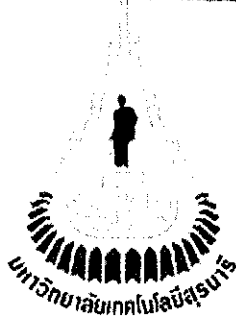




# CONTRIBUTION

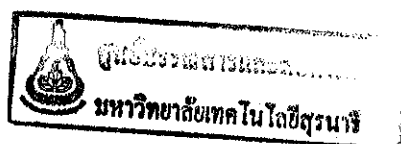


ตรวจสอบรายชื่อด้วยเครื่องอ่านลายนิ้วมือ โดยส่งผ่านข้อมูลแบบไร้สาย

โดย

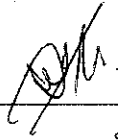
นางสาวจุฬาลักษณ์ เข้มสว่าง	รหัสนักศึกษา B4700739
นางสาวนุชนาฏ ฝาเพ็ญม	รหัสนักศึกษา B4701989
นายโกวิท ภิรมย์กิจ	รหัสนักศึกษา B4706472

รายงานนี้เป็นส่วนหนึ่งของการเรียนรายวิชา  
427499 โครงการวิศวกรรมโทรคมนาคม  
ประจำภาคการศึกษาที่ 3 ปีการศึกษา 2550  
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม  
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



# ตรวจสอบรายชื่อด้วยเครื่องอ่านลายนิ้วมือแบบไร้สาย

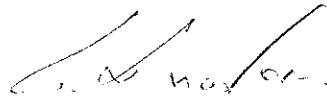
คณะกรรมการสอบโครงการ



(ผู้ช่วยศาสตราจารย์ ดร. วิภาวี หัตถกรรม)  
อาจารย์ที่ปรึกษาโครงการ



(ผู้ช่วยศาสตราจารย์ ดร. รังสรรค์ ทองทา)  
กรรมการ



(ผู้ช่วยศาสตราจารย์ ดร. ชาญชัย ทองโสภณ)  
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นำรายงานโครงการฉบับนี้ เป็นส่วนหนึ่งของ  
การศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม วิชา 427 499 โครงการวิศวกรรม  
โทรคมนาคม ประจำปีการศึกษา 2550

โครงการงาน	ตรวจสอบรายชื่อด้วยเครื่องอ่านลายนิ้วมือ โดยส่งผ่านข้อมูลแบบไร้สาย	
ผู้ดำเนินงาน	นางสาวจุฬาลักษณ์	เข็มสว่าง
	นางสาวนุชนาฏ	ฝาเพ็ญม
	นายโกวิท	ภิรมย์กิจ
อาจารย์ที่ปรึกษา	อ.ดร.วิภาวี	หัตถกรรม
สาขาวิชา	วิศวกรรมโทรคมนาคม	
ภาคการศึกษา	3/2550	

### บทคัดย่อ

โครงการนี้ได้ประยุกต์ใช้เทคโนโลยี Biometric ในการตรวจสอบรายชื่อการเข้าเรียน โดยส่งผ่านข้อมูลแบบไร้สาย เพื่อความสะดวกในการใช้งานจึงประยุกต์จากการใช้สายในการส่งข้อมูลระหว่างเครื่องตรวจสอบลายนิ้วมือและฐานข้อมูลบนคอมพิวเตอร์มาเป็นการส่งข้อมูลผ่าน Wireless แทน โดยการเก็บลายนิ้วมือของนักศึกษากลุ่มตัวอย่างไว้ในฐานข้อมูล โดยใช้ไมโครซอฟต์ แอคเซส (Microsoft Access) เป็นฐานข้อมูลและได้ทำการโปรแกรมการเชื่อมต่อระหว่างฐานข้อมูลและเครื่องอ่านลายนิ้วมือ และได้ทำการตรวจสอบโปรแกรมที่ได้ทำขึ้นในชั้นเรียน ซึ่งทำให้การตรวจสอบรายชื่อ มีความสะดวกรวดเร็วและแม่นยำ และยังสามารถนำโปรแกรมฐานข้อมูลนี้และเครื่องอ่านลายนิ้วมือ ไปประยุกต์ใช้ในชีวิตประจำวันอื่นๆได้อีก

การเรียนการสอนในระดับอุดมศึกษานั้น มีเนื้อหาของการสอนที่มีความละเอียดและลึกซึ้ง จึงต้องใช้ความตั้งใจอย่างสูงในการเรียน ที่จำเป็นอย่างมากในขั้นต้นก็คือการเข้าเรียน และจะมีนักเรียนจำนวนหนึ่งนั้นได้ขาดเรียนด้วยเหตุผลที่ไม่จำเป็น จึงทำให้ต้องมีการตรวจสอบรายชื่อขึ้น ซึ่งการตรวจสอบรายชื่อโดยการเรียกชื่อนั้นก็อาจจะทำให้เสียเวลาในการเรียนการสอนไป โครงการนี้จึงได้ทำการประยุกต์ใช้เทคโนโลยีชีวมาตร (Biometric) ในการตรวจสอบรายชื่อ ด้วยการใช้นิ้วมือของแต่ละบุคคล โดยการเก็บลายนิ้วมือของนักศึกษากลุ่มตัวอย่างไว้ในฐานข้อมูลโดยใช้ไมโครซอฟต์ แอคเซส (Microsoft Access) เป็นฐานข้อมูลและได้ทำการโปรแกรมการเชื่อมต่อระหว่างฐานข้อมูลและเครื่องอ่านลายนิ้วมือ และได้ทำการตรวจสอบโปรแกรมที่ได้ทำขึ้นในชั้นเรียน ซึ่งทำให้การตรวจสอบรายชื่อ มีความสะดวกรวดเร็วและแม่นยำ และยังสามารถนำโปรแกรมฐานข้อมูลนี้และเครื่องอ่านลายนิ้วมือไปประยุกต์ใช้ในชีวิตประจำวันอื่นๆได้อีก

## กิตติกรรมประกาศ (ACKNOWLEDGEMENT)

การทำโครงการชิ้นนี้สำเร็จลงได้เพราะ คณะผู้จัดทำโครงการได้รับความช่วยเหลือจากบุคคลหลายท่าน เริ่มจากอาจารย์ที่ปรึกษาโครงการ อ.ดร.วิภาวี หัตถกรรม อาจารย์สาขาวิชาวิศวกรรมโทรคมนาคม ที่คอยดูแลและให้คำปรึกษาในด้านต่างๆแก่คณะผู้จัดทำอย่างใกล้ชิดมาโดยตลอด ขอขอบคุณบริษัท ไอทีเวิร์ค จำกัด ในการให้คำแนะนำเกี่ยวกับการใช้งานเครื่องอ่านลายนิ้วมือ นาย นพพร พรหมเจริญ นักศึกษาสาขาวิชา วิศวกรรมคอมพิวเตอร์ รุ่นที่ 9 ที่คอยช่วยเหลือและให้คำแนะนำด้านการเขียนโปรแกรม ขอขอบพระคุณท่านคณาจารย์และบุคลากรของสาขาวิชาวิศวกรรมโทรคมนาคมทุกท่าน ที่คอยให้ความช่วยเหลือแก่คณะผู้จัดทำ และขอขอบคุณเพื่อนพี่และน้องสาขาวิชาวิศวกรรมโทรคมนาคมที่คอยช่วยเหลือและเป็นกำลังใจเสมอมา

