สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบติดตามวัตถุ

OBJECT TRACKING SYSTEM

TO50200

โดย

นาย สังวาล บกสุวรรณ

นาย สุเมธ จันทรท่าจีน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50200
วัน,เดือน,ปี 27 เม.ย. 2547
.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

สาขาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบติดตามวัตถุ

OBJECT TRACKING SYSTEM

ผู้จัดทำ 1. นายสังวาล บกสุวรรณ 43015333

2. นายสุเมธ จันทรท่าจีน 43015336

.....อาจารย์ที่ปรึกษา

(อาจารย์ถาวร เบญจนราสุทธิ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบติดตามวัตถุ

โดย
นาย สังวาล บกสุวรรณ 43015333
นาย สุเมธ จันทรท่าจีน 43015336

อาจารย์ที่ปรึกษา
อาจารย์ ถาวร เบญจนราสุทธิ์

บทคัดย่อ

วัตถุประสงค์ของโครงงานนี้ เพื่อออกแบบระบบซึ่งใช้เลเซอร์ชี้ตำแหน่งของวัตถุซึ่งกำลังเคลื่อนที่ในระนาบ 2 มิติ โดยอาศัยกล้องวีดีโอในการตรวจจับภาพและโปรแกรมคอมพิวเตอร์สำหรับทำการประมวลผลภาพ ในลำดับแรกภาพจากกล้องวีดีโอถูกนำมาประมวลผลโดยโปรแกรมคอมพิวเตอร์เพื่อหาตำแหน่งปัจจุบันของวัตถุและเลเซอร์ จากนั้นทำการคำนวณผลต่างระหว่างตำแหน่งของวัตถุและแสงเลเซอร์ ลำดับต่อไปทำการควบคุมสเตปเปอร์มอเตอร์ โดยอาศัยแนวความคิดของการควบคุมแบบพี หลังจากปรับตำแหน่งของเลเซอร์แล้ว ก็จะทำการประมวลผลภาพใหม่ที่ได้จากกล้องวีดีโอ หากเลเซอร์ยังไม่สามารถชี้ตำแหน่งวัตถุได้ ก็จะทำการปรับสเตปเปอร์มอเตอร์ต่อไปจนกระทั่งเลเซอร์ชี้วัตถุได้พอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OBJECT TRACKING SYSTEM

By

Mr. SUNGWAN BOKSUWAN

Mr. SUMATE CHANTACHEEN

Advisor

TAWORN BENJANARASUTH

Academic Year : 2002

ABSTRACT

The purpose of this project is to design the system employing the laser to point the position of the object moving slowly in the x,y plane. First, the picture of the object and the laser is acquired by the digital video and the prepared computer program is used to identify their current positions. Next, the error, which is different between the position of the object and the laser, is calculated. The step motors controlling the position of the x axis and y axis are adjusted by means of this error. This process is continuously executed until the laser can exactly point the object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้รวมถึงตัวโครงงาน สามารถดำเนินการได้สำเร็จลุล่วงไปด้วยดี ทั้งนี้เนื่องมาจากการได้รับความกรุณาเป็นอย่างสูงจากอาจารย์ถาวร เบญจนราสุทธิ์ ซึ่งเป็นอาจารย์ที่ปรึกษาและคอยให้ชี้แนะ เสนอแนวทางอันเป็นประโยชน์ในการจัดทำโครงงาน และทั้งนี้ต้องขอขอบพระคุณอาจารย์ทุกท่านที่ได้ประสิทธิประสาทวิชาความรู้ทั้งหลายแก่ผู้จัดทำ ซึ่งผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

ผู้จัดทำขอขอบคุณทุกๆท่าน ที่ให้การสนับสนุนและช่วยเหลือในการจัดทำปริญญานิพนธ์ชิ้นนี้ขึ้นมาได้อย่างสมบูรณ์

ผู้จัดทำ
นาย สังวาล บกสุวรรณ
นาย สุเมธ จันทรท่าจีน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	i
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญรูปภาพ	vi
สารบัญตาราง	viii
บทที่1 บทนำ	
1.1 ความเป็นมาของโครงงาน	1
1.2 วัตถุประสงค์ในการดำเนินงาน	1
1.3 ขั้นตอนการศึกษาและจัดทำโครงงาน	2
1.4 เนื้อหาที่จะกล่าวถึงในปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	3
2.1 ทฤษฎีการประมวลผลภาพ	3
2.1.1 การกำหนดสี	3
2.1.2 องค์ประกอบของสี	5
2.1.3 การควบคุมกราฟฟิก	6
2.2 ความรู้เบื้องต้นของพอร์ตขนาน	9
2.2.1 ลักษณะทางกายภาพของพอร์ทขนาน	9
2.2.2 พอร์ตดาต้า	13
2.2.3 พอร์ตควบคุม	15
2.2.4 พอร์ตแสดงสถานะ	17
2.2.5 การนำพอร์ตไปใช้งาน	18
2.3 สเตปเปอร์มอเตอร์	22
2.3.1 ชนิดของสเตปเปอร์มอเตอร์	22
2.3.2 การทำงานของสเตปเปอร์มอเตอร์	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
2.3.3 การกระตุ้นเพื่อขับสเตปเปอร์มอเตอร์	25
2.4 บอร์ดขับสเตปเปอร์มอเตอร์	28
บทที่ 3 หลักการออกแบบ	30
3.1 ส่วนประกอบของโครงงาน	30
3.2 การประมวลผลภาพ	31
3.3 การควบคุมตำแหน่ง	33
3.4 โครงสร้างทางกลศาสตร์	35
3.4.1 การติดตั้งและการส่งกำลังขับ	36
3.4.2 การติดตั้งกล้องดิจิตอลวีดีโอ	37
3.4.3 การติดตั้งเครื่องมือทั้งหมด	38
บทที่ 4 การทดลองและผลการทดลอง	38
4.1 การรับสัญญาณภาพจากกล้องดิจิตอลวีดีโอ	38
4.2 การแปลงสีภาพของวัตถุ	39
4.3 การประมวลผลภาพของวัตถุและแสงเลเซอร์	40
4.4 การทดลองโปรแกรมการติดตามวัตถุ	41
4.5 การทดลองโปรแกรมการติดตามวัตถุ (ในแนวแกน x และแกน y พร้อมกัน)	46
4.6 สรุปผลการทดลอง	50
บทที่ 5 บทวิจารณ์และสรุป	51
5.1 สรุปผลการทดลอง	51
5.2 วิจารณ์การทดลอง	51
5.3 ประโยชน์ที่ได้รับจากการทำโครงงานนี้	51
5.4 ปัญหาที่พบในการทำโครงงานนี้	52
5.5 แนวทางการพัฒนา	52
หนังสืออ้างอิง	53
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
รูปที่ 2.1	กล่องสี RGB	4
รูปที่ 2.2	แสดงการดึงภาพจากกล้องดิจิตอลวีดีโอ	7
รูปที่ 2.3	แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์	9
รูปที่ 2.4	แสดงระบบบัสภายในของพอร์ตขนาน	12
รูปที่ 2.5	วงจรภายในของพอร์ต Data	13
รูปที่ 2.6	วงจรภายในของพอร์ต Control	16
รูปที่ 2.7	แสดงวงจรภายในของพอร์ตแสดงสถานะ	17
รูปที่ 2.8	ลักษณะการพันขดลวดบนสเตเตอร์	23
รูปที่ 2.9	ทิศทางการหมุนโรเตอร์ของสเตปเปอร์มอเตอร์ 4 เฟส	24
รูปที่ 2.10	การพันขดลวดของมอเตอร์แบบยูนิโพล่าร์	25
รูปที่ 2.11	วงจรขับสเตปเปอร์มอเตอร์	28
รูปที่ 3.1	แสดงบล็อกไดอะแกรมแสดงการทำงานของระบบ	30
รูปที่ 3.2	แผงผังลำดับการทำงานประมวลผลภาพ	31
รูปที่ 3.3	การแยกสี	32
รูปที่ 3.4	แผงผังลำดับ การควบคุมตำแหน่ง	33
รูปที่ 3.5	แสดงเครื่องมือกลชี้ตำแหน่ง	35
รูปที่ 3.6	การขับเพื่อการเคลื่อนที่ไปในแนวแกน x	36
รูปที่ 3.7	การขับเพื่อการเคลื่อนที่ไปในแนวแกน y	36
รูปที่ 3.8	แสดงการติดตั้งกล้องดิจิตอลวีดีโอ	37
รูปที่ 3.9	แสดงการติดตั้งเครื่องมือทั้งหมดลงกล่อง	37
รูปที่ 4.1	การดึงภาพเข้ามาประมวลผล	38
รูปที่ 4.2	การประมวลผลภาพ	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.3 ประมวลผลภาพ วัตถุ พร้อม เลเซอร์พอย์นเตอร์ 40

รูปที่ 4.4 แสดงการเคลื่อนที่เมื่อรัน โปรแกรมการประมวลผลภาพ 41

รูปที่ 4.5 กราฟแสดงลักษณะการเคลื่อนที่ของเลเซอร์ในแนวแกน x และ y 44

รูปที่ 4.6 กราฟแสดงการเคลื่อนที่ของแกน x เปรียบเทียบกับเวลา (วินาที) 45

รูปที่ 4.7 กราฟแสดงการเคลื่อนที่ของแกน y เปรียบเทียบกับเวลา (วินาที) 45

รูปที่ 4.8 แสดงการเคลื่อนที่เมื่อรัน โปรแกรมการประมวลผลภาพที่เขียนขึ้นใหม่ 46

รูปที่ 4.9 กราฟแสดงการเคลื่อนที่ของแกน x เปรียบเทียบกับเวลา (วินาที) 49

รูปที่ 4.10 กราฟแสดงการเคลื่อนที่ของแกน y เปรียบเทียบกับเวลา (วินาที) 49

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 2.1	สัญญาณสำคัญของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์	11
ตารางที่ 2.2	แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน	15
ตารางที่ 2.3	แสดงแอดเดรสของพอร์ตขนาน	18
ตารางที่ 2.4	การทำงานการขับสเตปเปอร์มอเตอร์แบบฟูลสเตป 1 เฟส	26
ตารางที่ 2.5	แสดงรูปแบบการขับสเตปเปอร์มอเตอร์แบบฟูลสเตป 2 เฟส	27
ตารางที่ 2.6	แสดงรูปแบบการขับสเตปเปอร์มอเตอร์ แบบฮาล์ฟสเต็ป	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

ปัจจุบันมีการประยุกต์ใช้กล้องวีดีโอในหลายๆด้าน เนื่องจากการประมวลผลภาพสามารถทำได้สะดวกโดยอาศัยโปรแกรม MICROSOFT VISUAL BASIC ซึ่งเมื่ออาศัยโปรแกรมนี้สามารถระบุสี หรือขนาดของวัตถุเป็นจำนวนพิกเซล (pixel) ได้

คณะผู้จัดทำ จึงคิดทำโครงงานที่เกี่ยวกับการระบุตำแหน่งของวัตถุที่อยู่กับที่หรือกำลังเคลื่อนที่ หรือกล่าวอีกนัยหนึ่งทำการติดตามการเคลื่อนไหวของวัตถุโดยอาศัยหลักการข้างต้น หลังจากได้ตำแหน่งของวัตถุแล้ว จึงนำไปคำนวณเพื่อหาสัญญาณควบคุมสเตปเปอร์มอเตอร์

โครงงานนี้ต้องอาศัยความรู้ เกี่ยวกับการประมวลผลภาพ สำหรับใช้ในการวิเคราะห์ภาพที่ได้จากกล้อง วีดีโอ และความรู้เกี่ยวกับการควบคุมสเตปเปอร์มอเตอร์ สำหรับการควบคุมการเคลื่อนที่

1.2 วัตถุประสงค์ในการดำเนินงาน

1. เพื่อศึกษาการประมวลผลภาพ และนำไปประยุกต์ใช้ในการควบคุมระบบต่าง ๆ
2. เพื่อศึกษาการเขียนโปรแกรมและใช้งานโปรแกรม MICROSOFT VISUAL BASIC
3. เพื่อให้ได้รับความรู้เกี่ยวกับการติดต่อและส่งข้อมูลผ่านทางพอร์ตขนาน (Parallel port) และขั้นตอนการทำงานของพอร์ตขนาน แล้วสามารถนำไปประยุกต์ใช้งานได้จริง
4. เพื่อสร้างเครื่องต้นแบบเพื่อใช้เป็นแนวทางในการพัฒนาปรับปรุงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขั้นตอนการศึกษาและจัดทำโครงงาน

การศึกษาและจัดทำโครงงานนี้เริ่มต้นจาก การออกแบบและจัดสร้างเครื่องมือกลชี้ตำแหน่ง จากนั้นทำการเขียนโปรแกรมที่จะทำการประมวลผลภาพ เพื่อที่จะส่งข้อมูลออกทางพอร์ตขนาน ไปขับสเตปเปอร์มอเตอร์

โครงงานชิ้นนี้จะต้องอาศัยความรู้เกี่ยวกับการประมวลผลภาพ สำหรับใช้วิเคราะห์ภาพที่ได้จากกล้องดิจิตอลวีดีโอ เพื่อให้คอมพิวเตอร์หรือว่าตัวควบคุม รู้ว่าเลเซอร์ชี้ตำแหน่งอยู่ที่ตำแหน่งใด และยังต้องมีความรู้เกี่ยวกับการขับสเตปเปอร์มอเตอร์สำหรับการขับให้อุปกรณ์ชี้ตำแหน่งเคลื่อนที่

ในโครงงานชิ้นนี้ได้ทำการเขียนโปรแกรมส่งภาพมาจากกล้องได้โดยตรง ทำให้เราสามารถนำค่ามาคำนวณได้อย่างรวดเร็วซึ่งไม่ต้องมาจับภาพจากหน้าจอเอาไปคำนวณซึ่งทำให้ผลการตอบสนองที่ช้ากว่า สุดท้ายทำการจูนค่าของตัวควบคุมแบบพี เพื่อให้เลเซอร์ชี้ตำแหน่งของวัตถุได้อย่างรวดเร็วและเหมาะสม

1.4 เนื้อหาที่จะกล่าวในปริญญานิพนธ์

เนื้อหาที่จะกล่าวถึงในปริญญนิพนธ์ฉบับนี้คือ ในบทที่ 2 จะแสดงหลักการและทฤษฎีที่เกี่ยวข้อง ในการออกแบบและนำเอาความรู้ไปประยุกต์ใช้ในการจัดทำโครงงาน และการประมวลผลภาพ รวมถึงการใช้งานสเตปเปอร์มอเตอร์ ในบทที่ 3 จะอธิบายเกี่ยวกับการออกแบบระบบจริงและส่วนประกอบทางด้านอาร์ดแวร์ของระบบ รวมถึงแนวคิดในการออกแบบ บทที่ 4 เป็นส่วนของการวิเคราะห์ผลและการทดลองละเปรียบเทียบในการขับสเตปเปอร์มอเตอร์ทีละแกน และแบบขับทั้งสองแกนพร้อมกัน ในบทที่ 5 จะเป็นส่วนวิจารณ์และสรุปผลการดำเนินงานและปัญหาที่ประสบ และแนวทางการปรับปรุงพัฒนาโครงงานชิ้นนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและความรู้ที่เกี่ยวข้อง

ในการจัดทำโครงงานชิ้นนี้เราจำเป็นจะต้องศึกษาเกี่ยวกับทฤษฎีของการประมวลผลภาพ การเขียนโปรแกรมติดต่อกับพอร์ตขนานและความเข้าใจในการขับสเตปเปอร์มอเตอร์เพื่อที่จะนำไปใช้งานได้อย่างถูกต้องและเหมาะสม ส่วนของทฤษฎี จะขอกล่าวถึงมีหัวข้อดังต่อไปนี้

2.1 ทฤษฎีการประมวลผลภาพ

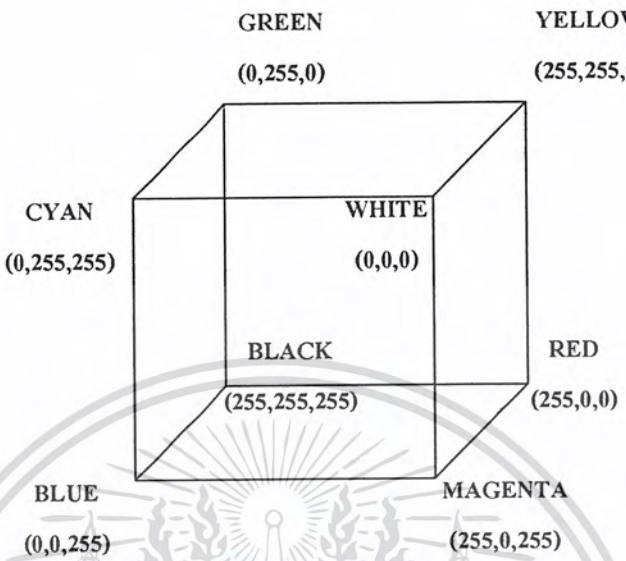
2.1.1 การกำหนดสี

ใน คอมพิวเตอร์ สีทุกสีจะประกอบด้วย 3 สีพื้นฐาน คือ แดง เขียว และน้ำเงิน(RGB) เราสามารถที่จะออกแบบได้เกือบทุกสี โดยการผสมสัดส่วนของสีพื้นฐาน โมเดล ของการออกแบบสีตั้งอยู่บนพื้นฐานของความเข้มข้นขององค์ประกอบของ RGB ซึ่งเรียกว่า RGB โมเดล และนี่ก็เป็นแนวความคิดพื้นฐานของ คอมพิวเตอร์ กราฟฟิก

ทุก ๆ สีซึ่งเราสามารถจินตนาการได้สามารถสร้างขึ้นมาได้ โดยการผสมสัดส่วนที่เหมะสมของสามสีพื้นฐาน ดังนั้นแต่ละสีถูกนำเสนอโดยสัดส่วน (Red,Green,Blue) ซึ่งสีพื้นฐานทั้งสามในแต่ละสี ถูกนำเสนอด้วยขนาดไบต์ค่าที่เล็กที่สุด 0 ชี้ให้เป็นถึงการไม่มีสีเลย ค่าที่ใหญ่ที่สุด 255 บ่งบอกถึงความเข้มข้นสูงสุด สัดส่วน (0,0,0) เป็นสีดำ เพราะสีทุกสีไม่มีความเข้มข้นเลย และสัดส่วน (255,255,255) เป็นสีขาว สีอื่นๆ ซึ่งมีการรวมกันได้มากมายเช่น (255,0,0) เป็นสีแดงบริสุทธิ์ (0,255,255) เป็นสีน้ำเงินเข้ม และ (0,128,128) เป็นสีน้ำเงินอ่อนๆ การรวมกันที่เป็นไปได้ของสีพื้นฐานเป็น 256x256x256 หรือ 16,777,216 สี

กระบวนการของการสร้างสี ซึ่งมีส่วนประกอบพื้นฐาน 3 ส่วนประกอบ ตั้งอยู่บนพื้นฐานของ RGB สี่เหลี่ยมสี ดังรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.1 กล่องสี RGB

สามมิติของสี่เหลี่ยมสีสอดคล้องกับสามสีที่เป็นพื้นฐาน มุมของสี่เหลี่ยมแทนด้วยสีต่างๆดังรูป สีพื้นฐานอยู่ที่แกน x,y และz ตามลำดับ สีที่เป็นส่วนกลับ(complementary colors)สามารถคำนวณได้อย่างง่ายๆโดยการลบค่าของสีจาก 255 ยกตัวอย่าง สี (0,0,255)เป็นสีน้ำเงินบริสุทธิ์ดังนั้นสีน้ำเงินและเหลืองเป็นสีในส่วนกลับกัน และทั้งสองอยู่ที่มุมตรงกันข้ามของสี่เหลี่ยมสี ส่วนมุมอื่น ๆ ก็เช่นเดียวกัน

องค์ประกอบของสีที่มุมของสี่เหลี่ยมสี จะมีความเข้มข้นสูงที่สุดหรือ ไม่มีความเข้มข้นเลยเท่านั้น เมื่อคุณเคลื่อนที่จากมุมหนึ่งไปยังอีกมุมหนึ่งตามขอบที่เหมือนกันจะมีเพียงแค่องค์ประกอบเดียวเท่านั้นที่เปลี่ยนแปลง ยกตัวอย่างเช่น เมื่อเราทำการเคลื่อนที่จากมุมสีเขียว ไปยังมุมสีเหลือง องค์ประกอบของสีแดงจะเปลี่ยนแปลงจาก 0 ถึง 255 เมื่อเราเคลื่อนที่ออกจากมุมของสีเหล่านี้เราจะได้รับทุกโทนสี จากเขียวไปเหลือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 องค์ประกอบของสี

