

การออกแบบฟังก์ชันการทำงานและการเขียนโปรแกรมสำหรับชุดสาธิตการบรรจุยาที่ใช้
หุ่นยนต์สกราราและวิชันเซนเซอร์ของมิตซูบิชิ อีเล็คทริก
Function Design and Programming for Medicine Packing Demonstration Kit
Using Mitsubishi Electric SCARA Robot and Vision Sensor



นายgunที คำประเสริฐ
นายชนกันต์ ขวาลธาดา
นายธงชัย พรหมรักษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบฟังก์ชันการทำงานและการเขียนโปรแกรมสำหรับชุดสาธิตการบรรจุยาที่ใช้
หุ่นยนต์สกราราและวิชันเซนเซอร์ของมิตซูบิชิ อีเล็คทริก
Function Design and Programming for Medicine Packing Demonstration Kit
Using Mitsubishi Electric SCARA Robot and Vision Sensor



นายgunที คำประเสริฐ
นายชนกันต์ ขวาลธาดา
นายธงชัย พรหมรักษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTION DESIGN AND PROGRAMING FOR MEDICINE PACKING
DEMONSTRATION KIT USING MITSUBISHI ELECTRIC SCARA ROBOT
AND VISION SENSOR



KUNNATEE KHAMPRASOET
CHANAKAN CHVAVLATADA
THONGCHAI POMRAK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN AUTOMATION ENGINEERING
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิศวกรรมอัตโนมัติ
ภาควิชาวิศวกรรมการวัดและควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การออกแบบฟังก์ชันการทำงานและการเขียนโปรแกรมสำหรับชุดสาธิต
การบรรจุยาที่ใช้หุ่นยนต์สการาและวิชันเซนเซอร์ของมิตซูบิชิ อิเล็กทริก

Thesis title Function Design and Programming for Medicine Packing
Demonstration Kit Using Mitsubishi Electric SCARA Robot
and Vision Sensor

นักศึกษา นายกุนที คำประเสริฐ รหัสประจำตัว 62010071
นายชนกันต์ ขวาลธาดา รหัสประจำตัว 62010149
นายธงชัย พรหมรักษ์ รหัสประจำตัว 62010330

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมอัตโนมัติ

อาจารย์ที่ปรึกษา รศ.ดร. อีร์วัฒน์ เทพมณี

อาจารย์ที่ปรึกษาร่วม รศ.ดร. ไสว พงศ์สวัสดิ์
รศ.ดร. ประภาช อุคคกิม่าพันธุ์

ปีการศึกษา 2565

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.ดร. อีร์วัฒน์ เทพมณี	<i>T.Thepmanee</i>
รศ.ดร. ไสว พงศ์สวัสดิ์	<i>Sawai Pongswatd</i>
รศ.ดร. ประภาช อุคคกิม่าพันธุ์	<i>Prapach Ukukkima</i>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การออกแบบฟังก์ชันการทำงานและการเขียนโปรแกรมสำหรับชุดสาธิตการบรรจุยาที่ใช้หุ่นยนต์สการาและวิชันเซนเซอร์ของมิตซูบิชิ อีเล็กทริก	
นักศึกษา	นายกุนที คำประเสริฐ	รหัสประจำตัว 62010071
	นายชนกันต์ ขวาลธาดา	รหัสประจำตัว 62010149
	นายธงชัย พรหมรักษ์	รหัสประจำตัว 62010330
หลักสูตร	วิศวกรรมอัตโนมัติ	
คณะ	วิศวกรรมศาสตร์	
อาจารย์ที่ปรึกษา	รศ.ดร. อธิวัฒน์ เทพมณี	
อาจารย์ที่ปรึกษาร่วม	รศ.ดร. ไสว พงศ์สวัสดิ์	
	รศ.ดร. ประภาช อุคคกิมพาน์	

บทคัดย่อ

ปริญญานิพนธ์นำเสนอเทคนิคทางวิศวกรรมในการออกแบบฟังก์ชันการทำงานและเขียนโปรแกรมสำหรับชุดการสาธิตการบรรจุยาเพื่อสาธิตฟังก์ชันการทำงานของผลิตภัณฑ์มิตซูบิชิ อีเล็กทริก ซึ่งได้แก่ หุ่นยนต์สการา (Selective Compliance Assembly Robot Arm: SCARA) รุ่น RH-6FH5520 พีแอลซี (Programmable Logic Controller: PLC) รุ่น R08CPU เอชเอ็มไอ (Human Machine Interface: HMI) รุ่น GT2708-STBA อินเวอร์เตอร์ (Inverter) รุ่น FR-E800 วิชันเซนเซอร์ (Vision Sensor) รุ่น VS80C-402R และโค้ดรีดเดอร์ (Code Reader) รุ่น CF26 ผู้ใช้งานสามารถเลือกชนิดยาและจำนวนเม็ดที่ต้องการสำหรับแต่ละคำสั่งในการบรรจุยา โดยมีชนิดยาให้เลือกได้จำนวน 6 ชนิด และจำนวนสูงสุดของยาแต่ละชนิดที่เลือกได้คือ 5 เม็ด นอกจากนี้ มีการสร้างเว็บแอปพลิเคชัน (Web Application) สำหรับระบบสั่งซื้อด้วยคิวอาร์โค้ด (QR Code) โดยใช้ Firebase และมีการส่งเอาต์พุตของโค้ดรีดเดอร์ไปยังพีแอลซีซึ่งติดตั้งเพื่อควบคุมสองสายพานลำเลียงที่ทำงานด้วยอินเวอร์เตอร์ โดยสายพานเส้นแรกใช้สำหรับการลำเลียงเม็ดยาที่ต้องการซึ่งถูกปล่อยจากเครื่องจ่ายยา ในขณะที่สายพานเส้นที่สองใช้สำหรับการลำเลียงกล่องเปล่าไปยังตำแหน่งที่กำหนด หุ่นยนต์สามารถหยิบยาที่ต้องการออกจากสายพานและวางลงในกล่องได้โดยอ้างอิงข้อมูลตำแหน่งซึ่งเป็นเอาต์พุตของวิชันเซนเซอร์ หลังจากการคัดแยกยาเสร็จสิ้น หุ่นยนต์จะปิดกล่องยาเพื่อสิ้นสุดการบรรจุยา ยิ่งไปกว่านั้น มีการแสดงสถานะการทำงานและข้อความข้อผิดพลาดของหุ่นยนต์พีแอลซี และอินเวอร์เตอร์บนหน้าจอเอชเอ็มไอที่สร้างขึ้น ผลการทดสอบยืนยันได้ว่า ชุดการสาธิตสามารถทำงานตามฟังก์ชันที่ได้ออกแบบ

คำสำคัญ : โค้ดรีดเดอร์, ชุดสาธิต, เอชเอ็มไอ, พีแอลซี, หุ่นยนต์สการา, การคัดแยก, วิชันเซนเซอร์, วิชันแทรกคิง, เว็บแอปพลิเคชัน

Thesis Title: Function Design and Programming for Medicine Packing
Demonstration Kit Using Mitsubishi Electric SCARA Robot
and Vision Sensor

Students: Mr. Kunnatee Khamprasoet **Student ID:** 62010071
Mr. Chanakan Chvavlatada **Student ID:** 62010149
Mr. Thongchai Pomrak **Student ID:** 62010330

Program: Automation Engineering

School Of: Engineering

Thesis Advisor: Assoc. Prof. Dr. Teerawat Ngamchuchit

Thesis Co Advisors: Assoc. Prof. Dr. Sawai Pongswatd
Assoc. Prof. Dr. Prapart Ukakipaparn

ABSTRACT

In order to demonstrate functionalities of Mitsubishi Electric products including selective compliance assembly robot arm (SCARA) modeled CR751-D, programmable logic controller (PLC) modeled R08CPU, human machine interface (HMI) modeled GT2708-STBA, inverter modeled FR-E800, vision sensor modeled VS80C-402R and code reader modeled GT2708-STBA, engineering techniques of function design and programming for demonstration kit of medicine packing are proposed in this thesis. For each packing order, there are 6 different types of medicine tablets available for a user to choose, and the maximum tablet number of each medicine type is set to 5. In addition, the web application for QR code ordering system is created by using Firebase, and the code reader output is applied to the PLC, which is installed to control two conveyor belts running with the inverters. The first conveyor belt is used to convey the desired medicine tablets discharged from a dispenser, while the latter is used to convey the empty box to the specified position. Based on the vision sensor's position data output, the robot can pick the desired medicine tablets on the belt and place them into the box. After the medicine sorting is finished, the medicine box will be covered by the robot to finish the packing. Moreover, the operating statuses and error messages of the robot, PLC, and inverters are displayed on the created HMI screens. Test results confirm that the demonstration kit can operate in according to the designed functions.

Keywords: Code reader, Demonstration kit, HMI, PLC, SCARA robot, Sorting, Vision sensor, Vision Tracking, Web application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและตั้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตกรรมประกาศ

รายงานปริญญานิพนธ์นี้สำเร็จลุล่วงด้วยดีเนื่องจากความกรุณาของ รศ.ดร. ไสว พงศ์สวัสดิ์ รศ.ดร. อธิวัฒน์ เทพมณี และรศ.ดร. ประภาส อุคคิกิมาพันธ์ ที่ได้ให้คำแนะนำแนวองค์ความรู้เกี่ยวกับการสร้างชุดสาธิตแนะนำการแก้ไขปัญหาทางเทคนิค และให้โอกาสคณะผู้จัดทำทำปริญญานิพนธ์ฉบับนี้ และร่วมงานกับทาง Mitsubishi Electric Factory Automation Thailand (MELFT) รวมไปถึงการติดต่อและช่วยอำนวยความสะดวกในการจัดทำปริญญานิพนธ์จนสำเร็จลุล่วงตามเป้าหมาย

ขอขอบคุณบริษัท Mitsubishi Electric Factory Automation Thailand และผู้สนับสนุนทุกท่าน สำหรับการให้โอกาสคณะผู้จัดทำได้ร่วมทำปริญญานิพนธ์นี้ร่วมกัน รวมไปถึงการสนับสนุนทางด้านอุปกรณ์งบประมาณ และการให้ความรู้ทางเทคนิค รวมไปถึงทักษะทั่วไปเปิดโอกาสให้ซักถาม ตลอดจนการอำนวยความสะดวกให้ทางคณะผู้จัดทำสามารถทำปริญญานิพนธ์จนสำเร็จลุล่วงได้

ขอขอบคุณคุณภาวิ สมศักดิ์ และคุณสุรรัตน์ ผดุงศักดิ์วิริยะ ที่ได้ช่วยชี้แนะแนวทางการสร้างชุดสาธิตและช่วยจัดหาบริษัทสร้างตัวฮาร์ดแวร์ของชุดสาธิตที่ช่วยเพิ่มความสะดวกในทางด้านการเดินทาง และการทำงานของคณะผู้จัดทำอีกทั้งขอขอบคุณคุณวิศรุต จันทริโอ และคุณนาวิน คุ่มสุวรรณ สำหรับการให้คำชี้แนะแนวทางการแก้ปัญหาเฉพาะทางและการช่วยแนะแนวทางการเขียนโปรแกรมตัวชุดสาธิตเพื่อเพิ่มความรวดเร็วและประสิทธิภาพในการทำงานและช่วยอำนวยความสะดวกทางด้านการเดินทางของคณะผู้จัดทำ

ขอขอบคุณครอบครัวและเพื่อน ๆ ทุกคนที่คอยให้การสนับสนุนตลอดมาในด้านต่าง ๆ สุดท้ายนี้ คณะผู้จัดทำหวังว่าปริญญานิพนธ์ฉบับนี้คงเป็นประโยชน์สำหรับผู้อ่านและทางหน่วยงานที่เกี่ยวข้องและผู้ที่เกี่ยวข้องศึกษาต่อไป หากมีข้อผิดพลาดประการใดก็ขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

สารบัญ

	หน้า
กิตกรรมประกาศ.....	iii
สารบัญ.....	iv
สารบัญตาราง.....	viii
สารบัญรูป.....	ix
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของงาน.....	2
1.4 ขั้นตอนการดำเนินงาน	4
1.5 ผลที่คาดว่าจะได้รับ	5
บทที่ 2 แนวคิดและหลักการที่เกี่ยวข้อง.....	6
2.1 กล่าวนำ	6
2.2 หน่วยงานสภาราของมิตซูบิชิ อิเล็กทริก.....	6
2.3 พีแอลซีของมิตซูบิชิ อิเล็กทริก	8
2.4 อินเวอร์เตอร์ของมิตซูบิชิ อิเล็กทริก	10
2.5 ฟังก์ชันวิชันแทรกคิงสำหรับหุ่นยนต์สภาราของมิตซูบิชิ อิเล็กทริก	11
2.5.1 วิชันเซนเซอร์ของมิตซูบิชิ อิเล็กทริก	12
2.5.2 เอ็นโค้ดเดอร์แบบหมุน.....	13
2.6 โพรโตคอลที่เกี่ยวข้อง.....	14
2.6.1 อีเทอร์เน็ต	14
2.6.2 พื้นฐาน CC-Link IEF	14
2.6.3 CC-Link	15
2.7 โค้ดรีดเดอร์.....	16
2.8 เว็บแอปพลิเคชัน.....	17
2.8.1 ยูอาร์แอล	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและแจ้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.8.2 เฟรมเวิร์ก	18
2.9 แพลตฟอร์มและเครื่องมือพัฒนาเว็บแอปพลิเคชัน	19
2.9.1 Visual Studio Code	19
2.9.2 Codesandbox.io	20
2.9.3 Google Developer Tool	20
2.10 เว็บโฮสติ้ง	21
บทที่ 3 การออกแบบฟังก์ชันการทำงานและการเขียนโปรแกรมสำหรับชุดสาธิต.....	22
3.1 กล่าวนำ	22
3.2 แนวคิดการออกแบบชุดสาธิต.....	22
3.3 ส่วนประกอบของชุดสาธิต.....	26
3.3.1 โครงสร้างเครื่องชุดสาธิตโดยรวม	26
3.3.2 กลไกของการปล่อยยาและกล่องบรรจุยา.....	27
3.3.3 อุปกรณ์ควบคุม.....	28
3.3.4 การเดินสายไฟฟ้า	29
3.4 การตั้งค่าอุปกรณ์สำหรับการเชื่อมต่ออุปกรณ์เข้าด้วยกัน	30
3.4.1 การส่งข้อมูลระหว่างพีแอลซีกับอุปกรณ์อื่น	30
3.4.2 การเชื่อมต่อหุ่นยนต์กับอุปกรณ์อื่น	38
3.5 การสร้างเว็บแอปพลิเคชัน.....	40
3.5.1 การเลือกเฟรมเวิร์กและแผนการดำเนินงาน.....	41
3.5.2 Component และการทำงานของ Component ในแต่ละส่วนของเว็บ	42
3.5.3 การใช้งานและติดตั้งส่วนเสริมใน “Project”	52
3.5.4 การ Deploy โปรแกรมขึ้น Firebase โฮสติ้งด้วย Visual Studio Code	53
3.5.5 หลักการสร้างคิวอาร์โค้ดเพื่อแลกเปลี่ยนข้อมูลกับชุดสาธิต	55
3.6 การตั้งค่าการทำงานของโค้ดรีดเตอร์	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและทำซ้ำอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.7 การเขียนโปรแกรมพีแอลซี.....	59
3.7.1 การควบคุมหลักในการเปิดและปิดหรือเปลี่ยนโหมดการทำงานของชุดสาธิต	59
3.7.2 การควบคุมโซลินอยด์วาล์วเพื่อควบคุมกระบอกสูบในโลกใต้อ่างน้ำ 66	66
3.7.3 การควบคุมการทำงานของสายพาน.....	68
3.7.4 การควบคุมไค์ดรีเตอร์ CF26	69
3.7.5 การควบคุมคิกและการปล่อยยา.....	71
3.7.6 การแสดงผลแจ้งเตือน.....	72
3.8 การเขียนโปรแกรมหน้าจอเอชเอ็มไอ	74
3.8.1 การใช้หน้าจอ GOT ในการควบคุมชุดสาธิต.....	78
3.8.2 การใช้หน้าจอ GOT ในการทดสอบการทำงานต่าง ๆ.....	82
3.8.3 การใช้หน้าจอ GOT ในการระบุข้อผิดพลาดและการแจ้งเตือนของชุดสาธิตได้	85
3.8.4 การใช้หน้าจอ GOT ในการแสดงข้อมูลเกี่ยวกับตัวเครื่องและแสดงค่า OEE	89
3.9 การเขียนโปรแกรมวิชันเซนเซอร์.....	91
3.9.1 การตั้งค่าก่อนใช้งานโปรแกรม	92
3.9.2 การเขียนโปรแกรมวิชันเซนเซอร์ให้สามารถคัดแยกยาได้	94
3.9.3 การเขียนโปรแกรมวิชันเซนเซอร์ให้ส่งข้อมูลเข้าหุ่นยนต์ได้	95
3.10 การเขียนโปรแกรมควบคุมหุ่นยนต์	97
3.10.1 การเขียนโปรแกรมหยิบยาใส่กล่อง.....	97
3.10.2 การเขียนโปรแกรมวิชันแทรกคิง.....	101
บทที่ 4 การทดสอบฟังก์ชันการทำงานของชุดสาธิต.....	116
4.1 กล่าวนำ	116
4.2 การทดสอบการใช้งานเว็บแอปพลิเคชัน	116
4.3 การทดสอบฟังก์ชันสแกนคิวอาร์โค้ด	120
4.4 ความแม่นยำในการหยิบยา.....	120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ [vi](#) อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.5 การทำตามคำสั่งซื้อ.....	122
บทที่ 5 สรุปผล.....	124
5.1 สรุปผลการดำเนินงาน.....	124
5.2 ปัญหาและอุปสรรคในการดำเนินงาน.....	124
5.3 แนวทางการแก้ไขปัญหา	124
เอกสารอ้างอิง.....	125



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1.1 รูปร่างและชื่อของยาทั้ง 6 ชนิด.....	3
1.2 แผนการดำเนินงาน.....	5
3.1 ข้อดีและข้อเสียของการใช้วิชันเซนเซอร์.....	23
3.2 Address การส่งข้อมูลแบบบิตระหว่างหุ่นยนต์และพีแอลซี.....	31
3.3 Address การส่งข้อมูลประเภทตัวเลขระหว่างหุ่นยนต์และพีแอลซี.....	33
3.4 สัญญาณ X1000 และ Y1000 ที่พีแอลซีใช้ในการควบคุมอินเวอร์เตอร์และไค์ดรีดเตอร์.....	36
3.5 สัญญาณ W1000 และ W1100 ที่พีแอลซีใช้ในการควบคุมอินเวอร์เตอร์และไค์ดรีดเตอร์.....	37
3.6 การแจ้งเตือนที่แสดงทั้งหมด.....	87
4.1 ความสำเร็จในการหีบยา.....	120
4.2 ผลการทดสอบคำสั่งชื่อ 2-5 เม็ดต่อชนิด.....	122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ ^{viii}อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 หุ่นยนต์สการาของมิตซูบิชิ อีเล็กทริก.....	6
2.2 พื้นที่การทำงานของหุ่นยนต์สการาของมิตซูบิชิ อีเล็กทริก	6
2.3 จลนศาสตร์ของหุ่นยนต์สการา	7
2.4 จลนศาสตร์ผกผันของหุ่นยนต์สการา.....	8
2.5 ซีพียูของพีแอลซีของมิตซูบิชิ อีเล็กทริก.....	9
2.6 หน่วยฐานของพีแอลซีของมิตซูบิชิ อีเล็กทริก	9
2.7 แหล่งจ่ายไฟพีแอลซีของมิตซูบิชิ อีเล็กทริก.....	9
2.8 โมดูลอินพุต/เอาต์พุตดิจิทัลและโมดูลอินพุต/เอาต์พุตแอนะล็อกของมิตซูบิชิ อีเล็กทริก	10
2.9 การแปลงไฟฟ้าเพื่อเปลี่ยนความถี่ไฟฟ้าและควบคุมมอเตอร์	10
2.10 อินเวอร์เตอร์ของมิตซูบิชิ อีเล็กทริก รุ่น FR-E800.....	11
2.11 ตัวอย่างการบรรจุอาหารแปรรูป	11
2.12 ตัวอย่างการเรียงชิ้นส่วน	12
2.13 ตัวอย่างการประกอบอุปกรณ์ไฟฟ้าขนาดเล็ก	12
2.14 ค่าขอบเขตการมองเห็นวิชันของหุ่นยนต์.....	13
2.15 ส่วนประกอบของเอ็นโค้ดเดอร์แบบหมุน.....	13
2.16 ตัวอย่างสัญญาณพัลส์ของเอ็นโค้ดเดอร์แบบหมุน.....	13
2.17 การเทียบเทคโนโลยีสื่อสารอีเทอร์เน็ตกับ OSI model	14
2.18 ตัวอย่างโครงสร้างระบบพื้นฐาน CC-Link IE Field เชื่อมต่อกับอุปกรณ์ต่าง ๆ.....	15
2.19 การเชื่อมต่อหุ่นยนต์จำนวนมากด้วย CC-Link	16
2.20 Code scanner ของบริษัท MELFT	16
2.21 ส่วนประกอบของคิวอาร์โค้ด	17
2.22 รูปแบบยูอาร์แอล	18
2.23 ตัวอย่างโปรแกรมใน index.html.....	19
2.24 Visual Studio Code.....	20
2.25 ตัวอย่างการใช้งาน Codesandbox.io.....	20
2.26 ตัวอย่างการใช้งาน Google Developer Tool	21
3.1 การทำงานของแนวคิดเริ่มแรกของชุดสาธิต	22
3.2 โครงสร้างโดยรวมของระบบที่ใช้วิชันเซนเซอร์	23
3.3 การทำงานของชุดสาธิตที่ติดตั้งวิชันเซนเซอร์ไว้ที่ปลายแขนหุ่นยนต์.....	24
3.4 โครงสร้างโดยรวมของระบบที่ใช้เอ็นโค้ดเดอร์ร่วมกับวิชันเซนเซอร์.....	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ ix อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.5 การทำงานของชุดสาธิตที่ใช้เอ็นโค้ดเดอร์ร่วมกับวิชชันเซนเซอร์.....	25
3.6 ภาพรวมการทำงานของชุดสาธิต.....	25
3.7 ต้นแบบและขนาดของชุดสาธิต.....	26
3.8 ตัวปล่อยยาและปล่อยกล่อง.....	27
3.9 กลไกปล่อยยาจากด้านบน.....	28
3.10 รูปด้านหลังของกลไกปล่อยกล่อง.....	28
3.11 แผงควบคุมของชุดสาธิต.....	29
3.12 แบบการจ่ายไฟอย่างง่าย.....	30
3.13 การตั้งค่า Communication Mode ของ CC-Link.....	30
3.14 การตั้งค่า CC-Link Network Configurator.....	31
3.15 การตั้งค่า Link Refresh Settings ของ CC-Link.....	31
3.16 การตั้งค่า Module Parameter เพื่อใช้งานพื้นฐาน CC-Link IEF.....	33
3.17 การตั้งค่า Network Configuration Setting ของพื้นฐาน CC-Link IEF.....	34
3.18 การตั้งค่า Link Refresh Settings ของพื้นฐาน CC-Link IEF.....	34
3.19 สัญญาณที่ใช้ในการควบคุมอินเวอร์เตอร์ด้วยพีแอลซีจากคู่มือ.....	35
3.20 สัญญาณที่ใช้ในการควบคุมไค์ดรีดเดอร์ด้วยพีแอลซีจากคู่มือ.....	36
3.21 การตั้งค่า Parameter CC-Link ของหุ่นยนต์ในซอฟต์แวร์ RT ToolBox3.....	38
3.22 หน้าการตั้งค่า Parameter หุ่นยนต์เพื่อเชื่อมต่อวิชชันเซนเซอร์.....	39
3.23 การตั้งค่าของวิชชันเซนเซอร์เพื่อเชื่อมต่อหุ่นยนต์.....	39
3.24 ผังงานการทำงานของการสร้างคิวอาร์โค้ดจากเว็บแอปพลิเคชัน.....	41
3.25 ไฟล์ทั้งหมดใน “Project”.....	43
3.26 Component ของแต่ละหน้า.....	44
3.27 โปรแกรมในไฟล์ index.js.....	45
3.28 โปรแกรมในไฟล์ App.js ส่วนจัดการหน้าจอเพื่อไปยังหน้าต่าง ๆ.....	45
3.29 แนวคิดการรับและส่งข้อมูลของตะกร้าสินค้า.....	46
3.30 ส่วน Appheader ในมือถือ.....	46
3.31 โปรแกรมส่วนกำหนดเงื่อนไขการใส่ชื่อ.....	47
3.32 ส่วนการรับส่งข้อมูลของ Cartdata และ InCartdata.....	47
3.33 โปรแกรมในไฟล์ Ad_slider_bar.js.....	48
3.34 การทำงานโปรแกรมการเพิ่มตัวยา.....	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและข้อมูลอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.35 ตัวอย่างโปรแกรมรูปแบบคำถาม.....	50
3.36 ตัวอย่างโปรแกรมการนำทางของคำถาม	51
3.37 โปรแกรมส่วน AppFooter.....	51
3.38 การแสดงผลส่วน AppFooter	51
3.39 ไฟล์ Packet.json	52
3.40 หน้าการสร้าง “Firebase Project”	53
3.41 การตั้งค่าเพิ่มฟังก์ชัน Firebase ในคอมใน Visual Studio Code.....	54
3.42 การตั้งค่าของ Hosting	54
3.43 การวิเคราะห์ข้อมูลบน Firebase.....	54
3.44 การ Deploy ขึ้นหน้าเว็บ.....	55
3.45 รูปแบบการส่งข้อมูลตัวยาให้โค้ดรีดเดอร์ CF26	55
3.46 ไฟล์การสร้างคิวอาร์โค้ด.....	56
3.47 หน้าซอฟต์แวร์การเลือกอุปกรณ์เชื่อมต่อ	56
3.48 ฟังก์ชันการถ่ายทอดสดของตัวโค้ดรีดเดอร์.....	57
3.49 รูปแบบค่าที่รับจากคิวอาร์โค้ด	58
3.50 ที่อยู่ IP การเชื่อมต่อ GOT.....	58
3.51 โปรแกรมโหมดออโต้และโหมดแมนนวล	59
3.52 โปรแกรมสั่งหยุดการทำงานหุ่นยนต์ฉุกเฉิน.....	59
3.53 เงื่อนไขที่กำหนดในการ Master On และ Start.....	60
3.54 โปรแกรมลำดับในการเริ่มโปรแกรมหุ่นยนต์.....	60
3.55 การส่งค่าคำสั่งซื้อพร้อมให้กับหุ่นยนต์.....	61
3.56 โปรแกรมนับเวลา.....	61
3.57 โปรแกรมในการเพิ่มลดค่าจากปุ่มใน GOT.....	62
3.58 โปรแกรมในการตั้งค่าความเร็วหุ่นยนต์	62
3.59 โปรแกรมอ่านค่าความเร็วหุ่นยนต์.....	63
3.60 โปรแกรมในการกำหนดค่าให้กับหน้าต่างฟ็อบบี้ในจอ GOT.....	63
3.61 การตั้งค่าฟ็อบบี้ให้สามารถสั่งเปิดผ่านโปรแกรมพีแอลซีได้	63
3.62 โปรแกรมแสดงฟ็อบบี้กรณีที่ไม่สามารถกด Start ได้	64
3.63 โปรแกรมแสดงฟ็อบบี้แสดงหลังจากเสร็จคำสั่งซื้อว่าเป็นสำเร็จหรือไม่	64
3.64 โปรแกรมกดปรับความเร็วสายพานผ่านจอ GOT	65

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.65 โปรแกรม Stock Inventory.....	65
3.66 โปรแกรมตรวจสอบคำสั่งซื้อ	66
3.67 โปรแกรมกำหนดความเร็วอินเวอร์เตอร์	66
3.68 โปรแกรมในการสั่งตัวปล่อยยาและหน่วงด้วยตัวจับเวลา.....	67
3.69 โปรแกรมควบคุมกระบอกลมดันกล่อง	67
3.70 โปรแกรมส่งค่าไปกลับกับหุ่นยนต์	68
3.71 โปรแกรมลำดับในการเคลื่อนที่ของสายพานฝั่งกล่อง	68
3.72 ผังงานการทำงานของการปล่อยกล่องบรรจุยา	69
3.73 โปรแกรมควบคุมไค์ดรีตเตอร์ส่วนการสั่งถ่าย	70
3.74 การทำงานของโปรแกรมจัดการค่าจาก QR code	71
3.75 โปรแกรมในการนับจำนวนปล่อยยา.....	72
3.76 โปรแกรมแจ้งเตือนหากมีแอกชูเอเตอร์ติด.....	72
3.77 โปรแกรมคำนวณค่า OEE	73
3.78 องค์ประกอบของเอชเอ็มไอที่จัดสร้าง	74
3.79 หน้า Start/help.....	75
3.80 หน้า Main Menu.....	76
3.81 การตั้งรหัสผ่านของ level 1 ในซอฟต์แวร์ GT Designer3.....	76
3.82 การปรับการแสดงผลและสัญญาณขาเข้าของปุ่มเป็นระดับ 1 ในซอฟต์แวร์ GT Designer3	76
3.83 การตั้งค่าปุ่ม Log out ในซอฟต์แวร์ GT Designer3.....	77
3.84 หน้า Main Control	78
3.85 หน้า Product Order.....	79
3.86 ปุ่มปุ่ม Manual Add Order	79
3.87 หน้า Scan here.....	80
3.88 ปุ่มปุ่ม Confirm Order	80
3.89 หน้า Stock Inventory	81
3.90 หน้า Mechanical Operation.....	82
3.91 หน้า Inverter Operation	82
3.92 หน้า Vision Monitoring ก่อนเชื่อมวิชันเซนเซอร์.....	83
3.93 หน้า Vision Monitoring หลังเชื่อมวิชันเซนเซอร์.....	83
3.94 การตั้งค่าปุ่มเข้า Vision Monitoring.....	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ xii อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.95 หน้าหุ่นยนต์ Operation	84
3.96 หน้า Error Current.....	85
3.97 หน้า Error History.....	85
3.98 หน้า Error Current System.....	86
3.99 ฟ็ออป้อป Alarm.....	86
3.100 หน้า Dashboard	89
3.101 หน้าโครงสร้างโดยรวมของระบบ	89
3.102 หน้าโครงสร้างโดยรวมของระบบ IP.....	90
3.103 การตั้งค่าการตั้งค่าการเปิดใช้งาน Emulator	91
3.104 การตั้งค่าการเก็บไฟล์รูปและดึงไฟล์รูปมาใช้งาน.....	92
3.105 การตั้งค่าการตั้งค่าการส่งถ่ายของ HSOUT2 หรือไฟส่องสว่างหน้าวิชันเซนเซอร์	92
3.106 ตัวอย่างการตั้งค่าการเชื่อมต่อคอมพิวเตอร์กับวิชันเซนเซอร์.....	93
3.107 การปรับโฟกัสและตั้งค่าการรับรูปเบื้องต้น	93
3.108 การใช้ฟังก์ชัน PatMax Pattern (1-10) ในการแยกตัวยา.....	94
3.109 การตั้งค่าของ PatMax Pattern (1-10) ในการแยกตัวยา	94
3.110 การกำหนดสีเพื่อให้วิชันเซนเซอร์แยกสี	95
3.111 เงื่อนไขการคัดแยกข้อมูลจำนวนยาในเม็ดยาที่แยกสี.....	96
3.112 เงื่อนไขการคัดแยกข้อมูลตำแหน่งยาในเม็ดยาที่แยกสี	96
3.113 String ที่ส่งไปยังหุ่นยนต์.....	96
3.114 ตัวอย่างข้อมูลที่ส่งไปยังตัวหุ่นยนต์	97
3.115 ตำแหน่งช่องสำหรับวางยา.....	97
3.116 ผังงานการทำงานของโปรแกรมหยิบยาใส่กล่อง	98
3.117 ผังงานการทำงานของโปรแกรมจัดที่วางแบบจัดยาเอง	99
3.118 ผังงานการทำงานของโปรแกรมจัดที่วางแบบแบ่งตามมือ	100
3.119 ผังงานการทำงานของโปรแกรมวางยา	101
3.120 หน้าตั้งค่าพารามิเตอร์ฟังก์ชันแทรกคิง.....	102
3.121 ผังงานการทำงานของโปรแกรม A1	102
3.122 การเปลี่ยนพิกัดของโปรแกรม A1	103
3.123 ตำแหน่งแรกของโปรแกรม A1.....	103
3.124 โปรแกรม A1 หยุดที่บรรทัดที่ 17	104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและข้อมูลอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.125 ตำแหน่งที่ 2 ของโปรแกรม A1.....	104
3.126 ผังงานการทำงานของโปรแกรม B1.....	105
3.127 การวางรูปเพื่อทำการสอบเทียบ.....	106
3.128 การเลือกตำแหน่งที่ทำการสอบเทียบ.....	106
3.129 จุดที่ใส่พิกัดที่ต้องใส่เพื่อทำการสอบเทียบ.....	107
3.130 โปรแกรม B1 หยุดที่บรรทัดที่ 15.....	107
3.131 ตำแหน่งที่หุ่นยนต์สามารถขยับได้โดยไมเซนวิชันเซนเซอร์.....	108
3.132 การบันทึกข้อมูลสอบเทียบเพื่อนำไปใช้.....	108
3.133 ผังงานการทำงานของโปรแกรม C1.....	109
3.134 การเลือกข้อมูลที่สอบเทียบมาใช้.....	110
3.135 โปรแกรม C1 หยุดที่บรรทัดที่ 28.....	110
3.136 ผังงานการทำงานของโปรแกรม 1.....	111
3.137 ผังงานการทำงานของโปรแกรม CM1.....	112
3.138 ผังงานการทำงานของโปรแกรม 1 ที่รวมกับโปรแกรมวางยาแล้ว.....	113
3.139 รูปแบบการส่งข้อมูลที่วิชันเซนเซอร์ส่งให้หุ่นยนต์.....	114
3.140 การอ่านข้อมูลที่วิชันเซนเซอร์ส่งมาของหุ่นยนต์.....	114
3.141 ผังงานการทำงานของโปรแกรม CM1 ที่ปรับปรุงแล้ว.....	115
4.1 หน้าแรกของเว็บแอปพลิเคชัน.....	116
4.2 หน้าตะกร้าสินค้า.....	117
4.3 คิวอาร์โค้ดจากคำสั่งซื้อแบบเลือกยาเอง.....	117
4.4 หน้าใส่ชื่อก่อนถามอาการ.....	118
4.5 หน้าเลือกตอบอาการ.....	118
4.6 คิวอาร์โค้ดจากคำสั่งซื้อแบบเลือกอาการ.....	119
4.7 คิวอาร์โค้ดของเว็บแอปพลิเคชัน.....	119
4.8 GOT เมื่อสแกนคิวอาร์โค้ดได้.....	120
4.9 ความผิดพลาดของยาชนิด B ที่เกิดขึ้นจากการตั้งออกมาจากตัวปล่อยยา.....	121
4.10 ความผิดพลาดของยาชนิด C ที่เกิดขึ้นจากการกริ่งออกจากสายพาน.....	121
4.11 รูปแบบการปล่อยยาที่ไม่มีความผิดพลาด.....	122
4.12 การปล่อยยาชนิด C ออกมาเพิ่มหลังจากเกิดข้อผิดพลาด.....	123

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและข้อมูลอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

เนื่องจากทางบริษัท MELFT (Mitsubishi Electric Factory Automation Thailand) ต้องการชุดสายิตที่ประกอบด้วยสินค้าของบริษัท MELFT ได้แก่

1. หุ่นยนต์สกราก (Selective Compliance Assembly Robot Arm: SCARA) รุ่น RH-6FH5520-1D-S11 มีความแม่นยำและความเร็วในการหยิบและวาง (Pick and Place) สูง

2. พีแอลซี (Programmable Logic Controller: PLC) รุ่น R08CPU มีความสามารถในการประมวลผลและควบคุมอุปกรณ์ผ่านโปรโตคอลต่าง ๆ

3. เอชเอ็มไอ (Human Machine Interface: HMI) รุ่น GT2708-STBA หรือ จีโอที (Graphic Operation Terminal: GOT) ซึ่งเป็นคำเรียกเฉพาะของมิตซูบิชิ อิเล็กทริก มีความสามารถถูกออกแบบหน้าจอเพื่อให้เข้ากับการใช้งานเครื่องจักรต่าง ๆ ได้อย่างเหมาะสม

4. อินเวอร์เตอร์ (Inverter) รุ่น FR-E800 สามารถควบคุมความเร็วและกระแส เพื่อควบคุมมอเตอร์ไฟฟ้า นอกจากนี้ยังสามารถสื่อสารโปรโตคอลต่าง ๆ เพื่อสามารถถูกควบคุมจากพีแอลซีได้

5. วิชันเซนเซอร์ (Vision Sensor) รุ่น VS80C-402R สามารถตัดแยกสีและรูปร่างได้ในการระบุชนิดหรือตรวจดีเฟกต์ (Defect) ของชิ้นงาน เมื่อใช้งานร่วมกับเอ็นโค้ดเดอร์ (Encoder) สามารถใช้งานฟังก์ชันวิชันแทรคคิง (Vision Tracking) ที่มีเอานต์พุดเป็นพิกัด X Y Angle เพื่อติดตามวัตถุบนสายพานให้หุ่นยนต์สามารถหยิบวัตถุในขณะที่สายพานเคลื่อนที่ได้

6. โค้ดรีดเดอร์ (Code Reader) รุ่น CF26 สามารถอ่านค่าโค้ดต่าง ๆ รวมถึงคิวอาร์โค้ด (QR Code) และส่งข้อมูลที่อ่านจากคิวอาร์โค้ดให้กับพีแอลซีได้

โดยที่อุปกรณ์เหล่านี้สามารถทำงานร่วมกันได้เพื่อนำชุดสายิตแสดงในงาน Manufacturing Expo 2023 ชุดสายิตสามารถสร้างความคิดและมุมมองในการประยุกต์งานใช้ผลิตภัณฑ์ของทางบริษัท MELFT ให้กับผู้ที่สนใจการสร้างระบบอัตโนมัติและสนใจสินค้าของทางบริษัท MELFT

คณะผู้จัดทำสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (KMITL) และบริษัท MELFT จึงมีความร่วมมือในการออกแบบฟังก์ชันการทำงานและการเขียนโปรแกรมสำหรับชุดสายิตที่สามารถทำให้อุปกรณ์ข้างต้นสามารถทำงานร่วมกันได้ เนื่องจากสถานการณ์การระบาดของโควิด-19 จึงออกแบบชุดสายิตการบรรจุยา (Medicine Packing) เพื่อให้เข้ากับสถานการณ์

1.2 วัตถุประสงค์

การออกแบบฟังก์ชันการทำงานและการเขียนโปรแกรมสำหรับชุดสายิตการบรรจุยาดังนี้คือ การรับข้อมูลคำสั่งซื้อของผู้ใช้งานเพื่อส่งให้ชุดสายิตผ่านคิวอาร์โค้ด กลไกปล่อยยาให้ทำการปล่อยยาตามจำนวนที่สั่งลงบนสายพาน และหุ่นยนต์สกรากใช้งานฟังก์ชันวิชันแทรคคิงหยิบยาในขณะที่สายพานเคลื่อนที่ลงกล่อง โดยที่ยาชนิดเดียวกันอยู่ในกล่องช่องเดียวกัน และนำกล่องที่มียาครบจำนวนตามที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและทำซ้ำอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการออกจากชุดสาธิตให้ผู้ใช้งานเมื่อทำงานตามคำสั่งซื้อของผู้ใช้งานสำเร็จ แล้วนำกล่องใหม่กลับมายังจุดรอรับคำสั่งซื้อต่อไป เพื่อให้ได้ผลตามที่ออกแบบจำเป็นต้องมีการเขียนโปรแกรมควบคุมในส่วนต่าง ๆ ของชุดสาธิตดังนี้

1. สร้างเว็บแอปพลิเคชัน (Web Application) สำหรับสร้างคิวอาร์โค้ด
2. โค้ดรีดเดอร์ CF26 ด้วยซอฟต์แวร์ Cognex DataMan Setup Tool for MELSENSOR
3. พีแอลซีรุ่น R08CPU ด้วยซอฟต์แวร์ GX Works3
4. หน้าจอเอชเอ็มไอรุ่น GT2708-STBA ด้วยซอฟต์แวร์ GT Designer3
5. วิชันเซนเซอร์รุ่น VS80C-402R ด้วยซอฟต์แวร์ In-Sight Explorer for MELSENSOR
6. ตัวควบคุมของหุ่นยนต์สการารุ่น CR751-D ด้วยซอฟต์แวร์ RT ToolBox3

1.3 ขอบเขตของงาน

การออกแบบฟังก์ชันการทำงานของชุดสาธิตดังนี้คือ

1. ให้ผู้ใช้งานเข้าเว็บแอปพลิเคชันทำการเลือกชนิดและจำนวนยาที่ต้องการและเมื่อยืนยันชนิดและจำนวนยาที่ต้องการ เว็บแอปพลิเคชันทำการสร้างคิวอาร์โค้ดเพื่อให้ผู้ใช้สามารถนำ คิวอาร์โค้ดสแกนหน้าโค้ดรีดเดอร์ CF26 เพื่อส่งข้อมูลชนิดและจำนวนยาให้กับพีแอลซีและตัวควบคุมหุ่นยนต์สการา
2. หุ่นยนต์สการาทำการเปิดกล่องและพีแอลซีสั่งกลไกปล่อยยาให้ทำการปล่อยยาลงสายพานตามจำนวนที่ถูกสั่งและมีการถ่ายเม็ดยาโดยใช้วิชันเซนเซอร์ VS80
3. ตัวควบคุมหุ่นยนต์สการางานฟังก์ชันวิชันแทรกคิงโดยใช้วิชันเซนเซอร์ถ่ายเพื่อระบุตำแหน่งเริ่มต้นและเอ็นโค้ดเดอร์เพื่อระบุตำแหน่งตามสายพานทำให้สามารถหยิบยาในขณะที่สายพานเคลื่อนที่ได้ วิชันเซนเซอร์ยังสามารถคัดแยกยาจากสีและรูปร่างและส่งสัญญาณเอาต์พุตให้ตัวควบคุมหุ่นยนต์สการาให้นำยาชนิดเดียวกันใส่ในกล่องช่องเดียวกัน
4. หุ่นยนต์สการาหยิบยาใส่กล่องจนครบซึ่งหุ่นยนต์สการาจะทำการปิดฝากล่อง จากนั้นพีแอลซีสั่งสายพานและกลไกเพื่อนำกล่องที่บรรจุยาตามที่ส่งออกมาให้ผู้ใช้และนำกล่องเก่าใหม่ประจำที่แทนเพื่อรอรับคำสั่งต่อไป

การเขียนโปรแกรมสำหรับชุดสาธิตสำหรับอุปกรณ์ต่อไปนี้

1. การสร้างจากเว็บแอปพลิเคชันด้วย Visual Studio Code โดยมีฟังก์ชันที่สามารถคัดแยกกลุ่มอาการของผู้ใช้งานเบื้องต้นหรือใส่ข้อมูลชนิดและจำนวนยาที่ต้องการ เพื่อทำการสร้างคิวอาร์โค้ดเพื่อส่งข้อมูลชนิดและจำนวนยาที่ต้องการได้
2. การตั้งค่าโค้ดรีดเดอร์ CF26 ด้วยซอฟต์แวร์ Cognex DataMan Setup Tool for MELSENSOR เพื่อรับค่าจากคิวอาร์โค้ดที่สร้างจากเว็บแอปพลิเคชัน
3. การเขียนโปรแกรมสำหรับพีแอลซีรุ่น R08CPU ด้วยซอฟต์แวร์ GX Works3 เพื่อสร้างฟังก์ชันการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 2 2 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การควบคุมหลักในการเปิดและปิด หรือ เปลี่ยนโหมดการทำงานเป็นโหมดอัตโนมัติ (Automatic Mode) และโหมดแมนนวล (Manual Mode)
- การทำงานของเอชเอ็มไอที่เชื่อมกับการทำงานส่วนต่าง ๆ ของชุดสาธิต รวมถึงการป้อนข้อมูลและคำนวณค่าต่าง ๆ ที่แสดงผลในหน้าจอเอชเอ็มไอ
- การควบคุมโซลินอยด์วาล์ว (Solenoid Valve) เพื่อควบคุมกระบอกสูบในกลไกต่าง ๆ
- การควบคุมการทำงานของสายพาน
- การควบคุมโค้ดรีดเดอร์ CF26 เช่น การส่งถ่ายและการประมวลผลข้อมูลที่ได้รับจากโค้ดรีดเดอร์
- การควบคุมคิกและการปล่อยยา
- การแสดงผลแจ้งเตือน

4. การเขียนโปรแกรมสำหรับหน้าจอเอชเอ็มไอรุ่น GT2708-STBA ด้วยซอฟต์แวร์ GT Designer3 เพื่อสร้างฟังก์ชันการทำงานดังนี้

- การใช้หน้าจอ GOT ในการควบคุมชุดสาธิต
- การใช้หน้าจอ GOT ในการทดสอบการทำงานต่าง ๆ เช่น การทำงานของสายพานและการทำงาน



ของกลไกปล่อยยาและปล่อยกล่องบรรจุยา

- การใช้หน้าจอ GOT ในการระบุข้อผิดพลาดและการแจ้งเตือนของชุดสาธิตได้
- การใช้หน้าจอ GOT ในการแสดงข้อมูลชุดสาธิตดังนี้ เวลาการทำงานของชุดสาธิต จำนวนยาที่ใช้ไป จำนวนคำสั่งที่ทำ จำนวนคำสั่งที่เป็น OK หรือ NG เวลาที่ใช้ในการทำคำสั่ง และค่าโออีอี (Overall Equipment Effectiveness: OEE)




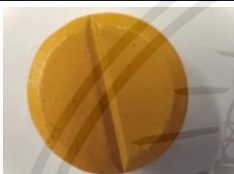
5. การเขียนโปรแกรมสำหรับวิชชันเซนเซอร์รุ่น VS80C-402R ด้วยซอฟต์แวร์ In-Sight Explorer เพื่อสร้างฟังก์ชันการทำงานดังนี้

- การคัดแยกชนิดของยาทั้ง 6 ชนิดตามตารางที่ 1.1

ตารางที่ 1.1 รูปร่างและชื่อของยาทั้ง 6 ชนิด

รูปยา	ชื่อยา
	ยาเม็ดกลมสีขาว (ยา A)
	ยาเม็ดรูปแคปซูลสีขาว (ยา B)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและทั้ง 3 อังอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ยาแคปซูลสีขาวเหลือง (ยา C)
	ยาเม็ดกลมสีชมพู (ยา D)
	ยาเม็ดกลมสีส้ม (ยา E)
	ยาเม็ดกลมสีเหลือง (ยา F)

- การระบุตำแหน่งศูนย์กลางของยาเพื่อใช้ในการหยิบ
- การส่งข้อมูลชนิดและตำแหน่งให้กับตัวควบคุมหุ่นยนต์

6. การเขียนโปรแกรมสำหรับตัวควบคุมของหุ่นยนต์สการารุ่น CR751-D ด้วยซอฟต์แวร์ RT ToolBox3 เพื่อสร้างฟังก์ชันการทำงานดังนี้

- การเขียนโปรแกรมให้หุ่นยนต์ใช้งานฟังก์ชันวิชันแทรคคิง
- การเขียนโปรแกรมหยิบยาใส่กล่อง

1.4 ขั้นตอนการดำเนินงาน

โดยมีการดำเนินงานในช่วงเดือน สิงหาคม พ.ศ. 2565 จนถึง เดือน พฤษภาคม พ.ศ. 2566 ตามตารางที่ 1.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 4.4 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.2 แผนการดำเนินงาน

ลำดับ	รายละเอียด	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1	ออกแบบฟังก์ชันการทำงานของชุดสาธิต										
2	ออกแบบเว็บแอปพลิเคชัน										
3	ติดต่อบริษัทผู้รวมระบบสำหรับการสร้างชุดสาธิต										
4	ออกแบบเอชเอ็มไอ										
5	สร้างจากเว็บแอปพลิเคชัน										
6	เขียนโปรแกรมสำหรับพีแอลซี										
7	เขียนโปรแกรมสำหรับหน้าจอเอชเอ็มไอ										
8	เขียนโปรแกรมสำหรับวิชชันเซนเซอร์										
9	ตั้งค่าโคคิรีดเดอร์										
10	เขียนโปรแกรมสำหรับตัวควบคุมหุ่นยนต์สการา										
11	รวมระบบ										
12	ทดสอบและแก้ไขระบบ										

1.5 ผลที่คาดว่าจะได้รับ

ชุดสาธิตสามารถบรรจุยาด้วยหุ่นยนต์สการาโดยรับคำสั่งด้วยการสแกนคิวอาร์โค้ดที่ถูกสร้างตามคำสั่งที่ได้รับมาได้และสามารถนำไปจัดแสดงในงาน Manufacturing Expo 2023 (21-24 มิ.ย. 2566) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ/หรืออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 แนวคิดและหลักการที่เกี่ยวข้อง

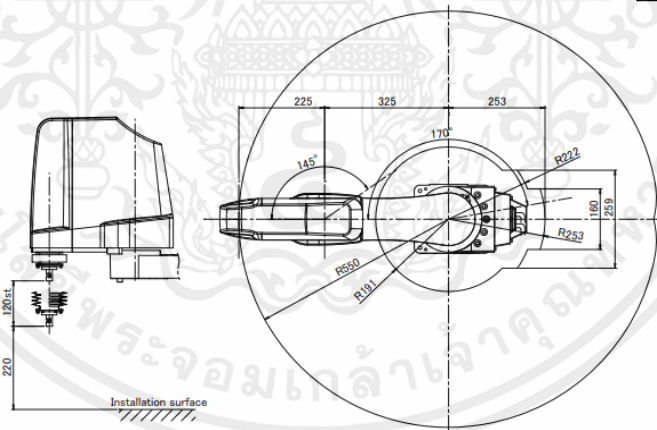
2.1 กล่าวนำ

ในบทนี้กล่าวถึงแนวคิดและหลักการที่เกี่ยวข้องในการดำเนินการปฏิรูประบบนี้ซึ่งประกอบไปด้วยหุ่นยนต์สกรู พัลลิ่ง อินเวอร์เตอร์ ฟังก์ชันวิชันแทรกคิง โปรโตคอล และเครื่องมือพัฒนาเว็บแอปพลิเคชัน

2.2 หุ่นยนต์สกรูของมิตซูบิชิ อีเล็กทริก

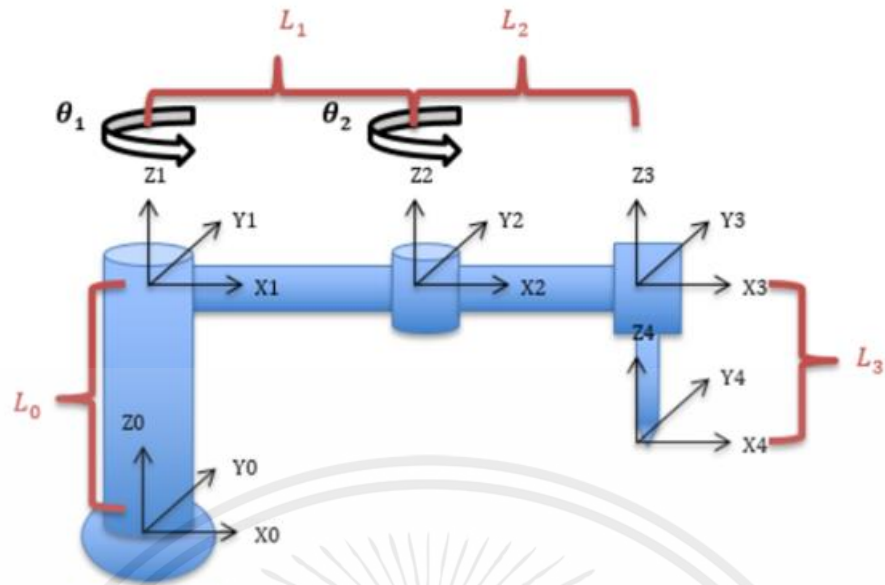


รูปที่ 2.1 หุ่นยนต์สกรูของมิตซูบิชิ อีเล็กทริก [1]



รูปที่ 2.2 พื้นที่การทำงานของหุ่นยนต์สกรูของมิตซูบิชิ อีเล็กทริก [1]

หุ่นยนต์สกรูเป็นหนึ่งในประเภทของแขนหุ่นยนต์ มักใช้ในงานประกอบชิ้นส่วน และงานหยาบวงชิ้นส่วน มีหน้าตาของหุ่นยนต์ตามรูปที่ 2.1 พื้นที่การทำงาน (Workspace) ของหุ่นยนต์สกรูมีพื้นที่รูปทรงเป็นทรงกระบอกโดยมีรูตรงกลางดังรูปที่ 2.2 ต้องออกแบบการทำงานของกระบวนการให้อยู่ในขอบเขตของหุ่นยนต์สกรู



รูปที่ 2.3 จลนศาสตร์ของหุ่นยนต์สการา [2]

$$x = C_1(C_2L_2 + L_1) - L_2S_1S_1 \quad (2.1)$$

$$y = C_1L_2S_2 + L_2C_2S_1 + L_1S_1 \quad (2.2)$$

$$z = L_0 - L_3 \quad (2.3)$$

เมื่อ x คือ พิกัด x ที่ได้จากการคำนวณ
 y คือ พิกัด y ที่ได้จากการคำนวณ
 z คือ พิกัด z ที่ได้จากการคำนวณ

C_1 คือ ค่า \cos ของมุมหุ่นยนต์ในแกนที่ 1

C_2 คือ ค่า \cos ของมุมหุ่นยนต์ในแกนที่ 2

S_1 คือ ค่า \sin ของมุมหุ่นยนต์ในแกนที่ 1

S_2 คือ ค่า \sin ของมุมหุ่นยนต์ในแกนที่ 2

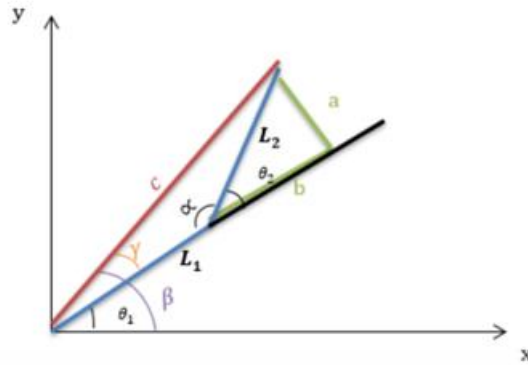
L_0 คือ ความยาวหุ่นยนต์ในช่วงที่ 1

L_1 คือ ความยาวหุ่นยนต์ในช่วงที่ 2

L_2 คือ ความยาวหุ่นยนต์ในช่วงที่ 3

L_3 คือ ความยาวหุ่นยนต์ในช่วงที่ 4

จลนศาสตร์ (Kinematic) ของหุ่นยนต์คือการคำนวณหาตำแหน่งปลายแขนหุ่นยนต์จากระยะแต่ละส่วนของแขนหุ่นยนต์และมุมแต่ละมุมของหุ่นยนต์โดยจลนศาสตร์ของหุ่นยนต์สการาสามารถคำนวณได้จากสมการที่ (2.1) (2.2) และ (2.3) โดยมีมุมและความยาวตามรูปที่ 2.3



รูปที่ 2.4 จลนศาสตร์ผกผันของหุ่นยนต์สกรารา [2]

$$\theta_1 = \arctan\left(\frac{Y}{X}\right) - \arctan\left(\frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}\right) \quad (2.4)$$

$$\theta_2 = \arctan\left(\frac{\pm \sqrt{1-A^2}}{A}\right) \quad (2.5)$$

$$A = \frac{X^2 + Y^2 - L_1^2 - L_2^2}{2L_1 L_2} \quad (2.6)$$

$$L_3 = L_0 - Z \quad (2.7)$$

เมื่อ X คือ พิกัดแกน x ที่ต้องการให้หุ่นยนต์อยู่

Y คือ พิกัดแกน y ที่ต้องการให้หุ่นยนต์อยู่

Z คือ พิกัดแกน z ที่ต้องการให้หุ่นยนต์อยู่

L_1 คือ ความยาวหุ่นยนต์ในช่วงที่ 2

L_2 คือ ความยาวหุ่นยนต์ในช่วงที่ 3

L_3 คือ ความยาวหุ่นยนต์ในช่วงที่ 4

A คือ รุบย่อรวมตัวแปรที่ใช้ในการคำนวณในสมการที่ (2.5)

