร์ดดิสก์ เคลื่อนที่มายังตำแหน่งที่พร้อมสำหรับติดสติ๊กเกอร์แล้ว ก็จะสั่งให้ระบบวิทัศน์ (Vision system) อ่านค่าบาร์โค้ด (Barcode) และส่งค่ามาตรวจสอบกับข้อมูลบาร์โค้ดที่อ่านได้บนฮาร์ดดิสก์ และค้นหาความคลาดเคลื่อนของตำแหน่งสติ๊กเกอร์ที่อยู่บนเครื่องพิมพ์ หากถูกต้องจะสั่งให้แขนกล (Robot) ทำการหยิบสติ๊กเกอร์โดยปรับตำแหน่งการหยิบจากค่าความคลาดเคลื่อนที่ได้รับจากระบบวิทัศน์ แล้วจึงนำมาติดบนฝาครอบ (Lower case) และรอที่จะปล่อยทั้งฮาร์ดดิสก์ (Hard disk) และฝาครอบ (Lower case) ไปสู่สถานีถัดไปพร้อมๆกัน จะแสดงให้เห็นภาพรวมของเครื่องดังรูปที่ 3.3



รูปที่ 3.3 รูปแสดงลักษณะโดยรวมของเครื่องติดสติ๊กเกอร์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

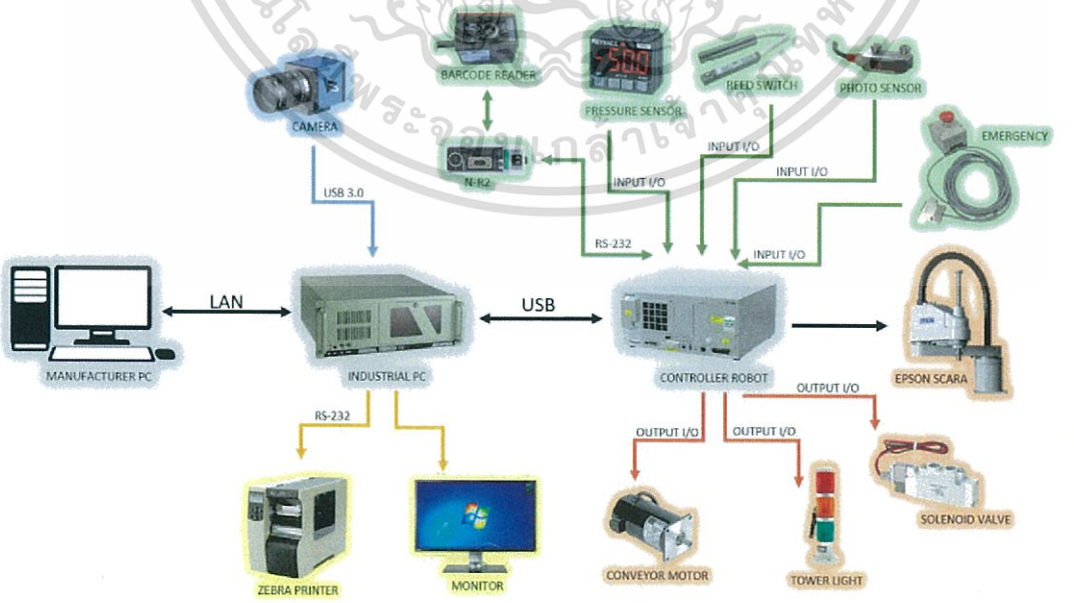


รูปที่ 3.4 รูปแสดงสายพานลำเลียงฮาร์ดดิสก์ (Hard disk) และฝาครอบ (Lower case)

การเชื่อมต่อต่างๆภายในเครื่องติดสติ๊กเกอร์นั้น จะประกอบด้วยตัวประมวลผลหลักๆ 2 ส่วน คือ คอนโทรลเลอร์แขนกล (Robot Controller) และ คอมพิวเตอร์ (Computer) โดยสามารถทำการเชื่อมต่อและส่งข้อมูลกันด้วยระบบแลน (LAN) หรือสามารถเชื่อมต่อผ่านสาย USB ก็ได้

การควบคุมการสั่งการเอาท์พุทต่างๆ จะถูกกระทำผ่านพอร์ต I/O ของคอนโทรลเลอร์ เช่น การควบคุมการเคลื่อนที่ของแขนกล การควบคุมการทำงานของโซลินอยด์วาล์ว, มอเตอร์และไฟสัญญาณต่างๆ โดยรับอินพุทมาจากเซนเซอร์ต่างๆ เช่น ตัวอ่านบาร์โค้ด (Barcode reader) , เซนเซอร์วัดแรงดัน, ลีดสวิตช์ (Reed switch) , เซนเซอร์แสง (Photo sensor) และปุ่มEmergency เพื่อทำการประมวลผล

โดยคอมพิวเตอร์นั้นจะทำหน้าที่ประมวลผลในส่วนของระบบวิทัศน์ (Vision) , การเชื่อมต่อกับคอมพิวเตอร์ของโรงงานและควบคุมการพิมพ์ของเครื่องพิมพ์บาร์โค้ด โดยแสดงผลผ่านหน้าจอ GUI (Graphic user interface) ซึ่งจะแสดงให้เห็น ดังรูปที่ 3.5



รูปที่ 3.5 รูปแสดงการเชื่อมต่อต่างๆภายในเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.5 ที่แสดงการเชื่อมต่อต่างๆภายในเครื่อง จะเห็นว่า การควบคุมเอาท์พุทและอินพุท รวมถึงขั้นตอนการทำงานต่างๆ ต้องใช้เซนเซอร์และอุปกรณ์ต่างๆเพื่อควบคุมการทำงานในเครื่องติดสติ๊กเกอร์ โดยจำนวนที่ต้องใช้นั้นจะแสดงดังตารางที่ 3-1

ตารางที่ 3-1 ตารางแสดงอุปกรณ์ที่ใช้ภายในเครื่องติดสติ๊กเกอร์อัตโนมัติ

รายชื่ออุปกรณ์	จำนวน
1. แขนกล (Robot)	1
2. คอนโทรลเลอร์ (Robot controller)	1
3. คอมพิวเตอร์ (Computer)	1
4. ตัวแปลงไฟ 24 โวลต์	1
5. พอร์ต Controller I/O	2
6. I/O Terminal	2
7. ปุ่ม Emergency	1
8. เซนเซอร์แสง (Photo sensor)	17
9. เซนเซอร์วัดความดัน (Pressure sensor)	1
10. ลีดสวิตช์ (Reed switch)	27
11. กล้อง (Camera)	1
12. เมาส์ (Mouse)	1
13. แป้นพิมพ์ (Keyboard)	1
14. ตัวอ่านบาร์โค้ด (Barcode reader)	1
15. รีเลย์ (Relay)	7
16. โซลินอยด์วาล์ว (Solinoid valve)	22
17. กระบอกสูบ 2 ทาง	19
18. หลอดไฟ (Bar light)	1
19. ทาวเวอร์ไลท์ (Tower light)	1
20. ชุดสายพาน (Conveyor)	1
21. เครื่องพิมพ์บาร์โค้ด (Printer)	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมสัญญาณเอาท์พุทและอินพุทของคอนโทรลเลอร์นั้น จำเป็นต้องกำหนดพอร์ตการเชื่อมต่อหรือตำแหน่งของขา I/O ก่อน เพื่อแบ่งและแยกแยะหน้าที่ของแต่ละพอร์ตให้ชัดเจน ทำให้มีความสะดวกและลดการสับสนในการเขียนโปรแกรมและการต่อสาย (Wiring) โดยคอนโทรลเลอร์นั้นได้เพิ่มการ์ด I/O มาอีก 1 ตัว เนื่องจากจำเป็นต้องใช้อินพุทและเอาท์พุทจำนวนมาก และยังมีการต่อ I/O ที่ออกจากคอนโทรลเลอร์เข้ากับ Terminal I/O ทำให้มีชื่อตำแหน่งตัวแปรที่ใช้ และตำแหน่งพิน (Pin) ของ Terminal I/O ที่แตกต่างกัน โดยจะแบ่งออกเป็นชื่อสัญญาณที่ใช้ในโปรแกรมและชื่อตำแหน่งพิน (Pin) ที่ต่อออกจากขา Terminal I/O แสดงดังตาราง

ตารางที่ 3-2 ตารางแสดงอินพุทของบอร์ด 1

I/O INPUT ROBOT		
I/O Board 1		
PIN	Signal Name	Name
1	InCom 0	Input common No. 0 to 7
2	In No.0	Photo sensor Lifter
3	In No.1	Photo sensor Lower case 1
4	In No.2	Photo sensor Lower case 2
5	In No.3	Photo sensor Lower case 3
6	In No.4	Photo sensor Lower case 4
7	In No.5	Photo sensor Lower case 5
8	In No.6	Photo sensor Lower case 6
9	In No.7	Photo sensor Lower case 7
18	InCom 1	Input common No. 8 to 15
19	In No.8	Reed Switch Stopper Lifter up
20	In No.9	Reed Switch Stopper Lifter down
21	In No.10	Reed Switch Lifter up
22	In No.11	Reed Switch Lifter down
23	In No.12	Reed Switch Lower case 1 up
24	In No.13	Reed Switch Lower case 2 up
25	In No.14	Reed Switch Lower case 3 up
26	In No.15	Reed Switch Lower case 4 up
34	InCom 2	Input common No. 16 to 23
35	In No.16	Reed Switch Lower case 5 up
36	In No.17	Reed Switch Lower case 6 up
37	In No.18	Reed Switch Lower case 7 up
38	In No.19	Reed Switch Table Feed up
39	In No.20	Reed Switch Push up
40	In No.21	Lower case In Ready UPLINK
41	In No.22	Hard disk In Ready UPLINK
42	In No.23	Hard disk Out Request DOWNLINK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3-3 ตารางแสดงอินพุทของบอร์ด 2

I/O INPUT ROBOT		
I/O Board 2		
PIN	Signal Name	Name
1	InCom 0	Input common No. 0 to 7
2	In No.64	Photo sensor Stopper 1
3	In No.65	Photo sensor Stopper 2
4	In No.66	Photo sensor Stopper 3
5	In No.67	Photo sensor Stopper 4
6	In No.68	Photo sensor Stopper 5
7	In No.69	Photo sensor Stopper 6
8	In No.70	Photo sensor Stopper 7
9	In No.71	Photo sensor Stopper 8
18	InCom 1	Input common No. 8 to 15
19	In No.72	Reed Switch Stopper1 up
20	In No.73	Reed Switch Stopper1 down
21	In No.74	Reed Switch Stopper2 up
22	In No.75	Reed Switch Stopper2 down
23	In No.76	Reed Switch Stopper3 up
24	In No.77	Reed Switch Stopper3down
25	In No.78	Reed Switch Stopper4 up
26	In No.79	Reed Switch Stopper4 down
34	InCom 2	Input common No. 16 to 23
35	In No.80	Reed Switch Stopper5 up
36	In No.81	Reed Switch Stopper5 down
37	In No.82	Reed Switch Stopper6 up
38	In No.83	Reed Switch Stopper6 down
39	In No.84	Reed Switch Stopper7 up
40	In No.85	Reed Switch Stopper8 up
41	In No.86	Photo sensor Label
42	In No.87	Sensor Vacuum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3-4 ตารางแสดงเอาท์พุทของบอร์ด 1

I/O OUTPUT ROBOT		
I/O Board 1		
PIN	Signal Name	Name
10	Out No.0	Solenoid Main
11	Out No.1	Solenoid Lifter
12	Out No.2	Solenoid Stopper 1
13	Out No.3	Solenoid Stopper 2
14	Out No.4	Solenoid Stopper 3
15	Out No.5	Solenoid Stopper 4
17	OutCom 0	Output common No. 0 to 7
27	Out No.6	Solenoid Stopper 5
28	Out No.7	Solenoid Stopper 6
29	Out No.8	Solenoid Stopper 7
30	Out No.9	Solenoid Stopper 8
31	Out No.10	Solenoid Stopper Lifter
33	OutCom 0	Output common No. 8 to 15
43	Out No.11	Solenoid Vacuum
44	Out No.12	Solenoid Blow
45	Out No.13	Lower case In Request UPLINK
46	Out No.14	Hard disk In Request UPLINK
47	Out No.15	Hard disk Out Ready DOWNLINK

ตารางที่ 3-5 ตารางแสดงเอาท์พุทของบอร์ด 2

I/O OUTPUT ROBOT		
I/O Board 2		
PIN	Signal Name	Name
10	Out No.64	Solenoid Stopper Lower case 1
11	Out No.65	Solenoid Stopper Lower case 2
12	Out No.66	Solenoid Stopper Lower case 3
13	Out No.67	Solenoid Stopper Lower case 4
14	Out No.68	Solenoid Stopper Lower case 5
15	Out No.69	Solenoid Stopper Lower case 6
17	OutCom 0	Output common No. 0 to 7
27	Out No.70	Solenoid Stopper Lower case 7
28	Out No.71	Solenoid Push
29	Out No.72	Spare
30	Out No.73	Spare
31	Out No.74	Solenoid Table Feed
33	OutCom 0	Output common No. 8 to 15
43	Out No.75	Conveyor
44	Out No.76	Tower Light Red
45	Out No.77	Tower Light Yellow
46	Out No.78	Tower Light Green
47	Out No.79	Tower Light Buzzer

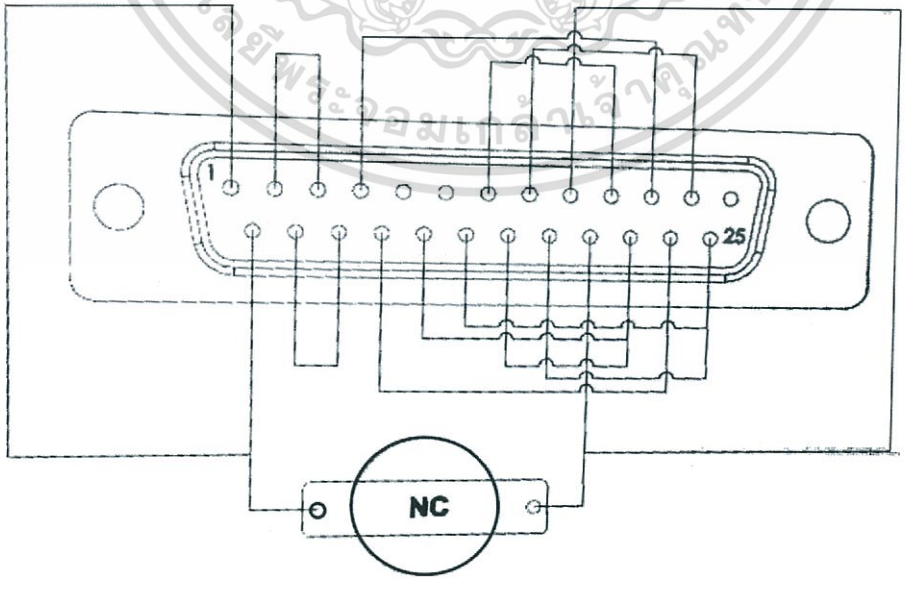
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การติดตั้งอุปกรณ์

การดำเนินการหลังจากการวางแผน คือ การดำเนินการด้านฮาร์ดแวร์ ทั้งการติดตั้งอุปกรณ์ชนิดต่างๆบนตำแหน่งที่เหมาะสม การต่อสาย (Wiring) และการทำสายเชื่อมต่อ (Connector) การเตรียมความพร้อมของอุปกรณ์ต่างๆ รวมทั้งการ Calibrate แขนกล เพื่อให้เกิดความสะดวกในการทดสอบโปรแกรมในขั้นต่อไป ซึ่งการทำสายเชื่อมต่อนั้น สายแต่ละชนิดจะมีการเชื่อมต่อกับหัวเชื่อมต่อที่แตกต่างกันออกไป เช่น สาย RS-232, ปุ่ม Emergency และ Hand checking

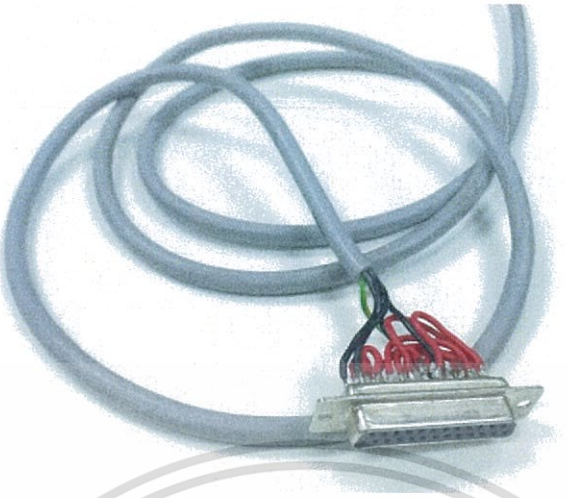
3.3.1 ปุ่มฉุกเฉิน (Emergency button)

รูปที่ 3.6 รูปแสดงอุปกรณ์สำหรับทำปุ่มฉุกเฉินของคอนโทรลเลอร์

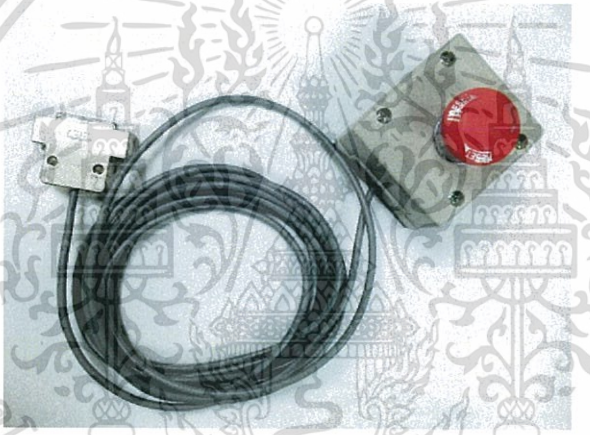


รูปที่ 3.7 Emergency Connection DB-25 Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

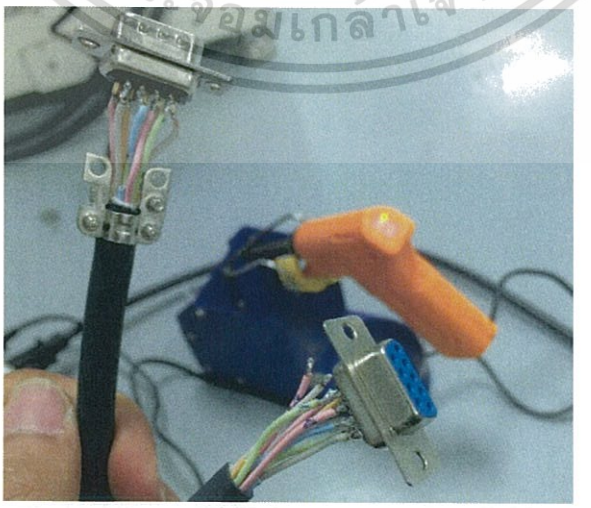


รูปที่ 3.8 รูปแสดง Emergency connector หลังทำการบัดกรี



รูปที่ 3.9 รูปแสดงปุ่ม Emergency สำหรับคอนโทรลเลอร์

3.3.2 สายมาตรฐานแนะนำรุ่น -232 (Recommended Standard: RS-232)

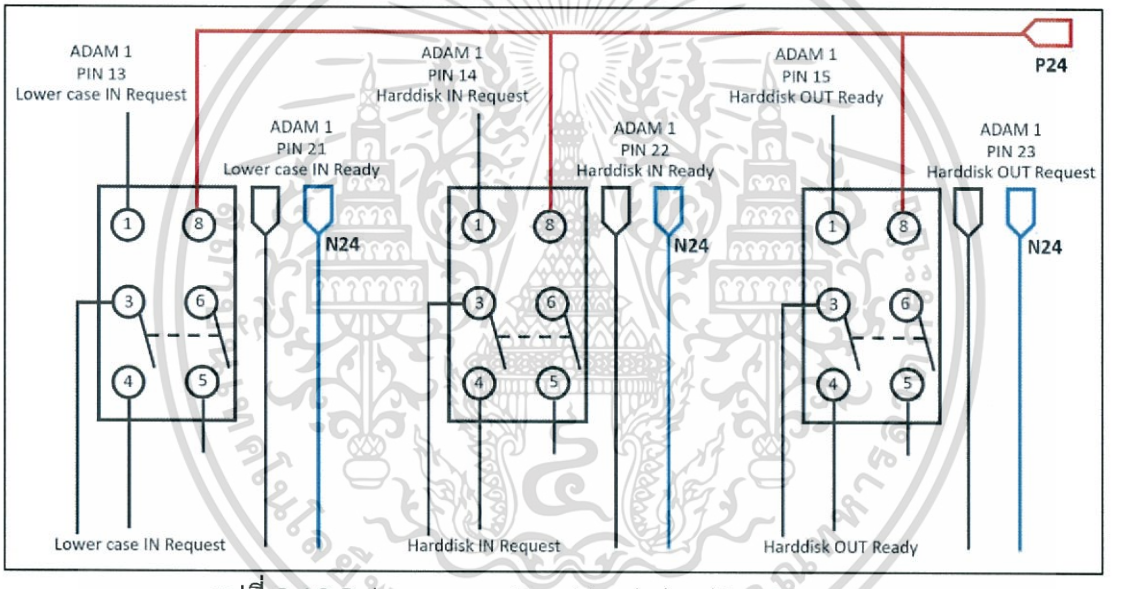


รูปที่ 3.10 รูปแสดงการบัดกรีสายเชื่อมต่อ RS-232

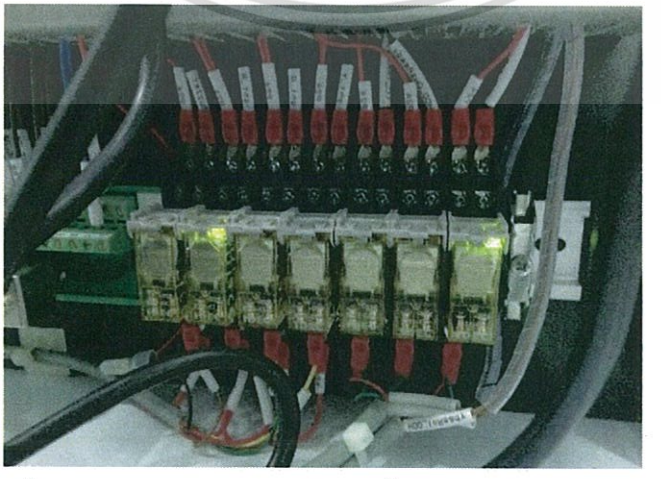
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 รูปแสดงลักษณะสายเชื่อมต่อ RS-232

3.3.3 Hand checking



รูปที่ 3.12 Relay connection Hand checking Diagram

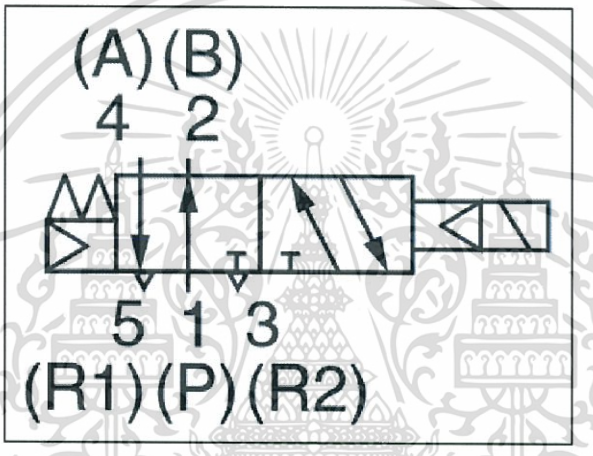


รูปที่ 3.13 รูปแสดงตำแหน่งการติดตั้งรีเลย์ภายในแผงควบคุม

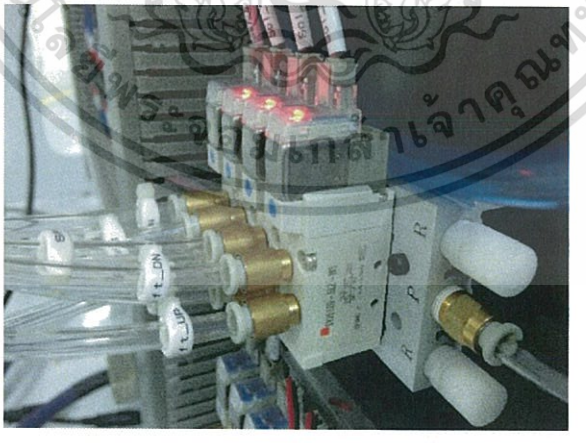
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.14 Hand checking connector

