สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบตรวจสอบลายนิ้วมือระบุบุคคล
Fingerprint Identify System

T055463

โดย

นาย ขวัญชัย เรืองบุญคุ้มสุข

นาย สุเมธ สำเร็จศิลป์

นาย แสนศักดิ์ บวรจิโรภาส

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช

เลขหมู่.....
เลขทะเบียน 55463
วัน,เดือน,ปี - 9 พ.ค. 2548

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

ระบบตรวจสอบลายนิ้วมือระบุบุคคล

Fingerprint Identify System

โดย

นาย ขวัญชัย เรืองบุญคุ้มสุข 43010040

นาย สุเมธ สำเร็จศิลป์ 43010495

นาย แสนศักดิ์ บวรจิโรภาส 43010508

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สังวรศิลป์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชา อิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจสอบลายนิ้วมือระบุบุคคล

Fingerprint Identify System

จัดทำโดย นาย ขวัญชัย เรืองบุญคุ้มสุข รหัส 43010040

นาย สุเมธ สำเร็จศิลป์ รหัส 43010495

นาย แสนศักดิ์ บวรจิโรภาส รหัส 43010508

อาจารย์ที่ปรึกษา รศ.ดร. มนัส สังวรศิลป์

ปริญญานิพนธ์ฉบับนี้ได้ผ่านการตรวจสอบจากอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....อาจารย์ที่ปรึกษา

(รศ.ดร. มนัส สังวรศิลป์)

วันที่ 7 / ธค / 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบลายนิ้วมือระบุบุคคล

นาย ขวัญชัย เรืองบุญคุ้มสุข รหัส 43010040
นาย สุเมธ สำเร็จศิลป์ รหัส 43010495
นาย แสนศักดิ์ บวรจิโรภาส รหัส 43010508
รศ.ดร. มนัส สังวรศิลป์ (อาจารย์ที่ปรึกษา)
ปีการศึกษา 2546

บทคัดย่อ

โครงงานนี้ได้นำเสนอเครื่องสแกนลายนิ้วมือ ซึ่งสามารถแสดงภาพของลายนิ้วมือ แล้วนำภาพที่ได้นั้นไปใช้ตรวจสอบเพื่อระบุตัวบุคคลที่ต้องการ เครื่องตรวจสอบลายนิ้วมือนี้มีหลักการทำงานดังนี้คือ วางนิ้วมือลงบนตัวสแกน แล้วเครื่องจะทำการสแกนลายนิ้วมือโดยแปลงลายนิ้วมือนั้นให้เป็นรหัสดิจิตอลส่งเข้าคอมพิวเตอร์ จากนั้นให้โปรแกรมวิชวลเบสิกนำข้อมูลที่ได้มาทำการประมวลผลเพื่อสร้างเป็นภาพและตรวจสอบระบุตัวบุคคลต่อไป ซึ่งจะนำไปใช้ประโยชน์ในการตรวจสอบบุคคลก่อนที่จะเข้าสู่ระบบฐานข้อมูลในคอมพิวเตอร์ หรือใช้ในงานด้านการรักษาความปลอดภัย เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fingerprint Identify System

Mr. Kwanchai Rungbunkumsuk ID. 43010040
Mr. Sumate Samretsin ID. 43010495
Mr. Sansak Bowornjiropas ID. 43010508
Assoc.Prof.Dr.Manus Sangworrasilp (Advisor)
Education Year 2003

Abstact

This project, "Fingerprint Identify System", display fingerprint image on Computer which can be applied to identify each persons. The objective is for use in authentication to prove each person before using the system. It provides security. The process of this project begin with putting finger on Fingerprint Scanner and start it. The Fingerprint Scanner A/D converts and sent data to Computer. Visual Basic programming is create and analyzed the image.

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ได้ ก็เพราะได้รับความช่วยเหลือจากผู้ที่มีส่วนเกี่ยวข้องมากมายดังนี้ คือ รศ.ดร. มนัส สังวรศิลป์ อาจารย์ที่ปรึกษาโครงงานนี้ ซึ่งได้เอื้อเฟื้อทางด้านข้อมูล อุปกรณ์ต่างๆ ที่ใช้ในการทำโครงงานนี้ รวมทั้งการให้คำแนะนำปรึกษาเมื่อประสบปัญหา

ขอขอบคุณ รศ.ดร. มนัส สังวรศิลป์ เพื่อนๆ รุ่นพี่ และรุ่นน้องทุกคนที่คอยให้ข้อมูลเกี่ยวกับโครงงานนี้และคอยเป็นกำลังใจให้ผ่านพ้นปัญหาต่างๆ ไปได้ด้วยดี

ถ้าหากรายงานฉบับนี้มีข้อบกพร่องหรือความผิดพลาดประการใด คณะผู้จัดทำขออภัยไว้ด้วย และจะทำการแก้ไขปรับปรุงให้ถูกต้องในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
บทที่ 1 บทนำ	1
1.1 รายละเอียดโดยย่อของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 ประโยชน์หรือผลที่คาดว่าจะได้รับ	1
บทที่ 2 ความรู้เบื้องต้นของลายนิ้วมือ	2
2.1 ลักษณะลายนิ้วมือ	2
2.1.1 เส้นขอบ	2
2.1.2 เส้นคอน	2
2.1.3 จุดใจกลาง	3
2.1.4 บริเวณลายนิ้วมือที่อยู่ภายใน	3
2.2 ชนิดและรูปแบบลายนิ้วมือ	3
2.2.1 กลุ่มเส้นโค้ง	3
2.2.1.1 โค้งราบ	
2.2.1.2 โค้งกระโจม	
2.2.2 กลุ่มมัดหวาย	4
2.2.2.1 มัดหวายปัดขวา	
2.2.2.2 มัดหวายปัดซ้าย	
2.2.2.3 มัดหวายคู่	
2.2.3 กลุ่มก้นหอย	5
2.2.3.1 ก้นหอยธรรมดา	
2.2.3.2 ก้นหอยกระเป๋ากวางปัดขวา	
2.2.3.3 ก้นหอยกระเป๋ากวางปัดซ้าย	
2.2.3.4 ก้นหอยกระเป๋าข้างปัดขวา	
2.2.3.5 ก้นหอยกระเป๋าข้างปัดซ้าย	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 กลุ่มซับซ้อน 6

สารบัญ(ต่อ)

บทที่ 3 ทฤษฎีและหลักการเบื้องต้น 7
3.1 ระดับเกรย์ 7
3.2 การแปลงระดับเกรย์ 7
3.3 พื้นฐานและระบบของสีโมเดล 8
3.4 ความสว่าง 8
3.5 ไบนารีของรูปภาพ 9
3.6 อาร์จีบีของรูปภาพ 9
บทที่ 4 พอร์ทอนุกรม 11
4.1 การสื่อสารข้อมูลแบบอะซิงโครนัส 11
4.2 มาตรฐานพอร์ทอนุกรมแบบ RS-232 11
4.3 การเขียนโปรแกรมติดต่อกับ Serial Port 13
4.3.1 การติดต่อแบบอินเตอร์รัพท์ 13
4.3.2 การติดต่อแบบโพลลิ่ง 13
4.4 องค์ประกอบในการใช้ MSComm 14
4.4.1 การตั้งค่าติดต่อกับพอร์ท 14
4.4.2 การใช้ Buffer ในการรับส่งข้อมูล 14
4.4.3 ด้านฮาร์ดแวร์ 14
4.5 เรามาลองดูการกำหนดคุณสมบัติของ MSComm Control 14
4.5.1 Property ชื่อ CommPort 14
4.5.2 Property ชื่อ Setting 15
4.5.3 Property ชื่อ InputLen 15
4.5.4 Property ชื่อ PortOpen 15
4.5.5 Property ชื่อ Rthreshold 15
บทที่ 5 MBF200 Solid State Fingerprint Sensor 16
5.1 คุณสมบัติ 16
5.1.1 การนำไปใช้งาน 16
5.1.2 การทำงาน 16
5.2 การใช้งาน 17
5.3 Microprocessor Bus Interface 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 โฟลชาร์ท แสดงขั้นตอนการทำงานของ MBF200 21

สารบัญ(ต่อ)

บทที่ **6** การใช้งาน MCS-51 30

6.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 30

6.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ของตระกูล MCS-51 30

6.1.2 โครงสร้างภายในของ 8051 30

6.1.3 พอร์ทของ 8051 31

6.1.4 ผังเวลาของซีพียู (CPU Timing) 33

6.1.5 การแบ่งประเภทหน่วยความจำ 34

6.1.6 การสื่อสารทางพอร์ทอนุกรมของ MCS-51 35

6.1.6.1 รีจิสเตอร์ควบคุมการทำงานของพอร์ทอนุกรม

6.1.6.2 การสื่อสารระหว่างซีพียูหลายตัว

บทที่ **7** วิธีการหาจุด Core และ Delta 42

7.1 การแปลงภาพลายนิ้วมือเป็นภาพลายทิศทาง 43

7.2 การตรวจหาจุดเฉพาะ 43

7.2.1 ขั้นตอนที่ 1 การหาจุดที่คาดว่าน่าจะเป็นจุด core และ delta 43

7.2.2 ขั้นตอนที่ 2 การกำจัดจุดที่ไม่น่าจะใช่ทิ้ง 45

7.2.3 ขั้นตอนที่ 3 การตรวจสอบส่วนโค้ง 46

7.2.4 ขั้นตอนที่ 4 การรวมกลุ่ม 47

7.2.5 ขั้นตอนที่ 5 การหาแกนของพื้นที่ 48

7.2.6 ขั้นตอนที่ 6 การหาบล็อกทิศทางที่แทนจุด core และ delta 49

บทที่ **8** ส่วนควบคุมเซนเซอร์ 51

บทที่ **9** ผลการทดลอง 52

9.1 การปรับค่าต่างๆในรีจิสเตอร์ 52

9.2 การทำภาพให้บาง(core) 55

9.3 การหาจุดกลางของลายนิ้วมือ 56

9.4 การแปลงภาพเป็นภาพทิศทางและการเก็บข้อมูลใน Data Base 57

9.5 ระบบการตรวจสอบลายนิ้วมือ 59

บทที่ **10** สรุปและวิจารณ์ผลการทดลอง 60

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โครงงานนี้ เป็นการสร้างสแกนลายนิ้วมือ ซึ่งสามารถสแกนลายนิ้วมือของคนเราแล้วนำมาตรวจเช็คเพื่อระบุตัวบุคคล จะเห็นว่าสามารถนำไปใช้ประโยชน์ได้ เช่น นำมาใช้ในการตรวจหาบุคคลที่ต้องการ ใช้ในระบบรักษาความปลอดภัยในเวบไซต์ หรือใช้ในระบบต่างๆที่อาจใช้ระบบลายนิ้วมือเป็นตัวตรวจสอบ เป็นต้น

1.1 รายละเอียดโดยย่อของโครงงานนี้

โครงงานนี้เป็นการสร้างเครื่องสแกนลายนิ้วมือ โดยมี MBF200 เป็นตัว Fingerprint Sensor สำหรับแปลงลายนิ้วมือซึ่งเป็นสัญญาณอนาลอกให้เปลี่ยนเป็นสัญญาณดิจิตอล แล้วส่งให้คอมพิวเตอร์ทำการประมวลผลเพื่อตรวจเช็คและแสดงผลต่อไป

1.2 วัตถุประสงค์ของโครงงานนี้

สร้างเครื่องสแกนลายนิ้วมือ โดยการนำภาพลายนิ้วมือที่ได้ไปทำการตรวจเช็คระบุตัวบุคคล หรือนำไปใช้ในระบบรักษาความปลอดภัย เช่น เปิดปิดประตูบ้าน หรือใช้ในการทำเป็น password แทนการพิมพ์ที่ใช้ในปัจจุบันนี้ ในการทำการค้าในอีคอมเมอร์ซ

1.3 ขอบเขตของโครงงานนี้

สร้างเครื่องสแกนลายนิ้วมือ ซึ่งสามารถตรวจสอบหาตัวบุคคลที่ต้องการและแสดงผลเป็นภาพบนจอ computer ได้

1.4 ประโยชน์และผลที่คาดว่าจะได้รับจากงานนี้

เนื่องจาก ปัจจุบันนี้ ภาพลายนิ้วมือที่ได้จากกล้อง CCD ซึ่งภาพที่ได้จะมีความละเอียดและความถูกต้องน้อย เราจึงสร้างเครื่องสแกนลายนิ้วมือขึ้นมาเพื่อให้ได้ภาพที่มีความละเอียดและถูกต้องมากขึ้น ซึ่งภาพที่ได้จะนำไปใช้ประโยชน์ต่อไป เช่นใช้ในการยืนยันบุคคลเปิดปิดประตูบ้าน หรือใช้ในการทำเป็น password แทนการพิมพ์ที่ใช้ในปัจจุบันนี้ ในการทำการค้าในอีคอมเมอร์ซ และสามารถนำไปใช้ได้ในอนาคตคือการใช้ลายนิ้วมือที่ได้จากการสแกนแทนการพิมพ์ลายนิ้วมือในการทำบัตรประชาชน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้เบื้องต้นของลายนิ้วมือ

2.1 ลักษณะลายนิ้วมือ

ผิวหนังบริเวณปลายนิ้วมือ ประกอบด้วยลายเส้นสองชนิด ชนิดหนึ่งเราเรียกว่าเส้นนูน (Ridges) อีกชนิดหนึ่งเรียกว่า รอยร่องหรือเส้นร่อง (Furrows) จะอยู่สลับกันไปตลอด โดยถ้าเราใช้หมึกดำทาบบนนิ้วมือ และกดนิ้วมือลงบนกระดาษขาวจะได้ลายเส้นสีดำและสีขาวสลับกัน เราเรียกเส้นสีดำว่า เส้นนูน และเรียกเส้นสีขาวว่า เส้นร่อง เราสามารถแยกลักษณะของลายนิ้วมือที่เป็นจุดสำคัญต่างๆ ดังนี้

2.1.1 เส้นขอบ (Type Line)

หมายถึงเส้นคู่ขนานคู่ในสุดซึ่งได้เดินขบวนกันมาพอสมควรแล้วแยกตัวออกจากกัน เพื่อโอบล้อมหรือพยายาม โอบล้อมบริเวณลายนิ้วมือที่อยู่ภายในและเส้นขอบไม่จำเป็นจะต้องเป็นเส้นยาวและราบเรียบติดต่อกันตลอด และอาจเป็นเส้นที่ขาดกลางเส้น แต่เราก็ยังจัดเป็นเส้นขอบด้วย

รูปที่ 2.1 แสดงตำแหน่งเส้นขอบ

2.1.2 เส้นดอน (Delta)

หมายถึง ลายเส้นในนิ้วมือ ซึ่งอยู่ตรงหน้าและใกล้ที่สุดกับกึ่งกลางของปากทางแยกเส้นขอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 แสดงตำแหน่งเส้นดอน

2.1.3 จุดใจกลาง (Core)

หมายถึง จุดใดจุดหนึ่งบนปลายเส้น หรือไหล่ของเส้นวกกลับที่อยู่บนสุด

รูปที่ 2.3 จุดใจกลาง

2.1.4 บริเวณลายนิ้วมือที่อยู่ภายใน (Pattern Area)

หมายถึง พื้นที่บริเวณภายในของลายนิ้วมือที่ถูกเส้นขอบโอบล้อมไว้

2.2 ชนิดและรูปแบบของลายนิ้วมือ

เราสามารถจัดรูปแบบลายนิ้วมือออกเป็น 4 กลุ่ม คือ

2.2.1 กลุ่มเส้นโค้ง (Arch)

2.2.1.1 โค้งราบ (Plain Arch = PA)

ลายเส้นวิ่งหรือไหลออกไปข้างหนึ่ง ไม่เกิดมุมแหลมหรือเส้นพุ่งขึ้นตรงกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 โค้งราบ

2.2.1.2 โค้งกระโจม (Tented Arch = TA)

ลายเส้นตรงกลางเกิดเป็นลายเส้นพุ่งขึ้นจากแนวนอนเป็นมุมแหลมหรือมุมฉาก

รูปที่ 2.5 โค้งกระโจม

2.2.2 กลุ่มมัดหวาย (Loop)

2.2.2.1 มัดหวายปัดขวา (Right Slant Loop = RSL)

มีสันดอนเพียงจุดเดียว มีเส้นวกหลักที่สมบูรณ์อย่างน้อย 1 เส้น มีทิศทางไปด้านขวา

2.2.2.2 มัดหวายปัดซ้าย (Left Slant Loop = LSL)

มีสันดอนเพียงจุดเดียว มีเส้นวกหลักที่สมบูรณ์อย่างน้อย 1 เส้นมีทิศทางไปด้านซ้าย

2.2.2.3 มัดหวายคู่ (Double = D)

มีลักษณะคล้ายกับลายนิ้วมือแบบมัดหวายข้างบนแต่มากอดหรือกล้ำกันจนเกิดมีสันดอน 2 จุด โดยไม่จำเป็นต้องมีขนาดเท่ากัน ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.6 แสดงลายนิ้วมือแบบมัดหวาย

2.2.3 กลุ่มก้นหอย (Whorl)

ลายนิ้วมือที่มีเส้นเวียนรอบเป็นวงจร ลักษณะคล้ายรูปไข่ วงกลม หรือมีลักษณะอื่นๆ ประกอบด้วย

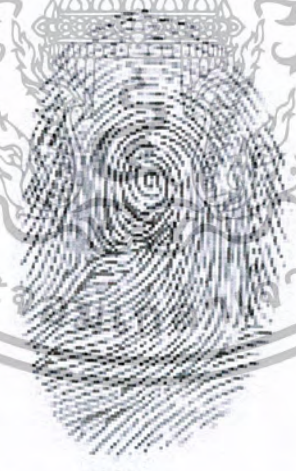
2.2.3.1 ก้นหอยธรรมดา (Plain Whorl = W)

2.2.3.2 ก้นหอยกระเป๋ากลางปัดขวา (Right Central Pocket = RCP)

2.2.3.3 ก้นหอยกระเป๋ากลางปัดซ้าย (Left Central Pocket = LCP)

2.2.3.4 ก้นหอยกระเป๋าข้างปัดขวา (Right Lateral Pocket = RLP)

2.2.3.5 ก้นหอยกระเป๋าข้างปัดซ้าย (Left Lateral Pocket = LLP)



รูปที่ 2.7 แสดงลายนิ้วมือแบบก้นหอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 กลุ่มซับซ้อน (Accidental Whorl = AW)

ลายนิ้วมือที่มีลักษณะพิเศษที่ไม่สามารถจัดเข้ากับกลุ่มใดกลุ่มหนึ่งได้มักประกอบจากลายนิ้วมือ 2 กลุ่มมาผสมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีและหลักการเบื้องต้น

3.1 ระดับเกรย์ (Gray Scale)

หากเราต้องการค่าของข้อมูลที่ละเอียดมากขึ้น จะต้องเพิ่มจำนวนบิตในการแสดงค่าของแต่ละพิกเซล เช่น ถ้าต้องการแบ่งความเข้มของการส่องสว่างให้มี 4 ระดับ ก็ต้องใช้เลขฐานสอง 2 บิต , 4 บิต สำหรับ 16 ระดับ และ 8 บิต สำหรับ 256 ระดับ ซึ่งจำนวนระดับที่ใช้ในระดับเกรย์นี้ มักเป็นเลขยกกำลังของ 2 ส่วนค่าต่ำสุดคือ 0 ถูกกำหนดให้เป็นสีดำ และ 1 หรือตัวเลขที่น้อยกว่าค่าสูงสุดของระดับเกรย์อยู่ 1 (เช่น 15 สำหรับระดับเกรย์ 16 ระดับ) ถูกกำหนดให้เป็นสีขาว ค่าที่กำหนดให้ในแต่ละพิกเซลมักเป็นจำนวนเต็ม

Gray Scale		Gray Value Range
2^1	2 values	0,1
2^3	8 values	0 to 7
2^4	16 values	0 to 15
2^8	256 values	0 to 255

ในยุคแรกๆของระบบการมองเห็นภาพ (Vision System) จะใช้ระบบเลขฐานสอง แต่ในปัจจุบันเทคโนโลยีของไมโปรเซสเซอร์ (Microprocessors) เข้ามามีบทบาทมากขึ้น ซึ่งในไมโครโปรเซสเซอร์จะมีหน่วยความจำอย่างน้อย 8 บิต เพราะฉะนั้นการแบ่งระดับเป็น 16 , 64 หรือ 256 จึงเป็นเรื่องธรรมดา และในตอนนี้กำลังจะมีมากกว่า 256 ระดับ แต่ในการมองเห็นของมนุษย์ สามารถแยกความแตกต่างได้เพียง 10 ถึง 15 ระดับเท่านั้น ดังนั้นการแบ่งโดยละเอียดเป็น 64 หรือ 256 ระดับ ทำให้มนุษย์ไม่สามารถแยกความแตกต่างได้ จึงนำไปประยุกต์ใช้กับงานการประมวลผลภาพ หรือ ใช้ในขบวนการทางอุตสาหกรรมต่างๆ

จึงอาจกล่าวได้ว่าจำนวนระดับเกรย์เป็นตัวจำกัดรายละเอียดของภาพ ยิ่งแบ่งระดับเกรย์เป็นหลายระดับ ก็จะยิ่งเพิ่มคุณภาพของภาพนั้นด้วย และการเพิ่มจำนวนพิกเซล เช่น จาก 32 x 32 เป็น 250 x 250 ก็จะเป็นการเพิ่มความละเอียด (Resolution) และรายละเอียด (Detail) ของภาพ ซึ่งจะแตกต่างกับการขยายภาพ (Zoom) เพราะการขยายภาพ เป็นการเพิ่มขนาดของแต่ละพิกเซลให้ใหญ่ขึ้น ไม่ได้เป็นการเพิ่มจำนวนพิกเซล แต่การแบ่งระดับเกรย์เป็นการเพิ่มจำนวนของพิกเซล

3.2 การแปลงระดับเกรย์ (Gray-level Transformation)

1. การเปลี่ยนแปลงความสว่าง (Global Alternative in brihtness)

เป็นการใช้ค่าคงที่บวกหรือลบออกจากทุกพิกเซลของภาพ เพื่อเพิ่มหรือลดความสว่างของภาพ

2. การทำเทรชโฮล (Threhold)

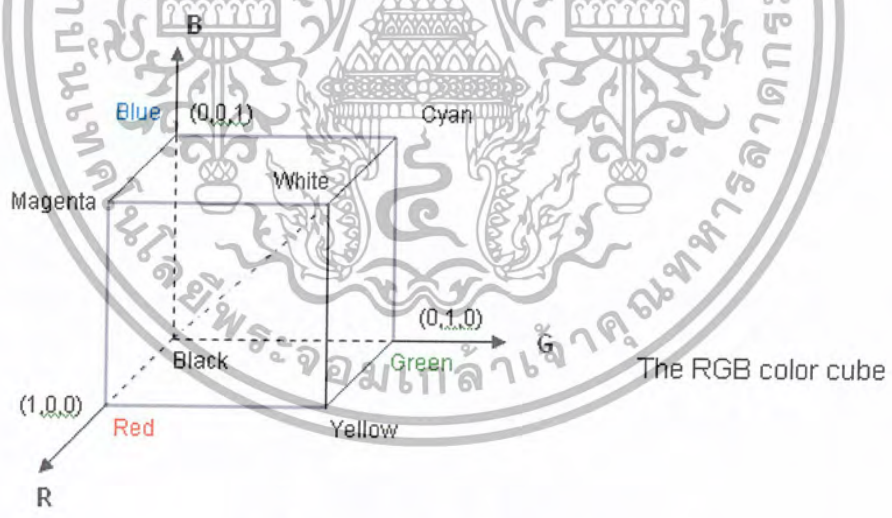
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการเปลี่ยนแปลงหรือหาแนวโน้มของค่าระดับเกรย์ในภาพเพื่อทำให้เป็นดีสครีท (Discreet) มากขึ้น โดยจะนำภาพมาฮิสโตแกรม แล้วกำหนดค่าระดับเกรย์ที่แน่นอนขึ้นมาเพื่อที่จะเป็นจุดตัดทำเทรชโฮล จากนั้นจะทำการตัดหรือปัดส่วนของระดับเกรย์ที่เราไม่ต้องการออกไป

ประโยชน์ของเทรชโฮล ได้แก่ การทำให้เป็นภาพแบบไบนารี จุดที่อยู่ในช่วงทางซ้ายมือทั้งหมดของจุดตัดเทรชโฮล T จะถูกทำเป็นสีขาว และจุดในช่วงทางขวามือทั้งหมดจะถูกทำให้เป็นสีดำ หรือ การช่วยให้หาขอบของภาพได้ง่ายขึ้น เป็นต้น แต่ทั้งนี้ก็เป็นการยากที่จะกำหนดจุดตัดเทรชโฮลที่ดีที่สุดออกมาได้

3.3 พื้นฐานและระบบของสีโมเดล RGB (Red , Green , Blue)

ในโมเดลนี้ สีแต่ละสีจะอยู่ในรูปของสีปฐมภูมิ (แดง , เขียว , น้ำเงิน) โมเดลนี้มีโครงสร้างอยู่ในระบบของพิกัดคาร์ทีเซียน (Cartesian coordinate) มีลักษณะเป็นทรงลูกบาศก์ ค่าสีแดง , เขียว , น้ำเงิน จะอยู่ที่มุมทั้งสาม บนเส้นทแยงมุมตรงข้ามกัน และค่าสีกรม , มาเกนตา , เหลือง จะอยู่ที่มุมทั้งสามในลักษณะเดียวกัน ส่วนสีดำจะอยู่ที่จุดกำเนิด สีขาวอยู่ที่มุมที่มีระยะทางไกลที่สุดจากแหล่งกำเนิด ค่าของระดับสีเทา (Gray Scale) จะอยู่บนเส้นที่เชื่อมระหว่างสีดำกับสีขาว ค่าสี คือ จุดที่อยู่บนผิวหรือในลูกบาศก์ถูกกำหนดค่าโดยเวกเตอร์ที่ชี้ออกจากจุดกำเนิด เพื่อความสะดวกเราจะสมมติให้ค่าสีถูกนอมอลไลซ์ (Normalize) ทั้งสามสี โดยให้มีค่าอยู่ระหว่าง 0 ถึง 1 ลูกบาศก์จึงเป็นลูกบาศก์หนึ่งหน่วย



รูปที่ 3.1 ระนาบสีของรูปภาพอาร์จีบี

ภาพในโมเดล RGB ประกอบด้วยภาพสามระนาบที่เป็นอิสระจากกัน สำหรับแต่ละสีปฐมภูมิเมื่อป้อนเข้าไปในมอนิเตอร์ (monitor) ที่เป็นแบบ RGB ภาพทั้งสามสีจะรวมตัวกันที่จอภาพกลายเป็นภาพสีผสม ดังนั้นการใช้โมเดล RGB ในการประมวลผลภาพนั้นจะสมเหตุสมผลเมื่อภาพถูกแยกออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยธรรมชาติให้อยู่ในเทอมของทั้งสามสี กล้องภาพสีส่วนใหญ่ที่ให้ภาพสีดิจิตอล จะอยู่ในรูปแบบของโมเดล RGB ดังนั้นโมเดลนี้จึงเป็นโมเดลที่สำคัญมากในการประมวลผลภาพ

3.4 ความสว่าง (Brightness)

ความสว่างเป็นสมบัติของแสงที่มนุษย์เป็นผู้กำหนดให้ระหว่างความสว่างมากที่สุดกับความมืด ความสว่างเป็นสิ่งที่รับรู้ได้แต่ไม่สามารถสัมผัสได้ ความสว่างสามารถรับรู้ได้เมื่อมีแสงตกกระทบกับเรตินารูปแท่งและรูปกรวยในตาของคนเรา ซึ่งจะก่อให้เกิดการตอบสนองที่ไม่เป็นเชิงเส้นและสลับซับซ้อน ความไวในการรับแสงของคนเราจะลดลงเมื่อแสงมีขนาดใหญ่มากขึ้น สำหรับความสว่างที่อยู่ในรูปภาพนั้น สามารถแสดงได้โดย Brightness Histogram ซึ่งเป็นกราฟการขยายตัวของกลุ่มตัวเลข ในกราฟนี้จะแสดงถึงระดับสีเทาของแต่ละพิกเซลที่อยู่ในรูปภาพ หรืออาจจะกล่าวได้ว่า เป็นการแสดงให้เห็นถึงจำนวนของพิกเซลในรูปภาพที่อยู่ในแต่ละระดับของสีเทาว่ามีอยู่เป็นจำนวนเท่าไร