ในการประมวลผลภาพเราต้องการทราบองค์ประกอบของสีเหล่านั้น นั่นคือเราต้องอ่านค่าของพิกเซล แล้วแยกองค์ประกอบของ RGBจากนั้นนำไปใช้งานแยกกัน ดังนั้นเราจึงจำเป็นจะต้องเขียนฟังก์ชั่นซึ่งทำหน้าที่แยกองค์ประกอบของสี เพื่อที่จะทำเช่นนี้ เราจำเป็นที่จะต้องพิจารณาว่าค่าองค์ประกอบของสีทั้งสามถูกเก็บไว้อย่างไร ในตัวแปร LONG INTERGER ในตัวแปร ดังกล่าวประกอบด้วย 4 ไบต์ โดยไบต์แรก (THE MOST SIGNIFICANT BYTE) เก็บ 0 ต่อมาเป็นองค์ประกอบของสีน้ำเงิน เขียว และแดง ตามลำดับ ค่า LONG INTERGER ซึ่งสอดคล้องกับสัดส่วน (64,32,192) เป็น12591168 จำนวนนี้ไม่มีความเหมือนกับจำนวนเดิมเลย ถ้าเรานำเสนอองค์ประกอบของสีด้วยเลขฐานสิบหก (HEXADECIMAL FORMAT) แล้ว RGB สัดส่วนเป็น (40,20,C0) ถ้าเราทำการวางจำนวนนี้สลับซึ่งกันและกัน (C0,20,40) ผลจะเป็นเลขฐานสิบหกของ LONG INTERGER สองหลักสุดท้ายสอดคล้องกับองค์ประกอบสีแดง สองหลักถัดมาสอดคล้องกับองค์ประกอบสีเขียว และหลักที่มีความสำคัญสูงสุดสอดคล้องกับค่าสีน้ำเงิน และนี่ก็เป็นเหตุผลว่าทำไมเลขฐานสิบหก จึงถูกใช้มากในการกำหนดค่าสี

STATEMENT ด้านล่างเป็นการแยกค่าขององค์ประกอบของสี จากค่าของสีที่ถูกเก็บในตัวแปร 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 Cordinate ของจุดซึ่งเราทำการตรวจสอบว่าค่าเนื่องจาก ค่าของสีแดงเก็บไว้ไน ไบต์ สุดท้าย มันจึงเป็นเศษของการหารด้วย 256

เพื่อที่จะแยกไบต์ถัดไป เราจำเป็นจะต้องทำให้ไบต์ด้านหน้าและไบต์ด้านหลังของมันเป็นศูนย์ ซึ่งทำได้โดยการใช้ตัวดำเนินการ AND กับเลขฐานสิบหก 00FF00 ค่าของสีเขียวถูกเก็บไว้ในไบต์ที่สอง ดังนั้นเพื่อที่จะลดให้มันอยู่ที่ ไบต์ที่หนึ่ง นำค่าที่ได้หารด้วย 256

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายเราทำเช่นเดียวกับไบต์ที่ 3 โดยดำเนินการ AND ค่าสีด้วยค่า FF0000 และหลังจากนั้นหารผลที่ได้ โดย 65536 ซึ่งจะเท่ากับค่า (256x256) หรืออาจใช้อีกโปรแกรมหนึ่งซึ่งทำงานได้เช่นเดียวกัน

```
Function Color RGB ( Color1 aslong , R,G,B as interger )
R = Color1 mod256
G = Color 1\65536
B = (Color1\256)mod256
End Function
```

2.1.3 การควบคุมกราฟฟิก (Manipulating Graphics)

ส่วนนี้อธิบายถึงเทคนิคบางอย่างสำหรับการควบคุม การแสดงภาพโดยอาศัย API Graphics Function ของ VISUAl BASIC

- Bitblt ( ) Function

ฟังก์ชั่นนี้ถูกใช้เพื่อที่จะ copy ภาพจากต้นกำเนิด (Source) ไปยังจุดหมายปลายทาง (Dentination) ฟังก์ชั่นจะทำการส่งผ่าน (transfer) พิกเซลจากต้นกำเนิดไปยังอุปกรณ์ปลายทาง เราสามารถประกาศ (declare) Bitblt () Function ได้ดังนี้

```
Declare Function Bitblt Lib " gdi32 "Alias "Bitblt"
Byval hDestDC AS Long, Byval x as Long ,Byval y As Long , Byval n Width As Long ,
Byval n Hight as long ,Byval h scrcdc AsLong ,Byval x src as Long Byval ysrc As
Long ,Byval dwrop as Long ) AsLong
```

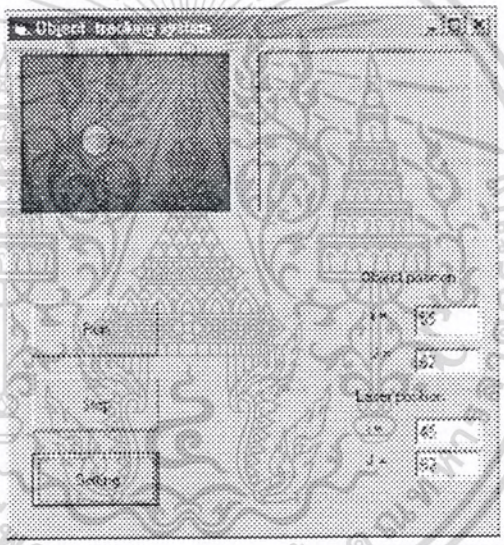
x และ y เป็น coordinate และถ้าเราจำเป็นจะต้องให้ ความสูง และความกว้างพร้อมทั้งจุดเริ่มต้นของภาพต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง

BitBlt Picture 2. hdc,0,0,320,240,Picture.hdc,0,0,vbsrc copy

Picture2 เป็นจุดหมายปลายทางเริ่มต้นที่ x =0 และ y=0 โดยมีความกว้าง 320 และความสูง 240 ซึ่ง copy ภาพจาก Picture 1 (ต้นกำเนิด) เริ่ม copyที่ x = 0 , y = 0 โดย copy หมดทั้งภาพ



รูปที่ 2.2 แสดงการดึงภาพจากกล้องดิจิตอลวีดีโอ

- GetPixel V() Function

จะคืนค่าของพิกเซลที่ coordinate (x,y) ซึ่งสามารถประกาศได้โดย

Public Declare Function GetPixel Lib " gdi32 " Alias "GetPixel
(Byval hdc As long, Byval x as Long , Byval y asLong) as Long

Argument ตัวแรกของ GetPixel () Function คือ hdc คุณสมบัติของ PictureBox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Setpixel () Function

ทำหน้าที่กำหนดค่าของแต่ละพิกเซล หรือพิกเซลที่กำหนด สามารถประกาศได้ดังนี้

```
Public Declare Function Setpixel Lib "gdi32" Alias "setpixel"
(Byval hdc  As Long ,Byval x as long ,Byval y As long ,byval crcolor as long)  as  Long
```

หน้าที่ของแต่ละ Argument คือ

hdc Picturebox สำหรับการกำหนดค่า

x และ y เป็น coordinate

crcolor ค่าของสีที่ ต้องการกำหนดไปยัง pixel

ตัวอย่าง

```
For I    = 0 to 320
For j    = 0 to 240
Color1   = GetPixel ( Piture1.hdc,i,j)
Setpixel  (Picture2.hdc,i,j,Color1)
Next i
Next j
```

จะเป็นการทำสำเนา ภาพจาก Picture1 ไปยัง Picture2 ครั้งละ 1 พิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

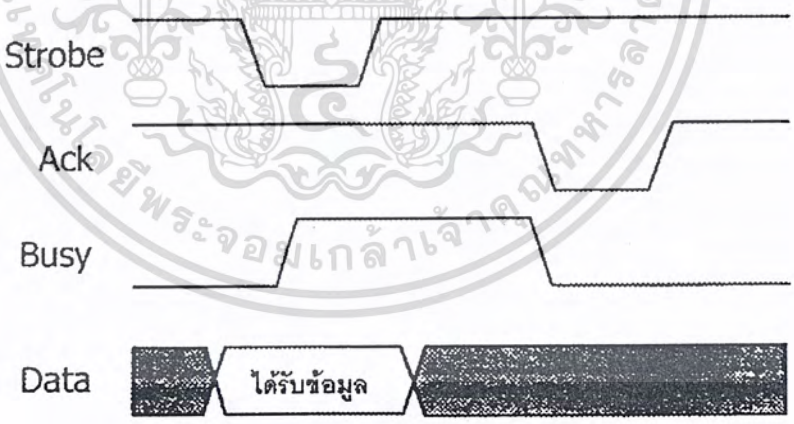
2.2 ความรู้เบื้องต้นของพอร์ตขนาน

พอร์ตขนาน (Parallel port) สาเหตุที่มีชื่อนี้ เนื่องจากการถ่ายทอดข้อมูลของพอร์ตนี้เป็นแบบขนาน สำหรับชื่อเรียกที่นิยมของพอร์ตขนานอีกชื่อหนึ่งคือ พอร์ตเครื่องพิมพ์ (Printer port) เนื่องจากพอร์ตนี้ใช้สำหรับต่อเครื่องพิมพ์นั่นเอง

ด้วยการถ่ายทอดข้อมูลแบบขนานนี้เอง ทำให้พอร์ตขนานมีอัตราการถ่ายทอดข้อมูลสูงกว่าการถ่ายทอดข้อมูลแบบอนุกรมประมาณ 8-10 เท่า และการประมวลผลข้อมูลส่วนใหญ่จะมีขนาด 8 บิต ดังนั้นพอร์ตขนานจึงสามารถรองรับการถ่ายทอดข้อมูล 8 บิตได้โดยไม่ต้องต่อส่วนเพิ่มเติมใด ๆ

2.2.1 ลักษณะทางกายภาพของพอร์ตขนาน

เพื่อให้เข้าใจถึงการนำเอาพอร์ตขนานไปใช้งาน ก่อนอื่นต้องมาทำความเข้าใจก่อนว่า ปกตินั้นการสั่งพิมพ์งานจากคอมพิวเตอร์ไปยังพอร์ตขนานนั้นมีรูปแบบการทำงานภายในอย่างไร ในรูปที่ 2.3 แสดงไดอะแกรมเวลาของติดต่อระหว่างพอร์ตขนานกับเครื่องพิมพ์ ซึ่งจะเห็นได้ว่ามีสัญญาณที่ใช้งานจริง ๆ มีไม่มาก เริ่มจากสัญญาณพอร์ต Data ถูกส่งออกไปยังเครื่องพิมพ์ พร้อมทั้งส่ง



รูปที่ 2.3 แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณสโตรบ (Strobe) ออกไปด้วย เพื่อให้เครื่องพิมพ์รับรู้ว่ามีการส่งข้อมูลใหม่มาที่ขา Data แล้ว จากนั้นคอมพิวเตอร์จะต้องรอการตอบกลับจากเครื่องพิมพ์ นั่นคือเครื่องพิมพ์จะสร้างสัญญาณ Busy หรือเพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเมื่อเครื่องพิมพ์พร้อม เครื่องพิมพ์จะสร้างสัญญาณ ACK ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว

สัญญาณข้อมูลขนาด 8 บิต, สัญญาณ Strobe และสัญญาณ ACK (acknowledge) เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้วส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่น ๆ ร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่ถึง 3 อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์,พิมพ์ข้อมูลที่รับเข้ามาและตอบสนองต่อการใช้งานของผู้ใช้ เช่น การเปลี่ยนฟอนต์ เป็นต้น บางครั้งอาจเกิดเหตุการณ์ที่ไม่ปกติ เช่น บัฟเฟอร์สำหรับรับข้อมูลเต็ม (เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ที่ทำงานทางกล ย่อมทำงานได้ช้ากว่าการส่งข้อมูลของคอมพิวเตอร์) เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากสามารถรับข้อมูลมากกว่านี้ได้แล้ว สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์คือสัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่น กระดาษติด เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์เรียกว่าสัญญาณ Error นอกจากนี้เมื่อคอมพิวเตอร์ต้องการรีเซตเครื่องพิมพ์ คอมพิวเตอร์จะต้องส่งสัญญาณไปยังเครื่องพิมพ์เพื่อรีเซตเครื่องพิมพ์ด้วย สามารถสรุปขาสัญญาณที่จำเป็นสำหรับการติดต่อดังในตารางที่ 2.1

จากตารางที่ 2.1 จะเห็นได้ว่าพอร์ตขนานของคอมพิวเตอร์ยังแยกย่อยออกเป็นอีก 3 พอร์ต ได้แก่ พอร์ตเอาต์พุตที่ทำหน้าที่ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาต์พุตสำหรับสัญญาณ สโตรบ และ รีเซต พอร์ตอินพุตสำหรับการอ่านค่าสัญญาณ Acknowledge, Busy และสัญญาณ Error จากเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิต	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมาใหม่	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลแล้ว	เครื่องพิมพ์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูลใหม่	เครื่องพิมพ์
Error	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	เครื่องพิมพ์
Reset	รีเซตเครื่องพิมพ์	คอมพิวเตอร์

ตารางที่ 2.1 สัญญาณสำคัญ ๆ ของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณอยู่ทั้งหมด 17 เส้น สายสัญญาณเหล่านั้นจะมีรีจิสเตอร์ 3 ตัวควบคุมการทำงาน ดังนี้

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

บล็อคไดอะแกรมในรูปที่ 2.4 แสดงระบบบัสของคอมพิวเตอร์สำหรับการติดต่อกับพอร์ตขนาน สัญญาณเอาต์พุตจากพอร์ตขนานจะถูกส่งไปยังคอนเน็กเตอร์แบบ DB-25 สำหรับคอมพิวเตอร์ส่วนใหญ่ในปัจจุบันพอร์ตขนาน จะมีมาพร้อมกับเมนบอร์ดไม่จำเป็นต้องใช้การ์ดเสียบเพิ่มเติมเหมือนในอดีต พร้อมทั้งมีฟังก์ชันการทำงานที่ซับซ้อนขึ้น แต่ยังคงสนับสนุนการทำงานของพอร์ตขนานในรูปแบบมาตราฐาน (SPP) อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

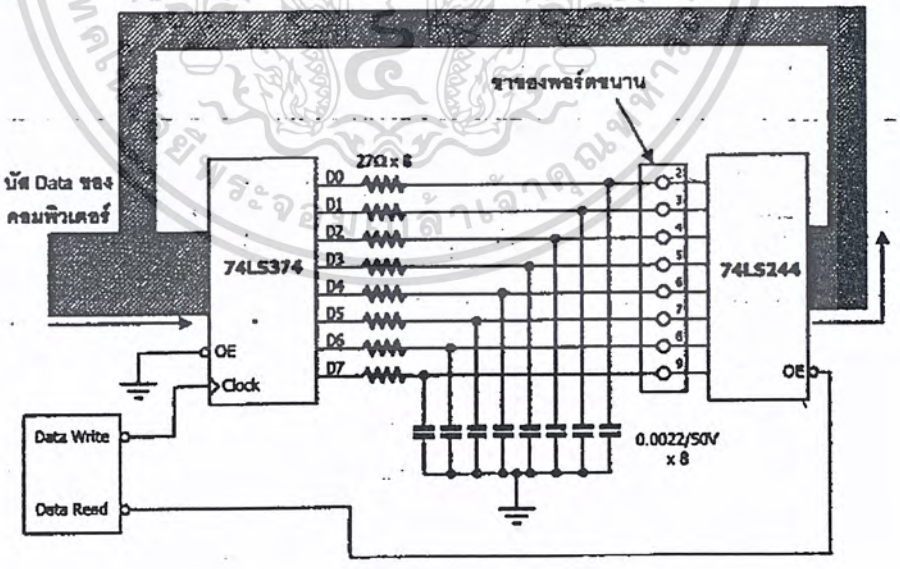
Control Write สัญญาณอีนาเบิลสำหรับนำข้อมูลที่อยู่ในบัส Data ไปออกที่ขา Control ของพอร์ตขนานสำหรับพอร์ตนี้นอกจากจะส่งข้อมูลไปยังพอร์ตขนานแล้ว ยังทำหน้าที่อีนาเบิลการอินเตอร์รัปต์ของการเปลี่ยนแปลงสัญญาณที่พอร์ต สถานะอีกด้วย

Control Read สัญญาณอีนาเบิลสำหรับอ่านค่าข้อมูลจากขา Control มาเก็บไว้ในบัส Data

Status Read สัญญาณอีนาเบิลสำหรับอ่านค่าข้อมูลจากขาพอร์ตสถานะมาเก็บไว้ในบัส Data แอดแดรสของพอร์ตขนานซึ่งกำหนดไว้ 3 ตำแหน่ง คือ LPT1,LPT2 และ LPT3

2.2.2 พอร์ตดาต้า (Data port)

จากรูปที่ 2.5 แสดงให้เห็นว่าพอร์ตดาต้า Data ประกอบไปด้วยบัฟเฟอร์ 1 ตัวและไอซีแลตช์อีก 1 ตัว เมื่อคอมพิวเตอร์ต้องการส่งข้อมูลไปยังเครื่องพิมพ์ คอมพิวเตอร์จะเขียนข้อมูลไปยังไอซีแลตช์ 1 ทั้ง 8 บิต เอาต์พุตของไอซีแลตช์ 1 คือ DO-D7 ซึ่งเอาต์พุตนี้จะไปปรากฏอยู่ที่พอร์ตขนานในตำแหน่งขา 2 ถึงขา 9 และที่ขาเอาต์พุตนี้สัญญาณ Data จะส่งกลับไปเป็นอินพุตของบัฟเฟอร์ 1 ด้วย ทำให้คอมพิวเตอร์สามารถอ่านค่าสถานะปัจจุบันที่เกิดขึ้นกับพอร์ต Data ได้



รูปที่ 2.5 วงจรภายในของพอร์ต Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อคอมพิวเตอร์ส่งข้อมูล ข้อมูลจะถูกส่งมาจากบัสข้อมูลของคอมพิวเตอร์ผ่านไปให้กับไอซี 74L374 ซึ่งเป็นไอซีแลตช์ข้อมูล และเมื่อต้องการให้ข้อมูลปรากฏที่เอาต์พุต คอมพิวเตอร์จะส่งสัญญาณ Data Write ออกไปที่ขา CLK ของ 74LS374 เอาต์พุตจาก 74LS374 จะถูกกรองด้วยวงจร RC ซึ่งประกอบด้วยตัวต้านทานค่า 27 Ω และตัวเก็บประจุ 0.0022 μF เพื่อให้ช่วงเวลาที่เปลี่ยนจากลอจิก "0" เป็นลอจิก "1" หรือจากลอจิก "1" เป็นลอจิก "0" เป็นไปอย่างช้า ๆ เนื่องจากการเปลี่ยนแรงดันที่รวดเร็วทำให้ให้เกิดสัญญาณรบกวนเหนี่ยวนำข้ามไปยังข้อมูลบิตอื่น ๆ ได้ ทำให้ข้อมูลที่ส่งออกไปมีข้อผิดพลาด จากค่าตัวต้านทานและตัวเก็บประจุในวงจรทำให้เกิดการหน่วงเวลาไปประมาณ 60 นาโนวินาที จากวงจรในรูปที่ 2.6 ทำให้เอาต์พุตของพอร์ต Data มีคุณสมบัติดังนี้

- กระแสซิงก์สูงสุด 24 mA
- กระแสซอร์สสูงสุด 206 mA
- ระดับแรงดันของลอจิก "1" ต่ำสุดเท่ากับ 2.4 V
- ระดับแรงดันสูงสุดสำหรับลอจิก "0" เท่ากับ 0.5 V

