

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องสแกนลายนิ้วมือสำหรับวิชาปฏิบัติการ
FINGERPRINT SCAN FOR LABORATORY



T104260



เลขหมู่.....
เลขทะเบียน..... 104260
วัน,เดือน,ปี..... 30 ต.ค. 2552

.....
.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FINGERPRINT SCAN FOR LABORATORY



**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE
OF BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาบัตร เครื่องสแกนลายนิ้วมือสำหรับวิชาปฏิบัติการ
FINGERPRINT SCAN FOR LABORATORY

ชื่อนักศึกษา นายกวิน ตั้งวิทยานุกูล รหัสนักศึกษา 48010031
นายคัมภีร์ ศุภรรกุล รหัสนักศึกษา 48010097
นางสาวนัฐชนก ปุ่มพิมาย รหัสนักศึกษา 48010438

อาจารย์ที่ปรึกษา รศ. อุทัย ศรีธีระวิโรจน์
อ.สรพงษ์ วชิรรัตนพรกุล

ระดับการศึกษา ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศ

ภาควิชา วิศวกรรมสารสนเทศ

ปีการศึกษา 2551

ปริญญาบัตรฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

(รศ. อุทัย ศรีธีระวิโรจน์)

(อ.สรพงษ์ วชิรรัตนพรกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาบัตร เครื่องสแกนลายนิ้วมือสำหรับวิชาปฏิบัติการ
ชื่อนักศึกษา นายกวิน ตั้งวิทยานุกูล รหัสนักศึกษา 48010031
นายคัมภีร์ ศุภชรกุล รหัสนักศึกษา 48010097
นางสาวนัฐชนก ปุ่มพิมาย รหัสนักศึกษา 48010438
อาจารย์ที่ปรึกษา รศ. อุทัย ศรีธีระวิโรจน์
อ.สรพงษ์ วชิรรัตนพรกุล
ระดับการศึกษา ปริญญาตรีวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศ
ภาควิชา วิศวกรรมสารสนเทศ
ปีการศึกษา 2551

บทคัดย่อ

เครื่องสแกนลายนิ้วมือสำหรับห้องปฏิบัติการ เป็นการนำเอาเทคโนโลยีการสแกนลายนิ้วมือมาประยุกต์ใช้ในการศึกษา เพื่ออำนวยความสะดวก และการจัดการข้อมูลของนักศึกษา ในการตรวจสอบการเข้าเรียนวิชาปฏิบัติ และที่สำคัญใช้สำหรับเก็บหลักฐานเพื่อป้องกันการปลอมแปลงข้อมูลการเข้าเรียนของนักศึกษา โดยสามารถตรวจสอบได้ ทำให้เป็นประโยชน์แก่ผู้สอน ทั้งนี้อาศัยหลักการสแกนลายนิ้วมือ ความแตกต่างกันของลายนิ้วมือของแต่ละบุคคล ทำให้ได้ผลที่ถูกต้องแม่นยำและเชื่อถือได้ยิ่งขึ้น

Project Title FINGERPRINT SCAN FOR LABORATORY

Student Mr. Kawin Tangwitthayakul ID 4800031
Mr. Kumpee Supathornkul ID 48010097
Miss. Natchanok Pumphimai ID 48010438

Advisor Assist. Prof. Uthai Sriteerawiroj
Mr. Sorapong Wachiratanapomkul

Graduate Level Bachelor Degree of Information Engineering

Department Information Engineering

Academic Year 2551

Abstract

Fingerprint scanner for a laboratory takes technology fingerprint scan applied for use in education. It's convenience for manage and check the students' information that comes to study the laboratory class. It's protecting for forge a signature because the student must scan their fingerprint so it can make sure and more believable for check the list of the student.

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จเรียบร้อยได้ด้วยดี ก็ด้วยได้รับคำแนะนำและชี้แนวทางในการศึกษาค้นคว้าข้อมูล รายละเอียด และช่วยแก้ไขในส่วนที่บกพร่องต่างๆจาก รองศาสตราจารย์ อุทัย ศรีธีระวิโรจน์ และ อ.สรพงษ์ วชิรรัตนพรกุล คณะผู้จัดทำขอขอบพระคุณท่านอาจารย์ทั้งสองเป็นอย่างสูง

กราบขอบพระคุณคุณพ่อ คุณแม่ที่คอยดูแลห่วงใย มอบกำลังใจและให้การสนับสนุนในทุกๆเรื่อง รวมทั้งขอขอบคุณพี่ๆและน้องๆ ทุกคนที่เป็นกำลังใจและให้การช่วยเหลือในด้านต่างๆเสมอมาโดยตลอด

ขอขอบคุณนาย ภาณุพล ชีรมโนภาพ(นุ) ที่มาช่วยและแนะนำสิ่งต่างๆทั้งการออกแบบและแนวทางการเขียนโปรแกรม อีกทั้งยังสละเวลาช่วยเหลือในการทำโครงการนี้ ตลอดจนการทำโครงการนี้สำเร็จ

สุดท้ายขอขอบคุณเพื่อนๆห้อง E12-1002 และเพื่อนๆภาควิชาวิศวกรรมสารสนเทศรุ่น 8 ทุกคนที่คอยช่วยเหลือ ไม่ทอดทิ้งกัน คณะผู้จัดทำขอขอบคุณจากใจจริง

นายกวิน ตั้งวิทยานุกุล

นายคัมภีร์ ศุภชรกุล

นางสาวนัญชนก ปุ่มพิมาย

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ช
สารบัญตาราง	ฉ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของปริญญาโท	1
1.2 รายละเอียดโดยย่อของ โครงการงาน	1
1.3 วัตถุประสงค์ของ โครงการงาน	1
1.4 ขอบเขตของ โครงการงาน	2
1.5 ขั้นตอนการดำเนินงาน	2
1.6 ประโยชน์และผลที่คาดว่าจะได้รับจาก โครงการงานนี้	3
1.7 เนื้อหาภายในปริญญาโท	3
บทที่ 2 ทฤษฎีพื้นฐาน	4
2.1 ลักษณะของจุดต่างๆบนลายนิ้วมือ	5
2.1.1 จุดกึ่งกลาง(Core)	5
2.1.2 ลักษณะสำคัญ(Minutiae)	5
2.2 ชนิดและรูปแบบลายนิ้วมือ	6
2.2.1 กลุ่มเส้นโค้ง(Arch)	7
2.2.2 กลุ่มมัดหวาย(Loop)	8
2.2.3 กลุ่มก้นหอย(Whorl)	10
2.2.4 กลุ่มซับซ้อน(Accidental Whorl)	10
2.3 หลักการวิเคราะห์ลายนิ้วมือ	11
2.4 คุณสมบัติของเครื่อง U.R.U 4000	13
2.4.1 หลักการทำงานของเครื่อง U.R.U 4000	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.5 โปรแกรมจัดการฐานข้อมูล Microsoft Access	16
2.5.1 ฐานข้อมูล (Database)	16
2.5.2 ฐานข้อมูลอยู่ในคอมพิวเตอร์ดีกว่าอยู่ในสมุดหรือแฟ้มกระดาษ	17
2.5.3 ประโยชน์ของระบบการจัดการฐานข้อมูล	17
2.5.4 การออกแบบฐานข้อมูล (Database Design)	17
2.5.5 Flat-file databases และ Relational databases	18
2.5.6 ขั้นตอนการออกแบบฐานข้อมูล	18
2.5.7 วัตถุฐานข้อมูล (Database Objects)	19
2.5.8 Tables (ตารางข้อมูล)	19
2.5.9 การออกแบบ Tables	20
2.5.10 ความสัมพันธ์ระหว่างสอง Tables	21
2.5.11 Queries (แบบสอบถาม)	21
2.5.12 Reports (รายงาน)	23
2.6 Microsoft Visual Studio	24
2.6.1 Visual Studio	24
2.6.2 เครื่องมือพัฒนาโปรแกรมใน Visual Studio	25
2.7 Visual Basic ใน Access	26
2.7.1 การทำงานของ Visual Basic	26
2.7.2 ไวยากรณ์ของ Visual Basic	27
บทที่ 3 การทำงานและการออกแบบ	29
3.1 แนวคิดการออกแบบระบบงาน	29
3.2 ส่วนประกอบของระบบงาน (System Diagram)	30
3.3 ขั้นตอนการทำงาน Flow chart	31
3.3.1 การออกแบบโปรแกรมส่วนของข้อมูลนักศึกษา	31
3.3.2 การออกแบบโปรแกรมส่วนรายงานการเข้าเรียน	32
3.3.3 การออกแบบโปรแกรมส่วนของวิชา	33
3.3.4 การออกแบบโปรแกรมส่วนของข้อมูลชั้นเรียน	34
3.3.5 การออกแบบโปรแกรมส่วนบันทึกการเข้าเรียน	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

3.4	ขั้นตอนการทำงานของระบบ	36
3.5	คำอธิบายขั้นตอนการดำเนินงาน	36
3.6	การออกแบบฐานข้อมูล	36
3.7	ภาษาและเครื่องมือที่ใช้ในการพัฒนา	38
3.7.1	ภาษาที่ใช้พัฒนา	38
3.7.2	ซอฟต์แวร์	38
3.7.1	ฮาร์ดแวร์	39
บทที่ 4	การใช้งาน	40
4.1	เริ่มต้นเข้าสู่โปรแกรมการใช้งาน	40
4.2	ปุ่มข้อมูลนักศึกษา	41
4.3	ปุ่มวิชา	47
4.4	ปุ่มข้อมูลชั้นเรียน	48
4.5	ปุ่มบันทึกการเข้าเรียน	50
4.6	ปุ่มรายงานการเข้าเรียน	53
บทที่ 5	บทวิจารณ์และบทสรุป	54
5.1	สรุปผลการดำเนินปริญญานิพนธ์	54
5.2	ข้อจำกัดของระบบ	54
5.3	แนวทางในการพัฒนาต่อ	55
บรรณานุกรม		56
ภาคผนวก ก.	คู่มือการใช้งาน+คุณสมบัติของระบบ	57
ภาคผนวก ข.	คู่มือการติดตั้งและการตั้งค่า	66
ภาคผนวก ค.	Data sheet ของ IDWORKS Integrator SDK	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 บริเวณที่มีลักษณะเป็นจุดกึ่งกลาง	5
รูปที่ 2.2 จุดแยกของลายนูน	6
รูปที่ 2.3 จุดปลายของลายนูน	6
รูปที่ 2.4 รูปแบบกลุ่มโค้งราบ(Arch)	7
รูปที่ 2.5 รูปแบบเส้นโค้งกระโจม(Tented Arch)	7
รูปที่ 2.6 รูปแบบกลุ่มมัดหวายเอียงขวา(Right Loop)	8
รูปที่ 2.7 รูปแบบกลุ่มมัดหวายเอียงซ้าย (Left Loop)	9
รูปที่ 2.8 รูปแบบของกลุ่มมัดหวายคู่ (Double Loop)	9
รูปที่ 2.9 รูปแบบของกลุ่มก้นหอย (Whorl)	10
รูปที่ 2.10 รูปแบบกลุ่มซับซ้อน (Accidental Whorl)	10
รูปที่ 2.11 กระบวนการบันทึกลายนิ้วมือ	11
รูปที่ 2.12 กระบวนการเปรียบเทียบลายนิ้วมือ	12
รูปที่ 2.13 จุดลักษณะเฉพาะบนลายนิ้วมือ	12
รูปที่ 2.14 เครื่องสแกนนิ้ว U.R.U 4000	13
รูปที่ 2.15 โครงสร้างการทำงานของคอมพิวเตอร์ Biokey.ocx	14
รูปที่ 2.16 ขั้นตอนการอ่านลายนิ้วมือ และ บันทึก ของคอมพิวเตอร์ Biokey.ocx	15
รูปที่ 2.17 แสดงการตรวจสอบลายนิ้วมือ ของคอมพิวเตอร์ Biokey.ocx	16
รูปที่ 2.18 ตัวอย่าง Objects	19
รูปที่ 2.19 ตัวอย่างTable อย่างง่าย (เมื่อกรอกข้อมูลแล้ว)	20
รูปที่ 2.20 หน้าต่างใช้สร้างโครงสร้างของ Table	20
รูปที่ 2.21 ตัวอย่าง Relationships	21
รูปที่ 2.22 แสดงการสร้าง Query	22
รูปที่ 2.23 หน้าตาของ Report อย่างง่าย	23
รูปที่ 2.24 Mailing Label (ฉลากจำหน่ายของจดหมาย)	23

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 3.1 รูปแบบและขั้นตอนการทำงาน	30
รูปที่ 3.2 Flowchart ข้อมูลนักศึกษา	31
รูปที่ 3.3 Flowchart รายงานการเข้าเรียน	32
รูปที่ 3.4 Flowchart วิชา	33
รูปที่ 3.5 Flowchart ข้อมูลชั้นเรียน	34
รูปที่ 3.6 Flowchart บันทึกการเข้าเรียน	35
รูปที่ 3.7 ความสัมพันธ์ระหว่างตาราง	37
รูปที่ 3.8 ตาราง CheckClass	37
รูปที่ 3.9 ตาราง Course	37
รูปที่ 3.10 ตาราง Course Teacher	38
รูปที่ 3.11 ตาราง IdFingerprint	38
รูปที่ 3.12 ตาราง Student	38
รูปที่ 4.1 หน้าต่างการใช้งานหลัก	40
รูปที่ 4.2 หน้าต่างข้อมูลนักศึกษา	41
รูปที่ 4.3 ตัวอย่างการค้นหารายชื่อนักศึกษา	42
รูปที่ 4.4 หน้าจอการบันทึกสลายนิ้วมือก่อนใส่สลายนิ้วมือ	42
รูปที่ 4.5 หน้าจอการใส่สลายนิ้วมือที่ผิดพลาด	43
รูปที่ 4.6 หน้าจอแสดงสลายนิ้วมือที่คุณภาพต่ำ	44
รูปที่ 4.7 หน้าจอแสดงสลายนิ้วมือที่คุณภาพต่ำ	45
รูปที่ 4.8 หน้าจอการบันทึกสลายนิ้วมือเมื่อใส่สลายนิ้วมือเสร็จสิ้น	46
รูปที่ 4.9 หน้าจอการใส่รหัสสลายนิ้วมือหลังจากเก็บข้อมูลสลายนิ้วมือแล้ว	46
รูปที่ 4.10 หน้าต่างแสดงผลการบันทึกสลายนิ้วมือ	47
รูปที่ 4.11 หน้าต่างวิชาเรียน	47
รูปที่ 4.12 ตัวอย่างการค้นหารายชื่อวิชา	48
รูปที่ 4.13 หน้าต่างข้อมูลชั้นเรียน	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 4.14 ตัวอย่างการเพิ่มชื่ออาจารย์ในแต่ละวิชา	49
รูปที่ 4.15 หน้าต่างแสดงการเพิ่มชื่อเรียบร้อยแล้ว	50
รูปที่ 4.16 หน้าต่างการบันทึกการเข้าเรียน	51
รูปที่ 4.17 ตัวอย่างการบันทึกถลายนิ้วมือ	51
รูปที่ 4.18 หน้าต่างแสดงผลเวลาเข้าเรียน	52
รูปที่ 4.19 หน้าต่างแสดงผลเวลาเลิกเรียน	52
รูปที่ 4.20 หน้าต่างแสดงรายชื่อนักศึกษาที่บันทึกเวลาเรียนในรายวิชานั้นๆ	53
รูปที่ 4.21 หน้าต่างรายงานการเข้าเรียนของนักศึกษา	53
รูปที่ ก.1 หน้าต่างการใช้งานหลัก	59
รูปที่ ก.2 หน้าต่างข้อมูลนักศึกษา	60
รูปที่ ก.3 หน้าต่างการบันทึกถลายนิ้วมือ	61
รูปที่ ก.4 หน้าต่างใส่รหัสถลายนิ้วมือหลังจากเก็บข้อมูลถลายนิ้วมือแล้ว	61
รูปที่ ก.5 หน้าต่างวิชาเรียน	62
รูปที่ ก.6 หน้าต่างข้อมูลชั้นเรียน	63
รูปที่ ก.7 หน้าต่างแสดงการเพิ่มชื่ออาจารย์เรียบร้อยแล้ว	63
รูปที่ ก.8 หน้าต่างการบันทึกการเข้าเรียน	64
รูปที่ ก.9 ตัวอย่างการบันทึกถลายนิ้วมือ	64
รูปที่ ก.10 หน้าต่างแสดงรายชื่อนักศึกษาที่บันทึกเวลาเรียนในรายวิชานั้นๆ	65
รูปที่ ก.11 หน้าต่างรายงานการเข้าเรียนของนักศึกษา	65
รูปที่ ข.1 เริ่มต้นติดตั้ง FingerLab	67
รูปที่ ข.2 รายละเอียดในการติดตั้ง	67
รูปที่ ข.3 ข้อตกลงในการติดตั้ง	68
รูปที่ ข.4 เลือกโฟลเดอร์การติดตั้ง	68
รูปที่ ข.5 โปรแกรมแจ้งเตือนว่ายังไม่มีโฟลเดอร์นี้ ต้องการจะสร้างเพิ่มหรือไม่	69
รูปที่ ข.6 ยืนยันการติดตั้ง	69
รูปที่ ข.7 กำลัง Install	70
รูปที่ ข.8 เลือกติดตั้ง Driver เพิ่ม	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

	หน้า
รูปที่ ข.9 เสร็จสิ้นการติดตั้ง FingerLab	71
รูปที่ ข.10 หน้าต่างเลือกภาษา	71
รูปที่ ข.11 หน้าต่างการติดตั้ง IDWORKS Integrator 2.4.1	72
รูปที่ ข.12 ข้อตกลงเรื่องลิขสิทธิ์	72
รูปที่ ข.13 กำลัง Install IDWORKS Integrator 2.4.1	73
รูปที่ ข.14 ติดตั้ง IDWORKS Integrator 2.4.1 กำลังเสร็จสิ้น	73
รูปที่ ข.15 เลือกภาษาในการติดตั้ง Driver	74
รูปที่ ข.16 เริ่มการติดตั้ง Driver	74
รูปที่ ข.17 ข้อตกลงในการติดตั้ง Driver	75
รูปที่ ข.18 เลือกไฟล์เคอร์ในการติดตั้ง Driver	75
รูปที่ ข.19 หน้าต่างพร้อมติดตั้ง Driver	76
รูปที่ ข.20 กำลัง Install Driver	76
รูปที่ ข.21 หน้าต่างใส่ Security Key	77
รูปที่ ข.22 ติดตั้งเสร็จสมบูรณ์	77

สารบัญตาราง

ตารางที่ 1.1 ขั้นตอนวิธีการดำเนินงาน

หน้า

2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของปริญญาบัตร

รูปแบบของการเช็คชื่อการเข้าเรียนในปัจจุบันมีข้อจำกัดในหลายรูปแบบ โดยทั่วไปจะอาศัยการขานชื่อ หรือการลงลายมือชื่อ ซึ่งก่อให้เกิดปัญหาการที่นักศึกษาทำการเช็คชื่อแทนกัน โดยที่ไม่ต้องมาเข้าเรียนเอง และเป็นการยากต่อการที่จะตรวจสอบลายมือชื่อนั้น จึงทำให้เกิดแนวคิดที่จะมีการเช็คชื่อที่แม่นยำในการตรวจสอบว่านักศึกษาเข้าห้องเรียนจริงหรือไม่ โดยเราต้องมองหาสิ่งในที่จะไม่สามารถยืมหรือฝากบุคคลอื่นมาเช็คแทนได้ ซึ่งส่วนนั้นในร่างกายมนุษย์ก็มีอยู่แล้วนั่นก็คือ "ลายนิ้วมือ" ซึ่งเป็นลักษณะเฉพาะของแต่ละบุคคลที่มีความแตกต่างกัน ไม่สามารถแก้ไข ปลอมแปลง หรือลักลอบใช้แทนกันได้ ด้วยคุณสมบัติเด่นนี้ ทำให้ลายนิ้วมือได้ถูกเลือกใช้ในการตรวจสอบการเข้าเรียนวิชาปฏิบัติการ

1.2 รายละเอียดโดยย่อของโครงการ

โครงการนี้เป็นโครงการเครื่องสแกนลายนิ้วมือที่สามารถทำงานร่วมกับเครื่องคอมพิวเตอร์ โดยใช้เครื่องสแกนลายนิ้วมือเป็นตัวรับภาพลายนิ้วมือและทำการประมวลผลเปรียบเทียบลายนิ้วมือ และจะทำการเชื่อมต่อฐานข้อมูลเข้ากับเครื่องคอมพิวเตอร์ ซึ่งสามารถตรวจสอบดูผลรายงานได้ว่านักศึกษาเข้าเรียนเวลาเท่าไร และออกเวลาเท่าไร ทำให้สามารถตรวจเช็คการเข้าเรียนของนักศึกษาสำหรับวิชาปฏิบัติการได้อย่างถูกต้องแม่นยำ

1.3 วัตถุประสงค์ของโครงการ

- 1) เพื่อพัฒนาระบบการเช็คชื่อเพื่อประโยชน์ของนักศึกษาให้มีความแม่นยำมากขึ้น ความถูกต้องมากขึ้น
- 2) ตรวจสอบการเข้าห้องเรียนหรือห้องปฏิบัติการและสามารถทราบรายละเอียดเกี่ยวกับการเข้าเรียน และออกได้
- 3) เพื่อศึกษาการทำงานของฮาร์ดแวร์
- 4) เพื่อศึกษาการเขียนโปรแกรมเชื่อมระหว่างฐานข้อมูลและฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตของโครงการ

- 1) สามารถตรวจสอบการเข้าเรียนของนักศึกษาได้จริง
- 2) สามารถแสดงผลบอก วัน และ เวลา ในการเข้าเรียนและบอกได้ว่าขาดเรียนกี่ครั้ง
- 3) สามารถรอกข้อมูล และแสดงผลรายละเอียดเกี่ยวกับนักศึกษา

1.5 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1

ขั้นตอนที่	วิธีการดำเนินงาน	ระยะเวลาการดำเนินงาน								
		มิถุนายน	กรกฎาคม	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน	ธันวาคม	มกราคม	กุมภาพันธ์
1.	ศึกษาการเขียนโปรแกรมต่างๆ	↔								
2.	ศึกษาโครงสร้างฮาร์ดแวร์ที่เกี่ยวข้องกับการตรวจลายนิ้วมือและการแสดงผล	↔								
3.	เริ่มการเขียนโปรแกรมและจัดทำฐานข้อมูลของนักศึกษา	↔								
4.	จัดทำเอกสารการสอบวิชา Project 1 ในเทอม 1				↔					
5.	ศึกษาและสรุปรงานที่จะต้องทำเพิ่มเติมในเทอม 2						↔			
6.	ทำชิ้นงานตรวจลายนิ้วมือและทดสอบการใช้งานด้วย							↔		
7.	จัดทำต้นฉบับปริญญาบัตร									↔

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ประโยชน์และผลที่คาดว่าจะได้รับจากโครงการนี้

- 1) ระบบที่พัฒนาขึ้น สามารถนำไปใช้ในสถาบันการศึกษาได้ทุกระดับ
- 2) ฮาร์ดแวร์ที่ได้ทำขึ้นมาสามารถใช้ประโยชน์ได้จริงในการใช้งานจริง
- 3) ได้เรียนรู้การทำงานของ ฮาร์ดแวร์ และหลักการเขียนโปรแกรมเชื่อมต่อฮาร์ดแวร์

1.7 เนื้อหาภายในปฏิญญาพันธ

ปฏิญญาพันธฉบับนี้ประกอบไปด้วยเนื้อหาในบทต่างๆดังนี้

บทที่ 1 บทนำเป็นบทที่กล่าวถึงความเป็นมาและความสำคัญของปฏิญญาพันธ วัตถุประสงค์ ขอบเขต ประโยชน์ที่ได้รับจากการทำปฏิญญาพันธ ขั้นตอนการดำเนินงาน และอุปกรณ์ที่ใช้ในการทำปฏิญญาพันธ

บทที่ 2 นำทฤษฎีข้อมูลของเนื้อหาที่เกี่ยวข้องกับปฏิญญาพันธฉบับนี้มารวบรวมไว้ เพื่อเป็นพื้นฐานในการศึกษาหรือการทดลองตามปฏิญญาพันธนี้

บทที่ 3 การออกแบบการทดลองเป็นการออกแบบ โครงสร้างทั้งด้านซอฟต์แวร์และด้านฮาร์ดแวร์ เพื่อเตรียมให้ชุดอุปกรณ์พร้อมใช้งานจริง

บทที่ 4 เป็นการนำเสนอข้อมูลผลการทดลองตามความเป็นจริงที่ได้จากการทดลอง เพื่อเตรียมนำไปสรุปผล

บทที่ 5 บทสุดท้ายที่นำผลการทดลองในบทที่ 4 มาสรุปผล แล้วชี้แจงปัญหาและอุปสรรคที่เกิดขึ้นระหว่างการออกแบบและสร้างชุดอุปกรณ์ หรือระหว่างการทดลอง พร้อมชี้แนะแนวทางในการศึกษาและพัฒนาชุดอุปกรณ์ต่อไป

บทที่ 2

ทฤษฎีพื้นฐาน

ลายนิ้วมือของแต่ละคน เริ่มปรากฏขึ้นตั้งแต่เป็นตัวอ่อนอายุ 3 ถึง 4 เดือนในครรภ์มารดา ลายนิ้วมือไม่เปลี่ยนแปลงรูปแบบตั้งแต่แรกเกิด จนกระทั่งวันที่ตาย แต่ อาจเปลี่ยนแปลงขนาดได้ตามขนาดร่างกาย เหมือนกับการที่เราวาดรูปไว้บนลูกโป่ง ซึ่งไม่ว่าลูกโป่งจะเล็ก หรือถูกเป่าให้พองใหญ่อย่างไร ก็ยังเป็นรูปเดิมเพียงแต่มีขนาดใหญ่ขึ้นเท่านั้น

การที่ลายนิ้วมือมีรูปแบบเฉพาะในแต่ละคนนั้น จึงมีการเก็บและเปรียบเทียบลายนิ้วมือซึ่งยังไม่มีมีการพบการเหมือนกันของรายนิ้วมือ และจากการศึกษาของ Sir Francis Galton(1892) ซึ่งได้ประมาณไว้ว่า “โอกาสที่คนสองคนจะมีลายนิ้วมือเหมือนกันนั้นมีความน่าจะเป็นอยู่ที่ $1/64 \times 10^9$ ” ซึ่งเป็นการประเมินค่าโดยใช้การแบ่งรายละเอียดรูปแบบของลายนิ้วมือออกเป็นส่วนๆ และหาความน่าจะเป็นของการซ้ำกันของแต่ละส่วนนั้น แล้วนำความน่าจะเป็นของแต่ละส่วนมาคูณกันเพื่อหาความน่าจะเป็นทั้งหมด” ท่าน Sir Francis Galton ถือว่าเป็นบุคคลแรกที่ศึกษาถึงการใช้ลายนิ้วมือในการระบุตัวบุคคล และเป็นบุคคลแรกที่ทำการพิสูจน์ว่าลายนิ้วมือของแต่ละคนมีลักษณะเฉพาะ (Individuality) และไม่มีการเปลี่ยนแปลงรูปแบบ (Permanence) และยังเป็นผู้ที่กำหนดและแบ่งแยกประเภทของรูปแบบลายนิ้วมืออีกด้วย

เมื่อเราหงายมือขึ้นและสังเกตไปที่นิ้วมือ จะพบว่านิ้วนั้นมีลักษณะเป็นลาย โดยลายนิ้วมือมีลักษณะประกอบไปด้วย 2 ส่วนใหญ่คือ ส่วนที่เป็นลายนูน (Ridges) และร่อง (Valleys or Furrows) วังวนเป็นลักษณะส่วนโค้งไปรอบๆ ขวงปลายนิ้วมือสลับกันไประหว่างร่องและลายนูน ทำให้เกิดเป็นลายขึ้นมา ลายดังกล่าวนี้จะมีส่วนของจุดที่เป็น จุดเฉพาะ (Singular Point) คือ จุดกึ่งกลาง (Core) และ สันดอน (Delta) รวมทั้งลักษณะสำคัญ (Minutiae) ซึ่งก็คือ จุดปลายของลายนูน (Ridge Ending) และจุดแยกของลายนูน (Ridge Bifurcation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 ลักษณะของจุดต่างๆบนลายนิ้วมือ

2.1.1 จุดกึ่งกลาง(Core)

จุดบนเส้นโค้งของลายนิ้วมือต้องมีลักษณะเป็นโค้งขึ้น และเริ่มที่จะกลับตัว หรือเริ่มที่จะกลายเป็นโค้งลง แล้วจึงวิ่งสวนทางกลับกันกับในตอนแรกก่อนที่จะโค้งขึ้น เส้นโค้งลายนิ้วมือนี้ต้องเป็นโค้งนูนที่อยู่ชั้นในที่สุดของบริเวณที่มีลักษณะรูปแบบเป็นจุดกึ่งกลางนั้น เป็นลักษณะของเส้นโค้งขึ้นมาซ้อนๆกันหลายๆเส้น ในลักษณะของรูปมัดหวายจากโค้งวงกว้างลดลงมาเรื่อยๆจนโค้งเล็กสุด (เส้นที่อยู่ภายในที่สุดอาจไม่เป็นเส้นโค้งก็ได้) ลักษณะของรูปแบบเส้นโค้งที่อยู่ในบริเวณที่มีลักษณะรูปแบบเป็นจุดกึ่งกลาง



รูปที่ 2.1 บริเวณที่มีลักษณะเป็นจุดกึ่งกลาง

2.1.2 ลักษณะสำคัญ(Minutiae)

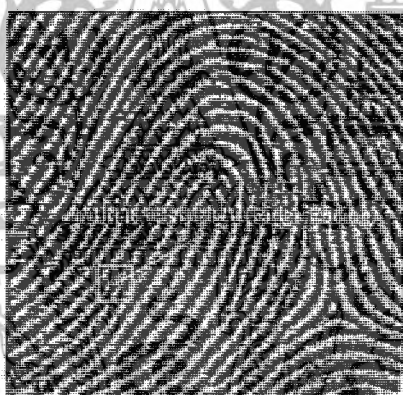
เป็นลักษณะสำคัญในรูปลายนิ้วมือ ประกอบด้วยส่วนประกอบสำคัญ 2 ชนิด ได้แก่

- จุดแยกของลายนิ้วมือ(Ridge Bifurcation) เป็นจุดที่อยู่บนลายนิ้วมือ ที่เกิดจากการแยกจากหนึ่งเส้นทางเป็นสองเส้นทาง ดังรูปที่ 2.2



รูปที่ 2.2 จุดแยกของลายนิ้ว

- จุดปลายของลายนิ้ว (Ridge Ending) เป็นจุดที่อยู่บนลายนิ้ว ในบริเวณปลายสุดของลายนิ้ว ดังแสดงไว้ในรูปที่ 2.3



รูปที่ 2.3 จุดปลายของลายนิ้ว

2.2 ชนิดและรูปแบบลายนิ้วมือ

รูปแบบลายนิ้วในภาพลายนิ้วมือที่จับกลุ่มกันเป็นรูปแบบ สามารถแบ่งได้หลายชนิด ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 กลุ่มเส้นโค้ง

ลักษณะของรูปแบบในภาพลายนิ้วมือมีลักษณะการวิ่งขึงตายนูนจากลักษณะที่ขนานกับพื้นราบแล้วพุ่งโค้งขึ้น แล้วจึงมีการวิ่งในลักษณะขนานกับพื้นราบอีกครั้ง โดยรูปแบบของกลุ่มเส้นโค้งนั้นมี 2 รูปแบบ คือ

- กลุ่มเส้นโค้งราบ(Arch)
- กลุ่มเส้นโค้งกระโจม(Tented Arch)

โดยกลุ่มเส้นโค้งทั้งสองจะต่างกันโดยลักษณะความสูงของส่วนโค้ง ดังรูปที่ 2.4 และ 2.5



รูปที่ 2.4 รูปแบบกลุ่มโค้งราบ(Arch)



รูปที่ 2.5 รูปแบบเส้นโค้งกระโจม(Tented Arch)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 กลุ่มมัดหวาย(Loop)

- กลุ่มมัดหวายเอียงขวา(Right Loop)

มีลักษณะของการวิ่งของลายนิ้วจากบริเวณด้านซ้ายมาที่บริเวณกลางนิ้วมือแล้ววิ่งโค้งขึ้นไปด้านบน แล้วจึงโค้งลงและวิ่งวนกลับไปตามทางเดิมที่ได้วิ่งมา เมื่อสังเกตด้วยตาเปล่าจะเห็นว่าเมื่อมีการจับกลุ่มเป็นรูปแบบคล้ายกับมัดหวาย แล้วจะเอนเอียงไปทางด้านขวา จึงได้เรียกว่า “กลุ่มมัดหวายเอียงขวา” และมีสันครออยู่ทางด้านขวาของรูปแบบ



รูปที่ 2.6 รูปแบบกลุ่มมัดหวายเอียงขวา(Right Loop)

- กลุ่มมัดหวายเอียงซ้าย(Left Loop)

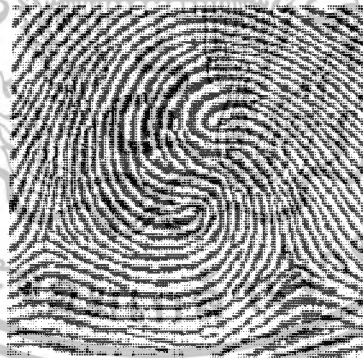
รูปแบบจะคล้ายลักษณะของกลุ่มมัดหวายเอียงขวา คือ มีการวิ่งของลายนิ้วจากทางด้านขวาที่บริเวณตรงกลางของนิ้วมือ จากนั้นจะวิ่งขึ้นไปด้านบนแล้วโค้ง และวิ่งกลับไปตามทางเดิมที่วิ่งมา เมื่อทำการสังเกตจะพบว่า มีลักษณะรูปแบบคล้ายกับมัดหวาย และรูปแบบดังกล่าวจะมีการเอียงตัวไปทางด้านซ้าย เราจึงเรียกว่า “กลุ่มมัดหวายเอียงซ้าย” ปกติมักจะมีสันครออยู่ทางด้านซ้ายของรูปแบบเช่นกัน



รูปที่ 2.7 รูปแบบกลุ่มมัดหอยเอียงซ้าย(Left Loop)

- กลุ่มมัดหอยคู่(Double Loop)

เป็นกลุ่มของรูปแบบที่เกิดจากการรวมกันของกลุ่มมัดหอยเดี่ยวสองกลุ่มรวมตัวกันเป็นรูปแบบเดียว โดยลักษณะการวิ่งของลายนิ้วนอาจมาจากทางเดียวกันแล้วรวมตัวกันเป็นลักษณะของกลุ่มมัดหอยเดี่ยวสองกลุ่ม โดยที่กลุ่มหนึ่งจะมีรูปแบบเป็นลักษณะของโค้งขึ้นและกลุ่มมัดหอยอีกกลุ่มหนึ่งจะมีลักษณะโค้งลง เมื่อรวมกันเป็นกลุ่มมัดหอยคู่ รูปแบบดังกล่าวอาจมีการเอียงซ้าย หรือเอียงขวา

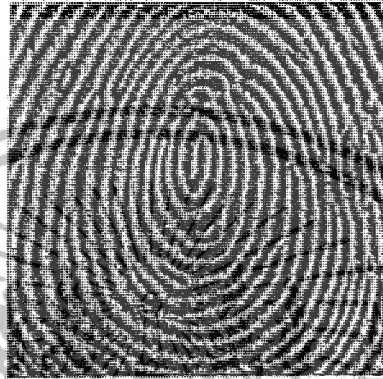


รูปที่ 2.8 รูปแบบกลุ่มมัดหอยคู่(Double Loop)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 กลุ่มกันหอย(Whorl)

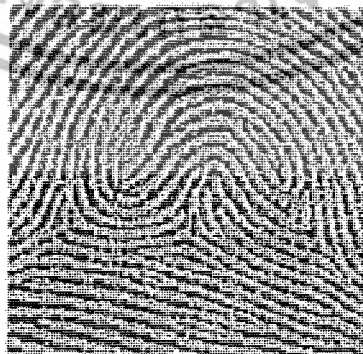
ลักษณะการวิ่งของลายนิ้วมือมีรูปแบบเป็นลักษณะเส้น โค้งเป็นรูปวงกลมจากวงใหญ่ ค่อยๆเล็กลงไปเรื่อยๆ จนเหลือวงกลมวงเล็กที่สุด



รูปที่ 2.9 รูปแบบกลุ่มกันหอย(Whorl)

2.2.4 กลุ่มซับซ้อน(Accidental Whorl)