สุดท้ายนี้คณะผู้จัดทำใคร่ขอกราบขอบพระคุณบิดา มารดา ที่ให้โอกาสในด้านการศึกษา ให้การดูแลเอาใจใส่เลี้ยงดู คอยเป็นกำลังใจและให้การสนับสนุนในทุกๆด้านมาโดยตลอด จึงเห็นสมควรที่จะมอบคุณความดีและเกียรติคุณแก่ทุกๆท่านที่ได้กล่าวถึงและไม่ได้กล่าวถึงมา ณ ที่นี้ด้วย

นางสาวจุฬาลักษณ์ เข้มสว่าง

นายโกวิท ภิรมย์กิจ

นางสาวนุชนาฏ ฝาเพ็ยม

## สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	ฉ
สารบัญรูป	ช
<b>บทที่ 1 บทนำ</b>	<b>1</b>
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของการทำงาน	1
1.4 ขั้นตอนการดำเนินงาน	2
<b>บทที่ 2 ทฤษฎีและหลักการ</b>	<b>3</b>
2.1 เทคโนโลยีชีวมาตร	3
2.2 ขั้นตอนของเทคโนโลยีชีวมาตร	6
2.3 การเปรียบเทียบผลของการใช้เทคโนโลยีชีวมาตรด้วยวิธีที่แตกต่างกัน	7
2.4 ตัวอย่างของเทคโนโลยีชีวมาตร	8
2.4.1 ลายม่านตา	8
2.4.2 ลายนิ้วมือ	10
2.5 การใช้งาน RS232 to RF-Wireless Converter รุ่น ET-RF24G V1.0	13
2.6 โหมดการทำงานของ ET-RF24G V1.0	14
2.6.1 การใช้งานเครื่อง ET-RF24G V1.0 ใน Run Mode	14
2.6.2 การใช้งานเครื่อง ET-RF24G V1.0 ใน Setup Mode	19
2.7 ข้อเสนอแนะในการกำหนดค่า Configuration	23
2.8 BioPrima OEM2000P-44b0 Fingerprint Identification Board	32
<b>บทที่ 3 การทดสอบโปรแกรม</b>	<b>35</b>
3.1 การเก็บลายนิ้วมือ	35

เรื่อง	หน้า
3.2 ทดสอบการทำงานของโปรแกรม	37
<b>บทที่ 4 ขั้นตอนการทำงาน</b>	<b>45</b>
4.1 การทำงานของFINGER PRINT OEM2000P-44b0 เมื่อทำงานเชื่อมต่ออุปกรณ์แบบไร้สาย	45
4.2 ขั้นตอนการใช้งานโปรแกรมสแกนลายนิ้วมือแบบไร้สาย	46
4.2.1 เปิดการใช้งาน ตั้งค่าโปรแกรม และการจัดเก็บลายนิ้วมือ(Enroll)	46
4.3 ขั้นตอนการตรวจสอบลายนิ้วมือ (Identification)	53
<b>บทที่ 5 สรุปผลและข้อเสนอแนะ</b>	<b>55</b>
5.1 สรุปผล	55
5.2 ขั้นตอนการลบข้อมูล เมื่อจำเป็นต้องลบลายนิ้วมือ	58
5.3 ปัญหาและอุปสรรคในการทำโครงงาน	59
5.4 ข้อจำกัดในการทำโครงงาน	59
5.5 ข้อเสนอแนะ	59
<b>ภาคผนวก ก โปรแกรม</b>	<b>60</b>
1. โปรแกรมเพื่อติดต่อกับฐานข้อมูล (DBAccess)	60
2. โปรแกรมที่อ่านข้อมูลจาก Text File (ReadDafile)	62
3. DBManage	64
4. โปรแกรมสำคัญ (frmMain)	65
5. โปรแกรมเพื่อติดต่อกับ Serial Port และกำหนดBaudrete (rs232)	82
6. โปรแกรมเพื่อกำหนดค่าให้กับ Input (ARM7_DEMODlg)	92
6.1 การปิดพอร์ต	105
6.2 การเปิดพอร์ต	107
6.3 การกดปุ่ม Enroll แล้วพิมพ์ลายนิ้วมือ 3 ครั้ง	108
6.4 การตรวจสอบลายนิ้วมือ	112
6.5 การลบข้อมูลลายนิ้วมือจากเครื่องอ่านลายนิ้วมือ	114
6.6 การหมดเวลาพิมพ์ลายนิ้วมือ (5นาที)	116
6.7 โปรแกรมอ่านลายนิ้วมือต้องตั้งค่าจุดบนลายนิ้วมือ	117

เรื่อง	หน้า
ภาคผนวก ข การใช้งานและการดูแลรักษาเครื่องอ่านลายนิ้วมือ	118
เอกสารอ้างอิง	121
ประวัติผู้เขียน	122



## สารบัญตาราง

	หน้า
ตารางที่ 1 แสดงการเปรียบเทียบผลการใช้เทคโนโลยีชีวมาตร	6
ตารางที่ 2 แสดงข้อมูลลายนิ้วมือของนักศึกษากลุ่มตัวอย่าง	14
ตารางที่ 3 แสดงผลการตรวจสอบรายชื่อนักศึกษา	25
ตารางที่ 4 ตารางตัวอย่างการเก็บข้อมูล	30

## สารบัญรูป

	หน้า
รูปที่ 1 แสดงการทำงานของม่านตา	7
รูปที่ 2 แสดงประเภทของลายนิ้วมือ	10
รูปที่ 3 แสดงการตรวจสอบลายนิ้วมือโดยใช้คอมพิวเตอร์ตรวจสอบลักษณะของสัน	10
รูปที่ 4 แสดงการทำงานของเครื่องอ่านลายนิ้วมือ	11
รูปที่ 5 แสดงรูปแบบการทำงาน	26
รูปที่ 6 แสดงผลเมื่อกดปุ่ม Activate sensor & Start capturing	27
รูปที่ 7 แสดงผลเมื่อกดปุ่ม Enroll finger & save to database	28
รูปที่ 8 แสดงหน้าต่างเพื่อให้ใส่หมายเลข ID	29
รูปที่ 9 แสดงหน้าต่างเพื่อให้ใส่ชื่อและนามสกุล	29
รูปที่ 10 แสดงหน้าต่างให้ใส่รหัสประจำตัวนักศึกษา	29
รูปที่ 11 แสดงผลเมื่อทำการกดปุ่ม Load fingers from database	31
รูปที่ 12 แสดงผลเมื่อมีการแตะนิ้วมือลงบนเครื่องอ่านลายนิ้วมือ	32
รูปที่ 13 แสดงหน้าต่างเลือกตาราง CheckData	33
รูปที่ 14 แสดงตัวอย่างรายชื่อนักศึกษาที่เข้าเรียน	34
รูปที่ 15 แสดงวิธีทำความสะอาด	57

