

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรักษาความปลอดภัยบ้านด้วยลายนิ้วมือ
SECURITY SYSTEM VIA FINGERPRINT IDENTIFICATION



โดย
นายเกรียงศักดิ์ สุดหอม 41012041
นายอังกูร พรไพศาลศิลป์ 41012077

อาจารย์ที่ปรึกษา
อาจารย์บุญชัยชนะ ภูระหงษ์

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรอุตสาหกรรมศาสตรบัณฑิต
ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหม.....

เลขทะเบียน 37119

ปีการศึกษา 2542

วัน, เดือน, ปี 4 ก.ย. 2543

สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาโท

ระบบรักษาความปลอดภัยบ้านด้วยลายนิ้วมือ

security system via fingerprint identification

โดย

นายเกรียงศักดิ์ สุขหอม 41012041

นายอังกูร พรไพศาลศิลป์ 41012077

อาจารย์ที่ปรึกษา

อาจารย์บุญชัชชนะ ภูระหงษ์

ปีการศึกษา

2542

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้
ปริญญาโทฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร
บัณฑิต

ลงชื่อ.....อาจารย์ที่ปรึกษา

(อาจารย์บุญชัชชนะ ภูระหงษ์)

คณะกรรมการการสอบปริญญาโท

กรรมการ

()

กรรมการ

()

กรรมการ

()

กรรมการ

()

กรรมการ

()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรักษาความปลอดภัยบ้าน

จัดทำโดย	นายเกรียงศักดิ์ สุขหอม	41012041
	นายอังกูร พรไพศาลศิลป์	41012077
อาจารย์ที่ปรึกษา	อาจารย์บุญชัยชนะ ภูระหงษ์	
ปีการศึกษา	2542	

บทคัดย่อ

ในปัจจุบัน การตรวจสอบลายนิ้วมือมีบทบาทมากขึ้น ในการแยกแยะลักษณะเฉพาะของแต่ละบุคคล ซึ่งสามารถนำไปใช้งานต่างๆ ได้อย่างมีประสิทธิภาพ ปรินูญานินพนธ์นี้ได้แสดงการแก้ไขภาพลายนิ้วมือพร้อมทั้งปรับปรุงให้ดีขึ้นตามสภาพความเป็นจริง ปรับปรุงเทคนิคการตรวจเทียบให้ถูกต้อง ซึ่งเป็นส่วนของการใช้งานและทำอัตโนมติ จากการทดสอบกลายนิ้วมือ จะถ่ายเป็นภาพระดับเทาเข้ามา ต่อมาได้แปลงข้อมูลภาพเป็นข้อมูลทิศทางของฮิสโตแกรมทิศทาง ถ้าผลการเปรียบเทียบเป็นลายนิ้วมือเดียวกันจะมีสัญญาณออกไปควบคุมอุปกรณ์ภายนอก แต่ถ้าไม่ก็จะไม่มีสัญญาณออกไปควบคุมอุปกรณ์ภายนอก ซึ่งโปรแกรมจะทำงานโดยอัตโนมัติ

SECURITY SYSTEM VIA FINGERPRINT IDENTIFICATION

By Mr.Kriengsak Sudhom 41012041

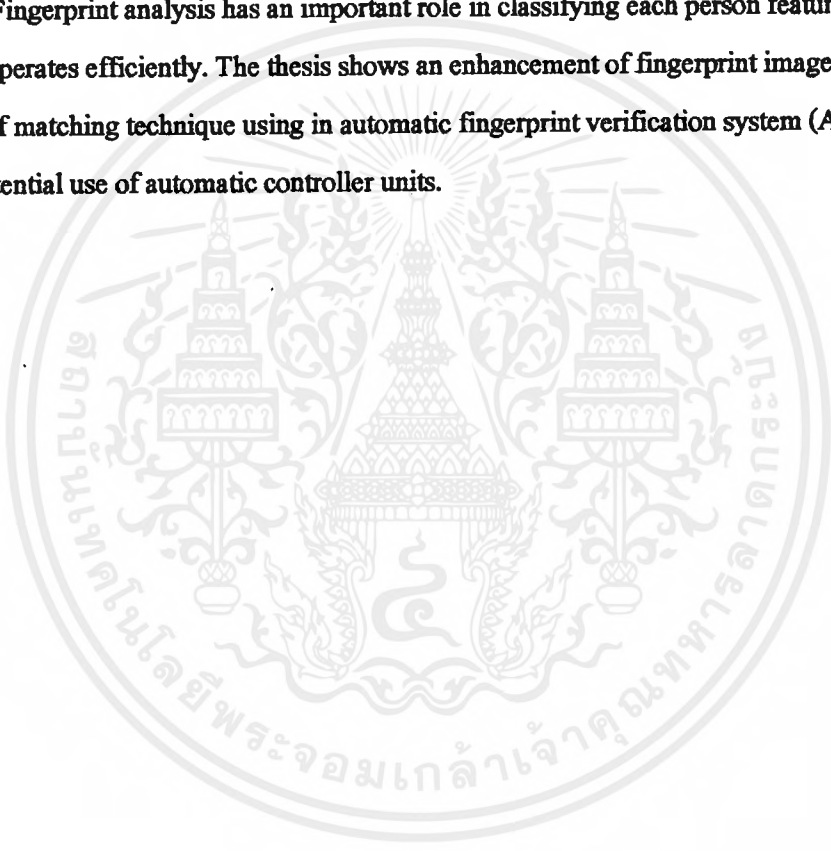
Mr.Ungoon Pornpaisarnsin 41012077

Adviser Boonchana Phurahong

Academic Year 1999

Abstract

Now, Fingerprint analysis has an important role in classifying each person features. That make process operates efficiently. The thesis shows an enhancement of fingerprint images and improvement of matching technique using in automatic fingerprint verification system (AFVS). It proofs for a potential use of automatic controller units.



กิตติกรรมประกาศ

ปริญญาโทฉบับนี้สามารถสำเร็จได้ด้วยคำแนะนำและคำปรึกษาจากอาจารย์
บุญชนะ ภูระหงษ์ (อาจารย์ที่ปรึกษา) ขอขอบคุณอาจารย์พิทักษ์ ชรรณวาริน และอาจารย์มยุรี
เลิศเวชกุล ที่ได้ให้คำแนะนำและให้ความช่วยเหลือในทุกๆด้าน และขอขอบพระคุณ ภาคเทคนิค
อุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้
ให้การสนับสนุน จนกระทั่งโครงการนี้สำเร็จลุล่วงไปด้วยดี
คุณค่าและประโยชน์ของปริญญาโทฉบับนี้ขอมอบแก่ผู้มีพระคุณทุกท่าน

นายเกรียงศักดิ์ สุตหอม
นายอังกร พรไพศาลศิลป์

สารบัญ

บทคัดย่อ

Abstract

กิตติกรรมประกาศ

	เรื่อง	หน้า
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและความเป็นมาของลายนิ้วมือ	2
	2.1 ประวัติโดยย่อของเครื่องอ่านลายนิ้วมือ	2
	2.2 ชนิดและความรู้เบื้องต้นทั่วไปของลายนิ้วมือ	3
บทที่ 3	สถาปัตยกรรมของระบบรักษาความปลอดภัยบ้าน	18
	ขั้นตอนการทำงานของระบบ	18
	1. ส่วนของการรับภาพลายนิ้วมือ	19
	2. ส่วนของการประมวลผล	20
	1. INPUT	20
	2. PREPROCESS	21
	- การปรับปรุงภาพด้วยวิธี Sharpen	21
	- การปรับปรุงภาพด้วยวิธี Histogram Equalize	22
	- การปรับปรุงภาพด้วยวิธี High Pass Filter	23
	3. RECOGNIZE	25
	4. HISTOGRAM	27
	- การใช้วิธีการฮิสโตแกรมทิศทางแทนฮิสโตแกรมเดิม	28
	- วิธีการแปลงข้อมูลภาพเป็นข้อมูลทิศทาง 4 ทิศทาง	30
	- ตัวอย่างการพิจารณาทิศทางขนาด 8 x 8 จุดภาพเป็นหนึ่งบล็อก	31
	- การพิจารณาหาทิศทาง โดยรอบ $G(i_{m,j_m})$ 4 ทิศทาง	32
	5. OUTPUT	34
	3. ส่วนควบคุมการแสดงผลทางเอาท์พุต	35
	การสร้างฮิสโตแกรมทิศทาง	36
	การเตรียมข้อมูลเพื่อจะนำไปบันทึกลงฐานข้อมูล	36
	1. การแปลงข้อมูลภาพลายนิ้วมือ	36
	2. การแบ่งพื้นที่ I แพทเทิร์นออกเป็นพื้นที่ย่อย	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4	การทดลอง	41
	การทดลองเกี่ยวกับการปรับปรุงภาพ	41
	สรุปผลการทดลองการปรับปรุงภาพ	45
	การทดลองกับภาพที่เกิด Error ขึ้น	46
บทที่ 5	สรุป,ปัญหาในการทำโครงการและข้อเสนอแนะ	50
	5.1 สรุป	50
	5.2 ปัญหา	51
	5.3 ข้อเสนอแนะ	51

ภาคผนวก ก Code โปรแกรมที่ใช้ทำการตรวจเทียบลายนิ้วมือ

ภาคผนวก ข แสดงอุปกรณ์ Hardware

ภาคผนวก ค แสดงรายละเอียดของบอร์ดควบคุม CP-S8252

บรรณานุกรม



สารบัญรูป

	รูป	หน้า
บทที่2	รูปที่1แสดงตัวอย่างแบบต่างๆของลายนิ้วมือ	3
	รูปที่2แสดงลายเส้นที่เกิดขึ้นระหว่างเส้นนูนกับเส้นร่อง	4
	รูปที่3แสดงจุดลักษณะสำคัญพิเศษบนนิ้วมือ	4
	รูปที่4แสดงการเดินของเส้นขอบ	5
	รูปที่5แสดงลักษณะของสันคอน	6
	รูปที่6แสดงลักษณะจุดใจกลาง	6
	รูปที่7แสดงลักษณะบริเวณลายนิ้วมือที่อยู่ภายใน	7
	รูปที่8แสดงลักษณะโค้งราบ	7
	รูปที่9แสดงลักษณะโค้งกระโจม	8
	รูปที่10แสดงลักษณะมัดหวายปิดขวา	8
	รูปที่11แสดงลักษณะมัดหวายปิดซ้าย	9
	รูปที่12แสดงลักษณะกั้นหอยธรรมดา	9
	รูปที่13แสดงลักษณะกั้นหอยกระเป๋ากลางปิดขวา	10
	รูปที่14แสดงลักษณะกั้นหอยกระเป๋ากลางปิดซ้าย	10
	รูปที่15แสดงลักษณะกั้นหอยกระเป๋าช้างปิดขวา	10
	รูปที่16แสดงลักษณะกั้นหอยกระเป๋าช้างปิดซ้าย	11
	รูปที่17แสดงลักษณะมัดหวายคู่แบบ1	11
	รูปที่18แสดงลักษณะมัดหวายคู่แบบ2	12
	รูปที่19แสดงลักษณะซับซ้อน	12
	รูปที่20แสดงลักษณะเส้นวกกลับ	13
รูปที่21แสดงการนับเส้นลายนิ้วมือ	14	
รูปที่22แสดงการสาวเส้นลายนิ้วมือ	15	
บทที่3	รูปที่3.1แสดงการประกอบอุปกรณ์ที่ใช้ในระบบ	18
	รูปที่3.2แสดงภาพการวางตำแหน่งมือ	19
	รูปที่3.3แสดงการรับภาพจากกล้องนำเข้าสู่ Form การปรับปรุงภาพและบันทึกภาพ	19
	รูปที่3.4แสดงตัวโปรแกรมที่รับภาพจากกล้อง	20
	รูปที่3.5แสดงการรับภาพจากกล้องนำเข้าสู่ Form การปรับปรุงภาพและบันทึกภาพ	21
	รูปที่3.6แสดงรูปที่ทำการปรับปรุงภาพแบบ Sharpen Filter	22

สารบัญรูป(ต่อ)

	รูป	หน้า
	รูปที่3.7แสดงรูปที่ทำการปรับปรุงภาพแบบ Histogram Equalize	22
	รูปที่3.8แสดงรูปที่ยังไม่ได้ผ่าน High Pass Filter	23
	รูปที่3.9แสดงรูป Histogram ของภาพลายนิ้วมือที่ยังไม่ได้ผ่าน High Pass Filter	23
	รูปที่3.10แสดงรูปที่ผ่าน High Pass Filter	24
	รูปที่3.11แสดงรูป Histogram ของรูปลายนิ้วมือที่ผ่าน High Pass Filter	24
	รูปที่3.12แสดงการติดตามหาจุดภาพตามลายเส้น	25
	รูปที่3.13แสดงการแทนจุดภาพของช่องหน้าต่างที่สแกนไปตามแกน x_i และแกน y_i	26
	รูปที่3.14แสดงจุดภาพ,แนวแกน x_i ,แกน y_i และแกนเฉลี่ยรวม i	26
	รูปที่3.15รูปตัวอย่างลายนิ้วมือ	27
	รูปที่3.16รูปแสดงกราฟฮิสโตแกรมของลายนิ้วมือข้างบน	28
	รูปที่3.17รูปแสดงทิศทางที่เป็นไปได้ 4 ทิศทาง	29
	รูปที่3.18รูปแสดงการหาทิศทางของโปรแกรมหาทิศทางลายนิ้วมือ	30
	รูปที่3.19รูปแสดงการหาทิศทางของ S_0	31
	รูปที่3.20รูปแสดงการหาทิศทาง โดยรอบทั้ง 4 ทิศทาง	32
	รูปที่3.21รูปแสดงตารางการบันทึกข้อมูลใน database	34
	รูปที่3.22รูปแสดง Form แสดงผลว่าเป็นเจ้าของลายนิ้วมือคนเดียวกัน	34
	รูปที่3.23รูปแสดง Form เมื่อผลการเปรียบเทียบออกมาไม่ใช่เจ้าของลายนิ้วมือ	35
	รูปที่3.24รูปแสดง Microcontroller Board CP-S8252	35
	รูปที่3.25รูปแสดงทิศทางรวมมากที่สุดเมื่อแทนทิศทางเดียวกันต่อหนึ่งบล็อก	37
	รูปที่3.26รูปแสดงโปรแกรมหาทิศทาง 4 ทิศทาง	38
	รูปที่3.27รูปแสดงโปรแกรมหาทิศทาง 8 ทิศทาง	38
	รูปที่3.28รูปแสดงขั้นตอนการเตรียมข้อมูล	39
	รูปที่3.29รูปแสดงการแบ่งภาพออกเป็น 9 พื้นที่	40
บทที่4	รูปที่4.1รูปแสดงลายนิ้วมือที่ยังไม่ได้ปรับปรุงภาพ	41
	รูปที่4.2รูปแสดงลายนิ้วมือที่ทำการปรับปรุงภาพโดยวิธี Sharpen	41

สารบัญรูป(ต่อ)

รูป	หน้า
รูปที่4.3รูปแสดงลายนิ้วมือที่ทำการปรับปรุงภาพ โดยวิธี Histogram Equalize	42
รูปที่4.4รูปแสดงลายนิ้วมือที่ทำการปรับปรุงภาพ โดยวิธี High Pass Filter	42
รูปที่4.5รูปแสดงภาพการหาทิศทางของภาพที่ยังไม่ได้รับการปรับปรุงภาพ	43
รูปที่4.6รูปแสดงการหาทิศทางของภาพที่ปรับปรุงภาพแบบ Sharpen	43
รูปที่4.7รูปแสดงการหาทิศทางของภาพที่ทำการปรับปรุงภาพแบบ Histogram Equalize	44
รูปที่4.8รูปแสดงการหาทิศทางของภาพที่ทำการปรับปรุงภาพแบบ High Pass Filter	44
รูปที่4.9รูปแสดงลักษณะการแบ่งภาพออกเป็น 9 ส่วน	46
รูปที่4.10รูปแสดงลักษณะของลายนิ้วมือที่สมบูรณ์	47
รูปที่4.11รูปแสดงภาพที่ไม่สมบูรณ์ด้านซ้ายของภาพ	47
รูปที่4.12รูปแสดงภาพที่ไม่สมบูรณ์ด้านบนของภาพ	48
รูปที่4.13รูปแสดงภาพที่ไม่สมบูรณ์ด้านขวาของภาพ	48
รูปที่4.14รูปแสดงภาพที่ไม่สมบูรณ์ด้านล่างของภาพ	49

บทที่ 1

บทนำ

ในปัจจุบันการจะจำแนกบุคคลว่าเป็นใครนั้น นอกจากจะใช้ภาพถ่าย รหัสลับประจำตัว บัตรประชาชน หรือบัตรแม่เหล็กแล้ว ก็กำลังจะถูกแทนที่ด้วยระบบพิสูจน์ลักษณะจำเพาะของอวัยวะของเราแทน ระบบรักษาความปลอดภัยนี้เรียกว่า “Biometrics Identification” ลักษณะการทำงานของระบบจะบันทึกลักษณะทางกายภาพของอวัยวะส่วนใดส่วนหนึ่งของร่างกายในรูปรหัสดิจิทัล เก็บไว้เป็นฐานข้อมูลกลางของระบบ เมื่อมีการเข้าออกหรือกระทำอย่างใดกับสิ่งที่ป้องกันไว้ จะต้องผ่านการตรวจสอบกับข้อมูลกลางเพื่อที่จะตัดสินใจในการอนุญาตหรือไม่กับบุคคลนั้น

ในการศึกษานี้ได้มุ่งเน้นพัฒนา algorithm ของระบบเปรียบเทียบลายนิ้วมือ โดยใช้กล้องวีดีโอและอุปกรณ์ทางซอฟต์แวร์เพื่อช่วยในการ detect ลายนิ้วมือของบุคคลทุกๆ ไปและแก้ไขให้อยู่ในรูปขาวดำ (binary) พร้อมทั้งปรับการแก้ไขภาพให้ชัดเจน ต่อเนื่องให้สมบูรณ์ใกล้เคียงกับความเป็นจริงและพิจารณาจุดสำคัญบนลายนิ้วมือเพื่อเป็นประโยชน์ในการวิเคราะห์ภาพตลอดจนเทคนิคต่างๆ ในการพิสูจน์ลายนิ้วมือของบุคคลหนึ่งต่อลายนิ้วมือที่มีอยู่ จากนั้นเก็บไว้ในฐานข้อมูลเพื่อใช้เป็นแบบตรวจสอบและกำหนดให้เป็นชื่อ file หนึ่งๆ ซึ่งแตกต่างกันออกไปตามกลุ่มแบบโดยมีทั้งหมด 12 แบบมาตรฐานซึ่งมีข้อมูลเป็นจำนวนมากที่ได้นำมาเก็บบันทึกไว้เป็นหลักฐาน

บทที่ 2 ทฤษฎีและความเป็นมาของลายนิ้วมือ

2.1 ประวัติโดยย่อของเครื่องอ่านลายนิ้วมือ

เครื่องอ่านลายนิ้วมือ (Fingerprint Reader) ได้ใช้ประโยชน์ในกิจการที่เกี่ยวกับความปลอดภัยและการพิสูจน์หลักฐานของกรมตำรวจเป็นส่วนมาก เครื่องอ่านชนิดนี้จะแบ่งออกเป็น 2 ชนิด คือ แบบที่ใช้ในระบบ Fingerprint Identification และระบบ Fingerprint Verification ทั้งสองระบบมีความแตกต่างกันคือ ระบบ Identification จะเป็นระบบที่ทำการเปรียบเทียบลายนิ้วมือที่ป้อนเข้ามา กับลายนิ้วมือที่อยู่ในฐานข้อมูลของระบบทั้งหมด แล้วหาลายนิ้วมือที่ใกล้เคียงกันที่สุด พร้อมทั้งดึงข้อมูลส่วนบุคคลที่ตรงกันกับนิ้วมือนั้นออกมา ระบบนี้จะเป็นการเปรียบเทียบ (Matching) แบบ One-to-many เครื่องอ่านลายนิ้วมือแบบนี้ ส่วนมากอ่านจากกระดาษที่พิมพ์ลายนิ้วมือเอาไว้ด้วยหมึกพิมพ์ มักใช้ในกิจการของพิสูจน์หลักฐานของกรมตำรวจ ส่วนแบบ Verification นั้น จะเปรียบเทียบลายนิ้วมือที่ป้อนเข้ามา กับลายนิ้วมือที่อยู่ในฐานข้อมูลไปที่ละอัน เมื่อได้รับอันที่ตรงกันก็จะดึงเอาข้อมูลที่เกี่ยวข้องกันออกมา ระบบนี้ถือว่าการ Matching แบบ One-to-One โดยเครื่องอ่านลายนิ้วมือที่ได้ออกแบบในการอ่านแบบ Total Reflection Method ได้โดยตรง แล้วทำการประมวลผลเบื้องต้น (Preprocessing) และ Matching แบบ Real time ใช้เวลาเพียงชั่วครู่ก็สามารถตอบกลับมาได้ว่า เป็นลายนิ้วมือที่ต้องการหรือไม่ ระบบนี้มักใช้ในกิจการด้านรักษาความปลอดภัย เช่น เชื่อมกับระบบเปิดปิดประตูใหญ่ของสถานที่สำคัญ หรือใช้ในกิจการธนาคาร เป็นต้น

ปัจจุบันระบบเหล่านี้มีราคาแพงมาก เพราะเป็นเทคโนโลยีนำเข้ามาทั้งหมด และในประเทศไทยก็เริ่มที่จะใช้ระบบเหล่านี้ในกิจการด้านต่าง ๆ มากขึ้น ประเทศที่ได้ส่งออกเทคโนโลยีประเภทนี้ เช่น ญี่ปุ่น มีระบบ AFIS-NEC และระบบ AFVS-HITACHI, อเมริกามีระบบ PRINTRAX PIV100, Ridge Reader และ IDX-40 เป็นต้น หากประเทศไทยสามารถสร้างระบบเหล่านี้ขึ้นมาเอง ก็จะทำให้ประหยัดค่าใช้จ่ายและงบประมาณลงอย่างมาก

2.2 ชนิดและความรู้เบื้องต้นทั่วไปของลายนิ้วมือ



โค้งราบ

(Plain Arch = PA)



โค้งกระโจม

(Tented Arch = TA)



มัดทวยสัดขวา

(Right Slant Loop = RSL)



มัดทวยสัดซ้าย

(Left Slant Loop = LSL)



ก้นทอยธรรมดา

(Plain Whorl = W)



ก้นทอยกระเป๋ากลางสัดขวา

(Right Central Pocket=RCP)



ก้นทอยกระเป๋ากลางสัดซ้าย

(Left Central Pocket=LCP)



ก้นทอยกระเป๋าล้างสัดขวา

(Right Lateral Pocket=RLP)



ก้นทอยกระเป๋าล้างสัดซ้าย

(Left Lateral Pocket=LLP)



มัดทวยคู่หรือมัดทวยแฝดแบบ1

(Double Loop Type1 = D1)



มัดทวยคู่หรือมัดทวยแฝดแบบ2

(Double Loop Type2 = D2)



ซัดซ้อ

(Accidental Whorl=AW)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณารูปที่ 1 แบบต่าง ๆ จะเห็นว่าที่ผิวหนังบริเวณปลายนิ้วมือ ประกอบด้วยลายเส้นสองชนิด ชนิดหนึ่งเราเรียกว่า เส้นนูน อีกชนิดหนึ่งเราเรียกว่า รอยร่องหรือเส้นร่อง เส้นนูนกับเส้นร่องจะอยู่สลับกันไปตลอด

2.2.1 เส้นนูน (Ridges) คือการเกิดของรอยนูนที่อยู่สูงขึ้นมาจากผิวหนังส่วนนอกร

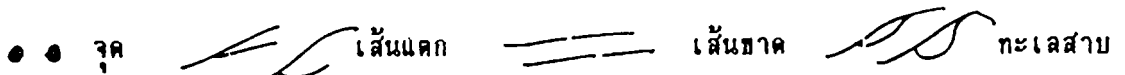
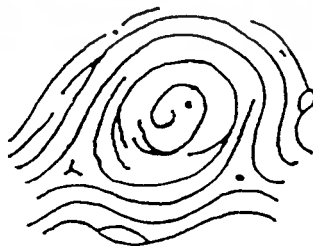
2.2.2 รอยร่องหรือเส้นร่อง (Furrows) คือรอยลึกที่อยู่ต่ำกว่าระดับของเส้นนูน

เส้นนูนกับเส้นร่องนี้อยู่สลับกันไป ดังนั้นจะแลเห็นได้ว่า ที่ผิวหนังบนนิ้วมือจะมีเส้นหนึ่งสูงขึ้นมาและเส้นหนึ่งอยู่ลึกต่ำลงไป (สลับกัน) โดยถ้าเราใช้นิ้วกดนิ้วมือไปบนกระดาษขาว จะมีลายเส้นสีดำกับสีขาวสลับกันไป เราเรียกลายสีดำว่า เส้นนูน เพราะคือส่วนของรอยนูนที่ถูกกับน้ำหมึก ดังนั้นเส้นร่องไม่ถูกกับน้ำหมึกและไม่ถูกกับกระดาษจึงเป็นสีขาว และถ้าเป็นการถ่ายรูปหรือ scan ภาพเข้าไปต้องทำให้เป็นภาพ binary ก่อนจึงจะได้ลายนิ้วที่เป็นลายเส้นสีดำกับสีขาวตามต้องการดังรูปที่ 2 (ก) และลายเส้นของลายนิ้วมือจะไม่มีการวิ่งหรือไหลเรียบร้อยไปตลอด ตามรูปที่ 2 (ข)



รูปที่ 2 แสดงลายเส้นที่เกิดขึ้นระหว่างเส้นนูนกับเส้นร่อง

แต่ลายนิ้วมือจะต้องวิ่งขาดตอน, บางที่เป็นรอยแตก, บางที่เกิดเป็นทะเลสาบ, บางทีก็เป็นจุด ดังรูปที่ 3



รูปที่ 3 แสดงจุดลักษณะสำคัญพิเศษบนนิ้วมือ

2.2.3 จุด, เส้นแตก, เส้นขาด, เส้นทะเลสาบ คือ ตำนานิบนลายนิ้วมือ ตำนานนี้เรียกว่า จุดลักษณะสำคัญพิเศษ (Characteristics) จุดลักษณะสำคัญพิเศษเหล่านี้จึงนำมาใช้เป็นหลักฐานสำคัญในการตรวจพิสูจน์เปรียบเทียบเพื่อยืนยันตัวบุคคล

ข้อควรจำ ลายเส้นในลายนิ้วมือเป็นหลักฐานสำคัญในการตรวจพิสูจน์ด้วยเหตุผล 2 ประการคือ

- (1) ลายเส้นของนิ้วมือจะไม่มี การเปลี่ยนแปลงเลยตั้งแต่เกิดจนกระทั่งตาย
- (2) ไม่มีบุคคลใด ๆ ที่จะมีลายนิ้วมือที่เหมือนกันหรือซ้ำกันได้เลย

2.2.4 คำจำกัดความที่สำคัญ 4 ข้อ ในลายนิ้วมือ ได้แก่

[ก] เส้นขอบ (Type Line)

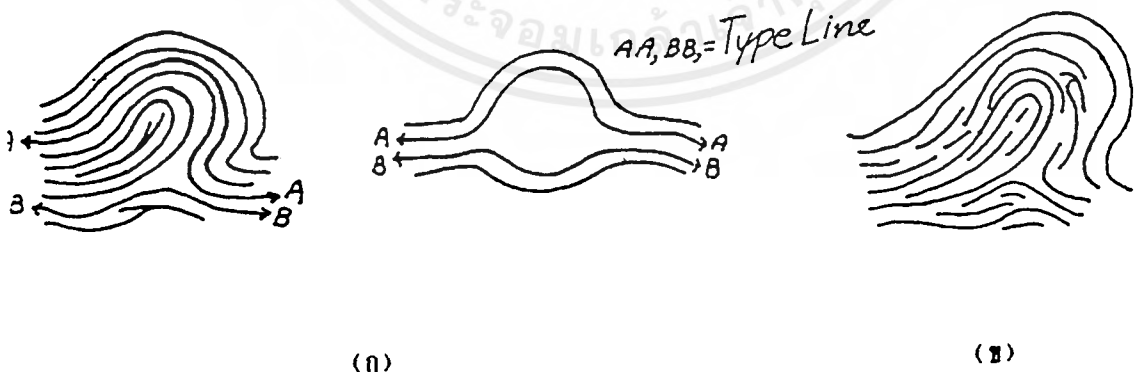
[ข] สันดอน (Delta)

[ค] จุดใจกลาง (Core)

[ง] บริเวณลายนิ้วมือที่อยู่ภายใน (Pattern Area)

[ก] เส้นขอบ (Type Line) คือ เส้นคู่ขนานคู่ในสุดซึ่งได้เดินทางมาคู่กันมาพอสมควร แล้วแยกออกจากกัน เพื่อที่จะโอบล้อมบริเวณลายนิ้วมือที่อยู่ภายใน ดังรูปที่ 4 (ก) และเส้น

ขอบไม่จำเป็นต้องเป็นเส้นยาวและราบเรียบติดต่อกันไปตลอด อาจจะเป็นเส้นที่ขาดหักงอกลางคันอย่างเห็นได้ชัด ถ้าเป็นกรณีนี้ เส้นที่อยู่ด้านนอกของเส้นขอบที่ขาดหักงอไปนั้น ถือว่าเป็นเส้นที่เดินหรือไหลต่อเนื่องกันไปเสมือนหนึ่งว่า เส้นขอบนั้นมีได้ขาดลงนั่นเอง ดูรูปที่ 4 (ข)



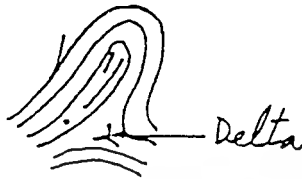
รูปที่ 4 แสดงการเดินทางของเส้นขอบ

[ข] สันดอน (Delta) คือ ลายเส้นในลายนิ้วมือซึ่งอยู่ตรงหน้า และใกล้ที่สุดกับ
กึ่งกลางของปากทางแยกของเส้นขอบ โดยสันดอนมีลักษณะตามรูปที่ 5

1. จุด



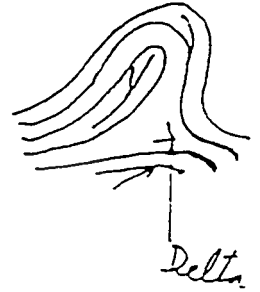
2. เส้นแตก



3. ปลายเส้น



4. เส้นหักมุม

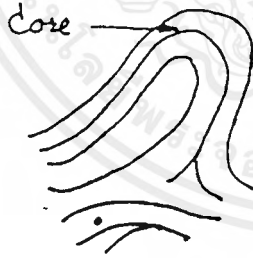


5. จุดใดจุดหนึ่งบนเส้น



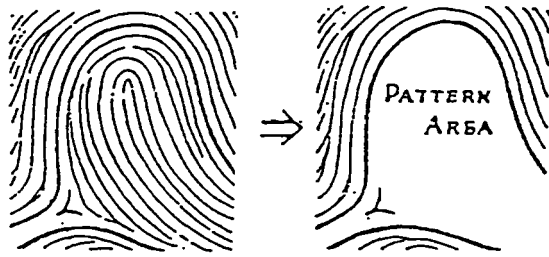
รูปที่ 5 แสดงลักษณะของสันดอน

[ค] จุดใจกลาง (Core) คือจุดใดจุดหนึ่งบนปลายเส้นหรือบนบ่า หรือไหล่ของ
เส้นวกกลับ ดูรูปที่ 6



รูปที่ 6 แสดงลักษณะจุดใจกลาง

[ง] บริเวณลายนิ้วมือที่อยู่ภายใน (Pattern Area) คือ พื้นที่บริเวณภายในของ
ลายนิ้วมือที่ถูกเส้นขอบโอบล้อม ดูรูปที่ 7



รูปที่ 7 แสดงลักษณะบริเวณลายนิ้วมือที่อยู่ภายใน

2.2.5 แบบของลายนิ้วมือ แบ่งออกเป็น 12 แบบคือ

[1] ไค้กราบ (Plain Arch = PA) คือ ลักษณะของลายเส้นในลายนิ้วมือ ที่ตั้งต้นจากขอบเล็บข้างหนึ่งแล้ววิ่งหรือไหลออกไปอีกข้างหนึ่ง ลักษณะลายเส้นดูง่าย ไม่มีเส้นเกือกมาไม่เกิดมุมแหลมคมที่เห็นได้ชัดตรงกลาง หรือไม่มีเส้นพุ่งสูงขึ้นตรงกลาง ดูรูปที่ 8



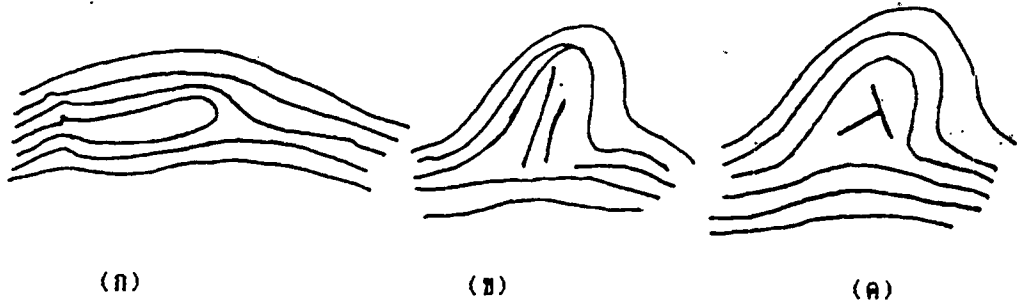
รูปที่ 8 แสดงลักษณะไค้กราบ

[2] ไค้กระโจม (Tented Arch = TA) คือ ลักษณะลายเส้นในลายนิ้วมือชนิดไค้กราบ แต่แตกต่างจากไค้กราบทั่วไป อาทิเช่น

(ก) มีลายเส้นเส้นหนึ่งหรือมากกว่า ซึ่งอยู่ตรงกลางไม่ได้วิ่งหรือไหลออกไปยังอีกข้าง

(ข) ลายเส้นที่อยู่ตรงกลางของลายนิ้วมือเส้นหนึ่งหรือมากกว่า เกิดเป็นเส้นพุ่งขึ้นจากแนวนอนหรือ

(ค) มีเส้นสองเส้นมาพบกันตรงกลางเป็นมุมแหลมคมหรือมุมฉาก



รูปที่ 9 แสดงลักษณะโค้งกระโจม

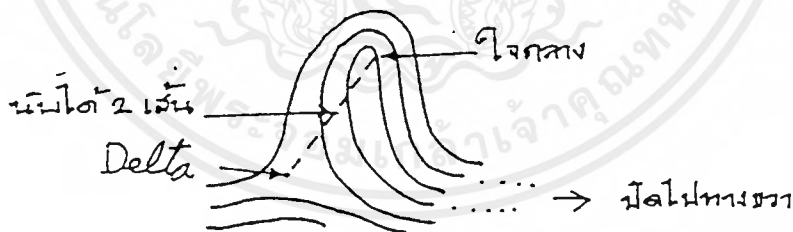
[3] มัดหวายบิดขวา (Right Slant Loop = RSL) หมายถึง ลายนิ้วมือแบบมัดหวายมืออยู่ประมาณ 65% ของลายนิ้วมือมนุษย์ ซึ่งมากกว่าลายนิ้วมือชนิดอื่น ๆ โดยมีกฎเกณฑ์ดังนี้

(ก) ต้องมีสันตอนข้อใดข้างหนึ่งเพียงข้างเดียว

(ข) ต้องมีเส้นวงกลับที่เห็นได้ชัดอย่างน้อยหนึ่งรูป

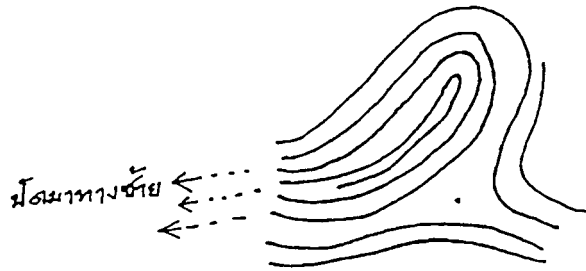
(ค) ต้องมีจุดใจกลางและต้องนับเส้นจากจุดสันตอนหนึ่งไปถึงจุดใจกลางได้อย่างน้อย 1 เส้น โดยเส้นที่นับนี้ต้องเป็นเส้นวงกลับที่สมบูรณ์อย่างน้อย 1 เส้น

ดังนั้นมัดหวายรูปใดที่มีปลายเส้นเกือบมามัดปลายไปทางมือขวาหรือด้านขวามือ เรียกว่า มัดหวายบิดขวา



รูปที่ 10 แสดงลักษณะมัดหวายบิดขวา

[4] มัดหวายบิดซ้าย (Left Slant Loop = LSL) คือ มัดหวายรูปใดที่มีปลายเส้นเกือบมามัดปลายไปทางมือซ้าย หรือทางด้านซ้าย เรียกว่า มัดหวายบิดซ้าย