สำหรับบัฟเฟอร์สำหรับการอ่านข้อมูลกลับได้แก่เบอร์ 74LS244 ซึ่งเมื่อต้องการอ่านค่าคอมพิวเตอร์จะส่งสัญญาณ DataRead ออกมาเพื่ออีนาเบิลไอซี 74LS244 สำหรับพอร์ตขนานแบบมาตรฐาน (Standard Parallel Port : SPP) พอร์ต Data จะต้องใช้เพื่อการส่งค่าออกเอาต์พุตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DR-25	รีจิสเตอร์	ทิศทาง	ตำแหน่งบิต	ชื่อขาสัญญาณ
1	control	out	$\overline{CO}$	$\overline{STROBE}$
2-9	Data	out	D1-D8	DATA1 - DATA5
10	Status	In	S6	ACK
11	Status	In	$\overline{S7}$	$\overline{BUSY}$
12	Status	In	S5	PE
13	Status	In	S4	SELECT
14	Control	Out	$\overline{C1}$	AUTO FEED
15	Status	In	S3	$\overline{ERROR}$
16	Control	out	C2	$\overline{INIT}$
17	Control	out	$\overline{C3}$	$\overline{SELECT-IN}$
18-25			GND	

ตารางที่ 2.2 แสดงสัญญาณทั้งหมดที่อยู่บนพอร์ตขนาน

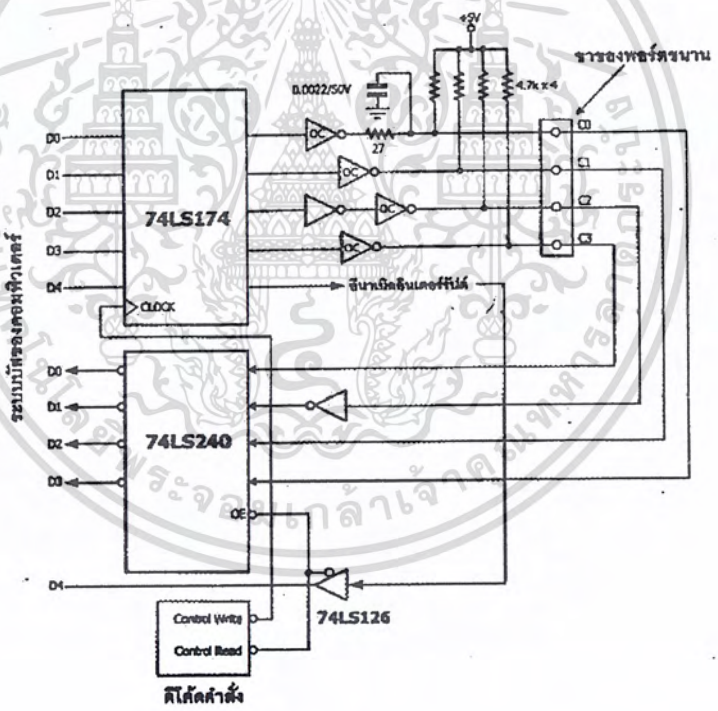
แต่สำหรับพอร์ตขนานที่มีการสื่อสารสองทิศทาง (Bidirectional Parallel Port) สามารถอ่านค่าจากพอร์ต Data ได้ด้วย แต่ก่อนที่จะอ่านค่าต้องจำไว้เสมอว่าจะต้องป้อนค่าเอาต์พุตให้มีค่าลอจิก "1" ทั้งหมดก่อน

2.2.3 พอร์ตควบคุม (Control Port)

พอร์ตควบคุม ใช้สำหรับคอมพิวเตอร์ควบคุมเครื่องพิมพ์ จากตารางที่ 2.2 จะเห็นว่าพอร์ตควบคุม ประกอบไปด้วยบิตเอาต์พุต 4 บิตที่ต่อออกไปยังเครื่องพิมพ์ ส่วนบิตอินาเกิลอินเตอร์รัปต์ไม่ได้ถูกต่อออกไป รูปที่ 2.6 แสดงบล๊อกไดอะแกรมของพอร์ตควบคุม เอาต์พุตของพอร์ตควบคุม มีอินเวอร์เตอร์แบบคอลเล็กเตอร์เปิดต่อร่วมอยู่ โดยเอาต์พุตเหล่านี้จะถูกพูลอัปไว้ด้วยตัวต้านทานค่า 4.7 KΩ สำหรับบิต C2 จะผ่านอินเวอร์เตอร์ถึงสองตัวทำให้ที่เอาต์พุตของบิต C2 ไม่มีการกลับสถานะลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะของพอร์ตควบคุม สามารถอ่านกลับได้โดยการใช้บัฟเฟอร์เบอร์ 74LS240 ซึ่งเอาต์พุตของ 74LS240 มีอินเวอร์เตอร์อยู่ภายใน ทำให้ค่าที่อ่านได้ตรงกับค่าที่ส่งออกไป การควบคุมการอ่านและเขียนข้อมูลกับพอร์ตควบคุมคอมพิวเตอร์จะส่งข้อมูลมาที่ขา Control Write และ Control Read เนื่องจากเอาต์พุตของพอร์ตควบคุมเป็นแบบคอลเล็กเตอร์เปิด ดังนั้นผู้ใช้งานสามารถใช้พอร์ตนี้ในการอ่านค่าสัญญาณอินพุตจากภายนอกได้ โดยก่อนที่จะอ่านค่าจะต้องทำให้ขาพอร์ตที่ต้องการอ่านค่าลอจิก “1” เสียก่อน



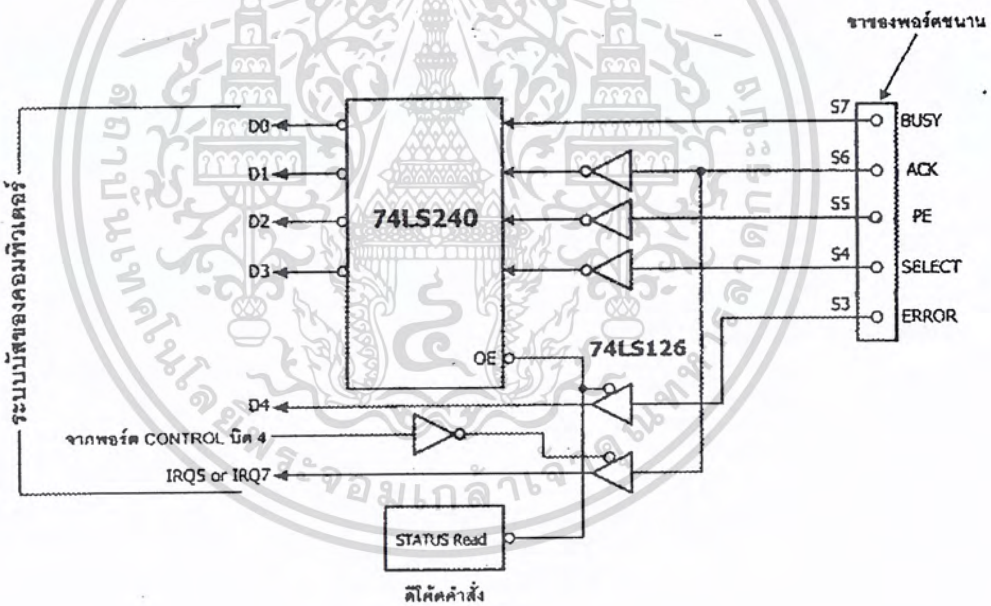
รูปที่ 2.6 วงจรภายในของพอร์ตควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.4 พอร์ตแสดงสถานะ (Status Port)

พอร์ตสถานะเป็นพอร์ตที่คอมพิวเตอร์ใช้สำหรับการอ่านค่าสถานะ จากเครื่องพิมพ์ดังแสดงในรูปที่ 2-7 แสดงรายละเอียดภายในของพอร์ตสถานะ จะสังเกตได้ว่ามีขาสัญญาณอยู่ทั้งหมด 5 สัญญาณด้วยกันและจะเรียกชื่อเป็น S3, S4 , S5, S6 และ S7 ซึ่งตัวเลขนั้นหมายถึงตำแหน่งบิตของขาเหล่านี้ภายในรีจีสเตอร์ Status นั่นเอง สำหรับ S7 จะมีข้อแตกต่างจากบิตอื่น ๆ ที่เมื่อสัญญาณจากภายนอกส่งเข้ามาแล้วจะไม่ผ่านอินเวอร์เตอร์ ในขณะที่ขาอื่นๆ ผ่านอินเวอร์เตอร์ทั้งหมด ดังนั้นเมื่อข้อมูลจากขาอินพุตไปยัง 74LS240 ซึ่งเอาต์พุตมีการกลับสถานะทำให้บิต S7 เป็นบิตเดียวที่มีการกลับสถานะ นอกจากนี้ ในการใช้งานถ้าต้องการให้มีการสร้างสัญญาณอินเตอร์รัปต์ จากขอบขาขึ้นของขา S6 สามารถ กำหนดค่า ได้จากพอร์ต ควบคุมบิต 4



รูปที่ 2 .7 แสดงวงจรภายในของพอร์ตสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50200

2.2.5 การนำพอร์ตขนานไปใช้งาน

สำหรับพอร์ตขนานแบบมาตรฐาน ผู้ใช้งานสามาถนำพอร์ตสถานะจำนวน 5 บิต มาใช้เป็นพอร์ตอินพุตได้ และนำพอร์ตควบคุมจำนวน 4 บิต มาทำพอร์ตเอาท์พุต และพอร์ตเอาต์พุตอีก 8 บิต (พอร์ต Data) ไปใช้งานได้ โดยตรง โดยที่ 4 บิตของพอร์ตเอาต์พุตหรือพอร์ตควบคุม นั้นสามารถดัดแปลงให้ใช้งานเป็นพอร์ตอินพุตขนาด 4 บิตได้ด้วยดังนั้นผู้ใช้งานจึงสามารถนำสัญญาณจากพอร์ตขนานที่มีมากถึง 17 เส้นไปใช้งานในการควบคุมโดยใช้ระดับสัญญาณ TTL

การเขียนโปรแกรมติดต่อกับพอร์ตขนาน

พอร์ตขนานของคอมพิวเตอร์จะมีลักษณะเช่นเดียวกับอุปกรณ์อินพุตเอาต์พุตตัวอื่น ๆ คือ เมื่อต้องการติดต่อจะต้องกำหนดแอสแดรสที่ต้องการติดต่อด้วย ตารางที่ 2.3 แสดงแอสแดรสของพอร์ตขนาน โดยแบ่งออกเป็น 3 ตำแหน่งคือ แอดแดรสของรีจีสเตอร์ Data, รีจีสเตอร์ Status, และรีจีสเตอร์ Control โดยแอดแดรสนี้จะมีอยู่ 3 ชุด สำหรับพอร์ตขนาน 3 ชุดคือ LPT1, LPT2 และ LPT3

ชื่อพอร์ต	LPT 1		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 2.3 แสดงแอดเดรสของพอร์ตขนาน

เมื่อต้องการต้องการติดต่อกับพอร์ตขนานในตำแหน่งใด ก็ให้ส่งค่าข้อมูลออกไปที่พอร์ตขนานในตำแหน่งนั้น ๆ ยกตัวอย่างการเขียนโปรแกรมด้วย QBASIC เพื่อส่งค่าลอจิก "1" ออกไปทุกบิตของพอร์ต Data ของ LPT1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจะต้องเขียนโปรแกรมดังนี้

OUT &H378, &HFF

โดยที่เครื่องหมาย &H ที่แสดงนั้นหมายถึงตัวเลขฐานสิบหก

คำสั่ง OUT เป็นการส่งค่าข้อมูลออกเอาต์พุตของพอร์ตอินพุตเอาต์พุต

คำสั่ง 378 เป็นแอดเดรสของรีจีสเตอร์ Data สำหรับ LPT1

ค่าข้อมูล FF เป็นข้อมูลเลขฐานสิบหก ซึ่งหมายถึงการให้บิตทุกบิตของรีจีสเตอร์ Data มีลอจิกเป็น "1" นั่นเอง

ส่วนการอ่านค่าจากพอร์ตขนานมายังคอมพิวเตอร์ผ่านทางพอร์ตสถานะของ LPT1 สามารถเขียนโปรแกรมด้วย QBASIC ได้ดังนี้

Temp = INP (&H379)

โดยที่

คำสั่ง INP(-) เป็นคำสั่งสำหรับการอ่านค่าข้อมูล

ค่า 379 เป็นตำแหน่งแอดเดรสของรีจีสเตอร์ STATUS สำหรับ LPT1 ในตัวเลขฐานสิบหก

ตัวแปร Temp เป็นตัวแปรที่ใช้เก็บข้อมูลที่อ่านได้จากพอร์ตขนาน

สำหรับโปรแกรมอื่น ๆ เช่น แอสเซมบลี้,เทอร์โบปาสคาล หรือเทอร์โบซี จะมีรูปแบบการเขียนโปรแกรมที่แตกต่างกันบ้าง แต่อย่างไรก็ตามโปรแกรมทุกตัวต่างก็ใช้วิธีการเดียวกันคือ กำหนดแอสเดรสที่ทำการติดต่อจากนั้นจึงติดต่อกับแอสเดรสเหล่านั้นด้วยคำสั่งสำหรับการอ่านหรือเขียน

การเขียนโปรแกรมติดต่อกับพอร์ตขนานด้วย VISUAL BASIC

การเขียนโปรแกรมด้วย VISUAL BASIC ชุดคำสั่งส่วนใหญ่จะมีรูปแบบใกล้เคียงกับ QBASIC แต่ VISUAL BASIC จะไม่มีคำสั่งสำหรับการติดต่อกับพอร์ตโดยตรงคือ คำสั่ง Inp ( )และคำสั่ง Out ( ) เหมือนกับ QBASIC ดังนั้นเพื่อให้สามารถติดต่อกับพอร์ตขนานได้จึงจำเป็นต้องเพิ่มโปรแกรมบางตัวเข้าไป โดยโปรแกรมที่เพิ่มเข้าไปนี้จะอยู่ในรูปของ DLL (Dynamic Linked Library)

ไฟล์ DLL นี้จะมีอยู่ 2 ไฟล์คือ inpout.dll และ inpout32.dll โดย inpout.dll นั้นใช้สำหรับระบบปฏิบัติการ 16 บิตหรือวินโดวส์ 3.1 นั่นเอง ส่วน inpout32.dll จะใช้สำหรับระบบปฏิบัติการที่เป็น 32 บิตซึ่งก็คือวินโดวส์95 หรือวินโดวส์ 98

สำหรับตำแหน่งที่ใช้เก็บไฟล์ inpout.dll หรือ inpout32.dll นั้นจะต้องเก็บไว้ที่ไดเรกตอรี่ SYSTEM ซึ่งอยู่ภายในไดเรกตอรี่ที่เก็บโปรแกรมวินโดวส์ โดยส่วนใหญ่จะมีชื่อเป็น Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่าในโปรแกรมเพื่อเรียกใช้งานไฟล์ DLL มีรูปแบบการกำหนดค่าดังนี้

สำหรับระบบปฏิบัติการ 16 บิต

```
Declare Function Inp% Lip "Inpout.Dll" Alias "Inp16" (ByVal PortAddress%)
Declare Sub Out Lip "Inpout.Dll" Alias "Out 16" (ByVal PortAddress%,ByVal ByteToWrite%)
```

สำหรับระบบแบบปฏิบัติการ 32 บิต

```
Public Declare Function Inp Lip "inpout 32.dll"
Alias "Inp32" (ByVal PortAdress As Integer) As Integer
Public Declare Sub Out Lip "inpout32.dll"-
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

ในกรณีที่มีการใช้งานโปรแกรมทั้ง 2 ระบบปฏิบัติการ สามารถเพิ่มเติมคำสั่ง IF เข้าไปเพื่อตรวจสอบระบบปฏิบัติการก่อนที่จะเลือกใช้งาน DLL ตัวที่ต้องการ โดยสามารถเขียนโปรแกรมได้ดังนี้

