สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องคัดขนาดวัตถุด้วย Image Processing

OBJECT SIZING MACHINE WITH IMAGE PROCESSING

T 036932

โดย

นายมนิศร ศิริวงศ์วัฒนา

นายยุทธศักดิ์ เภาสมบัติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน..... 36932
วัน, เดือน, ปี..... 29 ส.ค. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคัดขนาดวัตถุด้วย Image Processing

OBJECT SIZING MACHINE

WITH IMAGE PROCESSING

โดย

นายมนิศร ศิริวงศ์วัฒนา 39014403

นายยุทธศักดิ์ เภาสมบัติ 39014417

อาจารย์ที่ปรึกษา

อาจารย์ เทอดศักดิ์ ลิ่วหาทอง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2542

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องคัดขนาดวัตถุด้วย Image Processing

OBJECT SIZING MACHINE WITH IMAGE PROCESSING

ผู้จัดทำ 1. นายมนิศร ศิริวงศ์วัฒนา

2. นายยุทธศักดิ์ เภาสมบัติ

.....อาจารย์ที่ปรึกษา

(อาจารย์เทอดศักดิ์ ลิ่วหาทอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคัดขนาดวัตถุด้วย Image Processing

OBJECT SIZING MACHINE WITH IMAGE PROCESSING

นายมนิศร ศิริวงศ์วัฒนา 39014403

นายยุทธศักดิ์ เภาสมบัติ 39014417

โครงงานนี้ได้ผ่านการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(อาจารย์เทอดศักดิ์ ลิ่วหาทอง)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคัดขนาดวัตถุด้วย Image Processing

นายมนิศร ศิริวงศ์วัฒนา
นายยุทธศักดิ์ เภาสมบัติ
อาจารย์เทอดศักดิ์ ลิ่วหาทอง
ปีการศึกษา 2542

บทคัดย่อ

ระบบคัดขนาดวัตถุถูกออกแบบให้สามารถประมวลผลสัญญาณภาพของวัตถุที่ต้องการคัดขนาดได้โดยอัตโนมัติ รายงานฉบับนี้ได้นำเสนอวิธีการแยกวัตถุออกจากภาพและการหาขนาดของวัตถุ ซึ่งมีขั้นตอนดังนี้ ในอันดับแรกจะทำการแยกวัตถุในภาพออกจากพื้นหลังด้วยวิธีเลือกค่าเทรชโฮลที่เหมาะสม จากนั้นจะทำการหารอยขอบภาพด้วยวิธี Contour Techniques และหาขนาดของวัตถุจากรอยขอบภาพ เพื่อนำมาเปรียบเทียบแยกออกเป็นขนาดมาตรฐาน โดยจะคัดขนาดวัตถุประเภทผลไม้รูปทรงกลม เช่น ส้ม หรือ มะนาว เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OBJECT SIZING MACHINE
WITH IMAGE PROCESSING

Mr. Manit Siriwongwatthana
Mr.Yuthasak Paosombati
Mr. Thurdsak Leauhatong(Advisor)
1999

Abstract

OBJECT SIZING MACHINE is designed to evaluate the object's size by automatically. In this report , we provide the method that can detect the object area from picture and find the object's size.

At first , we detect the object area from picture by using a suitable threshold value then we will find border of object area by contour techniques , evaluate the object's size from border of object and compare the object's size with standard size. Thus the object will be cyclic shape fruit such as orange or lime.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	
Abstract	
สารบัญ	
บทที่ 1 บทนำ	1
บทที่ 2 การประมวลผลภาพ	2
2.1 การประมวลผลภาพเชิงเลข	2
2.1.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล	2
2.1.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล	3
2.2 สัญญาณข้อมูลภาพจากดิจิตอลวิดีโอ	4
2.3 ข้อมูลภาพชนิดบิตแมป	5
2.3.1 รูปแบบของไฟล์ข้อมูลภาพชนิดบิตแมป	5
2.3.2 โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป	5
2.4 การสร้างภาพไบนารี	9
2.4.1 การหาค่าเทรชโฮลโดยการกำหนดล่วงหน้า	11
2.4.2 การหาค่าเทรชโฮลจากค่ากลาง	11
2.5 เทคนิคการติดตามรอยขอบภาพ	12
บทที่ 3 กล้อง Color Quick Cam	14
3.1 สิ่งที่จำเป็นสำหรับการใช้งานกล้อง Color Quick Cam	14
3.2 คุณสมบัติพิเศษของกล้อง Cam Color Quick	15
3.3 ความสามารถของกล้อง Color Quick Cam	15
3.3.1 เก็บภาพเคลื่อนไหวใน PC	15
3.3.2 เก็บภาพนิ่งใน PC	15
บทที่ 4 ทฤษฎีพื้นฐานและหลักการทำงานของสเตปมอเตอร์	16
4.1 คุณลักษณะทั่วไป	16
4.2 หลักการทำงาน	17
4.3 ชนิดของสเตปมอเตอร์	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 สเตปมอเตอร์ชนิดปรับค่ารีลัคแตนซ์ได้ 19
4.3.2 สเตปมอเตอร์แบบแม่เหล็กถาวร 21
4.3.3 สเตปมอเตอร์แบบไฮบริจด์ 23
4.3.4 การควบคุมการทำงานของสเตปมอเตอร์ 24
4.3.4.1 การกระตุ้นแบบซิงเกิลเฟส 24
4.3.4.2 การกระตุ้นแบบ 2 เฟส 24
4.3.4.3 การกระตุ้นแบบครึ่งสเตป 24
บทที่ 5 การควบคุมการทำงานเครื่องคัดขนาดวัตถุ 26
5.1 โครงสร้างโดยรวมของระบบ 26
5.2 ระบบการประมวลผลภาพด้วย image processing 29
5.3 โครงสร้างโปรแกรม 30
5.3.1 โปรแกรม capture image 30
5.3.2 โปรแกรม bright และ threshold 31
5.3.3 ส่วนของการ contour 34
5.3.4 ส่วนของการคัดขนาด 35
5.4 ระบบควบคุมชุดลำเลียงและชุดแยกขนาดด้วย microprocessor 36
5.4.1 การควบคุมด้วย microprocessor 36
5.4.1.1 ชุดลำเลียง 36
5.4.1.2 ชุดแยกขนาด 37
5.4.2 การ์ดส่งข้อมูลจาก PC 37
5.4.2.1 ISA slot 37
5.4.2.2 หลักการทำงานการ์ดส่งข้อมูลจาก PC 39
5.4.3 วงจรขับสเตปมอเตอร์ 39
บทที่ 6 ผลการทดสอบ 40
6.1 ขั้นตอนการทดสอบ 40
6.2 ผลการทดสอบการเตรียมข้อมูลภาพ 40
6.3 การทดลองส่วนการหาขอบและขนาดวัตถุ 41
6.4 การทดลองเก็บสถิติเพื่อหาค่าขนาดมาตรฐาน 43
6.5 การทดสอบประสิทธิภาพของเครื่องคัดขนาด 44
6.5.1 ความเร็วในการคัดแยกขนาดต่อ 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7 สรุปและวิจารณ์ 27
7.1 สรุปและวิจารณ์ผลการทดสอบ 46
7.2 สิ่งที่ได้ดำเนินการในภาคเรียนที่ 1 47
7.3 สิ่งที่ได้ดำเนินการในภาคเรียนที่ 2 47
7.4 แนวทางการปรับปรุงและแก้ไขเครื่องคัดขนาดวัตถุ 47
ภาคผนวก
กิตติกรรมประกาศ
เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 แสดงโครงสร้างการทำงานโดยรวมของระบบ	1
รูปที่ 2.1 ภาพการส่งสัญญาณวิดีโอที่อัตรา 24 เฟรมต่อวินาที	5
รูปที่ 2.2 โครงสร้างของบิตแมปไฟล์	6
รูปที่ 2.3 แสดงการเก็บข้อมูลของแต่ละพิกเซล	9
รูปที่ 2.4 ภาพแบบไบนารีและข้อมูลของแต่ละพิกเซล	10
รูปที่ 2.5 แสดงลักษณะการเดินตามรอยขอบภาพ	13
รูปที่ 3.1 แสดงลักษณะภายนอกของกล้อง Color Quick Cam	14
รูปที่ 3.2 แสดงลักษณะภายในของกล้อง Color Quick Cam	14
รูปที่ 4.1 แสดงโครงสร้างของไฮบริดจ์สเตปมอเตอร์ที่มีจำนวนสเตปต่อรอบเท่ากับ 12	16
รูปที่ 4.2 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบ เต็มสเตปหนึ่งเฟส	17
รูปที่ 4.3 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบเต็มสเตปสองเฟส	18
รูปที่ 4.4 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบครึ่งสเตป	19
รูปที่ 4.5 แสดงภาพตัดขวางของสเตปมอเตอร์แบบ 3 เฟส	20
รูปที่ 4.6 แสดงตำแหน่งสมดุลย์ เมื่อเฟสใดเฟสหนึ่งของสเปมอเตอร์ถูกกระตุ้น	20
รูปที่ 4.7 แสดงแรงภายนอกที่มีผลต่อเส้นแรงแม่เหล็ก	20
รูปที่ 4.8 แสดงขั้นตอนการเคลื่อนที่ของโรเตอร์เมื่อสเตปมอเตอร์ถูกกระตุ้น	21
รูปที่ 4.9 แสดงโครงสร้างของสเตปมอเตอร์แบบแม่เหล็กถาวร	22
รูปที่ 4.10 แสดงการทำงานของสเตปมอเตอร์แบบแม่เหล็กถาวรขนาด 4 เ ฟส	22
รูปที่ 4.11 แสดงโครงสร้างของไฮบริจด์สเตปมอเตอร์	23
รูปที่ 5.1 โครงสร้างเครื่องคัดขนาดวัตถุ	26
รูปที่ 5.2 ขั้นตอนการทำงานของระบบโดยรวม	27
รูปที่ 5.3 แสดงวัตถุและตำแหน่งกล้องในการใช้งานจริง	28
รูปที่ 5.4 แสดงโครงสร้างโปรแกรมการประมวลผลภาพด้วย image processing	29
รูปที่ 5.5 Flow Chart แสดงการจับภาพจากกล้องวีดีโอ	32
รูปที่ 5.6 Flow Chart แสดง กระบวนการ Brightness และ Threshold	33
รูปที่ 5.7 Flow Chart แสดงกระบวนการ Contour	34
รูปที่ 5.8 Flow Chart แสดงกระบวนการหาขนาดวัตถุ	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.9 โครงสร้างชุดลำเลียง 36
รูปที่ 5.10 โครงสร้างของชุดแยกขนาด 37
รูปที่ 5.11 แสดงขาของ slot แบบ62 ขา 38
รูปที่ 5.12 วงจรขับสเตปมอเตอร์ 39
รูปที่ 6.1 (a) แสดงภาพ True color ของวัตถุ 41
รูปที่ 6.1 (b) แสดงภาพที่ผ่านกระบวนการ Threshold 41
รูปที่ 6.1 (c) แสดงภาพ Threshold ที่ให้แสงมากเกินไป 41
รูปที่ 6.1 (d) แสดงภาพ Threshold ที่ให้แสงน้อยเกินไป 41
รูปที่ 6.2 (a) แสดงภาพจากการ Threshold 42
รูปที่ 6.2 (b) แสดงภาพการหาขอบภาพวัตถุด้วย Contouring Technic 42
รูปที่ 6.3 (a) แสดงภาพ True color ที่ให้แสงน้อยเกินไป 43
รูปที่ 6.3 (b) แสดงภาพที่ได้จากการ Threshold 43
รูปที่ 6.3 (c) แสดงภาพที่ได้จากการ Contouring Technic 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงข้อมูลในBitmapfileheader	6
ตารางที่ 6.1 แสดงค่าของวัตถุขนาดต่างๆ	43
ตารางที่ 6.2 แสดงค่าเฉลี่ยของขนาดวัตถุชิ้นเดียวกันในเวลาต่างๆ	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันนี้ คอมพิวเตอร์เข้ามามีบทบาทต่อชีวิตคนเรามากขึ้น สามารถช่วยในการทำงานที่ยุ่งยากซับซ้อนได้รวดเร็วขึ้น และ ช่วยให้มนุษย์มีความสะดวกสบายมากขึ้นกว่าแต่ก่อน ดังนั้นจึงได้มีการนำเอาคอมพิวเตอร์ มาใช้ในการประมวลผลต่าง ๆ ซึ่งในโครงงานนี้จะนำเอากล้อง Quickcam (CCD) มาต่อเข้ากับคอมพิวเตอร์ เพื่อจับภาพวัตถุ แล้วส่งข้อมูลไปให้คอมพิวเตอร์ประมวลผลหาขนาดออกมา แล้วสั่งให้เครื่องคัดขนาด แยกขนาดวัตถุตามที่ประมวลผลได้ ซึ่งโดยปกติแล้ว ถ้าใช้เครื่องคัดขนาดที่เป็นระบบ Mechanic ทั่วไป จะมีความลำบากและยุ่งยากในด้านโครงสร้างมาก

ลักษณะโครงสร้างของระบบจะประกอบด้วยส่วนต่าง ๆ ดังรูป

รูปที่ 1.1 แสดงโครงสร้างการทำงานโดยรวมของระบบ

(1) ส่วนการจับภาพ จะใช้กล้อง Quick Cam จับภาพวัตถุ แล้วส่งข้อมูลให้กับคอมพิวเตอร์ โดยลักษณะข้อมูลจะเป็น True Color 24 bit / Pixel

(2) ส่วนประมวลผล คอมพิวเตอร์จะนำข้อมูลจากกล้องมาประมวลผลตาม กระบวนการ (Process) เพื่อหาขนาดของวัตถุ

(3) ส่วนของการคัดขนาด จะเป็นส่วนที่เป็นระบบ Mechanic ที่มีการเชื่อมต่อกับคอมพิวเตอร์ โดยจะคัดขนาดจากข้อมูลที่ได้จากการประมวลผลของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การประมวลผลภาพ

2.1 การประมวลผลภาพเชิงเลข (Digital Image Processing)

การประมวลผลภาพเชิงตัวเลข หมายถึง การนำภาพที่พบทั่วไปมาประมวลผลด้วยเครื่องคอมพิวเตอร์ โดยภาพที่นำมาประมวลผลด้วยเครื่องคอมพิวเตอร์นี้จะถูกแทนที่ด้วยตัวเลขให้อยู่ในรูปของเมตตริกซ์ แต่ภาพที่ได้โดยส่วนมากแล้วจะเป็นภาพที่ได้จากตัวรับสัญญาณ ซึ่งอยู่ในรูปของฟังก์ชัน f(x,y) ที่ต่อเนื่องในระนาบ 2 มิติ โดยจะเป็นสัดส่วนกับความสว่างหรือความเข้มของภาพที่ตำแหน่ง (x,y) ซึ่งเรียกว่า ระดับสีเทา (Gray level)

2.1.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล

ภาพข้อมูลแบบดิจิตอล (Digital Image) เป็นภาพที่ถูกดัดแปลงมาจากอนาลอก อยู่ในรูปของตัวเลข โดยภาพอนาลอกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า พิกเซล (pixel) ในแต่ละพิกเซล จะถูกระบุตำแหน่งโดย (x,y) และค่าระดับสีเทาของพิกเซล โดยเราสามารถแปลงภาพเป็นข้อมูลแบบดิจิตอลได้โดยมีขั้นตอนและวิธีการดังนี้

เมื่อเรานำสัญญาณอนาลอกที่ต้องการประมวลผลมาผ่านส่วนที่เรียกว่าดิจิไตเซอร์ (Digitizor) ซึ่งมีหน้าที่ในการเปลี่ยนสัญญาณอนาลอกเป็นสัญญาณดิจิตอล อุปกรณ์ส่วนนี้ได้แก่ กล้องโทรทัศน์ดิจิไตเซอร์ จากนั้นทำการควอนไตซ์ (Quantizing) เพื่อที่จะประมวลผลสัญญาณด้วยระบบคอมพิวเตอร์ฟังก์ชันของภาพ f(x,y) จะถูกทำให้เป็นสัญญาณที่ไม่ต่อเนื่องทั้งระนาบของภาพ ซึ่งเราเรียกว่า การสุ่มภาพ (Image Sampling) ของฟังก์ชันที่ได้เรียกว่า การควอนไตเซชันระดับสีเทา (Gray level quantization) ก็จะได้ข้อมูลที่เป็นดิจิตอล

สมมุติว่าสัญญาณภาพต่อเนื่อง f(x,y) ถูกดิจิไตซ์ในระนาบ x และ y เป็นช่วงเท่า ๆ กัน เราสามารถจัด f(x,y) ให้อยู่ในรูปของเมตริกซ์ ขนาด NxN ได้ดังสมการที่ 2.1

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,2)\ldots f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2)\ldots f(1,N-1) \\ . & . & . \quad . \\ . & . & . \quad . \\ f(N-1,0) & f(N-1,1) & f(N-1,2)\ldots f(N-1,N-1) \end{matrix} \qquad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งทางขวาของสมการ จะเรียกได้ว่า ภาพดิจิตอล และทุกๆสมาชิกของเมตริกซ์จะเรียกว่า พิกเซล จากขบวนการสร้างภาพดิจิตอลข้างต้นจะเห็นได้ว่าเราสามารถทราบขนาดของความละเอียดของภาพ NxN พิเซล และจำนวนระดับของเกรย์สเกล ในทางปฏิบัติการทำควอนไตเซชันในระบบภาพดิจิตอล จะมีค่าดังสมการที่ 2.2

$$B = NxNxM \text{ บิต} \qquad (2.2)$$

เมื่อ B = ขนาดของข้อมูลภาพที่เป็นดิจิตอล

G = จำนวนของเกรย์สเกลที่ต้องใช้ในการเก็บภาพ

M = จำนวนบิตที่ใช้ในการแทนข้อมูลภาพ 1 พิกเซล

โดย M สามารถหาได้จาก

$$G = 2^M$$

2.1.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล

โดยทั่วไปแล้ว ข้อมูลภาพจะมีความเข้มตั้งแต่ 2 ระดับขึ้นไป แต่ที่ใช้กันมากจะใช้กันที่ค่าระดับความเข้มของจุดภาพเท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของจุดภาพอยู่ในช่วง (0-255) โดยใช้เนื้อที่เก็บข้อมูลภาพขนาด 1 ไบต์ หรือ 8 บิต สำหรับข้อมูล 1 จุดภาพ ($2^8 = 256$) ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มสูง ๆ อาจจะต้องการจำนวนบิตสำหรับเก็บข้อมูลมากกว่า 8 บิต อาจจะเป็น 16 หรือ 24 บิต โดยค่าความเข้มของจุดภาพเท่ากับ 2^{16} และ 2^{24} โดยจะแยกให้ชัดเจนดังนี้

1. ภาพ 2 ระดับ คือมีเพียงแค่จุดขาวกับจุดดำเท่านั้นโดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต
2. ภาพ 16 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 4 บิต ซึ่งทำให้สามารถแสดงได้ 16 ระดับสี หรือ 16 เกรย์สเกล ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือขาวดำ
3. ภาพ 256 ระดับ คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงได้ 256 ระดับสี หรือ 256 เกรย์สเกล ขึ้นอยู่กับภาพนั้นเป็นภาพสีหรือขาวดำ
4. ภาพทรูคัลเลอร์ (True color) คือ ในแต่ละจุดภาพจะมีขนาดของข้อมูล 24 บิต ทำให้สามารถแสดงผลภาพได้เหมือนจริงที่สุด เพราะสามารถแสดงสีได้ถึง 16,777,216 ระดับสี โดยจะแสดงได้แต่ภาพสีเท่านั้น ไม่สามารถแสดงภาพขาวดำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงภาพนี้ใช้วิธีตั้งค่าของแม่สีในตารางสี โดยอาจเลือกสีเป็นแบบ 16 สี จาก 64 สี หรือ 16 สี จาก 262,144 สี หรือ 256 สี จาก 262,144 สี ขึ้นอยู่กับโหมดการแสดงผลสำหรับทรูคัลเลอร์ จะไม่มีการการเลือกสี แสดงผลโดยการส่งค่าสี RGB ผ่าน D/A สีละ 8 บิต ออกไปความแตกต่างของการแสดงภาพสีและขาวดำ คือ ภาพขาวดำจะต้องตั้งให้แม่สีทั้งสามสีมีค่าเท่ากัน เนื่องจาก VGA กำหนดให้แม่สีแต่ละสีใช้ได้เพียง 64 ระดับสีเท่านั้น หากต้องการให้เห็นทั้ง 256 ระดับ ต้องแสดงในโหมดทรูคัลเลอร์ แล้วให้ RGB มีค่าเท่ากัน ซึ่งในโหมดนี้ จะสามารถใช้รีจีสเตอร์ได้ 8 บิต สำหรับแต่ละแม่สี

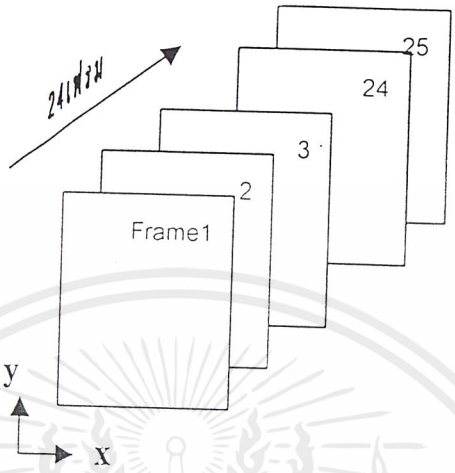
โดยทั่วไปวิธีการประมวลผลภาพเชิงเลขที่ทำให้คอมพิวเตอร์สามารถรู้จักวัตถุภายในภาพได้นั้นแบ่งได้สองระดับด้วยกันคือ การประมวลผลภาพในระดับต่ำ (Low level Image Processing) และการประมวลผลภาพในระดับสูง (High level Image Processing) การประมวลผลภาพระดับต่ำจะเป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมด เพื่อหาตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ โดยมีจุดประสงค์เพื่อนำตัวแปรเหล่านั้นไปใช้ในการประมวลผลระดับสูงต่อไป

การประมวลผลภาพในระดับสูง เป็นการนำผลลัพธ์ที่ได้จากการประมวลผลระดับต่ำมาตีความหรือเพื่อให้คอมพิวเตอร์สามารถรู้จักและเข้าใจภาพได้สำหรับความแตกต่างของการประมวลผลภาพระดับต่ำและสูง คือ ข้อมูลที่นำมาใช้ในการประมวลผล โดยที่การประมวลผลภาพระดับต่ำจะใช้ค่าความสว่างของจุดโดยตรงส่วนการประมวลผลภาพระดับสูงนั้นข้อมูลภาพที่นำมาประมวลผลจะถูกแสดงในรูปของสัญลักษณ์ ซึ่งจะแสดงถึงสิ่งต่าง ๆ ที่อยู่ในภาพ เช่น ขนาด หรือ รูปร่างของวัตถุในภาพ

2.2 สัญญาณข้อมูลภาพจากดิจิตอลวิดีโอ

การส่งสัญญาณข้อมูลภาพจากวิดีโอ จะมีลักษณะการส่งที่เป็นลำดับภาพเดี่ยวหรือเฟรม (frame) ที่ฉายต่อเนื่องกันดังรูปที่ 2.1 เช่น ระบบวิดีโอ NTSC จะส่งด้วยอัตราเร็ว 30 เฟรมต่อวินาที โดยดิจิตอลวิดีโอแต่ละเฟรมจะเป็นข้อมูลภาพดิจิตอลในลักษณะของเมตริกซ์ ซึ่งแต่ละจุดจะเรียกว่าพิกเซล มีค่าของระดับความเข้มสี โดยทั่วไปจะใช้เกรย์สเกลที่มีค่าตั้งแต่ 0-255 โดย 0 แทนความมืดมากที่สุด และ 255 แทนความสว่างมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ภาพการส่งสัญญาณวิดีโอที่อัตรา 24 เฟรมต่อวินาที

