

สำนักหอสมุดกลาง พระจอมเกล้าพระนครใหม่

หุ่นยนต์ขนถ่ายวัสดุใน โรงงานอุตสาหกรรม
INDUSTRIAL MOBILE ROBOT



เลขหมู่.....
เลขทะเบียน..... 62890
วัน,เดือน,ปี 23 ส.ค. 2549

5 11 2011
1.....

ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทบริหารการศึกษา 2548

ภาควิชา วิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์ขนถ่ายวัสดุในโรงงานอุตสาหกรรม

INDUSTRIAL MOBILE ROBOT

ผู้จัดทำ

- | | |
|---------------------------|-----------------------|
| 1.นาย จิรายุ หอมกลิ่นแก้ว | รหัสประจำตัว 45010128 |
| 2.นาย ขวลิต กิตติไกวัด | รหัสประจำตัว 45010166 |
| 3.นาย ชนาคม อธิธิพลโสกา | รหัสประจำตัว 45010193 |



อาจารย์ที่ปรึกษา

(ดร.อำนาจ คณะรัฐ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ขนถ่ายในอุตสาหกรรม

นายจิรายุ หอมกลิ่นแก้ว 45010128
 นายชวลิต กิตติไคว้ว 45010166
 นายชนาคม อธิธิพลโสภ 45010193
 คร.อำนาจ คณะรัฐ อาจารย์ที่ปรึกษา
 ปีการศึกษา 2548

บทคัดย่อ

โครงการนี้นำเสนอเกี่ยวกับหุ่นยนต์อุตสาหกรรม สำหรับการขนส่งวัสดุในโรงงานอุตสาหกรรมโดย หุ่นยนต์นี้ถูกออกแบบให้ง่ายต่อการใช้งาน การควบคุมแบบพีไอจะถูกระบุการใช้และพัฒนาในไมโครคอนโทรลเลอร์ดีเอสพีเพื่อควบคุมความเร็วและตำแหน่งของดีซีมอเตอร์ โปรแกรมควบคุมที่ใช้ควบคุมหุ่นยนต์สามารถนำมาประยุกต์ใช้ในการสร้างเส้นทางเคลื่อนที่ของหุ่นยนต์ได้หลากหลายแบบ และสามารถกำหนดจุดรับ – ส่งวัสดุได้อย่างไม่จำกัด และหุ่นยนต์ยังสามารถทำงานในโหมดจดจำเส้นทางได้อีกด้วย ข้อดีดังกล่าว ทำให้หุ่นยนต์นี้สามารถนำไปประยุกต์ใช้งานจริงในโรงงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Industrial Mobile Robot

Jirayu Homklinkaew

Chawarit Kittikaiwan

Thanakom Ittiponsopa

Dr. Amnart Kanarat

Advisor

ABSTRACT

This project presents an industrial robot for transferring material or products in industrial factories. The robot is designed for ease of control and usage. A PID feedback control algorithm was implemented on a dsPIC microcontroller to control the velocity and position of a brushed-DC motor. A computer program used to control the robot can be applied in two ways. First, the program can be easily modified to construct different robot paths. Second, the program can memorize routes for transporting material. With these advantages, the robot is suitable for industrial usage.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ อำนวย คณะรัฐ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก ขอขอบคุณ คุณนคร และ คุณอรรณพล ที่ให้คำแนะนำเกี่ยวกับไมโครคอนโทรลเลอร์และวิซวลเบสิก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

จิรายุ หอมกลิ่นแก้ว
ชวลิต กิตติไถวัล
ธนาคม อธิพิลโสภา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	3
1.4 วิธีการดำเนินงาน	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 ระบบควบคุม	4
2.1.1 ระบบควบคุมแบบวงเปิด	4
2.1.2 ระบบควบคุมแบบวงปิด	4
2.1.3 คุณสมบัติของตัวคอนโทรลเลอร์	5
2.2 องค์ประกอบของระบบเซอร์โว	8
2.2.1 หลักการทำงานของดีซีมอเตอร์	9
2.2.2 การแยกประเภทของดีซีมอเตอร์	9
2.2.3 โมเดลคณิตศาสตร์ของดีซีมอเตอร์	17
2.3 วิชาลเบสิก	22
2.4 การสื่อสารแบบอนุกรม	24
2.4.1 มาตรฐานพอร์ตอนุกรมแบบ RS-232	25
2.4.2 การรับและส่งข้อมูลแบบอนุกรมด้วยวิชาลเบสิก	26
2.5 ความรู้เบื้องต้นของไมโครคอนโทรลเลอร์ dsPIC30F2010	34
2.5.1 คุณสมบัติทางเทคนิคของ dsPIC30F2010	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2	สถาปัตยกรรมโดยสรุปของ dsPIC30F2010	37
2.6	หลักการจูนค่าพีไอดี	39
2.6.1	วิธีการปรับค่าคอนโทรลเลอร์แบบพีไอดี	40
2.7	ทฤษฎีหุ่นยนต์ 2 ล้อ	40
บทที่ 3	การออกแบบ และ การสร้าง	42
3.1	การออกแบบส่วนของฮาร์ดแวร์ (Hardware)	42
3.1.1	ชุดบอร์ดดีเอสพีค dsPIC	42
3.1.2	ชุดบอร์ดดีบีคเกอร์ ไอซีดีเอ็กซ์ - 30	44
3.1.3	แหล่งจ่ายไฟ	44
3.1.4	ชุดดีซีมอเตอร์	45
3.1.5	ชุดขั้วมอเตอร์	45
3.1.6	โครงสร้างหุ่นยนต์	47
3.2	การออกแบบส่วนของซอฟต์แวร์ (Software)	48
3.2.1	การติดตั้งซอฟต์แวร์ของ MPLAB IDE และ MPLAB C30	48
3.2.2	การกำหนดการคุณสมบัติการทำงานของ MPLAB C30 ร่วมกับ MPLAB IDE	48
3.2.3	การสร้างโปรเจกต์ภาษา C ด้วย Project Wizzard ของ MPLAB	51
3.2.4	การใช้งาน MPLAB IDE กับ ICDEX-30	60
บทที่ 4	การทดลองและผลการทดลอง	65
4.1	การทดลองจูนค่าพีไอดีและการทดสอบการทำงานของพีไอดี	65
4.2	การทดสอบการเคลื่อนที่ของหุ่นยนต์ในระยะทางสั้น	67
4.3	การทดสอบการเคลื่อนที่ของหุ่นยนต์ในโหมดโปรแกรมเส้นทางและ โหมดจำเส้นทางใน รูปแบบต่างๆ	74
บทที่ 5	วิจารณ์และสรุปผล	83
5.1	บทสรุป	83
5.2	ปัญหาที่เกิดขึ้นในการจัดทำโครงงาน	83
5.3	แนวทางการแก้ไขและพัฒนา	84
ภาคผนวก ก. ผังการทำงาน โปรแกรม		
ภาคผนวก ข. การประกอบและใช้งานหุ่นยนต์ต้นแบบ		
ภาคผนวก ค. Library , Source Code ในแผ่นซีดี		
บรรณานุกรม		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้าที่
ตารางที่ 2-1 แสดง Event ที่เกิดขึ้นจาก Mouse	23
ตารางที่ 2-2 แสดง Event ที่เกิดขึ้นจาก Keyboard	23
ตารางที่ 4-1 ผลการเปรียบเทียบของผลตอบสนองที่ค่าความเร็วต่างๆ	67
ตารางที่ 4-2 ผลการเคลื่อนที่ระยะทาง 50 ซม	68
ตารางที่ 4-3 ผลการเคลื่อนที่ระยะทาง 100 ซม	69
ตารางที่ 4-4 ผลการเคลื่อนที่ที่รัศมีการเลี้ยว 39 ซม(คิดเป็นระยะทาง 63 cm)	70
ตารางที่ 4-5 ผลการเคลื่อนที่ที่รัศมีการเลี้ยว 50 ซม(คิดเป็นระยะทาง 80 cm)	71
ตารางที่ 4-6 ผลการเคลื่อนที่ระยะทาง 50 ซม	71
ตารางที่ 4-7 ผลการเคลื่อนที่ระยะทาง 100 ซม	72
ตารางที่ 4-8 ผลการเคลื่อนที่ที่รัศมีการเลี้ยว 39 ซม(คิดเป็นระยะทาง 63 cm)	73
ตารางที่ 4-9 ผลการเคลื่อนที่ที่รัศมีการเลี้ยว 50 ซม(คิดเป็นระยะทาง 80 cm)	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้าที่
รูปที่ 2-1 ระบบควบคุมแบบวงเปิด	4
รูปที่ 2-2 ระบบควบคุมแบบวงปิด	4
รูปที่ 2-3 ตัวคอนโทรลเลอร์แบบปร็อพออร์ชันแนล	5
รูปที่ 2-4 แสดงผลตอบสนองแบบพี	5
รูปที่ 2-5 ตัวคอนโทรลเลอร์แบบอินติกรัล	6
รูปที่ 2-6 แสดงผลตอบสนองแบบไอ	6
รูปที่ 2-7 ตัวคอนโทรลเลอร์แบบดิริเวทีฟ	6
รูปที่ 2-8 ตัวคอนโทรลเลอร์แบบปร็อพออร์ชันแนล-อินติกรัล	7
รูปที่ 2-9 ตัวคอนโทรลเลอร์แบบปร็อพออร์ชันแนล-อินติกรัล	7
รูปที่ 2-10 ตัวคอนโทรลเลอร์แบบปร็อพออร์ชันแนลอินติกรัลดิริเวทีฟ	8
รูปที่ 2-11 แสดงผลตอบสนองแบบพีไอดี	8
รูปที่ 2-12 เซอร์โวมอเตอร์แบบโรเตอร์เป็นแกนเหล็ก	10
รูปที่ 2-13 เซอร์โวมอเตอร์แบบโรเตอร์เป็นรูปถ้วย	10
รูปที่ 2-14 ดิซีมอเตอร์แบบอาร์มาเจอร์ตัดอ่อนุกรมกับขดลวดสนามแม่เหล็ก	11
รูปที่ 2-15 คุณสมบัติระหว่างความเร็วและแรงบิดของดิซีมอเตอร์อ่อนุกรม	11
รูปที่ 2-16 ดิซีมอเตอร์แบบแยกปรับสนามแม่เหล็กได้	11
รูปที่ 2-17 แสดงคุณสมบัติระหว่างความเร็วและแรงบิดของซันท์มอเตอร์ภายใต้ภาวะ อาร์มาเจอร์โวลท์ เด็กลงที่และการกระตุ้นสนามแม่เหล็กคงที่	12
รูปที่ 2-18 ดิซีมอเตอร์แบบฟิลด์เป็นแม่เหล็กถาวร	12
รูปที่ 2-19 แสดงรูปหน้าตัดของดิซีมอเตอร์แบบอาร์มาเจอร์เป็นแกนเหล็กส่วนฟิลด์เป็นแม่เหล็ก	13
รูปที่ 2-20 แสดงรูปหน้าตัดของดิซีมอเตอร์แบบมีขดลวดบนพื้นผิวและฟิลด์เป็นแม่เหล็กถาวร	14
รูปที่ 2-21 หน้าตัดของดิซีมอเตอร์แบบมีโรเตอร์เป็นขดลวดเคลื่อนที่และฟิลด์เป็นแม่เหล็กถาวร	14
รูปที่ 2-22 แสดงถึงรูปหน้าตัดซึ่งมองด้านข้างของดิซีมอเตอร์แบบมีโรเตอร์เป็นขดลวดหมุน	15
รูปที่ 2-23 แสดงถึงดิซีมอเตอร์แบบโรเตอร์เป็นขดลวดหมุนมีรูปร่างเป็นจาน	15
รูปที่ 2-24 รูปหน้าตัดของดิซีมอเตอร์แบบไม่มีแปรงถ่านและโรเตอร์เป็นแม่เหล็กถาวร	16
รูปที่ 2-25 แสดงถึงส่วนประกอบที่สำคัญในระบบการบังคับดิซีมอเตอร์แบบไม่มีแปรงถ่าน	16
รูปที่ 2-26 ตัวอย่างโครงสร้างอินเวอร์เตอร์ของดิซีมอเตอร์แบบไม่มีแปรงถ่านชนิด 2 เฟส โดยใช้ เพาเวอร์ซัพพลายสองค่า	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-27 ตัวอย่างอินเวอร์ทของดิซิมอเตอร์ของดิซิมอเตอร์แบบไม่มีแปรงถ่านชนิด 3 เฟส โดยใช้ เพาเวอร์ซัพพลายเพียงค่าเดียว	17
รูปที่ 2-28 โมเดลของดิซิมอเตอร์แบบฟิลด์แยกกระตุ้น	18
รูปที่ 2-29 แสดงถึงแรงบิดต่างๆที่เกิดขึ้นต่อ โทลคของมอเตอร์	19
รูปที่ 2-30 บล็อกไดอะแกรมของดิซิมอเตอร์โมเดล	21
รูปที่ 2-31 รูปแบบของข้อมูลอนุกรม	24
รูปที่ 2-32 รูปแบบของข้อมูลแบบอะซิงโครนัส	24
รูปที่ 2-33 แสดงลักษณะของล้อที่มีความเร็วต่างกัน	40
รูปที่ 3-1 บอร์ดทดลอง (NX-dsPIC28)	42
รูปที่ 3-2 บอร์ดพัฒนา (JX-dsPIC28)	42
รูปที่ 3-3 ส่วนวงจรที่สมบูรณ์	43
รูปที่ 3-4 ชุดบอร์ดดีบั๊กเกอร์ ไอซีดีอีเอ็กซ์ - 30	44
รูปที่ 3-5 แหล่งจ่ายไฟเลี้ยงบอร์ดชุดบอร์ดดีเอสพี	45
รูปที่ 3-6 แหล่งจ่ายไฟชุดขับเคลื่อนมอเตอร์	45
รูปที่ 3-7 แสดงมอเตอร์กระแสตรง	45
รูปที่ 3-8 แสดงบล็อกไดอะแกรมของ H-BRIDGE DRIVE	46
รูปที่ 3-9 ตำแหน่งและรายละเอียดข้อต่อ	46
รูปที่ 3-10 มุมมองโครงสร้างด้านบน	47
รูปที่ 3-11 มุมมองโครงสร้างด้านหน้า	47
รูปที่ 3-12 มุมมองโครงสร้างด้านข้าง	47
รูปที่ 3-13 แสดงการกำหนดพาธให้กับ MPLAB ASM30 Assembler	48
รูปที่ 3-14 แสดงการกำหนดพาธให้กับ MPLAB C30 C Compiler	49
รูปที่ 3-15 แสดงการกำหนดพาธให้กับ MPLAB LINK30 Object Linker	49
รูปที่ 3-16 แสดงหน้าต่าง set language Tool Locations ที่ถูกเปิดขึ้น	50
รูปที่ 3-17 แสดงการกระจายรากหน้าต่าง set language Tool Locations ทั้งหมด	50
รูปที่ 3-18 แสดงการกำหนดพาธให้กับ Include Search Part, \$(INCDIR)	51
รูปที่ 3-19 แสดงการกำหนดพาธให้กับ Library Search Part, \$(LIBDIR)	51
รูปที่ 3-20 แสดง MPLAB Editor สำหรับการสร้างไฟล์ใหม่	52
รูปที่ 3-21 แสดงการจัดเก็บไฟล์ first.c ไว้ภายในโฟลเดอร์ first (ผู้พัฒนาต้องสร้างเอง)	53
รูปที่ 3-22 แสดงเมนูการสร้างโปรเจกต์แบบ Project Wizard	54
รูปที่ 3-23 แสดงหน้าต่าง Project Wizard ที่ถูกเปิดขึ้นมาใหม่	54
รูปที่ 3-24 แสดงหน้าต่างหัวข้อ Select a device	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-25 แสดงหน้าต่างหัวข้อ Select a language toolsuite	55
รูปที่ 3-26 แสดงหน้าต่างหัวข้อ Name your project	56
รูปที่ 3-27 แสดงหน้าต่างหัวข้อ Add any existing files to your project	56
รูปที่ 3-28 แสดงไฟล์ที่ถูกเลือกเพื่อผนวกเข้ากับโปรเจกต์ first ที่กรอบด้านขวามือ	57
รูปที่ 3-29 แสดงหน้าต่างรายละเอียดของเครื่องมือที่ใช้และพาธของโปรเจกต์	57
รูปที่ 3-30 แสดงหน้าต่างผังงานของโปรเจกต์ first	58
รูปที่ 3-31 แสดงหน้าต่างโปรแกรมของไฟล์ first.c ที่ถูกเปิดขึ้นจากโปรเจกต์ first	58
รูปที่ 3-32 แสดงการเลือกเมนูคำสั่งคอมไพล์โปรเจกต์	59
รูปที่ 3-33 แสดงหน้าต่าง Output สำหรับรายงานผลการคอมไพล์	59
รูปที่ 3-34 แสดงไฟล์ first.hex ที่ถูกสร้างขึ้นให้อยู่ในพาธเดียวกับโปรเจกต์	60
รูปที่ 3-35 แสดงหน้าต่าง System Properties	60
รูปที่ 3-36 แสดงหน้าต่าง ของ System Properties	60
รูปที่ 3-37 หน้าตา Communications Port Properties	61
รูปที่ 3-38 แสดงรายการของ Port Settings	61
รูปที่ 3-39 แสดงหน้าต่าง Advance port setting	61
รูปที่ 3-40 แสดงหน้าต่างเลือกไฟล์ ICD10010206.hex	62
รูปที่ 3-41 แสดงหน้าต่าง ICD2 Setting	62
รูปที่ 3-42 แสดงหน้าต่างรายละเอียดของแท็บ Communication	63
รูปที่ 3-43 แสดงหน้าต่าง Output ที่แจ้งสถานการณ์เชื่อมต่อสัฟร้อมทำงาน	63
รูปที่ 3-44 แสดงการเลือกเมนู File → Import ...	64
รูปที่ 3-45 แสดงขั้นตอนการดาวน์โหลด	64
รูปที่ 4-1 บล็อกไดอะแกรมระบบควบคุม	65
รูปที่ 4-2 ลักษณะผลตอบสนองของพีไอดีที่ความเร็วคงที่ 8 ชม/วินาที	65
รูปที่ 4-3 ลักษณะผลตอบสนองของพีไอดีที่ความเร็วคงที่ 16 ชม/วินาที	66
รูปที่ 4-4 ลักษณะผลตอบสนองของพีไอดีที่ความเร็วคงที่ 22 ชม/วินาที	66
รูปที่ 4-5 ลักษณะผลตอบสนองของพีไอดีที่ความเร็วคงที่ 30 ชม/วินาที	66
รูปที่ 4-6 ผลตอบสนองของพีไอดีเมื่อทำการเพิ่มความเร็วจนถึง 22 ชม/วินาที	67
รูปที่ 4-7 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 50 ชม	68
รูปที่ 4-8 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 100 ชม	69
รูปที่ 4-9 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 80 ชม	70
รูปที่ 4-10 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 50 ชม	71
รูปที่ 4-11 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 100 ชม	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-12 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 80 ซม	73
รูปที่ 4-13 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อขวา	74
รูปที่ 4-14 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อซ้าย	74
รูปที่ 4-15 แสดงการเคลื่อนที่ในแนวแกน X	75
รูปที่ 4-16 แสดงการเคลื่อนที่ในแนวแกน Y	75
รูปที่ 4-17 แสดงการเปลี่ยนแปลงทิศทางของหุ่นยนต์ในรูปแบบองศา	75
รูปที่ 4-18 แสดงภาพรวมการเคลื่อนที่ของหุ่นยนต์	76
รูปที่ 4-19 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อขวา	76
รูปที่ 4-20 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อซ้าย	76
รูปที่ 4-21 แสดงการเคลื่อนที่ในแนวแกน X	77
รูปที่ 4-22 แสดงการเคลื่อนที่ในแนวแกน Y	77
รูปที่ 4-23 แสดงการเปลี่ยนแปลงทิศทางของหุ่นยนต์ในรูปแบบองศา	77
รูปที่ 4-24 แสดงภาพรวมการเคลื่อนที่ของหุ่นยนต์	78
รูปที่ 4-25 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อขวา	78
รูปที่ 4-26 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อซ้าย	78
รูปที่ 4-27 แสดงการเคลื่อนที่ในแนวแกน X	79
รูปที่ 4-28 แสดงการเคลื่อนที่ในแนวแกน Y	79
รูปที่ 4-29 แสดงการเปลี่ยนแปลงทิศทางของหุ่นยนต์ในรูปแบบองศา	79
รูปที่ 4-30 แสดงภาพรวมการเคลื่อนที่ของหุ่นยนต์	80
รูปที่ 4-31 แสดงค่าความเร็วในรูปผลตอบสนองที่ได้จากการจำเส้นทางของล้อขวา	80
รูปที่ 4-32 แสดงค่าความเร็วในรูปผลตอบสนองที่ได้จากการจำเส้นทางของล้อซ้าย	80
รูปที่ 4-33 แสดงค่าความเร็วในรูปผลตอบสนองที่ได้จากการเคลื่อนที่ของล้อขวา	81
รูปที่ 4-34 แสดงค่าความเร็วในรูปผลตอบสนองที่ได้จากการเคลื่อนที่ของล้อซ้าย	81
รูปที่ 4-35 แสดงภาพรวมการเคลื่อนที่ของหุ่นยนต์	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ยุคเริ่มต้นของการส่งเสริมอุตสาหกรรมในประเทศไทยจะเห็นได้ว่ามีโรงงานต่าง ๆ เข้ามาตั้งฐานผลิตในเมืองไทยจำนวนมากทำให้เกิดนิคมอุตสาหกรรมขึ้นหลายแห่ง ทั้งนี้เนื่องจากรัฐบาลมีนโยบายส่งเสริมอุตสาหกรรมที่ชัดเจน ค่าแรงงานถูก ลดรายจ่ายเนื่องจากภาษีการนำเข้าของสินค้า และวัตถุดิบบางตัว แต่ ณ ปัจจุบันนี้ค่าแรงบ้านเราสูงขึ้นและสูงกว่าประเทศเพื่อนบ้าน เช่น จีน เวียดนาม อินโดนีเซีย ฯลฯ ในขณะที่เดียวกันคุณภาพแรงงานไม่ได้มาตรฐาน ขาดความรู้ และทักษะจึงทำให้หลายบริษัทได้ย้ายฐานการผลิตไปยังประเทศที่มีค่าแรงงานถูกกว่า เหล่านี้ทำให้คนไทยตกงานมากขึ้น และอีกหลายบริษัทที่ไม่ได้ย้ายฐานการผลิตแต่พยายามปรับตัวเอง โดยมีกรนำเทคโนโลยีระบบอัตโนมัติเข้ามาใช้งานเพื่อให้สินค้าสามารถแข่งขันในตลาดโลกได้ทั้งในเรื่องราคา และคุณภาพ โดยเฉพาะในเรื่องคุณภาพ ปัจจุบันโรงงานที่ผลิตสินค้าส่งออกหรือส่งให้กับลูกค้าที่เป็นบริษัทของต่างประเทศมักจะประสบปัญหาในเรื่องคุณภาพ มีทั้ง ผลิตสินค้าไม่ได้มาตรฐานตามที่ลูกค้ากำหนด หรือ ผลิตสินค้าไม่ทันตามกำหนดเวลา อาจเนื่องจากการเปลี่ยนรุ่นผลิตภัณฑ์อยู่บ่อยๆ ต้องใช้เวลาในการปรับเปลี่ยนรูปแบบการผลิต ปัจจุบันจึงมีการนำเทคโนโลยีต่าง ๆ เข้ามาใช้ หนึ่งในเทคโนโลยีที่มีความยืดหยุ่นสูง ได้แก่ หุ่นยนต์อุตสาหกรรม เนื่องจากการเปลี่ยนการทำงานสามารถทำได้โดยการเปลี่ยน โปรแกรม นอกจากนี้คุณภาพของผลิตภัณฑ์ที่มีความสม่ำเสมอเป็นมาตรฐานเดียวกัน

ปัจจุบันเทคโนโลยีของหุ่นยนต์นั้นได้เจริญก้าวหน้าไปมาก ต่างจากเมื่อก่อนที่หุ่นยนต์จะถูกนำไปใช้ ในงานอุตสาหกรรมเป็นส่วนใหญ่ แต่ปัจจุบันมีการนำมาใช้มากขึ้น ไม่ว่าจะเป็นหุ่นยนต์ที่ใช้ในทางการแพทย์ หุ่นยนต์สำหรับงานสำรวจ หุ่นยนต์ที่ใช้งานในอวกาศ หรือแม้แต่หุ่นยนต์ที่ถูกสร้างขึ้นเพื่อเป็นเครื่องเล่นของมนุษย์ จนกระทั่งในปัจจุบันนี้ได้มีการพัฒนาให้หุ่นยนต์นั้นมีลักษณะที่คล้ายมนุษย์ เพื่อให้อาศัยอยู่ร่วมกันกับมนุษย์ ให้ได้ในชีวิตประจำวัน ถึงแม้ว่าหุ่นยนต์นั้นจะถูกพัฒนาไปในหลายรูปแบบ แต่สามารถแบ่งประเภทของหุ่นยนต์ ออกเป็นสองประเภทใหญ่ ๆ ได้ดังนี้

ประเภทแรก คือหุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (Fixed Robot) ไม่สามารถเคลื่อนที่ไปไหนได้ด้วยตัวเอง หุ่นยนต์ประเภทนี้จะมีลักษณะเป็นแขนกล สามารถเคลื่อนไหวได้เฉพาะแต่ละข้อต่อ ภายในตัวเองเท่านั้น ส่วนมากมักถูกนำไปใช้ในโรงงานอุตสาหกรรม เช่น โรงงานประกอบรถยนต์ เป็นต้น

ประเภทที่สอง คือหุ่นยนต์ชนิดที่เคลื่อนที่ได้ (Mobile Robot) หุ่นยนต์ประเภทนี้จะสามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเอง บ้างก็เคลื่อนที่โดยการใช้ล้อ หรือบางแบบก็เคลื่อนที่โดยการไต่ขา ซึ่งหุ่นยนต์ประเภทนี้ ส่วนใหญ่ยังเป็นงานวิจัยที่อยู่ในห้องทดลอง เพื่อพัฒนาออกมาใช้งานในรูปแบบต่างๆ เช่น หุ่นยนต์สำรวจดาวอังคาร ขององค์การนาซ่า แต่ปัจจุบันก็ได้มีการพัฒนา ให้มีลักษณะเป็นสัตว์เลี้ยงอย่างสุนัข เพื่อให้มาเป็นเพื่อนเล่นกับคน หรือแม้กระทั่งมีการพัฒนาหุ่นยนต์ให้สามารถเคลื่อนที่แบบสองขาได้อย่างมนุษย์ เพื่ออนาคตจะสามารถนำไปใช้ในงานที่มีความเสี่ยงต่ออันตรายแทนมนุษย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนในปัจจุบันเทคโนโลยีได้พัฒนาไปอย่างรวดเร็วในทุกๆด้าน ไม่ว่าจะเป็นด้านการสื่อสาร โทรคมนาคม เครื่องจักรกล การขนส่ง ไมโครคอนโทรลเลอร์ ฯลฯ ถ้าหากนำเทคโนโลยีเหล่านี้มาประยุกต์ใช้เข้าด้วยกันก็จะทำให้เกิดสิ่งประดิษฐ์ใหม่ๆ ซึ่งช่วยอำนวยความสะดวก และก่อให้เกิดประโยชน์มากมาย รวมถึงความหลากหลายในการใช้งานในด้านต่างๆด้วย

ตัวอย่างโครงการหุ่นยนต์เคลื่อนที่ที่ได้เคยมีการประดิษฐ์ทั้งในประเทศไทยและต่างประเทศมีดังนี้ โครงการหุ่นยนต์เคลื่อนที่ โดยใช้ไมโครคอนโทรลเลอร์ MC68HC11 โครงสร้างของหุ่นยนต์เป็นลักษณะทรงกลม โดยการเคลื่อนที่ของหุ่นยนต์นั้นเป็นแบบการเคลื่อนที่ตามเส้น [1] และโครงการหุ่นยนต์รับส่งเอกสารชนิดเคลื่อนที่ตามเส้น ซึ่งเป็นหุ่นยนต์ส่งเอกสารหรือพัสดุโดยใช้ไมโครคอนโทรลเลอร์ MCS-51 โดยการเคลื่อนที่ของหุ่นยนต์นั้นเป็นแบบเคลื่อนที่ตามเส้น ซึ่งเป็นต้นแบบในงานส่งของแทนคนในโรงงานอุตสาหกรรม [2] ในส่วนต่างประเทศ หุ่นยนต์เคลื่อนที่สองล้อ โครงสร้างภายในจำลองระบบการทำงานของซัซซิริยอนต์ ทำให้สามารถเคลื่อนที่ไปบนเส้นทางที่ขรุขระได้ และสามารถรองรับแรงกระแทกสูงได้ [3] และ หุ่นยนต์เคลื่อนที่สองล้อ ขับเคลื่อนโดยระบบโซเวอร์มอเตอร์ โดยใช้เซ็นเซอร์สามตัวตรวจจับในการเคลื่อนที่ตามเส้น [4] จากตัวอย่างข้างต้น จะเห็นว่าการเคลื่อนที่ของหุ่นยนต์นั้น เป็นการเคลื่อนที่แบบจับเส้น ทำให้เกิดข้อจำกัดในการเคลื่อนที่ ซึ่งโครงการนี้จะเป็นการแก้ไข ข้อบกพร่องดังกล่าวโดยการพัฒนาการเคลื่อนที่ของหุ่นยนต์ โดยการใส่เส้นทางให้กับหุ่นยนต์ ดังนั้น หุ่นยนต์จะสามารถเคลื่อนที่ได้เองโดยไม่ต้องตรวจจับเส้นแนวการเคลื่อนที่ อีกทั้งยังมีการใช้ ไมโครคอนโทรลเลอร์ดีเอสพีค ซึ่งเป็นไมโครคอนโทรลเลอร์ที่ได้มีการพัฒนาฟังก์ชันการทำงานให้ เหมาะสมกับการทำงานของมอเตอร์ได้ดียิ่งขึ้น

หุ่นยนต์ขนถ่ายวัสดุ ในโรงงานอุตสาหกรรมนี้เป็นหุ่นยนต์อัตโนมัติ ซึ่งใช้ไมโครคอนโทรลเลอร์ดีเอสพีค ในการควบคุมมอเตอร์ซึ่งมีเอนโคเดอร์รวมอยู่ด้วย การควบคุมมอเตอร์นี้จะเป็นการควบคุมทั้งความเร็ว และความเร่งของมอเตอร์ โดยนำเอาการควบคุมแบบพีไอดี มาช่วยควบคุมให้การทำงานมีความเที่ยงตรง แม่นยำมากยิ่งขึ้น โดยขั้นตอนการสั่งงานนั้นมาจากการเขียนโปรแกรมด้วยภาษาซี ซึ่งจะทำให้ง่าย ต่อการเขียนโปรแกรม และแก้ไขโปรแกรมการทำงานของหุ่นยนต์ โดยโปรแกรมที่เขียนจะถูกเขียนลง ในไมโครคอนโทรลเลอร์ เพื่อควบคุมการทำงานของมอเตอร์และสามารถรับค่าระยะทางการเคลื่อนที่ ในรูปแบบต่างๆได้ จากที่กล่าวมาข้างต้นนี้ ทำให้หุ่นยนต์สามารถจำเส้นทางและสามารถเคลื่อนที่ไปยัง เป้าหมายได้เอง สำหรับปัญญาประดิษฐ์นี้จะนำเสนอหุ่นยนต์ขนถ่ายอัตโนมัติที่สามารถจำเส้นทางได้ ซึ่งจะสามารถนำไปประยุกต์ใช้ได้จริง

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 ศึกษาหลักการการทำงานเบื้องต้นของหุ่นยนต์
- 1.2.2 เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์
- 1.2.3 เพื่อศึกษาเกี่ยวกับโปรแกรมควบคุมการทำงานของหุ่นยนต์
- 1.2.4 นำความรู้ความสามารถทางด้านวิศวกรรมเครื่องกลและด้านต่างๆมาประยุกต์กับโครงการ
- 1.2.5 เพื่อประดิษฐ์หุ่นยนต์ต้นแบบตั้งแต่เริ่มต้น จนเสร็จสิ้นเป็น หุ่นยนต์ขนถ่ายโดยสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงการ

- 1.3.1 ศึกษาและออกแบบโครงสร้างการทำงานของหุ่นยนต์
- 1.3.2 เขียนโปรแกรมเพื่อควบคุมมอเตอร์
- 1.3.3 ประดิษฐ์หุ่นยนต์ให้สามารถจดจำเส้นทางและสามารถเคลื่อนที่ไปยังเป้าหมาย
- 1.3.4 ประดิษฐ์หุ่นยนต์ให้สามารถเคลื่อนที่ไปยังเป้าหมาย โดยการกำหนดเส้นทางให้หุ่นยนต์

1.4 วิธีการดำเนินงาน

- 1.4.1 ศึกษาข้อมูลจากหัวข้อโครงการที่ได้รับ
- 1.4.2 ศึกษาข้อมูล โดยละเอียดเกี่ยวกับการควบคุมมอเตอร์ด้วยไมโคร คอนโทรลเลอร์ดีเอสพี
- 1.4.3 ทำการเขียนโปรแกรมด้วยภาษาซีเพื่อให้หุ่นยนต์สามารถจดจำ เส้นทางและขับเคลื่อนมอเตอร์
- 1.4.4 นำระบบควบคุมแบบพีไอดีมาใช้เพื่อช่วยให้การควบคุมมอเตอร์มีประสิทธิภาพมากขึ้น
- 1.4.5 ทำการทดลองหุ่นยนต์ให้สามารถทำงานได้จริง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 เพื่อให้หุ่นยนต์สามารถทำงานได้ตามขอบเขตของโครงการ
- 1.5.2 เพิ่มประสบการณ์และความรู้ในด้านต่างๆแก่นักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 ระบบควบคุม

ในระบบคอนโทรลมอเตอร์ จุดมุ่งหมายของตัวคอนโทรลเลอร์ก็คือเพื่อที่จะปรับตัวแปรเอาต์พุตของระบบให้เข้าสู่ค่าที่กำหนดไว้(ตัวแปรอ้างอิง) และจะต้องรักษาค่าตัวแปรดังกล่าวให้อยู่ที่ค่าที่กำหนดนั้นเพื่อให้เป็นไปตามเป้าหมายนี้ ตัวคอนโทรลเลอร์จะต้องแก้ไขผลกระทบของสัญญาณรบกวนจากภายนอกในวิถีทางที่เหมาะสม

ชนิดของการควบคุมหรือกิจกรรมการควบคุม หมายถึง ชนิดหรือวิธีการทำงานของส่วนที่เป็นตัวควบคุม โดยการควบคุมชนิดของการควบคุมหรือตัวควบคุมเพื่อแก้ไขค่าความคลาดเคลื่อนระหว่างเอาต์พุตกับอินพุตของทั้งระบบ

2.1.1 ระบบควบคุมแบบวงเปิด

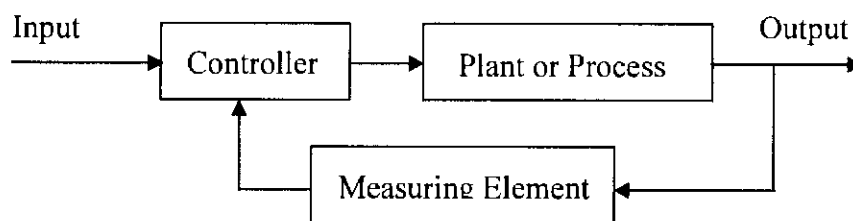
ระบบควบคุมแบบวงเปิดเป็นระบบควบคุมที่เอาต์พุตของระบบจะไม่มีผลต่อการควบคุมเลย นั่นคือในกรณีของระบบควบคุมแบบนี้เอาต์พุตของระบบจะไม่ถูกวัดหรือถูกป้อนกลับเพื่อนำมาเปรียบเทียบกับอินพุต รูปที่ 2-1 เป็นบล็อกไดอะแกรมแสดงถึงความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของระบบควบคุมแบบวงเปิด



รูปที่ 2-1 ระบบควบคุมแบบวงเปิด

2.1.2 ระบบควบคุมแบบวงปิด

ระบบควบคุมแบบนี้ เป็นระบบควบคุมที่นำสัญญาณเอาต์พุตที่ได้จากกระบวนการนั้นป้อนกลับมาเปรียบเทียบกับสัญญาณอินพุตที่ต้องการ ซึ่งจะได้สัญญาณความคลาดเคลื่อนโดยเป็นสัญญาณความแตกต่างระหว่างสัญญาณอินพุตและสัญญาณป้อนกลับและจะถูกป้อนให้กับตัวควบคุมอีกครั้ง เพื่อที่จะเป็นการลดค่าความคลาดเคลื่อนให้น้อยลง รูปที่ 2-2 แสดงบล็อกไดอะแกรมระบบควบคุม แบบปิดดังรูป



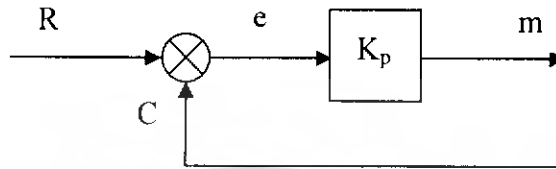
รูปที่ 2-2 ระบบควบคุมแบบวงปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 คุณสมบัติของตัวคอนโทรลเลอร์

2.1.3.1 ปรีออปฟอร์ชันแนลคอนโทรล (พี - คอนโทรล)

สัญญาณตัวแปรเพื่อการแก้ไขของตัวคอนโทรลเลอร์จะเป็นสัดส่วนกับสัญญาณเออร์เรอร์ ดังแสดงในรูปที่ 2-3



รูปที่ 2-3 ตัวคอนโทรลเลอร์แบบปรีออปฟอร์ชันแนล

ดังนั้นสมการของค่าสัญญาณควบคุม (m) หาได้ดังนี้

$$m = K_p e + m(0) \quad (1)$$

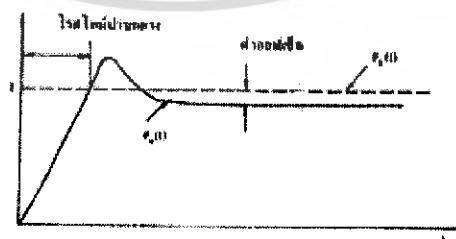
เมื่อ K_p คือกำลังขยายของตัวคอนโทรลเลอร์แบบปรีออปฟอร์ชันแนล

$m(0)$ คือเอาต์พุตของตัวคอนโทรลเลอร์เมื่อเออร์เรอร์เป็นศูนย์

การคอนโทรลแบบนี้มักจะแสดงอยู่ในรูปของปรีออปฟอร์ชันแนลแบนด์ (PB) มีหน่วยเป็นเปอร์เซ็นต์ ค่า PB คือค่าสัญญาณเออร์เรอร์ที่ทำให้สัญญาณควบคุมเปลี่ยนไปร้อยละ 50 เมื่อ PB เท่ากับ 50 % สัญญาณควบคุมจะเปลี่ยนไป 100 %

$$PB = 1/K_p \times 100$$

ปรีออปฟอร์ชันแนลคอนโทรลจะใช้งานได้เฉพาะในระบบที่ไม่มีการเปลี่ยนแปลงโหลดมากนัก การเกิดออฟเซ็ท แสดงดังรูปที่ 2-4 เป็นคุณลักษณะของระบบควบคุมแบบสัดส่วน ทั้งนี้เนื่องจากการทำงานของระบบควบคุมแบบสัดส่วนนั้น ไม่สามารถควบคุมระบบที่มีโหลดเปลี่ยนแปลงได้ดีเท่าที่ควร โดยที่ออฟเซ็ทคือค่าความแตกต่างของอินพุตและเอาต์พุตที่สภาวะคงที่เมื่อเป้าหมายคงที่

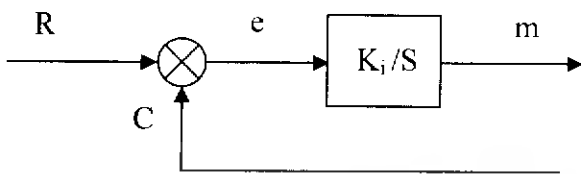


รูปที่ 2-4 แสดงผลตอบสนองแบบพี

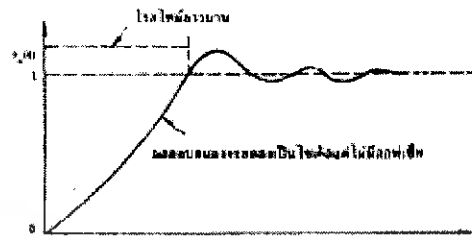
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 อินทิกรัลคอนโทรล (ไอ - คอนโทรล)

การคอนโทรลแบบอินทิกรัล ค่าสัญญาณควบคุม (m) จะเป็นสัดส่วนโดยตรงกับค่าอินทิกรัลของสัญญาณเออร์เรอร์ จึงสามารถแก้ค่าออฟเซ็ทได้ แสดงดังรูปที่ 2-5 และ 2-6



รูปที่ 2-5 ตัวคอนโทรลเลอร์แบบอินทิกรัล



รูปที่ 2-6 แสดงผลตอบสนองแบบไอ

ดังนั้นสมการของ m หาได้ดังนี้

$$m = (1/T_i)(e/s) + m(0) = K_i e/s + m(0) \tag{2}$$

เมื่อ K_i คือค่าคงที่มีหน่วย (%/วินาที/%)

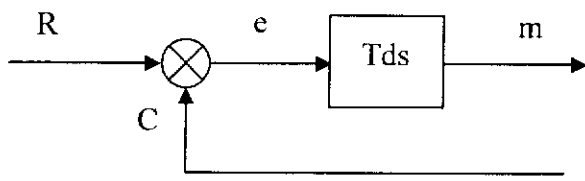
$T_i = 1/K_i$ มีหน่วยเป็น (วินาที)

T_i (อินทิกรัลไทม์) คือ เวลาที่ใช้ในการเพิ่มค่าของสัญญาณควบคุม จนกว่าสัญญาณผลต่างหมดไป

ในการควบคุมแบบอินทิกรัลนั้นค่าเอาต์พุทของตัวควบคุม m จะเปลี่ยนแปลงตามค่าความผิดพลาด ดังนั้น ถ้าความผิดพลาดซึ่งได้เกิดขึ้น ทำให้ระบบได้ค่าที่ผิดไปจากค่าที่ต้องการแล้วอุปกรณ์ควบคุมจะจัดการกับค่าความผิดพลาดโดยเร็ว เมื่อตัวแปรควบคุมอยู่ที่ค่าเป้าหมายแล้วอุปกรณ์ควบคุมสุดท้ายจะยังไม่ทำงาน ซึ่งแสดงให้เห็นว่า ระบบอยู่ในสภาวะคงที่แล้้งนั่นเอง

2.1.3.3 ดิริเวทิฟคอนโทรล (ดี - คอนโทรล)

การควบคุมแบบนี้สามารถแก้ค่าผิดพลาดโดยการกระทำล่วงหน้า ก่อนที่จะมีความผิดพลาดเกิดขึ้น สัญญาณควบคุมจะแปรตามอัตราการเปลี่ยนแปลงของสัญญาณเออร์เรอร์ การคอนโทรลแบบ ดิริเวทิฟนี้ไม่สามารถนำไปใช้งานแบบโดดเดี่ยวได้ เพราะว่าเมื่อเออร์เรอร์เป็นศูนย์หรือมีค่าคงที่คอนโทรลเลอร์จะไม่ให้สัญญาณเอาต์พุทเลย แสดงดังรูปที่ 2-7



รูปที่ 2-7 ตัวคอนโทรลเลอร์แบบดิริเวทิฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

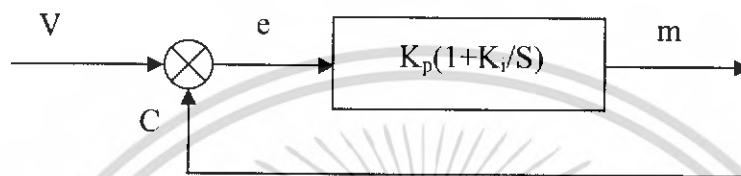
ดังนั้นสมการของ m หาได้ดังนี้

$$m = T_d \frac{de}{dt} + m(0) \quad (3)$$

เมื่อ T_d = กำลังขยายคงที่ของดิริเวทีฟคอนโทรล (มีหน่วยเป็นนาที)

2.1.3.4 ปรีอพออร์ชันแนล-อินติกรัลคอนโทรล

เป็นการควบคุมที่ค่าเอาต์พุต เป็นสัดส่วนแบบเชิงเส้นกับผลรวมของค่าอินพุตและค่าอินติกรัลเชิงเวลาของอินพุต แสดงดังรูปที่ 2-8



รูปที่ 2-8 ตัวคอนโทรลเลอร์แบบปรีอพออร์ชันแนล-อินติกรัล

ดังนั้นสมการของ m หาได้ดังนี้

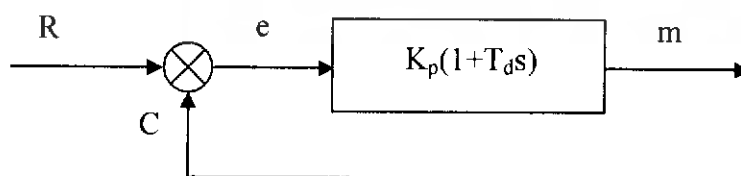
$$m = K_p e + K_p K_i \int e dt + m(0) \quad (4)$$

ข้อดีของ PI คอนโทรล คือ การคอนโทรลแบบนี้จะให้คุณสมบัติของทั้งแบบปรีอพออร์ชันแนล และ แบบอินติกรัลซึ่งสามารถทำให้ค่าออฟเซ็ทในแบบปรีอพออร์ชันแนลหมดไป และ K_i กำลังขยายของ การอินติเกรตสามารถจะปรับค่าได้อย่างอิสระ

ข้อเสียของ PI คอนโทรล คือระบบอาจจะไม่เสถียรภาพกรณีค่า T_i น้อยและไม่เหมาะสมกับกระบวนการ ที่มีไทม์แล็กมากๆ เพราะการตอบสนองของตัวแปรกระบวนการ (c) ช้ามากไม่สามารถ แก้ไข ข้อผิดพลาดได้ทันเวลา

2.1.3.5 ปรีอพออร์ชันแนล-ดิริเวทีฟคอนโทรล

เป็นการควบคุมซึ่งค่าเอาต์พุตเป็นสัดส่วน โดยตรงกับผลรวมของค่าอินพุตกับผลคูณค่าเวลากับอัตราการเปลี่ยนแปลงอินพุต แสดงดังรูปที่ 2-9



รูปที่ 2-9 ตัวคอนโทรลเลอร์แบบปรีอพออร์ชันแนล-อินติกรัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

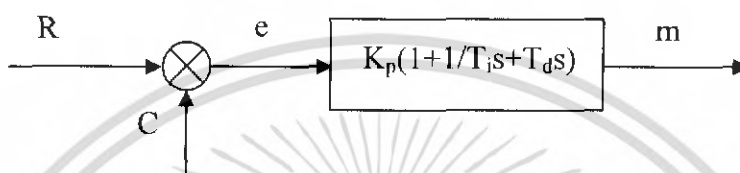
ดังนั้นสมการของ m หาได้ดังนี้

$$m = K_p e + K_p T_d (de/dt) + m(0) \quad (5)$$

PD คอนโทรล การคอนโทรลแบบนี้ไม่สามารถแก้ไขข้อผิดพลาดของปร็อบพอร์ชันแนลคอนโทรลเลอร์ได้ แต่สามารถควบคุมระบบที่มีโพลเปลี่ยนแปลงรวดเร็วได้

2.1.3.6 ปร็อบพอร์ชันแนลอินทิกรัลดิริเวทีฟคอนโทรล

พีไอดี-คอนโทรล คือการรวมระหว่างการควบคุมแบบ พี-คอนโทรล ไอ-คอนโทรล และ ดี-คอนโทรล แสดงดังรูปที่ 2-10

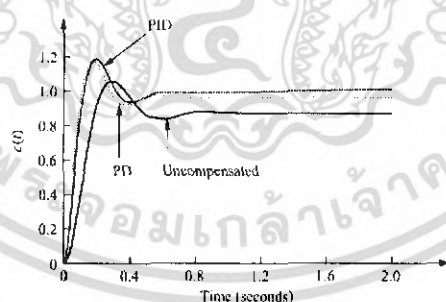


รูปที่ 2-10 ตัวคอนโทรลเลอร์แบบปร็อบพอร์ชันแนลอินทิกรัลดิริเวทีฟ

ดังนั้นสมการของ m หาได้ดังนี้

$$m = K_p/T_i \int e dt + K_p e + K_p T_d de/dt + m(0) \quad (6)$$

การคอนโทรลแบบ พีไอดีสามารถแก้ข้อผิดพลาดของปร็อบพอร์ชันแนลคอนโทรล และลด โอเวอร์ชูทที่จะทำให้เกิดการออสซิลเลท เนื่องจากอินทิกรัลคอนโทรล และจะให้ผลตอบสนองได้รวดเร็ว ตามคุณสมบัติของดิริเวทีฟคอนโทรล ผลตอบสนองที่ได้จากการคอนโทรลแบบพีไอดีแสดงดังรูปที่ 2-11



รูปที่ 2-11 แสดงผลตอบสนองแบบพีไอดี

2.2 องค์ประกอบของระบบเซอร์โว

จุดมุ่งหมายในบทนี้เป็นการอธิบายถึงส่วนประกอบต่างๆ ของระบบเซอร์โว เช่น ดีซีเซอร์โว มอเตอร์ (ดีซีมอเตอร์แบบต่างๆ) เอซีเซอร์โวมอเตอร์ (อินดิคชันมอเตอร์แบบ 2 เฟส) สเตปปีง มอเตอร์แบบต่างๆ และพวกส่วนประกอบประเภทเชิงกลที่ใช้กันทั่วไป ในระบบการคอนโทรลที่มี การป้อนกลับ เช่น ระบบลิฟต์ ระบบเฟือง ระบบสายพาน และระบบเกียร์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 หลักการทำงานของดีซีมอเตอร์

ดีซีมอเตอร์เป็นทรานส์ดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษ คือ แรงบิดของเพลลาของดีซีมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์มาเจอร์ แรงบิดของเพลลาของดีซีมอเตอร์จะได้จากผลระหว่างสนามแม่เหล็กและขดลวดตัวนำ หลักการในที่นี้กระแสที่ไหลในขดลวดตัวนำจะสร้างฟิลด์ที่ประกอบด้วยเส้นแรงแม่เหล็ก ϕ และขดลวดตัวนำเหล่านั้นอยู่ห่างจากศูนย์กลางการหมุนเท่ากับ r ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสเท่ากับ

$$T = K\phi I \quad (2-1)$$

เมื่อ T คือแรงบิดของเพลลา มีหน่วยเป็นนิวตัน-เมตร

ϕ คือเส้นแรงแม่เหล็ก มีหน่วยเป็นเวเบอร์

I คือกระแส มีหน่วยเป็นแอมแปร์

และ K คือตัวที่คงที่ ดังนั้นแรงบิดของเพลลาจะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแสเมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิดโวลต์เตจตกคร่อมตัวมันเอง โวลต์เตจนี้จะเป็นสัดส่วนกับความเร็วของเพลลาของมอเตอร์และด้านทางการไหลของกระแส ความสัมพันธ์ระหว่างโวลต์เตจย้อนกลับนี้และความเร็วของเพลลาของมอเตอร์คือ

$$E = K\phi\omega \quad (2-2)$$

เมื่อ E คือโวลต์เตจย้อนกลับ emf มีหน่วยเป็นโวลต์

θ คือเส้นแรงแม่เหล็ก มีหน่วยเป็นเวเบอร์

ω คือความเร็วของมอเตอร์ มีหน่วยเป็นเรเดียน/วินาที

สมการ (2-1) และ (2-2) เป็นสมการที่แสดงถึงหลักการทำงานพื้นฐานของดีซีมอเตอร์

2.2.2 การแยกประเภทของดีซีมอเตอร์

ดีซีมอเตอร์สามารถแบ่งออกได้เป็นหลายประเภทขึ้นอยู่กับลักษณะวิธีการสร้างสนามแม่เหล็กของตัวมอเตอร์และขึ้นอยู่กับพื้นฐานการออกแบบ โครงสร้างของอาร์มาเจอร์ การแบ่งประเภทตามลักษณะการจ่ายสนามแม่เหล็กแยกออกได้เป็น 2 แบบคือ

1. ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้
2. ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กมีค่าคงที่

ถ้าเราจะพิจารณาแยกประเภทตามลักษณะการออกแบบโครงสร้างอาร์มาเจอร์สามารถแยกออกได้เป็น 3 แบบคือ

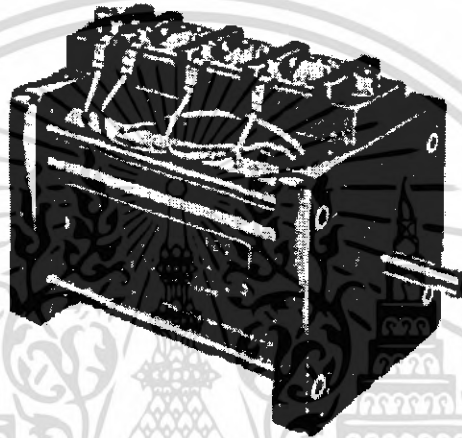
1. ดีซีมอเตอร์แบบอาร์มาเจอร์เป็นแกนเหล็ก
2. ดีซีมอเตอร์แบบอาร์มาเจอร์ที่มีขดลวดพันอยู่บนพื้นผิว

3. ดีซีมอเตอร์แบบอาร์มาเจอร์เป็นขดลวดหมุน

นอกจากนี้ยังมีดีซีมอเตอร์ชนิดพิเศษอีกแบบหนึ่งคือ แบบไม่มีแปรงถ่าน (brushless dc motor) ซึ่งมีหลักการทางเทคโนโลยีเหมือนกับดีซีมอเตอร์ชนิดมีแปรงถ่านยกเว้น การเชื่อมมิวเทชั่นกระทำโดยทางเทคนิคทางอิเล็กทรอนิกส์แทนที่จะกระทำโดยวิธีการทางเชิงกล ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้

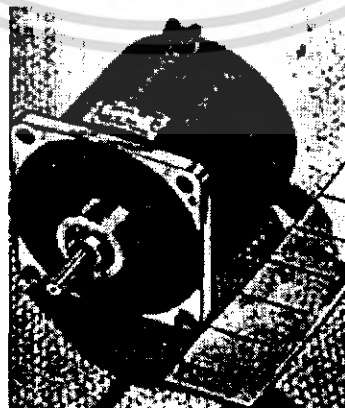
ดีซีมอเตอร์แบบปรับเส้นแรงแม่เหล็กได้ยังแบ่งแยกได้เป็น 2 แบบคือ

- ก) แบบขดลวดสนามแม่เหล็กต่ออนุกรมกับขดลวดอาร์มาเจอร์
- ข) แบบขดลวดสนามแม่เหล็กแยกกระตุ้น



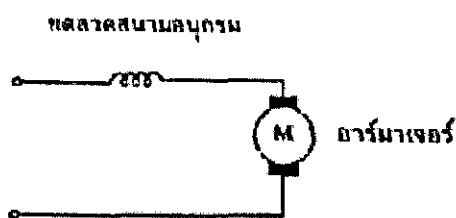
รูปที่ 2-12 เซอร์โวมอเตอร์แบบโรเตอร์เป็นแกนเหล็ก

ตัวอย่างของมอเตอร์แบบขดลวดสนามแม่เหล็กต่ออนุกรมแสดงดังในรูปที่ 2-14 มอเตอร์แบบนี้จะมีเส้นแรงแม่เหล็กเป็นสัดส่วนกับกระแส ดังนั้นเส้นแรงของสนามแม่เหล็กจึงสามารถปรับค่าได้ และเราจะได้ความสัมพันธ์ระหว่างความเร็วและแรงบิดเป็นนอนลิเนียร์ ดังแสดงในรูป 2-15 มอเตอร์ดังกล่าวจะใช้งานในภาวะเฉพาะเมื่อต้องการแรงบิดสูงที่ความเร็วต่ำ และแรงบิดต่ำที่ความเร็วสูงเช่นระบบการขับเคลื่อนของรถลาก

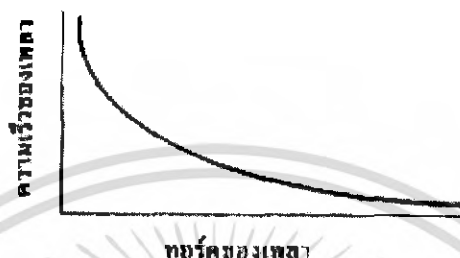


รูปที่ 2-13 เซอร์โวมอเตอร์แบบโรเตอร์เป็นรูปถ้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

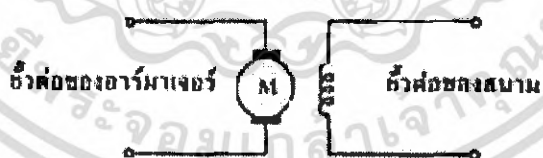


รูปที่ 2-14 ดีไซน์มอเตอร์แบบอาร์มาเจอร์ต่ออนุกรมกับขดลวดสนามแม่เหล็ก

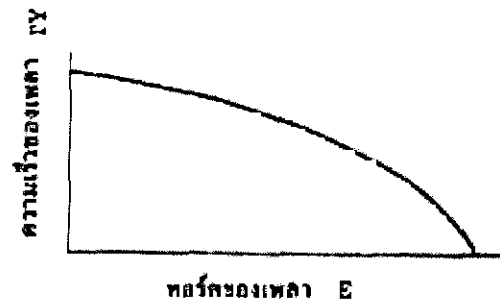


รูปที่ 2-15 คุณสมบัติระหว่างความเร็วและแรงบิดของดีไซน์มอเตอร์อนุกรมภายใต้ภาวะโวลต์เตจคงที่

ตัวอย่างของมอเตอร์แบบขดลวดสนามแม่เหล็กแยกกระตุ้นแสดงในรูปที่ 2-16 ดีไซน์มอเตอร์แบบนี้มักนิยมเรียกกันว่า มอเตอร์ชานาน (shunt motor) มอเตอร์แบบนี้สามารถปรับเส้นแรงแม่เหล็กได้อย่างอิสระต่อกระแสของอาร์มาเจอร์ ยังผลให้สามารถควบคุมพารามิเตอร์ของมอเตอร์ให้มีค่าคงที่ได้ตลอดช่วงพิสัยที่กว้าง มอเตอร์นี้มักจะใช้งานในกรณีระบบบังคับการเคลื่อนที่ที่ต้องการแรงบิดสูงใน รูปที่ 2-17 แสดงถึงคุณสมบัติระหว่างแรงบิดกับความเร็วของชั้นท์มอเตอร์ภายใต้ภาวะการกระตุ้นสนามแม่เหล็กคงที่และอาร์มาเจอร์โวลต์เตจคงที่



รูปที่ 2-16 ดีไซน์มอเตอร์แบบแยกปรับสนามแม่เหล็กได้



รูปที่ 2-17 แสดงคุณสมบัติระหว่างความเร็วและแรงบิดของซิงโครมอเตอร์ภายใต้ภาวะอาร์มาเจอร์โวลต์เตจคงที่ และการกระตุ้นสนามแม่เหล็กคงที่

ดีซีมอเตอร์แบบเส้นแรงแม่เหล็กคงที่

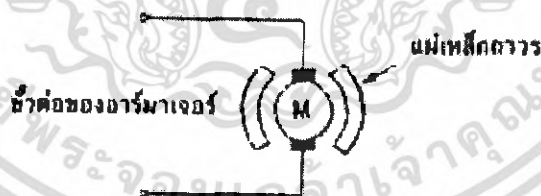
ระบบการกระตุ้นฟลักซ์ของมอเตอร์โดยทั่วไปในปัจจุบันมักใช้เป็นแบบ แม่เหล็กถาวร ดังแสดงในรูปที่ 2-18 ในระบบนี้เส้นแรงของฟลักซ์มีค่าคงที่ ดังนั้น อัตราส่วนระหว่างกระแสมาเจอร์ และแรงบิดจะมีค่าคงที่ ดังนั้นสมการ (2-1) เขียนให้ง่ายขึ้นเป็น

$$T = K_f I \quad (2-3)$$

และสมการ (2-2) ได้เป็น

$$E = \omega \quad (2-4)$$

ระบบนี้จะให้ความสัมพันธ์ระหว่าง กระแสอาร์มาเจอร์ แรงบิดและความเร็วอยู่ในลักษณะลิเนียร์ สมการทางไฟฟ้าของดีซีมอเตอร์แบบนี้เขียนได้เป็น



รูปที่ 2-18 ดีซีมอเตอร์แบบฟลักซ์เป็นแม่เหล็กถาวร

$$V = K_e \omega + L(di/dt) + Ri \quad (2-5)$$

เมื่อ V คือ โวลต์เตจที่ป้อนให้กับมอเตอร์

K_e คือ ค่าคงที่ของโวลต์เตจย้อนกลับ

L คือ อินдукแตนซ์ของอาร์มาเจอร์

R คือ ความต้านทานที่ขั้วของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการไดนามิกของมอเตอร์คือ

$$T_g = J(d\omega/dt) + B\omega + T_f + T_L \quad (2-6)$$

- T_g คือแรงบิดที่กำเนิดโดยมอเตอร์
- J คือผลรวมของโมเมนต์ของแรงเฉื่อยของมอเตอร์และโหลด
- B คือสัมประสิทธิ์ของวิสกอสแด็มพ์ปีง
- T_f คือแรงบิดเสียดทาน
- T_L คือแรงบิดโหลด

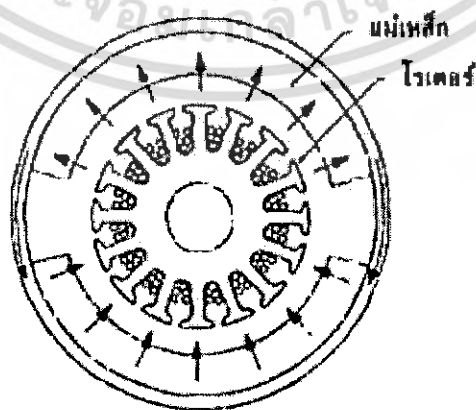
สมการต่างๆของมอเตอร์แบบแยกกระตุ้นผลัดจะเหมือนกับแบบกระตุ้นฟิลด์คงที่ อย่างไรก็ตาม ข้อดีของมอเตอร์แบบฟิลด์แม่เหล็กถาวรซึ่งเหนือกว่ามอเตอร์แบบมีม โครงสร้างของฟิลด์ด้วยการพันของขดลวดคือ ไม่มีกำลังสูญเสียในฟิลด์ มีประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่าเมื่อเทียบกับมอเตอร์ที่มีขนาดของกำลังน้ำเท่ากันนอกจากนั้นความสัมพันธ์เชิงเส้นในสมการ (2-1) ยังให้ค่าของกระแสอาร์มาเจอร์ที่สูงกว่าดีซีมอเตอร์แบบฟิลด์เป็นขดลวด การประยุกต์ใช้งานเหมาะสมกับระบบที่ต้องการแรงบิดของโหลดสูง

ดีซีมอเตอร์แบบอาร์มาเจอร์เป็นแกนเหล็ก

โครงสร้างของโรเตอร์และสเตเตอร์ของมอเตอร์แบบแกนเหล็กแสดงดังในรูปที่ 2-19 โครงสร้างของมอเตอร์แบบนี้มีโมเมนต์ของแรงเฉื่อยสูงสุด และมีค่าอินดักแตนซ์ของโรเตอร์สูงสุดด้วย ดังนั้นมอเตอร์นี้จึงมีปริมาณการจุกความร้อนได้สูง และสามารถจะทน โอเวอร์โหลดได้ในระยะเวลาที่ยาวนานโดยไม่ทำให้มอเตอร์เสียหาย

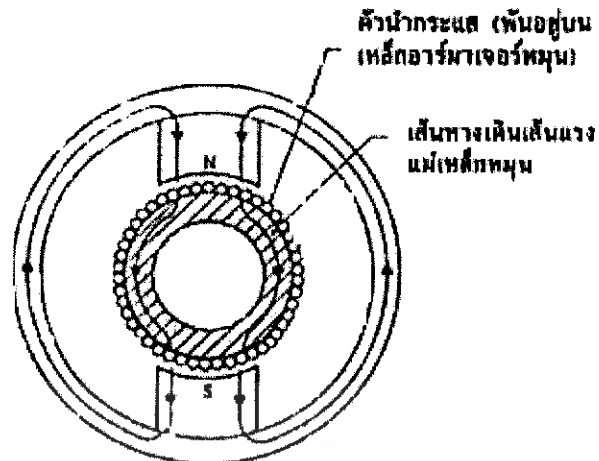
ดีซีมอเตอร์แบบอาร์มาเจอร์มีขดลวดพันอยู่บนพื้นผิว

ในรูปที่ 2-20 แสดงถึงการออกแบบของโรเตอร์ที่มีขดลวดพันอยู่บนพื้นผิว โดยไม่มีสล๊อท ทำให้ได้อินดักแตนซ์ของโรเตอร์ต่ำกว่าแบบแกนเหล็ก ข้อเสียคือ ทำให้ขนาดของมอเตอร์แบบนี้ใหญ่ขึ้นและราคาแพงกว่าแบบแกนเหล็กด้วย



รูปที่ 2-19 แสดงรูปหน้าตัดของดีซีมอเตอร์แบบอาร์มาเจอร์เป็นแกนเหล็กส่วนฟิลด์เป็นแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-20 แสดงรูปหน้าตัดของคีมอเตอร์แบบมีขลวดบนพื้นผิวและฟิลด์เป็นแม่เหล็กถาวร

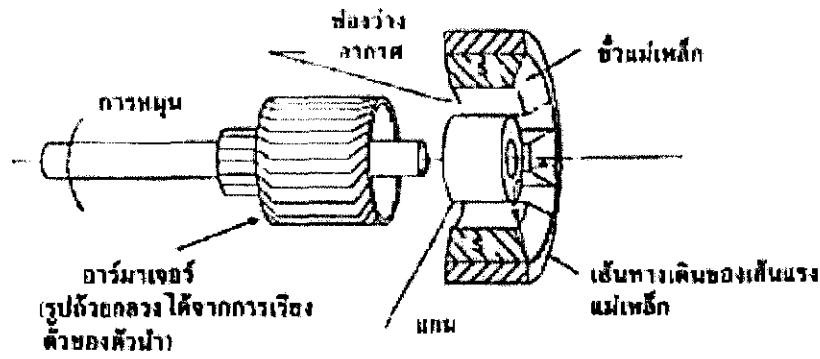
คีมอเตอร์แบบอาร์มาเจอร์เป็นขลวดหมุน

มอเตอร์แบบขลวดหมุนนี้ได้รับการออกแบบเพื่อให้มีโมเมนต์ของแรงเฉื่อยน้อยมาก ดังแสดงในรูปที่ 2-21 และรูปหน้าตัดซึ่งมองด้านข้างของมอเตอร์ดังกล่าวแสดงในรูปที่ 2-22 นอกจากนี้โครงสร้างของโรเตอร์มีความจุความร้อนต่ำมากถ้าหากเกิดการ โอเวอร์ โหลดก็จะทำให้มอเตอร์เสียหายได้ง่าย ขณะโรเตอร์ลักษณะนี้จะมีค่าอินดักแตนซ์ต่ำมากคือน้อยกว่า 10 ไมโครเฮนรี่



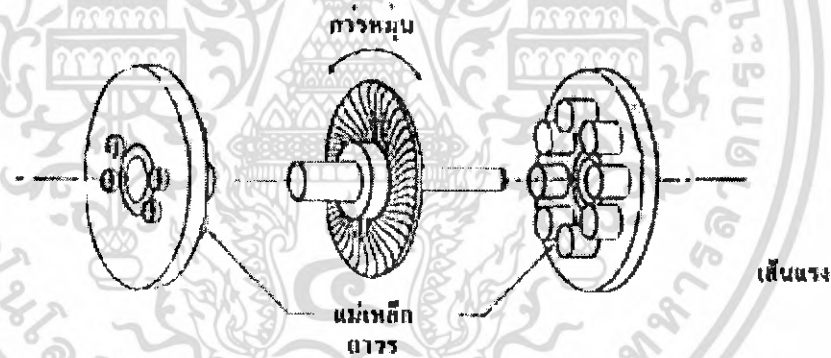
รูปที่ 2-21 หน้าตัดของคีมอเตอร์แบบมีโรเตอร์เป็นขลวดเคลื่อนที่และฟิลด์เป็นแม่เหล็กถาวร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-22 แสดงถึงรูปหน้าตัดซึ่งมองด้านข้างของดีซีมอเตอร์แบบมีโรเตอร์เป็นขดลวดหมุน

ในรูปที่ 2-23 แสดงถึงมอเตอร์ประเภทขดลวดเคลื่อนที่อีกลักษณะหนึ่งซึ่งมีโครงสร้างของ อาร์มาเจอร์เป็นรูปร่างจานซึ่งทำขึ้นจากขดลวดตัวนำซ้อนกันหลายๆชั้น ซึ่งเรามักจะเรียกกันว่า "printed motor" ในปัจจุบันดีซีมอเตอร์แบบโรเตอร์เป็นขดลวดหมุนนี้ให้คุณสมบัติการทำงานที่ดีเยี่ยมเหมาะสำหรับเป็นตัวขับเคลื่อนในระบบการบังคับตำแหน่ง และยังให้อัตราส่วนระหว่างแรงบิดและแรงเฉื่อยได้สูงและมีค่าอินдукแตนซ์ต่ำที่สุดเมื่อเทียบกับมอเตอร์แบบอื่นๆ นอกจากนั้นความสามารถในการเพิ่มอัตราเร่งยังกระทำได้สูงถึง 10^6 เรเดียน/วินาที²