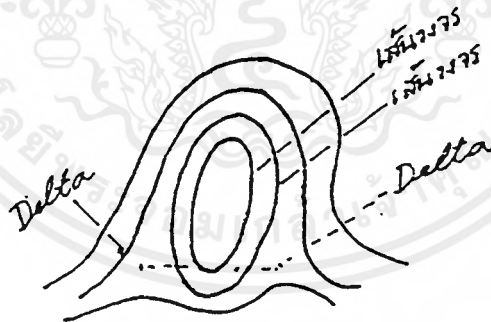


รูปที่ 11 แสดงลักษณะมัดหวายปิดซ้าย

[5] ก้นหอยธรรมดา (Plain Whorl = W) คือ ลายนิ้วมือที่มีเส้นเวียนรอบเป็นวง จร วงจรนี้อาจมีลักษณะเหมือนลานนาฬิกา เหมือนรูปไข่ เหมือนวงกลม หรือลักษณะอื่น ๆ

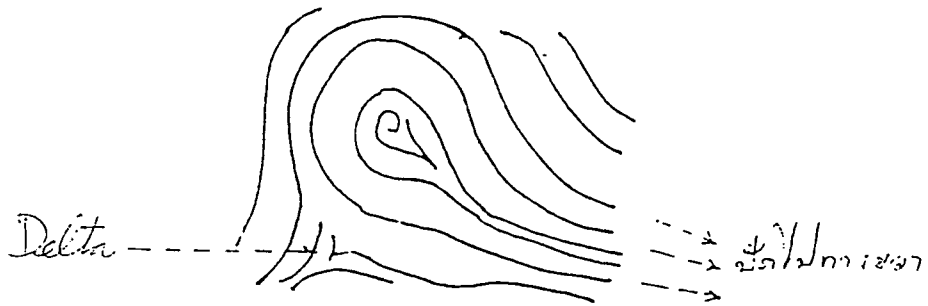
(1) ต้องมีจุดสันดอน 2 แห่ง และหน้าจุดสันดอนเข้าไปจะต้องมีรูปวงจรหรือเส้นเวียนอยู่ข้างหน้าจุดสันดอนทั้ง 2 จุด

(2) ถ้าลากเส้นสมมติจากจุดสันดอนข้างหนึ่งไปยังสันดอนอีกข้างหนึ่ง เส้นสมมติจะต้องสัมผัสเส้นวงจร หน้าจุดสันดอนทั้ง 2 ข้างอย่างน้อย 1 เส้น



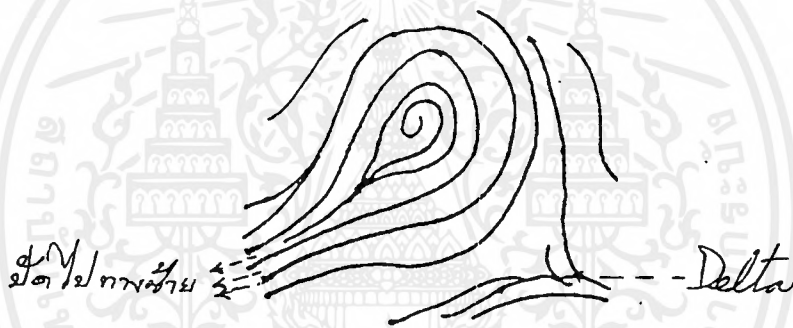
รูปที่ 12 แสดงลักษณะก้นหอยธรรมดา

[6] ก้นหอยกระเป๋ากลางปิดขวา (Right Central Pocket = RCP) คือ ลายนิ้วมือแบบก้นหอยธรรมดา แต่ผิดกันตรงที่ลากเส้นสมมติจากสันดอนหนึ่งไปยังสันดอนหนึ่ง เส้นสมมติจะไม่สัมผัสกับเส้นวงจรที่อยู่ตอนใน



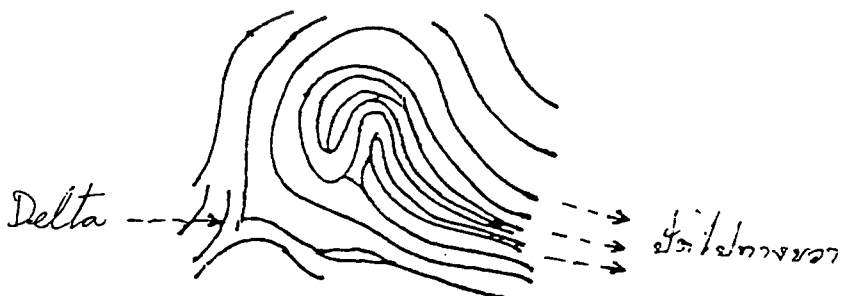
รูปที่ 13 แสดงลักษณะก้นหอยกระเป๋ากลางปิดขวา

[7] ก้นหอยกระเป๋ากลางปิดซ้าย (Left Central Pocket = LCP) คือ ลักษณะที่ตรงข้ามกับปิดขวาโดยปิดไปทางซ้าย



รูปที่ 14 แสดงลักษณะก้นหอยกระเป๋ากลางปิดซ้าย

[8] ก้นหอยกระเป๋าช้างปิดขวา (Right Lateral Pocket = RLP) คือ ลายนิ้วมือที่มีจุดใจกลางต่างจากก้นหอยทั้ง 3 แบบที่กล่าวมาแล้ว และสันตอนมีเพียงข้างเดียว โดยที่มีปลายเส้นจากก้นปิดไปทางขวามือ



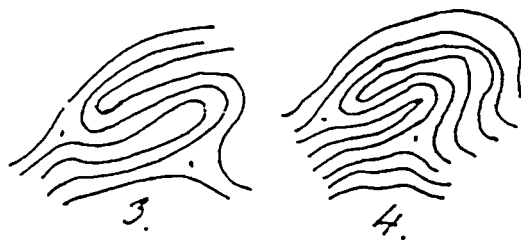
รูปที่ 15 แสดงลักษณะก้นหอยกระเป๋าช้างปิดขวา

[9] ก้นหอยกระเปาะข้างปิดซ้าย (Left Lateral Pocket = LLP) คือ ลายนิ้วมือที่มีจุดใจกลางต่างจากก้นหอยทั้ง 3 แบบที่กล่าวมาแล้ว และสันดอมนมีเพียงข้างเดียว โดยที่มีลายเส้นจากก้นปิดไปทางซ้ายมือ



รูปที่ 16 แสดงลักษณะก้นหอยกระเปาะข้างปิดซ้าย

[10] มัดหวายคู่หรือมัดหวายแฝดแบบ 1 (Double Loop Type1 = D1) คือ ลายนิ้วมือที่มีรูปคล้ายกับลายนิ้วมือแบบมัดหวาย 2 รูป มากอดหรือมากล้ำกันเป็นลายนิ้วมือที่มีสันดอมน 2 สันดอมน, รูปมัดหวาย 2 รูปมัดหวายและสองรูปที่ปรากฏอยู่ไม่จำเป็นต้องมีขนาดเท่ากัน



รูปที่ 17 แสดงลักษณะมัดหวายคู่แบบ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[11] มัดหน่วยคู่หรือมัดหน่วยแฝดแบบ 2 (Double Loop Type2 = D2) คือ ลายนิ้วมือที่มีรูปคล้ายกับลายนิ้วมือแบบมัดหน่วย 2 รูป มากอดหรือมากล้ำกันเป็นลายนิ้วมือที่มีสันดอน 2 สันดอน, รูปมัดหน่วย 2 รูปมัดหน่วย และสองรูปที่ปรากฏอยู่ไม่จำเป็นต้องมีขนาดเท่ากัน แต่ตรงข้ามกับแบบที่ 1 ตรงที่ลายนิ้วมือจะพลิกกลับ



รูปที่ 18 แสดงลักษณะมัดหน่วยคู่แบบ 2

[12] รั้งซ้อน (Accidental Whorl = AW) คือ ลายนิ้วมือที่ไม่เหมือนลายนิ้วมือชนิดอื่น ๆ มีลักษณะพิเศษเห็นได้ง่ายไม่จัดเข้าเป็นลายนิ้วมือชนิดใดชนิดหนึ่งใดโดยเฉพาะ เป็นลายนิ้วมือที่ประกอบด้วยลายนิ้วมือ 2 แบบมาผสมกันและมีสันดอน 2 สันดอนหรือมากกว่า



รูปที่ 19 แสดงลักษณะรั้งซ้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ

ลายนิ้วมือของบุคคลบางรายอาจมีลักษณะพิเศษซึ่งน้อยมากที่จะมีลายเส้นแบบนี้ คือ เส้นเกือกม้าหรือเส้นวงกลับ คือ เส้นที่วิ่งหรือไหลกลับเข้าไปในบริเวณภายในของลายนิ้วมือ ซึ่งอยู่หน้าสันดอนเข้าไปแล้วโค้งกลับมาหรือวกกลับมาจากเดียวกับที่วิ่งหรือไหลเข้าไป และได้จำแนกออกไปดังนี้ เส้นวงกลับเพียงเส้นเดียว, เส้นวงกลับหลายเส้น และ ไม่มีเส้นวงกลับเลย (ซึ่งในที่นี้ไม่ได้นำมาพิจารณามากนัก เพราะลายนิ้วมือประเภทนี้มีน้อย นอกจากนี้เรายังสามารถมองไปเป็นแบบมั่วหยาบได้) ดังรูปที่ 20



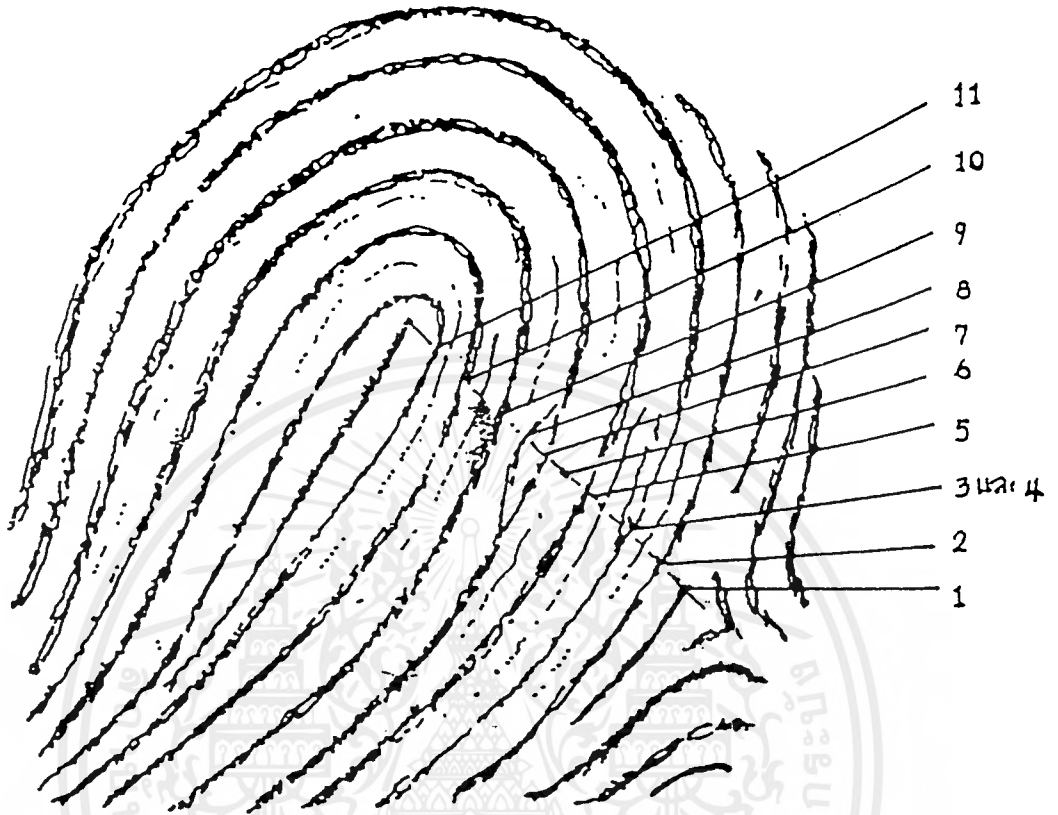
รูปที่ 20 แสดงลักษณะเส้นวงกลับ

2.2.6 หลักในการนับเส้นลายนิ้วมือ ได้วางหลักเกณฑ์ที่แน่นอนในขั้นต้นไว้ดังนี้

- (1) ใช้เส้นสมมติซึ่งมืออยู่ได้แน่นอนขยาย ทาบจากสันดอนไปยังใจกลาง
- (2) ให้สันดอนเป็นจุดเริ่มต้น และใจกลางเป็นจุดหมายปลายทาง
- (3) ให้นำจำนวนเส้นทุกเส้นที่เส้นสมมติได้แน่นอนขยายผ่านหรือแตะ ยกเว้นสันดอนและใจกลางไม่ต้องนับ
- (4) ถ้าเป็นเส้นแตกหรือเส้นเกาะเส้นสมมติที่ทาบไปแตะหรือผ่านตรงจุดเริ่มแตกพอดี ให้นับเป็น 2 เส้น
- (5) ถ้าเป็นจุด ให้พิจารณาว่า จุดนั้น ๆ มีความหนาเท่ากับเส้นหรือไม่ และเป็นจุดที่แท้จริงหรือเปล่า ถ้าใช่และมีความหนาเท่ากับเส้น ให้นับจุดนั้นรวมไปด้วย
- (6) เส้นที่บางกว่าลายเส้นที่แท้จริง ไม่นับ
- (7) ทุกเส้นที่นับต้องมีช่องว่างขาวพอจึงจะนับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 21 ประเภทมัดหวาย ในการนับเส้น



1. ปลายเส้น, 2. เส้น, 3.และ4. เส้นแตก, 5. เส้น, 6. จุด, 7.และ8. เส้นเกาะ
9.,10และ11. เส้น

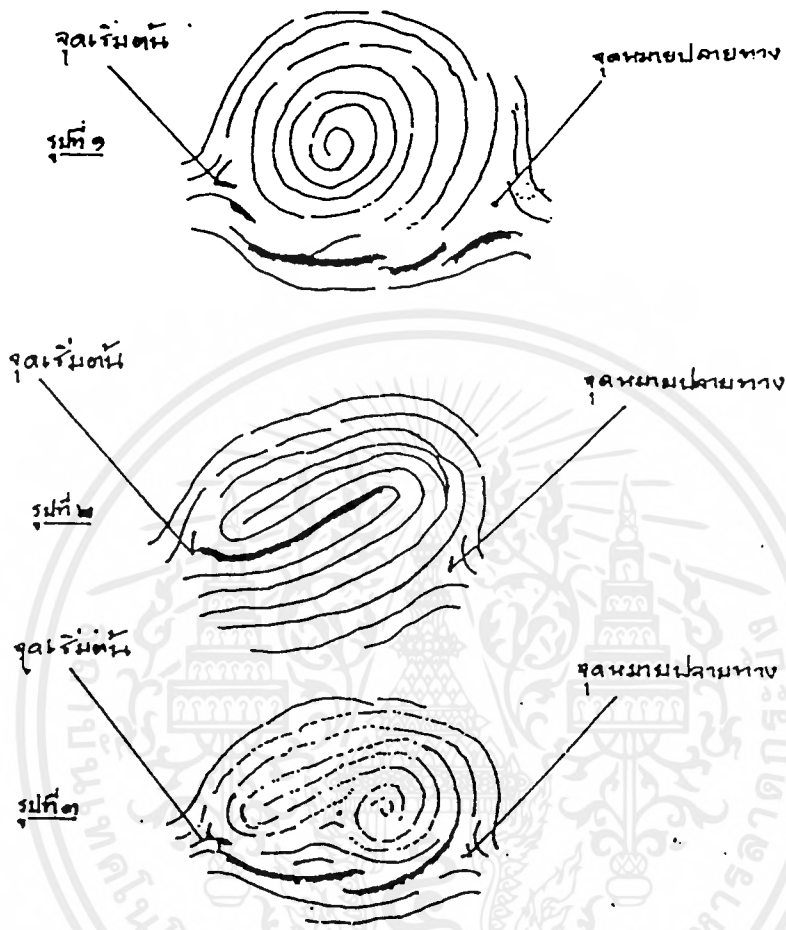
รูปที่ 21 แสดงการนับเส้นลายนิ้วมือ

2.2.7 หลักการสาวเส้นลายนิ้วมือ (ประเภทกันหอย) ได้วางหลักเกณฑ์ไว้ดังนี้

- (1) ให้สันคองซ้ายเป็นจุดเริ่มต้นและสันคองขวาเป็นจุดหมายปลายทาง
- (2) ถ้าสันคองเป็นจุดเริ่มต้น ให้สาวจากจุดแล้วสาวเส้นล่างที่ต่ำหรือถัดลงมา
- (3) ถ้าสันคองเป็นลายเส้น ให้สาวจากปลายเส้นซึ่งเป็นที่ตั้งจุดสันคองนั้น
- (4) ถ้าสันคองเป็นเส้นแตก ให้สาวจากจุดเริ่มแตก
- (5) เมื่อสาวเส้นไปแล้ว เส้นที่สาวไปขาดลง ให้สาวเส้นที่ถัดลงมาต่อไป
- (6) ถ้าเส้นที่สาวไปแล้วเป็นเส้นแตกเกิดขึ้น ให้สาวเส้นแตกเส้นล่างสุด

รูปที่ 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 22 แสดงการสาวเส้นลายนิ้วมือ

ในการสาวเส้นลายนิ้วมือกันหอยรับซ้อนที่มีสันดอน 3 สันดอน จงจำไว้ว่า ต้องสาวจากสันดอนซ้ายสุด ไปหาสันดอนขวาสุดเสมอ สันดอนกลางนั้นไม่ต้องพะวงถึง

การสาวเส้นลายนิ้วมือประเภทกันหอยทั้ง 4 ชนิด เส้นที่สาวไปแลงั้น อาจจะไปพบกับสันดอนขวามือที่เป็นจุดหมายปลายทาง หรืออาจจะเข้าไปอยู่ภายในหน้าสันดอนขวามือ หรืออาจจะออกไปอยู่นอกสันดอนขวามือก็ได้ ไม่แน่นอนแล้วแต่ลายเส้นที่สาวไปนั้น ดังนั้นจึงจำเป็นต้องมีกฎเกณฑ์เพื่อถือเป็นหลักปฏิบัติในการแยกประเภท ดังนี้

เมื่อสาวเส้นจากสันดอนซ้ายมือไปยังสันดอนขวามือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1) ให้นับจำนวนลายเส้นระหว่างเส้นที่สาวไป กับสันคองว่ามีจำนวนเท่าใด ยกเว้นเส้นที่สาวและสันคองไม่ต้องนับรวมไปด้วย

(2) ถ้าเส้นที่สาวไปแล้ว เข้าไปอยู่ข้างในหน้าสันคองขวามือนับได้ 3 เส้นหรือมากกว่า ใช้อักษรย่อว่า “ I ” (INNER)

(3) ถ้าเส้นที่สาวไปแล้วออกไปอยู่ข้างนอกสันคองขวามือนับได้ 3 เส้น หรือมากกว่า ใช้อักษรย่อว่า “ O ” (OUTER)

(4) ถ้าเส้นที่สาวไปแล้ว เข้าไปอยู่หน้าสันคองขวามือหรือออกไปอยู่นอกสันคองขวามือนับได้ 2 เส้น หรือน้อยกว่าหรือพบกับสันคองขวามือพอดี ใช้อักษรย่อว่า “ M ” (MEETING)



รูป (ก) เส้นที่สาวไปนั้นเข้าไปอยู่หน้าสันคองขวามือนับได้ 3 เส้นระหว่างเส้นที่สาวกับสันคอง ฉะนั้นลายนิ้วมือจึงต้องมีค่าเป็น “ I ” (ถ้าน้อยกว่า 3 เส้น เป็น M)



รูป (ข) ถ้าเส้นที่สาวไปนั้น ออกไปอยู่นอกสันคองขวามือนับได้ 3 เส้น ฉะนั้นลายนิ้วมือรูปนี้จึงมีค่าเป็น “ O ” (ถ้าน้อยกว่า 3 เส้น เป็น M)



รูป (ค) เส้นที่สาวไปนั้นพบกับสันดอนขวามือพอดี ฉะนั้นรูปลายนิ้วมือรูปนี้จึงมีค่าเป็น " M "
รูปที่ 23 แสดงการสาวเส้นสันดอนซ้ายไปหาสันดอนขวามือ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

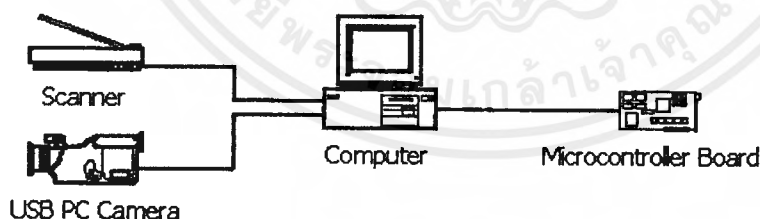
สถาปัตยกรรมของระบบรักษาความปลอดภัยบ้าน

ปริญญาานิพนธ์นี้ได้ประยุกต์งานทางด้านการประมวลผลภาพดิจิทัลกับภาพถ่ายนิ้วมือเพื่อพัฒนาอัลกอริทึม โดยวิเคราะห์โครงสร้างลายนิ้วมือที่มีการหักเหอย่างอิสระจากการถ่ายลายนิ้วมือสด ด้วยกล้องวิดีโอและซอฟต์แวร์ถ่ายภาพเข้ามาแล้วปรับปรุงภาพให้ชัดเจนขึ้นโดยใช้เทคนิควิธีการต่างๆ ทำให้ได้ภาพที่ดีขึ้น หลังจากนั้นจึงเก็บข้อมูลภาพเพื่อนำไปใช้วิเคราะห์ทดสอบต่อไป ต่อมาคำนวณลักษณะของลายนิ้วมือตามทิศทางแกน X_i ตามแกน Y_i และตามทิศทางเฉลี่ยรวม I เพื่อดูความแตกต่างและแยกแยะบุคคลออกจากกันตามลักษณะของลายนิ้วมือ แล้วสร้างฮิสโตแกรมแสดงความสัมพันธ์ของแกนความถี่กับแกนระดับเทาของภาพ และต่อมาสร้างกราฟฮิสโตแกรมทิศทางทั้งแบบ 4 ทิศทางและแบบ 8 ทิศทางเป็นการหาทิศทาง โดยหาจากภาพระดับเทา จากการพิจารณาค่าความแตกต่างของระดับสีแล้วจึงนำมาสร้างเป็นทิศทางเก็บทิศทางเป็นตัวเลขทิศทาง 0,1,2 และ 3 ลงฐานข้อมูลเพื่อเก็บไว้เปรียบเทียบต่อไป

ขั้นตอนการทำงานของระบบ

การทำงานของระบบแบ่งออก 3 ส่วนด้วยกันคือ

1. ส่วนของการรับภาพถ่ายนิ้วมือ
2. ส่วนของการประมวลผล
3. ส่วนควบคุมการแสดงผลทางเอาต์พุต



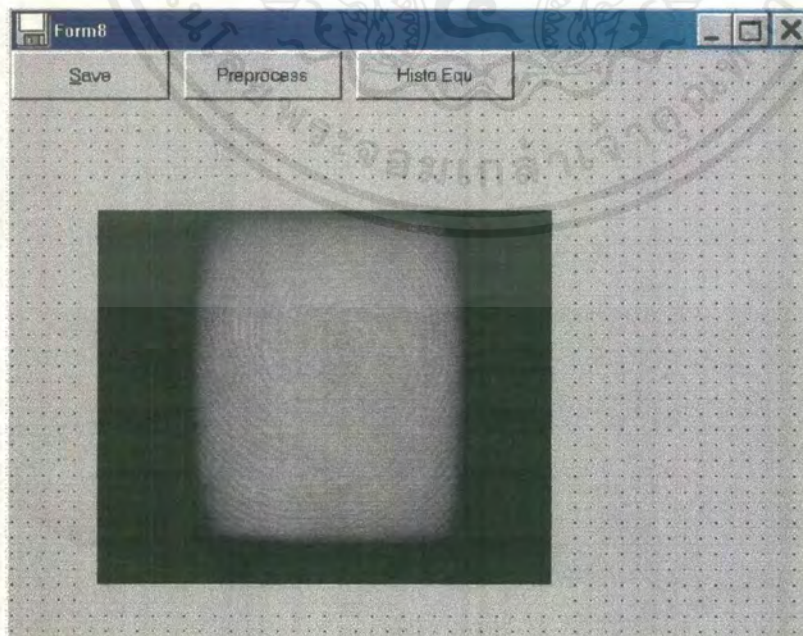
รูปที่ 3.1 แสดงการประกอบอุปกรณ์ที่ใช้ในระบบ

1. ส่วนของการรับภาพลายนิ้วมือ

จะรับภาพกล้อง Philips UBS PC Camera PCA646VC โดยรับภาพมาเป็น Gray Scale แล้วทำการบันทึกเป็นไฟล์ภาพ โดยภาพที่ได้จากกล้องจะเป็นภาพสดที่จะต้องมีการจัดแสงให้ กระทบบนนิ้วมือเท่าๆกันทุกพื้นที่ และจะต้องปรับโฟกัสของกล้องให้ได้ภาพที่ชัดที่สุด และการวางตำแหน่งของนิ้วมือจะต้องวางในตำแหน่งเดียวกันเสมอ



รูปที่ 3.2 แสดงภาพการวางตำแหน่งมือ



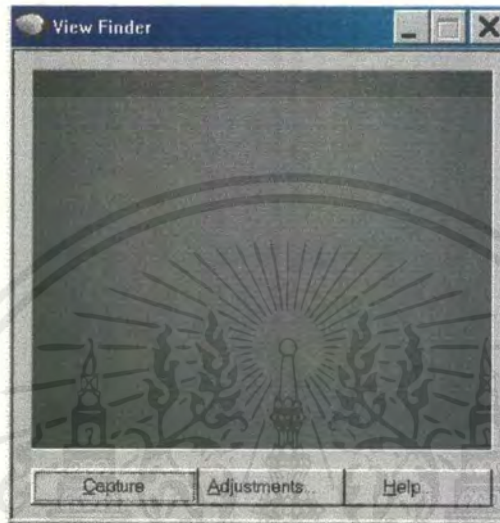
รูปที่ 3.3 แสดงการรับภาพจากกล้องนำเข้าสู่ Form การปรับปรุงภาพและบันทึกภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนของการประมวลผล

ส่วนของการประมวลผลจะเป็นการจัดด้วยตัวโปรแกรมทั้งหมดประกอบด้วย 5 ส่วนด้วยกันคือส่วน INPUT,PREPROCESS,RECOGNIZE,HISTOGRAM และ OUTPUT

1.INPUT เป็นโปรแกรมรับภาพมาจากกล้อง Philips USB PC Camera PCA646VC ตัวโปรแกรมประกอบด้วย 3 ปุ่มด้วยกันดังรูป



รูปที่ 3.4 แสดงตัว โปรแกรมที่รับภาพจากกล้อง

1.1 Capture ทำหน้าที่คล้ายปุ่ม Shutter ของกล้องถ่ายรูปที่ทำการถ่ายภาพเพื่อนำไปประมวลผลต่อไป

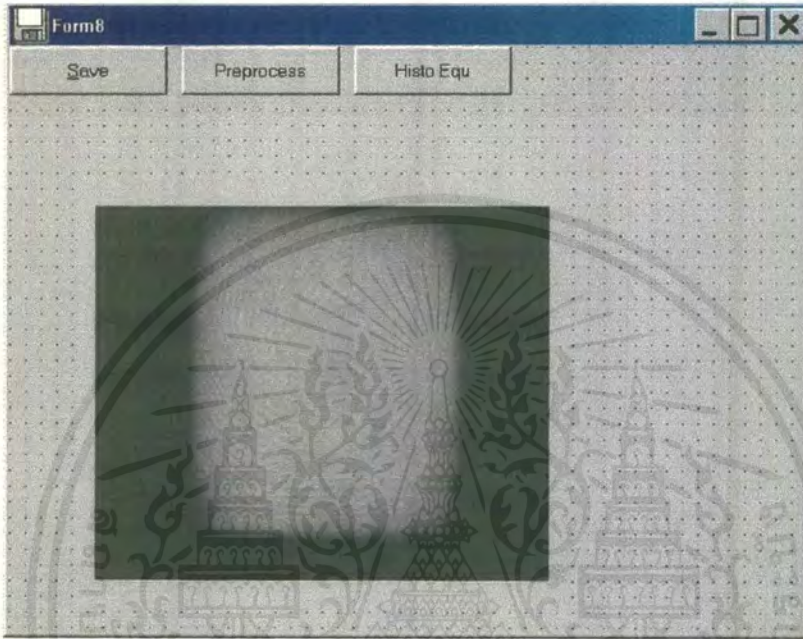
1.2 Adjustments ประกอบด้วยส่วนต่างๆ ดังนี้

- Files เป็นการกำหนดลักษณะของไฟล์ เช่นกำหนดนามสกุลของไฟล์ภาพ กำหนด directory ที่ทำการบันทึกภาพ รวมทั้งกำหนดให้ โปรแกรมทำ Auto Capture กำหนดระยะเวลาในการ Auto Capture และการกำหนดเกี่ยวกับการบันทึกเสียงแต่ไม่ถูกนำมาใช้
- White Balance/Exposure โดยปกติจะตั้งเป็น Auto ไว้
- Image Controls ทำการปรับแต่งความเข้มของ Brightness Contrast และ Gamma และปรับภาพว่าจะกำหนดให้เป็นภาพสีหรือว่าเป็น Gray Scale
- Source/Format เป็นการกำหนดแหล่งที่มาของภาพที่จะทำการบันทึก

1.3 Help เป็นส่วนอธิบายหน้าที่และวิธีการปรับแต่งค่าต่างๆของตัว โปรแกรมซึ่งจะทำการปรับเมื่อกดปุ่ม Adjustment

2. PREPROCESS

เป็นส่วนประมวลผลภาพเบื้องต้น ซึ่งจะทำการปรับปรุงภาพที่รับมาจากให้ดีขึ้น โดยจะทำการทดลองในวิธีการต่างที่ใช้ในการปรับปรุงภาพเช่น Histogram Equalize, Sharpen และ High Pass Filter เป็นต้น เพื่อที่ว่าจะทำให้การคำนวณทิศทางของลายนิ้วมือได้ดีขึ้นและถูกต้องมากขึ้น โดยจะทำด้วยวิธีการต่อไปนี้



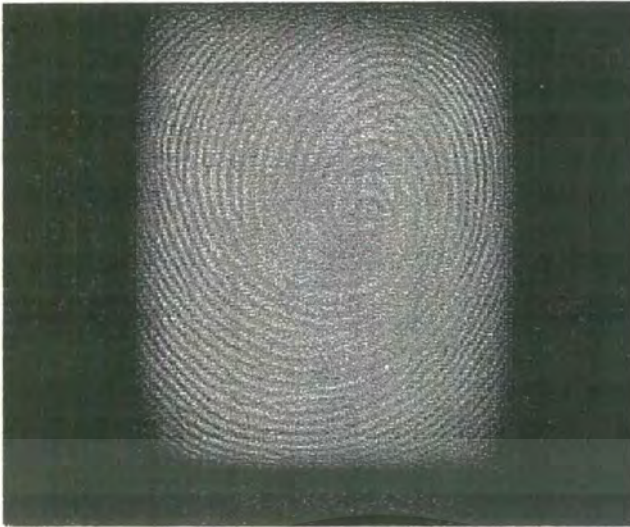
รูปที่ 3.5 แสดงการรับภาพจากกล้องนำเข้าสู่ Form การปรับปรุงภาพและบันทึกภาพ

จากรูปด้านบนจะเป็น form ที่ใช้ในการปรับปรุงภาพ 2 วิธีการด้วยกัน โดยที่ปุ่ม Preprocess จะทำการปรับปรุงภาพแบบ Sharpen ปุ่ม Histo Equ ทำการปรับปรุงภาพแบบ Histogram Equalize

การปรับปรุงภาพด้วยวิธี Sharpen

Sharpen Filter เป็นวิธีการปรับปรุงภาพที่พรมัว ซึ่งทำโดยการปรับ contrast แต่ละ pixels Sharpen filters ประกอบด้วย Sharpen Edges และ Unsharp Mask filters

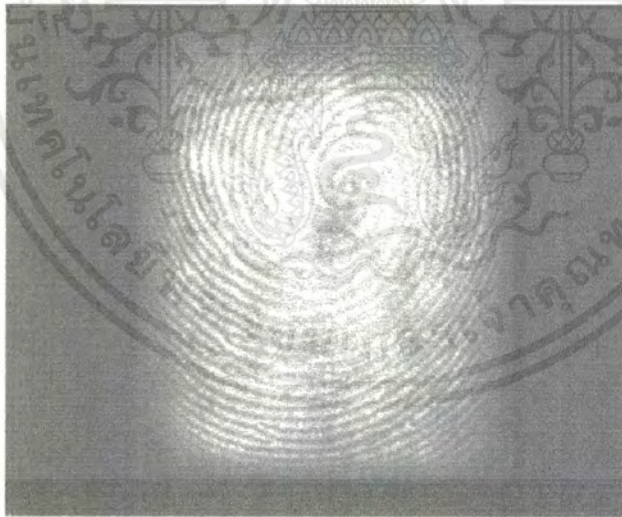
Unsharp Mask filter เป็นการ filter ภาพ โดยคิดที่ตำแหน่ง pixels นั้นๆ และ pixels รอบๆ โดยใช้ threshold โดยจะเจาะจงที่จะปรับค่า contrast ของแต่ละ pixels ซึ่งทำให้แต่ละพื้นที่ของภาพมีความชัดเจนมากขึ้น



รูปที่ 3.6 แสดงรูปที่ทำการปรับปรุงภาพแบบ Sharpen Filter

การปรับปรุงภาพด้วยวิธี Histogram Equalize

เป็นการปรับปรุงภาพ โดยการปรับค่าความสว่างของแต่ละ pixel ของภาพซึ่งจะทำการปรับปรุงภาพทั้งภาพ



รูปที่ 3.7 แสดงรูปที่ทำการปรับปรุงภาพแบบ Histogram Equalize

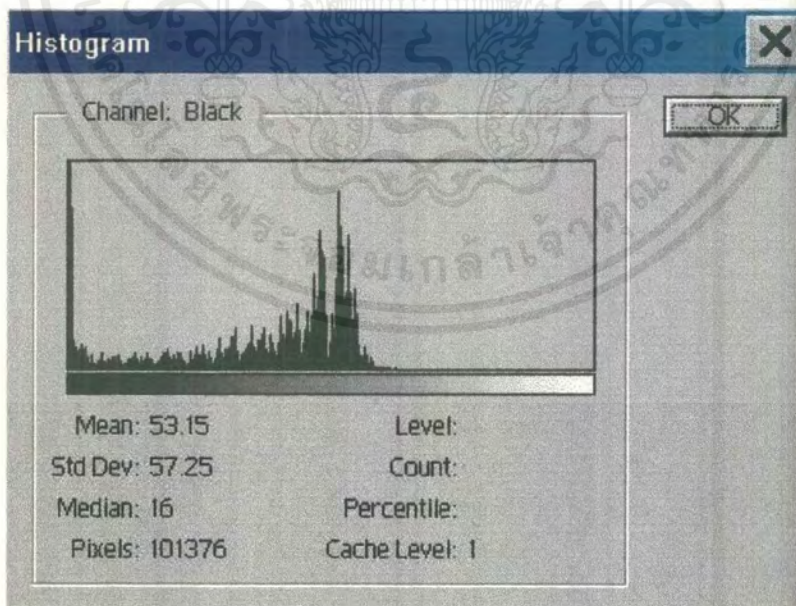
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับปรุงภาพด้วยวิธี High Pass Filter

เป็นการปรับปรุงภาพโดยใช้วิธีการกรองความถี่โดยตัดความถี่ต่ำออกยอมให้ความถี่สูงผ่าน ทำให้ภาพที่มีช่วงความถี่ลดลง ทำให้ความแตกต่างของระดับสีลดลง จึงทำให้ง่ายต่อการนำไปประมวลผลภาพเป็นฮิสโตแกรมทิศทาง