θ_1 คือ มุมการหันของหุ่นยนต์ที่ 1

θ_2 คือ มุมการหันของหุ่นยนต์ที่ 2

จลนศาสตร์ผกผัน (Inverse Kinematic) ของหุ่นยนต์คือการคำนวณย้อนกลับหาระยะแต่ละส่วนของแขนหุ่นยนต์และมุมแต่ละมุมของหุ่นยนต์ จากตำแหน่งปลายแขนหุ่นยนต์โดยจลนศาสตร์ผกผันของหุ่นยนต์สกราราสามารถคำนวณได้จากสมการที่ (2.4) (2.5) (2.6) และ (2.7) โดยมีมุมและความยาวตามรูปที่ 2.4

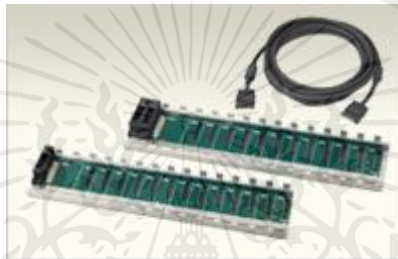
2.3 พีแอลซีของมิตซูบิชิ อิเล็กทริก

พีแอลซีคืออุปกรณ์ควบคุมการทำงานของเครื่องจักรและกระบวนการทำงานต่าง ๆ พีแอลซีสามารถส่งข้อมูลในรูปแบบของกระแสไฟฟ้าในการใช้เป็นอินพุต/เอาต์พุต (Input/Output: I/O) ได้ทั้งแบบดิจิทัลและแอนะล็อกเพื่อใช้ในการควบคุมระบบ พีแอลซีในรูปแบบโมดูลประกอบด้วยหลายส่วนดังต่อไปนี้ เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 8 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ซีพียูของพีแอลซีของมิตซูบิชิ อิเล็กทริก [3]

ซีพียู (CPU) ของมิตซูบิชิ อิเล็กทริกจากรูปที่ 2.5 คืออุปกรณ์ที่สามารถประมวลผลข้อมูลคณิตศาสตร์ทั้งในรูปแบบตรรกะและการคำนวณ ประมวลผลข้อมูลในรูปแบบของเลขฐาน 2 และพีแอลซีสามารถเชื่อมกับโมดูลต่าง ๆ ในการทำหน้าที่เฉพาะเจาะจงได้



รูปที่ 2.6 หน่วยฐานของพีแอลซีของมิตซูบิชิ อิเล็กทริก [3]

หน่วยฐาน (Base Unit) ของมิตซูบิชิ อิเล็กทริกจากรูปที่ 2.6 ใช้สำหรับการติดตั้งโมดูลต่าง ๆ เช่น แหล่งจ่ายไฟ ซีพียูและโมดูลอินพุตกับเอาต์พุต มีหน่วยฐานหลายขนาดเพื่อให้เหมาะสมกับความต้องการของระบบต่าง ๆ



รูปที่ 2.7 แหล่งจ่ายไฟพีแอลซีของมิตซูบิชิ อิเล็กทริก [3]

แหล่งจ่ายไฟของมิตซูบิชิ อิเล็กทริกจากรูปที่ 2.7 โมดูลนี้ใช้สำหรับการจ่ายพลังงานไฟฟ้าสำหรับโมดูลซีพียู โมดูลอินพุต/เอาต์พุต และโมดูลอื่น ๆ บนหน่วยฐาน

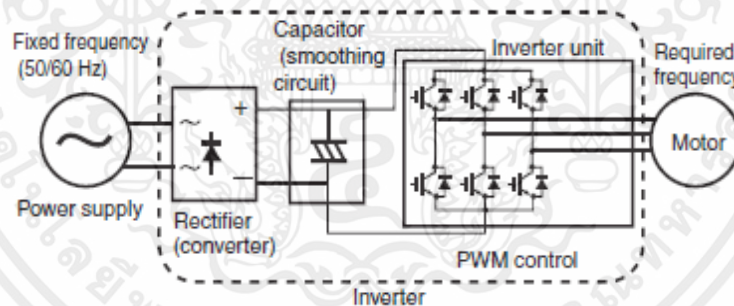


รูปที่ 2.8 โมดูลอินพุต/เอาต์พุตดิจิทัลและโมดูลอินพุต/เอาต์พุตแอนะล็อกของมิตซูบิชิ อิเล็กทริก [3]

โมดูลอินพุต/เอาต์พุตของมิตซูบิชิ อิเล็กทริกจากรูปที่ 2.8 ใช้สำหรับส่งข้อมูลดิจิทัลหรือแอนะล็อกไปยังอุปกรณ์ต่าง ๆ เช่น สวิตช์ ไฟแสดงสถานะ และเซนเซอร์ และเซนเซอร์ สามารถเชื่อมต่อกับระบบควบคุมได้อย่างง่ายดายโดยใช้โมดูลอินพุต/เอาต์พุตดิจิทัลและโมดูลอินพุต/เอาต์พุตแอนะล็อก โมดูลแอนะล็อกคืออินเทอร์เฟซหลักระหว่างกระบวนการควบคุมและระบบอัตโนมัติ และออกแบบมาโดยเฉพาะสำหรับอินเทอร์เฟซที่มีเซนเซอร์ ซึ่งประมวลผลแรงดันไฟฟ้าและสัญญาณกระแสไฟฟ้าต่าง ๆ

2.4 อินเวอร์เตอร์ของมิตซูบิชิ อิเล็กทริก

อินเวอร์เตอร์โดยทั่วไปหมายถึงการเปลี่ยนกระแสไฟฟ้าตรงเป็นกระแสไฟฟ้าสลับ แต่ในบริบทนี้หมายถึงอุปกรณ์ที่มีการเปลี่ยนไฟฟ้ากระแสสลับเป็นไฟฟ้ากระแสตรงและไฟฟ้ากระแสตรงเป็นไฟฟ้ากระแสสลับเพื่อให้สามารถควบคุมความถี่กระแสเพื่อควบคุมมอเตอร์ไฟฟ้าได้ โดยมีวงจรที่ใช้สำหรับเปลี่ยนกระแสไฟฟ้าตามรูปที่ 2.9 และสามารถคำนวณความเร็วจริงของมอเตอร์ได้ผ่านสมการที่ (2.8) และ (2.9)



รูปที่ 2.9 การแปลงไฟฟ้าเพื่อเปลี่ยนความถี่ไฟฟ้าและควบคุมมอเตอร์ [4]

$$n = n_s(1 - s) \quad (2.8)$$

$$n = \frac{120f}{p}(1 - s) \quad (2.9)$$

เมื่อ n คือ ความเร็วจริงของมอเตอร์

n_s คือ ความเร็วซิงโครนัสของมอเตอร์

f คือ ความถี่แหล่งจ่ายไฟ

p คือ จำนวนขั้วของสเตเตอร์

s คือ ค่าสลิปของมอเตอร์



รูปที่ 2.10 อินเวอร์เตอร์ของมิตซูบิชิ อิเล็กทรอนิกส์ FR-E800 [5]

อินเวอร์เตอร์ของมิตซูบิชิ อิเล็กทรอนิกส์ FR-E800 ตามรูปที่ 2.10 เป็นตัวควบคุมมอเตอร์และยังมีหน้าที่อื่น เช่น การทำระบบชิ่งโครนัส และการรองรับโปรโตคอลการสื่อสารต่าง เช่น CC-Link IEF Basic เพื่อใช้ส่งข้อมูลให้อุปกรณ์อื่น ๆ

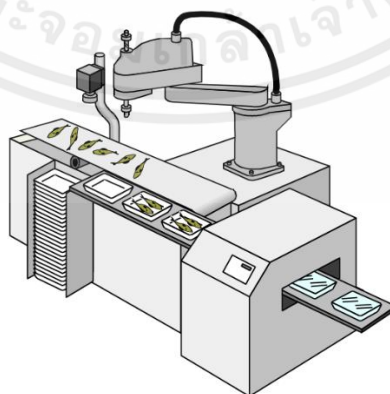
2.5 ฟังก์ชันวิชันแทรกคิงสำหรับหุ่นยนต์สการาของมิตซูบิชิ อิเล็กทรอนิกส์

ฟังก์ชันวิชันแทรกคิงเป็นฟังก์ชันที่ช่วยให้หุ่นยนต์สามารถติดตามชิ้นงานบนสายพานลำเลียงขณะที่กำลังเคลื่อนที่ได้ ด้วยฟังก์ชันนี้ทำให้สามารถขนย้ายชิ้นงานและอื่น ๆ ได้โดยไม่จำเป็นต้องหยุดสายพาน และยังลดกลไกเชิงกลต่าง ๆ ที่จำเป็นในการแก้ไขตำแหน่งชิ้นงาน ฟังก์ชันวิชันแทรกคิงมีคุณสมบัติดังนี้

- สามารถติดตามชิ้นงานที่เรียงกันซึ่งกำลังเคลื่อนที่บนสายพาน
- สามารถติดตามชิ้นงานที่ไม่ได้เรียงกันซึ่งกำลังเคลื่อนที่บนสายพาน
- สามารถติดตามชิ้นงานที่เปลี่ยนแปลงความเร็วได้เนื่องจากการคำนวณความเร็วการเคลื่อนที่ของสายพานอัตโนมัติ

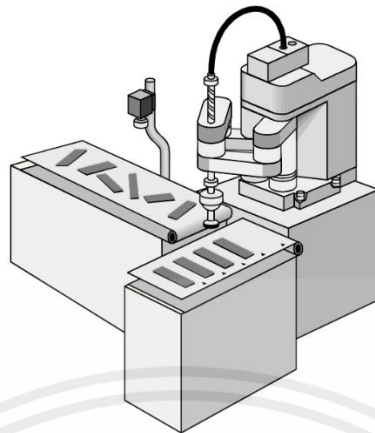
มีตัวอย่างการนำไปใช้งานดังนี้

- บรรจุอาหารแปรรูปโดยมีหุ่นยนต์สการาทำงานดังรูปที่ 2.11



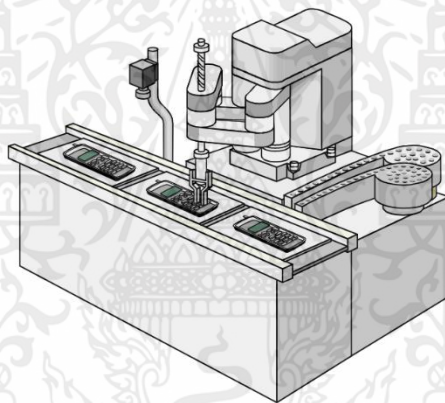
รูปที่ 2.11 ตัวอย่างการบรรจุอาหารแปรรูป [6]

- เรียงชิ้นส่วนโดยมีหุ่นยนต์สกราทำงานดังรูปที่ 2.12



รูปที่ 2.12 ตัวอย่างการเรียงชิ้นส่วน [6]

- ประกอบอุปกรณ์ไฟฟ้าขนาดเล็กโดยมีหุ่นยนต์สกราทำงานดังรูปที่ 2.13



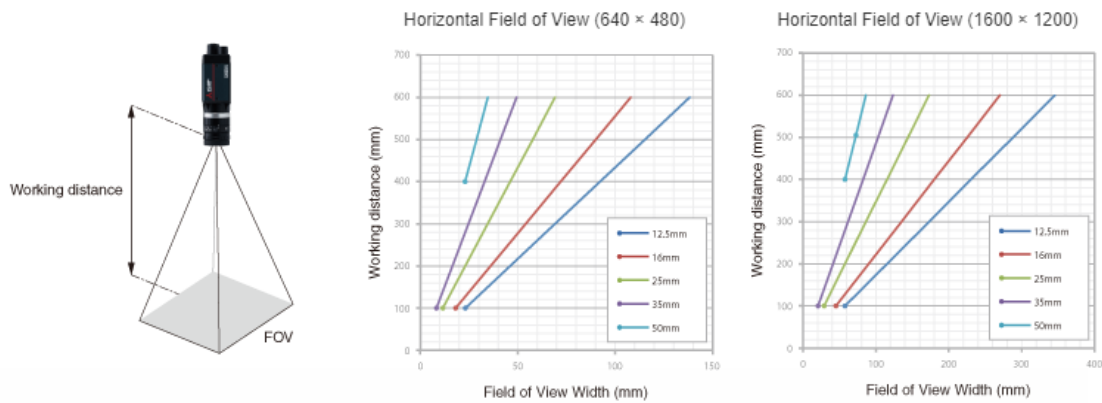
รูปที่ 2.13 ตัวอย่างการประกอบอุปกรณ์ไฟฟ้าขนาดเล็ก [6]

ในวิชันแทรกคิงมีการใช้งาน 2 อุปกรณ์ดังต่อไปนี้ร่วมกันดังต่อไปนี้

2.5.1 วิชันเซนเซอร์ของมิตซูบิชิ อีเล็กทริก

วิชันเซนเซอร์เป็นอุปกรณ์หลักในโรงงานที่ตรวจจับข้อมูลที่จำเป็นสำหรับการวิเคราะห์สินค้าและการคัดแยกประเภทหรือระบุตำแหน่ง ในรุ่น VS80 สามารถระบุชิ้นงานที่ผ่านไปด้วยความเร็วสูงได้ มีขนาดเล็กสามารถติดตั้งได้ง่าย และยังสามารถติดตั้งไว้ที่ปลายแขนของหุ่นยนต์แล้วทำงานร่วมกันได้ มีรุ่นสีและขาวดำให้เลือกใช้มาพร้อมกับการติดตั้งสาย PoE (Power over Ethernet) สามารถลดการเดินสายให้น้อยลงได้

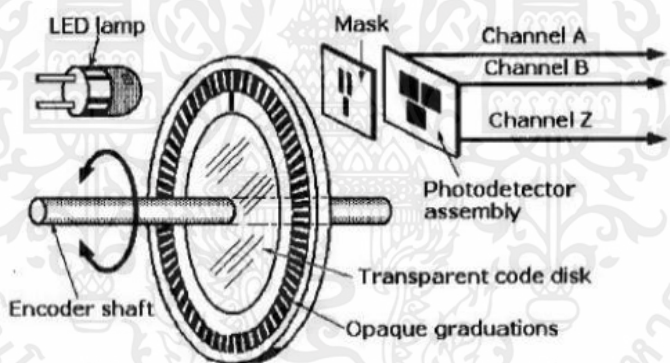
ระยะเวลาการทำงานหรือระยะทางจากเลนส์ถึงเป้าหมายการตรวจสอบและบริเวณที่วิชันเซนเซอร์มองเห็นได้ในระยะทางดังกล่าว โดยสิ่งที่ส่งผลให้วิชันเซนเซอร์มองเห็นได้นั้นจำเป็นต้องอยู่ในค่าขอบเขตการมองเห็น (Field of View: FOV) ดังรูปที่ 2.14



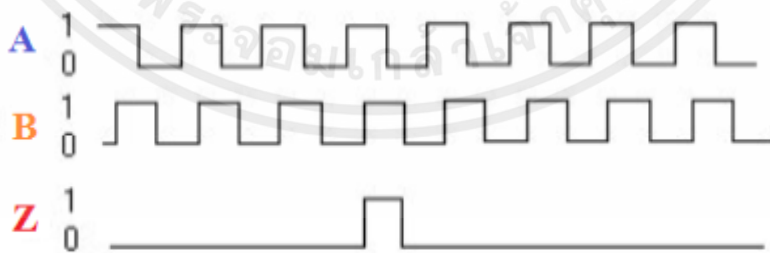
รูปที่ 2.14 ค่าขอบเขตการมองเห็นวิชันของหุ่นยนต์ [7]

2.5.2 เอ็นโค้ดเดอร์แบบหมุน

เอ็นโค้ดเดอร์แบบหมุน (Incremental Rotary Encoder) โครงสร้างดังรูปที่ 2.15 ประกอบด้วยจานหมุน และอุปกรณ์ตรวจจับ โดยจานหมุนมีช่องเล็ก ๆ เมื่อเพลลาของมอเตอร์หมุนทำให้จานหมุนไปตัดลำแสงของเซนเซอร์ (Sensor) ทำให้ชุดรับแสงได้รับสัญญาณเป็นช่วง ๆ จึงทำให้เอาต์พุตออกมาเป็นสัญญาณพัลส์ต่อรอบ (PPR) ดังรูปที่ 2.16



รูปที่ 2.15 ส่วนประกอบของเอ็นโค้ดเดอร์แบบหมุน [8]



รูปที่ 2.16 ตัวอย่างสัญญาณพัลส์ของเอ็นโค้ดเดอร์แบบหมุน [8]

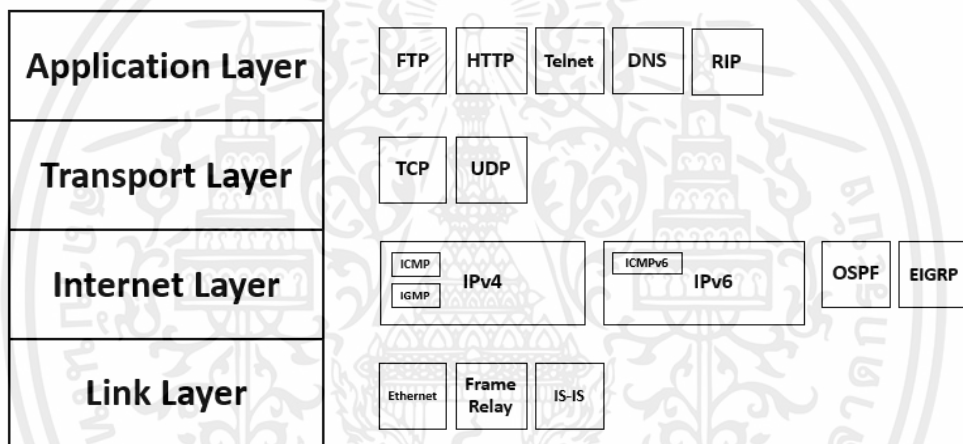
เอ็นโค้ดเดอร์แบบหมุนเป็นเอ็นโค้ดเดอร์ใช้หลักการเมื่อมีการหมุนของแกนเพลลาทำให้มีเอาต์พุตที่เป็นสัญญาณลูกคลื่นพัลส์สี่เหลี่ยม (Square wave) มี 3 แทรค (Tracks) คือ A B และ Z โดยสัมพันธ์กับระยะการเคลื่อนที่และตำแหน่งเอาต์พุตของเอ็นโค้ดเดอร์ A และ B มีมุมที่ห่างกัน 90 องศาทางไฟฟ้า ส่วน Z มีสัญญาณ 1 พัลส์ ต่อ 1 รอบหรือบางตัวเป็นพัลส์แบบกลับด้าน เช่น A- B- และ Z- ซึ่งเป็นสัญญาณที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 13 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลับเฟสกัน 90 องศาเพื่อเช็คทิศทางการหมุนของมอเตอร์เป็นต้น เอ็นโค้ดเดอร์แบบนี้มีข้อดีอยู่ในกรณีหากมีการถอดสายสัญญาณออกชั่วคราวหรือแหล่งจ่ายไฟฟ้าหายไปข้อมูลของการเคลื่อนที่นั้นหายไปไม่สามารถระบุตำแหน่งพัลส์หรือตำแหน่งองศาได้ทำให้ต้องมีการปรับที่จุดอ้างอิงใหม่อยู่ตลอดเวลา

2.6 โพรโทคอลที่เกี่ยวข้อง

2.6.1 อีเทอร์เน็ต

อีเทอร์เน็ต (Ethernet) เป็นเทคโนโลยีเครือข่ายคอมพิวเตอร์ที่เป็นฐานหลักของเทคโนโลยีสารสนเทศทั้งหมด ในอุตสาหกรรมอีเทอร์เน็ตเป็นเทคโนโลยีการสื่อสารที่เชื่อถือได้และเป็นสากล สามารถเชื่อมต่ออุปกรณ์ได้หลากหลาย ระบบอีเทอร์เน็ตนั้นใช้ภาษาสื่อสารมาตรฐานที่เรียกว่า "ทีซีพี/ไอพี" เป็นภาษาหลักในการสื่อสารกัน TCP ย่อมาจาก Transmission Control Protocol และ IP ย่อมาจาก Internet Protocol



รูปที่ 2.17 การเทียบเทคโนโลยีสื่อสารอีเทอร์เน็ตกับ OSI model [9]

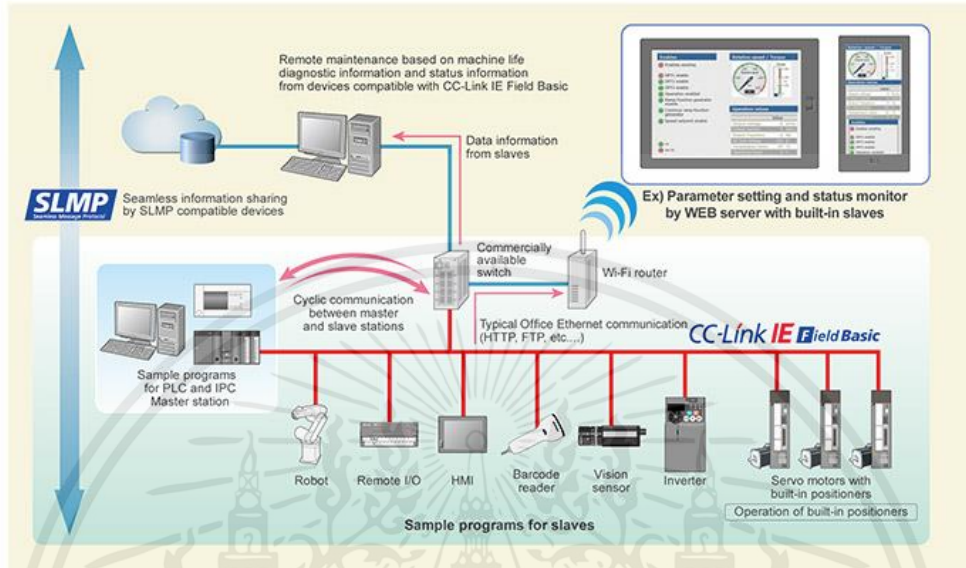
ทีซีพี/ไอพีคือโปรโตคอลสำหรับการใช้ในการสื่อสารผ่านระบบเครือข่ายอินเทอร์เน็ต โดยจุดประสงค์เพื่อให้สามารถทำการสื่อสารระหว่างเครื่องคอมพิวเตอร์ต้นทางและปลายทางได้ทีซีพี/ไอพีแบ่งงานสื่อสารออกเป็น 4 เลเยอร์ (Layers) ตามรูปที่ 2.17 แต่ละชั้นมีหน้าที่ต่างกัน โดยทีซีพี/ไอพีแยกข้อมูลออกเป็น ส่วน ๆ เรียกว่าแพ็กเก็ตและส่งแพ็กเก็ตเหล่านี้ผ่านเลเยอร์ทั้ง 4 ชั้น จากนั้นทีซีพี/ไอพีทำงานผ่านเลเยอร์ทั้ง 4 แบบย้อนกลับ และประกอบข้อมูลทั้งหมดเข้าด้วยกันก่อนแสดงข้อมูลนั้นให้แก่ผู้รับสาร

2.6.2 พื้นฐาน CC-Link IEF

พื้นฐาน CC-Link IEF คือการสื่อสารของ CC-Link IE ซึ่งการใช้อีเทอร์เน็ตเพื่องานอนเนกประสงค์บนเครือข่ายเขตข้อมูลสามารถใช้ได้กับอุปกรณ์ขนาดเล็ก การสื่อสารของ CC-Link IE โดยการใช้เทคโนโลยีอีเทอร์เน็ตเพื่องานอนเนกประสงค์สามารถใช้กับอุปกรณ์ขนาดเล็กได้ไม่จำเป็นต้องมีการควบคุมความเร็วสูงง่ายต่อการใช้งานและพัฒนา และสามารถสื่อสารแบบวนรอบบนเครือข่ายเขตข้อมูล CC-Link IE ได้ผ่านซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 14 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารแบบซิงโครนัสรอบตามที่กำหนด (Cyclic Communications) บนเครือข่าย CC-Link IE สามารถทำได้ด้วยการใช้ซอฟต์แวร์ สามารถพัฒนาระบบได้และมีอุปกรณ์ที่รองรับการใช้งานมากมายซึ่งสามารถจัดหาได้ก่อนขึ้นระบบ

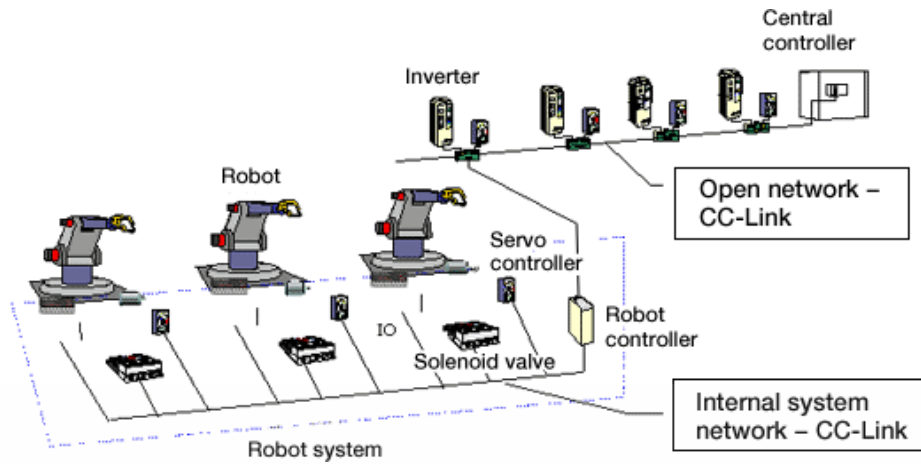


รูปที่ 2.18 ตัวอย่างโครงสร้างระบบพื้นฐาน CC-Link IE Field เชื่อมต่อกับอุปกรณ์ต่าง ๆ [10]

การสื่อสารสามารถทำควบคู่ไปพร้อม ๆ กันได้อย่างต่อเนื่องด้วยมาตรฐานการสื่อสารอีเทอร์เน็ต ทีซีพี/ไอพีไม่จำเป็นต้องทำการเดินสายสำหรับการควบคุมและสามารถร่วมเป็นเครือข่ายเดียวกันกับอีเทอร์เน็ตได้ดังแสดงในรูปที่ 2.18

2.6.3 CC-Link

CC-Link คือเครือข่ายเซตข้อมูลความเร็วสูงที่สามารถจัดการกับข้อมูลควบคุมและข้อมูลทั่วไปได้ในเวลาเดียวกันด้วยการสื่อสารความเร็วสูงขนาด 10 Mbps ทำให้ CC-Link สามารถส่งข้อมูลได้เป็นระยะทางสูงสุด 100 m และเชื่อมต่อได้ถึง 64 สเตชัน



รูปที่ 2.19 การเชื่อมต่อหุ่นยนต์จำนวนมากด้วย CC-Link [11]

การควบคุมระบบหุ่นยนต์หลายตัวดังรูปที่ 2.19 โดยใช้เครือข่าย CC-Link ทำให้โครงสร้างระบบ ต้นทุนต่ำเป็นไปได้ โดยเฉพาะอย่างยิ่งเมื่อประมวลผลวัตถุเป้าหมายควบคุมที่ซึ่งโครงในซบสายพานลำเลียง ตัวควบคุมหุ่นยนต์ง่ายขึ้นโดยใช้การควบคุมแบบรวมศูนย์ผ่านเครือข่าย กรณีศึกษาี้แสดงการใช้ CC-Link เป็นเครือข่ายพื้นที่ซึ่งใช้อุปกรณ์รุ่นต่าง ๆ จากผู้ผลิตหลายราย นอกจากนี้ CC-Link ยังใช้เพื่อเปิดใช้งานการ สื่อสารความเร็วสูงเป็นเครือข่ายระบบภายในสำหรับระบบหุ่นยนต์อีกด้วย

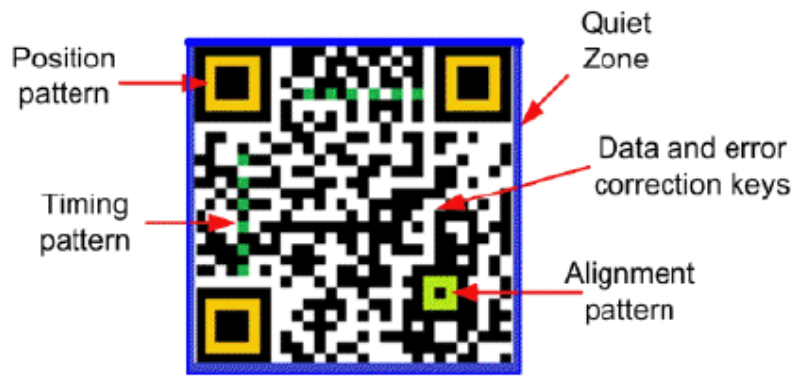
2.7 โค้ดรีดเดอร์

โค้ดรีดเดอร์ เป็นเซนเซอร์ทำหน้าที่อ่านรหัสแท่งและคิวอาร์โค้ดซึ่งคิวอาร์โค้ดย่อมาจาก Quick Response Code มีรูปร่างเป็นสี่เหลี่ยมมีสีขาวและสีดำ โดยโค้ดรีดเดอร์ส่งข้อมูลเข้าสู่ระบบซึ่งมี อัลกอริทึมในการถอดรหัสโค้ดชนิดต่าง ๆ และส่งข้อมูลไปทำงานร่วมกับอุปกรณ์อื่น ๆ ดังรูปที่ 2.20 นั้น แสดงให้เห็นถึงตัวอย่างโค้ดรีดเดอร์ของบริษัทเม็ลฟท์ซี อีเล็กทริก



รูปที่ 2.20 Code scanner ของบริษัท MELFT [12]

การสร้างคิวอาร์โค้ดนั้นเป็นการเข้ารหัสเปลี่ยนข้อมูลจากรูปแบบของตัวหนังสือให้กลายเป็นรูป โดยคิวอาร์โค้ดมีส่วนประกอบต่าง ๆ ยกตัวอย่างชนิด 2D การสร้างคิวอาร์โค้ดดังรูปที่ 2.21



รูปที่ 2.21 ส่วนประกอบของคิวอาร์โค้ด [13]

ในการสร้างคิวอาร์โค้ดทำการแปลง ASCII เป็นเลขฐานสองประกอบด้วยข้อมูลตัวเลขศูนย์กับหนึ่งเป็นค่าข้อมูลที่อยู่ในคิวอาร์โค้ดแสดงเป็นสีขาวดำ โดยความละเอียดขึ้นอยู่กับข้อมูลที่บรรจุอยู่ในคิวอาร์โค้ด

2.8 เว็บแอปพลิเคชัน

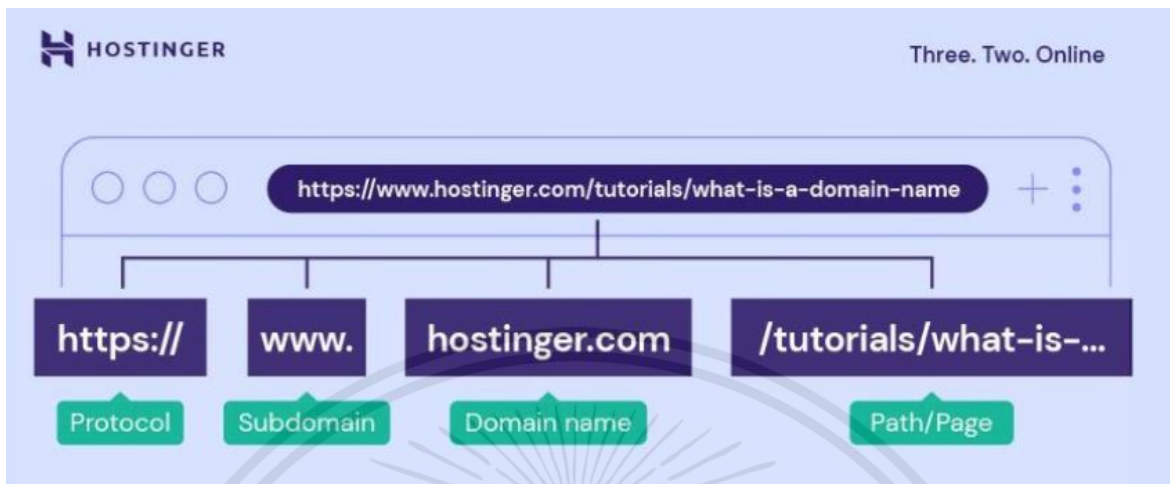
เว็บแอปพลิเคชันคือแอปที่ถูกเขียนขึ้นมาให้สามารถเปิดใช้เว็บเบราว์เซอร์ได้โดยไม่ต้องโหลดแอปพลิเคชันทำให้โดยรวมแล้วกินทรัพยากรค่อนข้างต่ำ ซึ่งแตกต่างจากเว็บไซต์ที่เน้นให้ผู้คนเข้ามาดูเป็นหลักแต่ เว็บแอปพลิเคชันเน้นการใช้ฟังก์ชันต่าง ๆ เพื่อให้ผู้คนเข้ามาใช้งานเป็นหลักโดยเว็บแอปพลิเคชันมีความซับซ้อนในการสร้างมากกว่าเว็บไซต์

2.8.1 ยูอาร์แอล

ยูอาร์แอล (Uniform Resource Locator : URL) คือตัวระบุที่อยู่ของข้อมูลต่าง ๆ บนอินเทอร์เน็ต ช่วยให้เข้าถึงไฟล์หรือหน้าเว็บเพจต่าง ๆ ซึ่งประกอบด้วยสองส่วนหลักได้แก่ ตัวระบุโปรโตคอล หรือ Scheme คือส่วนต้นของยูอาร์แอลเป็นส่วนที่ระบุโปรโตคอลที่ใช้ในการเข้าถึงเว็บไซต์นั้น ๆ โดยมีเซิร์ฟเวอร์เป็นตัวกลางรองรับข้อมูลในอินเทอร์เน็ตทำให้ผู้ใช้สามารถเข้าถึงข้อมูลต่าง ๆ ผ่านชื่อโดเมนได้โดยโปรโตคอลที่นิยมใช้ในปัจจุบันได้แก่ http และ https เป็นต้น อีกส่วนเรียกว่าตัวระบุที่มาหรือ Authority คือเป็นส่วนท้ายของยูอาร์แอลที่อยู่หลังเครื่องหมาย :// แบ่งองค์ประกอบย่อยภายในได้อีกหลายส่วนแล้วแต่โครงสร้างเว็บไซต์และหน้าเว็บเพจ ชื่อ โดเมน หรือ ดีเอ็นเอส หรือบางที่เรียกว่าชื่อโฮสต์เป็นองค์ประกอบสำคัญของยูอาร์แอลที่ช่วยให้ผู้ใช้ Path คือเส้นทางที่นำไปสู่ไฟล์หรือตำแหน่งต่าง ๆ ของข้อมูลบนเว็บไซต์โดย Path อยู่ด้านหลังของชื่อโดเมนเสมอและคั่นเครื่องหมายทับ (/) ตัวอย่างเช่น <https://www.mitsubishifa.co.th/th/Solution.php> โดยแยกส่วนประกอบของยูอาร์แอลได้ตามรูปที่ 2.22

ส่วนท้ายของ Path อาจจะไม่มีการมีหรือบางหน้าเว็บเพจอาจจะปิดท้ายด้วยนามสกุล Query String ซึ่งหมายถึงการค่าหรือเพิ่มตัวแปรเพื่อใช้ในการเข้าถึงข้อมูลที่ลึกหรือเฉพาะมากขึ้นบนเว็บไซต์เปรียบเสมือนเป็นการกรองข้อมูลให้แยกย่อยลงไปอีกตามที่ผู้ใช้ต้องการ มักปรากฏเครื่องหมาย “?” ตามด้วย

ค่าพารามิเตอร์ เช่น สีหรือขนาดของสินค้าบางอย่าง ซึ่งหากมีพารามิเตอร์มากกว่าหนึ่งค่ายูอาร์แอลแสดงค่าเป็นคู่โดยใช้เครื่องหมาย "&" คั่นระหว่างค่าและแบ่งคู่ด้วยเครื่องหมาย "="



รูปที่ 2.22 รูปแบบยูอาร์แอล [14]

2.8.2 เฟรมเวิร์ก

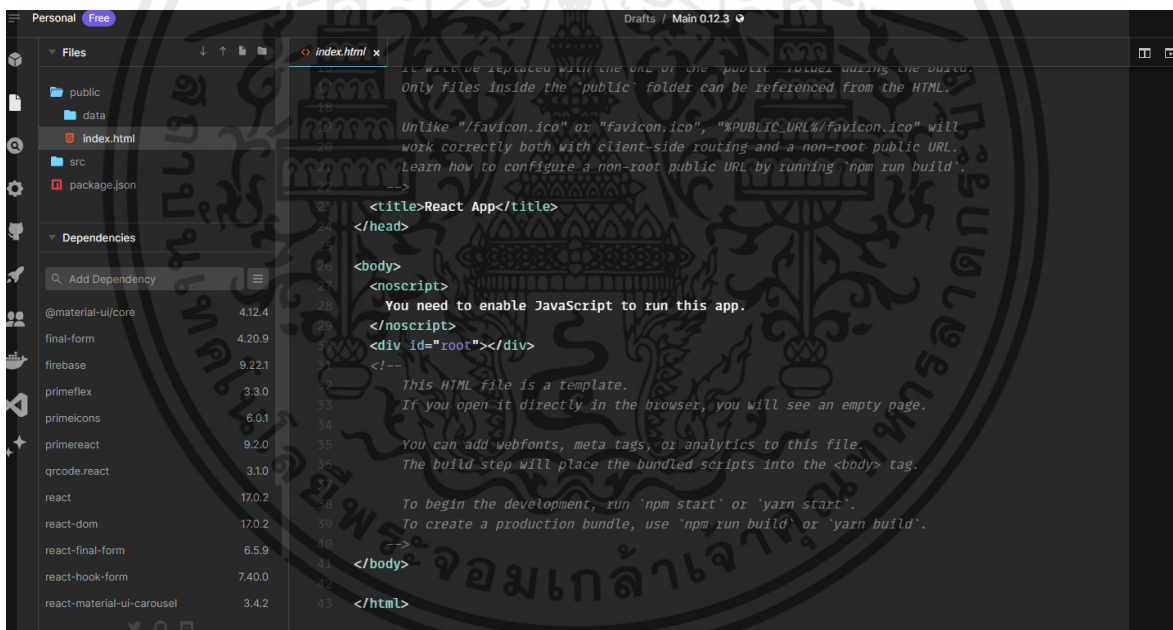
เฟรมเวิร์กหมายถึง ชุดคำสั่ง เครื่องมือ หรือโครงสร้างอย่างใดอย่างหนึ่งที่สร้างขึ้นมาเพื่ออำนวยความสะดวกแก่ผู้ใช้งานเหมาะกับการเขียนเว็บแอปพลิเคชัน ซึ่งการสร้างเว็บแอปพลิเคชันสามารถสร้างได้จากหลายภาษาด้วยกันและแต่ละภาษามีการใช้งานต่างกันและเฟรมเวิร์กที่ไม่เหมือนกัน เช่น หากเลือก JavaScript เป็นภาษาที่ต้องการเขียนก็อาจใช้ React VueJS Angular jQuery เป็นเฟรมเวิร์ก หากเลือก PHP เป็นภาษาที่ต้องการเขียนก็อาจใช้ Laravel Laminas Symfony CodeIgniter CakePHP เป็นเฟรมเวิร์กซึ่งในปริญญาโทจำเป็นต้องเลือกใช้ JavaScript เป็นภาษาสำหรับการเขียนโปรแกรมเนื่องจากมีชุมชนรองรับในการแก้ข้อผิดพลาดต่าง ๆ ช่วยให้แก้ปัญหาได้ง่ายขึ้นโดยเครื่องมือภาษาสำหรับการพัฒนาเว็บที่เป็นพื้นฐานหากใช้ JavaScript ในการพัฒนาเว็บได้แก่

JavaScript ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง JavaScript เป็นภาษาสคริปต์เชิงวัตถุเรียกกันว่า "สคริปต์" ซึ่งในการสร้างและพัฒนาเว็บใช้ร่วมกับ HTML เพื่อให้เว็บมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปที่ละคำสั่ง" สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์และทางฝั่งเซิร์ฟเวอร์

HTML ย่อมาจากคำว่า Hypertext Markup Language เป็นภาษาหลักที่ใช้ในการสร้างไฟล์เว็บเพจ โดยมีแนวคิดจากการสร้างเอกสารไฮเปอร์เท็กซ์เป็นภาษามาตรฐานที่ใช้พัฒนาเอกสารในรูปแบบของเว็บเพจเผยแพร่บนระบบเครือข่ายอินเทอร์เน็ต

CSS ย่อมาจากคำว่า Cascading Style Sheets เป็นคำสั่งที่ใช้คู่กับภาษา HTML เพื่อให้การ จัดรูปแบบของเว็บเพจได้ดีขึ้นและแก้ไขได้ง่ายขึ้นในหมวดนี้สามารถใช้ CSS เพื่อเปลี่ยนรูปแบบหน้าเว็บเพจ เป็นการสร้างหน้าตาให้เว็บมีหน้าตาที่สวยงามขึ้น

เมื่อทำการสร้าง “Project” ขึ้นเกิดไฟล์หลักขึ้นมาสามไฟล์ดังแสดงในรูปที่ 2.23 ได้แก่ ไฟล์ public ใช้สำหรับเก็บไฟล์ที่เป็นสาธารณะและสามารถเข้าถึงได้โดยตรงจากภายนอก ตัวอย่าง เช่น ไฟล์ HTML CSS รูป และไฟล์อื่น ๆ ที่ต้องการให้เปิดเผยต่อสาธารณะในการพัฒนาเว็บแอปพลิเคชัน ไฟล์ src ย่อมาจาก “source” เป็นที่เก็บไฟล์ต้นฉบับของโปรแกรมหรือแอปพลิเคชัน ภายในไฟล์ src ประกอบไปด้วยโค้ด JavaScript CSS รูป และไฟล์อื่น ๆ ที่เกี่ยวข้องกับการพัฒนาและการทำงานของโปรแกรมหรือแอปพลิเคชัน ไฟล์ package.json เป็นไฟล์ที่ใช้ในการกำหนดและบรรจุข้อมูลที่เกี่ยวข้องกับโปรเจกต์และ แพ็กเกจที่ใช้ในโปรเจกต์นั้นๆ ไฟล์นี้ระบุรายละเอียดเกี่ยวกับชื่อโปรเจกต์ เวอร์ชัน คำอธิบาย คุณสมบัติ สคริปต์สำหรับการสร้าง รายการแพ็กเกจที่จำเป็น สคริปต์ที่จำเป็นในการทดสอบและอื่น ๆ ไฟล์นี้มีความสำคัญในการจัดการและติดตั้งแพ็กเกจที่ใช้ในโปรเจกต์และให้ข้อมูลเพิ่มเติมเกี่ยวกับโปรเจกต์โดยส่วน index.js ที่อยู่ใน public ทำการดึงข้อมูลส่วน root



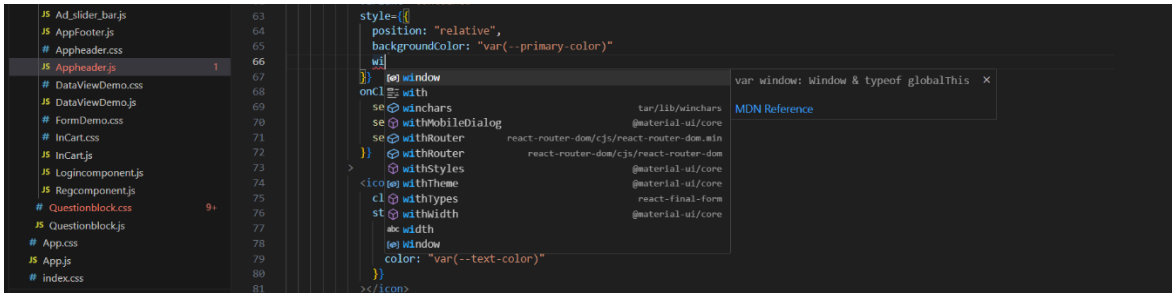
รูปที่ 2.23 ตัวอย่างโปรแกรมใน index.html

2.9 แพลตฟอร์มและเครื่องมือพัฒนาเว็บแอปพลิเคชัน

2.9.1 Visual Studio Code

เป็นโปรแกรมแก้ไขซอร์สโค้ดที่พัฒนาโดยไมโครซอฟท์สำหรับ Windows Linux และ macOS สนับสนุนในตัวสำหรับ JavaScript TypeScript และ Node.js และมีระบบนิเวศที่สมบูรณ์ทั้งในส่วนขอ ส่วนขยายสำหรับภาษาอื่น และยังสามารถขยายความสามารถของโปรแกรมผ่านส่วนขยาย เพิ่มคุณลักษณะเพิ่มเติมช่วยเพิ่มประสบการณ์โดยรวมและความง่ายในการใช้โปรแกรมส่วนต่างๆ ดังรูปที่ 2.24

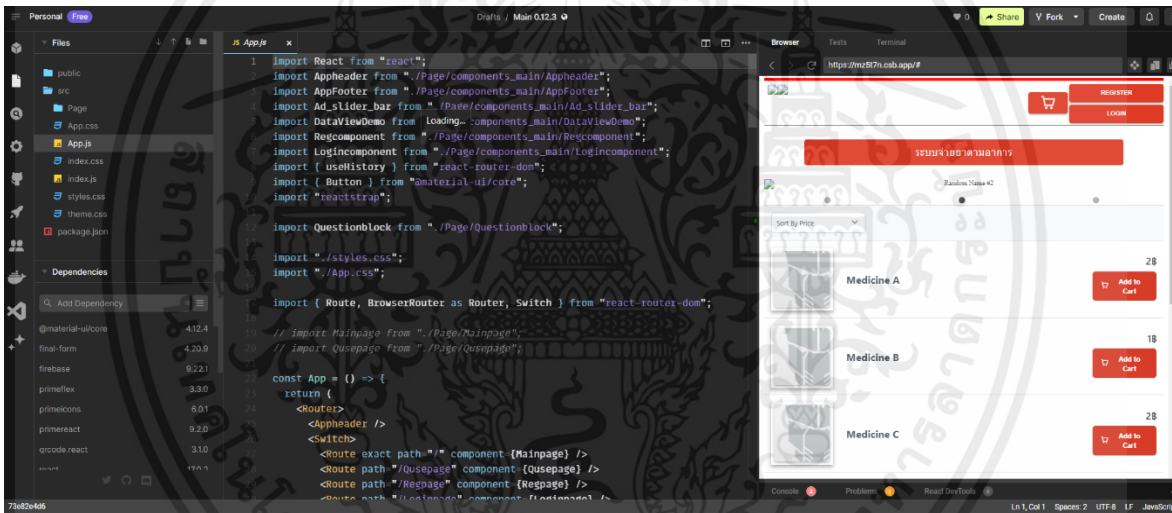
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 19 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 Visual Studio Code

2.9.2 Codesandbox.io

Codesandbox.io คือเป็นเว็บสำหรับทดสอบเขียนโปรแกรมทั้งเว็บแอปพลิเคชันและโปรแกรมอื่นๆ มีหลายเฟรมเวิร์ก เช่น React Javascript Vue Svelte Angular เป็นต้น สามารถทดสอบเขียนโปรแกรมบนเว็บ Codesandbox.io ได้เลยโดยไม่ต้องติดตั้งโปรแกรมเริ่มต้นในคอมพิวเตอร์ที่เขียนโปรแกรมอยู่เองหรือไม่ต้องติดตั้งเฟรมเวิร์กเองเพิ่มความสะดวกในการใช้งานดังรูปที่ 2.25

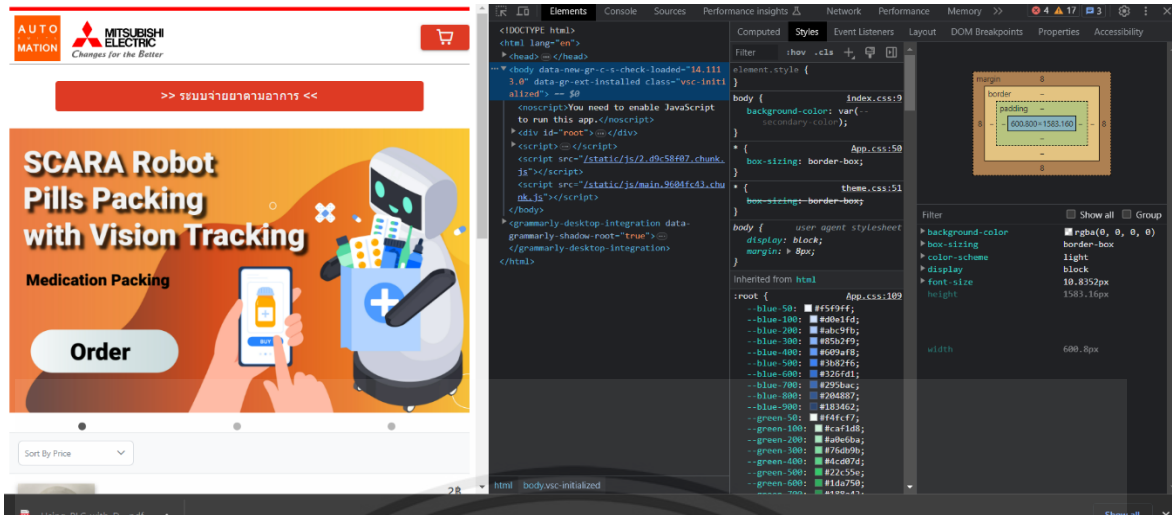


รูปที่ 2.25 ตัวอย่างการใช้งาน Codesandbox.io

2.9.3 Google Developer Tool

Google Developer Tool เป็นเครื่องมือของ Google Chrome สามารถช่วย Debug หรือดูองค์ประกอบต่าง ๆ ของเว็บได้โดยการกด F12 เพื่อดูรายละเอียดทั้งคอนโซลในการแสดงผลของโปรแกรมหรือ Element ซึ่งหมายถึงส่วนประกอบเล็กที่สุดที่ประกอบด้วย HTML ของหน้าเว็บช่วยในการปรับแก้หน้าเว็บได้เร็วและง่ายขึ้นโดยรวมจนไปถึงหาข้อผิดพลาดของโปรแกรมได้ดังรูปที่ 2.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 20 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 ตัวอย่างการใช้งาน Google Developer Tool

2.10 เว็บไซต์ตั้ง

เว็บไซต์ตั้งคือรูปแบบการให้บริการสำหรับผู้ใช้อินเทอร์เน็ตอย่างหนึ่ง โดยผู้ใช้งานสามารถฝากเว็บไซต์หรือเว็บแอปพลิเคชันของตนเองกับผู้ให้บริการเซิร์ฟเวอร์ (HSP: Hosting Service Provider) เพื่อให้เว็บไซต์ของตนเองนั้นออนไลน์อยู่บนโลกอินเทอร์เน็ตตลอดเวลา โดยที่ทางผู้ให้บริการจัดเก็บข้อมูลเว็บไซต์ ฐานข้อมูล อีเมล ฯลฯ ที่เครื่องเซิร์ฟเวอร์หรือที่เรียกกันว่า “เว็บเซิร์ฟเวอร์” ซึ่งเว็บเซิร์ฟเวอร์ทำหน้าที่เป็นสื่อกลางที่แสดงผลหน้าเว็บไซต์ให้กับผู้ท่องอินเทอร์เน็ตทั่วไปได้เข้าชมผ่านโดเมนเนม ดังนั้นผู้ที่ต้องการออนไลน์เว็บไซต์ของตนเองจึงต้องคำนึงถึงเว็บเซิร์ฟเวอร์ก่อน แต่ด้วยความที่เว็บเซิร์ฟเวอร์นั้นมีราคาค่อนข้างสูงและจำเป็นต้องมีผู้เชี่ยวชาญที่คอยดูแลเซิร์ฟเวอร์ในด้านเทคนิคต่างๆ อีกทั้งทำให้เจ้าของเว็บไซต์ส่วนใหญ่จึงหันมาใช้บริการ เว็บไซต์ตั้ง เนื่องจากค่าใช้จ่ายน้อยกว่ามาก

Firebase คือแพลตฟอร์มที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วนของระบบจัดการเว็บไซต์หรือฝั่งเซิร์ฟเวอร์ซึ่งทำให้สามารถสร้างแอปพลิเคชันได้อย่างมีประสิทธิภาพ และยังลดเวลาและค่าใช้จ่ายของการทำฝั่งเซิร์ฟเวอร์หรือการวิเคราะห์ข้อมูลให้อีกด้วย โดยมีทั้งเครื่องมือที่ฟรีและเครื่องมือที่มีค่าใช้จ่ายโดย Firebase Hosting เป็นบริการหนึ่งของ Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 21 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

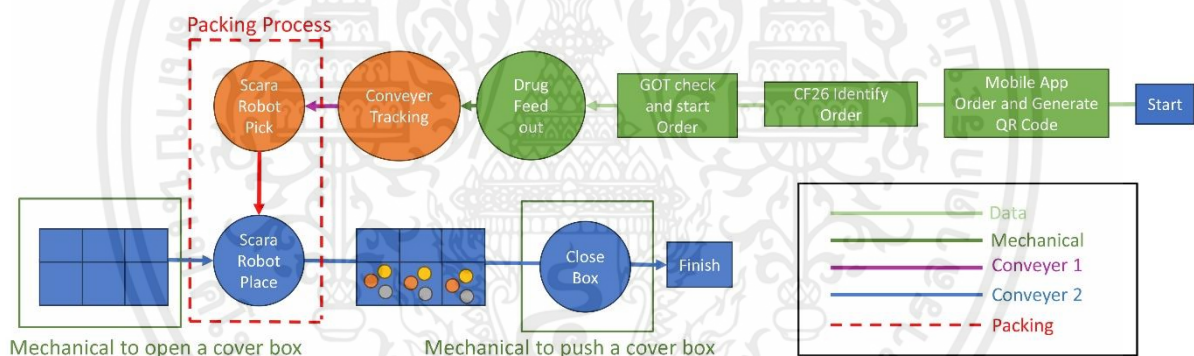
การออกแบบฟังก์ชันการทำงานและการเขียนโปรแกรมสำหรับชุดสาธิต

3.1 กล่าวนำ

ในบทนี้กล่าวถึงกระบวนการออกแบบและทดลองแนวคิดที่ได้ออกแบบ รวมถึงขั้นตอนการสร้างโปรแกรมและเว็บแอปพลิเคชันซึ่งเกี่ยวกับการสร้างชุดสาธิตตั้งแต่การวางแผนเลือกใช้พีแอลซี GOT หุ่นยนต์สการา และวิชชันเซนเซอร์ และการเขียนโปรแกรมด้วยซอฟต์แวร์ต่าง ๆ ของมิตซูบิชิ อิเล็กทริก เช่น GX Work GT Designer3 และ RT tool box 3

3.2 แนวคิดการออกแบบชุดสาธิต

แนวคิดเกี่ยวกับชุดสาธิตเริ่มแรกคือการแสดงศักยภาพของหุ่นยนต์สการาของมิตซูบิชิ อิเล็กทริก โดยแนวคิดเกี่ยวกับการใช้หุ่นยนต์สการาบรรจุลงในกล่องยาแบบรายวัน มีการทำงานของระบบดังรูปที่ 3.1 เริ่มจากสั่งยาบนแอปพลิเคชันบนมือถือ แล้วนำคิวอาร์โค้ดไปสแกนที่โค้ดรีดเดอร์ของ MELFT ชื่อรุ่น CF26 จากนั้นโค้ดรีดเดอร์ทำการส่งข้อมูลชุดตัวเลขไปยังพีแอลซีเพื่อสั่งการให้หุ่นยนต์สการาทำการหยิบยาจากสายพานลำเลียงที่ทำการปล่อยยาออกมาตามลำดับไปใส่ที่กล่องใส่ยาที่เปิดรอไว้