3.3.4 Solenoid valve



รูปที่ 3.15 รูปแสดงลักษณะการแบ่งลมของโซลินอยด์วาล์ว

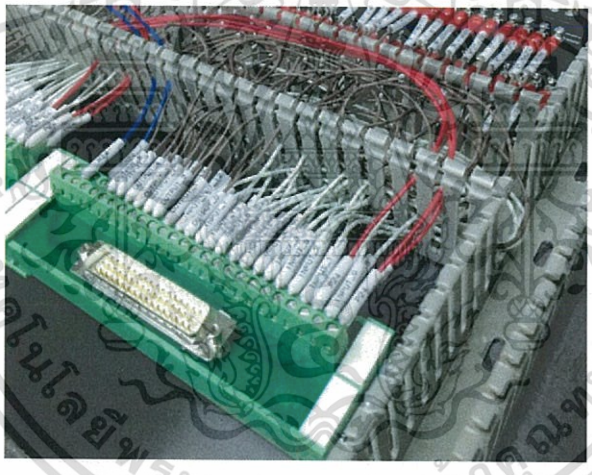


รูปที่ 3.16 รูปแสดงลักษณะการต่อสายลมกับโซลินอยด์วาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.17 รูปแสดงตำแหน่งการติดตั้งโซลินอยด์วาล์วบนแผงควบคุม

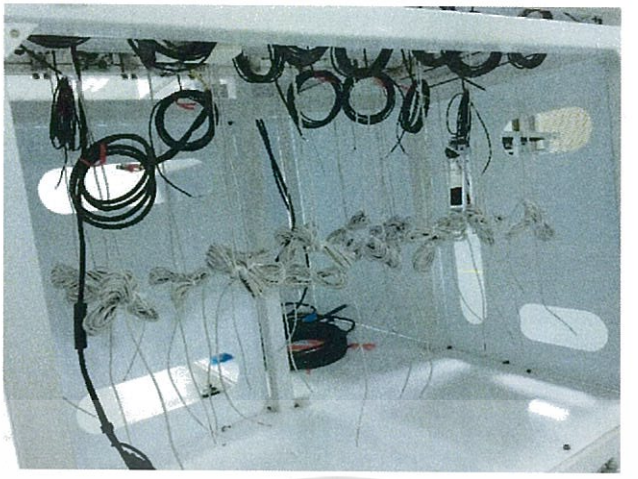
3.3.5 Wiring I/O



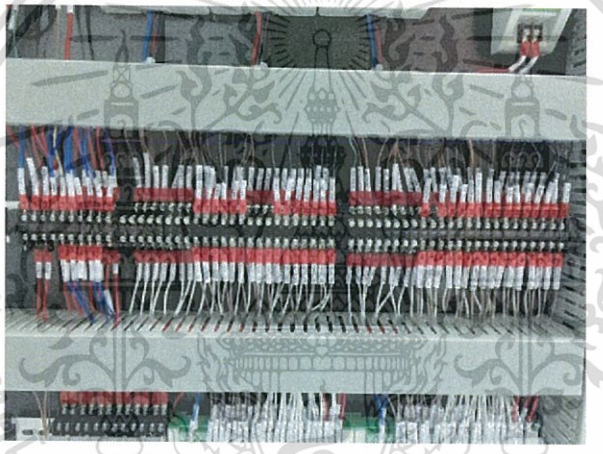
รูปที่ 3.18 รูปแสดงการต่อสาย I/O จาก Terminal I/O ของคอนโทรลเลอร์

รูปที่ 3.19 รูปแสดงตำแหน่งการติดตั้งเซนเซอร์ ลีดสวิตช์ และสตอปเปอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 รูปแสดงสายอินพุทจากเซนเซอร์และลีดสวิตช์



รูปที่ 3.21 รูปแสดงการต่อสายอินพุทจากเซนเซอร์ และเอาท์พุทต่างๆ เข้าสู่ Terminal I/O

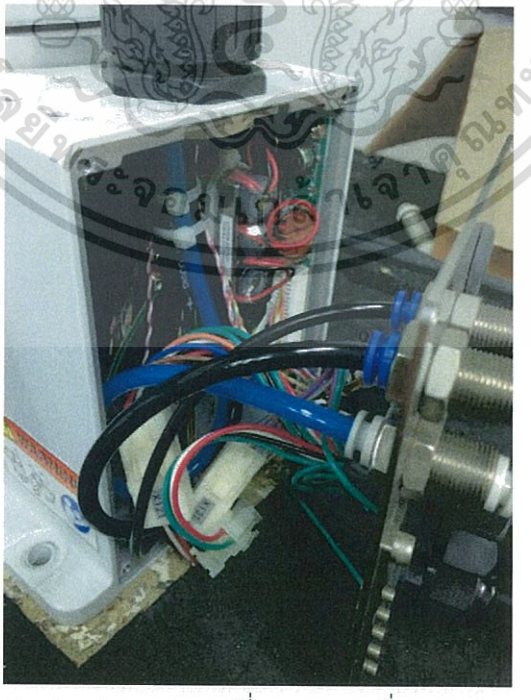
รูปที่ 3.22 รูปแสดงการต่อ I/O ทั้งหมดบนแผงควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.23 รูปแสดงการทดสอบความถูกต้องของการทำงานของ I/O ต่างๆ

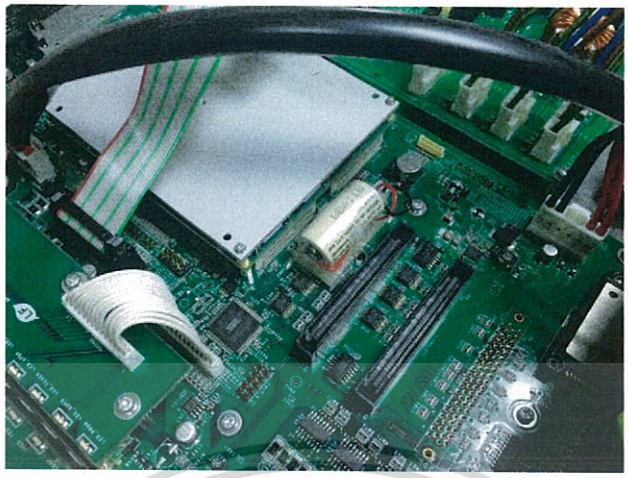
3.3.6 Robot setup

การเริ่มต้นใช้งานแขนกลนั้น จำเป็นต้องตรวจมีการตรวจสอบความพร้อมของอุปกรณ์ที่มี เช่น แบตเตอรี่ของแขนกล แบตเตอรี่ของคอนโทรลเลอร์ และการปรับตั้งค่า (Calibration) ของแขนกล เพื่อให้มีประสิทธิภาพ เพื่อเตรียมความพร้อมสำหรับการใช้งาน



รูปที่ 3.24 รูปแสดงการเปลี่ยนแบตเตอรี่ของแขนกล

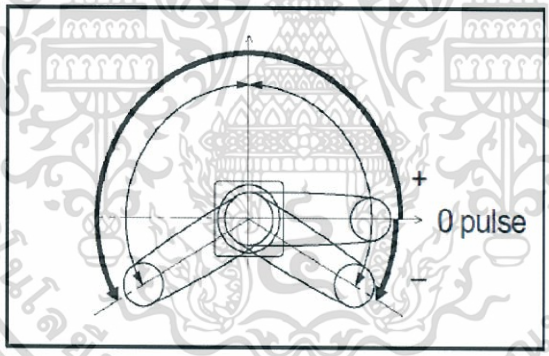
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



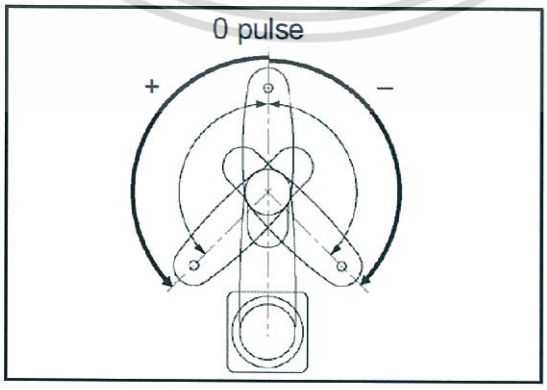
รูปที่ 3.25 รูปแสดงการเปลี่ยนแบตเตอรี่ของบอร์ดของคอนโทรลเลอร์

การ Calibration Robot มีขั้นตอนดังนี้

1. ปรับตำแหน่งของแขนกลให้ตรงกับตำแหน่งในการ calibrate แต่ละแกน

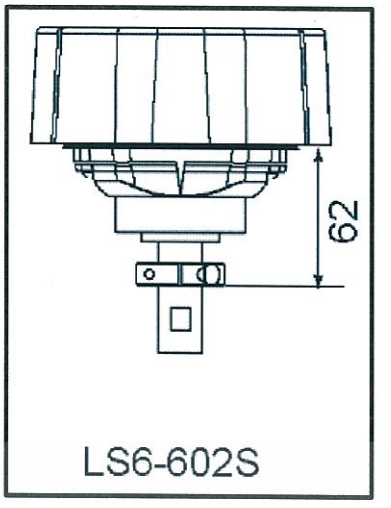


รูปที่ 3.26 รูปแสดงตำแหน่งการปรับตั้งค่า Joint#1 ของแขนกล

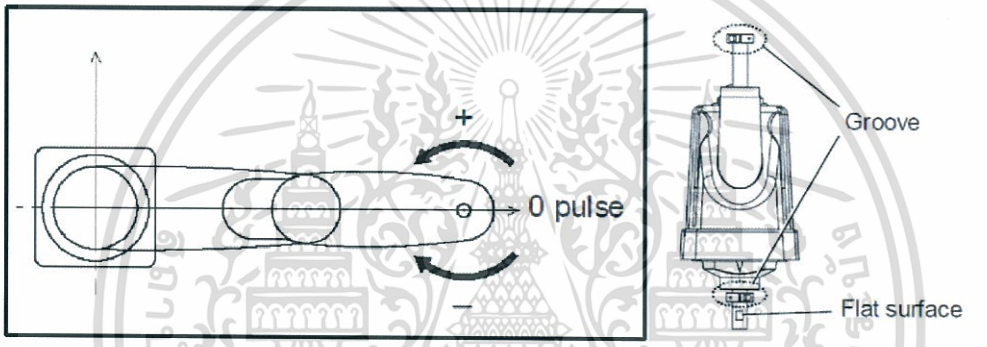


รูปที่ 3.27 รูปแสดงตำแหน่งการปรับตั้งค่า Joint#2 ของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 รูปแสดงตำแหน่งการปรับตั้งค่า Joint#3 ของแขนกล



รูปที่ 3.29 รูปแสดงตำแหน่งปรับตั้งค่า Joint#4 ของแขนกล

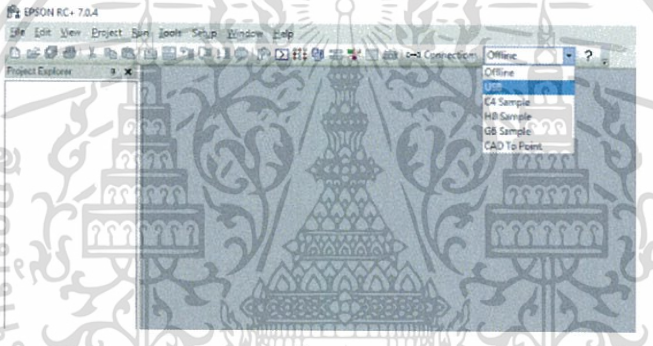
2. เปิดโปรแกรม EPSON RC+ 7.0

รูปที่ 3.30 รูปแสดงการเปิดโปรแกรม EPSON RC+ 7.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

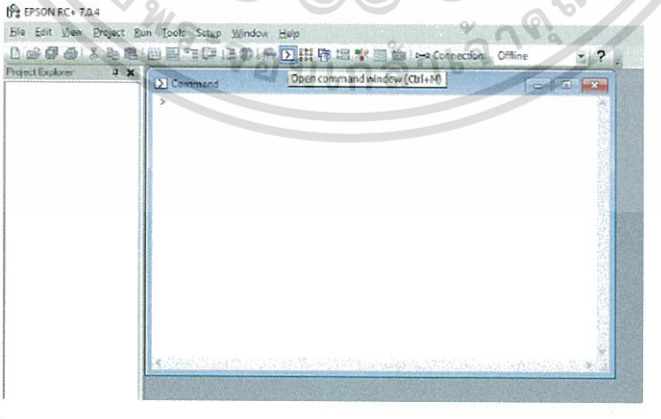
3. เชื่อมต่อคอนโทรลเลอร์กับคอมพิวเตอร์

รูปที่ 3.31 รูปแสดงการเชื่อมต่อคอนโทรลเลอร์กับคอมพิวเตอร์



รูปที่ 3.32 รูปแสดงการเลือกการเชื่อมต่อแบบ USB ใน EPSON RC+ 7.0

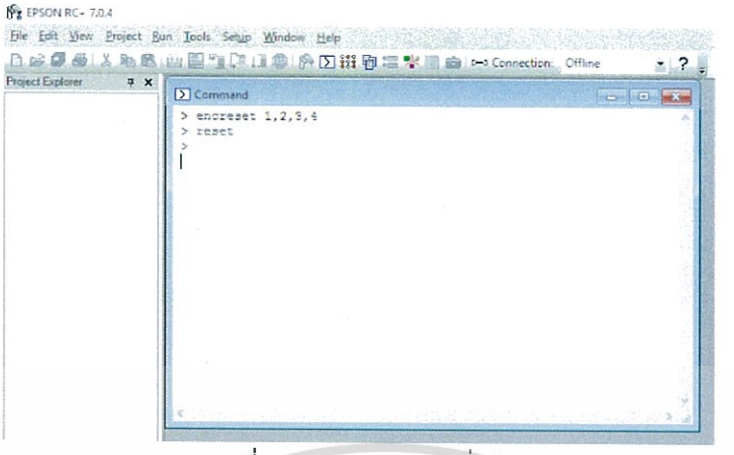
4. เปิดหน้าต่าง command window



รูปที่ 3.33 รูปแสดงการเปิดหน้าต่าง command window ใน EPSON RC+ 7.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

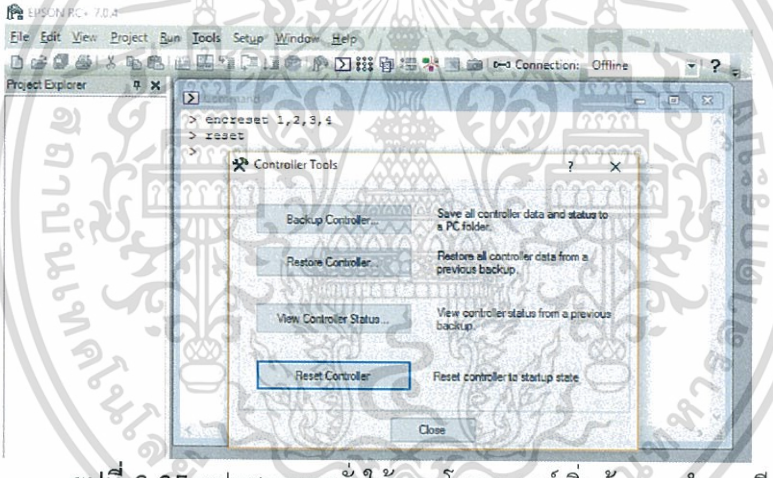
5. พิมพ์คำสั่งในการรีเซต encoder



รูปที่ 3.34 รูปแสดงคำสั่งรีเซต encoder

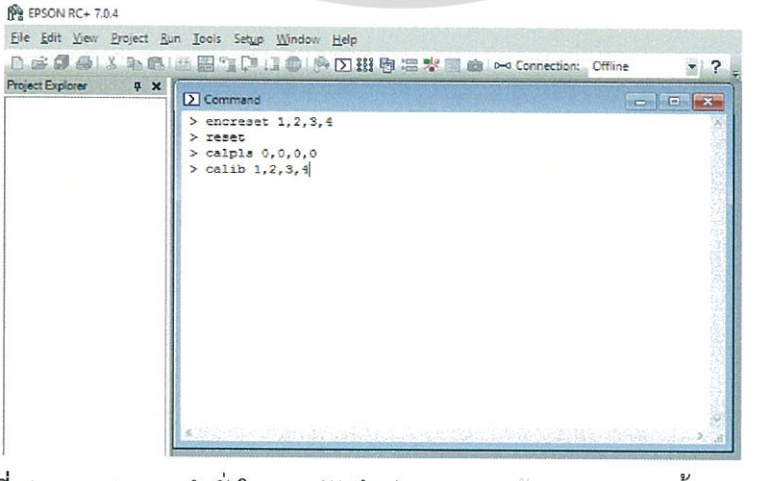
6. รีสตาร์ทคอนโทรลเลอร์

เลือก Tools > Controller > Reset Controller



รูปที่ 3.35 รูปแสดงการสั่งให้คอนโทรลเลอร์เริ่มต้นการทำงานอีกครั้ง

7. พิมพ์คำสั่งในการบันทึกค่าการ calibrate



รูปที่ 3.36 รูปแสดงคำสั่งในการบันทึกค่าการ calibrate แขนกลทั้ง 4 แกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การเขียนโปรแกรมควบคุม

สำหรับการควบคุมการทำงานในขั้นตอนต่างๆนั้น จะทำการเขียนโปรแกรมควบคุมผ่านคอนโทรลเลอร์ของแขนกล โดยใช้โปรแกรม EPSON RC+ 7.0 และการแสดงผลและสั่งการ จะถูกควบคุมผ่าน GUI (Graphic user interface) โดยใช้โปรแกรม Visual Studio 2015

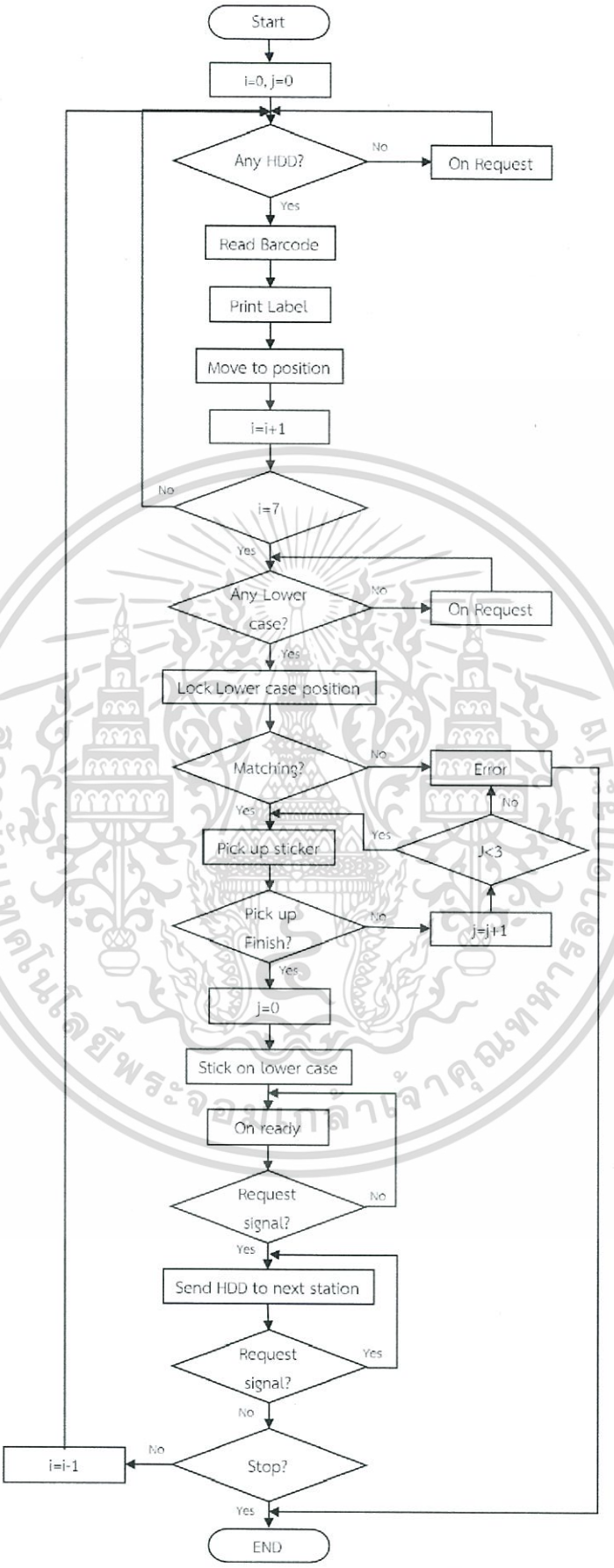
รูปที่ 3.37 รูปแสดงสัญลักษณ์โปรแกรม EPSON RC+ 7.0

รูปที่ 3.38 รูปแสดงสัญลักษณ์โปรแกรม Visual Studio 2015

3.4.1 Sequence

ขั้นตอนการทำงานของเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ คือ เมื่อเริ่มต้นการทำงาน (ต้องไม่มีฮาร์ดดิสก์และฝาครอบในเครื่อง) เครื่องจะทำการส่งสัญญาณไปยังสถานีก่อนหน้า ทั้งสถานีสำหรับปล่อยฮาร์ดดิสก์ และสถานีสำหรับปล่อยฝาครอบ เพื่อให้ส่งฮาร์ดดิสก์และฝาครอบเข้ามาภายในเครื่อง และเมื่อได้รับฮาร์ดดิสก์หรือฝาครอบแล้วก็จะทำการหยุดส่งสัญญาณไป หลังจากได้รับฮาร์ดดิสก์แล้วก็จะเริ่มสแกนบาร์โค้ดบนฮาร์ดดิสก์และส่งค่าซีเรียลนัมเบอร์ไปยังเครื่องพิมพ์ให้พิมพ์ฉลากออกมา เมื่อพิมพ์เสร็จแล้ว ก็จะลำเลียงฮาร์ดดิสก์ไปยังตำแหน่งสุดท้ายเพื่อรอการปล่อยสู่สถานีถัดไป และรอรับฮาร์ดดิสก์อันใหม่มาทำการสแกน จนครบ 7 ตัว แล้วจึงทำการตรวจสอบซีเรียลนัมเบอร์บนฉลากที่พิมพ์ว่าถูกต้องหรือไม่ โดยเทียบกับค่าซีเรียลนัมเบอร์ของฮาร์ดดิสก์ในโปรแกรม เมื่อค่าซีเรียลนัมเบอร์บนฉลากกับในโปรแกรมถูกต้องแล้ว จึงให้ Table Feed มายกฉลากไว้ แล้วให้แขนกลทำการหยิบสติ๊กเกอร์มารอสำหรับการติดบนฝาครอบต่อไป โดยในขณะเดียวกัน ก็จะทำการลำเลียงฝาครอบไปยังตำแหน่งสำหรับรอการติดสติ๊กเกอร์ด้วย เมื่อฝาครอบเคลื่อนที่มายังตำแหน่งสำหรับการติดแล้ว Lifter จะยกฝาครอบขึ้น และตัวผลักจะดันให้ฝาครอบไม่สามารถขยับได้ หลังจากนั้น จะให้ให้แขนกลนำสติ๊กเกอร์มาติดบนฝาครอบ เมื่อดำเนินการสำเร็จแล้ว ก็จะปล่อยตัวผลักออก และนำ Lifter ลง แล้วส่งสัญญาณไปยังสถานีถัดไป แล้วรอสัญญาณตอบกลับจากสถานีถัดไป จึงทำการส่งทั้งฮาร์ดดิสก์และฝาครอบที่ติดสติ๊กเกอร์แล้ว ไปสู่สถานีต่อไป แล้วรอให้สถานีถัดไปหยุดส่งสัญญาณกลับมา จึงดำเนินการฮาร์ดดิสก์ตัวถัดไปได้ โดยสามารถอธิบายได้จาก แผนภาพขั้นตอนการดำเนินงาน ดังรูปที่ 3.39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



