

การออกแบบหุ่นยนต์ขนาดเล็กชนิดคลื่นนิ่ง

DESIGN OF STANDING WAVE TYPE MICROROBOT



๕๙๗
๖๗๒๒ ๗
๒๕๔๗

เลขหมู่.....
เลขทะเบียน..... 61960
วัน,เดือน,ปี 25 ก.ค. 2549

b..... 11559019
i.....

ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
ภาควิชาวิศวกรรมระบบควบคุม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2547

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ผู้จัดทำ

1. นาย สิทธิชาญ จิตรดี รหัส 45015341

2. นาย สุริยา ดัดพันธ์ รหัส 45015343



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบหุ่นยนต์ขนาดเล็กชนิดคลื่นนิ่ง (Design of Standing Wave Type Microrobot)

นาย สติธิชาญ จิตรดี

นาย สุริยา คัดพันธ์

อ. ดอน อิศรากร

ปีการศึกษา 2547

บทคัดย่อ

ในช่วงศตวรรษที่ผ่านมา มอเตอร์ไฟฟ้าได้เข้ามามีบทบาทสำคัญในโรงงานอุตสาหกรรมเป็นอย่างมาก และใช้กันอย่างแพร่หลาย ต่อมา การพัฒนารูปแบบได้ถูกจำกัดลง ทำให้เกิดความต้องการที่จะพัฒนามอเตอร์ให้มีขนาดเล็กตามที่ต้องการ อัลตราโซนิกมอเตอร์จึงเป็นทางเลือกหนึ่งที่ถูกนำมาใช้ทดแทนในส่วนที่มอเตอร์ไฟฟ้าไม่สามารถทำได้ และอัลตราโซนิกมอเตอร์ยังสามารถนำไปใช้งานได้หลากหลาย

แต่อัลตราโซนิกก็ยังมีข้อจำกัดอยู่ในรูปแบบการเคลื่อนที่แบบหมุน หรือ ลิเนียร์ เท่านั้น ดังนั้นจึงเป็นการยากที่จะนำมอเตอร์ชนิดนี้ไปออกแบบสร้างอุปกรณ์กำหนดตำแหน่งที่สามารถเคลื่อนที่ได้หลายทิศทาง โครงการนี้จึงนำเสนอการนำอัลตราโซนิกมาออกแบบเป็นหุ่นยนต์ที่สามารถเคลื่อนที่บนพื้นราบได้ทุกทิศทาง โดยใช้หลักการคลื่นนิ่ง

Abstract

In the past century, an electric motor plays the important roles in the industrial plants and has widely been used. However the development of the electric motor is limited creating rules for small size motors. An ultrasonic motor is therefore the alternative and has been used to substitute in the electric motor in the area that an electric motor could not well function. The ultrasonic motor can be used in various applications.

However , the ultrasonic motor is limited in rotary and linear movement . It is difficult to take the motor to design the specific positioning equipment which is more moveable direction. Hence, This project is presented bringing the ultrasonic motors to build the robot which can move all directions of the surface by using standing wave principle.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
สารบัญ	II
สารบัญรูปภาพ	IV
สารบัญตาราง	VII
บทที่ 1 บทนำ	
1.1 ความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์ในการดำเนินงาน	1
1.3 ขั้นตอนการศึกษาและจัดทำโครงการ	1
1.4 เนื้อหาที่จะกล่าวถึงในปริิญยานี้	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	
2.1 อัลตราโซนิคมอเตอร์คืออะไร	3
2.2 หลักการทำงานของอัลตราโซนิคมอเตอร์	3
2.3 การแยกชนิดของอัลตราโซนิคมอเตอร์	5
2.4 การขับอัลตราโซนิคมอเตอร์	14
2.5 การออกแบบวงจรขับอัลตราโซนิคมอเตอร์ชนิดคลื่นนิ่ง	19
2.6 การออกแบบหุ่นยนต์ขนาดเล็กชนิดคลื่นนิ่ง	23
2.7 การประยุกต์ใช้งานและแนวโน้มการพัฒนา	24
2.8 ทฤษฎีการประมวลผลภาพ	25
2.8.1 การกำหนดสี	25
2.8.2 องค์ประกอบของสี	26
2.8.3 การควบคุมกราฟฟิก	28
2.9 ความรู้เบื้องต้นของพอร์ตขนาน	31
2.9.1 ลักษณะทางกายภาพของพอร์ตขนาน	31
2.9.2 พอร์ตข้อมูล	33
2.9.3 พอร์ตควบคุม	35
2.9.4 พอร์ตแสดงสถานะ	36
2.9.5 การนำพอร์ตขนานไปใช้งาน	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ใด ๆ การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบหุ่นยนต์ชนิดเคลื่อนที่และระบบควบคุมหุ่นยนต์	
3.1 ส่วนประกอบของโครงการแบ่งออกได้เป็นส่วนต่าง ๆ ดังนี้	42
3.2 ส่วนประกอบทางด้านฮาร์ดแวร์	42
3.2.1 ส่วนของหุ่นยนต์	42
3.2.2 การขับอัลตราโซนิกมอเตอร์	43
3.2.3 โครงสร้างของวงจรขับอัลตราโซนิกมอเตอร์	43
3.3 ส่วนประกอบทางด้านซอฟต์แวร์	48
3.3.1 ส่วนประมวลผลภาพ	48
3.3.2 การควบคุมแบบอิสระ	54
บทที่ 4 การทดลองและผลการทดลอง	
4.1 การหาค่าแรงดันที่จ่ายให้แก่ piezoelectric ceramic	55
4.2 การวิเคราะห์หาความเร็วเชิงเส้นในแต่ละช่วงความถี่	57
4.3 การรับสัญญาณภาพจากกล้องวิดีโอ	65
4.4 การปรับแต่ง (calibration) แม่สี RGB บนตัวของหุ่นยนต์	65
4.5 การประมวลผลภาพของหุ่นยนต์	66
4.6 การควบคุมหุ่นยนต์	67
บทที่ 5 บทวิจารณ์และสรุป	
5.1 สรุปผลการทดลอง	68
5.2 ปัญหาที่พบในโครงการ	68
5.3 ผลที่ได้รับจากโครงการ	68
5.4 แนวทางในการพัฒนาและการประยุกต์	69
ภาคผนวก	
ขนาดของชิ้นงานของหุ่นยนต์	71
โปรแกรมประมวลผลภาพ	74
กิตติกรรมประกาศ	98
เอกสารอ้างอิง	99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

	หน้า
รูปที่ 2.1 แสดง Ultrasonic Motor	3
รูปที่ 2.2 แสดงหลักการทำงานของอัลตราโซนิคมอเตอร์	4
รูปที่ 2.3 แสดงคลื่นนิ่งที่ขนาดสูงสุด	6
รูปที่ 2.4 แสดงการแพร่กระจายของคลื่น	6
รูปที่ 2.5 แสดงความสัมพันธ์ระหว่างคลื่นนิ่งและคลื่นเคลื่อนที่	7
รูปที่ 2.6 แสดงโครงสร้างพื้นฐานของมอเตอร์ชนิดคลื่นนิ่ง	8
รูปที่ 2.7 แสดงโครงสร้างพื้นฐานของมอเตอร์ชนิดคลื่นนิ่ง โดยใช้ piezoelectric ceramic 2 ตัว เป็นตัวจับ	8
รูปที่ 2.8 แสดงปรากฏการณ์ Piezoelectricity	9
รูปที่ 2.9 Poling of piezoelectric ceramics	10
รูปที่ 2.10 แสดงเครื่องหมายของแกนสำหรับ poled เซรามิก	11
รูปที่ 2.11 แสดงวงจรเสมือนพื้นฐานของ อัลตราโซนิคมอเตอร์	15
รูปที่ 2.12 แสดงถึง Blocking Impedance	15
รูปที่ 2.13 แสดงวงจรเสมือนเมื่อไม่คิดค่า r_d	17
รูปที่ 2.14 แสดงจุดที่เกิดความถี่ Resonant frequency ω_0 และ antiresonant frequency ω_r	17
รูปที่ 2.15 แสดงให้เห็นถึงการต่อโหลดเข้ากับอัลตราโซนิคมอเตอร์	18
รูปที่ 2.16 แสดงระยะการเคลื่อนที่ ที่ความถี่ในแต่ละจุด	19
รูปที่ 2.17 แสดงวงจรเรียงกระแสชนิดเต็มคลื่น	19
รูปที่ 2.18 แสดงวงจรกำเนิดความถี่ โดย IC TL494	20
รูปที่ 2.19 แสดงวงจรจับโดยใช้ทรานซิสเตอร์	21
รูปที่ 2.20 แสดง วงจร RC	21
รูปที่ 2.21 แสดงหลักการการเคลื่อนที่ของหุ่นยนต์	23
รูปที่ 2.22 การประยุกต์ใช้งานในงานด้านต่างๆ	25
รูปที่ 2.23 กล้องสี RGB	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 2.24 แสดงการใช้ฟังก์ชัน BitBlt Copy ภาพจากต้นกำเนิด ไปยังจุดหมายปลายทาง	29
รูปที่ 2.25 แสดงการใช้ฟังก์ชัน StretchBlt คัดลอกภาพจากต้นกำเนิด ไปยังจุดหมายปลายทาง	30
รูปที่ 2.26 แสดงไคอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์	32
รูปที่ 2.27 วงจรภายในของพอร์ตข้อมูล	34
รูปที่ 2.28 วงจรภายในของพอร์ตควบคุม	36
รูปที่ 2.29 วงจรภายในของพอร์ตสถานะ	36
รูปที่ 3.1 รูปแบบโคจรรวมของระบบ	41
รูปที่ 3.2 แสดงบล็อกไคอะแกรมแสดงการทำงานของระบบ	41
รูปที่ 3.3 แสดงการติดตั้งมอเตอร์	42
รูปที่ 3.4 แสดงทิศทางการเคลื่อนที่ของหุ่นยนต์	42
รูปที่ 3.5 รูปวงจรเรียงกระแสชนิดเต็มคลื่น	44
รูปที่ 3.6 แสดงวงจรกำเนิดความถี่ใช้งานจริง	45
รูปที่ 3.7 รูปสัญญาณที่ได้จากวงจรกำเนิดสัญญาณถี่เหลี่ยม	45
รูปที่ 3.8 แสดงภาพวงจรที่ใช้จริง	46
รูปที่ 3.9 แสดงวงจรรวมที่ใช้จริง	47
รูปที่ 3.10 แผนผังแสดงการทำงานประมวลผลภาพ	48
รูปที่ 3.11 แสดงการคำนวณหาค่ามุมของหุ่นยนต์	49
รูปที่ 3.12 แสดงการคำนวณหาค่ามุมระยะทางของหุ่นยนต์กับจุดอ้างอิง	50
รูปที่ 3.13 แสดงเงื่อนไขการเคลื่อนที่แบบเดินหน้า	51
รูปที่ 3.14 แสดงเงื่อนไขการเคลื่อนที่เลี้ยวขวา	51
รูปที่ 3.15 แสดงเงื่อนไขการเคลื่อนที่เลี้ยวซ้าย	52
รูปที่ 3.16 แสดงแผนผังการควบคุมการเคลื่อนที่ไปยังพิกัดอ้างอิง	53
รูปที่ 3.16 แสดงแผนผังการควบคุมอิสระ	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

	หน้า
รูปที่ 4.1 แสดงความสัมพันธ์ของ แรงดัน ความถี่ที่ป้อนให้แก่วงจร กับแรงดันที่ตกคร่อม piezoelectric ceramic โดยได้จากหุ่นยนต์ ตัวที่ 1 ที่น้ำหนัก (87 กรัม)	55
รูปที่ 4.2 แสดงความสัมพันธ์ของ แรงดัน ความถี่ที่ป้อนให้แก่วงจร กับแรงดันที่ตกคร่อม piezoelectric ceramic โดยได้จากหุ่นยนต์ ตัวที่ 2 ที่น้ำหนัก (134.5 กรัม)	56
รูปที่ 4.3 แสดงทิศทางการเคลื่อนที่ของหัวชุดขับแต่ละชุด	59
รูปที่ 4.4 แสดงการคำนวณความเร็วของหุ่นยนต์	60
รูปที่ 4.5 แสดงกราฟความเร็วเชิงมุมและความเร็วเชิงเส้น เทียบกับความถี่ ของ piezoelectric ceramic แต่ละตัว	64
รูปที่ 4.6 แสดงการดึงภาพมาแสดงผล	65
รูปที่ 4.7 แสดงการดึงภาพมาดูเฉพาะส่วน	65
รูปที่ 4.8 ก. ยานของแม่สี RGB ที่ทำการปรับแต่งได้ ข. การซูมภาพมาปรับแต่งแม่สี RGB	66
รูปที่ 4.9 ก. รูปที่รับเข้ามาจากกล้องวิดีโอ ข. รูปจำลองตำแหน่งพิกัดที่ได้จากการประมวลผลจากคอมพิวเตอร์	67
รูปที่ 4.10 ก. – รูปที่ 4.10 ข. รูปผลการทดลองของการทิศทางควบคุมหุ่นยนต์	67

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงการนำ piezoelectric ceramic มาใช้งาน	14
ตารางที่ 2.2 แสดงสัญญาณสำคัญของพอร์ตขนาน ที่ใช้ติดต่อกับเครื่องพิมพ์	33
ตารางที่ 2.3 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน	35
ตารางที่ 2.4 แสดงแอสเคตของพอร์ตขนาน	37



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ปัจจุบันเทคโนโลยีได้มีความก้าวหน้าอย่างรวดเร็ว อุปกรณ์ต่างๆที่ถูกพัฒนาขึ้นมามีขนาดเล็กลงแม้กระทั่งมอเตอร์ ก็มีการพัฒนาให้มีขนาดเล็กลงเพื่อที่จะได้นำไปเป็นส่วนประกอบในอุปกรณ์ที่ถูกพัฒนาให้มีขนาดเล็ก อัลตราโซนิคมอเตอร์เป็นมอเตอร์ชนิดหนึ่งที่มีขนาดเล็ก มีประสิทธิภาพสูงสามารถนำมาประยุกต์ใช้ในงานต่างๆได้มากมาย

จึงทำให้เกิดโครงการนี้ที่มีแนวคิดที่จะนำอัลตราโซนิคมอเตอร์มาพัฒนาให้สามารถเคลื่อนที่ได้ อย่างอิสระบนแนวระนาบ

1.2 วัตถุประสงค์ในการดำเนินงาน

1. เพื่อศึกษาการทำงานของอัลตราโซนิคมอเตอร์ และนำไปประยุกต์ใช้ในโครงการ
2. เพื่อศึกษาและออกแบบหุ่นยนต์ขนาดเล็กชนิดเคลื่อนที่
3. เพื่อศึกษาการเขียน โปรแกรม VISUAL BASIC กับการเชื่อมต่อกับอุปกรณ์ภายนอก
4. เพื่อศึกษาและออกแบบโปรแกรมเพื่อสร้างรูปแบบการควบคุมการเคลื่อนที่ของหุ่นยนต์

1.3 ขั้นตอนการศึกษาและจัดทำโครงการ

การศึกษาและจัดทำโครงการนี้เริ่มจาก การศึกษาการทำงานของ piezoelectric ceramic เพื่อที่จะนำมาออกแบบอัลตราโซนิคมอเตอร์ ที่ใช้ในการสร้างหุ่นยนต์ จากนั้นทำการสร้างวงจรขับเคลื่อนหุ่นยนต์ และ เขียนโปรแกรมควบคุมการเคลื่อนที่ของหุ่นยนต์โดยผ่านทางกล้องวีดีโอ

โครงการนี้จะต้องอาศัยความรู้เกี่ยวกับลักษณะการทำงานและคุณสมบัติภายในของ piezoelectric ceramic รวมทั้งหลักการทางฟิสิกส์ นำมาสร้างรูปแบบของอัลตราโซนิคมอเตอร์ จากนั้นต้องอาศัยความรู้เกี่ยวกับวงจรไฟฟ้าและอิเล็กทรอนิกส์เพื่อออกแบบสำหรับสร้างวงจรขับเคลื่อนหุ่นยนต์ และสุดท้าย อาศัยความรู้จากการศึกษาโปรแกรม VISUAL BASIC ทำการเขียนโปรแกรมควบคุมลักษณะการเคลื่อนที่ของหุ่นยนต์โดยมีกล้องวีดีโอเป็นภาคอินพุตรับภาพเข้ามาประมวลผลในโปรแกรมที่เขียนไว้และส่งค่าที่ประมวลผลแล้วออกทางพอร์ตขนาน และส่งกลับเข้าไปในส่วนของการควบคุมขับเคลื่อนเพื่อเป็นการป้อนกลับให้หุ่นยนต์เคลื่อนที่ตามตำแหน่งที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 เนื้อหาที่จะกล่าวในปฏิญญานิพนธ์

เนื้อหาที่จะกล่าวในปฏิญญานิพนธ์ฉบับนี้คือ

ในบทที่ 2 จะกล่าวถึงหลักการและทฤษฎีเกี่ยวกับ โครงสร้างและหลักการทำงานของอัลตราโซนิคมอเตอร์ การออกแบบหุ่นยนต์ขนาดเล็กชนิดคลื่นนิ่ง หลักการออกแบบขับเคลื่อนอัลตราโซนิคมอเตอร์ หลักการประมวลผลภาพผ่านกล้องวีดีโอ

ในบทที่ 3 การออกแบบหุ่นยนต์ชนิดคลื่นนิ่งและระบบควบคุมหุ่นยนต์

ในบทที่ 4 การทดลองและผลการทดลอง โดยจะกล่าวถึง การทดลองหาค่าแรงดันและค่าความถี่ที่เหมาะสมของตัว piezoelectric ceramic ทั้ง 4 ตัว และการทดสอบการเคลื่อนที่ของหุ่นยนต์ผ่านทางกล้องวีดีโอ

ในบทที่ 5 บทวิจารณ์และสรุป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและความรู้ที่เกี่ยวข้อง

อัลตราโซนิคมอเตอร์ทำให้เกิดการเปลี่ยนแปลงอย่างมากภายในวงการอุตสาหกรรม เนื่องจากมอเตอร์ชนิดนี้มีข้อดีหลายอย่างเมื่อเทียบกับมอเตอร์ไฟฟ้ากระแสตรง ซึ่งเป็นผลที่ได้จากการใช้ piezoelectric ceramic โดยตรงมาทำมอเตอร์ชนิดนี้ โดยหลักการทำงานของ piezoelectric ceramic คือ การยืดและหดอย่างรวดเร็ว เมื่อป้อนแรงดันไฟฟ้าเป็นสัญญาณพัลส์ โดยความเร็วของการยืดและหดขึ้นอยู่กับความถี่ที่ป้อนเข้าไปในตัว piezoelectric ceramic จากการยืดและหดตัวด้วยความถี่สูงนี้ จึงทำให้เกิดการสั่นเชิงกลขึ้น การสั่นเชิงกลที่เกิดขึ้นนี้สามารถนำไปปลั๊กวัตถุให้สามารถเคลื่อนที่ได้ ด้วยหลักการเช่นเดียวกันนี้ จึงนำมาประยุกต์ทำหุ่นยนต์ขนาดเล็กชนิดคลื่นนิ่ง

2.1 อัลตราโซนิคมอเตอร์คืออะไร ?

อัลตราโซนิคมอเตอร์ คือ มอเตอร์ที่ใช้ประโยชน์การสั่นเชิงกลของ piezoelectric ceramic ที่มีความถี่ในย่านอัลตราโซนิค (มากกว่า 20 kHz) เป็นตัวขับเคลื่อนให้มอเตอร์ทำงานได้ ดังนั้นมอเตอร์ชนิดนี้จึงไม่มีส่วนของขดลวดเหนี่ยวนำหรือแม่เหล็กถาวร ทำให้อัลตราโซนิคมอเตอร์มีข้อดีหลายอย่างที่เหนือกว่ามอเตอร์ที่ใช้การเหนี่ยวนำจากสนามแม่เหล็กในการขับเคลื่อน โดยอัลตราโซนิคมอเตอร์ที่ขายอยู่ในท้องตลาดมีตัวอย่าง แสดงดังรูปที่ 2.1

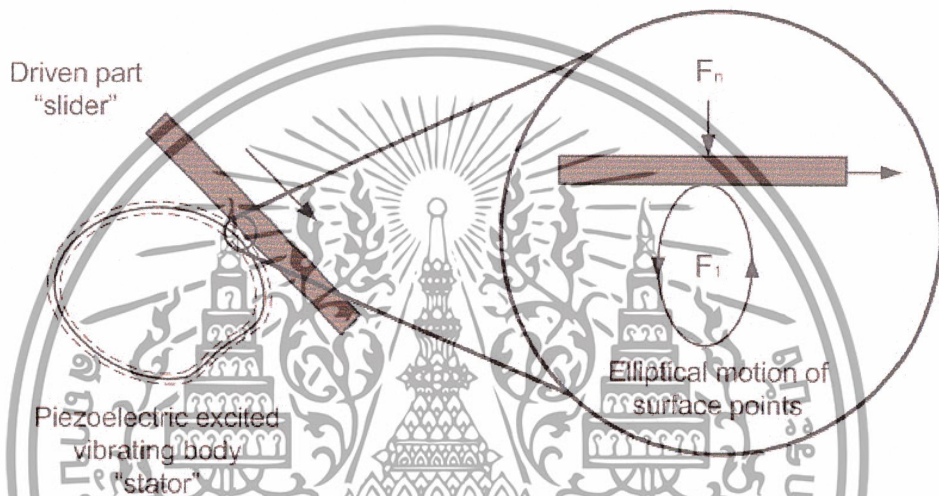


รูปที่ 2.1 แสดง Ultrasonic Motor (ที่มา www.psu.edu/ur/heartdevices/tinymotor.htm)

2.2 หลักการทำงานของอัลตราโซนิคมอเตอร์

หลักการทำงานพื้นฐานของ piezoelectric ceramic เป็นการเปลี่ยนแปลงพลังงานไฟฟ้าให้เป็นพลังงานทางกลในรูปแบบของการสั่น โดยส่วนใหญ่ piezoelectric ceramic จะถูกยึดติดอยู่กับโลหะราคาไม่แพงชนิดใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือยางที่ยืดหยุ่นได้ ซึ่งเป็นการกระตุ้นโดยแรงปฏิกิริยาของ piezoelectric ceramic โดยพื้นผิวบริเวณที่ถูกกระตุ้นนั้นจะทำการเคลื่อนที่เป็นแบบวงรี ด้วยความถี่ย่านอัลตราโซนิก ส่วนที่เคลื่อนที่ (slider) ถูกกดด้วยแรงตั้งฉาก (F_n : static normal force) และด้วยเหตุนี้ในแต่ละรอบของการสั่น จะทำให้ส่วนที่เคลื่อนที่แบบต่อเนื่อง ด้วยแรงผลัก (F_t : thrust force) โดยตัวอย่างการทำงานของอัลตราโซนิกมอเตอร์แสดงดังรูปที่ 2.2



รูปที่ 2.2 แสดงหลักการทำงานของอัลตราโซนิกมอเตอร์

บนพื้นฐานหลักการทำงานนี้ ทำให้เกิดการพัฒนาอัลตราโซนิกมอเตอร์หลากหลายรูปแบบซึ่งมีข้อดีและข้อด้อย จำแนกได้ดังนี้

ข้อดีที่พบในตัวของอัลตราโซนิกมอเตอร์

- ที่ความเร็วรอบต่ำ ให้แรงบิดสูง
- ผลตอบสนองของการทำงานเร็ว
- สามารถปรับความเร็วได้หลายระดับ
- มีแรงต้านทานระหว่างโรเตอร์และสเตเตอร์ในขณะหยุด ทำให้ไม่ต้องจ่ายพลังงาน
- มีความแม่นยำในการเคลื่อนที่สูงเมื่อใช้กับระบบควบคุมแบบปิด
- ให้กำลังมากเมื่อเทียบกับอัตราส่วนของน้ำหนัก
- ไม่มีเสียงรบกวนในขณะทำงาน

มีขนาดเล็กและน้ำหนักเบา เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โครงสร้างมอเตอร์ไม่ซับซ้อน
- ไม่มีกระแสสนามแม่เหล็กไฟฟ้า
- ไม่มีผลกระทบจากสนามแม่เหล็กและสนามไฟฟ้าจากภายนอก
- สามารถปรับความเร็วได้โดยไม่ต้องใช้เฟือง
- กินกำลังต่ำ

ข้อดีที่พบในตัวของอัลตราโซนิคมอเตอร์

- ต้องการแหล่งจ่ายพลังงานความถี่สูง
- มีลักษณะความไม่เป็นเชิงเส้นสูง
- มีอายุการใช้งานสั้น
- เนื่องจากมอเตอร์ต้องอาศัยแรงเสียดทานและการสัมผัสเนื่องในการเคลื่อนที่ ดังนั้นเมื่อทำงานไปสักระยะ จะเกิดความร้อนสะสมขึ้นภายในตัวมอเตอร์ ทำให้ความเร็วลดลง

2.3 การแยกชนิดของอัลตราโซนิคมอเตอร์

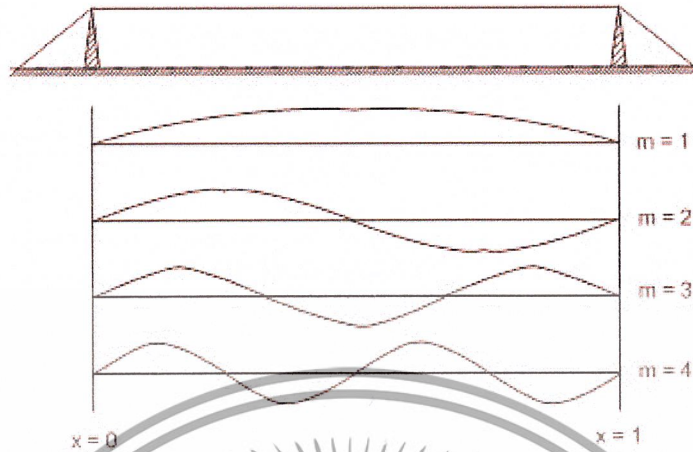
อัลตราโซนิคมอเตอร์สามารถแยกได้ตามความแตกต่างของ โครงสร้างทางกล เพื่อให้ได้รับการเคลื่อนที่ลักษณะวงรีบนพื้นผิวที่ต้องการ โดยหลักการแล้วลักษณะพิเศษสำหรับการแบ่งประเภทของอัลตราโซนิคมอเตอร์ แบ่งตามลักษณะการใช้ประโยชน์จากชนิดการสั่น ซึ่งการสั่นที่นำมาใช้นี้ถูกสร้างขึ้นในมอเตอร์ด้วยชนิดของคลื่นแบบใดแบบหนึ่งจากสองแบบ คือ ชนิดคลื่นนิ่ง (standing wave) หรือ ชนิดคลื่นเคลื่อนที่ (travelling wave)

คลื่นนิ่งและคลื่นเคลื่อนที่ (standing wave and travelling wave)

ถ้าหากพิจารณาถึงการเกิดคลื่นสองรูปแบบนี้ สามารถที่จะพิจารณาได้จากการยกตัวอย่างดังต่อไปนี้

ถ้าหากมีลวดเส้นหนึ่งตรึงไว้ให้แน่นที่ปลายทั้งสองด้าน เมื่อลวดถูกดีด ลวดเส้นนั้นจะเกิดการแกว่งขึ้นลงสลับกันไป จะสังเกตเห็นว่า ทุกจุดบนเส้นลวดจะมีความถี่ในการแกว่งเท่ากันแต่จะมีขนาดแตกต่างกันไป ณ เวลาใดๆ เมื่อนำขนาดบนจุดต่างๆที่ตำแหน่งเดียวกันมารวมกันจะทำให้รูปคลื่นนี้มีลักษณะเปรียบเสมือนเส้นตรง จึงเรียกคลื่นชนิดนี้ว่า คลื่นนิ่ง (standing wave) โดยการเกิดของคลื่นนิ่งในเส้นลวดแสดงดังรูป 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงคลื่นนิ่งที่ขนาดสูงสุด

ส่วนคลื่นอีกชนิดหนึ่ง คือ คลื่นเคลื่อนที่ ซึ่งพิจารณาได้จาก การกระเพื่อมของรูปคลื่นสัญญาณ จากจุดหนึ่งแพร่กระจายไปยังอีกจุดหนึ่ง โดยมีขนาดของคลื่นเท่ากันตลอดการแพร่กระจายไป โดยความเร็วที่คลื่นเคลื่อนที่ เคลื่อนไปข้างหน้าเรียกว่า ความเร็วเฟส (phase velocity) โดยสังเกตได้จากการเคลื่อนที่ของคลื่น แสดงดังรูปที่ 2.4



รูปที่ 2.4 แสดงการแพร่กระจายของคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

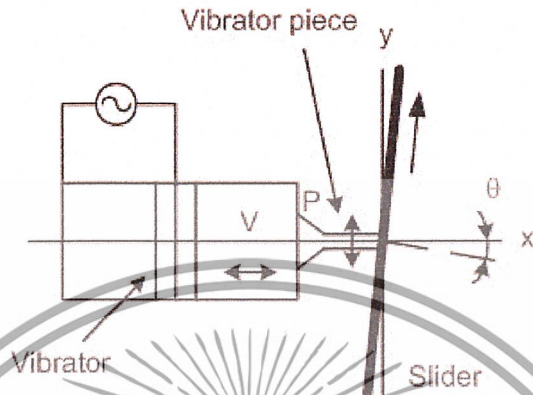
คลื่นนิ่งที่ปรากฏบนผิวน้ำเป็นการเสริมกันของคลื่นเคลื่อนที่ ที่แผ่ออกไปแล้วสะท้อนกลับมา คลื่นนิ่งและคลื่นเคลื่อนที่มีความสัมพันธ์ใกล้เคียงกันมาก พิจารณาที่อ่างน้ำขนาดใหญ่มีน้ำเต็มอ่าง เมื่อโยนหินลงในอ่างและทำการยกก้อนหินขึ้นจากอ่างน้ำอย่างรวดเร็ว ทันใดนั้น คลื่นนิ่งถูกสร้างขึ้น ดังแสดงให้เห็นดังรูปที่ 2.5 คลื่นเคลื่อนที่จะเคลื่อนที่ไปกระทบที่ขอบอ่าง หลังจากนั้น จะสะท้อนกลับ และเคลื่อนที่ในทิศทางตรงกันข้าม จากรูปที่ 2.5 คลื่นนิ่งสามารถแสดงให้เห็นด้วยผลลัพธ์ของการเสริมทับกันของคลื่นเคลื่อนที่เปรียบเสมือนการใช้ piezoelectric ceramic 2 ตัว เป็นตัวกำเนิด 2 แหล่ง เมื่อ piezoelectric ceramic ทั้ง 2 มีมุมเฟสต่างกัน 90 องศา และมีขนาดการสั่นที่เท่ากัน



รูปที่ 2.5 แสดงความสัมพันธ์ระหว่างคลื่นนิ่งและคลื่นเคลื่อนที่

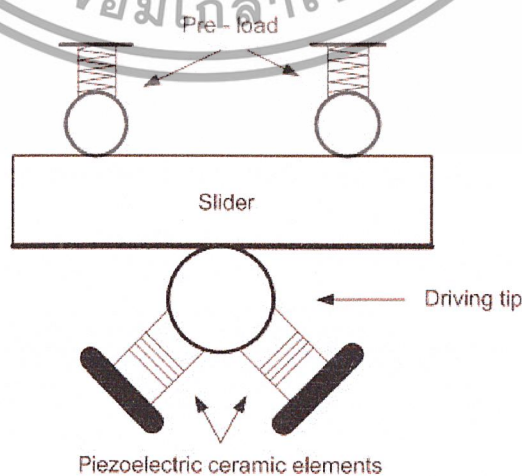
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ชนิดคลื่นนิ่ง