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมา

เนื่องจากการเรียนการสอนของทางมหาวิทยาลัยมีเนื้อหาเป็นจำนวนมากซึ่งเวลาในการเรียนการสอนอาจจะไม่เพียงพอทางคณะผู้จัดทำโครงการ จึงตระหนักถึงปัญหาในการใช้เวลาในการตรวจสอบรายชื่อนักศึกษาที่เข้าเรียนซึ่งใช้เวลาในการตรวจสอบรายชื่อนานพอสมควร จึงทำให้เกิดเป็นโครงการนี้ขึ้นมาเพื่อให้สามารถประหยัดเวลาในการตรวจสอบรายชื่อนักศึกษา

โครงการนี้ได้ประยุกต์ใช้เทคโนโลยีชีวมาตร (Biometric) ในการตรวจสอบรายชื่อการเข้าเรียนโดยส่งผ่านข้อมูลแบบไร้สาย เพื่อความสะดวกในการใช้งานจึงประยุกต์จากการใช้สายในการส่งข้อมูลระหว่างเครื่องอ่านลายนิ้วมือและฐานข้อมูลบนคอมพิวเตอร์มาเป็นการส่งข้อมูลแบบไร้สายแทน โดยการเก็บลายนิ้วมือของนักศึกษากลุ่มตัวอย่างไว้ในฐานข้อมูลโดยใช้ไมโครซอฟต์ แอคเซส (Microsoft Access) เป็นฐานข้อมูลและได้ทำการโปรแกรมการเชื่อมต่อระหว่างฐานข้อมูลและเครื่องอ่านลายนิ้วมือ และได้ทำการตรวจสอบโปรแกรมที่ได้ทำขึ้นในชั้นเรียน ซึ่งทำให้การตรวจสอบรายชื่อ มีความสะดวกรวดเร็วและแม่นยำ และยังสามารถนำโปรแกรมฐานข้อมูลนี้และเครื่องอ่านลายนิ้วมือไปประยุกต์ใช้ในชีวิตประจำวันอื่นๆได้อีก

#### 1.2 วัตถุประสงค์

- 1.2.1. เพื่อศึกษาเกี่ยวกับโปรแกรมวิซวล สตูดิโอ 2005 (Visual Studio 2005) เพื่อใช้ตรวจสอบลายนิ้วมือ
- 1.2.2. เพื่อศึกษาเกี่ยวกับโปรแกรม C++ เพื่อใช้เป็นคำสั่งในการตรวจสอบลายนิ้วมือ
- 1.2.3. เพื่อศึกษาเกี่ยวกับโปรแกรมไมโครซอฟต์ แอคเซส (Microsoft Access) เพื่อสร้างให้กับโปรแกรมวิซวล สตูดิโอ 2005 (Visual Studio 2005) และทำการเชื่อมต่อผ่านทางเทคโนโลยีไร้สาย
- 1.2.4. สามารถนำโครงการชิ้นนี้ไปใช้ในการตรวจสอบรายชื่อนักศึกษาที่เข้าเรียนได้จริง และสามารถนำไปใช้ในการเข้าสอบแทนการตรวจสอบจากบัตรนักศึกษาเพื่อเป็นการป้องกันการทุจริต

#### 1.3 ขอบเขตของการทำงาน

- 1.3.1. ศึกษาการนำเครื่องอ่านลายนิ้วมือมาประยุกต์ใช้เข้ากับโปรแกรมฐานข้อมูล

- 1.3.2. เก็บข้อมูลจากนักศึกษาในกลุ่มตัวอย่างผ่านเครื่องอ่านลายนิ้วมือแล้วนำข้อมูลที่ได้บันทึกลงในฐานข้อมูลที่เตรียมจะใช้งาน
- 1.3.3. ประยุกต์โปรแกรมที่จะใช้ในการตรวจสอบมาใช้ในการตรวจสอบรายชื่อของนักศึกษาในกลุ่มตัวอย่างเพื่อทำการทดสอบโปรแกรม
- 1.3.4. ในขั้นตอนของการตรวจสอบรายชื่อนักศึกษาและการเก็บข้อมูลจากนักศึกษาในกลุ่มตัวอย่างนั้นเราจะใช้การส่งผ่านข้อมูลด้วยเทคโนโลยีไร้สาย

#### 1.4 ขั้นตอนการดำเนินงาน

- 1.4.1. ศึกษาค้นหาหาข้อมูลเกี่ยวกับเครื่องอ่านลายนิ้วมือและข้อมูลเกี่ยวกับด้วยเทคโนโลยีไร้สาย
- 1.4.2. ติดต่อขอใช้เครื่องอ่านลายนิ้วมือและ ชุดรับส่งข้อมูลแบบไร้สาย
- 1.4.3. ศึกษาการทำงานของเครื่องตรวจสอบลายนิ้วมือและ ชุดรับส่งข้อมูลแบบไร้สาย
- 1.4.4. เขียนโปรแกรมวิซวล สตูดิโอ 2005 (Visual Studio 2005) และ C++ ในการเชื่อมต่อระหว่างเครื่องอ่านลายนิ้วมือและฐานข้อมูล
- 1.4.5. นำไปใช้งานจริงโดยการทดสอบกับการเรียนจริง
- 1.4.6. สรุปผลและวิเคราะห์การทำงาน