2.3 ข้อมูลภาพชนิดบิตแมป

2.3.1 รูปแบบของไฟล์ข้อมูลภาพชนิดบิตแมป

รูปแบบของไฟล์ข้อมูลภาพชนิดบิตแมป เป็นฟอร์แมตของวินโดว์บิตแมป (Bitmap) ซึ่งเป็นมาตรฐานสำหรับไฟล์กราฟิกบนวินโดว์ ซึ่งจะใช้ในการตัดต่อ หรือสำเนาภาพต่าง ๆ ลงบนคลิปบอร์ด (Clipboard) เมื่อเวลาจัดเก็บไฟล์ที่มีสกุล BMP ซึ่งเป็นฟอร์แมตที่ยังสามารถใช้เป็นวอลล์เปเปอร์ได้อีกด้วย

2.3.2 โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป

โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป จะประกอบด้วย 3 ส่วนคือ

1. ข้อมูลเฮดเดอร์ (Header)
2. ข้อมูลจานสี (Palette)
3. ข้อมูลภาพ (Data)

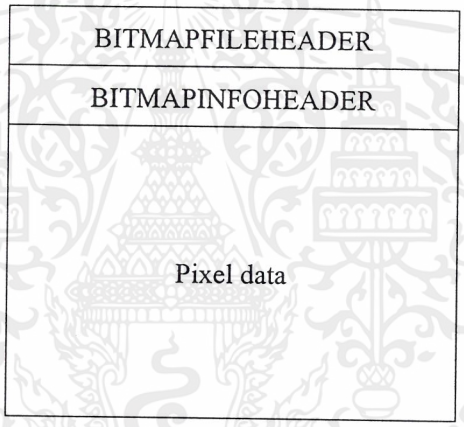
1. ข้อมูลเฮดเดอร์ คือ ข้อมูลที่อยู่บริเวณส่วนหัวของไฟล์ ซึ่งจะประกอบด้วยข้อมูลที่บอกรายละเอียดต่าง ๆ ของภาพ เช่น ความกว้าง ความยาวของภาพ จำนวนสี จำนวนบิต ความละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ข้อมูลจานสี คือ ข้อมูลที่บอกถึงชุดของจานสี ที่เกิดจากการผสมแม่สีทั้งสาม คือ แดง เขียว และน้ำเงิน มาผสมกันได้เป็นสีต่าง ๆ ซึ่งถ้ามีจำนวนสีน้อย ๆ ก็จะมีการเก็บค่าจานสีนี้ลงไฟล์ไปด้วย แต่ถ้าเป็นแบบ 24 บิต จะไม่มีค่าจานสี แต่จะใช้วิธีการเก็บค่าแม่สีลงไปเป็นข้อมูลแทน

3. ข้อมูลภาพ คือ ข้อมูลสีของภาพแต่ละจุดที่มาประกอบกันเป็นรูปภาพ ซึ่งค่าที่เก็บนี้จะเป็นค่าที่ใช้ใน การชี้ตาราง Palette หมายเลขอะไร เช่น จุดที่มีค่าเป็น 10 ก็ไปเปิดดูตารางหมายเลข 10 สมมุติว่า R = 0 , G = 100 , B = 0 ก็จะได้จุดนี้เป็นสีเขียว แต่ถ้าเป็นแบบ 24 บิตจะอ่านข้อมูลขึ้นมา 3 ค่า เป็นค่าแม่สี RGB แล้วนำไปผสมบนหน้าจอแทน

ไฟล์ข้อมูลชนิดบิตแมปมีโครงสร้างดังรูปที่ 2.1 แบ่งออกเป็น 3 ส่วน ได้แก่ Bitmapfileheader เป็นส่วนที่บอกข้อมูลของไฟล์ bitmapinfo เป็นส่วนที่แสดงขนาดและข้อมูลสีของภาพ ส่วนสุดท้ายคือ Pixel data เป็นส่วนเก็บข้อมูลสีแต่ละพิกเซล



รูปที่ 2.2 โครงสร้างของบิตแมปไฟล์

BITMAPINFOHEADER

Byte	Data	Detail
1-2	File Type	Must be ASCII text “BM”
3-6	Size of file	In double word (32-bit integer)
7-10	Reverse for future	Must be zero
11-14	Byte offset to bitmap data	Offset from bitmapfileheader

ตารางที่ 2.1 แสดงข้อมูลในBitmapfileheader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BITMAPINFO

โครงสร้างของ bitmapinfo เขียนได้เป็นดังนี้

```
typedef struct tagBITMAPINFO { // bmi
    BITMAPINFOHEADER    bmiHeader ;
    RGBQUAD             bmiColors[1] ;
}BITMAPINFO ;
```

BITMAPINFO ประกอบด้วย 2 ส่วน คือ BITMAPINFOHEADER เป็นส่วนที่บอกขนาดและข้อมูลสีของภาพบิตแมป และ RGBQUAD ซึ่งจะเก็บค่าตารางสีสำหรับเทียบค่าาจากต่าของแต่ละพิกเซล

BITMAPINFOHEADER

โครงสร้างสามารถเขียนได้ดังนี้

```
typedef struct tagBITMAPINFOHEADER { // bmih
    DWORD       biSize;
    LONG        biWidth;
    LONG        biHeight;
    WORD        biPlanes;
    WORD        biBitCount;
    DWORD       biCompression;
    DWORD       biSizeImage;
    LONG        biXiPelsPerMeter;
    LONG        biYiPelsPerMeter;
    DWORD       biClrUsed;
    DWORD       biClrImportant;
}BITMAPINFOHEADER;
```

โดยแต่ละฟิลด์จะมีความหมายดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

biSize	จำนวนไบต์ของ Header file
biWidth,biHeight	บอกขนาดความกว้าง และยาวของภาพในรูปของพิกเซล
biPlanes	เป็น 1 เสมอ
biBitCount	จำนวนบิตต่อ 1 พิกเซล
biCompression	แสดงการบีบอัดข้อมูล
biSizeImage	บอกขนาดของไฟล์
biXiPelsPerMeter	ความยาวแนวนอนในหน่วยพิกเซลต่อเมตร
biYiPelsPerMeter	ความยาวแนวตั้งในหน่วยพิกเซลต่อเมตร
biClrUsed	จำนวนสีในตารางสีที่จะถูกชี้ด้วยค่าพิกเซลในบิตแมป
biClrImportant	เป็นเลขที่แสดงว่าข้อมูลสีมีความสำคัญในการแสดงผลของบิตแมปถ้าเป็นศูนย์แสดงว่าทุกสีมีความสำคัญ

RGBQUAD

มีโครงสร้างดังนี้

```
typedef struct tagRGBQUAD { // rgbq
        BYTE        rgbBlue;
        BYTE        rgbGreen;
        BYTE        rgbRed;
        BYTE        rgbReserved;
}RGBQUAD;
```

RGBQUAD จะเป็นโครงสร้างที่แสดงความเข้มของสีแดง เขียว และ น้ำเงิน โดยมีความหมายของแต่ละฟิลด์ดังนี้

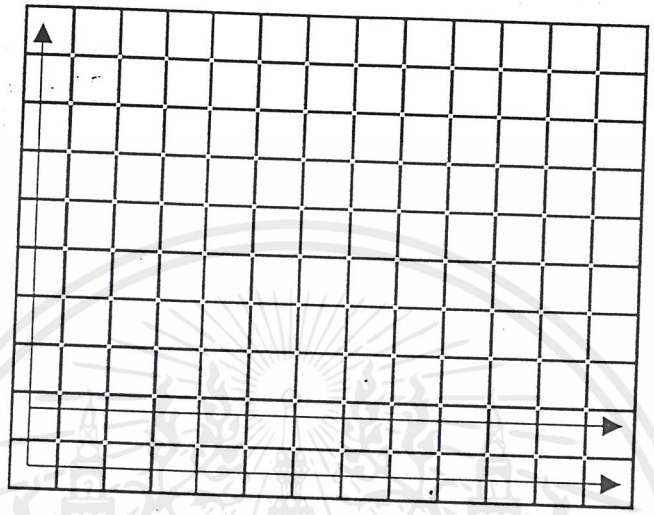
rgbBlue	แสดงความเข้มของสีน้ำเงิน
rgbGreen	แสดงความเข้มของสีเขียว
rgbRed	แสดงความเข้มของสีแดง
rgbReserved	ต้องมีค่าเป็น 0

ในส่วนของ bmiColors ของโครงสร้าง BITMAPINFO จะประกอบด้วยอาร์เรย์ของ RGBQUAD เพื่อเป็นตารางเปรียบเทียบ สีของข้อมูลในแต่ละพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pixel data

เป็นส่วนเก็บข้อมูลสีของแต่ละพิกเซลของภาพ โดยข้อมูลแรกจะเป็นค่าสีของพิกเซลที่อยู่แถวล่างสุดที่ตำแหน่งซ้ายสุด ข้อมูลลำดับต่อไปจะเรียงทางขวาจากแถวล่างจนถึงแถวบนสุด

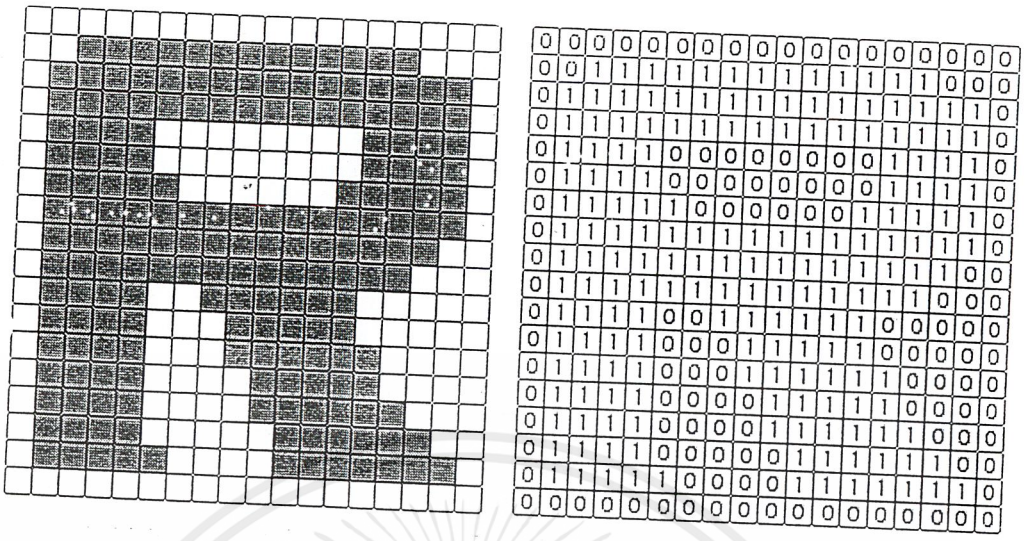


รูปที่ 2.3 แสดงการเก็บข้อมูลของแต่ละพิกเซล

2.4 การสร้างภาพไบนารี

การสร้างภาพไบนารี หมายถึง การแปลงข้อมูลภาพที่มีความเข้มหลายระดับ ให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ คือ 1 จุดภาพ มีค่าได้แค่ 2 ค่าเท่านั้น โดยเป็น 0 กับ 1 ซึ่ง 1 จะหมายถึงจุภาพที่มีสีดำ และ 0 จะหมายถึงจุดภาพที่มีสีขาว การแปลงข้อมูลภาพหลายระดับไปเป็นภาพไบนารีจึงมีความจำเป็นและมีประโยชน์มากในการแสดงผลภาที่มีระดับความเข้มของภาพหลายระดับบนอุปกรณ์ที่มีความสามารถในการแสดงผลได้ 2 ระดับ ประโยชน์อีกประการคือ การลดเนื้อที่การเก็บข้อมูลภาพให้เหลือเพียง 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ภาพแบบไบนารีและข้อมูลของแต่ละพิกเซล

ในการสร้างภาพไบนารี สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล (Thresholding Technics) โดยพิจารณาว่าจุดใดควรจะเป็นจุดขาวหรือจุดดำ จะกระทำโดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งเรียกว่า ค่าเทรชโฮล เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุ และพื้นหลัง โดยค่าของจุดภาพใด ๆ ทีมีค่าน้อยกว่าค่าเทรชโฮลจะถูกกำหนดให้เป็น ค่า 1 (จุดสีดำ) และถ้าค่าของจุดภาพมรค่ามากกว่าค่าเทรชโฮลจะถูกเปลี่ยนให้เป็นค่า 0 (จุดสีขาว) ซึ่งการทำงานสามารถแสดงได้ดังสมการที่ 2.3

$$b(x,y) = \begin{cases} 1 ; & g(x,y) < Thr \\ 0 ; & g(x,y) \geq Thr \end{cases} \qquad (2.3)$$

b(x,y) ข้อมูลภาพผลลัพธ์เป็นภาพไบนารี

g(x,y) ข้อมูลภาพอินพุทที่มีระดับความเข้ม 0 ถึง L ระดับ

Thr ค่าเทรชโฮลเป็นค่าคงที่ ระหว่าง 0 ถึง L ระดับ

1 จุดดำ

0 จุดขาว

โดยที่ L คือระดับความเข้มของจุดภาพสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสร้างภาพไบนารีโดยใช้เทคนิคเทรชโฮลเพื่อให้ได้ผลลัพธ์ที่ได้คมชัดและเหมาะสม สิ่งที่สำคัญที่สุดคือ ค่าเทรชโฮล เนื่องจากถ่เลือกใช้ค่าเทรชโฮลที่ไม่เหมาะสม ภาพที่ได้จะไม่คมชัด และรายละเอียดบางส่วนอาจขาดหายไป ดังนั้นปัญหาการสร้างภาพด้วยวิธีเทรชโฮลคือ ทำอย่างไร จึงสามารถคำนวณหาค่าเทรชโฮลที่เหมาะสมได้

2.4.1 การหาค่าเทรชโฮลโดยการกำหนดล่วงหน้า (Preassigned Threshold Value)

การหาค่าเทรชโฮลโดยวิธีการกำหนดล่าวงหน้านี้เป็นวิธีที่ง่ายที่สุด เป็นการกำหนดค่าเทรชโฮลเองจากผู้ใช้ ซึ่งจะขึ้นกับประสบการณ์ของผู้ใช้นั้น ๆ โดยการเลือกค่าคงที่ค่าหนึ่งจะเป็นค่าที่อยู่ระหว่างค่าต่ำสุดกับค่าสูงสุด ของระดับความเข้มของข้อมูลอินพุท

2.4.2 การหาค่าเทรชโฮลจากค่ากลาง (Mid Range Threshold Value)

การหาค่าวิธีนี้จะพิจารณาจากค่ากลาง โดยอาศัยการคำนวณพื้นฐานทางสถิติ ในการหาค่ากลางหรือค่าเฉลี่ย (Mean) มาประยุกต์ ค่าที่ได้จะเป็นค่ากึ่งกลางระหว่างค่าระดับความเข้มสูงสุดกับค่าต่ำสุดของข้อมูลภาพอินพุท สำหรับการหาค่ากึ่งกลางจะได้ ดังสมการที่ 2.4

$$Thr = \frac{Max(g(x,y)) + Min(g(x,y))}{2} \qquad (2.4)$$

โดยที่ Thr ค่าเทรชโฮล

g(x,y) ข้อมูลภาพอินพุท ที่มีระดับความเข้ม 0 ถึง L ระดับ

Max(g(x,y)) ค่าสูงสุดเกรย์สเกลของข้อมูลอินพุท

Min(g(x,y)) ค่าต่ำสุดเกรย์สเกลของข้อมูลอินพุท

การหาค่าเทรชโฮลจากค่าเฉลี่ยเลขคณิต หาได้จากสมการที่ 2.5

$$Thr = \frac{\sum gi(x,y)}{NxN} \qquad (2.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เทคนิคการติดตามรอยขอบภาพ (Contour Following)

เทคนิคการติดตามรอยขอบภาพ ถูกนำมาใช้ในการแยกขกและคัดลอกส่วนของรูปภาพใด ๆ ที่อยู่บนรูปใหญ่ ข้อมูลภาพที่จะนำมาประมวลผลด้วยเทคนิคนี้ต้องอยู่ในรูปของข้อมูลไบนารี นั่นคือ จุดภาพจะแสดงด้วยตัวเลข 0 กับ 1 เท่านั้น

การทำงานของเทคนิคการติดตามรอยขอบของภาพ เป็นการเดินไต่ไปตามขอบระหว่างส่วนที่เป็นรูปภาพ กับส่วนที่เป็นพื้นเบื้องหลัง โดยจะตรวจกวาดไปทุก ๆ พิกเซล โดยจะเริ่มจากจุดมุมซ้ายของภาพ ตรวจกวาดไปในทิศทางจากซ้ายไปขวา และเลื่อนลงจากบนลงล่าง เมื่อตรวจกวาดมาพบจุดใด ๆ ที่มีค่าของจุดภาพเป็น 1 ก็จะเปลี่ยนลักษณะการเคลื่อนที่ไปยังจุดถัดไปใหม่โดยมีเงื่อนไขการเคลื่อนที่ดังนี้

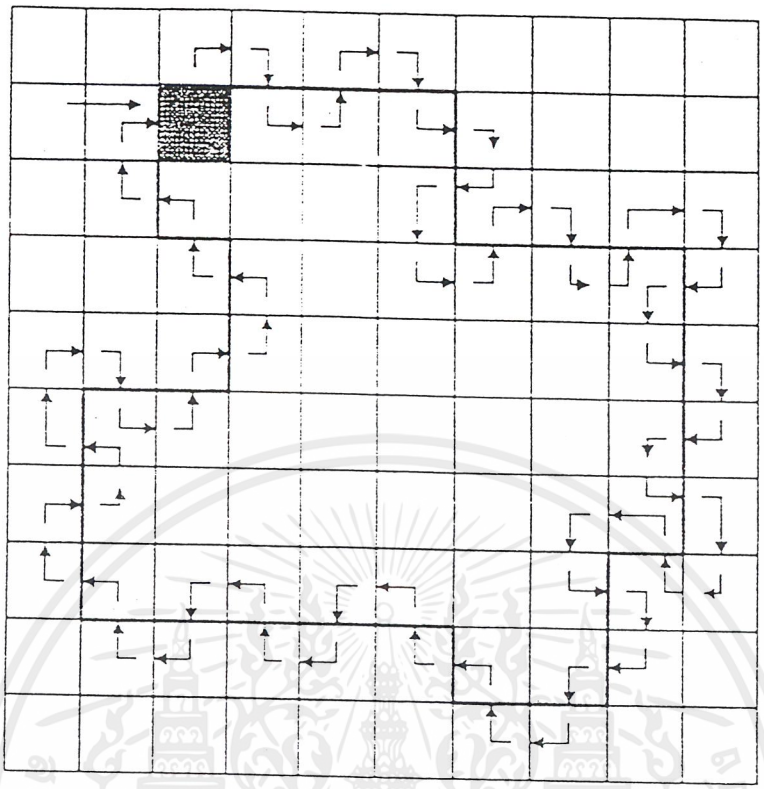
1. ถ้าจุดที่อยู่ปัจจุบันเป็นจุดของภาพหรือมีค่าของจุดเป็น 1 ให้เลี้ยวซ้ายแล้วก้าวเดินตรงไปข้างหน้ายังจุดถัดไป

2. ถ้าจุดที่อยู่ในปัจจุบันเป็นพื้นเบื้องหลัง หรือมีค่าของจุดเป็น 0 ให้เลี้ยวขวา แล้วก้าวเดินตรงไปข้างหน้าไปยังจุดถัดไป

3. การเคลื่อนที่จะสิ้นสุดลง เมื่อจุดที่อยู่ปัจจุบันเป็นจุดเดียวกันกับจุดเริ่มต้น

ในรูปที่ 2.5 ได้แสดงลักษณะการทำงานของเทคนิคการติดตามรอยขอบภาพซึ่งจะแสดงการเคลื่อนที่ไปตามจุดต่าง ๆ ที่เป็นขอบภาพ เริ่มจากจุดที่ถูกแรเงาไว้ซึ่งเป็นจุดของภาพแรกที่ตรวจกวาดตามเงื่อนไขที่กำหนด เมื่อการเคลื่อนที่วนกลับมาถึงจุดเริ่มต้น ก็จะทราบจุดที่เป็นขอบของภาพทั้งหมด ขณะที่เดินไต่ขอบนั้นก็จดจำค่าพิกัดไปด้วย เมื่อเดินถึงจุดเดิม ก็นำค่าพิกัดที่น้อยสุดและมากสุดมากำหนดขอบเขตของขนาดของวัตถุเพื่อใช้เปรียบเทียบกันต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

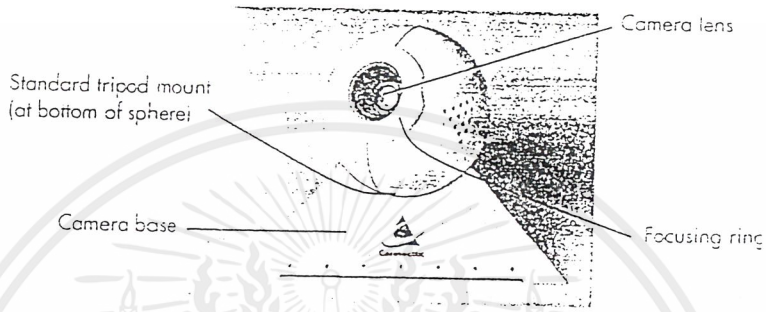


รูปที่ 2.5 แสดงลักษณะการเดินตามรอยขอบภาพ

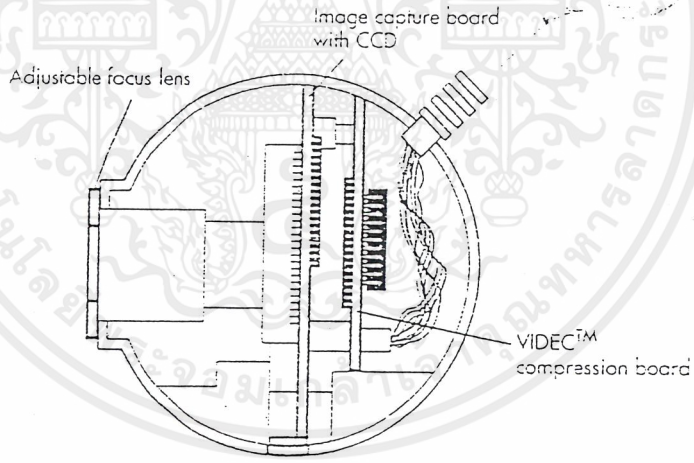
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

กล้อง Color Quick Cam



รูปที่ 3.1 แสดงลักษณะภายนอกของกล้อง Color Quick Cam



รูปที่ 3.2 แสดงลักษณะภายในของกล้อง Color Quick Cam

3.1 สิ่งที่จำเป็นสำหรับการใช้งานกล้อง Color Quick Cam

- เครื่อง PC ที่ใช้ CPU 486 ขึ้นไป
- Microsoft windows 3.1 , Windows for workgroups 3.11 , Windows 95
- หน่วยความจำ 8 MB สำหรับ Windows 3.1 และ 12 MB สำหรับ Windows 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Video for Windows 1.1e หรือ สูงกว่านั้น
- พอร์ทขนาน 1 พอร์ท โดยพอร์ทเป็นแบบทิศทางเดียว หรือ 2 ทิศทาง
- หน่วยความจำในฮาร์ดดิส 2 MB สำหรับซอฟแวร์ ของกล้อง Color Quick Cam
- หน่วยความจำในฮาร์ดดิสอย่างน้อย 5 MB สำหรับสรางภาพเคลื่อนไหว
- หน่วยความจำในฮาร์ดดิส 350 MB สำหรับภาพ 24 bit color 640x480 พิกเซล หรือ ประมาณ 80 KB สำหรับภาพ 320 X 240 พิกเซล ใน JPEG
- ไมโครโฟนสำหรับการบันทึกเสียงในภาพแบบ movie
- sound card ซึ่ง compatible กับ windows

