สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรักษาความปลอดภัยสำหรับที่จอดรถยนต์

SECURITY SYSTEM FOR PARKING PLACE

โดย

นายอภิชย์ เหมาคม 44010581

นายอรรถพร สุพลจิต 44010593

อาจารย์ที่ปรึกษา

ผศ.เกรียงไกร วงศ์โรจนภรณ์

รศ.ดร.สุวิพล สิทธิชีวภาค

เลขหมู่.....
เลขทะเบียน 61357
วัน,เดือน,ปี 17 ก.ค. 2549

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยสำหรับที่จอดรถยนต์

SECURITY SYSTEM FOR PARKING PLACE

ภาควิชา
วิศวกรรมโทรคมนาคม

โดย
นายอภิชย์ เหมาคม
นายอรรถพร สุพลจิต

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2547

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบรักษาความปลอดภัยสำหรับที่จอดรถยนต์**

SECURITY SYSTEM FOR PARKING PLACE

ผู้จัดทำ

1. นายอภิชย์ เหมาคม 44010581

2. นายอรรถพร สุพลจิต 44010593

..... อาจารย์ที่ปรึกษา

(ผศ. เกรียงไกร วงศ์โรจนภรณ์)

..... อาจารย์ที่ปรึกษา

(รศ.ดร. สุวิพล สิทธิชีวภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยสำหรับที่จอดรถยนต์

SECURITY SYSTEM FOR PARKING PLACE

โดย นายอภิชย์ เหมาคม 44010581
นายอรรถพร สุพลจิต 44010593

อาจารย์ที่ปรึกษา ผศ.เกรียงไกร วงศ์โรจนภรณ์
รศ.ดร.สุวิพล สิทธิชีวภาค

บทคัดย่อ

โครงงานนี้เป็นการสร้างระบบรักษาความปลอดภัยสำหรับที่จอดรถยนต์ โดยอาศัยบัตรแถบแม่เหล็กประจำตัวของผู้ใช้รถยนต์แต่ละคน ในการระบุตัวตนของผู้ใช้ และจะทำการเปรียบแผ่นป้ายทะเบียนรถยนต์กับข้อมูลของแผ่นป้ายที่เก็บไว้ในฐานข้อมูล เมื่อบัตรแถบแม่เหล็กและแผ่นป้ายทะเบียนถูกต้องผู้ใช้ก็จะสามารถนำรถเข้าจอดได้ โดยในการประมวลผลภาพจะใช้หลักการประมวลผลภาพดิจิตอล (Digital Image Processing)

ABSTRACT

This Project is made to create Security System for Parking Place by using a magnetic card to identify the user. And will compare a car license with the data of a car license stored in the database. When magnetic card and car license are correct the user can take the car into the parking place. In the processing will use Digital Image Processing .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงงาน	1
1.2 ขอบเขตของโครงงาน	1
1.3 การประยุกต์ใช้งาน	2
1.3.1 ระบบป้องกันการโจรกรรมรถยนต์	2
1.3.2 ระบบควบคุมการเข้าออก	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ทฤษฎีของบัตรแม่เหล็กและวงจรแม่เหล็ก	3
2.1.1 คุณสมบัติของบัตรแม่เหล็ก	4
2.1.1.1 แทร็คของบัตรแม่เหล็ก	4
2.1.1.2 รูปแบบของข้อมูลในแทร็คที่ 2	5
2.1.2 คุณลักษณะของวงจรแม่เหล็ก	6
2.1.3 การบันทึกข้อมูลลงบัตรแม่เหล็ก	9
2.1.4 รูปแบบการบันทึกข้อมูล	9
2.1.5 การอ่านข้อมูลจากบัตรแม่เหล็ก	10
2.1.6 การบันทึกข้อมูลบนบัตรแม่เหล็ก	11
2.2 การสื่อสารข้อมูลอนุกรม	13
2.2.1 ความเร็วของการสื่อสารข้อมูลอนุกรม	13
2.2.2 รูปแบบของการส่งข้อมูลอนุกรม	13
2.3 ไมโครคอนโทรลเลอร์ MCS-51	14
2.3.1 การใช้งานพอร์ทสื่อสารอนุกรม	14
2.3.2 กระบวนการรับและส่งข้อมูลอนุกรมของ 8051	15
2.3.2.1 พอร์ทอนุกรม (โหมด0)	16
2.3.2.2 พอร์ทอนุกรม (โหมด1)	16
2.3.2.3 พอร์ทอนุกรม (โหมด2)	16
2.3.2.4 พอร์ทอนุกรม (โหมด3)	16
2.3.3 รีจิสเตอร์ที่ใช้ควบคุมการรับส่งข้อมูลอนุกรม SCON	21
2.4 การประมวลผลภาพเชิงตัวเลข (Digital Image Processing)	21
2.4.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล	21
2.4.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล	22
2.4.3 ไฟล์ข้อมูลภาพชนิดบิตแมป	23
2.4.3.1 รูปแบบของไฟล์ข้อมูลชนิดบิตแมป	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

2.4.3.2 โครงสร้างของไฟล์ของข้อมูลภาพชนิดบิตแมป 23

2.4.3.3 การจัดเก็บไฟล์ข้อมูลชนิดบิตแมป 23

2.4.4 การสร้างภาพไบนารี 24

2.4.5 การแบ่งส่วนภาพ (Image Segmentation) 26

2.4.5.1 การตรวจรู้แนวเส้น (Line Detection) 26

2.4.5.2 การตรวจหาขอบ (Edge detection) 27

2.4.6 ทฤษฎีการแยกตัวอักขระออกจากภาพ 32

2.4.6.1 การหากรอบตัวอักขระโดยวิธี Line Crossing 32

2.4.6.2 เทคนิคการตามรอยขอบภาพ (Contour Following) 34

2.4.6.3 เทคนิคการแยกโดยใช้วิธีการ Contour with matrix 36

2.4.6.4 เทคนิคการแยกตัวอักขระที่ติดกันโดยใช้ฮิสโตแกรม (Histogram) 39

2.5 ฐานข้อมูล 42

2.6 โครงสร้าง แอลซีดี โมดูล (LCD module) 43

2.6.1 ส่วนประกอบหลักของ แอลซีดี โมดูล 43

2.6.2 การเชื่อมต่อ แอลซีดี โมดูล เข้ากับไมโครคอนโทรลเลอร์ 44

2.6.3 ชุดคำสั่งของ แอลซีดี โมดูล 45

2.7 การใช้งานคีย์แพด (Keypad) 50

บทที่ 3 การสร้างและการออกแบบ 52

3.1 ตัวเครื่องอ่านบัตรแม่เหล็ก 52

3.1.1 รูปแบบของการบันทึกข้อมูลบนบัตรแม่เหล็ก 54

3.1.2 วงจรพื้นฐานสำหรับการบันทึกข้อมูลบนแถบแม่เหล็ก 54

3.1.3 การอ่านข้อมูลจากบัตรแม่เหล็ก 56

3.1.3.1 วงจรพื้นฐานสำหรับการสร้างสัญญาณรูปแบบ ISO จากบัตรแม่เหล็ก 57

3.1.3.2 วงจรอ่านข้อมูลบัตรแม่เหล็ก 59

3.1.3.3 วงจรอ่านข้อมูลบัตรแม่เหล็กโดยใช้ MC54910P 61

3.2 วงจรการทำงานของไมโครคอนโทรลเลอร์ MCS-51 63

3.3 การออกแบบโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 64

3.4 การออกแบบโปรแกรมสำหรับการอ่านข้อมูลจากเครื่องอ่านบัตรแม่เหล็ก 67

3.5 การแปลงสัญญาณแรงดันเป็นสัญญาณ TTL 70

3.6 ฐานข้อมูล 71

3.7 การรับภาพจากล้องและการหาหมายเลขทะเบียนรถยนต์จากภาพ 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

3.7.1 การรับภาพจากกล้อง 72

3.7.2 การหาหมายเลขทะเบียนรถยนต์จากภาพ 72

3.7.2.1 การการทำไบนาไรเซชัน 74

3.7.2.2 การตรวจหาขอบภาพ 75

3.7.2.3 การหาตำแหน่งป้ายทะเบียน 76

3.7.2.4 การหาตำแหน่งหมายเลขทะเบียน 77

3.8 การรู้จำตัวอักขระ 79

3.8.1 การเทียบค่าจุดภาพ(Matching) 79

3.8.2 การตัดและจำแนก (Cut and Classification) 80

3.8.3 การหาตำแหน่งของเส้นสีขาว 81

3.8.4 ลักษณะของป้ายทะเบียนรถยนต์ 84

บทที่ 4 การทดลองและผลการทดลอง 85

4.1 เครื่องอ่านบัตรแม่เหล็ก 85

4.2 การส่งข้อมูลจากการอ่านบัตรแม่เหล็กเข้าคอมพิวเตอร์ 87

4.3 ฐานข้อมูล 89

4.4 การรับภาพจากกล้องและการหาหมายเลขทะเบียนรถยนต์จากภาพ 93

4.5 การแสดงสถานะการทำงานบนแอลซีดี โมดูล 105

บทที่ 5 บทวิจารณ์และสรุป 108

ภาคผนวก

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันรถยนต์มีจำนวนเพิ่มมากขึ้น ทำให้ไม่สะดวกและสิ้นเปลืองบุคลากรในการบันทึกข้อมูลการเข้าออก และการอนุญาตให้รถยนต์เข้าไปจอดในสถานที่จอดรถยนต์ส่วนบุคคลเช่น ที่จอดรถยนต์ในคอนโดมีเนียมต่างๆ หรือที่จอดรถของอาคารสำนักงานต่างๆ รวมทั้งในปัจจุบันการโจรกรรมรถยนต์ได้เพิ่มจำนวนมากขึ้นเรื่อยๆ จึงได้มีแนวความคิดที่จะพัฒนาระบบที่จอดรถส่วนบุคคลอัตโนมัติขึ้น โดยใช้หลักการของบัตรแถบแม่เหล็ก และการจัดเก็บข้อมูลของแผ่นป้ายทะเบียนรถยนต์ โดยอาศัยหลักการประมวลผลภาพเชิงตัวเลข(Digital Image Processing) เพื่อนำข้อมูลที่ได้ไปเปรียบเทียบกับระบบฐานข้อมูล(Database) หรือนำไปประยุกต์ใช้กับงานอื่นๆ

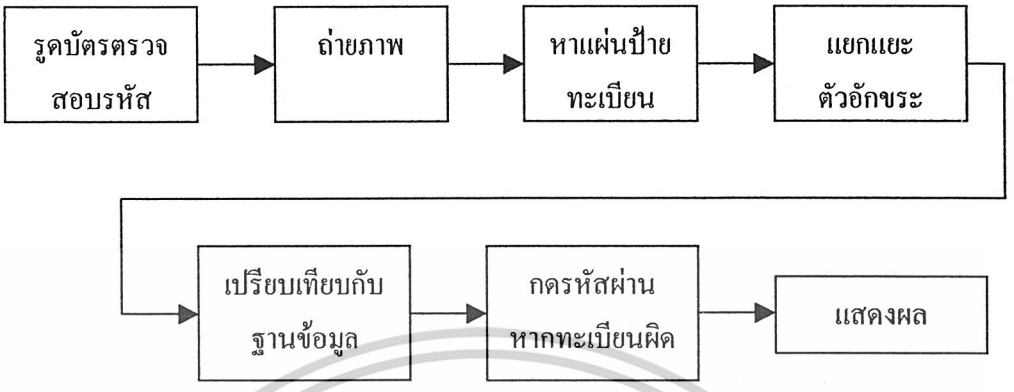
1.1 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาหลักการประมวลผลภาพเชิงตัวเลข
2) เพื่อศึกษาหลักการเขียนโปรแกรมด้วยภาษา Visual C++.NET
3) เพื่อศึกษาการใช้ประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS-51
4) เพื่อศึกษาการใช้งานบัตรแถบแม่เหล็ก
5) เพื่อศึกษาการเขียนโปรแกรมด้วยภาษาแอสเซมบลี(Assembly)
6) เพื่อศึกษาการใช้งานแอลซีดี โมดูล (LCD Module) และคีย์แพด (Keypad)

1.2 ขอบเขตของโครงงาน

ระบบที่จอดรถส่วนบุคคลอัตโนมัติที่จะนำเสนอนี้ ในส่วนของฐานข้อมูลจะทำการเก็บข้อมูลของเจ้าของรถยนต์โดยจะเก็บหมายเลขของแผ่นป้ายทะเบียนกับหมายเลขของบัตรแถบแม่เหล็กลงในฐานข้อมูล ในส่วนของการประมวลผลเมื่อผู้ใช้ต้องการจะนำรถเข้าหรือออกจากที่จอดรถจะต้องนำบัตรแถบแม่เหล็กมารูดหากเป็นบัตรที่ได้ทำการลงทะเบียนไว้แล้วระบบจะเริ่มทำการประมวลผลโดยการจับภาพและและทำการหาตำแหน่งของแผ่นป้ายทะเบียนรถยนต์โดยอาศัยขั้นตอนการประมวลผลภาพโดยมีขั้นตอนต่างๆ คือ การจับกรอบ(Edge Detection) การทำไบนาไรเซชัน(Binarization) การกำจัดสัญญาณรบกวน เมื่อหาแผ่นป้ายทะเบียนเจอแล้วจะนำมาแยกตัวอักษรและตัวเลขออกเป็นตัวๆ โดยอาศัยการหาฮิสโตแกรมและการตามรอยขอบภาพ หลังจากผ่านการแยกตัวอักษรและตัวเลขออกเป็นตัวๆแล้ว จะเข้าสู่กระบวนการเปรียบเทียบค่าโดยอาศัยการเปรียบเทียบ หากข้อมูลที่ได้ตรงกันก็จะแสดงผลว่าอนุญาตให้รถสามารถออกหรือเข้าที่จอดรถได้ หากการเปรียบเทียบไม่ถูกต้องระบบจะให้ผู้ใช้ทำการกดรหัสผ่านหากรหัสไม่ถูกต้องจะแสดงผลว่าไม่อนุญาตให้รถเข้าหรือออกจากที่จอดรถแต่ถ้าหากรหัสผ่านถูกต้องระบบก็จะแสดงผลว่าอนุญาต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 ขั้นตอนการทำงานของโปรแกรม

1.3 การประยุกต์ใช้งาน

1.3.1 ระบบป้องกันการโจรกรรมรถยนต์

ในสถานที่ที่มีความเสี่ยงต่อการโจรกรรมรถยนต์ ระบบนี้จะมีความปลอดภัยเพราะใช้การเปรียบเทียบหมายเลขบัตร, หมายเลขทะเบียน หรือรหัสผ่านเพื่อตรวจการเข้าออกของรถยนต์ โดยรถยนต์ที่จะสามารถเข้าออกได้จะต้องมีหมายเลขบัตรกับหมายเลขทะเบียนหรือรหัสผ่านตรงกับฐานข้อมูลที่เก็บไว้ในคอมพิวเตอร์ ดังนั้นระบบนี้จึงเป็นระบบที่มีความปลอดภัยสูง

1.3.2 ระบบควบคุมการเข้าออก

ในสถานที่สำคัญ หรือในสถานที่ที่มีที่จอดรถจำกัดคนที่ไม่ได้รับอนุญาตไม่สามารถเข้าไปในสถานที่นั้นได้ ดังนั้นระบบที่จอดรถส่วนบุคคลอัตโนมัติ นี้จะอนุญาตให้เฉพาะรถที่ลงทะเบียนแล้วเท่านั้นที่จะสามารถเข้าไปในสถานที่ต่างๆนั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ทฤษฎีของบัตรแม่เหล็กและวงจรแม่เหล็ก

บัตรแม่เหล็กที่ใช้กันอยู่ในปัจจุบัน มีอยู่ด้วยกันมากมายหลายชนิด ในยุคแรกๆ บัตรแม่เหล็กจะมีเนื้อวัสดุเป็นพลาสติกพีวีซีชนิดแข็งและแบบที่มีเนื้อวัสดุเป็นกระดาษ แต่ในปัจจุบันได้พัฒนาเนื้อวัสดุของบัตรแม่เหล็กมาเป็น Polyethylene Telephthalate หรือที่เรียกย่อๆ ว่า PET

ชนิด	เนื้อวัสดุ	ความหนาของบัตร	การใช้งาน
บัตรพลาสติก (Plastic Card)	- PVC ชนิดแข็ง - PVCA ชนิดแข็ง	0.76 มม.	- บัตร ATM - บัตรเครดิต - บัตรประจำตัว (ID)
PET Card	- Polyethylene Telephthalate (PET)	0.2 – 0.4 มม.	- บัตรโทรศัพท์
บัตรกระดาษ (Composite Paper Card)	- กระดาษอย่างดี - กระดาษเคลือบพลาสติก - กระดาษบันทึกข้อมูล	0.2 – 0.76 มม.	- ตั๋วรถไฟ - ตั๋วทางด่วน - ตั๋วเครื่องบิน - บัตรโรงแรม

ตารางที่ 2.1 ชนิดและการใช้งานบัตรแม่เหล็ก

การใช้งาน	ชนิดของบัตร
การเงิน การธนาคาร	- บัตร ATM - บัตรเครดิต - บัตรหลักทรัพย์
การคมนาคม	- ตั๋วรถไฟแม่เหล็ก - ตั๋วผ่านทางด่วน - ตั๋วเครื่องบิน - ตั๋วจอดรถ
การสื่อสาร	- บัตรโทรศัพท์
การสำนักงาน (OA,FA)	- บัตรประจำตัวพนักงาน - บัตรประจำตัวนักเรียน - บัตรตรวจโรค - บัตรโรงแรม
การรักษาความปลอดภัย	- บัตรปิดและเปิดล็อค

ตารางที่ 2.2 ตัวอย่างการใช้งานบัตรแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 คุณสมบัติของบัตรแม่เหล็ก

2.1.1.1 แทร็คของบัตรแม่เหล็ก

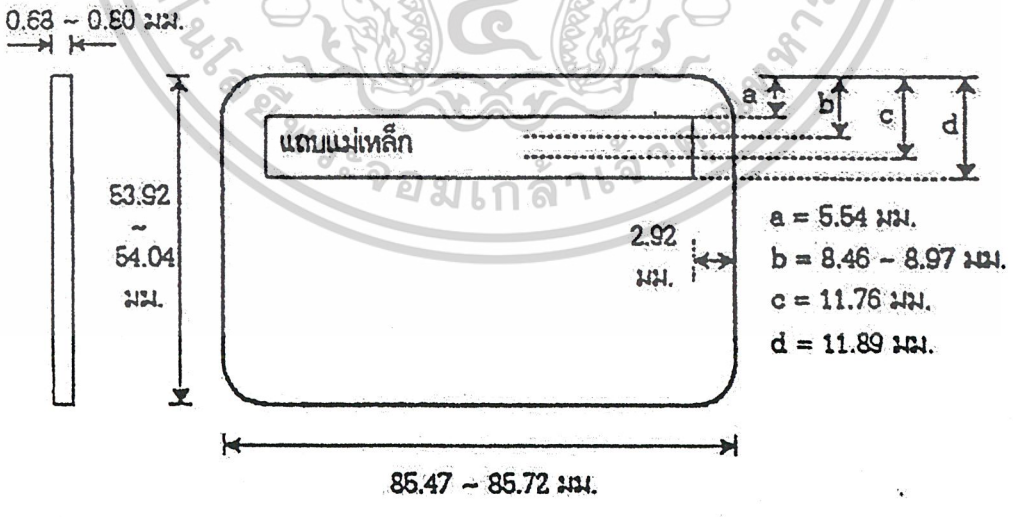
Magnetic Card หรือ บัตรแถบแม่เหล็ก ในรูปแบบของบัตรเงินสด บัตร ATM บัตรเครดิต บัตร VISA และบัตรอื่นๆอีกมากมาย ฯลฯ เป็นการ์ดแถบแม่เหล็ก ซึ่งบนตัวบัตรเองจะบันทึกข้อมูลและรายละเอียดต่างๆของบัตรไว้ในรูปของเส้นแรงแม่เหล็กภายในส่วนที่เป็นแถบแม่เหล็กในการ์ด โดยที่แถบแม่เหล็กที่อยู่บนการ์ด ซึ่งเรียกว่าแทร็ค (Track) นั้น ตามปกติแล้วจะถูกแบ่งออกเป็น 3 ส่วนด้วยกัน ซึ่งในแต่ละส่วนจะใช้เก็บข้อมูลซึ่งมีความหนาแน่นและลักษณะของข้อมูลที่มีความแตกต่างกัน ดังนี้คือ

แถบแม่เหล็กที่	ความหนาแน่นในการบันทึก	การเข้ารหัสข้อมูล	จำนวนตัวอักษรสูงสุด	ลักษณะของข้อมูลที่เก็บในบัตร
Track 1	210 BPI	ALPHA	79	ชื่อเจ้าของบัตรและหมายเลขบัตร
Track 2	75 BPI	BCD	40	หมายเลขบัตรและวันหมดอายุ
Track 3	210 BPI	BCD	107	หมายเลขบัตรและรหัสพิเศษ

* BPI = Byte Per Inch

ตารางที่ 2.3 ลักษณะการเก็บข้อมูลของแต่ละแทร็ค

ตำแหน่งแทร็คที่สองของบัตรแม่เหล็ก เป็นไปตามมาตรฐาน ISO 7811/5 ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 ตำแหน่งแทร็คของบัตรแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.1 บริเวณแทร็คที่สองจะอยู่ระหว่าง เส้นขนาน (b) ค่า 8.97 มิลลิเมตร (0.353 นิ้ว) (ค่าสูงสุด) หรือค่า 8.46 มิลลิเมตร (0.333 นิ้ว) (ค่าต่ำสุด) กับ เส้นขนาน (c) ค่า 11.76 มิลลิเมตร (0.463 นิ้ว)

2.1.1.2 รูปแบบของข้อมูลในแทร็คที่ 2

โดยรูปแบบของข้อมูลในแถบแม่เหล็กส่วนที่ 2 ซึ่งใช้บันทึกสัญญาณด้วยความหนาแน่นของสนามแม่เหล็ก 75 BPI ซึ่งจะบรรจุจำนวนรหัสของข้อมูลต่างๆในส่วนนี้ ได้สูงสุดไม่เกิน 40 ตัวอักษร โดยนับรวมรหัสควบคุมและรหัสตรวจสอบด้วย ซึ่งมีรายละเอียดดังนี้คือ

SS	PAN	FS	Additional Data	ES	LRC

SS = รหัสข้อมูลแสดงการเริ่มต้นของข้อมูลในบัตร (Start Sentinel) มีรหัสข้อมูลเป็น OBH (;)

PAN = ข้อมูลแสดงหมายเลขของบัตร (มีจำนวนข้อมูลในส่วนนี้สูงสุดไม่เกิน 19 หลัก)

FS = รหัสข้อมูลแสดงการแบ่งแยกข้อมูล (Field Seperator) มีรหัสข้อมูลเป็น (=)

Additional Data = เป็นข้อมูลเพิ่มเติมอื่นๆของบัตร เช่น เดือน/ปี ที่ออกบัตรและหมดอายุ

ES = รหัสข้อมูลแสดงการสิ้นสุดของข้อมูลในบัตร (End Sentinel) มีรหัสข้อมูลเป็น 0FH (?)

LRC = ข้อมูลตรวจสอบความผิดพลาด (Longitudinal Redundancy Check)

0000000000 SS Data,Data,Data,…,Data ES LRC 0000000000

จากภาพซึ่งใช้แสดง ลักษณะการจัดเรียงของข้อมูลที่เก็บไว้ในแถบแม่เหล็กในส่วนที่ 2 ซึ่งจัดเรียงลำดับความสำคัญจากซ้ายไปขวาอนุกรมทีละบิต โดยจะสังเกตเห็นว่ามีข้อมูลซึ่งเป็นลอจิกศูนย์หรือที่เรียกว่า “Clocking Bit” นำหน้าและปิดท้ายข้อมูลจริงอยู่ ซึ่งลักษณะการจัดเก็บข้อมูลของส่วนที่ 2 จะเก็บข้อมูลด้วยการเข้ารหัสแบบ “Modulo5” ซึ่งในแต่ละชุดข้อมูลจะประกอบด้วยข้อมูล 5 บิต โดย 4 บิตแรก (D0…D3) เป็นรหัสข้อมูลแบบ BCD ส่วนบิตที่ 5 (D4) เป็นพาริตี้แบบคี่ เพื่อใช้ตรวจสอบความถูกต้องของข้อมูลในแต่ละไบท์ (5 บิต) ที่อ่านได้ ซึ่งหากค่าของบิตพาริตี้ผิดพลาด แสดงว่า การอ่านข้อมูลนั้นล้มเหลว โดยเริ่มทำการอ่านข้อมูลเมื่อสัญญาณ Present เริ่มเป็นศูนย์ก่อน และทำการอ่านข้อมูลในทุกๆขอบขาลงของสัญญาณนาฬิกาเสมอ ซึ่งข้อมูลในส่วนก่อนเริ่มต้นและหลังจากสิ้นสุด ของการอ่านนี้จะมีค่าเป็นศูนย์ (สัญญาณข้อมูล= ลอจิก “1”) นำหน้าและปิดท้ายข้อมูลจริงอยู่ ซึ่งเรียกว่า “Clocking Bit” ซึ่งข้อมูลในส่วนนี้ไม่ต้องสนใจ แต่ให้ตรวจสอบและรอจนกว่าจะเริ่มเป็นข้อมูลสตาร์ทบิต (OBH หรือ 11010 ในที่นี้เรียงลำดับความสำคัญจากซ้ายไปขวา ซึ่งบิตเริ่มต้นของรหัส OBH ต้องเริ่มด้วย 1 เป็นบิตแรกเสมอ) ดังนั้นในการอ่านเราต้องรอจนกว่าจะพบสัญญาณข้อมูลมีค่าเป็น “1” (สัญญาณข้อมูล = ลอจิก “0” เพราะกลับสภาวะกันอยู่) จึงเริ่มเก็บข้อมูลชุดละ 5 บิต ไปเรื่อยๆจนถึงรหัสจบ (0FH) ซึ่งเมื่อพบรหัสจบและจะมีข้อมูลตามมาอีก 1 ไบท์ ซึ่งเป็นข้อมูลสำหรับตรวจสอบความผิดพลาด ของการอ่านข้อมูลทั้งหมดในส่วนที่ 2 เรียกว่า “LRC” ซึ่งค่าของ “LRC” สามารถหาได้จากการนำเอา ข้อมูลในแต่ละไบท์ (ไม่คิดพาริตี้บิต) ตั้งแต่เริ่มต้น จนสิ้นสุดมาทำการ XOR กัน โดยครั้งแรกให้นำไบท์เริ่มต้น (0BH) ทำการ XOR กับศูนย์แล้วนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ที่ได้ไป XOR กับไบท์ถัดไปเรื่อยๆ ตามลำดับ ซึ่งหากผลลัพธ์สุดท้ายที่ได้ไม่เท่ากับ ค่าของ "LRC" ที่อ่านมาได้แสดงว่าการอ่านข้อมูลทั้งหมดล้มเหลว

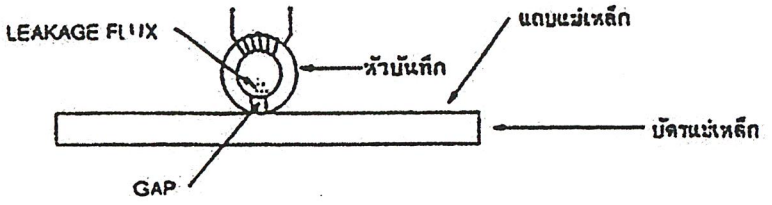
Parity	D3	D2	D1	D0	Character	Function
1	0	0	0	0	0 (0H)	Data
0	0	0	0	1	1 (1H)	Data
0	0	0	1	0	2 (2H)	Data
1	0	0	1	1	3 (3H)	Data
0	0	1	0	0	4 (4H)	Data
1	0	1	0	1	5 (5H)	Data
1	0	1	1	0	6 (6H)	Data
0	0	1	1	1	7 (7H)	Data
0	1	0	0	0	8 (8H)	Data
1	1	0	0	1	9 (9H)	Data
1	1	0	1	0	: (AH)	Control
0	1	0	1	1	; (BH)	Start Sentinel
1	1	1	0	0	< (CH)	Control
0	1	1	0	1	= (DH)	Field Seperator
0	1	1	1	0	> (EH)	Control
1	1	1	1	1	? (FH)	End Sentinel

ตารางที่ 2.4 ข้อมูลของรหัส BCD สำหรับเครื่องอ่านบัตรแถบแม่เหล็ก

2.1.2 คุณลักษณะของวงจรแม่เหล็ก

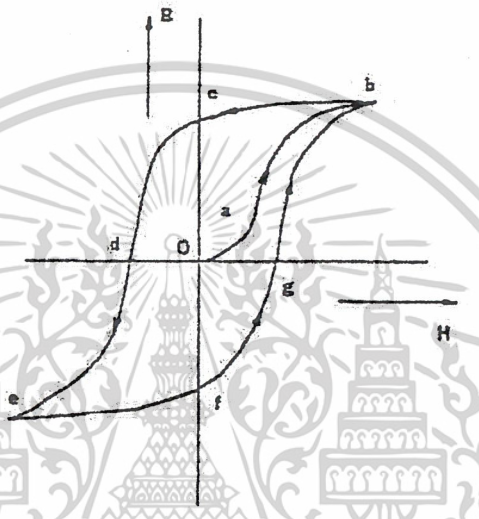
คุณลักษณะของวงจรแม่เหล็กสามารถแสดงได้ด้วยกราฟดังรูปที่ 2.2 โดยเป็นการพล็อทระหว่าง "ฟลักซ์แม่เหล็ก" (Flux Density) B และ "สนามแม่เหล็ก" (Magnetizing Force) H ทั้งนี้ความสัมพันธ์ของ B-H เป็นความสัมพันธ์ระหว่างแรงดันกับกระแสฟลักซ์แม่เหล็ก หมายถึงจำนวนเส้นแรงแม่เหล็กต่อตารางนิ้วที่ผ่านพื้นที่หน้าตัดของแกน ส่วน Magnetizing Force นั้นเท่ากับจำนวนแอมแปร์-รอบ (ampere turn) ต่อนิ้วของความยาว แกนในการทำงานหากสนามแม่เหล็ก (H) มีค่าเพิ่มขึ้นทีละน้อยจากศูนย์ ฟลักซ์แม่เหล็ก B ก็จะค่อยๆ เพิ่มขึ้นเป็นสัดส่วนด้วย (0-a-b) จนกระทั่งถึงจุดอิ่มตัวที่ b จากนั้นถ้าแรงดันคร่อมขดลวดตกลงเป็นศูนย์ฟลักซ์แม่เหล็ก B จะไม่กลับไปที่ศูนย์ แต่มันจะลดลง ตามเส้นโค้ง b-c เมื่อ H กลายเป็นศูนย์ โดยมีแมกนิจูดหรือขนาด 0-c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การบันทึกลงบนบัตรแม่เหล็ก

รูปที่ 2.2 วงจรของหัวบันทึก

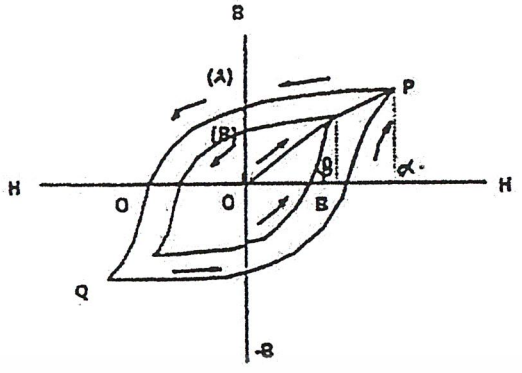


รูปที่ 2.3 B-H Curve

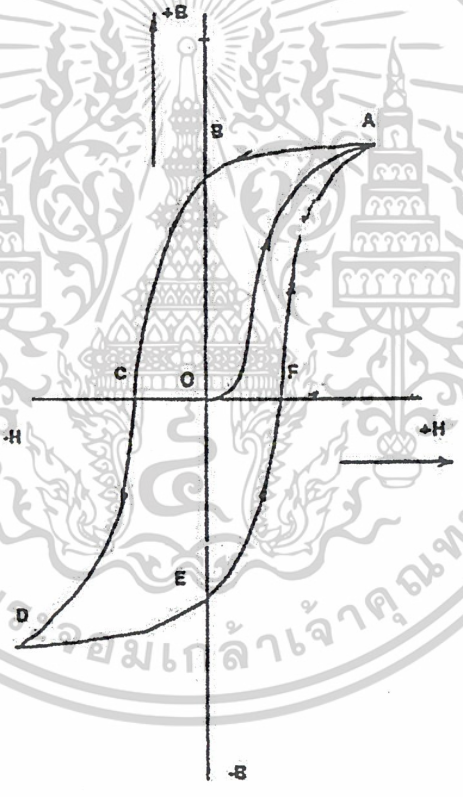
ถ้าหากลองป้อนกระแสหรือ H ในทิศทางตรงกันข้ามกับขดลวด ค่า B จะเป็นไปตามโค้ง จาก c-d-e ที่จุด e เป็นจุดที่มีการอิ่มตัวตรงข้ามกันของวัสดุแม่เหล็กจะเห็นได้ว่าหากเราป้อน แรงดันไฟฟ้ากระแสสลับเข้าที่ขดลวดตามรูปที่ 2.1 แล้วค่า B จะเดินทางเกิดเป็นวงรอบ หรือลูป (loop) ขึ้น คือจาก 0-a-b-c-d-e-f-g-b เราเรียกลูปดังกล่าวว่า “ฮีสเตอร์ริซิส”

- ระยะห่างระหว่างจุดตัดแกน (0) ถึง b เรียกว่า “Magnetization Curve ของวัสดุ”
- Closed loop เรียกว่า “Hysteresis Loop”
- ค่าของ flux density (B) จาก 0 ถึง c เรียกว่า “retentivity ของ magnetic substance”
- ค่าของ magnetizing curve (H) จาก 0 ถึง d เรียกว่า “coercive force”
- จากรูปที่ 2.2 นั้นหากปริมาณหรือขนาดของ H (ซึ่งป้อนเข้าไปครั้งแรก) หยุดที่จุดที่มีค่าต่ำๆ จุดหนึ่งแล้วกลับ จะมีผลทำให้เกิดเป็นลูปเล็ก (เล็กกว่าครั้งแรก) ขึ้นดังแสดงในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การเปลี่ยนแปลงเนื่องจากขนาดของ H



รูปที่ 2.5 ฮิสเทอริซิสลูปของเหล็ก

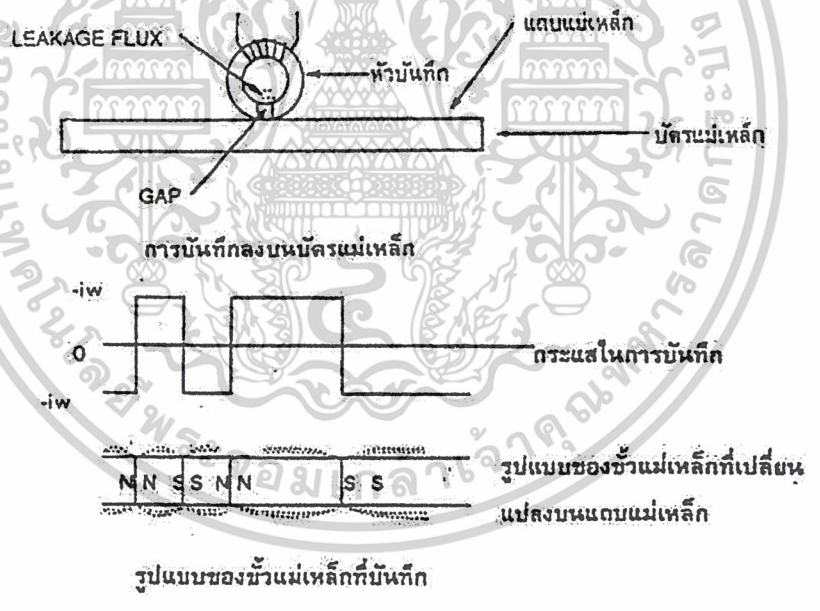
ในกรณีนี้หากป้อนแรงแม่เหล็กเข้าที่ตำแหน่ง $\propto$ แล้วแรงแม่เหล็ก (A) จะคงอยู่แม้เอาแรงแม่เหล็กออกแล้วก็ตาม และหากป้อนสนามแม่เหล็ก β เข้าไป (B) ก็จะคงอยู่ด้วยเช่นกัน อันนี้จึงเป็นการอธิบายถึงหลักการจำ (memory) ของการบันทึกข้อมูลด้วย hysteresis phenomenon "B-H characteristic"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกนัยหนึ่งก็คือ ฟังก์ชันความจำ (memory function) ของการบันทึกนั้นก็คือเปลี่ยนแปลงรูปคลื่นของข้อมูลที่จะบันทึกให้อยู่ในรูปของฟลักซ์แม่เหล็ก ในกรณีนี้จะกล่าวในเรื่องเครื่องบันทึกบัตรแม่เหล็กจากที่กล่าวมาข้างต้นเป็นทฤษฎีพื้นฐานซึ่งเป็นหลักการของเครื่องบันทึกเทปทั่วๆ ไป ซึ่งนำมาเป็นแนวทางสำหรับเครื่องอ่านและบันทึกแถบแม่เหล็กเท่านั้น โดยลำดับต่อไปจะกล่าวถึงการบันทึกข้อมูลลงบัตรแม่เหล็ก

2.1.3 การบันทึกข้อมูลลงบัตรแม่เหล็ก

การบันทึกข้อมูลลงบัตรแม่เหล็กจะใช้วิธีการบันทึกแบบดิจิตอล ในลักษณะเช่นเดียวกับที่ใช้ในแผ่นฟลอปปี้ดิสก์ หรือ เทปแม่เหล็กสำหรับเครื่องคอมพิวเตอร์ทั่วๆ ไป การบันทึกข้อมูลลงบนแถบแม่เหล็กนั้นจะต้องป้อนกระแสพัลส์ซึ่งมีทั้งด้านบวกและด้านลบ พร้อมทั้งมีขนาดเพียงพอเข้าที่ขดลวดของหัวบันทึกซึ่งกดอยู่บนแถบแม่เหล็กที่เคลื่อนที่ด้วยความเร็วคงที่ ดังในรูปที่ 2.6 แถบแม่เหล็กจะเกิดเป็นแม่เหล็กถาวรขนาดเล็กเรียงตัวกันตามขั้วบวกหรือลบของพัลส์และความกว้างของพัลส์สัญญาณที่บันทึก เนื่องจากกระแสพัลส์ที่ใช้ในการบันทึกมีขนาดเพียงพอที่หัวบันทึก จะทำให้แถบแม่เหล็กมีสนามแม่เหล็กอิ่มตัวได้ดังนั้นเมื่อทำการบันทึกข้อมูล ตัวข้อมูลที่เคยมีอยู่จะถูกเขียนทับและหายไป เหลือเพียงข้อมูลใหม่เท่านั้น



รูปที่ 2.6 การบันทึกข้อมูล

2.1.4 รูปแบบการบันทึกข้อมูล

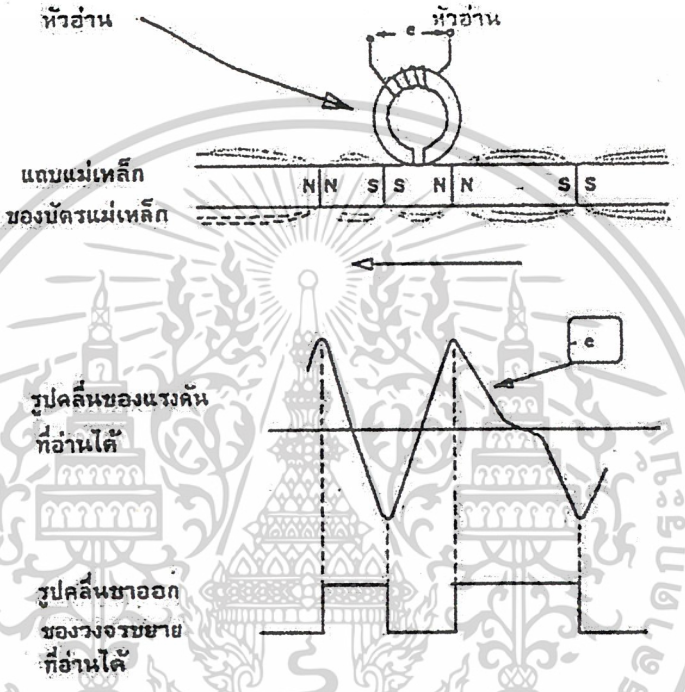
รูปแบบการบันทึกข้อมูลลงบนบัตรแม่เหล็กส่วนใหญ่ จะใช้รูปแบบ F2F หมายถึง Two Frequency Coherent Encoding และ FM หมายถึง Frequency Modulation การบันทึกข้อมูลในลักษณะเช่นนี้การบันทึกข้อมูล (Data) และคล็อก (Clock) เข้าไว้ในแทรคเดียวกัน นอกเหนือจากนี้ยังมีการบันทึกที่ข้อมูลและคล็อกแยกคนละแทรค เช่นแบบ NRZI (Non Zero Inverted Recording)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 การอ่านข้อมูลจากบัตรแม่เหล็ก

การอ่านข้อมูลจากบัตรแม่เหล็ก ทำได้โดยให้หัวอ่านสัมผัสกับแถบแม่เหล็กซึ่งเคลื่อนที่ด้วยความเร็วคงที่ในรูปที่ 2.7 ฟลักซ์ที่เกิดจากแม่เหล็กถาวรขนาดเล็กบนแถบแม่เหล็กจะผ่านจากแก็ปของหัวอ่านไปยังแกน (Core) การเปลี่ยนแปลงของฟลักซ์ ตามข้อมูลนั้น

จุดสูงสุด (Peak) ของแรงดันที่อ่านได้นั้นจะตรงกับจุดที่สนามแม่เหล็กบนแถบแม่เหล็กกลับทิศทางพอดี ดังนั้นถ้าขยายแรงดันนี้ขึ้นและตรวจหาจุดสูงสุด (Peak) ด้วยวิธีดิฟเฟอเรนเชียลแล้วเปลี่ยนเป็นสัญญาณพัลส์ ก็จะได้ข้อมูลที่บันทึกอยู่ในบัตรแม่เหล็ก (รูปที่ 2.7)



รูปที่ 2.7 การอ่านข้อมูล

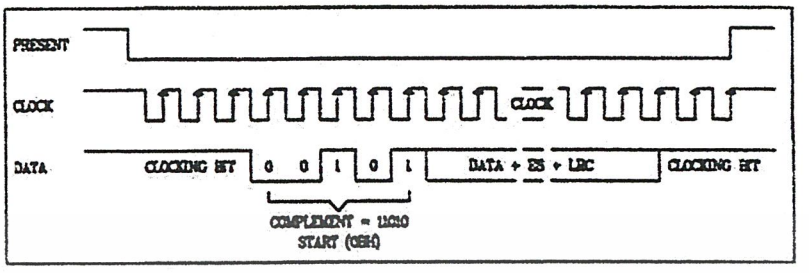
โดยสัญญาณที่จะออกมาจากเครื่องอ่านบัตรแม่เหล็กจะมีด้วยกัน 3 สัญญาณ คือ

1. สัญญาณข้อมูล โดยได้จากการอ่านข้อมูลแบบอนุกรมทีละบิตโดยต้องพิจารณาให้สัมพันธ์สอดคล้องกับสัญญาณนาฬิกา (Clock) ของบัตรด้วย ซึ่งสัญญาณข้อมูลที่อ่านได้นี้จะกลับสภาวะกับสัญญาณจริง คือมีสภาวะเป็นตรงข้ามกันอยู่ ดังนั้น เมื่ออ่านสัญญาณได้แล้วจะต้องกลับสภาวะของสัญญาณเสียก่อนจึงจะได้ข้อมูลที่ถูกต้อง

2. สัญญาณนาฬิกา (Clock) ใช้เป็นสัญญาณอ้างอิงในการอ่านสัญญาณข้อมูลจากบัตร โดยการอ่านสัญญาณข้อมูลแต่ละบิตนั้นต้องอ่านในขณะที่สัญญาณนาฬิกาเป็นขอบขาลงเสมอ ซึ่งสัญญาณข้อมูลที่อ่านได้นั้นจะเริ่มต้นจากบิตที่มีนัยสำคัญต่ำสุด(LSB) ก่อนเป็นอันดับแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

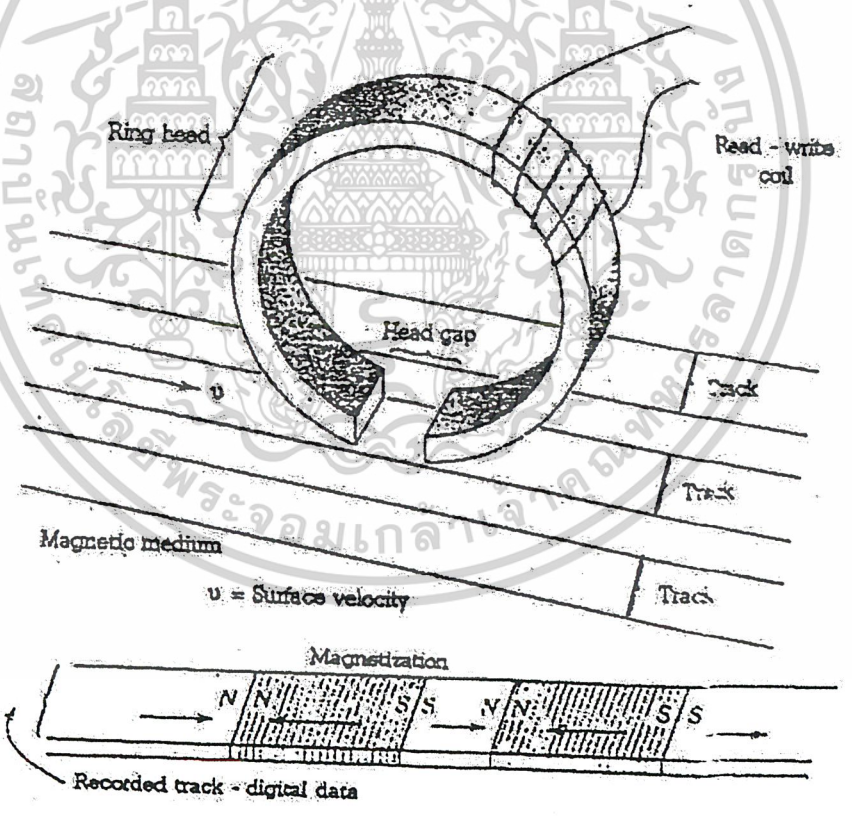
3. สัญญาณ Present ทำงานที่ลอจิกศูนย์ เมื่อสัญญาณนี้เกิด (“0”) จะบ่งบอกให้ทราบว่าเครื่องอ่านบัตรแม่เหล็กเริ่มต้นส่งข้อมูลออกมา หรือมีการนำบัตรแถบแม่เหล็กไปรูดผ่านหัวอ่านสัญญาณของเครื่อง เมื่อสัญญาณนี้หมดลง (กลับลอจิกเป็นหนึ่ง) อีกครั้งหนึ่งแสดงว่าข้อมูลจากการอ่านในครั้งนั้นถูกส่งออกไปหมดสิ้นเรียบร้อยแล้ว



รูปที่ 2.8 ความสัมพันธ์ของสัญญาณข้อมูล, สัญญาณนาฬิกาและสัญญาณ present

2.1.6 การบันทึกข้อมูลบนบัตรแม่เหล็ก

การบันทึกข้อมูลบนแถบแม่เหล็ก ประกอบด้วย แถบแม่เหล็กที่ใช้เป็นตัวกลางและหัวแม่เหล็กที่ใช้เป็นหัวบันทึก ซึ่งเคลื่อนที่ด้วยความเร็วคงที่ ดังแสดงในรูปที่ 2.9

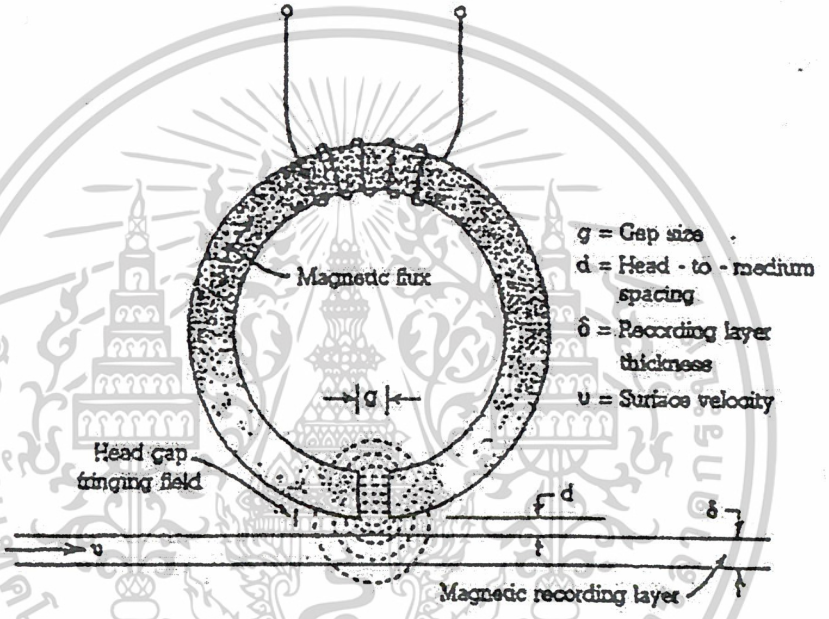


รูปที่ 2.9 หัวบันทึกและแถบแม่เหล็กในขบวนการบันทึกข้อมูลลงบนแถบแม่เหล็ก

จากรูปที่ 2.9 การบันทึกข้อมูลลงบนแถบแม่เหล็ก จะขึ้นอยู่กับคุณสมบัติของสารแม่เหล็กในแถบแม่เหล็กและหัวแม่เหล็กที่นำมาใช้เป็นหัวบันทึก โดยที่แกนของหัวบันทึกทั่วไป จะเป็นแกนที่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

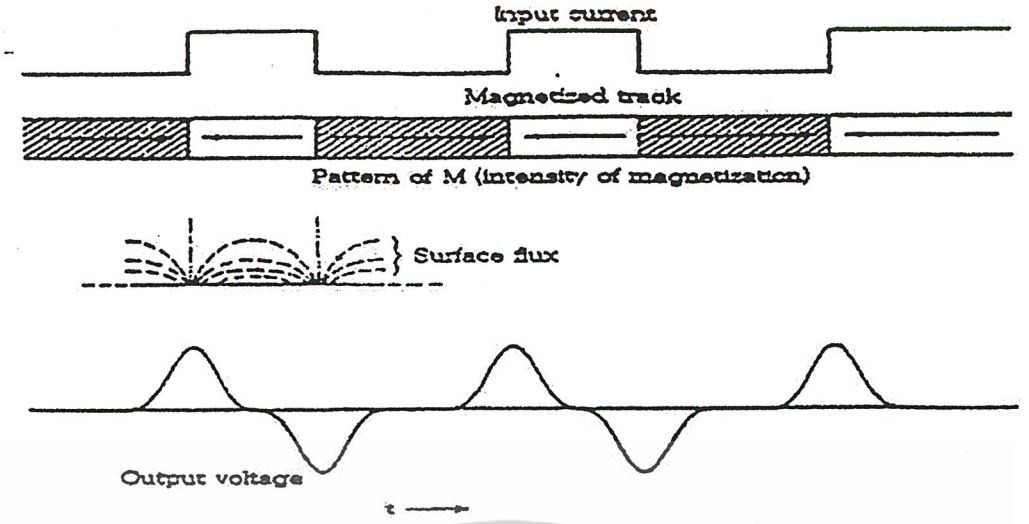
คุณสมบัติทางแม่เหล็กอย่างอ่อนๆ พันด้วยขดลวด และที่แกนมีช่องว่างอยู่ (gap) ส่วนบริเวณของสารแม่เหล็กจะถูกบันทึกข้อมูลเรียกว่าแทร็ค (track) โดยแต่ละแทร็คจะเรียงขนานกันบนแถบแม่เหล็ก สัญญาณเอาท์พุทจากขดลวดที่พันรอบแกนจะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงฟลักซ์ของหัวบันทึก และความกว้างของแทร็ค การบันทึกข้อมูลลงบนบัตรแม่เหล็กจะใช้วิธีการป้อนกระแสพัลส์ทั้งด้านบวกและด้านลบ พร้อมทั้งมีขนาดเพียงพอเข้าที่ขดลวดของหัวบันทึกที่วางอยู่ใกล้กับแถบแม่เหล็ก เมื่อป้อนกระแสพัลส์จะก่อให้เกิดสนามแม่เหล็กรอบบริเวณแก็ปของแกนหัวบันทึก ซึ่งสนามแม่เหล็กนี้จะใช้ในการบันทึกข้อมูล ดังแสดงในรูปที่ 2.10 ส่วนรูปแบบของกระแสพัลส์ แรงดันไฟฟ้า และขั้วแม่เหล็กในแถบแม่เหล็ก เมื่อมีการบันทึกแล้วแสดงในรูปที่ 2.11



รูปที่ 2.10 สนามแม่เหล็กรอบแก็ปของหัวบันทึก

จากรูปที่ 2.10 ค่าความกว้างแก็ปต้องมีค่าน้อยกว่าความกว้างของแทร็คข้อมูลเสมอ โดย ISO ได้ระบุความกว้างแก็ปของหัวบันทึกไว้ประมาณ 0.00625 มิลลิเมตร (0.00025 นิ้ว) หรือน้อยกว่า และค่าแก็ปของหัวอ่านมีค่าประมาณ 0.025 มิลลิเมตร (0.001 นิ้ว) หรือน้อยกว่า สำหรับค่าความหนาของเนื้อแถบแม่เหล็ก (δ) ISO ระบุไว้มีค่าไม่เกิน 0.038 มิลลิเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 กระแสพัลส์ รูปแบบของแทร็คที่ถูกบันทึก ฟลักซ์แม่เหล็กที่พื้นผิวแทร็คข้อมูลและลักษณะแรงดันไฟฟ้าของการบันทึก

2.2 การสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะกลุ่มของบิต คราวละหนึ่งบิตเรียงลำดับเรื่อยไปจนสิ้นสุด การสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมากเนื่องจากการสื่อสารข้อมูลแบบขนานมีการโอนย้ายมาพร้อมกัน จึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียงสองหรือสามเส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกลๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก

2.2.1 ความเร็วของการสื่อสารข้อมูลอนุกรม

เนื่องจากการสื่อสารข้อมูลแบบอนุกรมเป็นการรับ/ส่งข้อมูลในลักษณะกลุ่มของบิตข้อมูล (Bit Stream) ดังนั้น จึงต้องให้ความสนใจในการพิจารณาเรื่องอัตราเร็วในการรับ/ส่งบิตเหล่านี้เป็นอันดับแรก โดยทั่วไปมักจะระบุกันในหน่วยของจำนวนบิตข้อมูลภายในเวลาหนึ่งวินาที เรียกว่า อัตราบอด ตามค่ามาตรฐานเหล่านี้ ได้แก่ 110,150,300,1200,2400,4800,9600,19200 บอด ข้อมูลทั้งแปดบิตนี้หากว่าถูกส่งออกมาด้วยอัตรา 9600 บอด จะใช้เวลาในการส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ 1/9600 หรือ 104 us และเวลาในการส่งข้อมูลทั้งแปดบิตมีค่าเท่ากับ 8*104 หรือ 832 us

2.2.2 รูปแบบของการส่งข้อมูลอนุกรม

การสื่อสารอนุกรมแบบอะซิงโครนัสจะใช้การแปลงข้อมูลขนานให้เป็นอนุกรมแล้วเพิ่มเติมบิตบางอย่างร่วมไปกับการส่งข้อมูลจริงซึ่งได้แก่

1. บิตเริ่มต้น (Stat Bit)

บิตเริ่มต้นมีหน้าที่สำหรับการบ่งบอกให้ทราบถึงตำแหน่งเริ่มต้นก่อนบิตข้อมูล ตามปกติแล้วค่าของบิตเริ่มต้นจะเป็นระดับลอจิกต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. บิตแสดงภาวะความเป็นเลขคู่หรือเลขคี่ (Parity Bit)

บิตนี้มีหน้าที่เพื่อการตรวจสอบความถูกต้องของข้อมูลโดยทั่วไปมักเรียกว่าบิตพาริตี้และจะนำไปต่อท้ายบิตของข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของบิตที่เป็น 1 ซึ่งจะเป็นได้สองลักษณะคือ พาริตี้คู่ (Even Parity) หรือพาริตี้คี่ (Odd Parity) ตัวอย่างเช่นระบบที่ติดต่อกันโดยระบุว่าจะใช้พาริตี้คู่ ทางด้านส่งจะนำค่าข้อมูลที่จะส่งมาพิจารณาหาจำนวนของบิตที่มีค่า 1 หากเป็นเลขจำนวนคู่อยู่แล้ว ค่าของพาริตี้จะมีค่าเป็นศูนย์ แต่หากว่าจำนวนของบิตที่มีค่าเป็น 1 เป็นเลขจำนวนคี่ ค่าของพาริตี้ก็จะมีค่า 1 การพิจารณาทางด้านรับเป็นการตรวจสอบจำนวนบิตที่มีค่าเป็น 1 ของข้อมูลที่ได้รับมาทั้งหมดรวมทั้งบิตพาริตี้ ถ้ามีค่าเป็นเลขจำนวนคู่ แสดงว่าข้อมูลที่ได้รับเข้ามานี้ถูกต้องแต่หากไม่เป็นเลขจำนวนคู่แสดงว่าเกิดการผิดพลาดของข้อมูลขึ้น เป็นต้น

3. บิตสิ้นสุด (Stop Bit)

บิตสิ้นสุดเป็นบิตที่เพิ่มขึ้นเพื่อระบุถึงของเขตการสิ้นสุดของกลุ่มบิตข้อมูล บิตสิ้นสุดสามารถโปรแกรมได้คือ 1 บิต 1 ½ บิต และ 2 บิต ดังนั้นกรณีของการส่งข้อมูล 8 บิต หากข้อมูลถูกส่งออกไปด้วยอัตราเร็ว 9600 บอด เวลาโดยรวมในการส่งข้อมูลหนึ่งไบต์ จะมีค่าเป็น 12*104 หรือ1.25 ms

2.3 ไมโครคอนโทรลเลอร์ MCS-51

2.3.1 การใช้งานพอร์ทสื่อสารอนุกรม

พอร์ทสื่อสารอนุกรมมีโครงสร้างการทำงานในแบบที่เรียกว่าฟูลดูเพล็กซ์ (Full Duplex) สามารถรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน

Serial Port Buffer(SBUF) ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรมโดยมีอยู่ 2 ตัว

พอร์ทสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกทีละบิต SM1 และ SM0 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานของทั้ง 4 โหมด ของพอร์ทสื่อสารอนุกรม มีดังนี้

SM0	SM1	โหมด	การทำงาน
0	0	0	Shift Register ความเร็วในการรับหรือส่งข้อมูลเท่ากับ (1/12) ของ CPU Osc
0	1	1	8 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดได้จาก Timer1,2
1	0	2	9 Bit UART ความเร็วในการรับหรือส่งข้อมูล = (1/32) หรือ (1/64) เท่าของ CPU Osc โดยขึ้นกับบิต SMOD ใน PCON
1	1	3	9 BIT UART ความเร็วในการรับหรือส่งข้อมูลกำหนดที่ Timer1, 2

ตารางที่ 2.5 โหมดการทำงานของพอร์ทสื่อสารอนุกรม

โหมด 0 : พอร์ทสื่อสารอนุกรม 8 บิต โดยการส่งข้อมูลจะเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อนทางขา RxD เนื่องจากไม่มีการส่งบิตเริ่มต้น แต่จะส่ง shift clock ทางขา TxD (ความเร็ว 1/12 เท่าของ CPU Osc)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 1 : พอร์ทสื่อสารอนุกรม 10 บิต ข้อมูล 8 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของ Timer1 ,2

$$\text{บอดเรท โหมด 1,3} = \frac{2^{SMOD} \times \text{CPUOsc}}{32 \times 12 \times [256 - (TH1)]} \quad \text{โดยใช้ไทเมอร์ 1} \quad \ldots\ldots\ldots 2.1$$

$$\text{บอดเรท โหมด 1,3} = \frac{\text{CPUOsc}}{32 \times [65536 - (RCAP2H, RCAP2L)]} \quad \text{โดยใช้ไทเมอร์ 2} \quad \ldots\ldots\ldots 2.2$$

โหมด 2 : พอร์ทสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด (TB8 นิยมนำมาใช้ส่งพาริตี้บิต) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 หรือ 1/64 เท่าของ CPU Osc โดยขึ้นกับบิต SMOD ใน PCON

$$\text{บอดเรท (โหมด 2)} = \frac{(2^{SMOD})CPUOsc}{64} \quad \ldots\ldots\ldots 2.3$$

- บอดเรท (โหมด 2) = 1/32 CPU Osc เมื่อ SMOD = 1
- บอดเรท (โหมด 2) = 1/64 CPU Osc เมื่อ SMOD = 0