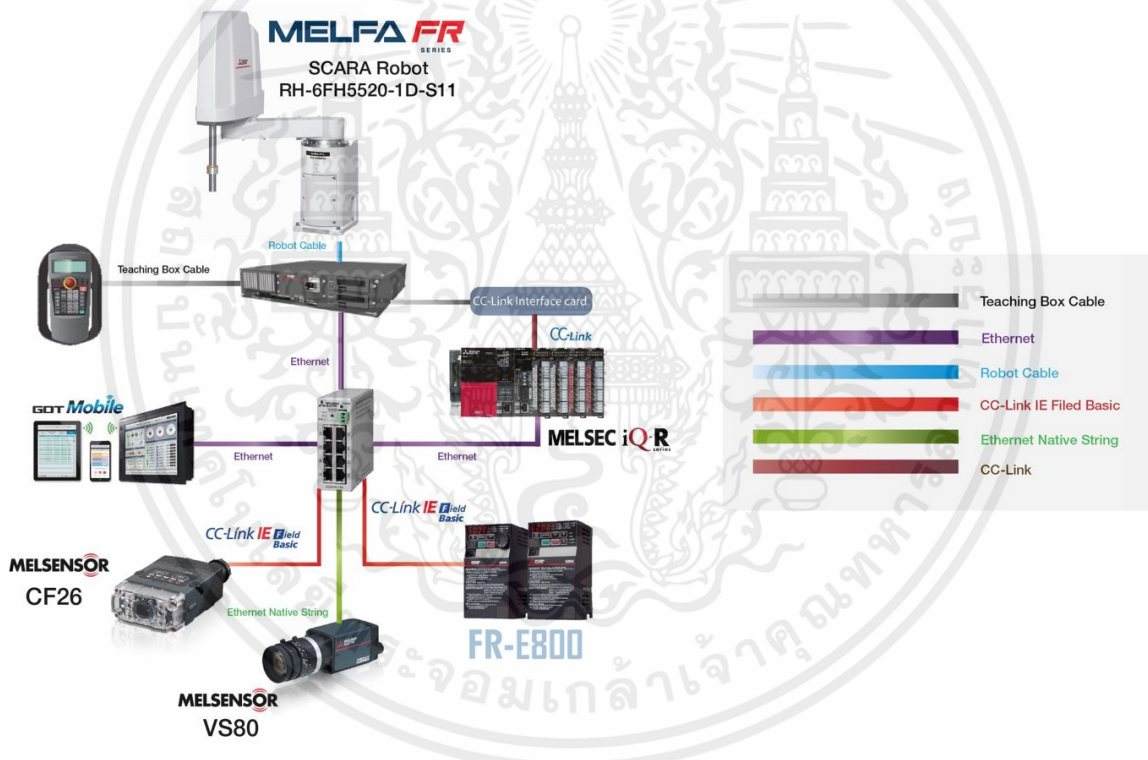
รูปที่ 3.1 การทำงานของแนวคิดเริ่มแรกของชุดสาธิต

ต่อมาหลังจากปรึกษากับบริษัทผู้รวมระบบถึงความเป็นไปได้ในการทำกลไกปล่อยยาและกลไกจัดเรียงยาให้เรียงตรงเป็นแนวในช่วงของการปล่อยยาและยาเริ่มออกจากจุดปล่อยยาและใช้เซนเซอร์อีกตัวติดตั้งบนสายพานเพื่อตรวจจับเม็ดยา ได้ความเห็นว่าไม่สามารถจัดเรียงตัวยายู่ตำแหน่งเดียวกันตลอดได้ เนื่องจากตัวยามีขนาดเล็กและขนาดไม่เท่ากัน จึงมีแนวคิดให้มีการนำวิชชันเซนเซอร์มาเป็นตัวเลือกในการจัดการปัญหา ทำให้เกิดการเปรียบเทียบระหว่างการใช้วิชชันเซนเซอร์และไม่ใช้วิชชันเซนเซอร์เพื่อหาตัวเลือกที่ดีที่สุด มีข้อดีข้อเสียดังตารางที่ 3.1

ตารางที่ 3.1 ข้อดีและข้อเสียของการใช้วิชันเซนเซอร์

วิชันเซนเซอร์	ข้อดี	ข้อเสีย
ไม่ใช่	ไม่ต้องกังวลเรื่องแสงที่ส่งผลการถ่ายภาพเมื่อเปลี่ยนสถานที่จัดแสดง	ต้องใช้กลไกที่มีขนาดใหญ่เพื่อให้ยาเรียงกันเป็นเส้นตรงออกมาตามลำดับที่ต้องการ
ใช่	ไม่ต้องมีกลไกเพื่อให้ยาเรียงกันเป็นเส้นตรงและตามลำดับ	ในการเปลี่ยนสถานที่จัดแสดงแสงอาจจะส่งผลการถ่ายภาพ

จากการเปรียบเทียบได้ทำการเลือกวิชันเซนเซอร์เข้ามาใช้เพื่อลดความซับซ้อนของท่ากลไกต่างๆ และนำเสนอสินค้าอย่างวิชันเซนเซอร์ของมิตซูบิชิ อิเล็กทรอนิกส์เพิ่ม ส่วนปัญหาเรื่องแสงมีการนำแฟลชมาใช้เพื่อช่วยลดปัญหาจากแสงที่เปลี่ยนแปลงไปจากการเปลี่ยนสถานที่จัดแสดง การเลือกใช้อุปกรณ์วิชันเซนเซอร์มีโครงสร้างโดยรวมของระบบดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างโดยรวมของระบบที่ใช้วิชันเซนเซอร์

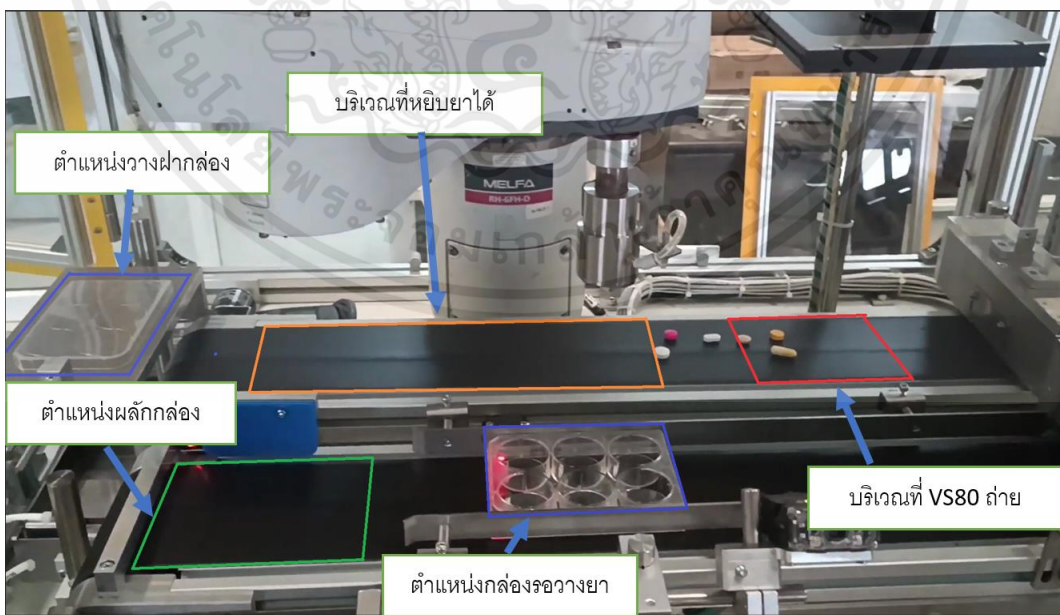
การทำงานของชุดสาธิตแบบมีวิชันเซนเซอร์ติดตั้งที่ปลายแขนหุ่นยนต์ดังรูปที่ 3.3 มีการทำงานโดยรวมของชุดสาธิตคล้ายกับแนวคิดก่อนหน้า แต่เปลี่ยนจากปล่อยยาตามลำดับ จัดเรียงให้เรียบร้อยแล้วให้เซนเซอร์ระบุตำแหน่งที่ละเม็ดเป็นปล่อยยาออกแบบสุ่มลงมาแบบกระจายแล้วให้หุ่นยนต์สาธิตที่มีวิชันเซนเซอร์ที่ติดตั้งที่ปลายแขนของหุ่นยนต์เคลื่อนที่ไปยังที่ตำแหน่งที่กำหนดเพื่อทำการถ่ายภาพ หลังจากถ่ายภาพวิชันเซนเซอร์ทำการจำแนกยาที่หยุดนิ่งบนสายพานแล้วส่งตำแหน่งให้หุ่นยนต์ทำการหยิบยาใส่กล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและ 23 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 การทำงานของชุดสายิตที่ใช้เอ็นโค้ดเดอร์ร่วมกับวิชันเซนเซอร์

การนำเอ็นโค้ดเดอร์มาใช้ร่วมกับวิชันเซนเซอร์จึงทำให้สามารถใช้งานวิชันแทรกคิงซึ่งหมายถึงฟังก์ชันคอนเวเยอร์แทรกคิงที่นำวิชันเซนเซอร์มาใช้ระบุตำแหน่งเริ่มต้นในการติดตามตำแหน่งของชิ้นงาน มีการเปลี่ยนการทำงานให้เมื่อมีคำสั่งซื้อเข้ามาทำการปล่อยยาตามจำนวนที่สั่งพร้อมกันแบบกระจายกระจายบนสายพาน เมื่อสายพานเคลื่อนที่นำตัวยาออกจากจุดปล่อยยา วิชันเซนเซอร์ทำการถ่ายรูปในพื้นที่ถ่ายรูปเมื่อยาเคลื่อนที่ถึงพื้นที่ของการหยิบยาหุ่นยนต์ทำการหยิบยาที่กำลังเคลื่อนที่บนสายพานไปใส่กล่องที่เปิดฝาไว้แล้วดังรูปที่ 3.6



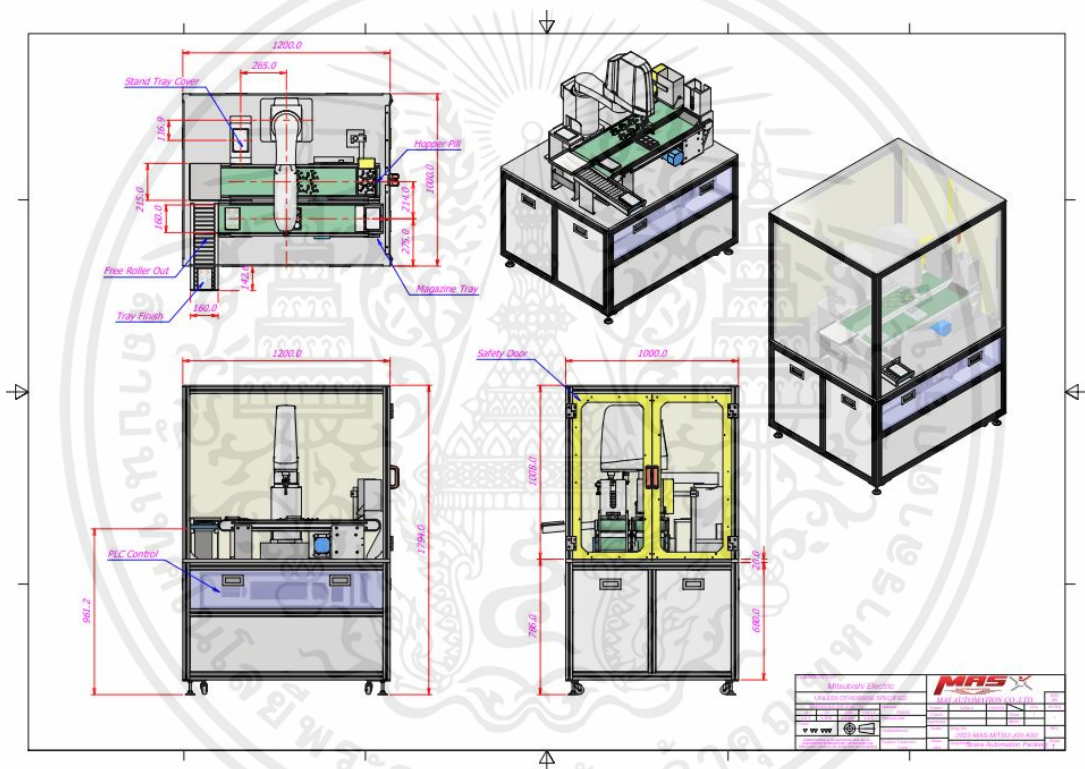
รูปที่ 3.6 ภาพรวมการทำงานของชุดสายิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 25 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนประกอบของชุดสาธิต

3.3.1 โครงสร้างเครื่องชุดสาธิตโดยรวม

โครงสร้างโดยรวมมีการปรับเปลี่ยนจากการ CAD (Computer Aided Design) ครั้งแรกแต่มีขนาดชิ้นส่วนต่าง ๆ โดยรวมเท่าเดิมโดยมีการเปลี่ยนที่ให้กับวิชั่นเซนเซอร์หรือหมายถึง VS80 ย้ายไปเข้าหาตัวหุ่นยนต์สกราราและเพิ่มโค้ดรีดเดอร์หรือหมายถึง CF26 ที่หน้าตู้ในกระจกเพิ่มที่ยึดหน้าจอเอชเอ็มไอหรือหมายถึง GOT2000 หรือ GOT ที่หน้าตู้ ทำการปรับให้ Stand Tray Cover ดังที่แสดงในรูปที่ 3.7 ย้ายให้อยู่ที่เดียวกับจุดรับปลายสายพานและปรับให้จุดรับยาปลายทางอยู่ต่ำลงแทน Free Roller Out ย้ายไปอยู่ข้างสายพานทางจุดปล่อยกล่องและเพิ่มตัวดันกล่องสำหรับการส่งกล่องบรรจุยาออกนอกตู้บรรจุยา ปรับแก้แบบ Hopper Pill เป็นแบบดีดยาออกที่ละเม็ดด้วยระบบนิวเมติกส์



รูปที่ 3.7 ต้นแบบและขนาดของชุดสาธิต

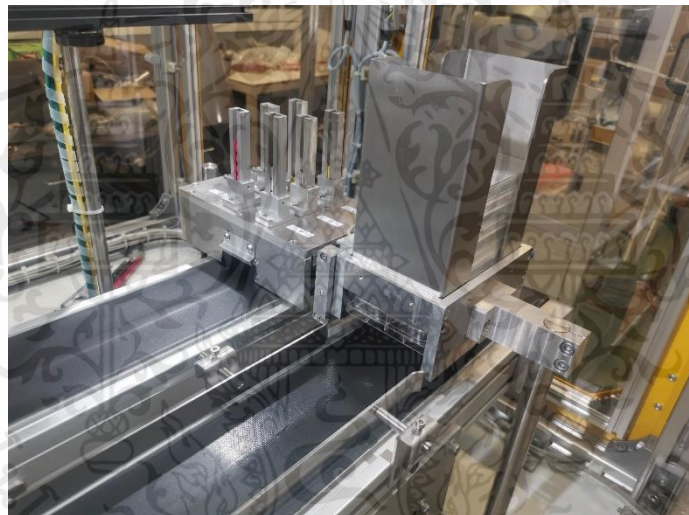
การทำงานของเครื่องเมื่อกดเริ่มการทำงานมีการทำงานดังนี้ ชุดสาธิตทำการปล่อยกล่องออกมารอบริเวณจุดพักกล่องและหุ่นยนต์สกราราอยู่ในตำแหน่งเริ่มต้นเพื่อรอสำหรับการทำงานตามคำสั่งผู้ใช้ที่สั่งมา โดยในปริณญาณินพจน์นี้เรียกคำสั่งของผู้ใช้ว่า “คำสั่งซื้อ” เมื่อกดสั่งคำสั่งซื้อเข้าไปเริ่มการทำงานจากการให้หุ่นยนต์สกราราหยิบยาไปวางในตำแหน่งวางฝากกล่องพร้อมกับพีแอลซีส่งปล่อยยาออกมาที่ละชุดโดยในหนึ่งชุดมียาชนิดละหนึ่งเม็ดตามคำสั่งซื้อที่สั่งและทำการปล่อยเรื่อยๆจนครบตามจำนวนคำสั่งซื้อ เมื่อวิชั่นเซนเซอร์เจอยาหุ่นยนต์สกราราเข้าสู่ตำแหน่งรอหยิบยาแล้วเมื่อยาเคลื่อนที่เข้าจุดหยิบยาหุ่นยนต์สกราราหยิบยาใส่กล่องทันทีเมื่อมียาในบริเวณที่สามารถหยิบยาได้ โดยให้หุ่นยนต์สกราราหยิบยาที่มีบนสายพานลงกล่องตามช่องที่จัดไว้จนครบตามจำนวนคำสั่งซื้อหากไม่ครบในรอบแรกทำการปล่อยยาซ้ำแล้วหยิบใหม่อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 26 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รอบ เมื่อหยิบครบแล้วหุ่นยนต์สกร่าไปหยิบฝามาปิดกล่องจากนั้นแสดงผล “Complete” ที่หน้าจอเอชเอ็มไอแต่หากไม่สามารถหยิบครบได้ในสามรอบทำการปิดฝาทันทีแล้วขึ้น “Fail” ที่หน้าจอเอชเอ็มไอ หลังจากหุ่นยนต์สกร่าปิดฝากล่องยาสายพานลำเลียงกล่องทำการเคลื่อนที่เพื่อนำกล่องไปยังจุดผลักออก ระหว่างนั้นกล่องใหม่ถูกปล่อยออกมาเพื่อมารอในตำแหน่งที่พร้อมสำหรับวางยาและหุ่นยนต์สกร่ากลับตำแหน่งเริ่มต้น

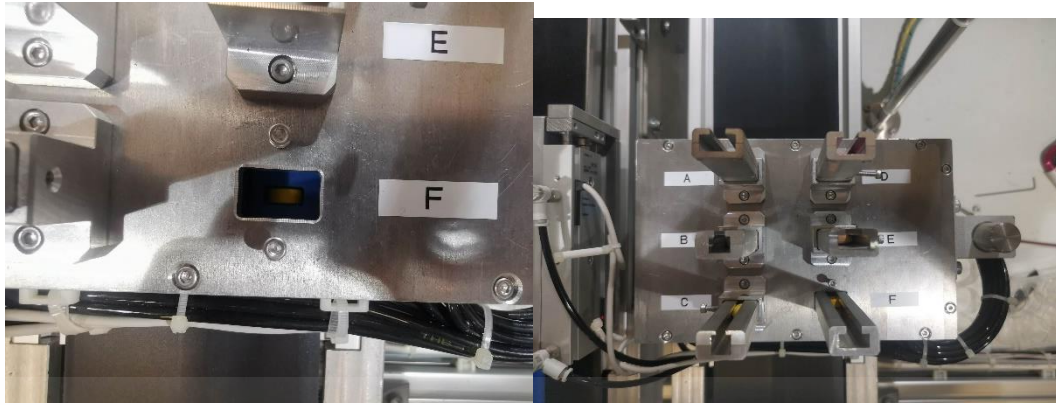
3.3.2 กลไกของการปล่อยยาและกล่องบรรจุยา

ส่วนปล่อยตัวยาและกล่องมีภาพรวมดังรูปที่ 3.8 โดยกลไกของส่วนนี้ใช้ระบบนิวแมติกส์เหมือนกัน โดยใช้พีแอลซีส่งคำสั่งควบคุมการทำงานของกระบอกลมโดยส่วนปล่อยยามีกระบอกลม 6 ตัวตามจำนวนยา 6 ชนิดโดยมีตำแหน่งในพีแอลซีเป็น Y30 ถึง Y35 และส่วนปล่อยกล่องเป็นกระบอกลมแบบมีเซนเซอร์ โดยกระบอกสูบทั้งหมดสามารถเพิ่มแรงดันได้ที่ตัววาล์วบนตัวแอกชูเอเตอร์ (actuator) ด้วยการหมุนที่วาล์ว



รูปที่ 3.8 ตัวปล่อยยาและปล่อยกล่อง

กลไกปล่อยยาภายในเป็นไปตามรูปที่ 3.9 การปล่อยยาเป็นแบบปล่อยยาออกมา 6 ชนิดชนิดละ 1 เม็ด ส่วนที่ใช้จัดเก็บยาเป็นแมกาซีนมีการทำงานเหมือนกับแมกาซีนของปืน ส่วนล่างสุดของที่บรรจุแมกาซีนทุกชนิดมีช่องเท่าขนาดยาทั้งความกว้าง ยาวและสูงเท่ากับขนาดยาในแต่ละชนิดเพื่อให้ตัวยาสามารถตกลงไปในช่องพอดีซึ่งมีฐานรองรับอยู่ ยาที่ถูกใส่เข้ามาเม็ดแรกจะลงไปยังช่องด้านล่างสุดและเม็ดต่อไปจะอยู่ในแมกาซีน เมื่อแอกชูเอเตอร์ทำงานแอกชูเอเตอร์ทำการดันตัวช่องออกไปในพื้นที่ที่ไม่มีฐานรองซึ่งทำให้ตัวยาหล่นลงไปยังสายพานและเมื่อแอกชูเอเตอร์หยุดทำงานตัวกระบอกสูบทำการดึงกลับและยาเม็ดต่อไปที่อยู่ในแมกาซีนหล่นลงช่องพร้อมสำหรับการปล่อยยาต่อไป



รูปที่ 3.9 กลไกปล่อยยาจากด้านบน

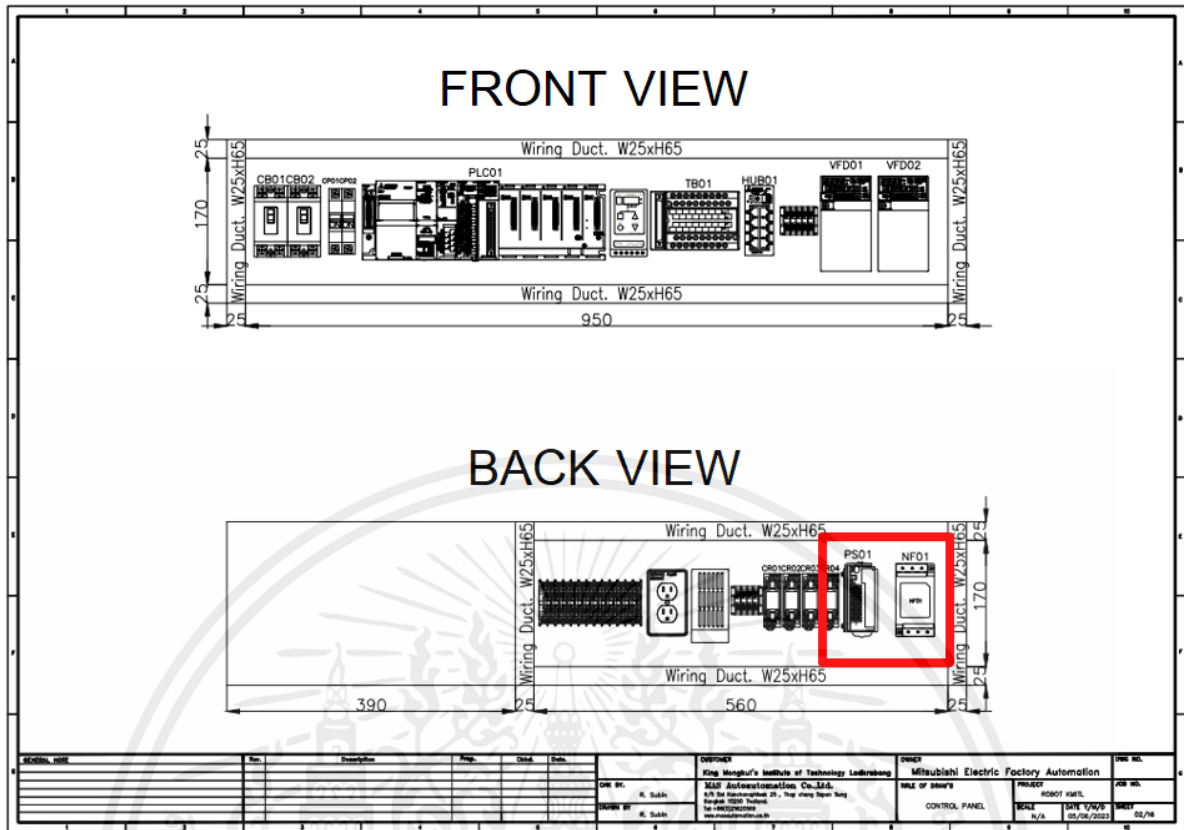
กลไกการปล่อยกล่องเมื่อกระบอกสูบได้รับสัญญาณแผ่นสีน้ำเงินในรูปที่ 3.10 ทำการดันตัวกล่องออก จากนั้นเมื่อกระบอกสูบไม่ได้รับสัญญาณทำการดึงตัวเองกลับแล้วกล่องถัดไปตกลงมาแทนตำแหน่งกล่องเก่าที่ถูกผลักออกไป



รูปที่ 3.10 รูปด้านหลังของกลไกปล่อยกล่อง

3.3.3 อุปกรณ์ควบคุม

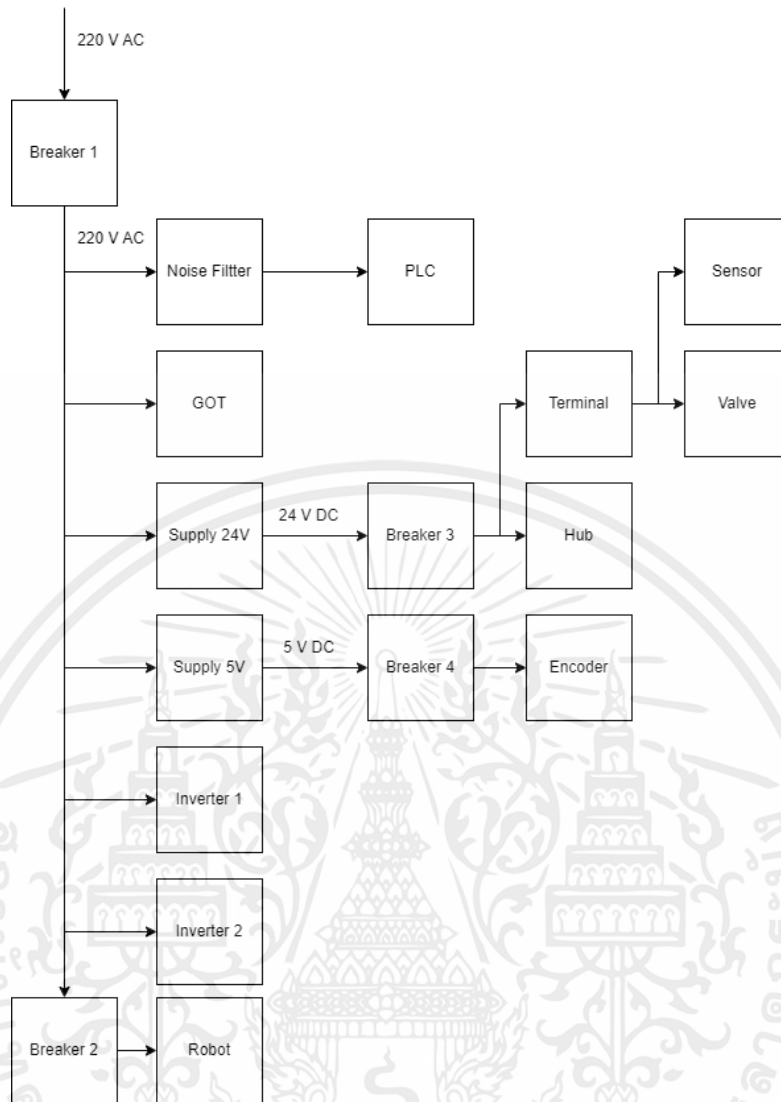
แผงการควบคุมหน้าตู้ขึ้นประกอบไปด้วยอุปกรณ์ตามรูปที่ 3.11 โดยมีเบรกเกอร์ส่วน Back view เพิ่มขึ้นโดยอยู่ระหว่าง PS01 และ NF01 เพิ่มขึ้นมาเพื่อควบคุมไฟรอบตัวหุ่นยนต์สการาและอุปกรณ์อื่น ๆ



รูปที่ 3.11 แผงควบคุมของชุดสาธิต

3.3.4 การเดินสายไฟฟ้า

ในส่วนนี้ทางบริษัทผู้รวมระบบเป็นผู้ออกแบบและเชื่อมต่อให้ มีการเชื่อมต่อคร่าว ๆ ดังรูปที่ 3.12



รูปที่ 3.12 แบบการจ่ายไฟอย่างง่าย

3.4 การตั้งค่าอุปกรณ์สำหรับการเชื่อมต่ออุปกรณ์เข้าด้วยกัน

3.4.1 การส่งข้อมูลระหว่างพีแอลซีกับอุปกรณ์อื่น

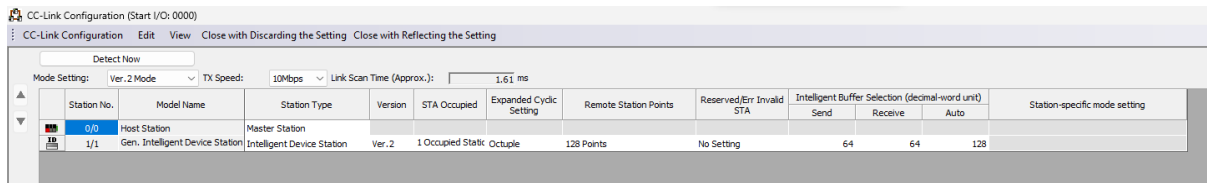
Setting Item	Item	Setting
Station Type	Station Type	Master Station
Mode	Communication Mode	Remote Net Ver.2 Mode
Station No.	Station No.	0
Transmission Speed	Transmission Speed	10Mbps
Parameter Setting Method	Setting Method of Basic/Application Settings	Parameter Editor

รูปที่ 3.13 การตั้งค่า Communication Mode ของ CC-Link

การตั้งค่าพารามิเตอร์ในซอฟต์แวร์ GX Work3 สำหรับการทำให้พีแอลซีสามารถส่งข้อมูลกับหุ่นยนต์ด้วยโปรโตคอล CC-Link ได้ ทำการตั้งค่า Communication Mode เป็น Remote Net Ver.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 30 อังอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode ตามรูปที่ 3.13 ซึ่งมีความแตกต่างจาก Remote Net Ver.1 Mode ตรงที่สามารถใช้งาน Point ได้มากขึ้น



รูปที่ 3.14 การตั้งค่า CC-Link Network Configurator

ทำการเลือก Intelligent Device Station มาใส่และตั้งค่าตามรูปที่ 3.14

No.	Link Side						CPU Side				
	Device Name	Points	Start	End	Target		Device Name	Points	Start	End	
-	SB	256	00000	000FF	↔	Specify Devi	SB	256	00000	000FF	
-	SW	256	00000	000FF	↔	Specify Devi	SW	256	00000	000FF	
1	RX	256	00000	000FF	↔	Specify Devi	X	256	02000	020FF	
2	RY	256	00000	000FF	↔	Specify Devi	Y	256	02000	020FF	
3	RWr	256	00000	000FF	↔	Specify Devi	W	256	01200	012FF	
4	RWw	256	00000	000FF	↔	Specify Devi	W	256	01300	013FF	
5					↔						

รูปที่ 3.15 การตั้งค่า Link Refresh Settings ของ CC-Link

ทำการเลือกฝั่ง CPU Side เป็น Device Name ที่ใช้งานกับ CC-Link ต้องเป็น Address ที่ไม่ซ้ำกับการใช้งานอื่น ๆ ตามรูปที่ 3.15 โดยรายละเอียดข้อมูลการส่งข้อมูลระหว่างพีแอลซีและตัวควบคุมหุ่นยนต์ เป็นไปตามตารางที่ 3.2 และ 3.3

ตารางที่ 3.2 Address การส่งข้อมูลแบบบิตระหว่างหุ่นยนต์และพีแอลซี

Robot to PLC			PLC to Robot		
		Comment			Comment
X2000	60 out	robot ready	Y2000	60 in	Order Ready
0	0	robot ready	0	0	Order Ready
1	1	robot finish	1	1	robot finish
2	2	trigger more drug	2	2	trigger more drug
3	3	order failed	3	3	order failed
4	4	robot start	4	4	robot start
A	10	robot stop here	A	10	
15	21	A Pick Finish	15	21	A Pick Finish
16	22	B Pick Finish	16	22	B Pick Finish
17	23	C Pick Finish	17	23	C Pick Finish

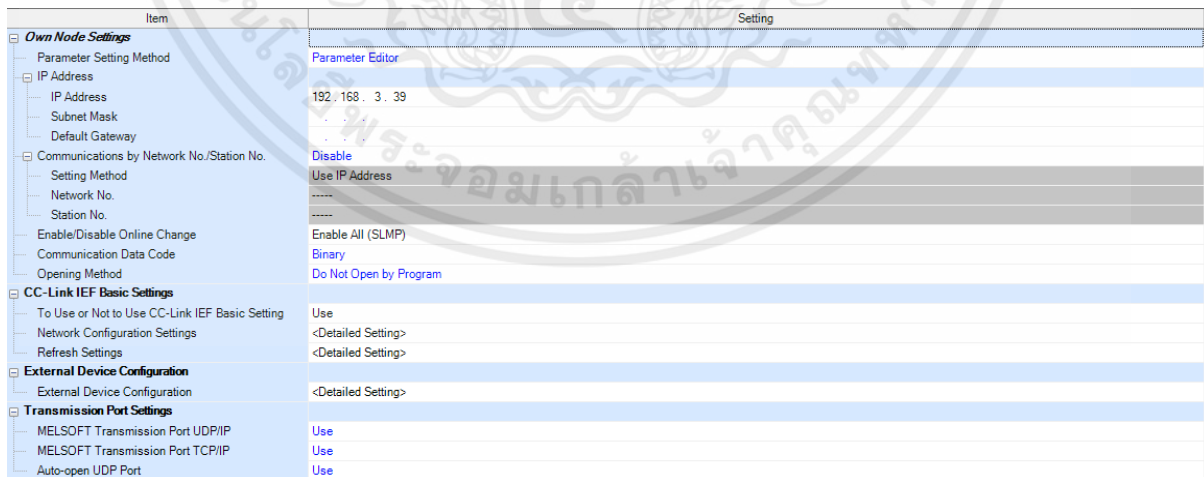
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 31 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18	24	D Pick Finish	18	24	D Pick Finish
19	25	E Pick Finish	19	25	E Pick Finish
1A	26	F Pick Finish	1A	26	F Pick Finish
20	32	Robot Can't find cover	20	32	Robot Can't find cover
21	33		21	33	Robot Pick Cover again
29	41		29	41	Low Speed
2A	42		2A	42	Normal Speed
2B	43		2B	43	High Speed
32	50		32	50	Robot auto enable
33	51	Robot Start	33	51	Robot Start
34	52	STOP2	34	52	Robot Stop
35	53	Prg sel enable	35	53	prg reset
36	54	Durring error	36	54	error reset
37	55	op enable	37	55	op enable
38	56	servo on	38	56	servo on
39	57	servo off	39	57	servo off
3A	58		3A	58	set ovrđ
3B	59		3B	59	ovrd output reg
3C	60		3C	60	err output reg
3D	61	E Stop	3D	61	
3E	62	H Error	3E	62	
3F	63	L Error	3F	63	
40	64	Caution	40	64	
41	65	Cycle Stop	41	65	Cycle Stop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 32 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 Address การส่งข้อมูลประเภทตัวเลขระหว่างหุ่นยนต์และพีแอลซี

Robot to PLC			PLC to Robot		
W1200	60 in	Comment	W1300	60 out	Comment
0	0	ovrd robot	0	0	ovrd robot
1	1	data reserve	1	1	data reserve
2	2	data reserve	2	2	data reserve
3	3	data reserve	3	3	data reserve
4	4	data reserve	4	4	data reserve
5	5	data reserve	5	5	data reserve
6	6	data reserve	6	6	data reserve
A	10	request drug A	A	10	order drug A
B	11	request drug B	B	11	order drug B
C	12	request drug C	C	12	order drug C
D	13	request drug D	D	13	order drug D
E	14	request drug E	E	14	order drug E
F	15	request drug F	F	15	order drug F

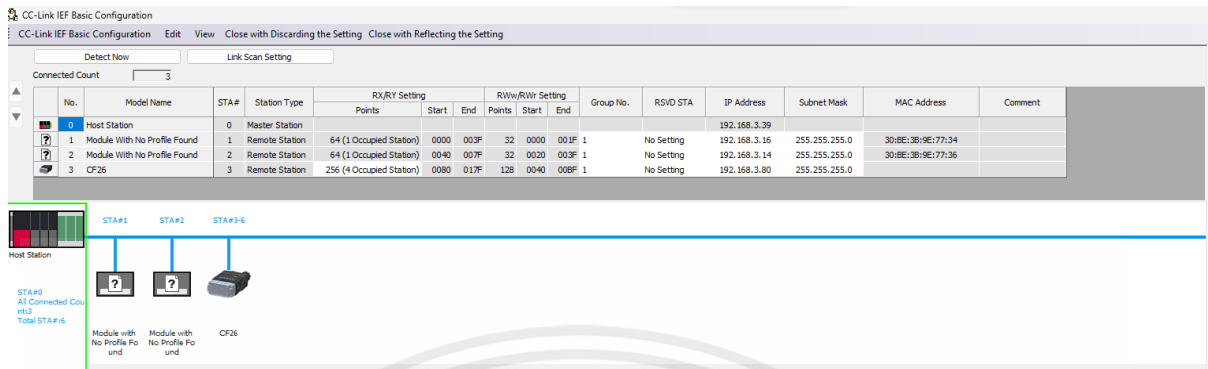


รูปที่ 3.16 การตั้งค่า Module Parameter เพื่อใช้งานพื้นฐาน CC-Link IEF

การตั้งค่าพารามิเตอร์ใน GX Work3 เพื่อให้พีแอลซีสามารถส่งข้อมูลกับอินเวอร์เตอร์ FR-E800 และโค้ดรีดเดอร์ CF26 ได้ผ่านโปรโตคอลพื้นฐาน CC-Link IEF ทำได้โดยการเข้าไปที่ Module Parameter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 33 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นหน้าตามรูปที่ 3.16 ให้ตั้งค่าที่ CC-Link IEF Basic Settings ให้ปรับเป็น Use หลังจากนั้นกดเข้าไปที่ Network Configuration Setting



รูปที่ 3.17 การตั้งค่า Network Configuration Setting ของพื้นฐาน CC-Link IEF

หลังจากเข้ามาที่หน้า CC-Link IEF Basic Settings สามารถทำการกด Detect Now ในหน้า Network Configuration Setting ตามรูปที่ 3.17 เพื่อตรวจสอบอุปกรณ์ที่เชื่อมต่ออยู่ได้ โดยมีอินเวอร์เตอร์ FR-E800 2 ตัว มีที่อยู่ IP เป็น 192.168.3.14 และ 192.168.3.16 และโค้ดรีดเดอร์ CF26 มีที่อยู่ IP เป็น 192.163.3.80 โดย Point ที่ใช้งานคือ 64 64 และ 256

Link Side				CPU Side					
Device Name	Points	Start	End	Target	Device Name	Points	Start	End	
RX	384	00000	0017F	↔	Specify Devi	X	384	01000	0117F
RY	384	00000	0017F	↔	Specify Devi	Y	384	01000	0117F
RWr	192	00000	000BF	↔	Specify Devi	W	192	01000	010BF
RWw	192	00000	000BF	↔	Specify Devi	W	192	01100	011BF

รูปที่ 3.18 การตั้งค่า Link Refresh Settings ของพื้นฐาน CC-Link IEF

ทำการเลือกฝั่ง CPU Side เป็น Device Name ที่ใช้งานกับพื้นฐาน CC-Link IEF ต้องเป็น Address ที่ไม่ซ้ำกับการใช้งานอื่น ๆ ตามรูปที่ 3.18 โดยรายละเอียดข้อมูลการส่งข้อมูลระหว่างพีแอลซี และตัวควบคุมหุ่นยนต์สการาเป็นไปตามตารางที่ 3.4 และ 3.5 ตามคู่มือการใช้งานในรูปที่ 3.19 และรูปที่ 3.20

Device No. ^{*7}	Signal	Refer to page	Device No. ^{*7}	Signal	Refer to page
RYn0	Forward rotation command ^{*2}	46	RXn0	Forward running	47
RYn1	Reverse rotation command ^{*2}	46	RXn1	Reverse running	47
RYn2	High-speed operation command (terminal RH function) ^{*1}	46	RXn2	Running (terminal RUN function) ^{*3}	47
RYn3	Middle-speed operation command (terminal RM function) ^{*1}	46	RXn3	Up to frequency ^{*2}	47
RYn4	Low-speed operation command (terminal RL function) ^{*1}	46	RXn4	Overload warning ^{*2}	47
RYn5	JOG operation command 2 ^{*2}	46	RXn5	Pr.193 assignment function (NET Y1) ^{*6}	47
RYn6	Second function selection ^{*2}	46	RXn6	Frequency detection (terminal FU function) ^{*3}	47
RYn7	Current input selection ^{*2}	46	RXn7	Fault (terminal ABC function) ^{*3}	47
RYn8	Pr.185 assignment function (NET X1) ^{*5}	46	RXn8	Pr.194 assignment function (NET Y2) ^{*6}	47
RYn9	Output stop (terminal MRS function) ^{*1}	46	RXn9	Pr.313 assignment function (DO0) ^{*4}	47
RYnA	Pr.186 assignment function (NET X2) ^{*5}	46	RXnA	Pr.314 assignment function (DO1) ^{*4}	47
RYnB	Pr.184 assignment function (RES) ^{*5}	46	RXnB	Pr.315 assignment function (DO2) ^{*4}	47
RYnC	Monitor command	46	RXnC	Monitoring	47
RYnD	Frequency setting command (RAM)	46	RXnD	Frequency setting completion (RAM)	47
RYnE	Frequency setting command (RAM, EEPROM)	46	RXnE	Frequency setting completion (RAM, EEPROM)	47
RYnF	Instruction code execution request	46	RXnF	Instruction code execution completed	47
RY(n+1)0 to RY(n+1)7	Reserved	—	RX(n+1)0 to RX(n+1)5	Reserved	—
			RX(n+1)6	Pr.195 assignment function (NET Y3) ^{*6}	47
			RX(n+1)7	Pr.196 assignment function (NET Y4) ^{*6}	47

Device No. ^{*7}	Signal	Refer to page	Device No. ^{*7}	Signal	Refer to page
RY(n+1)8	Not used (initial data process completion flag)	—	RX(n+1)8	Not used (initial data process request flag)	—
RY(n+1)9	Not used (initial data process request flag)	—	RX(n+1)9	Not used (initial data process completion flag)	—
RY(n+1)A	Error reset request flag	46	RX(n+1)A	Error status flag	47
RY(n+1)B	Pr.187 assignment function (NET X3) ^{*5}	46	RX(n+1)B	Remote station ready	47
RY(n+1)C	Pr.188 assignment function (NET X4) ^{*5}	46			
RY(n+1)D	Pr.189 assignment function (NET X5) ^{*5}	46	RX(n+1)C to RX(n+1)F	Reserved	—
RY(n+1)E	Reserved	—			
RY(n+1)F					

รูปที่ 3.19 สัญญาณที่ใช้ในการควบคุมอินเวอร์เตอร์ด้วยพีแอลซีจากคู่มือ [16]

■Control blocks

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved				Results Ack	Buffer Results Enable	Trigger	Trigger Enable
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
Reserved							
Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
Reserved						Initiate String Cmd	Set User Data
Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Soft Event							

■Status blocks

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved				Missed Acq	Reserved	Trigger Ack	Trigger Ready
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
General Fault	Reserved			Results Available	Results Buffer Overrun	Decode Complete Toggle	Decoding
Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
Reserved						String Cmd Ack	Set User Data Ack
Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
Soft Event Ack							

รูปที่ 3.20 สัญญาณที่ใช้ในการควบคุมไค้ตรีเตอร์ด้วยพีแอลซีจากคู่มือ [17]

ตารางที่ 3.4 สัญญาณ X1000 และ Y1000 ที่พีแอลซีใช้ในการควบคุมอินเวอร์เตอร์และไค้ตรีเตอร์

Device to PLC		PLC to Device	
X1000	Comment	Y1000	Comment
0		0	drug cv left
1		1	drug cv right
2		2	hi inverter in
3		3	med inverter in
4		4	low inverter in
1B	inverter drug ready	1B	
40		40	box cv left
41		41	box cv right
42		42	hi inverter out
43		43	med inverter out
44		44	low inverter out
5B	Inverter box Ready	5B	
80		80	Trigger cf26 enable
81		81	Trigger cf26

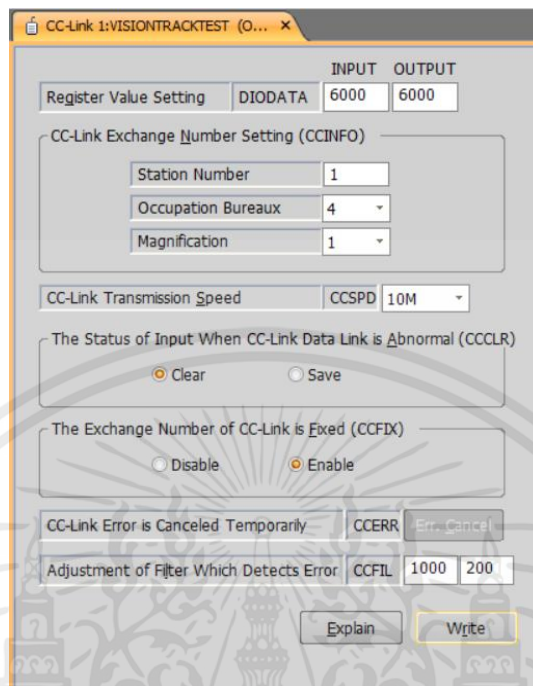
ตารางที่ 3.5 สัญญาณ W1000 และ W1100 ที่พีแอลซีใช้ในการควบคุมอินเวอร์เตอร์และโค้ดรีดเตอร์

Device to PLC		PLC to Device	
W1000	Comment	W11000	Comment
0		0	
1		1	Base Speed Value
21		21	Base Speed Value
43	have data in buffer	43	
45	data from CF26	45	
46	data from CF26	46	
47	data from CF26	47	
48	data from CF26	48	
49	data from CF26	49	
4A	data from CF26	4A	
4B	data from CF26	4B	
4C	data from CF26	4C	
4D	data from CF26	4D	
4E	data from CF26	4E	
4F	data from CF26	4F	
50	data from CF26	50	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 37 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การเชื่อมต่อหุ่นยนต์กับอุปกรณ์อื่น

1. การตั้งค่า CC-Link ของหุ่นยนต์

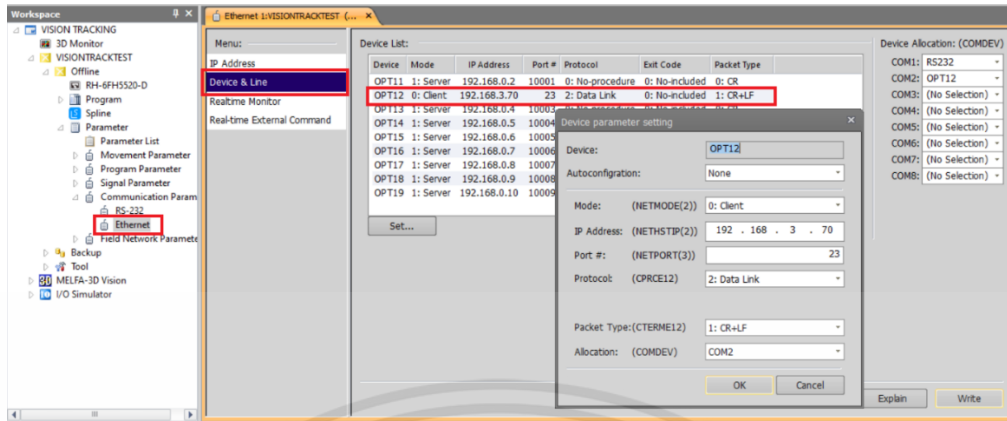


รูปที่ 3.21 การตั้งค่า Parameter CC-Link ของหุ่นยนต์ในซอฟต์แวร์ RT ToolBox3

ไปที่ Parameter เลือก Field Network Parameter มี CC-Link ให้เลือกหลังจากคลิกเข้าไปมีหน้าต่างขึ้นมาดังแสดงในรูปที่ 3.21 ให้ใส่ข้อมูลโดยข้อมูลที่จำเป็นต้องใส่มีดังนี้

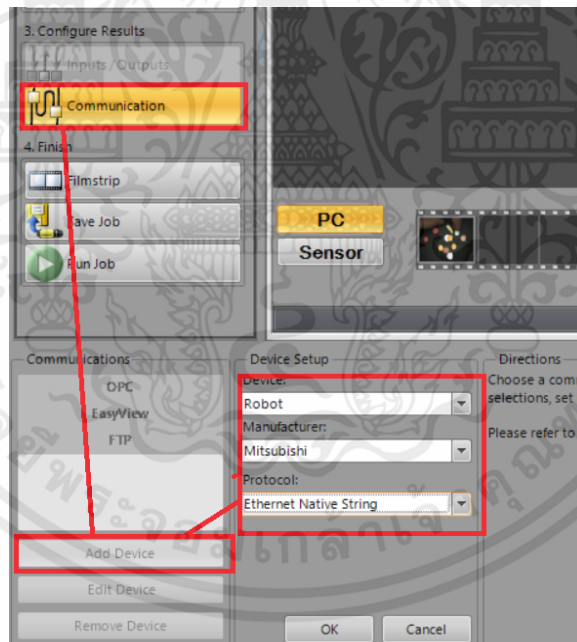
- Station Number ต้องตั้งเป็น 1 ให้ตรงกับที่ตั้งค่าในพีแอลซี
- Occupation Bureaux เป็นการเพิ่มจำนวนข้อมูลที่รับส่งได้ด้วยการคูณตั้งเป็น 4 เพื่อให้เพียงพอต่อการใช้งาน
- DIODATA ทั้งอินพุตและเอาต์พุตตั้งเป็น 6000 เพื่อให้ส่งค่าข้อมูลที่เกี่ยวข้องกับการควบคุมหุ่นยนต์ให้กับพีแอลซีเริ่มที่ 6000

3.4.2 การเชื่อมต่อหุ่นยนต์กับวิชันเซนเซอร์



รูปที่ 3.22 หน้าการตั้งค่า Parameter หุ่นยนต์เพื่อเชื่อมต่อวิชันเซนเซอร์

การตั้งค่า Parameter ของหุ่นยนต์สำหรับเชื่อมต่อวิชันเซนเซอร์ต้องไปที่ Communication Parameter เลือก Ethernet ที่หน้า Device Line มีให้เลือกที่ต้องการใช้ให้ทำการเลือก OPT12 ซึ่งเป็น COM port 2 จากนั้นตั้งค่าตามรูปที่ 3.22



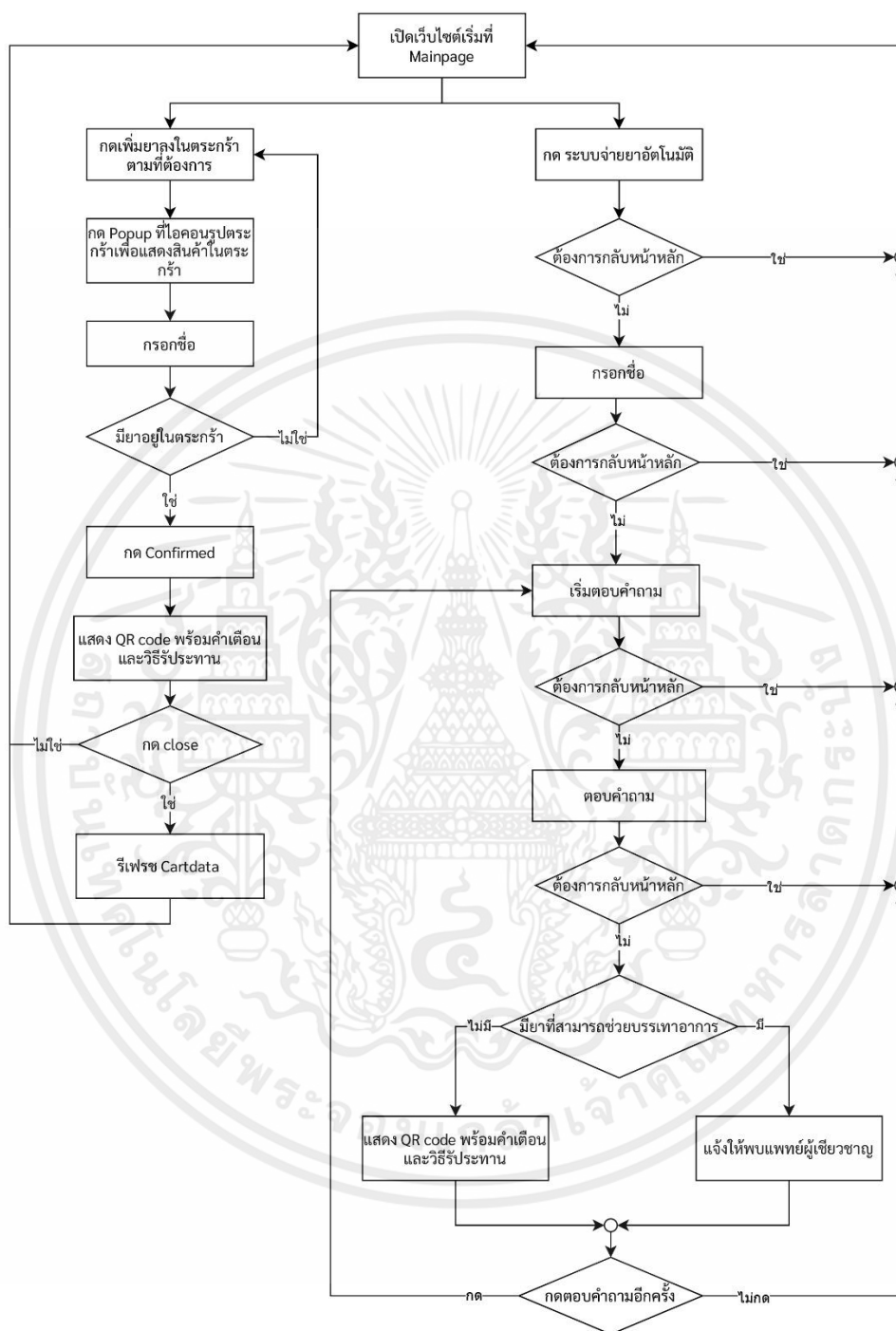
รูปที่ 3.23 การตั้งค่าของวิชันเซนเซอร์เพื่อเชื่อมต่อหุ่นยนต์

การตั้งค่าวิชันเซนเซอร์สำหรับการเชื่อมต่อกับหุ่นยนต์ ให้เข้าไปตั้งค่าที่ Communication ทำการกด Add Device แล้วเลือกตั้งค่าตามรูปที่ 3.23

3.5 การสร้างเว็บแอปพลิเคชัน

ขั้นตอนการกำหนดค่าเพื่อสร้างคิวอาร์โค้ดจากเว็บเป็นไปได้อย่างสองวิธี ได้แก่ การตอบคำถามเพื่อคัดกรองอาการและการเลือกตัวยาเพื่อสั่งยาที่ต้องการ โดยมีการทำงานตามผังงานดังรูปที่ 3.24 เมื่อเปิดเว็บแอปพลิเคชันหน้า Mainpage จะปรากฏขึ้นซึ่งในสามารถเลือกยาที่ต้องการได้โดยกดที่ปุ่มเพิ่มยาลงตะกร้าโดยมีชื่อปุ่มว่า “Add to cart” จากให้ให้เลือกจำนวนยาที่ต้องการและกดยืนยันสามารถทำได้มากที่สุดชนิดละ 5 เม็ด หากเพิ่มยาที่ต้องการครบแล้วให้กดที่ไอคอนรูปตะกร้าเพื่อแสดงสินค้าในตะกร้าจากนั้นให้กรอกชื้อจากนั้นกดยืนยันหากมีสินค้าอยู่ภายในตะกร้าจะสามารถกดปุ่ม “Confirmed” ได้หากไม่มีให้กลับไปยังหน้าเพิ่มยาที่ต้องการก่อน เมื่อกดยืนยันแล้วจะขึ้นพ็อปอัพ (Pop Up) ขึ้นมาซึ่งภายในจะแสดงราคารวมของสินค้าทั้งหมด จำนวนสินค้าที่เลือก รายละเอียดการรับประทาน ข้อควรระวังในการรับประทานยา และคิวอาร์โค้ด หากกดปุ่ม “close” ตัวเว็บแอปพลิเคชันจะทำการรีเฟรชหน้าเพื่อรีเซตค่าต่าง ๆ หากไม่กดแต่กลับไปหน้าเว็บแอปพลิเคชันสินค้าในตะกร้าจะยังอยู่คงเดิม หากกดปุ่ม “ระบบจ่ายยาอัตโนมัติ” ที่หน้า Mainpage เว็บแอปพลิเคชันแสดงหน้า Questionpage ซึ่งในหน้านี้สามารถกดกลับไปยังหน้า Mainpage ได้ตลอดในหน้าแรกของ Questionpage แสดงกล่องข้อความเพื่อให้กรอกชื้อผู้ใช้หากกรอกชื้อและกดยืนยันเรียบร้อยแล้วจะเริ่มถามคำถามกับผู้ใช้งาน เมื่อตอบคำถามจนครบหากมียาที่สามารถช่วยได้จะแสดงคิวอาร์โค้ดพร้อมคำเตือนในการรับประทานหลังจากนั้นจะปุ่ม “ตอบคำถามอีกครั้ง” จะปรากฏขึ้นหากกดที่ “ตอบคำถามอีกครั้ง” จะเริ่มถามคำถามใหม่หากกดที่กลับไปหน้าหลักเว็บแอปพลิเคชันแสดง Mainpage