3.2 คุณสมบัติพิเศษของกล้อง Color Quick Cam

- ให้ภาพที่มีความละเอียด 640 X 320 พิกเซล 24 bit color
- frame rate 30 fps (เปลี่ยนแปลงได้ตามขนาดของ windows และ PC)
- ได้รับไฟเลี้ยงจากคีย์บอร์ดด้วยไฟอย่างน้อย 2 วัตต์
- ปรับโฟกัสได้จาก 1 นิ้วถึงอนันต์ maximum lens speed = f/1.6

3.3 ความสามารถของกล้อง Color Quick Cam

3.3.1 เก็บภาพเคลื่อนไหวใน PC

การใช้งานกล้อง Color Quick Cam นั้นเป็นวิธีที่ง่าย และประหยัดวิธีหนึ่งในการเก็บภาพเคลื่อนไหวไว้ใน PC ทำให้สามารถที่จะติดต่อสื่อสารกันโดยแสดงภาพเคลื่อนไหวได้และสามารถหลีกเลี่ยงปัญหาในขั้นตอนการติดตั้ง Hardware , การซื้อ video board ที่แพง หรือการศึกษาถึงวิธีการใช้งาน การใช้กล้อง Color Quick Cam นั้นจะช่วยให้เราสามารถที่จะ capture ภาพได้ง่ายและทำการดัดแปลงแก้ไขได้ด้วย

3.3.2 เก็บภาพนิ่งใน PC

กล้อง Color Quick Cam มี software ที่ทำให้สามารถบันทึกภาพนิ่งให้อยู่ในรูปแบบ BMP , TIFF , JPEG ซึ่งสามารถนำมาประมวลผลในการนำเสนองาน หรือ โปรแกรม graphic เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ทฤษฎีพื้นฐานและหลักการทำงานของสเตปมอเตอร์

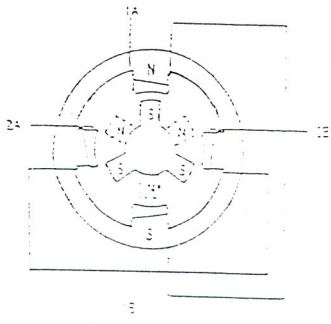
4.1 คุณลักษณะทั่วไป

สเตปมอเตอร์ คือ อุปกรณ์ที่เคลื่อนที่เป็นสเตป โดยการกระตุ้นด้วยวิธีการทางแม่เหล็กไฟฟ้าซึ่งเปลี่ยนแปลงสัญญาณดิจิตอลอินพุทซึ่งเป็นพัลซ์ไปเป็นการเคลื่อนที่แบบอนาลอกที่เอาท์พุทสเตปมอเตอร์บางครั้งถูกเรียกว่า สเตปปิ้งมอเตร์ หรือ สเตปเปอร์มอเตอร์เพราะว่าเป็นอุปกรณ์ที่เคลื่อนที่เมื่อถูกกระตุ้นด้วยโวลท์เตจหรือกระแส ซึ่งโดยมากจะเป็นไฟฟ้ากระแสตรง เอาท์พุทจะมีจำนวนของคำสั่ง อินพุทมีลักษณะเป็นพัลซ์ โดยเมื่อป้อนกระแสให้กับสเตปมอเตอร์แล้วจะมีการเคลื่อนที่เพิ่มขึ้น 1 สเตป

สเตปมอเตอร์มีขดลวดอยู่หลายขด โดยแต่ละขดถูกควบคุมการจ่ายกระแสโดยสัญญาณ ON –OFFทำให้สามารถควบคุมตำแหน่งในการเคลื่อนที่ได้ตราบที่ยังไม่เกิดการเกินกำลัง (OVERLOAD) จนกระทั่งมีการกระตุ้นโดยสัญญาณควบคุมการจ่ายกระแสอีกครั้ง และเนื่องจากสเตปมอเตอร์สามารถควบคุมตำแหน่งได้โดยการส่งสัญญาณกระตุ้นทำให้ไม่จำเป็นต้องใช้ตัววัดตำแหน่งในการทำงาน ข้อเสียที่สำคัญ คือ มีข้อจำกัดที่ความเร็วเมื่อเปรียบเทียบ DC SERVO MOTOR ที่มีขนาดเท่ากันและยังมีการสั่นสะเทือนสูงทำให้การเคลื่อนที่เป็นไปอย่างไม่ราบเรียบ

4.2 หลักการทำงานของสเตปมอเตอร์

สเตปมอเตอร์สามารถแบ่งโครงสร้างได้เป็น 2 ส่วน คือ สเตเตอร์และโรเตอร์ ตัวสเตเตอร์เป็นส่วนที่อยู่กับที่ ประกอบด้วยขดลวดทองแดงซึ่งพันรอบแกนเหล็กส่วนโรเตอร์เป็นส่วนที่เคลื่อนที่มีลักษณะเป็นเหล็กทรงกลม และที่ผิวนอกมีลักษณะเป็นซีกฟันซึ่งทำจากแม่เหล็กถาวรดังรูปที่ 4.1



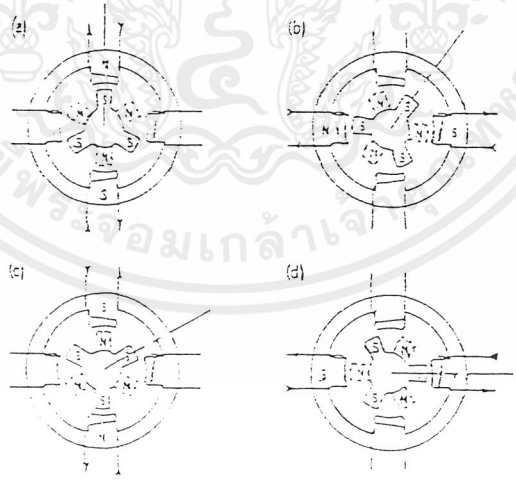
รูปที่ 4.1 แสดงโครงสร้างของไฮบริดจ์สเตปมอเตอร์ที่มีจำนวนสเตปต่อรอบเท่ากับ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เมื่อยังไม่มีการจ่ายกระแสให้กับขดลวดมอเตอร์ ซีกฟันอันใดอันหนึ่งของโรเตอร์จะอยู่ในตำแหน่งที่ตรงกันกันกับซีกอันใดอันหนึ่งของสเตเตอร์ ทั้งนี้เป็นเพราะแม่เหล็กถาวรที่ตัวของโรเตอร์พยายามที่จะทำให้ค่าความต้านทานทางแม่เหล็ก (RELUCTANCE) มีค่าน้อยที่สุด ซึ่งที่จุดที่ซีกฟันของตัวโรเตอร์และสเตเตอร์ตรงกันนั้นมีค่าความต้านทานทางแม่เหล็กน้อยที่สุดทำให้เกิดเส้นแรงแม่เหล็กไฟฟ้ามากที่สุด และจากรูปที่ 4.1 เส้นแรงแม่เหล็กไฟฟ้าจะทำให้เกิดขั้วแม่เหล็กเหนือและใต้ขึ้นมา 2 คู่ทั้งที่ตัวสเตเตอร์และโรเตอร์ ดังรูป ค่าทอร์ค (TORQUE) ที่ทำให้ตัวโรเตอร์สามารถยึดอยู่ในตำแหน่งดังกล่าวนี้เรียกว่า ดีเทนท์ทอร์ค (DETENTTORQUE) (หมายความว่า การที่จะทำให้มอเตอร์เคลื่อนที่ในขณะที่ไม่ได้จ่ายกระแสให้กับขดลวดของมอเตอร์จะต้องออกแรงมากกว่าค่าของดีเทนท์ทอร์ค จึงจะทำให้โรเตอร์เริ่มเคลื่อนที่ได้) รูปที่ 4.1 นั้นมี 12 ตำแหน่งที่สามารถเกิดดีเทนท์ทอร์คได้

เมื่อจ่ายกระแสให้กับขดลวดที่อยู่ในสเตเตอร์คู่ใดคู่หนึ่ง ดังรูปที่ 4.2(a) จะทำให้เกิดขั้วแม่เหล็กเหนือและใต้ที่ซีกฟันของตัวสเตเตอร์ ซึ่งจะดึงดูดซีกฟันของตัวโรเตอร์ที่มีขั้วแม่เหล็กที่มีศักย์ต่างกันที่อยู่ใกล้ที่สุดเข้าไว้ ตำแหน่งนี้เรียกว่า สเตเบิ้ลโพซิชัน (STABLE POSITION) ของโรเตอร์จะมีจำนวนตำแหน่งเท่ากับจำนวนซีกฟันของโรเตอร์ และแรงที่จะทำให้โรเตอร์เปลี่ยนตำแหน่งไปจากตำแหน่งสเตเบิ้ลโพซิชันได้นี้เรียกว่า โฮลดิ้งทอร์ค (HOLDING TORQUE)



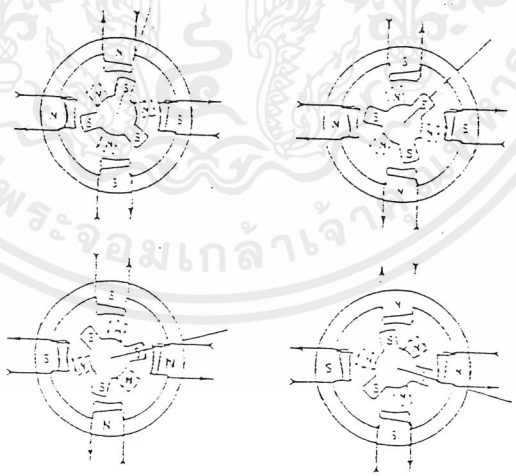
รูปที่ 4.2 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบ เต็มสเตปหนึ่งเฟส

เมื่อสับเปลี่ยนการจ่ายกระแสให้แก่ขดลวด จากขดหนึ่งไปอีกขด เนื่องจากขดลวดวางอยู่ในตำแหน่งที่ต่างกัน 90° ก็จะทำให้ตัวสเตเตอร์ดึงดูดซีกฟันของตัวโรเตอร์อีกซีกหนึ่งที่ใกล้ที่สุดเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
36932

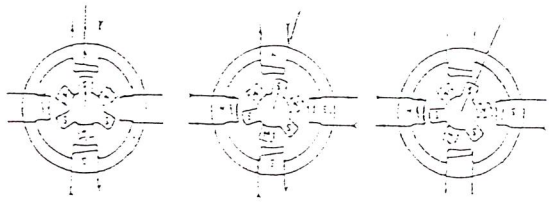
ไว้ ซึ่งจะทำให้โรเตอร์เคลื่อนที่ไป 1 สเตป หรือ 30° ดังรูปที่ 4.2(b) จากนั้นก็เปลี่ยนไปจ่ายกระแสให้กับขดลวดชุดแรกโดยในคราวนี้เปลี่ยนทิศทางการไหลของกระแสให้ตรงข้ามกับครั้งแรก ซึ่งจะทำให้ตัวโรเตอร์เคลื่อนที่ไปอีกหนึ่งสเตป ดังรูปที่ 4.2(c) หลังจากนั้นก็ไปจ่ายกระแสให้กับขดลวดชุดที่ 2 โดยกลับทิศทางของกระแสที่ป้อนให้เช่นกัน ทำให้โรเตอร์หมุนไป 90° ดังรูป 4.2(d) และถาหากเราป้อนกระแสให้กับมอเตอร์เหมือนกับที่เราป้อนในครั้งแรกแล้ว ซีกฟันซีกถัดไปของตัวโรเตอร์จะอยู่ในตำแหน่งที่เหมือนกับรูปที่ 4.2(a) อีกครั้ง ถ้าเราต้องการเคลื่อนที่หนึ่งรอบ เราต้องทำการกระตุ้นให้มอเตอร์เคลื่อนที่ไปจนครบ 12 สเตป และถ้าต้องการให้โรเตอร์หมุนไปอีกทิศทางหนึ่ง ก็ทำการสลับลำดับในการจ่ายกระแส จากรูปที่ 4.2(a) , (b) , (c) , (d) ไปเป็น (a) , (d) , (c) , (b) ตามลำดับ

ถ้าจ่ายกระแสให้แก่ขดลวดพร้อม ๆ กัน ซีกฟันของโรเตอร์จะอยู่ที่ตำแหน่งระหว่างซีกฟันของสเตเตอร์ เพราะฉะนั้นการจ่ายกระแสให้แก่มอเตอร์แบบนี้จะให้ทอร์คมากกว่าแบบที่จ่ายกระแสในเวลาขณะใดขณะหนึ่งเพียงขดเดียว ดังรูปที่ 4.3 แสดงถึงการป้อนกระแสให้กับขดลวดแต่ละขดเพื่อให้สเตปมอเตอร์เคลื่อนที่



รูปที่ 4.3 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบเต็มสเตปสองเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงขั้นตอนการทำงานของสเตปมอเตอร์แบบครึ่งสเตป

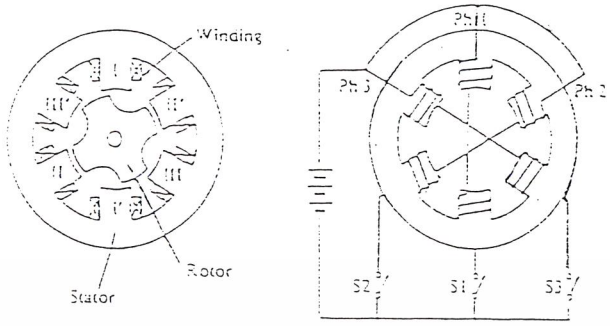
แต่ถ้าในตอนแรกจ่ายกระแสให้กับขดลวดพร้อม ๆ กันทั้งสองขด การทำสลับกันไปจะทำให้โรเตอร์เคลื่อนที่ไปสเตปละ 15° ดังรูปที่ 4.4 และในการขับเคลื่อนแบบนี้จะทำให้สเตปต่อรอบเพิ่มขึ้นเป็นเท่าตัว การขับสเตปมอเตอร์แบบนี้เรียกว่า การขับแบบครึ่งสเตป ซึ่งนิยมใช้กันในงานอุตสาหกรรมเป็นอย่างมาก แม้ว่าในบางครั้งจะทำให้ทอร์คน้อยกว่า แต่มีข้อดีกว่าการขับแบบเต็มสเตป คือ การเคลื่อนที่เป็นไปอย่างละเอียดกว่า อีกทั้งยังมีความแม่นยำสูงและการสั่นสะเทือนน้อยกว่า

4.3 ชนิดของสเตปมอเตอร์

4.3.1 สเตปมอเตอร์ชนิดปรับค่ารีลัคแตนซ์ได้ (VARIABLE RELUCTANCE STEPPING MOTOR)

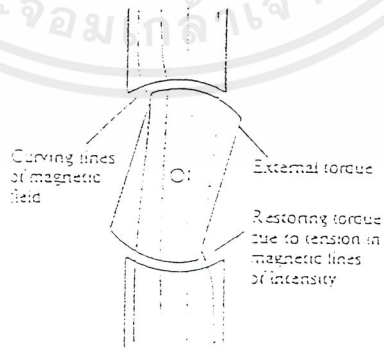
สเตปมอเตอร์ชนิดนี้สามารถปรับค่ารีลัคแตนซ์ได้ รูปที่ 4.5 แสดงภาพตัดขวางของสเตปมอเตอร์แบบ 3 เฟส โดยที่สเตเตอร์มีฟันทั้งหมด 6 ซี่ ซี่ที่อยู่ตรงข้ามกันทำมุม 180° ซึ่งกันและกันจะเป็นเฟสเดียวกัน ขดลวดที่พันอยู่ที่ส่วนของสเตเตอร์ในแต่ละเฟสจะต่ออนุกรมหรือขนานก็ได้ จากรูปที่ 4.5 เป็นการต่อแบบอนุกรม ส่วนของโรเตอร์นั้นมีฟัน 4 ซี่ ทั้งโรเตอร์และสเตเตอร์ทำมาจากโลหะซิลิกอน ซึ่งมีสภาพซึมซับทางแม่เหล็กสูงและยอมให้สนามแม่เหล็กจำนวนมากไหลผ่านได้ ฟันของสเตเตอร์ในเฟสเดียวกันจะมีขั้วต่างกัน โดยซี่ฟัน I , II , III เป็นขั้วเหนือและซี่ I', II' , III' เป็นขั้วใต้หลังจากถูกกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



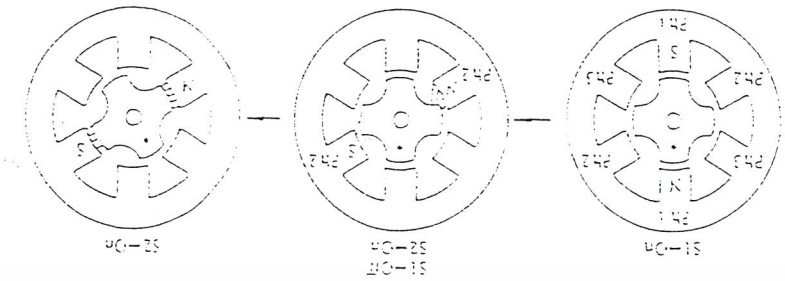
รูปที่ 4.5 แสดงภาพตัดขวางของสเตปมอเตอร์แบบ 3 เฟส

รูปที่ 4.6 แสดงตำแหน่งสมดุลย์ เมื่อเฟสใดเฟสหนึ่งของสเปมอเตอร์ถูกกระตุ้น



รูปที่ 4.7 แสดงแรงภายนอกที่มีผลต่อเส้นแรงแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



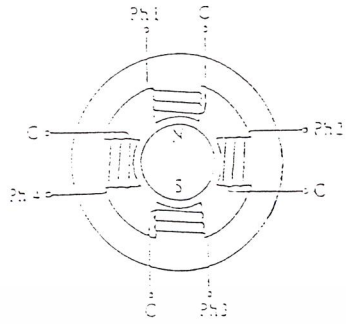
รูปที่ 4.8 แสดงขั้นตอนการเคลื่อนที่ของโรเตอร์เมื่อสเตปมอเตอร์ถูกกระตุ้น

กระแสไหลในแต่ละเฟสถูกควบคุมโดยสวิทช์ เปิด/ปิด ถ้าเฟส I ถูกกระตุ้นจะมีกระแสไหลและเกิดฟลักซ์แม่เหล็กดังรูปที่ 4.6 แกนโรเตอร์จะอยู่ตำแหน่งเดียวกับซี่ I และ I' ทำให้ทั้งโรเตอร์และสเตเตอร์อยู่ในแนวเดียวกัน กรณีนี้จะทำให้ค่ารีลัคแตนซ์มีค่าน้อยที่สุดซึ่งเป็นตำแหน่งที่สมดุลย์ ถ้าโรเตอร์ถูกกระทำจากแรงภายนอกจะทำให้เปลี่ยนตำแหน่ง ดังรูป 4.7 แรงบิดกระทำกับโรเตอร์ ในทิศทางตามเข็มนาฬิกาทำให้ตำแหน่งเปลี่ยนไป มีผลทำให้เส้นแรงแม่เหล็กเคลื่อนที่จากซี่ของโรเตอร์และสเตเตอร์ เมื่อโรเตอร์และสเตเตอร์ไม่ได้อยู่ในแนวเดียวกันแล้วค่ารีลัคแตนซ์จะมีค่ามาก จากนั้นสเตปมอเตอร์จะทำตัวให้มีค่ารีลัคแตนซ์น้อยที่สุด การย้ายจากมุมที่เกิดการกระตุ้นแต่ละครั้งให้กลับไปยังตำแหน่งเดิมเรียกว่า สเตป

4.3.2 สเตปมอเตอร์แบบแม่เหล็กถาวร

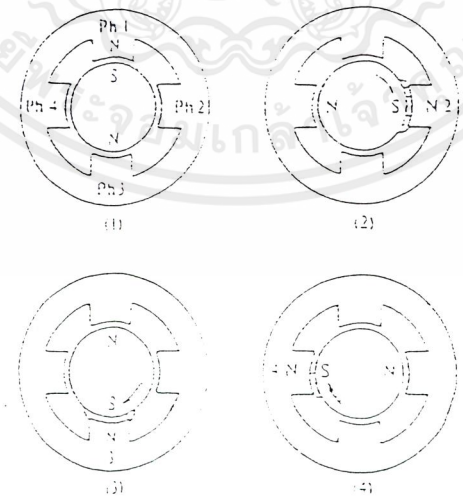
สเตปมอเตอร์ชนิดนี้ใช้แม่เหล็กถาวรเป็นโรเตอร์ และมีซี่ฟันของโรเตอร์ล้อมรอบ ซี่ฟันของสเตเตอร์ถูกพันด้วยขดลวดสำหรับสร้างสนามแม่เหล็ก เมื่อต้องการให้สเตปมอเตอร์แบบแม่เหล็กถาวรมีขนาดมุมของสเตปเล็กลงจะต้องเพิ่มขั้วแม่เหล็กของโรเตอร์และซี่ฟันของสเตเตอร์ แต่ก็มีขีดจำกัดในการเพิ่มจำนวนขั้วแม่เหล็กของโรเตอร์ เนื่องจากการสร้างสนามแม่เหล็กถาวรให้มีโครงสร้างโดยมีขั้วแม่เหล็กหลายขั้วทำได้ยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงโครงสร้างของสเตปมอเตอร์แบบแม่เหล็กถาวร

ตัวอย่างการทำงานของสเตปมอเตอร์แบบแม่เหล็กถาวร สมมติว่าสเตปมอเตอร์แบบแม่เหล็กถาวรขนาด 4 เฟส มีโรเตอร์เป็นแม่เหล็กถาวรและมีซี่สเตเตอร์ 4 ฟัน ซึ่งรอบ ๆ พันด้วยขดลวด มีรูปแบบพื้นฐานของการทำงานคือ เมื่อสร้างสัญญาณกระตุ้นตามลำดับเฟส โรเตอร์จะหมุนไปตามทิศทางของการกระตุ้น ดังแสดงในรูปที่ 4.10 ข้อเสียของสเตปมอเตอร์แบบนี้คือ มีขนาดมุมสเตปใหญ่ทำให้ความละเอียดต่อรอบน้อยเพราะว่าการสร้างแม่เหล็กถาวรให้มีหลายขั้วนั้นทำได้ยาก และส่วนใหญ่จะมีโครงสร้างขนาดเล็ก ทำให้ค่าทอร์คต่ำ ถ้าต้องการปรับปรุงประสิทธิภาพของทอร์ค แม่เหล็กถาวรที่ใช้ต้องมีสภาพความเป็นแม่เหล็กที่สูง

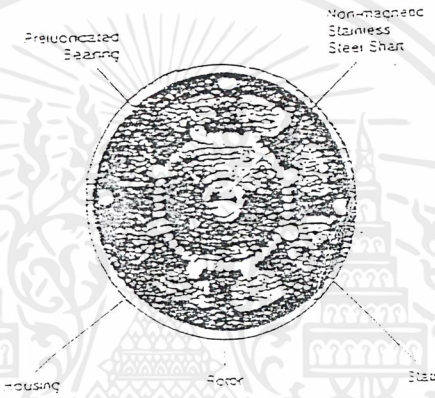


รูปที่ 4.10 แสดงการทำงานของสเตปมอเตอร์แบบแม่เหล็กถาวรขนาด 4 เ ฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 สเตปมอเตอร์แบบไฮบริจด์