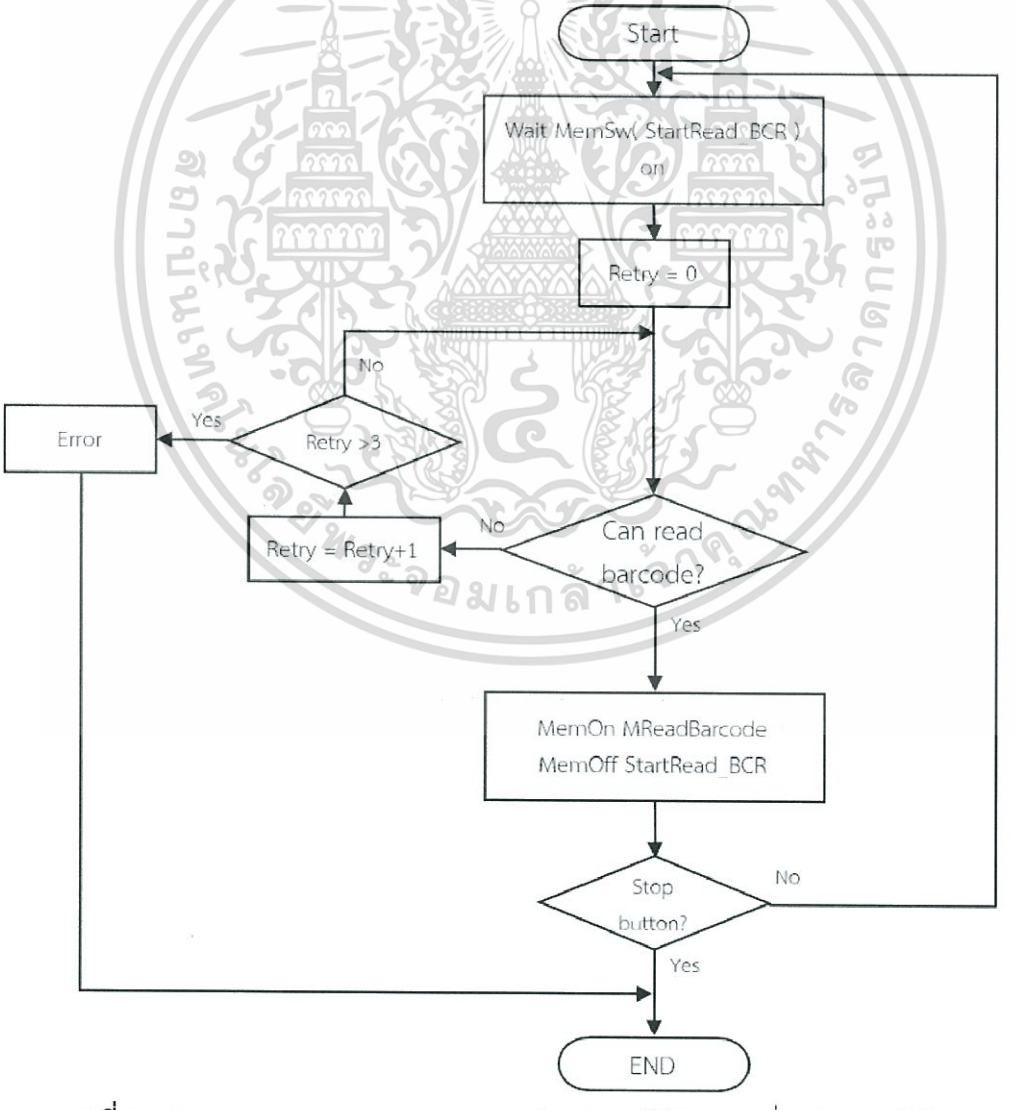
รูปที่ 3.39 แผนภาพแสดงการทำงานโดยรวมของเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของรายละเอียดการทำงานภายในเครื่องนั้น จะถูกแบ่งออกเป็น 7 ฟังก์ชั่นย่อยที่ทำงานพร้อมกัน ดังนี้

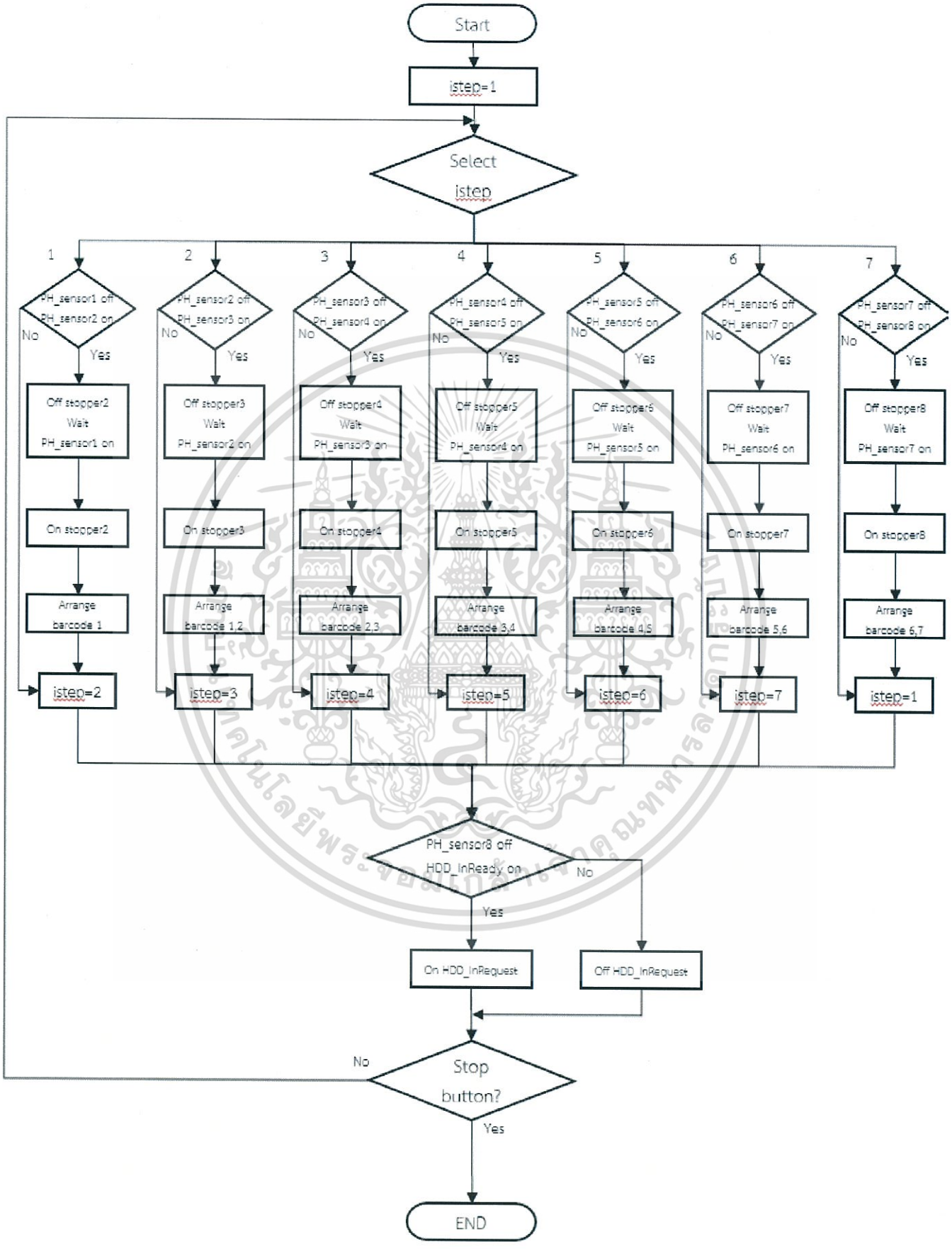
1. ฟังก์ชั่นการควบคุมการอ่านค่าบาร์โค้ดของเครื่องอ่านบาร์โค้ด
2. ฟังก์ชั่นการลำเลียงฮาร์ดดิสก์บนคอนเวเยอร์
3. ฟังก์ชั่นการลำเลียงฝาครอบบนคอนเวเยอร์
4. ฟังก์ชั่นการส่งข้อมูลไปยังเครื่องพิมพ์ฉลาก
5. ฟังก์ชั่นการยกลิฟเตอร์สำหรับยึดตำแหน่งของฝาครอบ
6. ฟังก์ชั่นการยกเทเบิ้ลฟีดสำหรับการหยิบสติ๊กเกอร์ของแขนกล
7. ฟังก์ชั่นการควบคุมการเคลื่อนที่ของแขนกล

โดยการทำงานของแต่ละฟังก์ชั่นนั้น จะถูกเชื่อมต่อ และรับส่งสัญญาณกันผ่านการควบคุมเมมโมรี่ในตัวคอนโทรลเลอร์ โดยจะแสดงขั้นตอนการทำงานของแต่ละฟังก์ชั่น ใน Flowchart ดังนี้



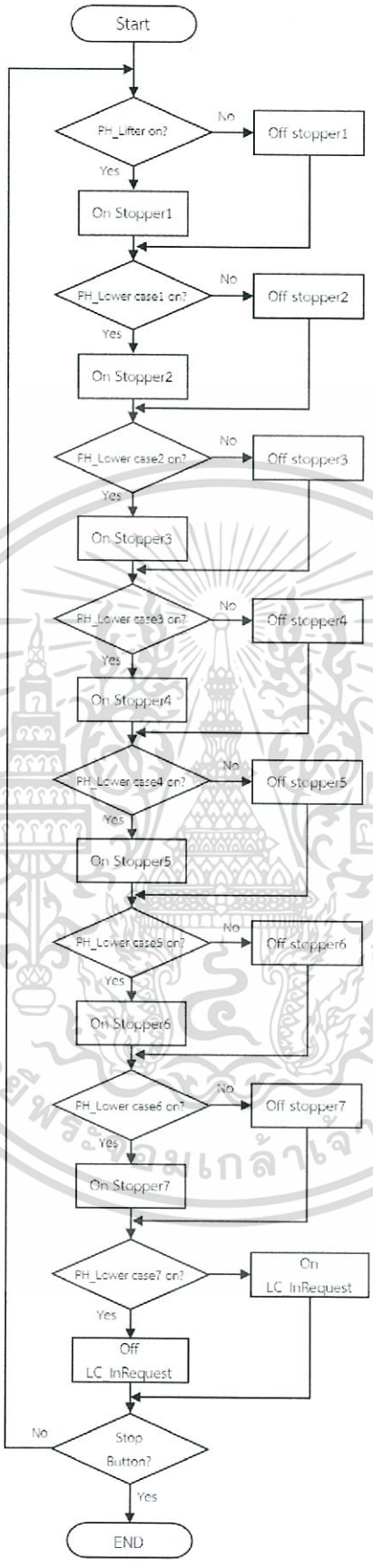
รูปที่ 3.40 แผนภาพแสดงการควบคุมการอ่านค่าบาร์โค้ดของเครื่องอ่านบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



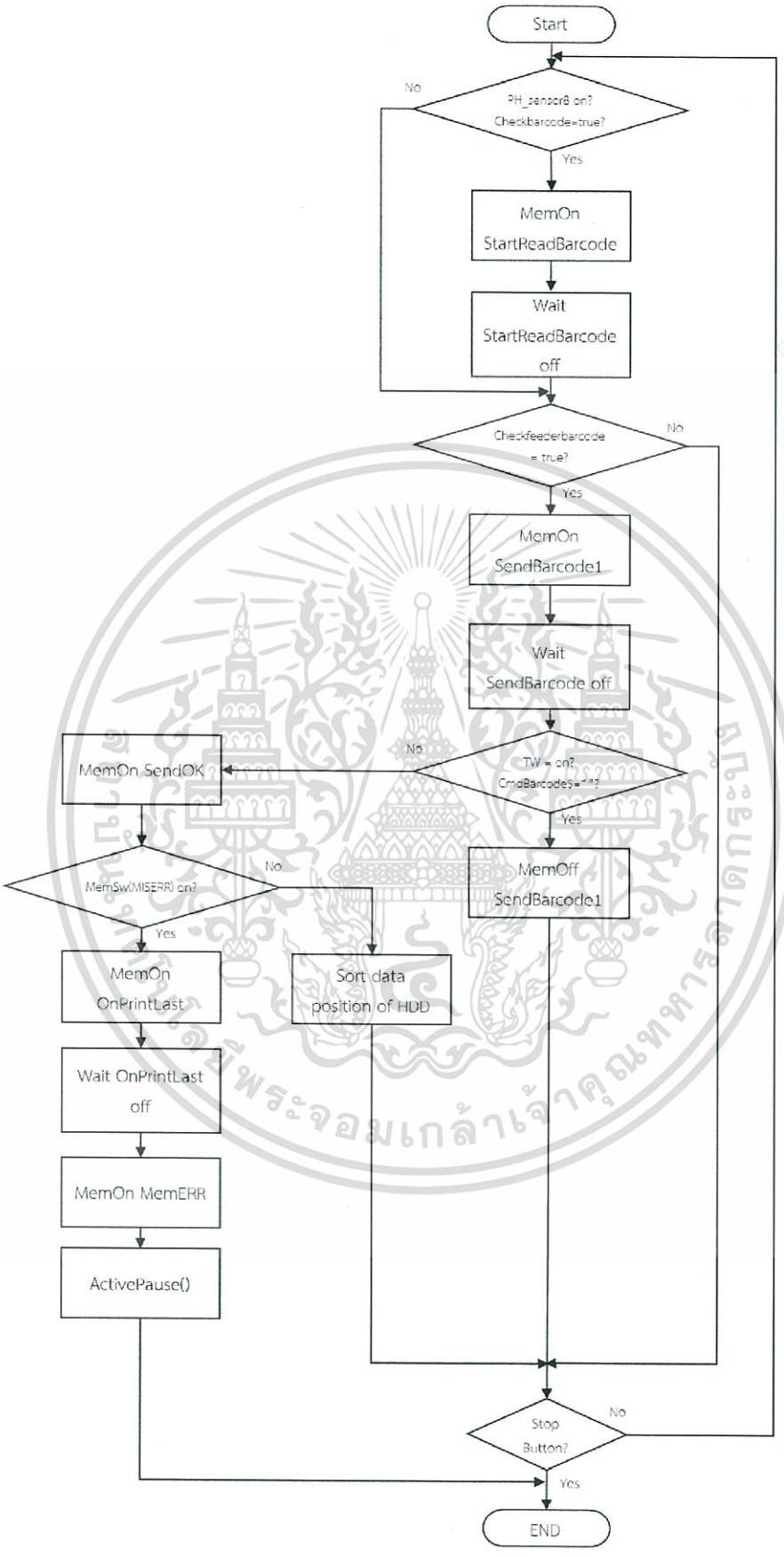
รูปที่ 3.41 แผนภาพแสดงการลำเลียงฮาร์ดดิสก์บนคอนเวเยอร์

เอกสารนี้เป็นเอกสาร[illegible]ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



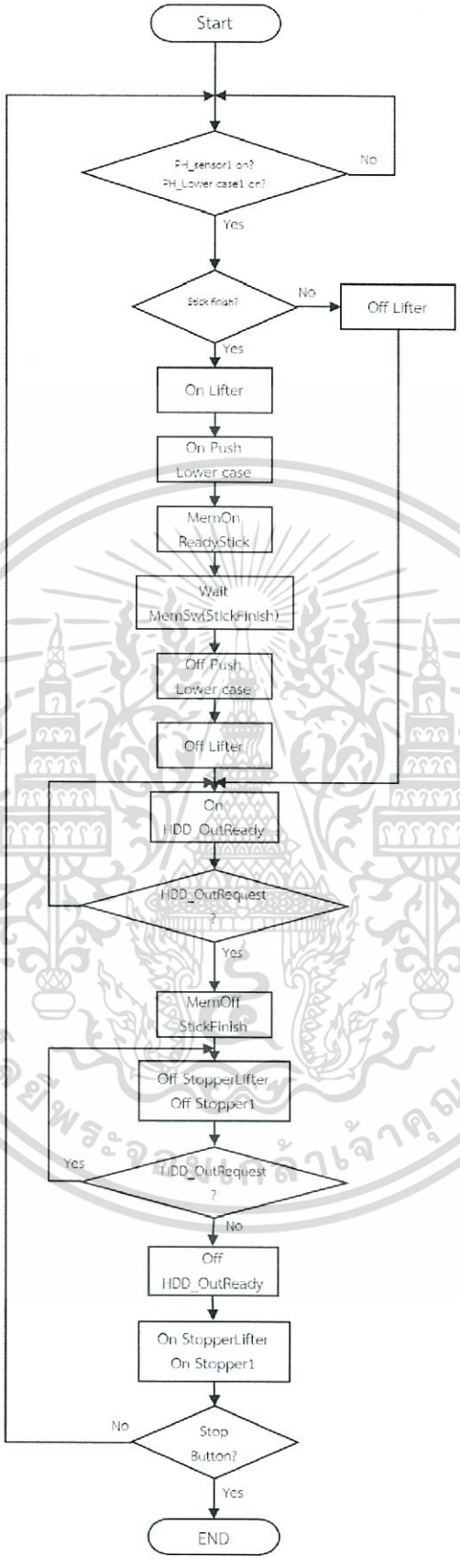
รูปที่3.42 แผนภาพแสดงการลำเลียงฝาครอบบนคอนเวเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



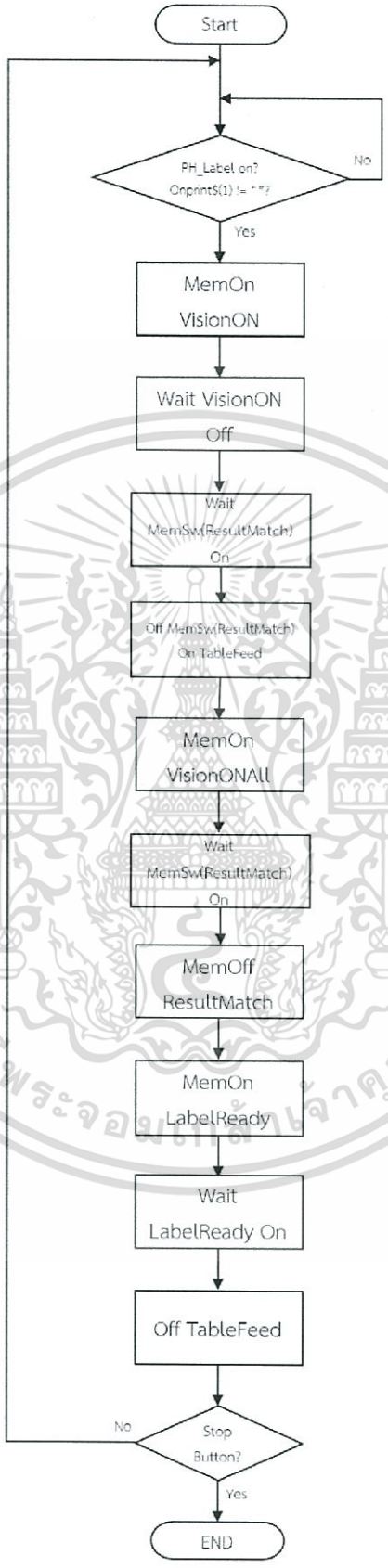
รูปที่ 3.43 แผนภาพแสดงการส่งข้อมูลไปยังเครื่องพิมพ์ฉลาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



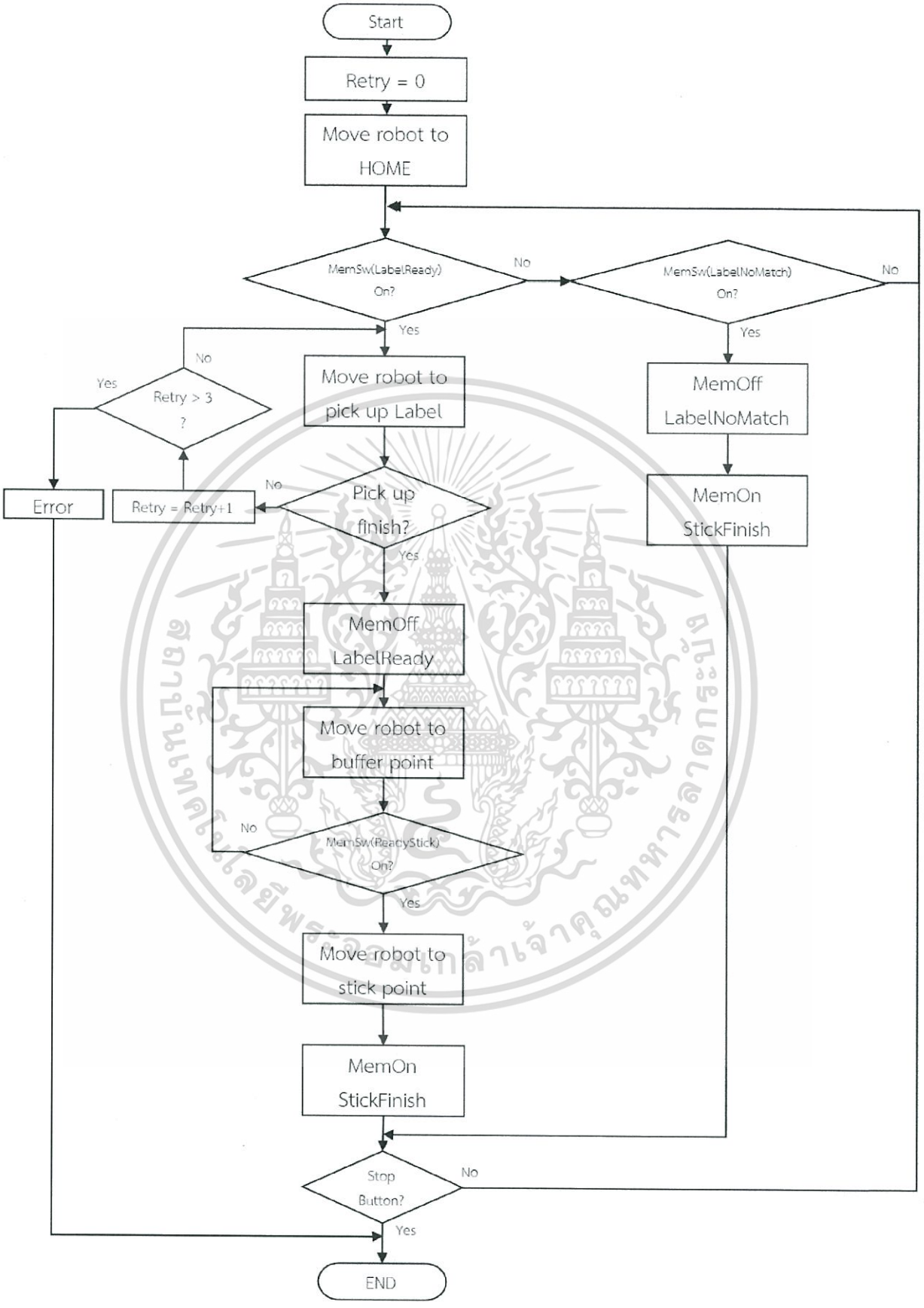
รูปที่ 3.44 แผนภาพแสดงการยกลิฟเตอร์สำหรับยึดตำแหน่งของฝาครอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.45 แผนภาพแสดงการยกเทเบิ้ลฟีดสำหรับการหยิบสติ๊กเกอร์ของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

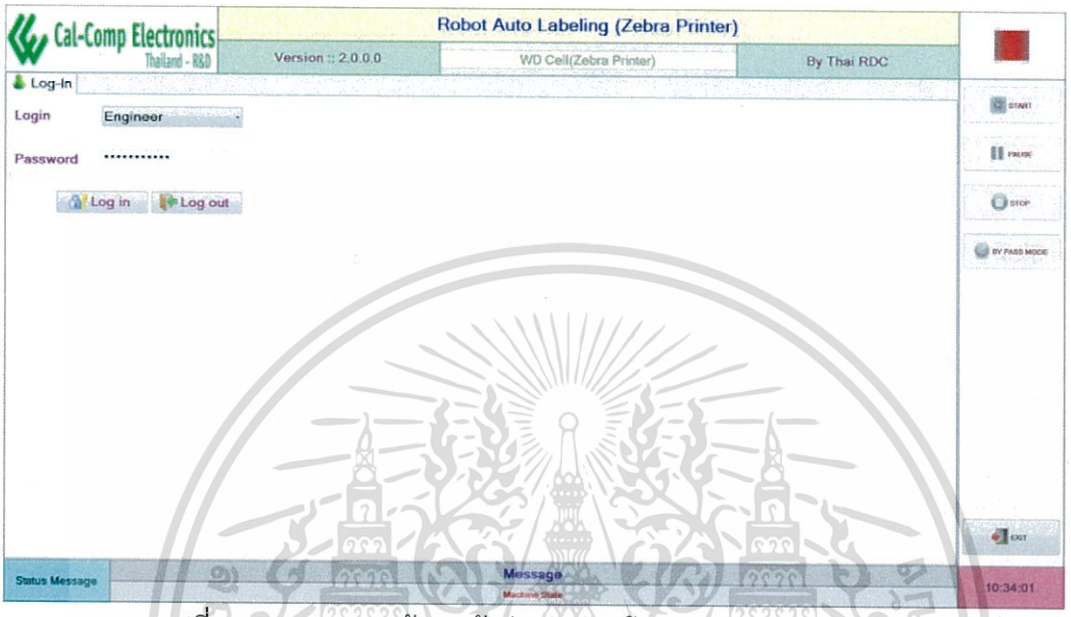


รูปที่ 3.46 แผนภาพแสดงการควบคุมการเคลื่อนที่ของแขนกล

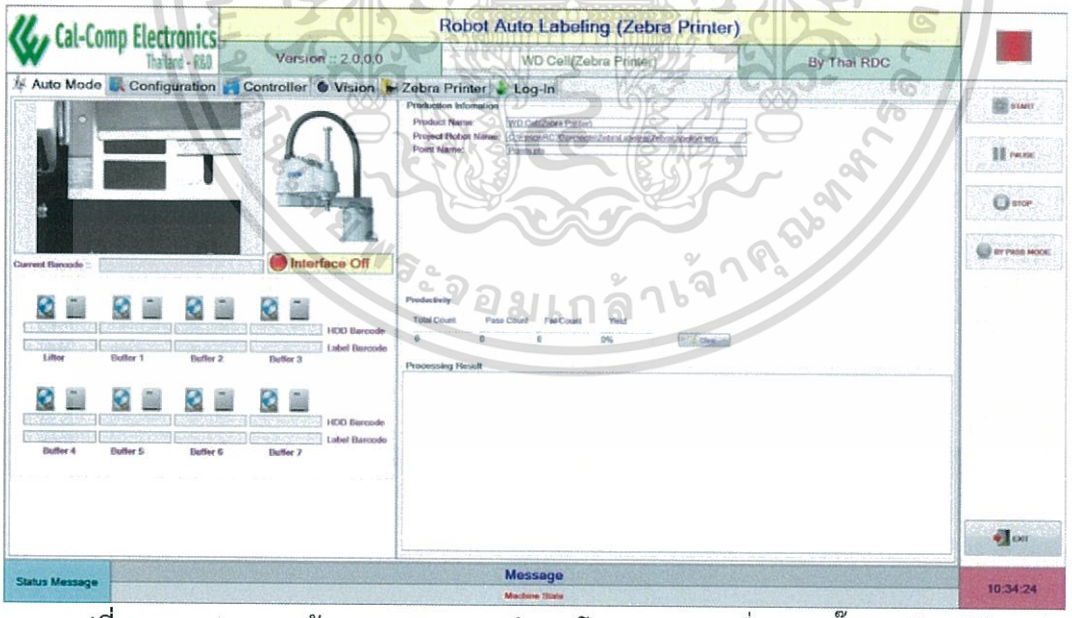
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 GUI (Graphic user interface)