โหมด 3 : พอร์ทสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โฟลว์ของไทเมอร์ 1 หรือ อัตราโอเวอร์โฟลว์ของ Timer 2

$$\text{บอดเรท โหมด 1,3} = \frac{2^{SMOD} \times \text{CPUOsc}}{32 \times 12 \times [256 - (TH1)]} \quad \text{โดยใช้ Timer 1} \quad \ldots\ldots\ldots 2.4$$

$$\text{บอดเรท โหมด 1,3} = \frac{\text{CPUOsc}}{32 \times [65536 - (RCAP2H, RCAP2L)]} \quad \text{โดยใช้ Timer 2} \quad \ldots\ldots\ldots 2.5$$

2.3.2 กระบวนการรับและส่งข้อมูลอนุกรมของ MCS-51

การส่งข้อมูลออกทางพอร์ทอนุกรมของ MCS-51 จะเริ่มต้นขึ้นภายหลังเมื่อมีการเขียนข้อมูลลงไปใน SBUF ข้อมูลนี้ จะถูกเลื่อนทีละบิต และส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ได้ส่งออกไปครบถ้วนแล้วจะทำให้ค่าของแฟล็ก TI เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้ SBUF ว่างและพร้อมที่จะส่งข้อมูลไบต์ต่อไปแล้ว ในกรณีที่ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่รอให้แฟล็ก TI มีค่าเป็น 1 ก่อน จะมีผลทำให้ข้อมูลที่ส่งไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ทอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่า REN (Receive Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีข้อมูลภายนอกถูกส่งเข้ามายัง 8051 ทีละบิตจนครบ และเมื่อบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายถูกเลื่อนเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นจะถูกย้ายมาเก็บไว้ยังรีจิสเตอร์ SBUF และแฟล็ก RI ก็จะมีค่าเป็น 1 (ถูกเซ็ต)

2.3.2.1 พอร์ทอนุกรม (โหมด0)

การทำงานของพอร์ทอนุกรม (โหมด 0) เป็นการรับและส่งข้อมูลอนุกรมจำนวน 8 บิต โดยใช้เพียงขาสัญญาณ RxD เท่านั้น (ขานี้ใช้งาน 2 หน้าที่ใช้ส่งและรับข้อมูล) ส่วนขาสัญญาณ TxD จะนำไปใช้เพื่อเป็นขาสัญญาณนาฬิกาในการให้จังหวะ การเลื่อนข้อมูลกับวงจร เลื่อนบิตภายนอก สำหรับอัตราเร็วจะถูกกำหนดไว้คงที่ที่ค่า 1/12 เท่าของ CPU Osc จากรูปที่ 2.12 แสดงให้เห็นถึงแผนภาพเวลาสัญญาณต่างๆ ในโหมด 0 เมื่อมีการรับหรือส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกาในการเลื่อนบิตนี้จะเกิดภายในตัว 8051 เอง เนื่องจากโหมดนี้ไม่มีการส่งบิตเริ่มต้นและบิตสิ้นสุด ดังนั้นจึงจำเป็นที่จะต้องส่งสัญญาณ shift clock ออกไป เพื่อใช้ synchronize ระหว่างฝ่ายรับและฝ่ายส่ง โดยจะใช้ขา TxD ส่วนการรับข้อมูลจะรับข้อมูลเข้าทางขา RxD และรับ shift clock เข้าทางขา TxD ถ้า CPU Osc มีค่าเท่ากับ 12MHz ก็จะส่งได้ถึง 1 ล้านบอด ซึ่งโหมด 0 เป็นโหมดที่ส่งข้อมูลได้เร็วที่สุด รายละเอียดผังเวลาในการรับส่งดังแสดงในรูปที่ 2.12

2.3.2.2 พอร์ทอนุกรม (โหมด1)

การทำงานในโหมด 1 เป็นการสื่อสารข้อมูลอนุกรมจำนวน 10 บิต ประกอบด้วยบิตเริ่มต้น 1 บิต ข้อมูลจำนวน 8 บิต และบิตสิ้นสุดอีก 1 บิต ดังแสดงในรูปที่ 2.13 โดยข้อมูลจะถูกส่งออกทาง TxD และรับเข้ามาทางขาสัญญาณ RxD ในส่วนของข้อมูล 8 บิต ที่ได้รับหรือทำการส่งออก จะเป็นบิตนัยสำคัญต่ำเป็นลำดับแรก ส่งทางฝ่ายรับค่าของบิตสิ้นสุด จะส่งเข้ามาจัดเก็บไว้ในบิต RB8 ภายในรีจิสเตอร์ SCON สำหรับอัตราเร็วในการส่งข้อมูลในการส่งข้อมูลของโหมด 1 นั้น สามารถกำหนดเลือกได้จากไทเมอร์ 1 ผังเวลาการทำงานแสดงดังในรูปที่ 2.13

2.3.2.3 พอร์ทอนุกรม (โหมด2)

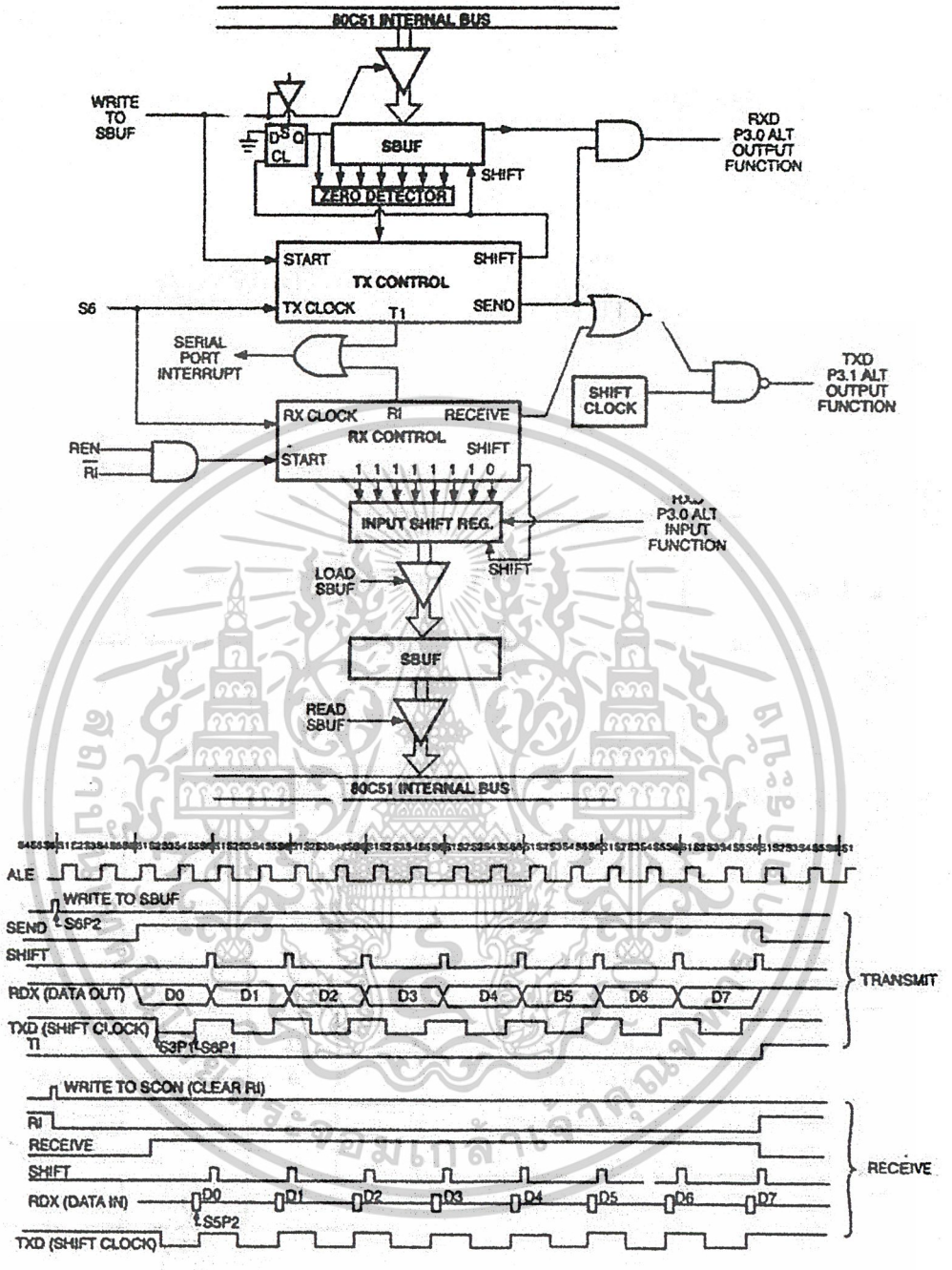
โหมดนี้ใช้ทั้งหมด 11 บิต โดยแบ่งเป็น บิตเริ่มต้น, 9 บิตข้อมูล และบิตสิ้นสุด โดยบิตที่ 9 ผู้ใช้สามารถกำหนดค่าเองได้ว่าจะส่งค่าอะไรออกไป โดยจะต้องนำไปใส่ไว้ในบิต TB8 ในรีจิสเตอร์ SCON ส่วนมากผู้ใช้มักจะนำบิตนี้มาใช้เป็นพาริตี้บิต โดยหลักค่ามาจากพาริตี้แฟล็กใน PSW ส่วนทางด้านรับบิตที่ 9 จะถูกนำมาเก็บไว้ใน RB8 อัตราเร็วในการส่ง/รับข้อมูลขึ้นกับCPU Osc และค่า SMOD ซึ่งอยู่ในบิต 7 ใน SCON ผังเวลาการทำงานแสดงดังในรูปที่ 2.14

2.3.2.4 พอร์ทอนุกรม (โหมด3)

เหมือนกับโหมด 2 ทุกอย่าง ยกเว้นความเร็วในการรับส่งข้อมูลจะขึ้นกับอัตราโอเวอร์โฟลว์ของไทเมอร์ 1 หรือ ไทเมอร์ 2 โดยมีผังการทำงานดังรูปที่ 2.15

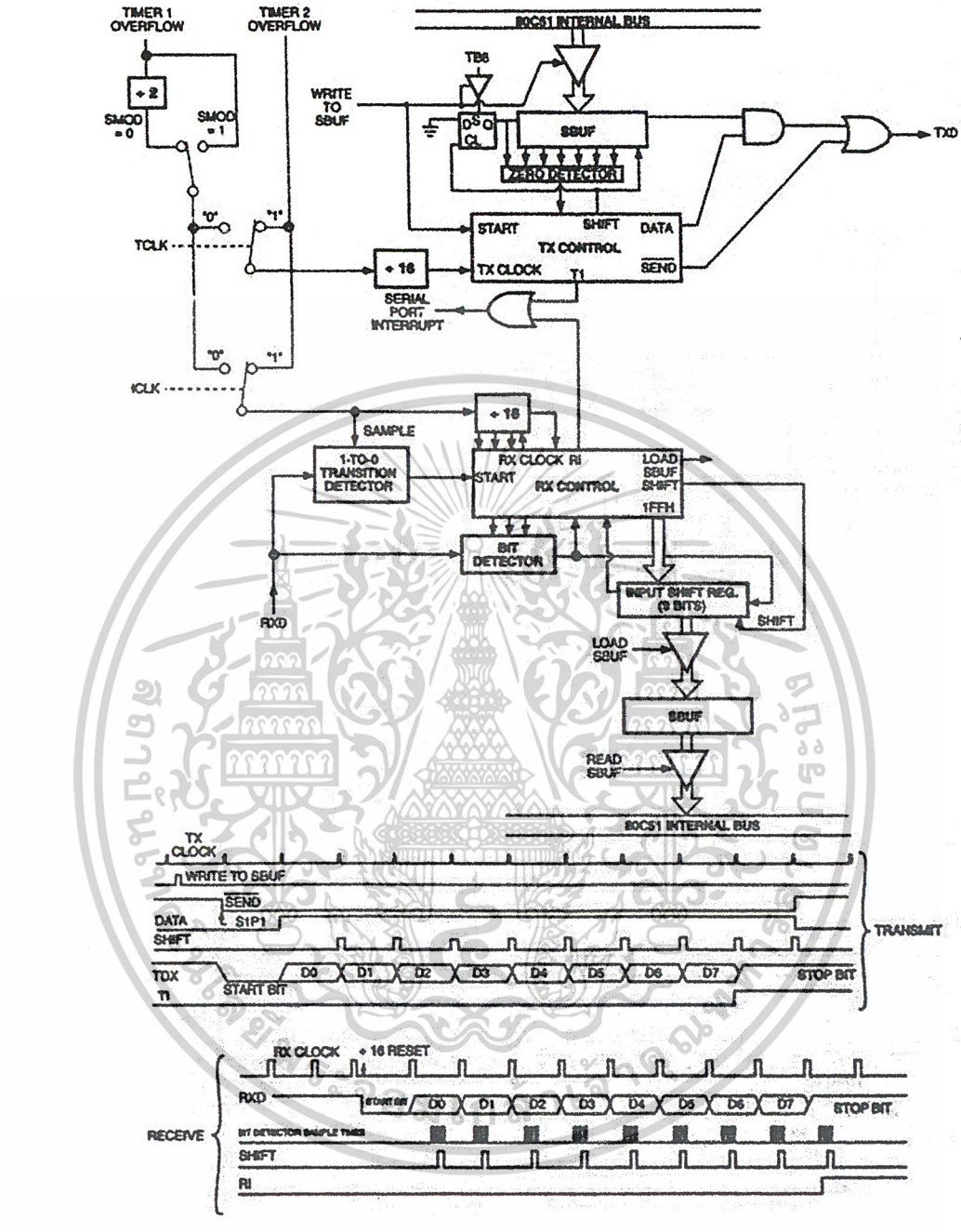
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



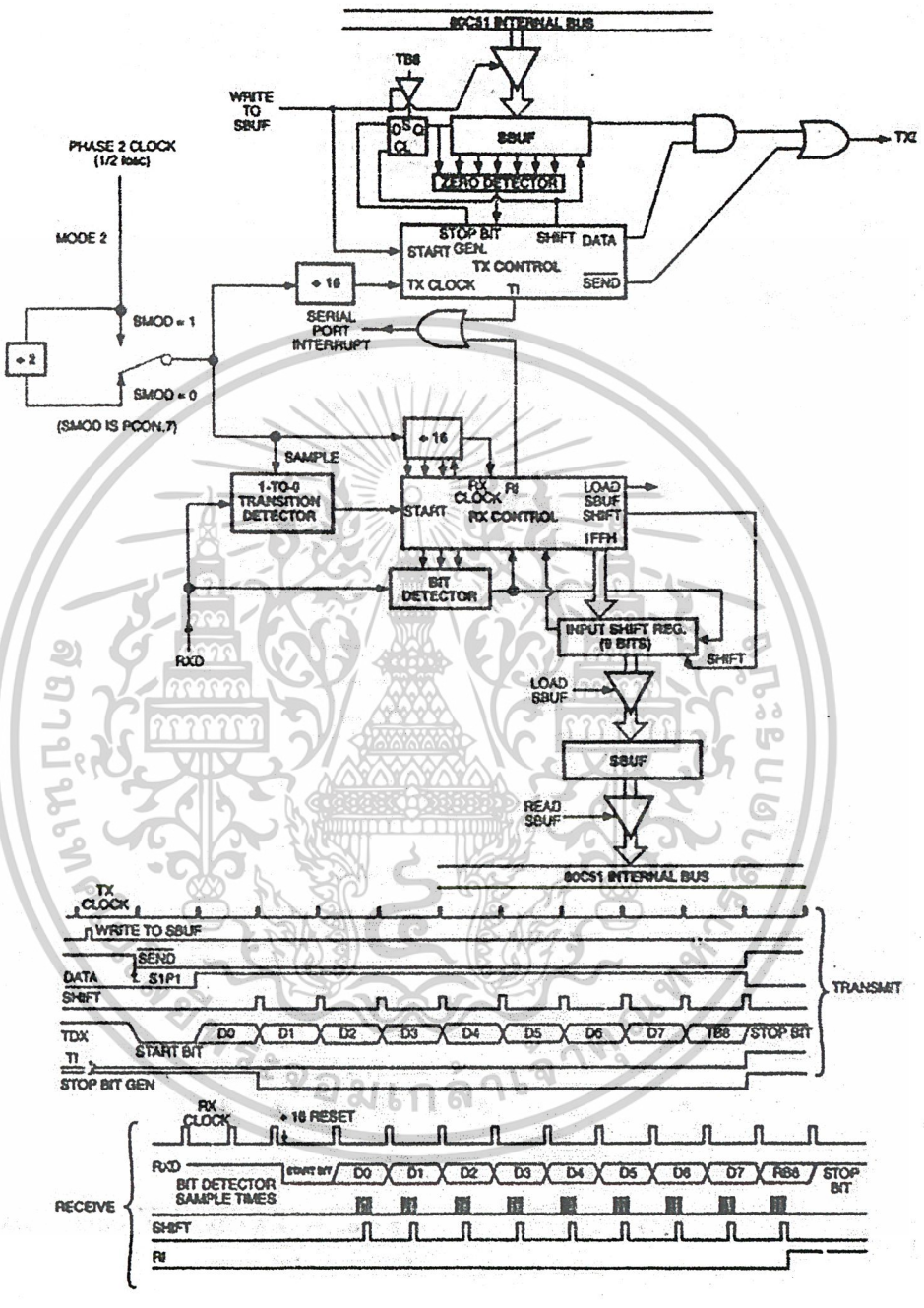
รูปที่ 2.12 ผังการทำงานโหมด 0

61357
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



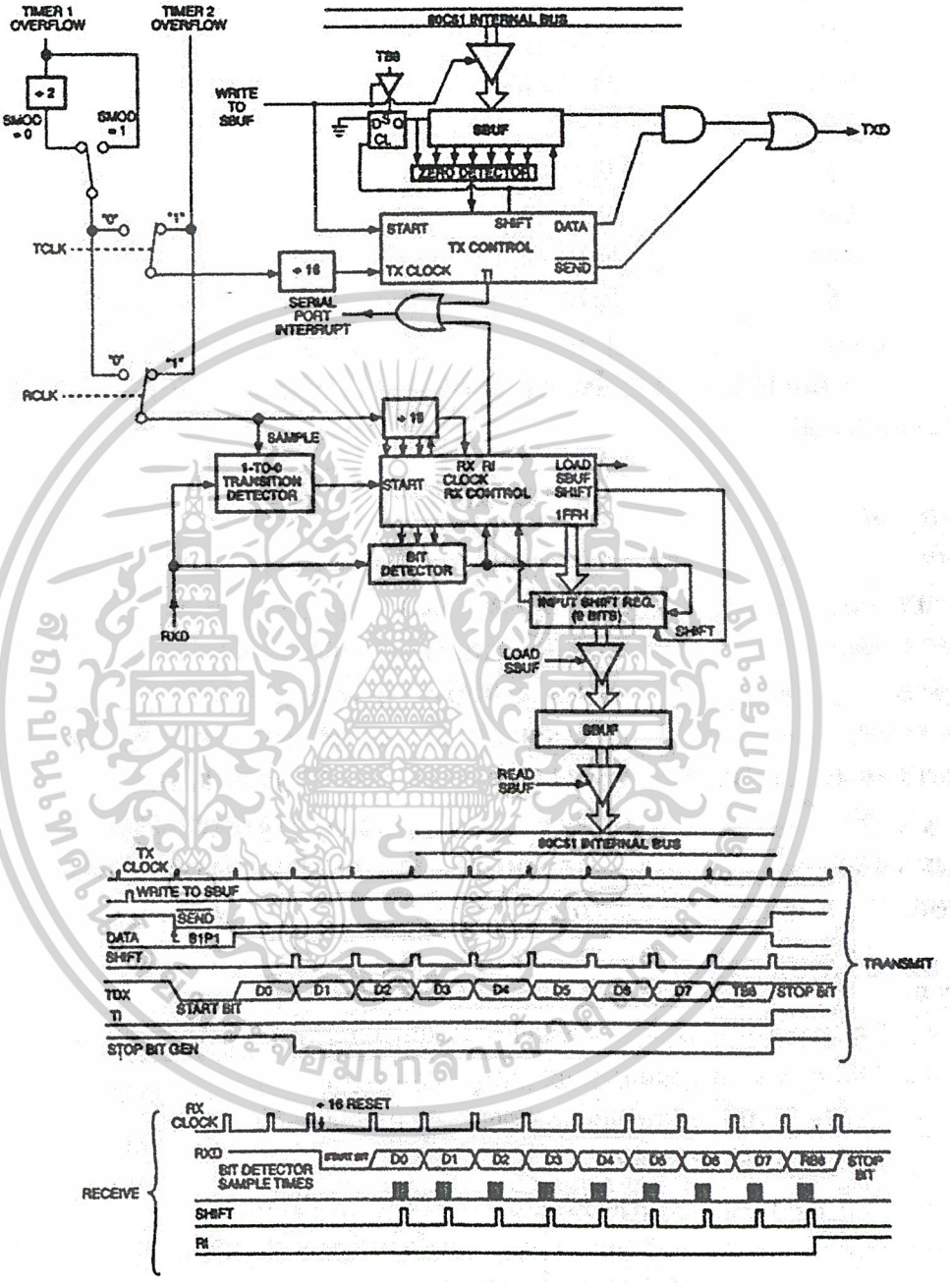
รูปที่ 2.13 ผังการทำงานโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ผังการทำงานโหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ผังการทำงานโหมด 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 รีจิสเตอร์ที่ใช้ควบคุมการรับส่งข้อมูลอนุกรม SCON

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0,SM1 : บิตเลือกโหมดการทำงาน

SM2 : บิตเลือกการทำงานแบบ Single Processor Environment หรือ Multiprocessor Environment

1 : เลือก Multiprocessor Environment ใช้ได้กับโหมด 2,3

0 : เลือก Single Processor Environment ใช้ได้กับทุกโหมด

REN(Receive Enable) : บิตควบคุมให้รับหรือไม่รับข้อมูล

1 : ให้รับข้อมูลได้

0 : ห้ามรับข้อมูล

TB8 (Transmit bit D8) : ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2,3 ให้ใส่ในบิตนี้

RB8 (Receive bit D8) : ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมาเก็บในบิตนี้

TI : แฟลกซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

RI : แฟลกซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์

2.4 การประมวลผลภาพเชิงตัวเลข (Digital Image Processing)

2.4.1 การแทนภาพด้วยข้อมูลแบบดิจิตอล

ภาพข้อมูลแบบดิจิตอล (Digital Image) เป็นภาพที่ถูกแปลงมาจากอนาลอก ให้อยู่ในรูปของตัวเลข โดยภาพอนาลอกถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็กๆ ที่เรียกว่าพิกเซล (Pixel) ในแต่ละพิกเซลจะถูกระบุตำแหน่งด้วยคู่โคออดิเนต (x,y) และค่าระดับสีเทาของพิกเซลนั้นๆ โดยเราสามารถแปลงภาพเป็นข้อมูลแบบดิจิตอลได้โดยมีขั้นตอนและวิธีการดังนี้

เมื่อเรานำสัญญาณอนาลอกที่ต้องการประมวลผลมาผ่านส่วนที่เรียกว่า ดิจิไตเซอร์ (Digitizor) ซึ่งจะมีหน้าที่ในการเปลี่ยนสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอล ไม่ว่าจะเป็นสแกนเนอร์ หรือกล้องดิจิไตเซอร์ จากนั้นทำการควอนไตซ์ (Quantizing) เพื่อที่จะประมวลผลสัญญาณด้วยระบบคอมพิวเตอร์ ฟังก์ชันของภาพ f(x,y) จะถูกทำให้เป็นสัญญาณไม่ต่อเนื่องทั้งระนาบของภาพ ซึ่งเราเรียกว่าการสุ่มภาพ (Image Sampling) ของฟังก์ชันที่ได้เรียกว่า การควอนไตซ์ระดับสีเทา (Gray Level Quantization) ก็จะได้ข้อมูลที่เป็นดิจิตอล

สมมติว่าสัญญาณภาพต่อเนื่อง f(x,y) ถูกดิจิไตซ์ในระนาบ x และ y เป็นช่วงเท่าๆกัน เราสามารถจัด f(x,y) ให้อยู่ในรูปของเมตริกซ์ขนาด M*N ได้ดังสมการที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ . & . & . & . & \\ . & . & . & . & \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{matrix} \quad \dots\dots\dots 2.6$$

ซึ่งทางขวาของสมการ จะเรียกได้ว่าข้อมูลภาพดิจิตอล และทุกๆสมาชิกของเมตริกซ์ จะเรียกว่าพิกเซลจากขบวนการสร้างภาพดิจิตอลดังข้างต้น จะเห็นได้ว่าเราสามารถทราบขนาด ของความละเอียดของภาพ N*N พิกเซล และจำนวนระดับเกรย์สเกล ในทางปฏิบัติการทำควอนไตเซชันในระบบภาพดิจิตอลจะมีค่าดังสมการที่ 2.7

$$B = N \times N \times M \text{ บิต} \quad \dots\dots\dots 2.7$$

เมื่อ B = ขนาดของข้อมูลภาพที่เป็นดิจิตอล

G = จำนวนของเกรย์สเกลที่ต้องการใช้ในการเก็บข้อมูลภาพ

M = จำนวนบิตที่ใช้ในการแทนข้อมูลภาพ 1 พิกเซล

โดย M สามารถหาได้จาก $G = 2^M$

2.4.2 ลักษณะการจัดเก็บข้อมูลภาพแบบดิจิตอล

โดยทั่วไปแล้วข้อมูลภาพจะมีความเข้มตั้งแต่ 2 ระดับขึ้นไป แต่ที่ใช้กันมากจะใช้กันที่ค่าระดับความเข้มของจุดภาพเท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของจุดภาพอยู่ในช่วง (0-255) โดยใช้เนื้อที่ในการเก็บข้อมูลขนาด 1 ไบต์หรือ 8 บิต สำหรับข้อมูล 1 จุดภาพ ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มสูงๆ อาจจะต้องการจำนวนบิตสำหรับการเก็บข้อมูลมากกว่า 8 บิต คืออาจจะเป็น 16 หรือ 24 บิต โดยจะแยกความแตกต่างของแต่ละระดับให้เห็นอย่างชัดเจนได้ดังนี้

1.ภาพ 2 ระดับ (binay image) คือมีเพียงแค่จุดขาวกับจุดดำเท่านั้น โดยแต่ละจุดภาพเป็นข้อมูลขนาด 1 บิต

2. ภาพ 16 ระดับ คือในแต่ละจุดภาพจะมีขนาดของข้อมูล 4 บิต ซึ่งทำให้สามารถแสดงได้ถึง 16 ระดับสี

3. ภาพ 256 ระดับ คือในแต่ละจุดภาพจะมีขนาดของข้อมูล 8 บิต ซึ่งทำให้สามารถแสดงภาพได้ความเข้มถึง 256 ระดับสี

4. ภาพทรูคัลเลอร์ (True Color) คือในแต่ละจุดภาพจะมีขนาดข้อมูล 24 บิต ทำให้สามารถแสดงภาพออกมาได้เหมือนจริงมากที่สุด เพราะสามารถแสดงสีได้ถึง 16,777,216 สี ภาพทรูคัลเลอร์สามารถแสดงได้เฉพาะภาพสีเท่านั้น ไม่สามารถแสดงเป็นภาพขาวดำได้

โดยทั่วไปวิธีการประมวลผลภาพเชิงตัวเลขที่ทำให้คอมพิวเตอร์สามารถรู้จักวัตถุภายในภาพได้นั้น พอจะแบ่งได้สองระดับด้วยกันคือ การประมวลผลภาพในระดับต่ำ (Low Level Image Processing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประมวลผลในระดับต่ำจะเป็นการประมวลผลในเชิงตัวเลขเกือบทั้งหมด เพื่อหาตัวแปรต่างๆ มาอธิบายข้อมูลภาพ โดยมีจุดประสงค์เพื่อนำตัวแปรเหล่านั้นไปประมวลผลในระดับสูงต่อไป โดยทั่วไปแล้วการประมวลผลภาพระดับต่ำจะประกอบด้วย การประมวลผลภาพก่อน (Preprocessing) การกำจัดสัญญาณรบกวน หรือการทำให้ภาพคมชัด การหาขอบภาพ เป็นต้น

การประมวลผลในระดับสูงเป็นการนำผลลัพธ์ หรือสัญลักษณ์ที่ได้จากการประมวลผลระดับต่ำมาตีความ หรือประมวลผลเพื่อให้คอมพิวเตอร์สามารถรู้จักและเข้าใจภาพได้ สำหรับความแตกต่างของการประมวลผลระดับต่ำและระดับสูงนั้นคือ ข้อมูลที่นำมาใช้ในการประมวลผลภาพโดยที่การประมวลผลภาพระดับต่ำจะใช้ความสว่างของจุดโดยตรง ส่วนการประมวลผลภาพในระดับสูงนั้นข้อมูลของภาพที่นำมาประมวลผลภาพจะถูกแสดงในรูปของสัญลักษณ์ ซึ่งสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่างๆ ที่มีอยู่ในภาพ เช่น ขนาดวัตถุ รูปร่าง และความสัมพันธ์กันระหว่างวัตถุภายในภาพ

2.4.3 ไฟล์ข้อมูลภาพชนิดบิตแมป

2.4.3.1 รูปแบบของไฟล์ข้อมูลชนิดบิตแมป

รูปแบบของไฟล์ข้อมูลชนิดบิตแมป เป็นฟอร์แมตของวินโดว์บิตแมป ซึ่งเป็นมาตรฐานสำหรับไฟล์กราฟฟิกบนวินโดว์ ซึ่งจะใช้ในการตัดต่อ หรือสำเนาภาพต่างๆ ลงบนคลิปบอร์ด (Clipboard)

2.4.3.2 โครงสร้างของไฟล์ของข้อมูลภาพชนิดบิตแมป

โครงสร้างของไฟล์ข้อมูลภาพชนิดบิตแมป จะประกอบด้วย 3 ส่วนคือ

1.ข้อมูลเฮดเดอร์ (Header)

2. ข้อมูลจานสี (Palette)

3. ข้อมูลภาพ (Data)

1.ข้อมูลเฮดเดอร์ คือ ข้อมูลที่อยู่บริเวณส่วนหัวของไฟล์ ซึ่งประกอบด้วยข้อมูลที่บอกรายละเอียดต่างๆ ของภาพ เช่น ความกว้าง ความยาวของภาพ จำนวนสี จำนวนบิต ความละเอียด เป็นต้น

2. ข้อมูลจานสี คือ ข้อมูลที่บอกถึงชุดของจานสีที่เกิดจากการผสมแม่สีทั้งสาม คือ แดง เขียว และน้ำเงิน มาผสมกันได้เป็นสีต่างๆ ตามจำนวนสีของภาพ เช่น รูปขนาด 4 บิตจะมี 16 ระดับสี รูป 8 บิต จะมีขนาด 256 ระดับสี เป็นต้น ซึ่งถ้ามีจำนวนสีน้อยๆ ก็จะมีการเก็บค่าจานสีนี้ลงไฟล์ด้วย แต่ถ้ารูปประเภท 24 บิตจะไม่มีค่าจานสี แต่จะใช้วิธีการเก็บค่าแม่สีทั้งสามลงไปเป็นข้อมูลแทน เพราะถ้าเก็บค่าจานสีที่มีถึง 16.7 ล้านสีลงไปด้วยจะเปลืองพื้นที่มาก ข้อแตกต่างที่สำคัญของบิตแมปขนาดนี้คือ ไฟล์บิตแมปจะเก็บค่าจานสีชุดละ 4 ไบต์ แต่ก็ใช้แค่ 3 ไบต์เช่นกันคือ แดง เขียว และน้ำเงิน

3. ข้อมูลภาพ คือ ข้อมูลสีของภาพแต่ละจุดที่มาประกอบกันเป็นรูปภาพ ซึ่งค่าที่เก็บนี้จะเป็นค่าที่ใช้ในการชี้ตารางข้อมูลจานสีหมายเลขอะไร เช่น จุดแรกมีค่าเป็น 10 ก็ให้ไปเปิดตารางข้อมูลจานสีหมายเลข 10 สมมติว่าแม่สีเป็น R=0,G=0 และ B=100 ก็จะได้จุดนี้เป็นสีน้ำเงิน ซึ่งถ้าเป็นกรณีของรูป 24 บิต จะเป็นการอ่านข้อมูลขึ้นมา 3 ค่า เป็นค่าของแม่สี RGB

2.4.3.3 การจัดเก็บไฟล์ข้อมูลชนิดบิตแมป

การจัดเก็บไฟล์ข้อมูลชนิดบิตแมป มีการเก็บอยู่ 2 แบบ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1)แบบบีบอัดข้อมูล

- RLE 4 เป็นการบีบอัดข้อมูลแบบ Run-length Encoder แบบ 4 บิต
- RLE 8 เป็นการบีบอัดข้อมูลแบบ Run-length Encoder แบบ 8 บิต

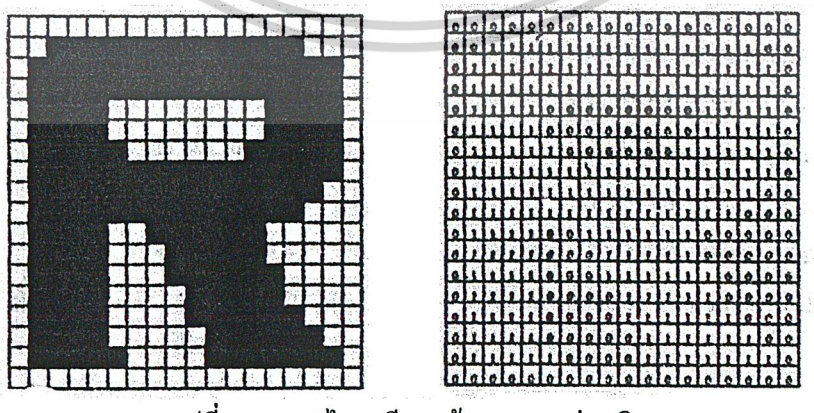
2) แบบไม่มีการบีบอัดข้อมูล

เป็นการเก็บข้อมูลจริงของสีของพิกเซล ซึ่งทำให้ขนาดไฟล์ค่อนข้างใหญ่ แต่จะทำการแสดงภาพได้เร็วกว่า เพราะไม่ต้องเสียเวลาในการคลายข้อมูล

2.4.4 การสร้างภาพไบนารี

อุปกรณ์ที่มีความสามารถในการแสดงผลได้แค่ 2 ระดับ หรือ 2 สี คือ สีขาวกับสีดำยังมีการใช้กันอย่างแพร่หลาย เช่น เครื่องพิมพ์ (Printer) เครื่องโทรสาร (Fax) จอภาพแสดงผลแบบโมโนโครม (Monochrome Monitor) เป็นต้น เนื่องจากอุปกรณ์เหล่านี้เป็นอุปกรณ์ที่มีราคาถูก ดังนั้นการที่จะแสดงผลหรือพิมพ์รูปภาพที่มีระดับความเข้มของภาพหลายระดับซึ่งมีค่ามากกว่าความสามารถในการแสดงผลของอุปกรณ์เหล่านั้นที่มีเพียงแค่ 2 ระดับเท่านั้น

จะเห็นได้ว่าการที่จะแก้ปัญหาการแสดงผลภาพที่มีความเข้มหลายระดับบนอุปกรณ์ที่สามารถแสดงผลได้ 2 ระดับนั้น จะต้องทำการแปลงข้อมูลที่มีความเข้มหลายระดับ (Multi Level Image) ให้เป็นภาพที่มีระดับความเข้มเพียง 2 ระดับ นั่นคือ 1 จุดภาพมีได้ 2 ค่าเท่านั้น คือ 0 กับ 1 โดยจุดภาพที่แทนด้วย 0 จะหมายถึงจุดภาพที่มีสีดำ ส่วนจุดที่แทนด้วย 1 จะหมายถึงจุดภาพที่มีสีขาว เมื่อทำการแปลงเป็นภาพไบนารีแล้วจึงนำภาพนั้นไปแสดงผลบนอุปกรณ์เหล่านั้น จะเห็นได้ว่าการแปลงข้อมูลภาพหลายระดับเป็นภาพไบนารีจึงมีความจำเป็นและมีประโยชน์มากในการแสดงผลภาพที่มีระดับความเข้มของภาพหลายระดับบนอุปกรณ์ที่มีความสามารถในการแสดงผลได้ 2 ระดับ สำหรับประโยชน์อีกประการหนึ่งในการแปลงข้อมูลภาพนั้นเป็นภาพไบนารี คือการลดเนื้อที่การเก็บข้อมูลภาพจะใช้ในการเก็บ 8 บิต เมื่อสร้างเป็นภาพไบนารีแล้วสามารถลดลงได้ถึง 8 เท่า นั่นคือ 1 จุดภาพจะใช้เนื้อที่ในการเก็บ 1 บิต อีกทั้งยังสามารถนำไปประยุกต์ใช้งานได้อย่างแพร่หลาย เช่น นำไปประยุกต์ใช้ในการวิเคราะห์เอกสารในขั้นตอนที่เรียกว่า การประมวลผลขั้นต้น (Preprocessing) เป็นต้น



รูปที่ 2.16 ภาพไบนารีและข้อมูลของแต่ละพิกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสร้างภาพไบนารี สามารถทำได้โดยใช้เทคนิคการทำเทรชโฮล (Thresholding Techinique) โดยพิจารณาว่าจุดภาพใดควรจะเป็นจุดขาวหรือจุดดำจะกระทำโดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่งที่เรียกว่า "ค่าเทรชโฮล" (Threshold Value) เทคนิคนี้ใช้กันมากในกรณีที่ข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุ (Object) และพื้นหลัง (Background) โดยค่าของจุดภาพใดๆ ที่มีค่าน้อยกว่าค่าเทรชโฮลจะถูกกำหนดให้เป็น 0 (จุดดำ) และถ้ามากกว่าค่าเทรชโฮลจะถูกกำหนดให้เป็น 1 (จุดขาว) ซึ่งการทำงานสามารถแสดงได้ดังสมการที่ 2.8

$$b(x,y) = \begin{cases} 0 \; ; \; g(x,y) < Thr \\ \\ 1 \; ; \; g(x,y) \geq Thr \end{cases} \qquad \text{.....2.8}$$

b(x,y) : ข้อมูลภาพผลลัพธ์เป็นภาพไบนารี

g(x,y) : ข้อมูลภาพอินพุทที่มีระดับความเข้ม 0 ถึง 1 ระดับ

Thr : ค่าเทรชโฮลเป็นค่าคงที่ที่อยู่ระหว่าง 0 ถึง L ระดับ

0 : จุดดำ

1 : จุดขาว

โดยที่ L : ระดับความเข้มของจุดภาพสูงสุด

ในการสร้างภาพไบนารีโดยใช้เทคนิคเทรชโฮลเพื่อให้ได้ผลลัพธ์ที่ได้เหมาะสมและคมชัดสิ่งที่สำคัญที่สุดคือ ค่าเทรชโฮล เนื่องจากถ้าเลือกค่าเทรชโฮลไม่เหมาะสม (ค่าเทรชโฮลที่มีค่าน้อยเกินไปหรือมีค่ามากเกินไป) ภาพที่ได้อาจไม่เหมาะสม ขาดความคมชัดและรายละเอียดบางส่วนขาดหายไป กล่าวคือภาพที่ได้อาจจะมืดเกินไป (จุดดำมากเกินไป) หรือสว่างมากเกินไป (จุดขาวมากเกินไป) หรือภาพที่ได้มีสิ่งรบกวนเกิดขึ้น (Noise) เกิดขึ้น อันเป็นผลให้ภาพผลลัพธ์ที่ได้ไม่สวยงานเท่าที่ควร ดังนั้นปัญหาของการสร้างภาพไบนารีโดยวิธีเทรชโฮลนี้คือ ทำอย่างไรจึงจะสามารถคำนวณหาค่าเทรชโฮลที่เหมาะสมสำหรับแต่ละภาพที่จะนำมาทำการสร้างภาพไบนารี ซึ่งมีวิธีการคำนวณหาค่าเทรชโฮลหลายวิธี โดยแต่ละวิธีเหมาะสมสำหรับการทำงานที่ต่างไป เช่นการหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า (Preassigned Threshold Value) การหาค่าเทรชโฮลจากค่ากลาง (Mid-Range Threshold Value) แต่ละวิธีอธิบายได้ดังนี้

การหาค่าเทรชโฮลโดยการกำหนดค่าล่วงหน้า (Preassigned Threshold Value) การหาค่าเทรชโฮลโดยวิธีการกำหนดล่วงหน้าเป็นวิธีที่ง่ายที่สุด เป็นการคำนวณค่าเทรชโฮลโดยการกำหนดเองจากผู้ใช้ซึ่งการกำหนดนี้ขึ้นอยู่กับประสบการณ์ของผู้ใช้นั้นๆ โดยการเลือกค่าคงที่ค่าหนึ่ง ซึ่งเรียกค่านั้นว่า ค่าเทรชโฮล โดยค่าที่เลือกมานี้จะเป็นค่าที่อยู่ระหว่างค่าต่ำสุดและค่าสูงสุดของระดับความเข้มของข้อมูลภาพอินพุท เช่น ภาพข้อมูลอินพุทมีเกรย์สเกล 256 ระดับ จะมีค่าเกรย์สเกลได้ตั้งแต่ 0-255 เมื่อเลือกค่าเทรชโฮลได้แล้วสามารถสร้างภาพไบนารีได้ดังสมการ 2.8

การหาค่าเทรชโฮลจากค่ากลาง (Mid-Range Threshold Value) การหาค่าเทรชโฮลโดยพิจารณาจากค่ากลาง เป็นการหาค่าเทรชโฮลที่แตกต่างจากการหาค่าเทรชโฮลวิธีแรก สำหรับวิธีนี้จะเป็นการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนวณหาค่าโดยอัตโนมัติ โดยไม่ต้องให้ผู้ใช้เป็นผู้กำหนด การหาค่าเทรชโฮลวิธีนี้ได้อาศัยการคำนวณพื้นฐานทางสถิติในเรื่องของการหาค่ากลางหรือค่าเฉลี่ย (Mean) มาประยุกต์ใช้ ค่าเทรชโฮลที่คำนวณได้จะเป็นค่าที่ได้จากกค่ากึ่งกลางที่อยู่ระหว่างค่าระดับความเข้มสูงสุดและค่าความเข้มต่ำสุดของข้อมูลภาพอินพุท สำหรับการคำนวณหาค่ากึ่งกลางนี้สามารถคำนวณได้จากสมการที่ 2.9

$$Thr = \frac{\text{Maximum}\left(g\left(x,y\right)\right)+\text{Minimum}\left(g\left(x,y\right)\right)}{2} \qquad \text{.....2.9}$$

โดยที่ Thr : ค่าเทรชโฮล

g(x,y) : ข้อมูลภาพอินพุทที่มีระดับความเข้ม 0 ถึง L ระดับ

Maximum(g(x,y)) : ค่าสูงสุดเกรย์สเกลของข้อมูลอินพุท

Minimum(g(x,y)) : ค่าต่ำสุดเกรย์สเกลของข้อมูลอินพุท

เมื่อทำการคำนวณค่าเทรชโฮลได้แล้ว ก็สามารถสร้างภาพไบนารีได้โดยนำค่าเทรชโฮลที่ได้มาแทนค่าในสมการ 2.8

การหาค่าเทรชโฮลจากค่าเฉลี่ยเลขคณิต หาได้จากสมการที่ 2.10

$$Thr = \frac{\sum_{i=0}^{N\times N} g_i(x,y)}{N\times N} \qquad \text{.....2.10}$$

2.4.5 การแบ่งส่วนภาพ (Image Segmentation)

บางครั้งการวิเคราะห์ภาพจะเน้นที่วัตถุ (Object) บนภาพ ซึ่งวัตถุเหล่านั้นจะถูกแยกแยะออกได้ด้วยวิธีการที่หลากหลายตั้งแต่วิธีการง่ายๆ เช่นการหาส่วนที่เป็นแนวของวัตถุ หรือส่วนที่เป็นขอบของวัตถุ ไปจนกระทั่งวิธีการที่สลับซับซ้อน สำหรับวิธีในการแบ่งส่วนภาพได้แก่

2.4.5.1 การตรวจรู้แนวเส้น (Line Detection)

สำหรับวิธีนี้เป็นการเปรียบเทียบค่าระดับสีเทาระหว่างจุดภาพที่พิจารณากับจุดภาพข้างเคียง ทั้งนี้จะเพิ่มน้ำหนักหรือให้ความสำคัญของจุดภาพข้างเคียงที่มีการเรียงตัวร่วมกับจุดภาพที่พิจารณาเป็นเส้นตรงรวม 4 แบบ คือ เส้นตรงในแนวดิ่ง เส้นตรงในแนวนอน เส้นตรงในแนวทแยงจากล่างซ้ายขึ้นไปทางบนขวา หรือเรียกว่าเส้นทแยง +45องศา และเส้นตรงในแนวทแยงจากบนขวาลงไปทางล่างซ้าย หรือเรียกว่าเส้นทแยง -45 องศา หมายความว่าแนวเส้นที่จะตรวจรู้เป็นแนวเส้นตั้ง แนวเส้นนอน และเส้นเอียงทำมุม +45 องศา หรือ -45 องศา เท่านั้น

กระบวนการตรวจรู้จะทำได้โดยนำหน้ากาก (mask) ขนาด 3x3 (อาจใช้ขนาดใหญ่กว่านี้ เช่น 5x5 หรือ 7x7 ก็ได้ แต่จะใช้เวลาในการประมวลผลเพิ่มมากขึ้น) ไปทาบกับจุดภาพในตำแหน่งที่จะทำการประมวลผล จากนั้นจะทำการคำนวณหาค่าตามสมการที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R = \sum_{k=1}^{9} W_k Z_k \qquad \text{.....2.11}$$

โดยที่ W_k เป็นค่าน้ำหนักของแต่ละจุดบนหน้ากาก ดังรูปที่ 2.17

Z_k เป็นค่าระดับสีเทาของจุดภาพในตำแหน่งที่ตรงกับจุดบนหน้ากาก กล่าวคือ

$Z_1 = F(x-1,y-1)$ $Z_2 = F(x,y-1)$ $Z_3 = F(x+1,y-1)$

$Z_4 = F(x-1,y)$ $Z_5 = F(x,y)$ $Z_6 = F(x+1,y)$

$Z_7 = F(x-1,y+1)$ $Z_8 = F(x,y+1)$ $Z_9 = F(x+1,y+1)$

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

ก) หน้ากากทั่วไป

-1	-1	-1
2	2	2
-1	-1	-1

ข) หน้ากากแนวนอน

-1	2	-1
-1	2	-1
-1	2	-1

ค) หน้ากากแนวตั้ง

-1	-1	2
-1	2	-1
2	-1	-1

ง) หน้ากากแนว $+45^\circ$

2	-1	-1
-1	2	-1
-1	-1	2

จ) หน้ากากแนว -45°

รูปที่ 2.17 หน้ากากที่ใช้ในการหาแนวเส้น

ในการตรวจหาแนวเส้นของภาพ จะนำหน้ากากไปทาบบนจุดต่างๆ จนครบทั้งภาพ แล้วคำนวณตามสมการที่ 2.11 จะได้ค่า R มา 1 ค่า คือ R_1แล้วเปลี่ยนหน้ากากไปเรื่อยๆ ตามรูปที่ 2.17 จนครบ 4 อัน จะได้ค่า R_2,R_3 และ R_4 ตามลำดับ จากนั้นทำการเปรียบเทียบค่า R ทั้ง 4 ค่า ถ้า $|R_i| > |R_j|$ ที่ทุกค่าของ $j \neq i$ จุดภาพนั้นจะถูกตัดสินว่าเป็นส่วนหนึ่งของแนวเส้นตามแบบของหน้ากากที่ i

บางครั้ง มีความต้องการหาแนวเส้นลักษณะใดลักษณะหนึ่ง ก็อาจทำได้โดยเลือกใช้หน้ากากแนวที่ต้องการทาบบนจุดภาพ และหาผลรวมตามสมการที่ 2.11 จากนั้นนำค่าที่ได้ไปเปรียบเทียบกับค่า เทรชโฮลที่ได้กำหนดไว้แล้ว ถ้าค่าที่ได้สูงกว่าค่าเทรชโฮล จุดภาพนั้นจะถูกตัดสินว่าเป็นส่วนหนึ่งของเส้น

2.4.5.2 การตรวจหาขอบ (Edge detection)

องค์ประกอบหรือวัตถุที่ถูกบันทึกภาพจะมีตำแหน่ง การจัดวาง ที่ทำให้ให้เกิดการสะท้อนแสดงที่ต่างกัน ค่าความเข้มแสงหรือค่าระดับสีเทาที่ปรากฏบนภาพ จะต้องมีขอบเขตองค์ประกอบนั้นๆ แสดงให้เห็นชัดเจนหรือค่อนข้างชัดเจน เนื่องจากบริเวณดังกล่าวมีค่าระดับสีเทาแตกต่างกัน ดังนั้นการตรวจหาขอบของวัตถุบนภาพ จึงใช้หลักการพิจารณาการเปลี่ยนค่าระดับสีเทา ที่มี 2 แนวทางคือ การพิจารณาอัตราหรือค่าความชันในการเปลี่ยนแปลงค่าระดับสีเทา ซึ่งเรียกว่าวิธีเกรเดียนต์ (Gradient Method) และการพิจารณาอัตราการเปลี่ยนแปลงค่าความชันในการเปลี่ยนแปลงค่าระดับสีเทา ซึ่งเราเรียกว่าวิธีลาปลาเซียน (Laplacian Method)

1. วิธีเกรเดียนต์ (Gradient Method)

จะใช้กับภาพที่มีองค์ประกอบส่วนใหญ่มีค่าระดับสีเทาไม่สม่ำเสมอตลอดทั้งภาพได้ แต่ความแตกต่างของค่าระดับสีเทาในวัตถุใดๆ จะมีไม่มาก คือ มีการเปลี่ยนแปลงของระดับสีเทาอย่างช้าๆ ในวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใดวัตถุหนึ่ง การตัดสินใจว่าบริเวณใดเป็นขอบก็ขึ้นอยู่กับค่าความชันในการเปลี่ยนแปลงค่าระดับสีเทา ซึ่งจะต้องมีค่ามากกว่าค่าเทรชโฮล ซึ่งภาพที่ได้จะหาขอบได้ดีหรือไม่ก็ขึ้นกับการกำหนดค่าเทรชโฮลนั่นเอง

เนื่องจากวิธีเกรเดียนต์ใช้หลักในการพิจารณาการเปลี่ยนแปลงของค่าระดับสีเทาของภาพ ถ้ามีการเปลี่ยนแปลงมาก ก็จะถูกกำหนดให้เป็นขอบของวัตถุ ทั้งนี้ค่าเกรเดียนต์ขอบภาพ f(x,y) ณ ตำแหน่ง (x,y) จะเป็นค่าเวคเตอร์

$$\nabla f = \begin{pmatrix} G_x \\ G_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix} \qquad \text{.....2.12}$$

โดยที่ขนาดของค่าเวกเตอร์ ∇f เป็นปริมาณที่มีความสำคัญในการหาขอบของวัตถุ โดยทั่วๆ ไป จะเรียกว่าค่าเกรเดียนต์ และใช้สัญลักษณ์ ∇f ที่มีค่า

$$\nabla f = \text{mag}\left(\nabla f\right) = \sqrt{G_x^2 + G_y^2} \qquad \text{.....2.13}$$

ในทางปฏิบัติ เพื่อให้ง่ายต่อการคำนวณ จะทำการประมาณค่าเกรเดียนต์ด้วยค่าสัมบูรณ์

$$\nabla f = \left|G_x\right| + \left|G_y\right| \qquad \text{.....2.14}$$

การหาค่าอนุพันธ์ย่อย Gx และ Gy ทำได้โดยการใช้หน้ากาก (mask) หรือหน้าต่าง (window) หรือตัวดำเนินการ (operator) ขนาด 3x3 ที่มีค่าสัมประสิทธิ์ ดังแสดงในรูปที่ 2.18ก เข้าช่วยปรับค่าสีเทา ณ จุดภาพที่สอดคล้องกัน ซึ่งตัวดำเนินการมีหลายแบบดังนี้

1) ตัวดำเนินการแบบโรเบิร์ตครอสส์เกรเดียนต์ (Robert-cross gradient operator)

ตัวดำเนินการแบบโรเบิร์ตครอสส์เกรเดียนต์ อาจพิจารณาได้เป็นหน้ากากขนาด 2x2 ดังรูปที่ 2.18 ข.1 หรือเป็นหน้ากากขนาด 3x3 ดังรูปที่ 2.18 ข.2 ที่ให้ค่าอนุพันธ์ย่อยเป็น

$$G_x = Z_5 - Z_9 \qquad \text{.....2.15ก}$$

$$G_y = Z_6 - Z_8 \qquad \text{.....2.15ข}$$

2) ตัวดำเนินการแบบโซเบล (Sobel Operator)

ตัวดำเนินการแบบโซเบล มีหน้ากากขนาด 3x3 ดังรูปที่ 2.18ค ทำให้ได้ค่าอนุพันธ์ย่อยเป็น

$$G_x = \frac{1}{4}\left[(Z_7+2Z_8+Z_9)-(Z_1+2Z_2+Z_3)\right] \qquad \text{.....2.16ก}$$

$$G_y = \frac{1}{4}\left[(Z_3+2Z_6+Z_9)-(Z_1+2Z_4+Z_7)\right] \qquad \text{.....2.16ข}$$

3) ตัวดำเนินการแบบพรีวิตต์ (Prewitt Operator)

ตัวดำเนินการแบบพรีวิตต์ มีหน้ากากขนาด 3x3 เช่นเดียวกับตัวดำเนินการโซเบลแต่มีค่าสัมประสิทธิ์แตกต่างกัน ดังรูปที่ 2.18ง ทำให้ได้ค่าอนุพันธ์ย่อยเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G_x = \frac{1}{3}[(Z_7+Z_8+Z_9)-(Z_1+Z_2+Z_3)] \quad \text{.....2.17ก}$$

$$G_y = \frac{1}{3}[(Z_3+Z_6+Z_9)-(Z_1+Z_4+Z_7)] \quad \text{.....2.17ข}$$

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

ก) หน้ากากขนาด 3x3 ที่จะใช้เป็นตัวดำเนินการ

1	0
0	-1

0	1
-1	0

หน้ากากที่ใช้คำนวณค่า G_x และ G_y

0	0	-1
0	1	0
0	0	0

-1	0	0
0	1	0
0	0	0

หน้ากากที่ใช้คำนวณค่า G_x และ G_y

ข) ตัวดำเนินการแบบโรเบิร์ตครอสส์เกรเดียนต์

$\frac{1}{4}$

-1	-2	-1
0	0	0
1	2	1

หน้ากากที่ใช้คำนวณค่า G_x

$\frac{1}{4}$

1	0	-1
2	0	-2
1	0	-1

หน้ากากที่ใช้คำนวณค่า G_y

ค) ตัวดำเนินการแบบโซเบล

$\frac{1}{3}$

-1	-1	-1
0	0	0
1	1	1

หน้ากากที่ใช้คำนวณค่า G_x

$\frac{1}{3}$

-1	0	1
-1	0	1
-1	0	1

หน้ากากที่ใช้คำนวณค่า G_y

ง) ตัวดำเนินการแบบพรีวิตต์

รูปที่ 2.18 ตัวดำเนินการที่ใช้ในการคำนวณหาค่าเกรเดียนต์

ข้อจำกัดของการหาขอบภาพที่ได้กล่าวมาข้างต้นคือความไม่สามารถหาขอบได้ในสภาวะที่มีสัญญาณรบกวนมาก ปัญหานี้สามารถแก้ไขโดยการใช้ขนาดของอาร์เรย์ที่ใหญ่ขึ้น ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ตัวดำเนินการแบบพรีวิตต์ ขนาด 7x7

$\frac{1}{21}$

1	1	1	0	-1	-1	-1
1	1	1	0	-1	-1	-1
1	1	1	0	-1	-1	-1
1	1	1	0	-1	-1	-1
1	1	1	0	-1	-1	-1
1	1	1	0	-1	-1	-1
1	1	1	0	-1	-1	-1

รูปที่ 2.19 หน้ากากที่ใช้คำนวณค่า G_yของตัวดำเนินการแบบพรีวิตต์ขนาด 7x7

การหาขอบในลักษณะนี้ถูกเรียกว่าการหาขอบแบบ Boxcar และหน้ากากที่ใช้คำนวณค่า G_x จะได้จากการหมุนหน้ากากที่ใช้คำนวณค่า G_y ไป 90 องศา

5) ตัวดำเนินการแบบ Truncated Pyramid

Abdout ได้เสนอการหาขอบชนิด Truncated Pyramid ที่มีการลดแฟคเตอร์ของพิกเซลที่อยู่ไกลออกไป

$\frac{1}{21}$

1	1	1	0	-1	-1	-1
1	2	2	0	-2	-2	-1
1	2	3	0	-3	-2	-1
1	2	3	0	-3	-2	-1
1	2	3	0	-3	-2	-1
1	2	2	0	-2	-2	-1
1	1	1	0	-1	-1	-1

รูปที่ 2.20 หน้ากากที่ใช้คำนวณค่า G_yของตัวดำเนินการแบบ Truncated Pyramid

หน้ากากที่ใช้คำนวณค่า G_x จะได้จากการหมุนหน้ากากที่ใช้คำนวณค่า G_y ไป 90 องศา

2. วิธีลาปลาเซียน (Laplacian Method)

วิธีลาปลาเซียน ใช้หลักการพิจารณาการเปลี่ยนแปลงค่าเกรเดียนต์ ของค่าระดับสีเทาอีกต่อหนึ่ง และจะกำหนดตำแหน่งที่มี Zero-Crossing เกิดขึ้นเป็นตำแหน่งขอบของวัตถุ ทั้งนี้ การหาอนุพันธ์ย่อยอันดับสองของภาพ f(x,y) คือการหาค่าลาปลาเซียนแบบ 2 มิติ เป็นต้น

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad \text{.....2.18ก}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้ากากของวิธีลาปลาเซียนขนาด 3x3 จะมีค่าสัมประสิทธิ์แตกต่างกันหลายแบบดังรูปที่ 2.21 และตัวอย่างการใช้ตัวดำเนินการแบบ 4-neighbor เพื่อหาค่าอนุพันธ์ย่อยอันดับสองของจุดภาพ f(x,y) คือ

$$\nabla^2 f = 4Z_5-(Z_2+Z_4+Z_6+Z_8) \qquad \text{.....2.18ข}$$

0	-1	0
-1	4	-1
0	-1	0

ก) แบบ 4-neighborhood

-1	-1	-1
-1	8	-1
-1	-1	-1

ข) แบบ 8-neighborhood

-2	1	-2
1	4	1
-2	1	-2

ค) แบบ seperable 8-neighborhood

รูปที่ 2.21 หน้ากากที่ใช้ในตัวดำเนินการลาปลาเซียน

3. การหาขอบแบบแคนนี (Canny edge detection)

การหาขอบแบบแคนนี เป็นการดัดแปลงวิธีแกรเดียนต์ให้เหมาะสมกับการตรวจหาขอบขององค์ประกอบในภาพที่มีสัญญาณรบกวน แต่อยู่บนเงื่อนไข 3 ประการ ดังนี้

ก) ไม่เกิดความผิดพลาดในการหาขอบที่สำคัญ และไม่ควรมีการหาขอบที่ผิดเกิดขึ้น

ข) การบอกตำแหน่งของขอบ จะต้องมีระยะระหว่างตำแหน่งจริงและตำแหน่งที่ห่างขอบสั้นที่สุด

ค) ถ้ามีค่าหลายค่าในการหาขอบของขอบใดขอบหนึ่ง ค่าที่น้อยที่สุดจะเป็นค่าของขอบที่แท้จริง เพื่อแก้ปัญหาสัญญาณรบกวน ซึ่งจะทำให้ได้ขอบที่ไม่เรียบ

สำหรับขั้นตอนวิธีการหาขอบแบบแคนนี มี 4 ขั้นตอน คือ

ขั้นที่1 ใช้ตัวกรองไบโนเมียล เพื่อลดสัญญาณรบกวน ดังสมการที่ 2.14

$$f(x,y) = \left[\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}\right]^n \qquad \text{.....2.19}$$

โดยที่ n = 2 เท่าของความแปรปรวนของการกระจายแบบเกาส์ (Gaussian distribution)

ขั้นที่ 2 ทำการคำนวณเกรเดียนต์ตามสมการที่ 2.10 และใช้ตัวดำเนินการแบบพรีวิตต์ ดังรูปที่ 2.3ง จะได้

$$\nabla f = \left|\left(Z_7+Z_8+Z_9\right)-\left(Z_1+Z_2+Z_3\right)\right| + \left|\left(Z_3+Z_6+Z_9\right)-\left(Z_1+Z_4+Z_7\right)\right|$$

ขั้นที่ 3 ทำการหาว่าจุดภาพมีค่า เกรเดียนต์เฉพาะถิ่นสูงสุด (Maximum local gradient) หรือไม่ สมมติค่าเกรเดียนต์ของแต่ละจุดภาพในกรอบขนาด 3x3 มีค่าเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P1 P2 P3

P4 P5 P6

P7 P8 P9

ทำการพิจารณาเปรียบเทียบค่าเกรเดียนต์ของจุดภาพกลางกับจุดภาพอื่นๆ ในแนวเส้นตรงต่างๆ ประกอบด้วย แนวตั้ง แนวนอน แนวแทยง +45 องศา และ –45 องศา