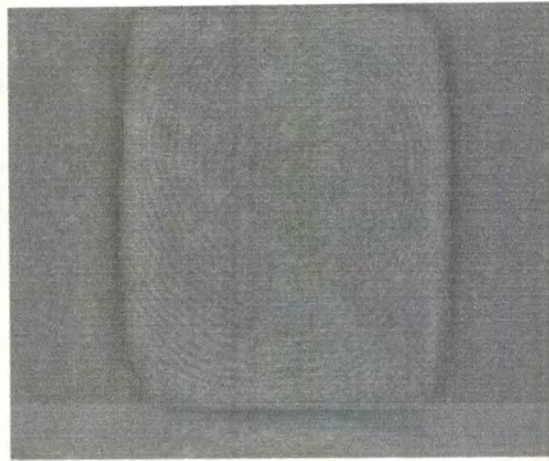


รูปที่ 3.8 แสดงรูปที่ยังไม่ได้ผ่าน High Pass Filter

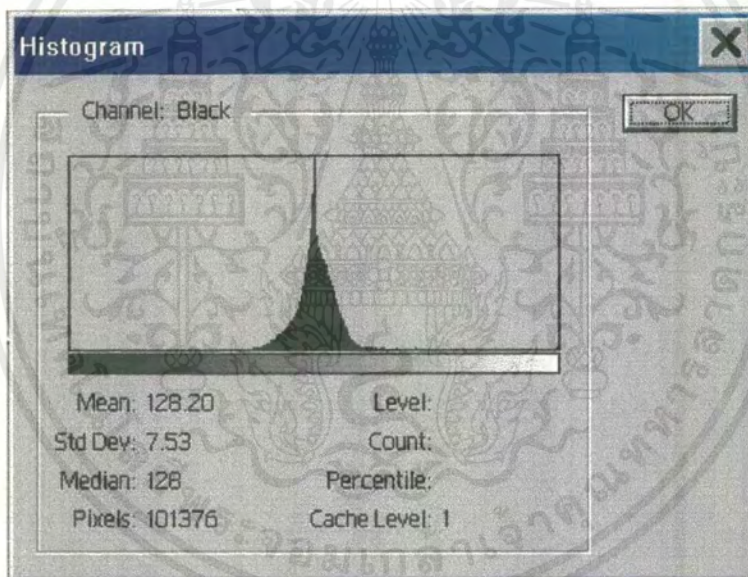


รูปที่ 3.9 แสดงรูป Histogram ของภาพลายนิ้วมือที่ยังไม่ได้ผ่าน High Pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงรูปที่ผ่าน High Pass Filter



รูปที่ 3.11 แสดงรูป Histogram ของรูปลายนิ้วมือที่ผ่าน High Pass Filter

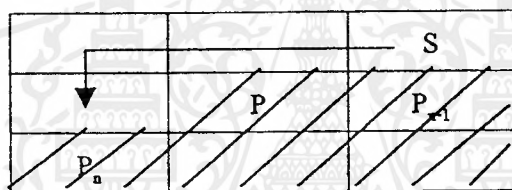
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. RECOGNIZE

3.1 Direct Avg. เป็นการสร้างกราฟทิศทางเฉลี่ยตามลายนิ้วมือ โดยหาข้อมูลตามแนวแกน x หรือสแกนข้อมูลทิศทางเฉลี่ยตามแนวราบ (x_i) ได้หนึ่งภาพ และหาข้อมูลตามแนวแกน y หรือสแกนข้อมูลทิศทางเฉลี่ยตามแนวตั้ง (y_i) ได้หนึ่งภาพ และนำเอามารวมกันแล้วหาร 2 เป็นทิศทางเฉลี่ยรวม i ซึ่งวิธีการนี้ได้นำเอาข้อมูลภาพที่เสร็จสิ้นการประมวลผลจากการทำให้เส้นบาง (thinning) เรียบร้อยแล้ว ได้พิจารณา ดังนี้

อ่านทิศทางแต่ละแนวตามแกน x_i และแกน y_i โดยใช้วิธีคิดตามขอบภาพที่เป็น "1" เนื่องจากได้ประมวลผลภาพแล้วนำเอาช่องหน้าต่างขนาด 3×3 จุดภาพของภาพที่ 4.1 ครอบอยู่ไปยังจุดที่ต้องการแล้วติดตามโดยสแกนทวนเข็มนาฬิกาเพื่อหาขอบภาพ แล้วเปลี่ยนตำแหน่งพิจารณาไปที่จุดนั้นๆ จุดภาพที่ 4.2 และกำหนดค่าต่างๆ ภายในช่องหน้าต่าง รูปที่ 4.3

รูปที่ 3.12



แสดงการติดตามหาจุดภาพตามลายเส้น

- เมื่อ P_{n-1} เป็น ตำแหน่งของภาพที่ผ่านมา
 P เป็น ตำแหน่งของจุดเริ่มต้นการตรวจเช็คทิศทาง
 P_n เป็น ตำแหน่งของขอบภาพต่อไป
 S เป็น ตำแหน่งทิศทางที่พิจารณาให้จุดภาพเป็น 1

รูปที่ 3.13

-1	-1	-1
0	1	0
+1	+1	+1

(ก)

-1	0	+1
-1	1	+1
-1	0	+1

(ข)

แสดงการแทนจุดภาพของช่องหน้าหน้าค่างที่สแกนไปตามแกน x_i และแกน y_j

สรุปผลของแนวทางและหลักการในการหาทิศทาง

ก. แนวทางที่กล่าวมาสามารถเขียนเป็นสมการได้ คือ สมการที่(1'),สมการที่ (2'),และสมการที่(3')

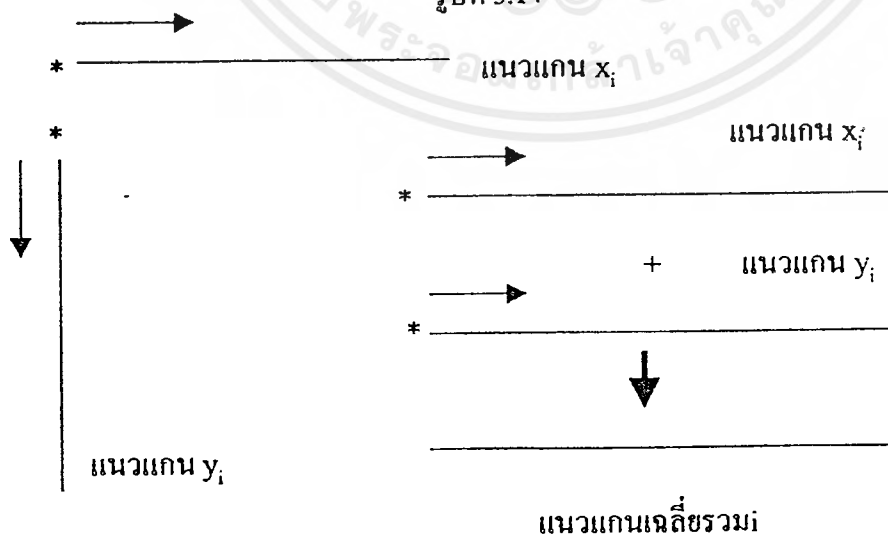
ทิศทางเฉลี่ยตามแกน $x_i =$ ผลรวมค่าทิศทางผลลัพธ์ตาม y เมื่อ p วางบนจุด '1' จุด Q' (1')
จำนวนจุดที่เป็น '1' ตามแนวแกนแกน y ที่คำนวณทิศทาง

ทิศทางเฉลี่ยตามแกน $y_j =$ ผลรวมค่าทิศทางผลลัพธ์ตาม x เมื่อ p วางบนจุด '1' จุด Q' (2')
จำนวนจุดที่เป็น '1' ตามแนวแกนแกน x ที่คำนวณทิศทาง

ทิศทางเฉลี่ยรวม $i =$ ทิศทางเฉลี่ยตามแกน $x_i +$ ทิศทางเฉลี่ยตามแกน y_j (3')
2

ข. การวางทิศทางเฉลี่ยของแนวแกน

รูปที่ 3.14



แสดงจุดภาพ, แนวแกน x_i , แกน y_j และแกนเฉลี่ยรวม i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักเรียนเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ

1. Q' คือ จำนวนเลข(1,2,3...)ที่ถูกรับเข้าไปเพื่อให้กราฟสูงจากแนวแกน
2. วิธีการนี้มีประโยชน์ในการตรวจดูลักษณะของกราฟที่เกิดขึ้นแต่ละรูป และแต่ละกลุ่มแบบทำให้ทราบแตกต่างกันออกไป จะสามารถคาดได้ว่าเป็นแบบใด
3. * หมายถึง ตำแหน่งแรกที่เป็นจุดเริ่มต้นของแต่ละแนวแกน
4. ถ้าตลอดทั้งทิศทางแนวแกน x_i และแกน y_i ไม่พบจุดที่เป็น 1 เลขเมื่อครอบหน้าต่างลงไปวางจึงไม่วางได้ให้ถือว่าที่ตรงตำแหน่งนั้นเท่านั้นเท่ากับ 0 และที่จุดภาพแรกและจุดภาพสุดท้ายตลอดแกน ไม่มีเพราะช่องหน้าต่างไม่สามารถวางขอบออกไปได้

3.2 User Define เป็นการวางและหาจุดสำคัญต่างๆ บนลายนิ้วมือ โดยให้ผู้ใช้มองจากภาพและใช้เมาส์กำหนดเอง เช่น การกำหนดจุดใจกลาง(core), ต้นคอน(delta), การนับจำนวนเส้นสมมติที่ตัดเส้นดำของลายนิ้วมือและการกำหนดจุดสำคัญต่างๆ อาทิ เช่น เส้นขาด, ทะเลสาบ, จุดหรือเกาะ เป็นต้น

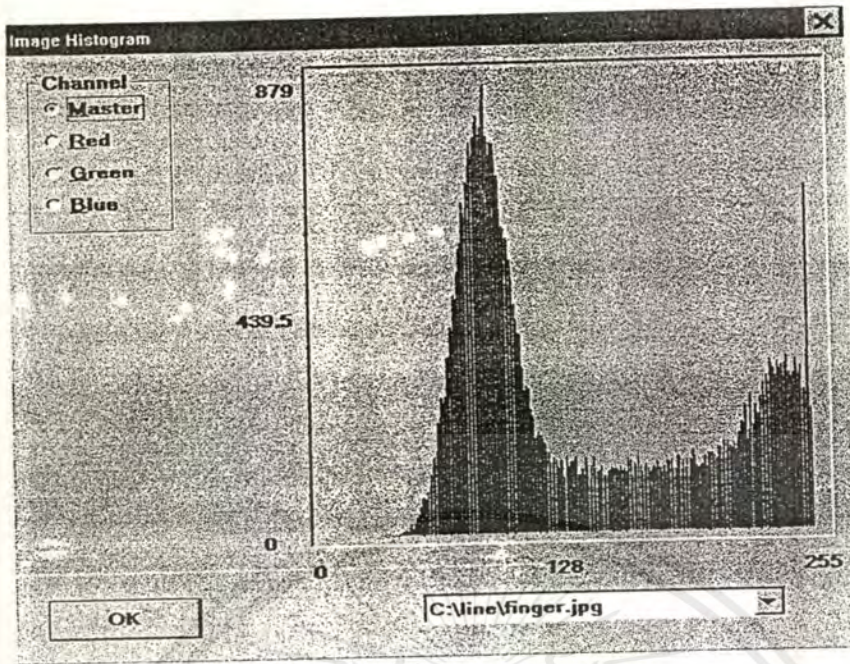
4. HISTOGRAM

เป็นการแสดงแผนผังฮิสโตแกรม โดยวาดกราฟแสดงค่าความถี่ของภาพระดับเทาเมื่อรับเทาเมื่อรับภาพเข้ามา โดยแสดงจำนวนจุดภาพของความถี่สูงสุด (f_{max})

รูปที่ 3.15



รูปตัวอย่างลายนิ้วมือ

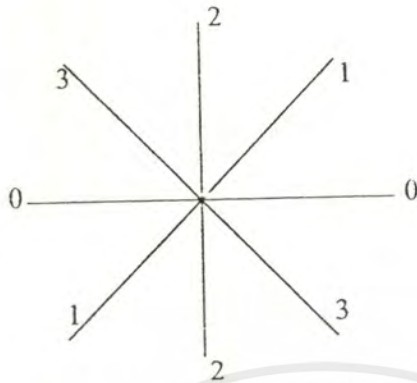


รูปที่ 3.16 รูปแสดงกราฟฮิสโตแกรมของลายนิ้วมือข้างบน
การใช้วิธีการฮิสโตแกรมทิศทางแทนฮิสโตแกรมเดิม

เป็นการเปลี่ยนแปลงจากฮิสโตแกรมมาใช้แนวทางของฮิสโตแกรมทิศทาง (Directional Histogram) ซึ่งเป็นวิธีการค้นหาทิศทางเดินของจุด โดยคำนวณทิศทางตามโครงสร้างของลายเส้นของลายนิ้วมือจะถูกอ่านเข้ามาเป็นระดับเทา แล้วเขียนเส้นตามทิศทาง จะพิจารณาระดับเทา จากตำแหน่งในแต่ละจุดภาพ ซึ่งตำแหน่งใดของจุดมีความแตกต่างระดับเทาน้อยที่สุดของภาพจะบันทึกไว้ พร้อมกับการคำนวณทิศทางที่เรากำหนดเส้นทางเป็น 4 ทิศทางดังนี้

ถ้าให้ D เป็นทิศทางของจุดภาพ ระดับสีเทาในภาพ G (ขนาด $g \times g$) และลายนิ้วมือประกอบไปด้วยทางเดินของจุดที่ทำให้เกิดเป็นสายเส้นด้วย แต่ละจุดจะมีทิศทาง $D(i,j)$ ที่จุด (i,j) โดยให้ d เป็นตัวเลขบอกทิศทางที่เลือกใช้ และ N เป็นจำนวนของที่ใช้ เมื่อให้ N มีค่าเท่ากับ 4 ดังนั้นทิศทางจะมีได้เพียง 4 ทิศทาง

รูปที่ 3.17



แสดงทิศทางที่เป็นไปได้ 4 ทิศทาง

ให้ M เป็นจำนวนภาพ ที่ถูกใช้ในการคำนวณ

S_d คือผลรวมความถี่ของความแตกต่างระดับเทาตามทิศทางของ d โดยการตรวจเทียบกันระหว่างจุด ถ้าจุดที่อยู่ใกล้เคียงกันจะมีระดับความเข้มที่เหมือนกันหรือใกล้เคียงกัน แสดงได้ว่ามีทิศทางไปทางเดียวกัน

ดังนั้น S_d ในทิศทางใดมีค่าน้อยจะแสดงทางเดินของจุดลายเส้นหรือร่องในทิศทาง d นั้น ค่า S_d ทั้งหมด จึงเปรียบเทียบเหมือนเป็นฮิสโตแกรมที่แสดงความเป็นไปได้ของทิศทางต่างๆ

$$S_d = \sum_{m=0}^{m-1} |G(i_m, j_m) - G(i, j)| \quad \text{สำหรับ } d=1,2,\dots,N \quad \dots(4')$$

กำหนดจุดที่ $G(i,j)$ เป็นจุดตรวจเทียบกับจุด $G(i_m, j_m)$ ซึ่งเป็นจุดที่พิจารณาโดยข้อมูลภาพ แสดงลายเส้นระดับสีเทา ถ้าผลงานแตกต่างระหว่างระดับสีเทาของจุด $G(i,j)$ กับจุด $G(i_m, j_m)$ มีความแตกต่างกันน้อยมากหรือใกล้เคียงกับทางเดินของจุดนั้นจะแสดงทิศทางของลายเส้น ดังนั้น จึงต้องมีการคำนวณหา S_0 ถึง S_3 เพื่อจะเลือกลายเส้นว่าควรเป็นทิศทางใดของค่า d แล้วนำมาพล็อตลงเป็นรูปทิศทาง อีกรูปหนึ่งเราจะทำการทดสอบ M จุด ในแต่ละทิศทางของ d ในที่นี้เราให้ค่าของ M มีค่าเท่ากับ 16 โดยคิดรวมกับจุด $G(i,j)$ ด้วยจำนวน N ของทิศทางและมีจำนวนของจุด M จะถูกพิจารณาคำนวณทิศทางที่มีความสัมพันธ์กับค่าส่วนใหญ่ สำหรับ N ควรมีความสัมพันธ์กับ M เพราะเป็นค่าที่จะใช้คำนวณการแทนทิศทางในการทดลองได้ เลือก $N=4$ พบว่ามีความเหมาะสมคือ

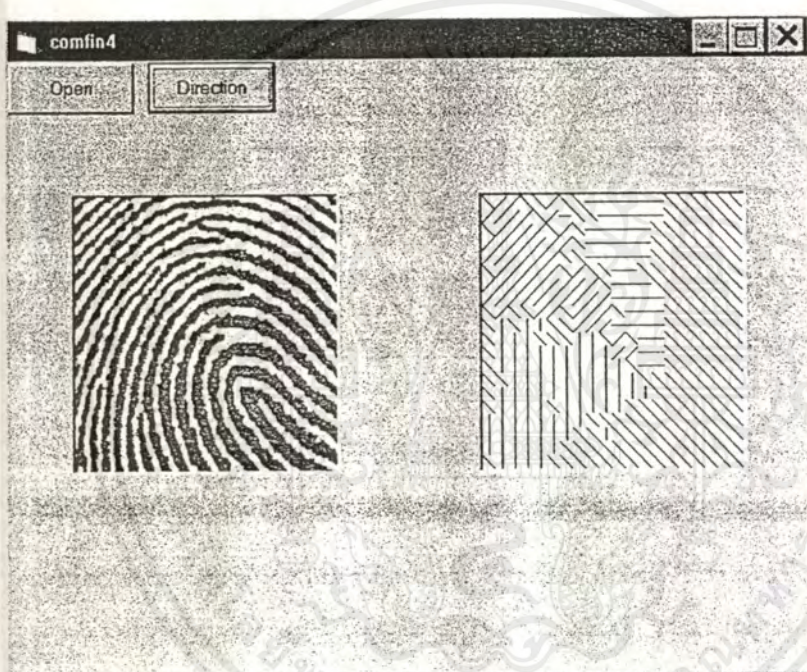
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเลือกค่าที่สูงกว่า 4 พบว่าจะต้องคำนวณเพิ่มขึ้นมากในรอบผลลัพธ์ที่ได้ยังมีลักษณะใกล้เคียงกับ $N=4$ โดยได้พิจารณาและเป็นผลดังนี้

1. วิธีการแปลงข้อมูลภาพเป็นข้อมูลทิศทาง 4 ทิศทาง

เมื่อถ่ายข้อมูลเข้ามาแล้วเก็บบันทึกเป็นไฟล์ที่มีนามสกุลภาพ แล้วนำไฟล์ภาพไปแสดงบน Object ที่ชื่อ Picture Box (Object ของ Visual Basic 6) แล้วทำการประมวลผล โดยคำนวณออกมาเป็น 4 ทิศทางคือ 1,2,3 และ 4 ในกรณีที่ต้องการประมวลผลโดยใช้ทิศทางเพียง 4 ทิศทางเท่านั้น และ 8 ทิศทางกรณีประมวลผลภาพแบบ 8 ทิศทาง ดังแสดงในรูป

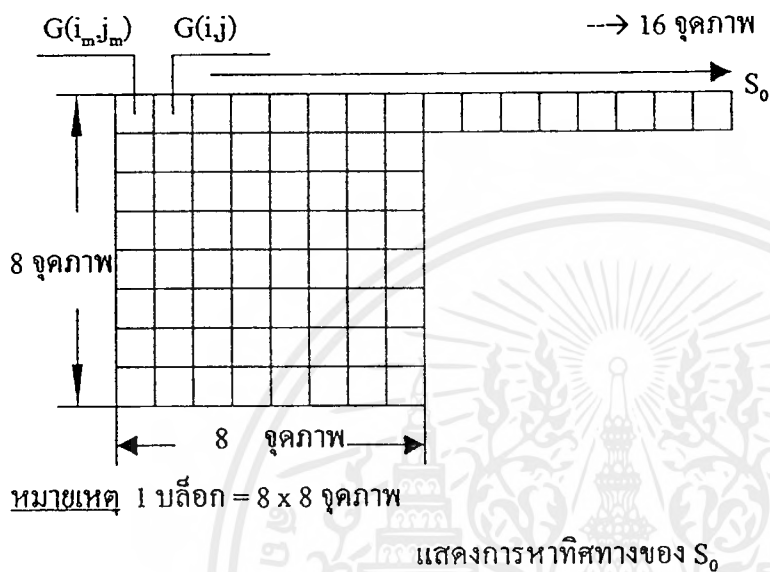
รูปที่ 3.18



แสดงการหาทิศทางของโปรแกรมหาทิศทางลายนิ้วมือ

2. ตัวอย่างการพิจารณาทิศทางขนาด 8×8 จุดภาพ เป็นหนึ่งบล็อก

รูปที่ 3.19

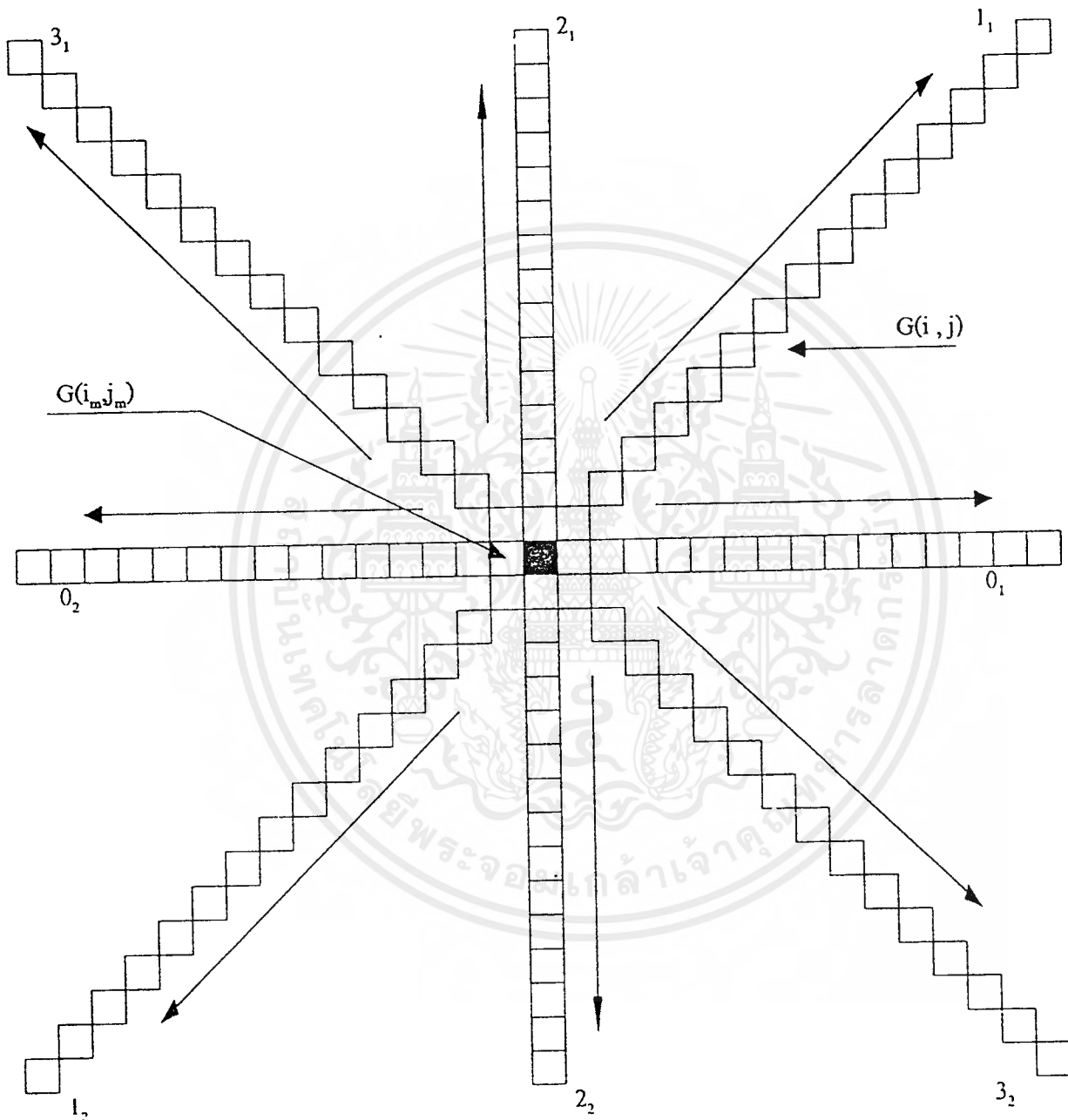


เนื่องจากลายนิ้วมือมีจุดสำคัญต่างๆ เช่น จุด core ,จุด delta, จุดขาด หรือ เส้นสั้น (Short line) เป็นต้น ซึ่งจะมีผลต่อการประมวลผลทิศทางทำให้เกิดการผิดพลาดขึ้น และระยะทางในการพิจารณาทิศทางของจุดแต่ละจุดหากกำหนดไม่เหมาะสมก็จะทำให้ได้ทิศทางที่ผิดพลาดได้

โดยที่เราจะไม่พิจารณาจุดสำคัญเนื่องจากเกิดการผิดพลาดได้ง่ายและเสียเวลาคำนวณเพิ่มขึ้น จึงใช้วิธีการหาทิศทางโดยเอาจุดภาพ $G(i_m, j_m)$ ลบ จุดภาพ $G(i, j)$ (เอาค่าสีที่อ่านได้ ณ.จุด $G(i, j)$ ลบด้วยค่าสีของจุดถัดไป $G(i_m, j_m)$) โดยไม่พิจารณาเครื่องหมาย)

3. การพิจารณาหาทิศทางโดยรอบ $G(i_m, j_m)$ 4 ทิศทาง

รูปที่ 3.20



แสดงการหาทิศทางโดยรอบทั้ง 4 ทิศทาง

พิจารณาทิศทางการคำนวณออกไป n จุดภาพ จาก $G(i_m, j_m)$ โดยทิศทางนั้นจะถูกกำหนดเอาไว้ 4 ทิศทาง ตามภาพที่ แสดงจุดทิศทาง การกำหนดทิศทางควรจะอยู่ทางทิศทางใดจะทำการตรวจเทียบตำแหน่งของจุดทุกๆ จุดในแต่ละจุดภาพของทิศทาง 0, 1, 2 และ 3 โดยดูจากผลของการลบกักระหว่างจุดในระดับสี่เทาของภาพ ถ้าผลต่างกันระหว่างจุดมีความแตกต่างกันระหว่างจุดมีความไม่เท่ากันใด ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกต่างกันน้อยที่สุดก็แสดงว่าทิศทางควรจะไปอยู่ในทิศทางใดใน 1 ถึง 4 ทิศทาง ต่อจากนั้นก็ให้
ทำในแต่ละระดับจอจนครบทั้งภาพ ก็จะได้รูปทิศทางของลายนิ้วมือ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.OUTPUT จะทำการบันทึกทิศทางของภาพลายนิ้วมือลงใน Database ในส่วนของการบันทึก และการดึงข้อมูลจาก Database มาเปรียบเทียบแล้วแสดงผลการเปรียบเทียบ โดยข้อมูลที่บันทึกจะบันทึกพื้นที่ , ทิศทางส่วนใหญ่ และจำนวนทิศทางส่วนใหญ่ดังรูปการบันทึก

Add	Update	Delete	Find	Refresh	Close
Field Name:	Value:				
area:	1				
direc:	1				
maxdirec:	67				

รูปที่ 3.21 แสดงตารางการบันทึกข้อมูลใน database

เมื่อผลการเปรียบเทียบออกมาไม่ว่าจะเป็นเจ้าของลายนิ้วมือหรือไม่ก็จะแสดงผลของการเปรียบเทียบดังนี้

หากผลการเปรียบเทียบเป็นเจ้าของลายนิ้วมือคนเดียวกันจะแสดง Form ดังรูป

link

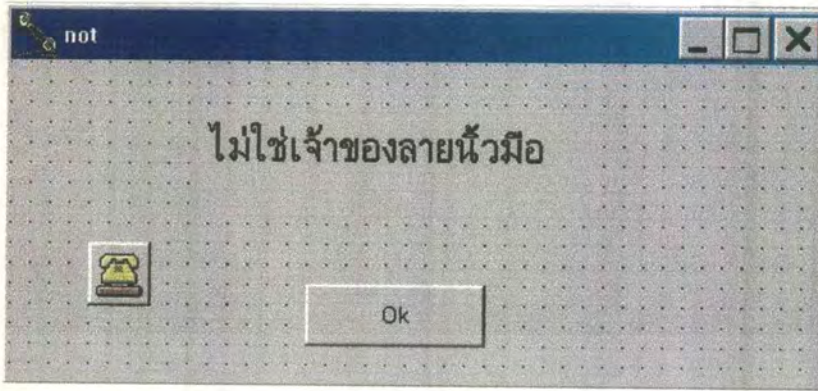
fingerprint project
เป็นเจ้าของลายนิ้วมือ

OK

รูปที่ 3.22 แสดงรูป Form แสดงผลว่าเป็นเจ้าของลายนิ้วมือคนเดียวกัน

โดย Form ข้างบนจะมี Control ชื่อ MSComm (รูปโทรศัพท์) ทำหน้าที่เป็นเชื่อมโยงข้อมูลสู่ Port RS232 ส่งข้อมูลไปยัง Microcontroller Board เพื่อควบคุมอุปกรณ์ภายนอกต่อไป

เมื่อผลการเปรียบเทียบออกมาไม่ใช่เจ้าของลายนิ้วมือก็จะแสดงผลโดยแสดง Form ดังรูป

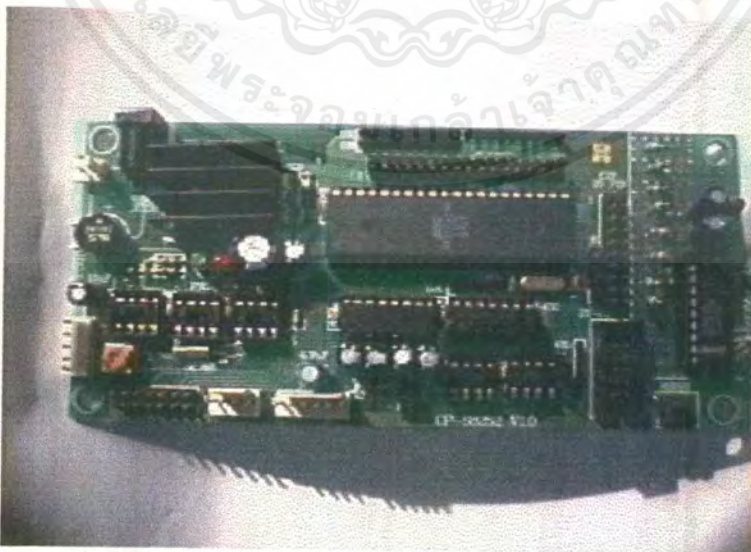


รูปที่ 3.23 แสดง Form เมื่อผลการเปรียบเทียบออกมาไม่ใช่เจ้าของลายนิ้วมือ

โดย Form ข้างบนจะมี Control ชื่อ MSComm (รูปโทรศัพท์) เช่นเดียวกับรูปที่ 3.22 ทำหน้าที่เป็นเชื่อมต่อข้อมูลสู่ Port RS232 ส่งข้อมูลไปยัง Microcontroller Board เพื่อปฏิบัติการควบคุมอุปกรณ์ภายนอก

3. ส่วนควบคุมการแสดงผลทางเอาต์พุต

เมื่อ Microcontroller Board (ใช้ MCS-51 #AT89S53 Microcontroller) รับข้อมูลจาก Serial Port RS232 โปรแกรมจะตรวจสอบว่าเป็นข้อมูลจาก Form ไหน แล้วแสดงผลทาง LED ว่าเป็นข้อมูลจากไหน ไซ้เจ้าของลายนิ้วมือที่ถูกต้องหรือไม่



รูปที่ 3.24 แสดง Microcontroller Board CP-S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างฮิสโตแกรมทิศทาง

โดยจะพิจารณาเฉพาะฮิสโตแกรมทิศทางเท่านั้น โดยไม่นำการหา f_{\max} ของความถี่ระดับเทามาใช้ โดยนำค่าทิศทาง 0,1,2 และ 3 มาเก็บเป็นคะแนนเพื่อที่จะบันทึกลงในฐานข้อมูล เพื่อนำมาเปรียบเทียบพิสัยส่วนตัวเจ้าของลายนิ้วมือต่อไป ผลการเปรียบเทียบจะนำไปควบคุมการปิดเปิดอุปกรณ์ต่อไป

สำหรับการเปรียบเทียบจะพิจารณาพื้นที่บนลายนิ้วมือทั้งหมด 9 พื้นที่ด้วยกัน เพื่อความละเอียดขึ้น เพราะบางจุดหรือบางตำแหน่งอาจมีความผิดพลาดได้ ดังนั้นจึงมีการเหลื่อมของพื้นที่หลายๆ จะมีผลดีมากขึ้น

การเตรียมข้อมูลเพื่อจะนำไปบันทึกลงฐานข้อมูล

โดยการนำข้อมูลภาพที่ได้จากการถ่ายลายนิ้วมือจริงจากกล้องดิจิตอลเข้ามา แล้วนำมาหาทิศทาง เพื่อให้การประมวลผลเป็นไปอย่างรวดเร็ว ได้ผลลัพธ์ออกมาทันที ดังนั้นจึงตัดบางขั้นตอนออกไปเช่น การกำจัด Noise การทำเส้นบาง เป็นต้น โดยจะใช้เฉพาะการหาทิศทางหรือฮิสโตแกรมทิศทางเท่านั้น แล้วนำไปบันทึกหรือเปรียบเทียบกับฐานข้อมูลต่อไป

1. การแปลงข้อมูลภาพลายนิ้วมือ

เป็นการแปลงฮิสโตแกรมทิศทางจากจุดภาพให้เป็นบล็อก โดยมีวิธีการต่อไปนี้

1.1 การพิจารณาข้อมูลภาพออกเป็นทิศทาง

เพื่อให้ง่ายขึ้นจะเปลี่ยนการคำนวณจากจุดภาพ(pixel) ไปเป็นบล็อกแทนมีผลทำให้ง่ายขึ้น ซึ่งเป็นไปตามหลักการหาทิศทางของแต่ละบล็อก($b_x \cdot b_y$) แทนการหาทิศทางของแต่ละจุดเพื่อลดการประมวลผลให้น้อยลงจากเป็น pixel $g \cdot g$ เหลือเป็น $[(g/b_x) \cdot (g/b_y)]$ รอบการกำหนดขนาดของบล็อก เพราะถ้าขนาดของบล็อกใหญ่เกินไปจะทำให้เกิดความไม่ต่อเนื่อง ทั้งนี้ขึ้นอยู่กับลักษณะภาพขอบเขตนั้นๆ คืออาจทำงานไม่ต่อเนื่องของทิศทางและถ้าบล็อกเล็กเกินไปจะทำให้ทิศทางของแต่ละจุดไม่สม่ำเสมอไม่มีทิศทางที่เด่นชัดของแต่ละบล็อก ถ้าบล็อกจำนวน N ทิศทางและจำนวนจุด M จุด จะพิจารณาสำหรับการคำนวณของทิศทางที่มีความสัมพันธ์กับค่า N ส่วนใหญ่ที่มีคือ M เพราะเป็นค่าที่สูงกว่า 4 พบว่าจะต้องคำนวณเพิ่มมากขึ้นในแต่ละรอบแต่ผลที่ได้ยังมีลักษณะใกล้เคียงกับ $N=4$ (ทิศทาง 0,1,2,3) หรือ ได้ถูกกำหนดไว้ 8 ทิศทางย่อย $(0_1, 0_2, 1_1, 1_2, 2_1, 2_2, 3_1, 3_2)$

นอกจากการพิจารณา $N=4$ แล้วยังพิจารณา $N=8$ เพื่อเปรียบเทียบกันว่าแบบไหนให้ผลดีกว่ากันแบบไหนมีการผิดพลาดมากกว่ากัน

1.2 การหาทิศทางรวมมากที่สุดของ 4 ทิศทาง

และเมื่อพิจารณาทิศทางของสมการ S_d จากบทที่ 4 คือ

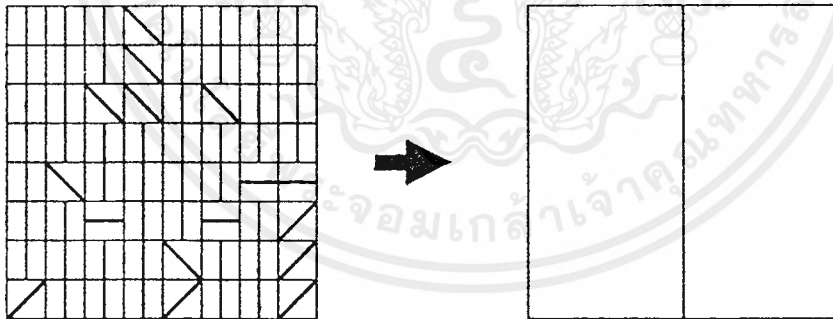
$$S_d = \sum_{m=0}^{m-1} |G(i_m, j_m) - G(i, j)| \text{ สำหรับ } \dots d = 1, 2, \dots, N$$

แทนตัวเลขทิศทางแนวราบด้านขวามือจะได้ 0_1 ดังนี้ (เมื่อกำหนดให้ $m=16$)

$$0_1 = \sum_{n=1}^{15} |G_0 - G_n| = |G_0 - G_1| + |G_0 - G_2| + \dots + |G_0 - G_{15}|$$

