

ระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือ
และบันทึกภาพใบหน้า

ACCESSING DOOR ENTRANCE USING FINGERPRINT VERIFICATION AND
FACE RECORDING SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือ
และบันทึกภาพใบหน้า

ACCESSING DOOR ENTRANCE USING FINGERPRINT VERIFICATION AND
FACE RECORDING SYSTEM



โดย

นายชลวัฒน์	มุกดาสนิท	55010247
นายชลวิทย์	ผาวันดี	55010249
นางสาวศรัณยา	สุขมาก	55011190

อาจารย์ที่ปรึกษา
รศ.ดร.ไกรลีน สงวัฒนา

อาจารย์ที่ปรึกษาร่วม
ผศ.ดร.พิเชฐ ม่วงนวล

เลขหมู่.....
เลขทะเบียน 144404
วันเดือนปี 24 พ.ย. 2559

b. 18918549
i.

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ผ่านการตรวจสอบเรียบร้อยแล้ว ปีการศึกษา 2558

ผ่านการตรวจชิ้นงานแล้ว

เอกสารที่ส่งมาเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์อื่นใด (ยกเว้นกรณีที่มีการขออนุญาต)
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลนี้ออกไป และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการ
วิศวกรรมโทรคมนาคม
Telecommunications Engineering

กรรมการผู้ตรวจชิ้นงาน
10/15/59
วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญาโทปีการศึกษา 2558

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

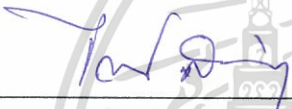
เรื่อง ระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือและบันทึกภาพใบหน้า

ACCESSING DOOR ENTRANCE USING FINGERPRINT VERIFICATION

AND FACE RECORDING SYSTEM


ผู้จัดทำ

1. นายชลวัฒน์ มุกดาสนิท 55010247
2. นายชลวิทย์ ผาวันดี 55010249
3. นางสาวศรัณยา สุขมาก 55011190



(รศ.ดร.ไกรสิน ส่องวัฒนา)

อาจารย์ที่ปรึกษา



(ผศ.ดร.พิเชฐ ม่วงนวล)

อาจารย์ที่ปรึกษาร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการจากการศึกษาฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด ซึ่งผู้มีพระคุณท่านแรกที่คุณศึกษาใคร่ขอกราบขอบพระคุณคือ ท่านรองศาสตราจารย์ด็อกเตอร์ไกรสิน ส่งวัฒนา และผู้ช่วยศาสตราจารย์ด็อกเตอร์พิเชฐ ม่วงนวล อาจารย์ผู้สอนที่ได้ให้ความรู้ สนับสนุนงบประมาณ คำแนะนำ ตรวจสอบ แก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ทุกขั้นตอน และเทคนิคการนำเสนอรายงานปากเปล่า เพื่อให้การค้นคว้าฉบับนี้สมบูรณ์ที่สุด ผู้ศึกษาใคร่ขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้ นอกจากนี้ขอขอบคุณภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้สนับสนุนงบประมาณในการค้นคว้า ขอขอบคุณนายชลธิศ ผาวันดี และ นายชโลธร ผาวันดี พี่ชายของนายชลวิทย์ ผาวันดี ที่ได้ให้คำแนะนำในการเขียนและออกแบบโปรแกรมด้วยภาษา C# ให้ความรู้และคำแนะนำเกี่ยวกับการออกแบบและใช้งานฐานข้อมูล และให้กำลังใจในการศึกษาค้นคว้าตลอดมา ขอขอบคุณอาจารย์ภาควิชาวิศวกรรมโทรคมนาคมทุกท่านที่ได้ฝึกสอน ได้คำแนะนำในการจัดทำโครงการการศึกษาค้นคว้าฉบับนี้ที่ไม่ได้กล่าวนาม

นายชลวัฒน์ มุกดาสนิท
นายชลวิทย์ ผาวันดี
นางสาวศรัณยา สุขมาก
ผู้จัดทำ

ระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือ
และบันทึกภาพใบหน้า

ACCESSING DOOR ENTRANCE USING FINGERPRINT
VERIFICATION AND FACE RECORDING SYSTEM

โดย	นายชลวัฒน์	มุกดาสนิท	55010247
	นายชลวิทย์	ผาวันดี	55010249
	นางสาวศรัณยา	สุขมาก	55011190

อาจารย์ที่ปรึกษา	รศ.ดร.ไกรสิน	สงวัฒนา
อาจารย์ที่ปรึกษาร่วม	ผศ.ดร.พิเชฐ	ม่วงนวล

บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อประดิษฐ์อุปกรณ์สำหรับควบคุมบุคคลเข้าออกประตูของห้องปฏิบัติการ และบันทึกข้อมูลการเข้าเรียนวิชาปฏิบัติการของนักศึกษาวิศวกรรมโทรคมนาคมชั้นปีที่สอง เพื่อความสะดวกจึงประดิษฐ์อุปกรณ์ที่สามารถจัดเก็บและตรวจสอบข้อมูลการเข้าห้องปฏิบัติการได้ โดยการทำงานของระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือและบันทึกภาพใบหน้า การตรวจสอบจะทำงานโดยใช้คอมพิวเตอร์สื่อสารกับไมโครคอนโทรลเลอร์สำหรับนักศึกษาที่จะเข้าเรียนวิชาปฏิบัติการสามารถปลดล็อคประตูได้โดยการตรวจกราดลายนิ้วมือและมีการบันทึกภาพและเวลาเข้าของบุคคลนั้นลงในฐานข้อมูลด้วย

ABSTRACT

This project proposes a device for controlling access through the door of the laboratory room and record entry information for the second year telecommunication engineering students. We have constructed a device that checks and stores entry information using fingerprint verification and face image recording system. The access door works with computer, microcontroller and webcam camera. Student can unlock the door by scanning their finger on the fingerprint scanner and device will match their fingerprint to the fingerprint that have been stored in scanner database. Then the device records the picture and entry information of students to the database when access.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 ไปโอเมตริกเบื้องต้น	2
2.1.1 คุณสมบัติของไปโอเมตริก	2
2.1.2 การใช้งานระบบไปโอเมตริกเบื้องต้น	3
2.2 ระบบรู้จำลายนิ้วมือเบื้องต้น	4
2.2.1 การเก็บภาพลายนิ้วมือ	5
2.2.2 การประเมินคุณภาพลายนิ้วมือ	6
2.2.3 การปรับปรุงภาพลายนิ้วมือ	7
2.2.4 การสกัดลักษณะเด่นจากภาพลายนิ้วมือ	8
2.2.5 การจับคู่ลายนิ้วมือ	12
2.3 MICROSOFT VISUAL C#	13
2.3.1 MICROSOFT .NET	13
2.3.2 MICROSOFT .NET FRAMEWORK	14
2.4 ไมโครซอฟท์ แอคเซส	18
2.4.1 คำศัพท์ที่ควรทราบ	18
2.4.2 ส่วนประกอบของฐานข้อมูลแอคเซส	19
2.4.3 การสร้างและการกำหนดคุณสมบัติของฟิลด์	20
2.5 ARDUINO	23
2.5.1 ARDUINO UNO R3	23
2.5.2 การสั่งงานผ่านโปรแกรม ARDUINO IDE	25
2.6 การทำงานของรีเลย์	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.6.1 ส่วนประกอบของรีเลย์	28
2.6.2 จุดต่อใช้งานมาตรฐาน	29
2.6.3 ข้อจำกัดในการใช้งานรีเลย์ทั่วไป	29
บทที่ 3 การออกแบบและการจัดทำโครงงาน	31
3.1 การออกแบบ	31
3.1.1 การเขียนโปรแกรมควบคุม MICROCONTROLLER	31
3.1.2 การการเขียนโปรแกรมควบคุมคอมพิวเตอร์	36
3.2 เครื่องมือที่ใช้ในการทดลอง	59
3.2.1 ประตูกлонควบคุมด้วยกระแสไฟฟ้า	59
3.2.2 ARDUINO UNO R3	60
3.2.3 RELAY HRS1H-S-DC3V	61
3.2.4 ADAFRUIT OPTICAL FINGERPRINT SENSOR	61
3.2.5 WEBCAM CAMERA	62
3.2.6 COMPUTER	62
3.3 การจัดเก็บผลการทดลอง	63
3.3.1 อัตราการล้มเหลวในการยึดเก็บสัญญาณ	63
3.3.2 อัตราการระบุ	63
บทที่ 4 ผลการทดลอง	64
4.1 การทดลองการยืนยันตัวตน	64
4.2 การทดลองส่งผลการบันทึกผ่านอีเมล	68
4.3 การวัดค่าความแม่นยำของ FINGERPRINT SENSOR	70
บทที่ 5 สรุปผลและข้อเสนอแนะ	71
5.1 สรุปผล	71
5.2 ข้อเสนอแนะ	71
เอกสารอ้างอิง	72
ภาคผนวก ก คำสั่งโปรแกรมที่ใช้ในการสั่งงาน	74
ภาคผนวก ข รายละเอียดอุปกรณ์	98
ภาคผนวก ค คู่มือการใช้งาน	104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	การลงทะเบียน (REGISTRATION)	4
2.2	การพิสูจน์ตัวตนจริง (AUTHENTICATION)	4
2.3	แผนภาพบล็อกพื้นฐานของระบบรู้จำลายนิ้วมือเบื้องต้น	4
2.4	เทคนิคการสะท้อนภายในกลับหมด	6
2.5	แผนภาพบล็อกกระบวนการตรวจหาเส้นสัน	9
2.6	การกำหนดตำแหน่งและมุมของมินูเซียแบบจุดแยกและจุดหยุด	10
2.7	การตรวจหาจุดมินูเซียด้วยหน้าต่างขนาด 3X3 จุดภาพ	11
2.8	แผนภาพบล็อกกระบวนการตรวจหาจุดมินูเซียจากภาพลายนิ้วมือนำเข้า	11
2.9	ผลลัพธ์จากที่ได้แต่ละขั้นตอนของการหาจุดมินูเซีย	12
2.10	การสื่อสารผ่านตัว.NET	14
2.11	ส่วนประกอบของเฟรมเวิร์ก	14
2.12	.NET FRAMEWORK 4.5	16
2.13	ความสัมพันธ์ระหว่างกลุ่มออบเจกต์ ADO.NET	16
2.14	แบบจำลองของกลุ่มออบเจกต์ OLEDB	17
2.15	วัตถุฐานข้อมูล(DATABASE OBJECT)	19
2.16	ตารางลูกค้า	19
2.17	ตัวอย่างในการสร้างฟิลต์	21
2.18	ARDUINO UNO R3	23
2.19	รูปแสดงหน้าที่ต่างๆของขา ARDUINO	24
2.20	เชื่อมต่อ ARDUINO เข้ากับคอมพิวเตอร์	25
2.21	รูปแสดงการติดตั้ง DRIVER	25
2.22	โปรแกรม ARDUINO IDE	26
2.23	การเลือกรุ่นบอร์ด	26
2.24	กำหนดเป้าหมายเพื่อแปลโปรแกรม คำสั่งไปที่พอร์ตที่เลือก	26
2.25	รูปแสดง SERIAL PORT อยู่ที่ PORT COM5	27
2.26	แปลคำสั่งโปรแกรม	27
2.27	เขียนคำสั่งลงบอร์ด	28
2.28	รูปร่างและสัญลักษณ์ของรีเลย์	28
2.29	การทำงานของรีเลย์	29
3.1	แผนผังการทำงานของโปรแกรมที่ดำเนินงานบน ARDUINO	32

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.2	คำสั่งตัวอย่างใน LIBRARY ของ FINGERPRINT SENSOR	32
3.3	แผนผังการทำงานของการทำงานของการจับคู่ลายนิ้วมือ	33
3.4	แผนผังการทำงานของการทำงานของการลงทะเบียน	34
3.5	การปลดล๊อคคกตอนประตู	36
3.6	ตารางฐานข้อมูล	39
3.7	แผนผังการทำงานของโปรแกรมบนคอมพิวเตอร์	41
3.8	แผนผัง ACTION การรับค่าจาก SERIAL PORT	42
3.9	แผนผัง ACTION การกดปุ่ม CONNECT	47
3.10	แผนผัง ACTION การกดปุ่ม START	47
3.11	แผนผัง ACTION การรับภาพจากกล้อง	48
3.12	แผนผัง ACTION การกดปุ่ม STOP	48
3.13	แผนผัง ACTION การกดปุ่ม DELETE	49
3.14	แผนผัง ACTION การเปลี่ยนแปลงของ INDEX ใน DOORCOMBOBOX	50
3.15	แผนผัง ACTION การกดปุ่ม ENROLL	51
3.16	แผนผัง ACTION การกดปุ่ม LOG-IN	53
3.17	แผนผัง ACTION การกดปุ่ม SEND EMAIL	54
3.18	แผนผัง ACTION การดำเนินงานของนาฬิกา	55
3.19	แผนผัง ACTION การกดปุ่ม UNLOCK	56
3.20	แผนผัง ACTION การเปลี่ยนแปลงของ INDEX ใน LABTIMECOMBOBOX	58
3.21	หน้าจอโปรแกรม GRAPHIC INTERFACE	58
3.22	ประตูกลอนควบคุมด้วยกระแสไฟฟ้า	60
3.23	ARDUINO UNO R3	61
3.24	RELAY HRS1H-S-DC3V	61
3.25	อุปกรณ์ FINGERPRINT SENSOR	62
3.26	WEBCAM CAMERA	62
3.27	คอมพิวเตอร์ที่ใช้งาน	63
4.1	การเชื่อมต่ออุปกรณ์	64
4.2	การเปิดโปรแกรม ARDUINO	64
4.3	การเปิดโปรแกรม MICROSOFT VISUAL STUDIO	65
4.4	การเปิดโปรแกรม MICROSOFT VISUAL STUDIO	65
4.5	การดำเนินงานโปรแกรม	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญรูป(ต่อ)

รูปที่		หน้า
4.6	ขั้นตอนการเปิดการสื่อสารระหว่างอุปกรณ์และโปรแกรม	66
4.7	ตัวอย่างแฟ้มลงบันทึกเข้าออก	67
4.8	ตัวอย่างการบันทึกผลในฐานข้อมูล	67
4.9	ตัวอย่างภาพใบหน้าที่ถูกบันทึก	68
4.10	การกรอกอีเมลเพื่อเตรียมการส่งอีเมล	68
4.11	ส่งอีเมล	69
4.12	ตรวจสอบผลการส่งอีเมล	69
ก.1	รูปแบบการวางวัตถุต่างๆ	79
ค.1	การเชื่อมต่ออุปกรณ์	105
ค.2	หน้าต่างโปรแกรม	105
ค.3	หน้าต่างโปรแกรมเมื่อเชื่อมต่อกับพอร์ตอนุกรมและเปิดกล่อง	106
ค.4	หน้าต่างโปรแกรมเมื่อเริ่มลงทะเบียนลายนิ้วมือ	106
ค.5	หน้าต่างโปรแกรมเมื่อลงทะเบียนลายนิ้วมือเสร็จสิ้น	107
ค.6	หน้าต่างโปรแกรมหากลงทะเบียนลายนิ้วมือซ้ำ	107
ค.7	หน้าต่างโปรแกรมเมื่อทำการกราดตรวจพบลายนิ้วมือ	108
ค.8	หน้าต่างโปรแกรมเมื่อทำการกราดตรวจไม่พบลายนิ้วมือ	108
ค.9	หน้าต่างโปรแกรมเมื่อกดปุ่ม UNLOCK	109
ค.10	หน้าต่างโปรแกรมเมื่อทำการลบข้อมูลลายนิ้วมือ	109

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างเนมสเปซของคลาสพื้นฐาน	15
2.2 หน้าที่ของออบเจ็กต์ ADO.NET	17
2.3 ชนิดข้อมูล (DATA TYPE) ที่มีในแอคเซส	21
2.4 คุณสมบัติ FIELD SIZE ของฟิลด์ที่มีชนิดข้อมูลเป็น NUMBER	22
2.5 คุณสมบัติ FORMAT ของฟิลด์ที่มีชนิดข้อมูลเป็น DATE/TIME	23
3.1 ค่าที่ ARDUINO ส่งให้คอมพิวเตอร์	36
3.2 EVENT ACTION ทั้งหมด	40
3.3 หน้าที่ของวัตถุต่างๆของโปรแกรม	59
3.4 คุณสมบัติของ ARDUINO UNO R3	60
3.5 ขาสัญญาณในการเชื่อมต่อของ FINGERPRINT SENSOR	62
4.1 ผลการทดลองการกราดตรวจลายนิ้วมือ	70

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันความปลอดภัยของอุปกรณ์และเครื่องมือต่างๆในห้องปฏิบัติการวิชาอิเล็กทรอนิกส์ของภาควิชาวิศวกรรมโทรคมนาคมจำเป็นต้องดูแลรักษาให้พ้นจากผู้ไม่ประสงค์ดี และการมาเข้าเรียนวิชาปฏิบัติการของนักศึกษาชั้นปีที่สองภาควิชาวิศวกรรมโทรคมนาคมนับเป็นสิ่งสำคัญเช่นกัน เพราะจะได้นำความรู้ไปใช้ในการเรียนการสอนและสามารถลงมือปฏิบัติได้หากเจอกับงานจริง ในโครงการนี้จึงเป็นโครงการสำหรับเพิ่มความปลอดภัยให้กับอุปกรณ์และเครื่องมือต่างๆในห้องปฏิบัติการวิชาอิเล็กทรอนิกส์ และเก็บผลการมาเข้าเรียนวิชาปฏิบัติการของนักศึกษาชั้นปีที่สองภาควิชาวิศวกรรมโทรคมนาคม

1.2 วัตถุประสงค์

- 1) เพื่อเพิ่มความปลอดภัยให้กับอุปกรณ์และเครื่องมือภายในห้องปฏิบัติการวิชาอิเล็กทรอนิกส์ที่ติดตั้งระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือและบันทึกภาพใบหน้า
- 2) เพื่อจัดเก็บข้อมูลสำหรับนักศึกษาชั้นปีที่สองภาควิชาวิศวกรรมโทรคมนาคมที่ต้องการเข้า-ออก สถานที่ซึ่งติดตั้งอุปกรณ์ไว้ และทำการส่งข้อมูลของผู้ที่เข้า-ออกไปตามอีเมลที่ต้องการ

1.3 ขอบเขตของปริญญาณิพนธ์

ระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือและบันทึกภาพใบหน้าเป็นโครงการที่สร้างขึ้นเพื่อปลดล็อคประตูห้องปฏิบัติการวิชาอิเล็กทรอนิกส์และเก็บข้อมูลวัน-เวลา ของผู้ที่ทำการกราดตรวจลายนิ้วมือ โดยในโครงการนี้ให้นักศึกษาชั้นปีที่สองภาควิชาวิศวกรรมโทรคมนาคมเป็นตัวอย่งการทดลอง เมื่อกราดตรวจลายนิ้วมือหากมีข้อมูลอยู่จะสามารถปลดล็อคประตูและถูกบันทึกภาพเก็บไว้ในฐานข้อมูล และนอกจากนี้ยังมีระบบส่งข้อมูลชื่อและวัน-เวลา การเข้า-ออกไปยังอีเมลที่ต้องการได้ข้อมูลนั้นตามเวลาที่กำหนดไว้ในแต่ละวันหรือสามารถส่งงานให้ส่งไปได้ทันที

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ไบโอเมตริกเบื้องต้น (Introduction to Biometric)

นิยามของคำว่า ไบโอเมตริก (Biometric) คือ ลักษณะรูปลักษณ์ที่มีความแตกต่างระหว่างบุคคล หรืออัตลักษณ์ของบุคคล โดยคำว่าอัตลักษณ์นี้มาจากคำว่า อัตตา ที่แปลว่าตัวตน กับลักษณะที่แปลว่ารูปร่างลักษณะ

2.1.1 คุณสมบัติของไบโอเมตริก (Biometric Properties)

การที่จะเป็นลักษณะเฉพาะของมนุษย์ที่จะเป็นไบโอเมตริกได้นั้น ต้องมีคุณสมบัติ 7 ประการที่โดดเด่นจึงจะเป็นไบโอเมตริกที่สมบูรณ์ เป็นคุณสมบัติในอุดมคติ มีดังต่อไปนี้

2.1.1.1 ความเป็นสากล (Universality)

ลักษณะเฉพาะนี้จะต้องเป็นพื้นฐานของมนุษย์ทั่วไป ไม่ใช่เฉพาะมีคนเดียวคนหนึ่ง อาทิเช่น ใบหน้า นิ้วมือ ใบหู จมูก ปาก เป็นต้น

2.1.1.2 ความเป็นเอกลักษณ์ (Uniqueness)

ลักษณะเฉพาะที่มีอยู่ในมนุษย์ทั่วไป จะต้องมีความแตกต่างกันในแต่ละบุคคลมากเพียงพอที่จะใช้ในการแยกแยะแต่ละบุคคลออกจากกันได้

2.1.1.3 ความคงทนถาวร (Permanence)

ลักษณะเฉพาะที่มีอยู่ในมนุษย์ทั่วไปนี้ จะต้องมีความคงทนไม่เปลี่ยนแปลงไปตามเวลา หรือ เปลี่ยนแปลงไประหว่างการใช้งาน

2.1.1.4 ความสามารถตรวจวัดได้ (Measurability)

ลักษณะเฉพาะนี้ จะสามารถแปลงไปเก็บเป็นข้อมูลแทน เพื่อสามารถนำไปเปรียบเทียบกันได้ โดยการเก็บข้อมูลแทนสามารถใช้อุปกรณ์เฉพาะได้ แต่ต้องไม่สร้างความลำบากให้แก่ผู้ใช้ทั้งทางด้านร่างกายและทางด้านจิตใจ อาทิเช่น การใช้วิธีวะะปกปิด นอกจากนี้ ข้อมูลที่ได้ต้องเพียงพอที่จะนำไปใช้วัดความแตกต่างระหว่างบุคคลได้

2.1.1.5 สมรรถนะ (Performance)

เทคโนโลยีไบโอเมตริกที่นำมาใช้กับลักษณะเฉพาะนี้ ต้องมีการทำงานที่เร็วและแม่นยำในระดับที่ต้องการ รวมทั้งมีราคาที่เหมาะสม เหมาะกับการนำไปใช้

2.1.1.6 ความเป็นที่ยอมรับ (Acceptability)

ลักษณะเฉพาะนี้ต้องเป็นที่ยอมรับจากกลุ่มคนผู้ใช้งาน โดยผู้ใช้งานยินดีให้ความร่วมมือ หรือได้ประโยชน์จากการใช้งานนั้น

2.1.1.7 การปลอมแปลงได้ยาก (Circumvention)

คุณลักษณะนี้ควรปลอมแปลงได้ยาก รวมทั้งสามารถทำลาย หรือเปลี่ยนแปลงได้

2.1.2 การใช้งานระบบไบโอเมตริกเบื้องต้น (Introduction to Biometric System Usage)

กระบวนการใช้งานระบบไบโอเมตริกเบื้องต้น สามารถแบ่งเป็นสองส่วน ส่วนแรกจะเป็นการเก็บข้อมูลไบโอเมตริกเพื่อการใช้งาน เรียกว่า การลงทะเบียน (Registration) ส่วนที่สองจะเป็นการเรียกข้อมูลไบโอเมตริกที่เก็บไว้มาเปรียบเทียบกับข้อมูลที่เข้ามา เรียกว่า การพิสูจน์ตัวจริง (Authentication)

การลงทะเบียนประกอบด้วยกระบวนการย่อยหลายกระบวนการ ดังในรูปที่ 2.1 กระบวนการลงทะเบียนมีหน้าที่ในการจัดเก็บข้อมูลไบโอเมตริกเข้าสู่ฐานข้อมูลเพื่อการเรียกใช้ในอนาคต จะประกอบด้วย

2.1.2.1 การเก็บภาพ (Image Acquisition)

การเก็บภาพ ซึ่งจะแปลงลายนิ้วมือจากนิ้วมาเป็นภาพ ใช้การเก็บภาพ แบบกราดตรวจสด (Live-Scan) โดยการเก็บลายนิ้วมือเชิงแสง

2.1.2.2 การตัดแยก (Segmentation)

เพื่อตัดแยกส่วนที่มีลายนิ้วมือกับไม่มีลายนิ้วมือ ซึ่งเรียกว่าพื้นหลัง (Background) เป็นการแบ่งที่ชัดเจนและมีผลลัพธ์เป็นสองสถานะหรือสองระดับ คือผลลัพธ์เป็นบริเวณที่มีลายนิ้วมือหรือเป็นบริเวณที่ไม่มีลายนิ้วมือ

2.1.2.3 การปรับปรุง (Enhancement)

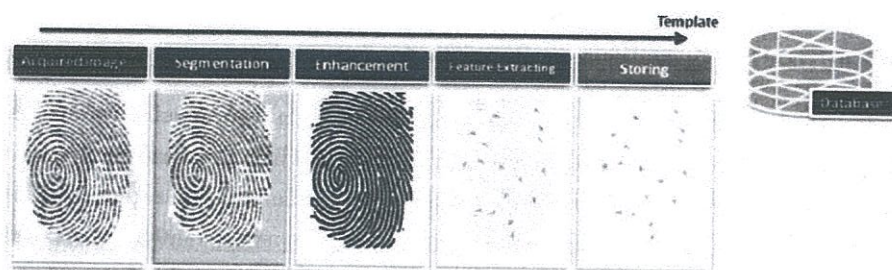
เป็นการปรับปรุงภาพลายนิ้วมือให้มีความคมชัดสมบูรณ์ขึ้นจากเดิม ซึ่งจะมีผลต่อความแม่นยำของระบบ เช่น ใช้เพื่อลดจุดขาวบนพื้นสีดำ หรือจุดดำบนพื้นสีขาว ต่อเส้นสั่นที่ขาดออกจากกัน การทำให้ภาพลายนิ้วมือสองระดับบาง (Thinning) ให้เส้นสั่นเหลือความหนาเพียง 1 จุดภาพ

2.1.2.4 การสกัดลักษณะเด่น (Feature Extracting)

เป็นการสกัดลักษณะเฉพาะที่สำคัญที่ใช้ในการเปรียบเทียบลายนิ้วมือ โดยการตรวจหาเส้นสั้นเส้นร่อง และการตรวจหามินูเซีย

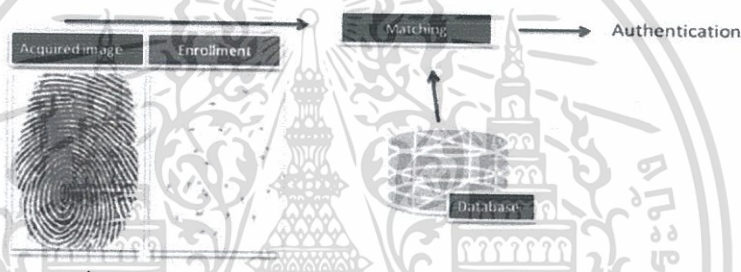
2.1.2.5 การเก็บรักษา (Storing)

ลักษณะเฉพาะที่ได้จะถูกเก็บไว้ในรูป แผ่นแบบลายนิ้วมือ (Fingerprint Template) ที่เก็บไว้ในฐานข้อมูล (Database) เพื่อเรียกใช้ต่อไปในอนาคต



รูปที่ 2.1 การลงทะเบียน (Registration) [15]

สำหรับการพิสูจน์ตัวจริงในการใช้งานระบบไบโอเมตริกหลายนิ้วมือ ดังแสดงในรูปที่ 2.2 โดยเป็นการใช้งานระบบไบโอเมตริกซึ่งต้องป้อนข้อมูลไบโอเมตริกเข้าระบบและผ่านกระบวนการเหมือนกับการลงทะเบียนที่กล่าวมาแล้ว เมื่อได้แผ่นแบบลายนิ้วมือนำเข้าแล้วก็พร้อมจะทำการจับคู่ (Matching) กับแผ่นแบบลายนิ้วมือที่อยู่ในฐานข้อมูล

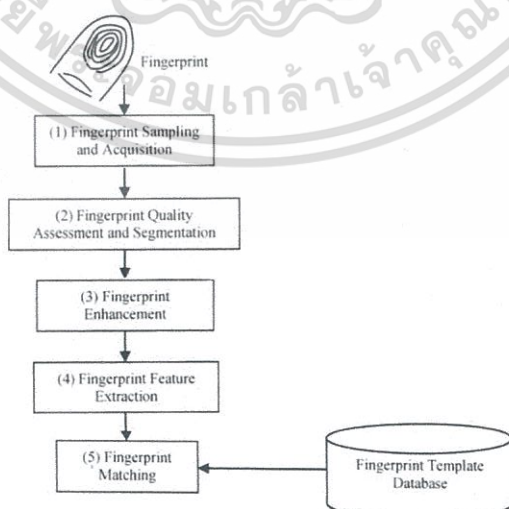


รูปที่ 2.2 การพิสูจน์ตัวจริง (Authentication) [15]

2.2 ระบบรู้จำลายนิ้วมือเบื้องต้น (Basic Fingerprint Recognition System)

ระบบรู้จำลายนิ้วมือพื้นฐานประกอบด้วยแผนภาพบล็อกกระบวนการต่างๆ ดังแสดงใน

รูปที่ 2.3



รูปที่ 2.3 แผนภาพบล็อกพื้นฐานของระบบรู้จำลายนิ้วมือเบื้องต้น [15]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การเก็บภาพลายนิ้วมือ (Fingerprint Acquisition)

เป็นการได้มาซึ่งภาพลายนิ้วมือ ซึ่งสามารถแยกประเภทการเก็บภาพลายนิ้วมือนำเข้าเป็น 2 แบบคือ แบบกราดตรวจสด (Live-Scan) และแบบไม่เชื่อมต่อตรง (Off-Line) โดยความแตกต่างกันคือ การเก็บภาพลายนิ้วมือแบบกราดตรวจสด ต้องนำนิ้วมือของบุคคลจริงมาแสดงอยู่หน้าอุปกรณ์รับรู้ลายนิ้วมือ (Fingerprint Sensor Device) แต่การเก็บภาพแบบไม่เชื่อมต่อตรง จะนำภาพลายนิ้วมือมาจากการพิมพ์หมึกบนกระดาษ จากการถ่ายภาพ หรือการเก็บลายนิ้วมือแฝงจากสถานที่เกิดเหตุ

2.2.1.1 การเก็บภาพลายนิ้วมือแบบกราดตรวจสด

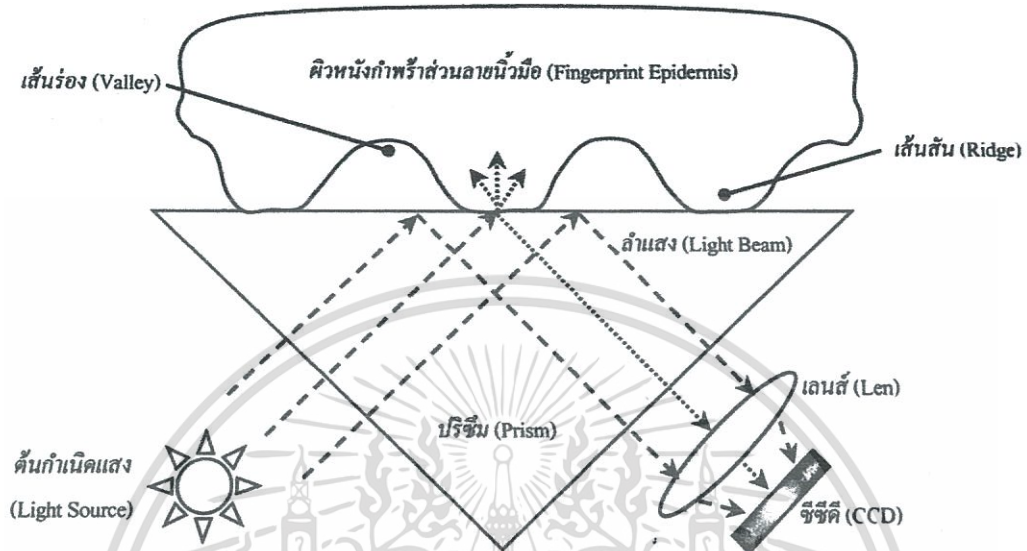
การเก็บภาพลายนิ้วมือแบบกราดตรวจสด (Live-Scan Fingerprint Acquisition) ซึ่งเป็นการเก็บลายนิ้วมือโดยตรงจากนิ้วมือที่มาแสดงหน้าตัวรับรู้ลายนิ้วมือ (Fingerprint Scanner) หรืออุปกรณ์กราดตรวจลายนิ้วมือ (Fingerprint Scanner Device) ซึ่งโดยทั่วไปแล้วต้องได้รับความร่วมมือหรือความสมัครใจจากเจ้าของลายนิ้วมือ โดยการเก็บลายนิ้วมือจะใช้อุปกรณ์อิเล็กทรอนิกส์ที่ใช้หลักการต่างๆ ทางฟิสิกส์ เพื่อเปลี่ยนสรีระของลายนิ้วมือในรูปแบบสามมิติให้กลายเป็นภาพลายนิ้วมอดิจิทัลสองมิติหรือสามมิติ โดยทั่วไปหลักการทางฟิสิกส์ที่ใช้ในการเก็บภาพลายนิ้วมือสามารถแบ่งแยกได้เป็น 6 ประเภท ดังต่อไปนี้

1. การเก็บลายนิ้วมือเชิงกล (Mechanical Fingerprint Acquisition)
2. การเก็บลายนิ้วมือเชิงแสง (Optical Fingerprint Acquisition)
3. การเก็บลายนิ้วมือเชิงไฟฟ้า (Electrical Fingerprint Acquisition)
4. การเก็บลายนิ้วมือเชิงไฟฟ้าแสง (Electro-Optical Fingerprint Acquisition)
5. การเก็บลายนิ้วมือเชิงคลื่นเหนือเสียง (Ultrasonic Fingerprint Acquisition)
6. การเก็บลายนิ้วมือเชิงอุณหภูมิ (Thermal Fingerprint Acquisition)

1) เทคนิคการสะท้อนภายในกลับหมด (Frustrated Total Internal Reflection (FTR) Technique) เป็นหนึ่งในเทคนิคการเก็บลายนิ้วมือเชิงแสง และเทคนิคการสะท้อนภายในกลับหมดยังเป็นเทคนิคดั้งเดิม และเป็นเทคนิคที่ใช้กันมากที่สุด ถูกนำเสนอโดย Hase และ Shimisu โดยมีส่วนประกอบหลักคือ ต้นกำเนิดแสง ปริซึมแก้ว และกล้องรับภาพ ซึ่งประกอบด้วยเลนส์และจอร์รับภาพที่เป็นอุปกรณ์ถ่ายเทประจุหรือซีซีดี (Charge-Couple Device (CCD)) ดังแสดงในรูปที่ 2.4 การทำงานสามารถอธิบายตามหลักการทางฟิสิกส์ของแสงที่ว่า การที่แสงผ่านเข้ามาในปริซึม และตกกระทบบริเวณรอยต่อระหว่างแก้วกับอากาศด้วยมุมที่กว้างกว่ามุมวิกฤต (Critical Angle) จะทำให้เกิดการสะท้อนกลับทั้งหมด (Total Reflection) ส่งผลให้แสงไม่สามารถทะลุผ่านปริซึมออกไปด้านนอกได้ แต่เมื่อมีผิวหนึ่งส่วนลายนิ้วมือมาสัมผัสบริเวณรอยต่อระหว่างแก้วกับอากาศ ซึ่งจะเปลี่ยนคุณสมบัติของตัวกลางจากอากาศเป็นผิวหนึ่ง ส่งผลให้แสงสามารถทะลุผ่านปริซึมไปและถูกดูดซับโดยผิวหนึ่งได้ ทำให้การสะท้อนกลับลดลง ซึ่งบริเวณลายนิ้วมือที่สัมผัสกับแก้วคือบริเวณเส้นสัน ดังนั้นบริเวณเส้นสันจะปรากฏเป็นเส้นทึบแสง และบริเวณเส้นร่องจะปรากฏ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นสว่างตามเดิม ทำให้กล้องรับภาพหรือซีซีดีที่ตั้งไว้ในมุมที่พอเหมาะจะสามารถรับภาพที่ทะลุผ่านปริซึมนี้ได้ดังแสดงดังรูปที่ 2.4



รูปที่ 2.4 เทคนิคการสะท้อนภายในกลับหมด [15]

2.2.2 การประเมินคุณภาพลายนิ้วมือและการตัดแยกลายนิ้วมือ (Fingerprint Quality Assessment and Segmentation)

การประเมินคุณภาพของภาพลายนิ้วมือกับการตัดแยกลายนิ้วมือออกจากพื้นหลังนั้น มีกระบวนการวิเคราะห์ที่ใกล้เคียงกันมาก โดยที่การประเมินคุณภาพลายนิ้วมือจะวิเคราะห์และระบุค่าคุณภาพของลายนิ้วมือออกมาเป็นตัวเลข เช่น คุณภาพของลายนิ้วมือมีค่า 0 ถึง 1 โดยที่ค่า 0 หมายถึงคุณภาพที่ต่ำที่สุด และ 1 หมายถึงคุณภาพที่สูงที่สุด เป็นต้น สำหรับการตัดแยกลายนิ้วมือออกจากพื้นหลังจะวิเคราะห์บริเวณเฉพาะที่ว่าเป็นลายนิ้วมือหรือพื้นหลัง โดยระบุว่าเป็นสถานะใดสถานะหนึ่งเท่านั้น ไม่สามารถเป็นสองสถานะได้ ซึ่งคล้ายกับการประเมินคุณภาพของภาพลายนิ้วมือแต่เป็นการแบ่งที่ชัดเจนและมีผลลัพธ์เป็นสองสถานะหรือสองระดับ คือผลลัพธ์เป็นบริเวณที่มีลายนิ้วมือหรือเป็นบริเวณที่ไม่มีลายนิ้วมือ

วิธีประเมินคุณภาพและการตัดแยกลายนิ้วมือสามารถแบ่งออกเป็น 2 ประเภทคือ วิธีประเมินคุณภาพและตัดแยกลายนิ้วมือในโดเมนปริภูมิ (Fingerprint Quality Assessment and Segmentation in Spatial Domain) และวิธีการประเมินคุณภาพและการตัดแยกลายนิ้วมือในโดเมนการแปลง (Fingerprint Quality Assessment and Segmentation in Transform Domain)

2.2.2.1 การประเมินคุณภาพและการตัดแยกลายนิ้วมือในโดเมนปริภูมิ

การประเมินคุณภาพและการตัดแยกลายนิ้วมือในโดเมนปริภูมิ

หมายถึงการนำลายนิ้วมือหรือบางส่วนของลายนิ้วมือมาทำการวิเคราะห์เพื่อประเมินคุณภาพเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และตัดแยก ซึ่งจะกระทำโดยตรงจากภาพลายนิ้วมือในโดเมนปริภูมิ ในกระบวนการตัดแยกลายนิ้วมือ จะทำการแยกแยะว่าส่วนใดถือเป็นลายนิ้วมือหรือที่เรียกว่าพื้นหน้า (Foreground) และส่วนใดไม่ใช่ลายนิ้วมือหรือที่เรียกว่าพื้นหลัง (Background) เช่นเดียวกัน ในกระบวนการประเมินคุณภาพลายนิ้วมือ นอกจากจะจำแนกว่าส่วนใดเป็นลายนิ้วมือและส่วนใดเป็นพื้นหลังแล้ว ส่วนที่เป็นลายนิ้วมือจะถูกประเมินต่อไปอีกเพื่อทำการจำแนกคุณภาพว่าส่วนใดมีคุณภาพสูง ส่วนใดมีคุณภาพรองลงมา และส่วนใดมีคุณภาพต่ำ โดยมีเป้าหมายคือทำให้ความน่าเชื่อถือหรือการระบุความมีเสถียรภาพของบริเวณต่างๆ ของลายนิ้วมือในกระบวนการประมวลผลลายนิ้วมือขั้นตอนต่อไป การตรวจหาลักษณะเด่นที่สำคัญจะถูกถ่วงน้ำหนักด้วยค่าคุณภาพที่ประเมินได้นี้