ขั้นตอนการใช้งานเพื่อสร้าง QR code



รูปที่ 3.24 ผังงานการทำงานของการสร้างคิวอาร์โค้ดจากเว็บแอปพลิเคชัน

3.5.1 การเลือกเฟรมเวิร์กและแผนการดำเนินงาน

ในการพัฒนาเว็บแอปพลิเคชันทางผู้จัดทำเลือก Codesandbox.io เป็นเครื่องมือในการพัฒนาเว็บแอปพลิเคชันในช่วงแรกและเลือก Visual studio code ในการพัฒนาช่วงหลังเนื่องด้วยความสะดวกสบายและเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 41 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สะดวกสบายของ และเร็วในการแสดงผลของ Codesandbox.io ที่สามารถพัฒนาเว็บแอปพลิเคชันบน Codesandbox.io ได้แบบออนไลน์ แต่ด้วยข้อจำกัดในการ Deploy ตัวเว็บแอปพลิเคชันลงเว็บโฮสติ้งจึงใช้ Visual studio code ในการพัฒนาช่วงหลัง ซึ่งการพัฒนาเว็บแอปพลิเคชันโดยเว็บแอปพลิเคชันนี้เน้นจัดการหน้าจอแสดงผลให้เข้ากับหน้าจอมือถือเป็นหลักในส่วนนี้เลือกใช้ React เป็นเฟรมเวิร์กเนื่องด้วยสาเหตุดังนี้

1. React มีการจัดการที่ดีและมีการอัปเดตแบบเป็นขั้นเป็นตอนเพราะใช้วิธีการทำงานแบบ Virtual DOM คือวิธีการเก็บ Document (โครงร่าง Element ถูกนำไปแทนใน HTML) ในรูปแบบ Object (วิธีการจัดเก็บข้อมูลซึ่งประกอบด้วยตัวแปรอยู่ในนั้น) ด้วยวิธีฝังงานต้นไม้และใช้ JavaScript เพื่อไปเข้าถึงหรือจัดการ HTML ในแต่ละส่วนซึ่งเป็นวิธีที่มีประสิทธิภาพสูงจึงช่วยลดการทำงานที่ไม่จำเป็นและทำให้ปรับเปลี่ยนส่วนที่แสดงผลบนหน้าเว็บแอปพลิเคชันในหน้าจอของผู้ใช้งานซึ่งในที่นี้หมายถึง UI ได้อย่างรวดเร็ว

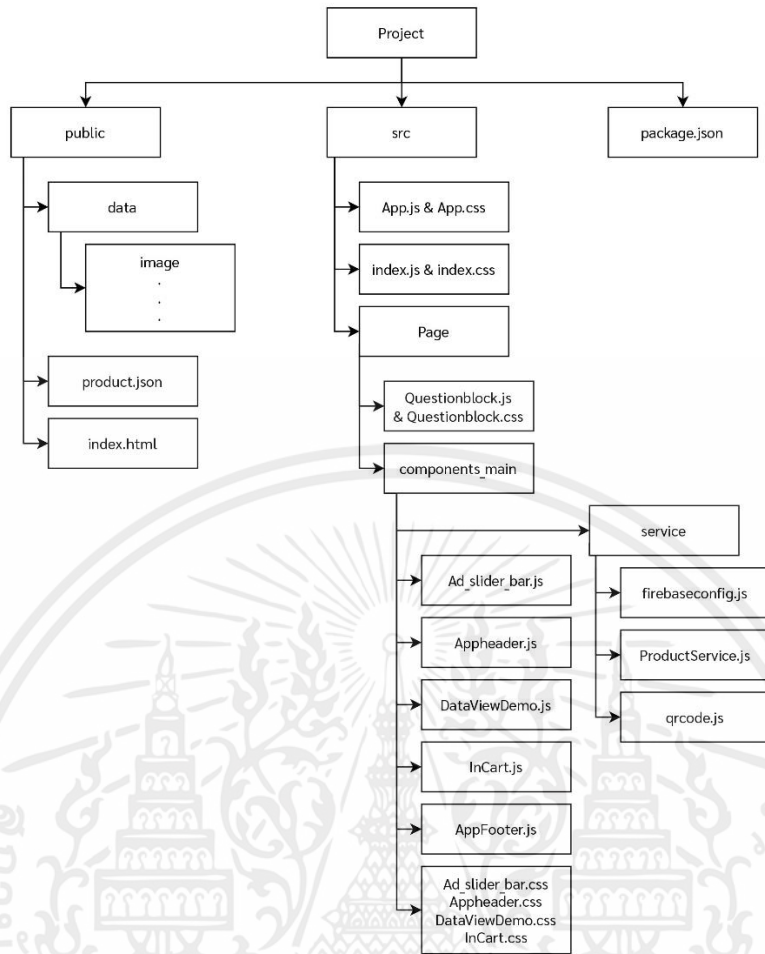
2. React เป็นที่นิยมและมีการใช้งานอย่างแพร่หลายซึ่งมีแหล่งข้อมูลรองรับมีจำนวนมากอีกทั้งยังมีส่วนเสริมที่มีอยู่มากมายที่สามารถช่วยในการพัฒนาและขยายความสามารถของแอปพลิเคชันได้นอกจากนี้ React ยังมีแหล่งข้อมูลที่มีสำหรับการเรียนรู้และแก้ไขปัญหา

3. React เป็นเฟรมเวิร์กที่ยืดหยุ่นและสามารถใช้ร่วมกับเครื่องมืออื่น ๆ ได้ เช่น Redux สำหรับการจัดการสถานะแอปพลิเคชัน React Router สำหรับการจัดการเส้นทาง และอื่น ๆ นอกจากนี้ React ยังสามารถใช้ร่วมกับภาษา JavaScript และ TypeScript ได้ อีกทั้งยังรองรับฟังก์ชันของผู้ให้บริการหลายเจ้า

การออกแบบหน้าจอและแก้ไขข้อผิดพลาดของเว็บแอปพลิเคชันนั้นใช้ Google Developer Tool ในการช่วยปรับแบบหน้าจอผู้ใช้ ตกแต่งหน้าจอผู้ใช้ จนไปถึงการแก้ไขข้อผิดพลาดของเว็บแอปพลิเคชัน

3.5.2 Component และการทำงานของ Component ในแต่ละส่วนของเว็บ

ใน React ส่วนของ Component ทำให้สามารถแบ่ง UI ได้เป็นส่วน ๆ เพื่อนำกลับมาใช้ซ้ำได้ React สามารถสร้าง Component ได้ทั้งแบบฟังก์ชันและแบบคลาสขึ้นอยู่กับการใช้งาน แต่ทั้งคู่แสดงผลออกมาเป็นเหมือน HTML Element ทั่วไปแต่ทำงานแยกกับส่วน HTML โดยใน “Project” นี้แบ่งส่วนหลักอยู่หลายส่วนเป็นไฟล์ JavaScript โดยมีไฟล์ทั้งหมดดังรูปที่ 3.25 ไฟล์ CSS ทั้งหมดมีเพื่อตกแต่งตัว Component ของไฟล์ JavaScript โดยไฟล์ส่วนใหญ่ในโปรแกรมนี้ใช้ไฟล์ CSS ชื่อเดียวกันกับไฟล์ JavaScript แต่บางไฟล์ใช้ไฟล์ CSS ของไฟล์อื่นซึ่งการออกแบบในส่วนนี้ออกแบบให้เป็น Responsive Web Design เป็นเทคนิคที่ทำให้ตัวเว็บแอปพลิเคชัน เปลี่ยนไปตามขนาดของหน้าจอไม่ว่าเป็นคอมพิวเตอร์หรือมือถือ



รูปที่ 3.25 ไฟล์ทั้งหมดใน “Project” ของเว็บแอปพลิเคชัน

การออกแบบเว็บแอปพลิเคชันโฟลเดอร์แบ่งเป็นสองโฟลเดอร์ใหญ่ ได้แก่ โฟลเดอร์ public ซึ่งประกอบด้วย ไฟล์รูป ไฟล์ ProductService ที่ทำหน้าที่รับข้อมูล Json ในรูปแบบ Promise ทำให้สามารถเรียกใช้การทำงานอื่นต่อไปได้ หากรับข้อมูลสำเร็จด้วยการรับข้อมูลรูปแบบ Promise โดยไฟล์ ProductService ประกอบด้วยฟังก์ชันย่อยเกี่ยวกับจัดการข้อมูล โฟลเดอร์ src ประกอบด้วยไฟล์การทำงานต่าง ๆ ตัวโปรแกรมแต่ละไฟล์ในโฟลเดอร์ src ได้ทำการแยกหน้าที่เป็นส่วน ๆ ดังนี้

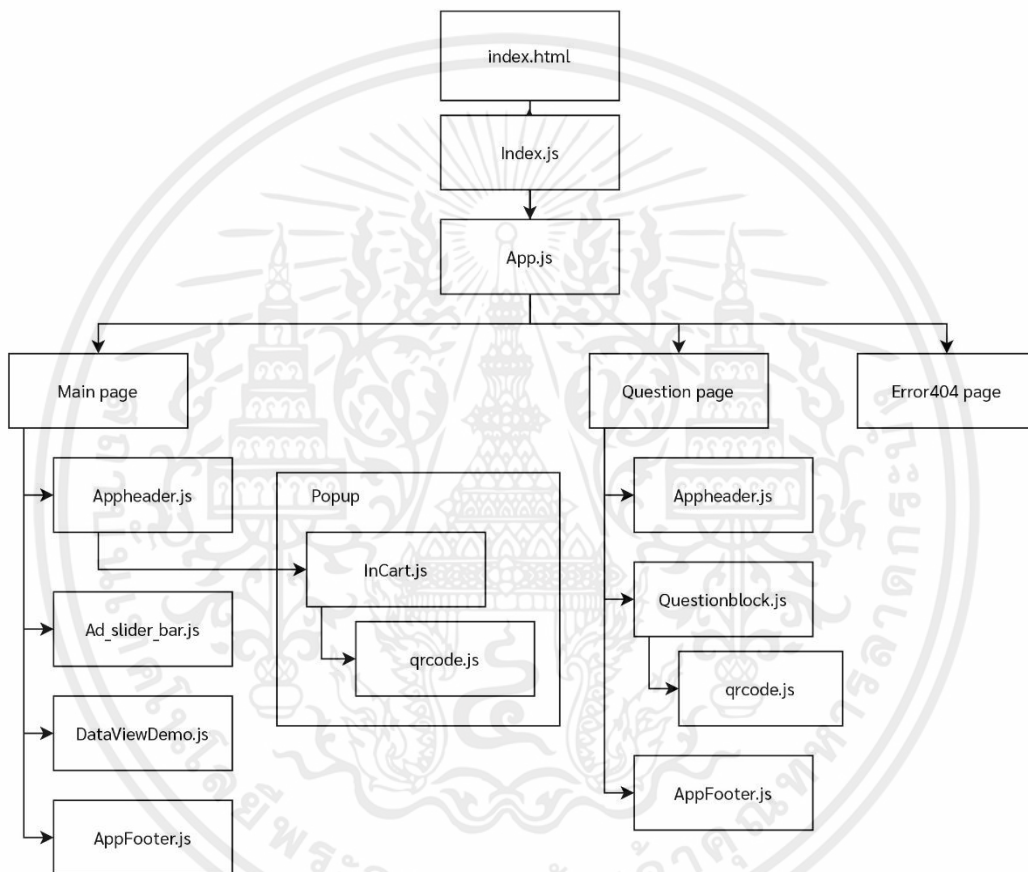
1. index.js เป็นส่วนส่งข้อมูลให้ index.html นำไปแสดงผลต่อไป
2. App.js เป็นส่วน routing หน้าเพจแต่ละหน้าระบบ navigate อยู่ในส่วนนี้เป็นหลัก
3. Ad_slider_bar.js เป็นส่วนแสดงรูปที่เปลี่ยนไปเรื่อย ๆ ตามรูปที่ตั้งค่า
4. Appheader.js เป็นส่วนบนสุดของเว็บทุกหน้า
5. DataViewDemo.js เป็นส่วนการเพิ่มสินค้าเข้าตะกร้าในหน้า Main_page
6. InCart.js เป็น ส่วนที่ออปชั่นที่อยู่ใน Appheader โดยทำหน้าที่จัดการข้อมูลในตะกร้าสินค้า
7. AppFooter.js เป็นส่วนแสดงผลท้ายหน้าจอของเว็บเพจทุกหน้า

8. Firebase.js เป็นไฟล์สำหรับการบันทึก Firebase API สำหรับการใช่ API (Application Programming Interface) ของ Firebase

9. ProductService.js เป็นไฟล์จัดการข้อมูล Cartdata และข้อมูลที่ได้รับมาจากไฟล์ Json

10. qrcode.js ทำหน้าที่สร้างรูปตัวคิวอาร์โค้ดออกมาแสดงผลโดยรับข้อมูลมาและนำไปสร้างคิวอาร์โค้ด

เมื่อแบ่งไฟล์ออกเป็น Component และจัดเรียง Component ในแต่ละหน้าได้ดังแสดงในรูปที่ 3.26



รูปที่ 3.26 Component ของแต่ละหน้า

โปรแกรมเริ่มทำงานที่ Index.html โดยเป็นหน้ากำหนดองค์หลักต่าง ๆ ของเว็บ เช่น ไอคอนที่ตัวเว็บเบราว์เซอร์ และชื่อบนตัวเว็บเบราว์เซอร์ การรับไฟล์มาจาก root โดยผ่าน `<div id="root"></div>` ดังรูปที่ 3.27 โดยมีไฟล์ index.js ซึ่งเป็นไฟล์ JavaScript ส่งข้อมูลที่ ต้องนำไปแสดงผลให้ `ReactDOM.render()`; ซึ่งเป็นส่วนเสริมจาก ReactDOM โดยกำหนดให้แสดงผลในส่วนของ `<App/>` ที่อยู่ในไฟล์ App.js ลงไปใน root

```
JS index.js X JS InCart.js JS qrcode.js # InCart.css
rc > JS index.js
1 import React from "react";
2 import ReactDOM from "react-dom";
3 import "./index.css";
4 import App from "./App";
5
6 ReactDOM.render(<App />, document.getElementById("root"));
```

รูปที่ 3.27 โปรแกรมในไฟล์ index.js

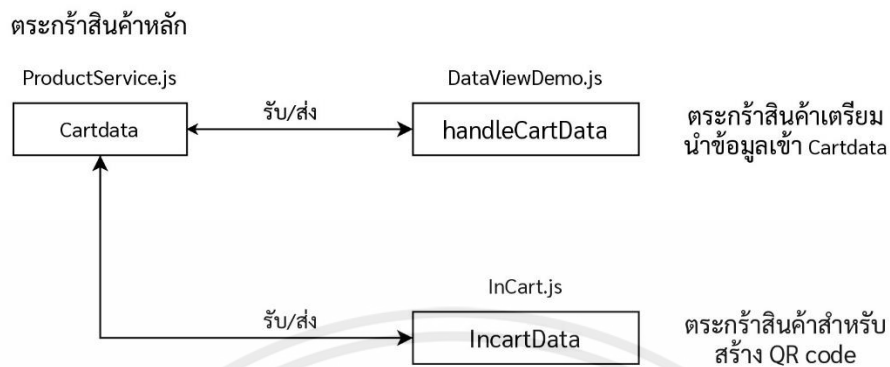
ซึ่งฟังก์ชัน App นั้นทำหน้าที่จัดการแสดงผลออกเป็นหน้า ๆ โดยทั้งหมดมีอยู่สามหน้าดังรูปที่ 3.28 ที่อยู่ใน Component แสดงถึงหน้าที่สามารถแสดงได้ประกอบด้วย หน้า Mainpage หน้า Qusepage และ หน้า Error404 โดย Path บอกถึงยูอาร์แอลส่วนหลัง domain หากไม่พบ Path ที่ส่งมา ซึ่งหมายความว่าไม่ใช่หน้า Mainpage และหน้า Qusepage แสดงหน้า Error404 ซึ่งแต่ละหน้าเป็น Component ของไฟล์ App.js มีการจัดการเป็นไฟล์เพื่อรวม Component ต่าง ๆ เข้าด้วยกัน ในการจัดเรียงให้หน้าเว็บเป็นระเบียบและง่ายต่อการจัดการซึ่งส่วนของ Mainpage มีปุ่มชื่อ “ระบบจ่ายยาตามอาการ” เป็นตัวนำทางไปยังหน้า Qusepage และ Qusepage มีปุ่มกลับไปยังหน้า Mainpage ได้ตลอด

```
const App = () => {
  return (
    <Router>
      <Appheader />
      <Switch>
        <Route exact path="/" component={Mainpage} />
        <Route path="/Qusepage" component={Qusepage} />
        <Route path="*" component={Error404} />
      </Switch>
      <link
        rel="stylesheet"
        href="https://unpkg.com/primeicons@6.0.1/primeicons.css"
      />
      <AppFooter />
    </Router>
  );
};
```

รูปที่ 3.28 โปรแกรมในไฟล์ App.js ส่วนจัดการหน้าจอเพื่อไปยังหน้าต่าง ๆ

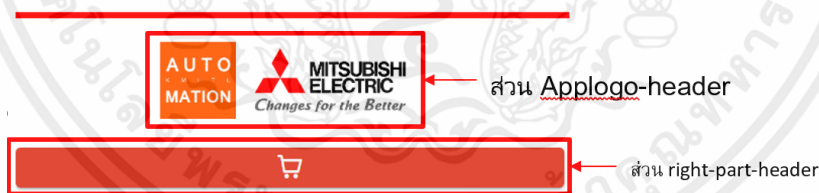
ระบบสั่งยานี้มี Array object หลักชื่อ Cartdata เป็นตัวเก็บข้อมูลตัวยาที่ผู้ใช้กดสั่งโดยมี Array object เป็นเก็บข้อมูลสำหรับแต่ละหน้าของเว็บแอปพลิเคชันเพื่อกันไม่ให้ event บางอย่างกระทบกับ Array object หลักประกอบด้วย handleCartdata และ InCartdata ดังรูปที่ 3.29 โดย handleCartdata เปรียบเสมือนตะกร้าสำหรับการรองรับข้อมูลก่อนกดยืนยันในส่วนพ็อบอัพหลังกด Add to cart ซึ่งเมื่อกด Add to cart ทำการดึงข้อมูล Cartdata มาเป็น handleCartdata ก่อนโดย handleCartdata นี้ทำหน้าที่รองรับและปรับข้อมูลจากการกดลดหรือเพิ่มสินค้าในตะกร้า เมื่อกดยืนยันทำการตัดลดค่า handleCartdata ลงไปใน Cartdata ซึ่งอยู่ในไฟล์ของ ProductService ส่วนของ

Array object ที่ชื่อ InCartdata นั้นทำการรับและส่งข้อมูลระหว่าง Cartdata และ InCartdata โดยทุกครั้ง InCartdata มีการเปลี่ยนแปลง ทำการนำข้อมูลใน InCartdata ไปสร้างคิวอาร์โค้ดเป็นการถัดไป



รูปที่ 3.29 แนวคิดการรับและส่งข้อมูลของตระกร้าสินค้า

Appheader.js เป็นส่วนบนสุดของทุกหน้าโดยประกอบไปด้วยส่วน Applogo-header ส่วนนี้แบ่งส่วนประกอบตามรูปที่ 3.30 โดยประกอบด้วยสัญลักษณ์ของบริษัทมิตซูบิชิ อิเล็กทริกซึ่งเชื่อมต่อกับเว็บไซต์ “https://www.mitsubishifa.co.th/th/AboutUs.php” และในส่วนของสัญลักษณ์ KMITL เชื่อมต่อไปยังเว็บ “http://instrumentation.kmitl.ac.th/2017/b-eng-mechatronic-automation/” ส่วนตระกร้าสินค้า เป็นปุ่มที่มีรูปไอคอนตระกร้าอยู่ตรงกลางโดยในส่วนนี้มีฟังก์ชัน Cartsystem ซึ่งอยู่ในไฟล์ Incart.js โดยเพื่ออัปเดตข้อมูลจากรายการสินค้าไปยังปุ่มที่มีรูปไอคอนตระกร้าจากนั้นหน้าเว็บแอปพลิเคชันแสดงผลส่วน Cartsystem เพื่อจัดการสินค้าในตระกร้าต่อไป



รูปที่ 3.30 ส่วน Appheader ในมือถือ

ส่วนเพื่ออัปเดตโปรแกรมในไฟล์ Incart.js ทำหน้าที่จัดการกับข้อมูลของสินค้าในตระกร้าโดยในฟังก์ชัน Cartsystem ซึ่ง Component ในส่วน Cartsystem มีการรับข้อมูลตัวแปร Confirmed setConfirmed showNextContext และ setShowNextContext มาจาก Appheader ซึ่งนำข้อมูลที่รับใช้ในการยืนยันคำสั่งซื้อและแสดงผลหน้าเพื่ออัปเดตเป็นลำดับถัดไปซึ่งเป็นส่วนของ incart ใช้ PrimeReact ในการแสดงผล <OrderList> ซึ่งเป็นฟังก์ชันการแสดงผลรายชื่อของใน Cartdata และปรับจำนวนของ Cartdata ระหว่างนั้นคำนวณราคารวมและใช้ Component textinput ในการรับข้อมูลโดยจำกัดให้ใส่ข้อมูลตามเงื่อนไขใน handleChange ดังรูปที่ 3.31 หลังจากนั้นข้อมูลใน InCartdata และ name ถูกเอามารวมกันและเปลี่ยนประเภทตัวแปรเป็น string ตามรูปแบบที่โค้ดรีดเดอร์ CF26 ต้องการและนำไปใส่ในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 46 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<QRGenerator text={qrCodeData} /> เพื่อทำการสร้างคิวอาร์โค้ดเป็นการถัดไป ส่วนของการอัปเดต คำนับใช้ฟังก์ชัน useState แต่ใช้ในไฟล์ ProductService จึงได้เปลี่ยนชื่อเป็น useData แทน และมีการ กำหนดตัวแปรเพื่อให้่ายต่อการเปลี่ยนค่าสูงสุดที่ผู้ใช้งานสั่งได้ที่ตัวแปร limititem ดังรูปที่ 3.32

```
let Calculateprice;
const handleChange = (e) => {
  const inputValue = e.target.value;
  const trimmedValue = inputValue.trim().slice(0, 8); // ตัดเอาเฉพาะ 8 ตัวอักษรและลบช่องว่างหน้าและหลัง
  const sanitizedValue = trimmedValue.replace(/[^a-zA-Z0-9]/g, ""); // ลบตัวอักษรที่ไม่ใช่ภาษาอังกฤษ

  setValue(sanitizedValue);
  setIsNameFilled(sanitizedValue.length > 0); // ตรวจสอบว่าชื่อถูกกรอกหรือไม่
};
```

รูปที่ 3.31 โปรแกรมส่วนกำหนดเงื่อนไขการใส่ชื่อ

The screenshot shows a code editor with the following code and annotations:

```
export default function Cartsystem({
  confirmed,
  setConfirmed,
  showNextContext,
  setShowNextContext
}) {
  let limititem = 5;
  const [IncartData, updateData] = useData();
  const [totalPrice, settotalprice] = useState(calculateTotalPrice(IncartData));
  const [limit_count, setlimit_count] = useState(false);
  const [value, setValue] = useState("");
  const [isNameFilled, setIsNameFilled] = useState(false);
  const handleConfirm = () => {
    if (value.trim() === "") {
      console.log("Please enter a name");
      return;
    }
    settotalprice(calculateTotalPrice(IncartData));
    setConfirmed(true);
    setShowNextContext(true);
    //console.log(IncartData);
  };
  useEffect(() => {
    ProductService.getCartProducts().then((data) => updateData(data));
  }, []);
  function removeFromCart(item) {
    ProductService.removeFromCart(item).then((data) => {
      updateData(data);
    });
  }
}
```

Annotations in Thai:

- A box labeled "จำนวนที่" (Quantity) points to the `limititem = 5;` line.
- A box labeled "ใช้ updateData เพื่ออัปเดตค่า" (Use updateData to update the value) points to the `updateData` parameter in `useData()` and the `updateData(data)` call in `useEffect`.
- A box labeled "ใช้เพื่ออัปเดตค่าของ Cartdata" (Use to update the value of Cartdata) points to the `updateData(data)` call in `useEffect`.

รูปที่ 3.32 ส่วนการรับส่งข้อมูลของ Cartdata และ InCartdata

Ad_slider_bar.js เป็นส่วนของการแสดงรูปที่มีการเปลี่ยนรูปไปเรื่อย ๆ ด้วยการจางหายไปและเปลี่ยนรูปตามเวลาที่กำหนดโดยรูปใช้ “react-material-ui-carousel” ในการช่วยเปลี่ยนสร้างโปรแกรม ในส่วนนี้ โดยสามารถเปลี่ยนรูปได้ที่ฟังก์ชัน ImageSlider แก่ที่ตัวแปร items เป็นไฟล์รูปที่ต้องการแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 47 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

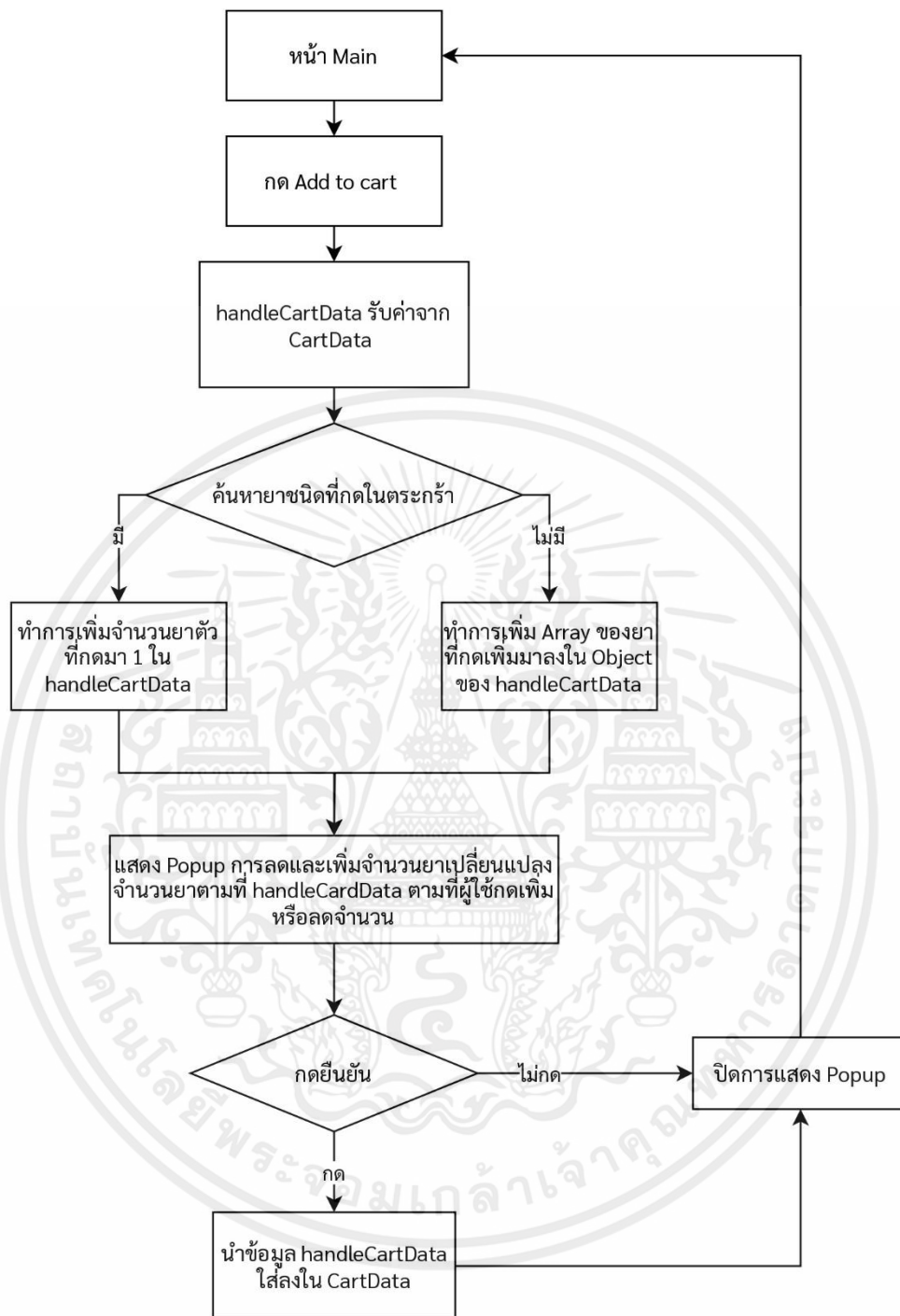
และ name เป็นการกำหนดชื่อรูปและ image เป็นการกำหนดที่อยู่ของไฟล์รูปนั้น ๆ ตามโปรแกรมในไฟล์ Ad_slider_bar.js ดังรูปที่ 3.33

```
JS Ad_slider_bar.js X sys_config_2.jpg products.json public... products.json build... JS App.js
src > Page > components.main > JS Ad_slider_bar.js > ImageSlider > items
1 import React from "react";
2 import "./Ad_slider_bar.css";
3 import Carousel from "react-material-ui-carousel";
4 import { Paper } from "@material-ui/core";
5
6 function ImageSlider(props) {
7   var items = [
8     {
9       name: "Random Name #1",
10      image: "/data/images/sys_config_1.jpg"
11    },
12    {
13      name: "Random Name #2",
14      image: "/data/images/sys_config_2.jpg"
15    },
16    {
17      name: "Random Name #3",
18      image: "/data/images/sys_config_3.jpg"
19    }
20  ];
21
22  return (
23    <Carousel
24      style={{ maxWidth: 200, margin: "auto", marginTop: 50, height: "auto" }}
25    >
26      {items.map((item, i) => (
27        <Paper key={i} style={{ textAlign: "center", display: "inline" }}>
28          <img
29            style={{
30              width: "100%",
31              height: "auto",
32              display: "flex",
33              margin: "auto",
34              justifyContent: "center",
35              alignItems: "center",
36              maxWidth: 1000
37            }}
38            src={item.image}
39            alt={item.name}
40          />
41        </Paper>
42      ))}
43    </Carousel>
44  );
45 }
```

รูปที่ 3.33 โปรแกรมในไฟล์ Ad_slider_bar.js

DataViewDemo.js ส่วนสั่งยาในหน้า Mainpage ส่วนนี้แสดงรูปยาทุกชนิดของชุดสาริตโดยใช้ PrimeReact ในการช่วยสร้างองค์ประกอบโดยรวมได้แก่ การแสดงผลตามไฟล์ Json และการเรียงยาตามราคามากที่สุดหรือน้อยสุดซึ่งสามารถปรับได้มุมขวาบนของ Component ในส่วนนี้ อีกทั้งยังสามารถกดเพิ่มและลดจำนวนยาแต่ละชนิดได้ เมื่อกด Add to cart โปรแกรมส่งค่าไปยัง Cartdata แทนใน handleCartdata และค้นหา Object ใน Array object โดยโปรแกรมหาตัวแปรที่มี Object id เดียวกับชนิดยาที่กด เมื่อพบ Object id ที่ตรงกันให้ทำการเพิ่มยาลงไปในตัวแปร handelCartdata เป็นการถัดไป จากนั้นแสดงฟ็อบอัปเพื่อเพิ่มหรือลดจำนวนจนกว่าผู้ใช้พอใจและนำค่านั้นกลับไปแทนที่ Cartdata หากกดปิดไม่ทำการปรับเปลี่ยนจำนวนตัวยาที่ Cartdata โดยมีฝั่งแสดงการกดเพิ่มจำนวนยาที่ DataViewDemo ดังรูปที่ 3.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและ 48 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.34 ผังงานการทำงานโปรแกรมการเพิ่มตัวยา

Questionblock.js เป็นส่วนสำหรับการตอบคำถามเมื่อเริ่มกดเข้าไปที่ “>>>ระบบจ่ายยาตามอาการ<<<” ทำการแสดงผลหน้า Questionblock ขึ้นมาโดยหน้าแรกหลังจากกดแสดงได้แก่หน้าใส่ชื่อผู้ใช้ เมื่อมีค่าอยู่ใน Name แสดงปุ่ม next ขึ้นมาโดยใช้เงื่อนไขการตรวจสอบเดียวกับ handleChange ใน Incart.js เมื่อกด Next ค่าตัวแปร currentQuestion เปลี่ยนเป็นค่าเริ่มต้นของส่วนเริ่มตอบคำถาม จากนี้สามารถกดตอบคำถามเพื่อไปยังหน้าคำถามถัดไปซึ่งคำถามที่คิดมานั้นอ้างอิงจากยาสามัญประจำบ้าน เมื่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 49อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบคำถามจนครบหากมียาที่ตรงกับอาการทำการสร้างคิวอาร์โค้ดมาให้ทันที หากไม่มีแนะนำให้ไปพบแพทย์เนื่องจากอาการบางอย่างหากไม่ทราบแน่ชัดได้หากเป็นโรคร้ายแรงส่งผลเสียต่อผู้ใช้งานแทนโดยมีคำถามให้ตอบอยู่ใน Object questions ดังรูปที่ 3.35

```
const questions = [
  {
    questionText: "ระบุอาการของคุณ ?",
    answerOptions: [
      { answerText: "ปวดหัว ปวดกล้ามเนื้อ เป็นไข้", ANS: 1 },
      { answerText: "มีน้ำมูก", ANS: 2 },
      { answerText: "ปวดท้อง", ANS: 3 },
      { answerText: "เจ็บคอ", ANS: 4 },
      { answerText: "อื่นๆ", ANS: 0 }
    ]
  },
  {
    questionText: "ปัจจุบันคุณเป็นโรคตับอยู่หรือไม่ ?",
    answerOptions: [
      { answerText: "เป็นโรคตับ", ANS: 0 },
      { answerText: "ไม่เป็นโรคตับ", ANS: 11 }
    ]
  }
]
```

รูปที่ 3.35 ตัวอย่างโปรแกรมรูปแบบคำถาม

เมื่อกดตอบคำถามทำการเปลี่ยนค่าตัวแปร Ans และเข้าเงื่อนไขการตรวจสอบคำตอบจากตัวแปร Ans ดังรูปที่ 3.36 ว่าคำตอบนั้นเป็นอะไร หากคำตอบนั้นมีคำถามถัดไปทำการแสดงคำถามถัดไปด้วยคำสั่ง setCurrentQuestion จนกว่าจบ เมื่อจบแล้วทำการแสดงผลลัพธ์ showsult ซึ่งเป็นตัวแปรที่บรรจุอาการที่ผู้ใช้จะเป็นซึ่งแสดงในหน้าสุดท้ายพร้อมกับวิธีรับประทานและข้อควรระวังพร้อมทั้งส่งค่า text เพื่อไปสร้างเป็นคิวอาร์โค้ดขึ้นมาเพื่อสแกนหน้าตู้และนำชื่อผู้เข้ามาต่อท้ายให้เป็นรูปแบบเดียวกับที่ตั้งค่าการรับที่ซอฟต์แวร์ของโค้ดรีดเดอร์ CF26 หากใช้เลข ID สีหลักหน้าเป็น 5001 เนื่องจากเป็นรูปแบบการจัดยาของหุ่นยนต์สกรารัจทยาเป็นรายวันแทนจัดยาแบบชนิด

```

const handleAnswerOptionClick = (ANS) => {
  text = null;
  let user;
  if (ANS === 0) {
    showresult(5);
    text = null;
  }
  if (ANS === 1) {
    setCurrentQuestion(1);
  }
  if (ANS === 2) {
    setCurrentQuestion(4);
  }
  if (ANS === 3) {
    setCurrentQuestion(5);
  }
  if (ANS === 4) {
    showresult(3);
    text = "5001000000060000";
  }
}

```

รูปที่ 3.36 ตัวอย่างโปรแกรมการนำทางของคำถาม

AppFooter.js เป็นส่วนท้ายของหน้าจอในทุกหน้า โดยเขียนเพื่อเป็นส่วนเสริมสำหรับการใส่ช่องทางติดต่อ โดยในส่วนนี้ไม่ต้องการให้เป็นที่โดดเด่นมากนักดังรูปที่ 3.37 และ รูปที่ 3.38

```

src > Page > components_main > JS AppFooter.js > ...
1  import React from "react";
2
3  export default function AppFooter() {
4    return (
5      <footer className="footer-container">
6        <p className="footer-text">เว็บมีปัญหาติดต่อ : Chanakan.Chavaltada@gmail.com</p>
7      </footer>
8    );
9  }

```

รูปที่ 3.37 โปรแกรมส่วน AppFooter

เว็บมีปัญหาติดต่อ : Chanakan.Chavaltada@gmail.com

รูปที่ 3.38 การแสดงผลส่วน AppFooter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 51 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 การใช้งานและติดตั้งส่วนเสริมใน “Project”

การลง Dependencies หรือส่วนเสริมในซอฟต์แวร์ Visual Studio Code จำเป็นต้องใช้ Node.js ในการช่วยติดตั้งด้วยการพิมพ์คำสั่งติดตั้ง npm install บน powershell ในการติดตั้งแพ็คเกจหรือโมดูลที่ระบุในไฟล์ package.json โดยไฟล์ package.json ใน “Project” นี้มีองค์ประกอบดังรูปที่ 3.39

```
"name": "main-012",
"version": "1.0.0",
"license": "ISC",
"keywords": [],
"main": "src/index.js",
"dependencies": {
  "@material-ui/core": "4.12.4",
  "final-form": "4.20.9",
  "firebase": "^9.22.1",
  "primeflex": "3.3.0",
  "primeicons": "6.0.1",
  "primereact": "9.2.0",
  "qrcode.react": "3.1.0",
  "react": "17.0.2",
  "react-dom": "17.0.2",
  "react-final-form": "6.5.9",
  "react-hook-form": "7.40.0",
  "react-material-ui-carousel": "3.4.2",
  "react-router-dom": "5.3.0",
  "react-scripts": "3.2.0",
  "react-transition-group": "4.4.5",
  "reactstrap": "9.1.10"
},
"devDependencies": {
  "@babel/runtime": "7.13.8",
  "typescript": "4.1.3"
},
"scripts": {
  "start": "react-scripts --openssl-legacy-provider start",
  "build": "react-scripts --openssl-legacy-provider build",
  "test": "react-scripts test --env=jsdom",
  "eject": "react-scripts eject"
},
"browserslist": [
  ">0.2%",
  "not dead",
  "not ie <= 11",
  "not op_mini all"
],
"author": "",
"description": ""
```

รูปที่ 3.39 ไฟล์ Packet.json

โดยส่วนเสริมหลักที่ได้แก่

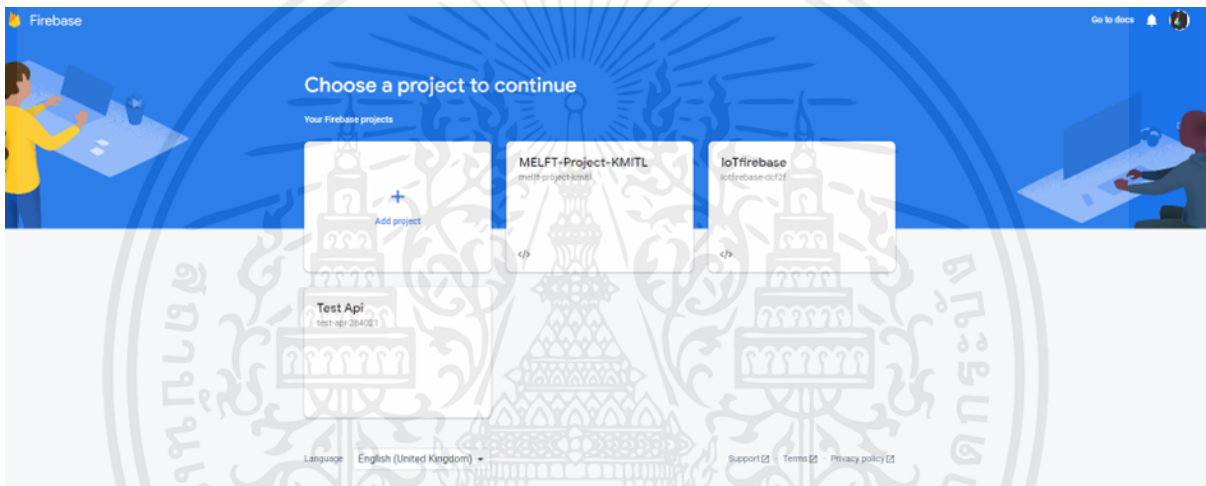
1. Firebase ใช้เพื่อเป็นเว็บโฮสติ้งเพื่อให้ผู้สามารถเข้าถึงได้โดยไม่ต้องอยู่ในเครือข่ายของชุดสาธิต
2. MUI และ PrimeReact ใช้สำหรับการตกแต่งหน้าจอและตั้ง Component ที่จำเป็นต้องใช้ในเว็บแอปพลิเคชันเพิ่มความยืดหยุ่นและลดเวลาในการสร้างเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 52 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. QRcode ใช้สำหรับการสร้างคิวอาร์โค้ดเพื่อแสดงบนหน้าจอ
4. react-dom มีหน้าที่ในการรวมและแสดงผล React Component ลงในต้นแบบ HTML
5. react-router-dom ใช้ในการเชื่อมต่อหน้าต่าง ๆ เข้าด้วยกันโดยไม่จำเป็นต้องรีเฟรชหน้าต่าง
ใด ๆ
6. Form ต่าง ๆ ใช้เพื่อสร้างแบบฟอร์มสำเร็จรูปหรือช่วยในการโยกย้ายข้อมูล

3.5.4 การ Deploy โปรแกรมขึ้น Firebase โฮสติ้งด้วย Visual Studio Code

ทำการสร้าง “Firebase Project” ขึ้นมาเป็นลำดับแรกในการสร้างเว็บแอปพลิเคชันบน Firebase และสามารถสร้างผ่านคอนโซลในซอฟต์แวร์ Visual Studio Code ก็สามารทำได้ หากสร้าง “Firebase Project” สำเร็จขึ้นดังรูปที่ 3.40



รูปที่ 3.40 หน้าการสร้าง “Firebase Project”

เมื่อสร้าง “Firebase Project” เสร็จให้ไปที่เมนูด้านข้างของ Firebase ทำการเลือกการพัฒนา จากนั้นไปที่โฮสติ้งจากนั้นติดตั้ง firebase—tool โดยใช้เทอร์มินัลบน Visual Studio Code จากนั้นพิมพ์ “npm install -g firebase-tools” ลงไปโดยทำการลือกอินผ่านเทอร์มินัล เช่น เดิมด้วยคำสั่ง “firebase login” เมื่อลือกอินเรียบร้อยขึ้น success จากนั้นทำการเริ่มต้น Firebase ด้วยคำสั่ง “firebase init” จากนั้นเลือกหัวข้อโฮสติ้งดังรูปที่ 3.41 และตอบคำถามเพื่อสร้าง “Project” ใน Visual Studio Code

```

#####  ###  #####  #####  #####  ###  #####  #####
##  ##  ##  ##  ##  ##  ##  ##  ##
#####  ##  #####  #####  #####  #####  #####  #####
##  ##  ##  ##  ##  ##  ##  ##  ##
##  #####  ##  #####  #####  ##  #####  #####

```

You're about to initialize a Firebase project in this directory:

? Which Firebase CLI features do you want to setup for this folder? Press Space to select features.

- Database: Deploy Firebase Realtime Database Rules
- Firestore: Deploy rules and create indexes for Firestore
- Functions: Configure and deploy Cloud Functions
- Hosting: Configure and deploy Firebase Hosting sites
- Storage: Deploy Cloud Storage security rules

รูปที่ 3.41 การตั้งค่าเพิ่มฟังก์ชัน Firebase ในคอมใน Visual Studio Code

```

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

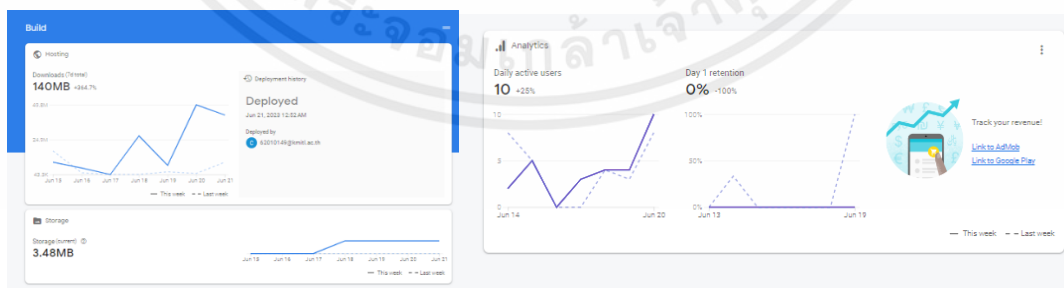
? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? Set up automatic builds and deploys with GitHub? No
? File build/index.html already exists. Overwrite? No
i Skipping write of build/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...

```

รูปที่ 3.42 การตั้งค่าของ Hosting

เมื่อสร้างเสร็จให้ทำการระบุ “Project” ที่ทำการ Deploy ลงไปโดย “firebase use --add” และทำการ Deploy ด้วย “firebase deploy” เป็นคำสั่งการ Deploy ตัวเว็บแอปพลิเคชันขึ้น Firebase โดย Firebase กำหนด Domain name ให้สามารถเข้าไปดูในหน้าที่ทำการ Deploy ได้ และสามารถเข้าไปดูสถานะการให้บริการโฮสติ้งได้ที่เว็บ Firebase ดังรูปที่ 3.43 ขึ้นมาให้สามารถดูยอดการกดเข้าชมเว็บแอปพลิเคชันได้ที่หน้าเว็บที่แจ้งมาในเทอร์มินัลตามรูปที่ 3.44



รูปที่ 3.43 การวิเคราะห์ข้อมูลบน Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 54 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

=== Deploying to 'scara-demo-melft-kmitl'...