ลายนิ้วมือที่มีลักษณะพิเศษที่ไม่จัดเข้าเป็นลายนิ้วมือชนิดใดโดยเฉพาะ ประกอบด้วยลายนิ้วมือ 2 แบบมาผสมกัน และมีสันคร 2 สันคอน หรือมากกว่า เช่นกรณีที่ไม่สามารถเข้ากับลายนิ้วมือกลุ่มที่กล่าวมาแล้วข้างต้นไม่ได้เลย โดยมีความยุ่งเหยิง และเป็นรูปแบบที่ไม่แน่นอน

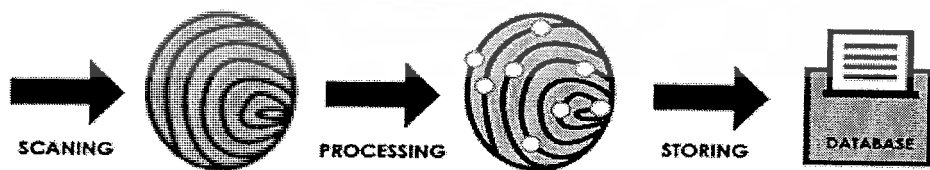


รูปที่ 2.10 รูปแบบกลุ่มซับซ้อน(Accidental Whorl)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หลักการวิเคราะห์ลายนิ้วมือ

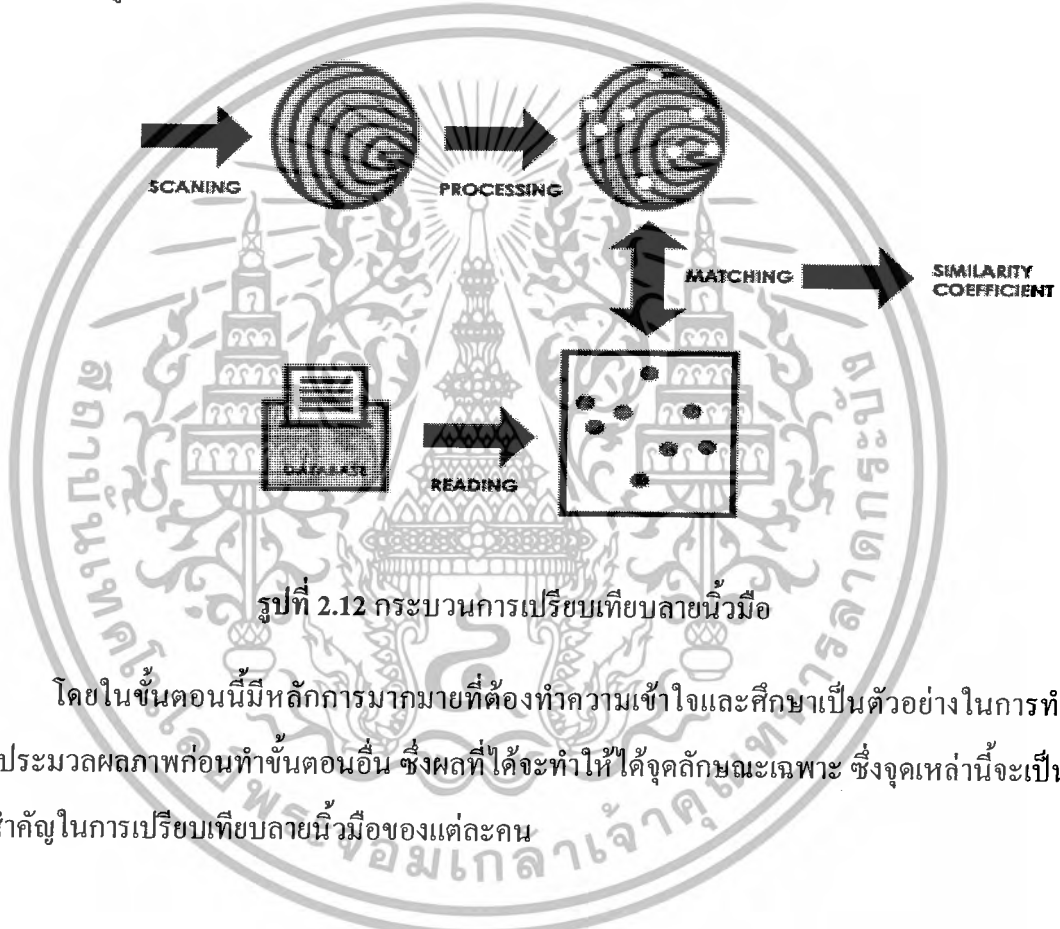
การวิเคราะห์ลายนิ้วมือของนั้น จะเริ่มด้วยการนำลายนิ้วมือของแต่ละบุคคลแต่ละนิ้ว มาหาจุดลักษณะเฉพาะที่สำคัญกระบวนการแรกเริ่มของการตรวจพิสูจน์ลายนิ้วมือ คือ การอ่านภาพลายนิ้วมือเข้ามาเก็บไว้ในหน่วยความจำถาวร ซึ่งในส่วนนี้จะใช้หน่วยความจำถาวร(EEPROM) เป็นส่วนที่เก็บข้อมูลไว้ โดยข้อมูลที่อ่านหรือสแกนเข้ามานั้นจะนำมาผ่านการประมวลผล (Processing) ก่อนแล้วจึงเก็บข้อมูลนั้นไว้ ซึ่งข้อมูลนี้จะถูกเก็บไว้เป็นต้นแบบหรือรหัสของผู้ใช้แต่ละคนในขั้นตอนก่อนที่จะนำลายนิ้วมือเข้าไปเก็บนั้นจะต้องผ่านขั้นตอนของการประมวลผล(Pre-Processing) ก่อนในกระบวนการนี้จะทำให้ภาพที่ได้รับการสแกนเข้ามาเกิดความสมบูรณ์มากขึ้น เพราะเมื่อเครื่องได้รับการสแกนภาพเข้ามาแล้ว ภาพที่อ่านได้อาจไม่ชัดเจน พร่าเลือน ก็จะทำให้การประมวลผลในขั้นตอนถัดไปทำได้ด้วยความยากลำบากหรือทำไม่ได้ ซึ่งจะทำให้ผลที่ได้ก็อาจไม่ถูกต้อง เมื่อเกิดปัญหาเช่นนี้จึงได้มีการกระทำหลายกระบวนการด้วยกัน คือ การกำจัดสัญญาณรบกวน, การปรับความมืด-สว่าง และความแตกต่างของตัวภาพและฉากของภาพ, การแปลงภาพเป็นภาพสองระดับ(Binary), การทำให้เส้นลายนิ้วมือบาง(Thinning) และอื่นๆ อีกมาก ซึ่งกระบวนการจะมากหรือน้อยขึ้นอยู่กับว่าตัวอุปกรณ์นั้นมีการอ่านค่าลายนิ้วมือที่ได้ภาพออกมาละเอียดและสมบูรณ์แค่ไหน เมื่อได้ลายนิ้วมือที่ผ่านการประมวลผลแล้ว ก็จะนำข้อมูลหรือภาพนี้ไปจัดเก็บใน EEPROM ที่สามารถลบข้อมูลใหม่ด้วยไฟฟ้า โดยภาพที่ถูกจัดเก็บไว้นี้จะถูกเก็บไว้เพื่อใช้ในการเปรียบเทียบกับลายนิ้วมือที่ได้รับการสแกนเข้ามาเมื่อนำตัวอุปกรณ์นี้ไปใช้งานรูปที่ 2.11 แสดงกระบวนการทำงาน โดยเริ่มด้วยการสแกนลายนิ้วมือเข้ามาแล้วนำภาพที่ได้ผ่านการประมวลผลซึ่งจะได้ภาพที่มีประสิทธิภาพมากขึ้นแล้วจึงเก็บภาพนั้นไว้



รูปที่ 2.11 กระบวนการบันทึกลายนิ้วมือ

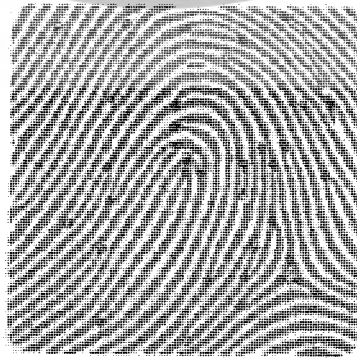
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเมื่อตัวอุปกรณ์ได้เก็บลายนิ้วมือของผู้ที่จะนำไปใช้แล้วขั้นตอนในการใช้ก็จะคล้ายกับขั้นตอนการอ่านลายนิ้วมือเข้ามาเก็บไว้ แต่การอ่านเข้ามาครั้งนี้ข้อมูลที่ได้จะถูกนำไปเก็บไว้ในหน่วยความจำชั่วคราว(RAM) หลังจากสแกนเข้ามาแล้วประมวลผลแล้วก็จะทำการเก็บข้อมูลไว้ในส่วนของหน่วยความจำชั่วคราว ส่วนในขั้นตอนถัดไปก็จะนำข้อมูลที่เก็บอยู่ในส่วนของหน่วยความจำถาวร กับส่วนที่เก็บอยู่ในหน่วยความจำชั่วคราวมาทำการเปรียบเทียบกัน เมื่อได้ผลแล้วก็จะแจ้งผลให้ผู้ใช้ทราบว่ามีความเหมือนกันมากน้อยแค่ไหน



รูปที่ 2.12 กระบวนการเปรียบเทียบลายนิ้วมือ

โดยในขั้นตอนนี้มีหลักการมากมายที่ต้องทำความเข้าใจและศึกษาเป็นตัวอย่างในการทำการประมวลผลภาพก่อนทำขั้นตอนอื่น ซึ่งผลที่ได้จะทำให้ได้จุดลักษณะเฉพาะ ซึ่งจุดเหล่านี้จะเป็นสิ่งสำคัญในการเปรียบเทียบลายนิ้วมือของแต่ละคน



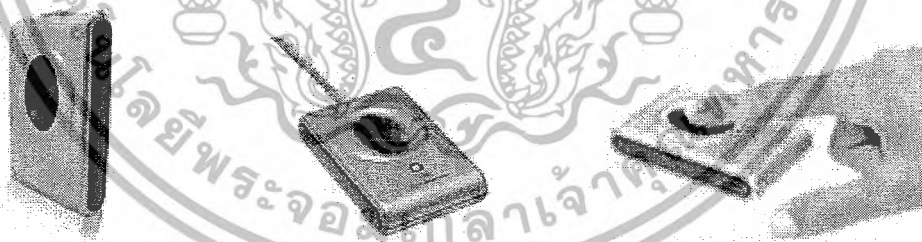
รูปที่ 2.13 จุดลักษณะเฉพาะบนลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.13 ได้แสดงให้เห็นถึงจุดลักษณะเฉพาะถึง 37 จุด ซึ่งถือเป็นลักษณะพิเศษของลายนิ้วมือแต่ละนิ้วแต่ละคน โดยไม่มีทางที่จะซ้ำกันได้เลย แต่ในการใช้งานเปรียบเทียบลายนิ้วมือจริงๆนั้น ระบบจะใช้แค่เพียง 8 จุดก็เพียงพอที่จะยืนยันตัวบุคคลได้แล้ว

2.4 คุณสมบัติของเครื่อง U.R.U 4000

- มีขนาดเล็ก
- คุณภาพในการบันทึกภาพสูง
- ปฏิเสธลายนิ้วมือแฝง
- สามารถอ่านลายนิ้วมือได้ที่ค่าผิดพลาดถึง 30 องศา
- ใช้งานได้ดีกับลายนิ้วมือที่ ชื้น แห้ง หรือ หยาบ
- ใช้งานได้ดีกับแอปพลิเคชัน URU ทั้งหมด
- สนับสนุนระบบปฏิบัติการ Windows 98, Me, NT 4.0, 2000 และ XP
- สามารถนำไปพัฒนาร่วมกับชุดพัฒนาโปรแกรม SDK (Software Develop Kit) เพื่อทำเป็นโปรแกรมต่างๆ ได้ตามความต้องการ (สามารถอ่านรายละเอียดได้ที่หน้า Biokey SDK)



รูปที่ 2.14 เครื่องสแกนนิ้ว U.R.U 4000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

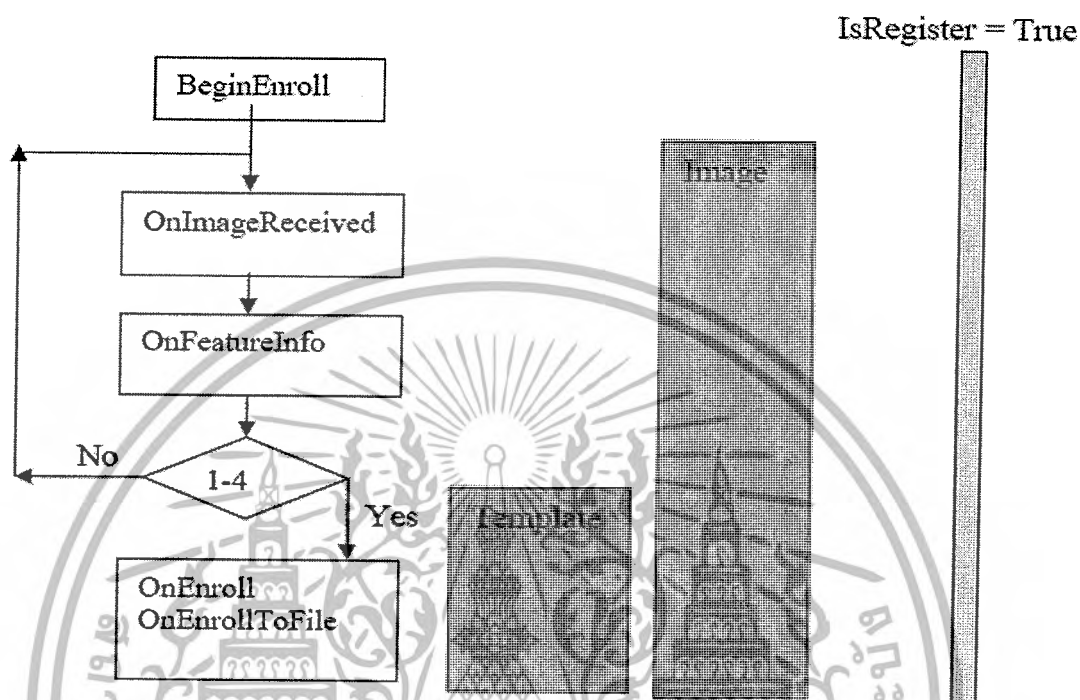
2.4.1 หลักการทำงานของเครื่อง U.R.U 4000

Component ที่ในการสืบค้นลายนิ้วมือ คือ Biokey.OCX มีลักษณะการทำงานดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังแสดงการรับค่าลายนิ้วมือเข้ามาทางเครื่องอ่านลายนิ้วมือ แสดงได้ดังภาพ



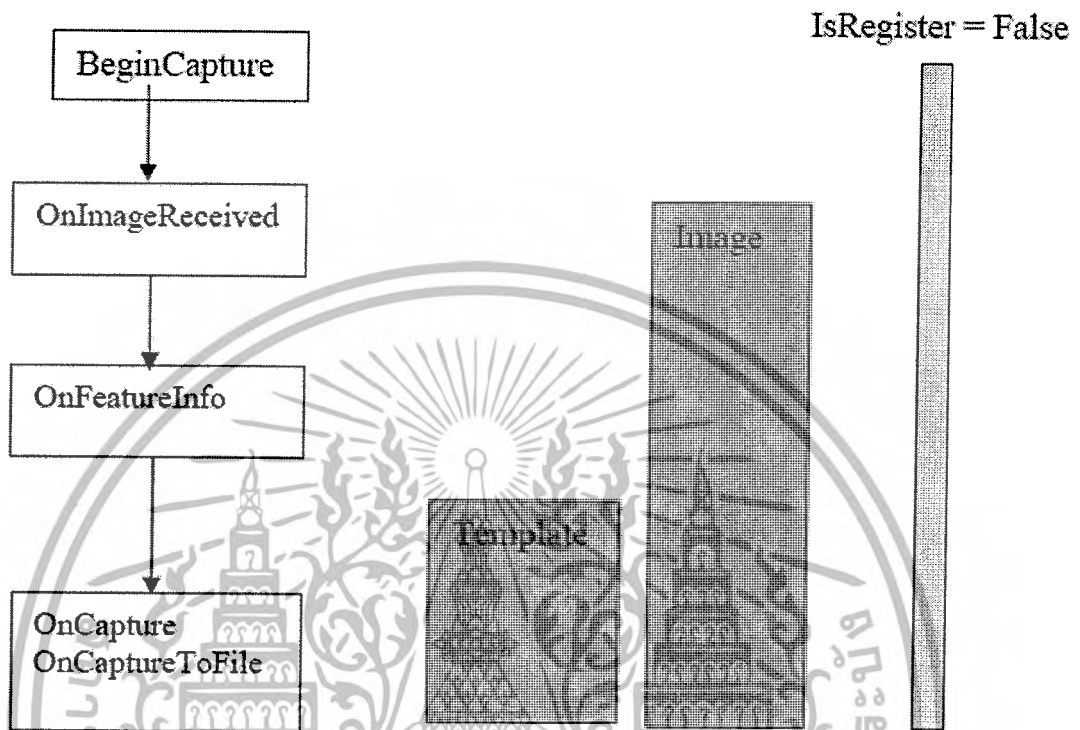
รูปที่ 2.16 ขั้นตอนการอ่านลายนิ้วมือ และ บันทึก ของคอมโพเนนต์ Biokey.ocx

ในการเรียกใช้ Component ของเครื่องอ่านลายนิ้วมือนั้นจำเป็นต้องกำหนดค่าเริ่มต้นให้กับเครื่องอ่านลายนิ้วมือก่อน จากนั้นจึงเรียกใช้ ฟังก์ชันต่างๆ ตามที่ระบุใน โฟซาร์ต ได้แก่

- ฟังก์ชัน BeginEnroll คือฟังก์ชันเริ่มต้นการบันทึกลายนิ้วมือลงในหน่วยความจำในเครื่องอ่านลายนิ้วมือ
- ฟังก์ชัน OnImageReceived คือฟังก์ชันรับรูปภาพลายนิ้วมือจากเครื่องอ่านลายนิ้วมือ
- ฟังก์ชัน OnFeatureInfo คือฟังก์ชันที่ตรวจสอบคุณภาพของลายนิ้วมือก่อนที่จะนำไปเข้ารหัส
- ฟังก์ชัน OnEnroll หรือ OnEnrollToFile คือฟังก์ชันรับลายนิ้วมือและนำมาเข้ารหัสซึ่งสามารถเก็บได้ทั้ง Template หรือ อักษร
- ค่าที่คืนมาจากฟังก์ชัน IsRegister จะมีค่าเป็นขณะที่มีการบันทึกลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังแสดงการสืบค้นข้อมูลจากลายนิ้วมือ แสดง ได้ดังรูป



รูปที่ 2.17 แสดงการตรวจสอบลายนิ้วมือ ของคอมพิวเตอร์ Biokey.ocx

2.5 โปรแกรมจัดการฐานข้อมูล Microsoft Access

2.5.1 ฐานข้อมูล(Database)

ฐานข้อมูล (Database) คือ กลุ่มข้อมูลที่เรารวบรวมไว้เพื่อจุดประสงค์อย่างใดอย่างหนึ่ง เช่น เพื่อติดตามการสั่งซื้อ(Orders) ของลูกค้า(Customers) เพื่อเก็บข้อมูลเพลง(CD เพลง) ที่สะสมไว้ให้เป็นระบบระเบียบ ฯลฯ ฐานข้อมูลช่วยให้เราจัดการข้อมูล ใช้ข้อมูล ให้บรรลุจุดประสงค์ได้ง่าย สะดวก รวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ฐานข้อมูลอยู่ในคอมพิวเตอร์ดีกว่าอยู่ในสมุดหรือแฟ้มกระดาษ

ถ้าเราไม่ได้จัดเก็บฐานข้อมูลไว้ในคอมพิวเตอร์ เราก็ต้องติดตามข้อมูลจากหลายแหล่ง ซึ่งต้องจัดระเบียบด้วยมือ(manual) เช่น ถ้าเก็บหมายเลขโทรศัพท์ลูกค้าไว้ในแฟ้มนามบัตร ในแฟ้มข้อมูลสินค้า และในบัญชีสั่งซื้อ ต่อมาหมายเลขโทรศัพท์ของลูกค้าคนหนึ่งมีการเปลี่ยนแปลง เราก็ต้องแก้ไขทั้งสามแห่ง แต่ถ้าฐานข้อมูลของเราอยู่ในคอมพิวเตอร์และเราจัดเก็บอย่างถูกวิธี เราแก้ไขข้อมูลนั้นเพียงหนึ่งแห่งเท่านั้น หมายเลขโทรศัพท์ของลูกค้าคนนั้นจะถูกปรับปรุง โดยอัตโนมัติในทุกแห่งทันทีทันใด

2.5.3 ประโยชน์ของระบบการจัดการฐานข้อมูล

ระบบการจัดการฐานข้อมูล(Database Management System: DBMS) ย่อมหมายถึงระบบที่ใช้คอมพิวเตอร์ช่วยในการจัดการฐานข้อมูล มีประโยชน์มากดังนี้

- รวดเร็ว
- ลดความผิดพลาด
- คล่องตัว จะเพิ่ม ลบ แก้ไข ทำได้ง่าย
- จัดทำรายงานได้ง่าย - ทำเมื่อใดก็ได้ ทันการ ทันเวลา เป็นปัจจุบัน(Update)
- ช่วยการวิเคราะห์ที่ได้ชัดเจน เพราะจัดเรียง(sort) กรอง(filter) ค้นหา นับ ฯลฯ ทำได้ง่าย
- ประหยัดเวลาและแรงงาน ในระยะยาว จึงเป็นการประหยัดค่าใช้จ่ายด้วย
- ความปลอดภัยของข้อมูลสูง เพราะทำสำเนา(back up) ได้ง่าย ทั้งที่เป็นแผ่นดิสก์ และ เอกสารที่พิมพ์ออกจากเครื่องคอมพิวเตอร์

2.5.4 การออกแบบฐานข้อมูล (Database Design)

การออกแบบฐานข้อมูล (Database) ที่ดีทำให้มั่นใจได้ว่าฐานข้อมูลของเราจะง่ายต่อการดูแลรักษา เราควรเก็บข้อมูลต่าง ๆ ในตาราง (Tables) ต่าง ๆ ที่เหมาะสม และในแต่ละตารางควรประกอบด้วยข้อมูลเพียงเรื่องเดียวเท่านั้น เช่น ข้อมูลที่เกี่ยวกับลูกค้า ก็เก็บในตารางลูกค้า(Customers) อย่าเก็บไว้ที่อื่นอีก ดังนั้นเมื่อเราปรับปรุงข้อมูลบางอย่าง เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมาจากข้อมูลทั้งหมด (เรียกว่า dataset / subset) เช่น แสดงลูกค้าเฉพาะในจังหวัดตาก เท่านั้น หรือข้อมูลจากการรวมกันของข้อมูลจากตารางอื่น ๆ ตัวอย่าง เช่น ข้อมูลการสั่งซื้อ (Orders) พร้อมกับข้อมูลลูกค้า (Customers) ที่สั่งซื้อ

2.5.5 Flat-file databases และ Relational databases

Flat-file databases(ฐานข้อมูลเรียบ / ตารางเดียว) คือฐานข้อมูลที่ประกอบด้วย ตารางข้อมูล (Table) เพียงตารางเดียว กรอกทุกสิ่งทุกอย่างลงตารางนี้ เช่น นางบุญ บารมี ซื้อสินค้า 100 ครั้ง ก็ต้องกรอกชื่อนาง 100 ครั้งพร้อมที่อยู่ (Address) 100 ครั้งด้วย เป็นการซ้ำซ้อน มาก ไม่มีประสิทธิภาพ

Relational databases(ฐานข้อมูลสัมพันธ์ / หลายตาราง) ประกอบด้วยตารางข้อมูล หลายตาราง แต่ละตารางกรอกข้อมูลที่เหมาะกับตารางนั้นๆ และแต่ละตารางมีความสัมพันธ์กับตารางอื่น อย่างน้อยหนึ่งตารางเสมอ ฐานข้อมูลแบบ Relational databases จึงดีกว่า เพราะการเก็บข้อมูลไม่ซ้ำซ้อน ส่งผลให้ฐานข้อมูลถูกต้อง เป็นที่เชื่อถือ ถิ่นเนื้อที่ฮาร์ดดิสก์น้อยกว่า การกรอกข้อมูล แก้ไข ปรับปรุง (update) ง่าย สะดวก มีประสิทธิภาพ

2.5.6 ขั้นตอนการออกแบบฐานข้อมูล

การออกแบบฐานข้อมูล ควรมีขั้นตอน ดังต่อไปนี้

- กำหนดข้อมูล(data)
- กำหนดเขตข้อมูล(fields)
- กำหนดตารางข้อมูล(tables)
- ปรับตารางข้อมูลให้อยู่ในรูปแบบปกติ(normalization)
- กำหนดคีย์หลัก(primary key)
- กำหนดความสัมพันธ์(relationship) ระหว่างตารางข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7 วัตถุฐานข้อมูล (Database Objects)

เครื่องมืออำนวยความสะดวกในการจัดการฐานข้อมูล Microsoft Access วัตถุฐานข้อมูล (Objects) ได้แก่

- Tables(ตาราง) ใช้เก็บข้อมูล
- Queries(แบบสอบถาม) ใช้ค้นหาหรือเรียกแต่เพียงข้อมูลที่เราต้องการ
- Reports (รายงาน) ใช้พิมพ์ แสดง เสนอข้อมูล หรือใช้วิเคราะห์ข้อมูล
- Modules(โมดูล) ใช้สร้าง โปรแกรมย่อย(procedures) ภาษา Visual BASIC ในฐานข้อมูล



รูปที่ 2.18 ตัวอย่าง Objects

2.5.8 Tables (ตารางข้อมูล)

Table คือ ตารางข้อมูล เป็นออบเจกต์หนึ่ง หรือองค์ประกอบหนึ่ง ที่สำคัญที่สุดของฐานข้อมูล ออบเจกต์อื่น ๆ สามารถทำงาน ที่เป็นขึ้นเป็นอันได้ ก็ล้วนต้องอิง Table ทั้งสิ้น กล่าวได้ว่า ถ้าปราศจาก Table แล้วไซ้ ฐานข้อมูลก็จะไร้ประโยชน์

สามารถสร้าง Table โดยไปที่ Tables > Create table in design view > Design ต่อจากนี้ ให้กรอกชื่อฟิลด์ เลือกชนิดข้อมูล และ ระบุขนาดฟิลด์

customerID	name	address	province
0001	นายชม มิชัย	12 ถนนมิชัย อ.เมือง	หนองคาย
0002	นายพร ชวนชม	5/7 ถนนพหลโยธิน อ.เมือง	อุบลราชธานี
0003	นางมณีน บารมี	245 ถนนกรุงเก่า อ.เมือง	อุบลราชธานี
0004	นายกวง แซ่ตั้ง	55 ถนนสุรวงศ์ เขตบางรัก	กรุงเทพฯ

รูปที่ 2.19 ตัวอย่าง Table อย่างง่าย (เมื่อกรอกข้อมูลแล้ว)

Field Name	Data Type	Description
customerID	Text	
name	Text	
address	Text	
province	Text	

Field Properties

General Lookup

Field Size: 4 ระบุขนาดของฟิลด์

Format

Input Mask

A field name can be

รูปที่ 2.20 หน้าต่างใช้สร้างโครงสร้างของ Table

2.5.9 การออกแบบ Tables

การออกแบบตารางข้อมูล ประกอบด้วยขั้นตอนต่าง ๆ ดังนี้

- ทำความเข้าใจให้ถ่องแท้ทั้งระบบ อันประกอบด้วย ข้อมูลที่จะแสดง เช่น Reports ที่ต้องการพิมพ์ออก (Output) ควรจะมีหน้าตาอย่างไรเป็นต้น
- ตัดสินใจ หรือกำหนดให้แน่ชัดว่า จำเป็นต้องสร้างกี่ Table และแต่ละ Table จะต้องกรอก (Input) ข้อมูลอะไรบ้าง
- กำหนดฟิลด์ต่าง ๆ ชนิดของแต่ละฟิลด์ และ ขนาดความกว้างของแต่ละฟิลด์
- กรอกข้อมูลสมมุติ เพื่อทดสอบ Table ที่เพิ่งสร้างขึ้น ดูว่าใช้ได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.10 ความสัมพันธ์ระหว่างสอง Tables

สร้างความสัมพันธ์(Relationships) ระหว่างเทเบิล customers กับเทเบิล orders ทำดังนี้

- คลิกปุ่มความสัมพันธ์ที่ทูลบาร์ > เลือก customers > Add > เลือก orders > Add > Close > กดเซฟฟิลด์ customerID(คีย์หลัก) ของเทเบิลแม่ ลากไปยังฟิลด์ customerID(คีย์นอก) ของเทเบิลลูก > Enforce Referential Integrity > Create เส้นความสัมพันธ์จะปรากฏให้เห็น



ฟิลด์ต่างๆ ในเทเบิล

ฟิลด์ต่างๆ ในเทเบิล

รูปที่ 2.21 ตัวอย่าง Relationships

2.5.11 Queries (แบบสอบถาม)

Queries คือ "คำถาม" ที่เราใช้ถามฐานข้อมูล(Database) เช่นถามว่า ลูกค้าแต่ละคน สั่งซื้อสินค้าแต่ละอย่าง เป็นเงินเท่าใด ผลบวกทั้งหมดของยอดขายเป็นเท่าใด บุคลากรคนใดบ้างที่มีเงินเดือนเกิน 8,000 บาท คะแนนเฉลี่ยของนักเรียนทั้งห้องเป็นเท่าใด มีลูกค้านคนใดบ้างที่อยู่ในจังหวัด กรุงเทพฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Select Queries(คิวรีคัดเลือกข้อมูลที่ต้องการ)

เมื่อต้องการค้นหา หรือ คัดเลือกเฉพาะข้อมูลที่ตรงกับ เงื่อนไข(Criteria) ที่เรา ระบุ หรือ ถาม ก็จงสร้าง Queries ขึ้นใช้



รูปที่ 2.22 แสดงการสร้าง Query

- Action Queries(คิวรีปฏิบัติการ)

เราสามารถใส่ปรับปรุง แก้ไข หรือ ลบระเบียน(Records) ใน Table ได้หลาย ระเบียน ในเวลาเดียวกัน โดยใช้ Action Queries(คิวรีปฏิบัติการ) ทำให้ข้อมูลใน Table เปลี่ยนแปลง(Update) ถาวร ดังนั้น จึงควรใช้ Action Queries อย่างระมัดระวัง

เรายังสามารถใช้ Queries สำหรับ ดู และวิเคราะห์ข้อมูลด้วยวิธีที่แตกต่างกันหลาย วิธี และยังสามารถใช้ Queries เป็นแหล่งระเบียน(Records) สำหรับสร้างฟอร์ม(Forms) รายงาน(Reports) และ เเพจ(Pages) ทั้งนี้ Microsoft Access มี Queries อยู่หลายชนิดด้วยกัน ที่เหมาะสมและสะดวกกับการใช้งานต่าง ๆ ที่ว่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.12 Reports(รายงาน)

Reports(รายงาน) มีจุดประสงค์หลัก คือ ใช้พิมพ์ให้ปรากฏบนแผ่นกระดาษ จึงมีข้อดี ที่สามารถเพิ่มสีสันและลูกเล่นเข้าไปได้มาก เพื่อให้สิ่งพิมพ์นั้นน่าดู น่าใช้ เช่น สามารถตกแต่งฟอนต์ให้สวยงาม ใส่รูปภาพ ใส่สีตัวหนังสือ สีพื้นหลัง(background) เป็นต้น

- เมื่อต้องการวิเคราะห์ข้อมูล หรือ นำเสนอข้อมูลด้วยรูปแบบเฉพาะในเอกสารที่พิมพ์ ก็จงสร้าง Report (รายงาน) ขึ้นใช้ ตัวอย่าง เช่น เราอาจพิมพ์ Report ที่จัดกลุ่มข้อมูลและคำนวณผลรวม (ใบเสร็จรับเงิน) หรือ Report ที่มีรูปแบบป้ายผนึก (Mailing Label) สำหรับติดหน้าซองจดหมายก็ได้

<i>Customers</i>			
<i>customerID</i>	<i>name</i>	<i>address</i>	<i>province</i>
0001	นายชม มีชัย	12 ถนนมีชัย อ.เมือง	หนองคาย
0002	นายพร ชวนชม	5/7 ถนนพหลโยธิน อ.เมือง	อยุธยา
0003	นางบุญ บารมี	245 ถนนกรุงเก่า อ.เมือง	อยุธยา
0004	นายกวง แซ่ตั้ง	55 ถนนสุรวงศ์ เขตบางรัก	กรุงเทพฯ

รูปที่ 2.23 หน้าตาของ Report อย่างง่าย

Labels customers : Report

To: นายชม มีชัย 12 ถนนมีชัย อ.เมือง หนองคาย 43000	To: นายพร ชวนชม 5/7 ถนนพหลโยธิน อ.เมือง อยุธยา 13000
To: นางบุญ บารมี 245 ถนนกรุงเก่า อ.เมือง อยุธยา 13000	To: นายกวง แซ่ตั้ง 55 ถนนสุรวงศ์ เขตบางรัก กรุงเทพฯ 10500

รูปที่ 2.24 Mailing Label(ฉลากจำหน่ายหน้าซองจดหมาย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reports นำเสนอข้อมูลจาก Table หรือ Query

- Reports เป็นวิธีที่ดีวิธีหนึ่งในการนำเสนอข้อมูลของเราด้วยรูปแบบที่ใช้ในการพิมพ์ เนื่องจากเราสามารถควบคุมขนาดและลักษณะที่ปรากฏของทุกสิ่งบน Reports ได้ เราจึงสามารถแสดงข้อมูลได้ตามรูปแบบที่เราต้องการ
- Reports ส่วนใหญ่จะผูกอยู่กับตาราง(Tables) และแบบสอบถาม(Queries) ในฐานข้อมูล แหล่งระเบียบ(Records) ของ Reports จะอ้างถึงเขตข้อมูล(Fields) ในตารางและแบบสอบถามที่เป็นต้นต่อข้อมูล Reports ไม่จำเป็นต้องประกอบด้วยข้อมูลทั้งหมดจากตารางหรือแบบสอบถามที่ Reports นั้นยึดตาม
- Reports ที่ผูกไว้จะได้รับข้อมูลจากแหล่งระเบียบต้นต่อ(source) ของ Reports นั้น รายละเอียดอื่น ๆ บน Reports เช่น ชื่อเรื่อง วันที่ และหมายเลขหน้า จะถูกเก็บไว้ในการออกแบบ Reports

2.6 Microsoft Visual Studio

Microsoft Visual Studio คือ "ชุดพัฒนาโปรแกรม (Integrated Development Environment) ประกอบด้วยโปรแกรมหลายๆ ตัวที่ใช้ในการสร้างโปรแกรมสำเร็จรูปบนระบบปฏิบัติการวินโดวส์ หรือใช้สร้างเว็บโปรแกรม สร้างเว็บบริการ จัดการฐานข้อมูล และอื่น ๆ อีกมากมาย

Visual Studio เป็นชุดที่รวบรวมเครื่องมือพัฒนาต่างๆ ที่ใช้ในการพัฒนาโปรแกรม ตั้งแต่หน้าที่ใช้พัฒนาโปรแกรม (development interface), เครื่องมือในการตรวจหาจุดผิดในโปรแกรม (debugging tool), ตัวช่วยอัตโนมัติใช้ในการเขียนโปรแกรม (wizard tool), ตัวจัดการฐานข้อมูล (database management) และส่วนประกอบอื่นๆ ที่จำเป็นในการพัฒนาโปรแกรม นำมาประกอบรวมกันเป็นชุด ซึ่งเราจะรวมเรียกชุดนี้ว่า Integrated Development Environment (ชุดพัฒนาโปรแกรม) หรือเรียกย่อๆ ว่า IDE