```
# If Win32 Then
'Declare Inp and Out for port I/0
Public Declare Function Inp Lip "inpout32.dll"
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lip "inpout32.dll"-
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
#Else
Declare Function Inp Lip "Inpout.dll" (ByVal Port%) As Integer
Declare Sub Out Lip "Inpout.dll" (ByVal Value%)
#End If
```

แต่การกำหนดในสองรูปแบบหลังนี้จะสามารถใช้งานได้กับ VISUAL BASIC ตั้งแต่เวอร์ชั่น 4 ขึ้นไปเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมาถึงตรงนี้ อาจเกิดคำถามขึ้นว่า ทำไมไมโครซอฟต์ผู้พัฒนาซอฟต์แวร์ VISUAL BASIC จึงไม่รวมคำสั่ง Inp และคำสั่ง Out ไว้ในโปรแกรม VISUAL BASIC เนื่องจากว่าการเขียนและอ่านข้อมูลไปยังพอร์ตหรือหน่วยความจำโดยตรงนั้นอาจทำให้เกิดมีปัญหาแฮงก์หรือทำงานผิดพลาดได้ และ VISUAL BASIC เป็นระบบปฏิบัติการ ที่ทำงานบนวินโดวส์ซึ่งมีการทำงานแบบมัลติทาร์กิ้ง (multitasking) มีโปรแกรมหลาย ๆ ตัวทำงานอยู่พร้อมกัน ดังนั้นเมื่อเกิดความเสียหายกับโปรแกรมตัวหนึ่งก็อาจจะส่งผลให้โปรแกรมที่ทำงานอยู่ทั้งหมดเสียหายได้ นอกจากนี้การเขียนข้อมูลโดยตรงไปยังพอร์ต อาจจะไปทับซ้อนกับโปรแกรมอื่น ๆ ที่มีการเขียนข้อมูลไปยังพอร์ตเช่นเดียวกัน ส่งผลให้โปรแกรมทำงานผิดพลาด

สำหรับวินโดวส์ 95 นอกจะสามารถใช้งาน DLL ในการติดต่อกับพอร์ตโดยตรงแล้ว ยังสามารถใช้งานโปรแกรมประเภท Visual Device Driver (Vxd) ในการติดต่อกับอุปกรณ์อินพุตเอาต์พุต โดย Vxd จะตัดปัญหาเรื่องการเข้าถึงพอร์ตพร้อมกันของโปรแกรมหลาย ๆ ตัวได้ แต่สำหรับโปรแกรมสั้น ๆ เช่น โปรแกรมอ่าค่าอุณภูมิ โปรแกรมควบคุมอุปกรณ์อินพุตเอาต์พุตปกติซึ่งไม่มีการติดต่อกับพอร์ตอยู่ตลอดเวลา คำสั่ง Inp และ Out ใน DLL ก็ยังทำงานได้ดีและมีรูปแบบการใช้งานที่ง่ายกว่า

ฟังก์ชั่นที่มีอยู่ใน io.dll

PortOut	ใช้ส่งข้อมูลขนาด 1 ไบต์ไปยังพอร์ทที่กำหนด
PortWordOut	ใช้ส่งข้อมูล 1 เวิร์ด (16 บิต) ไปยังพอร์ทที่กำหนด
PortDWordOut	ใช้ส่งข้อมูลดับเบิลเวิร์ด (32 บิต) ไปยังพอร์ทที่กำหนด
PortIn	ใช้อ่านข้อมูลขนาด 1 ไบต์จากพอร์ทที่กำหนด
PortWordIn	ใช้อ่านข้อมูล 1 เวิร์ด (16 บิต) จากพอร์ทที่กำหนด
PortDWordIn	ใช้อ่านข้อมูล ดับเบิลเวิร์ด (32 บิต) จากพอร์ทที่กำหนด
SetPortBit	ใช้เซตข้อมูลในระดับบิตของพอร์ทที่กำหนด
ClrPortBit	ใช้เคลียร์ข้อมูลในระดับของพอร์ทที่กำหนด
NotPortBit	ใช้กลับข้อมูลในระดับของพอร์ทที่กำหนด
GetPortBit	ใช้อ่านสถานะของบิตที่กำหนด
RightPortShift	ใช้เลื่อนข้อมูลของพอร์ทไปทางขวา จะได้ค่าของบิต LSB กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ DLL นี้มีประโยชน์อย่างมากสำหรับ การติดต่อกับพอร์ทอินพุตเอาท์พุตของคอมพิวเตอร์ โดยเฉพาะอย่างยิ่งกับพอร์ทขนานของระบบปฏิบัติการวินโดวส์ ซึ่งในปัจจุบันการติดต่อกับพอร์ทขนานทำได้ยากขึ้น แต่เมื่อใช้ไฟล์ DLL ตัวนี้จะช่วยให้สามารถเข้าถึงและ ควบคุมได้ง่ายและมีประสิทธิภาพ

io.dll สามารถนำไปใช้กับระบบปฏิบัติการวินโดวส์ 95/98/2000/NT หรือกระทั่ง XP จึงช่วยลดเวลาและขั้นตอนในการพัฒนาโปรแกรมสำหรับติดต่อพอร์ท ที่รันบนวินโดวส์ได้อย่างมาก

2.3 สเตปเปอร์มอเตอร์ (stepper motor)

สเตปเปอร์มอเตอร์ คือ อุปกรณ์ที่เคลื่อนที่เป็นสเตป โดยการกระตุ้นด้วยวิธีการทางแม่เหล็กไฟฟ้า ซึ่งเปลี่ยนแปลงสัญญาณ ดิจิตอลอินพุต ซึ่งเป็นการส่งสัญญาณแบบช่วงเวลาไปเป็นการเคลื่อนที่แบบอนาล็อกที่เอาท์พุต

สเตปเปอร์มอเตอร์ มีขดลวดอยู่หลายขด โดยแต่ละขดถูกควบคุมการจ่ายกระแสโดยสัญญาณเปิด – ปิด ทำให้สามารถควบคุมตำแหน่งในการเคลื่อนที่ได้ ตราบที่ยังไม่เกิดการเกินกำลัง จนกระทั่งมีการกระตุ้นโดยสัญญาณควบคุมการจ่ายกระแสอีกครั้ง และเนื่องจาก สเตปเปอร์มอเตอร์ สามารถควบคุมตำแหน่งได้โดยการส่งสัญญาณกระตุ้น ทำให้ไม่จำเป็นต้องใช้ตัววัดตำแหน่งในการทำงาน ข้อเสียที่สำคัญ คือมี ข้อจำกัด ที่ความเร็วเมื่อเปรียบเทียบ มอเตอร์กระแสไฟตรง ที่ขนาดเท่ากันและยังมีการสั่นสะเทือนสูงทำให้การเคลื่อนที่ของสเตปเป็นไปอย่างไม่ราบเรียบ

2.3.1 ชนิดของสเตปเปอร์มอเตอร์

สเตปเปอร์มอเตอร์ เป็นมอเตอร์ประเภทหนึ่งที่ถูกนำไปใช้งานมาก ไม่ว่าจะเป็นส่วนประกอบในหุ่นยนต์ เครื่องจักรกลที่ควบคุมด้วยคอมพิวเตอร์ ต่างก็ใช้สเตปเปอร์มอเตอร์ เป็นตัวขับเคลื่อน หรือในอุปกรณ์อิเลคทรอนิกส์อื่นๆ เราสามารถแบ่งชนิดของสเตปเปอร์มอเตอร์ ได้เป็น 6 ชนิด ตามโครงสร้างพื้นฐาน หรือความแตกต่างของรูปแบบของโรเตอร์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถแบ่งชนิดของสเตปเปอร์มอเตอร์ ได้เป็น 6 ชนิด

1.ชนิดวาริเอเบิลรีลักซ์แตนซ์ (variable reluctance :VR)

2.ชนิดไฮบริดจ์ (hybridge)

3.ชนิดแม่เหล็กถาวร (permanent magnet)

4.ชนิดแรเอิรธ์เพอร์มาเนนต์แม็กเน็ต (rare earth permanent magnet)

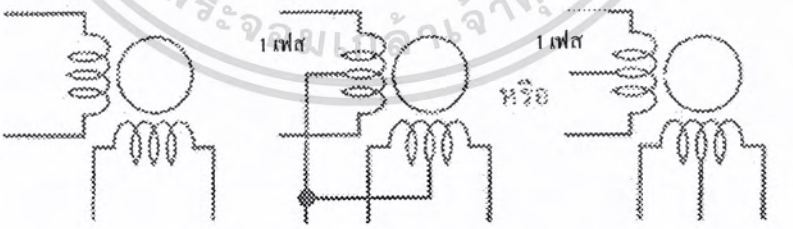
5.ชนิดไบโพล่าร์ (bipolar)

6.ชนิดยูนิโพลาร์ (unipolar)

สำหรับโครงงานนี้เราเลือกใช้แบบยูนิโพลาร์ (unipolar) เนื่องจากหาได้ง่ายและราคาไม่แพง

2.3.2 การทำงานของสเตปเปอร์มอเตอร์

การทำงานของสเตปเปอร์มอเตอร์จะมีความแตกต่างจากมอเตอร์ทั่วไป คือเมื่อป้อนกำลังไฟให้สเตปเปอร์มอเตอร์ ก็จะหมุนเพียงเล็กน้อยตามเส้นรอบวงแล้วหยุด สเตปเปอร์มอเตอร์สามารถกำหนดตำแหน่งการหมุน ได้อย่างละเอียดโดยใช้คอมพิวเตอร์เป็นตัวกำหนดและจัดเก็บตัวเลขของตำแหน่งเหล่านั้นไว้

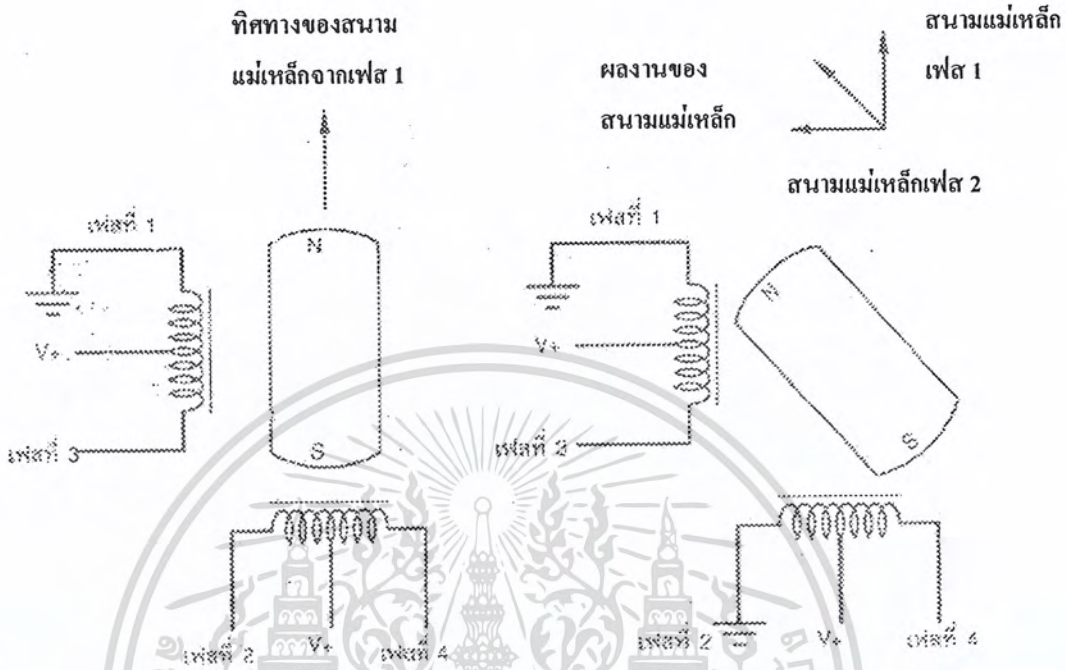


แบบ 4 สาย 2 เฟส แบบบ 5 สาย 4 เฟส แบบ 6 สาย 4 เฟส

(ก) แบบไบโพลาร์ (ข) แบบยูนิโพลาร์

รูปที่ 2.8 ลักษณะการพันขดลวดบนสเตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) เฟสที่ 1 ทำงาน (ข) เฟสที่ 1 และ 2 ทำงาน

รูปที่ 2.9 ทิศทางการหมุนโรเตอร์ของสเตปเปอร์มอเตอร์ 4 เฟส

หลักการทำงานอย่างง่าย ๆของสเตปเปอร์มอเตอร์ แบบยูนิโพล่าร์ 4 เฟส ตัวโรเตอร์เป็นแม่เหล็ก การให้พลังงานแก่ขดลวดใดขดลวดหนึ่งโรเตอร์ก็จะหมุนไป 90 องศา ดังรูปที่ 2.9 (ก.) แต่ถ้าให้ทีเดียว 2 ขดลวดพร้อมกัน โรเตอร์ก็จะหมุนเพียง 45 องศา ดังรูปที่ 2.9(ข.) จะแบบหลังจะสร้างแรงบิดได้มากกว่าแบบแรก สเตปเปอร์มอเตอร์จะมีมุมในการเคลื่อนที่แต่ละสเตปเป็น 1.8 องศา ดังนั้นที่โรเตอร์จะต้องมีขั้วแม่เหล็ก 50 ขั้ว (90/50=1.8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเตปเปอร์มอเตอร์แบบยูนิโพล่าร์

สเตปเปอร์มอเตอร์ชนิดนี้ได้รับการพัฒนามาอย่างต่อเนื่อง จนในปัจจุบันเป็นสเตปเปอร์มอเตอร์ ที่นิยมใช้อย่างแพร่หลายมากที่สุดและหาได้ง่าย มีลักษณะการพันขดลวดของมอเตอร์ดังแสดงในรูป 2.10

แบบ 5 สาย 4 เฟส แบบ 6 สาย 4 เฟส

รูป2.10 การพันขดลวดของมอเตอร์แบบยูนิโพล่าร์

สเตปเปอร์มอเตอร์แบบนี้มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ แต่ละขดแบ่งเป็น 2 เฟส รวมทั้งมอเตอร์ทั้งตัวจะมี 4 เฟสคือ เฟส 1 2 3 และ 4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่ายไฟเลี้ยง ทำให้สเตปเปอร์มอเตอร์แบบนี้มีทั้งแบบ 5 สายและแบบ 6 สาย ถ้าเป็นแบบ 5 สาย จะเป็นการนำสายไฟเลี้ยงของขดลวดทั้งสองมาต่อร่วมกันเป็นสายเดียว

2.3.3 การกระตุ้นเพื่อขับสเตปเปอร์มอเตอร์

สามารถทำได้โดยการจ่ายพลังงานไฟฟ้าไปยังขดลวดแต่ละขดบนแกนแม่เหล็กคงที่ ซึ่งต้องป้อนเป็น ลำดับตามรูปแบบที่ถูกต้อง โดยสามารถแบ่งได้ 3 รูปแบบคือ

1. ฟูลสเตป 1 เฟส (full – step 1 phase)
2. ฟูลสเตป 2 เฟส (full – step 2 phase)
3. ฮาล์ฟสเตป (half step)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แบบฟูลสเตป 1 เฟส (full – step 1 phase)

เป็นการกระตุ้นที่ง่ายที่สุด โดยกระตุ้นขดลวดทีละขดไล่เรียงกันไป เช่นเริ่มต้นที่ขด 1 2 3 และ 4 แล้ววนกลับมาขดที่ 1 หรือเริ่มที่ขดที่ 1 แล้วย้อนไปยังขดที่ 4 3 2 แล้วกลับมายังขดที่ 1 อีกครั้งทำให้ทิศทางการหมุนสวนกัน ในการกระตุ้นแบบนี้จึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น การกระตุ้นแบบนี้มีราคาถูกและง่าย สรุปขั้นตอนการทำงานของฟูลสเตป 1 เฟส ดังตาราง

สเตปที่	เฟสที่1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

ตารางที่ 2.4 การทำงานการขับสเตปเปอร์มอเตอร์แบบฟูลสเตป 1 เฟส

2. แบบฟูลสเตป 2 เฟส

จะกระตุ้นโดยการจ่ายแรงดันไปที่ขดลวด 2 ขด ที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไป ขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปเป็นขดที่ 3 และ 4 ถัดไปเป็นขดที่ 4 และ 1 แล้วกลับมาที่ขดที่ 1 และ 2 วนไปตามลำดับเช่นนี้ หรือเริ่มที่ขดที่ 1 และ 4 ตามด้วยขดที่ 4 และ 3ถัดไปเป็นขดที่ 3 และ 2 ต่อไปเป็นขดที่ 2 และ1 แล้ววนกลับมาที่ขดที่ 1 และ 4 ทิศทางการหมุนจะสวนกัน ดังแสดงขั้นตอนการกระตุ้นในตารางที่ 2.5