## บทที่ 2

### ทฤษฎีและหลักการ

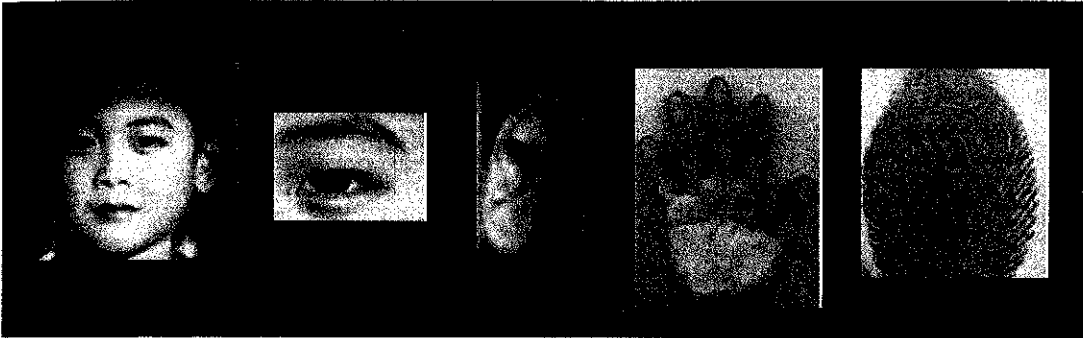
ในปัจจุบันเรามีระบบรักษาความปลอดภัยและพิสูจน์บุคคลหลากหลายรูปแบบ เช่น การป้อนรหัส ลายเซ็น ฯลฯ แต่ปัจจุบันนี้ การกระทำดังกล่าวไม่สามารถรักษาความปลอดภัยได้แล้ว เนื่องจาก การเข้ารหัสนั้นก็ยังมีบุคคลอื่นสามารถถอดรหัสนั้นของเราได้ รวมถึงลายเซ็นที่มีการปลอมแปลงเกิดขึ้นได้ ซึ่งทั้งสองอย่างนี้ก็เกิดขึ้นได้บ่อย ๆ เห็นได้จากข่าวเรื่องการขโมยบัตรเครดิตไปใช้ หรือการกดเงินจากตู้เอทีเอ็ม (ATM) ที่มีการใช้วิธีการต่าง ๆ เพื่อเข้าถึงรหัสบัตรนั้น เป็นต้น ดังนั้น ในหลาย ๆ องค์กรณ์ที่คุณต้องไปติดต่อทำธุรกรรมด้วยตัวคุณเองนั้น มักมีวิธีที่จะนำมาใช้ในการพิสูจน์ว่านั่นคือตัวคุณจริงๆหรือไม่ เพื่อความปลอดภัยขององค์กรเอง ไม่ว่าจะเป็นการตรวจบัตรประชาชน ใบขับขี่หรือแม้กระทั่งบัตรที่มีชื่อและรูปของคุณติดอยู่ เช่นบัตรพนักงาน เป็นต้น

แต่ถ้าหากคุณมีเหตุที่ทำให้ไม่สามารถแสดงบัตรเหล่านั้นได้ เช่น กระเป๋าตังค์โดนล้วงหรือหล่นหาย เราจะมียุทธวิธีในการพิสูจน์ว่าคุณนั้นคือคุณจริงๆได้อย่างไร ด้วยเหตุนี้เองจึงมีการนำวิธีการทางเทคโนโลยีชีวมาตรมาใช้ โดยใช้ส่วนต่าง ๆ ของร่างกายเป็นตัวบ่งชี้ว่าคุณคนคนนั้นเป็นใคร ซึ่งส่วนต่าง ๆ ในร่างกายที่เวลานี้อาจเป็น ลายนิ้วมือ (Fingerprints), ช่องตาดำ (Retina Scan), ม่านตา (Iris Scan), โครงสร้างฝ่ามือ (Palm Geometry), รูปหน้า (Facial Recognition), เสียงพูด (Voice Verification) และการลงลายมือชื่อ (Signature Dynamics) เป็นต้น ซึ่งการพิสูจน์ตัวบุคคลด้วยวิธีนี้ถือเป็นเรื่องที่แม่นยำและมีความน่าเชื่อถือมาก เนื่องจากบุคคลแต่ละคนจะมีลักษณะที่แตกต่างกันในทุก ๆ ส่วน ถึงแม้จะเป็นแฝดเหมือนที่อาจมีรูปหน้าที่คล้ายคลึงกันมาก แต่ลายนิ้วมือ ช่องตาดำ ม่านตา หรือโครงสร้างฝ่ามือจะมีเอกลักษณ์เฉพาะของแต่ละคนเอง

#### 2.1 เทคโนโลยีชีวมาตร (Biometric)

คำว่า ไบโอมेटริกซ์ (Biometrics หรือ Biometry) มีการนำมาใช้กันนับร้อยปีแล้ว โดยเป็นศาสตร์ด้านหนึ่งในการนำเอาวิธีการทางคณิตศาสตร์ หรือวิธีการทางสถิติ มาใช้ในการวิเคราะห์แก้ไขปัญหาทางด้านชีววิทยาต่างๆ เช่น การใช้วิธีทางสถิติวิเคราะห์ผลกระทบของมลพิษที่มีผลต่อสุขภาพของบุคคล, การวิเคราะห์ข้อมูลสภาพอากาศที่มีผลต่อการเพาะปลูก เป็นต้น แต่ความหมายของ เทคโนโลยีชีวมาตร (Biometric) ด้านนี้ไม่ใช่วัตถุประสงค์หลักของกลุ่มวิจัยนี้ แต่เป็นอีกความหมายหนึ่งของ เทคโนโลยีชีวมาตร (Biometric) ซึ่งเป็นศาสตร์ที่เกี่ยวข้องกับการใช้กระบวนการ ในการระบุตัวบุคคลหรือ ตรวจสอบตัวบุคคลโดยอัตโนมัติ โดยใช้ลักษณะทางกายภาพ ที่แตกต่างกันแต่ละบุคคล เช่น รูปแบบของลายนิ้วมือ (Fingerprint) , รูปลักษณะของมือ (Hand Geometry), ลักษณะของเรตินา (Retina Pattern), ลักษณะของม่านตา (Iris Pattern)

รูปลักษณะใบหน้า (Facial) เป็นต้น หรือใช้ลักษณะทางพฤติกรรมของแต่ละบุคคล เช่น เสียง (Voice) , เกล็ดลักษณะในการพิมพ์ (Keystroke Dynamics), ลักษณะท่าทางในการเดิน (Gait recognition) เป็นต้น



กระบวนการที่ทำให้ระบบคอมพิวเตอร์สามารถระบุบุคคลได้โดยอัตโนมัติ นั้นเป็นการเลียนแบบพฤติกรรมของมนุษย์ประเภทหนึ่ง มนุษย์เราใช้วิธีการทางเทคโนโลยีชีวมาตร (Biometric) ในการระบุ ตัวบุคคลอยู่ตลอดเวลา เราใช้ลักษณะจำเพาะทางรูปร่าง ใบหน้า น้ำเสียง หรือแม้กระทั่งกลิ่น ของแต่ละบุคคลในการระบุว่าคนที่เราพบเป็นคนที่เรารู้จักหรือไม่ ดังนั้นจึงถือได้ว่าเทคโนโลยีชีวมาตร (Biometric) เป็นรูปแบบหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence) นั่นเอง

เทคโนโลยีด้านนี้เริ่มมีการนำมาประยุกต์ใช้งานมานานนับสิบปีแล้ว ทั้งในภาครัฐบาลและภาคเอกชน แต่ประสิทธิภาพและความน่าเชื่อถือได้ยัง เป็นที่น่าสงสัยอยู่ อย่างไรก็ตามการที่บุคคลโดยทั่วไป เริ่มมีการใช้งานระบบคอมพิวเตอร์เพิ่มมากขึ้น ความจำเป็นและความสำคัญในการใช้เทคโนโลยีชีวมาตร (Biometric) ในการตรวจสอบตัวบุคคลก็มีความสำคัญและจำเป็นเพิ่มขึ้นไปด้วย