ถ้า P5 > Pk ทุกค่า k≠5 และ P5 > ค่าเทรชโฮลต่ำ (low threshold value) T_L แสดงว่าค่า P5 เป็นค่าเกรเดียนต์เฉพาะถิ่นสูงสุด และจะนำค่าดังกล่าวไปคำนวณตามขั้นตอนที่ 4 ต่อไป

ขั้นที่ 4 การทำเทรชโฮลแบบฮีสเทอรีซิส

ขั้นตอนสุดท้ายนี้เป็นขั้นตอนการตัดสินใจว่าจุดภาพหนึ่งๆ เป็นขอบขององค์ประกอบหรือไม่ มีเงื่อนไข 3 ประการ ดังนี้

- จุดภาพที่มีขนาดของเกรเดียนต์ใหญ่กว่าค่าเทรชโฮลสูง (High threshold value) T_H จุดภาพนั้นจะเป็นขอบขององค์ประกอบ
- ทุกจุดภาพที่จะประกอบกันเป็นแนวขอบแนวหนึ่ง จะต้องมีอย่างน้อย 1 จุดภาพในแนวนั้นที่มีค่าเกรเดียนต์ > T_H
- จุดภาพที่มีค่าเกรเดียนต์ < T_H และไม่สอดคล้องกับเงื่อนไขที่สอง จะถูกตัดออกไป

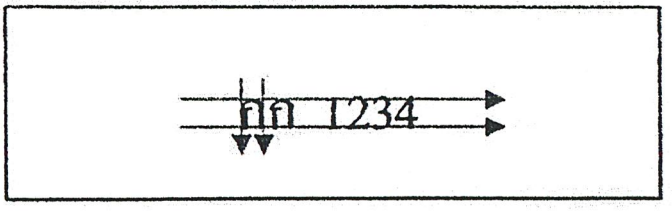
2.4.6 ทฤษฎีการแยกตัวอักขระออกจากภาพ

2.4.6.1 การหากรอบตัวอักขระโดยวิธี Line Crossing

เทคนิคนี้จะมีหลักการง่ายๆ คือ พยายามแบ่งอักขระแต่ละตัวออกมาให้ได้มากที่สุด โดยเริ่มจากกาสแกนเป็นแนวตรง ทั้งแนวนอนและแนวตั้ง ตัดไปตามช่องว่างทำให้เกิดเป็นตารางครอบตัวอักขระ ซึ่งตารางที่เล็กที่สุดที่ครอบตัวอักขระอยู่ถือเป็นการ segment ได้หนึ่งอักขระดังรูปที่ 2.22

แนวคิดของวิธีการ Line Crossing ดังรูปที่ 2.22 มีขั้นตอนดังต่อไปนี้

1. สแกนเป็นเส้นตรงตามแนวนอน
2. สแกนเป็นเส้นตรงตามแนวตั้ง
   (ขั้นตอนที่ 1 กับ 2 สามารถที่จะทำขั้นตอนใดก่อนก็ได้)
3. คัดลอกข้อมูลที่อยู่ในกรอบที่มีอักขระเก็บไว้ในบัฟเฟอร์ (buffer) เพื่อนำไปแสดงผล



รูปที่ 2.22 การแยกตัวอักขระด้วยวิธี Line Crossing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

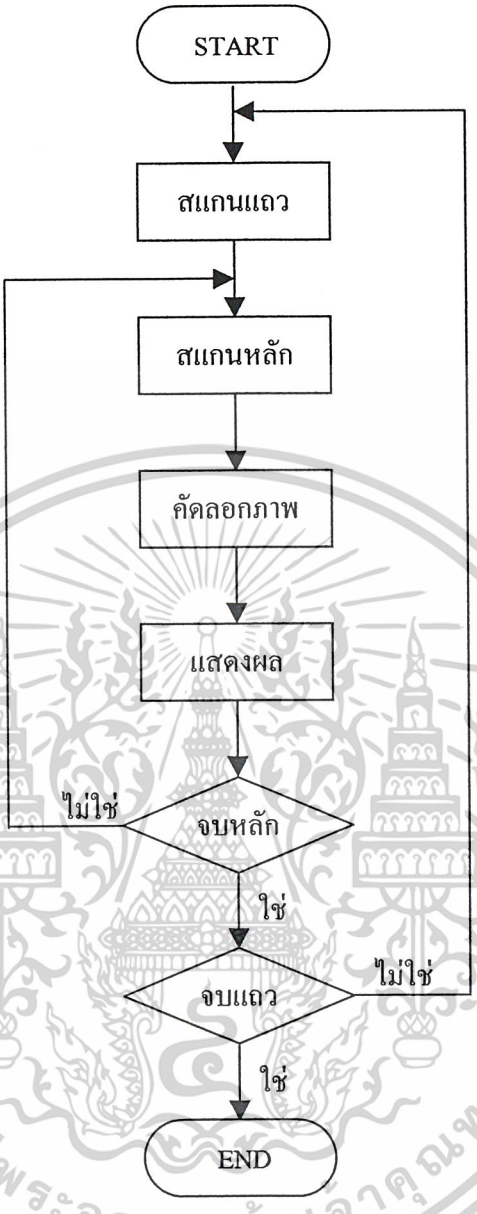
อธิบายโฟลว์ชาร์ตการแยกตัวอักขระด้วยวิธีการ Line Crossing จากรูปที่ 2.23

1. สแกนแถว : หาช่องว่างในแนวนอน
2. สแกนหลัก : หาช่องว่างในแนวตั้ง
3. คัดลอกภาพ : คัดลอกข้อมูลของส่วนที่ตัดได้
4. แสดงผล : แสดงรูปที่ทำการตัดได้
5. จบหลัก : การสแกนข้อมูลในแนวตั้งครบหมด
   - ใช่ : จบหลัก
   - ไม่ใช่ : สแกนหลัก
6. จบแถว : การสแกนข้อมูลในแนวนอนครบหมด
   - ใช่ : จบแถว
   - ไม่ใช่ : สแกนแถว

ข้อดีของการทำ Line Crossing ก็คือ ง่ายต่อการเขียนโปรแกรม และมีประสิทธิภาพในการแยกอักขระกับแบบอักขระที่มีข้อผิดพลาดน้อย ตัวอักขระไม่ติดกันมาก และถ้าเป็นการแยกตัวอักขระภาษาอังกฤษจะได้ผลลัพธ์ที่ดีกว่าตัวอักขระภาษาไทย

ข้อเสียของวิธีการนี้ที่เห็นได้ชัดก็คือ ในการทำการแยกตัวอักขระที่มีการเหลื่อมล้ำกันจะไม่สามารถใช้เทคนิคนี้ในการแยกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

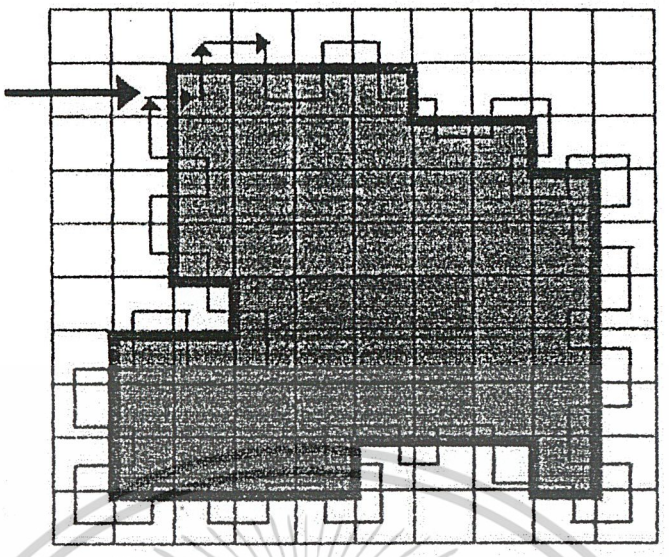


รูปที่ 2.23 โฟลว์ชาร์ตการแยกตัวอักขระด้วยวิธี Line Crossing

2.4.6.2 เทคนิคการตามรอยขอบภาพ (Contour Following)

เทคนิคการตามรอยขอบภาพจะช่วยให้เราทำการแยกตัวอักษรได้ง่ายขึ้น โดยไม่ต้องคำนึงถึงรูปร่างของวัตถุที่จะทำการแยก เมื่อพิจารณาขอบของวัตถุจะมีค่าสีที่มีความแตกต่างกันกับสีของพื้นหลัง เช่น ภาพตัวอักขระสีดำที่วางอยู่บนพื้นกระดาษสีขาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 การแยกตัวอักขระด้วยวิธีการตามรอยขอบภาพ

การทำงานของเทคนิคนี้ในการทำการตามรอยขอบภาพ ตามรูปที่ 2.24 ทำได้ตามขั้นตอนดังนี้

1. จุดภาพปัจจุบันมีข้อมูลเป็นดำ ให้เลี้ยวซ้ายจากทิศทางปัจจุบัน
2. จุดภาพปัจจุบันมีข้อมูลเป็นสีขาว ให้เลี้ยวขวาจากทิศทางปัจจุบัน
3. สิ้นสุดเมื่อจุดภาพปัจจุบันอยู่ตำแหน่งเดียวกับจุดเริ่มต้นพอดี

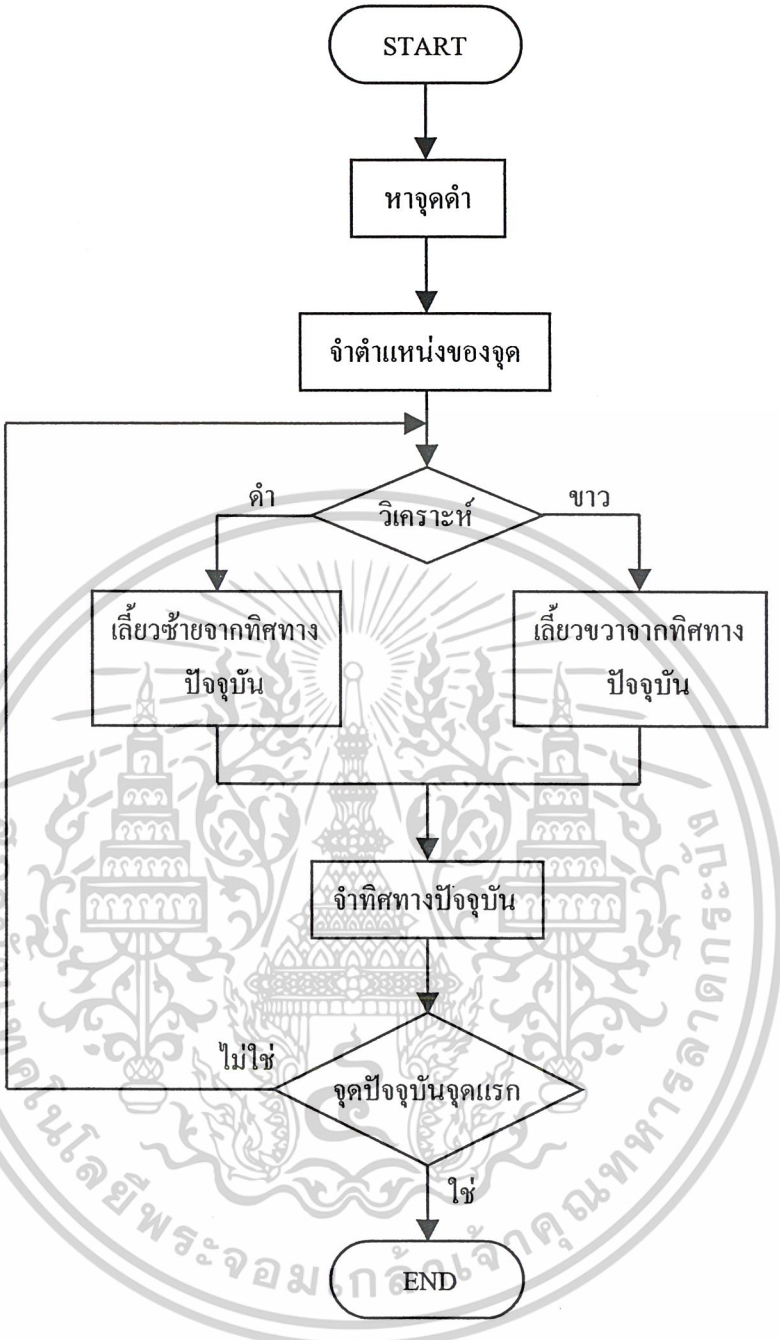
อธิบายโฟลว์ชาร์ตการแยกอักขระด้วยวิธีการการตามรอยขอบภาพ ตามรูปที่ 2.25

1. หาจุดดำ : เป็นการหาจุดดำจุดแรกที่พบ โดยการสแกนตามแนวนอน
2. จำตำแหน่งของจุด : เป็นการจำตำแหน่งของจุดดำที่พบ
3. วิเคราะห์จุด : มี 2 กรณี
   - ขาว : เลี้ยวขาวจากทิศทางปัจจุบัน
   - ดำ : เลี้ยวซ้ายจากทิศทางปัจจุบัน
4. จำทิศทางปัจจุบัน : จดจำทิศทางและตำแหน่งปัจจุบัน
5. จุดปัจจุบัน = จุดแรก : เป็นการเปรียบเทียบตำแหน่งปัจจุบันกับจุดเริ่มต้น
   - ใช่ : End
   - ไม่ใช่ : วิเคราะห์จุด

ข้อดีของวิธีการตามรอยขอบภาพก็คือ เราไม่ต้องสนใจว่ารูปร่างของตัวอักขระจะมีลักษณะเป็นอย่างไร เมื่อเราวิ่งไปตามขอบของภาพ เราก็จะได้ขอบของตัวอักขระออกมา

ข้อเสียของการทำวิธีการนี้คือ ถ้าอักขระมีส่วนที่ติดกันอยู่จะไม่สามารถใช้วิธีนี้ในการแยกตัวอักขระออกจากกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 โฟลว์ชาร์ตการแยกตัวอักขระด้วยวิธีการรอยขอบภาพ

2.4.6.3 เทคนิคการแยกโดยใช้วิธีการ Contour with matrix

เทคนิคนี้คล้ายกับการทำการตามรอยขอบภาพ แต่จะต่างกันตรงที่เทคนิคนี้จะมีการดึงเอาข้อมูลภาพที่อยู่ในขอบเขตของการทำการตามรอยขอบภาพไปใช้ด้วย ทำให้เราได้ตัวอักขระทั้งตัวไปใช้งานและการใช้งานไม่เพียงแต่ดูว่าข้อมูลตรงจุดนั้นเป็น 0 หรือ 1 แต่จะใช้เมตริกซ์ช่วยตรวจสอบแทน เมตริกซ์ที่ใช้มีขนาด 3x3 ดังรูปที่ 2.26

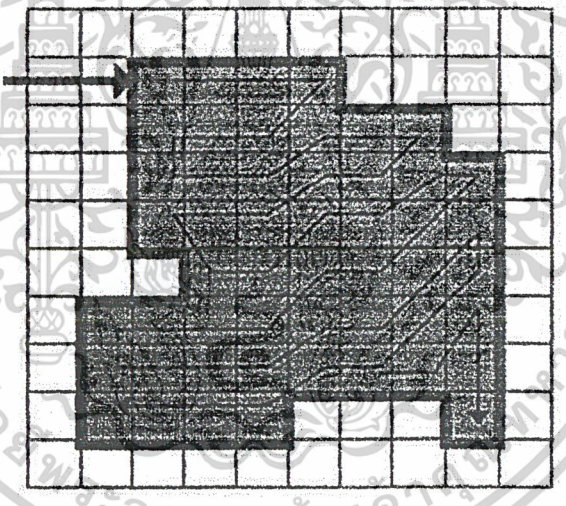
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P1	P2	P3
P4	P5	P6
P7	P8	P9

รูปที่ 2.26 เมตริกซ์ขนาด 3x3

แนวคิดของการทำ Contour with matrix ดังรูปที่ 2.27 มีขั้นตอนการทำงานดังต่อไปนี้

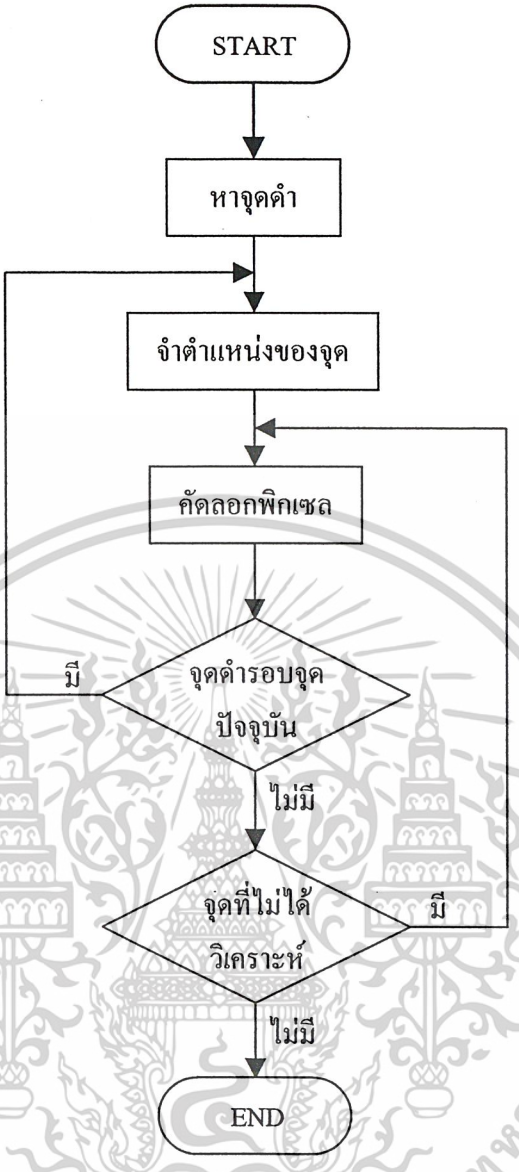
1. ให้กำหนดจุดดำที่พบเป็นจุดกลางของเมตริกซ์ (P5) คัดลอกจุดนี้ลงในบัฟเฟอร์แล้วลบจุดกลางนั้นออกจากรูปภาพ
2. หาจุดกลางถัดไป โดยตรวจสอบข้อมูลรอบๆ จุดกลางปัจจุบัน (P5) ซึ่งเป็นจุดสีดำ คือ ข้อมูลตัวอักขระ การตรวจสอบจะเริ่มจากด้านบนซ้ายก่อน คือ ตำแหน่ง P1,P2,P3,P4,P6,P7,P8 และ P9 ตามลำดับ กำหนดจุดดำที่พบจุดแรกให้เป็นจุดกึ่งกลางเมตริกซ์
3. เมื่อรอบๆ จุดกลางนั้นไม่มีจุดดำแล้ว จะทำการย้อนกลับไปยังจุดกลางตัวก่อน
4. การทำงานจะเป็นเช่นนี้ไปเรื่อยๆ จนกระทั่งไม่มีจุดดำรอบจุดกลางทุกตัว



รูปที่ 2.27 การหาขอบโดยใช้วิธี Contour with Matrix

ตัวอย่างการทำงานด้วยวิธีนี้แสดงดังรูปที่ 2.27 เส้นทางการวิ่งจะเป็นไปตามลูกศรที่แสดงในรูป ซึ่งรูปตัวอย่างเป็นรูปเดียวกับตัวอย่างที่แสดงในวิธีการตามรอยขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 โฟลว์ชาร์ตการหาขอบโดยวิธี Contour with matrix

อธิบายโฟลว์ชาร์ตการแยกอักขระด้วยวิธีการ Contour with matrix จากรูปที่ 2.28

1. หาจุดดำ : เป็นการหาจุดดำจุดแรกที่พบโดยการสแกนตามแนวนอน
2. จำตำแหน่งของจุด : จำตำแหน่งจุดดำที่พบ
3. คัดลอกพิกเซล : จะทำการคัดลอกจุดที่พบไว้ในบัฟเฟอร์และทำการลบจุดที่พบออกจากรูปภาพ
4. จุดดำรอบจุดปัจจุบัน : ตรวจว่ารอบจุดปัจจุบันมีจุดสีดำล้อมรอบหรือไม่
   - มี : จำตำแหน่งของจุด
   - ไม่มี : จุดที่ไม่ได้วิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จุดที่ไม่ได้วิเคราะห์ : ตรวจว่ามีจุดที่ยังไม่ได้ทำการวิเคราะห์หรือไม่

- มี : คัดลอกพิกเซล

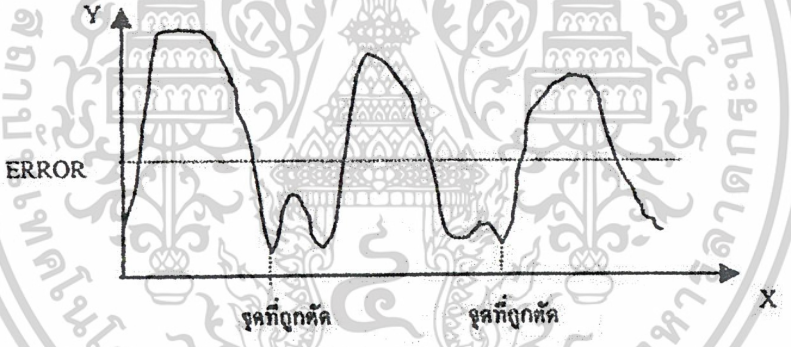
- ไม่มี : End

ข้อดีของวิธี Contour with matrix ก็คือ สามารถดึงเอาส่วนที่เป็นเนื้อตัวอักขระออกมาได้ทั้งหมด เป็นประโยชน์ต่อการนำไปใช้ในกระบวนการอื่นๆ ที่เนื้อของข้อมูลเป็นสิ่งจำเป็น

ข้อเสียของวิธีนี้คือ เมื่อเราต้องดึงเอาข้อมูลทุกส่วนของตัวอักษรออกมาก็จะทำให้เราต้องใช้เวลานานกับการคัดลอก และลบข้อมูล

2.4.6.4 เทคนิคการแยกตัวอักขระที่ติดกันโดยใช้ฮิสโตแกรม (Histogram)

ฮิสโตแกรมใช้เพื่อวัดความหนาแน่นของข้อมูลภาพในช่วงที่กำหนด โดยค่าความหนาแน่นที่ใช้เป็นจำนวนจุดดำ เพื่อช่วยในการวิเคราะห์ภาพตัวอักขระ โดยเก็บเป็นค่าเปอร์เซ็นต์ผิดพลาด เพื่อนำไปใช้เปรียบเทียบกับตัวอักขระแต่ละตัว เพื่อหาตัวอักขระที่คาดว่ามีการติดกันอยู่ และการใช้ฮิสโตแกรมจะช่วยให้โปรแกรมสามารถตัดสินใจได้ว่า ควรที่จะตัดส่วนที่ติดกันของอักขระที่ตรงส่วนใด เพื่อให้สามารถแยกตัวอักขระที่ติดกัน ให้ออกมาเป็นตัวอักขระเดี่ยวให้ได้มากที่สุด รูปที่ 2.29 แสดงการเลือกจุดที่จะทำการแยกตัวอักขระออกจากกัน



รูปที่ 2.29 การกำหนดจุดตัดด้วยฮิสโตแกรม

การแยกส่วนที่ติดกันของตัวอักขระ โดยใช้ฮิสโตแกรมมีวิธีการดังนี้

1. ทำการหาค่าฮิสโตแกรมของตัวอักขระที่สามารถแยกออกมาจากรูปภาพได้
2. กำหนดค่าเปอร์เซ็นต์ผิดพลาดเพื่อใช้ในการแยกตัวอักขระออกจากกัน
3. ทำการแยกตัวอักขระตรงส่วนที่มีค่าความหนาแน่นเท่ากับ หรือต่ำกว่าค่าเปอร์เซ็นต์ผิดพลาดที่ได้กำหนดไว้ โดยจะเลือกแยกตัวอักขระตรงส่วนที่มีค่าความหนาแน่นต่ำที่สุด
4. ถ้าค่าความหนาแน่นที่ต่ำที่สุดมีหลายตำแหน่ง จะใช้ตำแหน่งที่อยู่ตรงกลาง

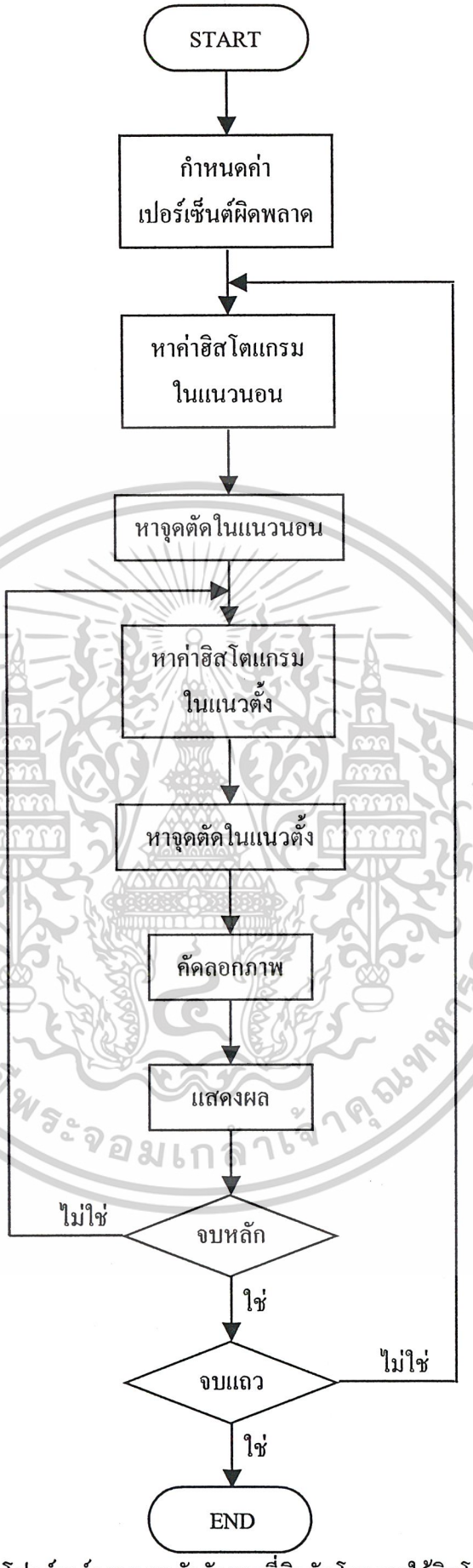
อธิบายโฟลว์ชาร์ตการแยกตัวอักขระที่ติดกันด้วยการใช้ฮิสโตแกรมจากรูปที่ 2.30

1. กำหนดค่าเปอร์เซ็นผิดพลาด : เป็นการกำหนดค่าเปอร์เซ็นต์ผิดพลาดเพื่อใช้ในการแยกอักขระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หาค่าฮิสโตแกรมในแนวนอน : หาค่าของฮิสโตแกรมโดยหาค่าความหนาแน่นในแนวนอน
3. หาจุดตัดในแนวนอน : เลือกตำแหน่งในการแยกตัวอักขระออกจากกันในแนวนอน
4. หาค่าฮิสโตแกรมในแนวตั้ง : หาค่าของฮิสโตแกรมโดยหาค่าความหนาแน่นในแนวตั้ง
5. หาจุดตัดในแนวตั้ง : เลือกตำแหน่งในการแยกตัวอักขระออกจากกันในแนวตั้ง
6. คัดลอกภาพ : คัดลอกข้อมูลของส่วนที่ตัดได้
7. แสดงผล : แสดงรูปที่ทำการตัดได้
8. จบหลัก : การสแกนข้อมูลในแนวตั้งครบหมด
    - ใช่ : จบหลัก
    - ไม่ใช่ : หาค่าฮิสโตแกรมในแนวตั้ง
9. จบแถว : การสแกนข้อมูลในแนวนอนครบหมด
    - ใช่ : จบแถว
    - ไม่ใช่ : หาค่าฮิสโตแกรมในแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 โฟลว์ชาร์ตการแยกตัวอักขระที่ติดกันโดยการใช้ฮิสโตแกรม

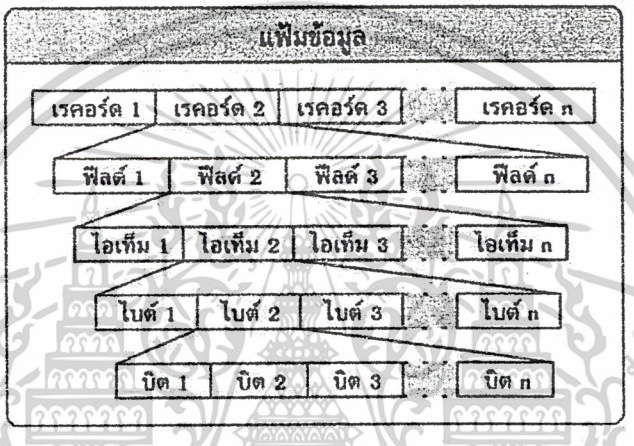
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ฐานข้อมูล

ฐานข้อมูลคือ กลุ่มข้อมูลที่เป็นข้อเท็จจริงที่ถูกนำมาเก็บรวบรวมไว้ในที่เดียวกันอย่างเป็นระบบ เพื่อนำไปใช้ในวัตถุประสงค์อย่างใดอย่างหนึ่งโดยกลุ่มผู้ใช่ตั้งแต่หนึ่งกลุ่มขึ้นไป ข้อมูลเหล่านี้อาจจะเป็นข้อเท็จจริงที่เกียวกับบุคคล สิ่งของ สถานที่ หรือเหตุการณ์ใดๆ ซึ่งเป็นได้ทั้งตัวเลข ข้อความ รูปภาพ หรืออื่นๆ

ลักษณะของฐานข้อมูลประกอบด้วย

- ข้อมูลทั้งหมดจะต้องถูกเก็บรวบรวมไว้ด้วยกัน
- จะต้องมีการจัดการข้อมูลนั้นอย่างเป็นระบบ
- ต้องสามารถนำข้อมูลนั้นไปใช้ได้ตามต้องการ



รูปที่ 2.31 โครงสร้างข้อมูลที่นำมาประมวลผลด้วยคอมพิวเตอร์

จากรูปที่ 2.31 สามารถอธิบายได้ดังนี้

- บิต (Bit) เป็นหน่วยข้อมูลที่เล็กที่สุด ที่แทนด้วยเลขฐานสอง (0 หรือ 1)

- ไบต์ (Byte) คือ กลุ่มของบิตที่แทนตัวอักษร ตัวเลข หรือสัญลักษณ์พิเศษหนึ่งตัว (character) เช่น รหัส ASCII 1 ไบต์ ซึ่งเก็บบิต 01000001 จะหมายถึงตัวอักษร A

-ไอเท็ม (Item) คือ ข้อมูลที่เกิดจากตัวอักษร ตัวเลข หรือสัญลักษณ์พิเศษมาเรียงต่อกันและมีความหมาย เช่น Smith แทนชื่อคน, Bangkok แทนชื่อจังหวัด

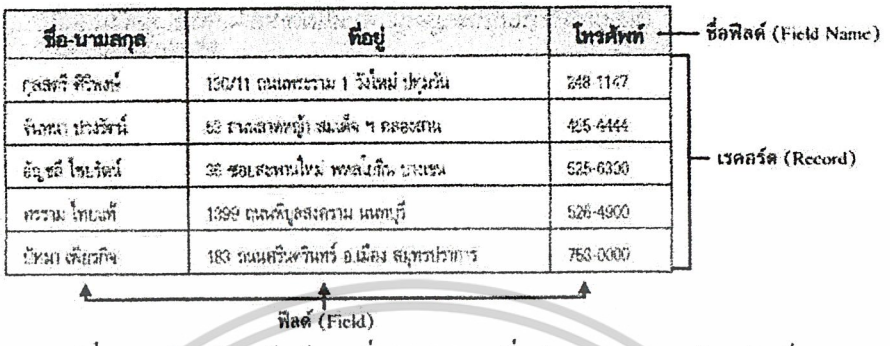
- คอลัมน์ (Column) คือ ข้อมูลที่ประกอบด้วยไอเท็มตั้งแต่ 1 ไอเท็มขึ้นไป เช่น ไอเท็มวัน เดือน และปี รวมกันเป็นคอลัมน์วันเกิด หรือไอเท็มชื่อและนามสกุลรวมกันเป็นคอลัมน์ชื่อ-นามสกุล เป็นต้น คอลัมน์ที่ประกอบด้วยไอเท็มตั้งแต่ 2 ไอเท็มจะเป็นคอลัมน์แบบ Group Item ถ้าประกอบด้วยหนึ่งไอเท็มจะเป็นคอลัมน์แบบ Elementary item ตัวอย่างเช่น คอลัมน์วันเกิดเป็น Group Item เนื่องจากสามารถแยกข้อมูลลงไปเป็นวัน เดือน และปีเกิดได้ ในขณะที่ วัน เดือน และปีเกิดเป็น Elementary Item เนื่องจากแยกย่อยลงไปอีกไม่ได้ ถ้าแยกย่อยลงไปจะไม่ได้ข้อมูลที่สื่อความหมาย

- แถว (Row) คือ กลุ่มของคอลัมน์ที่มีความสัมพันธ์กัน เช่น แถวพนักงานประกอบด้วยคอลัมน์รหัสพนักงาน ชื่อ นามสกุล แผนก ตำแหน่ง สถานภาพสมรส วันเข้าทำงาน ที่อยู่และอื่นๆ เป็นต้น หนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แถวจะเก็บข้อมูลพนักงาน 1 คน โดยคอลัมน์เดียวกันในทุกๆแถวจะต้องเก็บข้อมูลชนิดเดียวกัน เช่น คอลัมน์ชื่อและนามสกุล จะเก็บข้อมูลแบบตัวอักษรเท่านั้น

-ไฟล์ (File) หรือ แฟ้มข้อมูล คือกลุ่มแถวที่ข้อมูลเป็นเรื่องเดียวกัน เช่น แฟ้มข้อมูลพนักงาน แฟ้มข้อมูลลูกค้า แฟ้มข้อมูลการสั่งซื้อสินค้า แฟ้มข้อมูลสินค้า เป็นต้น



รูปที่ 2.32 ตัวอย่างแฟ้มข้อมูลชื่อ นามสกุล ที่อยู่และหมายเลขโทรศัพท์

2.6 โครงสร้าง แอลซีดี โมดูล (LCD Module)

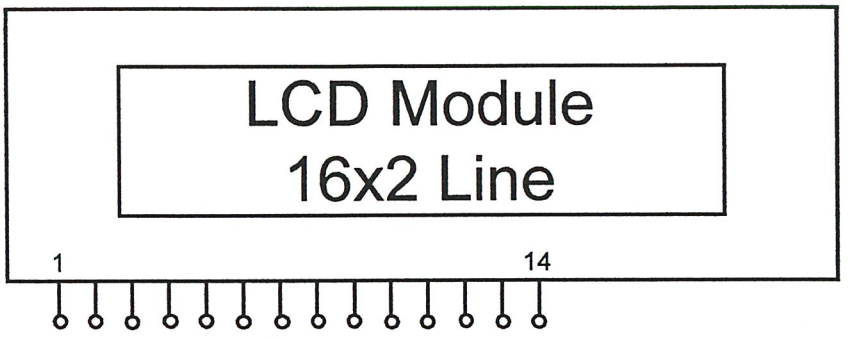
2.6.1 ส่วนประกอบหลักของ แอลซีดี โมดูล

ใน แอลซีดี โมดูล จะมีส่วนประกอบหลักๆ 3 ส่วน ดังนี้

1. ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมมองข้อมูลที่แสดงผลบนจอ
2. ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของ แอลซีดี โมดูลเช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะ ชิปที่นิยมใช้ คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม แอลซีดี แบบอักขระ ส่วน HD61830 ใช้ควบคุม แอลซีดี แบบกราฟฟิก
3. ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่นี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

แอลซีดี โมดูลมีอยู่หลายรุ่น และคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 แบบ คือ แบบ Dot matrix และ Graphic โดยแบบ Dot matrix จะแสดงผลเป็นแบบ 5 x 8 Dot. หรือ 5 ป 10 Dot. มีตั้งแต่ 1 Line, 2 Line และ 4 Line ซึ่งการใช้งานแต่ละแบบจะใกล้เคียงกัน ลักษณะขาสัญญาณของ แอลซีดี โมดูลแบบ 1 Line ดังรูปที่ 2.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.33 ลักษณะของตัว แอลซีดี โมดูล

ตัว แอลซีดี โมดูลจะมีขาใช้งานทั้งหมด 14 ขาด้วยกันหน้าที่ของแต่ละขามีดังนี้ คือ

ขา 1 (GND)	เป็น Ground ใช้ต่อกับระบบ Ground ของไมโครคอนโทรลเลอร์
ขา 2 (VCC)	เป็นไฟเลี้ยงวงจรของ แอลซีดี มีขนาด +5 VDC
ขา 3 (V_{ee})	เป็นขาสำหรับปรับความเข้มของจอ แอลซีดี โดยที่เมื่อต่อกับ VCC จะมีความเข้มต่ำสุด และเมื่อต่อกับ Ground จะมีความเข้มมากสุด โดยปกติจะต่ออยู่กับ Ground เสมอเพื่อความสะดวกในการต่อ
ขา 4 (RS)	Register Select ใช้สำหรับบอก แอลซีดี ทราบว่าข้อมูลที่ส่งให้นั้นเป็น Instruction หรือ Data โดยเมื่อขานี้ เป็น “0” หมายถึง Instruction เป็น “1” หมายถึง Data
ขา 5 (R/ $\overline{W}$)	ใช้สำหรับกำหนดว่าเป็นการอ่านหรือเขียนข้อมูลให้กับ แอลซีดี โดยเมื่อขานี้ เป็น “0” หมายถึงเป็นการเขียนข้อมูล เป็น “1” หมายถึงเป็นการอ่านข้อมูล
ขา 6 (E)	เป็นขา Enable ขานี้ เป็น “1” ใช้สำหรับบอก แอลซีดี ว่าอุปกรณ์ภายนอก ต้องการติดต่อด้วย เป็น “0” ตัว แอลซีดี จะไม่สนใจในสัญญาณ RS, R/ $\overline{W}$ และ (DB_7-DB_0)
ขา 7 – 14 (DB_7-DB_0)	เป็นขา Data Bus สำหรับอ่านหรือเขียนข้อมูลให้กับตัว แอลซีดี

ตารางที่ 2.6 หน้าที่ของขาใช้งานของ แอลซีดี โมดูล

2.6.2 การเชื่อมต่อ แอลซีดี โมดูล เข้ากับไมโครคอนโทรลเลอร์

การเชื่อมต่อ แอลซีดี โมดูล เข้ากับไมโครคอนโทรลเลอร์สามารถต่อได้โดยตรงกับตัว MCS-51 หรือต่อผ่าน 8255 ก็ได้ ในที่นี้จะต่อโดยตรงกับตัว MCS-51 ดังนี้

- ขาสัญญาณข้อมูล D0-D7 (ขา 7-14) ต่อเข้ากับ MCS-51 พอร์ต 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา RS (ขา 4) ต่อเข้ากับ MCS-51 พอร์ต 0 บิต 0
- ขา R/ $\overline{W}$ (ขา 5) ต่อเข้ากับ MCS-51 พอร์ต 0 บิต 1
- ขา E (ขา 6) ต่อเข้ากับ MCS-51 พอร์ต 0 บิต 2

2.6.3 ชุดคำสั่งของ แอลซีดี โมดูล

INSTRUCTION	RS	R/ $\overline{W}$	DATA BIT								EXECUTE TIME (nS)
			7	6	5	4	3	2	1	0	
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B	40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	*	*	40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	40
SET CGRAM ADDR.	0	0	0	1	CGRAM ADDRESS						40
SET DDRAM ADDR.	0	0	1	DDRAM ADDRESS							40
BUSY,ADDR. READ	0	1	BF	ADDRESS							0
CGRAM, DDRAM WR	1	0	WRITE DATA								40
CGRAM, DDRAM RD	1	1	READ DATA								40

ตารางที่ 2.7 ชุดคำสั่งและเวลาที่ แอลซีดี โมดูลใช้ในการทำงานแต่ละคำสั่ง

1. คำสั่งเคลียร์ตัวแสดงผล (CLEAR DISPLAY)

RS	R/ $\overline{W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

ตารางที่ 2.8 คำสั่งเคลียร์ตัวแสดงผล

คำสั่ง CLEAR DISPLAY เป็นคำสั่งที่ใช้เขียนข้อมูลหรือตัวอักษรว่าง (Space) ลงใน DDRAM ทั้งหมด และทำการกำหนดค่า DDRAM ADDRESS เป็น 0 และเคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งบนซ้ายสุดของจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คำสั่ง CURSOR AT HOME

RS	$\overline{R/W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

ตารางที่ 2.9 คำสั่ง CURSOR AT HOME

คำสั่ง CURSOR AT HOME หรือ RETURN HOME เป็นคำสั่งที่ใช้ในการเลื่อนตำแหน่งของเคอร์เซอร์ไปอยู่ที่ตำแหน่งบนซ้ายสุดของจอแสดงผล โดยข้อมูลที่อยู่ใน DDRAM หรือที่หน้าจอแสดงผลจะไม่เปลี่ยนแปลง

3.คำสั่งโหมดในการป้อนข้อมูล (ENTRY MODE SET)

RS	$\overline{R/W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

ตารางที่ 2.10 คำสั่งโหมดในการป้อนข้อมูล

คำสั่งโหมดในการป้อนข้อมูล (ENTRY MODE SET) ใช้สำหรับกำหนดการเลื่อนของเคอร์เซอร์และตำแหน่งแอดเดรสของ DDRAM ดังนี้

- I/D เป็นบิตที่ใช้ในการกำหนดการเลื่อนของเคอร์เซอร์และตำแหน่งแอดเดรสของ DDRAM ว่าจะให้เพิ่มหรือลดลงเมื่อเขียนหรืออ่านข้อมูลแล้ว

  บิต I/D = 0 แอดเดรสของ DDRAM จะลดลง

  บิต I/D = 1 แอดเดรสของ DDRAM จะเพิ่มขึ้นส่วนเคอร์เซอร์จะเลื่อนตามตำแหน่งแอดเดรสของ DDRAM

- S เป็นบิตที่ใช้กำหนดลักษณะของการแสดงผลเมื่อมีการเขียนข้อมูล

  บิต S = 1 เมื่อเขียนข้อมูลใหม่ลงไปแล้วตัวเคอร์เซอร์จะอยู่กับที่แต่ตัวอักษรข้อมูลเดิมจะถูกผลักไปทางซ้าย

  บิต S = 0 เมื่อเขียนข้อมูลใหม่ลงไปแล้วตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

4. คำสั่งควบคุมการแสดงผล (DISPLAY ON/OFF)

RS	$\overline{R/W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

ตารางที่ 2.11 คำสั่งควบคุมการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งควบคุมการแสดงผล เป็นคำสั่งที่ใช้ในการเปิดปิดจอแสดงผลและเคอร์เซอร์มีลักษณะดังนี้

D = 0 กำหนดให้ปิดจอแสดงผล (Display OFF)

D = 1 กำหนดให้เปิดจอแสดงผล (Display ON)

C = 0 กำหนดให้ปิดเคอร์เซอร์ (Cursor OFF)

C = 1 กำหนดให้เปิดเคอร์เซอร์ (Cursor ON)

B = 0 กำหนดให้ไม่มีการกระพริบที่เคอร์เซอร์

B = 1 กำหนดให้มีการกระพริบที่เคอร์เซอร์ (กระพริบเป็นรูปสี่เหลี่ยมทึบ)

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร (DISPLAY SHIFT)

RS	$\overline{R/W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

ตารางที่ 2.12 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร

คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร เป็นการควบคุมการเลื่อนของเคอร์เซอร์และตัวอักษรบนจอแสดงผล โดยขึ้นอยู่กับการกำหนดบิต S/C และ R/L โดยมีลักษณะดังนี้

S/C	R/L	ลักษณะการเลื่อน
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย
0	1	เลื่อนเคอร์เซอร์ไปทางขวา
1	0	เลื่อนตัวอักษรตัวใหม่ไปทางซ้าย
1	1	เลื่อนตัวอักษรตัวใหม่ไปทางขวา

ตารางที่ 2.13 การกำหนดบิต S/C และ R/L

6. คำสั่งการกำหนดฟังก์ชันการทำงาน (FUNCTION SET)

RS	$\overline{R/W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

ตารางที่ 2.14 คำสั่งการกำหนดฟังก์ชันการทำงาน

DL = 0 กำหนดให้ติดต่อกับ แอลซีดี โมดูลเป็นแบบ 4 บิต

DL = 1 กำหนดให้ติดต่อกับ แอลซีดี โมดูลเป็นแบบ 8 บิต

N = 0 กำหนดการแสดงผลแบบ 1 บรรทัด

N = 1 กำหนดการแสดงผลตั้งแต่ 2 บรรทัดขึ้นไป

F = 0 กำหนดความละเอียดของการแสดงผลเป็น 5 x 8 Dot.

F = 1 กำหนดความละเอียดของการแสดงผลเป็น 5 x 10 Dot.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ

1. การกำหนดบิต F สำหรับ แอลซีดี โมดูลแบบ 5 x 7 Dot.
2. เนื่องจากการกำหนดค่า DL สามารถกระทำได้ที่บิต (DB4-DB7) ถ้ามีการกำหนดให้เป็นแบบ 4 บิตตั้งแต่ครั้งแรก หลังจากจ่ายไฟเลี้ยงให้กับตัว แอลซีดี โมดูลแล้วก็จะทำให้เป็นการติดต่อกับแอลซีดี โมดูลเป็นแบบ 4 บิตทันที

7. คำสั่งเลือกแอดเดรสของ CGRAM (SET CGRAM ADDRESS)

RS	$\overline{R/W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

ตารางที่ 2.15 คำสั่งเลือกแอดเดรสของ CGRAM

คำสั่งนี้ใช้สำหรับกำหนดตำแหน่ง Address ของ Character Generator หรือ CGRAM โดยจะต้องกำหนดค่านี้ทุกครั้งในการเขียนหรืออ่านข้อมูลกับ CGRAM ซึ่งกำหนดที่ (DB0-DB5) ส่วน DB6 ต้องเป็น “1” และ DB7 ต้องเป็น “0” (01XX XXXX B) ซึ่งก็คือ (40H-7FH)

8. คำสั่งเลือกแอดเดรสของ DDRAM (SET DDRAM ADDRESS)

RS	$\overline{R/W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

ตารางที่ 2.16 คำสั่งเลือกแอดเดรสของ DDRAM

แอดเดรสของ แอลซีดี 16 อักษร 4 บรรทัด

บรรทัดที่ 1 1000 XXXXB = (80-8F)H

บรรทัดที่ 3 1001 XXXXB = (90-9F)H

บรรทัดที่ 2 1100 XXXXB = (C0-CF)H

บรรทัดที่ 4 1101 XXXXB = (D0-DF)H

คำสั่งนี้ใช้สำหรับกำหนดตำแหน่งแอดเดรสของ Display Data RAM หรือ DDRAM หรือตำแหน่งของเคอร์เซอร์สำหรับการแสดงผลทางหน้าจอ แอลซีดี ซึ่งเมื่อมีการอ่านหรือเขียนค่าตัวอักษรให้กับ แอลซีดี ในแต่ละครั้งนั้น ค่าตำแหน่งของ DDRAM Address จะเพิ่มขึ้นหรือลดลง 1 ตำแหน่ง โดยอัตโนมัติเสมอ ซึ่งจะเพิ่มหรือลดนั้นกำหนดได้จากบิต I/D ใน ENTRY MODE SET แต่เราก็สามารถที่จะกำหนดตำแหน่งแอดเดรสของ DDRAM ณ ตำแหน่งใดๆ ก็ได้บนจอ แอลซีดี ที่เราต้องการให้แสดงผล ณ จุดนั้นๆ ได้เองโดยกำหนดแอดเดรสนี้ก่อนที่จะทำการอ่านหรือเขียนตัวอักษรให้กับ DDRAM ADDRESS เสมอ สำหรับแอดเดรสของ DDRAM ของ แอลซีดี แต่ละแบบนั้น แสดงดังตารางข้างล่างนี้ซึ่งจะกำหนดให้บิต DB7 เท่ากับ “1” เสมอ เพื่อความสะดวกในการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ 16 ตัวอักษร 2 บรรทัด

80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

ตารางที่ 2.17 ตำแหน่งแอดเดรสของ DDRAM

9. คำสั่งอ่านแฟลก BUSY และแอดเดรส(READ BUSY FLAG & ADDRESS)

RS	$\overline{R/W}$	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	CGRAM/DDRAM ADDRESS						

ตารางที่ 2.18 คำสั่งอ่านแฟลก BUSY และแอดเดรส

คำสั่งนี้ใช้สำหรับการอ่านค่าของ Busy Flag (BF) ซึ่งบอกถึงความพร้อมของ แอลซีดี ในการรับข้อมูล

ถ้า BF = 0 หมายถึง แอลซีดี พร้อมที่จะรับข้อมูลต่อไปได้

ถ้า BF = 1 หมายถึงว่ายังไม่พร้อมที่จะรับข้อมูล

นอกจากนี้แล้วทุกครั้งที่อ่านค่าแฟลก BF เข้ามาแล้วก็จะได้ตำแหน่งของ CGRAM หรือ DDRAM Address ณ ในตำแหน่งของ (DB0-DB6)

สำหรับตัวอักษรที่ใช้ในการแสดงผลของ แอลซีดี โมดูลนั้นเป็นรหัส ASCII ที่ใช้กันทั่วๆไป ดังแสดงในรูปที่ 2.10 ตัวอย่างเช่นต้องการให้แสดงเป็นตัวอักษร “A” จะต้องส่งค่า 41H ออกไปให้ที่ตัว แอลซีดี หรือ “a” ก็จะต้องส่งค่า 61H ออกไป ส่วนค่า 00H จนถึง 07H หรือ 08H จนถึง 0FH (บิต D_3 นั้นจะเป็น “0” หรือ “1” ก็ได้ 0000 XYYY) YYY คือค่าตำแหน่งได้ทั้งหมด 8 ตำแหน่ง เพื่อนำตัวอักษรที่โหลดเองใน CGRAM ออกไปแสดงผล

การติดต่อกับ DDRAM ของ แอลซีดี โมดูล

1. ในการเขียนและอ่านข้อมูลให้กับ แอลซีดี โมดูลจะมี 2 ลักษณะ คือ Instruction หรือ รหัสคำสั่งควบคุม และ Data หรือข้อมูลที่จะให้แสดงผล จะถูกกำหนดโดยขาสัญญาณ RS

   RS = 0 หมายถึง Instruction หรืออ่านค่าของแฟลกสถานะการทำงานของ แอลซีดี โมดูล

   RS = 1 หมายถึง การเขียนหรืออ่านข้อมูลตัวอักษรกับตัว แอลซีดี โมดูล

   การเขียนข้อมูลตัวอักษรให้กับ แอลซีดี จะต้องให้ $R/\overline{W}$ เป็น “0”

   การอ่านข้อมูลจากตัว แอลซีดี จะต้องให้ $R/\overline{W}$ เป็น “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อเขียนข้อมูลให้กับ แอลซีดี โมดูลลงไปแล้ว ตัว แอลซีดี จะต้องใช้เวลาในการทำงานภายในตัว แอลซีดี เองชั่วเวลาหนึ่ง (ดังตารางที่ 2.4) ซึ่งในการเขียนโปรแกรมสามารถตรวจสอบได้โดยเช็คสถานะได้จาก Busy Flag (BF) ถ้าเรียบร้อยแล้วจึงสามารถส่งตัวอักษรตัวต่อไปได้ หรืออีกวิธีหนึ่งคือ ใช้วิธีการหน่วงเวลาแทนก็ได้ ซึ่งปกติแล้วจะใช้วิธีการหน่วงเวลาแทน เนื่องจากง่ายในการออกแบบวงจรและเขียนโปรแกรม

การเขียนข้อมูลให้กับ แอลซีดี โมดูล สามารถทำได้ทั้ง 8 บิต และ 4 บิต โดยการใช้งานแบบ 4 บิตจะใช้ขาสัญญาณเพียง 4 เส้นคือ (DB_7-DB_4) ซึ่งการเขียนหรืออ่านข้อมูลนั้นกระทำเช่นเดียวกับแบบ 8 บิตเพียงแต่ต้องทำ 2 ครั้ง คือ (DB_7-DB_4) ก่อนตามด้วย (DB_0-DB_3) ซึ่งจะต้องกำหนดคุณสมบัติที่บิต DL ใน Function Set ก่อนแล้ว

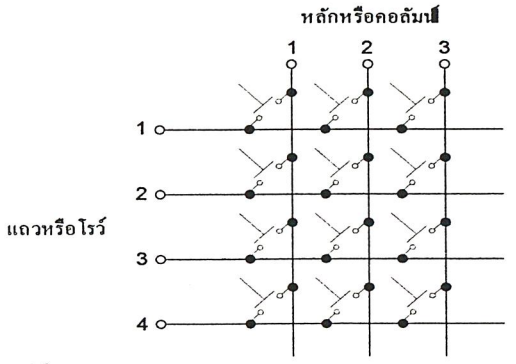
2.7 การใช้งานคีย์แพด (Keypad)

วิธีการอ่านค่าหรือรับค่าจากสวิตซ์นั้นมีด้วยกัน 2 ลักษณะใหญ่ๆคือ แบบต่อเข้ากับไฟเลี้ยงหรือกราวด์โดยตรง และต่อแบบวงจรเมทริกซ์ (Matrix Switch) ในรูปที่ 2.9 จะเห็นว่าสวิตซ์จะถูกต่อในแนวแกนตั้งและแนวแกนนอน จะเรียกแนวแกนตั้งว่าหลักหรือคอลัมน์ (Column) และเรียกแนวนอนว่าแถวหรือโรว์ (Row) ดังนั้นค่าของสวิตซ์ในแต่ละตำแหน่งจะต้องประกอบด้วยหลักและแถว

กระบวนการในการอ่านค่าของสวิตซ์แบบนี้นั้นจะใช้วิธีการเขียนโปรแกรมในไมโครคอนโทรลเลอร์ โดยจะต้องใช้สายทั้งหมด 7 เส้น ซึ่งเป็นสายของหลัก 3 เส้นและเป็นสายของแถว 4 เส้น สำหรับสวิตซ์แบบ 4x3 ต่อเข้ากับไมโครไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล “0” ไปยังพอร์ตที่ต่อกับด้านแถวทีละเส้นตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายด้านแถวของคีย์แพดไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ด้านหลักเข้ามาด้วย หากไม่มีการกดค่าของด้านหลักก็จะเป็น “1” ทั้งหมด ถ้าหากมีการกดคีย์ค่าของด้านหลักก็จะไม่เป็น “111” อีกต่อไป เป็นการแจ้งให้ทราบว่ามีการกดคีย์แพดเกิดขึ้นแล้ว จากนั้นไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งนั้นสิ่งที่จะได้มาอย่างแรกคือค่าตำแหน่งของคีย์นั้น จากนั้นก็จะนำค่าตำแหน่งนั้นไปเปิดตารางข้อมูล เพื่อที่จะได้ค่าที่ต้องการนำไปแสดงผลที่แท้จริง

ข้อดีของสวิตซ์แบบนี้ก็คือสามารถรองรับการเพิ่มของสวิตซ์ได้อย่างสะดวก เพียงแค่แก้ไขซอร์ฟแวร์เพียงเล็กน้อยเท่านั้น ทำให้วงจรสวิตซ์แบบเมทริกซ์นี้เป็นที่นิยมกันมากในระบบควบคุมอัตโนมัติหรือกึ่งอัตโนมัติ โดยในการใช้งานทั่วไปจะเรียกสวิตซ์แบบเมทริกซ์นี้ว่า “คีย์แพด”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

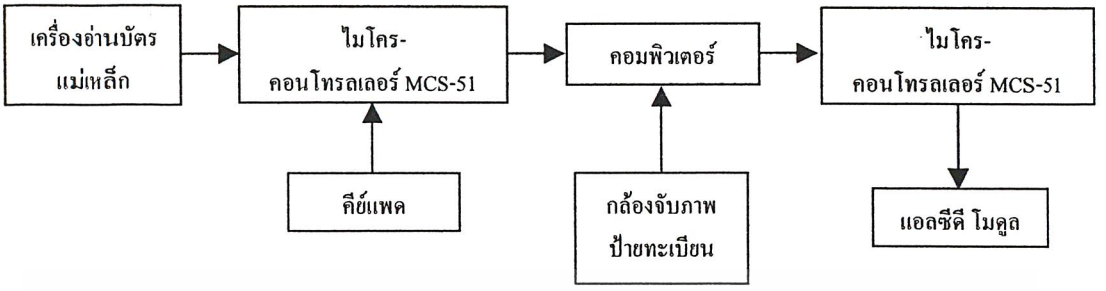


รูปที่ 2.34 วงจรสวิตซ์แบบเมทริกซ์ หรือ คีย์แพด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและการออกแบบ

พิจารณาบล็อกไดอะแกรม(Bloack diagram) ดังรูปที่ 3.1



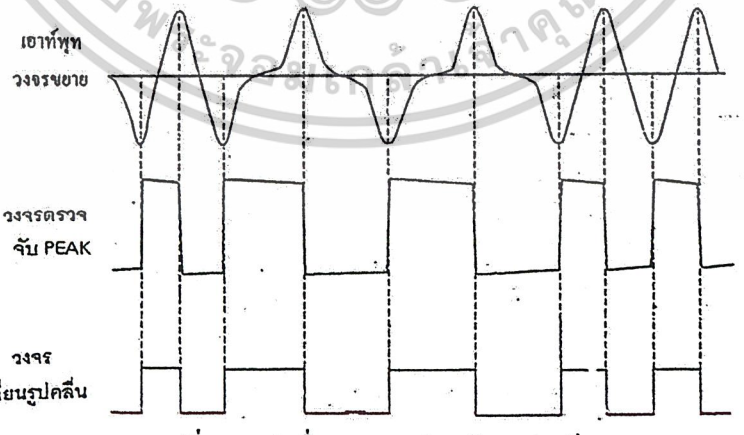
รูปที่ 3.1 บล็อคไดอะแกรมระบบรักษาความปลอดภัยสำหรับที่จอดรถยนต์

จากรูปที่ 3.1 สามารถแยกอธิบายการทำงานได้ดังนี้

3.1 ตัวเครื่องอ่านบัตรแม่เหล็ก

ในส่วนของเครื่องอ่านบัตรแม่เหล็กจะมีบล็อกไดอะแกรมของเครื่องอ่านบัตรแม่เหล็ก จะเป็นดังรูปที่ 3.2 จงจรหลักของเครื่องประกอบด้วยหัวอ่านซึ่งจะจับสัญญาณการอ่านที่วงจรตรวจจับ Peak to Peak ที่ตรวจพบ จะนำมาแปลงเป็นรูปพัลส์ด้วยวงจรเปลี่ยนรูปคลื่นเพื่อให้ได้สัญญาณพัลส์เหมือนที่บันทึกลง จะมีรูปดังรูปที่ 3.3

รูปที่ 3.2 บล็อกไดอะแกรมเครื่องอ่านบัตรแม่เหล็ก

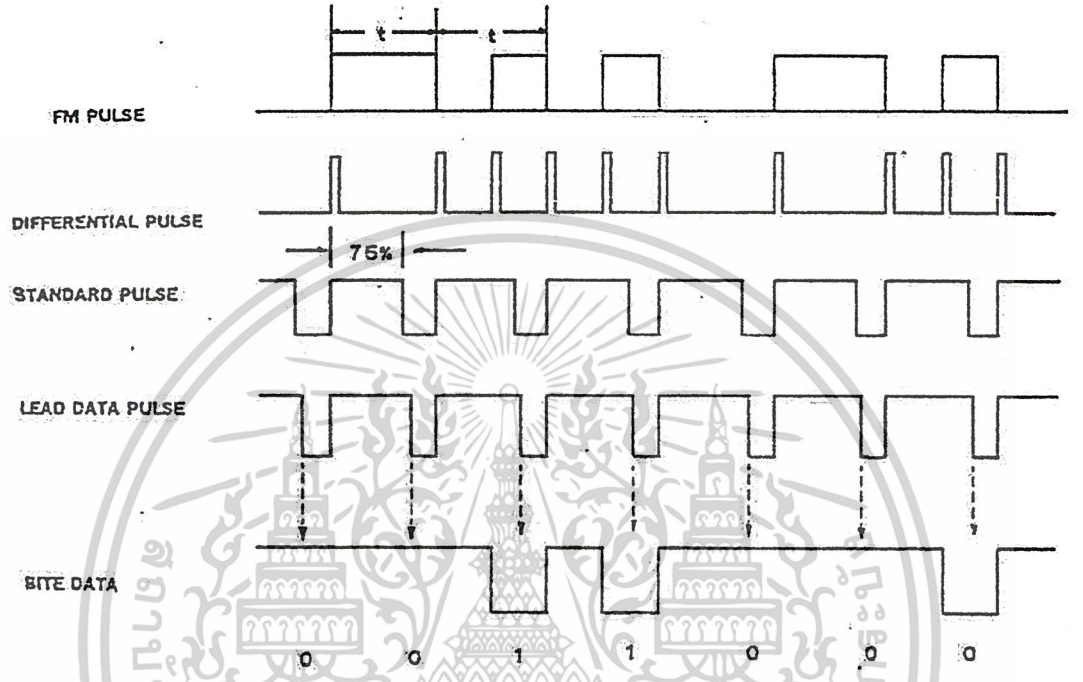


รูปที่ 3.3 รูปคลื่นของการอ่านบัตรแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถอดเฉพาะสัญญาณข้อมูลออกมาจากสัญญาณพัลส์ที่ได้นี้จะต้องตรวจดูว่ามีการกลับขั้วของพัลส์เอฟเอ็มในช่วงเวลามาตรฐาน (Standard Time) ซึ่งมีค่าประมาณ 75% ของความยาว 1 บิต (75% ของเวลา) หรือไม่ ถ้าไม่มี ค่าของข้อมูลจะเป็น “0” แต่ถ้ามีค่าของข้อมูลจะเป็น “1”