2.6.1 Visual Studio

เป็นชุดพัฒนาจากบริษัทไมโครซอฟท์(Microsoft) ใช้ในการสร้างสิ่งต่างๆ ดังต่อไปนี้

- โปรแกรม(program) สามารถสร้างโปรแกรมที่ทำงานบนวินโดวส์ทั้งแบบ 32 บิตและ 64 บิต, โปรแกรมที่ทำงานบนพ็อคเก็ตพีซี(Pocket PC) หรือ สมาร์ทโฟน (smartphones)
- เว็บไซต์(website) สามารถสร้างหน้าจอบริบทเว็บไซค์ได้ ทั้งแบบ HTML,ASP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมเว็บ(web application) บางครั้งเรียกย่อๆ ว่า "webapp" หรือ "weblication" คือ โปรแกรมที่เรียกผ่านเว็บเบราว์เซอร์ โปรแกรมประเภทนี้ เราไม่ต้องติดตั้งโปรแกรมบนเครื่องเรา แค่เปิดเบราว์เซอร์คลิกไปที่โปรแกรม นั้นอยู่ โปรแกรมจะโหลดขึ้นมาทางหน้าจอบริษัทให้เราได้ใช้งานได้ เช่น โปรแกรมรับส่งเมลล์ในเว็บไซด์ hotmail เราสามารถเข้าไปอ่านเมลล์ ลบเมลล์ ส่งเมลล์ได้ นี่ตัวอย่างโปรแกรมบนเว็บที่เราจักกันดี
- บริการบนเว็บ(web service) คือชุดที่รวมหลายๆ เครื่องมือเข้าไว้ด้วยกัน เครื่องมือหลักก็คือเครื่องมือที่ใช้ในการเขียน โปรแกรมภาษาต่างๆ Visual Studio สามารถเขียนโปรแกรมจากภาษาใดก็ได้ที่เราหนด โดยใช้เครื่องมือต่างๆ ดังต่อไปนี้

2.6.2 เครื่องมือพัฒนาโปรแกรมใน Visual Studio

- Visual Basic ใช้พัฒนาโปรแกรมด้วยภาษาเบสิก สำหรับผู้ที่มีพื้นฐานภาษาเบสิก
- Visual C++ ใช้พัฒนาโปรแกรมด้วยภาษาซีพลัสพลัส สำหรับผู้ที่มีพื้นฐานภาษาซี
- Visual C# ใช้พัฒนาโปรแกรมด้วยภาษาซีชาร์ป สำหรับผู้ที่มีพื้นฐานภาษาซี
- Visual j# ใช้พัฒนาโปรแกรมด้วยภาษาเจชาร์ป สำหรับผู้ที่มีพื้นฐานภาษาจาวา (java)
- Visual InterDev ใช้ในการสร้างเว็บแอปพลิเคชัน (web application) การเขียน ASP และการสร้างหน้าเว็บ HTML
- Visual J++ สำหรับผู้ที่ชอบการเขียนโปรแกรมภาษาจาวา
- Visual FoxPro สำหรับผู้ที่ชอบการเขียนโปรแกรมจัดการฐานข้อมูลด้วยฟ็อกซ์
- Microsoft SQL Server ตัวจัดการฐานข้อมูล (database management) ใช้ในการสร้างฐานข้อมูล เพื่อเก็บข้อมูลในโปรแกรมที่เราสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 Visual Basic ใน Access

Visual Basic คือ ชุดคำสั่งที่ใช้บริหาร และควบคุมการทำงานของโปรแกรมและอ็อบเจกต์ ซึ่ง Visual Basic เป็นระบบภาษาแบบ Object Oriented Programming เพื่อสนับสนุนการใช้งานอ็อบเจกต์ต่างๆ ภายในโปรแกรม Access เช่น คิวรี่ ฟอรั่ม หรือฐานข้อมูล แม้กระทั่งตัว Access สามารถนับเป็นอ็อบเจกต์หนึ่งได้เช่นกัน

อ็อบเจกต์จะต้องมีค่าคุณสมบัติของตัวเอง เช่น คุณสมบัตินี้ของฟอรั่ม สามารถกำหนดแบบของฟอรั่ม Event การป้อนข้อมูลหรือ Text Box ในฟอรั่ม สามารถกำหนดชนิดตัวอักษร แหล่งข้อมูล ฯลฯ

2.7.1 การทำงานของ Visual Basic

การนำ Visual Basic ไปใช้งานกับอ็อบเจกต์ โดย Method และ Event

- คำสั่ง คือ คำสั่งในการทำงานกับอ็อบเจกต์
- Event คือ สภาพแวดล้อมของอ็อบเจกต์หรือตัว Control เพื่อใช้ควบคุมการทำงานของโปรแกรม ตัวอย่างเช่น เมื่อเปิดฟอรั่มให้คำสั่งทำงาน เช่น ให้มีข้อความเตือนผู้ใช้งานโดยใช้ Msg box

Procedure การทำงานของ Visual Basic จะรวบรวมคำสั่งไว้ใน Procedure ซึ่ง Procedure อาจจะมีคำสั่งเดียว หรือชุดคำสั่งตามการใช้งาน แบ่งออกเป็น 2 แบบ คือ

- ฟังก์ชัน เป็น Procedure ที่ส่งคืนออกมาภายหลังการประมวลผล สามารถใช้เป็น Expression ในการคำนวณ
- Sub Procedure เป็น Procedure ที่ใช้ประมวลผล แต่ไม่สามารถส่งคืนค่าออกมา

โมดูล (Module) เป็นที่เก็บ Procedure ของ Visual Basic แบ่งออกเป็น 2 แบบ คือ

- Standard Module เป็นการเก็บ Procedure ที่เรียกใช้ได้จากทุกส่วนของโปรแกรม หรือเป็น Procedure ที่ใช้งานร่วมกันของทุกอ็อบเจกต์
- Class Module เป็นการเก็บ Procedure ที่ใช้เฉพาะฟอรั่มหรือรายงาน โดยอ็อบเจกต์อื่นไม่สามารถเรียกมาใช้งานได้

Event Procedure ในการสร้างและใช้งาน Sub Procedure ต่างๆ จะต้องคำนึงถึง Event ที่ใช้งาน เช่น ต้องการคลิกให้กำหนด Sub Procedure ที่คุณสมบัตินี้ On Click หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการให้คำสั่งทำงาน เมื่อมีการเปิดฟอร์ม ให้กำหนด Sub Procedure ที่คุณสมบัติ On Open เป็นต้น

Visual Basic Editor ทำหน้าที่ในการเรียกคำสั่ง Visual Basic มาแก้ไข

Locals Window ใช้ในการทดสอบคำสั่งของ Visual Basic โดยสามารถติดตามการประมวลผลตามขั้นของชุดคำสั่ง การเปิด Locals window ให้คลิก Locals window จากเมนู View (เลือกคำสั่ง View + Locals Window) ใน Visual Basic Editor

Immediate Window ใช้ในการทดสอบค่าของตัวแปร การเปิด Immediate window ให้คลิก Immediate window จากเมนู View (เลือกคำสั่ง View + Immediate window) ใน Visual Basic Editor

2.7.2 ไวยากรณ์ของ Visual Basic

ในการใช้ภาษา Visual Basic ต้องคำนึงถึงไวยากรณ์ที่ถูกต้องจึงจะใช้คำสั่งของ Visual Basic ในการทำงานได้

- ไวยากรณ์ของ Procedure

รูปแบบ

ประเภท Procedure (อากิวเมนต์) as ประเภทข้อมูล

คำสั่ง

End Sub

- ไวยากรณ์ของ Function

รูปแบบ

Function (อากิวเมนต์) as ประเภทข้อมูล

คำสั่ง

End Sub

อากิวเมนต์ของฟังก์ชันเป็นตัวแปรที่ต้องนำเข้ามาใช้ในการคำนวณ

- ไวยากรณ์กำหนดตัวแปร

รูปแบบ Dim ชื่อตัวแปร as ประเภทข้อมูล

Dim strCriteria As String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไวยากรณ์การใช้คำสั่งมาโคร

รูปแบบ Doccmd ชื่อคำสั่งมาโคร, (อาทิวเมนต์)

DoCmd.OpenForm "frmEmployee", acNormal

- ไวยากรณ์การใช้ไอ้บเจค

รูปแบบ ชื่อไอ้บเจค คุณสมบัติ = Expression

cmdExit.Enable = False

- การอ้างอิงชื่อไอ้บเจค เช่น เรียกไอ้บเจคมาใช้งานตาม Expression

รูปแบบ Object Name Property = Expression

Object Name คือ ชื่อของไอ้บเจค

Property คือ ค่าคุณสมบัติที่อ้างอิง

Expression คือ ค่ากำหนดคุณสมบัติ

- การเปลี่ยนค่าของตัว Control

รูปแบบ Form!Order subform!UnitsPrice = 50

หมายถึง ตัว Control ชื่อ [UnitPrice] ในฟอร์มชื่อ Orders Subform เท่ากับ 50

- การเปลี่ยนค่าของคุณสมบัติ

รูปแบบ Form!Order subform!UnitsPrice.Enable = False

หมายถึง ตัว Control ชื่อ [Unit Price] ในฟอร์มชื่อ Orders Subform ไม่ให้เข้าไป

แก้ไขข้อมูลในตัว Control ชื่อ UnitPrice การอ้างอิงชื่อตัว Control ในฟอร์มที่กำลัง

ใช้งานอยู่ สามารถเขียนย่อเป็น

Unit Price.Enable = False

เพราะขณะที่กำลังใช้ฟอร์ม [Order subform] Access จะรับรู้ว่ามี ไอ้บเจค

ตัวใด

บทที่ 3

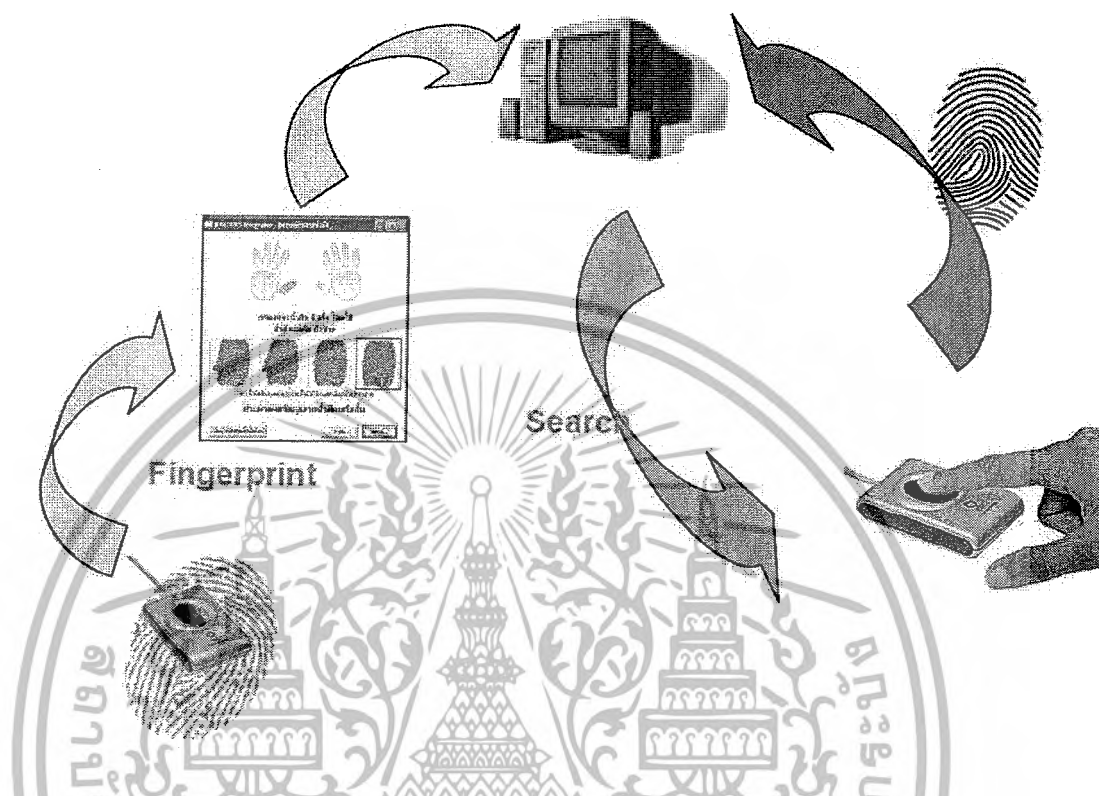
การทำงานและการออกแบบ

3.1 แนวคิดการออกแบบระบบงาน

เปลี่ยนระบบการเช็คชื่อข้อมูลบุคคลจากระบบการเช็คชื่อแบบเดิม เป็นการเช็คชื่อด้วยระบบใหม่ที่มีการนำเอาเทคโนโลยีในการตรวจสอบ และยืนยันตัวบุคคลด้วยลายนิ้วมือเข้ามาช่วยในการตรวจสอบ จะทำการตรวจสอบข้อมูลโดยการรับค่าผ่านทางเครื่องอ่านลายนิ้วมือและนำค่ารหัสลายนิ้วมือที่ได้ไปเปรียบเทียบกับค่ารหัสลายนิ้วมือที่มีอยู่ในฐานข้อมูล แล้วทำการประมวลผลหาค่ารหัสลายนิ้วมือที่ตรงกัน เพื่อให้สามารถสืบค้นข้อมูลบุคคลได้อย่างรวดเร็ว และอาศัยลักษณะเฉพาะของลายนิ้วมือของแต่ละบุคคลช่วยในการยืนยันตัวบุคคลเพื่อให้การเช็คชื่อเกิดความแม่นยำมากยิ่งขึ้น ทั้งยังสามารถลดปัญหาการปลอมแปลงบุคคลได้อีกด้วย

ในที่นี้ได้ทำการสร้างระบบการรักษาความปลอดภัย โดยการนำเอาเทคโนโลยีการตรวจสอบลายนิ้วมือเข้ามาใช้ภายในระบบเพื่อให้การตรวจสอบ และการยืนยันตัวบุคคลให้มีความถูกต้องแม่นยำมากยิ่งขึ้น

3.2 ส่วนประกอบของระบบงาน (System Diagram)



รูปที่ 3.1 รูปแบบและขั้นตอนการทำงาน

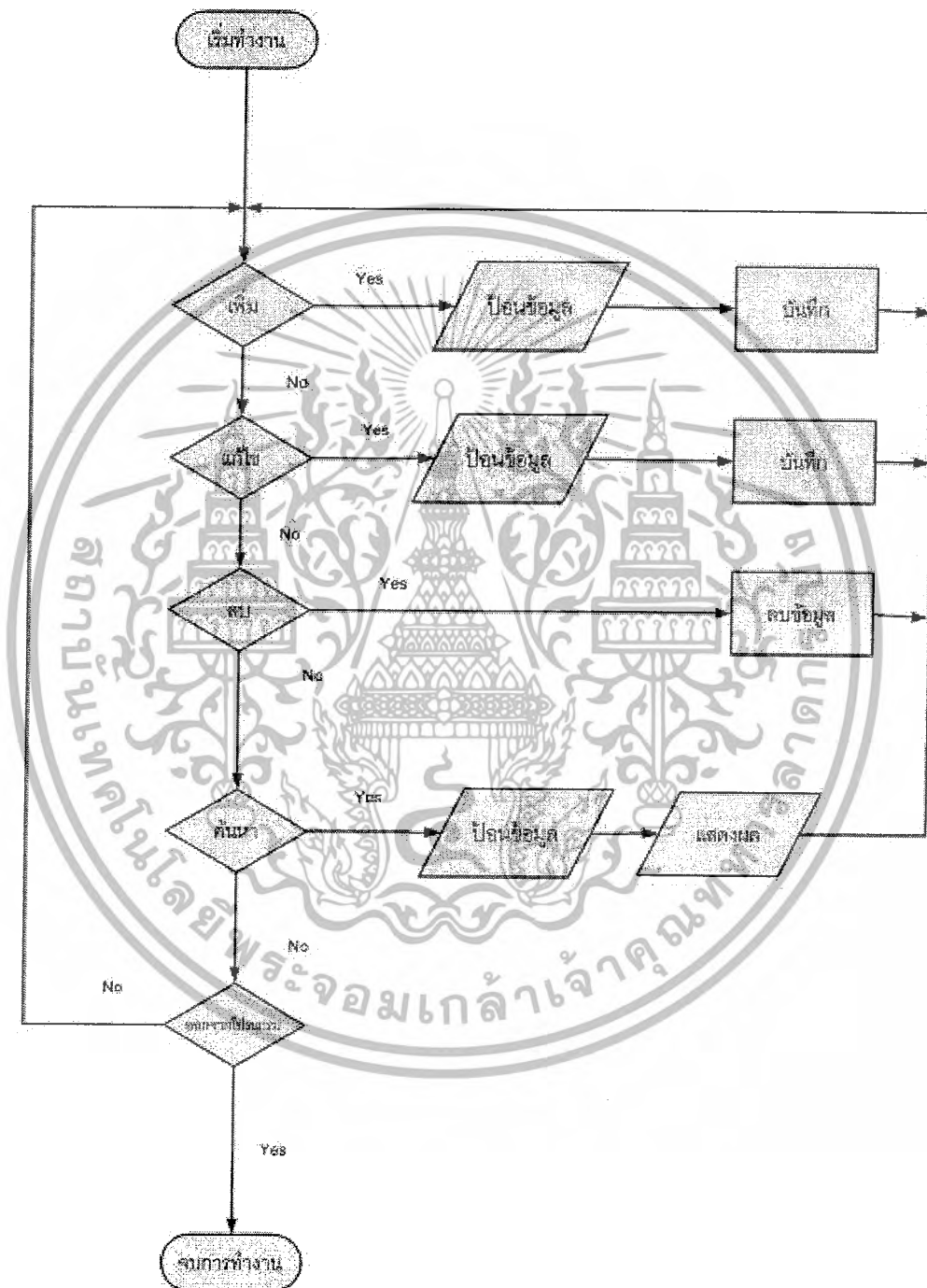
จากรูปที่ 3.1 จะแสดงให้เห็นถึงขั้นตอนการทำงานโดยประกอบด้วย การจัดเก็บลายนิ้วมือ (Enrollment) และตรวจสอบลายนิ้วมือ (Identification) โดยลูกศรสีเหลืองจะแสดงการทำงานในส่วนของการจัดเก็บลายนิ้วมือจากผู้ใช้งานบันทึกลงในฐานข้อมูลในคอมพิวเตอร์ ส่วนลูกศรสีเขียวและน้ำเงิน แสดงขั้นตอนการแตะนิ้วและประมวลผลเพื่อยืนยันลายนิ้วมือ

การจัดเก็บลายนิ้วมือ ทำเพื่อใช้เก็บข้อมูลลักษณะของนิ้วนั้นๆ แล้วใช้เปรียบเทียบในภายหลัง ซึ่งค่าที่ได้จากการเก็บลายนิ้วมือจะเป็นข้อมูลในรูปแบบ Binary มีความยาวประมาณ 400-600 ไบต์ หลังจากได้ข้อมูลลายนิ้วมือแล้วสามารถนำไปเก็บไว้ในสื่อเก็บข้อมูลชนิดใดก็ได้ เช่น hard disk ซึ่งอาจเก็บในรูปแบบของ database หรือ file ข้อมูลทั่วไป หรือเก็บลงในบัตร smart card จากนั้นเมื่อต้องการทำการเปรียบเทียบลายนิ้วมือ ต้องอ่านข้อมูลที่บันทึกไว้เพื่อส่งให้ระบบเก็บไว้ในหน่วยความจำ เมื่อมีการแตะนิ้ว ระบบจะทำการเปรียบเทียบลายนิ้วมือจากเครื่องอ่านลายนิ้วมือ กับข้อมูลที่อยู่ในหน่วยความจำเพื่อประมวลผลว่าพบหรือไม่ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ขั้นตอนการทำงาน Flow Chart

3.3.1 การออกแบบโปรแกรมส่วนของข้อมูลนักศึกษา



รูปที่ 3.2 Flowchart ข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

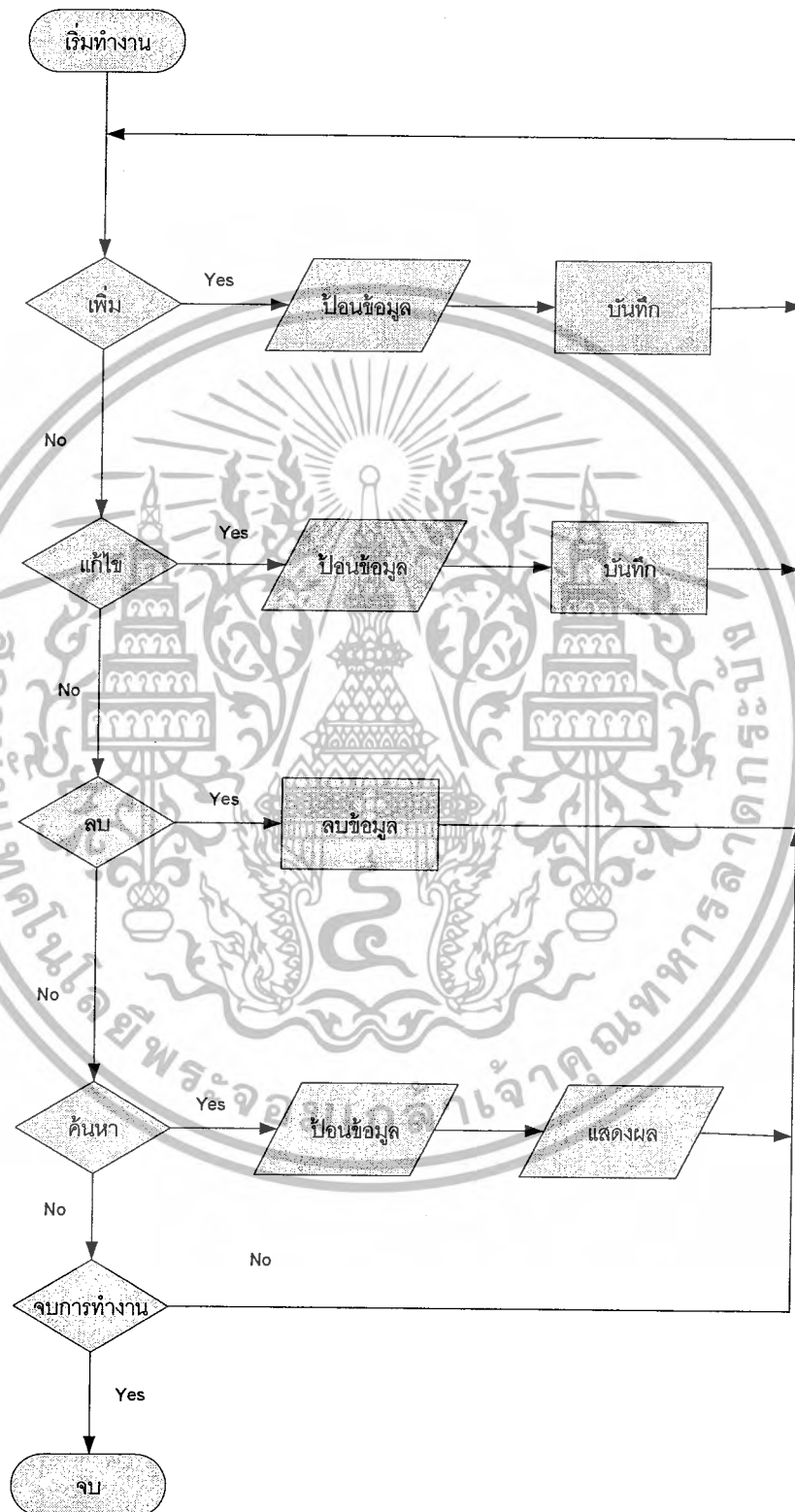
3.3.2 การออกแบบโปรแกรมส่วนรายงานการเข้าเรียน



รูปที่ 3.3 Flowchart รายงานการเข้าเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

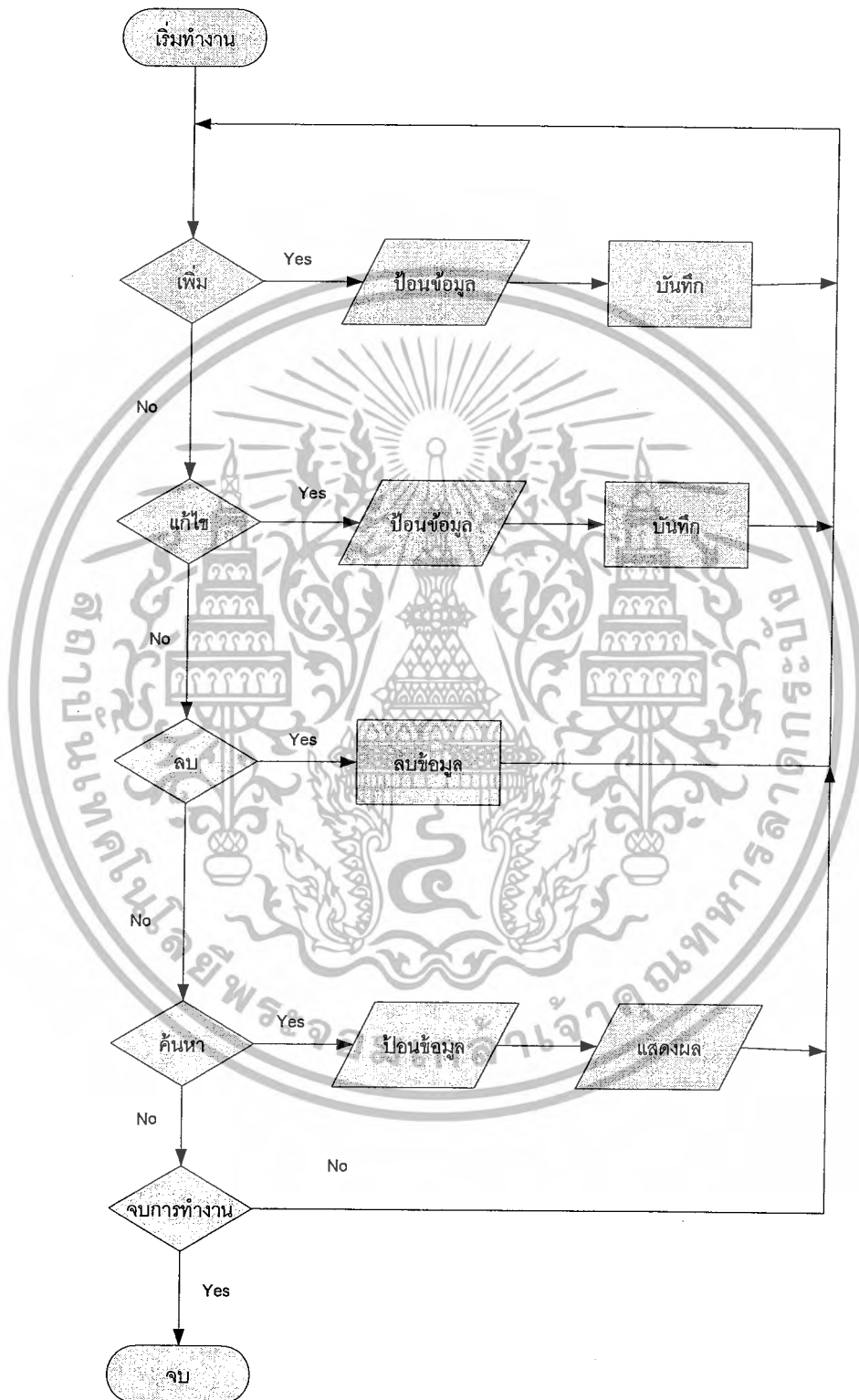
3.3.3 การออกแบบโปรแกรมส่วนของวิชา



รูปที่ 3.4 Flowchart วิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 การออกแบบโปรแกรมส่วนของข้อมูลชั้นเรียน



รูปที่ 3.5 Flowchart ข้อมูลชั้นเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 การออกแบบโปรแกรมส่วนบันทึกการเข้าเรียน



รูปที่ 3.6 Flowchart บันทึกการเข้าเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ขั้นตอนการทำงานของระบบ

1. ส่วนของการลงทะเบียน

การลงทะเบียนจะทำเพียงครั้งเดียวก่อนการเริ่มใช้งานระบบ เพื่อให้ผู้ใช้ทำการสแกนภาพถ่ายนิ้วมือ บันทึกภาพถ่าย รวมทั้งกรอกข้อมูลส่วนบุคคลต่างๆ แล้วบันทึกในขั้นตอนแรกลงในฐานข้อมูลเพื่อใช้เป็นแม่แบบ(Template)ในการตรวจสอบ

2. ส่วนของการตรวจสอบหรือการระบุตัวบุคคล

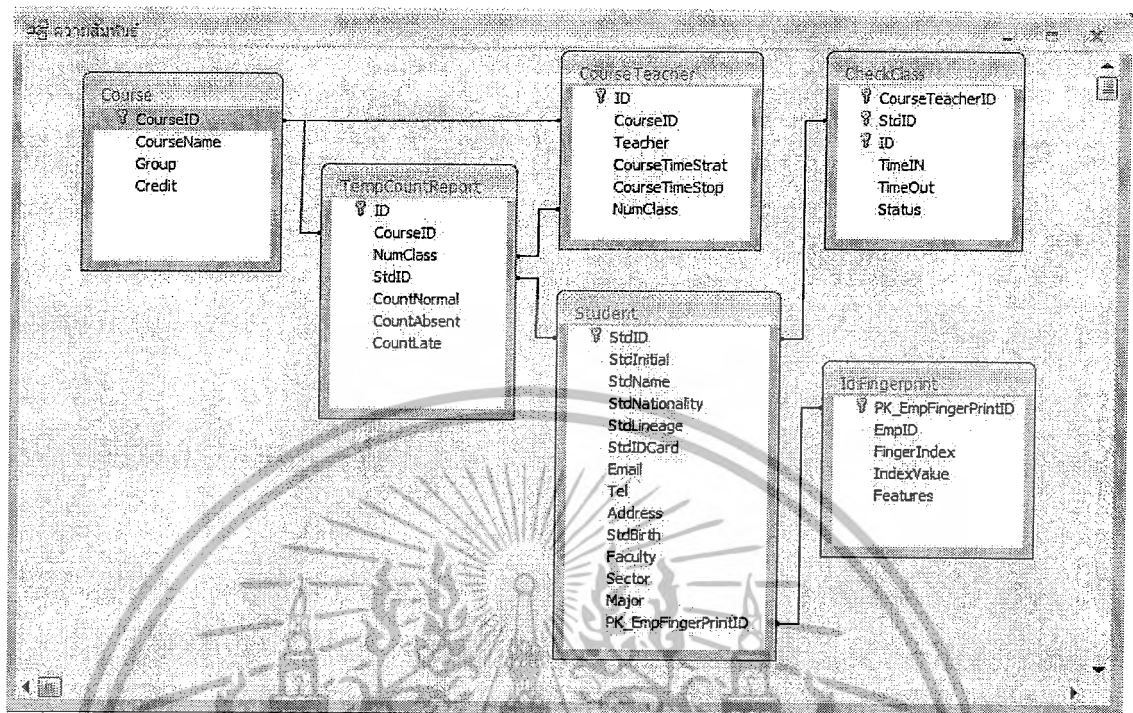
จะกระทำทุกครั้งเมื่อต้องการตรวจหาตัวบุคคล โดยการตรวจสอบหรือการระบุตัวบุคคลจะมีการเก็บตัวอย่างภาพถ่ายนิ้วมือของบุคคลที่ต้องการตรวจสอบ แล้วทำการเปรียบเทียบกับแม่แบบที่เก็บไว้ในฐานข้อมูลเพื่อตรวจสอบความเหมือนกันระหว่างภาพถ่ายนิ้วมือตัวอย่างกับแม่แบบในฐานข้อมูล

3.5 คำอธิบายขั้นตอนการดำเนินงาน

1. ทำการเชื่อมต่อเครื่องสแกนลายนิ้วมือทางพอร์ต USB
2. Install driver
3. เปิดใช้งาน
4. ทำการลงทะเบียนข้อมูลนักศึกษา
5. ทำการลงทะเบียนรายวิชา
6. ทำการลงทะเบียนเวลาเรียน
7. ทำการเลือกวิชาที่จะเช็คชื่อ
8. ทำการสแกนลายนิ้วมือเพื่อเปรียบเทียบ
9. เลือกรายงานเพื่อแสดงรายละเอียดของนักศึกษาในการเข้าเรียน

3.6 การออกแบบฐานข้อมูล

เพื่อจำลองระบบการตรวจสอบการเข้าเรียนโดยใช้ลายนิ้วมือ จึงได้ออกแบบระบบฐานข้อมูลที่ใช้ในการเก็บข้อมูลเกี่ยวกับระบบดังนี้



รูปที่ 3.7 ความสัมพันธ์ระหว่างตาราง

CourseTeacher	StdID	ID	TimeIN	TimeOut	Status
*		(สร้าง)			

รูปที่ 3.8 ตาราง CheckClass

CourseID	CourseName	Group	Credit
*			

รูปที่ 3.9 ตาราง Course

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CourseTeacher						
ID	CourseID	Teacher	CourseTime	CourseTime	NumClass	
*	(สร้าง)					

รูปที่ 3.10 ตาราง Course Teacher

IdFingerprint				
PK_EmpFingerPrintID	EmpID	FingerIndex	IndexValue	Features
*	(สร้าง)			

รูปที่ 3.11 ตาราง IdFingerprint

Student									
StdID	StdInitial	StdName	StdNational	StdLineage	StdIDCard	Email	Tel	Address	StdBirth
*									

รูปที่ 3.12 ตาราง Student

3.7 ภาษาและเครื่องมือที่ใช้ในการพัฒนา

3.7.1 ภาษาที่ใช้พัฒนา

- Visual Basic ซึ่งเป็นภาษาที่เขียนโปรแกรมได้ง่าย และสะดวกในการสร้างฟอร์มต่างๆ

3.7.2 ซอฟต์แวร์

- Visual Studio 2005
- Microsoft Access 2003
- ID Works Integrator Gold Fingerprint Recognition Software 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.3 ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ Core 2 Duo (CPU 2.33 GHz. , Hard disk 160 GB. , RAM 2 GB.)
- เครื่องอ่านลายนิ้วมือ รุ่น URU4000

เครื่องอ่านลายนิ้วมือ URU4000 แบบ USB ที่ถูกออกแบบมาสำหรับใช้งานกับ แอปพลิเคชันต่างๆ ผู้ใช้งานเพียงแต่แตะนิ้วของตนบนช่องหน้าต่างเรืองแสง เครื่องก็จะทำการจับภาพลายนิ้วมืออย่างรวดเร็วโดยอัตโนมัติแพลงวงจรควบคุมการทำงานในเครื่องจะทำการตั้งค่าการใช้งานต่างๆและเข้ารหัสข้อมูลก่อนที่จะส่งข้อมูลไปยังเครื่องคอมพิวเตอร์ทางพอร์ต USB เพื่อรับภาพลายนิ้วมือจากเครื่อง client แล้วนำภาพที่ได้ไปใช้ในการตรวจสอบต่อไป โดยคุณสมบัติของเครื่อง URU4000 มีดังนี้

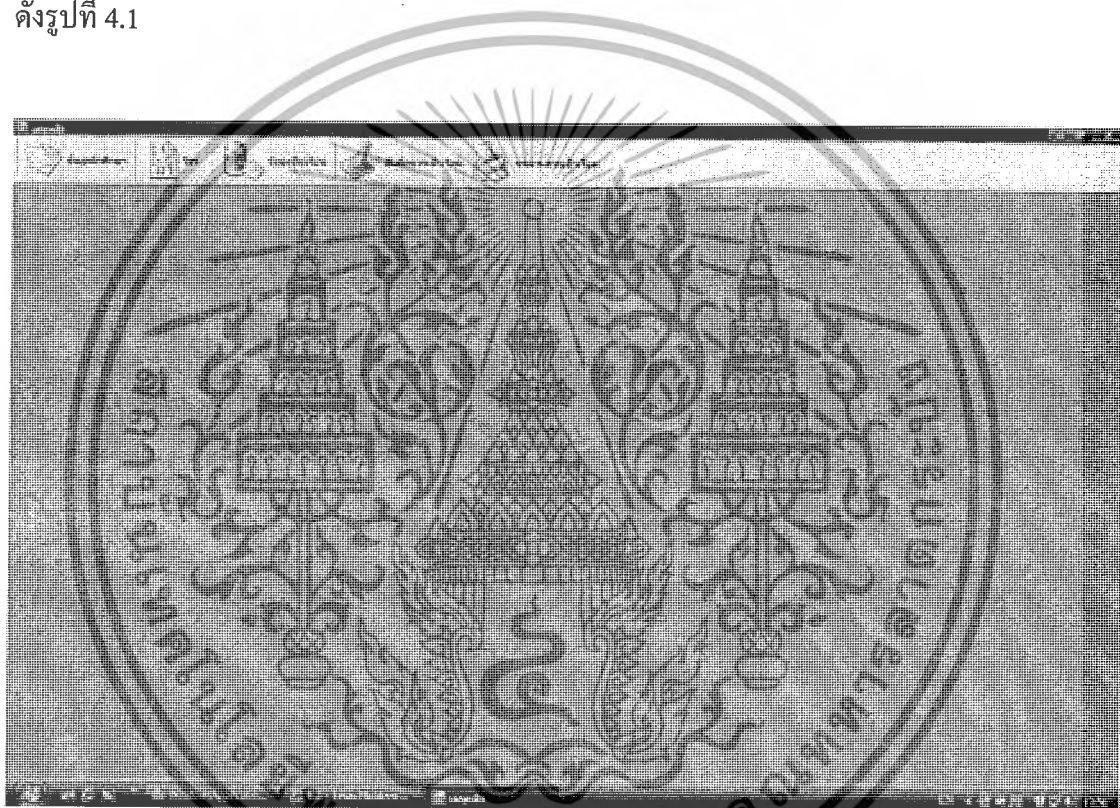
- มีขนาดกะทัดรัดมีการเข้ารหัสข้อมูล
- ไม่ยอมรับลายนิ้วมือปลอม
- สามารถหมุนนิ้วเป็นองศาต่างๆ ขณะแตะได้
- ทำงานได้ดีแม้แต่นิ้วมือที่แห้ง/ชื้น/ขรุขระได้