สเตปมอเตอร์ชนิดนี้มีแกนโรเตอร์เป็นแม่เหล็กถาวร โดยมีการทำงานร่วมกันของมอเตอร์แบบแม่เหล็กถาวรและมอเตอร์แบบปรับค่ารีลัคแตนซ์ได้ ไฮบริจด์สเตปมอเตอร์นี้มีโครงสร้างของสเตเตอร์คล้ายกับของแบบปรับค่ารีลัคแตนซ์ได้ แต่ต่างกันที่การต่อขดลวด โดยที่แต่ละเฟสของสเตปมอเตอร์แบบปรับค่ารีลัคแตนซ์ได้จะมี 2 ขด แต่ละขดจะมีขั้วต่างกันแต่ไฮบริจด์ ขดลวดทั้งสองจะพันอยู่ที่ขั้วเดียวกัน เรียกว่า ไบโพลาร์ (BIPOLAR) ซึ่งในการกระตุ้นแต่ละครั้งจะให้ขั้วที่แตกต่างกัน



รูปที่ 4.11 แสดงโครงสร้างของไฮบริจด์สเตปมอเตอร์

โครงสร้างของมอเตอร์จะมีสนามแม่เหล็กถาวรอยู่ตรงกลางระหว่างทั้งสองเฟส การเหนี่ยวนำสนามแม่เหล็กทำได้โดยใช้สนามแม่เหล็กซึ่งสร้างจากสเตเตอร์ซึ่งเป็นสนามแม่เหล็กแบบเฮทเทอโรโพลาร์ (HETEROPOLAR FIELD) ดังนั้นทอร์คเกิดจากการทำงานร่วมกันของสนามแม่เหล็ก 2 ชนิด คือ สนามจากแม่เหล็กถาวรและสนามแม่เหล็กเหนี่ยวนำที่เกิดจากการกระตุ้นของขดลวดแต่ละขด โครงสร้างของซี่ฟันของสเตเตอร์จะใหญ่กว่าซี่ฟันของโรเตอร์เล็กน้อยเพื่อเพิ่มความถูกต้องทางตำแหน่งของการเคลื่อนที่ ข้อดีของไฮบริจด์สเตปมอเตอร์ คือ มีขนาดเล็กมีความละเอียดต่อรอบสูง มีค่าทอร์คสูง แต่แบบปรับค่ารีลัคแตนซ์ได้มีแรงเฉื่อยทางแมคคานิคส์น้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การควบคุมการทำงานของสเตปมอเตอร์

วิธีการควบคุมการทำงานของสเตปมอเตอร์โดยการกระตุ้นเฟสมี 3 วิธีคือ

4.3.4.1 การกระตุ้นแบบซิงเกิลเฟส (Single Phase Excitation)

เป็นการกระตุ้นแบบเฟสเดียวตามสัญญาณที่ป้อนเข้ามาที่ชุดขับสเตปมอเตอร์ดังนี้

PHASE	PULSE							
A	1	0	0	0	1	0	0	0
B	0	1	0	0	0	1	0	0
C	0	0	1	0	0	0	1	0
D	0	0	0	1	0	0	0	1

4.3.4.2 การกระตุ้นแบบสองเฟส (Two Phase Excitation)

เป็นการกระตุ้นแบบทีละสองเฟสคู่พร้อมกัน ดังนี้

PHASE	PULSE							
A	1	0	0	1	1	0	0	1
B	1	1	0	0	1	1	0	0
C	0	1	1	0	0	1	1	0
D	0	0	1	1	0	0	1	1

4.3.4.3 การกระตุ้นแบบครึ่งสเตป (Half Step Excitation)

เป็นการกระตุ้นแบบการหมุนของทั้งสองแบบไว้ในแบบเดียวกัน ดังนี้

PHASE	PULSE							
A	1	1	0	0	0	0	0	1
B	0	1	1	1	0	0	0	0
C	0	0	0	1	1	1	0	0
D	0	0	0	0	0	1	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการกระตุ้นทั้ง 3 แบบนี้จะมีคุณสมบัติที่แตกต่างกัน ดังนั้นการเลือกวิธีการกระตุ้นจึง จำเป็นต้องพิจารณาให้เหมาะสม แบบเฟสเดียวนั้นจะให้ความเทียงตรงสูง แต่ทอร์คต่ำ แบบ 2 เฟส ความเที่ยงตรงจะน้อยกว่าแบบแรก แต่มีแรงบิดที่สูงกว่า และการกระตุ้นแบบครึ่งสเตปจะมีความเที่ยงตรงน้อยมากแต่จะให้ทอร์คที่สูงที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

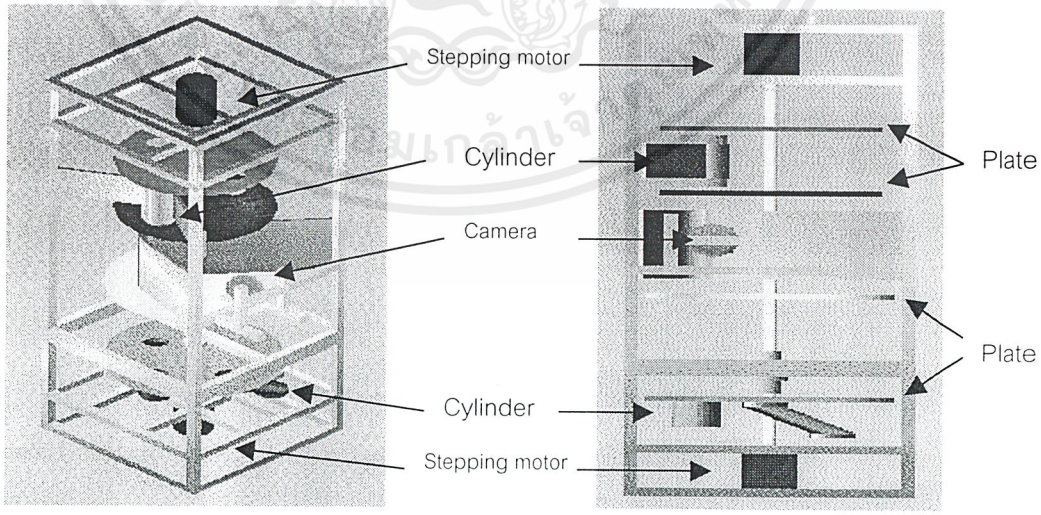
บทที่ 5
การควบคุมการทำงาน เครื่องคัดขนาดวัตถุ

5.1 โครงสร้างโดยรวมของระบบ

เครื่องคัดขนาดวัตถุ เป็นเครื่องมือที่ถูกออกแบบเพื่อใช้ในการคัดแยกขนาดวัตถุ (มะนาว) ออกเป็นขนาดต่าง ๆ กัน เพื่อสะดวกในการคัดแยกจำนวนมาก ๆ โดยระบบจะประกอบไปด้วย

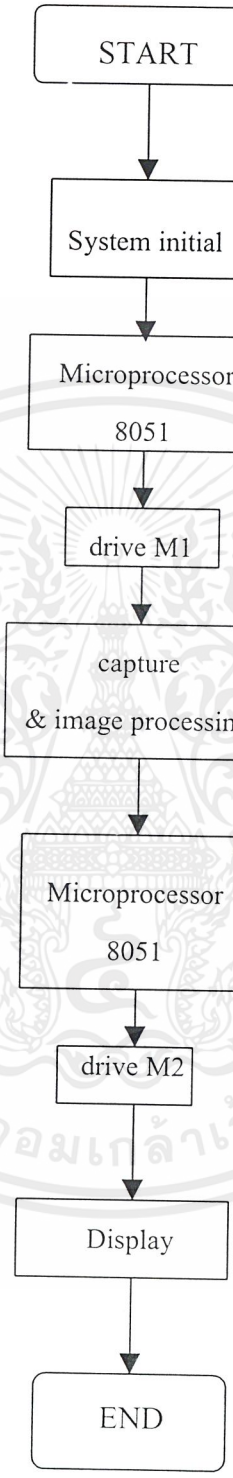
1. ชุดลำเลียง
2. ชุดแยกขนาด
3. การ์ดส่งข้อมูลจาก PC
4. ชุดไมโครโปรเซสเซอร์
5. วงจรขับสเตปมอเตอร์และสเตปมอเตอร์
6. กล้อง color quick cam
7. software การประมวลผลภาพและควบคุมสเตปมอเตอร์

หลักการทำงานของระบบ คือ การจับภาพวัตถุแล้วนำมาประมวลผลด้วยวิธี image processing เพื่อที่จะหาขนาดแล้วนำไปเปรียบเทียบกับขนาดมาตรฐานที่กำหนดไว้ล่วงหน้า โดยชุดลำเลียงจะส่งวัตถุเข้ามาทีละ 1 ลูก จากนั้นจะถูกจับภาพด้วยกล้อง color quick cam เพื่อนำไปหาขนาดและเปรียบ แล้วส่งไปที่ชุดแยกขนาดเพื่อบรรจุแยกภาชนะ



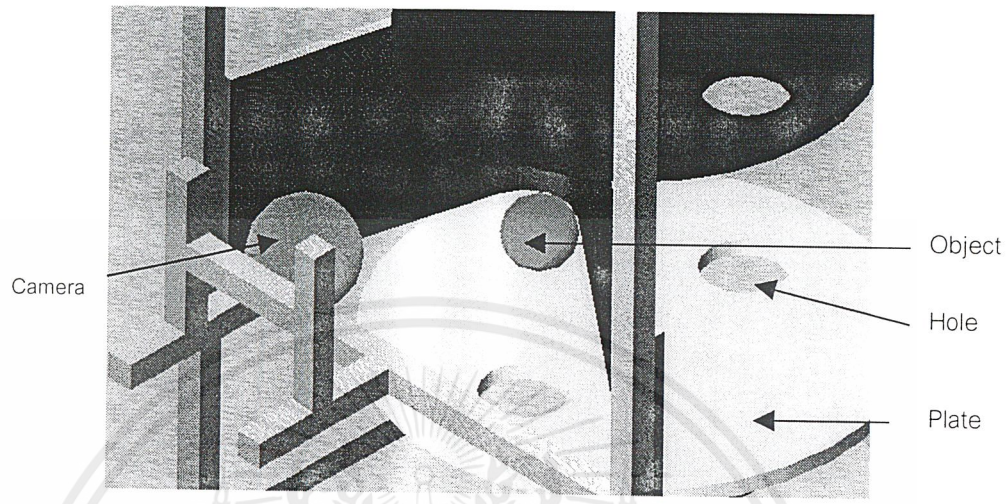
รูปที่ 5.1 โครงสร้างเครื่องคัดขนาดวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 ขั้นตอนการทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงวัตถุและตำแหน่งกล้องในการใช้งานจริง

system initial : เป็นการตั้งค่าเริ่มต้นให้กับ software ที่ใช้ควบคุมการทำงานของระบบ ดังนี้
- ตำแหน่งเริ่มต้นของมอเตอร์ M1 ,M2
- ขนาดมาตรฐานที่จะใช้เปรียบเทียบ

Micripro 8051 : เป็นตัวควบคุมการหมุนของ M1 ,M2 ให้เป็นไปตามสเตปที่กำหนดไว้

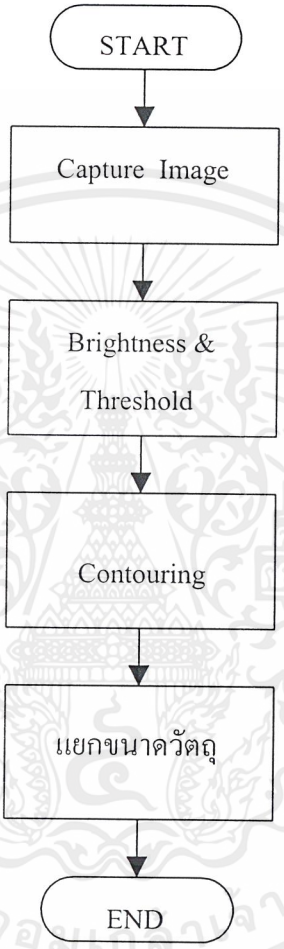
Drive M1 , M2 : เป็นวงจรขับกระแสให้แก่ M1 ,M2 ซึ่งถูกสั่งงานจาก Microprocessor 8051 โดย M1 จะ เป็นมอเตอร์ของชุดลำเลียงและ M2 จะเป็นของชุดแยกขนาด

Capture & Image Processing : เป็นโปรแกรมการจับภาพด้วยกล้อง quick cam เพื่อที่จะนำไปหาขนาดและเปรียบเทียบ

Display : แสดงการแยกขนาดออกเป็นจำนวนผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ระบบการประมวลผลภาพด้วย image processing



รูปที่ 5.4 แสดงโครงสร้างโปรแกรมการประมวลผลภาพด้วย image processing

การทำงานของโปรแกรมของเครื่องคัดขนาดวัตถุ มีดังนี้

1. Capture Image เป็นขั้นตอนการจับภาพเคลื่อนไหวจากกล้องวีดีโอเป็นภาพนิ่ง มาเก็บในหน่วยความจำ เพื่อไปทำการประมวลผลในขั้นต่อ ๆไป

2. Brightness & Thershold เป็นการนำภาพที่ได้จากขั้นตอนแรกที่เป็นภาพสีจริง (True Color) มาแปลงให้เป็นภาพระดับความสว่าง (Brightness) หรือ ภาพ Grayscale แล้วทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Threshold คือ ตั้งระดับความสว่างไว้ ค่าๆหนึ่งแล้วเปรียบเทียบระดับความสว่างนี้กับระดับความสว่างในภาพ Grayscale โดย ถ้าระดับ GrayScale นี้ต่ำกว่าระดับที่ตั้งไว้ ข้อมูลใหม่จะเป็น "0" ถ้าระดับ Grayscale สูงกว่าระดับที่ตั้งไว้ ข้อมูลใหม่จะเป็น "1" นั่นคือข้อมูลที่ได้จากการ Threshold จะเป็น ข้อมูลแบบดิจิตอล 1 Bit / Pixel

3. Contouring เป็นขั้นตอนที่นำภาพจากการ Threshold มาหาขอบของวัตถุในภาพ โดยใช้หลักการ Contour ตามที่ได้กล่าวมาแล้วและเก็บข้อมูลของภาพ เฉพาะส่วนขอบที่หาได้เอาไว้เท่านั้น

4. แยกขนาดวัตถุ เป็นการนำเอาภาพที่ได้จากการ Contour มาหาขนาดโดยเทียบอัตราส่วนในภาพ ให้เป็นขนาดจริง แล้วแยกขนาดตามระดับที่ตั้งไว้

5.3 โครงสร้างโปรแกรม

ภาษาที่ใช้เขียนโปรแกรมนี้จะใช้ MFC VISUAL C++ ซึ่งเป็นภาษาที่ทำงานบนระบบ Windows ทำให้ไม่มีปัญหาเรื่องหน่วยความจำไม่พอ เนื่องจากการประมวลผลเกี่ยวกับรูปภาพเป็นกระบวนการที่ต้องใช้หน่วยความจำมากพอสมควร โดยโปรแกรมจะแบ่งเป็นส่วน ๆ ดังนี้

5.3.1 โปรแกรม Capture Image

โดยปกติ ข้อมูลภาพจากกล้องวีดีโอจะเปลี่ยนแปลงไปเรื่อย ๆ ตามอัตราการจับภาพของกล้อง (30 Frame /second) โดยทุก ๆ ครั้งที่มี Frameของภาพใหม่เข้ามา จะมีการเรียก Interupt service routine ของการรับภาพ โปรแกรมจะมีการตรวจสอบว่า คำสั่ง SetOnframeCallback เป็น TRUE หรือ FALSE ถ้าเป็น FALSE โปรแกรมก็จะรับภาพเข้ามาปกติ แต่ถ้าเป็น TRUE จะรับภาพเข้ามาเช่นกัน และมีการเรียก ฟังก์ชั่น OnFrameCallback มาใช้งาน ซึ่งโปรแกรมการจับภาพ จะอยู่ในฟังก์ชั่น OnFrameCallback นี้

การรับข้อมูลภาพเข้ามาแต่ละ Frameนั้น นอกจากจะได้ข้อมูลภาพจริงๆ มาแล้ว ยังได้รายละเอียดเกี่ยวกับภาพด้วย เช่น

- ความสูงของภาพ ในหน่วย Pixel
- ความกว้างของภาพ ในหน่วย Pixel
- ความละเอียดของภาพ ในหน่วย Pixel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งรายละเอียดต่างๆ เหล่านี้ จะทำให้สามารถสร้าง Bitmap File ขึ้นมาใหม่ได้ แล้วนำข้อมูลภาพเหล่านั้นมาทำการประมวลผล ซึ่งในส่วนนี้จะเป็นขั้นตอนใน Routine ของฟังก์ชั่น OnFrameCallback ทั้งสิ้น โดยจะแสดงภาพที่ Capture ได้เป็นภาพนิ่งบนจอ Monitor และมีการ Copy ข้อมูลภาพนั้นไว้ในหน่วยความจำของ คอมพิวเตอร์ โดยตั้งชื่อว่า BMP1

5.3.2 โปรแกรม Brightness และ Threshold

จะนำภาพที่ได้จากการจับภาพมาแปลงเป็น Brightness Bitmap ก่อนโดยใช้สมการดังนี้

$$\text{Brightness} = (R+G+B)/3$$

โดย Brightness : ความสว่างของแต่ละ Pixel

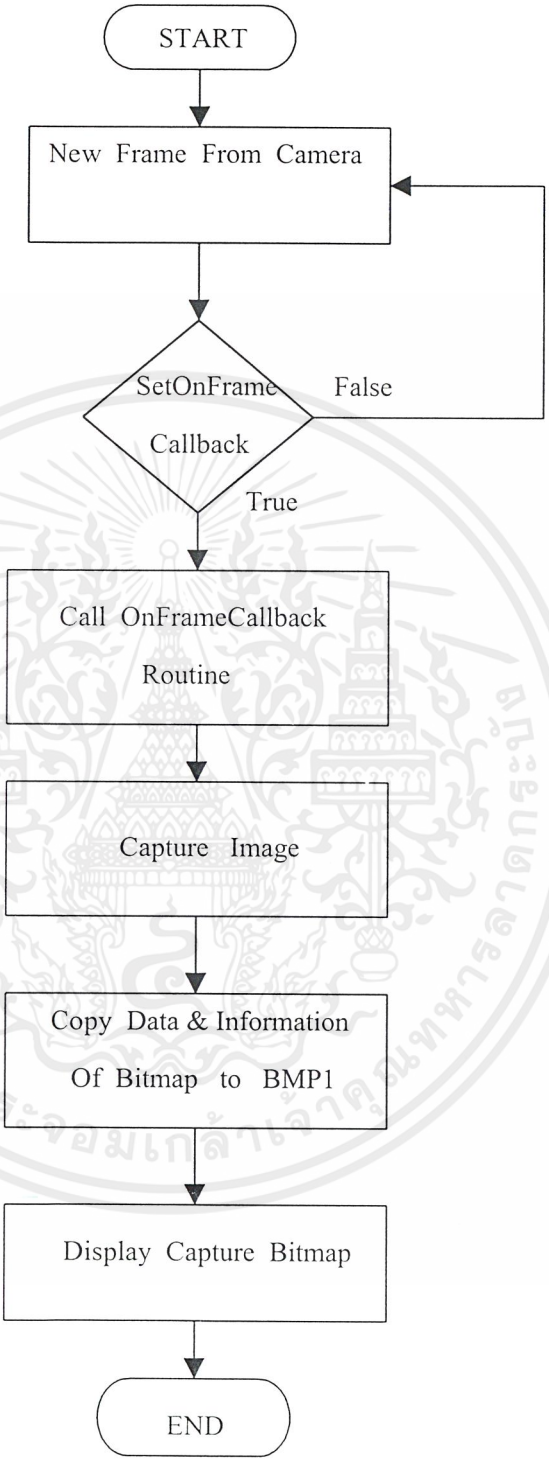
R : ค่า RED Component เป็นองค์ประกอบสีแดงในแต่ละ Pixel

G : ค่า GREEN Component เป็นองค์ประกอบสีเขียวในแต่ละ Pixel

B : ค่า BLUE Component เป็นองค์ประกอบสีน้ำเงินในแต่ละ Pixel

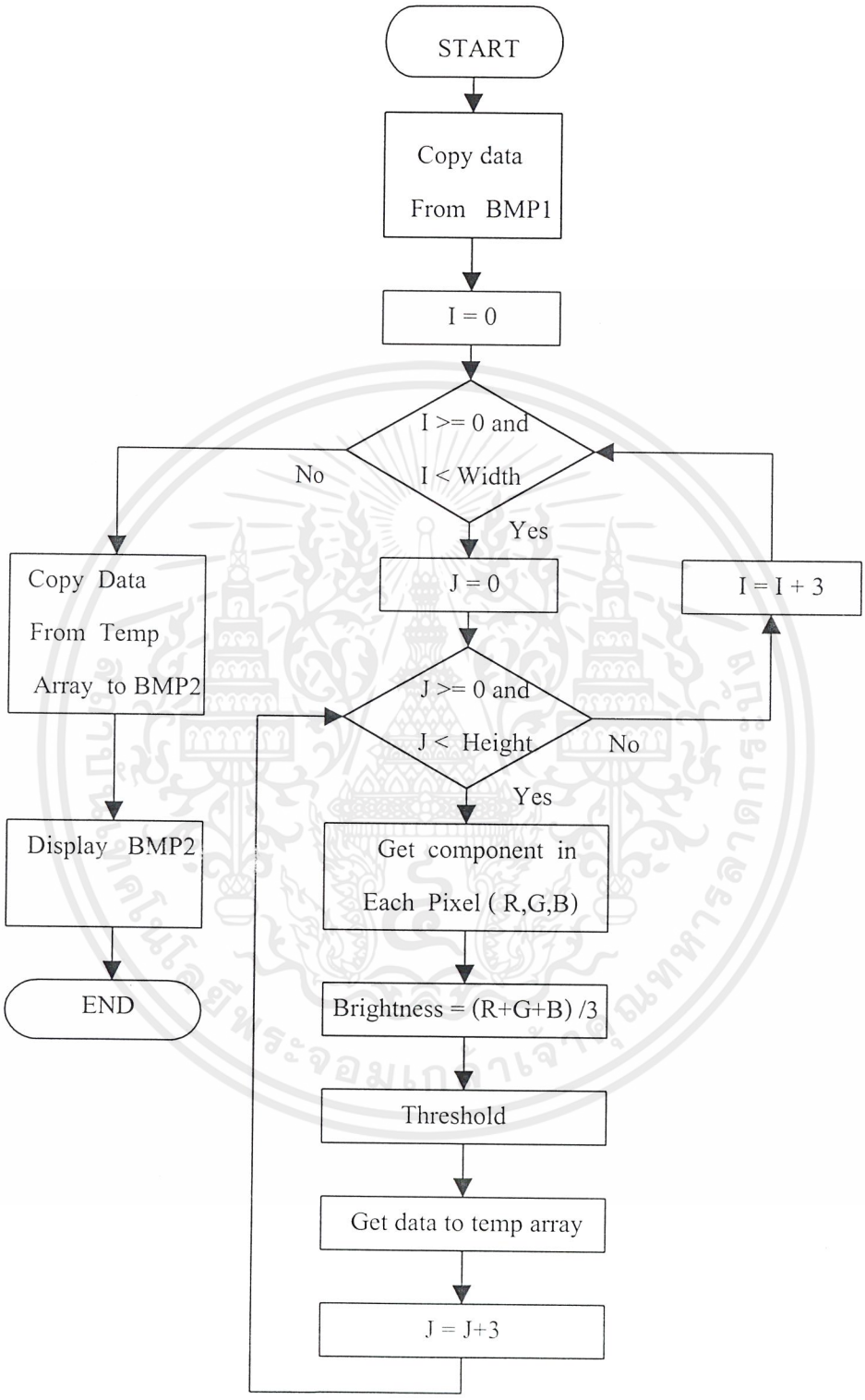
แล้วพิจารณา Brightness ว่าอยู่ต่ำกว่า หรือ สูงกว่า ระดับ Threshold ที่ตั้งไว้ซึ่งจะได้ข้อมูลออกมาเป็น “0” หรือ “1” เท่านั้น (ขาว หรือ ดำ) โดยขั้นตอนการทำงานแสดงได้ตาม Flow Chart ดังรูปที่ 5.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.5 Flow Chart แสดงการจับภาพจากกล้องวีดีโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