รูปที่ 2.6 แสดง โครงสร้างพื้นฐานของมอเตอร์ชนิดคลื่นนิ่ง

มอเตอร์ชนิดคลื่นนิ่ง คือ มอเตอร์ที่มีตัวกำเนิดการสั่นภายในเป็นชนิดคลื่นนิ่ง โดยหัวชุดจับ (P) ถูกเชื่อมต่อไปยังตัวกำเนิดการสั่น (V) และการทำงานในส่วนของหัวชุดจับ ณจุดที่หัวชุดจับสัมผัสกับส่วนที่เคลื่อนที่ (slider) จะมีลักษณะการเคลื่อนที่เป็นแบบวงรี ดังแสดงให้เห็นในรูปที่ 2.6 การเคลื่อนที่ที่ส่วนของหัวชุดจับ (P) ในขณะที่ตัวกำเนิดการสั่น (V) กำลังทำการสั่นอยู่ จะทำการจับเคลื่อนแกน X และแกน Y ตามปกติ ในขณะที่ตัวกำเนิดการสั่นทำงานนั้น ที่ส่วนปลายของชุดหัวจับจะเคลื่อนที่ในทิศทางตามแนวแกน X ของตัวกำเนิดการสั่น อย่างไรก็ตามในการเคลื่อนที่ในทิศทางตามแนวแกน y นั้นสามารถเคลื่อนที่ได้ก็เพราะว่าแรงผลัด F_t ซึ่งเป็นส่วนที่อยู่ในทิศทางของเส้นสัมผัสดวงของแรงที่ค้างอยู่ที่ส่วนของหัวจับมุมที่ลาดเอียง



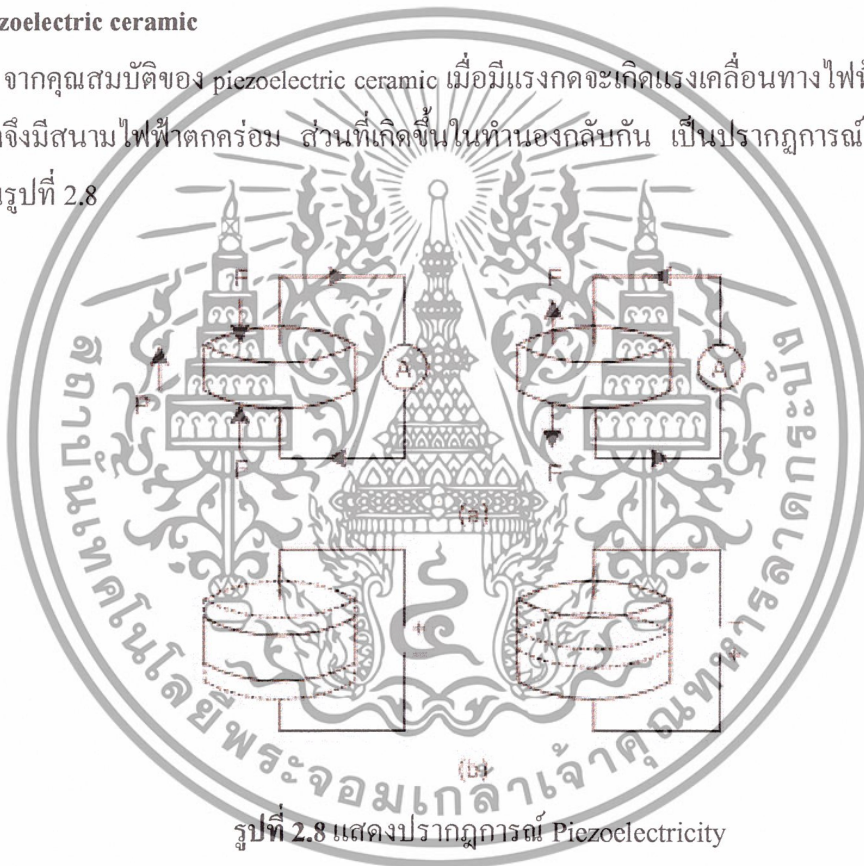
รูปที่ 2.7 แสดง โครงสร้างพื้นฐานของมอเตอร์ชนิดคลื่นนิ่ง โดยใช้ piezoelectric ceramic 2 ตัว เป็นตัวขับเคลื่อน
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่โดยไม่เสียค่าใช้จ่าย
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 แสดงถึงโครงสร้างของอัลตราโซนิคมอเตอร์อีกประเภทหนึ่งที่ใช้ตัวขับเร้าสองตัวต่ออยู่กับหัวชุดขับ ทำให้สามารถเคลื่อนที่ได้ทั้งสองทิศทาง คือซ้ายและขวา โดยมี Pre-load ต่ออยู่ทำหน้าที่เป็นตัวรับแรงที่กระดอนที่มาจากหัวชุดขับ

มอเตอร์ชนิดกลั่นนิ่งนั้นมีข้อดีคือ ต้นทุนการผลิตต่ำ มีโครงสร้างที่ง่าย และมีประสิทธิภาพการทำงานสูง

วัสดุ piezoelectric ceramic

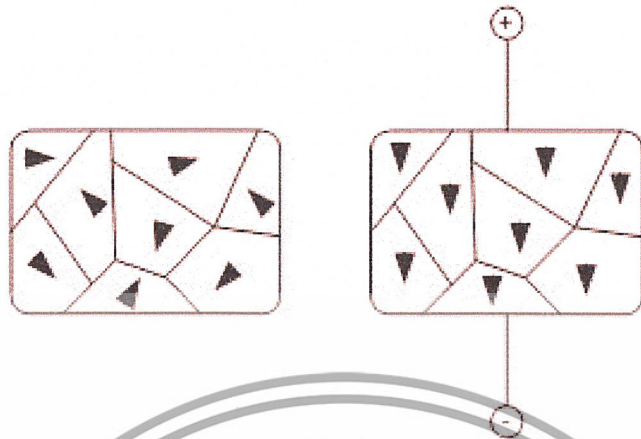
จากคุณสมบัติของ piezoelectric ceramic เมื่อมีแรงกดจะเกิดแรงเคลื่อนทางไฟฟ้าเกิดขึ้น และผลที่ตามมาจึงมีสนามไฟฟ้าตกคร่อม ส่วนที่เกิดขึ้นในทำนองกลับกัน เป็นปรากฏการณ์ piezoelectricity แสดงในรูปที่ 2.8



รูปที่ 2.8 แสดงปรากฏการณ์ Piezoelectricity

จากรูปที่ 2.8 จะเห็นว่าเมื่อมีสนามไฟฟ้าเกิดขึ้นเกิดขึ้นบน piezoelectric ceramic ตัวเซรามิกจะหดและขยายหรือนัยหนึ่งสารจะเป็นตัวเปลี่ยนพลังงานไฟฟ้าไปเป็นพลังงานกลและในทำนองกลับกัน piezoelectric ceramic จะมี polarization ตามธรรมชาติ ซึ่งแตกต่างจาก ferroelectric สาร piezoelectric ceramic หลังจากการ sinter ใหม่ๆ dipoles ที่มีอยู่ใน grains ต่างๆ นั้นไม่ได้เรียงไปในแนวทางเดียวกัน ดังนั้นจึงต้องมีกระบวนการ poling เพื่อให้ dipoles ต่างๆ เรียงไปในทิศทางเดียวกัน ดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



After high voltage is applied

รูปที่ 2.9 Poling of piezoelectric ceramics (จาก Catalog ของบริษัท Murata)

หากปราศจากกระบวนการ poling แล้ว สารเพียโซจะ ไม่แสดงสมบัติพิเศษใดๆ เลย สมการที่อธิบายเกี่ยวกับผลกระทบ ของ piezoelectric อาจเขียน ได้ดังนี้

$$D = dE + \epsilon^T E \tag{2.1}$$

$$S = s^E T + dE \tag{2.2}$$

เมื่อ D = electric displacement

T = stress

E = Electric field

S = strain

d = piezoelectric coefficient

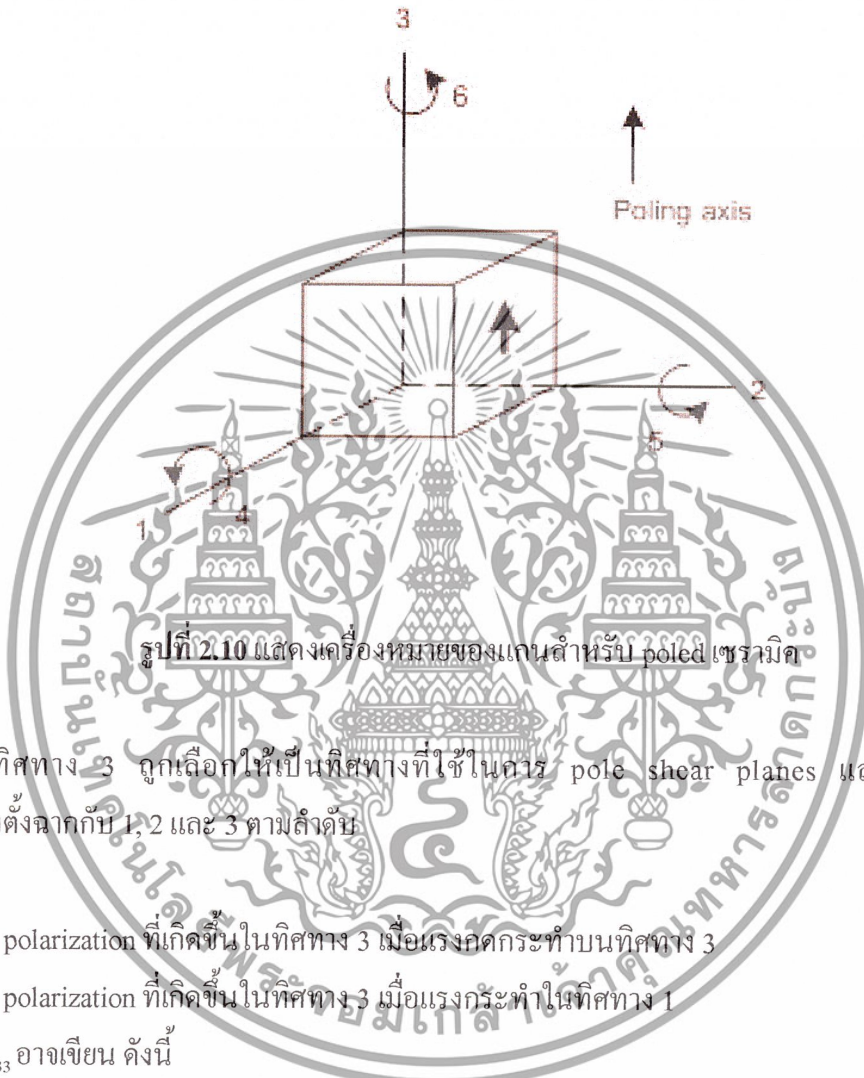
ϵ = dielectric constant

s = material compliance (modulus of elasticity)

สำหรับค่า ϵ^T หมายถึงค่า ϵ เมื่อ stress คงที่ หมายถึงว่าไม่มีแรงกดใดๆ บนสาร และ s^E คือค่า ความยืดหยุ่น ขณะที่สนามไฟฟ้าคงที่ สมการ (2.1) และ (2.2) อยู่ในรูป Matrix อธิบายเกี่ยวกับชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสมการที่เกี่ยวข้องกับสมบัติของสารในทิศทางต่างๆ ของวัสดุ เพื่อให้สะดวกในการอธิบายทิศทางต่างๆ ของสาร piezoelectric ceramics ถูกกำหนดไว้ในรูปที่ 2.10



รูปที่ 2.10 แสดงเครื่องหมายของแกนสำหรับ poled เซรามิค

ทิศทาง 3 ถูกเลือกให้เป็นทิศทางที่ใช้ในการ pole shear planes แสดงด้วย 4, 5 และ 6 ซึ่งตั้งฉากกับ 1, 2 และ 3 ตามลำดับ

d_{33} แสดง polarization ที่เกิดขึ้นในทิศทาง 3 เมื่อแรงกดกระทำบนทิศทาง 3
 d_{31} แสดง polarization ที่เกิดขึ้นในทิศทาง 3 เมื่อแรงกระทำในทิศทาง 1
 สมการ d_{33} อาจเขียน ดังนี้

$$D_3 = d_{33}T_3 \quad \text{(direct effect) (2.3)}$$

$$S_3 = d_{33}E_3 \quad \text{(converse effect) (2.4)}$$

ค่า d จะเท่ากันทั้งสองสมการ ในเชิงตัวเลขมีหน่วยเป็น 10^{-12} c/N (สำหรับ direct effect) และมีหน่วยเป็น 10^{-12} m/V (สำหรับ converse effect) สัมประสิทธิ์ piezoelectric จะเกี่ยวข้องกับสนามไฟฟ้าที่เกิดจากแรงเค้น อาจเขียนในรูปของ แรงดัน (g) มีหน่วยเป็น โวลต์ต่อนิวตันต่อตารางเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ในรูป

$$g = \frac{\text{open circuit electric field}}{\text{applied mechanical stress}} \dots\dots\dots(2.5)$$

ค่า g เป็นค่าที่มีค่า เกี่ยวกันกับ d ดังนี้

$$g = \frac{d}{\epsilon} = \frac{d}{\epsilon_r \epsilon_0} \dots\dots\dots(2.6)$$

- เมื่อ ϵ = permittivity
- ϵ_r = dielectric constant (relative permittivity)
- ϵ_0 = permittivity in free space (8.854×10^{-12} F/m)

ดังนั้น ถ้า ϵ_r ค่า g, d จะสูง ซึ่งมักจะเป็นเซรามิกที่ไม่ switch polarization ทันทันทันใด ทำให้ค่า ϵ_r ค่า กลุ่มสารเหล่านี้ใช้ใน gas ignitors หรือ lighters

สัมประสิทธิ์อีกตัวหนึ่ง ที่ใช้ประเมินค่าความสามารถในการเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล หรือในทางกลับกัน ได้แก่ piezoelectric coupling factor (k_{33} , k_{31} และ k_p) ค่า k นิยามว่า

$$K = \sqrt{\frac{\text{Energy Output}}{\text{Total Mechanical Energy Input}}}$$

สำหรับ direct effect และในทางกลับกันสำหรับ converse effect ทั้งสองกรณี K จะมีค่าน้อยกว่า 1 เสมอ

สำหรับ K_p (planar coupling coefficient) เป็นสัมประสิทธิ์ที่เกี่ยวกับความถี่ที่ maximum impedance parallel antiresonance, f_a และ minimum series resonance, f_s แล้วคำนวณตามสูตร

$$K_p = [(1 - K_p^2) \text{ (Bessel Function)}] \dots\dots\dots(2.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากความยุ่งยากสูตรในสมการที่ 2.6 ลดรูปเป็น

$$K_p = \left[2.5 \frac{(f_a^2 - f_r^2)}{f_r^2} + 0.038 \right] \dots\dots\dots (2.8)$$

ตัวอย่างของการประยุกต์

สาร piezoelectric ที่ใช้ในปัจจุบันมักอยู่ในกลุ่มของ BaTiO₃, PZT, PbN₂O₆, PT, PLZT, PMN และยังเปลี่ยนสารเจือทำให้ขอบเขตของสาร piezoelectric นี้กว้างขวางมากขึ้นในด้านการประยุกต์นั้น ในปัจจุบันได้มีการใช้อย่างกว้างขวาง เช่น

- piezoelectric buzzers เป็นแผ่นเซรามิกติดกับแผ่นโลหะบาง เมื่อมีสัญญาณไฟฟ้าประยุกต์ แผ่นเซรามิกก็จะสั่นเกิดเสียงขึ้น ใช้ใน alarms, sound producing calculators และ electronic watches
- Blood pressure sensors แผ่น piezoelectric จะรับสัญญาณของความดันโลหิตแล้วเปลี่ยนเป็นสัญญาณไฟฟ้า
- Ultrasonic sensors แผ่นเซรามิกอาจใช้ในการ sensor โดยไว้ที่หลังของรถเมื่อเวลาถอยหลังมีแผ่น ceramics จะสั่น ส่งคลื่นเสียงออกไป เมื่อคลื่นวิ่งชนวัตถุด้านหลังรถจะสะท้อนกลับแล้ว detect ด้วย ceramics ส่งสัญญาณบอกเป็นเสียงหรือแสงเตือน
- Filters การสั่น piezoelectric ceramics ขึ้นกับขนาดของตัวสาร ดังนั้นสัญญาณที่มีความถี่เท่ากับความถี่ธรรมชาติของตัวสารเท่านั้น จะทำให้ตัวสารสั่นอย่างแรง ด้วยสมบัติดังกล่าว จะสามารถกรองคลื่น Surface Acoustic ในบางความถี่ผ่านตัวสารเท่านั้น

ยังมีการประยุกต์หลายอย่าง ดังตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Generators (millivolts to kilovolts)	Motor/Generators (combination devices)
Hydrophones Microphones Gas ignitor Power supplies Photoflash actuators Sensors Piezoelectric pens	Sonar Ranging transducers Medical ultrasound Piezo transformers
Motors microns to millimetres Actuators Loud speakers Camera shutter Buzzers Ink jet printers Fish finder Ultrasonic motors	Resonant Devices kilohertz to megahertz Ultrasonic cleaners Ultrasonic welders Filters (SAW) Delay lines

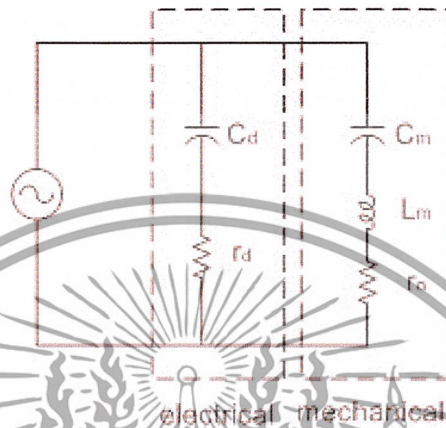
ตารางที่ 2.1 แสดงการนำ piezoelectric ceramic มาใช้งาน

2.4 การขับเคลื่อนอัลตราโซนิคมอเตอร์

piezoelectric ceramic แสดงให้เห็นว่าเป็นส่วนประกอบสำคัญมากที่สุดในอัลตราโซนิคมอเตอร์ piezoelectric ceramic จะทำตัวเป็นตัวกำเนิดการสั่น โดยการแปลงพลังงานไฟฟ้าที่รับเข้ามาเปลี่ยนเป็นพลังงานการสั่นเชิงกล

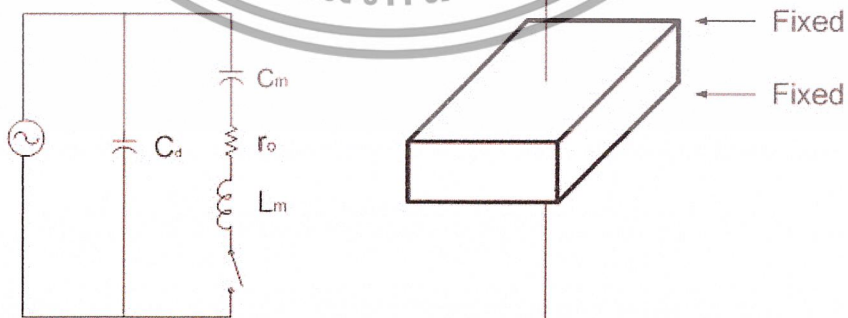
เพื่อที่จะได้เข้าใจว่า จะสามารถขับเคลื่อนอัลตราโซนิคมอเตอร์ได้อย่างไร ในส่วนนี้จะให้ความสนใจกับวงจรเสมือนของ piezoelectric ceramic โดยพิจารณาได้จากวงจรเสมือนพื้นฐานของ piezoelectric - เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ceramic ที่มีการสั้นอย่างอิสระในขณะที่ไม่มีโหลด โดยที่ความต้านทานทั้งสองตัวแสดงให้เห็นถึงความสูญเสียที่เกิดขึ้น ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 แสดงวงจรเสมือนพื้นฐานของอัลตราโซนิคมอเตอร์

C_d : เป็นค่าความเก็บประจุที่เกิดมาจากคุณสมบัติของความเป็นฉนวนทางไฟฟ้าของตัว piezoelectric ceramic ซึ่งถูกเรียกว่า Blocking Capacitance ในส่วนของ Blocking Capacitance อธิบายดังรูปที่ 2.12 Blocking Capacitance เป็นการวัดค่าความเก็บประจุในขณะที่ piezoelectric ceramic ถูกยึดติดอยู่กับที่ ซึ่งจะไม่มีกระแสเกิดขึ้น และในสภาวะนี้เองจะไม่มีกระแสเคลื่อนที่ที่เกิดขึ้นกับระบบ จึงเป็นการแสดงให้เห็นว่าไม่มีการทำงานในกิ่งของ C_m และ L_m หรือมีความต้านทานเป็นอนันต์



รูปที่ 2.12 แสดงถึง Blocking Impedance

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการทำวิจัยและพัฒนาเท่านั้น ไม่สามารถนำเอกสารไปใช้ประโยชน์อื่นใดได้
 เมื่อใช้แรงดันกระแสสลับป้อนให้แก่ piezoelectric ceramic ความหนาแน่น
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พลักซ์ไฟฟ้า(D) มีเฟสล่าหลังช้ากว่าสนามไฟฟ้า (E) โดยความหมายนี้เองฮีสเตอร์รีซิสถูกสร้างขึ้นระหว่าง E และ D และสร้างความร้อนออกมา ทำให้รู้ว่านี่คือค่าความสูญเสียที่เกิดขึ้นที่ฉนวนทางไฟฟ้า (Dielectric Loss) ถึงแม้ว่าความสูญเสียนี้สามารถละเว้นได้ที่ความถี่ต่ำ แต่ที่ความถี่สูง ๆ ไม่สามารถละเว้นได้ ตัวอย่างของการใช้ประโยชน์ของค่าความสูญเสียที่เกิดขึ้นที่ฉนวนทางไฟฟ้า ได้แก่ เต้าไมโครเวฟซึ่งใช้ความร้อนที่สร้างขึ้นจากค่าความสูญเสียที่เกิดขึ้นที่ฉนวนไฟฟ้าของมัน

ในการรวมกันของ C_d และ r_d ในรูปแบบของ $(r_d + \frac{1}{j\omega C_d})$ ถูกเรียกว่า Blocking-Impedance แต่สำหรับ r_d สามารถละเว้นได้สำหรับการใช้งานในย่านความถี่ของอัลตราโซนิค โดยจะยกเว้นค่า r_d ในรูปวงจรที่ 2.13

r_o : ดังที่รู้กันอยู่แล้วว่าปฏิกิริยา piezoelectricคือการทำให้สนามไฟฟ้าเรียงตัวไปในทิศทางเดียวกันจึงทำให้ piezoelectric ceramics ยึดออก โดยเกิดล้าหลังช้ากว่าสนามไฟฟ้าเล็กน้อยด้วยเหตุนี้เองทำให้เกิดความสูญเสียขึ้น และนอกจากนี้ความสูญเสียที่เกิดขึ้นเป็นค่าความสูญเสียทางกลที่เกิดขึ้นในส่วนประกอบต่าง ๆ เช่นการลื่นของแผ่นเหล็กที่ยึดติดกับตัว piezoelectric ceramic โดยความสูญเสียที่เกิดขึ้นส่วนใหญ่จะเกิดขึ้นที่ขอบทั้งสองข้างของ piezoelectric ceramic รวมทั้งที่เกิดขึ้นที่ สเตเตอร์ และขั้วทางไฟฟ้า ในวงจรเสมือนค่าความสูญเสียต่าง ๆ ที่เกิดขึ้นทั้งหมดถูกแสดงให้เห็นในรูปของตัวแปร r_o ซึ่งถูกเรียกว่า Internal Resistance เมื่อ r_o, C_m และ L_m รวมกันในรูปแบบของ $r_o + j\omega L_m + \frac{1}{j\omega C_m}$ นั้นแสดงให้เห็นถึง Motion Impedance

K : เมื่อแรงดันไฟฟ้ากระแสตรงถูกป้อนเข้าสู่ piezoelectric ceramic พลังงานไฟฟ้าจะถูกเก็บสะสมไว้ใน C_d และ C_m พลังงานที่สะสมไว้ใน C_m แสดงให้เห็นถึงความเครียดของ piezoelectric ceramic และแผ่นเหล็กที่ยึดติดอยู่กับตัวของมัน โดยค่ารากที่สองของ K คือสัมประสิทธิ์การเชื่อมต่อกับเครื่องกลไฟฟ้า (Electromechanical Coupling Coefficient) หาได้จากสมการ

$$K = \sqrt{\frac{C_m}{C_d + C_m}}$$

ในขณะที่เดียวกันที่การแกว่งของสนามไฟฟ้าที่ใช้เพื่อเป็นการกระตุ้นให้ piezoelectric ceramic เกิดการสั่นทางกล เมื่อปรับค่าความถี่ f_s จะเป็นการปรับแต่งเพื่อให้เกิดการสั่นทางกล โดยให้ความถี่ทางกล ω_0 ของตัว piezoelectric ceramic เป็นการสั่นด้วยขนาดที่สูงสุด ปรากฏการณ์นี้ถูกเรียกว่า “piezoelectric resonant” ซึ่งเป็นประโยชน์อย่างมากในการใช้ตัวกำเนิดการสั่นประเภทนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

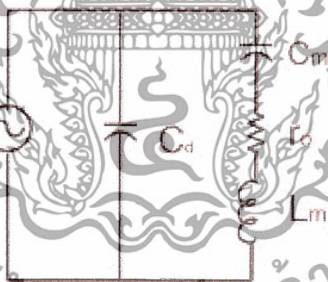
โดยความถี่เรโซแนนซ์ที่จะเกิดขึ้นขึ้นอยู่กับตัวแปรที่สำคัญคือ Q-factor เป็นตัวแปรที่ต้องกำหนดเพื่อให้ได้รับประสิทธิภาพสูงสุด อัลตราโซนิกมอเตอร์ควรจะถูกขับด้วยความถี่เรโซแนนซ์ที่เกิดขึ้นจากการคำนวณระหว่างค่า C_m และ L_m ในวงจรเสมือนดังรูปที่ 2.13 ความชัดเจนที่สุดของความถี่เรโซแนนซ์ ถูกแสดงให้เห็นโดย Q-factor ซึ่งกำหนดได้จาก

$$Q = \frac{\omega_o m}{r_o}$$

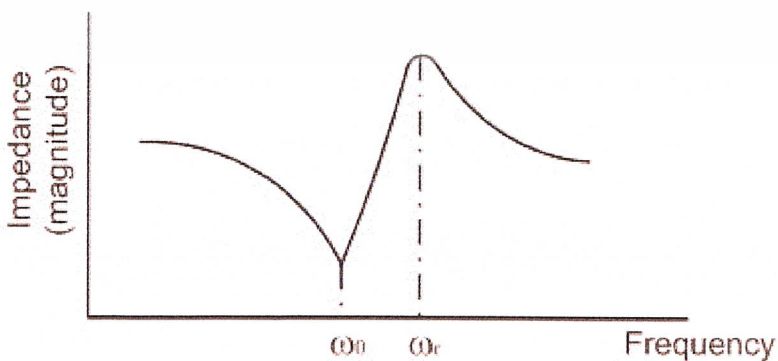
เมื่อ ω_o คือ มุมของความถี่เรโซแนนซ์ กำหนดได้โดย

$$\omega_o = \sqrt{\frac{1}{C_m L_m}} = \sqrt{\frac{K}{m}}$$

ที่ความถี่เรโซแนนซ์ ω_o ค่า อิมพีแดนซ์ของวงจรเสมือนจากรูปที่ 2.14 จะมีค่าน้อยที่สุด ในทางตรงกันข้าม ที่ค่าอิมพีแดนซ์มีค่ามากที่สุดถูกเรียกว่า “antiresonant frequency : ω_r ” ดังแสดงในเห็นดังรูปที่ 2.14

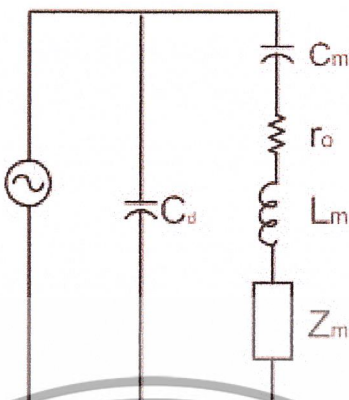


รูปที่ 2.13 แสดงวงจรเสมือนเมื่อไม่คิดค่า r_a



รูปที่ 2.14 แสดงจุดที่เกิดความถี่ Resonant frequency ω_o และ antiresonant frequency ω_r

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

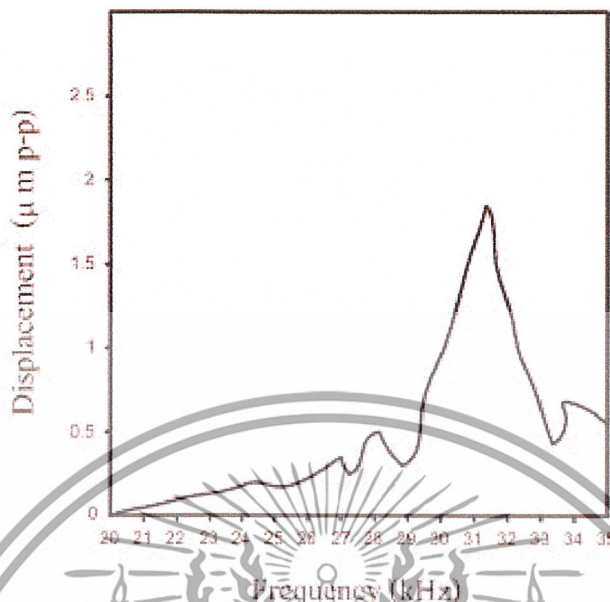


รูปที่ 2.15 แสดงให้เห็นถึงการต่อโหลดเข้ากับอัลตราโซนิคมอเตอร์

ในรูปที่ 2.15 เมื่อต่อโหลดเข้ากับอัลตราโซนิคมอเตอร์ Z_m แสดงให้เห็นถึงโหลด ซึ่งมีค่าเป็นความสูญเสียต่างๆ ที่เกิดขึ้น เช่น การเกิดความต้านทานระหว่างหัวชุดจับกับพื้น น้ำหนักของโหลด เป็นต้น

แรงดันที่จ่ายให้กับอัลตราโซนิคมอเตอร์ จะเป็นรูปคลื่นสัญญาณแบบใดก็ได้ เช่น สัญญาณไซน์ สัญญาณขั้นเลื่อย สัญญาณสามเหลี่ยม สัญญาณสี่เหลี่ยม แต่ต้องเป็นแรงดันไฟตรง เท่านั้น โดยแรงดันที่จ่ายให้กับอัลตราโซนิคมอเตอร์ จะทำให้เกิดผลของการยึดและหยดของอัลตราโซนิคมอเตอร์ ที่แตกต่างกันตามลักษณะของรูปสัญญาณ