การกระตุ้นแบบนี้สามารถเพิ่มแรงบิดได้มากกว่าแบบ 1 เฟส แกนแม่เหล็กเคลื่อนที่ภายในมอเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน และหมุนต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือต้องใช้กำลังไฟฟ้าเพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเตปที่	เฟสที่1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

ตารางที่ 2.5 แสดงรูปแบบการขับสเตปเปอร์มอเตอร์แบบฟูลสเตป 2 เฟส

3. แบบฮาล์ฟสเตป

เป็นรูปแบบที่ผสมผสานระหว่างการกระตุ้นแบบ 1 และ 2 เฟส เพื่อเพิ่มจำนวนของสเตปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะกระตุ้นขดลวดเรียงกันไปเป็นลำดับดังนี้ เริ่มจากขดลวดที่ 1 1 และ 2 2 2 และ 3 3 3 และ 4 4 4 และ 1 แล้ววนกลับมายังขดลวดที่ 1 ดังแสดงในตารางที่ 2.6

แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้น เนื่องจากช่วงของการเคลื่อนที่ในแต่ละสเตปมีระยะสั้นลง แต่ละสเตปเกิดแรงดึงจากขดลวด 2 ขด ที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องพึงระวังว่า เมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องหมุนถึง 2 สเตป จึงจะได้ระยะเท่ากับแบบ 2 เฟสเป็นอย่างน้อย จึงจะเพียงพอ

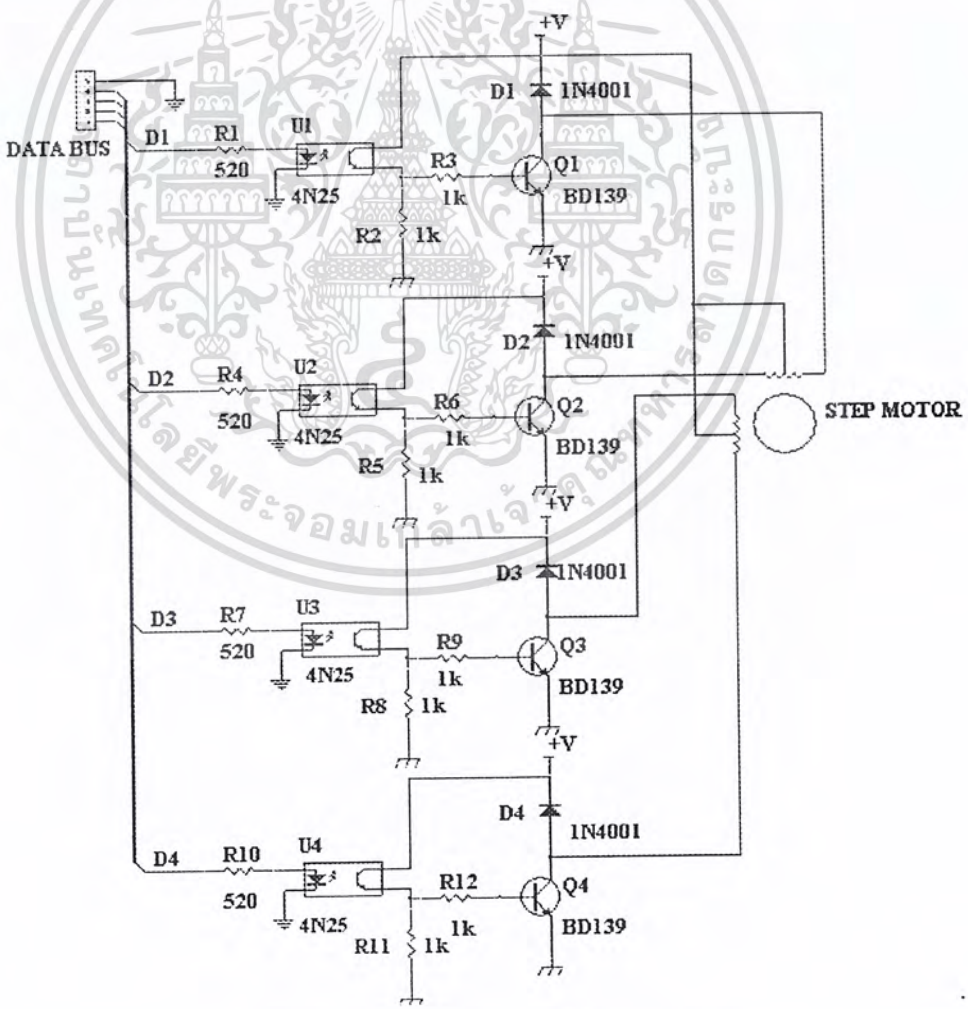
สเตปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

ตารางที่ 2.6 แสดงรูปแบบการขับสเตปเปอร์มอเตอร์ แบบฮาล์ฟสเต็ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 บอร์ดขับสเตปเปอร์มอเตอร์

โดยความสามารถพื้นฐานของพอร์ตขนานแล้วไม่สามารถนำไปขับอุปกรณ์เอาท์พุตโดยตรงได้ เนื่องจากข้อจำกัดด้านความสามารถในด้านการจ่ายกระแส ดังนั้นกรณีที่ต้องการจ่ายพลังงานสูงจะต้องมีอุปกรณ์ที่ทำหน้าที่จ่ายแรงดันและกระแสที่สูงโดยเฉพาะ เราเรียกอุปกรณ์เหล่านี้ว่าอุปกรณ์ขับ ในโครงงานนี้เราเลือกที่จะใช้ทรานซิสเตอร์ในการขับโดยทำการสร้างบอร์ดทดลอง และบอร์ดที่ทำการสร้างขึ้นมีการแยกกราวด์ทางไฟฟ้าของส่วนอินพุตที่เชื่อมต่อกับพอร์ทขนานของคอมพิวเตอร์ และเอาท์พุต ที่เชื่อมต่อกับสเตปเปอร์มอเตอร์ออกจากกัน โดยใช้การเชื่อมต่อทางแสงอาศัยอุปกรณ์ที่เรียกว่า ออปโต้คัปเปลอร์ (opto-coupler)ในการถ่ายทอดสัญญาณควบคุม วงจรของบอร์ดขับสเตปเปอร์มอเตอร์ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 วงจรขับสเตปเปอร์มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณอินพุตมาจากคอนเน็กเตอร์ DATABUS ซึ่งเชื่อมต่อเข้ากับคอมพิวเตอร์ แล้วส่งไปขับ LED อินฟราเรดซึ่งอยู่ภายในออปโต้คัปเปลอร์ U1- U4 โดยผ่านตัวต้านทาน R1, R4, R7 และ R10 เพื่อจำกัดกระแสให้กับ LED ในขณะที่เอาท์พุตของ ออปโต้คัปเปลอร์ จะต่อเข้ากับทรานซิสเตอร์ Q1-Q4 เบอร์ BD139 เพื่อขับมอเตอร์ต่อไป เมื่อป้อนลอจิก "1" ให้กับออปโต้คัปเปลอร์ จะทำให้ LED ภายในออปโต้คัปเปลอร์ทำงาน จะกระตุ้นให้โฟโต้ทรานซิสเตอร์ภายในออปโต้คัปเปลอร์ทำงาน เกิดแรงดันผ่านตัวต้านทานป้อนเข้าที่ขาเบสของทรานซิสเตอร์ ทำให้ทรานซิสเตอร์ BD139 นำกระแส เสมือนว่าขดลวดด้านหนึ่งถูกต่อลงกราวด์ จึงเกิดกระแสไหลผ่านขดลวด ทำให้เกิดการหมุนขึ้น การหมุนของสเตปเปอร์มอเตอร์จะเป็นอย่างไร ขึ้นอยู่กับข้อมูลที่ป้อนให้แก่วงจรขับบนบอร์ดนี้

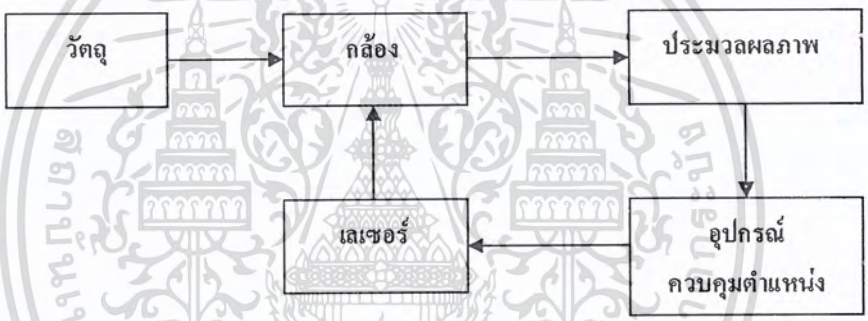
สำหรับไฟเลี้ยงของสเตปเปอร์มอเตอร์ จะแยกออกจากวงจรอินพุตที่เชื่อมต่อกับคอมพิวเตอร์ โดยต้องป้อนแรงดันตามพิกัดของมอเตอร์ที่ต้องการขับ แต่มีข้อจำกัดของวงจรนี้ว่าไม่ควรเกิน 30 โวลต์ กระแสไฟฟ้าประมาณ 1 แอมป์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการออกแบบ

ภาพของวัตถุและเลเซอร์ถูกถ่ายผ่านกล้องวีดีโอ ก่อนที่ภาพจะถูกประมวลผล เพื่อหาตำแหน่งโดยอาศัยอุปกรณ์ประมวลผลภาพ (computer) ถ้าตำแหน่งของวัตถุและตำแหน่งของ เลเซอร์ แตกต่างกัน อุปกรณ์ควบคุมตำแหน่ง จะปรับตำแหน่งของเลเซอร์ให้ทับกับตำแหน่งของวัตถุ หรือกล่าวอีกนัยหนึ่ง คือให้ผลต่างระหว่างตำแหน่งของภาพและของ เลเซอร์กลายเป็นศูนย์ โดยมีแนวความคิดในการทำงานของระบบโดยรวมดังรูปที่ 3.1



รูปที่ 3.1 แสดงบล็อกไดอะแกรมแสดงการทำงานของระบบ

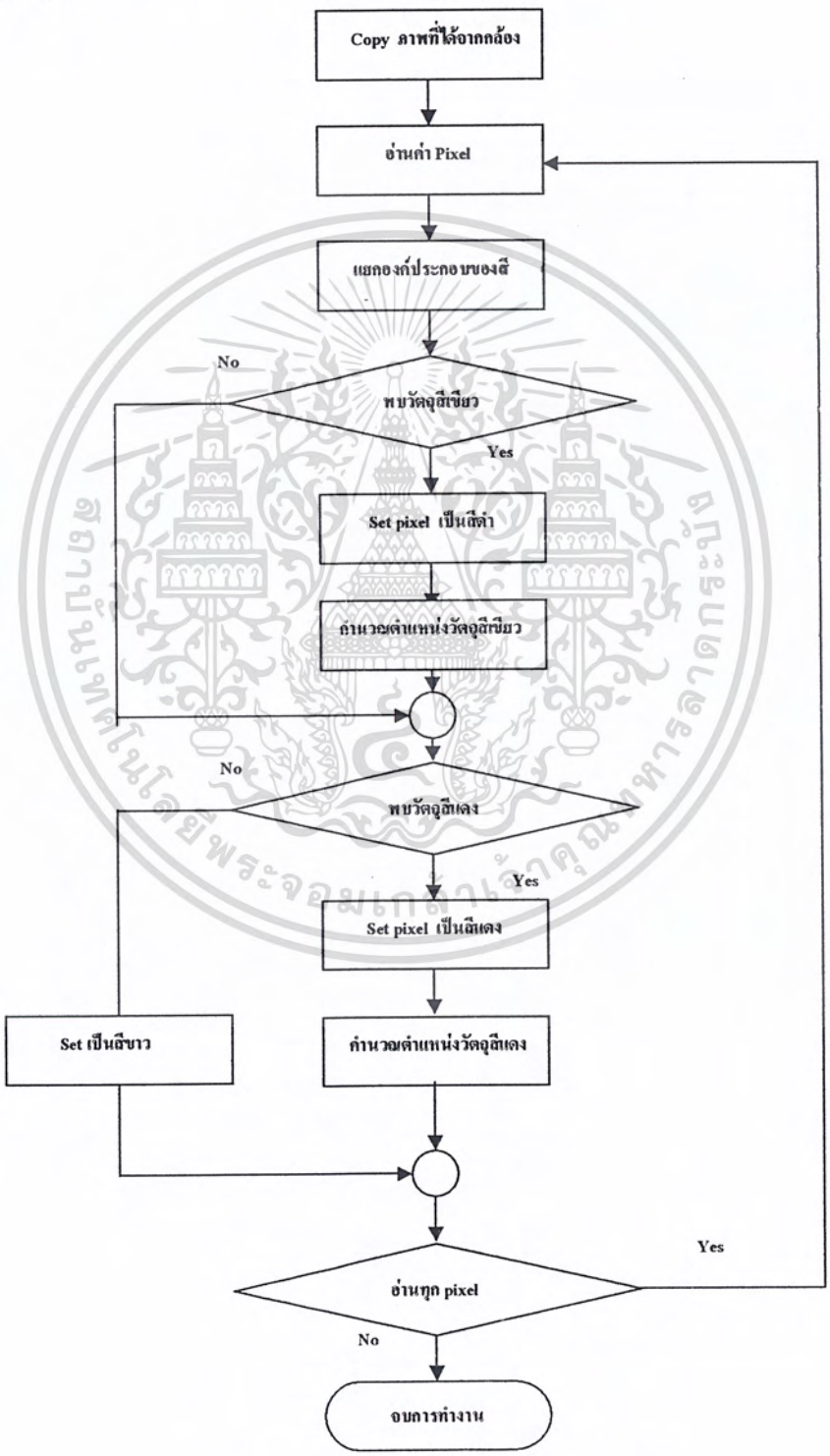
ส่วนประกอบของโครงงานนี้แบ่งออกเป็นส่วนต่างๆดังนี้

1. กล้องวีดีโอสำหรับจับภาพวัตถุและบันทึกภาพเป็นดิจิตอล โดยกล้องวีดีโอเป็นชนิดที่ต่อกับพอร์ต USB ของคอมพิวเตอร์ เพื่อดึงภาพเข้ามาใช้ในการประมวลผลในคอมพิวเตอร์ได้โดยตรง
2. โปรแกรมหาตำแหน่งของวัตถุ โดยใช้โปรแกรม MICROSOFT VISUAL BASIC 6 ทำการวิเคราะห์ภาพเพื่อหาตำแหน่งของวัตถุ โดยการจับเอาสีที่กำหนดไว้เป็นสำคัญเพื่อคำนวณแล้วส่งค่าไปควบคุมชุดขับมอเตอร์เพื่อไปควบคุมแหล่งกำเนิดแสงเลเซอร์ ให้ชี้ไปที่วัตถุและติดตามวัตถุได้
3. สเตปเปอร์มอเตอร์ และวงจรขับสเตปเปอร์มอเตอร์จำนวน 2 ชุด เพื่อให้ชุดขับเคลื่อนแหล่งกำเนิดแสงเลเซอร์ เคลื่อนที่ไปตามตำแหน่งที่ต้องการในแกน (x,y) โดยเราจะใช้คอมพิวเตอร์ทำการคำนวณหาตำแหน่ง และสั่งให้ส่งค่าไปยังวงจรขับสเตปเปอร์มอเตอร์ เพื่อให้ได้ตำแหน่งที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การประมวลผลภาพ

ภาพจากกล้อง จะถูกนำมาประมวลผล เพื่อให้ทราบถึงตำแหน่งของวัตถุขณะนั้น แสดงโดย Flow chart ดังนี้



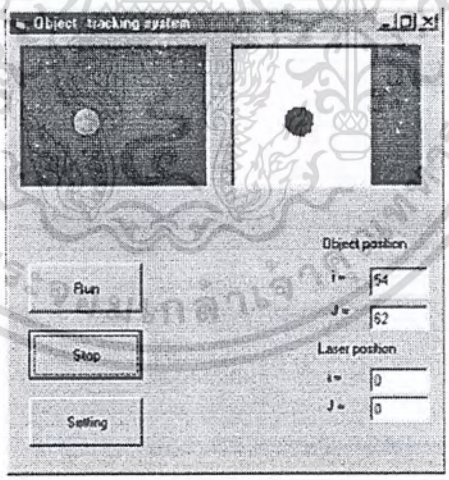
รูปที่ 3.2 แผงผังลำดับการทำงานประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Flow chart สามารถอธิบายรายละเอียดได้ดังนี้

เริ่มด้วยการอ่านภาพจากกล้องมาเก็บไว้ยัง Pixelbox โดยทำการอ่านทั้งภาพ ซึ่งทำได้โดย Bitblt() Function ต่อไปอ่านค่าของพิกเซล โดยอาศัย Getpixel () Function จากนั้นนำค่าที่อ่านได้นี้มาทำการแยกองค์ประกอบของสี แดง เขียว น้ำเงิน โดยอาศัย Color RGB Function จากนั้นตรวจสอบว่าองค์ประกอบสีเขียวมากกว่าองค์ประกอบสีแดง และมากกว่าองค์ประกอบสีน้ำเงินหรือไม่ ถ้ามากกว่าจริงแสดงว่าค่าของพิกเซลที่อ่านได้เป็นวัตถุสีเขียว ให้ทำการกำหนดให้พิกเซลนี้เป็นสีดำแล้วทำการเก็บค่าตำแหน่งที่พบไว้ ค่าของตำแหน่งนี้จะถูกนำออกมาบวกกัน และสุดท้ายจะหารด้วยจำนวนครั้งที่พบเพื่อหาค่าเฉลี่ยของตำแหน่งวัตถุ ถ้าปริมาณของสีเขียว น้อยกว่าสีแดง และสีแดงมากกว่าสีน้ำเงิน แสดงว่าค่าพิกเซล ที่อ่านได้ เป็นวัตถุสีแดงให้ทำการกำหนดพิกเซล ดังกล่าวเป็นสีแดง และเก็บค่าตำแหน่งไว้เช่นกัน

ถ้าไม่เป็นไปตามเงื่อนไขที่ผ่านมาแสดงว่าตำแหน่งดังกล่าวเป็นสีขาวไม่พบวัตถุ ก็ให้กำหนดเป็นสีขาวเช่นเดิม จากนั้นทำเช่นเดียวกัน จนกระทั่งประมวลผลครบทุก พิกเซลซึ่งจะได้ตำแหน่งของวัตถุทั้งสอง คือค่าในแนวแกน x และในแนวแกน y ซึ่งจะได้ผลดังรูป



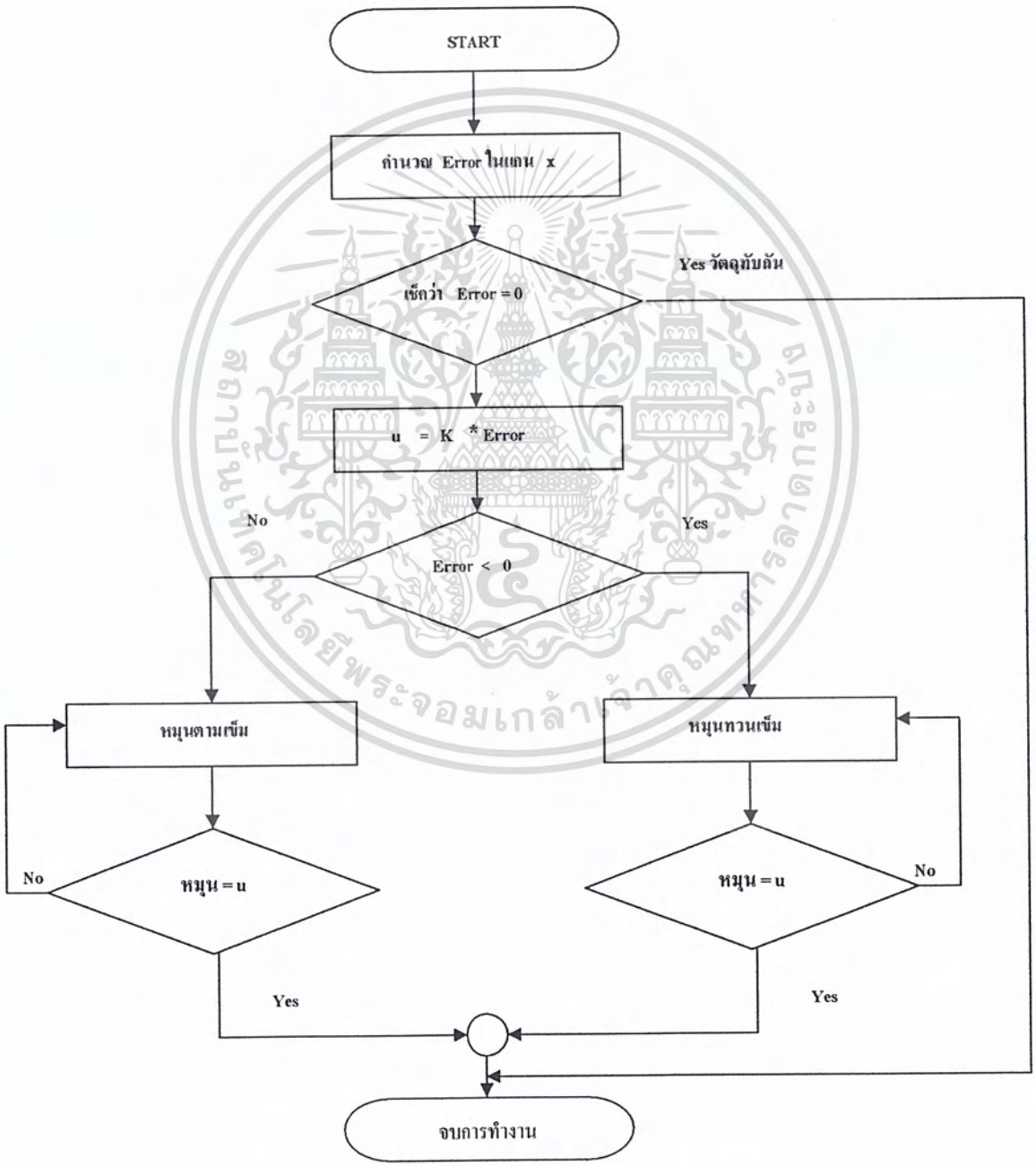
รูปที่ 3.3 การแยกสี

จากวิธีการประมวลผลภาพที่ผ่านมา พบว่าทุก ๆครั้งจำเป็นจะต้องประมวลผลครบทุก พิกเซล ซึ่งต้องใช้เวลาค่อนข้างมาก ดังนั้นเราทำการปรับปรุงโดยการคำนวณค่าเริ่มต้น ของการอ่านพิกเซล และค่าสุดท้ายด้วยวิธีนี้ จะทำให้การประมวลผลเฉพาะบริเวณ ที่วัตถุอยู่เท่านั้น กล่าวคือ จะทำการประมวลผลหมดทั้งภาพเฉพาะ ครั้งแรกเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การควบคุมตำแหน่ง

เป้าหมายของการควบคุมตำแหน่งคือทำให้ตำแหน่งของแสงเลเซอร์ (วัตถุสีแดง) ทับกับวัตถุสีเขียว ตำแหน่งของแสง ควบคุมผ่านมอเตอร์ 2 ตัว ซึ่งทำหน้าที่ควบคุมแกน x และ y ตามลำดับ กระบวนการควบคุมตำแหน่งในแกน x สามารถแสดงโดย Flow chart ดังนี้



รูปที่ 3.4 แสดงลำดับการทำงานของการควบคุมตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Flow chart สามารถอธิบายการทำงานได้ดังนี้

การควบคุมตำแหน่งตั้งอยู่บนพื้นฐานของ Error กล่าวคือถ้า Error เท่ากับศูนย์ หมายความว่าวัตถุทั้งสองทับกันพอดี จะไม่มีการปรับตำแหน่งของมอเตอร์ ดังนั้นเริ่มต้น โปรแกรมจะคำนวณ Error ในแนวแกน x และในแนวแกน y ต่อไปตรวจสอบว่า Error เป็นศูนย์หรือไม่ ถ้า Error เท่ากับศูนย์ ให้จบรอบการทำงาน แต่ถ้า Error ไม่เท่ากับศูนย์ นำ Error ของแกน x ไปคูณกับ Gain สำหรับแกน x และนำ Error ของแกน y ไปคูณกับ Gain ของแกน y ทำการตรวจสอบผลที่ได้ว่า Error ที่ได้นี้มากกว่าหรือน้อยกว่าศูนย์ กล่าวคือถ้า Error มากกว่าศูนย์หมายความว่าตำแหน่งปัจจุบันของวัตถุสีแดง (ในแกน x) ยังไม่ถึงวัตถุสีเขียว กรณีนี้จะต้องบังคับมอเตอร์ให้หมุนเพื่อเพิ่มตำแหน่งแกน x ตามผลของค่า k * error ในทางกลับกันถ้า Error น้อยกว่าศูนย์นี้หมายความว่า ตำแหน่งปัจจุบันของ Laser เลยตำแหน่งของวัตถุสีเขียวไป จำเป็นต้องควบคุมให้ลดตำแหน่งของ Laser ลงมาตามจำนวนของ k * error

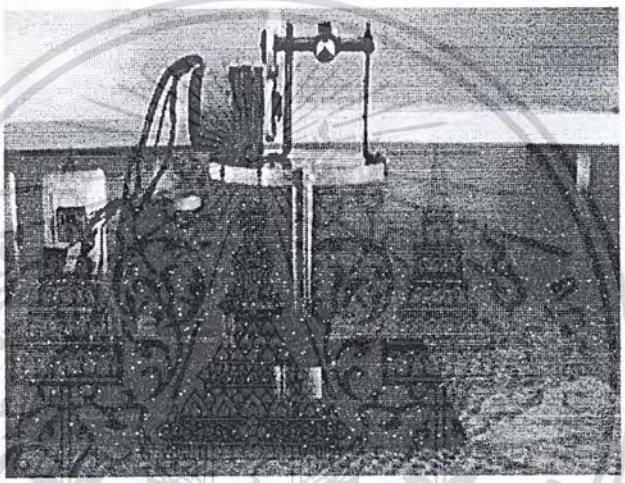