3.5 ไบนารีของรูปภาพ

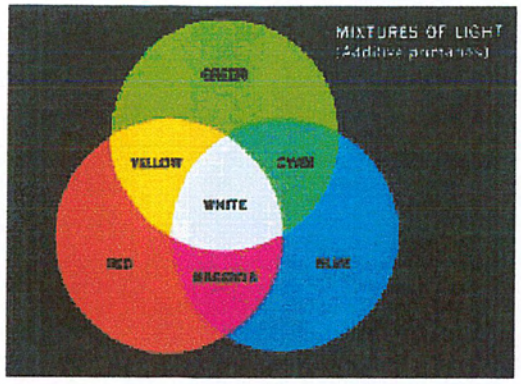
ในไบนารีของรูปภาพ แต่ละพิกเซลจะสมมติให้เป็นหนึ่งในสองค่าที่ไม่ต่อเนื่อง ค่าสองค่านี้จะเป็นลักษณะของเปิด หรือ ปิด ไบนารีของรูปภาพจะถูกเก็บในเมทริกซ์สองมิติของ 0 (ปิดพิกเซล) และ 1 (เปิดพิกเซล)

ไบนารีของรูปภาพถูกพิจารณาเป็นชนิดพิเศษของความหนาแน่นของรูปภาพ ซึ่งประกอบด้วยสีขาวและดำเท่านั้น อย่างไรก็ตาม เราสามารถคิดถึงไบนารีของรูปภาพเป็นลักษณะของดัชนีรูปภาพที่มีเพียงสองสีเท่านั้นก็ได้

3.6 อาร์จีบีของรูปภาพ

อาร์จีบีของรูปภาพ หรืออาจเรียกเป็น สีแท้จริงของรูปภาพ (truecolor image) เป็นอาร์เรย์ m x n x 3 ซึ่งจะกำหนดคอมโพเนนท์สีแดง , สีเขียวและสีน้ำเงินสำหรับแต่ละพิกเซล อาร์จีบีของรูปภาพไม่ได้ใช้แพทเลตของสี สีในแต่ละพิกเซลจะถูกกำหนดโดยการรวมกันของความหนาแน่นของสีแดง , สีเขียว และสีน้ำเงินที่เก็บอยู่ในแต่ละระนาบของสีที่ตำแหน่งของแต่ละพิกเซล รูปแบบไฟล์กราฟฟิกจะเก็บอาร์จีบีของรูปภาพเป็น 24 บิต ซึ่งแต่ละคอมโพเนนท์ของสีแดง , สีเขียว และสีน้ำเงิน จะมีขนาด 8 บิต ซึ่งจะทำให้เกิดจำนวนของสีที่เป็นไปได้ 16 ล้านสี เนื่องจากความละเอียดนี้เอง เราจึงสามารถแทนรูปภาพต่างๆ ที่พบเห็นได้ในชีวิตจริงด้วยรูปภาพแบบสีแท้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Additivity of primary colors

รูปที่ 3.2 แสดงแม่สีปฐมภูมิ

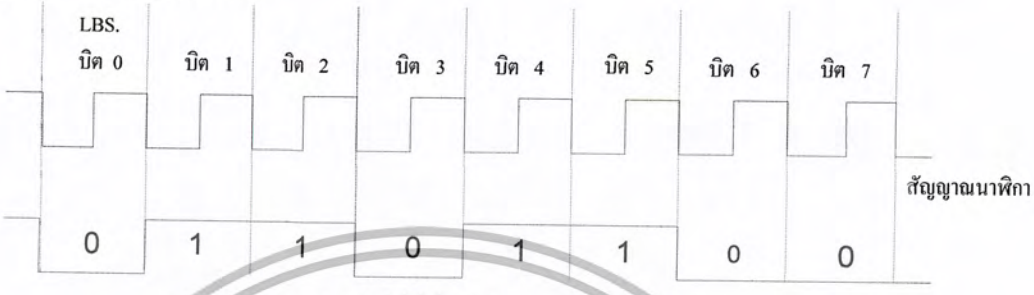
ในแต่ละสีระนาบสีที่แยกจากกันในรูปภาพประกอบไปด้วยพื้นที่ของสีขาว สีขาวนี้จะแสดงถึงค่าสูงสุดของแต่ละสีที่แยกจากกัน ตัวอย่างเช่น ในรูปภาพที่เป็นระนาบของสีแดง สีขาวจะแสดงถึงค่าสูงสุดของสีแดง ในขณะที่สีแดงถูกผสมด้วยสีเขียวหรือสีน้ำเงิน พิกเซลที่เป็นสีเทาจะปรากฏออกมา ส่วนพื้นที่ที่เป็นสีดำในรูปภาพนั้นจะแสดงค่าของพิกเซลที่ไม่มีค่าของสีแดงอยู่เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

พอร์ทอนุกรม

การสื่อสารแบบอนุกรมทำได้เป็น 2 แบบคือ แบบซิงโครนัสและอะซิงโครนัส แบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับส่งข้อมูลด้วย ซึ่งแบบซิงโครนัสนี้จะมีสายในการเชื่อมต่อ 3 เส้น คือสัญญาณนาฬิกา ข้อมูล กราวด์



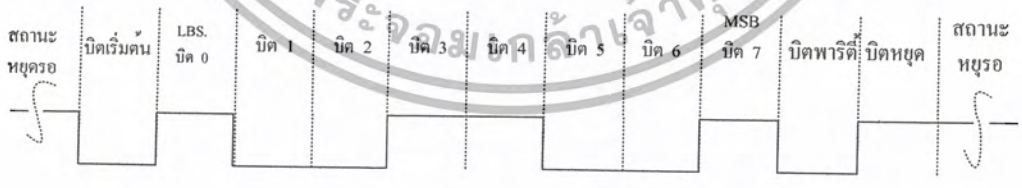
รูปที่ 4.1 แสดงการต่อแบบซิงโครนัส

4.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

เป็นการรับส่งข้อมูลโดยไม่ต้องมีสัญญาณนาฬิกามาร่วมอยู่ด้วย แต่ต้องกำหนดสัญญาณนาฬิกาทั้งภาครับ ภาคส่งให้มีค่าเท่ากัน ซึ่งเราเรียกว่า บอดเรทมีหน่วยเป็น บิตต่อวินาที

รูปแบบการรับส่งข้อมูลแบบอะซิงโครนัสประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start bit) ซึ่งมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5 6 7 หรือ 8 บิต
3. บิตตรวจสอบจะมีขนาด 1 บิต หรือไม่มี
4. บิตปิดท้ายจะมีขนาด 1,1.5,2 บิต



รูปที่ 4.2 แสดงข้อมูลแบบอะซิงโครนัส

4.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232

พอร์ตอนุกรมแบบ RS-232 เป็น IC สำหรับพอร์ตอนุกรมมีรายละเอียดการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Carrier : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะมีแอกทีฟเมื่อมีการส่งสัญญาณพาร์จากอุปกรณ์สื่อสารข้อมูลเช่นโมเด็ม

Receive Data : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่ได้ไว้ในรีจิสเตอร์บัฟเฟอร์

Transmitted Data : TD หรือ TxD ใช้ส่งข้อมูลจากคอมพิวเตอร์

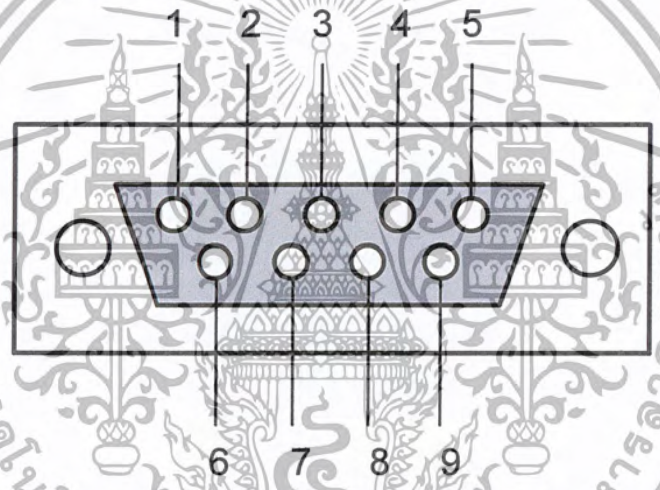
Data Terminal Ready : DTR คือ เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้

Request To Send : RTS คือ เป็นขาที่ส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งสัญญาณกลับมา

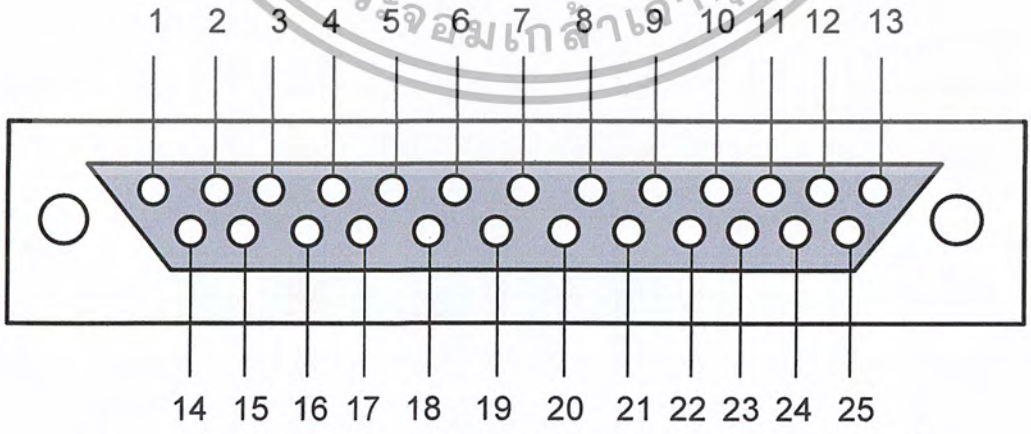
Data Set Ready : DSR เป็นขาที่รับสัญญาณจากภายนอก

Clear To Send : CTS ใช้เพื่อตรวจสอบว่าอุปกรณ์ภายนอกพร้อมรับสัญญาณหรือยัง

Ring Indicator : RI ใช้แสดงจากสถานะเรียกจากสายโทรศัพท์



(ก) คอนเน็คเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



(ข) คอนเน็คเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนเน็คเตอร์ DB-9	คอนเน็คเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุท
2	3	Received Data : RxD	อินพุท
3	2	Transmitted Data : TxD	เอาท์พุท
4	20	Data Terminal Ready : DTR	เอาท์พุท
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุท
7	4	Request To Send : RTS	เอาท์พุท
8	5	Clear To Send : CTS	อินพุท
9	22	Ring Indicator : RI	อินพุท

ตารางแสดงขาของพอร์ทอนุกรมทั้งแบบ 9 ขา และ 25 ขา

4.3 การเขียนโปรแกรมติดต่อกับ Serial Port สามารถทำได้ 2 วิธี คือ

4.3.1 การติดต่อแบบอินเตอร์รัพท์

ขบวนการอินเตอร์รัพท์ อุปกรณ์รอบข้างเกือบทุกชิ้นจะต้องปฏิบัติงานอยู่เพื่อส่งสัญญาณไปให้แก่ซีพียูเสมอ ถ้าอุปกรณ์นั้นพร้อมที่จะรับส่ง ที่เคยเจอจากการทำโครงงานอุปกรณ์ จะส่งเป็นรหัสแอสกี้ เราจะเขียนโปรแกรมอินเตอร์รัพท์ โดยเมื่อข้อมูลเข้ามาก็จะทำให้มี CommEvent กับ OnCommEvent

4.3.2 การติดต่อแบบโพลลิ่ง

ในระบบพีซี การโพลมีบ้างที่ใช้การส่งผ่านข้อมูลระหว่าง Terminal กับ CPU กรณีข้อมูลเป็นประเภทไบท์ที่ส่งจากคีย์บอร์ด โดยวิธีการนี้จะตรวจสอบ คีย์บอร์ดว่ามีข้อมูลส่งมาหรือเปล่า โดยจะตรวจสอบตลอดเวลา การทำงานกับข้อมูลที่รับเข้ามาจะตรวจสอบด้วยความเร็วที่สูงกว่าอัตราความเร็วข้อมูลที่ส่งเข้ามาทางคีย์บอร์ด การที่ CPU ส่งสัญญาณออกไปตรวจสอบพบว่ามีข้อมูลที่ต้องส่งเข้ามาเรียกว่า "Wet Poll" ซึ่งจะเสียช่วงเวลา 90 เปอร์เซ็นต์ คาบเวลาที่เสียไปนั้น เราเลี่ยงไปใช้เทคนิค การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพลแบบ "Round Robin" แทน แต่ใน VB เราจะใช้การตรวจสอบข้อมูลที่มาจาก Serial Port ตลอด โดยจะใช้ Control Timer เข้ามาช่วยในการเขียนโปรแกรมซึ่งสามารถตรวจสอบได้ถึงระดับ 1 มิลลิวินาที หรือจะใช้ Do...Loop ก็ได้ครับ

ในตัวคอนโทรล MSComm มี Event ที่ใช้เพียง Event เดียวเท่านั้นเอง ก็คือ OnCommEvent ซึ่งจะใช้ในการติดต่อแบบอินเตอร์รัพท์ การเขียนโปรแกรมติดต่อ Serial Port แบบธรรมดาจะใช้ comEvent เพียง comEvReceive,comEvSend ถ้าเป็นการติดต่อสื่อสารแบบโมเด็มจะใช้หลายตัวในการตรวจสอบสัญญาณ ผมไม่อยากแจ้งรายละเอียดเยอะเพราะมีใน Help Visual Basic อยู่แล้ว

4.4 องค์ประกอบในการใช้ MSComm

4.4.1 การตั้งค่าติดต่อกับพอร์ต

ComPort คือ เราต้องกำหนดหมายเลข Port ที่ใช้ต่อ RS-232 (Com1,Com2)

Setting คือ เราต้องกำหนดอัตรา Baud, Parity, Data (จำนวนบิต),Stop ตัวอย่าง 1200,n,8,1 เป็นต้น

HandShaking คือ เราจะกำหนดได้ 4 แบบ คือ 1.comNone 2. comXonXoff 3. comRTS 4. comTRSXonXoff

4.4.2 การใช้ Buffer ในการรับส่งข้อมูล

InBuffersize คือ การกำหนด Buffer ในการรับข้อมูลเข้ามา

OutBuffersize คือ การกำหนด Buffer ในการส่งข้อมูลออกไป

Rthreshold คือ การที่เรากำหนดการเกิด Event-driven ในการรับข้อมูลเข้ามา

Sthreshold คือ การที่เรากำหนดการเกิด Event-driven ในการส่งข้อมูลออกไป

Inputlen คือ จำนวนของข้อมูลที่ไปอ่านใน Buffer รับข้อมูล

EOFEnable คือ การที่บอกว่าสิ้นสุดของไฟล์ (EOF) End of file

4.4.3 ด้านฮาร์ดแวร์

ParityReplace คือ ค่าของคาแลกเตอร์ที่จะแทนในเมื่อเกิด Parity Error

NullDiscard คือ การกำหนดให้รับหรือไม่รับ NULL CHARACTER

RTSEnable คือ ทำให้มีสัญญาณ RTS (Request To Send)

DTSEnable คือ ทำให้มีสัญญาณ DTR (Data Terminal Ready)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 เรามาลองดูการกำหนดคุณสมบัติของ MSComm Control ให้สามารถติดต่อกับพอร์ตได้

4.5.1 Property ชื่อ CommPort คือ เลือกคอมพอร์ตที่เราจะต่อใช้งาน มาดูการเขียนโค๊ดกัน

ตัวอย่าง MSComm1.CommPort=1

ในที่นี้เลือกจะใช้ Com1 อยู่ที่ด้านหลังเครื่องคอมพิวเตอร์

4.5.2 Property ชื่อ Settings คือ การตั้งค่าของการรับส่งข้อมูล ซึ่งจะต้องรู้ด้วยว่าอัตราบอดของอุปกรณ์ที่จะติดต่อด้วยเป็นเท่าไร โดยมีรายละเอียดการใส่ค่าต่างๆ ดังนี้

MSComm1.Settings= "Buad (อัตราการรับส่งข้อมูล),Parity(ถ้าไม่ใช้ใส่ N,จำนวนบิตข้อมูล,บิตสต๊อบ)" มาดูการเขียนโค๊ดกัน

ตัวอย่าง MSComm1.Settings= "1200,N,8,1"

4.5.3 Property ชื่อ InputLen คือ กำหนดขนาดขณะที่มีข้อมูลเข้ามาให้ไปอ่านข้อมูลทั้งหมดที่อยู่ในบัฟเฟอร์ มาดูการเขียนโค๊ดกัน

ตัวอย่าง MSComm1.InputLen=1

4.5.4 Property ชื่อ PortOpen คือ จะเปิดให้พอร์ตใช้งานหรือไม่ ถ้าเปิด = True ถ้าปิด = False มาดูการเขียนโค๊ดกัน

ตัวอย่าง MSComm1.PortOpen=True

4.5.5 Property ชื่อ Rthreshold คือ ทำให้เกิดการกระตุ้นด้วย Event-driven เมื่อมีข้อมูลในบัฟเฟอร์รับข้อมูล (Comport) มันทำให้เกิด CommEvent ใน OnCommEvent มาดูการเขียนโค๊ดกัน

ตัวอย่าง MSComm1.Rthreshold=1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

MBF200

Solid State Fingerprint Sensor

5.1 คุณสมบัติ

- มีความละเอียด 500 phi
- มีขนาด 1.28 cm * 1.50 cm
- มีจำนวนแถว 300 แถว จำนวนหลัก256 หลัก
- ช่วงการทำงาน 3.3 V ถึง 5 V
- สแกนภาพเป็นแบบ Gray scale
- ได้ข้อมูล 8 bit จากการเปลี่ยน Analog เป็น Digital
- สามารถส่งข้อมูลออกได้ 3 ทางคือ 1. USB Port 2. Serial Pheripherral 3. Microprocessor
- ระบบสแกนลายนิ้วมือโดยอัตโนมัติ

5.1.1 การนำไปใช้งาน

- ในระบบรักษาความปลอดภัย
- ในบัตร Smart Card
- ระบุบุคคลสำหรับบัตร ATM และบัตรประชาชน
- ใช้เป็นรหัสผ่านใน Internet

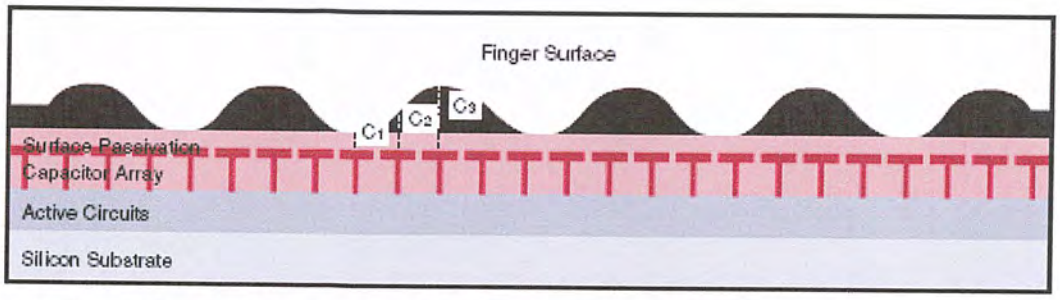
5.1.2 การทำงาน

ตัวเซนเซอร์จะมีขนาด 256 หลัก 300 แถว ในหน้าตัดเซนเซอร์ การทำงานจะเกี่ยวข้องกับวงจร sample and hold 2 ตัว โดยการทำงานจะจับภาพหรือส่งค่าทีละแถว เรียกว่า Row Capture ซึ่งจะประกอบด้วยการทำงาน 2 ขั้นตอน ในขั้นตอนแรกเซนเซอร์จะทำการกำหนดแถวที่จะทำการจับภาพ และจะนำกระแสที่ขา VDD มาอัดประจุไปที่พื้นผิวของแถวที่จะจับภาพ และคงค่านี้ไว้ เรียกขั้นตอนนี้ว่า pre – charge

ขั้นตอนที่ 2 แถวที่จับภาพจะทำการคายประจุคล้ายกับแหล่งจ่าย อัตราที่แผ่นเซนเซอร์จะคายประจุจะกำหนดได้โดยรีจิสเตอร์ DCR (Discharge current) หลังจากนั้นช่วงเวลาหนึ่ง (กำหนดโดยรีจิสเตอร์ DTR (Discharge Time)) และจะคงค่านี้ไว้ เรียกช่วงนี้ว่าช่วง discharge

ผลต่างระหว่างค่า pre-charge และ discharge จะเป็นค่าของความจุไฟฟ้าในแต่ละเซลล์ หลังจากเสร็จสิ้นการ Row Capture แต่ละเซลล์ในแถวจะถูกกระบวนการ A/D เพื่อแปลงค่าเป็นตัวเลขส่งเป็นข้อมูลออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- The difference in flow-out current generated by the difference between points of contact and non-contact on the finger is measured.
- Fingerprint information of 90,000 pixels is converted to 8-bit gray scale data.

รูปที่ 5.1 แสดงให้เห็นถึงเทคโนโลยีและการทำงานของ static-capacity semiconductor sensor หาก มองจากภาพตัดขวาง จะเห็นได้ว่ามีตัวคาปาซิเตอร์จำนวนมากถูกจัดวางเรียงอยู่ที่ ส่วนผิวชั้นบนของ ตัวชิป เมื่อนิ้วมือถูกวางลงบนผิว ค่าของคาปาซิเตอร์จะแปรเปลี่ยนไปตามสภาพความลึกตื้นของเส้น ลายนิ้วมือ จากนั้นค่าแต่ละค่าของกาปาซิเตอร์จะถูกแปลงโดย A/D Converter 8 bits เป็นภาพขนาด 300x256 พิกเซล ซึ่งจะให้ความละเอียดของภาพลายนิ้วมือได้ 500 phi

5.2 Pin Descriptions

VDDA1,VDDA2 (ขา 1 และ ขา 7)

เป็นขาจ่ายไฟใช้ +5V ให้กับส่วน Sensor

VSSA1,VSSA2 (ขา 2 และ ขา 6)

เป็นขากราวด์ให้กับส่วน Sensor

VDD1,VDD2,VDD3 (ขา 25, 16 และ 39)

เป็นขาจ่ายไฟใช้จ่าย +5V ให้กับส่วน Digital logic และ I/O

VSS1,VSS2,VSS3 (ขา 24, 15 และ 40)

เป็นขากราวด์ใช้ +5V ให้กับส่วน Digital logic และ I/O

ISET(ขา 3)

ต่อ R 200k ระหว่างขา VSSA1 กับ ISET เป็นขาอ้างอิงกระแสไฟฟ้าภายใน

AIN(ขา 4)

เป็นขาที่ใช้รับสัญญาณจากภายนอกมาเปลี่ยนจาก Analog เป็น Digital โดยสามารถควบคุมได้ จากกกำหนด AINSEL บิตในรีจิสเตอร์ CTRLA ถ้าไม่ใช้ต่อขานี้ลงกราวด์

FSET(ขา 5)

ต่อขานี้กับ R 56k ลงกราวด์ เพื่อให้ Clock ภายในทำงาน

XTIAL1(ขา 27)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

55463

เป็นขาที่ใช้สำหรับต่อ Cystal เพื่อกำเนิด clock แต่ถ้าใช้แหล่งกำเนิดภายนอกต่อขานี้กับแหล่งกำเนิดสัญญาณนั้น

XTIAL2(ขา 26)

เป็นขาที่ใช้สำหรับต่อ Cystal เพื่อกำเนิด clock แต่ถ้าใช้แหล่งกำเนิดภายนอกให้ขานี้ปล่อยลอย

D [7:0] (ขา 11-14 ,17-20)

เป็นขารับ-ส่งข้อมูล สามารถค้างค่าไว้ได้ ไม่ต้องใช้ขานี้ใน SPI , USB mode

A0 (ขา 21)

เป็นขา Address input A0 เป็น 0 (0 Volt) ใช้เป็น Address index register เป็น 1 (5 Volt) ใช้เป็น Data buffer ไม่ต้องใช้ขานี้ใน SPI , USB mode

RD (ขา 22)

เป็นขาใช้สำหรับนำข้อมูลออกจาก Chip ไปใช้งานให้เป็น 0 (Active low) ไม่ใช้ขานี้ใน SPI , USB mode

WR (ขา 23)

เป็นขาใช้สำหรับเขียนข้อมูลลง Chip ไปใช้งานให้เป็น 0 (Active low) ไม่ใช้ขานี้ใน SPI , USB mode

CSO/SCS (ขา 32)

เป็นขาใช้สำหรับเลือก Mode การทำงาน เมื่อใช้งานให้เป็น 0 (Active low) ใช้ขานี้จะใช้คู่กับขา MODE 1 , MODE 0

MODE[1:0] = 00b (Microprocecesor Bus Interface Mode)

CSO/SCS ให้เป็น 0 สำหรับเลือกเป็น อินพุท และให้ CS 1 เป็น 1

MODE[1:0] = 01b (SPI Slave Mode)

CSO/SCS ให้เป็น 0 สำหรับเลือกเป็น อินพุทต่อ R-Pull up ระหว่าง CSO/SCS กับ Vdd

MODE[1:0] = 10b (USB Interface Mode , Using Internal ROM)

ไม่มีหน้าที่ใช้งาน

MODE[1:0] = 11b ((USB Interface Mode , Using External ROM)

CSO/SCS สำหรับเลือกเป็นเอาพุทของ Chip ให้เป็น 0 สำหรับเลือก Serial ROM ต่อ R Pull up ระหว่าง CSO/SCS กับ Vdd

CS1/SCLK (ขา 31)

เป็นขาใช้สำหรับเลือก Mode การทำงาน เมื่อใช้งานให้เป็น 1 (Active hight) ใช้ขานี้จะใช้คู่กับขา

MODE 1 , MODE 0

MODE[1:0] = 00b (Microprocessor Bus Interface Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CS1/SCLK ให้เป็น 1 สำหรับเลือกเป็น อินพุท และให้ CSO/CSC เป็น 0

MODE[1:0] = 01b (SPI Slave Mode)

CS1/SCLK ให้เป็น 0 สำหรับเลือกเป็นเอาพุท

MODE[1:0] = 10b (USB Interface Mode , Using Internal ROM)

ไม่มีหน้าที่ใช้งาน

MODE[1:0] = 11b ((USB Interface Mode , Using External ROM)

CS1/SCLK ให้เป็น 0 สำหรับเลือกเป็นเอาพุท Serial clock ต่อ R Pull up ระหว่าง CS1/SCLK กับ Vdd

EXTINT (ขา 30)

เป็นขารับสัญญาณ Interrupt จากภายนอก

INTR (ขา 28)

เป็นส่งสัญญาณ Interrupt จากภายในออกไป

WAIT (ขา 29)

เป็นขาที่ส่งสัญญาณออกจากตัว Chip ขณะที่อ่านค่าจากลายนิ้วมือจะมีสถานะเป็น 0 เมื่ออ่านค่าเสร็จ (แปลงA/D) มีสถานะเป็น 1

MOSI (ขา 33)

เป็นขาเอาพุทของ SPI mode โดยใช้คู่กับขา MODE1 , MODE0 ไม่ต้องต่อขานี้ใน MCU mode , USB mode

MODE [1:0] = 00b (Microprocessor Bus Interface Mode)

ไม่มีหน้าที่ใช้งาน

MODE[1:0] = 01b (SPI Slave Mode)

ใช้เป็น Serial Data Output ของ Chip

MODE[1:0] = 10b (USB Interface Mode , Using Internal ROM)

ไม่มีหน้าที่ใช้งาน

MODE[1:0] = 11b ((USB Interface Mode , Using External ROM)