การพิจารณาว่าจะเลือกใช้สัญญาณใดนั้น ขึ้นอยู่กับการนำไปใช้งานว่าในการใช้งานนั้นเหมาะสมกับงานลักษณะใด ในโครงการนี้ได้เลือกใช้สัญญาณรูปสี่เหลี่ยมเป็นสัญญาณที่จ่ายให้แก่อัลตราโซนิคมอเตอร์ ด้วยเหตุผลที่ว่า ต้องการให้อัลตราโซนิคมอเตอร์ยึดและหยดอย่างรวดเร็ว และข้อดีอีกข้อหนึ่งในการเลือกใช้สัญญาณสี่เหลี่ยมคือ วงจรไม่มีความซับซ้อนมากนัก โดยการเปลี่ยนแปลงขนาดของแรงดัน จะทำให้ความเร็วของมอเตอร์เปลี่ยนแปลงไปด้วย นี่เป็นวิธีการควบคุมความเร็วของมอเตอร์วิธีหนึ่ง ส่วนอีกวิธีที่สามารถทำได้คือการเปลี่ยนแปลงความถี่ที่ป้อนให้กับมอเตอร์ซึ่งมอเตอร์จะมีความเร็วสูงสุดที่ความถี่เรโซแนนท์ วิธีนี้เรียกว่า resonant drive แสดงให้เห็นดังรูปที่ 2.16

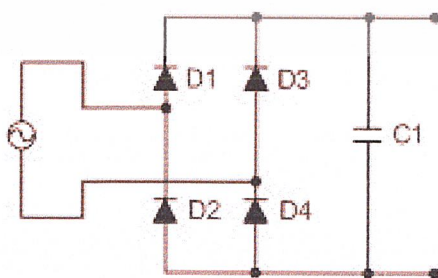


รูปที่ 2.16 แสดงระยะการเคลื่อนที่ ที่ความถี่ในแต่ละจุด

2.5 การออกแบบวงจรขับอัลตราโซนิคมอเตอร์ชนิดคลื่นนิ่ง

การออกแบบวงจรขับของอัลตราโซนิคมอเตอร์ชนิดคลื่นนิ่ง ใช้หลักการป้อนสัญญาณสี่เหลี่ยมให้กับตัว piezoelectric ceramic เพื่อให้เกิดการสั่นเชิงกล และทำให้มอเตอร์เคลื่อนที่ได้ โดยประกอบด้วยวงจรเรียงกระแสชนิดเต็มคลื่น วงจรกำเนิดความถี่ และ วงจร RC ซึ่งวงจรขับนี้สามารถควบคุมความเร็วของอัลตราโซนิคมอเตอร์ ได้ด้วยการเปลี่ยนแปลงแรงดันหรือความถี่

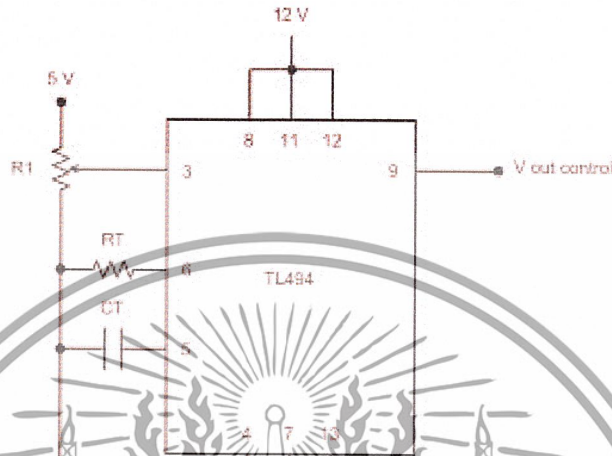
ภาควงจรเรียงกระแสชนิดเต็มคลื่นจะทำการแปลงแรงดันไฟสลับ ให้เป็นไฟตรงตามขนาดที่ต้องการ โดยจะมี Diode Rectifier 4 ตัว ทำหน้าที่แปลงแรงดันไฟสลับให้เป็นแรงดันไฟตรง แล้วก็จะนำไปผ่านส่วนของฟิวเตอร์ซึ่งมี C ต่ออยู่ เพื่อทำการปรับให้ไฟตรงให้เรียบมากขึ้น โดยแสดงดังรูปที่ 2.17



รูปที่ 2.17 แสดงวงจรเรียงกระแสชนิดเต็มคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของวงจรกำเนิดความถี่ ในโครงงานนี้ได้ใช้ไอซีเบอร์ TL494 ซึ่งเป็นไอซี pulse-width-modulation เป็นตัวสร้างสัญญาณพัลส์ ซึ่งสามารถปรับ duty cycle และ ความถี่ได้ อยู่ในช่วง 1-300 KHz เพื่อนำไปป้อนให้วงจรขับต่อไป โดยวงจรสร้างสัญญาณพัลส์แสดงดังรูปที่ 2.18



รูปที่ 2.18 แสดงวงจรกำเนิดความถี่ โดยใช้ IC TL494

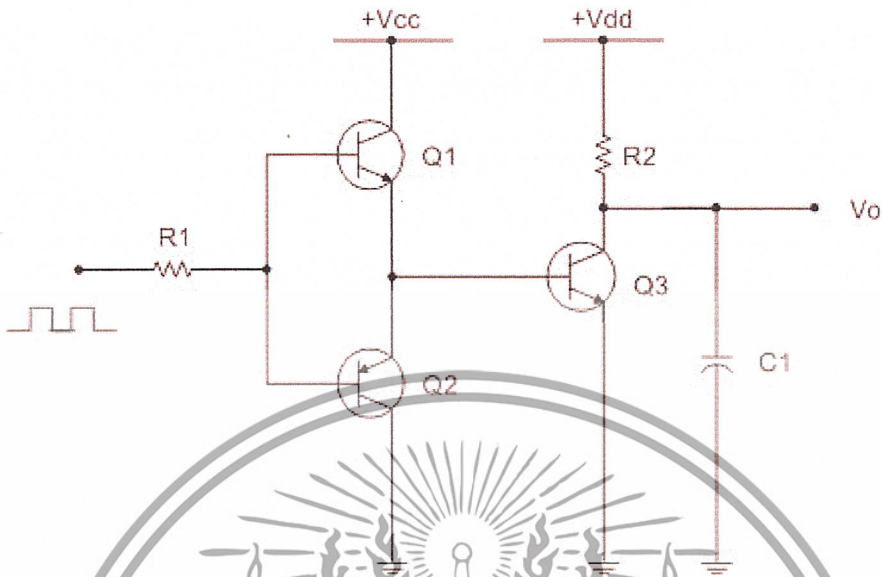
การกำหนดความถี่ f_{osc} สามารถทำได้โดยการปรับความถี่ที่ค่า R_T และ C_T จากสูตร

$$f_{osc} = \frac{1}{R_T \times C_T}$$

ในการปรับ duty cycle ทำได้โดยการปรับแรงดันที่ R_1 โดยจะมีแรงดันอยู่ในช่วง 0 – 3 V สำหรับ ไอซี TL 494 นี้ จะสามารถปรับความถี่ได้ โดยไม่มีผลให้ค่า duty cycle เปลี่ยนแปลงไป

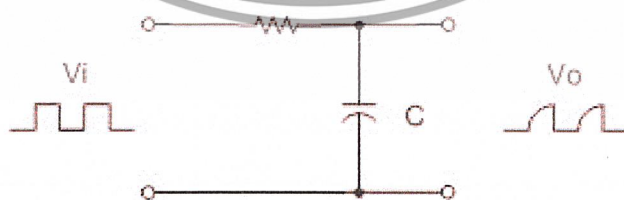
ในการออกแบบวงจรขับใช้ทรานซิสเตอร์ เพื่อเป็นสวิตซ์แรงดันสูง สำหรับวงจรขับเบสจะเป็นการใช้คู่ของวงจรอิมิตเตอร์ตาม (double emitter-follower) หรือ โทเท็ม โพล (totem pole) ซึ่งเป็นวงจรที่ประกอบด้วยคู่ไบโพลาร์ทรานซิสเตอร์แบบ NPN และ PNP เมื่อแรงดันด้านเข้าของวงจรขับเบสมีค่าสูง ทรานซิสเตอร์ Q1 จะเริ่มนำกระแสและ ทรานซิสเตอร์ Q2 ไม่นำกระแส เป็นผลทำให้ทรานซิสเตอร์กำลังเริ่มนำกระแส และเมื่อแรงดันด้านเข้าของวงจรขับเบสมีค่าต่ำ ทรานซิสเตอร์ Q1 จะไม่นำกระแสและ ทรานซิสเตอร์ Q2 นำกระแส ซึ่งจะเป็นการย้ายประจุจากขาเบสลงกราวด์ เป็นผลทำให้ทรานซิสเตอร์เริ่มหยุดนำกระแส โดยแสดงดังรูปที่ 2.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 แสดงวงจรขับ โดยใช้ทรานซิสเตอร์

ในการออกแบบในส่วนของวงจร RC ซึ่งเป็นระบบอันดับหนึ่ง เนื่องจากทรานซิสเตอร์ทำงานเป็นสวิตช์แรงดันสูง มีผลทำให้เกิดแรงดันกระชากขึ้นขอมาขึ้นมองเอาท์พุท จึงต้องใช้วงจร RC โดยแสดงดังรูปที่ 2.20 เพื่อลดแรงดันกระชากในส่วนของเอาท์พุท และเนื่องจาก piezoelectric ceramic มีลักษณะการทำงานในรูปแบบของตัวเก็บประจุ ดังนั้น piezoelectric ceramic จึงถูกนำมาใช้แทนตัวเก็บประจุ โดยแสดงความสัมพันธ์ได้จากสมการ



รูปที่ 2.20 แสดง วงจร RC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KVL

$$V_i(t) = V_R(t) + V_C(t)$$

$$V_i(t) = Ri(t) + \frac{1}{C} \int i(t) dt$$

ทำการแปลงลาปลาซ

$$V_i(s) = RI(s) + \frac{1}{CS} I(s)$$

$$V_i(s) = \left(R + \frac{1}{CS} \right) I(s) \dots \dots \dots (1)$$

เนื่องจาก

$$V_o(t) = V_C(t)$$

$$V_o(t) = \frac{1}{C} \int i(t) dt$$

ทำการแปลงลาปลาซ

$$V_o(s) = \frac{1}{CS} I(s)$$

$$\therefore I(s) = CSV_o(s) \dots \dots \dots (2)$$

นำสมการที่ (2) แทนลงใน (1)

$$V_i(s) = CSV_o(s) \left(\frac{RCS + 1}{CS} \right)$$

ดังนั้นทรานเฟอร์ฟังก์ชันของระบบคือ

$$\frac{V_o(s)}{V_i(s)} = \left(\frac{1}{RCS + 1} \right)$$

ซึ่งแสดงให้เห็นว่าเป็นระบบอันดับหนึ่งซึ่งเมื่อทำการแปลงลาปลาซย้อนกลับ เมื่อใช้แรงดันอินพุตเป็นสัญญาณสี่เหลี่ยมจะได้สมการดังนี้

$$V_o(t) = 1 - e^{-t/RC}$$

ซึ่งจากสมการนี้แสดงให้เห็นว่า RC เป็นตัวกำหนดค่าคงที่ของเวลาในการเข้าสู่สภาวะคงที่ เมื่อค่า RC มีค่าน้อย ๆ จะทำให้ระบบอันดับหนึ่งนี้เข้าสู่สภาวะคงที่ได้เร็วกว่าค่า RC ที่มีค่ามาก ๆ และจากสมการข้างบนถ้าแทนเวลา t ด้วยค่า RC จะได้แรงดันเอาต์พุตอยู่ที่ 63.2% ของแรงดันอินพุต แต่ถ้าให้ค่า t มีค่าเท่ากับ 4RC จะได้แรงดันเอาต์พุตอยู่ที่ 98.2% ของแรงดันอินพุต ด้วยเหตุนี้จึงได้นำเงื่อนไขไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาทำการคำนวณหาค่า R เนื่องจากค่าตัวเก็บประจุอยู่แล้วมีค่า เท่ากับ $750 \mu F$ ดังนั้นจึงคำนวณหาค่า R ได้ดังนี้

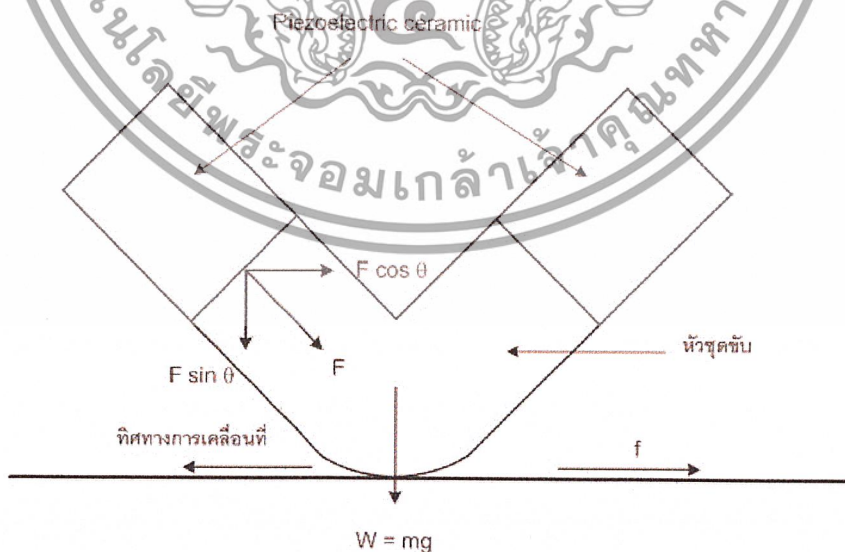
$$0.5T = 4RC$$

$$T = \frac{1}{f}$$

$$\therefore R = \frac{0.5}{4fC}$$

2.6 การออกแบบหุ่นยนต์ขนาดเล็กชนิดเคลื่อนที่

หลักการเคลื่อนที่ของหุ่นยนต์นั้น ใช้หลักการที่ว่าเมื่อ piezoelectric ceramic ทำงานจะเกิดการขยายตัวออกอย่างรวดเร็วทำให้เกิดแรงผลักกระทำกับพื้นผิว และเมื่อทำการแตกแรงนี้ออกตามแนวแกนนอนและแกนตั้งจะเห็นได้ว่า มีส่วนประกอบของแรงทั้งในแนวแกนนอนและแนวแกนตั้ง แสดงดังรูปที่ 2.21 แรงที่เกิดขึ้นในแนวแกนตั้งจะเกิดการเสริมกันกับน้ำหนักของหุ่นยนต์ที่ตกลงที่พื้น ส่วนแรงในแนวแกนนอนจะกลายมาเป็นแรงที่ทำให้เกิดการเคลื่อนที่ โดยการเคลื่อนที่จะเกิดจากแรงที่หวัชขึ้นมากระทบกับพื้นแล้วเกิดเป็นแรงสะท้อนหวัชกลับดันออกไปทางด้านที่แรงกระทำมา หรือเรียกว่าเป็นแรงปฏิกิริยานั่นเอง โดยแสดงดังรูปที่ 2.21



รูปที่ 2.21 แสดงหลักการการเคลื่อนที่ของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบหัวชุดขับของหุ่นยนต์ จะต้องออกแบบให้ส่วนที่สัมผัสกับพื้นผิวสัมผัสมีค่าน้อยที่สุดเพื่อลดแรงเสียดทานระหว่างหัวชุดขับ และพื้นผิว และต้องคำนึงถึงมุมที่เกิดระหว่างพื้นผิวกับตัว piezoelectric ceramic ที่ยึดติดอยู่กับหัวชุดขับเพราะมุมจะส่งผลโดยตรงกับแรงที่มากระทำกับพื้นผิวเพื่อทำให้เกิดแรงผลักดันให้ตัวหุ่นยนต์เคลื่อนที่ได้ ในส่วนของน้ำหนักของหุ่นยนต์ ในการออกแบบต้องออกแบบให้น้ำหนักของหุ่นยนต์มีขนาดที่มากกว่าแรงที่ หัวชุดขับกระทำกับพื้นผิวในแนวแกน y เพราะไม่เช่นนั้น เมื่อหุ่นยนต์ทำงานจะทำให้เกิดการสั่นในแนวแกน y หรืออาจจะทำให้หุ่นยนต์ไม่สามารถเคลื่อนที่ได้ ในส่วนที่เป็นเสมือนขาของหุ่นยนต์เพื่อให้หุ่นยนต์สามารถตั้งได้ จะต้องใช้วัสดุที่มีจุดสัมผัสกับพื้นน้อยที่สุด เพื่อไม่ให้เกิดแรงเสียดทานในขณะที่หุ่นยนต์เคลื่อนที่

2.7 การประยุกต์ใช้งานและแนวโน้มการพัฒนา

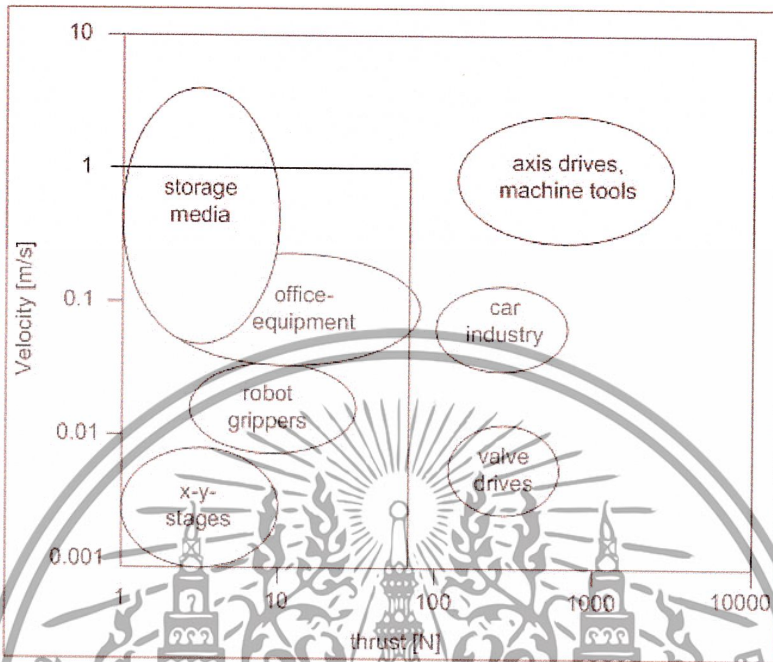
อัลตราโซนิกมอเตอร์เป็นสิ่งสำคัญ ที่ใช้มากในงานที่มีความละเอียดสูง รูปที่ 2.22 แสดงการนำอัลตราโซนิกมอเตอร์ ไปประยุกต์ใช้ในงานต่างๆ ซึ่งได้ครอบคลุมการทำงานทั่วไป

เห็นได้ชัดว่า อัลตราโซนิกมอเตอร์ถูกใช้ประโยชน์ในเครื่องมือเก็บข้อมูล หรืออุปกรณ์สำนักงานอื่นๆ โดยเฉพาะอย่างยิ่ง ในส่วนของหุ่นยนต์ขนาดเล็ก ส่วนการประยุกต์อื่นๆเช่น ในงานอุตสาหกรรมรถยนต์ที่ต้องการแรงและความเร็วสูง

ในปัจจุบัน อัลตราโซนิกมอเตอร์ถูกใช้ประโยชน์ในงานที่มีความละเอียดขนาดเป็นไมโคร เนื่องจาก ขนาดที่เล็ก ดังนั้น การออกแบบลักษณะของกรพัฒนาอัลตราโซนิกมอเตอร์แบบใหม่จะต้องถูกพิจารณาถึง

1. มีโครงสร้างง่าย ซึ่งจำนวนและส่วนประกอบมีขนาดเล็กได้ตามต้องการ
2. กระบวนการสร้างทำได้ง่าย
3. ตัวกำเนิดการสั่นและส่วนประกอบสำหรับวงจรขับเคลื่อนมีจำนวนน้อยสำหรับการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 การประยุกต์ใช้งานในงานด้านต่างๆ

2.8 ทฤษฎีการประมวลผลภาพ

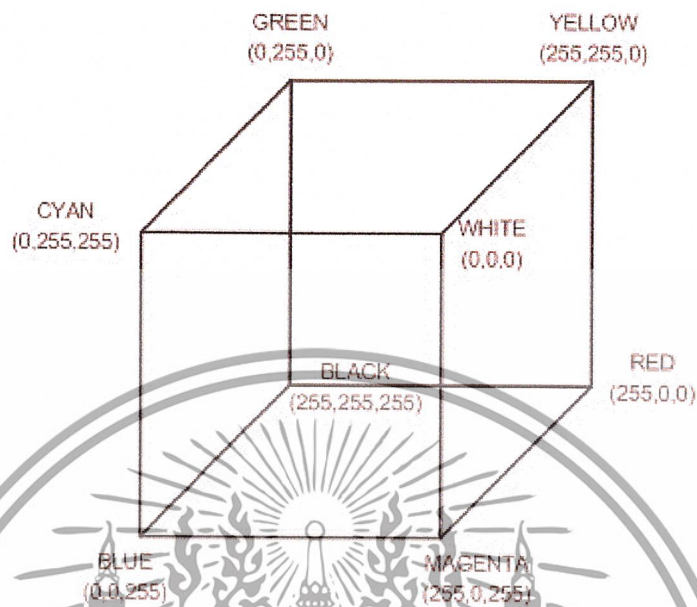
ทฤษฎีในการประมวลผลภาพนี้จะกล่าวถึงที่มาของสี ว่าสีที่ใช้งานในคอมพิวเตอร์กราฟฟิกมีรูปแบบและการนำไปใช้งานอย่างไร

2.8.1 การกำหนดสี

ในคอมพิวเตอร์ สีทุกสีจะประกอบด้วย 3 สีพื้นฐาน คือ แดง เขียว และน้ำเงิน (RGB) ที่สามารถจะออกแบบสีได้เกือบทุกสี โดยการผสมสัดส่วนของสีพื้นฐาน การออกแบบสีตั้งอยู่บนพื้นฐานของความเข้มขององค์ประกอบของ RGB ซึ่งเรียกว่า RGB โมเดล

ทุกๆสีสามารถสร้างขึ้นมาได้โดยผสมสัดส่วนที่เหมาะสมของสามสีพื้นฐาน ดังนั้นแต่ละสีจะถูกเสนอในสัดส่วน (Red, Green, Blue) ซึ่งสีพื้นฐานทั้งสามในแต่ละสีถูกนำเสนอด้วยทั้งสามไบต์ เป็นค่าที่เล็กที่สุด คือ ไม่มีค่าสีเลย ค่าที่ใหญ่ที่สุดคือ 255 บอกถึงความเข้มสูงสุด สัดส่วน (0,0,0) จะเป็นสีดำ เพราะทุกสีไม่มีความเข้มเลย และสัดส่วน (255,255,255) เป็นสีขาว ส่วนสีอื่นๆ สามารถรวมกันได้โดยอาศัยกระบวนการซึ่งมีส่วนประกอบพื้นฐานอยู่ด้วยกันสามส่วน ตั้งอยู่บนพื้นฐานของ RGB จากกล่องสีเหลี่ยมสีดังรูปที่ 2.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 กล่องสี RGB

กล่องสีสามมิติจะมีสีที่สอดคล้องกับสามสีพื้นฐานซึ่งมุมของสีเหลี่ยมแทนด้วยสีต่าง ๆ ดังรูป สีพื้นฐานอยู่ที่แกน X,Y และ Z ตามลำดับ สีที่เป็นส่วนกลับสามารถคำนวณได้อย่างง่าย ๆ โดยการลบค่าของสีจาก 255 ยกตัวอย่างสี (0,0,255) เป็นสีน้ำเงินบริสุทธิ์ ดังนั้นสีน้ำเงินและสีเหลืองจะเป็นสีในส่วนกลับกันและทั้งสองอยู่ที่มุมตรงกันข้ามของสีเหลี่ยมสี ส่วนมุมอื่นๆ ก็เช่นเดียวกัน

เมื่อเคลื่อนที่จากมุมหนึ่ง ไปยังอีกมุมหนึ่งตามขอบที่เหมือนกันจะมีแฉงค์ประกอบเดียวเท่านั้นที่เปลี่ยนแปลง ยกตัวอย่างเช่น เมื่อทำการเคลื่อนที่จากมุมสีเขียวไปยังมุมสีเหลือง องค์ประกอบของสีแดงจะเปลี่ยนแปลงจาก 0-255 เมื่อเคลื่อนที่วัตถุออกจากมุมสีเหล่านี้จะได้รับทุกโทนสีจากสีเขียวไปเหลือง

2.8.2 องค์ประกอบของสี

ในการประมวลผลภาพจะต้องทราบองค์ประกอบของสีเหล่านี้ นั่นคือต้องหาค่าพิกเซลแยกองค์ประกอบของ RGB จากนั้นนำไปใช้งานแยกกัน ดังนั้นจึงจำเป็นต้องเขียนฟังก์ชันซึ่งทำหน้าที่แยกองค์ประกอบของสี ซึ่งจะต้องพิจารณาว่าองค์ประกอบของสีทั้งสามถูกเก็บไว้อย่างไร ในตัวแปร LONG INTEGER ซึ่งตัวแปรดังกล่าวประกอบด้วย 4 ไบต์ โดยไบต์แรกเก็บค่า 0 ไบต์ต่อมาเป็นองค์ประกอบสีน้ำเงิน เขียวและแดง ตามลำดับค่า LONG INTEGER ซึ่งสอดคล้องกับสัดส่วน (64,32,192)

เป็น 12591168 จำนวนนี้ไม่มีความเหมือนกับจำนวนเต็มเลข ถ้านำเสนอองค์ประกอบของสีด้วยเลขฐานเอกซารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิบหกแล้ว RGB มีสัดส่วนเป็น (C0,20,40) ผลจะเป็นเลขฐานสิบหกของ LONG INTEGER สองหลัก สุดท้ายสอดคล้องกับองค์ประกอบสีแดง สองหลักถัดมาสอดคล้องกับองค์ประกอบสีเขียว และหลักที่มีความสำคัญสูงสุดสอดคล้องกับค่าสีน้ำเงิน และนี่ก็เป็นเหตุผลที่ว่าทำไมเลขฐานสิบหกจึงถูกใช้มาก ในการกำหนดค่าสี

STATEMENT ด้านล่างเป็นการแยกค่าขององค์ประกอบของสีจากค่าของสีที่ถูกเก็บในตัวแปร PIXEL

PIXEL = FORM PICTURE 1 POINT(j,i)

RED = PIXEL MOD 256

GREEN = ((PIXEL and &HFF00FF)/256)

BLUE = (PIXEL and &HFF0000/65535)

ตัวแปร i และ j เป็น Coordinate ของจุดซึ่งทำการตรวจสอบว่าค่าเนื่องจากค่าของสีแดงเก็บไว้ในไบต์สุดท้าย มันจึงเป็นเศษของการหารด้วย 256 เพื่อที่จะแยกไบต์ถัดไป จั๊วเป็นจะต้องทำให้ไบต์ด้านหน้าและไบต์ด้านหลังของมันเป็นศูนย์ ซึ่งทำได้โดยการใช้ตัวดำเนินการ AND กับเลขฐานสิบหก FF00FF00 ค่าของสีเขียวถูกเก็บไว้ในไบต์ที่สอง ดังนั้นเพื่อที่จะลดให้มันอยู่ที่ไบต์ที่หนึ่งนำค่าที่ได้หารด้วย 256

สุดท้ายทำเช่นเดียวกับไบต์ที่สาม โดยดำเนินการ AND ค่ำสีด้วย FF0000 และหลังจากนั้นหารผลที่ได้โดย 65536 ซึ่งจะเท่ากับค่า (256x256) หรืออาจใช้อีกโปรแกรมหนึ่งซึ่งทำงานได้เช่นเดียวกัน

Function color RGB (color aslong R,G,B as integer)

R = color1 mod256

G = color1\65536

B = (color1\256)mod256

End function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.3 การควบคุมกราฟฟิก (Manipulating Graphics)

ในส่วนนี้จะกล่าวถึงเทคนิคบางอย่างสำหรับการควบคุมการแสดงผลภาพ โดยอาศัย API Graphic functions ของ Visual Basic

- BitBlt() function

ฟังก์ชัน Bitblt ย่อมาจากคำว่า Bit Block Transfer เป็นการถ่ายเทข้อมูลระหว่างหน่วยความจำหนึ่ง ณ ตำแหน่งหนึ่งไปยังหน่วยความจำตำแหน่งอื่นๆ ฟังก์ชันนี้ถูกใช้เพื่อที่จะคัดลอกภาพจากต้นกำเนิด (Source) ไปยังจุดหมายปลายทาง (Destination) ฟังก์ชันนี้จะทำการส่งผ่านพิกเซลจากต้นกำเนิดไปยังอุปกรณ์ปลายทาง โดยสามารถประกาศ (declare) BitBlt() function ดังนี้

```
Public Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long
```

ในการประกาศฟังก์ชัน BitBlt() ซึ่งเป็นฟังก์ชันจากไดนามิกลิงก์ไลบรารีชื่อ gdi32.dll นั้นค่าอาร์กิวเมนต์ที่ฟังก์ชันนี้ต้องการคือ

ByVal hDestDC As Long ค่า hDC ของพื้นที่เป้าหมาย

ByVal X As Long ค่า x เพื่อกำหนดจุดมุมบนซ้ายของพื้นที่เป้าหมายที่จะให้ทำการ Blitting

ByVal Y As Long ค่า y เพื่อกำหนดจุดมุมบนซ้ายของพื้นที่เป้าหมายที่จะให้ทำการ Blitting

ByVal nWidth As Long ค่าความกว้างของพื้นที่เป้าหมายที่จะทำการ Blitting

ByVal nHeight As Long ค่าความสูงของพื้นที่เป้าหมายที่จะทำการ Blitting

ByVal hSrcDC As Long ค่า hDC ของพื้นที่อันเป็นต้นฉบับ

ByVal xSrc As Long ค่า x เพื่อกำหนดจุดมุมบนซ้ายของพื้นที่ต้นฉบับที่จะถูกใช้เป็นตัวต้นฉบับ

ByVal ySrc As Long ค่า y เพื่อกำหนดจุดมุมบนซ้ายของพื้นที่ต้นฉบับที่จะถูกใช้เป็นตัวต้นฉบับ

ByVal dwRop As Long ค่าคงที่ Raster Operation กำหนดรูปแบบการจัดการระดับบิต

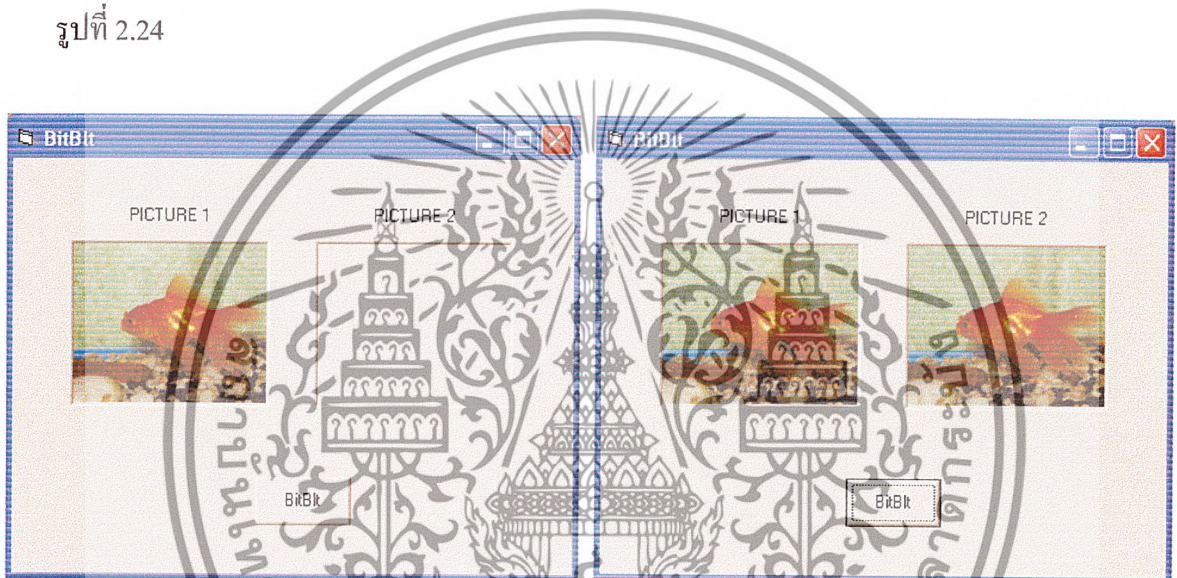
x และ y เป็น coordinate และถ้าจำเป็นต้องให้ความสูง และความกว้างพร้อมทั้งจุดเริ่มต้นของภาพต้นฉบับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง

BitBlt Picture2.hdc,0,0,160,150, Picture1.hdc,0,0,vbSrcCopy

Picture2 เป็นจุดหมายปลายทาง เริ่มต้นที่ $x=0$ และ $y=0$ โดยมีความกว้าง 160 และความสูง 150 ซึ่ง คัดลอกภาพจาก Picture 1 (ต้นกำเนิด) เริ่มคัดลอกที่ $x=0$, $y=0$ โดยคัดลอกภาพทั้งหมด โดยแสดงดังรูปที่ 2.24



รูปที่ 2.24 แสดงการใช้ฟังก์ชัน BitBlt Copy ภาพจากต้นกำเนิด ไปยังจุดหมายปลายทาง

StretchBlt() function

ฟังก์ชัน StretchBlt คล้ายกับฟังก์ชัน BitBlt ใช้ในกรณีที่ต้องการให้ภาพที่ปรากฏบนพื้นที่เป้าหมายมีขนาดที่ไม่เท่ากับภาพต้นฉบับ โดยการเพิ่มค่าอาร์กิวเมนต์ของภาพต้นฉบับเพิ่มขึ้นอีก 2 ตัว เพื่อทำภาพหดหรือยืดภาพตามขนาดที่กำหนด โดยแสดงดังรูป 2.25

สามารถประกาศ (declare) StretchBlt() function เป็นดังนี้

```
Public Declare Function StretchBlt Lib "gdi32" (ByVal hdc As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal nSrcWidth As Long, ByVal nSrcHeight As Long, ByVal dwRop As Long) As Long
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการประกาศฟังก์ชัน StretchBlt () ซึ่งเป็นฟังก์ชัน จากไดนามิกลิงก์ไลบรารีชื่อ gdi32.dll นั้น ค่าอาร์กิวเมนต์ที่ฟังก์ชันเหมือนกับฟังก์ชัน BitBlt แต่มีค่าอาร์กิวเมนต์ของภาพต้นฉบับเพิ่มขึ้นอีก 2 ตัว คือ

ByVal nSrcWidth As Long ค่าความกว้างของภาพต้นฉบับที่จะนำไปเป็นภาพต้นฉบับ
ByVal nSrcHeight As Long ค่าความสูงของภาพต้นฉบับที่จะนำไปเป็นภาพต้นฉบับ

ตัวอย่าง

```
StretchBlt Picture2.hdc,0,0,Picture2.ScaleWidth,Picture2.ScaleHeight,Picture1.hdc,0,0,
Picture1.ScaleWidth,Picture1.ScaleHeight,vbSrcCopy
```



รูปที่ 2.25 แสดงการใช้ฟังก์ชัน StretchBlt คัดลอกภาพจากต้นกำเนิดไปยังจุดหมายปลายทาง

- GetPixel () function

ฟังก์ชันจะคืนค่าของ pixel ที่ coordinate (x,y) ซึ่งสามารถประกาศได้โดย

```
Public Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal X As Long, ByVal Y As Long) As Long
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Setpixel () function

ทำหน้าที่กำหนดค่าของแต่ละ pixel หรือ pixel m กำหนด สามารถประกาศได้โดย

```
Public Declare Function SetPixel Lib "gdi32" (ByVal hdc As Long, ByVal X As Long, ByVal Y As Long, ByVal crColor As Long) As Long
```

โดย hdc คือ PictureBox ที่กำหนด

ตัวอย่าง

```
For i = 0 to 320
For j = 0 to 240
Color = GetPixel (Picture1.hdc,i,j)
Setpixel picture2.hdc,i,j,color1
Next i
Next j
```

จะเป็นการ copy ภาพจาก picture1 ไปยัง picture2 ครั้งละ 1 pixel

2.9 ความรู้เบื้องต้นของพอร์ตนาน

พอร์ตนาน (Parall Port) สาเหตุที่มีชื่อนี้ เนื่องจากการถ่ายทอดข้อมูลของพอร์ตเป็นแบบขนาน หรือเรียกอีกอย่างคือ พอร์ตเครื่องพิมพ์ (Printer Port)

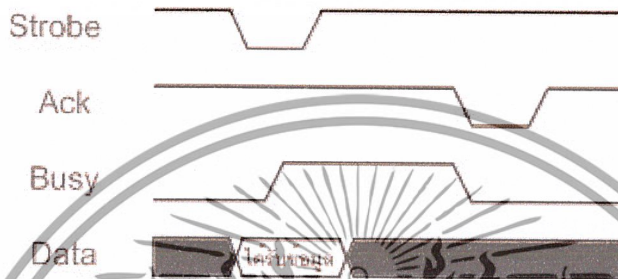
ด้วยการส่งแบบขนานนี้เอง ทำให้พอร์ตนานมีอัตราการถ่ายทอดข้อมูลสูงกว่าการถ่ายทอดข้อมูลแบบอนุกรมประมาณ 8-10 เท่า และการประมวลผลข้อมูลจะมีขนาด 8 บิต

2.9.1 ลักษณะทางกายภาพของพอร์ตนาน

เพื่อให้เข้าใจถึงการนำพอร์ตนานมาใช้งาน ก่อนอื่นต้องเข้าใจก่อนว่า ปกติการส่งพิมพ์งาน จากคอมพิวเตอร์ไปยังพอร์ตนานนั้นมีรูปแบบการทำงานภายในอย่างไร ในภาพรูปที่ 2.26 แสดงไทม์ไลน์ของเวลาของการติดต่อระหว่างพอร์ตนานและเครื่องพิมพ์ ในตอนแรก คอมพิวเตอร์จะส่งสัญญาณ Strobe ออกไปเพื่อให้เครื่องพิมพ์รู้ว่ามันมีข้อมูลกำลังจะส่งออกมาที่ขา Data จากนั้นคอมพิวเตอร์จะรอการตอบกลับจากเครื่องพิมพ์ นั่นคือเครื่องพิมพ์จะสร้างสัญญาณ Busy หรือ เพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเมื่อเครื่องพิมพ์พร้อม เครื่องพิมพ์จะส่งสัญญาณ Ack ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว



รูปที่ 2.26 แสดงไทม์ไลน์เวลาของการส่งข้อมูล ไปยังเครื่องพิมพ์

สัญญาณข้อมูลขนาด 8 บิต, สัญญาณ Strobe และสัญญาณ Ack (Acknowledge) เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้วในการติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่น ๆ ร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำงานถึง 3 อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์ พิมพ์ข้อมูลที่รับเข้ามา และตอบสนองการใช้งานของผู้ใช้ บางครั้งอาจเกิดเหตุการณ์ที่ผิดปกติเช่น บัฟเฟอร์สำหรับรับข้อมูล (เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ที่ทำงานทางกล ย่อมช้ากว่าการส่งข้อมูลของคอมพิวเตอร์) เครื่องพิมพ์ต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากไม่สามารถรับข้อมูลได้มากกว่านี้แล้ว สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์คือสัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่นกระดาษติด เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์เรียกว่าสัญญาณ Error นอกจากนี้เมื่อเครื่องคอมพิวเตอร์ต้องการรีเซ็ตเครื่องพิมพ์ คอมพิวเตอร์จะส่งสัญญาณ Reset ไปยังเครื่องพิมพ์เพื่อรีเซ็ตเครื่องพิมพ์ด้วย สามารถดูรูปขาสัญญาณที่จำเป็นสำหรับการติดต่อดังในตารางที่ 2.2

จากตารางที่ 2.2 จะเห็นได้ว่าพอร์ตขนานของคอมพิวเตอร์ยังแบ่งย่อยเป็นอีก 3 พอร์ต ได้แก่ พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตสำหรับสัญญาณ Strobe และ Reset พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge ,Busy และ Error จากเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิต	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมาใหม่	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลใหม่แล้ว	เครื่องพิมพ์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูล	เครื่องพิมพ์
Error	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	เครื่องพิมพ์
Reset	รีเซ็ตเครื่องพิมพ์	คอมพิวเตอร์

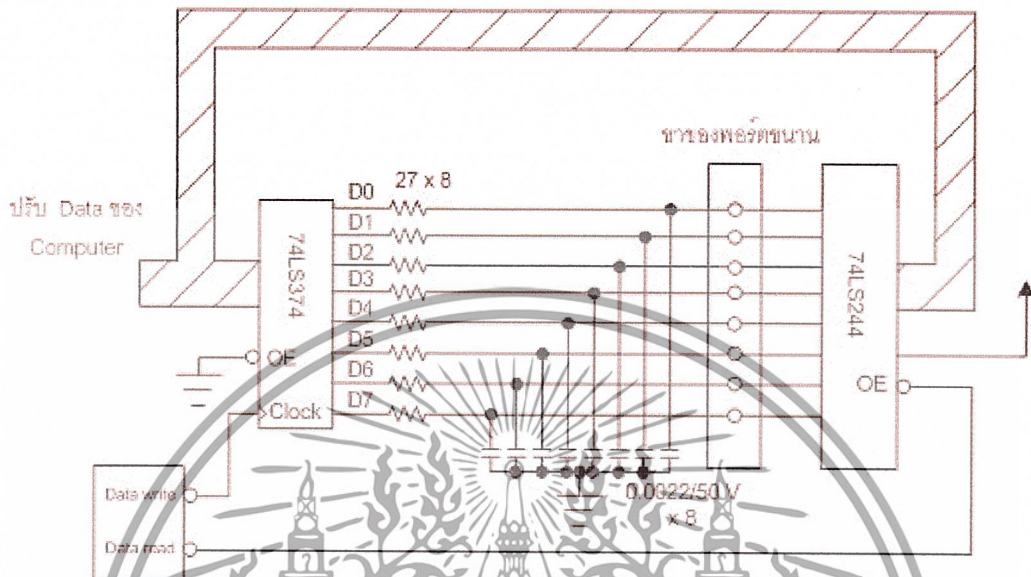
ตาราง 2.2 แสดงสัญญาณสำคัญของพอร์ตขนาน ที่ใช้ติดต่อกับเครื่องพิมพ์

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านั้น จะมีรีจิสเตอร์ 3 ตัวควบคุมการทำงาน ดังนี้

1. พอร์ตเอาต์พุตสำหรับสัญญาณข้อมูล 8 เส้น มีรีจิสเตอร์ Data ควบคุม
2. พอร์ตอินพุตสำหรับการอ่านค่าสถานะต่าง ๆ จากภายนอกมีอยู่ด้วยกัน 5 เส้น ใช้รีจิสเตอร์ Status ในการควบคุม
3. พอร์ตเอาต์พุตสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอก มีอยู่ด้วยกัน 4 เส้น ใช้รีจิสเตอร์ Control ในการควบคุม

2.9.2 พอร์ตข้อมูล (Data port)

จากรูปที่ 2.27 แสดงให้เห็นว่า พอร์ตข้อมูลจะประกอบไปด้วยบัพเฟอร์ 1 ตัว และไอซีแลตซ์อีก 1 ตัว เมื่อคอมพิวเตอร์ต้องการส่งข้อมูลไปยังเครื่องพิมพ์ คอมพิวเตอร์จะเขียนข้อมูลไปยังไอซีแลตซ์ ทั้ง 8 บิต เอาต์พุตของไอซีแลตซ์ 1 คือ D0-D7 ซึ่งเอาต์พุตนี้จะ ไปปรากฏอยู่ที่พอร์ตขนานในตำแหน่งขา 2 ถึง ขา 9 และที่ขาเอาต์พุตนี้สัญญาณ Data จะส่งกลับไปเป็นอินพุตของบัพเฟอร์ 1 ตัว ทำให้คอมพิวเตอร์สามารถอ่านค่าสถานะปัจจุบันที่เกิดขึ้นกับพอร์ตข้อมูลได้



รูปที่ 2.27 วงจรภายในของพอร์ตข้อมูล

เมื่อคอมพิวเตอร์ส่งข้อมูล ข้อมูลจะถูกส่งมาจากบัสข้อมูลของคอมพิวเตอร์ผ่านไปให้กับไอซี 74LS374 ซึ่งเป็นไอซีแลตช์ข้อมูล และเมื่อต้องการให้ข้อมูลปรากฏที่เอาต์พุต คอมพิวเตอร์จะส่งสัญญาณ Data Write ออกไปที่ขา CLK ของ 74LS374 เอาต์พุตจาก 74LS374 จะถูกกรองด้วยวงจร RC ซึ่งประกอบด้วยตัวต้านทานค่า 27 โอห์ม และตัวเก็บประจุ $0.0022 \mu F$ เพื่อให้ช่วงเวลาที่เปลี่ยนจากลอจิก "0" เป็นลอจิก "1" หรือจากลอจิก "1" เป็น ลอจิก "0" เป็นไปอย่างช้าๆ เนื่องจากการเปลี่ยนแปลงแรงดันที่รวดเร็วทำให้เกิดสัญญาณรบกวนเหนี่ยวนำข้ามไปยังข้อมูลบิตอื่นๆ ได้ ทำให้ข้อมูลที่ส่งออกไปมีข้อผิดพลาด จากค่าความต้านทานและตัวเก็บประจุในวงจรทำให้เกิดการหน่วงเวลาไปประมาณ 60 นาโนวินาที จากวงจรในภาพประกอบ ทำให้เอาต์พุตของพอร์ตข้อมูล มีคุณสมบัติดังนี้

- กระแสซิงค์สูงสุด 24 มิลลิแอมป์
- กระแสซอร์สสูงสุด 206 มิลลิแอมป์
- ระดับแรงดันของลอจิก "1" ต่ำสุดเท่ากับ 4 โวลต์
- ระดับแรงดันสูงสุดสำหรับลอจิก "0" เท่ากับ 0.5 โวลต์

สำหรับบัฟเฟอร์สำหรับการอ่านข้อมูลกลับได้แก่เบอร์ 74LS244 ซึ่งเมื่อต้องการอ่านค่าคอมพิวเตอร์จะส่งสัญญาณ Data Read ออกมาเพื่ออินพุตไอซี 74LS244 สำหรับพอร์ตขนานแบบมาตรฐาน (Standard Parallel Port :SPP) พอร์ตข้อมูล จะต้องใช้เพื่อการส่งค่าออก เอาต์พุตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.3 พอร์ตควบคุม (Control Port)

พอร์ตควบคุมใช้สำหรับควบคุมเครื่องพิมพ์ ตารางที่ 2.3 จะเห็นว่าพอร์ตควบคุม ประกอบไปด้วยบิตเอาต์พุต 4 บิตที่ต่อออกไปยังเครื่องพิมพ์ ส่วนบิตอินพุตอินเทอร์รัปต์ไม่ได้ถูกต่อออกไป รูปที่ 2.28 แสดงวงจรภายในของพอร์ตควบคุม เอาต์พุตของพอร์ตควบคุม มีอินเวอร์เตอร์แบบคอลเล็กเตอร์เปิดต่อรวมอยู่ โดยเอาต์พุตเหล่านี้จะถูกพูลอัพไว้ด้วยตัวต้านทานค่า 4.7 $\text{k}\Omega$ สำหรับบิต C2 ไม่มีกรกลับสถานะลอจิก

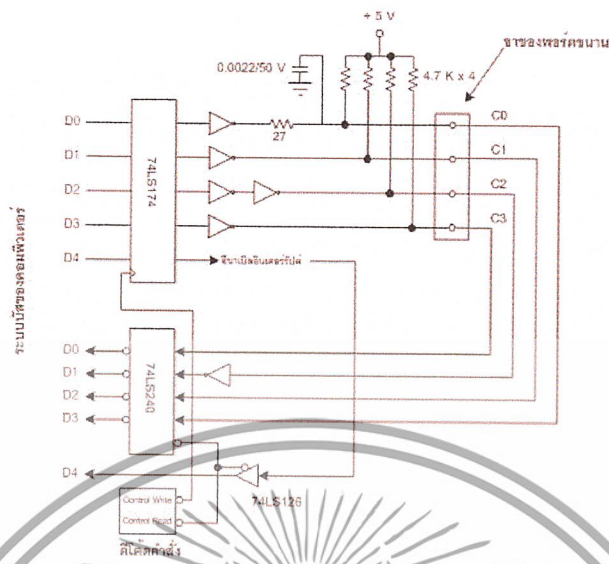
สถานะของพอร์ตควบคุม สามารถอ่านกลับได้โดยใช้ชิปเฟอร์ 74LS240 ซึ่งเอาต์พุตของ 74LS240 มีอินเวอร์เตอร์อยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าที่ส่งออกไป การควบคุมการอ่านและเขียนข้อมูลกับพอร์ตอนุกรม คอมพิวเตอร์จะส่งข้อมูลมาที่ขา Control Write และ Control Read

เนื่องจากเอาต์พุตของพอร์ตอนุกรม เป็นแบบคอลเล็กเตอร์เปิด ดังนั้นผู้ใช้งานสามารถใช้พอร์ตนี้ในการอ่านค่าสัญญาณอื่นพุดจากภายนอกได้ โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาพอร์ตที่ต้องการอ่านค่าลอจิก “1” เสียก่อน

DB-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ	หน้าที่การทำงาน
1	Control	Out	$\overline{C0}$	\overline{STROBE}	แอกทีฟ “0” ส่งค่าออกไปเพื่อบอกว่าที่ขา Data มีข้อมูลอยู่แล้ว
2-9	Data	Out	D1-D8	DATA1-DATA8	ขาส่งข้อมูลเอาต์พุตและรับข้อมูล
10	Status	In	S6	nACK	เป็นพัลซลอจิก “0” ที่ส่งมาจากเครื่องพิมพ์เพื่อบอกว่าได้รับข้อมูลที่ส่งไปแล้ว
11	Status	In	$\overline{S7}$	\overline{BUSY}	เป็นสัญญาณแจ้งมาจากเครื่องพิมพ์ว่ายังไม่พร้อมรับข้อมูล
12	Status	In	S5	PE	แจ้งกระดาษหมด
13	Status	In	S4	SELECT	แจ้งว่าเครื่องพิมพ์ต่ออยู่
14	Control	Out	$\overline{C1}$	$\overline{AUTO FEED}$	สั่งเครื่องพิมพ์ให้เลื่อนบันทึด
15	Status	In	S3	\overline{ERROR}	สัญญาณจากเครื่องพิมพ์มายังคอมพิวเตอร์เพื่อแสดงข้อผิดพลาดจากการพิมพ์
16	Control	Out	C2	\overline{INIT}	รีเซ็ตเครื่องพิมพ์โดยให้ลอจิก “0”
17	Control	Out	$\overline{C3}$	$\overline{SELECT - IN}$	ส่งสัญญาณไปยังเครื่องพิมพ์เพื่อแจ้งว่าต้องการเลือกเครื่องพิมพ์เครื่องนี้
16-25				GND	กราวด์

ตารางที่ 2.3 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน

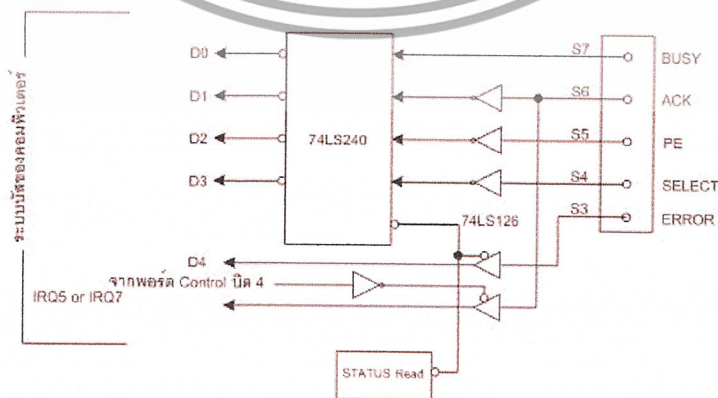
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 วงจรภายในของพอร์ตควม

2.9.4 พอร์ตแสดงสถานะ (Status Port)

พอร์ตสถานะเป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับการอ่านค่าสถานะจากคอมพิวเตอร์ รูปที่ 2.29 แสดงรายละเอียดภายในของพอร์ตสถานะจะสังเกตได้ว่ามีขาสัญญาณอยู่ทั้งหมด 5 สัญญาณด้วยกัน และจะเรียกชื่อเป็น S3,S4,S5,S6 และ S7 ซึ่งตัวเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้ภายในรีจิสเตอร์ Status นั่นเอง สำหรับ S7 จะมีชื่อแตกต่างจากบิตอื่นๆ ที่เมื่อสัญญาณภายนอกส่งเข้ามา แล้วจะไม่ผ่านอินเวอร์เตอร์ ในขณะที่ขาอื่นๆ ผ่านอินเวอร์เตอร์ทั้งหมด ดังนั้นเมื่อข้อมูลจากขาอินพุตไปยัง 74LS240 ซึ่งเอาทพุตมีการกลับสถานะทำให้บิต S7 เป็นบิตเดียวที่มีการกลับสถานะ นอกจากนี้ในการใช้งานถ้าต้องการให้มีการสร้างสัญญาณอินเตอร์รัปต์จากขอบขาขึ้นของเขา S6 สามารถกำหนดค่าได้จากพอร์ตควม บิต 4



รูปที่ 2.29 วงจรภายในของพอร์ตสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.5 การนำพอร์ตขนานไปใช้งาน

สำหรับพอร์ตขนานแบบมาตรฐานผู้ใช้งานสามารถนำพอร์ตอินพุต 5 บิต (พอร์ต status) พอร์ตเอาต์พุต 4 บิต (พอร์ต Control) และพอร์ตเอาต์พุตอีก 8 บิต (พอร์ตข้อมูล) ไปใช้งานได้โดยตรง โดยที่ 4 บิตของพอร์ตเอาต์พุตหรือพอร์ตควบคุม นั้นสามารถดัดแปลงให้ใช้งานเป็นพอร์ตอินพุตขนาด 4 บิตได้ด้วยดังนั้นผู้ใช้งานจึงสามารถนำสัญญาณจากพอร์ตขนานที่มีมากถึง 17 เส้นไปใช้งานในการควบคุมโดยใช้ระดับสัญญาณ TTL

● การเขียนโปรแกรมติดต่อกับพอร์ตขนาน

พอร์ตขนานของคอมพิวเตอร์จะมีลักษณะเช่นเดียวกับอุปกรณ์อินพุตเอาต์พุตตัวอื่น ๆ คือ เมื่อต้องการติดต่อก็ต้องกำหนดแอสเตรลที่ติดต่อกับพอร์ตด้วย ตาราง 2.4 แสดงแอสเตรลของพอร์ตขนาน โดยแบ่งออกเป็น 3 ตำแหน่งคือ แอสเตรลของรีจิสเตอร์ Data, รีจิสเตอร์ Status, และรีจิสเตอร์ Control โดยแอสเตรลนี้จะมียู่ 3 ชุด สำหรับพอร์ตขนาน 3 ชุดคือ LPT1, LPT2 และ LPT3

ชื่อพอร์ต	LPT1:		LPT2:		LPT3:	
	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก	ฐานสิบ	ฐานสิบหก
DATA	888	378H	956	3BCH	632	278H
STATUS	889	379H	957	3BDH	633	279H
CONTROL	890	37AH	958	3BEH	634	27AH

ตารางที่ 2.4 แสดงแอสเตรลของพอร์ตขนาน

เมื่อต้องการติดต่อกับพอร์ตขนานในตำแหน่งใด ก็ให้ส่งค่าข้อมูลออกไปที่พอร์ตขนานในตำแหน่งนั้น ๆ ยกตัวอย่างการเขียนโปรแกรมด้วย QBASIC เพื่อส่งค่าลอจิก “1” ออกไปทุกบิตของพอร์ตดาต้า ของ LPT1 ดังนั้นจะต้องเขียน โปรแกรมดังนี้

OUT &H378,&HFF โดยที่เครื่องหมาย &H ที่แสดงนั้นหมายถึงตัวเลขฐานสิบหก

คำสั่ง OUT เป็นการส่งค่าข้อมูลออกเอาต์พุตของพอร์ตอินพุตเอาต์พุต

คำสั่ง 378 เป็นแอสเตรลของรีจิสเตอร์ Data สำหรับ LPT1

ค่าข้อมูล FF เป็นข้อมูลเลขฐานสิบหก ซึ่งหมายถึงการให้บิตทุกบิตของรีจิสเตอร์ Data มีลอจิกเป็น “1” นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการอ่านค่าจากพอร์ตขนานมายังคอมพิวเตอร์ผ่านทางพอร์ตสถานะของ LPT1 สามารถเขียนโปรแกรมด้วย QBASIC ได้ดังนี้

Temp = INP(&H379) โดยที่

คำสั่ง INP() เป็นคำสั่งสำหรับการอ่านค่าข้อมูล

ค่า 379 เป็นตำแหน่งแอดเดรสของรีจิสเตอร์ Status สำหรับ LPT1 ในตัวเลขฐานสิบหก

ตัวแปร Temp เป็นตัวแปรที่ใช้เก็บข้อมูลที่อ่านได้จากพอร์ตขนาน

สำหรับโปรแกรมอื่น ๆ เช่น แอสเซมบลี, เทอร์โบปาสคาล หรือเทอร์โบซี จะมีรูปแบบการเขียนโปรแกรมที่แตกต่างกันบ้างอย่างไรก็ตาม โปรแกรมทุกตัวอย่างก็ใช้วิธีการเดียวกันคือ กำหนดแอสเซรตส์ที่ทำการติดต่อกันนั้นจึงติดต่อกับแอสเซรตส์เหล่านั้นด้วยคำสั่งสำหรับการอ่านหรือเขียน

การเขียนโปรแกรมด้วย Visual Basic ชุดคำสั่งส่วนใหญ่จะมีรูปแบบใกล้เคียงกับ QBASIC แต่ Visual Basic จะไม่มีคำสั่งสำหรับการติดต่อกับพอร์ตโดยตรงคือ คำสั่ง Inp() และคำสั่ง OUT เหมือนกับ QBASIC ดังนั้นเพื่อให้สามารถติดต่อกับพอร์ตขนานได้จึงจำเป็นต้องเพิ่มโปรแกรมบางตัวเข้า โดยโปรแกรมที่เพิ่มเข้าไปนี้จะอยู่ในรูปของ DLL (Dynamic Linked Library)

ไฟล์ DLL นี้จะมีอยู่ 2 ไฟล์คือ inpout.dll และ inpout32.dll โดยที่ inpout.dll นั้นใช้สำหรับระบบปฏิบัติการ 16 บิต หรือบนวินโดวส์ 3.1 นั้นเอง ส่วน inpout32.dll จะใช้กับระบบปฏิบัติการที่เป็น 32 บิต ซึ่งก็คือวินโดวส์ 95 หรือวินโดวส์ 98

สำหรับตำแหน่งที่ใช้เก็บไฟล์ inpout32.dll นั้นจะต้องเก็บไว้ที่ไดเรกทอรี SYSTEM ซึ่งอยู่ภายในไดเรกทอรีที่เก็บ โปรแกรมวินโดวส์ โดยส่วนใหญ่จะมีชื่อเป็น Windows

- การกำหนดค่าในโปรแกรมเพื่อเรียกใช้งานไฟล์ DLL มีรูปแบบการกำหนดค่าดังนี้
- สำหรับระบบปฏิบัติการ 16 บิต

```
Declare Function Inp% Lib "Input.dll" Alias "Inp16"(ByVal PortAddress%)
```

```
Declare Sub Out Lib "Iput.dll" Alias "Out 16" (ByVal PortAddress%, ByVal BtVal ByteToWrite)
```

- สำหรับระบบปฏิบัติการ 32 บิต

```
Public Declare Function Inp Lib "Inpout32.dll"
```

```
Alias "Inp32"(ByVal PortAddress As Integer) As Integer
```

```
Public Declare Sub Out Lib "inpout32.dll"
```

```
Alias "Out32" (ByVal PortAddress As Integer,ByVal Value As Integer)
```