กรณีของการควบคุมแกน y จะทำเหมือนทำการปรับแกน x ลำดับการควบคุมแกน y ก็เป็นเช่นเดียวกับการควบคุมแกน x ทุกประการ

หลังจากปรับแกน x และแกน y เรียบร้อยแล้วก็เสร็จสิ้นหนึ่งรอบการทำงาน ลำดับต่อไปก็จะกลับไปประมวลผลภาพตำแหน่งที่ได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โครงสร้างทางกลศาสตร์

ระบบติดตามวัตถุ จำเป็นที่จะต้องมีเครื่องมือกลที่จะคอยเคลื่อนที่ตำแหน่งของลำแสงเลเซอร์ ให้เคลื่อนที่ตามวัตถุในระนาบ 2 มิติ การออกแบบต้องเคลื่อนที่ได้ใน 2 แกน และใช้สเตปเปอร์มอเตอร์เป็นตัวขับ เพื่อให้ตำแหน่งของแสงเลเซอร์เปลี่ยนแปลง ดังรูปที่ 3.5 และ 3.6 แสดงโครงสร้างทั้งหมดของเครื่องมือกลชี้ตำแหน่ง

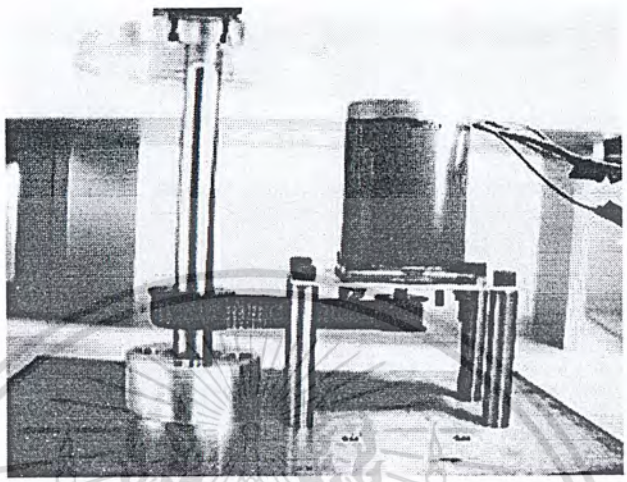


รูปที่ 3.5 แสดงเครื่องมือกลชี้ตำแหน่ง

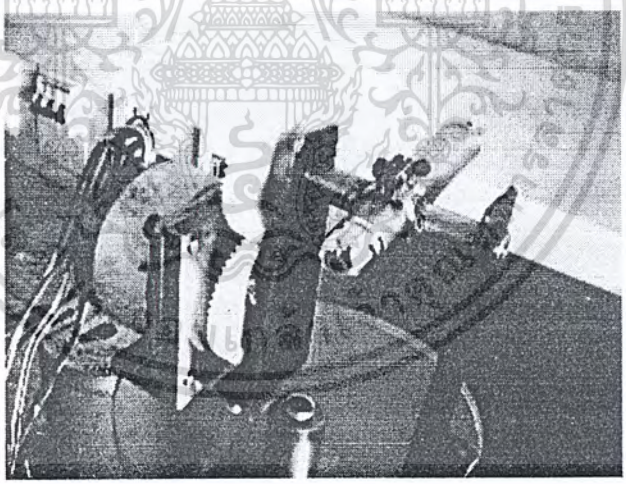
ระบบติดตามวัตถุต้องการเครื่องมือกลที่สามารถเคลื่อนที่ได้อย่างละเอียดในทุกตำแหน่ง และยังต้องการความถูกต้องสูง การออกแบบได้มีการนำสายพานเป็นตัวส่งแรงขับให้มีความละเอียดที่สูงในแนวแกน x และโครงสร้างของฐานเลือกให้ใช้เหล็กเพื่อการรับน้ำหนักของเครื่องมือทั้งหมด ส่วนด้านบนติดตั้งชุดเครื่องมือควบคุมการเคลื่อนที่ในแนวแกน y เลือกใช้ฐานเป็นพลาสติกอะครีลิคเพื่อให้เบาและง่ายต่อการขับ จากการทดลองทำการขับสเตปเปอร์มอเตอร์ พบว่าได้ความละเอียดในแนวแกน x สูง และในแนวแกน y อยู่ในเกณฑ์ที่พอใช้ แต่จากการออกแบบเครื่องมือกลในแนวคิดนี้ประสบปัญหาในเรื่องตำแหน่งแต่ละสเตปของเลเซอร์ จะมีการเคลื่อนที่ที่ไม่เท่ากัน อันเนื่องมาจากมุมในการเคลื่อนที่ของเลเซอร์ ถ้าระยะที่เลเซอร์ชี้ตำแหน่งมีระยะทางที่ไกล แต่ละสเตปจะมีระยะทางการเคลื่อนที่ที่ต่างกันมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 การติดตั้งและการส่งกำลังขับ



รูปที่ 3.6 การขับเพื่อการเคลื่อนที่ตำแหน่งไปในแนวแกน x



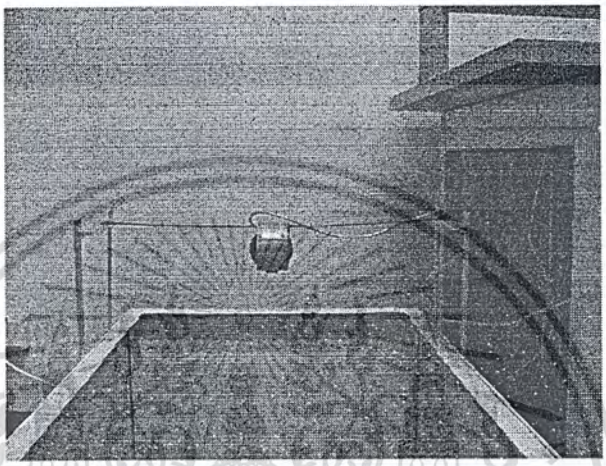
รูปที่ 3.7 แสดงการขับเพื่อการเคลื่อนที่ไปยังแนวแกน y

การติดตั้งมอเตอร์ในตำแหน่งที่ขับในแนวแกน x จะใช้สายพานเป็นตัวส่งกำลังซึ่งจะมีความละเอียดสูง และมอเตอร์ที่ใช้จำเป็นต้องมีแรงบิดที่สูงเพราะต้องนำไปขับส่วนของเครื่องมือชี้ตำแหน่งในแนวแกน y ทั้งนี้ก็จะต้องออกแบบส่วนของเครื่องกลการชี้ตำแหน่ง ในแนวแกน y ให้ฐานมีน้ำหนักที่เบาด้วย จึงเลือกที่จะใช้พลาสติกอะครีลิคแทนเหล็ก ซึ่งจะมีน้ำหนักที่เบาและไม่เป็นภาระในการขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การติดตั้งกล้องดิจิตอลวีดีโอ

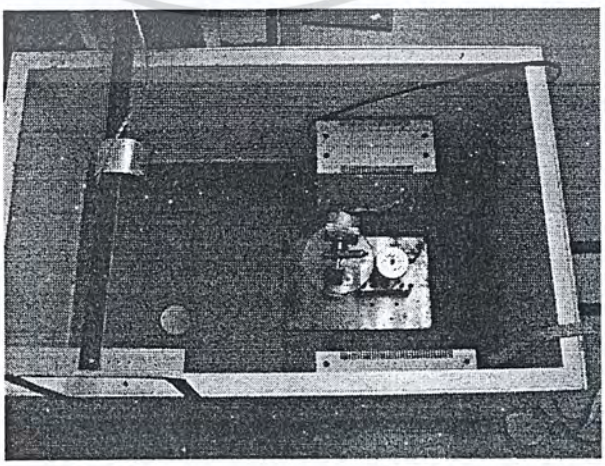
ติดตั้งกล้องให้มีความสูงจากพื้น ในการทดลองพบว่าปรับให้สูงราว 50 เซนติเมตรจะสามารถครอบคลุมเนื้อที่ที่ใช้ทดลองได้ กว้าง 50 เซนติเมตร ยาว 40 เซนติเมตร



รูปที่ 3.8 แสดงการติดตั้งกล้องดิจิตอลวีดีโอ

3.4.3 การติดตั้งเครื่องมือทั้งหมดลงกล่องพร้อมทดลอง

การติดตั้งทั้งหมดของเครื่องมือชี้ตำแหน่งพร้อมทำการทดลอง แสดงดังรูปที่ 3.9 ทางด้านซ้ายมือ ฐานสีน้ำเงินเป็นพื้นที่ที่ใช้ทดลองการเคลื่อนที่ของวัตถุ และทางด้านขวามือเป็นส่วนของการติดตั้งชุดเครื่องมือกลชี้ตำแหน่ง



รูปที่ 3.9 แสดงการติดตั้งเครื่องมือทั้งหมดลงกล่องพร้อมทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

เมื่อเราได้ทำการติดตั้งฮาร์ดแวร์ และทำการเชื่อมต่อพอร์ทขนานเข้ากับคอมพิวเตอร์แล้ว ทำการรันโปรแกรม ซอร์ฟแวร์ต่าง ๆ ที่เขียนขึ้นจึงทำการทดลองและแสดงผลการทดลองที่ Control Form ของโปรแกรม

4.1 การรับสัญญาณภาพจากกล้องดิจิตอลวีดีโอ

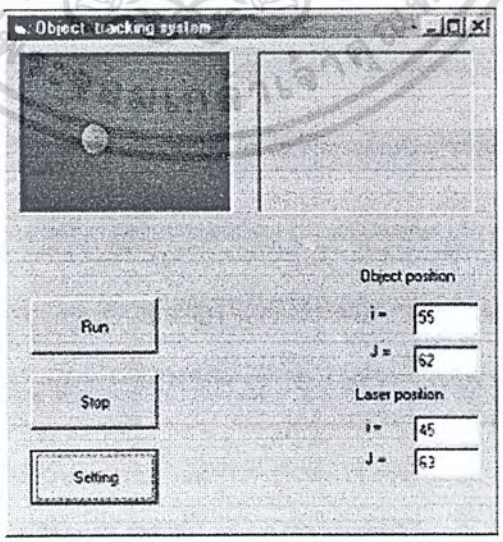
การดึงสัญญาณภาพจากกล้องวีดีโอ เข้ามาประมวลผลในโปรแกรม MICROSOFT VISUAL BASIC และแสดงผลออกทางหน้าจอของคอมพิวเตอร์

การทดลอง

ทำการต่อกล้องวีดีโอ เข้ากับ พอร์ต USB ของคอมพิวเตอร์ แล้วถ่ายภาพวัตถุให้นำภาพเข้ามาประมวลผลในโปรแกรม MICROSOFT VISUAL BASIC โดยตรง โดยเราทำการกำหนดให้พื้นเป็นสีน้ำเงิน และวัตถุเป็นสีเขียว

ผลการทดลอง

สามารถรับภาพ และแสดงผลทางหน้าจอคอมพิวเตอร์ ทางด้านซ้ายของโปรแกรมดังรูปที่ 4.1



รูปที่ 4.1 การดึงภาพเข้ามาประมวลผล

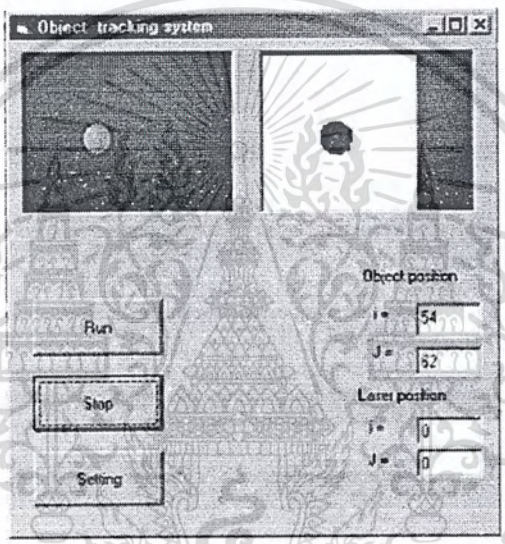
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การแปลงสีภาพของวัตถุ

การทดลอง

นำภาพที่ถ่ายด้วยกล้องดิจิตอลวีดีโอ เข้ามาทำการประมวลผลภาพในโปรแกรม MICROSOFT VISUAL BASIC ที่ได้เขียนขึ้น โดยการแปลงภาพสีให้กลายเป็นสีขาว- ดำ

ผลการทดลอง



รูปที่4.2 การประมวลผลภาพ

จากผลการทดลองที่ได้ในรูปที่ 4.2 แสดงภาพที่ถ่ายด้วยกล้องดิจิตอลวีดีโอซึ่งในภาพทางด้านซ้ายมือจะเห็นวัตถุเป็นสีเขียว จะทำการแปลงให้เป็นสีดำ และพื้นที่ ๆมีความเข้มเป็นสีน้ำเงิน จะทำการแปลงให้เป็นสีขาว และจะปรากฏทางด้านขวามือ

เนื่องจากในการทดลองบางมีข้อจำกัดหลายข้อ และตัวรบกวนทางแสงภายนอกจะส่งผลอย่างมากทำให้การประมวลผลภาพผิดเพี้ยนไป จึงต้องปรับแก้โปรแกรมบางส่วน เพื่อลดผลของแสงและเงาต่าง ๆของวัตถุ

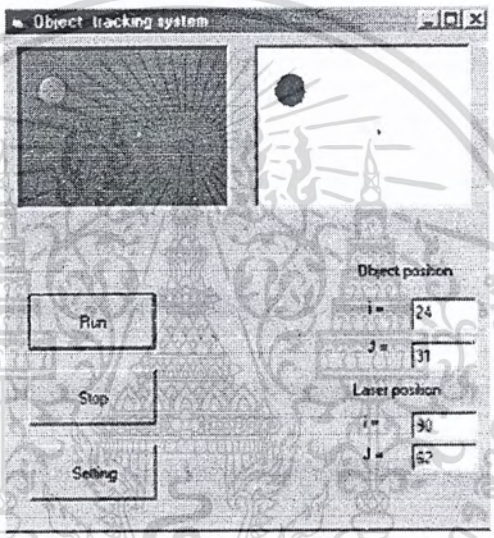
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การประมวลผลภาพของวัตถุและแสงเลเซอร์

การทดลอง

ทำการเปิดสวิตช์ตัวส่งแสงหรือเลเซอร์พอย์นเตอร์ ที่จะให้ชี้ตำแหน่งและติดตามวัตถุ เข้ามาประมวลผลภาพพร้อมกับวัตถุ

ผลการทดลอง



รูปที่ 4.3 ประมวลผลภาพ วัตถุ พร้อม เลเซอร์พอย์นเตอร์

จากผลการทดลองดังรูปที่ 4.3 แสดงการประมวลผลภาพของแสงเลเซอร์พร้อมกับวัตถุ ภาพทางด้านซ้ายมือวัตถุจะเป็นสีเขียว และแสงเลเซอร์เป็นสีแดง จากโปรแกรมจะคำนวณและแยกสีของภาพที่ได้จากด้านซ้ายมือ จะถูกแปลงให้ตัวแสงเลเซอร์เป็นสีแดงตามความเข้มสีที่กำหนดไว้ และวัตถุเป็นสีดำ ส่วนพื้นที่สีน้ำเงินจะถูกแปลงเป็นสีขาว และแสดงผลออกทางหน้าจอของคอมพิวเตอร์ทางด้านขวามือ

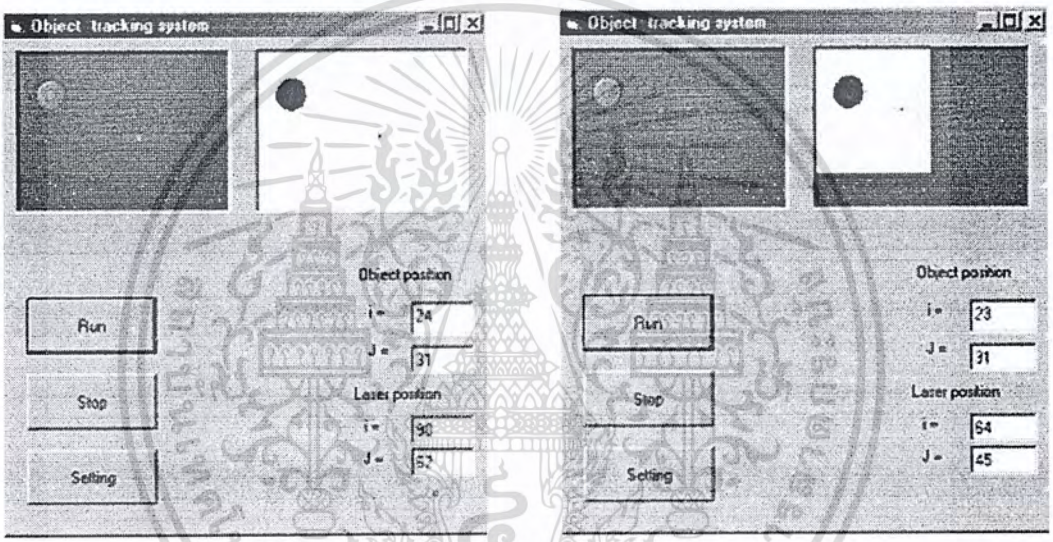
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองโปรแกรมการติดตามวัตถุ

การทดลอง

โดยจะทำการประมวลผลในขั้นตอน 4.1 ถึง 4.3 ในแต่ละลูปและทำการควบคุมสเตปเปอร์มอเตอร์ตามแนวคิดในหัวข้อ 3.3

ผลการทดลอง



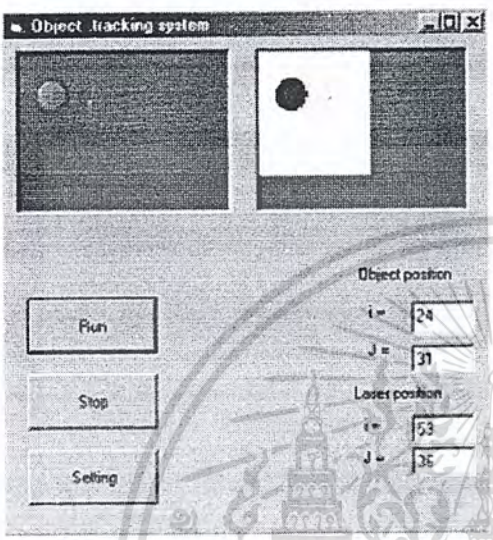
รูปที่ 4.4 ก. **รูปที่ 4.4 ข.**

เมื่อทำการรันโปรแกรมการประมวลผลภาพใน loop โปรแกรมแรก จะได้ผลการทดลองเป็นดังรูปที่ 4.4 ก. โดยที่โปรแกรมจะทำการวิเคราะห์หาตำแหน่ง วัตถุ ว่าอยู่ที่ตำแหน่ง I = 24 J = 31 และตำแหน่งของเลเซอร์พอย์นเตอร์ อยู่ที่ I = 90, J = 62 (โดยการสแกนตำแหน่งเริ่มจากมุมซ้ายบน คือตำแหน่ง I = 0, J = 0)

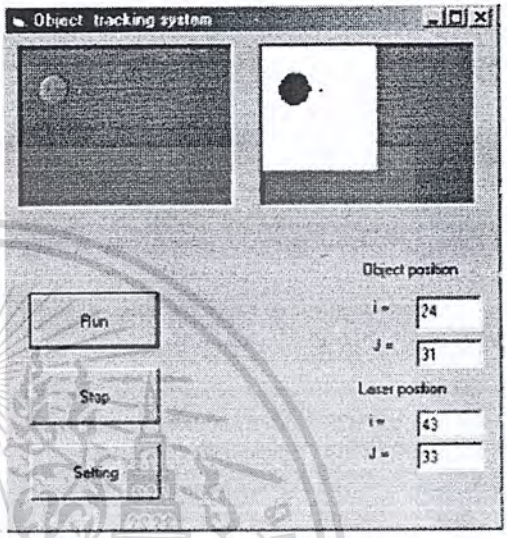
หลังจากการปรับตำแหน่งหนึ่งลูปโดยการสัญญาณควบคุมไปขับสเตปเปอร์มอเตอร์ จะได้ผลการทดลอง ดังแสดงในรูป 4.4 ข. ซึ่งตำแหน่งของเลเซอร์พอย์นเตอร์ จะเปลี่ยนเป็น I = 64 , J = 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

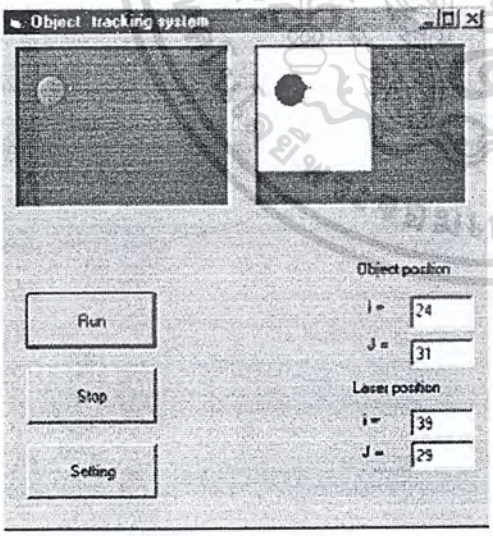
ต่อไปโปรแกรมจะทำการประมวลผลภาพที่ได้และทำการปรับตำแหน่ง ในลูปต่อมา จะได้ผลดังนี้



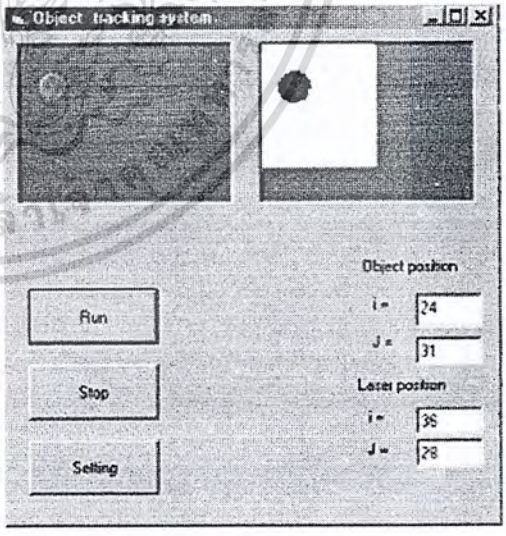
รูปที่ 4.4 ค.



รูปที่ 4.4 ง.

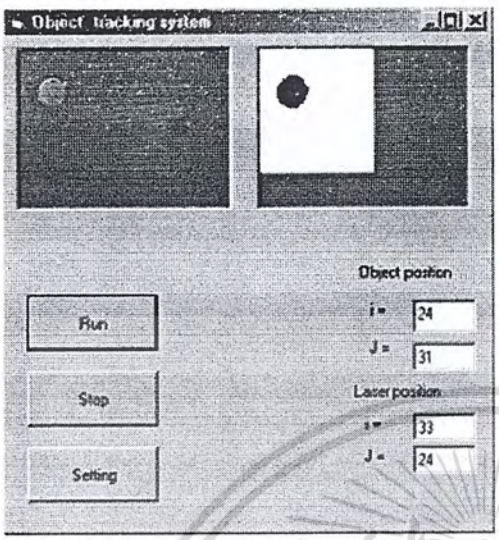


รูปที่ 4.4 จ.

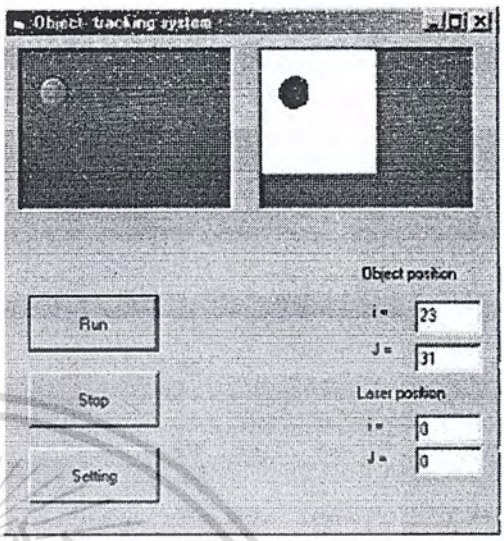


รูปที่ 4.4 ฉ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ช.

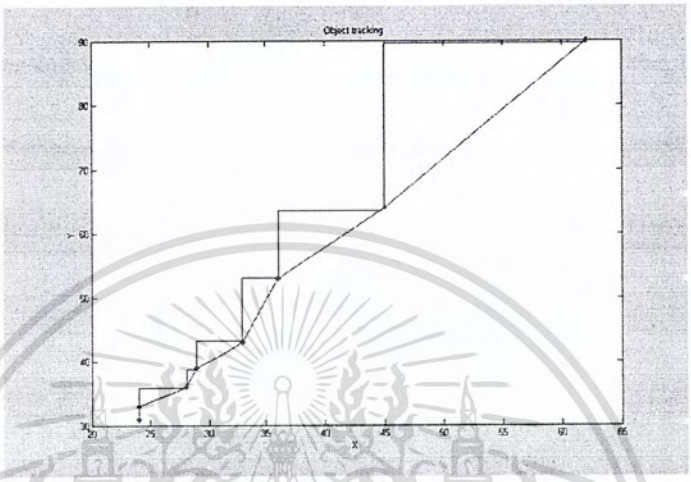


รูปที่ 4.4 ซ.

จากการประมวลผลภาพ และทำการควบคุมตามโปรแกรมที่ได้เขียนขึ้น ในแต่ละลูปควบคุมจะได้ผลเรียงตามรูป ที่ได้แสดงไว้ข้างต้น ตามลำดับรูปที่ 4.4 ก. - ซ. ซึ่งใน รูปที่ 4.4 ซ. จะพบว่าเลเซอร์ชี้ตำแหน่งลงไปทับกับวัตถุได้พอดี โดยโปรแกรมจะตั้งให้ค่านี้แสดงตำแหน่งเลเซอร์เป็น I = 0 , J = 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองข้างต้นสามารถแสดงการเคลื่อนที่ ในรูปแบบกราฟ ดังนี้

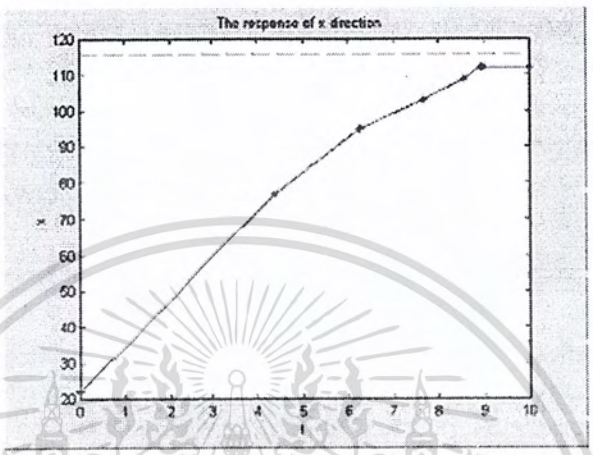


รูปที่ 4.5 กราฟแสดงลักษณะการเคลื่อนที่ของเลเซอร์พอยน์เตอร์ ในแนวแกน x และ y

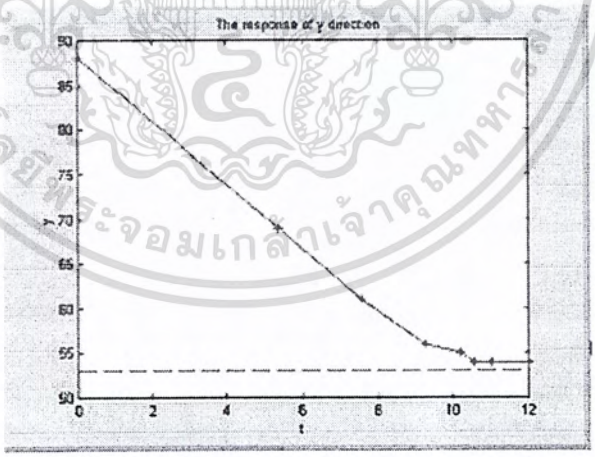
จากรูปที่ 4.4 ก. วัตถุอยู่ที่ตำแหน่ง $x = 24$ และ $y = 31$ เลเซอร์อยู่ที่ $x = 62$ และ $y = 90$ ในลำดับแรก เลเซอร์จะถูกปรับในแนวแกน x และตามด้วยแกน y ซึ่งทำให้ตำแหน่งของเลเซอร์เปลี่ยนแปลงไปเป็น $x = 45$ และ $y = 64$ จากนั้นจะทำการประมวลผลภาพแล้ว ทำการปรับตำแหน่งเช่นนี้ จนกระทั่ง $x = 24$ และ $y = 31$ ซึ่งเป็นตำแหน่งเป้าหมายซึ่งวัตถุอยู่ เป็นการจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟแสดงการเคลื่อนที่ของแกนเปรียบเทียบกับเวลา



รูปที่ 4.6 กราฟแสดงการเคลื่อนที่ของแกน x เปรียบเทียบกับเวลา (วินาที)



รูปที่ 4.7 กราฟแสดงการเคลื่อนที่ของแกน y เปรียบเทียบกับเวลา (วินาที)