การระบุตัวบุคคลโดยใช้เทคโนโลยีชีวมาตร (Biometric) สามารถนำมาประยุกต์ใช้งานได้ทั้งในภาครัฐบาลและภาคเอกชน เช่น งานทางด้านรักษาความปลอดภัย, ช่วยผู้รักษากฎหมายในการจับตัวผู้กระทำผิด, ช่วยในการตรวจสอบผู้ใช้งานของระบบเครือข่ายคอมพิวเตอร์, การจัดการระบบบริหารงานบุคคล (เช่น งานตรวจสอบเวลาการทำงาน), ช่วยในการตรวจสอบตัวบุคคลในการซื้อขายสินค้าผ่านทางอินเทอร์เน็ต, การจัดการเรื่องการเงินส่วนบุคคลของสถาบันการเงิน เป็นต้น

**ข้อดี** ของการนำเอาเทคโนโลยีชีวมาตร (Biometric) มาใช้ในการตรวจสอบหรือระบุตัวบุคคลคือ

- เทคโนโลยีชีวมาตร (Biometric) ทำให้ผู้ใช้ไม่จำเป็นต้องใช้ความจำ หรือจำเป็นต้องถือบัตรผ่านใดๆ ทำให้สะดวก และรวดเร็ว ผู้ใช้ไม่จำเป็นต้องพกบัตร และต้องจำรหัสผ่าน อีกทั้งยังเป็นการช่วยเพิ่ม ความปลอดภัย และป้องกันการสูญหายของบัตรผ่าน หรือการลักลอบนำเอารหัสผ่านไปใช้
- เทคโนโลยีชีวมาตร (Biometric) ยากต่อการปลอมแปลง และยากต่อการลักลอบนำไปใช้
- การใช้เทคโนโลยีชีวมาตร (Biometric) ทำให้ผู้ใช้ไม่สามารถปฏิเสธความรับผิดชอบได้ เช่นในกรณีของ การใช้รหัสผ่าน หรือบัตรผ่าน เจ้าของบัตรอาจอ้างได้ว่ารหัสผ่านหรือบัตรถูกผู้อื่นลักลอบนำไปใช้ แต่ถ้าใช้ การใช้การตรวจสอบหรือระบุตัวบุคคลด้วยเทคโนโลยีชีวมาตร (Biometric) ทำให้ผู้ใช้ไม่สามารถปฏิเสธความรับผิดชอบได้
- ช่วยลดเวลาในการตรวจสอบลายชื่อ และป้องกันการทุจริตในการเช็คชื่อแทนกัน เช่น ช่วยในการป้องกันพนักงานลงเวลาแทนกัน

เทคโนโลยีชีวมาตร (Biometric) สามารถแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ การใช้ลักษณะทางกายภาพ (Physiological Biometrics) และการใช้ลักษณะทางพฤติกรรม (Behavioural Biometrics) ในการระบุตัวบุคคล

#### **ลักษณะทางกายภาพ (Physiological Biometrics)**

- ลายนิ้วมือ Fingerprint
- ลักษณะใบหน้า Facial Recognition
- ลักษณะของมือ Hand Geometry
- ลักษณะของนิ้วมือ Finger Geometry
- ลักษณะใบหู Ear Shape
- Iris และ Retina ภายในดวงตา
- กลิ่น Human Scent

#### **ลักษณะทางพฤติกรรม (Behavioural Biometrics)**

- การพิมพ์ Keystroke Dynamics
- การเดิน Gait Recognition

- เสียง Voice Recognition
- การเซ็นชื่อ Signature

กระบวนการในการตรวจสอบ หรือระบุตัวบุคคลด้วย เทคโนโลยีชีวมาตร (Biometric) ไม่  
ว่าจะเป็นการใช้ลักษณะเฉพาะแบบใดก็ตาม จะมีขั้นตอนเหมือนกันดังต่อไปนี้

1. ผู้ใช้ระบบต้องทำการให้ตัวอย่าง (Samples) ของลักษณะทางเทคโนโลยีชีวมาตร (Biometric) ที่จะใช้ หรือเป็นการลงทะเบียนเริ่มต้นก่อนที่จะทำการใช้ระบบ
2. ตัวอย่างทางเทคโนโลยีชีวมาตร (Biometric) ที่ถูกเก็บมาในขั้นตอนแรก จะถูกทำการแปลงและจัดเก็บ ให้เป็นแม่แบบ (Template) ที่จะใช้ในการเปรียบเทียบ
3. เมื่อผู้ใช้ต้องการที่จะใช้ระบบ ก็จะถูกตรวจสอบ หรือระบุผู้ใช้ โดยทำการเก็บตัวอย่างทาง เทคโนโลยีชีวมาตร (Biometric) ของผู้ใช้และทำการเปรียบเทียบกับแม่แบบ (Template) ที่เก็บไว้ แล้วทำการตรวจสอบความเหมือนของตัวอย่างกับแม่แบบ จากนั้นก็จะทำการอนุญาต หรือปฏิเสธ การเข้ามาใช้งานระบบของผู้ใช้

เราเรียกขั้นตอนที่ 1 และ 2 ว่าเป็นขั้นตอนของการลงทะเบียน (Enrolment) ซึ่งจะเป็นการทำเพียงครั้งเดียว ก่อนการที่จะเริ่มใช้งาน ส่วนขั้นตอนที่ 3 เป็นกระบวนการตรวจสอบ (Authentication) หรือ ระบุตัวผู้ใช้ (Identification) ซึ่งผลของการตรวจสอบหรือระบุตัวผู้ใช้นี้มีผลออกมาได้ 4 กรณีดังนี้

1. *Correct Accept* : อนุญาตให้ผู้ใช้ที่มีสิทธิใช้ระบบ เข้าใช้ระบบ
2. *Correct Reject* : ปฏิเสธผู้ที่ไม่มีความสิทธิใช้ระบบ
3. *False Accept* : อนุญาตให้ผู้ใช้ที่ไม่มีสิทธิ เข้าใช้ระบบ จำนวนของ False Accept ถ้าคำนวณออกมาเป็นเปอร์เซ็นต์ จะเรียกว่า อัตราการอนุญาตผิดพลาด (False Accept Rate หรือ FAR)
4. *False Reject* : ปฏิเสธผู้ใช้ที่มีสิทธิใช้ระบบ ไม่ให้เข้าใช้ระบบ จำนวนของ False Reject ถ้าคำนวณออกมาเป็นเปอร์เซ็นต์ จะเรียกว่า อัตราการปฏิเสธผิดพลาด (False Reject Rate หรือ FRR)