ในกรณีที่มีการใช้งานโปรแกรมทั้ง 2 ระบบปฏิบัติการ สามารถเพิ่มคำสั่ง If เข้าไปเพื่อตรวจสอบระบบปฏิบัติการก่อนที่จะเลือกใช้งาน DLL ตัวที่ต้องการ โดยสามารถเขียนโปรแกรมได้ดังนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#If Win32 Then
"Declare Function Inp Lib "inout32.dll"
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inout32.dll"
Alias "Out32" (ByVal PortAddress As Integer,ByVal Value As Integer)
#Else
Declare Function Inp Lib "input.dll"(ByVal Port%) As Integer
Declare Sub Out Lib "Input.dll(ByVal Value%)
#End If
```

แต่การกำหนดในสองรูปแบบหลังนี้จะสามารถใช้งานได้กับ Visual Basic ตั้งแต่เวอร์ชัน 4 ขึ้นไปเท่านั้น

เมื่อมาถึงตรงนี้ อาจเกิดคำถามว่า ทำไมไมโครซอฟต์ผู้พัฒนาซอฟต์แวร์ Visual Basic จึงไม่รวมคำสั่ง Inp และคำสั่ง Out ไว้ในโปรแกรม Visual Basic เนื่องจากว่าการเขียนและอ่านข้อมูลไปยังพอร์ตหรือหน่วยความจำโดยตรงนั้นอาจทำให้เกิดปัญหาแอสคหรือทำงานผิดพลาดได้ และ Visual Basic เป็นระบบปฏิบัติการ ที่ทำงานบนวินโดวส์ซึ่งมีการทำงานแบบมัลติทาสกิ้ง(Multitasking) มีโปรแกรมหลาย ๆ ตัวทำงานอยู่พร้อมกัน ดังนั้นเมื่อเกิดความเสียหายกับโปรแกรมตัวหนึ่งอาจส่งผลให้โปรแกรมที่ทำงานอยู่ทั้งหมดเสียหายได้ นอกจากนี้การเขียนข้อมูลโดยตรงไปยังพอร์ต อาจจะไปทับซ้อนกับโปรแกรมอื่น ๆ ที่มีกรเขียนข้อมูลไปยังพอร์ตเดียวกัน ส่งผลให้โปรแกรมทำงานผิดพลาด

สำหรับวินโดวส์ 95 นอกจากสามารถใช้งาน DLL ในการติดต่อพอร์ตโดยตรงแล้ว ยังสามารถใช้งานในประเภท Visual Device Driver (VXD) ในการติดต่อกับอุปกรณ์อินพุตเอาต์พุต โดย VXD จะตัดปัญหาเรื่องการเข้าถึงพอร์ตพร้อมกันของโปรแกรมหลาย ๆ ตัวได้ แต่สำหรับโปรแกรมสั้น ๆ เช่น โปรแกรมอ่านค่าอุณหภูมิ โปรแกรมควบคุมอุปกรณ์อินพุตเอาต์พุตปกติ ซึ่งไม่มีการติดต่อกับพอร์ตอยู่ตลอดเวลา คำสั่ง Inp และ Out ใน DLL ก็ยังทำงานได้ดีและมีรูปแบบการใช้งานที่ง่ายกว่าฟังก์ชันที่มีอยู่ใน io.dll

PortOut	ใช้ส่งข้อมูลขนาด 1 ไบต์ ไปยังพอร์ตที่กำหนด
PortWordOut	ใช้ส่งข้อมูล 1 เวิร์ด (16 บิต) ไปยังพอร์ตที่กำหนด
PortDwordOut	ใช้ส่งข้อมูลดับเบิลเวิร์ด (32 บิต) ไปยังพอร์ตที่กำหนด
PortIn	ใช้อ่านข้อมูลขนาด 1 ไบต์ จากพอร์ตที่กำหนด
PortWordIn	ใช้อ่านข้อมูล 1 เวิร์ด (16 บิต) จากพอร์ตที่กำหนด
PortDwordIn	ใช้ส่งข้อมูลดับเบิลเวิร์ด (32 บิต) จากพอร์ตที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ผู้อื่นใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SetPortBit	ใช้เซตข้อมูลในระดับบิตของพอร์ตที่กำหนด
ClrPortBit	ใช้เคลียร์ข้อมูลในระดับของพอร์ตที่กำหนด
NotPortBit	ใช้กับข้อมูลในระดับของพอร์ตที่กำหนด
GetPortBit	ใช้อ่านสถานะของบิตที่กำหนด
RightPortShift	ใช้เลื่อนข้อมูลของพอร์ตไปทางขวา จะได้ค่าของบิต LSB กลับมา

ไฟล์ DLL นี้มีประโยชน์อย่างมากสำหรับ การติดต่อกับพอร์ตอินพุตเอาต์พุตของคอมพิวเตอร์ โดยเฉพาะอย่างยิ่งกับพอร์ตขนานของระบบปฏิบัติการวินโดวส์ ซึ่งในปัจจุบันการติดต่อกับพอร์ตขนานทำได้ยากขึ้น แต่เมื่อใช้ไฟล์ DLL ตัวนี้จะช่วยให้สามารถเข้าถึงและควบคุมได้ง่ายและมีประสิทธิภาพ io.dll สามารถนำไปใช้กับระบบปฏิบัติการวินโดวส์ 95/98/2000/NT หรือกระทั่ง XP จึงช่วยลดเวลาและขั้นตอนในการพัฒนาโปรแกรมสำหรับติดต่อพอร์ต ที่ระบบวินโดวส์ได้อย่างมาก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบหุ่นยนต์ชนิดเคลื่อนที่และระบบควบคุมหุ่นยนต์

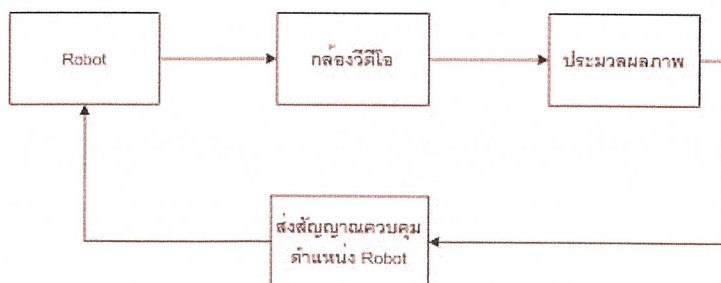
ภาพของหุ่นยนต์แต่ละตัวจะถูกรับภาพเข้ามาผ่านทางกล้องวิดีโอ ภาพที่ได้มานั้นจะมองเห็นในมุมมองส่วนด้านบน (Top View) ของหุ่นยนต์ เมื่อทำการประมวลผลภาพ (Image Processing) โดยคอมพิวเตอร์ เพื่อหาตำแหน่ง พิกัด และมุมของหุ่นยนต์ได้แล้วจากนั้นจะนำพิกัดที่ได้มาทำการสร้างอัลกอริทึมในการควบคุมการเคลื่อนที่ของหุ่นยนต์ไปยังตำแหน่งตามที่ต้องการ โดยมีโครงสร้างโดยรวมของระบบแสดงดังรูปที่ 3.1



รูปที่ 3.1 รูปแบบ โดยรวมของระบบ

จากโครงสร้างโดยรวมของระบบจะเห็นได้ถึงส่วนประกอบต่าง ๆ ของโครงงานนี้ซึ่งมีหน้าที่แตกต่างกันไป ซึ่งทุกส่วนจะถูกเชื่อมโยงเข้าหากัน โดยมีแนวคิดในการทำงานของระบบแสดงดังรูปที่

3.2



เอกสารนี้เป็นเอกสารที่สงวน **รูปที่ 3.2** แสดงบล็อกไดอะแกรมแสดงการทำงานของระบบ ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ส่วนประกอบของโครงการแบ่งออกได้เป็นส่วนต่าง ๆ ดังนี้

1. ส่วนประกอบทางด้านฮาร์ดแวร์ เป็นในส่วนของรูปแบบ และหลักการทำงานของหุ่นยนต์
2. ส่วนประกอบทางด้านซอฟต์แวร์ เป็นในส่วนของโปรแกรมผลภาพและอัลกอริทึมในการควบคุมหุ่นยนต์

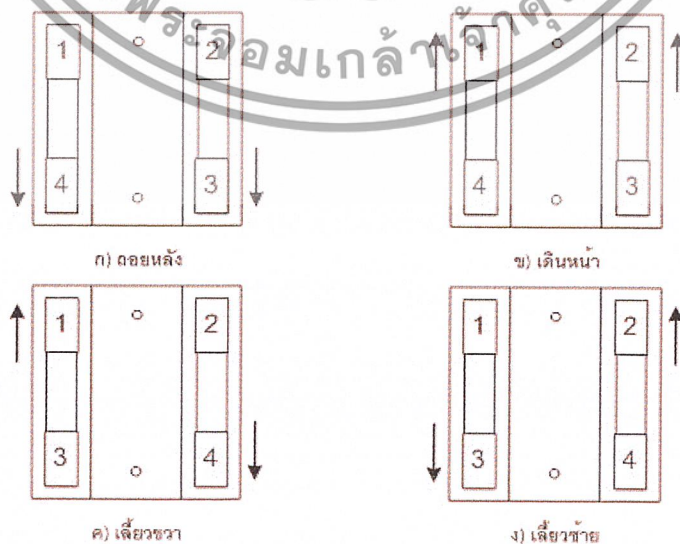
3.2 ส่วนประกอบทางด้านฮาร์ดแวร์

3.2.1 ส่วนของหุ่นยนต์

ในส่วนโครงสร้างของหุ่นยนต์ได้ออกแบบเป็นอัลตราโซนิคมอเตอร์ สองชุดต่อขนานกัน แสดงดังรูปที่ 3.3 ส่วนการเคลื่อนที่ของหุ่นยนต์สามารถเคลื่อนที่ได้ทุกทิศทาง อธิบายดังรูปที่ 3.4



รูปที่ 3.3 แสดงการติดตั้งมอเตอร์



รูปที่ 3.4 แสดงทิศทางการเคลื่อนที่ของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณและการสร้างวงจรขับอัลตราโซนิคมอเตอร์นั้นจำเป็นต้องทราบลักษณะต่างๆของสัญญาณที่ใช้ในการขับให้อัลตราโซนิคมอเตอร์ทำงาน

3.2.2 การขับอัลตราโซนิคมอเตอร์

อัลตราโซนิคมอเตอร์ที่ใช้ในโครงการนี้ต้องการสัญญาณที่ใช้ในการขับมีรูปสัญญาณสี่เหลี่ยม โดยมีแรงดันขนาด 80 Vrms ในช่วงความถี่อัลตราโซนิก จำนวน 4 สัญญาณ ที่มีเฟสเดียวกันจ่ายให้อัลตราโซนิคมอเตอร์แต่ละตัว ซึ่งการควบคุมความเร็วของอัลตราโซนิคมอเตอร์นี้มีด้วยกันสองลักษณะ คือ

1. ควบคุมความเร็วด้วยการเปลี่ยนความถี่
2. ควบคุมความเร็วด้วยการเปลี่ยนขนาดแรงดัน

ในโครงการนี้ได้สร้างวงจรขับอัลตราโซนิคมอเตอร์ ซึ่งสามารถควบคุมความเร็วได้ด้วย การควบคุมความถี่ เพื่อให้หุ่นยนต์เคลื่อนที่ได้ด้วยความเร็วและตำแหน่งตามที่ต้องการ

3.2.3 โครงสร้างของวงจรขับอัลตราโซนิคมอเตอร์

วงจรขับอัลตราโซนิคมอเตอร์ประกอบด้วยส่วนสำคัญ 3 ส่วนคือ วงจรเรียงกระแสชนิดเต็มคลื่น วงจรกำเนิดความถี่ วงจรขับ

● ส่วนของวงจรเรียงกระแสชนิดเต็มคลื่น

จากหลักการที่ได้กล่าวมาแล้วในบทที่สอง หัวข้อที่ 2.5 ด้วยวงจรเรียงกระแสชนิดเต็มคลื่นนี้ได้ทำการสร้างวงจรที่ใช้จริงและมีการคำนวณค่าต่างๆดังนี้

คำนวณหาค่าตัวเก็บประจุ (C) ได้จาก

$$C = \frac{P}{f(V_{\max}^2 - V_{\min}^2)}$$

เมื่อ P คือ กำลังของโหลดที่ใช้ 80 W

f คือ ค่าความถี่ของแรงดันไฟสลับที่ป้อนให้กับวงจรมีค่าเท่ากับ 50 Hz

ΔV คือ ความต่างแรงดันค่าที่แกว่งสูงสุด และค่าที่แกว่งต่ำสุดของแรงดัน V_{dc} มีค่าเท่ากับ

10 V เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V_{\max} คือ ค่าความแวงสูงสุดของ V_{dc} ที่ยอมรับได้มีค่าเท่ากับ $56\sqrt{2}$

V_{\min} หาได้จาก $V_{\max} - \Delta V$ ซึ่งจะมีค่าเท่ากับ $56\sqrt{2} - 10$

จากการคำนวณ ได้ค่าตัวเก็บประจุ เท่ากับ $1066 \mu F$ ดังนั้นค่าตัวเก็บประจุที่ใช้ต้องมากกว่าค่าที่คำนวณได้ จึงเลือกใช้ค่าตัวเก็บประจุที่ค่า $8000 \mu F$ และทนแรงดันได้มากกว่า $56\sqrt{2}$ V.

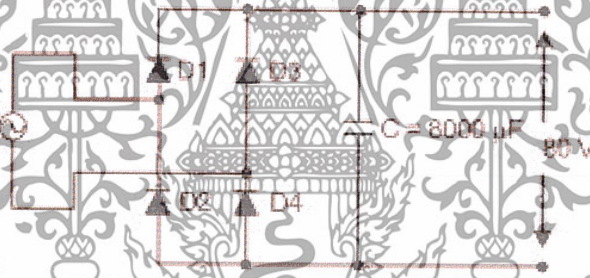
โดยค่า V_{dc} หาได้จาก

$$V_{dc} = \sqrt{2} \tilde{V}$$

เมื่อ \tilde{V} คือ ค่าแรงดันไฟสลับที่ป้อนให้กับวงจรเรียงกระแสชนิดเต็มคลื่น

ในที่นี้ต้องการแรงดัน V_{dc} ที่มีขนาดเท่ากับ 80 V. เพื่อจ่ายให้กับวงจรขับ ดังนั้น \tilde{V} จะต้องมีขนาดแรงดันเท่ากับ 56 V.

ดังนั้นสามารถนำไปต่อใช้งาน ดังรูปที่ 3.5



รูปที่ 3.5 รูปวงจรเรียงกระแสชนิดเต็มคลื่น

- ส่วนของวงจรกำเนิดความถี่

จากสมการที่กล่าวมาในบทที่ 2 ต้องการหาความถี่ที่ใช้สามารถคำนวณได้จาก

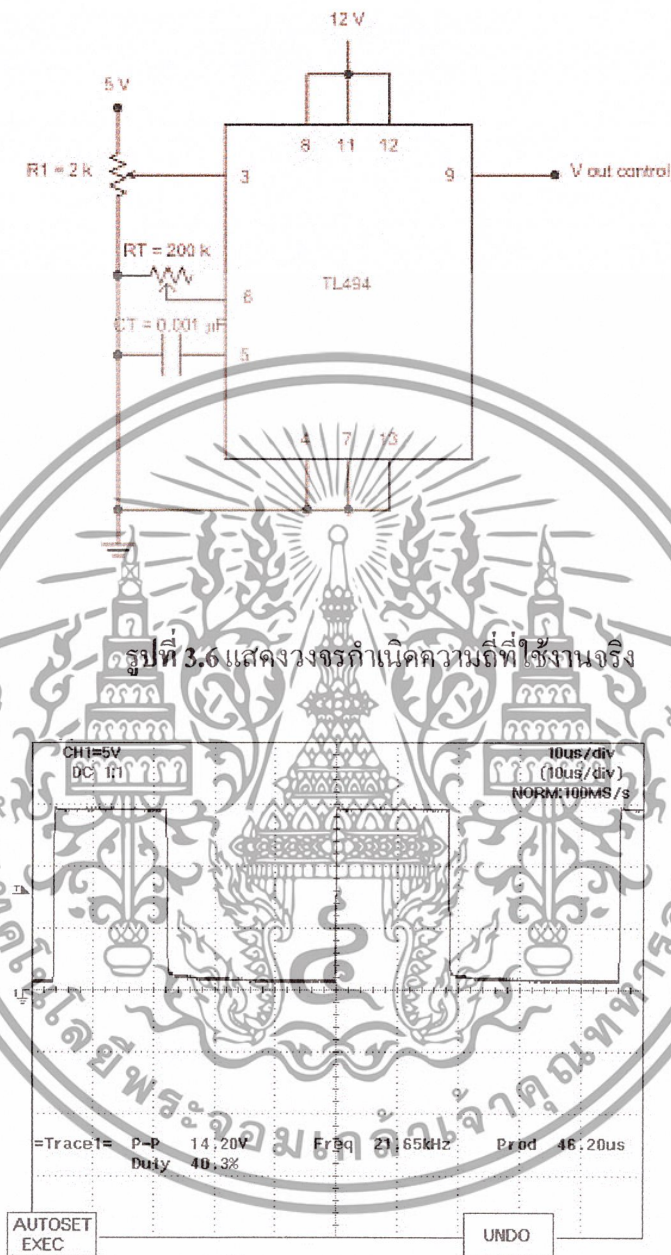
$$f_{osc} = \frac{1.1}{R_T \times C_T}$$

เมื่อ f_{osc} คือ ความถี่ที่ต้องการ 20KHz

C_T กำหนดให้มีค่าเท่ากับ $0.001 \mu F$

ดังนั้น R_T ที่ได้จากการคำนวณมีค่าเท่ากับ $55 K\Omega$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 รูปสัญญาณที่ได้จากวงจรกำเนิดสัญญาณสี่เหลี่ยม

ในการจะเปลี่ยนแปลงความถี่ที่ใช้งานต้องเปลี่ยนค่าความต้านทาน R_T จะทำให้ความถี่เปลี่ยนแปลงไป สำหรับในวงจรใช้ความต้านทานแบบปรับค่าได้ $200\text{ K}\Omega$ สามารถปรับความถี่ได้ตั้งแต่ $0-40\text{ KHz}$ โดยเมื่อความถี่เปลี่ยนแปลงไป ค่า duty cycle จะไม่เปลี่ยนแปลงไปตามความถี่ แต่สามารถปรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า duty cycle ได้จากการปรับค่าแรงดันที่ตกคร่อมความต้านทานที่ขา 3 (R_1) ของ TL494 อยู่ในช่วง 30-80% ดังในรูปวงจรที่ 3.6 ส่วนสัญญาณที่ได้จากวงจรสร้างสัญญาณพัลส์แสดงดังรูปที่ 3.7

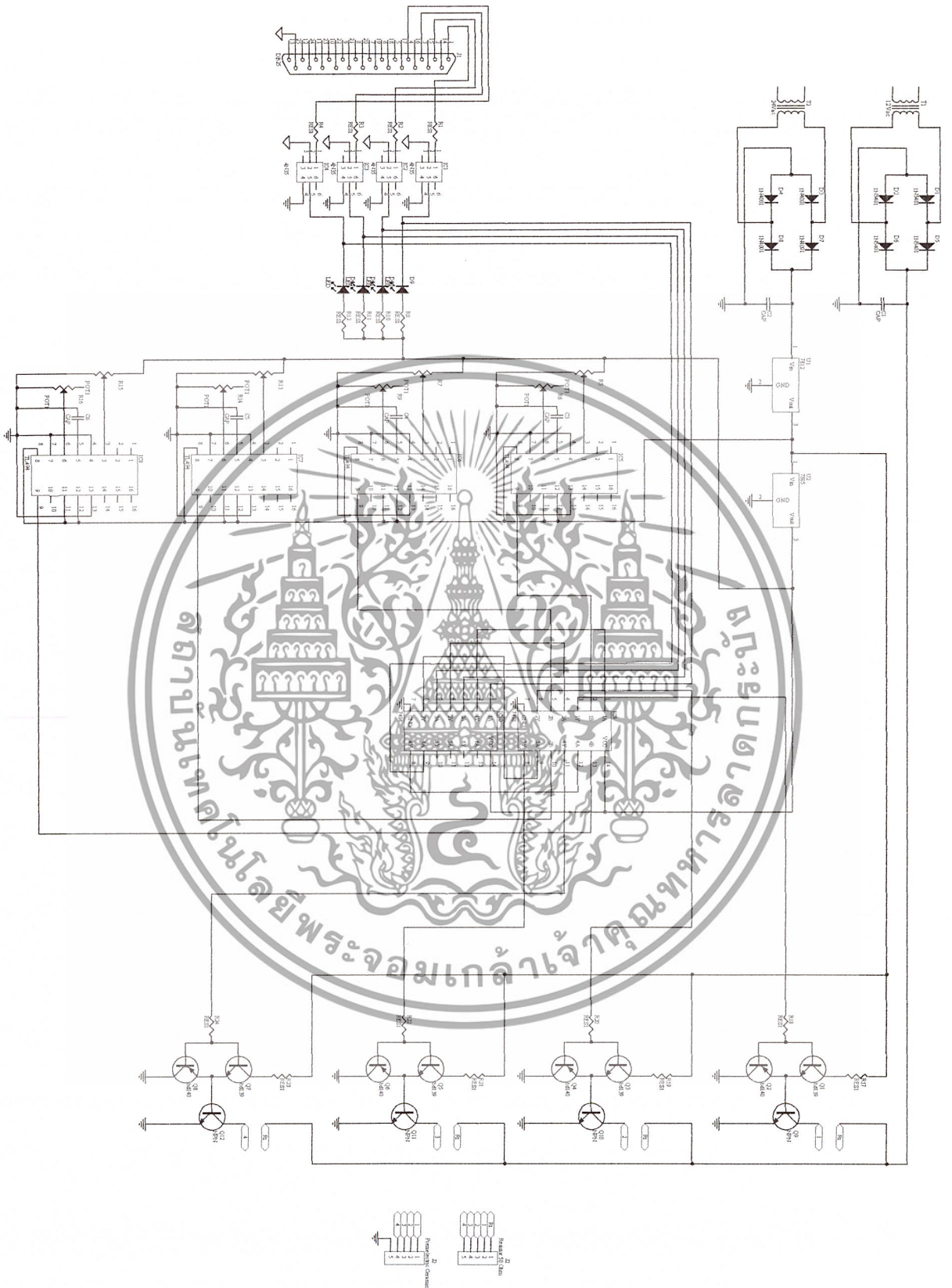
- ส่วนของวงจรขับ

การขับ piezoelectric ceramic ให้ทำงาน เมื่อได้พัลส์จากวงจรกำเนิดความถี่แล้วจะนำสัญญาณพัลส์ที่ได้นำมาทำการ AND กับสัญญาณที่ส่งมาจากคอมพิวเตอร์ เพื่อทำการเลือกขับ piezoelectric ceramic แต่ละตัว โดยจะใช้ ทรานซิสเตอร์ เบอร์ TIP 122 ซึ่งเป็นทรานซิสเตอร์ต่อภายในแบบคาร์ลิงตัน ทำหน้าที่เป็นสวิตซ์แรงดันไฟสูงจ่ายไฟให้กับ piezoelectric ceramic โดยจะมีวงจร RC ทำหน้าที่กรองความถี่ต่ำผ่านเพื่อลดการกระชากของแรงดันเมื่อทำการอนทรานซิสเตอร์ โดยวงจรทั้งหมดที่กล่าวมาเมื่อนำมาใช้งานจริงแสดงดังรูปที่ 3.8 และวงจรรวมทั้งหมดแสดงดังรูปที่ 3.9



รูปที่ 3.8 แสดงภาพวงจรที่ใช้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



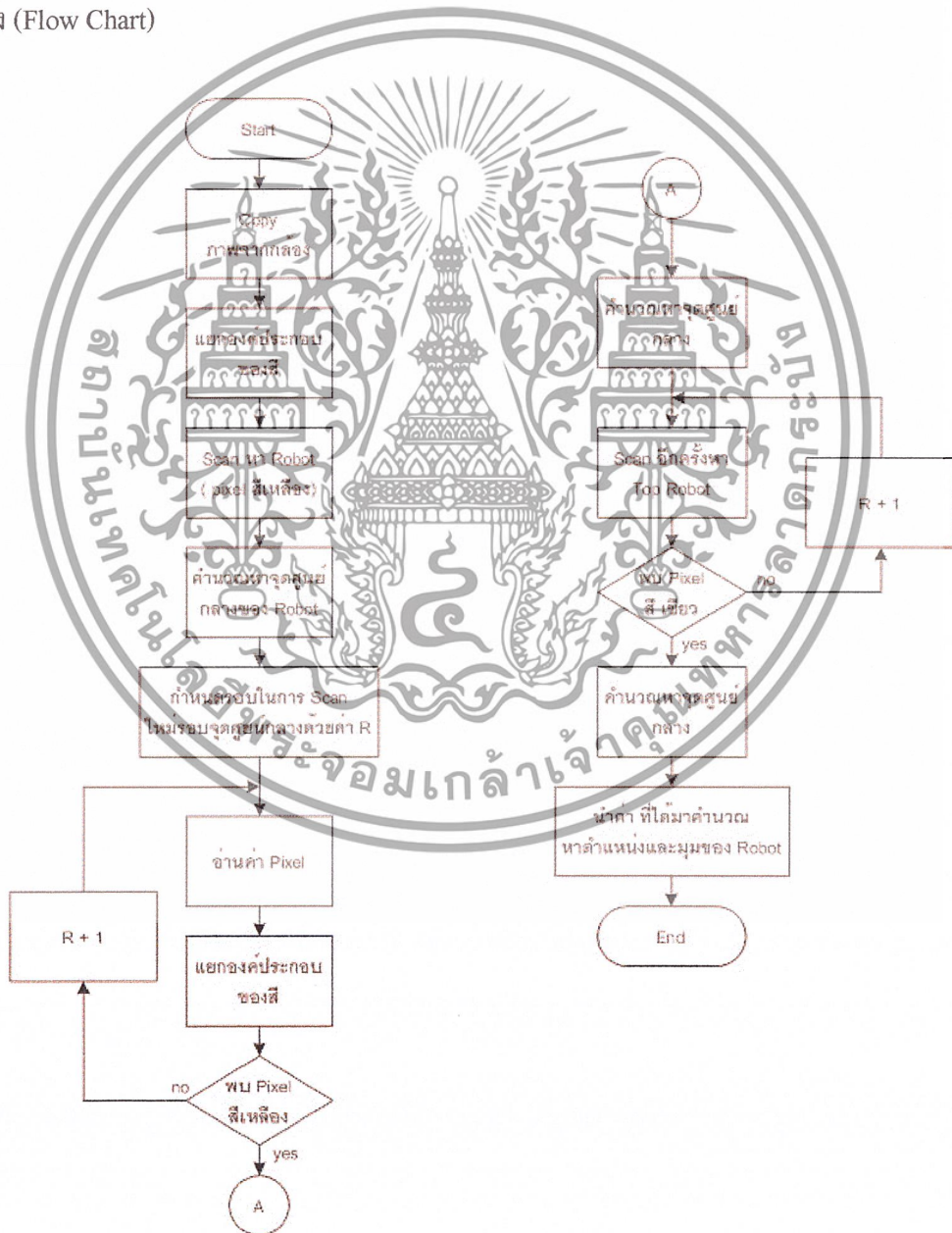
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 3.9 แสดงวงจรรวมที่ใช้จริง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนประกอบทางด้านซอฟต์แวร์

ส่วนประกอบทางด้านซอฟต์แวร์แบ่งได้ออกเป็นสองส่วน

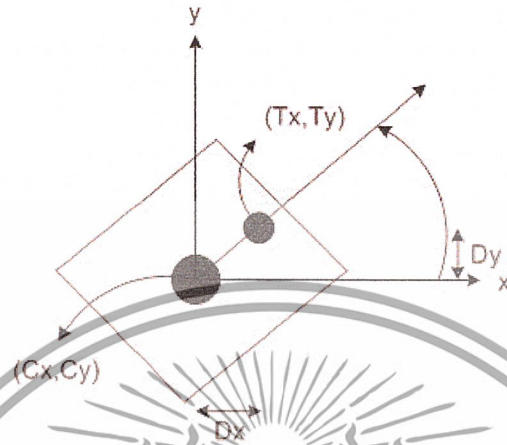
3.3.1 ส่วนประมวลผลภาพ

ทำหน้าที่รับภาพจากกล้องวิดีโอเข้ามา จากนั้นทำการคำนวณหาพิกัดต่างๆของหุ่นยนต์ ซึ่งค่าตัวแปรเหล่านี้จะถูกส่งให้ส่วนควบคุมต่อไป หลักการทำงานในส่วนนี้จะถูกแสดงดังรูปที่3.10 เป็นแผนผัง (Flow Chart)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ผู้สอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.10 แผนผังแสดงการทำงานประมวลผลภาพ
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาค่ามุมและระยะทางของหุ่นยนต์สามารถคำนวณได้จาก



รูปที่ 3.11 แสดงการหาค่ามุมและระยะทางของหุ่นยนต์

$$Dx = Tx - Cx$$

$$Dy = Ty - Cy$$