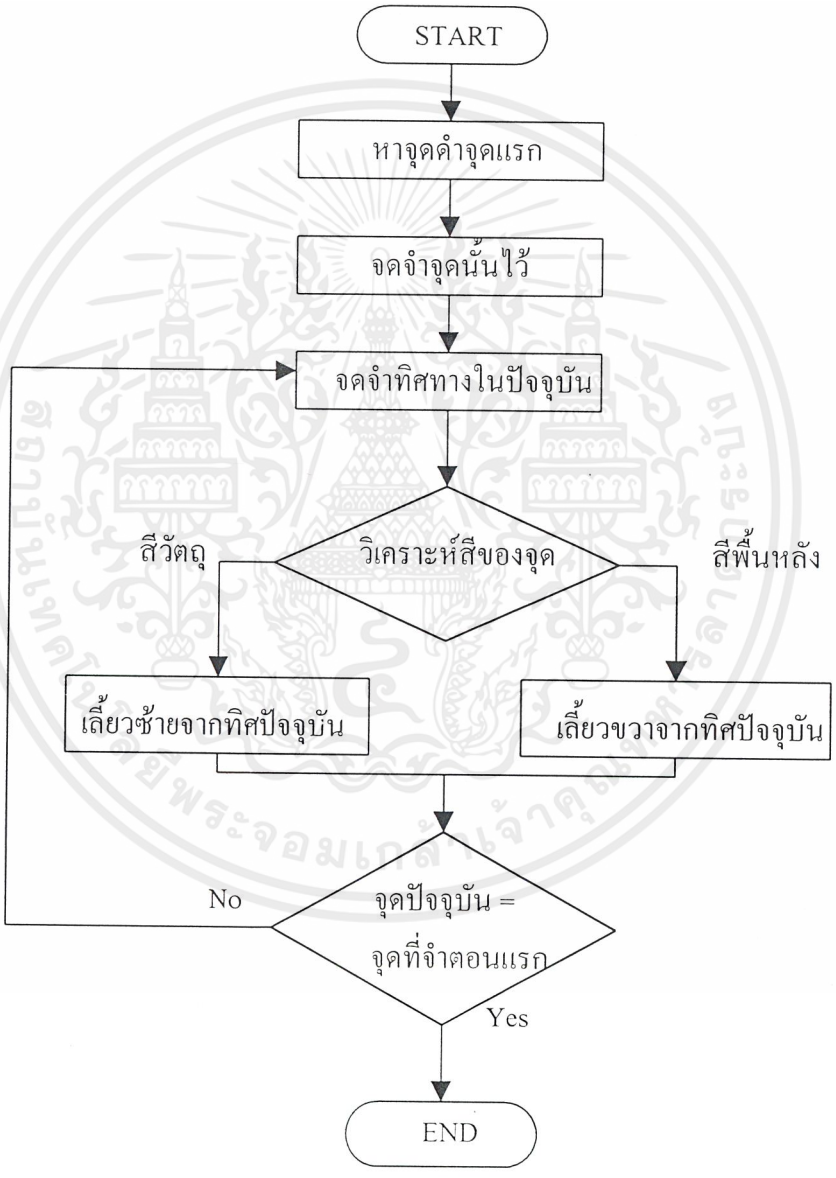


รูปที่ 5.6 Flow Chart แสดง กระบวนการ Brightness และ Threshold

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 ส่วนของการ Contour

ส่วนนี้เป็นส่วนของการหาขอบของวัตถุในภาพโดยจะทำการ Scan หาจุดดำจุดแรก ออกมา แล้วจดจำจุดนั้นไว้เพื่อเป็นจุดเปรียบเทียบในการวนหา ขณะเดียวกันจะจดจำทิศทางเอาไว้ เพื่อใช้อ้างอิงเวลาวนรอบ หลังจากนั้นจะพิจารณาในจุดต่อไป ถ้าเป็นจุดภาพ จะวนไปทางซ้าย แต่ถ้าเป็นจุดพื้น จะวนทางขวาจนพบจุดแรกที่เป็นจุดอ้างอิงจึงจะหยุดการทำงาน

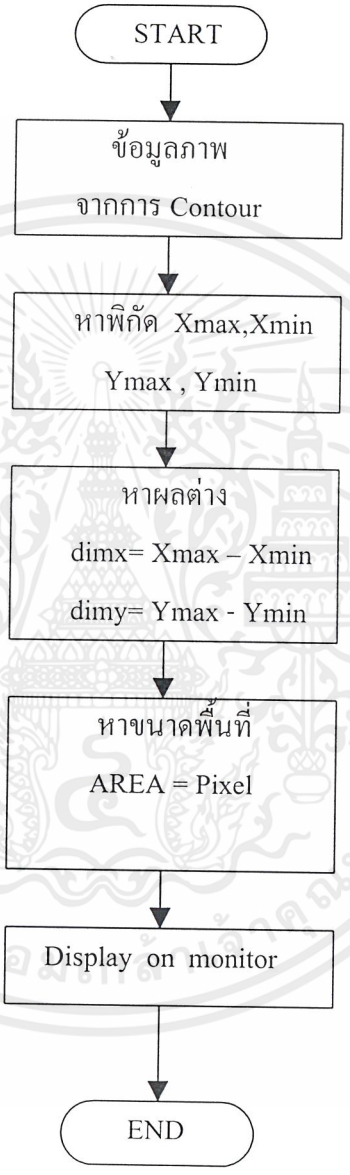


รูปที่ 5.7 Flow Chart แสดงกระบวนการ Contour

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 ส่วนของการคัดขนาด

ส่วนนี้เป็นการนำเอาภาพที่ได้จากการ Contour มาหาขนาดที่แท้จริง โดยจะหาขนาดของรูปภาพแล้ว ทำการ Scaling ขนาดให้กลายเป็นขนาดของวัตถุจริง แล้วแสดงผลออกมาทาง Monitor



รูปที่ 5.8 Flow Chart แสดงกระบวนการหาขนาดวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ระบบควบคุมชุดลำเลียงและชุดแยกขนาดด้วย Microprocessor

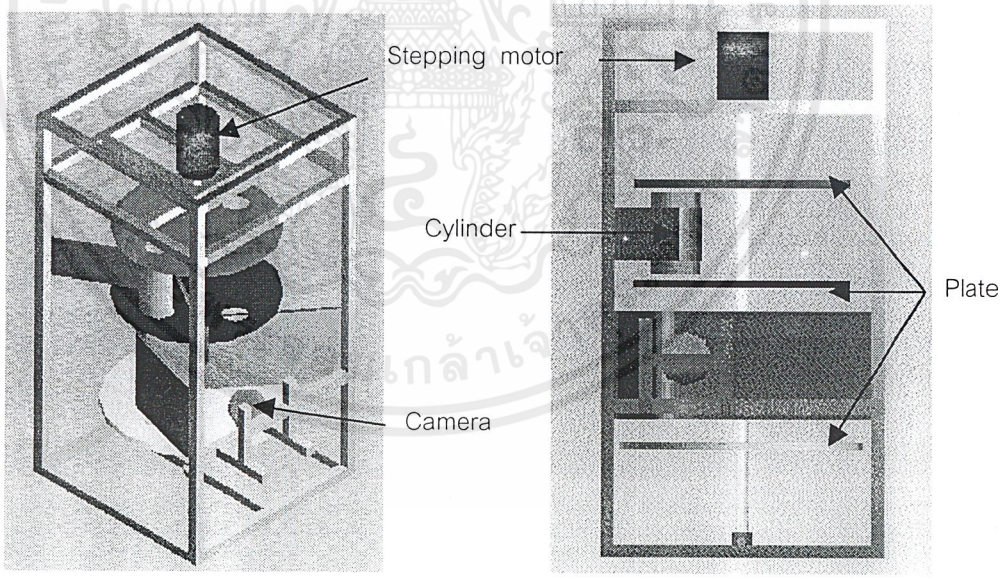
1. การควบคุมด้วย Microprocessor
2. การ์ดส่งข้อมูลจาก PC
3. วงจร drive สเตปมอเตอร์

5.4.1 การควบคุมด้วย Microprocessor

ปัจจุบันนี้ Microprocessor มีบทบาทในชีวิตประจำวันอย่างมาก เนื่องจากอุปกรณ์อิเล็กทรอนิกส์มีความซับซ้อนและความละเอียดในการทำงานมากขึ้น ซึ่งในโครงงานนี้ได้เลือกใช้ Microprocessor มาใช้ในการควบคุมชุดลำเลียงและแยกขนาด โดยจะรับคำสั่งอินพุทจากการ์ดส่งข้อมูลอีกทีหนึ่ง ซึ่งจะแยกการทำงานออกเป็นดังนี้

5.4.1.1 ชุดลำเลียง

จะประกอบด้วยสเตปมอเตอร์ 1 ตัวเป็นตัวขับและ แผ่นเพลท 3 แผ่นซ้อนกันซึ่งเจาะช่องไว้แผ่นละ 2 ช่องทำมุมกัน 180 องศา การวางซ้อนกันแต่ละแผ่นต้องให้แนวช่องที่เจาะไว้ตั้งฉากกัน พร้อมทั้งมีที่กันลูกทำให้วัตถุตกตรงช่องที่ต้องการ ดังรูป



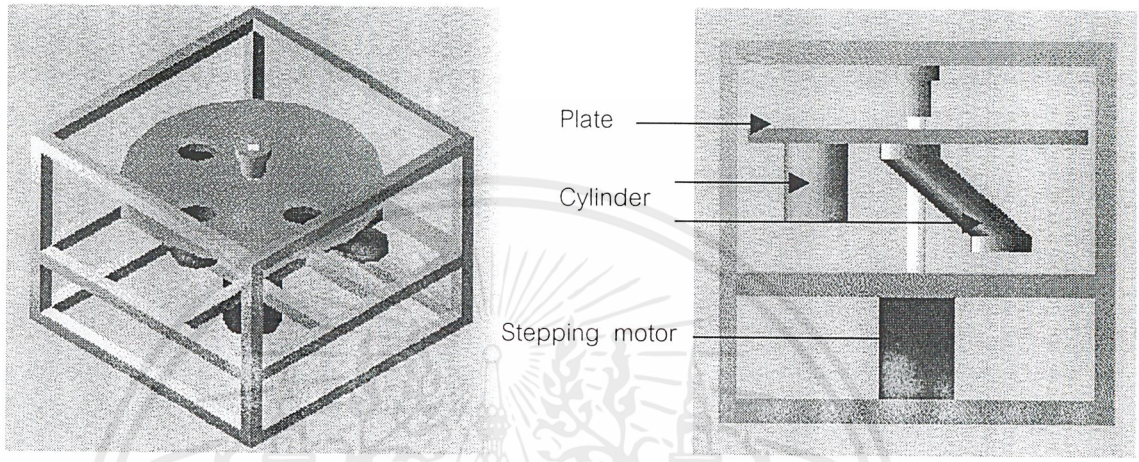
รูปที่ 5.9 โครงสร้างชุดลำเลียง

โดยเขียนโปรแกรม Microprocessor ควบคุมให้สเตปมอเตอร์หมุนไปสเตปละ 90 องศา เพื่อต้องการให้วัตถุตกลงไปที่ส่วนจับภาพทีละหนึ่งลูก โดยเมื่อเริ่มระบบชุดลำเลียงต้องทำงานเป็นลำดับแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.1.2 ชุดแยกขนาด

จะประกอบไปด้วยแผ่นเพลท 1 แผ่นซึ่งเจาะช่อง 3 ช่องแต่ละช่องทำมุมกัน 90 องศา เพื่อแยกขนาดออกเป็น 3 ขนาด แล้วส่งไปที่ภาชนะบรรจุ ดังรูป



รูปที่ 5.10 โครงสร้างของชุดแยกขนาด

เมื่อการหาขนาดและเปรียบเทียบเสร็จสิ้นแล้วสัญญาณควบคุมการหมุนจะถูกส่งจาก PC ผ่านการ์ดส่งข้อมูลและ Microprocessor โดยเขียนโปรแกรมให้แผ่นเพลทหมุนสเตปละ 90 , 270 องศา เมื่อหมุนเสร็จแล้วจะส่งข้อมูลออกไปควบคุมให้ชุดลำเลียงหมุนต่อไป

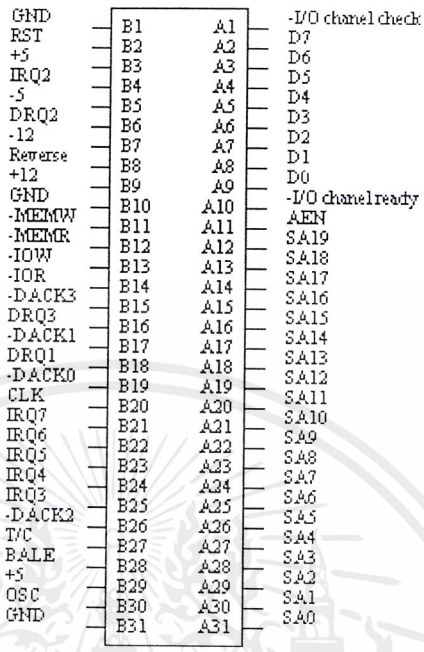
5.4.2 การ์ดส่งข้อมูลจาก PC

เนื่องจาก Parallel port ของ PC ถูกกล้อง quick cam ใช้งานไปแล้ว ดังนั้นจึงต้องหา port ใหม่เพื่อสามารถส่งข้อมูลออกไปควบคุมสเตปมอเตอร์ได้ โดยสร้างการ์ดชุดนี้ขึ้นมา

5.4.2.1 ISA slot

ที่ main board ของ PC จะมีช่องเสียบ card เอาไว้เพื่อเป็นการเชื่อมต่อระหว่าง PC กับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 แสดงขาของ slot แบบ62 ขา

สัญญาณที่ออกจาก PCเพื่อใช้กับการ์ดมีดังนี้

A2-A9 : เป็นแอดเดรสของระบบที่ใช้ติดต่อกับหน่วยความจำและอุปกรณ์อินพุท เอาท์พุท

D0-D7 : เป็นสัญญาณข้อมูลขนาด 8 บิตที่ใช้ติดต่อกับหน่วยความจำ Microprocessor

-IOW : เป็นสัญญาณเขียนข้อมูลลงบนอุปกรณ์อินพุท/เอาท์พุท

AEN : อินาเบิลแอดเดรส

ส่วนประกอบของการ์ดมีดังนี้

- ส่วนเปรียบเทียบแอดเดรส
- ส่วนรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2.2 หลักการทำงานของการ์ดส่งข้อมูลจาก PC

- ส่วนเปรียบเทียบแอดเดรส

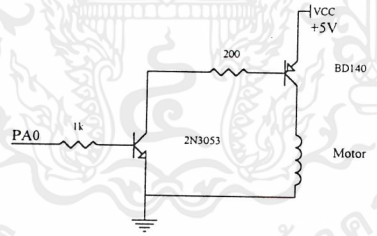
จะใช้ IC 74LS688 เป็นตัวเปรียบเทียบค่าแอดเดรส A2-A9 ของ PC กับค่าแอดเดรสที่ตั้งไว้ด้วย DIP-SW ว่ามีค่าตรงกันหรือไม่ ถ้าตรงกันก็จะให้สัญญาณ Enable ออกไปซึ่งตัว 74LS688 จะได้รับสัญญาณ AEN ของ PC โดยตรง

- ส่วนรับส่งข้อมูล

ใช้ IC 74LS373 เป็นตัว latch ข้อมูลจาก PC และส่งต่อไปให้กับ Microprocessor ทำหน้าควบคุมการหมุน,ตำแหน่งและทิศทางของสเตปมอเตอร์แล้วจึงส่งข้อมูลผ่านไปยังวงจรขับสเตปมอเตอร์โดย 74LS373 จะถูก Enable ด้วย output ของ NOR gate ซึ่งมี input คือ –IOW และ สัญญาณ Enable จาก 74LS688

5.4.3 วงจรขับสเตปมอเตอร์

ข้อมูล 4 บิตจะถูกส่งจาก Microprocessor ไปควบคุมสเตปมอเตอร์จะต่อเข้ากับขดลวด phase1-phase4 ของมอเตอร์ ผ่านวงจรขับโดยตรงเนื่องจาก แรงดันที่ใช้ขับมอเตอร์มีค่า 5V เท่ากัน วงจรขับจะประกอบไปด้วย transistor เบอร์ 2N3053 และ เบอร์ BD 140 ดังรูป



รูปที่ 5.12 วงจรขับสเตปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
ผลการทดสอบ

6.1 ขั้นตอนการทดลอง

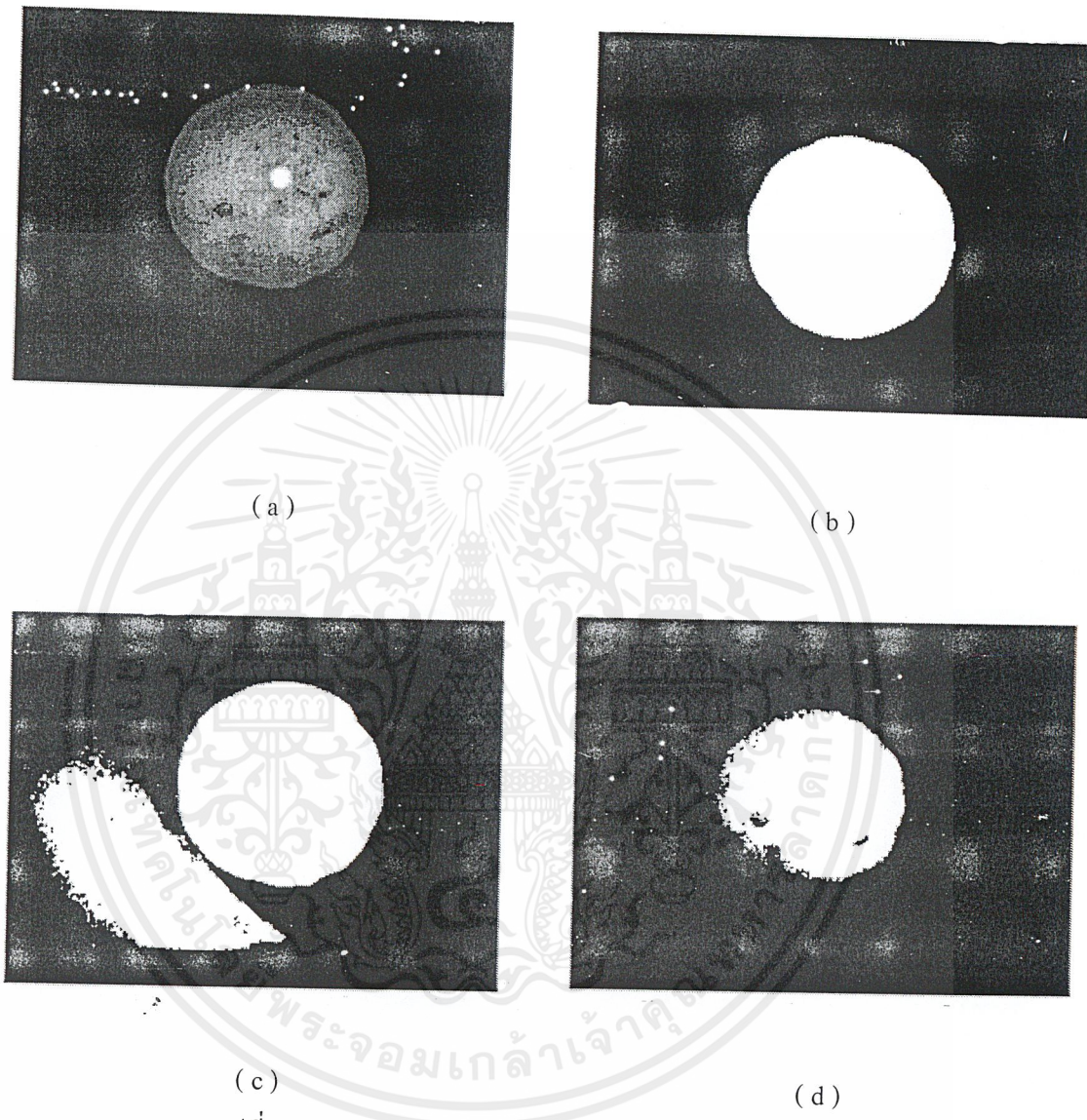
1. ติดตั้งกล้อง Color Quick Cam เข้ากับชุดโครงสร้าง
2. นำวัตถุที่ต้องการจะคัดขนาดวางบนแท่นโครงสร้าง
3. ทำการ Run โปรแกรมคัดขนาดวัตถุ
4. สังเกตขนาดวัตถุในหน้ากรอบภาพ ถ้ามีขนาดใหญ่เกินกรอบภาพต้องระดับกล้องให้สูงขึ้นเพื่อที่จะให้วัตถุไม่หลุดจากกรอบภาพ.
5. ทำการทดลองและสังเกตค่าที่ได้จากวัตถุต่างขนาดกัน

6.2 ผลการทดสอบการเตรียมข้อมูลภาพ

สำหรับการทดลอง จะศึกษาปัจจัยต่าง ๆ ที่มีผลต่อการทำงานของโปรแกรม เพื่อเป็นแนวทางในการปรับปรุงแก้ไขระบบต่อไป โดยจะใช้ภาพในรูปแบบบิตแมปทรูคัลเลอร์ 24 บิตขนาด 320x240 พิกเซล ดังรูปที่ 6.1(a) แล้วจึงแปลงเป็นภาพเกรย์สเกล 256 ระดับที่ขนาดภาพเดียวกัน

ในการทดลองเบื้องต้น จะทดลองใช้โปรแกรมเพื่อแยกวัตถุออกจากพื้นหลังโดยในขั้นการแปลงภาพเป็นภาพไบนารี จะใช้ค่าเทรชโฮลที่กำหนดขึ้นมา ดังรูปที่6.1(b) ซึ่งจากผลการทดลองการทำงานของโปรแกรมพบว่าขนาดวัตถุที่วัดได้มีขนาดที่คลาดเคลื่อนกับขนาดจริง เมื่อพิจารณาตัวอย่างในรูปที่ 6.1(c) พบว่าหลังจากที่แปลงภาพทรูคัลเลอร์เป็นภาพไบนารีแล้ว ขนาดของวัตถุที่วัดได้มีขนาดใหญ่กว่าขนาดจริง เนื่องจากการให้แสงเข้าไปที่ระบบมีค่าสูงเกินไป จึงทำให้พื้นหลังบางส่วนถูกมองว่าเป็นวัตถุ และในตัวอย่างที่รูปที่ 6.2 พบว่าขนาดวัตถุที่วัดได้มีขนาดที่เล็กกว่าขนาดจริง เนื่องจากการให้แสงเข้าไปที่ระบบมีค่าต่ำเกินไปจึงทำให้เนื้อที่วัตถุบางส่วนถูกมองว่าเป็นพื้นหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) (b)

(c) (d)