เมื่อได้ทิศทาง 0_1 และค่าทิศทางอื่นๆ $1_1, 1_2, 2_2, 3_1$ และ 3_2 ก็จะได้ค่าออกมาเช่นกันต่อมานำเอาค่าทิศทางมารวมกันให้พิจารณาค่าที่น้อยที่สุด เพราะในทิศทางเดียวกันเมื่อนำเอาค่ามาลบกันจะเข้าใกล้ศูนย์ หรือได้ค่าเท่ากับศูนย์ และเมื่อนำมารวมกันคือ $(0_1, +0_2)$, $(1_1, +1_2)$, $(2_1, +2_2)$ และ $(3_1, +3_2)$ ก็ยังมีค่าเข้าใกล้ศูนย์ แต่ถ้าเป็นทิศทางอื่นจะมีค่าเป็นตัวเลขที่ไม่ใช่ศูนย์หรือไม่เข้าใกล้ศูนย์ สรุปได้ว่าทิศทางเป็นเส้นตรงแทนลายนิ้วมือใน 1 จุดภาพ \rightarrow 1 บล็อก แล้วคิดทิศทางทั้งหมดเป็นทิศนั้น โดยไม่สนใจทิศทางอื่น

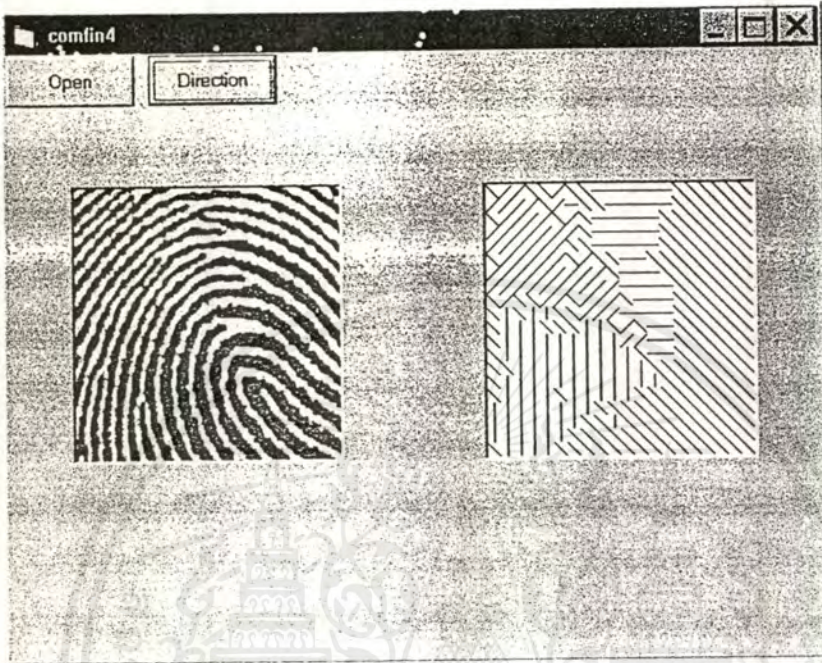
รูปที่ 3.25



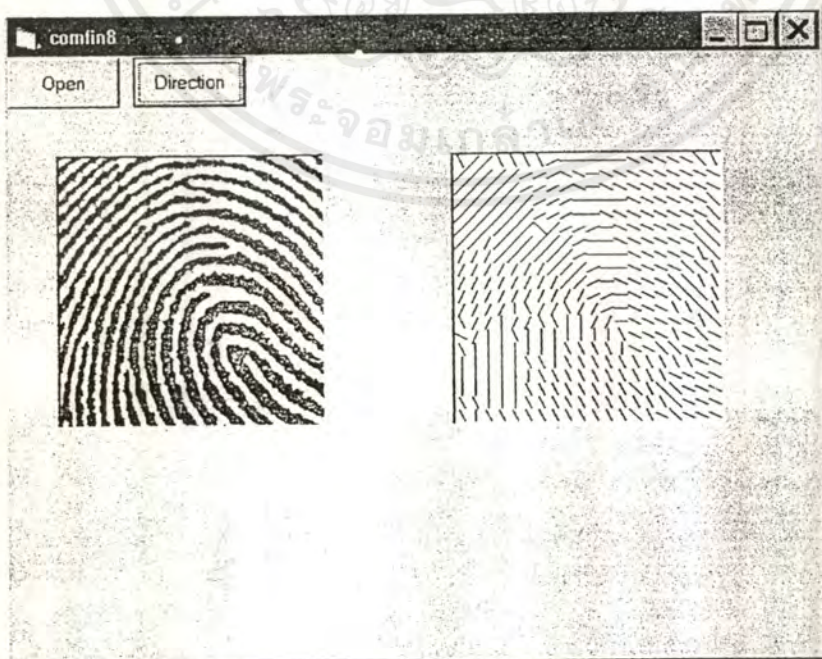
แสดงทิศทางรวมมากที่สุดเมื่อแทนทิศทางเดียวกัน ต่อ หนึ่งบล็อก

1.3 ขอบเขตการพิจารณาหาทิศทางรอบจุด G_0 ทั้ง 4 ทิศทาง

รูปที่ 3.26 แสดง โปรแกรมหาทิศทาง 4 ทิศทาง



รูปที่ 3.27 แสดง โปรแกรมหาทิศทาง 8 ทิศทาง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

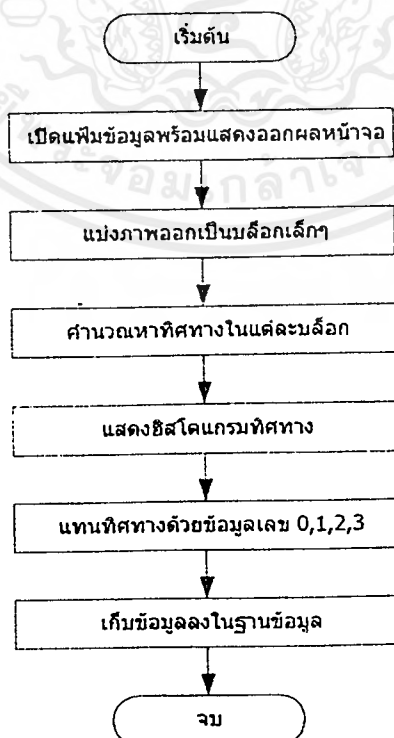
ภาพที่ 5.2 เป็นผลของ 4 ทิศทางหรือ 8 ทิศทางย่อย คือ $0_1, 1_1, 2_1, 3_1, 0_2, 1_2, 2_2$ และ 3_2 โดยบล็อก อยู่บนบนซ้ายสุดวิ่งคำนวณ ไปทางมุมบนขวามือสุดจนถึงสิ้นสุดหนึ่งแพทเทิร์น โดยพิจารณาเฉพาะจุด ภาพที่อยู่ภายในและทิศทางไม่วิ่งออกนอกขอบภาพจึงคิดนับให้แล้วเสร็จ 1 บล็อก ตั้งแต่ตัวเอง $(G_0) \rightarrow (G_{15})$

ส่วนภาพที่ 5. เป็นผลของ 8 ทิศทางหรือ 16 ทิศทางย่อย คือ $0_1, 1_1, 2_1, 3_1, 4_1, 5_1, 6_1, 7_1, 0_2, 1_2, 2_2, 3_2, 4_2, 5_2, 6_2$ และ 7_2 โดยบล็อกอยู่บนบนซ้ายสุดวิ่งคำนวณ ไปทางมุมบนขวามือสุดจนถึงสิ้นสุดหนึ่งแพทเทิร์น โดยพิจารณาเฉพาะจุดภาพที่อยู่ภายในและทิศทางไม่วิ่งออกนอกขอบภาพจึงคิดนับให้แล้วเสร็จ 1 บล็อก ตั้งแต่ตัวเอง $(G_0) \rightarrow (G_{15})$

1.4 สรุปลำดับขั้นตอนการเตรียมข้อมูลภาพ

นำเอาข้อมูลภาพที่เก็บไว้มาเตรียมเป็นข้อมูลฮิสโตแกรมทิศทาง โดยเอาภาพมาแบ่งออกเป็นบล็อกเล็กๆ แล้วคำนวณสร้างฮิสโตแกรมทิศทางโดยใช้สูตร S_i จากนั้นแทนทิศทางด้วยข้อมูลเลข $0, 1, 2, 3$ ทั้ง 4 ทิศทางในกรณีหาทิศทาง 4 ทิศทางและข้อมูลเลข $0, 1, 2, 3, 4, 5, 6, 7$ และ 7 ในกรณีหาทิศทาง 8 ทิศทาง ที่อธิบายและแสดงไว้ในบทที่ 4 และเก็บข้อมูลลงแฟ้มข้อมูลไว้เปรียบเทียบ

รูปที่ 3.28 แสดงขั้นตอนการเตรียมข้อมูล

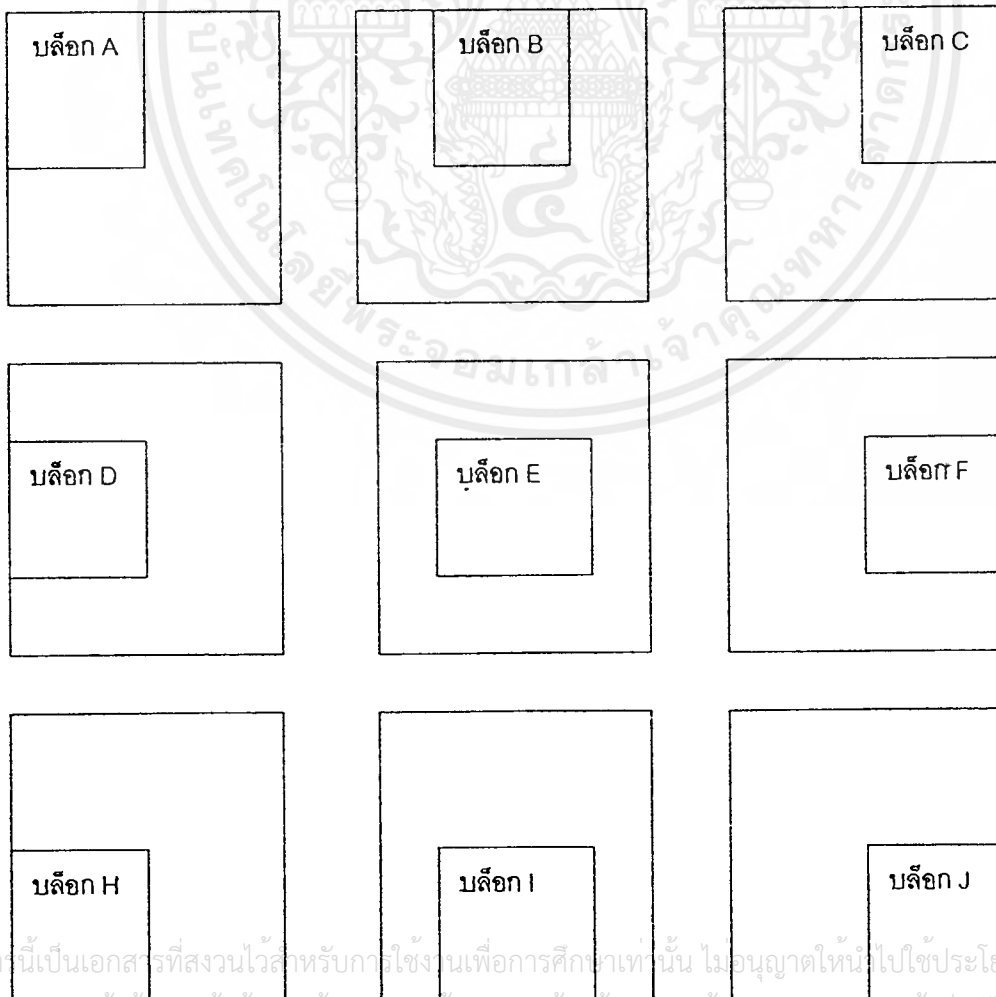


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การแบ่งพื้นที่ 1 แพทเทิร์นออกเป็นพื้นที่ย่อย

เป็นการนำเอาวิธีการฮิสโตแกรมทิศทางมาใช้ เมื่อถ่ายลายนี้นิวแม่มือจริง (สด) เข้ามาประมวลผลภาพ โดยการเตรียมข้อมูลตามแนวของลายเส้นหรือตามความเข้มของระดับเทา โดยแบ่งออกเป็นไปตามทิศทางที่จะกำหนดขึ้นมาบนพื้นที่ขนาด 2000x2000 จุดภาพแต่ต่อมาเห็นว่าจำนวนจุดภาพมากเกินไปทำให้สิ้นเปลืองความจำของคอมพิวเตอร์และต้องเสียเวลาในการประมวลผลมาก เพราะจำนวนจุดภาพมากเกินไป เมื่อพิจารณาเห็นว่าข้อมูลนั้นใหญ่เช่น ภายหลังจากการแบ่งบล็อกๆแล้ว จุดภาพ(รูปภาพที่ 5.1) ข้อมูลภาพของพื้นที่ใหญ่เท่ากับ 40x40 บล็อก เมื่อคำนวณออกมาผลคือลดการใช้ความจำของเครื่องคอมพิวเตอร์ลง จึงได้แบ่งพื้นที่ใหญ่ออกเป็นส่วนย่อยๆ 9 ส่วนเท่าๆกันตามรูปที่ 5.5 แล้วแบ่งพื้นที่หนึ่งๆให้มีขนาด 20x20 บล็อก จะทำให้ง่ายต่อการประมวลผลภาพให้เป็นทิศทางออกมาสะดวก, ง่ายและรวดเร็ว จึงมีพื้นที่ A,B,C,D,E,F,H,I และพื้นที่ J เป็นพื้นที่ๆใส่เข้ามาเพื่อให้ข้อมูลยังคงอยู่ไม่ขาดหายไป เพราะผลจากการแบ่ง 4 พื้นที่ อาจทำให้ข้อมูลตรงกลางที่บริเวณจุดใจกลางของลายนิวแม่มือบางจุดขาดหายไป จึงสร้างข้อมูลคลุมทำให้การเปรียบเทียบข้อมูลได้ดีขึ้น

รูปที่ 3.29 แสดงการแบ่งภาพออก 9 พื้นที่

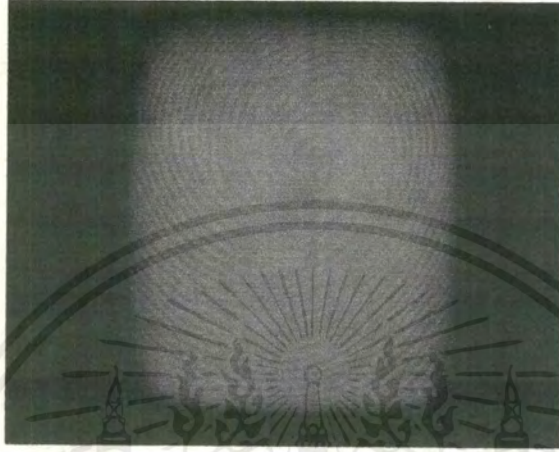


บทที่ 4

การทดลอง

การทดลองเกี่ยวกับการปรับปรุงภาพ

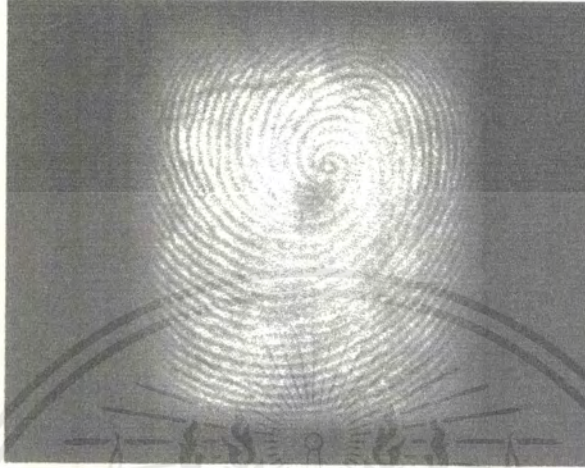
โดยทำการปรับปรุงภาพให้คมชัดขึ้นแสดงดังรูปต่อไปนี้



รูปที่ 4.1 แสดงรูปลายนิ้วมือที่ยังไม่ได้ปรับปรุงภาพ



รูปที่ 4.2 แสดงรูปลายนิ้วมือที่ทำการปรับปรุงภาพ โดยวิธี Sharpen

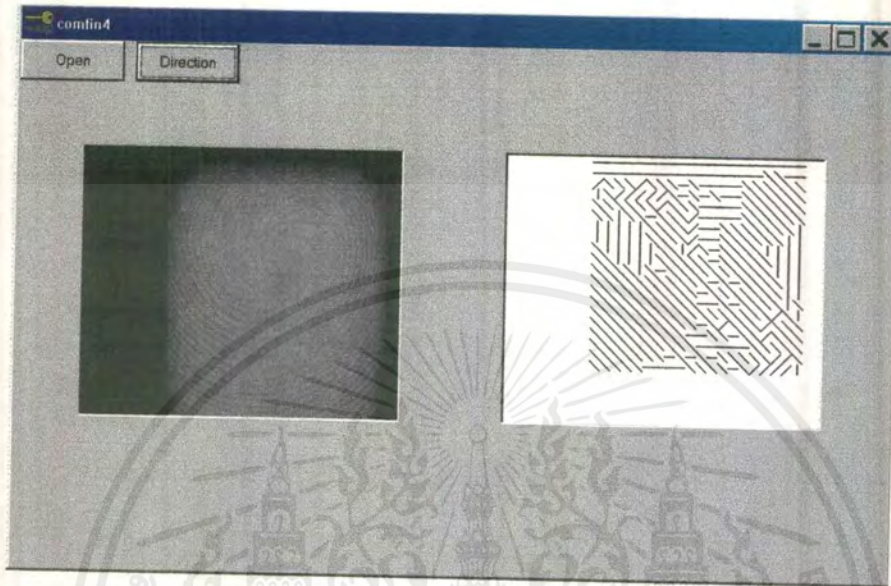


รูปที่ 4.3 แสดงรูปลายนิ้วมือที่ทำการปรับปรุงภาพ โดยวิธี Histogram Equalize

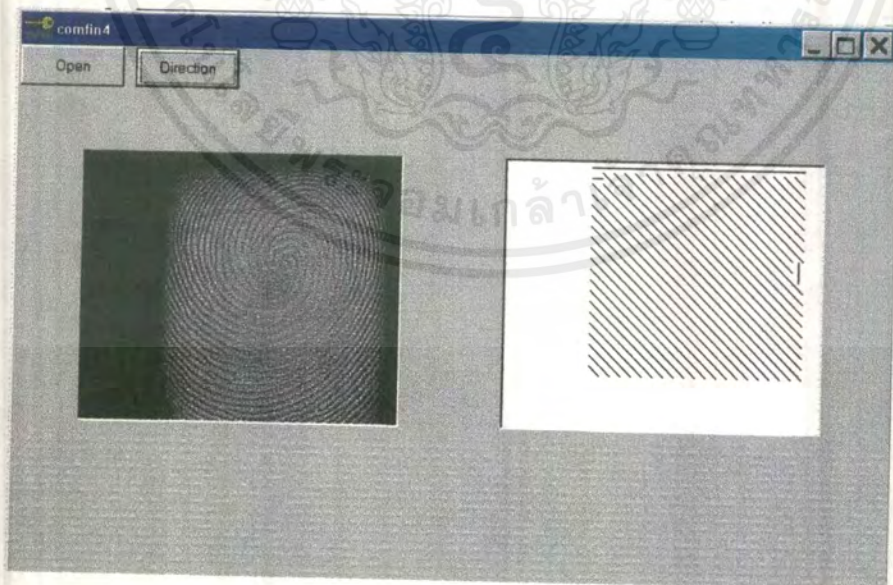


รูปที่ 4.4 แสดงรูปลายนิ้วมือที่ทำการปรับปรุงภาพ โดยวิธี High Pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

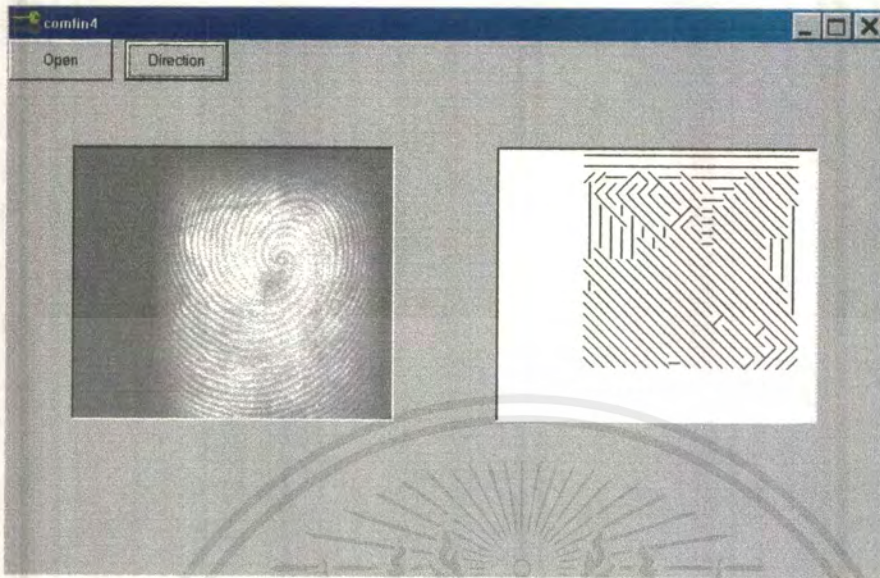


รูปที่ 4.5 แสดงภาพการหาทิศทางของภาพที่ยังไม่ได้รับการปรับปรุงภาพ

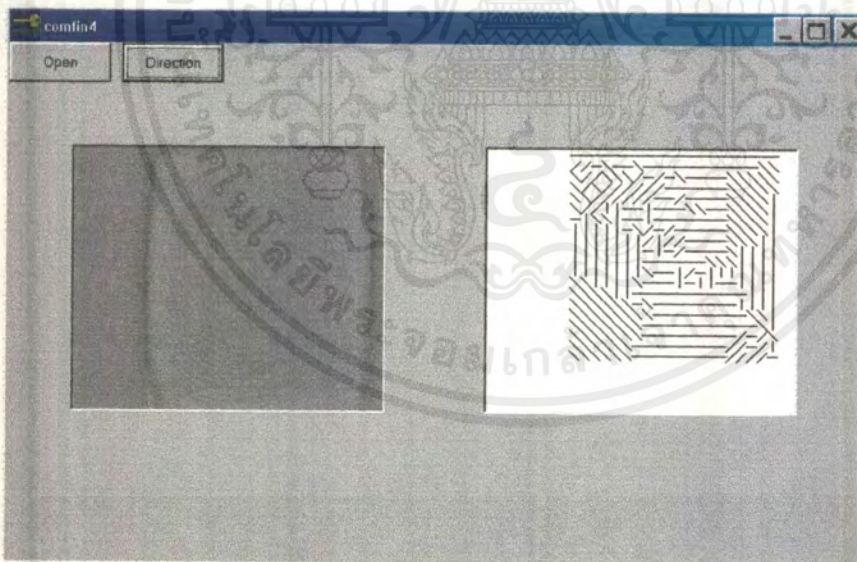


รูปที่ 4.6 แสดงการหาทิศทางของภาพที่ปรับปรุงภาพแบบ Sharpen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงรูปการหาทิศทางของภาพที่ทำการปรับปรุงภาพแบบ Histogram Equalize



รูปที่ 4.8 แสดงรูปการหาทิศทางของภาพที่ทำการปรับปรุงภาพแบบ High Pass Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงสถิติการเปรียบเทียบภาพที่ผ่าน High Pass Filter

ตารางแสดงจำนวนทิศทางของแต่ละพื้นที่ของภาพ

Area	Direction	Pic1	Pic2	Pic3	Pic4	Pic5	Pic6	Pic7	Pic8	Pic9	Pic10	Maximum	Minimum	average
Area1	0	43	40	33	43	35	41	33	34	33	35	43	33	37
Area2	0	76	70	69	76	70	76	70	67	70	69	76	69	71.3
Area3	0	54	49	56	57	52	55	55	52	55	50	57	49	53.5
Area4	2	42	41	44	46	45	45	44	44	44	46	46	41	44.1
Area5	0	73	72	67	68	73	73	68	66	68	66	73	66	69.4
Area6	0	48	44	50	49	51	48	51	45	51	46	51	44	48.3
Area7	3	52	46	54	44	51	50	51	49	52	44	54	44	49.3
Area8	0	74	83	78	76	82	83	84	78	84	81	84	74	80.3
Area9	0	54	58	59	56	60	57	64	54	64	57	64	54	58.3

โดยที่ได้ทำการทดลองกับภาพที่ผ่าน High Pass Filter จำนวนทั้งหมด 10 ภาพเหตุที่เลือกวิธี High Pass Filter มาทำการทดลองเนื่องจากเมื่อทำการปรับปรุงภาพแล้วทำการหาทิศทางได้ดีที่สุดและถูกต้องที่สุดและเมื่อนำภาพลายนิ้วมือของเจ้าของลายนิ้วมือเดียวกันมาเปรียบเทียบกับกันได้ผลดังตาราง

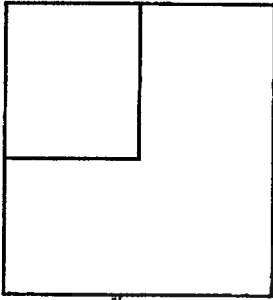
สรุปผลการทดลองการปรับปรุงภาพ

จากที่ได้ทำการปรับปรุงภาพหลายวิธีด้วยกันปรากฏว่าวิธีที่พออยู่ในข่ายวิธีที่ดีมีทั้งหมด 3 วิธีด้วยกันคือวิธี Sharpen, Histogram Equalize และ High Pass Filter ซึ่งเมื่อมองด้วยสายตาแล้วจะเห็นได้ว่าวิธี Sharpen ให้ภาพที่ดีที่สุดเมื่อเทียบกับวิธีอื่นๆ แต่เมื่อทำการหาทิศทางแล้วปรากฏว่าวิธี High Pass Filter สามารถหาทิศทางได้ดีที่สุด ตามด้วยภาพที่ยังไม่แก้ไข, Histogram Equalize ส่วนวิธี Sharpen กลับหาทิศทางไม่ได้ซึ่งเป็นเพราะการเรียงตัวของจุดสีของแต่ละ pixel เรียงตัวกันอย่างไม่ต่อเนื่อง

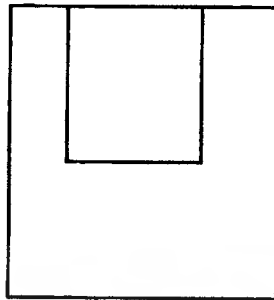
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองกับภาพที่เกิด Error ขึ้น

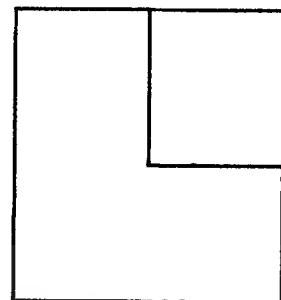
โดยที่พิจารณาการเกิด Error ขึ้นในแต่ละพื้นที่ของภาพโดยแบ่งพื้นที่ดังนี้



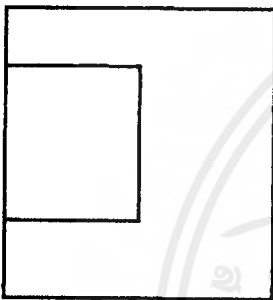
พื้นที่ที่ 1



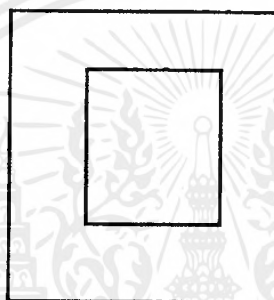
พื้นที่ที่ 2



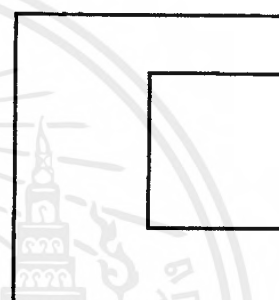
พื้นที่ที่ 3



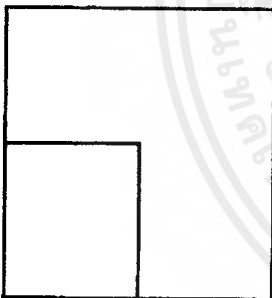
พื้นที่ที่ 4



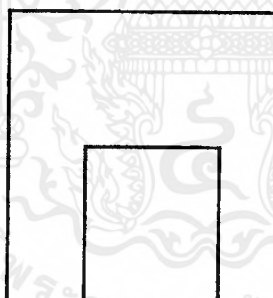
พื้นที่ที่ 5



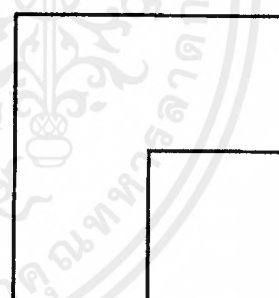
พื้นที่ที่ 6



พื้นที่ที่ 7



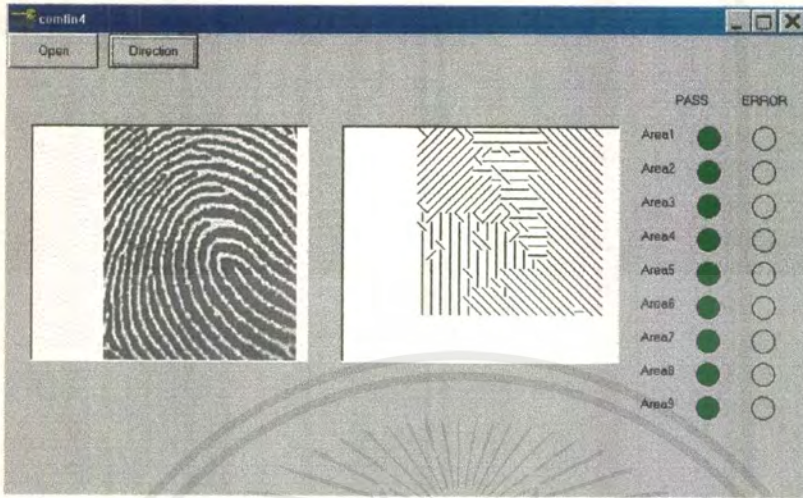
พื้นที่ที่ 8



พื้นที่ที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

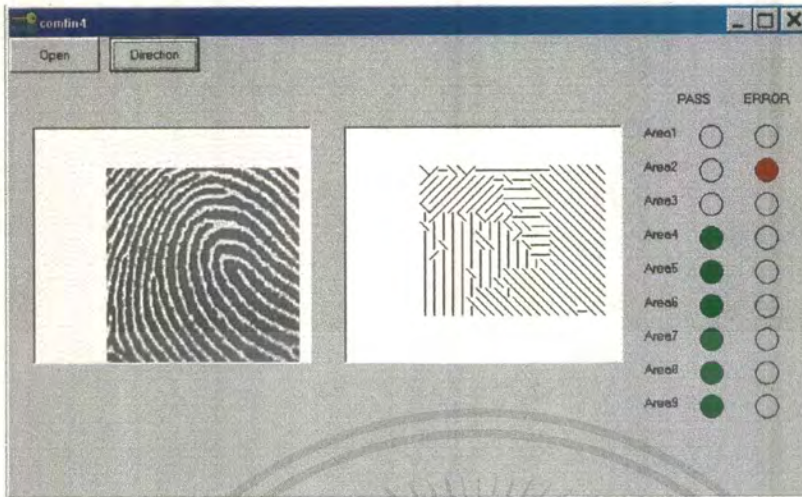
รูปที่ 4.9 แสดงลักษณะการแบ่งภาพออกเป็น 9 ส่วน
ผลการทดลองมีดังนี้



รูปที่ 4.10 แสดงลักษณะของที่สมบูรณ์



รูปที่ 4.11 แสดงภาพที่เกิดภาพไม่สมบูรณ์ด้านซ้ายของภาพ
ซึ่งจากภาพนี้ทำให้เกิด Error ที่ Area1,4 และ 7 ขึ้น ส่วน Area อื่นผ่าน



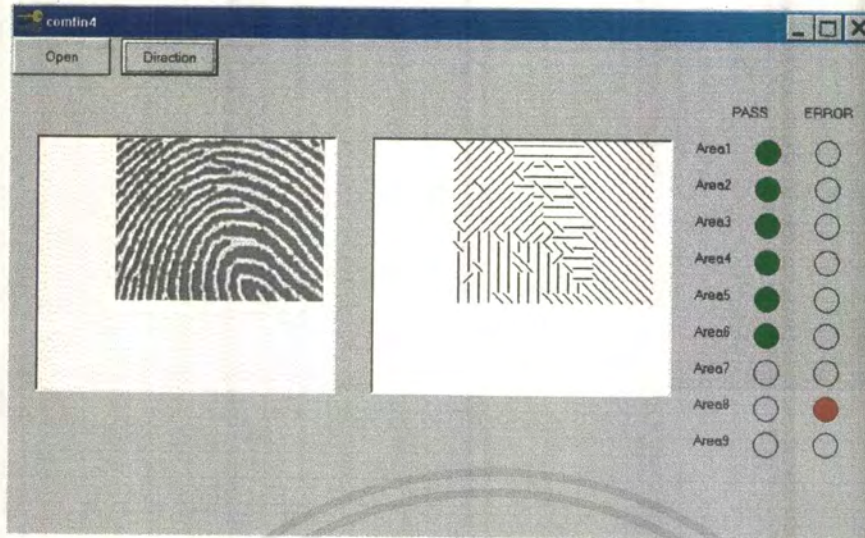
รูปที่ 4.12 แสดงภาพที่ไม่สมบูรณ์ด้านบนของภาพ

จากรูปจะเห็นได้ว่า Area1,2 และ 3 ไม่ผ่านการเปรียบเทียบ โดยที่ Area 2 เกิด Error จนเป็นทิศทางส่วนใหญ่ ส่วน Area 1 และ Area 3 การเกิด Error ไม่ได้เป็นทิศทางส่วนใหญ่แต่ก็มีผลทำให้การเปรียบเทียบไม่ผ่าน



รูปที่ 4.13 แสดงภาพที่ไม่สมบูรณ์ด้านขวาของภาพ

จากรูปจะเห็นได้ว่า Area3,5 และ 9 ไม่ผ่านการเปรียบเทียบ โดยที่ Area 3และ5 เกิด Error จนเป็นทิศทางส่วนใหญ่ ส่วน Area 9 การเกิด Error ไม่ได้เป็นทิศทางส่วนใหญ่แต่ก็มีผลทำให้การเปรียบเทียบไม่ผ่าน



รูปที่ 4.14 แสดงภาพที่ไม่สมบูรณ์ด้านล่างของภาพ

จากรูปจะเห็นได้ว่า Area7,8 และ 9 ไม่ผ่านการเปรียบเทียบโดยที่ Area 8 เกิด Error จนเป็นทิศทางส่วนใหญ่ ส่วน Area 7 และ9 การเกิด Error ไม่ได้เป็นทิศทางส่วนใหญ่แต่ก็มีผลทำให้การเปรียบเทียบไม่ผ่าน

บทที่ 5

สรุป,ปัญหาในการทำโครงการและข้อเสนอแนะ

5.1 สรุป

การตรวจสอบลายนิ้วมือมีขั้นตอนเริ่มจากการรับภาพเข้ามาโดยใช้กล้อง Phillips USB PC Camera PCA646VC รับภาพเข้ามา โดยกล้องจะอยู่ภายในกล่องซึ่งสร้างขึ้นมาเพื่อควบคุมแสงที่ส่องกระทบลายนิ้วมือให้เท่ากัน ทำการปรับให้กล้องรับภาพเข้ามาเป็นภาพระดับเทา(Gray Scale) เพื่อที่จะได้ไม่ต้องเขียน โปรแกรมเปลี่ยนภาพสีเป็นระดับเทาเพิ่มขึ้นอีก

ในขั้นตอนการปรับปรุงภาพได้ทำการทดลองวิธีการปรับปรุงภาพหลายวิธีด้วยกัน โดยวิธีที่นำมาใช้ในการปรับปรุงภาพ ได้แก่วิธี Sharpen , Histogram Equalize และ High Pass Filter โดยที่จากการทดลองพบว่าวิธีสุดท้ายคือ High Pass Filter เป็นวิธีที่ดีที่สุดเมื่อนำมาหาทิศทางแล้ว ได้ทิศทางที่ถูกต้องที่สุด

ขั้นตอนในการบันทึกข้อมูลและเปรียบเทียบจะเริ่มจากการแบ่งพื้นที่ของภาพออกเป็นทั้งหมด 9 พื้นที่ด้วยกันเพื่อลดจำนวนข้อมูลในการบันทึกลงในฐานข้อมูลและการเปรียบเทียบนอกจากนี้ก็เพื่อลดปัญหาความผิดพลาดหากรูปลายนิ้วมือมีข้อผิดพลาด เช่น สกปรก เป็นแผล ภาพไม่ชัดเจนบางส่วน เป็นต้น

ขั้นตอนการแสดงผลการเปรียบเทียบจะส่งผลการเปรียบเทียบออกพอร์ทอนุกรม RS-232 ส่งให้ Microcontroller แปลความหมายและตัดสินใจในการควบคุมการปิดเปิดอุปกรณ์ต่อไป