$$\theta_r = \tan^{-1}(Dy/Dx)$$

จากรูปที่ 3.11 ค่า θ_r เป็นมุมที่หาได้มีค่ามุมตั้งแต่ $0 - 90^\circ$ ซึ่งมีค่าเพียง Quadrant เดียวเท่านั้น จึงต้องมีการหาค่ามุมของหุ่นยนต์เพื่อให้ได้มุมครบตั้งแต่ $0 - 360^\circ$ โดยการเช็คเงื่อนไข Dx และ Dy

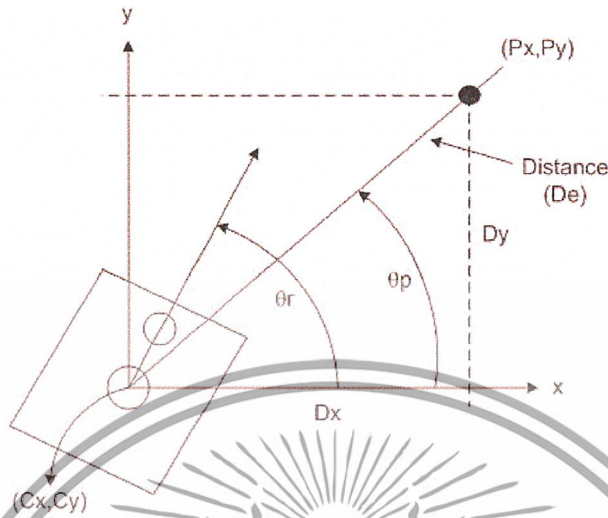
การหาค่ามุม $0 - 360^\circ$ ของหุ่นยนต์

จากสมการด้านบนนำค่ามุมที่หาได้มาทำการเช็คเงื่อนไขดังนี้

1. ถ้า $Dx \geq 0$ และ $Dy \leq 0$ ดังนั้นให้ $\theta_r = \theta_r \times (-1)$
2. ถ้า $Dx \leq 0$ และ $Dy \leq 0$ ดังนั้นให้ $\theta_r = (90^\circ - \theta_r) + 90^\circ$
3. ถ้า $Dx \leq 0$ และ $Dy \geq 0$ ดังนั้นให้ $\theta_r = (\theta_r \times (-1)) + 180^\circ$
4. ถ้า $Dx \geq 0$ และ $Dy \geq 0$ ดังนั้นให้ $\theta_r = 360^\circ - \theta_r$

จากสมการที่ผ่านมาสามารถที่จะหาค่ามุมและระยะทางระหว่างจุดที่พิกัดที่ต้องการได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงการคำนวณหาค่ามุมระยะทางของหุ่นยนต์กับจุดอ้างอิง

$$Dx = Px - Cx$$

$$Dy = Py - Cy$$

$$\theta_p = \tan^{-1}(Dy / Dx)$$

$$De = \sqrt{Dx^2 + Dy^2}$$

จากรูปที่ 3.12 ค่ามุม θ_p ที่หาได้ก็นำไปเทียบเงื่อนไขเหมือนกับการหามุมของหุ่นยนต์เพื่อให้ได้ค่ามุม 360° จากนั้นเมื่อได้ค่ามุมของหุ่นยนต์และมุมกับระยะทางของจุดที่ต้องการ จากนั้นนำค่ามุมทั้งสองมาเปรียบเทียบกันแล้วทำให้สามารถสร้างเงื่อนไขในการเคลื่อนที่ของหุ่นยนต์ได้ดังนี้

1. เมื่อค่า De มีค่าน้อยกว่าค่าระยะทางที่ยอมให้หุ่นยนต์หยุดได้ (Ds) ในที่นี้กำหนดให้ค่า Ds มีค่า 10 Pixel ถ้ายังมีค่ามากกว่าทำการเซ็คเงื่อนไขข้อ 2 และ 3 ต่อไป
2. ถ้าค่า θ_p มากกว่า θ_r
 - 2.1 เมื่อ $\theta_p - \theta_r$ แล้วน้อยกว่า 8° ให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า
 - 2.2 เมื่อ $\theta_p - \theta_r$ แล้วน้อยกว่า 180° ให้หุ่นยนต์เลี้ยวซ้าย
 - 2.3 เมื่อ $\theta_p - \theta_r$ แล้วน้อยกว่า 270° ให้หุ่นยนต์เลี้ยวขวา
 - 2.4 เมื่อ $\theta_p - \theta_r$ แล้วมากกว่า 352° ให้หุ่นยนต์เคลื่อนที่ไปข้างหน้า
3. ถ้าค่า θ_p น้อยกว่า θ_r
 - 3.1 เมื่อค่าสัมประสิทธิ์ของ $(\theta_p - \theta_r)$ มีค่าน้อยกว่า 8° ให้หุ่นยนต์เคลื่อนที่ไปข้าง

หน้าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

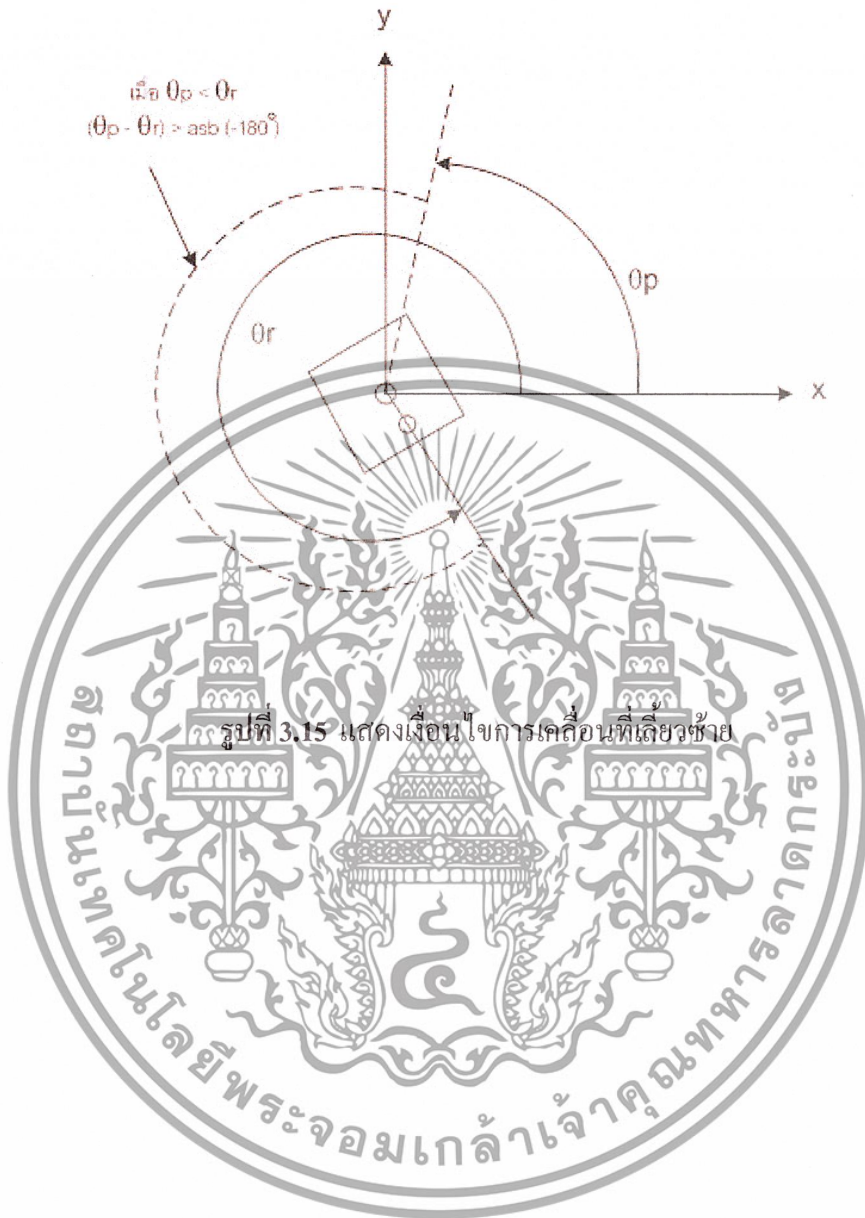
- 3.2 เมื่อค่าสัมประสิทธิ์ของ $(\theta_p - \theta_r)$ มีค่าน้อยกว่า 180° ให้หุ่นยนต์เคลื่อนที่เร็วขวา
- 3.3 เมื่อค่าสัมประสิทธิ์ของ $(\theta_p - \theta_r)$ มีค่ามากกว่า 270° ให้หุ่นยนต์เคลื่อนที่เร็วซ้าย
- 3.4 เมื่อค่าสัมประสิทธิ์ของ $(\theta_p - \theta_r)$ มีค่ามากกว่า 356° ให้หุ่นยนต์เคลื่อนที่ไป

ข้างหน้า

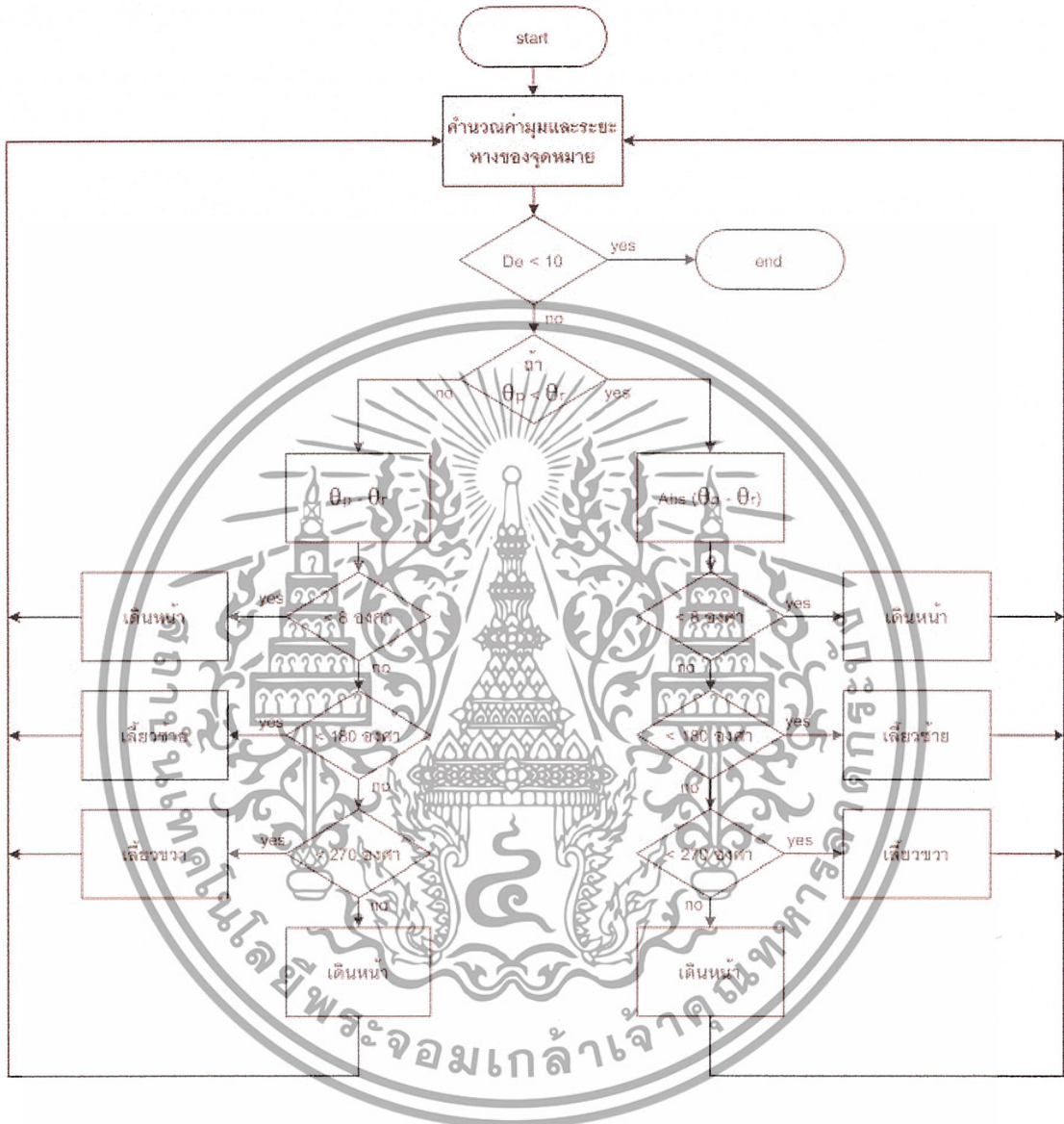
โดยเงื่อนไขทั้งหมดที่กำหนดทิศทางการเคลื่อนที่แสดงดังรูปที่ 3.13 ,3.14 และ 3.15



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงแผนผังการควบคุมการเคลื่อนที่ไปยังพิกัดอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การควบคุมแบบอิสระ

ในการควบคุมแบบอิสระ ในส่วนแรกจะใช้อัลกอริทึมการประมวลผลภาพและใช้การควบคุมการเคลื่อนที่ไปยังจุดอ้างอิงรูปแบบเดิม เพียงการเปลี่ยนแปลงค่าตัวแปร 2 ตัว คือ พิกัดของหุ่นยนต์และพิกัดของจุดอ้างอิง โดยจุดอ้างอิงนี้เปลี่ยนเป็นตำแหน่งเมาส์ในการชี้ตำแหน่งอ้างอิงให้กับหุ่นยนต์ โดยการทำงานของโปรแกรมแสดงดังรูปแผนผังที่ 3.17



รูปที่ 3.17 แสดงแผนผังการควบคุมอิสระ

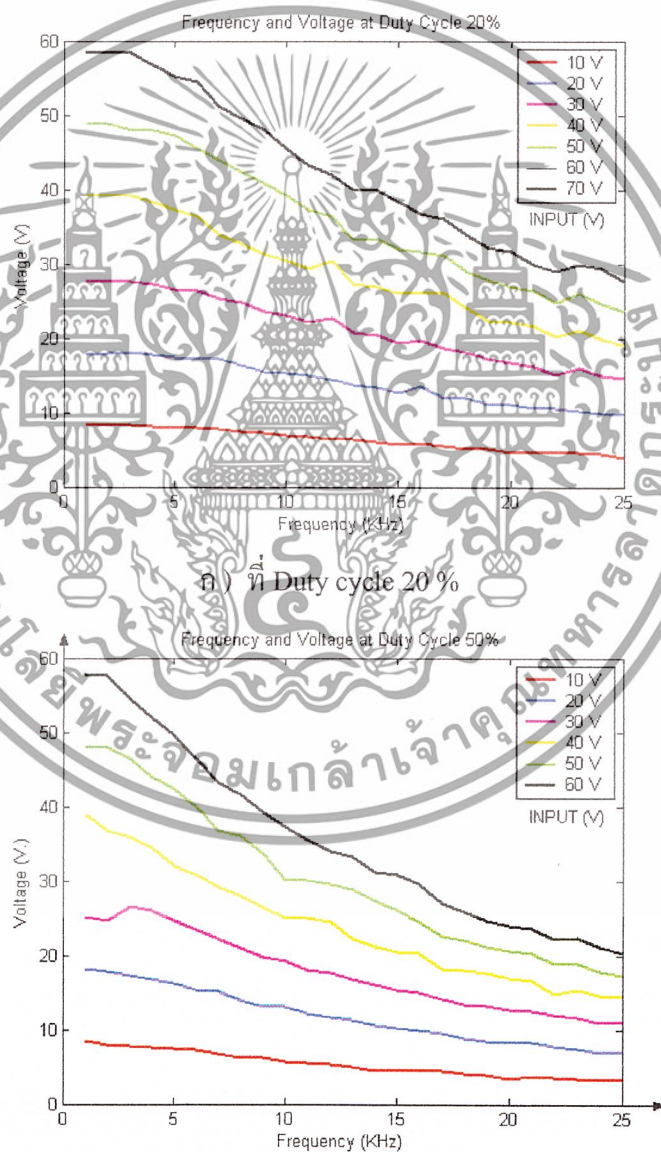
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

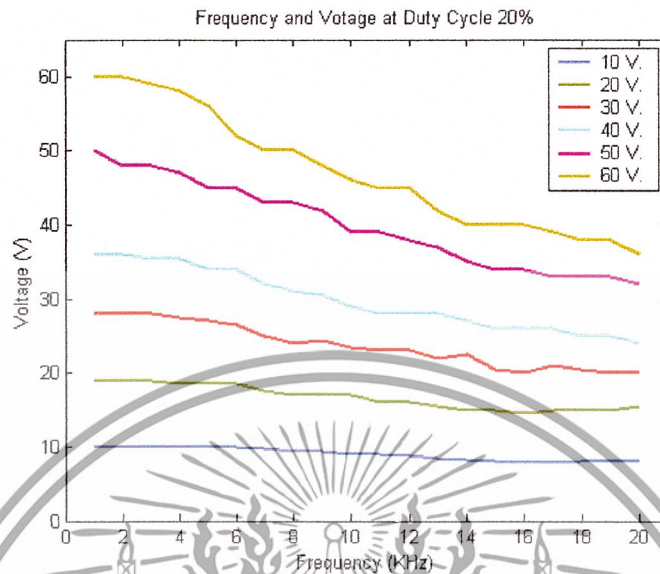
4.1 การหาค่าแรงดันที่จ่ายให้แก่ piezoelectric ceramic

หลังจากที่ได้สร้างวงจรเพื่อนำไปขับให้ หุ่นยนต์ทำงาน สามารถหาความสัมพันธ์ได้ระหว่างแรงดันที่จ่ายให้ขั้ววงจร ความถี่ที่ป้อนให้กับวงจร และค่าแรงดันที่ตกคร่อม piezoelectric ceramic โดยทำการวัดคร่อมที่ตัว piezoelectric ceramic โดยตรง แรงดันที่ตกคร่อมแสดงได้ดังรูปที่ 4.1 และ 4.2

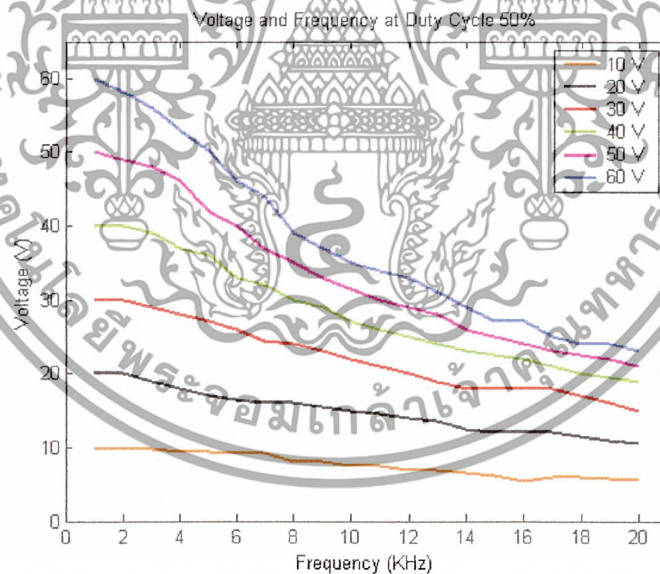


จ) ที่ Duty cycle 50 %

รูปที่ 4.1 แสดงความสัมพันธ์ของ แรงดัน ความถี่ที่ป้อนให้แก่วงจร กับแรงดันที่ตกคร่อม piezoelectric ceramic โดยได้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ของหุ่นยนต์ตัวที่ 1 ที่น้ำหนัก (87 กรัม) มาใช้ในการนำเข้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ที่ Duty cycle 20 %



ข) ที่ Duty cycle 50 %

รูปที่ 4.2 แสดงความสัมพันธ์ของ แรงดัน ความถี่ที่ป้อนให้แก่วงจร กับแรงดันที่ตกคร่อม piezoelectric ceramic โดยได้จากหุ่นยนต์ตัวที่ 2 ที่น้ำหนัก (134.5 กรัม)

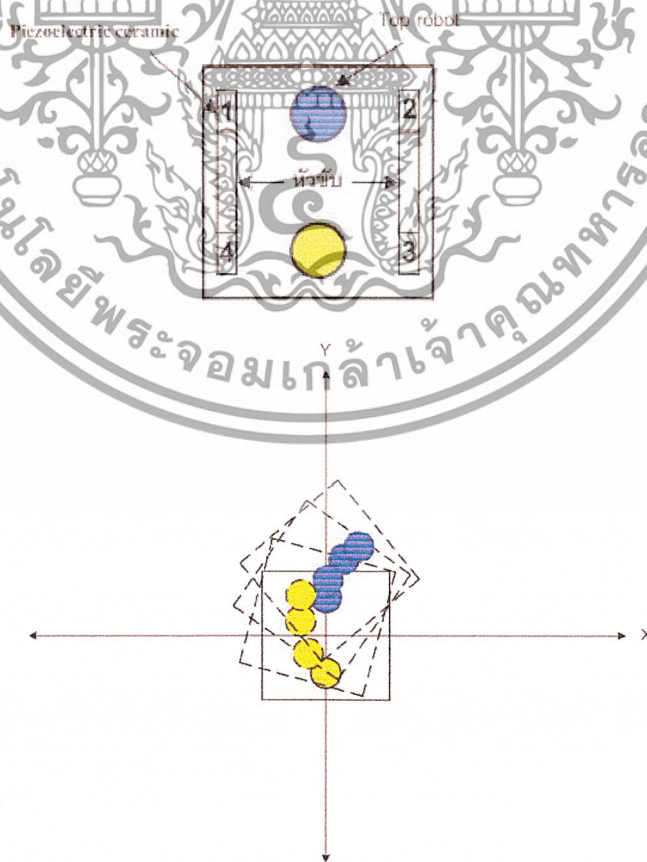
จากกราฟที่ 4.1 และ 4.2 สามารถอธิบายได้ว่า ปัจจัยที่มีผลกระทบต่อตัว piezoelectric ceramic คือ ค่าความถี่ที่เปลี่ยนแปลงไป และค่า Duty cycle โดยเมื่อความถี่ที่ป้อนให้แก่วงจรสูงขึ้นทำให้แรงดันที่ตกคร่อมตัว piezoelectric มีค่าลดลง เนื่องมาจากที่ทางด้านเอาต์พุตของ ทรานซิสเตอร์ มี ตัวต้านทาน ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ ตัว piezoelectric ceramic ต่ออยู่ ซึ่งทำหน้าที่ เป็นตัวเก็บประจุ ทำให้เหมือนมีวงจร RC ต่ออยู่ ทำหน้าที่เป็นวงจรกรองความถี่ต่ำผ่าน ทำให้ยิ่งเพิ่มค่าความถี่ ยิ่งทำให้แรงดันยิ่งตกลง ส่วนค่า Duty cycle ที่ 20% จะทำให้ ทรานซิสเตอร์มีช่วงอนน้อยกว่า ที่ Duty cycle ที่ 50% ทำให้มีแรงดันตกคร่อมที่ piezoelectric ceramic มากกว่า

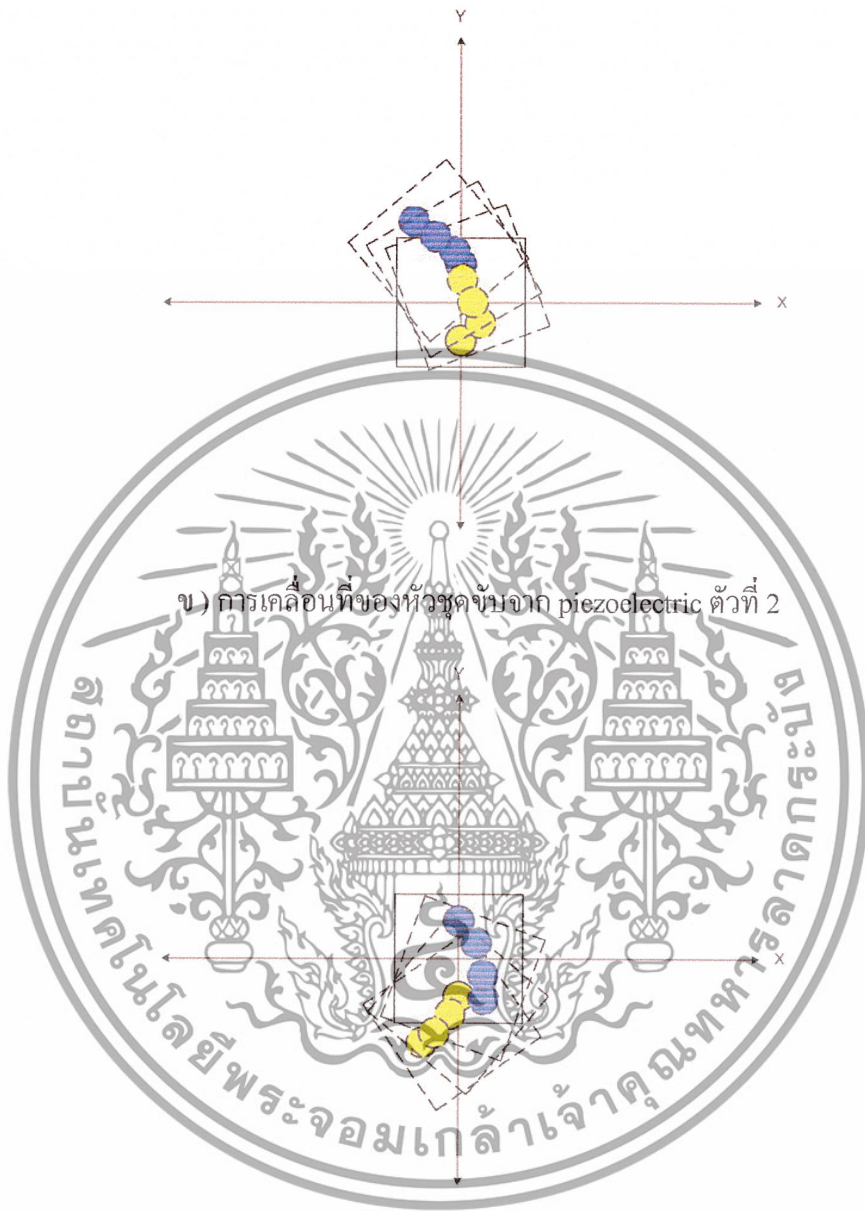
จะเห็นได้ว่าที่แรงดันประมาณ 40 V และที่ Duty cycle 20 % จะเกิดความสูญเสียน้อย อีกทั้ง ที่แรงดันจุดนี้ หุ่นยนต์สามารถเคลื่อนที่ได้ด้วยความเร็วที่ค่าหนึ่ง ในทุกช่วงความถี่ จึงเลือกใช้แรงดันที่ค่านี้จ่ายให้แก่หุ่นยนต์

4.2 การวิเคราะห์หาความเร็วเชิงเส้นในแต่ละช่วงความถี่

เมื่อได้แรงดันคงที่ที่จ่ายให้แก่หุ่นยนต์แล้ว ก็ตามรหหาความเร็วในการเคลื่อนที่ของหุ่นยนต์ได้ โดยการเก็บค่าความเร็วในการเคลื่อนที่ของหัวชุดขับแต่ละชุด โดยแต่ละชุดจะมีคุณสมบัติที่แตกต่างกันไปตามคุณสมบัติของ piezoelectric ceramic แต่ละตัว ที่ส่งแรงมากระทำต่อหัวชุดขับนั้นๆ โดยทิศทางการเคลื่อนที่ของหัวขับแต่ละชุดแสดงดังรูปที่ 4.3



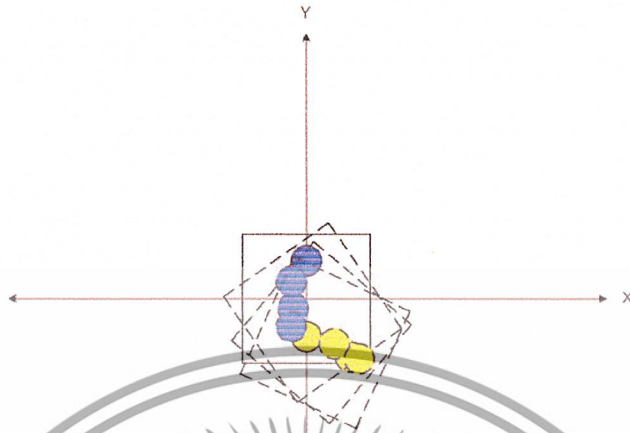
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ข) การเคลื่อนที่ของหัวชุดขับจาก piezoelectric ตัวที่ 2

ค) การเคลื่อนที่ของหัวชุดขับจาก piezoelectric ตัวที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ง) การเคลื่อนที่ของหัวขูดจับจาก piezoelectric ตัวที่ 4

รูปที่ 4.3 แสดงทิศทางการเคลื่อนที่ของหัวขูดจับแต่ละชุด

เมื่อทำการเปลี่ยนความถี่ในแต่ละย่านความถี่ จะทำให้หุ่นยนต์เคลื่อนที่ ที่ความเร็วแตกต่างกันไป โดยสามารถหาค่าความเร็วเชิงมุมโดยมองการเคลื่อนที่ของหุ่นยนต์เป็นส่วนหนึ่งของวงกลมแสดงดังรูปที่ 4.4 ดังนั้นจึงได้สมการดังนี้

$$\vec{\omega} = \frac{d\theta}{dt}$$

และความเร็วเชิงเส้นจากสมการ

$$\vec{v} = \vec{r} \times \vec{\omega}$$

โดย $\vec{\omega}$ คือ ความเร็วเชิงมุม มีหน่วยเป็น rad/sec

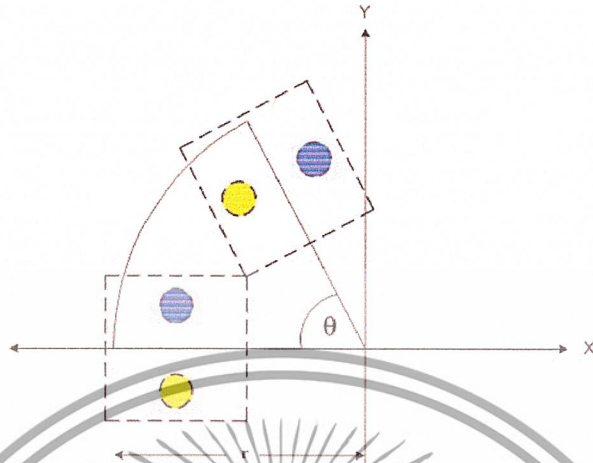
θ คือ มุมที่เกิดจากการเคลื่อนที่ของหุ่นยนต์ มีหน่วยเป็น rad

t คือ เวลาในการเคลื่อนที่ของหุ่นยนต์ มีหน่วยเป็นวินาที

\vec{v} คือ ความเร็วเชิงเส้น มีหน่วยเป็น เมตร/วินาที

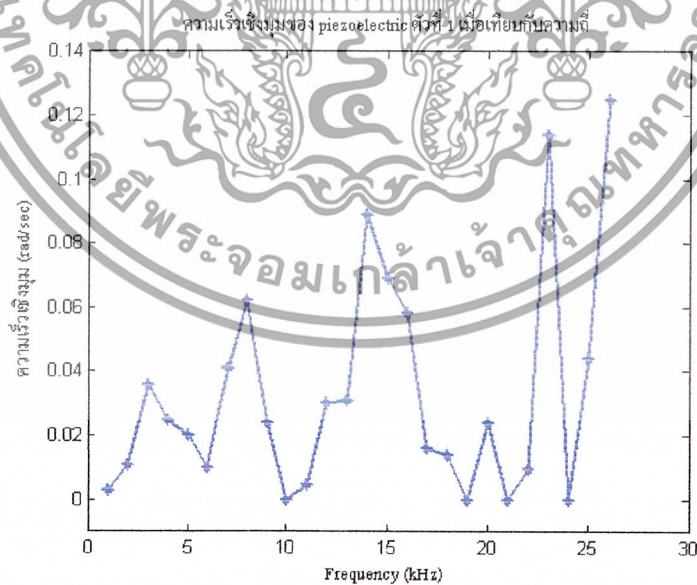
\vec{r} คือ รัศมีการเคลื่อนที่ของหัวจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



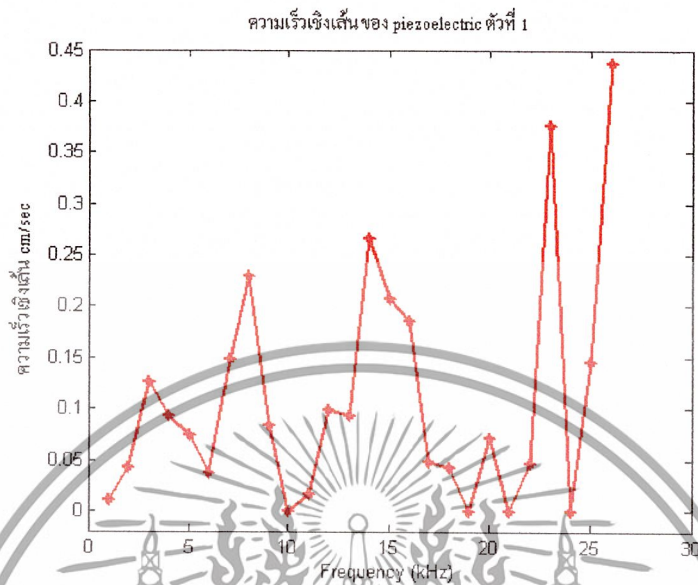
รูปที่ 4.4 แสดงการคำนวณความเร็วของหุ่นยนต์

จากการทดลองเปลี่ยนความถี่เพื่อหาความเร็วของหุ่นยนต์ จะ ได้ความเร็วเชิงมุมและความเร็วเชิงเส้นของหัวชุดขับเคลื่อนแต่ละชุดแตกต่างกันไป แสดงดังรูปที่ 4.5

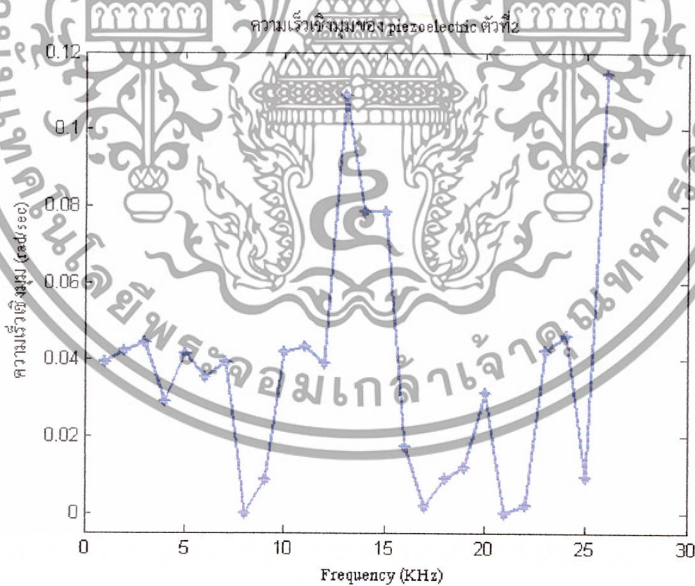


ก) ความเร็วเชิงมุมของ piezoelectric ceramic ตัวที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

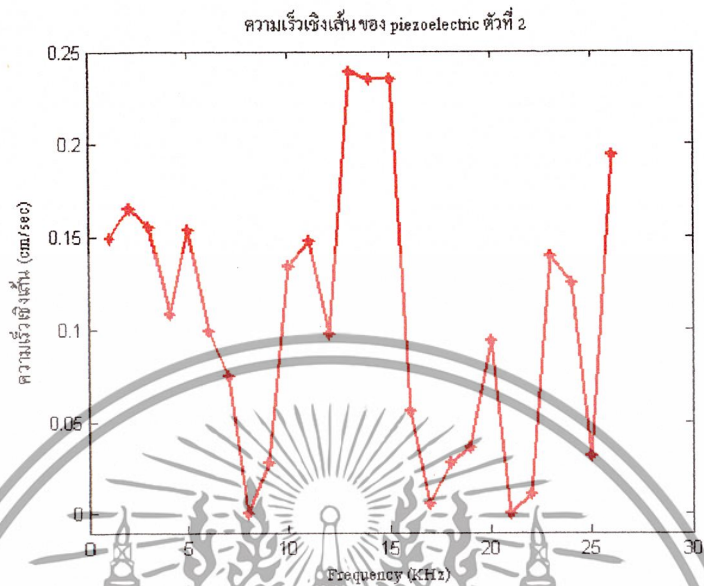


ข) ความเร็วเชิงเส้นของ piezoelectric ceramic ตัวที่ 1

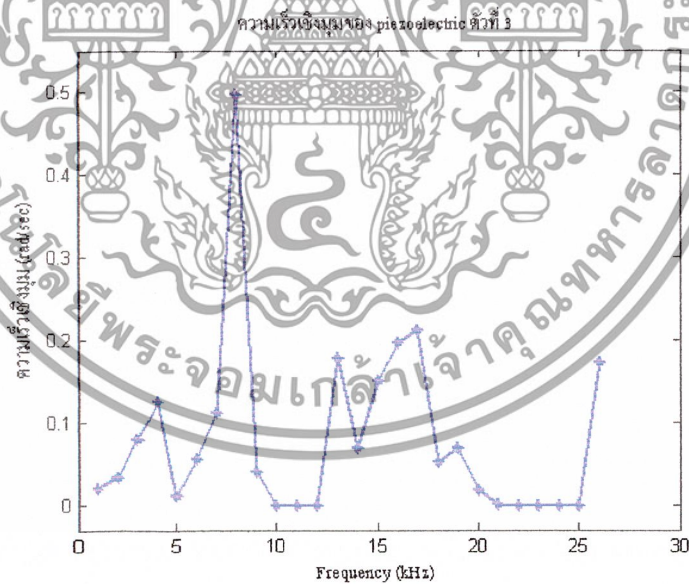


ค) ความเร็วเชิงมุมของ piezoelectric ceramic ตัวที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

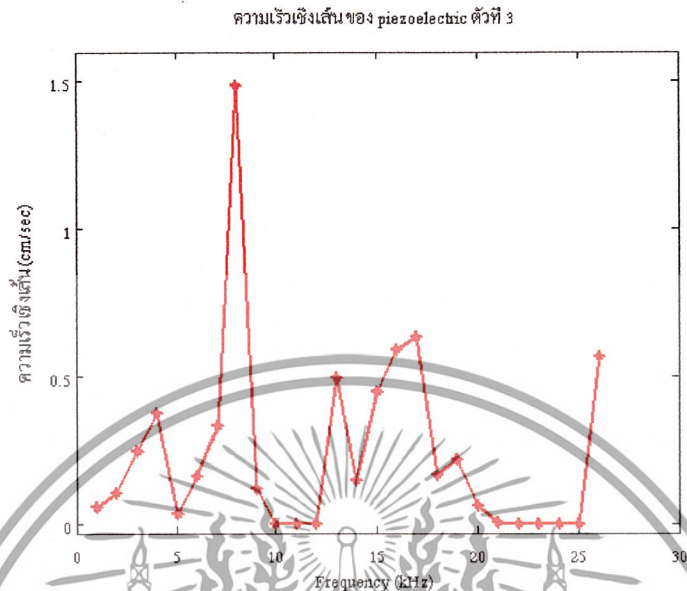


ก) ความเร็วเชิงเส้นของ piezoelectric ceramic ตัวที่ 2

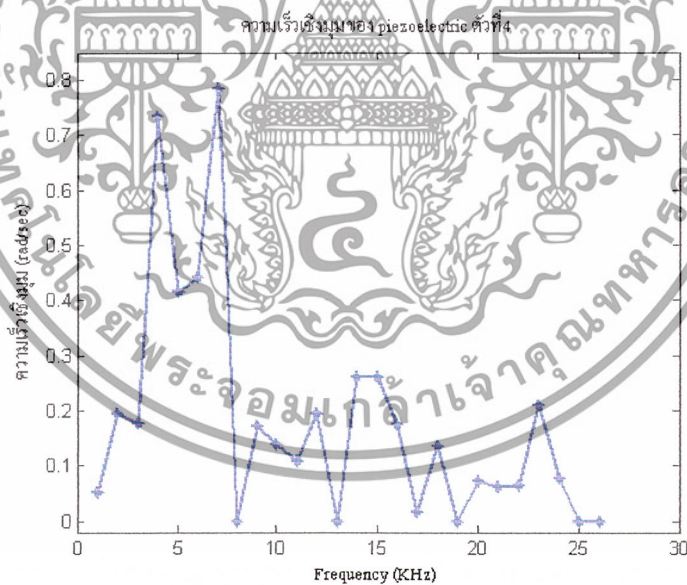


ข) ความเร็วเชิงมุมของ piezoelectric ceramic ตัวที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

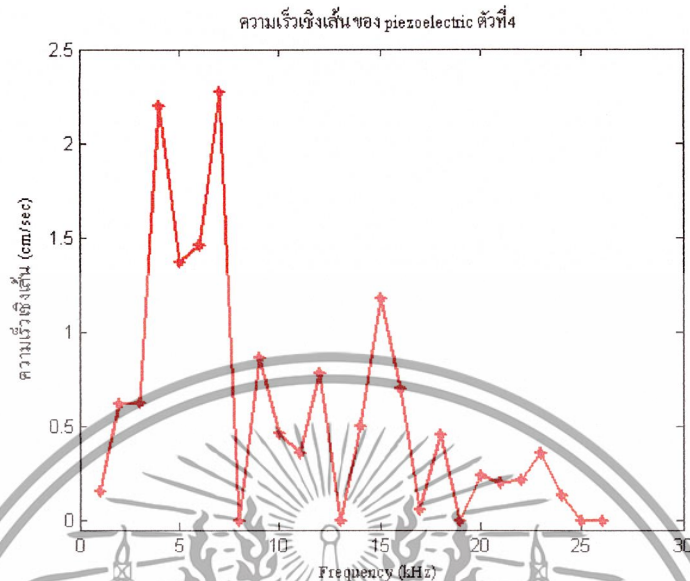


ด) ความเร็วเชิงเส้นของ piezoelectric ceramic ตัวที่ 3



ข) ความเร็วเชิงมุมของ piezoelectric ceramic ตัวที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ข) ความเร็วเชิงเส้นของ piezoelectric ตัวที่ 4

รูปที่ 4.5 แสดงกราฟความเร็วเชิงมุมและความเร็วเชิงเส้น เทียบกับความถี่ ของ piezoelectric ceramic แต่ละตัว

จากกราฟที่ 4.5 สามารถทำให้ทราบถึงความเร็วเชิงเส้นที่ความถี่ต่างๆของ หัวชุดจับแต่ละชุด ทำให้สามารถนำมาวิเคราะห์หาจุดที่ความเร็วของหัวชุดจับ ทั้ง 4 ตัว มีความเร็วที่ใกล้เคียงกันเพื่อให้หุ่นยนต์เคลื่อนที่ได้ตามต้องการที่จุดต่างๆ อย่างแม่นยำ โดยเมื่อดูจากกราฟ จุดที่มีความเร็วเชิงเส้นที่มีความเร็วที่ใกล้เคียงกันของ piezoelectric ceramic ทั้ง 4 ตัว คือจุดที่มีความเร็วเชิงเส้นเท่ากับ 0.1 cm/sec

โดยที่ piezoelectric ceramic ตัวที่ 1 มีความถี่ที่ 12 kHz

piezoelectric ceramic ตัวที่ 2 มีความถี่ที่ 6 kHz

piezoelectric ceramic ตัวที่ 3 มีความถี่ที่ 5 kHz

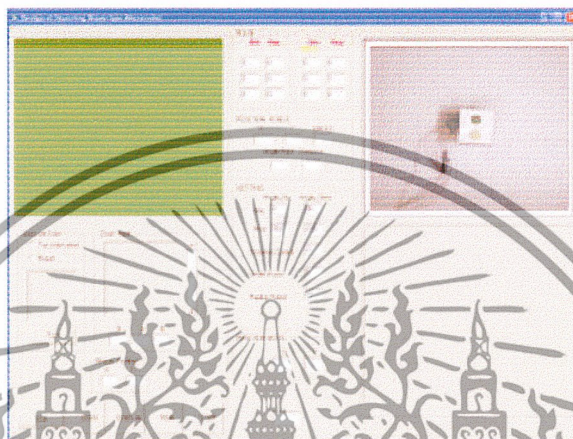
piezoelectric ceramic ตัวที่ 4 มีความถี่ที่ 17 kHz

เมื่อ ได้ความถี่ที่ทำให้ piezoelectric ceramic มีความเร็วเชิงเส้นที่เท่ากันแล้ว หลังจากนั้นก็จะทำการทดลองควบคุม ทิศทางการเคลื่อนที่ของหุ่นยนต์โดยใช้โปรแกรมที่ได้จัดเตรียมไว้ผ่านทางกล้องวิดีโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การรับสัญญาณภาพจากกล้องวิดีโอ

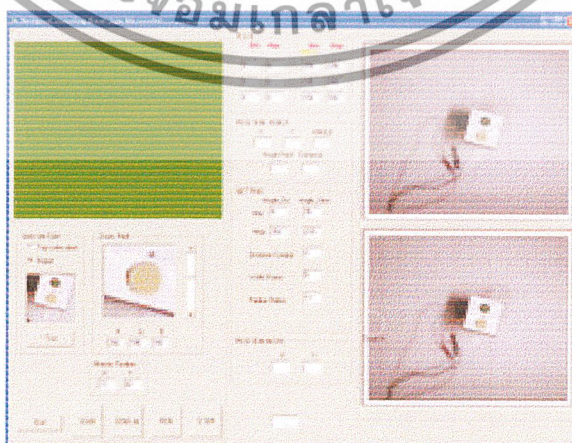
การดึงสัญญาณภาพจากกล้องวิดีโอ เข้ามาประมวลผลใน โปรแกรม VISUAL BASIC ทำได้ โดยการต่อกล้องวิดีโอเข้าทางพอร์ต USB ของคอมพิวเตอร์ แล้วทำการถ่ายภาพหุ่นยนต์จากด้านบนเข้ามาประมวลผลในโปรแกรมที่ได้เขียนไว้โดยตรง แสดงดังรูปที่ 4.6