การควบคุมการทำงานต่างๆ และการแสดงผลจากกระดำเนินงานต่างๆของเครื่อง จะแสดงผลอยู่บนหน้าจอ GUI ดังนี้

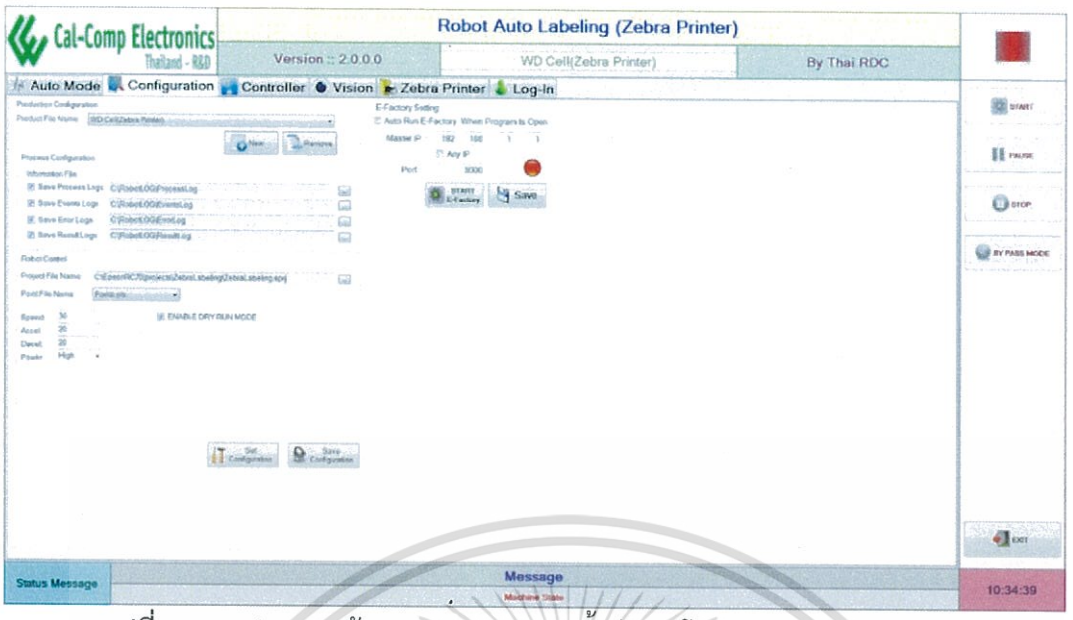


รูปที่ 3.47 รูปแสดงหน้าจอเข้าสู่ระบบของโปรแกรม Robot Auto Labeling

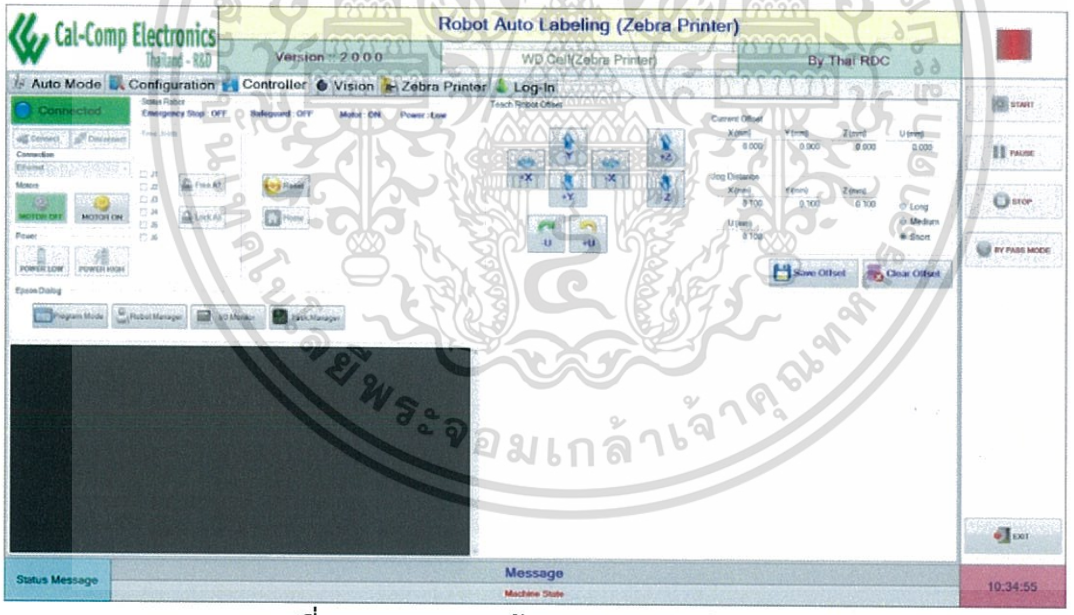


รูปที่ 3.48 รูปแสดงหน้าจอแสดงผลการทำงานโดยรวมของเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

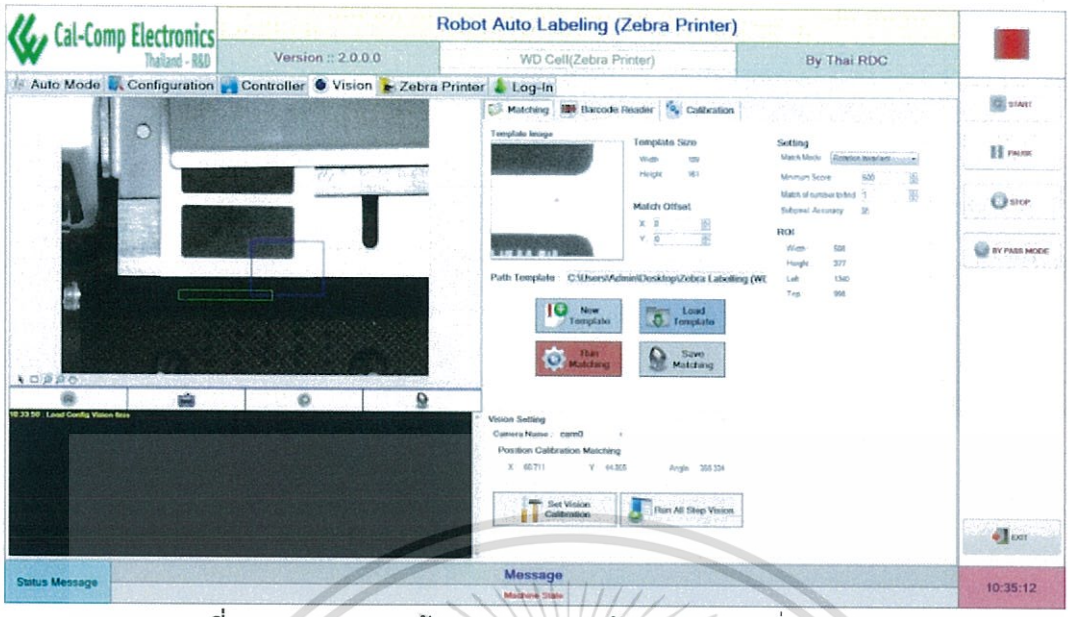


รูปที่ 3.49 รูปแสดงหน้าจอการควบคุมการตั้งค่าของโปรแกรม Robot Auto Labeling

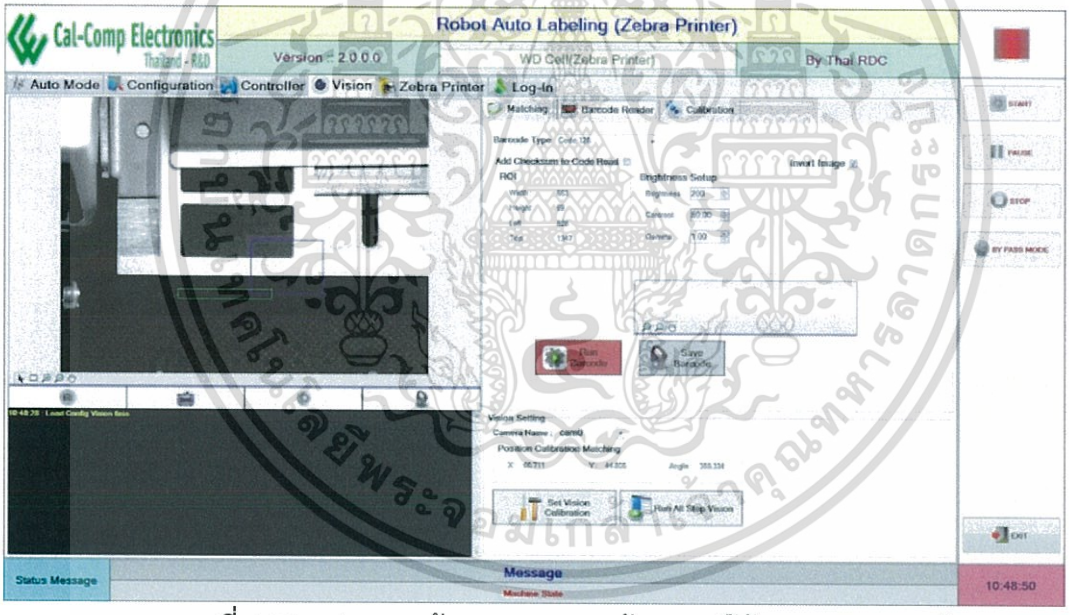


รูปที่ 3.50 รูปแสดงหน้าจอควบคุมการทำงานของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

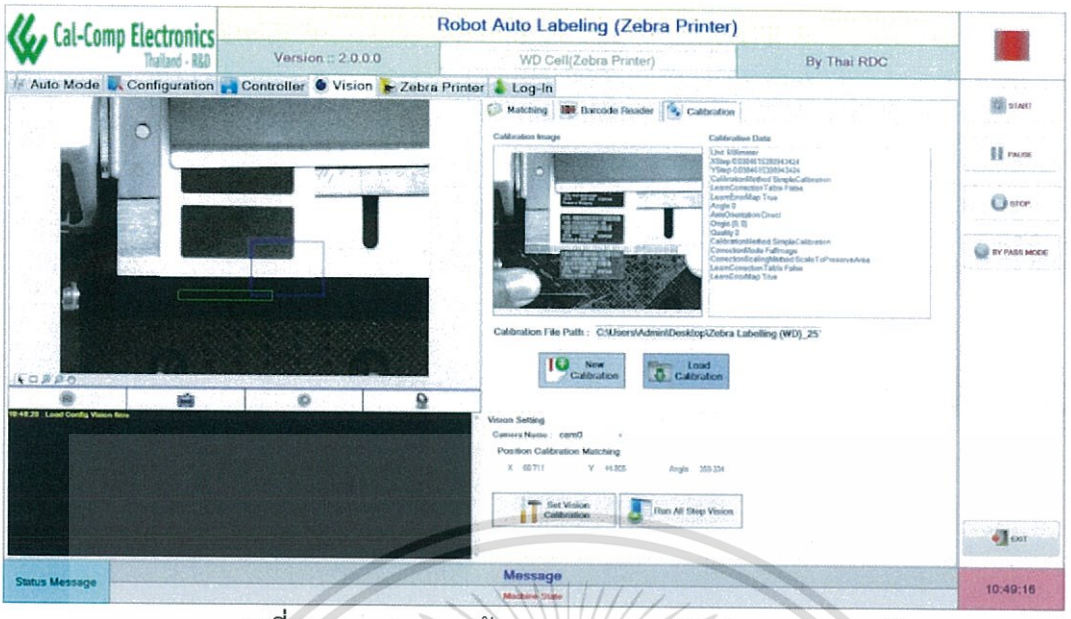


รูปที่ 3.51 รูปแสดงหน้าจอควบคุมการค้นหาตำแหน่งที่ผิดพลาดของฉลาก

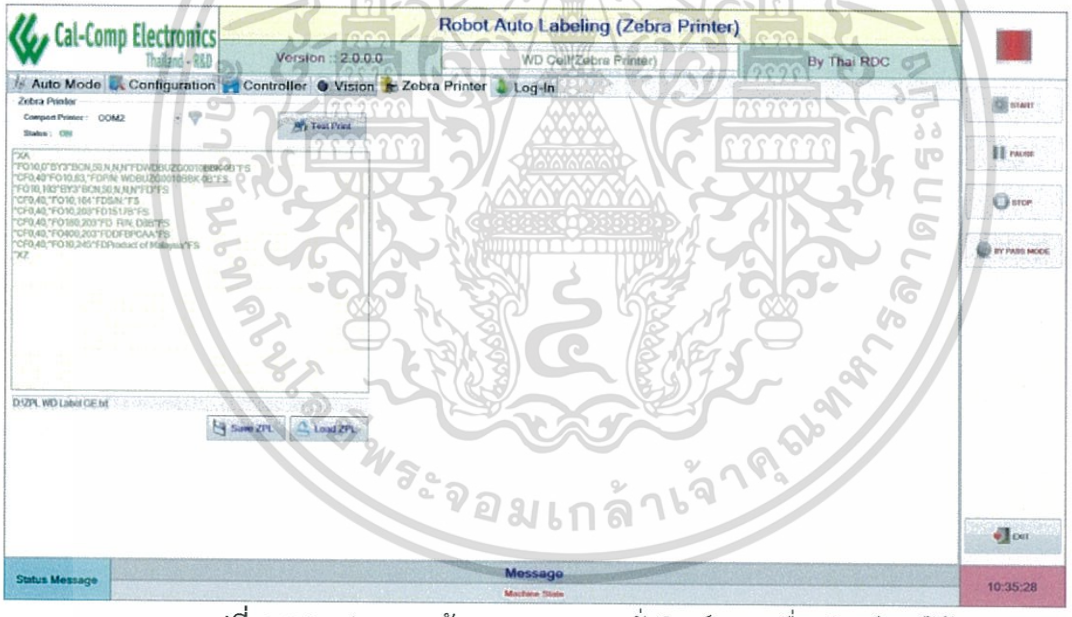


รูปที่ 3.52 รูปแสดงหน้าจอควบคุมการค้นหาบาร์โค้ดบนฉลากของกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.53 รูปแสดงหน้าจอควบคุมการปรับค่าการวัดของกล้อง



รูปที่ 3.54 รูปแสดงหน้าจอควบคุมการสั่งพิมพ์ของเครื่องพิมพ์บาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงาน

การดำเนินการตามขั้นตอนต่างๆเกี่ยวกับเครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัตินั้น เริ่มดำเนินการในส่วนของฮาร์ดแวร์แล้วจึงดำเนินการต่อในส่วนของซอฟต์แวร์ โดยในขั้นตอนแรกสำหรับการเริ่มดำเนินการ ด้านการติดตั้งอุปกรณ์นั้น ได้ทำการติดตั้งอุปกรณ์ และส่วนประกอบต่างๆภายในเครื่อง เช่น เซนเซอร์, สตอปเปอร์ และการต่อวงจรภายในแผงควบคุมการทำงาน ได้ผลลัพธ์ดังรูปที่ 4.1

รูปที่ 4.1 รูปแสดงการติดตั้งอุปกรณ์ต่างๆในเครื่องติดสติ๊กเกอร์

โดยหลังจากดำเนินการด้านฮาร์ดแวร์เสร็จแล้ว จึงทำการเขียนโปรแกรม และทดสอบการทำงานของอินพุตและเอาท์พุตขั้นต้น โดยทดสอบการทำงานตามขั้นตอนต่างๆ ซึ่งจะพบปัญหาจากการติดตั้งเซนเซอร์ที่มีตำแหน่งที่ไม่เหมาะสม ทำให้ไม่สามารถตรวจจับวัตถุที่ต้องการได้ และเกิดการทำงานผิดขั้นตอนของโปรแกรม ดังรูปที่ 4.2

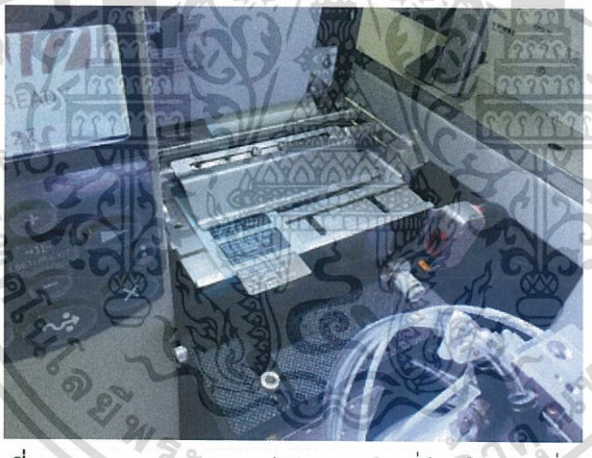
รูปที่ 4.2 รูปแสดงปัญหาตำแหน่งที่ไม่เหมาะสมในการติดตั้งเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

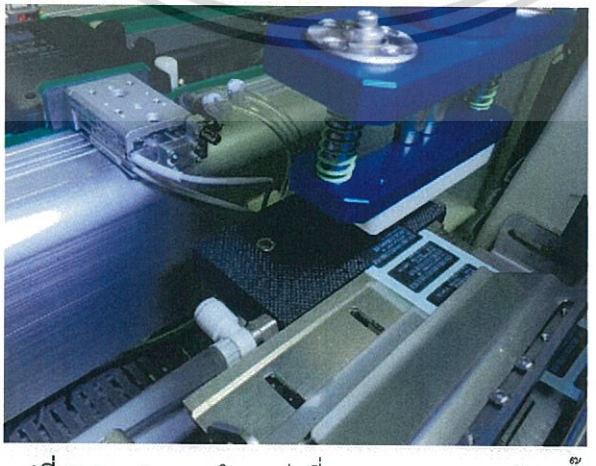
ดังนั้นจึงจำเป็นต้องแก้ไขตำแหน่งการติดตั้งเซนเซอร์ให้อยู่ในตำแหน่งที่เหมาะสม เมื่อตำแหน่งของเซนเซอร์และอุปกรณ์ต่างๆอยู่ในตำแหน่งที่เหมาะสมแล้ว ดังรูปที่ 4.3 จึงดำเนินการด้านโปรแกรมในขั้นต่อไป

รูปที่ 4.3 รูปแสดงตำแหน่งที่เหมาะสมในการติดตั้งเซนเซอร์

โดยจำเป็นต้องคำนึงถึงตำแหน่งของฉลากที่แขนกลสามารถหยิบไปติดบนฝาครอบได้ ซึ่งจำเป็นต้องใช้ Buffer ในการพิมพ์ฉลาก โดยวัดระยะจากตำแหน่งที่พิมพ์ถึงตำแหน่งที่แขนกลสามารถหยิบสติ๊กเกอร์ได้ มีระยะเทียบกับขนาดของฉลากได้ทั้งหมด 7ตัว ดังรูปที่ 4.4 และรูปที่ 4.5 ตามลำดับ



รูปที่ 4.4 รูปแสดงการลำเลียง buffer ที่ต้องใช้ในเครื่องพิมพ์ฉลาก



รูปที่ 4.5 รูปแสดงตำแหน่งที่แขนกลสามารถหยิบสติ๊กเกอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากแขนกลไม่สามารถหยิบสติ๊กเกอร์บนตำแหน่งที่พิมพ์เสร็จได้นั้น ทำให้การเขียนโปรแกรมมีความซับซ้อนมากยิ่งขึ้น โดยจำเป็นต้องจัดลำดับการพิมพ์ก่อน-หลัง ให้ตรงกับลำดับการเคลื่อนที่ของฮาร์ดดิสก์บนคอนเวเยอร์ ดังรูปที่ 4.6 เพื่อป้องกันความผิดพลาดในการติดฉลากบนฝาครอบผิดตัว

รูปที่ 4.6 รูปแสดงการลำเลียงฮาร์ดดิสก์บนคอนเวเยอร์

เมื่อฉลากเคลื่อนที่มาถึงตำแหน่งที่แขนกลสามารถหยิบได้แล้วนั้น จะพบความคลาดเคลื่อนของตำแหน่งที่ฉลากเคลื่อนที่ออกมา ดังนั้น จึงจำเป็นต้องใช้ระบบวิทัศน์ ในการช่วยค้นหาตำแหน่งความคลาดเคลื่อนของฉลาก ดังรูปที่ 4.7เพื่อให้แขนกลสามารถหยิบฉลากได้ในตำแหน่งที่แม่นยำมากขึ้น

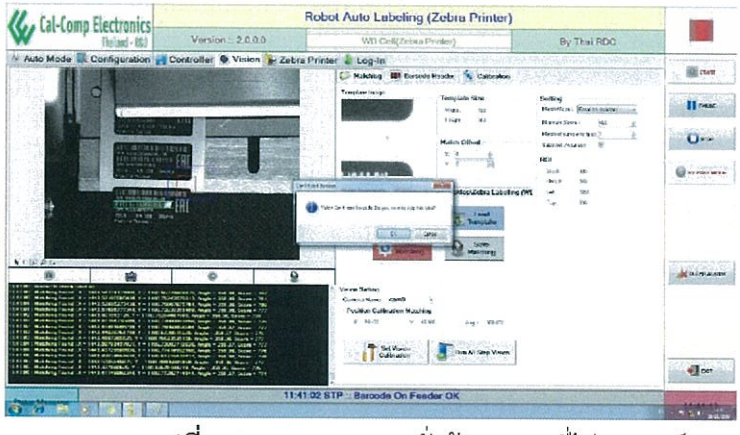
รูปที่ 4.7 รูปแสดงการใช้ระบบวิทัศน์ในการค้นหาตำแหน่งของฉลาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.8 รูปแสดงผลการติดสติ๊กเกอร์ด้วยแขนกล

รูปที่ 4.9 รูปแสดงผลการทดสอบการติดสติ๊กเกอร์ที่ตำแหน่งเดิมซ้ำๆ

ในบางครั้งอาจพบปัญหาจากเครื่องพิมพ์ โดยการพิมพ์ที่ผิดพลาดของเครื่องพิมพ์ ทำให้ฉลากที่พิมพ์เกิดความเสียหาย และบาร์โค้ดไม่สมบูรณ์ ซึ่งจำเป็นต้องสร้างเงื่อนไขภายในโปรแกรมสำหรับการปล่อยข้ามฉลากที่ไม่สมบูรณ์ออกไป เนื่องจากการสั่งพิมพ์ฉลากตัวที่เสียหายอีกครั้งนั้น จะทำให้การจัดลำดับของฉลากและลำดับของฮาร์ดดิสก์ไม่ตรงกัน จึงจำเป็นต้องปล่อยทิ้งไป การแจ้งเตือนจะเริ่มทำงาน เมื่อระบบวิทัศน์ไม่สามารถตรวจจับบาร์โค้ดได้ภายในระยะเวลาที่กำหนด จะให้โปรแกรมแจ้งเตือน ดังรูปที่ 4.10



รูปที่ 4.10 รูปแสดงการสั่งข้ามฉลากที่ไม่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งหลังจากดำเนินการตามขั้นตอนของเครื่องทั้งหมดเสร็จแล้ว ยังต้องคำนึงถึงระยะเวลาที่ใช้ทั้งหมดสำหรับการติดสติ๊กเกอร์ด้วย เพื่อไม่ให้ระยะเวลาที่ใช้ดำเนินการมีค่าสูงเกินไป จึงจำเป็นต้องปรับแก้ขั้นตอนการทำงานของโปรแกรมให้เครื่องมีความรวดเร็วในการดำเนินการมากยิ่งขึ้น และต้องทำการทดสอบฟังก์ชั่นต่างๆในเครื่องให้ทำงานถูกต้องตามที่ต้องการ