## 2.2 ขั้นตอนของเทคโนโลยีชีวมาตร

- เก็บตัวอย่างคุณลักษณะที่ต้องการวัด เช่น สแกนลายนิ้วมือออกมาเป็นภาพถ่ายลายนิ้วมือ



- เก็บข้อมูลเทคโนโลยีชีวมาตร (Biometric) จากตัวอย่างที่สแกนได้ จะเก็บข้อมูลเชิงปริมาณจากภาพถ่ายลายนิ้วมือด้วยการคำนวณโดยใช้อัลกอริทึมเฉพาะ
- เปรียบเทียบข้อมูลเชิงปริมาณที่วัดได้จากข้อ 2 กับข้อมูลที่ได้บันทึกเอาไว้ก่อนหน้านี้ ซึ่งอาจบันทึกไว้ในฐานข้อมูลกลาง หรือบันทึกไว้บนสมาร์ทการ์ด
- พิจารณาผลการเปรียบเทียบว่า ถูกต้องตรงกันหรือไม่

ตัดสินใจว่าบุคคลนี้เป็นใคร (Identification) หรือเป็นตัวจริงตามที่มีการกล่าวอ้าง (Verification) หรือไม่

### 2.3 การเปรียบเทียบผลของการใช้เทคโนโลยีชีวมาตร (Biometric) ด้วยวิธีที่แตกต่างกัน

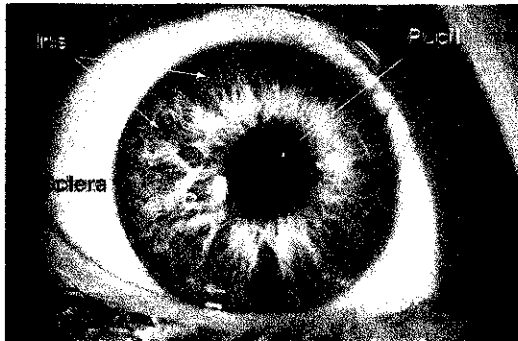
Characteristic	Fingerprints	Hand Geometry	Retina	Iris	Face	Signature	Voice
Base of Use	High	High	Low	Medium	Medium	High	High
Error Incidence	Dryness, dirt, age	Hand injury, age	Glasses	Poor Lighting	Lighting, age, glasses, hair	Changing signatures	Noise, colds, weather
Accuracy	High	High	Very high	Very high	High	High	High
User acceptance	Medium	Medium	Medium	Medium	Medium	Medium	High
Required security level	High	Medium	High	Very high	Medium	Medium	Medium
Long-term stability	High	Medium	High	High	Medium	Medium	Medium

#### ตารางที่ 1 แสดงการเปรียบเทียบผลการใช้เทคโนโลยีชีวมาตร (Biometric)

จากตารางจะเห็นได้ว่าในการใช้เทคโนโลยีภาพด้วยลายนิ้วมือนั้น การเกิดข้อผิดพลาดนั้นมีด้วยกันอยู่ 3 ประการ คือ นิ้วมือแห้ง, สิ่งสกปรก และ อายุ การใช้ลายนิ้วมือนั้นจะให้ความแม่นยำสูง อย่างไรก็ตามการใช้ด้วยวิธีการอื่นๆก็ยังมีข้อดีและข้อเสียที่แตกต่างกันออกไป

## 2.4 ตัวอย่างของเทคโนโลยีชีวมาตร (Biometric)

### 2.4.1 ลายม่านตา



รูปที่ 2.1 แสดงการทำงานของม่านตา

ระบบรู้จำลายม่านตา (Iris Recognition System) เป็นชนิดหนึ่งของระบบตรวจสอบอัตลักษณ์ (Biometric System) ซึ่งเป็นระบบที่ใช้ลักษณะทางกายภาพของแต่ละบุคคลมาทำการยืนยันบุคคลนั้นๆ หรืออาจกล่าวได้ว่าเป็นการใช้ข้อวิยะ หรือส่วนประกอบของร่างกายเป็นรหัสของแต่ละบุคคล ตัวอย่างของลักษณะทางกายภาพที่สามารถนำมาทำการยืนยันบุคคล ได้แก่ ลายนิ้วมือ (Fingerprint), ฝ่ามือ (Palm print), หน้าตา (Face), ลายม่านตา (Iris) หรือ ดีเอ็นเอ (DNA) เป็นต้น

การยืนยันตัวบุคคลเหล่านี้สามารถนำไปประยุกต์ใช้งานกับระบบต่างๆ ได้อย่างมากมาย เช่น การใช้ผ่านเข้าออกสถานที่แทนกุญแจหรือการ์ดต่างๆ, การใช้แทนบัตรประชาชน, การใช้เป็นรหัสในการทำธุรกรรมทางการเงิน, ใช้เข้าถึงข้อมูลส่วนตัวในระบบอินเทอร์เน็ต และการใช้ยืนยันแทนพาสปอร์ต (Passport) เป็นต้น

ระบบรู้จำลายม่านตาเป็นระบบที่มีความแม่นยำ และมีการเปรียบเทียบรหัสที่มีความเร็วสูง เมื่อเปรียบเทียบกับระบบตรวจสอบอัตลักษณ์อื่นๆ ซึ่งทนทานต่อการเปลี่ยนแปลงของสภาพแวดล้อมรอบตัวแต่ละบุคคล อุปกรณ์ที่ใช้ทำการรู้จำลายม่านตานี้ไม่จำเป็นต้องสัมผัส ไม่เหมือนกับการตรวจสอบลายนิ้วมือ ซึ่งช่วยให้ผู้ใช้งานลดความเสี่ยงในการติดเชื้อโรคผ่านทางตัวอุปกรณ์ลง

วิธีการในการรู้จำลายม่านตาโดยทั่วไปจะมีอยู่ 4 ขั้นตอนดังต่อไปนี้

1. **ตรวจหาภาพลายม่านตา** ขั้นตอนนี้จะเป็นทำการตรวจสอบว่าภาพที่ได้รับเข้ามาเป็นภาพของดวงตามนุษย์หรือไม่
2. **ระบุตำแหน่งลายม่านตา** ขั้นตอนนี้จะทำการหาตำแหน่งของลายม่านตามนุษย์
3. **ดึงลักษณะเด่นและเข้ารหัสลายม่านตา** จะนำบริเวณของลายม่านตามาทำการดึงลักษณะเด่น และทำการเข้ารหัสลายม่านตา
4. **เปรียบเทียบรหัสลายม่านตา** รหัสลายม่านตาจะถูกนำมาเปรียบเทียบเพื่อหาความเหมือน และจะสามารถยืนยันในขั้นตอนนี้ได้ว่ารหัสลายม่านตาทั้งสองมาจากลายม่านตาเดียวกันหรือไม่