รูปที่ 6.1 (a) แสดงภาพ True color ของวัตถุ

(b) แสดงภาพที่ผ่านกระบวนการ Threshold

(c) แสดงภาพ Threshold ที่ให้แสงมากเกินไป

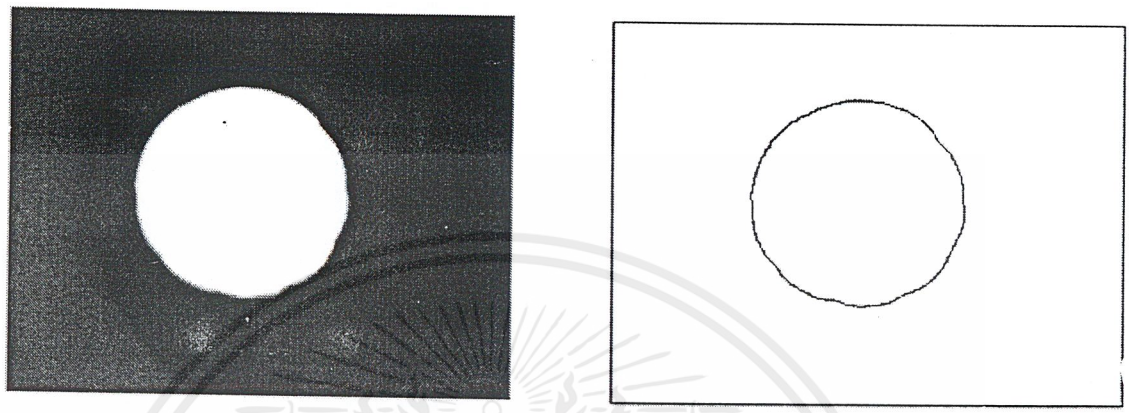
(d) แสดงภาพ Threshold ที่ให้แสงน้อยเกินไป

6.3 การทดลองส่วนการหาขอบและขนาดวัตถุ

หลังจากที่แปลงภาพเป็นภาพแบบไบนารีด้วยค่าเทรชโฮลที่เหมาะสมแล้ว จึงเข้าสู่กระบวนการหาขนาดของวัตถุ ด้วยวิธีการติดตามรอยขอบภาพ (Contour Technics) ซึ่งวิธีดังกล่าวจะทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

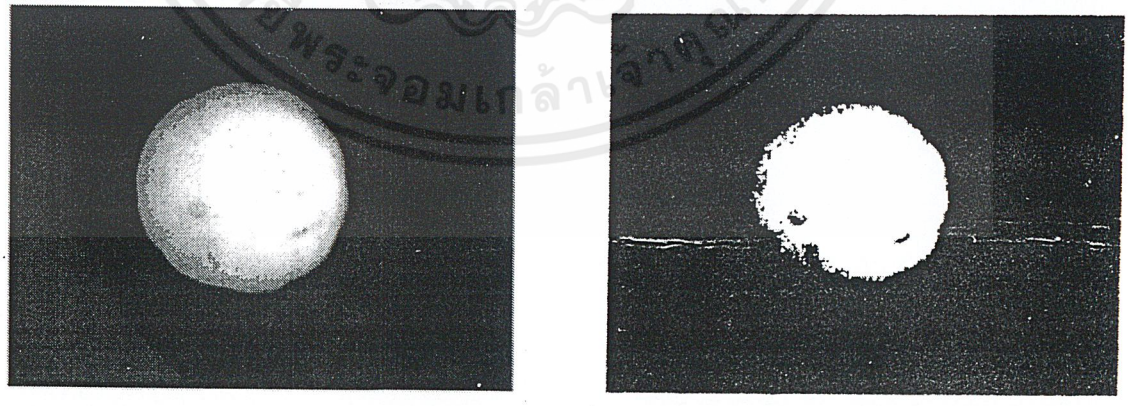
เราทราบว่ารอยขอบของวัตถุมีพิกัดต่ำสุดและสูงสุดในแนวแกน x และ y ที่ตำแหน่งใด แล้วนำมาหาผลต่างก็จะได้ขนาดเส้นผ่านศูนย์กลางของวัตถุแต่ขนาดที่ได้นี้จะเป็นหน่วยของพิกเซล ดังรูปที่ 6.2(b) และจะหาพื้นที่ของวัตถุจากพื้นที่ภายในของขอบวัตถุที่หาได้



(a) (b)

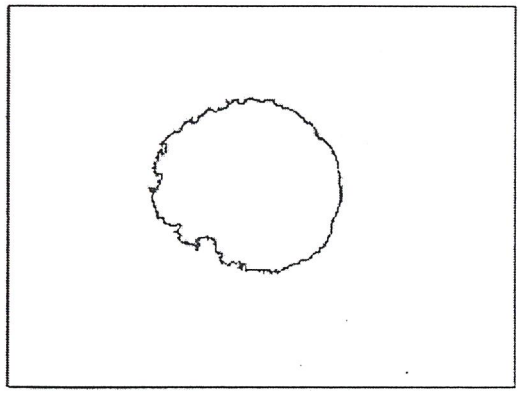
รูปที่ 6.2 (a) แสดงภาพจากการ Threshold
(b) แสดงภาพการหาขอบภาพวัตถุด้วย Contouring Technic

และจากการทำงานต่อเนื่องจากกระบวนการ Threshold ดังนั้น เมื่อเกิดการผิดพลาดจากการ Threshold แล้ว จะทำให้การติดตามหาขอบวัตถุเกิดการผิดพลาดด้วย ดังรูปที่ 6.3



(a) (b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(c)

รูปที่ 6.3 (a) แสดงภาพ True color ที่ให้แสงน้อยเกินไป
(b) แสดงภาพที่ได้จากการ Threshold
(c) แสดงภาพที่ได้จากการ Contouring Technic

6.4 ผลการทดลองเก็บสถิติเพื่อหาค่าขนาดมาตรฐาน

หลังจากผ่านขั้นตอนของการติดตามรอยขอบภาพ (contouring technics) แล้วจะต้องแบ่งขนาดออกเป็น 3 ขนาดมาตรฐานโดยในโครงงานนี้จะแบ่งออกเป็น 3 ขนาด คือ ขนาด A,B,C ซึ่งเกณฑ์ในการแบ่งขนาดจะใช้การเก็บข้อมูลจากลูกมะนาวที่คัดขนาดเอาไว้แล้ว

ลูกที่	ขนาด A (pixel)	ขนาด B (pixel)	ขนาด C (pixel)
1	12,115	16,263	18,685
2	12,517	17,181	18,049
3	10,883	17,903	19,993
4	10,197	16,879	18,746
5	13,109	17,657	20,051
6	12,047	16,779	18,492
7	11,143	15,059	19,031
8	13,247	15,037	18,092
9	12,011	15,021	18,745
10	13,946	16,059	20,357

ตารางที่ 6.1 แสดงค่าของวัตถุขนาดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองจะกำหนดเกณฑ์ในการวัดขนาดดังนี้

ขนาด A : 0 pixel - 14,000 pixel

ขนาด B : 14,000 pixel - 18,000 pixel

ขนาด C : 18,000 pixel - 32,000 pixel

6.5 การทดสอบประสิทธิภาพของเครื่องคัดขนาด

การทดสอบหาประสิทธิภาพของเครื่องคัดขนาด จะวัดจากคุณสมบัติ 2 อย่าง คือ

1. ความเร็วในการคัดแยกต่อ 1 ลูก
2. ความแม่นยำในการประมวลผลเพื่อหาขนาดของวัตถุ

6.5.1 ความเร็วในการคัดแยกขนาดต่อ 1 ลูก

ความเร็วในการคัดแยกขนาดต่อ 1 ลูก จะมีค่าไม่เท่ากันเนื่องจากตำแหน่งของแผ่นเพลททั้งของชุดลำเลียงและชุดแยกขนาด อาจจะไม่ได้อยู่ในตำแหน่งเดียวกันตลอดเวลา ดังนั้นค่าเวลาในการคัดแยกขนาดของแต่ละลูกจะมีค่าที่ไม่เท่ากัน โดยในการทดลองนี้จะวัดค่าเฉลี่ยในการคัดแยกต่อ 1 ลูก จากสมการดังนี้

$$\text{เวลาเฉลี่ยที่ใช้ในการแยกขนาด} = \frac{\text{เวลาที่ใช้มากที่สุดในการคัดแยก} + \text{เวลาที่ใช้น้อยสุดในการคัดแยก}}{2}$$

จากการทดลองจับเวลาจะได้ค่าดังนี้

เวลาที่ใช้ในการคัดแยกต่อ 1 ลูกมากที่สุด = 10.54 วินาที

เวลาที่ใช้ในการคัดแยกต่อ 1 ลูกน้อยที่สุด = 8.87 วินาที

$$\therefore \text{ เวลาเฉลี่ยที่ใช้ในการคัดแยก } = \frac{10.54 + 8.87}{2}$$

$$= 9.705 \text{ วินาที}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5.2 ความแม่นยำในการประมวลผลเพื่อหาขนาดของวัตถุ

ความแม่นยำในการประมวลผลเพื่อหาขนาดของวัตถุนี้วัดจาก การจับภาพจากวัตถุชิ้นเดียวกันด้วยความเข้มแสงที่คงที่ แล้วตรวจดูว่าค่าขนาดที่ได้มีการเปลี่ยนแปลงอย่างไร ซึ่งผลการทดลองที่ได้เป็นดังนี้

ครั้งที่	ขนาดวัตถุที่ได้ (pixel)	ครั้งที่	ขนาดของวัตถุที่ได้ (pixel)
1	17,265	9	17,059
2	17,151	10	17,019
3	17,135	11	17,103
4	17,159	12	17,139
5	17,089	13	17,013
6	17,167	14	17,139
7	17,147	15	17,037
8	17,047	16	17,017

ตารางที่ 6.2 แสดงค่าเฉลี่ยของขนาดวัตถุชิ้นเดียวกันในเวลาต่างๆ

จากตารางค่าเฉลี่ยของขนาดวัตถุ $= \dfrac{\sum \text{ขนาดของวัตถุ (n)}}{16}$

$= 17{,}105.375$ pixel

ค่าความเปลี่ยนแปลงของข้อมูล = ขนาดวัตถุ (max) – ขนาดวัตถุ (min)

$= 17{,}265 - 17{,}013$

$= 252$ pixel

$\therefore$ ค่าเปอร์เซ็นต์ความผิดพลาด $= \dfrac{\text{ค่าความเปลี่ยนแปลงของข้อมูล}}{\text{ค่าเฉลี่ยขนาดของวัตถุ}}$

$= \dfrac{252 \times 100}{17{,}105.375}$

$= 1.47\ \%$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปและวิจารณ์

7.1 สรุปและวิจารณ์ผลการทดสอบ

จากการทดลอง จะเห็นว่าปัจจัยที่มีผลกระทบต่อการทำงานของระบบ ซึ่งทำให้เกิดข้อผิดพลาดในการทำงาน และได้ข้อจำกัดในการใช้งานระบบโดยระบบจะทำงานได้ไม่สมบูรณ์ถ้าข้อมูลภาพมีลักษณะดังนี้

- แสงจากภายนอกระบบที่มีความเข้มสูง เข้ามารบกวนระบบ
- การติดตั้งกล้องจะต้องได้มุม และระดับไม่อยู่ที่ตำแหน่งเดียวกัน
- การให้แสงภายในระบบมีค่าไม่เหมาะสม
- มีส่วนใดส่วนหนึ่งของวัตถุหลุดออกจากกรอบภาพ

ดังนั้นในการปฏิบัติจะต้องนำข้อจำกัดเหล่านี้มาเป็นสิ่งสำคัญในการออกแบบระบบที่จะพัฒนาต่อไป นอกจากนี้ในขั้นตอนของการแปลงภาพสีจริงเป็นภาพไบนารี ด้วยวิธีการเลือกค่าเทรชโฮลที่เหมาะสม จะพบว่าการใช้ค่าเทรชโฮลที่ได้จากการหาค่าเฉลี่ยเลขคณิตมีความถูกต้องของผลลัพธ์มากกว่าการใช้ค่าเทรชโฮลจากค่ามัธยฐาน หรือค่ากลางของข้อมูล (= 128) ซึ่งในการใช้งานจริงควรที่จะเก็บสถิติ เพื่อที่จะสามารถหาค่าเฉลี่ยเลขคณิตที่เหมาะสมได้

จากผลการทดลองจะพบว่าแสงที่ฉายให้แก่ระบบ เป็นปัจจัยหลักที่จะทำให้เกิดความผิดพลาด มากที่สุด ซึ่งสังเกตได้จาก ขนาดพื้นที่ของภาพที่หาได้เมื่อ ระดับความเข้มแสงแตกต่างกัน ดังนั้นในการทำงานจริงจะต้องมีการควบคุมระบบ และ ป้องกันไม่ให้แสงจากภายนอก มีผลกระทบมากกว่าแสงของระบบ

นอกจากนี้ระบบการคัดขนาดวัตถุ จะเกิดการผิดพลาดขึ้นได้ง่าย ที่ค่าขนาดวัตถุมีค่าใกล้เคียงกันระหว่างค่ามาตรฐาน เช่น ค่าขนาดของ ขนาด B ซึ่งมีค่าตั้งแต่ 14,000 – 18,000 pixel ถ้าวัตถุมีขนาดที่แท้จริงเท่ากับ 18,050 pixel แต่เกิดความผิดพลาดจากการให้แสง ทำให้ขนาดที่วัดได้เป็น 17,980 pixel จะทำให้การคัดขนาดวัตถุผิดพลาด จากขนาด C เป็น B

ส่วนประสิทธิภาพของเครื่องคัดขนาด ถ้าพิจารณาในแง่ของความแม่นยำในการวัดหาขนาดของวัตถุ นับว่ามีประสิทธิภาพที่ดี เนื่องจากเกิดความผิดพลาดเพียง 1.47 % เท่านั้น แต่ถ้าในเรื่องของความเร็วในการคัดแยกต่อ 1 ลูก แล้วจะพบว่าใช้เวลาประมาณ 9.7 วินาที ซึ่งถือว่ามากเกินไป (ในระบบอุตสาหกรรมจริงใช้เวลาไม่เกิน 5 วินาที)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 สิ่งที่ได้ดำเนินการในภาคเรียนที่ 1

1. ส่วนของ software ประมวลผลภาพ
   - เขียนโปรแกรมการรับภาพจากกล้อง
   - เขียนโปรแกรมการประมวลผลภาพด้วย image processing
     โปรแกรม brightness & threshold เพื่อ detect วัตถุออกจากภาพ
     โปรแกรม contour technics เพื่อ detect วัตถุออกจากสิ่งรบกวนจากภายนอก
     โปรแกรมการหาขนาดวัตถุ
2. ส่วนของการควบคุมด้วย microprocessor
   - ศึกษาการประยุกต์ใช้งาน microprocessor 8051
   - ศึกษาการ interface ระหว่าง PC กับอุปกรณ์ภายนอก

7.3 สิ่งที่ได้ดำเนินการในภาคเรียนที่ 2

1. ส่วนของ hardware
   - ชุดลำเลียง
   - ชุดแยกขนาด
   - วงจร microprocessor 8051
   - วงจรขับสเตปมอเตอร์
   - การ์ดส่งข้อมูลจาก PC
2. ส่วนของ software ควบคุม microprocessor 8051
   - โปรแกรมควบคุมสเตปมอเตอร์ของชุดลำเลียงและชุดแยกขนาด

7.4 แนวทางการปรับปรุงและแก้ไขเครื่องคัดขนาดวัตถุ

1. ขนาดของมอเตอร์ที่ใช้ เนื่องจากมอเตอร์ที่ใช้มีขนาดเล็ก เมื่อเพิ่มความเร็วให้สูงเกินไป จะทำให้แรงขับไม่พอ ดังนั้นถ้าเปลี่ยนขนาดมอเตอร์ให้ใหญ่ กว่านี้จะสามารถได้แรงขับเพียงพอที่ความเร็วสูงได้

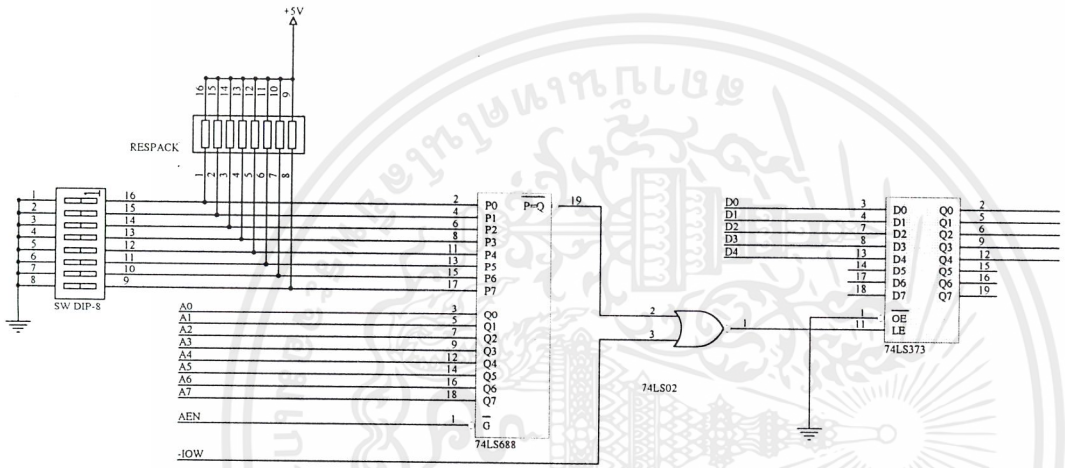
2. ความเร็วในการจับภาพของกล้อง จากคุณสมบัติของกล้องที่ใช้พบว่า จะเกิดการหน่วงของสัญญาณภาพมาก ถ้าหาก CPU ของ PC ที่ใช้มีความเร็วต่ำ ดังนั้นในการเพิ่มความเร็วกล้องอาจทำได้ 2 วิธี คือ

   - เพิ่มความเร็ว CPU
   - ใช้กล้องที่มีความเร็วสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

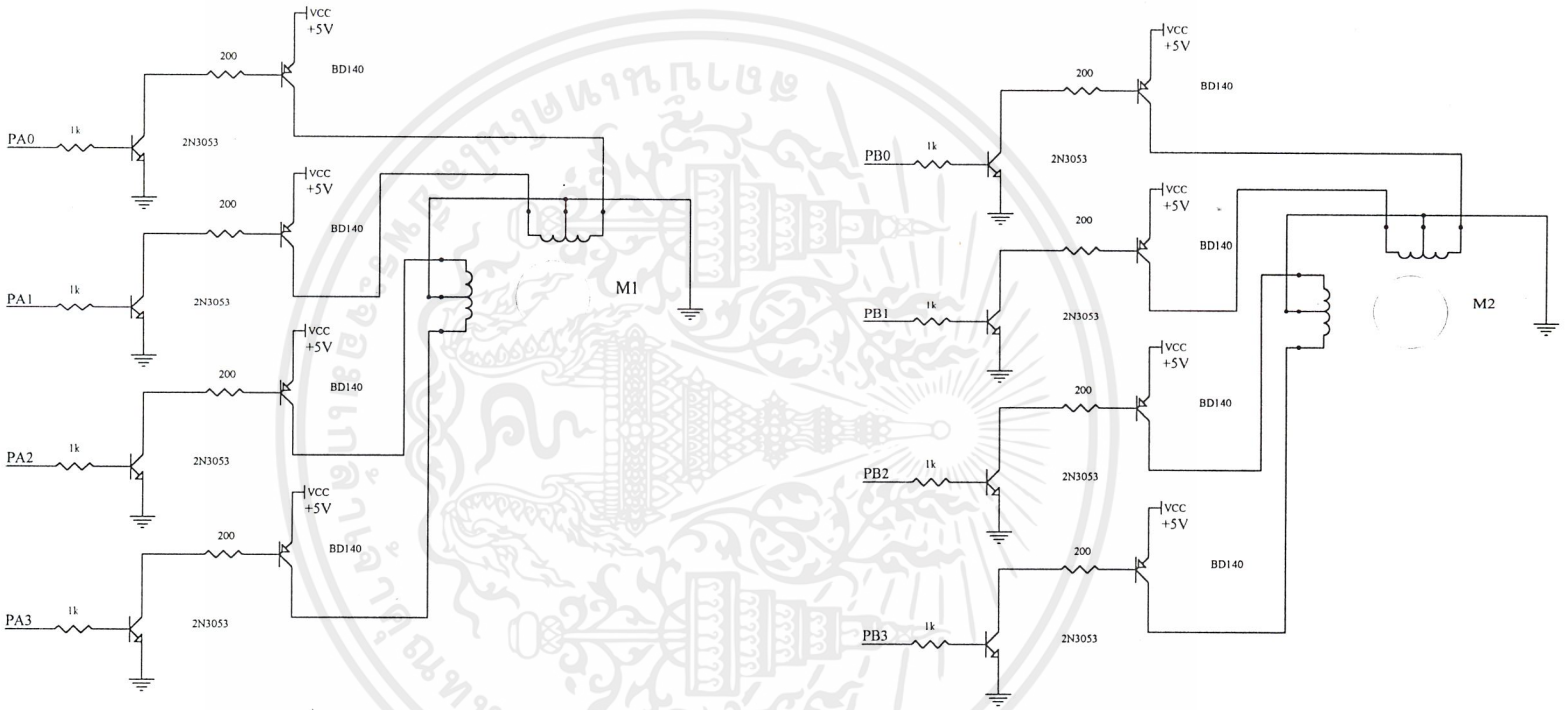
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



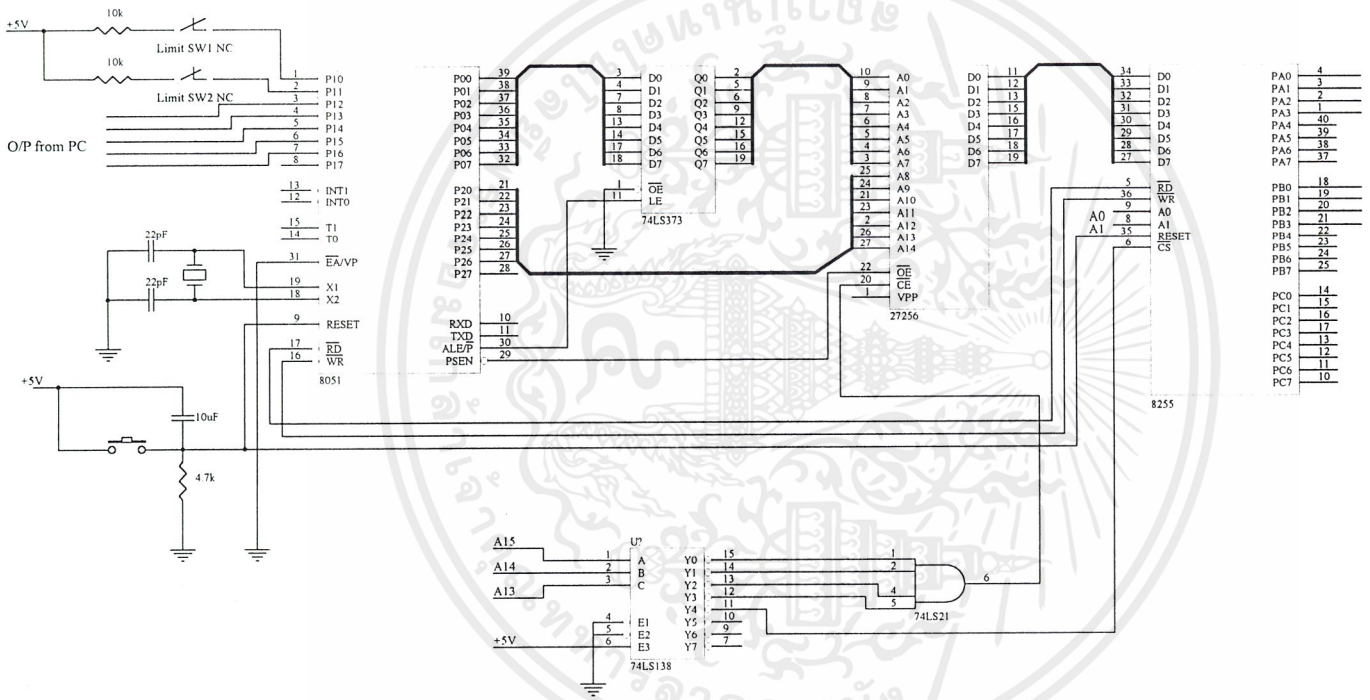
TO Micropro

Title	CARD INTERFACE CCT		
Size B	Number		Revision
Date:	28-Mar-2000	Sheet of	
File:	A:\CARDIN.SCH	Drawn By:	