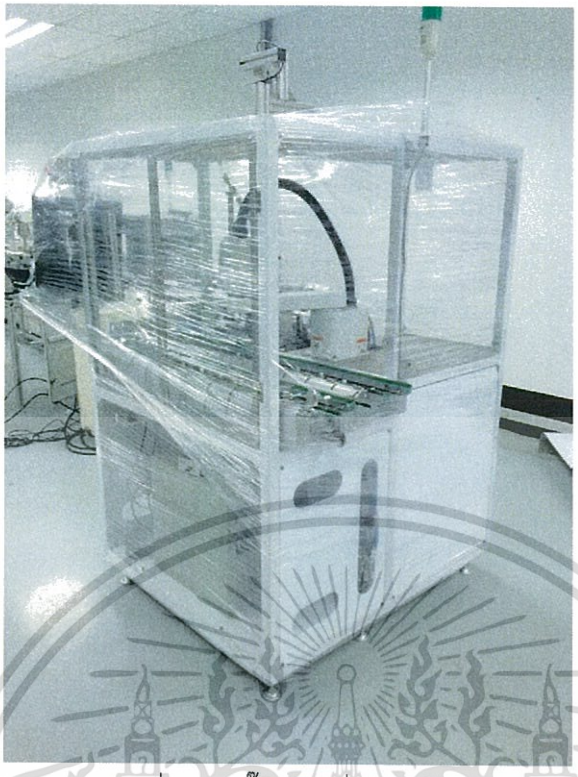
ตารางที่ 4-1 ตารางแสดงผลการทดสอบการทำงานของแต่ละฟังก์ชั่น

ฟังก์ชั่นที่ทดสอบการทำงาน	ผลการทดสอบ
ฟังก์ชั่นการควบคุมการอ่านค่าบาร์โค้ดของเครื่องอ่านบาร์โค้ด	สามารถทำงานได้
ฟังก์ชั่นการลำเลียงฮาร์ดดิสก์บนคอนเวเยอร์	สามารถทำงานได้
ฟังก์ชั่นการลำเลียงฝาครอบบนคอนเวเยอร์	สามารถทำงานได้
ฟังก์ชั่นการส่งข้อมูลไปยังเครื่องพิมพ์ฉลาก	สามารถทำงานได้
ฟังก์ชั่นการยกลิฟเตอร์สำหรับยึดตำแหน่งของฝาครอบ	สามารถทำงานได้
ฟังก์ชั่นการยกเทเบิ้ลฟีดสำหรับการหยิบสติ๊กเกอร์ของแขนกล	สามารถทำงานได้
ฟังก์ชั่นการควบคุมการเคลื่อนที่ของแขนกล	สามารถทำงานได้

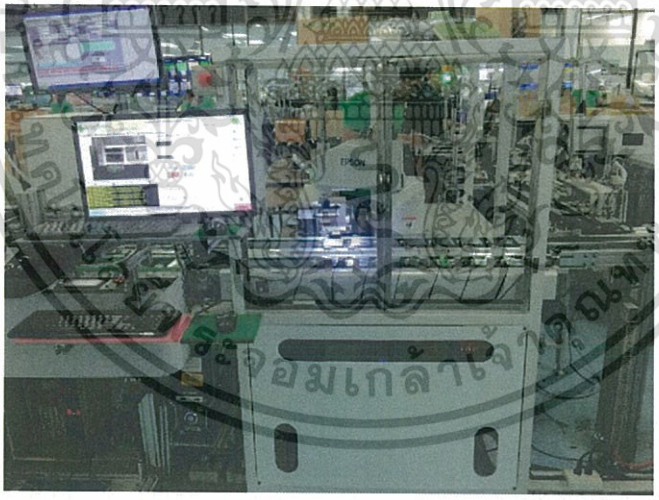
เมื่อทำการทดสอบการทำงานเสร็จแล้ว ก็จะนำเครื่องไปติดตั้งในสายการผลิต โดยจำเป็นต้องห่อหุ้มด้วยพลาสติก เพื่อความสะดวกในการเคลื่อนย้ายเครื่อง

รูปที่ 4.11 รูปแสดงเครื่องติดสติ๊กเกอร์บาร์โค้ดหลังทดสอบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 รูปแสดงเครื่องติดสติ๊กเกอร์ที่ถูกห่อด้วยพลาสติกสำหรับการเคลื่อนย้าย



รูปที่ 4.13 รูปแสดงตำแหน่งเครื่องติดสติ๊กเกอร์ที่ติดตั้งในสายการผลิต

หลังจากดำเนินการติดตั้งเครื่องติดสติ๊กเกอร์ในสายการผลิตแล้ว จึงทำการทดสอบการทำงานเพื่อเก็บผลการทำงานของแต่ละฟังก์ชั่นอีกครั้ง และดูผลการทำงานโดยรวม ระยะเวลาที่ใช้ คุณภาพการติด และความผิดพลาดของโปรแกรม โดยทำการเก็บผลการดำเนินงานเป็นระยะเวลา 5 วัน เพื่อทดสอบขั้นตอนการทำงาน ได้ผลดังตารางที่ 4-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4-2 ตารางแสดงผลการทำงานของเครื่องติดสติ๊กเกอร์

Auto Stick Label	Day 1	Day 2	Day 3	Day 4	Day 5
Total loading	295	166	477	305	285
Reject	10	8	11	11	6
%Yield rate	96.61%	95.18%	97.69%	96.39%	97.89%
%Program error	0.00%	0.00%	0.00%	0.00%	0.00%
%Have a bubble	0.00%	0.00%	0.00%	0.00%	0.00%
%Can't scan barcode	0.00%	0.00%	0.00%	0.00%	0.00%
%Stick fail	1.36%	3.01%	2.10%	2.95%	1.40%
%Printer error	2.03%	1.81%	0.21%	0.66%	0.70%
Robot program error	-	-	-	-	-
Can't scan barcode	-	-	-	-	-
Label have bubble	-	-	-	-	-
Stick fail position	4	5	10	9	4
Scratch defect	-	-	-	-	-
Printer error	6	3	1	2	2

จากตารางที่ 4.2 สรุปได้ว่า ระยะเวลาเฉลี่ยที่ใช้ดำเนินการแต่ละชิ้นคือ 4.66 วินาที และมีประสิทธิภาพการทำงานที่ 97.89% โดยมีอัตราความผิดพลาด 2.11%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

เครื่องติดสติ๊กเกอร์บาร์โค้ดอัตโนมัติ ที่ได้ศึกษาและพัฒนานั้น สามารถใช้งานได้ตามวัตถุประสงค์ คือ สามารถทำงานได้ตามขั้นตอนที่ต้องการ ทั้งการลำเลียงฝาครอบฮาร์ดดิสก์และ และการควบคุมการเคลื่อนไหวของแขนกล สามารถทำงานได้ ซีเรียลนัมเบอร์บนฮาร์ดดิสก์และฉลากไม่มีความผิดพลาด และตำแหน่งการติดสติ๊กเกอร์อยู่ในเกณฑ์มาตรฐาน ซึ่งมีประสิทธิภาพในการดำเนินงาน ดังแสดงในตาราง 5.1

ตารางที่ 5-1 ตารางแสดงประสิทธิภาพการทำงานของเครื่องติดสติ๊กเกอร์

หัวข้อในการเปรียบเทียบ	ก่อนปรับปรุง	หลังปรับปรุง	ประสิทธิภาพ
การจ้างแรงงาน	2 คน/วัน	0 คน/วัน	ลดลง 100%
ระยะเวลาที่สามารถดำเนินการ	15.67 ชั่วโมง/วัน	24 ชั่วโมง/วัน	เพิ่มขึ้น 65%
เวลาที่ใช้ในการผลิต	6 วินาที/ชิ้น	5.22 วินาที/ชิ้น	ลดลง 13%
ความสามารถในการผลิตสูงสุด	9200 ชิ้น/วัน	16202 ชิ้น/วัน	เพิ่มขึ้น 76%

5.2 ปัญหาและอุปสรรค

5.2.1 การเริ่มต้นดำเนินงานมีความล่าช้า เนื่องจากจำเป็นต้องรอวัสดุที่สั่งซื้อเป็นระยะเวลานาน

5.2.2 ระยะเวลาในการทดสอบการใช้งานมีน้อย ทำให้โอกาสเจอข้อผิดพลาดมีน้อยลง และฮาร์ดดิสก์ที่สามารถใช้ทดสอบมีจำนวนจำกัด ส่งผลให้การทดสอบไม่สามารถทำได้อย่างต่อเนื่อง

5.2.3 พอร์ต I/O มีจำนวนไม่เพียงพอต่อการใช้งาน จึงจำเป็นต้องตัดอินพุตบางตัวออกไป ซึ่งเมื่อทำการเขียนโปรแกรม อาจไม่ครอบคลุมขั้นตอนการทำงานได้ทั้งหมด

5.2.4 การเชื่อมต่อกับระบบของโรงงานหลังการติดตั้งมีความล่าช้า เนื่องจากโรงงานใช้เวลานานในการปรับแก้โปรแกรม ให้สามารถทำตามวัตถุประสงค์ของเครื่องได้

5.3 ข้อเสนอแนะ

5.3.1 ควรมีการจัดการวางแผนงานโปรเจคให้มีประสิทธิภาพและกำหนดการที่แน่นอนมากกว่านี้

5.3.2 เพิ่มจำนวนพอร์ต I/O เพื่อให้สามารถเขียนโปรแกรมครอบคลุมการทำงานได้ทั้งหมด

5.3.3 เปลี่ยนมอเตอร์ให้มีกำลังขับสูง จะทำให้คอนเวเยอร์สามารถหมุนได้เร็วมากขึ้น

5.3.4 ควรมีอุปกรณ์ที่ใช้ในการทดสอบมากกว่านี้ เพื่อการทดสอบการทำงานที่ต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ทศพล รัตนพงษ์วิสุทธิ์. 2558. สายพานลำเลียงแผ่นวงจรอิเล็กทรอนิกส์แบบอัตโนมัติของเครื่องทดสอบวงจร. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิศวกรรมระบบการผลิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2558.

[2] ทฤษฎีหุ่นยนต์อุตสาหกรรม [online], เข้าถึงได้จาก http://www.applicadthai.com/business/articles/industrial-robot-type/ (สืบค้นวันที่ 10 พฤศจิกายน 2559).

[3] ระบบนิวแมติกส์ [online], เข้าถึงได้จาก http://mte.kmutt.ac.th/elearning/Bicycle_Air_Engine/lesson1_2.html (สืบค้นวันที่ 10 พฤศจิกายน 2559).

[4] หลักการทำงานของโซลินอยด์วาล์ว (Solinoidvalve) [online], เข้าถึงได้จาก http://www.thaiwatersystem.com/article/45/หลักการทำงานของโซลินอยด์วาล์ว-solinoidvalve/ (สืบค้นวันที่ 10 พ.ย. 59).

[5] สรวิศ ศิริมาลีวัฒนา. 2558. หุ่นยนต์ใส่อุปกรณ์อิเล็กทรอนิกส์อัตโนมัติ. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิศวกรรมระบบการผลิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2558.

[6] Barcode Scanner เครื่องอ่านบาร์โค้ด คืออะไร [Online], เข้าถึงได้จาก http://www.esssyntech.com/index.php/Products-Knowledge/barcode-scanner-knowledge.html (สืบค้นวันที่ 19 พ.ย. 59).

[7] รีดสวิตช์ (reed switch) [Online], เข้าถึงได้จาก http://phchitchai.wbvschool.net/archives/1402 (สืบค้นวันที่ 19 พ.ย. 59).

[8] รีเลย์ (relay) สวิตช์ไฟฟ้าที่นักประดิษฐ์จำเป็นต้องรู้จัก [Online], เข้าถึงได้จาก http://www.inventor.in.th (สืบค้นวันที่ 20 พ.ย. 59).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

1 System management commands

Reset	Resets the controller.
SysConfig	Displays controller setup.
SysErr	Returns the latest error status or warning status.
Date	Sets the system date.
Time	Sets system time.
Date$	Returns the system date as a string.
Time$	Returns system time as a string.
Hour	Displays / returns controller operation time.
Stat	Returns controller status bits.
CtrlInfo	Returns controller information.
RobotInfo	Returns robot information.
RobotInfo$	Returns robot text information.
TaskInfo	Returns task information.
TaskInfo$	Returns task text information.
DispDev	Sets the current display device.
EStopOn	Return the Emergency Stop status.
CtrlDev	Returns the current control device number.
Cls	Clears the EPSON RC+ 6.0 Run, Operator, or Command window text area. Clears the TP print panel.
Toff	Turns off execution line display on the LCD.
Ton	Specifies a task which shows a execution line on the LCD.
SafetyOn	Return the Safety Door open status.
Eval	Executes a Command window statement from a program and returns the error status.
ShutDown	Shuts down EPSON RC+ and optionally shuts down or restarts Windows.
SetLCD	Sets or displays how the controller's LCD panel displays data.
TeachOn	Returns the Teach mode status.
WindowsStatus	Returns the Windows startup status.

2 Robot control commands

AtHome	Retuns if the current robot orientation is Home position or not.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Calib	Replaces the current arm posture pulse values with the current CalPls values.
CalPls	Specifies and displays the position and orientation pulse values for calibration.
Hofs	Returns the offset pulses used for software zero point correction.
MCal	Executes machine calibration for robots with incremental encoders.
MCalComplete	Returns status of MCal.
MCordr	Specifies and displays the moving joint order for machine calibration.
Mcal.	Required only for robots with incremental encoders.
Power	Sets / returns servo power mode.
Motor	Sets / returns motor status.
SFree	Removes servo power from the specified servo axis.
SLock	Restores servo power to the specified servo axis.
SyncRobots	Start the reserved robot motion.
Jump	Jumps to a point using point to point motion.
Jump3	Jumps to a point using 3D gate motion.
Jump3CP	Jumps to a point using 3D motion in continuous path.
Arch	Sets / returns arch parameters for Jump motion.
LimZ	Sets the upper Z limit for the Jump command.
LimZMargin	Sets / returns the margin for error detection when the operation starts at the position higher than LimZ value.
Sense	Sets / returns the condition to stop the manipulator above the target coordinate when Sense is specified by Jump command.
JS	Returns status of Sense operation.
JT	Returns the status of the most recent Jump command for the current robot.
Go	Moves the robot to a point using point to point motion.
Pass	Executes simultaneous four joint Point to Point motion, passing near but not through the specified points.
Pulse	Moves the robot to a position defined in pulses.
BGo	Executes Point to Point relative motion, in the selected local coordinate system.
BMove	Executes linear interpolation relative motion, in the selected local coordinate system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TGo	Executes Point to Point relative motion, in the current tool coordinate system.
TMove	Executes linear interpolation relative motion, in the selected tool coordinate system.
Till	Specifies motion stop when input occurs.
TillOn	Returns the current Till status.
!...!	Process statements during motion.
Speed	Sets / returns speed for point to point motion commands.
Accel	Sets / returns acceleration and deceleration for point to point motion.
SpeedFactor	Sets / returns speed for point to point motion commands.
Inertia	Specifies or displays the inertia settings of the robot arm.
Weight	Specifies or displays the weight settings of the robot arm.
Arc	Moves the arm using circular interpolation.
Arc3	Moves the arm in 3D using circular interpolation.
Move	Moves the robot using linear interpolation.
Curve	Defines the data and points required to move the arm along a curved path. Many data points can be defined in the path to improve precision of the path.
CVMove	Performs the continuous spline path motion defined by the Curve instruction.
SpeedS	Sets / returns speed for linear motion commands.
AccelS	Sets / returns acceleration and deceleration for linear motion.
SpeedR	Sets / returns speed for tool rotation.
AccelR	Sets / returns acceleration and deceleration for tool rotation.
AccelMax	Returns maximum acceleration value limit available for Accel.
Brake	Turns brake on or off for specified joint of the current robot.
Home	Moves robot to user defined home position.
HomeClr	Clears the home position definition.
HomeDef	Returns status of home position definition.
HomeSet	Sets user defined home position.
Hordr	Sets motion order for Home command.
InPos	Checks if robot is in position (not moving).
CurPos	Returns current position while moving.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCPSpeed	Returns calculated current tool center point velocity.
Pallet	Defines a pallet or returns a pallet point.
PalletClr	Clears a pallet definition.
Fine	Sets positioning error limits.
QP	Sets / returns Quick Pause status.
QPDecelR	Sets the deceleration speed of quick pause for the change of tool orientation during the CP motion.
QPDecelS	Sets the deceleration speed of quick pause in the CP motion.
CP	Sets CP (Continuous Path) motion mode.
Box	Specifies and displays the approach check area.
BoxClr	Clears the definition of approach check area.
BoxDef	Returns whether Box has been defined or not.
Plane	Specifies and displays the approach check plane.
PlaneClr	Clears (undefines) a Plane definition.
PlaneDef	Returns the setting of the approach check plane.
InsideBox	Displays a prompt in a dialog box, waits for the operator to input text or choose a button, and returns the contents of the box.
InsidePlane	Returns the check status of the approach check plane.
GetRobotInsideBox	Returns a robot which is in the approach check area.
GetRobotInsidePlane	Returns a robot which is in the approach check plane.
	Find Specifies or displays the condition to store coordinates during motion.
FindPos	Returns a robot point stored by Fine during a motion command.
PosFound	Returns status of Find operation.
WaitPos	Waits for robot to decelerate and stop at position before executing the next statement while path motion is active.
Robot	Selects the current robot.
RobotModel$	Returns the robot model name.
RobotName$	Returns the robot name.
RobotSerial$	Returns the robot serial number.
RobotType	Returns the robot type.
TargetOK	Returns a status indicating whether or not the PTP (Point to Point) motion from the current position to a target position is possible.
JRange	Sets / returns joint limits for one joint.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Range	Sets limits for all joints.
XYLim	Sets or displays the permissible XY motion range limits for the robot.
XYLimClr	Clears the XYLim definition.
XYLimDef	Returns whether XYLim has been defined or not.
XY	Returns a point from individual coordinates that can be used in a point expression.
Dist	Returns the distance between two robot points.
PTPBoost	Specifies or displays the acceleration, deceleration and speed algorithmic boost parameter for small distance PTP (point to point) motion.
PTPBoostOK	Returns whether or not the PTP (Point to Point) motion from a current position to a target position is a small travel distance.
PTPTime	Returns the estimated time for a point to point motion command without executing it.
CX	Sets / returns the X axis coordinate of a point.
CY	Sets / returns the Y axis coordinate of a point.
CZ	Sets / returns the Z axis coordinate of a point.
CU	Sets / returns the U axis coordinate of a point.
CV	Sets / returns the V axis coordinate of a point.
CW	Sets / returns the W axis coordinate of a point.
CR	Sets / returns the R axis coordinate of a point.
CS	Sets / returns the S axis coordinate of a point.
CT	Sets / returns the T axis coordinate of a point.
Pls	Returns the pulse value of one joint.
Agl	Returns joint angle at current position.
PAgl	Return a joint value from a specified point.
JA	Returns a robot point specified in joint angles.
AglToPls	Converts robot angles to pulses.
DegToRad	Converts degrees to radians.
RadToDeg	Converts radians to degrees.
Joint	Displays the current position for the robot in joint coordinates.
JTran	Perform a relative move of one joint.
PTran	Perform a relative move of one joint in pulses.
RealPls	Returns the pulse value of the specified joint.
RealPose	Returns the current position of the specified robot.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PPls	Return the pulse position of a specified joint value from a specified point.
LJM	Function Returns the point data with the orientation flags converted to enable least joint motion when moving to a specified point based on the reference point.
AutoLJM	Sets the Auto LJM
AutoLJM	Function Returns the state of the Auto LJM
AvoidSingularity	Sets the Singularity avoiding function
AvoidSingularity	Function Returns the state of the Singularity avoiding function
SingularityAngle	Sets the singularity neighborhood angle for the singularity avoiding function.
SingularityAngle	Function Returns the singularity neighborhood angle for the singularity avoiding function
SingularitySpeed	Sets the singularity neighborhood speed for the singularity avoiding function
SingularitySpeed	Function Returns the singularity neighborhood speed for the singularity avoiding function
AbortMotion	Aborts a motion command and puts the running task in error status.
Align	Function Returns point data converted to align robot orientation with the nearest coordinate axis in local coordinate system.
AlignECP	Function Returns point data converted to align robot orientation with a nearest coordinate axis in ECP coordinate system.
SoftCP	Sets / returns SoftCP motion mode.
SoftCP	Function Returns the status of SoftCP moton mode.
Here	Teach a robot point at the current position.
Where	Displays current robot position data.
PerformMode	Sets the mode of the robot.
PerformMode	Function Returns the robot performance mode number.

3 Input/output commands

On	Turns an output on.
Off	Turns an output off.
Oport	Reads status of one output bit.
Sw	Returns status of input.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In	Reads 8 bits of inputs.
InW	Returns the status of the specified input word port.
InBCD	Reads 8 bits of inputs in BCD format.
InReal	Reads an input data of 2 words (32 bits) as a floating-point data (IEEE754 compliant) of 32 bits.
Out	Sets / returns 8 bits of outputs.
OutW	Simultaneously sets 16 output bits.
OpBCD	Simultaneously sets 8 output bits using BCD format.
OutReal	Output the output data of real value as the floating-point data (IEEE754 compliant) of 32 bits to the output port 2 words (32 bits).
MemOn	Turns a memory bit on.
MemOff	Turns a memory bit off.
MemSw	Returns status of memory bit.
MemIn	Reads 8 bits of memory I/O.
MemOut	Sets / returns 8 memory bits.
MemInW	Returns the status of the specified memory I/O word port. Each word port contains 16 memory I/O bits.
MemOutW	Simultaneously sets 16 memory I/O bits.
Wait	Wait for condition or time.
TMOut	Sets default time out for Wait statement.
Tw	Returns the status of the Wait condition and Wait timer interval.
Input	Receives input data from the display device and stored in a variable(s).
Print	Display characters on current display window.
Line Input	Input a string from the current display window.
Input #	Allows string or numeric data to be received from a file, communications port, or database and stored in one or more variables.
Print #	Outputs data to the specified file, communications port, database, or device.
Line Input #	Reads data of one line from a file, communication port, database, or the device.
Lof	Checks whether the specified RS-232 or TCP/IP port has any lines of data in its buffer.
SetIn	For Virtual IO, sets specified input port (8 bits) to the specified value.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SetInW	For Virtual IO, sets specified input word (16 bits) to the specified value.
SetSw	For Virtual IO, sets specified input bit to the specified value.
IOLabel$	Returns the I/O label for a specified input or output bit, byte, or word.
IONumber	Returns the I/O number of the specified I/O label.
OpenCom	Open an RS-232 communication port.
OpenCom Function	Acquires the task number that executes OpenCom.
CloseCom	Close the RS-232C port that has been opened with OpenCom.
SetCom	Sets or displays parameters for RS-232C port.
ChkCom	Returns number of characters in the reception buffer of a communication port
OpenNet	Open a TCP/IP network port.
OpenNet Function	Acquires the task number that executes OpenNet.
CloseNet	Close the TCP/IP port previously opened with OpenNet.
SetNet	Sets parameters for a TCP/IP port.
ChkNet	Returns number of characters in the reception buffer of a network port
WaitNet	Wait for TCP/IP port connection to be established.
Read	Reads characters from a file or communications port.
ReadBin	Reads binary data from a file or communications port.
Write	Writes characters to a file or communication port without end of line terminator.
WriteBin	Writes binary data to a file or communications port.
InputBox	Displays a prompt in a dialog box, waits for the operator to input text or a button, and returns the contents of the box.
MsgBox	Displays a message in a dialog box and waits for the operator to choose a button.
RunDialog	Runs an EPSON RC+ 6.0 dialog from a SPEL+ program.
LatchEnable	Enable / Disable the latch function for the robot position by the R-I/O input.
LatchState Function	Returns the latch state of robot position using the R-I/O.
LatchPos Function	Returns the robot position latched using the R-I/O input signal.
SetLatch	Sets the latch function of the robot position using the R-I/O input.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 Program control commands

Function	Declare a function.
For...Next	Executes one or more statements for a specific count.
GoSub	Execute a subroutine.
Return	Returns from a subroutine.
GoTo	Branch unconditionally to a line number or label.
Call	Call a user function.
If..Then..Else..EndIf	Conditional statement execution
Else	Used with the If instruction to allow statements to be executed when the condition used with the If instruction is False. Else is an option for the If/Then instruction.
Select ... Send	Executes one of several groups of statements, depending on the value of an expression.
Do...Loop	Do...Loop construct.
Declare	Declares an external function in a dynamic link library (DLL).
Trap	Specify a trap handler.
OnErr	Defines an error handler.
Era	Returns the robot joint number for last error.
Erf$	Returns the function name for last error.
Erl	Returns the line number of error.
Err	Returns the error number.
Ert	Returns the task number of error.
Errb	Returns the robot number of error
ErrMsg$	Returns the error message.
Signal	Sends a signal to tasks executing WaitSig.
SyncLock	Synchronizes tasks using a mutual exclusion lock.
SynUnlock	Unlocks a sync ID that was previously locked with SyncLock.
WaitSig	Waits for a signal from another task.
ErrorOn	Returns the error status of the controller.
Error	Generates a user error.
EResume	Resumes execution after an error-handling routine is finished.
PauseOn	Returns the pause status.
Exit	Exits a loop construct or function.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 Program execution commands

Xqt	Execute a task.
Pause	Pause all tasks that have pause enabled.
Cont	Resumes the contoller after a Pause statement has been executed and continues the execution of all tasks.
Halt	Suspend a task.
Quit	Quits a task.
Resume	Resume a task in the halt state.
MyTask	Returns current task.
TaskDone	Returns the completion status of a task.
TaskState	Returns the current state of a task.
TaskWait	Waits to for a task to terminate.
Restart	Restarts the current main program group.
Recover	Executes safeguard position recovery and returns status.
RecoverPos	Returns the position where a robot was in when safeguard was open.
StartMain	Executes the main function from a background task.