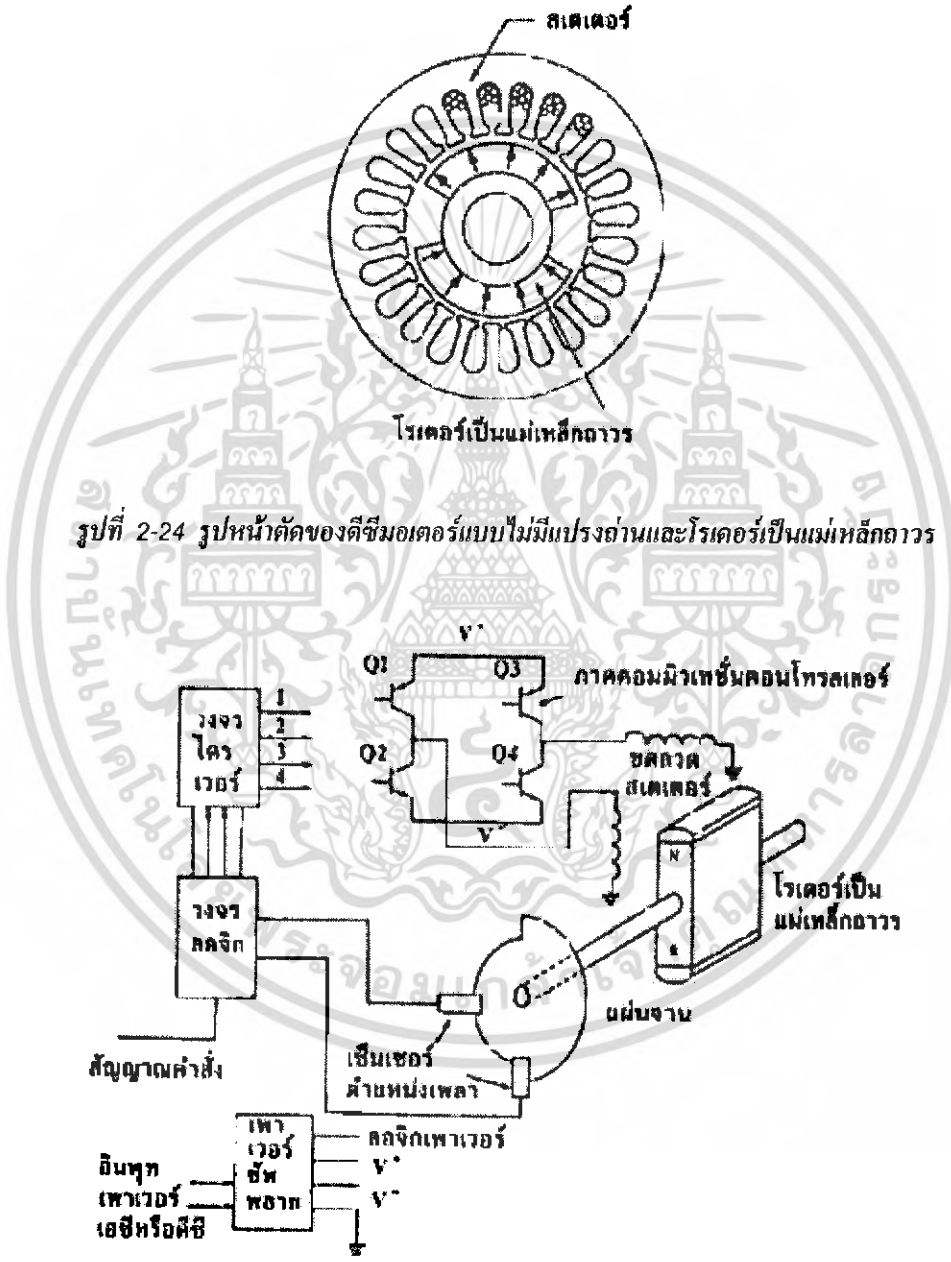
รูปที่ 2-23 แสดงถึงดีซีมอเตอร์แบบโรเตอร์เป็นขดลวดหมุนมีรูปร่างเป็นจาน

ดีซีมอเตอร์แบบไม่มีแปรงถ่าน

ดีซีมอเตอร์แบบไม่มีแปรงถ่านต่างจากดีซีมอเตอร์ที่กล่าวมาแล้วคือ การค้อมมิวเทชั่นกระแส อาร์มาเจอร์จะใช้วิธีการทางอิเล็กทรอนิกส์ โครงสร้างของมอเตอร์แบบนี้จะมีโรเตอร์เป็นแม่เหล็กหุ้ม "back iron" และขดลวดค้อมมิวเทจจะอยู่ภายนอกส่วนของตัวโรเตอร์ดังแสดงในรูป 2-24 (เปรียบเทียบกับดีซีมอเตอร์โดยทั่วไปแล้วจะมีโครงสร้างที่มีส่วนประกอบกลับกัน) การประยุกต์ใช้งานของดีซีมอเตอร์แบบไม่มีแปรงถ่านนี้มักจะใช้ในระบบที่ต้องการโมเมนต์แรงเฉื่อยต่ำ เนื่องจากโครงสร้างโรเตอร์ของ มอเตอร์แบบนี้สร้างขึ้นด้วยสารแม่เหล็กชนิดพิเศษเพื่อให้เป็นมอเตอร์ชนิดที่มีโมเมนต์ของแรงเฉื่อยต่ำ

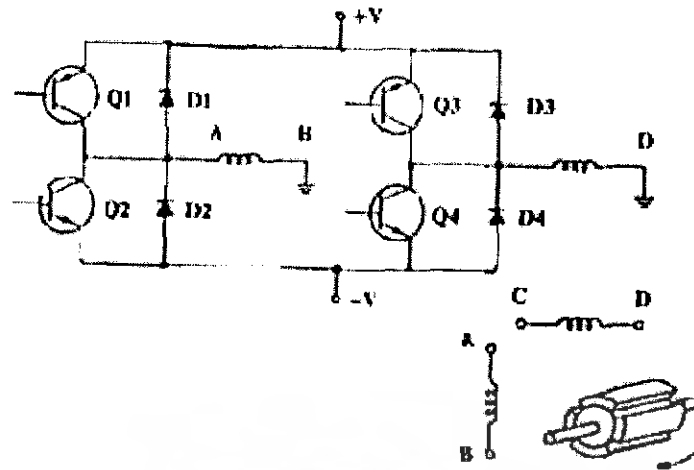
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดคอมมิวเทชันในมอเตอร์แบบไม่มีแปรงถ่านจะกระทำใ้ภายนอกตัวมอเตอร์ โดยใช้วงจรสวิตช์ซึ่งรวมเข้ากับมอเตอร์เพื่อให้เข้าใจการบังคับมอเตอร์แบบไม่มีแปรงถ่านดียิ่งขึ้น รูปที่ 2-25 จะแสดงถึงบล็อกไดอะแกรมของส่วนประกอบที่สำคัญในระบบการบังคับดีซีมอเตอร์แบบไม่มีแปรงถ่าน เอนโคเดอร์ที่เพลลาของมอเตอร์แสดงให้เห็นว่าจำเป็นต้องซึ่งใคร ในชั้คอมมิวเทชันสวิตช์ซึ่งกับตำแหน่งเชิงมุมของโรเตอร์ ในรูปที่ 2-26 และรูป 2-27 แสดงถึงโครงสร้างของตัวบังคับอีกแบบหนึ่ง ซึ่งเรียกว่า อินเวอร์เตอร์ซึ่งมักจะใช้กับดีซีมอเตอร์แบบไม่มีแปรงถ่าน



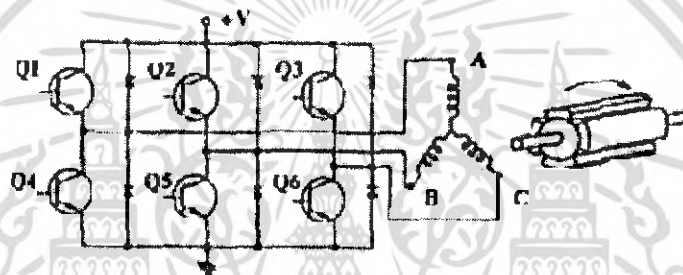
รูปที่ 2-25 แสดงถึงส่วนประกอบที่สำคัญในระบบการบังคับดีซีมอเตอร์แบบไม่มีแปรงถ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-26 ตัวอย่างโครงสร้างอินเวอร์เตอร์ของดีซีมอเตอร์แบบไม่มีแปรงถ่านชนิด 2 เฟส โดยใช้เพาเวอร์ซัพ

หลายสองค่า



รูปที่ 2-27 ตัวอย่างอินเวอร์ทของดีซีมอเตอร์ของดีซีมอเตอร์แบบไม่มีแปรงถ่านชนิด 3 เฟส โดยใช้เพาเวอร์ซัพ

หลายเพียงค่าเดียว

2.2.3 โมเดลคณิตศาสตร์ของดีซีมอเตอร์

ดีซีมอเตอร์ที่ใช้ร่วมกับดีซีแอมพลิไฟ์ทั้งในระบบการบังคับตำแหน่ง และการบังคับความเร็ว มักจะได้รับการประยุกต์ใช้เป็นส่วนประกอบสร้างกำลังงานในระบบการนำร่องและระบบการบังคับต่างๆ และเนื่องจากวิทยาการเกี่ยวกับสารแม่เหล็กถาวรได้รับความนิยมใช้เป็นส่วนประกอบการขับเคลื่อนในระบบการบังคับแบบปิดูปต่างๆมากขึ้น การออกแบบและการชดเชยระบบดังกล่าวได้อย่างเหมาะสมจะต้องใช้โมเดลคณิตศาสตร์ของส่วนประกอบทั้งหมดในระบบ ในหัวข้อนี้เราจะได้พัฒนาโมเดลของดีซีมอเตอร์แบบแม่เหล็กถาวรและแบบฟิลด์แยกกระตุ้น

2.2.3.1 โมเดลอิเล็กทรอนิกส์โทรมแมคทานิคอล

ส่วนสำคัญของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้นมีโมเดลดังแสดงในรูป 2-28

R_a : ความต้านทานของอาร์มาเจอร์

L_a : อินดักเตนซ์อาร์มาเจอร์

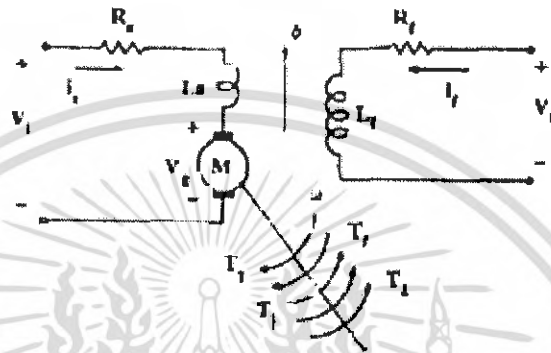
V_g : โวลต์เตจกำเนิดในอาร์มาเจอร์ (โวลต์เตจย้อนกลับ)

R_f : ความต้านทานของฟิลด์

L_f : อินดักเตนซ์ของฟิลด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อ **62890** ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ϕ : ช่องว่างอากาศของเส้นแรงสนามแม่เหล็ก
 ω : ความเร็วของเพลารี่มาเจอร์
 T_g : แรงบิดที่พัฒนาขึ้นในมอเตอร์
 T_r : แรงบิดเสียดทานของมอเตอร์
 T_j : แรงเฉื่อยของมอเตอร์
 T_L : แรงบิดโหลดบนเพลารี่มาเจอร์



รูป 2-28 โมเดลของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้น

ขั้นแรกเราจะหาสมการพื้นฐาน โมเดลของดีซีมอเตอร์ได้จากกฎของอาร์มาเจอร์

$$V_a(t) = R_a i_a + L_a [di_a(t)/dt] + V_g(t) \quad (2-7)$$

เทอมโวลต์เตจ ในสมการ (2-7) คือโวลต์เตจย้อนกลับของมอเตอร์ซึ่งเกิดขึ้นเมื่อเส้นลวดตัวนำของ อาร์มาเจอร์ หมุนตัดเส้นแรงแม่เหล็กซึ่งเกิดขึ้นโดยกระแสของฟิลด์ (i_f) ตามกฎของฟาราเดย์ของ เส้นลวดตัวนำหมุนในฟิลด์แม่เหล็กคงที่จะมีการเหนี่ยวนำโวลต์เตจขึ้นในขดลวดนั้น

$$v(t) = d\lambda(t)/dt \quad (2-8)$$

เมื่อ $\lambda(t)$ เส้นแรงแม่เหล็กที่วิ่งไปยังขดลวดและ t คือเวลาในการหมุนของคีมนิวเทเดอของมอเตอร์ การควบคุมวงจรของแต่ละส่วนของตัวนำในโรเตอร์จะเกิดโวลต์เตจขึ้นในส่วนของตัวนำนั้นตามสมการ (2-8) เมื่อ $d\lambda(t)/dt$ จะเป็นสัดส่วนต่อเส้นแรงแม่เหล็กในช่องว่างอากาศและ ความเร็วเชิงมุม $\omega(t)$ หรือเราจะได้ว่า

$$V_g(t) = K\phi(t) \omega(t) \quad (2-9)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมติให้กระแสของฟลักซ์มีค่าคงที่ละ ไม่คิดถึงส่วนการเปลี่ยนแปลงในเส้นแรงฟลักซ์เนื่องจากอาร์มาเจอร์รีแอคชั่นเส้นแรงฟลักซ์ก็จะมีค่าคงที่ดังนั้นสมการ (2-9) ก็จะกลายเป็น

$$V_g(t) = K_e \omega(t) \quad (2-10)$$

เมื่อเราสมมติให้เส้นแรงของฟลักซ์มีค่าคงที่ แรงบิดของแม่เหล็กไฟฟ้าซึ่งเกิดขึ้นแก่โรเตอร์ของมอเตอร์จะเป็นสัดส่วนกับกระแสอาร์มาเจอร์

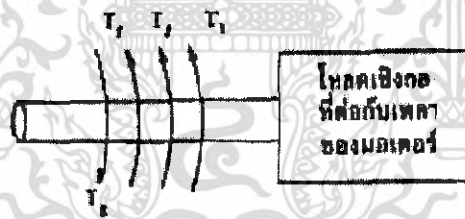
$$T_g(t) = K_t i_a(t) \quad (2-11)$$

เมื่อ K_t คือค่าคงที่ของแรงบิดของมอเตอร์

กำลังงานเชิงกลที่เกิดขึ้นใน โฉมเตอร์คือผลคูณของแรงบิดที่เกิดขึ้นและความเร็วเชิงมุม

$$P_g(t) = T_g(t) \omega(t) \quad (2-12)$$

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์ทั้งหมดนี้จะจ่ายไปยังโหลดที่ต่ออยู่กับเพลามอเตอร์แต่กำลังงานนี้บางส่วนจะสูญเสียไปในมอเตอร์ การสูญเสียเนื่องมาจากแรงเสียดทาน หมายถึงความหน่วงเนื่องจากลมที่มีต่อโรเตอร์ แรงเสียดทานตัวรองรับ โรเตอร์ กระแสที่ไหลวนในเหล็กของโรเตอร์และอิฐเทรีซิส จาก รูป 2-29 แสดงให้เห็นถึงแรงบิดต่างๆ ที่เกิดขึ้นต่อโหลดของมอเตอร์ คือ



รูปที่ 2-29 แสดงถึงแรงบิดต่างๆที่เกิดขึ้นต่อโหลดของมอเตอร์

- $T_g(t)$: แรงบิดของมอเตอร์
- $T_r(t)$: แรงบิดที่ต้องชนะการสูญเสียเนื่องจากแรงเสียดทาน
- $T_j(t)$: แรงบิดเพื่อเพิ่มอัตราเร่งแก่ความเฉื่อยของโหลด
- $T_L(t)$: แรงบิดโหลด

ในช่วงเวลาใดๆก็ตาม แรงบิดของมอเตอร์จะต้องเท่ากันและมีทิศทางตรงข้ามกับผลรวมของแรงบิด $T_r(t)$ $T_j(t)$ และ $T_L(t)$ ดังนั้น

$$T_g(t) = T_r(t) + T_L(t) + J(d\omega(t)/dt) \quad (2-13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ J คือผลรวมของโมเมนต์แรงเฉื่อยของโรเตอร์และโหลดที่ต่ออยู่ที่เพลลาของมอเตอร์
ผลรวมของแรงบิดเสียดทานที่ประกอบกันขึ้นที่เพลลาของมอเตอร์ซึ่งเป็นลิเนียร์ของฟังก์ชันกับความเร็ว
เชิงมุมของโรเตอร์เรียกว่า ส่วนประกอบเชิงวิศกีสฟริกชันและมักจะอยู่ในเทอมที่แยกออกจากฟริกชันอื่นๆ
ซึ่งแสดงได้ด้วยสมการต่อไปนี้

$$T_g(t) = T_r(t) + T_L(t) + J(d\omega(t)/dt) + B\omega(t) \quad (2-14)$$

เมื่อ B คือสัมประสิทธิ์ของวิสกอสฟริกชันของมอเตอร์และโหลดที่ต่ออยู่กับเพลลาของมอเตอร์ $T_r(t)$ คือ
ผลรวมของฟริกชันของโหลดและมอเตอร์ทั้งหมด มีแรงต้านของลมและการสูญเสียกำลังในเหล็กของ
เพลลามอเตอร์ยกเว้นฟริกชัน

สมการ (2-7), (2-10), (2-11) และ (2-14) เป็นจุดสมการพื้นฐานของดีซีมอเตอร์โมเดลและจากสมการ
เหล่านี้เราสามารถจะหาทรานสเฟอร์ฟังก์ชันของดีซีมอเตอร์ได้ โดยใส่ลาปลาซทรานสเฟอรั่มทั้งสองข้าง
ของจุดสมการพื้นฐานและเขียนได้ใหม่เป็น

$$V_1(s) - V_2(s) = (R_a + sL_a) I_a(s) \quad (2-15)$$

$$V_g(s) = K_g \omega(s) \quad (2-16)$$

$$T_g(s) = K_t I_a(s) \quad (2-17)$$

$$T_g(s) - T_r(s) - T_L(s) = (B + sJ) \omega(s) \quad (2-18)$$

บล็อกไดอะแกรมที่แสดงถึงสมการพื้นฐานเหล่านี้แสดงได้ดังในรูปที่ 2-30

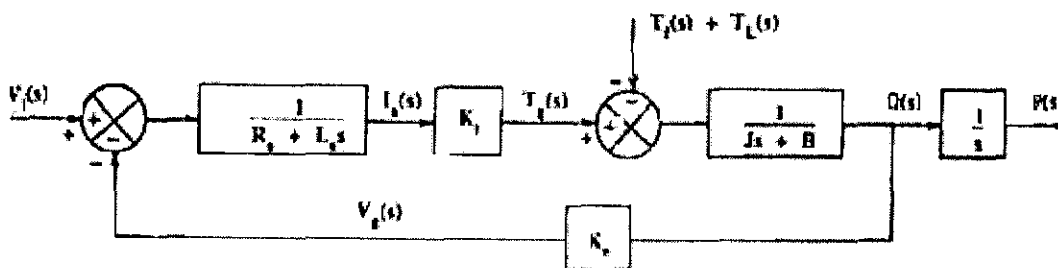
ข้อควรสังเกต

สมมติว่า โวลต์ที่ป้อนให้กับวงจรรออาร์มาเจอร์ของมอเตอร์มีค่าคงที่คั้งนั้นมอเตอร์จะหมุนด้วย
ความเร็วคงที่คือทำงานอยู่ที่สภาวะสงบนิ่งด้วยโหลดที่คงที่ กำลังงานเชิงกลที่เกิดขึ้นโดยโรเตอร์จะหาได้
จาก สมการ (2-12) และเมื่อรวมกับสมการ (2-11) ก็จะได้

$$P_g = T_g \omega = K_t I_a \omega \quad (2-19)$$

เมื่อทุกเทอมในสมการสุดท้ายมีค่าคงที่เนื่องจากมอเตอร์ทำงานอยู่ที่สภาวะสงบนิ่งกำลังไฟฟ้าที่ถูกดูดกลืน
โดยอาร์มาเจอร์ต้องเท่ากับผลคูณของโวลต์เต็มคร่อมอาร์มาเจอร์และกระแสที่ไหลผ่าน ดังนั้น

$$P = V_g I_a = K_g \omega I_a \quad (2-20)$$



รูป 2-30 บล็อกไดอะแกรมของดีซีมอเตอร์โวลต์

ดังนั้นเราจะได้ว่า กำลังงานเชิงกลที่เกิดขึ้นต้องเท่ากับกำลังงานไฟฟ้าที่ถูกดูดกลืนเข้าไปในโรเตอร์ สรุปได้ว่า $K_t = K_o$

2.2.3.2 ทรานสเฟอร์ฟังก์ชันของดีซีมอเตอร์

บล็อกไดอะแกรมของรูปที่ 2-30 แสดงถึงระบบที่มีสองอินพุต และมีเอาต์พุตเป็นทั้งความเร็วเชิงมุม ω และการเคลื่อนที่แบบเชิงมุม θ จากรูปที่ 2-30 ความเร็วเอาต์พุตของระบบเขียนได้เป็น

$$\omega(s) = G_1(s)V_f(s) + G_2(s) + [T_r(s) + T_L(s)] \quad (2-21)$$

เมื่อ

$$G_1(s) = \omega(s) / V_f(s) \quad (2-22)$$

$$G_2(s) = \omega(s) / (T_r(s) + T_L(s)) \quad (2-23)$$

$G_1(s)$ คือทรานสเฟอร์ฟังก์ชันระหว่างโวลต์ตั้งและความเร็ว

$$\begin{aligned} G_1(s) &= \omega(s) / V_f(s) = K_t / [(L_a s + R_a)(J s + B) + K_t K_o] \\ &= K_m / \alpha s^2 + \beta s + 1 \end{aligned} \quad (2-24)$$

เมื่อ

$$K_m = K_t / (R_a B + K_t K_o)$$

$$\alpha = L_a J / (R_a B + K_t K_o)$$

$$\beta = R_a J + L_a B / (R_a B + K_t K_o)$$

สมการ (2-24) เป็นโวลต์ตั้งทรานสเฟอร์ฟังก์ชันของดีซีมอเตอร์ในเมื่อสมมติว่า T_r และ T_L มีค่าเป็นศูนย์ สมการ (2-24) สามารถเขียนได้ใหม่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_1(s) = K_f / [R_a B(1+T_e s)(1+T_m s) + K_f K_e]$$

เมื่อ

$$T_e = L_a / R_a = \text{ไทม์คอนสแตนต์ทางไฟฟ้า}$$

$$T_m = J / B = \text{ไทม์คอนสแตนต์ทางเชิงกล}$$

ถ้าอินตักต์ของอาร์มาเจอร์ที่ค่าน้อย ไทม์คอนสแตนต์ทางไฟฟ้าสามารถตัดทิ้งได้และสมการ (2-24) ได้เป็น

$$\begin{aligned} G_v(s) = w(s) / V_f(s) &= K_f / [R_a (Js+B) + K_f K_e] \\ &= K_m / (Ts+1) \end{aligned} \quad (2-25)$$

เมื่อ

$$T = R_a J / (R_a B + K_f K_e)$$

ในสมการ (2-25) ค่าคงที่ K_m อาจเรียกได้ว่าเป็นค่าคงที่ของมอเตอร์ ทรานสเฟอร์ฟังก์ชันแรงบิดโหลด $G_2(s)$ หาได้เป็น

$$G_2(s) = w(s) / (T_f(s) + T_l(s)) = (-1 / (Js+B)) / [1 + (K_f K_e / (Js+B)(L_a s + R_a))] \quad (2-26)$$

ซึ่งถ้าอินตักต์ของอาร์มาเจอร์ไม่นำมาคิดถึง สมการ (2-26) ก็ลดลงได้

$$G_L(s) = w(s) / (T_f(s) + T_l(s)) = (-R_a K_m / K_f) / (Ts+1)$$

2.3 วิชาวลเบสิก

วิชาวลเบสิก (Visual basic) เป็นโปรแกรมภาษาที่ใช้ในการเขียน โปรแกรมบนระบบปฏิบัติการ วินโดวส์(Windows) ที่พัฒนาโดยบริษัทไมโครซอฟท์ ซึ่งที่รากฐานจากภาษาเบสิก(basic) วิชาวลเบสิก เป็นโปรแกรมภาษาที่มีวิธีการโปรแกรมในแบบ Graphic User Interface(GUI) ซึ่งแตกต่างจาก โปรแกรมคำสั่งแบบบรรทัดคำสั่ง ซึ่งจะต้องเขียนคำสั่งเพื่อกำหนดภาพลักษณ์ของสิ่งต่างๆ ด้วยการจับมา วางไว้บนจอภาพได้เลย

Event and Message

ระบบปฏิบัติการวินโดวส์จะประกอบด้วยส่วนต่างๆ เช่น ปุ่มกด ช่องใส่ข้อความ แถบเลือก เมนูบาร์ แถบเลื่อน ซึ่งส่วนต่างๆ นี้ก็เป็นวินโดวส์ทั้งสิ้น ระบบปฏิบัติการวินโดวส์มีวิธีการจัดการและควบคุมวินโดวส์ต่างๆเหล่านี้ด้วยการกำหนดหมายเลขไม่ซ้ำกันให้กับวินโดวส์แต่ละตัวเรียกว่า Windows handle(HAND) ระบบจะคอยเฝ้าดูอาการ(sign soft Activity) และเหตุการณ์(event) ที่เกิดขึ้นกับแต่ละวินโดวส์ เหตุการณ์ต่างๆ ที่เกิดขึ้นกับผู้ใช้(user) เลือกหรือคลิกปุ่มใดๆจากเมาส์(mouse) หรือ แป้นพิมพ์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(keyboard) ซึ่งก่อให้เกิดข่าวสาร(Massage) ส่งไปที่ระบบปฏิบัติการ(OS) เมื่อระบบปฏิบัติการ ได้รับข่าวสารก็จะประมวลผล(process) และกระจายข่าวสารให้กับวินโดวส์อื่นๆ แต่ละวินโดวส์ก็จะมีวิธีการที่เหมาะสมคอยรับข่าวสารเฉพาะสำหรับตัวเองเพื่อนำมาประมวลผลต่อไป

Event-Driver

การเขียนโปรแกรมในยุคแรกๆนั้น การทำงานของโปรแกรมนั้นจะมีการทำงานตามโค้ดของโปรแกรมไปเรื่อยๆ หากมีการเขียนโปรแกรมย่อย(procedure) โค้ดก็จะกระโดดไป procedure ฝั่งนั้นจนจบโปรแกรม

ในส่วนโปรแกรมแบบ Event-Driver นั้นการทำงานกับโค้ดโปรแกรมจะขึ้นอยู่กับเหตุการณ์(event) ที่มากระทำกับวัตถุ(object) หนึ่งๆ คือ หากมีการกระทำใดๆกับ Object ก็จะมีเหตุการณ์ต่างๆ เช่น Click, Double Click Mouse เป็นต้น จะเขียนโค้ดภายในเหตุการณ์(event) นั้นๆ เพื่อให้ โปรแกรมทำงานตามที่ต้องการเมื่อมีเหตุการณ์นั้นๆเกิดขึ้นกับวัตถุ (object) ซึ่งเหตุการณ์จะเกิดขึ้นโดย การกระทำของผู้ใช้

Event	ลักษณะการควบคุมจากผู้ใช้งาน
Mouse Down	กดปุ่ม Mouse ค้างไว้
Mouse Up	ปล่อยปุ่ม
Click	กดและปล่อยปุ่ม Mouse ในเวลาเดียวกัน
DbClick	กดและปล่อยปุ่ม Mouse 2 ครั้ง ในเวลาเดียวกัน
MouseMove	ขยับ Mouse
Drag	กดปุ่ม Mouse ค้างพร้อมกับเลื่อน Mouse
Drop	กดปุ่ม Mouse ค้างเลื่อน Mouse ปล่อยปุ่ม Mouse

ตารางที่ 2-1 แสดง Event ที่เกิดขึ้นจาก Mouse

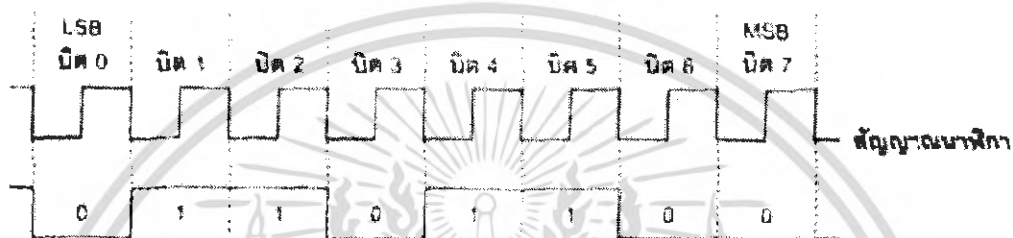
Event	ลักษณะการควบคุมจากผู้ใช้งาน
Key Down	กดปุ่มค้างไว้
Key Up	ปล่อยปุ่ม
Key Press	กดปุ่มและปล่อยปุ่มในเวลาเดียวกัน

ตารางที่ 2-2 แสดง Event ที่เกิดขึ้นจาก Keyboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกา่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่อกันอย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา, ข้อมูลและกราวด์ รูปที่ 2-31 แสดงให้เห็นถึงไทม์มิงไคอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2-31 รูปแบบของข้อมูลอนุกรม

การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกา ร่วมด้วย เหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกันคือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิต หรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต



รูปที่ 2-32 รูปแบบของข้อมูลแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2-32 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลที่ส่ง ขา DATA จะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ(waiting stste) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น จากนั้นบิต ข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด(LSB)ก่อน ซึ่งข้อมูลในแต่ละไบต์ที่จะส่งอาจ จะมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นตามด้วย บิตพาริตี ซึ่งใช้เพื่อตรวจสอบความ ผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่ส่งก็คือ บิตปิดท้าย ซึ่งจะให้ขาเด้ามีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต,1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

2.4.1 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทางโดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียวโดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์(Electronic Industries Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียก กันว่า EIA RS-232 มาตรฐานนี้ใช้ในช่วงแรกจะใช้คอนเน็คเตอร์เป็นแบบ DB-25 โดยกำหนด ความยาวสูงสุด ของสายไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12v แสดงว่ามีข้อมูล(Mark)และ +3 ถึง +12 แสดงว่าเป็นช่องว่าง(Space)

คอนเน็คเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 จะใช้คอนเน็คเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็คเตอร์แบบ DB-25 เนื่องจากขาอื่นๆที่เคยใช้งานในอดีต ปัจจุบันมีการ ใช้งานไม่มากนัก จึงถูกยกเลิกไป

สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232

Data Carrier Detect : DCD หรืออาจจะเรียกว่า Carrier Detect : CD ขานี้จะมีการแอกตีฟเมื่อมีการส่งสัญญาณจากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ถูกใช้งานมากนัก

Receive Data : RD หรือ RxD ขาใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำเอาข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

Transmitted Data : TD หรือ TxD ขาที่ใช้ส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป

Data Terminal Ready : DTR เป็นขาที่สัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางได้รับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทางและขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์

Signal Ground : GND ขากราวด์ของระบบ

Data Set Ready : DSR ขานี้จะถูกต่อกับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

Request To Sent : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RST และ CTS ของตัวของมันเองเข้าด้วยกันเพื่อจะได้ให้การรับ และ ส่งข้อมูลข่าวสารสามารถเกิดขึ้นได้ตลอดเวลา

Clear To Send : CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นจึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

Ring Indicator : RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยปกติสายนี้จะไม่ถูกใช้งาน จะใช้งานคือเมื่อมีการเชื่อมต่อกับโมเด็มและ โปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.4.2 การรับและส่งข้อมูลแบบอนุกรมด้วยวิซวลเบสิก

เนื่องจากระบบปฏิบัติการวินโดว ได้ฝังตัวพอร์ตอนุกรมเข้าเป็นส่วนหนึ่งของระบบปฏิบัติการแล้ว ดังนั้นการเรียกใช้งานจึงจำเป็นต้องเรียกผ่านเครื่องมือที่ติดต่อกับระบบปฏิบัติการ เช่น การใช้คอนโทรล MSCOMM32.OCX ของโปรแกรม Visual Basic คอนโทรล MSComm

สำหรับการใช้งาน Visual Basic ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual Basic จะมีคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมคอมพิวเตอร์มาให้ Mscomm จัดเตรียมเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูลทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์(event driven communication)เป็นรูปแบบที่มีการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่นเมื่อตัวอักษรส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier (DCD) หรือขา Request To Send(RTS) เหตุการณ์ OnComm ของ MSComm จะสามารถตรวจจับสัญญาณนั้นทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ commevent ต่อไป ส่วนทางเลือกที่สองเป็นการตรวจสอบค่าเหตุการณ์ และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ commevent หลังจากที่ให้โปรแกรมทำงานในฟังก์ชันต่างๆไปเรียบร้อยแล้ว ซึ่งวิธีนี้ ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คุณสมบัติของ MSComm

CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่อกับ(com1,com2,com3,com4)

รูปแบบการใช้งาน

Object.CommPort [= value]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดในช่วง 1-6 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด Error68 ขึ้นมา ซึ่งหมายถึงอุปกรณ์ตัวนี้ไม่อยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง PortOpen

Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอดพาริตีจำนวนบิตข้อมูลของบิตปิดท้าย

รูปแบบการใช้งาน

Object.Settings [= value]

ค่าของ Value มีชนิดข้อมูลแบบ String มีรูปแบบเป็น "BBBB,P,D,S"BBBB เป็นค่า อัตราบอด, P เป็นพาริตี, D เป็นจำนวนบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติค่านี้จะถูกกำหนดไว้เป็น "9600,N,8,1"

ค่าของบอดเรตมาตรฐานที่ใช้กับ MSComm มีดังนี้

110	บิตต่อวินาที
300	บิตต่อวินาที
600	บิตต่อวินาที
1,200	บิตต่อวินาที
2,400	บิตต่อวินาที
9,600	บิตต่อวินาที
14,400	บิตต่อวินาที
19,200	บิตต่อวินาที
28,800	บิตต่อวินาที
38,400	บิตต่อวินาที
56,000	บิตต่อวินาที
128,000	บิตต่อวินาที
256,000	บิตต่อวินาที

สำหรับมาตรฐานในการกำหนดค่าพาริตีบิตมีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตีคู่(Even)
M	ลอคิก "1" (MARK)
N	ไม่ใช่(ค่าปกติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- O พาริตี(Odd)
- S ลอจิก "0" (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4,5,6,7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่าคือ 1 (เป็นค่าปกติ),1.5 และ 2

ตัวอย่างการใช้งานคำสั่ง setting โดยจะเป็นการกำหนดค่าบอดเรตเท่ากับ 9600 ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต สามารถเขียนโปรแกรมได้ดังนี้

```
MSComm1.Settings = "9600,N,8,1"
```

หมายเหตุ สาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด " " เนื่องจาก ค่าที่กำหนดนี้อยู่ในรูปแบบตัวแปร String

PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม
รูปแบบการใช้งาน

```
Object.PortOpen [= value]
```

ค่า Value มีชนิดข้อมูลแบบบูลีน คือ True กับ False โดย True หมายถึง การเปิดพอร์ตอนุกรม และ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการเปิดพอร์ตนั้นจะมีการเตรียมรับเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะเปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัตินี้ของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

Input

อ่านค่าและลบค่าขบวนการข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

```
Object.Input
```

คุณสมบัตินี้ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัตินี้ Input การกำหนดให้ค่า InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัตินี้ Input ทำการอ่านข้อมูล ในบัฟเฟอร์รับข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดข้อมูลที่คุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น ComInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น ComInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปแบบของไบนารีและชนิดข้อมูลเป็นแบบ Variant

InBufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

Object.BufferCount[= value]

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป หรือการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนด InBufferCount มีค่าเป็น 0

หมายเหตุ อย่าสับสนระหว่างคำสั่ง InBufferSize และ InBufferCount คำสั่ง InBufferSize นั้นเพื่อใช้กำหนดบัฟเฟอร์ภาครับ

InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็น ไบต์

รูปแบบการใช้งาน

Object.InBufferSize[= value]

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆจะเหลือน้อย อย่างไรก็ตามการกำหนดค่าบัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิด โอเวอร์โฟลวหรือข้อมูลล้นบัฟเฟอร์ เว้นแต่จะมีการแฮนด์เช็ก ดังนั้นค่าปานกลางที่เหมาะสมคือค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลวแล้วจึงค่อยปรับค่าขนาด ของบัฟเฟอร์ให้มีความมากขึ้น

InputLen

การกำหนดค่าและคืนค่าจำนวนของตัวอักษรที่อ่านจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

Object.InputLen [=value]

ค่าเริ่มต้นของคุณสมบัติ InputLen มีค่าเท่ากับ "0" จะทำให้คำสั่ง Input ของ MSComm อ่านค่าข้อมูลที่อยู่ภายในบัฟเฟอร์ภาครับทั้งหมด

ถ้าไม่มีข้อมูลอยู่ในบัฟเฟอร์ภาครับมากเท่ากับจำนวน `UnputLen` จะส่งค่าว่าง (" ") กลับออกมา ผู้ใช้งานสามารถตรวจสอบข้อมูลในบัฟเฟอร์ภาครับได้โดยใช้คุณสมบัติ `InputBufferCount` โดยกำหนดให้มีข้อมูลอยู่ในบัฟเฟอร์ภาครับก่อนแล้วจึงค่อยอ่านค่าข้อมูล บัฟเฟอร์ภาครับ

คุณสมบัตินี้มักใช้กับการอ่านข้อมูลจากเครื่องมือหรือเครื่องจักรที่มีการกำหนดค่าขนาดความยาวของข้อมูลเอาไว้แล้ว