i deploying firestore, hosting
i firestore: reading indexes from firestore.indexes.json...
i cloud.firestore: checking firestore.rules for compilation errors...
+ cloud.firestore: rules file firestore.rules compiled successfully
i firestore: deploying indexes...
+ firestore: deployed indexes in firestore.indexes.json successfully for (default) database
i firestore: latest version of firestore.rules already up to date, skipping upload...
i hosting[scara-demo-melft-kmitl]: beginning deploy...
i hosting[scara-demo-melft-kmitl]: found 39 files in build
+ hosting[scara-demo-melft-kmitl]: file upload complete
+ firestore: released rules firestore.rules to cloud.firestore
i hosting[scara-demo-melft-kmitl]: finalizing version...
+ hosting[scara-demo-melft-kmitl]: version finalized
i hosting[scara-demo-melft-kmitl]: releasing new version...
+ hosting[scara-demo-melft-kmitl]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/scara-demo-melft-kmitl/overview
Hosting URL: https://scara-demo-melft-kmitl.web.app
PS C:\Users\Asus\Desktop\Project\web\main-012>
* History restored

```

รูปที่ 3.44 การ Deploy ขึ้นหน้าเว็บ

3.5.5 หลักการสร้างคิวอาร์โค้ดเพื่อแลกเปลี่ยนข้อมูลกับชุดสาริต

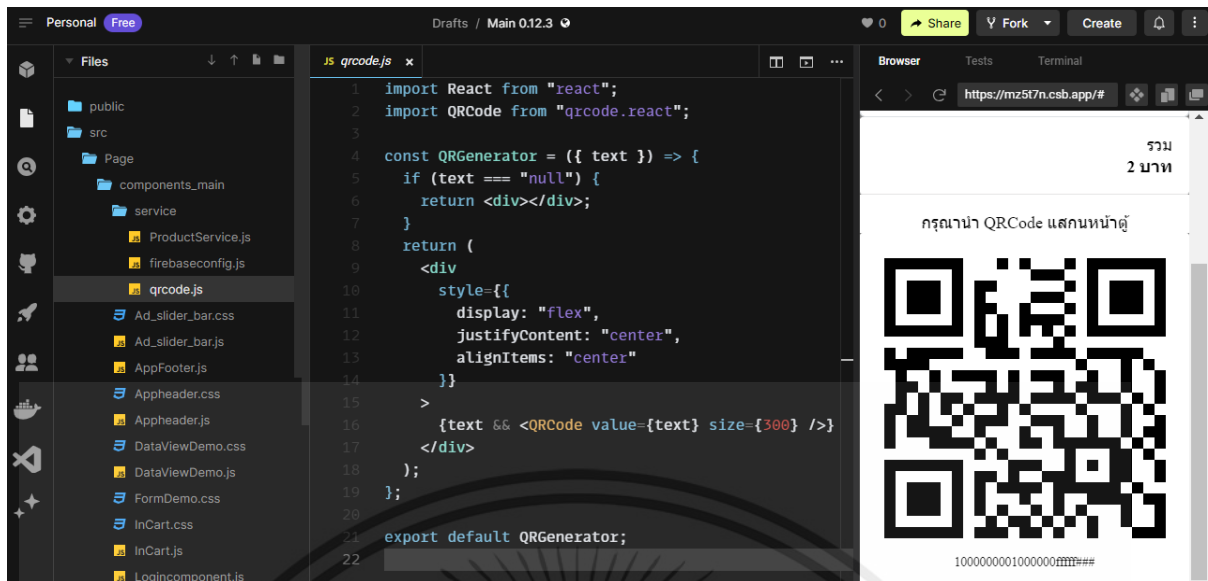
การสร้างคิวอาร์โค้ดใช้ library qrcode.react ซึ่งสามารถสร้างคิวอาร์โค้ดได้โดยการนำข้อมูลที่ต้องการแปลงเป็นคิวอาร์โค้ดที่กำหนดให้กับ value ใน `<QRCode value={text} />` ซึ่งที่นี่เป็นตัวแปร text จากนั้นตัวส่วนเสริมช่วยแปลงข้อความให้เป็นคิวอาร์โค้ด โดยกำหนดรูปแบบของคิวอาร์โค้ดส่งไปยังโค้ดรีดเดอร์ CF26 เป็นข้อมูลตัวเลขเท่านั้น โดยสิบหลักแรกเป็นข้อมูลส่วนหมายเลขที่ส่งยา โดยในสิบหลักนั้นสามารถแยกเป็นสี่หลัก ในสี่หลักแรกเป็นวิธีการจ่ายยาและอีกสิบสองหลักที่เหลือเป็นจำนวนยาแต่ละชนิดโดยจัดเรียงตามลำดับชนิด ชนิดละสองตำแหน่ง ในจำนวนยานี้ได้กำหนดจำนวนสูงสุดไว้ที่ชนิดยาละห้าเม็ด ในส่วนของในโปรแกรมเพิ่มยาเข้าตะกร้าสินค้าอีกแปดตำแหน่งท้ายเป็นชื่อผู้กำหนดเงื่อนไขเป็นตัวอักษรภาษาอังกฤษและตัวเลขห้ามใส่เกินแปดตัวอักษรหากใส่ไม่ครบแปดตำแหน่งในตำแหน่งที่ไม่ถูกใส่ข้อมูลถูกเปลี่ยนให้เป็นสัญลักษณ์ “#” แทนที่หลักที่ว่างเปล่าโดยข้อมูลทั้งหมดเป็น strings ยี่สิบสี่ตัวอักษรดังรูปที่ 3.45 เมื่อส่งข้อมูลไปสร้างคิวอาร์โค้ดคัดกรองข้อมูลเป็นการถัดไปในโค้ดรีดเดอร์ CF26



รูปที่ 3.45 รูปแบบการส่งข้อมูลด้วยคิวอาร์โค้ดรีดเดอร์ CF26

โค้ดดังรูปที่ 3.46 ในส่วนนี้เป็นการรับข้อมูล text เข้ามา จากนั้นเช็คว่ามีข้อมูลหรือไม่ หากไม่มีข้อมูลใด ๆ ใน text ให้แสดงไม่ทำการสร้างคิวอาร์โค้ด หากมีข้อมูลใน text ให้ทำการสร้างคิวอาร์โค้ดแสดงออกมา

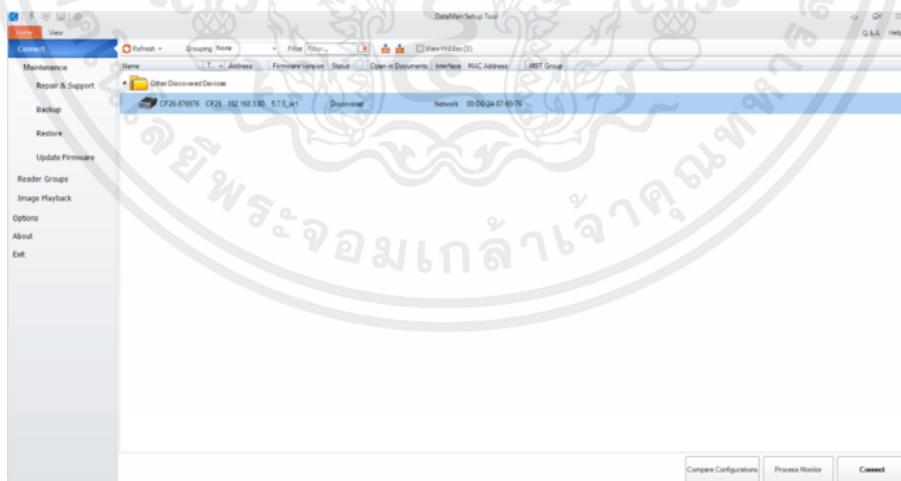
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 55 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.46 ไฟล์การสร้างคิวอาร์โค้ด

3.6 การตั้งค่าการทำงานของโค้ดรีดเดอร์

การตั้งค่าการทำงานของโค้ดรีดเดอร์ CF26 นั้นใช้ซอฟต์แวร์ Cognex DataMan Setup Tool for MELSENSOR v6.1.5 ในเว็บไซต์ของบริษัท COGNEX และใช้การเชื่อมต่อแบบ CC-Link IE field Basic ในการเชื่อมต่อกับพีแอลซี ใช้การเชื่อมต่อแบบ Ethernet ในการตั้งค่าโค้ดรีดเดอร์ CF26 ในการเริ่มต้นการตั้งค่านั้นให้ตั้งที่อยู่ IP ของคอมพิวเตอร์ให้อยู่ในวงเดียวกับโค้ดรีดเดอร์ CF26 ตัวซอฟต์แวร์แสดงชื่ออุปกรณ์ที่อยู่ IP และสถานะการเชื่อมต่อของโค้ดรีดเดอร์ CF26 ดังรูปที่ 3.47 เมื่อเชื่อมต่อกับโค้ดรีดเดอร์ CF26 ได้ให้ทำการตั้งค่าที่อยู่ IP ของโค้ดรีดเดอร์ CF26 ใหม่ให้อยู่ในวงเดียวกับอุปกรณ์อื่น ๆ ในชุดสาธิต

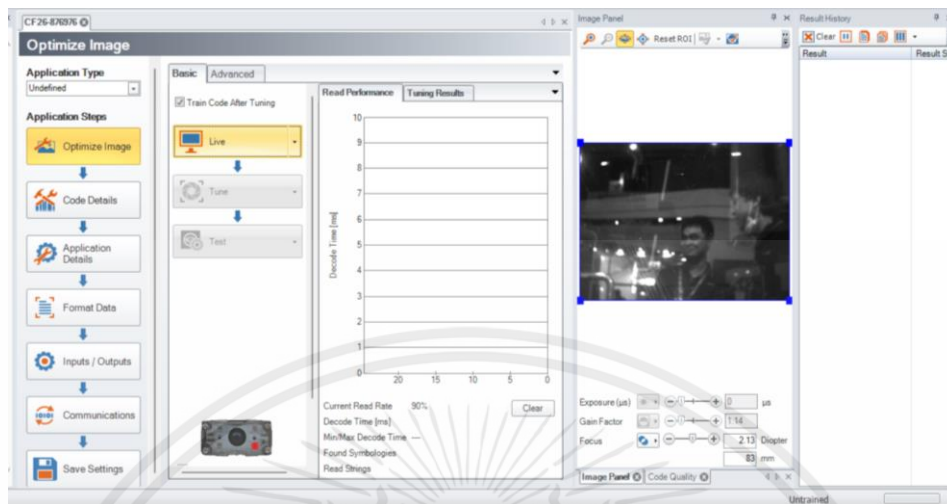


รูปที่ 3.47 หน้าซอฟต์แวร์การเลือกอุปกรณ์เชื่อมต่อ

จากนั้นปรับพิกัดโค้ดรีดเดอร์ CF26 ด้วยปุ่ม “Tune” ที่อยู่ในโปรแกรมจากนั้นนำคิวอาร์โค้ดที่ต้องการสแกนมาสแกนที่โค้ดรีดเดอร์ CF26 ค้างไว้ที่จนกว่า CF26 ปรับให้รูปชัดขึ้นโดยโปรแกรมบันทึกค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 56 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

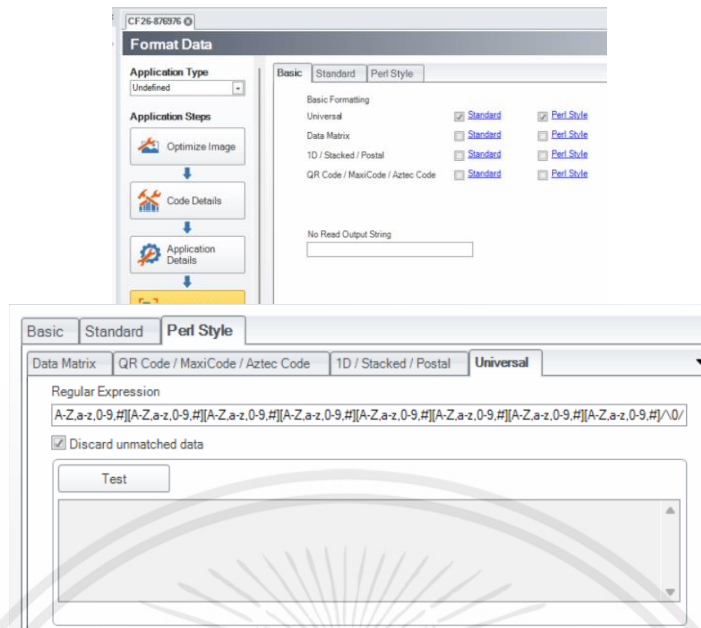
ชุดตัวแปรการตั้งค่า จากนั้นให้เลือกชุดตัวแปรที่ต้องการและเลือกสถานะการติดตั้งของไฟที่โค้ดรีดเดอร์ CF26 ที่ต้องการได้ดังรูปที่ 3.48



รูปที่ 3.48 ฟังก์ชันการถ่ายทอดสดของตัวโค้ดรีดเดอร์

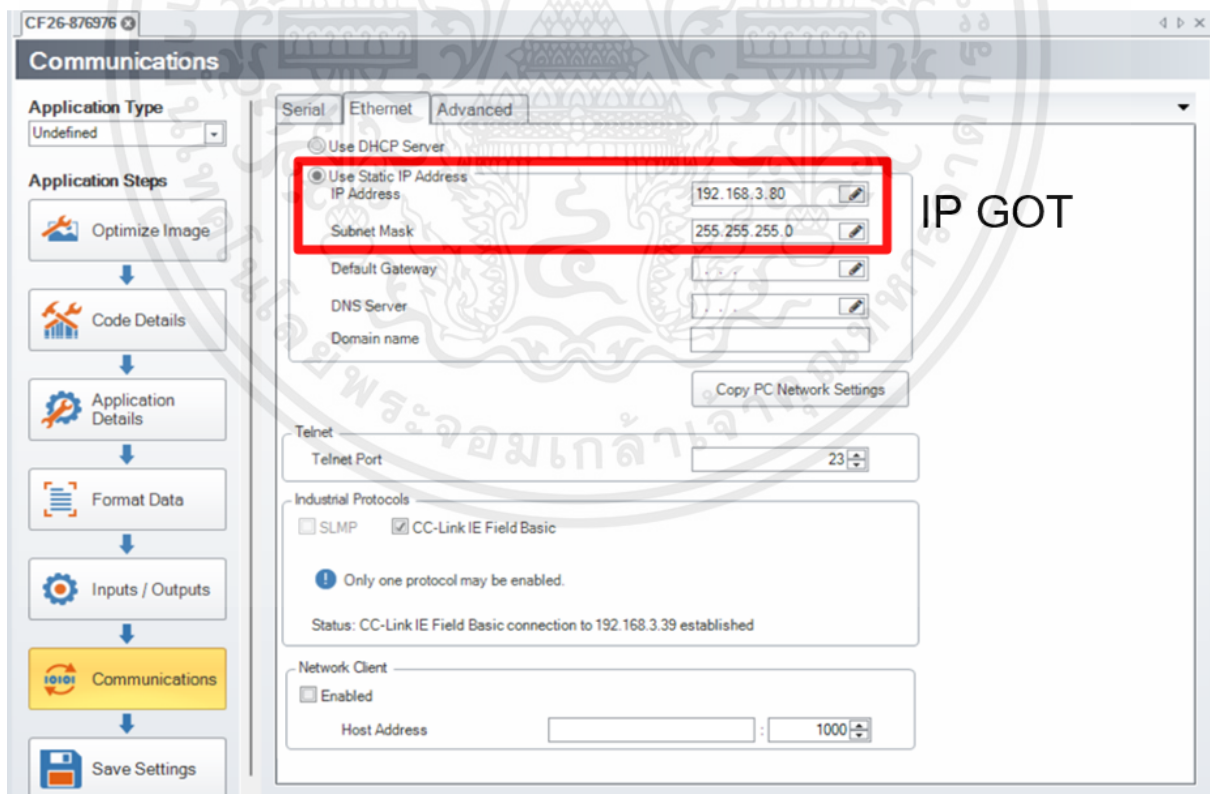
ตัวโค้ดรีดเดอร์ CF26 สามารถคัดกรองเลือกเฉพาะข้อมูลที่ต้องการได้ด้วยฟังก์ชัน Perl Style ที่สามารถเขียนโปรแกรมคัดกรองข้อมูลด้วยภาษา JavaScript ในการสร้างเงื่อนไขนั้นสัญลักษณ์ “[]” หมายถึงบิตนั้น ๆ ตามตำแหน่งที่จัดเรียงโดยสามารถเขียนรับเฉพาะอักขรที่ต้องการได้ภายในสัญลักษณ์ “[]” จากนั้นให้เลือก Discard unmatched data เพื่อคัดกรองข้อมูลให้ตรงกับรูปแบบที่ต้องการโดยมีตัวอย่างการใช้งาน Perl Style ดังรูปที่ 3.49

Regular Expression : /[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z,a-z,0-9,#][A-Z,a-z,0-9,#][A-Z,a-z,0-9,#][A-Z,a-z,0-9,#][A-Z,a-z,0-9,#][A-Z,a-z,0-9,#][A-Z,a-z,0-9,#][A-Z,a-z,0-9,#][A-Z,a-z,0-9,#]^/



รูปที่ 3.49 รูปแบบค่าที่รับจากคิวอาร์โค้ด

สามารถตั้งค่าในการส่งข้อมูลให้ GOT และสามารถบันทึกรูปลง SD card ได้ตามจำนวนที่ตั้งค่าใน ส่วนของ Communication และตั้งค่าที่อยู่ IP ของ GOT ได้ตามรูปที่ 3.50



รูปที่ 3.50 ที่อยู่ IP การเชื่อมต่อ GOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 58 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

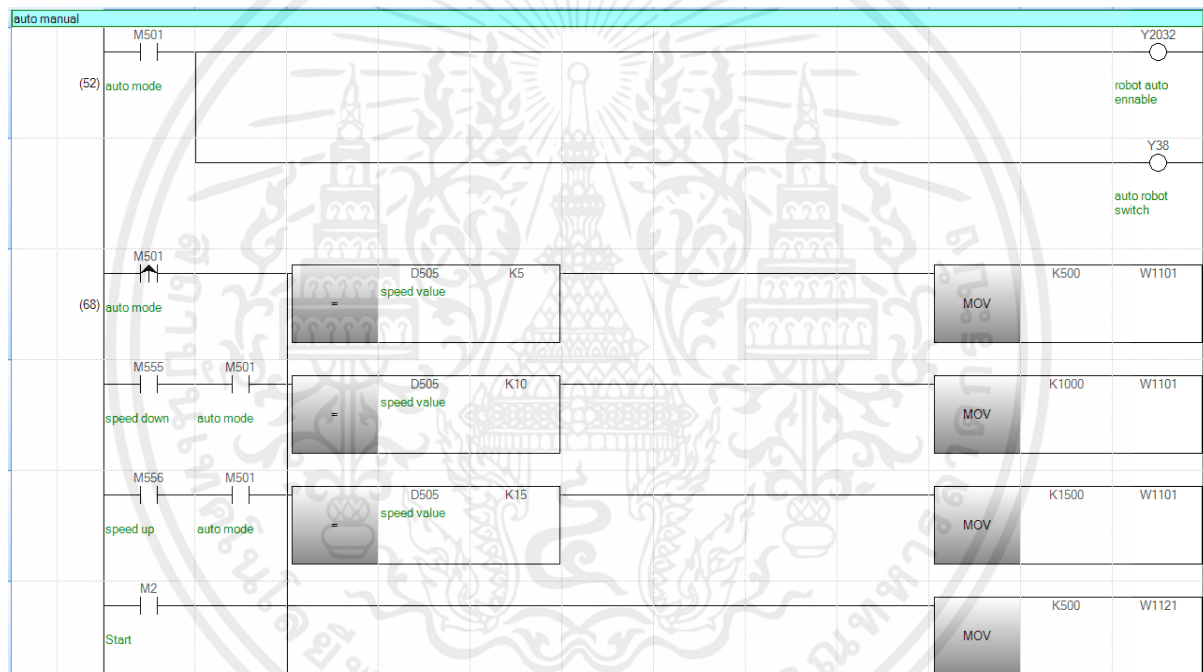
3.7 การเขียนโปรแกรมพีแอลซี

3.7.1 การควบคุมหลักในการเปิดและปิดหรือเปลี่ยนโหมดการทำงานของชุดสาธิต

โปรแกรมควบคุมหลักเป็นส่วนสำคัญในการควบคุมชุดสาธิตโดยการควบคุมมีสองโหมดคือโหมดอัตโนมัติและโหมดแมนนวล

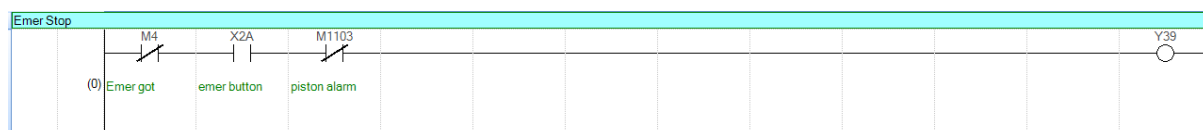
โหมดแมนนวลไม่สามารถกดเริ่มโปรแกรมหุ่นยนต์ผ่านทางจอ GOT ได้ หากต้องการเริ่มการทำงานของหุ่นยนต์ต้องใช้ Teaching Box เท่านั้น ในโหมดนี้สามารถควบคุมเอาต์พุตผ่านจอ GOT ได้อย่างอิสระ เช่น การควบคุมแอกชูเอเตอร์ และการควบคุมสายพาน

โหมดอัตโนมัติสามารถเริ่มโปรแกรมหุ่นยนต์ สายพานและแอกชูเอเตอร์ทั้งหมดโดยทำตามโปรแกรมลำดับที่ได้ตั้งค่าไว้ ในโหมดนี้ไม่สามารถทดสอบสั่งการทำงานต่าง ๆ ผ่านจอ GOT ได้เพื่อป้องกันการงานผิดพลาด



รูปที่ 3.51 โปรแกรมโหมดอัตโนมัติและโหมดแมนนวล

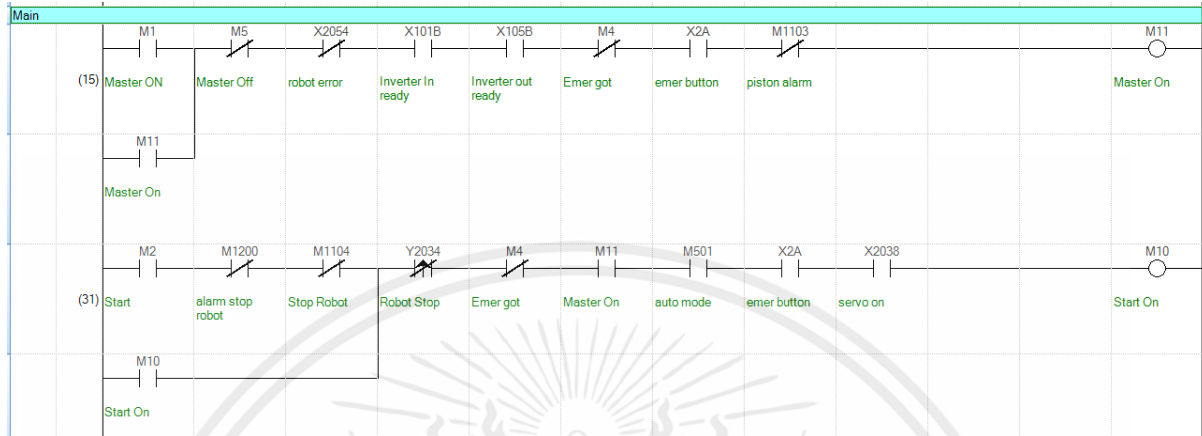
จากรูปที่ 3.51 การเปลี่ยนเข้าโหมดอัตโนมัติทำการเปิด Y38 ที่เป็นรีเลย์ที่ต่อสายการควบคุมโหมดการทำงานของหุ่นยนต์ การปรับเข้าโหมดอัตโนมัติเป็นการเปลี่ยนค่าความเร็วของสายพานเป็นแบบที่ตั้งไว้สามารถเปลี่ยนได้ที่เอชเอ็มไอหน้า Main Control มี 3 ค่าคือ 5 10 15 Hz ส่วนสายพานฝั่งกล่องถูกล็อกไว้ที่ 5 Hz ไม่สามารถเปลี่ยนได้



รูปที่ 3.52 โปรแกรมสั่งหยุดการทำงานของหุ่นยนต์ฉุกเฉิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 59 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.52 โปรแกรมนี้เป็นเงื่อนไขในการทำให้หุ่นยนต์เกิดการหยุดทำงานฉุกเฉินมี 3 เงื่อนไข คือ จากปุ่มหยุดฉุกเฉินในจอ GOT จากปุ่มหยุดทำงานฉุกเฉิน และจากแจ้งเตือนการทำงานของกระบอกสูบ ติดขัดทราบได้จากการสั่งงานกระบอกสูบกับสัญญาณรีดสวิตช์ (Reed switch) ไม่สอดคล้องกัน



รูปที่ 3.53 เงื่อนไขที่กำหนดในการ Master On และ Start

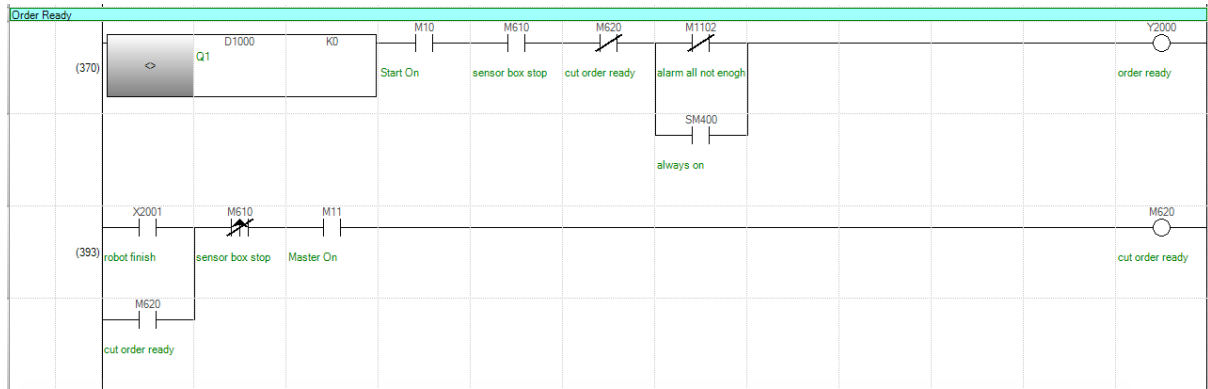
จากรูปที่ 3.53 การจะกด Master On ได้เครื่องจำเป็นต้องอยู่ในสภาพที่พร้อมใช้งาน โดยมีเงื่อนไข คือ การกด Start เปิดการเปิดการทำงานของโปรแกรมหุ่นยนต์ได้เครื่องจำเป็นต้องเป็นไปตามเงื่อนไขคือ

1. อยู่ในโหมดออโต้
2. หุ่นยนต์มีการส่ง servo on ตอบกลับมา
3. ประตูชุดสาธิตต้องไม่เปิดอยู่



รูปที่ 3.54 โปรแกรมลำดับในการเริ่มโปรแกรมหุ่นยนต์

จากรูปที่ 3.54 ขั้นตอนการเริ่มโปรแกรมหุ่นยนต์ต้องทำตามลำดับคือ สั่งขอสิทธิการควบคุมหุ่นยนต์ op right รอคำตอบ op right ของหุ่นยนต์กลับ สั่ง servo on รอหุ่นยนต์ตอบ servo on กลับ และสุดท้ายสั่งหุ่นยนต์ Start หุ่นยนต์ หากหุ่นยนต์สามารถเริ่มโปรแกรมได้ตามปกติหุ่นยนต์จะส่ง operating กลับมา



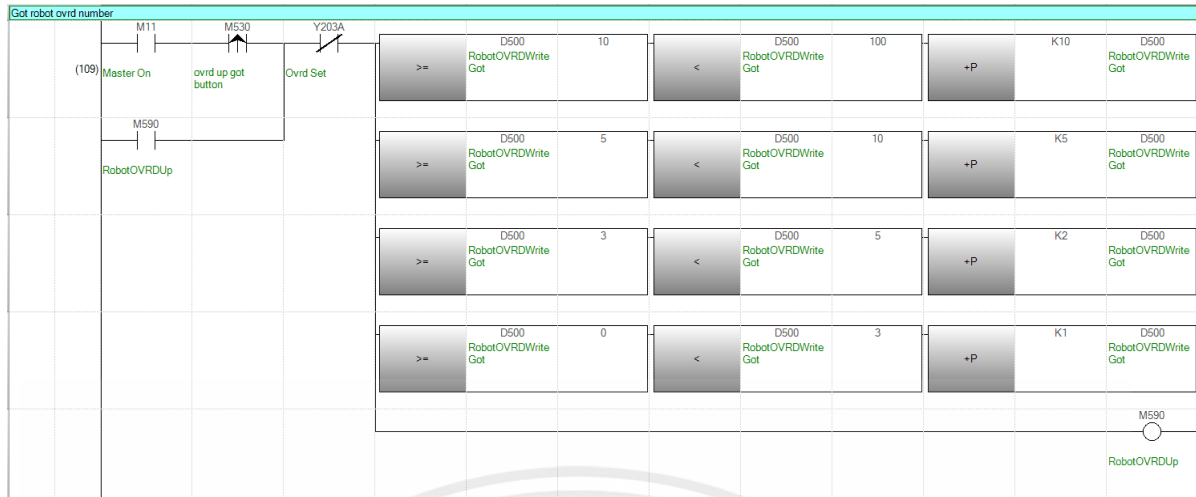
รูปที่ 3.55 การส่งค่าคำสั่งซื้อพร้อมให้กับหุ่นยนต์

จากรูปที่ 3.55 เงื่อนไขในการส่งคำสั่งซื้อพร้อมให้กับหุ่นยนต์คือ ต้อง Start อยู่และมีกล่องหยุดที่เซนเซอร์หยุดกล่อง โดยหลังจากหุ่นยนต์ทำงานเสร็จกล่องยังอยู่ที่เซนเซอร์อยู่แต่ไม่ต้องการส่งค่า คำสั่งซื้อพร้อมให้จึงให้การเสร็จโปรแกรมของหุ่นยนต์ทำการห้าม คำสั่งซื้อพร้อมไว้จนกว่าเจอขอบขาขึ้นของเซนเซอร์กล่องหยุด หรือก็คือมีกล่องใหม่เข้ามาการทำงานของเอชเอ็มไอที่เชื่อมกับการทำงานส่วนต่าง ๆ ของชุดสาคิต รวมถึงการป้อนข้อมูลและคำนวณค่าต่าง ๆ ที่แสดงผลในหน้าจอเอชเอ็มไอ



รูปที่ 3.56 โปรแกรมนับเวลา

จากรูปที่ 3.56 ฟังก์ชันการนับเวลาทำงานโดยใช้ SM412 เป็นตัวที่หน้าสัมผัสที่มีการปิดเปิดทุก 0.5 วินาทีหรือเปิดทุก 1 วินาทีทำให้สามารถนำมาใช้ในการนับวินาทีได้โดยรวมกับคำสั่ง INC ซึ่งเป็นการบวก 1 ทุก ๆ ครั้งที่ได้สัญญาณ หากค่าวินาทีที่มีค่ามากกว่า 60 ใช้คำสั่ง MOV นำค่า 0 เข้าไปแทนที่และบวกเวลานาทีเป็น 1



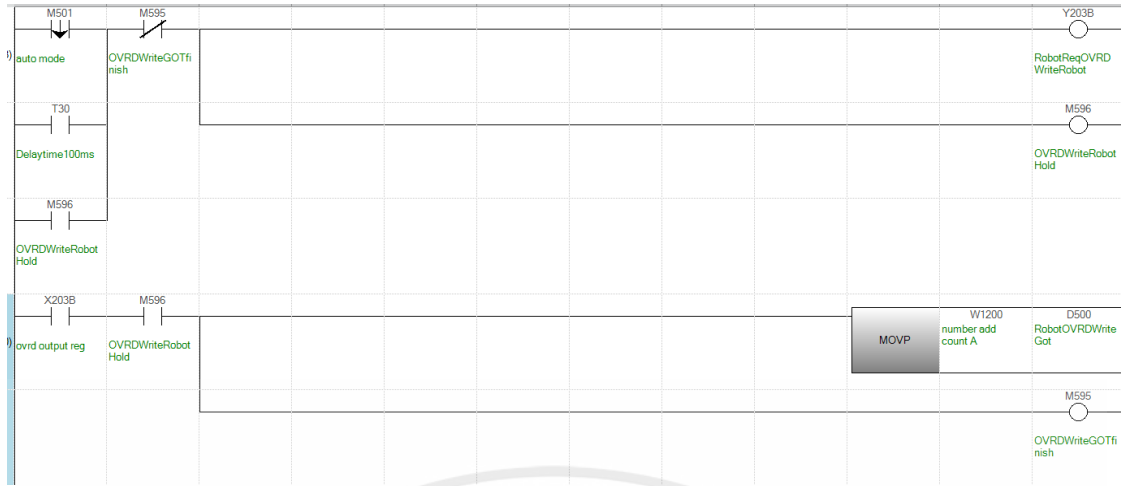
รูปที่ 3.57 โปรแกรมในการเพิ่มลดค่าจากปุ่มใน GOT

จากรูปที่ 3.57 ในการเพิ่มและลดค่า OVRD หรือความเร็วสูงสุดของหุ่นยนต์มีค่าให้เลือกดังนี้ 1 2 3 5 10 20 และเพิ่มขึ้นทีละ 10 จนถึง 100



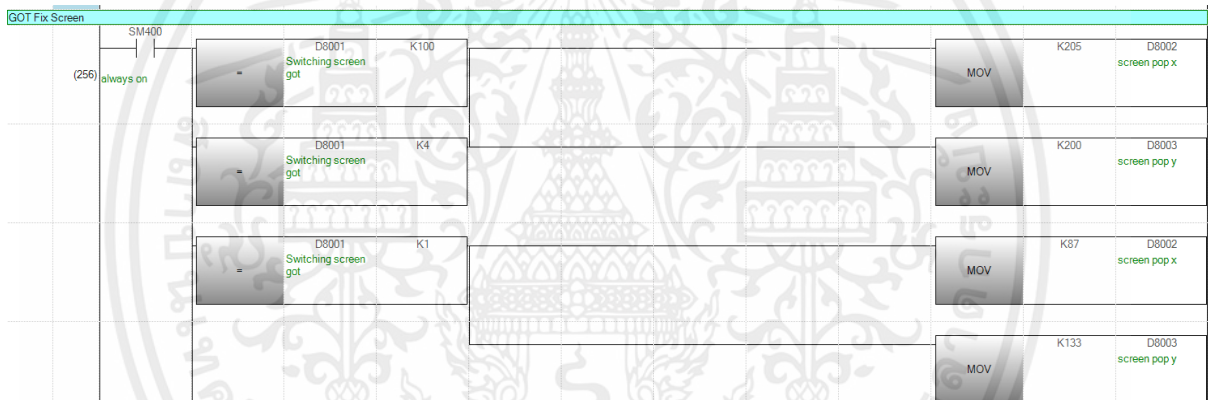
รูปที่ 3.58 โปรแกรมในการตั้งค่าความเร็วหุ่นยนต์

จากรูปที่ 3.58 ทุก ๆ ครั้งหลังจากกดเพิ่มลดความเร็วพีแอลซีส่งค่าและคำสั่งกำหนดค่าความเร็วในกับหุ่นยนต์เป็นเวลา 0.1 วินาที

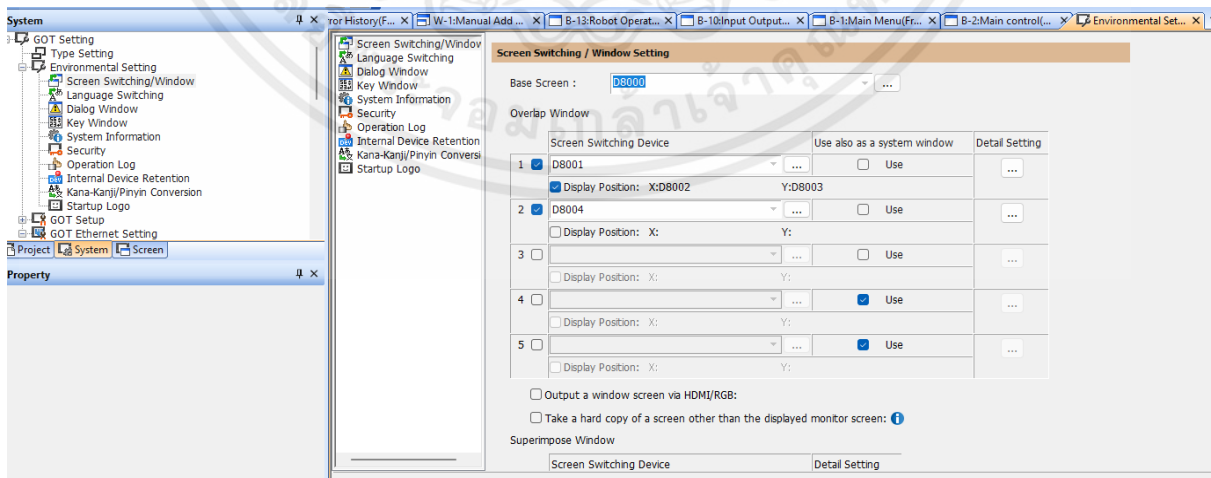


รูปที่ 3.59 โปรแกรมอ่านค่าความเร็วหุ่นยนต์

จากรูปที่ 3.59 หลังจากที่เกิดเปลี่ยนความเร็วแล้วพีแอลซีอ่านค่าความเร็วของหุ่นยนต์กลับมาเพื่อ ยืนยันว่าได้มีการเพิ่มหรือลดความเร็วจริงหรือไม่



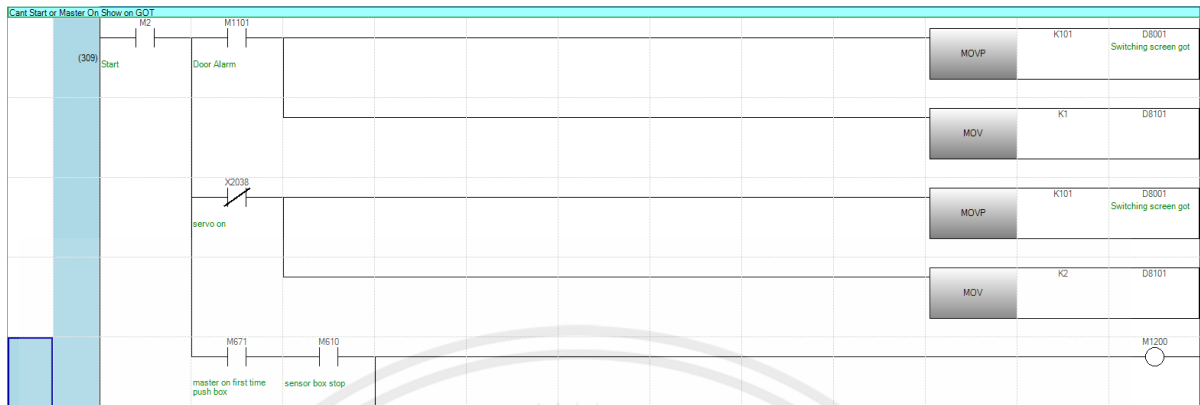
รูปที่ 3.60 โปรแกรมในการกำหนดค่าให้กับหน้าต่างพ็อบอัพในจอ GOT



รูปที่ 3.61 การตั้งค่าพ็อบอัพให้สามารถสั่งเปิดผ่านโปรแกรมพีแอลซีได้

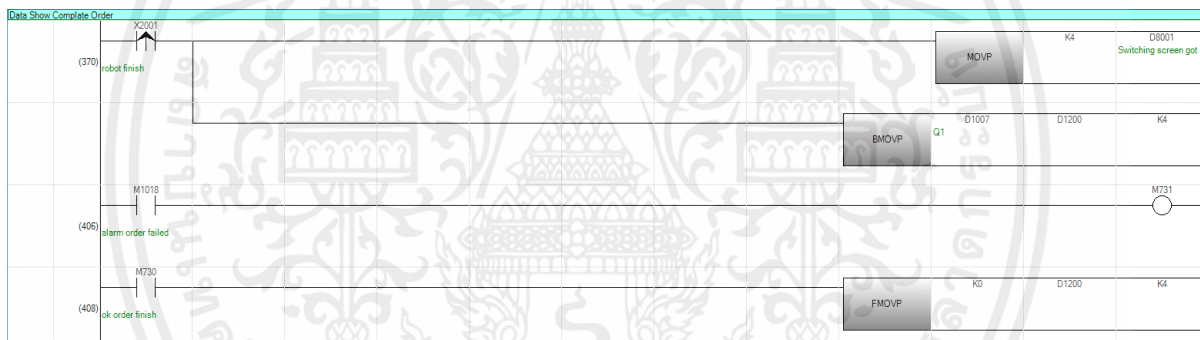
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 63 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากเปิดใช้งานฟังก์ชันนี้ในรูปที่ 3.61 จำเป็นต้องบอกตำแหน่งกำกับเพื่อให้พ็อบอัปขึ้นที่ตรงกลางของหน้าจอตามรูปที่ 3.60



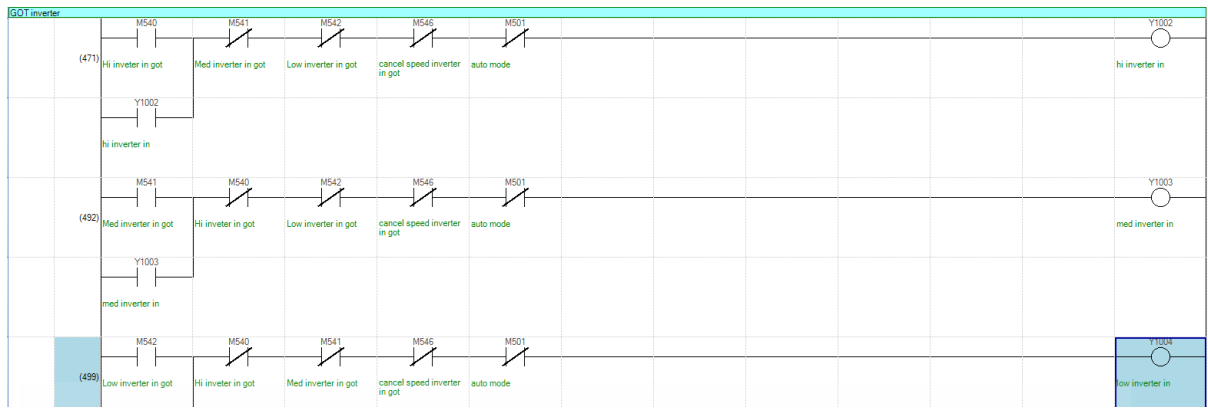
รูปที่ 3.62 โปรแกรมแสดงพ็อบอัปกรณีที่ไม่สามารถกด Start ได้

จากรูปที่ 3.62 โปรแกรมเขียนไว้ว่าหากกด Start แล้วเจอข้อห้ามต่าง ๆ ให้แสดงพ็อบอัปและข้อความขึ้นมา



รูปที่ 3.63 โปรแกรมแสดงพ็อบอัปแสดงหลังจากเสร็จคำสั่งถือว่าเป็นสำเร็จหรือไม่

จากรูปที่ 3.63 เป็นโปรแกรมแสดงพ็อบอัปแสดงหลังจากเสร็จคำสั่งถือว่าเป็นสำเร็จหรือไม่ โดยพ็อบอัปนี้ผุดขึ้นหลังจากหุ่นยนต์สกราดำเนินการปิดฝากล่องและส่งสัญญาณ X2001 เพื่อบอกว่าหุ่นยนต์ทำงานเสร็จสิ้นแล้ว



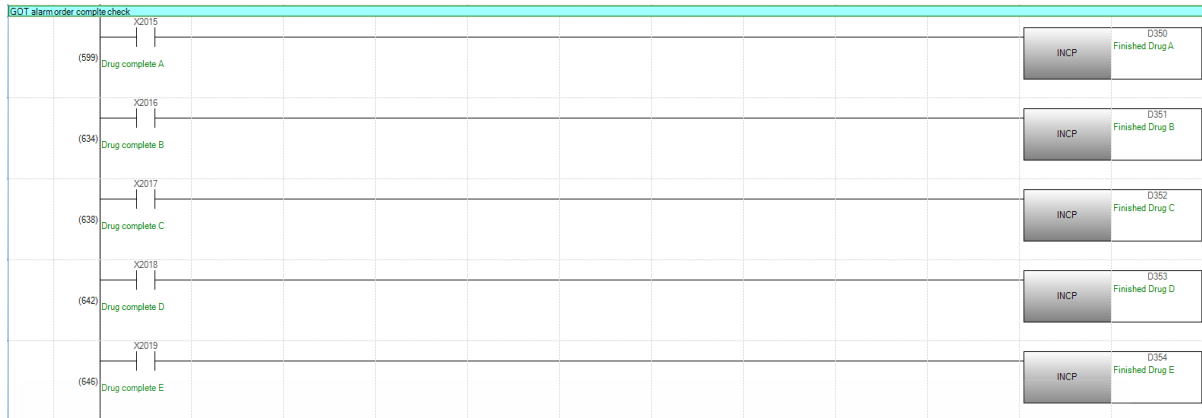
รูปที่ 3.64 โปรแกรมกดปรับความเร็วสายพานผ่านจอ GOT

จากรูปที่ 3.64 เป็นโปรแกรมที่สามารถปรับความเร็วได้ 4 ความเร็ว คือ ความเร็วสูง ความเร็วกลาง ความเร็วต่ำ และความเร็วฐาน โดยความเร็วฐานเป็นความเร็วที่กำหนดค่าให้ ส่วนความเร็วสูงกลางต่ำเป็นความเร็วที่ต้องตั้งค่าพารามิเตอร์ในอินเวอร์เตอร์โดยมีค่าพื้นฐานคือ 50Hz 30Hz และ 10Hz



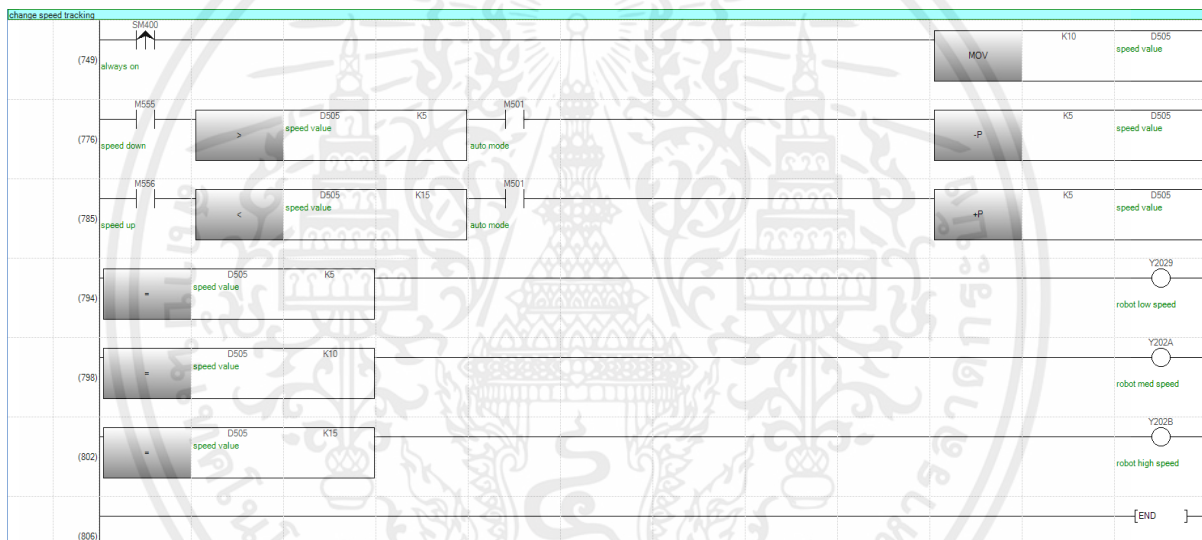
รูปที่ 3.65 โปรแกรม Stock Inventory

จากรูปที่ 3.65 โปรแกรมในส่วนของ Stock Inventory มีฟังก์ชันหลัก ๆ คือการตั้งค่าได้ว่า ในตอนนี้มียาแต่ละชนิดและกล่องอยู่เท่าใด สามารถกด Set Max เพื่อปรับให้เต็มได้และสามารถกด Clear Used เพื่อลบจำนวนครั้งที่เข้าไปแล้วได้



รูปที่ 3.66 โปรแกรมตรวจสอบคำสั่งซื้อ

จากรูปที่ 3.66 โปรแกรมทำสัญญาณจากหุ่นยนต์ที่ส่งมาว่ามียาไหนเสร็จบ้างมาทำการนับและนำขึ้นจอ GOT รวมทั้งมีการนำค่ามาลบกับจำนวนที่สั่งเพื่อดูว่าในกรณีที่ยาไม่ครบว่าขาดชนิดไหนบ้าง

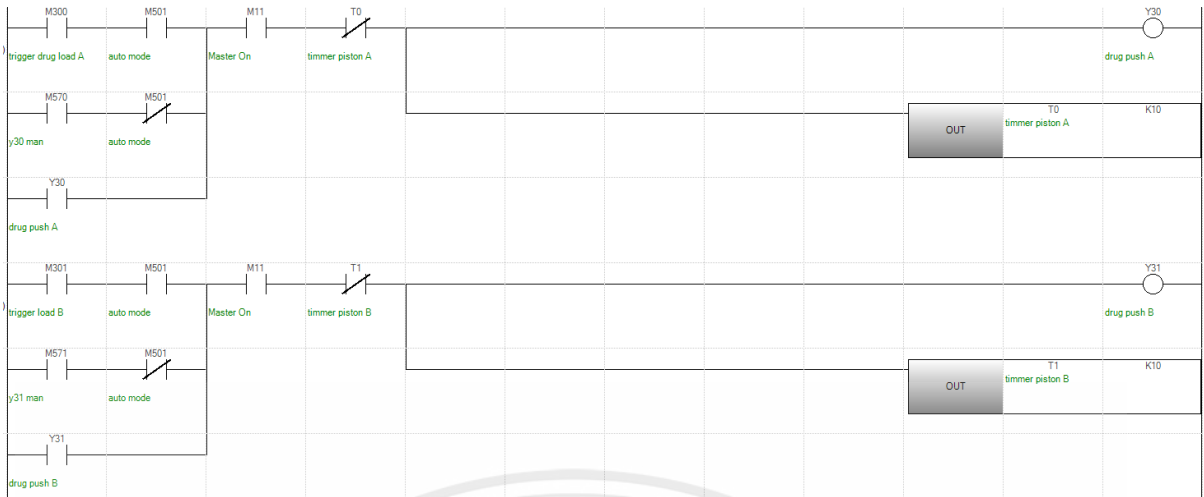


รูปที่ 3.67 โปรแกรมกำหนดความเร็วอินเวอร์เตอร์

จากรูปที่ 3.67 ในหน้าจอ GOT มีปุ่มเพิ่มและลดความเร็ว โดยสามารถเพิ่มลดได้ 3 ค่า คือ 5 10 และ 15 Hz และส่งข้อมูลระดับความเร็วไปให้หุ่นยนต์เพื่อให้หุ่นยนต์ปรับค่าต่าง ๆ ให้เหมาะสมกับความเร็ว

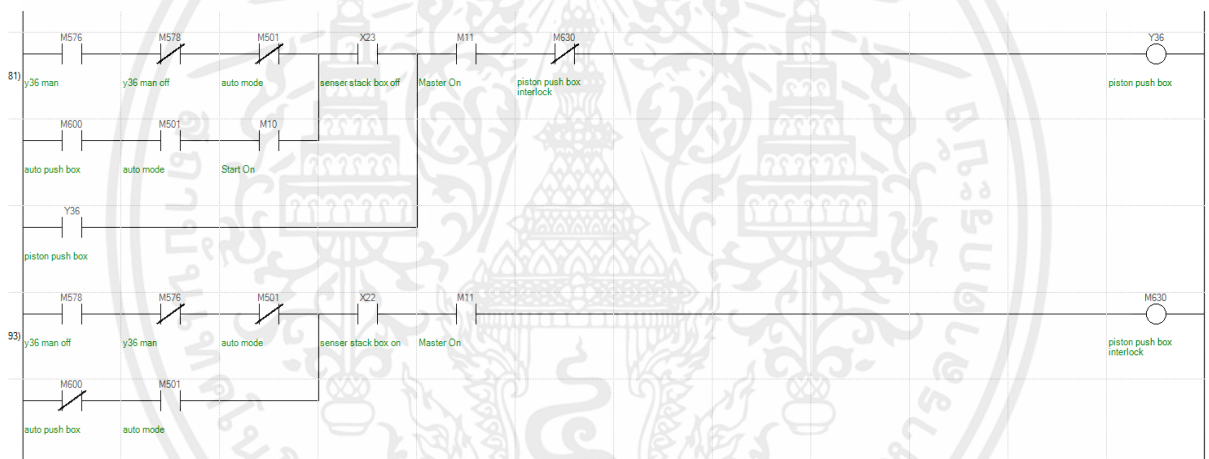
3.7.2 การควบคุมโซลินอยด์วาล์วเพื่อควบคุมกระบอกสูบในกลไกต่าง ๆ

ในส่วนนี้เป็นโปรแกรมส่วนที่รวมการส่งเอาต์พุตต่าง ๆ เนื่องจากการส่งเอาต์พุตไม่สามารถส่งซ้ำ 2 ตำแหน่งได้หรือ Double Coil ได้ การรวมการสั่งการที่เดียวกันทำให้ง่ายต่อการดูโปรแกรมและแก้ไข ในโปรแกรมส่วนนี้เป็นส่วนสั่งการทำงานของวาล์วลูกสูบสายพานและคัตรีดเดอร์



รูปที่ 3.68 โปรแกรมในการสั่งตัวปล่อยยาและหน่วงด้วยตัวจับเวลา

จากรูปที่ 3.68 แอคชูเอเตอร์ที่ใช้ในการปล่อยยามีการเขียนโปรแกรมค้ำค่าและรีเซ็ตค่าด้วยตัวจับเวลา



รูปที่ 3.69 โปรแกรมควบคุมกระบอกลมตันกล่อง

จากรูปที่ 3.69 แอคชูเอเตอร์ที่ปล่อยกล่องและดันกล่องออก มีการเขียนโปรแกรมค้ำค่าและรีเซ็ตด้วยการสั่งในทิศทางตรงข้ามโดยมีเงื่อนไขในการที่จำเป็นต้องดันให้สุดถึงรีดสวิทช์ก่อน

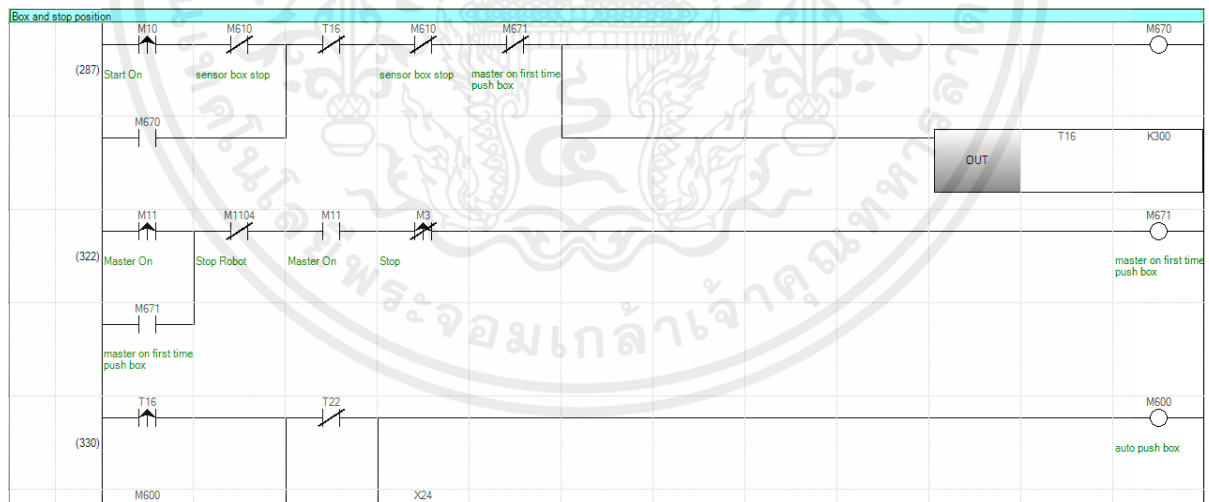


รูปที่ 3.70 โปรแกรมส่งค่าไปกลับกับหุ่นยนต์

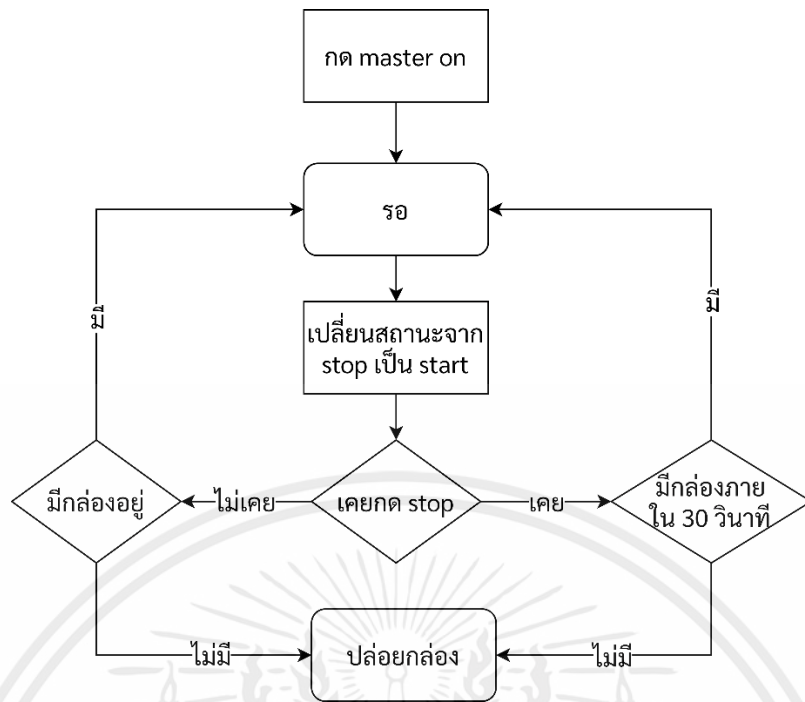
จากรูปที่ 3.70 ส่วนการรับส่งค่ากับหุ่นยนต์มีการเขียนหน่วงเวลาเล็กน้อย 0.1 วินาทีเพื่อให้ค่าของหุ่นยนต์สามารถนำมาใช้ซ้ำได้หลายรอบการสแกน

การทำงานทางกายภาพของชุดสาธิตทั้งหมดอยู่ในส่วนนี้ ส่วนที่ทำให้เกิดการเคลื่อนที่ถูกล่ามด้วย Master On ทั้งหมด เมื่อกดปุ่มหยุดทำงานฉุกเฉินด้วยเป็นการบังคับตัดการทำงานของ Master On และทำการสั่งให้หุ่นยนต์หยุดทำงานฉุกเฉินด้วย

3.7.3 การควบคุมการทำงานของสายพาน



รูปที่ 3.71 โปรแกรมลำดับในการเคลื่อนที่ของสายพานฝั่งกล่องบรรจุยา

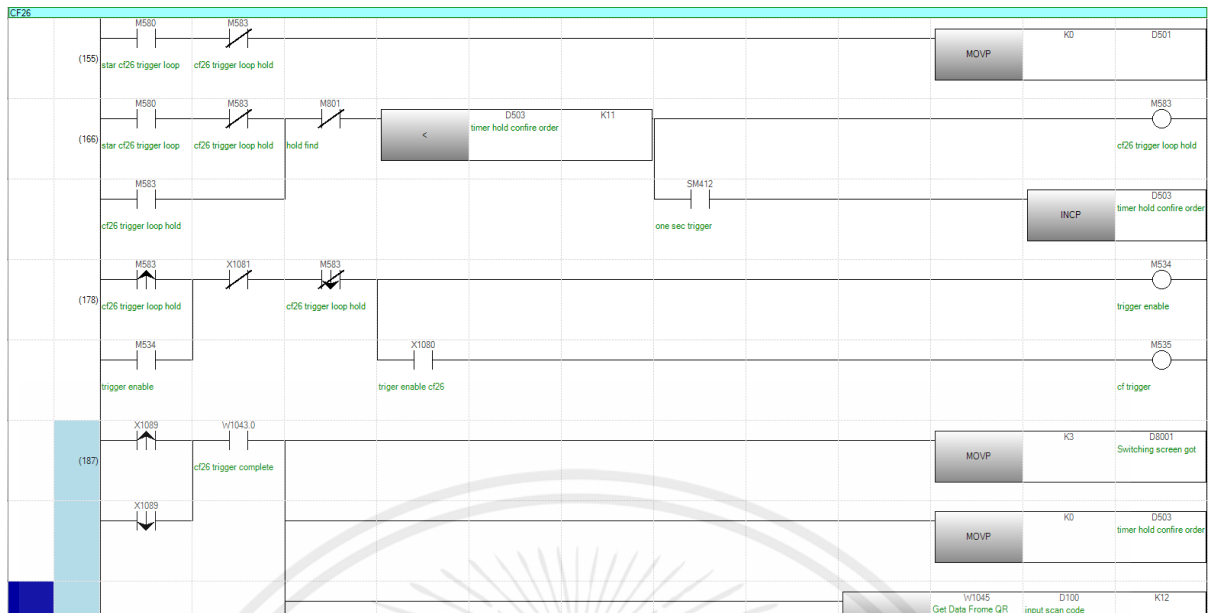


รูปที่ 3.72 ผังงานการทำงานของ การปล่อยกล่องบรรจุยา

จากรูปที่ 3.71 การทำงานของโปรแกรมนี้ มีการทำงานตามผังงานในรูปที่ 3.72 เริ่มแรกหากกด Start ตอนมีกล่องอยู่ที่เซนเซอร์หยุดชุดสาธิตจะไม่ทำงานและแจ้งว่าจำเป็นต้องเอากล่องออกก่อน หากไม่มีกล่องอยู่ที่เซนเซอร์หยุดสายพานและกด Start กล่องทำการปล่อยทันที หลังจากนั้นหากมีการเปิดประตูหรือ กด stop กด Start อีกรอบจะทำการปล่อยกล่องหลังจากผ่านไป 30 วินาทีเพื่อป้องกันการเคลื่อนที่ของกล่องที่มาทับกัน

3.7.4 การควบคุมโค้ดรีดเตอร์ CF26

อันดับแรกในการเขียนโปรแกรมคือการตั้งค่าการส่งข้อมูลระหว่างโค้ดรีดเตอร์ CF26 กับพีแอลซี โดยในชุดสาธิตนี้ใช้เป็น CC-Link IE field basic หลังจากตั้งค่าเสร็จสามารถดูจากคู่มือและทำการวางแผนได้ว่า Address อะไรของพีแอลซีใช้ในการสั่งการโค้ดรีดเตอร์ CF26



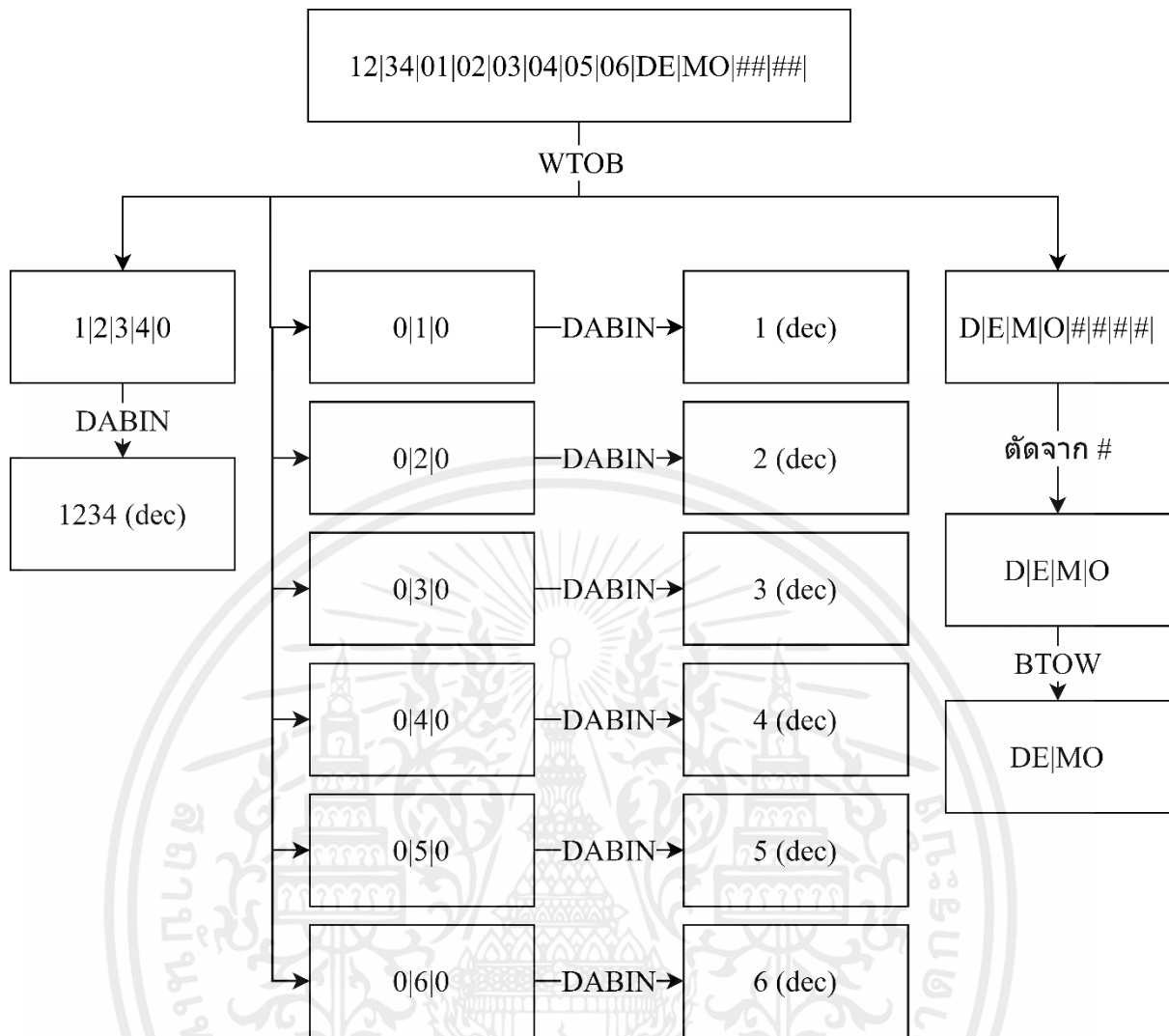
รูปที่ 3.73 โปรแกรมควบคุมไค้ดรีดเดอร์ส่วนการสั่งถ่าย

จากรูปที่ 3.73 ในการทำงานของไค้ดรีดเดอร์ CF26 ต้องสั่งการทำงาน 2 ชั้น คือ ชั้นแรกต้องสั่งเปิด bit enable เพื่อให้ไค้ดรีดเดอร์เริ่มการใช้งาน หลังจากนั้นต้องรอไค้ดรีดเดอร์ตอบกลับว่าไค้ดรีดเดอร์ enable จึงสามารถสั่งการทำงานชั้นสองคือการเริ่มถ่ายหรือคำสั่ง trigger ได้ โดยการถ่ายของไค้ดรีดเดอร์ตั้งเป็นการถ่าย 10 ครั้ง 10 วินาทีต่อการถ่ายหนึ่งครั้งไม่สามารถสั่งถ่ายแทรกได้ใน 10 วินาทีนั้น

ในโปรแกรมสร้าง ladder ที่ทำการคั่งค่าด้วยตัวจับเวลา 11 วินาทีเพิ่มทำการป้องกันการสั่งถ่ายซ้ำและใช้ขอบขาขึ้นของตัวคั่งค่าเป็นตัวในการสั่งถ่าย

หลังจากถ่ายเสร็จไค้ดรีดเดอร์ส่งค่าตอบกลับโดยการอ่านค่าตอบการสามารถเขียนโปรแกรมได้ตามรูป จากการเปิดคู่มือการใช้งาน

การถ่ายเสร็จโปรแกรมคั่งค่าตัวหนึ่งไว้เพื่อให้การถ่ายสามารถรู้ได้ว่าข้อมูลจากการถ่ายรอบนี้เป็นรอบล่าสุดหรือไม่เนื่องจากไค้ดรีดเดอร์ส่งค่าการถ่ายล่าสุดมาตลอดเวลา หากกด OK หรือ Cancel ในจอ GOT หลังสแกนเสร็จจะทำการรีเซตค่านี้



รูปที่ 3.74 การทำงานของโปรแกรมจัดการค่าจาก QR code

จากรูปที่ 3.74 โค้ดที่ได้รับมาเริ่มจากเป็น Word String คือใน 1 Address 16 Bit ของพีแอลซี ประกอบไปด้วยตัวหนังสือ 2 ตัว ขั้นตอนแรกคือทำการใช้คำสั่ง WTOB เพื่อแตกโครงสร้างข้อมูลให้กลายเป็น Byte String เพื่อจัดการข้อมูลต่อไป ตัวเลขสามารถเติม 0 ด้านหลังและใช้คำสั่ง DABIN เพื่อแปลงเป็นค่าตัวเลขได้ ส่วนชื่อจำเป็นต้องตัด “#” ออกมาเพื่อให้กลายเป็นเฉพาะข้อมูลที่ต้องการ

3.7.5 การควบคุมคิวและการปล่อยยา

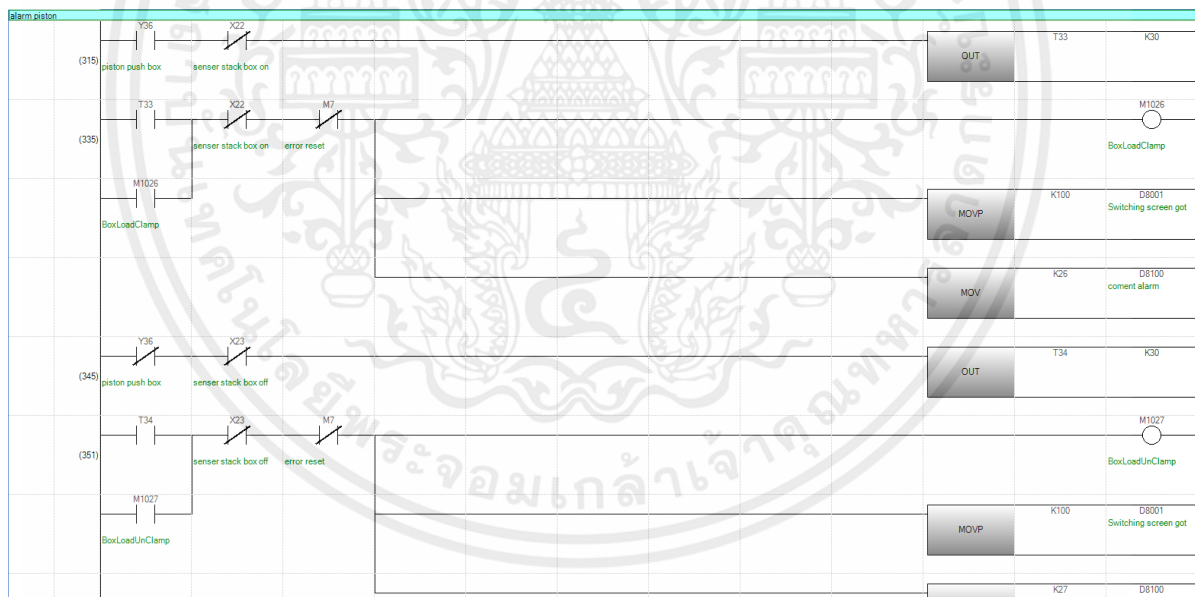
โปรแกรมส่วนของคิวมีการชี้คิวเพื่อบอกตำแหน่งที่ทำการแทรกข้อมูลเข้าไป ตัวชี้คิวมีการเพิ่มขึ้นเวลามีคำสั่งซื้อเข้ามาและลดลงเวลาคำสั่งซื้อเสร็จหรือมีการกดยกเลิก ทุกครั้งที่มีการยกเลิกหรือคำสั่งซื้อเสร็จทำการย้ายข้อมูลจากคิวที่สูงลงมาคิวที่ต่ำลง



รูปที่ 3.75 โปรแกรมในการนับจำนวนปล่อยยา

จากรูปที่ 3.75 ในส่วนโปรแกรมการปล่อยยาปล่อยยาครั้งแรกตอนหุ่นยนต์เริ่มทำงานหลังจากรับสัญญาณคำสั่งซื้อพร้อมไปและปล่อยทุก ๆ 10 วินาทีจนกว่าหมดที่ต้องการ ข้อมูลของคิวที่ 1 ก็ถูกส่งมาเพื่อใช้ในการนับและหากมีการปล่อยหรือหีบฉีดพลาดหุ่นยนต์ส่งข้อมูลกลับมาระบุว่าขาดยาอะไรบ้างก็เมื่อดส่งคำสั่งมาให้ปล่อยยาเพิ่ม ค่าของจำนวนยาที่ต้องการเพิ่มก็ถูกย้ายเข้าไปเพื่อทำการนับอีกรอบ

3.7.6 การแสดงผลแจ้งเตือน

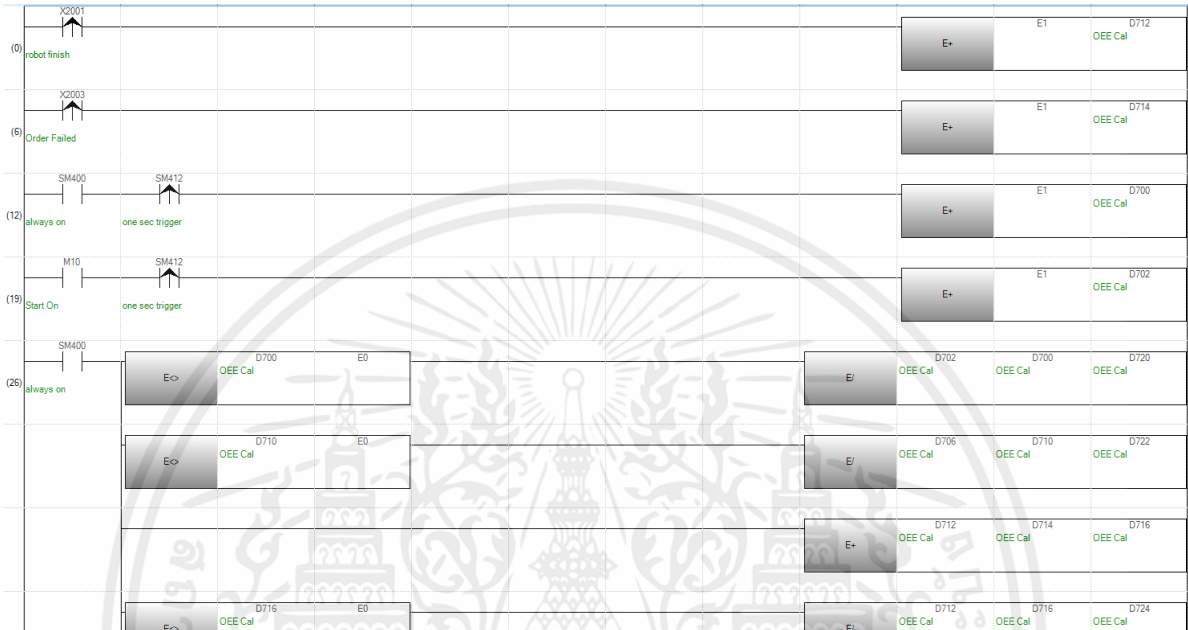


รูปที่ 3.76 โปรแกรมแจ้งเตือนหากมีแอกชูเอเตอร์ติด

จากรูปที่ 3.76 ในส่วนของการแจ้งเตือนหรือแจ้งเตือนมีฟ็อบอัปขึ้นมาแสดงและเปลี่ยนข้อความได้ โดยการ MOV K100 D8000 หมายถึงการแสดงผลหน้าจอฟ็อบอัปขึ้นมาและ MOV K26 D8100 หมายถึงการเปลี่ยนข้อความที่ขึ้นบนจอ

แจ้งเตือนที่มีการเขียนเป็นมาเพิ่มนอกเหนือจากแจ้งเตือนของระบบ มีการแจ้งเตือนระบอบกสูบติดขัดโดยเขียนโปรแกรมว่าหากมีการสั่งการระบอบกสูบแล้วรีดสวิตช์ไม่ทำงานให้ถือว่าระบอบกสูบติดหลังจากนั้นเกิดแจ้งเตือน

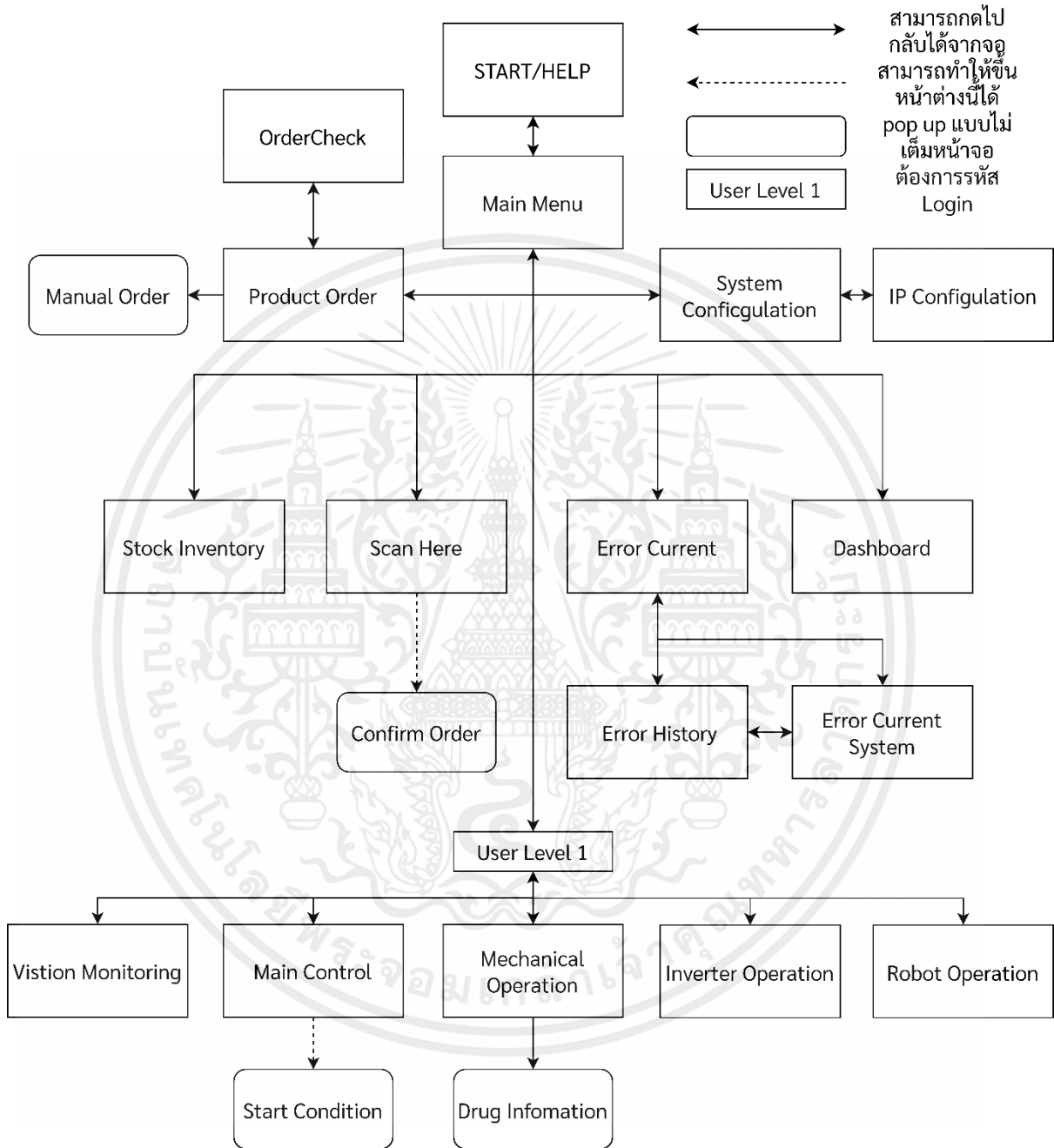
3.7.7 โปรแกรมส่วนการแสดงค่า OEE



รูปที่ 3.77 โปรแกรมคำนวณค่า OEE

จากรูปที่ 3.77 ในการคำนวณค่า OEE โดยค่า OEE คำนวณจากค่า A คือเวลาที่เครื่องทำงานหารด้วยเวลาที่เครื่องเปิดใช้งาน ค่า B คือ เวลาที่ใช้ในการทำ 1 รอบหารด้วยเวลาที่คิดว่าใช้ในการทำ 1 รอบ ค่า C คือจำนวนข้อที่ติหารด้วยจำนวนของทั้งหมด

3.8 การเขียนโปรแกรมหน้าจอเอชเอ็มไอ



รูปที่ 3.78 องค์ประกอบของเอชเอ็มไอที่จัดสร้าง

องค์ประกอบของเอชเอ็มไอที่จัดสร้างในรูปที่ 3.78 ประกอบไปด้วยสองส่วนหลักคือ

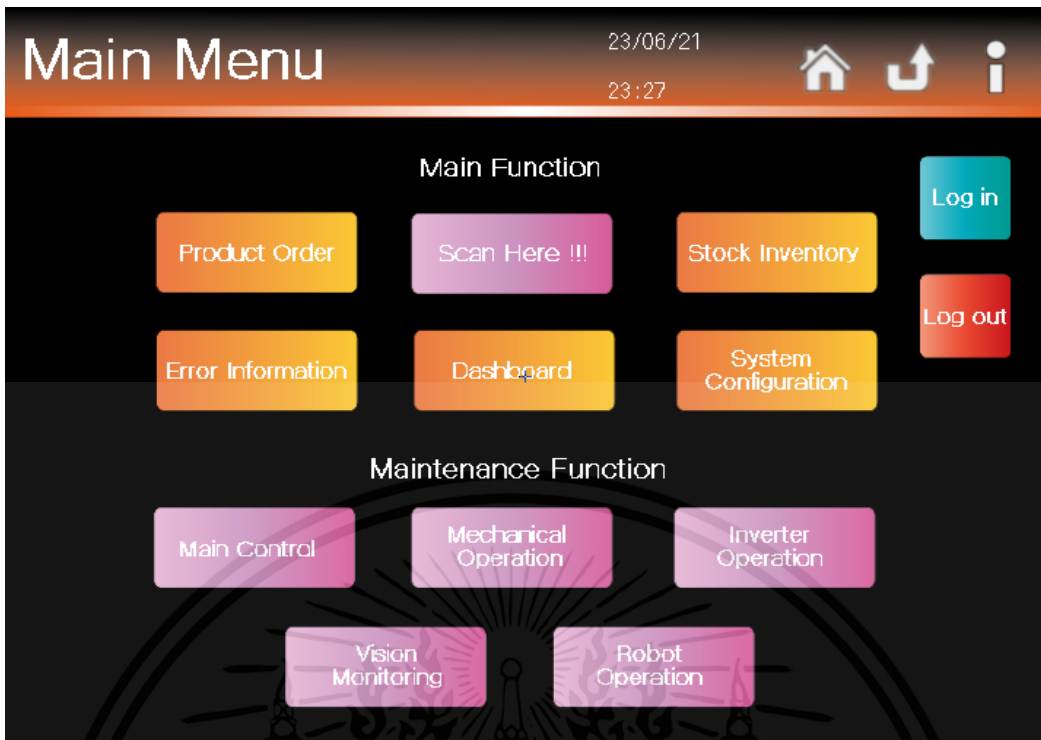
1. หน้า Main function
2. หน้า Maintenance function

ในส่วนของหน้า Main function ไม่ทำให้เครื่องมีปัญหาจากการกดผิดพลาดมากนักจึงเป็นหน้า ส่วนที่ไม่ได้จำเป็นต้องใส่รหัสผ่าน ส่วน Maintenance function หากมีการกดใช้งานผิดพลาดอาจทำให้ ชุดสาธิตเสียหายได้

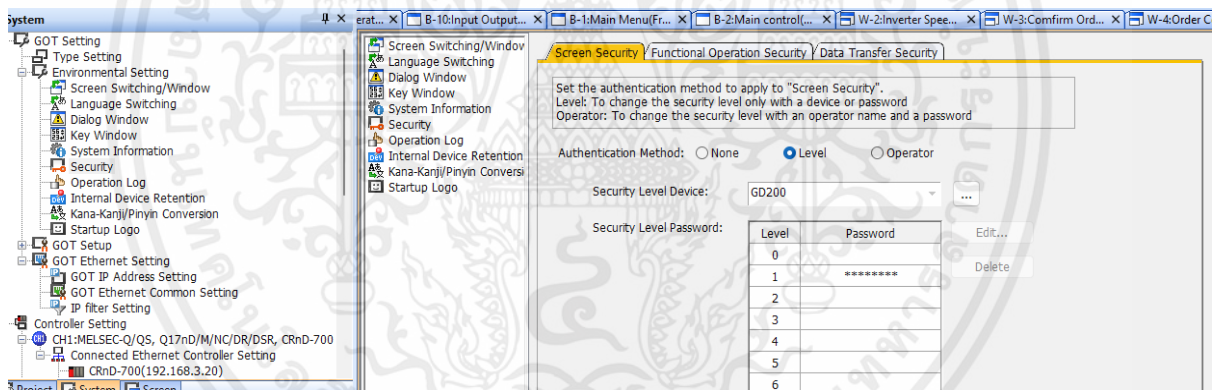


รูปที่ 3.79 หน้า Start/help

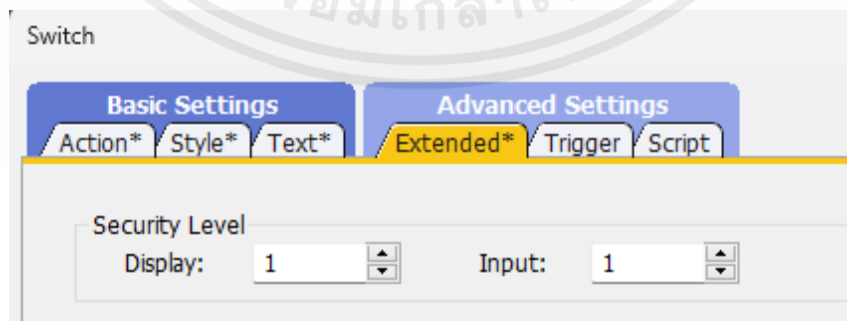
จากรูปที่ 3.79 หน้าแรกที่แสดงหลังเปิดชุดสาธิต Start/help เป็นหน้าเริ่มต้นของชุดสาธิต ประกอบไปด้วยข้อมูลเกี่ยวกับชุดสาธิตและปุ่มกดไปที่หน้า Main Menu



รูปที่ 3.80 หน้า Main Menu



รูปที่ 3.81 การตั้งรหัสผ่านของ level 1 ในซอฟต์แวร์ GT Designer3



รูปที่ 3.82 การปรับการแสดงผลและสัญญาณขาเข้าของปุ่มเป็นระดับ 1 ในซอฟต์แวร์ GT Designer3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 76 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Action (Word)

Device: ...

Data Type:

Mode:

Setting Value

Constant:

Indirect Device:

Initial Value Condition

รูปที่ 3.83 การตั้งค่าปุ่ม Log out ในซอฟต์แวร์ GT Designer3

จากรูปที่ 3.80 ในหน้า Main Menu และเป็นหน้าที่ใช้ในการเปลี่ยนไปที่หน้าต่าง ๆ ได้และมีปุ่ม Log in และ Log out เป็นจำกัดสิทธิในการทำงานของผู้ใช้ให้กับผู้ที่รู้รหัสเท่านั้นที่สามารถกดหน้า Maintenance Function ได้หากไม่ได้ Log in ทำการซ่อนหน้าต่างเอาไว้

ทำการตั้งรหัสตามรูปที่ 3.81 และซ่อนปุ่มรูปที่ 3.85 ส่วน Log out ทำได้โดยการเปลี่ยนค่าของ GD200 ที่ได้ตั้งเป็น security level device ก่อนหน้าเป็น 0 เพื่อเปลี่ยนระดับผู้ใช้ให้กลายเป็น 0 ตามรูปที่ 3.83

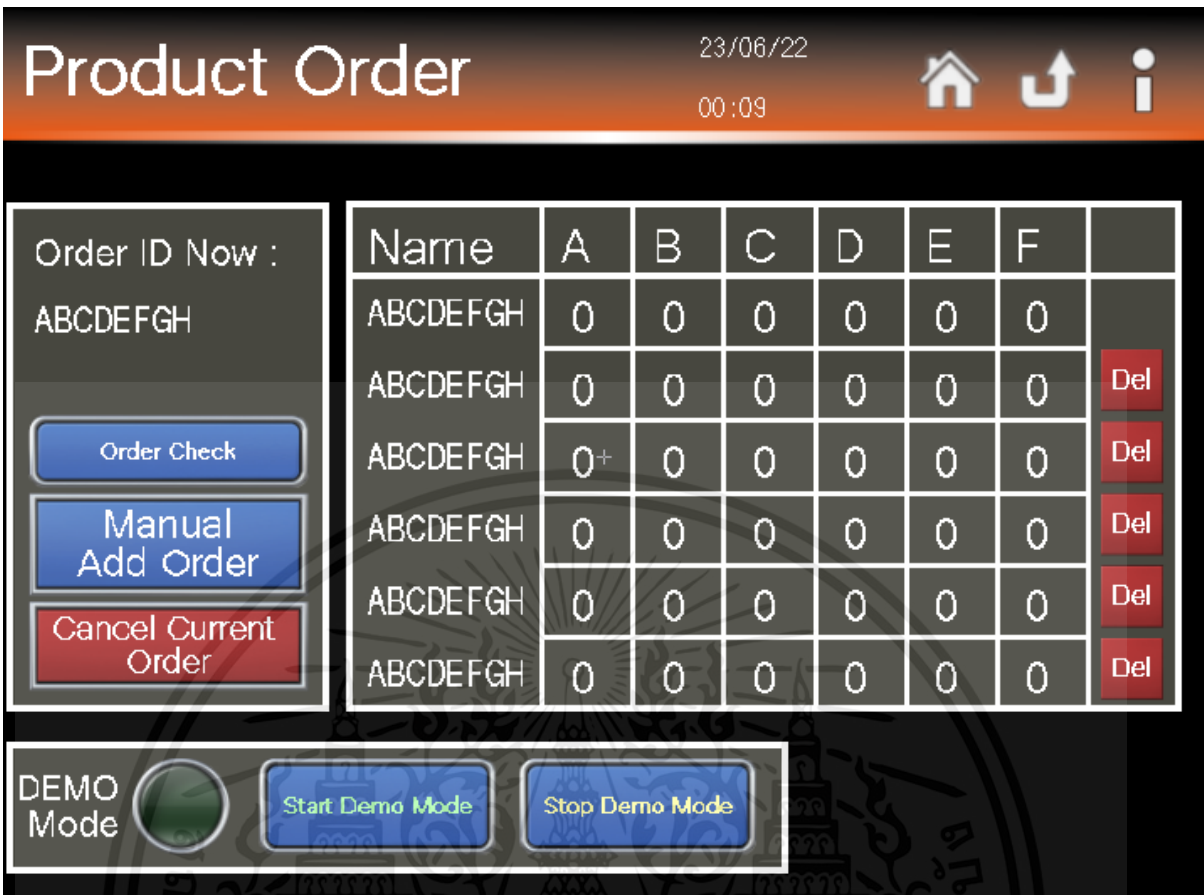
3.8.1 การใช้หน้าจอ GOT ในการควบคุมชุดสาธิต



รูปที่ 3.84 หน้า Main Control

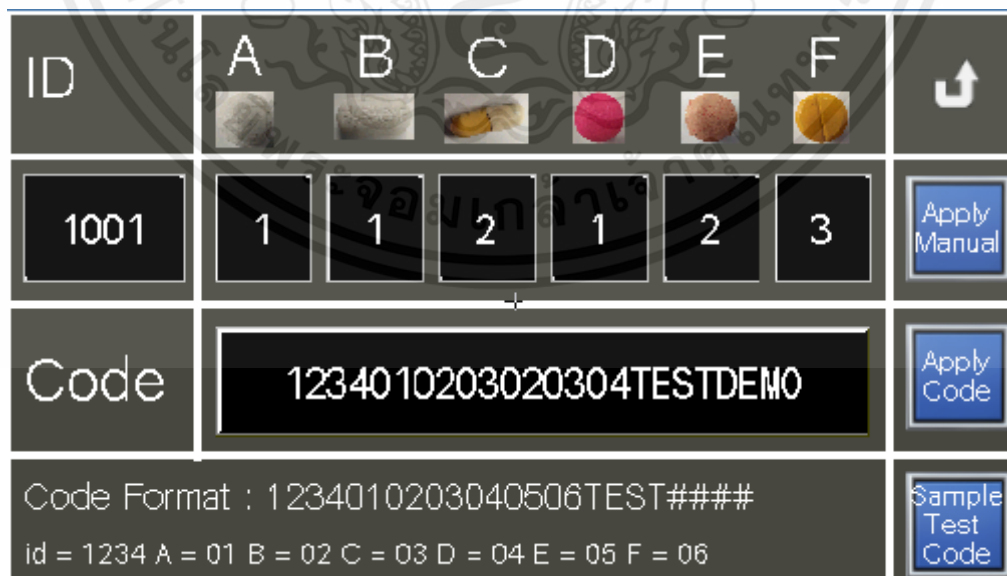
จากรูปที่ 3.84 หน้า Main Control เป็นหน้าที่สำคัญที่สุดใช้ในการเปิดเครื่องและควบคุมการทำงานต่าง ๆ มีการเปิดสองขั้นคือ

1. Master On ในการที่ Master On ได้จำเป็นต้องสภาพเครื่องพร้อมใช้คือไม่มีแจ้งเตือนที่รุนแรง ในโหมดนี้เครื่องไม่ทำงานแต่เป็นการแสดงว่าเครื่องพร้อมใช้งาน โดยเอาต์พุตเกือบทุกตัวถูกตัดไม่ทำงาน หากไม่ Master On การใช้งานทดสอบเครื่องแบบแมนนวลจำเป็นต้องกด Master On ด้วย
2. Start ปุ่มนี้กดไม่ได้ในโหมดแมนนวลต้องเป็นโหมดอัตโนมัติจึงสามารถกดได้ การกด Start ทำให้เริ่มทำลำดับการแสดงผลรวมถึงเป็นการเริ่มโปรแกรมหุ่นยนต์ด้วย