บทที่ 4

การใช้งาน

4.1 เริ่มต้นเข้าสู่โปรแกรมการใช้งาน

เมื่อเปิดโปรแกรมขึ้นมา เราจะเข้ามาสู่นำต่างหลักของโปรแกรมการใช้งานซึ่งมีหน้าตา
ดังรูปที่ 4.1



รูปที่ 4.1 หน้าต่างการใช้งานหลัก

จากรูปที่ 4.1 จะเห็นได้ว่าหลังจากเราเข้ามาสู่นำต่างการใช้งานหลักแล้ว จะมีปุ่มการใช้งานอีกย่อยๆทั้งหมด 5 ปุ่มหลัก ด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ปุ่มข้อมูลนักศึกษา

ปุ่มข้อมูลนักศึกษาเป็นปุ่มสำหรับเข้าสู่หน้าต่างการจัดการข้อมูลของนักศึกษา โดยมีรายละเอียดดังนี้

- ปุ่ม Add new สำหรับเพิ่มรายชื่อนักศึกษาใหม่
- ปุ่ม Delete สำหรับลบรายชื่อนักศึกษาที่ต้องการลบ
- ปุ่ม Save Data สำหรับบันทึกข้อมูลนักศึกษา
- ช่องสำหรับค้นหานักศึกษา โดยใช้รหัสนักศึกษาหรือส่วนหนึ่งของรหัสนักศึกษา
- ช่องสำหรับค้นหานักศึกษา โดยใช้ชื่อหรือส่วนหนึ่งของชื่อนักศึกษา
- ปุ่มบันทึกถาวรสำหรับบันทึกถาวรข้อมูลของนักศึกษา

รูปที่ 4.2 หน้าต่างข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลนักศึกษา

รหัสนักศึกษา: 48010031 ลำดับรายชื่อ: นาย ชื่อ - สกุล: อภิวัฒน์ วัฒนกุล

สัญชาติ: ไทย ชื่อชาติ: ไทย

หมายเลขบัตรประชาชน: 1309900301949 E-mail: makgl_lmc@hotmail.com เบอร์โทรฯ: 0859212722

ที่อยู่: 697 หมู่บ้านเขตรักษาพันธุ์สัตว์ป่าห้วยขาแข้ง อุทยานแห่งชาติเขาสก 10310

เกิด: 12 สิงหาคม 2530 อายุ: 22 ปี

คณะ: วิศวกรรมศาสตร์ ภาควิชา: วิศวกรรมศาสตร์ สาขา: วิศวกรรมศาสตร์

รหัสนักศึกษา	ลำดับรายชื่อ	ชื่อ - สกุล	หมายเลขบัตรประชาชน	คณะ	ภาควิชา	สาขา
48010031	นาย	อภิวัฒน์ วัฒนกุล	1101401002738	วิศวกรรมศาสตร์	วิศวกรรมศาสตร์	วิศวกรรมศาสตร์
48010097	นาง	ณิชาภัฏ วัฒนกุล	1933000231285	วิศวกรรมศาสตร์	วิศวกรรมศาสตร์	วิศวกรรมศาสตร์
48010438	นางสาว	ณิชาภัฏ วัฒนกุล				

รูปที่ 4.3 ตัวอย่างการค้นหารายชื่อนักศึกษา

IDWORKS Integrator - ลงทะเบียนลายนิ้วมือ

กรุณาแตะนิ้วมือ 4 ครั้ง โดยใช้
นิ้วหัวแม่มือ มือซ้าย

* ลายนิ้วมือสีที่แสดงหมายถึงลายนิ้วมือที่มีรายละเอียดต่ำกว่าค่าปกติ

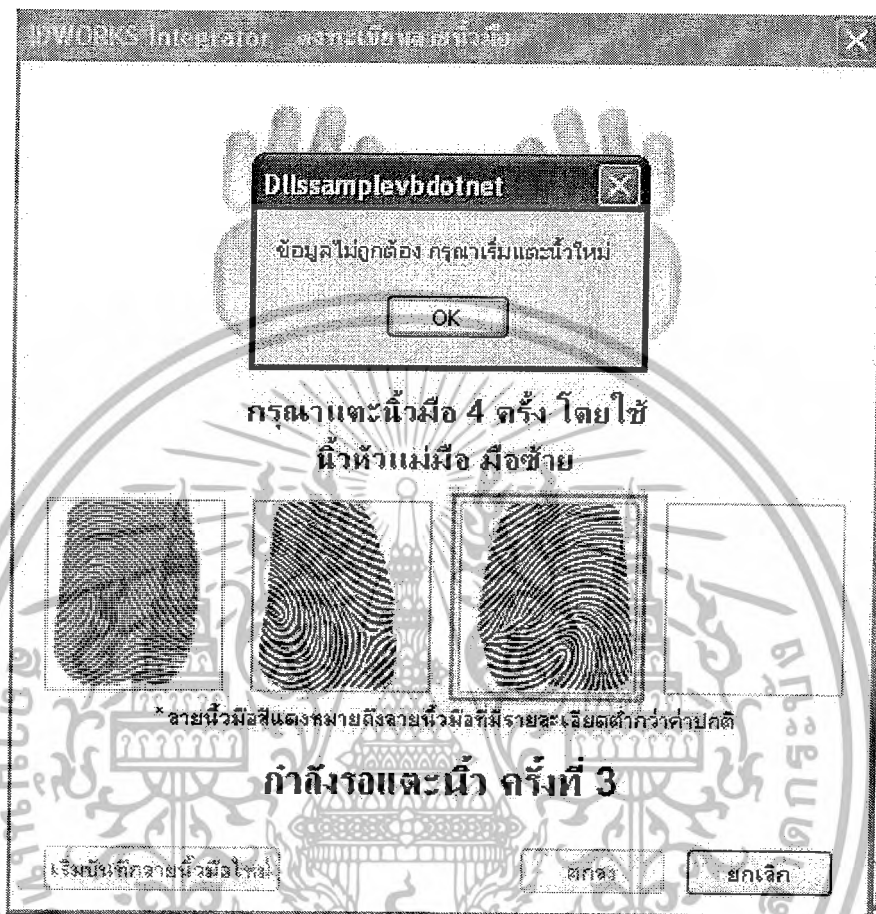
กำลังรอแตะนิ้ว ครั้งที่ 1

เริ่มบันทึกลายนิ้วมือของคุณ

ยกเลิก

รูปที่ 4.4 หน้าจอการบันทึกลายนิ้วมือก่อนใส่ลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 หน้าจอการใส่ลายนิ้วมือที่ฝึกพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



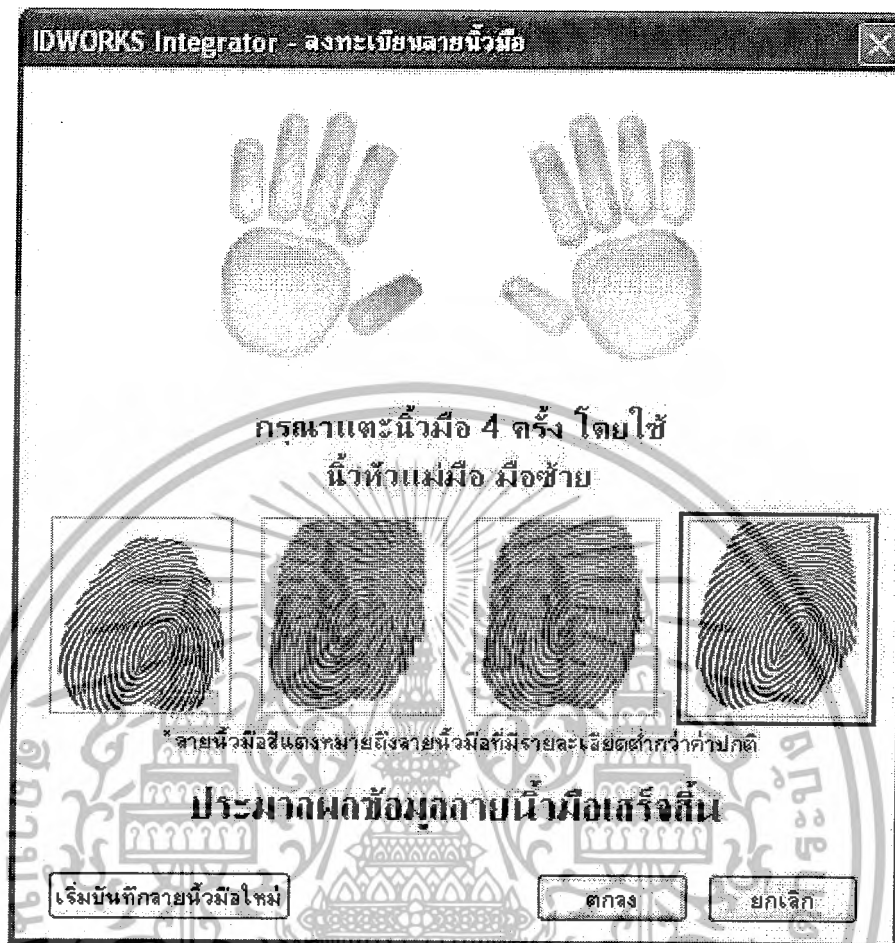
รูปที่ 4.6 หน้าจอแสดงลายนิ้วมือที่คุณภาพต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

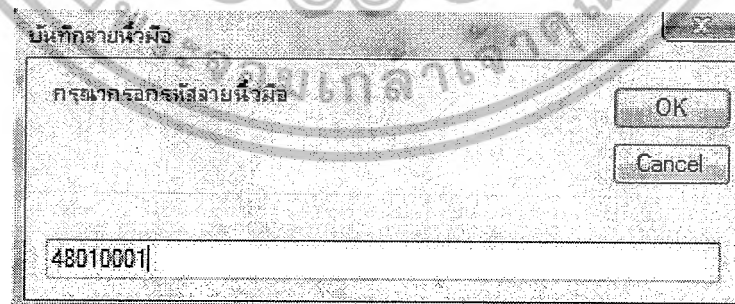


รูปที่ 4.7 หน้าจอแสดงลายนิ้วมือที่คุณภาพต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

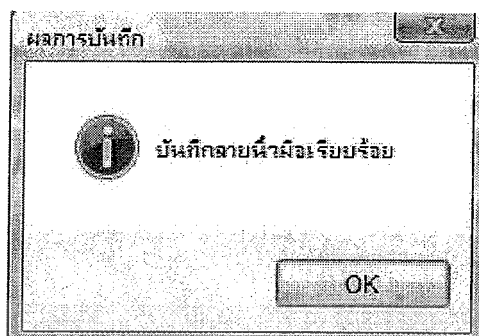


รูปที่ 4.8 หน้าจอการบันทึกลายนิ้วมือเมื่อใส่ลายนิ้วมือเสร็จสิ้น



รูปที่ 4.9 หน้าจอการใส่รหัสลายนิ้วมือหลังจากเก็บข้อมูลลายนิ้วมือแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 หน้าต่างแสดงผลการบันทึกภายในมีข้อผิดพลาด

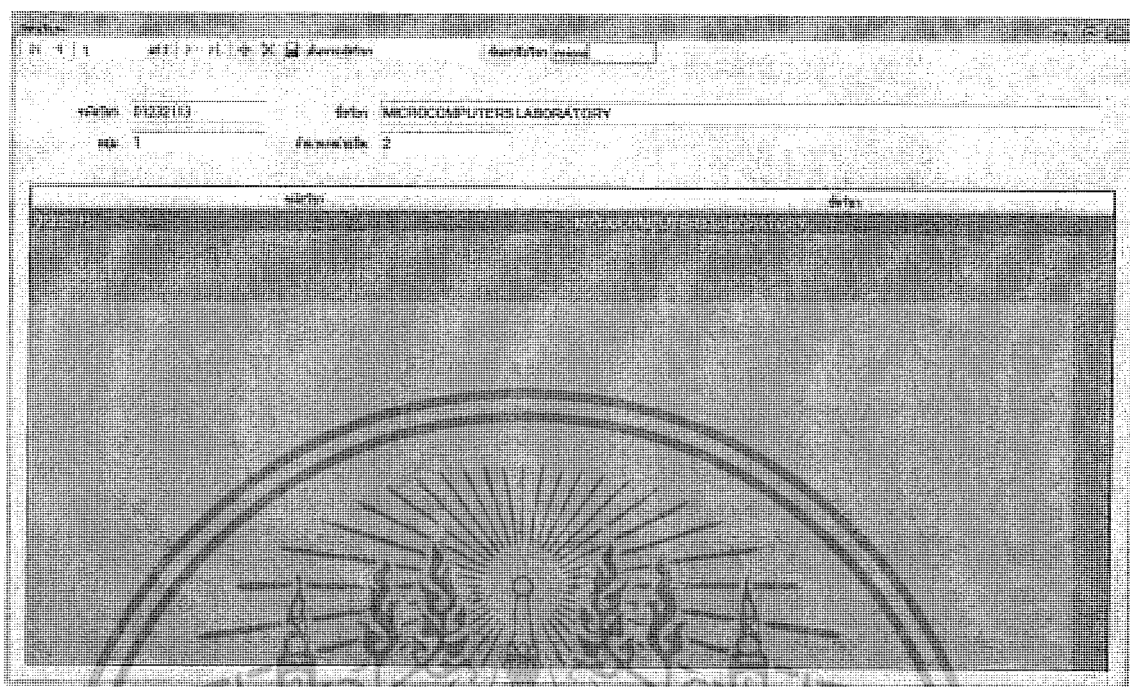
4.3 ปุ่มวิชา

ปุ่มวิชาเป็นปุ่มสำหรับเข้าสู่หน้าต่างวิชาเรียน ไว้สำหรับจัดการข้อมูลรายวิชา สามารถเพิ่มหรือลบหรือค้นหารายวิชาที่ต้องการได้ในหน้าต่างการใช้งานนี้



รูปที่ 4.11 หน้าต่างวิชาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ตัวอย่างการค้นหารายชื่อวิชา

4.4 ปุ่มข้อมูลชั้นเรียน

ปุ่มข้อมูลชั้นเรียนเป็นปุ่มสำหรับเข้าสู่หน้าต่างการจัดการข้อมูลของห้องเรียน โดยสามารถกำหนดได้ว่ารายวิชาในแต่ละวิชา มีอาจารย์ผู้สอนคือใครบ้าง สามารถกำหนดกลุ่มเรียน, เวลาเข้าเรียน, เวลาเลิกเรียน และจำนวนคาบเรียนที่ต้องเข้าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกอาจารย์ประจำวิชา

13 of 13

รหัสวิชา: 01233112 ชื่อวิชา: INFORMATION ENGINEERING LABORATORY

อาจารย์: อ.สรพงษ์ วัชรรัตน์แหกกล

เวลาเข้าเรียน: 9:00:00 เวลาเลิกเรียน: 12:00:00 จำนวนคาบเรียน: 1

รหัสวิชา	อาจารย์ผู้สอน	เวลาเข้าเรียน	เวลาเลิกเรียน	จำนวนคาบเรียน
01232114	อ.สรพงษ์ วัชรรัตน์แหกกล	9:00:00	12:00:00	1
01232114	จศ.ดร.ชวลิต เมฆสงคงประเสริฐ	9:00:00	12:00:00	1
01232114	อ.ศอานพร นพาม่วงศ์	9:00:00	12:00:00	1
01232114	ผศ.ดร.อรรถสิทธิ์ ห่อล้าสกุล	9:00:00	12:00:00	1
01232114	ดร.พิทักษ์ สารมาวัน	9:00:00	12:00:00	1
01232114	ผศ.สุธิดา พิธีธำรงเกษ์	9:00:00	12:00:00	1
01232114	อ.เนัจจาณี สัตยาภิรักษ์	9:00:00	12:00:00	1
01232114	ผศ.ดร.สมเกียรติ อุดมหาระพาสกุล	9:00:00	12:00:00	1
01232114	ผศ.มนต์ชัย แหม่นชัย	9:00:00	12:00:00	1
01232114	อ.วันวิสา ธัชวาทย์	9:00:00	12:00:00	1
01232113	อ.สรพงษ์ วัชรรัตน์แหกกล	9:00:00	12:00:00	1
01233111	อ.สรพงษ์ วัชรรัตน์แหกกล	9:00:00	12:00:00	1
01233112	อ.สรพงษ์ วัชรรัตน์แหกกล	9:00:00	12:00:00	1

รูปที่ 4.13 หน้าต่างข้อมูลชั้นเรียน

เลือกอาจารย์ประจำวิชา

14 of 14

รหัสวิชา: 01233112 ชื่อวิชา: INFORMATION ENGINEERING LABORATORY

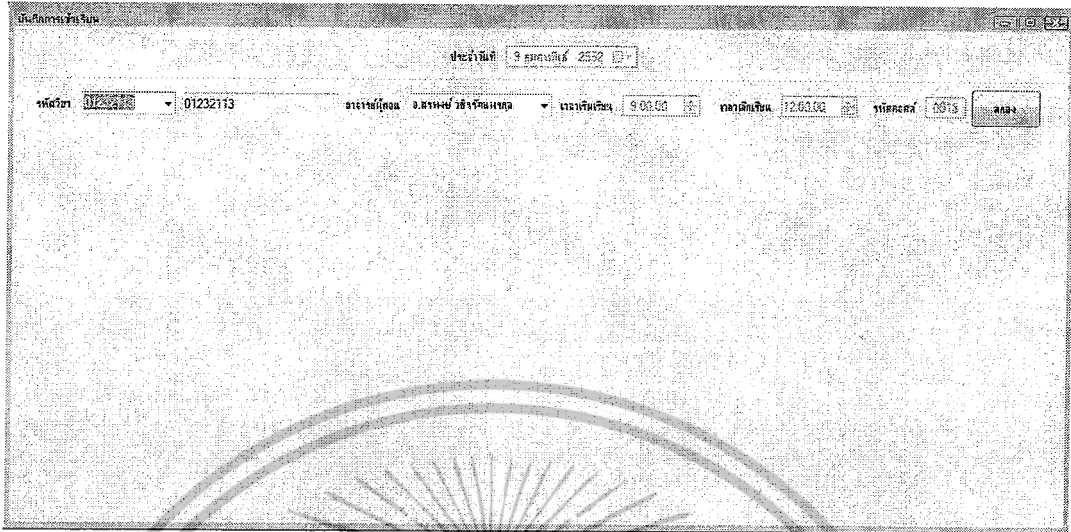
อาจารย์: อ.เนัจจาณี สัตยาภิรักษ์

เวลาเข้าเรียน: 9:00:00 เวลาเลิกเรียน: 12:00:00 จำนวนคาบเรียน: 1

รหัสวิชา	อาจารย์ผู้สอน	เวลาเข้าเรียน	เวลาเลิกเรียน	จำนวนคาบเรียน
01232114	อ.สรพงษ์ วัชรรัตน์แหกกล	9:00:00	12:00:00	1
01232114	จศ.ดร.ชวลิต เมฆสงคงประเสริฐ	9:00:00	12:00:00	1
01232114	อ.ศอานพร นพาม่วงศ์	9:00:00	12:00:00	1
01232114	ผศ.ดร.อรรถสิทธิ์ ห่อล้าสกุล	9:00:00	12:00:00	1
01232114	ดร.พิทักษ์ สารมาวัน	9:00:00	12:00:00	1
01232114	ผศ.สุธิดา พิธีธำรงเกษ์	9:00:00	12:00:00	1
01232114	อ.เนัจจาณี สัตยาภิรักษ์	9:00:00	12:00:00	1
01232114	ผศ.ดร.สมเกียรติ อุดมหาระพาสกุล	9:00:00	12:00:00	1
01232114	ผศ.มนต์ชัย แหม่นชัย	9:00:00	12:00:00	1
01232114	อ.วันวิสา ธัชวาทย์	9:00:00	12:00:00	1
01232113	อ.สรพงษ์ วัชรรัตน์แหกกล	9:00:00	12:00:00	1
01233111	อ.สรพงษ์ วัชรรัตน์แหกกล	9:00:00	12:00:00	1
01233112	อ.สรพงษ์ วัชรรัตน์แหกกล	9:00:00	12:00:00	1

รูปที่ 4.14 ตัวอย่างการเพิ่มชื่ออาจารย์ในแต่ละวิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 หน้าต่างการบันทึกการเข้าเรียน

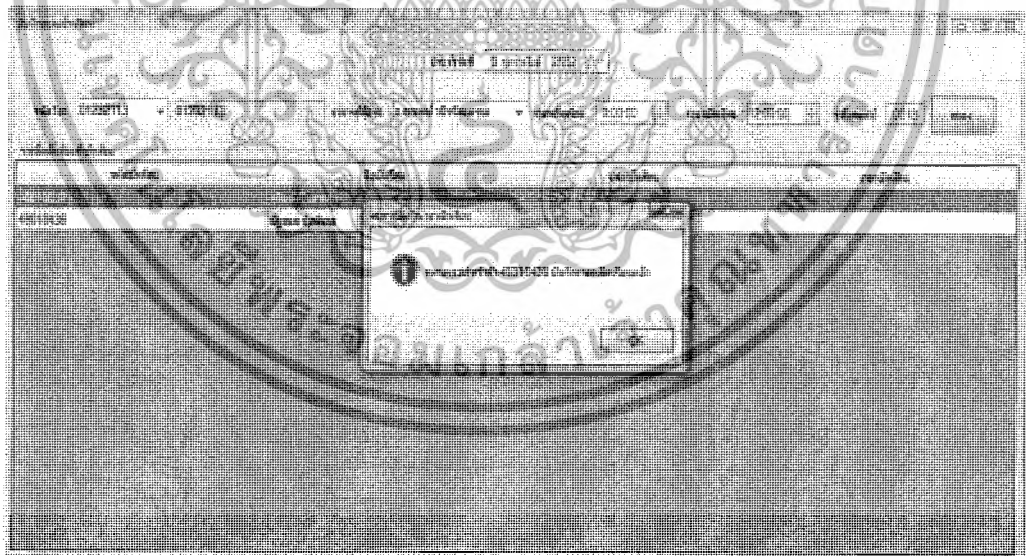


รูปที่ 4.17 ตัวอย่างการบันทึกลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 หน้าต่างแสดงผลเวลาเข้าเรียน



รูปที่ 4.19 หน้าต่างแสดงผลเวลาเลิกเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสประจำตัว	ชื่อ-นามสกุล	จำนวนชั่วโมง	เกรดเฉลี่ย
48010438	วิชากร อภิชาติ	9.000000	3.250000

รูปที่ 4.20 หน้าต่างแสดงรายชื่อนักศึกษาที่บันทึกเวลาเรียนในรายวิชานั้นๆ

4.6 ปุ่มรายงานการเข้าเรียน

ปุ่มรายงานการเข้าเรียนเป็นปุ่มสำหรับแสดงผลการเข้าเรียนของนักศึกษาทั้งหมด ว่ามาทั้งหมดกี่ครั้ง สายทั้งหมดกี่ครั้ง และขาดเรียนทั้งหมดกี่ครั้งในแต่ละรายวิชา

รายงานการเข้าเรียน

ภาควิชา 01232113 - MICROCOMPUTERS LABORATORY

รายวิชา: 01232113 ชื่อวิชา: MICROCOMPUTERS LABORATORY

จำนวนที่ห้องเข้าเรียน: 10 ครั้ง

รหัสประจำตัว	ตำแหน่งชื่อ	ชื่อ - สกุล	ปกติ (วัน)	ขาด (วัน)	มาสาย (วัน)
48010031	นาย	ศรินทร์ชานุกูล	10	0	0
48010438	นางสาว	นฤชนก ปุ่มพิมาย	9	0	1

รูปที่ 4.21 หน้าต่างรายงานการเข้าเรียนของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และบทสรุป

5.1 สรุปผลการดำเนินปริญญานิพนธ์

การสืบค้นฐานข้อมูลบุคคลโดยใช้ลายนิ้วมือ สามารถทำการสืบค้นข้อมูลบุคคลและยืนยันตัวบุคคลได้และสำเร็จโดยการรับค่าภาพลายนิ้วมือผ่านทางเครื่องอ่านลายนิ้วมือ และสามารถแสดงผลการสืบค้นได้ทันที ช่วยให้การสืบค้นเป็นไปได้อย่างถูกต้อง แม่นยำทำให้สามารถทำการยืนยันตัวบุคคลเพื่อตรวจสอบการเข้าเรียนได้อย่างถูกต้อง

5.2 ข้อจำกัดของระบบ

1. อ่านภาพลายนิ้วมือ 3 ครั้ง แล้วนำมาหาค่าเฉลี่ย แล้วจึงนำภาพลายนิ้วมือนั้นเข้าสู่กระบวนการรู้จำ แต่หากภาพลายนิ้วมือนั้นไม่สมบูรณ์ จะต้องทำการอ่านภาพลายนิ้วมือครั้งนั้นใหม่
2. ขณะที่บันทึกจะต้องไม่มีสิ่งรบกวน เช่น ความชื้น น้ำหนักการกดนิ้วมือขณะบันทึกภาพเพื่อให้ได้ภาพที่ชัดเจน
3. ระยะเวลาในการวางนิ้วมือบนเครื่องอ่านลายนิ้วมือต้องวางเกิน 2 วินาทีขึ้นไปเพื่อให้เครื่องมีเวลาอ่านค่าและบันทึกค่าได้
4. ทิศทางการวางนิ้วมือที่เบี่ยงเบนไป ก็อาจจะทำให้การสืบค้นข้อมูลบุคคลผิดพลาดได้ เนื่องจากภาพลายนิ้วมือมีการเปลี่ยนแปลงไป
5. ลายนิ้วมือที่มีรอยหรือขาดแผลจะไม่สามารถอ่านได้หรือถ้าอ่านได้จะมีค่าลายนิ้วมือต่ำ ซึ่งจะมีผลในการจัดเก็บข้อมูลทำให้เกิดค่าความผิดพลาดได้
6. ระบบกระบวนการทำงานของเครื่องสแกนลายนิ้วมือมีข้อจำกัดในการทำงานเนื่องจากความจำในเครื่องสแกนลายนิ้วมือไม่พอ
7. การตอบสนองของเครื่องสแกนลายนิ้วมือกับคอมพิวเตอร์นั้นตอบสนองช้าบางครั้งทำให้เกิดการส่งข้อมูลผิดพลาดหรือกระบวนการทำงานช้า
8. ในบางกรณีเมื่อทำการใช้งานเครื่องสแกนลายนิ้วมือนานเกินไปทำให้เกิดเครื่องสแกนลายนิ้วมือร้อนส่งผลทำให้เครื่องสแกนลายนิ้วมือส่งและรับข้อมูลผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางในการพัฒนาต่อ

ปฏิญญาพนันท์นี้สามารถนำไปพัฒนาต่อได้อย่างหลากหลาย เนื่องจากได้ทำการพัฒนาซอฟต์แวร์ในการทำงานรับส่งค่ากับเครื่องสแกนลายนิ้วมือและระบบฐานข้อมูล ซึ่งจะเป็นประโยชน์ในอนาคตโดยผู้พัฒนาต่อโดยการนำเทคโนโลยีอุปกรณ์ต่าง มาต่อเพิ่มเพื่อเพิ่มความสามารถในการทำงานต่อได้อย่างเช่นนำอุปกรณ์เกี่ยวกับ Network มาเชื่อมจะสามารถนำมาใช้ได้หลากหลายวิธีการและทำให้เกิดการสะดวกต่อการใช้งานด้วย ในปฏิญญาพนันท์นี้ นำ เครื่องสแกนลายนิ้วมือมาพัฒนาในเรื่องการเก็บฐานข้อมูลและแสดงผลโดยหน้าจอคอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. รัชชัย จำลอง.2549.มือใหม่ Access 2003 ใช้งานอย่างมือโปร . กรุงเทพฯ : สำนักพิมพ์ ซีเอ็ดดูเคชั่น
2. ศุภชัย สมพานิช , 2550. พัฒนาระบบฐานข้อมูลด้วย VB 2005 & VC# 2005 ฉบับมืออาชีพ . พิมพ์ครั้งที่ 1 : นนทบุรี : บริษัท ไอคิซี อินโฟ คิสทริบิวเตอร์ เซ็นเตอร์ จำกัด

แหล่งค้นคว้าทางอินเทอร์เน็ต

1. <http://www.com-system.th.gs/web-c/om-system/fingerprint.html>
2. http://www.itworksolution.com/idwork_integrator/index.html
3. http://www.itworksolution.com/support/TimeWORK_Download.html
4. <http://www.successmedia.com>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน

ก.1 คุณสมบัติของระบบ และคุณสมบัติของเครื่อง

- ใช้ในการตรวจสอบ และยืนยันตัวบุคคลด้วยลายนิ้วมือได้
- สามารถจัดเก็บ เพิ่ม และลบข้อมูลต่างๆเกี่ยวกับ นักศึกษา , วิชาเรียน , อาจารย์ได้
- ตรวจสอบเวลาเข้าเวลาออกของนักศึกษาได้จริง
- รหัสรายวิชาไม่สามารถใช้เลข 0 นำหน้าได้เนื่องจากเป็น Integer
- ความต้องการของระบบ ได้ทดลองใช้กับ

Hardware

- CPU 2.33 GHz.
- Hard disk 160 GB.
- RAM 2 GB.
- Port USB 5V
- เครื่องอ่านลายนิ้วมือ รุ่น URU4000

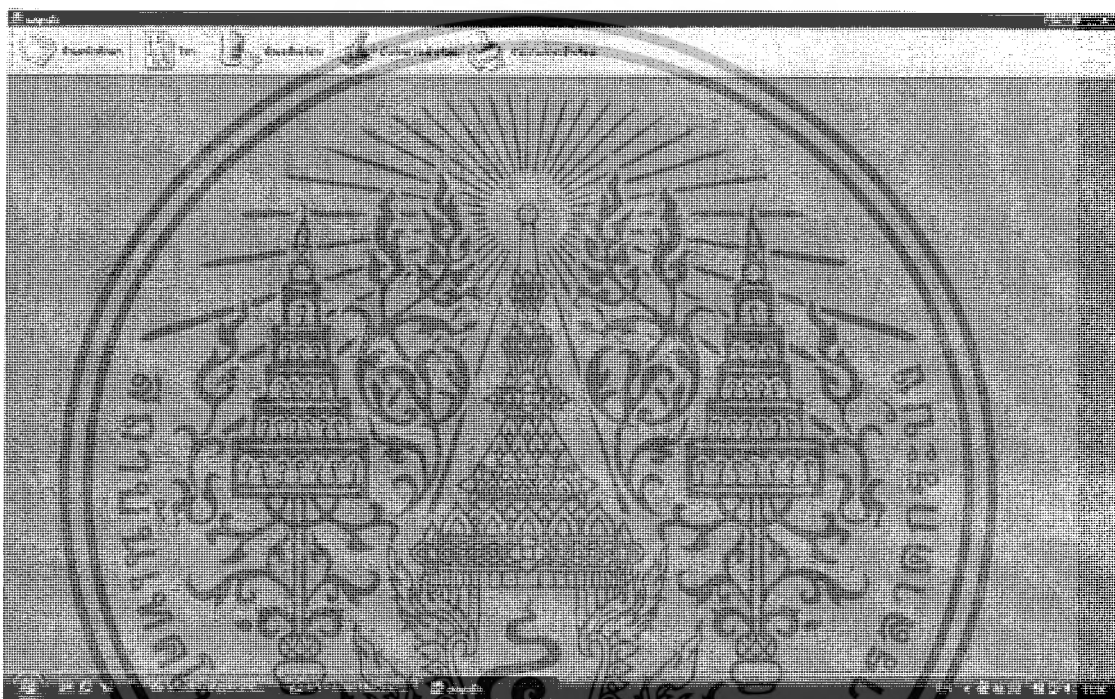
Software

- Microsoft SQL Server 2005
- Microsoft Visual C#
- Component ในการสืบค้นลายนิ้วมือ Biokey.OCX

ก.2 คู่มือการใช้งาน

1. เริ่มต้นเข้าสู่โปรแกรมการใช้งาน

เมื่อเปิดโปรแกรมขึ้นมา เราจะเข้ามาสู่หน้าต่างหลักของโปรแกรมการใช้งานซึ่งมีหน้าตา ดังรูปที่ ก.1



รูปที่ ก.1 หน้าต่างการใช้งานหลัก

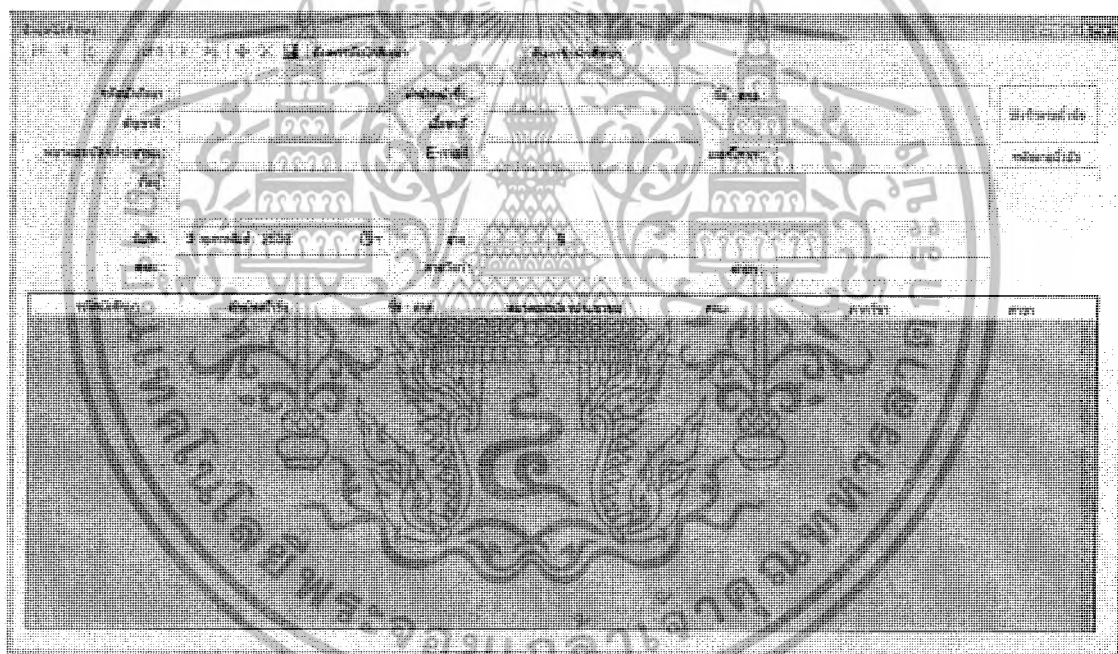
จากรูปที่ ก.1 จะเห็นได้ว่าหลังจากเราเข้ามาสู่หน้าต่างการใช้งานหลักแล้ว จะมีปุ่มการใช้งานอื่นๆ ทั้งหมด 5 ปุ่มหลัก ด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2, ปุ่มข้อมูลนักศึกษา

ปุ่มข้อมูลนักศึกษาเป็นปุ่มสำหรับเข้าสู่หน้าต่างการจัดการข้อมูลของนักศึกษา โดยมีรายละเอียดดังนี้

- ปุ่ม Add new สำหรับเพิ่มรายชื่อนักศึกษาใหม่
- ปุ่ม Delete สำหรับลบรายชื่อนักศึกษาที่ต้องการลบ
- ปุ่ม Save Data สำหรับบันทึกข้อมูลนักศึกษา
- ช่องสำหรับค้นหานักศึกษา โดยใช้รหัสนักศึกษาหรือส่วนหนึ่งของรหัสนักศึกษา
- ช่องสำหรับค้นหานักศึกษา โดยใช้ชื่อหรือส่วนหนึ่งของชื่อนักศึกษา
- ปุ่มบันทึกถายนิ้วมือ สำหรับบันทึกถายนิ้วมือของนักศึกษา