## 2.4.2 ลายนิ้วมือ

ลายนิ้วมือของแต่ละคน เริ่มปรากฏขึ้นตั้งแต่เป็นตัวอ่อนอายุ 3 ถึง 4 เดือนในครรภ์มารดา ซึ่งเป็นผิวหนังส่วนที่มีร่อง (Furrow) และมีสัน (Ridge) เอาไว้ใช้สำหรับอำนวยความสะดวกในการหยิบจับสิ่งของ สันและร่องที่ปรากฏนี้มีคุณลักษณะที่สำคัญสองประการ คือ ไม่มีการเปลี่ยนแปลงรูปแบบตามกาลเวลา (แต่อาจเปลี่ยนขนาดได้) และ การมีรูปแบบเฉพาะในแต่ละคน

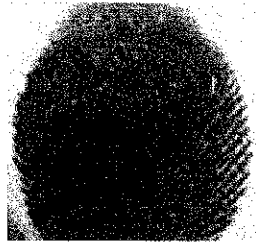
ลายนิ้วมือไม่เปลี่ยนแปลงรูปแบบ (Permanence) ตั้งแต่แรกเกิด จนกระทั่งวันที่เราตาย แต่อาจเปลี่ยนแปลงขนาดได้ตามขนาดร่างกาย เหมือนกับการที่เราวาดรูปไว้บนลูกโป่ง ซึ่งไม่ว่าลูกโป่งจะเล็ก หรือถูกเป่าให้พองใหญ่อย่างไร ก็ยังเป็นรูปเดิมเพียงแต่มีขนาดใหญ่ขึ้นเท่านั้น

การที่ลายนิ้วมือมีรูปแบบเฉพาะในแต่ละคน (Individuality) เป็นคุณสมบัติที่สำคัญอีกอย่างหนึ่งของลายนิ้วมือนั้น ตั้งแต่เริ่มมีการใช้เก็บและเปรียบเทียบลายนิ้วมือโดยใช้วิธีสมัยใหม่ซึ่งมีมาเร็วกว่าปีแล้ว ยังไม่มีการตรวจพบที่มีการเหมือนกันของลายนิ้วมือ อีกทั้งถ้าจะอธิบายด้วยหลักการทางคณิตศาสตร์ และวิทยาศาสตร์ ก็มีการศึกษาของ Sir Francis Galton (1892) ซึ่งได้ประมาณไว้ว่า โอกาสที่คนสองคนจะมีลายนิ้วมือเหมือนกันนั้นมีความน่าจะเป็นอยู่ที่  $1/64,000,000,000$  ซึ่งเป็นการประเมินค่าโดยใช้การแบ่งรายละเอียดรูปแบบของลายนิ้วมือออกเป็นส่วนๆ และหาความน่าจะเป็นของการซ้ำกันของแต่ละส่วนนั้น แล้วนำความน่าจะเป็นของแต่ละส่วนมาคูณกันเพื่อหาความน่าจะเป็นทั้งหมด Sir Francis Galton นี้เป็นผู้ที่เริ่มทำการวิจัยอย่างจริงจังกับลายนิ้วมือ และถือว่าเป็นบุคคลแรกที่ศึกษาถึงการให้ลายนิ้วมือในการระบุตัวบุคคล เป็นบุคคลแรกที่ทำการพิสูจน์ว่าลายนิ้วมือของแต่ละคนมีลักษณะเฉพาะ (Individuality) และไม่มีการเปลี่ยนแปลงรูปแบบ (Permanence) อีกทั้งยังเป็นผู้ที่กำหนดและแบ่งแยกประเภทของรูปแบบลายนิ้วมือที่ใช้กันอยู่จนถึงปัจจุบันนี้

ลายนิ้วมือของแต่ละคนนั้นมีลักษณะเฉพาะมากจนกระทั่งแม้แต่ คู่แฝดแท้ (Identical Twin) ก็ยังมีลายนิ้วมือที่แตกต่างกัน (แต่มีรูปแบบ DNA เหมือนกัน) อย่างไรก็ตามรูปแบบของลายนิ้วมือนั้นมีลักษณะความคล้ายกันของคนภายในครอบครัว หรือพูดได้อีกอย่างหนึ่งว่า รูปแบบของลายนิ้วมือมีการถ่ายทอดกันทางพันธุกรรม ซึ่งรูปแบบของลายนิ้วมือ สามารถแบ่งออกได้เป็นประเภทใหญ่ๆ ได้ 3 ประเภท คือแบบลายก้นหอย (Whorl), ลายมัดหวาย (Loop) และ ลายโค้ง (Arch)



ลายก้นหอย (Whorl)



ลายมัดหวาย (Loop)



ลายโค้ง (Arch)

### รูปที่ 2.2 แสดงประเภทของลายนิ้วมือ

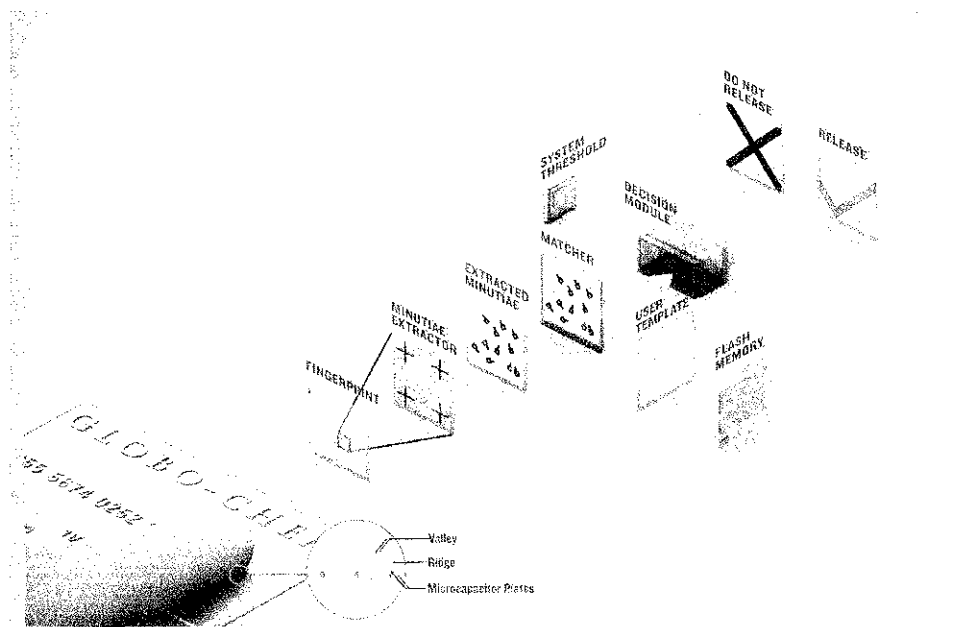
รูปแบบลายนิ้วมือนี้สามารถแบ่งย่อยให้ละเอียดขึ้นไปได้เป็น ลายมัดหวายเฉียงขวา (Right Loop) , ลายมัดหวายเฉียงซ้าย (Left Loop) , ลายโค้งสูงแบบกะโหลก (Tented Arch) เป็นต้น ลายนิ้วมือแบบลายก้นหอย (Whorl) มีประมาณ 30% ลายนิ้วมือแบบลายมัดหวาย (Loop) มีประมาณ 65% และลายนิ้วมือแบบลายโค้ง (Arch) มีประมาณ 5% การแบ่งลายนิ้วมือออกเป็นหลายประเภทนี้เพื่อวัตถุประสงค์ในการเพิ่มความรวดเร็วในการตรวจสอบลายนิ้วมือ แต่ไม่ได้เป็นสิ่งที่ใช้ในการบอกความเหมือน หรือความแตกต่างระหว่างลายนิ้วมือ แต่เป็นการใช้ลักษณะของสัน (Ridge) ของลายนิ้วมือเช่น การสิ้นสุดของสัน (Ridge Ending), สันแบบลายจุด (Dots) , สันที่แตกแขนง (Bifurcations) หรือรูปแบบต่างๆของสันที่เกิดขึ้น เป็นสิ่งที่ใช้ในการเปรียบเทียบลายนิ้วมือ