รูปที่ 4.6 แสดงการดึงภาพเข้ามาแสดงผล ภาพที่ดึงเข้ามาจะปรากฏที่หน้าต่างที่หน้าต่างด้านบนขวาของหน้าต่าง โปรแกรม

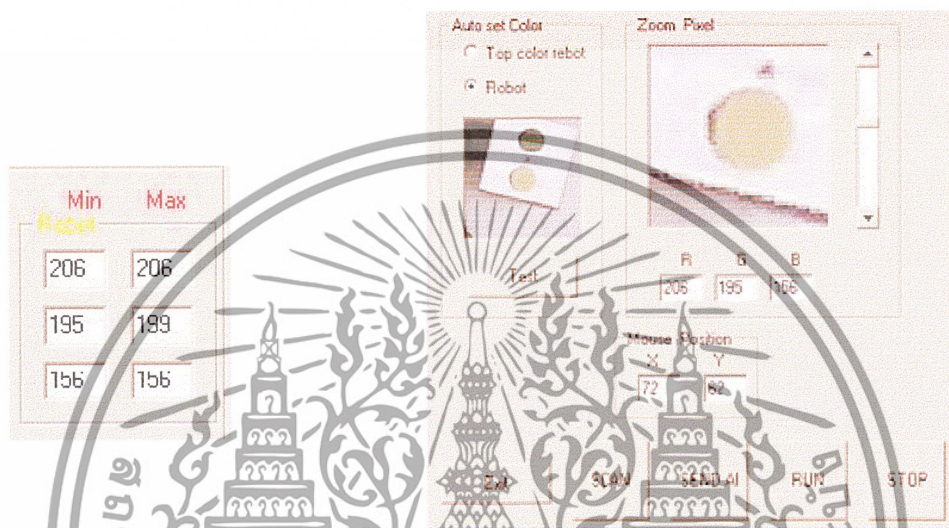
4.4 การปรับแต่ง (calibration) แมตริกซ์ RGB บนตัวของหุ่นยนต์

ที่บนตัวของหุ่นยนต์ ได้กำหนดจุดสีไว้ 2 สีเพื่อเป็นตัวบ่งบอกว่าส่วนใดเป็น Top Robot และ ส่วนใดเป็น Center Robot ซึ่งจะนำจุดทั้ง 2 นี้มาทำการปรับแต่งสี โดยทำการคัดลอกภาพจากกล้องวิดีโอมาประมวลผลอีกหน้าต่างหนึ่งจากนั้นทำจากคลิกเม้าส์เพื่อทำการขยายเฉพาะบริเวณ จากนั้นทำการปรับสเกลบาร์เพื่อขยายภาพเข้าออกเพื่อดูภาพเฉพาะจุด แสดงดังรูปที่ 4.7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรศึกษาใช้เพื่อการศึกษานี้เท่านั้น เมื่อผู้ดูแลเห็นว่าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

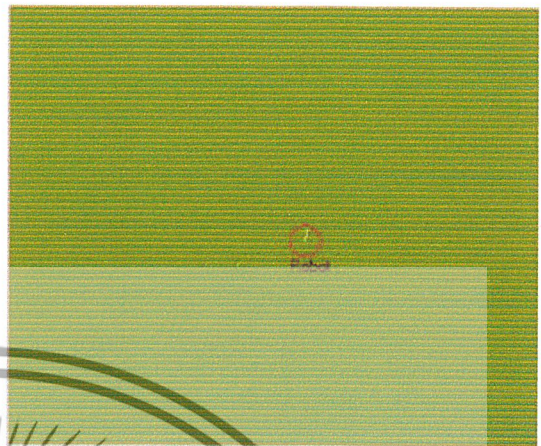
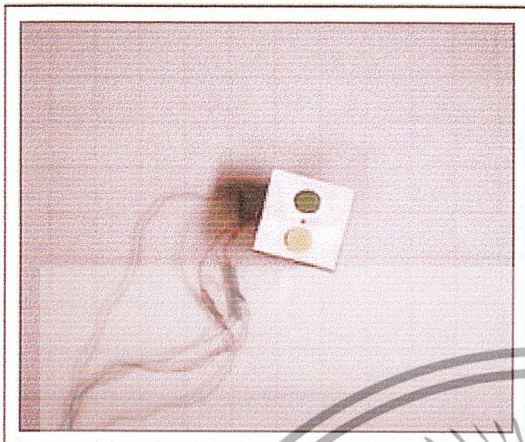
เมื่อทำการซูมภาพเข้ามาดูแล้ว หลังจากนั้นต้องทำการดับเบิลคลิกที่หน้าต่างรูปภาพที่ซูมมา ค้างไว้แล้วลากสีตามจุดที่ต้องการ ในส่วนของโปรแกรมจะทำการปรับจูนค่าแม่สี RGB โดยอัตโนมัติ โดยจะมีค่าสูงสุดและต่ำสุดของแม่สี RGB โดยสามารถดูได้จากหน้าต่างที่บอกค่าที่ถูกปรับแต่งแม่สี RGB แสดงดังรูปที่ 4.8



รูปที่ 4.8 ก. หน้าของแม่สี RGB ที่ทำการปรับแต่งได้
ข. การซูมภาพมาปรับแต่งแม่สี RGB

4.5 การประมวลผลภาพของหุ่นยนต์

หลังจากการปรับแต่งค่าสีของหุ่นยนต์แล้ว จะทำการสแกนภาพอีกครั้งเพื่อที่ส่งไปยังหน้าต่างจำลองตำแหน่งจริงของหุ่นยนต์ แสดงดังรูปที่ 4.9 ข โดยตัวหุ่นยนต์จะเป็นรูปวงกลมสีขาว โดยจะมีเส้นขีดจากจุดกลางไปยังจุดปลาย เพื่อเป็นการบอกรวมของหุ่นยนต์ และพิกัดที่หน้าต่างนี้จะมีขนาดเท่ากับพิกัดของหน้าต่างที่รับภาพเข้ามาประมวลผล และหน้าต่างนี้จะใช้เป็นหน้าต่างที่ใช้ในการควบคุมการเคลื่อนที่ของหุ่นยนต์



ก.

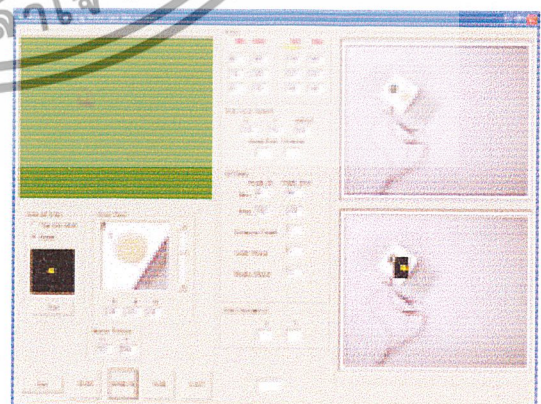
ข.

รูปที่ 4.9 ก. รูปที่รับเข้ามาจากกล้องวิดีโอ

ข. รูปจำลองตำแหน่งพิกัดที่ได้จากการประมวลผลจากคอมพิวเตอร์

4.6 การควบคุมหุ่นยนต์

ในการกำหนดทิศทางการเคลื่อนที่ของหุ่นยนต์ สามารถทำได้โดยการคลิกเมาส์ที่หุ่นยนต์จากหน้าต่างจำลองที่จริง ดังรูปที่ 4.10 ก จากนั้นทำการลากเมาส์ไปยังตำแหน่งที่ต้องการ ดังรูปที่ 4.10 ข จากนั้นหุ่นยนต์จะเคลื่อนที่ไปยังตำแหน่งที่ต้องการ โดยในการเคลื่อนที่ของหุ่นยนต์ จะทำงานในลักษณะแบบ on-off โดยโปรแกรมจะทำงานวนลูปไปเรื่อยๆ จนกว่าพิกัดของหุ่นยนต์ที่ รับค่าเข้ามาได้ จะมีค่าเท่ากับค่าพิกัดที่กำหนด



รูปที่ 4.10 ก.

รูปที่ 4.10 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากผลการทดลองที่กล่าวมาในข้างต้น แสดงให้เห็นว่าหุ่นยนต์สามารถเคลื่อนที่ได้ในแนวราบ ตามที่ออกแบบไว้ โดยทำงานในลักษณะ on-off control โดยในการควบคุมความเร็วของหุ่นยนต์ สามารถทำได้โดยการปรับแรงดันและความถี่ในหัวชุดขับแต่ละชุดให้เหมาะสม

ในส่วนของการควบคุมโดยผ่านทางกล้องวีดีโอ สามารถกำหนดตำแหน่งให้หุ่นยนต์เคลื่อนที่ ไปยังแต่ละตำแหน่งที่ต้องการ ได้อย่างถูกต้องอยู่ในเกณฑ์ที่น่าพอใจ คือมีค่ามุมผิดพลาดที่ 5 องศา และ ระยะทางที่ 5-10 พิกเซล

5.2 ปัญหาที่พบในโครงงาน

เนื่องจากความเร็วในการทำงานของหุ่นยนต์ขนาดเล็ก ที่ใช้ piezoelectric ceramic มีความเร็วในการเคลื่อนที่แต่ละตัวไม่เท่ากัน จึงทำให้ความเร็วในแต่ละข้างของหุ่นยนต์เวลาเคลื่อนที่เกิดความไม่สมดุล เนื่องจากปัจจัยหลายอย่าง เช่น การประกอบ piezoelectric ceramic เข้ากับหัวชุดขับ ไม่ได้มุมตามที่ ออกแบบไว้ หัวชุดขับมีขนาดเล็กเมื่อทำการกดชิ้นงานไม่สามารถกดชิ้นงานให้ได้ขนาดและมุมตามที่ ต้องการ และน้ำหนักของตัวหุ่นยนต์มีผลต่อการเคลื่อนที่

ส่วนที่ต้องเป็นส่วนการควบคุม มีปัญหาในเรื่องแสงรบกวนจากภายนอกที่เข้ามาทำให้ โปรแกรมไม่สามารถ แสคนหาจุดที่อยู่บนหุ่นยนต์ได้ และในเรื่องของความเร็วของ โปรแกรม ยังไม่เร็ว พอที่จะทำงานในบางความถี่ซึ่งหุ่นยนต์เดินได้เร็ว

5.3 ผลที่ได้รับจากโครงงาน

ในการทำโครงงานนี้ต้องอาศัยความรู้ในหลาย ๆ ด้าน ทั้งความรู้ด้านการเขียน โปรแกรม ความรู้ด้านการออกแบบ สร้างทางแมคคานิกส์ ความรู้เกี่ยวกับการประมวลผลภาพดิจิทัล และความรู้ด้านอิเล็กทรอนิกส์ ซึ่งทุกอย่างเป็นส่วนสำคัญในการสร้างทักษะให้กับผู้ทำโครงงาน และยังได้รับประโยชน์จากการวิเคราะห์ การตัดสินใจในการประสานงานในการทำงานมีการแบ่งงานกันทำ ทำให้มีการฝึกการวางแผนอย่างมีระบบและคิดแก้ปัญหาต่างๆที่เกิดขึ้นร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางในการพัฒนาและการประยุกต์

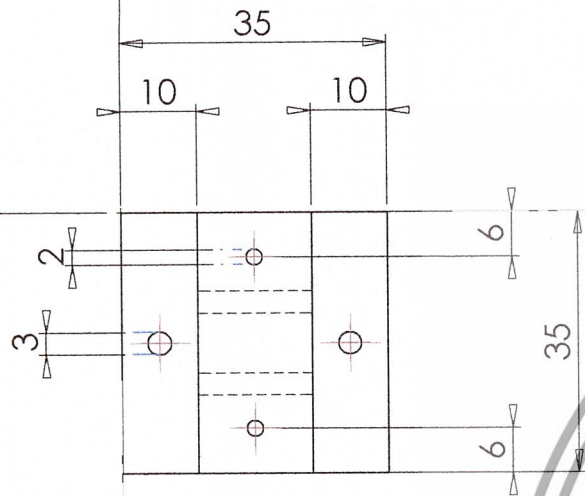
แนวทางในการปรับปรุงพัฒนา สามารถทำได้โดยใส่ตัวควบคุมเข้าไปควบคุมแทนการควบคุมแบบ on-off เช่นการใช้ PID Controller, Adaptive Control เพื่อให้หุ่นยนต์สามารถเคลื่อนที่ได้อย่างแม่นยำมากขึ้นและมีความเร็วมากขึ้น และโครงการนี้ยังสามารถประยุกต์ใช้ได้ในงานที่ต้องการความแม่นยำสูง และการทำงานที่ไม่ต้องการ การรบกวนจากสนามแม่เหล็ก



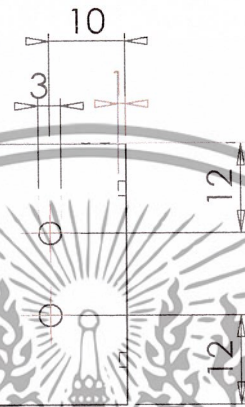
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



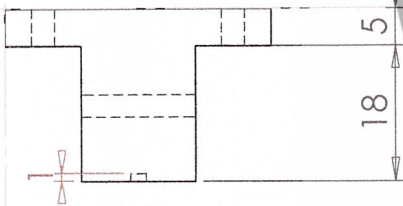
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



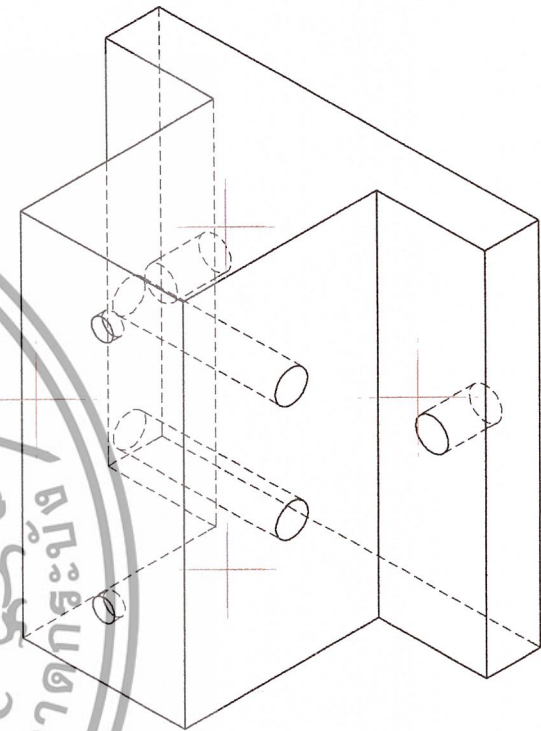
FRONT VIEW
SCALE 1:1



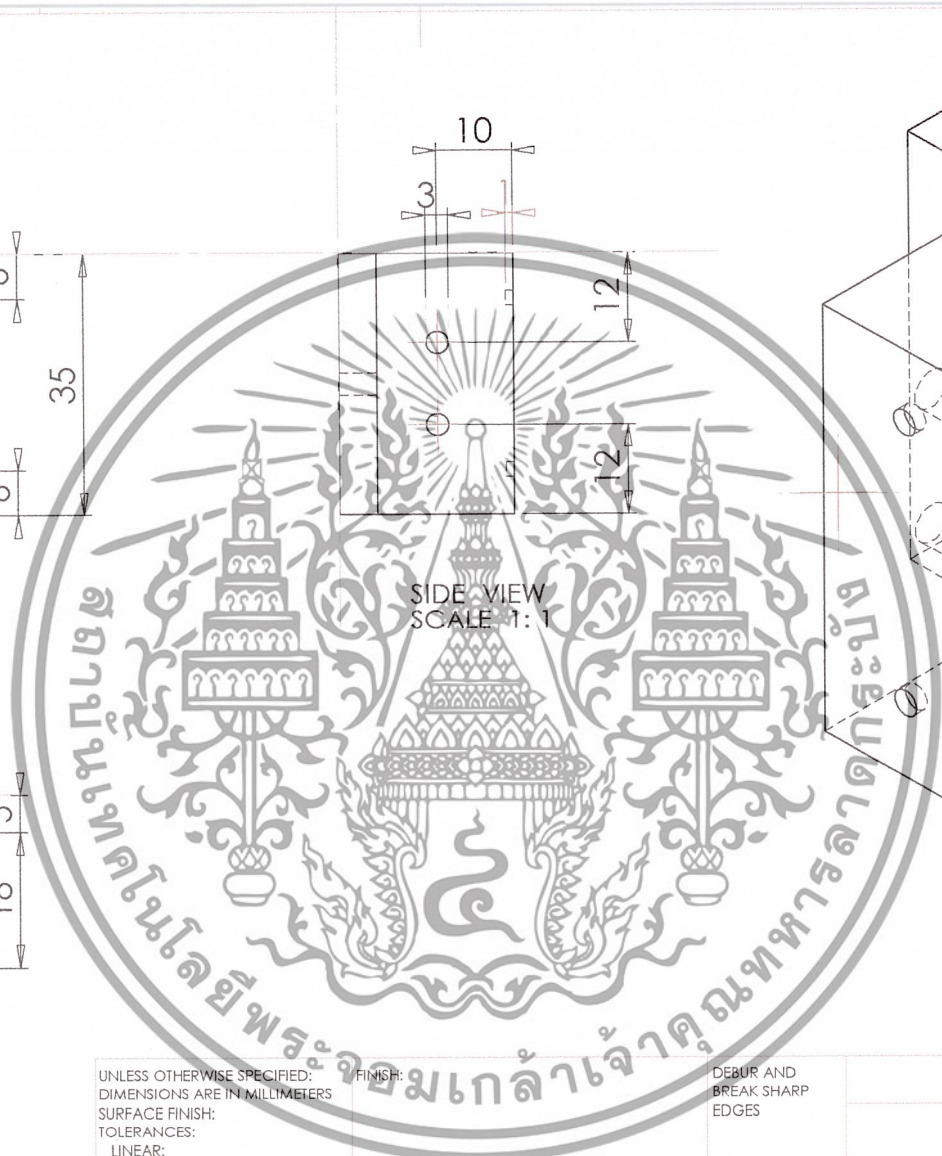
SIDE VIEW
SCALE 1:1



TOP VIEW
SCALE 1:1



ISOMETRIC VIEW
SCALE 2:1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE	
DRAWN				
CHK'D				
APP'VD				
MFG				
Q.A				
				MATERIAL:
				WEIGHT:

TITLE:

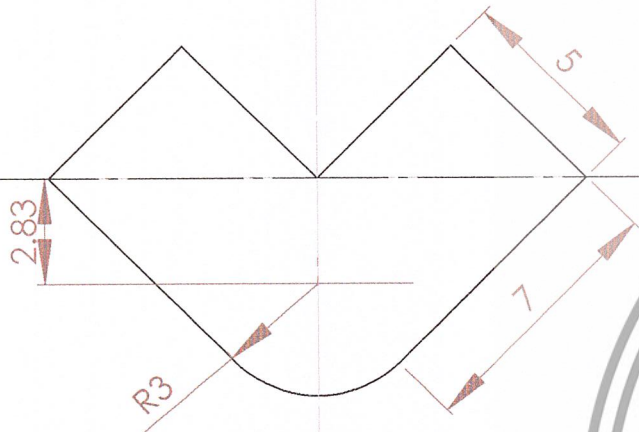
DWG NO.

base

A4

SCALE:1:1

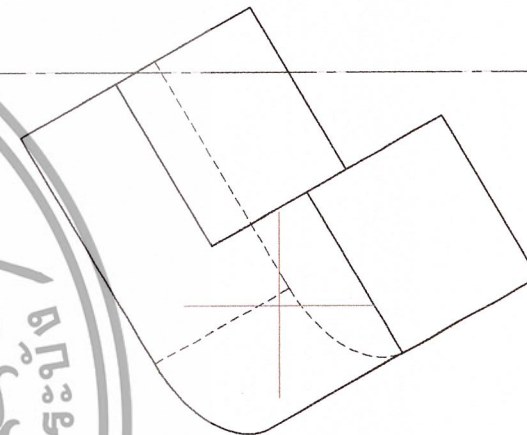
SHEET 1 OF 1



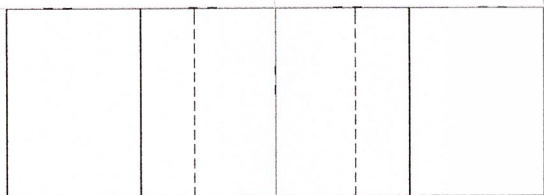
FRONT VIEW
SCALE 5 : 1



SIDE VIEW
SCALE 5 : 1



ISOMETRIC VIEW
SCALE 5 : 1



TOP VIEW
SCALE 5 : 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

	NAME	SIGNATURE	DATE	
DRAWN				
CHK'D				
APPVD				
MFG				
Q.A				

TITLE:	
DWG NO.	

MATERIAL:

WEIGHT:

SCALE:1:1

SHEET 1 OF 1

drivehead

A4

ภาคผนวก

โปรแกรมประมวลผลภาพ

Dim num As Integer

Dim Datax As String

Dim Datay As String

Dim Dataangle As String

Dim x_click As Integer

Dim y_click As Integer

Dim R_min As Long

Dim R_max As Long

Dim G_min As Long

Dim G_max As Long

Dim B_min As Long

Dim B_max As Long

Dim R_min_t As Long

Dim R_max_t As Long

Dim G_min_t As Long

Dim G_max_t As Long

Dim B_min_t As Long

Dim B_max_t As Long

Dim R_min_R1 As Long

Dim R_max_R1 As Long

Dim G_min_R1 As Long

Dim G_max_R1 As Long

Dim B_min_R1 As Long

Dim B_max_R1 As Long

Dim Robot_cenx(1 To 6) As Long

Dim Robot_ceny(1 To 6) As Long

Dim Count_cen(1 To 6) As Long

Dim R_ROBOT As Integer

Dim Robot_Topx(1 To 6) As Long

Dim Robot_Topy(1 To 6) As Long

Dim Count_top(1 To 6) As Long

เอกสารนี้เป็นทรัพย์สินของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim angle_R(1 To 6) As Integer

Dim Dx(1 To 6) As Integer

Dim Dy(1 To 6) As Integer

Dim Px(1 To 6) As Integer

Dim Py(1 To 6) As Integer

Dim Tx(1 To 6) As Integer

Dim Ty(1 To 6) As Integer

Dim O_R(1 To 6) As Integer

Const pi As Double = 3.141578

Const dOneDegree As Double = pi / 180

Dim Angle As Integer

Dim W(1 To 7) As Integer

Dim h(1 To 7) As Integer

Dim rad As Integer

Dim RC_X(1 To 7) As Integer

Dim RC_Y(1 To 7) As Integer

Dim Ro_mx(1 To 6) As Long

Dim Ro_my(1 To 6) As Long

Dim Robot_CR1_i As Long

Dim Robot_CR1_j As Long

Dim count_R1 As Long

Dim Robot_mx(1 To 6) As Long

Dim Robot_my(1 To 6) As Long

Dim t(1 To 4) As Integer

Dim rad_R As Integer

/// *parameter calculator move* ///

Dim Pmx(1 To 4) As Integer

Dim Pmy(1 To 4) As Integer

Dim dis_er(1 To 4) As Integer

Dim Dmx(1 To 4) As Double

Dim Dmy(1 To 4) As Double

Dim Tmx(1 To 4) As Integer

Dim Tmy(1 To 4) As Integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim Om_R(1 To 4) As Integer
```

```
Dim anglem_R(1 To 4) As Integer
```

```
Dim M_an1 As Integer
```

```
Dim M_an2 As Integer
```

```
Dim an_sl1 As Integer
```

```
Dim an_sl2 As Integer
```

```
Dim distance As Integer
```

```
Dim distance_ob As Integer
```

```
Dim W_R As Integer
```

```
Dim D1_2 As Integer
```

```
Dim D1_3 As Integer
```

```
Dim D2_3 As Integer
```

```
Private Sub cmdrun_Click() '-----คลิกปุ่ม run-----'
```

```
Call setting
```

```
Do
```

```
DoEvents
```

```
quit = 0
```

```
Call program1
```

```
Loop Until quit = 1
```

```
End Sub
```

```
Sub setting() '-----กำหนด พิกัดที่จุดปลายทาง-----'
```

```
DoEvents
```

```
Robot_mx(1) = Robot_CR1_i
```

```
Robot_my(1) = Robot_CR1_j
```

```
Text6.Text = Robot_mx(1)
```

```
Text7.Text = Robot_my(1)
```

```
End Sub
```

```
Private Sub cmdscan_Click() '-----ทำการสแกนหาพิกัดหุ่นยนต์ครั้งแรก-----'
```

```
DoEvents
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim i As Integer, j As Integer
Dim Color1 As Long, r As Integer, G As Integer, b As Integer
Robot_CR1_i = 0
Robot_CR1_j = 0
count_R1 = 0
R_min_R1 = RM11.Text
R_max_R1 = RMA1.Text
G_min_R1 = GM11.Text
G_max_R1 = GMA1.Text
B_min_R1 = BM11.Text
B_max_R1 = BMA1.Text
BitBlt Picture2.hdc, 0, 0, 373, 309, Picture1.hdc, 0, 0, vbSrcCopy
For i = 0 To 373
For j = 0 To 309
Color1 = GetPixel(Picture2.hdc, i, j)
Call ColorRGB(Color1, r, G, b)
If (r + G + b) <= (R_max_R1 + G_max_R1 + B_max_R1) And r >= R_min_R1 And r <= R_max_R1 And G
>= G_min_R1 And G <= G_max_R1 And b >= B_min_R1 And b <= B_max_R1 Then
Color1 = 589823
Robot_CR1_i = Robot_CR1_i + i
Robot_CR1_j = Robot_CR1_j + j
count_R1 = count_R1 + 1
Else
Color1 = 0
End If
SetPixel Picture2.hdc, i, j, Color1
DoEvents
Next j
Next i
Robot_CR1_i = Robot_CR1_i / count_R1
Robot_CR1_j = Robot_CR1_j / count_R1
Text7.Text = Robot_CR1_i
Text8.Text = Robot_CR1_j