InputMode

กำหนดค่าและคืนค่าชนิดของข้อมูลโดยคำสั่ง `Input`

รูปแบบการใช้งานของคำสั่ง

`Object.InputMode[=value]`

คุณสมบัติ `InputMode` ใช้กำหนดว่าข้อมูลชนิดไหนที่รับเข้ามาผ่านคำสั่ง `Input` โดยข้อมูลจะเลือกได้ 2 ประเภทคือ

`com InputModeText` สำหรับข้อมูลที่อยู่ในรูปของข้อความตัวอักษรตามมาตรฐาน ANSI โดยจะต้องกำหนดค่าเป็น 0 และค่าเริ่มต้นของการรับค่าข้อมูลก็จะเป็นค่านี้

`com InputModeBinary` สำหรับข้อมูลอื่นๆซึ่งจะเก็บในรูปไบนารีรวมกันอยู่เป็นไบนารีข้อมูล

Output

ใช้ในการส่งขบวนข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

`Object.Output [=value]`

ค่า `value` เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติ `Output` สามารถส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีก็ได้ โดยการส่งข้อมูลเป็นแบบตัวอักษรจะต้องกำหนดข้อมูลแบบ `Variant` และมีข้อมูลภายในแบบ `String` สำหรับการส่งข้อมูลไบนารีจะต้องกำหนดชนิดของข้อมูลเป็นแบบ `Variant` และมีข้อมูลภายในแบบ `Byte`

OutBufferCount

คืนค่าจำนวนของข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาครับ สามารถใช้คำสั่งนี้เคลียร์บัฟเฟอร์ภาครับได้อีกด้วย

รูปแบบการใช้งาน

`Object.OutBufferCount [=value]`

ผู้ใช้งานสามารถเคลียร์บัฟเฟอร์ภาครับได้โดยการกำหนดค่า `OutBufferCount` เท่ากับ "0"

หมายเหตุ ระวังการสับสนระหว่างคุณสมบัติ `OutBufferCount` กับ `OutBufferSize` ซึ่ง `OutBufferSize` ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดของตัวแปรเป็นแบบไบต์
รูปแบบการใช้งาน

Object.OutBufferSize[=value]

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยปกติค่าที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาคส่งที่มากเกินไปจะทำให้ มีหน่วยความจำเหลือใช้งานน้อย แต่อย่างไรก็ตามถ้ากำหนดค่าน้อยเกินไป จะทำให้ข้อมูลสั้นบัฟเฟอร์ขึ้นได้ยกเว้นจะมีการใช้แฮนด์เช็กวิธีการที่ถูกต้องในการกำหนดค่าคือ ทดลองใช้ค่าเริ่มต้นคือค่า 523 ไบต์ดูก่อนถ้าโปรแกรมทำงานแล้วเกิดการล้นของข้อมูลค่อยเพิ่มค่าของ OutBufferSize ให้มากขึ้น

PartyReplace

กำหนดค่าและคืนค่าตัวอักษรที่ไปวางแทนตำแหน่งที่เกิดข้อผิดพลาดจากพาริตี
รูปแบบและการใช้งานคำสั่ง

Object.PartyReplace[=value]

บิตพาริตี เป็นบิตที่ทางภาคส่งข้อมูลทำการส่งมาพร้อมกับข้อมูล เพื่อตรวจสอบข้อผิดพลาดของข้อมูล โดยเมื่อมีการใช้บิตพาริตีคอนโทรล MSCOMM จะทำการบวกบิตทุกบิตที่มีค่าลอจิก "1" ในแต่ละไบต์ ละทำการตรวจสอบผลลัพธ์ว่าบิตที่อ่านได้นั้นมีจำนวนลอจิก "1" เป็นเลขคู่หรือเลขคี่ และตรงคีย์ที่กำหนดไว้แต่ต้นหรือไม่ ถ้าค่าที่นำมาบวกพาริตีที่ไม่ตรงแสดงว่าการรับส่งข้อมูลผิดพลาด

การกำหนดค่าเริ่มต้นให้กับ PartyReplace นั้นกำหนดให้ใช้เครื่องหมาย(?) ไปวางไว้ที่ตำแหน่งที่เกิดพาริตีผิดพลาด ถ้ากำหนดค่า PartyReplace ให้เป็นค่าว่าง (" ")จะเป็นการยกเลิกใช้งาน PartyReplace และไม่มีการป้อนข้อมูลแทนเมื่อตรวจพบข้อผิดพลาด

PartyReplace ใช้ข้อมูลเป็นแบบสตริงแต่การกำหนด จะกำหนดได้เพียงไบต์เดียวเท่านั้น ซึ่งสามารถใช้ค่าใดๆก็ได้ที่เป็นโค้ด ANSI มีอยู่ระหว่าง 0 - 255

DTERnable

ใช้ในการกำหนดสถานะลอจิกขา Data Terminal Ready(DTR) โดยสัญญาณของ DTR จะส่งออกจากคอมพิวเตอร์ไปยังโมเด็มเพื่อแสดงว่าคอมพิวเตอร์พร้อมที่จะรับข้อมูลแล้ว ชนิดข้อมูล เป็นแบบบูลีน

รูปแบบการใช้งาน

object.DTERnable[=value]

ค่า value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิกของขา DTR ให้เป็น "0" หรือ "1" โดย

True หมายถึง ให้ขา DTR มีลอจิก "1"

False หมายถึง ให้ขา DTR มีลอจิกเป็น "0"(เป็นค่าปกติ)

หมายเหตุ เมื่อขา TDR ถูกกำหนดสถานะให้เป็น True ที่ขา DTR จะมีสถานะเป็นลอจิก "1" เมื่อทำการเปิดพอร์ตและจะปิดเมื่อมีสถานะเป็น "0" เมื่อมีการปิดพอร์ต เมื่อขา DTR ถูกกำหนดสถานะเป็น False เมื่อขา DTR จะมีสถานะลอจิก "0" ตลอดเวลาไม่ว่าจะใช้คำสั่งเปิดพอร์ตหรือปิดพอร์ต

สำหรับการใช้งานกับโมเด็ม การทำให้ขา DTR เป็นลอจิก "0" จะเป็นการวางหูโทรศัพท์หรือการยกเลิกการติดต่อ

RTSEnable

เพื่อใช้กำหนดสถานะลอจิกให้ขา Request To Send(RTS)โดยที่ขา RTS เป็นสัญญาณที่ส่งออกจากคอมพิวเตอร์ไปยังโมเด็มเพื่อขอร้องขอส่งข้อมูล ชนิดของข้อมูลเป็นแบบ Boolean

รูปแบบการใช้งาน

Object.RTSEnable[=value]

ค่า value เป็นค่าสถานะ True หรือ False เพื่อกำหนดลอจิก "0" หรือ "1" ให้ขา RTS โดย

True หมายถึง ให้ขา RTS มีลอจิก "1"

False หมายถึง ให้ขา RTS มีลอจิก "0"(เป็นค่าปกติ)

หมายเหตุ เมื่อขา RTSEnable ถูกกำหนดให้เป็น True ขา RTS จะมีสถานะลอจิก "1" เมื่อเปิดพอร์ตและมีสถานะลอจิกเป็น "0" เมื่อปิดพอร์ต

EOFEnable

เป็นการกำหนดให้ MSComm รอสัญลักษณ์แสดงส่วนท้ายที่สุดของไฟล์(End Of File : EOF) ระหว่างรับอินพุตเข้ามา ถ้าพบสัญลักษณ์ EOF ภาคอินพุตจะหยุดรับข้อมูล และเหตุการณ์ OnComm จะถูกระตุ้นให้ทำงาน คุณสมบัติ CommEvent จะมีค่าเท่ากับ 7 หรือ ConEvEOF

รูปแบบการใช้งาน

object.EOFEnable[=value]

โดยใช้ value เป็นสถานะ True หรือ False เพื่ออินามิเบิลหรือดิสอเบิ้ล การทำงานของเหตุการณ์ OnComm เมื่อตรวจพบสัญลักษณ์ EFO โดย

True หมายถึง เหตุการณ์ OnComm จะถูกกระตุ้นให้ทำงานด้วย EOF

False หมายถึง เหตุการณ์ OnComm จะไม่ถูกกระตุ้นให้ทำงานด้วย EOF(เป็นค่าปกติ)

เมื่อ EOFEnable กำหนดให้เป็น False ส่วนควบคุมจะไม่มีการตรวจสอบสัญลักษณ์ EOF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CTSHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Clear To Send(CTS) ได้ว่ามีสถานะลอจิก "0" หรือ "1" โดยค่าที่อ่านได้จะเป็นบูลีน True และ False 5hk8jk CTSHolding เป็น True ขา CTS จะมีสถานะลอจิก "1" ถ้าค่า CTSHolding เป็น False ขา CTS จะมีสถานะลอจิก "0"

รูปแบบการใช้งาน

object.CTSHolding

เมื่อขา CTS เป็นลอจิก"0"(CTSHolding=False)และเกิดไทย์เอาต์คอนโทรล MsComm จะกำหนดคุณสมบัติ CommEvent มีค่าเป็น comEventCTSTO (Clear To Send Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

CDHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา Data Carrier Detect(DCD)ได้ว่ามีสถานะลอจิก เป็น "1" หรือ "0" โดยค่าที่อ่านได้จะเป็นบูลีน Truse และ False ถ้าค่า CDHolding เป็น True ขา DCD มีสถานะลอจิก "1" ถ้าค่า CDHolding เป็น False ขา DCD มีสถานะลอจิก "0"

รูปแบบการใช้งาน

Object.CDHolding

เมื่อขา DCD มีลอจิก "1"(CDHolding=True)และเกิดไทย์เอาต์คอนโทรล MsComm จะกำหนดคุณสมบัติ CommEvent มีค่าเป็น comEventCTSTO (Clear To Send Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

DSRHolding

ผู้ใช้งานสามารถตรวจสอบการทำงานของขา DSR ได้ว่ามีสถานะลอจิกเป็น "1" หรือ "0" โดยค่าที่อ่านได้จะเป็นบูลีน True และ False ถ้าค่าDSRHolding เป็น True ขา DSR มีสถานะลอจิก "1" ถ้าค่า DSRHolding เป็น False ขา DSR มีสถานะลอจิก "0"

รูปแบบการใช้งาน

object.DSRHolding

เมื่อขา CTS เป็นลอจิก "1"(CTSHolding=True)และเกิดไทย์เอาต์คอนโทรล MsComm จะกำหนดคุณสมบัติ CommEvent มีค่าเป็น comEventDSRTO (Data Set Ready Timeout) และกระตุ้นให้เกิดเหตุการณ์ OnComm

Handshaking

กำหนดคุณสมบัติและค่าน่ารูปแบบแฮนด์เช็กทางฮาร์ดแวร์

รูปแบบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Object.Handshaking[=value]

ค่าตัวแปร Value ที่ใช้กำหนดได้ 4 รูปแบบด้วยกันคือ

1.comNone ค่าที่กำหนดคือ 0 เป็นการกำหนดให้ไม่มีการแฮนด์เช็ก(เป็นค่าเริ่มต้น)

2.comXOnXOff ค่าที่กำหนดคือ 1 เป็นการกำหนดให้แฮนด์เช็ก XONXOFF

3.comRTS ค่าที่กำหนดคือ 2 เป็นการกำหนดให้ใช้ขา RTS/CTS(Request To Send / Clear To Send)

4.CommRTSXOnXOff ค่าที่กำหนดคือ 3 เป็นการกำหนดให้ใช้ทั้งแบบ Request To Send และ XOnXOff

คุณสมบัติ Handshaking ใช้เพื่อกำหนดการสื่อสารภายใน ระหว่างที่ข้อมูลถูกส่งไปยังบัพเฟอร์ภาครับ เมื่อข้อมูลตัวอักษรถูกส่งมายังพอร์ตอนุกรม อุปกรณ์สื่อสารข้อมูลจะนำการย้ายข้อมูล บัพเฟอร์ภาครับ เพื่อที่จะทำให้โปรแกรมสามารถอ่านค่าไปใช้งานได้ถ้าไม่มีบัพเฟอร์ภาครับ โปรแกรมที่ใช้งานจะต้องทำการอ่านค่าข้อมูลโดยตรงจากฮาร์ดแวร์ของพอร์ตอนุกรม ซึ่งผู้ใช้งานจะเกิดปัญหาข้อมูลสูญหาย เนื่องจากการเปลี่ยนแปลงข้อมูลที่ส่งเข้ามามีการเปลี่ยนแปลงอย่างรวดเร็ว

คุณสมบัติ Handshaking ช่วยให้ผู้ใช้งานแน่ใจว่าข้อมูลที่ได้รับมานั้น ไม่มีการสูญหายบัพเฟอร์ภาครับที่รับข้อมูลนั้นเกิดข้อมูลล้นหรือโอเวอร์โฟลว โดยใช้วิธีการตรวจสอบความพร้อมของบัพเฟอร์ว่าพร้อมรับข้อมูลหรือไม่ก่อนที่จะส่งข้อมูลมาให้

Break

ใช้ในการเซตและเคลียร์ค่าสัญญาณ Break ชนิดข้อมูลเป็นแบบบูลีน
รูปแบบการใช้งาน

Object.break[=value]

โดย value เป็นค่าบูลีน ถ้า value=True หมายถึง การส่งสัญญาณ Break ออกไป ถ้า value=False หมายถึงการเคลียร์สัญญาณ Break

เมื่อกำหนดสัญญาณ Break เป็น True จะเป็นการหยุดการส่งข้อมูลจนกว่าจะมีการสั่งให้สัญญาณ Break เป็น False

2.5 ความรู้เบื้องต้นของไมโครคอนโทรลเลอร์ dsPIC30F2010

dsPIC คือชื่อของไมโครคอนโทรลเลอร์ 16 บิตจาก Microchip Technology Inc. ผู้ผลิตไมโครคอนโทรลเลอร์ PIC ซึ่งรู้จักกันดีในแวดวงนักพัฒนาระบบไมโครคอนโทรลเลอร์ โดย Microchip Technology ได้กำหนดชื่ออย่างเป็นทางการสำหรับไมโครคอนโทรลเลอร์อนุกรมใหม่นี้ว่า Digital Signal Controller หรือ DSC นั้นหมายความว่า dsPIC เป็นไมโครคอนโทรลเลอร์ ที่ได้รับการออกแบบมาเป็นพิเศษเพื่องานประมวลผลสัญญาณดิจิทัลสำหรับสร้างระบบควบคุมอัตโนมัติที่มีความสามารถสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 คุณสมบัติทางเทคนิคของ dsPIC30F2010 มีดังนี้

คุณสมบัติของซีพียู

- มี 84 คำสั่งมาตรฐาน สามารถรองรับรูปแบบการอ้างแอดเดรสได้อย่างอิสระ
- ชุดคำสั่งมีขนาด 24 บิต สามารถประมวลผลข้อมูลได้ 16 บิต
- มีหน่วยความจำโปรแกรมแบบแฟลช ความจุ 12 กิโลไบต์ ลบและเขียนใหม่ได้ไม่น้อยกว่า 100,000 ครั้ง สามารถป้องกันการอ่านได้
- สามารถโปรแกรมหน่วยความจำโปรแกรมได้ด้วยตัวเอง โดยการใช้กระบวนการทางซอฟต์แวร์
- มีหน่วยความจำข้อมูลอีพรอม 1 กิโลไบต์ ลบและเขียนใหม่ได้ไม่น้อยกว่า 1,000,000 ครั้ง
- มีหน่วยความจำข้อมูลแรม 512 ไบต์
- รีจิสเตอร์ W จัดในรูปของอะเรย์ มีขนาด 16 บิต จำนวน 16 ตัว
- ความเร็วในการทำงานสูงถึง 30 ล้านคำสั่งต่อวินาที
- ความถี่สัญญาณนาฬิกาจากภายนอก ตั้งแต่ย่านไฟตรงจนถึง 40 เมกะเฮิร์ต
- ความถี่สัญญาณนาฬิกาในกรณีใช้งานร่วมกับวงจรเฟลลีสติกภายใน ตั้งแต่ 4 เมกะเฮิร์ต ถึง 10 เมกะเฮิร์ต เลือกได้ 3 ระดับคือ 4, 8 หรือ 16 เท่า
- สามารถกำหนดระดับความสำคัญในการตอบสนองอินเทอร์รัปต์ได้ 8 ระดับ
- มีอินเทอร์รัปต์เวกเตอร์ 48 ตำแหน่ง
- มีวงจรตรวจจับความดันไฟเลี้ยงต่ำกว่ากำหนดในโปรแกรมได้
- มีเพาเวอร์ - ออนรีเซต, เพาเวอร์ - อับไทเมอร์ และออสซิลเลเตอร์สตาร์ท - อับไทเมอร์
- มีวอตช์ด็อกไทมเมอร์แบบโปรแกรมได้
- มีวงจรตรวจสอบการทำงานของวงจรถูกกำเนิดสัญญาณนาฬิกา หากผิดพลาดจะเข้าสู่โหมดสัญญาณนาฬิกา RC พลังงานต่ำทันที
- รองรับการโปรแกรมในวงจรแบบอนุกรม(ICSP : In-Circuit Serial Programming)
- สามารถเลือกโหมดการใช้พลังงานได้
- ย่านไฟเลี้ยง 2.5 ถึง 5.5 โวลต์ กระแสไฟฟ้า 2.6 ถึง 44 มิลลิแอมป์ ที่ไฟเลี้ยง +5 โวลต์ ขึ้นอยู่กับการกำหนดความเร็วในการทำงาน

คุณสมบัติด้านการประมวลสัญญาณดิจิทัล

- มีแอกคิวมูเลเตอร์ขนาด 40 บิต 2 ตัว รองรับการประมวลผลทางคณิตศาสตร์ได้เป็นอย่างดี
- มาหน่วยประมวลผลด้านการคูณและหารเลข 17 บิต ในรูปของฮาร์ดแวร์ จึงทำให้สามารถทำการคูณและหารเลขได้อย่างรวดเร็ว
- ทำการคูณเลข 16 บิต ได้ภายในสัญญาณนาฬิกาเพียง 1 ไซเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีตัวเลื่อนข้อมูลบาร์เรล 40 สเตจ ช่วยให้การประมวลผลข้อมูลที่จำนวนบิตมากทำได้รวดเร็ว
- มีวงจรเฟตซ์ข้อมูลคู่ จึงทำให้สามารถประมวลผลข้อมูลได้อย่างรวดเร็ว

คุณสมบัติของไมโครพังก์ชันพิเศษ

- สามารถจ่ายกระแสออกทางขาพอร์ตได้ 25 มิลลิแอมป์ ทั้งแบบกระแสซิงก์และซอร์ส
 - ไทมเมอร์/เคาเตอร์ 16 บิต 3 ตัว สามารถต่อใช้งานร่วมกันเป็นไทมเมอร์/เคาเตอร์ 32 บิตได้
 - มีไมโครตรวจจับสัญญาณดิจิทัลขนาด 16 บิต 4 ชุด
 - มีไมโครเปรียบเทียบข้อมูลและกำเนิดสัญญาณ PWM ความละเอียด 16 บิต 2 ชุด
- ในการเปรียบเทียบข้อมูลสามารถเลือกการทำงานได้ 2 โหมด
- มีส่วนเชื่อมต่ออุปกรณ์อนุกรมแบบ SPI
 - มีส่วนเชื่อมต่ออุปกรณ์ผ่านระบบบัส I²C ทั้งแบบ 7 และ 10 บิต กำหนดเป็นมาสเตอร์หรือสเลฟได้
 - มีไมโครสื่อสารข้อมูลอนุกรม UART พร้อมบัฟเฟอร์แบบ FIFO
 - มีไมโครสร้างสัญญาณ PWM สำหรับควบคุมมอเตอร์ 6 ช่อง
 - เลือกรูปแบบเอาต์พุตได้ทั้งแบบคอมพลีเมนต์และแบบอิสระ
 - มีโหมดปรับตำแหน่งการหมุนทั้งแบบปรับขอบสัญญาณและแบบกึ่งกลาง
 - มีส่วนกำเนิดควิตซ์ไจกิล 4 ชุด
 - มีสัญญาณกระตุ้นเพื่อให้ทำงานสัมพันธ์กับวงจรแหล่งสัญญาณอนาลอกเป็นดิจิทัลภายในไมโครคอนโทรลเลอร์
 - สามารถควบคุมสัญญาณเอาต์พุตได้
 - คุณสมบัติด้านการประมวลสัญญาณดิจิทัล
 - มีไมโครเชื่อมต่อตัวเข้ารหัสแบบควอดราเจอร์
 - มีอินพุต Phase A, Phase B ละรับสัญญาณพัลส์เพื่อกำหนดตำแหน่ง
 - มีตัวนับจำนวนขนาด 16 บิต นับได้ทั้งขึ้นและลง
 - แสดงสถานะของทิศทางการนับได้
 - กำหนดโหมดของการวัดตำแหน่งได้ 2 โหมดคือ x2 และ x4
 - มีวงจรรองสัญญาณรบกวนแบบดิจิทัลจากอินพุตแบบโปรแกรมนได้
 - สำหรับการกำหนดการทำงานให้เป็นไทมเมอร์/เคาเตอร์ ขนาด 16 บิตได้
 - กำเนิดสัญญาณอินเตอร์รัปต์จากตำแหน่งที่นับเกิน(Rollover) หรือนับแบบขาด(Underflow)
 - มีวงจรแหล่งสัญญาณอนาลอกเป็นดิจิทัล ความละเอียด 10 บิต 6 ช่อง
 - อัตราการสุ่มและแปลงสัญญาณ 500 กิโลแซมเปิลต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถแหล่งสัญญาณเมื่อไมโครคอนโทรลเลอร์ทำงานในโหมดสลีปและไอดีลได้

2.5.2 สถาปัตยกรรมโดยสรุปของ dsPIC30F2010

2.5.2.1 หน่วยประมวลผลกลาง

หน่วยประมวลผลกลางของ dsPIC30F2010 ใช้คำสั่งที่มีความยาว 1 เวิร์ด ขนาด 24 บิต โดยมี โปรแกรมเคาเตอร์ขนาด 23 บิต (จริงๆแล้วโดยโครงสร้างมี 24 บิต แต่ไม่สนใจบิต MSB ซึ่งก็คือบิต 23 และบิต LSB หรือบิต 0 กำหนดเป็น "0" จึงทำให้สามารถติดต่อหน่วยความจำ โปรแกรมได้สูงสุด (4 เมกะเวิร์ด) เพื่อแฉ่งแอดเดรสของหน่วยความจำโปรแกรมที่เข้าไปประมวลผล dsPIC30F2010 มีความจุของหน่วยความจำโปรแกรม 12 กิโลไบต์ เมื่อคำสั่งมีความยาว 24 บิต จึงบรรจุคำสั่งได้จริง 4 กิโลเวิร์ด

รีจิสเตอร์หลักที่ใช้ในการทำงานคือ รีจิสเตอร์ W (Working register) สำหรับใน dsPIC จะแตกต่างจากไมโครคอนโทรลเลอร์ PIC อย่างมาก โดยรีจิสเตอร์ W ได้รับการจัดโครงสร้างเป็นอะเรย์ขนาด 16 บิต จึงทำให้สามารถรองรับทั้งข้อมูล, ค่าแอดเดรส หรือค่าของรีจิสเตอร์ใดๆที่ต้องนำมาประมวลผลโดยใน dsPIC มีรีจิสเตอร์ W ให้ใช้งานถึง 16 ตัว ส่วนใหญ่ใช้ในการประมวลผลหลัก ส่วนอีกตัวหนึ่งคือ รีจิสเตอร์ W15 จะใช้ทำงานร่วมกับตัวชี้สแต็กในการทำงานของโปรแกรมย่อยและบริการอินเตอร์รัปต์

ด้านการตอบสนองอินเตอร์รัปต์นั้น dsPIC30F2010 มีการจัดสรรพื้นที่เก็บค่าอินเตอร์รัปต์เวกเตอร์ไว้มากถึง 54 ตำแหน่ง และยังสามารถกำหนดระดับความสำคัญได้อีก 8 ระดับด้วย

2.5.2.2 หน่วยความจำ

dsPIC30F2010 มีหน่วยความจำโปรแกรม 4 กิโลเวิร์ด แอดเดรสอยู่ในช่วง 0x000100 ถึง 0x001FFE สามารถโปรแกรมหรือเขียนข้อมูลลงไปได้ 2 วิธีคือ

1. โดยการใช้โปรแกรมในวงจรแบบอนุกรมหรือ ICSP ผ่านทางขา PGD และ PGC(ขาที่ 17 และ 18) แล้วป้อนสัญญาณพัลส์แรงดันสูงสำหรับโปรแกรมผ่านเข้ามาทางขา MCLR
2. โปรแกรมตัวเองในขณะที่ทำงานหรือ RTSP(Run Time Self-Programming) ส่วนหน่วยความจำข้อมูลแรมนั้น dsPIC30F2010 ได้จัดสรรเป็น 2 ส่วนคือ หน่วยความจำข้อมูลแรม X และ Y แต่ละส่วนมีขนาด 16 บิต ความจุ 256 ไบต์ รวมเป็น 512 ไบต์ โดยในแต่ละส่วนจะมีตัวกำหนดแอดเดรสแยกออกจากกัน เรียกว่า AGU(Address Generation Unit)

ในขณะที่หน่วยความจำข้อมูลอีพรอม dsPIC30F2010 จัดสรรไว้ที่แอดเดรส 0x7FFC00 ถึง 0x7FFFFE มีความจุ 1 กิโลไบต์

2.5.2.3 ส่วนประมวลผลสัญญาณดิจิทัล

นับเป็นส่วนประกอบที่สำคัญอย่างยิ่งของ dsPIC เนื่องจาก dsPIC ได้รับการออกแบบมา ให้ทำงานในด้านการประมวลผลสัญญาณดิจิทัลเป็นหลัก ดังนั้นจึงต้องมีการเพิ่มความสามารถใน หน่วยคำนวณทางคณิตศาสตร์และลอจิกอย่างมาก โดยในส่วนประมวลผลสัญญาณดิจิทัลมีหน่วย จัดการคูณเลขขนาด 17×17 บิตความเร็วสูง, หน่วยประมวลผลทางคณิตศาสตร์และลอจิกหรือ ALU ขนาด 40 บิต, แอควิวเลเตอร์ขนาด 40 บิต อีก 2 ตัว และตัวเลื่อนข้อมูล 2 ทิศทางแบบ บารเรล (Barrel Shifter) ขนาด 40 บิต จึงทำให้สามารถจัดการข้อมูลขนาด 16 บิต ได้เสร็จสิ้น ภายในสัญญาณนาฬิกาเพียงไซเคิลเดียว

2.5.2.4 โมดูลฟังก์ชันพิเศษ

dsPIC30F2010 ได้รวมเอาโมดูลสำหรับทำงานเฉพาะทางไว้อย่างมากมาย ไม่ว่าจะเป็นโมดูล แปลงสัญญาณอะนาล็อกเป็นดิจิทัล ความละเอียด 10 บิต, โมดูลเชื่อมต่ออุปกรณ์อนุกรมหรือ SPI, โมดูลสื่อสารข้อมูลบนระบบบัส I²C, โมดูลสื่อสารผ่านพอร์ตอนุกรมหรือ UART, ไทเมอร์และ โมดูลเข้ารหัส ควอดร่าเจอร์ โดยสามารถใช้งานร่วมกันเพื่อสร้างระบบควบคุมมอเตอร์แบบปิด ประสิทธิภาพสูง

2.5.2.5 พอร์ตอินพุตเอาต์พุต

dsPIC30F2010 มีพอร์ตให้ใช้งานมากถึง 5 พอร์ต รวม 20 ขา ดังนี้

พอร์ต B มี 6 ขาคือ RB0 - RB5 โดยทุกขาสามารถกำหนดให้เป็นพอร์ตอินพุตหรือ เอาพุตได้และยังสามารถขับเคลื่อนได้ทั้งแบบซิงค์และซอร์สได้สูงถึง 25 มิลลิแอมป์

พอร์ต C มี 3 ขาคือ RC13 - RC15

พอร์ต D มี 2 ขาคือ RD0 และ RD1

พอร์ต E มี 7 ขาคือ RE0 - RE5 และ RE8

พอร์ต F มี 2 ขาคือ RF2 และ RF3

2.5.2.6 โครงสร้างทางโปรแกรมที่ควรทราบ

โครงสร้างทางโปรแกรมหรือ Programmer's model ของ dsPIC ที่นักพัฒนาควรจะต้อง ทราบ ประกอบด้วย

- รีจิสเตอร์ W ขนาด 16 บิต มีมากถึง 16 ตัว โดยตัวที่ใช้งานเป็นหลักคือ W0 ส่วนตัวที่ถูกนำไปใช้ใน ส่วนประมวลผลสัญญาณดิจิทัลมี 10 ตัว คือ W4 - W13 ส่วน W14 ถูกนำไปใช้ร่วมเป็นตัวชี้เฟรม และ W15 ถูกนำไปใช้ร่วมเป็นตัวชี้สเตจ
- แอควิวเลเตอร์ 40 บิต จำนวน 2 ตัว ใช้ในงานประมวลผลสัญญาณดิจิทัลเป็นหลัก
- โปรแกรมเคาเตอร์ ขนาด 24 บิต นำมาใช้ในการแจ้งแอดเดรส 23 บิต โดยไม่สนใจ บิต MSB และบิต LSB ต้องเป็น "0"
- รีจิสเตอร์หลัก อันประกอบไปด้วย

STATUS ซึ่งใช้แสดงสถานการณ์ทำงาน มีขนาด 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต หากมีข้อผิดพลาดประการใด ขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CORCON ใช้ควบคุมการทำงานของหน่วยประมวลผลกลาง มีขนาด 16 บิต
- TBLPAG เป็นรีจิสเตอร์กำหนดเพจของตารางข้อมูลในหน่วยความจำ โปรแกรมมีขนาด 8 บิต
- PSVPAG เป็นรีจิสเตอร์แสดงเพจแอดเดรสของพื้นที่โปรแกรม มีขนาด 8 บิต
- RCOUNT เป็นรีจิสเตอร์เก็บค่าตัวนับจำนวนรอบของลูปที่ทำซ้ำ
- DCOUNT เป็นรีจิสเตอร์เก็บค่าตัวนับจำนวนรอบของลูปที่ทำงาน
- DOSTART เป็นรีจิสเตอร์กำหนดแอดเดรสเริ่มต้นทำงานของโปรแกรมลูป
- DOEND เป็นรีจิสเตอร์กำหนดแอดเดรสปลายทางที่ต้องการทำโปรแกรมลูปสำหรับรีจิสเตอร์ DCOUNT, DOSTART และ DOEND เป็นรีจิสเตอร์เงา(Shadow Register) หมายความว่า เป็น รีจิสเตอร์ ที่ถูกสร้างขึ้นชั่วคราวเพื่อเก็บค่าก่อนที่จะมีการ ถ่ายทอดออกไปทำงาน จึงไม่สามารถเข้าถึงรีจิสเตอร์เหล่านี้ได้ โดยตรง

2.6 หลักการจูนค่าพีไอดี

การคอนโทรลที่ใช้ในการปรับแต่งค่าพีไอดี คือ การคอนโทรล แบบคิจิตรัลคอนโทรล ซึ่งถูกควบคุมโดยโปรแกรมภาษาซีในไมโคร คอนโทรลเลอร์ ซึ่งสมการที่ใช้เป็นดังนี้คือ

$$PWM = K_1 e_1(t) + K_2 e_2(t) + K_3 e_3(t) \tag{2-27}$$

เมื่อ PWM เป็นสัญญาณควบคุมในการขับมอเตอร์

K_1 เป็นค่าคงที่ ปรีออปพอร์ชันแนล

K_2 เป็นค่าคงที่ อินติกรัล

K_3 เป็นค่าคงที่ คิริวิทิว

$e_1(t)$ เป็นสัญญาณเออร์เรอร์ที่ถูกป้อนกลับจากสัญญาณ เอาท์พุทซึ่งมีค่าดังนี้

$$e_1(t) = V_{ref} - V_{output} \tag{2-28}$$

โดยที่ V_{ref} เป็นความเร็วที่ใช้อ้างอิงในการปรับแต่งค่า พีไอดี

V_{output} เป็นความเร็วที่ใช้ขับมอเตอร์ที่เวลานั้น ๆ

$e_2(t)$ เป็นค่าอินติกรัลของสัญญาณเออร์เรอร์ $e_1(t)$ ซึ่งมีค่าดังนี้

$$e_2(t) = \sum [e_1(t) + (e_1(t) - e_1(t-1)) / 2] \Delta T \tag{2-29}$$

โดยที่ ΔT เป็นช่วงของเวลาที่ใช้ในการคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$e_3(t)$ เป็นค่าดิฟเฟอเรนเชียลของค่าออร์เรอร์ $e_1(t)$ ซึ่งมีค่าดังนี้

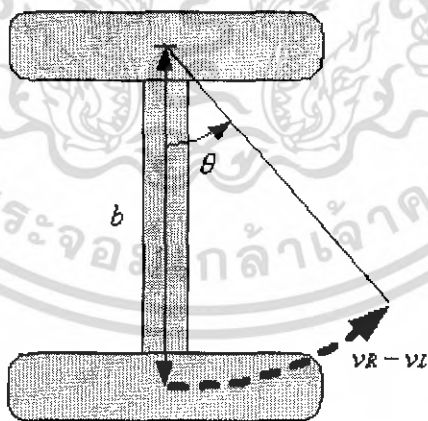
$$e_3(t) = [e_1(t) - e_1(t-1)] / \Delta T \quad (2-30)$$

2.6.1 วิธีการปรับค่าคอนโทรลเลอร์แบบพีไอดี

การทดลองนำสมการที่ (2-27) ซึ่งเป็นสมการที่ใช้ในการควบคุมแบบ พีไอดี โดยทำการให้ค่า V_{ref} เป็นสเต็ปอินพุท แล้วทำการ เพิ่มค่าคงที่ K_1 , K_2 และ K_3 โดยที่หลักในการเพิ่มค่าคงที่แต่ละตัวนั้น ซึ่งเริ่มต้นจากการเพิ่มค่าคงที่ K_1 ขึ้นไปเรื่อยๆจนได้ค่า K_1 ที่ดีที่สุดซึ่งผลตอบสนองที่ได้จากค่าคงที่ K_1 นั้นจะมีค่าเข้าใกล้ค่าอินพุท ไรสไทม์มีค่าปานกลาง และ สัญญาณความเร็วที่ขับออกมาไม่เกิดการ ออสซิลเรท จากนั้นทำการปรับค่าคงที่ K_2 ซึ่งผลตอบสนองที่ได้จะเป็นระบบพีไอ และลักษณะการจูนที่ดี ผลตอบสนองที่ได้มีค่าใกล้เคียงค่าอินพุทมาก อาจมีโอเวอร์ชูตบ้างเล็กน้อย แต่ช่วงเวลาของไรสไทม์ จะยาวขึ้นจาก นั้นทำการจูนค่าคงที่ K_3 ซึ่งผลตอบสนองที่ได้จะเป็นระบบพีไอดี ลักษณะการจูนที่ดี ผลตอบสนองที่ได้มีค่าใกล้เคียงค่าอินพุทมาก หรือ อยู่ในระดับเดียวกับค่าอินพุทของระบบ โอเวอร์ชูตลดลง ช่วงเวลาของไรสไทม์จะสั้นลง และช่วงเซตติงไทม์สั้นขึ้น

2.7 ทฤษฎีหุ่นยนต์ 2 ล้อ

ฟังก์ชันในการเคลื่อนที่โดยปกติจะพิจารณาในกรณีที่ล้อของหุ่นยนต์ทั้ง 2 ล้อมีความเร็วเท่ากัน แต่โรครณีที่หุ่นยนต์เคลื่อนที่ในวิถีโค้งซึ่งล้อทั้ง 2 ข้างมีความเร็วไม่เท่ากัน เราพิจารณาจากรูปที่ 2-33 ได้ดังนี้