การแยกสัญญาณข้อมูลออกจากสัญญาณคล็อก จะใช้วงจรดีมอดูเลเตอร์ โดยมีผังเวลาของเอฟเอ็มดีมอดูเลชัน ดังรูปที่ 3.4



รูปที่ 3.4 ผังเวลาของเอฟเอ็มมอดูเลชัน

การสร้าง ช่วงเวลามาตรฐาน เพื่อแยกข้อมูลออกมามี 2 แบบคือ แบบความยาวคงที่และแบบความยาวเปลี่ยนแปลงได้

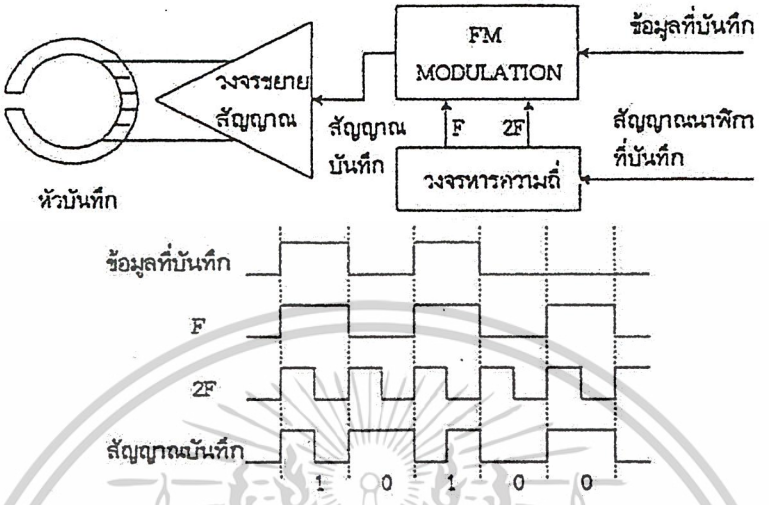
แบบความยาวคงที่จะกำหนด ช่วงเวลามาตรฐาน จะให้ค่าคงที่และเปรียบเทียบกับทุกบิต ส่วนแบบที่ความยาวเปลี่ยนแปลงได้ จะวัดความยาวของบิตก่อนหน้าและสร้าง ช่วงเวลามาตรฐาน จากจุดนั้นแล้วนำไปเปรียบเทียบกับบิตถัดไป

สำหรับการสร้าง ช่วงเวลามาตรฐาน แบบเปลี่ยนแปลงความยาวได้นั้น ขณะที่กำลังอ่านบัตรอยู่ถ้าความเร็วในการป้อนบัตรเปลี่ยนแปลง ช่วงเวลามาตรฐาน จะเปลี่ยนแปลงตามไปด้วย ซึ่งวิธีนี้เหมาะกับการใช้กับเครื่องอ่านบัตรที่ป้อนบัตรด้วยมือ เนื่องจากความเร็วในการรูดบัตรด้วยมือนั้นมีค่าไม่แน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 รูปแบบของการบันทึกข้อมูลบนบัตรแม่เหล็ก

รูปแบบพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก จะใช้สัญญาณความถี่สองความถี่ซึ่งมีเฟสตรงกันมาใช้ในการบันทึกข้อมูล ดังแสดงไว้ในรูปที่ 3.5 ซึ่งการบันทึกในลักษณะเช่นนี้จะบันทึกข้อมูล และสัญญาณนาฬิกาที่ใช้ในการบันทึกเข้าไว้ในแทร็คเดียวกัน ซึ่งคิดค้นโดย Aiken (1954)

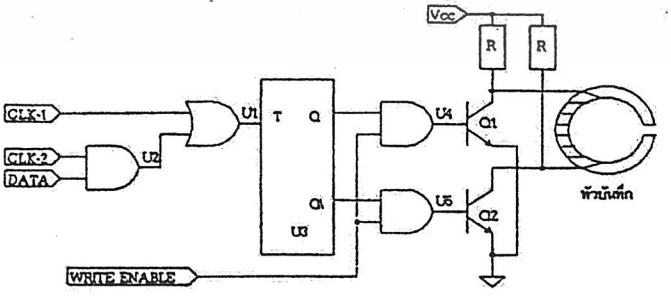


รูปที่ 3.5 รูปแบบของการบันทึกโดยใช้สัญญาณความถี่สองความถี่ที่มีเฟสตรงกัน

จากรูปที่ 3.5 สัญญาณบันทึกจะประกอบด้วยบิตของข้อมูลที่บันทึก และบิตของสัญญาณนาฬิกา (2F) สำหรับการเปลี่ยนแปลงฟลักซ์แม่เหล็กของสัญญาณบันทึก จะเกิดขึ้นในระหว่างที่สัญญาณข้อมูลมีสภาวะ “1” ทางลอจิก หรือระหว่างสัญญาณบิตข้อมูลติดกัน และจะไม่มีการเปลี่ยนแปลงของฟลักซ์แม่เหล็ก ในระหว่างที่สัญญาณข้อมูลมีสภาวะ “0” ทางลอจิกสำหรับการบันทึกบิต ข้อมูลของแต่ละตัวเลขหรือตัวอักษรบนแถบแม่เหล็ก จะเป็นการบันทึกข้อมูลที่มีค่านัยสำคัญต่ำสุดก่อน และบิตพาริตี้ เป็นบิตสุดท้ายที่ถูกบันทึก ส่วนค่าความหนาแน่นในการบันทึกข้อมูลในแทร็คที่สอง ISO ได้กำหนดให้บันทึกด้วยความหนาแน่น 75±20 บิตต่อนิ้ว

3.1.2 วงจรพื้นฐานสำหรับการบันทึกข้อมูลบนแถบแม่เหล็ก

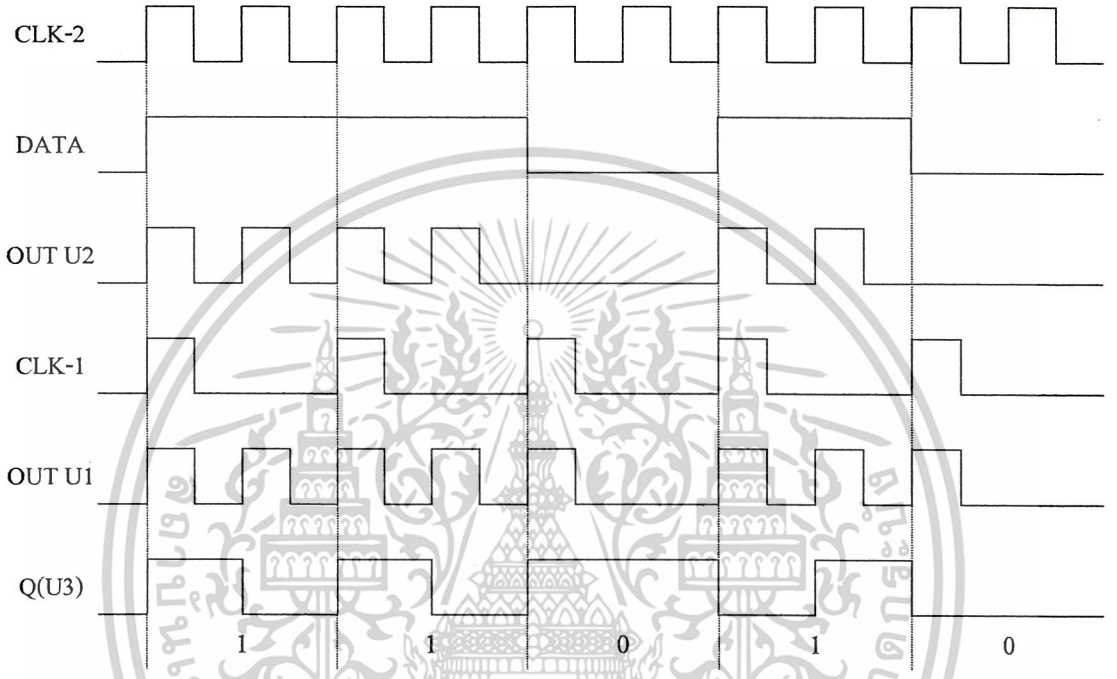
วงจรพื้นฐานสำหรับการบันทึกข้อมูลบนแถบแม่เหล็ก แสดงในรูป 3.6



รูปที่ 3.6 รูปวงจรพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

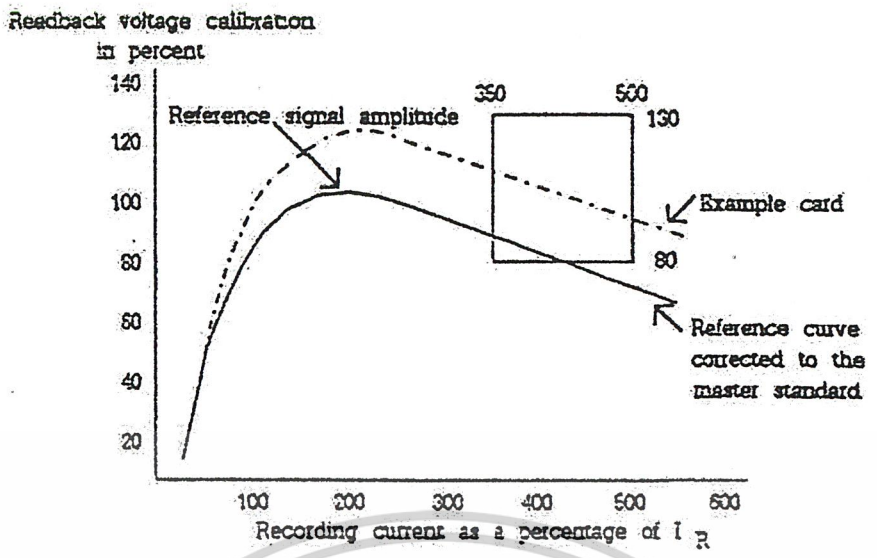
จากรูปที่ 3.6 ไอซี U1,U2,U3,U4 และ U5 จัดรูปวงจรมอดูเลทสัญญาณข้อมูลและสัญญาณนาฬิกาให้อยู่ในรูปแบบของสัญญาณเอฟเอ็ม แล้วป้อนสัญญาณดังกล่าว ให้กับวงจรจ่ายกระแสไฟฟ้าให้กับขดลวดที่พันรอบแกนของหัวบันทึก เมื่อทำการบันทึกข้อมูล สัญญาณ WRITE ENABLE มีลอจิกเป็น "1" พร้อมกับทำให้ทรานซิสเตอร์ Q1 และ Q2 ทำงาน ซึ่งทรานซิสเตอร์ทั้งคู่จะสลับกันทำงานเพื่อป้อนกระแสพัลส์ด้านบวก และด้านลบให้แก่ขดลวดตามสัญญาณการบันทึกโดยรูปสัญญาณการทำงานของการจัดวงจรพื้นฐานนี้แสดงในรูปที่ 3.7



รูปที่ 3.7 รูปสัญญาณการทำงานของวงจรพื้นฐานสำหรับการบันทึกข้อมูลบนบัตรแม่เหล็ก

สำหรับกระแสพัลส์ที่ใช้ในการบันทึกข้อมูล สามารถหาได้จากกราฟแสดงคุณสมบัติการอิ่มตัวของแถบแม่เหล็กบนบัตรแม่เหล็ก ดังแสดงในรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

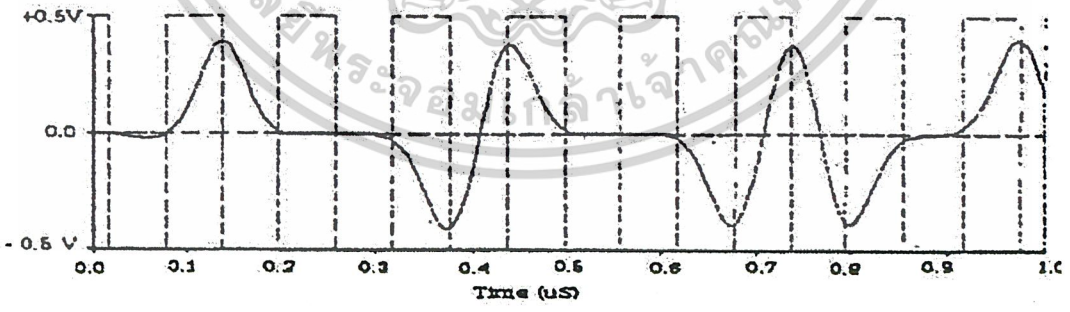


รูปที่ 3.8 กราฟคุณสมบัติการอิ่มตัวของแถบแม่เหล็กบนบัตรแม่เหล็กมาตรฐาน

จากรูปที่ 3.8 กระแสพัลส์สำหรับการบันทึกสามารถใช้ค่า 350% หรือ 500% ของกระแสอ้างอิง (I_R) คือ ค่าแอมปลิจูดที่น้อยที่สุดของกระแสบันทึก ณ ตำแหน่ง 80% ของค่าแอมปลิจูดแรงดันไฟฟ้าสูงสุดที่อ่านได้จากบัตร

3.1.3 การอ่านข้อมูลจากบัตรแม่เหล็ก

การอ่านข้อมูลจากบัตรแม่เหล็ก สามารถทำได้โดยให้แถบแม่เหล็กสัมผัสกับหัวอ่าน ซึ่งเคลื่อนที่ด้วยความเร็วคงที่ ดังแสดงในรูปที่ 3.9 ฟลักซ์แม่เหล็กที่เกิดขึ้นจะผ่านจากแกป (gap) ของแกนหัวอ่านไปยังขดลวดที่พันอยู่รอบแกน ซึ่งการเปลี่ยนแปลงของฟลักซ์แม่เหล็กตามข้อมูลที่บันทึกจะทำให้เกิดแรงดันไฟฟ้าที่ขดลวดของหัวอ่านนั้นตามข้อมูลนั้น ดังแสดงในรูปที่ 3.9



รูปที่ 3.9 สัญญาณแรงดันไฟฟ้าจากหัวอ่านถูกขยายด้วยอัตราขยายประมาณ 100 เท่า

จากรูปที่ 3.9 ตำแหน่งแรงดันสูงสุดของแรงดันไฟฟ้าที่อ่านได้ จะตรงกับตำแหน่งที่สนามแม่เหล็กบนแถบแม่เหล็กมีการกลับทิศทาง ทำให้สามารถนำตำแหน่งนี้มาสร้างสัญญาณพัลส์ เพื่ออ่านข้อมูลที่บันทึกอยู่บนบัตรแม่เหล็กได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.1 วงจรพื้นฐานสำหรับการสร้างสัญญาณรูปแบบ ISO จากบัตรแม่เหล็ก

ส่วนประกอบของวงจรพื้นฐานสำหรับสร้างสัญญาณรูปแบบ ISO จากแถบแม่เหล็ก แสดงในรูปที่ 3.10

รูปที่ 3.10 ส่วนประกอบของวงจรพื้นฐานสร้างสัญญาณรูปแบบ ISO จากบัตรแม่เหล็ก

1. วงจรขยายสัญญาณแรงดันไฟฟ้า

สัญญาณแรงดันไฟฟ้าจากหัวอ่านข้อมูลมีค่าค่อนข้างต่ำ (5-10 มิลลิโวลท์) จึงจำเป็นที่ต้องขยายสัญญาณเพื่อสะดวกต่อการนำไปสร้างเป็นสัญญาณรูปเหลี่ยม โดยการใช้วงจรขยายสัญญาณสองชุด และกำหนดค่ากำลังขยายสัญญาณรวมมีค่าประมาณ 1000 เท่า (ไม่ให้สัญญาณอิ่มตัว) ดังแสดงไว้ในรูปที่ 3.11

ก. วงจรขยายสัญญาณความแตกต่าง (Differential Amplifier)

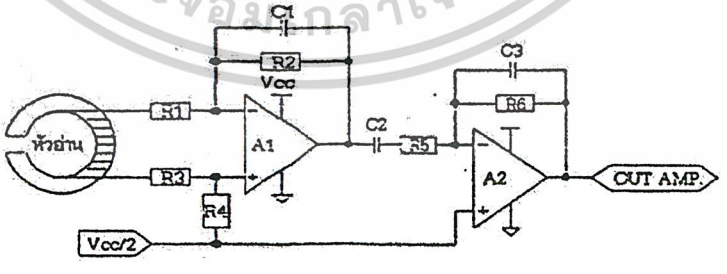
เป็นวงจรขยายสัญญาณชุดแรกที่รับสัญญาณข้อมูลจากขั้วทั้งสองของหัวอ่านข้อมูลซึ่งมีค่าความต้านทานต่ำ และใช้ IC A_1 ทำหน้าที่ขยายสัญญาณความแตกต่าง มีค่าอัตราขยาย

$$A_V = -R_2/R_1$$

ข. วงจรขยายสัญญาณแบบกลับเฟส (Inverting Amplifier)

เพื่อให้ได้สัญญาณข้อมูลที่มีค่าแรงดันไฟฟ้าสูงขึ้นพอที่จะป้อนให้กับวงจรสร้างสัญญาณรูปเหลี่ยม จึงนำสัญญาณข้อมูลที่เข้ามาผ่านวงจรขยายแบบกลับเฟส ซึ่งอัตราขยายแรงดันไฟฟ้า

$$A_V = -R_6/R_5$$



รูปที่ 3.11 วงจรขยายสัญญาณแรงดันไฟฟ้า สำหรับการอ่านบัตรแม่เหล็ก

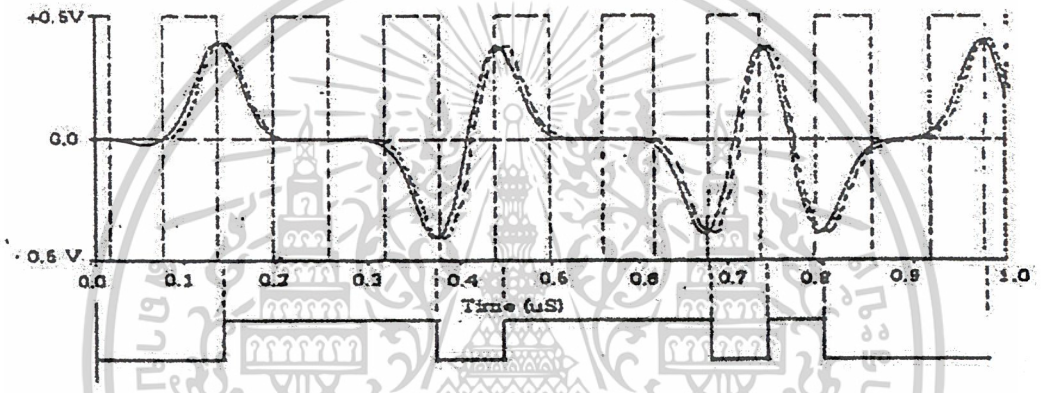
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วงจรสร้างสัญญาณรูปเหลี่ยม (รูปแบบ ISO)

จากสัญญาณข้อมูลที่ขยายสัญญาณแล้ว (จากรูปที่ 3.9) ทำให้สามารถพิจารณารูปแบบการสร้างสัญญาณรูปเหลี่ยมได้หลายรูปแบบ โดยเฉพาะการนำตำแหน่งจุดยอดของสัญญาณข้อมูลทั้งด้านบนและด้านล่าง มาใช้ในการสร้างสัญญาณรูปเหลี่ยม จะทำให้ได้สัญญาณพัลส์ที่เหมือนกับสัญญาณที่ใช้ในการบันทึกข้อมูลบนบัตรแม่เหล็ก สำหรับการสร้างสัญญาณรูปเหลี่ยมสามารถสร้างได้ 2 วิธีคือ

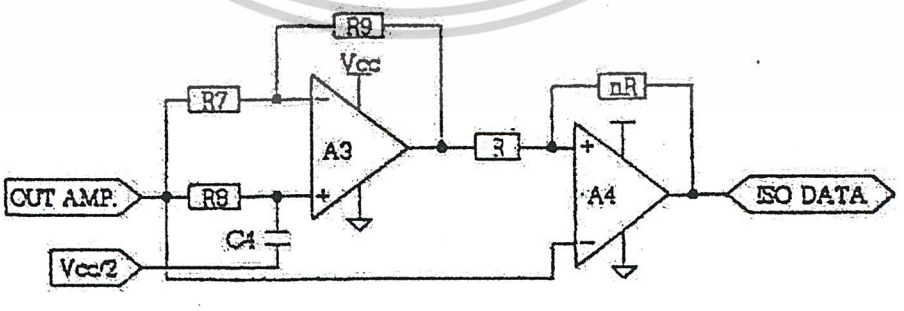
ก. วงจรเลื่อนเฟส และวงจรเปรียบเทียบแรงดันไฟฟ้า

ใช้หลักการหน่วงสัญญาณข้อมูล ในช่วงเวลาสั้นมากๆ แล้วนำสัญญาณข้อมูลจริง (เส้นทึบ) เปรียบเทียบกับสัญญาณข้อมูลที่ถูกหน่วง (เส้นประ) ซึ่งสัญญาณเอาท์พุทจากการเปรียบเทียบนี้แสดงไว้ในรูปที่ 3.12



รูปที่ 3.12 รูปการเปรียบเทียบสัญญาณและลักษณะสัญญาณที่ต้องการ

จากรูปที่ 3.12 วงจรที่ใช้งานจะใช้ออปแอมป์ 2 ตัว โดยที่ออปแอมป์ตัวแรกจัดรูปวงจรเป็นวงจรเลื่อนเฟสเพื่อให้ได้สัญญาณข้อมูลที่ถูกหน่วง (เอาท์พุทเป็นสัญญาณเส้นประ) และออปแอมป์ตัวที่สองจัดรูปวงจรเปรียบเทียบสัญญาณแรงดันระหว่างสัญญาณข้อมูลกับสัญญาณข้อมูลที่ถูกหน่วง ดังแสดงไว้ในรูปที่ 3.13

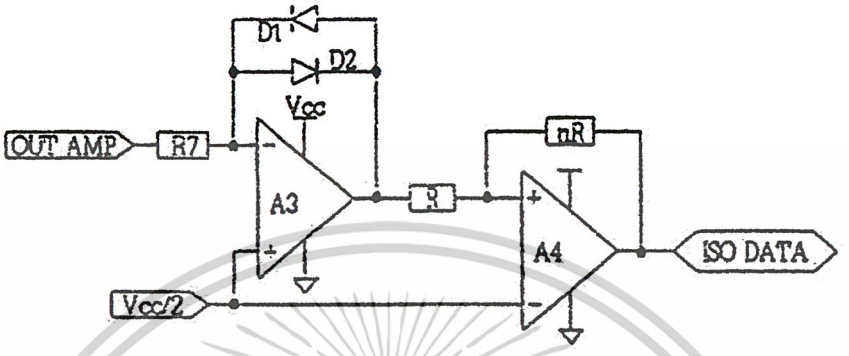


รูปที่ 3.13 วงจรเลื่อนเฟสและวงจรเปรียบเทียบสัญญาณแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. วงจรดิฟเฟอร์เรนทิเอเตอร์และวงจรเปรียบเทียบแรงดันไฟฟ้า

แนวทางที่สอง จะใช้วงจรดิฟเฟอร์เรนทิเอเตอร์ ซึ่งมีไดโอด D1 และ D2 ประพฤติตัวเป็นเสมือนตัวต้านทานป้อนกลับด้านลบของวงจร ทำงานร่วมกับวงจรเปรียบเทียบแรงดันไฟฟ้า ดังแสดงในรูปที่ 3.14



รูปที่ 3.14 วงจรดิฟเฟอร์เรนทิเอเตอร์และวงจรเปรียบเทียบแรงดันไฟฟ้า

จากรูปที่ 3.14 เมื่อพิจารณาสัญญาณข้อมูลจากหัวอ่านข้อมูล ซึ่งมีลักษณะของสัญญาณคล้ายรูปคลื่นสามเหลี่ยม เมื่อป้อนสัญญาณข้อมูลให้กับวงจรดิฟเฟอร์เรนทิเอเตอร์สัญญาณเอาท์พุทที่ได้จะเป็นสัดส่วนกับค่าอนุพันธ์ของสัญญาณอินพุท เมื่อสัญญาณข้อมูลเป็นรูปคลื่นสามเหลี่ยม ที่มีค่าสโลปเป็นบวกและค่าลบที่สวิงบนค่าแรงดันอ้างอิง (Vcc/2) สัญญาณเอาท์พุทของวงจรดิฟเฟอร์เรนทิเอเตอร์ สัญญาณรูปคลื่นสี่เหลี่ยมมีค่าแรงดันไฟฟ้าเอาท์พุทตามสมการที่ 3.1

$$V_o = -RCk_n \qquad \text{.....3.1}$$

เมื่อค่า k_n เป็นค่าสโลปของสัญญาณอินพุท (โวลท์ต่อวินาที) และใช้วงจรเปรียบเทียบสัญญาณแรงดันไฟฟ้าเป็นส่วนสร้างสัญญาณรูปเหลี่ยมให้ดียิ่งขึ้น

สำหรับวงจรเปรียบเทียบแรงดันไฟฟ้าของทั้งสองแนวทาง จะใช้ตัวต้านทาน R และ nR จัดรูปวงจรเป็นแบบป้อนกลับทางด้านบวก เพื่อป้องกันมิให้วงจรเกิดการออสซิลเลท โดยมีค่า Upper Threshold Voltage (V_{UT}) , Lower Threshold Voltage (V_{LT}) และค่าศักดาไฟฟ้าฮิสเตอริซิส (V_T) ตามสมการที่ 3.2, 3.3 และ 3.4 ตามลำดับ

$$V_{UT} = Vref\,[1+(1/n)\,] - (-Vsat/n) \qquad \text{.....3.2}$$

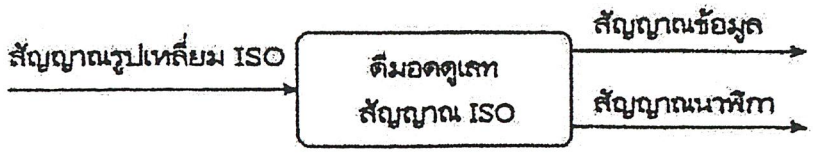
$$V_{LT} = Vref\,[1+(1/n)\,] - (+Vsat/n) \qquad \text{.....3.3}$$

$$V_T = [\,(+Vsat) - (-Vsat)\,] / n \qquad \text{.....3.4}$$

3.1.3.2 วงจรอ่านข้อมูลบัตรแม่เหล็ก

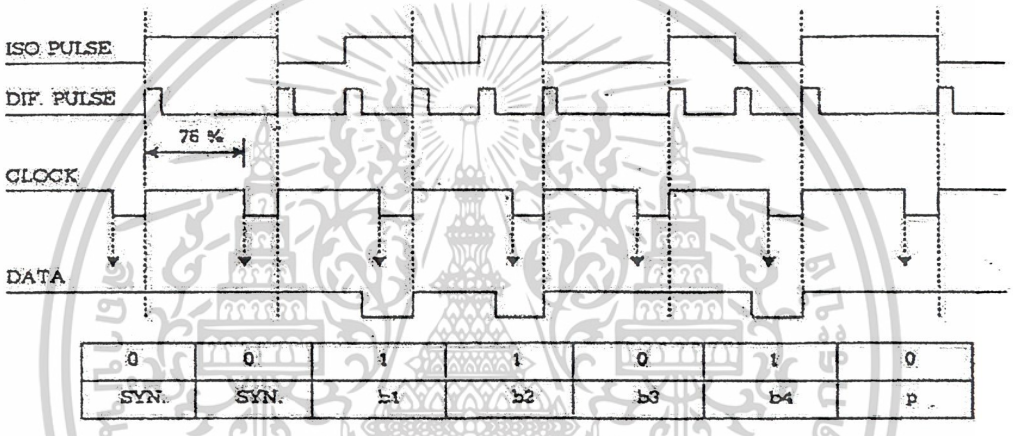
จากข้อมูลพื้นฐานในหัวข้อ 3..3.1 สามารถนำสัญญาณดังกล่าวมาประยุกต์ใช้งานต่อเนื่องได้เพื่อให้อ่านข้อมูลจากบัตรแม่หล็กได้สะดวกขึ้นโดยการดีมอดูเลทสัญญาณรูปเหลี่ยม ISO เพื่อให้ได้สัญญาณนาฬิกาสำหรับการอ่านและสัญญาณข้อมูล ดังแสดงในรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แผนภาพของวงจรอ่านบัตรแม่เหล็ก

การแยกสัญญาณข้อมูลออกจากสัญญาณนาฬิกา โดยใช้หลักการดีมอดูเลทสัญญาณ ISO สามารถพิจารณาได้จากคลื่นสัญญาณ ดังแสดงในรูปที่ 3.16



รูปที่ 3.16 รูปคลื่นสัญญาณของ ISO พัลส์ที่ถูกดีมอดูเลท

จากรูปที่ 3.16 การแยกสัญญาณข้อมูลออกจากสัญญาณรูปเหลี่ยม (ISO) ได้นั้น จะต้องพิจารณาการกลับขั้วของสัญญาณรูปเหลี่ยม ISO ในช่วง ช่วงเวลามาตรฐาน หรือไม่ ถ้ามีการกลับขั้วของสัญญาณรูปเหลี่ยม ISO แสดงว่าข้อมูลมีค่าทางลอจิกเป็น “1” และถ้าไม่มีการกลับขั้วของสัญญาณรูปเหลี่ยม ISO แสดงว่าข้อมูลมีค่าทางลอจิกเป็น “0” โดยที่ค่า ช่วงเวลามาตรฐาน มีค่าประมาณ 75% ของคาบเวลา 1 บิต ส่วนสัญญาณที่ได้จากการดีมอดูเลท จะมีสัญญาณข้อมูลและสัญญาณนาฬิกาสำหรับอ่านข้อมูล โดยที่ทุกๆ ขอบขาลงของสัญญาณนาฬิกา บิตข้อมูลจริงจะกลับลอจิกกับสัญญาณข้อมูลที่อ่านได้ สำหรับการสร้างสัญญาณ ช่วงเวลามาตรฐาน เพื่อใช้สำหรับแยกสัญญาณข้อมูลออกมานั้นมี 2 วิธี คือ แบบความยาวคงที่และแบบเปลี่ยนความยาวได้

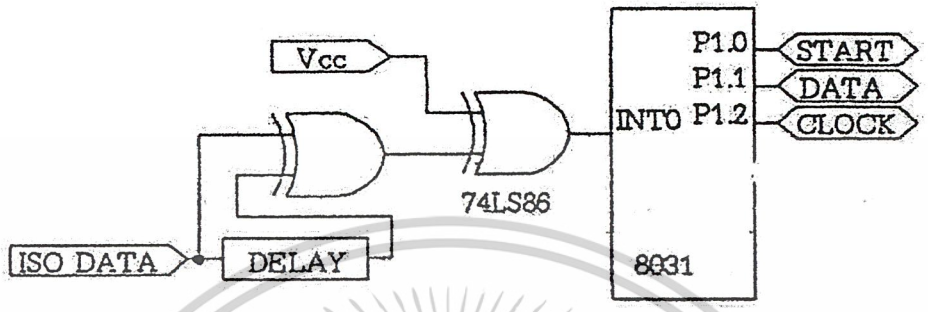
- แบบความยาวคงที่ สามารถกำหนด ช่วงเวลามาตรฐาน ให้มีค่าคงที่และเปรียบเทียบกับทุกบิต วิธีนี้เหมาะกับชุดอ่านบัตรแม่เหล็กที่ใช้มอเตอร์ดึงบัตรผ่านหัวอ่านด้วยความเร็วคงที่

- แบบความยาวเปลี่ยนแปลงได้ ค่า ช่วงเวลามาตรฐาน มีค่าไม่คงที่ สัญญาณ ช่วงเวลามาตรฐาน จะถูกสร้างจากความยาวของบิตก่อนหน้านี้ 1 บิต และสัญญาณ ช่วงเวลามาตรฐาน ลูกต่อไปจะสร้างจากความยาวของบิตปัจจุบัน โดยจะเปรียบเทียบกันไปจนหมดข้อมูล ซึ่งวิธีนี้ใช้กับชุดอ่านบัตรแบบที่ใช้การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

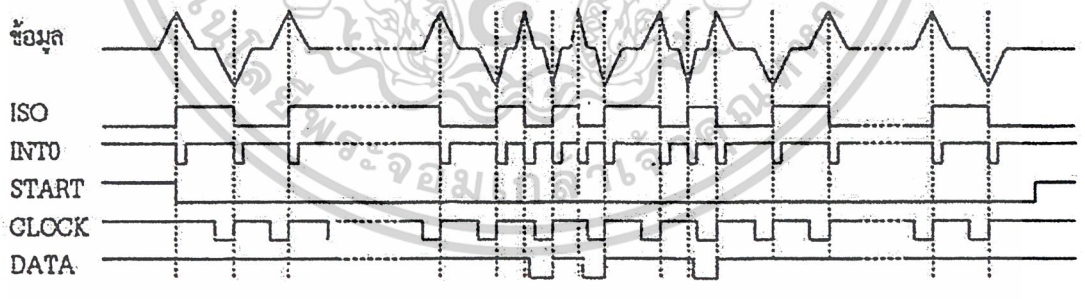
รูดผ่านหัวอ่านด้วยมือ เนื่องจากความเร็วในการรูดมีค่าไม่แน่นอน และจากรูปคลื่น สัญญาณที่แสดงในรูปที่ 3.16 ได้นำ

8031 ไมโครคอนโทรลเลอร์มาประยุกต์ใช้ในการสร้างสัญญาณข้อมูล และสัญญาณ ช่วงเวลามาตรฐาน จากสัญญาณรูปเหลี่ยม ISO โดยจัดรูปวงจรแสดงไว้ในรูปที่ 3.17



รูปที่ 3.17 วงจรสร้างสัญญาณข้อมูลและสัญญาณนาฬิกาอ่านข้อมูลโดยใช้ 8031

จากรูปที่ 3.17 หลักการทำงานจะใช้ไอซี 74LS86 เป็นตัวสร้างสัญญาณ Difference Pulse แล้วป้อนให้กับ 8031 ทางขา INT0 การทำงานในส่วนของ 8031 จะใช้ ไทเมอร์ 0 ของ 8031 เป็นตัวจับคาบเวลาแต่ละช่วงเวลาของ Difference Pulse ถ้าช่วงเวลามีค่ามากกว่า ช่วงเวลามาตรฐาน (75%ของช่วงเวลาระหว่าง difference pulse ชุดก่อน) จะสร้างสัญญาณนาฬิกาและสร้างสัญญาณข้อมูลเป็น “1” แต่ถ้าช่วงเวลามีค่าน้อยกว่า ช่วงเวลามาตรฐาน จะไม่สร้างสัญญาณนาฬิกาทันที แต่จะไปสร้างสัญญาณนาฬิกา ณ เวลาที่ 75% ของช่วงเวลาชุดก่อน

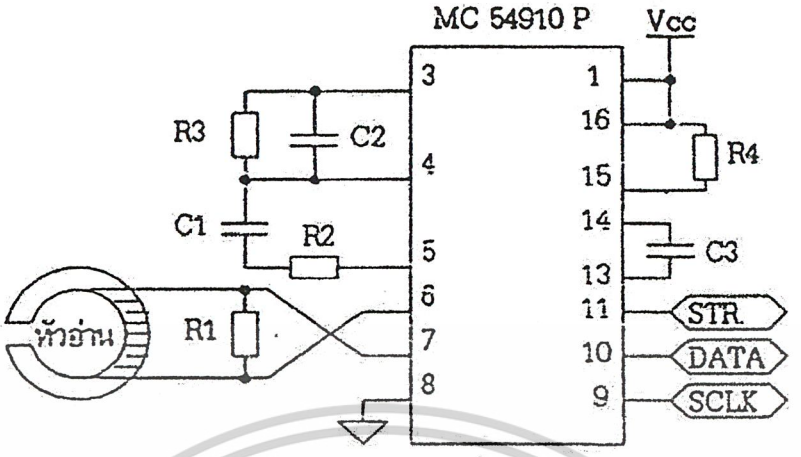


รูปที่ 3.18 รูปสัญญาณเอาท์พุทของ 8031 สำหรับสร้างสัญญาณอ่านบัตรแม่เหล็ก

3.1.3.3 วงจรอ่านข้อมูลบัตรแม่เหล็กโดยใช้ MC54910P

MC54910P เป็นชิพไอซีที่ใช้ในการอ่านบัตรแม่เหล็กแบบรูดบัตรด้วยมือโดยเฉพาะ จัดรูปวงจรร่วมกับอุปกรณ์รอบข้างดังแสดงในรูปที่ 3.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



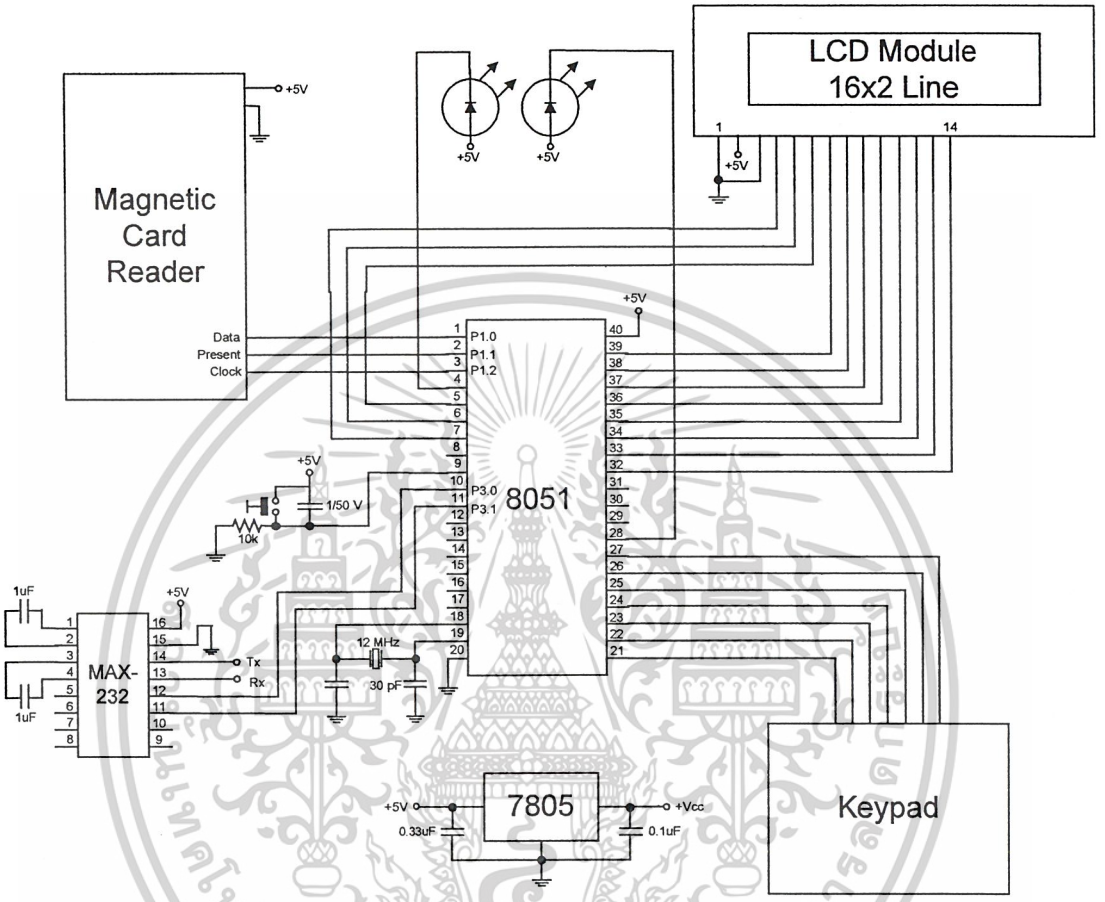
รูปที่ 3.19 รูปวงจรอ่านบัตรแม่เหล็กโดยใช้ MC54910P

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจรการทำงานของไมโครคอนโทรลเลอร์ MCS-51

ในการควบคุมการทำงานของเครื่องอ่านบัตร,แอลซีดี โมดูล, คีย์แพด และการรับ/ส่งข้อมูลกับคอมพิวเตอร์จะกระทำโดยไมโครคอนโทรเลอร์ MCS-51 โดยมีรูปวงจรดังรูปที่ 3.20

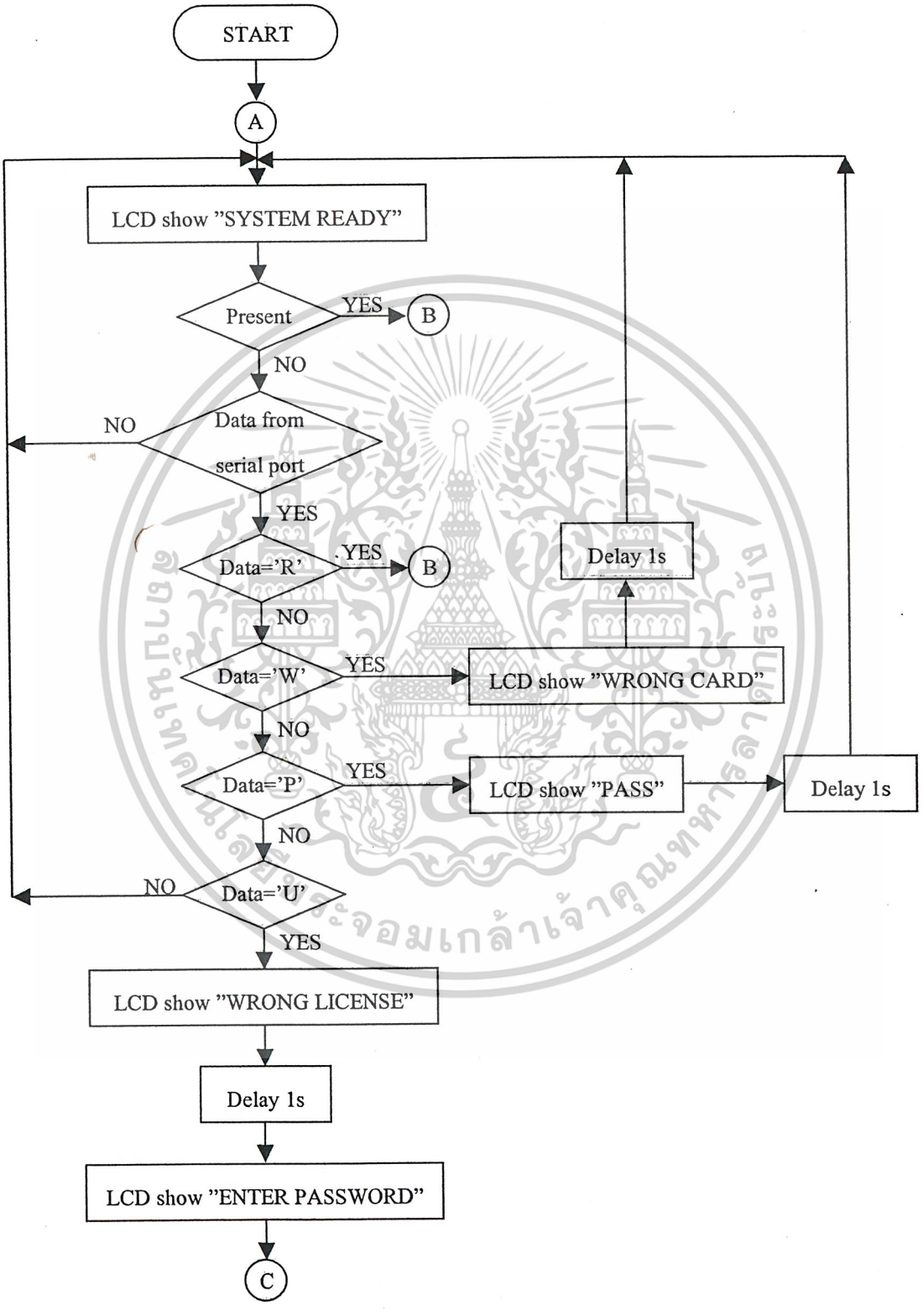


รูปที่ 3.20 รูปวงจรการทำงานของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

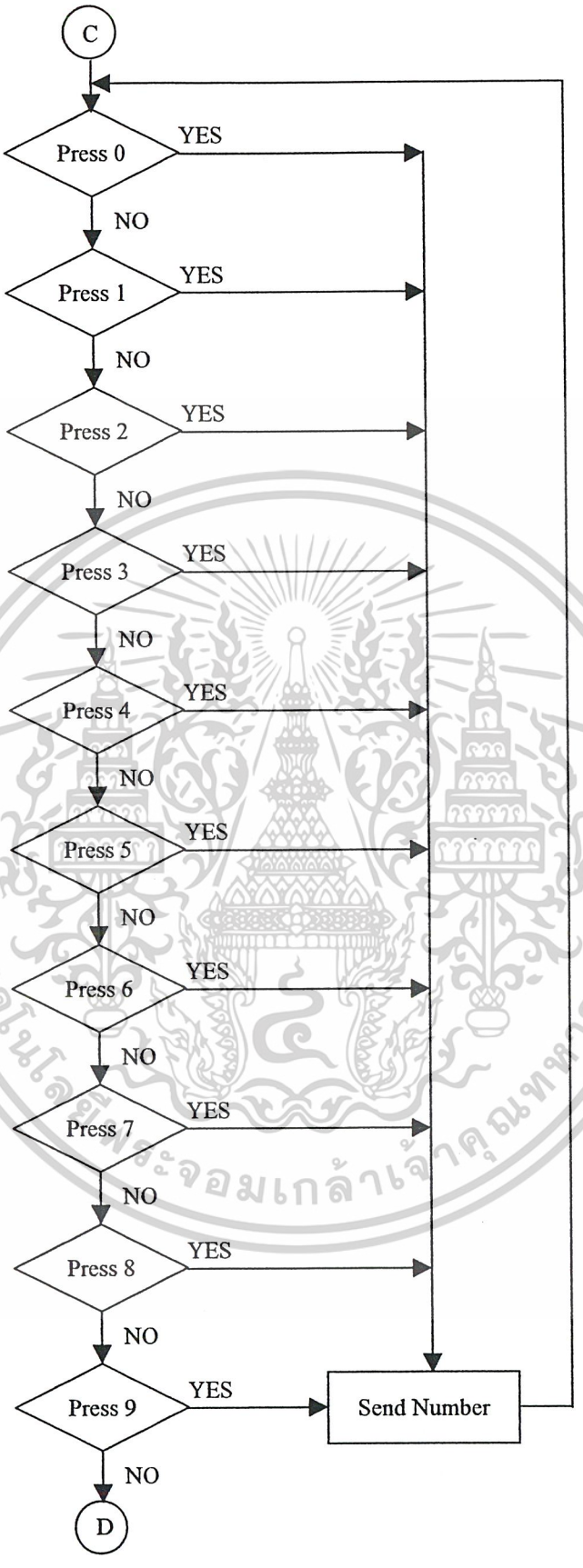
3.3 การออกแบบโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51

ในการสร้างโปรแกรมควบคุมการทำงานของเครื่องอ่านบัตรแม่เหล็ก,แอลซีดี โมดูล,คีย์แพด และการรับ/ส่งข้อมูลกับคอมพิวเตอร์ของไมโครคอนโทรลเลอร์ MCS-51 มีโฟลว์ชาร์ตดังนี้



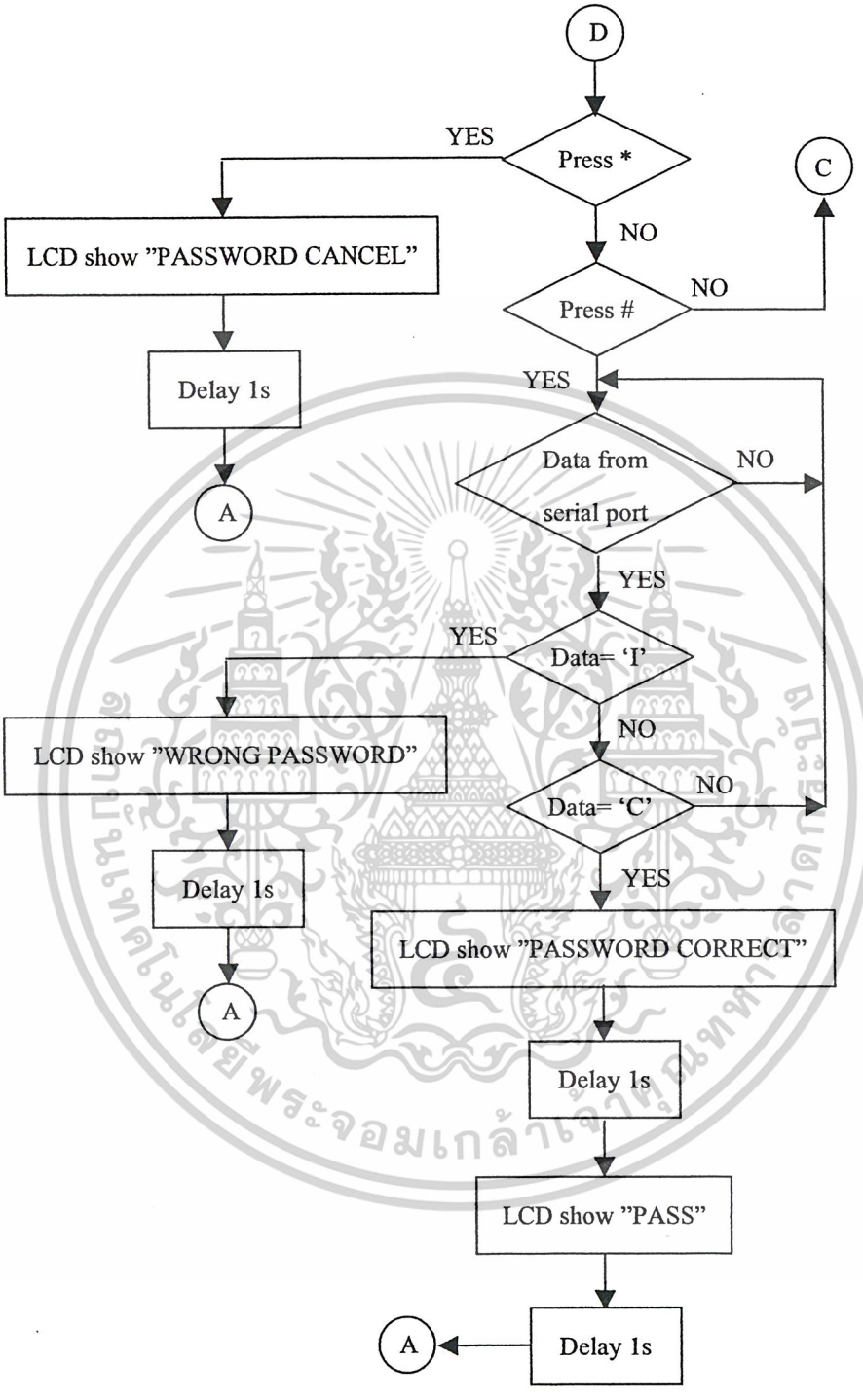
รูปที่ 3.21 โฟล์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 โฟล์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ (ต่อ)

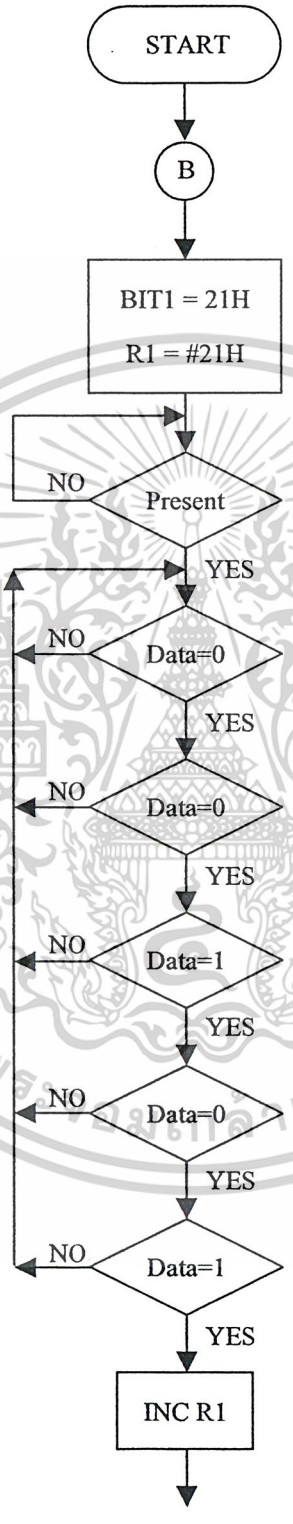
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 โฟล์ชาร์ตการทำงานของไมโครคอนโทรลเลอร์ (ต่อ)

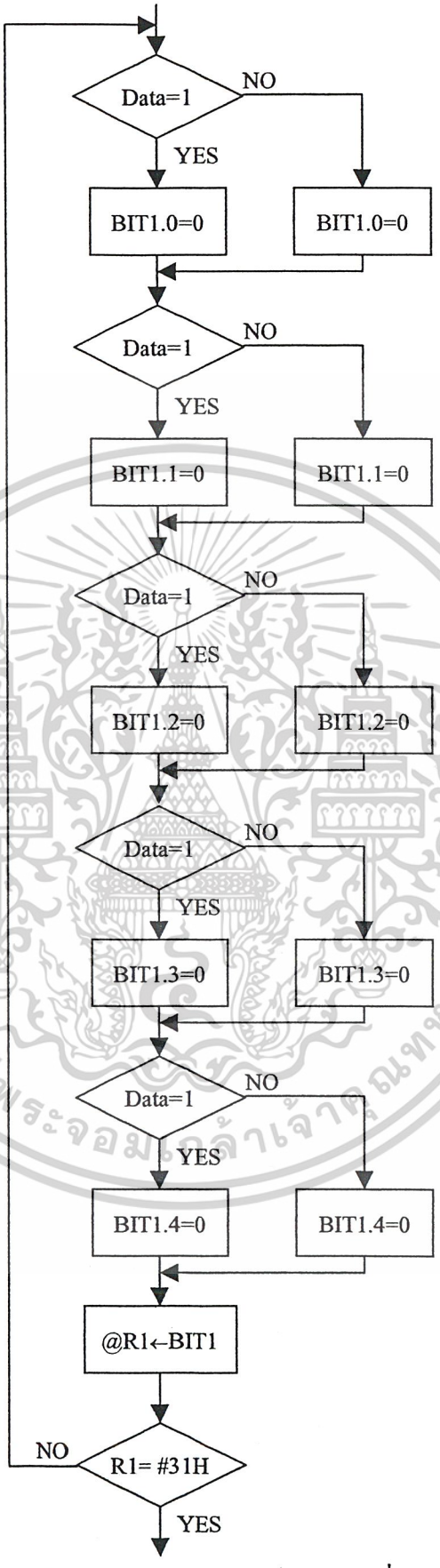
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบโปรแกรมสำหรับการอ่านข้อมูลจากเครื่องอ่านบัตรแม่เหล็ก



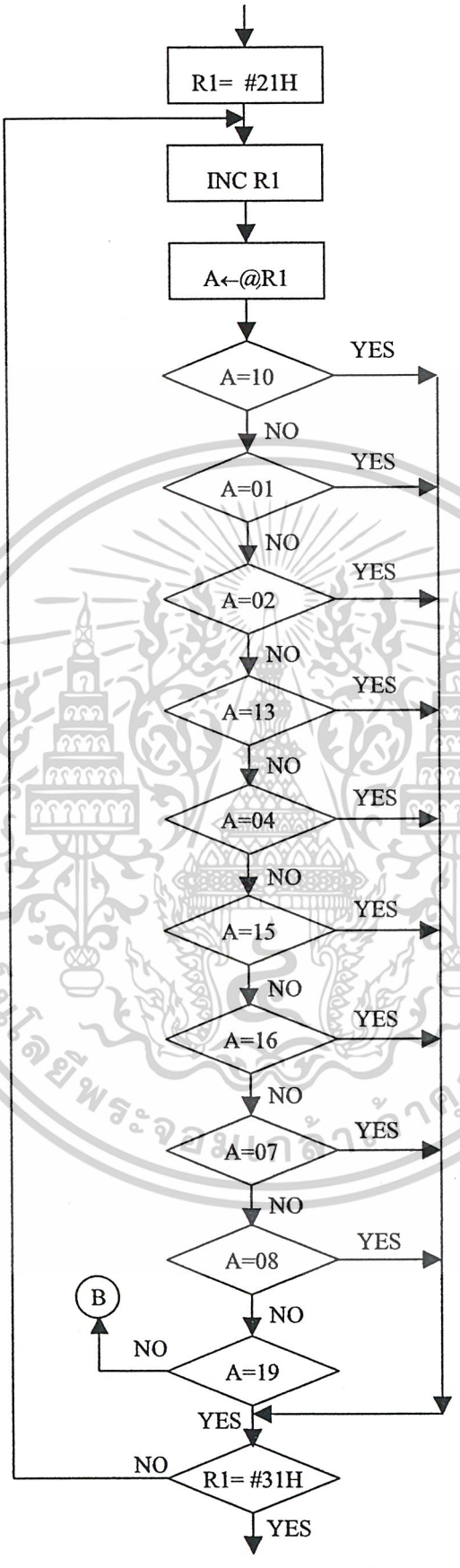
รูปที่ 3.22 โฟลว์ชาร์ตการอ่านข้อมูลจากเครื่องอ่านบัตรแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



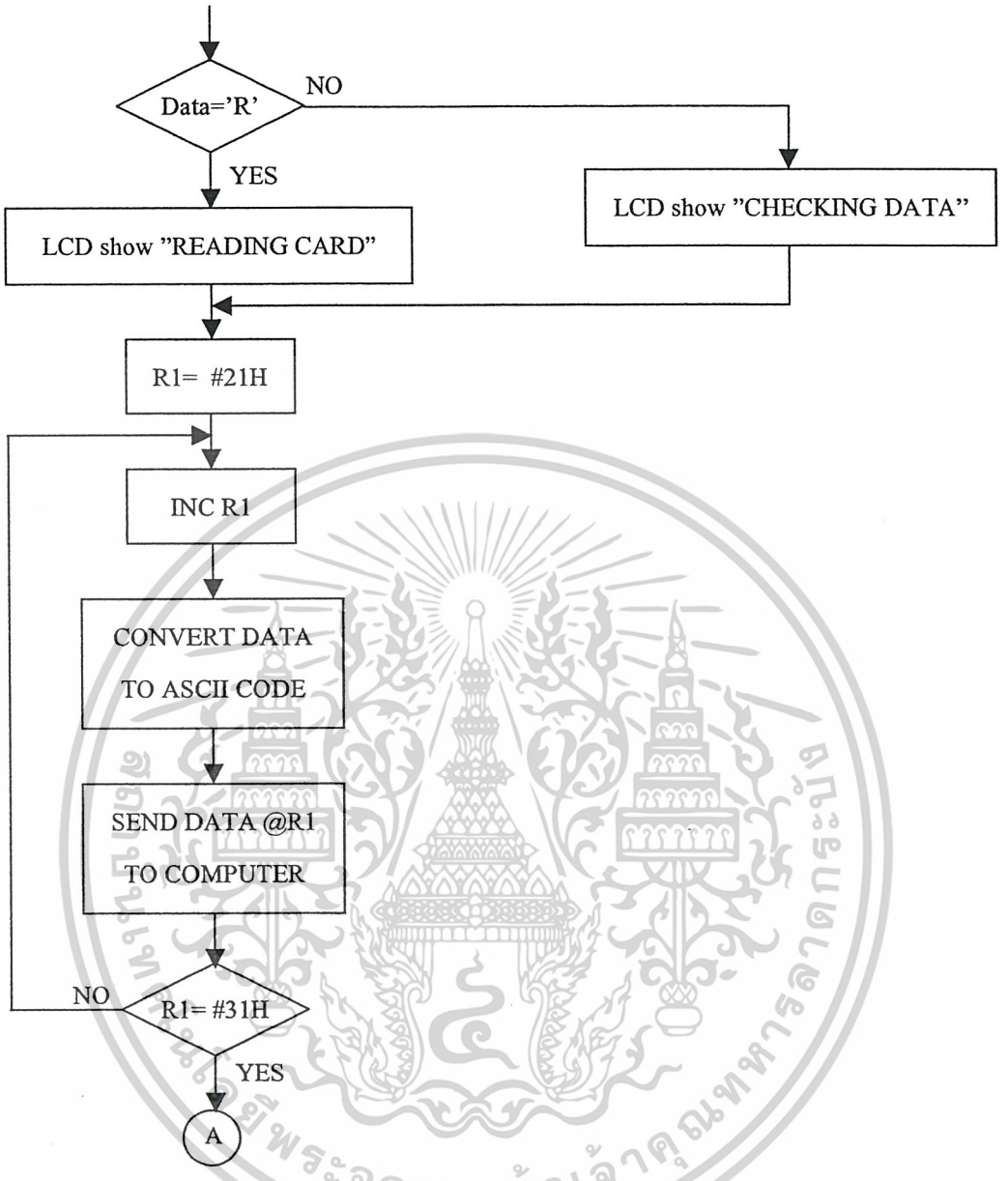
รูปที่ 3.22 โฟลว์ชาร์ตการทำงานของโปรแกรมสำหรับการอ่านข้อมูลจากเครื่องอ่านบัตรแม่เหล็ก(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 โฟลว์ชาร์ตการทำงานของโปรแกรมสำหรับการอ่านข้อมูลจากเครื่องอ่านบัตรแม่เหล็ก(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