5.2 ปัญหา

ปัญหาค้าน Hardware

- กล้องที่ใช้ในการรับภาพ ไม่ว่าจะชัดเจนแค่ไหนแต่การแยกแยะความแตกต่างของระดับสีระหว่างร่องลายนิ้วมือกับรอยขนบนลายนิ้วในแต่พื้นที่จะขึ้นอยู่กับการจัดแสงจึงเป็นเรื่องยากที่จะทำให้แสงที่กระทบกับลายนิ้วมือเท่ากันทุกพื้นที่
- คอมพิวเตอร์ที่ช้าจะทำให้การประมวลผลช้าไปด้วย
- ไม่สามารถหากกล้องหรือตัวรับภาพลายนิ้วมือที่ให้ความชัดเจนได้ในเมืองไทยในราคาที่เหมาะสมได้
- ไม่สามารถเปรียบเทียบข้อมูลภาพที่ได้จากตัวรับภาพอื่นอีกเนื่องจากงบประมาณและไม่สามารถหาได้
- เนื่องจากอุปกรณ์ที่ใช้ไม่สามารถที่จะใช้บของภาควิชาได้จึงต้องใช้งบส่วนตัวทั้งหมด

ปัญหาด้าน Software

- ผู้จัดทำไม่มีความรู้เกี่ยวกับการเขียนโปรแกรมโดยใช้ภาษา Visual Basic มาก่อนทำให้เสียเวลาในการศึกษาวิธีการเขียนโปรแกรมไป
- การเขียนโปรแกรมให้เป็นโปรแกรมอัตโนมัติทำค่อนข้างยากเนื่องจากต้องเชื่อมโยงหลายๆ ส่วนเข้าด้วยกัน
- เนื่องจากการเขียนโปรแกรมบน Windows มีการดึงไฟล์ต่างๆ ของระบบมาใช้ทำในโปรแกรมทำให้โปรแกรมมีขนาดใหญ่

5.3 ข้อเสนอแนะ

เพื่อเป็นแนวทางในการพัฒนาโปรแกรมต่อไปในอนาคต เสนอเป็นข้อๆ ดังนี้

- อุปกรณ์ในการรับภาพควรรับภาพชัดเจนสามารถแยกแยะความแตกต่างระหว่างร่องและรอยนูนของลายนิ้วมือ ได้อย่างชัดเจนโดยไม่ต้องทำการจัดแสง
- ควรใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงๆ จะทำให้การพัฒนาโปรแกรมที่มีประสิทธิภาพสูงได้ดี โดยไม่ต้องกังวลปัญหาเกี่ยวกับระยะเวลาในการประมวลผล
- ควรศึกษาการเขียนโปรแกรมในระดับสูง โดยเฉพาะการเขียนโปรแกรมสำหรับวิศวกรรม

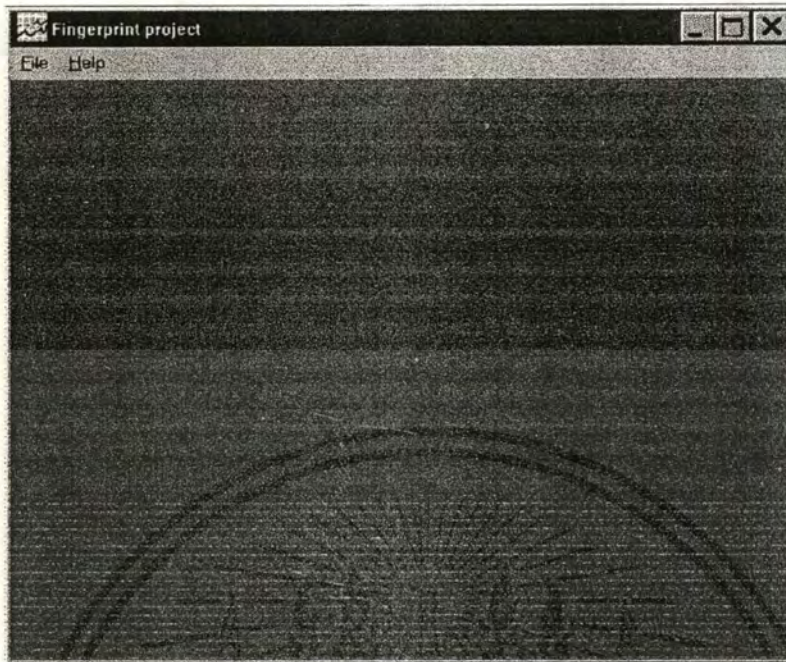
ภาคผนวก ก

Code ของ โปรแกรมที่ทำการตรวจเทียบลายนิ้วมือ
ประกอบด้วยตัว Form ทั้งหมดของโปรแกรมและ Code ของแต่ละ Form



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. MDIForm1 เป็น Form หลัก



SOURCE CODE

```
Private Sub mnuabout_Click()
```

```
Form5.Show
```

```
End Sub
```

```
Private Sub mnuAdd_data4_Click()
```

```
Form2.Show
```

```
End Sub
```

```
Private Sub mnuAdd_data8_Click()
```

```
Form1.Show
```

```
End Sub
```

```
Private Sub mnuCapture_Click()
```

```
Form8.Show
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub mnuCompare4_Click()
```

```
Form4.Show
```

```
End Sub
```

```
Private Sub mnuCompare8_Click()
```

```
Form3.Show
```

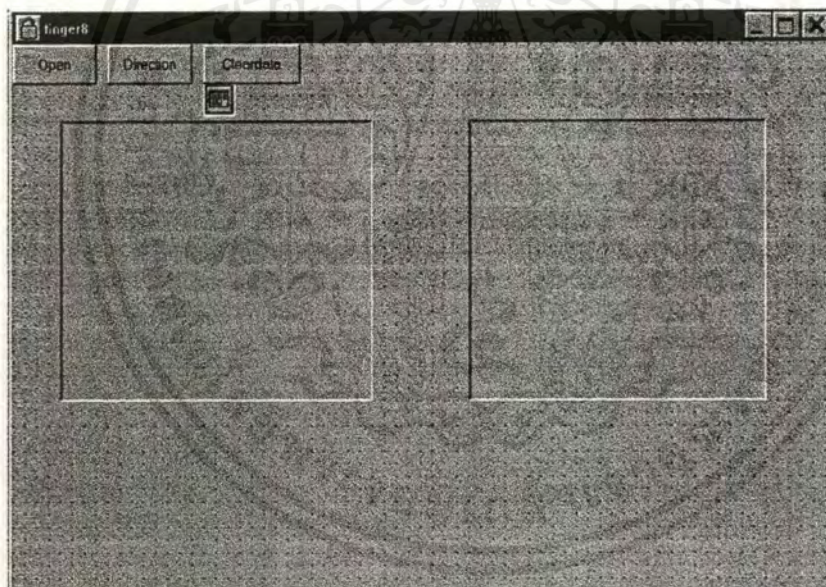
```
End Sub
```

```
Private Sub mnuExit_Click()
```

```
End
```

```
End Sub
```

2. Form1 เป็น Form บันทึกข้อมูล 8 ทิศทาง



SOURCE CODE

```
Option Explicit
```

```
Dim dblock(50, 50) As Single
```

```
Dim originx As Integer
```

```
Dim originy As Integer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdclear_Click()
Dim dbase As Database
Dim dtable As Recordset
Dim i As Integer
Set dbase = OpenDatabase(App.Path + "\finger.mdb")
Set dtable = dbase.OpenRecordset("finger8", dbOpenDynaset)
dtable.MoveLast
If dtable.RecordCount <> 0 Then
For i = 1 To dtable.RecordCount
dtable.Delete
dtable.MovePrevious
Next i
End If
dbase.Close
End Sub

```

```

Private Sub cmdDirection_Click()
Dim i As Integer, j As Integer, l As Integer, k As Integer
Dim xPixels As Double
Dim yPixels As Double
xPixels = 2000 - 1
yPixels = 2000 - 1
l = 0
Picture3.BackColor = QBColor(15)
For j = 0 To yPixels Step 100
k = 0
For i = 0 To xPixels Step 100
dblock(k, l) = directionblock(i, j)
Select Case dblock(k, l)
Case 0
Picture3.Line (i, (j + 50))-(i + 100, (j + 50))
Case 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Picture3.Line (i, (j + 100))-((i + 100), j)

Case 2

Picture3.Line ((i + 50), j)-((i + 50), (j + 100))

Case 3

Picture3.Line (i, j)-((i + 100), (j + 100))

Case 4

Picture3.Line (i, (j + 75))-((i + 100), (j + 25))

Case 5

Picture3.Line ((i + 25), (j + 100))-((i + 75), j)

Case 6

Picture3.Line ((i + 25), j)-((i + 75), (j + 100))

Case 7

Picture3.Line (i, (j + 25))-((i + 100), (j + 75))

End Select

k = k + 1

Next

l = l + 1

Next

Call rec

End Sub

Sub rec()

Dim area1(8) As Integer, area2(8) As Integer, area3(8) As Integer

Dim area4(8) As Integer, area5(8) As Integer, area6(8) As Integer

Dim area7(8) As Integer, area8(8) As Integer, area9(8) As Integer

Dim maxdirec(10) As Integer, maxis(10) As Integer

Dim d As Integer, i As Integer, j As Integer

Dim dbase As Database

Dim dtable As Recordset

'areal

For j = 0 To 9

For i = 0 To 9

d = countdirec(i, j)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

area1(d) = area1(d) + 1

Next

Next

maxdirec(1) = area1(0)

For i = 1 To 7

If (maxdirec(1) < area1(i)) Then

maxdirec(1) = area1(i)

maxis(1) = i

End If

Next

'area2

For j = 0 To 9

For i = 6 To 15

d = countdirec(i, j)

area2(d) = area2(d) + 1

Next

Next

maxdirec(2) = area2(0)

For i = 1 To 7

If (maxdirec(2) < area2(i)) Then

maxdirec(2) = area2(i)

maxis(2) = i

End If

Next

'area3

For j = 0 To 9

For i = 10 To 19

d = countdirec(i, j)

area3(d) = area3(d) + 1

Next

Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

maxdirec(3) = area3(0)

For i = 1 To 7

If (maxdirec(3) < area3(i)) Then

maxdirec(3) = area3(i)

maxis(3) = i

End If

Next

'area4

For j = 6 To 15

For i = 0 To 9

d = countdirec(i, j)

area4(d) = area4(d) + 1

Next

Next

maxdirec(4) = area4(0)

For i = 1 To 7

If (maxdirec(4) < area4(i)) Then

maxdirec(4) = area4(i)

maxis(4) = i

End If

Next

'area5

For j = 6 To 15

For i = 6 To 15

d = countdirec(i, j)

area5(d) = area5(d) + 1

Next

Next

maxdirec(5) = area5(0)

For i = 1 To 7

If (maxdirec(5) < area5(i)) Then

maxdirec(5) = area5(i)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

maxis(5) = i

End If

Next

'area6

For j = 6 To 15

For i = 10 To 19

d = countdirec(i, j)

area6(d) = area6(d) + 1

Next

Next

maxdirec(6) = area6(0)

For i = 1 To 7

If (maxdirec(6) < area6(i)) Then

maxdirec(6) = area6(i)

maxis(6) = i

End If

Next

'area7

For j = 10 To 19

For i = 0 To 9

d = countdirec(i, j)

area7(d) = area7(d) + 1

Next

Next

maxdirec(7) = area7(0)

For i = 1 To 7

If (maxdirec(7) < area7(i)) Then

maxdirec(7) = area7(i)

maxis(7) = i

End If

Next

'area8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For j = 10 To 19
    For i = 6 To 15
        d = countdirec(i, j)
        area8(d) = area8(d) + 1
    Next
Next
maxdirec(8) = area8(0)
For i = 1 To 7
    If (maxdirec(8) < area8(i)) Then
        maxdirec(8) = area8(i)
        maxis(8) = i
    End If
Next

```

```

'area9
For j = 10 To 19
    For i = 10 To 19
        d = countdirec(i, j)
        area9(d) = area9(d) + 1
    Next
Next
maxdirec(9) = area9(0)
For i = 1 To 7
    If (maxdirec(9) < area9(i)) Then
        maxdirec(9) = area9(i)
        maxis(9) = i
    End If
Next

```

```
Set dbase = OpenDatabase(App.Path + "\finger.mdb")
```

```
Set dtable = dbase.OpenRecordset("finger8", dbOpenDynaset)
```

```
If dtable.BOF = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i = 1 To 9
    dtable.AddNew
    dtable("area") = i
    dtable("direc") = maxis(i)
    dtable("maxdirec") = maxdirec(i)
    dtable.Update
Next
End If

dbase.Close

End Sub

Function countdirec(x As Integer, y As Integer)
    Dim direc As Integer
    Select Case dblock(x, y)
        Case 0
            direc = 0
        Case 1
            direc = 1
        Case 2
            direc = 2
        Case 3
            direc = 3
        Case 4
            direc = 4
        Case 5
            direc = 5
        Case 6
            direc = 6
        Case 7
            direc = 7
    
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Select

countdirec = direc

End Function

Private Sub cmdOpen_Click()

CommonDialog1.CancelError = True

On Error GoTo ErrHandler

CommonDialog1.Flags = cdIOFNHideReadOnly

CommonDialog1.Filter = "Picture Files (*.BMP)|*.bmp|JPEG(*.JPG)|*.jpg|All Files (*.*)|*.*"

CommonDialog1.FilterIndex = 2

CommonDialog1.ShowOpen

Picture1.Picture = LoadPicture(CommonDialog1.FileName)

Exit Sub

ErrHandler:

Exit Sub

End Sub

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)

Form1.Caption = Picture1.Point(x, y)

End Sub

Function directionblock(x As Integer, y As Integer)

Dim bdirec As Single

Dim averdirec As Single

Dim arrpixdirec(1000, 1000) As Single

Dim x1 As Integer, y1 As Integer, x2 As Integer, y2 As Integer

Dim i As Integer, j As Integer, k As Integer, l As Integer

Dim ddirec(8) As Single

x1 = x

y1 = y

x2 = x + 100

y2 = y + 100

originx = 0

originy = 0

For i = 0 To 3

 ddirec(i) = 0

Next

l = 0

For j = y1 To y2 Step 10

 k = 0

 For i = x1 To x2 Step 10

 'Picture2.PSet (i, j), Picture1.Point(i, j)

 arrpixdirec(k, l) = directionpixel(i, j)

 Select Case arrpixdirec(k, l)

 Case 0

 ddirec(0) = ddirec(0) + 1

 Case 1

 ddirec(1) = ddirec(1) + 1

 Case 2

 ddirec(2) = ddirec(2) + 1

 Case 3

 ddirec(3) = ddirec(3) + 1

 Case 4

 ddirec(4) = ddirec(4) + 1

 Case 5

 ddirec(5) = ddirec(5) + 1

 Case 6

 ddirec(6) = ddirec(6) + 1

 Case 7

 ddirec(7) = ddirec(7) + 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Select
.k = k + 1
Next
l = l + 1
Next

averdirec = 0
k = ddirec(0)
For i = 1 To 7
    If (k < ddirec(i)) Then
        averdirec = i
        k = ddirec(i)
    End If
Next
directionblock = averdirec

End Function
Function directionpixel(x As Integer, y As Integer)
Dim x1 As Integer, x2 As Integer, y1 As Integer, y2 As Integer
Dim sumdirec(16) As Single
Dim unidirec(8) As Single
Dim direc As Single, direcold As Single
Dim i As Integer, j As Integer
Dim pex As Single, pext As Single
Dim dd As Integer
Dim dis As Integer
dis = 20
dd = 5
x1 = 800
y1 = 0
x2 = 2800
y2 = 2000

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pex = Picture1.Point(x, y)

For i = 0 To 7

 sumdirec(i) = 0

Next

'direction 0

i = x + 1

If (i <= x2) Then

 Do While ((i <= x + 1 + dis) And (i <= x2))

 pext = Picture1.Point(i, y)

 sumdirec(0) = sumdirec(0) + Abs(pex - pext)

 i = i + dd

 Loop

Else: sumdirec(0) = 1E+24

End If

'direction 1

i = x + 1

j = y - 1

If ((i <= x2) And (j >= y1)) Then

 Do While ((i <= x + 1 + dis) And (i <= x2) And (j >= y - 1 - dis) And (j >= y1))

 pext = Picture1.Point(i, j)

 sumdirec(1) = sumdirec(1) + Abs(pex - pext)

 i = i + dd

 j = j - dd

 Loop

Else: sumdirec(1) = 1E+24

End If

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

'direction2

j = y - 1

'dd = 0

If (j >= y1) Then

Do While ((j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(x, j)

sumdirec(2) = sumdirec(2) + Abs(pex - pext)

j = j - dd

Loop

Else: sumdirec(2) = 1E+24

End If

'direction3

i = x - 1

j = y - 1

If ((i >= x1) And (j >= y1)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(3) = sumdirec(3) + Abs(pex - pext)

i = i - dd

j = j - dd

Loop

Else: sumdirec(3) = 1E+24

End If

'direction4

i = x - 1

j = y + 1

If ($i \geq x1$) Then

Do While ($(i \geq x - 1 - \text{dis})$ And ($i \geq x1$))

pext = Picture1.Point(i, y)

sumdirec(4) = sumdirec(4) + Abs(pex - pext)

i = i - dd

Loop

Else: sumdirec(4) = 1E+24

End If

'direction5

i = x - 1

j = y + 1

If ($(i \geq x1)$ And ($j \leq y2$)) Then

Do While ($(i \geq x - 1 - \text{dis})$ And ($i \geq x1$) And ($j \leq y + 1 + \text{dis}$) And ($j \leq y2$))

pext = Picture1.Point(i, j)

sumdirec(5) = sumdirec(5) + Abs(pex - pext)

i = i - dd

j = j + dd

Loop

Else: sumdirec(5) = 1E+24

End If

'direction6

j = y + 1

If ($j \leq y2$) Then

Do While ($(j \leq y + 1 + \text{dis})$ And ($j \leq y2$))

pext = Picture1.Point(x, j)

sumdirec(6) = sumdirec(6) + Abs(pex - pext)

j = j + dd

Loop

Else: sumdirec(6) = 1E+24

End If

'direction7

i = x + 1

j = y + 1

If ((i <= x2) And (j <= y2)) Then

Do While ((i >= x + 1 + dis) And (i <= x2) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(7) = sumdirec(7) + Abs(pex - pext)

i = i + dd

j = j + dd

Loop

Else: sumdirec(7) = 1E+24

End If

'direction 8

i = x + 1

j = y - 1

If ((i <= x2) And (j >= y1)) Then

Do While ((i <= x + 1 + dis) And (i <= x2) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(8) = sumdirec(8) + Abs(pex - pext)

i = i + dd

j = j - (dd / 2)

Loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Else: sumdirec(1) = 1E+24

End If

'direction 9

i = x + 1

j = y - 1

If ((i <= x2) And (j >= y1)) Then

Do While ((i <= x + 1 + dis) And (i <= x2) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(9) = sumdirec(9) + Abs(pex - pext)

i = i + (dd / 2)

j = j - dd

Loop

Else: sumdirec(1) = 1E+24

End If

'direction10

i = x - 1

j = y - 1

If ((i >= x1) And (j >= y1)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(10) = sumdirec(10) + Abs(pex - pext)

i = i - (dd / 2)

j = j - dd

Loop

Else: sumdirec(3) = 1E+24

End If

'direction 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$i = x - 1$

$j = y - 1$

If $((i \geq x1) \text{ And } (j \geq y1))$ Then

Do While $((i \geq x - 1 - \text{dis}) \text{ And } (i \geq x1) \text{ And } (j \geq y - 1 - \text{dis}) \text{ And } (j \geq y1))$

$\text{pext} = \text{Picture1.Point}(i, j)$

$\text{sumdirec}(11) = \text{sumdirec}(11) + \text{Abs}(\text{pex} - \text{pext})$

$i = i - \text{dd}$

$j = j - (\text{dd} / 2)$

Loop

Else: $\text{sumdirec}(3) = 1\text{E}+24$

End If

'direction 12

$i = x - 1$

$j = y + 1$

If $((i \geq x1) \text{ And } (j \leq y2))$ Then

Do While $((i \geq x - 1 - \text{dis}) \text{ And } (i \geq x1) \text{ And } (j \leq y + 1 + \text{dis}) \text{ And } (j \leq y2))$

$\text{pext} = \text{Picture1.Point}(i, j)$

$\text{sumdirec}(12) = \text{sumdirec}(12) + \text{Abs}(\text{pex} - \text{pext})$

$i = i - \text{dd}$

$j = j + (\text{dd} / 2)$

Loop

Else: $\text{sumdirec}(5) = 1\text{E}+24$

End If

'direction 13

$i = x - 1$

$j = y + 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

If ((i >= x1) And (j <= y2)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(13) = sumdirec(13) + Abs(pex - pext)

i = i - (dd / 2)

j = j + dd

Loop

Else: sumdirec(5) = 1E+24

End If

'direction 14

i = x + 1

j = y + 1

If ((i <= x2) And (j <= y2)) Then

Do While ((i >= x + 1 + dis) And (i <= x2) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(14) = sumdirec(14) + Abs(pex - pext)

i = i + (dd / 2)

j = j + dd

Loop

Else: sumdirec(7) = 1E+24

End If

'direction 15

i = x + 1

j = y + 1

If ((i <= x2) And (j <= y2)) Then

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Do While ((i >= x + 1 + dis) And (i <= x2) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(15) = sumdirec(15) + Abs(pex - pext)

i = i + dd

j = j + (dd / 2)

Loop

Else: sumdirec(7) = 1E+24

End If

unidirec(0) = sumdirec(0) + sumdirec(4)

unidirec(1) = sumdirec(1) + sumdirec(5)

unidirec(2) = sumdirec(2) + sumdirec(6)

unidirec(3) = sumdirec(3) + sumdirec(7)

unidirec(4) = sumdirec(8) + sumdirec(12)

unidirec(5) = sumdirec(9) + sumdirec(13)

unidirec(6) = sumdirec(10) + sumdirec(14)

unidirec(7) = sumdirec(11) + sumdirec(15)

direcold = unidirec(0)

direc = 0

For i = 0 To 7

If (direcold > unidirec(i)) Then

direcold = unidirec(i)

direc = i

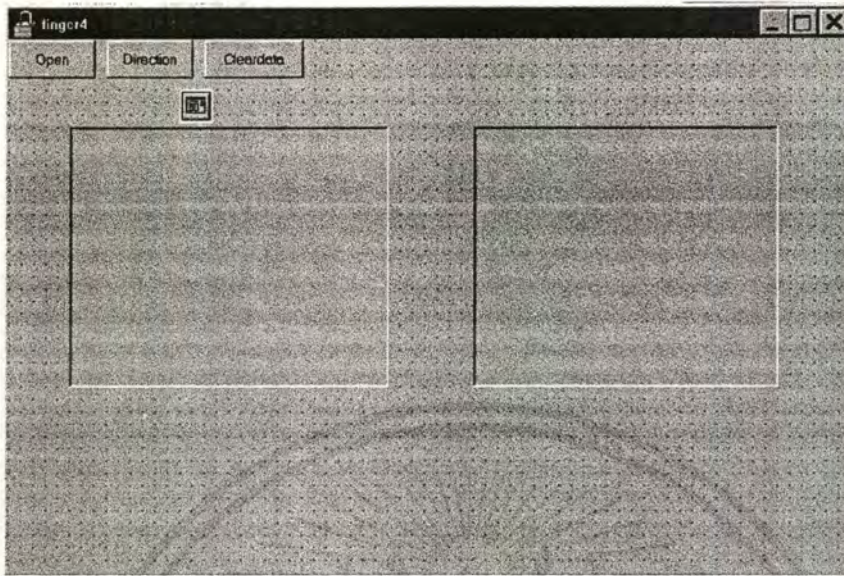
End If

Next

directionpixel = direc

End Function

3. Form2 เป็น Form บันทึกข้อมูล 4 ทิศทาง



SOURCE CODE

Option Explicit

Dim dblock(50, 50) As Single

Dim originx As Integer

Dim originy As Integer

Private Sub cmdclear_Click()

Dim dbase As Database

Dim dtable As Recordset

Dim i As Integer

Set dbase = OpenDatabase(App.Path + "\finger.mdb")

Set dtable = dbase.OpenRecordset("finger4", dbOpenDynaset)

dtable.MoveLast

If dtable.RecordCount < 0 Then

For i = 1 To dtable.RecordCount

 dtable.Delete

 dtable.MovePrevious

Next i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

dbase.Close

End Sub

Private Sub cmdDirection_Click()

Dim i As Integer, j As Integer, l As Integer, k As Integer

Dim xPixels As Double

Dim yPixels As Double

xPixels = 2800 - 1

yPixels = 2000 - 1

l = 0

Picture2.BackColor = QBColor(15)

For j = 0 To yPixels Step 100

k = 0

For i = 800 To xPixels Step 100

dblock(k, l) = directionblock(i, j)

Select Case dblock(k, l)

Case 0

Picture2.Line (i, (j + 50))-((i + 100), (j + 50))

Case 1

Picture2.Line (i, (j + 100))-((i + 100), j)

Case 2

Picture2.Line ((i + 50), j)-((i + 50), (j + 100))

Case 3

Picture2.Line (i, j)-((i + 100), (j + 100))

End Select

k = k + 1

Next

l = l + 1

Next

Call rec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub

Sub rec()
Dim area1(8) As Integer, area2(8) As Integer, area3(8) As Integer
Dim area4(8) As Integer, area5(8) As Integer, area6(8) As Integer
Dim area7(8) As Integer, area8(8) As Integer, area9(8) As Integer
Dim maxdirec(10) As Integer, maxis(10) As Integer

Dim d As Integer, i As Integer, j As Integer

Dim dbase As Database
Dim dtable As Recordset

'area1
For j = 0 To 9
    For i = 0 To 9
        d = countdirec(i, j)
        area1(d) = area1(d) + 1
    Next
Next

maxdirec(1) = area1(0)
For i = 1 To 7
    If (maxdirec(1) < area1(i)) Then
        maxdirec(1) = area1(i)
        maxis(1) = i
    End If
Next

'area2
For j = 0 To 9
    For i = 6 To 15
        d = countdirec(i, j)
        area2(d) = area2(d) + 1
    Next
Next

maxdirec(2) = area2(0)

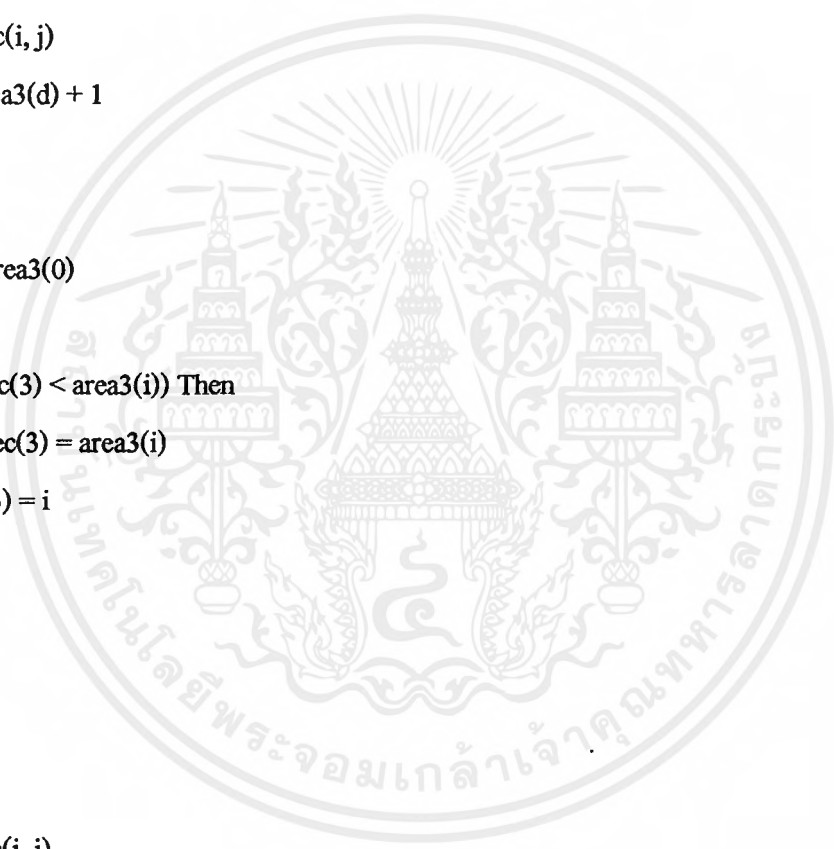
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i = 1 To 7
    If (maxdirec(2) < area2(i)) Then
        maxdirec(2) = area2(i)
        maxis(2) = i
    End If
Next
'area3
For j = 0 To 9
    For i = 10 To 19
        d = countdirec(i, j)
        area3(d) = area3(d) + 1
    Next
Next
maxdirec(3) = area3(0)
For i = 1 To 7
    If (maxdirec(3) < area3(i)) Then
        maxdirec(3) = area3(i)
        maxis(3) = i
    End If
Next
'area4
For j = 6 To 15
    For i = 0 To 9
        d = countdirec(i, j)
        area4(d) = area4(d) + 1
    Next
Next
maxdirec(4) = area4(0)
For i = 1 To 7
    If (maxdirec(4) < area4(i)) Then
        maxdirec(4) = area4(i)
        maxis(4) = i

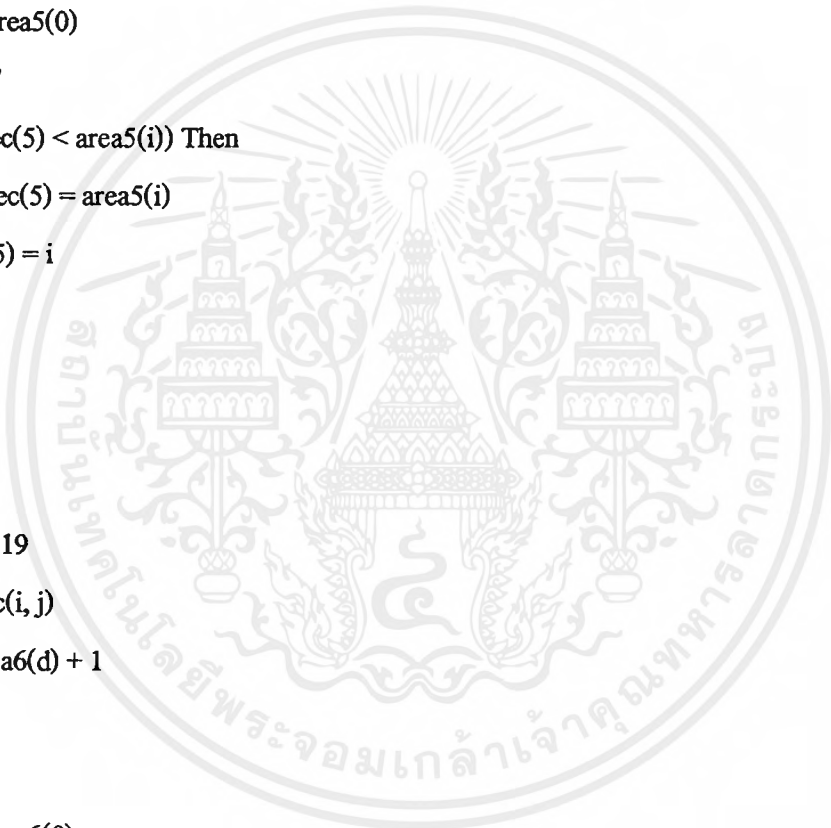
```



```

End If
Next
'area5
For j = 6 To 15
  For i = 6 To 15
    d = countdirec(i, j)
    area5(d) = area5(d) + 1
  Next
Next
maxdirec(5) = area5(0)
For i = 1 To 7
  If (maxdirec(5) < area5(i)) Then
    maxdirec(5) = area5(i)
    maxis(5) = i
  End If
Next
'area6
For j = 6 To 15
  For i = 10 To 19
    d = countdirec(i, j)
    area6(d) = area6(d) + 1
  Next
Next
maxdirec(6) = area6(0)
For i = 1 To 7
  If (maxdirec(6) < area6(i)) Then
    maxdirec(6) = area6(i)
    maxis(6) = i
  End If
Next
'area7
For j = 10 To 19

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i = 0 To 9
d = countdirec(i, j)
area7(d) = area7(d) + 1
Next
Next
maxdirec(7) = area7(0)
For i = 1 To 7
If (maxdirec(7) < area7(i)) Then
maxdirec(7) = area7(i)
maxis(7) = i
End If
Next
'area8
For j = 10 To 19
For i = 6 To 15
d = countdirec(i, j)
area8(d) = area8(d) + 1
Next
Next
maxdirec(8) = area8(0)
For i = 1 To 7
If (maxdirec(8) < area8(i)) Then
maxdirec(8) = area8(i)
maxis(8) = i
End If
Next
'area9
For j = 10 To 19
For i = 10 To 19
d = countdirec(i, j)
area9(d) = area9(d) + 1
Next

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Next

maxdirec(9) = area9(0)

For i = 1 To 7

If (maxdirec(9) < area9(i)) Then

maxdirec(9) = area9(i)

maxis(9) = i

End If

Next

Set dbase = OpenDatabase(App.Path + "\finger.mdb")

Set dtable = dbase.OpenRecordset("finger4", dbOpenDynaset)

If dtable.BOF = True Then

For i = 1 To 9

dtable.AddNew

dtable("area") = i

dtable("direc") = maxis(i)

dtable("maxdirec") = maxdirec(i)

dtable.Update

Next

End If

dbase.Close

End Sub

Function countdirec(x As Integer, y As Integer)

Dim direc As Integer

Select Case dblock(x, y)

Case 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

direc = 0

Case 1

direc = 1

Case 2

direc = 2

Case 3

direc = 3

End Select

countdirec = direc

End Function

Private Sub cmdOpen_Click()

CommonDialog1.CancelError = True

On Error GoTo ErrHandler

CommonDialog1.Flags = cdlOFNHideReadOnly

CommonDialog1.Filter = "Picture Files (*.BMP)|*.bmp|JPEG(*.JPG)|*.jpg|All Files (*.*)|*.*"

CommonDialog1.FilterIndex = 2

CommonDialog1.ShowOpen

Picture1.Picture = LoadPicture(CommonDialog1.FileName)

Exit Sub

ErrHandler:

Exit Sub

End Sub

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)

Form1.Caption = Picture1.Point(x, y)

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function directionblock(x As Integer, y As Integer)

Dim bdirec As Single

Dim averdirec As Single

Dim arrpixdirec(1000, 1000) As Single

Dim x1 As Integer, y1 As Integer, x2 As Integer, y2 As Integer

Dim i As Integer, j As Integer, k As Integer, l As Integer

Dim ddirec(8) As Single

x1 = x

y1 = y

x2 = x + 100

y2 = y + 100

originx = 0

originy = 0

For i = 0 To 3

 ddirec(i) = 0

Next

l = 0

For j = y1 To y2 Step 10

 k = 0

 For i = x1 To x2 Step 10

 arrpixdirec(k, l) = directionpixel(i, j)

 Select Case arrpixdirec(k, l)

 Case 0

 ddirec(0) = ddirec(0) + 1

 Case 1

 ddirec(1) = ddirec(1) + 1

 Case 2

 ddirec(2) = ddirec(2) + 1

 Case 3

 ddirec(3) = ddirec(3) + 1