รูปที่ 2-33 แสดงลักษณะของล้อที่มีความเร็วต่างกัน

จากรูปจุดที่เลือกใช้เป็นจุดอ้างอิงหรือจุดศูนย์กลางการเคลื่อนที่ของหุ่นยนต์คือล้อซ้าย การเคลื่อนที่ของล้อขวาจะเคลื่อนที่ในทิศทางตั้งฉากกับจุดอ้างอิง โดยเคลื่อนที่เป็นวงกลมรอบจุดอ้างอิง

การหาสมการเชิงอนุพันธ์ของการเปลี่ยนทิศทางที่สัมพันธ์กับเวลา จะได้ความเร็วเชิงมุมจะสัมพันธ์กับอัตราส่วนของล้อขวาที่เคลื่อนที่เป็นส่วนโค้งต่อเวลา กับระยะทางจากล้อถึงจุดศูนย์กลางรัศมี ดังแสดงในสมการที่ (2-31)

$$d\theta/dt=(v_R-v_L)/b \quad (2-31)$$

เมื่อทำการอินทิเกรตสมการที่ (2-31) และ จุดทิศทางเริ่มต้นของหุ่นยนต์ คือ $\theta(0) = \theta_0$ เราจะทำการคำนวณหาค่าฟังก์ชันได้ดังนี้

$$\theta(t)=(v_R-v_L)t/b+\theta_0 \quad (2-32)$$

การเปลี่ยนตำแหน่งเชิงมุมของหุ่นยนต์จะสัมพันธ์กับตำแหน่งเชิงเส้น ซึ่งตำแหน่งเชิงเส้นของหุ่นยนต์ที่เปลี่ยนแปลง เกี่ยวข้องกับความเร็วรวมเฉลี่ยทั้ง 2 ล้อของตัวหุ่นยนต์ หรือ $(v_R+v_L)/2$ ซึ่งสามารถหาสมการเชิงอนุพันธ์ได้ดังนี้

$$dx/dt=[(v_R+v_L)/2]\cos(\theta(t)) \quad (2-33)$$

$$dy/dt=[(v_R+v_L)/2]\sin(\theta(t)) \quad (2-34)$$

จากสมการที่ (2-33) และ (2-34) เมื่อทำการอินทิเกรต และตำแหน่งเริ่มต้นของตัวหุ่นยนต์เป็น $x(0)=x_0, y(0)=y_0$ จะได้

$$x(t)=x_0+b(v_R+v_L)/2(v_R-v_L) [\sin((v_R-v_L)t/b+\theta_0) - \sin(\theta_0)] \quad (2-35)$$

$$y(t)=y_0-b(v_R+v_L)/2(v_R-v_L) [\cos((v_R-v_L)t/b+\theta_0) - \cos(\theta_0)] \quad (2-36)$$

จากสมการที่ (2-35) และ (2-36) เทอม $b(v_R+v_L)/2$ คือ รัศมีการเลี้ยวเป็นส่วนโค้งที่จุดศูนย์กลางหุ่นยนต์ และเมื่อความเร็วทั้ง 2 ล้อ ของหุ่นยนต์เท่ากัน หรือ $v_R+v_L = 0$ แล้ว หุ่นยนต์จะเคลื่อนที่ได้เป็นเส้นตรง

บทที่ 3

การออกแบบ และ การสร้าง

ชุดปฏิบัติการหุ่นยนต์ควบคุมด้วยไมโครคอนโทรลเลอร์ ดีเอสพิก 30 เอฟ 2010 (dsPIC 30F2010) ประกอบด้วย 2 ส่วน คือ ส่วนของฮาร์ดแวร์ (Hardware) และส่วนของซอฟต์แวร์ (Software) ซึ่งส่วนของซอฟต์แวร์ คือ โปรแกรมเอ็มพีแล็บ (MPLAB) สำหรับควบคุมการทำงาน โดยโหลดข้อมูลโปรแกรมจากคอมพิวเตอร์ ลงหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ ดีเอสพิก ผ่านบอร์ดดีบั๊กเกอร์ ไอซีดีเอ็กซ์ - 30 (ICDX - 30) และส่วนที่เป็นฮาร์ดแวร์นั้น ประกอบไปด้วยวงจรต่างๆโดยได้ออกแบบโครงสร้างอย่างง่าย โดยประกอบดังต่อไปนี้

1. ชุดบอร์ดดีเอสพิก (dsPIC)
2. ชุดบอร์ดดีบั๊กเกอร์ ไอซีดีเอ็กซ์ - 30 (ICDX - 30)
3. แหล่งจ่ายไฟ
4. ชุดดีซีมอเตอร์
5. ชุดขั้วมอเตอร์

3.1 การออกแบบส่วนของฮาร์ดแวร์ (Hardware)

3.1.1 ชุดบอร์ดดีเอสพิก dsPIC

มีหน้าตาของบอร์ดทดลอง (NX-dsPIC28) แสดงในรูปที่ 3-1 และบอร์ดพัฒนา (JX-dsPIC28) แสดงในรูปที่ 3-2 ส่วนวงจรที่สมบูรณ์แสดงในรูปที่ 3-3 คุณสมบัติทางเทคนิคมีดังนี้



รูปที่ 3-1 บอร์ดทดลอง (NX-dsPIC28) รูปที่ 3-2 บอร์ดพัฒนา (JX-dsPIC28)

- ใช้ได้กับไมโครคอนโทรลเลอร์ dsPIC ตัวถึง DIP ขนาด 28 ขา บนบอร์ดใช้กับ dsPIC30F2010
- ทำการโปรแกรมและดีบั๊กได้ด้วยบอร์ดดีบั๊กเกอร์ (ICDX-30) ของ i-nex หรือ ICD2 ของ Microchip
- เลือกโหมดการโปรแกรมและรันด้วยสวิตช์กดเพียงตัวเดียว พร้อมไฟแสดงสถานะ
- ความถี่สัญญาณพิกานบนบอร์ด 7.3728 เมกะเฮิร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ชุดบอร์ดดีบั๊กเกอร์ ไอซีดีเอ็กซ์ - 30

มีหน้าตาและอุปกรณ์ประกอบแสดงในรูปที่ 3-4 ส่วนคุณสมบัติทางเทคนิคมีดังนี้

- เชื่อมต่อกับคอมพิวเตอร์ทางพอร์ตอนุกรม RS-232
- ดีบั๊กเพื่อตรวจสอบการทำงานของไมโครคอนโทรลเลอร์ dsPIC ได้
- ใช้เป็นเครื่องโปรแกรมไมโครคอนโทรลเลอร์ dsPIC wfh
- ใช้งานร่วมกับ MPLAB IDE V6.2 ขึ้นไป
- มีฟังก์ชันทั้งหมดที่ต้องใช้ในการดีบั๊ก ไม่ว่าจะเป็นการกำหนดจุดหยุด การตรวจสอบ หรือ break point, การรันโปรแกรมแบบ Single step หรือแบบเต็ม พร้อมทั้งแสดงผลการทำงานด้วย
- สามารถอัปเดตเพื่อเพิ่มเติมเบอร์ไมโครคอนโทรลเลอร์ได้ด้วยตนเองผ่านทางคอมพิวเตอร์
- มี LED แสดงผลการทำงาน POWER(แสดงการจ่ายไฟเลี้ยง), BUSY(แจ้งควาพร้อมในการทำงาน) และ ERROR(แจ้งสถานะความผิดพลาด)
- สามารถอ่านและเขียนข้อมูลในหน่วยความจำข้อมูลอีพรอมภายในไมโครคอนโทรลเลอร์ได้
- ใช้ไฟเลี้ยงจากภายนอก +9V ถึง +16V
- ใช้แจ็กแบบโมดูลาร์ 6 ขา(modular jack) สำหรับต่อสายสัญญาณเพื่อติดต่อกับบอร์ดเป้าหมาย (target board) โดยมีการจัดขาตรงกับแจ็ก ICD ของ ICD2 ของ Microchip



รูปที่ 3-4 ชุดบอร์ดดีบั๊กเกอร์ ไอซีดีเอ็กซ์ - 30

3.1.3 แหล่งจ่ายไฟ

แหล่งจ่ายไฟ นับเป็นส่วนที่มีความสำคัญมาก เนื่องจากหากแหล่งจ่ายไฟไม่มีเสถียรภาพในการทำงาน ก็อาจประสบปัญหาต่างๆมากมาย ดังนั้นจึงควรให้ความสำคัญในส่วนนี้ โดยใน โครงการงานนี้ จะแบ่งแหล่งจ่ายไฟออกเป็น 2 ส่วนได้ดังนี้

1. ส่วนของไฟเลี้ยงบอร์ดชุดบอร์ดดีเอสพีค (dsPIC) แรงดัน +5 โวลต์ แสดงในรูปที่ 3-5
2. ส่วนของแหล่งจ่ายไฟแก่ชุดขับเคลื่อนมอเตอร์ แรงดัน +12 โวลต์ แสดงในรูปที่ 3-6



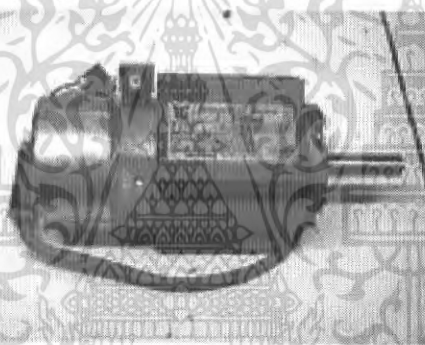
รูปที่ 3-5 แหล่งจ่ายไฟเลี้ยงบอร์ดชุดบอร์ดดีเอสพี



รูปที่ 3-6 แหล่งจ่ายไฟชุดขับเคลื่อนมอเตอร์

3.1.4 ชุดดีซีมอเตอร์

มอเตอร์ที่เลือกใช้จะเป็นมอเตอร์กระแสตรงและมีอุปกรณ์ป้องกันกลับ ดังแสดงในรูปที่ 3-7 โดยจะใช้แบบ อินครีเมนทัลเอนโค้ดเดอร์ ที่มีความละเอียดของพัลส์ถึง 3600 พัลส์ต่อ 1 รอบ เพื่อทำหน้าที่ตรวจวัดความเร็ว ทิศทางการหมุนของมอเตอร์และ ตำแหน่ง เฟลาของโรเตอร์แล้ว ส่งผลกลับไปยังคอนโทรลเลอร์เพื่อจะรับรู้ ว่ามอเตอร์กำลังหมุนด้วยทิศทาง

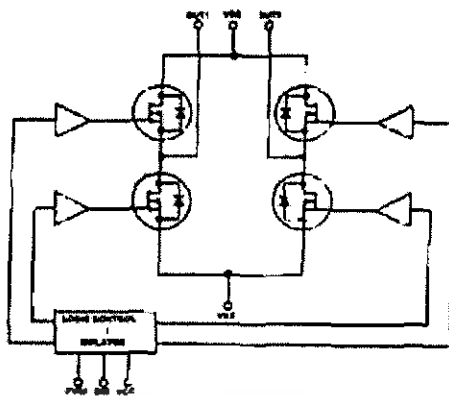


รูปที่ 3-7 แสดงมอเตอร์กระแสตรง

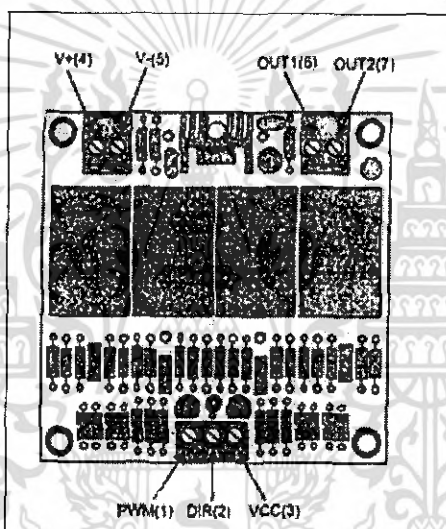
3.1.5 ชุดขับเคลื่อนมอเตอร์

เป็นชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงแบบมีแปรงถ่านที่ออกแบบมาด้วยโครงสร้าง แบบ เอชบริดจ์ (H-BRIDGE) ประกอบด้วยเพาเวอร์มอสเฟตแบบ N-Channel 4 ตัว สามารถขับ กระแสได้สูง และความต้านทานช่วงนำกระแสของมอสเฟตมีค่าต่ำมาก โดยสัญญาณควบคุมใช้แค่เพียง 3 เส้น คือ PWM VCC DIR ซึ่งสัญญาณภาคควบคุมจะแยกเด็ดขาดทางไฟฟ้าด้วย OPTOISOLATOR กับวงจรภาคกำลังสามารถควบคุมการทำงานได้ในโหมดของ Pulse Width Modulation (PWM) และในโหมดของ ON - OFF ภาคกำลังมีวงจรสับเบอร์ดป้องกันความเสียหายของมอสเฟตจากแรงดัน สูงชั่วขณะ ในช่วงมอสเฟตหยุดนำกระแส วงจรถูกออกแบบมาให้มีขนาดเล็กและใช้งานได้สะดวก สามารถประยุกต์ใช้งานได้หลากหลาย ตำแหน่งและรายละเอียดข้อต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-8 แสดงบล็อกไดอะแกรมของ H-BRIDGE DRIVE



รูปที่ 3-9 ตำแหน่งและรายละเอียดขั้วต่อ

ขั้วที่ 1 PWM สัญญาณสั่งงานการหมุนของมอเตอร์เมื่อสัญญาณที่เข้ามาเป็นลอจิก "0" จะเป็นการสั่งให้มอเตอร์ทำงาน เมื่อสัญญาณลอจิกเป็น "1" มอเตอร์จะหยุดทำงาน

ขั้วที่ 2 DIR สัญญาณควบคุมทิศทางการหมุนของมอเตอร์เมื่อสัญญาณที่เข้ามาเป็นลอจิก "0" มอเตอร์จะหมุนตามเข็มนาฬิกา เมื่อสัญญาณลอจิกเป็น "1" มอเตอร์หมุนทวนเข็มนาฬิกา

ขั้วที่ 3 VCC สัญญาณแรงดันไฟฟ้าเลี้ยงชุดควบคุม

ขั้วที่ 4 V+ ขั้วแรงดันไฟฟ้าบวกของแหล่งจ่ายไฟของมอเตอร์

ขั้วที่ 5 V- ขั้วแรงดันไฟฟ้าลบของแหล่งจ่ายไฟของมอเตอร์

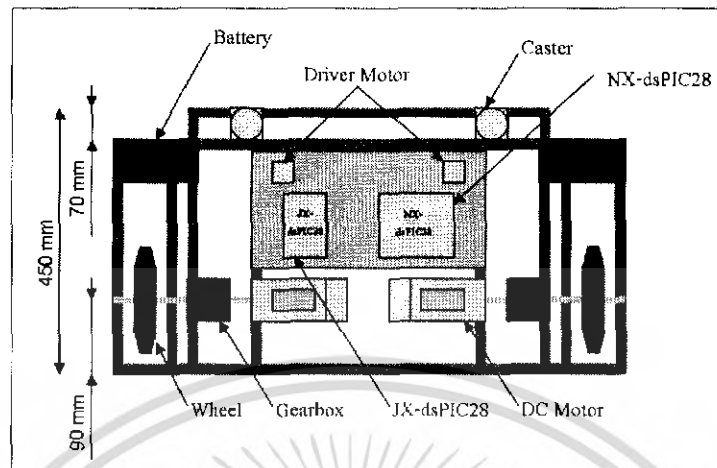
ขั้วที่ 6 OUT1 ขั้วต่อมอเตอร์

ขั้วที่ 7 OUT2 ขั้วต่อมอเตอร์

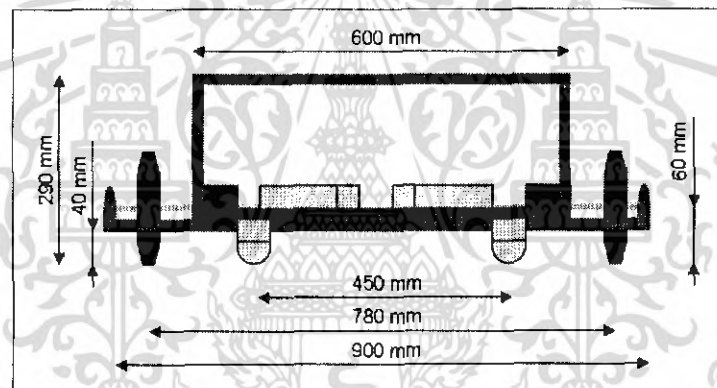
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 โครงสร้างหุ่นยนต์

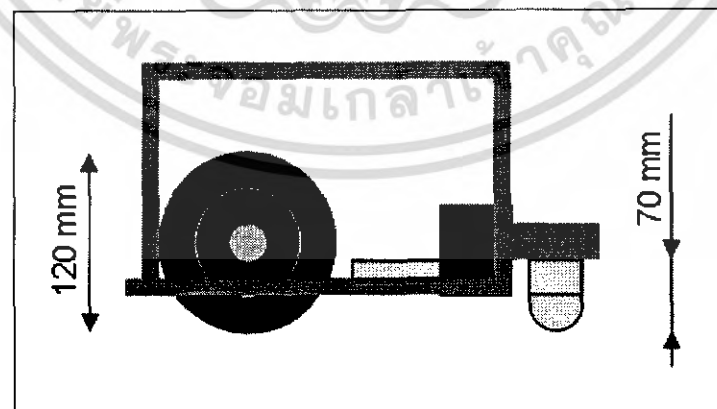
การออกแบบโครงสร้างหุ่นยนต์



รูปที่ 3-10 มุมมองโครงสร้างด้านบน



รูปที่ 3-11 มุมมองโครงสร้างด้านหน้า



รูปที่ 3-12 มุมมองโครงสร้างด้านข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบส่วนของซอฟต์แวร์ (Software)

3.2.1 การติดตั้งซอฟต์แวร์ของ MPLAB IDE และ MPLAB C30

ต้องเริ่มจากการติดตั้ง MPLAB IDE ก่อน โดยดับเบิลคลิกที่ไฟล์เซตอัป (MPLAD IDE v6.xx.exe) หลังจากนั้นจะเข้าสู่หน้าต่างขั้นตอนการติดตั้ง ให้ดำเนินการเหมือนกับการติดตั้งซอฟต์แวร์ประยุกต์บนวินโดวส์ทั่วไป จนกระทั่งเสร็จสิ้นลง หลังจากนั้นให้ติดตั้ง MPLAB C30 vxxx.exe เมื่อเสร็จสิ้นลงให้รีสตาร์ทเครื่องคอมพิวเตอร์ใหม่เพราะ MPLAB IDE จะสามารถทำงานได้อย่างสมบูรณ์ หลังจากรีสตาร์ทคอมพิวเตอร์ใหม่

3.2.2 การกำหนดการคุณสมบัติการทำงานของ MPLAB C30 ร่วมกับ MPLAB IDE

หลังจากที่ติดตั้ง MPLAB C30 เรียบร้อยแล้ว ขั้นตอนต่อไปที่จะต้องทำคือ การระบุพาธ หรือตำแหน่งที่ติดตั้งแจ้งให้กับ MPLAB IDE ทราบ เพื่อทำการเชื่อมโยงเครื่องมือสำหรับเรียก คอมไพล์จาก MPLAB C30 โดยกำหนดได้จากหัวข้อเครื่องมือ Select language Toolsuit และ Set language Tool location

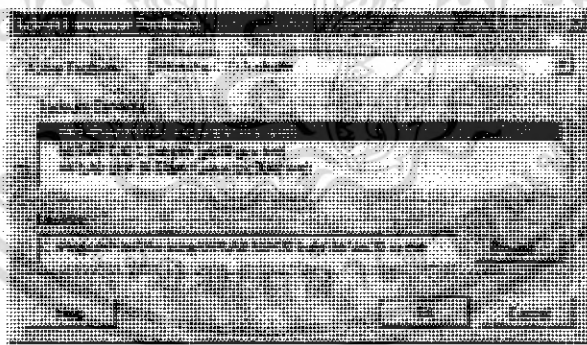
3.2.2.1 กำหนดการเชื่อมโยงเครื่องมือจาก Select language Toolsuit

ผู้พัฒนาสามารถเข้าถึงหน้าต่างการกำหนดเครื่องมือการคอมไพล์นี้โดยเลือกจากเมนูคำสั่ง Project → Select language Toolsuit... หลังจากนั้นจะปรากฏหน้าต่างให้กำหนดพาธเครื่องมือพร้อมกันนั้นผู้พัฒนาจำเป็นต้องกำหนดพาธเครื่องมือของ MPLAB C30 ตามรูปที่ 3-13

สำหรับในขั้นตอนของการกำหนดนั้นทำได้ตามขั้นตอนดังนี้

Active Toolsuit

เลือกเครื่องมือเป็น Microchip C30 Toolsuit



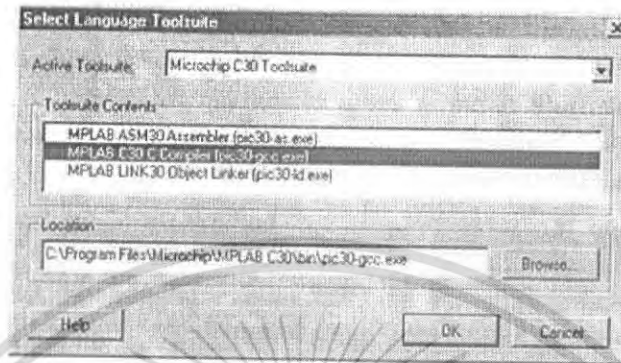
รูปที่ 3-13 แสดงการกำหนดพาธให้กับ MPLAB ASM30 Assembler(ใช้กับ MPLAB7)

Toolsuite Contents แบ่งเป็น 3 ขั้นตอน

(1) คลิกที่รายการบรรทัด MPLAB ASM30 Assembler(pic30-as.exe) แล้วทำการกำหนดพาธการมองไปยังที่ไฟล์เอ็กซีคิวทีฟที่ชื่อ pic30-as.exe ซึ่งกำหนดพาธในส่วนของ Location เป็น C:\Program File\Microchip\MPLAB ASM30 Suite\bin\pic30-as.exe ตามรูปที่ 3-13

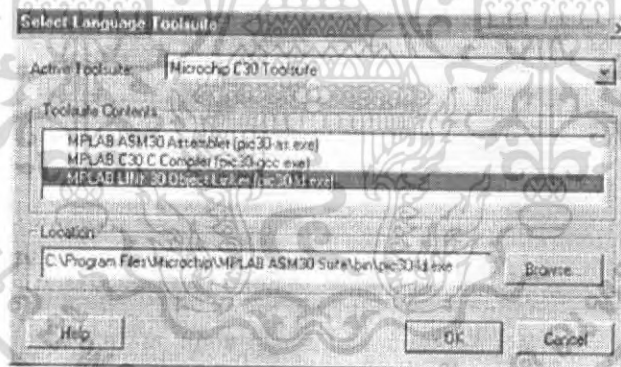
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) คลิกที่รายการบรรทัด MPLAB C30 C Compiler(pic30-gcc.exe) แล้วทำการกำหนดพารามิเตอร์ไปยังที่ไฟล์เอ็กซีกิวซ์ที่ชื่อ pic30-gcc.exe ซึ่งกำหนดพารามิเตอร์ในส่วนของ Location เป็น C:\Program File\Microchip\MPLAB ASM30 Suite\bin\pic30-gcc.exe ตามรูปที่ 3-14



รูปที่ 3-14 แสดงการกำหนดพารามิเตอร์ให้กับ MPLAB C30 C Compiler

(3) คลิกที่รายการบรรทัด MPLAB LINK30 Object Linker(pic30-ld.exe) แล้วทำการกำหนดพารามิเตอร์ไปยังที่ไฟล์เอ็กซีกิวซ์ที่ชื่อ pic30-ld.exe ซึ่งกำหนดพารามิเตอร์ในส่วนของ Location เป็น C:\Program File\Microchip\MPLAB ASM30 Suite\bin\pic30-gld.exe ตามรูปที่ 3-15



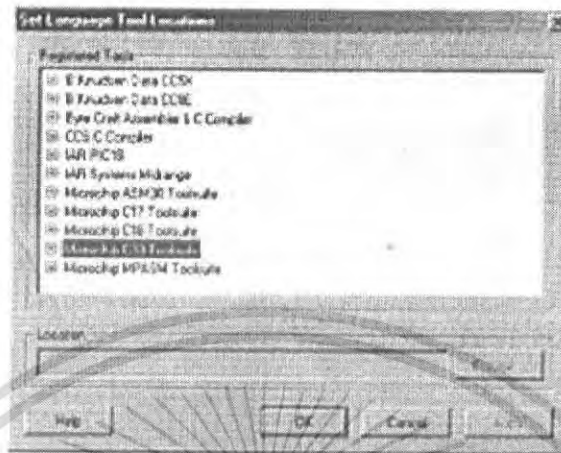
รูปที่ 3-15 แสดงการกำหนดพารามิเตอร์ให้กับ MPLAB LINK30 Object Linker

3.2.2.2 กำหนดการเชื่อมโยงเครื่องมือจาก Set language Tool Location

สำหรับในหัวข้อนี้จะคล้ายกับ Select language Toolsuite โดยมีการเพิ่มเติมในส่วนของการกำหนดค่า Default พารามิเตอร์เพื่อช่วยในการมองหาไฟล์อื่นๆที่จำเป็นในการประมวลผล ไม่ว่าจะเป็นไฟล์ไลบรารีแบบเฮดเดอร์ที่มีนามสกุลเป็น *.h, ไฟล์ไลบรารีที่มีนามสกุลเป็น *.a, ไฟล์ลิงก์เกอร์ที่มีนามสกุลเป็น *.gld ในกรณีนี้เครื่องมือคอมไพล์คือ MPLAB C30 ดังนั้นต้องเลือกกำหนดค่าคุณสมบัติของ MPLAB C30

ผู้พัฒนาสามารถเข้าถึงหน้าต่างกำหนดเครื่องมือคอมไพล์นี้โดยเลือกจากเมนูคำสั่ง Project → Set language Tool Location... หลังจากนั้นจะปรากฏหน้าต่างรายการเครื่องมือของกลุ่ม Third เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

party ที่ MPLAB IDE สนับสนุนการทำงานตามรูปที่ 3-16 และหนึ่งในนั้นจะปรากฏรายชื่อ Microchip C30 Toolsuite โดยภายในรายชื่อของรายการจะแสดงรายละเอียดของหัวข้อแยกย่อยต่อไปได้อีกเมื่อคลิกที่เครื่องหมาย + เพื่อแสดงรายการ โดยหัวข้อทั้งหมดจะกระจายรายการแสดงตามรูปที่ 3-17



รูปที่ 3-16 แสดงหน้าต่าง set language Tool Locations ที่ถูกเปิดขึ้น



รูปที่ 3-17 แสดงการกระจายรายการหน้าต่าง set language Tool Locations ทั้งหมด

สำหรับในขั้นตอนของการกำหนดนั้นทำได้ตามขั้นตอนดังนี้

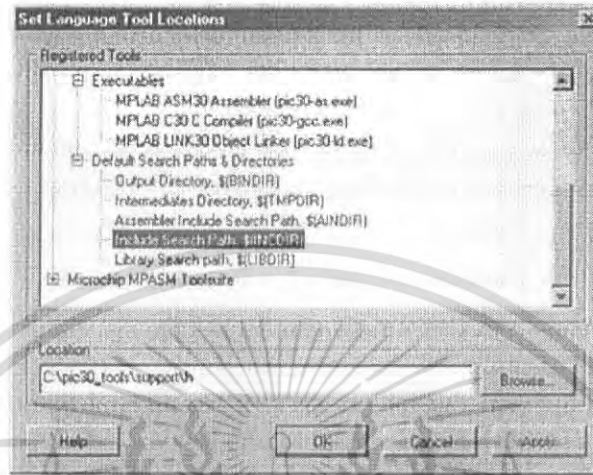
Executables สำหรับรายการบรรทัด MPLAB ASM30 Assembler(pic30-as.exe), MPLAB C30 C Compiler(pic30-gcc.exe) และ MPLAB LINK30 Object Linker(pic30-ld.exe) จะมีการกำหนดพาธเช่นเดียวกับการกำหนดใน Select language Toolsuite สำหรับหัวข้อก่อนหน้านี้ ซึ่งจะเป็นการแชร์ลิงก์กันระหว่างหัวข้อทั้งสอง กล่าวคือสามารถเปลี่ยนแปลงพาธการเชื่อมโยงของส่วนกลางหรือ default ที่ตั้งค่าไว้ได้จากทั้ง 2 หัวข้อเลย

Default Search Parts & Directories มี 2 ส่วนที่สำคัญสำหรับการกำหนดพาธ

(1) คลิกที่รายการบรรทัด Include Search Part, \$(INCDIR) สำหรับการกำหนดพาธที่เก็บไฟล์ไลบรารีแบบเฮดเดอร์ที่มีนามสกุลเป็น *.h เพื่อกำหนดให้คอมไพเลอร์ และลิงก์เกอร์

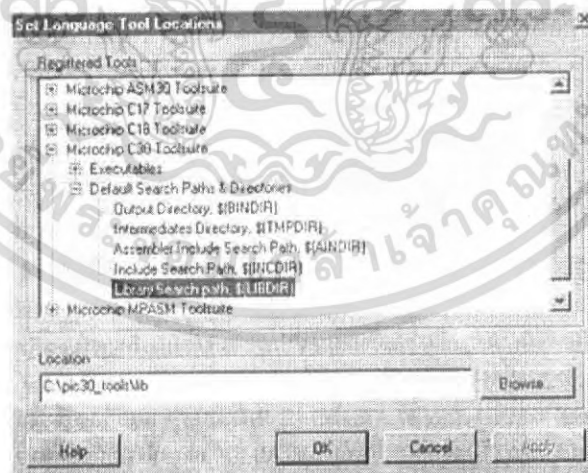
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ้อมมองเห็นไฟล์ไลบรารีที่บรรจุอยู่ในพารนี้และสามารถเชื่อมโยงได้ถูกต้อง ในระหว่างการ คอมไพล์ แต่สำหรับในขั้นต้นแนะนำว่าควรบรรจุพาร C:\pic30_tools\support\h ตามรูปที่ 3-18 ซึ่งเป็น พารที่เก็บ dsPIC Peripheral Library ทั้งหมดที่มาพร้อมกับ MPLAB C30 และหลังจากนั้นสามารถ กำหนด พารเพิ่มเติมได้อีกด้วยการค้นเครื่องหมาย ; แล้วตามด้วยพารที่ต้องการกำหนด



รูปที่ 3-18 แสดงการกำหนดพารให้กับ Include Search Part, \$(INCDIR)

(2) คลิกที่รายการบรรทัด Library Search Part, \$(LIBDIR) สำหรับการ กำหนดพารที่เก็บไฟล์ไลบรารีที่มีนามสกุลเป็น *.lib เพื่อกำหนดให้คอมไพล์เลอร์มองเห็น ไฟล์ไลบรารีที่ บรรจุอยู่ในพารนี้และสามารถเชื่อมโยงได้ถูกต้องในระหว่างการคอมไพล์ โดยบรรจุ ไฟล์ pic30_tool\lib ตามรูปที่ 3-19 ซึ่งเป็นพารที่เก็บไลบรารีที่มาพร้อมกับ MPLAB C30



รูปที่ 3-19 แสดงการกำหนดพารให้กับ Library Search Part, \$(LIBDIR)

3.2.3 การสร้างโปรเจกต์ภาษา C ด้วย Project Wizzard ของ MPLAB

หลังจากทำการติดตั้ง MPIDE LAB และ MPLAB C30 Compiler รวมทั้งจัดการคุณสมบัติ เชื่อมโยงการทำงานระหว่าง 2 โปรแกรมดังกล่าว ให้ทำงานร่วมกันเป็นที่เรียบร้อยแล้ว ในหัวข้อนี้จะ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวถึงวิธีสร้างไฟล์เฮดไฟล์นามสกุล .hex สำหรับการควบคุมการทำงานของไมโครคอนโทรลเลอร์ ซึ่งต้องผ่านขั้นตอนการสร้างไฟล์ในรูปแบบโปรเจกต์หรือผังงานซึ่งเป็นรูปแบบที่ MPLAB ขอมให้ผู้พัฒนาสามารถผนวกไฟล์ชนิดต่างๆที่เกี่ยวข้องในการพัฒนาเข้าด้วยกัน ไม่ว่าจะเป็นไลบรารี, ไฟล์ผนวก (include) ต่างๆที่มาพร้อมกับ MPLAB C30 Compiler หรืออาคารสร้างด้วยตัวผู้พัฒนาเองก็ตาม โดยจะได้รับการประมวลผลและเชื่อมโยงด้วย MPLAB IDE และ MPLAB C30 Compiler ในขั้นตอนของการคอมไพล์ จนท้ายที่สุดจะได้ไฟล์ผลลัพธ์คือไฟล์ *.hex ออกมาเพื่อนำไปดาวน์โหลดลงในหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ dsPIC เพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ต่อไป

การสร้างโปรเจกต์ภาษา C ด้วย Project Wizard ของ MPLAB มีขั้นตอนดังนี้

1) เปิดโปรแกรม MPLAB IDE เลือกเมนู File > New เพื่อสร้างไฟล์ใหม่เพื่อที่จะเขียนโปรแกรมภาษาซี หลังจากนั้นจะปรากฏหน้าต่าง MPLAB Editor ซึ่งว่างเปล่าพร้อมสำหรับการกำหนดคำสั่งที่ต้องการตามรูปที่ 3-20



รูปที่ 3-20 แสดง MPLAB Editor สำหรับการสร้างไฟล์ใหม่

2) กำหนดชุดคำสั่งภาษาซี เพื่อใช้ทดสอบตามรูปที่ 3-21 แล้วทำการบันทึกชื่อไฟล์เป็น first.c และเพื่อความเป็นระเบียบความสร้างไฟล์เดอร์ชื่อเดียวกับไฟล์

ชุดคำสั่งภาษาซี ของไฟล์ first.c

```
#include <p30F2010.h>
```

```
int counter;
```

```
int main (void)
```

```
{
```

```
    counter = 1;
```

```
    TRISB = 0;
```

```
    While(1)
```

```
    {
```

```
        PORTB = counter;
```

```
        counter++;
```

```
    }
```

```
    return 0;
```

```
}
```

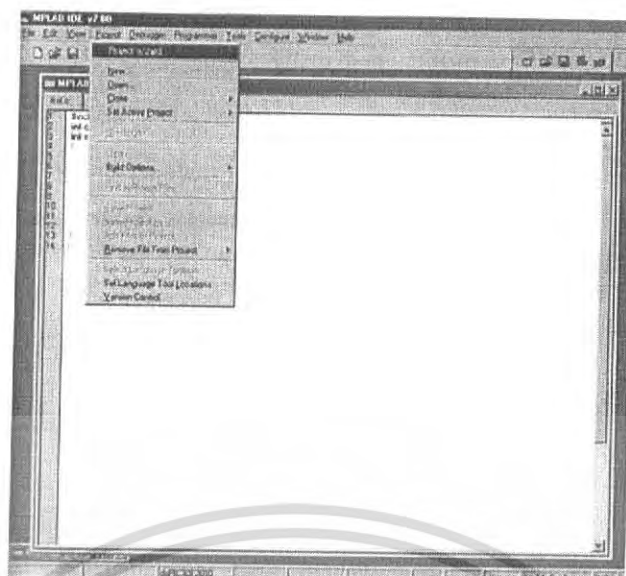


รูปที่ 3-21 แสดงการจัดเก็บไฟล์ first.c ไว้ภายในโฟลเดอร์ first (ผู้พัฒนาต้องสร้างเอง)

หลังจากได้สร้างไฟล์โปรแกรมภาษาซี ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์แล้ว ขั้นตอนในลำดับต่อไปจะต้องสร้างโปรเจกต์ขึ้นมาให้ทำการผนวกไฟล์ first.c เข้ากับโปรเจกต์