```

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdsendai_Click()/'-----ทำการสแกนหาพิกัดหุ่นยนต์ครั้งที่สอง-----/'
DoEvents
quit = 0
'Do
Do
DoEvents
Call scancen
Call cul_angle
Call send_AI
Robot_CR1_i = Robot_cenx(1)
Robot_CR1_j = Robot_ceny(1)
Call plot
'Loop Until quit = 1
num = num + 1
Loop Until num = 100
End Sub
Private Sub plot()
Dim fsox
Dim fsoy
Dim fsoz

'Check File
Set fsox = CreateObject("Scripting.filesystemobject")
If fsox.fileExists(Datax) Then GoTo sub1

sub1: Set fsoy = CreateObject("Scripting.filesystemobject")
If fsoy.fileExists(Datay) Then GoTo sub2

sub2: Set fsoz = CreateObject("Scripting.filesystemobject")
If fsoz.fileExists(Dataangle) Then
GoTo sub3

```

Open Datax For Output As #1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Print #1, Text7.Text '---->Edit Parameter plot X

Close #1 'Close the file

Open Datay For Output As #2

Print #2, Text8.Text '---->Edit Parameter plot Y

Close #2 'Close the file

Open Dataangle For Output As #3

Print #3, Text9.Text '---->Edit Parameter plot Z

Close #3 'Close the file

End If

sub3:

Open Datax For Append As #1

Print #1, Text7.Text '---->Edit Parameter plot X

Close #1 'Close the file

Open Datay For Append As #2

Print #2, Text8.Text '---->Edit Parameter plot Y

Close #2 'Close the file

Open Dataangle For Append As #3

Print #3, Text9.Text '---->Edit Parameter plot Z

Close #3 'Close the file

End Sub

Private Sub cmdtest_Click() '/-----ทดสอบสี่ของหุ่นยนต์-----/'

DoEvents

Dim i As Integer, j As Integer

Dim Color1 As Long, r As Integer, G As Integer, b As Integer

If Option1.Value = True Then

R_min = RMIT.Text

R_max = RMAT.Text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

G_min = GMIT.Text
G_max = GMAT.Text
B_min = BMIT.Text
B_max = BMAT.Text
ElseIf Option2.Value = True Then
R_min = RMI1.Text
R_max = RMA1.Text
G_min = GMII.Text
G_max = GMA1.Text
B_min = BMI1.Text
B_max = BMA1.Text
End If
StretchBlt Picture4.hdc, 0, 0, 85, 85, Picture1.hdc, (x_click - (85 / 2)), (y_click - (85 / 2)), 85, 85, vbSrcCopy
For i = 0 To 85
For j = 0 To 85
Color1 = GetPixel(Picture4.hdc, i, j)
Call ColorRGB(Color1, r, G, b)
If (r + G + b) <= (R_max + G_max + B_max) And r >= R_min And r <= R_max And G >= G_min And G <=
G_max And b >= B_min And b <= B_max Then

If Option1.Value = True Then Color1 = 16777215
If Option2.Value = True Then Color1 = 589823

Else
Color1 = 0
End If
SetPixel Picture4.hdc, i, j, Color1
DoEvents
Next j
Next i
End Sub

Private Sub Command1_Click() '/-----click ปุ่ม stop-----/'
quit = 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

```
Private Sub Command2_Click() /-----clickปุ่ม exit-----/'
```

```
Timer1.Enabled = False
```

```
Unload Form1
```

```
quit = 1
```

```
Out &H378, &HFF
```

End Sub

```
Private Sub Form_Load()
```

```
ezVidCap1.Preview = True
```

```
ezVidCap1.AutoSize = False
```

```
ezVidCap1.FrameEventEnabled = False
```

```
ezVidCap1.Width = 373
```

```
ezVidCap1.Height = 309
```

```
Datax = App.Path & "robotx.txt"
```

```
Datay = App.Path & "roboty.txt"
```

```
Dataangle = App.Path & "robotangle.txt"
```

End Sub

```
Private Sub Picture2_Click() /-----ส่งรูปหุ่นยนต์เพื่อหาคำสี่-----/'
```

```
DoEvents
```

```
x_click = Text1.Text
```

```
y_click = Text2.Text
```

```
StretchBlt Picture4.hdc, 0, 0, Picture4.ScaleWidth, Picture4.ScaleHeight, Picture1.hdc, (x_click -  
((Picture4.ScaleWidth) / 2)), (y_click - ((Picture4.ScaleHeight) / 2)), Picture4.ScaleWidth, Picture4.ScaleHeight,  
vbSrcCopy
```

```
StretchBlt Picture3.hdc, 0, 0, Picture3.ScaleWidth, Picture3.ScaleHeight, Picture1.hdc, (Text1.Text -  
(VScroll1.Value / 2) - 2), (Text2.Text - (VScroll1.Value / 2) + 2), VScroll1.Value, VScroll1.Value, vbSrcCopy
```

End Sub

```
Private Sub Picture2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
DoEvents
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim p_rgb_x As Long, p_rgb_y As Long
Dim Color1 As Long, r As Integer, G As Integer, b As Integer
Dim i As Integer, j As Integer
BitBlt Picture2.hdc, 0, 0, Picture2.ScaleWidth, Picture2.ScaleHeight, Picture1.hdc, 0, 0, vbSrcCopy
Text1.Text = X
Text2.Text = Y
Picture2.Circle (X, Y), VScroll1 / 2, RGB(255, 255, 255)
p_rgb_x = Text1.Text
p_rgb_y = Text2.Text
Color1 = GetPixel(Picture2.hdc, p_rgb_x, p_rgb_y)
Call ColorRGB(Color1, r, G, b)
Text3.Text = r
Text4.Text = G
Text5.Text = b
Picture2.CurrentX = X - VScroll1 / 2
Picture2.CurrentY = Y + VScroll1 / 2
Picture2.Print " Zoom in "
End Sub

Private Function ColorRGB(Color1 As Long, r As Integer, G As Integer, b As Integer) '---ทำค่าสี RGB---'
r = Color1 Mod 256
b = Color1 \ 65536
G = (Color1 \ 256) Mod 256
End Function

```

```
Private Sub Picture3_DblClick() '----- กำหนดจุดสีของตัวหุ่นยนต์-----'
```

```
If Option1.Value = True Then
```

```
    R_min_t = Text3.Text
```

```
    RMIT.Text = R_min_t
```

```
    R_max_t = Text3.Text
```

```
    RMAT.Text = R_max_t
```

```
    G_min_t = Text4.Text
```

```
    GMIT.Text = G_min_t
```

```
    G_max_t = Text4.Text
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GMAT.Text = G_max_t
B_min_t = Text5.Text
BMIT.Text = B_min_t
B_max_t = Text5.Text
BMAT.Text = B_max_t
ElseIf Option2.Value = True Then
    R_min_R1 = Text3.Text
    RMI1.Text = R_min_R1
    R_max_R1 = Text3.Text
    RMA1.Text = R_max_R1
    G_min_R1 = Text4.Text
    GMI1.Text = G_min_R1
    G_max_R1 = Text4.Text
    GMA1.Text = G_max_R1
    B_min_R1 = Text5.Text
    BMI1.Text = B_min_R1
    B_max_R1 = Text5.Text
    BMA1.Text = B_max_R1
End If
End Sub

```

```

Private Sub Picture3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Dim p_rgb_x As Long, p_rgb_y As Long
Dim Color1 As Long, r As Integer, G As Integer, b As Integer
Dim i As Integer, j As Integer
Text1.Text = X
Text2.Text = Y
p_rgb_x = Text1.Text
p_rgb_y = Text2.Text
Color1 = GetPixel(Picture3.hdc, p_rgb_x, p_rgb_y)
Call ColorRGB(Color1, r, G, b)
Text3.Text = r
Text4.Text = G

```

```
Text5.Text = b
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

If Button = 1 Then

'///// Condition /////'

If Option1.Value = True Then

If Text3.Text < R_min_t Then

R_min_t = Text3.Text

RMIT.Text = R_min_t

ElseIf Text3.Text > R_max_t Then

R_max_t = Text3.Text

RMAT.Text = R_max_t

ElseIf Text4.Text < G_min_t Then

G_min_t = Text4.Text

GMIT.Text = G_min_t

ElseIf Text4.Text > G_max_t Then

G_max_t = Text4.Text

GMAT.Text = G_max_t

ElseIf Text5.Text < B_min_t Then

B_min_t = Text5.Text

BMIT.Text = B_min_t

ElseIf Text5.Text > B_max_t Then

B_max_t = Text5.Text

BMAT.Text = B_max_t

End If

ElseIf Option2.Value = True Then

If Text3.Text < R_min_R1 Then

R_min_R1 = Text3.Text

RMI1.Text = R_min_R1

ElseIf Text3.Text > R_max_R1 Then

R_max_R1 = Text3.Text

RMA1.Text = R_max_R1

ElseIf Text4.Text < G_min_R1 Then

G_min_R1 = Text4.Text

GMI1.Text = G_min_R1

ElseIf Text4.Text > G_max_R1 Then

G_max_R1 = Text4.Text

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GMA1.Text = G_max_R1
    ElseIf Text5.Text < B_min_R1 Then
        B_min_R1 = Text5.Text
        BMI1.Text = B_min_R1
    ElseIf Text5.Text > B_max_R1 Then
        B_max_R1 = Text5.Text
        BMA1.Text = B_max_R1
    End If
End If
End Sub

Private Sub Picture4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Text1.Text = X
    Text2.Text = Y
End Sub

Private Sub Picture5_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    DoEvents
    If X <= (Ro_mx(1) + rad_R) And X >= (Ro_mx(1) - rad_R) And Y <= (Ro_my(1) + rad_R) And Y >= (Ro_my(1) - rad_R) Then
        Picture5.Circle (X, Y), rad_R, RGB(255, 0, 0)
        Robot_mx(1) = X
        Robot_my(1) = Y
        Text14.Text = Robot_mx(1)
        Text15.Text = Robot_my(1)

        Picture5.Line (Ro_mx(1), Ro_my(1))-(X, Y), RGB(255, 0, 0)
        Picture5.CurrentX = X
        Picture5.CurrentY = Y + rad_R
        Picture5.Print "Move Robot"

    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub Picture5_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

DoEvents

quit = 0

Text1.Text = X

Text2.Text = Y

Ro_mx(1) = Text7.Text

Ro_my(1) = Text8.Text

If Button = 1 Then

If X <= (Ro_mx(1) + rad_R) And X >= (Ro_mx(1) - rad_R) And Y <= (Ro_my(1) + rad_R) And Y >= (Ro_my(1) - rad_R) Then

Picture5.Circle (X, Y), rad_R, RGB(255, 0, 0)

Robot_mx(1) = X

Robot_my(1) = Y

Text14.Text = Robot_mx(1)

Text15.Text = Robot_my(1)

Picture5.Line (Ro_mx(1), Ro_my(1)-(X, Y), RGB(255, 0, 0))

Picture5.CurrentX = X

Picture5.CurrentY = Y + rad_R

Picture1.Print "Move Robot"

End If

End If

End Sub

Private Sub Timer1_Timer() '-----กำหนด timer-----/'

Dim fsox

Dim fsoy

Dim fsoz

'Check File

Set fsox = CreateObject("Scripting.filesystemobject")

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

If fsox.fileExists(Datax) Then GoTo sub1

sub1: Set fsoy = CreateObject("Scripting.filesystemobject")

If fsoy.fileExists(Datay) Then GoTo sub2

sub2: Set fsoz = CreateObject("Scripting.filesystemobject")

If fsoz.fileExists(Dataangle) Then

GoTo sub3

Open Datax For Output As #1

Print #1, Text7.Text '---->Edit Parameter plot X

Close #1 'Close the file

Open Datay For Output As #2

Print #2, Text8.Text '---->Edit Parameter plot Y

Close #2 'Close the file

Open Dataangle For Output As #3

Print #3, Text9.Text '---->Edit Parameter plot Z

Close #3 'Close the file

End If

sub3:

Open Datax For Append As #1

Print #1, Text7.Text '---->Edit Parameter plot X

Close #1 'Close the file

Open Datay For Append As #2

Print #2, Text8.Text '---->Edit Parameter plot Y

Close #2 'Close the file

Open Dataangle For Append As #3

Print #3, Text9.Text '---->Edit Parameter plot Z

Close #3 'Close the file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub VScroll1_Change() ‘/-----ปรับการซูม-----/’

StretchBlt Picture3.hdc, 0, 0, Picture3.ScaleWidth, Picture3.ScaleHeight, Picture1.hdc, (x_click - (VScroll1.Value / 2) - 2), (y_click - (VScroll1.Value / 2) + 2), VScroll1.Value, VScroll1.Value, vbSrcCopy

End Sub

Sub scancen() ‘/-----ทำการเคลื่อนหุ่นยนต์อีกรอบ-----/’

Dim i As Integer, j As Integer

Dim Color1 As Long, r As Integer, G As Integer, b As Integer

Dim i_1 As Long, i_2 As Long, j_1 As Long, j_2 As Long

Robot_cenx(1) = 0

Robot_ceny(1) = 0

Count_cen(1) = 0

R_min_t = RMIT.Text

R_max_t = RMAT.Text

G_min_t = GMIT.Text

G_max_t = GMAT.Text

B_min_t = BMIT.Text

B_max_t = BMAT.Text

Robot_Topx(1) = 0

Robot_Topy(1) = 0

Count_top(1) = 0

R_ROBOT = Text6.Text & ""width robot""

Do

 DoEvents

 BitBlt Picture2.hdc, 0, 0, 373, 309, Picture1.hdc, 0, 0, vbSrcCopy

 R_ROBOT = R_ROBOT + 5

 i_1 = Robot_CR1_i - R_ROBOT

 i_2 = Robot_CR1_i + R_ROBOT

 j_1 = Robot_CR1_j - R_ROBOT

 j_2 = Robot_CR1_j + R_ROBOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i = i_1 To i_2
For j = j_1 To j_2

Color1 = GetPixel(Picture2.hdc, i, j)
Call ColorRGB(Color1, r, G, b)

If (r + G + b) <= (R_max_R1 + G_max_R1 + B_max_R1) And r >= R_min_R1 And r <= R_max_R1 And G
>= G_min_R1 And G <= G_max_R1 And b >= B_min_R1 And b <= B_max_R1 Then
    Color1 = 589823
    Robot_cenx(1) = Robot_cenx(1) + i
    Robot_ceny(1) = Robot_ceny(1) + j
    Count_cen(1) = Count_cen(1) + 1
ElseIf (r + G + b) <= (R_max_t + G_max_t + B_max_t) And r >= R_min_t And r <= R_max_t And G >=
G_min_t And G <= G_max_t And b >= B_min_t And b <= B_max_t Then
    Color1 = 16777215
    Robot_Topx(1) = Robot_Topx(1) + i
    Robot_Topy(1) = Robot_Topy(1) + j
    Count_top(1) = Count_top(1) + 1
Else
    Color1 = 0
End If
SetPixel Picture2.hdc, i, j, Color1
DoEvents
Next j
Next i

Loop While Count_cen(1) = 0 Or Count_top(1) = 0

```

'-----cul position -----'

```
Robot_cenx(1) = Robot_cenx(1) / Count_cen(1)
```

```
Robot_ceny(1) = Robot_ceny(1) / Count_cen(1)
```

```
Text7.Text = Robot_cenx(1)
```

```
Text8.Text = Robot_ceny(1)
```

```
Robot_Topx(1) = Robot_Topx(1) / Count_top(1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Robot_Topy(1) = Robot_Topy(1) / Count_top(1)
End Sub

Sub cul_angle() ' /-----คำนวณมุมของหุ่นยนต์-----/'
DoEvents
Tx(1) = Robot_Topx(1)
Ty(1) = Robot_Topy(1)
Px(1) = Robot_CR1_i
Py(1) = Robot_CR1_j
Dx(1) = Tx(1) - Px(1)
Dy(1) = Ty(1) - Py(1)
If Dx(1) = 0 Then
    Dx(1) = 1
End If
angle_R(1) = Int(((Atn(Dy(1)/Dx(1))) * 180) / pi)
If Dx(1) >= 0 And Dy(1) <= 0 Then
    O_R(1) = angle_R(1) * (-1)
ElseIf Dx(1) <= 0 And Dy(1) <= 0 Then
    O_R(1) = (90 - angle_R(1)) + 90
ElseIf Dx(1) <= 0 And Dy(1) >= 0 Then
    O_R(1) = (angle_R(1) * (-1)) + 180
ElseIf Dx(1) >= 0 And Dy(1) >= 0 Then
    O_R(1) = 360 - angle_R(1)
End If
Text9.Text = O_R(1)

End Sub

```

```

Sub send_AI() ' /-----แสดงมุมของหุ่นยนต์-----/'

```

```
Picture5.Cls
```

```
DoEvents
```

```
rad_R = Text12.Text "radius robot"
```

```
RC_X(1) = Robot_cenx(1)
```

```
RC_Y(1) = Robot_ceny(1)
```

```
Angle = O_R(1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

W(1) = Int(Cos(Angle * dOneDegree) * rad_R)
    If Sgn(W(1)) = -1 Then
        W(1) = W(1) * (-1)
    End If
h(1) = Int(Sin(Angle * dOneDegree) * rad_R)
    If Sgn(h(1)) = -1 Then
        h(1) = h(1) * (-1)
    End If
Picture5.ScaleMode = 3
Picture5.Circle (RC_X(1), RC_Y(1), rad_R, RGB(255, 0, 0))
    If Angle >= 0 And Angle <= 90 Then
        Picture5.Line (RC_X(1), RC_Y(1)-(RC_X(1)+W(1)), (RC_Y(1)-h(1))), vbWhite
    ElseIf Angle >= 91 And Angle <= 180 Then
        Picture5.Line (RC_X(1), RC_Y(1)-(RC_X(1)-W(1)), (RC_Y(1)-h(1))), vbWhite
    ElseIf Angle >= 181 And Angle <= 270 Then
        Picture5.Line (RC_X(1), RC_Y(1)-(RC_X(1)-W(1)), (RC_Y(1)+h(1))), vbWhite
    ElseIf Angle >= 271 And Angle <= 360 Then
        Picture5.Line (RC_X(1), RC_Y(1)-(RC_X(1)+W(1)), (RC_Y(1)+h(1))), vbWhite
    End If
    Picture5.CurrentX = RC_X(1) + rad_R
    Picture5.CurrentY = RC_Y(1) + rad_R
    Picture5.Print "Robot"
End Sub

```

```
Sub program1()
```

```
Call scancen1
```

```
Call cul_angle1
```

```
Call send_AI1
```

```
Robot_CR1_i = Robot_cenx(1)
```

```
Robot_CR1_j = Robot_ceny(1)
```

```
Call Error_point1
```

```
Call Controll
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Sub scancen1()
Dim i As Integer, j As Integer
Dim Color1 As Long, r As Integer, G As Integer, b As Integer
Dim i_1 As Long, i_2 As Long, j_1 As Long, j_2 As Long
Robot_cenx(1) = 0
Robot_ceny(1) = 0
Count_cen(1) = 0
Robot_Topx(1) = 0
Robot_Topy(1) = 0
Count_top(1) = 0
R_min_t = RMIT.Text
R_max_t = RMAT.Text
G_min_t = GMIT.Text
G_max_t = GMAT.Text
B_min_t = BMIT.Text
B_max_t = BMAT.Text
R_ROBOT = Text16.Text

Do
DoEvents
BitBlt Picture2.hdc, 0, 0, 373, 309, Picture1.hdc, 0, 0, vbSrcCopy
R_ROBOT = R_ROBOT + 5
i_1 = Robot_CR1_i - R_ROBOT
i_2 = Robot_CR1_i + R_ROBOT
j_1 = Robot_CR1_j - R_ROBOT
j_2 = Robot_CR1_j + R_ROBOT

For i = i_1 To i_2
For j = j_1 To j_2
Color1 = GetPixel(Picture2.hdc, i, j)
Call ColorRGB(Color1, r, G, b)
If (r + G + b) <= (R_max_R1 + G_max_R1 + B_max_R1) And r >= R_min_R1 And r <= R_max_R1 And G >=
G_min_R1 And G <= G_max_R1 And b >= B_min_R1 And b <= B_max_R1 Then

```

Color1 = 589823

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Robot_cenx(1) = Robot_cenx(1) + i
Robot_ceny(1) = Robot_ceny(1) + j
Count_cen(1) = Count_cen(1) + 1
ElseIf (r + G + b) <= (R_max_t + G_max_t + B_max_t) And r >= R_min_t And r <= R_max_t And G >= G_min_t
And G <= G_max_t And b >= B_min_t And b <= B_max_t Then
    Color1 = 16777215
    Robot_Topx(1) = Robot_Topx(1) + i
    Robot_Topy(1) = Robot_Topy(1) + j
    Count_top(1) = Count_top(1) + 1
Else
    Color1 = 0
End If
SetPixel Picture2.hdc, i, j, Color1
DoEvents
Next j
Next i
Loop While Count_cen(1) = 0 Or Count_top(1) = 0

Robot_cenx(1) = Robot_cenx(1) / Count_cen(1)
Robot_ceny(1) = Robot_ceny(1) / Count_cen(1)
Text7.Text = Robot_cenx(1)
Text8.Text = Robot_ceny(1)
Robot_Topx(1) = Robot_Topx(1) / Count_top(1)
Robot_Topy(1) = Robot_Topy(1) / Count_top(1)

End Sub

```

Sub cul_angle1()

DoEvents

Tx(1) = Robot_Topx(1)

Ty(1) = Robot_Topy(1)

Px(1) = Robot_CR1_i

Py(1) = Robot_CR1_j

Dx(1) = Tx(1) - Px(1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dy(1) = Ty(1) - Py(1)
If Dx(1) = 0 Then
    Dx(1) = 1
angle_R(1) = Int(((Atn(Dy(1) / Dx(1))) * 180) / pi)
If Dx(1) >= 0 And Dy(1) <= 0 Then
    O_R(1) = angle_R(1) * (-1)
ElseIf Dx(1) <= 0 And Dy(1) <= 0 Then
    O_R(1) = (90 - angle_R(1)) + 90
ElseIf Dx(1) <= 0 And Dy(1) >= 0 Then
    O_R(1) = (angle_R(1) * (-1)) + 180
ElseIf Dx(1) >= 0 And Dy(1) >= 0 Then
    O_R(1) = 360 - angle_R(1)
End If
Text9.Text = O_R(1)
End If
End Sub

Sub send_AI()
    '-----clear display AI-----'
    Picture5.Cls
    '-----'

    DoEvents
    rad_R = Text12.Text
    RC_X(1) = Robot_cenx(1)
    RC_Y(1) = Robot_ceny(1)
    Angle = O_R(1)
    W(1) = Int(Cos(Angle * dOneDegree) * rad_R)
    If Sgn(W(1)) = -1 Then
        W(1) = W(1) * (-1)
    End If
    h(1) = Int(Sin(Angle * dOneDegree) * rad_R)
    If Sgn(h(1)) = -1 Then
        h(1) = h(1) * (-1)
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Picture5.ScaleMode = 3
Picture5.Circle (RC_X(1), RC_Y(1)), rad_R, RGB(255, 0, 0)
If Angle >= 0 And Angle <= 90 Then
    Picture5.Line (RC_X(1), RC_Y(1))-((RC_X(1) + W(1)), (RC_Y(1) - h(1))), RGB(0, 0, 0)
ElseIf Angle >= 91 And Angle <= 180 Then
    Picture5.Line (RC_X(1), RC_Y(1))-((RC_X(1) - W(1)), (RC_Y(1) - h(1))), RGB(0, 0, 0)
ElseIf Angle >= 181 And Angle <= 270 Then
    Picture5.Line (RC_X(1), RC_Y(1))-((RC_X(1) - W(1)), (RC_Y(1) + h(1))), RGB(0, 0, 0)
ElseIf Angle >= 271 And Angle <= 360 Then
    Picture5.Line (RC_X(1), RC_Y(1))-((RC_X(1) + W(1)), (RC_Y(1) + h(1))), RGB(0, 0, 0)
End If
Picture5.CurrentX = RC_X(1) - rad_R
Picture5.CurrentY = RC_Y(1) + rad_R
Picture5.Print "Robot"
End Sub

Sub Error_point1()
Pmx(1) = Text14.Text
Pmy(1) = Text15.Text
Tmx(1) = Text7.Text
Tmy(1) = Text8.Text

Dmx(1) = Pmx(1) - Tmx(1)
Dmy(1) = Pmy(1) - Tmy(1)

If Dmx(1) = 0 Then
    Dmx(1) = 1
End If

anglem_R(1) = Int(((Atn(Dmy(1) / Dmx(1))) * 180) / pi)
If Dmx(1) >= 0 And Dmy(1) <= 0 Then
    Om_R(1) = anglem_R(1) * (-1)
ElseIf Dmx(1) <= 0 And Dmy(1) <= 0 Then
    Om_R(1) = (90 - anglem_R(1)) + 90
ElseIf Dmx(1) <= 0 And Dmy(1) >= 0 Then
    Om_R(1) = (anglem_R(1) * (-1)) + 180

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ElseIf Dmx(1) >= 0 And Dmy(1) >= 0 Then
    Om_R(1) = 360 - anglem_R(1)
End If
Text16.Text = Om_R(1)
dis_er(1) = Int(Sqr((Dmx(1) * Dmx(1)) + (Dmy(1) * Dmy(1))))
Text17.Text = dis_er(1)

End Sub

Sub Control10 ('/-----ส่งค่าที่ประมวลผลออกไปควบคุม-----/')
    M_an1 = Text18.Text
    M_an2 = Text19.Text
    an_sl1 = Text20.Text
    an_sl2 = Text21.Text
    distance = Text22.Text
    If dis_er(1) < distance Then
        Out &H378, &H9F
    ElseIf Om_R(1) > O_R(1) Then
        If (Om_R(1) - O_R(1)) < M_an1 Then
            //driver_speed GO
            Out &H378, &H9E
        ElseIf (Om_R(1) - O_R(1)) < an_sl1 Then
            //driver_slow left
            Out &H378, &H9B
        ElseIf (Om_R(1) - O_R(1)) <= 180 Then
            //driver_speed left
            Out &H378, &H8F
        ElseIf (Om_R(1) - O_R(1)) > M_an2 Then
            //driver_GO
            Out &H378, &H9E
        ElseIf (Om_R(1) - O_R(1)) > an_sl2 Then
            //driver_slow Right
            Out &H378, &H97
        ElseIf (Om_R(1) - O_R(1)) > 180 Then
            //driver_speed Right

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Out &H378, &H93
  End If
  ElseIf Om_R(1) < O_R(1) Then
    If (Abs(Om_R(1) - O_R(1))) < M_an1 Then
      //driver_GO
      Out &H378, &H9E
    ElseIf (Abs(Om_R(1) - O_R(1))) < an_sl1 Then
      //driver_slow Right
      Out &H378, &H97
    ElseIf (Abs(Om_R(1) - O_R(1))) <= 180 Then
      //driver_speed Right
      Out &H378, &H93
    ElseIf (Abs(Om_R(1) - O_R(1))) > M_an2 Then
      //driver_GO
      Out &H378, &H9E
    ElseIf (Abs(Om_R(1) - O_R(1))) > an_sl2 Then
      //driver_slow left
      Out &H378, &H9B
    ElseIf (Abs(Om_R(1) - O_R(1))) > 180 Then
      //driver_speed left
      Out &H378, &H8F
    End If
  End If
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการนี้ออกแบบหุ่นยนต์ขนาดเล็กชนิดเคลื่อนที่ ผ่านสำเร็จไปได้ด้วยดี ต้องขอขอบคุณ อาจารย์ภาควิชาวิศวกรรมระบบควบคุมที่สอนและให้ความรู้ อาจารย์คอน ที่ช่วยให้แนวคิดวิธีแก้ปัญหาต่างๆ และ เงินช่วยเหลือต่างๆ รวมถึงเพื่อนๆ ที่คอยช่วยเหลือและช่วยเป็นกำลังใจ และโครงการนี้จะไม่สามารสรสำเร็จลุล่วงไปได้ ถ้าปราศจาก บิดา มารดา อันเป็นที่เคารพ จากการทำงานโครงการครั้งนี้สามารถนำความรู้และประสบการณ์ที่ได้ไปประยุกต์ใช้กับการทำงานอื่นๆอีกต่อไปในอนาคต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ชาริน สิทธิธรรมชารี , “คู่มือการเขียน โปรแกรม Visual Basic Version 6.0” , บริษัท ซัคเซส มีเดีย จำกัด , 392 หน้า , 2544
2. Toshiiku Sashida , “An Introduction to Ultrasonic Motors” , Biddles Ltd , 242 p. , 2536
3. Kenji Uchino , “Micromechatronics” , Marcel Dekker AG , 489 p. , 2546



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้