```

End Select
k = k + 1
Next
l = l + 1
Next

averdiree = 0
k = ddirec(0)
For i = 1 To 3
    If (k < ddirec(i)) Then
        averdirec = i
        k = ddirec(i)
    End If
Next
directionblock = averdirec
'Debug.Print averdirec; ddirec(0); ddirec(1); ddirec(2); ddirec(3)
End Function
Function directionpixel(x As Integer, y As Integer)
Dim x1 As Integer, x2 As Integer, y1 As Integer, y2 As Integer
Dim sumdirec(16) As Single
Dim unidirec(8) As Single
Dim direc As Single, direcold As Single
Dim i As Integer, j As Integer
Dim pex As Single, pext As Single
Dim dd As Integer
Dim dis As Integer
dis = 20
dd = 5
x1 = 800
y1 = 0
x2 = 2800
y2 = 2000

```

pex = Picture1.Point(x, y)

For i = 0 To 7

 sumdirec(i) = 0

Next

'direction 0

i = 0

i = x + 1

If (i <= x2) Then

 Do While ((i <= x + 1 + dis) And (i <= x2))

 pext = Picture1.Point(i, y)

 sumdirec(0) = sumdirec(0) + Abs(pex - pext)

 i = i + dd

 Loop

Else: sumdirec(0) = 1E+24

End If

'direction 1

i = j = 0

i = x + 1

j = y - 1

If ((i <= x2) And (j >= y1)) Then

 Do While ((i <= x + 1 + dis) And (i <= x2) And (j >= y - 1 - dis) And (j >= y1))

 pext = Picture1.Point(i, j)

 sumdirec(1) = sumdirec(1) + Abs(pex - pext)

 i = i + dd

 j = j - dd

 Loop

Else: sumdirec(1) = 1E+24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

'direction2

j = 0

j = y - 1

If (j >= y1) Then

Do While ((j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(x, j)

sumdirec(2) = sumdirec(2) + Abs(pex - pext)

j = j - dd

Loop

Else: sumdirec(2) = 1E+24

End If

'direction3

i = j = 0

i = x - 1

j = y - 1

If ((i >= x1) And (j >= y1)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(3) = sumdirec(3) + Abs(pex - pext)

i = i - dd

j = j - dd

Loop

Else: sumdirec(3) = 1E+24

End If

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

'direction4

i = j = 0

i = x - 1

j = y + 1

If (i >= x1) Then

Do While ((i >= x - 1 - dis) And (i >= x1))

pext = Picture1.Point(i, y)

sumdirec(4) = sumdirec(4) + Abs(pex - pext)

i = i - dd

Loop

Else: sumdirec(4) = 1E+24

End If

'direction5

i = j = 0

i = x - 1

j = y + 1

If ((i >= x1) And (j <= y2)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(5) = sumdirec(5) + Abs(pex - pext)

i = i - dd

j = j + dd

Loop

Else: sumdirec(5) = 1E+24

End If

'direction6

j = 0

$j = y + 1$

If ($j \leq y2$) Then

Do While (($j \leq y + 1 + dis$) And ($j \leq y2$))

$pext = \text{Picture1.Point}(x, j)$

$\text{sumdirec}(6) = \text{sumdirec}(6) + \text{Abs}(pex - pext)$

$j = j + dd$

Loop

Else: $\text{sumdirec}(6) = 1E+24$

End If

'direction7

$i = j = 0$

$i = x + 1$

$j = y + 1$

If (($i \leq x2$) And ($j \leq y2$)) Then

Do While (($i \geq x + 1 + dis$) And ($i \leq x2$) And ($j \leq y + 1 + dis$) And ($j \leq y2$))

$pext = \text{Picture1.Point}(i, j)$

$\text{sumdirec}(7) = \text{sumdirec}(7) + \text{Abs}(pex - pext)$

$i = i + dd$

$j = j + dd$

Loop

Else: $\text{sumdirec}(7) = 1E+24$

End If

$\text{unidirec}(0) = \text{sumdirec}(0) + \text{sumdirec}(4)$

$\text{unidirec}(1) = \text{sumdirec}(1) + \text{sumdirec}(5)$

$\text{unidirec}(2) = \text{sumdirec}(2) + \text{sumdirec}(6)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

unidirec(3) = sumdirec(3) + sumdirec(7)

direcold = unidirec(0)

direc = 0

For i = 0 To 3

 If (direcold > unidirec(i)) Then

 direcold = unidirec(i)

 direc = i

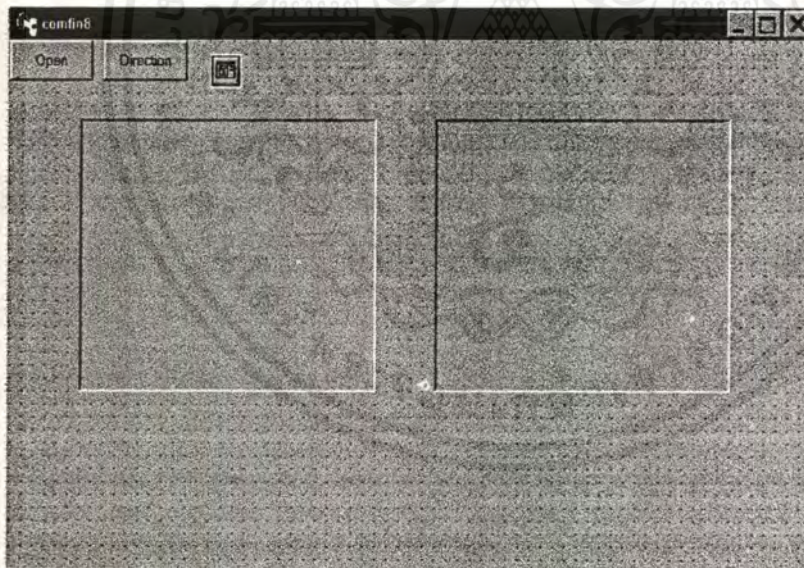
 End If

Next

directionpixel = direc

End Function

3. Form3 เป็น Form เปรียบเทียบข้อมูล 8 ทิศทาง



SOURCE CODE

Option Explicit

Dim dblock(50, 50) As Single

Dim originx As Integer

Dim originy As Integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Sub cmdDirection_Click()

Dim i As Integer, j As Integer, l As Integer, k As Integer

Dim xPixels As Double

Dim yPixels As Double

xPixels = 2800 - 1

yPixels = 2000 - 1

l = 0

Picture2.BackColor = QBColor(15)

For j = 0 To yPixels Step 100

 k = 0

 For i = 800 To xPixels Step 100

 dblock(k, l) = directionblock(i, j)

 Select Case dblock(k, l)

 Case 0

 Picture2.Line (i, (j + 50))-((i + 100), (j + 50))

 Case 1

 Picture2.Line (i, (j + 100))-((i + 100), j)

 Case 2

 Picture2.Line ((i + 50), j)-((i + 50), (j + 100))

 Case 3

 Picture2.Line (i, j)-((i + 100), (j + 100))

 Case 4

 Picture2.Line (i, (j + 75))-((i + 100), (j + 25))

 Case 5

 Picture2.Line ((i + 25), (j + 100))-((i + 75), j)

 Case 6

 Picture2.Line ((i + 25), j)-((i + 75), (j + 100))

 Case 7

 Picture2.Line (i, (j + 25))-((i + 100), (j + 75))

End Select

 k = k + 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Next

l = l + 1

Next

Call rec

End Sub

Sub rec()

Dim areal(8) As Integer, area2(8) As Integer, area3(8) As Integer

Dim area4(8) As Integer, area5(8) As Integer, area6(8) As Integer

Dim area7(8) As Integer, area8(8) As Integer, area9(8) As Integer

Dim maxdirec(10) As Integer, maxis(10) As Integer

Dim d As Integer, i As Integer, j As Integer

Dim comarea(10) As Integer, comdirec(10) As Integer, commaxis(10) As Integer

Dim dbase As Database

Dim dtable As Recordset

'areal

For j = 0 To 9

For i = 0 To 9

d = countdirec(i, j)

areal(d) = areal(d) + 1

Next.

Next

maxdirec(1) = areal(0)

For i = 1 To 7

If (maxdirec(1) < areal(i)) Then

maxdirec(1) = areal(i)

maxis(1) = i

End If

Next

'area2

For j = 0 To 9

For i = 6 To 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

d = countdirec(i, j)
area2(d) = area2(d) + 1
Next
Next
maxdirec(2) = area2(0)
For i = 1 To 7
    If (maxdirec(2) < area2(i)) Then
        maxdirec(2) = area2(i)
        maxis(2) = i
    End If
Next
'area3
For j = 0 To 9
    For i = 10 To 19
        d = countdirec(i, j)
        area3(d) = area3(d) + 1
    Next
Next
maxdirec(3) = area3(0)
For i = 1 To 7
    If (maxdirec(3) < area3(i)) Then
        maxdirec(3) = area3(i)
        maxis(3) = i
    End If
Next
'area4
For j = 6 To 15
    For i = 0 To 9
        d = countdirec(i, j)
        area4(d) = area4(d) + 1
    Next
Next

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

maxdirec(4) = area4(0)

For i = 1 To 7

If (maxdirec(4) < area4(i)) Then

maxdirec(4) = area4(i)

maxis(4) = i

End If

Next

'area5

For j = 6 To 15

For i = 6 To 15

d = countdirec(i, j)

area5(d) = area5(d) + 1

Next

Next

maxdirec(5) = area5(0)

For i = 1 To 7

If (maxdirec(5) < area5(i)) Then

maxdirec(5) = area5(i)

maxis(5) = i

End If

Next

'area6

For j = 6 To 15

For i = 10 To 19

d = countdirec(i, j)

area6(d) = area6(d) + 1

Next

Next

maxdirec(6) = area6(0)

For i = 1 To 7

If (maxdirec(6) < area6(i)) Then

maxdirec(6) = area6(i)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

maxis(6) = i

End If

Next

'area7

For j = 10 To 19

For i = 0 To 9

d = countdirec(i, j)

area7(d) = area7(d) + 1

Next

Next

maxdirec(7) = area7(0)

For i = 1 To 7

If (maxdirec(7) < area7(i)) Then

maxdirec(7) = area7(i)

maxis(7) = i

End If

Next

'area8

For j = 10 To 19

For i = 6 To 15

d = countdirec(i, j)

area8(d) = area8(d) + 1

Next

Next

maxdirec(8) = area8(0)

For i = 1 To 7

If (maxdirec(8) < area8(i)) Then

maxdirec(8) = area8(i)

maxis(8) = i

End If

Next

'area9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
For j = 10 To 19
```

```
  For i = 10 To 19
```

```
    d = countdirec(i, j)
```

```
    area9(d) = area9(d) + 1
```

```
  Next
```

```
Next
```

```
maxdirec(9) = area9(0)
```

```
For i = 1 To 7
```

```
  If (maxdirec(9) < area9(i)) Then
```

```
    maxdirec(9) = area9(i)
```

```
    maxis(9) = i
```

```
  End If
```

```
Next
```

```
Set dbase = OpenDatabase(App.Path + "\finger.mdb")
```

```
Set dtable = dbase.OpenRecordset("finger8", dbOpenDynaset)
```

```
If dtable.RecordCount <> 0 Then
```

```
  dtable.MoveFirst
```

```
  i = 1
```

```
  Do While dtable.EOF <> True
```

```
    comarea(i) = dtable("area")
```

```
    commaxis(i) = dtable("direc")
```

```
    comdirec(i) = dtable("maxdirec")
```

```
    i = i + 1
```

```
  dtable.MoveNext
```

```
Loop
```

```
End If
```

```
dbase.Close
```

```
'For i = 1 To 9
```

```
'Debug.Print comarea(i); maxis(i); maxdirec(i)
```

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'Debug.Print comarea(i); commaxis(i); comdirec(i)
'Next
If (commaxis(1) = maxis(1)) And (commaxis(2) = maxis(2)) And (commaxis(3) = maxis(3)) And
(commaxis(4) = maxis(4)) And (commaxis(5) = maxis(5)) And (commaxis(6) = maxis(6)) And
(commaxis(7) = maxis(7)) And (commaxis(8) = maxis(8)) And (commaxis(9) = maxis(9)) Then
If (Abs((maxdirec(1) - comdirec(1)) / comdirec(1)) <= 0.1) And (Abs((maxdirec(2) - comdirec
(2)) / comdirec(2)) <= 0.1) And (Abs((maxdirec(3) - comdirec(3)) / comdirec(3)) <= 0.1) And
(Abs((maxdirec(4) - comdirec(4)) / comdirec(4)) <= 0.1) And (Abs((maxdirec(5) - comdirec(5)) /
comdirec(5)) <= 0.1) And (Abs((maxdirec(6) - comdirec(6)) / comdirec(6)) <= 0.1) And (Abs
((maxdirec(7) - comdirec(7)) / comdirec(7)) <= 0.1) And (Abs((maxdirec(8) - comdirec(8)) /
comdirec(8)) <= 0.1) And (Abs((maxdirec(9) - comdirec(9)) / comdirec(9)) <= 0.1) Then
Form6.Show
End If
Else: Form7.Show
End If
'For i = 0 To 19
'Debug.Print dblock(0, i); dblock(1, i); dblock(2, i); dblock(3, i); dblock(4, i); dblock(5, i); dblock
(6, i); dblock(7, i); dblock(8, i); dblock(9, i); dblock(10, i); dblock(11, i); dblock(12, i); dblock
(13, i); dblock(14, i); dblock(15, i); dblock(16, i); dblock(17, i); dblock(18, i); dblock(19, i)
'Next
End Sub

```

Function countdirec(x As Integer, y As Integer)

Dim direc As Integer

Select Case dblock(x, y)

Case 0

direc = 0

Case 1

direc = 1

Case 2

direc = 2

Case 3

 direc = 3

Case 4

 direc = 4

Case 5

 direc = 5

Case 6

 direc = 6

Case 7

 direc = 7

End Select

 countdirec = direc

End Function

Private Sub cmdOpen_Click()

 CommonDialog1.CancelError = True

 On Error GoTo ErrHandler

 CommonDialog1.Flags = cdlOFNHideReadOnly

 CommonDialog1.Filter = "Picture Files (*.BMP)|*.bmp|JPEG(*.JPG)|*.jpg|All Files (*.*)|*.*"

 CommonDialog1.FilterIndex = 2

 CommonDialog1.ShowOpen

 Picture1.Picture = LoadPicture(CommonDialog1.FileName)

Exit Sub

ErrHandler:

 Exit Sub

End Sub

Private Sub Picture1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Form1.Caption = Picture1.Point(x, y)
```

```
End Sub
```

```
Function directionblock(x As Integer, y As Integer)
```

```
Dim bdirac As Single
```

```
Dim averdirac As Single
```

```
Dim arrpixdirac(1000, 1000) As Single
```

```
Dim x1 As Integer, y1 As Integer, x2 As Integer, y2 As Integer
```

```
Dim i As Integer, j As Integer, k As Integer, l As Integer
```

```
Dim ddirac(8) As Single
```

```
x1 = x
```

```
y1 = y
```

```
x2 = x + 100
```

```
y2 = y + 100
```

```
originx = 0
```

```
originy = 0
```

```
For i = 0 To 3
```

```
    ddirac(i) = 0
```

```
Next
```

```
l = 0
```

```
For j = y1 To y2 Step 10
```

```
    k = 0
```

```
    For i = x1 To x2 Step 10
```

```
        Picture2.PSet (i, j), Picture1.Point(i, j)
```

```
        arrpixdirac(k, l) = directionpixel(i, j)
```

```
        Select Case arrpixdirac(k, l)
```

```
            Case 0
```

```
                ddirac(0) = ddirac(0) + 1
```

```
            Case 1
```

```
                ddirac(1) = ddirac(1) + 1
```

```
            Case 2
```

```
                ddirac(2) = ddirac(2) + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Case 3

$$\text{ddirec}(3) = \text{ddirec}(3) + 1$$

Case 4

$$\text{ddirec}(4) = \text{ddirec}(4) + 1$$

Case 5

$$\text{ddirec}(5) = \text{ddirec}(5) + 1$$

Case 6

$$\text{ddirec}(6) = \text{ddirec}(6) + 1$$

Case 7

$$\text{ddirec}(7) = \text{ddirec}(7) + 1$$

End Select

$$k = k + 1$$

Next

$$l = l + 1$$

Next

$$\text{averdirec} = 0$$

$$k = \text{ddirec}(0)$$

For i = 1 To 7

If (k < ddirec(i)) Then

$$\text{averdirec} = i$$

$$k = \text{ddirec}(i)$$

End If

Next

$$\text{directionblock} = \text{averdirec}$$

'Debug.Print averdirec; ddirec(0); ddirec(1); ddirec(2); ddirec(3)

End Function

Function directionpixel(x As Integer, y As Integer)

Dim x1 As Integer, x2 As Integer, y1 As Integer, y2 As Integer

Dim sumdirec(16) As Single

Dim unidirec(8) As Single

Dim direc As Single, direcold As Single

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim i As Integer, j As Integer

Dim pex As Single, pext As Single

Dim dd As Integer

Dim dis As Integer

dis = 20

dd = 5

x1 = 800

y1 = 0

x2 = 2800

y2 = 2000

pex = Picture1.Point(x, y)

For i = 0 To 7

 sumdirec(i) = 0

Next

'direction 0

i = x + 1

If (i <= x2) Then

 Do While ((i <= x + 1 + dis) And (i <= x2))

 pext = Picture1.Point(i, y)

 sumdirec(0) = sumdirec(0) + Abs(pex - pext)

 i = i + dd

 Loop

Else: sumdirec(0) = 1E+24

End If

'direction 1

i = x + 1

j = y - 1

If ((i <= x2) And (j >= y1)) Then

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Do While ((i <= x + 1 + dis) And (i <= x2) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(1) = sumdirec(1) + Abs(pex - pext)

i = i + dd

j = j - dd

Loop

Else: sumdirec(1) = 1E+24

End If

'direction2

j = y - 1

If (j >= y1) Then

Do While ((j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(x, j)

sumdirec(2) = sumdirec(2) + Abs(pex - pext)

j = j - dd

Loop

Else: sumdirec(2) = 1E+24

End If

'direction3

i = x - 1

j = y - 1

'dd = 0

If ((i >= x1) And (j >= y1)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(3) = sumdirec(3) + Abs(pex - pext)

i = i - dd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

j = j - dd

Loop

Else: sumdirec(3) = 1E+24

End If

'direction4

i = x - 1

j = y + 1

If (i >= x1) Then

Do While ((i >= x - 1 - dis) And (i >= x1))

pext = Picture1.Point(i, y)

sumdirec(4) = sumdirec(4) + Abs(pex - pext)

i = i - dd

Loop

Else: sumdirec(4) = 1E+24

End If

'direction5

i = x - 1

j = y + 1

If ((i >= x1) And (j <= y2)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(5) = sumdirec(5) + Abs(pex - pext)

i = i - dd

j = j + dd

Loop

Else: sumdirec(5) = 1E+24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

'direction6

j = y + 1

If (j <= y2) Then

Do While ((j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(x, j)

sumdirec(6) = sumdirec(6) + Abs(pex - pext)

j = j + dd

Loop

Else: sumdirec(6) = 1E+24

End If

'direction7

i = x + 1

j = y + 1

If ((i <= x2) And (j <= y2)) Then

Do While ((i >= x + 1 + dis) And (i <= x2) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(7) = sumdirec(7) + Abs(pex - pext)

i = i + dd

j = j + dd

Loop

Else: sumdirec(7) = 1E+24

End If

'direction 8

i = x + 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$j = y - 1$

If $((i \leq x2) \text{ And } (j \geq y1))$ Then

Do While $((i \leq x + 1 + \text{dis}) \text{ And } (i \leq x2) \text{ And } (j \geq y - 1 - \text{dis}) \text{ And } (j \geq y1))$

$\text{pext} = \text{Picture1.Point}(i, j)$

$\text{sumdirec}(8) = \text{sumdirec}(8) + \text{Abs}(\text{pex} - \text{pext})$

$i = i + \text{dd}$

$j = j - (\text{dd} / 2)$

Loop

Else: $\text{sumdirec}(1) = 1\text{E}+24$

End If

'direction 9

$i = x + 1$

$j = y - 1$

If $((i \leq x2) \text{ And } (j \geq y1))$ Then

Do While $((i \leq x + 1 + \text{dis}) \text{ And } (i \leq x2) \text{ And } (j \geq y - 1 - \text{dis}) \text{ And } (j \geq y1))$

$\text{pext} = \text{Picture1.Point}(i, j)$

$\text{sumdirec}(9) = \text{sumdirec}(9) + \text{Abs}(\text{pex} - \text{pext})$

$i = i + (\text{dd} / 2)$

$j = j - \text{dd}$

Loop

Else: $\text{sumdirec}(1) = 1\text{E}+24$

End If

'direction10

$i = x - 1$

$j = y - 1$

If $((i \geq x1) \text{ And } (j \geq y1))$ Then

Do While $((i \geq x - 1 - \text{dis}) \text{ And } (i \geq x1) \text{ And } (j \geq y - 1 - \text{dis}) \text{ And } (j \geq y1))$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pext = Picture1.Point(i, j)
sumdirec(10) = sumdirec(10) + Abs(pex - pext)
i = i - (dd / 2)
j = j - dd
```

Loop

Else: sumdirec(3) = 1E+24

End If

'direction 11

i = x - 1

j = y - 1

If ((i >= x1) And (j >= y1)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(11) = sumdirec(11) + Abs(pex - pext)

i = i - dd

j = j - (dd / 2)

Loop

Else: sumdirec(3) = 1E+24

End If

'direction 12

i = x - 1

j = y + 1

If ((i >= x1) And (j <= y2)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(12) = sumdirec(12) + Abs(pex - pext)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Loop

Else: $\text{sumdirec}(7) = 1E+24$

End If

'direction 15

$i = x + 1$

$j = y + 1$

If $((i \leq x2) \text{ And } (j \leq y2))$ Then

Do While $((i \geq x + 1 + \text{dis}) \text{ And } (i \leq x2) \text{ And } (j \leq y + 1 + \text{dis}) \text{ And } (j \leq y2))$

$\text{pext} = \text{Picture1.Point}(i, j)$

$\text{sumdirec}(15) = \text{sumdirec}(15) + \text{Abs}(\text{pex} - \text{pext})$

$i = i + \text{dd}$

$j = j + (\text{dd} / 2)$

Loop

Else: $\text{sumdirec}(7) = 1E+24$

End If

$\text{unidirec}(0) = \text{sumdirec}(0) + \text{sumdirec}(4)$

$\text{unidirec}(1) = \text{sumdirec}(1) + \text{sumdirec}(5)$

$\text{unidirec}(2) = \text{sumdirec}(2) + \text{sumdirec}(6)$

$\text{unidirec}(3) = \text{sumdirec}(3) + \text{sumdirec}(7)$

$\text{unidirec}(4) = \text{sumdirec}(8) + \text{sumdirec}(12)$

$\text{unidirec}(5) = \text{sumdirec}(9) + \text{sumdirec}(13)$

$\text{unidirec}(6) = \text{sumdirec}(10) + \text{sumdirec}(14)$

$\text{unidirec}(7) = \text{sumdirec}(11) + \text{sumdirec}(15)$

$\text{direcold} = \text{unidirec}(0)$

$\text{direc} = 0$

For $i = 0$ To 7

If $(\text{direcold} > \text{unidirec}(i))$ Then

$\text{direcold} = \text{unidirec}(i)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

direc = i

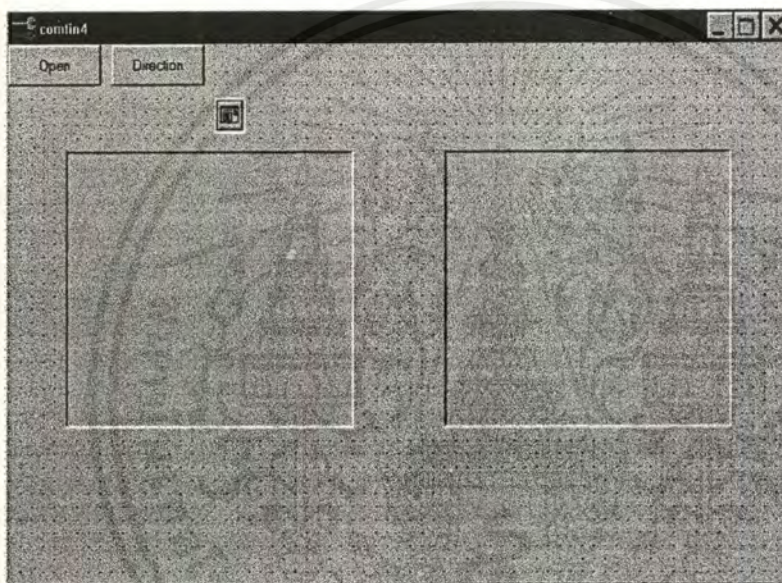
End If

Next

directionpixel = direc

End Function

5. Form5 เป็น Form เปรียบเทียบข้อมูล 4 ทิศทาง



SOURCE CODE

Option Explicit

Dim dblock(50, 50) As Single

Dim originx As Integer

Dim originy As Integer

Sub direction()

Dim i As Integer, j As Integer, l As Integer, k As Integer

Dim xPixels As Double

Dim yPixels As Double

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

· xPixels = 2000 - 1
yPixels = 2000 - 1
l = 0
Picture2.BackColor = QBColor(15)
For j = 0 To yPixels Step 100
    k = 0
    For i = 0 To xPixels Step 100
        dblock(k, l) = directionblock(i, j)
        Select Case dblock(k, l)
        Case 0
            Picture2.Line (i, (j + 50))-((i + 100), (j + 50))
        Case 1
            Picture2.Line (i, (j + 100))-((i + 100), j)
        Case 2
            Picture2.Line ((i + 50), j)-((i + 50), (j + 100))
        Case 3
            Picture2.Line (i, j)-((i + 100), (j + 100))
        End Select
        k = k + 1
    Next
    l = l + 1
Next
Call rec
End Sub
Sub rec()
Dim area1(8) As Integer, area2(8) As Integer, area3(8) As Integer
Dim area4(8) As Integer, area5(8) As Integer, area6(8) As Integer
Dim area7(8) As Integer, area8(8) As Integer, area9(8) As Integer
Dim maxdirec(10) As Integer, maxis(10) As Integer
Dim d As Integer, i As Integer, j As Integer
Dim comarea(10) As Integer, comdirec(10) As Integer, commaxis(10) As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.Dim dbase As Database

Dim dtable As Recordset

'area1

For j = 0 To 9

For i = 0 To 9

d = countdirec(i, j)

area1(d) = area1(d) + 1

Next

Next

maxdirec(1) = area1(0)

For i = 1 To 7

If (maxdirec(1) < area1(i)) Then

maxdirec(1) = area1(i)

maxis(1) = i

End If

Next

'area2

For j = 0 To 9

For i = 6 To 15

d = countdirec(i, j)

area2(d) = area2(d) + 1

Next

Next

maxdirec(2) = area2(0)

For i = 1 To 7

If (maxdirec(2) < area2(i)) Then

maxdirec(2) = area2(i)

maxis(2) = i

End If

Next

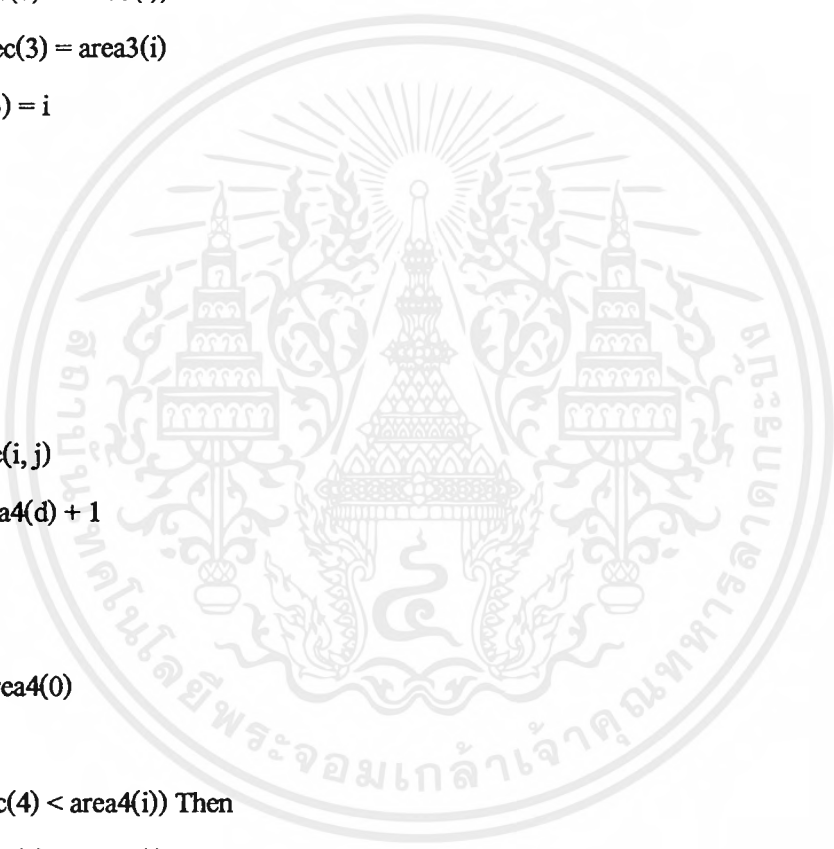
'area3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For j = 0 To 9
  For i = 10 To 19
    d = countdirec(i, j)
    area3(d) = area3(d) + 1
  Next
Next
maxdirec(3) = area3(0)
For i = 1 To 7
  If (maxdirec(3) < area3(i)) Then
    maxdirec(3) = area3(i)
    maxis(3) = i
  End If
Next
'area4
For j = 6 To 15
  For i = 0 To 9
    d = countdirec(i, j)
    area4(d) = area4(d) + 1
  Next
Next
maxdirec(4) = area4(0)
For i = 1 To 7
  If (maxdirec(4) < area4(i)) Then
    maxdirec(4) = area4(i)
    maxis(4) = i
  End If
Next
'area5
For j = 6 To 15
  For i = 6 To 15
    d = countdirec(i, j)
    area5(d) = area5(d) + 1

```



Next

Next

maxdirec(5) = area5(0)

For i = 1 To 7

If (maxdirec(5) < area5(i)) Then

maxdirec(5) = area5(i)

maxis(5) = i

End If

Next

'area6

For j = 6 To 15

For i = 10 To 19

d = countdirec(i, j)

area6(d) = area6(d) + 1

Next

Next

maxdirec(6) = area6(0)

For i = 1 To 7

If (maxdirec(6) < area6(i)) Then

maxdirec(6) = area6(i)

maxis(6) = i

End If

Next

'area7

For j = 10 To 19

For i = 0 To 9

d = countdirec(i, j)

area7(d) = area7(d) + 1

Next

Next

maxdirec(7) = area7(0)

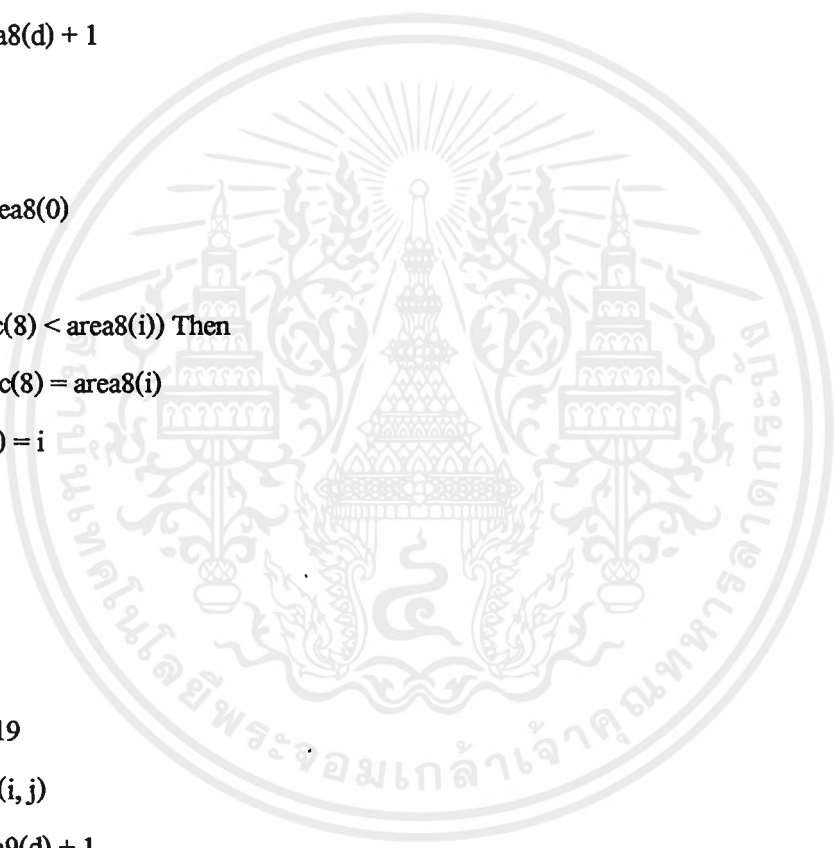
For i = 1 To 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If (maxdirec(7) < area7(i)) Then
    maxdirec(7) = area7(i)
    maxis(7) = i
End If
Next
'area8
For j = 10 To 19
    For i = 6 To 15
        d = countdirec(i, j)
        area8(d) = area8(d) + 1
    Next
Next
maxdirec(8) = area8(0)
For i = 1 To 7
    If (maxdirec(8) < area8(i)) Then
        maxdirec(8) = area8(i)
        maxis(8) = i
    End If
Next
'area9
For j = 10 To 19
    For i = 10 To 19
        d = countdirec(i, j)
        area9(d) = area9(d) + 1
    Next
Next
maxdirec(9) = area9(0)
For i = 1 To 7
    If (maxdirec(9) < area9(i)) Then
        maxdirec(9) = area9(i)
        maxis(9) = i
    End If

```



Next

```
Set dbase = OpenDatabase(App.Path + "\finger.mdb")
```

```
Set dtable = dbase.OpenRecordset("finger4", dbOpenDynaset)
```

```
If dtable.RecordCount <> 0 Then
```

```
    dtable.MoveFirst
```

```
    i = 1
```

```
    Do While dtable.EOF <> True
```

```
        comarea(i) = dtable("area")
```

```
        commaxis(i) = dtable("direc")
```

```
        comdirec(i) = dtable("maxdirec")
```

```
        i = i + 1
```

```
    dtable.MoveNext
```

```
    Loop
```

```
End If
```

```
dbase.Close
```

```
For i = 1 To 9
```

```
    Debug.Print comarea(i); maxis(i); maxdirec(i); commaxis(i); comdirec(i)
```

```
Next
```

```
If (commaxis(1) = maxis(1)) And (commaxis(2) = maxis(2)) And (commaxis(3) = maxis(3)) And
```

```
(commaxis(4) = maxis(4)) And (commaxis(5) = maxis(5)) And (commaxis(6) = maxis(6)) And
```

```
(commaxis(7) = maxis(7)) And (commaxis(8) = maxis(8)) And (commaxis(9) = maxis(9)) Then
```

```
If (Abs(maxdirec(1) - comdirec(1)) <= 10) And (Abs(maxdirec(2) - comdirec(2)) <= 10) And
```

```
(Abs(maxdirec(3) - comdirec(3)) <= 10) And (Abs(maxdirec(4) - comdirec(4)) <= 10) And (Abs
```

```
(maxdirec(5) - comdirec(5)) <= 10) And (Abs(maxdirec(6) - comdirec(6)) <= 10) And (Abs
```

```
(maxdirec(7) - comdirec(7)) <= 10) And (Abs(maxdirec(8) - comdirec(8)) <= 10) And (Abs
```

```
(maxdirec(9) - comdirec(9)) <= 10) Then
```

```
'Label1.Caption = "ผ่าน"
```

```
Form6.Show
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Else:

'Label1.Caption = "ไม่ผ่าน"

Form7.Show

End If

Else:

'Label1.Caption = "ไม่ผ่าน"

Form7.Show

End If

End Sub

Function countdirec(x As Integer, y As Integer)

Dim direc As Integer

Select Case dblock(x, y)

Case 0

 direc = 0

Case 1

 direc = 1

Case 2

 direc = 2

Case 3

 direc = 3

End Select

countdirec = direc

End Function

Private Sub cmdDirection_Click()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim i As Integer, j As Integer, l As Integer, k As Integer

Dim xPixels As Double

Dim yPixels As Double

xPixels = 2800 - 1

yPixels = 2000 - 1

l = 0

Picture2.BackColor = QBColor(15)

For j = 0 To yPixels Step 100

 k = 0

 For i = 800 To xPixels Step 100

 dblock(k, l) = directionblock(i, j)

 Select Case dblock(k, l)

 Case 0

 Picture2.Line (i, (j + 50))-((i + 100), (j + 50))

 Case 1

 Picture2.Line (i, (j + 100))-((i + 100), j)

 Case 2

 Picture2.Line ((i + 50), j)-((i + 50), (j + 100))

 Case 3

 Picture2.Line (i, j)-((i + 100), (j + 100))

 End Select

 k = k + 1

 Next

 l = l + 1

Next

Call rec

End Sub

Private Sub cmdOpen_Click()

 CommonDialog1.CancelError = True

 On Error GoTo ErrorHandler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CommonDialog1.Flags = cdlOFNHideReadOnly
CommonDialog1.Filter = "Picture Files (*.BMP)|*.bmp|JPEG(*.JPG)|*.jpg|All Files (*.*)|*.*"
CommonDialog1.FilterIndex = 2
CommonDialog1.ShowOpen
Picture1.Picture = LoadPicture(CommonDialog1.FileName)
Exit Sub
```