2.2.2.2 การประเมินคุณภาพและการตัดแยกลายนิ้วมือในโดเมนการแปลง

วิธีการประเมินคุณภาพและการตัดแยกลายนิ้วมือในโดเมนการแปลง หมายถึงใช้การเทคนิคแปลงภาพลายนิ้วมือไปอยู่ในโดเมนอื่นหรือรูปแบบการแทนอื่นก่อนแล้วทำการวิเคราะห์และแปลความหมายของสัมประสิทธิ์การแปลงเหล่านั้นหรือรูปแบบการแทนเหล่านั้นให้บ่งบอกถึงคุณภาพและตำแหน่งของส่วนที่เป็นลายนิ้วมือ โดยธรรมชาติแล้วลายนิ้วมือมีลักษณะของลายเส้นที่ขนานกันและโค้งไปมาในลักษณะที่มีรูปแบบเฉพาะ ซึ่งจะเห็นได้ว่าการแปลงภาพลายนิ้วมือไปโดเมนความถี่ โดเมนเวลา-ความถี่ หรือโดเมนอื่นๆ ที่สามารถทำให้ข้อมูลเส้นลายนิ้วมือกระจุกตัวง่ายต่อการประมวลผล ส่งผลให้สามารถประเมินคุณภาพหรือระบุตำแหน่งลายนิ้วมือได้ง่ายขึ้นกว่าการประเมินหรือระบุตำแหน่งในโดเมนปริภูมิ การประเมินคุณภาพและการตัดแยกลายนิ้วมือโดยวิธีการแปลงจะใช้ประโยชน์จากความเข้าใจในการแปลงแบบต่างๆ ที่เกี่ยวข้องกับความถี่และทิศทางของเส้นลายนิ้วมือที่สามารถสะท้อนถึงคุณภาพและตำแหน่งของลายนิ้วมือได้เป็นอย่างดี เพื่อให้ได้มาซึ่งดัชนีที่บ่งบอกคุณภาพของลายนิ้วมืออย่างมีประสิทธิภาพ วิธีประเมินคุณภาพและการตัดแยกภาพลายนิ้วมือ ยกตัวอย่างเช่น การใช้การแปลงฟูเรียร์ การใช้การแปลงกabor การใช้การแปลงเวฟเลต เป็นต้น

2.2.3 การปรับปรุงภาพลายนิ้วมือ (Fingerprint Enhancement)

เทคโนโลยีการปรับปรุงภาพลายนิ้วมือในอดีตจนถึงปัจจุบันได้มีการพัฒนาไปมาก โดยช่วงเริ่มต้นจะเน้นการแก้ไขปัญหาพื้นฐาน เช่น เส้นลายนิ้วมือขาดเป็นเส้นประที่เกิดจากความแห้ง หรือเส้นลายนิ้วมือที่หนาติดกันเป็นแถบของลายนิ้วมือที่เปียกและมีความชื้นมาก หรือเกิดจากการพิมพ์หมึกที่ใช้หมึกหนาเกินไป หรือพยายามแก้ไขบริเวณที่ลายนิ้วมือมีรอยแตกทำให้ขาดออกจากกัน จนถึงยุคที่เริ่มพัฒนาลายนิ้วมือที่บริเวณคุณภาพไม่ดีให้ดีขึ้นและมีความน่าเชื่อถือมากขึ้น ในปัจจุบันงานวิจัยมุ่งไปสู่การปรับปรุงภาพลายนิ้วมือแฝงและการสร้างกลับลายนิ้วมือที่ถูกทำลายจากสถานการณ์ต่างๆ

การปรับปรุงภาพลายนิ้วมือจะเป็นการปรับปรุงภาพลายนิ้วมือเพื่อจุดประสงค์ในการเพิ่มประสิทธิภาพในการรู้จำลายนิ้วมือของระบบระบุลายนิ้วมืออัตโนมัติ ซึ่งไม่ได้คำนึงถึงความเหมือนภาพต้นฉบับมากไปกว่าความแม่นยำในการเปรียบเทียบลายนิ้วมือที่สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การสกัดลักษณะเด่นจากภาพลายนิ้วมือ (Fingerprint Feature Extraction)

ลักษณะเด่นของลายนิ้วมือสามารถแบ่งออกเป็น 3 ระดับ ตามการแบ่งแยกลักษณะของลายนิ้วมือ คือ ลักษณะเด่นระดับที่ 1 เป็นลักษณะเด่นครอบคลุม ซึ่งครอบคลุมภาพรวมของลายนิ้วมือในระดับมหภาค รวมทั้งจุดเอกฐาน สนามทิศทาง และ สนามความถี่ ลักษณะเด่นระดับที่ 2 เป็นลักษณะเด่นเฉพาะที่ ซึ่งกำหนดให้เป็นเส้นสัน เส้นร่อง และจุดมินูเซียชนิดต่างๆ และสุดท้าย ลักษณะเด่นระดับที่ 3 เป็นลักษณะเด่นเฉพาะที่มาก ซึ่งหมายถึง รูขุมขน รอยแผลเป็น รอยแตก หูด หรือ ลักษณะเฉพาะอื่นๆ ที่ใช้ในการระบุความแตกต่างของลายนิ้วมือได้ แต่ที่ใช้กันมากในการเปรียบเทียบลายนิ้วมือคือระดับที่สอง

2.2.4.1 ลักษณะเด่นระดับที่สอง: ลักษณะเด่นเฉพาะที่ (Level-2 Features: Local Features)

ลักษณะเด่นระดับที่สอง หรือลักษณะเด่นเฉพาะที่ จะเป็นลักษณะเด่นที่มีความละเอียดมากกว่าลักษณะเด่นระดับที่หนึ่งที่มีเพียงจุดแยกฐานและสนามทิศทาง ลักษณะเด่นระดับที่สองจะประกอบไปด้วยโครงสร้างของเส้นลายนิ้วมือซึ่งแยกเป็นเส้นสัน (Ridge) ซึ่งเป็นส่วนที่นูนออกมาของลายนิ้วมือ และเส้นร่อง (Valley) ซึ่งเป็นส่วนที่เป็นร่องในลายนิ้วมือ โดยทั่วไปในการเก็บตัวอย่างภาพลายนิ้วมือนั้นเส้นสันจะมีสีเข้มและเส้นร่องจะมีสีอ่อน โดยเฉลี่ยแล้วเส้นสันนั้นจะมีความหนาตั้งแต่ 100 ไมโครเมตร ถึง 300 ไมโครเมตร และจะมีคาบระหว่างเส้นสันและเส้นร่องอยู่ประมาณ 500 ไมโครเมตร

การประมวลผลภาพลายนิ้วมือในส่วนสกัดลักษณะเด่นระดับที่สอง ส่วนใหญ่จะใช้ภาพลายนิ้วมือที่ผ่านการปรับปรุงแล้ว ซึ่งอัลกอริทึมการปรับปรุงภาพในยุคปัจจุบันส่วนใหญ่ จะปรับปรุงภาพระดับเทา (Gray-Scale Image) ของภาพลายนิ้วมุดันฉบับจนเกือบจะกลายเป็นภาพสองระดับ (Binary Image) ซึ่งหมายถึงภาพลายนิ้วมือระดับเทาที่ผ่านการปรับปรุงภาพแล้ว เส้นสันจะมีค่าเข้าใกล้ 0 ในขณะที่เส้นร่องจะมีค่าใกล้ 255 ในภาพระดับเทาขนาด 8 บิต ดังนั้นเมื่อนำภาพระดับเทานี้มาผ่าน การทำเป็นภาพสองระดับ (Binarization) ค่าขีดเริ่มเปลี่ยนที่ใช้จะไม่มีผลมากนักเนื่องจากผลลัพธ์ที่ได้ใกล้เคียงกัน

สำหรับการหาลักษณะเด่นเฉพาะที่ สามารถแบ่งตามลักษณะเด่นได้ 2 แบบ คือ การตรวจหาเส้นสันและเส้นร่อง และการตรวจหาจุดมินูเซีย รายละเอียดดังต่อไปนี้

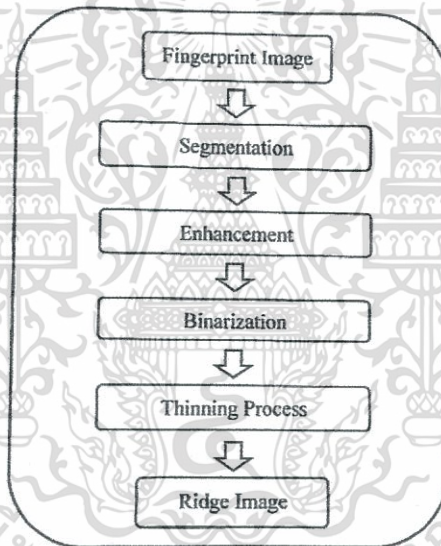
1) การตรวจหาเส้นสันและเส้นร่อง (Ridge and Valley Detection) การตรวจหาเส้นสันและเส้นร่องของลายนิ้วมือในปัจจุบันเป็นเรื่องที่ไม่ซับซ้อน เนื่องจากอัลกอริทึมการปรับปรุงภาพลายนิ้วมือที่สามารถทำงานได้เป็นอย่างดี จะทำการปรับปรุงเส้นสันและเส้นร่องให้เห็นอย่างเด่นชัดการใช้ค่าขีดเริ่มเปลี่ยน T_B ในการแบ่งระหว่างเส้นสันกับเส้นร่องจึงเป็นเรื่องง่าย และค่าขีดเริ่มเปลี่ยน T_B จะถูกตั้งไว้ที่ค่าเฉลี่ยของภาพได้ การทำให้เป็นภาพสองระดับ (Binarization) สามารถทำได้โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$I_B(x, y) = \begin{cases} 1 & \text{If } I_{Enh}(x, y) \geq T_B \\ 0 & \text{If } I_{Enh}(x, y) < T_B \end{cases} \quad (2.1)$$

โดยที่ $I_{Enh}(x, y)$ เป็นจุดภาพของลายนิ้วมือที่ผ่านการปรับปรุงภาพแล้วที่พิกัดตำแหน่ง (x, y) และ $I_B(x, y)$ คือจุดภาพของภาพสองระดับของลายนิ้วมือ

กระบวนการตรวจหาเส้นสันโดยทั่วไปสามารถแสดงได้ดังรูปที่ 2.5 เพื่อความสะดวกในการประมวลผลและการสกัดลักษณะเด่น เส้นสันในภาพลายนิ้วมือสองระดับจะถูกลดให้เหลือความหนาเพียง 1 จุดภาพ โดยใช้กระบวนการประมวลภาพมอฟอลอจิก (Morphologic Image Processing) ซึ่งใช้กระบวนการทำให้เส้นบาง (Thinning) ทำให้ได้โครงของเส้นสันทั้งภาพของลายนิ้วมือ ซึ่งเพียงพอต่อการนำไปจับคู่ลายนิ้วมือต่อไป และการตรวจหาเส้นร่องสามารถทำได้โดยวิธีเดียวกัน โดยการกลับสี จากสีขาวเป็นสีดำ และสีดำเป็นสีขาวที่เรียกว่า ภาพเชิงลบ (Negative Image) แล้วผ่านกระบวนการตรวจหาเส้นสันเหมือนเดิม ผลที่ได้จะเป็นเส้นร่องแทนที่จะเป็นเส้นสัน



รูปที่ 2.5 แผนภาพบล็อกกระบวนการตรวจหาเส้นสัน [15]

2) การตรวจหา minutiae จากภาพเส้นสันบาง (Minutiae Detection from Thinned Ridge Image) โดยทั่วไปแล้วลักษณะเด่นจุด minutiae จะถูกเก็บอยู่ในรูปของพิกัดตำแหน่งและมุมตามตัวอย่างดังแสดงในรูปที่ 2.6 ซึ่งเป็นการเข้ารหัสจากเส้นสัน (Ridge Encoding) โดยที่เส้นสีดำคือเส้นสันและเส้นสีขาวคือเส้นร่อง เมื่อเส้นสันแยกออกจากกัน จะถือเป็นจุดแยกดังรูปที่ 2.6(ข) โดยพิกัดตำแหน่งของจุด minutiae ทั้งสองอยู่ที่ตำแหน่ง (x, y) สำหรับมุม θ ของจุดแยกคือ มุมของเส้นก่อนแยกกระทำกับแกนนอนของภาพ และมุม ϕ ของจุดหยุดคือ มุมของเส้นที่ทำให้เกิดจุดหยุดอิงกับแกนนอนของภาพ



รูปที่ 2.6 การกำหนดตำแหน่งและมุมของมินูเซียแบบจุดแยกและจุดหยุด [15]

กระบวนการตรวจหาจุดมินูเซียนั้น เป็นขั้นตอนสำคัญต่อระบบรู้จำลายนิ้วมือ เนื่องจากความผิดพลาดจากขั้นตอนนี้จะส่งผลต่อประสิทธิภาพของระบบรู้จำลายนิ้วมือโดยรวม ในยุคแรกเริ่มนั้นจะหาจุดมินูเซียโดยตรงหลังการทำภาพลายนิ้วมือเป็นภาพสองระดับ แต่ในยุคหลัง จะนำภาพลายนิ้วมือสองระดับไปผ่านการประมวลภาพมอโฟลจิก (Morphology Image Processing) ก่อน เช่น การดำเนินการเปิด (Opening) หรือ การดำเนินการปิด (Closing) จะใช้เพื่อลดจุดโดดหรือจุดที่โดดเด่นออกมาเช่นเป็นจุดขาวบนพื้นสีดำ หรือจุดดำบนพื้นสีขาว นอกจากนี้การประมวลภาพมอโฟลจิกยังช่วยต่อเส้นสั้นที่ขาดออกจากกันในภาพสองระดับ จากนั้นจะนำภาพลายนิ้วมือสองระดับผ่านการทำให้เส้นบาง (Thinning) ให้เส้นสั้นเหลือความหนาเพียง 1 จุดภาพ และทำการตรวจหาจุดมินูเซียด้วยหน้าต่างขนาด 3×3 จุดภาพ โดยใช้การนับจำนวนจุดเชื่อมต่อกจุดกลาง และพิจารณาการวนรอบจุดกลางเพื่อระบุว่าเป็นมินูเซียชนิดใด ซึ่งวิธีนี้เป็นที่นิยมจนถึงปัจจุบัน ตามสมการดังนี้

$$cr(x, y) = \sum_{i=1}^8 p_i \text{ where}$$

$$p_1 = I_B(x - 1, y - 1), p_2 = I_B(x, y - 1),$$

$$p_3 = I_B(x + 1, y - 1), p_4 = I_B(x + 1, y),$$

$$p_5 = I_B(x - 1, y + 1), p_6 = I_B(x, y + 1),$$

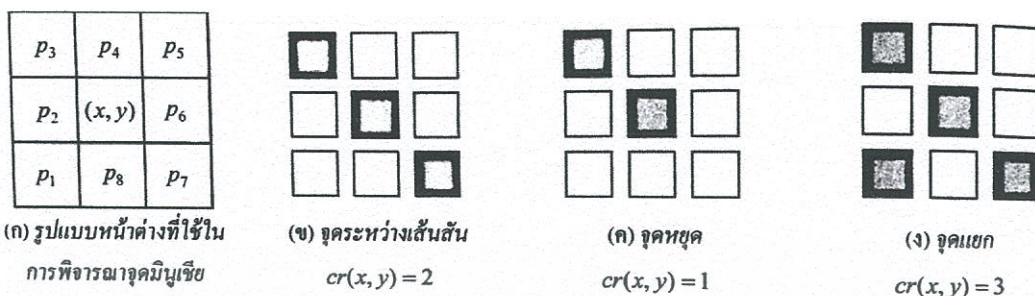
$$p_7 = I_B(x - 1, y + 1), p_8 = I_B(x - 1, y)$$
(2.2)

โดยที่ $cr(x, y)$ คือจำนวนจุดที่เชื่อมต่อกับจุดกลางที่พิกัดตำแหน่ง (x, y) และตำแหน่งของจุด p_i ดังแสดงในรูปที่ 2.7 (ก) โดยค่าของ $cr(x, y)$ จะเป็นตัวบอกประเภทของมินูเซียดังต่อไปนี้

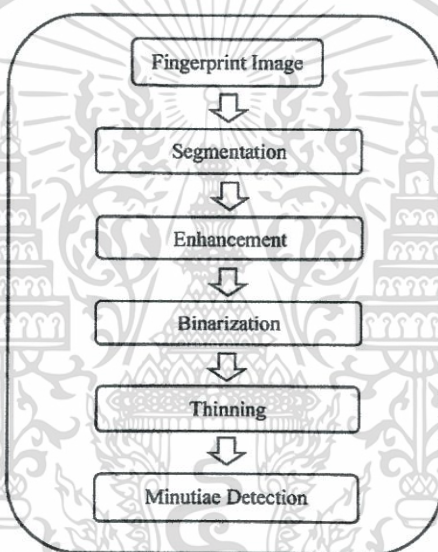
1. ถ้า $cr(x, y) = 0$ หมายถึงจุด (x, y) เป็นจุดโดด สามารถเอาออกไปได้
2. ถ้า $cr(x, y) = 1$ หมายถึงจุด (x, y) เป็นจุดมินูเซียประเภท จุดหยุด
3. ถ้า $cr(x, y) = 2$ หมายถึงจุด (x, y) เป็นจุดบนเส้นสั้น ไม่ใช่จุดมินูเซีย
4. ถ้า $cr(x, y) = 3$ หมายถึงจุด (x, y) เป็นจุดมินูเซียประเภท จุดแยก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ถ้า $cr(x, y) > 3$ หมายถึงจุด (x, y) เป็นจุดมินูเชียประเภทซับซ้อน อาจเป็นจุดกากบาท หรืออาจซับซ้อนกว่านั้น



รูปที่ 2.7 การตรวจหาจุดมินูเชียด้วยหน้าต่างขนาด 3x3 จุดภาพ [15]



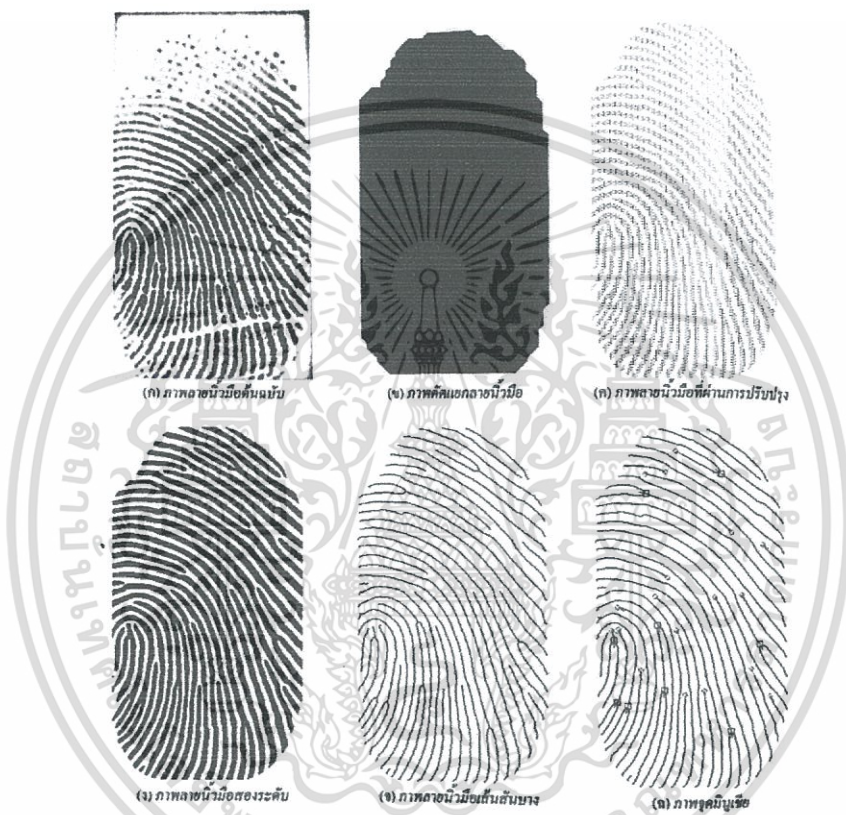
รูปที่ 2.8 แผนภาพบล็อกกระบวนการตรวจหาจุดมินูเชียจากภาพลายนิ้วมือนำเข้า [15]

เมื่อได้ทราบถึงขั้นตอนการหาตำแหน่งและประเภทของจุดมินูเชียแล้ว ต่อไปจะกล่าวถึงกระบวนการตรวจหาจุดมินูเชียจากภาพลายนิ้วมือ ซึ่งสามารถเขียนเป็นแผนภาพบล็อกได้ดังแสดงในรูปที่ 2.8 โดยแต่ละขั้นตอนมีการแจกแจงดังต่อไปนี้

1. การตัดแยกภาพลายนิ้วมือ (Fingerprint Segmentation) สามารถแสดงตัวอย่างผลการตัดแยกดังรูปที่ 2.9 (ข)
2. การปรับปรุงภาพลายนิ้วมือ (Fingerprint Enhancement) เป็นการปรับปรุงภาพลายนิ้วมือให้ดีขึ้นโดยมีการต่อเส้นที่ขาดและทำให้เส้นคม สามารถแสดงตัวอย่างของการปรับปรุงภาพดังรูปที่ 2.9 (ค)
3. การทำให้เป็นภาพสองระดับ (Fingerprint Binarization) เป็นการเปลี่ยนภาพระดับเทาที่ผ่านการปรับปรุงแล้วให้เป็นภาพสองระดับ โดยแสดงตัวอย่างผลการทำให้เป็นภาพสองระดับดังรูปที่ 2.9 (ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การทำให้เส้นสันบาง (Ridge Thinning) เป็นการทำให้เส้นสันบางเพียง 1 จุดภาพ โดยแสดงตัวอย่างผลการทำเส้นสันให้บางดังรูปที่ 2.9 (จ)
5. การตรวจจับจุดมินูเชีย (Minutiae Detection) เป็นการตรวจหาตำแหน่งของจุดมินูเชียพร้อมกับแบบของมินูเชียว่าเป็นจุดหยุดหรือจุดแยก โดยแสดงตัวอย่างผลการทำเส้นสันให้บางดังรูปที่ 2.9 (ฉ) โดยกำหนดให้สีเหลี่ยมแทนจุดแยก และวงกลมแทนจุดหยุด



รูปที่ 2.9 ผลลัพธ์จากที่ได้แต่ละขั้นตอนของการหาจุดมินูเชีย [15]

2.2.5 การจับคู่ลายนิ้วมือ (Fingerprint Matching)

ระบบการจับคู่ลายนิ้วมือ (Fingerprint Matching System) เป็นการนำลายนิ้วมือที่เรียกว่า ลายนิ้วมือนำเข้า (Input Fingerprint) ซึ่งเป็นลายนิ้วมือที่ได้รับจากบุคคลโดยตรง หรือเป็นลายนิ้วมือแฝงเพื่อที่จะนำเข้ามาในระบบมาทำการจับคู่กับลายนิ้วมือแม่แบบ (Template Fingerprint) ซึ่งเป็นลายนิ้วมือที่มีอยู่ในฐานข้อมูล โดยการหาค่าความเหมือน หรือค่าความแตกต่างที่เปลี่ยนออกมาเป็นคะแนนที่เป็นตัวบอกความเหมือนระหว่างสองลายนิ้วมือที่มาเปรียบเทียบกับกันนี้ เรียกว่า คะแนนความเหมือน (Similarity Score)

เนื่องจากระบบการจับคู่ลายนิ้วมือไม่สามารถควบคุมลายนิ้วมือนำเข้าได้อย่างสมบูรณ์ ดังนั้นระบบจะต้องมีความสามารถในการทนทานต่อการเลื่อน (Translation) และ การหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Rotation) ของลายนิ้วมือนำเข้า แต่ปัญหาที่ท้าทายของลายนิ้วมือนำเข้าคือความบิดเบี้ยวยืดหยุ่น (Elastic Distortion) ของนิ้วมือ ที่ทำให้การกดนิ้วแต่ละครั้งภาพลายนิ้วมือที่รับเข้ามาได้จะมีความแตกต่างกันตามแรงกด แรงบิด รวมทั้งบริเวณที่สัมผัส ทำให้อัลกอริทึมการจับคู่ลายนิ้วมือต้องทนทานต่อภาพลายนิ้วมือที่บิดเบี้ยวที่เกิดจากความยืดหยุ่นนี้ให้ได้ด้วย

การปรับตรง (Alignment) หมายถึงการทำให้ลายนิ้วมือนำเข้ากับลายนิ้วมือแผ่นแบบอยู่ในแนวทางเดียวกัน อาทิเช่น ตำแหน่งการเลื่อนและองศาการหมุน ถูกปรับให้อยู่ในแนวทางเดียวกัน ซึ่งจะทำให้สามารถเปรียบเทียบกันได้

ปัญหาของการจับคู่ลายนิ้วมือ นอกจากการเลื่อน การหมุน และการยืดหยุ่นแล้ว ยังอาจมีปัญหาการสเกล (Scaling) เนื่องจากการเจริญเติบโตของนิ้วจากวัยเด็กสู่วัยรุ่นและผู้ใหญ่ นอกจากนี้ยังมีปัญหาอื่นๆ อาทิเช่น ลายนิ้วมือมีคุณภาพต่ำ (Low Quality Fingerprint) ลายนิ้วมือแฝง (Latent Fingerprint) ลายนิ้วมือเพียงบางส่วน (Partial Fingerprint) ขนาดพื้นที่รับภาพของตัวรับรู้ที่แตกต่างกัน (Different Types of Fingerprint Sensors) ซึ่งปัญหาเหล่านี้ยังคงเป็นปัญหาที่ต้องการการแก้ไข การจับคู่ที่ใช้มากคือการจับคู่ลักษณะเด่นเฉพาะที่

2.2.5.1 การจับคู่ระดับที่สอง การจับคู่ลักษณะเด่นเฉพาะที่ (Level-2 Matching: Local Feature Matching)

ลักษณะเด่นระดับที่สองหรือลักษณะเด่นเฉพาะที่ของลายนิ้วมือ จะหมายถึงจุดมินูเซียและเส้นสัน ดังนั้นการจับคู่ลายนิ้วมือในระดับที่สองหมายถึงเป็นการจับคู่ลายนิ้วมือโดยใช้ลักษณะเด่นเฉพาะที่ อาทิเช่น จุดมินูเซียที่ใช้เพียงจุดแยกและจุดหยุดที่มีอัตลักษณ์เฉพาะตำแหน่ง หรือเส้นสันที่มีโครงสร้างเฉพาะตัว นอกจากนี้ การใช้เส้นสันร่วมกับจุดมินูเซียก็จะยังทำให้ลักษณะเด่นมีความชัดเจนและสามารถนำไปใช้ในการเปรียบเทียบรู้จำลายนิ้วมือได้อย่างมีประสิทธิภาพยิ่งขึ้น

2.3 Microsoft Visual C#

Visual C# (VC#) จัดเป็นเครื่องมือที่มีสภาพแวดล้อมที่รวมเครื่องมือใช้งาน (Integrated Development Environment - IDE) เพื่อสนับสนุนการพัฒนาแอปพลิเคชันได้รวดเร็ว (Rapid Application Development - RAD) ตัวหนึ่ง เป็นเครื่องมือที่ตัดส่วนมาเฉพาะภาษา C# (ซีชาร์ป) จากชุดเต็มของ Visual Studio

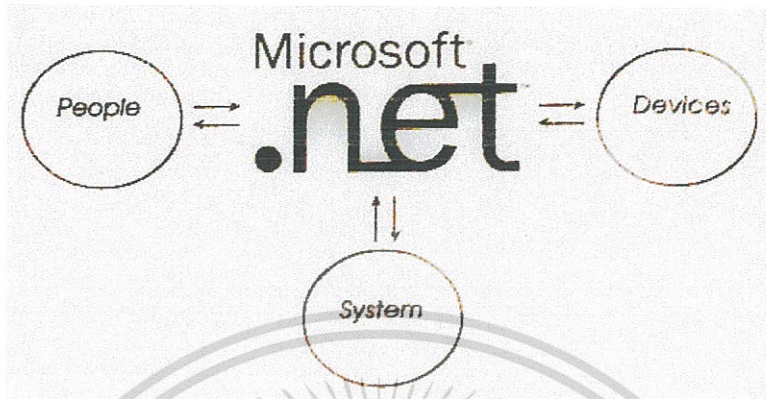
2.3.1 Microsoft .NET

เป็นซอฟต์แวร์ของไมโครซอฟต์ที่มีความสามารถในการรับส่งข้อมูลระหว่างมนุษย์ (People) ระบบ (System) และอุปกรณ์ (Devices) โดยทั้งหมดจะสามารถสื่อสารกันรู้เรื่องผ่านตัว .NET ดังรูปที่ 2.10

.NET จะใช้ XML เป็นพื้นฐานในการรับส่งข้อมูล ดังนั้นไม่ว่าระบบจะใช้ระบบปฏิบัติการวินโดวส์หรือไม่ .NET ก็สามารถรับและส่งข้อมูลได้เข้าใจ ยกตัวอย่างเช่น ถ้าเราใช้ Microsoft Office เราสามารถติดต่อกับโปรแกรมอื่นในระบบปฏิบัติการอื่นได้โดยผ่านตัว .NET

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

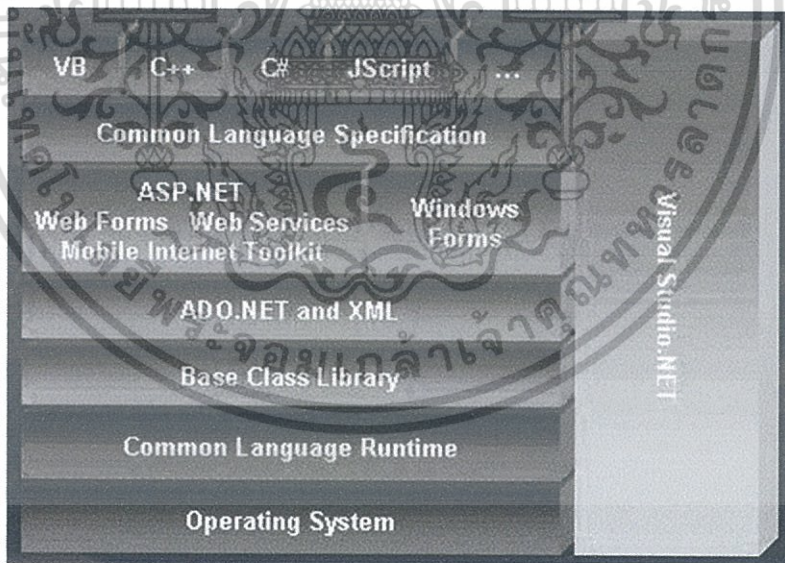
.NET จะทำให้การสื่อสารทำได้ง่ายขึ้น และจะเป็นศูนย์กลางในการเชื่อมโยงทุกระบบเข้าไว้ด้วยกัน และสามารถพูดคุยกันได้อย่างเข้าใจ



รูปที่ 2.10 การสื่อสารผ่านตัว.NET [25]

2.3.2 Microsoft .NET Framework

Microsoft .NET Framework คือเฟรมเวิร์กที่มีความสามารถของ .NET สามารถเชื่อมโยงโปรแกรมต่างๆ ให้สื่อสารเข้าใจกันได้ ส่วนประกอบของเฟรมเวิร์กเป็นดังรูปที่ 2.11 โดยมีส่วนประกอบหลักๆ ดังนี้



รูปที่ 2.11 ส่วนประกอบของเฟรมเวิร์ก [25]

2.3.2.1 Windows Forms

เป็นองค์ประกอบพื้นฐานของวินโดวส์ เช่น คอนโทรลพื้นฐาน คำสั่งพื้นฐาน โปรแกรมสำเร็จรูปที่พัฒนาบน .NET Framework จะเรียกใช้องค์ประกอบพื้นฐานนี้เหมือนกัน ทำให้โปรแกรมสำเร็จรูปมีขนาดเล็กลงเพราะไม่ต้องเก็บส่วนประกอบพื้นฐานไว้ในตัวโปรแกรม แต่เรียกดึงจากตัว .NET Framework แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาค้นคว้าเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2 ASP .NET

เป็นองค์ประกอบด้านแอปพลิเคชัน ใช้ในการสร้าง เผยแพร่ และประมวลผลเว็บ ซึ่งสามารถแบ่งได้เป็น 2 ส่วน

1) Web Forms คือองค์ประกอบพื้นฐานบนหน้าเว็บ เช่น เท็กซ์บ็อกซ์ (Textbox) ที่เราใช้กรอกข้อมูล หรือปุ่มกด (Submit button) เป็นต้น Web Forms สนับสนุนการทำงานของเบราว์เซอร์หลายๆ ชนิดด้วย และยังสามารถที่จะปรับแต่งผลลัพธ์ เพื่อใช้ความสามารถเฉพาะของเบราว์เซอร์แต่ละตัวได้ Web Forms จะทำงานได้กับเบราว์เซอร์อย่าง IE 5.5 หรือสูงกว่าที่สนับสนุน DHTML และยังใช้กับเบราว์เซอร์ความสามารถน้อยที่มีในเครื่อง Palm หรือ PocketPC ได้ Web Forms จะปรับตัวเองให้ทำงานอย่างเหมาะสมสำหรับอุปกรณ์แต่ละชนิด และ Web Forms สามารถสร้างโดยใช้ภาษาใดก็ได้ที่ .NET สนับสนุน ทำให้สามารถสร้างเว็บแอปพลิเคชันด้วยภาษาที่ถนัดได้

2) Web Services คือบริการ (Service) บนอินเทอร์เน็ตที่เราสามารถเรียกใช้ได้ บริการนี้จะใช้มาตรฐานที่กำหนดไว้ เช่น XML, SOAP และ WSDL ในการให้บริการและติดต่อสื่อสาร Web Services สามารถเรียกได้จาก Windows Forms หรือ Web Forms นั้นหมายความว่าเราสามารถเขียนโปรแกรมเพื่อเรียกใช้บริการ Web Services ของผู้อื่นได้ โดยตัว Web Service จะใช้ภาษา XML เป็นหลักในการติดต่อสื่อสาร

2.3.2.3 .NET Framework Base Class Libraries

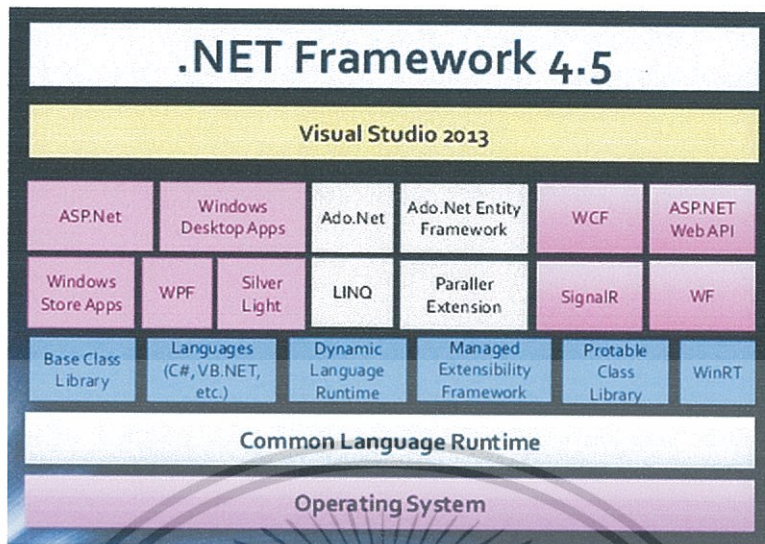
เป็นคลาสพื้นฐานที่ใช้ในการเขียนโปรแกรมใน .NET Framework จะจัดคลาสที่มีการทำงานคล้ายๆ กันเป็นกลุ่มเดียวกัน เรียกว่า เนมสเปซ (Namespace) โดยเนมสเปซของคลาสพื้นฐานจะขึ้นต้นด้วยคำว่า System ดังตัวอย่างคลาสต่างๆ ในตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างเนมสเปซของคลาสพื้นฐาน

Namespace	รายละเอียด
System.Data	เป็นคลาสที่ใช้ในการติดต่อกับฐานข้อมูล เช่น DataSet
System.Diagnostics	เป็นคลาสที่ใช้สำหรับแก้จุดบกพร่องของแอปพลิเคชัน และตรวจสอบการทำงานของคำสั่ง
System.IO	เป็นคลาสที่ใช้ในการอ่านเขียนแฟ้มข้อมูล เช่น แฟ้มข้อมูล, StreamReader
System.Math	เป็นคลาสที่ใช้คำนวณค่าต่างๆ ทางคณิตศาสตร์ เช่น การหาค่าราก

สำหรับโปรแกรม Visual Studio 2013 จะมาพร้อมกับ .NET Framework 4.5 โดยมีส่วนประกอบดังรูปที่ 2.12 โครงสร้าง .NET ส่วนบนสุดจะเป็น Development Framework แบ่งออกเป็น 3 ส่วนคือ ส่วนติดต่อกับผู้ใช้ ส่วนการใช้ข้อมูล และ ส่วนบริการ

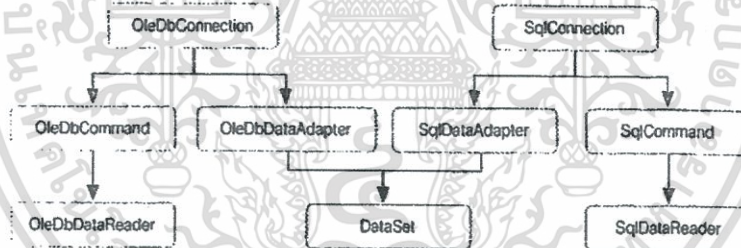
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 .NET Framework 4.5

2.3.2.4 กลุ่มออบเจ็กต์ ADO.NET

เป็นกลุ่มออบเจ็กต์สำหรับทำงานกับฐานข้อมูลซึ่งอยู่ในส่วนการใช้ข้อมูลของ .NET Framework กลุ่มออบเจ็กต์หลักของสถาปัตยกรรม ADO.NET แสดงดังรูปที่ 2.13



รูปที่ 2.13 ความสัมพันธ์ระหว่างกลุ่มออบเจ็กต์ ADO.NET [21]

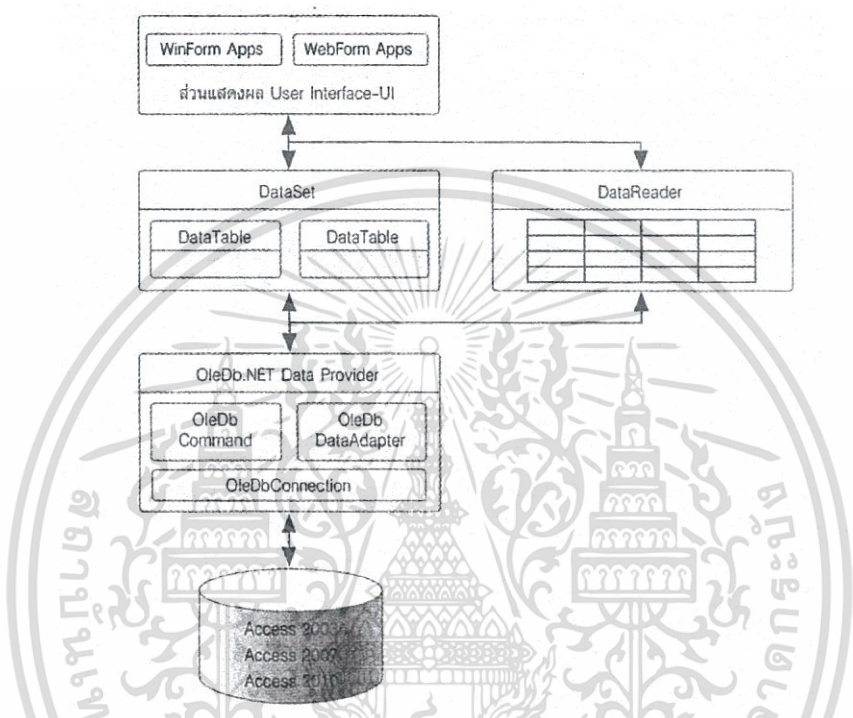
จากรูปที่ 2.13 หน้าที่ของออบเจ็กต์แต่ละตัวมีรายละเอียดดังนี้

- 1) กลุ่มออบเจ็กต์ที่อาศัย OLEDB Data Provider ทำหน้าที่เข้าถึงข้อมูลในฐานข้อมูล มี 3 ตัวคือ ออบเจ็กต์ OleDbConnection, OleDbDataAdapter และออบเจ็กต์ OleDbCommand ใช้งานร่วมกับฐานข้อมูลชนิดแอกเซส
- 2) กลุ่มออบเจ็กต์ที่อาศัย SQL Server Data ทำหน้าที่เข้าถึงข้อมูลในฐานข้อมูล มี 3 ตัวคือ ออบเจ็กต์ SqlConnection, SqlDataAdapter และออบเจ็กต์ SqlCommand ใช้งานร่วมกับฐานข้อมูลชนิด SQL Server
- 3) กลุ่มออบเจ็กต์ที่ใช้เก็บผล กลุ่มออบเจ็กต์ที่ใช้เก็บผล มี 3 ตัวคือ ออบเจ็กต์ DataSet, OleDbDataReader และออบเจ็กต์ SqlDataReader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.5 กลุ่มออบเจ็กต์ที่อาศัย OLEDB Data Provider