ใช้เป็น Serial Data Output ของ Chip

MISO (ขา 34)

เป็นขาอินพุทของ SPI mode โดยใช้คู่กับขา MODE1 , MODE0 ไม่ต้องต่อขานี้ใน MCU

MODE [1:0] = 00b (Microprocessor Bus Interface Mode)

ไม่มีหน้าที่ใช้งาน

MODE[1:0] = 01b (SPI Slave Mode)

ใช้เป็น Serial Data Input ของ Chip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE[1:0] = 10b (USB Interface Mode , Using Internal ROM)

ไม่มีหน้าที่ใช้งาน

MODE[1:0] = 11b ((USB Interface Mode , Using External ROM)

ใช้เป็น Serial Data Input ของ Chip จาก External ROM

P0 (ขา 9)

ขาเอาพุทถูกควบคุมโดย bit 0 ของ CTRLC register

P1 (ขา 10)

ขาเอาพุทถูกควบคุมโดย bit 1 ของ CTRLC register

DP (ขา 38)

เป็นขา Data ใน USB D+ ต้องต่อ R 1.5K ระหว่างขา DP และ VDD3 (ใช้3.3V – 3.6V) ใช้ R 43 โอห์ม ต่ออนุกรมกับขานี้ขึ้นไป หรือต่อลงกราวด์ใน MCU mode , SPI mode

DM (ขา 37)

เป็นขา Data ใน USB D-R 43 โอห์ม ต่ออนุกรมกับขานี้ขึ้นไฟ หรือต่อลงกราวด์ใน MCU mode , SPI mode

MODE [1:0] (ขา 35, 36)

MODE [1:0] = 00b (Microprocessor Bus Interface Mode)

MODE[1:0] = 01b (SPI Slave Mode)

MODE[1:0] = 10b (USB Interface Mode , Using Internal ROM)

MODE[1:0] = 11b ((USB Interface Mode , Using External ROM)

TEST (ขา 8)

ใช้ทดสอบ Chip ใช้สำหรับโรงงานตรวจสอบเท่านั้น

No Connect (ขา 41-80)

5.3 Microprocessor Bus Interface

ในการทำงานโหมดนี้ จะใช้ขา : D[7:0] , A0 , RD , WR , CSO , CS1 , EXTINT , INTR , WAIT และขา XTAL1/XTAL2 เป็นขาสำหรับกำเนิดความถี่

ภายใน Chip ตัวนี้มีขารับ-ส่งข้อมูล 8 ขา (D[7:0]) และมีขาสำหรับเลือกการทำงาน คือขาA0 เมื่อให้ขา A0 เป็น “0” เป็นการสั่งให้ชี้ Index Register ถ้าให้ขา A0 เป็น 1 เป็นการเขียนข้อมูลลง Index Register และมีขาอินพุทควบคุมอีก 4 ขา คือ CS0 , CS1 , RD ,WR

มีขาสถานะ 2 ขา คือขา INTR ,WAIT โดยที่ขา INTR ทำหน้าที่เป็นขาไว้ยืนยันเมื่อมีการเกิดการอินเตอร์รัพท์เกิดขึ้น และขา WAIT จะเป็น low เมื่อ A/D Converter ถูกอ่านขณะมันกำลังทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา WAIT และ INTR ทั้งคู่จะเป็น High Impedance ขณะมันไม่ทำงาน และจะเป็น low เมื่อมัน Active

โหมด SPI และ USB จะไม่ได้ใช้เมื่อกำลังทำงานอยู่ในโหมด Microprocessor bus interface ข้างล่างนี้เป็น Truth table ของโหมด Microprocessor bus interface

Truth Table for the Microprocessor Bus Interface

CS0	CS1	A0	RD	WR	Mode	Data Lines
H	X	X	X	X	De-selected	High Impedance
X	L	X	X	X	De-selected	High Impedance
L	H	X	H	H	Standby	High Impedance
L	H	L	L	H	Read Index Register	Output
L	H	L	H	L	Write Index Register	Input
L	H	H	L	H	Read Data Register	Output
L	H	H	H	L	Write Data Register	Input

5.4 Function Register

RAH 0x00

เป็นบิตสูงของตัวชี้ตำแหน่งแถวไว้ใช้สำหรับกำหนดแถวแรกที่จะทำการจับภาพในการ read mode sub-image

RAL 0x01

เป็นบิตต่ำของตัวชี้ตำแหน่งแถวไว้ใช้สำหรับกำหนดแถวแรกที่จะทำการจับภาพในการ read mode sub-image

CAL 0x02

เป็นตัวชี้ตำแหน่งหลักไว้ใช้สำหรับกำหนดหลักแรกที่จะจับภาพในการ read mode sub-image

REH 0x03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นบิตสูงของตัวชี้ตำแหน่งแถวไว้ใช้สำหรับกำหนดแถวสุดท้ายในการจับภาพในการ read mode sub-image

REL 0x04

เป็นบิตต่ำของตัวชี้ตำแหน่งแถวไว้ใช้สำหรับกำหนดแถวสุดท้ายที่จะทำการจับภาพในการ read mode sub-image

CEL 0x05

เป็นตัวชี้ตำแหน่งหลักไว้ใช้สำหรับกำหนดหลักสุดท้ายที่จะจับภาพในการ read mode sub-image

DTR 0x06

รีจิสเตอร์ควบคุมเวลาในการคายประจุ

DCR 0x07

รีจิสเตอร์ควบคุมการคายประจุของกระแส

CTRLA 0x08

เขียนที่รีจิสเตอร์นี้เพื่อสั่งให้เริ่มการทำงาน การอ่านเพื่อรับค่าจาก A/D Converter

การเขียนรีจิสเตอร์นี้เพื่อกำหนดโหมดการอ่านรูปภาพลายนิ้วมือโดยมี 3 โหมดคือ

1. GETIMG คือโหมดที่จะให้เซนเซอร์ทำการเริ่มต้นจับภาพตั้งแต่ แถวแรก หลักแรก จนจบแถวสุดท้าย ในคราวเดียวกัน
2. GETSUB คือโหมดที่ต้องกำหนดจุดเริ่มต้นและจุดสิ้นสุดในรีจิสเตอร์ RAH , RAL , CAL , REH , REL , CEL ก่อนเพิ่มทำการจับภาพในบริเวณที่กำหนด
3. GETROW คือโหมดที่จับภาพทีละแถว ต้องกำหนดแถวในรีจิสเตอร์ RAH , RAL เพื่อกำหแถวที่จะทำการจับภาพ

CTRLB 0x09

เป็นรีจิสเตอร์ที่ทำหน้าที่ปิดเปิดการทำงานของชิป

CTRLC 0x0A

เป็นตัวควบคุมขา Output ของ P0 และ P1

SRA 0x0B

เป็นรีจิสเตอร์ Read Only ขานี้ทำหน้าที่เป็นเงาของรีจิสเตอร์ CTRLA เพื่อทำการตรวจสอบว่าเซนเซอร์กำลังทำงานอยู่ในโหมดอะไร

PGC 0x0C

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมอัตราขยายของเซนเซอร์

ICR 0x0D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นรีจิสเตอร์ที่ทำหน้าที่เปิดปิดการอินเตอร์รัพท์

ISR 0x0E

เป็นรีจิสเตอร์ที่ทำหน้าที่บอกสถานการณ์อินเตอร์รัพท์

THR 0x0F

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมค่า Threshold ในสถานะอินเตอร์รัพท์ในโหมด Auto detection

CIDH 0x10

เป็นขา Read Only ทำหน้าที่เก็บค่ารุ่นของชิป

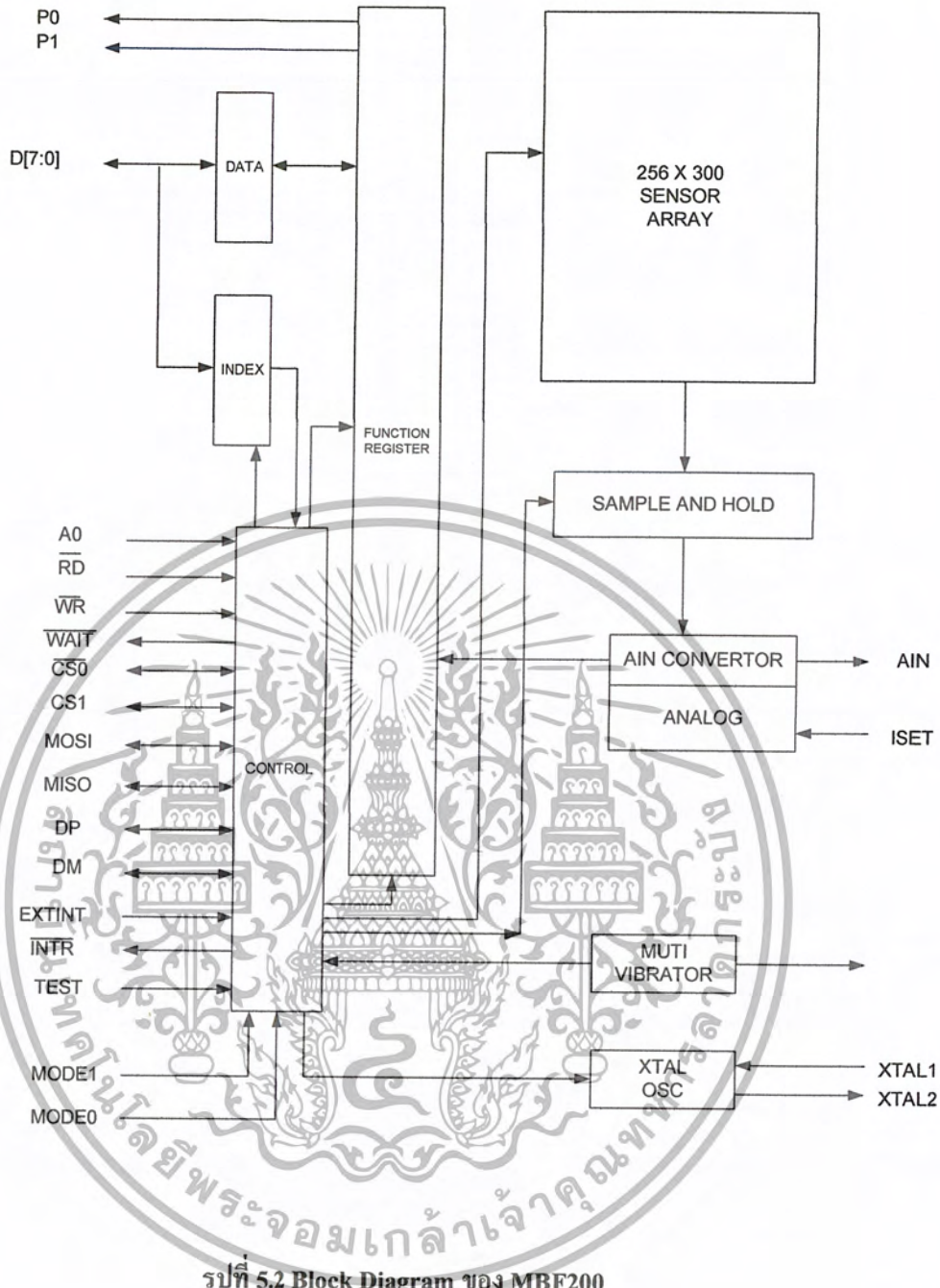
CIDL 0x11

เป็นขา Read Only ทำหน้าที่เก็บค่ารุ่นของชิป

TST 0x12

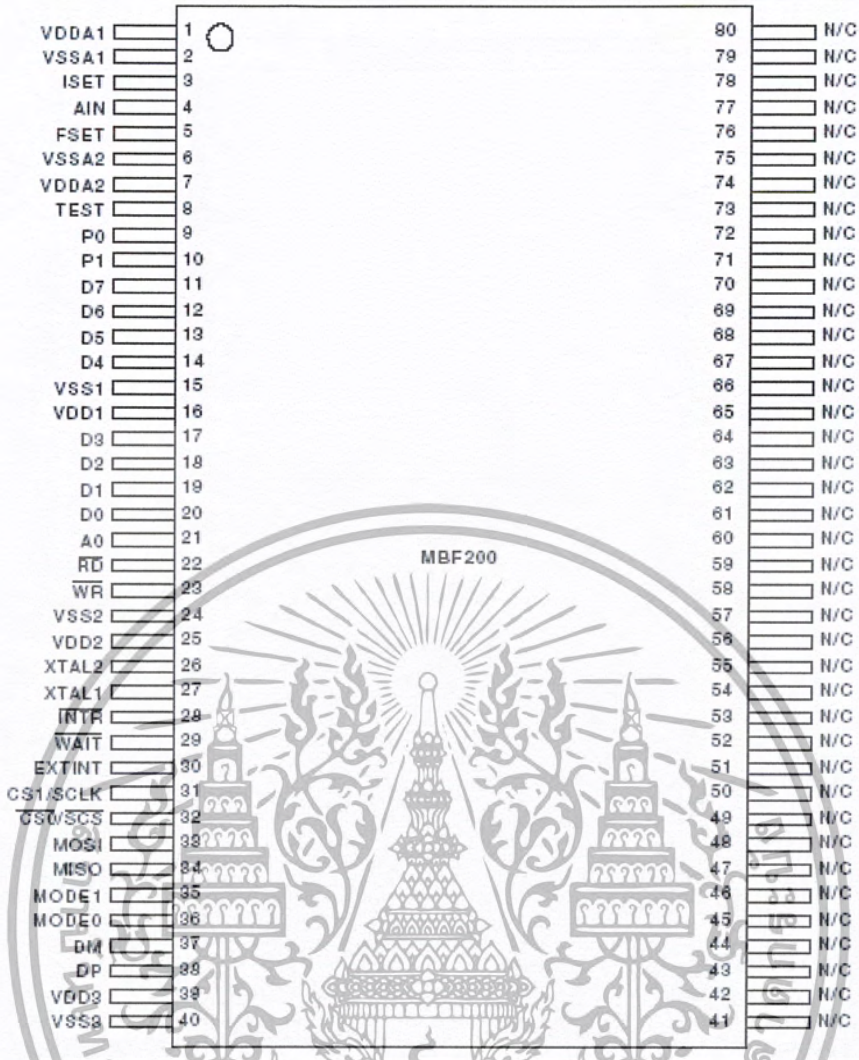
ใช้สำหรับ Test การทำงานของชิป สงวนไว้ใช้สำหรับโรงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



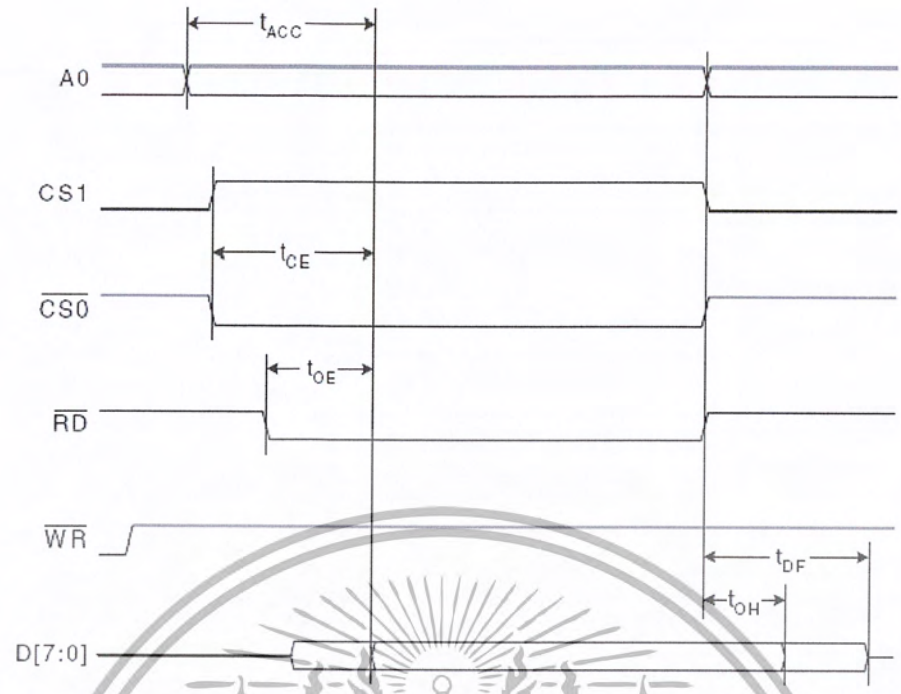
รูปที่ 5.2 Block Diagram ของ MBF200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



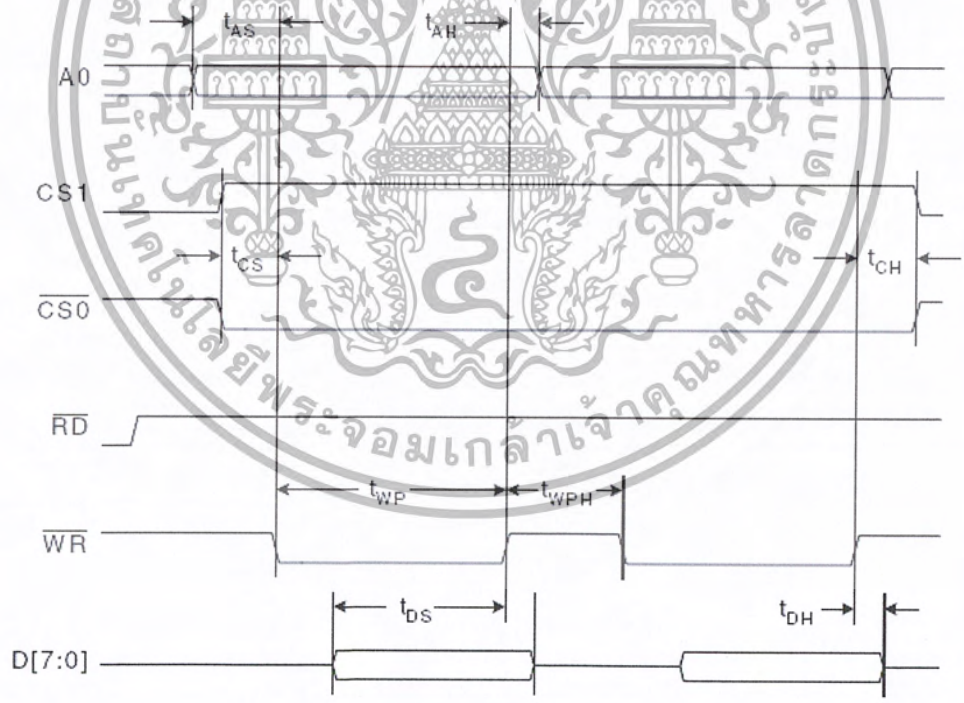
รูปที่ 5.3 แสดงขาของ MBF200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4

Microprocessor Mode Read Cycle

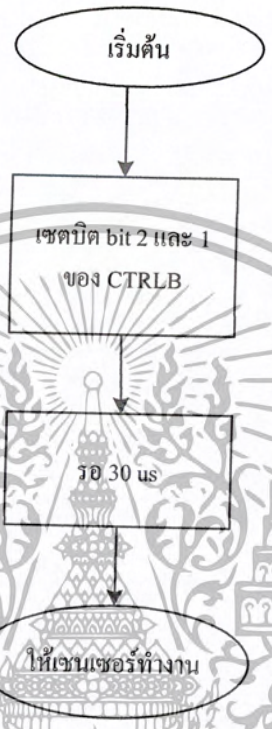


รูปที่ 5.5

Microprocessor Mode Write Cycle

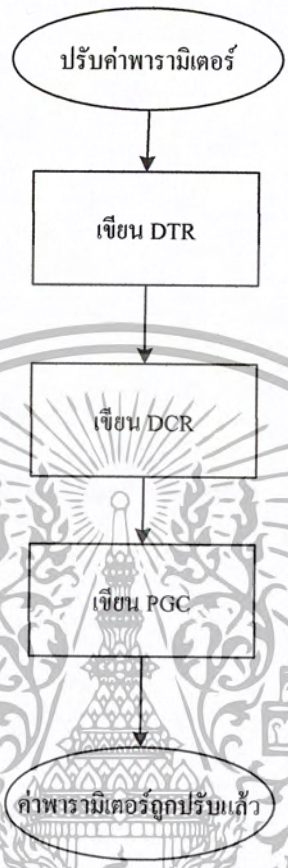
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 โฟลชาร์ท แสดงขั้นตอนการสั่งงานการทำงานของ MBF200



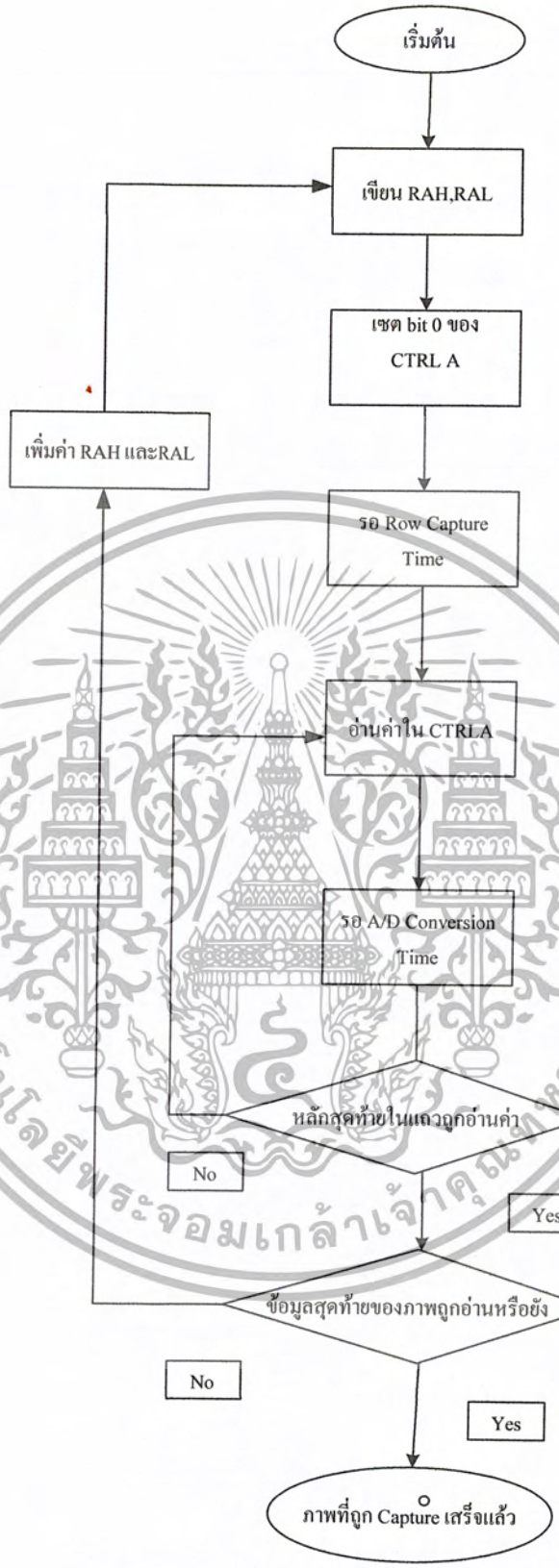
รูปที่ 5.6 แสดงการเลือกโหมดการทำงาน MBF200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 แสดงการค่าพารามิเตอร์ที่มีผลต่อความคมชัดของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.8 แสดงขั้นตอนการสแกนภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การใช้งาน MCS-51

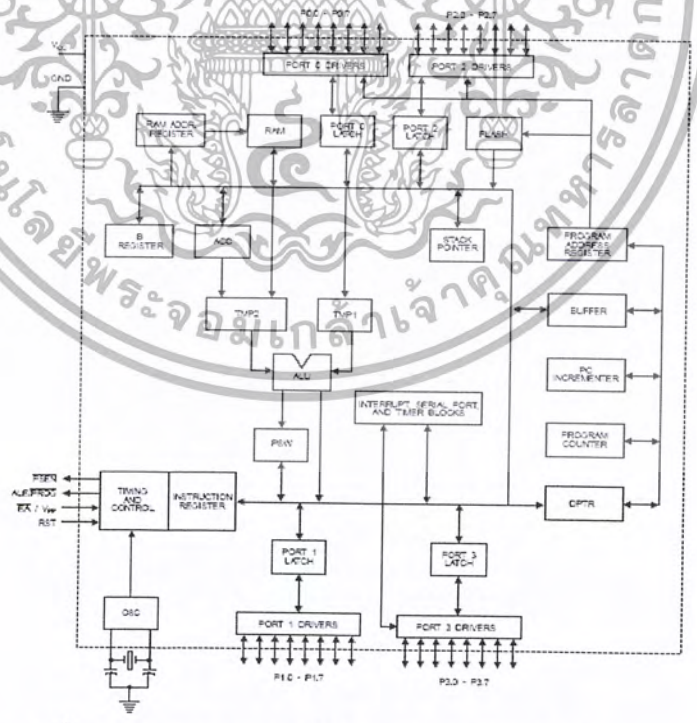
6.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

6.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ของตระกูล MCS-51

- ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 และ 8031 สำหรับเบอร์ 8052 มีหน่วยความจำถึง 8 ไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับเบอร์ 8052 ขึ้นไป มีถึง 256 ไบต์
- หน่วยความจำสำหรับโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
- มีไทเมอร์ เคาน์เตอร์ ขนาด 16 บิต 2 ชุด
- รับอินเตอร์รัพท์ได้ 6 แหล่ง 5 เวคเตอร์
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ตแบบ Full Duplex เลือกรูปได้ 4 โหมด
- มีคำสั่งในการทำ AND , OR หรือ Complement ได้ทั้งแบบ 8 บิต และ 1บิต

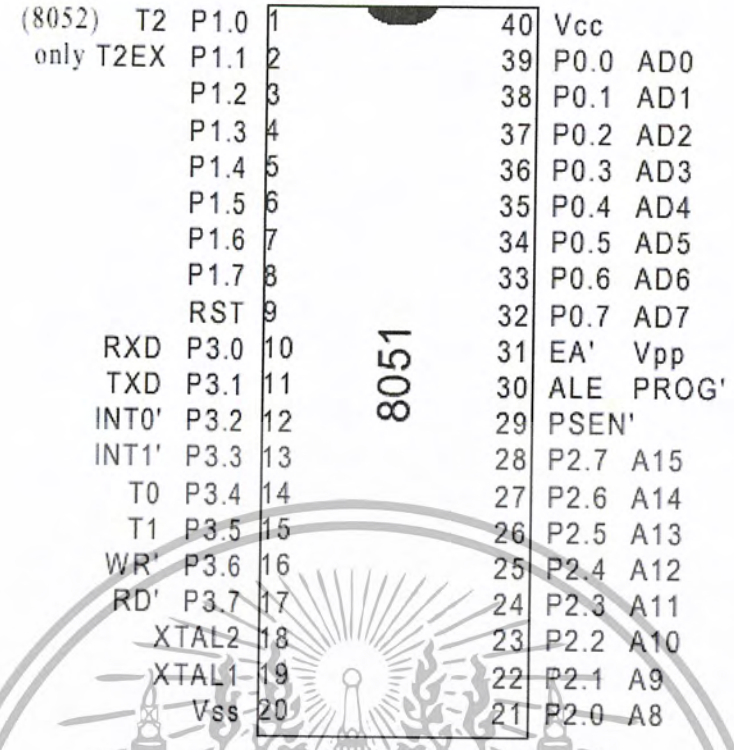
6.1.2 โครงสร้างภายในของ 8051

MCS-51 ใช้เทคโนโลยีในการผลิตเป็นแบบ NMOS และ CMOS โครงสร้างภายในสำหรับเบอร์ 8051 ดังแสดงในรูปที่ 6.1.1



รูปที่ 6.1.1 แสดง 8051 บล็อกไดอะแกรมของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1.2 แสดงตำแหน่งขาของ 89C2051