ErrorHandler:

```
Exit Sub
```

End Sub

```
Function directionblock(x As Integer, y As Integer)
Dim bdirec As Single
Dim averdirec As Single
Dim arrpixdirec(1000, 1000) As Single
Dim x1 As Integer, y1 As Integer, x2 As Integer, y2 As Integer
Dim i As Integer, j As Integer, k As Integer, l As Integer
Dim ddirec(8) As Single
```

```
x1 = x
```

```
y1 = y .
```

```
x2 = x + 100
```

```
y2 = y + 100
```

```
originx = 0
```

```
originy = 0
```

```
For i = 0 To 3
```

```
    ddirec(i) = 0
```

```
Next
```

```
l = 0
```

```
For j = y1 To y2 Step 10
```

```
    k = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For i = x1 To x2 Step 10

 arrpixdirec(k, l) = directionpixel(i, j)

 Select Case arrpixdirec(k, l)

 Case 0

 ddirec(0) = ddirec(0) + 1

 Case 1

 ddirec(1) = ddirec(1) + 1

 Case 2

 ddirec(2) = ddirec(2) + 1

 Case 3

 ddirec(3) = ddirec(3) + 1

 End Select

 k = k + 1

Next

l = l + 1

Next

averdirec = 0

k = ddirec(0)

For i = 0 To 3

 If (k < ddirec(i)) Then

 averdirec = i

 k = ddirec(i)

 End If

Next

directionblock = averdirec

'Debug.Print averdirec; ddirec(0); ddirec(1); ddirec(2); ddirec(3)

End Function

Function directionpixel(x As Integer, y As Integer)

Dim x1 As Integer, x2 As Integer, y1 As Integer, y2 As Integer

Dim sumdirec(16) As Single

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim unidirec(8) As Single

Dim direc As Single, direcold As Single

Dim i As Integer, j As Integer

Dim pex As Single, pext As Single

Dim dd As Integer

Dim dis As Integer

dis = 20

dd = 5

x1 = 800

y1 = 0

x2 = 2800

y2 = 2000

pex = Picture1.Point(x, y)

For i = 0 To 7

 sumdirec(i) = 0

Next

'direction 0

i = 0

i = x + 1

If (i <= x2) Then

 Do While ((i <= x + 1 + dis) And (i <= x2))

 pext = Picture1.Point(i, y)

 sumdirec(0) = sumdirec(0) + Abs(pex - pext)

 i = i + dd

 Loop

Else: sumdirec(0) = 1E+24

End If

'direction 1

i = j = 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$i = x + 1$

$j = y - 1$

If (($i \leq x2$) And ($j \geq y1$)) Then

Do While (($i \leq x + 1 + dis$) And ($i \leq x2$) And ($j \geq y - 1 - dis$) And ($j \geq y1$))

$pext = \text{Picture1.Point}(i, j)$

$\text{sumdirec}(1) = \text{sumdirec}(1) + \text{Abs}(pex - pext)$

$i = i + dd$

$j = j - dd$

Loop

Else: $\text{sumdirec}(1) = 1E+24$

End If

'direction2

$j = 0$

$j = y - 1$

If ($j \geq y1$) Then

Do While (($j \geq y - 1 - dis$) And ($j \geq y1$))

$pext = \text{Picture1.Point}(x, j)$

$\text{sumdirec}(2) = \text{sumdirec}(2) + \text{Abs}(pex - pext)$

$j = j - dd$

Loop

Else: $\text{sumdirec}(2) = 1E+24$

End If

'direction3

$i = j = 0$

$i = x - 1$

$j = y - 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

If ((i >= x1) And (j >= y1)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j >= y - 1 - dis) And (j >= y1))

pext = Picture1.Point(i, j)

sumdirec(3) = sumdirec(3) + Abs(pex - pext)

i = i - dd

j = j - dd

Loop

Else: sumdirec(3) = 1E+24

End If

'direction4

i = 0

i = x - 1

If (i >= x1) Then

Do While ((i >= x - 1 - dis) And (i >= x1))

pext = Picture1.Point(i, y)

sumdirec(4) = sumdirec(4) + Abs(pex - pext)

i = i - dd

Loop

Else: sumdirec(4) = 1E+24

End If

'direction5

i = j = 0

i = x - 1

j = y + 1

If ((i >= x1) And (j <= y2)) Then

Do While ((i >= x - 1 - dis) And (i >= x1) And (j <= y + 1 + dis) And (j <= y2))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pext = Picture1.Point(i, j)
sumdirec(5) = sumdirec(5) + Abs(pex - pext)
i = i - dd
j = j + dd
```

Loop

Else: sumdirec(5) = 1E+24

End If

'direction6

j = 0

j = y + 1

If (j <= y2) Then

Do While ((j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(x, j)

sumdirec(6) = sumdirec(6) + Abs(pex - pext)

j = j + dd

Loop

Else: sumdirec(6) = 1E+24

End If

'direction7

i = j = 0

i = x + 1

j = y + 1

If ((i <= x2) And (j <= y2)) Then

Do While ((i >= x + 1 + dis) And (i <= x2) And (j <= y + 1 + dis) And (j <= y2))

pext = Picture1.Point(i, j)

sumdirec(7) = sumdirec(7) + Abs(pex - pext)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$i = i + dd$

$j = j + dd$

Loop

Else: $\text{sumdirec}(7) = 1E+24$

End If

$\text{unidirec}(0) = \text{sumdirec}(0) + \text{sumdirec}(4)$

$\text{unidirec}(1) = \text{sumdirec}(1) + \text{sumdirec}(5)$

$\text{unidirec}(2) = \text{sumdirec}(2) + \text{sumdirec}(6)$

$\text{unidirec}(3) = \text{sumdirec}(3) + \text{sumdirec}(7)$

$\text{direcold} = \text{unidirec}(0)$

$\text{direc} = 0$

For $i = 0$ To 3

If ($\text{direcold} > \text{unidirec}(i)$) Then

$\text{direcold} = \text{unidirec}(i)$

$\text{direc} = i$

End If

Next

$\text{directionpixel} = \text{direc}$

End Function

6. Form5 เป็น Form About



SOURCE CODE

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
End Sub
```

7. Form6 เป็น Form แสดงผลเมื่อการเปรียบเทียบเป็นจริง



SOURCE CODE

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim i As Integer
```

```
MSComm1.CommPort = 2
```

```
MSComm1.Settings = "9600,N,8,1"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

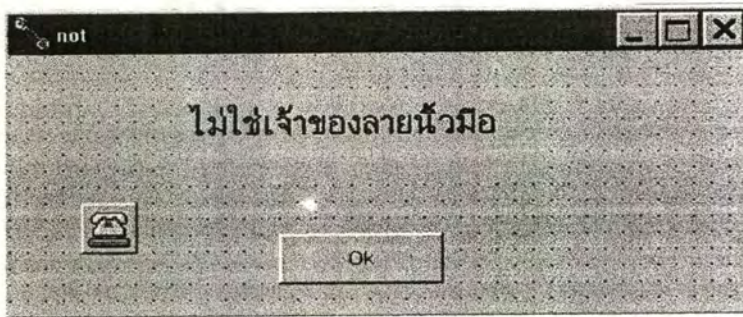
```
MSComm1.InputLen = 0
MSComm1.PortOpen = True
MSComm1.Output = "+"
MSComm1.PortOpen = False
```

```
If i <> 5 Then
    Call Timer1_Timer
End If
End Sub
```

```
Private Sub Timer1_Timer()
    Static Stage
    Stage = Not Stage
    If Stage = True Then
        Shape1.FillStyle = 0
    End If
    If Stage = False Then
        Shape1.FillStyle = 1
    End If
End Sub
```



8. Form7 เป็น Form แสดงผลเมื่อผลการเปรียบเทียบไม่เป็นจริง



SOURCE CODE

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
MSComm1.CommPort = 2
```

```
MSComm1.Settings = "9600,N,8,1"
```

```
MSComm1.InputLen = 0
```

```
MSComm1.PortOpen = True
```

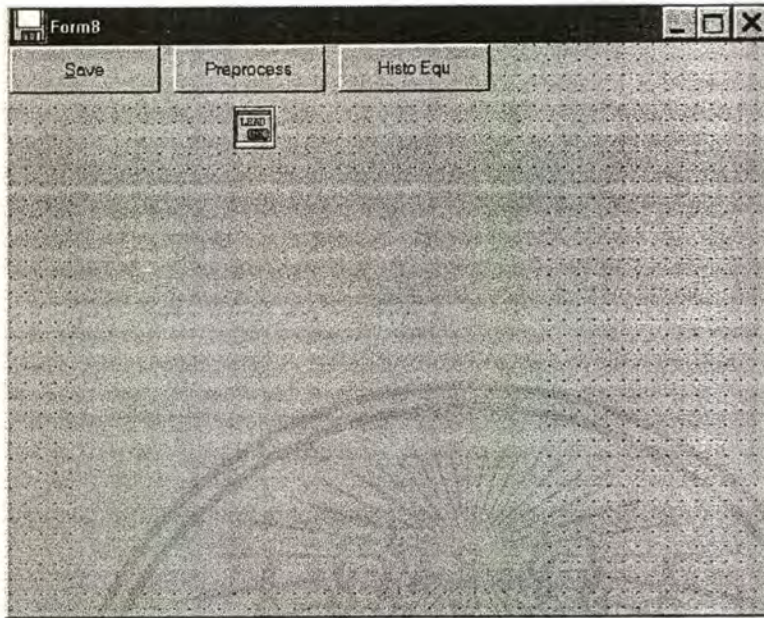
```
MSComm1.Output = "p"
```

```
MSComm1.PortOpen = False
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Form8 เป็น Form บันทึกภาพที่ได้จากกล้อง



SOURCE CODE

```
Private Sub Command1_Click()
```

```
Dim i As Integer
```

```
Dim fUserOK As Boolean
```

```
Dim nBitsToSave As Integer
```

```
Dim nFormat As Integer
```

```
Dim nQFactor As Integer
```

```
Dim nRet As Integer
```

```
Dim fStamp As Boolean
```

```
Dim nStampWidth As Integer
```

```
Dim nStampHeight As Integer
```

```
Dim nStampBits As Integer
```

```
Static First As Integer
```

```
Static SaveIdx As Integer
```

```
Static Filename As String
```

```
Static fMultipage As Boolean
```

```
Dim msg
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim style

Dim Title

If First = 0 Then

First = 1

Filename = Command

fMultipage = False

SaveIdx = 0

End If

LEADDlg1.Filter = "All Files|*.*"

LEADDlg1.FileDlgFlags = cdlOFNFileMustExist

LEADDlg1.DialogTitle = "Save File"

LEADDlg1.UIFlags = DLG_FS_95STYLE + DLG_FS_STAMP + DLG_FS_MULTIPAGE +
DLG_FS_QFACTOR

LEADDlg1.SaveFormatFlags = DLG_FS_ALL

On Error GoTo SaveError

LEADDlg1.FileSave hWnd

Filename = LEADDlg1.Filename

If Filename <> "" Then

fStamp = LEADDlg1.SaveWithStamp

fMultipage = LEADDlg1.SaveMulti

nFormat = LEADDlg1.SaveFormat

nQFactor = LEADDlg1.SaveQFactor

nBitsToSave = LEADDlg1.SaveBitsPerPixel

If fStamp Then

nStampWidth = LEADDlg1.SaveStampWidth

nStampHeight = LEADDlg1.SaveStampWidth

nStampBits = LEADDlg1.SaveStampBits

nRet = Form8.LEAD1.SaveWithStamp(Filename, nFormat, nBitsToSave, nQFactor,

nStampWidth, nStampHeight, nStampBits)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Else

nRet = Form8.LEAD1.Save(Filename, nFormat, nBitsToSave, nQFactor, fMultipage)

End If

MOUSEDEFAULT

If nRet <> 0 Then

msg = "Error " & CStr(nRet) & ""

style = vbOKOnly

Title = "SaveFile Error"

MsgBox msg, style, Title

End If

End If

Exit Sub

SaveError:

If Err <> ERROR_DLG_CANCELED Then

MOUSEDEFAULT

msg = "VB Error: " & CStr(Err.Number) & " - " & Err.Description & "."

style = vbOKOnly

Title = "SaveFile Error"

MsgBox msg, style, Title

End If

End Sub

Private Sub Command2_Click()

LEADDlg1.Change = 100

LEADDlg1.UIFlags = DLG_IMG_SHARPEN + DLG_IMG_SHOWPREVIEW

LEADDlg1.Bitmap = Form8.LEAD1.Bitmap

LEADDlg1.GetChange hWnd

LEADDlg1.Bitmap = 0

StatusInit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form8.LEAD1.Sharpen LEADDlg1.Change

StatusTerm

Exit Sub

End Sub

Private Sub Command3_Click()

StatusInit

Form8.LEAD1.HistoEqualize

StatusTerm

End Sub

Private Sub Form_Load()

Dim nRet As Integer

Dim msg

Dim style

Dim Title

nRet = LEAD1.TwainAcquire(hWnd)

If nRet <> 0 Then

msg = "Error " & CStr(nRet) & ", " & Main.DecodeError(nRet) & " was generated."

style = vbOKOnly

Title = "FileInfo Error"

MsgBox msg, style, Title

End If

End Sub

Private Sub MOUSEDEFAULT()

Screen.MousePointer = 0

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Public Sub StatusInit()

MOUSEWAIT

Form8.LEAD1.AutoRepaint = False

Form8.LEAD1.EnableProgressEvent = gmStatusEvent

gInProcess = True

End Sub

Public Sub StatusTerm()

MOUSEDEFAULT

gInProcess = False

Form8.SetFocus

Form8.LEAD1.AutoRepaint = True

End Sub

Private Sub MOUSEWAIT()

Screen.MousePointer = 11

End Sub

10. Module1

Global Const RESIZE_TYPE = 0

Global Const RESIZEREGION_TYPE = 1

Global Const PAINT_AUTO = 0

Global Const PAINT_FIXED = 1

Global Const PAINT_NETSCAPE = 2

Global Const SO_LEAD = 0

Global Const SO_JFIF = 1

Global Const SO_JTIF = 2

Global Const SO_AWD = 3

Global Const SO_CALS = 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Global Const SO_CUR = 5
Global Const SO_CCITT = 6
Global Const SO_DIC_GRAY = 7
Global Const SO_DIC_COLOR = 8
Global Const SO_EXIF = 9
Global Const SO_FAX = 10
Global Const SO_EPS = 11
Global Const SO_FPX = 12
Global Const SO_GEM = 13
Global Const SO_GIF = 14
Global Const SO_ICO = 15
Global Const SO_IOCA = 16
Global Const SO_PCT = 17
Global Const SO_MAC = 18
Global Const SO_MSP = 19
Global Const SO_OS2 = 20
Global Const SO_PCX = 21
Global Const SO_PNG = 22
Global Const SO_PSD = 23
Global Const SO_RAS = 24
Global Const SO_TGA = 25
Global Const SO_TIF = 26
Global Const SO_WBMP = 27
Global Const SO_WBMP_RLE = 28
Global Const SO_WFX = 29
Global Const SO_WMF = 30
Global Const SO_WPG = 31
Global Const NUM_SAVE_TYPES = 32
Global Const NUM_OPEN_TYPES = 23

Global Const QF_CUSTOM = 9

Global gDitheringType As Integer

Global gBitonalScaling As Integer

Global gPaintScaling As Integer

Global gPalette As Integer

Global gUseNetscape As Boolean

Global gNumChildren As Integer

Global gInProcess As Integer

Global gEndApp As Integer

Global gLoadRepaint As Boolean

Public Const BITSPIXEL = 12

Public Const PLANES = 14

Global Const IDM_TOOLNONE = 0

Global Const IDM_TOOLRECT = 1

Global Const IDM_TOOLELLIPSE = 2

Global Const IDM_TOOLRNDRECT = 3

Global Const IDM_TOOLFREEHAND = 4

Global Kids() As Variant

#If Win32 Then

Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hWnd As Long, ByVal lpString As String, ByVal aint As Long) As Long

Declare Function GetDeviceCaps Lib "gdi32" (ByVal hdc As Long, ByVal nIndex As Long) As Long

Declare Function GetDC Lib "user32" (ByVal hWnd As Long) As Long

Declare Function ReleaseDC Lib "user32" (ByVal hWnd As Long, ByVal hdc As Long) As Long