ประกอบไปด้วยกลุ่มออบเจ็กต์ที่ขึ้นต้นด้วยคำว่า OleDb เช่น OleDbConnection, OleDbDataAdapter ฯลฯ ใช้งานร่วมกับฐานข้อมูลชนิดแอคเซส ดังรูปที่ 2.14 เมื่อต้องใช้งานกลุ่มออบเจ็กต์ OLEDB Data Provider จะต้องเพิ่มนามสเปซ System.Data.OleDb นี้ก่อนเสมอ



รูปที่ 2.14 แบบจำลองของกลุ่มออบเจ็กต์ OleDb [21]

หน้าที่การทำงานหลักของออบเจ็กต์ ADO.NET แต่ละตัวแสดงดังตารางที่ 2.2 ตารางที่ 2.2 หน้าที่ของออบเจ็กต์ ADO.NET

ออบเจ็กต์	หน้าที่
ออบเจ็กต์ Connection	ทำหน้าที่สร้างการเชื่อมต่อเข้ากับฐานข้อมูล โดยที่ OleDbConnection เชื่อมต่อกับฐานข้อมูลชนิดแอคเซส ส่วน SqlConnection ทำหน้าที่เชื่อมต่อกับฐานข้อมูลชนิด SQL Sever
ออบเจ็กต์ DataAdapter	ทำหน้าที่เก็บชุดคำสั่ง SQL เพื่อคิวรีข้อมูลออกจากฐานข้อมูล ผลลัพธ์ที่ได้คือ ออบเจ็กต์ DataSet
ออบเจ็กต์ DataSet	ทำหน้าที่เก็บผลลัพธ์จากการทำคิวรีข้อมูลจากฐานข้อมูล ที่ได้จากออบเจ็กต์ DataAdapter เป็นออบเจ็กต์หลักที่มีบทบาทและมีความสำคัญมากที่สุด
ออบเจ็กต์ Command	ทำหน้าที่ดำเนินงานชุดคำสั่ง SQL ผลลัพธ์ที่ได้คือ ออบเจ็กต์ DataReader
ออบเจ็กต์ DataReader	ทำหน้าที่เก็บผลลัพธ์ของการทำคิวรีข้อมูลจากฐานข้อมูล เช่นเดียวกับ ออบเจ็กต์ DataSet แต่จะได้มาจากออบเจ็กต์ Command

2.4 ไมโครซอฟต์แอคเซส (Microsoft Access)

ไมโครซอฟต์แอคเซส (Microsoft Access) เรียกสั้นๆว่า แอคเซส เป็นโปรแกรมทางด้านการจัดการข้อมูล (Database Management) ที่มีขีดความสามารถสูงแต่ใช้งานง่าย ผู้ใช้สามารถใช้แอคเซสนี้ในการจัดเก็บข้อมูล, ค้นหาข้อมูล และนำข้อมูลเหล่านั้นมาแสดงในรูปแบบที่สวยงาม หรือจัดพิมพ์เป็นรายงานได้อย่างง่ายดาย

แอคเซส เป็นโปรแกรมจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งผู้ใช้จะทำงานร่วมกับฐานข้อมูลในรูปตาราง (Table)

2.4.1 คำศัพท์ที่ควรทราบ

คำศัพท์เฉพาะซึ่งเป็นพื้นฐานหลักๆที่เกี่ยวข้องกับฐานข้อมูลเชิงสัมพันธ์ คือ Entity, Relation, Tuple, Attribute, Key และ Relationship

2.4.1.1 Entity

เอนทิตี หมายถึง กลุ่มข้อมูลประเภทเดียวกันที่จะนำมาเก็บในฐานข้อมูลนั้น

2.4.1.2 Attribute

แอตทริบิวต์ หมายถึง สิ่งที่ใช้บอกองค์ประกอบหรือเนื้อหา (Subject) ของเอนทิตี

2.4.1.3 Relation

รีเลชันในที่นี้เป็นคำศัพท์ทางคณิตศาสตร์ที่หมายถึงรูปแบบของตารางแบบ 2 มิติที่ประกอบด้วยแต่ละแถวที่เรียกว่า ทูเพิล (Tuple) และแต่ละคอลัมน์เรียกว่า แอตทริบิวต์ (Attribute)

เนื้อหาต่อจากนี้คำว่า ตาราง เรคอร์ด และฟิลด์ ในฐานข้อมูลเชิงสัมพันธ์ที่สร้างจากแอคเซส จะมีความหมายเดียวกับ รีเลชัน ทูเพิล และแอตทริบิวต์ ตามลำดับ

2.4.1.4 Primary Key

คีย์หลัก เป็นคีย์ที่กำหนดจากฟิลด์ที่จะต้องไม่มีข้อมูลซ้ำกันเลย (Unique) ในฟิลด์นั้น เราจะนำคีย์หลักไปใช้จัดเรียงและจำแนกข้อมูลแต่ละเรคอร์ดออกจากกัน โดยฟิลด์ที่เป็นคีย์หลักต้องมีค่าเสมอ เป็นฟิลด์ว่าง (Null) ไม่ได้

2.4.1.5 Relationship

1) ความสัมพันธ์แบบ One-to-One (1:1) ความสัมพันธ์ที่เรคอร์ดใดๆ ในตารางหนึ่ง สามารถจับคู่กับเรคอร์ดในอีกตารางหนึ่งได้เพียงเรคอร์ดเดียวเท่านั้น

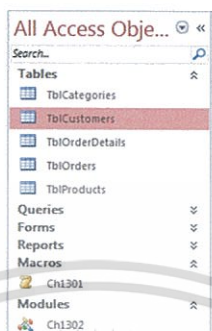
2) ความสัมพันธ์แบบ One-to-Many (1:N) ความสัมพันธ์ที่เรคอร์ดใดๆ ในตารางหนึ่ง สามารถจับคู่กับเรคอร์ดในอีกตารางหนึ่งได้หลายๆ เรคอร์ดพร้อมกัน

3) ความสัมพันธ์แบบ Many-to-Many (M:N) ความสัมพันธ์ที่หลายๆ เรคอร์ดในตารางหนึ่ง สามารถจับคู่กับเรคอร์ดในอีกตารางหนึ่งได้หลายๆ เรคอร์ดพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ส่วนประกอบของฐานข้อมูลแอคเซส

เมื่อเราสร้างฐานข้อมูล ภายในฐานข้อมูลจะเก็บส่วนประกอบที่เรียกว่า วัตถุ (Object) ต่างๆเอาไว้ดังรูปที่ 2.15



รูปที่ 2.15 วัตถุฐานข้อมูล (Database Object) [24]

สำหรับวัตถุแต่ละอย่างมีรายละเอียดดังนี้

2.4.2.1 Tables

ส่วนตาราง เป็นส่วนที่เก็บโครงสร้างของฐานข้อมูล และข้อมูลต่างๆ ที่เรามี เช่น ตารางลูกค้าเก็บข้อมูลเกี่ยวกับชื่อและที่อยู่ลูกค้าแต่ละรายไว้ ตารางจะเก็บข้อมูลในรูปแบบแถวและคอลัมน์ โดยข้อมูลในแต่ละแถวจะเรียกว่า เรคอร์ด (Record) ซึ่งจะเป็นข้อมูลลูกค้าแต่ละราย และข้อมูลในแต่ละคอลัมน์จะเรียกว่า ฟิลด์ (Field) เช่น ในตารางลูกค้า จะมีฟิลด์รหัส ชื่อ และที่อยู่ของลูกค้า เป็นต้น ดังรูปที่ 2.16

ชื่อลูกค้า	รหัสเขตที่อยู่	รหัสไปรษณีย์	Click to Add
91 โครงการมั่นคง	18/18 ถนนจันทนา	10640	
92 ห้างสรรพสินค้า	100/6 ถนนสุขุมวิท	10260	
94 บริษัทแห่ง	1/3 ถนนการค้า	10110	
95 NORTHBRIDGE	69/428 หมู่8	10400	
96 ร้านลาบ	1770 ถนนเพชรบุรี	10900	
		0	

รูปที่ 2.16 ตารางลูกค้า [24]

2.4.2.2 Queries

ส่วนคิวรี เป็นเครื่องมือที่ดึงข้อมูลดิบที่เก็บในตารางออกมาใช้อย่างเป็นระเบียบ โดยเราสามารถเรียกดูข้อมูลทั้งหมด หรือเฉพาะข้อมูลเจาะจง และสามารถจัดเรียงข้อมูลจากมากไปน้อย จากน้อยไปมาก รวมกลุ่มข้อมูลและสามารถหาค่ารวม (Summary) ของข้อมูลแต่ละกลุ่มได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.3 Form

ส่วนฟอร์ม เป็นการสร้างหน้าจอเพื่อใช้กรอกข้อมูลลงตาราง

2.4.2.4 Report

ส่วนรายงานเป็นเครื่องมือที่ใช้แสดงผลข้อมูลในฐานข้อมูลออกมา เช่น การสร้างรายงานที่ประกอบไปด้วยกราฟและรูปภาพเพื่อความน่าสนใจในรายงาน

2.4.2.5 Macro

ส่วนแมโครเป็นคำสั่งต่างๆ ที่เราเขียนเพื่อให้ทำงานอัตโนมัติ

2.4.2.6 Module

ส่วนโมดูล มีหน้าที่เหมือนกับแมโคร แต่เราสามารถควบคุมการทำงานได้มากกว่า โมดูลจะเป็นการเขียนคำสั่งโปรแกรมที่เราเรียกว่า VBA (Visual Basic for Application)

2.4.3 การสร้างและการกำหนดคุณสมบัติของฟิลด์

การสร้างแต่ละฟิลด์ขึ้นมาในมุมมอง Table Design โดยใส่ชื่อฟิลด์และกำหนดคุณสมบัติพื้นฐานของฟิลด์ คือ ชนิดข้อมูล (Data Type) ดังตารางที่ 2.3 และขนาดของฟิลด์ (Field Size) กำหนดคุณสมบัติต่างๆ ก็เพื่อจุดประสงค์ดังนี้

2.4.3.1 เพื่อให้มีประสิทธิภาพในการทำงาน

ทั้งด้านเนื้อที่ใช้ในการเก็บข้อมูล และความเร็วในการทำงานกับข้อมูล เช่น ข้อมูลชนิดข้อความโดยทั่วไปจะเสียเนื้อที่ในการเก็บมากกว่าข้อมูลชนิดตัวเลข และเลขจำนวนเต็มจะใช้เวลาในการคำนวณน้อยกว่าเลขทศนิยม เป็นต้น

2.4.3.2 เพื่อให้ตรงกับจุดประสงค์ในการใช้ข้อมูล

เช่น ฟิลด์ราคาสินค้าจะต้องเป็นข้อมูลชนิดตัวเลขที่ใช้ในการคำนวณได้ ไม่ใช่เป็นข้อมูลชนิดข้อความ ซึ่งนอกจากคำนวณไม่ได้แล้ว ยังสิ้นเปลืองเนื้อที่ในการเก็บข้อมูลอีกด้วย

คุณสมบัติ Field Size จะใช้กำหนดขนาดของฟิลด์ ซึ่งอยู่ในส่วน Field Properties คุณสมบัตินี้จะมีในข้อมูลชนิด Text และ Number เท่านั้น สำหรับข้อมูลชนิด Text จะเป็นการกำหนดจำนวนตัวอักษรที่สามารถเก็บได้สูงสุด ส่วนข้อมูลชนิด Number จะให้เราเลือกชนิดข้อมูลย่อย ดังตารางที่ 2.4

นอกจากนี้ เรายังจะต้องกำหนดคุณสมบัติเพิ่มเติมให้กับฟิลด์ด้วย ดังรูปที่ 2.17 เป็นตัวอย่างในการสร้างฟิลด์ ProductName ที่เป็นชื่อสินค้า เราจะกำหนดชนิดข้อมูลเป็น Short Text และ Field Size = 50 หมายความว่าเก็บข้อมูลได้ 50 อักขระ

Field Name	Data Type	Description (Optional)
ProductID	AutoNumber	รหัสสินค้า
CategoryID	Number	รหัสชนิดสินค้า
ProductName	Short Text	ชื่อสินค้า
UnitPrice	Currency	ราคาสินค้า (บาท)
VATFlag	Yes/No	สินค้าต้องรวมภาษีด้วยหรือไม่
UnitInStock	Number	จำนวนที่เหลือในคลังสินค้า

Field Properties

General	Lookup
Field Size	50
Format	
Input Mask	
Caption	ชื่อสินค้า
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	No
Indexed	Yes (Duplicates OK)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

The data type determines the kind of values that users can store in the field. Press F1 for help on data types.

รูปที่ 2.17 ตัวอย่างในการสร้างฟิลด์ [24]

ตารางที่ 2.3 ชนิดข้อมูล (Data Type) ที่มีในแอคเซส

ชนิดข้อมูล	คำอธิบาย
Short Text	เป็นข้อมูลชนิดข้อความที่เก็บตัวอักษรได้ไม่เกิน 255 ตัว โดยจำนวนตัวอักษรที่สามารถเก็บได้สูงสุดจะต้องกำหนดในคุณสมบัติ field Size
Long Text	เป็นข้อมูลชนิดข้อความเช่นเดียวกับ Short Text แต่สามารถเก็บตัวอักษรได้ไม่เกิน 65,535 ตัว
Number	เป็นข้อมูลชนิดตัวเลขที่สามารถกำหนดให้เป็นเลขจำนวนเต็ม หรือเลขทศนิยมก็ได้ โดยจะกำหนดในคุณสมบัติ Field Size
Date/Time	เป็นข้อมูลชนิดเวลาและวันที่
Currency	เป็นข้อมูลชนิดตัวเลขทศนิยมที่มีตำแหน่งหลังจุดทศนิยม 4 ตำแหน่งเสมอ จึงเหมาะที่จะใช้เก็บค่าเงินที่ต้องการความถูกต้องแม่นยำ
AutoNumber	เป็นข้อมูลชนิดตัวเลขที่จะกำหนดค่าให้แต่ละเรคอร์ดที่เพิ่มเข้าไปในตารางอัตโนมัติ โดยอาจจะเพิ่มค่า หรือสุ่มเอา (แล้วแต่จะกำหนดคุณสมบัติ New Values) ชนิดข้อมูลชนิดนี้เหมาะสำหรับสร้างฟิลด์ที่เป็นคีย์หลัก เพราะระบบจะเติมค่าที่ไม่ซ้ำกันให้เอง และไม่อนุญาตให้ผู้แก้ไข หรือเปลี่ยนค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ชนิดข้อมูล (Data Type) ที่มีในแอคเซส (ต่อ)

ชนิดข้อมูล	คำอธิบาย
Yes/No	เป็นข้อมูลชนิดบูลีน ที่มีเพียงค่าใดค่าหนึ่งเท่านั้นเราสามารถกำหนดรูปแบบในการแสดงผลเป็น True/False, Yes/No หรือ On/Off ก็ได้
OLE Object	เป็นข้อมูลชนิดออบเจกต์ เช่น รูปภาพ เสียง หรือออบเจกต์ที่สร้างจากโปรแกรมที่สนับสนุน OLE ก็ได้
Hyperlink	เป็นข้อมูลที่อยู่ของแฟ้มข้อมูลในระบบเครือข่ายอินเทอร์เน็ต หรือ อินทราเน็ต
Attachment	เป็นข้อมูลที่เก็บแฟ้มข้อมูลแนบ เช่น แฟ้มข้อมูลภาพ หรือ แฟ้มข้อมูลเอกสาร สามารถเก็บแฟ้มข้อมูลแนบได้พร้อมกันหลายๆ แฟ้มข้อมูล
Calculated	เป็นข้อมูลที่เก็บสมการ ใช้ในการสร้างฟิลด์ที่เก็บผลจากการคำนวณ เช่น [Quantity]*[Unit Price]
Lookup Wizard	เก็บข้อมูลที่เชื่อมโยงมาจากตารางอื่น ให้สร้างฟิลด์ Lookup (ฟิลด์ที่เชื่อมโยงมาจากตารางอื่น)

ตารางที่ 2.4 คุณสมบัติ Field Size ของฟิลด์ที่มีชนิดข้อมูลเป็น Number

ชนิดข้อมูลย่อย	รายละเอียด
Byte	เก็บเลขจำนวนเต็มที่มีค่าตั้งแต่ 0 ถึง 255 (ใช้พื้นที่ 1 ไบต์)
Integer	เก็บเลขจำนวนเต็มที่มีค่าตั้งแต่ -32,768 ถึง +32,767 (ใช้พื้นที่ 2 ไบต์)
Long Integer	เก็บจำนวนเต็มที่มีค่าตั้งแต่ -2,147,483,648 ถึง +2,147,483,647 (ใช้พื้นที่ 4 ไบต์)
Single	เก็บเลขทศนิยมที่มีค่าตั้งแต่ -3.4×10^{38} ถึง $+3.4 \times 10^{38}$ (ใช้พื้นที่ 4 ไบต์)
Double	เก็บเลขทศนิยมที่มีค่าตั้งแต่ -1.797×10^{308} ถึง $+1.797 \times 10^{308}$ (ใช้พื้นที่ 8 ไบต์)
Replication ID	เก็บค่า GUID (Globally Unique Identifier) ซึ่งเป็นตัวเลขสุ่มเทียม (pseudo-random) ที่มีโอกาสซ้ำกันน้อยมาก (ใช้พื้นที่ 16 ไบต์)
Decimal	เก็บเลขที่มีค่าตั้งแต่ -9.9×10^{27} ถึง $+9.9 \times 10^{27}$ (ใช้พื้นที่ 12 ไบต์)

อีกคุณสมบัติคือ คุณสมบัติ Format สำหรับการกำหนดรูปแบบการแสดงผลของชนิดข้อมูลในฟิลด์ที่เลือกอยู่ โดยจะมีรูปแบบให้เลือกแตกต่างกันไปตามชนิดข้อมูล ในที่นี้ขอยกตัวอย่างคุณสมบัติ Format ของฟิลด์ที่มีชนิดข้อมูลเป็น Date/Time ดังตารางที่ 2.5 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 คุณสมบัติ Format ของฟิลด์ที่มีชนิดข้อมูลเป็น Date/Time

รูปแบบข้อมูล	การแสดงผล
General Date	12/11/2558
Long Date	12 พฤศจิกายน 2558
Medium Date	12-พ.ย.-58
Short Date	12/11/2558
Long Time	17:34:23
Medium Time	5:34 PM
Short Time	17:34

2.5 Arduino

Arduino เป็น Microcontroller ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือ มีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้อย่างสามารถดัดแปลงเพิ่มเติมพัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ดหรือเพื่อความสะดวกสามารถต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาต่อกับด้านบนของบอร์ด Arduino แล้วจึงเขียนโปรแกรมพัฒนาต่อได้

2.5.1 Arduino Uno R3

Arduino Uno R3 แสดงได้ดังรูปที่ 2.18 เป็น Microcontroller แบบ Open Source ถูกออกแบบมาให้ใช้งานง่าย ใช้ชิป ATmega328 โดยจะทำงานอยู่ที่ความถี่ 16 MHz , หน่วยความจำแฟลช 32 kB, แรม 2 kB, บอร์ดใช้ไฟเลี้ยง 7 ถึง 12 โวลต์ มีระดับแรงดันไฟฟ้าในการทำงานและขั้วสัญญาณอยู่ที่ 5 โวลต์ (TTL) มี Digital Input/Output 14 ขา ซึ่งใช้เป็น PWM ได้ 6 ขา มี Analog Input 6 ขา, Serial UART 1 ชุด, I2C 1 ชุด และ SPI 1 ชุด เขียนโปรแกรมบนซอฟต์แวร์ Arduino IDE และเชื่อมต่อผ่านพอร์ต USB เหมาะสำหรับผู้สนใจเริ่มต้นเรียนรู้การพัฒนา Arduino แม้แต่ผู้ที่ไม่เคยเรียนรู้ด้านอิเล็กทรอนิกส์มาก่อนก็สามารถนำมาสร้างต้นแบบเกี่ยวกับอิเล็กทรอนิกส์ได้



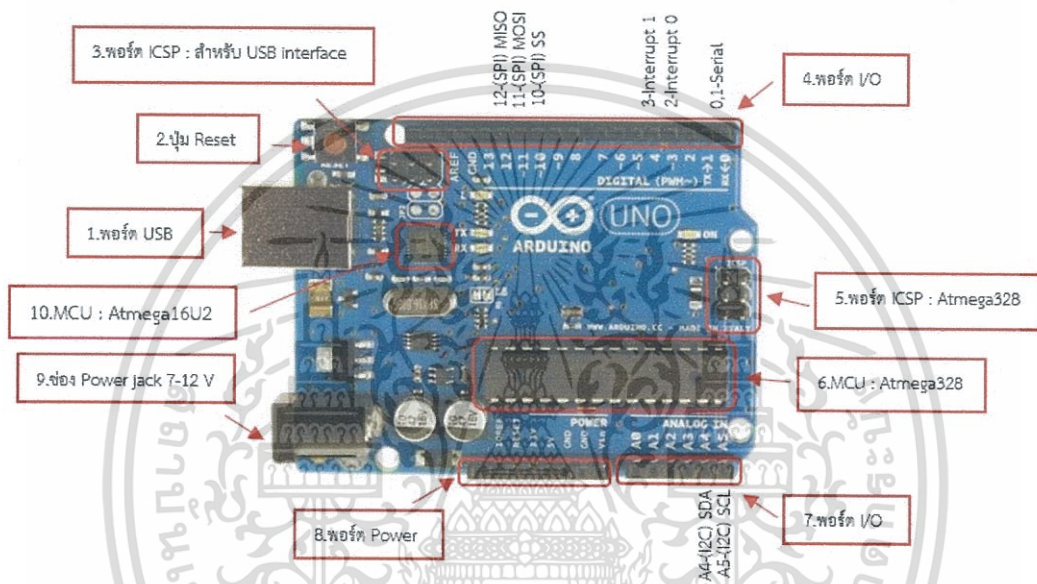
รูปที่ 2.18 Arduino Uno R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน Arduino Uno R3 ยังเพิ่มขา SDA และ SCL ไว้ถัดไปจากขา AREF นอกจากนี้ยังเพิ่มขาใหม่ 2 ขาข้างปุ่ม Reset หนึ่งคือขา IOREF ที่สามารถใช้กับ Shield เพื่อให้สามารถใช้ไฟจากบอร์ดได้ อีกขาหนึ่งไม่มีหน้าที่เป็นขาที่สำรองไว้สำหรับการใช้งานในอนาคต Arduino Uno R3 สามารถใช้งานได้กับ Shield ได้ทุกชนิดแต่สามารถนำไปใช้กับ Shield ชนิดใหม่ได้โดยใช้ขาที่เพิ่มขึ้นมานี้

2.5.1.1 หน้าที่ของขาของบอร์ด Arduino Uno R3

หน้าที่ของขาของบอร์ด Arduino Uno R3 แสดงได้ดังรูปที่ 2.19



รูปที่ 2.19 รูปแสดงหน้าที่ต่างๆของขา Arduino

1) USB Port ใช้สำหรับต่อกับคอมพิวเตอร์เพื่ออัปเดตโปรแกรมเข้า Arduino และจ่ายไฟให้กับบอร์ด

2) Reset Button เป็นปุ่ม reset ใช้กดเมื่อต้องการให้ Arduino เริ่มต้นการทำงานใหม่

3) ICSP Port เป็นพอร์ตของ ATmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com Port บน ATmega16U2

4) I/O Port เป็น Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้บางขาจะทำหน้าที่อื่นๆ เพิ่มเติมด้วยเช่นขา 0,1 เป็นขา Tx, Rx Serial ขา 3,5,6,9,10 และ 11 เป็นขา PWM

5) ICSP Port เป็นพอร์ตของ ATmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

6) MCU ใน Arduino Uno R3 ใช้ ATmega328P-PU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) I/O Port นอกจากจะเป็น Digital I/O แล้วยังสามารถเปลี่ยนมาใช้เป็นช่องรับสัญญาณแอนะล็อกตั้งแต่ขา A0-A5

8) Power Port พอร์ตไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5 V, GND, Vin

9) Power Jack ช่องรับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7 ถึง 12V

10) MCU ATmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย ATmega328 จะติดต่อกับคอมพิวเตอร์ผ่าน ATmega16U2

2.5.2 การสั่งงานผ่านโปรแกรม Arduino IDE

ในการใช้งาน Arduino จำเป็นจะต้องกำหนดคำสั่งลงไปว่าต้องการให้ Arduino ทำงานอะไรบ้างโดย Arduino จะมีโปรแกรมเขียนคำสั่งเพื่อสั่งใช้งาน Arduino ชื่อของโปรแกรมคือ Arduino IDE ซึ่งเป็นซอฟต์แวร์ที่สามารถดาวน์โหลดมาใช้งานได้เลยจากเว็บไซต์ของ Arduino

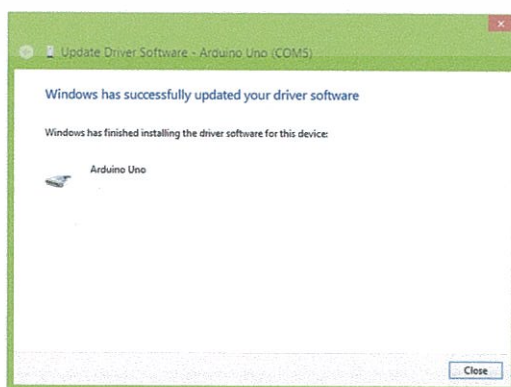
2.5.2.1 ขั้นตอนการเขียนโปรแกรมลงบอร์ด Arduino

1. ต่อบอร์ดเข้ากับคอมพิวเตอร์ผ่าน USB แสดงได้ดังรูปที่ 2.20



รูปที่ 2.20 เชื่อมต่อ Arduino เข้ากับคอมพิวเตอร์

2. ติดตั้ง Driver (กรณีต่อกับคอมพิวเตอร์ครั้งแรก) ได้ดังรูปที่ 2.21



รูปที่ 2.21 รูปแสดงการติดตั้ง Driver

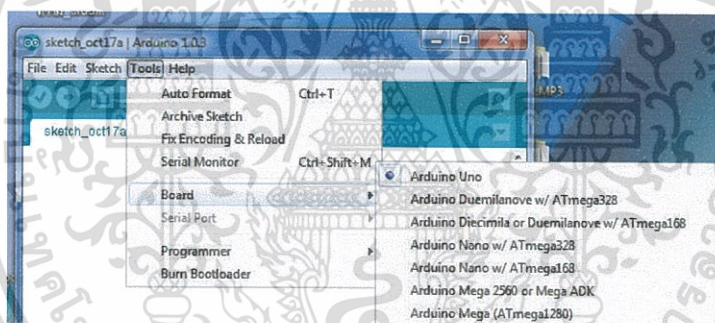
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เปิดโปรแกรม Arduino IDE แสดงได้ดังรูปที่ 2.22



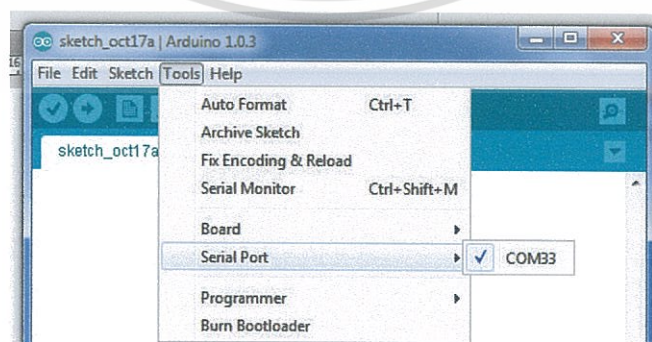
รูปที่ 2.22 โปรแกรม Arduino IDE

4. เลือกบอร์ด Arduino ที่ใช้งาน แสดงได้ดังรูปที่ 2.23



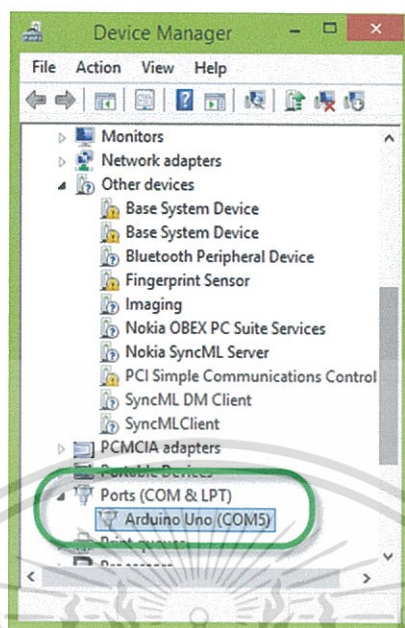
รูปที่ 2.23 การเลือกบอร์ด

5. เลือกพอร์ตที่ต่อบอร์ดเข้าไปเพื่อกำหนดเป้าหมาย แสดงได้ดังรูปที่ 2.24 ในกรณีที่ไม่ทราบให้ไปที่ Device Manager และดูที่ USB แสดงได้ดังรูปที่ 2.25



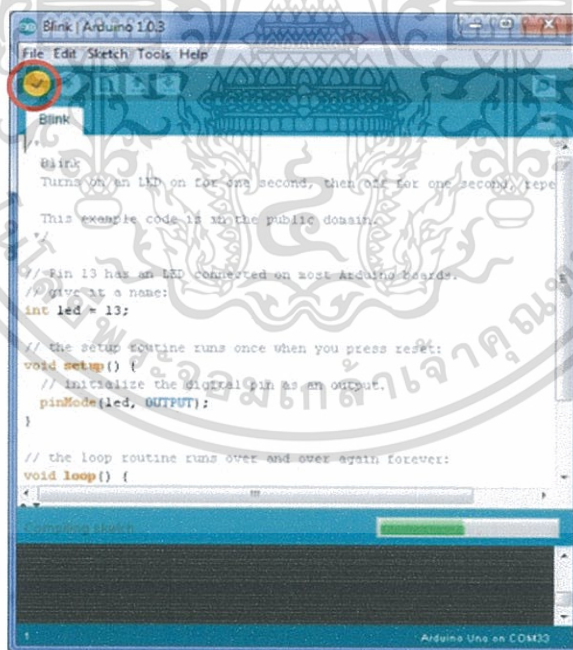
รูปที่ 2.24 กำหนดเป้าหมายเพื่อแปลโปรแกรม คำสั่งไปที่พอร์ตที่เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 Serial Port อยู่ที่ Port COM5

6. เขียนคำสั่งที่ต้องการเพื่อใช้งานบอร์ด
7. กด Verify เพื่อแปลคำสั่งโปรแกรม แสดงได้ดังรูปที่ 2.26