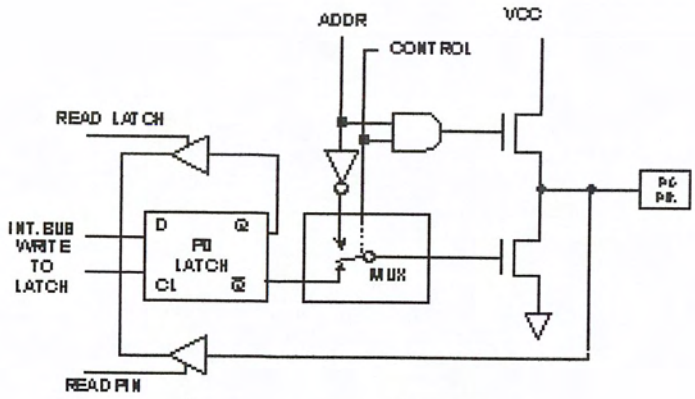
6.1.3 พอร์ตของ 8051

8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่างๆ ดังนี้

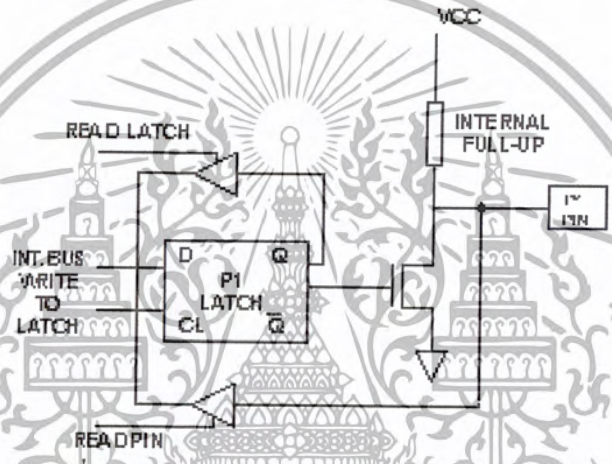
- Vcc (ขา 40) ต่อกับ +5V
- Vss (ขา 20) เป็นขา GND
- พอร์ท 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.0-P0.7) มีโครงสร้างแบบ Open-Drain Bi-directional ดังรูป

พอร์ท 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.0-P0.7) ใช้งานได้ 2 หน้าที่คือ แอดเดรสบัส และดาต้าบัสเมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือเป็นไอโอพอร์ท ถ้าต้องการให้ทำงานเป็นอินพุทพอร์ทต้องส่งลอจิก "1" ไปยังพอร์ทนี้ จะมีผลให้ Q ของ D-FF เป็น "1" ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุทพอร์ทแลทช์โดยส่งสัญญาณ READ LATCH ไปกระตุ้นที่ Tri-State Buffer ตัวบนและการอ่าน Port (pin) จะใช้สัญญาณ READ (pin)

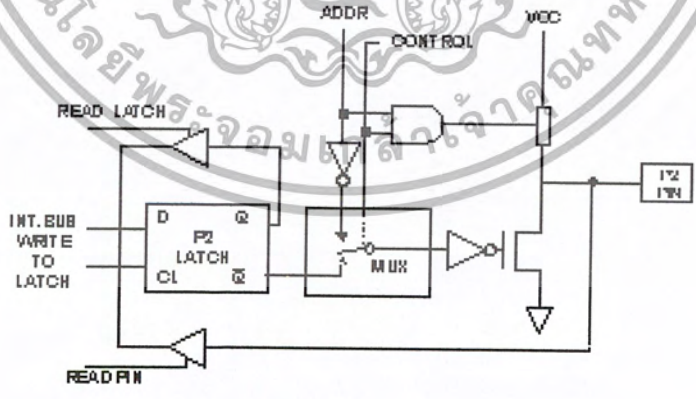
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1.3 แสดงโครงสร้าง พอร์ท 0

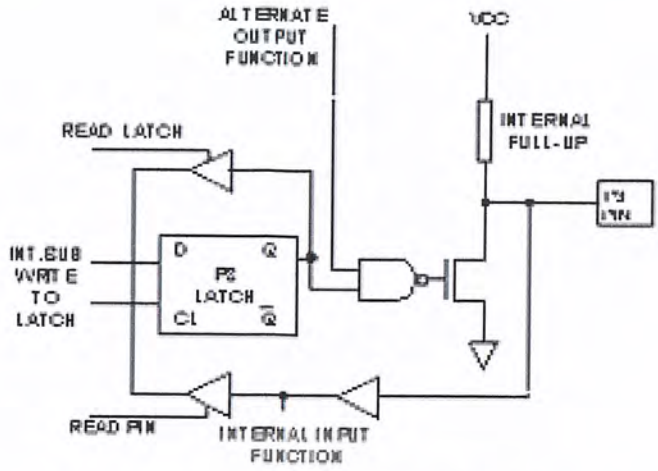


รูปที่ 6.1.4 แสดงโครงสร้าง พอร์ท 1



รูปที่ 6.1.5 แสดงโครงสร้าง พอร์ท 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1.6 แสดงโครงสร้าง พอร์ท 3

พอร์ท 1 (ขา 1-8) มีทั้งหมด 8 บิตคือ (P1.0-P1.7) มีโครงสร้างคล้าย พอร์ท 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน

พอร์ท 2 (ขา 21-28) มีทั้งหมด 8 บิตคือ (P2.0-P2.7) มีโครงสร้างคล้ายพอร์ท 0 โดยมี FET ตัวล่างตัวเดียว ส่วนด้านบนใช้ความต้านทานพูลอัพแทน พอร์ทนี้ทำงาน 2 หน้าที่ คือ สามารถใช้เป็นแอดเดรสบัสขนาด 8 บิต (A15-A8) และเป็น ไอโอพอร์ทใช้งานทั่วไปเมื่อจะใช้งานเป็นอินพุทพอร์ทต้องส่งลอจิก "1" มาที่พอร์ทนี้ก่อนเพื่อบังคับให้ FET อยู่ในสภาวะ OFF

พอร์ท 3 (ขา 10-17) มีทั้งหมด 8 บิต คือ ขา (P3.7-P3.0) มีโครงสร้างคล้ายพอร์ท 1 ทำงานได้ 2 หน้าที่ คือเป็น ไอโอพอร์ท ถ้าจะให้เป็นอินพุทต้องส่งลอจิก "1" มาที่พอร์ทนี้ก่อน และอีกหน้าที่หนึ่งคือใช้ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้ามา โดยสัญญาณต่างๆ มีดังนี้

- P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)
- P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม (UART)
- P3.2/INT0 (External Interrupt 0) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0
- P3.3/INT1 (External Interrupt 1) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1
- P3.4/T0 (Counter 0 External Input) ขารับสัญญาณพัลซ์อินพุทเข้าไปยังวงจร Counter 1
- P3.5/T1 (Counter 1 External Input) ขารับสัญญาณพัลซ์อินพุทเข้าไปยังวงจร Counter 0
- P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก
- P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก
- ALE (ขา 30) เป็นขาส่งสโตรบสำหรับใช้ในการแลตช์แอดเดรสไบต์ต่ำ (A7-A0) ที่ส่งออกมาจากพอร์ท (0) สัญญาณนี้จะแอคทีฟทุกๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PSEN (ขา 29) เป็นขาสโตรบที่ใช้สำหรับอ่านข้อมูลจาก Program Memory ภายนอก สัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขานี้
- EA (ขา 31) ใช้เลือกหน่วยความจำโปรแกรมภายนอก
  ป้อน "0" จะอ่านโปรแกรมจากภายนอกชิพ
  ป้อน "1" จะอ่านโปรแกรมจากภายนอกในชิพ
- RST (ขา 9) ขารีเซ็ท จะรีเซ็ทได้ก็ต่อเมื่อป้อนลอจิก "1" เข้าที่ขานี้นานอย่างน้อย 2 แมชชีนไซเคิล
- XTAL1 (ขา 19) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุทของวงจรออสซิสเลเตอร์ภายใน
- XTAL 2 (ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นเอาท์พุทของวงจรออสซิสเลเตอร์ภายใน

6.1.4 ผังเวลาของซีพียู (CPU Timing)

การทำงานใน 1 คำสั่งต่ำสุดกินเวลาเพียง 1 us ซึ่งจะใช้คล็อคไปเท่ากับ 12 ลูก โดยปกติซีพียูจะ RUN ด้วยความเร็วเท่ากับ 12 MHz ดังนั้น คล็อค 12 ลูกจะกินเวลาเท่ากับ 1 S

6.1.5 การแบ่งประเภทของหน่วยความจำ แบ่งได้ 2 ชนิด คือ

- หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory)

เป็นหน่วยความจำที่ใช้เก็บโปรแกรมสั่งงานบรรจุอยู่ในชิพ 8051 ส่วนที่เป็น Program Memory ก็คือ ROM ขนาด 4 กิโลไบต์นั่นเอง แต่ถ้าเป็นเบอร์ 8052 จะมี ROM ขนาด 8 กิโลไบต์

- หน่วยความจำสำหรับเก็บข้อมูล (Data Memory) แบ่งเป็น 2 ส่วนคือ หน่วยความจำข้อมูลภายในชิพมีเพียง 128 ไบต์ หน่วยความจำข้อมูลภายนอกชิพมีความจุ 64 กิโลไบต์
- พื้นที่หน่วยความจำที่เข้าถึงข้อมูลโดยทางอ้อมเท่านั้น (Indirect Address Area)

พื้นที่หน่วยความจำบริเวณ (80h-FFh) เป็นพื้นที่ซ้อนกันอยู่อย่างละ 128 ไบต์โดยส่วนแรกจะเป็น SFR แอดเดรสและ Indirect Address Area ดังนั้นถ้าต้องการติดต่อกับ SFR ต้องใช้คำสั่งแบบเข้าถึงข้อมูลโดยตรงเท่านั้น (Direct Address Area) ส่วนพื้นที่อีกส่วนจะเข้าถึงข้อมูลแบบทางอ้อมเท่านั้น (Indirect Address Area)

ส่วนตำแหน่ง (00h-7Fh) จะเข้าถึงข้อมูลได้ทั้ง 2 แบบ

พื้นที่หน่วยความจำที่เข้าถึงข้อมูลโดยตรงและทางอ้อม (Direct and Indirect Address Area)

พื้นที่ 128 ไบต์ ล่างสุดจะแบ่งเป็น 3 ส่วน คือ

1) รีจิสเตอร์แบงค์ (Register Bank 0-3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตั้งแต่ตำแหน่ง (00h-1Fh) เป็นส่วนของรีจิเตอร์แบงค์ (0-3) โดยแบ่งเป็นแบงค์ละ 8 ไบต์ รวมแล้วได้ 32 ไบต์ (แต่ละแบงค์ R0,R1,R2,R3,R4,R5,R6,R7) ถ้าซีพียูทำงานอยู่ที่แบงค์ 3 เมื่อถูกรีเซ็ทก็จะกลับมาทำงานที่แบงค์ 0 เสมอ และ SP จะมาเริ่มต้นที่ตำแหน่ง 07h ทันที

2) บริเวณหน่วยความจำที่ใช้คำสั่งอ่านเขียนทีละบิตได้ (Bit Address Area)
พื้นที่ตั้งแต่แอดเดรส (20h-2Fh) จำนวน 16 ไบต์หรือแบ่งเป็นบิตจะได้เท่ากับ 128 บิตซึ่งตำแหน่งบิตนี้มีดังนี้ 00,01,02,03,04,05,06,07 จนถึง 2Fh

3) บริเวณหน่วยความจำที่ใช้งานทั่วไป (Scratch Pad Area)

พื้นที่ตั้งแต่ (30h-7Fh) จะเขียนข้อมูลได้ทีละไบต์เท่านั้นไม่สามารถใช้คำสั่งเกี่ยวกับบิตได้ ถ้าย้ายเนื้อที่สแตคมาบริเวณนี้ต้องระวังในการเขียนข้อมูลมาทับสแตค

ไมโครคอนโทรลเลอร์เบอร์ 89C2051 เป็นประเภท CMOS 8 Bit ไมโครคอมพิวเตอร์ที่มีประสิทธิภาพการทำงานสูง มีส่วนของหน่วยความจำที่สามารถโปรแกรมใหม่และลบข้อมูลใน ROM ได้ ประเภท Flash Programmable and Only Memory (PEROM) จำนวน 2 กิโลไบต์ และ 4 กิโลไบต์ สำหรับเบอร์ 98C4051 ผลิตตรงตามมาตราฐานของไมโครคอนโทรลเลอร์ของตระกูล MCS-51 โดยบริษัท Atmel Corporation เป็นไมโครคอมพิวเตอร์ที่ราคาถูก มีประสิทธิภาพในการนำไปใช้งานได้หลายด้าน และหาซื้อได้ทั่วไปตามท้องตลาด ซึ่งคุณสมบัติโดยทั่วไปจะเหมือนกับไมโครคอนโทรลเลอร์เบอร์ 8051 ดังนั้นจึงได้รับการผลิตขึ้นมาให้มีส่วนของ Memory ที่สามารถโปรแกรมใหม่ได้โดยง่าย (PEROM) จึงเหมาะที่จะนำมาใช้เพื่อการพัฒนาเทคโนโลยีแบบใหม่

6.1.6 การสื่อสารทางพอร์ตอนุกรมของ MCS-51

การสื่อสารแบบอนุกรมเป็นการส่งข้อมูลครั้งบิตแบบต่อเนื่องกันไปโดยส่งบิตต่ำออกไปก่อนแล้วตามด้วยบิตสูง ซึ่งแตกต่างกับการส่งข้อมูลแบบขนานที่ส่งข้อมูลทุกบิตออกไปพร้อมกัน การส่งข้อมูลแบบขนานจะใช้สายสัญญาณ 1 เส้นต่อ 1 บิต แต่การส่งข้อมูลแบบอนุกรมจะใช้สายสัญญาณเพียงเส้นเดียวสำหรับข้อมูลทุกบิต

การส่งข้อมูลแบบอนุกรมมีโครงสร้างที่สำคัญหลายอย่างเช่นความเร็วในการส่งข้อมูลซึ่งเรียกว่า Baud rate มีหน่วยเป็นบิต/วินาที เช่นความเร็ว 1200,2400,4800 หรือ 9600 เป็นต้น หากใช้ความเร็ว 2400 บิต/วินาที จะใช้เวลาในการส่ง 1 บิตเท่ากับ 1/2400 วินาที ซึ่งสามารถคำนวณเวลาในการส่งข้อมูลได้

ภายในไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีพอร์ตการสื่อสารแบบอนุกรม (Serial Port) ที่สามารถรับและส่งข้อมูลแบบ Full Duplex อยู่ 1 พอร์ต การรับส่งแบบ Full Duplex หมายถึงว่าพอร์ตอนุกรมสามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน การควบคุมการทำงานของพอร์ตอนุกรมทางด้านส่งและรับข้อมูล ทำโดยการกำหนดค่าให้กับรีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมชื่อ SCON

6.1.6.1 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม SCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ SCON เป็นรีจิสเตอร์เฉพาะที่ทำหน้าที่ควบคุมโหมดการทำงานของพอร์ตอนุกรม และเป็นที่เก็บข้อมูลบิตที่ 9 ของการรับข้อมูล (บิต TB8 และ RB8) และมีแฟลกของการร้องขออินเตอร์รัพต์ของพอร์ตอนุกรมรวมอยู่ด้วย บิตต่างๆ การควบคุมการทำงานเราจะกำหนดบิตต่างๆ ในรีจิสเตอร์ตัวนี้ด้วย คำสั่งการโอนย้ายข้อมูล (MOV) หรือใช้คำสั่งการเซต (SET) หรือ เคลียร์ (CLR) บิตก็ได้เนื่องจาก รีจิสเตอร์ SCON อ้างอิงตำแหน่งแบบบิตได้

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SCON : Serial Port Control Register

รูปที่ 6.1.7 แสดงบิตของรีจิสเตอร์ SCON

ความหมายของบิตต่างๆในรีจิสเตอร์ SCON เป็นดังนี้

- SM0 และ SM1 เป็นบิตกำหนดโหมดการทำงานของพอร์ตอนุกรมซึ่งมี 4 โหมดดังนี้

SM0	SM1	โหมด	การทำงาน	อัตรารับส่ง
0	0	0	Shift Register	Fosc/12
0	1	1	8 bit UART	Variable
1	0	2	9 bit UART	F_{osc}/32 หรือ f_{osc}/64
1	1	3	9 bit UART	Variable

ตารางที่ 6.1 แสดงการเลือกโหมดการทำงานของพอร์ตอนุกรม

- SM2 เป็นบิตควบคุมให้ทำงานในลักษณะการเชื่อมต่อไมโครคอนโทรลเลอร์หลายตัวเข้าด้วยกัน สำหรับการใช้งานในโหมด 2 หรือ โหมด 3 เป็นดังนี้

SM2 = 1 จะทำให้แฟลกอินเตอร์รัพต์ทางด้านรับ (RI) ไม่ถูกเซตเมื่อรับข้อมูลเข้ามาแล้วมีค่าบิตที่ 9 เป็น 0 (อยู่ในบิต RB8) สำหรับการทำงานในโหมด 1 ถ้าเซต

SM2 = 1 แฟลกอินเตอร์รัพต์ทางด้านรับ (แฟลก RI) จะไม่ถูกเซตหากข้อมูลที่รับเข้ามาไม่มี STOP BIT การใช้งานในโหมด 0 ต้องกำหนด SM2= 0

- REN เซตหรือรีเซตด้วยซอฟแวร์เป็นตัวควบคุมการรับข้อมูลของพอร์ตอนุกรม ดังนี้

1 = ให้มีการรับข้อมูล

2 = ไม่ให้มีการรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

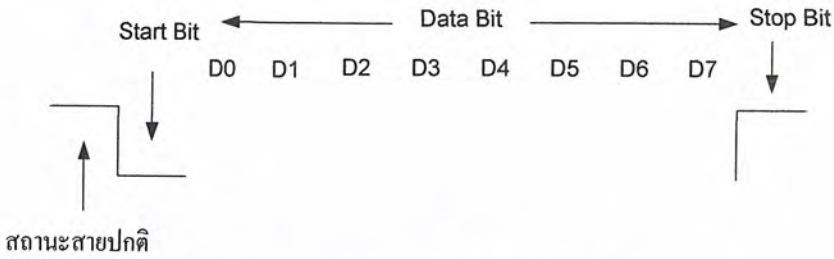
- TB8 เป็นบิตข้อมูลที่ 9 ที่ต้องการส่งในโหมด 2 และ 3 สามารถเซ็ตหรือเคลียร์ด้วยซอฟแวร์
- RB8 เป็นบิตเก็บข้อมูลที่รับเข้ามาบิตที่ 9 ในโหมด 2 หรือ 3 สำหรับการทำงานในโหมด 1 หากกำหนดให้ บิต SM2 = 0 บิต RB8 จะเป็นคาของ STOP BIT ที่รับเข้ามาสำหรับโหมด 0 ไม่มีการใช้ RB8
- TI แฟลกของการอินเตอร์รัพต์ด้านส่งข้อมูล แฟลกนี้จะถูกเซตด้วยฮาร์ดแวร์เมื่อจบการส่งข้อมูลบิตที่ 8 ในโหมด 0 หรือเมื่อเริ่มต้นส่ง STOP BIT ในโหมด 1,2 หรือ 3 เราต้องเคลียร์แฟลกนี้ด้วยซอฟต์แวร์ เมื่อจบโปรแกรมตอบสนองการอินเตอร์รัพต์ของการส่งข้อมูลแล้ว
- RI แฟลกอินเตอร์รัพต์ด้านรับข้อมูล ถูกเซตด้วยฮาร์ดแวร์เมื่อข้อมูลบิตที่ 8 ในโหมด 0 ถูกรับเข้ามาหรือเมื่อ STOP BIT ถูกรับเข้ามาในครึ่งเวลาแรกในโหมด 1,2 หรือ 3 เราต้องเคลียร์แฟลกนี้ด้วยซอฟแวร์เมื่อจบโปรแกรมตอบสนองการอินเตอร์รัพต์ของการรับข้อมูลแล้ว

การส่งและรับข้อมูลของพอร์ตอนุกรมจะรีจิสเตอร์ ที่ทำหน้าที่รับและส่งข้อมูลอยู่ 1 ตัวคือรีจิสเตอร์ SBUF การส่งข้อมูลออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ทำโดยการใส่ข้อมูลลงในรีจิสเตอร์ SBUF การอ่านข้อมูลจากภายนอกที่รับเข้ามาทางพอร์ตอนุกรมจะอ่านจากรีจิสเตอร์ SBUF เช่นกัน วงจรด้านรับจะมีบัพเฟอร์ขนาด 1 ไบต์ที่ทำหน้าที่เก็บข้อมูลที่รับเข้ามาประกอบอยู่ภายใน (การมีบัฟเฟอร์รับข้อมูลทำให้ด้านรับสามารถรับข้อมูลไบต์ที่ 2 เข้ามาได้ทันทีหลังจากข้อมูลไบต์แรกเข้ามาแล้วแม้ยังไม่ถูกอ่านออกไป แต่ถ้าข้อมูลไบต์ยังไม่ถูกอ่านก่อนที่ข้อมูลไบต์ที่ 2 จะเข้ามาครบข้อมูลไบต์ที่ 2 จะถูกยกเลิก)

การทำงานของพอร์ตอนุกรมของไมโครคอนโทรลเลอร์สามารถแบ่งออกได้เป็น 4 โหมดคือ โหมด 0,1,2 และ 3 แต่ละโหมดมีการทำงานดังต่อไปนี้

- โหมด 0 ข้อมูลขนาด 8 บิต แบบอนุกรมรับเข้ามาทางขา RXD และข้อมูล 8 บิต ส่งออกแบบอนุกรมทางขา TXD การรับและส่งข้อมูลจะเริ่มจากบิตต่ำ (LSB) ก่อนอัตราการรับส่งข้อมูล (Baud rate) จะเป็น 1/2 เท่าของสัญญาณนาฬิกา
- โหมด 1 ใช้การรับและส่งข้อมูลแบบ 10 บิต ซึ่งข้อมูลอนุกรม 10 บิตเข้ามาทาง RXD และข้อมูลขนาด 10 บิต ส่งออกทางขา TXD โดยข้อมูล10 บิตประกอบด้วย 1 Start bit (ค่า 0) ตามด้วย 8 บิตข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และ 1 Stop bit (ค่า 1) ด้านรับข้อมูลจะนำค่า Stop bit ที่รับเข้ามาไปเก็บในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON อัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ

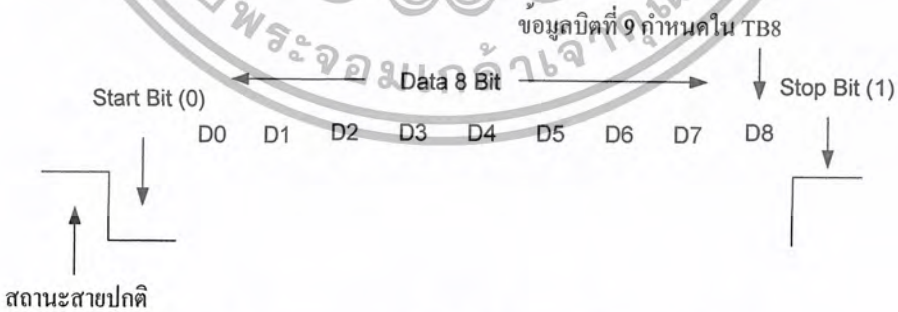
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.1.6 สัญญาณของการรับส่งข้อมูลของโหมด 1

- โหมด 2 ใช้การรับส่งข้อมูลขนาด 11 บิต ข้อมูลแบบอนุกรมรับเข้ามาทางขา RXD และส่งออกไปทางขา TXD ซึ่งข้อมูล 11 บิต ประกอบด้วย 1 Start bit (ค่า 0) ตามด้วย 8 บิตข้อมูล (การรับ/ส่งเริ่มจากบิตต่ำก่อน), 1 บิตข้อมูลที่กำหนดค่าได้ และ 1 Stop bit (ค่า 1) สำหรับด้านส่งข้อมูลบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ในรีจิสเตอร์ SCON ซึ่งเราสามารถกำหนดเป็น 1 หรือ 0 ก็ได้ ในการใช้งานเราอาจใช้บิตข้อมูลบิตที่ 9 เป็นบิตตรวจสอบ (Parity) ก็ได้โดยการนำค่าในแฟลก P มากำหนดให้กับ TB8 เมื่อข้อมูล 8 บิตมาจากแอคคิวมูเลเตอร์ ซึ่งจะทำให้เราได้การตรวจสอบการส่งข้อมูลแบบพาริตี้คู่ ในกรณีของการรับข้อมูลจะนำข้อมูลบิตที่ 9 ที่รับเข้ามาไปเก็บในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON ส่วน Stop bit จะไม่มีการนำมาเก็บ อัตราการรับและส่งข้อมูลในโหมดนี้จะเลือกใช้ความเร็วได้ 2 ค่าคือ 1/31 หรือ 1/64 เท่าของสัญญาณนาฬิกา
- โหมด 3 การทำงานในโหมด 3 จะมีลักษณะคล้ายกับโหมด 2 แต่ในโหมด 3 เราสามารถกำหนดอัตราการรับและส่งข้อมูลได้ตามต้องการ

การทำงานของพอร์ตอนุกรมทั้ง 4 โหมด การส่งข้อมูลจะเริ่มต้นเมื่อมีการกำหนดข้อมูลให้กับรีจิสเตอร์ SBUF ซึ่งการกำหนดค่าให้กับรีจิสเตอร์ SBUF จะใช้คำสั่งการโอนย้ายข้อมูล เช่น MOV SBUF,#15H และ MOV SBUF,@R1 ก็ได้ การรับข้อมูลในโหมด 0 จะเริ่มต้นรับข้อมูลเมื่อค่าของบิต RI = 0 และ REN = 1 ส่วนในโหมดอื่นๆ การรับข้อมูลจะเริ่มต้นเมื่อกำหนดบิต REN = 1 และมี Start bit เข้ามาที่ขา

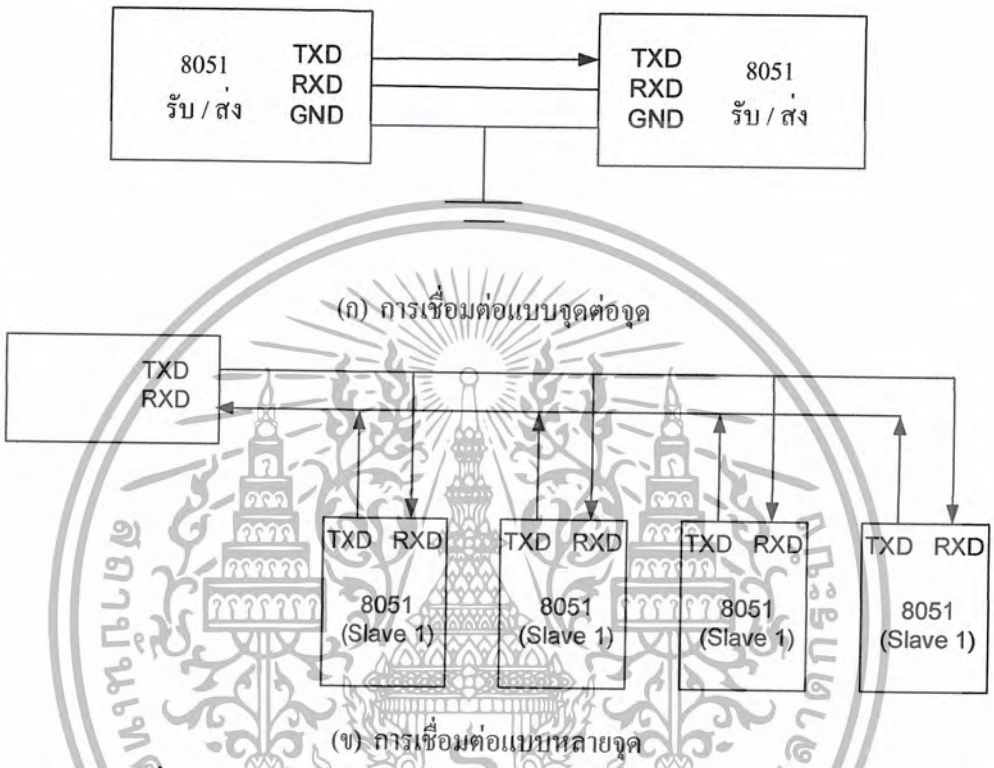


รูปที่ 6.1.7 สัญญาณของการรับส่งข้อมูลของโหมด 2 และโหมด 3

6.1.6.2 การสื่อสารระหว่างซีพียูหลายตัว (Multi Processor Communication)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อสื่อสารทั่วไปจะใช้การติดต่อสื่อสารแบบจุดต่อจุดซึ่งมีตัวส่ง 1 ตัวและตัวรับ 1 ตัว แต่ในบางครั้งการควบคุมจำเป็นต้องทำการสื่อสารแบบหลายจุด ซึ่งเป็นการสื่อสารระหว่างไมโครคอนโทรลเลอร์หลายตัวที่เชื่อมต่อเข้าด้วยกันเพื่อให้เกิดการควบคุมที่มีประสิทธิภาพมากขึ้น ไมโครคอนโทรลเลอร์เบอร์ 8051 สามารถเชื่อมต่อเพื่อรับส่งข้อมูลกันได้หลายตัว ดังแสดงในรูปที่ 6.1.8