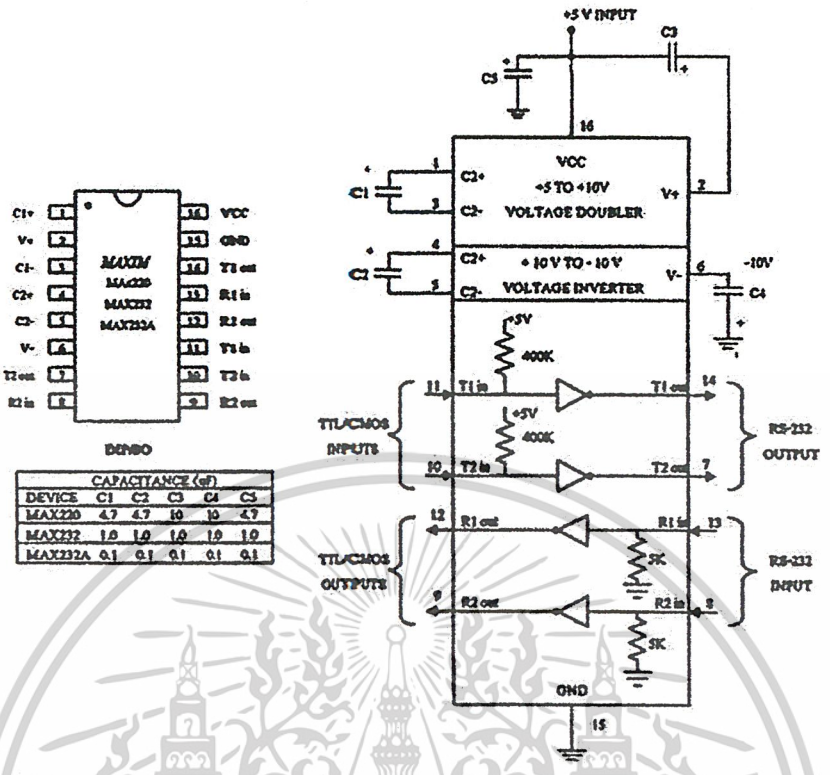


รูปที่ 3.22 โฟลว์ชาร์ตการทำงานของโปรแกรมสำหรับการอ่านข้อมูลจากเครื่องอ่านบัตรแม่เหล็ก(ต่อ)

3.5 การแปลงสัญญาณแรงดันเป็นสัญญาณทีทีแอล (TTL)

ไอซี MAX-232 มีหน้าที่ในการเปลี่ยนสัญญาณแรงดันที่ออกจากพอร์ทอนุกรม RS-232 ให้เป็นสัญญาณทีทีแอลโดยเมื่อแรงดันที่ออกจาก RS-232 มีขนาดแรงดันเท่ากับ 12 โวลท์ หรือลอจิก "0" จะทำการเปลี่ยนไปเป็นสัญญาณทีทีแอล ขนาด 0 โวลท์ และเมื่อมีแรงดันที่ออกจาก RS-232 มีขนาดแรงดันเท่ากับ –12 โวลท์หรือลอจิก "1" จะทำการเปลี่ยนไปเป็นสัญญาณทีทีแอล ขนาด 5 โวลท์ ซึ่งค่าต่างๆ ในวงจรและรูปวงจรการต่อแสดงดังรูป 3.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 วงจร MAX-232

3.6 ฐานข้อมูล

ในการเขียนโปรแกรมฐานข้อมูลจะใช้คลาส(Class) ใน VC++ ที่ชื่อว่า CObList ในการจัดการเกี่ยวกับฐานข้อมูล โดยมีข้อมูล 3 ฟิลด์คือ ฟิลด์หมายเลขบัตร, ฟิลด์หมายเลขทะเบียนรถยนต์ และฟิลด์รหัสผ่านโดยคลาส CObList มีเมมเบอร์ฟังก์ชัน (Member Function) ที่สำคัญ ดังนี้

- GetHeadPositon ใช้เพื่อแสดงข้อมูลแรกสุดในฐานข้อมูล
- GetTailPosition ใช้เพื่อแสดงข้อมูลท้ายสุดในฐานข้อมูล
- GetPrev ใช้เพื่อแสดงข้อมูลที่อยู่ก่อนหน้าข้อมูลที่แสดงอยู่ปัจจุบัน
- GetNext ใช้เพื่อแสดงข้อมูลที่อยู่หลังข้อมูลที่แสดงอยู่ปัจจุบัน
- RemoveAt ใช้เพื่อลบข้อมูล ณ ตำแหน่งที่ระบุ
- InsertAfter ใช้เพื่อเพิ่มข้อมูล ณ ตำแหน่งหลังจากที่อยู่ปัจจุบัน
- InsertBefore ใช้เพื่อเพิ่มข้อมูล ณ ตำแหน่งก่อนที่อยู่ปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การรับภาพจากล้องและการหาหมายเลขทะเบียนรถยนต์จากภาพ

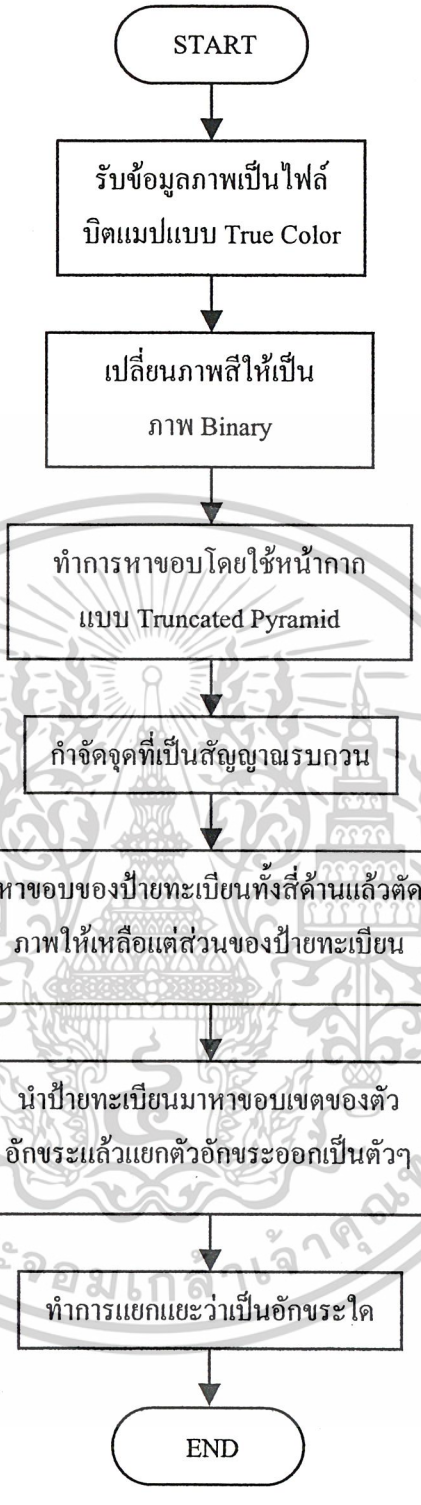
3.7.1 การรับภาพจากกล้อง

ในการรับภาพจากกล้องจะใช้คอนโทรล (Control) ใน VC++ ที่ชื่อว่า VideoOCXCtrl ร่วมกับคลาสที่ชื่อว่า CvideoOCX ซึ่งสามารถทำการรับภาพและบันทึกภาพได้

3.7.2 การหาหมายเลขทะเบียนรถยนต์จากภาพ

การแยกหมายเลขทะเบียนรถยนต์ออกมาจากภาพมีขั้นตอนดังต่อไปนี้คือ ในขั้นแรกจะเป็นการแปลงภาพที่รับเข้ามาเป็นภาพเกรย์สเกล (Gray Scale) 256 ระดับ จากนั้นนำมาหาขอบภาพ (Edge Detection) โดยใช้วิธีเกรเดียนต์ (Gradient Operator) โดยใช้หน้ากาก (Mask) แบบ Truncated Pyramid และทำการกำจัดจุดที่เป็นสัญญาณรบกวนออกไป เมื่อผ่านขั้นตอนนี้แล้วก็จะได้ภาพที่มีสีดำและขอบวัตถุที่มีสีขาว แล้วจึงนำมาหาส่วนที่น่าจะเป็นทะเบียนรถยนต์ โดยการวิเคราะห์จุดสีขาวที่มีการเรียงตัวในแนวแกน X และ แกน Y ที่มีความยาวในระดับที่น่าจะเป็นทะเบียนรถยนต์ได้ จากนั้นจะทำการจดจำตำแหน่งและไปดึงภาพมาจากต้นฉบับและทำการเปลี่ยนภาพต้นฉบับเป็นภาพไบนารี และทำการแยกแยะขอบเขตของตัวอักษรในป้ายทะเบียนต่อไปโดยใช้วิธีพิจารณาการเปลี่ยนของค่าจุดภาพจากขาวไปดำ หรือ ดำไปขาว โดยการดูฮิสโตแกรมของภาพในแต่ละแนวแกน เมื่อได้ขอบเขตของตัวอักขระที่แน่นอนแล้วก็นำภาพตัวอักขระมาทำการแยกแยะว่าเป็นตัวอักขระใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

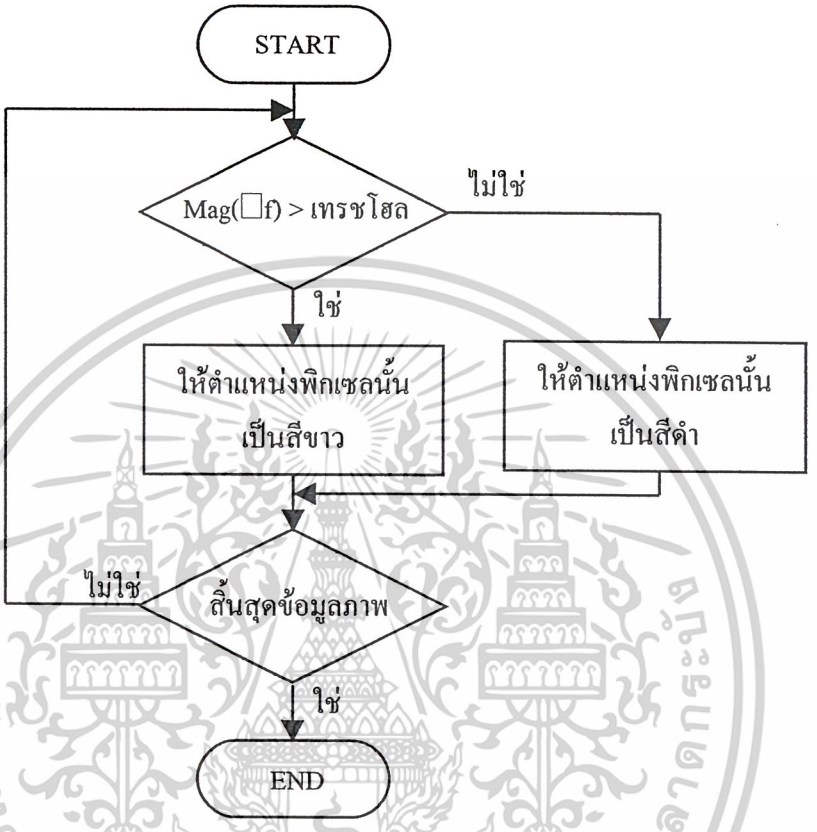


รูปที่ 3.24 โฟล์ชาร์ตการหาหมายเลขทะเบียนรถยนต์จากภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2.1 การทำไบนาไรเซชัน

เมื่อผ่านการทำการหาขอบภาพแล้ว จะนำภาพที่ได้มาแปลงเป็นภาพขาวดำ เพื่อให้ง่ายต่อการตรวจสอบขอบเขตของป้ายทะเบียน โดยกำหนดค่าเทรชโฮลที่เหมาะสม ในส่วนโปรแกรมแสดงได้ดังรูปที่ 3.25

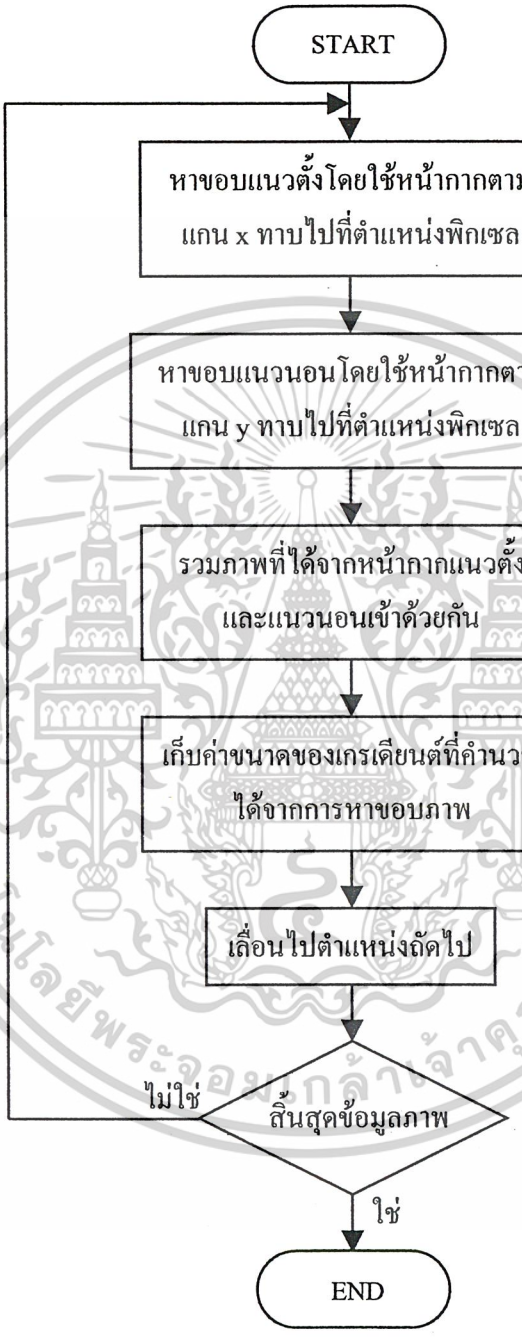


รูปที่ 3.25 โฟลว์ชาร์ตการทำไบนาไรเซชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2.2 การตรวจหาขอบภาพ

เพื่อให้ง่ายต่อการหาขอบของป้ายทะเบียน จึงต้องมีการหาขอบของภาพโดยรวมทั้งหมดก่อน ซึ่งจะใช้วิธีเกรเดียนต์ หน้ากากแบบ Truncated Pyramid

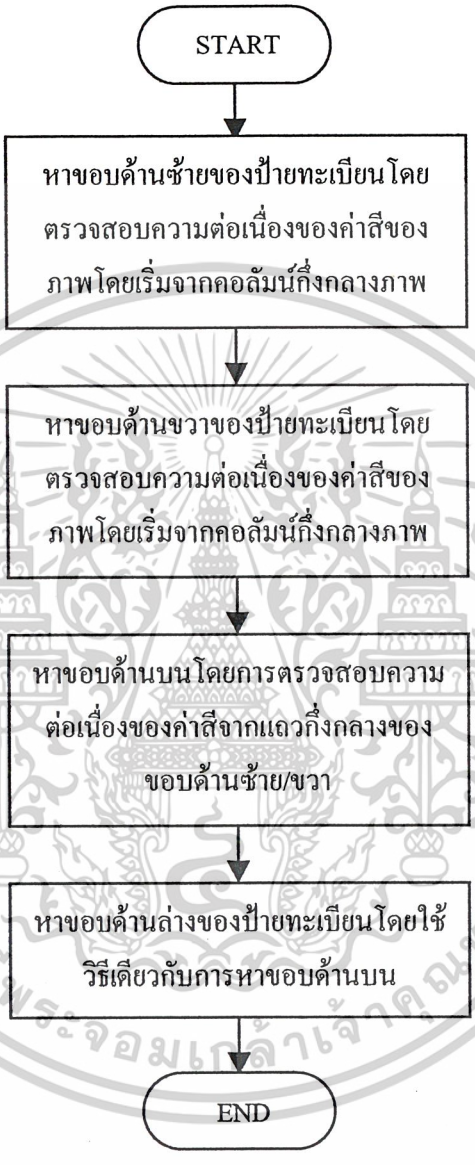


รูปที่ 3.26 โฟลว์ชาร์ตการตรวจหาขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2.3 การหาตำแหน่งป้ายทะเบียน

ในการหาตำแหน่งของป้ายทะเบียน จะนำภาพขาวดำที่ได้มาหาขอบเขตของป้ายทะเบียนโดยการหาตำแหน่งสีขาวที่มีลักษณะต่อเนื่องกันอยู่ในช่วงที่น่าจะเป็นป้ายทะเบียนรถยนต์ได้ ดังแสดงในโปรแกรมดังรูปที่ 3.27



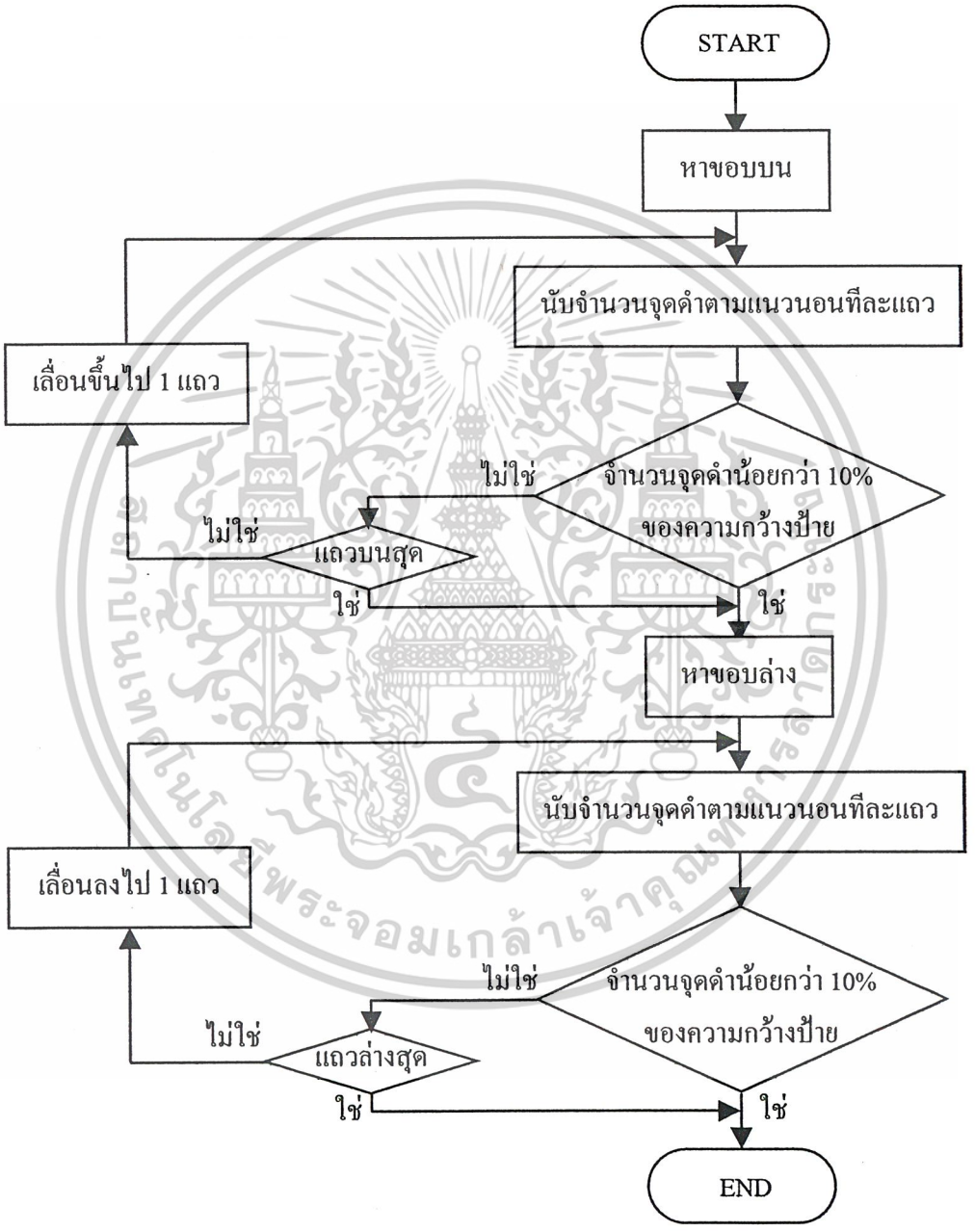
รูปที่ 3.27 โฟลว์ชาร์ตการหาตำแหน่งป้ายทะเบียน

เมื่อได้ภาพที่เหลือแต่ส่วนของทะเบียนรถยนต์ ก็จะนำภาพนี้มาตรวจหาตัวอักษรต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2.4 การหาตำแหน่งหมายเลขทะเบียน

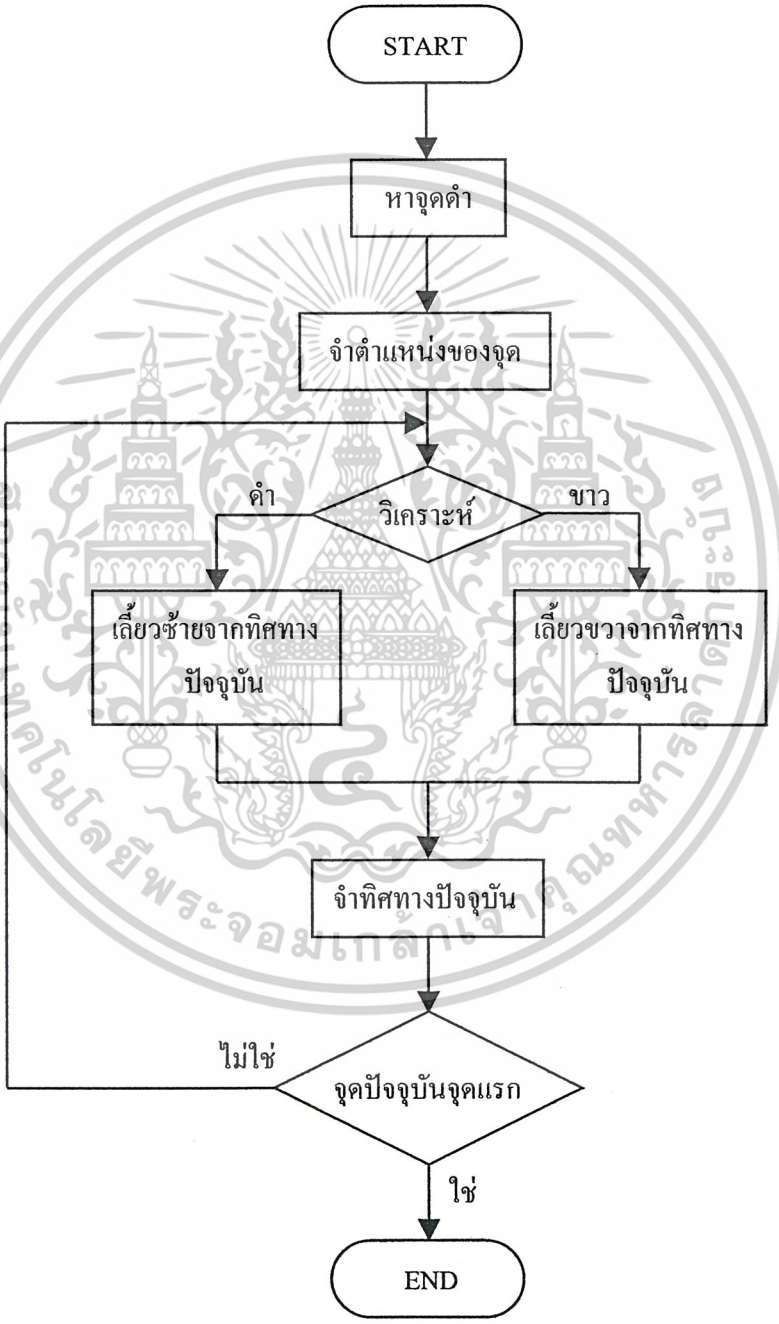
เมื่อได้ภาพป้ายทะเบียนแล้ว จะนำภาพมาแยกขอบเขตของหมายเลขทะเบียน ในที่นี้จะใช้การนับจำนวนจุดดำในแต่ละแถวของป้ายทะเบียนแล้วพิจารณาจำนวนจุดดำในภาพเพื่อหาขอบเขตบนและขอบเขตล่างของหมายเลขทะเบียน (ให้ตัวอักขระทุกตัวมีขอบเขตบนและล่างเป็นค่าเดียวกัน) ดังแสดงในโฟลว์ชาร์ตในรูปที่ 3.28



รูปที่ 3.28 โฟลว์ชาร์ตการหาขอบเขตบนและล่างของหมายเลขทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการ การแยกตัวอักขระด้วยวิธีการตามรอยขอบภาพโดยมีขั้นตอนการทำงานคือขั้นแรกจะทำการหาตำแหน่งที่มีการเปลี่ยนค่าสีโดยเริ่มจากทางด้านซ้ายสุดของวัตถุ และเมื่อเจอตำแหน่งเริ่มต้นแล้วก็จะทำการวนไปตามขอบของวัตถุหรือตัวอักขระไปเลื่อยๆจนกว่าจะวนกลับมาเจอตำแหน่งเริ่มต้นแล้วจึงหยุด และเมื่อทำครบทุกตัวแล้วก็จะนำค่าตำแหน่งที่วิ่งผ่านไปทั้งหมดมาหาค่ามากสุดน้อยสุดในแต่ละแกน ก็จะได้ค่าตำแหน่งของอักขระที่ถูกต้อง โดยมีโฟลว์ชาร์ตดังแสดงในรูปที่ 3.29



รูปที่ 3.29 โฟลว์ชาร์ตการแยกตัวอักขระจากภาพ

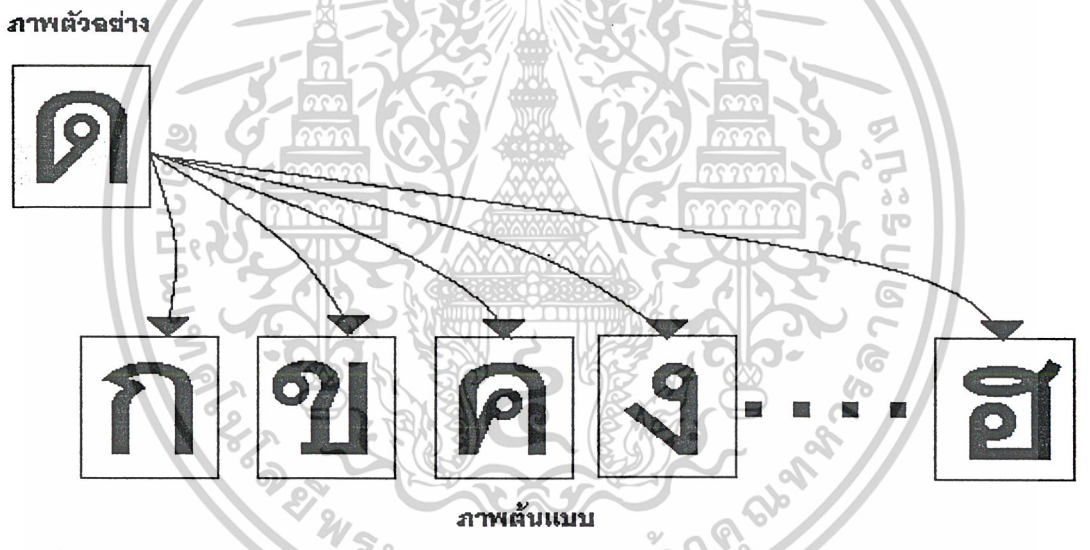
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การรู้จำตัวอักขระ

ในการรู้จำตัวตัวอักขระจากข้อมูลถาพนั้นมีวิธีที่ใช้ได้มากมายหลายวิธี เช่น การใช้เทคโนโลยีโครงข่ายนิวรอล (Neural Network), การเทียบค่าจุดภาพ (Matching), การตัดและจำแนก(Cut and Classification) ซึ่งแต่ละวิธีก็มีข้อดีข้อเสียแตกต่างกันออกไปแล้วแต่วัตถุประสงค์ของการใช้งาน ซึ่งในที่นี้จะเลือกใช้วิธีการตัดและจำแนก(Cut and Classification) แต่ละวิธีมีดังต่อไปนี้

3.8.1 การเทียบค่าจุดภาพ(Matching)

เป็นการจำแนกตัวอักขระ โดยการนำภาพตัวอักขระซึ่งเป็นภาพแบบขาวดำมาเปรียบเทียบกับภาพต้นแบบที่ละจุด จนครบทั้งภาพแล้วนับจุดภาพที่เหมือนกับต้นแบบ โดยจะต้องมีภาพต้นแบบเป็นภาพตัวอักขระที่มีลักษณะชัดเจนครบทุกตัว เมื่อภาพตัวอักขระนั้นเหมือนต้นแบบตัวใดมากที่สุด หรือมีจำนวนจุดภาพที่เหมือนกันมากที่สุดก็แสดงว่าตัวอักขระนั้นคือตัวต้นแบบตัวนั้น โดยภาพตัวอย่างและภาพต้นแบบจะต้องมีขนาดที่เท่ากัน จึงจะทำการเปรียบเทียบจุดภาพได้อย่างถูกต้อง



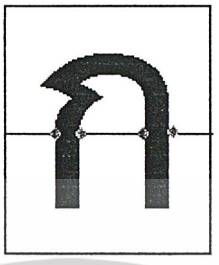
รูปที่ 3.30 การเทียบค่าจุดภาพ

ข้อจำกัดของการรู้จำตัวอักขระแบบนี้คือ ถ้าหากข้อมูลภาพของตัวอักขระเอียงก็จะทำให้การรู้จำตัวอักขระมีความผิดพลาดได้ อีกทั้งภาพต้นแบบจะต้องมีความชัดเจนมากและมีความผิดพลาดของข้อมูลรูปภาพน้อยจึงจะทำให้ผลการเปรียบเทียบอักขระออกมาได้อย่างถูกต้อง และวิธีนี้ยังอาจเกิดข้อผิดพลาดจากการที่รูปภาพต้นแบบกับภาพตัวอย่างจะต้องมีขนาดเท่ากัน เพราะฉะนั้นเวลาที่นำกระบวนการแบบนี้มาใช้จริงจะต้องนำภาพตัวอย่างมาทำการขยายขนาด หรือ นำมาทำการลดขนาดของภาพให้เท่ากับขนาดของภาพต้นแบบ ซึ่งการทำแบบนี้อาจส่งผลให้เกิดการผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.2 การตัดและจำแนก (Cut and Classification)

หลักการของการตัดคือการตรวจสอบค่าสีไปตามแนวแถวหรือหลักเพื่อหาจุดขอบหรือจุดบ่งชี้ที่มีการเปลี่ยนแปลงจากขาวเป็นดำ และดำเป็นขาวหลังจากนั้นก็จะทำการนับจำนวนของตำแหน่งที่เกิดการเปลี่ยนแปลงของค่าจุดสีตามแนวต่างที่ได้กำหนดไว้



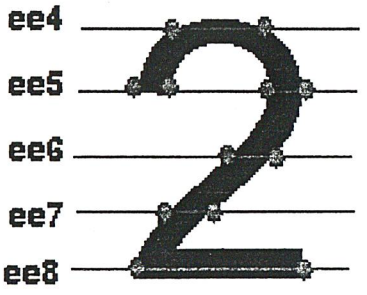
รูปที่ 3.31 ตัวอย่างการหาจุดบ่งชี้ตามแนวนอน

รูปที่ 3.32 ตัวอย่างการหาจุดบ่งชี้ตามแนวตั้ง

โดยในที่นี้จะใช้การตัดทั้งหมด 8 แถว โดยจะแบ่งออกเป็นแนวนอน 5 แถวและแนวตั้ง อีก3แถว โดยใช้ตำแหน่งในการตัดดังรูป 3.33 และ 3.34

รูปที่ 3.33 การหาจุดบ่งชี้ตามแนวตั้ง 3 แถวที่ใช้ในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

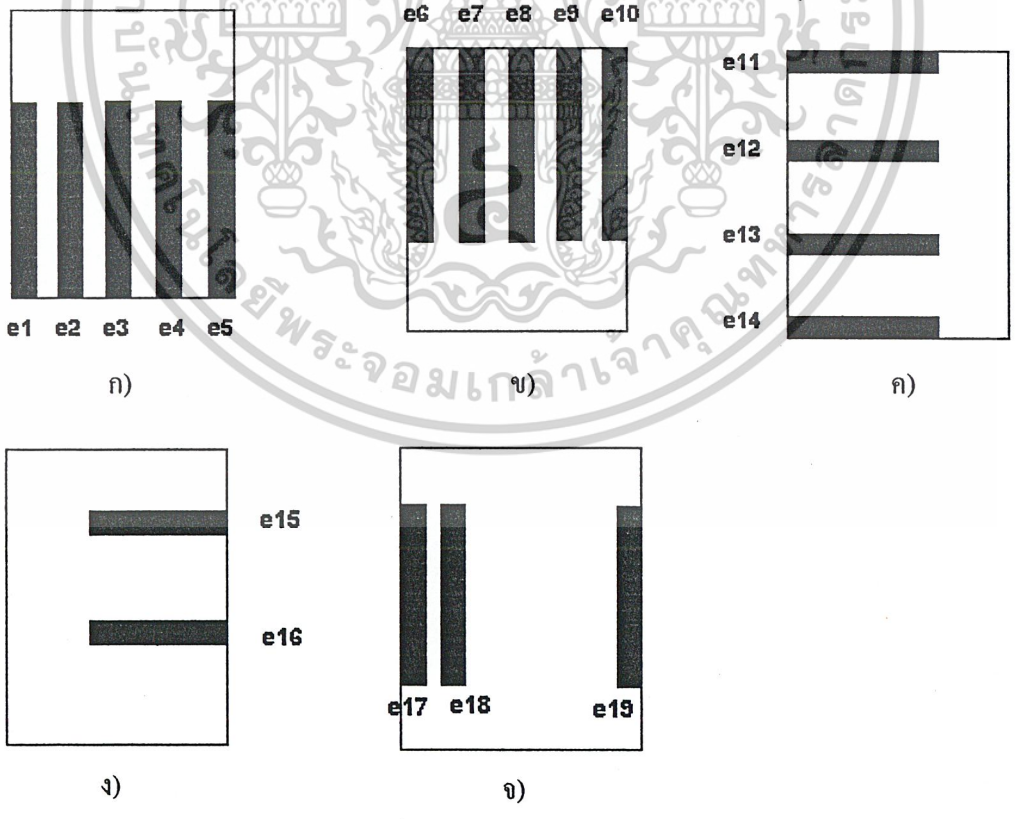


รูปที่ 3.34 การหาจุดบ่งชี้ตามแนวนอน 5 แถวที่ใช้ในโปรแกรม

จากรูป 3.33 และ 3.34 จะได้ค่าต่างๆออกมาดังนี้ ee1= 4 , ee2=6 , ee3=4 , ee4=2 , ee5=4 , ee6=2 , ee7=2 , ee8=2

3.8.3 การหาตำแหน่งของเส้นสีขาว

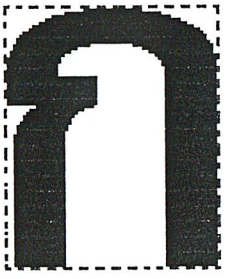
เป็นรูปแบบในการหาลักษณะเด่นของตัวอักษรแต่ละตัวว่ามีความแตกต่างกันอย่างไร โดยใช้การหาตำแหน่งของเส้นสีขาวที่ปรากฏอยู่นะตำแหน่งต่างๆของตัวอักษร โดยในที่นี้จะใช้ตำแหน่งในการหาเส้นสีขาวทั้งหมด 19 ตำแหน่งดังรูป 3.35



รูปที่ 3.35 การหาเส้นสีขาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเมื่อนำตัวอักษร "ก" มาทำการหาตำแหน่งเส้นสีขาวทั้ง 19 ตำแหน่งหากตำแหน่งนั้นมีเส้นสีขาวอยู่จะให้ค่า ณ ตำแหน่งนั้นเท่ากับ 1 แต่ถ้าไม่มีเส้นสีขาวจะให้ค่า ณ ตำแหน่งนั้นเท่ากับ 0



รูปที่ 3.36 ตัวอย่างอักษร "ก"

ตัวอย่างจากรูป 3.36 จะได้ค่าต่างๆ ออกมาดังนี้

1) e1 = 0	2) e2 = 1	3) e3 = 1	4) e4 = 1
5) e5 = 0	6) e6 = 0	7) e7 = 0	8) e8 = 0
9) e9 = 0	10) e10 = 0	11) e11 = 0	12) e12 = 0
13) e13= 0	14) e14 =0	15) e15 = 0	16) e16 = 0
17) e17 =0	18) e18 =0	19) e19 = 0	

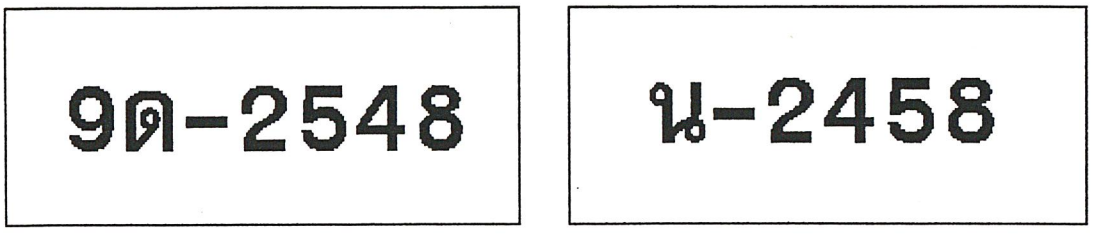
และในการจำแนกตัวอักขระก็จะนำค่าที่ได้จากการตัดและจำแนก และ การหาตำแหน่งของเส้นสีขาว มารวมกันเพื่อใช้เป็นรูปแบบในการแยกตัวอักขระต่างๆออกจากกัน ซึ่งรูปแบบนี้จะให้ผลที่ดีกว่ารูปแบบอื่นและมีความรวดเร็วในการทำงานของโปรแกรมเพราะจะพิจารณาเพียงแถวหรือหลักที่ต้องการเท่านั้น แต่ในทางตรงกันข้ามระบบนี้ก็มีข้อเสียคือ เมื่อภาพของตัวอักษรมีจุดที่เป็นสัญญาณรบกวนเกิดขึ้นจากการแปลงภาพสีเป็นขาวดำ หรือตัวอักขระเกิดการเอียงก็จะทำให้ผลที่ได้ออกมาเกิดความผิดพลาดได้

3.8.4 ลักษณะของป้ายทะเบียนรถยนต์

ป้ายทะเบียนรถยนต์ในประเทศไทยนั้น จะสามารถแบ่งออกได้เป็น 2 แบบ ตามลักษณะของตัวอักขระในป้ายทะเบียน ได้แก่

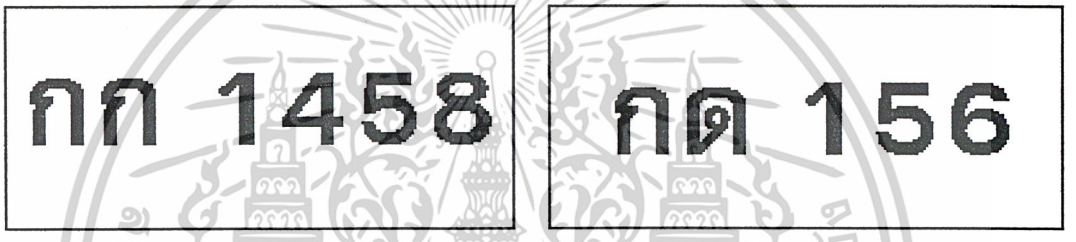
1. ป้ายทะเบียนแบบเก่า มีรูปแบบคือ ตัวอักขระตัวแรกเป็นตัวเลข (อาจมีหรือไม่มีก็ได้) และตัวต่อมาจะเป็นตัวอักษร 1 ตัว และตามด้วยตัวเลขอีก 4 ตัวดังรูป 3.37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.37 ป้ายทะเบียนรถยนต์แบบเก่า

2. ป้ายทะเบียนแบบใหม่มี รูปแบบคือ ตัวพยัญชนะ 2 ตัวแล้วตามด้วยตัวเลขอีก 1 ถึง 4 ตัว ดังรูป 3.38



รูปที่ 3.38 ป้ายทะเบียนรถยนต์แบบใหม่

และในการรู้จำตัวอักขระในป้ายทะเบียนรถยนต์นั้นอาจจะแบ่งแยกออกเป็นสองส่วน ส่วนแรกคือส่วนตัวเลขและพยัญชนะ และส่วนที่สองคือส่วนของตัวเลขอย่างเดียว ดังรูป 3.39

รูปที่ 3.39 การแบ่งส่วนของตัวอักขระในป้ายทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแยกพิจารณาเป็นส่วนๆนี้จะทำให้ลดความผิดพลาดในการวิเคราะห์ตัวอักขระได้ในส่วนที่สอง เนื่องจากตัวเลขมีเพียง 10 ตัว จึงพิจารณาเพียงกรณีของตัวเลขซึ่งก็คือ 10 กรณีเท่านั้น จึงทำให้ผลออกมาถูกต้องมากขึ้น

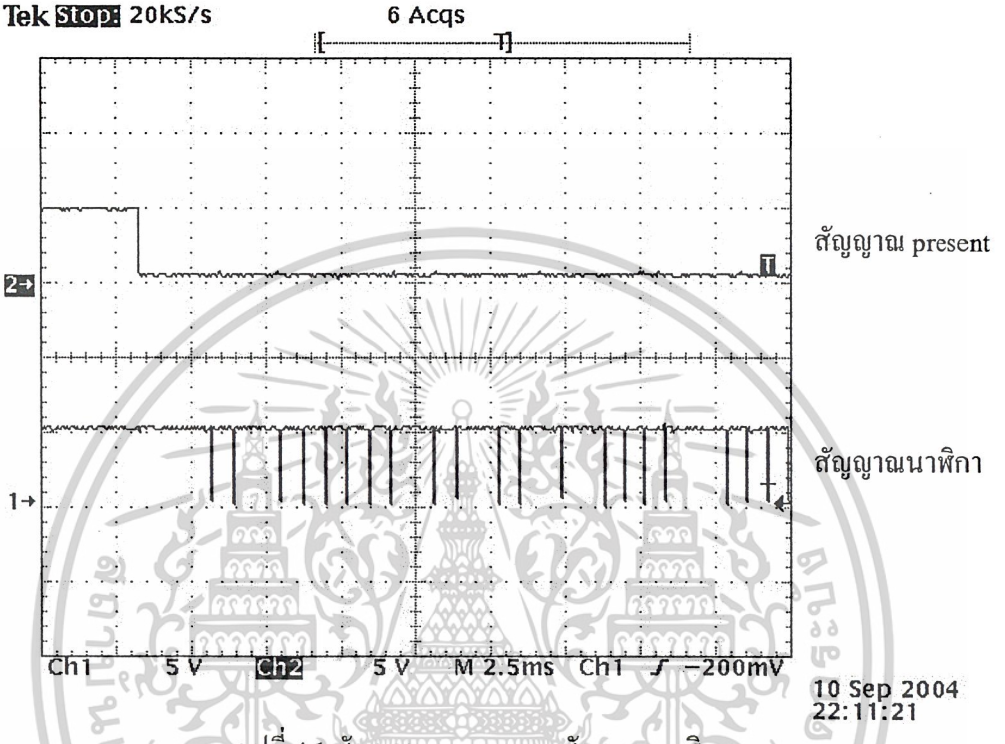
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

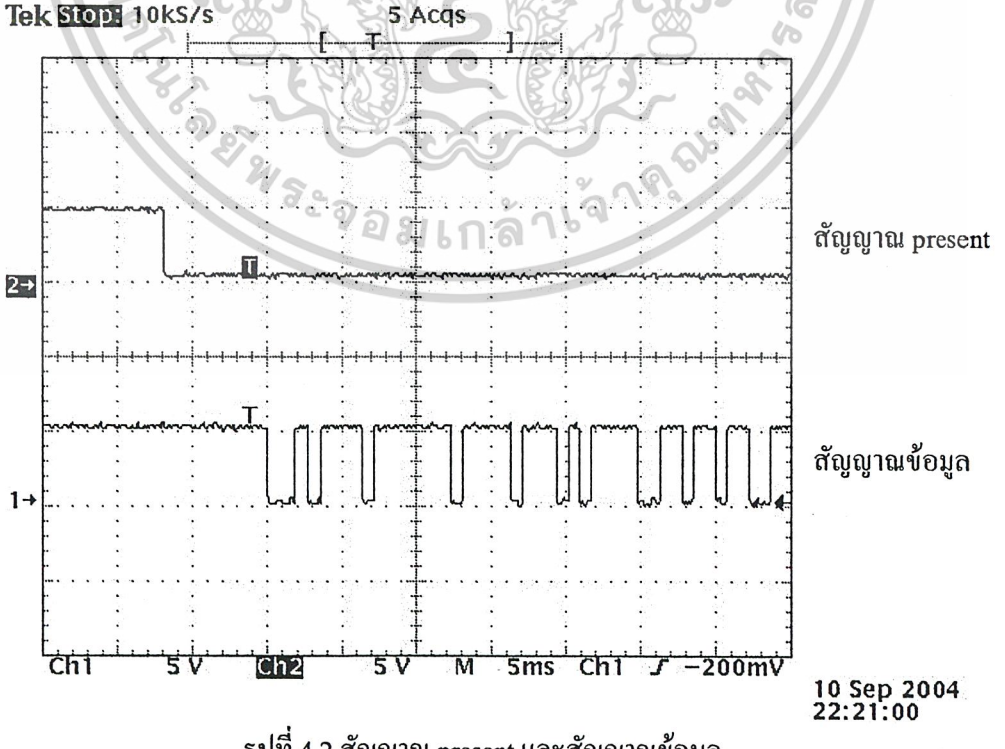
การทดลองและผลการทดลอง

4.1 เครื่องอ่านบัตรแม่เหล็ก

เมื่อทำการรูดบัตรผ่านหัวอ่านของเครื่องอ่านบัตรแม่เหล็กจะสัญญาณ present จะมีสถานะเป็น low และจะได้สัญญาณนาฬิกา กับสัญญาณข้อมูล ดังรูป 4.1 และ 4.2



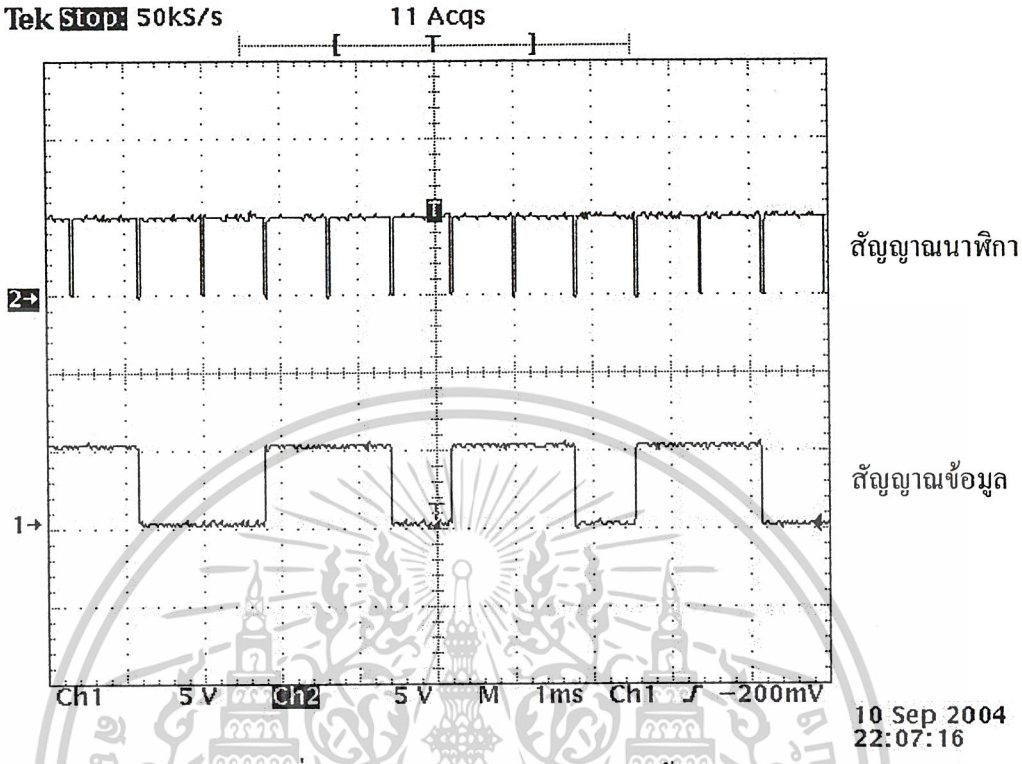
รูปที่ 4.1 สัญญาณ present และสัญญาณนาฬิกา



รูปที่ 4.2 สัญญาณ present และสัญญาณข้อมูล

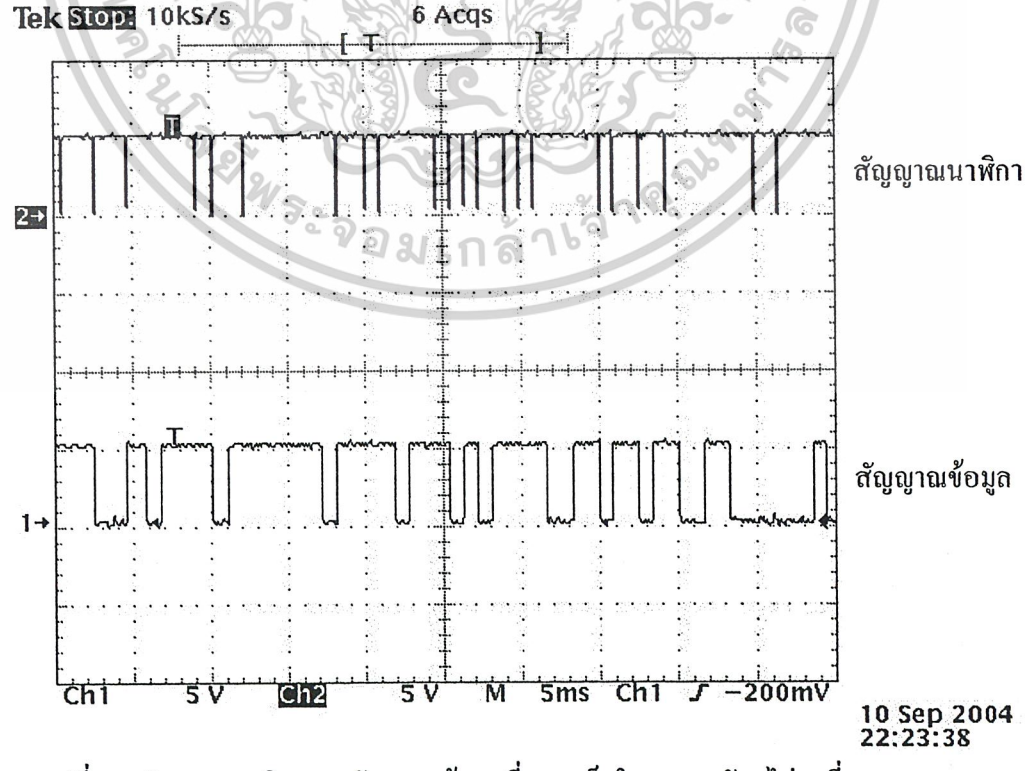
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขณะที่ทำการอ่านบัตรหากต้องการอ่านข้อมูลต้องอ่านข้อมูลในช่วงที่เป็นขอบขาลงของสัญญาณนาฬิกาดังรูป 4.3



รูปที่ 4.3 สัญญาณนาฬิกาและสัญญาณข้อมูล

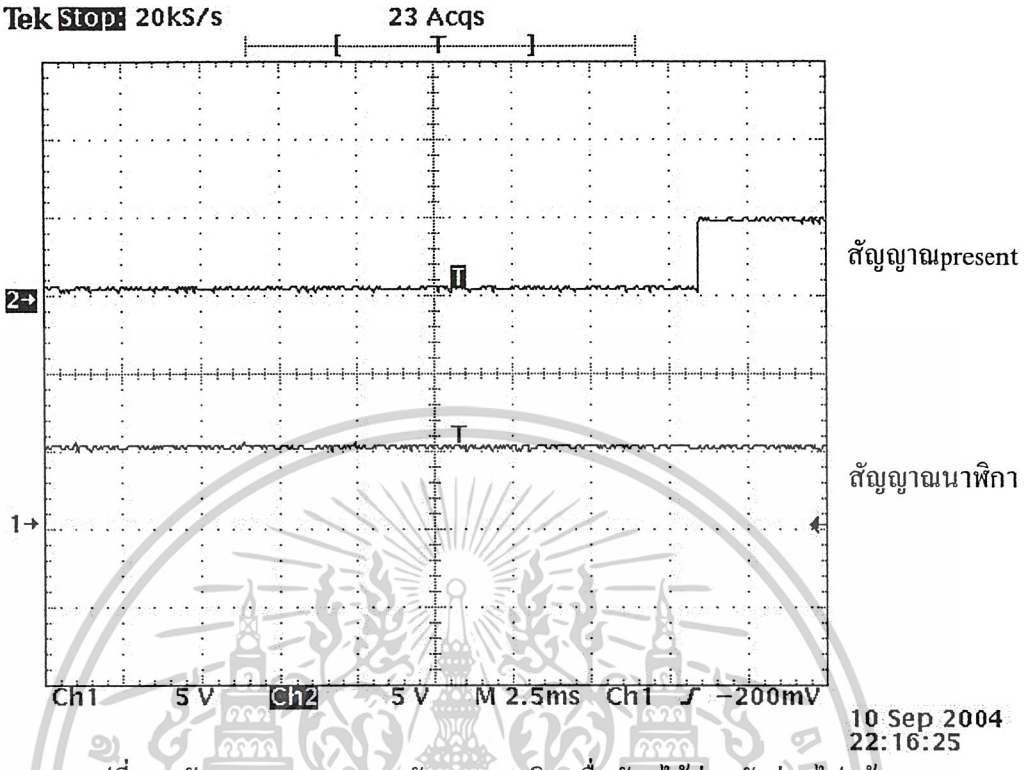
ในบางครั้งการรูดบัตรผ่านหัวอ่านอาจมีความเร็วไม่คงที่ซึ่งจะทำให้ได้สัญญาณนาฬิกาที่มีความถี่ไม่คงที่ดังรูป 3.4



รูปที่ 4.4 สัญญาณนาฬิกาและสัญญาณข้อมูลที่ความเร็วในการรูดบัตรไม่คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

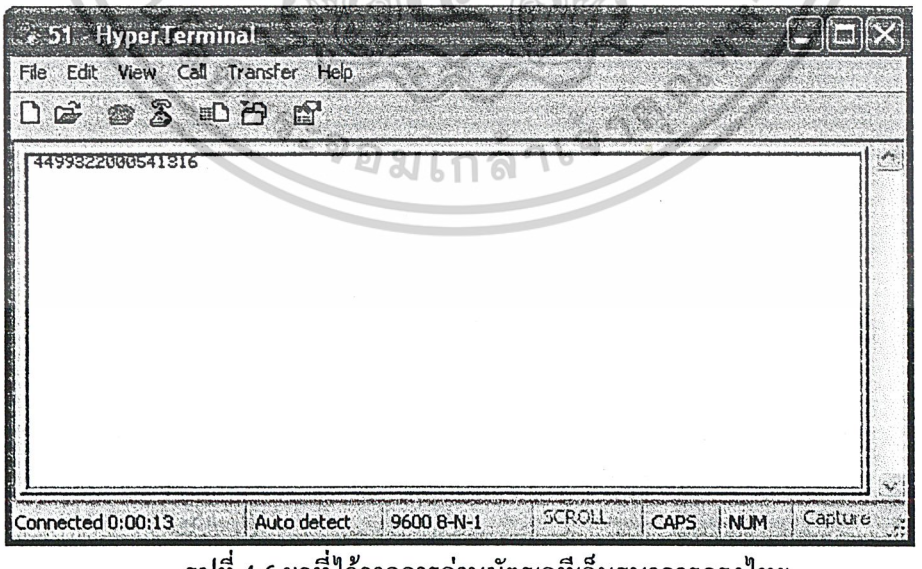
เมื่อบัตรได้ผ่านออกจากหัวอ่านไปแล้ว สัญญาณนาฬิกาจะหมดไป และสัญญาณ present จะมีสถานะกลับมาเป็น high



รูปที่ 4.5 สัญญาณ present และสัญญาณนาฬิกาเมื่อบัตรได้ผ่านหัวอ่านไปแล้ว

4.2 การส่งข้อมูลจากการอ่านบัตรแม่เหล็กเข้าคอมพิวเตอร์

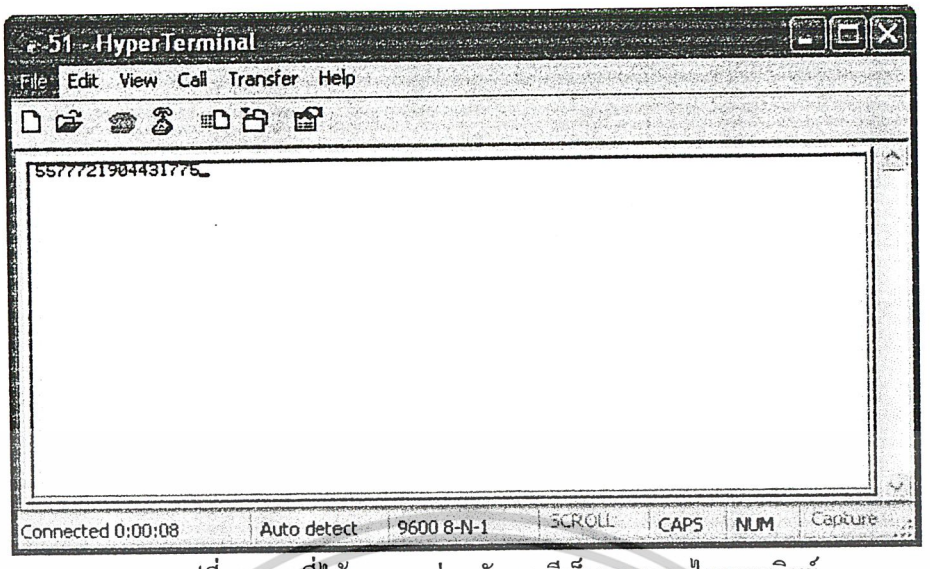
ในการทดสอบการส่งข้อมูลเข้าคอมพิวเตอร์จะใช้โปรแกรม Hyper Terminal ในการทดสอบ ซึ่งผลที่ได้จะเป็นข้อมูลของบัตรที่อ่านได้



รูปที่ 4.6 ผลที่ได้จากการอ่านบัตรเอทีเอ็มธนาคารกรุงไทย

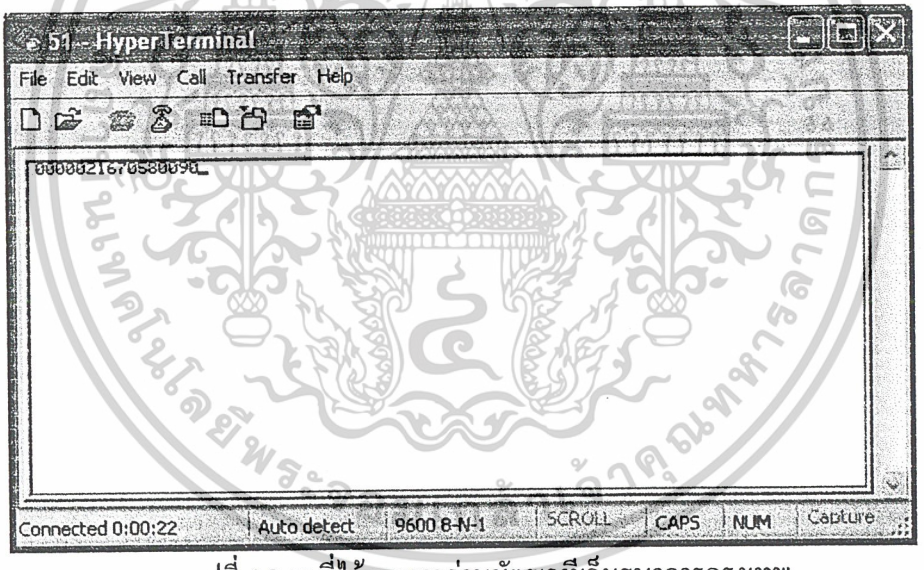
ข้อมูลที่อ่านได้จากบัตรเอทีเอ็มธนาคารกรุงไทยจะตรงกับหมายเลขที่ระบุอยู่บนบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ผลที่ได้จากการอ่านบัตรเอทีเอ็มธนาคารไทยพาณิชย์

ข้อมูลที่อ่านได้จากบัตรเอทีเอ็มธนาคารไทยพาณิชย์จะตรงกับหมายเลขที่ระบุอยู่บนบัตร



รูปที่ 4.8 ผลที่ได้จากการอ่านบัตรเอทีเอ็มธนาคารกรุงเทพ

ข้อมูลที่อ่านได้จากบัตรเอทีเอ็มธนาคารกรุงเทพจะไม่ตรงกับหมายเลขที่ระบุอยู่บนบัตรแต่มีบางส่วนตรงกับหมายเลขบัญชี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