รูปที่ 2.26 แปลคำสั่งโปรแกรม

8. กด Upload เพื่อเขียนคำสั่งลงบอร์ด รอจนขึ้น Done Uploading จึงเสร็จการเขียนคำสั่งลงบอร์ด แสดงได้ดังรูปที่ 2.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Blink
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 * This example code is in the public domain.
 */
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
Binary sketch size: 1,632 bytes (of a 32,000 byte maximum)
Arduino Mega2560 or Mega ADK on COM33

```

รูปที่ 2.27 เขียนคำสั่งลงบอร์ด

2.6 การทำงานของรีเลย์

รีเลย์ (Relay) เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสถานะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งสามารถนำรีเลย์ไปประยุกต์ใช้ในการควบคุมวงจรไฟฟ้าต่าง ๆ ในงานอิเล็กทรอนิกส์มากมาย รูปร่างและสัญลักษณ์ของรีเลย์แสดงดังรูปที่ 2.28



รูปที่ 2.28 รูปร่างและสัญลักษณ์ของรีเลย์

2.6.1 ส่วนประกอบของรีเลย์

รีเลย์ประกอบด้วย 2 ส่วนหลัก ดังนี้

2.6.1.1 ส่วนของขดลวด (Coil)

ทำหน้าที่สร้างสนามแม่เหล็กไฟฟ้าให้แกนโลหะเบนไปสัมผัสกับหน้าสัมผัส ทำงานโดยการรับแรงดันไฟฟ้าจากภายนอกต่อคร่อมที่ขดลวดเหนี่ยวนำนี้ เมื่อขดลวด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้รับแรงดันไฟฟ้า (ค่าแรงดันไฟฟ้าที่รีเลย์ต้องการขึ้นกับชนิดและรุ่นตามที่ถูกผลิตกำหนด) จะเกิดสนามแม่เหล็กไฟฟ้าทำให้แกนโลหะเบนไปสัมผัสกับหน้าสัมผัสที่จุดต่อ NO

2.6.1.2 ส่วนของหน้าสัมผัส (Contact)

ทำหน้าที่เหมือนสวิตช์จ่ายกระแสไฟฟ้าให้กับอุปกรณ์ที่เราต้องการ

2.6.2 จุดต่อใช้งานมาตรฐาน

การทำงานของรีเลย์แสดงได้ดังรูป 2.29 แต่ละจุดมีความหมายดังต่อไปนี้

2.6.2.1 จุดต่อ NC

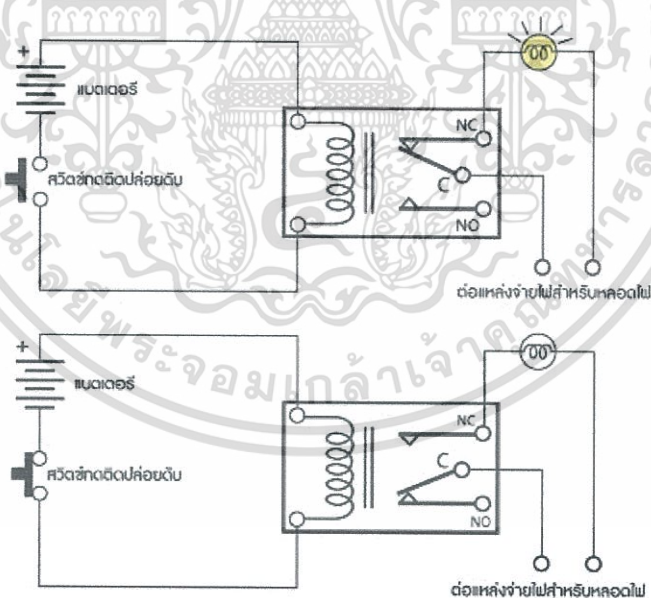
ย่อมาจาก Normally Close หมายความว่า ปกติปิด หรือหากยังไม่จ่ายไฟฟ้าให้ขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการให้ทำงานตลอดเวลา

2.6.2.2 จุดต่อ NO

ย่อมาจาก Normally Open หมายความว่า ปกติเปิด หรือหากยังไม่จ่ายไฟฟ้าให้ ขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน โดยทั่วไปเรามักต่อจุดนี้เข้ากับอุปกรณ์หรือเครื่องใช้ไฟฟ้าที่ต้องการควบคุมการเปิด-ปิด เช่น โคมไฟสนาม ไฟหน้าบ้าน เป็นต้น

2.6.2.3 จุดต่อ C

ย่อมาจาก Common คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟฟ้า



รูปที่ 2.29 การทำงานของรีเลย์

2.6.3 ข้อคำนึงในการใช้งานรีเลย์ทั่วไป

2.6.3.1 แรงดันไฟฟ้าใช้งาน

คือ แรงดันไฟฟ้าที่ทำให้รีเลย์ทำงานได้ หากเราดูที่ตัวรีเลย์จะระบุค่าแรงดันไฟฟ้าใช้งานไว้ (หากใช้ในงานอิเล็กทรอนิกส์ ส่วนมากจะใช้ไฟฟ้ากระแสตรงในการใช้งาน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น 12 VDC คือ ต้องใช้แรงดันไฟฟ้ากระแสตรงที่ 12 โวลต์ เท่านั้น หากใช้มากกว่านี้ขดลวดภายในตัวรีเลย์อาจจะขาดได้ หรือหากใช้แรงดันไฟฟ้าต่ำกว่ามาก รีเลย์จะไม่ทำงาน ส่วนในการต่อวงจรนั้นสามารถต่อขั้วใดก็ได้ เพราะตัวรีเลย์จะไม่ระบุขั้วต่อไว้ (นอกจากชนิดพิเศษ)

2.6.3.2 การใช้งานกระแสไฟฟ้าผ่านหน้าสัมผัส

ตัวรีเลย์จะระบุไว้ เช่น 10A 220AC คือ หน้าสัมผัสของรีเลย์นั้นสามารถทนกระแสไฟฟ้าได้ 10 แอมแปร์ที่ 220VAC แต่การใช้ก็ควรจะใช้งานที่ระดับกระแสไฟฟ้าต่ำกว่านี้ เพราะถ้ากระแสไฟฟ้ามากกว่าหน้าสัมผัสของรีเลย์จะละลายเสียหายได้

2.6.3.3 จำนวนหน้าสัมผัสการใช้งาน

ควรดูว่ารีเลย์นั้นมีหน้าสัมผัสให้ใช้งานกี่อัน และมีขั้ว Common ด้วยหรือไม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปฏิญญาฉบับ

3.1 การออกแบบ

ในการออกแบบระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือและบันทึกภาพใบหน้า คุณสมบัติที่เราต้องการให้มีในระบบมีดังต่อไปนี้

1. มีการลงทะเบียนลายนิ้วมือ และเก็บลงในฐานข้อมูล
2. เมื่อกราดตรวจลายนิ้วมือ ถ้ามีชื่ออยู่ในฐานข้อมูลประตูจะถูกปลดล็อก พร้อมกับมีการบันทึกเวลา และ บันทึกภาพใบหน้าเก็บลงในฐานข้อมูล
3. สามารถปลดล็อกประตูจากคอมพิวเตอร์ได้ ซึ่งจะถูกระบบบันทึกฐานข้อมูล โดยจะถูกบันทึกด้วยคำว่า Admin
4. มีการส่งอีเมลแจ้งเตือนบันทึกเข้าออกอัตโนมัติทุกวันเวลาที่เวลา 20.00 น. หรือกดส่งด้วยตัวเองก่อนหรือหลังเวลาดังกล่าวได้

เนื่องจากมีอุปกรณ์หลักในการควบคุมมีด้วยกันสองตัว คือ Microcontroller และ คอมพิวเตอร์ จึงแบ่งขั้นตอนในการออกแบบออกเป็น 2 ส่วน คือ ส่วนการเขียนโปรแกรมควบคุม Microcontroller และ การเขียนโปรแกรมควบคุมคอมพิวเตอร์

3.1.1 การเขียนโปรแกรมควบคุม Microcontroller

ในส่วนนี้เป็นส่วนของโปรแกรมที่จะดำเนินงานบน Arduino UNO R3 (Microcontroller) ซึ่งสามารถเขียนคำสั่งผ่านโปรแกรม Arduino IDE โดยภาษาที่ใช้เขียนคำสั่งนั้น จะใช้รูปแบบการเขียนของภาษาซี

Arduino นี้เป็นตัวสื่อสารกับ Fingerprint Sensor และ เป็นตัวปลดล็อกประตู ดังนั้นจึงแบ่งการเขียนโปรแกรมได้เป็น 2 ส่วนใหญ่ๆ และสามารถเขียนแผนผังการทำงานได้ดังรูปที่ 3.1 โดย Arduino จะวนซ้ำตรวจสอบว่ามีการรับค่าจาก Serial Port (ช่องทางรับส่งข้อมูลกับคอมพิวเตอร์) หรือไม่ กรณีที่มีค่าส่งมา จะทำการเก็บลงตัวแปรที่ชื่อว่า serip จากนั้นนำไปตรวจสอบว่ามีค่าเป็น a หรือ b ถ้าเป็น a จะเป็นการลงทะเบียนลายนิ้วมือ ถ้าเป็น b จะเป็นการปลดล็อกประตู กรณีที่ไม่มีค่าใดส่งมาจะเข้าสู่การรับค่าลายนิ้วเพื่อจับคู่ลายนิ้วมือ

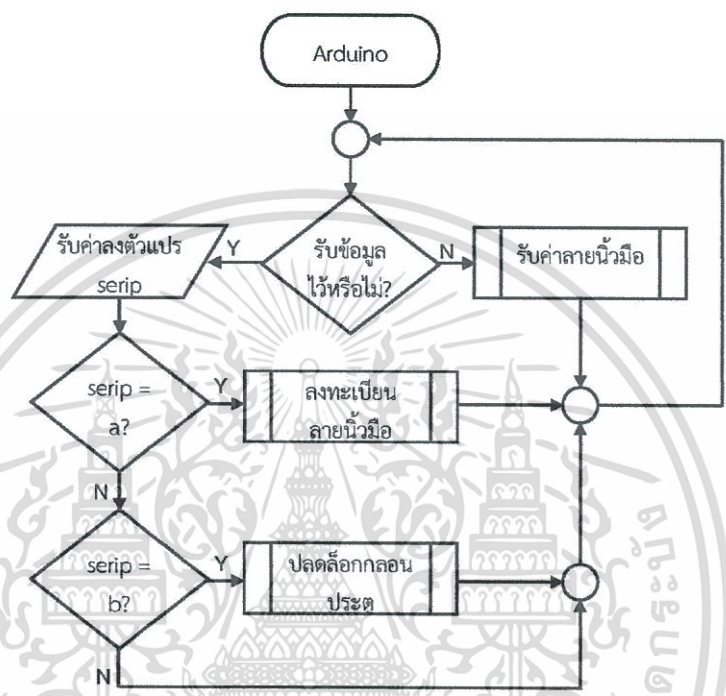
3.1.1.1 Arduino สื่อสารกับ Fingerprint Sensor

คำสั่งควบคุมในส่วนนี้ต้องมีการเรียก Library ของ Fingerprint Sensor ด้วย และยังมีคำสั่งตัวอย่างอีกด้วย ซึ่งคำสั่งตัวอย่างที่เลือกนำมาใช้งานคือ fingerprint และ enroll แสดงได้ดังรูปที่ 3.2

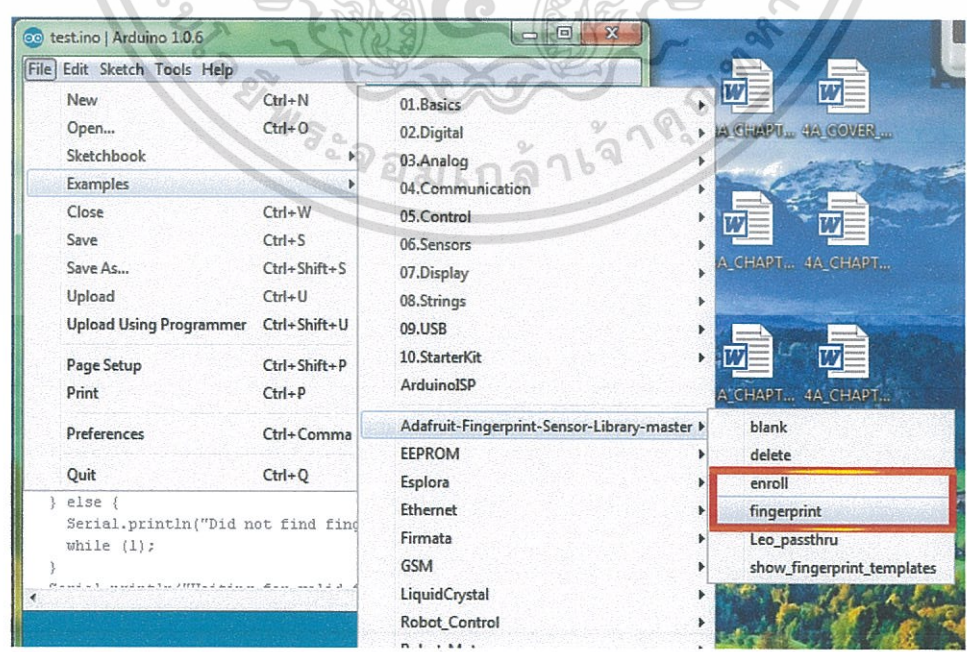
จากคำสั่งตัวอย่าง fingerprint เป็นส่วนของการจับคู่ลายนิ้วมือ จะกระทำเมื่อไม่มีการรับค่าจาก Serial Port ทำให้เข้าสู่การรับค่าลายนิ้วมือเพื่อจับคู่ลายนิ้วมือ แสดงแผนผังการทำงานของการทำงานของการจับคู่ลายนิ้วมือได้ดังรูปที่ 3.3 เริ่มจากรับภาพลายนิ้วมือเข้ามาเพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงภาพเป็น Character File เพื่อนำไปเปรียบเทียบกับลายนิ้วมือที่มีการลงทะเบียนไว้ เมื่อจับคู่สำเร็จจะส่งค่า FingerID ที่เจอออกไป แต่ FingerID เป็นตัวเลขได้ตั้งแต่ 1 ถึง 162 ซึ่งเราต้องการส่งเป็นเลขสามหลักจึงต้องเติม "0" ไปข้างหน้า FingerID ที่มีเลขไม่เป็นสามหลักก่อนส่งออกไปด้วย แต่ถ้าจับคู่ไม่สำเร็จจะส่ง "N" ออกไปทาง Serial Port

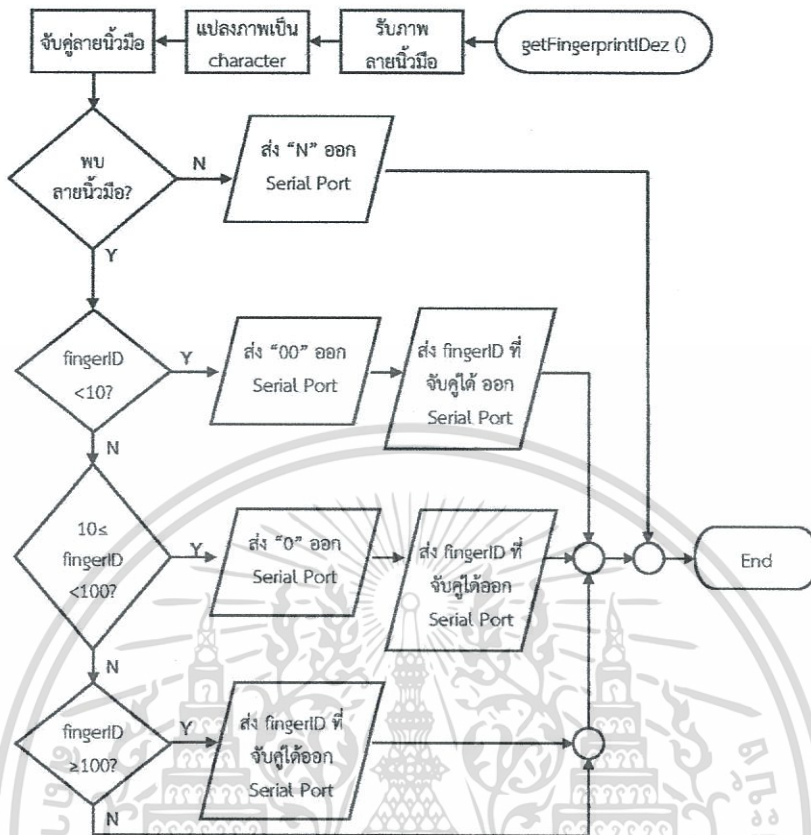


รูปที่ 3.1 แผนผังการทำงานของโปรแกรมที่ดำเนินงานบน Arduino



รูปที่ 3.2 คำสั่งตัวอย่างใน Library ของ Fingerprint Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

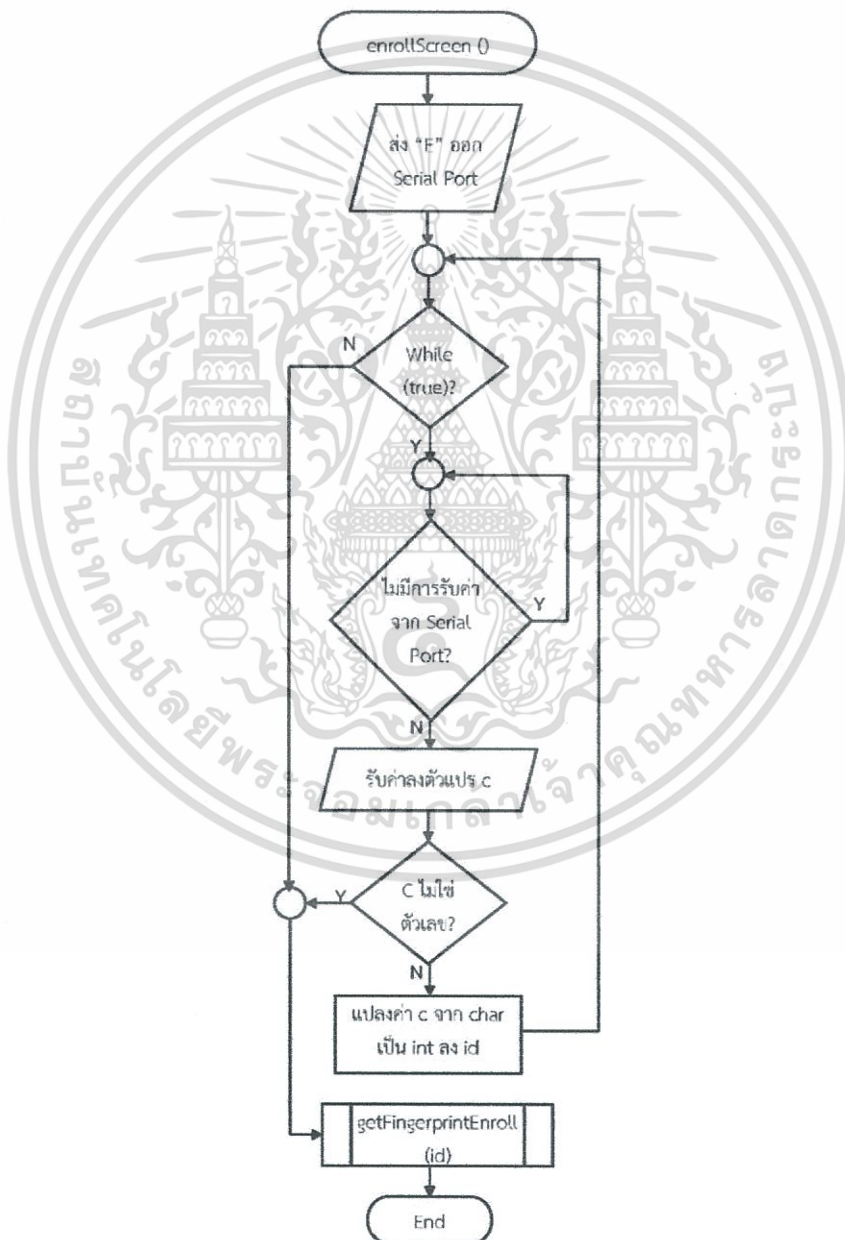


รูปที่ 3.3 แผนผังการทำงานของการทำงานการจับคู่ลายนิ้วมือ

สำหรับการลงทะเบียนลายนิ้วมือจะต้องได้รับค่า a จากคอมพิวเตอร์ก่อนจึงเริ่มการลงทะเบียนลายนิ้วมือ แสดงแผนผังการทำงานของการทำงานลงทะเบียนได้ดังรูปที่ 3.4 เมื่อเข้าสู่การลงทะเบียนจะส่ง "E" ออกไปที่ Serial Port ซึ่งย่อมาจาก Enrolling จากนั้นจะเข้าการวนซ้ำอนันต์เพื่อรอรับค่าจาก Serial Port ลงตัวแปร c ซึ่งเป็นตัวแปรชนิด Character จากนั้น นำค่า c มาตรวจสอบว่าเป็นตัวเลขหรือไม่ ถ้าเป็นตัวเลขจะแปลงเป็น integer ก่อน แล้วเก็บตัวเลขทั้งหมดลงตัวแปรใหม่ที่ชื่อว่า id และวนรับค่าไปเรื่อยๆ จนกว่า c จะไม่ใช่ตัวเลข จึงจะออกจากการวนซ้ำอนันต์ แล้วจะทำกระบวนการย่อย `getFingerprintEnroll()` ต่อไป

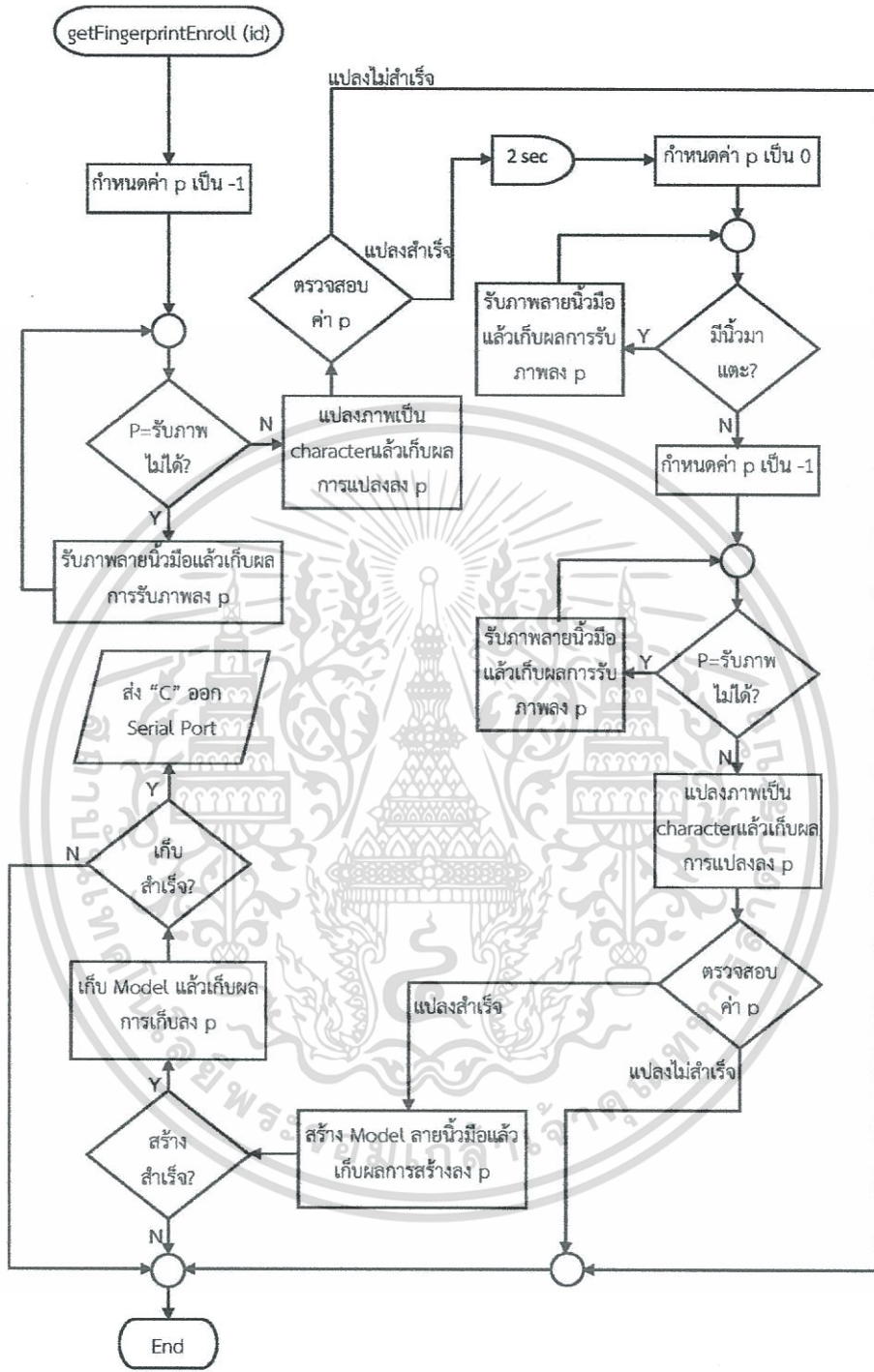
เมื่อเข้ากระบวนการย่อย `getFingerprintEnroll()` จะกำหนดค่าเริ่มต้นให้ p ซึ่งค่า p นี้จะใช้ในการตรวจสอบสถานะของกระบวนการแต่ละขั้นในการลงทะเบียน เช่น การรับภาพลายนิ้วมือ ถ้ารับภาพได้ ให้ $p=0$ ในทางกลับกันถ้ารับภาพไม่ได้ จะให้ $p \neq 0$ เป็นต้น หลังจากกำหนดค่าเริ่มต้นให้ p ต่อมาคือการรับภาพลายนิ้วมือ จะมีการนำค่า p มาตรวจสอบ ถ้ารับภาพไม่ได้จะวนซ้ำรับภาพลายนิ้วมือ แต่ถ้ารับได้จะนำภาพที่ได้เข้ากระบวนการแปลงภาพเป็น Character ซึ่งค่า p หรือผลจากการแปลงจะถูกตรวจสอบ ถ้าแปลงค่าไม่สำเร็จจะจบการทำงาน ถ้าแปลงค่าสำเร็จจะรอ 2 วินาที เพื่อรอให้ยกนิ้วมือออกจากแท่นวางนิ้ว จากนั้นจะวนซ้ำเพื่อตรวจสอบว่ายกนิ้วออกไปหรือยัง นั่นคือถ้ายังมีการรับภาพลายนิ้วมือได้แสดงว่ายังไม่ยกนิ้วออกและวนซ้ำไปเรื่อยๆ เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อยๆ ถ้ารับภาพไม่ได้จะวนซ้ำรอรับไปเรื่อยๆ พอรับภาพลายนิ้วมือครั้งที่ 2 ได้แล้วจะนำมาแปลงเป็น Character File แล้วตรวจสอบค่า p ว่าแปลงสำเร็จไหม ถ้าไม่สำเร็จจะจบการทำงาน แต่ถ้าแปลงสำเร็จจะนำ Character File ของลายนิ้วมือทั้งสองครั้งมาสร้าง Template File ซึ่งจะเก็บเอกลักษณ์ต่างๆของลายนิ้วมือไว้ และตรวจสอบด้วยว่าสร้างสำเร็จไหม ถ้าไม่สำเร็จจะจบการทำงาน แต่ถ้าสร้างสำเร็จจะทำการจัดเก็บลงหน่วยความจำถาวร และยังคงตรวจสอบค่า p อยู่เหมือนเดิมว่าเก็บสำเร็จหรือไม่ ถ้าไม่สำเร็จจะจบการทำงาน ถ้าสำเร็จจะส่ง "C" ออกไปที่ Serial Port ย่อมาจาก Completed แล้วจึงจบการทำงาน



รูปที่ 3.4 แผนผังการทำงานของ การลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แผนผังการทำงานของเครื่องลงทะเบียน (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

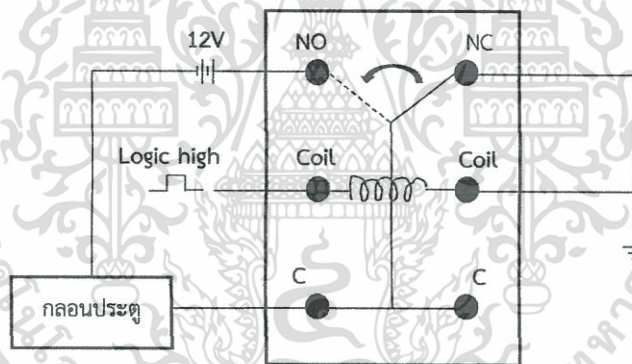
ค่าที่ Arduino ส่งให้คอมพิวเตอร์สามารถสรุปได้ดังตารางที่ 3.1

ตารางที่ 3.1 ค่าที่ Arduino ส่งให้คอมพิวเตอร์

ค่าที่ส่ง	ความหมาย
E	เพื่อให้โปรแกรมแจ้งผู้ใช้งานว่า Enrolling... คือ กำลังลงทะเบียนลายนิ้วมือ
C	เพื่อให้โปรแกรมแจ้งผู้ใช้งานว่า Completed คือ ลงทะเบียนลายนิ้วมือเสร็จแล้ว
N	เพื่อให้โปรแกรมแจ้งผู้ใช้งานว่า No fingerprint matched, please try again คือ ไม่พบลายนิ้วมือ

3.1.1.2 Arduino ปลดล็อกประตู

Arduino จะปลดล็อกประตูเมื่อได้รับค่า “b” จาก Serial Port การปลดล็อกประตูอาศัยรีเลย์ในการควบคุมการจ่ายไฟเข้ากลอนประตู โดยถ้าเราต้องการให้ประตูเปิด Arduino ส่ง Logic High (3.3 โวลต์) ออกไป รีเลย์จากที่อยู่จุดต่อ NC จะเปลี่ยนมาอยู่ที่ NO ทำให้ไฟฟ้า 12 โวลต์จากแหล่งจ่ายนั้นจ่ายเข้ากลอนทำให้กลอนปลดล็อกได้ ดังรูปที่ 3.5 จากนั้นจะรอ 3 วินาที ก่อนที่จะให้ล็อกกลอนดั้งเดิมโดยส่ง Logic Low (0 โวลต์) ออกไป



รูปที่ 3.5 การปลดล็อกกลอนประตู

3.1.2 การเขียนโปรแกรมควบคุมคอมพิวเตอร์

เป็นส่วนของการเขียนโปรแกรมควบคุมคอมพิวเตอร์ โดยใช้โปรแกรม Visual Studio 2013 เพื่อออกแบบ Graphic Interface ที่ใช้ในการติดต่อกับผู้ใช้งานผ่านคอมพิวเตอร์ และติดต่อกับฐานข้อมูล ในการสร้างฐานข้อมูลใช้โปรแกรม Access 2013 ดังนั้นจะขอแบ่งการออกแบบเป็น 2 ส่วนสำคัญ คือ การสร้างฐานข้อมูลแอคเซส และ การสร้าง Graphic Interface

3.1.2.1 การสร้างฐานข้อมูลแอคเซส

ข้อมูลที่ต้องการให้มีในตารางฐานข้อมูลเกี่ยวกับนักศึกษาได้แก่

1. หมายเลข id ของลายนิ้วมือ 3 หลัก กำหนดให้เป็น Primary Key กำหนด Field Properties ดังนี้

Data Type = Short Text

Field Size = 255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั่นเอง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Required	= Yes
Allow Zero length	= No
Indexed	= Yes (No Duplicates)
Unicode Compression	= No
IME Mode	= No Control
IME Sentence Mode	= None
Text Align	= General

2. รหัสนักศึกษา 8 หลัก กำหนด Field Properties ดังนี้

Data Type	= Short Text
Field Size	= 255
Required	= Yes
Allow Zero length	= No
Indexed	= Yes (No Duplicates)
Unicode Compression	= Yes
IME Mode	= No Control
IME Sentence Mode	= None
Text Align	= General

3. ชื่อจริงของนักศึกษา กำหนด Field Properties ดังนี้

Data Type	= Short Text
Field Size	= 255
Required	= Yes
Allow Zero length	= No
Indexed	= Yes (No Duplicates)
Unicode Compression	= Yes
IME Mode	= No Control
IME Sentence Mode	= None
Text Align	= General

ข้อมูลที่ต้องการให้มีในตารางฐานข้อมูลเกี่ยวกับการบันทึกการ

เกรดตรวจलयนิ้วมือได้แก่

1. เลขบอกลำดับการเกรดตรวจलयนิ้วมือ กำหนด Field Properties ดังนี้

Data Type	= AutoNumber
Field Size	= Long Integer
New Values	= Increment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Indexed = Yes (No Duplicates)

Text Align = Left

2. หมายเลข id ของลายนิ้วมือ กำหนด Field Properties ดังนี้

Data Type = Short Text

Field Size = 255

Required = No

Allow Zero length = No

Indexed = Yes (No Duplicates)

Unicode Compression = No

IME Mode = No Control

IME Sentence Mode = None

Text Align = General

3. รหัสนักศึกษา 8 หลัก กำหนด Field Properties ดังนี้

Data Type = Short Text

Field Size = 255

Required = Yes

Allow Zero length = No

Indexed = Yes (No Duplicates)

Unicode Compression = Yes

IME Mode = No Control

IME Sentence Mode = None

Text Align = General

4. ชื่อจริงของนักศึกษา กำหนด Field Properties ดังนี้

Data Type = Short Text

Field Size = 255

Required = Yes

Allow Zero length = Yes

Indexed = Yes (No Duplicates)

Unicode Compression = Yes

IME Mode = No Control

IME Sentence Mode = None

Text Align = General

5. วัน เวลา ที่ทำการกราดตรวจลายนิ้วมือ กำหนด Field Properties ดังนี้

Data Type	= Date/Time
Required	= Yes
Indexed	= Yes (No Duplicates)
IME Mode	= No Control
IME Sentence Mode	= None
Text Align	= General
Show Date Picker	= For dates

6. Path รูปภาพที่บันทึกได้ กำหนด Field Properties ดังนี้

Data Type	= Hyperlink
Required	= No
Allow Zero length	= Yes
Indexed	= No
Unicode Compression	= Yes
IME Mode	= No Control
IME Sentence Mode	= None
Text Align	= General
Append Only	= No

นำข้อมูลที่ต้องการใหม่ มาสร้างเป็นตารางได้ 2 ตารางดังรูปที่ 3.6



รูปที่ 3.6 ตารางฐานข้อมูล

3.1.2.2 การสร้าง Graphic Interface

การสื่อสารระหว่าง Arduino กับ คอมพิวเตอร์ ได้ทำการออกแบบโปรแกรม เพื่อให้ง่ายต่อการใช้งานจึงแสดงในรูปแบบ Graphic Interface แสดงแผนผังการทำงานได้ดังรูปที่ 3.7 เริ่มต้นกำหนดค่าตัวแปรต่างๆ จากนั้นตั้งค่าเริ่มต้นของปุ่มต่างๆ คือ ปุ่ม Connect จะใช้ได้ต่อเมื่อเจอ Serial Port และปุ่ม Unlock จะใช้ได้ก็ต่อเมื่อเปิดกล่องแล้ว ปุ่ม Enroll, Start, Stop จะใช้ได้จนกว่าจะมีการเปิด Serial Port แล้ว เมื่อตั้งค่าเริ่มต้นปุ่มต่างๆ แล้วจะกราดตรวจหาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

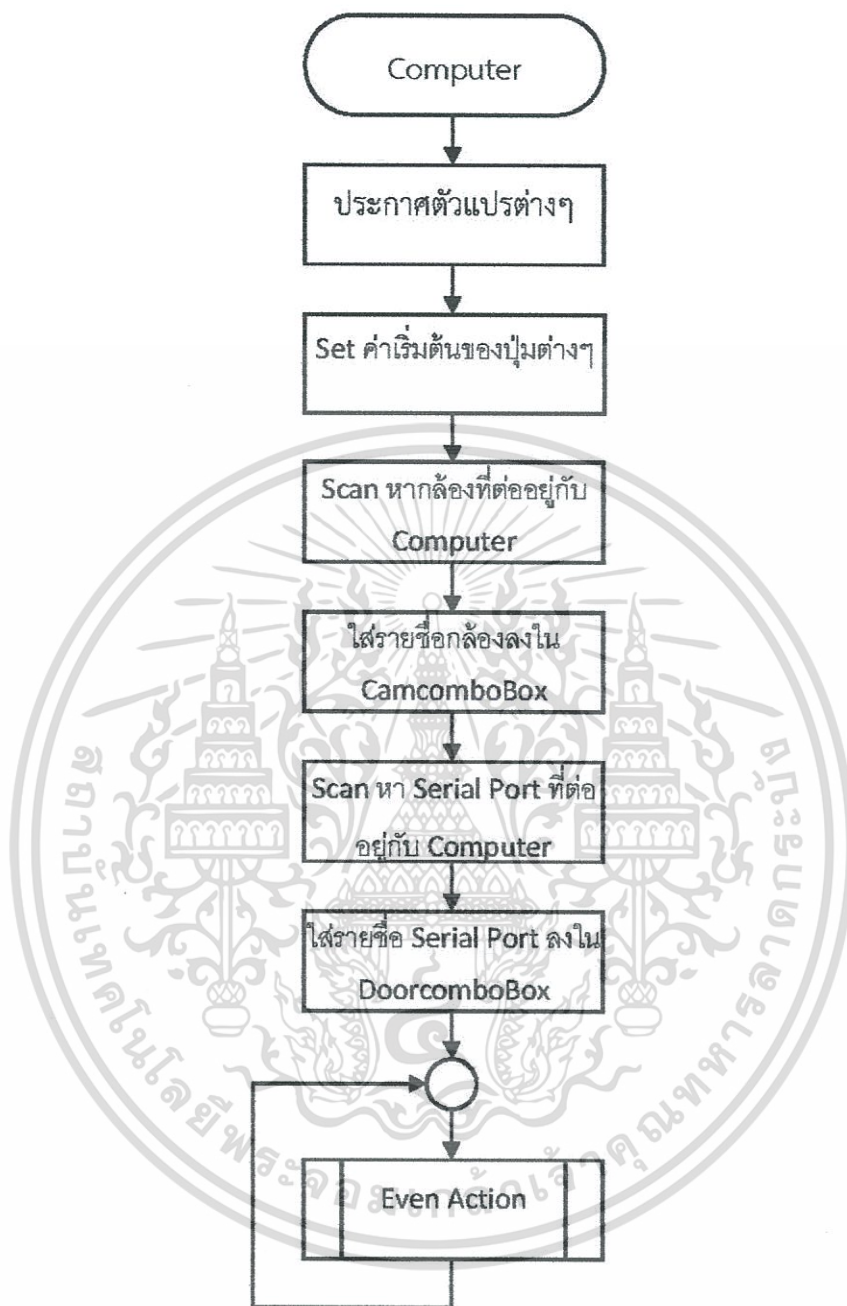
กล่องที่เชื่อมต่ออยู่กับคอมพิวเตอร์ แล้วนำมาใส่ลงใน Combo Box (กล่องแสดงรายการให้ผู้ใช้งาน ได้กดเลือก) ที่ชื่อว่า CamcomboBox จากนั้นกราดตรวจหา Serial Port ที่ต่ออยู่กับคอมพิวเตอร์ และใส่ใน Combo Box ที่ชื่อว่า DoorcomboBox จากนั้นจะรอรับ Action ต่างๆ จากผู้ใช้งาน คือ การได้รับค่าจาก Serial Port, การรับภาพจากกล้อง, การเปลี่ยนแปลงของ Index, การดำเนินงานของนาฬิกา และการกดปุ่มต่างๆ ที่แสดงอยู่บนโปรแกรม

โดย Event Action คือ สิ่งที่โปรแกรมจะดำเนินการตามคำสั่งที่ เขียนไว้ในแต่ละส่วน ซึ่งสามารถสรุป Event Action ทั้งหมดไว้ดังตารางที่ 3.2

ตารางที่ 3.2 Event Action ทั้งหมด

Action	Process
การรับค่าจาก Serial Port	รับค่าจาก Arduino เพื่อมาประมวลผล ประกอบด้วยการ Query ข้อมูลนักศึกษา การบันทึกการกราดตรวจลายนิ้วมือและการแจ้ง การทำงานของระบบให้แก่ผู้ใช้
การกดปุ่ม Connect	เปิดการเชื่อมต่อกับ Arduino
การกดปุ่ม Start	เริ่มรับภาพจากกล้อง Webcam
การรับภาพจากกล้อง	แสดงภาพที่รับจากกล้อง Webcam
การกดปุ่ม Stop	หยุดรับภาพจากกล้อง Webcam
การกดปุ่ม Delete	ลบข้อมูลนักศึกษาตามรหัสนักศึกษาที่กรอกออกจากฐานข้อมูล
การเปลี่ยนแปลง Index ของ DoorcomboBox	เลือก Select Port ที่จะเชื่อมต่อ
การกดปุ่ม Enroll	เพิ่มข้อมูลนักศึกษาลงฐานข้อมูล และส่งข้อมูลที่ใช้สำหรับ ลงทะเบียนให้ Arduino เพื่อลงทะเบียนลายนิ้วมือ
การกดปุ่ม Login	เก็บอีเมลผู้รับ ผู้ส่ง และรหัสผ่านผู้ส่งลงตัวแปร
การกดปุ่ม Send Email	สำหรับส่งอีเมลด้วยตัวเอง
การดำเนินงานของนาฬิกา	แสดงวันที่-เวลาปัจจุบัน และส่งอีเมลอัตโนมัติเมื่อถึงเวลาที่กำหนด
การกดปุ่ม Unlock	ส่งค่าให้ Arduino เพื่อให้ Arduino ปลดล็อกประตู
การเปลี่ยนแปลง Index ของ LabTimecomboBox	เลือกเวลาปฏิบัติการเพื่อเป็นการกำหนดเวลาเข้าสายและเวลาออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

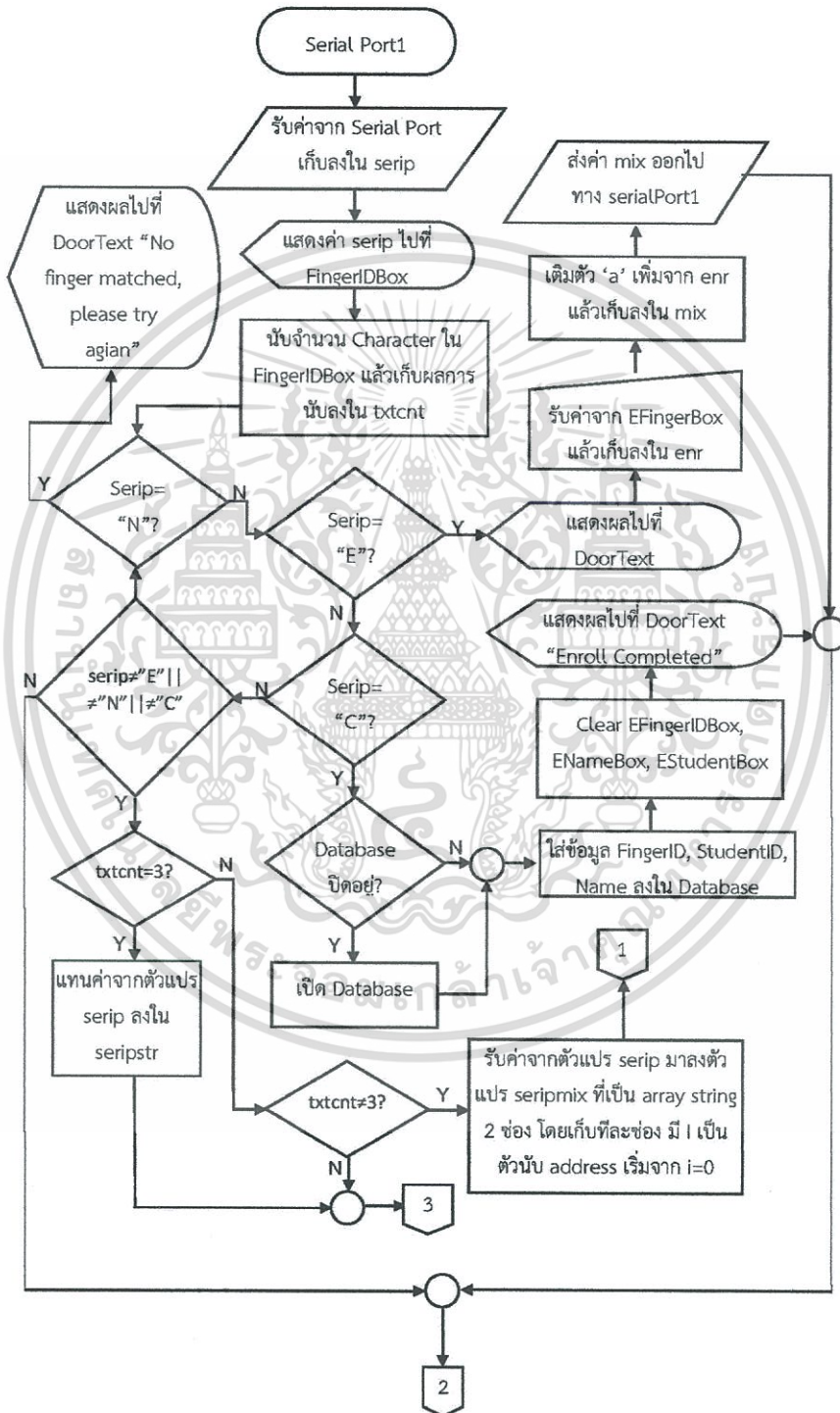


รูปที่ 3.7 แผนผังการทำงานของโปรแกรมบนคอมพิวเตอร์

1) Action การได้รับค่าจาก Serial Port

แสดงแผนผังการทำงานได้ดังรูปที่ 3.8 เมื่อได้รับค่าจาก Serial Port จะนำมาเก็บลงตัวแปร serip และแสดงค่าออกทาง Text Box (กล่องแสดงข้อความ) ที่ชื่อว่า FingerIDBox มีการนับจำนวน Character ลงตัวแปร txtcnt ไว้ด้วยและนำ serip มาตรวจสอบ serip มีค่าเป็น "E" แสดงข้อความว่า "Enrolling..." ออกทาง Text Box ที่ชื่อว่า DoorText จากนั้นรับค่าจาก Text Box ที่ชื่อว่า EFingerBox (กล่องที่ผู้ใช้ใช้กรอกเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลข FingerID) แล้วเก็บค่าลงตัวแปร enr (FingerID) ก่อนส่งค่าออกไปต้องเติม "a" เข้าไปอีกก่อน เพื่อเป็นจุดสิ้นสุดของการรับค่าตัวเลขของ Arduino



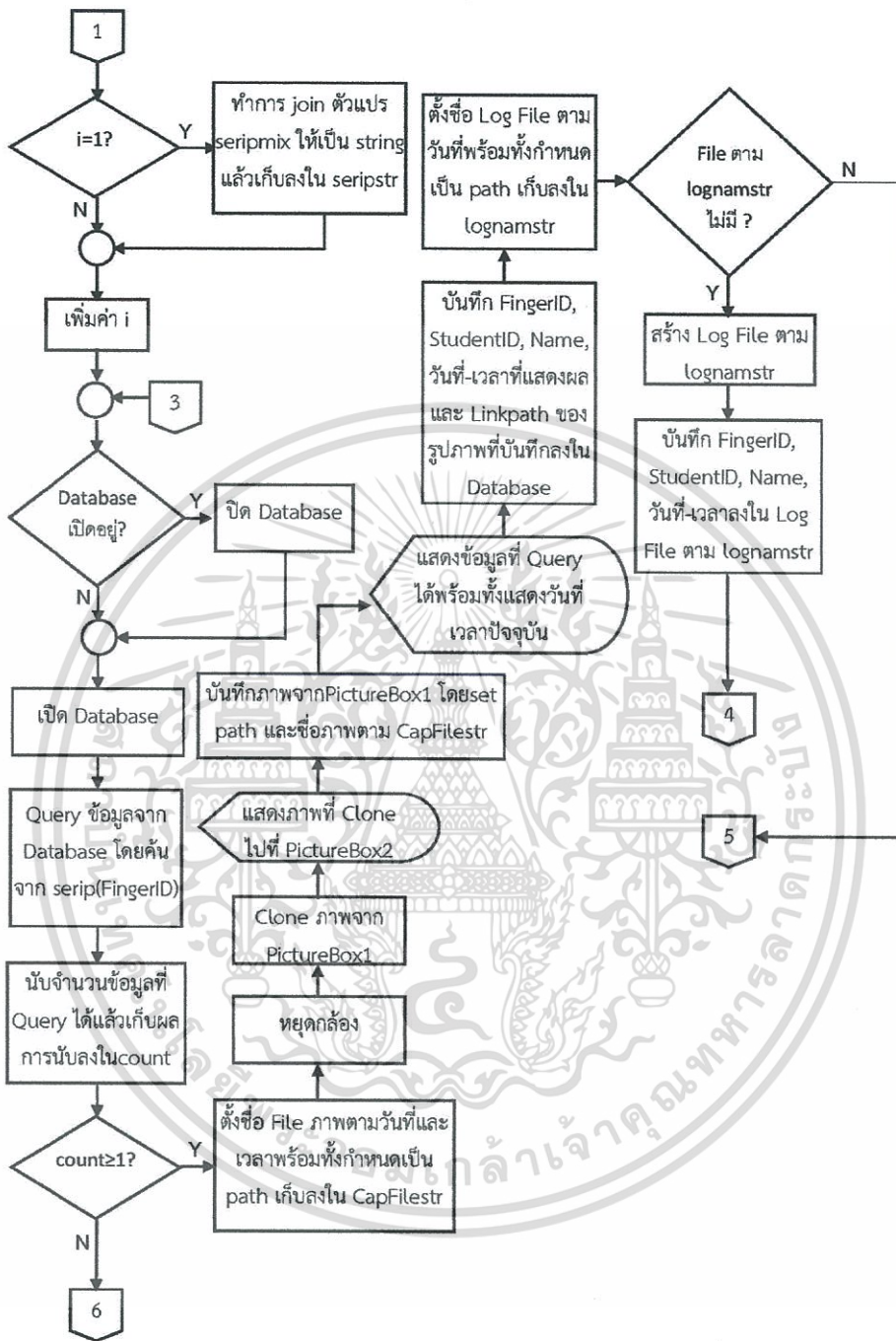
รูปที่ 3.8 แผนผัง Action การรับค่าจาก Serial Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