6 Logical operations

And	Performs logical and bitwise AND operation.
Or	Or operator.
LShift	Shifts bits to the left.
Mod	Modulus operator.
Not	Not operator.
RShift	Shifts bits to the right.
Xor	Exclusive Or operator.
Mask	Performs bitwise AND operation in Wait statements.

7 Variable commands

Boolean	Declares Boolean variables.
Byte	Declares byte variables.
Double	Declares double variables.
Global	Declares global variables.
Int32	Declares 4-byte integer variables.
Integer	Declares 2-byte integer variables.
Long	Declares long integer variables.
Real	Declares real variables.
Short	Declares 2-byte integer variables.
String	Declares string variables.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

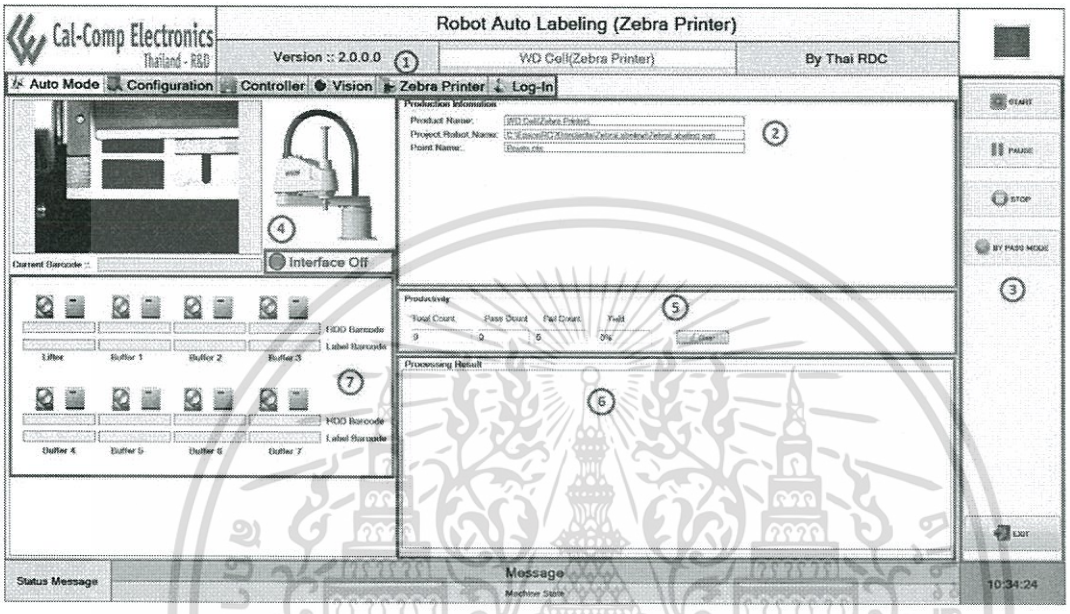
UByte	Declares unsigned integer variables.
UInt32	Declares unsigned 4-byte integer variables.
UShort	Declares unsigned 2-byte integer variables.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

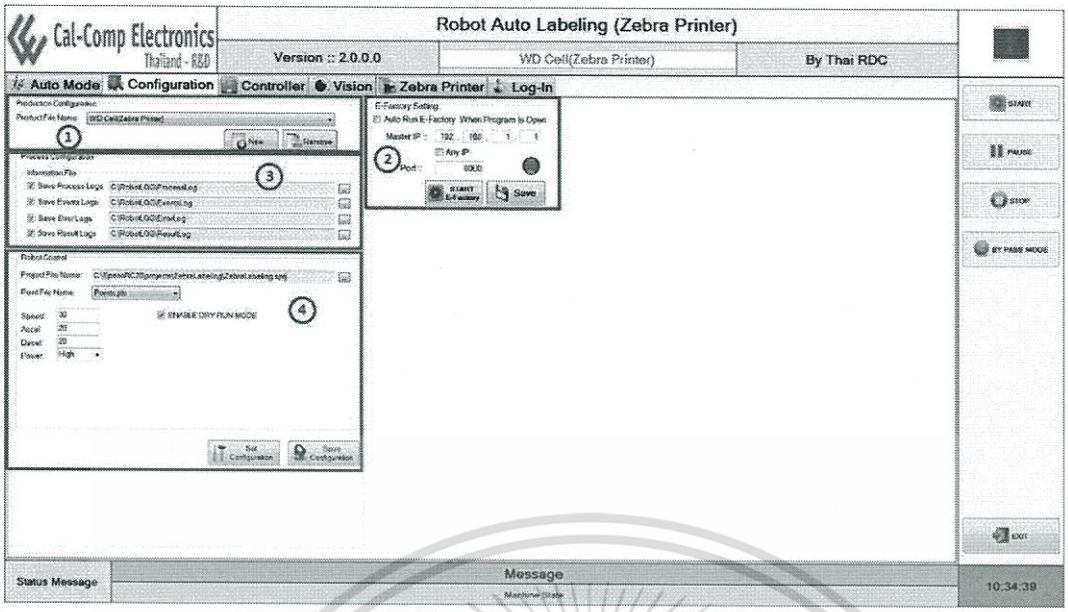
Manual for Auto Stick Label from Zebra Printer

About Program



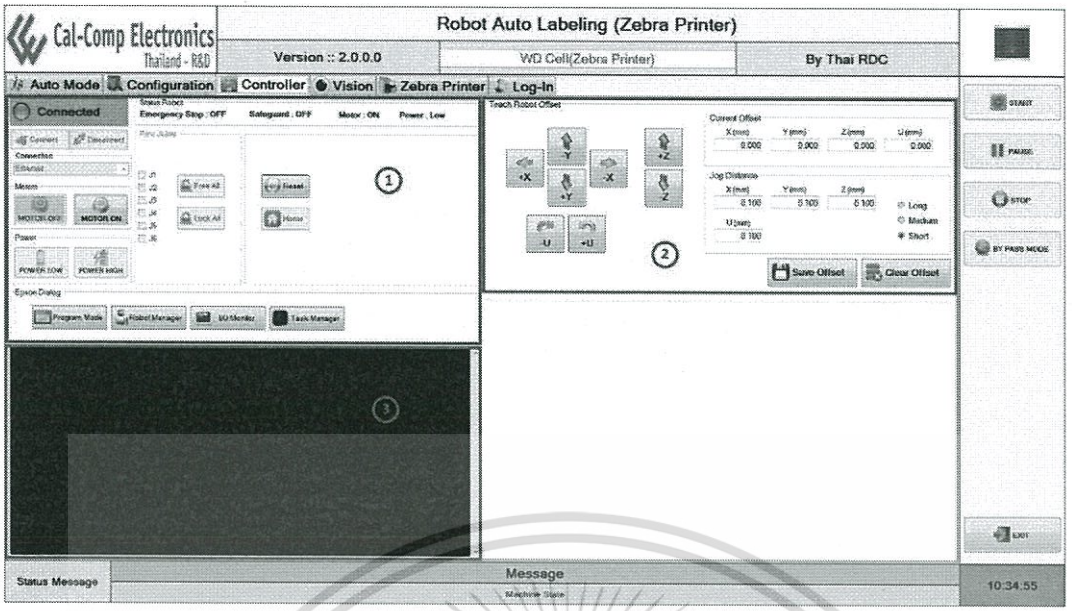
1. Button for change any system of the program.
2. Area for show detail of file configuration.
3. Button for control working of the program.
4. Show the status of Interface connection.
5. Area for show productivity count in the program.
   - You can click button “Clear” for reset all product count.
6. Area for show the status of all process result.
7. Area for show the status of Hard disk, Lower case and serial number on the machine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. Area for show the production configuration of the program.
   - You can change the product file name by drop down the "ProductFile Name" or create new product file name by button "New" or click button "Remove" for delete the file name.
2. Area for setting about E-Factory
   - You can set the IP address for connect to the factory at "Master IP ::" and select port for connection and save configuration by click button "Save".
   - You can start connecting to E-factory by click button "START E-Factory".
3. Area for setting process configuration file
   - You can tick on the tick button for select save logs file and you can browse the location for save logs file by click button ".."
4. Area for control robot project file name, point file name and set speed of robot.
   - You can browse the project file name for use by click button "..".
   - You can change the point file by drop down the button "Point File Name:"
   - You can set the robot speed, robot accelerate and robot decelerate by insert the number into the textbox. And You can change the power of the robot low or high by select drop down "Power:".
   - You can test run without connecting to the factory by tick the button "ENABLE DRY RUN MODE".
   - You have to click button "Save Configuration" and "Set configuration" after setting for save your setting.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

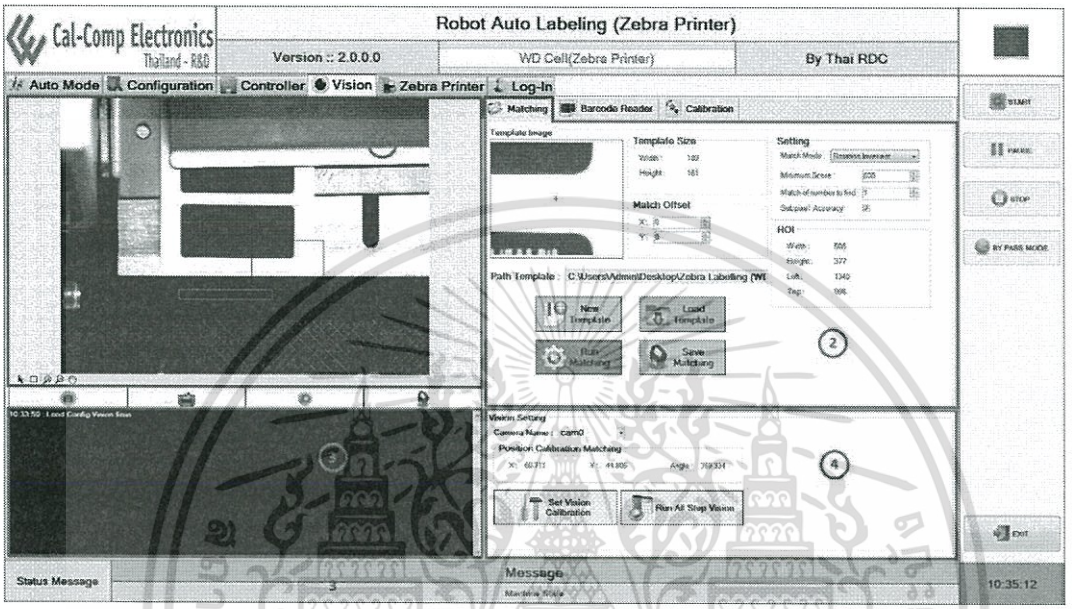


1. Area for control the robot and controller.
   - If you already connected to controller the color will be green and show "Connected" and if you are not connect to controller yet, the color will be red and show disconnected.
   - You can change the connection to controller by drop down the button "Connection".
   - You can control the robot motor by click the button "MOTOR ON" to turn on the robot motor. And turn off the motor by click the button "MOTOR OFF".
   - You can select the power of the robot by click the button "POWER LOW" or "POWER HIGH".
   - If you already turn on the motor you can click the button "Free All" for move the robot by yourself or you can tick into the box "J1" to set free joint move for only joint 1. And you can do like this for another joint.
   - If you have robot error you have to click button "Reset" for reset the robot.
   - You can click "HOME" button for move robot to point home after you turn on the robot motor.
   - You can click "Program Mode" button to open the program EPSON RC+7.0
   - You can click "Robot Manager" button to open the Robot Manager in the program EPSON RC+7.0
   - You can click "I/O Monitor" button to see the input and output of controller.
   - You can click "Task Manager" to open task manager.
2. Area for teach robot offset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- If your position to stick is not good, you can add the offset to stick by click the direction button “+X”, “-X” for X axis, “+Y”, “-Y” for Y axis, “+Z”, “-Z” for up and down axis, “+U”, “-U" for rotate Z axis.
- After finished set offset, you have to click “Save Offset” button for save.

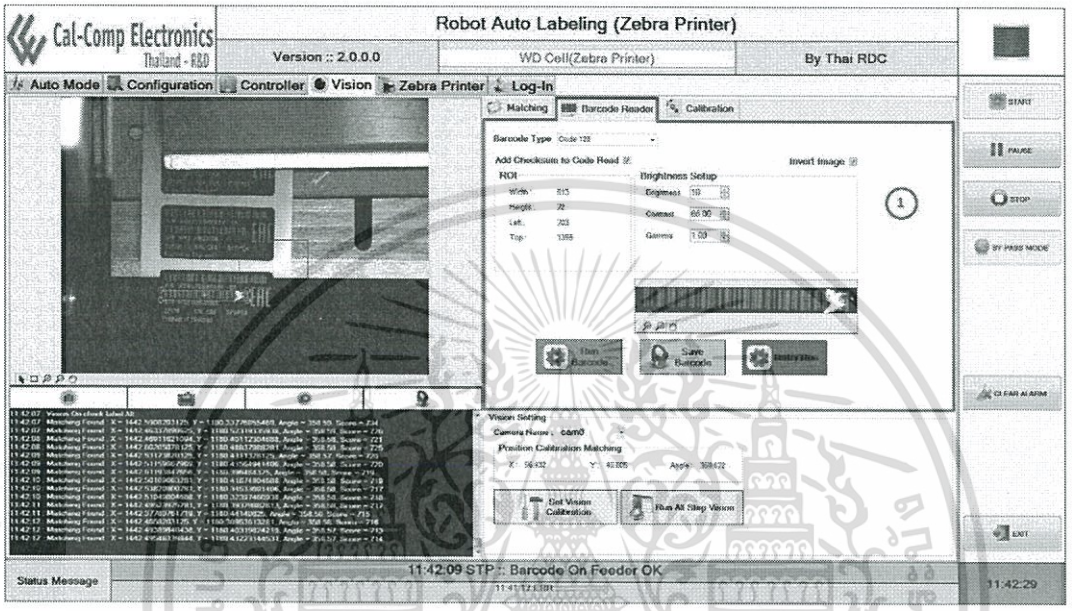
3. Area for show status of the controller.



1. Area for show the photo from the camera.
   - You can use tools on the left down corner of this area for zoom in, zoom out and move picture from camera.
2. Area for set matching position of label from printer.
   - You can set matching mode “Rotation Invariant” or “Shift Invariant” by drop down the “Match Mode” button.
   - You can set minimum score of matching and number to find matching in “Setting” area.
   - “Path Template” show the location of the template file.
   - You can set new template by click “New Template” button.
   - You can load another template by click “Load Template” button.
   - You can find the matching of your current photo by “Run Matching” button.
   - If you finished set the matching, you have to click “Save Matching” button for save matching to your configuration.
3. Area for show the status of the vision system.
4. Area for set vision system.

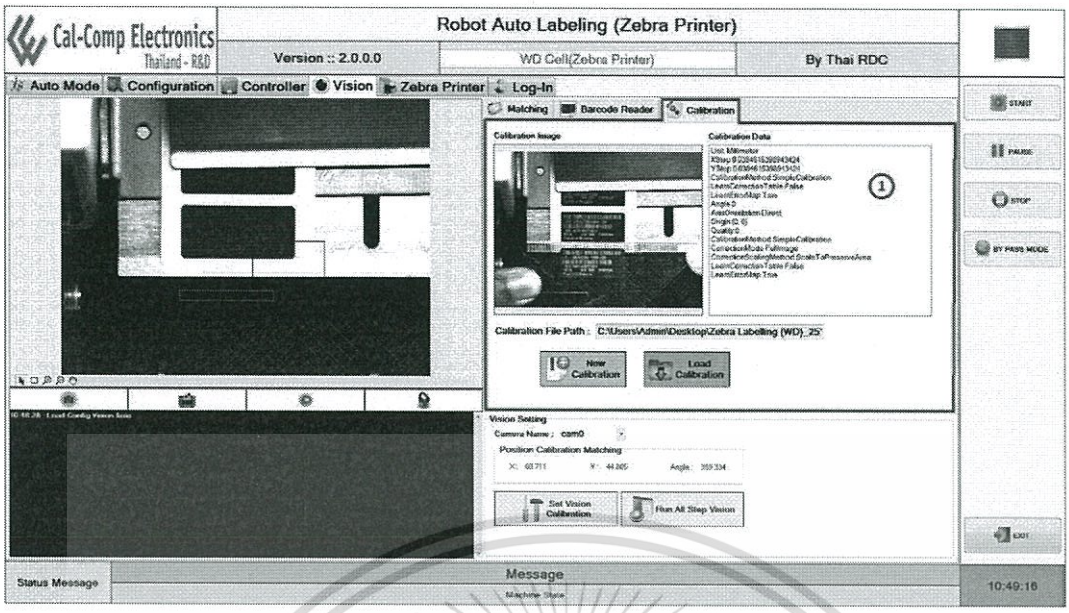
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- You can select your camera by click drop down “Camera Name”
- You can see the position matching in “Position Calibration Matching” area.
- You can test all of the vision system by click “Run All Step Vision”
- If you finished setting the vision calibrate, you have to click “Set Vision Calibration” button to save your setting.

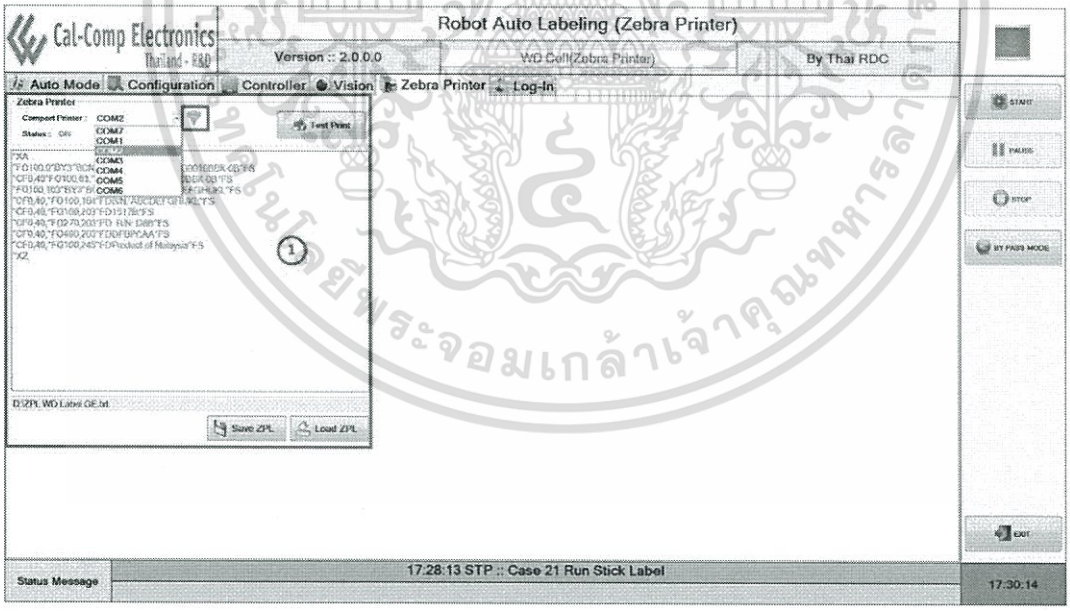


1. Area for set barcode reader from camera.
   - You can change barcode type by drop down the “Barcode Type” button.
   - You can set the brightness, contrast and gamma of the photo by change the value of it.
   - “ROI” area show your position of barcode reader.
   - You can click “Run Barcode” button to test read barcode from camera.
   - You can click “Save Barcode” button when you finish setting barcode.
   - “Retry Run” button will show up when your machine is running but cannot find the barcode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

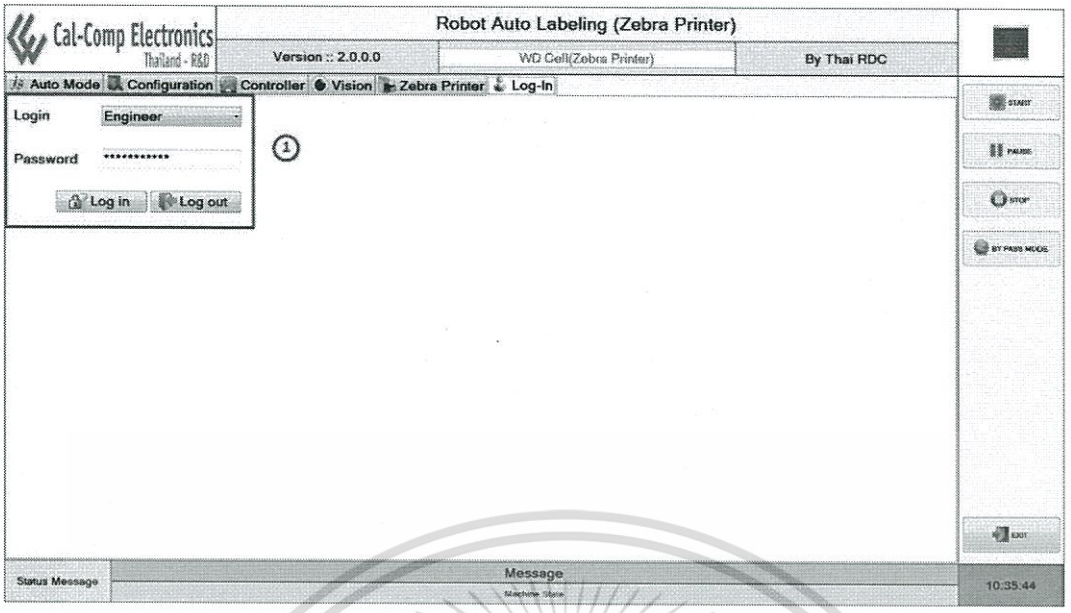


1. Area for set the calibration of camera
   - "Calibration File Path" show the location of the calibration file.
   - You can set new calibrate by click "New Calibration" button.
   - You can browse and load the calibration by click "Load Calibration" button.