#Else

Declare Function GetWindowText Lib "User" (ByVal hWnd As Integer, ByVal lpString As String, ByVal aint As Integer) As Integer

```
Declare Function GetDeviceCaps Lib "GDI" (ByVal hdc As Integer, ByVal nIndex As Integer)
As Integer
Declare Function GetDC Lib "User" (ByVal hWnd As Integer) As Integer
Declare Function ReleaseDC Lib "User" (ByVal hWnd As Integer, ByVal hdc As Integer) As
Integer
#End If
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

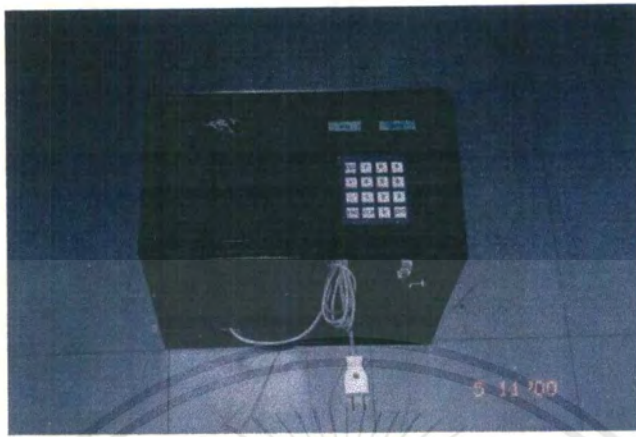
ภาคผนวก ข
แสดงรูปอุปกรณ์ส่วน Hardware



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รูปถ่าย Hardware ที่ใช้ในการทดลองในปฏิญานิพนธ์นี้

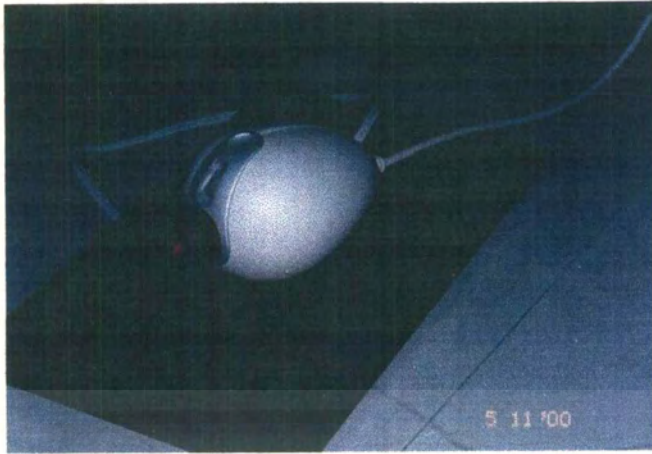


รูปที่ ข.1 แสดงรูปด้านบนของอุปกรณ์



รูปที่ ข.2 แสดงรูปด้านหน้าของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.3 แสดงรูปกล้องที่ใช้ในการทดลอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

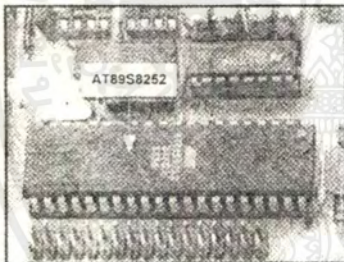
ภาคผนวก ค
แสดงรายละเอียดของบอร์ดควบคุม CP-S8252



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ CP-S53

- MCS-51 #AT89S53 Microcontroller
- On-Board Down Load (ET-SPI)
- 4 Bit LCD(Character) Interface
- 12 Bit 2 Channel Analog to Digital
- RTC #DS1307
- Eeprom #24XX Interface
- RS232 Communication
- RS422/RS485 Communication
- 72IOZ80 ETT Interface
- EXP 20 PIN Emulate (AT89C1051, AT89C2051, AT89C4051)
- 5VDC Rectifier

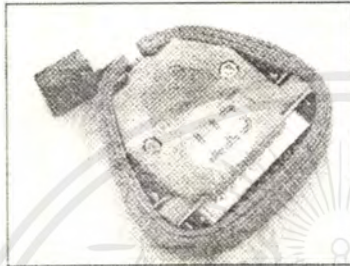
MICROCONTROLLER UNIT

- Nine Interrupt Sources
- SPI Serial Interface
- Programmable Watchdog Timer
- Dual Data Pointer
- 2 K Byte EEPROM (100,000 Write/Erase Cycle by Atmel Specification)
- Low Power Idel and Power Down Mode
- Interrupt Recovery From Power Down
- Power Off Flag
- Programmable UART Serial Channel
- AT89S8252 Microcontroller
- Compatible with MCS-51 Product
- 8 K Byte of In-System Reprogrammable Downloadable Flash Memory (1,000 Write/Erase Cycles by Atmel Specification)
- Full Static Operation : 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 Byte Internal RAM
- 32 Programmable I/O Line
- Three 16-bit Timer/Counter

● ET-SPI UNIT

หน่วย ET-SPI ถูกออกแบบให้มีหน้าที่หลักเป็นหน่วย Down Load ข้อมูลลงในหน่วยความจำของ Microcontroller(AT89S8252) ซึ่งจะทำให้การพัฒนาโปรแกรมบน CP-S8252 มี

ความสะดวกและประหยัดค่าใช้จ่ายโดยไม่ต้องมีอุปกรณ์อื่นๆ มาช่วยพัฒนาโปรแกรม อีกทั้งยังสามารถใช้ CP-S8252 เป็นเครื่องมือโปรแกรม AT89S8252 ตัวอื่นๆ ได้อีก

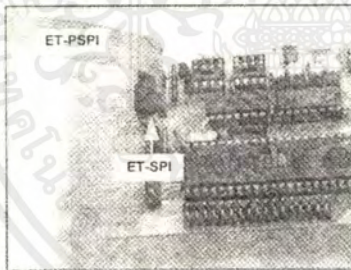


ส่วนประกอบของหน่วย ET-SPI ถูกแบ่งออกเป็น 2 ส่วนคือ ส่วนของ Hard Ware และ ส่วนของ Soft Ware

จากรูปแสดงสาย ET-PSPI ซึ่งต้องใช้ Down Load ข้อมูลจาก PC ผ่านทาง Printer Port ลงในหน่วยความจำของ Microcontroller

ขั้นตอนการติดตั้ง Hard Ware ในการ โหลดข้อมูล

1. ต่อสาย Down Load (ET-PSPI) เข้ากับ Connector 10 pin (ET-SPI) โดยต่อสายให้ถูกต้อง ตามตำแหน่งขาของ Connector
2. ต่อสาย Down Load อีกด้านหนึ่งของ ET-PSPI เข้ากับ Printer Port ของ PC
3. ต่อ Power Supply เข้ากับ Board CP-S8252
4. เมื่อทุกอย่างไม่มีปัญหาพร้อมที่จะทำในขั้นตอนของ Soft Ware ต่อไป



ขั้นตอนการติดตั้ง Soft Ware

ระบบ Soft Ware ที่ใช้งานกับ CP-S8252 สามารถใช้งานได้ทั้งระบบปฏิบัติการ DOS และ WINDOWS โดย Soft Ware ที่ทำงานภายใต้ DOS และ WINDOWS สามารถปฏิบัติงานได้ทัดเทียมกัน ขึ้นอยู่กับความถนัดของผู้ใช้งาน

DOS Operating System

Soft Ware ที่ทำงานบนระบบปฏิบัติการ DOS จะทำงานแบบ Command Line ซึ่งทำให้การใช้งานง่ายและสามารถทำเป็น FILE.BAT เพื่อให้การใช้งานสะดวกมากขึ้น ซึ่งมีรูปแบบของคำสั่งดังต่อไปนี้

S8252 <Filename> <Option> ↵

Filename :

คือ File ที่ต้องการ Load ลงในหน่วยความจำของ Microcontroller โดยจะต้องเป็น File ที่ได้รับการแปลงออกมาในรูปของ Intel Hex เรียบร้อยแล้ว เช่น LCD.HEX ซึ่งเป็น File ที่อยู่ในโปรแกรมตัวอย่าง

Option :

คือ ส่วนที่ใช้เลือกคุณสมบัติต่างๆของโปรแกรมให้ตรงหรือเหมาะสมกับความต้องการของผู้ใช้งาน

A = LPT1 (Down Load โปรแกรมผ่าน LPT1)

B = LPT2 (Down Load โปรแกรมผ่าน LPT2)

1 - 9 = ค่า Delay time (ใส่ก็ต่อเมื่อ Computer มีความเร็วมากจนไม่สามารถ Down Load ข้อมูลลงใน CPU ได้)

S = Security (ป้องกันการ COPY ข้อมูลจาก Microcontroller)

Comment : ถ้าไม่ใส่ OPTION ใดๆ โปรแกรมจะกำหนดให้เป็น LPT1,DELAY = 1, NO SECURITY

WINDOWS Operating System

โปรแกรมที่ใช้งานบนระบบปฏิบัติการ WINDOW ชื่อว่า WINS8252.EXE ซึ่งสามารถทำงานได้กับระบบปฏิบัติการ WINDOWS3.1 WINDOWS95 ,WINDOWS98 โดยมีรายละเอียดการใช้งานดังต่อไปนี้

ขั้นตอนการใช้งาน

1. เตรียมระบบโดยการต่อสายและจ่ายไฟให้กับ CP-S8252
2. เรียก WINS8252 .EXE ใน WINDOWS
3. เลือก PROGRAM จาก MEMU BAR
4. ใส่ชื่อ FILE.HEX ที่ต้องการจะ DOWN LOAD โดย FILE จะต้องมี FORMAT เป็น INTEL HEX
5. กดปุ่ม OK ใน PROGRAM DIALOGBOX



MENU ERASE : เป็น MENU ที่ใช้ ERASE ข้อมูลใน Microcontroller โดยไม่มีเขียนข้อมูลใดๆ ลงในหน่วยความจำ

MENU SECURITY : เป็น MENU ที่ใช้ทำ SECURITY เพื่อ LOCK ข้อมูลที่อยู่ใน Microcontroller เพื่อไม่ให้ถูกอ่านออกมาอีก

MENU OPTION : เป็น MENU ที่ใช้เลือก LPT (LPT1 or LPT2) ที่ใช้ Down Load ข้อมูล และใช้เลือกค่า Delay ในกรณีที่เครื่อง PC มีความเร็วมากจนไม่สามารถ Down Load ข้อมูลลงในหน่วยความจำของ Microcontroller ได้

ทั้ง PROGRAM ที่ทำงานบน DOS และ WINDOWS ของ CP-S8252 จะทำ AUTO ERASE ทุกครั้งที่เขียนข้อมูลลงใน Microcontroller แต่จะไม่ทำ SECURITY

● RTC UNIT (OPTION)

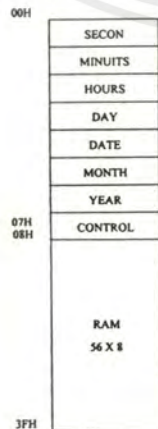
RTC UNIT เป็นระบบเวลาจริง ที่สามารถต่อเพิ่มเติม(OPTION) ลงบน CP-S8252 ได้ ซึ่งบนบอร์ด CP-S8252 ได้จัดเตรียม SOCKET เพื่อเพิ่ม RTC ไว้ให้เรียบร้อยแล้ว

ระบบ RTC บน CP-S8252 ถูกออกแบบให้ใช้ IC #DS1307 ของ DALLAS ซึ่งการติดตั้งในส่วนที่เป็น HARD WARE ของ RTC UNIT เพียงแต่ใส่ DS1307 ลงใน SOCKET ที่จัดเตรียมไว้บน CP-S8252 ดังแสดงในรูปก็เป็นอันเสร็จสิ้น ซึ่งจะเหลือเพียงขั้นตอนการเขียนโปรแกรมเพื่อควบคุมการทำงานของ DS1307 อุปกรณ์ที่ใช้ร่วมกับ DS1307 คือ XTAL ค่าความถี่ 13.768 KHz ซึ่งบนบอร์ด CP-S8252 ได้จัดเตรียมไว้ให้แล้ว



รายละเอียดของ DS1307 และหลักการเขียนโปรแกรม

● RTC and RAM Address Map



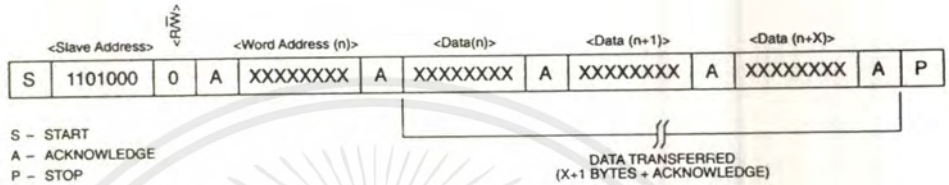
ส่วนวงจรของ CP-S8252 (ดูวงจรท้ายเล่ม) ขา SDA ซึ่งเป็นขา DATA ของ DS1307 จะต่อเข้ากับ ขา P2.6 ของ AT89S8252 และขา SCL ซึ่งเป็นขา CLOCK ของ DS1307 จะถูกต่อเข้ากับ P2.5 ของ AT89S8252 ซึ่งระบบดังกล่าว เรียกว่า I²C BUS โดยเมื่อใช้ DS1307 จะต้องมี BAT 3V ในการทำให้นาฬิกาทำงาน อยู่ตลอดเวลาเมื่อไม่มีไฟเลี้ยงวงจร

ในการติดตั้ง DS1307 ลงบน CP-S8252 จะทำให้ DS1307 มี ADDRESS อยู่ที่ 68H ซึ่งหมายความว่า การเขียนโปรแกรมควบคุมการทำงานของ DS1307 จะต้องเขียนไปที่ตำแหน่ง ADDRESS 86H เท่านั้น และในส่วนของ ขา SQW ของ DS1307 จะเป็นขาที่สามารถสร้างสัญญาณ SQUARE WAVE ออกมาตามที่โปรแกรมสั่งงาน DS1307 แต่ บนบอร์ด CP-S8252 ไม่ได้ต่อขานี้ออกมาให้หากผู้ใช้ต้องการใช้งานขา SQW ของ DS1307 จะต้องจัดการต่อขานี้ออกมาใช้งานเอง

ตำแหน่ง Address ของ RTC และ RAM จะแสดงในรูป โดย Address ของค่าเวลากายใน (SECON, MINUITE, HOURS, DAY, DATE, MONTH, YEAR, CONTROL) ของ RTC จะอยู่ที่ Address 00H - 07H และตำแหน่ง Address ของ Ram จะอยู่ที่ Address 08H - 3FH

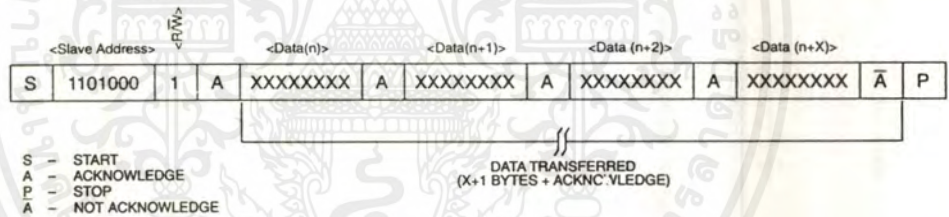
• การเขียนข้อมูลลง DS1307 (DATA WRITE)

การเขียนข้อมูลลงใน DS1307 ข้อมูลจะถูกส่งเข้าที่ขา SDA และสัญญาณ CLOCK จะถูกส่งเข้าที่ขา SCL โดย Address จะมีขนาด 7 bit มีค่าคงที่เป็น 68H และตามด้วยบิตที่กำหนดสถานะการอ่านหรือเขียนข้อมูล โดยถ้าบิต RW มีลอจิกเป็น 0 จะเป็นการเขียนข้อมูลลงใน DS1307 ซึ่ง Diagram การเขียนข้อมูลแสดงในรูป



• การอ่านข้อมูลจาก DS1307 (DATA READ)

การอ่านข้อมูลจาก DS1307 จะมีลักษณะคล้ายกับการเขียนข้อมูล แต่จะแตกต่างกันที่บิตกำหนดสถานะการอ่านหรือการเขียน (RW) ซึ่งบิตนี้จะต้องมีสถานะเป็นลอจิก 1 เพื่อแสดงการอ่านข้อมูล



การใช้งานตัวอย่างโปรแกรม

C:\S8252 DS1307.HEX ↵

การทำงานของโปรแกรมตัวอย่างจะอ่านค่าเวลาใน DS1307 แล้วส่ง ออก ทาง PORT RS232 ซึ่งผู้ใช้งานสามารถใช้ โปรแกรม TERMINAL (PROCOMM PLUS, ELSE) รับข้อมูลทาง RS232 จาก CP-S8252 ซึ่งกำหนดเงื่อนไขการสื่อสาร

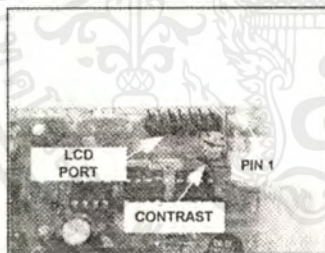
แบบ N,8,1 ที่ค่า BAUD 9600 ซึ่งค่าเวลา ที่อ่านได้จะถูกส่งออกไปแสดงบนหน้าจอของ PC

โดยโปรแกรมตัวอย่างจะเป็นแนวทางในการใช้งาน DS1307 และเป็นโปรแกรมที่ใช้ในการตรวจสอบการทำงานของบอร์ด CP-S8252 ว่าการติดต่อกับ DS1307 ได้ถูกต้องหรือไม่

• LIQUIT CYSTAL DISPLAY ; LCD UNIT

หน่วยแสดงผลแบบ LCD ถูกออกแบบให้ใช้งานได้กับ LCD ชนิด Character โดยควบคุมการทำงานจอ LCD แบบ 4 Bit เพื่อประหยัด Port การใช้งาน ซึ่งตำแหน่งของ PORT LCD ขนาด 14 Pin ของ CP-S8252 จะแสดงผังรูป โดยการต่อใช้งานจะต้องพิจารณาเรื่องของตำแหน่งขาของ PORT LCD บนบอร์ด (ขาสัญญาณบนบอร์ด CP-S8252 จะสลับกับขาสัญญาณที่อยู่บน LCD)

จากรูปจะพบว่าบนบอร์ด CP-S8252 ได้จัดให้มี VR ที่ใช้ปรับความสว่าง(CONTRAST) ของ LCD ซึ่งในหัวข้อถัดไปจะอธิบายถึงการต่อสาย LCD เข้ากับบอร์ด CP-S8252 แต่ก่อนที่จะอธิบายถึงการต่อสายของ LCD จะขอกล่าวถึงขั้นตอนการใช้งาน LCD อย่าง คร่าวๆก่อนโดยส่วนใหญ่ LCD ที่มีจำหน่ายในเมืองไทยจะมีลักษณะคล้ายกันไม่ว่าจะเป็นชุดคำสั่งในการควบคุมการ



การเชื่อมต่อ LCD เข้ากับ CP-S8252

จากรูปแสดงการต่อสาย LCD เพื่อที่จะนำไปต่อกับบอร์ด CP-S8252 โดยการต่อสาย LCD ลงบน LCD จะต้องระวังเรื่องของการต่อสายเป็นสิ่งสำคัญ เนื่องจากถ้าต่อสายผิดจะทำให้ LCD เกิดความเสียหายได้ซึ่งการต่อ LCD จะต้องพิจารณาจากคู่มือของ LCD ในเรื่องของขาต่อใช้

ทำงานของ LCD หรือ คาบเวลาในการควบคุมการทำงานของ LCD

ซึ่งโดยส่วนใหญ่แล้ว LCD ชนิด CHARACTER ไม่ว่าจะมิขนาดเท่าใดจะมีการควบคุมการทำงาน เหมือนกันหมด LCD จะมีขาสัญญาณที่ต้องใช้ควบคุม 14 สัญญาณ ยกเว้น LCD ที่มีไฟส่องหลัง(BACK LIGHT) จะมีสัญญาณเพิ่มขึ้นไปอีก 2 ขา คือขา VCC และ GND ของไฟส่องหลัง

การใช้งาน LCD สามารถแบ่งออกเป็น 2 ลักษณะ คือการควบคุมการทำงาน LCD แบบ 8 บิต และการควบคุมการทำงานแบบ 4 บิต

การใช้งาน LCD แบบ 8 บิตโดยส่วนใหญ่จะถูกใช้งานเมื่อต่อ LCD อยู่กับ BUS ซึ่งจะทำการใช้งาน LCD มีความง่ายขึ้น (การใช้งานแบบ 8 บิตหมายถึงการใช้สายสัญญาณ DATA ทั้ง 8 บิต ในการควบคุม LCD)

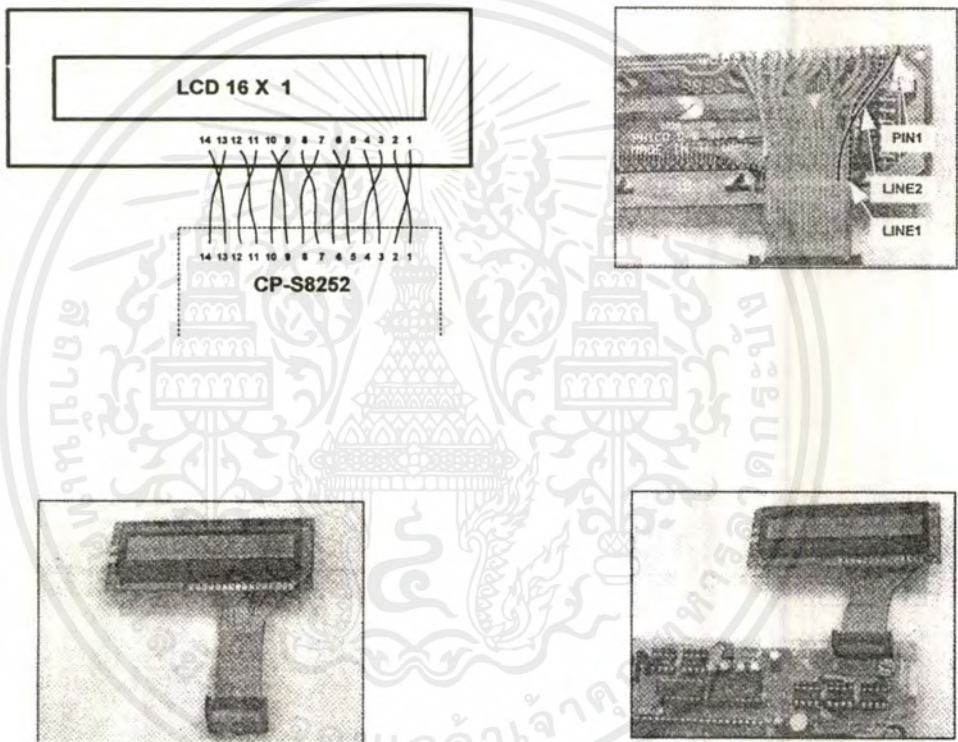
การใช้งาน LCD แบบ 4 บิต (ถูกใช้งานใน CP-S8252)โดยส่วนใหญ่จะถูกใช้งานเมื่อต่อ LCD อยู่กับ PORT ซึ่งจะช่วยให้ประหยัด PORT ของ Microcontroller (การใช้งานแบบ 4 บิต หมายถึงการต่อใช้งานขา DATA เพียง 4 บิต บน (D7 - D4) เท่านั้น โดย 4 บิตล่าง (D3- D0)จะถูกต่อลงกราวด์)

งานเป็นสำคัญ ซึ่งจะต้องต่อให้ตรงกับ PIN ขที่กำหนดบน BOARD CP-S8252

ตัวอย่างการต่อ LCD แบบ 16 x 1 ในการต่อสายจะต้องสลับสายระหว่าง 1-2,3-4,.. 13-14 อันเนื่องมาจากบนบอร์ด CP-S8252 จะมีการสลับสายของขาสัญญาณในลักษณะเป็นคู่ๆไป

Comment: เพื่อให้แน่ใจว่าการต่อสาย LCD ไม่มีผิดพลาด ควรใช้ METER ในการตรวจสอบสาย หลังจาก
 แนใจแล้วว่า LCD ต่อสายถูกให้ต่อไฟเข้ากับ BOARD CP-S8252 แล้วปรับความสว่างของ จอ LCD จะเห็น
 จอ LCD เป็นแถบดำครึ่งจอ

จากรูปด้านล่างแสดงการต่อสายไขว้สลับกันระหว่าง 1-2,3-4,...13-14 เพื่อให้เห็นลักษณะการต่อสายที่
 ชัดเจนขึ้น แต่เพียงระวังเรื่องของการต่อสายไฟและกราวด์ จะทำให้ LCD เกิดความเสียหายได้



การใช้โปรแกรมตัวอย่าง

C:\S8252 LCD.HEX ↵

เมื่อ RUN PROGRAM ตัวอย่าง แล้วจะได้ผลออกจอ LCD เป็นข้อความ CP-S8252

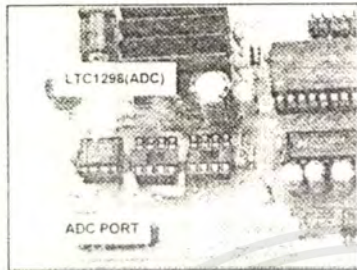
ANALOG TO DIGITAL UNIT

หน่วยแปลงสัญญาณอนาลอกเป็นสัญญาณ

ดิจิทัล(ADC) จะใช้อุปกรณ์ในการแปลงสัญญาณ

LTC1298 เป็น ADC แบบ Serial ADC

2 CHANNEL 12 bit

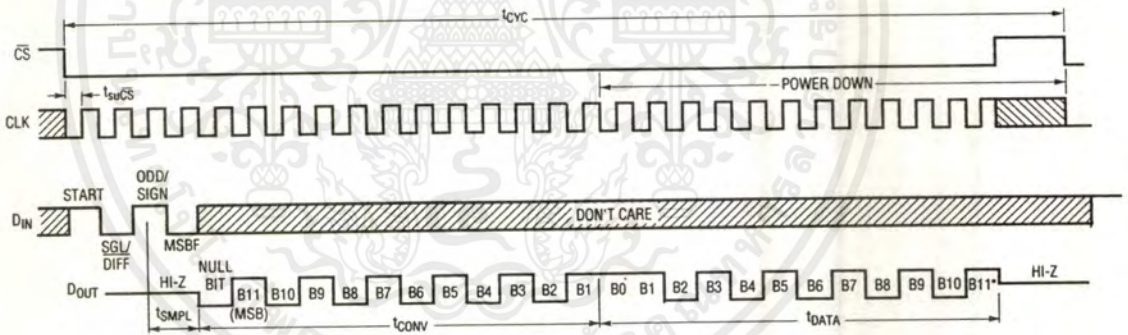


รายละเอียดของ LTC1298

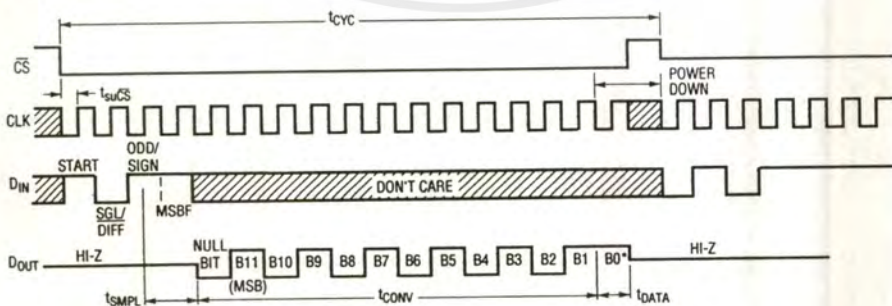
Data Transfer

การสื่อสารข้อมูลระหว่าง Microcontroller กับ LTC1298 จะเป็นลักษณะ Synchronizes ระหว่าง Clock กับ Data ซึ่ง Diagram ของการสื่อสารข้อมูล แสดงดังในรูป

MSB-First Data (MSBF = 0)



MSB-First Data (MSBF = 1)



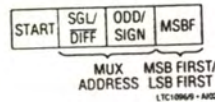
*AFTER COMPLETING THE DATA TRANSFER, IF FURTHER CLOCKS ARE APPLIED WITH CS LOW, THE ADC WILL OUTPUT ZEROS INDEFINITELY.

tDATA: DURING THIS TIME, THE BIAS CIRCUIT AND THE COMPARATOR POWER DOWN AND THE REFERENCE INPUT BECOMES A HIGH IMPEDANCE NODE, LEAVING THE CLK RUNNING TO CLOCK OUT LSB-FIRST DATA OR ZEROES.

LTC1298-02

- Input Data Word

LTC1298 จะต้องส่ง Input Data Word เข้าที่ขา Din จำนวน 4 bit ต่อจาก bit START โดยมี Diagram ของ Input Data Word ดังรูป



- Start Bit

คือบิตแรกที่ Shift เข้าในขา DIN หลังจากขา CS มีลอจิก LOW

- Multiplexer (MUX) Address

บิตที่ Shift ต่อจากบิต START จะเป็นบิตที่ใช้ในการเลือก CHANNEL การอ่านสัญญาณอนาล็อก ซึ่งในรูปแบบด้านล่างจะแสดงตารางเงื่อนไขของ CHANNEL INPUT ทั้ง 2 CHANNEL โดยสามารถกำหนดให้ LTC1298 ทำงานในลักษณะ SINGLE-ENDED หรือ DIFFERENTIAL อีกทั้งสามารถที่จะกำหนดให้ แต่ละ CHANNEL ได้มีศักย์ไฟฟ้าเป็นบวกกับอีก CHANNEL หนึ่งได้

LTC1298 Channel Selection

	MUX ADDRESS		CHANNEL #		GND
	SGL/DIFF	ODD/SIGN	0	1	
SINGLE-ENDED MUX MODE	1	0	+	-	-
DIFFERENTIAL MUX MODE	0	0	+	-	-
	0	1	-	+	+

LTC1096A - AN2

ตัวอย่างการใช้งาน ADC #LTC1298]

C:\S825 ADC.HEX ↵

ผลที่ได้จากตัวอย่างจะต้องป้อนแรงดัน DC 0 – 5 V เฟซที่อินพุตทั้ง 2 CHANNEL แล้วต่อสาย RS232 จาก CP-S8252 เข้ากับ COM PORT ของ PC แล้ว RUN โปรแกรม TERMINAL (PROCOMM PLUS) โดยการ เซ็ต BAUD ไว้ที่ 9600 N,8,1 ซึ่งจะทำให้ค่า Analog To Digital แสดงที่หน้าจอของ PC

• **EEPROM MEMORY**



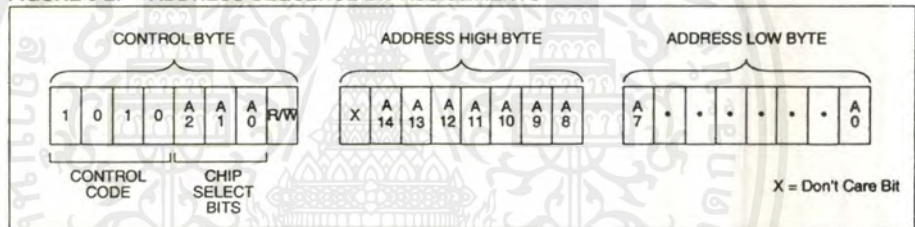
หน่วยความจำใช้เก็บข้อมูลถาวร (Eeprom Memory Unit) ถูกออกแบบให้ใช้หน่วยความจำแบบ Eeprom 24XX โดยกำหนดค่าให้มีการติดต่อสื่อสารระหว่าง Eeprom กับ MCU กำหนดให้ Address ของ Eeprom อยู่ที่ 00H

รายละเอียดของ 24C256

• **Device Addressing**

บิตที่ใช้กำหนดตำแหน่งของ CHIP เพื่อให้ตรงกับที่กำหนดโดย Hard Ware ที่ขา A2,A1,A0 จะกำหนดได้ บิต A2,A1,A0 ใน Control Byte ดังแสดงในรูป

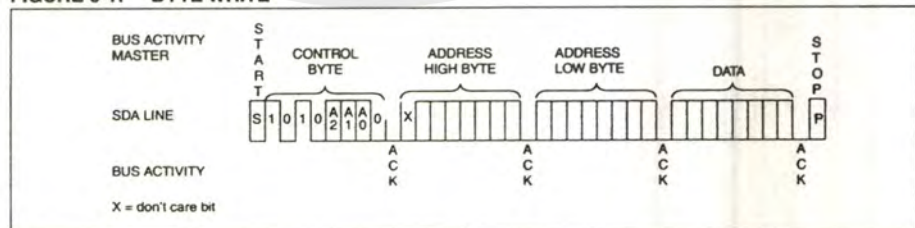
FIGURE 5-2: ADDRESS SEQUENCE BIT ASSIGNMENTS



• **Data Byte Write**

โดยเริ่มจาก START และตามด้วย CONTROL CODE 4 BIT แล้วจึงตามด้วย ADDRESS ของ 42C256 อีก 3 BIT และปิดท้ายด้วย BIT R/W ซึ่งเมื่อต้องการเขียนบิต R/W จะต้องเป็น 0 และต่อท้ายด้วย 3 Byte ของ Address High, Address Low และ Data ตามลำดับ ซึ่งรายละเอียดแสดงดังรูป

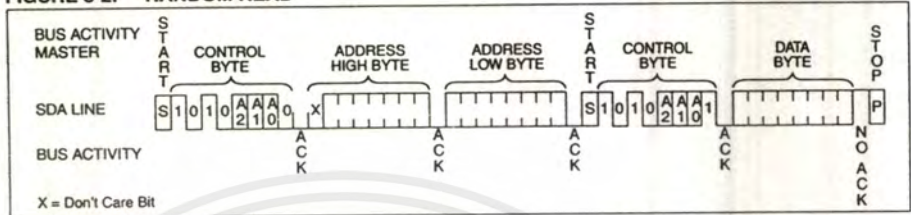
FIGURE 6-1: BYTE WRITE



• Data Byte Read

ขั้นตอนการอ่านข้อมูลจะเหมือนกับขั้นตอนการเขียนข้อมูลจะแตกต่างกันที่บิต R/W จะต้องถูกเซ็ทเป็น 1 ซึ่ง Diagram ของ การอ่านเป็น Byte หรือ RANDOM READ ดังแสดงในรูป

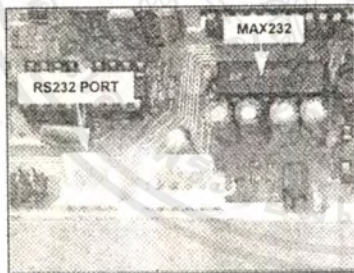
FIGURE 8-2: RANDOM READ



ตัวอย่างการใช้งาน #24XX

CAS8252 EE24XX.HEX ↵

RS232 UNIT



CP-S8252 จัดให้มี PORT ที่ใช้ในการสื่อสารแบบ RS232 โดยการใช้ PORT SERIAL ของ MCU และเปลี่ยนระดับสัญญาณแรงดันเป็น STANDART RS232

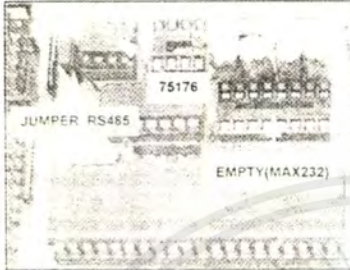
ตัวอย่างการใช้งาน RS232

C:\S8252 RS232.HEX ↵

การทดสอบโปรแกรมตัวอย่างจะต้องต่อสายสัญญาณ TX,RX,GND ระหว่าง CP-S8252 และ PC ให้ถูกต้อง และใช้โปรแกรม TERMINAL (PROCOMM PLUS, ELSE) เป็นตัวรับข้อมูลจาก CP-S8252 โดยการตั้งให้ มีการสื่อสารแบบ N,8,1 และ BAUD 9600

● RS422 / RS485 (OPTION)

นอกจากการสื่อสารแบบ RS232 แล้ว CP-S8252 ได้จัดให้มีการสื่อสารแบบ RS422

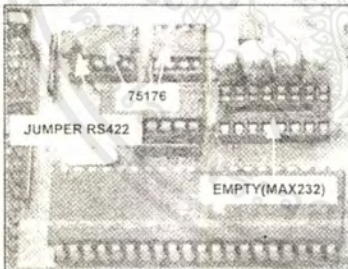


และ RS485 ซึ่งเป็นการสื่อสารระยะไกล การสื่อสารแบบ RS422 ที่แสดงในรูปจะเป็นการสื่อสารแบบ FULL DUPLEX โดยจะต้องเพิ่ม OPTION RS422 ลงใน CP-S8252 โดยการเพิ่ม IC #75176 จำนวน 2 ตัว และ เปลี่ยน JUMPER ให้อยู่ตำแหน่งดังรูป

ตัวอย่างการใช้ RS422

C:\S8252 RS422.HEX ↵

การทดลองโปรแกรมตัวอย่าง RS422 จะต้องมีอุปกรณ์ด้านรับ(PC) ที่เป็น Line Driver ของ RS422 ซึ่งอาจจะเป็น ET-RS422/485 ต่อเข้ากับ



COM PORT ของ PC และใช้ Soft Ware Terminal (Procomm Plus) ในการรับข้อมูลจาก CP-S8252 โดยกำหนดเงื่อนไขการส่งข้อมูลเป็น N,8,1 ที่ BAUD 9600

การใช้งานในส่วนขง RS485 ในตัวอย่างต่อไปนี้องค์จัดซื้อ IC #75176 1 ตัวติดตั้งลงใน CP-S8252 และเปลี่ยนตำแหน่งของ JUMPER ดังรูป

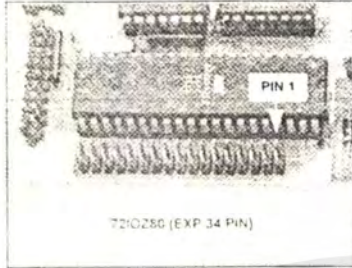
ตัวอย่างการใช้ RS485

C:\S8252 RS485.HEX ↵

การทดลองโปรแกรม RS485 จะต้องใช้ Line Driver ของ RS485 ต่อกับทาง PC เช่นกัน กับ RS422 และใช้ โปรแกรม TERMINAL เป็น

ตัวสื่อสารข้อมูลเช่นกัน โดยกด KEY ใดๆบน PC แล้วข้อมูลจะถูกส่งกลับมาขง PC โดย CP-S8252

• 72IOZ80 EXP 34 PIN UNIT



การเชื่อมต่อแบบ 34 Pin ถูกออกแบบมาให้ใช้เชื่อมต่อ กับอุปกรณ์ภายนอกที่เป็นมาตรฐาน 72IOZ80 ซึ่งจะใช้เชื่อมต่อกับกับอุปกรณ์ I/O ของ ETT หรือสามารถใช้เชื่อมต่อกับอุปกรณ์อื่นได้แต่ต้อง พิจารณาเรื่องขาสัญญาณและไฟด้วย จากรูปแสดง Connector ขนาด 34 pin (72IOZ80) ซึ่งจะมี ประโยชน์มากเมื่อผู้ใช้ ต้องการขยายระบบ เช่น ต้องการต่อชุดควบคุม STEPIING MOTER

ข้อควรระวัง ควรดูตำแหน่งของขาสัญญาณและไฟเพื่อไม่ให้เกิดความเสียหาย

ตัวอย่างการใช้งาน EXP 34 PIN

C:\S8252 EXP34.HEX ↙

ในตัวอย่างโปรแกรมจะเป็นการแสดงผลไฟกระพริบ ออกที่ 34 PIN 72IOZ80 ซึ่งสามารถดูค่าการ

แสดงได้โดยการต่อ LED หรือใช้ ET-TEST I/O ต่อที่ 34 PIN ของ CP-S8252

• EXP 20 PIN EMULATE UNIT

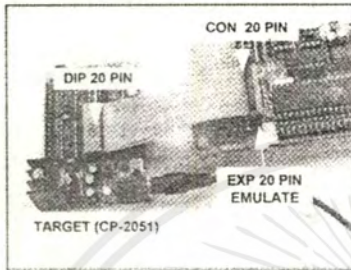
การเชื่อมต่อแบบ 20 Pin นอกจากถูกออกแบบให้ใช้งานในการเชื่อมต่อกับอุปกรณ์ภายนอกทั่วไปแล้วยังถูกออกแบบให้ใช้งานเป็น Emulate ของ MCU(AT89C1051, AT89C2051,

AT89C4051) ขนาดเล็กได้ โดยไม่ต้องหาชุด Emulate ของ MCU เหล่านั้น

โดยการใช้งานหลัก ๆ แล้วจะต้องเขียนโปรแกรมที่มีความยาวไม่เกินกับขนาดของหน่วยความจำของ MCU ที่ต้องการ Emulate และต้องไม่ใช่ทรัพยากรอื่นนอกเหนือจากที่ MCU ที่ต้องการ Emulate จะมีใช้ได้

ตัวอย่างการ EMULATE EXP20 PIN เป็น AT98C2051

ขั้นตอนการใช้งานหลักๆคือ ต้องจัดหา สายที่ด้านหนึ่งเป็น Connector 20 และ อีกด้าน หนึ่งเป็น DIP ขนาด 20 pin (สามารถจัดหาได้ที่



ETT) และจะต้องต่อสายด้านที่เป็น Connector ขนาด 20 pin เข้ากับ Connector 20 pin ที่อยู่บน CP-S8252 และต่อสายด้านที่เป็น DIP 20 pin ลง ในตำแหน่งของ SOCKET ของ MCU ที่ต้องการ Emulate ใน Target Board ดังแสดงในรูป

- POWER SUPPLY UNIT**

หน่วยจ่ายพลังงาน หรือที่รู้จักกันในชื่อของ Power Supply Unit เป็นหน่วยที่มีหน้าที่จ่ายและรักษา แรงดันและกระแสให้กับอุปกรณ์บนบอร์ดควบคุม

(Control Board) ทั้งหมดโดยจะรักษาระดับแรงดัน ค้านเอท์ทุกไว้ที่ 5 VDC ที่กระแสสูงสุด 1 A

คุณสมบัติ (Specification)

Voltage Input : 9-12 V DC or AC

Current Input : More than 1A

Bridge Rectifier : 1A 50V

Regulator : 7805

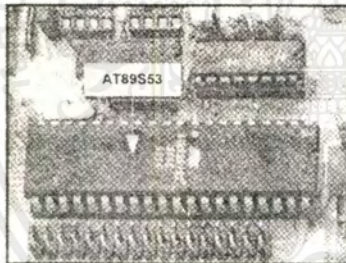
Input Capacitor Filter : 470uF

Output Capacitor Filter : 10uF

Output Voltage Over Protection

คุณสมบัติของ CP-S8252

- MCS-51 #AT89S8252 Microcontroller
- On-Board Down Load (ET-SPI)
- 4 Bit LCD(Character) Interface
- 12 Bit 2 Channel Analog to Digital
- RTC #DS1307
- Eeprom #24XX Interface
- RS232 Communication
- RS422/RS485 Communication
- 72IOZ80 ETT Interface
- EXP 20 PIN Emulate (AT89C1051, AT89C2051, AT89C4051)
- 5VDC Rectifier

MICROCONTROLLER UNIT

- Nine Interrupt Sources
- SPI Serial Interface
- Programmable Watchdog Timer
- Dual Data Pointer
- Low Power Idel and Power Down Mode
- Interrupt Recovery From Power Down
- Power Off Flag
- Programmable UART Serial Channel
- AT89S53 Microcontroller
- Compatible with MCS-51 Product
- 12 K Byte of In-System Reprogrammable Downloadable Flash Memory (1,000 Write/Erase Cycles by Atmel Specification)
- Full Static Operation : 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8 Byte Internal RAM
- 32 Programmable I/O Line
- Three 16-bit Timer/Counter

• ET-SPI UNIT

ขั้นตอนการติดตั้ง Hard Ware

ขั้นตอนการติดตั้ง HARD WARE ใช้ขั้นตอนเดียวกับการติดตั้ง Hard Ware ในรุ่น CP-S8252

ขั้นตอนการติดตั้ง Solf Ware

ระบบ Solf Ware ที่ใช้งานกับ ทำงานภายใต้ DOS และ WINDOWS สามารถ CP-S53 สามารถใช้งานได้ทั้งระบบปฏิบัติการ ปฏิบัติงานได้ทัดเทียมกัน ขึ้นอยู่กับความถนัดของ DOS และ WINDOWS โดย Solf Ware ที่ ผู้ใช้งาน

DOS Operating System

Solf Ware ที่ทำงานบนระบบปฏิบัติการ DOS จะทำงานแบบ Command Line ซึ่งทำให้การใช้งานง่ายและสามารถทำเป็น FILE.BAT เพื่อให้การใช้งานสะดวกมากขึ้น ซึ่งมีรูปแบบของคำสั่งดังต่อไปนี้

S53 <Filename> <Option> ↵

Filename :

คือ File ที่ต้องการ Load ลงในหน่วยความจำของ Microcontroller โดยจะต้องเป็น File ที่ได้รับการแปลงออกมาในรูปของ Intel Hex เรียบร้อยแล้ว เช่น LCD.HEX ซึ่งเป็น File ในโปรแกรมตัวอย่าง

Option :

คือ ส่วนที่ใช้ในการเลือกคุณสมบัติต่างๆของโปรแกรมให้ตรงหรือเหมาะสมกับความต้องการของผู้ใช้งาน

A = LPT1 (Down Load โปรแกรมผ่าน LPT1)

B = LPT2 (Down Load โปรแกรมผ่าน LPT2)

1 - 9 = ค่า Delay time (ใส่ก็ต่อเมื่อ Computer มีความเร็วมากจนไม่สามารถ Down Load ข้อมูลลงใน CPU ได้)

S = Security (ป้องกันการ COPY ข้อมูลจาก Microcontroller)

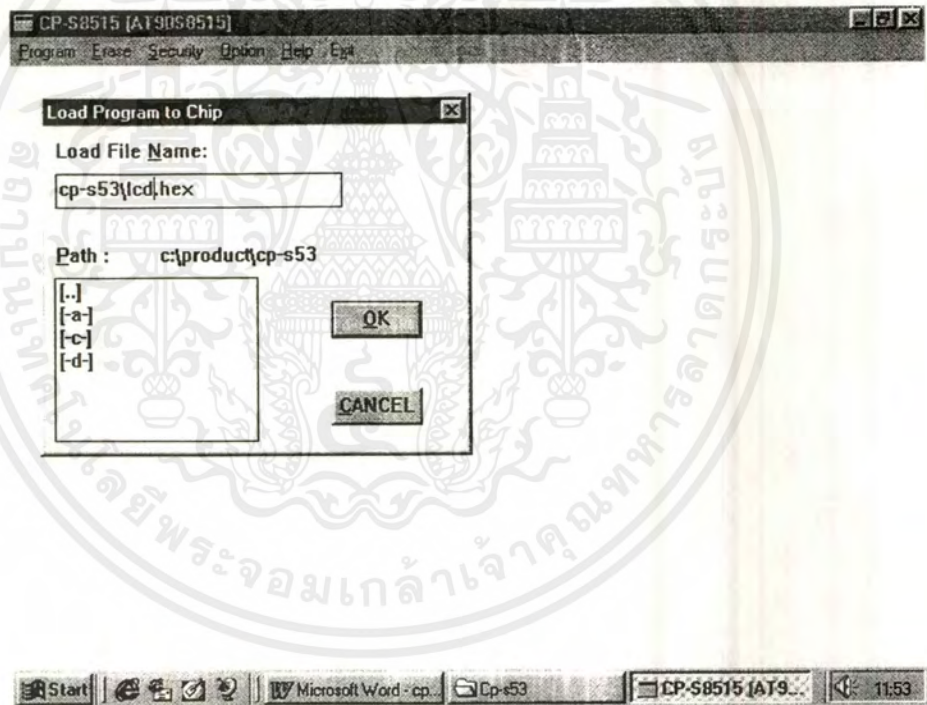
Comment : ถ้าไม่ใส่ OPTION โปรแกรมจะกำหนดให้เป็น LPT1,DELAY = 1, NO SECURITY

WINDOWS Operating System

โปรแกรมที่ใช้ทำงานบนระบบปฏิบัติการ WINDOW ชื่อว่า WINS53.EXE ซึ่งสามารถทำงานได้กับระบบปฏิบัติการ WINDOWS3.1 WINDOWS95,WINDOWS98 โดยมีรายละเอียดการใช้งานดังต่อไปนี้

ขั้นตอนการใช้งาน

1. เตรียมระบบโดยการต่อสายและจ่ายไฟให้กับ CP-S53
2. เรียก WINS53 .EXE ใน WINDOWS
3. เลือก PROGRAM จาก MEMU BAR
4. ใส่ชื่อ FILE.HEX ที่ต้องการจะ DOWN LOAD โดย FILE จะต้องมี FORMAT เป็น INTEL HEX
5. กดปุ่ม OK ใน PROGRAM DIALOGBOX



MENU ERASE : เป็น MENU ที่ใช้ ERASE ข้อมูลใน Microcontroller โดยไม่มีเขียนข้อมูลใดๆ ลงในหน่วยความจำ MENU

SECURITY : เป็น MENU ที่ใช้ทำ SECURITY เพื่อ LOG ข้อมูลที่อยู่ใน Microcontroller เพื่อไม่ให้ถูกอ่านออกมาอีก

MENU OPTION : เป็น MENU ที่ใช้เลือก LPT (LPT1 or LPT2) ที่ใช้ Down Load ข้อมูล และใช้เลือกค่า Delay ในกรณีที่เครื่อง PC มีความเร็วมากจนไม่สามารถ Down Load ข้อมูลลงในหน่วยความจำของ Microcontroller ได้

ทั้ง PROGRAM ที่ทำงานบน DOS และ WINDOWS ของ CP-S53 จะทำ AUTO ERASE ทุกครั้ง ที่เขียนข้อมูลลงใน Microcontroller แต่จะไม่ทำ SECURITY

• RTC UNIT (OPTION)

การใช้งานตัวอย่างโปรแกรม

C:\S53 DS1307.HEX ↵

การทำงานของโปรแกรมตัวอย่างจะอ่านค่าเวลาใน DS1307 แล้วส่ง ออก ทาง PORT RS232 ซึ่งผู้ใช้งานสามารถใช้ โปรแกรม TERMINAL (PROCOMM PLUS, ELSE) รับข้อมูลทาง RS232 จาก CP-S53 ซึ่งกำหนดเงื่อนไขการสื่อสารแบบ

N,8,1 ที่ค่า BAUD 9600 ซึ่งค่าเวลา ที่อ่านได้จะถูกส่งออกไปแสดงบนหน้าจอของ PC

โดยโปรแกรมตัวอย่างจะเป็นแนวทางในการใช้งาน DS1307 และเป็นโปรแกรมที่ใช้ในการตรวจสอบการทำงานของบอร์ด CP-S53 ว่าการติดต่อกับ DS1307 ยังคงทำงานได้ถูกต้อง

• LIQUIT CRYSTAL DISPLAY ; LCD UNIT

การใช้โปรแกรมตัวอย่าง

C:\S53 LCD.HEX ↵

เมื่อ RUN PROGRAM ตัวอย่าง แล้วจะได้ผลออกจอ LCD เป็นข้อความ CP-S53

• ANALOG TO DIGITAL UNIT

ตัวอย่างการใช้งาน ADC #LTC1298

C:\S53 ADC.HEX ↵

ผลที่ได้จากตัวอย่างจะต้องป้อนแรงดัน DC (PROCOMM PLUS) โดยการ เซ็ต BAUD RATE
0-5 V หน้าที่อินพุตทั้ง 2 CHANNEL แล้วต่อสาย 9600 N,8,1 ซึ่งจะทำให้ค่า Analog To Digital
RS232 จาก CP-S53 เข้ากับ COM PORT ของ แสดงที่หน้าจอของ PC
PC แล้ว RUN โปรแกรม TERMINAL

• EEPROM MEMORY

ตัวอย่างการใช้งาน #24XX

C:\S53 EE24XX.HEX ↵

RS232 UNIT

ตัวอย่างการใช้งาน RS232

C:\S53 RS232.HEX ↵

การทดสอบโปรแกรมตัวอย่างจะต้องต่อสาย
สัญญาณ TX,RX,GND ระหว่าง CP-S53 และ
PC ให้ถูกต้อง

และใช้โปรแกรม TERMINAL (PROCOMM PLUS, ELSE) เป็นตัวรับข้อมูลจาก CP-S53 โดย
การเซตให้มีการสื่อสารแบบ N,8,1 และ BAUD
9600

• RS422 / RS485 (OPTION)

ตัวอย่างการใช้ RS422

C:\S53 RS422.HEX ↵

การทดลองโปรแกรมตัวอย่าง RS422 จะต้องมีอุปกรณ์ด้านรับ(PC) ที่ขึ้น Line Driver ของ RS422 ซึ่งอาจจะเป็น ET-RS422/485 ต่อเข้ากับ COM PORT ของ PC และใช้ Soft Ware

Terminal (Procomm Plus) ในการรับข้อมูลจาก CP-S8252 โดยกำหนดเงื่อนไขการส่งข้อมูลเป็น N,8,1 ที่ BAUD 9600

ตัวอย่างการใช้ RS485

C:\S53 RS485.HEX ↵

การทดลองโปรแกรม RS485 จะต้องใช้ Line Driver ของ RS485 ต่อกับทาง PC เช่นกัน กับ RS422 และใช้ โปรแกรม TERMINAL เป็น

ตัวสื่อสารข้อมูลเช่นกัน โดยกด KEY ใดๆบน PC แล้วข้อมูลจะถูกส่งกลับมายัง PC โดย CP-S53

• 72IOZ80 EXP 34 PIN UNIT

ตัวอย่างการใช้งาน EXP 34 PIN

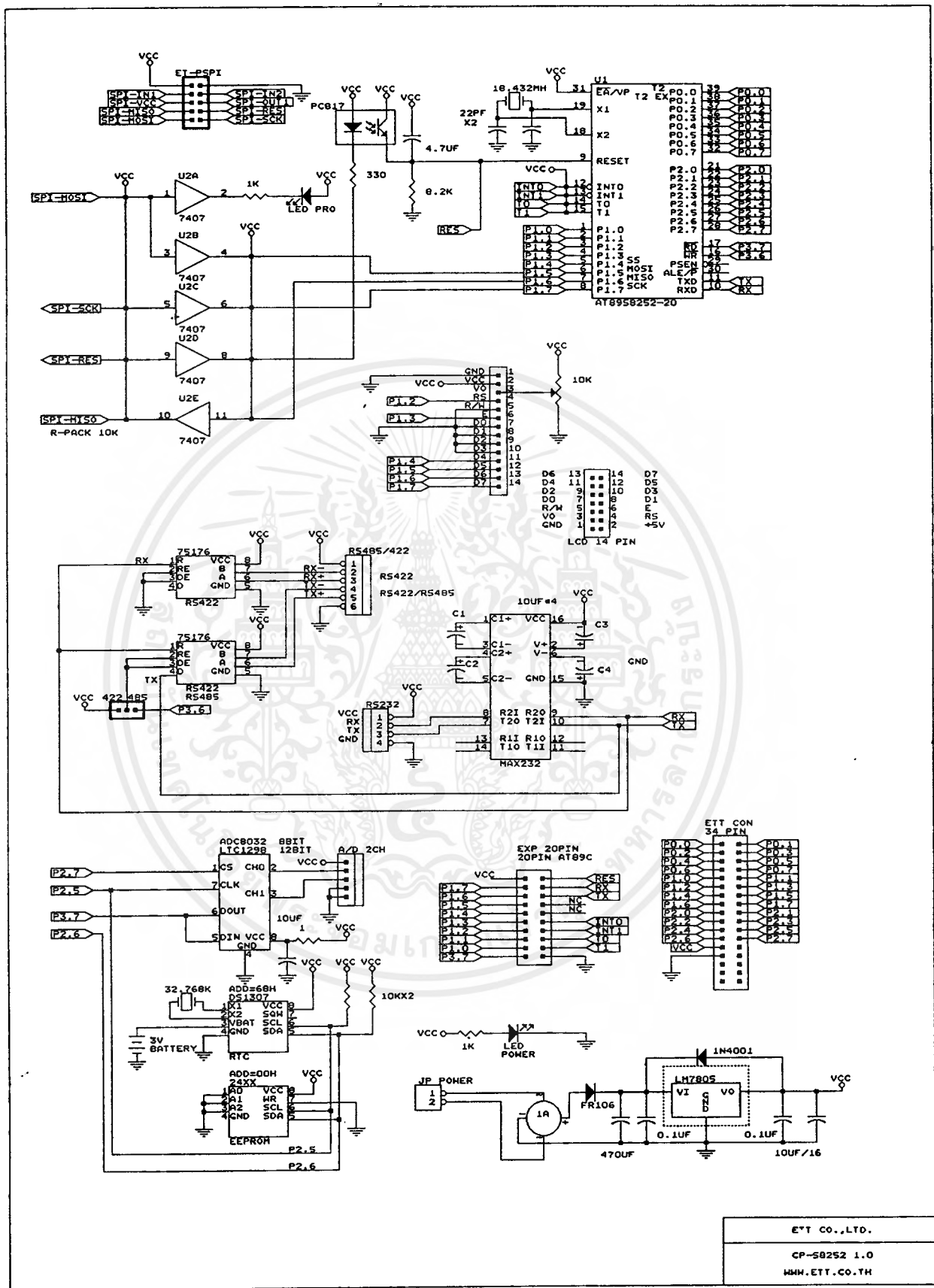
C:\S53 EXP34.HEX ↵

ในตัวอย่างโปรแกรมจะเป็นการแสดง ไฟกระพริบ ออกที่ 34 PIN 72IOZ80 ซึ่งสามารถดูค่าการ

แสดงได้โดยการต่อ LED หรือใช้ ET-TEST I/O ต่อที่ 34 PIN ของ CP-S53

• EXP 20 PIN EMULATE UNI

การใช้งาน EXP 20 PIN ดูรายละเอียดได้ใน CP-S8252



ETT CO., LTD.
 CP-S8252 1.0
 WHM.ETT.CO.TH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. การตรวจหาจุดเฉพาะในภาพพิมพ์ลายนิ้วมือ, นายวาทีต วิโรจนาวีตรและนายวัชรระ ศิริอ่อน
2. ระบบเปรียบเทียบลายนิ้วมืออัตโนมัติ, รศ. ครรชิต ไมตรี
3. Fingerprint Recognition , นายธานีินทร์ พินทอง และ นายศรนรินทร์ ตพรัตน์
4. เทคนิคสำหรับตรวจจับจุดเฉพาะบนลายนิ้วมือ , นายสมศักดิ์ แซ่จ้ง และ นายสุรัชย์ บุญมาพึ่ง
5. ระบบตรวจเทียบลายนิ้วมืออัตโนมัติโดยการใช้ฮาร์ดแวร์ทศทางและโครงข่ายนิวรอล, เพื่อประโยชน์การนำไปใช้ควบคุมอุปกรณ์, เกรียง ไกร โจนเจริญสุข
6. Dinesh P Mital and Eam Khwang Teoh, School of Electrical and Electronic Engineering, Nanyang Technological University, "An Automated Matching Technique for Fingerprint Identification", IEEE, vol.2, pp.88-92, 1996
7. Raffaele Cappelli, Alessandra Lumini, Dario Maio, "Fingerprint Classification by Directional Image Partitioning", IEEE, vol.21, no 5, May 1999
8. Kalle Karu, Anil K. Jain, "Fingerprint Classification", Pattern Recognition, vol.29, no 3, pp 389-404, 1996
9. ฉัททวุฒิ พิษผล, พิชิต ตันติกุลานนท์, "คู่มือเรียน Visual Basic 6", กรุงเทพมหานคร: บริษัท โปรวิชั่น จำกัด
10. อุดม จีนประดับ, "ไมโครคอนโทรลเลอร์ MCS-51", กรุงเทพมหานคร: สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
11. สุพจน์ แซ่เตีย, "แนะนำการเขียน Visual Basic ในงานวิศวกรรม", วารสารเซมิคอนดักเตอร์, มีนาคม 2542