serip มีค่าเป็น “C” หมายความว่า การลงทะเบียนลายนิ้วมือสำเร็จ ให้มาตรวจสอบว่าเปิดการเชื่อมต่อฐานข้อมูลแอสเซสอยู่หรือไม่ ถ้าปิดก็ให้ทำการเปิดการเชื่อมต่อ แต่ถ้าเปิดแล้ว ให้นำข้อมูล FingerID, รหัสนักศึกษา และชื่อ (ข้อมูลนี้ผู้ใช้ต้องกรอกให้เรียบร้อยก่อนดำเนินการลงทะเบียน) เก็บลงในฐานข้อมูล เมื่อเก็บลงฐานข้อมูลเรียบร้อยแล้ว ให้นำข้อความใน Text Box ที่มี FingerID, รหัสนักศึกษา และชื่อ ให้อ้างเพื่อรอรับค่าในอนาคต และแสดงผลออกไปที่ DoorText ว่า “Enroll Completed”

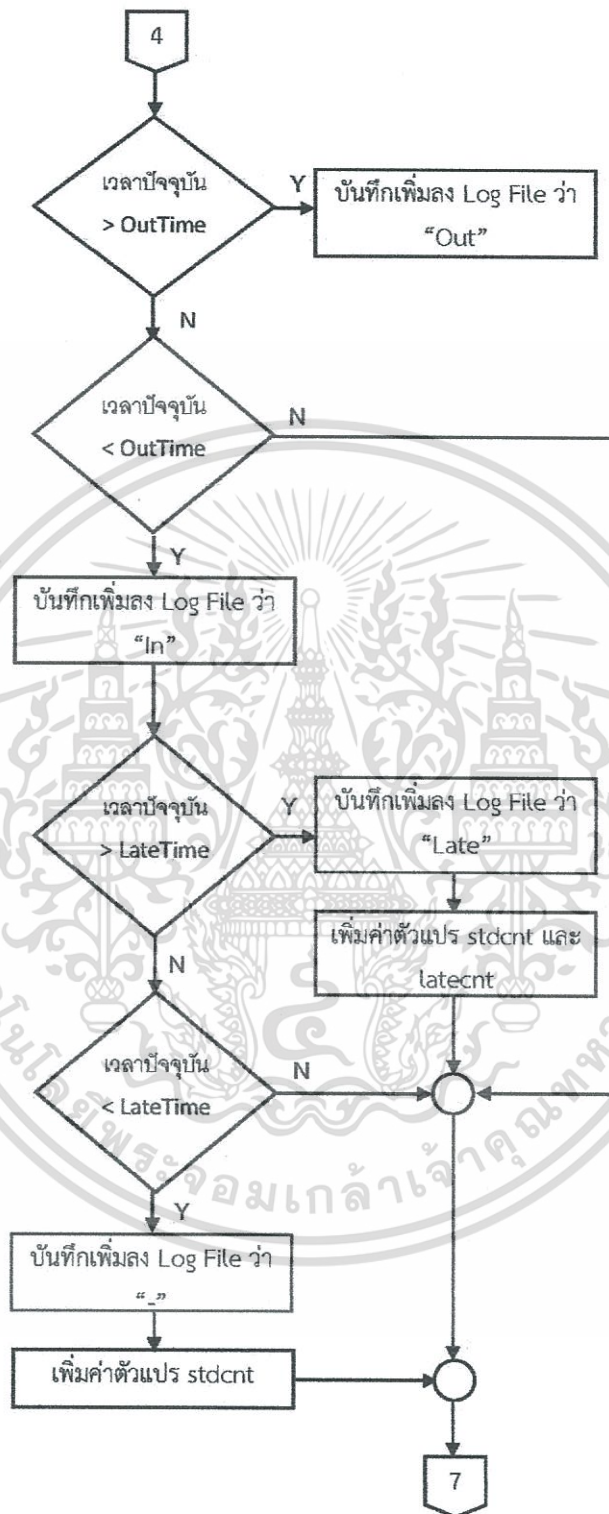
serip มีค่าไม่เป็น “E, N, C” ให้ตรวจสอบว่าจำนวน Character ในตัวแปร txtcnt เท่ากับ 3 หรือไม่ ถ้าเท่าให้แทนค่า serip ลงในตัวแปร seripstr จากนั้นให้ตรวจสอบว่าเปิดการเชื่อมต่อฐานข้อมูลแอสเซสอยู่หรือไม่ ถ้าเปิดให้ปิดก่อนแล้วเปิดใหม่ ถ้าปิดอยู่ก็ให้เปิด จากนั้นทำการ Query ข้อมูลจากฐานข้อมูลโดยใช้ค่าจากตัวแปร serip เป็นค่าสำคัญในการ Query ข้อมูล เมื่อ Query ข้อมูลแล้วให้นำจำนวนข้อมูลที่ Query ได้ แล้วให้เก็บผลการนับลงในตัวแปร count แล้วให้ตรวจสอบว่า count มีค่ามากกว่าหรือเท่ากับ 1 หรือไม่ ถ้าไม่ใช่ให้จบการทำงาน แต่ถ้าใช่ให้ตั้งชื่อแฟ้มข้อมูลรูปภาพตามวันที่และเวลาพร้อมทั้งกำหนดเป็น path เก็บลงใน CapFilestr จากนั้นทำการหยุดกล้องแล้ว Clone ภาพจาก PictureBox1 ไปแสดงที่ PictureBox2 แล้วบันทึกภาพจาก PictureBox1 โดยกำหนด path และชื่อตาม CapFilestr จากนั้นจะทำการแสดงข้อมูลที่ Query ได้พร้อมทั้งแสดงวันที่ เวลาปัจจุบัน แล้วบันทึก FingerID, StudentID, Name, วันที่ เวลาที่แสดงผล และ path ของรูปภาพที่บันทึกลงในฐานข้อมูล จากนั้นตั้งชื่อ แฟ้มลงบันทึกเข้าออกตามวันที่พร้อมทั้งกำหนดที่อยู่ เก็บลงใน lognamstr แล้วตรวจสอบว่าแฟ้มข้อมูลตาม lognamstr มีหรือไม่ หากมีอยู่แล้วให้ทำการบันทึกข้อมูลลงไปเลย แต่ถ้าหากไม่มีให้ทำการสร้างแฟ้มข้อมูลขึ้นมา ก่อนแล้วจึงเริ่มบันทึกข้อมูล โดยหลังจากบันทึกข้อมูลแล้วให้ทำการเปรียบเทียบเวลาปัจจุบันเพื่อตัดสินใจว่าเป็นการเข้าหรือออก หากเป็นการเข้าจะตรวจสอบอีกครั้งว่ามาสายหรือไม่ แล้วทำการเขียนเพิ่มลงไป จากนั้นจึงจบการทำงาน

serip มีค่าเป็น “N” หมายความว่า ไม่พบลายนิ้วมือ ระบบจะแสดงข้อมูลให้ผู้ใช้ทราบว่า “No fingerprint matched, please try again.”



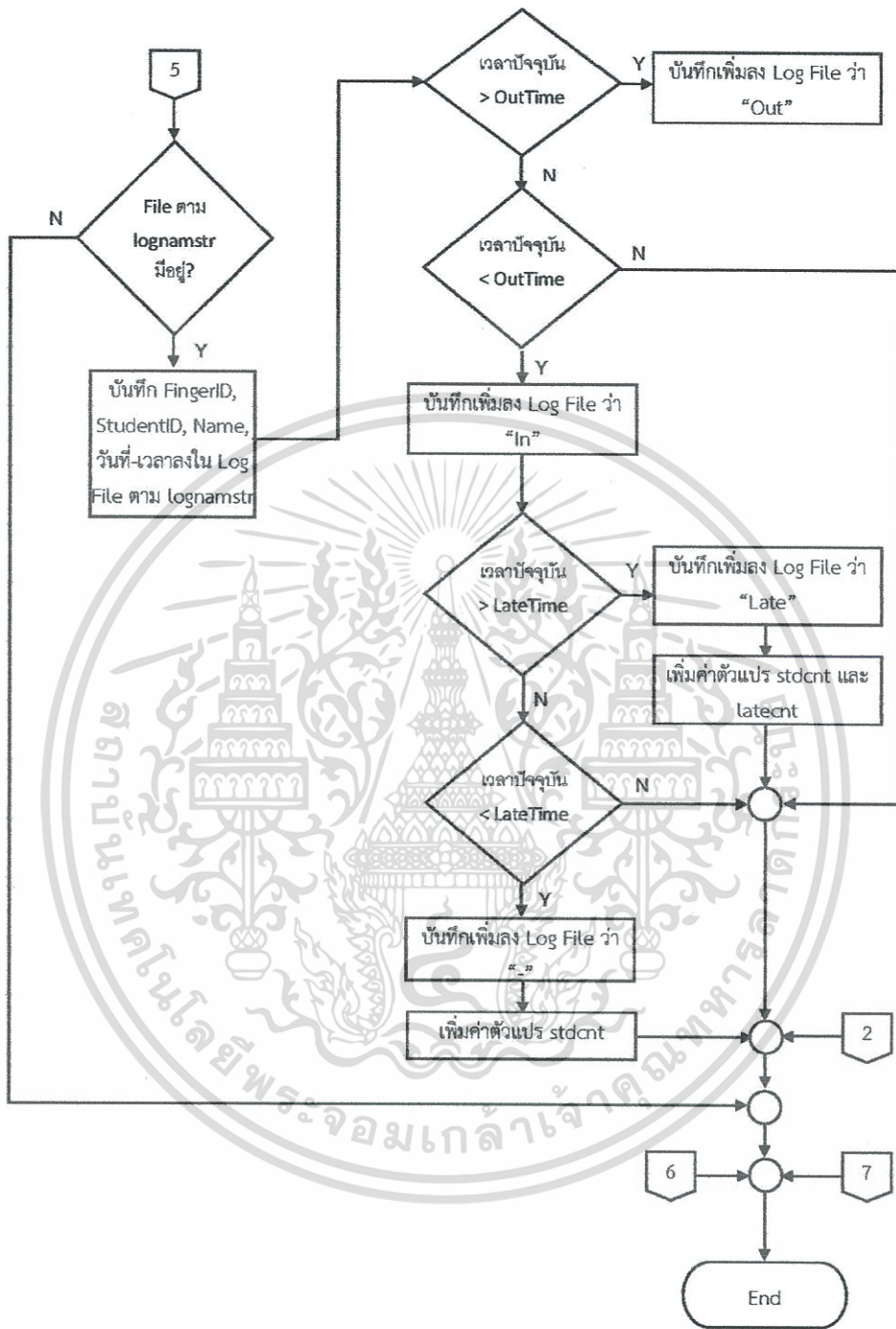
รูปที่ 3.8 แผนผัง Action การรับค่าจาก Serial Port (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แผนผัง Action การรับค่าจาก Serial Port (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

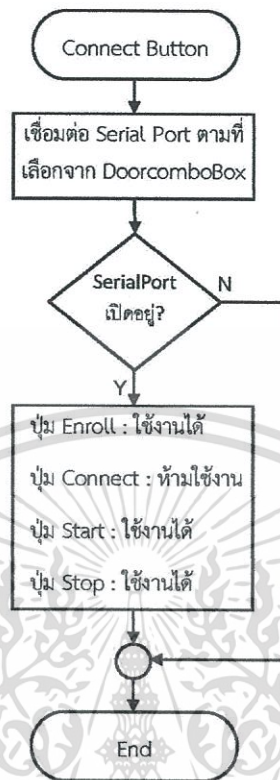


รูปที่ 3.8 แผนผัง Action การรับค่าจาก Serial Port (ต่อ)

2) Action การกดปุ่ม Connect

แสดงแผนผังการทำงานได้ดังรูปที่ 3.9 เมื่อกดปุ่ม Connect จะเปิดการเชื่อมต่อกับ Serial Port (Arduino) ที่ผู้ใช้เลือกจากกล่อง DoorcomboBox ถ้าเปิดการเชื่อมต่อได้จะไปตั้งค่าการทำงานของปุ่มเหล่านี้ ปุ่ม Enroll, Start, Stop จะใช้งานได้ ปุ่ม Connect จะห้ามใช้งาน ถ้าเปิดการเชื่อมต่อไม่สำเร็จจะจบการทำงาน

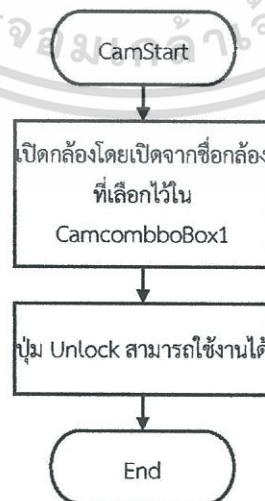
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ ซึ่งไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แผนผัง Action การกดปุ่ม Connect

3) Action การกดปุ่ม Start

แสดงแผนผังการทำงานได้ดังรูปที่ 3.10 เมื่อกดปุ่ม Start จะเปิดการเชื่อมต่อกับกล้องที่ผู้ใช้เลือกจาก CamcomboBox เมื่อเชื่อมต่อเรียบร้อยปุ่ม Unlock จะสามารถใช้งานได้

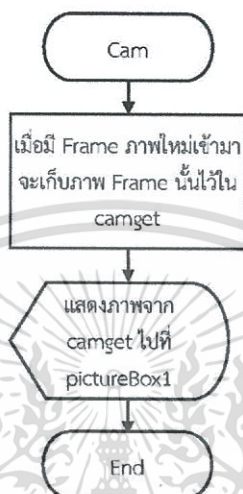


รูปที่ 3.10 แผนผัง Action การกดปุ่ม Start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) Action การรับภาพจากกล้อง

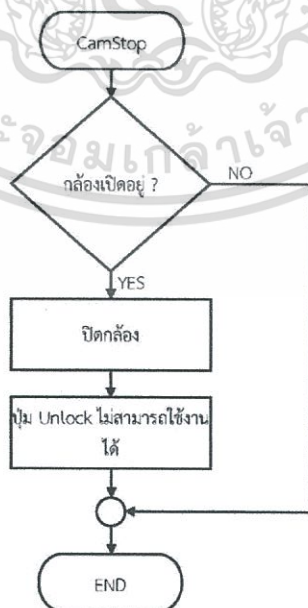
แสดงแผนผังการทำงานได้ดังรูปที่ 3.11 หลังจากทำการเชื่อมต่อกล้องเรียบร้อยแล้ว จะเข้าสู่ Action การรับภาพจากกล้อง เมื่อมี Frame ภาพใหม่เข้ามาจะเก็บลงตัวแปร camget และแสดงภาพนั้นไปที่ pictureBox1



รูปที่ 3.11 แผนผัง Action การรับภาพจากกล้อง

5) Action การกดปุ่ม Stop

แสดงแผนผังการทำงานได้ดังรูปที่ 3.12 เมื่อกดปุ่ม Stop จะมีการตรวจสอบว่ากล้องเปิดอยู่หรือไม่ ถ้าปิดการเชื่อมต่ออยู่ก็จบการทำงาน ถ้าเปิดการเชื่อมต่ออยู่ ก็ จะปิด และทำให้ปุ่ม Unlock ไม่สามารถใช้งานได้

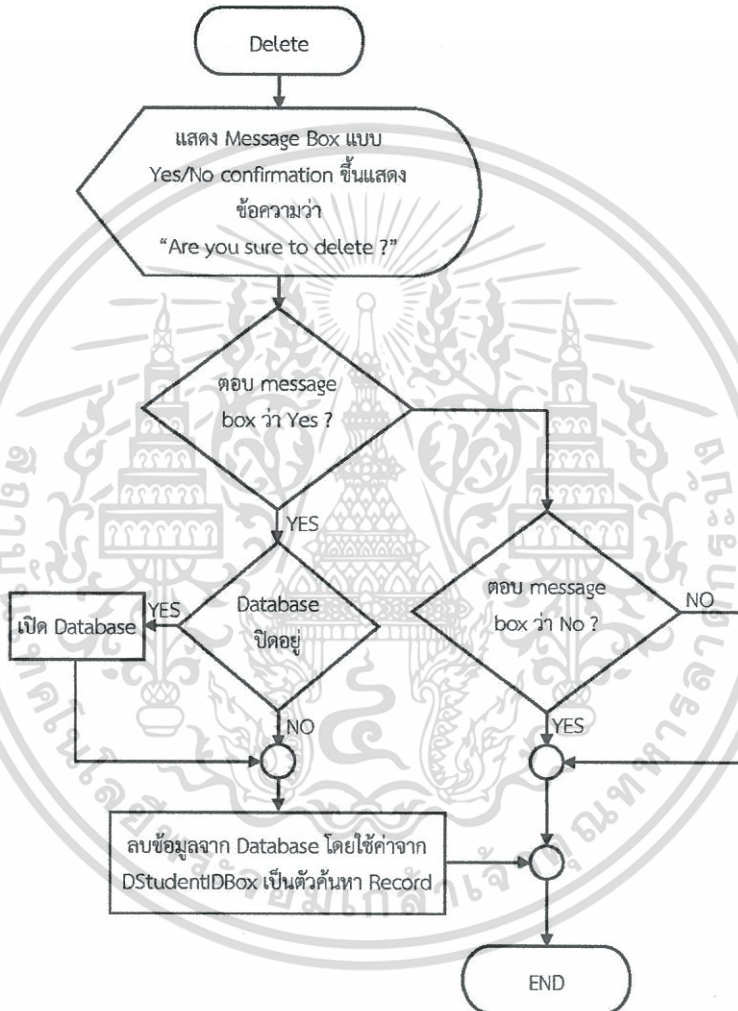


รูปที่ 3.12 แผนผัง Action การกดปุ่ม Stop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) Action การกดปุ่ม Delete

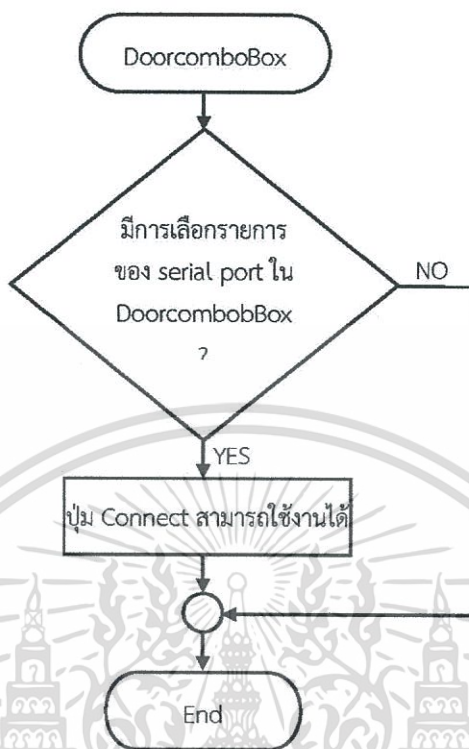
แสดงแผนผังการทำงานได้ดังรูปที่ 3.13 เมื่อกดปุ่ม Delete จะแสดง Message Box แบบ Yes/No confirmation ดังขึ้นมาถามว่า “Are you sure to delete?” ถ้าผู้ใช้ตอบว่า Yes จะไปตรวจสอบว่าปิดการเชื่อมต่อฐานข้อมูลแอสเซสหรือไม่ ถ้าปิดให้เปิด จากนั้นจะลบข้อมูลในฐานข้อมูลโดยใช้ค่าที่ผู้ใช้กรอกใน DStudentIDBox เป็นตัวระบุ Record ที่จะลบ



รูปที่ 3.13 แผนผัง Action การกดปุ่ม Delete

7) Action การเปลี่ยนแปลงของ Index ใน DoorcomboBox

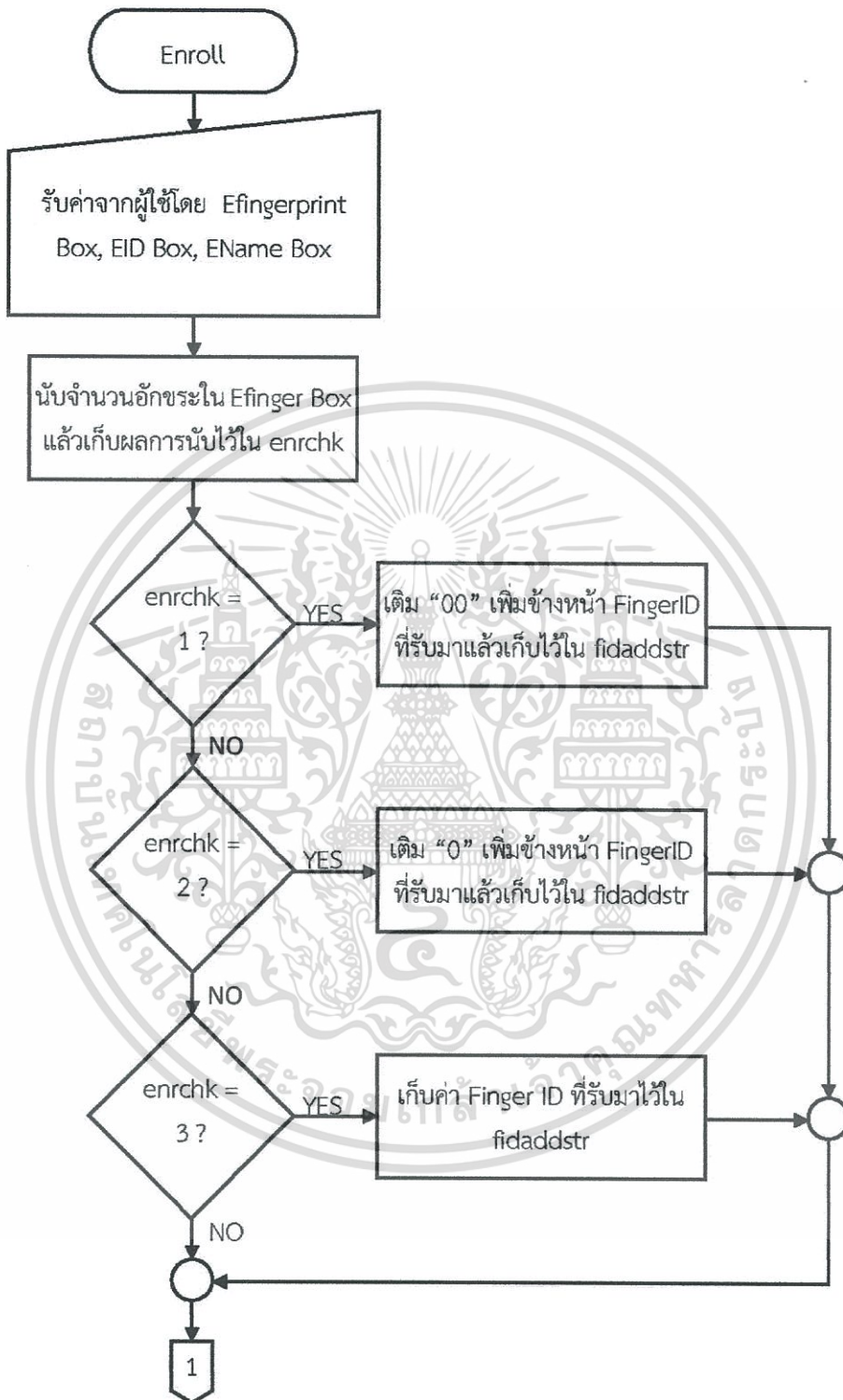
แสดงแผนผังการทำงานได้ดังรูปที่ 3.14 เมื่อมีการเลือกรายการ Serial Port ที่ DoorcomboBox จะทำให้ปุ่ม Connect สามารถทำงานได้ แต่ถ้ายังไม่เลือก Serial Port จะไม่สามารถกดปุ่ม Connect ได้



รูปที่ 3.14 แผนผัง Action การเปลี่ยนแปลงของ Index ใน DoorcomboBox

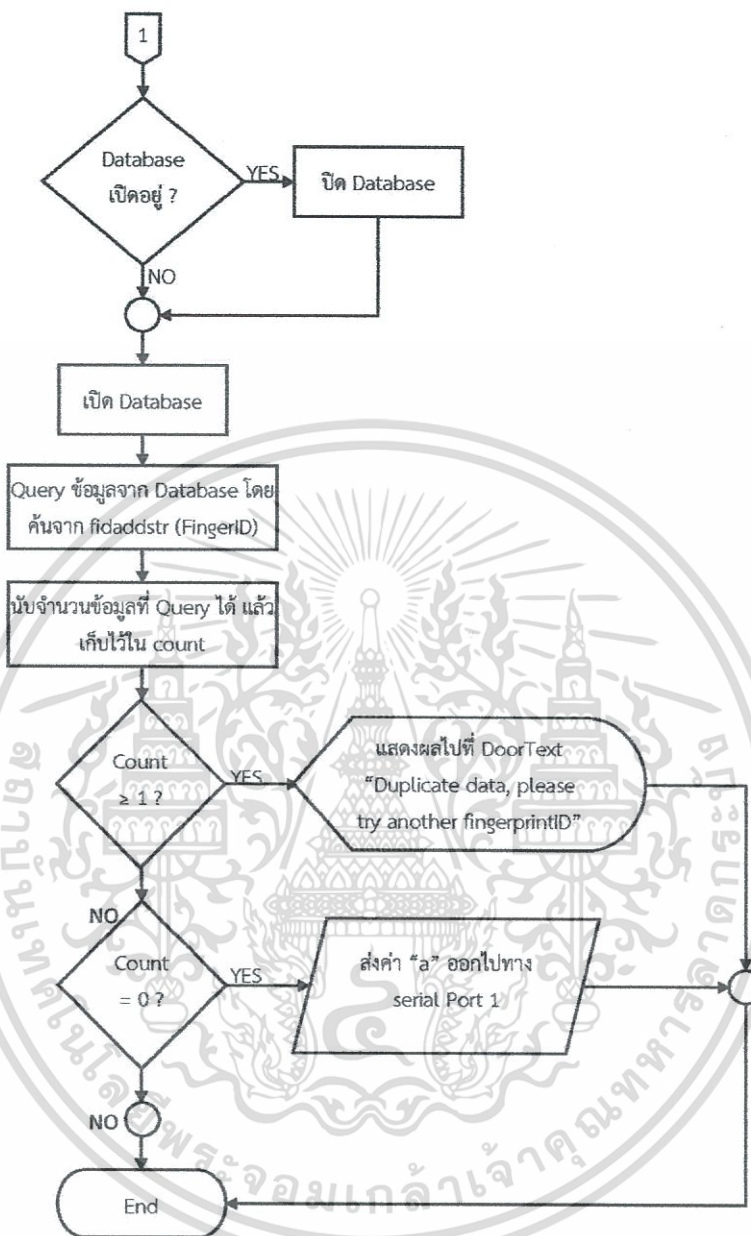
8) Action การกดปุ่ม Enroll

แสดงแผนผังการทำงานได้ดังรูปที่ 3.15 เมื่อกดปุ่ม Enroll ให้ทำการรับค่าจากผู้ใช้โดยใช้ EFingerBox, EIDBox และ ENameBox ในการรับค่า จากนั้นนับจำนวนอักขระใน EFingerBox แล้วเก็บผลการนับลงในตัวแปร enrchk แล้วทำการตรวจสอบค่า enrchk หากมีค่าเป็น 1 ให้เติม 00 เข้าไปหน้า FingerID ที่รับมาแล้วเก็บไว้ในตัวแปร fidaddstr หากมีค่าเป็น 2 ให้เติม 0 เข้าไปหน้า FingerID ที่รับมาแล้วเก็บไว้ในตัวแปร fidaddstr หากมีค่าเป็น 3 ให้เก็บ FingerID ไว้ในตัวแปร fidaddstr จากนั้นทำการตรวจสอบฐานข้อมูลว่าเปิดอยู่หรือไม่ หากเปิดอยู่ให้ปิดก่อนแล้วเปิดใหม่อีกครั้ง แต่ถ้าหากปิดอยู่ให้เปิด แล้วทำการ Query ข้อมูลจากฐานข้อมูลโดยใช้ค่าจาก fidaddstr ซึ่งเป็นค่า FingerID เป็นค่าสำคัญในการ Query ข้อมูล จากนั้นนับจำนวนข้อมูลที่ Query ได้แล้วเก็บผลการนับไว้ในตัวแปร count แล้วตรวจสอบค่า count ว่ามากกว่าหรือเท่ากับ 1 หรือไม่ หากใช่ ให้ทำการแสดงผลว่า Duplicate data, please try another fingerprint ID แล้วจบการทำงาน หากไม่ใช่ ให้ส่งค่า "a" ออกไปทาง Serial Port 1 เพื่อให้ Arduino เริ่มต้นลงทะเบียนลายนิ้วมือ แล้วจบการทำงาน



รูปที่ 3.15 แผนผัง Action การกดปุ่ม Enroll

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

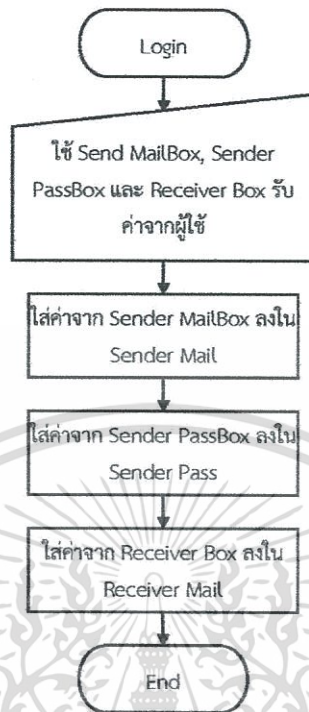


รูปที่ 3.15 แผนผัง Action การกดปุ่ม Enroll (ต่อ)

9) Action การกดปุ่ม Login

แสดงแผนผังการทำงานได้ดังรูปที่ 3.16 เมื่อกดปุ่ม Login ให้ใช้ SenderMailBox, SenderPassBox และ ReceiverMailBox รับค่าจากผู้ใช้ แล้วเก็บค่าจาก SenderMailBox ลงใน SenderMail, เก็บค่าจาก SenderPassBox ลงใน SenderPass และ เก็บค่าจาก ReceiverMailBox ลงใน ReceiverMail แล้วจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แผนผัง Action การกดปุ่ม Login

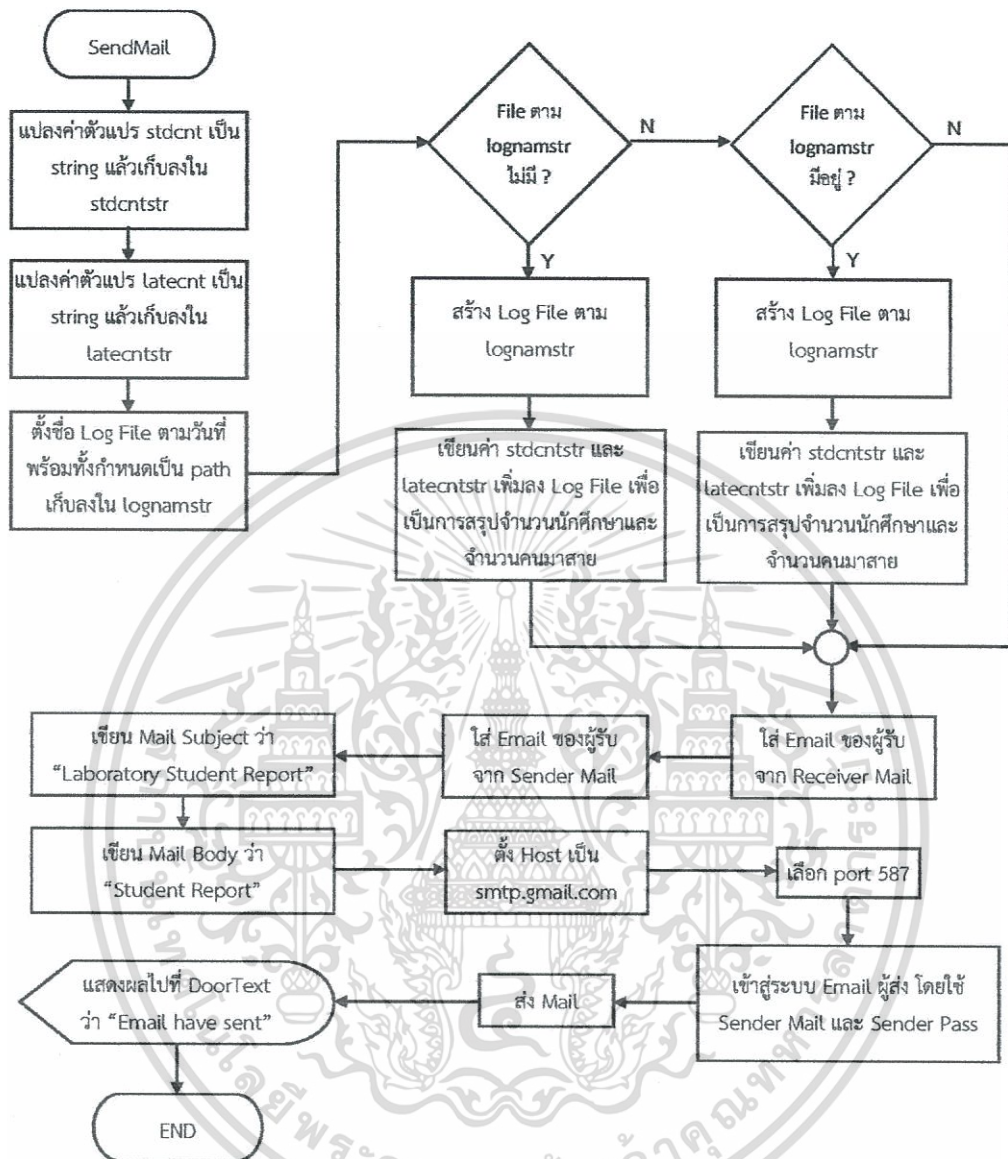
10) Action การกดปุ่ม Send Email

แสดงแผนผังการทำงานได้ดังรูปที่ 3.17 เมื่อกดปุ่ม Send Email ให้เริ่มต้นโดยการสรุปจำนวนนักศึกษาที่เข้าห้องและจำนวนนักศึกษาที่มาสายเพิ่มลงไปในแฟ้มลงบันทึกเข้าออกของวัน那一天 จากนั้นเริ่มต้นกระบวนการส่งอีเมลโดยการใส่อีเมลของผู้รับและผู้ส่ง จากนั้นเขียนหัวข้ออีเมลและเนื้อหาพร้อมทั้งแนบแฟ้มข้อมูลบันทึกการเข้าออกไปด้วย จากนั้นตั้ง Host เป็น smtp.gmail.com แล้วเลือก Port เป็น Port 587 แล้วเข้าสู่ระบบอีเมลผู้ส่ง จากนั้นทำการส่งอีเมลแล้วแสดงผลว่า “Email have sent.” แล้วจบการทำงาน

11) Action การดำเนินงานของนาฬิกา

แสดงแผนผังการทำงานได้ดังรูปที่ 3.18 เมื่อนาฬิกามีการเปลี่ยนเวลา (วินาที) ให้เก็บวันที่ปัจจุบันไว้ใน sdate และเก็บเวลาปัจจุบันไว้ใน stime แล้วแสดงค่าทั้งสองไปที่ label12 จากนั้นให้เก็บค่าชั่วโมงปัจจุบันไว้ใน mtime แล้วนำมาเปรียบเทียบว่าเท่ากับ 20 หรือไม่ ถ้าไม่เท่าให้เคลียร์ค่าตัวแปร actchk แล้วจบการทำงาน แต่ถ้าเท่าตรวจสอบค่า actchk ว่ามีค่าหรือไม่ ถ้ามีค่าให้จบการทำงาน แต่ถ้าไม่มีค่าให้สรุปจำนวนนักศึกษาที่เข้าห้องและจำนวนนักศึกษาที่มาสายเพิ่มลงไปแฟ้มลงบันทึกเข้าออกของวัน那一天 แล้วเริ่มต้นกระบวนการส่งอีเมลโดยการใส่อีเมลของผู้รับและผู้ส่ง จากนั้นเขียนหัวข้ออีเมลและเนื้อหาพร้อมทั้งแนบ แฟ้มลงบันทึกเข้าออก ไปด้วย จากนั้นตั้ง Host เป็น smtp.gmail.com แล้วเลือก Port เป็น Port 587 แล้วเข้าสู่ระบบอีเมลผู้ส่ง จากนั้นทำการส่งอีเมลแล้วแสดงผลว่า “Email have sent.” แล้วจบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

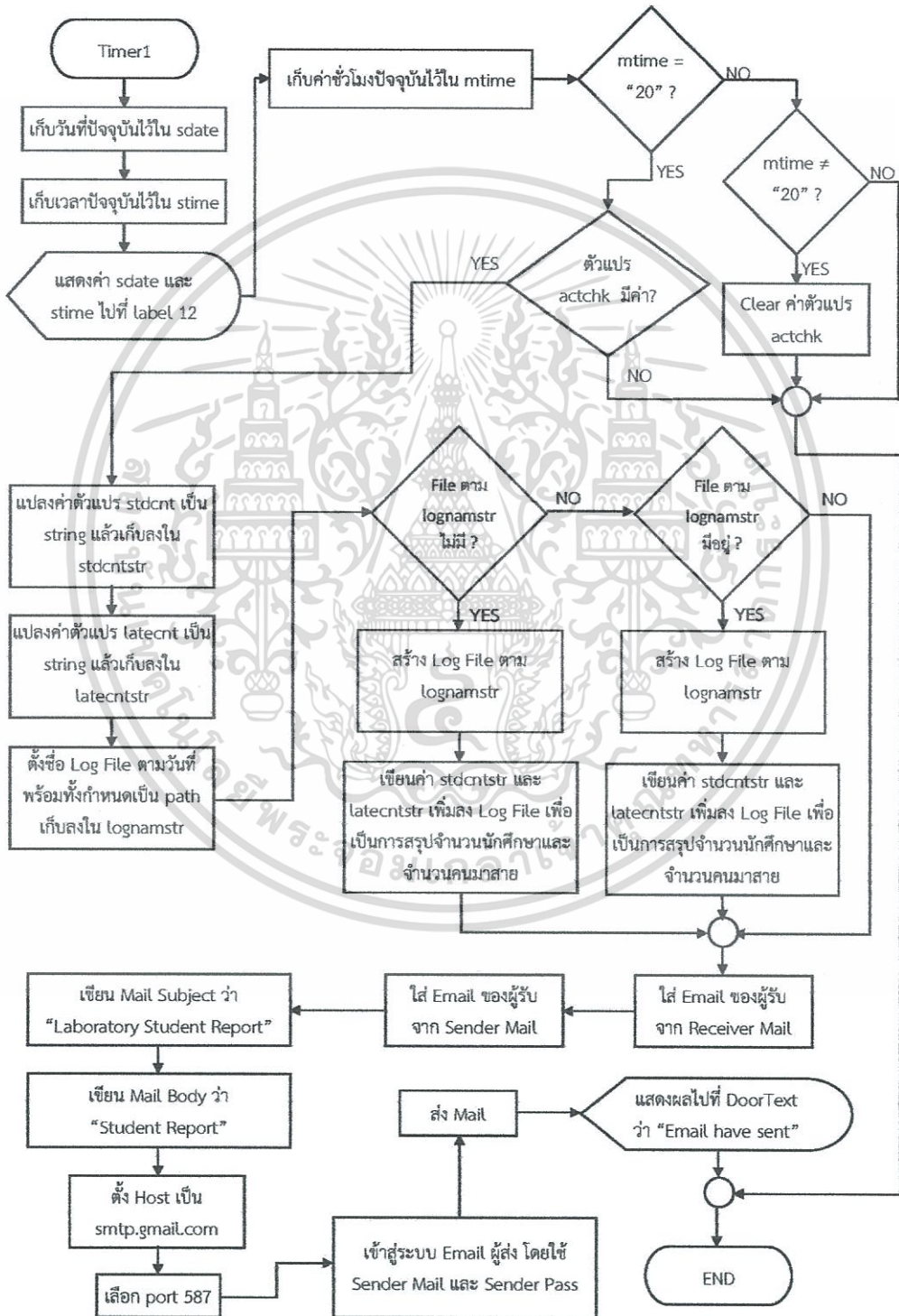


รูปที่ 3.17 แผนผัง Action การกดปุ่ม Send Email

12) Action การกดปุ่ม Unlock

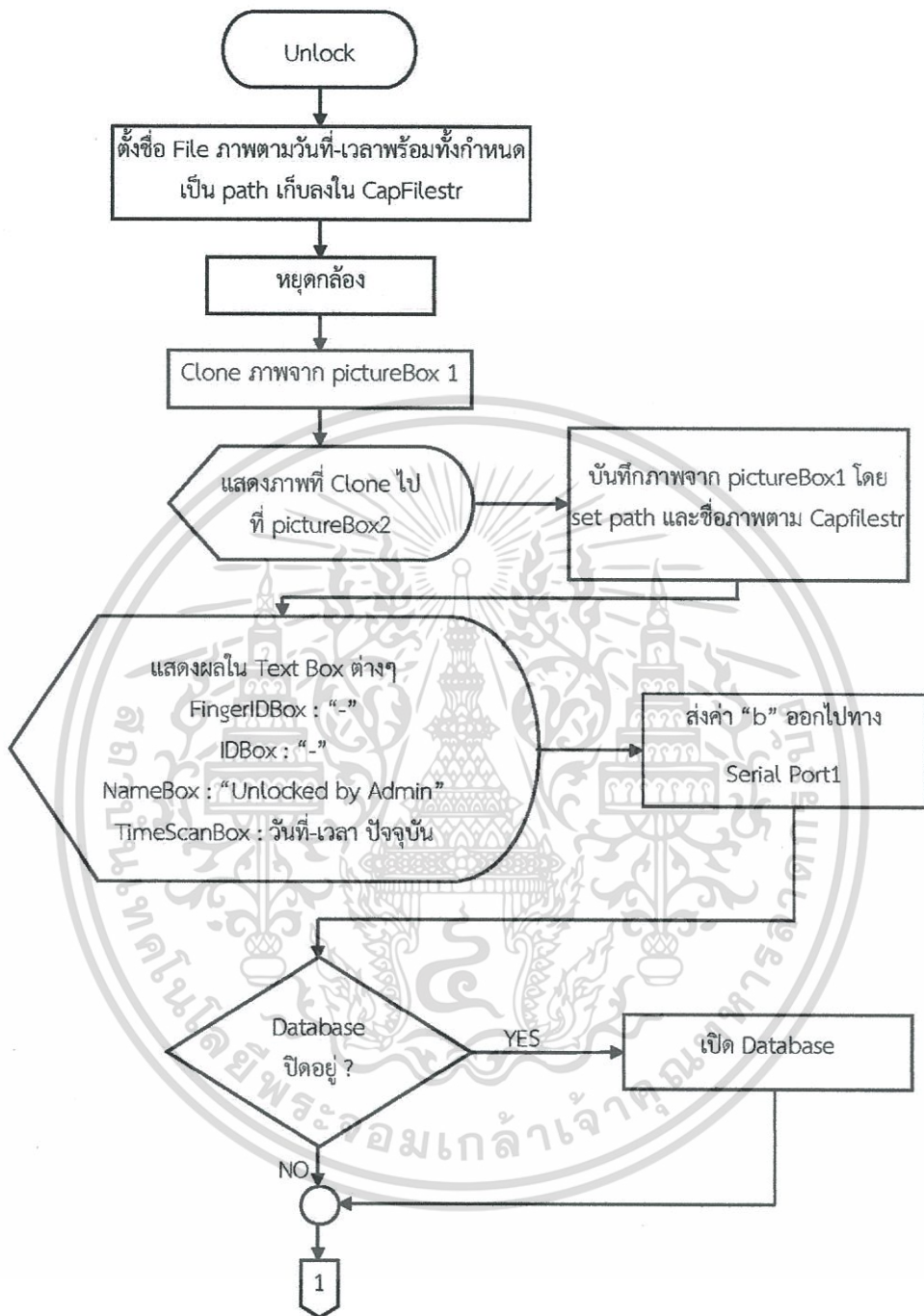
แสดงแผนผังการทำงานได้ดังรูปที่ 3.19 เมื่อกดปุ่ม Unlock แล้วให้ตั้งชื่อแฟ้มข้อมูลภาพตามวันที่ เวลา พร้อมทั้งกำหนดเป็น path เก็บลงใน CapFilestr จากนั้นหยุดกล่องแล้วทำการ Clone ภาพจาก PictureBox1 ไปแสดงที่ PictureBox2 แล้วบันทึกภาพจาก PictureBox2 โดยกำหนดที่อยู่และชื่อภาพตาม CapFilestr จากนั้นแสดงผลใน NameBox ว่า Unlocked by Admin. และแสดงวันที่ เวลาปัจจุบันลงใน TimeScanBox แล้วส่งค่า "b" ออกไปทาง Serial Port 1 เพื่อให้ Arduino ทำการปลดล็อกประตู จากนั้นตรวจสอบฐานข้อมูลว่าปิดอยู่หรือไม่ ถ้าปิดอยู่ให้เปิดก่อน จากนั้นทำการบันทึกข้อมูลลงในฐานข้อมูล โดยบันทึกลงในฟิลด์ต่างๆ ดังนี้ ฟิลด์ FingerID ให้บันทึกว่า ADMIN, ฟิลด์ StudentID ให้บันทึกว่า ADMIN, ฟิลด์ FullName เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้บันทึกว่า ADMIN, ฟิลด์ DayTime ให้บันทึกวันที่ เวลาปัจจุบัน และ ฟิลด์ Captured ให้บันทึก Path ตาม CapFilestr จากนั้นให้ตั้งชื่อแฟ้มลงบันทึกเข้าออกตามวันที่ พร้อมทั้งกำหนดเป็น path เก็บลงใน lognamstr แล้วตรวจสอบว่าแฟ้มลงบันทึกเข้าออกนี้มีอยู่แล้วหรือไม่ หากยังไม่มีให้สร้างแฟ้มข้อมูลก่อน แล้วจึงทำการบันทึกข้อมูลลงแฟ้มลงบันทึกเข้าออก แล้วจึงจบการทำงาน



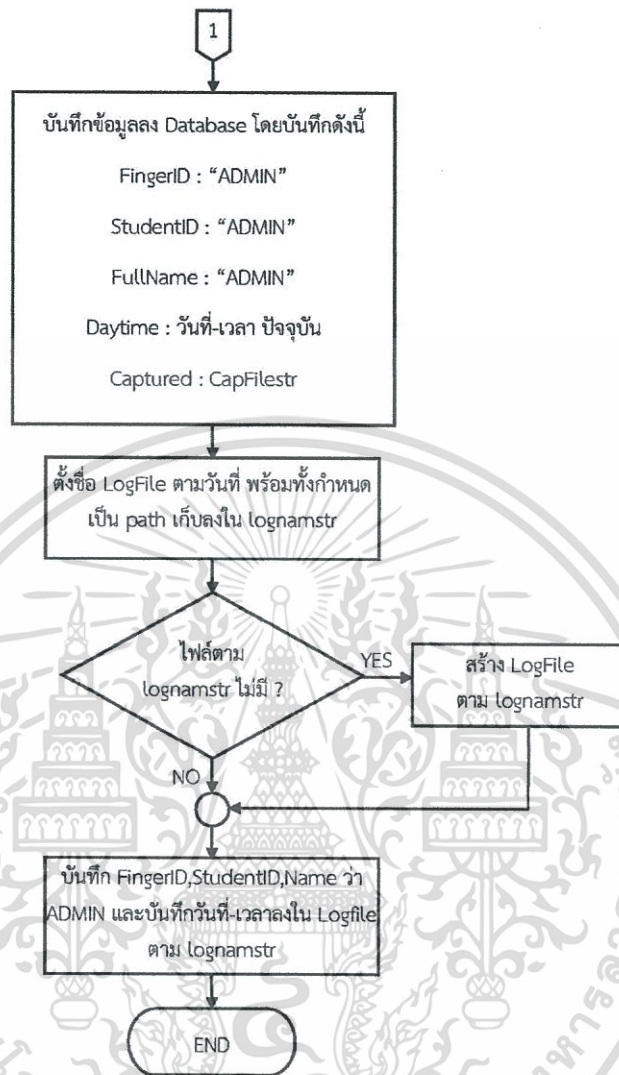
รูปที่ 3.18 แผนผัง Action การดำเนินงานของนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเชิงนี้เพื่อการรคกชเท่านั้น ไม่นุญให้เดีนาไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แผนผัง Action การกดปุ่ม Unlock

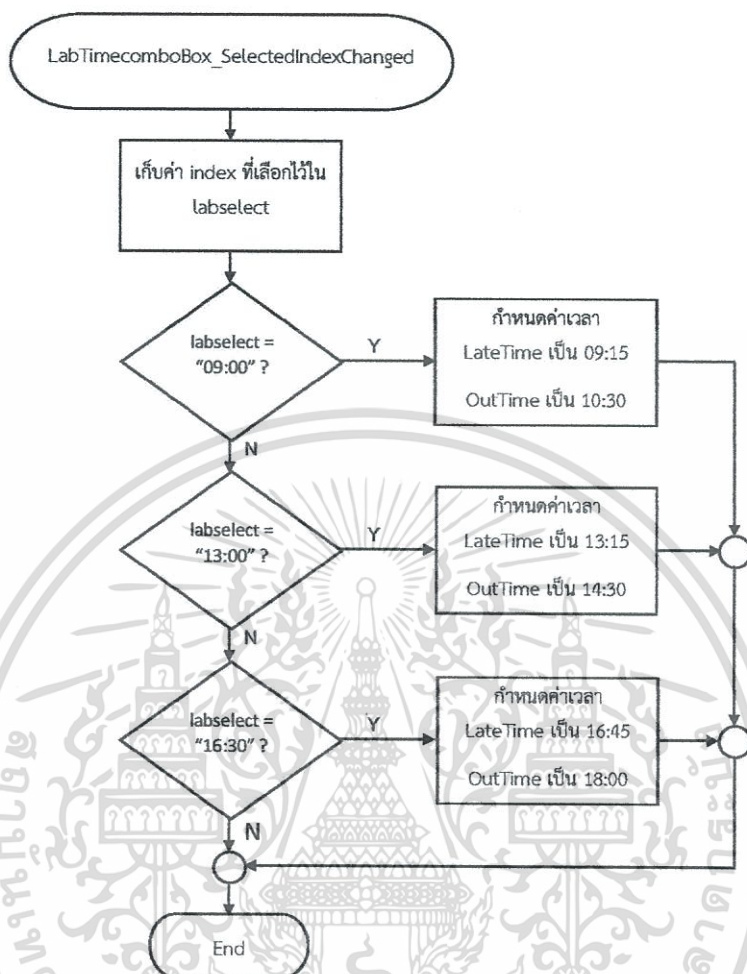
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แผนผัง Action การกดปุ่ม Unlock (ต่อ)

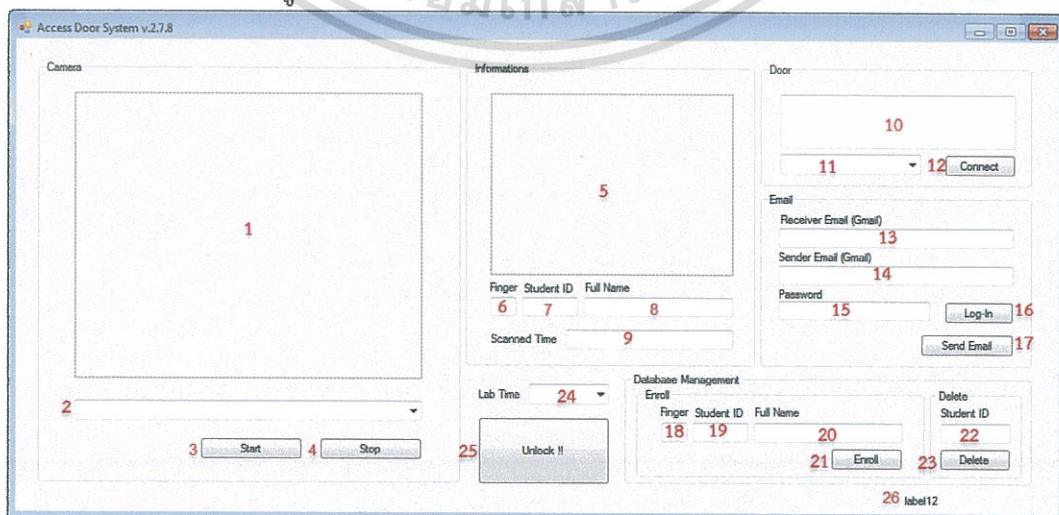
13) Action การเปลี่ยนแปลงของ Index ใน LabTimecomboBox แสดงแผนผังการทำงานได้ดังรูปที่ 3.20 เมื่อเลือกเวลาปฏิบัติการไปที่เวลา 09:00 น. ระบบจะทำการตั้ง OutTime (เวลาออก) ไว้ที่เวลา 10:30 น. และตั้ง LateTime (เวลาสาย) ไว้ที่เวลา 09:15 น. เมื่อเลือกเวลาปฏิบัติการไปที่เวลา 13:00 น. ระบบจะทำการตั้ง OutTime (เวลาออก) ไว้ที่เวลา 14:30 น. และตั้ง LateTime (เวลาสาย) ไว้ที่เวลา 13:15 น. เมื่อเลือกเวลาปฏิบัติการไปที่เวลา 16:30 น. ระบบจะทำการตั้ง OutTime (เวลาออก) ไว้ที่เวลา 16:45 น. และตั้ง LateTime (เวลาสาย) ไว้ที่เวลา 18:00 น.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 แผนผัง Action การเปลี่ยนแปลงของ Index ใน LabTimecomboBox

ในส่วนของการจัดวางวัตถุต่างๆของโปรแกรม Graphic Interface จะทำการจัดวางดังแสดงในรูปที่ 3.21 ซึ่งหน้าที่ของแต่ละส่วนสามารถแสดงได้ดังตารางที่ 3.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่สู่สาธารณะ
รูปที่ 3.21 หน้าจอโปรแกรม Graphic Interface

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 หน้าทีของวัตถุต่างๆของโปรแกรม

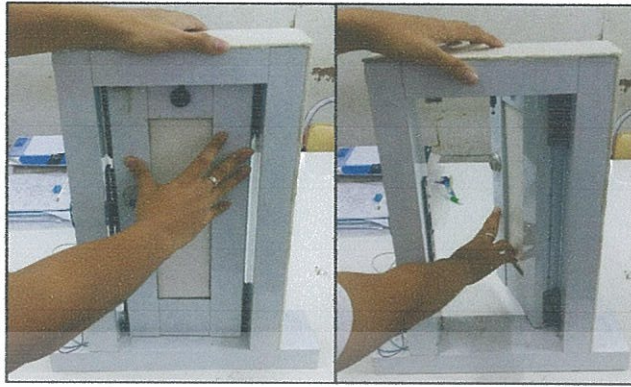
หมายเลข	ชื่อวัตถุ	การทำงาน
1	pictureBox1	แสดงภาพจากกล้อง
2	CamcomboBox1	สำหรับเลือกกล้องที่จะทำการแสดงภาพ
3	CamStart	เริ่มแสดงภาพจากกล้องที่เลือกไว้จาก CamcomboBox1
4	CamStop	หยุดแสดงภาพจากกล้อง
5	pictureBox2	แสดงภาพที่ทำการถ่ายเก็บไว้
6	FingerIDBox	แสดงรหัสลายนิ้วมือที่ทำการกราดตรวจ
7	IDBox	แสดงรหัสนักศึกษาของลายนิ้วมือที่ทำการกราดตรวจ
8	NameBox	แสดงรหัสชื่อของลายนิ้วมือที่ทำการกราดตรวจ
9	TimeScanBox	แสดงวันที่-เวลาที่ทำการกราดตรวจ
10	DoorText	แสดงผลตอบสนองของระบบ
11	DoorcomboBox	สำหรับเลือก Serial Port ที่ต้องการเชื่อมต่อ
12	ConnectBut	เชื่อมต่อกับ Serial Port ที่เลือกไว้จาก DoorcomboBox
13	ReceiverBox	สำหรับกรอก Email Address ผู้รับ
14	SenderMailBox	สำหรับกรอก Email Address ผู้ส่ง
15	SenderPassBox	สำหรับกรอกรหัสผ่านของ ผู้ส่ง
16	LoginButt	จำค่า Email และ รหัสผ่าน เพื่อใช้ในการส่ง Email
17	SendMailButt	กดเพื่อส่ง Email
18	EFingerBox	สำหรับกรอกรหัสลายนิ้วมือที่ต้องการลงทะเบียน
19	EStudentIDBox	สำหรับกรอกรหัสนักศึกษาที่ต้องการลงทะเบียน
20	ENameBox	สำหรับกรอกชื่อที่ต้องการลงทะเบียน
21	EnrollBut	กดเพื่อเริ่มต้นการลงทะเบียน
22	DStudentIDBox	สำหรับกรอกรหัสนักศึกษาที่ต้องการลบข้อมูล
23	DeleteButt	กดเพื่อลบข้อมูลลายนิ้วมือ
24	LabTimecomboBox	สำหรับเลือกเวลาเริ่มวิชาปฏิบัติการ
25	UnlockBut	กดเพื่อปลดล็อกประตู
26	label12	แสดงวันที่-เวลาปัจจุบัน

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 ประตูกлонควบคุมด้วยกระแสไฟฟ้า

ในสถานะที่กลอนประตูไม่มีการจ่ายไฟเข้า กลอนจะล็อก แต่เมื่อมีไฟฟ้า 12 โวลต์เข้า จะทำให้กลอนประตูปลดล็อก รูปประตูแสดงได้ดังรูปที่ 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 ประตูก่อนควบคุมด้วยกระแสไฟฟ้า

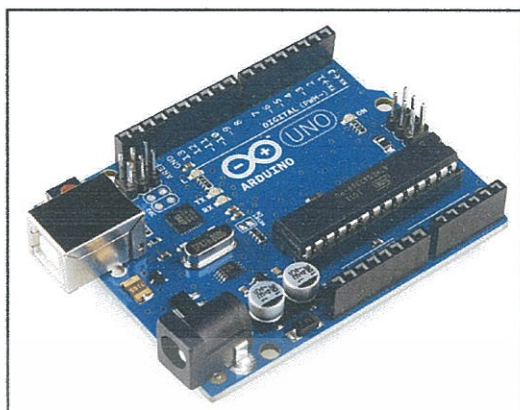
3.2.2 Arduino Uno R3

Arduino Uno R3 แสดงได้ดังรูปที่ 3.23 ซึ่งเป็น Microcontroller board ที่ใช้ ATmega328 เป็น MCU หลัก ซึ่งตัวนี้จะมีขา Digital 14 ขา INPUT/OUTPUT (สามารถทำเป็น PWM ได้ถึง 6 ขา) และมีขา Analog INPUT ได้อีก 6 ขา, ทำงานที่ความถี่ 16 MHz มี USB Connector และ Power Jack DC ซึ่งแนวความคิดของ Arduino Board นี้ทำมาเพื่อความสะดวกในการเชื่อมต่อเข้ากับคอมพิวเตอร์ สามารถต่อ USB เข้ากับช่อง COM PORT ทำให้ดำเนินงานโปรแกรมที่ Board ได้

ตารางที่ 3.4 คุณสมบัติของ Arduino Uno R3

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 kB
SRAM	2 kB
EEPROM	1 kB
Clock Speed	16 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 Arduino Uno R3

3.2.3 Relay HRS1H-S-DC3V

Relay แสดงได้ดังรูปที่ 3.24 เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่ง ทำหน้าที่ตัดต่อวงจรแบบเดียวกับสวิตช์ โดยควบคุมการทำงานด้วยไฟฟ้า

3.2.3.1 รายละเอียดคุณสมบัติรีเลย์ แสดงได้ดังนี้

- | | | |
|---------------------------|------|-----|
| 1. Coil nominal | 3 | VDC |
| 2. Coil resistance +/-10% | 45 | Ohm |
| 3. Operate voltage | 2.25 | VDC |
| 4. Release voltage | 0.30 | VDC |

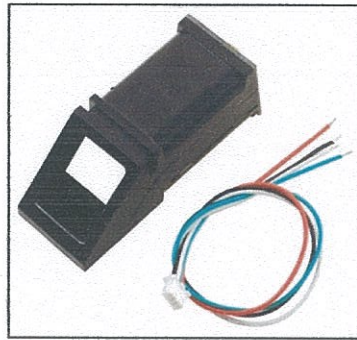


รูปที่ 3.24 Relay HRS1H-S-DC3V

3.2.4 Adafruit Optical Fingerprint Sensor

3.2.4.1 รายละเอียดอุปกรณ์

โมดูลเซ็นเซอร์ตรวจสอบลายนิ้วมือแบบออปติคัล (Optical Fingerprint Sensor) แสดงได้ดังรูปที่ 3.25 สามารถตรวจจับและตรวจสอบลายนิ้วมือ สามารถเก็บข้อมูลและค้นหาได้สูงสุด 162 ลายนิ้วมือบนหน่วยความจำภายใน ทำงานที่แรงดัน 3.6 ถึง 5 โวลต์ กินกระแสสูงสุด 150mA เชื่อมต่อแบบอนุกรม UART ค่า Default Baud Rate คือ 57600 bps



รูปที่ 3.25 อุปกรณ์ Fingerprint Sensor

ตารางที่ 3.5 ขาสัญญาณในการเชื่อมต่อของ Fingerprint Sensor

ชื่อ	คำอธิบาย	สีสาย
Vcc	ไฟเลี้ยง 5V DC	แดง
Tx	ขาส่งข้อมูล	เขียว
Rx	ขารับข้อมูล	ขาว
GND	กราวด์	ดำ

3.2.5 Webcam Camera

3.2.5.1 รายละเอียดอุปกรณ์

กล้องสามารถหมุนปรับองศาได้ และมีไฟLED 6 ดวง เชื่อมต่อผ่าน USB แสดงได้ดังรูปที่ 3.26



รูปที่ 3.26 Webcam Camera

3.2.6 คอมพิวเตอร์

3.2.6.1 รายละเอียดอุปกรณ์

คอมพิวเตอร์ที่ใช้งาน แสดงได้ดังรูปที่ 3.27 ใช้ระบบปฏิบัติการ Microsoft Windows 7 และติดตั้งโปรแกรม Microsoft Visual Studio 2013, Microsoft Access 2013 และ Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 คอมพิวเตอร์ที่ใช้งาน

3.3 การจัดเก็บผลการทดลอง

3.3.1 อัตราการล้มเหลวในการยึดเก็บสัญญาณ (Failure to Capture Rate (FTC))

หรืออัตราการล้มเหลวในการเก็บข้อมูล (Failure to Acquire Rate (FTA)) เกิดจากความผิดพลาดที่ไม่สามารถรับสัญญาณลักษณะเฉพาะไปโอเมตริกของบุคคลได้ ทั้งๆที่บุคคลนั้นได้แสดงตัวอยู่ที่เครื่องรับสัญญาณลักษณะเฉพาะไปโอเมตริกแล้ว โดยค่านี้จะวัดโดยการกำหนดช่วงเวลาที่จำกัดค่าหนึ่ง ถ้าเกินเวลาจำกัดนี้ถือว่าล้มเหลวในการเก็บข้อมูล ซึ่งปัญหานี้เกิดจากคุณภาพของสัญญาณลักษณะเฉพาะไปโอเมตริกที่อ่านค่าเข้ามาไม่มีความคมชัดเพียงพอที่จะนำไปใช้งานหรืออาจเกิดการบกพร่องทางเทคนิคที่ตัวรับสัญญาณ การเสื่อมสภาพ หรืออุปกรณ์ขัดข้องจากสภาพแวดล้อม เป็นต้น สามารถคำนวณได้ตามสมการที่ 3.1

$$\frac{\text{Total of Failure to Captured}}{\text{Total of scanned}} \times 100\% \quad (3.1)$$

3.3.2 อัตราการระบุ (Identification Rate)