Title	DRIVE STEPING MOTOR CCT		
Size: B	Number		Revision
Date: 28-Mar-2000		Sheet of	
File: A:\DRVLEMON.SCH		Drawn By:	

VCC



Title	MICROPROCESSOR 8051 CCT		
Size B A	Number		Revision
Date:	28-Mar-2000	Sheet of	
File:	A:\LEMONPRO.SCH	Drawn By:	

```
// s_ccd3View.cpp : implementation of the CS_ccd3View class
//
#include "stdafx.h"
#include "s_ccd3.h"
#include "s_ccd3Doc.h"
#include "s_ccd3View.h"
#include "Conop.h"                                    ///
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

int       Width;
int Height;
int sw=1;
unsigned char Lightness;
int x,y;
int dx,dy,total;
int x1,y1;
int x2,y2;
int       st;
int st1,st2;
int dimx,dimy;
int xmax,xmin,ymax,ymin;

BYTE Red;
BYTE Green;
BYTE Blue;
int Theshold[320][240];
int       Getborder[320][240];
int       Getborder2[320][240];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// s_ccd3View.cpp : implementation of the CS_ccd3View class
//

#include "stdafx.h"
#include "s_ccd3.h"
#include "s_ccd3Doc.h"
#include "s_ccd3View.h"
#include "Conop.h"
#include "conio.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

int         Width;
int Height;
int sw=1;
unsigned char Lightness;
int x,y;
int dx,dy,total,kind;
int x1,y1;
int x2,y2;
int         st;
int st1,st2;
int dimx,dimy;
int xmax,xmin,ymax,ymin;
int stopbit;
int tim_st;



BYTE Red;
BYTE Green;
BYTE Blue;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int Theshold[320][240];
int        Getborder[320][240];
int        Getborder2[320][240];


/////////////////////////////////////////////////////////////////////////////
// CS_ccd3View

IMPLEMENT_DYNCREATE(CS_ccd3View, CView)

BEGIN_MESSAGE_MAP(CS_ccd3View, CView)
        //{{AFX_MSG_MAP(CS_ccd3View)
        ON_COMMAND(run, Onrun)
        ON_COMMAND(red, Onred)
        ON_COMMAND(theshold, Ontheshold)
        ON_COMMAND(contour, Oncontour)
        ON_COMMAND(org, Onorg)
        ON_COMMAND(flow, Onflow)
        ON_WM_TIMER()
        ON_COMMAND(automate, Onautomate)
        ON_COMMAND(stop, Onstop)
        ON_COMMAND(initial, Oninitial)
        ON_COMMAND(size_A, OnA)
        ON_COMMAND(size_B, OnB)
        ON_COMMAND(size_C, OnC)
        ON_COMMAND(turn_up180, Onup180)
        ON_COMMAND(clear, Onclear)
        ON_COMMAND(initial1, Oninitial1)
        ON_COMMAND(start1, Onstart1)
        ON_COMMAND(clear1, Onclear1)
        //}}AFX_MSG_MAP
        // Standard printing commands
        ON_COMMAND(ID_FILE_PRINT, CView::OnFilePrint)
        ON_COMMAND(ID_FILE_PRINT_DIRECT, CView::OnFilePrint)
        ON_COMMAND(ID_FILE_PRINT_PREVIEW, CView::OnFilePrintPreview)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CS_ccd3View construction/destruction

CS_ccd3View::CS_ccd3View()
{
        // TODO: add construction code here

}

CS_ccd3View::~CS_ccd3View()
{
}

BOOL CS_ccd3View::PreCreateWindow(CREATESTRUCT& cs)
{
        // TODO: Modify the Window class or styles here by modifying
        //  the CREATESTRUCT cs

        return CView::PreCreateWindow(cs);
}

/////////////////////////////////////////////////////////////////////////////
// CS_ccd3View drawing

void CS_ccd3View::OnDraw(CDC* pDC)
{
        BMP->DisPlay(pDC,CPoint(400,10));
        BMP2->DisPlay(pDC,CPoint(400,260));

        CS_ccd3Doc* pDoc = GetDocument();
        ASSERT_VALID(pDoc);

        pDC->TextOut(50,300,"    DIMETRES OF OBJECT ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CString  str1,str2,str3,str4;
        str1.Format("%d",dimx);
        str2.Format("%d",dimy);
        str3.Format("%d",total);
        str4.Format("%d",kind);


        pDC->TextOut(70,330,"  Xmax      =          Pixel");
        pDC->TextOut(160,330,str1);
        pDC->TextOut(70,350,"  Ymax      =          Pixel");
        pDC->TextOut(160,350,str2);
        pDC->TextOut(70,370," Total Area  =          Pixel");
        pDC->TextOut(160,370,str3);
        pDC->TextOut(70,400,"   SIZE      =         TYPE");
        pDC->TextOut(160,400,str4);


        // TODO: add draw code for native data here
}

/////////////////////////////////////////////////////////////////////////////
// CS_ccd3View printing

BOOL CS_ccd3View::OnPreparePrinting(CPrintInfo* pInfo)
{
        // default preparation
        return DoPreparePrinting(pInfo);
}

void CS_ccd3View::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
        // TODO: add extra initialization before printing
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void CS_ccd3View::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*pInfo*/)
{
	// TODO: add cleanup after printing
}

/////////////////////////////////////////////////////////////////////////////
// CS_ccd3View diagnostics

#ifdef _DEBUG
void CS_ccd3View::AssertValid() const
{
	CView::AssertValid();
}

void CS_ccd3View::Dump(CDumpContext& dc) const
{
	CView::Dump(dc);
}

CS_ccd3Doc* CS_ccd3View::GetDocument() // non-debug version is inline
{
	ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CS_ccd3Doc)));
	return (CS_ccd3Doc*)m_pDocument;
}
#endif //_DEBUG

/////////////////////////////////////////////////////////////////////////////
// CS_ccd3View message handlers

void CS_ccd3View::OnInitialUpdate()
{
	CView::OnInitialUpdate();

	// TODO: Add your specialized code here and/or call the base class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
VidCap.SetView((CView*)this);
VidCap.Create(WS_CHILD|WS_VISIBLE,CRect(50,10,0,0),this);
VidCap.DriverConnect(0);
VidCap.AutoSize();
VidCap.Preview(TRUE);
VidCap.SetCallbackOnFrame(FALSE);
dimx=0;
dimy=0;
total=0;
kind=0;
stopbit=0;
tim_st=0;

LPBITMAPINFO Picinfo;
Picinfo = VidCap.GetVideoFormat();
long Size;
Width = Picinfo->bmiHeader.biWidth;
Height = Picinfo->bmiHeader.biHeight;
WORD BitPerPixel = Picinfo->bmiHeader.biBitCount;

//      switch(BitPerPixel)

            BMP=(CDib*) new CDibTrue(Width,Height);
            BMP2=(CDib*) new CDibTrue(Width,Height);

            Size = sizeof(BITMAPINFOHEADER);
}


      void Myvidcap::OnFrameCallback(LPVIDEOHDR lpvidhdr)
      {
        SetCallbackOnFrame(FALSE);
        Preview(FALSE);
        CDib* BMP=((CS_ccd3View*)View)->GetBitmap();
        CDib* BMP2=((CS_ccd3View*)View)->GetBitmap2();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char* Des=BMP->ReceivePic();
unsigned char* Des2=BMP2->ReceivePic();
unsigned char* Src=lpvidhdr->lpData;
unsigned char* SrcOld=Des;
unsigned char* SrcOld2=Des2;

long Size;
if(BMP->It24Bit())
        Size=BMP->ReceiveBMPInfoHeader()->biSizeImage;
else
        Size=BMP->ReceiveWidth()*BMP->ReceiveHeight();


switch(sw)
{
case 1:
      // copy  image
       int i,j;
            for(i=0;i<Size;i++)
            {
              *Des=*Src;
                  Des++;
                  Src++;
            }
            Des=SrcOld;
            break;
case 2:

        Red=255;
            for(i=0;i<(Size/3);i++)
            {
                  *Des=*Src;
                  Des++;
                  Src++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                *Des=*Src;
                        Des++;
                        Src++;
                *Des=Red;

                        Des++;
                        Src++;
                }
        Des=SrcOld;
                break;

case 3:                                                         //threshold
        x=0;
                y=0;
                BYTE color3;

                for(i=0;i<(Size/3);i++)
                {
                        *Des=*Src;
                        Blue=*Des;
                        Des++;Src++;

                        *Des=*Src;
                        Green=*Des;
                        Des++;Src++;

                        *Des=*Src;
                        Red=*Des;
                        Des++;Src++;

                        Lightness=((Red*7)+(Green*2)+(Blue*5))/14;
                        if ((Lightness>=0)&&(Lightness<200))
                        {       color3=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Des--;Src--;*Des=color3;
                    Des--;Src--;*Des=color3;
                    Des--;Src--;*Des=color3;
                    Theshold[x][y]=1;          // black pixel

                    Des=Des+3;
                    Src=Src+3;
                }
                else
                {
                    color3=255;
                    Des--;Src--;*Des=color3;
                    Des--;Src--;*Des=color3;
                    Des--;Src--;*Des=color3;
                    Theshold[x][y]=0;
//white pixel
                    Des=Des+3;
                    Src=Src+3;
                }
                x++;
                if(x>319)
                {
                  y--;
                  x=0;
                }
            }

//              total=0;
//              for(y=0;y<240;y++)
//              {
//                  for(x=0;x<320;x++)
//                  {
//                      if(Theshold[x][y]==0)
//                      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//                                                  total=total+1;
//                                              }
//                                          }
//                                  }

                                    Des=SrcOld;

                                    break;


                        case 4:                                                 //contour
                                    int l;
                                    int b;
                                    int empty;
                                    BYTE color4;
                                    for(i=0;i<320;i++)
                                    {
                                            for(j=0;j<240;j++){Getborder[i][j]=1;}
                                    }
                                    for(i=1;i<319;i++)
                                    {
                                            for(j=1;j<239;j++){Getborder[i][j]=0;}
                                    }


                                    b=0;empty=0;
                                    x1=0;y1=0;
                                    for(y=2;y<238;y++)
                                    {
                                            for(x=2;x<318;x++)
                                            {
                                                    if((Theshold[x][y]==0)&&(b==0))
                                                    {
                                                            st=3;
                                                            Getborder[x][y]=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        x1=x;y1=y;
                                        b=1;
                                }
                        }
                }

                //******* add*******
                if((x1==0)&&(y1==0))
                {
                        for(y=2;y<238;y++)
                        {
                                for(x=2;x<318;x++){Getborder[x][y]=0;}
                        }

                        empty=1;
                        goto e_contour;
                }


                //******* end add *******
                x=x1;
                y=y1+1;
                b=0;
                while (b==0)
                {
s_contour:
                        if(st==1)
                        {
                                if((x==x1)&&(y==y1)){b=1;goto e_contour;}
                                if(Theshold[x][y]==0){Getborder[x]
[y]=1;x++;st=2;goto s_contour;}
                                if(Theshold[x][y]==1){x--;st=4;goto
s_contour;}
                                goto s_contour;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                    if(st==2)
                    {
                        if((x==x1)&&(y==y1)){b=1;goto e_contour;}
                        if(Theshold[x][y]==0){Getborder[x]
[y]=1;y++;st=3;goto s_contour;}
                        if(Theshold[x][y]==1){y--;st=1;goto
s_contour;}
                        goto s_contour;
                    }
                    if(st==3)
                    {
                        if((x==x1)&&(y==y1)){b=1;goto e_contour;}
                        if(Theshold[x][y]==0){Getborder[x][y]=1;x--
;st=4;goto s_contour;}
                        if(Theshold[x][y]==1){x++;st=2;goto
s_contour;}
                        goto s_contour;
                    }
                    if(st==4)
                    {
                        if((x==x1)&&(y==y1)){b=1;goto e_contour;}
                        if(Theshold[x][y]==0){Getborder[x][y]=1;y--
;st=1;goto s_contour;}
                        if(Theshold[x][y]==1){y++;st=3;goto
s_contour;}
                        goto s_contour;
                    }

                }
e_contour:
                x=0;
                y=239;
                for(I=0;I<(Size/3);I++)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(Getborder[x][y]==0){color4=255;}
            if(Getborder[x][y]==1){color4=0;}

            *Des2=color4;Des2++;
            *Des2=color4;Des2++;
            *Des2=color4;Des2++;

            x++;
            if(x>319){y--;x=0;}
}
xmax=0;xmin=320;ymax=0;ymin=240;
for(y=2;y<238;y++)
{
        for(x=2;x<318;x++)
        {
            if(Getborder[x][y]==1)
            {
                if(x<xmin){xmin=x;}
                if(x>xmax){xmax=x;}
                if(y<ymin){ymin=y;}
                if(y>ymax){ymax=y;}
            }
        }
}
dimx=xmax-xmin;
dimy=ymax-ymin;

if(empty==1){dimx=0;dimy=0;}



st1=0;st2=0;
total=0;
for(y=ymin;y<ymax;y++)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                for(x=2;x<318;x++)
                {
                        if((Getborder[x][y]==1)&&(st1==0)){x1=x-
1;st1=1;}
                        if((Getborder[320-x][y]==1)&&(st2==0))
{x2=320-x+1;st2=1;}
                }
                dx=x2-x1;

                st1=0;st2=0;
                total=total+dx;
        }

        kind=0;
        if((total>=0)&&(total<14000))
        {
                kind=1;
                _outp(0x275,0x01);
                Sleep(70);
                _outp(0x275,0x00);
        }
        if((total>14000)&&(total<18000))
        {
                kind=2;
                _outp(0x275,0x02);
                Sleep(70);
                _outp(0x275,0x00);

        }
        if((total>18000)&&(total<50000))
        {
                kind=3;
                _outp(0x275,0x04);
                Sleep(70);
                _outp(0x275,0x00);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

Preview(TRUE);

Des2=SrcOld2;
break;

case 5:

Theshold[100][0]=1;
Theshold[100][1]=1;
Theshold[100][2]=1;
Theshold[100][3]=1;
Theshold[100][4]=1;
Theshold[100][5]=1;
Theshold[100][6]=1;
int x3,y3;
BYTE color;
x3=0;
y3=239;
for(I=0;I<(Size/3);I++)
{
                if(Theshold[x3][y3]==1){color=0;}
                if(Theshold[x3][y3]==0){color=255;}
                *Des=color;Des++;

                *Des=color;Des++;
                *Des=color;Des++;
        x3++;
        if(x3>319){y3--;x3=0;}
}
Des=SrcOld;
break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 6:
        //************ start  theshold part **************
          x=0;
                y=0;
                BYTE color3_ca6;



                for(i=0;i<(Size/3);i++)
                {
                        *Des=*Src;
                        Blue=*Des;
                        Des++;Src++;

                        *Des=*Src;
                        Green=*Des;
                        Des++;Src++;

                        *Des=*Src;
                        Red=*Des;
                        Des++;Src++;

                        Lightness=((Red*7)+(Green*2)+(Blue*5))/14;
                        if ((Lightness>=0)&&(Lightness<200))
                        {       color3_ca6=0;
                                Des--;Src--;*Des=color3_ca6;
                                Des--;Src--;*Des=color3_ca6;
                                Des--;Src--;*Des=color3_ca6;
                                Theshold[x][y]=1;       // black pixel

                                Des=Des+3;
                                Src=Src+3;
                        }
                        else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        color3_ca6=255;
                        Des--;Src--;*Des=color3_ca6;
                        Des--;Src--;*Des=color3_ca6;
                        Des--;Src--;*Des=color3_ca6;
                        Theshold[x][y]=0;
//white pixel
                        Des=Des+3;
                        Src=Src+3;
                }
                x++;
                if(x>319)
                {
                 y--;
                 x=0;
                }
               }
                Des=SrcOld;
 //************  end theshold part ***************
 //************  start contour part  *************
                int l_ca6;
                int b_ca6;
                int empty_ca6;
                BYTE color4_ca6;
                for(i=0;i<320;i++)
                {
                        for(j=0;j<240;j++){Getborder[i][j]=1;}
                }
                for(i=1;i<319;i++)
                {
                        for(j=1;j<239;j++){Getborder[i][j]=0;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}


b_ca6=0;empty_ca6=0;
x1=0;y1=0;
for(y=2;y<238;y++)
{
        for(x=2;x<318;x++)
        {
                if((Theshold[x][y]==0)&&(b_ca6==0))
                {
                        st=3;
                        Getborder[x][y]=1;
                        x1=x;y1=y;
                        b_ca6=1;
                }
        }
}
//******* add*******
if((x1==0)&&(y1==0))
{
        for(y=2;y<238;y++)
        {
                for(x=2;x<318;x++){Getborder[x][y]=0;}
        }

        empty_ca6=1;
        goto e1_contour;
}


//******* end add *******
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    x=x1;
                    y=y1+1;
                    b_ca6=0;
                    while (b_ca6==0)
                    {
s1_contour:
                        if(st==1)
                        {
                            if((x==x1)&&(y==y1)){b_ca6=1;goto
e1_contour;}
                            if(Theshold[x][y]==0){Getborder[x]
[y]=1;x++;st=2;goto s1_contour;}
                            if(Theshold[x][y]==1){x--;st=4;goto
s1_contour;}
                            goto s1_contour;
                        }
                        if(st==2)
                        {
                            if((x==x1)&&(y==y1)){b_ca6=1;goto
e1_contour;}
                            if(Theshold[x][y]==0){Getborder[x]
[y]=1;y++;st=3;goto s1_contour;}
                            if(Theshold[x][y]==1){y--;st=1;goto
s1_contour;}
                            goto s1_contour;
                        }
                        if(st==3)
                        {
                            if((x==x1)&&(y==y1)){b_ca6=1;goto
e1_contour;}
                            if(Theshold[x][y]==0){Getborder[x][y]=1;x--
;st=4;goto s1_contour;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    if(Theshold[x][y]==1){x++;st=2;goto
s1_contour;}
                    goto s1_contour;
                }
                if(st==4)
                {
                    if((x==x1)&&(y==y1)){b_ca6=1;goto
e1_contour;}
                    if(Theshold[x][y]==0){Getborder[x][y]=1;y--
;st=1;goto s1_contour;}
                    if(Theshold[x][y]==1){y++;st=3;goto
s1_contour;}
                    goto s1_contour;
                }

            }
e1_contour:
            x=0;
            y=239;
            for(l_ca6=0;l_ca6<(Size/3);l_ca6++)
            {
                if(Getborder[x][y]==0){color4_ca6=255;}
                if(Getborder[x][y]==1){color4_ca6=0;}

                *Des2=color4_ca6;Des2++;
                *Des2=color4_ca6;Des2++;
                *Des2=color4_ca6;Des2++;

                x++;
                if(x>319){y--;x=0;}
            }
            xmax=0;xmin=320;ymax=0;ymin=240;
            for(y=2;y<238;y++)
            {
                for(x=2;x<318;x++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        if(Getborder[x][y]==1)
                        {
                                if(x<xmin){xmin=x;}
                                if(x>xmax){xmax=x;}
                                if(y<ymin){ymin=y;}
                                if(y>ymax){ymax=y;}
                        }
                }
        }
        dimx=xmax-xmin;
        dimy=ymax-ymin;
        if(empty_ca6==1){dimx=0;dimy=0;}


        st1=0;st2=0;
        total=0;
        for(y=ymin;y<ymax;y++)
        {
                for(x=2;x<318;x++)
                {
                        if((Getborder[x][y]==1)&&(st1==0)){x1=x-
1;st1=1;}

                        if((Getborder[320-x][y]==1)&&(st2==0))
{x2=320-x+1;st2=1;}
                }
                dx=x2-x1;


                st1=0;st2=0;
                total=total+dx;
                kind=0;
        }


        if((total>=0)&&(total<14000))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        kind=1;
                        _outp(0x275,0x01);
                        Sleep(70);
                        _outp(0x275,0x00);


                }
                if((total>14000)&&(total<18000))
                {
                        kind=2;
                        _outp(0x275,0x02);
                        Sleep(70);
                        _outp(0x275,0x00);

                }
                if((total>18000)&&(total<50000))
                {
                        kind=3;
                        _outp(0x275,0x04);
                        Sleep(70);
                        _outp(0x275,0x00);
                }

                Preview(TRUE);



                Des2=SrcOld2;


//              SetTimer(2,1000,NULL);

                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
            ((CS_ccd3View*)View)->InvalidateRect(NULL,FALSE);
  }
void CS_ccd3View::Onrun()
{
        // TODO: Add your command handler code here

        VidCap.SetCallbackOnFrame(TRUE);
        sw=1;
}
void CS_ccd3View::Onred()
{
        // TODO: Add your command handler code here
        sw=2;
        VidCap.SetCallbackOnFrame(TRUE);