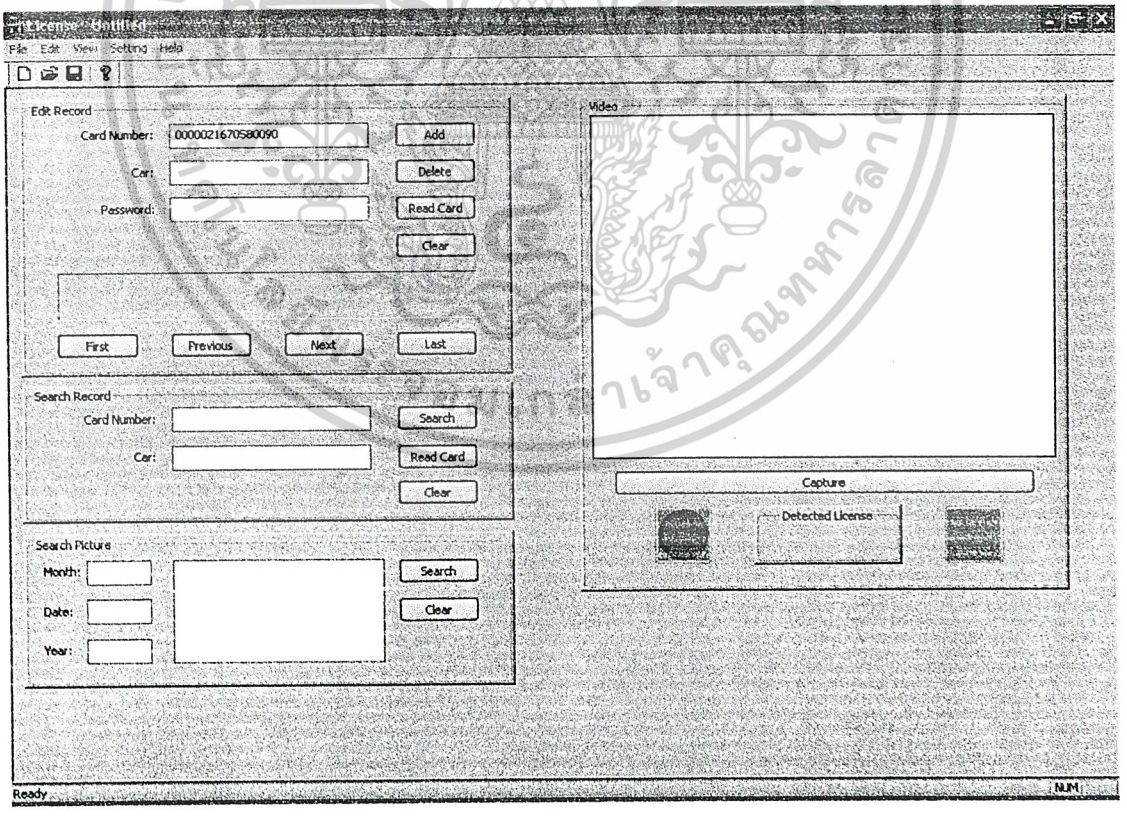
4.3 ฐานข้อมูล

ก่อนที่จะเข้าโปรแกรมเพื่อทำการแก้ไขหรือบันทึกข้อมูล โปรแกรมจะร้องขอให้มีการรูดบัตร โดยบัตรที่ใช้รูดนั้นต้องเป็นบัตรที่มีหมายเลขเฉพาะที่ได้มีการตั้งไว้เท่านั้นซึ่งผู้ควบคุมระบบจะเป็นผู้ถือบัตรใบนั้น ดังแสดงในรูป 4.9 ก) และหากบัตรที่ใช้นั้นถูกต้องก็จะสามารถเข้าไปทำการแก้ไขหรือบันทึกข้อมูลได้ แต่หากบัตรที่ใช้ไม่ถูกต้องโปรแกรมจะแสดงผลดังรูป 4.9 ข)

ก) ข)

รูปที่ 4.9 โปรแกรมร้องขอให้รูดบัตรเฉพาะและผลลัพธ์หากไม่ใช้บัตรเฉพาะ

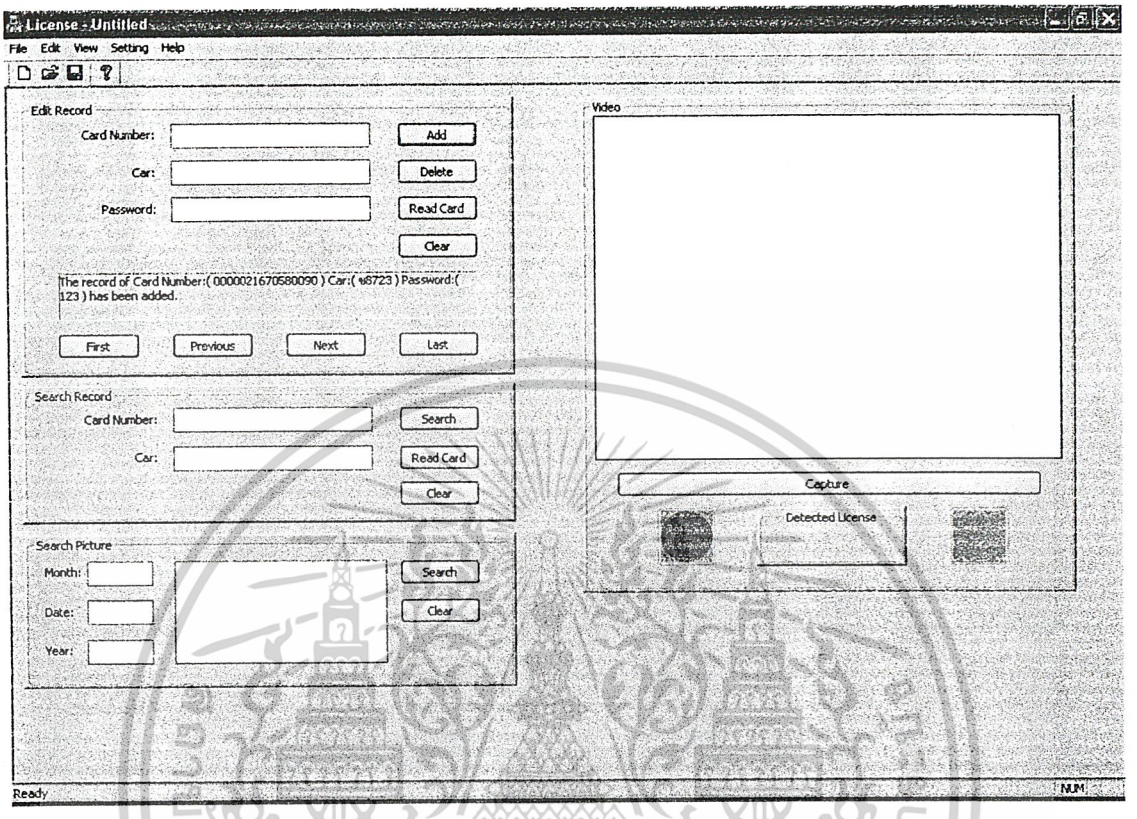
ในส่วนฐานข้อมูลสามารถเก็บข้อมูลได้ 2 ฟิลด์ คือฟิลด์หมายเลขบัตร และฟิลด์ทะเบียนรถยนต์ โดยในการกรอกข้อมูลหมายเลขบัตรสามารถทำได้ 2 วิธีคือ วิธีแรกผู้ใช้ทำการป้อนเองวิธีที่สองคือทำการกดปุ่ม Read Card จากนั้นผู้ใช้จึงทำการรูดบัตร ข้อมูลหมายเลขบัตรก็จะแสดงออกมาในช่อง Card Number ดังรูป 4.10



รูปที่ 4.10 ข้อมูลหมายเลขบัตรที่อ่านได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้ทำการป้อนข้อมูลหมายเลขบัตรและหมายเลขทะเบียนรถยนต์ในส่วนของ Edit Record เรียบร้อยแล้วและทำการเพิ่มเรคคอร์ด(Record)โดยการกดปุ่ม Add ผลลัพธ์ที่ได้แสดงดังรูป 4.11

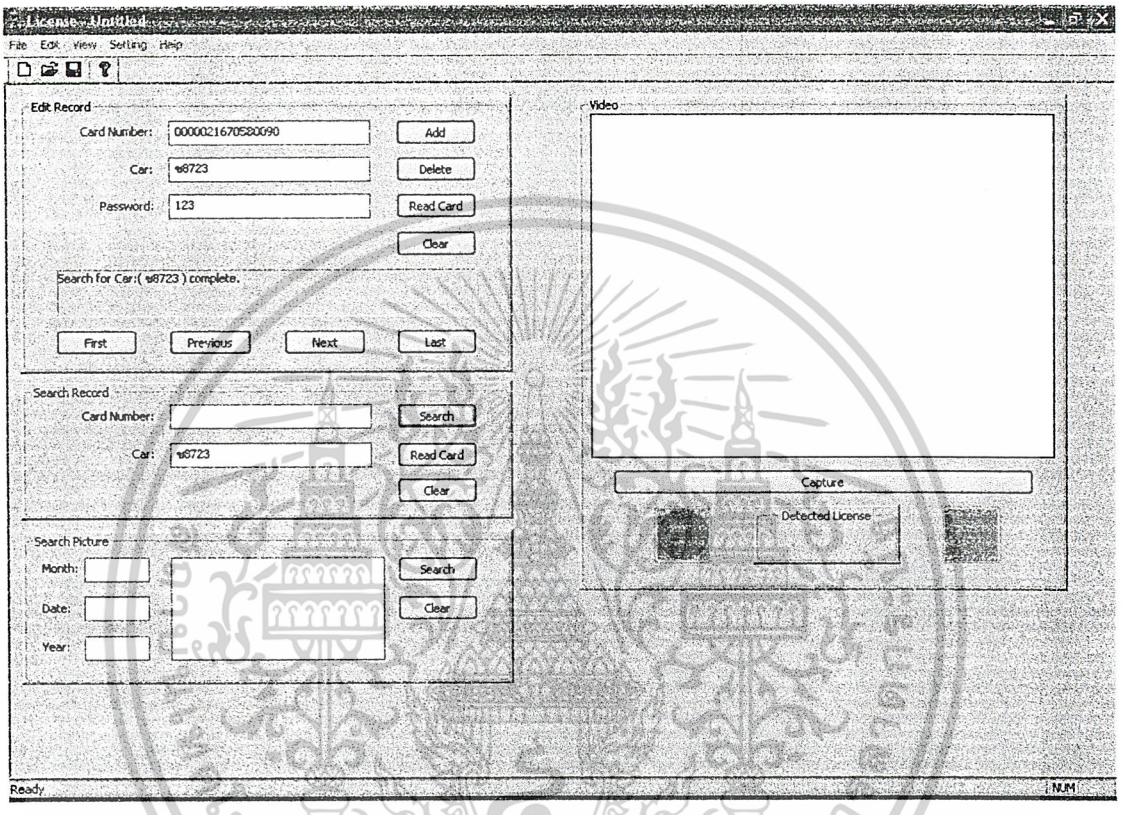


รูปที่ 4.11 ผลลัพธ์เมื่อโปรแกรมทำการเพิ่มข้อมูลสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการดูข้อมูลสามารถทำได้โดยกดปุ่ม First เพื่อดูเรคคอร์ดแรกสุด, ปุ่ม Last เพื่อดูเรคคอร์ดท้ายสุด, ปุ่ม Previous เพื่อดูเรคคอร์ดลำดับก่อนหน้า และปุ่ม Next เพื่อดูเรคคอร์ดลำดับถัดไป

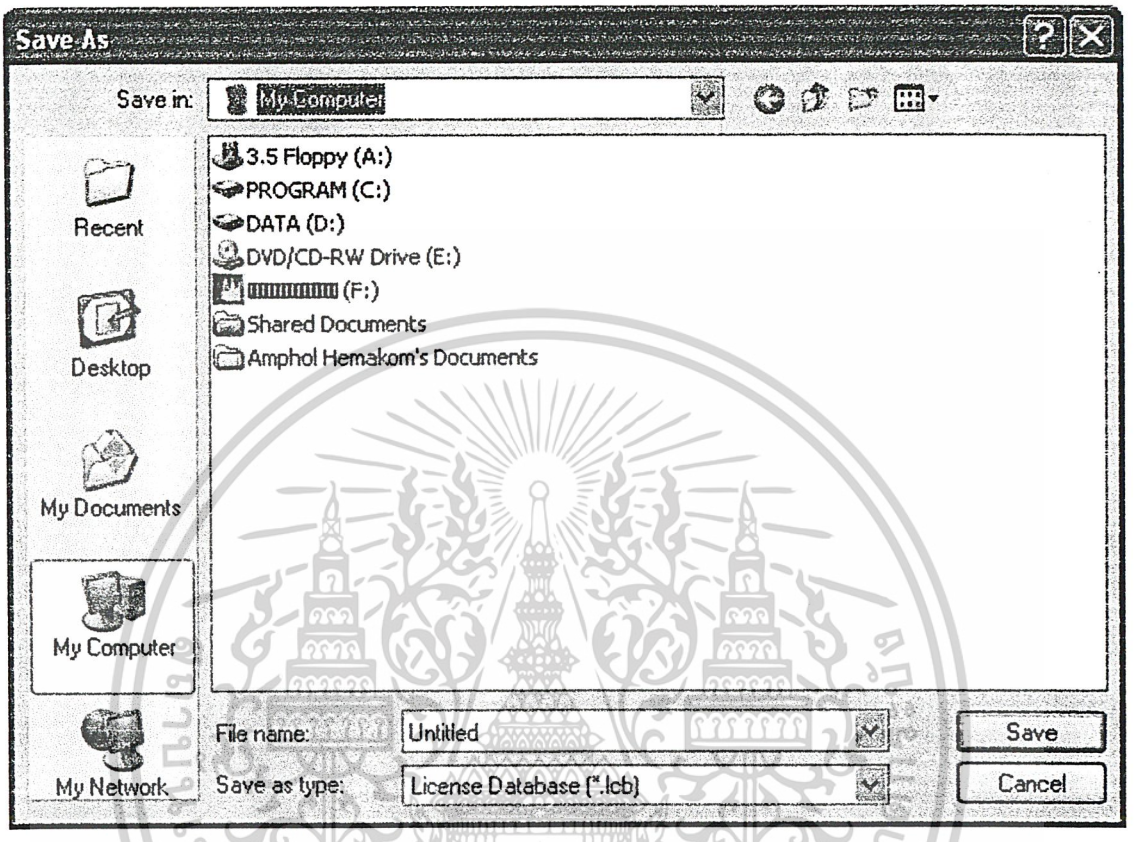
นอกจากนี้ผู้ใช้ยังสามารถค้นหาข้อมูลได้โดยการป้อนข้อมูลหมายเลขบัตรหรือหมายเลขทะเบียนรถยนต์อย่างใดอย่างหนึ่งในส่วนของ Search Record เพื่อค้นหาข้อมูลตามที่ต้องการได้ด้วย หากผลการค้นหาพบว่ามีข้อมูลที่ผู้ใช้ต้องการดูอยู่ในฐานข้อมูล ผลลัพธ์แสดงดังรูป 4.12



รูปที่ 4.12 ผลลัพธ์เมื่อการค้นหาข้อมูลพบข้อมูลที่ผู้ใช้ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้ต้องการบันทึกข้อมูลที่ได้เก็บลงในคอมพิวเตอร์ผู้ใช้สามารถทำได้โดยการกดปุ่ม Save และทำการใส่ชื่อของไฟล์ที่ต้องการลงไป โดยชนิดของไฟล์ที่ทำการบันทึกจะอยู่ในรูปแบบ lcb (license database file) ดังรูป 4.13

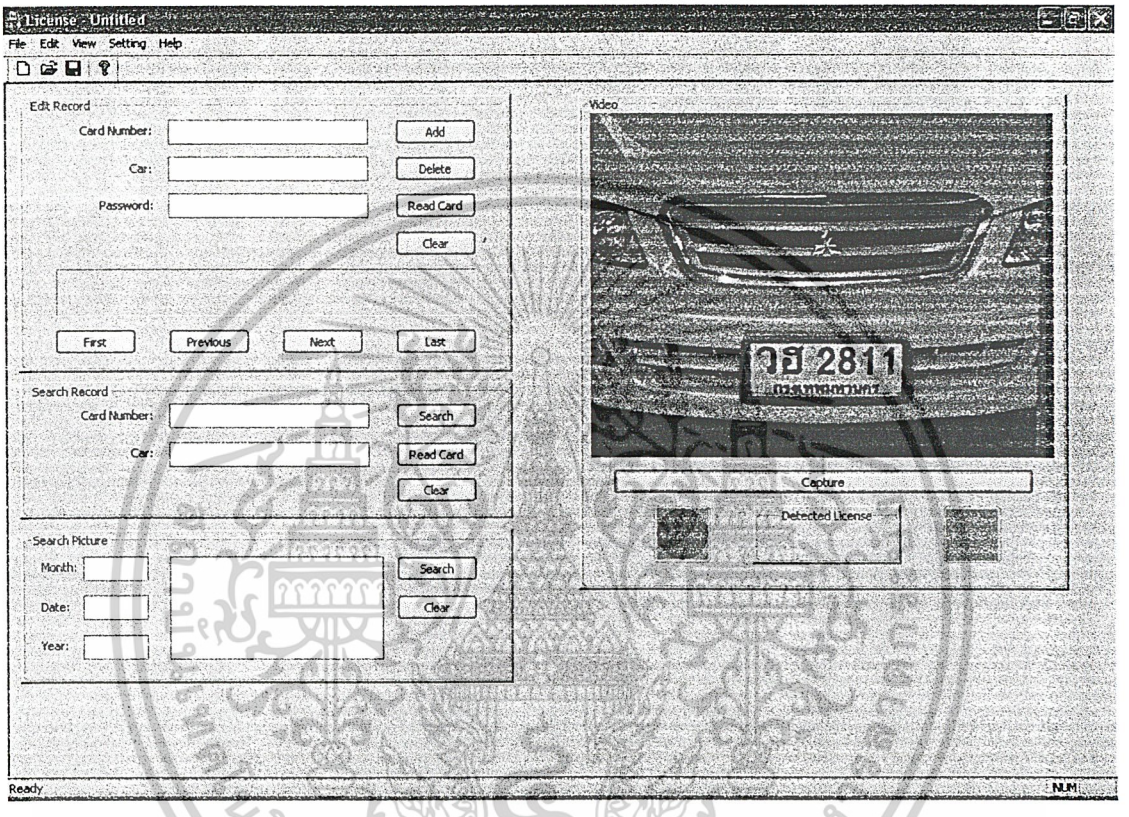


รูปที่ 4.13 การบันทึกข้อมูลลงไฟล์โดยไฟล์มีรูปแบบเป็น lcb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การรับภาพจากกล้องและการหาหมายเลขทะเบียนรถยนต์จากภาพ

โปรแกรมจะทำการรับภาพจากกล้องเมื่อผู้ควบคุมระบบกดปุ่ม Capture บนหน้าจอโปรแกรม และจะจับภาพเมื่อมีการรูดบัตรและจากนั้นจะทำการแยกหมายเลขทะเบียนรถยนต์ออกมาจากภาพ โดยในขณะที่ทำการรับภาพจากกล้องอยู่นั้นโปรแกรมจะถูกตั้งให้ไม่สามารถทำการแก้ไขหรือบันทึกข้อมูลในฐานข้อมูลได้ ซึ่งหากต้องการจะกลับมาทำการแก้ไขหรือบันทึกข้อมูลผู้ควบคุมระบบจะต้องใช้บัตรเฉพาะมารูดเพื่อให้โปรแกรมหยุดทำการรับภาพจากกล้อง



รูปที่ 4.14 หน้าจอโปรแกรมในขณะรับภาพจากกล้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อโปรแกรมทำการจับภาพได้แล้วโปรแกรมจะทำการแยกหมายเลขทะเบียนรถยนต์ออกมาจากภาพ

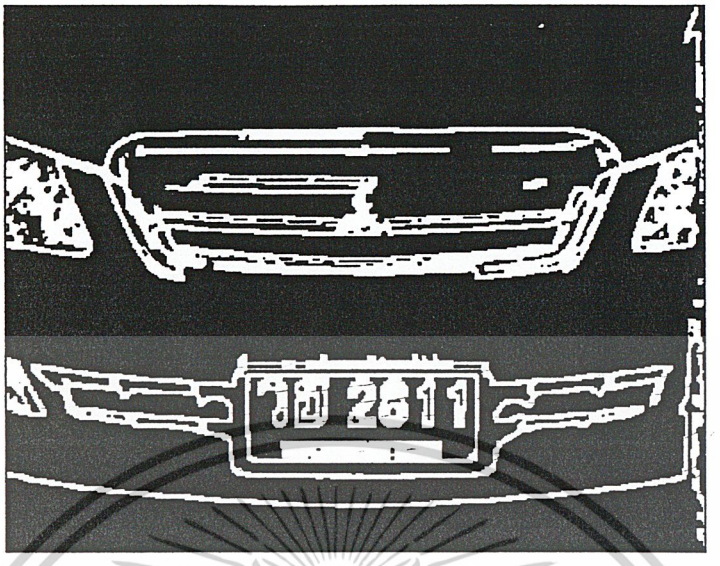
รูปที่ 4.15 ภาพต้นฉบับ

ภาพต้นฉบับที่ได้จะถูกนำมาแปลงเป็นภาพขาวดำ ผลลัพธ์แสดงดังรูป 4.16

รูปที่ 4.16 ภาพต้นฉบับเมื่อเปลี่ยนให้เป็นภาพไบนารี

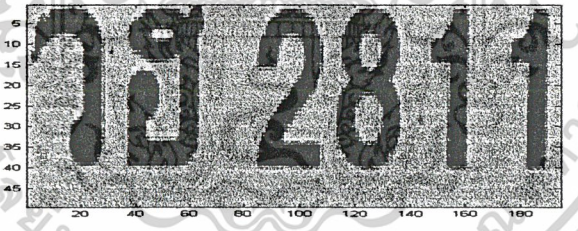
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพไบนารีที่ได้จะถูกนำมาหาขอบและกำจัดสัญญาณรบกวนผลลัพธ์แสดงดังรูป 4.17



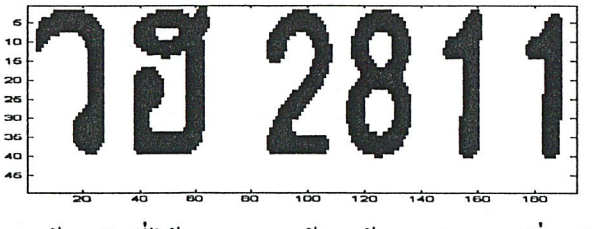
รูปที่ 4.17 ภาพที่ถูกหาขอบและกำจัดจุดที่เป็นสัญญาณรบกวน

โปรแกรมจะทำการหาขอบเขตของป้ายทะเบียนซึ่งจะทำการจดจำตำแหน่งและไปคัดลอกภาพ ณ ตำแหน่งเดียวกันมาจากภาพต้นฉบับ



รูปที่ 4.18 ภาพต้นฉบับที่ได้หาขอบของป้ายทะเบียนแล้ว

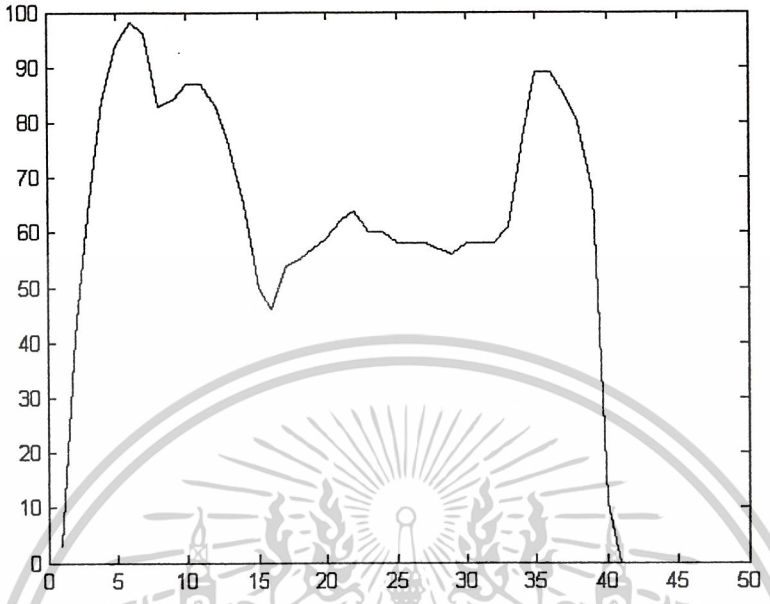
ภาพที่ถูกคัดลอกมาจะถูกเปลี่ยนให้เป็นภาพไบนารี ผลลัพธ์แสดงดังรูป 4.19



รูปที่ 4.19 ภาพต้นฉบับที่ได้หาขอบของป้ายแล้วและทำการเปลี่ยนเป็นภาพไบนารี

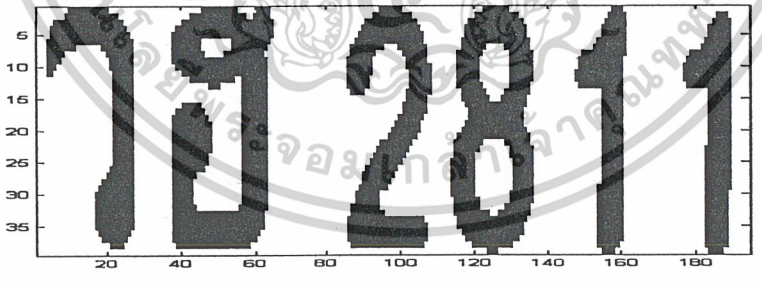
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นโปรแกรมจะทำการหาฮิสโตแกรมของภาพในแนวนอน ฮิสโตแกรมที่ได้แสดงดังรูป 4.20



รูปที่ 4.20 ฮิสโตแกรมของภาพในแนวนอน

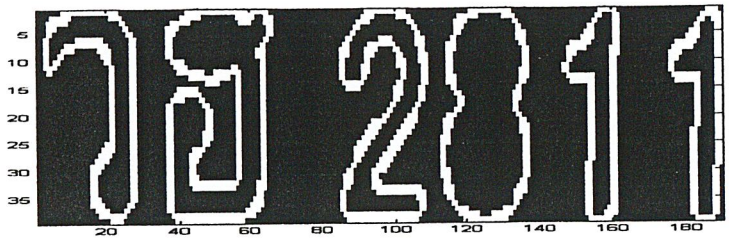
ฮิสโตแกรมของภาพในแนวนอนที่ได้จะถูกนำมาพิจารณาเพื่อหาขอบเขตของหมายเลขทะเบียนตามแนวนอน ภาพที่ได้หาขอบเขตของหมายเลขทะเบียนตามแนวนอนแล้ว แสดงดังรูป 4.21



รูปที่ 4.21 ภาพที่ถูกแยกออกตามแนวนอน

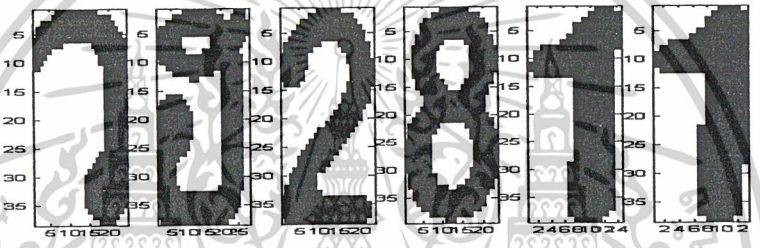
จากนั้นจะนำภาพที่ได้ผ่านการหาขอบเขตของหมายเลขทะเบียนมาทำการหาตำแหน่งของตัวอักษรโดยใช้กระบวนการวนรอบวัตถุ (Contour Following) ผลที่ได้แสดงดังรูป 4.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



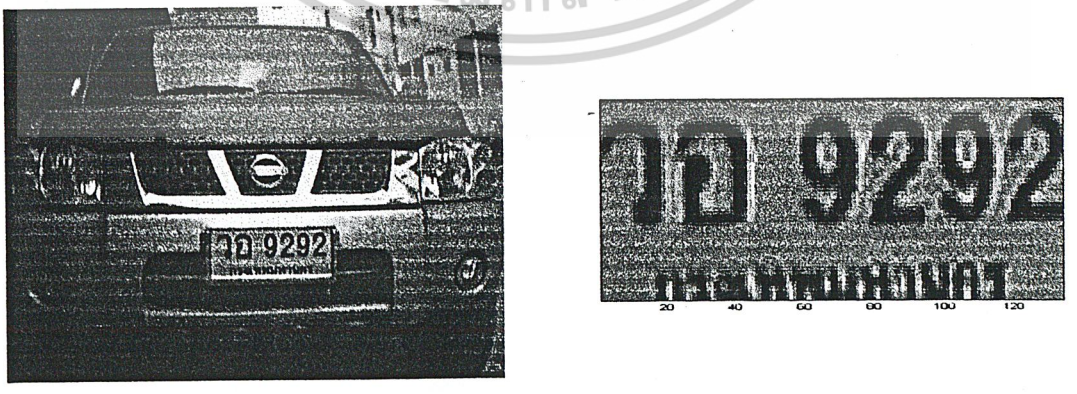
รูปที่ 4.22 การหาตำแหน่งของอักขระแต่ละตัว

ค่าตำแหน่งที่ถูกทำการวนรอบจะถูกนำมาพิจารณาเพื่อหาขอบเขตของอักขระแต่ละตัวเพื่อทำการแยกอักขระแต่ละตัวออกมา ผลลัพธ์แสดงดังรูป 4.23



รูปที่ 4.23 อักขระที่แยกได้

ภาพที่สามารถตัดป้ายทะเบียนได้นั้นจะต้องเป็นภาพที่มีป้ายทะเบียนอยู่ประมาณกึ่งกลางของภาพและต้องไม่เอียงมากนัก แสงและเงาของรูปก็จะต้องชัด ไม่มีแสงสะท้อนบริเวณตัวรถมากนัก หรือมีเงามาบังบริเวณป้ายทะเบียน ตัวอย่างของภาพที่สามารถแยกป้ายทะเบียนออกมาได้ชัดเจนแสดงดังรูป 4.24



รูปที่ 4.24 ตัวอย่างภาพที่แยกป้ายทะเบียนได้ชัดเจน

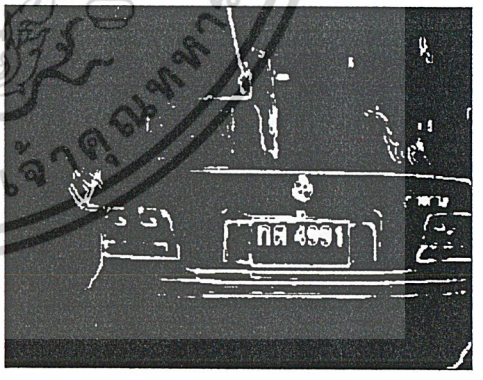
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ไม่สามารถตรวจหาขอบเขตของป้ายทะเบียนได้ มีสาเหตุดังต่อไปนี้คือ

- ภาพมีขนาดที่ไม่เหมาะสม ดังตัวอย่างรูปที่ 4.25

รูปที่ 4.25 ตัวอย่างภาพที่มีขนาดป้ายทะเบียนเล็กเกินไป

- สภาพแวดล้อมไม่เหมาะสม ดังแสดงตัวอย่างในรูปที่ 4.26



ก) ภาพต้นฉบับ ข) ภาพที่ผ่านการหาขอบ

รูปที่ 4.26 ตัวอย่างภาพที่มีสภาพแวดล้อมไม่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.26 บริเวณป้ายทะเบียนด้านซ้ายบางส่วนจะมีสีเดียวกับตัวรถเนื่องจากมีแสงสะท้อนที่ตัวรถบริเวณนั้น ทำให้หาบริเวณขอบของป้ายทะเบียนไม่ได้ เพราะการหาขอบ จะหาขอบได้เฉพาะบริเวณที่มีสีแตกต่างกันเท่านั้น ดังจะเห็นได้จากรูป 4.26ข บริเวณขอบของป้ายทะเบียนด้านซ้ายจะขาดหายไป

เมื่อแยกตัวอักขระออกจากภาพได้แล้วจะนำอักขระทั้งหมดที่ได้ไปทำการหาว่าเป็นอักขระใดบ้าง จากนั้นผลลัพธ์ที่ได้จะเป็นหมายเลขทะเบียนที่ได้จากภาพนั้น ผลการหาหมายเลขทะเบียนแสดงดังตารางที่ 4.1

หมายเลขทะเบียน	หมายเลขที่หาได้	หมายเลขทะเบียน	หมายเลขที่หาได้
กก 8712	กก 8712	ธ 2226	ธ 2226
พน 2205	หาไม่ได้	ธข 1154	ธข 1154
กก 4849	กก 4849	วฌ 3632	วฌ 3632
วอ 9292	วอ 9292	ภค 8488	ภค 8488
กต 4991	กต 4991	พว 728	พว 728
พน 2205	?น ฮ205	วห 3814	วห 3814
ก 4393	ก 4393	6ษ 8583	6ษ 8583
3ธ 8926	3ธ 8926	พอ 5195	พอ ?5195
2ฐ 0563	ฮฐ 0563	ศจ 5165	ศจ 5165
ฌต 5780	ฌต 5780	ฉม 3944	ฉม 39??
ก 6239	ก 62??	1ฮ 4505	1ฮ 4505
กง 6760	กง 6760	1ษ 6930	1ษ 6930
1ฐ 3230	1ฐ 3230	ฉอ 7412	ฉอ 7412
ษจ 8385	ษจ 8385	ฉล 7472	ฉล 7472
กก 4849	กก ??4?	ภฉ 140	ภฉ 140
กฉ 1909	กฉ 1909	2ฐ 1763	2ฐ 1763
ษล 6409	ษล 6409	ลน 4399	น 4399
กค 3477	กค 3477	กก 1309	กก 1309
วฉ 2674	วฉ 2674	7ฮ 1501	7ฮ 1501
ลก 588	ลก 588	ษข 4918	ษข 4918
กร 4923	ถ? 4923	วฐ 6773	ว? 6773
9ฮ 8163	9ฮ 8163	ภธ 1949	ภธ 1949

ตารางที่ 4.1 ผลการหาหมายเลขทะเบียนจากภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขทะเบียน	หมายเลขที่หาได้	หมายเลขทะเบียน	หมายเลขที่หาได้
ธก 1307	ธก 1307	วท 2053	วท 2053
ฉพ 9417	ฉ? 9417	ษษ 2144	ษษ 2144
กบ 3857	กบ 3857	วม 6850	วม 6850
วอ 1335	วอ 1335	ภน 5292	ภน 5292
ฐท 142	ฐท 142	พอ 8143	พ? 8143
วท 8685	ว? 8685	ก 9658	ก 9658
วภ 4305	วภ 4305	จค 1826	จค 1826
วต 2283	วต 2283	วพ 7929	วพ 7929
ษต 2113	ษต 2113	วน 1543	วน 1543
วว 5678	วว 5678	2ธ 6919	ฮธ 691?
วษ 2367	วษ ฐ367	ษร 3236	ษร 3236
วบ 8970	วบ 8970	กข 4350	กข 4350
7ฐ 1228	7ฐ 1228	3พ 9235	3พ 9235
กพ 518	กพ 518	ษข 2886	ษข 2886
กก 9647	กก 9647	กข 5982	กข 5982
1ฮ 2611	1ฮ 2611	3ณ 9998	หาไม่ได้
ษอ 6821	ษอ 6821	วร 7074	วร 7074
ภจ 9396	ภจ 9396	ษอ 4192	ษอ 419?
ฉย 7284	ฉ? 7284	กต 3429	หาไม่ได้
วฮ 2811	วฮ 2811	ภฉ 6954	หาไม่ได้
รฐ 4427	รฐ 4427	พย 7018	พย 7018
วม 1984	วม 1984	ษณ 7174	ษณ 7174
ษต 6582	ษต 6582	กข 8200	กข 8200
วศ 9427	วศ 9427	ษต 2113	ษต 2113
กท 8562	กท 8562	ศก 9890	ก 9890
ษห 326	ษห 326	5ฉ 2467	5ฉ 2467
ษณ 6905	ษณ 6905	ศก 5726	ก 5726
4ฐ 2661	?ฐ 2661	บท 6189	หาไม่ได้
ษพ 1783	ษพ 1783	วบ 7654	วบ 7654
ษห 6238	ษห 6238	ษร 4699	ษร 4699

ตารางที่ 4.1 ผลการหาหมายเลขทะเบียนจากภาพ(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขทะเบียน	หมายเลขที่หาได้	หมายเลขทะเบียน	หมายเลขที่หาได้
บง 6287	ง 6287	ก 9658	ก 9658
ภก 6054	ภก 6054	ฉจ 3413	ฉจ 3413
ก 3949	ก 3949	ษก 1330	ษก 1330
กต 7976	กต 7976	9พ 5620	9พ 5620
ฉล 9092	ฉล 9092	วท 5620	วท 5620
วน 1543	วน 1543	กง 7236	กง 7236
6ว 9981	หาไม่ได้	1ณ 3626	1ณ 3626
ษธ 5068	ษธ 5068	ภพ 8688	หาไม่ได้
บง 6511	บง 6511	กง 2671	กง 2671
ฐอ 6990	ฐอ 6990	วณ 5439	วณ 5439
ก 6378	ก 6378	ศก 6566	ก 6566
ษฐ 3013	ษฐ 3013	กง 7236	กง 7236
ษอ 6821	ษอ 6821	9ศ 2896	2896
จ 9296	? 9296	ภพ 3673	ภพ 3673
กธ 6963	ก? 6963	ษอ 6821	ษอ 6821
กก 9647	กก 9647	พฮ 4438	พฮ 4438
ษข 4468	ษข 4468	ศจ 5956	ศจ 5956
วน 7872	วน 7872	ษจ 548	ษจ 548
ษอ 4192	ษอ 4192	ณฉ 9088	หาไม่ได้
ภข 2124	ภข 2124	ษอ 5824	ษอ 5824
ภพ 3673	ภพ 3673	1ศ 0556	1ศ 0556
พล 4522	พล 4522	ม 1489	ม 1489
พฮ 4438	พฮ 4438	วม 1050	วม 1050
ภจ 3097	ภจ 3097	กฉ 5807	กฉ 5807

ตารางที่ 4.1 ผลการหาหมายเลขทะเบียนจากภาพ(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนภาพที่สามารถหาหมายเลขทะเบียนได้ถูกต้องจากภาพทั้งหมด แสดงดังตารางที่ 4.2

ผลการทดลอง	จำนวนภาพ (จากทั้งหมด 150 ภาพ)	เปอร์เซ็นต์
หาหมายเลขทะเบียนได้ถูกต้อง	123	82

ตารางที่ 4.2 จำนวนภาพที่สามารถหาหมายเลขทะเบียนได้ถูกต้องและจำนวนเปอร์เซ็นต์

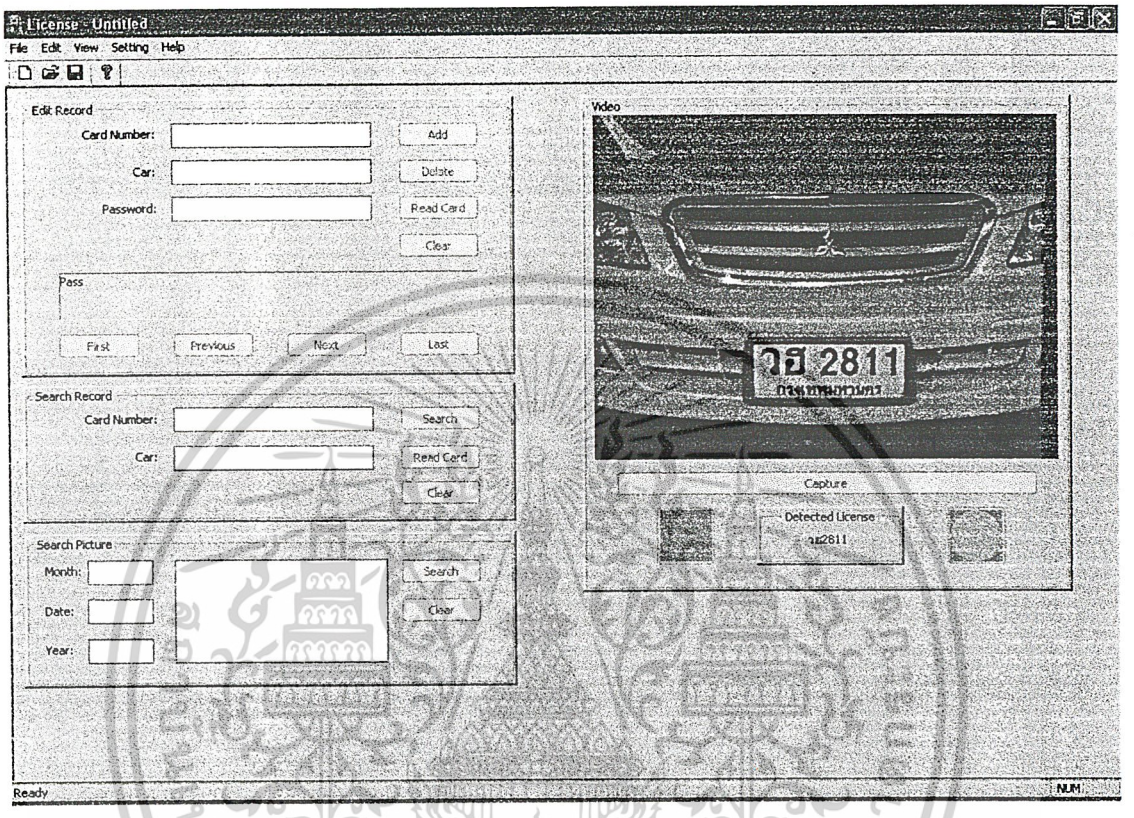
หากคิดแยกเป็นตัวอักขระที่สามารถหาได้ถูกต้องผลที่ได้แสดงดังตาราง 4.3

อักขระ	เปอร์เซ็นต์	อักขระ	เปอร์เซ็นต์
0	100	ต	100
1	100	ท	75
2	95.1	ธ	100
3	98.5	น	100
4	94.3	บ	85.7
5	100	พ	80
6	98.7	ภ	100
7	100	ม	100
8	98.6	ย	50
9	97.3	ร	80
ก	97.9	ล	83.3
ข	100	ว	100
ค	100	ศ	57
ง	100	ษ	100
จ	83.3	ห	100
ฉ	100	ฬ	100
ฐ	92.8	อ	88.2
ณ	100	ฮ	100

ตารางที่ 4.3 เปอร์เซ็นต์ผลการหาอักขระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

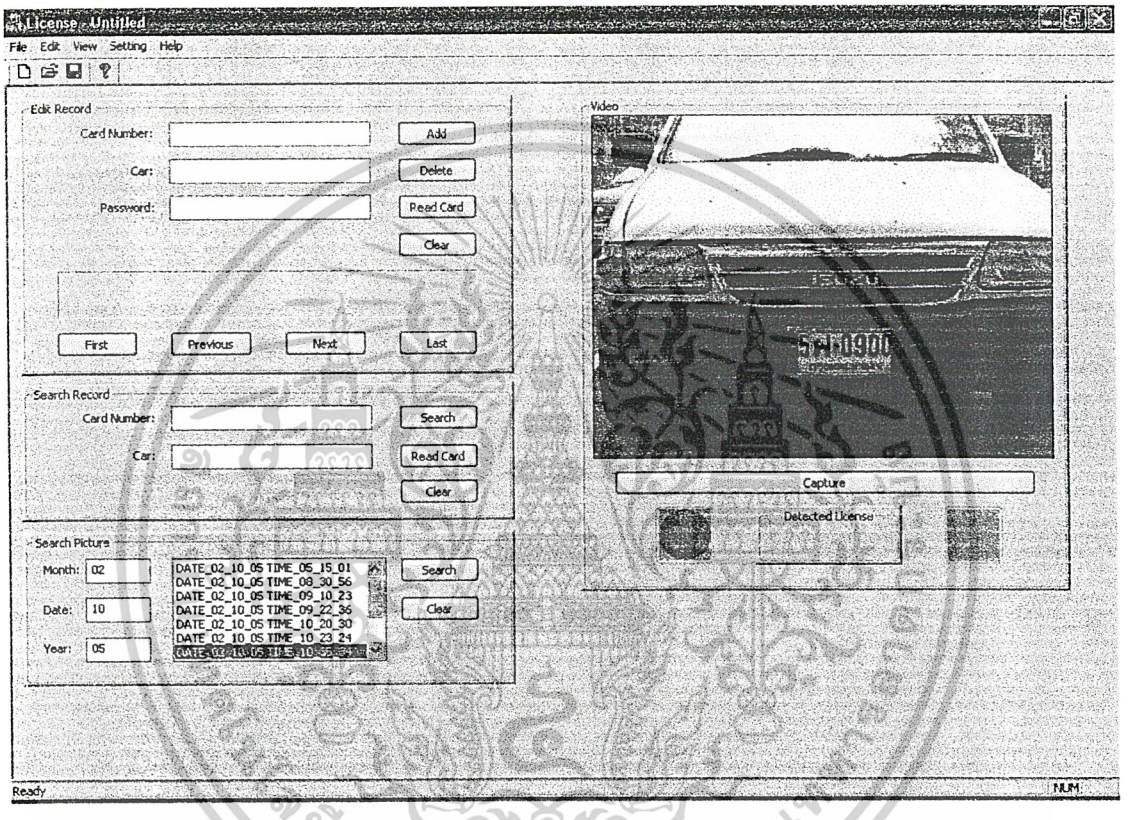
เมื่อการตรวจสอบพบว่าหมายเลขบัตรถูกต้องและหมายเลขทะเบียนรถยนต์ถูกต้อง หรือในอีกกรณีหนึ่ง หมายเลขบัตรถูกต้องโดยหมายเลขทะเบียนรถยนต์ไม่ถูกต้องแต่การกดรหัสถูกต้อง ระบบจะอนุญาตให้ผู้ใช้นำรถเข้า/ออก จากที่จอดรถได้ โปรแกรมจะแสดงหมายเลขทะเบียนรถยนต์ที่หาได้และแสดงผลดังรูป 4.27



รูปที่ 4.27 หน้าจอโปรแกรมเมื่อระบบอนุญาตให้ผู้ใช้นำรถเข้า/ออก จากที่จอดรถได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบันทึกภาพเข้าเครื่องคอมพิวเตอร์จะมีรูปแบบของชื่อไฟล์เป็น "DATE_เดือน_วัน_ปี TIME_ชั่วโมง_นาที_วินาที" ตัวอย่างเช่น ภาพที่บันทึกเมื่อ เดือน 2 วันที่ 20 ปี 2005 เวลา 19 นาฬิกา 20 นาที 30 วินาที จะมีชื่อไฟล์เป็น "DATE_02_20_05 TIME_19_20_30" โดยในโปรแกรมผู้ควบคุมสามารถทำการค้นหาภาพเพื่อจะทำการเรียกดูได้โดยทำการระบุ เดือน วัน และปีของภาพที่ต้องการดูลงไปในโปรแกรม จากนั้นโปรแกรมจะแสดงภาพที่บันทึกไว้ในวันนั้นขึ้นมายังลิสต์บอกซ์ (List Box) และเมื่อทำการคลิกเลือกที่ชื่อภาพ ภาพจะปรากฏทางด้านขวามือของโปรแกรม ดังรูป 4.28



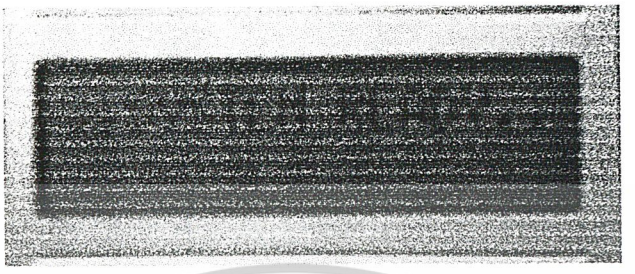
รูปที่ 4.28 หน้าจอโปรแกรมเมื่อผู้ควบคุมระบบทำการเรียกดูภาพที่ได้บันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การแสดงสถานะการทำงานบนแอลซีดี โมดูล

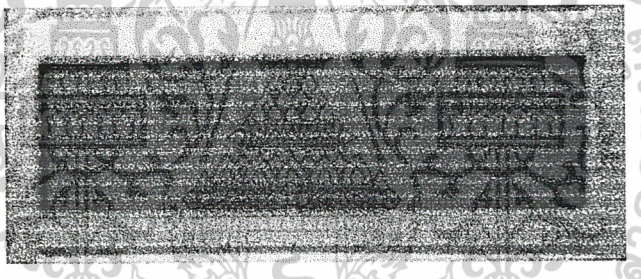
แอลซีดี โมดูล มีหน้าที่ในการแสดงสถานะการทำงานของระบบให้ผู้ใช้ได้ทราบว่าผลการใช้งานเป็นอย่างไร

เมื่อระบบอยู่ในสถานะรอรับการรูดบัตร แอลซีดี โมดูล จะแสดงผลดังรูป 4.28



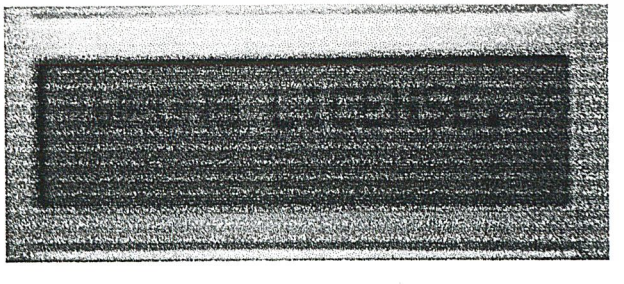
รูปที่ 4.29 แอลซีดี โมดูล แสดงสถานะรอรับการรูดบัตร

เมื่อผู้ใช้ทำการรูดบัตร หากหมายเลขบัตรไม่ตรงกับที่มีอยู่ในฐานข้อมูล หรือการรูดบัตรเกิดการผิดพลาด แอลซีดี โมดูล จะแสดงผลดังรูป 4.29



รูปที่ 4.30 แอลซีดี โมดูล แสดงผลการรูดบัตรเมื่อหมายเลขบัตรที่ใช้ไม่ตรงกับที่มีอยู่ในฐานข้อมูลหรือการรูดบัตรเกิดการผิดพลาด

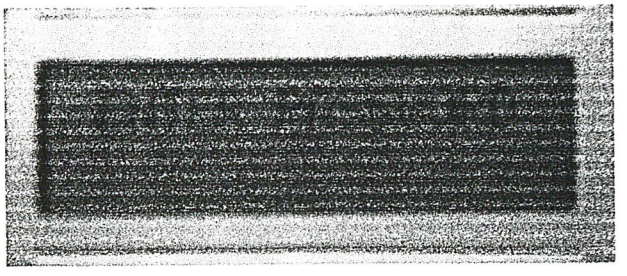
หากหมายเลขบัตรตรงกับที่มีอยู่ในฐานข้อมูล แต่หมายเลขทะเบียนรถยนต์ไม่ถูกต้องหรือระบบไม่สามารถหาหมายเลขทะเบียนจากภาพได้ แอลซีดี โมดูล จะแสดงผลดังรูป 4.30



รูปที่ 4.31 แอลซีดี โมดูล แสดงผลว่าหมายเลขทะเบียนรถยนต์ไม่ถูกต้องหรือระบบไม่สามารถหาหมายเลขทะเบียนได้

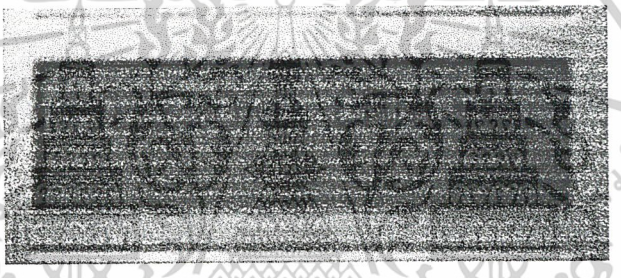
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อไม่สามารถหาหมายเลขทะเบียนรถยนต์จากภาพได้ ระบบจะให้ผู้ใช้ทำการกดรหัสผ่านเป็นการทดแทน แอลซีดี โมดูล จะแสดงผลดังรูป 4.31



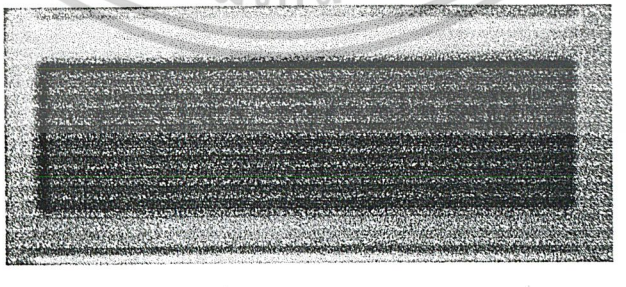
รูปที่ 4.32 แอลซีดี โมดูล แสดงผลให้ผู้ใช้ทำการกดรหัสผ่าน

หากผู้ใช้ต้องการยกเลิกการกดรหัส ผู้ใช้สามารถกดเครื่องหมาย '*' บนคีย์แพดได้ จากนั้นระบบจะกลับมาอยู่ในสถานะรอรับการรูดบัตรอีกครั้ง



รูปที่ 4.33 แอลซีดี โมดูล แสดงผลการกดเครื่องหมาย '*' เมื่อผู้ใช้ต้องการยกเลิกการกดรหัส

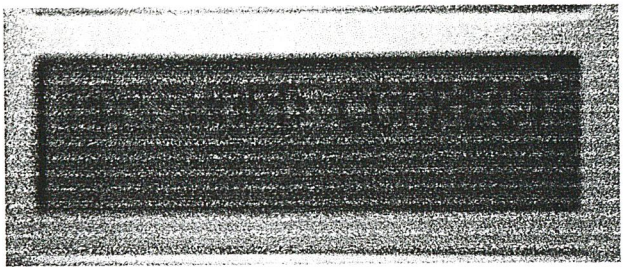
เมื่อผู้ใช้ไม่ได้ทำการกดเครื่องหมาย '*' เพื่อยกเลิกการกดรหัสและผู้ใช้ทำการกดรหัสจนครบแล้ว ผู้ใช้ต้องทำการกดเครื่องหมาย '#' เพื่อบอกให้ระบบทราบว่ากดรหัสครบแล้ว จากนั้นระบบจะทำการตรวจสอบรหัสที่ผู้ใช้กดว่าถูกต้องหรือไม่ โดยถ้ารหัสไม่ถูกต้องแอลซีดี โมดูล จะแสดงผลดังรูป 4.33



รูปที่ 4.34 แอลซีดี โมดูล แสดงผลการกดรหัสไม่ถูกต้อง

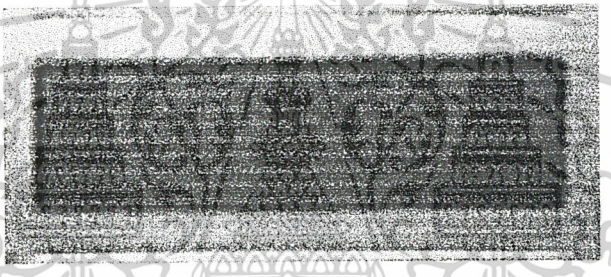
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าผู้ใช้กดรหัสถูกต้อง แอลซีดี โมดูล จะแสดงผลดังรูป 4.34



รูปที่ 4.35 แอลซีดี โมดูล แสดงผลการกดรหัสถูกต้อง

เมื่อการตรวจสอบพบว่าหมายเลขบัตรถูกต้องและหมายเลขทะเบียนรถยนต์ถูกต้อง หรือในอีกกรณีหนึ่ง หมายเลขบัตรถูกต้องโดยหมายเลขทะเบียนรถยนต์ไม่ถูกต้องแต่การกดรหัสถูกต้อง ระบบจะอนุญาตให้ผู้ใช้นำรถเข้า/ออก จากที่จอดรถได้ แอลซีดี โมดูล จะแสดงผลดังรูป 4.35



รูปที่ 4.36 แอลซีดี โมดูล แสดงผลว่าระบบอนุญาตให้ผู้ใช้นำรถเข้า/ออก จากที่จอดรถได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

ในการทำระบบรักษาความปลอดภัยสำหรับที่จอดรถยนต์ต้องใช้ความรู้หลายๆด้าน ไม่ว่าจะเป็นการเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาซี การเขียนโปรแกรมไมโครคอนโทรลเลอร์ การควบคุม การรับ-ส่งสัญญาณผ่านทาง RS-232 และการแสดงผลผ่านทางแอลซีดี โมดูล และความรู้เรื่องการประมวลผลจากภาพเชิงตัวเลข (Digital Image processing)

จากการทดลองจะพบว่าในส่วนของเครื่องอ่านบัตรแม่เหล็กจะทำงานได้อย่างถูกต้องหากบัตรแม่เหล็กมีสภาพสมบูรณ์ ในส่วนของโปรแกรมรับสัญญาณจาก RS-232 และจัดทำระบบฐานข้อมูลสามารถทำงานได้อย่างถูกต้อง และในส่วนของการแยกป้ายทะเบียนออกจากภาพสามารถทำงานได้อย่างมีประสิทธิภาพพอสมควรหากมีสภาพแวดล้อมที่เหมาะสมไม่มีเงาหรือวัตถุมาบังป้ายทะเบียน และสุดท้ายการการแยกตัวอักษรออกเป็นตัวๆก็สามารถทำงานได้ถูกต้องหากมีสภาพแวดล้อมเหมาะสม ไม่มีฝุ่นหรือโคลนมาติดที่ป้ายทะเบียน