อัตราการระบุ เป็นค่าอัตราความแม่นยำในการระบุตัวของการค้นหาแบบหนึ่งต่อหลาย โดยส่วนใหญ่จะเป็นอัตราความถูกต้องที่จะหาบุคคลให้เจอในจำนวนบุคคลที่จำกัด อาทิเช่น ระบบระบุตัวบุคคลมีอัตราการระบุอยู่ที่ 95% เมื่อเลือก 8 บุคคลแรก หมายถึง ระบบมีความสามารถในการระบุตัวค้นหาบุคคลจากข้อมูลไปโอเมตริกให้เจอใน 8 บุคคลแรก โดยเลือกจากบุคคล 8 บุคคลที่มีคะแนนการเปรียบเทียบสูงสุด และให้ความแม่นยำถึงร้อยละ 95 เป็นต้น

$$\frac{\text{Total of True Identified}}{\text{Total of Identified}} \times 100\% \quad (3.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

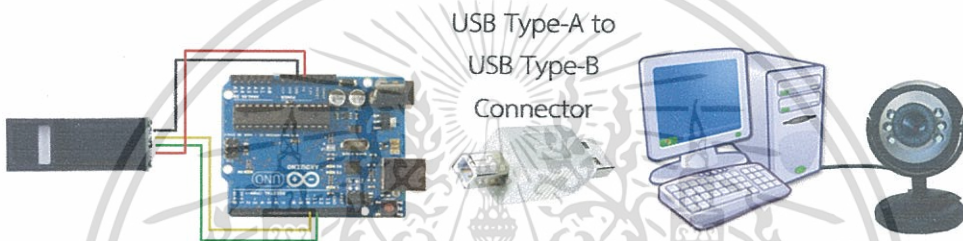
บทที่ 4

ผลการทดลอง

4.1 การทดลองการยืนยันตัวตน

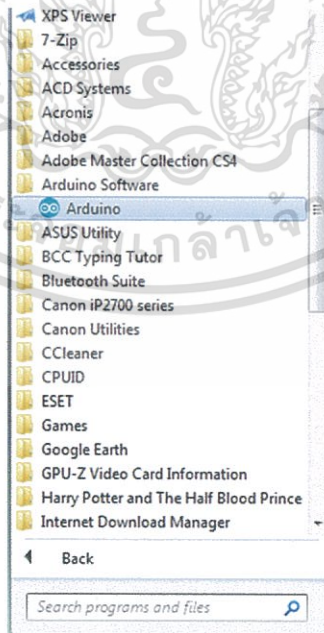
การทดลองในส่วนนี้เป็นการทดลองการยืนยันตัวตนเมื่อทำการกราดตรวจลายนิ้วมือและวัดค่าความแม่นยำของ Fingerprint Sensor โดยทำการกราดตรวจลายนิ้วมือของนักศึกษาชั้นปีที่สอง ภาควิชาวิศวกรรมโทรคมนาคม ก่อนเข้าและออกจากห้องปฏิบัติการ เพื่อเป็นการทดสอบว่าการยืนยันตัวตนและบันทึกข้อมูลถูกต้องหรือไม่ ซึ่งมีขั้นตอนการทดลองดังนี้

1. ทำการเชื่อมต่ออุปกรณ์ ดังรูปที่ 4.1



รูปที่ 4.1 การเชื่อมต่ออุปกรณ์

2. ทำการเปิดโปรแกรม Arduino IDE ที่มีอยู่ในคอมพิวเตอร์ โดยการเข้าจาก Start Menu -> All Programs -> Arduino Software -> Arduino ดังรูปที่ 4.2



รูปที่ 4.2 การเปิดโปรแกรม Arduino

3. เขียนคำสั่ง Arduino ตามภาคผนวก ก คำสั่งสำหรับ Arduino (Microcontroller)
4. ทำการอัปโหลดคำสั่งลง Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

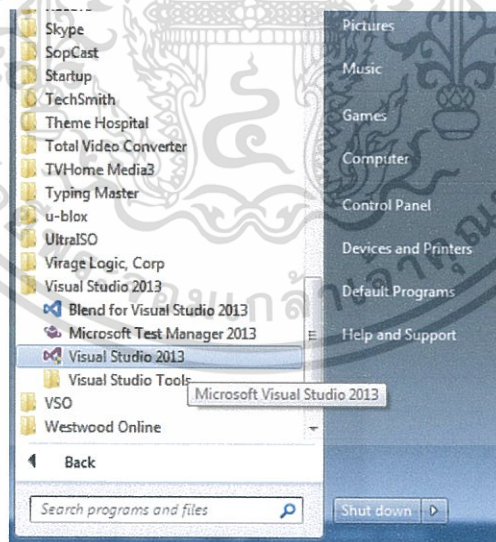
5. เปิดโปรแกรม Microsoft Access ที่มีอยู่ในคอมพิวเตอร์ โดยการเข้าจาก Start Menu -> All Programs -> Microsoft Office -> Access ดังรูปที่ 4.3



รูปที่ 4.3 การเปิดโปรแกรม Microsoft Access

6. ทำการออกแบบฐานข้อมูลตามหัวข้อที่ 3.1.2.1

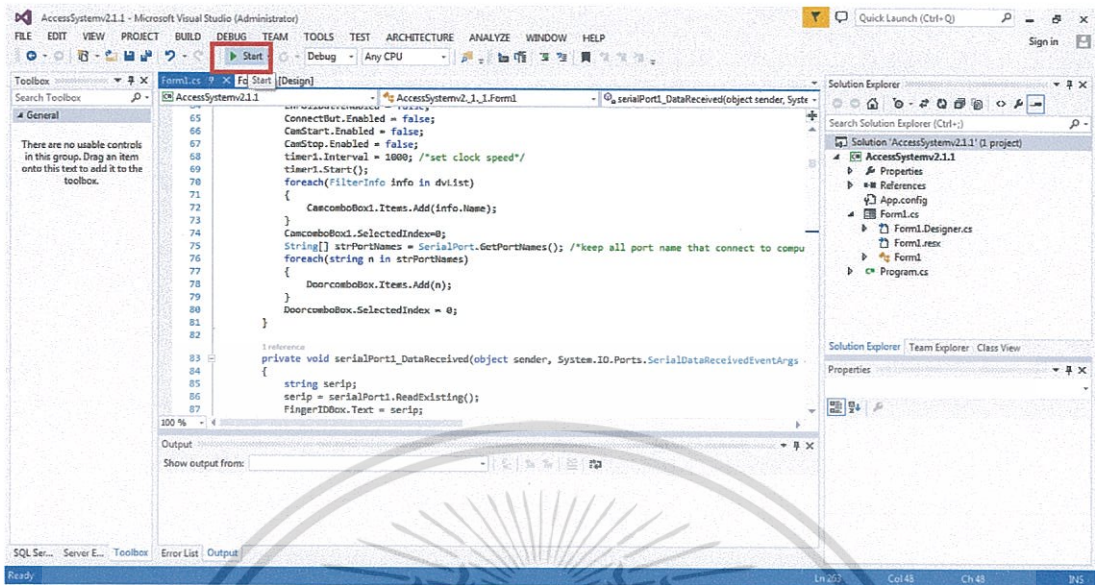
7. เปิดโปรแกรม Microsoft Visual Studio ที่มีอยู่ในคอมพิวเตอร์ โดยการเข้าจาก Start Menu -> All Programs -> Visual Studio -> Visual Studio ดังรูปที่ 4.4



รูปที่ 4.4 การเปิดโปรแกรม Microsoft Visual Studio

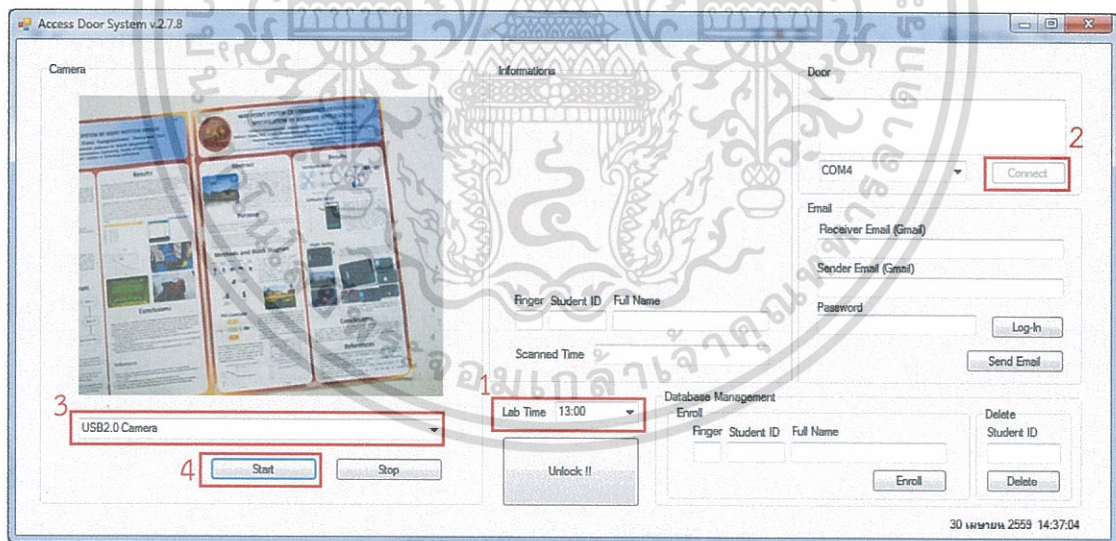
8. ทำการออกแบบหน้าต่างโปรแกรม แล้วเขียนคำสั่งตามภาคผนวก ก คำสั่ง C# สำหรับโปรแกรมที่ดำเนินงานบนคอมพิวเตอร์ ดำเนินงานโปรแกรมที่เขียนโดยการกดปุ่ม Start ดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การดำเนินงานโปรแกรม

9. ทำการเลือกเวลาปฏิบัติการแล้วเปิด Serial Port โดยการกดปุ่ม Connect และเปิดกล้อง โดยทำการเลือกกล้องที่ต้องการเปิดแล้วกดปุ่ม Start ดังรูปที่ 4.6



รูปที่ 4.6 ขั้นตอนการเปิดการสื่อสารระหว่างอุปกรณ์และโปรแกรม

เริ่มทำการกราดตรวจลายนิ้วมือ โดยเมื่อทำการกราดตรวจแล้วพบลายนิ้วมือทั้งใน Fingerprint Sensor และในฐานข้อมูล ระบบจะทำการบันทึกข้อมูลผู้กราดตรวจลายนิ้วมือ, วันที่-เวลาที่กราดตรวจและภาพใบหน้าของผู้กราดตรวจลายนิ้วมือแล้วเก็บไว้ในฐานข้อมูลดังรูปที่ 4.8 และเพิ่มลงบันทึกเข้าออกดังรูปที่ 4.7 โดยเพิ่มลงบันทึกเข้าออก จะไม่มีการบันทึกภาพใบหน้าลงไปด้วย จากนั้นระบบจะสั่ง Arduino ให้ทำการปลดล็อกประตู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No	StudentID	Fullname	DayTime	Captured
126	57011491	อรชิวา สิริเศรษฐนันท์	1 เม.ย. 2559 18:36:54	Out
103	57011215	ศรัทธา เกษศิริชัย	1 เม.ย. 2559 18:37:11	Out
129	57011516	อัญชลี ใจซื่อ	1 เม.ย. 2559 18:37:24	Out
090	57011046	รมณิศา สุนทร	1 เม.ย. 2559 18:37:49	Out
096	57011137	วฤนดา ปทุมสกุลสุนทร	1 เม.ย. 2559 18:38:03	Out
092	57011070	รุ่งรอง โชติชาลาวัฒนกุล	1 เม.ย. 2559 18:38:14	Out
110	57011299	สนันไพา สิงห์โตทอง	1 เม.ย. 2559 18:38:44	Out
109	57011294	ศดาบย ตระกูลวงษาณนท์	1 เม.ย. 2559 18:38:51	Out
122	57011458	อริวิตร เบียมทอง	1 เม.ย. 2559 18:39:34	Out
120	57011437	เสฏฐวิษณุ สุกิจพิทยานนท์	1 เม.ย. 2559 18:39:42	Out
108	57011261	ศุภกากร จริยาขงค์	1 เม.ย. 2559 18:39:56	Out
116	57011408	ศุภษา เชื้อบุญชัย	1 เม.ย. 2559 18:40:22	Out
112	57011337	สันติสุข คชแก้ว	1 เม.ย. 2559 18:40:27	Out
121	57011444	หทัยทิพย์ บุตดี	1 เม.ย. 2559 18:40:33	Out
128	57011513	อัญญา วรยานนท์	1 เม.ย. 2559 18:41:10	Out
123	57011474	อภิญญา เอื้องอุดมสิน	1 เม.ย. 2559 18:41:20	Out
127	57011509	อรชิวา ปุ่มเพ็ชร	1 เม.ย. 2559 18:41:32	Out

Total (ln) : 43
Late : 0

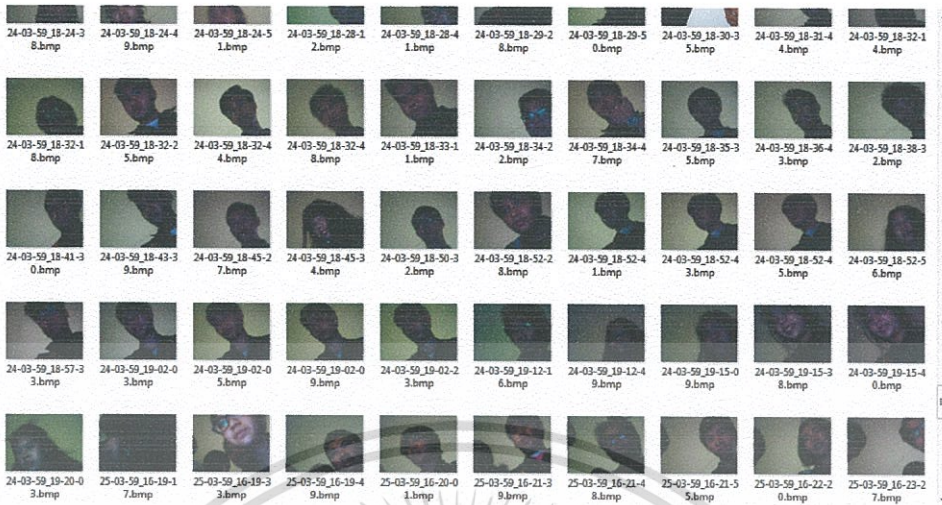
รูปที่ 4.7 ตัวอย่างแฟ้มลงบันทึกเข้าออก

No	FingerID	StudentID	Fullname	DayTime	Captured
768	089	57011044	รริกร เนตรอารสง	1 เม.ย. 2559 18:25:55	C:\Access System\Captured\01-04-59_18-25-55.bmp
769	117	57011442	ศุภกรชัย เชื้อเงิน	1 เม.ย. 2559 18:28:29	C:\Access System\Captured\01-04-59_18-28-29.bmp
770	115	57011405	ศุภนิลา สิงหนานวิวัฒน์	1 เม.ย. 2559 18:32:51	C:\Access System\Captured\01-04-59_18-32-50.bmp
771	100	57011186	วิษณุ สุวิวัฒน์	1 เม.ย. 2559 18:32:59	C:\Access System\Captured\01-04-59_18-32-59.bmp
772	140	57011429	ศุภวิวัฒน์ อภิศกุลโรจน์	1 เม.ย. 2559 18:33:17	C:\Access System\Captured\01-04-59_18-33-17.bmp
773	101	57011197	ศิริพงษ์ แก้วมณี	1 เม.ย. 2559 18:33:39	C:\Access System\Captured\01-04-59_18-33-38.bmp
774	099	57011187	วิศุฒ สิงแก้ว	1 เม.ย. 2559 18:35:50	C:\Access System\Captured\01-04-59_18-35-30.bmp
775	124	57011481	อภิศวัฒน์ แสงทองดี	1 เม.ย. 2559 18:36:26	C:\Access System\Captured\01-04-59_18-36-26.bmp
776	125	57011491	อรชิวา สิริเศรษฐนันท์	1 เม.ย. 2559 18:36:54	C:\Access System\Captured\01-04-59_18-36-53.bmp
777	103	57011215	ศรัทธา เกษศิริชัย	1 เม.ย. 2559 18:37:11	C:\Access System\Captured\01-04-59_18-37-10.bmp
778	129	57011516	อัญชลี ใจซื่อ	1 เม.ย. 2559 18:37:24	C:\Access System\Captured\01-04-59_18-37-24.bmp
779	090	57011046	รมณิศา สุนทร	1 เม.ย. 2559 18:37:49	C:\Access System\Captured\01-04-59_18-37-49.bmp
780	096	57011137	วฤนดา ปทุมสกุลสุนทร	1 เม.ย. 2559 18:38:03	C:\Access System\Captured\01-04-59_18-38-03.bmp
781	092	57011070	รุ่งรอง โชติชาลาวัฒนกุล	1 เม.ย. 2559 18:38:14	C:\Access System\Captured\01-04-59_18-38-14.bmp
782	110	57011299	สนันไพา สิงห์โตทอง	1 เม.ย. 2559 18:38:44	C:\Access System\Captured\01-04-59_18-38-44.bmp
783	109	57011294	ศดาบย ตระกูลวงษาณนท์	1 เม.ย. 2559 18:38:51	C:\Access System\Captured\01-04-59_18-38-51.bmp
784	122	57011458	อริวิตร เบียมทอง	1 เม.ย. 2559 18:39:34	C:\Access System\Captured\01-04-59_18-39-34.bmp
785	120	57011437	เสฏฐวิษณุ สุกิจพิทยานนท์	1 เม.ย. 2559 18:39:42	C:\Access System\Captured\01-04-59_18-39-42.bmp
786	108	57011261	ศุภกากร จริยาขงค์	1 เม.ย. 2559 18:39:56	C:\Access System\Captured\01-04-59_18-39-55.bmp
787	116	57011408	ศุภษา เชื้อบุญชัย	1 เม.ย. 2559 18:40:22	C:\Access System\Captured\01-04-59_18-40-22.bmp
788	112	57011337	สันติสุข คชแก้ว	1 เม.ย. 2559 18:40:27	C:\Access System\Captured\01-04-59_18-40-27.bmp
789	121	57011444	หทัยทิพย์ บุตดี	1 เม.ย. 2559 18:40:33	C:\Access System\Captured\01-04-59_18-40-33.bmp

รูปที่ 4.8 ตัวอย่างการบันทึกผลในฐานข้อมูล

จากการทดลองข้างต้นสามารถสรุปได้ว่า เมื่อมีการกราดตรวจลายนิ้วมือแล้วพบลายนิ้วมือ ระบบจะทำการแสดงและบันทึกข้อมูลของลายนิ้วมือ, วันที่-เวลาที่กราดตรวจ และรูปภาพใบหน้าของผู้กราดตรวจลายนิ้วมือแล้วเก็บไว้ พร้อมทั้งประมวลผลว่าเป็นการเข้าเรียนสายหรือไม่ หรือเป็นการออกเรียนดังรูปที่ 4.9 โดยการเก็บบันทึกผลนั้นจะแบ่งเก็บเป็น 2 ส่วนคือ ส่วนของฐานข้อมูลจะเก็บทั้งข้อมูลลายนิ้วมือ, วันที่-เวลาที่กราดตรวจ และรูปภาพใบหน้าของผู้ที่กราดตรวจลายนิ้วมือ และส่วนของแฟ้มลงบันทึกเข้าออก จะไม่มีการเก็บรูปภาพใบหน้าของผู้ที่กราดตรวจลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

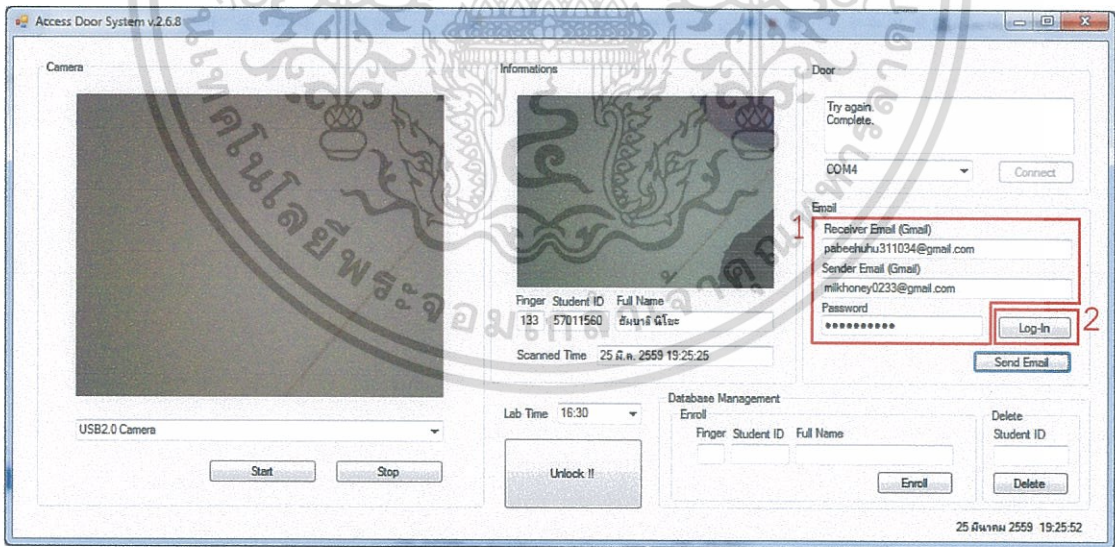


รูปที่ 4.9 ตัวอย่างภาพใบหน้าที่ถูกบันทึก

4.2 การทดลองส่งผลการบันทึกผ่านอีเมล

การทดลองนี้เป็นการทดลองการส่งผลการบันทึกข้อมูลการกราดตรวจลายนิ้วมือผ่านอีเมล เพื่อเป็นการทดสอบระบบว่าสามารถส่งอีเมลได้จริงๆ หรือไม่ ซึ่งมีขั้นตอนการทดลองดังนี้

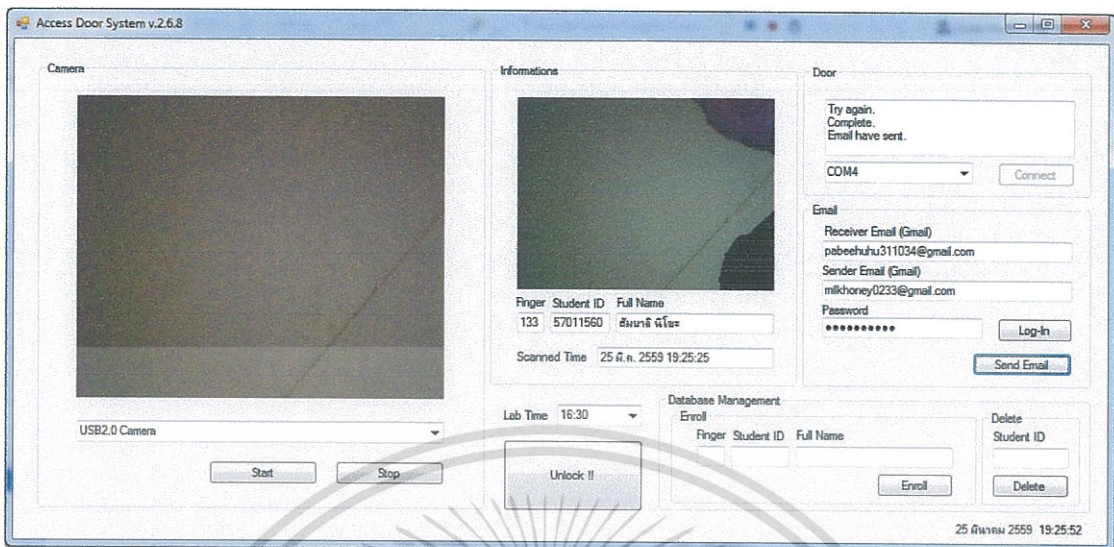
1. หลังจากที่ทำกรทดลอง 4.1 เสร็จสิ้นแล้วให้ทำการกรอกอีเมลของผู้รับ, อีเมลและรหัสผ่านของผู้ส่ง แล้วกดปุ่ม Log-In ดังรูปที่ 4.10



รูปที่ 4.10 การกรอกอีเมลเพื่อเตรียมการส่งอีเมล

2. กดส่งปุ่ม Send Email หากส่งสำเร็จจะขึ้นแจ้งว่า Email have sent. ดังรูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ส่งอีเมล

3. ตรวจสอบกล่องข้อความจะพบอีเมลที่ถูกส่งมาโดยมีแฟ้มข้อมูลแนบมาด้วย ดังรูปที่ 4.12



รูปที่ 4.12 ตรวจสอบผลการส่งอีเมล

4. รอเวลาจนถึง 20:00 น. ระบบจะทำการส่งอีเมลโดยอัตโนมัติ

5. ตรวจสอบกล่องข้อความอีกครั้งจะพบอีเมลที่ถูกส่งมาโดยมีแฟ้มข้อมูลแนบมาด้วย จากผลการทดลองข้างต้นสามารถสรุปได้ว่า ในการส่งอีเมลนั้น จะสำเร็จได้ก็ต่อเมื่อมีการใช้งานโปรแกรมในการปลดล็อกประตูไปแล้วอย่างน้อย 1 ครั้ง เนื่องจากหากยังไม่มีการใช้งานโปรแกรมมาก่อนจะไม่มีเรียกใช้แฟ้มข้อมูลสำหรับบันทึกข้อมูล ทำให้โปรแกรมไม่ทราบที่ต้องส่งแฟ้มข้อมูลไหนออกไป

นอกจากนั้นในการส่งอีเมลนั้น จำเป็นต้องใช้ Gmail ทั้งผู้รับและผู้ส่ง เนื่องจากผู้จัดทำได้เขียนโปรแกรมสำหรับส่งผ่าน Gmail ไว้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การวัดค่าความแม่นยำของ Fingerprint Sensor

การทดลองนี้เป็นการทดลองการวัดค่าความแม่นยำของ Fingerprint Sensor เพื่อประเมินความแม่นยำของ Fingerprint Sensor ในกรณีต่างๆโดยรวม ซึ่งมีขั้นตอนการทดลองดังนี้

1. ทำการรวบรวมผลการทดลองจากการทดลองที่ 4.1 โดยแบ่งนักศึกษากลุ่มทดลองออกเป็น 3 กลุ่ม (Section) ได้ดังตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองการกราดตรวจลายนิ้วมือ

	Section 1	Section 2	Section 3	รวม
จำนวนครั้งที่ทำการกราดตรวจทั้งหมด (ครั้ง)	102	107	85	294
จำนวนครั้งที่ไม่สามารถรับลายนิ้วมือได้ (ครั้ง)	14	11	19	44
จำนวนครั้งที่ระบุตัวบุคคลผิดพลาด (ครั้ง)	0	0	0	0

2. หาอัตราการล้มเหลวในการยึดเก็บสัญญาณ (FTC) และ อัตราการระบุ

$$\begin{aligned} \text{FTC} &= \frac{44}{294} \times 100\% \\ &= 14.97\% \end{aligned}$$

$$\begin{aligned} \text{อัตราการระบุ} &= \frac{(294 - 44) - 0}{294 - 44} \times 100\% \\ &= 100\% \end{aligned}$$

3. หาค่าความแม่นยำของ Fingerprint Sensor

$$\begin{aligned} \text{ค่าความแม่นยำ} &= 100 - 14.97 \\ &= 85.03\% \end{aligned}$$

จากผลการทดลองข้างต้นสามารถคำนวณหาอัตราการล้มเหลวในการยึดเก็บสัญญาณ (FTC) ได้เท่ากับ 14.97 เปอร์เซ็นต์ ซึ่งหาจากสมการที่ 3.1 จากกรณีที่ไม่มีกราดตรวจตัวตนผิดพลาดเลย ทำให้หาค่าอัตราการระบุได้เท่ากับ 100 เปอร์เซ็นต์ ซึ่งหาจากสมการที่ 3.2 และสามารถคำนวณหา ค่าความแม่นยำได้เท่ากับ 85.03 เปอร์เซ็นต์ อัตราการล้มเหลวในการยึดเก็บสัญญาณส่วนใหญ่มาจาก ความผิดพลาดของผู้ที่นิ้วลอก และเกิดจากความชื้นจากเหงื่อ ทำให้เมื่อนำนิ้วมือไปแตะกับ Fingerprint Sensor แล้วทำให้อ่านลายนิ้วมือได้ยากขึ้น

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ระบบควบคุมการเข้าออกประตูโดยใช้การตรวจสอบลายนิ้วมือและบันทึกภาพใบหน้า เป็นระบบที่พัฒนาขึ้นมาโดยมีจุดประสงค์หลัก คือ เพื่อควบคุมการเข้าออกประตูโดยการกราดตรวจลายนิ้วมือและบันทึกรายชื่อของผู้ที่ทำการกราดตรวจลายนิ้วมือ โดยเป็นการเน้นไปที่การตรวจสอบลายนิ้วมือก่อนที่จะทำการเข้าเรียนในวิชาปฏิบัติการของนักศึกษาชั้นปีที่สอง ภาควิชาวิศวกรรมโทรคมนาคม หากเป็นนักศึกษาที่มีการลงทะเบียนลายนิ้วมือไว้จะสามารถปลดล็อกประตูและถูกบันทึกข้อมูลการมาเข้าเรียนโดยมีการระบุวันที่และเวลาที่เข้าเรียนผ่านการกราดตรวจลายนิ้วมือ

5.2 ข้อเสนอแนะ

1. ในการที่จะนำไปใช้จำเป็นต้องมีเวลาให้ Fingerprint Sensor หยุดพักเป็นระยะ เนื่องจากหากเกิดความร้อนมากเกินไปจะทำให้อุปกรณ์มีประสิทธิภาพต่ำลง
2. ระบบนี้จำเป็นต้องใช้ไฟฟ้าตลอดเวลาเพื่อให้ระบบสามารถทำงานได้ หากเกิดเหตุการณ์ไฟฟ้าดับจะไม่สามารถปลดล็อกประตูจากภายนอกได้เลยนอกจากใช้กุญแจไข จึงควรมีอุปกรณ์สำรองไฟฟ้าไว้ป้องกันอุปกรณ์หยุดทำงานด้วย
3. หากติดตั้งอุปกรณ์นี้ไว้ในสภาพแวดล้อมที่มีแสงค่อนข้างน้อย ไฟ LED ที่ติดกับกล้องอาจให้แสงสว่างไม่เพียงพอ ทำให้มองใบหน้าจากโปรแกรมไม่ชัดเจนเท่าที่ควร
4. ในกรณีที่นักศึกษามีปัญหาเกี่ยวกับมือ เช่น นิ้วลอก มือมีเหงื่อมาก ลายนิ้วมือไม่ชัด จำเป็นต้องกราดตรวจลายนิ้วมือหลายครั้งจนกว่าจะสำเร็จ ดังนั้นหากใช้อุปกรณ์ที่มีประสิทธิภาพสูงกว่าเดิมอาจจะทำให้ลดจำนวนครั้งในการกราดตรวจลายนิ้วมือได้

เอกสารอ้างอิง

- [1] ชีระพล ลิ้มศรัทธา. เริ่มต้นเขียนโปรแกรม C# ด้วย Visual C# 2010 Express. พิมพ์ครั้งที่ 1. บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), 2554.
- [2] Tabby Near. “C# ติดต่อกับกล้องเว็บแคมด้วย AForge.NET.” <http://404noirdiary.blogspot.com/2013/04/blog-post.html>
- [3] Tabby Near. “Capture ภาพจาก Webcam แล้วบันทึกเป็นไฟล์ภาพ.” <http://404noirdiary.blogspot.com/2013/04/part-2-capturewebcam.html>
- [4] บริษัท PSP TECH จำกัด. “รีเลย์ (RELAY)”. <http://www.psptech.co.th/>
- [5] บริษัท ธนาบุตร จำกัด. “หลักการการทำงานของเครื่องสแกนลายนิ้วมือ.” <http://www.tanabutr.co.th/th/timerecorder/biometric/protocal>
- [6] บริษัท V.E.C.L.THAI จำกัด. “Fingerprint System”. <http://www.vecthai.com/>
- [7] เอกภพ สกฤตกิจกาญจน์. “ระบบเซ็นเซอร์เข้าและออกงานด้วยลายนิ้วมือผ่านทางเว็บไซต์.” สารนิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต, สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, มหาวิทยาลัยมหานคร, 2552.
- [8] Noah Coad. “Serial Port in C#”. <http://blogs.msmvps.com/coad/2005/03/23/>
- [9] Supachai Maiklang. “การเขียนภาษา C# เพื่ออ่านค่าที่ส่งมาจาก Serial Port.” <http://arttechnologyevolution.blogspot.com/2013/12/c-serial-port.html>
- [10] Sam Allen. “C# Textbox”. <http://www.dotnetperls.com/textbox>
- [11] อุกฤษฏ์ ศรีเสื่อขาม. “การประมวลผลลายพิมพ์นิ้วมือเบื้องต้นสำหรับระบบตรวจพิสูจน์ลายนิ้วมืออัตโนมัติ.” วิทยานิพนธ์ปริญญาวิทยาศาสตรมหาบัณฑิต, สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ, คณะวิทยาศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2541.
- [12] ณิชฎพล ตอพล, ณิชฎพล มุลศิริ และ ธนพล จันทรรักษ์มี. “การตรวจสอบรายชื่อด้วยเครื่องตรวจสอบลายนิ้วมือ.” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต, สาขาวิชาวิศวกรรมโทรคมนาคม, คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2552.
- [13] กวิน ตั้งวิทยานุกูล, คัมภีร์ ศุภจรกุล และ นัฏชนก ปุ่มพิมาย. “เครื่องสแกนลายนิ้วมือสำหรับวิชาปฏิบัติการ.” วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต, สาขาวิชาวิศวกรรมสารสนเทศ, คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2551.

- [14] อรรถวิท ศรีคำไทย และ นรภัทร ปัตตาเทสสัง. “ระบบตรวจสอบการเข้าออกสำนักงาน โดยใช้หลักการตรวจสอบลายนิ้วมือ.” ปรินญาณิพนธ์วิศวกรรมศาสตรบัณฑิต, สาขาวิชาวิศวกรรมโทรคมนาคม, คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2553.
- [15] วุฒิพงศ์ อารีกุล. การประมวลลายนิ้วมือนิติดิจิทัล. พิมพ์ครั้งที่ 1. บริษัท ยูโอเพน จำกัด, 2557.
- [16] สิทธิชัย ประสานวงศ์. แบบเรียนวิชา การใช้โปรแกรมฐานข้อมูล สร้างและจัดการฐานข้อมูลด้วย Microsoft Access 2010. พิมพ์ครั้งที่ 1. สำนักพิมพ์ซอฟต์แวร์, 2556.
- [17] ธนพล ฉันทจรวิชัย. ปฏิบัติการ Access 2010 ฉบับ Workshop. พิมพ์ครั้งที่ 1. บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน), 2554.
- [18] นันทินี แขวงโสภาก. คู่มือ Access2007 ฉบับสมบูรณ์. พิมพ์ครั้งที่ 1. บริษัท โปรวิชั่น จำกัด, 2551.
- [19] พร้อมเลิศ หล่อวิจิตร. คู่มือเรียน Visual Basic2010. พิมพ์ครั้งที่ 1. บริษัท โปรวิชั่น จำกัด, 2554.
- [20] ศุภชัย สมพานิช. พัฒนาระบบงานฐานข้อมูล ฉบับมืออาชีพ กับ Visual Studio 2008. พิมพ์ครั้งที่ 1. บริษัท ไอดีซี พรีเมียร์ จำกัด, 2553.
- [21] ศุภชัย สมพานิช. Professional Database Programming with VB 2010 & VC# 2010. พิมพ์ครั้งที่ 1. บริษัท ไอดีซี พรีเมียร์ จำกัด, 2553.
- [22] Daniel Hilgarth. “Create a .txt file if doesn’t exist.”. <http://stackoverflow.com/questions/9907682/create-a-txt-file-if-doesnt-exist-and-if-it-does-append-a-new-line>
- [23] Ranadheer Reddy. “ Sending email with attachment from c#” . <http://stackoverflow.com/questions/2825950/sending-email-with-attachments-from-c-attachments-arrive-as-part-1-2-in-thunde>
- [24] อัมรินทร์ เพ็ชรกุล. Microsoft Access 2013 ฉบับสมบูรณ์. พิมพ์ครั้งที่ 1. บริษัท รีไวว่า จำกัด, 2558.
- [25] ธีรวัฒน์ ประกอบผล. การเขียนแอปพลิเคชันด้วย Visual Basic 2010 ฉบับสมบูรณ์. พิมพ์ครั้งที่ 2. บริษัท รีไวว่า จำกัด, 2558.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งสำหรับ Arduino (Microcontroller)