3) ที่โปรแกรม MPLAB IDE เลือก Project → Project Wizard ตามรูปที่ 3-22 ซึ่งหน้าต่าง Project Wizard จะถูกเปิดขึ้นมาตามรูปที่ 3-23 หลังจากนั้นคลิกปุ่ม Next เพื่อเข้าสู่ขั้นตอนการกำหนดคุณสมบัติของโปรเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



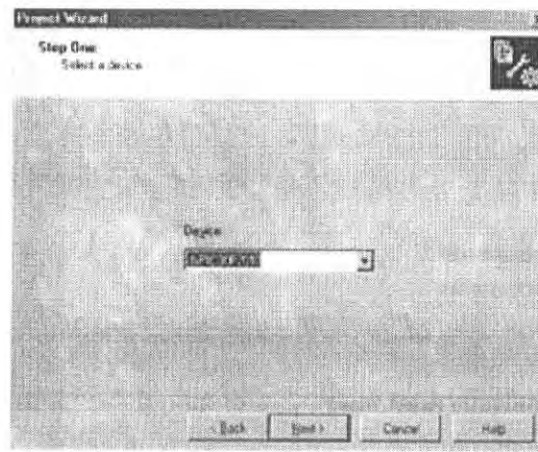
รูปที่ 3-22 แสดงเมนูการสร้างโปรเจกต์แบบ Project Wizard



รูปที่ 3-23 แสดงหน้าต่าง Project Wizard ที่ถูกเปิดขึ้นมาใหม่

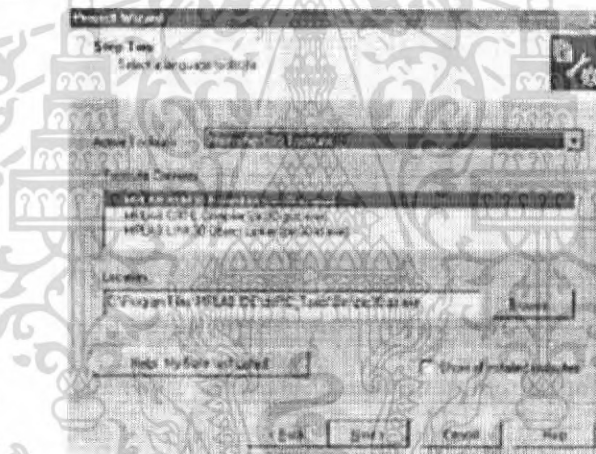
- 4) หลังจากนั้นจะเข้าสู่หัวข้อ Select a device เพื่อกำหนดบอร์ดไมโครคอนโทรลเลอร์ โดยกำหนดให้เป็นบอร์ด dsPIC2010 จากนั้นคลิกปุ่ม Next เพื่อเริ่มเข้าสู่ขั้นตอนต่อไป ตามรูปที่ 3-24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



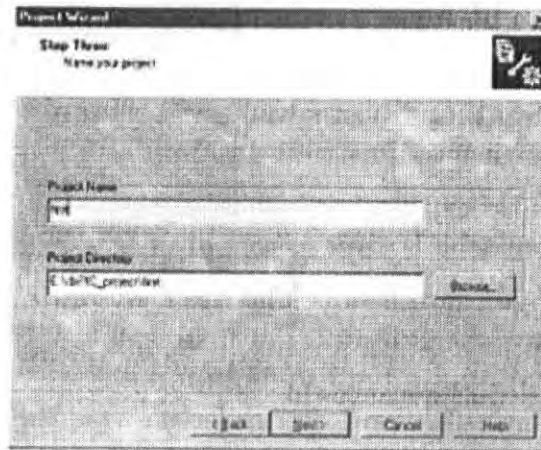
รูปที่ 3-24 แสดงหน้าต่างหัวข้อ Select a device

5) หลังจากนั้นจะเข้าสู่หัวข้อ Select a language toolsuite เพื่อกำหนดภาษาและเครื่องมือ ที่ใช้ในการพัฒนา ตามรูปที่ 3-25 ซึ่งในหัวข้อ Active toolsuite ให้เลือกเป็น Microchip C30 Toolsuite หลังจากนั้นให้คลิกปุ่ม Next เพื่อเริ่มเข้าสู่ขั้นตอนถัดไป



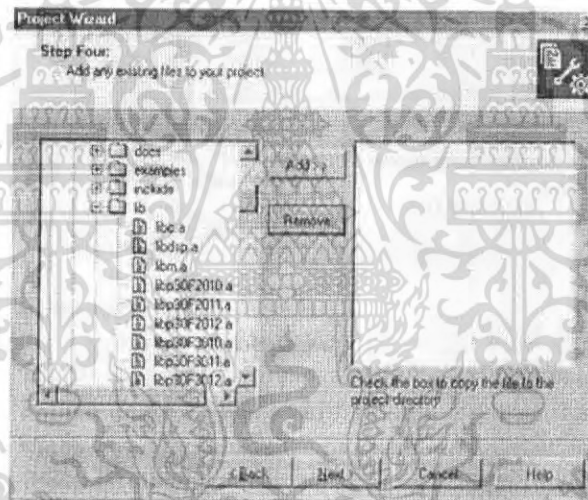
รูปที่ 3-25 แสดงหน้าต่างหัวข้อ Select a language toolsuite

6) หลังจากนั้นจะเข้าสู่หัวข้อ Name your project เพื่อกำหนดชื่อ โปรเจกต์และตำแหน่งพาท ในการจัดเก็บกลุ่มไฟล์ผลลัพธ์จากโปรเจกต์ที่พัฒนา ตามรูปที่ 3-26 ซึ่งในหัวข้อ Project Name ให้กำหนดชื่อเป็น first ส่วนในหัวข้อ Project Directory ให้กำหนดเป็นพาทของ โพลเดอร์ชื่อ first ที่จัดเก็บไฟล์โปรเจกต์นี้หรือคลิกปุ่ม Browse เพื่อกำหนดพาทแบบเลือกเส้นทางจากเมนูเริ่มต้นไม่ หลังจากนั้นให้คลิกปุ่ม Next เพื่อเข้าสู่ขั้นตอนถัดไป



รูปที่ 3-26 แสดงหน้าต่างหัวข้อ Name your project

7) หลังจากนั้นจะเข้าสู่หัวข้อ Add any existing file to your project เพื่อเลือกไฟล์ต่างๆ ที่เกี่ยวข้องในการเชื่อมโยงการทำงานของโปรเจกต์ ตามรูปที่ 3-27 ซึ่งกรอบทางด้านขวามือจะแสดง ไฟล์ที่ถูกผนวกเข้ากับโปรเจกต์โดยเลือกจากกรอบด้านซ้ายมือ



รูปที่ 3-27 แสดงหน้าต่างหัวข้อ Add any existing files to your project

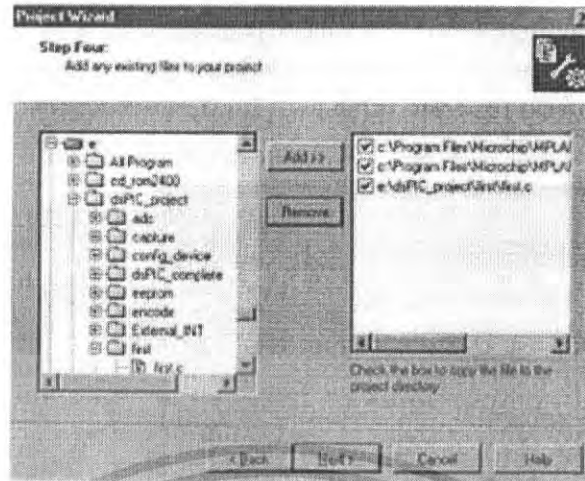
ในที่นี้ให้ทำการ Add ไฟล์ทั้งสิ้น 3 ไฟล์ดังนี้

7.1) จากกรอบซ้ายมือเข้าสู่พาธ c:\pic30_tools\lib แล้วทำการดับเบิลคลิกเลือกไฟล์ libpic30f2010.a หลังจากนั้นไฟล์นี้จะเข้าไปอยู่ทางกรอบขวามือ ให้ยืนยันการเลือกโดยการคลิกเครื่องหมายที่กรอบสี่เหลี่ยม ตามรูปที่ 3-28

7.2) จากกรอบด้านซ้ายเข้าสู่พาธ c:\pic30_tools\support\gld แล้วทำการเลือกไฟล์ libpic30f2010.gld โดยวิธีการเดียวกับข้อ 7.1)

7.3) จากกรอบด้านซ้ายเข้าสู่พาธ E:\dsPIC_project\first แล้วทำการเลือกไฟล์ first.c โดยวิธีการเดียวกับข้อ 7.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-28 แสดงไฟล์ที่ถูกเลือกเพื่อผนวกเข้ากับโปรเจกต์ first ที่กรอบด้านขวามือ

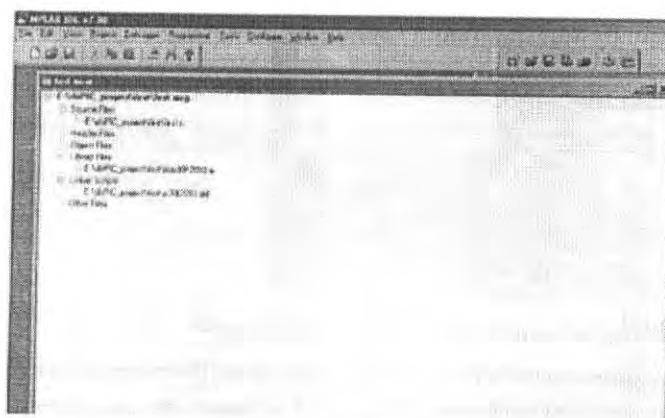
8) หลังจากนั้นจะปรากฏหน้าต่างแสดงรายละเอียดของเครื่องมือที่ใช้และพารามิเตอร์ของโปรเจกต์ที่ได้กำหนดไว้ทำการกำหนดไว้ หลังจากนั้นให้คลิกที่ปุ่ม Finish ซึ่งในขั้นตอนนี้เป็นลำดับสุดท้ายของการสร้างโปรเจกต์ด้วย Project Wizard



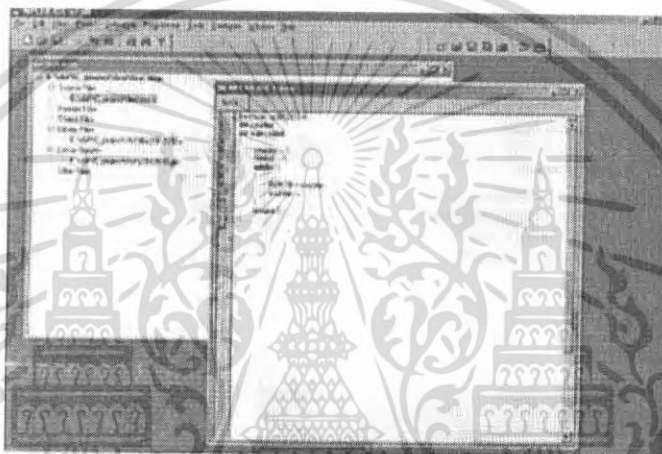
รูปที่ 3-29 แสดงหน้าต่างรายละเอียดของเครื่องมือที่ใช้และพารามิเตอร์ของโปรเจกต์

9) หลังจากนั้นจะปรากฏหน้าต่างผังงานของโปรเจกต์ first ตามรูปที่ 3-30 โดยการแสดงรายละเอียดของการผนวกไฟล์ทั้ง 3 ที่ถูกเลือกไว้ในขั้นตอนก่อนหน้านี้เข้าไปในแต่ละส่วนของตัวผังงาน หลังจากนั้นให้ดับเบิลคลิกที่พารามิเตอร์ของ first.c เพื่อเรียกไฟล์ first.c ขึ้นมาสำหรับตรวจสอบและพัฒนาต่อ ซึ่งจะปรากฏหน้าต่างไฟล์ first.c ขึ้นมาตามรูปที่ 3-31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



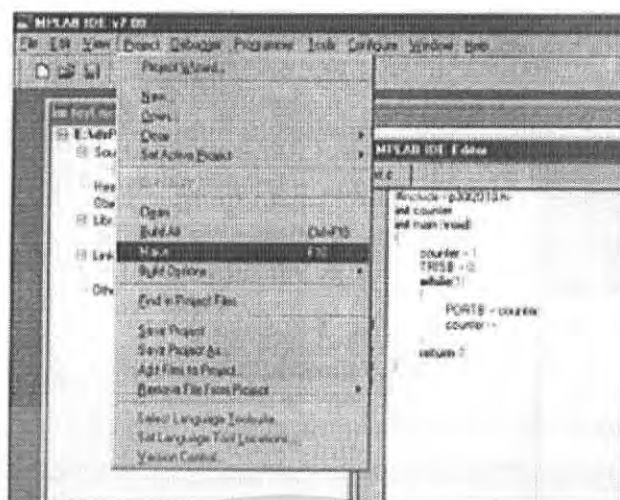
รูปที่ 3-30 แสดงหน้าต่างฝั่งงานของโปรเจกต์ *first*



รูปที่ 3-31 แสดงหน้าต่างโปรแกรมของไฟล์ *first.c* ที่ถูกเปิดขึ้นจากโปรเจกต์ *first*

- 10) หลังจากนั้นทำการคอมไพล์เพื่อให้ได้ไฟล์ผลลัพธ์เป็น *first.hex* โดยที่ไฟเลือกที่เมนู Project → Make แสดงตามรูปที่ 3-32 หรือกดปุ่ม F10 เมื่อการคอมไพล์เป็นไปอย่างสมบูรณ์ที่หน้าต่าง Output จะรายงานชื่อไฟล์ผลลัพธ์ *.hex ที่สร้างขึ้นพร้อมกับขนาดของหน่วยความจำที่ต้องใช้สำหรับไมโครคอนโทรลเลอร์ แสดงตามรูปที่ 3-33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-32 แสดงการเลือกเมนูคำสั่งคอมไพล์โปรเจกต์



รูปที่ 3-33 แสดงหน้าต่าง Output สำหรับรายงานผลการคอมไพล์

11) หลังจากทำการคอมไพล์โปรเจกต์ first เป็นไฟล์อย่างสมบูรณ์สำหรับในขั้นตอนที่ผ่านมา ไฟล์ first.hex ได้ถูกสร้างขึ้นพารเดียวกับโปรเจกต์เรียบร้อยแล้วเพียงแค่นำไฟล์ดังกล่าวนี้ไปดาวน์โหลดลงในไมโครคอนโทรลเลอร์ จากนั้นไมโครคอนโทรลเลอร์จะทำงานตามที่ได้ออกแบบ โปรแกรมไว้แสดงตามรูปที่ 3-34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



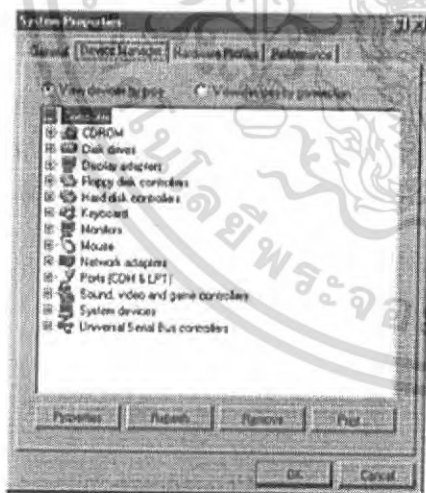
รูปที่ 3-34 แสดงไฟล์ first.hex ที่ถูกสร้างขึ้นให้อยู่ในพารเดียวกับโปรเจกต์

3.2.4 การใช้งาน MPLAB IDE กับ ICDX-30

3.2.4.1 การปรับแต่งคอมพิวเตอร์ก่อนการใช้งาน ICDX-30

1) ปรับแต่งการทำงานของพอร์ตอนุกรมที่ใช้ในการติดต่อกับ ICDX-30 โดยการยกเลิกการตั้งค่าของ FIFO ของพอร์ตอนุกรม ซึ่งช่วยลดข้อผิดพลาดในการสื่อสารข้อมูล ระหว่างคอมพิวเตอร์กับ ICDX-30 โดยการคลิกปุ่มเมาส์ขวาที่ My Computer แล้วเลือกการ Properties จะปรากฏหน้าต่าง Properties ตามรูปที่ 3-35

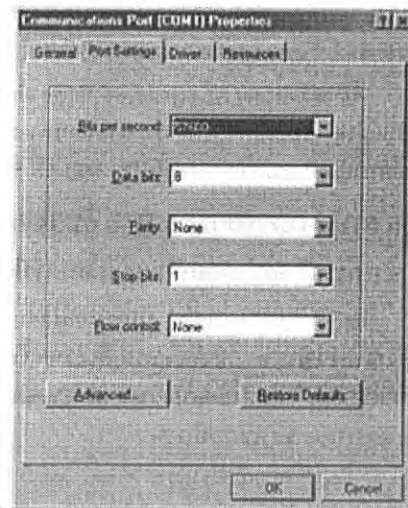
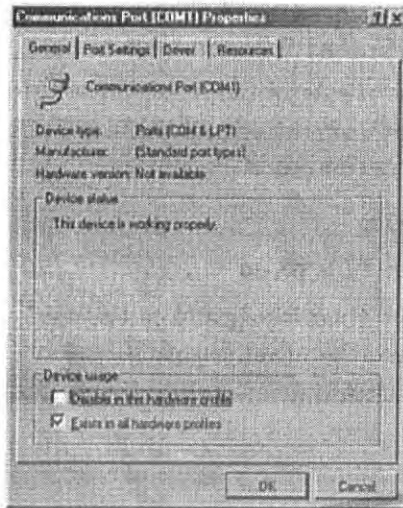
2) คลิกเลือกการ Properties Ports(COM&LPT) จะพบรายการ Communications Port(COMx) ทำการดับเบิลคลิกเลือกที่รายการนี้ตามรูปที่ 3-36



รูปที่ 3-35 แสดงหน้าต่าง System Properties รูปที่ 3-36 แสดงหน้าต่าง ของ System Properties

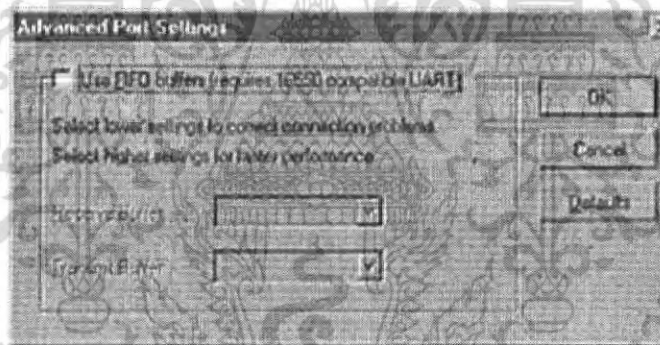
3) ปรากฏหน้าต่างตามรูปที่ 3-37 ซึ่งมีแท็บหัวข้อต่างๆ ให้เลือกหลายรายการ ให้เลือกที่แท็บ Port Setting จะปรากฏรายการต่างๆ แสดงตามรูปที่ 3-38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-37 หน้าต่าง Communications Port Properties รูปที่ 3-38 แสดงรายการของ Port Settings

- 4) ที่หัวข้อ Bits per second ให้เลือกเป็น 57600
- 5) คลิกที่ปุ่ม Advance... ด้านล่าง จะปรากฏหน้าต่างตามรูปที่ 3-39 โดยให้คลิกยกเลิกรายการ Use FIFO buffers(requires 165500 compatible UART)

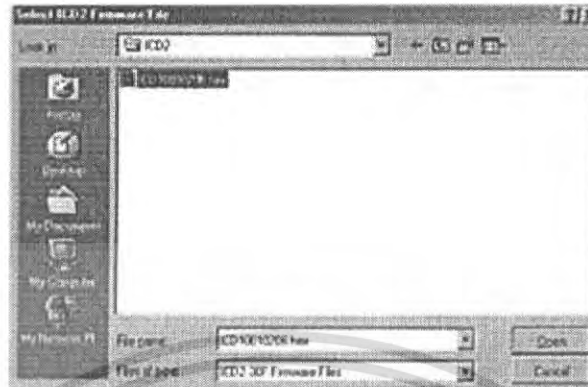


รูปที่ 3-39 แสดงหน้าต่าง Advance port setting

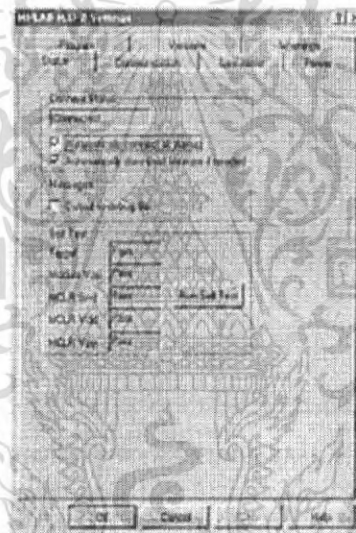
- 6) หลังจากนั้นให้ยืนยันการกำหนดค่าต่างๆด้วยการคลิกปุ่ม OK ทุกๆหน้าต่าง หลังจากนั้นให้ทำการรีสตาร์ทคอมพิวเตอร์เพื่อเริ่มต้นการทำงานใหม่
- 7) เมื่อบูตคอมพิวเตอร์ใหม่เสร็จเรียบร้อยแล้ว ให้เชื่อมต่อ ICDX-30 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ แล้วจ่ายไฟเลี้ยงให้แก่ ICDX-30
- 8) เปิดโปรแกรม MPIDE LAB ไปที่เมนู Programmer → Download ICD2 Operating System เพื่อดาวน์โหลดเฟิร์มแวร์(firmware) สำหรับจัดการระบบดาวินโหนดของ ICDX-30 ซึ่งจะปรากฏหน้าต่างให้เลือกไฟล์ โดยเลือกที่ไฟล์ ICD10010203.hex ตามรูปที่ 3-40 โดยในขั้นตอนนี้จะใช้ช่วงเวลาช่วงหนึ่งสำหรับการติดตั้งเฟิร์มแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) หลังจากนั้นไปที่เมนู Programmer → Setting... เพื่อกำหนดคุณสมบัติการทำงาน บางอย่างของ ICDX-30 โดยภายหลังการเลือกจะปรากฏหน้าต่าง ICD2 ตามรูปที่ 3-41



รูปที่ 3-40 แสดงหน้าต่างเลือกไฟล์ ICD10010206.hex



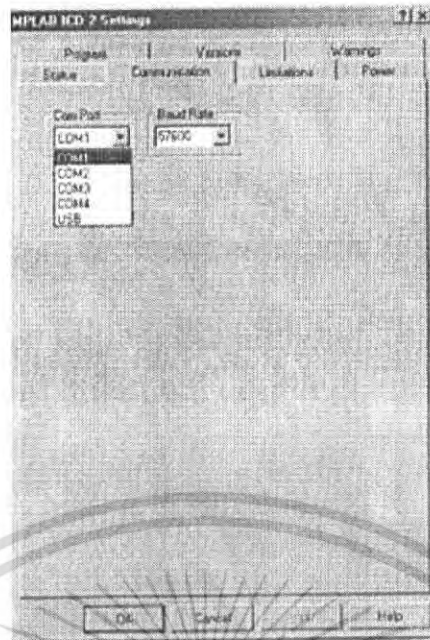
รูปที่ 3-41 แสดงหน้าต่าง ICD2 Setting

10) หลังจากนั้นเลือกที่แท็บ Communication จะปรากฏรายละเอียดตามรูปที่ 3-42

- Com port ให้กำหนดเป็น Com Port ตำแหน่งที่อยู่อยู่กับ ICDX-30 หรือ ตำแหน่งที่วางพร้อมใช้งาน
- Baud Rate ให้กำหนดเป็น 57600

หลังจากนี้ ICDX-30 ก็พร้อมใช้งานแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-42 แสดงหน้าต่างรายละเอียดของแท็บ Communication

3.2.4.2 ขั้นตอนการดาวน์โหลดไฟล์เอาต์พุต .hex ด้วย ICDX-30

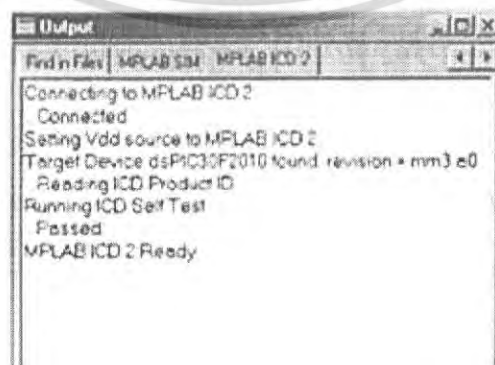
(1) เชื่อมต่อบอร์ด ICDX-30 เข้ากับบอร์ด dsPIC

(1.1) เชื่อมต่อสาย ICD ระหว่าง ICDX-30 กับบอร์ด dsPIC

(1.2) เชื่อมต่อสายระหว่างพอร์ตอนุกรมของคอมพิวเตอร์กับ ICDX-30

(1.3) จ่ายไฟให้กับ ICDX-30 LED ที่ตำแหน่ง PWR(สีเขียว) ของ ICDX-30 จะติดสว่าง

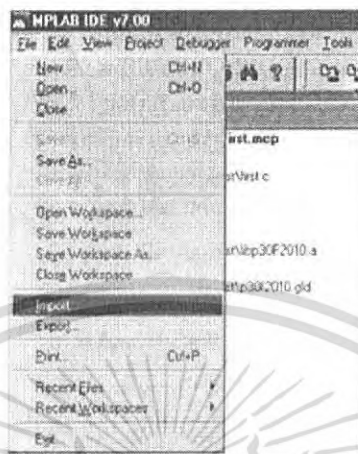
(2) ทดสอบการดาวน์โหลดไฟล์ *.hex ลงยังไมโครคอนโทรลเลอร์ dsPIC ที่ติดตั้งอยู่บน บอร์ด dsPIC โดยเปิดโปรเจกต์ของไฟล์ที่ต้องการดาวน์โหลดขึ้นมา ซึ่งปกติเป็น ไฟล์นามสกุล .mcp ให้สังเกตหน้าต่าง Output ในแท็บของ MPLAB ICD2 จะแสดงสถานะ ของการเชื่อมต่อ โดยถ้าการเชื่อมต่อถูกต้องจะปรากฏข้อความแสดงตามรูปที่ 3-43 ถ้าหากไม่มี ข้อความใดๆให้ทำการเชื่อมต่อโดยไปที่เมนู Programmer → Connect



รูปที่ 3-43 แสดงหน้าต่าง Output ที่แจ้งสถานะการเชื่อมต่อพร้อมทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) ทำการ Import ไฟล์ first.hex เพื่อเตรียมดาวน์โหลดลงสู่บอร์ดพัฒนาที่ต่อกับ ICDX-30 โดยไปที่เมนู File → Import ... ตามรูปที่ 3-44 แล้วเลือกไฟล์ .hex ในตำแหน่งที่ผู้พัฒนาจัดเก็บไว้



รูปที่ 3-44 แสดงการเลือกเมนู File → Import ...

(4) หลังจากนั้นคลิกที่ปุ่ม Program Target device เพื่อทำการดาวน์โหลด หรือไปที่เมนู Programmer → program โดยในระหว่างขั้นตอนการดาวน์โหลด LED ตำแหน่ง BUSY (สีเขียว) ของ ICDX-30 จะกระพริบ จนกระทั่งการดาวน์โหลดสิ้นสุดลง และพร้อมกันนั้นที่หน้าต่าง Output ในแท็บของ MPLAB ICD2 จะแสดงสถานะขั้นตอนการดาวน์โหลดตามรูปที่ 3-45



รูปที่ 3-45 แสดงขั้นตอนการดาวน์โหลด

(5) ถ้ามีข้อผิดพลาดในการดาวน์โหลดเกิดขึ้น จะมีข้อความแจ้งที่หน้าต่างเอาท์พุทเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

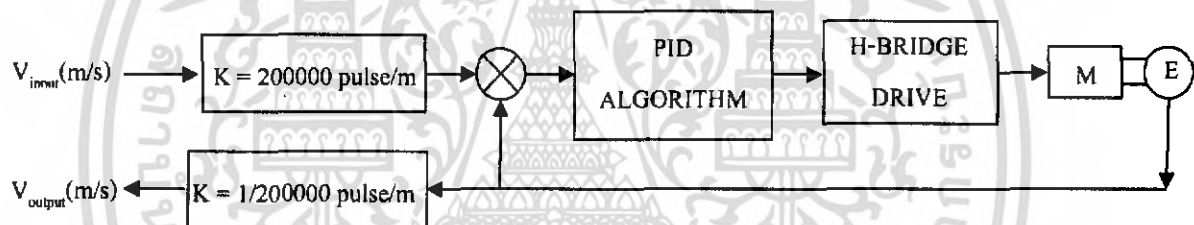
การทดลองและผลการทดลอง

จากที่ได้กล่าวมาแล้วในบทที่ 3 เกี่ยวกับการสร้างและการออกแบบ ทางด้านฮาร์ดแวร์และซอฟต์แวร์ ในบทนี้จะเป็นการทดลองและผลการทดลองที่ได้ทำการออกแบบไว้ ดังนั้นในการทดลอง จึงแยกย่อยออกเป็นการทดลองดังนี้

4.1 การทดลองจูนค่าพีไอดีและการทดสอบการทำงานของพีไอดี

4.1.1 ลำดับขั้นการทดลอง

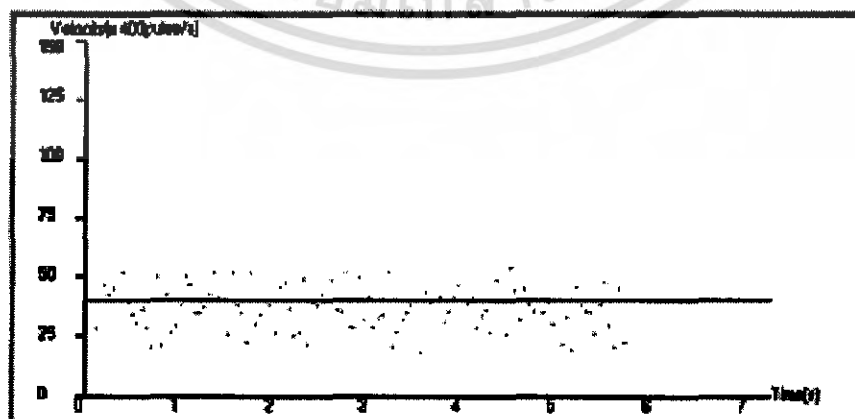
1. ทำการเชื่อมต่อพอร์ทอนุกรมกับชุดบอร์ด
2. จ่ายแรงดันให้แก่บอร์ดและชุดขับเคลื่อน
3. ทำการดาวน์โหลดโปรแกรมลงบนไมโครคอนโทรลเลอร์
4. รันโปรแกรม
5. ทำการทดลองจูนค่าพีไอดี และ ทดสอบการทำงานของค่าพีไอดี



รูปที่ 4-1 บล็อกไดอะแกรมระบบควบคุม

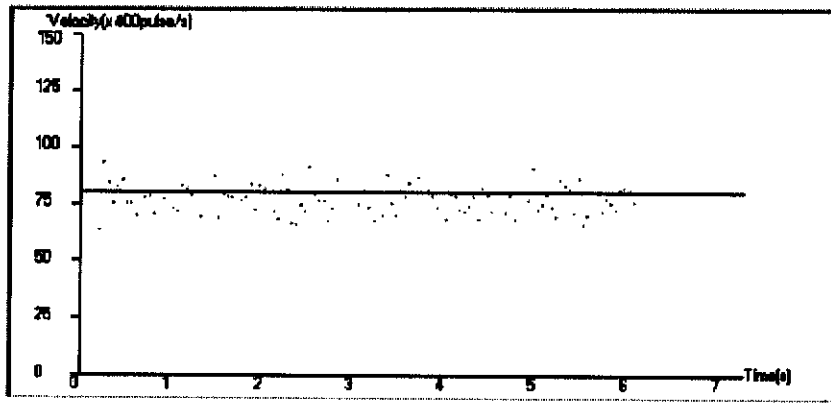
4.1.2 ผลการทดลอง

เมื่อทำการทดลองจูนค่าพีไอดีแก่ระบบ ซึ่งทำการทดลองที่ค่าความเร็วคงที่แบบต่างๆ และ ทำการทดสอบการทำงานของระบบได้ดังนี้

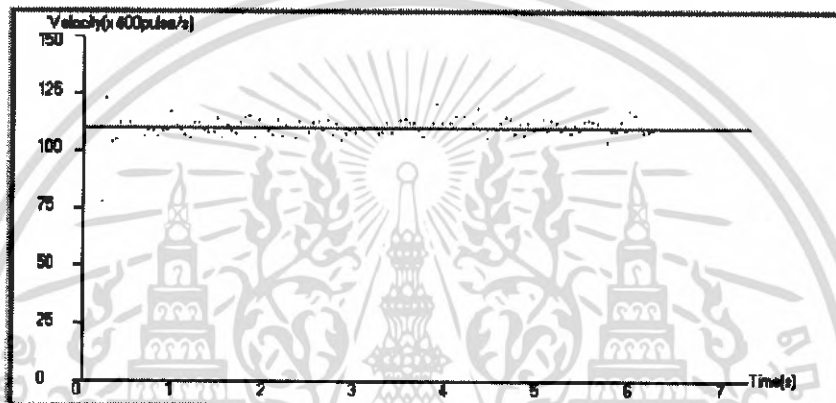


รูปที่ 4-2 ลักษณะผลตอบสนองของพีไอดีที่ความเร็วคงที่ 266.67 รอบ/นาที

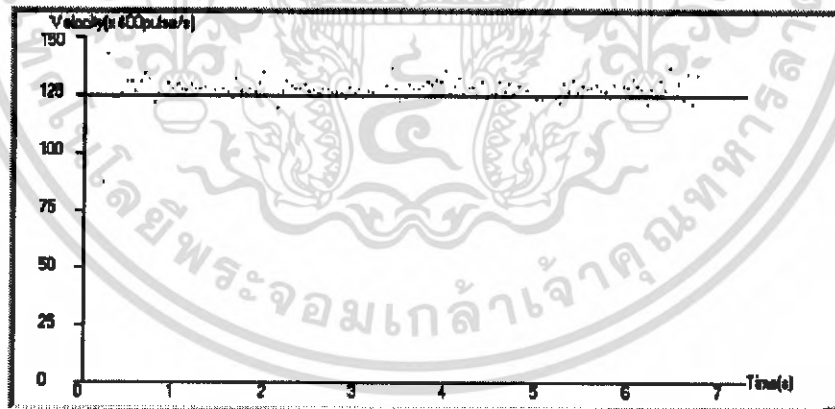
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-3 ลักษณะผลตอบสนองของฟิโอดีที่ความเร็วงที่ 533.33 รอบ/นาที



รูปที่ 4-4 ลักษณะผลตอบสนองของฟิโอดีที่ความเร็วงที่ 733.33 รอบ/นาที

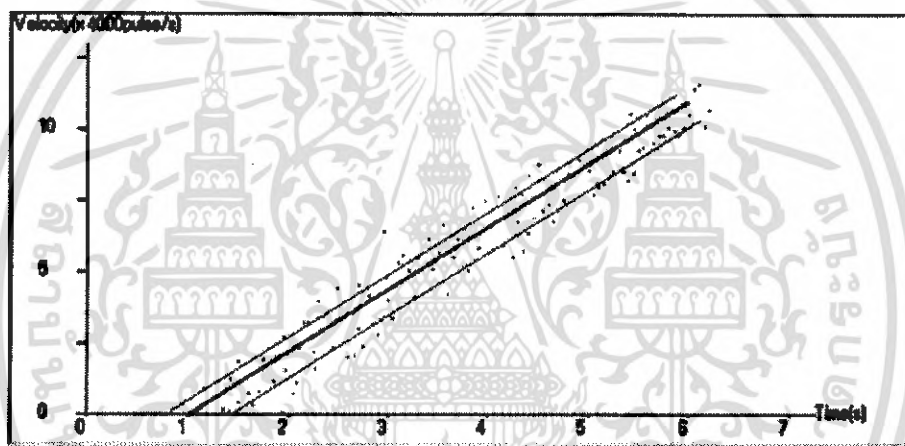


รูปที่ 4-5 ลักษณะผลตอบสนองของฟิโอดีที่ความเร็วงที่ 1030.33 รอบ/นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P = 1.7 I = 0.5 D = 10E-11	Velocity = 266.67 รอบ/นาทีย (ที่ 16000 pulse/s)	Velocity = 533.33 รอบ/นาทีย (ที่ 32000 pulse/s)	Velocity = 733.33 รอบ/นาทีย (ที่ 44000 pulse/s)	Velocity = 1033.33 รอบ/นาทีย (ที่ 62000 pulse/s)
Rise Time (ms)	50	50	50	50
Setting Time(ms)	450	450	550	550
Peak Time (ms)	100	100	100	100
% OS	16.25	13.75	11.82	14
Steady State	14000	30400	43400	51000
Error(%)	12.5	5	1.36	2