1. Area for control printer
   - You can select port of your printer by drop down and click on picture button.
   - You can test your printer by click "Test Print" button.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. Area for login
   - Before you can use this program you have to login first.
   - Username “Engineer” and Password “Thairdc2016”
   - Username “Operator” and “Operator”
   - Username “Production” and “Production”

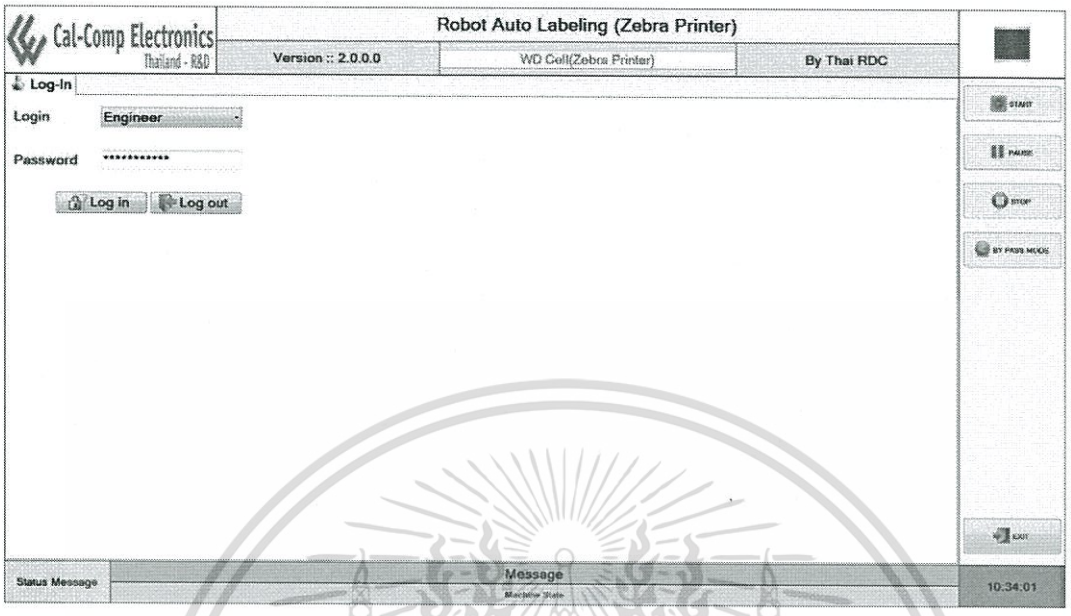
How to use

1. Open the program

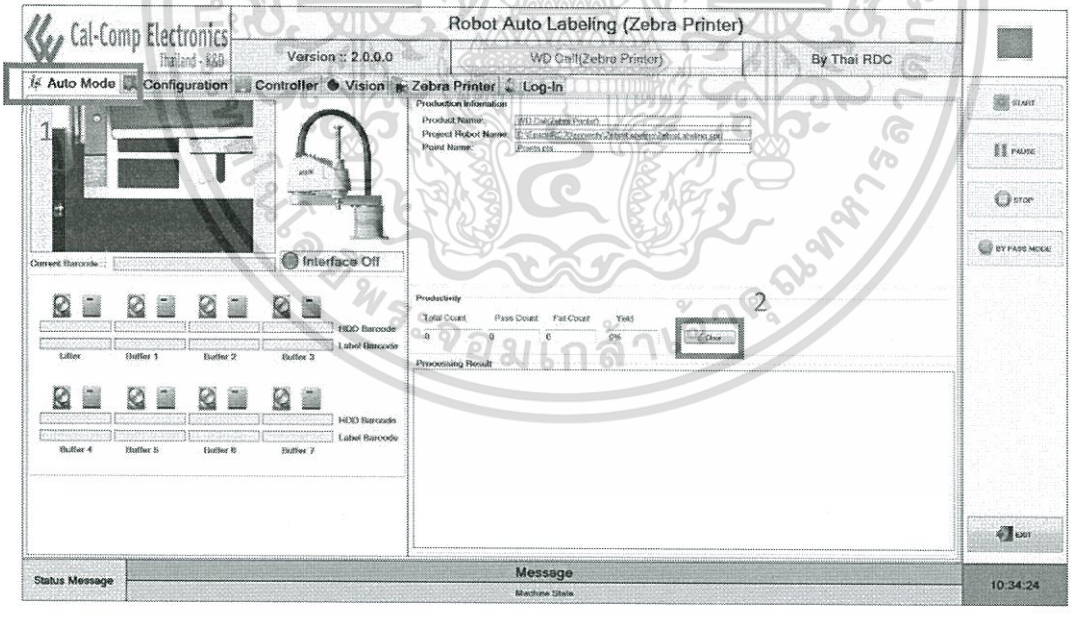
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Login: Engineer

   Password: Thairdc2016



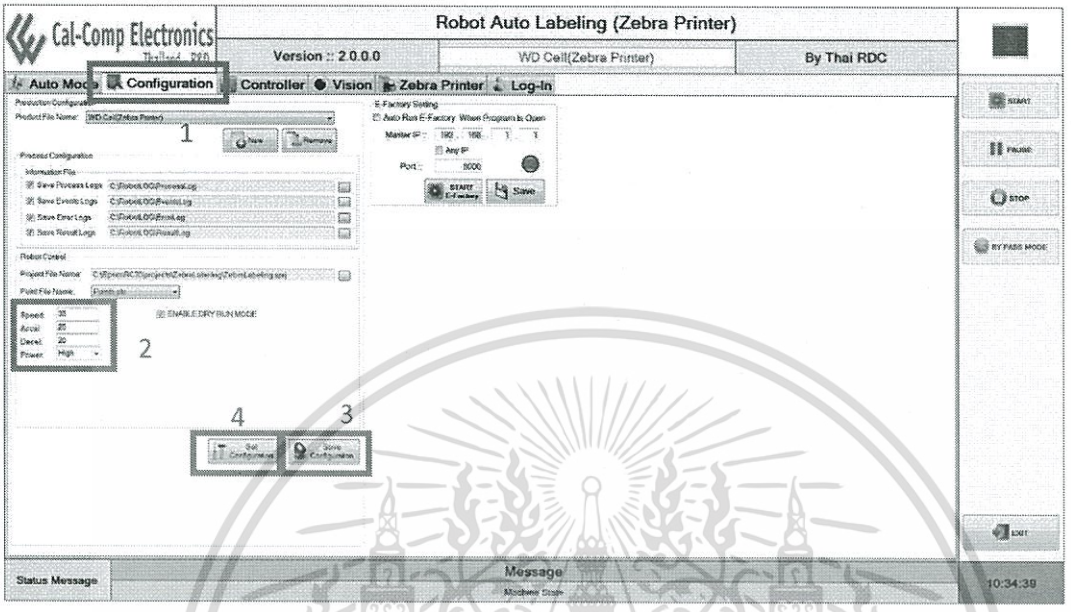
3. Select Auto Mode > Clear Productivity



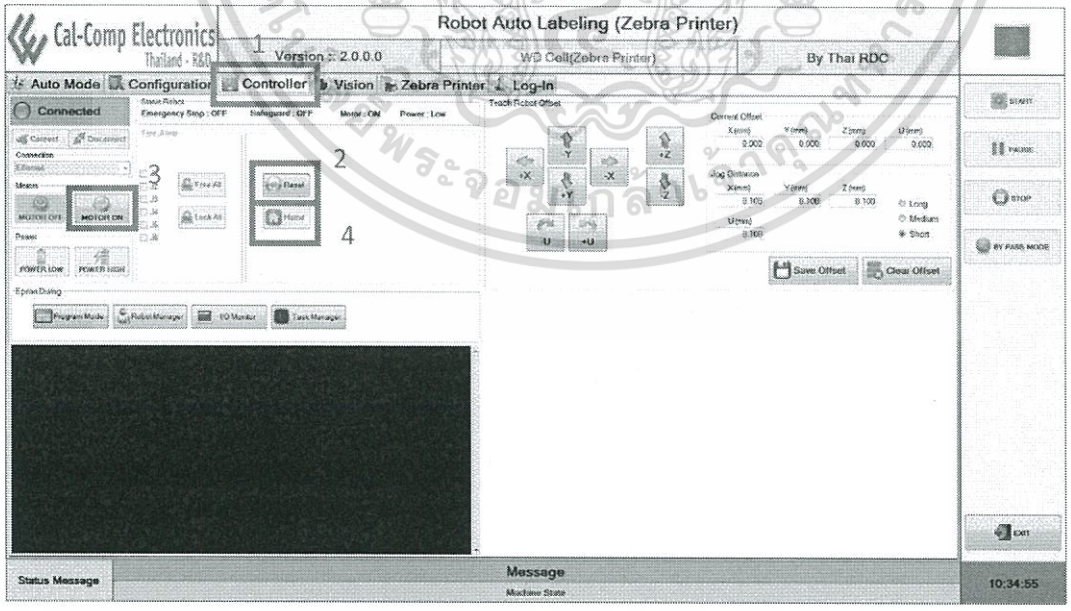
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Select configuration > Insert speed of robot
   - Click save configuration > Click set configuration

   *Tick dry run mode for test machine without connect to the factory

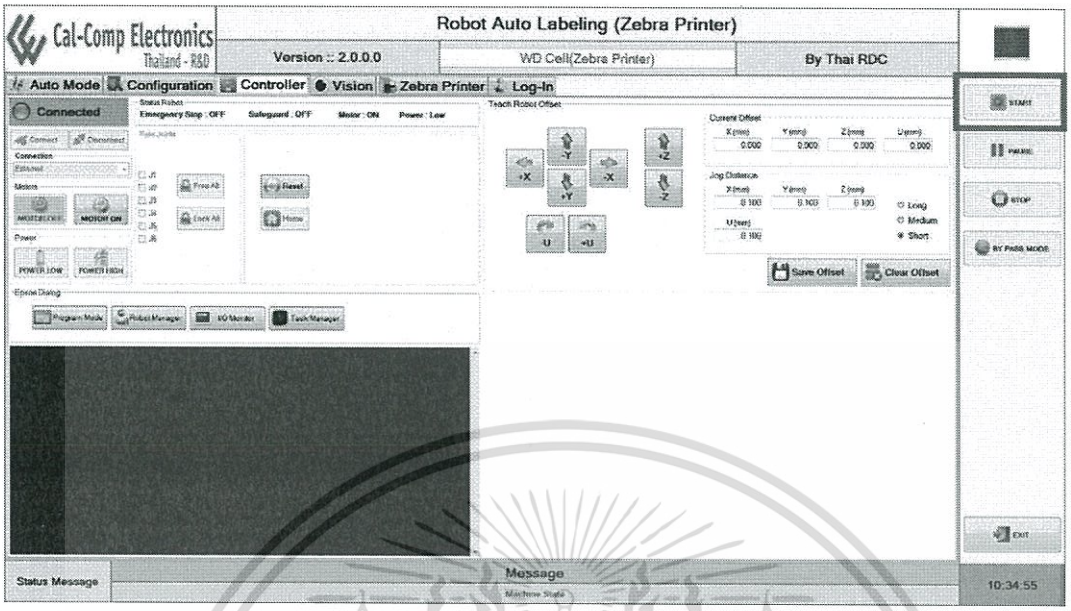


5. Select Controller > Click Reset > Click MOTOR ON > Click HOME

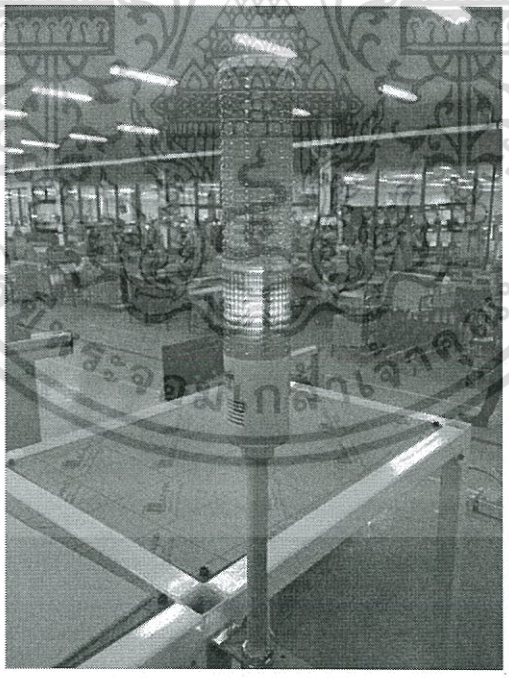


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Click START button



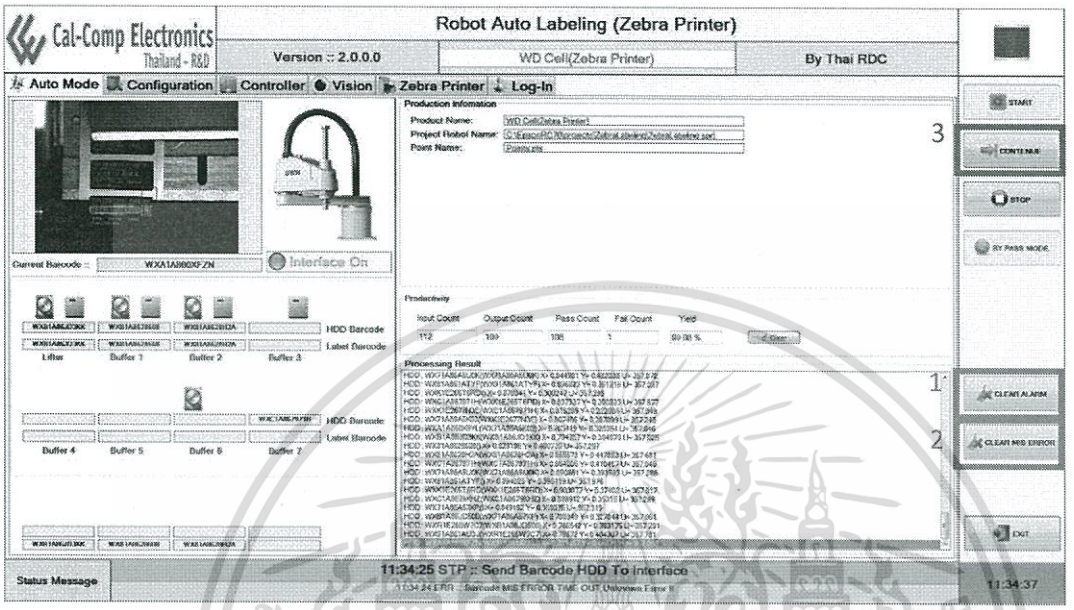
7. Waiting for tower light green > Ready to work



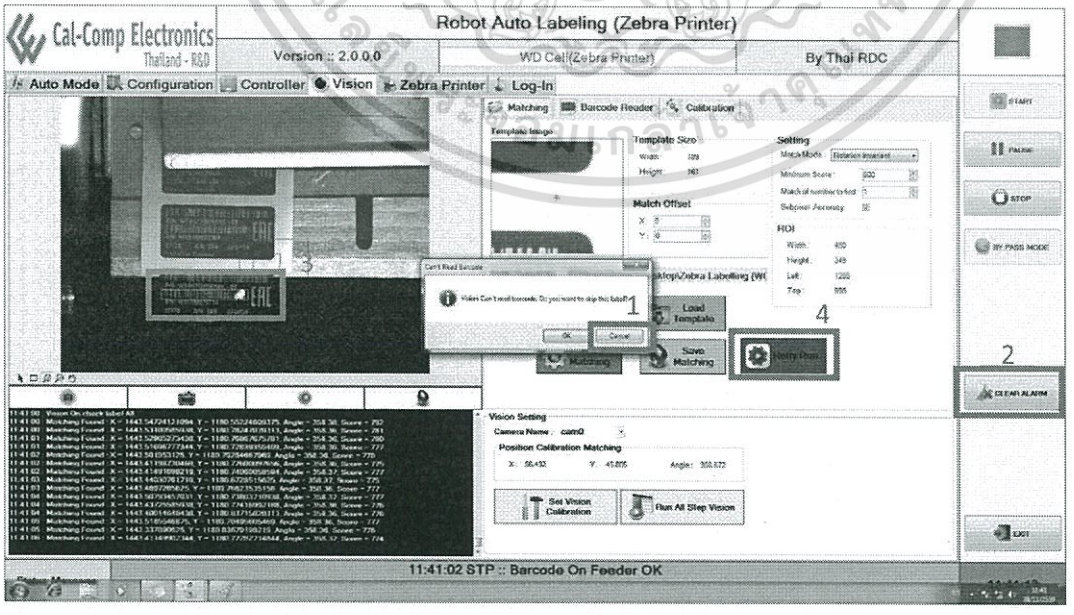
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Solve the problem

1. For error MIS
   - Clear alarm > Clear MIS error > Continue



2. For Error label from printer have 2 way to solve the problem
   - Select OK
   - Select Cancel > Clear alarm > Adjust position > Retry Run



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ฟังก์ชั่นต่างๆที่ใช้ในการควบคุมคอนโทรลเลอร์

Function MAIN

```
Global Preserve Integer IniSpeed
Global Preserve Integer IniAccel
Global Preserve Integer IniDecel
Global Preserve String IniPower$
Global Preserve Integer HDDPass
Global Preserve Integer HDDFail
Global Preserve Integer ContenueFail(8)
Global Preserve Integer HDDIncomming
Global Preserve String STA_Label$
Global Preserve String OnHandBarcode$
Global Preserve String OnLifterBarcode$
Global Preserve String OnFeed$
Global Preserve String CurrentBarcode$
Global Preserve Boolean PrintLast
Global Preserve String OnPrint$(8)
Global Preserve String OnSPBarcode$(8)
Global Preserve String OnFDBarcode$(8)
Global Preserve Real X_Cal
Global Preserve Real Y_Cal
Global Preserve Real A_Cal
Global Preserve Integer Offset(5)
Global Preserve Boolean DRY_RUN_MODE
Global Preserve Integer TESTING_TIME
Global Preserve Integer A_perPass
Global Preserve Integer B_perPass
Global Boolean PassRead
Global String BufBarcode$
Global String BufBarcode_B$
Global String BufBarcode_A_Retest$
Global String BufBarcode_B_Retest$
#define PrintERR(data$) Print Time$, " ERR :: ", data$
#define PrintSta(data$) Print Time$, " :: ", data$
Function main
        Call TowerLightYellow
        Call Initial_IO
        Xqt EStopMonitor
        Xqt ErrorMonitor

        Xqt TCP_Server_Tester
        Wait MemSw(StaTCP1) = On
        Xqt BarcodeReader
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Xqt Auto_StopperBuffer
        Xqt Auto_Lifter
        Xqt Auto_SortterHDD
        Xqt Auto_Feeder
        Wait 1
        Xqt Auto_TableFeed
        Xqt Auto_Run
        Xqt Auto_cover
        Call TowerLightGreen
Fend
```

Function Auto_SortterHDD

```
Function Auto_SortterHDD
Integer iStep
iStep = 0
PrintSta("Start Auto Sorter HDD")
Do
        Select iStep
                Case 0 'Select case to run step
                        If Sw(PH_Sensor2) = On And Sw(RS_Stopper2_up) = On And OnSPBarcode$(1) <> "" Then
                                iStep = 1
                        Elself Sw(PH_Sensor3) = On And Sw(RS_Stopper3_up) = On And OnSPBarcode$(2) <> "" Then
                                iStep = 2
                        Elself Sw(PH_Sensor4) = On And Sw(RS_Stopper4_up) = On And OnSPBarcode$(3) <> "" Then
                                iStep = 3
                        Elself Sw(PH_Sensor5) = On And Sw(RS_Stopper5_up) = On And OnSPBarcode$(4) <> "" Then
                                iStep = 4
                        Elself Sw(PH_Sensor6) = On And Sw(RS_Stopper6_up) = On And OnSPBarcode$(5) <> "" Then
                                iStep = 5
                        Elself Sw(PH_Sensor7) = On Then
                                If OnSPBarcode$(6) <> "" Then
                                        iStep = 6
                                Endlf
                        Endlf
                Case 1
                        If Sw(PH_Sensor1) = Off And Sw(RS_Stopper1_up) = On And OnLifterBarcode$ = "" And
Sw(PH_Sensor2) = On Then
                                Off Stopper2
                                Wait Sw(RS_Stopper2_dn) = On And Sw(PH_Sensor1) = On
                                On Stopper2
                                OnLifterBarcode$ = OnSPBarcode$(1)
                                OnSPBarcode$(1) = ""
                                OnFeed$ = OnFDBarcode$(1)
                                OnFDBarcode$(1) = ""
                                iStep = 0
                        Else
                                iStep = 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    EndIf
            Case 2
                If Sw(PH_Sensor2) = Off And Sw(RS_Stopper2_up) = On And OnSPBarcode$(1) = "" And
Sw(PH_Sensor3) = On Then
                    Off Stopper3
                    Wait Sw(RS_Stopper3_dn) = On And Sw(PH_Sensor2) = On
                    On Stopper3
                    OnSPBarcode$(1) = OnSPBarcode$(2)
                    OnSPBarcode$(2) = ""
                    OnFDBarcode$(1) = OnFDBarcode$(2)
                    OnFDBarcode$(2) = ""
                    iStep = 0
                Else
                    iStep = 3
                EndIf
            Case 3
                If Sw(PH_Sensor3) = Off And Sw(RS_Stopper3_up) = On And OnSPBarcode$(2) = "" And
Sw(PH_Sensor4) = On Then
                    Off Stopper4
                    Wait Sw(RS_Stopper4_dn) = On And Sw(PH_Sensor3) = On
                    On Stopper4
                    OnSPBarcode$(2) = OnSPBarcode$(3)
                    OnSPBarcode$(3) = ""
                    OnFDBarcode$(2) = OnFDBarcode$(3)
                    OnFDBarcode$(3) = ""
                    iStep = 0
                Else
                    iStep = 4
                EndIf
            Case 4
                If Sw(PH_Sensor4) = Off And Sw(RS_Stopper4_up) = On And OnSPBarcode$(3) = "" And
Sw(PH_Sensor5) = On Then
                    Off Stopper5
                    Wait Sw(RS_Stopper5_dn) = On And Sw(PH_Sensor4) = On
                    'Wait 0.3
                    On Stopper5
                    OnSPBarcode$(3) = OnSPBarcode$(4)
                    OnSPBarcode$(4) = ""
                    OnFDBarcode$(3) = OnFDBarcode$(4)
                    OnFDBarcode$(4) = ""
                    iStep = 0
                Else
                    iStep = 5
                EndIf
            Case 5
                If Sw(PH_Sensor5) = Off And Sw(RS_Stopper5_up) = On And OnSPBarcode$(4) = "" And
Sw(PH_Sensor6) = On Then
                    Off Stopper6
                    Wait Sw(RS_Stopper6_dn) = On And Sw(PH_Sensor5) = On
                    'Wait 0.3
                    On Stopper6
                    OnSPBarcode$(4) = OnSPBarcode$(5)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    OnSPBarcode$(5) = ""
                    OnFDBarcode$(4) = OnFDBarcode$(5)
                    OnFDBarcode$(5) = ""
                    iStep = 0
                Else
                    iStep = 6
                EndIf
            Case 6
                If Sw(PH_Sensor6) = Off And Sw(RS_Stopper6_up) = On And OnSPBarcode$(5) = "" And
Sw(PH_Sensor7) = On And OnSPBarcode$(6) <> "" And OnFDBarcode$(6) <> "" Then
                    Off Stopper7
                    'Wait Sw(RS_Stopper7_dn) = On And Sw(PH_Sensor6) = On
                    Wait Sw(PH_Sensor6) = On
                    'Wait 0.3
                    On Stopper7
                    OnSPBarcode$(5) = OnSPBarcode$(6)
                    OnSPBarcode$(6) = ""
                    OnFDBarcode$(5) = OnFDBarcode$(6)
                    OnFDBarcode$(6) = ""
                    iStep = 0
                Else
                    iStep = 0
                EndIf
            Send
        Loop
Fend
```

Function Auto_Lifter

```
Function Auto_Lifter
PrintSta("Start Auto Lifter")
        Do
            If Sw(PH_Lifter) = On And Sw(RS_Lifter_dn) = On And Sw(RS_SPLift_up) = On And OnLifterBarcode$ <> ""
And MemSw(StickFinis) = Off Then
                Wait 0.5
                On Lifter
                Wait Sw(RS_Lifter_up) = On
                'Off StopperLifter
                Wait 0.3
                On PutHDD
                Wait Sw(RS_Push_up) = On, 1
                Wait 1
                On StopperLifter
                MemOn ReadyStick
                Wait MemSw(StickFinis) = On
                Off PutHDD
                Wait Sw(RS_Push_up) = Off, 1
                Off Lifter
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Do
    If Sw(PH_Lifter) = On And Sw(PH_Sensor1) = On Then
        On HDD_OutReady
        Do
            If Sw(HDD_OutRequest) = On Then
                Off StopperLifter
                Off Stopper1
                Exit Do
            EndIf
        Loop
        OnLifterBarcode$ = ""
        OnFeed$ = ""
        Wait Sw(RS_Lifter_dn) = On And Sw(PH_Lifter) = Off And Sw(PH_Sensor1) =
Off And Sw(RS_Stopper1_dn) = On
        Wait Sw(HDD_OutRequest) = Off
        'Wait 1.2
        Off HDD_OutReady
        MemOff TonLabel
        MemOff StickFinis
        On StopperLifter
        On Stopper1
        Exit Do
    EndIf
Loop
ElseIf Sw(PH_Lifter) = On And Sw(RS_Lifter_dn) = On And Sw(RS_SPLift_up) = On And OnLifterBarcode$ <> ""
And MemSw(StickFinis) = On Then
Do
    If Sw(PH_Lifter) = On And Sw(PH_Sensor1) = On Then
        On HDD_OutReady
        Do
            If Sw(HDD_OutRequest) = On Then
                Off StopperLifter
                Off Stopper1
                Exit Do
            EndIf
        Loop
        OnLifterBarcode$ = ""
        OnFeed$ = ""
        Wait Sw(RS_Lifter_dn) = On And Sw(PH_Lifter) = Off And Sw(PH_Sensor1) =
Off And Sw(RS_Stopper1_dn) = On
        Wait Sw(HDD_OutRequest) = Off
        'Wait 1.2
        Off HDD_OutReady
        MemOff TonLabel
        MemOff StickFinis
        On StopperLifter
        On Stopper1
        Exit Do
    EndIf
Loop
EndIf
Loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Fend
```