ปัญหาที่พบในส่วนของเครื่องอ่านบัตรแม่เหล็กหากบัตรแถบแม่เหล็กมีสภาพไม่สมบูรณ์ก็จะไม่สามารถทำงานได้ ในส่วนของการแยกป้ายทะเบียนออกจากภาพหากภาพที่จับได้มีสภาพแวดล้อมไม่เหมาะสม เช่น หากระยะทางของรถกับตัวกล้องห่างกันมากเกินไป หรือหากตำแหน่งที่รถจอดอยู่เป็นตำแหน่งที่เกิดเงา ก็จะไม่สามารถทำการแยกป้ายทะเบียนออกจากภาพได้ และปัญหาในการแยกป้ายทะเบียนออกจากภาพหากภาพอีกอย่างคือหากภาพที่ได้เบลอหรือมีความแตกต่างของค่าสีน้อยก็จะทำให้การหาป้ายทะเบียนผิดพลาด และในส่วนของการแยกตัวอักษรออกเป็นตัวๆ หากแผ่นป้ายทะเบียนมีสภาพไม่สมบูรณ์ เช่น มีการติดน๊อตหรือมีฝุ่นติดเป็นจำนวนมาก ก็จะทำให้ไม่สามารถทำแยกตัวอักษรออกเป็นตัวๆได้อย่างถูกต้องได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ก. โปรแกรมในส่วนของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      ORG 0000H
      BIT1 EQU 21H
      DATA EQU P1.0
      MOV PCON,#00H
      MOV SCON,#50H
      MOV TMOD,#20H
      MOV TH1,#0FDH
START: MOV 45H,#00H
      MOV 49H,#00H
      MOV P0,#00H
      CLR P1.4
      CLR P1.5
      CLR P1.6
      CLR P1.3
      SETB P2.7
      MOV A,#38H
      LCALL LCDWI
      MOV A,#0CH
      LCALL LCDWI
START4:     MOV A,#01H
      LCALL LCDWI
      MOV A,#06H
      LCALL LCDWI
      MOV A,#80H
      LCALL LCDWI
      MOV DPTR,#SYSTEM_READY
      MOV R7,#10H
      LCALL LCDLDS
      MOV A,#11110111B
      MOV P1,A
AAB:  JB P1.1,AAC ;CHECK PRESENT SIGNAL
      LJMP AA5    ;HAVE PRESENT SIGNAL
AAC:  SETB TR1    ;CHECK FOR DATA IN
      JNB RI,START4
      CLR RI
      MOV A,SBUF
CHK_R:      CJNE A,#01010010B,CHK_W
      MOV 45H,A
      JB P1.1,$
      LJMP AA5
CHK_W: CJNE A,#01010111B,CHK_P
      LCALL INITIAL_LCD
      MOV A,#83H
      LCALL LCDWI
      MOV DPTR,#WRONG_CARD
      MOV R7,#0AH
      LCALL LCDLDS
      LCALL DELAY_1S
      LJMP START
CHK_P:      CJNE A,#01010000B,CHK_U
      SETB P1.3
PP:   CLR P2.7
      LCALL INITIAL_LCD
      MOV A,#86H
      LCALL LCDWI
      SETB P1.3
      MOV DPTR,#PASS
      MOV R7,#04H
      LCALL LCDLDS
      SETB P1.3
      CLR P2.7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL DELAY_1S
KK:     LJMP START
CHK_U:      CJNE A,#01010101B,KK
        CLR P1.3
        LCALL INITIAL_LCD
        MOV A,#81H
        LCALL LCDWI
        CLR P1.3
        MOV DPTR,#WRONG_LICENSE
        MOV R7,#0EH
        LCALL LCDLDS
        CLR P1.3
        LCALL DELAY_1S
        LCALL INITIAL_LCD
        MOV A,#00011100B
        CLR P1.3
        LCALL LCDWI
        MOV A,#80H
        LCALL LCDWI
        CLR P1.3
        MOV DPTR,#ENTER_PASSWORD
        MOV R7,#0EH
        LCALL LCDLDS
        CLR P1.3
        MOV A,#0C0H
        LCALL LCDWI
START2:     CLR P1.3
        MOV P2,#0FFH
        CLR A
R01:    CLR P2.4
R01C1:      JB P2.6,R01C2
        LCALL WAIT
        JB P2.6,R01C2
        MOV A,#31H
        LCALL SEND
        CLR P1.3
        JNB P2.6,$
        LCALL WAIT
        JNB P2.6,$
        LJMP START2
R01C2:      JB P2.2,R01C3
        LCALL WAIT
        JB P2.2,R01C3
        MOV A,#32H
        LCALL SEND
        CLR P1.3
        JNB P2.2,$
        LCALL WAIT
        JNB P2.2,$
        LJMP START2
R01C3:      JB P2.5,R02
        LCALL WAIT
        JB P2.5,R02
        MOV A,#33H
        LCALL SEND
        CLR P1.3
        JNB P2.5,$
        LCALL WAIT
        JNB P2.5,$
        LJMP START2
R02:    SETB P2.4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      CLR P2.3
R02C1:      JB P2.6,R02C2
      LCALL WAIT
      JB P2.6,R02C2
      MOV A,#34H
      LCALL SEND
      CLR P1.3
      JNB P2.6,$
      LCALL WAIT
      JNB P2.6,$
      LJMP START2
R02C2:      JB P2.2,R02C3
      LCALL WAIT
      JB P2.2,R02C3
      MOV A,#35H
      LCALL SEND
      CLR P1.3
      JNB P2.2,$
      LCALL WAIT
      JNB P2.2,$
      LJMP START2
R02C3:      JB P2.5,R03
      LCALL WAIT
      JB P2.5,R03
      MOV A,#36H
      LCALL SEND
      CLR P1.3
      JNB P2.5,$
      LCALL WAIT
      JNB P2.5,$
      LJMP START2
R03:  CLR P1.3
      MOV A,#80H
      LCALL LCDWI
      CLR P1.3
      MOV DPTR,#ENTER_PASSWORD
      MOV R7,#0EH
      LCALL LCDLDS
      CLR P1.3
      MOV A,49H
      ADD A,#0C0H
      ;MOV A,#0C0H
      LCALL LCDWI
      SETB P2.3
      CLR P2.1
R03C1:      JB P2.6,R03C2
      LCALL WAIT
      JB P2.6,R03C2
      MOV A,#37H
      LCALL SEND
      CLR P1.3
      JNB P2.6,$
      LCALL WAIT
      JNB P2.6,$
      LJMP START2
R03C2:      JB P2.2,R03C3
      LCALL WAIT
      JB P2.2,R03C3
      MOV A,#38H
      LCALL SEND
      CLR P1.3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JNB P2.2,$
        LCALL WAIT
        JNB P2.2,$
        LJMP START2
R03C3:      JB P2.5,R04
        LCALL WAIT
        JB P2.5,R04
        MOV A,#39H
        LCALL SEND
        CLR P1.3
        JNB P2.5,$
        LCALL WAIT
        JNB P2.5,$
        LJMP START2
R04:    SETB P2.1
        CLR P2.0
R04C1:      JB P2.6,R04C2
        LCALL WAIT
        JB P2.6,R04C2
        MOV A,#2AH
        LCALL SEND
        CLR P1.3
        LCALL INITIAL_LCD
        MOV A,#80H
        LCALL LCDWI
        MOV DPTR,#PASSWORD_CANCEL
        MOV R7,#10H
        LCALL LCDLDS
        CLR P1.3
        LCALL DELAY_1S
        JNB P2.6,$
        LCALL WAIT
        JNB P2.6,$
        LJMP START
R04C2:      JB P2.2,R04C3
        LCALL WAIT
        JB P2.2,R04C3
        MOV A,#30H
        LCALL SEND
        CLR P1.3
        JNB P2.2,$
        LCALL WAIT
        JNB P2.2,$
        LJMP ST2
R04C3:      JB P2.5,ST2
        LCALL WAIT
        JB P2.5,ST2
        MOV A,#23H
        LCALL SEND
        CLR P1.3
        JNB P2.5,$
        LCALL WAIT
        JNB P2.5,$
HH:     JNB RI,$
        CLR RI
        MOV A,SBUF
CHK_I:      CJNE A,#01001001B,CHK_C
        LCALL INITIAL_LCD
        MOV A,#81H
        LCALL LCDWI
        MOV DPTR,#WRONG_PASSWORD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      MOV R7,#0EH
      LCALL LCDLDS
      LCALL DELAY_1S
ST3:  LJMP START
CHK_C:      CJNE A,#01000011B,HH
      LCALL INITIAL_LCD
      MOV A,#80H
      LCALL LCDWI
      MOV DPTR,#PASSWORD_CORRECT
      MOV R7,#10H
      LCALL LCDLDS
      LCALL DELAY_1S
      LJMP PP
ST2:  LJMP START2
WAIT: MOV R4,#13H;13
LOOP4A:     MOV R3,#14H;14
LOOP3A:     MOV R2,#0CH;19
LOOP2A:     CLR P1.3
      DJNZ R2,LOOP2A
      DJNZ R3,LOOP3A
      DJNZ R4,LOOP4A
      RET
SEND: MOV SBUF,A
      JNB TI,$
      CLR TI
      MOV A,#2AH
      LCALL LCDWD
      MOV A,49H
      INC A
      MOV 49H,A
      RET
      LJMP START
AA5:
 A1:  LCALL CLOCK
      JNB DATA,A2
      SJMP A1
 A2:  LCALL CLOCK
      JNB DATA,A3
      SJMP A1
 A3:  LCALL CLOCK
      JB DATA,A4
       SJMP A1
 A4:  LCALL CLOCK
      JNB DATA,A5
      SJMP A1
 A5:  LCALL CLOCK
      JB DATA,READDATA
      SJMP A1
 ;********************************* READ DATA
*********************************************
 READDATA: MOV R1,#21H
           MOV BIT1,#00H
 READ_1:   INC R1
      LCALL CLOCK
      JB DATA,D1_0
      SETB BIT1.0
      SJMP E1_0
 D1_0:      CLR BIT1.0
 E1_0:      LCALL CLOCK
            JB DATA,D1_1
      SETB BIT1.1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      LJMP E1_1
D1_1:       CLR BIT1.1
E1_1:    LCALL CLOCK
            JB DATA,D1_2
      SETB BIT1.2
      LJMP E1_2
D1_2:       CLR BIT1.2
E1_2:    LCALL CLOCK
            JB DATA,D1_3
      SETB BIT1.3
      LJMP E1_3
D1_3:       CLR BIT1.3
E1_3:       LCALL CLOCK
            JB DATA,D1_4
      SETB BIT1.4
      LJMP E1_4
D1_4:       CLR BIT1.4
E1_4:       MOV @R1,BIT1
      CJNE R1,#31H,READ_1
;************************************** SEND DATA
**********************************
      MOV A,45H
      CJNE A,#01010010B,NO_R
      LCALL INITIAL_LCD
      MOV A,#82H
      LCALL LCDWI
      MOV DPTR,#READING_CARD
      MOV R7,#0CH
      LCALL LCDLDS
      LJMP SEND_DATA
NO_R: LCALL INITIAL_LCD
      MOV A,#81H
      LCALL LCDWI
      CLR P1.3
      MOV DPTR,#CHECKING_DATA
      MOV R7,#0EH
      LCALL LCDLDS
      CLR P1.3
SEND_DATA:  MOV R1,#21H
 LOOP1:     INC R1
      MOV A,@R1
      MOV BIT1,A
      CLR BIT1.4
      MOV A,BIT1
      ADD A,#30H
      MOV SBUF,A
 WAIT1: JNB TI,WAIT1
      CLR TI
      CJNE A,#3FH,NOT_END
      SJMP START3
 NOT_END:   CJNE R1,#31H,LOOP1
      LCALL DELAY
 START3:MOV R1,#0C8H;***************200D***************
HAHA: MOV R2,#3CH;******************60D***************
ABCD: MOV R3,#29H;************41D***************
      DJNZ R3,$
      DJNZ R2,ABCD
      DJNZ R1,HAHA
      LCALL INITIAL_LCD
      MOV A,#81H
      LCALL LCDWI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      CLR P1.3
      MOV DPTR,#CHECKING_DATA
      MOV R7,#0EH
      LCALL LCDLDS
      CLR P1.3
      LCALL DELAY_1S
      LJMP START

;*********************************************CLOCK*****************
*******************
 CLOCK: JNB P1.2,$
      JB P1.2,$
      RET


;************************************************************************
*******************
DELAY:      MOV R4,#14H
LOOP4:      MOV R3,#64H
LOOP3:      MOV R2,#64H
LOOP2:      DJNZ R2,LOOP2
      DJNZ R3,LOOP3
      DJNZ R4,LOOP4
      RET
;*****************************************LCD*************************
**********
INITIAL_LCD:      MOV A,#38H
      LCALL LCDWI
      MOV A,#0CH
      LCALL LCDWI
      MOV A,#01H
      LCALL LCDWI
      MOV A,#06H
      LCALL LCDWI
      RET
LCDWI:      MOV R4,A
      CLR ACC.4    ;RS=0
      CLR ACC.5    ;R/W=0
      CLR ACC.6    ;ENABLE=0
      MOV P1,A
      SETB ACC.6   ;ENABLE=1
      MOV P1,A
      MOV A,R4
      MOV P0,A
      CLR ACC.4
      CLR ACC.5
      CLR ACC.6
      MOV P1,A
      MOV A,#0E0H
LCDWI1:DEC A
      JNZ LCDWI1
      RET
LCDWD:      MOV R5,A
      SETB ACC.4   ;RS=0
      CLR ACC.5    ;R/W=0
      CLR ACC.6    ;ENABLE=0
      MOV P1,A
      SETB ACC.6   ;ENABLE=1
      MOV P1,A
      MOV A,R5
      MOV P0,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SETB ACC.4
        CLR ACC.5
        CLR ACC.6
        MOV P1,A
        MOV A,#0E0H
LCDWD1:DEC A
        JNZ LCDWD1
        RET
DELAYA:MOV R2,#10H
ABC:    MOV R3,#10H
        DJNZ R3,$
        DJNZ R2,ABC
        DJNZ R1,DELAYA
        RET
LCDLDS:LCALL LCDWI
LCDLDS1:CLR A
        MOVC A,@A+DPTR
        LCALL LCDWD
        INC DPTR
        DJNZ R7,LCDLDS1
        RET
DELAY_1S:    MOV R4,#0C8H
LOOP44:      MOV R3,#3CH
LOOP34:      MOV R2,#29H
LOOP24:      DJNZ R2,LOOP24
             DJNZ R3,LOOP34
             DJNZ R4,LOOP44
             RET
SYSTEM_READY:      DB "..SYSTEM READY.."
READING_CARD:      DB "READING CARD"
CHECKING_DATA:     DB "CHECKING DATA."
WRONG_CARD:              DB "WRONG CARD"
WRONG_LICENSE:     DB "WRONG LICENSE."
ENTER_PASSWORD:    DB "ENTER PASSWORD"
WRONG_PASSWORD:    DB "WRONG PASSWORD."
PASSWORD_CORRECT:        DB "PASSWORD CORRECT"
PASS:       DB "PASS"
PASSWORD_CANCEL: DB "PASSWORD CANCEL."
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ข. โปรแกรมในส่วนของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ LicenseView.cpp

```
// LicenseView.cpp : implementation of the CLicenseView class
#include <string.h>
#include "time.h"
#include "stdafx.h"
#include "License.h"
#include "Licenses.h"
#include "LicenseDoc.h"
#include "LicenseView.h"
#include <iostream>
#include <iso646.h>
#include "PImage.h"
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include "SetPath.h"
#include "SetComPort.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#endif
int bufpic=0;
CPImage image1,image2;
// CBitOpView
CString out_11;
CPImage Pimage;
char out_1[30];
CPImage Pimage1;
HANDLE hComp;
DCB dcb;
char sum[15];
char summ[15];
CString buff;
CString buff2;
CSetPath Set;
char chBuff;
int i;
int valReturn;
unsigned long nReadPort;
unsigned long nWritePort;

// CLicenseView

IMPLEMENT_DYNCREATE(CLicenseView, CFormView)

BEGIN_MESSAGE_MAP(CLicenseView, CFormView)
      ON_BN_CLICKED(IDC_ADD_RECORD, OnBnClickedAddRecord)
      ON_BN_CLICKED(IDC_DELETE_RECORD, OnBnClickedDeleteRecord)
      ON_BN_CLICKED(IDC_FIRST_RECORD, OnBnClickedFirstRecord)
      ON_BN_CLICKED(IDC_PRE_RECORD, OnBnClickedPreRecord)
      ON_BN_CLICKED(IDC_NEXT_RECORD, OnBnClickedNextRecord)
      ON_BN_CLICKED(IDC_LAST_RECORD, OnBnClickedLastRecord)
      ON_BN_CLICKED(IDC_RECV, OnBnClickedRecv)
      ON_BN_CLICKED(IDC_CLEAR, OnBnClickedClear)
      ON_BN_CLICKED(IDC_FIND, OnBnClickedFind)
      ON_BN_CLICKED(IDC_BUTTON2, OnBnClickedButton2)
      ON_BN_CLICKED(IDC_RECV2, OnBnClickedRecv2)
      ON_BN_CLICKED(IDC_WORK, OnBnClickedWork)
      ON_WM_CLOSE()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	ON_COMMAND(ID_PATH_SETPATH, OnPathSetpath)
	ON_COMMAND(ID_CAPTURE_VIDEOSOURCE, OnCaptureVideosource)
	ON_COMMAND(ID_CAPTURE_VIDEOFORMAT139, OnCaptureVideoformat139)
	ON_COMMAND(ID_SETTING_COMPORT, OnSettingComport)
	ON_BN_CLICKED(IDC_RADIO, OnBnClickedRadio)
	ON_BN_CLICKED(IDC_RADIO2, OnBnClickedRadio2)
END_MESSAGE_MAP()

// CLicenseView construction/destruction

CLicenseView::CLicenseView()
	: CFormView(CLicenseView::IDD)
	, m_strcard(_T(""))
	, m_strcar(_T(""))
	, m_strcomport(_T(""))
	, m_strshow(_T(""))
	, m_strcardfound(_T(""))
	, m_strcarfound(_T(""))
	, m_find(_T(""))
	, m_find2(_T(""))
	, m_strpw(_T(""))
	, m_strpwfound(_T(""))
	, m_Running(false)
	, m_Image(0)

	, m_time(_T(""))
	, flag1(false)

	  ,m_heigh(0)
	 ,m_width(0)
	 , BPP(0)
	 , heigh(0)
	, width(0)
	, m_hu(_T(""))
	, cca(true)
	, m_comport(_T(""))
	, flag2(false)


{
	m_emList=0;
	m_pCurrent=0;
	m_bFlag=FALSE;
	InitialSerial("COM1");
	nWritePort=0;
	pic="IDB_BITMAP1";
	pic2="IDB_BITMAP3";
m_get=0;

	WORK();
}

CLicenseView::~CLicenseView()
{
}

void CLicenseView::DoDataExchange(CDataExchange* pDX)
{
	CFormView::DoDataExchange(pDX);
	DDX_Text(pDX, IDC_CARD, m_strcard);
	DDV_MaxChars(pDX, m_strcard, 20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	DDX_Text(pDX, IDC_CAR, m_strcar);
	DDV_MaxChars(pDX, m_strcar, 10);
	DDX_Text(pDX, IDC_SHOWSTRING, m_strshow);
	DDX_Text(pDX, IDC_STRFIND, m_find);
	DDV_MaxChars(pDX, m_find, 20);
	DDV_MaxChars(pDX, m_find2, 10);
	DDX_Text(pDX, IDC_STRFIND2, m_find2);
	DDX_Text(pDX, IDC_PW, m_strpw);
	DDX_Control(pDX, IDC_ADD_RECORD, m_ButtonAdd);
	DDX_Control(pDX, IDC_VIDEOOCXCTRL1, m_Video);
	DDX_Control(pDX, IDC_DELETE_RECORD, m_ButtonDelete);
	DDX_Control(pDX, IDC_RECV, m_ButtonRecv);
	DDX_Control(pDX, IDC_WORK, m_Capture);
	DDX_Control(pDX, IDC_CLEAR, m_clear);
	DDX_Control(pDX, IDC_FIRST_RECORD, m_first);
	DDX_Control(pDX, IDC_PRE_RECORD, m_prev);
	DDX_Control(pDX, IDC_NEXT_RECORD, m_next);
	DDX_Control(pDX, IDC_LAST_RECORD, m_last);
	DDX_Control(pDX, IDC_FIND, m_search);
	DDX_Control(pDX, IDC_RECV2, m_recv2);
	DDX_Control(pDX, IDC_BUTTON2, m_clear2);
	DDX_Text(pDX, IDC_HU, m_hu);

}

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

	return CFormView::PreCreateWindow(cs);
}

void CLicenseView::OnInitialUpdate()
{	OnUpdate(this,0,0);
	CFormView::OnInitialUpdate();
	GetParentFrame()->RecalcLayout();
	ResizeParentToFit();
}


// CLicenseView diagnostics

#ifdef _DEBUG
void CLicenseView::AssertValid() const
{
	CFormView::AssertValid();
}

void CLicenseView::Dump(CDumpContext& dc) const
{
	CFormView::Dump(dc);
}

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// CLicenseView message handlers

void CLicenseView::OnBnClickedAddRecord()
{

	m_bFlag = TRUE;
	UpdateData(TRUE);
	if (m_strcard.IsEmpty())
	{	AfxMessageBox("Please enter the card number!");
		return;
	}
	if (m_strcar.IsEmpty())
	{	AfxMessageBox("Please enter the car!");
		return;
	}
	if (m_strpw.IsEmpty())
	{	AfxMessageBox("Please enter the password!");
		return;
	}
	for (int i=0;i<=(m_strpw.GetLength()-1);i++)
	{
		if ((m_strpw.GetAt(i)!='0') && (m_strpw.GetAt(i)!='1')&&
(m_strpw.GetAt(i)!='2') && (m_strpw.GetAt(i)!='3')&&(m_strpw.GetAt
(i)!='4') &&
			(m_strpw.GetAt(i)!='5')&&(m_strpw.GetAt(i)!='6') &&
(m_strpw.GetAt(i)!='7')&&(m_strpw.GetAt(i)!='8') && (m_strpw.GetAt
(i)!='9'))
		{	AfxMessageBox("Password must not be a letter!");
			m_strpw="";
			UpdateData(FALSE);
			return;
		}
	}

	if (!m_emList->IsEmpty()){
		m_pCurrent = m_emList->GetHeadPosition();
		CLicenses *pLicenses = (CLicenses *) m_emList->GetAt
(m_pCurrent);
		m_strcardfound=pLicenses->GetCard();
		m_strcarfound=pLicenses->GetCar();
		m_strpwfound=pLicenses->GetPw();
		if (m_strcardfound==m_strcard)
		{	AfxMessageBox("The record for Card Number:(
"+m_strcard+" ) is already exists.");
			return;
			}
		if (m_strcarfound==m_strcar)
		{	AfxMessageBox("The record for Car:( "+m_strcar+" )
is already exists.");
			return;
				}

		else
			{
			for (long int l=2;l<=(m_emList->GetSize());l++)
				{
				m_emList->GetNext(m_pCurrent);
				if (m_pCurrent)
{CLicenses *pLicenses = (CLicenses *) m_emList->GetAt(m_pCurrent);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m_strcardfound=pLicenses->GetCard();
                    m_strcarfound=pLicenses->GetCar();
                    m_strpwfound=pLicenses->GetPw();
               if (m_strcardfound==m_strcard)
               {    AfxMessageBox("The record for Card Number:(
"+m_strcard+" ) is already exists.");
                    return;
                    }
               if (m_strcarfound==m_strcar)
               {    AfxMessageBox("The record for Car:( "+m_strcar+" )
is already exists.");
                    return;
                              }

                              }

                         }}}
               Add();
          UpdateData(TRUE);
          m_strshow="The record of Card Number:( "+m_strcard+" ) Car:(
"+m_strcar+" ) Password:( "+m_strpw+" ) has been added.";
          ClearRecord();
          UpdateData(FALSE);
}


void CLicenseView::OnBnClickedDeleteRecord()
{
          m_bFlag = FALSE;
          ClicenseDoc* pDoc = GetDocument();
          ASSERT_VALID(pDoc);
          POSITION pos;
          CObject * pE = 0;
          pos = m_pCurrent;
          if (AfxMessageBox("Are you sure you want to delete this
record?",MB_YESNO) != IDYES)
               return;
          if (m_pCurrent !=NULL)
          { m_emList->GetNext(pos);
               if (pos == NULL)
               {    pos = m_emList->GetHeadPosition();
                    if (pos == m_pCurrent)
                              pos = NULL;
               }
               pE = m_emList->GetAt(m_pCurrent);
               m_emList->RemoveAt(m_pCurrent);
               delete pE;
               m_pCurrent = pos;

               pDoc->SetModifiedFlag();
               pDoc->UpdateAllViews(this);
               m_strshow="The record has been deleted.";
               UpdateData(FALSE);
          }
          else
          {    AfxMessageBox("No Record for deletion!!!",MB_OK |
MB_ICONINFORMATION);
               OnBnClickedAddRecord();
          }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void CLicenseView::OnBnClickedFirstRecord()
{     m_bFlag = FALSE;
      if (!m_emList->IsEmpty())
      {     m_pCurrent = m_emList->GetHeadPosition();
            ShowRecord(m_pCurrent);
            m_strshow=(_T(""));
            UpdateData(FALSE);
      }
      else
      {AfxMessageBox("There is no record.",MB_OK|MB_ICONINFORMATION);
      m_strshow=(_T(""));
      UpdateData(FALSE);}
}

void CLicenseView::OnBnClickedPreRecord()
{     m_bFlag = FALSE;
      POSITION pos;
      pos = m_pCurrent;
      if (pos != NULL)
      {     m_emList->GetPrev(pos);
            if(pos)
            {     m_pCurrent = pos;
                  ShowRecord(m_pCurrent);
                  m_strshow=(_T(""));
            UpdateData(FALSE);
            }
            else
            {AfxMessageBox("First record
already.",MB_OK|MB_ICONINFORMATION);
            m_strshow=(_T(""));
            UpdateData(FALSE);}
      }
      else
      {AfxMessageBox("There is no record.",MB_OK|MB_ICONINFORMATION);
      m_strshow=(_T(""));
      UpdateData(FALSE);}
}

void CLicenseView::OnBnClickedNextRecord()
{     m_bFlag = FALSE;
      POSITION pos;
      pos = m_pCurrent;
      if (pos != NULL)
      {     m_emList->GetNext(pos);
            if (pos)
            {     m_pCurrent = pos;
                  ShowRecord(m_pCurrent);
                  m_strshow=(_T(""));
            UpdateData(FALSE);
            }
            else
            {     AfxMessageBox("Last record
already.",MB_OK|MB_ICONINFORMATION);
            m_strshow=(_T(""));
            UpdateData(FALSE);}
      }
      else  {AfxMessageBox("There is no
record.",MB_OK|MB_ICONINFORMATION);
      m_strshow=(_T(""));
      UpdateData(FALSE);}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void CLicenseView::OnBnClickedLastRecord()
{     m_bFlag = FALSE;
      if (!m_emList->IsEmpty())
      {     m_pCurrent = m_emList->GetTailPosition();
            ShowRecord(m_pCurrent);
            m_strshow=(_T(""));
            UpdateData(FALSE);
      }
      else
      {AfxMessageBox("There is no record.",
MB_OK|MB_ICONINFORMATION);
      m_strshow=(_T(""));
      UpdateData(FALSE);}
}

void CLicenseView::OnUpdate(CView* /*pSender*/, LPARAM /*lHint*/,
CObject* /*pHint*/)
{
      // TODO: Add your specialized code here and/or call the base
class
      m_emList = GetDocument()->GetLicenseList();
      m_pCurrent = m_emList->GetHeadPosition();
      ShowRecord(m_pCurrent);

//    clientDC.BitBlt(600,395,size.cx,size.cy,&memDC,0,0,SRCCOPY);
}

void CLicenseView::ShowRecord(POSITION position)
{     if (position)
      { CLicenses *pLicenses= (CLicenses *) m_emList->GetAt
(position);
      m_strcard = pLicenses->GetCard();
      m_strcar = pLicenses ->GetCar();
      m_strpw= pLicenses->GetPw();
      }
else
{     ClearRecord();
}
UpdateData(FALSE);
}

void CLicenseView::ClearRecord(void)
{
      m_strcar=(_T(""));
      m_strcard=(_T(""));
      m_strpw=(_T(""));
}

void CLicenseView::OnBnClickedRecv()
{           CString ab;
      CString ba;
      UpdateData(TRUE);
      ab=m_find;
      ba=m_find2;
            BYTE bb='R';
            DWORD iBytesWritten=0;
            if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
            exit(2);
      for ( i=0;i<=15;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
      ReadFile(hComp,&chBuff,1,&nReadPort,NULL);
            if (chBuff=='?')
                  break;
      sum[i]=chBuff;
            }

      buff=sum;
if (buff=="99999999_")
{                 AfxMessageBox("Wrong
Card.",MB_OK|MB_ICONINFORMATION);}
      else if(buff2=="99999998")
AfxMessageBox("Reading Card error.",MB_OK|MB_ICONINFORMATION);

      else{
                  m_strcard=buff;
                  m_strshow=(_T(""));
                  m_strcar=(_T(""));
                  m_find=ab;
                  m_find2=ba;
                  UpdateData(FALSE);}
      }

void CLicenseView::InitialSerial(CString com)
{     m_strcomport=com;
hComp = CreateFile(m_strcomport, GENERIC_READ | GENERIC_WRITE, 0,
                                          NULL, OPEN_EXISTING, 0,NULL
);


if (hComp == INVALID_HANDLE_VALUE) {
   AfxMessageBox("Can not use "+m_strcomport);
   AfxMessageBox("Please setup ComPort again!!!");
   return;
   }

valReturn = GetCommState(hComp, &dcb);

if (!valReturn) {
      AfxMessageBox("Can not use "+m_strcomport);

   return;
}

  dcb.BaudRate = CBR_9600;
  dcb.ByteSize = 8;
  dcb.Parity = NOPARITY;
  dcb.StopBits = ONESTOPBIT;

  valReturn = SetCommState(hComp, &dcb);

  if (!valReturn) {
     m_strshow="Com Port Setting Error";
     UpdateData(FALSE);
       return;
  }

}

void CLicenseView::OnBnClickedClear()
{     ClearScreen();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	CString bb;
	CString cc;
	UpdateData(TRUE);
	bb=m_find;
	cc=m_find2;
	m_strcard=(_T(""));
	m_strcar=(_T(""));
	m_strshow=(_T(""));
	m_strpw=(_T(""));
	m_find=bb;
	m_find2=cc;
	UpdateData(FALSE);

}

void CLicenseView::Add(void)
{ClicenseDoc* pDoc = GetDocument();
	ASSERT_VALID(pDoc);
	UpdateData(TRUE);
	if (m_strcard.IsEmpty())
	{	AfxMessageBox("Please enter the card number!");
		return;
	}
	CLicenses * pLicenses = new CLicenses;
		pLicenses->SetCar(m_strcar);
		pLicenses->SetCard(m_strcard);
		pLicenses->SetPw(m_strpw);
	if (m_bFlag)
	{
	if (!m_emList->IsEmpty())
		{ m_pCurrent = m_emList->GetTailPosition();
	 m_pCurrent = m_emList->InsertAfter(m_pCurrent, pLicenses);
		}
	else
		{ m_pCurrent = m_emList->InsertBefore(m_pCurrent,
pLicenses);
		}
	}
	else
	{	BOOL btailStatus = FALSE;
		POSITION pos;
		CObject * pE = 0;
		pos = m_pCurrent;
		if (m_pCurrent !=NULL)
		{	m_emList->GetNext(pos);
			if (pos == NULL)
			{	btailStatus = TRUE;
				pos = m_emList->GetHeadPosition();
				if (pos == m_pCurrent)
					pos = NULL;
			}
			pE = m_emList->GetAt(m_pCurrent);
			m_emList->RemoveAt(m_pCurrent);
			delete pE;
			m_pCurrent = pos;
	if (pos == m_emList->GetHeadPosition() && btailStatus)
			{	m_pCurrent = m_emList->GetTailPosition();
		m_pCurrent = m_emList->InsertAfter(m_pCurrent,pLicenses);
			}
 else  m_pCurrent = m_emList->InsertBefore(m_pCurrent,pLicenses);
		}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        pDoc->SetModifiedFlag();
        pDoc->UpdateAllViews(this);
}

void CLicenseView::OnBnClickedFind()
{       UpdateData(TRUE);
if ((m_find.IsEmpty())&&(m_find2.IsEmpty()))
{       AfxMessageBox("Please enter data to  find.");
        return;
};
if ((!m_find.IsEmpty())&&(!m_find2.IsEmpty()))
{       AfxMessageBox("Please do not enter 2 data at the same time.");
        return;
};
        m_bFlag=FALSE;
        if (!m_emList->IsEmpty())
        {       m_pCurrent = m_emList->GetHeadPosition();
CLicenses *pLicenses = (CLicenses *) m_emList->GetAt(m_pCurrent);
                m_strcardfound=pLicenses->GetCard();
                m_strcarfound=pLicenses->GetCar();
                if ((m_strcardfound==m_find)||(m_strcarfound==m_find2))
                {       m_strcard=m_strcardfound;
                        m_strcar=m_strcarfound;
                        m_strpw=pLicenses->GetPw();
                        if (m_find==(_T("")))
        m_strshow="Search for Car:( "+m_find2+ " ) complete.";
        if (m_find2==(_T("")))
        m_strshow="Search for Card Number:( "+m_find+ " ) complete.";
                UpdateData(FALSE);
                }
                else
                {

        for (long int l=2;l<=(m_emList->GetSize());l++)
                                {
                                m_emList->GetNext(m_pCurrent);
                                if (m_pCurrent)
 {CLicenses *pLicenses = (CLicenses *) m_emList->GetAt(m_pCurrent);
                m_strcardfound=pLicenses->GetCard();
                m_strcarfound=pLicenses->GetCar();
        if ((m_strcardfound==m_find)||(m_strcarfound==m_find2))
                {m_strcard=m_strcardfound;
                m_strcar=m_strcarfound;
                m_strpw=pLicenses->GetPw();
                if (m_find==(_T("")))
                m_strshow="Search for Car:( "+m_find2+ " ) complete.";
                if (m_find2==(_T("")))
        m_strshow="Search for Card Number:( "+m_find+ " ) complete.";
                        UpdateData(FALSE);
                        return;
                                                }
                                        }

                                }
        AfxMessageBox("Record not found.", MB_OK|MB_ICONINFORMATION);

                        }
        }
        else
        AfxMessageBox("There is no record.",MB_OK|MB_ICONINFORMATION);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void CLicenseView::OnBnClickedButton2()
{     UpdateData(TRUE);
      CString bb;
      CString cc;
      bb=m_strcard;
      cc=m_strcar;
      m_find=(_T(""));
      m_find2=(_T(""));
      m_strcard=bb;
      m_strcar=cc;
      UpdateData(FALSE);
}

void CLicenseView::OnBnClickedRecv2()
{
      CString ab;
      CString ba;
      CString bab;
      UpdateData(TRUE);
      ab=m_strcard;
      ba=m_strcar;
      bab=m_strshow;
            BYTE bb='R';
            DWORD iBytesWritten=0;
            if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
            exit(2);
if (dcb.fInX=TRUE)
      {for ( i=0;i<=15;i++)
            {
      ReadFile(hComp,&chBuff,1,&nReadPort,NULL);
      sum[i]=chBuff;     }                    }
      buff2=sum;
                  if (buff2=="99999999_")
            {     AfxMessageBox("Wrong
Card.",MB_OK|MB_ICONINFORMATION);}


                  else if(buff2=="99999998")
AfxMessageBox("Card reading error.",MB_OK|MB_ICONINFORMATION);

else{m_strcard=ab;
                  m_strshow=(_T(""));
                  m_strcar=ba;
                  m_find=buff2;
                  m_find2=(_T(""));
                  UpdateData(FALSE);}
      }


UINT CaptureThread(LPVOID pParam)
{
      // Capture Thread
      CLicenseView *dlg;            // reference to parent dialog
      dlg = (CLicenseView *)pParam;
      while (dlg->m_Running)
      {
                  if (dlg->m_Video.Capture(dlg->m_Image))
                  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    dlg->m_Video.Show(dlg->m_Image);
                        }
            }
                  Sleep(5);
            return(0);
}
void CLicenseView::OnBnClickedWork()
{     ClearScreen();
      WORK2();
}

void CLicenseView::OnClose()
{
STOP();
}

void CLicenseView::STOP(void)
{if (m_Capture.EnableWindow(FALSE))
            flag1=TRUE;
      else flag1=FALSE;
      m_Capture.EnableWindow(TRUE);
      m_ButtonAdd.EnableWindow(TRUE);
      m_ButtonDelete.EnableWindow(TRUE);
      m_ButtonRecv.EnableWindow(TRUE);
      m_clear.EnableWindow(TRUE);
      m_first.EnableWindow(TRUE);
      m_next.EnableWindow(TRUE);
      m_prev.EnableWindow(TRUE);
      m_last.EnableWindow(TRUE);
      m_search.EnableWindow(TRUE);
      m_recv2.EnableWindow(TRUE);
      m_clear2.EnableWindow(TRUE);
      ClearRecord();
      m_strshow=(_T(""));
      UpdateData(FALSE);
      STOP2();
}

void CLicenseView::WORK(void)
{     MessageBox("กรุณารูดบัตรเพื่อเปิดโปรแกรม", "License");

for ( i=0;i<=15;i++)
            {
      ReadFile(hComp,&chBuff,1,&nReadPort,NULL);
            if (chBuff=='?')
                  break;
      sum[i]=chBuff;
            }
            CString sum2;
            sum2=sum;
            if (sum2!="6627993002416994_")
      {
            MessageBox("ท่านใช้บัตรผิด กรุณาเปิดโปรแกรมใหม่อีก
ครั้ง ","License");
      exit(2);
      }

}

void CLicenseView::SAVE(void)
{     char dbuffer [9];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   char tbuffer [9];
   _strdate( dbuffer );
   _strtime( tbuffer );
CString path,dot;
path=Set.GetPath();
if (path.GetAt(path.GetLength()-1)=='\\')
      path.Delete(path.GetLength()-1);
strcpy(savename,path);
strcat(savename,"\\DATE_");
strcat(savename,dbuffer);
strcat(savename," TIME_");
strcat(savename,tbuffer);
strcat(savename,".jpg");

for (int i=2;i<=50;i++)
{     if ((savename[i]=='/') | (savename[i]==':'))
      savename[i]='_';
}
            if (m_Image != 0)
      {     m_Video.SaveJPEG(m_Image, 100,savename);
      }
      else
            MessageBox("Nothing to save. Press Capture first.",
"License", MB_OK);
}

void CLicenseView::WORK2(void)
{

      STOP2();
      m_Capture.EnableWindow(FALSE);
      m_ButtonAdd.EnableWindow(FALSE);
      m_ButtonDelete.EnableWindow(FALSE);
      m_ButtonRecv.EnableWindow(FALSE);
      m_clear.EnableWindow(FALSE);
      m_first.EnableWindow(FALSE);
      m_next.EnableWindow(FALSE);
      m_prev.EnableWindow(FALSE);
      m_last.EnableWindow(FALSE);
      m_search.EnableWindow(FALSE);
      m_recv2.EnableWindow(FALSE);
      m_clear2.EnableWindow(FALSE);

      if (m_Image != 0)
            return;

      if (!m_Video.Init())
      {
            STOP();

            MessageBox(m_Video.GetLastErrorString(), "", MB_OK);
            return;
      }

            m_Image = m_Video.GetColorImageHandle();


 if (!m_Video.SetPreview(TRUE))
      {
            MessageBox(m_Video.GetLastErrorString(), "", MB_OK);
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if (m_Video.Start())                                    // start
capture mode
      {
            m_Running = TRUE;
            AfxBeginThread(CaptureThread, this);      // start
capture thread
      }
      else
      {
            MessageBox(m_Video.GetLastErrorString(), "", MB_OK);
      }
for ( i=0;i<=15;i++)
            {
      ReadFile(hComp,&chBuff,1,&nReadPort,NULL);
            if (chBuff=='?')
                  break;
      sum[i]=chBuff;
            }           ;
CString sum2;
            sum2=sum;

      if (sum2=="6627993002416994_")
                  {STOP();
                  return;
                  }
            else
            {
      m_bFlag=FALSE;
      SAVE();
      if (!m_emList->IsEmpty())
      //เรคอร์ดที่หนึ่ง
      {     m_pCurrent = m_emList->GetHeadPosition();
CLicenses *pLicenses = (CLicenses *) m_emList->GetAt(m_pCurrent);
            m_strcardfound=pLicenses->GetCard();
            m_strcarfound=pLicenses->GetCar();
            m_strpwfound=pLicenses->GetPw();
            char test='P';
            if (m_strcardfound==sum2)//พบการ์ด
            {           m_strshow="Card Found";
                        UpdateData(FALSE);
                  Detect();

                  if (out_11.GetLength()==0)//หาทะเบียนไม่เจอ
                  {     cca=FALSE;
                  }
                  else//หาทะเบียนเจอ
                  for (int i=0;i<=out_11.GetLength();i++)
{     if ((out_11.GetAt(i)=='-') || (out_11.GetAt(i)==' '))
            {out_11.Delete(i,1);
                                          i=i-1;
                                          }
                                    }

                                    int l_str1;
                  l_str1=m_strcarfound.GetLength();
                  for (int i=0;i<=(l_str1-1);i++)
                  {     if (out_11.GetAt(i)!=m_strcarfound.GetAt
(i))//ถ้าทะเบียนผิด cca=false
                  {           for (int j=0;j<=(l_str1-1);j++)
                                    {           if (out_11.GetAt
(j+1)!=m_strcarfound.GetAt(j))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                                    {
        cca=FALSE;

        break;
                                                    }
                                        }
                                break;
                               }
                        }
                        }
                               if (cca==TRUE)//ถ้า ทะเ บีย นถูก ต้อ ง
                               {      m_strshow="Pass";
                                      m_hu=m_strcarfound;
                                      UpdateData(FALSE);
                                      pic="IDB_BITMAP3";
                                      Draw1(pic);
                                      pic2="IDB_BITMAP2";
                                      Draw2(pic2);
                                      BYTE bb='P';
                                      DWORD iBytesWritten=0;

        if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
                                      exit(2);
                                      Sleep( (clock_t)2.75 * CLOCKS_PER_SEC
);
                                      ClearScreen();
                                      pic="IDB_BITMAP1";
                                      Draw1(pic);
                                      pic2="IDB_BITMAP3";
                                      Draw2(pic2);
                               }
                               else //ถ้า ทะเ บีย นไม่ถูก ต้อ ง
                               {
                                      cca=TRUE;
                                      m_hu=out_11;
                               m_strshow="Enter Password";
                               UpdateData(FALSE);
                               KEY();
                                      CString pwbuff;
                               pwbuff=summ;

                               for (int i=0;i<=pwbuff.GetLength();i++)
                               {      if (pwbuff.GetAt(i)=='*')
                                      {                flag2=true;
                                                       break;
                                      }
                               }
                               if (flag2==true)
                               {      m_strshow="Password Cancel";
                                      UpdateData(FALSE);
                                      flag2=false;
                                      Sleep( (clock_t)1 * CLOCKS_PER_SEC );
                                      ClearScreen();
                               }
                               else if (m_strpwfound==pwbuff)
                               {      BYTE bb='C';
                                      DWORD iBytesWritten=0;

        if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
                                             exit(2);
                                      m_strshow="Password Correct";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    UpdateData(FALSE);
                                    Sleep( (clock_t)1.75 * CLOCKS_PER_SEC
);
                                    m_strshow="Pass";
                                    UpdateData(FALSE);
                                    pic="IDB_BITMAP3";
                                    Draw1(pic);
                                    pic2="IDB_BITMAP2";
                                    Draw2(pic2);

                                    Sleep( (clock_t)1 * CLOCKS_PER_SEC );
                                    ClearScreen();
                                    pic="IDB_BITMAP1";
                                    Draw1(pic);
                                    pic2="IDB_BITMAP3";
                                    Draw2(pic2);
                               }
                               else
                               {    m_strshow="Wrong Password";
                                    UpdateData(FALSE);
                                    BYTE bb='I';
                                    DWORD iBytesWritten=0;

     if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
                                         exit(2);
                                    Sleep( (clock_t)1 * CLOCKS_PER_SEC );
                                    ClearScreen();

                               }}
                    }
                    else //เรคอร์ดที่สองขึ้นไป
                         {    POSITION m_pCurrent2=m_pCurrent;
                              long ro;
                              long siz=m_emList->GetSize();
                              for (ro=2;ro<=siz;ro++)
                              { int abcd1=ro;
                              POSITION m_pCurrent1=m_pCurrent2+0x0000000C;
                              m_pCurrent2=m_pCurrent1;
                              if (m_pCurrent1)
{CLicenses *pLicenses = (CLicenses *) m_emList->GetAt(m_pCurrent1);
                              m_strcardfound=pLicenses->GetCard();
                              m_strcarfound=pLicenses->GetCar();
                              m_strpwfound=pLicenses->GetPw();
                    if (m_strcardfound==sum2)//พบการ์ด
                                        {    m_strshow="Card Found";
                                             UpdateData(FALSE);

                                             Detect();
                              if (out_11.GetLength()==0)//หาทะเบียนไม่เจอ
                              {    cca=FALSE;
                              }
                              else//หาทะเบียนเจอ
                         for (int i=0;i<=out_11.GetLength();i++)
               {if ((out_11.GetAt(i)=='-') || (out_11.GetAt(i)==' '))
                                             {out_11.Delete(i,1);
                                             i=i-1;
                                             }
                                        }

                              int l_str1;
                              l_str1=m_strcarfound.GetLength();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    for (int i=0;i<=(l_str1-1);i++)
{if (out_11.GetAt(i)!=m_strcarfound.GetAt(i))//ถ้า ทะเบียนผิด cca=false
                        {           for (int j=0;j<=(l_str1-1);j++)
            if (out_11.GetAt(j+1)!=m_strcarfound.GetAt(j))
                                                {     cca=FALSE;
                                                      break;
                                                }
                                    }
                         break;
                        }
                    }
                    }

                    if (cca==TRUE)//ถ้า ทะเบียนถูกต้อง
                                                {     m_strshow="Pass";
                                                      m_hu=m_strcarfound;
                                                      UpdateData(FALSE);
                                                      pic="IDB_BITMAP3";
                                    Draw1(pic);
                                    pic2="IDB_BITMAP2";
                                    Draw2(pic2);
                                                      BYTE bb='P';
                                DWORD iBytesWritten=0;
      if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
                                                      exit(2);
      Sleep( (clock_t)2.75 * CLOCKS_PER_SEC );
                                    ClearScreen();
                                    pic="IDB_BITMAP1";
                                    Draw1(pic);
                                    pic2="IDB_BITMAP3";
                                    Draw2(pic2);
                                break;//เรคอร์ดสองขึ้นไปต้องมีเบรคเพราะวนloop
                                                }
                                                else //ถ้า ทะเบียนไม่ถูกต้อง
                                                {
                                                      cca=TRUE;
                                                      m_hu=out_11;
                                                m_strshow="Enter Password";
                                                      UpdateData(FALSE);
                                                            KEY();
                                                            CString pwbuff;
                                                            pwbuff=summ;
      for (int i=0;i<=pwbuff.GetLength();i++)
{if (pwbuff.GetAt(i)=='*')

      {flag2=true;

      break;}
                                                                  }
            if (flag2==true)

      {     m_strshow="Password Cancel";

      UpdateData(FALSE);
      flag2=false;
      Sleep( (clock_t)1 * CLOCKS_PER_SEC );
      ClearScreen();
      break;
                                                                  }
      else  if (m_strpwfound==pwbuff)
{     BYTE bb='C';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DWORD iBytesWritten=0;
        if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
        exit(2);
        m_strshow="Password Correct";
        UpdateData(FALSE);
        Sleep( (clock_t)1.75 * CLOCKS_PER_SEC );
        m_strshow="Pass";
        UpdateData(FALSE);
        pic="IDB_BITMAP3";
                                Draw1(pic);
                                pic2="IDB_BITMAP2";
                                Draw2(pic2);
              Sleep( (clock_t)1 * CLOCKS_PER_SEC );
                                ClearScreen();
                                pic="IDB_BITMAP1";
                                Draw1(pic);
                                pic2="IDB_BITMAP3";
                                Draw2(pic2);
                                        break;
                                            }
                                                    else
                                                    {
        m_strshow="Wrong Password";
        UpdateData(FALSE);
        BYTE bb='I';
              DWORD iBytesWritten=0;
        if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
                    exit(2);
        Sleep( (clock_t)1 * CLOCKS_PER_SEC );
                          ClearScreen();
                                                                    break;
                                                                    }}
                                                }
                                            else //ไม่พบการ์ด
                                            {
        if (ro==m_emList->GetSize())
                    {
        BYTE bb='W';
              DWORD iBytesWritten=0;
        if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
                                        exit(2);

              m_strshow="Card not found";
              UpdateData(FALSE);
              Sleep( (clock_t)3 * CLOCKS_PER_SEC );
              ClearScreen();
                                                    }
                                            }
                                    }

                            }
                    }
}
        else//ไม่พบการ์ด
        AfxMessageBox("There is no record.",MB_OK|MB_ICONINFORMATION);
              }
WORK2();
}

void CLicenseView::STOP2(void)
{     if (m_Running)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	{	m_Running = FALSE;	// tell thread to end
	Sleep(100);		// wait for thread to end

	if (!m_Video.Stop())
	{
		MessageBox(m_Video.GetLastErrorString(), "", MB_OK);
	}}
	if (!m_Video.Close())
	{
		MessageBox(m_Video.GetLastErrorString(), "", MB_OK);
	}
	m_Video.ReleaseImageHandle(m_Image);
	m_Image = 0;
}

void CLicenseView::KEY(void)
{ BYTE bb='U';
	DWORD iBytesWritten=0;
	if(WriteFile(hComp,&bb,1,&iBytesWritten,NULL)==0)
	exit(2);
	for ( i=0;i<=15;i++)
		{
	ReadFile(hComp,&chBuff,1,&nReadPort,NULL);
		if (chBuff=='#')
			break;
	summ[i]=chBuff;
		if (chBuff=='*')
			break;

		}

}

void CLicenseView::Detect(void)
{CPImage image;
CPImage pImage;
CString savename1;
savename1=savename;
for (int i=0;i<=50;i++)
{	if (savename[i]=='\\')
	savename1.Insert(i+1,'\\');
}
pImage.Load(savename1);
	  width=pImage.GetWidth();
		heigh=pImage.GetHeight();
		BPP=pImage.GetBPP();

    {
		CPImage image1,image2;
		VERIFY(image1.CreateEx(641,481,24,BI_RGB));
		int aswq=1;
		double sub[7][7]={{3/21.0,3/21.0,3/21.0,0.0,-3/21.0,-
3/21.0,-3/21.0},

{3/21.0,3/21.0,3/21.0,0.0,-3/21.0,-3/21.0,-3/21.0},

	{3/21.0,3/21.0,3/21.0,0.0,-3/21.0,-3/21.0,-3/21.0},

	{3/21.0,3/21.0,3/21.0,0.0,-3/21.0,-3/21.0,-3/21.0},

	{3/21.0,3/21.0,3/21.0,0.0,-3/21.0,-3/21.0,-3/21.0},
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      {3/21.0,3/21.0,3/21.0,0.0,-3/21.0,-3/21.0,-3/21.0},

      {3/21.0,3/21.0,3/21.0,0.0,-3/21.0,-3/21.0,-3/21.0}};

                  double subc[7][7]={{-3/21.0,-3/21.0,-3/21.0,-
3/21.0,-3/21.0,-3/21.0,-3/21.0},
                                                {-3/21.0,-3/21.0,-
3/21.0,-3/21.0,-3/21.0,-3/21.0,-3/21.0},
                                                   {-3/21.0,-3/21.0,-
3/21.0,-3/21.0,-3/21.0,-3/21.0,-3/21.0},

      {0.0,0.0,0.0,0.0,0.0,0.0,0.0},

      {3/21.0,3/21.0,3/21.0,3/21.0,3/21.0,3/21.0,3/21.0},

{3/21.0,3/21.0,3/21.0,3/21.0,3/21.0,3/21.0,3/21.0},

      {3/21.0,3/21.0,3/21.0,3/21.0,3/21.0,3/21.0,3/21.0}};

            double buf_intdata2;
            double buf_intdata3;
            double buf_intdata4;
            BYTE *pSrc1;
            BYTE *pSrc;
            BYTE picc[640][480];
            BYTE bb[640][480];
            for (int i=0;i<heigh;i++)
                  {
                        for (int j=0;j<width;j++)
                        {  pSrc1 =(BYTE*)pImage.GetPixelAddress(j,i);
                        BYTE kkk0=pSrc1[0];
                        BYTE kkk1=pSrc1[1];
                        BYTE kkk2=pSrc1[2];
                        pSrc=(BYTE*)image1.GetPixelAddress(j+1,i+1);
                        pSrc[0]=kkk0;
                        pSrc[1]=kkk1;
                        pSrc[2]=kkk2;
                        }
                  }
             for (int i=1;i<heigh;i++)
                  {
                        for (int j=1;j<width;j++)
                        {
                  pSrc =(BYTE*)image1.GetPixelAddress(j,i);
                        if (pSrc[2]>100)
                              {picc[j][i]=255;
                              }
                              else {
                                    picc[j][i]=0;
                              }
                              bb[j][i]=pSrc[2];
                        }
                  }

                  buf_intdata2=0;
                        buf_intdata3=0;

                  BYTE c[640-6][480-6];
                  double fff;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    for (int i=1;i<heigh-7;i++)
                   {
                         for (int j=1;j<width-7;j++)
                         {
                          for (int a=0;a<7;a++)
                               {
                                     for (int b=0;b<7;b++)
          {  fff=(double)picc[j+a][i+b];
          buf_intdata2=buf_intdata2+((double)(fff*sub[a][b]));
          buf_intdata3=buf_intdata3+((double)(fff*subc[a][b]));

                                     }
                               }

     buf_intdata2=(double)sqrt(buf_intdata2*buf_intdata2);
     buf_intdata3=(double)sqrt(buf_intdata3*buf_intdata3);
     buf_intdata4=(buf_intdata3+buf_intdata2);
     if (buf_intdata4>=100)  {c[j+1][i+1]=255;}
     if (buf_intdata4<100){c[j+1][i+1]=0;}
                               buf_intdata2=0;
                          buf_intdata3=0;
                         }
                   }
                   //delete [] picc;
                         //แก้   error
                    for (int i=1;i<heigh-7;i++)
                   {
                         for (int j=1;j<width-7;j++)
                         {
                          int s3=0;
     for (int s1=-1;s1<2;s1++)
     {
          for (int s2=-1;s2<2;s2++)
          { if (c[j][i]==0)
          { if (c[j+s2][i+s1]==255)s3=s3+1;

                                                  }
                                                       }
                                                  }
          if(s3>4)c[j][i]=255;
                                   }
                         } //จบแก้  error
       //delete picc;

     int xx[320];
      int xa[260];
      int xj[320];
      char out_1[30];
     out_1[1]='e';

                   // หาแผ่น ป้า ย ทะ เบีย น**************************
                   //************************************

int z=0;
                   for (int j=320;j<width-6-125;j++)
                   { int a=0;
                         for (int i=1;i<heigh-7;i++)
                         { if (c[j][i]==0)a=0;
                               if (c[j][i]==255)a=a+1;
                               if (a>70)
                               {xx[z]=j;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                xa[z]=i;
                xj[z]=a;
                z=z+1;
                if (z>237){break;break;}
                }
        }
}

//*********************************
if (z==0)
{for (int j=320;j<width-6-125;j++)
{ int a=0;
        for (int i=1;i<heigh-7;i++)
        { if (c[j][i]==0)a=0;
                if (c[j][i]==255)a=a+1;
                if (a>50)
                {xx[z]=j;
                xa[z]=i;
                xj[z]=a;
                z=z+1;
                if (z>237){break;break;}
                }
        }
}

}
//*********************************
if (z==0)
{for (int j=320;j<width-6-125;j++)
{ int a=0;
        for (int i=1;i<heigh-7;i++)
        { if (c[j][i]==0)a=0;
                if (c[j][i]==255)a=a+1;
                if (a>40)
                {xx[z]=j;
                xa[z]=i;
                xj[z]=a;
                z=z+1;
                if (z>237){break;break;}
                }
        }
}

}

//*********************************
if (z==0)
{for (int j=320;j<width-6-125;j++)
{ int a=0;
        for (int i=1;i<heigh-7;i++)
        { if (c[j][i]==0)a=0;
                if (c[j][i]==255)a=a+1;
                if (a>35)
                {xx[z]=j;
                xa[z]=i;
                xj[z]=a;
                z=z+1;
                if (z>237){break;break;}
                }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
//********************************** หา xmax
for (int aa=z;aa<320;aa++){ xx[aa]=680;}

int xmax=680;
//if(sd==320){goto data_error;}
int sx=10;
for (int sd=0;sd<320;sd++)
{if (xmax>xx[sd]&& xx[sd]!=NULL)
      { xmax=xx[sd];sx=sd;}
}
if (sx==10){goto data_error;}///
int xmean;
xmean=xa[sx];
int x1;x1=(xj[sx]/2);
xmean=xmean-x1;
//int x1=xj[i]/2;

//หา  xmin**************************
int xxx[321];
z=0;
int a;
for(int j=125;j<320;j++)
{ a=0;
      for (int i=1;i<heigh-7;i++)
      { if (c[j][i]==0)a=0;
        if(c[j][i]==255)a=a+1;
              if(a==70)
              {xxx[z]=j;
              z=z+1;
               if (z>237){break;break;}
              }
      }
}

//**********************************
if (z==0){
z=0;
for(int j=125;j<320;j++)
{ a=0;
      for (int i=1;i<heigh-7;i++)
      { if (c[j][i]==0)a=0;
        if(c[j][i]==255)a=a+1;
              if(a==60){xxx[z]=j;z=z+1;
              if (z>237){break;break;}}
      }
}
}
//**********************************
if (z==0){
z=0;
for(int j=125;j<320;j++)
{ a=0;
      for (int i=1;i<heigh-7;i++)
      { if (c[j][i]==0)a=0;
        if(c[j][i]==255)a=a+1;
              if(a==50){xxx[z]=j;z=z+1;
              if (z>237){break;break;}}
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        //**********************************
        if (z==0){
        z=0;
        for(int j=125;j<320;j++)
        { a=0;
              for (int i=1;i<heigh-7;i++)
              { if (c[j][i]==0)a=0;
                if(c[j][i]==255)a=a+1;
                      if(a==40){xxx[z]=j;z=z+1;
                      if (z>237){break;break;}}
              }
        }
        }
        //**********************************
if (z==0){
        z=0;
        for(int j=125;j<320;j++)
        { a=0;
              for (int i=1;i<heigh-7;i++)
              { if (c[j][i]==0)a=0;
                if(c[j][i]==255)a=a+1;
                      if(a==35){xxx[z]=j;z=z+1;
                      if (z>237){break;break;}}
              }
        }
        }
        //********* หา xmin ***************************


        int xmin=0;
        for (int sd=0;sd<321;sd++)
        {if (xmin<xxx[sd]&& xxx[sd]!=NULL)
              { xmin=xxx[sd];}
        }
        //*****************************************
        //********* หา  ymax************************
        int yy[480];
         z=0;
        for (int i=xmean;i<heigh-7;i++)
        { a=0;
              for (int j=xmin;j<xmax;j++)
              {  if (c[j][i]==0)a=0;
                    if (c[j][i]==255)a=a+1;
                    if (a==120)
                    {yy[z]=i;
                    z=z+1;
                    if(z>470){break;break;}
                    }
              }
        }
        //**************************************
        if (z==0){
        for (int i=xmean;i<heigh-7;i++)
        { a=0;
              for (int j=xmin;j<xmax;j++)
              {  if (c[j][i]==0)a=0;
                    if (c[j][i]==255)a=a+1;
                    if (a==110)
                    {yy[z]=i;
                    z=z+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if(z>470){break;break;}
                }
        }
}
}
//***********************************
if (z==0){
for (int i=xmean;i<heigh-7;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==100)
            {yy[z]=i;
            z=z+1;
            if(z>470){break;break;}}
      }}}
//***********************************
if (z==0){
for (int i=xmean;i<heigh-7;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==90)
            {yy[z]=i;
            z=z+1;
            if(z>470){break;break;}}
      }}}
//***********************************
if (z==0){
for (int i=xmean;i<heigh-7;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==80)
            {yy[z]=i;
            z=z+1;
            if(z>470){break;break;}}
      }}}
//***********************************
if (z==0){
for (int i=xmean;i<heigh-7;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==70)
            {yy[z]=i;
            z=z+1;
            if(z>470){break;break;}}
      }}}
//***********************************
if (z==0){
for (int i=xmean;i<heigh-7;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==60)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        {yy[z]=i;
                        z=z+1;
                        if(z>470){break;break;}}
                  }}}
            //**********************************
            if (z==0){
            for (int i=xmean;i<heigh-7;i++)
            { a=0;
                  for (int j=xmin;j<xmax;j++)
                  {  if (c[j][i]==0)a=0;
                        if (c[j][i]==255)a=a+1;
                        if (a==50)
                        {yy[z]=i;
                        z=z+1;
                        if(z>470){break;break;}}
                  }}}
            //**********************************
            //**  หา ค่า  ymax ***************
            for (int aa=z;aa<480;aa++){ yy[aa]=680;}
            int ymax=680;
            for (int sd=0;sd<480;sd++)
            {if (ymax>yy[sd]&& yy[sd]!=NULL)
                  { ymax=yy[sd];}
            }
            //**************************************
            // ***** หา ymin*********************
int yyy[480];
             z=0;
            for (int i=1;i<xmean;i++)
            { a=0;
                  for (int j=xmin;j<xmax;j++)
                  {  if (c[j][i]==0)a=0;
                        if (c[j][i]==255)a=a+1;
                        if (a==120)
                        {yyy[z]=i;
                        z=z+1;
                        if(z>470){break;break;}
                        }}}
            //**********************************
            if (z==0){
            for (int i=1;i<xmean;i++)
            { a=0;
                  for (int j=xmin;j<xmax;j++)
                  {  if (c[j][i]==0)a=0;
                        if (c[j][i]==255)a=a+1;
                        if (a==110)
                        {yyy[z]=i;
                        z=z+1;
                        if(z>470){break;break;}
                        }}}}
            //**********************************
            if (z==0){
            for (int i=1;i<xmean;i++)
            { a=0;
                  for (int j=xmin;j<xmax;j++)
                  {  if (c[j][i]==0)a=0;
                        if (c[j][i]==255)a=a+1;
                        if (a==100)
                        {yyy[z]=i;
                        z=z+1;
                        if(z>470){break;break;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }}}}
//**********************************
if (z==0){
for (int i=1;i<xmean;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==90)
            {yyy[z]=i;
            z=z+1;
            if(z>470){break;break;}
            }}}}
//**********************************
if (z==0){
for (int i=1;i<xmean;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==80)
            {yyy[z]=i;
            z=z+1;
            if(z>470){break;break;}
            }}}}
//**********************************
if (z==0){
for (int i=1;i<xmean;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==70)
            {yyy[z]=i;
            z=z+1;
            if(z>470){break;break;}
            }}}}
//**********************************
if (z==0){
for (int i=1;i<xmean;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==60)
            {yyy[z]=i;
            z=z+1;
            if(z>470){break;break;}
            }}}}
//**********************************
if (z==0){
for (int i=1;i<xmean;i++)
{ a=0;
      for (int j=xmin;j<xmax;j++)
      {  if (c[j][i]==0)a=0;
            if (c[j][i]==255)a=a+1;
            if (a==50)
            {yyy[z]=i;
            z=z+1;
            if(z>470){break;break;}
            }}}}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                //*********************************
                //**** หา  ymin ***************
                int ymin=0;
                for (int sd=0;sd<480;sd++)
                {if (ymin<yyy[sd]&& yyy[sd]!=NULL)
                      { ymin=yyy[sd];}
                }
                //*******************************

      if (xmax==NULL | xmin==NULL |ymax==NULL|ymin==NULL) {int aswq;
                aswq=1;
                m_strshow="Wrong License";
                UpdateData(FALSE);
                xa[1]=100;
                return;
                goto data_error;}
      if (xmax>630 | xmin<10 | ymax>470 | ymin<10){int aswq;
                aswq=1;
                xa[1]=100;
                m_strshow="Wrong License";
                UpdateData(FALSE);
                return;
                goto data_error;}
                // ********* หาค่า  mean threshold***************
             int thr1;
             int thr;
             thr1=0;

             for (int i=1;i<ymax-ymin-8+1;i++)
                  { for (int j=1;j<xmax-xmin-12+1;j++)
       { pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4));
                          thr1=thr1+pSrc[2];//+pSrc[1]+pSrc[0];
                     }}
                thr=(BYTE)(thr1/((ymax-ymin-8)*(xmax-xmin-12)));
                thr=thr-30;
       BYTE bbbf=0;
int m1=ymax-ymin-8;
int n1=xmax-xmin-12;
                for (int i=1;i<ymax-ymin-8+1;i++)
                { for (int j=1;j<xmax-xmin-12+1;j++)
                { if (bb[j+xmin+8][i+ymin+4]>thr){bbbf=200;}
                     if (bb[j+xmin+8][i+ymin+4]<=thr){bbbf=50;}
pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4));

                     pSrc[2]=bbbf;
                     pSrc[1]=bbbf;
                     pSrc[0]=bbbf;
                }}

                //************ แก้ค่า error จากการทำ binary ********


                int s3=0;
           for (int i=2;i<(m1);i++)
           {     for (int j=2;j<(n1);j++)
                 { s3=0;
                           for (int s1=0;s1<3;s1++)
                           {     for (int s2=0;s2<3;s2++)
       { pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4));
                                 if (pSrc[2]==50)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s2-1),(i+ymin+4+s1-
1));
                                    if (pSrc[2]==200){s3=s3+1;}
                                                    }}}
if (s3>4){pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4));
                                    pSrc[2]=200;pSrc[1]=255;pSrc[0]=255;}}}

            //***************   หาขอบ histrogram ด้า น  y *************
            int ax;
            int ass[480];
            for (int i=1;i<m1;i++)
            {     ax=0;
                  for(int j=1;j<n1;j++)
      {  pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4));
                        if (pSrc[2]==50)ax=ax+1;
                  }
                  ass[i]=ax;
            }
            int kki=0   ;
            int ss=0;
            int ymax1;
                  for (int s=int(m1/2);s<m1;s++)
                  { if (ass[s]>=10)ss=ss+1;
                    if (ass[s]<10 & ss!=0)
                    { ymax1=s;kki=1; break;}
                    if (ass[s]<10 & ss==0)
                    { ymax1=s;kki=1; break;}
                  }
                  if (kki== 0){int aswq=1;goto data_error;}

             ss=0;
           int ymin1;
 s=(m1/2)-1;
 int kki1=0;
                  for (;;s--)
                  { if (ass[s]>=10){ss=ss+1;}
                    if (ass[s]<10 & ss!=0)
                    { ymin1=s;kki1=1; break;}
                    if (ass[s]<10 & ss==0)
                    { ymin1=s; kki1=1;break;}
                    if (s==1 & ss!=0)ymin1=s;
                  }if (kki1==0){int aswq=1;goto data_error;}
                  int cdf;

                        if (ymin1<3){cdf=0;
                        for (int i=1;i<ymax1-ymin1;i++)
                              { for (int j=1;j<n1;j++)
 {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4+ymin1-
 cdf));
       if (pSrc[2]==50){pSrc[2]=0;pSrc[1]=0;pSrc[0]=0;}
             if (pSrc[2]==200){pSrc[2]=255;pSrc[0]=255;pSrc[1]=255;}
                                    }
                              }}

                        if (ymin1>=3){cdf=2;
                        for (int i=1;i<ymax1-ymin1+2;i++)
                              { for (int j=1;j<n1;j++)
 {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4+ymin1-
 cdf));
                  if (pSrc[2]==50){pSrc[2]=0;pSrc[1]=0;pSrc[0]=0;}
            if (pSrc[2]==200){pSrc[2]=255;pSrc[0]=255;pSrc[1]=255;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        }
                                    }}


 //******************************** หาตำแหน่ง ตัว อัก ษ ร โดยใช้วิธี
contour following
  int mpp;int npp;
      if (cdf=0)  mpp=ymax1-ymin1;
      if (cdf=2)  mpp=ymax1-ymin1+2;
      npp=n1;
      VERIFY(image2.CreateEx(npp,mpp,24,BI_RGB));
      BYTE *pSrc2;
if (cdf==0){
                          for (int i=1;i<ymax1-ymin1;i++)
                                { for (int j=1;j<n1;j++)
{pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4+ymin1-
cdf));
      pSrc2 =(BYTE*)image2.GetPixelAddress((j),(i));
if (pSrc[2]==50){pSrc[2]=0;pSrc[1]=0;pSrc[0]=0;pSrc2[2]=0;pSrc2
[1]=0;pSrc2[0]=0;}
if (pSrc[2]==200){pSrc[2]=255;pSrc[0]=255;pSrc[1]=255;pSrc2
[2]=255;pSrc2[0]=255;pSrc2[1]=255;}


                                    }
                                }}
if (cdf==2){
                          for (int i=1;i<ymax1-ymin1+2;i++)
                                { for (int j=1;j<n1;j++)
{pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8),(i+ymin+4+ymin1-
cdf));
pSrc2 =(BYTE*)image2.GetPixelAddress((j),(i));

if (pSrc[2]==50){pSrc[2]=0;pSrc[1]=0;pSrc[0]=0;pSrc2[2]=0;pSrc2
[1]=0;pSrc2[0]=0;}
if (pSrc[2]==200){pSrc[2]=255;pSrc[0]=255;pSrc[1]=255;pSrc2
[2]=255;pSrc2[0]=255;pSrc2[1]=255;}


                                    }
                                }}


bufpic=1;

for (int ggg=0;ggg<30;ggg++){out_1[ggg]=' ';}

int bf=1;
int bbb=1;
int aqq=1;
int cc1[10000];
int cc2[10000];
int maxxx;
int maxyy;
int minxx;
int minyy;
int sdr;sdr =(int)(0.933333*npp);
            int   a_minx[30];
            int a_miny[30];
            int a_maxx[30];
          int a_maxy[30];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            int   a_xx1[30];
            int   a_xx2[30];
            int   a_xx3[30];
            int   a_xx4[30];
while (bf<sdr)// start loop while
{int f=(mpp/2);
maxxx=0;
maxyy=0;
minxx=640;
minyy=640;
for (int b=bf;b<npp;b++)
      { pSrc =(BYTE*)image1.GetPixelAddress((b+xmin+8),
(f+ymin+4+ymin1-cdf));
            if (pSrc[2]==0 & b!=1)
            { cc1[1]=b-1;
              cc2[1]=f;
              cc1[2]=b;
              cc2[2]=f;
              break;
            }
            if (pSrc[2]==0 & b==1)
            { cc1[1]=b;
              cc2[1]=f;
              cc1[2]=b;
              cc2[2]=f-1;
              break;
            }
      }
if (b>npp-1)break;

//*****************************
//int g=2;
int xx1=0;
int xx2=0;
int xx3=0;
int xx4=0;
int sssd=0;
for(int g=2;g<9999;g++)
{     if (cc2[g]==cc2[g-1] & cc1[g]>cc1[g-1])
      {     pSrc =(BYTE*)image1.GetPixelAddress((cc1[g]+xmin+8),(cc2
[g]+ymin+4+ymin1-cdf));
            if (pSrc[2]==0 & cc2[g]!=1+cdf)
            {cc1[g+1]=cc1[g];cc2[g+1]=cc2[g]-1;
            }
            if (pSrc[2]==0 & cc2[g]==1+cdf)
            {cc1[g+1]=cc1[g]+1;cc2[g+1]=cc2[g];xx1=1;
            }
            if (pSrc[2]==0 & cc1[g]==npp-1)
            {cc1[g+1]=cc1[g];cc2[g+1]=cc2[g]+1;
            }
            if (pSrc[2]==255 )
            {cc1[g+1]=cc1[g];cc2[g+1]=cc2[g]+1;
            }
      }
      if (cc2[g]==cc2[g-1] & cc1[g]<cc1[g-1])
      {     pSrc =(BYTE*)image1.GetPixelAddress((cc1[g]+xmin+8),(cc2
[g]+ymin+4+ymin1-cdf));
            if (pSrc[2]==0 & cc2[g]!=mpp-1)
            {cc1[g+1]=cc1[g];cc2[g+1]=cc2[g]+1;
            }
            if (pSrc[2]==0 & cc2[g]==mpp-1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {cc1[g+1]=cc1[g]-1;  cc2[g+1]=cc2[g];  xx2=1;
            }
            if (pSrc[2]==0 & cc1[g]==1)
            {cc1[g+1]=cc1[g];    cc2[g+1]=cc2[g]-1;
            }
            if (pSrc[2]==255)
            { cc1[g+1]=cc1[g];  cc2[g+1]=cc2[g]-1;
            }
      }
      if ( cc1[g]==cc1[g-1] & cc2[g]>cc2[g-1])
      {     pSrc =(BYTE*)image1.GetPixelAddress((cc1[g]+xmin+8),(cc2
[g]+ymin+4+ymin1-cdf));
            if (pSrc[2]==0 & cc1[g]!=npp-1)
            { cc1[g+1]=cc1[g]+1; cc2[g+1]=cc2[g];
            }
            if (pSrc[2]==0 & cc1[g]==npp-1)
            {cc1[g+1]=cc1[g];  cc2[g+1]=cc2[g]+1; xx3=1;
            }
            if (pSrc[2]==0 & cc2[g]==mpp-1)
            {cc1[g+1]=cc1[g]-1;  cc2[g+1]=cc2[g];
            }
            if (pSrc[2]==255)
            {cc1[g+1]=cc1[g]-1;  cc2[g+1]=cc2[g];
            }
      }
      if (cc1[g]==cc1[g-1] & cc2[g]<cc2[g-1])
      {pSrc =(BYTE*)image1.GetPixelAddress((cc1[g]+xmin+8),(cc2
[g]+ymin+4+ymin1-cdf));
            if (pSrc[2]==0 & cc1[g]!=1)
            { cc1[g+1]=cc1[g]-1; cc2[g+1]=cc2[g];
            }
            if (pSrc[2]==0 & cc1[g]==1)
            {cc1[g+1]=cc1[g];  cc2[g+1]=cc2[g]-1; xx4=1;
            }
            if (pSrc[2]==0 & cc2[g]==1+cdf)
            {cc1[g+1]=cc1[g]+1;  cc2[g+1]=cc2[g];
            }
            if (pSrc[2]==255)
            {cc1[g+1]=cc1[g]+1;  cc2[g+1]=cc2[g];
            }
      }
      if( ((cc1[1]==cc1[g]) & (cc2[1]==cc2[g]) & g >20) | ((cc1
[2]==cc1[g]) & (cc2[2]==cc2[g]) & g>20)){ goto asew;}
      //g=g+1;
}

asew:g=g-2;
                  for (int sd=1;sd<g;sd++)
                  {if (maxxx<cc1[sd])
                        { maxxx=cc1[sd];}
                  }

                  for (int sd=1;sd<g;sd++)
                  {if (maxyy<cc2[sd])
                        { maxyy=cc2[sd];}
                  }

                  for (int sd=1;sd<g;sd++)
                  {if (minyy>cc2[sd]& cc2[sd]>0)
                        { minyy=cc2[sd];}
                  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    for (int sd=1;sd<g;sd++)
                    {if (minxx>cc1[sd]& cc2[sd]>0)
                          { minxx=cc1[sd];}
                    }
                    minyy=minyy+1-cdf;
                    minxx=minxx+1;
                    if (minxx==1 & xx4==1)
                          minxx=1;
                    if (minyy==1 & xx1==1)
                          minyy=1;
                    if( maxxx==npp  & xx3==1)
                          maxxx=npp+1;
                    if  (maxyy==mpp & xx2==1)
                          maxyy=mpp+1;

                    for (int aqw=20;aqw<9999;aqw++)
                    {if ((cc1[aqw]==cc1[1] & cc2[aqw]==cc2[1])|(cc1
[aqw]==cc1[2] & cc2[aqw]==cc2[2]))break;
                    }
                    int ggg1=aqw-1;

                    for (int x=1;x<ggg1;x++)
                    {if (cc1[x]>-1&cc2[x]>-1){
                          pSrc =(BYTE*)image1.GetPixelAddress((cc1
[x]+xmin+8),(cc2[x]+ymin+4+ymin1-cdf));
                          pSrc[1]=125;pSrc[0]=125;}
                    }
              a_minx[bbb]=minxx;
                  a_miny[bbb]=minyy;
                  a_maxx[bbb]=maxxx;
                  a_maxy[bbb]=maxyy;
                  a_xx1[bbb]=xx1;
                  a_xx2[bbb]=xx2;
                  a_xx3[bbb]=xx3;
                  a_xx4[bbb]=xx4;

                  bf=maxxx+1;
                  bbb=bbb+1;
                   if (bf==npp)break;
}// end loop while
             //*****************************************************

int s21=0;
int r1[640];
int mm;
int nn;
if (bbb!=0){// start if bbb!=0
for (int rr=1;rr<bbb;rr++)
      {//start loop for
            mm=a_maxy[rr]-a_miny[rr];
            nn=a_maxx[rr]-a_minx[rr];
      if(  mm>nn  & nn>5){//start if  mm>nn  & nn>5
      //*********************%แนวตั้ง แถวที่
1*********************************
int g1=(0.15*nn)+1;
int qq=1;
int s43=0;
int s35=a_minx[rr]-1;
int s36=a_miny[rr];
//s21=s21+s35;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (i=2;i<mm+1;i++)
{pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(i+ymin+4+ymin1-
cdf+s36));
 BYTE sdp;sdp=pSrc[2];
 pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(i-
1+ymin+4+ymin1-cdf+s36));
 BYTE sdp1;sdp1=pSrc[2];
 if(sdp!=sdp1)
 {r1[qq]=i-1;
 qq=qq+1;
 s43=s43+1;
 }}
pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=1;
qq=qq+1;
int assd=1;
}
pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(mm+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=mm;
qq=qq+1;
}
qq=qq-1;
int e1=0;
int e2=0;
int e3=0;

for (int ii=1;ii<qq+1;ii++)
{if (r1[ii]<(mm*0.333))e1=e1+1;
 if ((r1[ii]>=(mm*0.333))&(r1[ii]<=(mm*0.66)))e2=e2+1;
if (r1[ii]>(mm*0.66))e3=e3+1;
}
int ee1=e1+e2+e3;

//*******************แนวตั้ง แถวที่
2*************************************
 g1=(0.5*nn);
 qq=1;
s43=1;
 //s35=a_minx[rr];
 //s36=a_miny[rr];
//s21=s21+s35;
for (i=2;i<mm+1;i++)
{pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(i+ymin+4+ymin1-
cdf+s36));
 BYTE sdp;sdp=pSrc[2];
 pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(i-
1+ymin+4+ymin1-cdf+s36));
 BYTE sdp1;sdp1=pSrc[2];
 if(sdp!=sdp1)
 {r1[qq]=i-1;
 qq=qq+1;
 s43=s43+1;
 }}
pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
qq=qq+1;
//int assd=1;
}
pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(mm+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=mm;
qq=qq+1;
}
qq=qq-1;
int e4=0;
int e5=0;
int e6=0;

for (int ii=1;ii<qq+1;ii++)
{if (r1[ii]<(mm*0.333))e4=e4+1;
 if ((r1[ii]>=(mm*0.333))&(r1[ii]<=(mm*0.66)))e5=e5+1;
if (r1[ii]>(mm*0.66))e6=e6+1;
}
int ee2=e4+e5+e6;
//*********************************************************
//*******************แนวตั้ง แถวที่
3*************************************
 g1=(0.85*nn);
 qq=1;
s43=1;
 //s35=a_minx[rr];
 //s36=a_miny[rr];
//s21=s21+s35;
for (i=2;i<mm+1;i++)
{pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(i+ymin+4+ymin1-
cdf+s36));
 BYTE sdp;sdp=pSrc[2];
 pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(i-
1+ymin+4+ymin1-cdf+s36));
 BYTE sdp1;sdp1=pSrc[2];
 if(sdp!=sdp1)
 {r1[qq]=i-1;
 qq=qq+1;
 s43=s43+1;
 }}
pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=1;
qq=qq+1;
//int assd=1;
}
pSrc =(BYTE*)image1.GetPixelAddress((g1+xmin+8+s35),(mm+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=mm;
qq=qq+1;
}
qq=qq-1;
int e7=0;
int e8=0;
int e9=0;

for (int ii=1;ii<qq+1;ii++)
{if (r1[ii]<(mm*0.333))e7=e7+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 if ((r1[ii]>=(mm*0.333))&(r1[ii]<=(mm*0.66)))e8=e8+1;
if (r1[ii]>(mm*0.66))e9=e9+1;
}
int ee3=e7+e8+e9;
//****************แนวนอนแถวที่
1**************************************
g1=(0.075*mm)+1;
 qq=1;
s43=1;
 //s35=a_minx[rr];
 //s36=a_miny[rr];
//s21=s21+s35;
for (i=2;i<nn+1;i++)
{pSrc =(BYTE*)image1.GetPixelAddress((i+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
 BYTE sdp;sdp=pSrc[2];
 pSrc =(BYTE*)image1.GetPixelAddress((i-1+xmin+8+s35),
(g1+ymin+4+ymin1-cdf+s36));
 BYTE sdp1;sdp1=pSrc[2];
 if(sdp!=sdp1)
 {r1[qq]=i-1;
 qq=qq+1;
 s43=s43+1;
 }}
pSrc =(BYTE*)image1.GetPixelAddress((1+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=1;
qq=qq+1;
//int assd=1;
}
pSrc =(BYTE*)image1.GetPixelAddress((nn+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=nn;
qq=qq+1;
}
qq=qq-1;
int e10=0;
int e11=0;
int e12=0;

for (int ii=1;ii<qq+1;ii++)
{if (r1[ii]<(nn*0.333))e10=e10+1;
 if ((r1[ii]>=(nn*0.333))&(r1[ii]<=(nn*0.66)))e11=e11+1;
if (r1[ii]>(nn*0.66))e12=e12+1;
}
int ee4=e10+e11+e12;

//****************แนวนอนแถวที่
2**************************************
g1=(0.35*mm);
 qq=1;
s43=1;
 //s35=a_minx[rr];
 //s36=a_miny[rr];
//s21=s21+s35;
for (i=2;i<nn+1;i++)
{pSrc =(BYTE*)image1.GetPixelAddress((i+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
 BYTE sdp;sdp=pSrc[2];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 pSrc =(BYTE*)image1.GetPixelAddress((i-1+xmin+8+s35),
(g1+ymin+4+ymin1-cdf+s36));
 BYTE sdp1;sdp1=pSrc[2];
 if(sdp!=sdp1)
 {r1[qq]=i-1;
 qq=qq+1;
 s43=s43+1;
 }}
pSrc =(BYTE*)image1.GetPixelAddress((1+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=1;
qq=qq+1;
//int assd=1;
}
pSrc =(BYTE*)image1.GetPixelAddress((nn+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=nn;
qq=qq+1;
}
qq=qq-1;
int e13=0;
int e14=0;
int e15=0;

for (int ii=1;ii<qq+1;ii++)
{if (r1[ii]<(int)(nn*0.277))e13=e13+1;
 if ((r1[ii]>=(int)(nn*0.277))&(r1[ii]<=(int)(nn*0.71)))e14=e14+1;
if (r1[ii]>(int)(nn*0.71))e15=e15+1;
}
int ee5=e13+e14+e15;

//***************แนวนอนแถวที่
3*************************************
g1=(0.5*mm);
 qq=1;
s43=1;
 //s35=a_minx[rr];
 //s36=a_miny[rr];
//s21=s21+s35;
for (i=2;i<nn+1;i++)
{pSrc =(BYTE*)image1.GetPixelAddress((i+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
 BYTE sdp;sdp=pSrc[2];
 pSrc =(BYTE*)image1.GetPixelAddress((i-1+xmin+8+s35),
(g1+ymin+4+ymin1-cdf+s36));
 BYTE sdp1;sdp1=pSrc[2];
 if(sdp!=sdp1)
 {r1[qq]=i-1;
 qq=qq+1;
 s43=s43+1;
 }}
pSrc =(BYTE*)image1.GetPixelAddress((1+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=1;
qq=qq+1;
//int assd=1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pSrc =(BYTE*)image1.GetPixelAddress((nn+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=nn;
qq=qq+1;
}
qq=qq-1;
int e16=0;
int e17=0;
int e18=0;

for (int ii=1;ii<qq+1;ii++)
{if (r1[ii]<(int)(nn*0.28))e16=e16+1;
 if ((r1[ii]>=(int)(nn*0.28))&(r1[ii]<=(int)(nn*0.7)))e17=e17+1;
if (r1[ii]>(int)(nn*0.7))e18=e18+1;
}
int ee6=e16+e17+e18;
//****************แนวนอนแถวที่
4*************************************
g1=(0.7*mm);
 qq=1;
s43=1;
 //s35=a_minx[rr];
 //s36=a_miny[rr];
//s21=s21+s35;
for (i=2;i<nn+1;i++)
{pSrc =(BYTE*)image1.GetPixelAddress((i+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
 BYTE sdp;sdp=pSrc[2];
 pSrc =(BYTE*)image1.GetPixelAddress((i-1+xmin+8+s35),
(g1+ymin+4+ymin1-cdf+s36));
 BYTE sdp1;sdp1=pSrc[2];
 if(sdp!=sdp1)
 {r1[qq]=i-1;
 qq=qq+1;
 s43=s43+1;
 }}
pSrc =(BYTE*)image1.GetPixelAddress((1+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=1;
qq=qq+1;
//int assd=1;
}
pSrc =(BYTE*)image1.GetPixelAddress((nn+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=nn;
qq=qq+1;
}
qq=qq-1;
int e19=0;
int e20=0;
int e21=0;

for (int ii=1;ii<qq+1;ii++)
{if (r1[ii]<(int)(nn*0.333))e19=e19+1;
 if ((r1[ii]>=(int)(nn*0.333))&(r1[ii]<=(int)(nn*0.66)))e20=e20+1;
if (r1[ii]>(int)(nn*0.66))e21=e21+1;
}
int ee7=e19+e20+e21;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//****************แนวนอนแถวที่
5**************************************
g1=(0.935*mm);
 qq=1;
s43=1;
 //s35=a_minx[rr];
 //s36=a_miny[rr];
//s21=s21+s35;
for (i=2;i<nn+1;i++)
{pSrc =(BYTE*)image1.GetPixelAddress((i+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
 BYTE sdp;sdp=pSrc[2];
 pSrc =(BYTE*)image1.GetPixelAddress((i-1+xmin+8+s35),
(g1+ymin+4+ymin1-cdf+s36));
 BYTE sdp1;sdp1=pSrc[2];
 if(sdp!=sdp1)
 {r1[qq]=i-1;
 qq=qq+1;
 s43=s43+1;
 }}
pSrc =(BYTE*)image1.GetPixelAddress((1+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=1;
qq=qq+1;
//int assd=1;
}
pSrc =(BYTE*)image1.GetPixelAddress((nn+xmin+8+s35),(g1+ymin+4+ymin1-
cdf+s36));
if (pSrc[2]==0)
{r1[qq]=nn;
qq=qq+1;
}
qq=qq-1;
int e22=0;
int e23=0;
int e24=0;

for (int ii=1;ii<qq+1;ii++)
{if (r1[ii]<(int)(nn*0.333))e22=e22+1;
 if ((r1[ii]>=(int)(nn*0.333))&(r1[ii]<=(int)(nn*0.66)))e23=e23+1;
if (r1[ii]>(int)(nn*0.66))e24=e24+1;
}
int ee8=e22+e23+e24;
//************** หา e25
**********************************************
int ww=0;
int w1;
int e25=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int i=(mm*0.2);i<(mm*0.5);i++)
{w1=0;
      for (int j=nn;j>(nn*0.5);j--)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.455)){e25=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//************** หา e26
*********************************************
 ww=0;
//int w1;
int e26=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int i=(mm*0.55);i<(mm*0.85);i++)
{w1=0;
      for (int j=1;j<(nn*0.8);j++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.455)){e26=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e27
*********************************************
 ww=0;
//int w1;
int e27=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=(nn*0.3);j<(nn*0.7);j++)
{w1=0;
      for (int i=mm;i>(mm*0.5);i--)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.357)){e27=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }if(e27==1){break;}
}
//************** หา e28
*********************************************
 ww=0;
//int w1;
int e28=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=(nn*0.3);j<(nn*0.7);j++)
{w1=0;
      for (int i=1;i<(mm*0.5);i++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.357)){e28=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e29
*********************************************
 ww=0;
//int w1;
int e29=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int i=(mm*0.2);i<(mm*0.5);i++)
{w1=0;
      for (int j=1;j<(nn*0.8);j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.32)){e29=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e30
*********************************************
 ww=0;
//int w1;
int e30=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=(nn*0.2);j<(nn*0.5);j++)
{w1=0;
      for (int i=mm;i>(mm*0.5);i--)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.45)){e30=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e31
*********************************************
 ww=0;
//int w1;
int e31=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=(nn*0.5);j<(nn*0.8);j++)
{w1=0;
      for (int i=1;i<(mm*0.5);i++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.357)){e31=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e32
*********************************************
 ww=0;
//int w1;
int e32=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=(nn*0.8);j<nn+1;j++)
{w1=0;
      for (int i=1;i<(mm*0.5);i++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.40)){e32=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}

//************** หา e33
*********************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 ww=0;
//int w1;
int e33=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
 int fr=0;
int fr1=0;
int fr2=0;

for (int j=1;j<(nn*0.2);j++)
{w1=0;
 fr=0;
 fr1=0;
 fr2=0;
      for (int i=1;i<mm+1;i++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==0 & fr==0){fr=1;fr1=1;}
      if(pSrc[2]==255 & fr1==1){fr2=1;w1=w1+1;}
      if(pSrc[2]==0&fr2==1)
      { if (w1>(mm*0.35)){e33=1;ww=1;break;}
      if (w1<=(mm*0.35)){e33=0;ww=0;w1=0;fr=0;fr1=0;fr2=0;}
      }
      }if (e33==1){break;}
}

//************** หา e34
*******************************************
 ww=0;
//int w1;
int e34=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int i=3;i<(mm*0.2);i++)
{w1=0;
      for (int j=1;j<(nn*0.8);j++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.47)){e34=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e35
*******************************************
 ww=0;
//int w1;
int e35=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=(nn*0.5);j<(nn*0.85);j++)
{w1=0;
      for (int i=mm;i>(mm*0.5);i--)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.45)){e35=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//************** หา e36
*********************************************
 ww=0;
//int w1;
int e36=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=(nn*0.15);j<(nn*0.5);j++)
{w1=0;
      for (int i=1;i<(mm*0.5);i++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.4)){e36=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e37
*********************************************
 ww=0;
//int w1;
int e37=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=1;j<(nn*0.2);j++)
{w1=0;
      for (int i=mm;i>(mm*0.5);i--)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.5)){e37=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e38
*********************************************
 ww=0;
//int w1;
int e38=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=(nn*0.85);j<nn+1;j++)
{w1=0;
      for (int i=mm;i>(mm*0.2);i--)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.45)){e38=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e39
*********************************************
 ww=0;
//int w1;
int e39=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int j=1;j<(nn*0.15);j++)
{w1=0;
      for (int i=1;i<(mm*0.5);i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.4)){e39=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}
//************** หา e40
*********************************************
 ww=0;
//int w1;
int e40=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
  fr=0;
 fr1=0;
 fr2=0;

for (int j=(nn*0.85);j<nn+1;j++)
{w1=0;
 fr=0;
 fr1=0;
 fr2=0;
      for (int i=1;i<mm+1;i++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==0 & fr==0){fr=1;fr1=1;}
      if(pSrc[2]==255 & fr1==1){fr2=1;w1=w1+1;}
      if(pSrc[2]==0&fr2==1)
      { if (w1>(mm*0.4)){e40=1;ww=1;break;}
      if (w1<=(mm*0.4)){e40=0;ww=0;w1=0;fr=0;fr1=0;fr2=0;}
      }
      }
}
//************** หา e41
*********************************************
 ww=0;
//int w1;
int e41=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int i=(mm*0.8);i<mm+1;i++)
{w1=0;
      for (int j=1;j<(nn*0.8);j++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.357)){e41=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}

//************** หา e42
*********************************************
 ww=0;
//int w1;
int e42=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
  fr=0;
 fr1=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 fr2=0;

for (int j=(nn*0.2);j<(nn*0.5);j++)
{w1=0;
 fr=0;
 fr1=0;
 fr2=0;
      for (int i=1;i<mm+1;i++)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==0 & fr==0){fr=1;fr1=1;}
      if(pSrc[2]==255 & fr1==1){fr2=1;w1=w1+1;}
      if(pSrc[2]==0&fr2==1)
      { if (w1>(mm*0.37)){e42=1;ww=1;break;}
      if (w1<=(mm*0.37)){e42=0;ww=0;w1=0;fr=0;fr1=0;fr2=0;}
      }
      }


}
//************** หา e43
*********************************************
 ww=0;
 w1;
int e43=0;
//s35=a_minx[rr];
 //s36=a_miny[rr];
for (int i=(mm*0.55);i<(mm*0.85);i++)
{w1=0;
      for (int j=nn;j>(nn*0.5);j--)
      {pSrc =(BYTE*)image1.GetPixelAddress((j+xmin+8+s35),
(i+ymin+4+ymin1-cdf+s36));
      if(pSrc[2]==255)w1=w1+1;
      if(w1>(nn*0.455)){e43=1;ww=1;}
      if(pSrc[2]==0){w1=0;break;}
      }
}

//*****************************************************
//******************** การแยกค่า ตัว เ ล ข ************************
//*****************************************************
out_1[rr]='#';
if((ee1==2|ee1==4) &  (ee3==2|ee3==4) & (ee4==2|ee4==4)  & ee2 ==4 &
ee5==4 & ee6==4 & ee7==4   & e25==0 &e28==0 &e27==0 &e29==0)
{out_1[rr]='0';}
if(ee1==2 &ee2==2 &ee3 ==2 & (ee4==2|ee4==4)& ee5==2& ee6==2& ee7==2
&(ee8==2|ee8==4 ))
{out_1[rr]='1';}
if ((ee1==4|ee1==6) & ee6==2 & ee7==2 &  (ee8==2|ee8==4 ) &
(ee4==2|ee4==4) & (ee3==4|ee3==6) & (ee2==6|ee2==4) & (ee5==2 |
ee5==4) & e29==1 &e43==1)
{out_1[rr]='2';}
if ((ee4==2|ee4==4) & ee6==2 & (ee8==2|ee8==4 ) & (ee3==2 | ee3==4) &
(ee5==2 | ee5==4)& (ee7==2 | ee7==4)  & ee1==4 & e25==0 & e27==0
&e29==1 &e26==1 &e43==0)
{out_1[rr]='3';}
if (ee1==2 & ee3 ==2 & (ee4==2|ee4==4) & (ee7 ==2|ee7==4) &
(ee8==2|ee8==4 ) & ( ee5 ==2 | ee5 ==4)  & (ee2 ==4/*|ee2==2*/) &
(ee6==4|ee6==2)  &e32==1 &e41==1 &e39==1 &e25==0  & e29==1 )
{out_1[rr]='4';}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ( ee1==4 & ee2==6 & (ee3==4|ee3==6) & (ee4==2|ee4==4) &
(ee5==2|ee5==4) & (ee6==2|ee6==4) & (ee7==2|ee7==4) & (ee8==2|ee8==4
) & e25==1 & e26==1 &e29==0)
{out_1[rr]='5';}
if (ee1==2 & ee2==6 & ee3==4 & (ee4==2|ee4==4) & (ee5==2|ee5==4) &
(ee6==2|ee6==4) & ee7==4 & (ee8==2|ee8==4 )  & e25==1 & e26==0)
{out_1[rr]='6';}
if ((ee1==2|ee1==4) & (ee2==4|ee2==2) & ee3==2 & (ee4==2|ee4==4) &
ee5==2 & ee6==2 & ee7==2 & (ee8==2|ee8==4 )& e29==1  & e25==0
&e28==0&e38==1 )
{out_1[rr]='7';}
if ((ee1==2|ee1==4) & ee2==6 & (ee3==2|ee3==4|ee3==6) &
(ee4==2|ee4==4) & ee5==4 & (ee6==2|ee6==4) & ee7==4 &(ee8==2|ee8==4 )
& e25==0 & e26==0 &e27==0 &e28==0)
{out_1[rr]='8';}
if ((ee1==4|ee1==2) & ee2==6 & ee3==2 & (ee4==2|ee4==4) & ee5==4 &
(ee6==2|ee6==4) & (ee7==2|ee7==4) & (ee8==2|ee8==4 )& e25==0 & e26==1
&e29==0 )
{out_1[rr]='9';}

//*****************ตัว อัก ษร*****************************************
*******************

if(out_1[1]=='1'){aqq=aqq-1;}
//if(out_1[1]==NULL){aqq=aqq-1;}
if (aqq>rr & rr<3){aqq=rr;}
if (rr<4 & aqq<3 & out_1[rr-1]!='-')//start rr<4

{
      if((ee1==2 & (ee2==8|ee2==6) & e26==1 & e25==0 &
(ee5==2|ee5==4)  & ee7==2  &e29==1)| (e39==1 & e29==1 & e26==1 &
e25==0 & ee7==2 & e40==1))
{out_1[rr]='ฐ';aqq=aqq+1;}
if ((ee1==2|ee1==4) & ee2==2 & (ee3==2|ee3==4) & (ee4==4|ee4==2) &
ee5==4  & ee6==4 & ee7==4 & (ee8==4|ee8==2)  & e27==1 & e28==0
&e25==0&e26==0)
{out_1[rr]='ก';aqq=aqq+1;}
if (e30==1 & e35==0 & e31==1 & e36==0 & ee6==4 & ee8==2 & e10!=0 &
e12!=0 & e34==0  )
{out_1[rr]='ข';aqq=aqq+1;}
if (ee1==2 & (ee2==2|ee2==4) & ee3==2 & ee4==2 & (ee5==2|ee5==4) &
ee6==2 & ee7==2 & ee8==2 & e26==1 & e27==1 & e34==0 &e29==1)
{out_1[rr]='ว';aqq=aqq+1;}
if (ee1==2 & (ee2==2|ee2==4) & ee3==4 & (ee4==4|ee4==2)  & ee6==2 &
ee7==2 & ee8==2 & e26==1 & e27==1  )
{out_1[rr]='ร';aqq=aqq+1;}

if( (ee1==4 & (ee2==4|ee2==6) & ee3==2 & e29==1 & e26==0 & ee6==4)|
(e37==1& e25==0 & e26==0 & e27==0 & e28==0 &e29==0 &
(ee2==4|ee2==6)))
{out_1[rr]='อ';aqq=aqq+1;}
if (e41==1& e28==1 & e29==1 & ee1==2 &(ee2==2|ee2==4) & ee3==2  &
ee6==4)
{out_1[rr]='ง';aqq=aqq+1;}
if (e30==1 & e31==1 & e32==1 & ee6==6 & ee5==6 )//& e41==0)
{out_1[rr]='ณ';aqq=aqq+1;}
if (e30==0 & e31==1 & e32==0 & ee6==6 & ee5==6 & e41==0)
{out_1[rr]='ฆ';aqq=aqq+1;}
if (e30==1 & e31==1 & e32==0 & ee6==6 & ee5==6 & e41==1)
{out_1[rr]='ญ';aqq=aqq+1;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (e27==1 & ee1==2 & ee2==4 & ee3==2 & ee4==4 & ee6==6 &
(ee7==4|ee7==6) & ee8==4)
{out_1[rr]='ต';aqq=aqq+1;}
if (e27==1 & ee1==2 & ee2==4 & ee3==2 & ee4==2 & ee6==6 &
(ee7==4|ee7==6) & ee8==4)
{out_1[rr]='ด';aqq=aqq+1;}

if    (ee3==2   & (ee2 ==2|ee2==4) & ee5==4 & ee6==4 & ee7==4  &
ee8==4 & e33==1 & e27==1 &e28==0)
{out_1[rr]='ภ';aqq=aqq+1;}
if    (e42==1 & ee3==2 & ee4==2  & (ee2 ==2|ee2==4) & ee5==4 & ee6==4
& ee7==4  & ee8==4 & e33==0 & e27==1)
{ out_1[rr]='ถ';aqq=aqq+1;}
if  (e37==1 & e36==1 & e35==1 & ee7==4 & ee8==4 & ee2==2 & ee2==2
&e31==1)
{out_1[rr]='ห';aqq=aqq+1;}
if  (e37==1 & e36==1 & e35==1 & ee7==4 & ee6==4 & e31==0 & ee8==4 &
ee2==2 & e38==0)
{out_1[rr]='ท';aqq=aqq+1;}
if  (e37==1 & e36==1 & e31==1 & e38==1 & e32==1 & ee2==4 & e35==0)
{out_1[rr]='ษ';aqq=aqq+1;}
if (e29==1 & e26==1 & e30==1 & e37==1 & ee1==2 & ee2==4 & ee6==4 &
e34==0 & ee7==2 & ee8==2)
{out_1[rr]='จ';aqq=aqq+1;}
if (e29==1 & ee1==4 & ee2==6 & e25==1 & e30==0 & e27==0 & ee7==4 &
ee3==4)
{out_1[rr]='ธ';aqq=aqq+1;}
if (e37==1 & e28==1 & e25==0 & e26==0 & e35==0 & (ee4==4|ee4==2) &
ee6==4 & ee8==4 & ee2==2 & e38==0)
{out_1[rr]='น';aqq=aqq+1;}
if   (e37==0 & ee3==2 & ee4==4  & ee2 ==2 & ee5==4 & ee6==4 &
(ee7==4|ee7==2)  & ee8==4 & e33==1  & e28==1)//& e27==0
{out_1[rr]='ม';aqq=aqq+1;}
if (e37==1 & e25==0 & e26==0 & e27==1 & ee1==2 & ee2==2 & ee3==2 &
ee5==6 & ee6==2 & ee7==4 & ee8==4  & e34==0)
{out_1[rr]='พ';aqq=aqq+1;}
if (e37==1 & e25==0 & e26==0  & ee1==2 & ee2==2 & ee3==2 & ee4==2   &
ee8==4 & e28==1 & e34==1 &  e12!=0 & e10==0)
{out_1[rr]='ฟ';aqq=aqq+1;}
if (e37==0 & e25==0 & e26==0 & e27==0 & ee1==2 & ee2==2 & ee3==2 &
ee4==4 & ee6==4  & ee8==4 & e28==1 & e34==0 &e33==0 )
{out_1[rr]='ผ';aqq=aqq+1;}
if (e37==0 & e25==0 & e26==0 & e27==0 & ee1==2 & ee2==2 & ee3==2 &
ee4==2 & ee6==4  & ee8==4 & e28==1 & e34==1 & e12!=0 & e10==0
&e33==0)
{out_1[rr]='ฝ';aqq=aqq+1;}
if (e34==0 & e31==1 & e36==1 & e25==0 & e26==0 & e27==0 & e28==1 &
ee4==4 &(ee2==2|ee2==4) & ee6==4 & ee8==2 &e37==0 & e17==1 &
(ee1==2|ee1==4))
{out_1[rr]='ย';aqq=aqq+1;}
if (e34==0 & e31==1 & e36==1 & e25==0 & e26==0 & e27==0 & e28==1 &
ee4==4 &ee2==2 & ee6==4 & ee8==2 &e37==1 & ee1==2)
{out_1[rr]='บ';aqq=aqq+1;}
if (e34==1 & e31==1 & e36==1 & e25==0 & e26==0 & e27==0 & e28==1 &
ee4==2 &ee2==2 & ee6==4 & ee8==2 &e37==1& e12!=0 & e10==0 &e41==0)
{out_1[rr]='ป';aqq=aqq+1;}
if (ee1==4 & ee2==4 & ee3==2 & ee4==2 & ee8==4 & e25==0 & e43==0 &
e26==0 & e28==0 & e37==1 & e32==1)
{out_1[rr]='ฉ';aqq=aqq+1;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if ( (ee1==4|ee1==6) & (ee2==4|ee2==6) & ee4==2 & ee8==4 & e25==0 &
e43==0 & e26==0 & e28==0 & e32==0 & e27==1 & e33==0)
{out_1[rr]='ล';aqq=aqq+1;}

if (e27==1 &e30==1 &e35==1 & ee1==2 & ee2==4 & ee3==2 & ee4==2 &
ee6==4 & ee7==4 & ee8==4 & e29==0 & e38==0 & e28==0 & e25==0)
{out_1[rr]='ค';aqq=aqq+1;}
if (e37==0 & e25==0 & e26==0 & e27==0 & e28==0 &e29==1 & ee2==6 &
ee1==4 &(ee3==2|ee3==4) )
{out_1[rr]='ย';aqq=aqq+1;}
if (e27==1 &e30==1 &e35==1 & ee1==2 & ee2==4 & ee3==2 & ee4==2 &
ee6==4 & ee7==4 & ee8==4 & e29==0 & e38==1 & e28==0 & e25==0)
{out_1[rr]='ศ';aqq=aqq+1;}

if (ee1==2 & (ee2==4|ee2==6) & ee3==2 & ee8==4 & e38==1 & e26==0 &
e25==0 & e43==0 & e37==1 )
{out_1[rr]='ฟ';aqq=aqq+1;}


} //end rr<4



if (out_1[rr]=='#'){out_1[rr]='?';}

if (w1==1)int adw=1;//ลบได้นะ
}//end if  mm>nn  & nn>5
if ((mm<10 |nn<10)  & (mm<=nn)){out_1[rr]='-';}
if ((mm<12  & nn<12)  ){out_1[rr]='-';}
if (nn<7 &mm>10){out_1[rr]='1';}
}//end loop for
}//end  if bbb!=0
//m_hu=out_1;

                double temp;
                temp=(double)xmax;
                double tem;
                tem=(double)xmin;
                 double tee;
                tee=(double)ymax;
                double tee1;
                tee1=(double)ymin;
                 for (int i=0;i<heigh;i++)
                {
                      for (int j=0;j<width;j++)
                      {
                      pSrc=(BYTE*)image1.GetPixelAddress(j+1,i+1);
                      BYTE kkk0=pSrc[0];
                      BYTE kkk1=pSrc[1];
                      BYTE kkk2=pSrc[2];
                      pSrc1 =(BYTE*)pImage.GetPixelAddress(j,i);
                      pSrc1[0]=kkk0;
                      pSrc1[1]=kkk1;
                    pSrc1[2]=kkk2;
                      }
                }
                aswq=1;
                out_11=out_1;

                image2.Destroy();
                image1.Destroy();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Pimage.Destroy();
                    Pimage1.Destroy();
                    pImage.Destroy();
data_error: if (aswq==0) {m_hu="error";}
                    if (out_1[1]=='e')
                    {out_11="   ";}

}}
void CLicenseView::OnPathSetpath()
{

      Set.DoModal();

}

void CLicenseView::ClearScreen(void)
{     CString bb;
      CString cc;
      UpdateData(TRUE);
      bb=m_find;
      cc=m_find2;
      m_strcard=(_T(""));
      m_strcar=(_T(""));
      m_strshow=(_T(""));
      m_strpw=(_T(""));
      m_find=bb;
      m_find2=cc;
      UpdateData(FALSE);
}

void CLicenseView::OnCaptureVideosource()
{STOP();
      if (!m_Video.Init())
      {
            MessageBox(m_Video.GetLastErrorString(), "License Error",
MB_OK);
            return;           // exit
      }
            if (m_Video.GetMode() == 0)
      {
            if (!m_Video.ShowSourceDlg())
            {
            }
            if (flag1==TRUE)
            {     flag1=FALSE;
                  OnBnClickedWork();
            }
      }
      else
      {
      }
}

void CLicenseView::OnCaptureVideoformat139()
{


if (!m_Video.Init())
      {
            MessageBox(m_Video.GetLastErrorString(), "License Error",
MB_OK);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return;           // exit
      }
      if (m_Video.GetMode() == 0)
      {
            if (!m_Video.ShowFormatDlg())
            {
                  MessageBox(m_Video.GetLastErrorString(), "",
MB_OK);
            }

            m_Video.ReleaseImageHandle(m_Image);

      }
      else
      {
            MessageBox("This feature is only available in VfW
Mode.","VideoOCX Sample Error", MB_OK + MB_ICONINFORMATION);

      }
      STOP2();
}


void CLicenseView::OnSettingComport()
{
      CSetComPort setcom;
      setcom.DoModal();
      CloseHandle(hComp);
      m_comport=setcom.GetComPort();
      InitialSerial(m_comport);
}

void CLicenseView::OnBnClickedRadio()
{
      m_comport="COM1";
}

void CLicenseView::OnBnClickedRadio2()
{
      m_comport="COM2";
}

void CLicenseView::Draw1(CString ppic)
{CBitmap bmp;
      BITMAP bm;
      CClientDC clientDC(this);
      CDC memDC;
            CSize size;
            if (ppic=="IDB_BITMAP1")
            bmp.LoadBitmap(IDB_BITMAP1);
            else bmp.LoadBitmap(IDB_BITMAP3);
            bmp.GetBitmap(&bm);
            size.cx=bm.bmWidth;
            size.cy=bm.bmHeight;
            memDC.CreateCompatibleDC(&clientDC);
            CBitmap* pOldBitmap=memDC.SelectObject(&bmp);

      clientDC.BitBlt(600,395,size.cx,size.cy,&memDC,0,0,SRCCOPY);
            memDC.SelectObject(pOldBitmap);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void CLicenseView::Draw2(CString ppic2)
{CBitmap bmp;
	BITMAP bm;
	CClientDC clientDC(this);
	CDC memDC;
		CSize size;
		if (ppic2=="IDB_BITMAP2")
		bmp.LoadBitmap(IDB_BITMAP2);
		else bmp.LoadBitmap(IDB_BITMAP3);
		bmp.GetBitmap(&bm);
		size.cx=bm.bmWidth;
		size.cy=bm.bmHeight;
		memDC.CreateCompatibleDC(&clientDC);
		CBitmap* pOldBitmap=memDC.SelectObject(&bmp);

	clientDC.BitBlt(866,396,size.cx,size.cy,&memDC,0,0,SRCCOPY);
		memDC.SelectObject(pOldBitmap);
}

void CLicenseView::OnDraw(CDC* /*pDC*/)
{
	Draw1(pic);
	Draw2(pic2);
}

void CLicenseView::Sleep(clock_t wait)
{	clock_t goal;
   goal = wait + clock();
   while( goal > clock() )
	;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ LicenseView.h