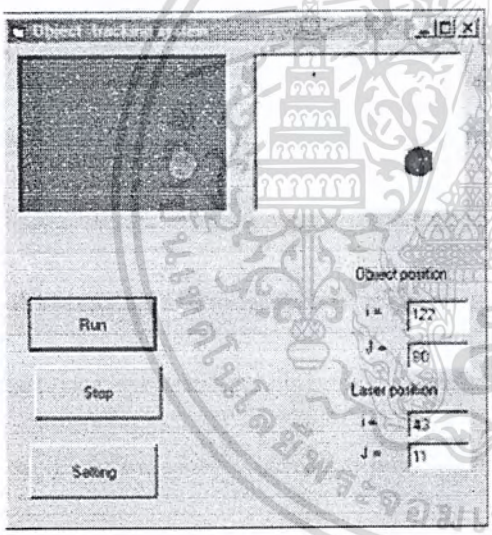
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองโปรแกรมการติดตามวัตถุ (ในแนวแกน x และแกน y พร้อมกัน)

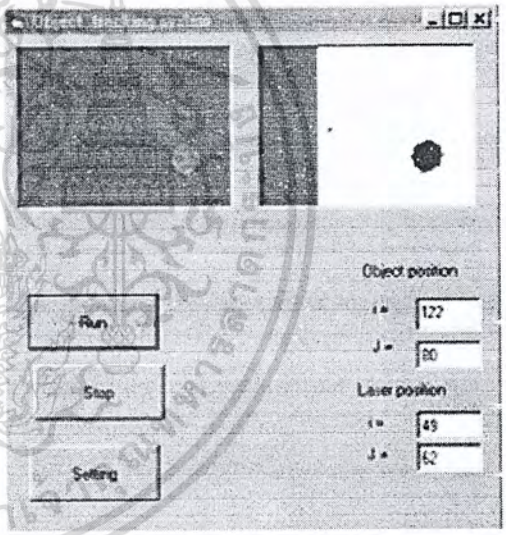
เนื่องจากการเขียนโปรแกรมประมวลผลภาพ และแนวคิดในการส่งค่าการเคลื่อนที่ไปขับสเตปเปอร์มอเตอร์ให้เคลื่อนที่ไปทีละแกนนั้น จะกินเวลาในการทำงานมาก จึงได้ทดลองปรับเปลี่ยนโปรแกรมในการขับสเตปเปอร์มอเตอร์เสียใหม่ ให้ทำการประมวลผล และส่งค่าไปขับมอเตอร์ ในแนวแกน x และ แกน y พร้อมกัน

ผลการทดลอง

เมื่อทำการเปลี่ยนโปรแกรมการเคลื่อนที่ของแนวแกน x และแกน y พร้อมกัน จะได้ผลการทดลอง ดังในรูปที่ เป็นการประมวลผลภาพในลูปโปรแกรมแรก ก่อนจะส่งค่าตำแหน่งไปขับสเตปเปอร์มอเตอร์



รูปที่ 4.8 ก.

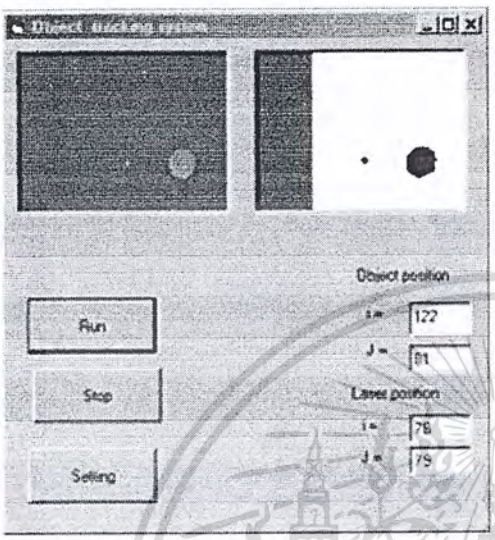


รูปที่ 4.8 ข.

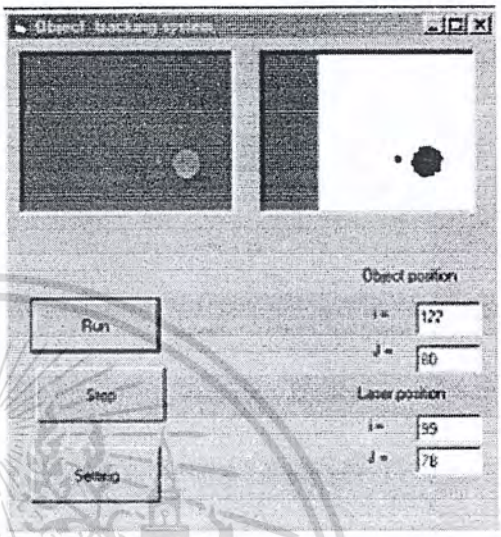
จากรูป 4.8 ก. โปรแกรมจะทำการวิเคราะห์หาตำแหน่งวัตถุ อยู่ที่ตำแหน่ง I = 122 J = 80
และตำแหน่งของเลเซอร์พอย์นเตอร์ อยู่ที่ I = 43 J = 11
หลังจากการปรับตำแหน่งแล้ว โดยทำการปรับทั้งแกน x และ แกน y พร้อมกัน
กตำแหน่งของเลเซอร์พอย์นเตอร์ อยู่ที่ I = 49 J = 62 ดังแสดงในรูปที่ 4.8 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

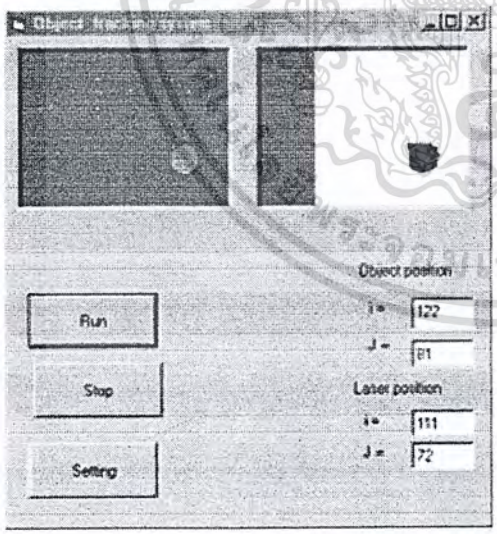
ต่อไปโปรแกรมจะทำการประมวลผลภาพที่ได้และทำการปรับตำแหน่ง ในลูปต่อไป จะได้ผลดังนี้



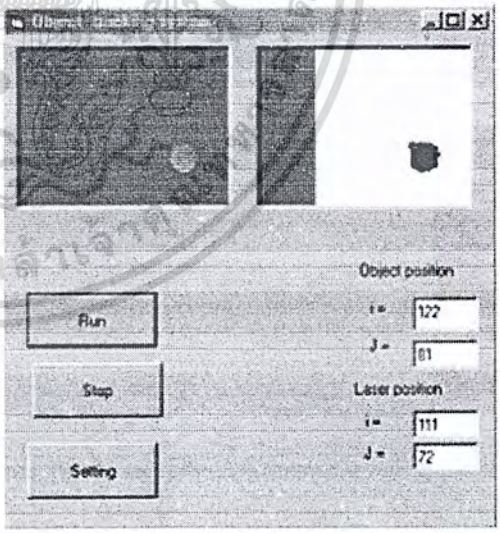
รูปที่ 4.8 ค.



รูปที่ 4.8 ง.

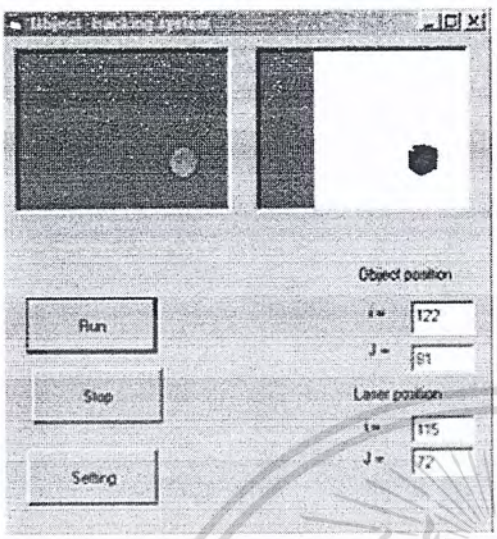


รูปที่ 4.8 จ.

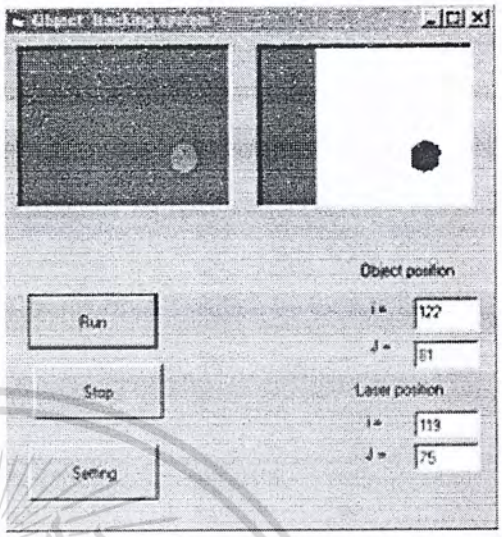


รูปที่ 4.8 ฉ.

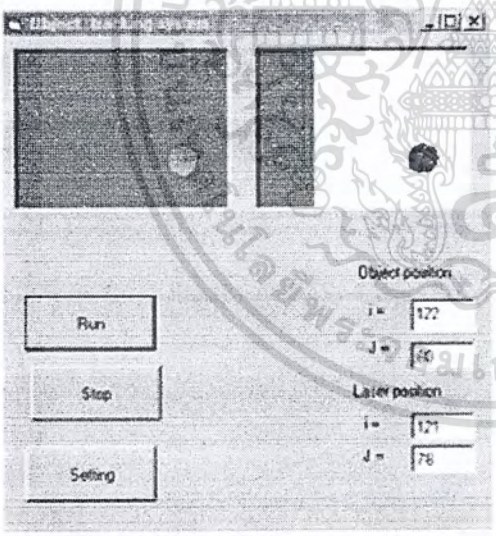
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ช.



รูปที่ 4.8 ซ.

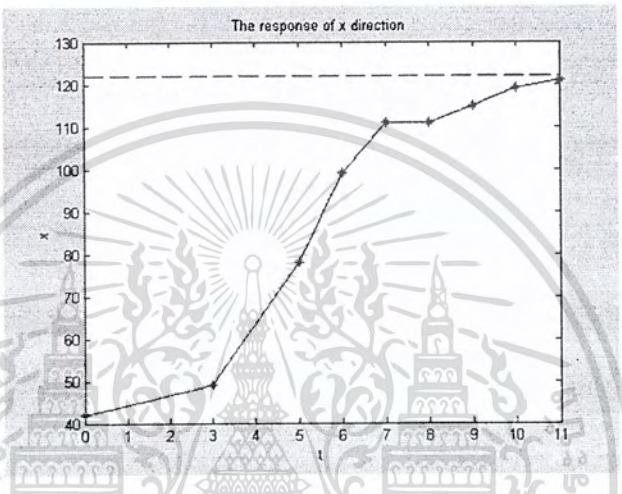


รูปที่ 4.8 ฌ.

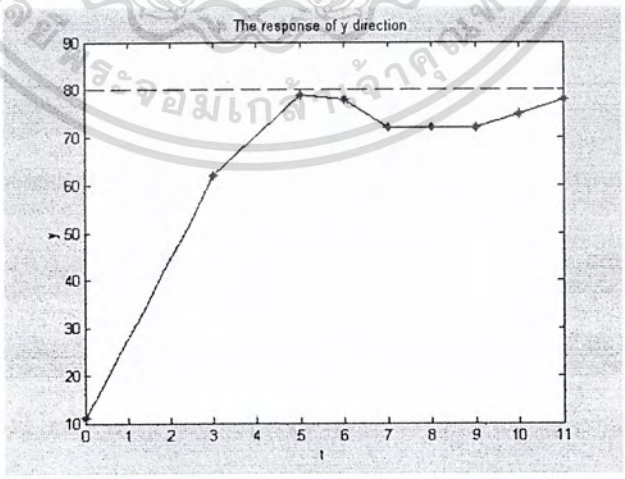
จากการประมวลผลภาพ และทำการควบคุมตามโปรแกรมที่ได้เขียนขึ้น ในแต่ละลูปคอนโทรล จะได้ผลเรียงตามรูป ที่ได้แสดงไว้ข้างต้น ตามลำดับรูปที่ 4.8 ก. - ฌ. ซึ่งใน รูปที่ 4.4 ฌ. จะพบว่าเลเซอร์ชี้ตำแหน่งลงไปทับกับวัตถุได้พอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถแสดงการเคลื่อนที่ของเลเซอร์พอยน์เตอร์ในรูปแบบกราฟได้ดังนี้ เมื่อเราทำการขับสเตปเปอร์มอเตอร์ให้ทำการเคลื่อนที่ไปทั้ง 2 แกนพร้อมกัน ในแต่ละลูปการทำงาน จะได้ผลการการเคลื่อนที่ของแกน x เทียบกับเวลาดังแสดงในรูปที่ 4.9 และ การเคลื่อนที่ของแกน y เทียบกับเวลาดังแสดงในรูปที่ 4.10



รูปที่ 4.9 กราฟแสดงการเคลื่อนที่ของแกน x เปรียบเทียบกับเวลา (วินาที)



รูปที่ 4.10 กราฟแสดงการเคลื่อนที่ของแกน y เปรียบเทียบกับเวลา (วินาที)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 สรุปผลการทดลอง

จากการทดลองคอมพิวเตอร์ สามารถรับภาพจากกล้องดิจิตอลวีดีโอ เข้ามาประมวลผลภาพในโปรแกรม MICROSOFT VISUAL BASIC และแสดงผลออกทางหน้าจอของคอมพิวเตอร์ได้ และการทดลองแปลงสีภาพทั้งภาพของวัตถุ พื้น และแสงเลเซอร์ได้ผลการทดลองที่ถูกต้อง แต่ทั้งนี้ต้องมีการควบคุมในเรื่องของแสงภายในห้องจะส่งผลทำให้การรับภาพจากกล้องดิจิตอลวีดีโอ ได้สีที่ไม่ตรงกับค่าที่ตั้งไว้ ส่งผลให้การประมวลผลภาพผิดพลาดและจะทำให้ไม่สามารถชี้ตำแหน่งของวัตถุได้ถูกต้อง เราอาจปรับแก้โดยการควบคุมแสงสว่างภายในห้องทดลอง หรือทำการปรับเซตค่าพารามิเตอร์ต่าง ๆที่ตัวกล้องทุกครั้งก่อนทดลอง

ในด้านการประมวลผลภาพและส่งสัญญาณควบคุมไปขับสเตปเปอร์มอเตอร์เพื่อให้เลเซอร์เคลื่อนที่นั้นในการทดลองในหัวข้อที่ 4.4 เป็นการขับสเตปเปอร์มอเตอร์ทีละแกน ในลูปการทำงานแรก ใช้เวลาการทำงานที่มาก แต่เมื่อเราทำการปรับจูนค่าของ K ที่เหมาะสมทำให้ระบบมีความรวดเร็วยิ่งขึ้นกว่าเดิม และเมื่อทำการปรับเปลี่ยนโปรแกรมให้มีการขับสเตปมอเตอร์ให้ทำงานพร้อมกัน คือให้เกิดการเคลื่อนที่ทั้งในแนวแกน x และ แกน y ดังการทดลองหัวข้อที่ 4.5 จะช่วยลดเวลาในการทำงานและลดช่วงเวลาประมวลผลลงได้มาก ซึ่งผลการทดลองที่ได้มีความถูกต้อง อยู่ในเกณฑ์ที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากการทดลองการรับภาพจากกล้องดิจิตอลวีดีโอเข้ามาประมวลผลในโปรแกรมวิชชวลเบสิก และแสดงผลออกจากหน้าจอคอมพิวเตอร์ รวมถึงการประมวลผลภาพได้ผลตามที่ต้องการและการทดลองโปรแกรมควบคุมให้แสงเลเซอร์เคลื่อนที่ติดตามวัตถุที่ใช้ระบบควบคุมแบบพี ให้ชี้ตำแหน่งของวัตถุ พบว่าสามารถติดตามวัตถุที่กำลังเคลื่อนที่ได้ แต่ยังไม่สามารถติดตามวัตถุที่เคลื่อนที่ด้วยความเร็วสูงได้

5.2 วิจารณ์การทดลอง

การทำโครงงานนี้จำเป็นจะต้องทำการศึกษาความรู้พื้นฐานต่างๆ หลายแนว ทั้งทางส่วนของเครื่องมือกล และส่วนของโปรแกรมคอมพิวเตอร์ ส่วนสำคัญที่เราทำการศึกษาได้แก่ การประมวลผลภาพดิจิตอล วิธีการรับส่งข้อมูลแบบขนาน การเขียนโปรแกรมวิชชวลเบสิก ในการควบคุมการทำงานทุกส่วน และการใช้งานสเตปเปอร์มอเตอร์

5.3 ประโยชน์ที่ได้รับจากการทำโครงงานนี้

ในการทำโครงงานนี้ต้องอาศัยความรู้ในหลายๆด้าน ทั้งความรู้ในด้าน การเขียนโปรแกรม ความรู้ทางด้านการออกแบบและสร้างเครื่องมือกล ความรู้ในด้านทฤษฎีของระบบควบคุมมาใช้ ความรู้เกี่ยวกับการประมวลผลภาพดิจิตอลเพราะการส่งข้อมูลกลับนั้นต้องอาศัยกล้องดิจิตอลวีดีโอ อีกทั้งยังทำให้เกิดทักษะในด้านการเขียนโปรแกรมภาษาวิชชวลเบสิก และความรู้ทางด้านวงจรอิเล็กทรอนิกส์จากการสร้างวงจรควบคุมสเตปเปอร์มอเตอร์และ เลือกวิธีการขับสเตปเปอร์ให้เหมาะสมกับงานที่ใช้ ประโยชน์อีกด้านของการทำงานนี้คือฝึกการแก้ไขปัญหาต่างๆที่เกิดขึ้น ทั้งปัญหาที่เกิดจาก ปัจจัยภายนอก และปัญหาเกี่ยวกับวงจร หรือโปรแกรม และประโยชน์ทางด้านการประสานงานในการทำงานมีการแบ่งงานกันทำ ทำให้มีการฝึกการวางแผนอย่างเป็นระบบและการคิดแก้ไขปัญหาต่างๆที่เกิดขึ้นร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ปัญหาที่พบในการทำโครงงานนี้

ปัญหาที่ประสบจากการทำโครงงานนี้อย่างแรกคือ ขาดความรู้และประสบการณ์ในการทำงาน เป็นปัจจัยที่สำคัญที่สุดและเป็นอุปสรรคในการทำงานมาก ในช่วงแรกต้องทำการศึกษาเกี่ยวกับการเขียนโปรแกรมภาษาวิชชวลเบสิกและ ทฤษฎีการประมวลผลภาพ เกิดมาจากพื้นฐานทางด้านโปรแกรมมีน้อยและไม่เคยศึกษาทางด้านการประมวลผลภาพมาก่อน ในช่วงแรกจึงเสียเวลาในการศึกษาก่อนการทำโครงงานไปพอสมควร

ปัญหาในด้านเครื่องมือกลชี้ตำแหน่งทำการเลือกใช้อุปกรณ์ที่ไม่ละเอียดพอ เฟืองที่ใช้ควบคุมการเคลื่อนที่ในด้านแนวแกน y มีความละเอียดที่ต่ำเนื่องจากจำนวนฟันเฟืองมีน้อย ปัญหานี้จะส่งผลเป็นอย่างมากเมื่อนำไปใช้กับ วัตถุที่ต้องการติดตามมีขนาดเล็ก การชี้ตำแหน่งอาจจะเกิดการผิดพลาดหรือไม่สามารถชี้ตำแหน่งได้

นอกจากนั้นยังมีปัญหาทางด้านโปรแกรมประมวลผลภาพ ซึ่งเป็นข้อจำกัดของความสามารถของเครื่องมือ คอมพิวเตอร์ใช้เวลาในการประมวลผลภาพค่อนข้างช้า เมื่อเราปรับขนาดเฟรม หรือทำการสแกนพื้นที่ที่มีขนาดกว้าง หรือปรับให้มีความละเอียดสูงนั้น จะกินเวลาในการประมวลผลที่นาน ถ้าวัตถุมีการเคลื่อนที่ที่เร็วจะไม่สามารถติดตามวัตถุได้ทัน

5.5 แนวทางการพัฒนา

ในการทำโครงงานนี้มีข้อผิดพลาดที่ควรแก้คือในด้านของเครื่องมือกลชี้ตำแหน่งในแนวแกน y ที่มีความละเอียดในตำแหน่งต่ำ เราสามารถแก้ไขได้โดยการเลือกใช้เฟืองที่มีฟันเฟืองที่ละเอียดขึ้น หรือถ้าจะเลือกใช้การส่งกำลังขับของสายพานเหมือนกับการเคลื่อนที่ในแนวแกน x และการแก้ไขในด้านสเตปเปอร์มอเตอร์อาจจะเลือกใช้การขับแบบนาโนสเตป ซึ่งจะทำให้มีความละเอียดที่สูงขึ้น ซึ่งเราสามารถเลือกใช้ได้ตามความเหมาะสม และการทางด้านการพัฒนาในส่วนของโปรแกรมให้ ทำการประมวลผล และควบคุมแสงเลเซอร์ให้สามารถติดตามเป้าหมายได้เร็วยิ่งขึ้น เพื่อนำไปติดตามวัตถุที่มีความเร็วสูงได้ และเขียนโปรแกรมทำการเปรียบเทียบคุณสมบัติของระบบควบคุมแบบต่าง ๆ เพื่อหาข้อดีข้อเสียของแต่ละระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. สิทธิศักดิ์ มณีแสง, "การควบคุมตำแหน่งลูกบอลบนคาน" ปริญญานิพนธ์ปีการศึกษา 2544, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2545
2. กิตติ ภักดีวัฒนะกุล , จำลอง ครูอุตสาหะ , "วิชชวลเบสิก 6 ฉบับโปรแกรมเมอร์," หจก. ไทยเจริญการพิมพ์
3. อรรถพล บุญยะโภคา , ชัยวัฒน์ ลิ้มวิจิตรวิไล , "เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผานพอร์ตขนาน," บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, พิมพ์ครั้งที่ 2
4. ฉัททวุฒิ พีชผล , พิชิต สันติกุลานนท์, "คู่มือเรียน Visual Basic 6," โปรวิชั่น , 2544, พิมพ์ครั้งที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ก -1 โปรแกรมการประมวลผลภาพ