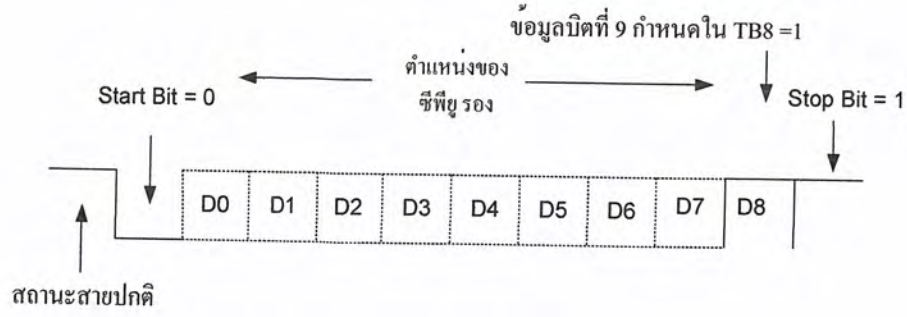
รูปที่ 6.1.8 การเชื่อมต่อไมโครคอนโทรลเลอร์แบบจุดต่อจุดและหลายจุด

การทำงานของพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 ในโหมด 2 และ โหมด 3 มีการทำงานพิเศษ ที่สามารถใช้สำหรับการเชื่อมต่อสื่อสารระหว่างซีพียูหลายตัวได้ การทำงานทั้ง 2 โหมดนี้ ข้อมูลบิตที่ 9 จะถูกนำไปเก็บในบิต RB8 เราสามารถโปรแกรมเพื่อกำหนดให้พอร์ตอนุกรมส่งสัญญาณร้องขออินเตอร์รัพต์ได้เมื่อรับข้อมูลเข้ามาแล้วได้ค่าใน RB8 = 1 ซึ่งการควบคุมนี้กำหนดจากการเซตบิต SM2 ที่อยู่ในรีจิสเตอร์ SCON การทำงานในการเชื่อมต่อสื่อสารกับซีพียูหลายตัวจะมีลักษณะการทำงานดังนี้ ในระบบที่เชื่อมต่อแบบหลายจุดเรากำหนดให้ซีพียูรอง (Slave) สัญญาณ TXD ของซีพียูหลักต่อเข้ากับสัญญาณ RXD ของซีพียูรองทุกตัว ทำให้ข้อมูลจากซีพียูหลักสามารถส่งไปยังซีพียูรองได้ทุกตัว ที่ตัวซีพียูรองจะมีตำแหน่งของตัวเอง (Address) ที่กำหนดไว้จากโปรแกรมหรือฮาร์ดแวร์ โดยมีการแบ่งลักษณะของข้อมูลที่ใช้ส่งออกเป็น 2 แบบคือ

1) ข้อมูลที่เป็นตัวกำหนดตำแหน่ง (Address Byte)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

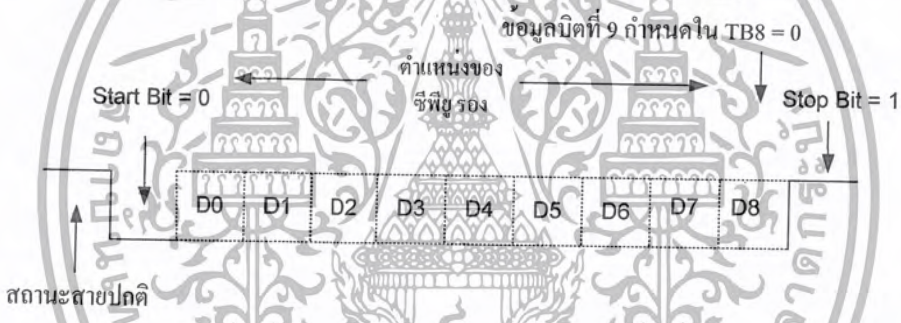
ข้อมูลที่เป็นตัวกำหนดตำแหน่ง จะมีคาบิตข้อมูลบิตที่ 9 (D8) มีค่าเป็น 1 ใช้สำหรับกำหนดตำแหน่งของซีพียูรอง ที่ซีพียูหลักต้องการติดต่อด้วย ลักษณะของข้อมูลที่กำหนดตำแหน่งแสดงในรูป



รูปที่ 6.1.9 แสดงข้อมูลกำหนดตำแหน่ง (Address Byte)

2) ข้อมูลที่ใช้เป็นข้อมูลข่าวสาร (Data Byte)

เป็นไบต์ที่ใช้สำหรับการรับและส่งข้อมูล และนำไปใช้งาน โดยข้อมูลไบต์นี้จะกำหนดให้บิตที่ 9 (D8) เป็น 0 ซึ่งรูปแบบของไบต์ข้อมูลจะเป็นดังรูป



รูปที่ 6.1.10 แสดงข้อมูลข่าวสาร (Data Byte)

เมื่อซีพียูหลักต้องการส่งข้อมูล 1 บล็อกไปให้ซีพียูรองตัวใดตัวหนึ่งในหลายๆตัว ขั้นแรกซีพียูหลักต้องส่งค่า 1 ไบต์ ซึ่งถือเป็นตัวบอกตำแหน่งของซีพียูรอง (Address byte) ออกไปก่อนเพื่อเป็นการเลือกซีพียูรองที่ต้องการให้รับข้อมูล โดยข้อมูล 1 ไบต์นี้จะแตกต่างกับข้อมูลอื่นๆ คือบิตที่ 9 มีค่าเป็น 1 ข้อมูลไบต์นี้ถูกส่งออกไปยังซีพียูรองทุกตัว หากเรากำหนดให้ซีพียูรองทุกตัวมีค่าบิต SM2 = 1 เมื่อซีพียูรองทุกตัวได้รับข้อมูลไบต์ตำแหน่งแล้วจะเกิดการอินเตอร์รัพต์ขึ้นภายในตัวซีพียูรอง ซีพียูรองจะทำการตรวจสอบว่าตำแหน่งที่รับมานั้นมีค่าตรงกับตัวเองหรือไม่ ซีพียูรองตำแหน่งที่มีตำแหน่งตรงกับตำแหน่งที่รับได้จะทำการเคลียร์บิต SM2 และเตรียมรับข้อมูลที่เป็นข่าวสารต่อไป สำหรับซีพียูรองตำแหน่งที่ไม่ตรงกับตำแหน่งที่รับมาจะยังคงค่า SM2 = 1 อยู่ต่อไป และจบโปรแกรมตองสนองการอินเตอร์รัพต์แล้วกลับไปทำโปรแกรมที่ค้างอยู่ และรอการอินเตอร์รัพต์ที่เกิดขึ้นจากการรับไบต์ตำแหน่งในครั้งใหม่อีก หลังจากซีพียูหลักส่งข้อมูลไบต์แรก ซึ่งเป็นตำแหน่งออกไปแล้ว จะเริ่มส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลซึ่งถือเป็นไบต์ข้อมูลมีบิตที่ 9 เป็น 0 ตามออกไป ข้อมูลนี้จะมีเพียงซีพียูที่มีตำแหน่งตรงกันเท่านั้นที่รับข้อมูลไว้ การส่งข้อมูลก็จะเป็นระหว่างซีพียูหลักกับซีพียูรองที่มีตำแหน่งตรงกันเท่านั้น เมื่อการส่งข้อมูลสิ้นสุดลงซีพียูหลักจะต้องส่งข้อมูลไบต์สุดท้ายที่ใช้เป็นตัวบอกให้ด้านรับทราบว่าสิ้นสุดข้อมูล ซึ่งอาจใช้รหัส ASCII ค่า 03 (ETX End of text) หรือ 04 (EOT End of transmission) ก็ได้ เมื่อซีพียูรองได้รับรหัสนี้แล้วก็จะทราบว่าเป็นข้อมูลตัวสุดท้ายและหยุดการรับข้อมูล แล้วเซตค่าบิต SM2 = 1 เพื่อเริ่มต้นรอรับการส่งข้อมูลที่เป็นตำแหน่งในรอบต่อไป

บิต SM2 จะไม่มีผลในโหมด 0 สำหรับการใช้งานในโหมด 1 หาก SM2 = 1 การอินเตอร์รัพท์ของการรับข้อมูลจะไม่เกิดขึ้นหากไม่มี Stop bit เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

วิธีการหาจุด Core และ Delta

มีขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1 การแปลงภาพลายนิ้วมือเป็นภาพลายทิศทาง

ในการแปลงภาพลายนิ้วมือที่ได้มาเป็นทิศทาง (Directional Image) ภาพลายนิ้วมือที่ได้มาจะประกอบด้วย ลายนูน (ridge) และร่อง(valley) โดยที่ทิศทางในแต่ละ pixel ของภาพจะแทนทิศทางการเดินของลายนูนและร่องรวมกันเป็นภาพลายนิ้วมือออกมา ภาพลายนิ้วมือที่นำมาแปลงเป็นภาพทิศทางนั้นเราให้ทิศทางแทนทิศทางการเดินของลายนูน โดยการตัดบล็อกขนาด 8x8 pixel (ในที่นี้ภาพลายนิ้วมือที่เราใช้จะเป็นภาพ grey image มีขนาด 256x256) ลงบนภาพลายนิ้วมือจนทั่วทั้งภาพ จะได้บล็อกทิศทางขนาด 32x32 บล็อก แล้วในแต่ละบล็อกนั้นเราจะทำการหาทิศทาง การที่ใช้บล็อกขนาด 8x8 pixel เนื่องจากว่าเราได้ทำการทดลองแล้วพบว่าเหมาะสมสามารถล้อมรอบลายนูนแต่ละลายได้การตัดบล็อกที่มีขนาดใหญ่กว่านี้สามารถทำได้เช่นกัน แต่ผลที่ได้นั้นคือในหนึ่งบล็อกนั้นจะมีจำนวนลายนูนสองลายหรือมากว่าซึ่งทำให้ทิศทางที่ได้ไม่ใช่ทิศทางที่แทนทิศทางการเดินของลายนูนส่วนถ้าบล็อกที่เราสร้างมีขนาดเล็กเกินไปแล้วก็จะทำให้บล็อกบางบล็อกไม่มีลายนูนปรากฏอยู่หรือบล็อกไม่สามารถล้อมรอบลายนูอนได้ผลของทิศทางที่ได้จากการตัดบล็อกลักษณะนี้ก็ไม่ได้ทิศทางที่แทนทิศทางการเดินของลายนูนอีกเช่นกัน ดังนั้นเราจึงได้ใช้บล็อกขนาด 8x8 pixel ต่อมาในแต่ละบล็อกเราจะหาทิศทางที่ลายนูนได้โดยการหาผลรวมของผลต่างของค่า grey value ตามทิศทาง d ที่มีค่าของผลรวมที่ได้น้อยที่สุด กำหนดทิศทางดังกล่าวแทนทิศทางการเดินของลายนูนซึ่งเป็นไปตามสูตร

$$S_d = \sum \left| G(i_m, j_m) - G(i,j) \right| \qquad \text{สำหรับ } d = 1 \ldots N$$

M คือ จำนวนที่ถูกเลือกมาใช้ในการคำนวณในที่นี้กำหนดให้มีค่าเท่ากับ 8 เนื่องจากเราใช้บล็อก 8x8 pixel และ d คือทิศทางที่เราใช้ในการคำนวณ ส่วน N เป็นจำนวนทิศทางทั้งหมดที่ใช้กำหนดให้มีค่าเท่ากับ 8 (ทิศทาง d พิจารณาจากมุม $\prod(d-1)/N$ เรเดียน) เพราะว่าถ้าเรากำหนดให้ทิศทางมีน้อยเกินไปจุด core และ delta ที่ได้ในขั้นตอนสุดท้ายจะไม่ตรงกับจุดในภาพลายนิ้วมือจริง ส่วนการกำหนดให้มีทิศทางมาจนเกินไปก็จะทำให้ภาพทิศทางมีทิศทางที่ละเอียดจนเกินไป จุด core และ delta ที่ได้จะไม่ตรงกันกับจุด core และ delta ที่อยู่ในภาพลายนิ้วมือจริงอีกเช่นกัน

7.2 การตรวจหาจุดเฉพาะ

การตรวจหาจุดเฉพาะหรือการตรวจหาจุด core และ delta นั้นเนื่องจากภาพบล็อกทิศทางที่ได้จะมีลักษณะเหมือนการเดินของลายนูนของภาพลายนิ้วมือจริง เราจึงสามารถตรวจหาจุดเฉพาะ (จุด core และจุด delta) ได้จากภาพบล็อกทิศทางโดยอาศัยฮิสโตรแกรมของทิศทางที่อยู่ล้อมรอบกับจุดเฉพาะ ซึ่งเป็นส่วนประกอบที่ทำให้เกิดจุดเฉพาะขึ้นด้วยวิธีการตามขั้นตอนต่างๆต่อไปนี้

7.2.1 ขั้นตอนที่ 1 การหาจุดที่คาดว่าน่าจะเป็นจุด core และ delta

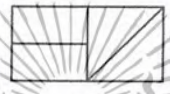
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขั้นตอนนี้เป็นการเลือกบล็อกทิศทางที่ใช้แทนจุด core คาดว่าน่าจะเป็นจุด core ในภาพทิศทางและเลือกบล็อกทิศทางที่ใช้แทนจุด delta คาดว่าน่าจะเป็นจุด delta ในภาพทิศทาง ซึ่งเราเลือกจุที่คาดว่าน่าจะเป็นจุดดังกล่าวทุกจุดในภาพทิศทางโดยใช้หลักการดังนี้

จุด core บล็อกทิศทางที่เลือกเป็นบล็อกทิศทางของจุด core นั้นจะต้องมีมุมที่มีความแตกต่างจากบล็อกทางซ้ายมากกว่าหรือเท่ากับ k1 ในที่นี้เรากำหนดให้ k1 มีค่าเท่ากับ 45 องศา เนื่องจากลักษณะของบล็อกทิศทางที่อยู่ล้อมรอบจุด core นั้นจะต้องมีบล็อกของส่วนประกอบที่อยู่ทางด้านขวามีลักษณะเอียงขวาและมีความแตกต่างของมุมกับบล็อกข้างเคียงทางด้านซ้ายเท่ากับ 45 องศาซึ่งเป็นไปตามสมการ

$$d\{B(i,j) - B(i-1,j-1)\} \geq k_1$$

ให้ B(i,j) เป็นบล็อกทิศทางที่แทนจุด core และคาดว่าน่าจะเป็นจุด core ในภาพทิศทาง ตัวอย่างของบล็อกทิศทางที่เลือกเป็นไปดังรูปที่ 7.1



B(i-1j) B(I,j)

รูปที่ 7.1 แสดงตัวอย่างของทิศทางที่เลือกเป็นบล็อกทิศทางของจุด core

จุด delta บล็อกทิศทางที่คาดว่าน่าจะเป็นบล็อกทิศทางของจุด delta และถูกเลือกเป็นจุด delta ในภาพทิศทาง เมื่อบล็อกทิศทางนั้นมีลักษณะดังเมตริกต่อไปนี้

$$\begin{vmatrix} Xs & B(i,j) & Xs \\ B(i-1,j) & Xs & B(i+1,j+1) \end{vmatrix}$$

จากเมตริกข้างต้นนี้บล็อกของทิศทางของจุดที่เป็นจุด delta นั้นต้องมีส่วนประกอบข้างเคียงหรือส่วนที่อยู่รอบข้างเหมือนกับเมตริกนั้น ให้ xs เป็นค่าอะไรก็ได้ ส่วนค่าของ B(i,j),B(i-1,j+1) และ B(i+1,j+1) ต้องเป็นไปตามเงื่อนไขต่อไปนี้

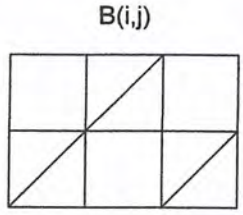
$$B(i-1,j+1) < B(i,j) \leq B(i+1,j+1)$$

$$B(i-1,j+1) < B(i,j) < B(i+1,j+1)$$

$$B(i-1,j+1) < k_2 \text{ และ } B(i+1,j+1) \geq k_2$$

ดังนั้น บล็อกทิศทางที่เราเลือก B(i,j) เป็นบล็อกทิศทางของจุดที่เป็นจุด delta ในภาพทิศทางจะต้องเป็นไปตามเงื่อนไขในข้างต้น และให้ k2 มีค่าเท่ากับ 90 องศา บล็อกทิศทาง xs เป็นค่าอะไรก็ได้ (don't care) ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.2 แสดงตัวอย่างของบล็อกทิศทางที่ถูกเลือกเป็นบล็อกทิศทางของจุด delta

จากวิธีการในข้างต้นเมื่อเราใช้วิธีการหาจุดที่คาดว่าน่าจะเป็นจุด core และ delta ในภาพทิศทางแล้ว จะได้จุดเป็นจำนวนมาก ตัวอย่างในรูปที่ ...และ... ซึ่งเป็นการนำภาพทิศทางมาทการคัดเลือกจุดที่คาดว่าน่าจะเป็นจุด core และ delta ในภาพทิศทาง

บล็อกสีขาวจะแทนจุดที่คาดว่าน่าจะเป็นจุด delta ส่วนบล็อกสีดำแทนจุดที่คาดว่าน่าจะเป็นจุด core และบล็อกสีเทาแทนจุดที่คาดว่าน่าจะเป็นทั้งจุด core และจุด delta ในภาพทิศทาง

7.2.2 ขั้นตอนที่ 2 การกำจัดจุดที่ไม่น่าจะใช่ทิ้ง

จากขั้นตอนที่ 1 เราได้ทำการเลือกบล็อกทิศทางที่คาดว่าน่าจะเป็นบล็อกทิศทางของจุด core และบล็อกทิศทางของจุด delta ในภาพทิศทาง มาถึงขั้นตอนนี้จะเป็นการลบหรือกำจัดบล็อกที่ไม่น่าจะเป็นจุดดังกล่าวออก เพราะบล็อกที่เราเลือกมาในขั้นตอนมีจำนวนมากมาย ดังนั้นเราจึงจำเป็นต้องกำจัดบล็อกที่ไม่น่าจะใช่บล็อกทิศทางของจุดเฉพาะเหล่านั้นออก เพื่อให้บล็อกทิศทางที่ใช้แทนจุด core และจุด delta ในภาพทิศทางมีน้อยลง เพื่อให้บล็อกทิศทางที่ใช้แทนจุด core และจุด delta ในภาพทิศทางมีน้อยลง เพื่อใช้ในการบล็อกทิศทางที่ใช้แทนจุดที่เป็นจุดเฉพาะในขั้นตอนต่อๆไป หลักในการลบจุดที่ไม่น่าจะใช่ออกเราใช้เงื่อนไขต่อไปนี้

จุด delta ที่ถูกเลือกมาในภาพทิศทาง ให้นำบล็อกทิศทางที่อยู่รอบๆบล็อกทิศทางที่เราเลือกให้เป็นจุด delta มาพิจารณาโดยการตัดบล็อกขนาด 7x7 ล้อมรอบบล็อกทิศทางของจุดที่เลือกเป็นจุด delta

จุด core จะมีการกระทำคล้ายๆกับจุด delta โดยการอาศัยบล็อกทิศทางข้างเคียงที่อยู่ล้อมรอบมาร่วมพิจารณา วิธีนี้เราจะทำการตัดบล็อกล้อมรอบบล็อกของจุดที่คาดว่าน่าจะเป็นจุด core ในขั้นตอนแรกด้วยบล็อกขนาด 5x5 แล้วพิจารณาว่าบล็อกทิศทางที่ล้อมรอบอยู่นั้นสนับสนุนให้บล็อกทิศทางที่เป็นจุด core ในภาพทิศทางนั้น ควรเป็นบล็อกทิศทางของจุด core ในภาพทิศทางหรือไม่ โดยใช้เงื่อนไขต่อไปนี้

$$MAX \quad f_1(H4(k)) < t_2$$

เมื่อ

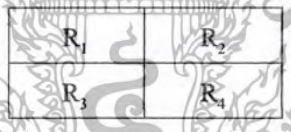
$$f_1(H(k)) = \sum H(j)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการข้างต้นฟังก์ชันของฮิตโตแกรมหรือความถี่ของทิศทางตั้งแต่ทิศทางที่ 0 ถึง 7 นั้น จะต้องมีค่าน้อยกว่า t_2 (ค่า threshold) โดยฟังก์ชันของความถี่ของทิศทางได้มาจากผลของความถี่ทิศทางข้างเคียงและทิศทางของฟังก์ชันนั้นเอง การ mod ด้วยค่า N นั้นวิธีการ mod ซึ่งเป็นลักษณะของการหารด้วยค่า N แล้วเอาเฉพาะเศษเท่านั้นมาเป็นผลลัพธ์โดยที่ผลลัพธ์นั้นจะมีค่าเท่ากับ 0 ถึง N-1 เท่านั้น ค่า threshold t_2 นั้นเป็นค่าคงที่ที่ได้มาจากการทดลอง เป็นตัวกำหนดว่าทิศทางที่อยู่ล้อมรอบจุด core ในภาพทิศทางควรมีทิศทางไปทางเดียวกันไม่เกินจำนวนเท่าใด การกำหนดให้มีค่าน้อยมีผลให้บล็อกทิศทางที่คาดว่าน่าจะเป็นบล็อกทิศทางของจุด core ในภาพทิศทางผ่านขั้นตอนนี้มีจำนวนน้อยลง และถ้ากำหนดให้มีค่ามากทำให้บล็อกทิศทางดังกล่าวผ่านขั้นตอนนี้ได้จำนวนมากขึ้น โดยในการทดลองจะกำหนดให้มีค่าเป็น 14-18 ขึ้นอยู่กับความชัดเจนของทิศทางหรือลักษณะทิศทางที่อยู่รอบๆจุดที่เราคาดว่าน่าจะเป็นจุด core ในภาพทิศทาง

7.2.3 ขั้นตอนที่ 3 การตรวจสอบส่วนโค้ง

ในขั้นตอนนี้เป็นการนำบล็อกทิศทางที่คาดว่าน่าจะเป็นจุด core ในภาพทิศทางที่ผ่านขั้นตอนการกำจัดจุดที่ไม่น่าจะใช่ทิ้งแล้ว ซึ่งจะได้บล็อกทิศทางของจุดที่คาดว่าน่าจะเป็นจุด core ในภาพทิศทางจำนวนหนึ่ง มาทำการตรวจสอบกับทิศทางที่อยู่ข้างเคียงว่าบล็อกทิศทางของจุดดังกล่าวนั้นอยู่ภายใต้ลักษณะส่วนโค้งที่มีลักษณะเป็นอย่างไรเป็นลักษณะโค้งขึ้นหรือโค้งลง วิธีการของขั้นตอนนี้จะทำการตัดบล็อกล้อมรอบๆจุด core ในภาพทิศทางและแบ่งบล็อกนี้เป็นย่าน 4 ย่าน ในแต่ละย่านคือ ย่าน R1,R2,R3 และ R4 จะสร้างเป็นบล็อกขนาด 5x5 เพื่อใช้ในการพิจารณา



รูปที่ 7.3 แสดงการแบ่งย่านของบล็อกที่ล้อมรอบจุด core ในภาพทิศทาง

จากรูปที่ 7.3 เมื่อทำการแบ่งบล็อกที่ตัดเป็น 4 ย่านดังรูป ซึ่งจะประกอบไปด้วยย่าน R1,R2,R3 และ R4 แล้ว นำความถี่ของทิศทาง H1,H2,H3 และ H4 ในย่าน R1,R2,R3 และ R4 ตามลำดับ มาพิจารณาลักษณะของส่วนโค้งในบล็อกที่เราตัดล้อมรอบจุด core โดยการพิจารณาการเอียงของทิศทางที่อยู่รอบๆบล็อกทิศทางที่คาดว่าน่าจะเป็นจุด core ในภาพทิศทางนั้น โดยใช้เงื่อนไขต่อไปนี้ในการพิจารณา

บล็อกทิศทางที่คาดว่าน่าจะเป็นจุด core จะอยู่ในลักษณะของส่วนโค้งขึ้น

ถ้า $f3+ (H1,H2) > f3- (H3,H4)$

และ $f3+ (H1,H2) > t3$

บล็อกทิศทางที่คาดว่าน่าจะเป็นจุด core จะอยู่ในลักษณะของส่วนโค้งลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า $f3+(H1,H2) < f3-(H3,H4)$

และ $f3+(H1,H2) < t3$

และฟังก์ชัน f3 กำหนดได้ดังนี้

$$f_{3+}(H_m,H_n) = \sum w_\alpha(j)H_m(j) + \sum w_\beta(j)H_m(j) \quad \text{เมื่อ } \alpha = 2, \beta = 6$$

$$f_{3-}(H_m,H_n) = \sum w_\alpha(j)H_m(j) + \sum w_\beta(j)H_m(j) \quad \text{เมื่อ } \alpha = 6, \beta = 2$$

เมื่อฟังก์ชัน f_{3+} เป็นฟังก์ชันของฮิสโตแกรมทิศทางที่อยู่ในลักษณะของส่วนโค้งขึ้น ซึ่งเราจะพิจารณาในย่าน R1 และ R2 ส่วนฟังก์ชัน f_{3-} เป็นฟังก์ชันของฮิสโตแกรมทิศทางที่อยู่ในลักษณะของส่วนโค้งลงใช้พิจารณาในย่าน R3 และ R4

ค่าของ $w_\alpha(j)$ กับ $w_\beta(j)$เป็นค่าคงที่ที่ได้กำหนดขึ้นมา โดยที่ค่าของ $w_\alpha(j)$เมื่อ $\alpha = j$ จะมีค่ามากกว่าค่า $w_\alpha(j)$เมื่อ$\alpha \neq j$ และค่า $w_\beta(j)$เมื่อ $=j$ จะมีค่ามากกว่า $w_\beta(j)$ เมื่อ $\neq j$ ค่าคงที่สองตัวนี้ใช้คูณเข้าไปกับความถี่ของทิศทางเพื่อเป็นการสร้างความแตกต่างระหว่างทิศทางที่เราพิจารณา (เป็นทิศทางที่ทำให้เกิดลักษณะโค้ง) กับทิศทางอื่นๆ และค่า threshold t_3 เป็นค่าที่ได้มาจากการทดลอง ใช้เพื่อพิจารณาฟังก์ชัน f_{3+}และฟังก์ชัน f_{3-} ว่าควรมีค่าเท่าใดจึงจะเป็นฟังก์ชันของลักษณะส่วนโค้งขึ้นและโค้งลง จากสมการข้างต้นถ้าบล็อกทิศทางที่คาดว่าน่าจะเป็นจุด core ในภาพทิศทาง จุดใดอยู่ในลักษณะของส่วนโค้งขึ้นก็จะมีค่าฟังก์ชัน f_{3+} ซึ่งอยู่ในย่าน R1 และ R2 มากกว่าฟังก์ชัน f_{3-} ซึ่งอยู่ในย่าน R3 และ R4 โดยที่ฟังก์ชัน f_{3+} ต้องมีค่ามากกว่าค่า t_3 ด้วยส่วนกรณีของลักษณะส่วนโค้งลง ฟังก์ชัน f_{3-} ซึ่งอยู่ในย่าน R3 และ R4 ก็ต้องมากกว่าฟังก์ชัน f_{3+} ในย่าน R1 และ R2 รวมทั้งฟังก์ชัน f_{3-} ต้องมีค่ามากกว่า t_3 ด้วย เช่นกัน ในการทดลองเราใช้ค่า $w_\alpha(j))$ และค่า $w_\beta(j))$ มีค่าเท่ากับ 2 คูณเข้าไปในทิศทางที่เราพิจารณาในแต่ละย่านเท่านั้น ที่เราใช้ค่าเพียงแค่ 2 เพราะต้องการให้เกิดความแตกต่างระหว่างทิศทางที่เราต้องการพิจารณากับทิศทางอื่นๆมากจนเกินไป ซึ่งจากที่เราทำการตัดบล็อกทิศทาง