ตารางที่ 4-1 ผลการเปรียบเทียบของผลตอบสนองที่ค่าความเร็วต่างๆ



รูปที่ 4-6 ผลตอบสนองของพีไอดีเมื่อทำการเพิ่มความเร็วจนถึง 733.33 รอบ/นาทีย

จากผลการทดลองลักษณะของการแทรกความเร็วของพีไอดีนั้น โดยทำการเพิ่มความเร็วจนถึง 733.33 รอบ/นาทีย จะมีค่าความผิดพลาดอยู่ในช่วง +2000 พัลส์ ถึง -4000 พัลส์ จากแกนอ้างอิง

4.2 การทดสอบการเคลื่อนที่ของหุ่นยนต์ในระยะทางสั้น

4.2.1 ลำดับขั้นการทดลอง

1. ทำการเชื่อมต่อพอร์ทอนุกรมกับชุดบอร์ด
2. จ่ายแรงดันให้แก่บอร์ดและชุดขับเคลื่อน
3. ทำการดาวน์โหลดโปรแกรมลงบนไมโครคอนโทรลเลอร์
4. รันโปรแกรม
5. ทำการทดสอบการเคลื่อนที่ของหุ่นยนต์แบบเส้นตรงและแบบเลี้ยว

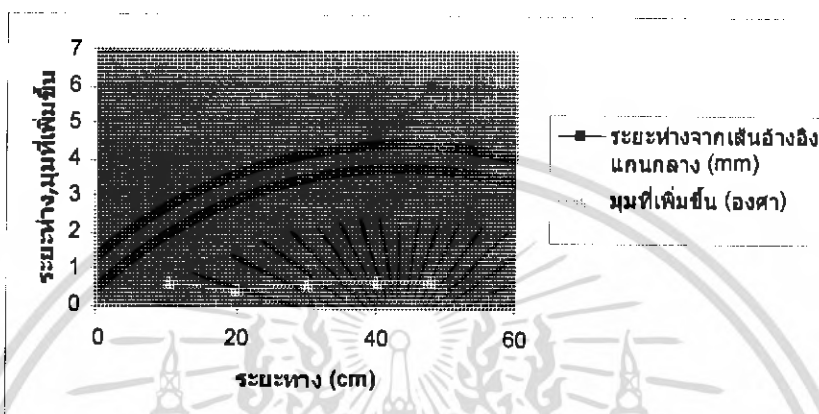
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดลอง

เมื่อทำการทดสอบการเคลื่อนที่ของหุ่นยนต์ ทั้งแบบเส้นทางตรงและแบบเลี้ยวในโหมดโปรแกรมเส้นทางและโหมดจำเส้นทาง ได้ผลการทดลองดังนี้

4.2.2.1 การทำงานในโหมดโปรแกรมเส้นทาง

การเคลื่อนที่เส้นทางตรงที่ 50 ซม



รูปที่ 4-7 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 50 ซม

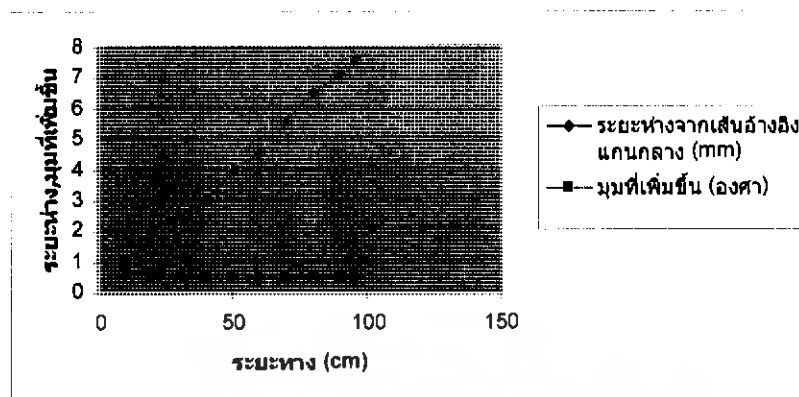
จากรูปที่ 4-7 แสดงค่าความผิดพลาดจากทิศทางและการเคลื่อนที่ของหุ่นยนต์ ในการเคลื่อนที่ทิศทางตรง ที่ระยะ 50 ซม ซึ่งสัมพันธ์กับตารางที่ 4-2

ระยะที่เคลื่อนที่ได้ (cm)	ระยะห่างจากเส้นอ้างอิงแกนกลาง (mm)	มุมที่เพิ่มขึ้น (องศา)
10	1	0.6
20	1.5	0.43
30	3.3	0.63
40	4.7	0.67
48	6	0.72

ตารางที่ 4-2 ผลการเคลื่อนที่ระยะทาง 50 ซม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเคลื่อนที่เส้นทางตรงที่ 100 ซม



รูปที่ 4-8 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 100 ซม

จากรูปที่ 4-8 แสดงค่าความผิดพลาดจากทิศทางและการเคลื่อนที่ของหุ่นยนต์ ในการเคลื่อนที่ทิศทางตรง ที่ระยะ 100 ซม ซึ่งสัมพันธ์กับตารางที่ 4-3

ระยะที่เคลื่อนที่ได้ (cm)	ระยะห่างจากเส้นอ้างอิงแกนกลาง (mm)	มุมที่เพิ่มขึ้น (องศา)
10	1	0.6
20	2	0.57
30	2.5	0.48
40	3	0.43
50	4	0.46
60	4.5	0.43
70	5.5	0.45
80	6.5	0.47
90	7	0.45
96	7.5	0.45

ตารางที่ 4-3 ผลการเคลื่อนที่ระยะทาง 100 ซม

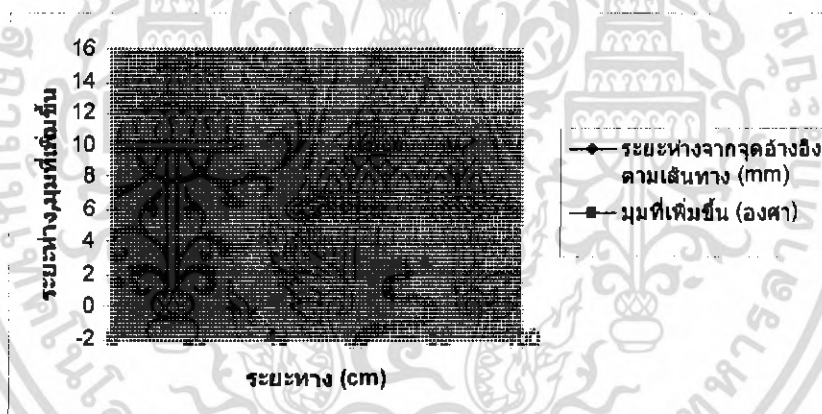
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลี้ยวของหุ่นยนต์ที่รัศมีการเลี้ยวที่ 39 ซม(คิดเป็นระยะทาง 63 cm)

ระยะที่เคลื่อนที่ได้ (cm)	ระยะห่างจากจุดอ้างอิงตามเส้นทาง (mm)	มุมที่เพิ่มขึ้น (องศา)
10	0	0
20	0	0
30	0	0
40	0	0
50	0	0
60	0	0
63	0	0

ตารางที่ 4-4 ผลการเคลื่อนที่ที่รัศมีการเลี้ยว 39 ซม(คิดเป็นระยะทาง 63 cm)

การเลี้ยวของหุ่นยนต์ที่รัศมีการเลี้ยวที่ 50 ซม(คิดเป็นระยะทาง 80 cm)



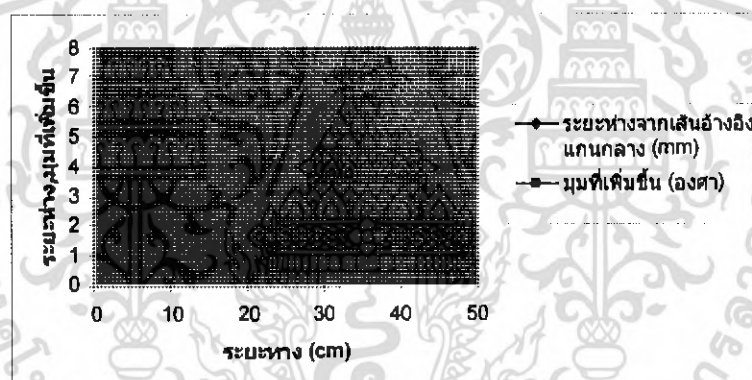
รูปที่ 4-9 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 80 ซม

จากรูปที่ 4-9 แสดงค่าความผิดพลาดจากทิศทางและการเคลื่อนที่ของหุ่นยนต์ ในการเคลื่อนที่แบบเลี้ยว ที่รัศมีการเลี้ยวที่ 50 ซม ซึ่งสัมพันธ์กับตารางที่ 4-5

ระยะที่เคลื่อนที่ได้ (cm)	ระยะห่างจากจุดอ้างอิงตามเส้นทาง (mm)	มุมที่เพิ่มขึ้น (องศา)
10	0	0
20	0	0
30	0	0
40	3	0.3
50	6	3
60	9	3
70	10	2.49
77	14	2.67

ตารางที่ 4-5 ผลการเคลื่อนที่ที่รัศมีการเลี้ยว 50 ซม (คิดเป็นระยะทาง 80 ซม)

4.2.2.2 การทำงานในโหมดจำเส้นทาง การเคลื่อนที่เส้นทางตรงที่ 50 ซม



รูปที่ 4-10 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 50 ซม

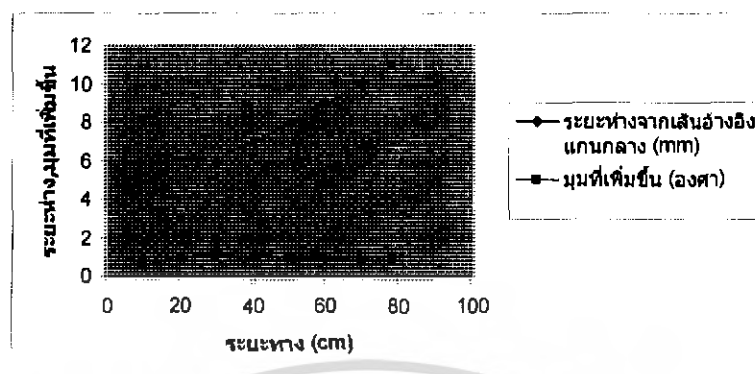
จากรูปที่ 4-10 แสดงค่าความผิดพลาดจากทิศทางและการเคลื่อนที่ของหุ่นยนต์ ในการเคลื่อนที่ทิศทางตรง ที่ระยะ 50 ซม ซึ่งสัมพันธ์กับตารางที่ 4-6

ระยะที่เคลื่อนที่ได้ (cm)	ระยะห่างจากเส้นอ้างอิงแกนกลาง (mm)	มุมที่เพิ่มขึ้น (องศา)
10	1	0.57
20	3	0.89
30	6	1.15
38.5	7.5	1.12

ตารางที่ 4-6 ผลการเคลื่อนที่ระยะทาง 50 ซม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเคลื่อนที่เส้นทางตรงที่ 100 ซม



รูปที่ 4-11 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 100 ซม

จากรูปที่ 4-11 แสดงค่าความผิดพลาดจากทิศทางและการเคลื่อนที่ของหุ่นยนต์ ในการเคลื่อนที่ ทิศทางตรง ที่ระยะ 100 ซม ซึ่งสัมพันธ์กับตารางที่ 4-7

ระยะที่เคลื่อนที่ได้ (cm)	ระยะห่างจากเส้นอ้างอิงแกนกลาง (mm)	มุมที่เพิ่มขึ้น (องศา)
10	1	0.57
20	3	0.86
30	6	1.15
40	7	1
50	8	0.92
60	9	0.86
70	10	0.8
78	11	0.8

ตารางที่ 4-7 ผลการเคลื่อนที่ระยะทาง 100 ซม

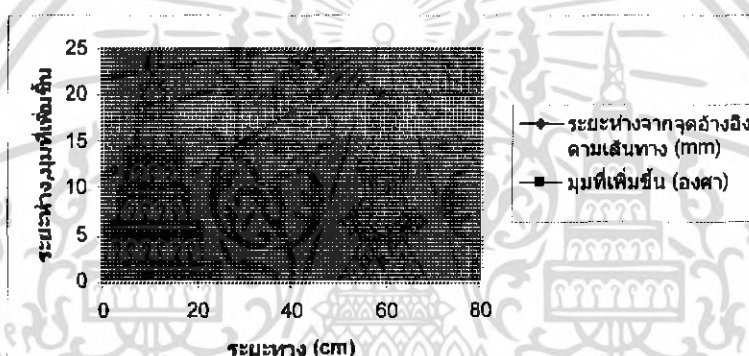
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลี้ยวของหุ่นยนต์ที่รัศมีการเลี้ยวที่ 39 ซม(คิดเป็นระยะทาง 63 cm)

ระยะที่เคลื่อนที่ได้ (cm)	ระยะห่างจากจุดอ้างอิงตามเส้นทาง (mm)	มุมที่เพิ่มขึ้น (องศา)
10	0	0
20	0	0
30	0	0
40	0	0
42	0	0

ตารางที่ 4-8 ผลการเคลื่อนที่ที่รัศมีการเลี้ยว 39 ซม(คิดเป็นระยะทาง 63 cm)

การเลี้ยวของหุ่นยนต์ที่รัศมีการเลี้ยวที่ 50 ซม(คิดเป็นระยะทาง 80 cm)



รูปที่ 4-12 แสดงระยะทางและมุมที่เปลี่ยนแปลงที่ 80 ซม

จากรูปที่ 4-12 แสดงค่าความผิดพลาดจากทิศทางและการเคลื่อนที่ของหุ่นยนต์ ในการเคลื่อนที่แบบเลี้ยว ที่รัศมีการเลี้ยวที่ 50 ซม ซึ่งสัมพันธ์กับตารางที่ 4-9

ระยะที่เคลื่อนที่ได้ (cm)	ระยะห่างจากจุดอ้างอิงตามเส้นทาง (mm)	มุมที่เพิ่มขึ้น (องศา)
10	1	0.57
20	11	3.15
30	15	2.86
40	18	2.58
50	22	2.52
59	20	1.94

ตารางที่ 4-9 ผลการเคลื่อนที่ที่รัศมีการเลี้ยว 50 ซม(คิดเป็นระยะทาง 80 cm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบการเคลื่อนที่ของหุ่นยนต์ในโหมดโปรแกรมเส้นทางและโหมดจำลองเส้นทางในรูปแบบต่างๆ

4.3.1 ลำดับขั้นการทดลอง

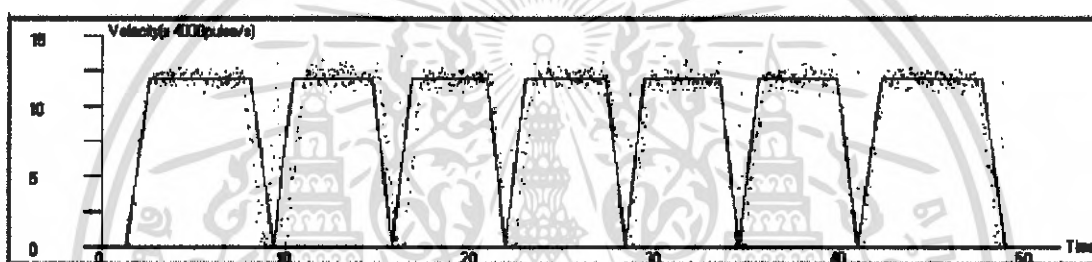
1. ทำการเชื่อมต่อพอร์ทอนุกรมกับชุดบอร์ด
2. จ่ายแรงดันให้แก่บอร์ดและชุดขับเคลื่อน
3. ทำการดาวน์โหลดโปรแกรมลงบนไมโครคอนโทรลเลอร์
4. รันโปรแกรม
5. ทำการทดสอบการเคลื่อนที่ของหุ่นยนต์ในโหมดโปรแกรมเส้นทางและโหมดจำลองเส้นทาง

4.3.2 ผลการทดลอง

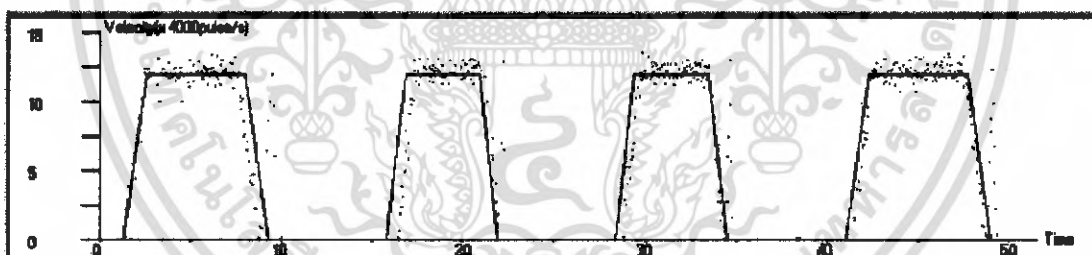
เมื่อทำการทดสอบการเคลื่อนที่ทั้งโหมดโปรแกรมเส้นทางและโหมดจำลองเส้นทาง ได้ผลดังนี้

4.3.2.1 การทำงานในโหมดโปรแกรมเส้นทาง

การเคลื่อนที่แบบสี่เหลี่ยมจตุรัส

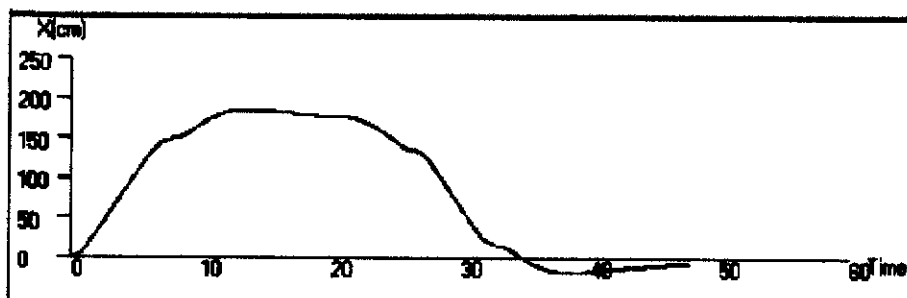


รูปที่ 4-13 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อขวา

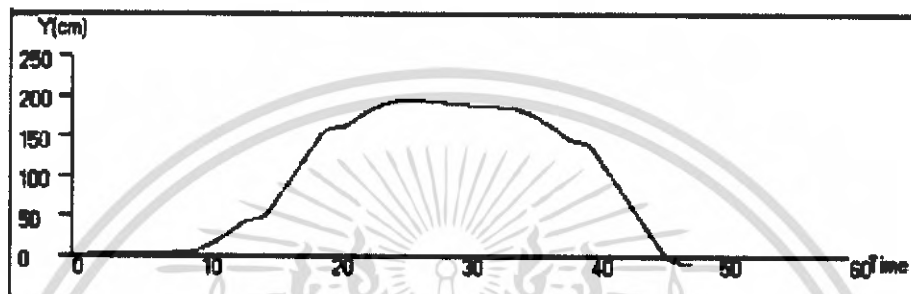


รูปที่ 4-14 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อซ้าย

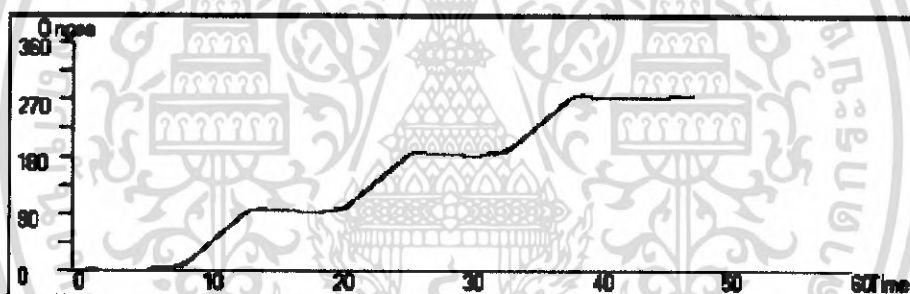
จากรูปที่ 4-13 และ 4-14 แสดงความเร็วจากการเคลื่อนที่ของล้อขวาและล้อซ้ายของหุ่นยนต์ในการเคลื่อนที่จริง (จุดสีน้ำเงิน) กับความเร็วที่ได้จากการกำหนดให้หุ่นยนต์เคลื่อนที่ (เส้นสีแดง)



รูปที่ 4-15 แสดงการเคลื่อนที่ในแนวแกน X



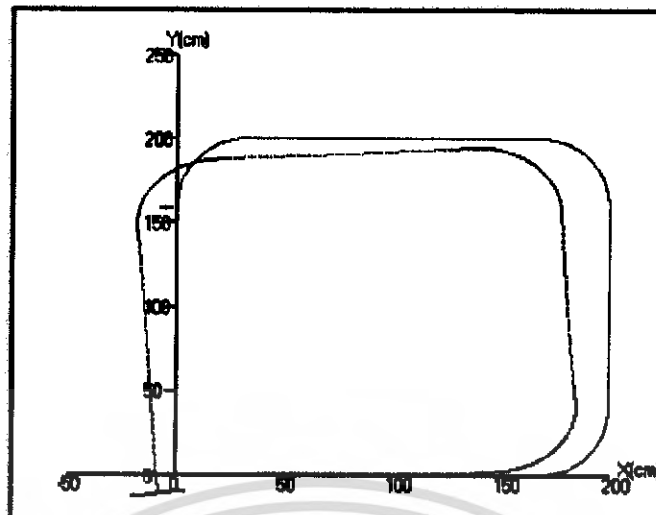
รูปที่ 4-16 แสดงการเคลื่อนที่ในแนวแกน Y



รูปที่ 4-17 แสดงการเปลี่ยนแปลงทิศทางของหุ่นยนต์ในรูปแบบของคา

จากรูปที่ 4-15 และ 4-16 แสดงการเคลื่อนที่ของหุ่นยนต์ตามแนวแกน X และแนวแกน Y และรูปที่ 4-17 แสดงทิศทางที่หุ่นยนต์เคลื่อนที่ จากความสัมพันธ์ดังกล่าวสามารถแสดงรวมในรูปที่ 4-18

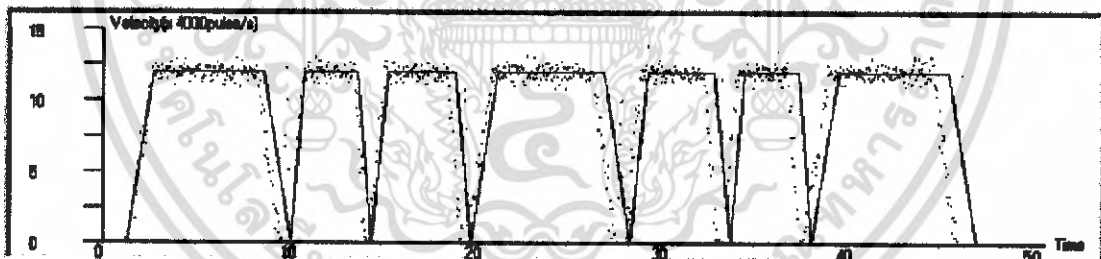
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



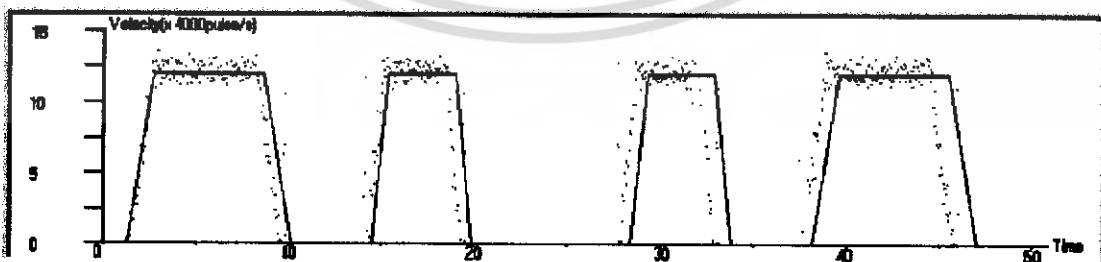
รูปที่ 4-18 แสดงภาพรวมการเคลื่อนที่ของหุ่นยนต์

จากรูปที่ 4-18 เส้นทางการเคลื่อนที่ของหุ่นยนต์จากจุดสุดท้ายกับจุดเริ่มต้นนั้น มีค่าความคลาดเคลื่อนจากการเคลื่อนที่ในแกน X 11 ซม. ในแนวแกน Y 12 ซม. และ มุมมีค่าความคลาดเคลื่อนไป 5.8 องศา โดยที่เส้นสีแดง แสดงเส้นทางที่กำหนดให้หุ่นยนต์เคลื่อนที่ และ เส้นสีน้ำเงิน แสดงเส้นทางที่หุ่นยนต์เคลื่อนที่ได้จริง

การเคลื่อนที่แบบสี่เหลี่ยมขนมเปียกปูน

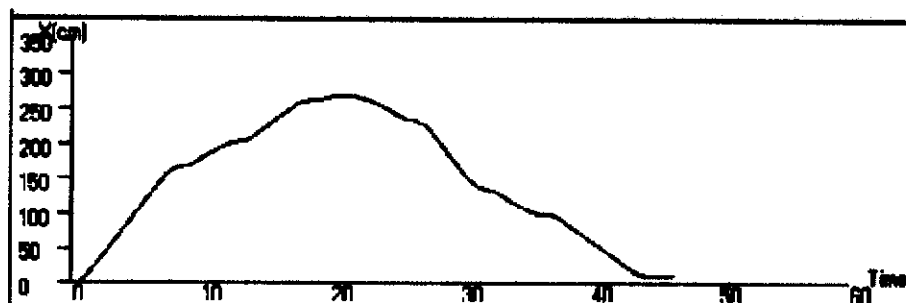


รูปที่ 4-19 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อขวา

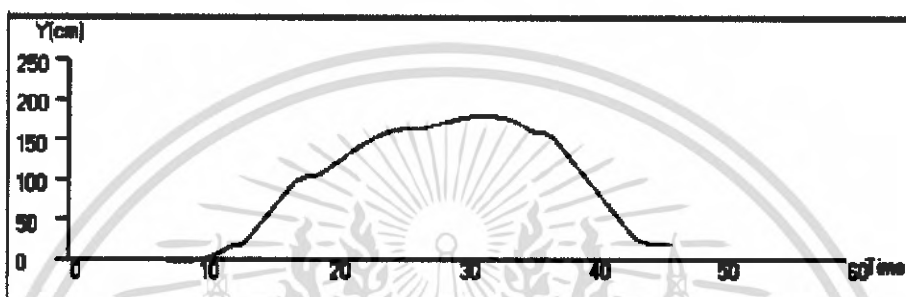


รูปที่ 4-20 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อซ้าย

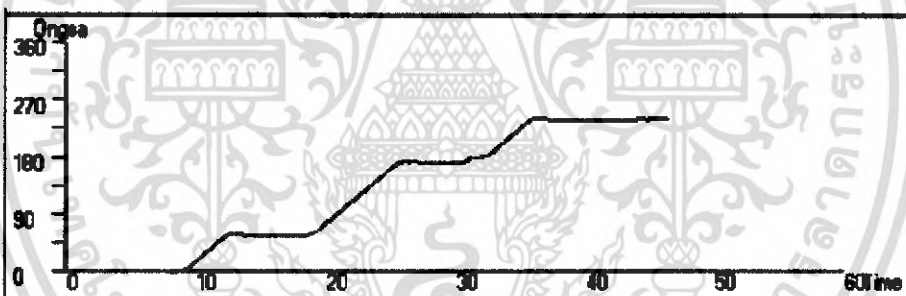
จากรูปที่ 4-19 และ 4-20 แสดงความเร็วจากการเคลื่อนที่ของล้อขวาและล้อซ้ายของหุ่นยนต์ในการเคลื่อนที่จริง (จุดสีน้ำเงิน) กับความเร็วที่ได้จากการกำหนดให้หุ่นยนต์เคลื่อนที่ (เส้นสีแดง) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นใบเซอร์เชียนด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-21 แสดงการเคลื่อนที่ในแนวแกน X



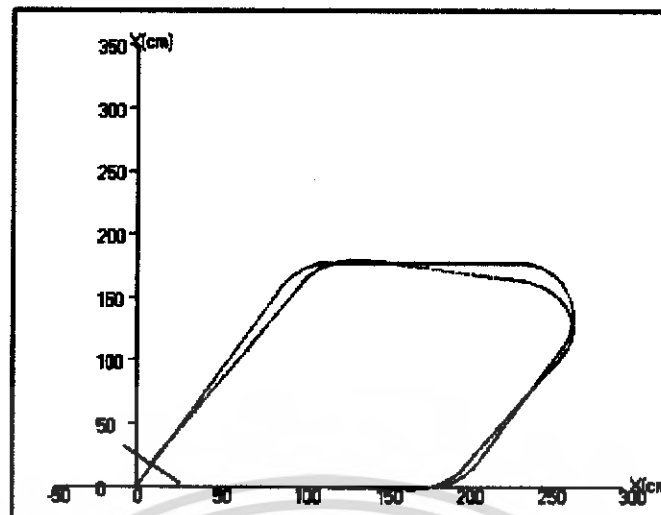
รูปที่ 4-22 แสดงการเคลื่อนที่ในแนวแกน Y



รูปที่ 4-23 แสดงการเปลี่ยนแปลงทิศทางของหุ่นยนต์ในรูปแบบองศา

จากรูปที่ 4-21 และ 4-22 แสดงการเคลื่อนที่ของหุ่นยนต์ตามแนวแกน X และแนวแกน Y และรูปที่ 4-23 แสดงทิศทางที่หุ่นยนต์เคลื่อนที่ จากความสัมพันธ์ดังกล่าวสามารถแสดงรวมในรูปที่ 4-24

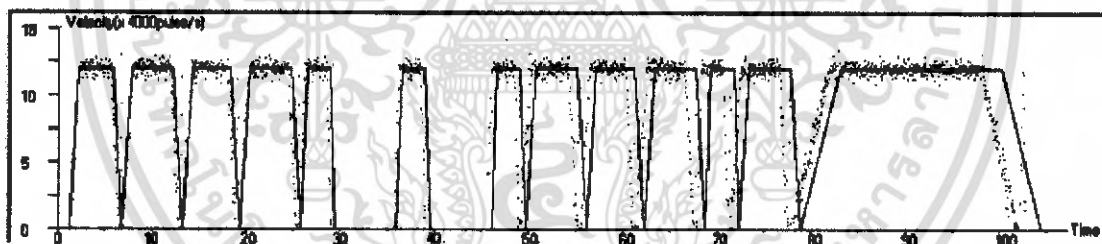
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



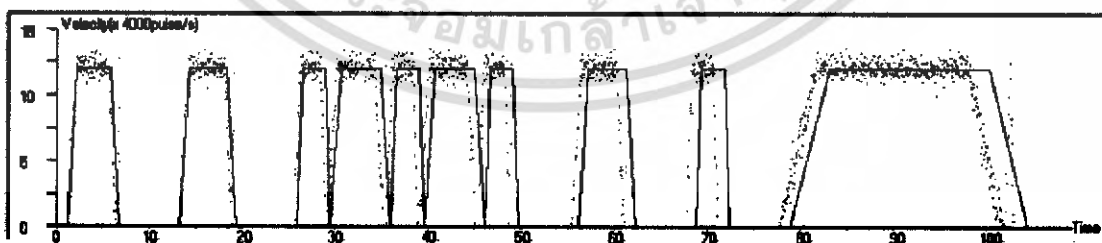
รูปที่ 4-24 แสดงภาพรวมการเคลื่อนที่ของหุ่นยนต์

จากรูปที่ 4-24 เส้นทางการเคลื่อนที่ของหุ่นยนต์จากจุดสุดท้ายกับจุดเริ่มต้นนั้น มีค่าความคลาดเคลื่อนจากการเคลื่อนที่ในแกน X 14 ซม. ในแนวแกน Y 19 ซม. และ มุมมีค่าความคลาดเคลื่อนไป 4 องศา โดยที่เส้นสีแดง แสดงเส้นทางที่กำหนดให้หุ่นยนต์เคลื่อนที่ และ เส้นสีน้ำเงิน แสดงเส้นทางที่หุ่นยนต์เคลื่อนที่ได้จริง

การเคลื่อนที่แบบหลายเส้นทาง



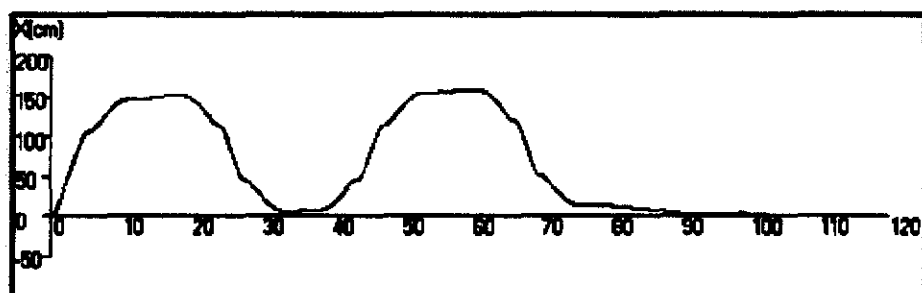
รูปที่ 4-25 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อขวา



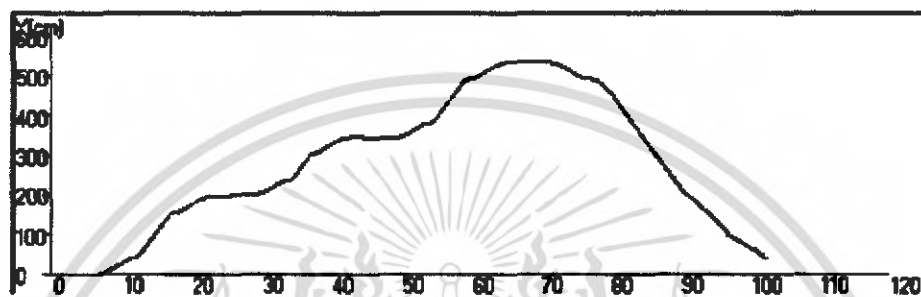
รูปที่ 4-26 แสดงผลการเกาะความเร็วในการเคลื่อนที่ของล้อซ้าย

จากรูปที่ 4-25 และ 4-26 แสดงความเร็วจากการเคลื่อนที่ของล้อขวาและล้อซ้ายของหุ่นยนต์ ในการเคลื่อนที่จริง (จุดสีน้ำเงิน) กับความเร็วที่ได้จากการกำหนดให้หุ่นยนต์เคลื่อนที่ (เส้นสีแดง)

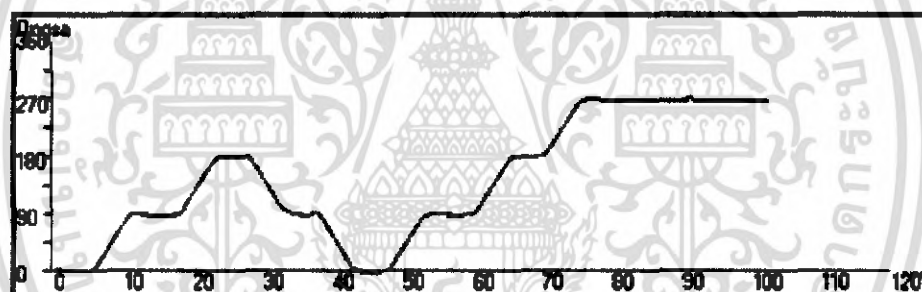
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-27 แสดงการเคลื่อนที่ในแนวแกน X