รูปที่ 3.85 หน้า Product Order

จากรูปที่ 3.85 หน้านี้เป็นอีกหนึ่งในหน้าสำคัญใช้ในการดูคำสั่งล่าสุดของชุดสาธิตสามารถเข้าไปที่หน้าตรวจสอบข้อมูลและหน้าเพิ่มคำสั่งยาแบบไม่ผ่านการสแกนได้



รูปที่ 3.86 ป็ออปป์ Manual Add Order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 79 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.86 สามารถใส่จำนวนยาแบบเลือกจำนวนเองได้และในส่วนของโค้ดตรงนี้เป็นแบบป้อนค่าและแสดงลง โดยโค้ดที่อยู่ในช่องนี้เป็นโค้ดจากการแสดงครั้งล่าสุด



รูปที่ 3.87 หน้า Scan here

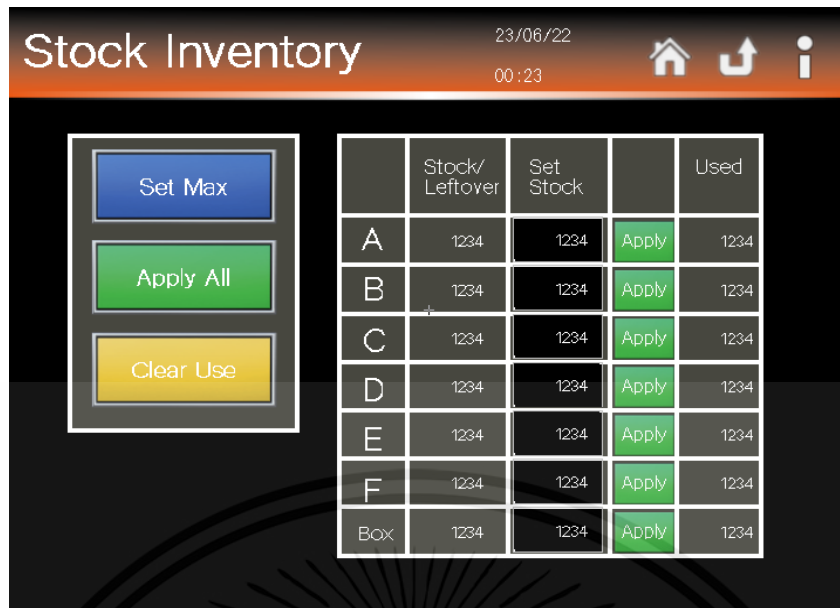
จากรูปที่ 3.87 เป็นหน้าที่ใช้ในการกดปุ่มเพื่อทำการสแกนคิวอาร์โค้ดและส่งข้อมูลในกับพีแอลซี โดยทำการสแกนต่อเนื่องหลังกดปุ่ม Start Scan 10 วินาที



รูปที่ 3.88 ป๊อปอัพ Confirm Order

จากรูปที่ 3.88 หลังจากที่โค้ดรีดเตอร์สแกนเจอขึ้นหน้าต่างให้กดยืนยันข้อมูล

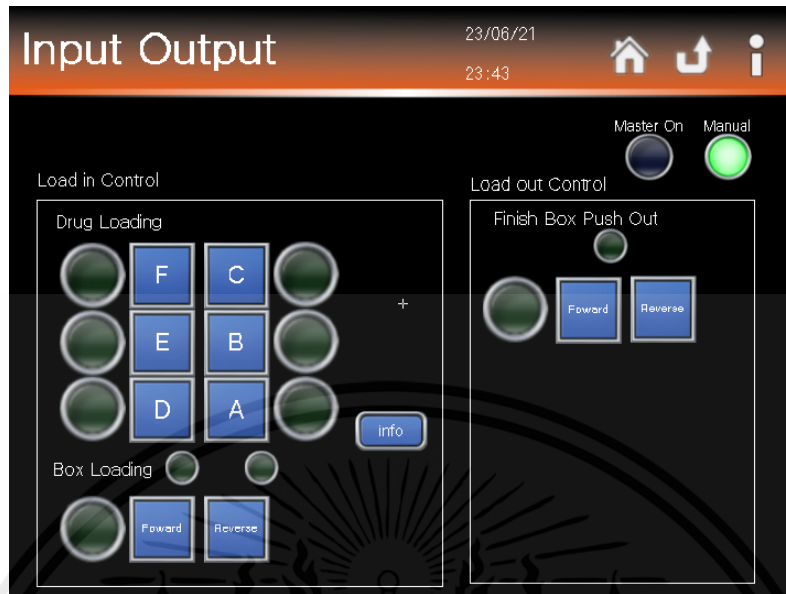
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 80 อังอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.89 หน้า Stock Inventory

จากรูปที่ 3.89 หน้านี้เป็นหน้าที่ใช้ในการกตตั้งค่าจำนวนยาที่มีเพื่อนำไปใช้ในการคำนวณยา
คงเหลือ ทำให้สามารถแจ้งเตือนได้เมื่อมีจำนวนยาไม่พอ

3.8.2 การใช้หน้าจอ GOT ในการทดสอบการทำงานต่าง ๆ



รูปที่ 3.90 หน้า Mechanical Operation

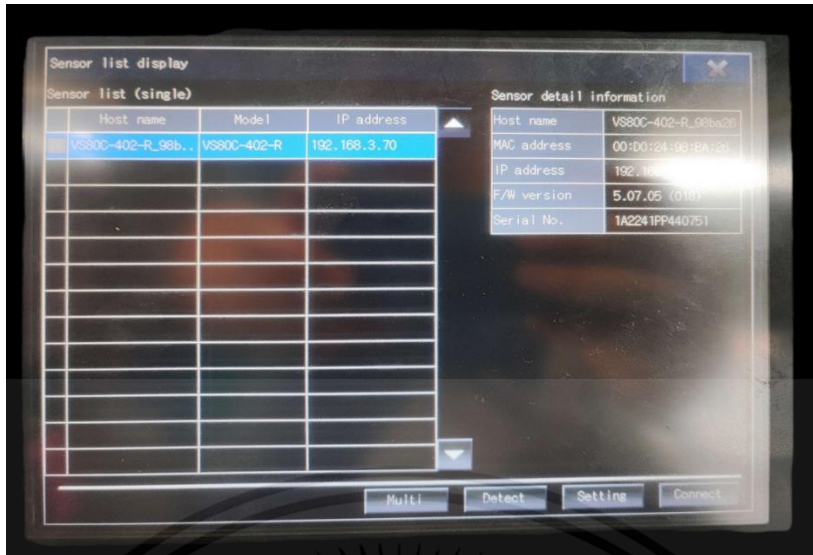
จากรูปที่ 3.90 ในหน้านี้สร้างขึ้นเพื่อการทดสอบการใช้งานส่วนของกลไกต่าง ๆ ของชุดสาธิตว่าสามารถใช้งานได้หรือไม่ โดยมีส่วนควบคุมเอาต์พุตคือโซลินอยด์วาล์วทั้งหมด 7 เป็นแบบจังหวะเดียว (Single Action) และไฟแสดงการทำงานของส่วนเอาต์พุต ส่วนไฟแสดงรีดสวิตช์ของตัวลูกสูบ



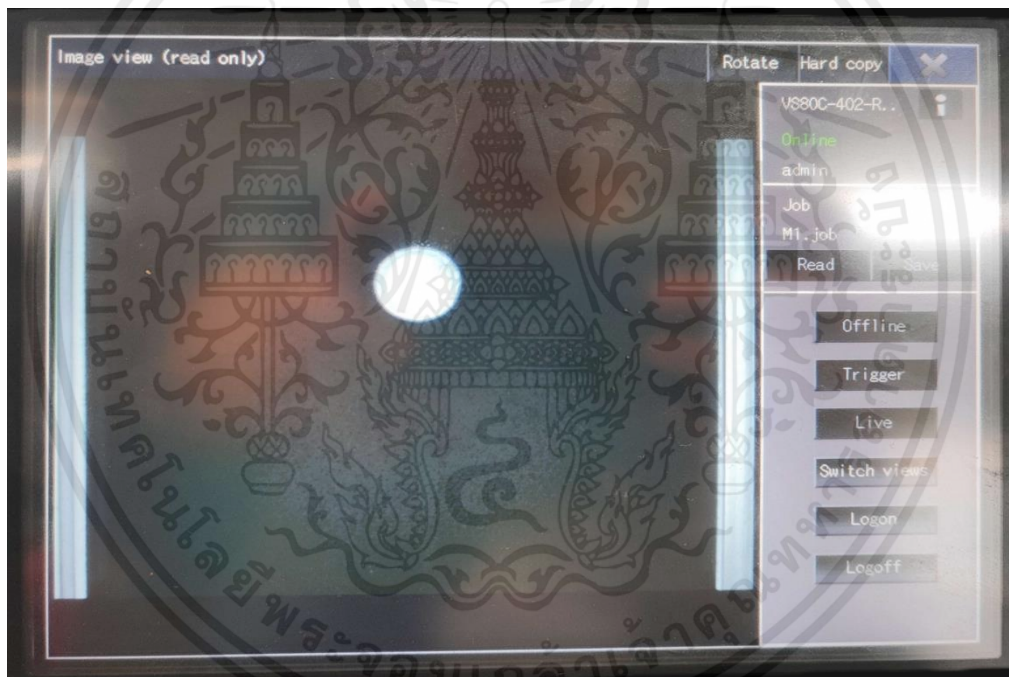
รูปที่ 3.91 หน้า Inverter Operation

จากรูปที่ 3.91 หน้านี้ใช้เพื่อทดสอบการทำงานของสายพานและอินเวอร์เตอร์ โดยมีไฟแสดงสถานะพร้อมทำงาน สามารถกดไปทางซ้ายหรือขวาและกดค้างแบบสวิตช์สลับกดเปิดกดปิดได้ ปรับความเร็วได้ 3 ระดับโดยมีความเร็วพื้นฐานที่เป็นไปตามค่าที่ตั้งความเร็วไว้

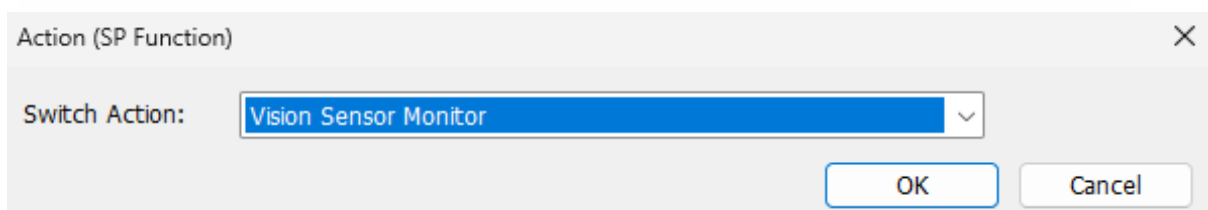
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 82 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.92 หน้า Vision Monitoring ก่อนเชื่อมวิชันเซนเซอร์



รูปที่ 3.93 หน้า Vision Monitoring หลังเชื่อมวิชันเซนเซอร์



รูปที่ 3.94 การตั้งค่าปุ่มเข้า Vision Monitoring

จากรูปที่ 3.92 และ รูปที่ 3.93 เป็นหน้า Vision Monitoring มีหน้าที่สำหรับดูรูปที่วิชันเซนเซอร์
ถ่ายได้ เป็นฟังก์ชันสำเร็จรูปที่มีให้ในโปรแกรม ตามการตั้งค่าในรูปที่ 3.94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและ 83 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.95 หน้า หุ่นยนต์ Operation

จากรูปที่ 3.95 หน้านี้เป็นหน้าที่ไม่ค่อยได้ใช้งานเนื่องจากการทำงานที่สำคัญของหุ่นยนต์ส่วนใหญ่อยู่ในหน้า Main Control อยู่แล้วในหน้านี้มีหน้าที่สำคัญคือสามารถใช้งานการดูแจ้งเตือนของหุ่นยนต์ได้ว่าตอนนี้เป็นแจ้งเตือนแบบไหนและสามารถปรับความเร็วการเคลื่อนที่สูงสุดของหุ่นยนต์ได้

3.8.3 การใช้หน้าจอ GOT ในการระบุข้อผิดพลาดและการแจ้งเตือนของชุดสาริตได้

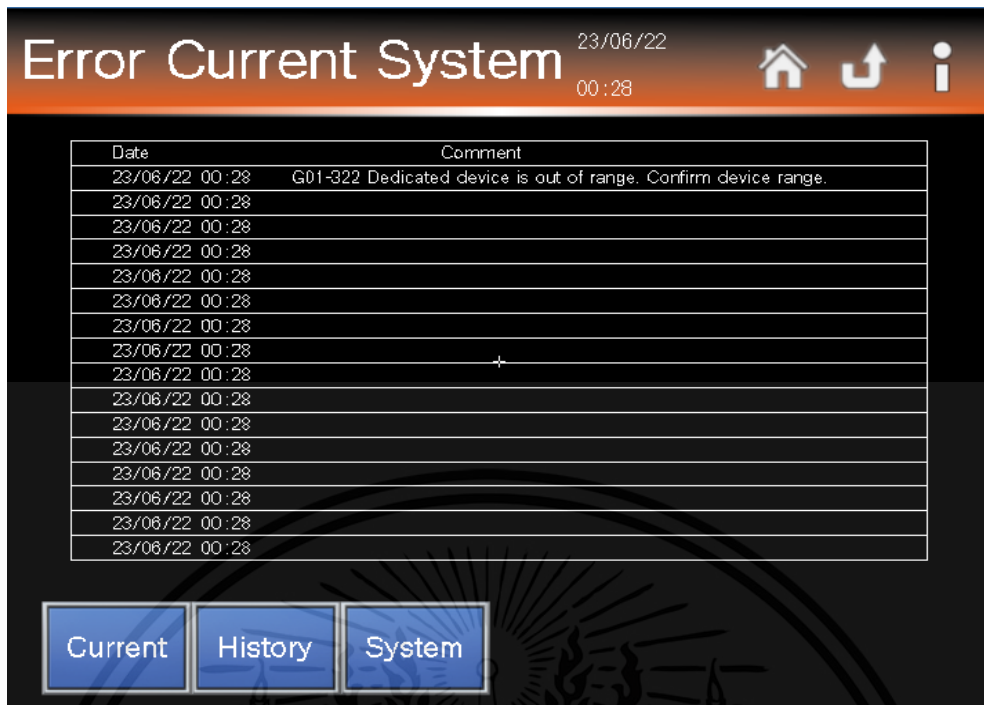


รูปที่ 3.96 หน้า Error Current



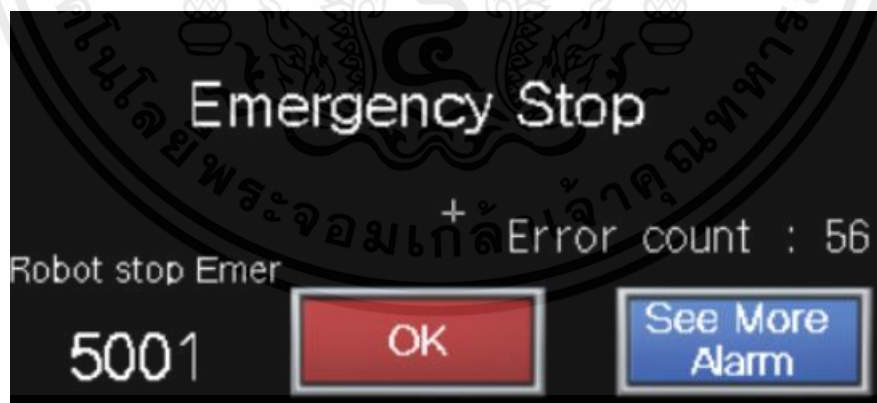
รูปที่ 3.97 หน้า Error History

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 85 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.98 หน้า Error Current System

หน้า Error ทั้งสามแบบในรูปที่ 3.96 -รูปที่ 3.98 มีความแตกต่างกันคือ Error Current แสดงเฉพาะแจ้งเตือนที่เกิดขึ้นอยู่เท่านั้นหากแจ้งเตือนนั้นหายไป ตัวแจ้งเตือนตัวนั้นหายไปจากตารางเอง ส่วน Error History เกิดขึ้นจนกว่าข้อมูลแจ้งเตือนเกินจนโดนเขียนทับหรือกดลบออกไป โดย Error History บอกแจ้งเตือนที่เกิดขึ้นรวมทั้งเวลาในการเกิดและเวลาที่แจ้งเตือนหายไป และ Error Current System บอกแจ้งเตือนที่มาจากระบบ



รูปที่ 3.99 ป๊อปอัพ Alarm

จากรูปที่ 3.99 เมื่อชุดสาริตเกิดการแจ้งเตือนขึ้นในหน้าจอ GOT แสดงป๊อปอัพ Alarm เพื่อบอกสาเหตุของการแจ้งเตือนโดยมีการแจ้งเตือนทั้งหมดดังตารางที่ 3.6 จากตารางที่ 3.6 แบ่งระดับเป็น 3 ระดับ ได้แก่ ระดับ 1 เตือนให้รู้ ระดับ 2 สั่ง Stop หยุดชั่วคราวสามารถสั่งทำงานต่อได้ และระดับ 3 หยุดทันทีหุ่นยนต์ส่งเสียงแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 86 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 การแจ้งเตือนที่แสดงทั้งหมด

อันดับ	ชื่อที่แสดง	รายละเอียด	ระดับ
1	Emergency Stop	เกิดการหยุดฉุกเฉินขึ้นจากข้อผิดพลาดหรือ ผู้ใช้งานกดปุ่มหยุดฉุกเฉิน	3
2	Robot Error	หุ่นยนต์เกิดข้อผิดพลาด	3
3	Robot Not Ready	กด Master On แต่หุ่นยนต์ไม่ยอมตอบ OP Right กลับ(เมื่ออยู่ในโหมดออโต้)	2
4	Robot Link Error	CC-Link ที่ใช้เชื่อมพีแอลซีและหุ่นยนต์ไม่มีการ เชื่อมต่อ	2
5	Inverter Error	อินเวอร์เตอร์เกิดข้อผิดพลาด	2
6	Inverter Link Error	อินเวอร์เตอร์ไม่พร้อมหรือการเชื่อมต่อขาดหาย	2
7	Robot Can not Pick Box Case	หุ่นยนต์หยิบกล่องไม่ได้	2
8	Box Load Clamp	สั่งแล้วกระบอกลูกสูบปล่อยกล่องไม่ยอมเปลี่ยนท่า ตาม	3
9	Box Load Unclamp	สั่งแล้วกระบอกลูกสูบปล่อยกล่องไม่ยอมเปลี่ยนท่า ตาม	3
10	Box Push Clamp	สั่งแล้วกระบอกลูกสูบผลักกล่องออกไม่ยอม เปลี่ยนท่าตาม	3
11	Box Push Unclamp	สั่งแล้วกระบอกลูกสูบผลักกล่องออกไม่ยอม เปลี่ยนท่าตาม	3
12	Right Front Door Open	ประตูฝั่งขวาด้านหน้าเปิด	2
13	Right Back Door Open	ประตูฝั่งซ้ายด้านหน้าเปิด	2
14	Left Front Door Open	ประตูฝั่งขวาด้านหลังเปิด	2
15	Left Back Door Open	ประตูฝั่งซ้ายด้านหลังเปิด	2
16	CF26 Not Ready	กด Start Scan แล้ว CF26 ไม่ตอบกลับ	1
17	Empty Box Part	กล่องหมด	1
18	Not Enough Drug A	มียา A ไม่พอสำหรับทำตามคำสั่งซื้อ	1
19	Not Enough Drug B	มียา B ไม่พอสำหรับทำตามคำสั่งซื้อ	1
20	Not Enough Drug C	มียา C ไม่พอสำหรับทำตามคำสั่งซื้อ	1
21	Not Enough Drug D	มียา D ไม่พอสำหรับทำตามคำสั่งซื้อ	1
22	Not Enough Drug E	มียา E ไม่พอสำหรับทำตามคำสั่งซื้อ	1
23	Not Enough Drug F	มียา F ไม่พอสำหรับทำตามคำสั่งซื้อ	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 87 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

24	Drug A Empty Stock	ยา A หมด	1
25	Drug B Empty Stock	ยา B หมด	1
26	Drug C Empty Stock	ยา C หมด	1
27	Drug D Empty Stock	ยา D หมด	1
28	Drug E Empty Stock	ยา E หมด	1
39	Drug F Empty Stock	ยา F หมด	1
1	Emergency Stop	เกิดการหยุดฉุกเฉินขึ้นจากข้อผิดพลาดหรือ ผู้ใช้งานกดปุ่มหยุดฉุกเฉิน	3
2	Robot Error	หุ่นยนต์เกิดข้อผิดพลาด	3
3	Robot Not Ready	กด Master On แต่หุ่นยนต์ไม่ยอมตอบ OP Right กลับ(เมื่ออยู่ในโหมดออโต้)	2
4	Robot Link Error	CC-Link ที่ใช้เชื่อมพีแอลซีและหุ่นยนต์ไม่มีการ เชื่อมต่อ	2
5	Inverter Error	อินเวอร์เตอร์เกิดข้อผิดพลาด	2
6	Inverter Link Error	อินเวอร์เตอร์ไม่พร้อมหรือการเชื่อมต่อขาดหาย	2
7	Robot Can not Pick Box Case	หุ่นยนต์หยิบกล่องไม่ได้	2
8	Box Load Clamp	สั่งแล้วกระบอกสูบปล่อยกล่องไม่ยอมเปลี่ยนท่า ตาม	3
9	Box Load Unclamp	สั่งแล้วกระบอกสูบปล่อยกล่องไม่ยอมเปลี่ยนท่า ตาม	3
10	Box Push Clamp	สั่งแล้วกระบอกสูบผลักกล่องออกไม่ยอม เปลี่ยนท่าตาม	3

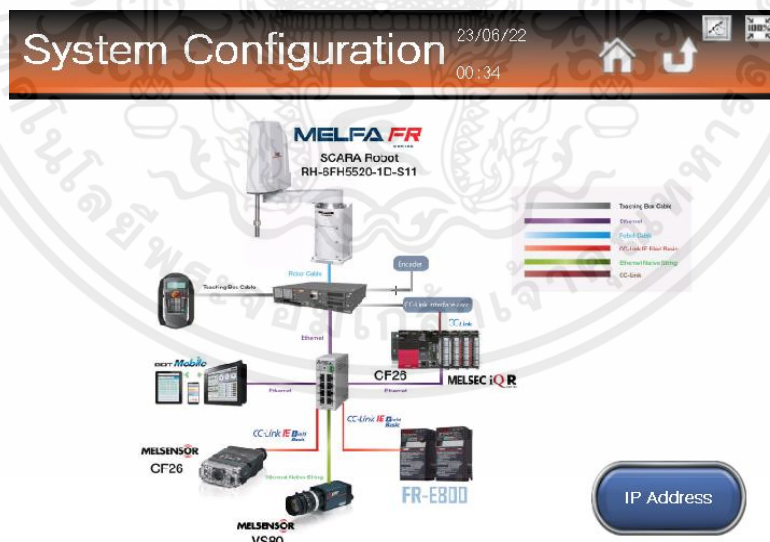
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 88 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.4 การใช้หน้าจอ GOT ในการแสดงข้อมูลเกี่ยวกับตัวเครื่องและแสดงค่า OEE



รูปที่ 3.100 หน้า Dashboard

จากรูปที่ 3.100 เป็นหน้าที่ใช้แสดงค่าต่าง ๆ เช่น เวลาในการเดินเครื่อง เวลาในการหยุดเครื่องแบบไม่อยู่ในแผน เวลาที่ใช้ในหนึ่งรอบการทำงาน และงานที่ขึ้นเป็น OK หรือ NG เพื่อนำมาคำนวณและแสดงผลค่า OEE



รูปที่ 3.101 หน้าโครงสร้างโดยรวมของระบบ

จากรูปที่ 3.101 เป็นหน้าที่ใช้ในการแสดงการเชื่อมต่อทั้งหมดของอุปกรณ์รวมถึงโปรโตคอลที่ใช้ในการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 89 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

System Configuration IP		
		23/06/22
		00:35
Device	IP Adress	Connection
GOT	192.168.3.18	Ethernet
ROBOT	192.168.3.20	Ethernet
VS80	192.168.3.70	Ethernet
PLC	192.168.3.39	Ethernet
CF26	192.168.8.80	CC-link IE Field
Inverter E800 Drug	192.168.3.16	CC-link IE Field
Inverter E800 Base	192.168.3.14	CC-link IE Field

Login Operator Lv1 Manual Control
Password : 1234

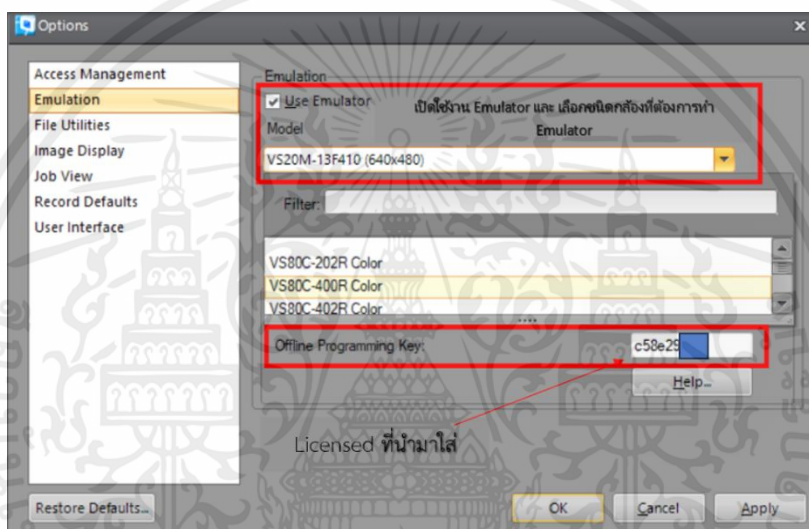
รูปที่ 3.102 หน้าโครงสร้างโดยรวมของระบบ IP

จากรูปที่ 3.102 เป็นหน้าที่ใช้ในการแสดงที่อยู่ IP ของอุปกรณ์ต่าง ๆ

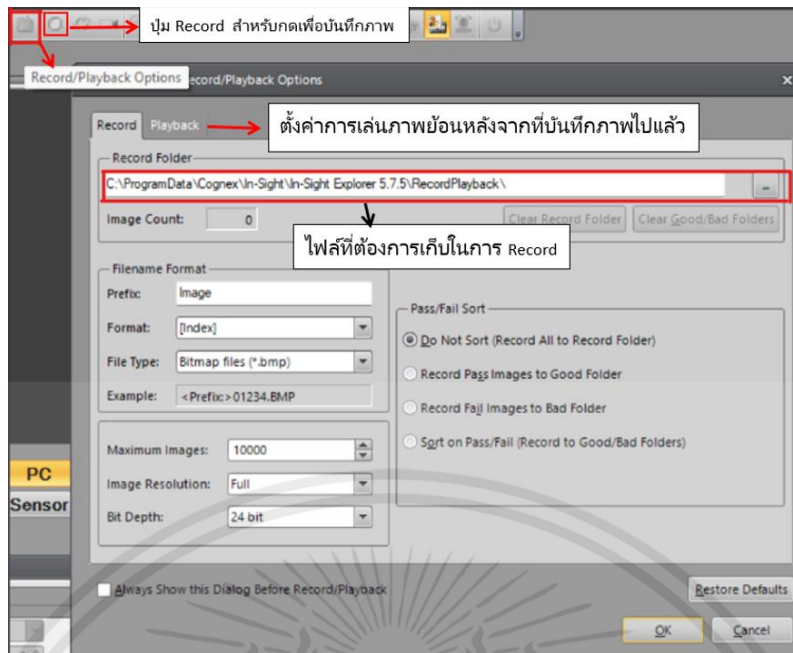
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 90 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 การเขียนโปรแกรมวิชันเซนเซอร์

วิชันเซนเซอร์รุ่น VS80C-402R หรือเรียกโดยย่อว่า VS80 การใช้งานเชื่อมต่อแบบ Ethernet native string ในการสื่อสารกับหุ่นยนต์สการา โดยใช้ซอฟต์แวร์ In-Sight Explorer for MELSENSOR Vision ในการใช้งานจำเป็นต้องลงทะเบียนในเว็บไซต์ของ Cognex นำ licensed Key มาใส่ใน Offline Program Key ก่อดังรูปที่ 3.103 ในการเขียนโปรแกรมและตั้งค่าตัววิชันเซนเซอร์แบบไม่เชื่อมต่อวิชันเซนเซอร์ โดยตัวซอฟต์แวร์สามารถทำการจำลองเชื่อมต่อกับวิชันเซนเซอร์รุ่นต่าง ๆ ได้ด้วยใช้ฟังก์ชัน Emulation ซึ่งส่งผลให้การเขียนโปรแกรมโดยไม่มีอุปกรณ์จริงนั้นสามารถทำได้ด้วยการจำลองตัวคอมพิวเตอร์เป็นวิชันเซนเซอร์รุ่นนั้น ๆ ช่วยเพิ่มความสะดวกในการเขียนโปรแกรม อีกทั้งยังสามารถนำรูปที่เคยถ่ายมาใช้ในการทดสอบโปรแกรมได้อีกด้วยโดยตั้งค่าตามรูปที่ 3.104

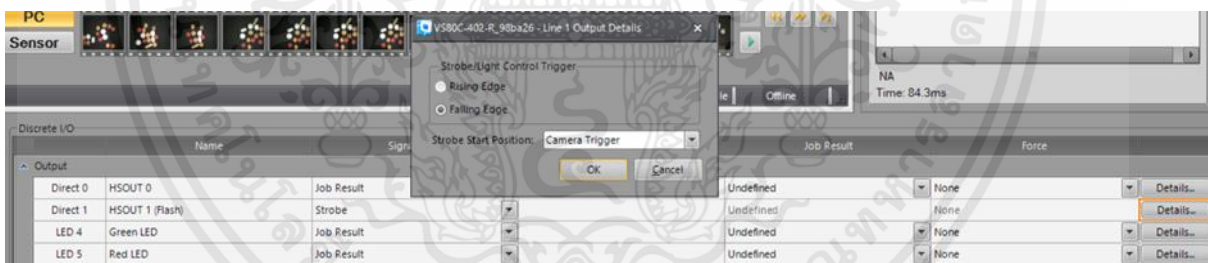


รูปที่ 3.103 การตั้งค่าการตั้งค่าการเปิดใช้งาน Emulator



รูปที่ 3.104 การตั้งค่าการเก็บไฟล์รูปและตั้งไฟล์รูปมาใช้งาน

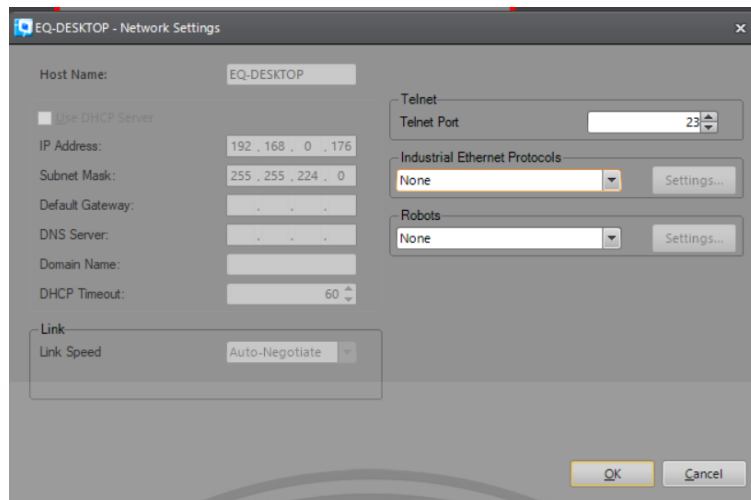
ซอฟต์แวร์สามารถตั้งค่าการสั่ง trigger อุปกรณ์อื่นให้สามารถทำงานได้ที่ฟังก์ชันอินพุต/เอาต์พุต โดยสามารถตั้งค่าวิชันเซนเซอร์ให้ส่งสัญญาณให้อุปกรณ์อื่นซึ่งในที่นี้คือไฟส่องสว่างที่ติดกับตัววิชันเซนเซอร์ การทำงานของวิชันเซนเซอร์ส่งสัญญาณให้แหล่งจ่ายไฟชนิดควบคุมสัญญาณได้ โดยแหล่งจ่ายไฟทำหน้าที่ควบคุมสัญญาณให้ไฟส่องสว่างหน้าวิชันเซนเซอร์ โดยมีการตั้งค่าดังรูปที่ 3.105



รูปที่ 3.105 การตั้งค่าการตั้งค่าการสั่งถ่ายของ HSOUT2 หรือไฟส่องสว่างหน้าวิชันเซนเซอร์

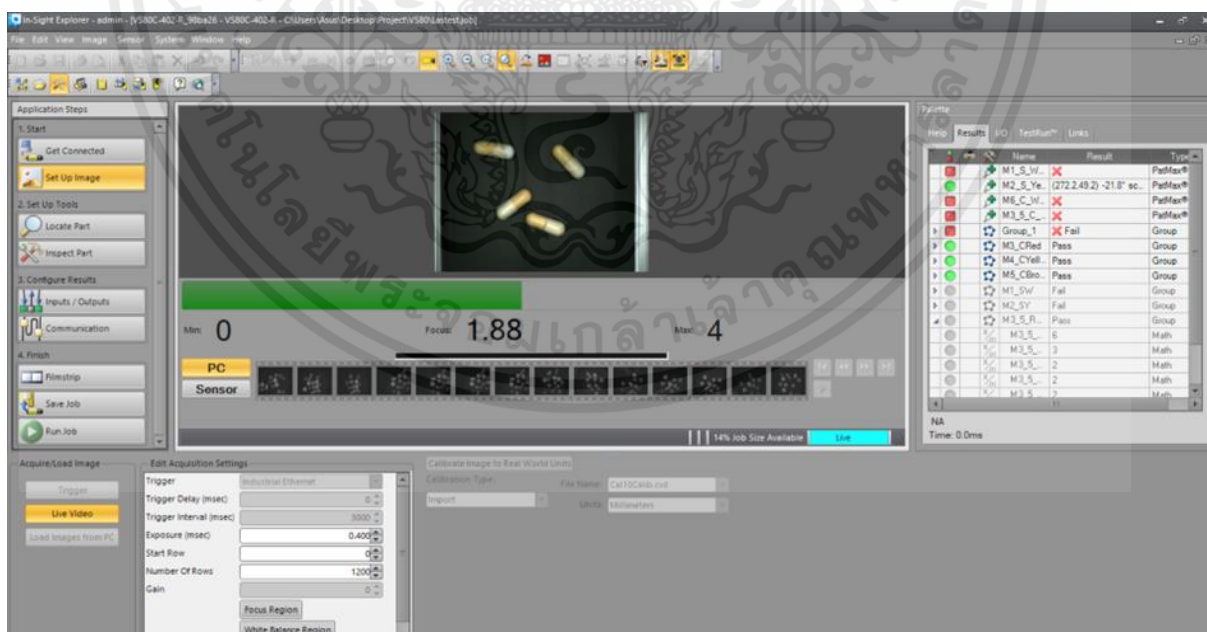
3.9.1 การตั้งค่าก่อนใช้งานโปรแกรม

การตั้งค่าวิชันเซนเซอร์ให้ทำเรียงลำดับไปตามลำดับที่ตัวโปรแกรมเรียงให้โดยเริ่มจากการเชื่อมต่อกับตัววิชันเซนเซอร์โดยจำเป็นต้องปรับที่อยู่ IP ให้ตรงกับคอมพิวเตอร์ก่อน หากใช้ต้องการตั้งค่าโปรแกรมแบบไม่เชื่อมต่อวิชันเซนเซอร์ให้เลือกเชื่อมต่อชื่อคอมพิวเตอร์นั้น ๆ เมื่อเชื่อมต่อวิชันเซนเซอร์ครั้งแรกให้ทำการเปลี่ยนที่อยู่ IP ของวิชันเซนเซอร์ให้อยู่ในวงเดียวกับเครือข่ายของตัวชุดสาธิตก่อน โดยสามารถตั้งค่าได้ที่ Network Settings ดังรูปที่ 3.106



รูปที่ 3.106 ตัวอย่างการตั้งค่าการเชื่อมต่อคอมพิวเตอร์กับวิชันเซนเซอร์

จากนั้นให้ทำการตั้งค่าตัวรูปที่วิชันเซนเซอร์มองเห็นโดยการกดปุ่ม Live Video ดังรูปที่ 3.107 และ ปรับระยะโฟกัสวิชันเซนเซอร์ด้วยการหมุนที่ตัวเลนส์วิชันเซนเซอร์สามารถดูค่าความชัดของรูปได้เมื่อกด Live Video เมื่อรูปชัดให้ปรับค่าตามต้องการได้โดยค่า Exposure เป็นค่าความสว่างของรูปจำเป็นต้องปรับทุกครั้งเมื่อตั้งค่าโปรแกรมใหม่เพื่อให้วิชันเซนเซอร์ยังสามารถจับรูปวัตถุได้ชัดเจน จากนั้นทำ Set White balance เพื่อปรับค่าพื้นฐานของสีขาวเพื่อให้รูปสามารถถ่ายรายละเอียดของวัตถุสีขาวได้ชัดเจน หากทำการปรับเทียบตำแหน่งในโปรแกรม B1 ของหุ่นยนต์สกราราแล้วให้เลือก Calibration type เป็น Import จากนั้นเลือกชื่อไฟล์ที่ทำการตั้งค่าตอนปรับเทียบ

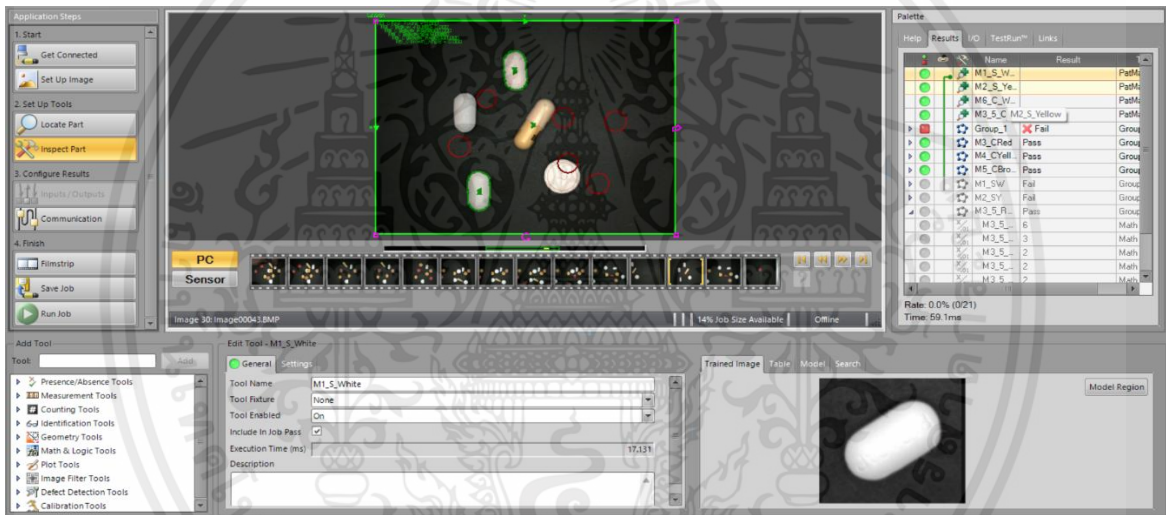


รูปที่ 3.107 การปรับโฟกัสและตั้งค่าการรับรูปเบื้องต้น

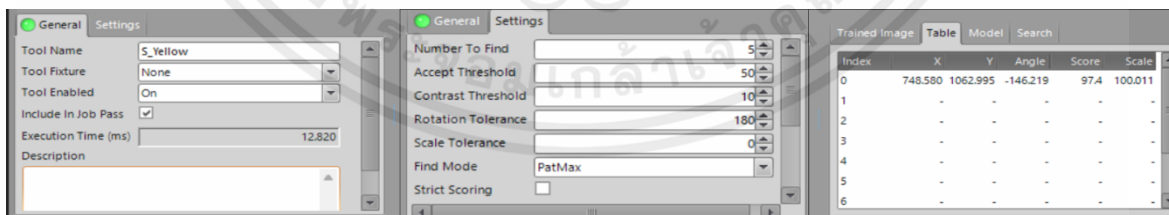
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 93 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9.2 การเขียนโปรแกรมวิชันเซนเซอร์ให้สามารถคัดแยกยาได้

การคัดแยกยาคัดแยกด้วยรูปร่างและสีของตัวยาโดยใช้ฟังก์ชัน PatMax Pattern (1-10) ในการแยกตัวยาคือการแยกเป็นสี่ชนิดเป็น ได้แก่ ชนิดสีเหลืองทรงแท่ง ทรงแท่งสั้นซึ่งมีรูปทรงที่ต่างจากแบบแรก ทรงกระบอกขนาดใหญ่ และทรงกระบอกขนาดเล็ก ฟังก์ชันที่ทำการรับค่าตำแหน่งพิกัด X Y Angle และ Scale ของแต่ละตัวโดยมีวัตถุไม่เกินสิบชิ้น โดยตรวจจับวัตถุจากความเหมือนของรูปที่สอนโดยเทียบกับรูปต้นแบบโดยมีคะแนนความเหมือนของวัตถุให้ยาที่เจอแต่ละชิ้นดังรูปที่ 3.108 โดยสามารถปรับจำนวนที่ต้องการให้พบในการถ่ายหนึ่งครั้งทำรูปเทียบกันและให้คะแนนความเหมือนของรูปเพื่อแบ่งแยกกว่าวัตถุชิ้นนั้นเหมือนกับรูปต้นแบบหรือไม่ โดยคัดแยกผ่านเกณฑ์ค่าการยอมรับหากรูปเหมือนต้นแบบตามเกณฑ์คะแนนที่ตั้งค่าจึงหมายความว่าเจอวัตถุชนิดนั้นนอกจากค่าคะแนนความเหมือนนี้ยังมีค่าความชัดและค่าการหมุนของวัตถุหากเจอวัตถุตัวเดียวกันแต่องศาต่างกันให้ตั้งค่าองศาให้ยอมรับได้ว่าผ่านเกณฑ์การเจอวัตถุชนิดนั้นดังรูปที่ 3.109



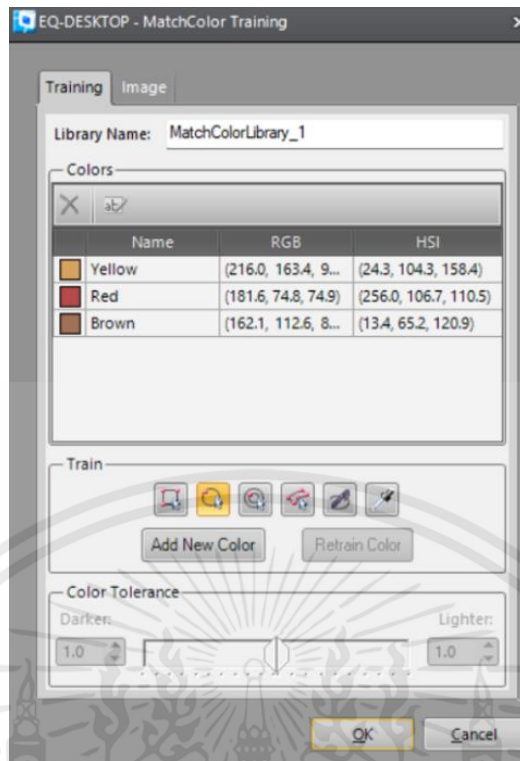
รูปที่ 3.108 การใช้ฟังก์ชัน PatMax Pattern (1-10) ในการแยกตัวยา



รูปที่ 3.109 การตั้งค่าของ PatMax Pattern (1-10) ในการแยกตัวยา

เมื่อสามารถแยกยาได้ 4 ชนิดแล้วชนิดรูปร่างกลมเล็กไปแยกสีดังรูปที่ 3.108 ในส่วนการตั้งค่า Inspect Part ในฟังก์ชันถัดไปใช้ Color ในการแยกสีเนื่องจากมีวัตถุที่ขนาดเท่ากันและรูปทรงเดียวกันอยู่ จึงใช้สีในการระบุเพิ่มเติมในการคัดแยก โดยตั้งค่าให้ทำงานต่อกับฟังก์ชันก่อนที่ Tool Fixture การตั้งค่าการแยกสีทำได้โดยการกด Train color และเลือกช่วงสีที่ต้องการให้ใกล้เคียงกับสีของตัวยาแต่ละชนิดตามรูปที่ 3.110

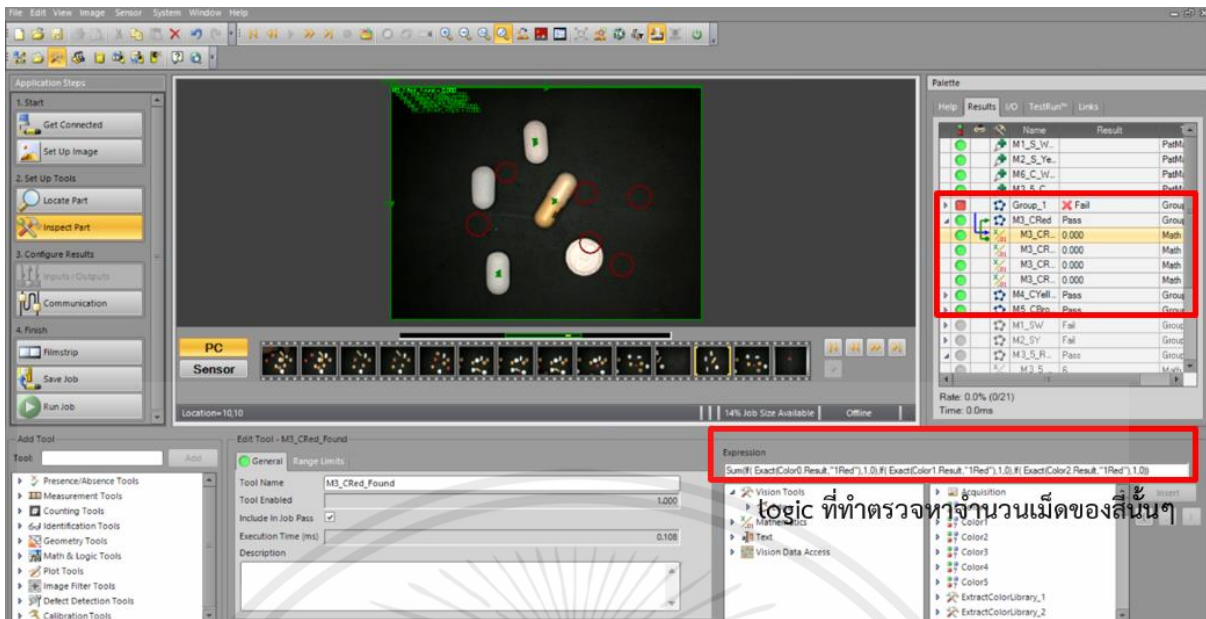
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 94 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



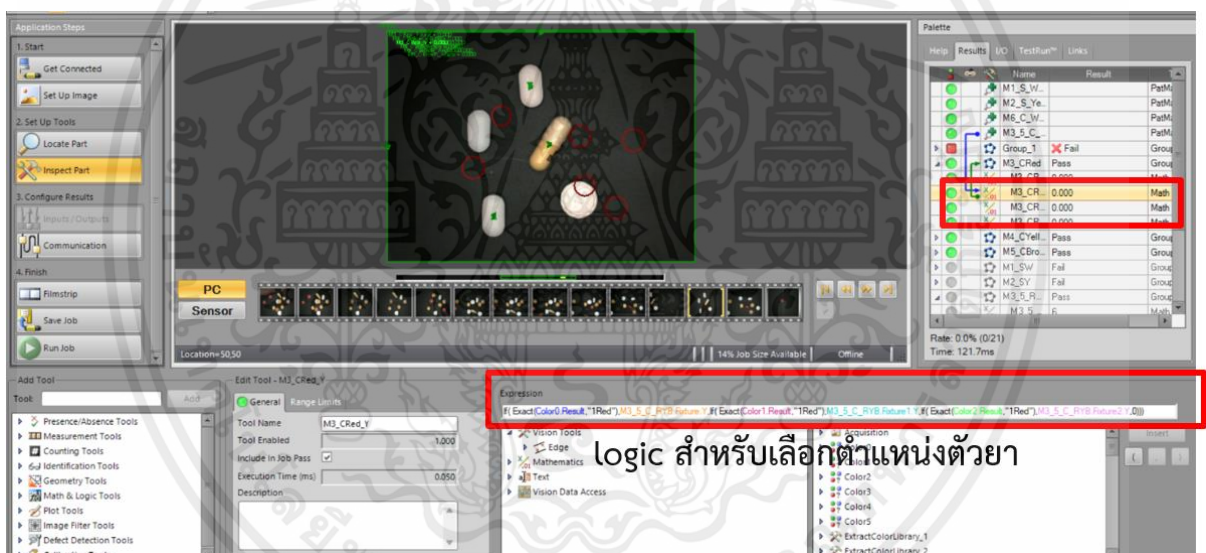
รูปที่ 3.110 การกำหนดสีเพื่อให้วัชชีเซนเซอร์แยกสี

3.9.3 การเขียนโปรแกรมวิชันเซนเซอร์ให้ส่งข้อมูลเข้าหุ่นยนต์ได้

เนื่องจากการส่งข้อมูลนั้นให้เป็นรูปแบบเดียวกันเป็นแพ็คเกจ ได้แก่ จำนวนที่เจอ พิกัดแกน X พิกัดแกน Y และมุมองศา Angle วนไปจนครบทุกชนิดชนิดละเม็ด ข้อมูลเม็ดยาที่เป็นสีปัจจุบันอยู่ในฟังก์ชัน PatMax Pattern (1-10) ซึ่งบอกแค่จำนวนยาที่เจอทั้งหมดซึ่งจำนวนยาจริงทั้งหมดที่จำเป็นต้องคัดแยกสีทั้งสามสีนั้นถูกรวมกันในฟังก์ชันเดียวจึงต้องทำการใช้ฟังก์ชัน logic เขียนเพื่อหาจำนวนยาแต่ละสีว่าแต่ละสีพบเป็นจำนวนกี่ชิ้นและชิ้นสีนั้น ๆ อยู่พิกัดใดอีกครั้งหนึ่งดังรูปที่ 3.111 และรูปที่ 3.112



รูปที่ 3.111 เงื่อนไขการคัดแยกข้อมูลจำนวนยาในเม็ดยาที่แยกสี



รูปที่ 3.112 เงื่อนไขการคัดแยกข้อมูลตำแหน่งยาในเม็ดยาที่แยกสี

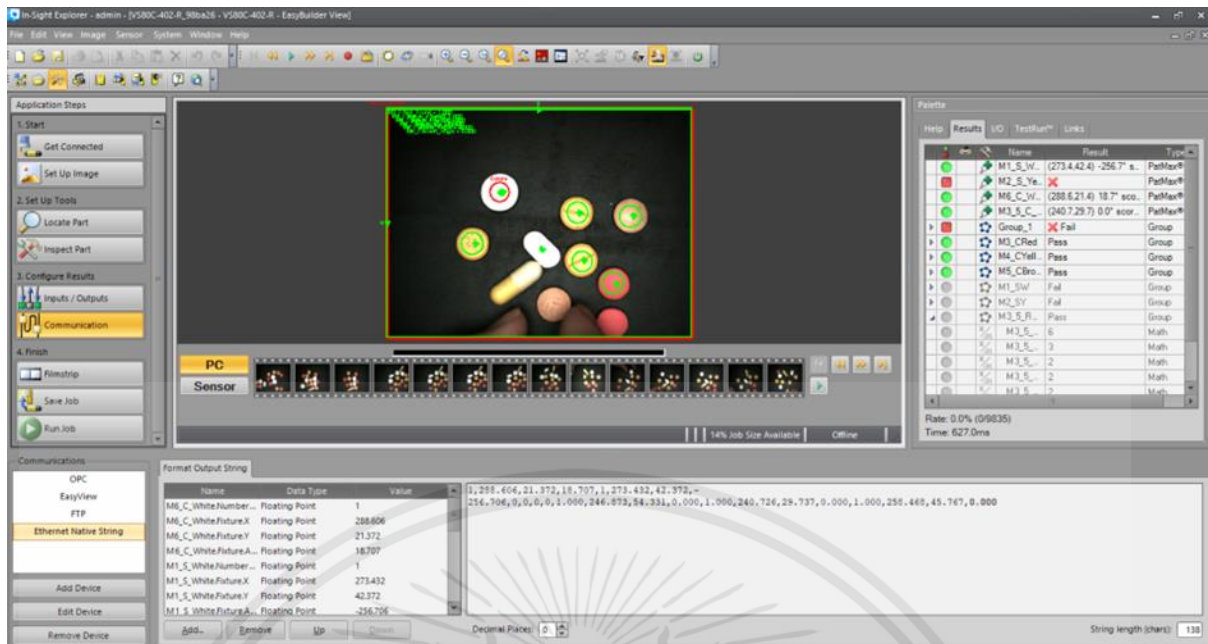
จากนั้นไปที่ส่วนของ Communication เลือกเป็น Ethernet Native String ในการส่งข้อมูลให้หุ่นยนต์สกรานั้นให้เลือก Add และเลือกข้อมูลที่ต้องการส่งไปว่าเป็นข้อมูลใดบ้างโดยข้อมูลที่ส่งไปให้หุ่นยนต์สกรานั้นเป็นดังรูปที่ 3.113

กลุ่มขาว CW	แก๊งขาว SW				แก๊งเหลือง SY				กลุ่มเล็กน้ำตาล CB				กลุ่มเล็กแดง CR				กลุ่มเล็กเหลือง CY						
NumberF	x	y	angle	NumberF	x	y	angle	NumberF	x	y	angle	NumberF	x	y	angle	NumberF	x	y	angle	NumberF	x	y	angle
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

รูปที่ 3.113 String ที่ส่งไปยังหุ่นยนต์

จากนั้นให้ Go online เพื่อให้หุ่นยนต์สกราทดสอบการรับข้อมูลเป็นการถัดไปและยังสามารถดูข้อมูลที่ส่งให้หุ่นยนต์สกรานั้นได้แบบเรียลไทม์ดังรูปที่ 3.114

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 96 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



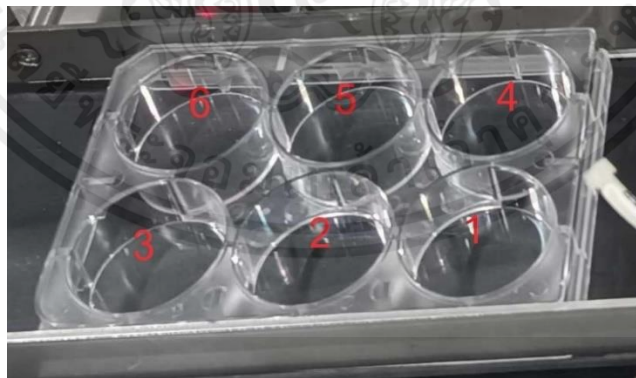
รูปที่ 3.114 ตัวอย่างข้อมูลที่ส่งไปยังตัวหุ่นยนต์

3.10 การเขียนโปรแกรมควบคุมหุ่นยนต์

ในการเขียนโปรแกรมควบคุมหุ่นยนต์มีการแบ่งส่วนการทำงานแล้วเขียนตามขั้นตอนดังนี้

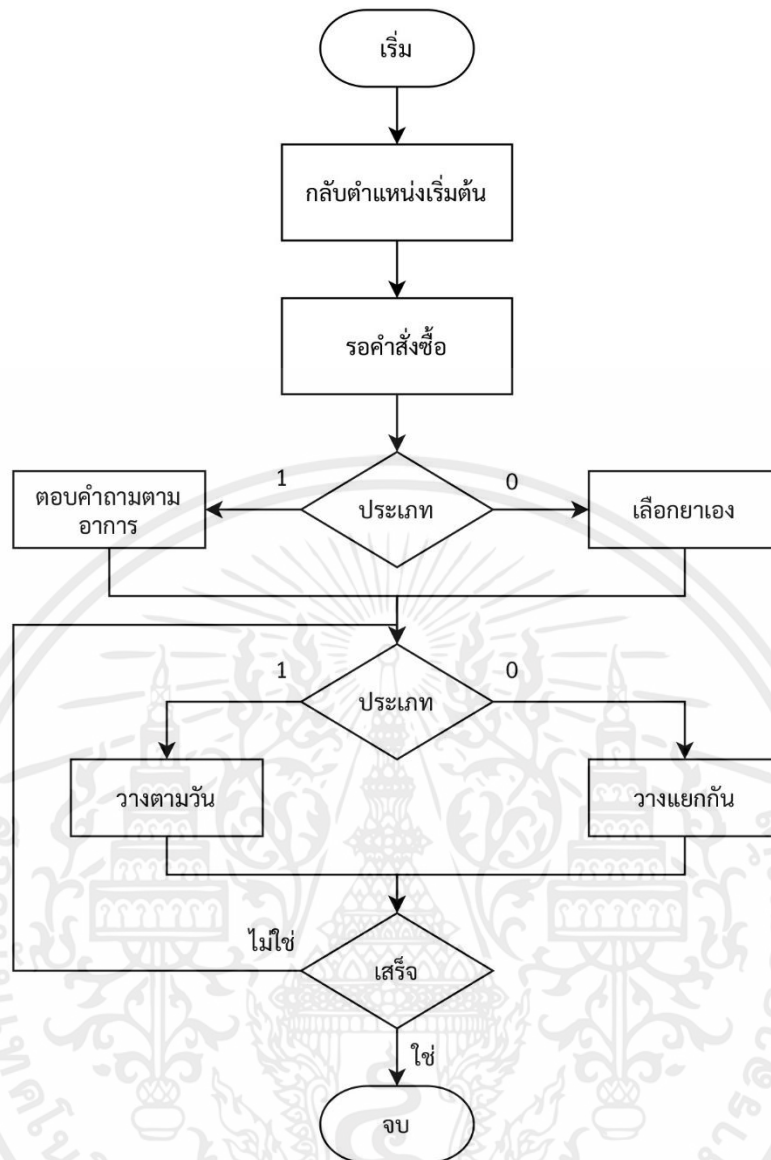
3.10.1 การเขียนโปรแกรมหยิบยาใส่กล่อง

ในช่วงการใช้เพียงการจำลองบนซอฟต์แวร์ได้เริ่มเขียนโปรแกรมให้ทำการรับค่าคำสั่งซื้อจากพีแอลซีและจัดการจุดของการวางยาลงกล่อง โดยให้หยิบยาจากตำแหน่งเดิมไปวางที่ตำแหน่งที่เรียงกัน 6 ช่องดังรูปที่ 3.115 มีการวางยา 2 รูปแบบคือจัดยาตามวันและวางแยกชนิดกันมีการทำงานดังรูปที่ 3.116 โดยให้ตัวแปรที่พีแอลซีส่งมาให้เป็นบิต 1 เป็นจัดยาตามวันและ 0 เป็นแบบวางแยกชนิดกัน



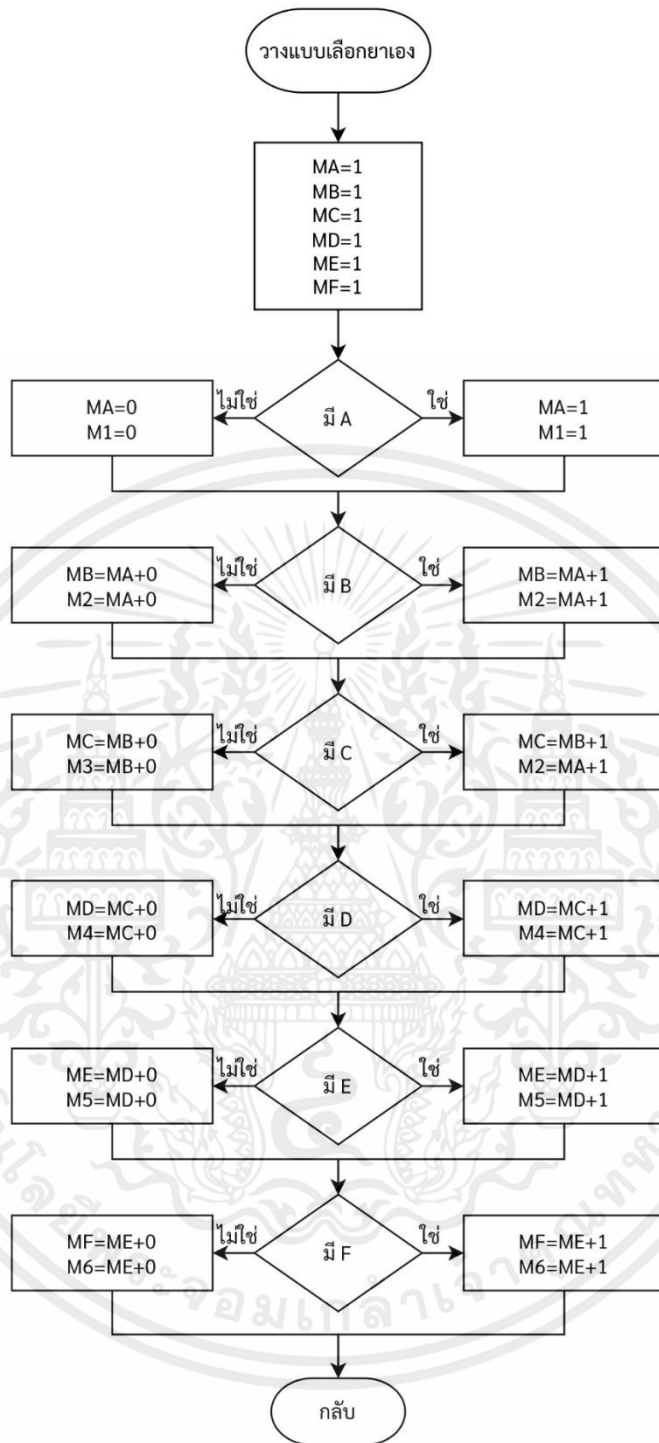
รูปที่ 3.115 ตำแหน่งช่องสำหรับวางยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 97 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



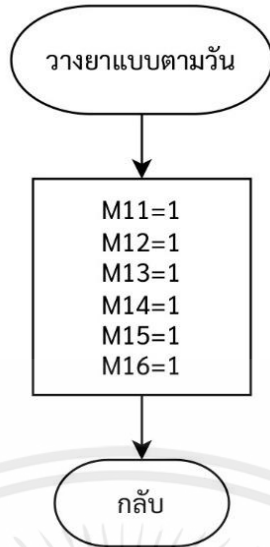
รูปที่ 3.116 ผังงานการทำงานของโปรแกรมหยิบยาใส่กล่อง

ในการจัดการตำแหน่งวางยาแบบแยกกันนั้นมีการทำงานดังรูปที่ 3.117 เริ่มวางยาที่มีในคำสั่งซื้อจากช่องที่ 1 เป็นต้นไปหากไม่มียา A ทำการวางยา B ในช่องที่ 1 แทนหรือก็คือหากมีคำสั่งซื้อยา A ตำแหน่งเริ่มจะเป็น 1 หากไม่มีจะเป็น 0 ยาชนิดต่อไปตั้งแต่ B จนถึง F หากมีคำสั่งซื้อตำแหน่งเริ่มจะเป็นตำแหน่งก่อนหน้าบวก 1 หากไม่มีจะเป็นตำแหน่งก่อนหน้าบวก 0 มีตัวแปรต่อไปนี้ MA MB MC MD ME และ MF คือจำนวนช่องที่ใช้ไปแล้วและ M1 ถึง M6 คือจำนวนช่องที่กำหนดให้เป็นช่องวางยาเรียงตามลำดับของยา A ถึง F



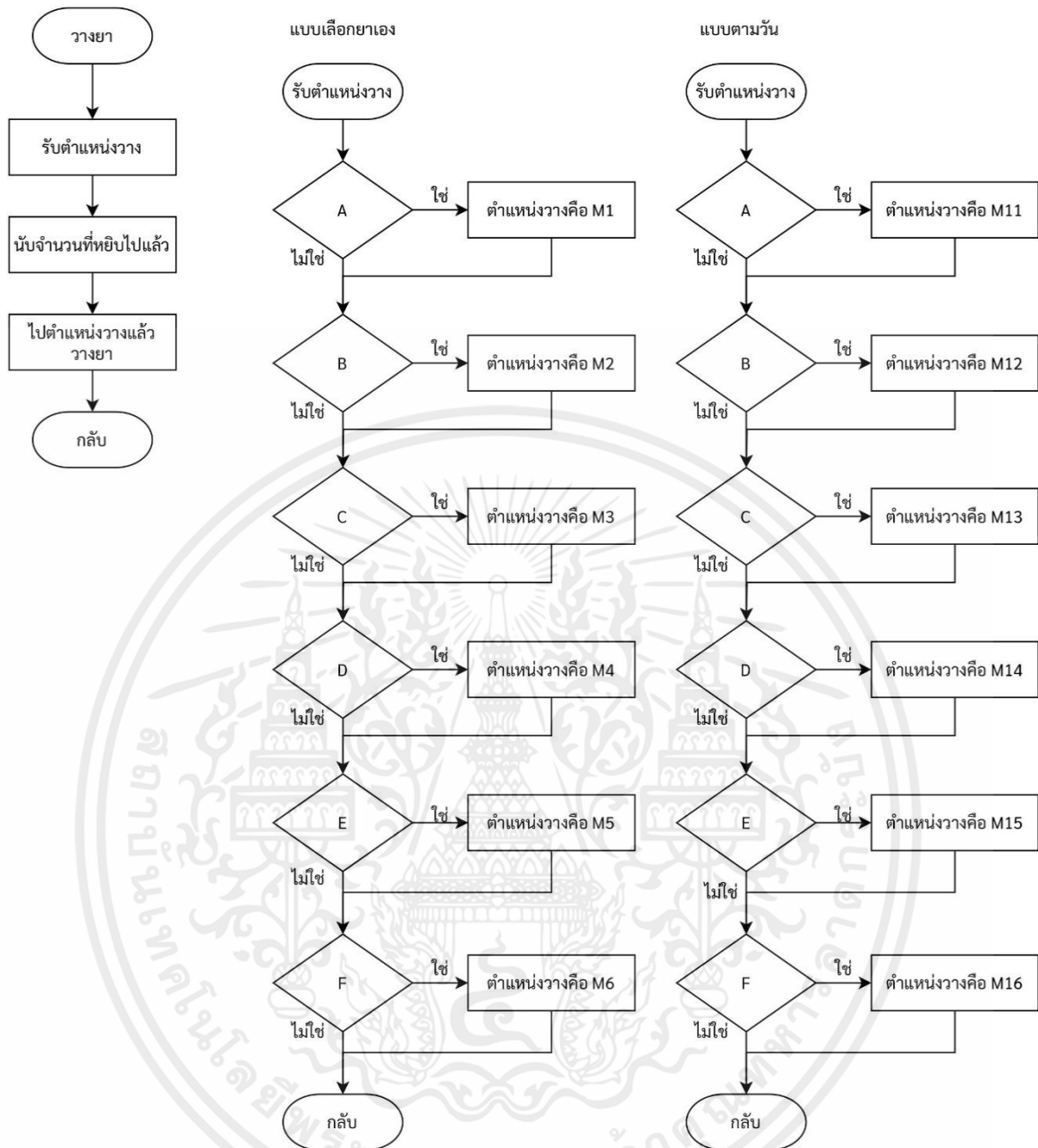
รูปที่ 3.117 ผังงานการทำงานโปรแกรมจัดที่วางแบบจัดยาเอง

ในส่วนของการวางยาตามแบบรายวันแบ่งเป็นมือเริ่มวางจากช่องที่ 1 เสมอเนื่องจากยาที่ต้องทาน ถูกแบ่งออกเป็น 6 มือ ให้ตัวแปร M11-M16 เป็นตัวแปรของตำแหน่งตั้งแต่ยา A จนถึงยา F ตามลำดับมี การทำงานดังรูปที่ 3.118



รูปที่ 3.118 ผังงานการทำงานโปรแกรมจัดที่วางแบบแบ่งตามมือ

ในส่วนของโปรแกรมในการวางยาทั้งแบบวางแยกและวางแบบแบ่งตามมือนั้นต่างกันที่ตัวแปรในการนำมาใช้เป็นตำแหน่งของการวาง ในการวางแบบแบ่งตามมือนั้นใช้ตัวแปร M11-M16 แต่ในการแบ่งแบบเลือกยาเองเป็นการใช้ตัวแปร M1-M6 มีการทำงานดังรูปที่ 3.119 รับค่าตำแหน่งวางแล้วทำการนับจำนวนยาที่หยิบแล้วค่อยไปยังตำแหน่งวาง ในการวางยาแบบแยกยาดำแหน่งของยาแต่ละชนิดไม่เปลี่ยน แต่ในการวางแบบแบ่งตามมื่อยาถูกเปลี่ยนตำแหน่งวางลงช่องตามมือ

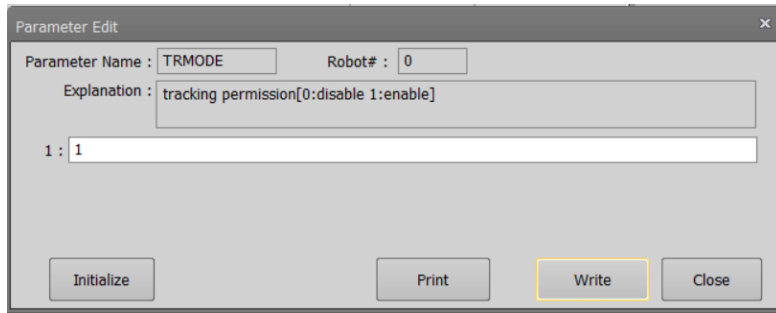


รูปที่ 3.119 ฟังงานการทำงานของโปรแกรมวางยา

3.10.2 การเขียนโปรแกรมวิชันแทรกคิง

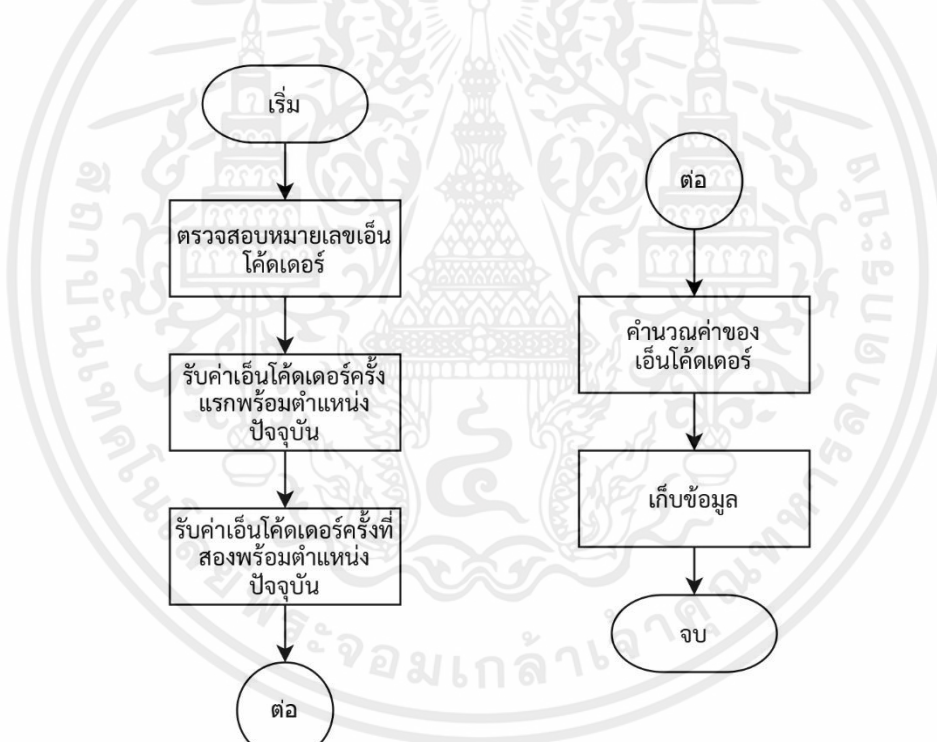
การเขียนโปรแกรมวิชันแทรกคิงต้องทำการตั้งค่าพารามิเตอร์และสอบเทียบ (Calibration) ตำแหน่งสายพานและหุ่นยนต์ ตามด้วยการทำสอบเทียบตำแหน่งวิชันเซนเซอร์และหุ่นยนต์ และการสอบเทียบตำแหน่งของการหยิบชิ้นงานตามลำดับ แล้วจึงเริ่มทำการเขียนโปรแกรมวิชันแทรกคิง

เริ่มจากการตั้งค่าพารามิเตอร์ของหุ่นยนต์ให้ค้นหาชื่อพารามิเตอร์ที่ชื่อว่า TRMODE เมื่อคลิกเข้าไปแล้วพบกับหน้าต่างดังรูปที่ 3.120 เป็นหน้าต่างการตั้งค่าของพารามิเตอร์นี้ให้ทำการใส่ 1 ลงไปเพื่อเปิดใช้งานฟังก์ชันแทรกคิง



รูปที่ 3.120 หน้าตั้งค่าพารามิเตอร์ฟังก์ชันแทรกคิง

หลังจากนั้นทำการสอบเทียบตำแหน่งสายพานและหุ่นยนต์เพื่อทำให้หุ่นยนต์รู้ตำแหน่งของสายพานที่ทำการแทรกคิงตัวเมื่อยาด้วยการคำนวณการเคลื่อนที่ของหุ่นยนต์ต่อจำนวนค่าของพัลส์เปลี่ยนแปลงไปของเอ็นโค้ดเดอร์โดยใช้โปรแกรม A1 ที่เป็นโปรแกรมสำหรับการสอบเทียบสายพานกับหุ่นยนต์มีการทำงานของโปรแกรมตามผังงานดังรูปที่ 3.121 รับค่าพิกัดของหุ่นยนต์และค่าของเอ็นโค้ดเดอร์ 2 ตำแหน่งที่ต่างกัน



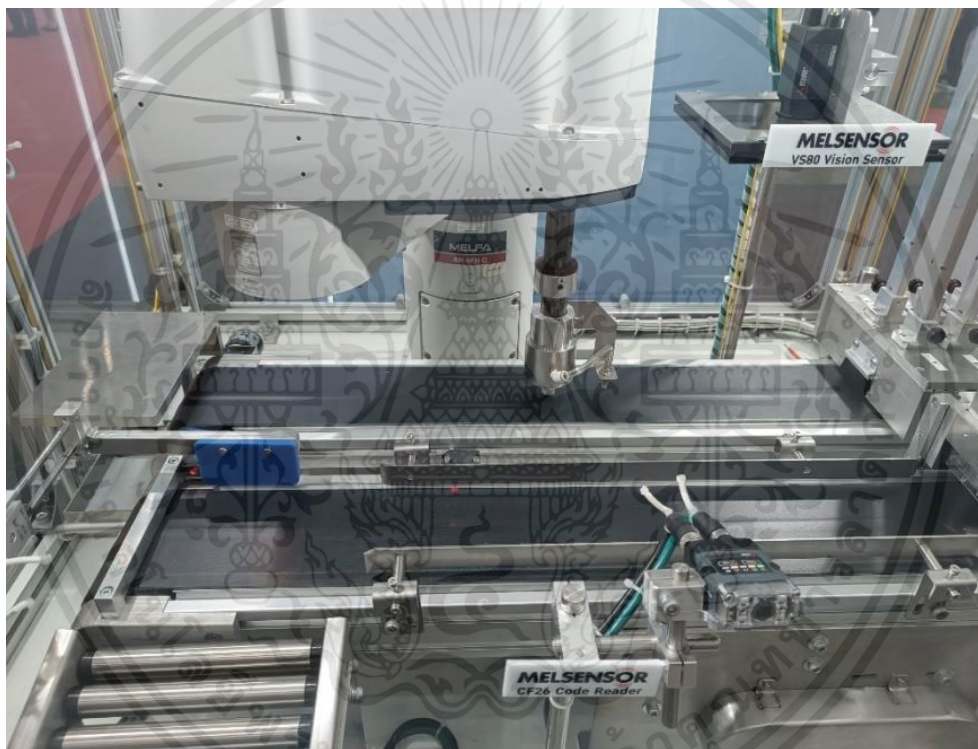
รูปที่ 3.121 ผังงานการทำงานของโปรแกรม A1

ในข้อมูลตำแหน่งในแกน X แกน Y และแกน Z สามารถถูกนำมาใช้เป็นตัวแปรในโปรแกรมได้ดังตัวอย่างในรูปที่ 3.122 คือ PE.X ถูกใช้เป็นตัวแปรสำหรับหมายเลขของเอ็นโค้ดเดอร์

Position											
<input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>											
Name	X	Y	Z	A	B	C	L1	L2	FLG1	FLG2	
PE	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0	...	0
PX10PS1	550.000	0.000	300.000	0.000	0.000	0.000	0.000	0.000	4	...	0
PX10PS2	550.000	0.000	300.000	0.000	0.000	0.000	0.000	0.000	4	...	0
PY10ENC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0	...	0

รูปที่ 3.122 การเปลี่ยนพิกัดของโปรแกรม A1

จากนั้นทำการกำหนดจุดขึ้นมาแล้วให้หุ่นยนต์ไปที่จุดที่ต้องการเริ่มทำการแทรกคิงตัวเม็ดยาดังรูปที่ 3.123 แล้วทำการเริ่มโปรแกรมหุ่นยนต์สการาให้ทำงานไปถึงบรรทัดที่ 17 ของโปรแกรม A1 ดังรูปที่ 3.124



รูปที่ 3.123 ตำแหน่งแรกของโปรแกรม A1

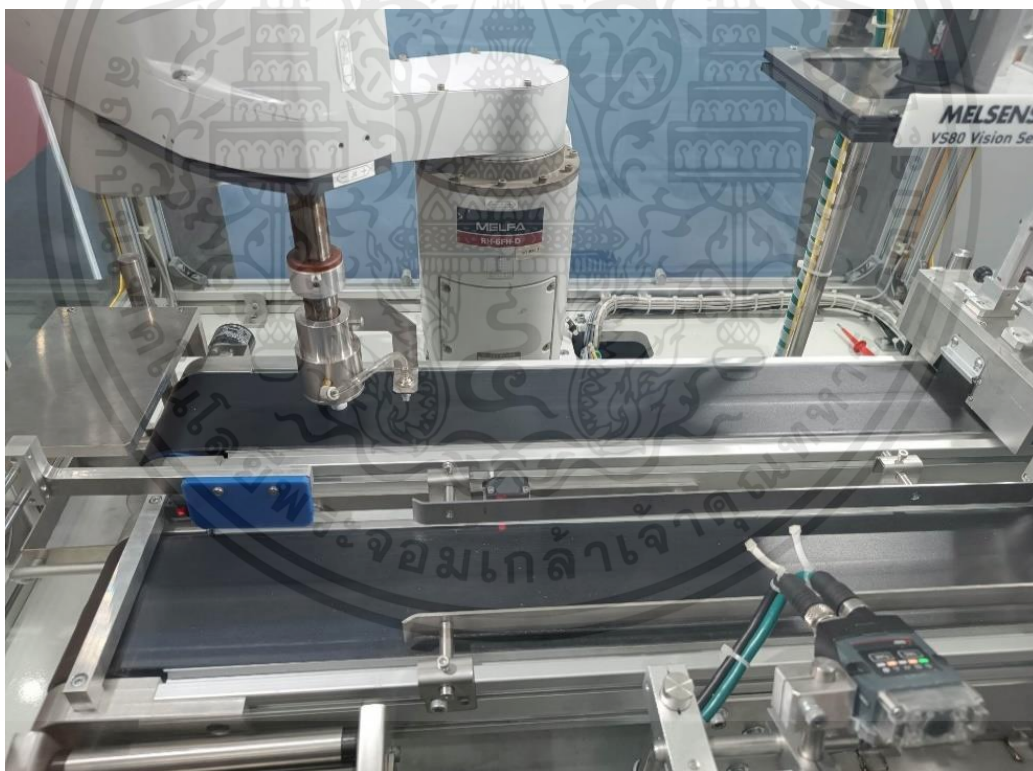
```