ในแต่ละย่านให้มีขนาดใหญ่ก็ทำให้บล็อกทิศทางที่เราสร้างขึ้นสามารถครอบคลุมทิศทางที่ทำให้เกิดลักษณะส่วนโค้งที่เราต้องการแล้ว ค่า t_3 ใช้ค่าประมาณ 64 ถึง 70 ขึ้นอยู่กับลักษณะของทิศทางที่อยู่ล้อมรอบจุด core ในภาพทิศทางด้วย กรณีที่ทิศทางที่อยู่รอบข้างจุดที่เป็นจุด core ในภาพทิศทางไม่ชัดเจน จุด core ในภาพทิศทางไม่สามารถผ่านการตรวจสอบส่วนโค้งไปได้เลยโปรแกรมที่เราเขียนขึ้นมาจะใช้ค่าต่ำเองโดยอัตโนมัติ

7.2.4 ขั้นตอนที่ 4 การรวมกลุ่ม

ในขั้นตอนนี้เราจะนำบล็อกทิศทางที่ผ่านขั้นตอนที่แล้ว มาทำการจัดบล็อกทิศทางเหล่านั้นให้เป็นกลุ่มๆ ซึ่งการทดลองหลังจากที่ผ่านขั้นตอนของการตรวจสอบของลักษณะส่วนโค้งจุดที่คาดว่าน่าจะเป็นจุด core และจุดที่คาดว่าน่าจะเป็นจุด core และจุดที่คาดว่าน่าจะเป็นจุด delta ในภาพทิศทางจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วางตัวอยู่ในลักษณะเป็นกลุ่มๆ อยู่แล้วแต่การจัดกลุ่มของบล็อกทิศทางในขั้นตอนนี้ก็เพื่อเป็นการหาจุดศูนย์กลางและรวมกลุ่มเหล่านั้นซึ่งจะวางตัวอยู่ในลักษณะที่มีลักษณะรูปแบบเป็นบริเวณของ delta หรือบริเวณของ core และมักจะเกิดการเอียงขึ้น เราจึงจำเป็นต้องหาลักษณะของการเอียงของบริเวณที่มีลักษณะเป็น delta หรือ core ดังกล่าว หลักในการจัดรวมกลุ่มและการหาจุดศูนย์กลางกลุ่มจะใช้ทฤษฎีของ k-mean algorithm ใช้การพิจารณาการหาสมาชิกจากจุดใดๆ โดยกำหนดว่าจุดนั้นใกล้จุดศูนย์กลางจุดใดมากกว่ากัน ก็กำหนดให้จุดนั้นเป็นสมาชิกของจุดศูนย์กลางที่ใกล้ที่สุด แต่เงื่อนไขที่เพิ่มเข้าไปจะกำหนดด้วยว่าจุดที่จะเป็นสมาชิกของศูนย์กลางใดๆ ต้องมีระยะทางห่างจากจุดศูนย์กลางนั้นไม่เกินค่า t_1 ถ้าเกินค่า t_1 แล้วจุดนั้นไม่เป็นสมาชิกของศูนย์กลางใดเลย ก็จะกำหนดให้จุดนั้นเป็นจุดศูนย์กลางตัวใหม่ วิธีการทั้งหมดในขั้นตอนนี้คือ ในขั้นแรกเลือกบล็อกใดก็ได้เป็นจุดศูนย์กลางกลุ่มซึ่งกำหนดให้เป็นจุดศูนย์กลางเริ่มต้นจากนั้นหาสมาชิกโดยวัดระยะทางของจุด x ใดๆ ถึงจุดศูนย์กลางกลุ่มต้องน้อยกว่า t_1 และถ้ามากกว่าแล้วจุด x นั้นจะไม่เป็นสมาชิกของกลุ่มหลังจากที่ทำการหาสมาชิกของศูนย์กลางที่จุดเรียบร้อยแล้วถ้าจุดใดยังไม่เป็นสมาชิกของกลุ่มใดเลยจะให้จุดนั้นเป็นจุดศูนย์กลางเริ่มต้นอีกตัว จากนั้นทำการปรับปรุงจุดศูนย์กลางของกลุ่มโดยการนำสมาชิกทั้งหมดของกลุ่มมาใช้ในการคำนวณหาจุดศูนย์กลางใหม่ ขั้นตอนสุดท้ายถ้าจุดศูนย์กลางที่ได้มีตำแหน่งเปลี่ยนไปจากจุดเดิมก็ให้กลับไปทำการหาสมาชิกใหม่โดยใช้จุดศูนย์กลางของกลุ่มตำแหน่งใหม่มาหาสมาชิกใหม่ ซึ่งตอนนี้จะมีการเปลี่ยนแปลงของจำนวนสมาชิกทำเช่นนี้ไปเรื่อยๆจนได้จุดศูนย์กลางของกลุ่มที่คำนวณมาใหม่มีตำแหน่งเดียวกันกับตำแหน่งเดิมจึงเสร็จ จุดมุ่งหมายของขั้นตอนนี้นอกจากจะหาจุดศูนย์กลางกลุ่มแล้วยังต้องทำการจัดแบ่งจุด จุดที่มีทั้งหมดเป็นกลุ่มซึ่งส่วนใหญ่จะสามารถแบ่งได้โดยสะดวกเพราะจากการทดลองจุดที่มีทั้งหมดที่เข้ามาในขั้นตอนนี้จะเกาะกลุ่มกันเองอยู่แล้ว แล้วจึงมาใช้ในการคำนวณในขั้นตอนต่อไป ค่า t_1 กำหนดมีค่าเท่ากับ 8 เพราะจากการทดลองจุดที่เป็น core และ delta ในภาพทิศทางที่ผ่านจากขั้นตอนที่ 3 มานั้นจะมีลักษณะการเกาะกลุ่มรวมกันอยู่ค่า threshold ที่กำหนดสามารถครอบคลุมจุดทั้งหมดของกลุ่มได้

7.2.5 ขั้นตอนที่ 5 การหาแกนของพื้นที่

ในการหาแกนของพื้นที่ เป็นการหาลักษณะการเอนเอียงไปของบริเวณที่มีลักษณะเป็น core หรือ delta ซึ่งจากภาพลายนิ้วมือที่แปลงเป็นภาพทิศทางมักจะมีบริเวณที่มีลักษณะเป็น core หรือ delta มีการเอนเอียงไป แต่การหาแกนในขั้นตอนนี้ก็ยังรวมทิศทางปกติของบริเวณที่มีลักษณะเป็น core หรือ delta ที่ไม่มีการเอนเอียงเอาไว้ด้วยเพื่อความถูกต้องในกรณีที่บริเวณดังกล่าวไม่มีการเอนเอียงไป การหาแกนหลักของกลุ่มหลังจากที่เราได้ทำการจัดรวมกลุ่มและหาจุดศูนย์กลางของกลุ่มในขั้นตอนที่แล้ว วิธีการหาแกนหลักนี้แกนจะถูกหมุนไปตามแต่ละทิศทางต่างๆเป็นจำนวน 8 ทิศทาง (ใช้ทิศทางของการแปลงภาพลายนิ้วมือไปเป็นภาพทิศทางในขั้นตอนแรก) จากนั้นวิเคราะห์ส่วนประกอบข้างเคียงที่ใกล้จุดศูนย์กลางในย่านที่เรากำหนด ขั้นตอนคือ เราจะทำการตัดบล็อกรอบบล็อกทิศทางขนาด 5*7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล้อมรอบจุดศูนย์กลางของกลุ่มที่เป็นจุด core ที่มาจากขั้นตอนที่แล้วต่อมาหมุนแกน A_k ไปเป็นมุมเท่ากับ $\prod u/N$ เรเดียน นำทิศทางของบล็อกทิศทางข้างเคียงมาช่วยในการคำนวณ

การหมุนแกน A_k นั้น ได้มาจากการคำนวณโดยอาศัยความถี่ของทิศทาง H_μและ $\mu = 0 \ldots N-1$ ซึ่งเป็นความถี่ของทิศทางของบล็อกข้างเคียงที่ล้อมรอบจุดศูนย์กลางของกลุ่มที่อยู่ในวงปิดของบล็อกที่เราได้สร้างขึ้น และแกนที่หมุนไปตามค่าของ...นั้นเป็นค่าของฟังก์ชัน f_s ที่มีค่าสูงที่สุดกำหนดได้ดังนี้

$f_{s\alpha}(H_\alpha) > f_{s\mu}(H_\mu)$ ที่ทุกๆค่าของ $\alpha \neq \mu, 0 \leq \alpha, \mu \leq N-1$

เมื่อ

$f_{s\mu} = \sum w_\mu(j)H(i)$

ส่วนการจัดของจุด delta ก็จะทำการตัดบล็อกขนาด 7x7 ล้อมรอบจุด delta ซึ่งเป็นจุดศูนย์กลางของกลุ่ม การใช้บล็อกขนาด 7x7 เพื่อให้สามารถครอบคลุมบริเวณที่มีลักษณะเป็น delta ได้ทั้งหมด ซึ่งจากการทดลองพบว่ามีความเหมาะสมสามารถครอบคลุมบริเวณที่เป็น delta ได้ ส่วนการใช้บล็อกขนาดใหญ่หรือเล็กกว่านี้สามารถทำได้เช่นกัน แต่จะครอบคลุมบริเวณที่เป็น delta น้อยเกินไปหรือมากจนเกินไป ซึ่งทำให้ผลการวิเคราะห์การหมุนไปของแกนผิดพลาดได้ จากนั้นทำการหมุนแกนเช่นเดียวกับการหมุนแกนที่ล้อมรอบจุด core แล้วทำการแบ่งย่านเพื่อพิจารณา

7.2.6 ขั้นตอนที่ 6 การหาบล็อกทิศทางที่แทนจุด core และ delta

ในขั้นตอนสุดท้ายของการหาบล็อกทิศทางที่แทนจุด core และบล็อกทิศทางที่แทนจุด delta ในภาพทิศทาง โดยการนำบล็อกทิศทางที่แทนจุด core และ delta ที่อยู่ในแต่ละกลุ่มมาพิจารณาว่าบล็อกทิศทางใดมีความน่าจะเป็นสูงที่สุด ที่จะเป็นบล็อกทิศทางที่แทนจุด core และจุด delta ในภาพทิศทางได้ จุด delta บล็อกทิศทางที่มีความน่าจะเป็น P สูงที่สุดคือ บล็อกทิศทางของจุด delta ซึ่งเป็นฟังก์ชันของความถี่ของทิศทาง H1,H2,H3 และ H4 เป็นความถี่ของทิศทางของย่าน R1,R2,R3 และ R4 ตามลำดับ โดยเป็นย่านที่ถูกสร้างขึ้นเหมือนในขั้นตอนที่ 2 ดังรูปด้านล่างแล้วพิจารณาความน่าจะเป็น P ดังนี้

$$P = \sum f_{6\alpha}(H_j)$$

α คือค่าของการหมุนของแกนจากกลุ่มที่เราพิจารณาซึ่งได้มาจากในขั้นตอนที่แล้วและฟังก์ชัน $f_{6\alpha}$ ได้มาจาก

$$f_{6\alpha} = \sum w_\alpha(i)H(i)$$

ค่าของ $w_\alpha(i)$ เป็นค่าคงที่ที่เราได้กำหนดขึ้น ซึ่งจะเหมือนกับค่าคงที่ในตอนที่แล้วมาซึ่งมีค่าเท่ากับ 3 และ 2 ใช้ในการคูณเข้ากับทิศทางที่เราใช้พิจารณาซึ่งเป็นทิศทางตามแกนการหมุนและทิศทางข้างเคียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุด core บล็อกทิศทางของจุด core จะต้องมีค่าความน่าจะเป็น P สูงที่สุดและความน่าจะเป็น P เป็นฟังก์ชันของความถี่ของทิศทาง H1,H2,H3 และ H4 เป็นความถี่ของทิศทางของย่าน R1,R2,R3 และ R4 ตามลำดับ ย่านดังกล่าวนี้เราจะทำการแบ่งตามลักษณะรูปที่ได้แสดงไว้ด้านล่างซึ่งเป็นการหาลักษณะของโค้งขึ้นเท่านั้น เพราะเป็นไปตามนิยามของจุด core ดังนั้นความน่าจะเป็น P จะพิจารณาเฉพาะกับจุดที่เป็นโค้งขึ้นเท่านั้น เราจะใช้หลักการต่อไปนี้

$$P = \sum f_{\gamma\alpha}(H_j)$$

ฟังก์ชัน $f_{\gamma\alpha}$ กำหนดได้จาก

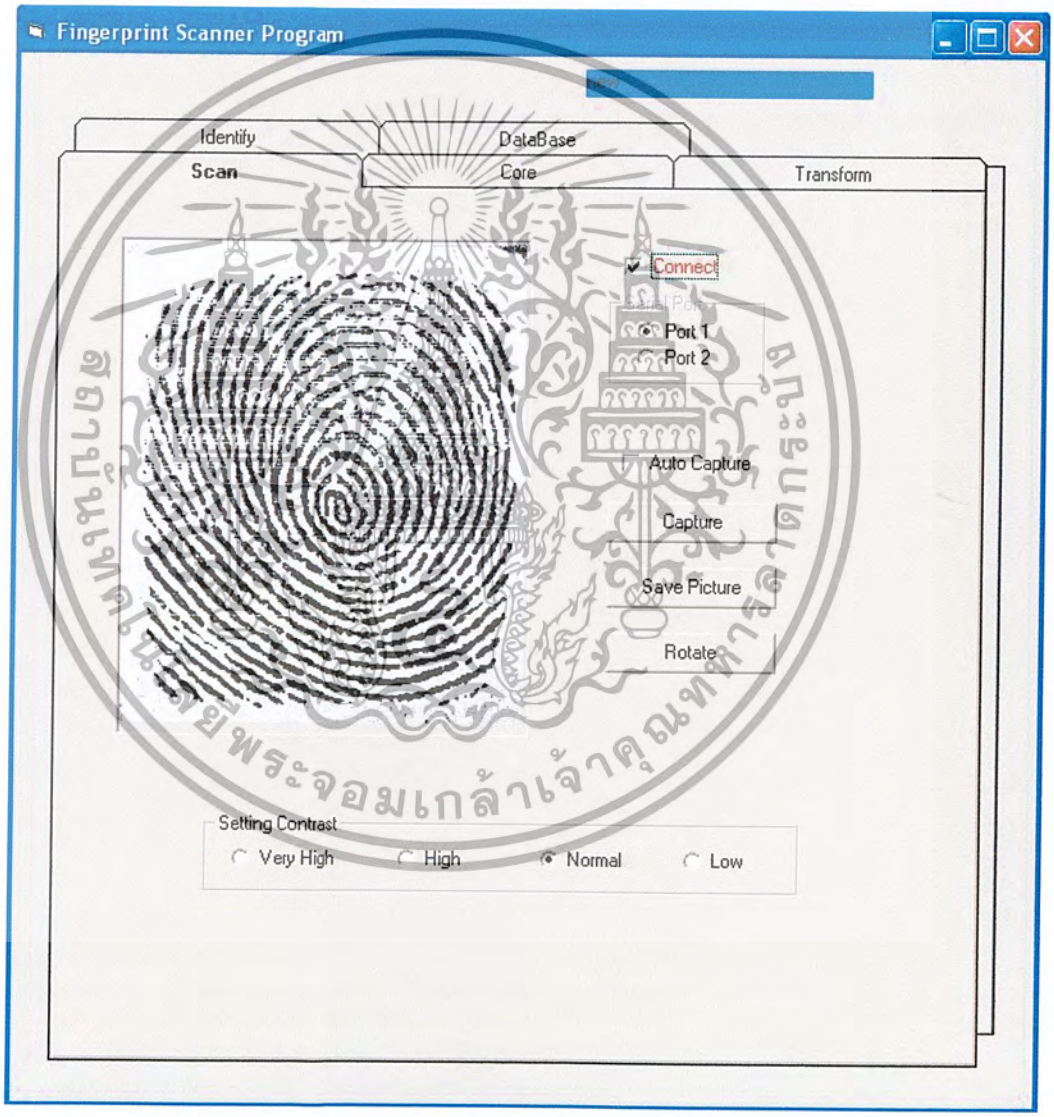
$$f_{\gamma\alpha} = \sum w_{\alpha}(i)H(i)$$

ในกรณีนี้เราไม่พิจารณาจุดที่เป็นโค้งลงเพราะจุดที่เป็นจุด core จากความหมายนั้นจะมีส่วนประกอบเป็นโค้งขึ้นเท่านั้น การใช้ค่า $w_{\alpha}(i)$จะพิจารณาจากมุมที่แกน A_k ถูกหมุนไปจากขั้นตอนที่ผ่านมาเช่นเดียวกันกับการใช้ในขั้นตอนของการหา delta โดยให้ค่า $w_{\alpha}(i)$ ที่ตรงกับมุมที่แกนหมุนไปมีค่ามากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
ส่วนควบคุมเซนเซอร์

การทำงานเริ่มจากโปรแกรม Fingerprint Scan สั่งให้ MCS-51 เริ่มการทำงานและ MCS-51 จะสั่งงานให้เซนเซอร์เริ่มทำการจับภาพส่งข้อมูลกับมาที่ MCS-51 จะนำข้อมูลจากแรมส่งผ่านพอร์ทอนุกรมกลับมายังโปรแกรม Fingerprint Scan และโปรแกรมจะทำการแปลงข้อมูลเป็นรูปภาพขนาด 256*300 พิกเซล



รูปที่ 8.1 ภาพแสดง window ของโปรแกรมควบคุมเครื่องสแกนลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

ผลการทดลอง

ผลการทดลองจะเป็นการจับภาพลายนิ้วมือจากเซนเซอร์ โดยจะแบ่งเป็น 4 ส่วน คือ

1. การปรับค่าต่างๆ ในรีจิสเตอร์เพื่อควบคุมความเข้ม และ แสงของรูปภาพ
2. การทำภาพให้บาง (Thining)
3. การหาจุดกลางของลายนิ้วมือ (Core)
4. การแปลงภาพเป็นภาพทิศทางและทำการเก็บข้อมูลใน Data Base

9.1 การปรับค่าต่าง ๆ ในรีจิสเตอร์

มีการปรับค่าควบคุมในรีจิสเตอร์ต่างๆ คือ DTR DCR และ PGC เพื่อควบคุมความเข้มและพื้นหลังของรูปภาพ โดยจะแบ่งเป็น 3 การทดลอง คือ

1. คงค่า DTR, DCR และเปลี่ยนค่า PGC

เมื่อให้ DTR= 3AH DCR= 02H PGC= 04H

เมื่อให้ DTR= 3AH DCR= 02H PGC= 06H

เมื่อให้ DTR= 3AH DCR= 02H PGC= 07H

รูปที่ 9.1 แสดงลายนิ้วมือเมื่อคงค่า DTR, DCR และเปลี่ยนค่า PGC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่าค่า PGC มีผลต่อความเข้มของเส้นลายนิ้วมือแต่ถ้าปรับให้มีค่าสูง รูปที่ได้ลายเส้นจะเข้ม แต่ลายจะเป็นริ้วๆ ไม่ชัด

2. คงค่า PGC, DTR และเปลี่ยนค่า DCR

เมื่อให้ DTR= 30H DCR= 01H PGC= 06H

เมื่อให้ DTR= 30H DCR= 02H PGC= 06H

เมื่อให้ DTR= 30H DCR= 03H PGC= 06H

รูปที่ 9.2 แสดงลายนิ้วมือเมื่อคงค่า PGC, DTR และเปลี่ยนค่า DCR

จะเห็นว่าการเพิ่มค่า DCR จะทำให้ภาพที่ได้จางลงเนื่องจาก การเพิ่มค่า DCR คือการเพิ่มกระแส Discharge ทำให้ประจุที่ถูกเก็บในเซลล์ถูกคายออกไปมากจึงทำให้ประจะที่เหลือในเซลล์มีค่าน้อยเมื่อผ่านการแปลง A/D จึงทำให้ภาพที่ออกมามีความเข้มน้อยลงเมื่อเพิ่มค่า DCR โดยที่ค่า DCR จะมีผลต่อความเข้มของทั้งภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คงค่า DCR,PGC และเปลี่ยนค่า DTR

เมื่อให้ DTR= 10H DCR= 03H PGC= 07H

เมื่อให้ DTR= 20H DCR= 03H PGC= 07H

เมื่อให้ DTR= 30H DCR= 03H PGC= 07H

รูปที่ 9.3 แสดงลายนิ้วมือเมื่อคงค่า DCR, PGC และเปลี่ยนค่า DTR

จากรูปจะเห็นว่าการเพิ่มค่า DTR ภาพที่ได้จะมีความเข้มลดลง เนื่องจากการเพิ่มค่า DTR คือการเพิ่มเวลาในการ Discharge ประจุทำให้เมื่อเพิ่มมากเวลาในการคายประจุจะยิ่งมากตามทำให้มีประจุคงเหลือในเซลล์น้อย เมื่อผ่านการแปลง A/D จะทำให้ภาพมีความเข้มน้อยลงเมื่อเพิ่มค่า DTR โดยที่ค่า DTR จะมีผลต่อความเข้มของฉากหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2 การทำภาพให้บาง (Thinning)

คือการทำให้ภาพบางลงเป็นเส้นสามารถแสดงโฟลชาร์ทการทำงานได้ดังนี้

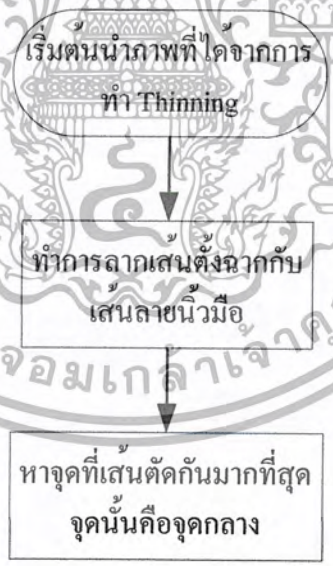
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Before After

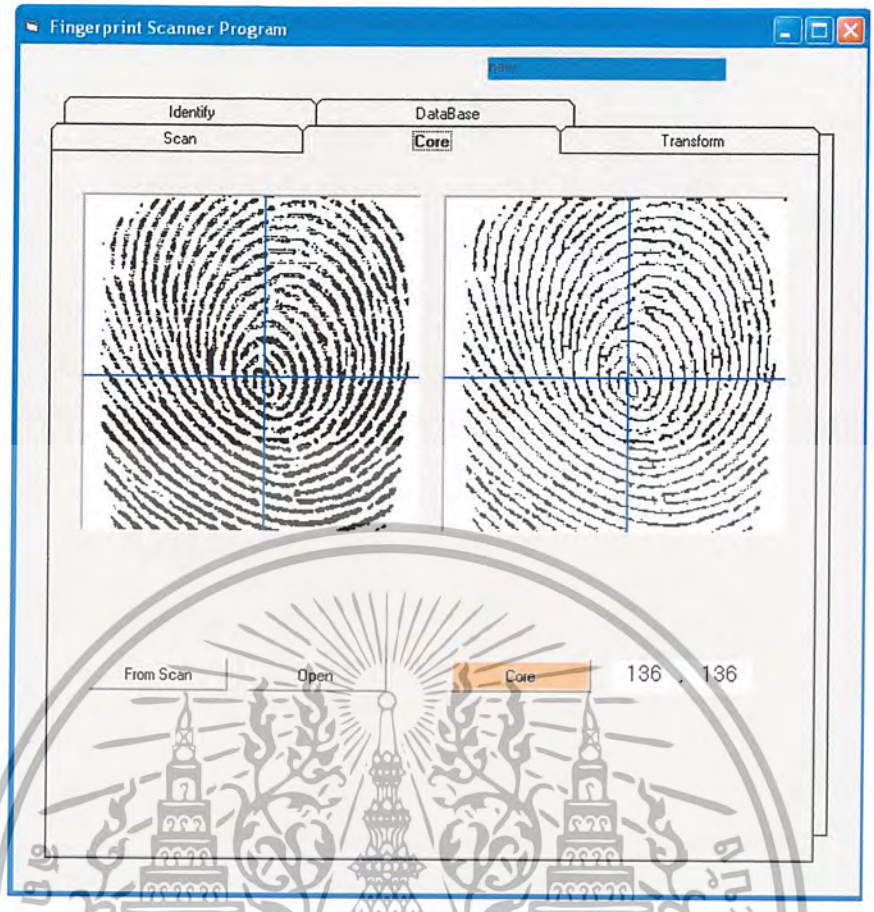
รูปที่ 9.4 แสดงรูปภาพก่อนการทำและหลังทำ Thinning

9.3 การหาจุดกลางของลายนิ้วมือ (Core)

แสดงหลักการทำงานของโปรแกรมดังนี้



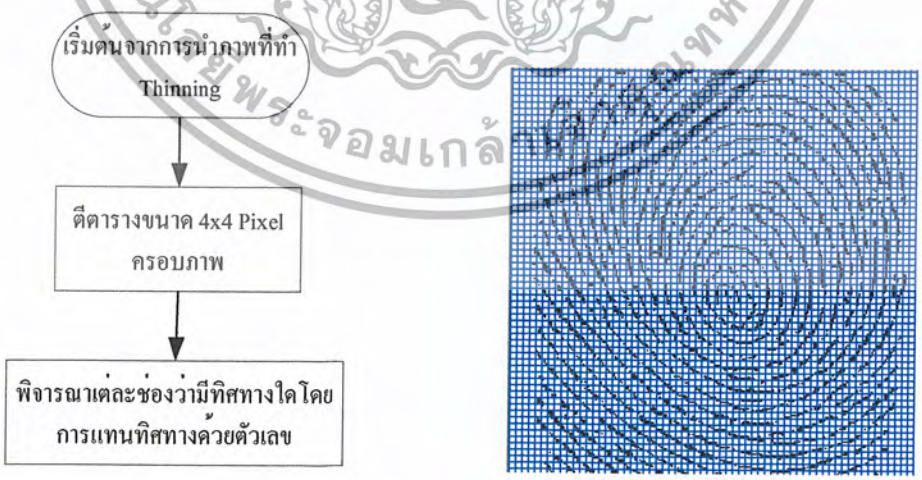
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9.5 แสดงจุดกลางของลายนิ้วมือ (Core)

9.4 การแปลงภาพเป็นภาพทิศทางและทำการเก็บข้อมูลใน Data Base

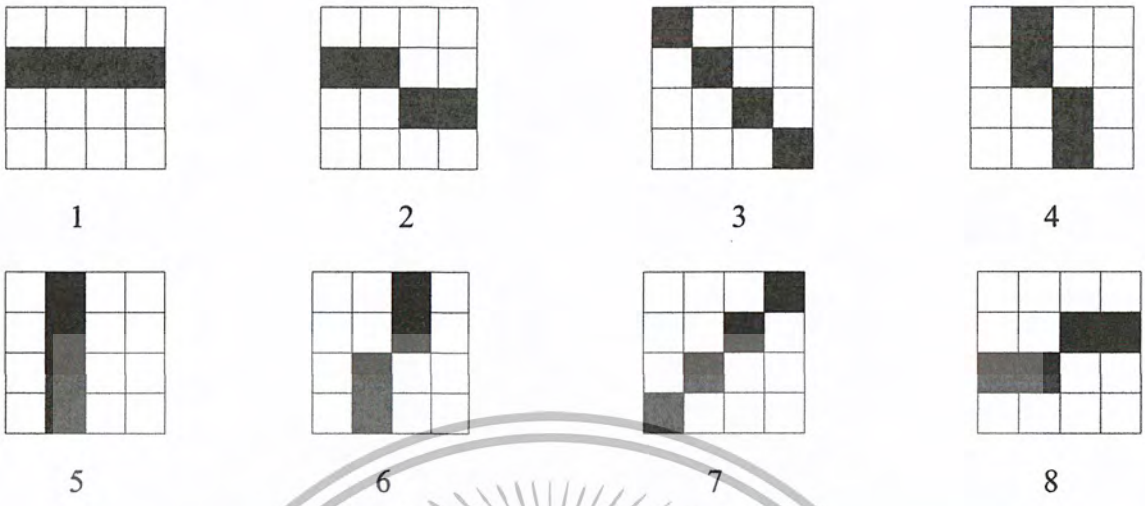
9.4.1 แสดงหลักการทำงานของโปรแกรมในการแปลงภาพเป็นภาพทิศทางดังนี้