```

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

int getFingerprintIDez();
SoftwareSerial mySerial(11,10);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
uint8_t getFingerprintEnroll(uint8_t id);

void setup()
{
  Serial.begin(9600);
  finger.begin(57600);
  pinMode(12,OUTPUT);
}

void loop()
{
  if (Serial.available())
  {
    char serip = Serial.read();
    if(serip=='a')
    {
      enrollScreen();
    }
    else if(serip=='b')
    {
      unlockDoor();
    }
  }
  else if (!Serial.available())
  {
    delay(50);
    getFingerprintIDez();
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void unlockDoor()
{
    digitalWrite(12,HIGH);
    delay(3000);
    digitalWrite(12,LOW);
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez()
{
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK)
    {
        Serial.print("N");
        return -1;
    }
    if(finger.fingerID < 10)
    {
        Serial.print("00");
        Serial.print(finger.fingerID);
    }
    else if(finger.fingerID >= 10 && finger.fingerID < 100)
    {
        Serial.print("0");
        Serial.print(finger.fingerID);
    }
    else if(finger.fingerID >= 100)
    {
        Serial.print(finger.fingerID);
    }
    return finger.fingerID;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void enrollScreen()
{
  Serial.print("E");
  uint8_t id = 0;
  while (true) {
    while (! Serial.available());
    char c = Serial.read();
    if (! isdigit(c)) break;
    id *= 10;
    id += c - '0';
  }
  getFingerprintEnroll(id);
}

uint8_t getFingerprintEnroll(uint8_t id) {
  uint8_t p = -1;
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
  }

  p = finger.image2Tz(1);
  switch (p) {
    case FINGERPRINT_OK:
      break;
    default:
      return p;
  }

  delay(2000);
  p = 0;
  while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
  }

  p = -1;
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
p = finger.image2Tz(2);
switch (p) {
  case FINGERPRINT_OK:
    break;
  default:
    return p;
}

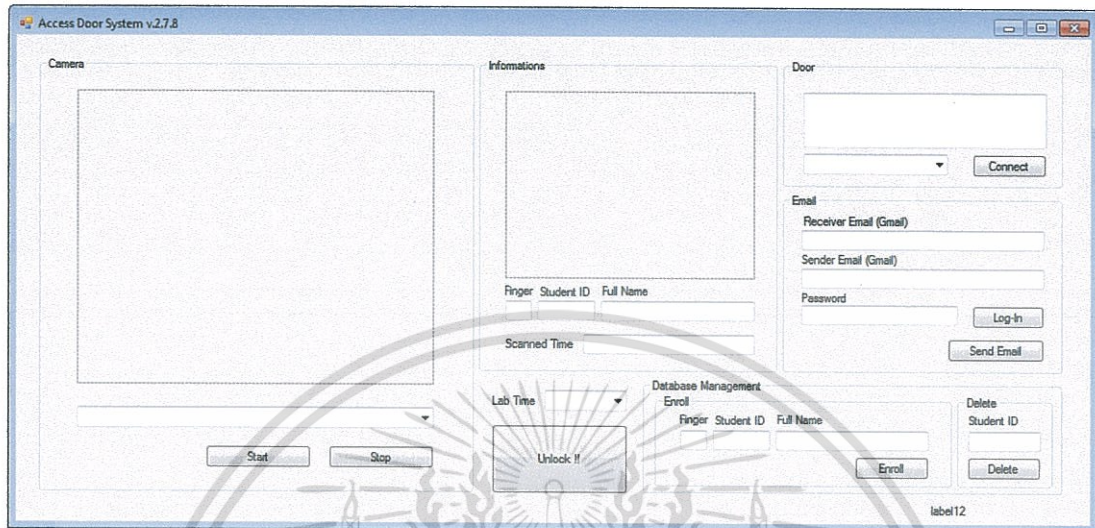
p = finger.createModel();
if (p != FINGERPRINT_OK) {
  return p;
}

p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
  Serial.print("C");
} else {
  return p;
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง C# สำหรับโปรแกรมที่ดำเนินงานบนคอมพิวเตอร์ ออกแบบ Interface ดังนี้



serialPort1 timer1 tooltip1

รูปที่ ก.1 รูปแบบการวาง Text Box และ ปุ่มต่างๆ ของโปรแกรม

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using AForge.Video;
using AForge.Video.DirectShow;
using System.IO;
using System.Drawing.Imaging;
using System.Data.OleDb;
using System.Net.Mail;
using System.Threading;

namespace AccessSystemv2._1._1
{
    public partial class Form1 : Form
    {
        FilterInfoCollection dvList; /*keep device list that connect to
computer*/
        VideoCaptureDevice cam; /*connect to selected device*/

        OleDbConnection Conn = new OleDbConnection();
        OleDbDataAdapter da;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DataSet ds = new DataSet();
OleDbCommandBuilder cbd;
BindingSource bs = new BindingSource();
DataRow dRow;
OleDbCommand sqlLog = null;
OleDbCommand sqlAdd = null;
OleDbCommand sqlDel = null;
OleDbParameter ParaIns;
String strConn = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\\Access System\\Database.accdb";
string lognamstr;
string SenderMail;
string SenderPass;
string ReceiverMail;
string adm = "ADMIN";
string sen = "a";
string senunl = "b";
string mtime;
string sdate;
string stime;
string actchk = null;
int txtcnt;
int stdcnt = 0;
int latecnt = 0;
string fidaddstr;
int i = 0;
string seripstr;
String[] seripmix = new string[2];
DateTime LateTime;
DateTime OutTime;

public Form1()
{
    InitializeComponent();
    CheckForIllegalCrossThreadCalls = false;
}

private void Form1_Load(object sender, EventArgs e)
{
    dvList = new
FilterInfoCollection(FilterCategory.VideoInputDevice); /*device name list*/
UnlockBut.Enabled = false;
EnrollBut.Enabled = false;
ConnectBut.Enabled = false;
CamStart.Enabled = false;
CamStop.Enabled = false;
timer1.Interval = 1000; /*set clock speed*/
timer1.Start();
foreach(FilterInfo info in dvList)
{
    CamcomboBox1.Items.Add(info.Name);
}
CamcomboBox1.SelectedIndex=0;
String[] strPortNames = SerialPort.GetPortNames(); /*keep all port
name that connect to computer*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

foreach(string n in strPortNames)
{
    DoorcomboBox.Items.Add(n);
}
DoorcomboBox.SelectedIndex = 0;
}

private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
    string serip;
    serip = serialPort1.ReadExisting();
    FingerIDBox.Text = serip;
    txtcnt = this.FingerIDBox.Text.Length;
    FingerIDBox.Clear();
    IDBox.Clear();
    NameBox.Clear();
    TimeScanBox.Clear();
    pictureBox2.Image = null;
    Console.WriteLine(txtcnt);

    if (serip == "N")
    {
        DoorText.AppendText("No fingerprint matched, please try
again.\n");
    }
    else if (serip == "E")
    {
        DoorText.AppendText("Enrolling...\n");
        String[] psen = new string[2];
        String enr;
        enr = EFingerBox.Text;
        psen[0] = enr;
        psen[1] = "a";
        string mix = string.Join(" ", psen);
        try
        {
            serialPort1.Write(mix);
        }
        catch (Exception)
        {
            DoorText.AppendText("Cannot connect to
microcontroller.\n");
        }
    }

    else if (serip == "C")
    {
        if (Conn.State == ConnectionState.Closed)
        {
            Conn.ConnectionString = strConn;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        try
        {
            Conn.Open();
        }
        catch (Exception)
        {
            DoorText.AppendText("Cannot connect to
database.\n");
        }
    }
    sqlAdd = new OleDbCommand("INSERT INTO
Personal(FingerID,StudentID,FullName) VALUES('" + fidaddstr + "', '" +
EStudentIDBox.Text + "', '" + ENameBox.Text + "')", Conn);
    try
    {
        sqlAdd.ExecuteNonQuery();
    }
    catch (Exception)
    {
        DoorText.AppendText("Cannot record log to
database.\n");
    }
    EStudentIDBox.Clear();
    ENameBox.Clear();
    EFingerBox.Clear();
    DoorText.AppendText("Enroll Completed.\n");
}
else if (serip != "E" || serip != "N" || serip != "C")
{
    if (txtcnt == 3)
    {
        seripstr = serip;
    }
    else if (txtcnt != 3)
    {
        seripmix[i] = serip;
        if (i == 1)
        {
            seripstr = string.Join("", seripmix);
        }
    }
    i++;
}

if (Conn.State == ConnectionState.Open)
{
    Conn.Close();
}
Conn.ConnectionString = strConn;
try
{
    Conn.Open();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        catch (Exception)
        {
            DoorText.AppendText("Cannot connect to database.\n");
        }
        String sqlPersonal = "SELECT * FROM Personal WHERE
FingerID='" + seripstr + "'";
        try
        {
            OleDbDataAdapter da = new
OleDbDataAdapter(sqlPersonal, Conn);
            da.Fill(ds, "Personal");
        }
        catch (Exception)
        {
            DoorText.AppendText("Cannot query data from
database.\n");
        }
        int count = 0;
        foreach (DataRow DR in ds.Tables["Personal"].Rows)
        {
            count++;
        }
        if (count >= 1)
        {
            string Capdt = DateTime.Now.ToString("dd-MM-yy_HH-mm-
ss");
            String[] CapFile = new string[9];
            CapFile[0] = "C:";
            CapFile[1] = @"\";
            CapFile[2] = "Access System";
            CapFile[3] = @"\";
            CapFile[4] = "Captured";
            CapFile[5] = @"\";
            CapFile[6] = Capdt;
            CapFile[7] = @". ";
            CapFile[8] = "bmp";
            string CapFilestr = string.Join("", CapFile);
            try
            {
                cam.Stop();
                pictureBox2.Image =
(Bitmap)pictureBox1.Image.Clone();
                pictureBox1.Image.Save(CapFilestr,
ImageFormat.Bmp);
                cam.Start();
            }
            catch (Exception)
            {
                DoorText.AppendText("Please open camera.\n");
            }
            dRow = ds.Tables[0].Rows[0];
            FingerIDBox.Text =
dRow.ItemArray.GetValue(0).ToString();
            IDBox.Text = dRow.ItemArray.GetValue(1).ToString();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NameBox.Text = dRow.ItemArray.GetValue(2).ToString();
TimeScanBox.Text = DateTime.Now.ToString();
DateTime dtt = DateTime.Now;
String fid = dRow.ItemArray.GetValue(0).ToString();
String stdid = dRow.ItemArray.GetValue(1).ToString();
String namm = dRow.ItemArray.GetValue(2).ToString();
Console.WriteLine(CapFilestr);
sqlLog = new OleDbCommand("INSERT INTO
LogTable(FingerID,StudentID,FullName,DayTime,Captured) VALUES('" + fid + "',
'" + stdid + "' ,'" + namm + "' ,'" + dtt + "' , '" + CapFilestr + "')",
Conn);

try
{
    sqlLog.ExecuteNonQuery();
}
catch (Exception)
{
    DoorText.AppendText("Cannot record log to
database.\n");
}
ds.Clear();
string Logdt = DateTime.Now.ToString("dd-MM-yy");
string getDateChk = DateTime.Now.ToString("HH:mm");
DateTime DateChk = DateTime.Parse(getDateChk);
String[] lognam = new string[8];
lognam[0] = "C:";
lognam[1] = @"\\";
lognam[2] = "Access System";
lognam[3] = @"\\";
lognam[4] = "Log";
lognam[5] = @"\\";
lognam[6] = Logdt;
lognam[7] = ".txt";
lognamstr = string.Join("", lognam);
if (!File.Exists(lognamstr))
{
    try
    {
        File.Create(lognamstr).Dispose();
    }
    catch (Exception)
    {
        DoorText.AppendText("Cannot create log
file.\n");
    }
    try
    {
        StreamWriter tw = new StreamWriter(lognamstr,
true);

        tw.Write(FingerIDBox.Text);
        tw.Write("\t");
        tw.Write(IDBox.Text);
        tw.Write("\t");
        tw.Write(NameBox.Text);
        tw.Write("\t\t");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        stdcnt++;
        latecnt++;
    }

    else if (DateChk < LateTime)
    {
        tw.WriteLine("-");
        stdcnt++;
    }
}
tw.Close();
}
catch (Exception)
{
    DoorText.AppendText("Cannot record to
Log.\n");
}
}
try
{
    serialPort1.Write("b");
}
catch (Exception)
{
    DoorText.AppendText("Cannot connect to
microcontroller.\n");
}
DoorText.AppendText("Complete.\n");
}
if(i > 1)
{
    try
    {
        i = 0;
        dRow.Delete();
        count = 0;
    }
    catch(Exception)
    {
    }
}
seripstr = null;
}
}

private void ConnectBut_Click(object sender, EventArgs e)
{
    serialPort1.PortName = DoorcomboBox.SelectedItem + "";
    try
    {
        serialPort1.Open();
    }
    catch(Exception)
    {
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        DoorText.AppendText("Cannot connect to microcontroller.\n");
    }
    if(serialPort1.IsOpen)
    {
        EnrollBut.Enabled = true;
        ConnectBut.Enabled = false;
        CamStart.Enabled = true;
        CamStop.Enabled = true;
    }
}

private void EnrollBut_Click(object sender, EventArgs e)
{
    int enrchk = this.EFingerBox.TextLength;
    String[] fidadd = new string[2];
    if(enrchk == 1)
    {
        fidadd[0] = "00";
        fidadd[1] = EFingerBox.Text;
        fidaddstr = string.Join(",", fidadd);
    }
    else if(enrchk == 2)
    {
        fidadd[0] = "0";
        fidadd[1] = EFingerBox.Text;
        fidaddstr = string.Join(",", fidadd);
    }
    else if(enrchk == 3)
    {
        fidaddstr = EFingerBox.Text;
    }

    if(Conn.State == ConnectionState.Open)
    {
        Conn.Close();
    }
    Conn.ConnectionString = strConn;
    try
    {
        Conn.Open();
    }
    catch (Exception)
    {
        DoorText.AppendText("Cannot connect to database.\n");
    }
    String sqlPersonal = "SELECT * FROM Personal WHERE FingerID='" +
fidaddstr + "'";
    try
    {
        OleDbDataAdapter da = new OleDbDataAdapter(sqlPersonal, Conn);
        da.Fill(ds, "Personal");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

catch(Exception)
{
    DoorText.AppendText("Cannot query data from database.\n");
}

int count = 0;
foreach(DataRow DR in ds.Tables["Personal"].Rows)
{
    count++;
}
if(count >= 1)
{
    MessageBox.Show("Duplicate data, please try another
fingerprint ID.");
    ds.Clear();
}
else if (count == 0)
{
    try
    {
        serialPort1.Write(sen);
    }
    catch(Exception)
    {
        DoorText.AppendText("Cannot connect to
microcontroller.\n");
    }
    ds.Clear();
}
}

private void CamStart_Click(object sender, EventArgs e)
{
    try
    {
        cam = new
VideoCaptureDevice(dvList[CamcomboBox1.SelectedIndex].MonikerString);
        cam.NewFrame += new NewFrameEventHandler(cam_NewFrame);
        cam.Start();
    }
    catch(Exception)
    {
        DoorText.AppendText("Cannot open camera.\n");
    }

    if(cam.IsRunning && serialPort1.IsOpen)
    {
        UnlockBut.Enabled = true;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void cam_NewFrame(object sender, NewFrameEventArgs eventArgs)
{
    try
    {
        Bitmap camget = (Bitmap)eventArgs.Frame.Clone();
        pictureBox1.Image = camget;
    }
    catch(Exception)
    {
        DoorText.AppendText("Please open camera.\n");
    }
}

private void CamStop_Click(object sender, EventArgs e)
{
    try
    {
        if (cam.IsRunning)
        {
            cam.Stop();
            UnlockBut.Enabled = false;
        }
    }
    catch(Exception)
    {
        DoorText.AppendText("Camera is now closed.\n");
    }
}

private void UnlockBut_Click(object sender, EventArgs e)
{
    string Capdt = DateTime.Now.ToString("dd-MM-yy_HH-mm-ss");
    String[] CapFile = new string[8];
    CapFile[0] = "C:";
    CapFile[1] = @"\";
    CapFile[2] = "Access System";
    CapFile[3] = @"\";
    CapFile[4] = "Captured";
    CapFile[5] = @"\";
    CapFile[6] = Capdt;
    CapFile[7] = ".bmp";
    string CapFilestr = string.Join("", CapFile);
    try
    {
        cam.Stop();
        pictureBox2.Image = (Bitmap)pictureBox1.Image.Clone();
        pictureBox1.Image.Save(CapFilestr, ImageFormat.Bmp);
        cam.Start();
    }
    catch(Exception)
    {
        DoorText.AppendText("Please open camera.\n");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

FingerIDBox.Text = "-";
IDBox.Text = "-";
NameBox.Text = "Unlocked by Admin.";
TimeScanBox.Text = DateTime.Now.ToString();

try
{
    serialPort1.Write(senun1);
}
catch(Exception)
{
    DoorText.AppendText("Cannot connect to microcontroller.\n");
}