รูปที่ ก.2 หน้าต่างข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.3 หน้าต่างการบันทึกลายนิ้วมือ



รูปที่ ก.4 หน้าต่างใส่รหัสลายนิ้วมือหลังจากเก็บข้อมูลลายนิ้วมือแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ปุ่มวิชา

ปุ่มวิชาเป็นปุ่มสำหรับเข้าสู่หน้าต่างวิชาเรียน ไว้สำหรับจัดการข้อมูลรายวิชา สามารถเพิ่มหรือลบหรือค้นหารายวิชาที่ต้องการได้ในหน้าต่างการใช้งานนี้



รูปที่ ก.5 หน้าต่างวิชาเรียน

4. ปุ่มข้อมูลชั้นเรียน

ปุ่มข้อมูลชั้นเรียนเป็นปุ่มสำหรับเข้าสู่หน้าต่างการจัดการข้อมูลของห้องเรียน โดยสามารถกำหนดได้ว่ารายวิชาในแต่ละวิชา มีอาจารย์ผู้สอนคือใครบ้าง สามารถกำหนดกลุ่มเรียน, เวลาเรียน, เวลาเลิกเรียน และจำนวนคาบเรียนที่ต้องเข้าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกอาจารย์ประจำวิชา

14 4 13 of 13

รหัสวิชา 01233112 ชื่อวิชา INFORMATION ENGINEERING LABORATORY

อาจารย์ อ.ศรพงษ์ วิชาจิตแหวกล

เวลาเข้าเรียน 9:00:00 เวลาเลิกเรียน 12:00:00 จำนวนคาบเรียน 1

รหัสวิชา	อาจารย์ผู้สอน	เวลาเข้าเรียน	เวลาเลิกเรียน	จำนวนคาบเรียน
01232114	อ.ศรพงษ์ วิชาจิตแหวกล	9:00:00	12:00:00	1
01232114	รศ.ดร.ชาลิต บุญอาจประเสริฐ	9:00:00	12:00:00	1
01232114	อ.สถาพร พรหมวงษ์	9:00:00	12:00:00	1
01232114	ผศ.ดร.จรรยาสิทธิ์ ห่อคำสกุล	9:00:00	12:00:00	1
01232114	ดร.พิทักษ์ อรรจนาวิน	9:00:00	12:00:00	1
01232114	ผศ.ศุฉิรา พันธุธาดาเกษย์	9:00:00	12:00:00	1
01232114	อ.เนจจารีย์ สัตยาภิรมย์	9:00:00	12:00:00	1
01232114	ผศ.ดร.สัมพันธ์ อุดมพรหมมากุล	9:00:00	12:00:00	1
01232114	ผศ.มณฑิลา มณฑิลา	9:00:00	12:00:00	1
01232114	อ.วันวิสา อึ้งวงษ์	9:00:00	12:00:00	1
01232113	อ.ศรพงษ์ วิชาจิตแหวกล	9:00:00	12:00:00	1
01233111	อ.ศรพงษ์ วิชาจิตแหวกล	9:00:00	12:00:00	1
01233112	อ.ศรพงษ์ วิชาจิตแหวกล	9:00:00	12:00:00	1

รูปที่ ก.6 หน้าต่างข้อมูลชั้นเรียน

เลือกอาจารย์ประจำวิชา

14 4 14 of 14

รหัสวิชา 01233112 ชื่อวิชา INFORMATION ENGINEERING LABORATORY

อาจารย์ อ.เนจจารีย์ สัตยาภิรมย์

เวลาเข้าเรียน 9:00:00 เวลาเลิกเรียน 12:00:00 จำนวนคาบเรียน 1

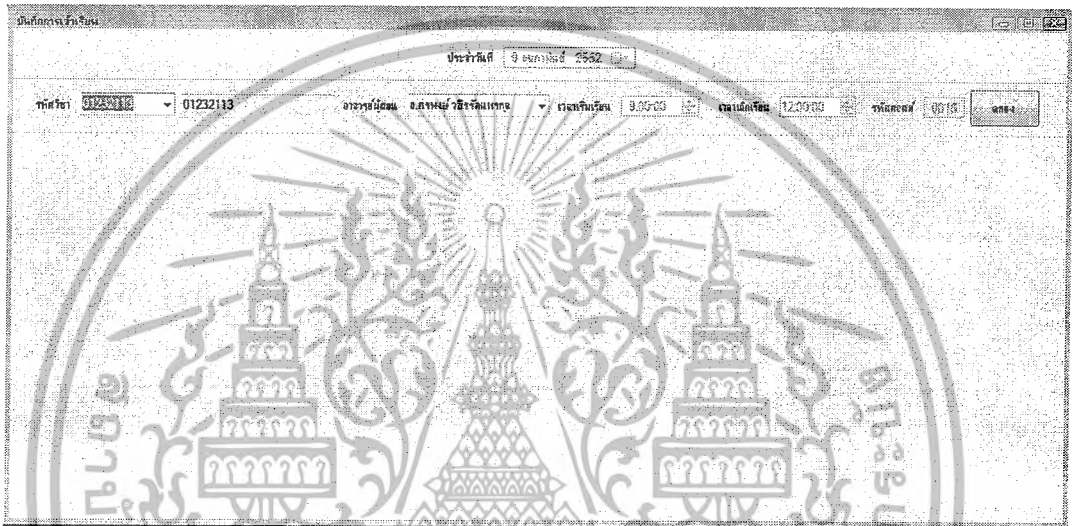
รหัสวิชา	อาจารย์ผู้สอน	เวลาเข้าเรียน	เวลาเลิกเรียน	จำนวนคาบเรียน
01232114	อ.ศรพงษ์ วิชาจิตแหวกล	9:00:00	12:00:00	1
01232114	รศ.ดร.ชาลิต บุญอาจประเสริฐ	9:00:00	12:00:00	1
01232114	อ.สถาพร พรหมวงษ์	9:00:00	12:00:00	1
01232114	ผศ.ดร.จรรยาสิทธิ์ ห่อคำสกุล	9:00:00	12:00:00	1
01232114	ดร.พิทักษ์ อรรจนาวิน	9:00:00	12:00:00	1
01232114	ผศ.ศุฉิรา พันธุธาดาเกษย์	9:00:00	12:00:00	1
01232114	อ.เนจจารีย์ สัตยาภิรมย์	9:00:00	12:00:00	1
01232114	ผศ.ดร.สัมพันธ์ อุดมพรหมมากุล	9:00:00	12:00:00	1
01232114	ผศ.มณฑิลา มณฑิลา	9:00:00	12:00:00	1
01232114	อ.วันวิสา อึ้งวงษ์	9:00:00	12:00:00	1
01232113	อ.ศรพงษ์ วิชาจิตแหวกล	9:00:00	12:00:00	1
01233111	อ.ศรพงษ์ วิชาจิตแหวกล	9:00:00	12:00:00	1
01233112	อ.ศรพงษ์ วิชาจิตแหวกล	9:00:00	12:00:00	1
01233112	อ.เนจจารีย์ สัตยาภิรมย์	9:00:00	12:00:00	1

รูปที่ ก.7 หน้าต่างแสดงการเพิ่มชื่ออาจารย์เรียบร้อยแล้ว

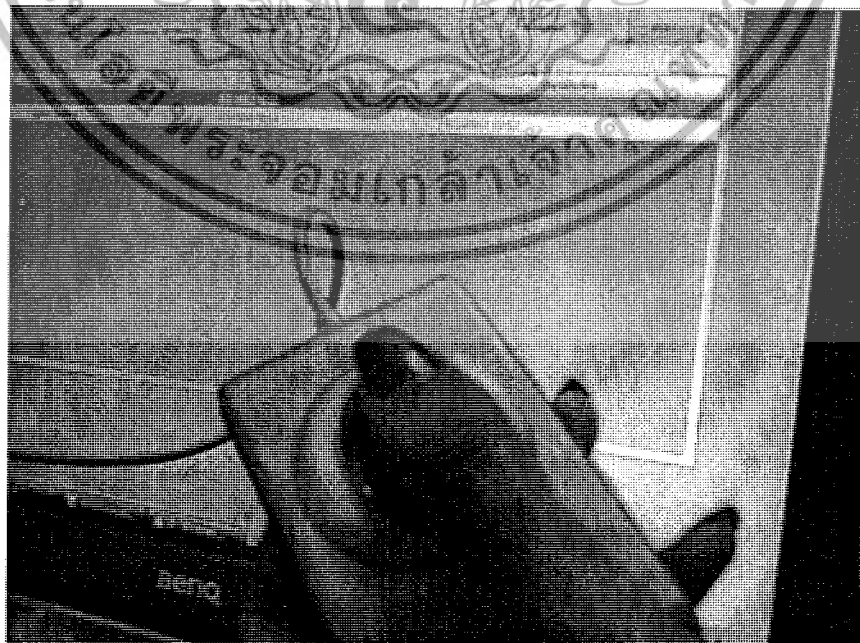
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ปุ่มบันทึกการเข้าเรียน

ปุ่มบันทึกการเรียนเป็นปุ่มสำคัญที่ใช้สำหรับการบันทึกเวลาในการเข้าเรียน/ออกเรียน โดยสามารถเลือกรายวิชาและรายชื่ออาจารย์ผู้สอนก่อนที่จะทำการบันทึกได้ เมื่อเลือกถูกต้องตามที่ต้องการแล้ว ก็ให้กดปุ่มตกลงเพื่อทำการเริ่มบันทึกเวลานี้ว่ามี



รูปที่ ก.8 หน้าต่างการบันทึกการเข้าเรียน



รูปที่ ก.9 ตัวอย่างการบันทึกเวลานี้ว่ามี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสประจำตัว	ชื่อ-นามสกุล	เลขที่โต๊ะ	วันที่เข้าเรียน
48010438	นางสาว อรุณรัตน์	10031002 1053	00000000 0000

รูปที่ ก.10 หน้าต่างแสดงรายชื่อนักศึกษาที่บันทึกเวลาเรียนในรายวิชานั้นๆ

6. ปุ่มรายงานการเข้าเรียน

ปุ่มรายงานการเข้าเรียนเป็นปุ่มสำหรับแสดงผลการเข้าเรียนของนักศึกษาทั้งหมด ว่ามาทั้งหมดกี่ครั้ง สายทั้งหมดกี่ครั้ง และขาดเรียนทั้งหมดกี่ครั้งในแต่ละรายวิชา

รายงานการเข้าเรียน

รหัสวิชา: 01232113 - MICROCOMPUTERS LABORATORY

รายงานการเข้าเรียน

รหัสวิชา: 01232113 ชื่อวิชา: MICROCOMPUTERS LABORATORY

จำนวนที่ต่อมเข้าเรียน: 10 ครั้ง

รหัสประจำตัว	ตำแหน่งชื่อ	ชื่อ - สกุล	ปกติ (วัน)	ขาด (วัน)	มาสาย (วัน)
48010031	นาย	กวัน ตั้งวิษยาบุณกุล	10	0	0
48010438	นางสาว	นฤชนก ปุบพิณาย	9	0	1

รูปที่ ก.11 หน้าต่างรายงานการเข้าเรียนของนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

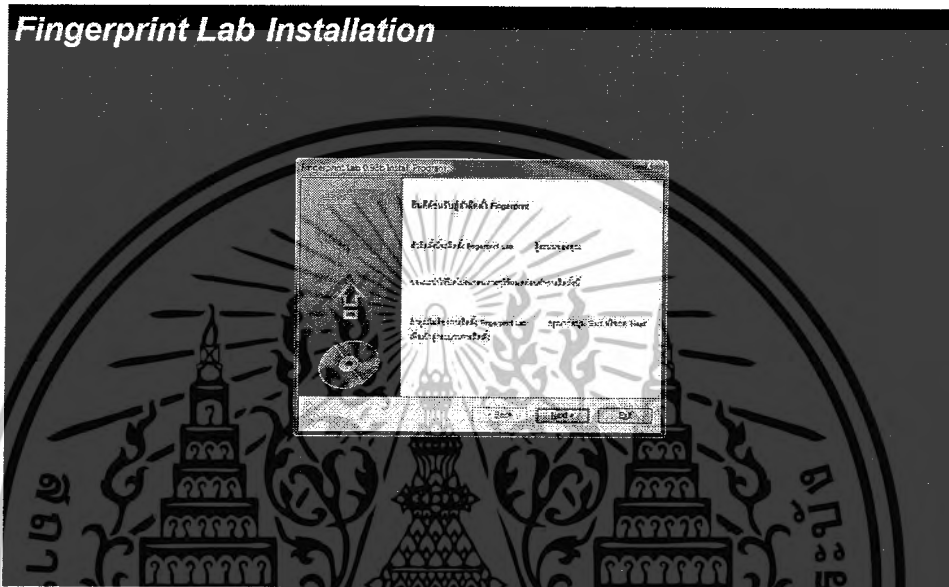


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการติดตั้งโปรแกรม

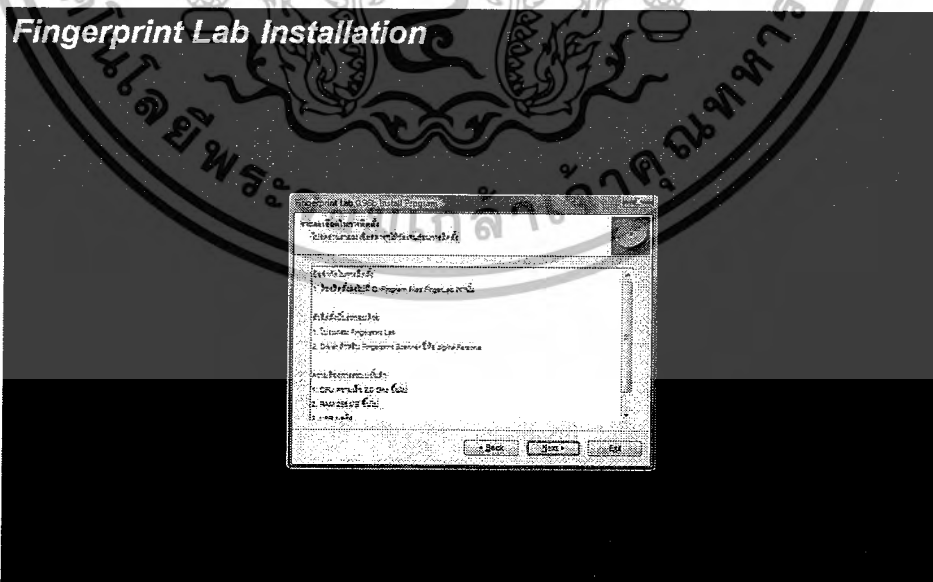
ข.1 การติดตั้ง FINGERPRINT SCAN FOR LABORATORY

1. กด Next เพื่อทำการติดตั้ง FINGERPRINT SCAN FOR LABORATORY



รูปที่ ข.1 เริ่มต้นติดตั้ง FingerLab

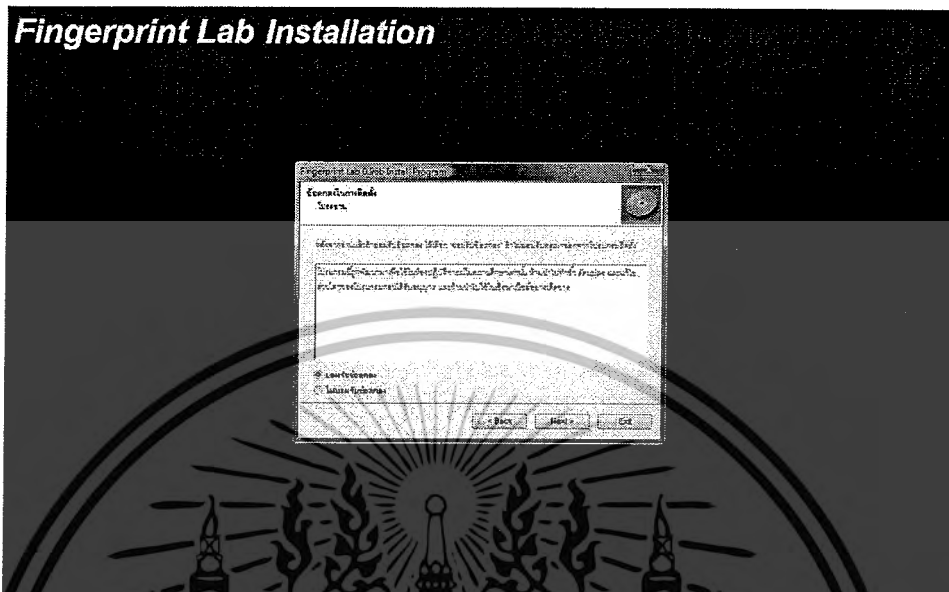
2. กด Next อีกครั้ง



รูปที่ ข.2 รายละเอียดในการติดตั้ง

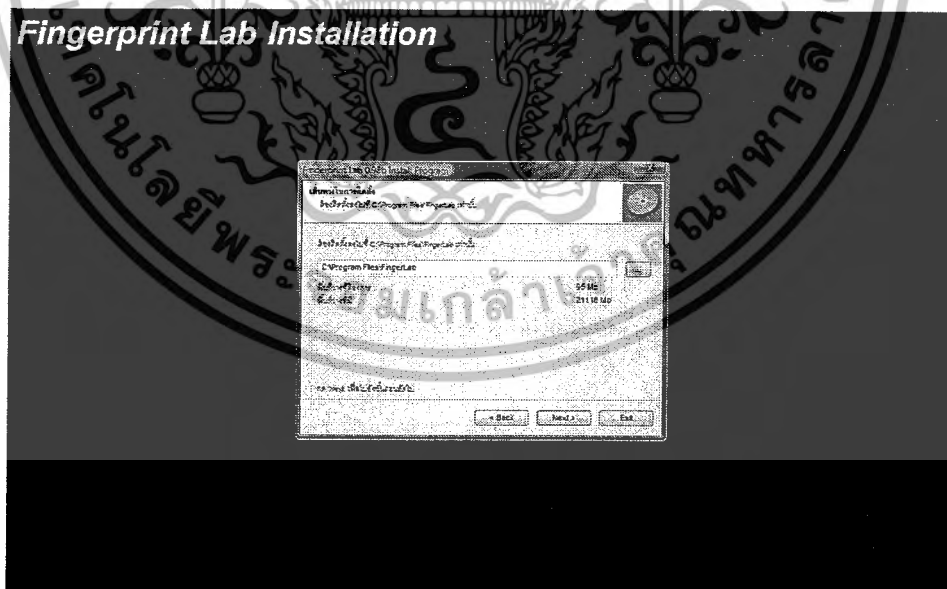
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เลือกยอมรับข้อตกลงแล้วกด Next



รูปที่ ข.3 ข้อตกลงในการติดตั้ง

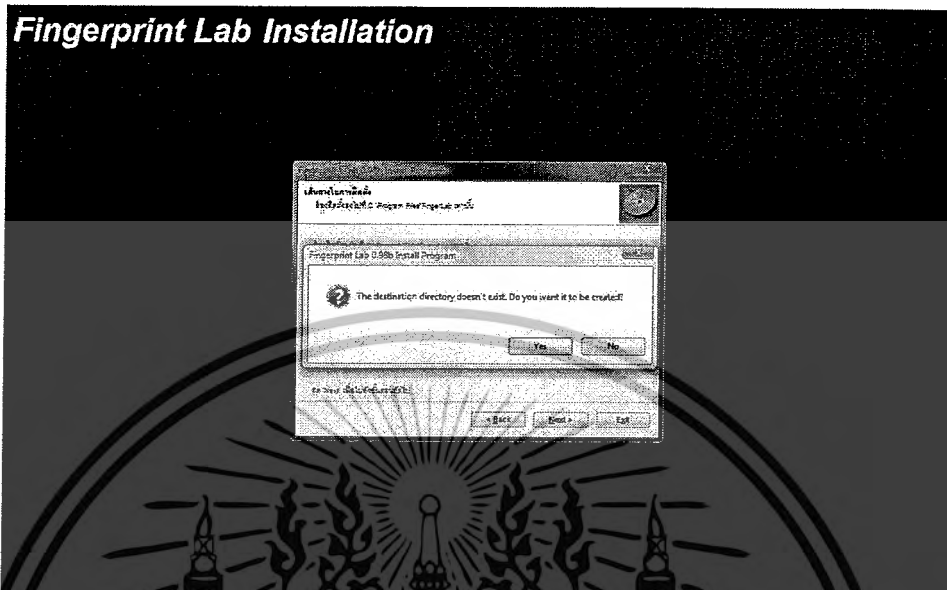
4. เลือก Directory ที่ต้องการ Install โปรแกรม ให้ Install ลงไปที่ C:\Program Files\FingerLab เท่านั้น ห้ามเปลี่ยนเด็ดขาด หลังจากนั้น ให้กด Next



รูปที่ ข.4 เลือกโฟลเดอร์การติดตั้ง

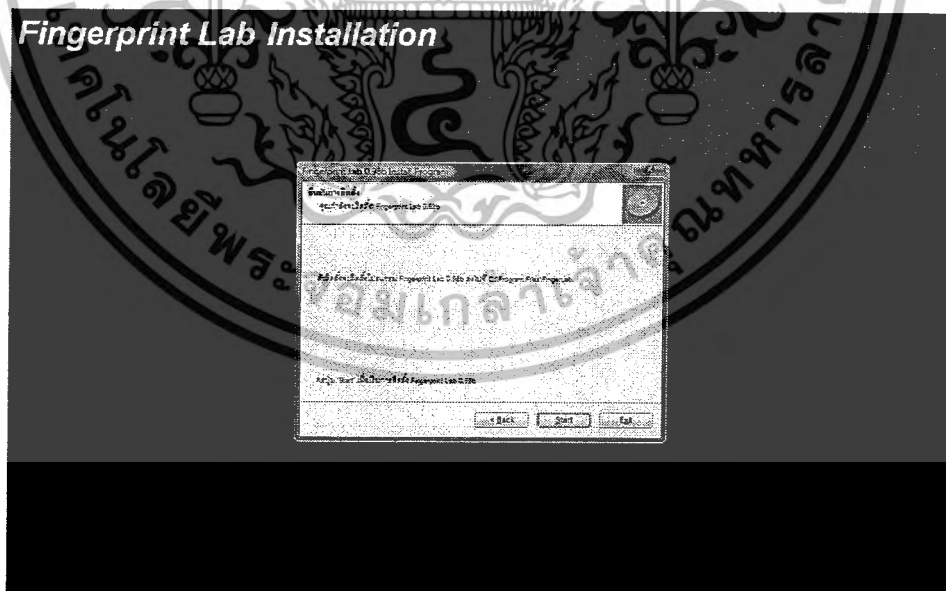
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ถ้าขึ้นดังรูปให้กด Yes เพื่อสร้างโฟลเดอร์ใหม่ขึ้นมา



รูปที่ ข.5 โปรแกรมแจ้งเตือนว่ายังไม่มีโฟลเดอร์นี้ ต้องการจะสร้างเพิ่มหรือไม่

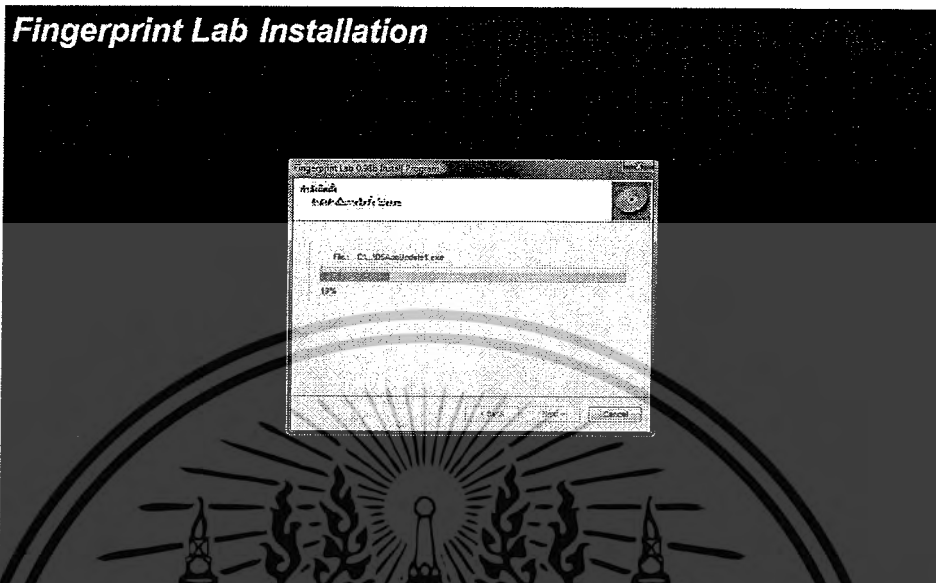
6. กด Start เพื่อเริ่ม Install โปรแกรม



รูปที่ ข.6 ยืนยันการติดตั้ง

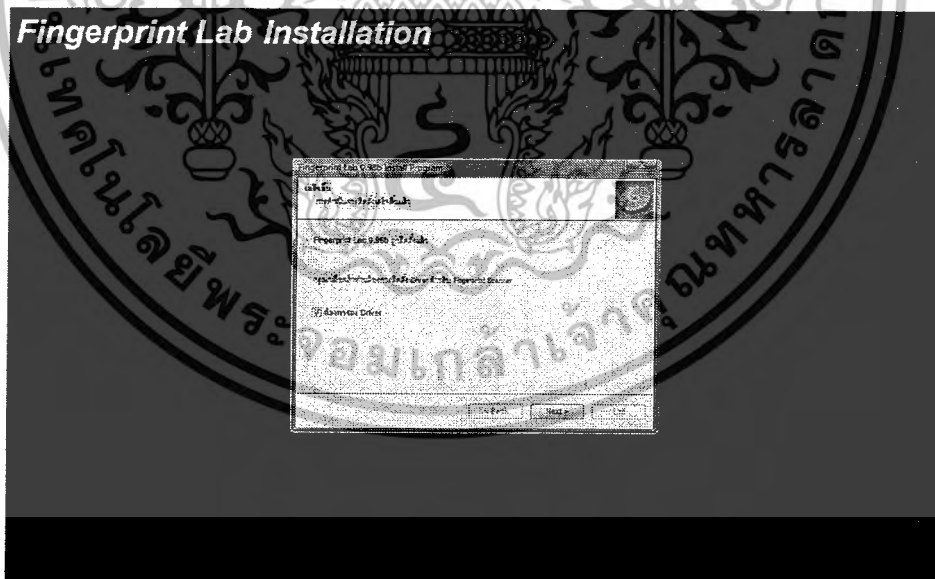
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. รอคอบ 100% เป็นอันลงโปรแกรมเสร็จ



รูปที่ ข.7 กำลัง Install

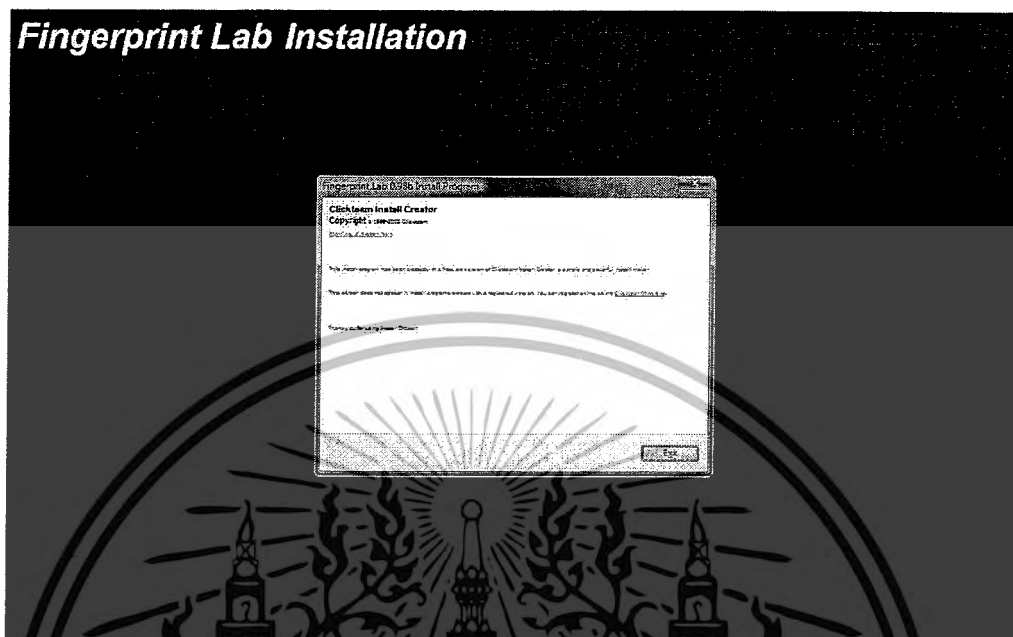
8. หากท่านยังไม่มี Driver สำหรับเครื่องสแกนลายนิ้วมือ ให้คลิกถูกที่หน้าช่อง [ต้องการลง Driver] แล้วกด Next



รูปที่ ข.8 เลือกติดตั้ง Driver เพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ลงโปรแกรม FingerLab เสร็จแล้วกด Exit



รูปที่ ข.9 เสร็จสิ้นการติดตั้ง FingerLab

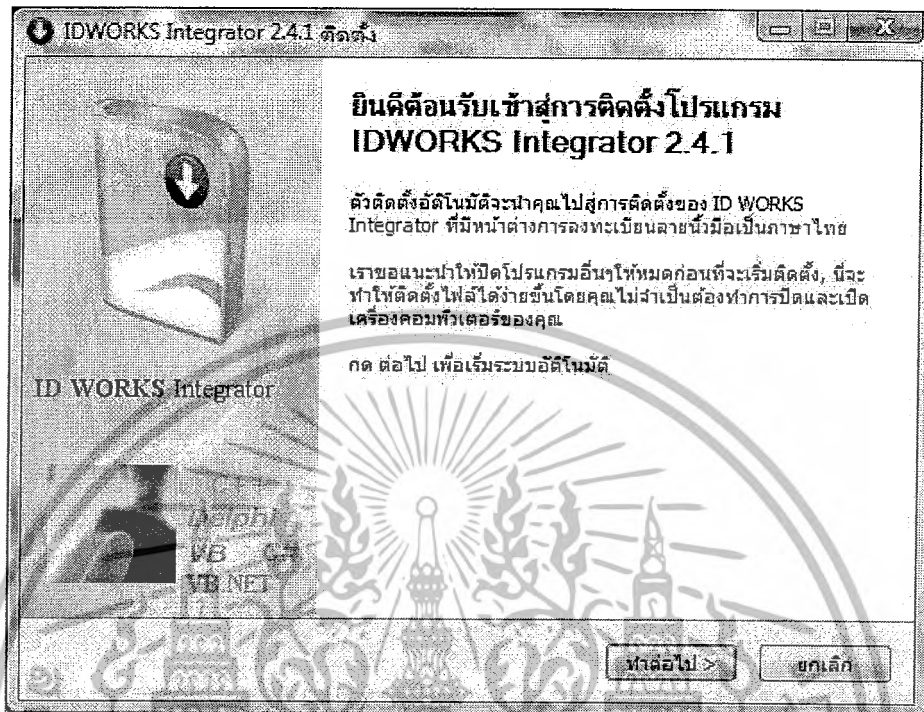
10. เลือกภาษาที่ต้องการใช้ในการติดตั้ง แล้วกดปุ่ม OK



รูปที่ ข.10 หน้าต่างเลือกภาษา

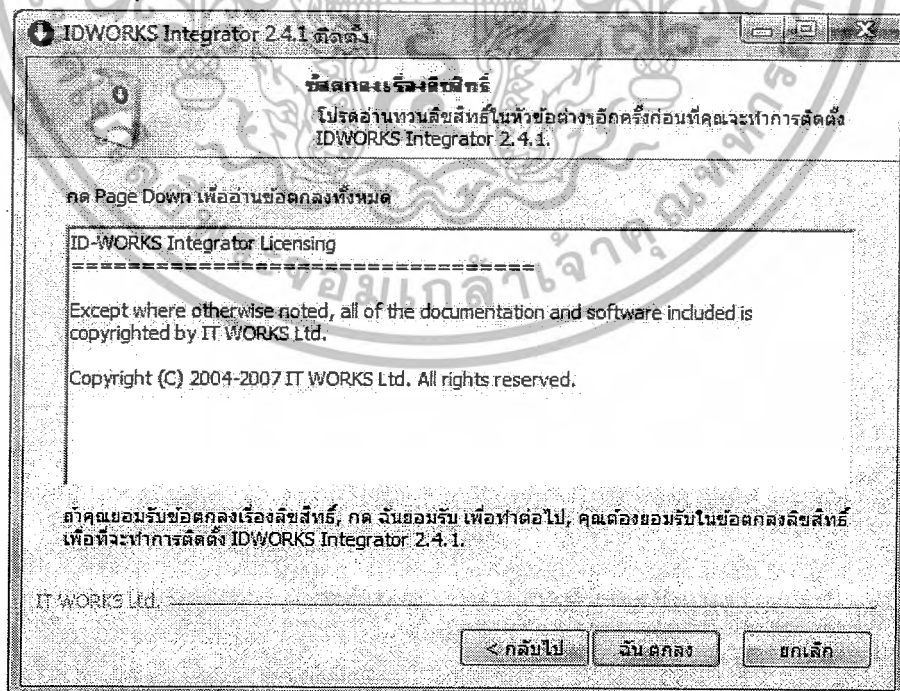
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. กดปุ่ม ทำต่อไป หรือ Next



รูปที่ ข.11 หน้าต่างการติดตั้ง IDWORKS Integrator 2.4.1

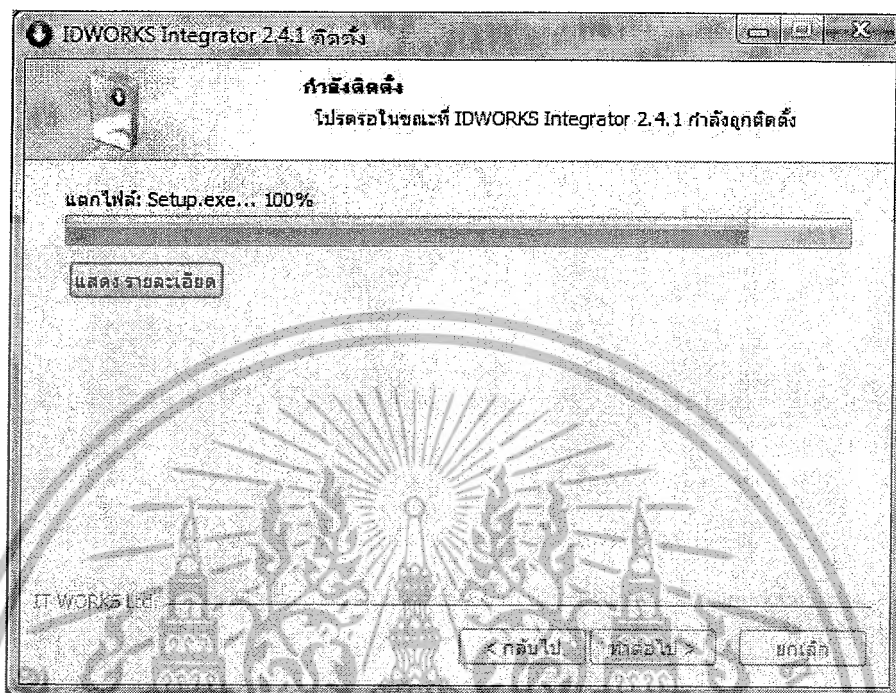
12. กดปุ่ม ถัดลง



รูปที่ ข.12 ข้อตกลงเรื่องลิขสิทธิ์

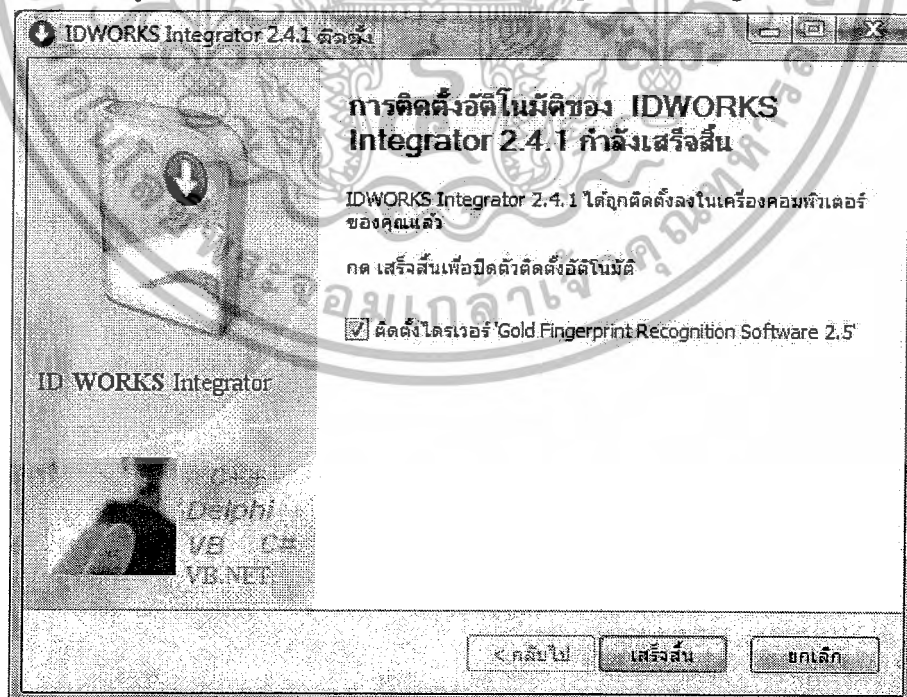
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. รอกจนครบ 100%