Status:  Program:  Display Format : 
9 MECMAX=8 'The maximum encoder number value (for checking)
10 If PE.X<1 Or PE.X>MECMAX Then Error 9101 'Encoder number out of range
11 MENCNO=PE.X 'Acquire the encoder number
12 '(2) Attach a marking sticker on the conveyer upstream side/
13 '(3) Move the robot to the position right at the center of the attached sticker/
14 MX10EC1#=M_Enc(MENCNO) 'Acquire encoder data (first time)
15 PX10PS1=P_Zero 'Set all elements to ZERO
16 PX10PS1=P_Fbc(1) 'Acquire the current position (first time)
17 Hit
18 '(4) Raise the robot/
19 '(5) Move the sticker in the forward direction of the conveyer/
20 '(6) Move the robot to the position right at the center of the moved sticker/
21 MX10EC2#=M_Enc(MENCNO) 'Acquire encoder data (second time)
22 PX10PS2=P_Zero 'Set all elements to ZERO
23 PX10PS2=P_Fbc(1) 'Acquire the current position (second time)
24 '(7) Raise the robot/

```

รูปที่ 3.124 โปรแกรม A1 หยุดที่บรรทัดที่ 17

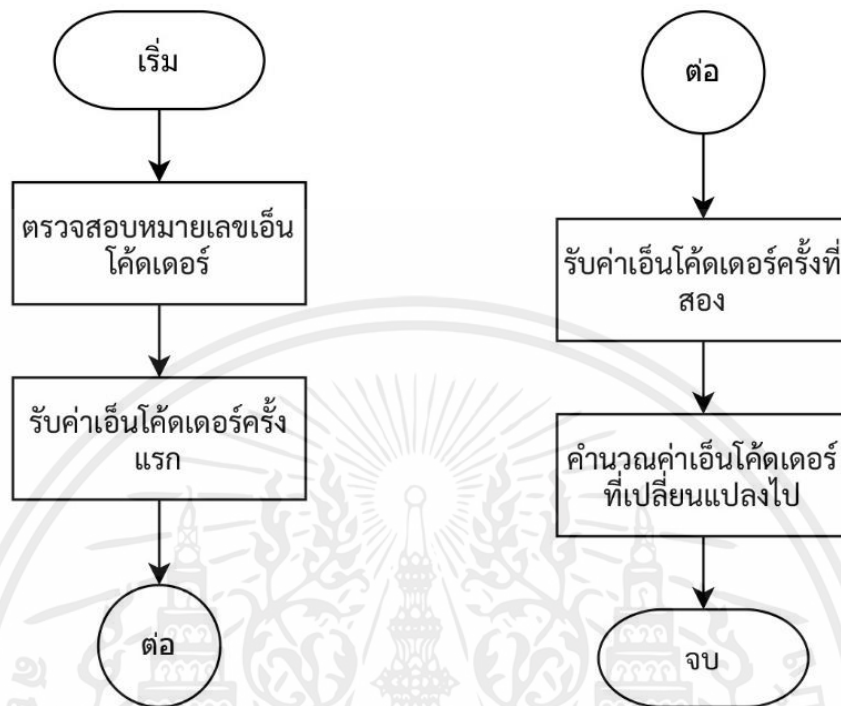
จากนั้นทำการเลื่อนสายพานให้จุดที่เลือกไว้ย้ายตำแหน่งไปยังตำแหน่งที่กำหนดไว้ดังรูปที่ 3.125
เมื่อย้ายจุดแล้วทำการย้ายตำแหน่งหุ่นยนต์ไปยังจุดนั้นแล้วทำการสั่งให้โปรแกรมทำงานต่อจนจบ



รูปที่ 3.125 ตำแหน่งที่ 2 ของโปรแกรม A1

หลังจากสอบเทียบตำแหน่งสายพานกับหุ่นยนต์เสร็จแล้วทำการสอบเทียบวิชันเซนเซอร์และหุ่นยนต์ต่อ โดยการสอบเทียบตำแหน่งวิชันเซนเซอร์และหุ่นยนต์เป็นการทำให้หุ่นยนต์รู้ตำแหน่งของชิ้นงานที่วิชันเซนเซอร์ถ่ายได้เป็นพิกัดแกน X และแกน Y ด้วยโปรแกรม B1 โปรแกรมนี้รับค่าเอ็นโค้ดเดอร์มา 2 ค่า คือ จุดที่วิชันเซนเซอร์ถ่ายและอีกจุดคือจุดที่หุ่นยนต์สามารถเคลื่อนที่ไปตำแหน่งบนเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 104 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระดาษที่ใช้สอบเทียบได้ ทำการนำสองค่านี้มีค่านวนหาตำแหน่งของวิชันเซนเซอร์ มีการทำงานของโปรแกรมตามผังงานดังรูปที่ 3.126



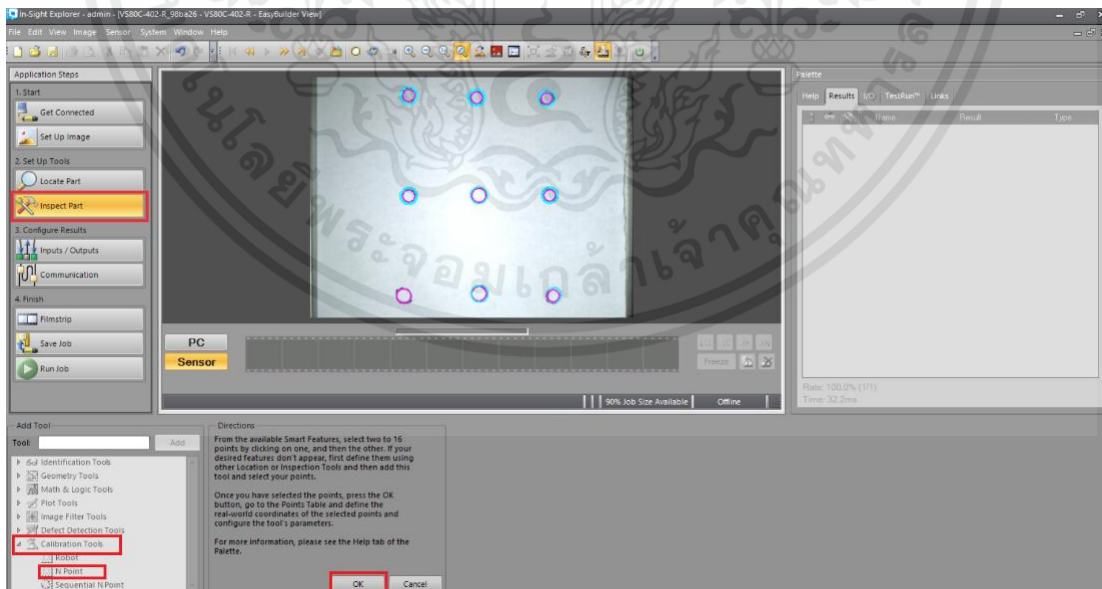
รูปที่ 3.126 ผังงานการทำงานของโปรแกรม B1

เปลี่ยนค่าของตำแหน่งในพิกัดแกน X และแกน Y ของตำแหน่ง PRM1 เป็น 1 ซึ่งเป็นหมายเลขรุ่น และ หมายเลขของเอ็นโค้ดเดอร์ ทำการสร้างรูปที่มีวงกลมตั้งแต่ 5 จุดขึ้นไปแล้วนำไปวางในตำแหน่งที่ต้องการให้วิชันเซนเซอร์ถ่ายเพื่อจับตำแหน่งด้วยาดังรูปที่ 3.127



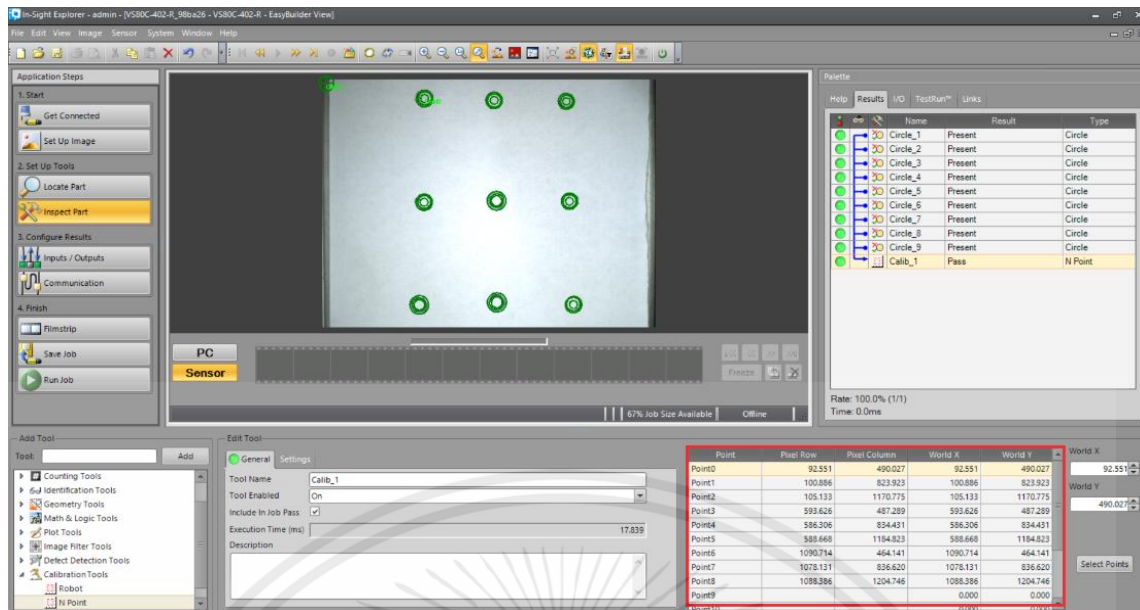
รูปที่ 3.127 การวางรูปเพื่อทำการสอบเทียบ

เมื่อวางรูปให้อยู่ในตำแหน่งถ่ายแล้วไปที่ซอฟต์แวร์ In-Sight Explorer for MELSENSOR เชื่อมต่อกับวิชั่นเซนเซอร์ให้เรียบร้อย เมื่อเชื่อมต่อแล้วตัวซอฟต์แวร์จะแสดงรูปล่าสุดที่วิชั่นเซนเซอร์ถ่ายไว้ดังรูปที่ 3.128 ทำการเลือกที่ Inspect Part แล้วไปที่ Calibration Tools แล้วเลือกฟังก์ชัน N Point มีวงกลมที่พบขึ้นมาให้คลิกวงกลมทั้งหมดที่พบแล้วกด OK ได้ตำแหน่งเป็น Pixel มาให้ดังรูปที่ 3.129



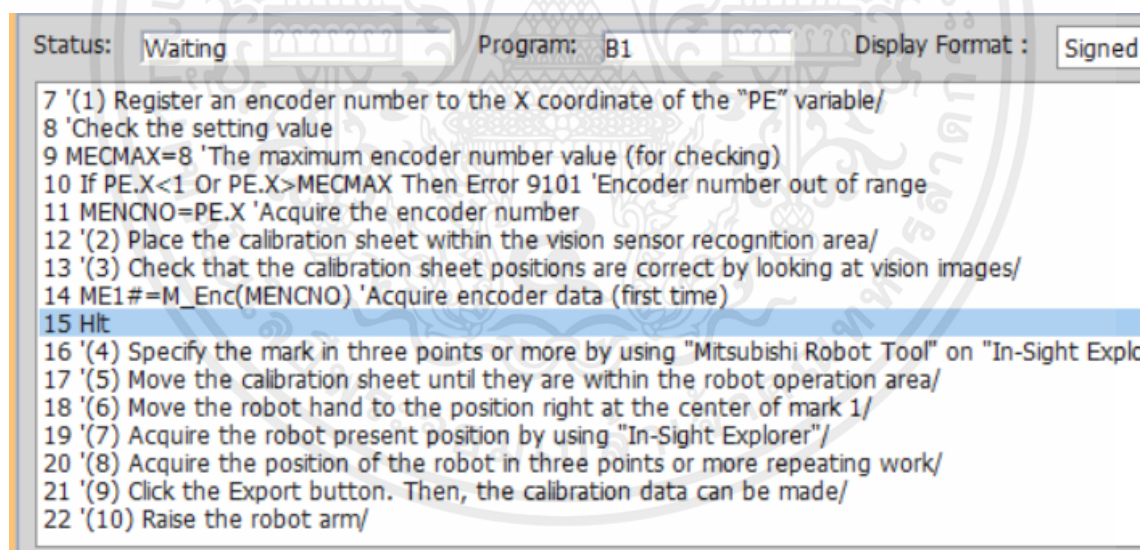
รูปที่ 3.128 การเลือกตำแหน่งที่ทำการสอบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 106 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



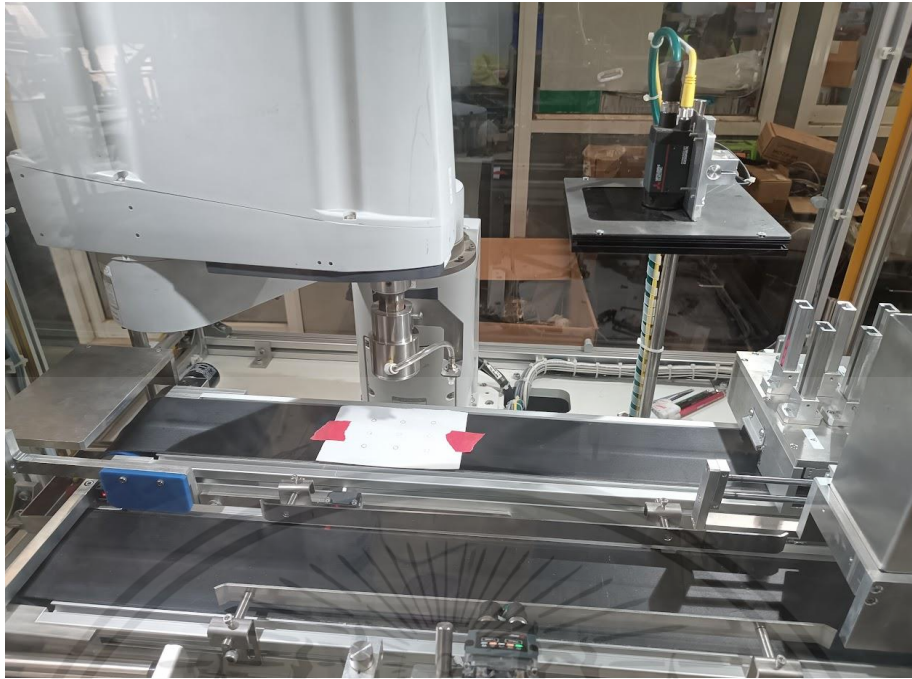
รูปที่ 3.129 จุดที่ใส่พิกัดที่ต้องใส่เพื่อทำการสอบเทียบ

หลังจากนั้นให้กดโปรแกรมหุ่นยนต์ทำงานจนถึงบรรทัดที่ 15 ดังรูปที่ 3.130 แล้วทำการเลื่อนสายพานย้ายตำแหน่งของกระดาษไปยังตำแหน่งที่ห่างจากวิชชันเซนเซอร์แล้วหุ่นยนต์สามารถขยับไปยังตำแหน่งตามบนกระดาษได้ตามรูปที่ 3.131



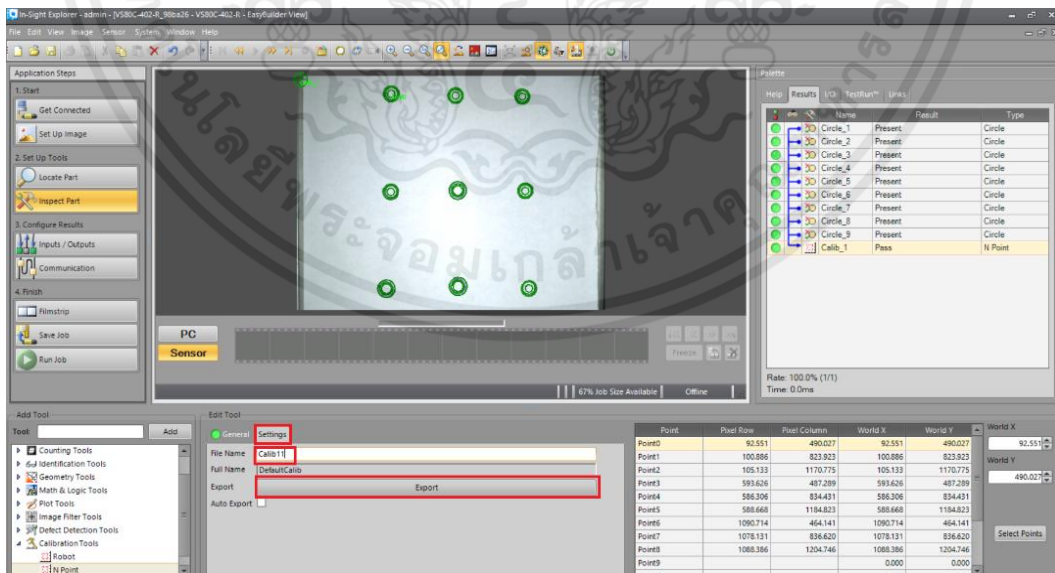
รูปที่ 3.130 โปรแกรม B1 หุ่นที่บรรทัดที่ 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 107 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.131 ตำแหน่งที่หุ่นยนต์สามารถหยิบได้โดยไม่ชนวิชันเซนเซอร์

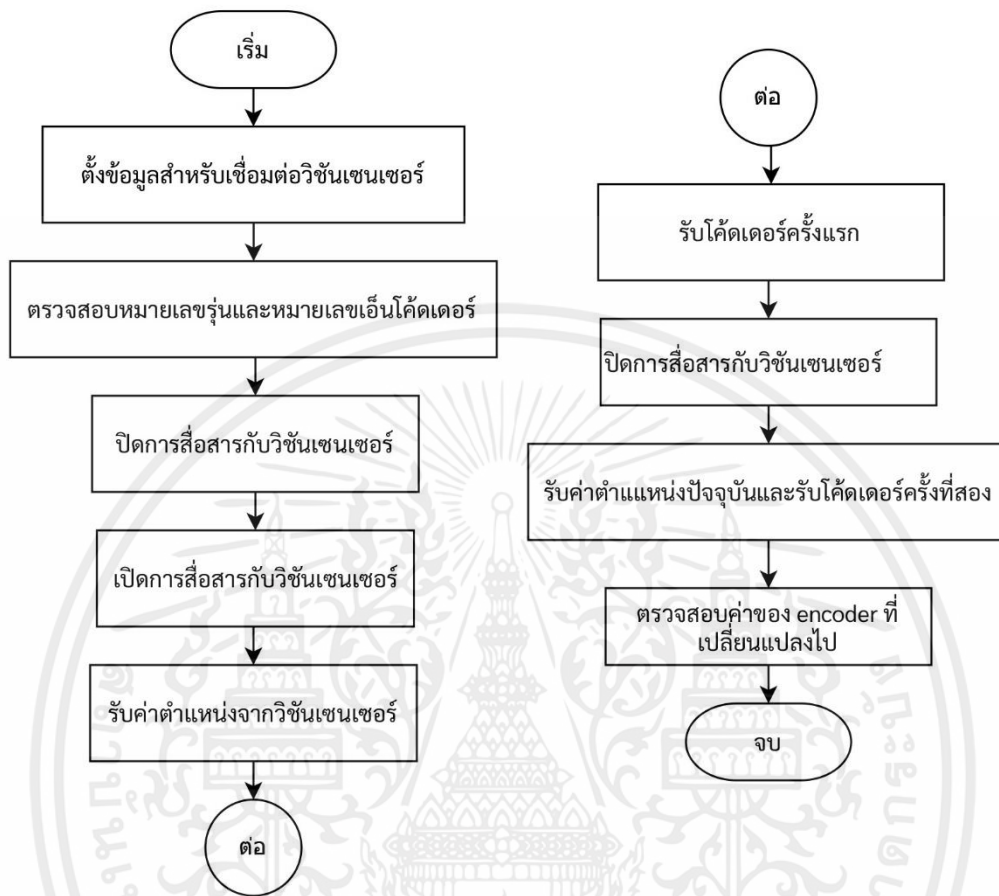
ทำการใช้หุ่นยนต์ขยับวงกลมแต่ละวงจากนั้นนำตำแหน่งวงกลมที่เป็นพิกัดแกน X และแกน Y ของหุ่นยนต์มาใส่ตามลำดับของวงกลมในซอฟต์แวร์ In-Sight Explorer for MELSENSOR ทำซ้ำจนครบทุกจุดเมื่อใส่ครบทุกจุดแล้วไปที่ Setting ที่ทำการเปลี่ยนชื่อเป็นชื่อที่ใช้แล้วทำการกด Export ตามรูปที่ 3.132 โปรแกรมทำการคำนวณพิกัดต่าง ๆ ออกมาเพื่อไปเทียบกับสายพานจริงให้ผลออกมาเป็นไฟล์เทียบพิกัดระหว่างสายพานและพิกัดที่วิชันเซนเซอร์พบวัตถุ



รูปที่ 3.132 การบันทึกข้อมูลสอบเทียบเพื่อนำไปใช้

เมื่อมีข้อมูลของการระบุตำแหน่งวิชันเซนเซอร์แล้วให้ทำการสอบเทียบตำแหน่งของชิ้นงานต่อเพื่อระบุตำแหน่งของชิ้นงานที่หยิบด้วยโปรแกรม C1.Prg โปรแกรมสอบเทียบนี้เริ่มจากการอ่านค่าที่เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 108 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

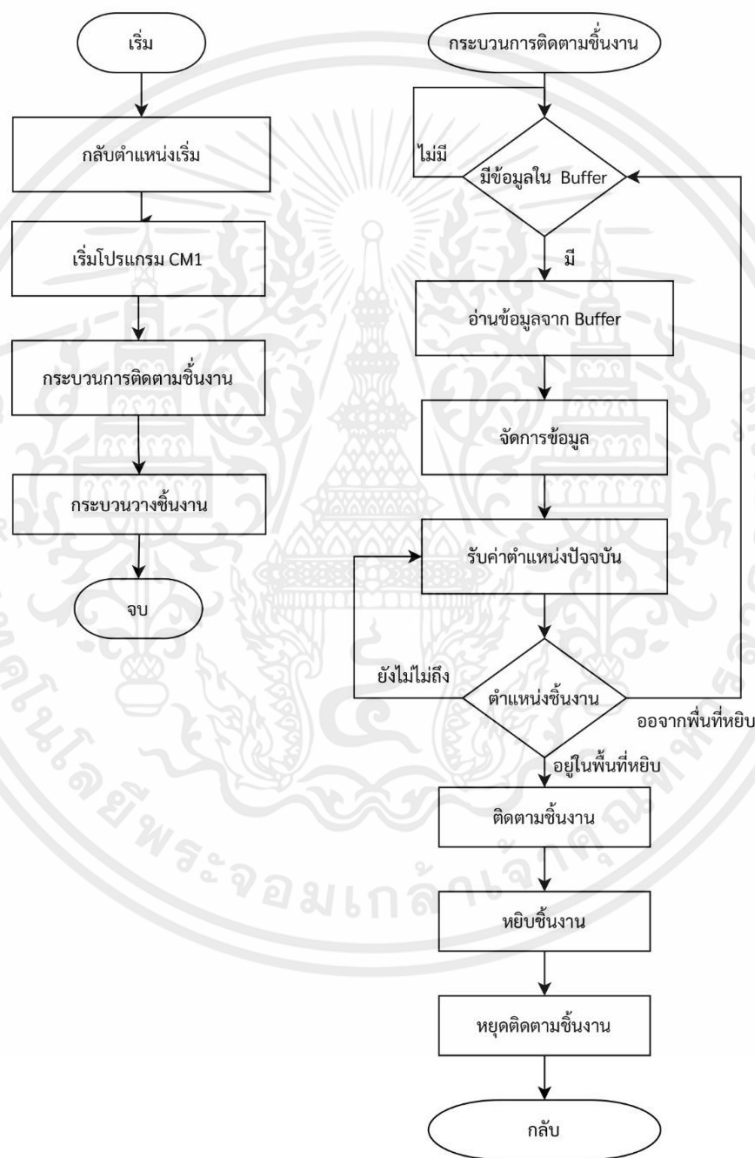
วิชันเซนเซอร์ส่งมาแล้วนำไปคำนวณหาจุดหยาบชิ้นงานจากค่าเอ็นโค้ดเดอร์ตำแหน่งถ่ายและตำแหน่งหยาบ มีการทำงานตามผังงานดังรูปที่ 3.133



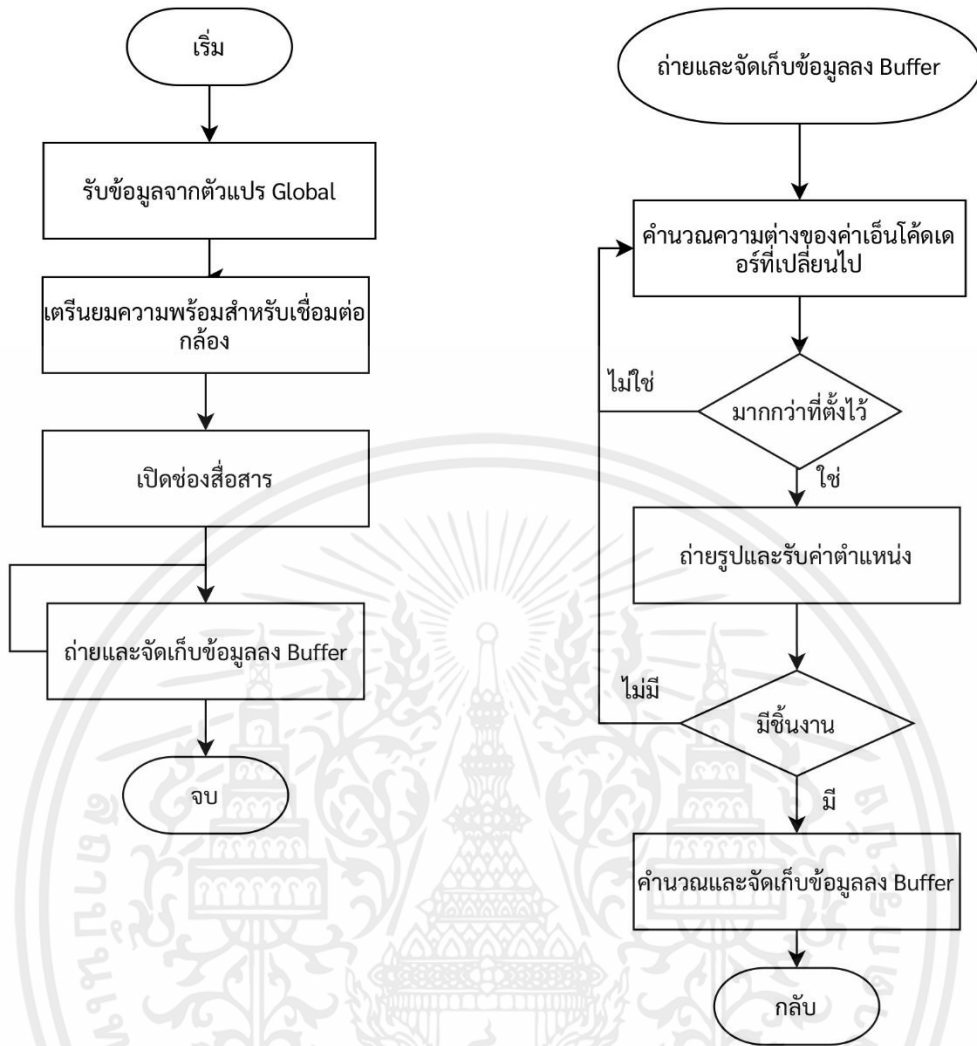
รูปที่ 3.133 ผังงานการทำงานของโปรแกรม C1

เปลี่ยนค่าของตำแหน่งในพิกัดแกน X และแกน Y ของตำแหน่ง PRM1 เป็น 1 เหมือนกับโปรแกรม B1 ต่อมาใน In-Sight Explorer for MELSENSOR โดยการไปที่ Set Up Image ไปที่ Calibration Type เลือกเป็น Import ที่ File Name เลือกเป็นชื่อไฟล์ที่เคย Export ตอนโปรแกรม B1 หลังจากนั้นนำยามาวางไว้ทำการถ่ายรูปยา A ที่เป็นวงกลมและมีขนาดใหญ่ง่ายต่อการสอบเทียบ แล้วออนไลน์วิชันเซนเซอร์ที่ปุมดังรูปที่ 3.134

ในโปรแกรมตัวอย่างจากคู่มือโปรแกรมวิชันแทคกิงมีการใช้โปรแกรม 2 โปรแกรมทำงานไปพร้อมกันให้โปรแกรม 1 ซึ่งเป็นโปรแกรมที่ใช้ในการติดตามชิ้นงานบนสายพานและโปรแกรม CM1 เป็นการทำงานสั่งการวิชันเซนเซอร์ให้ถ่ายรูปแล้วนำข้อมูลของตำแหน่งไปเก็บยัง Buffer การทำงานของโปรแกรม 1 ดังรูปที่ 3.136 มีขั้นตอนเริ่มจากกลับตำแหน่งเริ่มต้นแล้วสั่งให้โปรแกรม CM1 ทำงานแล้วรอข้อมูลจาก Buffer เพื่อทำการติดตามชิ้นงานบนสายพาน และโปรแกรม CM1 รูปที่ 3.137 มีการทำงานคือคอยสั่งให้วิชันเซนเซอร์ทำการถ่ายรูปเมื่อค่าของเอ็นโค้ดเดอร์มีการเปลี่ยนแปลงไปจนถึงค่าที่กำหนดไว้ เมื่อเจอชิ้นงานจะทำการคำนวณตำแหน่งแล้วจัดเก็บลงไปยัง Buffer



รูปที่ 3.136 ผังงานการทำงานของโปรแกรม 1



รูปที่ 3.137 ผังงานการทำงานของโปรแกรม CM1

ในโปรแกรม 1 ต้องเปลี่ยนค่าของตัวแปรตำแหน่งดังนี้เพื่อเริ่มใช้งานดังต่อไปนี้
 PUP1 ทั้ง 3 แกน คือความสูงของตำแหน่งหยิบชิ้นงาน ให้เป็น 50 ทั้ง 3 แกน
 PUP2 ทั้ง 3 แกน คือความสูงของตำแหน่งวางชิ้นงาน ให้เป็น 50 ทั้ง 3 แกน
 PTN แกน X คือรูปแบบของตำแหน่งสายพานให้เป็น 2 เป็นรูปแบบที่มีสายพานอยู่ด้านหน้า
 เคลื่อนที่จากซ้ายไปขวา

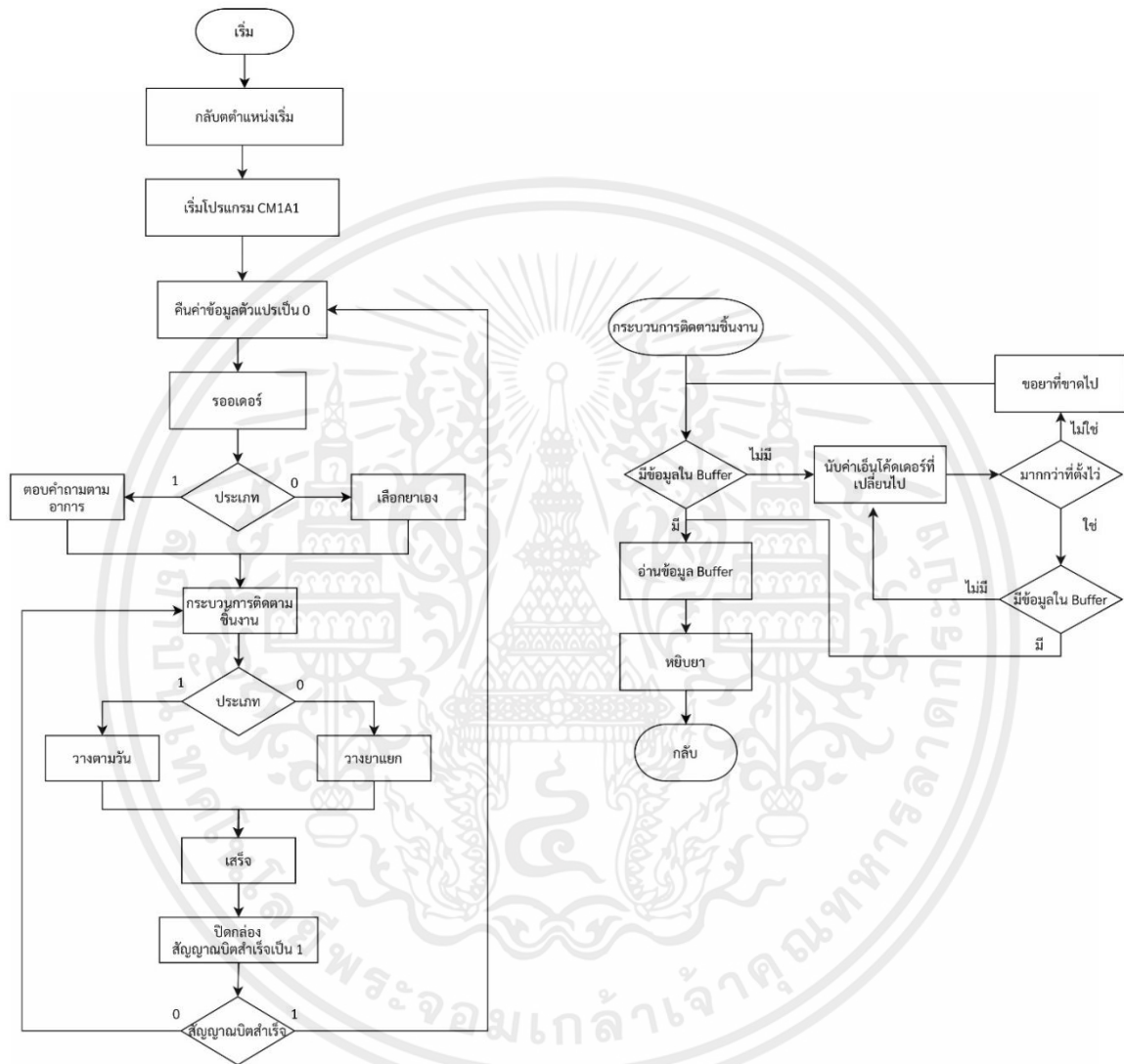
PRNG แกน X คือจุดเริ่มหยิบชิ้นงานและแกน Y คือจุดที่เลิกหยิบชิ้นงาน ให้จุดเริ่มหยิบอยู่ที่ 80
 และหยุดหยิบอยู่ที่ 170

PPT คือตำแหน่งวางชิ้นงานที่มาจากการสอนหุ่นยนต์สการา

P1 คือตำแหน่งเริ่มที่มาจากการสอนหุ่นยนต์สการา

หลังจากเข้าใจการทำงานของโปรแกรมทั้งสองแล้วก็นำโปรแกรมหยิบยาใส่กล่องไปรวมกับ
 โปรแกรม โปรแกรม 1 ที่เป็นส่วนของการหยิบยาและวางยาโดยมีการทำงานตามแล้วได้การทำงานดังรูปที่
 3.138 เริ่มการทำงานเมื่อกดเริ่มโปรแกรม หุ่นยนต์สการากลับตำแหน่งเริ่มต้นพร้อมกับรีเซ็ตค่าตัวแปร
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 112 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นค่าเริ่มต้นที่กำหนดไว้และสั่งเริ่มการทำงานของโปรแกรม CM1 แล้วรอคำสั่งเริ่มการทำงานจากพีแอลซี เมื่อมีคำสั่งจากพีแอลซีมาจะทำการแยกชนิดการจัดวางแล้วจัดตำแหน่งวางตามชนิดแล้วเปิดฝากล่อง เมื่อมียาเคลื่อนที่มาพื้นที่ที่หยิบได้จะเริ่มหยิบทันที เมื่อหยิบยาครบแล้วหุ่นยนต์จะปิดฝากล่องหลังจากปิดฝากล่องเสร็จจะส่งสัญญาณให้พีแอลซีว่าคำสั่งนี้สำเร็จและทำการรีเซตค่ากลับเป็นค่าเริ่มต้น



รูปที่ 3.138 ผังงานการทำงานของโปรแกรม 1 ที่รวมกับโปรแกรมวางยาแล้ว

ในส่วนของโปรแกรม CM1 มีการเปลี่ยนให้รับค่าตามโปรแกรมของวิชันเซนเซอร์ในโปรแกรมวิชันเซนเซอร์มีการส่งค่าเป็นจำนวนของยาที่พบและตำแหน่งของยาให้กับหุ่นยนต์โดยข้อมูลที่ส่งไปยังหุ่นยนต์ถูกส่งเรียงลำดับตามที่ตั้งดังรูปที่ 3.139 โดยกำหนดให้เรียงเป็นรูปแบบเรียงตามลำดับเป็น จำนวนพิกัดในแกน X พิกัดในแกน Y และองศา

1,288.606,21.372,18.707,1,273.432,42.372,-
256.706,0,0,0,1.000,246.873,54.331,0.000,1.000,240.726,29.737,0.000,1.000,258.468,45.767,0.000

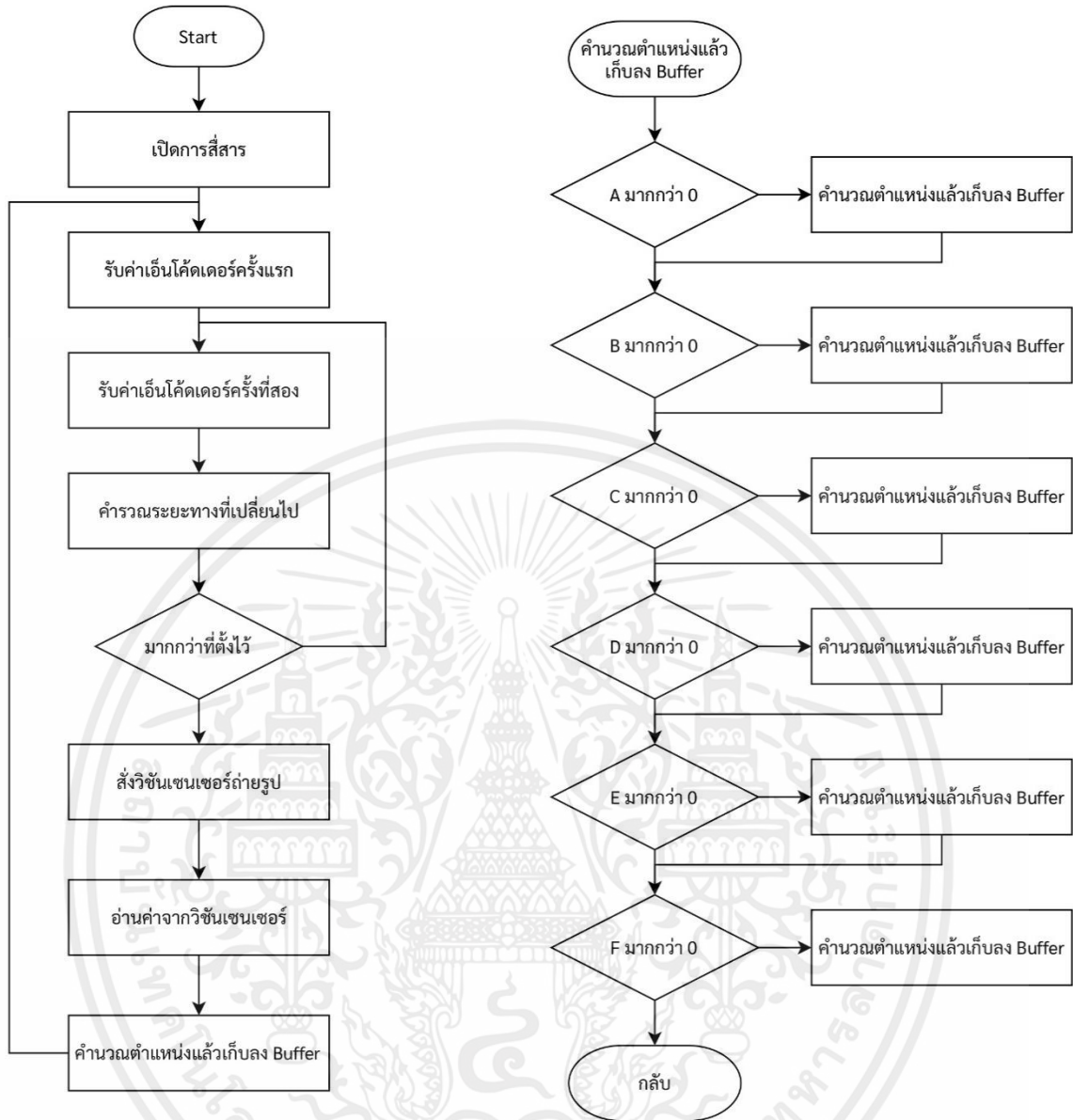
รูปที่ 3.139 รูปแบบการส่งข้อมูลที่วิชันเซนเซอร์ส่งให้หุ่นยนต์

ในโปรแกรมหุ่นยนต์การอ่านค่าจากวิชันเซนเซอร์จำนวนของยาที่ส่งมาถูกนำไปเก็บในตัวแปรที่เป็นตัวเลข ตำแหน่งรวมค่า พิกัดในแกน X พิกัดในแกน Y และองศามาไว้ในตัวแปรตำแหน่งเดียวดังรูปที่ 3.140