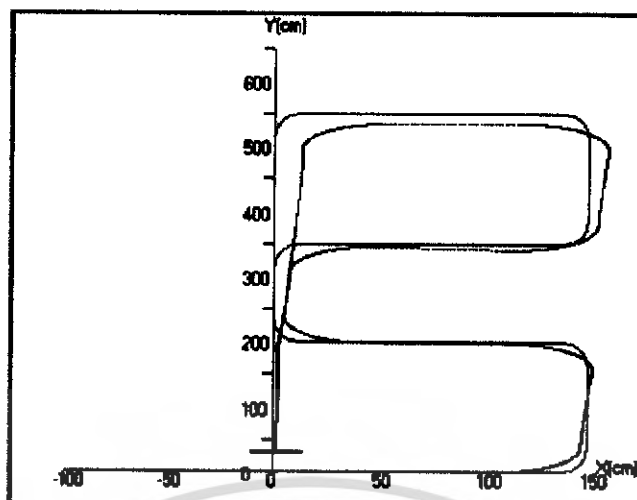


รูปที่ 4-28 แสดงการเคลื่อนที่ในแนวแกน Y



รูปที่ 4-29 แสดงการเปลี่ยนแปลงทิศทางของหุ่นยนต์ในรูปแบบของคา

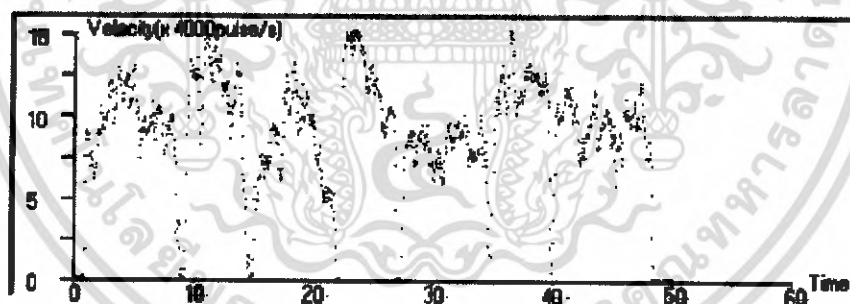
จากรูปที่ 4-27 และ 4-28 แสดงการเคลื่อนที่ของหุ่นยนต์ตามแนวแกน X และแนวแกน Y และรูปที่ 4-29 แสดงทิศทางที่หุ่นยนต์เคลื่อนที่ จากความสัมพันธ์ดังกล่าวสามารถแสดงรวมในรูปที่ 4-30



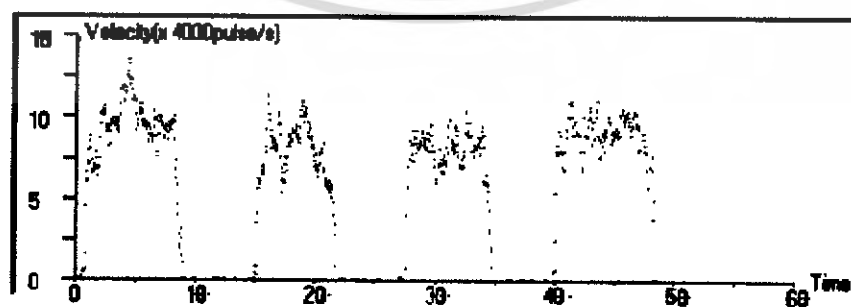
รูปที่ 4-30 แสดงภาพรวมการเคลื่อนที่ของหุ่นยนต์

จากรูปที่ 4-30 เส้นทางการเคลื่อนที่ของหุ่นยนต์จากจุดสุดท้ายกับจุดเริ่มต้นนั้น มีค่าความคลาดเคลื่อนจากการเคลื่อนที่ในแกน X 2 ซม. ในแนวแกน Y 31 ซม. และ มุมมีค่าความคลาดเคลื่อนไป 1 องศา โดยที่เส้นสีแดง แสดงเส้นทางที่กำหนดให้หุ่นยนต์เคลื่อนที่ และ เส้นสีน้ำเงิน แสดงเส้นทางที่หุ่นยนต์เคลื่อนที่ได้จริง

4.3.2.2 การทำงานในโหมดจำเส้นทาง



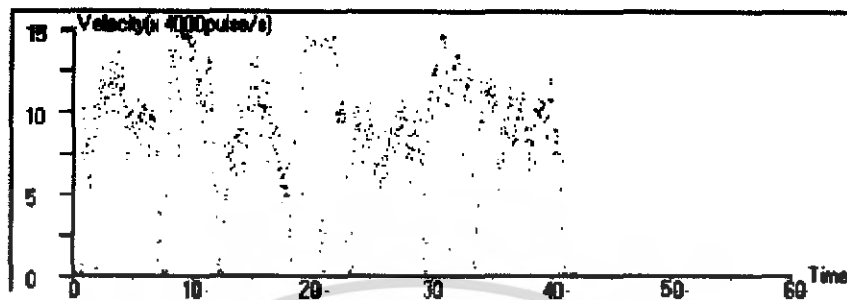
รูปที่ 4-31 แสดงค่าความเร็วในรูปผลตอบสนองที่ได้จากการจำเส้นทางของล้อขวา



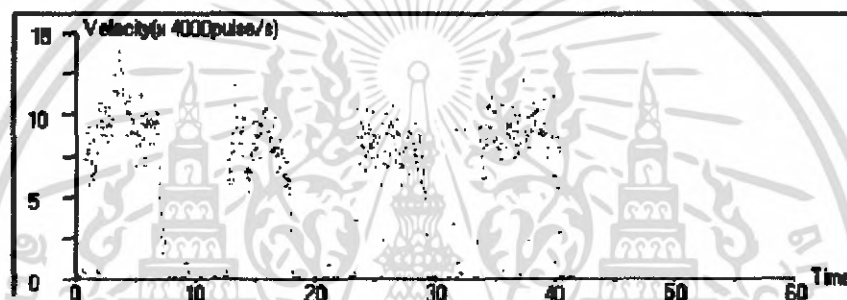
รูปที่ 4-32 แสดงค่าความเร็วในรูปผลตอบสนองที่ได้จากการจำเส้นทางของล้อซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4-31 และ 4-32 แสดงความเร็วของล้อขวาและล้อซ้ายของหุ่นยนต์ ที่ได้รับการบันทึกในการทำงานในโหมดจำเส้นทาง

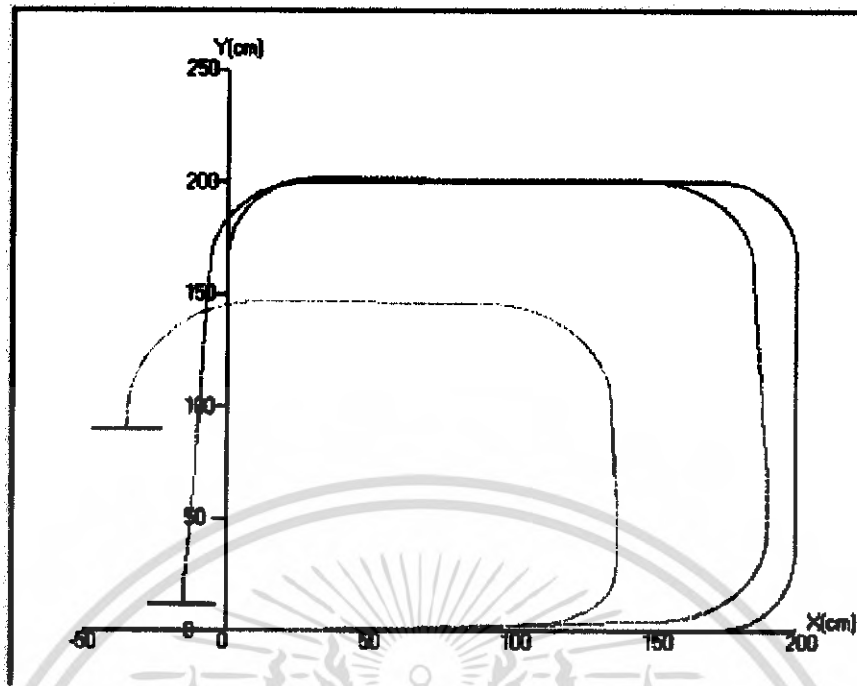


รูปที่ 4-33 แสดงค่าความเร็วในรูปผลตอบสนองที่ได้จากการเคลื่อนที่ของล้อขวา



รูปที่ 4-34 แสดงค่าความเร็วในรูปผลตอบสนองที่ได้จากการเคลื่อนที่ของล้อซ้าย

จากรูปที่ 4-33 และ 4-34 แสดงค่าความเร็วของล้อขวาและล้อซ้าย จากการเคลื่อนที่ของหุ่นยนต์ ซึ่งผลตอบสนองของรูปทั้งสอง จะสังเกตเห็นว่ามีแนวโน้มความเร็วจากการเคลื่อนที่เป็นลักษณะเดียวกันกับรูปที่ 4-31 และ 4-32 แต่ความยาวของผลตอบสนองนั้นจะสั้นกว่าผลตอบสนองจากการจำเส้นทาง ทั้งนี้สาเหตุจากความผิดพลาดดังกล่าว สามารถสรุปได้ในบทต่อไป



รูปที่ 4-35 แสดงภาพรวมการเคลื่อนที่ของหุ่นยนต์

จากรูปที่ 4-35 แสดงการเคลื่อนที่ของหุ่นยนต์ในโหมดจำเส้นทาง โดยที่เส้นสีแดง แสดงเส้นทางที่กำหนดให้หุ่นยนต์เคลื่อนที่ และ เส้นสีน้ำเงิน แสดงเส้นทางที่ได้จากการเงินหุ่นยนต์ และ เส้นสีเขียว แสดงเส้นทางที่หุ่นยนต์เคลื่อนที่ได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วิจารณ์และสรุปผล

5.1 บทสรุป

จากผลการทดลองข้างต้น เมื่อทำการเปลี่ยนชุดขับเคลื่อนมอเตอร์ และ การจูนค่าพีไอดี ผลตอบสนองของระบบพีไอดีที่ได้ คือ $P = 1.7$, $I = 0.5$ และ $D = 1E-11$ เมื่อนำการ ควบคุมแบบพีไอดีมาใช้ในการควบคุมการเคลื่อนที่ของหุ่นยนต์ ผลที่ได้จากการทดลองนั้นจะเห็นได้ว่า การทำงานในโหมดโปรแกรมเส้นทางและจำเส้นทาง หุ่นยนต์สามารถเคลื่อนที่ไปตามเส้นทางที่กำหนด ได้โดยมีค่าความผิดพลาดที่เกิดขึ้นจากการเคลื่อนที่เป็นไปตามผลการทดลองข้างต้น ซึ่งค่าความผิดพลาด ที่เกิดขึ้นสรุปได้ดังนี้

1. การรับส่งค่าข้อมูลทุกๆ 50 มิลลิวินาที ทำให้ไม่ได้ค่าความเร็วที่ต่อเนื่อง นอกเหนือจากช่วงระยะเวลาการรับส่งข้อมูล
2. ค่าตัวเลขที่ได้จากการคำนวณในการขับให้หุ่นยนต์เคลื่อนที่นั้น – ค่าตัวเลขที่ได้นั้นมีค่าเป็นทศนิยม ทำให้มีการปัดค่าตัวเลขลงในการส่งข้อมูล ทำให้ระยะทางที่หุ่นยนต์เคลื่อนที่น้อยลง
3. การเคลื่อนที่ของหุ่นยนต์ในโหมดจำเส้นทางที่ยังมีความผิดพลาดในการเคลื่อนที่นั้น มีสาเหตุดังนี้
 - การเซ็นหุ่นยนต์ที่เร็วเกินไป ถ้าความเร็วที่ได้จากการเซ็นมีค่าสูงกว่าความเร็วสูงสุดของหุ่นยนต์ หุ่นยนต์จะสามารถเคลื่อนที่ด้วยความเร็วสูงสุดของตัวมัน แต่ไม่อาจขับความเร็วที่เกินได้จึงทำให้เกิดความผิดพลาด
 - การเซ็นหุ่นยนต์ที่ความเร็วต่ำ ค่าความผิดพลาดที่เกิดขึ้นเกิดจากการทำงานของระบบพีไอดีที่ความเร็วต่ำนั้น ผลตอบสนองที่ได้เกิดการออสซิลเลททำให้การเคลื่อนที่ของหุ่นยนต์นั้นเกิดความผิดพลาด

5.2 ปัญหาที่เกิดขึ้นในการจัดทำโครงการ

ในการจัดทำโครงการชุดทดลองนี้ สามารถที่จะสรุปปัญหาที่เกิดขึ้นได้ดังนี้

1. ได้รับชุดทดลองล่าช้า อันเนื่องมาจาก เป็นชุดทดลองที่พียงนำเข้ามาในประเทศไทยเป็น ครั้งแรก
2. ใช้เวลาในการศึกษาการเขียนโปรแกรมและทดลองใช้งานทำให้สูญเสียเวลาในการดำเนินงาน
3. ไม่เข้าใจในการทำงานของฟังก์ชันต่างๆของตัวไมโครคอนโทรลเลอร์ ดีเอส พิกอย่างดีพอ
4. การทดลองของโครงการนี้ ตัวอุปกรณ์ในชุดทดลองเสียหายบ่อยทำให้เสียเวลาในการซ่อมแซม
5. ใช้เวลานานในการแก้ไขปรับปรุงโปรแกรมให้มีข้อผิดพลาดน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการแก้ไขและพัฒนา

1. ลดข้อจำกัดในการทำงานของหุ่นยนต์ โดยการเปลี่ยนรูปแบบการติดต่อรับส่งข้อมูลให้สะดวกขึ้น
2. ปรับปรุงรูปแบบการใช้งานของโปรแกรมให้สะดวกขึ้นในการใช้งาน
3. การทดลองระบบควบคุมพีไอดีนั้นสามารถนำไปใช้ในการเรียนการสอนได้จริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

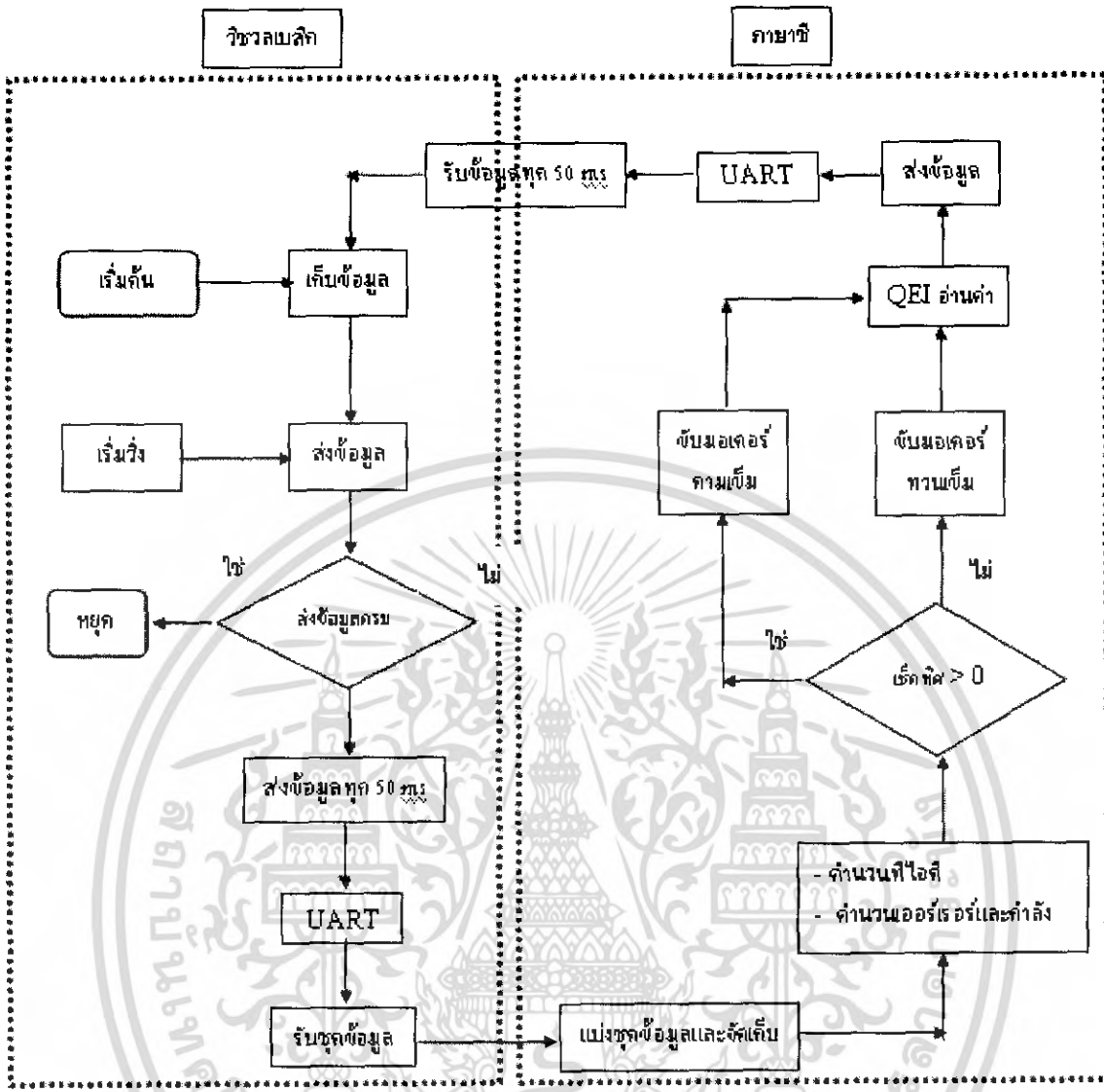
- [1] นาย กุลศักดิ์ ช่วยชู , นาย วัลลภ อินทรสังขนาวิน , โครงการหุ่นยนต์เคลื่อนที่ คณะวิทยาศาสตร์ ภาควิชาฟิสิกส์ประยุกต์ ปีการศึกษา 2542
- [2] นาย ดำรงค์ฤทธิ์ ภักฑารักษ์สกุล, นาย สิทธิชัย เฉขพร, นาย สุทัศน์ พริ้งสกุลชัย, โครงการหุ่นยนต์รับส่งเอกสารชนิดเคลื่อนที่ตามเส้น คณะวิศวกรรมศาสตร์ สาขาวิชาโทรคมนาคม ปีการศึกษา 2543
- [3] D.O'Halloran, A.Wolf, H.Choset : “ *Design of a High – Impact Survivable Robot* ”
- [4] Daniel Nikovski, Illah Nourbakhsh : “ *Learning Probabilistic Models for State Tracking of Mobile Robots* ”
- [5] นคร ภักดีชาติ, ณัฐพล วงศ์สุนทรชัย,ชัยวัฒน์ ลิ้มพรจิตรวิไล, คู่มือการทดลอง ดีเอสพี ไมโครคอนโทรลเลอร์ เบื้องต้นด้วยโปรแกรมภาษาซี
- [6] ดร. โยธิน เปรมปราณีรัชต์, ระบบเซอร์โวและอิเล็กทรอนิกส์คอนโทรลมอเตอร์
- [7] Norman S.Nise : “*Control System Engineering*”
- [8] อรรถพล บุญยะโกคา,วรพจน์ กรแก้ววัฒนกุล,ชัยวัฒน์ ลิ้มพรจิตรวิไล, เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม
- [9] สัจจะ จรัสรุ่งเรือง, คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6
- [10] G.W. Lucas : “*A Tutorial and Elementary Trajectory Model for The Differential Steering System. htm*”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



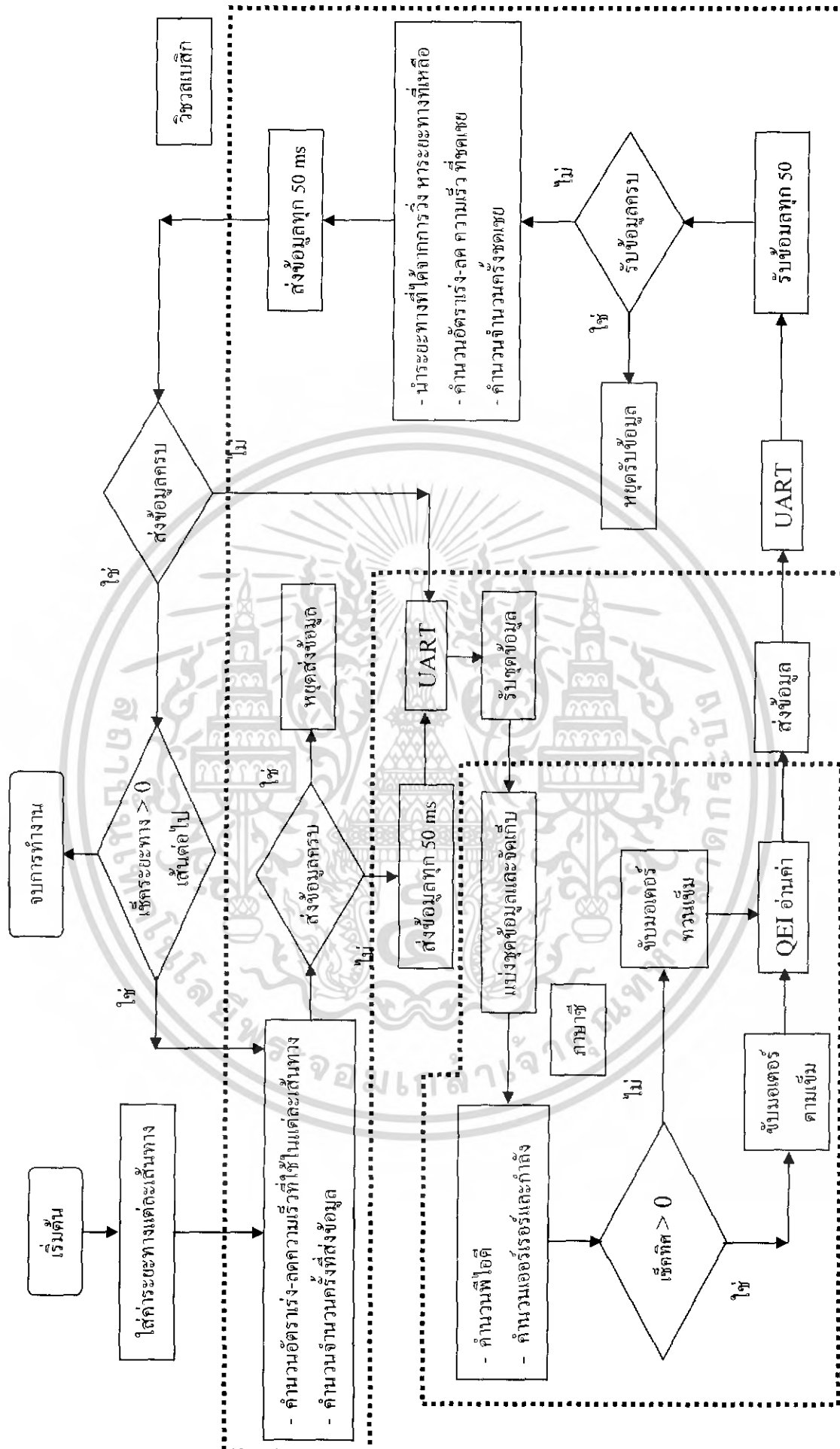
ภาคผนวก ก
ผังการทำงานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 แผนผังการทำงานโหมคจำเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 แผนผังการทำงานโปรแกรมเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

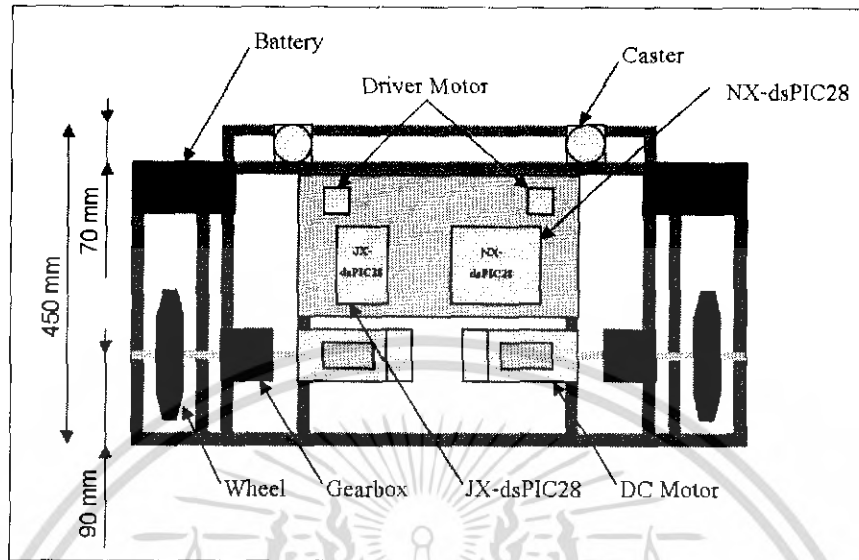


ภาคผนวก ข
การประกอบและใช้งานหุ่นยนต์ต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อโครงสร้างของหุ่นยนต์

การเชื่อมต่อโครงสร้างของหุ่นยนต์ แสดงดังรูปที่ ข.1



รูปที่ ข.1 แสดงองค์ประกอบและการเชื่อมต่อของอุปกรณ์

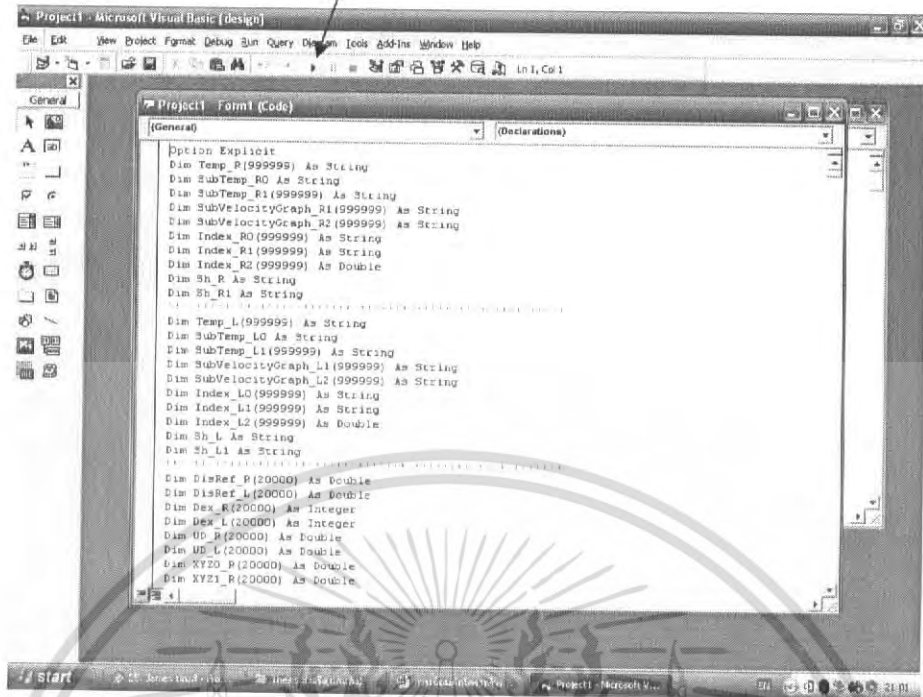
จากรูปโครงสร้างของหุ่นยนต์นั้นประกอบด้วย ชุดบอร์ดทดลอง (NX-dsPIC28) และ ชุดบอร์ดพัฒนา (JX-dsPIC28) ที่ได้รับการลงโปรแกรมให้กับไมโครคอนโทรลเลอร์เรียบร้อยแล้ว โดยที่บอร์ดทดลองและบอร์ดพัฒนาทำการเชื่อมต่อกับชุดขับเคลื่อนมอเตอร์แบบ เอชบริดจ์ (H- BRIDGE) ซึ่งได้รับพลังงานจากแบตเตอรี่ อีกส่วนหนึ่งของชุดบอร์ดทั้งสองนั้น ได้ทำการเชื่อมต่อกับเอนโคเดอร์ของมอเตอร์ โดยที่มอเตอร์ได้รับการประกอบชุดเฟืองทดกับล้อทั้งสองข้าง การทำงานของหุ่นยนต์นั้นจะติดต่อกับคอมพิวเตอร์ โดยผ่านพอร์ตอนุกรมซึ่งเชื่อมต่อกับบอร์ดทดลองและบอร์ดพัฒนา

การใช้งานหุ่นยนต์

1. ทำการเชื่อมต่อแบตเตอรี่กับชุดขับเคลื่อนมอเตอร์ และต่อแบตเตอรี่ให้กับบอร์ดทดลองและบอร์ดพัฒนา
2. ทำการเปิดโปรแกรม วิชาพลบลิค
3. ทำการโหลดโปรแกรมในโหมดโปรแกรมเส้นทางหรือใน โหมดจำเส้นทาง ดังแสดงดังรูปที่ ข.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

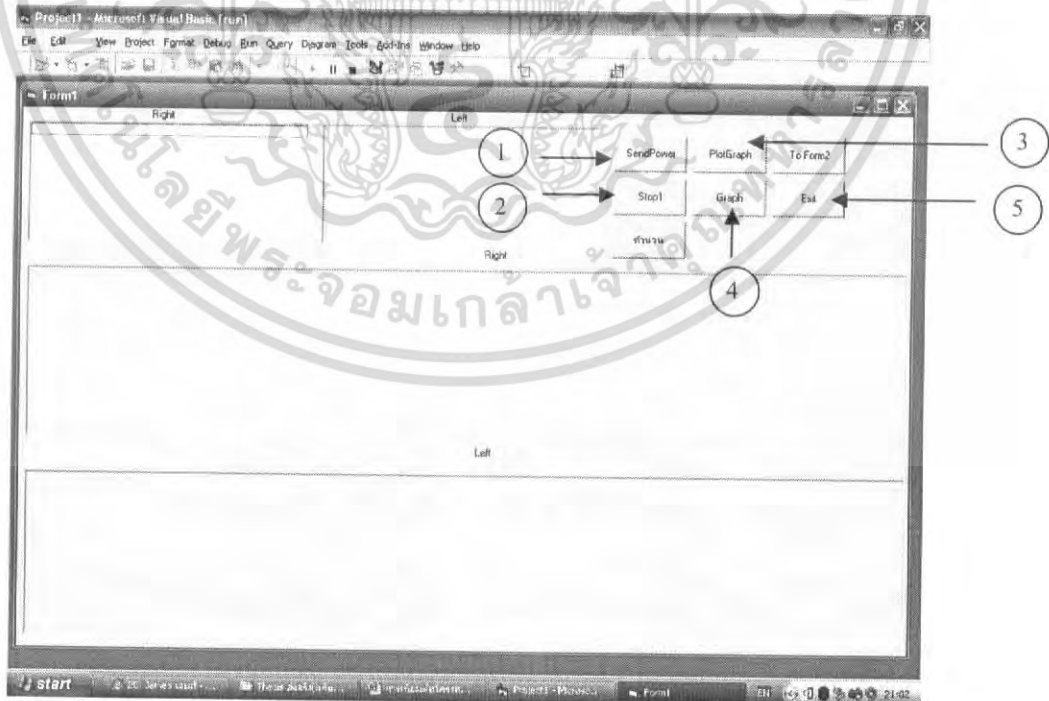
ปุ่มรันโปรแกรม



รูปที่ ข.2 แสดงหน้าจอโปรแกรมที่โหลดขึ้นมา

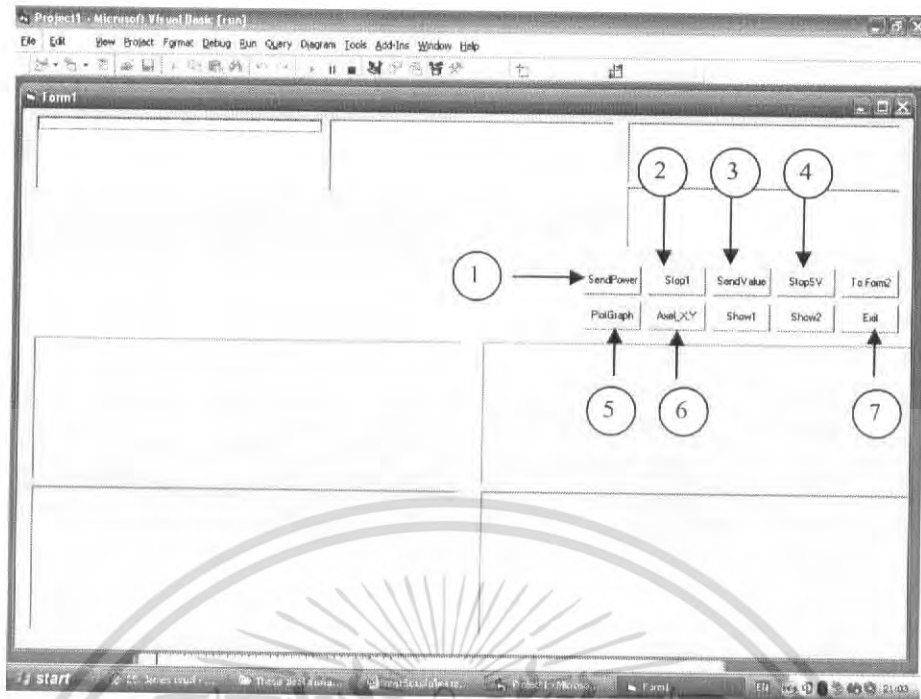
4. ทำการรันโปรแกรม โดยกดปุ่มดังแสดงในรูปที่ ข.2

เมื่อทำการรันโปรแกรมแล้วจะแสดงหน้าจอการทำงาน ดังแสดงในรูปที่ ข.3 และ รูปที่ ข.4



รูปที่ ข.3 แสดงหน้าจอโปรแกรมที่รันขึ้นมาในโหมดโปรแกรมเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.4 แสดงหน้าจอโปรแกรมที่รันขึ้นมาในโหมดจำเส้นเส้นทาง

ขั้นตอนการใช้งานโดยสังเขป

การทำงานในโหมดโปรแกรมเส้นทาง รูปที่ ข.3

- ทำการกดปุ่ม หมายเลข 1 ซึ่งเริ่มสั่งให้โปรแกรมทำงาน
- ทำการเปิดสวิตช์ให้หุ่นยนต์ทำงาน
- เมื่อหุ่นยนต์ทำงานเสร็จเรียบร้อย ให้ทำการกดปุ่ม หมายเลข 2 เพื่อเป็นการหยุดโปรแกรม
- ปุ่มหมายเลข 3 และ 4 แสดงการพล็อตกราฟ
- ปุ่มหมายเลข 5 ออกจากโปรแกรม

การทำงานในโหมดจำเส้นทาง รูปที่ ข.4

- ทำการกดปุ่ม หมายเลข 1 ซึ่งเริ่มสั่งให้โปรแกรมทำงาน
- ทำการเปิดสวิตช์ให้หุ่นยนต์ทำงาน
- เมื่อหุ่นยนต์เสร็จเรียบร้อย ให้ทำการกดปุ่ม หมายเลข 2 เพื่อเป็นการบันทึกค่า
- ทำการกดปุ่ม หมายเลข 3 ซึ่งเริ่มสั่งให้หุ่นยนต์เคลื่อนที่
- เมื่อหุ่นยนต์ทำงานเสร็จเรียบร้อย ให้ทำการกดปุ่ม หมายเลข 4 เพื่อเป็นการหยุดโปรแกรม
- ปุ่มหมายเลข 5 และ 6 แสดงการพล็อตกราฟ
- ปุ่มหมายเลข 7 ออกจากโปรแกรม

6. ถ้าต้องการเปลี่ยนเส้นทางในโหมดโปรแกรมเส้นทางให้ทำการแก้ไขที่ตัวโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้