รูปที่ ข.13 กำลัง Install IDWORKS Integrator 2.4.1

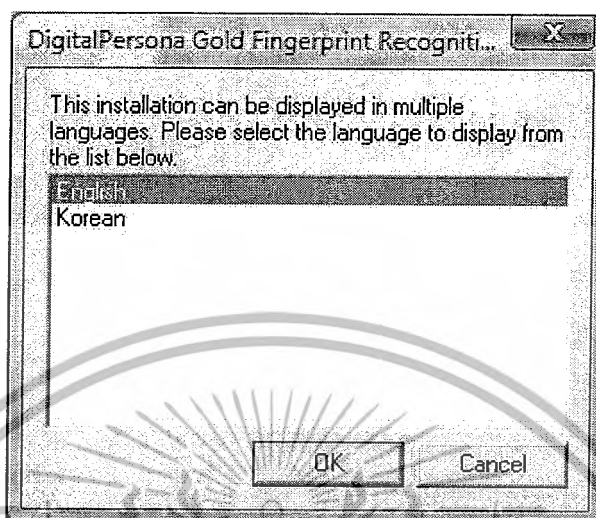
14. ตักถูกที่หน้าช่อง ติดตั้งไดรเวอร์ 'Gold FingerPrint Recognition Software 2.5'



รูปที่ ข.14 ติดตั้ง IDWORKS Integrator 2.4.1 กำลังเสร็จสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15. เลือกภาษาในการทำงาน



รูปที่ ข.15 เลือกภาษาในการติดตั้ง Driver

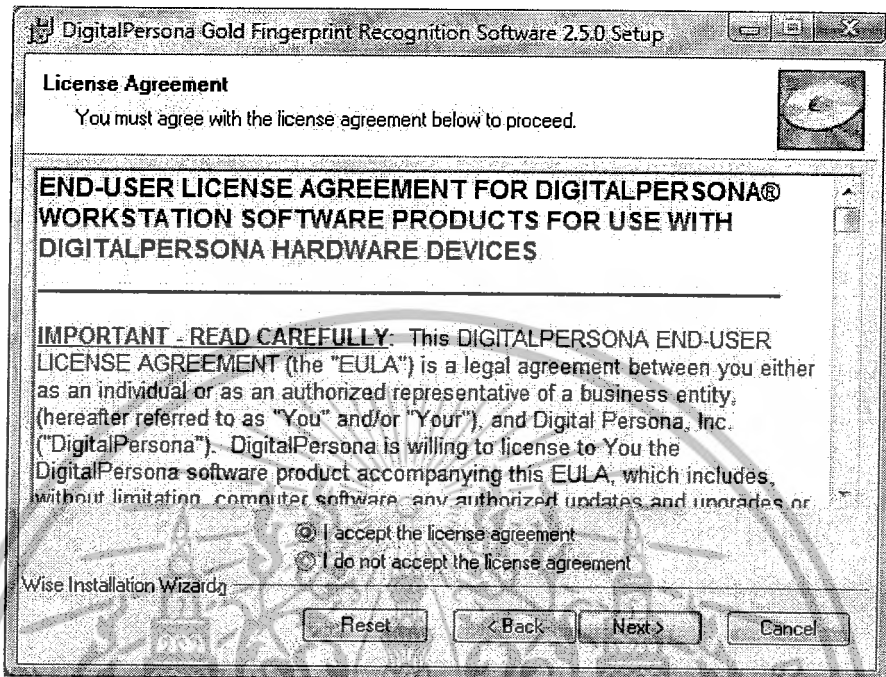
16. กด Next



รูปที่ ข.16 เริ่มการติดตั้ง Driver

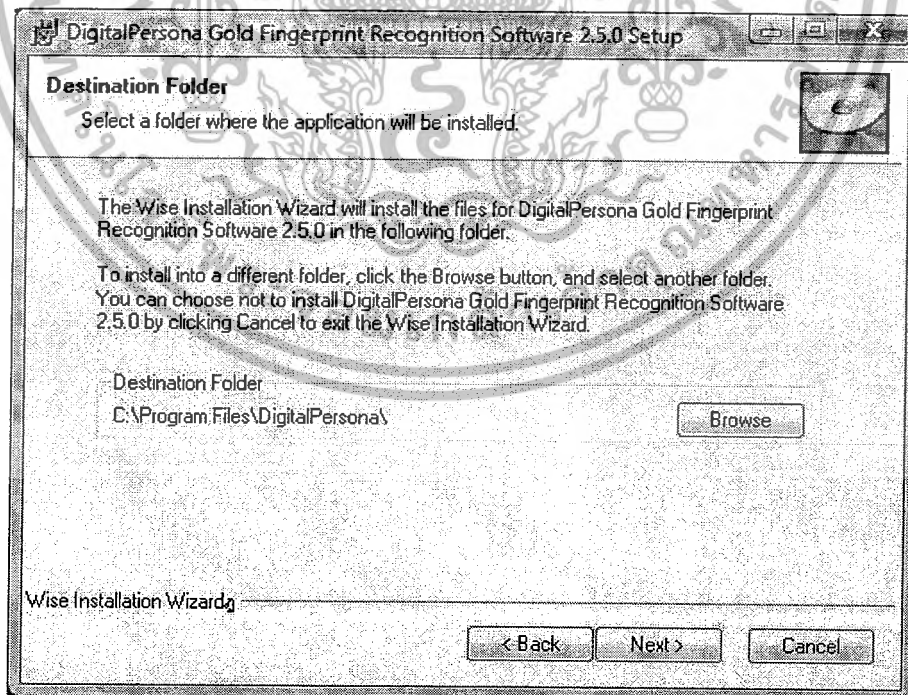
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. เลือก I accept the license agreement



รูปที่ ข.17 ข้อตกลงในการติดตั้ง Driver

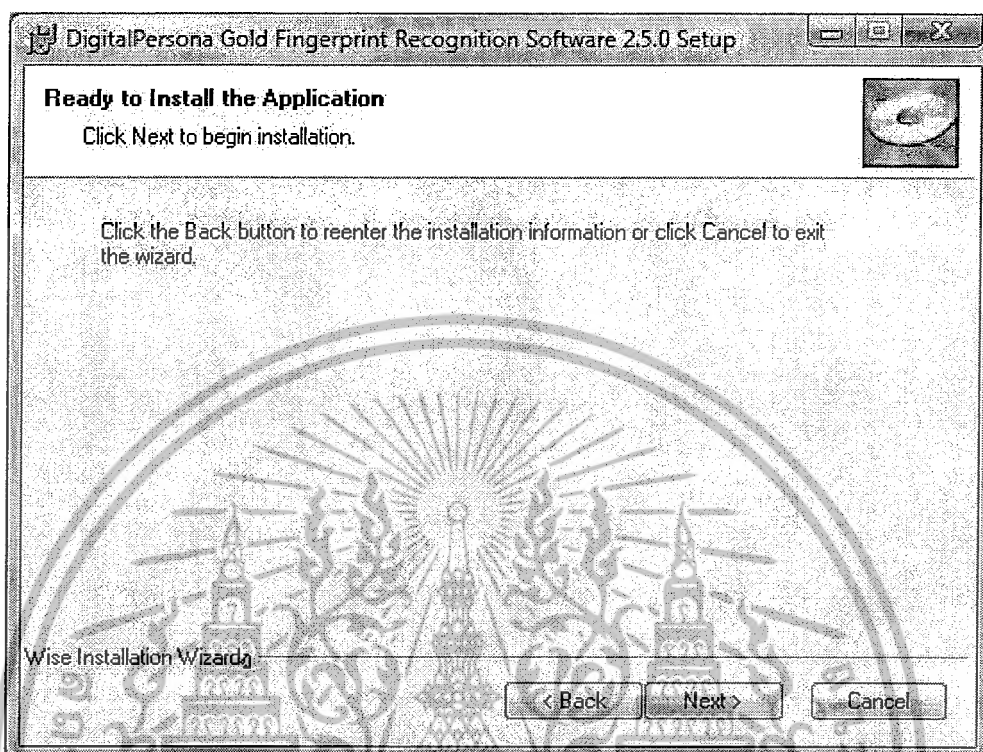
18. เลือก Directory ที่ต้องการติดตั้ง



รูปที่ ข.18 เลือกโฟลเดอร์ในการติดตั้ง Driver

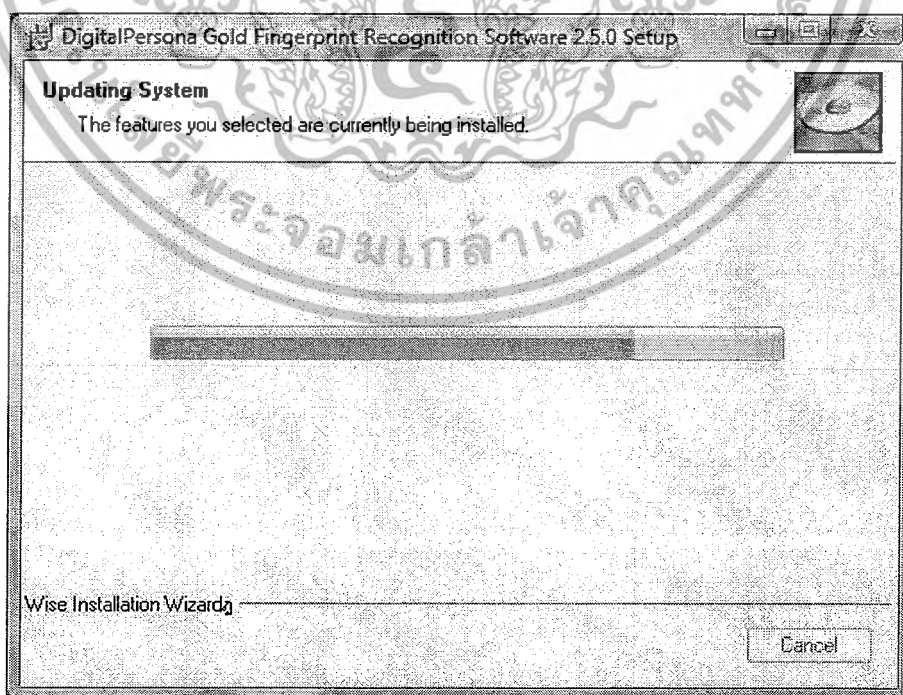
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19. กด Next



รูปที่ ข.19 หน้าต่างพร้อมติดตั้ง Driver

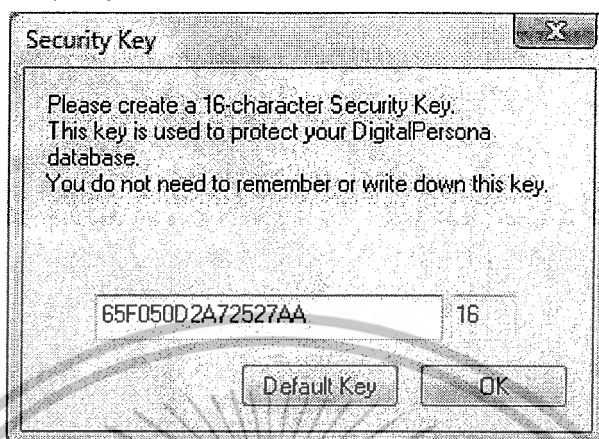
20. รอจนครบ 100%



รูปที่ ข.20 กำลัง Install Driver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

21. ใส่ Security Key 65F050D2A72527AA



รูปที่ ข.21 หน้าต่างใส่ Security Key

22. กดปุ่ม Finish เป็นอันติดตั้งเสร็จสมบูรณ์พร้อมใช้งาน โปรแกรม FingerLab



รูปที่ ข.22 ติดตั้งเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



IT WORKS
IDWORKS Integrator SDK (DLLs)
API References

V2.0
Updated: 07 September 2005



การซัพพอร์ตผลิตภัณฑ์

ไอที เวิร์คส์ ตั้งใจที่จะให้ผู้พัฒนาที่ใช้งาน *ID-WORKS Integrator™ SDK* สามารถใช้งานชุด SDK ดังกล่าวได้ตามความต้องการและได้รับความพึงพอใจสูงสุดในการใช้งานผลิตภัณฑ์

สำหรับผู้ใช้งานที่มีข้อสงสัยในการใช้งานหรือต้องการติดต่อกับทีมวิศวกรของไอที เวิร์คส์ กรุณาติดต่อผ่านทาง

E-mail (Preferred) biosupport@itworksolutions.com

MSN Messenger [idissupp001@hotmail.com](msn://idissupp001@hotmail.com)

การแจ้งข้อผิดพลาดในเอกสาร

สำหรับ "*ID-WORKS Integrator SDK - API References*" ฉบับนี้ทางทีมงาน ไอที เวิร์คส์ ตั้งใจที่จะทำให้รายละเอียดในเอกสารมีความถูกต้องและครบถ้วนที่สุด ไอที เวิร์คส์ ขอสงวนสิทธิ์ในการตรวจสอบและแก้ไขเนื้อหาในเอกสารฉบับนี้เป็นระยะๆ โดยไม่จำเป็นต้องแจ้งต่อผู้ใดทั้งสิ้น ถ้าหากท่านพบข้อผิดพลาดใดๆในเอกสารฉบับนี้ กรุณาส่งอีเมลคำชี้แจงมายัง

E-mail (Preferred) info@itworksolutions.com

หรือติดต่อเราได้ที่

IT WORKS Ltd.

100/103 Wongwanich Complex B Tower, 30th Fl, Rama IX Road, Huaykwang Subdistrict, Huaykwang District, Bangkok 10320

TEL: (+662) 645-1200-7 FAX: (+662) 645-1233

คำชี้แจง

ไอที เวิร์คส์ ตั้งใจที่จะให้ผู้พัฒนาที่ใช้งาน *ID-WORKS Integrator™ SDK* สามารถใช้งานผลิตภัณฑ์ดังกล่าวได้อย่างที่ต้องการและได้รับความพึงพอใจสูงสุด อย่างไรก็ตาม ไอที เวิร์คส์ ขอสงวนสิทธิ์ไม่รับผิดชอบหรือรับประกันไม่ว่าในกรณีใดๆทั้งสิ้นถึงความเหมาะสมและความเข้ากันได้ของ *ID-WORKS Integrator™ SDK* กับระบบที่ท่านพัฒนา

ID-WORKS Integrator SDK และ Software ทุกตัวของไอที เวิร์คส์ ที่แสดงในเอกสารฉบับนี้จำหน่ายในรูปแบบของผลิตภัณฑ์สำเร็จรูป เว้นแต่กรณีที่มีการตกลงกันกับไอทีเวิร์คส์เป็นกรณีไป

เครื่องหมายการค้า

ไอทีเวิร์คส์ ตั้งใจที่จะบ่งบอกถึงเจ้าของผลิตภัณฑ์ทุกครั้งที่มีการกล่าวถึงเครื่องหมายการค้าของบริษัทอื่น เครื่องหมายการค้าจะมีการกล่าวถึงอยู่ในจุดต่างๆของเอกสารฉบับนี้ โดยไอที เวิร์คส์ใช้เครื่องหมายการค้าต่างๆเพื่อผลในการอธิบายและเพื่อผลประโยชน์ของเจ้าของเครื่องหมายการค้าเท่านั้น ไอที เวิร์คส์ไม่มีความตั้งใจใดๆในการลอกเลียนแบบใดๆทั้งสิ้น

Microsoft™, Windows XP™, Windows 98™, Windows 2000™, Windows Millennium™, Windows NT™ เป็นเครื่องหมายการค้าของ บริษัท ไมโครซอฟท์ คอร์ปอเรชั่น

DigitalPersona™, U.are.U™ เครื่องหมายการค้าของ บริษัท Digital Persona Inc.

ID-WORKS™, ID-WORKS Integrator™, TimeWORKS Integrator™, TimeWORKS™ ระบบลงเวลาด้วยลายนิ้วมือ เป็นเครื่องหมายการค้าของ บริษัท ไอที เวิร์คส์ จำกัด

สารบัญ

ารชีพพอร์ทผลิตภัณฑ์	2
การแจ้งข้อผิดพลาดในเอกสาร	2
คำชี้แจง	2
เครื่องหมายการค้า	2
ารบัญ	3
หน้า	8
ความต้องการระบบ	9
ความต้องการต่ำสุด	9
ความต้องการของระบบที่แนะนำ	9

DWORKS INTEGRATOR API

10

สรุปฟังก์ชันทั้งหมดใน IDWORKS INTEGRATOR - ส่วนการประมวลผลลายนิ้วมือ	10
การเริ่มและเลิกใช้งานไลบรารี	10
การสร้างและปิด context และการ Activate	10
การควบคุมเครื่องอ่านลายนิ้วมือ	10
การเพิ่มและลบลายนิ้วมือ	10
สรุปฟังก์ชันทั้งหมดใน IDWORKS INTEGRATOR - ส่วนการลงทะเบียนลายนิ้วมือ	10
การเริ่มและเลิกใช้งานไลบรารี	10
การสร้างและปิด context และการ Activate	10
การลงทะเบียนลายนิ้วมือ	11
สรุปฟังก์ชันทั้งหมดใน IDWORKS INTEGRATOR - ส่วนข้อมูลเครื่องอ่านลายนิ้วมือ	11
การเริ่มและเลิกใช้งานไลบรารี	11
การดึงข้อมูลเครื่องอ่านลายนิ้วมือ	11
IDWORKS INTEGRATOR - ส่วนการประมวลผลลายนิ้วมือ	12
รายละเอียดฟังก์ชัน (Function Definitions)	12
idi_InitModule	12
Function Definition	12
Parameters	12
Return Value	12
idi_FinalizeModule	12
Function Definition	12
Parameters	13
Return Value	13
idi_CreateContext	13
Function Definition	13
Parameters	13
Return Value	13
idi_CloseContext	13
Function Definition	14
Parameters	14
Return Value	14
idi_ActivateSensor	14
Function Definition	14

Parameters	15
Return Value.....	15
idi_DeactivateSensor	15
Function Definition.....	16
Parameters	16
Return Value.....	16
idi_StartCapturing	16
Function Definition.....	17
Parameters	17
Return Value.....	17
idi_StopCapturing	17
Function Definition.....	17
Parameters	18
Return Value.....	18
idi_RegisterSensorEventProc	18
Function Definition.....	18
Parameters	19
Return Value.....	19
idi_UnregisterSensorEventProc	19
Function Definition.....	19
Parameters	20
Return Value.....	20
idi_RegisterIdentEventProc	20
Function Definition.....	20
Parameters	21
Return Value.....	21
idi_UnregisterIdentEventProc	21
Function Definition.....	21
Parameters	21
Return Value.....	22
idi_fp_Add	22
Function Definition.....	22
Parameters	22
Return Value.....	22
idi_fp_ClearAll	23
Function Definition.....	23
Parameters	23
Return Value.....	23
ประเภทข้อมูล (Data Types)	24
IDI_RCODE	24
Data-type Definition.....	24
IDI_CONTEXT	24
Data-type Definition.....	24
IDI_IDENTEVENTMSG	24
Data-type Definition.....	24
หมายเหตุ.....	25
IDI_IDENTEVENTTYPE	25
Data-type Definition.....	25
IDI_IDENTEVENTPARAM	25
Data-type Definition.....	26
IDI_REFPARAM	26
Data-type Definition.....	26
IDI_IDENTEVENTPROC	26
Data-type Definition.....	26
Parameters	27
IDI_IDENTEVENTMATCHEDSTRUCT	27
Data-type Definition.....	27
Members.....	27
IDI_SENSOREVENTMSG	27
Data-type Definition.....	27
Members.....	28
หมายเหตุ.....	28
IDI_SENSOREVENTPROC	28

Data-type Definition	29
Parameters	29
IDI_SENSOREVENTIMAGEACQUIREDSTRUCT	29
Data-type Definition	29
Members	29
FP_FPINFO	30
Data-type Definition	30
Members	30
ค่าคงที่ (Constants)	32
ค่าคงที่สำหรับ IDI_RCODE	32
ค่าคงที่สำหรับ IDI_IDENTEVENTTYPE	32
ค่าคงที่สำหรับ IDI_SENSORTYPE	32
ค่าคงที่สำหรับ IDI_SENSOREVENTTYPE	32
ID-WORKS INTEGRATOR – ส่วนข้อมูลเครื่องอ่านลายนิ้วมือ	33
รายละเอียดฟังก์ชัน (Function Definitions)	33
isl_IsModuleInited	33
Function Definition.....	33
Parameters	33
Return Value.....	33
isl_InitModule	33
Function Definition.....	33
Parameters	34
Return Value.....	34
isl_FinalizeModule	34
Function Definition.....	34
Parameters	34
Return Value.....	34
isl_GetDeviceCount	34
Function Definition.....	34
Parameters	35
Return Value.....	35
isl_GetDeviceList	35
Function Definition.....	35
Parameters	35
Return Value.....	36
ประเภทข้อมูล (Data Types)	37
ISL_RCODE	37
Data-type Definition.....	37
ISL_DEVICETYPE	37
Data-type Definition.....	37
ISL_DEVICEID	37
Data-type Definition	37
ISL_CHAR	38
Data-type Definition	38
ISL_SENSORINFO	38
Data-type Definition	38
Members.....	38
ค่าคงที่ (Constants)	40
ค่าคงที่ทั่วไป	40
ค่าคงที่สำหรับ ISL_RCODE	40
ค่าคงที่สำหรับ ISL_DEVICETYPE	40
ID-WORKS INTEGRATOR – ส่วนช่วยการบันทึกลายนิ้วมือ	41
รายละเอียดฟังก์ชัน (Function Definitions)	41
fpe_InitModule	41
Function Definition.....	41
Parameters	41
Return Value.....	41
fpe_FinalizeModule	41

Function Definition.....	41
Parameters	42
Return Value.....	42
fpe_CreateContext.....	42
Function Definition.....	42
Parameters	42
Return Value.....	42
fpe_CloseContext.....	42
Function Definition.....	42
Parameters	43
Return Value.....	43
fpe_ActivateSensor	43
Function Definition.....	43
Parameters	44
Return Value.....	44
fpe_StartEnrolment.....	44
Function Definition.....	44
Parameters	45
Return Value.....	45
ประเภทข้อมูล (Data Types)	46
FPE_RCODE	46
Data-type Definition	46
FPE_CONTEXT.....	46
Data-type Definition	46
FPE_SENSORTYPE	46
Data-type Definition	46
ค่าคงที่ (Constants).....	47
ค่าคงที่ทั่วไป.....	47
ค่าคงที่สำหรับ ISL_RCODE.....	47
ค่าคงที่สำหรับ ISL_DEVICETYPE.....	47

APPENDIX: A - การแก้ไขข้อผิดพลาดระหว่างการติดตั้ง..... 48

การแก้ไขข้อผิดพลาดระหว่างการติดตั้ง.....	48
--	----

APPENDIX: B - การใช้งานและการดูแลรักษาเครื่องอ่านลายนิ้วมือ..... 49

การใช้งานเครื่องอ่านลายนิ้วมืออย่างถูกวิธี.....	49
พยายามแตะให้กว้างครอบคลุมพื้นที่มากที่สุด	49
วางนิ้วให้ขนานกับแนวราบ	49
ไม่ควรกดแรงเกินไป	49
เล็บยาว?.....	49
อย่าหมุนนิ้ว	49
มือแห้ง	49
การป้องกันปัญหาที่อาจเกิดขึ้น.....	50
มือสกปรก?.....	50
แสงแดด	50
แอลกอฮอล์และน้ำยาทำความสะอาด	50
น้ำตาล	50
การทำความสะอาดเครื่องอ่านลายนิ้วมือ	50
วิธีที่แนะนำในการทำความสะอาด	50

ข้อควรระวังในการทำความสะอาด..... 50



บทนำ

ไอที เวอร์คส์ คือผู้นำในการนำเทคโนโลยีไบโอเมตริกส์มาสู่การพัฒนาและประยุกต์ใช้อย่างแพร่หลายในแอปพลิเคชันต่างๆ เพื่อผลักดันการนำเทคโนโลยีการตรวจสอบลายนิ้วมือมาประยุกต์ใช้ในวงกว้าง เราได้พัฒนา ID-WORKS Integrator ขึ้นให้คุณพัฒนาแอปพลิเคชันด้วยเทคโนโลยีนี้ได้อย่างสะดวก รวดเร็วและมีประสิทธิภาพสูงสุด

นักพัฒนาสามารถนำ ID-WORKS Integrator SDK มาประยุกต์ใช้กับแอปพลิเคชันต่างๆ ของตนได้ในหลายๆกรณี เช่น การใช้ลายนิ้วมือแทนรหัสผ่าน การใช้ลายนิ้วมือแทนบัตรพนักงาน บัตรคนไข้ บัตรนักเรียน บัตรห้องสมุดและบัตรอื่นๆ เพื่อความสะดวกถูกต้องในการยืนยันตัวตนบุคคลและความประหยัดจากการที่ไม่ต้องจัดทำบัตร มีแอปพลิเคชันหลายประเภทที่สามารถนำเทคโนโลยีการยืนยันลายนิ้วมือของ ID-WORKS Integrator มาใช้งานได้เหมาะสม ยกตัวอย่างเช่น

- Time & Attendance Applications
- Healthcare Applications
- Membership Management Application
- Childcare Security Applications
- Point of Sale Applications
- Library Management Systems
- Secure Data Storage Card/Smart Card Applications
- Physical/Logical Access Control Applications
- Internet/Financial Transaction Authentication

ชุดพัฒนา ID-WORKS Integrator	ได้ถูกออกแบบมาให้ทำงานบน Windows Platform
โดยชุดที่จำหน่ายทั่วไปจะมาพร้อมกับเครื่องอ่าน DigitalPersona U.are.U 4000 - 4000B	
และจะทำงานกับเครื่องอ่าน U.are.U 4000 เท่านั้น	อย่างไรก็ตามชุด SDK

นี้อาจถูกปรับปรุงให้ทำงานร่วมกับเครื่องอ่านลายนิ้วมือจากผู้ผลิตอื่นๆ ได้ในกรณีที่ต้องการ ID-WORKS Integrator มีความสามารถเหนือกว่า SDK ของผู้ผลิตเครื่องอ่านลายนิ้วมือยี่ห้อต่างๆ เช่น ชุดพัฒนา DigitalPersona's SDK-DLL Gold Version และ DigitalPersona's SDK-DCOM Platinum โดย ID-WORKS Integrator รองรับการค้นหาลายนิ้วมือแบบ Optimized 1:N (Identification) (มีความเร็วในการค้นหาเฉลี่ยมากกว่า 2,000 ลายนิ้วมือ/วินาที (P4, 1.8 GHz) เทียบกับความเร็วประมาณ 100 นิ้ว/วินาทีที่ได้จากการนำ 1:1 verify module จาก SDK ของผู้ผลิตเครื่องอ่านลายนิ้วมือมาทำการค้นหา) ในกรณีทั่วไปนักพัฒนาจึงสามารถพัฒนาโปรแกรมค้นหาลายนิ้วมือให้ใช้งานได้โดยมีประสิทธิภาพโดยไม่ต้องมีการใช้งานร่วมกับรหัส ID ใด

ชุดพัฒนา ID-WORKS Integrator ใช้ได้กับภาษาที่สนับสนุนการเรียก DLL หรือ ActiveX เช่น C/C++, Delphi, VB, .NET ฯลฯ โดยจะประกอบด้วย ไลบรารีสำหรับประมวลผลลายนิ้วมือในรูปแบบของ DLL และ ActiveX, คู่มืออธิบายการใช้งาน และ Code ตัวอย่างสำหรับการ implement เพื่อใช้งานไลบรารี

ข้อมูลลายนิ้วมือที่ได้จากชุด SDK	จะได้มาจากการดึงข้อมูลรายละเอียดต่างๆ	ออกมาจากรูปลายนิ้วมือ
ซึ่งเมื่อจัดเก็บจะอยู่ในรูปข้อมูล binary	ซึ่งมีขนาดประมาณ	400-650 bytes

และสามารถนำไปจัดเก็บตามที่ผู้พัฒนาต้องการได้ ซึ่งอาจจะจัดเก็บในฐานข้อมูล ในไฟล์ หรือในบัตรสมาร์ทการ์ด

ความต้องการระบบ

ความต้องการต่ำสุด

- เครื่อง PC แบบ Pentium III หรือดีกว่า
- หน่วยประมวลผลความเร็ว 800 MHz หรือเร็วกว่า
- พื้นที่ว่างในฮาร์ดดิสก์ อย่างน้อย 20 MB
- หน่วยความจำอย่างต่ำ 64 MB
- มีช่อง USB Port ว่างอย่างน้อย 1 ช่อง
- DigitalPersona U.are.U 4000
- DigitalPersona U.are.U Integrator Gold (จะมาพร้อมกับชุด IDWORKS Integrator SDK)

ความต้องการของระบบที่แนะนำ

- เครื่อง PC แบบ Pentium 4 หรือดีกว่า
- หน่วยประมวลผลความเร็ว 1.6 GHz หรือเร็วกว่า
- พื้นที่ว่างในฮาร์ดดิสก์ อย่างน้อย 500 MB
- หน่วยความจำอย่างต่ำ 256 MB
- มีช่อง USB Port ว่างอย่างน้อย 1 ช่อง
- DigitalPersona U.are.U 4000
- DigitalPersona U.are.U Integrator Gold (จะมาพร้อมกับชุด IDWORKS Integrator SDK)



IDWORKS Integrator API

สรุปฟังก์ชันทั้งหมดใน IDWORKS Integrator - ส่วนการประมวลผลลายนิ้วมือ

การเริ่มและเลิกใช้งานไลบรารี

ชื่อฟังก์ชัน	คำอธิบาย
idi_InitModule	เริ่มการทำงานของไลบรารี
idi_FinalizeModule	จบการทำงานของไลบรารี

การสร้างและปิด context และการ Activate

ชื่อฟังก์ชัน	คำอธิบาย
idi_CreateContext	สร้าง IDWORKS processor context
idi_CloseContext	ปิด IDWORKS processor context
idi_ActivateSensor	ลงทะเบียนเครื่องอ่านลายนิ้วมือ ก่อนเริ่มการใช้งาน

การควบคุมเครื่องอ่านลายนิ้วมือ

ชื่อฟังก์ชัน	คำอธิบาย
idi_StartCapturing	เริ่มการรอรับการแตะนิ้วมือ
idi_StopCapturing	หยุดรอรับการแตะนิ้วมือ
idi_RegisterIdentEventProc	register callback function สำหรับเรียกกลับ เพื่อแจ้งผลการค้นหาลายนิ้วมือกับ application
idi_UnregisterIdentEventProc	unregister callback function ที่ register ไว้โดยฟังก์ชัน idi_RegisterIdentEventProc

การเพิ่มและลบลายนิ้วมือ

ชื่อฟังก์ชัน	คำอธิบาย
idi_fp_Add	เพิ่มลายนิ้วมือเข้าไปในฐานข้อมูลชั่วคราวใน context
idi_fp_ClearAll	ลบลายนิ้วมือทั้งหมดในฐานข้อมูลชั่วคราวใน context

สรุปฟังก์ชันทั้งหมดใน IDWORKS Integrator - ส่วนการลงทะเบียนลายนิ้วมือ

การเริ่มและเลิกใช้งานไลบรารี

ชื่อฟังก์ชัน	คำอธิบาย
fpe_InitModule	เริ่มการทำงานของไลบรารี
fpe_FinalizeModule	จบการทำงานของไลบรารี

การสร้างและปิด context และการ Activate

ชื่อฟังก์ชัน	คำอธิบาย
fpe_CreateContext	สร้าง context สำหรับการลงทะเบียนลายนิ้วมือ
fpe_CloseContext	ปิด context สำหรับการลงทะเบียนลายนิ้วมือ
fpe_ActivateSensor	ลงทะเบียนเครื่องอ่านลายนิ้วมือ ก่อนเริ่มการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีนำไปใช้

การลงทะเบียนลายนิ้วมือ

ชื่อฟังก์ชัน	คำอธิบาย
<code>fpe_StartEnrolment</code>	เริ่มการลงทะเบียนลายนิ้วมือ

รูปฟังก์ชันทั้งหมดใน IDWORKS Integrator – ส่วนข้อมูลเครื่องอ่านลายนิ้วมือ

การเริ่มและเลิกใช้งานไลบรารี

ชื่อฟังก์ชัน	คำอธิบาย
<code>isl_InitModule</code>	เริ่มการทำงานของไลบรารี
<code>isl_FinalizeModule</code>	จบการทำงานของไลบรารี
<code>isl_IsModuleInited</code>	ตรวจสอบว่าได้ทำการ Initialize แล้วหรือไม่

การดึงข้อมูลเครื่องอ่านลายนิ้วมือ

ชื่อฟังก์ชัน	คำอธิบาย
<code>isl_GetDeviceCount</code>	ใช้นับจำนวนเครื่องอ่านลายนิ้วมือในระบบ
<code>isl_GetDeviceList</code>	ดึงข้อมูลเครื่องอ่านลายนิ้วมือทั้งหมดในระบบ

IDWORKS Integrator – ส่วนการประมวลผลลายนิ้วมือ

DLL Name : tmwidw.dll

ทุกฟังก์ชันจะ return ค่าเป็น IDI_R_OK หากทำงานได้สำเร็จ ยกเว้นหากได้หมายเหตุไว้ว่าเป็นอย่างอื่น

รายละเอียดฟังก์ชัน (Function Definitions)

idi_InitModule

IDWORKS Integrator Module: tmwitw.dll

ก่อนการใช้งาน โมดูล จะต้องเรียกฟังก์ชันนี้ก่อนที่จะมีการใช้งานฟังก์ชันอื่นๆ เสมอ

Function Definition

C/C++-style function definition:

```
typedef void (__stdcall *idi_InitModule) (void);
```

Pascal-style function definition:

```
idi_InitModule: procedure; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Sub idi_InitModule Lib "TMWIDW.DLL" ()
```

Parameters

ไม่มี parameter

Return Value

ไม่มี return value

idi_FinalizeModule

IDWORKS Integrator Module: tmwitw.dll

เมื่อเลิกใช้งาน โมดูล ต้องเรียกฟังก์ชันนี้เพื่อเคลียร์ข้อมูลต่างๆ ที่อาจค้างอยู่ใน โมดูล

Function Definition

C/C++-style function definition:

```
typedef void (__stdcall *idi_FinalizeModule) (void);
```

Pascal-style function definition:

```
idi_FinalizeModule: procedure; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Sub idi_FinalizeModule Lib "TMWIDW.DLL" ()
```

Parameters

ไม่มี parameter

Return Value

ไม่มี return value

idi_CreateContext

IDWORKS Integrator Module: tmwitw.dll

Context เป็นเสมือนตัวในการเข้าใช้ฟังก์ชัน หลังจากได้ตัวหรือ context แล้ว เมื่อต้องการเรียกใช้ฟังก์ชันอื่นๆ ต้องส่งตัวหรือ context นี้ไปเป็น parameter ในการเรียกฟังก์ชันเสมอ

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE ( __stdcall *idi_CreateContext) (
    IDI_CONTEXT * AProcessorContext
);
```

Pascal-style function definition:

```
idi_CreateContext: function(
    var AProcessorContext : IDI_CONTEXT
) : IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_CreateContext Lib "TMWIDW.DLL"
(
    _
    ByRef AProcessorContext As Long _
) As Long
```

Parameters

AProcessorContext

ถ้าการสร้าง context สำเร็จ AProcessorContext จะ return ค่า Context ออกมา

Return Value

ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้

- IDI_R_OK : การทำงานเสร็จสมบูรณ์
- IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
- IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_CloseContext

IDWORKS Integrator Module: tmwitw.dll

ฟังก์ชัน idi_CloseContext ใช้สำหรับปิดการทำงานของ context

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE (__stdcall *idi_CloseContext)(
    IDI_CONTEXT AProcessorContext
);
```

Pascal-style function definition:

```
idi_CloseContext: function(
    AProcessorContext: IDI_CONTEXT
): IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_CloseContext Lib "TMWIDW.DLL" _
    ( _
        ByVal AProcessorContext As Long _
    ) As Long
```