```
83 MEP#=MEC# 'Set the encoder pulse current position to the previous value
84 NVIrg #1, 5, MTR1#,MTR2#,MTR3#,MTR4#,MTR5#,MTR6#,MTR7#,MTR8# 'Imaging request
85 'Acquisition of recognition data
86 IF M_NrOpen(1) > 1 THEN Error 5100 AND M_Ous(5001) = 1 'Communication error
87 EBRRead #1,,MNUM,PVS1,MNUM2,PVS2,MNUM3,PVS3,MNUM4,PVS4,MNUM5,PVS5,MNUM6,PVS6 '1
88 obj 0.0
89 If MNUM=0 Then GoTo *Bypass1 'If no workpieces are recognized
90 MX=PVS1.X 'Data acquisition
91 MY=PVS1.Y
```

รูปที่ 3.140 การอ่านข้อมูลที่วิชันเซนเซอร์ส่งมาของหุ่นยนต์

การทำงานของโปรแกรม CM1 ที่เปลี่ยนใหม่มีการทำงานตามผังงาน รูปที่ 3.141 มียาสูงสุด 6 ชนิดใน 1 รอบการถ่ายยา ในยาแต่ละชนิดวิชันเซนเซอร์ทำการส่งตำแหน่งของยามาเพียง 1 ตำแหน่งเท่านั้น หลังจากที่หุ่นยนต์ได้ตำแหน่งของยาหากยาชนิดไหนถูกเจอยาชนิดนั้นจะส่งจำนวนที่เจอเป็น 1 หากไม่เจอจะเป็น 0 เมื่อรับค่าจำนวนและตำแหน่งมาเรียบร้อยแล้วทำการตรวจสอบว่ามียาชนิดนั้นหรือไม่ ยาชนิดไหนที่มีจำนวนยามากกว่า 0 จะไปยังขั้นตอนคำนวณตำแหน่งของการหยิบแล้วทำการเขียนข้อมูลตำแหน่งลงไป Buffer หากยาชนิดไหนจำนวนเป็น 0 ยาชนิดนั้นจะถูกข้ามไปชนิดต่อไป



รูปที่ 3.141 ผังงานการทำงานของโปรแกรม CM1 ที่ปรับปรุงแล้ว

บทที่ 4

การทดสอบฟังก์ชันการทำงานของชุดสาธิต

4.1 กล่าวนำ

ในบทนี้กล่าวถึงการทดสอบฟังก์ชันการทำงานของชุดสาธิตประกอบด้วย การทดสอบการใช้งานเว็บแอปพลิเคชันสร้างคิวอาร์โค้ด การทดสอบการสแกนคิวอาร์โค้ดด้วยโค้ดรีดเดอร์ การสอบความแม่นยำในการหยิบยาแต่ละชนิด และการทำตามคำสั่งซื้อ

4.2 การทดสอบการใช้งานเว็บแอปพลิเคชัน

ในการทดสอบการใช้งานเว็บแอปพลิเคชันทำการทดสอบด้วยการสั่งยาแบบจ่ายตามอาการและแบบเลือกยาเอง ในการเริ่มการใช้งานเมื่อเข้าเว็บแอปพลิเคชันด้วยการสแกนคิวอาร์โค้ดบนหน้าจอเอชเอ็มไอเข้ามาที่หน้าแรกของเว็บแอปพลิเคชันมีหน้าตาเป็นดังรูปที่ 4.1



รูปที่ 4.1 หน้าแรกของเว็บแอปพลิเคชัน

ทำการทดสอบกดสั่งซื้อยาชนิดใดก็ได้แล้วไปที่ตะกร้าปรากฏหน้าต่างะกร้าดังรูปที่ 4.2 ให้พิมพ์ชื่อใส่ลงไปแล้วกดยืนยันหน้าที่มีจำนวนยาที่สั่งซื้อและได้คิวอาร์โค้ดดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 116 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 หน้าตะกร้าสินค้า



รูปที่ 4.3 คิวอาร์โค้ดจากคำสั่งซื้อแบบเลือกยาเอง

ต่อมาทำการทดสอบระบบจ่ายยาตามอาการเมื่อกดเข้าไปมีหน้าให้ใส่ชื่อตามรูปที่ 4.4 หลังจากใส่ชื่อแล้วกดยืนยันแล้วเข้าสู่หน้าที่เป็นการถามอาการดังรูปที่ 4.5 เมื่อเลือกอาการไปจนสิ้นสุดคำถามสุดท้ายมีข้อมูลจำนวนยาพร้อมกับคิวอาร์โค้ดมาให้ดังรูปที่ 4.6

A mobile app interface for entering a name. At the top, there is a red bar with a white shopping cart icon. Below it is a red button with the text "กลับหน้าหลัก" (Return to Home). The main content area is a white rounded rectangle containing the text "NAME" in large, bold, black letters. Below this is a white input field with a light gray border. Underneath the input field is the text "Please enter 1-8 characters or number only" in a small, gray font. At the bottom of the white rounded rectangle is a red button with the text "next" in white.

รูปที่ 4.4 หน้าใส่ชื่อก่อนถามอาการ

A mobile app interface for selecting a symptom. At the top, there is a red bar with a white shopping cart icon. Below it is a red button with the text "กลับหน้าหลัก" (Return to Home). The main content area is a white rounded rectangle containing the text "Question" in large, bold, black letters. Below this is the text "ระบุนอาการของคุณ ?" (Describe your symptoms?). There are five red buttons with white text, each representing a symptom: "ปวดหัว ปวดสามเื้อ เป็นไข" (Headache, fever, chills), "มีน้ำมูก" (Runny nose), "ปวดท้อง" (Stomach pain), "เจ็บคอ" (Sore throat), and "อื่นๆ" (Other). The background of the app is a large, faint watermark of the University of Phayathai logo.

รูปที่ 4.5 หน้าเลือกตอบอาการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 118 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 คิวอาร์โค้ดจากคำสั่งซื้อแบบเลือกอาการ

ทั้งหมดนี้สามารถเข้าเยี่ยมชมเว็บแอปพลิเคชันได้ที่ “<https://scara-demo-melft-kmitl.web.app/>” หรือเข้าชมตามคิวอาร์โค้ดตามดังรูปที่ 4.7



รูปที่ 4.7 คิวอาร์โค้ดของเว็บแอปพลิเคชัน

4.3 การทดสอบฟังก์ชันสแกนคิวอาร์โค้ด

นำข้อมูลคิวอาร์โค้ดจากเว็บแอปพลิเคชันมาทำการทดสอบโค้ดรีดเดอร์ด้วยการสแกนเข้าไปหลายๆ คำสั่งซื้อได้ผลว่าสามารถสแกนได้อย่างง่ายดาย แต่หากจอมือถือสว่างไม่มากพอจะสแกนไม่ได้ให้ทำการปรับความสว่างของจอมือถือเพิ่มขึ้นก็สามารถสแกนได้ปกติ เมื่อสามารถสแกนได้แล้วขั้นที่เอชเอ็มไอเป็นชื่อที่ใส่ลงไปในการสั่งซื้อให้ทำการกดยืนยันว่าถูกต้องด้วยปุ่ม OK ดังรูปที่ 4.8



รูปที่ 4.8 GOT เมื่อสแกนคิวอาร์โค้ดได้

4.4 ความแม่นยำในการหยิบยา

ในการทดสอบความแม่นยำในการหยิบยาใช้คำสั่งซื้อที่มียาชนิดละ 1 เม็ดสแกนคิวอาร์โค้ดเข้าไปทำการใช้คำสั่งซื้อจำนวน 100 ครั้ง โดยนับเฉพาะยาที่ถูกปล่อยออกมาเท่านั้นหากมีการติดขัดในแมกาสินไม่ถูกนับได้ผลตามตารางที่ 4.1 ดังนี้

ตารางที่ 4.1 ความสำเร็จในการหยิบยา

ชนิดยา	หยิบได้	หยิบไม่ได้
A	100	0
B	76	24
C	67	33
D	87	13
E	89	11
F	97	3

ความผิดพลาดของยาแต่ละชนิดที่เกิดขึ้นเกิดจากกรณีดังต่อไปนี้

ยา A เป็นยาชนิดเดียวที่ไม่เคยหยิบพลาดเลยเนื่องจากมีขนาดเม็ดที่ใหญ่แม้โดนทับก็ทับโดนแค่ขอบของยาทำให้ยังสามารถตรวจจับยาได้อยู่

ความผิดพลาดของยา B เกิดจากยาที่มีการตั้งออกมาเป็นบางครั้งดังรูปที่ 4.9 และแปรงกันหน้าจุดปล่อยยาไม่สามารถทำให้ยาล้มได้เนื่องจากยา B มีขนาดเล็กหากลดความสูงของแปรงลงไปอีกจะทำให้ยา C ที่มีขนาดใหญ่ไม่สามารถออกมาจากจุดปล่อยยาได้



รูปที่ 4.9 ความผิดพลาดของยาชนิด B ที่เกิดขึ้นจากการตั้งออกมาจากตัวปล่อยยา

ความผิดพลาดของยา C เกิดจากยาเป็นทรงกระบอกกลม สามารถลื่นได้เมื่อเกิดการสั่นจากการเคลื่อนที่ของหุ่นยนต์ส่งผลให้ตัวเม็ดยาลื่นออกจากสายพานตั้งแต่วันที่วิชันเซนเซอร์เจอยา ดังรูปที่ 4.10 รูปที่ 4.11 และอีกกรณีที่ทำให้ไม่สามารถหยิบตัวยาคได้เนื่องจากยาถูกถ่ายด้วยวิชันเซนเซอร์แล้วแต่ยามีการลื่นเล็กน้อยทำให้ยาเคลื่อนที่ไปจากจุดที่เจอทำให้ไม่สามารถหยิบได้



รูปที่ 4.10 ความผิดพลาดของยาชนิด C ที่เกิดขึ้นจากการลื่นออกจากสายพาน

ความผิดพลาดของยา D และ E เกิดการซ้อนกัน ในยาสองชนิดนี้มักมีชนิดใดชนิดหนึ่งถูกทับทำให้ยาที่ถูกทับไม่สามารถถูกตรวจเจอได้แต่อีกกรณีคือสามารถตรวจเจอยาชิ้นที่อยู่ด้านล่างได้แต่ไม่สามารถตรวจยาที่อยู่ด้านบนได้เนื่องจากเอียง

ความผิดพลาดของยา F มีโอกาสเกิดต่ำมาก แต่ก็มีการหยิบพลาดเกิดขึ้นเนื่องจากยาลิ่งมาทับจนส่งผลให้ทำให้วิชันเซนเซอร์ไม่สามารถแยกได้ว่าเป็นยาชนิดไหน และมีโอกาสที่ยาตั้งออกมาแม้เกิดยากก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและ 121 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทำตามคำสั่งซื้อ

ในการทดสอบนี้ทำการสั่งยาชนิดละ 2 เม็ดจนถึงชนิดละ 5 เม็ด ทำการวัดผลการทดสอบ 3 ระดับ ได้แก่ สามารถทำให้คำสั่งซื้อสำเร็จได้ตั้งแต่การปล่อยยารอบแรกหรือสำเร็จแบบสมบูรณ์แบบ (Perfect Complete) สามารถทำให้คำสั่งซื้อสำเร็จได้แต่ต้องปล่อยยาเพิ่มหากไม่สามารถทำคำสั่งซื้อให้สำเร็จได้ในรอบแรก (Complete) และไม่สามารถทำให้คำสั่งซื้อสำเร็จได้ (Fail) ได้ผลดังตารางที่ 4.2 ดังนี้

จำนวนยาต่อชนิด	Perfect	Complete	Fail
2	3	6	0
3	3	5	0
4	2	8	0
5	0	9	1

ในคำสั่งซื้อตั้งแต่ชนิดละ 2 เม็ดถึงชนิดละ 4 เม็ด มีคำสั่งซื้อที่สามารถทำสำเร็จได้แบบสมบูรณ์แบบเนื่องจากตัวชุดสาธิตสามารถบรรจุและส่งออกยาออกมาครบตามจำนวนยาออกมาจากจุดจ่ายยาในรูปแบบที่สามารถตรวจเจอได้ง่ายดังรูปที่ 4.11



รูปที่ 4.11 รูปแบบการปล่อยยาที่ไม่มีคุณภาพ

ยาที่ถูกปล่อยออกมานั้นหากเกิดข้อผิดพลาดจากเหตุผลในหัวข้อ 4.4 หรือยาออกมาไม่ครบเนื่องจากตัวยาติดอยู่ภายในแมกกาซีน ชุดสาธิตยังสามารถทำตามคำสั่งซื้อจนสำเร็จได้หากสามารถหยิบยาได้หลังจากมีการปล่อยยาที่ขาดเพิ่มเติมดังรูปที่ 4.12 หรือแก๊สกลไกที่มีการติดขัดได้ทันเวลา



รูปที่ 4.12 การปล่อยยาชนิด C ออกมาเพิ่มหลังจากเกิดข้อผิดพลาด

คำสั่งซื้อที่มียาชนิดละ 5 เม็ด เป็นคำสั่งซื้อที่มีโอกาสล้มเหลวเนื่องจากการปล่อยยา 5 เม็ดจำนวน 5 รอบ มีโอกาสน้อยมากที่ไม่เกิดการติดขัดของแมกกาซีนหรือตัวหุ่นยนต์เคลื่อนที่เร็วจนทำให้ชุดสาริตเกิดแรงสั่นสะเทือนส่งผลให้เกิดให้ออกาสผิดพลาดในหัวข้อ 4.4 หรือ เกิดข้อผิดพลาดอื่น ๆ เพิ่มขึ้นตั้งนั้นจากการทดสอบจึงมีโอกาสนี้ไม่สามารถทำให้คำสั่งซื้อสำเร็จได้โดยสมบูรณ์ตามคำสั่งซื้อได้

บทที่ 5

สรุปผล

5.1 สรุปผลการดำเนินงาน

การจัดทำปฏิญญาพันธันนี้ผู้จัดทำได้ดำเนินงานโดยมีขั้นตอนดังนี้

1. ศึกษาการออกแบบฟังก์ชันการทำงานของชุดสาธิต สร้างจากเว็บแอปพลิเคชัน และการเขียนโปรแกรมสำหรับอุปกรณ์มิตซูบิชิ อิเล็กทรอนิกส์ต่าง ๆ
2. ออกแบบฟังก์ชันการทำงานของชุดสาธิต ออกแบบเว็บแอปพลิเคชัน และออกแบบเอชเอ็มไอให้เหมาะสม
3. ติดต่อบริษัทผู้รวมระบบสำหรับการสร้างชุดสาธิต ตามที่ได้ออกแบบไว้
4. สร้างจากเว็บแอปพลิเคชัน เขียนโปรแกรมสำหรับพีแอลซี เขียนโปรแกรมสำหรับหน้าจอเอชเอ็มไอ เขียนโปรแกรมสำหรับวิชชันเซนเซอร์ ตั้งค่าโค้ดรีดเตอร์ และเขียนโปรแกรมสำหรับตัวควบคุมหุ่นยนต์สกการา
5. รวมระบบและทดสอบและแก้ไขระบบให้สมบูรณ์ที่สุดก่อนนำไปโชว์ในงาน Manufacturing Expo 2023 ผลจากการดำเนินงาน โปรแกรมที่ได้ออกแบบไว้สามารถทำงานได้ถูกต้องตามวัตถุประสงค์ที่กำหนดข้างต้น

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

1. ออกแบบฟังก์ชันการทำงานของชุดสาธิตทำได้ยากเพราะไม่มีประสบการณ์ มีผลกับการปรับปรุงแก้ไขทำให้ใช้เวลาในการออกแบบฟังก์ชันการทำงานของชุดสาธิตนานขึ้น
2. วางแผนการเขียนโปรแกรมทำได้ไม่ดี เลยต้องเขียนใหม่และแก้บ่อยครั้ง
3. ทำความเข้าใจผิดในการทำการใช้โปรแกรมการสอบเทียบ ทำให้ใช้เวลาในการปรับความแม่นยำนาน

5.3 แนวทางการแก้ไขปัญหา

1. ศึกษาชุดสาธิตหลาย ๆ แบบให้มากขึ้นและนำมาใช้เป็นตัวอย่าง
2. วางแผนการเขียนโปรแกรมให้รัดกุมมากขึ้น มีการทำผังงานก่อนการเขียนโปรแกรม
3. อ่านและทำความเข้าใจในสิ่งต่าง ๆ อย่างละเอียดเพื่อป้องกันความผิดพลาด

เอกสารอ้างอิง

- [1] หุ่นยนต์สการามิตซูบิชิ อิเล็กทริก. (19 สิงหาคม 2565). เข้าถึงได้จาก https://www.mitsubishielectric.com/fa/th_en/products/rbt/robot/items/horizontal/index.html
- [2] Nilli. (19 สิงหาคม 2565). สมการ จนกลศาสตร์และ Invert จนกลศาสตร์ของ SCARA robot. เข้าถึงได้จาก <https://www.nillisisniega.com/scara-robot/>
- [3] พีแอลซีมิตซูบิชิ อิเล็กทริก. (19 สิงหาคม 2565). เข้าถึงได้จาก https://www.mitsubishielectric.com/fa/th_en/products/cnt/plcr/items/index.html
- [4] Omron. (19 สิงหาคม 2565). วงจรแปลงกระแสไฟฟ้าของ Inverter ที่ใช้ในการควบคุมมอเตอร์. เข้าถึงได้จาก <https://www.ia.omron.com/support/guide/9/introduction.html>
- [5] อินเวอร์เตอร์มิตซูบิชิ อิเล็กทริก. (19 สิงหาคม 2565). เข้าถึงได้จาก https://www.mitsubishifa.co.th/th/Products-Category_Detail.php?id=Mjc=
- [6] Tracking Function Instruction Manual. (7 มีนาคม 2565). เข้าถึงได้จาก https://www.allied-automation.com/wp-content/uploads/2015/02/MITSUBISHI_CR750CR751-series-controller-Tracking-Function-Manual.pdf
- [7] วิชั่นเซนเซอร์มิตซูบิชิ อิเล็กทริก VS80. (30 ตุลาคม 2565). เข้าถึงได้จาก <https://www.mitsubishielectric.com/fa/products/snsr/sensor/pmerit/vs/features/vs80.html>
- [8] PRIMUS. (7 มีนาคม 2566). เอนโค้ดเดอร์. เข้าถึงได้จาก <https://www.primusthai.com/primus/Knowledge/info?ID=157>
- [9] mariaugnich. (19 สิงหาคม 2565). Ethernet และ TCP/IP. เข้าถึงได้จาก <https://devjourneys.com/2020/04/04/tcp-ip>
- [10] CLPA. (19 สิงหาคม 2565). CC-Link IE Field Basic . เข้าถึงได้จาก https://th.cc-link.org/th/cclink/cclinkie/cclinkie_f_b
- [11] CLPA. (19 สิงหาคม 2565). CC-Link. เข้าถึงได้จาก <https://th.cc-link.org/th/cclink/slmp>
- [12] โค้ดรีดเดอร์มิตซูบิชิ อิเล็กทริก. (19 สิงหาคม 2565). เข้าถึงได้จาก <https://www.mitsubishielectric.com/fa/products/snsr/sensor/pmerit/cr/index.html>
- [13] การแปลงข้อมูลคิวอาร์โค้ด. (30 กันยายน 2565). เข้าถึงได้จาก https://www.researchgate.net/figure/Structure-and-Component-s-of-QR-code_fig2_343368029
- [14] รูปแบบ URL. (30 กันยายน 2565). เข้าถึงได้จาก <https://www.hostinger.de/tutorials/was-ist-eine-url/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและ 125 อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [15] Phonbopit, C. (30 กันยายน 2565). Firebase-hosting. เข้าถึงได้จาก
<https://devahoy.com/blog/2019/01/how-to-deploy-react-with-firebase-hosting>
- [16] คู่มือการเชื่อมต่ออินเวอร์เตอร์ FR-E800. (20 มกราคม 2566). เข้าถึงได้จาก
https://www.lcautomation.com/wb_documents/Mitsubishi/Mitsubishi%20E800%20Manual%20-%20Comms.pdf
- [17] คู่มือการเชื่อมต่อไค์ดรีดเตอร์ CF26. (20 มกราคม 2566). เข้าถึงได้จาก
<https://dl.mitsubishielectric.com/dl/fa/document/manual/sensor/bcn-p5999-1074/bcnp59991074e.pdf>