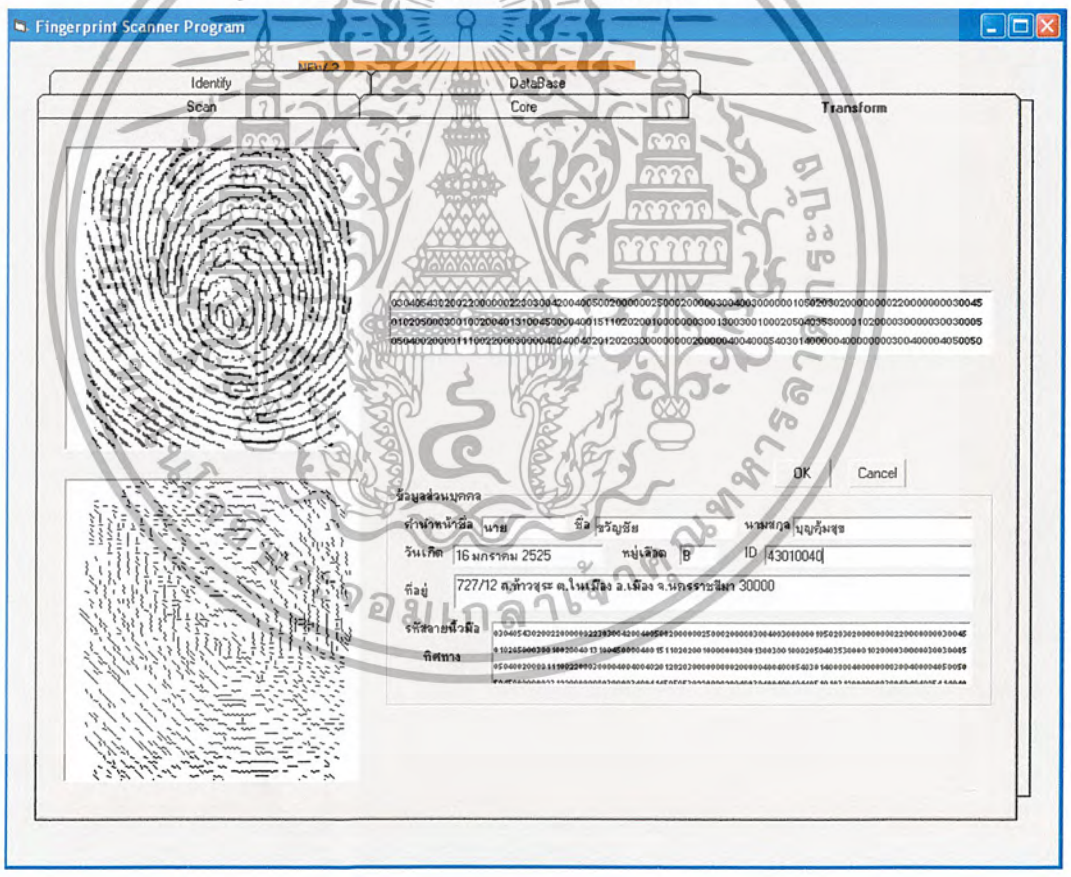
รูปที่ 9.6 แสดงขั้นตอนการแปลงภาพเป็นภาพทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทำการแบ่งทิศทางทั้งหมดจะได้ทิศทาง 16 ทิศทาง และจะได้ 8 แนว ทำการแทนทิศทางด้วยตัวเลข



รูปที่ 9.7 แสดงการแทนภาพทิศทางเป็นตัวเลข

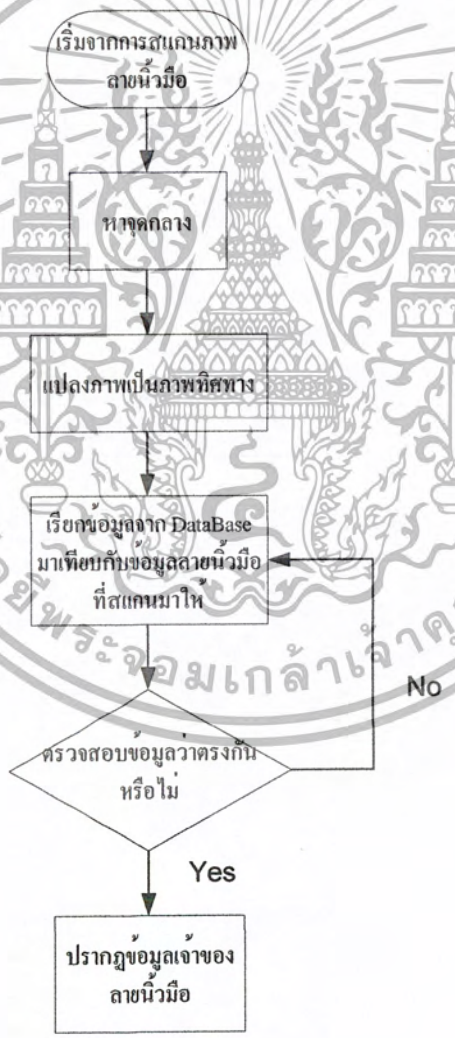


รูปที่ 9.8 แสดงการแปลงภาพเป็นภาพทิศทางและการเก็บข้อมูลลงใน Data Base

9.4.2 แสดงโฟลชาร์ทการเก็บข้อมูลใน Data Base ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.5 ระบบการตรวจสอบลายนิ้วมือสามารถแสดงดังโฟลชาร์ทได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 10
สรุปและวิจารณ์ผลการทดลอง

จากการทดลองตามบทที่ผ่านข้างต้น ในส่วนการทดลองโหมดการจับภาพทั้ง 2 แบบสามารถทำได้โดยไม่มีความผิดพลาดเกิดขึ้น ส่วนของการเปลี่ยนค่ารีจิสเตอร์ควบคุมจะเห็นว่ารีจิสเตอร์ DTR,DCR เมื่อเพิ่มค่าจะทำให้รูปที่ได้มีความเข้มน้อยลง โดย DTR จะมีผลต่อความเข้มของพื้นหลัง DCR จะมีผลต่อความเข้มของทั้งรูป ส่วนรีจิสเตอร์ PGC จะทำหน้าที่เพิ่มความเข้มของเส้นลายนิ้วมือ

ในการทดลองจะพบว่าเกิดปัญหาคือ ภาพลายนิ้วมือที่ได้จากเครื่องสแกนลายนิ้วมือมีความผิดพลาดในบางจุดพิกเซล เนื่องจากการเลื่อนของข้อมูลที่ส่งมาจากตัวเซ็นเซอร์ เพราะว่าระบบของตัวเซ็นเซอร์มีการเพิ่มตำแหน่งอัตโนมัติเมื่อขา RD เปลี่ยนจาก Low เป็น High ในบางครั้งการสั่งอ่านข้อมูลจะเกิดความผิดพลาดเนื่องจากความถี่ในการเปลี่ยนระดับแรงดันที่ขา RD มีค่าสูงการส่งแรงดันผ่านสายที่ยาวทำให้มีการกระเพื่อมขณะเปลี่ยนแรงดัน (ปกติตัวชิปถูกออกแบบมาเพื่อใช่ในอุปกรณ์ขนาดเล็ก เช่น PDA) ทำให้เกิดการผิดพลาดขึ้นทำให้บางข้อมูลมีการเลื่อนกันให้ภาพที่ได้ไม่สมบูรณ์ในบางครั้ง

อีกปัญหาหนึ่งคือ เมื่อเพิ่มค่า PGC มากขึ้นจะทำให้ตัวเซนเซอร์รับสัญญาณกระเพื่อมได้ไวขึ้นมากมีผลทำให้รูปภาพที่ได้มีการเลื่อนมากขึ้น

การแก้ปัญหาคือ ต่อตัวเก็บประจุ เพื่อลดการกระเพื่อมของการเปลี่ยนระดับแรงดันเพื่อให้ได้ผลที่ถูกต้องมากขึ้น ซึ่งผลที่ได้ก็ทำให้การเลื่อนของข้อมูลลดลงไปมากแต่ก็ยังไม่สามารถลดจนหมดได้จึงทำให้มีการเลื่อนของข้อมูลอยู่บ้างเล็กน้อย

เนื่องจากถ้าสามารถได้รูปลายนิ้วมือที่ถูกต้องสมบูรณ์เราสามารถนำไปใช้ประโยชน์ในการจดจำบุคคลได้อย่างแม่นยำ ซึ่งมีประโยชน์ในการอำนวยความสะดวกได้หลายประการ ถ้าลายนิ้วมือมีความถูกต้องสมบูรณ์จะทำให้การจดจำบุคคลทำได้อย่างแม่นยำและมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; *******************************************************
; Program       : Fingerprint Scanner
; Description   : Receiving and Sending Fingerprint Data
                  between Fingerprint Scanner with Computer
                  by Passing the Serial Port and Using Value Cystal
                  Frequency 22.1184 MHz
; *******************************************************

DATA_PORT  EQU   P1
DRIVE_LE   BIT   P3.2
WAIT_PIN   BIT   P3.3
INTR_PIN   BIT   P3.4
ADD_RAM    BIT   P3.5

CS1_PIN    BIT   P0.0
A0         BIT   P0.1
WR_PIN     BIT   P0.2
RD_PIN     BIT   P0.3

           ORG   0000H

           MOV   DATA_PORT,#00H
           SETB  WAIT_PIN
           SETB  INTR_PIN
           CLR   ADD_RAM

; ******** SET SERIAL PORT ***************

           ORL   PCON,#10000000B
           MOV   SCON,#01010000B
           MOV   TMOD,#00100000B
           MOV   TH1,#0FFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV   TL1,#0FFH

                SETB  TR1

; ****** WAIT COMMAND FORM COM *******

                CLR   RI
MAIN:           JNB   RI,$
                MOV   A,SBUF
                CLR   RI

                CJNE  A,#01H,MAIN

; ******** START SETTING MBF200 *********

START:          MOV   P0,#00001110B
                SETB  DRIVE_LE
                CLR   DRIVE_LE

                MOV   R0,#08H                   ; เซตค่า CTRLA
                MOV   A,#00H

                LCALL SENSOR_EN

                MOV   R0,#09H                   ; เซตค่า CTRLB
                MOV   A,#00000101B

                LCALL SENSOR_EN

                MOV   R0,#06H                   ; เซตค่า DTR
                MOV   A,#20H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL SENSOR_EN

                MOV   R0,#07H                    ; เซตค่า DCR
                MOV   A,#02H

                LCALL SENSOR_EN

                MOV   R0,#0CH                    ; เซตค่า PGC
                MOV   A,R1

                LCALL SENSOR_EN

                MOV   R7,25
                DJNZ  R7,$

                AJMP  BEGIN

; ******* READ FINGERPRINT DATA FROM MBF200 TO RAM **********

BEGIN:          MOV   R1,#20
                MOV   R2,#0FFH
                MOV   R3,#97H
                MOV   R4,#97H
CAPTURE1:
                CLR   ADD_RAM
                MOV   DPTR,#0000H

                MOV   R5,#00H                    ; เซตค่ารีจิสเตอร์ที่ช่วยในการ
Row Capture
                MOV   R6,#00H

C_ROW1:         DJNZ  R4,W_ROW1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SJMP    CAPTURE2

W_ROW1:         MOV     R0,#00H                 ;เซตค่า RAH
                MOV     A,R5

                LCALL SENSOR_EN

                MOV     R0,#01H                 ; เซตค่า RAL
                MOV     A,R6

                LCALL SENSOR_EN

                MOV     R0,#08H                 ; เซตคำสั่ง Row Capture
                MOV     A,#00000001B

                LCALL SENSOR_EN

                DJNZ    R1,$
                MOV     R1,#20

                MOV     DATA_PORT,#0FFH

                MOV     P0,#00000111B           ;เริ่มทำการ Read Data 256
ครั้งแรก

                SETB    DRIVE_LE
                CLR     DRIVE_LE

                MOV     A,DATA_PORT
                MOV     P0,#00001110B
                SETB    DRIVE_LE
                CLR     DRIVE_LE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOVX @DPTR,A
                INC     DPTR

LOOK1:          DJNZ    R2,S_ROW1
                MOV     R2,#0FFH

                INC     R6

                LJMP C_ROW1

S_ROW1:         MOV     P0,#00000111B
                SETB    DRIVE_LE
                CLR     DRIVE_LE

                MOV     A,DATA_PORT
                MOV     P0,#00001110B
                SETB    DRIVE_LE
                CLR     DRIVE_LE

                MOVX @DPTR,A
                INC     DPTR

                AJMP LOOK1


CAPTURE2:

                SETB    ADD_RAM
                MOV     DPTR,#0000H

C_ROW2:         DJNZ    R3,W_ROW2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LJMP    EXIT

W_ROW2:         CJNE    R6,#00H,NEXT

                ACALL ADDROW


NEXT:           MOV     R0,#00H                         ; เซตค่า RAH
                MOV     A,R5

                LCALL SENSOR_EN

                MOV     R0,#01H                         ; เซตค่า RAL
                MOV     A,R6

                LCALL SENSOR_EN

                MOV     R0,#08H                         ; เซตคำสั่ง Row Capture
                MOV     A,#00000001B

                LCALL       SENSOR_EN

                DJNZ    R1,$
                MOV     R1,#20

                MOV     DATA_PORT,#0FFH

                MOV     P0,#00000111B                   ; เริ่มทำการ Read Data 256   ครั้งหลัง

                SETB    DRIVE_LE
                CLR     DRIVE_LE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV   A,DATA_PORT
                MOV   P0,#00001110B
                SETB  DRIVE_LE
                CLR   DRIVE_LE

                MOVX @DPTR,A
                INC   DPTR

LOOK2:          DJNZ  R2,S_ROW2
                MOV   R2,#0FFH

                INC   R6

                LJMP C_ROW2

S_ROW2:         MOV   P0,#00000111B
                SETB  DRIVE_LE
                CLR   DRIVE_LE

                MOV   A,DATA_PORT
                MOV   P0,#00001110B
                SETB  DRIVE_LE
                CLR   DRIVE_LE

                MOVX @DPTR,A
                INC   DPTR

                AJMP LOOK2

EXIT:           MOV   R0,#09H                    ; หยุดการทำงาน MBF200
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV    A,#00H

                ACALL SENSOR_EN


; *******SEND DATA FORM RAM TO COM********

                MOV    R2,#0FFH
                MOV    R3,#97H
                MOV    R4,#97H

PAGE1:          CLR    ADD_RAM
                MOV    DPTR,#0000H

RTC1:           JNB    RI,$
                CLR    RI

                DJNZ   R4,SEND1
                SJMP   PAGE2

SEND1:          MOVX A,@DPTR
                INC    DPTR

                ACALL SEND_DATA

FOR1:           DJNZ   R2,S_RTC1
                MOV    R2,#0FFH

                AJMP RTC1

S_RTC1:         MOVX A,@DPTR
                INC    DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL SEND_DATA

                AJMP FOR1


PAGE2:          SETB   ADD_RAM
                MOV    DPTR,#0000H

                SJMP   SKIP

RTC2:           JNB    RI,$
                CLR    RI

SKIP:           DJNZ   R3,SEND2
                SJMP EXITJOB

SEND2:          MOVX A,@DPTR
                INC    DPTR

                ACALL SEND_DATA

FOR2:           DJNZ   R2,S_RTC2
                MOV    R2,#0FFH

                AJMP RTC2

S_RTC2:         MOVX A,@DPTR
                INC    DPTR

                ACALL SEND_DATA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                AJMP FOR2

EXITJOB:        AJMP MAIN

; ************ PROCEDURE ***************

ADDROW:         MOV   R5,#01H
                MOV   R6,#00H
                RET

SENSOR_EN:      MOV   DATA_PORT,R0

                MOV   P0,#00001001B
                SETB  DRIVE_LE
                CLR   DRIVE_LE

                MOV   P0,#00001101B
                SETB  DRIVE_LE
                CLR   DRIVE_LE

                MOV   DATA_PORT,A

                MOV   P0,#00001011B
                SETB  DRIVE_LE
                CLR   DRIVE_LE

                MOV   P0,#00001110B
                SETB  DRIVE_LE
                CLR   DRIVE_LE
                RET

SEND_DATA:      CLR   TI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   SBUF,A
        JNB   TI,$
        RET

        END

;*************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit

Dim Inform1 As Variant

Dim Row As Integer
Dim Col As Integer
Dim Data(0 To 255, 0 To 299) As Integer

Dim i As Integer ' ตัวแปรวน loop
Dim j As Integer ' ตัวแปรวน loop
Dim a As Integer ' ตัวแปรวน loop พิเศษ
Dim b As Integer ' ตัวแปรวน loop พิเศษ
Dim aa As Integer
Dim bb As Integer
Dim c As Integer
Dim cc As Integer
Dim d As Integer
Dim dd As Integer
Dim ddd As Integer
Dim e As Integer
Dim f As Integer
Dim R As Integer
Dim Round As Integer

Dim startX As Integer
Dim endX As Integer
Dim startY As Integer
Dim endY As Integer
Dim X1 As Integer
Dim X2 As Integer
Dim Y1 As Integer
Dim Y2 As Integer

Dim Threshold As Byte
Dim Colorcode As Long
Dim Grayscale(0 To 255, 0 To 299) As Integer
Dim Red As Integer
Dim Green As Integer
Dim Blue As Integer
Dim RefreshCore As Boolean

Dim ArrayBuff0(0 To 255, 0 To 299) As Byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim ArrayBuff1(0 To 255, 0 To 299) As Byte
Dim Apicture0(0 To 255, 0 To 299) As Byte
Dim Apicture1(0 To 255, 0 To 299) As Byte
Dim Apicture00(0 To 255, 0 To 299) As Integer
Dim Apicture11(0 To 255, 0 To 299) As Integer
Dim Apicture2(0 To 255, 0 To 299) As Byte

Dim RealLine As Boolean
Dim Bcompletethin As Boolean ' boolean ตรวจสอบว่าการทำบางสมบูรณ์แล้วหรือไม่
Dim Berase As Boolean ' boolean บ่งบอกว่า pixel นั้นจะต้องทำการลบหรือไม่
Dim xi As Integer ' พิกัด x ที่มีคะแนนสุงสุด
Dim xj As Integer ' พิกัด y ที่มีคะแนนสุงสุด
Dim Ascore(0 To 255, 0 To 255) As Long ' array เก็บคะแนนที่สะสมจากการมีเส้น
ลากผ่านซ้ำ
Dim maxscore As Long ' คะแนนสูงสุด
Dim TransferarrayPicture As Boolean
Dim boolOutBlack As Boolean

Dim GetFirst As Boolean
Dim Abox(1 To 32, 1 To 32) As Byte
Dim Dbox(-7 To 7, -7 To 7) As Byte
Dim Xbox(-7 To 7, -7 To 7) As Byte
Dim TheBox(0 To 8, 1 To 8, 1 To 8) As Byte
Dim g As Integer
Dim h As Integer
Dim m As Integer
Dim n As Integer
Dim k As Single
Dim s As String
Dim Pointnum As Byte
Dim PointnumOK As Byte
Dim RefreshTransform As Boolean
Dim stDataFinger As String
Dim stDataBase As String
Dim boxX As Integer
Dim boxY As Integer
Dim stBuff As String
Dim stBuff2 As String
Dim stBuffer As String
Dim stBuffer2 As String
Dim IdentifyOK As Boolean
Dim AnstDataFinger(1 To 225) As Byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim AnstDataBase(1 To 225) As Byte
Dim Correctscore As Integer
Dim Correctpercent As Single
Dim PercentOK As Single

Dim AScanTabScanpic(0 To 255, 0 To 299) As Byte
Dim AIdentifyTabFromCorepic(0 To 255, 0 To 299) As Byte
Dim AIdentifyTabDTBpic(0 To 255, 0 To 299) As Byte
Dim ACoreTabScanOpenpic(0 To 255, 0 To 255) As Byte
Dim ACoreTabThinpic(0 To 255, 0 To 255) As Byte
Dim ATransformTabFromCorepic(0 To 255, 0 To 255) As Byte
Dim ATransformTabDirectpic(0 To 255, 0 To 255) As Byte
Dim ADTBtabDTBpic(0 To 255, 0 To 255) As Byte
Dim ADTBtabThinpic(0 To 255, 0 To 255) As Byte

Dim Iend As Integer
Dim Jend As Integer
Dim Picmode As Byte
Dim Picnow As String
Dim Picsource As String
Dim Picdestination As String
Public Sub pPBerase1_2() ' สำหรับพิจารณาลบ pixel ที่อยู่เดี่ยวๆ หรืออยู่เป็นคู่
        For j = 1 To 254
            For i = 1 To 254
                If Apicture0(i, j) = 1 Then
                c = Apicture0(i - 1, j - 1) + Apicture0(i - 1, j) + _
                    Apicture0(i - 1, j + 1) + Apicture0(i, j - 1) + _
                    Apicture0(i, j + 1) + Apicture0(i + 1, j - 1) + _
                    Apicture0(i + 1, j) + Apicture0(i + 1, j + 1)
                    If c = 0 Then
                        Apicture0(i, j) = 0
                    End If
                    If c = 1 Then
                        pPBerase2
                    End If
                End If
            Next i
        Next j
End Sub
Public Sub pPBerase2() ' สำหรับลบ pixel ที่อยู่เป็นคู่
        For a = -1 To 1
            For b = -1 To 1
        If (Apicture0(i + a, j + b) = 1) And ((a <> 0) And (b <> 0)) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            cc = Apicture0(i - 1 + a, j - 1 + b) + Apicture0(i - 1 + a, j + b) + _
                 Apicture0(i - 1 + a, j + 1 + b) + Apicture0(i + a, j - 1 + b) + _
                 Apicture0(i + a, j + 1 + b) + Apicture0(i + 1 + a, j - 1 + b) + _
                 Apicture0(i + 1 + a, j + b) + Apicture0(i + 1 + a, j + 1 + b)
            If cc = 1 Then
                 Apicture0(i, j) = 0
                 Apicture0(i + a, j + b) = 0
                          End If
                      End If
                  Next b
             Next a
End Sub
Public Sub pPBclearscore() ' สำหรับเคลียร์คะแนนที่ใช้นับเพื่อหาจุดกลาง
             For i = 0 To 255
                  For j = 0 To 255
                       Ascore(i, j) = 0
                  Next j
             Next i
End Sub
Public Sub pPBa2adetectcore() ' สำหรับถ่ายโอนจาก arrayภาพ มาให้ arrayที่ใช้คำนวณ
การทำ Thining
             For j = 0 To 255
                For i = 0 To 255
                ' เตรียม array ก่อนทำบาง
                   Apicture0(i, j) = ACoreTabScanOpenpic(i, j)
                   Apicture1(i, j) = Apicture0(i, j)
                   Apicture00(i, j) = 0
                   Apicture11(i, j) = 0
                Next i
           Next j
End Sub
Public Sub pPBReverseApic1_Apic0() ' สำหรับคืนค่า array ที่ใช้ระหว่างการทำ
Thining เพื่อวน loop ซ้ำต่อไป
             For i = 0 To 255
                       For j = 0 To 255
                            Apicture0(i, j) = Apicture1(i, j)
                       Next j
                  Next i
End Sub
Public Sub pPBReverseApic11_Apic00() ' สำหรับคืนค่า array ที่ใช้ระหว่างการทำ
Thining เพื่อวน loop ซ้ำต่อไป
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            For i = 0 To 255
                For j = 0 To 255
                    Apicture00(i, j) = Apicture11(i, j)
                Next j
            Next i
End Sub

Public Sub pPBfating() ' สำหรับทำเส้นให้หนา
    For R = 1 To Round                          ' พิจารณา pixel ที่เป็นสีดำ
        For i = startX + 1 To endX - 1
            For j = startY + 1 To endY - 1
                If Apicture0(i, j) = 1 Then ' ทำจุดรอบข้างให้เป็นสีทั้งหมด
                    For a = -1 To 1
                        For b = -1 To 1
                            Apicture1(i + a, j + b) = 1
                        Next b
                    Next a
                End If
            Next j
        Next i
        ' คืนค่า Apicture1 > Apicture0
        pPBReverseApic1_Apic0
    Next R
End Sub
Public Sub pPBprethiningV3() ' สำหรับเตรียมข้อมูลก่อนการทำ Thining
    For j = startY + 2 To endY - 2
        For i = startX + 2 To endX - 2
            'A0
            ' pixขาว = 0
            ' pixดำ = 1
            'A2
            ' ขาว = 0
            ' ดำใน = 1
            ' ดำนอก = 2
            ' เส้นจริง = 20
            ' ==================

            If Apicture0(i, j) = 0 Then
                Apicture00(i, j) = 0 ' ขาว
            Else
            c = Apicture0(i - 1, j - 1) + Apicture0(i - 1, j) + _
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Apicture0(i - 1, j + 1) + Apicture0(i, j - 1) + _
                    Apicture0(i, j + 1) + Apicture0(i + 1, j - 1) + _
                    Apicture0(i + 1, j) + Apicture0(i + 1, j + 1)
        If c = 8 Then
            Apicture00(i, j) = 1 ' ดำใน
                    Else ' pixelวงนอก
              boolOutBlack = False
            For a = -1 To 1
            For b = -1 To 1
cc = Apicture0(i + a - 1, j + b - 1) + Apicture0(i + a - 1, j + b) + _
     Apicture0(i + a - 1, j + b + 1) + Apicture0(i + a, j + b - 1) + _
     Apicture0(i + a, j + b + 1) + Apicture0(i + a + 1, j + b - 1) + _
     Apicture0(i + a + 1, j + b) + Apicture0(i + a + 1, j + b + 1)
          If cc = 8 Then
     boolOutBlack = boolOutBlack Or True
          Else
               boolOutBlack = boolOutBlack Or False
                    End If
                  Next b
                  Next a
          If boolOutBlack = True Then
          Apicture00(i, j) = 10 ' ดำนอก
          Else
               Apicture00(i, j) = 100 ' เส้นจริง
          End If
          End If
          End If