Parameters

AProcessorContext
Context ที่ต้องการเลิกการใช้งาน

Return Value

- ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้
- IDI_R_OK : การทำงานเสร็จสมบูรณ์
 - IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
 - IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_ActivateSensor

IDWORKS Integrator Module: tmwitw.dll

ฟังก์ชันนี้ใช้สำหรับเริ่มการใช้งาน Sensor โดยการส่งค่า parameter ของ sensor ที่ต้องการใช้งานเข้าไป

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE (__stdcall *idi_ActivateSensor)(
    IDI_CONTEXT AProcessorContext,
    IDI_SENSORTYPE ASensorType,
    char *ASensorSerialNo,
    char *AAppSerial,
    char *AAppKey
);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โดยไม่ได้รับอนุญาตทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารฉบับนี้ด้วยทุกครั้งไปใช้

Pascal-style function definition:

```

idi_ActivateSensor: function(
    AProcessorContext: IDI_CONTEXT;
    ASensorType: IDI_SENSORTYPE;
    ASensorSerialNo: PChar;
    AAppSerial: PChar;
    AAppKey: PChar
): IDI_RCODE; stdcall;
    
```

Visual Basic-style function definition:

```

Public Declare Function idi_ActivateSensor Lib "TMWIDW.DLL" _
    ( _
        ByVal AProcessorContext As Long, _
        ByVal ASensorType As Long, _
        ByVal ASensorSerialNo As String, _
        ByVal AAppSerial As String, _
        ByVal AAppKey As String _
    ) As Long
    
```

Parameters

- AProcessorContext*
IDWORKS Processor Context
- ASensorType*
ประเภทของ Sensor
(สามารถดูได้จากค่าคงที่ที่เริ่มต้นด้วย IDI_ST_)
- ASensorSerialNo*
เลข Serial ของ Sensor (null-terminated)
- AAppSerial*
เลข IDWORKS Integrator Serial (null-terminated)
- AAppKey*
เลข IDWORKS Integrator Key (null-terminated)

Return Value

- ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้
- IDI_R_OK : การทำงานเสร็จสมบูรณ์
 - IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
 - IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_DeactivateSensor

IDWORKS Integrator Module: tmwitw.dll

ใช้สำหรับยกเลิกการใช้งานหมายเลขเครื่องอ่านลายนิ้วมือ ตามหมายเลขเครื่องอ่านลายนิ้วมือที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 15
 ไม่รู้นิใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารแหล่งที่มาทุกครั้งไปใช้

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE (__stdcall *idi_DeactivateSensor)(
    IDI_CONTEXT AProcessorContext,
    IDI_SENSORTYPE ASensorType,
    char* ASensorSerialNo
);
```

Pascal-style function definition:

```
idi_DeactivateSensor: function(
    AProcessorContext: IDI_CONTEXT;
    ASensorType: IDI_SENSORTYPE;
    ASensorSerialNo: PChar
): IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_DeactivateSensor Lib "TMWIDW.DLL" _
    ( _
        ByVal AProcessorContext As Long, _
        ByVal ASensorType As Long, _
        ByVal ASensorSerialNo As String _
    ) As Long
```

Parameters

- AProcessorContext*
IDWORKS Processor Context
- ASensorType*
ประเภทของ Sensor
(สามารถดูได้จากค่าคงที่ที่ขึ้นต้นด้วย IDI_ST_)
- ASensorSerialNo*
เลข Serial ของ Sensor (null-terminated)

Return Value

- ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้
- IDI_R_OK : การทำงานเสร็จสมบูรณ์
 - IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
 - IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_StartCapturing

IDWORKS Integrator Module: tmwitw.dll

เมื่อเรียกฟังก์ชันนี้ โปรแกรมจะเริ่มการทำงานของ sensor เพื่อรับลายนิ้วมือจากผู้ใช้ หากมีการแตะลายนิ้วมือ ลายนิ้วมือจะถูกประมวลผลโดยอัตโนมัติ ผลการประมวลผลของลายนิ้วมือ (เช่น "รู้จักลายนิ้วมือ" หรือ "ไม่รู้จัก") จะถูกส่งในรูปของ Message ผ่านไปทาง callback function ที่ได้ register ไว้โดยฟังก์ชัน `idi_RegisterIdentEventProc`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โดยไม่ได้รับอนุญาตทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงแหล่งเอกสารตลอดทั้งฉบับ

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE ( __stdcall *idi_StartCapturing) (
    IDI_CONTEXT AProcessorContext,
    char *ASensorSerialNo
);
```

Pascal-style function definition:

```
idi_StartCapturing : function(
    AProcessorContext: IDI_CONTEXT;
    ASensorSerialNo: PChar
): IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_StartCapturing Lib "TMWIDW.DLL" _
(
    ByVal AProcessorContext As Long,
    ByVal ASensorSerialNo As String
) As Long
```

Parameters

AProcessorContext
IDWORKS Processor Context

ASensorSerialNo
หมายเลข Sensor ที่ต้องการให้เริ่มการทำงาน
หมายเหตุ : Sensor ที่สามารถใช้ได้ จะต้องผ่านการ Activate โดยใช้
ฟังก์ชัน `idi_ActivateSensor` ก่อน

Return Value

ค่าที่ส่งกลับมาเป็น IDI_RCODE โดยมีความหมายดังนี้

- IDI_R_OK : การทำงานเสร็จสมบูรณ์
- IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
- IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_StopCapturing

IDWORKS Integrator Module: `tmwitw.dll`

ฟังก์ชันนี้จะหยุดการทำงานของ Sensor ตัวที่ระบุ

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE ( __stdcall *idi_StopCapturing) (
    IDI_CONTEXT AProcessorContext,
    char *ASensorSerialNo
);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โดยไม่ได้รับอนุญาตทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงแหล่งเอกสารตลอดทั้งนี้

Pascal-style function definition:

```
idi_StopCapturing: function(
    AProcessorContext: IDI_CONTEXT;
    ASensorSerialNo: PChar
): IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_StopCapturing Lib "TMWIDW.DLL" _
    ( _
        ByVal AProcessorContext As Long, _
        ByVal ASensorSerialNo As String _
    ) As Long
```

Parameters

AProcessorContext
IDWORKS Processor Context

ASensorSerialNo
หมายเลข Sensor ที่ต้องการหยุดการทำงาน

Return Value

ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้

- IDI_R_OK : การทำงานเสร็จสมบูรณ์
- IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
- IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_RegisterSensorEventProc

IDWORKS Integrator Module: tmwitw.dll

ทำการรีจิสเตอร์ callback function เพื่อรับ event ของเครื่องอ่านลายนิ้วมือ

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE ( __stdcall *idi_RegisterSensorEventProc) (
    IDI_CONTEXT AProcessorContext,
    IDI_SENSOREVENTPROC AEventProc,
    IDI_REFFPARAM ARefParam
);
```

Pascal-style function definition:

```
idi_RegisterSensorEventProc: function(
    AProcessorContext: IDI_CONTEXT;
    AEventProc: idi_SENSOREVENTPROC;
    ARefParam: idi_REFFPARAM
): IDI_RCODE; stdcall;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงแหล่งเอกสารทุกครั้ง ¹⁸
 IDWORKS Integrator SDK (DLLs) ใช้

Visual Basic-style function definition:

```
Public Declare Function idi_RegisterSensorEventProc Lib "TMWIDW.DLL" _
    ( _
        ByVal AProcessorContext As Long, _
        ByVal AEventProc As Long, _
        ByRef ARefParam As Any _
    ) As Long
```

Parameters

AProcessorContext

IDWORKS Processor Context

AEventProc

Pointer หรือ Address ของ Callback function ที่จะถูกเรียกเมื่อมี event เกี่ยวกับเครื่องอ่านลายนิ้วมือ

ARefParam

เป็นค่าที่จะถูกส่งกลับมากับ Message ใน IDI_SensorEventMsg.RefParam (มีประโยชน์เพื่อใช้อ้างอิง เช่น สามารถตั้งค่าเป็น Pointer ของ object ที่ต้องการได้)

Return Value

ค่าที่ส่งกลับมากจะเป็น IDI_RCODE โดยมีคความหมายดังนี้

- IDI_R_OK : การทำงานเสร็จสมบูรณ์
- IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
- IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_UnregisterSensorEventProc

IDWORKS Integrator Module: tmwitw.dll

ยกเลิกการรีจิสเตอร์ callback function ที่ระบุ

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE ( __stdcall *idi_UnregisterSensorEventProc) (
    IDI_CONTEXT AProcessorContext,
    IDI_SENSOREVENTPROC AEventProc
);
```

Pascal-style function definition:

```
idi_UnregisterSensorEventProc: function(
    AProcessorContext: IDI_CONTEXT;
    AEventProc: idi_SENSOREVENTPROC
): IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_UnregisterSensorEventProc Lib "TMWIDW.DLL" _
    ( _
```

ByVal AProcessorContext As Long

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงแหล่งเอกสารทุกครั้ง
 IDWORKS Integrator SDK (DLLs)

ByVal AEventProc As Long _

) As Long

Parameters

AProcessorContext

IDWORKS Processor Context

AEventProc

Callback function ที่ต้องการยกเลิกการรับ event

Return Value

ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้

- IDI_R_OK : การทำงานเสร็จสมบูรณ์
- IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
- IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_RegisterIdentEventProc

IDWORKS Integrator Module: tmwitw.dll

Register ฟังก์ชัน callback สำหรับรับ event การประมวลผลหลายนิ้วมือ

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE ( __stdcall *idi_RegisterIdentEventProc) (
    IDI_CONTEXT AProcessorContext,
    IDI_IDENTEVENTPROC AEventProc,
    IDI_REFPARAM ARefParam
);
```

Pascal-style function definition:

```
idi_RegisterIdentEventProc: function(
    AProcessorContext: IDI_CONTEXT;
    AEventProc: idi_IdentEVENTPROC;
    ARefParam: idi_REFPARAM
): IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_RegisterIdentEventProc Lib "TMWIDW.DLL" _
    ( _
        ByVal AProcessorContext As Long, _
        ByVal AEventProc As Long, _
        ByVal ARefParam As Long _
    ) As Long
```

Parameters

AProcessorContext

IDWORKS Processor Context

AEventProc

Callback function ที่จะถูกเรียกเมื่อมีการประมวลผลลายนี้อำมือ

ARefParam

เป็นค่าที่จะถูกส่งกลับมากับ Message ใน IDI_IdentEventMsg.RefParam (มีประโยชน์เพื่อใช้อ้างอิง เช่น สามารถตั้งค่าเป็น Pointer ของ object ที่ต้องการได้)

Return Value

ค่าที่ส่งกลับมากจะเป็น IDI_RCODE โดยมีความหมายดังนี้

- IDI_R_OK : การทำงานเสร็จสมบูรณ์
- IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
- IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_UnregisterIdentEventProc

IDWORKS Integrator Module: tmwitw.dll

ยกเลิกการ Register ที่ได้ทำไปแล้วจากการเรียกฟังก์ชัน `idi_RegisterIdentEventProc`

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE (__stdcall *idi_UnregisterIdentEventProc) (
    IDI_CONTEXT AProcessorContext,
    IDI_IDENTEVENTPROC AEventProc
);
```

Pascal-style function definition:

```
idi_UnregisterIdentEventProc: function(
    AProcessorContext: IDI_CONTEXT;
    AEventProc: idi_IdentEVENTPROC
): IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_UnregisterIdentEventProc Lib "TMWIDW.DLL" _
    ( _
        ByVal AProcessorContext As Long, _
        ByVal AEventProc As Long _
    ) As Long
```

Parameters

AProcessorContext

IDWORKS Processor Context

AEventProc

Callback function ที่ต้องการยกเลิกการ register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีนำไปใช้

Return Value

- ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้
- IDI_R_OK : การทำงานเสร็จสมบูรณ์
 - IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
 - IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_fp_Add

IDWORKS Integrator Module: tmwitw.dll

เพิ่มलयนิ้วมือเข้าไปในฐานข้อมูลชั่วคราวภายในของโมดูลलयนิ้วมือที่เพิ่มเข้าไปจะทำให้โมดูลรู้จักलयนิ้วมือนั้นเมื่อมีการประมวลผลलयนิ้วมือ

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE (__stdcall *idi_fp_Add)(
    IDI_CONTEXT AProcessorContext,
    FP_FPINFO *AFPInfoPtr
);
```

Pascal-style function definition:

```
idi_fp_Add: function(
    AProcessorContext: IDI_CONTEXT;
    AFPInfoPtr: FP_FPInfoPtr
): IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_fp_Add Lib "TMWIDW.DLL" (
    _
    ByVal AProcessContext As Long, _
    ByRef AFPInfo As FP_FPInfo _
) As Long
```

Parameters

AProcessorContext
IDWORKS Processor Context ;

AFPInfoPtr
Pointer ที่ชี้ไปยังข้อมูลलयนิ้วมือ (FP_FPInfo) ที่ต้องการเพิ่มเข้าไปในฐานข้อมูลภายใน

Return Value

- ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้
- IDI_R_OK : การทำงานเสร็จสมบูรณ์
 - IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
 - IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

idi_fp_ClearAll

IDWORKS Integrator Module: tmwitw.dll

เป็นการล้างฐานข้อมูลภายในของโมดูล

Function Definition

C/C++-style function definition:

```
typedef IDI_RCODE ( __stdcall *idi_fp_ClearAll) (
    IDI_CONTEXT processorContext
);
```

Pascal-style function definition:

```
idi_fp_ClearAll : function(
    AProcessorContext: IDI_CONTEXT
) : IDI_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function idi_fp_ClearAll Lib "TMWIDW.DLL" _
    ( _
        ByVal AProcessorContext As Long _
    ) As Long
```

Parameters

AProcessorContext
IDWORKS Processor Context

Return Value

ค่าที่ส่งกลับมาจะเป็น IDI_RCODE โดยมีความหมายดังนี้

- IDI_R_OK : การทำงานเสร็จสมบูรณ์
- IDI_R_MODULENOTINITED : ยังไม่มีการ Initialize Module
- IDI_R_INVALIDCONTEXT : ค่า Context ไม่ถูกต้อง

ประเภทข้อมูล (Data Types)

IDI_RCODE

IDI_RCODE เป็นประเภทข้อมูล 32 บิต ที่เป็นค่าที่ส่งกลับมาเมื่อเรียกฟังก์ชัน เพื่อบอกถึงผลลัพธ์ของฟังก์ชันนั้นๆ

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long IDI_RCODE;
```

Pascal-style data type definition:

```
IDI_RCODE = Integer;
```

Visual Basic-style data type definition:

ใช้ประเภท long

IDI_CONTEXT

IDI_CONTEXT เป็นประเภทข้อมูล 32-bit สำหรับ Processor Context ที่เสมือนกับเป็นหน่วยในการประมวลผลข้อมูล ซึ่งสามารถสร้างได้หลาย context แยกกัน ในการใช้งานแต่ละครั้ง

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long IDI_CONTEXT;
```

Pascal-style data type definition:

```
IDI_CONTEXT = Longword;
```

Visual Basic-style data type definition:

ใช้ประเภท Long

IDI_IDENTEVENTMSG

IDI_IDENTEVENTMSG เป็นประเภทข้อมูลแบบ structure ที่จะถูกส่งกลับมาให้กับ callback function ที่ได้ทำการ register ไว้โดยฟังก์ชัน `idi_RegisterIdentEventProc`

Data-type Definition

C/C++-style data type definition:

```
struct IDI_IDENTEVENTMSG {
    IDI_CONTEXT ProcessorContext;
    IDI_IDENTEVENTTYPE MsgID;
    IDI_IDENTEVENTPARAM MsgParam;
    IDI_REFPARAM RefParam;
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มี **24**
 IDWORKS Integrator SDK (DLLs)

Pascal-style data type definition:

```
IDI_IDENTEVENTMSG = record
    ProcessorContext : IDI_CONTEXT;
    MsgID : IDI_IDENTEVENTTYPE;
    MsgParam : IDI_IDENTEVENTPARAM;
    RefParam : IDI_REFPARAM;
end;
IDI_IDENTEVENTMSG_PTR = ^IDI_IDENTEVENTMSG;
```

Visual Basic-style data type definition:

```
Public Type IDI_IDENTEVENTMSG
    ProcessorContext As Long
    MsgID As Long
    MsgParam As Long
    RefParam As Long
End Type
```

หมายเหตุ

ค่าที่ถูส่งมาใน MsgParam จะขึ้นกับค่า MsgID ซึ่งจะมีคความหมายดังตารางต่อไปนี้

ค่า MsgID	คความหมายของค่าใน MsgParam
IDI_ID_FINGERPRINTMATCHED	Pointer ที่ชี้ไปยัง IDI_IDENTEVENTMATCHSTRUCT
IDI_ID_FINGERPRINTNOTMATCHED	0

IDI_IDENTEVENTTYPE

IDI_IDENTEVENTTYPE เป็นประเภทข้อมูลขนาด 32-bit สำหรับเก็บชนิดของ Identification Message ที่จะถูกส่งกลับมายังฟังก์ชันที่ได้ทำการ register ไว้โดยใช้ฟังก์ชัน RegisterIdentEventProc ซึ่งประเภทข้อมูลนี้จะใช้ใน IDI_IDENTEVENTMSG.MsgID

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long IDI_IDENTEVENTTYPE;
```

Pascal-style data type definition:

```
IDI_IDENTEVENTTYPE = Longword;
```

Visual Basic-style data type definition:

```
ใช้ประเภท Long
```

IDI_IDENTEVENTPARAM

IDI_IDENTEVENTPARAM เป็นประเภทข้อมูลขนาด 32-bit ที่จะถูกส่งมากับ Identification Message IDI_IDENTEVENTMSG ซึ่งความหมายจะแตกต่างกันสำหรับแต่ละ message ขึ้นอยู่กับค่า IDI_IDENTEVENTMSG.MsgID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในลักษณะใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีก้นำไปใช้

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long IDI_IDENTEVENTPARAM;
```

Pascal-style data type definition:

```
IDI_IDENTEVENTPARAM = Longword;
```

Visual Basic-style data type definition:

ใช้ประเภท Long

IDI_REFPARAM

IDI_REFPARAM เป็นข้อมูล Pointer ที่ไว้ใช้อ้างอิงกับการเรียกใช้ฟังก์ชัน `idi_RegisterIdentEventProc` ได้ โดยค่าที่ส่งกลับมาจะเป็นค่าเดียวกันกับค่า `ARefParam` ที่ส่งเข้าไปเมื่อเรียกฟังก์ชัน `idi_RegisterIdentEventProc`

Data-type Definition

C/C++-style data type definition:

```
typedef void *IDI_REFPARAM;
```

Pascal-style data type definition:

```
IDI_REFPARAM = Pointer;
```

Visual Basic-style data type definition:

ใช้ประเภท Long

IDI_IDENTEVENTPROC

IDI_IDENTEVENTPROC เป็นการประกาศประเภทของฟังก์ชัน ที่ใช้สำหรับการ callback เมื่อมีการแตะนิ้วมือ โดย pointer ของฟังก์ชันที่ใส่ส่งเข้าไปเป็นพารามิเตอร์ `AEventProc` ของ `idi_RegisterIdentEventProc` จะต้องมี arguments เหมือนกับที่ประกาศใน IDI_IDENTEVENTPROC

Data-type Definition

C/C++-style data type definition:

```
typedef void (__stdcall *IDI_IDENTEVENTPROC) (
    IDI_IDENTEVENTMSG *AIdentEventMsgPtr
);
```

Pascal-style data type definition:

```
IDI_IDENTEVENTPROC = procedure (
    AIdentEventMsgPtr : IDI_IDENTEVENTMSG_PTR
);stdcall;
```

Visual Basic-style data type definition:

```
Public Sub IdentEventProc(
    ByRef audtIdentEventMsg As IDI_IDENTEVENTMSG
)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีนำไปใช้



Parameters

IdentEventMsgPtr

Pointer ที่ชี้ไปยัง IDI_IDENTEVENTMSG ซึ่งบอกรายละเอียดของ Message

IDI_IDENTEVENTMATCHEDSTRUCT

ในกรณีที่มีการแตะนิ้วมือและพบข้อมูล IDI_IDENTEVENTMATCHEDSTRUCT จะเป็นข้อมูลที่ถูกส่งกลับมากับ IDI_IDENTEVENTMSG ซึ่ง IDI_IDENTEVENTMSG.MsgParam จะเป็น pointer ที่ชี้ไปยังข้อมูลดังกล่าว

Data-type Definition

C/C++-style data type definition:

```
struct IDI_IDENTEVENTMATCHEDSTRUCT {
    FP_FPINFO *FPInfoPtr;
    int MatchingScore;
};
```

Pascal-style data type definition:

```
IDI_IDENTEVENTMATCHEDSTRUCT = record
    FPInfoPtr: FP_FPINFO_PTR;
    MatchingScore : Integer;
end;
IDI_IDENTEVENTMATCHEDSTRUCT_PTR = ^IDI_IDENTEVENTMATCHEDSTRUCT;
```

Visual Basic-style data type definition:

```
Public Type IDI_IDENTEVENTMATCHEDSTRUCT
    FPInfoPtr As Long
    MatchingScore As Long
End Type
```

Members

FPInfoPtr

เป็น Pointer ที่ชี้ไปยังข้อมูล FP_FPINFO ที่บอกรายละเอียดของลายนิ้วมือ

MatchingScore

เป็นค่าคะแนนความเหมือนกันของลายนิ้วมือ

IDI_SENSOREVENTMSG

IDI_SENSOREVENTMSG เป็นประเภทข้อมูลแบบ structure ที่จะถูกส่งกลับมาให้กับ callback function ที่ได้ทำการ register ไว้โดยฟังก์ชัน `idi_RegisterSensorEventProc`

Data-type Definition

C/C++-style data type definition:

```
struct IDI_SENSOREVENTMSG {
    IDI_CONTEXT ProcessorContext;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โดยไม่ได้รับอนุญาต ทั้งนี้ ห้ามนำไปใช้เพื่อวัตถุประสงค์อื่นใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีนำไปใช้

```

IDI_SENSOREVENTTYPE MsgID;
IDI_SENSOREVENTPARAM MsgParam;
IDI_REFPARAM RefParam;
};

```

Pascal-style data type definition:

```

IDI_SENSOREVENTMSG = record
    ProcessorContext : IDI_CONTEXT;
    MsgID : IDI_SENSOREVENTTYPE;
    MsgParam : IDI_SENSOREVENTPARAM;
    RefParam : IDI_REFPARAM;
end;

```

```

IDI_SENSOREVENTMSG_PTR = ^IDI_SENSOREVENTMSG;

```

Visual Basic-style data type definition:

```

Public Type IDI_SENSOREVENTMSG
    ProcessorContext As Long
    MsgID As Long
    MsgParam As Long
    RefParam As Long
End Type

```

Members

- ProcessorContext*
Context ที่เป็นผู้ส่ง message มาให้
- MsgID*
ประเภทของ Message
- MsgParam*
ข้อมูลของ Message นั้นๆ
- RefParam*
เลขอ้างอิงที่ถูกส่งเข้ามากับการเรียก `idi_RegisterSensorEventProc` ก่อนหน้านี้

หมายเหตุ

ค่าที่ถูกส่งมาใน `MsgParam` จะขึ้นกับค่า `MsgID` ซึ่งจะมีความหมายดังตารางต่อไปนี้

ค่า <code>MsgID</code>	ความหมายของค่าใน <code>MsgParam</code>
<code>IDI_SE_IMAGEACQUIRED</code>	Pointer ที่ชี้ไปยัง <code>IDI_SENSORIMAGEACQUIREDSTRUCT</code>
<code>IDI_SE_IMAGEPROCESSED</code>	Pointer ที่ชี้ไปยัง <code>IDI_SENSORIMAGEACQUIREDSTRUCT</code>

IDI_SENSOREVENTPROC

`IDI_SENSOREVENTPROC` เป็นการประกาศประเภทของฟังก์ชัน ที่ใช้สำหรับการ callback เมื่อมี event เกี่ยวกับเครื่องอ่านลายนิ้วมือ เช่น ได้รับรูปลายนิ้วมือ ประเภทฟังก์ชันนี้จะใช้กับฟังก์ชัน `IDI_RegisterSensorEventProc` โดย pointer หรือ address ของฟังก์ชันที่ใช้ส่งเข้าไปเป็นพารามิเตอร์ `AEventProc` จะต้องมี arguments เหมือนกับที่ประกาศใน `IDI_SENSOREVENTPROC`

Data-type Definition

C/C++-style data type definition:

```
typedef void __stdcall (*IDI_SENSOREVENTPROC) (
    IDI_SENSOREVENTMSG *ASensorEventMsgPtr
);
```

Pascal-style data type definition:

```
IDI_SENSOREVENTPROC = procedure (
    ASensorEventMsgPtr: IDI_SENSOREVENTMSG_PTR;
); stdcall;
```

Visual Basic-style data type definition:

```
Public Sub SensorEventProc(ByRef ASensorEventMsg As IDI_SENSOREVENTMSG)
```

Parameters

ASensorEventMsgPtr

Pointer ที่ชี้ไปยัง IDI_SENSOREVENTMSG ซึ่งบอกรายละเอียดของ Message

IDI_SENSOREVENTIMAGEACQUIREDSTRUCT

เมื่อมีการแตะลายนิ้วมือ และเครื่องอ่านลายนิ้วมืออ่านรูปลายนิ้วมือสำเร็จ โมดูลจะส่ง event ให้กับ function ที่ได้ทำการรีจิสเตอร์ไว้โดย `idi_RegisterSensorEvent` ซึ่งใน `MsgParam` ของ event ดังกล่าว จะเป็นค่า pointer ของ `IDI_SENSOREVENTIMAGEACQUIREDSTRUCT`

Data-type Definition

C/C++-style data type definition:

```
struct IDI_SENSOREVENTIMAGEACQUIREDSTRUCT {
    int ImageWidth;
    int ImageHeight;
    void* ImageBitmapPtr;
};
```

Pascal-style data type definition:

```
IDI_SENSOREVENTIMAGEACQUIREDSTRUCT = record
    ImageWidth : Integer;
    ImageHeight : Integer;
    ImageBitmapPtr : Pointer;
end;
```

```
IDI_SENSOREVENTIMAGEACQUIREDSTRUCT_PTR = ^IDI_SENSOREVENTIMAGEACQUIREDSTRUCT;
```

Visual Basic-style data type definition:

```
Public Type IDI_SENSOREVENTIMAGEACQUIREDSTRUCT
    ImageWidth As Long
    ImageHeight As Long
    ImageBitmapPtr As Long
End Type
```

Members

ImageWidth

ความกว้างของรูปลายนิ้วมือ

ImageHeight

ความสูงของรูปลายนิ้วมือ

เอกสารนี้เป็นเอกสารของบริษัท IDWORKS Integrator สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โดยไม่ได้รับอนุญาตจาก IDWORKS Integrator ทุกครั้งที่มีกฏนำไปใช้

ImageBitmapPtr

Pointer ที่ชี้ไปยังข้อมูล grayscale bitmap ของลายนิ้วมือ โดยมีขนาดเท่ากับ ImageWidth x ImageHeight และมีรูปแบบเป็น 8-bit grayscale

FP_FPINFO

FP_FPINFO เป็นประเภทข้อมูลแบบ structure ที่ใช้สำหรับส่งผ่านข้อมูลลายนิ้วมือเข้าไปยังฐานข้อมูลชั่วคราวใน Processor Context

Data-type Definition

C/C++-style data type definition:

```
struct FP_FPINFO {
    int FingerprintIndexingValue;
    int FingerIndex;
    void *FingerprintRawDataPtr;
    unsigned long FingerprintRawDataSize;
    int Tag;
}
```

Pascal-style data type definition:

```
FP_FPINFO = record
    FingerprintIndexingValue : Integer;
    FingerIndex : Integer;
    FingerprintRawDataPtr: PByte;
    FingerprintRawDataSize: Longword;
    Tag : Integer;
end;
FP_FPINFO_PTR = ^FP_FPINFO;
```

Visual Basic-style data type definition:

```
Public Type FP_FPINFO
    FingerprintIndexingValue As Long
    FingerIndex As Long
    FingerprintRawDataPtr As Long
    FingerprintRawDataSize As Long
    Tag As Long
End Type
```

Members

FingerprintIndexingValue

เป็นค่า Indexing ของลายนิ้วมือ ที่ใช้สำหรับ optimize การค้นหาลายนิ้วมือ

FingerIndex

เป็นหมายเลขขอกนิ้ว ซึ่งจะไล่ตั้งแต่ 0 คือนิ้วก้อยมือซ้าย ถึง 9 คือนิ้วก้อยมือขวา

FingerprintRawDataPtr

เป็น Pointer ที่ชี้ไปยัง memory buffer ที่เก็บข้อมูลลายนิ้วมือ

FingerprintRawDataSize

เป็นขนาดของ memory buffer ที่ชี้โดย FingerprintRawDataPtr

Tag

เป็นเลขกำกับลายนิ้วมือ ซึ่งผู้พัฒนาสามารถใช้อ้างอิงได้ เช่น สามารถเก็บเลข ID ของบุคคลในฐานข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงข้อมูลเอกสารทุกครั้งที่มีนำไปใช้

ค่าคงที่ (Constants)

ค่าคงที่สำหรับ IDI_RCODE

ชื่อค่าคงที่	ค่า	ความหมาย
IDI_R_OK	0	การทำงานเสร็จสิ้น ไม่มีข้อผิดพลาด
IDI_R_GENERALFAILURE	-1	มีข้อผิดพลาดทั่วไป
IDI_R_INIT_FAILED	-2	ไม่สามารถเริ่มการใช้งานได้
IDI_R_MODULENOTINITED	-3	เรียกฟังก์ชันโดยไม่มี การ Initialize Module ก่อน
IDI_R_SENSORNOTFOUND	-4	ไม่พบเครื่องอ่านลายนิ้วมือตามที่กำหนด
IDI_R_SENSORALREADYACTIVATED	-5	เครื่องอ่านลายนิ้วมือ ได้ถูก Activate ไปก่อนหน้านั้นแล้ว
IDI_R_INVALIDSENSORTYPE	-6	ประเภทเครื่องอ่านลายนิ้วมือ ไม่ถูกต้อง
IDI_R_INVALIDREGISTRATION	-7	การ Activate ไม่ถูกต้อง
IDI_R_INVALIDCONTEXT	-16	Context ไม่ถูกต้อง

ค่าคงที่สำหรับ IDI_IDENTEVENTTYPE

ชื่อค่าคงที่	ค่า	ความหมาย
IDI_ID_FINGERPRINTMATCHED	0	พบลายนิ้วมือ
IDI_ID_FINGERPRINTNOTMATCHED	1	ไม่พบลายนิ้วมือ

ค่าคงที่สำหรับ IDI_SENSORTYPE

ชื่อค่าคงที่	ค่า	ความหมาย
IDI_ST_UAREU	1	เครื่องอ่านลายนิ้วมือ DigitalPersona U.are.U 4000

ค่าคงที่สำหรับ IDI_SENSOREVENTTYPE

ชื่อค่าคงที่	ค่า	ความหมาย
IDI_SE_IMAGEACQUIRED	1	Event รับรูปจากเครื่องอ่านลายนิ้วมือ
IDI_SE_IMAGEPROCESSED	2	Event ประมวลผลรูปลายนิ้วมือเสร็จสิ้น

ID-WORKS Integrator – ส่วนข้อมูลเครื่องอ่านลายนิ้วมือ

DLL Name: ITWSSL.DLL

เป็นไลบรารีสำหรับดึงข้อมูลของเครื่องอ่านลายนิ้วมือที่ต่ออยู่ทั้งหมด

รายละเอียดฟังก์ชัน (Function Definitions)

isl_IsModuleInited

IDWORKS Integrator Module: ITWSSL.DLL

ตรวจสอบว่า module ได้ถูก initialize แล้วหรือไม่

Function Definition

C/C++-style function definition:

```
typedef ISL_RCODE (__stdcall *isl_IsModuleInited) (void);
```

Pascal-style function definition:

```
isl_IsModuleInited: function : ISL_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function isl_IsModuleInited Lib "ITWSSL.DLL" () As Long
```

Parameters

ไม่มี parameters

Return Value

- ISL_R_OK : โมดูลได้ initialize แล้ว
- ISL_R_MODULENOTINITED : โมดูลยังไม่ได้ initialize

isl_InitModule

IDWORKS Integrator Module: ITWSSL.DLL

เริ่มการทำงานของโมดูล

Function Definition

C/C++-style function definition:

```
typedef ISL_RCODE (__stdcall *isl_InitModule) (void);
```

Pascal-style function definition:

```
isl_InitModule: function : ISL_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function isl_InitModule Lib "ITWSSL.DLL" () As Long
```

Parameters

ไม่มี parameters

Return Value

- ISL_R_OK : การเริ่มต้น
- ISL_R_MODULENOTINITED : ไม่สามารถเริ่มต้นโมดูลได้

isl_FinalizeModule

IDWORKS Integrator Module: ITWSSL.DLL

เลิกการทำงานของโมดูล เมื่อเลิกการใช้งานโมดูล (เช่น ในช่วงก่อนปิดโปรแกรม)จะต้องเรียกฟังก์ชันนี้เพื่อเคลียร์ข้อมูลต่างๆที่อาจค้างอยู่ในโมดูล

Function Definition

C/C++-style function definition:

```
typedef ISL_RCODE (__stdcall *isl_FinalizeModule) (void);
```

Pascal-style function definition:

```
isl_FinalizeModule: function : ISL_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function isl_FinalizeModule Lib "ITWSSL.DLL" () As Long
```

Parameters

ไม่มี parameters

Return Value

- ISL_R_OK : สำเร็จ
- ISL_R_MODULENOTINITED : ไม่สามารถเริ่มต้นโมดูลได้

isl_GetDeviceCount

IDWORKS Integrator Module: ITWSSL.DLL

ใช้สำหรับนับจำนวน sensor ทั้งหมดที่ต่ออยู่

Function Definition

C/C++-style function definition:

```
typedef ISL_RCODE (__stdcall *isl_GetDeviceCount) (
    unsigned int *ASensorCount
);
```

Pascal-style function definition:

```
isl_GetDeviceCount: function (
    var ASensorCount : Cardinal
) : ISL_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function isl_GetDeviceCount Lib "ITWSSL.DLL" _
    ( _
        ByRef ASensorCount As Long _
    ) As Long
```

Parameters

ASensorCount
เป็นจำนวน sensor ที่ส่งกลับมา

Return Value

- ISL_R_OK : สำเร็จ จำนวน sensor จะถูกใส่เข้ามาใน ASensorCount
- ISL_R_MODULENOTINITED : ฟังก์ชัน isl_InitModule ยังไม่ถูกเรียก

isl_GetDeviceList

IDWORKS Integrator Module: ITWSSL.DLL

ใช้สำหรับดึงข้อมูลของ sensor ทั้งหมดที่ต่ออยู่

Function Definition

C/C++-style function definition:

```
typedef ISL_RCODE ( __stdcall *isl_GetDeviceList)(
    ISL_SENSORINFO *ASensorList,
    unsigned int ASensorCount
);
```

Pascal-style function definition:

```
isl_GetDeviceList: function(
    ASensorList : Pointer;
    ASensorCount : Cardinal
) : ISL_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function isl_GetDeviceList Lib "ITWSSL.DLL" _
    ( _
        ByRef ASensorList As Any, _
        ByVal ASensorCount As Long _
    ) As Long
```

Parameters

ASensorList
เป็น pointer ของ array (แบบ C/C++) ของ FP_FPInfo ซึ่งต้องมีความยาวอย่างน้อยเท่ากับ ASensorCount

ASensorCount
คือจำนวน sensor ที่ต้องการดึงข้อมูล สามารถหาได้จากฟังก์ชัน isl_GetDeviceCount

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีโอกาสนำไปใช้

Return Value

- ISL_R_OK : สำเร็จ จำนวน sensor จะถูกใส่ค่ามาใน ASensorCount
- ISL_R_MODULENOTINITED : ฟังก์ชัน isl_InitModule ยังไม่ถูกเรียก



ประเภทข้อมูล (Data Types)

ISL_RCODE

ISL_RCODE เป็นประเภทข้อมูล 32 บิต ที่เป็นค่าที่ส่งกลับมาเมื่อเรียกฟังก์ชัน เพื่อบอกถึงผลลัพธ์ของฟังก์ชันนั้นๆ

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long ISL_RCODE;
```

Pascal-style data type definition:

```
ISL_RCODE = Integer;
```

Visual Basic-style data type definition:

ใช้ประเภท Long

ISL_DEVICETYPE

ISL_DEVICETYPE เป็นประเภทข้อมูล 32 บิตที่ใช้สำหรับระบุประเภทของเครื่องอ่านลายนิ้วมือ