```
// LicenseView.h : interface of the CLicenseView class
//

#pragma once
#include "atlcomtime.h"
#include "Capture.h"
#include "afxwin.h"
#include "videoocx.h"

class CLicenseView : public CFormView
{     private:
      CObList* m_emList;
      POSITION m_pCurrent;
      BOOL m_bFlag;
      void ShowRecord(POSITION position);
      void ClearRecord(void);

protected: // create from serialization only
      CLicenseView();
      DECLARE_DYNCREATE(CLicenseView)

public:
      enum{ IDD = IDD_LICENSE_FORM };

// Attributes
public:
      ClicenseDoc* GetDocument() const;

// Operations
public:

// Overrides
      public:
virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
protected:
      virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV
support
      virtual void OnInitialUpdate(); // called first time after
construct

// Implementation
public:
      virtual ~CLicenseView();
#ifdef _DEBUG
      virtual void AssertValid() const;
      virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
      DECLARE_MESSAGE_MAP()
public:
      CString m_strcard;
      CString m_strcar;
public:
      afx_msg void OnBnClickedAddRecord();
      afx_msg void OnBnClickedDeleteRecord();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        afx_msg void OnBnClickedFirstRecord();
        afx_msg void OnBnClickedPreRecord();
        afx_msg void OnBnClickedNextRecord();
        afx_msg void OnBnClickedLastRecord();
protected:
        virtual void OnUpdate(CView* /*pSender*/, LPARAM /*lHint*/,
CObject* /*pHint*/);

public:
        afx_msg void OnBnClickedRecv();
        void InitialSerial(CString com);
        CString m_strcomport;
        CString m_strshow;
        afx_msg void OnBnClickedClear();
        afx_msg void OnBnClickedCom1();
        afx_msg void OnBnClickedCom2();
        void Add(void);


        afx_msg void OnBnClickedFind();
        CString m_strcardfound;
        CString m_strcarfound;
        CString m_find;
        CString m_find2;
        afx_msg void OnBnClickedButton2();
        afx_msg void OnBnClickedRecv2();
        afx_msg void OnEnChangeCar();
        CString m_strpw;
        CString m_strpwfound;
        afx_msg void OnBnClickedWork();
        CButton m_ButtonAdd;
        bool m_Running;
        long m_Image;
        CVideoOCX m_Video;
        CButton m_ButtonDelete;
        CButton m_ButtonRecv;
        afx_msg void OnClose();
        void STOP(void);
        CString m_time;
        CButton m_Capture;
        BOOL flag1;
        void WORK(void);
        void SAVE(void);
        void WORK2(void);
        void STOP2(void);
        CButton m_clear;
        CButton m_first;
        CButton m_prev;
        CButton m_next;
        CButton m_last;
        CButton m_search;
        CButton m_recv2;
        CButton m_clear2;
        void KEY(void);
protected:
        CSize m_size;
public:
       int m_get;
        double m_heigh;
        double m_width;
        int BPP;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      int heigh;
      int width;
      int bufpic;
    BYTE*pSrc;
      CString m_hu;
      void Detect(void);
      char sumbuffer[50];//41
      afx_msg void OnPathSetpath();
      bool cca;
      void ClearScreen(void);
      afx_msg void OnCaptureVideosource();
      afx_msg void OnCaptureVideoformat139();
      afx_msg void OnSettingComport();
      afx_msg void OnBnClickedRadio();
      CString m_comport;
      afx_msg void OnBnClickedRadio2();
      bool flag2;

      void Draw1(CString ppic);
      void Draw2(CString ppic2);
      CString pic;
      CString pic2;
protected:
      virtual void OnDraw(CDC* /*pDC*/);
public:
      void Sleep(clock_t wait);
};

#ifndef _DEBUG  // debug version in LicenseView.cpp
inline ClicenseDoc* CLicenseView::GetDocument() const
   { return reinterpret_cast<ClicenseDoc*>(m_pDocument); }
#endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ LicenseDoc.cpp

```
// licenseDoc.cpp : implementation of the ClicenseDoc class
//

#include "stdafx.h"
#include "License.h"

#include "LicenseDoc.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif


// ClicenseDoc

IMPLEMENT_DYNCREATE(ClicenseDoc, CDocument)

BEGIN_MESSAGE_MAP(ClicenseDoc, CDocument)
END_MESSAGE_MAP()


// ClicenseDoc construction/destruction

ClicenseDoc::ClicenseDoc()
{
	// TODO: add one-time construction code here

}

ClicenseDoc::~ClicenseDoc()
{
}

BOOL ClicenseDoc::OnNewDocument()
{
	if (!CDocument::OnNewDocument())
		return FALSE;

	// TODO: add reinitialization code here
	// (SDI documents will reuse this document)

	return TRUE;
}

// ClicenseDoc serialization

void ClicenseDoc::Serialize(CArchive& ar)
{
	if (ar.IsStoring())
	{
		// TODO: add storing code here
		m_licenseList.Serialize(ar);
	}
	else
	{
		// TODO: add loading code here
		m_licenseList.Serialize(ar);
	}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CObList* ClicenseDoc::GetLicenseList()
{     return &m_licenseList;
}
// ClicenseDoc diagnostics

#ifdef _DEBUG
void ClicenseDoc::AssertValid() const
{
      CDocument::AssertValid();
}

void ClicenseDoc::Dump(CDumpContext& dc) const
{
      CDocument::Dump(dc);
}
#endif //_DEBUG


// ClicenseDoc commands

void ClicenseDoc::DeleteContents()
{
      while (!m_licenseList.IsEmpty())
      { delete m_licenseList.RemoveHead();
      }
      CDocument::DeleteContents();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ LicenseDoc.h

```
//LicenseDoc.h : interface of the ClicenseDoc class
//

#pragma once
class ClicenseDoc : public CDocument
{
protected: // create from serialization only
      ClicenseDoc();
      DECLARE_DYNCREATE(ClicenseDoc)

// Attributes
public:

// Operations
public:

// Overrides
      public:
      virtual BOOL OnNewDocument();
      virtual void Serialize(CArchive& ar);

// Implementation
public:
      CObList* GetLicenseList();
      virtual ~ClicenseDoc();
#ifdef _DEBUG
      virtual void AssertValid() const;
      virtual void Dump(CDumpContext& dc) const;
#endif

protected:
      CObList m_licenseList;
// Generated message map functions
protected:
      DECLARE_MESSAGE_MAP()
public:
      virtual void DeleteContents();

};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ Licenses.cpp

```
#include "StdAfx.h"
#include "Licenses.h"
IMPLEMENT_SERIAL (CLicenses, CObject, Versions);
CLicenses::CLicenses(void)
: m_card(_T(""))
, m_pw(_T(""))
{
}

CLicenses::~CLicenses(void)
{
}

void CLicenses::SetCard(CString card)
{     m_card=card;
}

void CLicenses::SetCar(CString car)
{     m_car=car;
}

CString CLicenses::GetCard()
{
      return m_card;
}

CString CLicenses::GetCar()
{
      return m_car;
}

void CLicenses::Serialize(CArchive &ar)
{     CObject::Serialize(ar);
      if (ar.IsStoring())
      {     ar<<m_car<<m_card<<m_pw;
      }
      else
      {     ar>>m_car>>m_card>>m_pw;
      }
}

void CLicenses::SetPw(CString pw)
{     m_pw=pw;
}

CString CLicenses::GetPw(void)
{
      return m_pw;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ Licenses.h

```
#pragma once
#include "afx.h"
const Versions=1;
class CLicenses :
      public CObject
{
            DECLARE_SERIAL (CLicenses);
public:
      CLicenses(void);
      ~CLicenses(void);
      void SetCard(CString card);
      void SetCar(CString car);
      CString GetCard(void);
      CString GetCar(void);
      void Serialize(CArchive &ar);
private:
      CString m_card;
      CString m_car;


public:
      void SetPw(CString pw);
      CString GetPw(void);
private:
      CString m_pw;
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ SetComPort.cpp

```
// SetComPort.cpp : implementation file
//

#include "stdafx.h"
#include "License.h"
#include "SetComPort.h"


// CSetComPort dialog

IMPLEMENT_DYNAMIC(CSetComPort, CDialog)
CSetComPort::CSetComPort(CWnd* pParent /*=NULL*/)
	: CDialog(CSetComPort::IDD, pParent)
	, m_comport(_T(""))
	, rad(0)
{	m_comport="COM1";
}

CSetComPort::~CSetComPort()
{
}

void CSetComPort::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	DDX_Radio(pDX, IDC_RADIO, rad);
}


BEGIN_MESSAGE_MAP(CSetComPort, CDialog)

END_MESSAGE_MAP()


// CSetComPort message handlers



CString CSetComPort::GetComPort(void)
{	return m_comport;
}

void CSetComPort::OnOK()
{
	// TODO: Add your specialized code here and/or call the base
class
	if(IsDlgButtonChecked(IDC_RADIO))
	{	m_comport="COM1";
	}
	if(IsDlgButtonChecked(IDC_RADIO2))
	{	m_comport="COM2";
	}
	if(IsDlgButtonChecked(IDC_RADIO3))
	{	m_comport="COM3";
	}
	if(IsDlgButtonChecked(IDC_RADIO4))
	{	m_comport="COM4";
	}
	if(IsDlgButtonChecked(IDC_RADIO5))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	{	m_comport="COM5";
	}

		CDialog::OnOK();
}

BOOL CSetComPort::OnInitDialog()
{
	CDialog::OnInitDialog();

	return TRUE;  // return TRUE unless you set the focus to a
control
	// EXCEPTION: OCX Property Pages should return FALSE
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ SetComPort.h

```
#pragma once


// CSetComPort dialog

class CSetComPort : public CDialog
{
	DECLARE_DYNAMIC(CSetComPort)

public:
	CSetComPort(CWnd* pParent = NULL);   // standard constructor
	virtual ~CSetComPort();

// Dialog Data
	enum { IDD = IDD_DIALOG2 };

protected:
	virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV
support

	DECLARE_MESSAGE_MAP()
public:

	CString m_comport;

	CString GetComPort(void);
protected:
	virtual void OnOK();
public:
	virtual BOOL OnInitDialog();
	int rad;

};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ SetPath.cpp

```
// SetPath.cpp : implementation file
//

#include "stdafx.h"
#include "License.h"
#include "SetPath.h"

// CSetPath dialog

IMPLEMENT_DYNAMIC(CSetPath, CDialog)
CSetPath::CSetPath(CWnd* pParent /*=NULL*/)
	: CDialog(CSetPath::IDD, pParent)
{	m_path="C:\\pic";
}

CSetPath::~CSetPath()
{
}

void CSetPath::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	DDX_Text(pDX, IDC_EDIT1, m_path);
}


BEGIN_MESSAGE_MAP(CSetPath, CDialog)
	ON_BN_CLICKED(IDOK, OnBnClickedOk)
END_MESSAGE_MAP()


// CSetPath message handlers

void CSetPath::OnBnClickedOk()
{
	// TODO: Add your control notification handler code here
	UpdateData(TRUE);
	OnOK();

}

CString CSetPath::GetPath(void)
{
	return m_path;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ SetPath.h

```
#pragma once


// CSetPath dialog

class CSetPath : public CDialog
{
	DECLARE_DYNAMIC(CSetPath)

public:
	CSetPath(CWnd* pParent = NULL);   // standard constructor
	virtual ~CSetPath();

// Dialog Data
	enum { IDD = IDD_DIALOG1 };

protected:
	virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV
support

	DECLARE_MESSAGE_MAP()
public:
	CString m_path;
	afx_msg void OnBnClickedOk();
	CString GetPath(void);
};
```

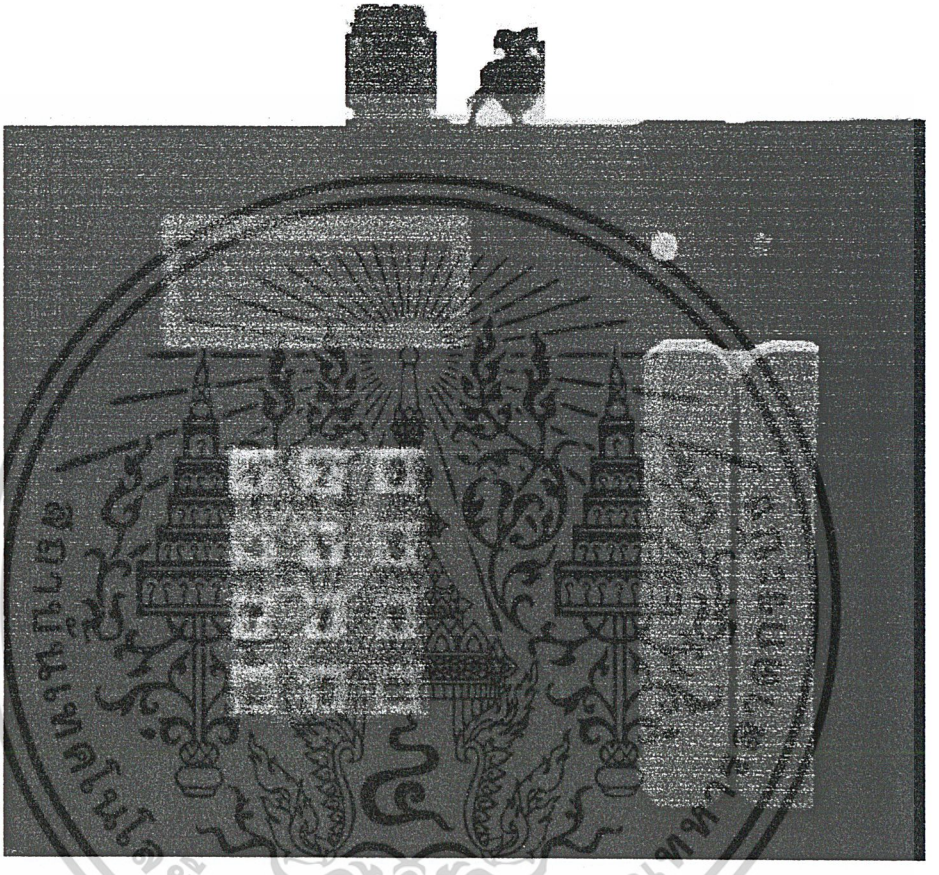
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก
ค. รูปชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก
ค. รูปชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ISO 7881, "Identification Cards-Recording Technique", 1995
2. Albert S.Hoagland and James E.Monson, "Digital Magnetic Recording", John Wiley & Sons Inc, 1991
3. "การประยุกต์ใช้บัตรแม่หล็ก", ปริญญานิพนธ์ภาควิชาวิศวกรรมอิเล็กทรอนิกส์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพฯ, 2541
4. อำไพ สินลิขิตกุล, "อินไซท์ SQL Server7", สำนักพิมพ์ โปรวิชัน, กรุงเทพฯ, 2544
5. รศ.สมยศ จุณณะปิยะ, "การประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล MCS-51" คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546
6. ยุทธนา ลีลาศวัฒนกุล, " คู่มือการเขียนโปรแกรมและใช้งาน Visual C++.NET", สำนักพิมพ์อินโฟเพรส, นนทบุรี, 2546
7. ยุทธนา ลีลาศวัฒนกุล, "คู่มือการเขียนโปรแกรมวินโดว์ขั้นสูงด้วย Visual C++.NET Episode One", ดวงกมลสมัย, กรุงเทพฯ, 2546
8. Gonzalez, and R.E. Wood, "Digital Image Processing", Addison-Wiley, 1992
9. "ระบบจดจำป้ายทะเบียนรถยนต์, ปริญญานิพนธ์ภาควิชาวิศวกรรมอิเล็กทรอนิกส์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพฯ, 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้