DateTime dtun1 = DateTime.Now;
if (Conn.State == ConnectionState.Closed)
{
    Conn.ConnectionString = strConn;
    try
    {
        Conn.Open();
    }
    catch (Exception)
    {
        DoorText.AppendText("Cannot connect to database.\n");
    }
}
sqlLog = new OleDbCommand("INSERT INTO
LogTable(FingerID,StudentID,FullName,DayTime,Captured) VALUES('" + adm + "',
'" + adm + "', '" + adm + "', '" + dtun1 + "', '" + CapFilestr + "')", Conn);
try
{
    sqlLog.ExecuteNonQuery();
}
catch(Exception)
{
    DoorText.AppendText("Cannot record log to database.\n");
}

string Logdt = DateTime.Now.ToString("dd-MM-yy");
String[] lognam = new string[8];
lognam[0] = "C:";
lognam[1] = @"\";
lognam[2] = "Access System";
lognam[3] = @"\";
lognam[4] = "Log";
lognam[5] = @"\";
lognam[6] = Logdt;
lognam[7] = ".txt";
lognamstr = string.Join("", lognam);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (!File.Exists(lognamstr))
{
    try
    {
        File.Create(lognamstr).Dispose();
    }
    catch(Exception)
    {
        DoorText.AppendText("Cannot create log file.\n");
    }

    try
    {
        TextWriter tw = new StreamWriter(lognamstr, true);
        tw.Write("ADMIN");
        tw.Write("\t");
        tw.Write("ADMIN");
        tw.Write("\t");
        tw.Write("ADMIN");
        tw.Write("\t\t");
        tw.WriteLine(DateTime.Now.ToString());
        tw.Close();
    }
    catch(Exception)
    {
        DoorText.AppendText("Cannot record to Log.\n");
    }
}
else if (File.Exists(lognamstr))
{
    try
    {
        TextWriter tw = new StreamWriter(lognamstr, true);
        tw.Write("ADMIN");
        tw.Write("\t");
        tw.Write("ADMIN");
        tw.Write("\t\t");
        tw.Write("ADMIN");
        tw.Write("\t\t\t");
        tw.WriteLine(DateTime.Now.ToString());
        tw.Close();
    }
    catch(Exception)
    {
        DoorText.AppendText("Cannot record to Log.\n");
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void CamcomboBox1_SelectedIndexChanged(object sender,
EventArgs e)
{
}

private void pictureBox1_Click(object sender, EventArgs e)
{
}

private void DoorcomboBox_SelectedIndexChanged(object sender,
EventArgs e)
{
    if(DoorcomboBox.SelectedItem != "")
    {
        ConnectBut.Enabled = true;
    }
}

private void DoorText_TextChanged(object sender, EventArgs e)
{
}

private void pictureBox2_Click(object sender, EventArgs e)
{
}

private void label1_Click_1(object sender, EventArgs e)
{
}

private void DeleteButt_Click(object sender, EventArgs e)
{
    DialogResult cmf = MessageBox.Show("Are you sure to delete ?",
"Confirmation", MessageBoxButtons.YesNo);
    if(cmf == DialogResult.Yes)
    {
        if (Conn.State == ConnectionState.Closed)
        {
            Conn.ConnectionString = strConn;
            try
            {
                Conn.Open();
            }
            catch(Exception)
            {
                DoorText.AppendText("Cannot connect to database.\n");
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า. ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        sqlDel = new OleDbCommand("DELETE FROM Personal WHERE
StudentID = '" + DStudentIDBox.Text + "'", Conn);
        try
        {
            sqlDel.ExecuteNonQuery();
        }
        catch(Exception)
        {
            DoorText.AppendText("Cannot delete data.\n");
        }
        DStudentIDBox.Clear();
    }
    else if(cmf == DialogResult.No)
    {
    }
}

private void LoginButt_Click(object sender, EventArgs e)
{
    SenderMail = SenderMailBox.Text;
    SenderPass = SenderPassBox.Text;
    ReceiverMail = ReceiverBox.Text;
}

private void SendMailButt_Click(object sender, EventArgs e)
{
    string stdcntstr = stdcnt.ToString();
    string latecntstr = latecnt.ToString();
    string Logdt = DateTime.Now.ToString("dd-MM-yy");
    String[] lognam = new string[8];
    lognam[0] = "C:";
    lognam[1] = @"\\";
    lognam[2] = "Access System";
    lognam[3] = @"\\";
    lognam[4] = "Log";
    lognam[5] = @"\\";
    lognam[6] = Logdt;
    lognam[7] = ".txt";
    lognamstr = string.Join("", lognam);
    if (!File.Exists(lognamstr))
    {
        try
        {
            File.Create(lognamstr).Dispose();
        }
        catch (Exception)
        {
            DoorText.AppendText("Cannot create log file.\n");
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

try
{
    StreamWriter tw = new StreamWriter(lognamstr, true);
    tw.WriteLine("-----");

    tw.Write("Total (In) : ");
    tw.WriteLine(stdcntstr);
    tw.Write("Late : ");
    tw.WriteLine(latecntstr);
    tw.Close();
}
catch (Exception)
{
    DoorText.AppendText("Cannot record to Log.\n");
}
}
else if (File.Exists(lognamstr))
{
    try
    {
        StreamWriter tw = new StreamWriter(lognamstr, true);
        tw.WriteLine("-----");

        tw.Write("Total (In) : ");
        tw.WriteLine(stdcntstr);
        tw.Write("Late : ");
        tw.WriteLine(latecntstr);
        tw.Close();
    }
    catch (Exception)
    {
        DoorText.AppendText("Cannot record to Log.\n");
    }
}
try
{
    MailMessage mail = new MailMessage();
    mail.To.Add(ReceiverMail);
    mail.From = new MailAddress(SenderMail);
    mail.Subject = "Laboratory Student Report";
    mail.Body = "Student Report";
    mail.IsBodyHtml = true;
    mail.Attachments.Add(new Attachment(lognamstr));
    SmtplibClient smtp = new SmtplibClient();
    smtp.Host = "smtp.gmail.com";
    smtp.Port = 587;
    smtp.Credentials = new
System.Net.NetworkCredential(SenderMail, SenderPass);
    smtp.EnableSsl = true;
    smtp.Send(mail);
    DoorText.AppendText("Email have sent.\n");
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

catch(Exception)
{
    DoorText.AppendText("Cannot send Email.\n");
}
}

private void timer1_Tick(object sender, EventArgs e)
{
    sdate = DateTime.Now.ToLongDateString();
    stime = DateTime.Now.ToLongTimeString();
    label12.Text = sdate + " " + stime;
    mtime = DateTime.Now.ToString("HH");
    if(mtime == "20")
    {
        if(actchk == null)
        {
            actchk = "1";
            try
            {
                string stdcntstr = stdcnt.ToString();
                string latecntstr = latecnt.ToString();
                string Logdt = DateTime.Now.ToString("dd-MM-yy");
                String[] lognam = new string[8];
                lognam[0] = "C:";
                lognam[1] = @"\\";
                lognam[2] = "Access System";
                lognam[3] = @"\\";
                lognam[4] = "Log";
                lognam[5] = @"\\";
                lognam[6] = Logdt;
                lognam[7] = ".txt";
                lognamstr = string.Join("", lognam);
                if (!File.Exists(lognamstr))
                {
                    try
                    {
                        File.Create(lognamstr).Dispose();
                    }
                    catch (Exception)
                    {
                        DoorText.AppendText("Cannot create log
file.\n");
                    }
                }
                try
                {
                    TextWriter tw = new StreamWriter(lognamstr,
true);
                    tw.WriteLine("-----");
                    tw.WriteLine("Total (In) : ");
                    tw.WriteLine(stdcntstr);
                    tw.WriteLine("Late : ");
                    tw.WriteLine(latecntstr);
                    tw.Close();
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        catch (Exception)
        {
            DoorText.AppendText("Cannot record to
Log.\n");
        }
    }
    else if (File.Exists(lognamstr))
    {
        try
        {
            TextWriter tw = new StreamWriter(lognamstr,
true);
            tw.WriteLine("-----");
            tw.WriteLine("-----");
            tw.WriteLine("-----");
            tw.WriteLine("Total (In) : ");
            tw.WriteLine(stdcntstr);
            tw.WriteLine("Late : ");
            tw.WriteLine(latecntstr);
            tw.Close();
        }
        catch (Exception)
        {
            DoorText.AppendText("Cannot record to
Log.\n");
        }
    }

    MailMessage mail = new MailMessage();
    mail.To.Add(ReceiverMail);
    mail.From = new MailAddress(SenderMail);
    mail.Subject = "Laboratory Student Report";
    mail.Body = "Student Report";
    mail.IsBodyHtml = true;
    mail.Attachments.Add(new Attachment(lognamstr));
    SmtplibClient smtp = new SmtplibClient();
    smtp.Host = "smtp.gmail.com";
    smtp.Port = 587;
    smtp.Credentials = new
System.Net.NetworkCredential(SenderMail, SenderPass);
    smtp.EnableSsl = true;
    smtp.Send(mail);
    DoorText.AppendText("Email have sent.\n");
}
catch(Exception)
{
    DoorText.AppendText("Cannot send Email.\n");
}
}
}
else if(mtime != "20")
{
    actchk = null;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void LabTimecomboBox_SelectedIndexChanged(object sender,
EventArgs e)
{
    string labselect = LabTimecomboBox.Text;
    if (labselect == "09:00")
    {
        LateTime = DateTime.Parse("09:15");
        OutTime = DateTime.Parse("10:30");
    }
    else if (labselect == "13:00")
    {
        LateTime = DateTime.Parse("13:15");
        OutTime = DateTime.Parse("14:30");
    }
    else if (labselect == "16:30")
    {
        LateTime = DateTime.Parse("16:45");
        OutTime = DateTime.Parse("18:00");
    }
}
}
}
}

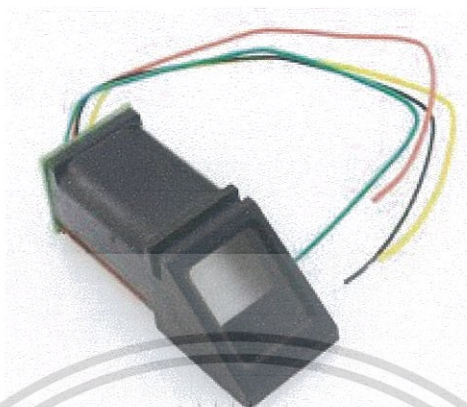
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ADAFRUIT FINGERPRINT SENSOR

Specifications

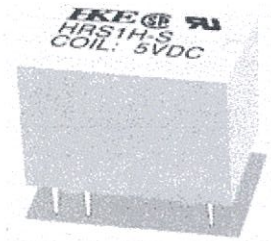
- **Supply voltage:** 3.6 - 6.0VDC
- **Operating current:** 120mA max
- **Peak current:** 150mA max
- **Fingerprint imaging time:** <1.0 seconds
- **Window area:** 14mm x 18mm
- **Signature file:** 256 bytes
- **Template file:** 512 bytes
- **Storage capacity:** 162 templates
- **Safety ratings (1-5 low to high safety)**
- **False Acceptance Rate:** <0.001% (Security level 3)
- **False Reject Rate:** <1.0% (Security level 3)
- **Interface:** UART
- **Baud rate:** 9600, 19200, 28800, 38400, 57600 (default is 57600)
- **Working temperature rating:** -20C to +50C
- **Working humidity:** 40%-85% RH
- **Full Dimensions:** 56 x 20 x 21.5mm
- **Exposed Dimensions (when placed in box):** 21mm x 21mm x 21mm triangular
- **Weight:** 20 grams

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HRS1(H) Relay

1 COIL DATA

1.1 Nominal Voltage.	3 VAC t 24 VAC
1.2 Coil Resistance	refer to Table 1
1.3 Operate Voltage	refer to Table 1
1.4 Release Voltage	refer to Table 1
1.5 Nominal Power Consumption	200 to 360 mW



HRS1(H) Relay

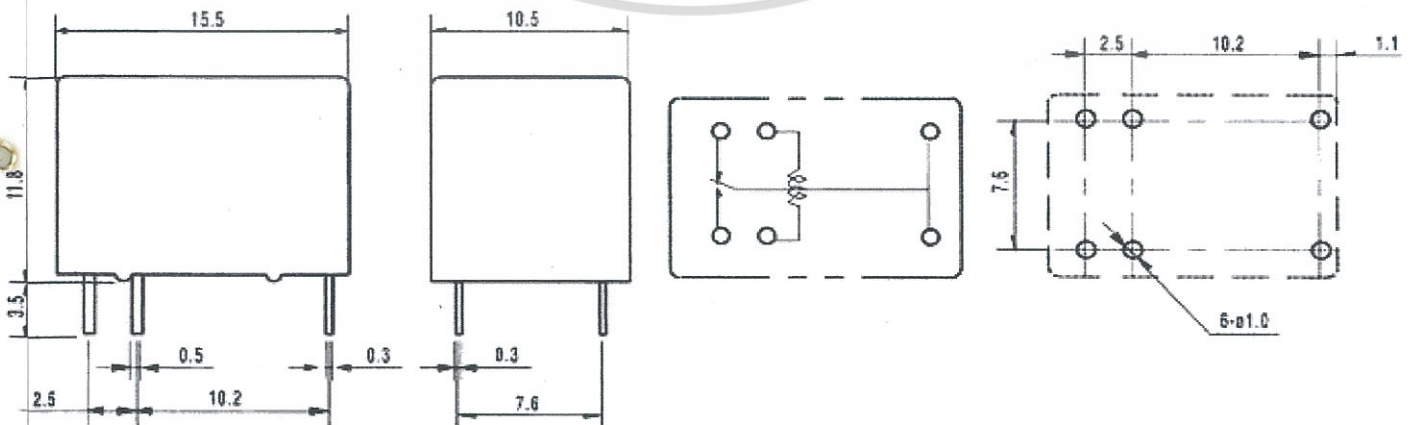
2 CONTACT DATA

2.1 Contact Arrangement	1 Form C
2.2 Contact Material	AuAg
2.3 Contact Rating	1A 24VDC/120VAC
2.4 Max-Switching Voltage	30 VAC/120VAC
2.5 Max-Switching Current	2A
2.6 Max.switching Power	120 VA,24W
2.7 Contact Resistance (Initial)	100 mΩ at 6 VAC 1A
2.8 Life Expectancy Electrical	100,000 operation at nominal load
Mechanical	10,000,000 operations

3 GENERAL DATA

3.1 Insulation Resistance	Min. 100mΩ at 500 VDC
3.2 Dielectric Strength	1000 VAC,1 min between open contact 1,500 VAC,1 min between contact and coil.
3.3 Operate time	Max. 5 mS
3.4 Release Time	Max. 5 mS
3.5 Temperature Range	-25 to +55
3.6 Shock Resistance	10 G
3.7 Vibration Resistance	10 - 55 Hz,Amplitude 1.5mm

4.DIMENSION (in mm)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 ORDERING CODE

HRS1

H-

S

DC5V

102

NOMINAL VOLTAGE DC3V,DC5V,DC6V,DC12,DC24V

ENCLOSURE
S - - WASHABLE

COIL
NIL --- STANDARD TYPE
H --- HIGH SENSITIVE TYPE

TYPE

6 COIL DATA CHART

ORDERING CODE	COIL NOMINAL VDC	COIL RESISTANCE +/-10%	OPERATE VOLTAGE VDC	RELEASE VOLTAGE VDC	COIL NOMINAL mW
HRS1-S DC3V	3	25	2.25	0.30	360
HRS1-S DC5V	5	70	3.75	0.50	
HRS1-S DC6V	6	100	4.50	0.60	
HRS1-S DC9V	9	220	6.75	0.90	
HRS1-S DC12V	12	400	9.00	1.20	
HRS1-S DC24V	24	1600	18.00	2.40	
HRS1H-S DC3V	3	45	2.25	0.30	200
HRS1H-S DC5V	5	120	3.75	0.50	
HRS1H-S DC6V	6	180	4.50	0.60	
HRS1H-S DC9V	9	400	6.75	0.90	
HRS1H-S DC12V	12	700	9.00	1.20	
HRS1H-S DC24V	24	2800	18.00	2.40	

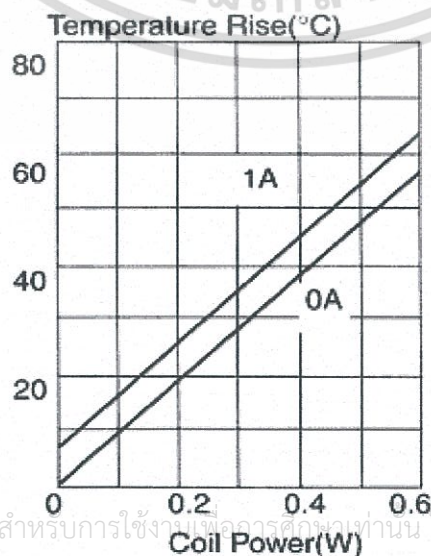
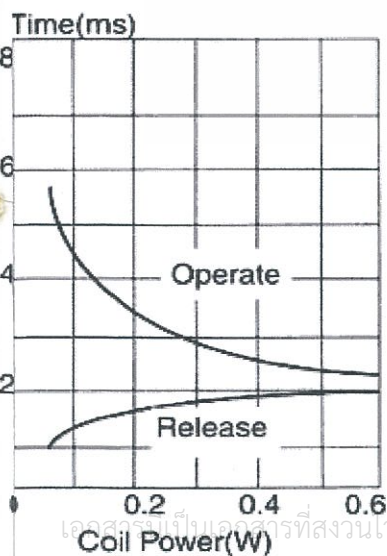
Table 1

7 HRS1(H) CHARACTERISTIC DATA

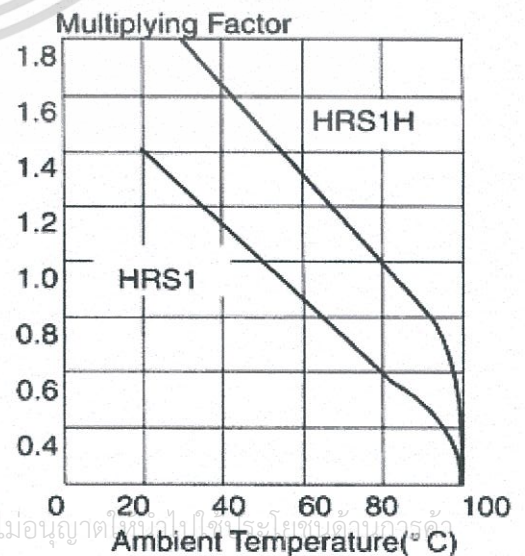
Timing

Coil Temperature

Operating Range



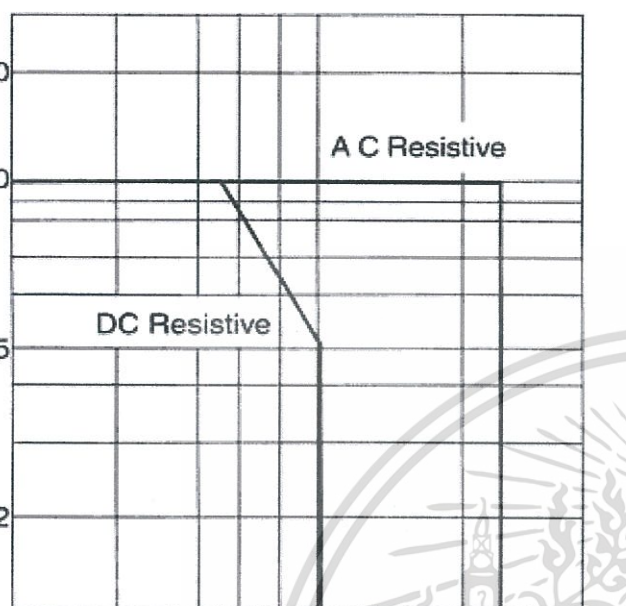
Nominal Voltage



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานของเครื่องวัดของคุณเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ต่อบุคคล
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

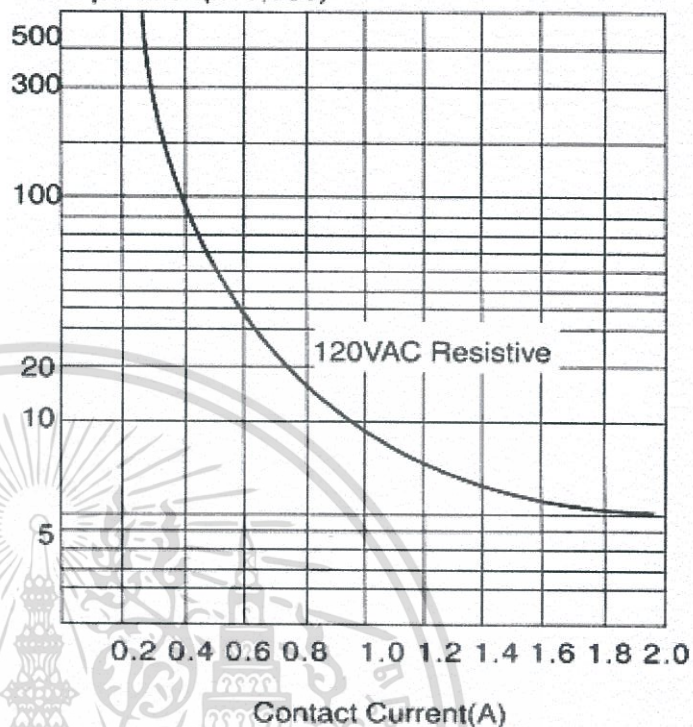
Maximum Switching Power

Contact Current(A)



Life Curve

Operation(x10,000)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

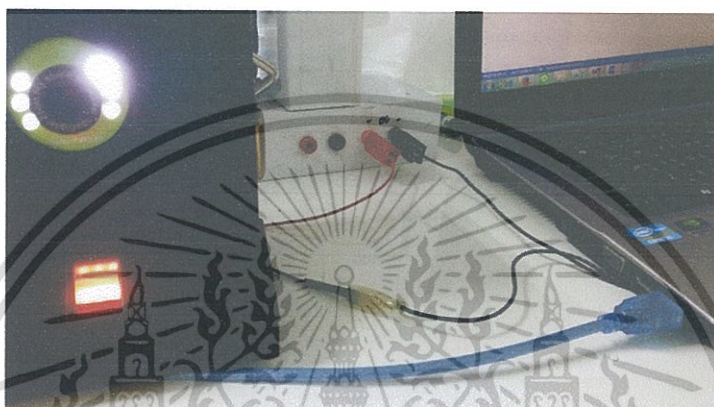


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานระบบ Access Door System

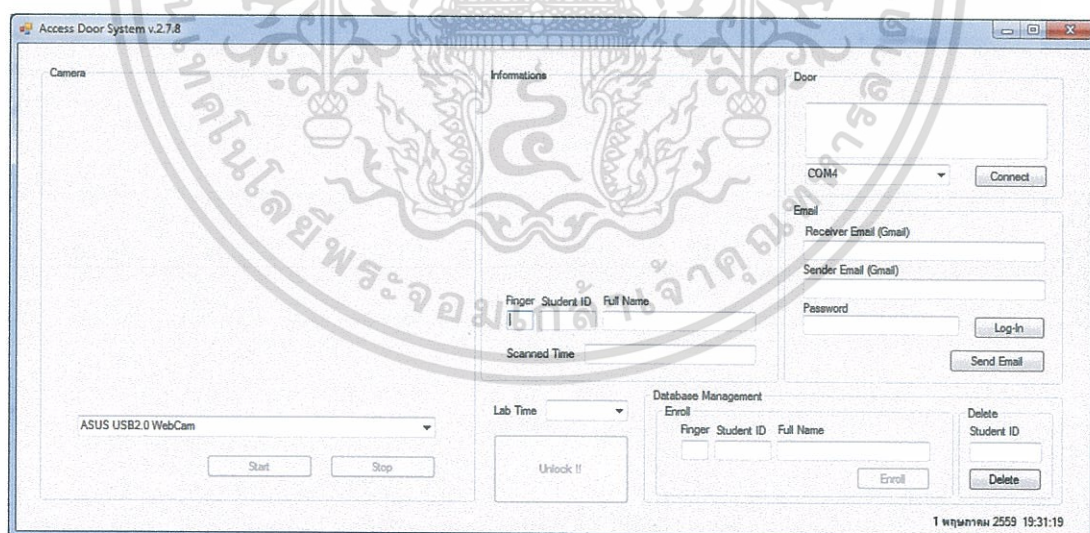
เริ่มต้นใช้งานระบบ

1. ทำการติดตั้งอุปกรณ์ โดยเชื่อมต่อสาย USB ของอุปกรณ์เข้ากับคอมพิวเตอร์, เชื่อมต่อสายสีแดงจากอุปกรณ์เข้ากับช่องเสียบสีแดง +DC ของประตู, เชื่อมต่อสายสีดำจากหม้อแปลงไฟฟ้า 12 โวลต์ เข้ากับช่องเสียบสีดำ -DC ของประตู และเชื่อมต่อสายทางปลาของอุปกรณ์เข้ากับสายทางปลาของหม้อแปลงไฟฟ้า 12 โวลต์ ดังรูปที่ ค.1



รูปที่ ค.1 การเชื่อมต่ออุปกรณ์

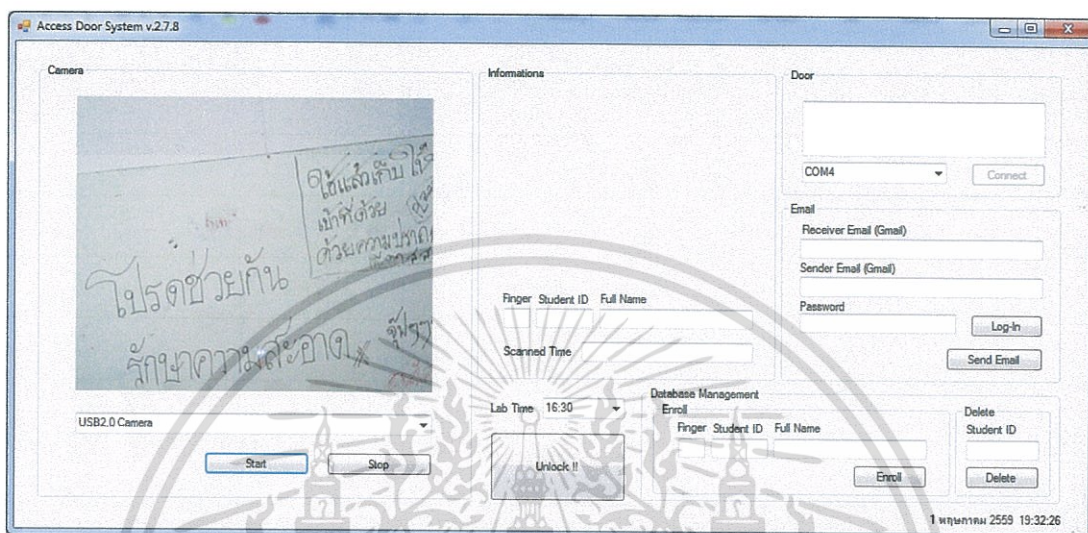
2. เปิดโปรแกรม Access Door System v2.7.8 จะปรากฏหน้าต่างดังรูปที่ ค.2



รูปที่ ค.2 หน้าต่างโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

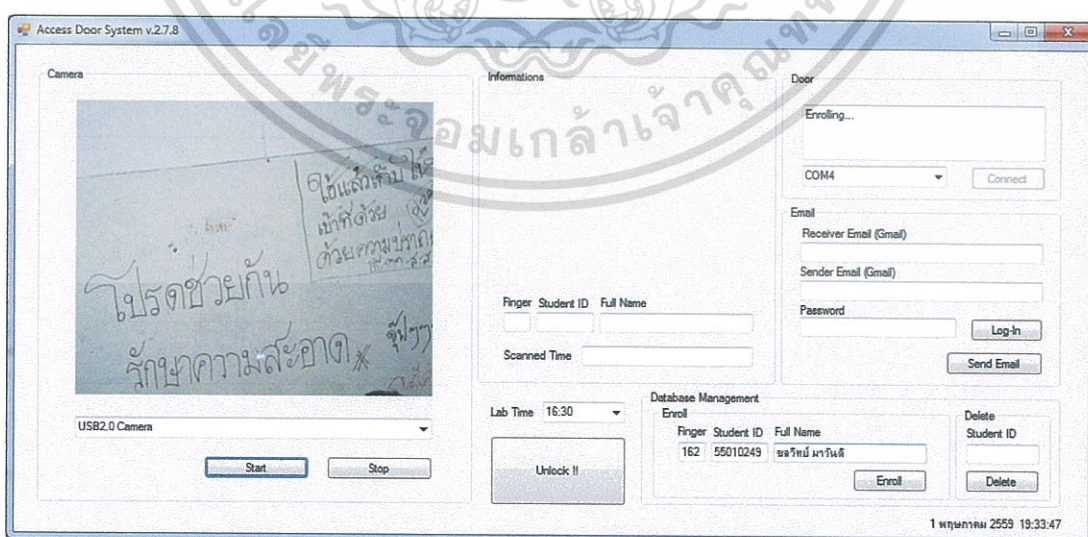
- เลือกเวลาปฏิบัติการแล้วกดปุ่ม Connect จากนั้นเลือกกล้องที่ต้องการใช้งานในส่วน Camera แล้วกด Start ปุ่ม Unlock จะสามารถใช้งานได้ และระบบจะพร้อมใช้งาน ดังรูปที่ ค.3



รูปที่ ค.3 หน้าต่างโปรแกรมเมื่อเชื่อมต่อกับพอร์ตอนุกรมและเปิดกล้อง

การลงทะเบียนลายนิ้วมือ

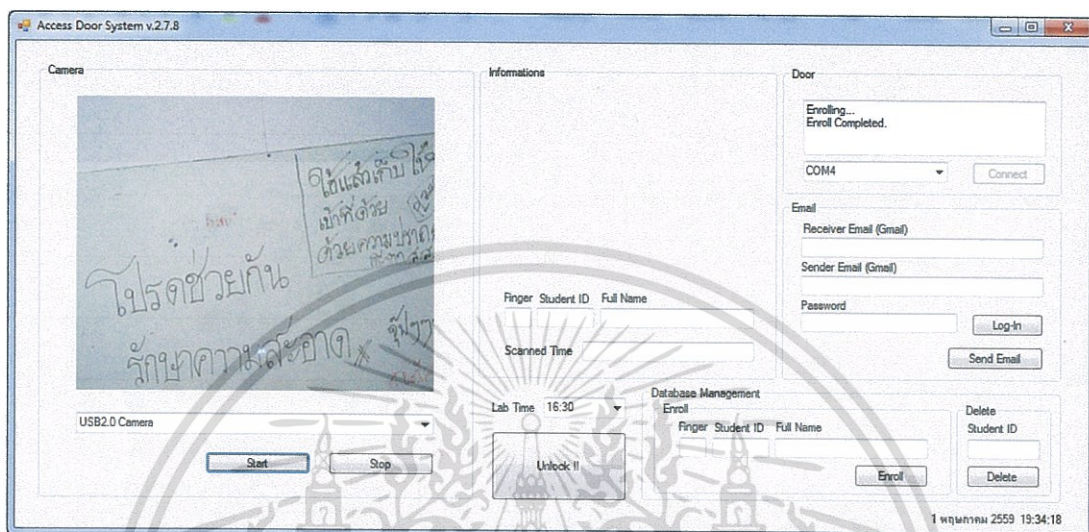
- ทำการกรอกรายละเอียดในส่วน Database Management ส่วนย่อย Enroll โดยกรอกรหัสลายนิ้วมือลงในช่อง Finger, กรอกรหัสนักศึกษาในช่อง Student ID, กรอกชื่อ-นามสกุลในช่อง Full Name แล้วกดปุ่ม Enroll ระบบจะเริ่มดำเนินการลงทะเบียนลายนิ้วมือและตอบสนองว่า Enrolling... ดังรูปที่ ค.4



รูปที่ ค.4 หน้าต่างโปรแกรมเมื่อเริ่มลงทะเบียนลายนิ้วมือ

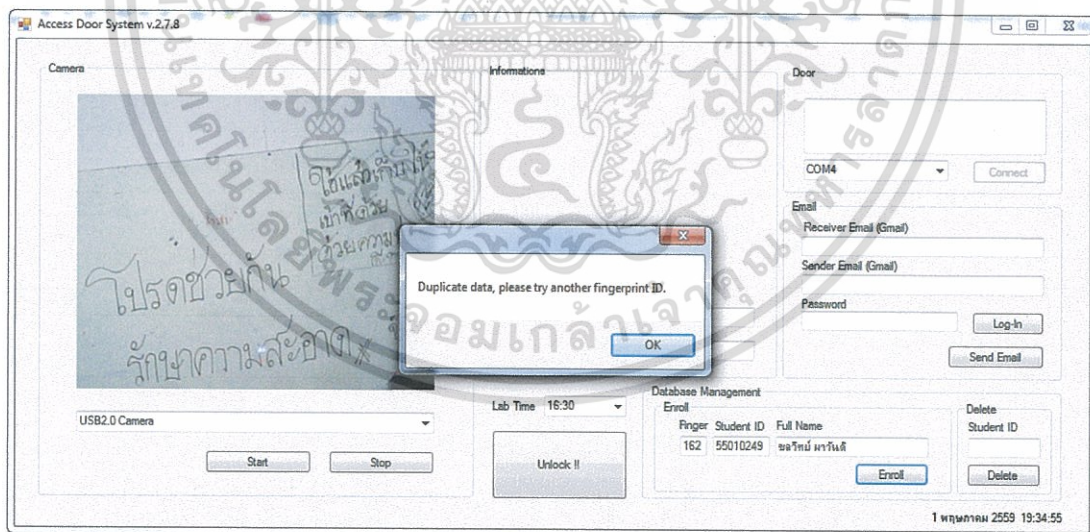
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการกราดตรวจลายนิ้วมือที่ Fingerprint Sensor 2 ครั้ง เมื่อระบบลงทะเบียนเสร็จแล้วจะตอบสนองดังรูปที่ ค.5



รูปที่ ค.5 หน้าต่างโปรแกรมเมื่อลงทะเบียนลายนิ้วมือเสร็จสิ้น

3. หากลายนิ้วมือนั้นมีข้อมูลอยู่แล้วระบบจะตอบสนองดังรูปที่ ค.6

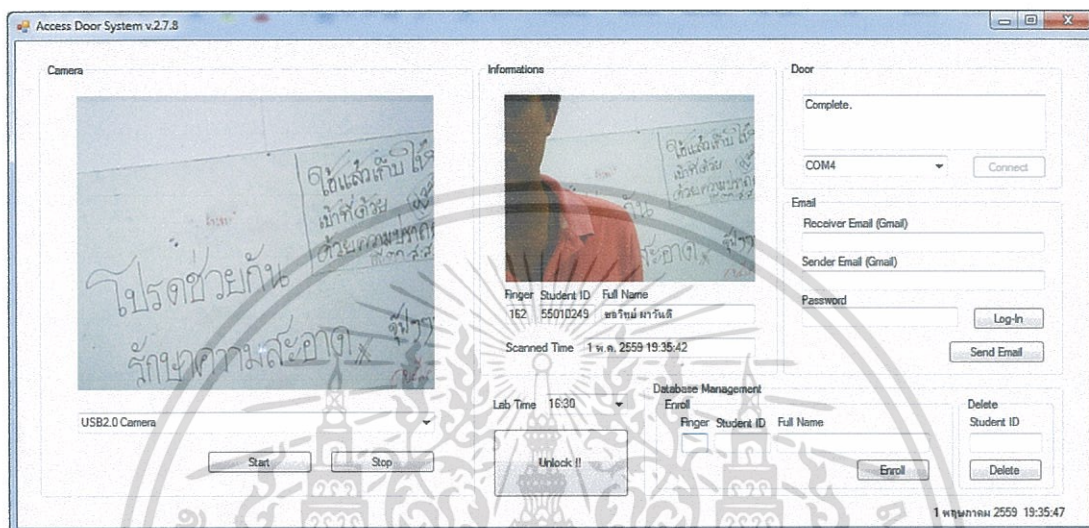


รูปที่ ค.6 หน้าต่างโปรแกรมหากลงทะเบียนลายนิ้วมือซ้ำ

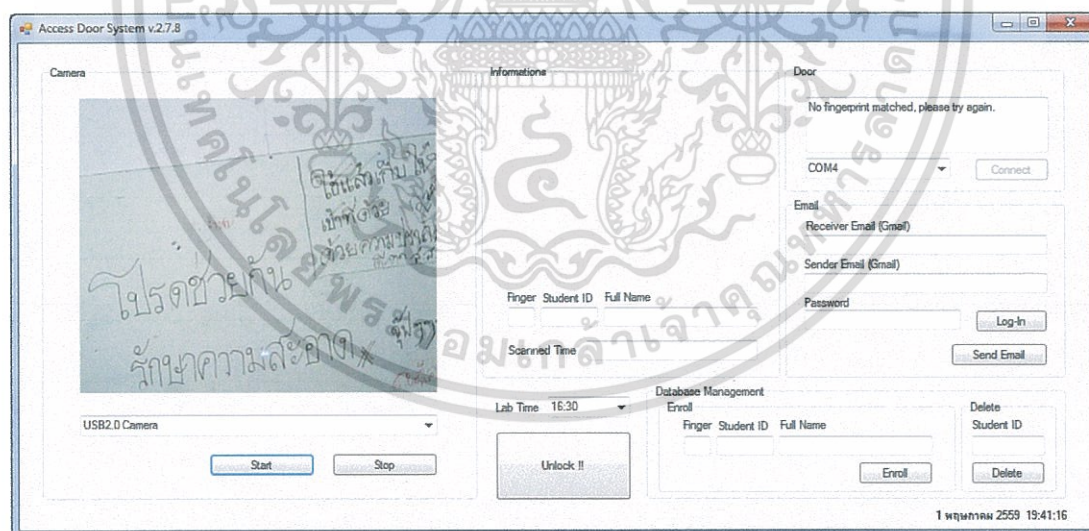
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปลดล็อกโดยใช้การกราดตรวจลายนิ้วมือ

1. วางนิ้วมือที่ลงทะเบียนแล้วไปที่ Fingerprint Sensor ระบบจะทำการตรวจสอบลายนิ้วมือ หากพบข้อมูลประตูจะถูกปลดล็อกและแสดงรายละเอียดดังรูปที่ ค.7 หากไม่พบลายนิ้วมือ ระบบจะแจ้งผู้ใช้งานว่า “No fingerprint matched, please try again.” ดังรูปที่ ค.8



รูปที่ ค.7 หน้าต่างโปรแกรมเมื่อทำการกราดตรวจพบลายนิ้วมือ



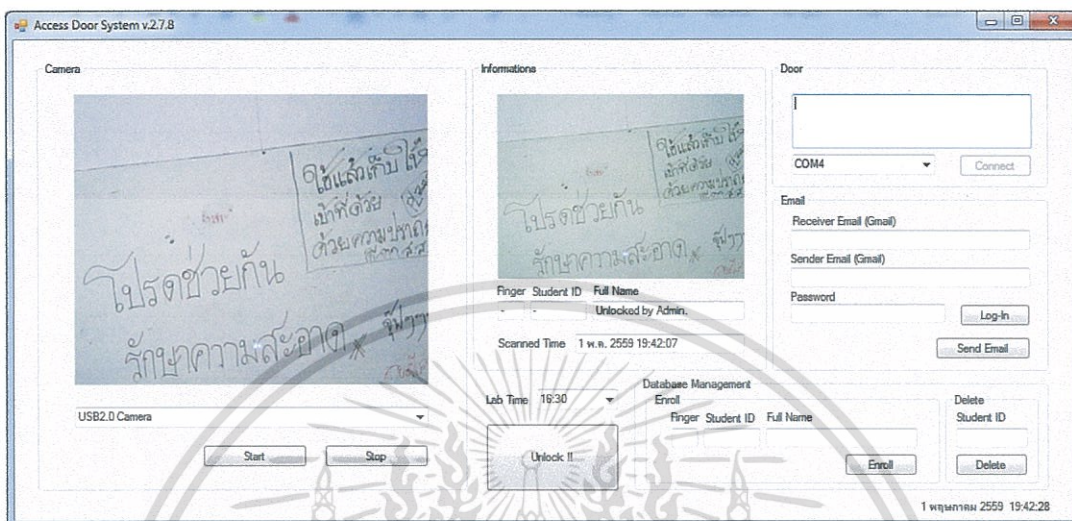
รูปที่ ค.8 หน้าต่างโปรแกรมเมื่อทำการกราดตรวจไม่พบลายนิ้วมือ

2. ประตูจะปลดล็อกเป็นเวลา 3 วินาที จากนั้นประตูจะทำการล็อกอีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

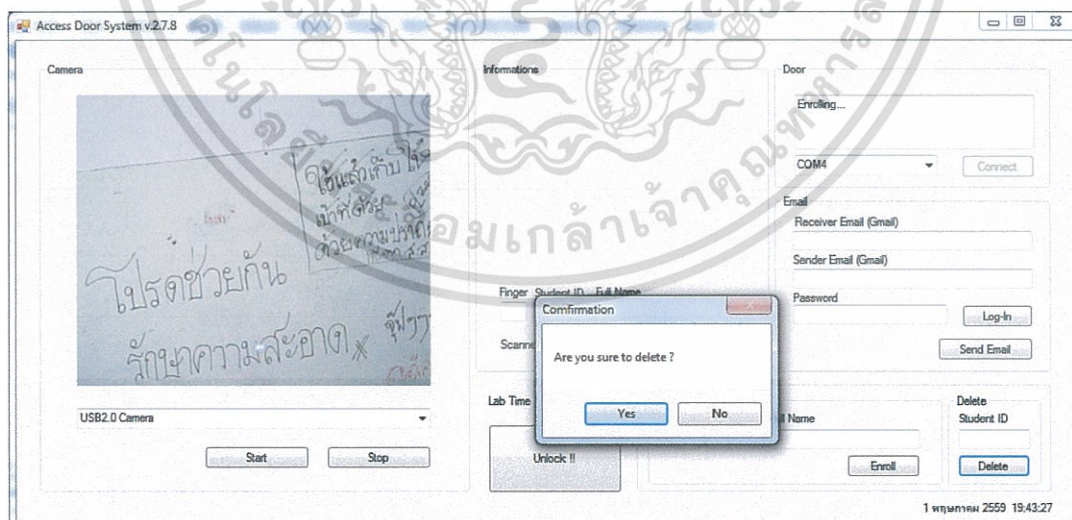
การปลดล็อกโดยใช้ปุ่ม Unlock

1. ทำการกดปุ่ม Unlock เมื่อกดแล้วประตูจะถูกปลดล็อกและแสดงรายละเอียดดังรูปที่ ค.9



รูปที่ ค.9 หน้าต่างโปรแกรมเมื่อกดปุ่ม Unlock

2. ประตูจะปลดล็อกเป็นเวลา 3 วินาที จากนั้นประตูจะทำการล็อกอีกครั้งหนึ่ง
- การลบข้อมูลหลายนิ้วมือ
1. ทำการรกรหัสนักศึกษาในส่วน Database Management ส่วนย่อย Delete โดยกรอกรหัสนักศึกษาในช่อง Student ID แล้วกด Delete จะปรากฏหน้าต่างยืนยันดังรูปที่ ค.10



รูปที่ ค.10 หน้าต่างโปรแกรมเมื่อทำการลบข้อมูลหลายนิ้วมือ

2. หากลบข้อมูลสำเร็จโปรแกรมจะไม่แจ้งเตือนกับผู้ใช้ แต่ถ้าหากมีความผิดพลาดในการลบข้อมูล โปรแกรมจะตอบสนองโดยการแสดงข้อความว่า Cannot delete data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้