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long ISL_DEVICETYPE;
```

Pascal-style data type definition:

```
ISL_DEVICETYPE = Longword;
```

Visual Basic-style data type definition:

ใช้ประเภท Long

ISL_DEVICEID

ISL_DEVICEID เป็นประเภทข้อมูล 64 บิต ที่ใช้เป็นเลขประจำตัวของเครื่องอ่านลายนิ้วมือ

Data-type Definition

C/C++-style data type definition:

```
typedef __int64 ISL_DEVICEID;
```

Pascal-style data type definition:

```
ISL_DEVICEID = int64;
```

Visual Basic-style data type definition:

ใน Visual Basic จะไม่มี Integer ขนาด 64 bit แต่ใน header file ของ Visual Basic ที่มาพร้อมกับ SDK จะแก้ปัญหาใน structure ที่ใช้ประเภทข้อมูลแบบนี้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โดยไม่ได้รับอนุญาตทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงที่มาของเอกสารตลอดทั้งฉบับ ³⁷ **IDWORKS Integrator SDK (DLLs)**

ISL_CHAR

ISL_CHAR เป็นประเภทข้อมูล 8 บิต ที่ใช้เก็บตัวอักษร จะใช้เก็บข้อความทั่วไปในโมดูล

Data-type Definition

C/C++-style data type definition:

```
typedef char ISL_CHAR;
```

Pascal-style data type definition:

```
ISL_CHAR = Char;
```

Visual Basic-style data type definition:

ใช้ประเภท Byte

ISL_SENSORINFO

ISL_SENSORINFO เป็นประเภทข้อมูลแบบ structure ที่ใช้เก็บข้อมูลเกี่ยวกับรายละเอียดต่างๆ ของเครื่องอ่านลายนิ้วมือ

Data-type Definition

C/C++-style data type definition:

```
struct ISL_SENSORINFO {
    ISL_DEVICETYPE DeviceType;
    ISL_DEVICEID DeviceId;
    ISL_CHAR DeviceName[ISL_MAXDEVICENAMELEN];
    ISL_CHAR DeviceSerial[ISL_MAXDEVICESERIALLEN];
}
```

Pascal-style data type definition:

```
ISL_SENSORINFO = record
    DeviceType: ISL_DEVICETYPE;
    DeviceId: ISL_DEVICEID;
    DeviceName: Array[0..ISL_MAXDEVICENAMELEN-1] of ISL_CHAR;
    DeviceSerial: Array[0..ISL_MAXDEVICESERIALLEN-1] of ISL_CHAR;
```

```
end;
```

```
ISL_SENSORINFO_PTR = ^ISL_SENSORINFO
```

Visual Basic-style data type definition:

```
Public Type ISL_SENSORINFO
    DeviceType As Long
    DeviceId As Long
    DeviceName(ISL_MAXDEVICENAMELEN) As Byte
    DeviceSerial(ISL_MAXDEVICESERIALLEN) As Byte
End Type
```

Members

DeviceType

ประเภทของเครื่องอ่านลายนิ้วมือ

DeviceID

หมายเลขประจำเครื่องอ่านลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารที่ปรากฏในเอกสารนี้

DeviceName

ชื่อเครื่องอ่านลายนิ้วมือ เช่น DigitalPersona U.are.U 4000

DeviceSerial

S/N ประจำเครื่องอ่านลายนิ้วมือ



ค่าคงที่ (Constants)

ค่าคงที่ทั่วไป

ชื่อค่าคงที่	ค่า	ความหมาย
ISL_MAXDEVICENAMELEN	64	เรียกฟังก์ชันโดยไม่มี Initialize Module ก่อน
ISL_MAXDEVICSERIALLEN	64	ไม่พบเครื่องอ่านลายนิ้วมือตามที่กำหนด

ค่าคงที่สำหรับ ISL_RCODE

ชื่อค่าคงที่	ค่า	ความหมาย
ISL_R_OK	0	การทำงานเสร็จสิ้น ไม่มีข้อผิดพลาด
ISL_R_MODULENOTINITED	-1	มีข้อผิดพลาดทั่วไป
ISL_R_ALREADYINITED	-2	ไม่สามารถเริ่มการใช้งานได้

ค่าคงที่สำหรับ ISL_DEVICETYPE

ชื่อค่าคงที่	ค่า	ความหมาย
ISL_DT_UAREU	0	การทำงานเสร็จสิ้น ไม่มีข้อผิดพลาด

ID-WORKS Integrator – ส่วนช่วยการบันทึกถายนิ้วมือ

DLL Name: FPENROL.DLL

- ทุกฟังก์ชันจะ return ค่าเป็น FPE_R_OK หากทำงานได้สำเร็จ ยกเว้นหากได้หมายเหตุไว้ว่าเป็นอย่างอื่น

รายละเอียดฟังก์ชัน (Function Definitions)

fpe_InitModule

IDWORKS Integrator Module: FPENROL.DLL

เริ่มการใช้งานไลบรารี

Function Definition

C/C++-style function definition:

```
typedef FPE_RCODE ( __stdcall *fpe_InitModule) (void);
```

Pascal-style function definition:

```
fpe_InitModule: function : FPE_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function fpe_InitModule Lib "FPENROL.DLL" () As Long
```

Parameters

ไม่มี Parameter

Return Value

- FPE_R_OK : การทำงานสำเร็จ
- FPE_R_MODULENOTINITED : ไม่สามารถเริ่มการทำงานได้

fpe_FinalizeModule

IDWORKS Integrator Module: FPENROL.DLL

เรียกเมื่อเลิกใช้งานไลบรารี

Function Definition

C/C++-style function definition:

```
typedef FPE_RCODE ( __stdcall *fpe_FinalizeModule) (void);
```

Pascal-style function definition:

```
fpe_FinalizeModule: function : FPE_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function fpe_FinalizeModule Lib "FPENROL.DLL" () As Long
```

Parameters

ไม่มี Parameter

Return Value

- FPE_R_OK : การทำงานสำเร็จ
- FPE_R_MODULENOTINITED : ไม่ได้มีการ Initialize Module ก่อนหน้านี้

fpe_CreateContext

IDWORKS Integrator Module: FPENROL.DLL

สร้าง context สำหรับ fingerprint enrolment

Function Definition

C/C++-style function definition:

```
typedef FPE_RCODE (__stdcall *LP_FUNC_FPE_CREATE_CONTEXT) (
    FPE_CONTEXT* AEnrolmentContext
);
```

Pascal-style function definition:

```
fpe_CreateContext: function(
    var AEnrolmentContext: FPE_CONTEXT
): FPE_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function fpe_CreateContext Lib "FPENROL.DLL" (
    _
    ByRef AEnrolmentContext As Long _
) As Long
```

Parameters

AEnrolmentContext

ถ้าการสร้าง Context สำเร็จ ค่าดังกล่าวจะเป็น Context สำหรับ Enrolment

Return Value

- FPE_R_OK : การทำงานสำเร็จ
- FPE_R_MODULENOTINITED : ไม่ได้มีการ Initialize Module ก่อนหน้านี้

fpe_CloseContext

IDWORKS Integrator Module: FPENROL.DLL

ปิด context ของ fingerprint enrolment

Function Definition

C/C++-style function definition:

```
typedef FPE_RCODE (__stdcall *fpe_CloseContext) (
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงด้วยเอกสารทุกครั้งที่มีโอกาสไปใช้

```
FPE_CONTEXT AEnrolmentContext
```

```
);
```

Pascal-style function definition:

```
fpe_CloseContext: function(  
    AEnrolmentContext: FPE_CONTEXT  
): FPE_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function fpe_CloseContext Lib "FPENROL.DLL" _  
    (  
        ByVal AEnrolmentContext As Long _  
    ) As Long
```

Parameters

AEnrolmentContext

หากการสร้าง context สำเร็จ AEnrolmentContext จะถูกตั้งค่าเป็นค่า context ที่ใช้ในการลงทะเบียนลายนิ้วมือ

Return Value

- FPE_R_OK : การทำงานสำเร็จ
- FPE_R_MODULENOTINITED : ไม่ได้มีการ Initialize Module ก่อนหน้านี้

fpe_ActivateSensor

IDWORKS Integrator Module: FPENROL.DLL

เริ่มการใช้งาน sensor ที่ต้องการ

Function Definition

C/C++-style function definition:

```
typedef FPE_RCODE (__stdcall *fpe_ActivateSensor)(  
    FPE_CONTEXT AEnrolmentContext,  
    FPE_SENSORTYPE ASensorType,  
    char *ASensorSerialNo,  
    char *AAppSerial,  
    char *AAppKey  
);
```

Pascal-style function definition:

```
fpe_ActivateSensor: function(  
    AEnrolmentContext: FPE_CONTEXT;  
    ASensorType: FPE_SENSORTYPE;  
    ASensorSerialNo : PChar;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โดยไม่ได้อำนาจใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีนำไปใช้

```

AAppSerial : PChar; AAppKey : PChar
) : FPE_RCODE; stdcall;

```

Visual Basic-style function definition:

```

Public Declare Function fpe_ActivateSensor Lib "FPENROL.DLL" _
(
    ByVal AEnrolmentContext As Long, _
    ByVal ASensorType As Long, _
    ByVal ASensorSerialNo As String, _
    ByVal AAppSerial As String, _
    ByVal AAppKey As String _
) As Long

```

Parameters

AProcessorContext
Context ที่ใช้ในการลงทะเบียนลายนิ้วมือ

ASensorType
ประเภทของ Sensor (สามารถดูได้จากค่าคงที่ที่ขึ้นต้นด้วย FPE_ST_)

ASensorSerialNo
เลข Serial ของ Sensor (null-terminated)

AAppSerial
เลข IDWORKS Integrator Serial (null-terminated)

AAppKey
เลข IDWORKS Integrator Key (null-terminated)

Return Value

- FPE_R_OK : การทำงานสำเร็จ และ Activate เครื่องอ่านลายนิ้วมือเรียบร้อยแล้ว
- FPE_R_MODULENOTINITED : ไม่ได้มีการ Initialize Module ก่อนหน้านี้

fpe_StartEnrolment

IDWORKS Integrator Module: FPENROL.DLL

เริ่มการ Enroll ลายนิ้วมือ

Function Definition

C/C++-style function definition:

```

typedef FPE_RCODE ( __stdcall *fpe_StartEnrolment)(
    FPE_CONTEXT AEnrolmentContext,
    int AFingerIndex,
    void *ARawDataPtr,
    unsigned int AMaxRawDataSize,
    unsigned int *AOutputRawDataSize,
    int *AIndexingValue

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงแหล่งเอกสารทุกครั้งที่ใช้ **44**

Pascal-style function definition:

```
fpe_StartEnrolment : function(
    AEnrolmentContext: FPE_CONTEXT;
    AFingerIndex: Integer;
    ARawDataPtr: PByte;
    AMaxRawDataSize: Cardinal;
    var AOutputRawDataSize: Cardinal;
    AIndexingValue: PInteger
): FPE_RCODE; stdcall;
```

Visual Basic-style function definition:

```
Public Declare Function fpe_StartEnrolment Lib "FPENROL.DLL" _
(
    ByVal AEnrolmentContext As Long, _
    ByVal AFingerIndex As Long, _
    ByRef ARawDataPtr As Any, _
    ByVal AMaxRawDataSize As Long, _
    ByRef AOutputRawDataSize As Long, _
    ByRef AIndexingValue As Long _
) As Long
```

Parameters

AEnrolmentContext

Fingerprint Enrolment Context

AFingerIndex

ขณะ Start Enrolment ให้เลือกนิ้วใด

ARawDataPtr

Pointer ที่ชี้ไปที่ buffer สำหรับเก็บข้อมูล binary ของลายนิ้วมือ

AMaxRawDataSize

ขนาดสูงสุดที่สามารถจัดเก็บข้อมูล binary ลงใน ARawDatPtr ได้

AOutputRawDataSize

จะเป็นขนาดที่ฟังก์ชันส่งออกมาเพื่อบอกขนาดของ buffer ที่ได้ใช้ไป หากเป็น 0 หมายถึง ผู้ใช้กดยกเลิก

AIndexingValue

ถ้าไม่ใช่ null ฟังก์ชันจะส่งค่า FingerprintIndexingValue ออกมาให้ เพื่อนำไปจัดเก็บ

Return Value

- FPE_R_OK : การทำงานสำเร็จ
- FPE_R_MODULENOTINITED : ไม่ได้มีการ Initialize Module ก่อนหน้านี้

หมายเหตุ

ถ้าผู้ใช้กดยกเลิก ค่าที่ return มากก็จะเป็น FPE_R_OK แต่ AOutputRawDataSize จะมีค่าเป็น 0 ซึ่งหมายถึงการทำงานเสร็จสมบูรณ์ แต่ไม่มีข้อมูลลายนิ้วมือ เนื่องจากผู้ใช้กดยกเลิก

ประเภทข้อมูล (Data Types)

FPE_RCODE

FPE_RCODE เป็นประเภทข้อมูล 32 บิต ที่เป็นค่าที่ส่งกลับมาเมื่อเรียกฟังก์ชัน เพื่อบอกถึงผลลัพธ์ของฟังก์ชันนั้นๆ

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long FPE_RCODE;
```

Pascal-style data type definition:

```
FPE_RCODE = Integer;
```

Visual Basic-style data type definition:

ใช้ประเภท Long

FPE_CONTEXT

FPE_CONTEXT เป็นประเภทข้อมูล 32-bit สำหรับ Context ที่เสมือนกับเป็นหน่วยย่อยในการลงทะเบียนลายนิ้วมือ

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long FPE_CONTEXT;
```

Pascal-style data type definition:

```
FPE_CONTEXT = Longword;
```

Visual Basic-style data type definition:

ใช้ประเภท Long

FPE_SENSORTYPE

FPE_SENSORTYPE เป็นประเภทข้อมูล 32 บิตที่ใช้สำหรับระบุประเภทของเครื่องอ่านลายนิ้วมือ

Data-type Definition

C/C++-style data type definition:

```
typedef unsigned long FPE_SENSORTYPE;
```

Pascal-style data type definition:

```
FPE_SENSORTYPE = Integer;
```

Visual Basic-style data type definition:

ใช้ประเภท Long

ค่าคงที่ (Constants)

ค่าคงที่ทั่วไป

ชื่อค่าคงที่	ค่า	ความหมาย
FPC_MAXRAWDATASIZE	4096	ขนาด memory buffer สูงสุดที่ต้องการในการลงทะเบียนลายนิ้วมือ

ค่าคงที่สำหรับ ISL_RCODE

ชื่อค่าคงที่	ค่า	ความหมาย
FPE_R_OK	0	การทำงานเสร็จสิ้น ไม่มีข้อผิดพลาด
FPE_R_INVALIDCONTEXT	-1	Context ไม่ถูกต้อง
FPE_R_GENERALFAILURE	-2	มีข้อผิดพลาดทั่วไป
FPE_R_MODULENOTINITED	-3	เรียกฟังก์ชันโดยไม่มี การ Initialize Module
FPE_R_SENSORNOTFOUND	-4	ไม่พบเครื่องอ่านลายนิ้วมือที่ระบุ
FPE_R_INTERNALERROR	-4096	มีข้อผิดพลาดภายใน module

ค่าคงที่สำหรับ ISL_DEVICETYPE

ชื่อค่าคงที่	ค่า	ความหมาย
FPE_ST_UAREU	1	ประเภทของเครื่องอ่านลายนิ้วมือ

Appendix: A - การแก้ไขข้อผิดพลาดระหว่างการติดตั้ง

การแก้ไขข้อผิดพลาดระหว่างการติดตั้ง

มีสาเหตุสองสามประการที่อาจทำให้เกิดปัญหาในการติดตั้งเครื่องอ่านลายนิ้วมือ เช่น

- ในบางครั้งคุณอาจเกิดปัญหาถ้าติดตั้งเครื่องอ่านลายนิ้วมือก่อนติดตั้งไดรเวอร์ หรือ
- คุณอาจเกิดปัญหาในกรณีที่เคยติดตั้งไดรเวอร์รุ่นเก่ามาก่อนที่ไม่สามารถทำงานร่วมกันได้ หรือ
- คุณอาจติดตั้งโปรแกรมอื่น ๆ ที่มีไดรเวอร์ที่ไม่สามารถทำงานร่วมกันได้ การอัปเดตเวอร์ชันของ Windows ก็อาจทำให้เกิดปัญหาได้

ปัญหาทั้งหมดที่กล่าวมาสามารถแก้ไขได้โดยวิธีเดียวกันคือ

1. ถอดสายเครื่องอ่านลายนิ้วมือออกจากเครื่องคอมพิวเตอร์ของคุณ
2. ถอดโปรแกรมและไดรเวอร์ทั้งหมดของเครื่องอ่านลายนิ้วมือออกจากเครื่องคอมพิวเตอร์ของคุณ
 - I. เลือกที่ปุ่ม "Start" (มุมซ้ายล่างของจอ)
 - II. เลือก "Setting Menu"
 - III. คลิกที่ "Control Panel"
 - IV. ดับเบิลคลิกที่ "Add Remove Programs"
 - V. เลือกถอดโปรแกรมทั้งหมดที่มีคำว่า "IDWORKS", "digitalPersona" หรือ "U.are.U" ทั้งหมด.
3. รีเซ็ตเครื่องคอมพิวเตอร์ของคุณ
4. ติดตั้งโปรแกรม IDWORKS ใหม่ทั้งหมดตั้งแต่เริ่ม รวมทั้งไดรเวอร์เครื่องอ่านลายนิ้วมือ
5. เสียบสายเครื่องอ่านลายนิ้วมืออีกครั้ง

สิ่งอื่น ๆ ที่อาจทำให้เกิดปัญหาอาจอยู่ที่สาย USB หรือ USB Hubs ที่ใช้งาน ถ้าหากว่าคุณสามารถใช้งานเครื่องอ่านลายนิ้วมือได้โดยการเสียบสายโดยตรงกับเครื่อง นั่นแสดงว่ามีปัญหาบางอย่างจากการต่อสาย สิ่งที่คุณควรสังเกตได้แก่

- สายอาจเสียหายหรือหลวม
- พลังงานไฟฟ้าไม่พอ ในกรณีที่มีการใช้งานอุปกรณ์ USB หลายๆ ชั้นบน Hub ตัวเดียวกันอาจทำให้พลังงานไฟฟ้าไม่พอ
- การต่อสายพ่วงที่ยาวมากๆ อาจทำให้เครื่องอ่านลายนิ้วมือไม่สามารถทำงานได้

Appendix: B – การใช้งานและการดูแลรักษาเครื่องอ่านลายนิ้วมือ

การใช้งานเครื่องอ่านลายนิ้วมืออย่างถูกวิธี

พยายามแตะให้กว้างครอบคลุมพื้นที่มากที่สุด

ในการแตะอุปกรณ์เพื่อลงทะเบียนหรือลงทะเบียน ผู้ใช้ควรวางปลายนิ้วไว้ที่ส่วนบนสุดของกระจกรับภาพของอุปกรณ์ และแตะนิ้วลงไปให้สัมผัสกระจกให้ได้ครอบคลุมพื้นที่มากที่สุด เพราะอุปกรณ์จะอ่านลายนิ้วมือจากส่วนที่สัมผัสกับกระจกเท่านั้น ดังนั้นผู้ใช้แตะพื้นผิวได้กว้างเท่าใดก็จะเป็นการดีสำหรับการประมวลผลข้อมูลลายนิ้วมือได้อย่างแม่นยำ

วางนิ้วให้ขนานกับแนวราบ

บ่อยครั้งที่ผู้ใช้อุปกรณ์จะแตะอุปกรณ์ในลักษณะเดียวกับการกดปุ่ม ซึ่งเป็นลักษณะการใช้งานที่ไม่ถูกต้อง เพราะในลักษณะนั้น ซึ่งจะทำให้ลายนิ้วมือที่อ่านได้มีคุณภาพต่ำ (โดยปกติอุปกรณ์จะไม่อ่านลายนิ้วมือจากการสัมผัสในลักษณะนี้ เนื่องจากมีพื้นที่สัมผัสน้อยกว่าที่กำหนด) ซึ่งลักษณะที่ถูกต้องในการสัมผัสอุปกรณ์ จะต้องวางนิ้วขนานกับแนวราบลงไปกับกระจกรับภาพของอุปกรณ์

ไม่ควรกดแรงเกินไป

ลายนิ้วมือที่อุปกรณ์อ่านได้นั้น มีที่มาจากรายละเอียดและความดันสีกบนพื้นผิวของลายนิ้วมือของบุคคลนั้นๆ การกดอุปกรณ์แรงเกินไปจะทำให้ลายนิ้วมือที่อ่านได้ไม่แม่นยำ และถูกต้องน้อยลง ในทางกลับกัน หากแตะเบาเกินไป จะทำให้สัมผัสกับอุปกรณ์น้อยกว่าที่จะอ่านข้อมูลได้ การใช้งานที่ถูกต้องคือ ผู้ใช้จะต้องวางนิ้วลงกับอุปกรณ์ให้พอดี และกดลงไปเบาๆ

เล็บยาว?

บ่อยครั้งที่ผู้ใช้ที่มีเล็บยาว จะมีปัญหาเกี่ยวกับการใช้เครื่องอ่านลายนิ้วมือ เนื่องจากผู้ใช้มักจะใช้ปลายเล็บวางไว้ที่ส่วนบนสุดของกระจกรับภาพ แทนที่จะเป็นปลายนิ้ว ซึ่งวิธีการที่ถูกต้องคือ จะต้องวางนิ้วให้เล็บเลียบกระจกรับภาพไปทางด้านบน และให้ปลายนิ้วส่วนที่เป็นเนื้อ วางอยู่ที่ขอบบนของกระจกพอดี

อย่าหมุนนิ้ว

ในบางครั้งผู้ใช้อาจจะคิดว่า การที่อุปกรณ์จะอ่านลายนิ้วมือได้ดี จะต้องกดแล้วเอียงนิ้วซ้าย-ขวา ในลักษณะของการพิมพ์ลายนิ้วมือบนกระดาษ ซึ่งเป็นวิธีการที่ไม่แนะนำอย่างยิ่งเนื่องจากการกดในลักษณะเช่นนั้นจะทำให้ข้อมูลที่ได้อาจจะไม่มีความแม่นยำเท่าที่ควร และจะไม่ได้ผลที่ดีกว่า เมื่อเทียบกับการกดนิ้วลงไปนิ่งๆ บนกระจกรับภาพ

มือแห้ง

ถึงแม้ว่าเราอาจจะพบกับปัญหานี้บ่อยมากก็ตาม อย่างไรก็ตาม ปัญหาเหล่านี้สามารถแก้ไขได้ ในบางกรณี ความชื้นที่อุ้งมือก็เป็นส่วนที่สำคัญอย่างมากสำหรับการอ่านลายนิ้วมือโดยเครื่องอ่านลายนิ้วมือ อย่างไรก็ตาม ปัญหาเหล่านี้สามารถแก้ไขได้ ในบางกรณี ผู้ใช้ที่มีมือที่แห้งมากจนเกินไป อาจจะมีปัญหาในการใช้อุปกรณ์ ซึ่งวิธีการแก้ปัญหาสำหรับผู้ใช้คือให้ผู้ใช้ใช้นิ้วถูกับฝ่ามืออีกข้างสักครู่ จนรู้สึกว่ามีมือที่ชื้นอยู่บ้าง และลองกดดูอีกครั้งหนึ่ง สำหรับผู้ใช้ที่มีปัญหาที่มือแห้งเป็นประจำ อาจจะต้องหมั่นใช้โลชั่นเพื่อให้มือมีความชุ่มชื้นบ้าง อย่างไรก็ตาม ผู้ใช้ควรระวังหลีกเลี่ยงการใช้งานอุปกรณ์ทันทีหลังจากเพิ่งทาโลชั่นเสร็จใหม่ๆ เนื่องจากจะทำให้มีคราบโลชั่นติดบนกระจกรับภาพของอุปกรณ์

การป้องกันปัญหาที่อาจเกิดขึ้น

มือสกปรก?

กรุณาตรวจสอบให้แน่ใจว่านิ้วมือของท่านสะอาดและไม่เปียกน้ำก่อนการใช้งานอุปกรณ์ทุกครั้ง โดยนิ้วมือของท่านไม่จำเป็นต้องสะอาดถึงขั้นที่จะต้องล้างมือก่อนการใช้งานแต่ก็ไม่ควรมีฝุ่นสกปรกที่อาจหลงเหลือติดอยู่บนเครื่องอ่านลายนิ้วมือ โดยทั่วไปการปิดฝุ่นสกปรกออกจากมือก่อนการใช้งานก็เพียงพอแล้วแม้แต่ในสภาพแวดล้อมที่มีการใช้งานอย่างสมบุกสมบันอย่างเช่นตามสถานที่ก่อสร้าง อย่างไรก็ตามเครื่องอ่านลายนิ้วมือที่ใช้งานในสภาพแวดล้อมเช่นนี้อาจต้องทำความสะอาดบ่อยครั้ง (กรุณาดูวิธีการทำความสะอาดเครื่องอ่านลายนิ้วมือในส่วนถัดไป)

แสงแดด

การวางเครื่องอ่านลายนิ้วมือไว้ในที่มีแสงแดดจ้าอาจรบกวนการทำงานของเครื่องอ่านลายนิ้วมือได้

แอลกอฮอล์และน้ำยาทำความสะอาด

แอลกอฮอล์และน้ำยาทำความสะอาด จะทำลายแผ่นฟิล์มบางๆบนเครื่องอ่านลายนิ้วมือได้

น้ำตาล

เช่นเดียวกับแอลกอฮอล์ น้ำตาลที่อยู่บนนิ้วมือเมื่อผสมกับเหงื่อจะกัดกร่อนแผ่นฟิล์มบางๆบนเครื่องอ่านลายนิ้วมือ ถ้าเป็นไปได้ผู้ใช้งานที่อาจมีน้ำตาลอยู่บนนิ้วมือเช่นผู้ใช้งานที่ทำงานในครัว ควรล้างมือก่อนใช้งานเครื่องอ่านลายนิ้วมือ

การทำความสะอาดเครื่องอ่านลายนิ้วมือ

การทำความสะอาดจะขึ้นกับปริมาณการใช้งาน เครื่องอ่านลายนิ้วมืออาจต้องมีการทำความสะอาดบ้างเนื่องจากฝุ่นละอองและเหงื่อที่อาจตกหล่นอยู่จนทำให้ช่องอ่านลายนิ้วมือมัวได้

วิธีที่แนะนำในการทำความสะอาด

ในการทำความสะอาดฝุ่นละอองที่อาจตกหล่นอยู่ ให้ใช้สก็อตเทปชิ้นเล็กๆด้านที่มีกาวเหนียวติดลงบนช่องอ่านลายนิ้วมือแล้วดึงออก



ถ้าต้องการเช็ดทำความสะอาด ต้องใช้ผ้านุ่มที่ไม่มีขุยหลุดออกมาและน้ำหรือน้ำยาเช็ดกระจกที่เป็นส่วนผสมของแอมโมเนียเช็ดเบาๆ

ข้อควรระวังในการทำความสะอาด

อย่าใช้กระดาษเช็ดช่องอ่านลายนิ้วมือ อย่าจุ่มหรือเทน้ำลงบนเครื่องอ่านลายนิ้วมือ และแอลกอฮอล์ จะทำลายแผ่นฟิล์มบางๆบนเครื่องอ่านลายนิ้วมือได้



IT WORKS
IDWORKS Integrator SDK
Programmer's Guide

V2.0
Updated: 07 September 2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



IDWORKS Integrator SDK
Programmer's Guide

การทำงานของ ID-WORKS Integrator

ID-WORKS Integrator เป็นชุดพัฒนาที่ใช้สำหรับเก็บข้อมูลและตรวจสอบลายนิ้วมือเพื่อยืนยันตัวบุคคล ซึ่งนักพัฒนาสามารถนำไปพัฒนาร่วมกับโปรแกรมประยุกต์ได้โดยง่าย โดยมีรูปแบบการทำงานซึ่งแสดงด้วยรูปภาพง่าย ๆ ดังนี้



ขั้นตอนการทำงานจะประกอบด้วย การจัดเก็บลายนิ้วมือ (Enrollment) และ การตรวจสอบลายนิ้วมือ (Identification) โดยลูกศรสีเหลืองจะแสดงการทำงานในส่วนของการจัดเก็บลายนิ้วมือจากผู้ใช้งานแล้วบันทึกลงในคอมพิวเตอร์ ส่วนลูกศรสีเขียวและสีน้ำเงิน แสดงขั้นตอนการแต่นิ้วและประมวลผลเพื่อยืนยันลายนิ้วมือ

การจัดเก็บลายนิ้วมือ ทำเพื่อใช้เก็บข้อมูลลักษณะของนิ้วนั้นๆ แล้วใช้เปรียบเทียบในภายหลัง ซึ่งค่าที่ได้จากการเก็บลายนิ้วมือจะเป็นข้อมูลในรูปแบบ binary มีความยาวประมาณ 400~600 byte หลังจากได้ข้อมูลลายนิ้วมือแล้วให้นำไปเก็บไว้ในสื่อเก็บข้อมูลชนิดใดก็ได้ เช่น hard disk ซึ่งอาจเก็บในรูปแบบของ database หรือ file ข้อมูลทั่วไป หรือเก็บลงในบัตร smart card จากนั้น เมื่อต้องการทำการเปรียบเทียบลายนิ้วมือ ต้องอ่านข้อมูลที่บันทึกไว้ เพื่อส่งให้ชุดพัฒนาเก็บไว้ในหน่วยความจำ เมื่อมีการแต่นิ้ว ชุดพัฒนาจะเปรียบเทียบลายนิ้วมือจากเครื่องอ่านลายนิ้วมือ กับ ข้อมูลลายนิ้วมือที่อยู่ในหน่วยความจำเพื่อประมวลผลว่าพบหรือไม่พบต่อไป



หลักการทำงานโดยสรุป

1. ก่อนเริ่มต้นการใช้งานทุกครั้ง ต้องสั่งให้เครื่องอ่านลายนิ้วมือเริ่มต้นการทำงานก่อน (Activate) ด้วยรหัส 3 ชุดที่ได้มากับตัวผลิตภัณฑ์ ซึ่งรหัสทั้ง 3 ชุดนี้จะเป็นคุณสมบัติเฉพาะของเครื่องอ่านลายนิ้วมือแต่ละเครื่องและไม่ซ้ำกัน โดยรหัสทั้ง 3 ชุดประกอบด้วย
 - a. Sensor ID เป็นรหัสประกอบด้วยตัวอักษรและตัวเลข ตัวอย่างเช่น
X12X123X-X12X-X123-X12X-X123XX1234XX
 - b. Serial Number เป็นรหัสประกอบด้วยตัวอักษรและตัวเลข ตัวอย่างเช่น
X12X-X123-XX12-X12X
 - c. Application Key เป็นรหัสประกอบด้วยตัวอักษรและตัวเลข ตัวอย่างเช่น
123X-12XX-1XXX-X123-XX12-XXX1-X11X-X22X(* รูปแบบและจำนวนอักษรของรหัสทั้ง 3 ชุดอาจเปลี่ยนแปลงได้ *)
2. หลังจาก Activate เครื่องอ่านลายนิ้วมือแล้ว สำหรับคนที่ยังไม่เคยเก็บลายนิ้วมือ (Enrollment) ต้องทำการเก็บลายนิ้วมือก่อน โดยชุดพัฒนาจะมีฟังก์ชันสำหรับเก็บลายนิ้วมือให้ เมื่อเรียกฟังก์ชันแล้ว ชุดพัฒนาจะแสดงหน้าต่างสำหรับเก็บลายนิ้วมือให้ ดังนี้



ผู้ที่ต้องการเก็บลายนิ้วมือต้องแตะนิ้วที่ต้องการบนเครื่องอ่านลายนิ้วมือ 4 ครั้งด้วยนิ้วเดียวกัน ชุดพัฒนาจะทำการประมวลผลลายนิ้วมือทั้ง 4 ครั้ง เพื่อหาค่า Template สำหรับนิ้วนั้น ๆ โดย

เอกสารนี้เป็นเอกสารที่... IDWORKS Integrator SDK Programmer's Guide



ของนิ้วนั้น ๆ ฟังก์ชันจะจบการทำงานเมื่อมีการกดปุ่มตกลงหรือยกเลิก หากมีการกดปุ่มตกลง (ฟังก์ชันทำงานสำเร็จ) ฟังก์ชันจะส่งค่าของ Template สำหรับนิ้วนั้น ๆ กลับมาให้ ซึ่งผู้พัฒนาต้องนำไปจับคู่กับหมายเลข ID ที่ต้องการ แล้วจัดเก็บลงเพิ่มข้อมูล, ฐานข้อมูล, Smart Card หรือ ที่เก็บข้อมูลอื่น ๆ ตามต้องการ หมายเลข ID ที่ใช้จับคู่กับลายนิ้วมือนี้สามารถใช้หมายเลขเดียวกันได้สำหรับลายนิ้วมือหลาย ๆ นิ้วโดยไม่จำกัด ซึ่งโดยทั่วไปจะใช้ primary key ของตารางที่เก็บข้อมูลพนักงานหรือผู้ใช้ เช่น สมมุติว่าตารางที่จัดเก็บข้อมูลพนักงานชื่อ User มี column ข้อมูล 2 Column ดังนี้

User ID	User Full Name
0001	นาย ทักษิณ สุขหล่อ
0002	นาย อภิสัทธ์ สุขเท่

ดังนั้นหากทำการจัดเก็บนิ้วของ นาย ทักษิณ สมมุติให้เป็น นิ้วชี้ข้างขวา หลังจากที่ได้ Template ข้อมูลแล้ว ผู้พัฒนาสามารถนำ User_ID ของ นาย ทักษิณ ซึ่งจากข้อมูลตัวอย่างเป็น 0001 มาจับคู่กับ Template ของลายนิ้วมือนี้ได้ และ หาก ทำการจัดเก็บลายนิ้วมือของนาย ทักษิณ เพิ่มเติมเช่น เก็บนิ้วนางข้างซ้าย ผู้พัฒนาอาจใช้ หมายเลข ID เดิมคือ 0001 จับคู่กับลายนิ้วมือใหม่นี้ได้ เพราะทั้งสองนิ้วเป็นลายนิ้วมือของคน ๆ เดียวกัน และ เมื่อนำข้อมูลที่จัดเก็บไว้ส่งให้ชุดพัฒนาเพื่อทำการเปรียบเทียบ ต่อมาเมื่อ นาย ทักษิณ ใช้นิ้วแตะเครื่องอ่านลายนิ้วมือ หากเป็นนิ้วชี้ข้างขวา หรือ นิ้วนางข้างซ้าย ชุดพัฒนาจะสามารถตรวจพบ ส่ง หมายเลข ID ที่จับคู่ไว้กลับมา ซึ่งในตัวอย่างจะเป็นหมายเลข 0001 ทำให้ผู้พัฒนาสามารถรู้ได้ว่านิ้วที่ถูกตรวจสอบเป็นนิ้วของใคร

- เมื่อต้องการให้ชุดพัฒนาในการตรวจสอบลายนิ้วมือ ผู้พัฒนาต้องอ่านข้อมูล Template และ หมายเลข ID ที่จับคู่ไว้จากแหล่งข้อมูล เพื่อส่งให้ ชุดพัฒนาเป็นคู่ ๆ และทำเช่นนี้สำหรับทุก ๆ ลายนิ้วมือที่ต้องการให้ระบบสามารถตรวจพบได้ ซึ่งหมายความว่าข้อมูล Template ทั้งหมดของทุก ๆ นิ้วที่ต้องการตรวจสอบจะถูกเขียนลงบนหน่วยความจำของเครื่องคอมพิวเตอร์ที่เชื่อมต่อกับเครื่องอ่านลายนิ้วมือ และการประมวลผลจะทำกับข้อมูลที่อยู่บนหน่วยความจำเครื่องนี้เท่านั้น
- เมื่อมีการแตะนิ้ว ชุดพัฒนาจะทำการประมวลผลลายนิ้วมือว่าเป็นลายนิ้วมือที่อยู่ในหน่วยความจำหรือไม่ ถ้าประมวลผลแล้วพบว่าเป็นนิ้วที่ได้ทำการจัดเก็บไว้แล้ว (อยู่ในหน่วยความจำ) ชุดพัฒนาจะให้ค่าที่ระบุ ว่า ตรวจพบลายนิ้วมือ และ ส่งหมายเลข ID ที่ถูกจับคู่ไว้กับลายนิ้วมือนั้นออกมา ซึ่งผู้พัฒนาสามารถนำไปใช้เพื่อระบุได้ว่าหมายเลข ID นี้เป็นใครต่อไป