       Next i
    Next j

        For j = 0 To 255
            Apicture00(0, j) = 0
            Apicture00(1, j) = 0
            Apicture00(254, j) = 0
            Apicture00(255, j) = 0
            Apicture00(j, 0) = 0
            Apicture00(j, 1) = 0
            Apicture00(j, 254) = 0
            Apicture00(j, 255) = 0
        Next j
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Sub pPBthiningV3() ' สำหรับการทำ Thining
  For R = 1 To Round
     For j = startY + 2 To endY - 2
        For i = startX + 2 To endX - 2
           If (Apicture00(i, j) = 0) Or (Apicture00(i, j) = 100) Then
                          ' ขาว เส้นจริง ไม่ต้องยุ่ง
                    Else
                          ' เป็นดำใน หรือ ดำนอก อย่างใดอย่างหนึ่ง
                          ' พิสูจน์ดำใน
             c = Apicture0(i - 1, j - 1) + Apicture0(i - 1, j) + _
                 Apicture0(i - 1, j + 1) + Apicture0(i, j - 1) + _
                 Apicture0(i, j + 1) + Apicture0(i + 1, j - 1) + _
                 Apicture0(i + 1, j) + Apicture0(i + 1, j + 1)
                          If c = 8 Then
                                ' เป็น ดำใน ยังไม่ต้องทำอะไร
                          Else
                          ' เป็น ดำนอก พิจารณาต่อ
                          ' พิสูจน์ว่ามีเพื่อนเป็นดำในหรือไม่ ถ้าไม่มีแสดงว่าเป็นเส้นจริง
  For a = -1 To 1
  For b = -1 To 1
    c = Apicture0(i - 1 + a, j - 1 + b) + Apicture0(i - 1 + a, j + b) + _
        Apicture0(i - 1 + a, j + 1 + b) + Apicture0(i + a, j - 1 + b) + _
        Apicture0(i + a, j + 1 + b) + Apicture0(i + 1 + a, j - 1 + b) + _
        Apicture0(i + 1 + a, j + b) + Apicture0(i + 1 + a, j + 1 + b)
   If c = 8 Then                    ' เพื่อนเป็น ดำใน
        Apicture11(i + a, j + b) = 1
        Else
    If (Apicture0(i, j) = 1) And (Apicture00(i, j) <> 100) Then
                                                  ' เพื่อนเป็น ดำนอก
       Apicture11(i + a, j + b) = 10
         End If
         End If
         Next b
         Next a
      cc = Apicture11(i - 1, j - 1) + Apicture11(i - 1, j) + _
           Apicture11(i - 1, j + 1) + Apicture11(i, j - 1) + _
           Apicture11(i, j + 1) + Apicture11(i + 1, j - 1) + _
           Apicture11(i + 1, j) + Apicture11(i + 1, j + 1)
                dd = cc Mod 10
                 If dd = 0 Then
' ดำนอก ไม่มีเพื่อนเป็น ดำใน แสดงว่าเป็น เส้นจริง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Apicture11(i, j) = 100
        Else
    ' ดำนอก มีเพื่อนเป็นเส้นจริง พิจารณาต่อ
    ' ใช้ cc ค่าเดิม
     ddd = cc \ 100
     Select Case ddd
    Case Is = 0
     ' ดำนอก มีเพื่อนดำใน ไม่มีเพื่อนเส้นจริง เป็น ขาว
    Apicture1(i, j) = 0
    Apicture11(i, j) = 0
    Case Is >= 2
    ' ดำนอก มีเพื่อนดำใน และมีเพื่อน 2เส้นจริง เป็น เส้นจริง
    Apicture11(i, j) = 100
    Case Is = 1
    ' ดำนอก มีเพื่อนดำใน และมีเพื่อน 1เส้นจริง
    ' ต้องดูต่อว่ามีเพื่อนดำในอยู่ตรงข้ามกับเพื่อนเส้นจริงหรือไม่
    ' ถ้าใช่ แสดงว่าตัวมันเองเป็นทางผ่านเส้นจริงซึ่งตัวมันคือ เส้นจริง
       RealLine = False
    ' พิจารณาจุดเพื่อนที่เป็นเส้นจริงว่า pixel ฝั่งตรงข้ามทั้ง 3
    ' เป็น ดำใน หรือไม่ ถ้าใช่ RealLine = True
       For a = -1 To 1
       For b = -1 To 1
         aa = (a * (-1))
         bb = (b * (-1))
  If Apicture00(i + a, j + b) = 100 Then ' เพื่อนที่เป็น เส้นจริง
     If (Abs(a) + Abs(b)) = 1 Then
    If (Apicture00(i + aa, j + bb) = 1) Or _

(Apicture00(i + aa + bb, j + bb + aa) = 1) Or (Apicture00(i + aa - bb, j +
bb - aa) = 1) Then

RealLine = True
End If
   End If
        If (Abs(a) + Abs(b)) = 2 Then ' มุม
          If (Apicture00(i + aa, j + bb) = 1) Or _

(Apicture00(i + aa, j + 0) = 1) Or (Apicture00(i + 0, j + bb) = 1) Then
RealLine = True
End If
 End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End If
            Next b
            Next a
             If RealLine = True Then  ' เป็นเส้นจริง
                Apicture11(i, j) = 100
             Else                     ' ลบเป็นขาว
                Apicture1(i, j) = 0
                Apicture11(i, j) = 0
            End If
            End Select
            End If
            End If
            End If
            Next i
            Next j                    ' คืนค่า
                pPBReverseApic1_Apic0
                pPBReverseApic11_Apic00
            Next R
End Sub

Public Sub pPBcollectscore() ' สำหรับรวบรวมคะแนนก่อนหาจุดกลาง
 For j = startY + 5 To endY - 5
   For i = startX + 5 To endX - 5
     If Apicture1(i, j) = 1 Then      ' ตาม้า บนซ้าย ล่างขวา
      If ((Apicture1(i - 1, j - 1) = 1) And (Apicture1(i, j + 1))) Or
((Apicture1(i, j - 1) = 1) And (Apicture1(i + 1, j + 1))) Then
                a = -1
                b = -2
               End If
             ' ตาม้า บนขวา ล่างซ้าย
  If ((Apicture1(i + 1, j - 1) = 1) And (Apicture1(i, j + 1))) Or
((Apicture1(i, j - 1) = 1) And (Apicture1(i - 1, j + 1))) Then
                a = 1
                b = -2
               End If
             ' ตาม้า ซ้ายบน ขวาล่าง
    If ((Apicture1(i - 1, j - 1) = 1) And (Apicture1(i + 1, j))) Or
((Apicture1(i - 1, j) = 1) And (Apicture1(i + 1, j + 1))) Then
                a = -2
                b = -1
                End If          ' ตาม้า ซ้ายล่าง ขวาบน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If ((Apicture1(i - 1, j + 1) = 1) And (Apicture1(i + 1, j))) Or
((Apicture1(i - 1, j) = 1) And (Apicture1(i + 1, j - 1))) Then
                a = -2
                b = 1
                End If                    ' แนวนอน --
If (Apicture1(i - 1, j) = 1) And (Apicture1(i + 1, j) = 1) Then
                a = 0
                b = -1
                End If                    ' แนวตั้ง |
If (Apicture1(i, j - 1) = 1) And (Apicture1(i, j + 1) = 1) Then
                a = -1
                b = 0
                End If                    ' เฉียงขึ้น /
If (Apicture1(i - 1, j + 1) = 1) And (Apicture1(i + 1, j - 1) = 1) Then
                a = -1
                b = -1
                End If                    ' เฉียงลง \
If (Apicture1(i - 1, j - 1) = 1) And (Apicture1(i + 1, j + 1) = 1) Then
                a = -1
                b = 1
                End If
                e = i
                f = j
    Do
        e = e + a
        f = f + b
        Ascore(e, f) = Ascore(e, f) + 1
    Loop Until ((e < startX + 5) Or (e > endX - 5)) Or ((f < startY + 5) Or
(f > endY - 5))
                e = i
                f = j
            Do
                e = e - a
                f = f - b
        Ascore(e, f) = Ascore(e, f) + 1
    Loop Until ((e < startX + 5) Or (e > endX - 5)) Or ((f < startY + 5) Or
(f > endY - 5))
                End If
            Next i
        Next j
End Sub
Public Sub pPBfindmaxscore() ' สำหรับหาจุดที่มีคะแนนมากที่สุด ซึ่งจุดนั้นคือจุดกลาง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ' หาจุดที่มีคะแนนสูงสุดซึ่งจุดนั้นเป็นจุดกลางของภาพ
        maxscore = 0
        For i = startX + 5 To endX - 5
            For j = startY + 5 To endY - 5
                If Ascore(i, j) > maxscore Then
                    maxscore = Ascore(i, j)
                    xi = i ' ค่าตำแหน่ง x ที่มีค่า maxscore
                    xj = j ' ค่าตำแหน่ง y ที่มีค่า maxscore
                End If
            Next j
        Next i
End Sub
Public Sub pPBshowcore() ' สำหรับแสดงภาพลายนิ้วมือพร้อมจุดกลาง
        For j = 0 To 255
            For i = 0 To 255
                If Apicture1(i, j) = 1 Then
                    CoreTabpicThinpic.PSet (i, j), RGB(0, 0, 0)
                Else
                    CoreTabpicThinpic.PSet (i, j), RGB(255, 255, 255)
                End If
            Next i
        Next j
    For i = 0 To 255
            CoreTabpicThinpic.PSet (i, xj), RGB(0, 0, 255)
            CoreTabpicScanOpenpic.PSet (i, xj), RGB(0, 0, 255)
    Next i
    For j = 0 To 255
            CoreTabpicThinpic.PSet (xi, j), RGB(0, 0, 255)
            CoreTabpicScanOpenpic.PSet (xi, j), RGB(0, 0, 255)
    Next j
    CoreTablblCoreX.Caption = xi
    CoreTablblCoreY.Caption = xj
End Sub

Private Sub CoreTabcmdCore_Click() ' ทำงานเมื่อกดปุ่มหาจุดกลาง

    RefreshCore = True
    CoreTabcmdCore.BackColor = &H80C0FF

    ' เคลียร์คะแนน
    pPBclearscore
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
startX = 0
endX = 255
startY = 0
endY = 255

' โอน array ภาพ ส่งให้ array ที่ใช้คำนวณ
pPBa2adetectcore

' รอบแรก
' ทำหนา
Round = 1
pPBfating
' ทำบาง
Round = 4
pPBprethiningV3
pPBthiningV3
' ลบจุดเดี่ยว และ จุดคู่
pPBerase1_2

' รอบสอง
' ทำหนา
Round = 1
pPBfating
' ทำบาง
Round = 2
pPBprethiningV3
pPBthiningV3
' ลบจุดเดี่ยว และ จุดคู่
pPBerase1_2

' หาจุดกลาง
startX = 5
endX = 250
startY = 0
endY = 250
pPBclearscore
pPBcollectscore
pPBfindmaxscore

'X1 = 25
'X2 = 182
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        'Y1 = 75
        'Y2 = 182

        X1 = 0
        X2 = 255
        Y1 = 0
        Y2 = 255
If ((xi >= X1) And (xi <= X2) And (xj >= Y1) And (xj <= Y2)) Then

            ' หา core รอบสอง
              startX = xi - 49
              endX = xi + 50
              startY = xj - 49
              endY = xj + 50
              pPBclearscore
              pPBcollectscore
              pPBfindmaxscore

            ' หา core รอบสาม
              startX = xi - 24
              endX = xi + 25
              startY = xj - 24
              endY = xj + 25
              pPBclearscore
              pPBcollectscore
              pPBfindmaxscore

            ' แสดงจุดกลาง
            pPBshowcore
            ' picture2arrray
            For j = 0 To 255
                For i = 0 To 255
                    ACoreTabThinpic(i, j) = Apicture1(i, j)
                Next i
            Next j
        Else
  Inform1 = MsgBox("จุดกลางที่พบอยู่ริมภาพเกินไป", vbExclamation + vbOKOnly,
"ข้อแนะนำ")
            SStab1.Tab = 0 ' Scan Tab
        End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub CoreTabcmdScanpic_Click() ' ทำงานเมื่อกดปุ่ม From Scan เพื่อถ่ายโอนภาพจาก ScanTab มายัง CoreTab

        RefreshCore = False
        CoreTabcmdCore.BackColor = &H8000000F

        ' แปลงค่าจากรูปภาพเป็นตัวเลขแล้วเก็บใน array
        pPBp2aScanTabScanpic
        startX = 0
        endX = 255
        startY = 0
        endY = 255
       For j = startY To endY
       For i = startX To endX
          ACoreTabScanOpenpic(i, j) = AScanTabScanpic(i, j + 22)
              If ACoreTabScanOpenpic(i, j) = 1 Then
                 CoreTabpicScanOpenpic.PSet (i, j), RGB(0, 0, 0)
              Else
           CoreTabpicScanOpenpic.PSet (i, j), RGB(255, 255, 255)
              End If
            Next i
       Next j
End Sub

Private Sub Form_Load() ' ทำงานเมื่อเปิดโปรแกรม
        SStab1.Tab = 0
        ScanTabpicScanpic.Tag = "12"
        startX = 0
        endX = 255
        startY = 0
        endY = 299
        Threshold = 100
        PointnumOK = 8
        PercentOK = 60
        RefreshCore = False
        RefreshTransform = False

        ' TheBox
        pPBsetTheBox

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Sub pPBsetTheBox() ' สำหรับกำหนดภาพทิศทางที่ใช้แทนตัวเลข จาก 0-9 ชื่อว่า
TheBox
        ' TheBox 0 ว่างเปล่า
        e = 0
        pPBresetTheBox

        ' TheBox 1 แนวนอน
        e = 1
        pPBresetTheBox
        For a = 1 To 8
            b = 4
            TheBox(e, a, b) = 1
        Next a

        ' TheBox 2 ลงขวาสูง
        e = 2
        pPBresetTheBox
        For b = 3 To 6
            a = ((b - 2) * 2) - 1
            TheBox(e, a, b) = 1
            TheBox(e, a + 1, b) = 1
        Next b

        ' TheBox 3 ลงขวา
        e = 3
        pPBresetTheBox
        For a = 1 To 8
            b = a
            TheBox(e, a, b) = 1
        Next a

        ' TheBox 4 ลงขวาต่ำ
        e = 4
        pPBresetTheBox
        For a = 3 To 6
            b = ((a - 2) * 2) - 1
            TheBox(e, a, b) = 1
            TheBox(e, a + 1, b) = 1
        Next a

        ' TheBox 5 แนวตั้ง
        e = 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        pPBresetTheBox
        For b = 1 To 8
            a = 4
            TheBox(e, a, b) = 1
        Next b

        ' TheBox 6 ลงซ้ายต่ำ
        e = 6
        pPBresetTheBox
        For a = 3 To 6
            b = 8 - (((a - 2) * 2) - 1)
            TheBox(e, a, b) = 1
            TheBox(e, a + 1, b) = 1
        Next a

        ' TheBox 7 ลงซ้าย
        e = 7
        pPBresetTheBox
        For a = 1 To 8
            b = 9 - a
            TheBox(e, a, b) = 1
        Next a

        ' TheBox 8 ลงซ้ายสูง
        e = 8
        For b = 3 To 6
            a = 8 - (((b - 2) * 2) - 1)
            TheBox(e, a, b) = 1
            TheBox(e, a + 1, b) = 1
        Next b

End Sub
Public Sub pPBresetTheBox() ' เคลียร์ค่า TheBox
        For c = 1 To 8
            For d = 1 To 8
                TheBox(e, c, d) = 0
            Next d
        Next c
End Sub

Private Sub MSComm1_OnComm() ' ทำงานเมื่อมีสัญญาณ input เข้าทาง serial port
  Dim Arr() As Byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim Color As Integer
Dim Buffer As Variant

        Select Case MSComm1.CommEvent
        Case comEvReceive

        ScanTabchkConnect.Enabled = False
        ScanTabchkAutoCapture.Enabled = False
        ScanTabcmdCapture.Enabled = False
        ScanTabcmdSavePic.Enabled = False
        ScanTabcmdRotate.Enabled = False

        Row = Row + 1

        Buffer = MSComm1.Input
        Arr = Buffer

        For Col = 0 To 255

        Data(Col, Row) = Arr(Col)
        Color = Data(Col, Row)

        ScanTabpicScanpic.PSet (Col, Row), RGB(Color, Color,
Color)

        Next Col

        If Row = 299 Then
        ScanTabchkConnect.Enabled = True
        ScanTabchkAutoCapture.Enabled = True
        ScanTabcmdCapture.Enabled = True
        ScanTabcmdSavePic.Enabled = True
        ScanTabcmdRotate.Enabled = True
        ScanTabfraSettingContrast.Enabled = True
        End If

        MSComm1.Output = Chr(255)

        End Select

End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Option1_Click() ' สำหรับกำหนดความเข้มของเส้น
        ScanTabpicScanpic.Tag = "15"
End Sub

Private Sub Option2_Click() ' สำหรับกำหนดความเข้มของเส้น
        ScanTabpicScanpic.Tag = "13"
End Sub

Private Sub Option3_Click() ' สำหรับกำหนดความเข้มของเส้น
        ScanTabpicScanpic.Tag = "12"
End Sub

Private Sub Option4_Click() ' สำหรับกำหนดความเข้มของเส้น
        ScanTabpicScanpic.Tag = "0"
End Sub
Public Sub pPBpic2direct() ' สำหรับแปลงภาพเป็นรหัสทิศทาง
        For g = 1 To 32
            For h = 1 To 32
                i = (8 * g) - 8
                j = (8 * h) - 8
                GetFirst = False
                Pointnum = 0

                For a = i To i + 7
                    For b = j To j + 7

                    If ATransformTabFromCorepic(a, b) = 1 Then
                      If GetFirst = False Then
                        c = a
                        d = b
                        GetFirst = True
                      End If
                        e = a
                        f = b
                    Pointnum = Pointnum + 1
                    End If
                        ' แนวต่างๆ มี 8 แนว จาก 16 ทิศ
                        '      --      \      |      /
                        ' 0  1  2  3  4  5  6  7  8
                If (GetFirst = False) Or ((c = a) And (d = f)) Or
(Pointnum < PointnumOK) Then
                    Abox(g, h) = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Else
        m = e - c
        n = f - d
    If m = 0 Then ' แนวตั้ง
      Abox(g, h) = 5
    End If
    If n = 0 Then ' แนวนอน
      Abox(g, h) = 1
    End If
    If (m <> 0) And (n <> 0) Then
        k = m / n
        Select Case k
              Case Is >= 3.5 Or k <= -3.5 ' direct 1 แนวนอน
              Abox(g, h) = 1
              Case Is > 1.5 And k < 3.5 ' direct 2 ขวาลงสูง
              Abox(g, h) = 2
              Case Is > 0.6 And k <= 1.5 ' direct 3 ลงขวา
               Abox(g, h) = 3
              Case Is >= 0.33 And k <= 0.6 ' direct 4 ลงขวาต่ำ
               Abox(g, h) = 4
             Case Is > -0.33 And k < 0.33 ' direct 5 แนวตั้ง
               Abox(g, h) = 5
              Case Is <= -0.6 And k <= -0.33 ' direct 6 ลงซ้ายต่ำ
               Abox(g, h) = 6
              Case Is <= -1.5 And k < -0.6 ' direct 7 ลงซ้าย
               Abox(g, h) = 7
              Case Is < -3.5 And k < -1.5 ' direct 8 ลงซ้ายสูง
               Abox(g, h) = 8
        End Select
               End If
               End If
               Next b
               Next a
               Next h
               Next g
        End Sub
Public Sub pPBdirect2pic() ' สำหรับรหัสทิศทางเป็นภาพทิศทาง
      For g = 1 To 32
          For h = 1 To 32
              i = (8 * g) - 8
              j = (8 * h) - 8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                c = 1
                d = 1
                For a = i To i + 7
                    For b = j To j + 7
                        e = Abox(g, h)
                  ATransformTabDirectpic(a, b) = TheBox(e, c, d)
                        d = d + 1
                    Next b
                        c = c + 1
                        d = 1
                Next a
            Next h
        Next g

    For j = startY To endY
        For i = startX To endX
            If ATransformTabDirectpic(i, j) = 1 Then
            TransformTabpicDirectpic.PSet (i, j), RGB(0, 0, 0)
                Else
             TransformTabpicDirectpic.PSet (i, j), RGB(255, 255, 255)
                End If
            Next i
        Next j

End Sub
Public Sub pPBfindboxcenter() ' สำหรับหาพิกัดของ Box ที่เป็นจุดกลาง
        boxX = (xi + 8) \ 8
        boxY = (xj + 8) \ 8
End Sub
Public Sub pPBAbox_Xbox() ' สำหรับถ่ายโอน Box ที่เก็บรหัสทิศทางทั้งภาพมายัง Box ที่
เก็บรหัสทิศทางพิเศษ
        For h = -7 To 7
            For g = -7 To 7
                Xbox(g, h) = Abox(boxX + g, boxY + h)
            Next g
        Next h
End Sub

Private Sub TransformTabcmdDirectTransform_Click() ' ทำงานเมื่อกดปุ่ม
Transform

        RefreshTransform = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        TransformTabcmdDirectTransform.BackColor = &H80C0FF

        pPBpic2direct
        pPBdirect2pic
        pPBfindboxcenter
        pPBAbox_Xbox
        pPBnum2str
        pPBXbox_A1
End Sub

Private Sub TransferTabcmdLoadThinPic_Click() ' ทำงานเมื่อกดปุ่ม Load เพื่อ
ถ่ายโอนภาพที่ทำ Thining แล้วจาก CoreTab มายัง TransformTab

        RefreshTransform = False
        TransformTabcmdDirectTransform.BackColor = &H8000000F

        pPBa2ap2pCore_Transfer
End Sub

Private Sub ScanTabchkConnect_Click() ' สำหรับ connect
        If ScanTabchkConnect.Value = 1 Then
            MSComm1.Settings = "115200,n,8,1"
            MSComm1.InputMode = comInputModeBinary
            MSComm1.InputLen = 256
            MSComm1.PortOpen = True
            MSComm1.RThreshold = 256
            ScanTabfraComport.Enabled = False

            'สั่งปุ่มให้ disable
            ScanTabchkConnect.ForeColor = &HFF&
            ScanTabchkAutoCapture.Value = 0
            ScanTabchkAutoCapture.Enabled = True
            ScanTabcmdCapture.Enabled = True
        Else
            MSComm1.PortOpen = False
            ScanTabfraComport.Enabled = True

            'สั่งปุ่มให้ enable
            ScanTabchkConnect.ForeColor = &H0&
            ScanTabchkAutoCapture.Enabled = False
            ScanTabcmdCapture.Enabled = False
        End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Private Sub ScanTabcmdCapture_Click() ' ทำงานเมื่อกดปุ่ม Capture
        MSComm1.InBufferCount = 0

        ScanTabpicScanpic.Cls

        Row = -1
        Col = 0

        ScanTabfraSettingContrast.Enabled = False

        MSComm1.Output = Chr(1)
        MSComm1.Output = Chr(ScanTabpicScanpic.Tag)
End Sub
Public Sub pPBp2aScanTabScanpic() ' สำหรับแปลงภาพที่สแกนได้มาเป็นรหัสตัวเลข 0-1
แล้วเก็บใน array
        endY = 299
        For j = startY To endY
            For i = startX To endX
                ' แปลงภาพเป็นตัวเลข ดำ > 1, ขาว > 0
                Colorcode = ScanTabpicScanpic.Point(i, j)
                Red = Colorcode Mod 256
                Green = (Colorcode \ 256) Mod 256
                Blue = Colorcode \ 65536
                Grayscale(i, j) = (Red \ 3) + (Green \ 3) + (Blue \ 3)
                If Grayscale(i, j) >= Threshold Then
                    ArrayBuff0(i, j) = 0
                Else
                    ArrayBuff0(i, j) = 1
                End If
                AScanTabScanpic(i, j) = ArrayBuff0(i, j)
            Next i
        Next j
End Sub
Private Sub ScanTabcmdRotate_Click() ' สำหรับพลิกภาพกลับหัว
        ' แปลงค่าจากรูปภาพเป็นตัวเลขแล้วเก็บใน array
        pPBp2aScanTabScanpic

        For j = startY To endY
            For i = startX To endX
                ArrayBuff1(i, j) = ArrayBuff0(endX - i, endY - j)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    AScanTabScanpic(i, j) = ArrayBuff1(i, j)
                    If AScanTabScanpic(i, j) = 1 Then
                        ScanTabpicScanpic.PSet (i, j), RGB(0, 0, 0)
                    Else
                        ScanTabpicScanpic.PSet (i, j), RGB(255, 255,
255)
                    End If
                Next i
            Next j
End Sub

Private Sub ScanTabcmdSavePic_Click() ' สำหรับบันทึกภาพ
            Dim sFile As String

              With CmmD
                .DialogTitle = "Save As"
                .CancelError = False
                'ToDo: set the flags and attributes of the common dialog
control
                .Filter = "BitMap (*.bmp)|*.bmp"
                .ShowSave
                If Len(.FileName) = 0 Then
                  Exit Sub
                Else
                  sFile = .FileName
                  SavePicture ScanTabpicScanpic.Image, sFile
                End If
              End With
End Sub

Private Sub ScanTaboptPort1_Click() ' สำหรับเลือก serial port1
            If ScanTaboptPort1.Value = True Then
                ScanTaboptPort2.Value = False
                MSComm1.CommPort = 1
            End If
End Sub

Private Sub ScanTaboptPort2_Click() ' สำหรับเลือก serial port2
            If ScanTaboptPort2.Value = True Then
                ScanTaboptPort1.Value = False
                MSComm1.CommPort = 2
            End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
Public Sub pPBa2ap2pCore_Transfer() ' สำหรับถ่ายโอนภาพและ arrayที่เก็บรหัสภาพ
จาก CoreTab มายัง TransferTab
        For j = startY To endY
            For i = startX To endX
                ATransformTabFromCorepic(i, j) =
ACoreTabThinpic(i, j)
                If ATransformTabFromCorepic(i, j) = 1 Then
                    TransformTabpicFromCorepic.PSet (i, j),
RGB(0, 0, 0)
                Else
                    TransformTabpicFromCorepic.PSet (i, j),
RGB(255, 255, 255)
                End If
            Next i
        Next j
End Sub

Private Sub SSTab1_Click(PreviousTab As Integer) ' ทำงานเมื่อมีการกดเปลี่ยน
Tab
  Dim selectedTab As Byte
        selectedTab = SStab1.Tab
        Select Case selectedTab
          ' Scan
          Case Is = 0
                startX = 0
                endX = 255
                startY = 0
                endY = 299
          ' Core
          Case Is = 1
          ' Transform
          Case Is = 2
                        endX = 255
                        endY = 255
                        startX = 0
                        startY = 0
                If RefreshCore = False Then
                        Inform1 = MsgBox("ต้องหาจุดกลางจองภาพก่อน",
vbExclamation + vbOKOnly, "ข้อแนะนำ")
                        SStab1.Tab = 1 ' Core Tab
                End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ' Identify
                Case Is = 3
                        If RefreshTransform = False Then
                            Inform1 = MsgBox("ต้องแปลงภาพเป็นภาพทิศทาง
ก่อน", vbExclamation + vbOKOnly, "ข้อแนะนำ")
                            SStab1.Tab = 3 ' Identify Tab
                        Else
                            endX = 255
                            endY = 255
                            startX = 0
                            startY = 0

                            ' สั่ง cursor ชี้ที่ record แรก
                            Adodc1.Recordset.MoveFirst
                            ' สั่ง textbox มองไม่เห็น

                            stDataFinger = RichTextBox1.Text
                            RichTextBox2.Text = stDataFinger
                            Correctscore = 0
                            Correctpercent = 0
                        End If
                ' DataBase
                Case Is = 4
                        endX = 255
                        endY = 255
                        startX = 0
                        startY = 0
            End Select
End Sub

Public Sub pPBnum2str() ' สำหรับเปลี่ยนตัวเลขเป็นตัวอักษร
        stDataFinger = ""
        stBuff = ""
        For h = -7 To 7
            For g = -7 To 7
                stBuff = Str(Xbox(g, h))
                stBuff2 = Trim(stBuff)
                stDataFinger = stDataFinger + stBuff2
            Next g
        Next h
        RichTextBox1.Text = stDataFinger
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub TransformTabcmdSavetoDataBase_Click() ' สำหรับเตือนเมื่อมีการกด
ปุ่ม Save to DataBase แต่ยังไม่ได้แปลงภาพเป็นภาพทิศทาง
        If RefreshTransform = False Then
                Inform1 = MsgBox("ต้องแปลงภาพเป็นภาพทิศทางก่อน",
vbExclamation + vbOKOnly, "ข้อแนะนำ")
        Else
                'pPBsaveDataBase
        End If
End Sub
Public Sub pPBXbox_A1() ' สำหรับแปลงตัวเลขใน Boxรหัสภาพทิศทางพิเศษ ให้เป็นตัวอักษร
ใน String เดียวกัน
        For h = -7 To 7
            For g = -7 To 7
                a = (h * 15) + g
                b = a + 113
                stBuff = Left(stDataFinger, b)
                stBuff2 = Right(stBuff, 1)
                AnstDataFinger(a) = stBuff2
            Next g
        Next h
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้