```
Private x As Integer
 Private Function colorRGB(color1 As Long, r As Integer, g As Integer, b As Integer)
 r = color1 Mod 256
 b = color1 \ 65536
 g = (color1 \ 256) Mod 256
 End Function
Private Sub CmdRun_Click()
 Dim i, j, n As Integer
 Dim i_start, i_stop, j_start, j_stop, dgr%, dgb%, r%, g%, b As Integer
 Dim posi_green_i, posi_green_j, count_green As Integer
 Dim posi_red_i, posi_red_j, count_red As Integer
 Dim color1 As Long

 'Variables for the stepping motors
 Dim Gain_x, Gain_y, count_mo, index_x, index_y, error_i, error_j As Integer

 'Array variables are used for containing the value utilized to trig each phase
 Dim X_axis(8) As Integer

X_axis(1) = 1
X_axis(2) = 3
X_axis(3) = 2
X_axis(4) = 6
X_axis(5) = 4
X_axis(6) = 12
X_axis(7) = 8
X_axis(8) = 9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim Y_axis(8) As Integer
 Y_axis(1) = &H10
Y_axis(2) = &H30
Y_axis(3) = &H20
Y_axis(4) = &H60
 Y_axis(5) = &H40
Y_axis(6) = &HC0
Y_axis(7) = &H80
Y_axis(8) = &H90
 count_green = 0          'initial value
 count_red = 0


 x = 1
 i_start = 0
 i_stop = 180
 j_start = 0
 j_stop = 120
 Do
 DoEvents
 'The position becomes zero because the following variabes are set to be zero
 posi_green_j = 0
 posi_green_i = 0
 posi_red_j = 0
 posi_red_i = 0
 BitBlt Picture2.hdc, 0, 0, 320, 240, Picture1.hdc, 0, 0, vbSrcCopy
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
For i = i_start To i_stop
  For j = j_start To j_stop
    color1 = GetPixel(Picture2.hdc, i, j)
    Call colorRGB(color1, r, g, b)
    dgr = g - r
    dgb = g - b

    If dgr > 20 And dgb > 20 Then
      color1 = 0

      'This loop is utilized for finding the position of a green object
      'If count_green < 10 Then
       ' posi_green_i = i
        ' posi_green_j = j
         'count_green = count_green + 1

      'End If
   count_green = count_green + 1
   posi_green_i = posi_green_i + i
   posi_green_j = posi_green_j + j

    ElseIf (r + b + g) > 600 Or (r - g > 20) Then

      color1 = 255
      'This loop is used for locating the existence of a red object

      If count_red < 2 Then
        posi_red_i = i
        posi_red_j = j
        count_red = count_red + 1

      End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Else

      color1 = 16777215
    End If
    SetPixel Picture2.hdc, i, j, color1

  Next j

    DoEvents
 Next i

'If we do not set count_green and count_red = 0,a loop for finding the  i and j position  can not be
executed
' we add
posi_green_i = posi_green_i \ count_green
posi_green_j = posi_green_j \ count_green
count_green = 0
count_red = 0

'Set the value of Gain_x
Gain_x = 1.5

'program for calculating error of both  x axis and  y axis
error_i = posi_green_i - posi_red_i
error_j = posi_green_j - posi_red_j
Text1.Text = posi_green_i
Text2.Text = posi_red_i
Text3.Text = posi_green_j
Text4.Text = posi_red_j
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If (posi_red_i = 0) Or (posi_green_i = 0) Then
  error_i = 0
End If
If (posi_red_j = 0) Or (posi_green_j = 0) Then
  error_j = 0
End If
'We improve the interval of scaning
i_start = posi_green_i - 60
i_stop = posi_green_i + 60
j_start = posi_green_j - 60
j_stop = posi_green_j + 60
```

ก- 2 โปรแกรมระบบควบคุม การเคลื่อนที่

```
'Program for driving a stepping motor of x axis

'if error_i is less than 0,we assume that both objects coincide

  If error_i < 0 Then

    For count_mo = 1 To (Abs(error_i) \ Gain_x)

      If (index_x >= 9 Or index_x < 1) Then

        index_x = 1
      End If

      Out &H378, X_axis(index_x)
      index_x = index_x + 1
      Call delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Next
    index_x = index_x - 1   'Since index_x is added by one in the peceding loop,
                            'it must be set to the current value
  End If
  ' This means that error_i is less than zero
  ' This part of the program reveres the direction of the stepping motor
    If error_i > 0 Then
    For count_mo = 1 To (Abs(error_i) \ Gain_x)

      If (index_x <= 0) Or (index_x > 8) Then

        index_x = 8
      End If

      Out &H378, X_axis(index_x)
      index_x = index_x - 1
      Call delay

    Next

    index_x = index_x + 1  'Since index_x is subtracted by one in the previous step,
                           'it must be set to hold the final value
  End If
 'y_axis
Gain_y = 3
If error_j < 0 Then

    For count_mo = 1 To (Abs(error_j) \ Gain_y)

      If (index_y >= 9 Or index_y < 1) Then

        index_y = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End If


        Out &H378, Y_axis(index_y)
        index_y = index_y + 1
        Call delay
      Next
      index_y = index_y - 1   'Since index_x is added by one in the peceding loop,
                              'it must be set to the current value
   End If
' This means that error_i is less than zero
   ' This part of the program reveres the direction of the stepping motor
      If error_j > 0 Then
      For count_mo = 1 To (Abs(error_j) \ Gain_y)


        If (index_y <= 0) Or (index_y > 8) Then


          index_y = 8
        End If


       Out &H378, Y_axis(index_y)
        index_y = index_y - 1
        Call delay


      Next


      index_y = index_y + 1  'Since index_x is subtracted by one in the previous step,
                             'it must be set to hold the final value
   End If
Loop While (x = 1)


End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Command2_Click()
ezVidCap1.ShowDlgVideoSource
End Sub

Private Sub Command3_Click()
x = 0
End Sub
Sub delay()
Times = Timer
Do
DoEvents
Loop Until Timer >= Times + 0.01
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

BD135,-6,-10,-16
BD137,-6,-10,-16
BD139,-6,-10,-16

PLASTIC MEDIUM POWER
SILICON NPN TRANSISTOR

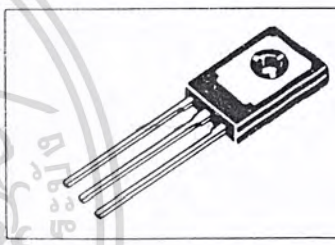
. . . designed for use as audio amplifiers and drivers utilizing complementary or quasi complementary circuits.

- Available in H_{FE} groups -6, -10, -16
- DC Current Gain—h_{FE} = 40 (Min) @ I_C = 0.15 Adc
- BD 135, 137, 139 are complementary with BD 136, 138, 140

1.5 AMPERE
POWER TRANSISTOR

NPN SILICON

45, 60, 80 VOLTS
10 WATTS



MAXIMUM RATINGS

Rating	Symbol	Type	Value	Unit
Collector-Emitter Voltage	V_{CEO}	BD 135 BD 137 BD 139	45 60 80	Vdc
Collector-Base Voltage	V_{CBO}	BD 135 BD 137 BD 139	45 60 100	Vdc
Emitter-Base Voltage	V_{EBO}		5	Vdc
Collector Current	I_C		1.5	Adc
Base Current	I_B		0.5	Adc
Total Device Dissipation @ T_A = 25°C Derate above 25°C	P_D		1.25 10	Watts mW/°C
Total Device Dissipation @ T_C = 25°C Derate above 25°C	P_D		12.5 100	Watt mW/°C
Operating and Storage Junction Temperature Range	T_J, T_{stg}		-55 to +150	°C

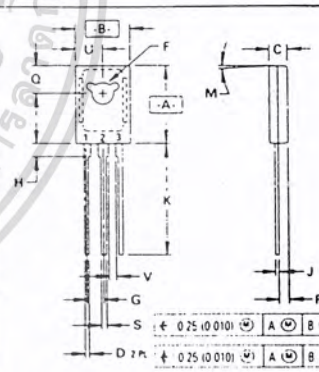
THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction to Case	θ_{JC}	10	°C/W
Thermal Resistance, Junction to Ambient	θ_{JA}	100	°C/W

ELECTRICAL CHARACTERISTICS (T_C = 25°C unless otherwise noted)

Characteristic	Symbol	Type	Min	Max	Unit
Collector-Emitter Sustaining Voltage* (I_C = 0.03 Adc, I_B = 0)	BV_{CEO}*	BD 135 BD 137 BD 139	45 60 80	— — —	Vdc
Collector Cutoff Current (V_{CB} = 30 Vdc, I_E = 0) (V_{CB} = 30 Vdc, I_E = 0, T_C = 125° C)	I_{CBO}		— — —	0.1 10	μAdc
Emitter Cutoff Current (V_{BE} = 5.0 Vdc, I_C = 0)	I_{EBO}		—	10	μAdc
DC Current Gain (I_C = 0.005A, V_{CE} = 2 V) (I_C = 0.15A, V_{CE} = 2 V) (I_C = 0.5A V_{CE} = 2 V)	h_{FE}*		25 40 25	— 250 —	—
Collector-Emitter Saturation Voltage* (I_C = 0.5 Adc, I_B = 0.05 Adc)	$V_{CE(sat)}$*		—	0.5	Vdc
Base-Emitter On Voltage* (I_C = 0.5 Adc, V_{CE} = 2.0 Vdc)	$V_{BE(on)}$*		—	1	Vdc

* Pulse Test: Pulse Width ≤ 300 μs, Duty Cycle ≤ 2.0%



NOTES
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

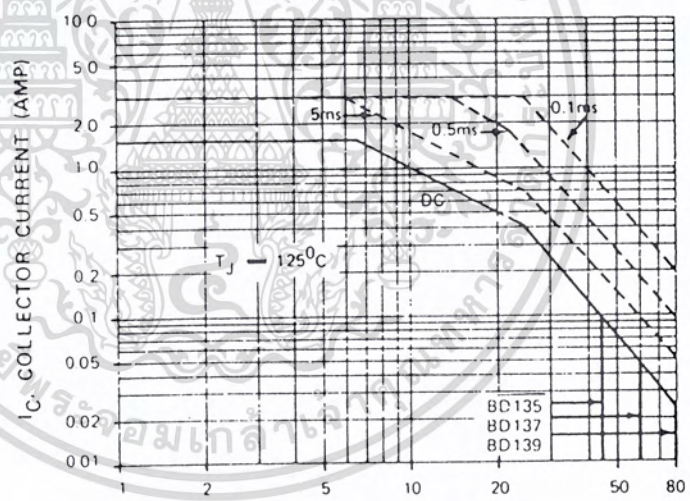
DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	10.80	11.04	0.425	0.435
B	7.50	7.74	0.295	0.305
C	2.42	2.66	0.095	0.105
D	0.51	0.66	0.020	0.026
F	2.93	3.17	0.115	0.125
G	2.39 BSC		0.094 BSC	
H	1.27	2.41	0.050	0.095
J	0.39	0.63	0.015	0.025
K	14.61	16.63	0.575	0.655
M	3° TYP		3° TYP	
Q	3.76	4.01	0.148	0.158
R	1.15	1.39	0.045	0.055
S	0.64	0.88	0.025	0.035
U	3.69	3.93	0.145	0.155
V	1.02	—	0.040	—

STYLE 1:
PIN 1 EMITTER
2 COLLECTOR
3 BASE

CASE 77-06
TO-225AA TYPE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 1 – ACTIVE REGION SAFE OPERATING AREA



Available in H_{FE} groups		Min.	Max.
(At I_C = 0.15 A, V_{CE} = 2 V) H_{FE} group	-6	40	100
	-10	63	160
	-16	100	250

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้