Function Auto_run

```
Function Auto_run
Integer iStep
        Integer iRetry
        iRetry = 1
        iStep = 0
        PrintSta("Start Auto Run")
        Do
                Select iStep
                        Case 0
                                Home
                                iStep = 1
                        Case 1
                                If MemSw(LabelReady) = On Then
                                        iStep = 2
                                ElseIf MemSw(LabelNoMatch) = On Then
                                        iStep = 3
                                EndIf
                        Case 2
                                PrintProcess("Case 2 Run Step Pickup Label")
                                Retry:
                                On TableFeed
                                Wait Sw(RS_TableFeed_up) = On
                                Print "Position Ship :: _X_ :: ", X_Cal, " _Y_ :: ", Y_Cal, " _U_ :: ", A_Cal
                                Print "Position Offset :: _X_ :: ", CX(PickFeed) + X_Cal, " _Y_ :: ", CY(PickFeed) + Y_Cal, "
_U_ :: ", CZ(PickFeed) + A_Cal
                                Print "HDD : ", CurrentBarcode$, "(", OnLifterBarcode$, ") X=", X_Cal, " Y=", Y_Cal, " U=",
A_Cal
                                Jump PickFeed +X(X_Cal) -Y(Y_Cal) '+U(A_Cal)
                                'Jump PickFeed
                                On Vacuum
                                Wait Sw(SenserVaccuum) = On, 3
                                If TW = True Then
                                        Jump BuferStick CP 'Go Here :Z(0) CP
                                        Off TableFeed
                                        Wait Sw(RS_TableFeed_up) = Off, 2
                                        Off Vacuum
                                        iRetry = iRetry + 1
                                        If iRetry <= 2 Then
                                                PrintProcess("Case 20 Retry Pickup Label :: ")
                                                Jump BuferStick
                                                GoTo Retry
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Else
                        iRetry = 1
                        HDDFail = HDDFail + 1
                        MemOn memERR
                        iStep = 8
                    EndIf
                Else
                    PrintProcess("Case 20 Pickup Label Finis")
                    Jump Stick :Z(0) +X(Offset(0)) +Y(Offset(1)) +Z(Offset(2)) +U(Offset(3)) CP 'Go
Here :Z(0) CP
                    Wait 0.3
                    If Sw(SenserVaccuum) = Off Then
                        Wait 0.5
                    EndIf
                    iStep = 200
                EndIf
            Case 200
                If Sw(SenserVaccuum) = Off Then
                    Wait 0.5
                    Off Vacuum
                    iRetry = iRetry + 1
                    If iRetry <= 2 Then
                        PrintProcess("Case 200 Retry Pickup Label :: ")
                        'GoTo Retry
                        iStep = 2
                    Else
                        iRetry = 1
                        HDDFail = HDDFail + 1
                        MemOn memERR
                        iStep = 8
                    EndIf
                Else
                    CurrentBarcode$ = ""
                    OnPrint$(1) = ""
                    MemOff LabelReady
                    'On TableFeed
                    iStep = 20
                EndIf

            Case 20
                PrintProcess("Case 20 Go point Buffer Stick Label")
                If MemSw(ReadyStick) = Off Then
                    Jump BuferStick CP 'Go BuferStick CP
                    iStep = 21
                Else
                    iStep = 21
                EndIf
            Case 21
                If Sw(RS_Lifter_up) = On And MemSw(ReadyStick) = On And CompareBarcode = True
Then
                    'If CurrentBarcode$ = OnFeed$ Then
                        PrintProcess("Case 21 Run Stick Label")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        Wait Sw(RS_Push_up) = On, 1
                                        Jump Stick +X(Offset(0)) +Y(Offset(1)) +Z(Offset(2)) +U(Offset(3))
                                        Wait 0.5
                                        Off Vacuum
                                        On Blow
                                        Go Here :Z(0) CP
                                        MemOn StickFinis
                                        HDDPass = HDDPass + 1
                                        Off Blow
                                        If MemSw(LabelReady) = Off Then
                                                Jump BuferStick
                                                iStep = 1
                                        Else
                                                iStep = 2
                                        EndIf
                                'Else
                        EndIf
Case 4
                        'MemOn StickFinis
                        HDDFail = HDDFail + 1
                        Off Blow
                        MemOff SkipLabel
                        MemOff NoPickLabel
                        MemOff LabelReady
                        Jump BuferStick
                        iStep = 3
Case 3
                        CurrentBarcode$ = ""
                        OnPrint$(1) = ""
                        MemOff LabelNoMatch
                        Off TableFeed
                        Wait Sw(PH_Lifter) = On
                        MemOn StickFinis
                        iStep = 1
Case 8
                        PrintProcess("Case 8 Error Go Home")
                        Off Vacuum
                        Home
                        Call TowerLightAram
                        MemOn NoPickLabel
                        PrintERR("Can't Pickup Label.")
                        iStep = 80
Case 80
                        iStep = 81
                        ActivePause()
Case 81
                        If MemSw(memERR) = Off Then
                                Wait 1
                                Jump Reject
                                On Blow
                                Wait 2
                                Off Blow
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        Jump BuferStick
                                        Off TableFeed
                                        'MemOff ResultMatch
                                        If MemSw(SkipLabel) = On Then
                                                iStep = 4
                                        Else
                                                iStep = 2
                                        EndIf
                                EndIf
                        Send
                Loop
Fend
```

Function Auto_cover

```
Function Auto_cover
        Integer iStep
        iStep = 0
        PrintSta("Start Auto Stopper Cover")
        On StopperLifter
        Off Stopper_LC1
        On Stopper_LC2
        On Stopper_LC3
        On Stopper_LC4
        On Stopper_LC5
        On Stopper_LC6
        On Stopper_LC7
        Do
                Select iStep
                        Case 0
                                'PrintSta("case 0")
                                If Sw(PH_Lifter) = On And Sw(RS_SPLift_up) = On Then
                                        On Stopper_LC1
                                        iStep = 1
                                Else
                                        iStep = 1
                                EndIf
                        Case 1
                                'PrintSta("case 1")
                                If Sw(PH_LC1) = On And Sw(RS_LC1_up) = On And Sw(PH_Lifter) = Off And
MemSw(StickFinis) = Off Then
                                        On StopperLifter
                                        Wait Sw(RS_SPLift_up) = On
                                        Off Stopper_LC1
                                        Wait Sw(PH_Lifter) = On
                                        Wait 0.3
                                        On Stopper_LC1
                                        iStep = 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ElseIf Sw(PH_LC1) = On And Sw(RS_LC1_up) = On And Sw(PH_Lifter) = On Then
                        On Stopper_LC2
                        iStep = 2
                Else
                        iStep = 2
                EndIf
        Case 2
                'PrintSta("case 2")
                If Sw(PH_LC2) = On And Sw(RS_LC2_up) = On And Sw(PH_LC1) = Off And
MemSw(StickFinis) = Off Then
                        On Stopper_LC1
                        Wait Sw(RS_LC1_up) = On
                        Off Stopper_LC2
                        Wait Sw(PH_LC1) = On
                        On Stopper_LC2
                        iStep = 3
                ElseIf Sw(PH_LC2) = On And Sw(RS_LC2_up) = On And Sw(PH_LC1) = On Then
                        On Stopper_LC3
                        iStep = 3
                Else
                        iStep = 3
                EndIf
        Case 3
                'PrintSta("case 3")
                If Sw(PH_LC3) = On And Sw(RS_LC3_up) = On And Sw(PH_LC2) = Off Then
                        On Stopper_LC2
                        Wait Sw(RS_LC2_up) = On
                        Off Stopper_LC3
                        Wait Sw(PH_LC2) = On
                        On Stopper_LC3
                        iStep = 4
                ElseIf Sw(PH_LC3) = On And Sw(RS_LC3_up) = On And Sw(PH_LC2) = On Then
                        On Stopper_LC4
                        iStep = 4
                Else
                        iStep = 4
                EndIf
        Case 4
                'PrintSta("case 4")
                If Sw(PH_LC4) = On And Sw(RS_LC4_up) = On And Sw(PH_LC3) = Off Then
                        On Stopper_LC3
                        Wait Sw(RS_LC3_up) = On
                        Off Stopper_LC4
                        Wait Sw(PH_LC3) = On
                        On Stopper_LC4
                        iStep = 5
                ElseIf Sw(PH_LC4) = On And Sw(RS_LC4_up) = On And Sw(PH_LC3) = On Then
                        On Stopper_LC5
                        iStep = 5
                Else
                        iStep = 5
                EndIf
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Case 5
                'PrintSta("case 5")
                If Sw(PH_LC5) = On And Sw(RS_LC5_up) = On And Sw(PH_LC4) = Off Then
                    On Stopper_LC4
                    Wait Sw(RS_LC4_up) = On
                    Off Stopper_LC5
                    Wait Sw(PH_LC4) = On
                    On Stopper_LC5
                    iStep = 6
                ElseIf Sw(PH_LC5) = On And Sw(RS_LC5_up) = On And Sw(PH_LC4) = On Then
                    On Stopper_LC6
                    iStep = 6
                Else
                    iStep = 6
                EndIf
            Case 6
                'PrintSta("case 6")
                If Sw(PH_LC6) = On And Sw(RS_LC6_up) = On And Sw(PH_LC5) = Off Then
                    On Stopper_LC5
                    Wait Sw(RS_LC5_up) = On
                    Off Stopper_LC6
                    Wait Sw(PH_LC5) = On
                    On Stopper_LC6
                    iStep = 7
                ElseIf Sw(PH_LC6) = On And Sw(RS_LC6_up) = On And Sw(PH_LC5) = On Then
                    On Stopper_LC7
                    iStep = 7
                Else
                    iStep = 7
                EndIf
            Case 7
                'PrintSta("case 7")
                If Sw(PH_LC7) = On And Sw(RS_LC7_up) = On And Sw(PH_LC6) = Off Then
                    On Stopper_LC6
                    Wait Sw(RS_LC6_up) = On
                    Off Stopper_LC7
                    Wait Sw(PH_LC6) = On
                    On Stopper_LC7
                    iStep = 0
                Else
                    iStep = 0
                EndIf
        Send
    If Sw(PH_LC7) = Off And Sw(RS_LC7_up) = On Then 'And Sw(HDD_InReady) = On Then
        On LC_InRequest
    ElseIf Sw(PH_LC7) = On Then
        Off LC_InRequest
    EndIf
    Loop
Fend
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Auto_StopperBuffer

```
Function Auto_StopperBuffer
	Integer iStep
	iStep = 0
	PrintSta("Start Auto Stopper Buffer")
	Do
		Select iStep
			Case 0
				If Sw(PH_Sensor8) = On And Sw(RS_Stopper8_up) = On Then
					iStep = 1
				EndIf
			Case 1
				If Sw(PH_Sensor7) = Off And Sw(RS_Stopper7_up) = On And OnSPBarcode$(6) = "" And
OnSPBarcode$(7) <> "" And Sw(PH_Sensor8) = On And PassRead = True And MemSw(SentOK) = On Then
					Off Stopper8
					Wait Sw(PH_Sensor8) = Off And Sw(PH_Sensor7) = On
					Wait 0.3
					On Stopper8
					OnSPBarcode$(6) = OnSPBarcode$(7)
					OnSPBarcode$(7) = ""
					PassRead = False
					MemOff SentOK
					iStep = 0
				Else
					iStep = 0
				EndIf
		Send
		If Sw(PH_Sensor8) = Off And Sw(RS_Stopper8_up) = On And Sw(HDD_InReady) = On Then
			On HDD_InRequest
		Elself Sw(PH_Sensor8) = On Then
			Off HDD_InRequest
		EndIf
	Loop
Fend
```

Function Auto_Feeder

```
Function Auto_Feeder
	PrintSta("Start Auto Feeder")
	String txtERR$
	Integer printtime
	Integer iStep
	iStep = 0
	printtime = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Sw(PH_Sensor1) = On And OnLifterBarcode$ = "" Then
        PrintERR("HDD On Stopper 1 no have barcode in program. --> Program Stop")
        Call TowerLightAram
        Quit All
ElseIf Sw(PH_Sensor1) = Off Then
        OnLifterBarcode$ = ""
        OnFeed$ = ""
        OnPrint$(1) = ""
EndIf
If Sw(PH_Sensor2) = On And OnSPBarcode$(1) = "" Then
        PrintERR("HDD On Stopper 2 no have barcode in program. --> Program Stop")
        Call TowerLightAram
        Quit All
ElseIf Sw(PH_Sensor2) = Off Then
        OnSPBarcode$(1) = ""
        OnFDBarcode$(1) = ""
        OnPrint$(2) = ""
EndIf
If Sw(PH_Sensor3) = On And OnSPBarcode$(2) = "" Then
        PrintERR("HDD On Stopper 3 no have barcode in program. --> Program Stop")
        Call TowerLightAram
        Quit All
ElseIf Sw(PH_Sensor3) = Off Then
        OnSPBarcode$(2) = ""
        OnFDBarcode$(2) = ""
        OnPrint$(3) = ""
EndIf
If Sw(PH_Sensor4) = On And OnSPBarcode$(3) = "" Then
        PrintERR("HDD On Stopper 4 no have barcode in program. --> Program Stop")
        Call TowerLightAram
        Quit All
ElseIf Sw(PH_Sensor4) = Off Then
        OnSPBarcode$(3) = ""
        OnFDBarcode$(3) = ""
        OnPrint$(4) = ""
EndIf
If Sw(PH_Sensor5) = On And OnSPBarcode$(4) = "" Then
        PrintERR("HDD On Stopper 5 no have barcode in program. --> Program Stop")
        Call TowerLightAram
        Quit All
ElseIf Sw(PH_Sensor5) = Off Then
        OnSPBarcode$(4) = ""
        OnFDBarcode$(4) = ""
        OnPrint$(5) = ""
EndIf
If Sw(PH_Sensor6) = On And OnSPBarcode$(5) = "" Then
        PrintERR("HDD On Stopper 6 no have barcode in program. --> Program Stop")
        Call TowerLightAram
        Quit All
ElseIf Sw(PH_Sensor6) = Off Then
        OnSPBarcode$(5) = ""
        OnFDBarcode$(5) = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            OnPrint$(6) = ""
        EndIf
        If Sw(PH_Sensor7) = On And OnSPBarcode$(6) = "" Then
            PrintERR("HDD On Stopper 7 no have barcode in program. --> Program Stop")
            Call TowerLightAram
            Quit All
        ElseIf Sw(PH_Sensor7) = Off Then
            OnSPBarcode$(6) = ""
            OnFDBarcode$(6) = ""
            OnPrint$(7) = ""
        EndIf
        If Sw(PH_Sensor8) = On And OnSPBarcode$(7) = "" Then
            'PassRead = False
            iStep = 9
        ElseIf Sw(PH_Sensor8) = Off Then
            'PassRead = False
            OnSPBarcode$(7) = ""
            OnFDBarcode$(7) = ""
        EndIf
        Do
            Select iStep
                Case 0
                    If OnLifterBarcode$ = "" Or OnSPBarcode$(1) = "" Or OnSPBarcode$(2) = "" Or
OnSPBarcode$(3) = "" Or OnSPBarcode$(4) = "" Or OnSPBarcode$(5) = "" Or OnSPBarcode$(6) = "" Or OnSPBarcode$(7) = "" Then
                        iStep = 9
                    Else
                        iStep = 1
                    EndIf
                Case 1
                    iStep = 9
                Case 8
                    If MemSw(RunLastHDD) = On And MemSw(OnPrintLast) = Off Then
                        If printtime < 6 Then
                            Off TableFeed
                            Wait Sw(RS_TableFeed_up) = Off
                            MemOn OnPrintLast
                            Wait MemSw(OnPrintLast) = Off
                            printtime = printtime + 1
                        Else
                            MemOff OnPrintLast
                            Quit Auto_Feeder
                        EndIf
                        iStep = 80
                    EndIf
                Case 80
                    If Sw(PH_Label) = On And Sw(RS_TableFeed_up) = Off Then
                        'MemOff OnPrintLast
                        iStep = 81
                    EndIf
                Case 81
                    If OnPrint$(1) = "" Then
                        OnPrint$(1) = OnPrint$(2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    OnPrint$(2) = OnPrint$(3)
                    OnPrint$(3) = OnPrint$(4)
                    OnPrint$(4) = OnPrint$(5)
                    OnPrint$(5) = OnPrint$(6)
                    OnPrint$(6) = OnPrint$(7)
                    iStep = 82
                EndIf
            Case 82
                If OnPrint$(1) = "" Then
                    iStep = 8
                EndIf
            Case 9    'if On Stppper 3 have Barcode try Read Barcode
                If CheckBarcode = True And Sw(PH_Sensor8) = On And Sw(RS_Stopper8_up) = On And
PassRead = False Then
                    '****Read Barcode HDD****
                    PrintProcess("Read Barcode on HDD")
                    MemOn StartRead_BCR
                    Wait MemSw(StartRead_BCR) = Off
                    PassRead = True
                    HDDIncomming = HDDIncomming + 1
                    iStep = 90
                Else
                    iStep = 90
                EndIf
            Case 90
                If CheckFeedderBarcode = True And OnSPBarcode$(7) <> "" And OnSPBarcode$(7) <>
BufBarcode$ And MemSw(SentOK) = Off Then
                    '****Send Barcode to interface software****
                    'CmdBarcode$ = Barcode$
                    PrintProcess("Send Barcode HDD To Interface")
                    CmdBarcode$ = OnSPBarcode$(7)
                    'BufBarcode$ = CmdBarcode$
                    MemOn SendBarcode1
                    Wait MemSw(SendBarcode1) = Off, 3
                    If TW = On And CmdBarcode$ = "" Then
                        MemOff SendBarcode1
                        iStep = 9
                    Else
                        MemOn SentOK
                        iStep = 901
                    EndIf
                Else
                    iStep = 9
                EndIf
            Case 901 'Wait barcode from MIS
                If ResultBarcode$ <> "" And MemSw(MISERR) = Off Then
                    PrintProcess("Recieve Result Barcode From Interface")
                    iStep = 911
                ElseIf ResultBarcode$ <> "" And MemSw(MISERR) = On Then
                    MemOn OnPrintLast
                    Wait MemSw(OnPrintLast) = Off
                    MemOn memERR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ActivePause()
                        iStep = 920
                    EndIf
                Case 911
                    If OnPrint$(1) = "" Then
                        OnPrint$(1) = OnPrint$(2)
                        iStep = 910
                    EndIf
                Case 910
                    If OnFDBarcode$(6) = "" Then
                        'OnPrint$(1) = OnPrint$(2)
                        OnPrint$(2) = OnPrint$(3)
                        OnPrint$(3) = OnPrint$(4)
                        OnPrint$(4) = OnPrint$(5)
                        OnPrint$(5) = OnPrint$(6)
                        OnPrint$(6) = OnPrint$(7)
                        OnPrint$(7) = ResultBarcode$
                        OnFDBarcode$(6) = OnPrint$(7)
                        ResultBarcode$ = ""
                        iStep = 9
                    EndIf
                Case 920
                    If MemSw(MISERR) = Off Then
                        'PassRead = False
                        Call TowerLightGreen
                        MemOff memERR
                        iStep = 911
                    EndIf

            Send
            If Sw(PH_Lifter) = On And Sw(PH_Sensor1) = On And Sw(PH_Sensor2) = On And Sw(PH_Sensor3) = On And
Sw(PH_Sensor4) = On And Sw(PH_Sensor5) = On And Sw(PH_Sensor6) = On And Sw(PH_Sensor7) = Off And MemSw(RunLastHDD)
= On And (TaskState(Auto_StopperBuffer) = 1 Or TaskState(Auto_StopperBuffer) = 2) Then
                Quit Auto_StopperBuffer
                On Stopper8
                'Off CV_Buffer
                iStep = 8
            EndIf
        Loop
Fend
```

Function Auto_TableFeed

```
Function Auto_TableFeed
        PrintSta("Start Auto Table Feeder")
        Integer iStep
        iStep = 2
        Do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Select iStep
        Case 2
            'If Sw(PH_Label) = On And OnPrint$(1) <> "" And OnPrint$(2) <> "" And OnPrint$(3) <>
"" And OnPrint$(4) <> "" Then
            If Sw(PH_Label) = On And OnPrint$(1) <> "" Then
                iStep = 220
            Else
                iStep = 2
            EndIf
        Case 220  'check status label
            PrintProcess("On Vision Check Label")
            MemOn VisionON
            Wait MemSw(VisionON) = Off
            iStep = 221
        Case 221
            If MemSw(ResultMatch) = On And Sw(PH_Label) = On And MemSw(TonLabel) = Off
Then
                PrintProcess("Label On Feeder OK")
                MemOff ResultMatch
                On TableFeed
                Wait Sw(RS_TableFeed_up) = On, 6
                If TW = True Then
        PrintERR("Can't up table feeder, Check Solinoid table feeder Or reed switch(UP)")
    Call TowerLightAram
        MemOn memERR
                EndIf
                iStep = 20
            EndIf
        Case 20
            'On TableFeed
            'Wait Sw(RS_TableFeed_up) = On, 6
            'If TW = True Then
'   PrintERR("Can't up table feeder, Check Solinoid table feeder Or reed switch(UP)")
'    Call TowerLightAram
'   MemOn memERR
            'EndIf

            'Wait 0.2
            PrintProcess("On Vision Check Barcode")
            MemOn VisionONAll
            Wait MemSw(VisionONAll)
            iStep = 21
        Case 21
            If MemSw(ResultMatch) = On Then
                PrintProcess("Barcode On Feeder OK")
                MemOff ResultMatch
                'Off TableFeed
                iStep = 200
            EndIf
        Case 200
            If CurrentBarcode$ <> "" Then
                If OnPrint$(1) = CurrentBarcode$ Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        iStep = 201
                                    Else
                                        Call TowerLightAram
                                        PrintERR("Barcode HDD and Label is not match")
                                        Print Time$, " ERR :: ", "Barcode HDD and Label is not match
(HDD: ", OnLifterBarcode$, ", Label HDD: ", OnPrint$(1), ", Label Vision Read: ", CurrentBarcode$, ")"
                                        iStep = 3
                                    EndIf
                                EndIf
                            Case 201 'On Label Ready
                                PrintProcess("Label On Feeder Ready")
                                MemOn LabelReady
                                Wait MemSw(LabelReady) = Off
                                Off TableFeed
                                iStep = 2
                            Case 3
                                MemOn LabelNoMatch
                                iStep = 300
                            Case 300
                                'If OnPrint$(2) = OnSPBarcode$(1) And OnPrint$(3) = OnSPBarcode$(2) Then
                                    Call TowerLightGreen
                                    PrintProcess("Run next label.")
                                    PrintERR("....")
                                    Wait MemSw(LabelNoMatch) = Off
                                    Off TableFeed 'New Edit
                                    Wait Sw(RS_TableFeed_up) = Off, 1
                                    Wait 0.3
                                    iStep = 2
                                'Else
                                    'PrintERR("Barcode HDD and Label is not match (On conveyer and feeder)")
                                    'iStep = 8
                                'EndIf
                        Send
                Loop
Fend
```

Function BarcodeReader

```
Function BarcodeReader
        String Action$
        String ClearBufData$
        Integer numLine
        Boolean try
        Integer CountErrRead
        Action$ = "LON"
        try = False
        CountErrRead = 0
        Barcode$ = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OnErr GoTo eHandler
GoTo RetryRead
eHandler:
        Call TowerLightAram
        PrintERR("Barcode ERROR")
        Print ErrMsg$(Err)
        EResume
RetryRead:
        MemOff ErrBarcode
        If ChkCom(1) = -3 Then
                OpenCom #1
                Print Time$, " :: ", "Open comport Barcode"
        EndIf
        Do
                If MemSw(StartRead_BCR) = On Then
                        '========== Send Command ==================
                        Print #1, Action$
                        Print Time$, " :: ", "RS-232 Send Read Barcode :: ", Action$
                        '==========================================
                        Xqt TimeOut_ReadBarcode(2)
                        Do
                                numLine = Lof(1)
                                If numLine > 0 Then
                                        Line Input #1, Barcode$
                                        Print Time$, " :: ", "RS-232 Recived Barcode :: ", Barcode$
                                        OnSPBarcode$(7) = Barcode$
                                        MemOn MReadBarcode
                                        MemOff StartRead_BCR
                                        Quit TimeOut_ReadBarcode
                                        Exit Do
                                EndIf
                                If MemSw(ErrBarcode1) = On And try = False Then
                                        If CountErrRead >= 3 Then
                                                PrintERR("Can't Read Barcode 1")
                                                Call TowerLightAram
                                                try = True
                                        ElseIf try = False Then
                                                CountErrRead = CountErrRead + 1
                                                GoTo RetryRead
                                        EndIf
                                EndIf
                        Loop
                EndIf
                numLine = Lof(1)
                If numLine > 0 Then
                        Line Input #1, ClearBufData$
                        Print Time$, " :: ", "TCP Barcode A Recived data :: ", ClearBufData$
                EndIf
        Loop
Fend
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	สิทธิชัย เสียงสนั่น
วัน เดือน ปีเกิด	8 พฤศจิกายน 2537
ที่อยู่	23/177 สมุทรนิเวศน์ ซอย10 หมู่2 ตำบลบางเมือง อำเภอเมือง จังหวัดสมุทรปราการ 10270
E-mail	supervirusG@Gmail.com
โทรศัพท์	063-989-8718
ประวัติการศึกษา	2556 – ปัจจุบัน กำลังศึกษาหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบการผลิต วิทยาลัยนวัตกรรมการผลิตขั้นสูง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประวัติการทำงานวิจัย	นักศึกษาฝึกงานภาคฤดูร้อน / สหกิจศึกษา แผนก Research and development บริษัท แคล-คอมพ์ อิเล็กทรอนิกส์ (ประเทศไทย) จำกัด มหาชน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้