}
void CS_ccd3View::Ontheshold()
{
        // TODO: Add your command handler code here
//      SetTimer(2,3000,NULL);
        sw=3;
        VidCap.SetCallbackOnFrame(TRUE);

}
void CS_ccd3View::Oncontour()
{
        // TODO: Add your command handler code here
        sw=4;
        VidCap.SetCallbackOnFrame(TRUE);
}
void CS_ccd3View::Onorg()
{
        // TODO: Add your command handler code here
        sw=5;
        VidCap.SetCallbackOnFrame(TRUE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void CS_ccd3View::Onflow()
{
	// TODO: Add your command handler code here
	VidCap.Preview(TRUE);
}

void CS_ccd3View::OnTimer(UINT nIDEvent)
{
	// TODO: Add your message handler code here and/or call default

	if(tim_st==0)
	{
		KillTimer(2);
		sw=6;
		VidCap.SetCallbackOnFrame(TRUE);
//		goto exit_ontim;
	}
	if(tim_st==1)
	{
		KillTimer(1);
		_outp(0x275,0x00);
		tim_st=0;
//		goto exit_ontim;
	}
//	if(tim_st==2)
//	{
//		KillTimer(3);
//		_outp(0x275,0x00);
//		tim_st=0;
//		goto exit_ontim;
//	}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//          if(stopbit==1){goto exit_ontim;}

//          SetTimer(2,5000,NULL);
//exit_ontim:
//          stopbit=0;
            CView::OnTimer(nIDEvent);
}

void CS_ccd3View::Onautomate()
{
            // TODO: Add your command handler code here
            _outp(0x275,0x08);
            SetTimer(2,3000,NULL);

}

void CS_ccd3View::Onstop()
{
            // TODO: Add your command handler code here
            stopbit=1;


}

void CS_ccd3View::Oninitial()
{
            // TODO: Add your command handler code here
   _outp(0x275,0x10);
            Sleep(80);
            _outp(0x275,0x00);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void CS_ccd3View::OnA()
{
	// TODO: Add your command handler code here
	_outp(0x275,0x01);
	Sleep(80);
	_outp(0x275,0x00);
}

void CS_ccd3View::OnB()
{
	// TODO: Add your command handler code here
	_outp(0x275,0x02);
	Sleep(80);
	_outp(0x275,0x00);
}

void CS_ccd3View::OnC()
{
	// TODO: Add your command handler code here
	_outp(0x275,0x04);
	Sleep(80);
	_outp(0x275,0x00);
}

void CS_ccd3View::Onup180()
{
	// TODO: Add your command handler code here
	_outp(0x275,0x08);
//	Sleep(70);
//	_outp(0x275,0x00);

}

void CS_ccd3View::Onclear()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        // TODO: Add your command handler code here
        _outp(0x275,0x00);
}

void CS_ccd3View::Oninitial1()
{
        // TODO: Add your command handler code here
   _outp(0x275,0x10);
        Sleep(70);
        _outp(0x275,0x00);

}

void CS_ccd3View::Onstart1()
{
        // TODO: Add your command handler code here
//      _outp(0x275,0x08);
//      SetTimer(2,100,NULL);
        tim_st=0;
        _outp(0x275,0x10);
        SetTimer(2,8000,NULL);

//      tim_st=1;
//      SetTimer(1,80,NULL);
        Sleep(80);
        _outp(0x275,0x00);

}

void CS_ccd3View::Onclear1()
{
        // TODO: Add your command handler code here
        _outp(0x275,0x00);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
////////////////          CdibPsuedo          /////////////////

#include "stdafx.h"
//#include "BinaryTree.h"
//#include "cdib.h"
#include "CdibPsuedo.h"
#include "math.h"
CDibPsuedo::CDibPsuedo(long x,long y)
{
        MakeDib(x,y,8);
}

CDibPsuedo::~CDibPsuedo()
{
}
/*
CDibPsuedo::CDibPsuedo(long x,long y,MAKETYPE MT)
{
        MakeDib(x,y,8);
        switch(MT)
        {
        case CIRCLECOLOR:
                MakeCircleColorLevel();
                break;
        case STRAIGHTCOLOR:
                MakeStraightColorLevel();
                break;
        case GRAY:
                MakeGrayLevel();
                break;
        }
}
*/
void CDibPsuedo::MakeStraightColorLevel(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        m_lpBMI->bmiColors[0].rgbBlue=0;
        m_lpBMI->bmiColors[0].rgbGreen=0;
        m_lpBMI->bmiColors[0].rgbRed=0;
        m_lpBMI->bmiColors[255].rgbBlue=255;
        m_lpBMI->bmiColors[255].rgbGreen=255;
        m_lpBMI->bmiColors[255].rgbRed=255;
        int i;
        double fR,fG,fB;
        double fStep;
        fR=250.0f;
        fG=0.0f;
        fB=0.0f;
        fStep=fR/127.0f;
        for (i=1;i<128;i++)
        {
                fR -=fStep;
                fG +=fStep;
                m_lpBMI->bmiColors[i].rgbBlue=(unsigned char)fB;
                m_lpBMI->bmiColors[i].rgbGreen=(unsigned char)fG;
                m_lpBMI->bmiColors[i].rgbRed=(unsigned char)fR;
        }
        for (i=128;i<255;i++)
        {
                fG -=fStep;
                fB +=fStep;
                m_lpBMI->bmiColors[i].rgbBlue=(unsigned char)fB;
                m_lpBMI->bmiColors[i].rgbGreen=(unsigned char)fG;
                m_lpBMI->bmiColors[i].rgbRed=(unsigned char)fR;
        }
}
//void CDibPsuedo::MakeCircleColorLevel(void)
//{
//      m_lpBMI->bmiColors[0].rgbBlue=0;
//      m_lpBMI->bmiColors[0].rgbGreen=0;
//      m_lpBMI->bmiColors[0].rgbRed=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//
//        m_lpBMI->bmiColors[255].rgbBlue=255;
//        m_lpBMI->bmiColors[255].rgbGreen=255;
//        m_lpBMI->bmiColors[255].rgbRed=255;
//
//        int i;
//        double fAngle;
//        double pi=3.1414926535;
//        double fAngleStep=pi/254.0f;
//        double fRadian=104.0f;
//        double fX;
//        double fY;
//        double fxR=0.8164965809f;
void CDibPsuedo::MakeGrayLevel(void)
{
        int i;
        for (i=0;i<256;i++)
        {
                m_lpBMI->bmiColors[i].rgbBlue=i;
                m_lpBMI->bmiColors[i].rgbGreen=i;
                m_lpBMI->bmiColors[i].rgbRed=i;
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//////////////          Cdib.cpp          ////////////////

#include "stdafx.h"
#include "cdib.h"
#include <windowsx.h>
//#include "CDibTrue.h"
IMPLEMENT_SERIAL(CDib,CObject,0)
////////////////////////////////
CDib::CDib()
{
        m_dwLength = 0L;
        m_nBits = 0;
        m_lpBuf = NULL;
}
////////////////////////////////
BOOL CDib::MakeDib(long x,long y,int Bits)
{
        DWORD Length;
        long LUTsize;
        long Width;
        switch(Bits)
        {
        case 24:
                LUTsize=0;
                Width=x*3L;
                break;
        case 8:
                LUTsize=sizeof(RGBQUAD)*256L;
                Width=x;
                break;
        default:return FALSE;
        }
        if(Width&3)Width=(Width&0xfffffc)+4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Length=sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER)+LUTsize+
(Width*y);

        m_nBits=Bits;
        m_dwLength=Length;
        if(!AllocateMemory())return FALSE;

        m_lpBMFH->bfType=0x4d42;
        m_lpBMFH->bfSize=Length;
        m_lpBMFH->bfReserved1=0;
        m_lpBMFH->bfReserved2=0;
        m_lpBMFH->bfOffBits=sizeof(BITMAPFILEHEADER)+LUTsize+sizeof
(BITMAPINFOHEADER);

        m_lpBMIH->biSize=sizeof(BITMAPINFOHEADER);
        m_lpBMIH->biWidth=x;
        m_lpBMIH->biHeight=y;
        m_lpBMIH->biPlanes=1;
        m_lpBMIH->biBitCount=Bits;
        m_lpBMIH->biCompression=BI_RGB;
        if(Bits==24)m_lpBMIH->biSizeImage=Width*y;
        else m_lpBMIH->biSizeImage=0;
        m_lpBMIH->biXPelsPerMeter=0;
        m_lpBMIH->biYPelsPerMeter=0;
        m_lpBMIH->biClrUsed=0;
        m_lpBMIH->biClrImportant=0;

        m_lpData=m_lpBuf+m_lpBMFH->bfOffBits;
        m_lpPic=(unsigned char*)m_lpData;
        return TRUE;
}

////////////////////////////////////
CDib::~CDib()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (m_lpBuf)
        {
                GlobalFreePtr(m_lpBuf);
        }
}
////////////////////////////////

void CDib::Serialize(CArchive&ar)
{
        WORD wBM;
        CObject::Serialize(ar);
        if(ar.IsStoring())
        {
                ar.GetFile()->WriteHuge(m_lpBuf,m_dwLength);
        }
        else
        {
                if(m_dwLength!=0)
                        DeleteData();
                ASSERT(m_dwLength==0L);
                ar>>wBM;
                if(wBM==(WORD)0x4d42)
                {
                        ar.Flush();
                        ar.GetFile()->SeekToBegin();
                        m_dwLength=ar.GetFile()->GetLength();
                        if(m_dwLength>0L)
                        {
                                if(!AllocateMemory())
                                {
                                        return;
                                }
                                ar.Flush();
                                ar.GetFile()->ReadHuge(m_lpBuf,m_dwLength);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m_lpData=(unsigned char*)m_lpBMFH+m_lpBMFH->
bfOffBits;
                    m_lpPic=(unsigned char*)m_lpBMFH+m_lpBMFH->
bfOffBits;
                    m_nBits=m_lpBMIH->biBitCount;
                }
            }
            else
            {
                AfxMessageBox("invalid DIB archive");
                m_dwLength=0L;
                m_nBits=0;
            }
        }
}

////////////////////////////////
BOOL CDib::DisPlay(CDC* pDC,CPoint origin)
{
        if(!m_lpBuf)
        {
            return FALSE;
        }

        if(!::SetDIBitsToDevice(pDC->GetSafeHdc(),origin.x,origin.y,
            (WORD)m_lpBMIH->biWidth,(WORD)m_lpBMIH->biHeight,0,0,0,
            (WORD)m_lpBMIH->biHeight,m_lpData,m_lpBMI,DIB_RGB_COLORS))
        {
            return FALSE;
        }
            return TRUE;
}
////////////////////////////////
BOOL CDib::AllocateMemory(BOOL bRealloc)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(bRealloc)
        {
                m_lpBuf=(unsigned char*)GlobalReAllocPtr(m_lpBuf,m_dwLength,
                                GHND);
        }
        else
        {
                m_lpBuf=(unsigned char*)GlobalAllocPtr(GHND,m_dwLength);
        }
        if(!m_lpBuf)
        {
                AfxMessageBox("Unable to allocate DIB memory");
                m_dwLength=0L;
                m_nBits=0;
                return FALSE;
        }
        m_lpBMFH=(LPBITMAPFILEHEADER)m_lpBuf;
        m_lpBMIH=(LPBITMAPINFOHEADER)(m_lpBuf+sizeof(BITMAPFILEHEADER));
        m_lpBMI=(LPBITMAPINFO)m_lpBMIH;
        return TRUE;
}

void CDib::DeleteData(void)
{
        GlobalFreePtr(m_lpBuf);
        m_lpBuf=NULL;
        m_dwLength=0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;********** Program Microcontroller 8051 **************************
;************  Control   2 Stepping Motors &  Input from PC ************

                ORG     0000H
                LJMP    MAIN
                ORG     0040H

        STL             EQU     01H
        STU             EQU     01H
        FIRSTL          EQU     00H
        FIRSTU          EQU     00H
        COUNTL          EQU     00H
        COUNTU          EQU     00H
        COUNT           EQU     00H
        SIZE            EQU     00H
        DCOUNT          EQU     100
        D5COU           EQU     00H
        PA              EQU     8000H
        PB              EQU     8001H
        PCC             EQU     8002H
        PCONT           EQU     8003H

MAIN:   MOV             TMOD,#011H
        MOV             DPTR,#PCONT
        MOV             A,#080H
        MOVX            @DPTR,A
MLOOP:  JNB             P1.6,M1             ;INITIAL MOTOR LOWER
        MOV             C,P1.6              ;START DEBOUCE
        LCALL           D50M
        ANL             C,P1.6
        JNC             M1                  ;END DEBOUCE
        JB              P1.1,MINU           ;CHECK LIMIT  LOWER
        JB              P1.1,M1             ;CHECK LIMIT  LOWER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,FIRSTL
        CJNE    A,#00H,MFL1             ;FIRST INITIAL LOWER
        MOV     DPTR,#PA
        MOV     A,#01H
        MOVX    @DPTR,A
        MOV     STL,#01H
        MOV     FIRSTL,#01H
MFL1:   LCALL   INTL                    ;SECOND & OTHERS  INITIAL LOWER
        MOV     SIZE,#02H


MINU:   JB      P1.0,MFU2
        MOV     A,FIRSTU
        CJNE    A,#00H,MFU1             ;FIRST INITIAL UPPER
        MOV     DPTR,#PB
        MOV     A,#01H
        MOVX    @DPTR,A
        MOV     FIRSTU,#01H
        MOV     STU,#01H
MFU1:
        LCALL   INTU                    ;SECOND & OTHERS  INITIAL UPPER
MFU2:   LCALL   MU9
        LCALL   MU9
        MOV     SIZE,#02H
        LJMP    M1
M1:     JNB     P1.2,M2                 ;SIZE 1
        MOV     C,P1.2                  ;START DEBOUCE
        LCALL   D50M
        ANL     C,P1.2
        JNC     M2                      ;END DEBOUCE
        MOV     A,SIZE
        CJNE    A,#01H,M11              ;FROM SIZE 1
        LCALL   MU9
        LCALL   MU9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LJMP    M1E

M11:    CJNE    A,#02H,M12          ;FROM SIZE 2
        LCALL   ML9
        LCALL   ML9
        LCALL   ML9
        LCALL   MU9
        LCALL   MU9
        LJMP    M1E
M12:    CJNE    A,#03H,M2           ;FROM SIZE 3
        LCALL   ML9
        LCALL   ML9
        LCALL   DELAY
        LCALL   MU9
        LCALL   MU9
        LJMP    M1E
M1E:    MOV     SIZE,#01H
M2:     JNB     P1.3,M3             ;SIZE 2
        MOV     C,P1.3              ;START DEBOUCE
        LCALL   D50M
        ANL     C,P1.3
        JNC     M3                  ;END DEBOUCE
        MOV     A,SIZE
        CJNE    A,#01H,M21          ;FROM SIZE 1
        LCALL   ML9
        LCALL   MU9
        LCALL   MU9
        LJMP    M2E
M21:    CJNE    A,#02H,M22          ;FROM SIZE 2
        LCALL   MU9
        LCALL   MU9
        LJMP    M2E
M22:    CJNE    A,#03H,M3           ;FROM SIZE 3
        LCALL   ML9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   ML9
        LCALL   ML9
        LCALL   MU9
        LCALL   MU9
        LJMP    M2E
M2E:    MOV     SIZE,#02H
M3:     JNB     P1.4,MRET           ;SIZE 3
        MOV     C,P1.4              ;START DEBOUCE
        LCALL   D50M
        ANL     C,P1.4
        JNC     MRET                ;END DEBOUCE
        MOV     A,SIZE
        CJNE    A,#01H,M31          ;FROM SIZE 1
        LCALL   ML9
        LCALL   ML9
        LCALL   MU9
        LCALL   MU9
        LJMP    M3E
M31:    CJNE    A,#02H,M32          ;FROM SIZE 2
        LCALL   ML9
LCALL   MU9
        LCALL   MU9
        LJMP    M3E
M32:    CJNE    A,#03H,M3E          ;FROM SIZE 3
LCALL   MU9
        LCALL   MU9
        LJMP    M3E
M3E:    MOV     SIZE,#03H
MRET:   LJMP    MLOOP


INTL:   MOV     COUNT,#00H          ;Initial  motor  lower  routine
INTLH:  MOV     A,STL
INTL1:  CJNE    A,#01,INTL2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     DPTR,#PA
        MOV     A,#01H
        MOVX    @DPTR,A
        LJMP    INTLE
INTL2:  CJNE    A,#02,INTL3
        MOV     DPTR,#PA
        MOV     A,#03H
        MOVX    @DPTR,A
        LJMP    INTLE
INTL3:  CJNE    A,#03,INTL4
        MOV     DPTR,#PA
        MOV     A,#02H
        MOVX    @DPTR,A
        LJMP    INTLE
INTL4:  CJNE    A,#04,INTL5
        MOV     DPTR,#PA
        MOV     A,#06H
        MOVX    @DPTR,A
        LJMP    INTLE
INTL5:  CJNE    A,#05,INTL6
        MOV     DPTR,#PA
        MOV     A,#04H
        MOVX    @DPTR,A
        LJMP    INTLE
INTL6:  CJNE    A,#06,INTL7
        MOV     DPTR,#PA
        MOV     A,#0CH
        MOVX    @DPTR,A
        LJMP    INTLE
INTL7:  CJNE    A,#07,INTL8
        MOV     DPTR,#PA
        MOV     A,#08H
        MOVX    @DPTR,A
        LJMP    INTLE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INTL8:  CJNE    A,#08,INTL9
        MOV     DPTR,#PA
        MOV     A,#09H
        MOVX    @DPTR,A
        LJMP    INTLE
INTL9:  CJNE    A,#09,INTLE
        MOV     DPTR,#PA
        MOV     A,#01H
        MOVX    @DPTR,A
        MOV     STL,#01H
        LJMP    INTLE

INTLE:  MOV     TH0,#HIGH(-8500)
        MOV     TL0,#LOW(-8500)
        SETB    TR0
INTLW:  JNB     TF0,INTLW
        CLR     TR0
        CLR     TF0
        INC     STL
        JNB     P1.1,INTLED
RET
INTLED: LJMP    INTLH

INTU:                                   ;Initial motor upper routine
INTUH:  MOV     A,STU
INTU1:  CJNE    A,#01,INTU2
        MOV     DPTR,#PB
        MOV     A,#01H
        MOVX    @DPTR,A
        LJMP    INTUE
INTU2:  CJNE    A,#02,INTU3
        MOV     DPTR,#PB
        MOV     A,#03H
        MOVX    @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LJMP    INTUE
INTU3:  CJNE    A,#03,INTU4
        MOV     DPTR,#PB
        MOV     A,#02H
        MOVX    @DPTR,A
        LJMP    INTUE
INTU4:  CJNE    A,#04,INTU5
        MOV     DPTR,#PB
        MOV     A,#06H
        MOVX    @DPTR,A
        LJMP    INTUE
INTU5:  CJNE    A,#05,INTU6
        MOV     DPTR,#PB
        MOV     A,#04H
        MOVX    @DPTR,A
        LJMP    INTUE
INTU6:  CJNE    A,#06,INTU7
        MOV     DPTR,#PB
        MOV     A,#0CH
        MOVX    @DPTR,A
        LJMP    INTUE
INTU7:  CJNE    A,#07,INTU8
        MOV     DPTR,#PB
        MOV     A,#08H
        MOVX    @DPTR,A
        LJMP    INTUE
INTU8:  CJNE    A,#08,INTU9
        MOV     DPTR,#PB
        MOV     A,#09H
        MOVX    @DPTR,A
        LJMP    INTUE
INTU9:  CJNE    A,#09,INTUE
        MOV     DPTR,#PB
        MOV     A,#01H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOVX    @DPTR,A
        MOV     STU,#01H
        LJMP    INTUE

INTUE:  MOV     TH1,#HIGH(-8500)
        MOV     TL1,#LOW(-8500)
        SETB    TR1
INTUW:  JNB     TF1,INTUW
        CLR     TR1
        CLR     TF1
        INC     STU
        JNB     P1.0,INTUED
        RET
INTUED: LJMP    INTUH

ML9:    MOV     COUNTL,#00H             ;Motor lower turn 90 degree
ML9H:   MOV     A,STL
ML91:   CJNE    A,#01,ML92
        MOV     DPTR,#PA
        MOV     A,#01H
        MOVX    @DPTR,A
        LJMP    ML9E
ML92:   CJNE    A,#02,ML93
        MOV     DPTR,#PA
        MOV     A,#03H
        MOVX    @DPTR,A
        LJMP    ML9E
ML93:   CJNE    A,#03,ML94
        MOV     DPTR,#PA
        MOV     A,#02H
        MOVX    @DPTR,A
        LJMP    ML9E
ML94:   CJNE    A,#04,ML95
        MOV     DPTR,#PA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,#06H
        MOVX    @DPTR,A
        LJMP    ML9E
ML95:   CJNE    A,#05,ML96
        MOV     DPTR,#PA
        MOV     A,#04H
        MOVX    @DPTR,A
        LJMP    ML9E
ML96:   CJNE    A,#06,ML97
        MOV     DPTR,#PA
        MOV     A,#0CH
        MOVX    @DPTR,A
        LJMP    ML9E
ML97:   CJNE    A,#07,ML98
        MOV     DPTR,#PA
        MOV     A,#08H
        MOVX    @DPTR,A
        LJMP    ML9E
ML98:   CJNE    A,#08,ML99
        MOV     DPTR,#PA
        MOV     A,#09H
        MOVX    @DPTR,A
        LJMP    ML9E
ML99:   CJNE    A,#09,ML9E
        MOV     DPTR,#PA
        MOV     A,#01H
        MOVX    @DPTR,A
        MOV     STL,#01H
        LJMP    ML9E

ML9E:   MOV     TH0,#HIGH(-8500)
        MOV     TL0,#LOW(-8500)
        SETB    TR0
ML9W:   JNB     TF0,ML9W
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR     TR0
        CLR     TF0
        INC     STL
        INC     COUNTL
        MOV     A,COUNTL
        CJNE    A,#064H,ML9ED
        RET
ML9ED:  LJMP    ML9H


MU9:    MOV     COUNTU,#00H             ;Motor upper turn 90 degree
MU9H:   MOV     A,STU
MU91U:  CJNE    A,#01,MU92U
        MOV     DPTR,#PB
        MOV     A,#01H
        MOVX    @DPTR,A
        LJMP    MU9EU
MU92U:  CJNE    A,#02,MU93U
        MOV     DPTR,#PB
        MOV     A,#03H
        MOVX    @DPTR,A
        LJMP    MU9EU
MU93U:  CJNE    A,#03,MU94U
        MOV     DPTR,#PB
        MOV     A,#02H
        MOVX    @DPTR,A
        LJMP    MU9EU
MU94U:  CJNE    A,#04,MU95U
        MOV     DPTR,#PB
        MOV     A,#06H
        MOVX    @DPTR,A
        LJMP    MU9EU
MU95U:  CJNE    A,#05,MU96U
        MOV     DPTR,#PB
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,#04H
        MOVX    @DPTR,A
        LJMP    MU9EU
MU96U:  CJNE    A,#06,MU97U
        MOV     DPTR,#PB
        MOV     A,#0CH
        MOVX    @DPTR,A
        LJMP    MU9EU
MU97U:  CJNE    A,#07,MU98U
        MOV     DPTR,#PB
        MOV     A,#08H
        MOVX    @DPTR,A
        LJMP    MU9EU
MU98U:  CJNE    A,#08,MU99U
        MOV     DPTR,#PB
        MOV     A,#09H
        MOVX    @DPTR,A
        LJMP    MU9EU
MU99U:  CJNE    A,#09,MU9EU
        MOV     DPTR,#PB
        MOV     A,#01H
        MOVX    @DPTR,A
        MOV     STU,#01H
        LJMP    MU9EU
MU9EU:  MOV     TH1,#HIGH(-8500)
        MOV     TL1,#LOW(-8500)
        SETB    TR1
MU9W:   JNB     TF1,MU9W
        CLR     TR1
        CLR     TF1
        INC     STU
        INC     COUNTU
        MOV     A,COUNTU
        CJNE    A,#065H,MU9EUD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 RET
MU9EUD: LJMP    MU9H

D50M:   MOV     TH0,#HIGH(-50000)          ;Delay  50 mSec  routine
        MOV     TL0,#LOW(-50000)
        SETB    TR0
D50LP:  JNB     TF0,D50LP
        CLR     TF0
        CLR     TR0
        RET

DELAY:  MOV     DCOUNT,#0100               ;Delay  8  Sec  routine
AGAIN:  MOV     TH0,#HIGH(-8000)
        MOV     TL0,#LOW(-8000)
        SETB    TR0
DLOOP:  JNB     TF0,DLOOP
        CLR     TF0
        CLR     TR0
        DJNZ    DCOUNT,AGAIN
        RET

D5:     MOV     D5COU,#0100
D5AG:   MOV     TH0,#HIGH(-20000)
        MOV     TL0,#LOW(-20000)
        SETB    TR0
D5LP:   JNB     TF0,D5LP
        CLR     TF0
        CLR     TR0
        DJNZ    D5COU,D5AG
        RET
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานและรายงานฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความรู้และคำแนะนำจาก อาจารย์ เทอดศักดิ์ ลิ่วหาทอง สถานที่ทำงานที่ shop ภาควิชาวิศวกรรมเกษตร อีกทั้งกำลังใจและความช่วยเหลือจากเพื่อน ๆ ที่ร่วมทำงานมาด้วยกัน ผู้จัดทำจึงขอขอบทุกท่านที่มีส่วนเกี่ยวข้องในความสำเร็จครั้งนี้ด้วย

.....

(นายมนิศร ศิริวงศ์วัฒนา)

.....

(นายยุทธศักดิ์ เภาสมบัติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Color Quick Cam User Guide , Connectix Corporation ,1996
2. Scott Stanfield with Ralph Arveson, "Visual C++4 How to" , waite Group Press A Division of Sams Publishing Corte Madera , CA ,1996
3. Steve Rimmerm "WINDOWS BITMAPPED GRAPHICS" , WINDCREST/McGraw – Hill , 1993
4. วรรณวิภา ติตถะสิริ , "การเขียนโปรแกรมภาษา C ด้วยตนเอง" , ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้