รูปที่ 2.3 แสดงการตรวจสอบลายนิ้วมือโดยใช้คอมพิวเตอร์ตรวจสอบลักษณะของสัน (Ridge)

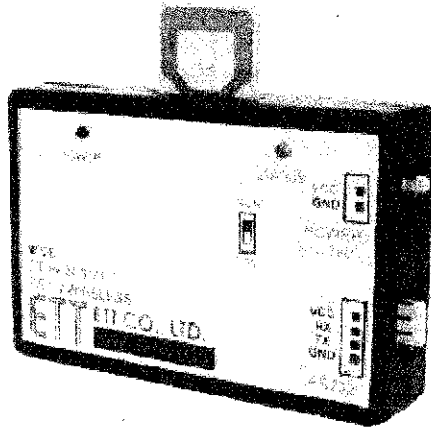
การตรวจสอบตัวบุคคลด้วยลายนิ้วมือนั้น ถือเป็นอีกวิธีหนึ่งในการตรวจสอบหาตัวบุคคลที่ได้ใช้กันอย่างแพร่หลาย เพราะมีวิธีการตรวจสอบที่ง่าย สะดวกและรวดเร็ว และให้ความแม่นยำสูง โดยอาศัยความแตกต่างของลายนิ้วมือของแต่ละบุคคล สันนุนของลายนิ้วมือ รอยของลายนิ้วมือ หลังจากที่มีการตรวจจับแล้วข้อมูลจะถูกเก็บในรูปของสัญญาณข้อมูลดิจิทัลและถูกเก็บไว้ในฐานข้อมูลเพื่อที่จะนำไปประยุกต์ใช้ต่อไป

หลักการการทำงานของเครื่องตรวจสอบลายนิ้วมือ ก่อนอื่นต้องทำการบรรจุนิ้วมือเข้าไป โดยวางนิ้วมือบนอุปกรณ์ตรวจจับ หลังจากนั้นหน่วยประมวลผลก็จะทำการดึงรายละเอียดบางอย่างเฉพาะที่เรียกว่า มินูทีเอ (Minutiae) ลักษณะเด่นเหล่านี้ประกอบด้วยตำแหน่งที่เส้นรอยยูนจบลง ตำแหน่งที่เป็นรอยยูนหลายเส้นมารวมกัน หรือตำแหน่งที่เส้นรอยยูนแยกตัวออกเป็นหลายเส้น ข้อมูลเกี่ยวกับมินูทีเอเหล่านี้จะถูกเข้ารหัสและจัดเก็บไว้พร้อมกับข้อมูลอื่นสำหรับใช้เป็นแม่แบบระบุตัว หลังจากนั้นต้องสัมผัสอุปกรณ์การอ่านลายนิ้วมือของตัวเปรียบเทียบซึ่งเป็นซอฟต์แวร์ที่ทำงานบนหน่วยประมวลผล ตัวเปรียบเทียบจะหาจำนวนของมินูทีเอที่ตรงกันและคำนวณความคล้ายกันของลายนิ้วมือที่เราจะเรียกว่า คะแนนในการเทียบ (matching score) ซึ่งในสถานการณ์อุดมคติ มินูทีเอของแม่แบบอาจจะไม่ตรงกับมินูทีเอที่ได้จากลายนิ้วมือที่นำมาเทียบทั้งหมด ดังนั้น ตัวเปรียบเทียบจะใช้ตัวแปรขีดแบ่ง (threshold) ในการตัดสินใจว่าลายนิ้วมือทั้งสองนั้นเป็นอันเดียวกันหรือไม่



รูปที่ 2.4 แสดงการทำงานของเครื่องอ่านลายนิ้วมือ

## 2.5 การใช้งาน RS232 to RF-Wireless (RF2.4GHz) Converter รุ่น ET-RF24G V1.0



รูปที่ 2.5 ET-RF24G V1.0

### ลักษณะโดยทั่วไป

ET-RF24G V1.0 เป็นชุด Signal Converter สำหรับใช้แปลงสัญญาณระหว่าง RS232 และ RF-Wireless โดยในโหมดการทำงานของการส่งข้อมูล (Transmitter) จะทำหน้าที่รองรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS232 จากขา RX แล้วแปลงเป็นสัญญาณความถี่ (GFSK) ส่งออกไปในอากาศ และในทางกลับกันในโหมดการทำงานแบบรับ (Receiver) ชุด ET-RF24G V1.0 ก็จะทำหน้าที่คอยตรวจจับข้อมูลที่อยู่ในรูปของสัญญาณความถี่ (GFSK) จากด้าน RF เพื่อแปลงกลับเป็นข้อมูลแบบ RS232 ส่งออกไปทางขา TX ได้ด้วย

ซึ่งจะเห็นได้ว่าชุดแปลงสัญญาณ ET-RF24G V1.0 นั้น สามารถนำไปต่อใช้งานร่วมกับพอร์ตสื่อสารอนุกรม แบบ RS232 เพื่อใช้งานในลักษณะของการสื่อสารอนุกรมแบบไร้สาย (Wireless Transceiver) ได้โดยตรงโดยจะมีข้อดีกว่า คือ สามารถรับส่งข้อมูลกันได้ในระยะทางที่ไกลกว่า RS232 หลายเท่าตัว และประการสำคัญ คือ ไม่จำเป็นต้องใช้สายสัญญาณที่เป็นตัวสัญญาณทางไฟฟ้าในการสื่อสารข้อมูลกัน ทำให้สามารถเปลี่ยนแปลงหรือเคลื่อนย้ายจุดรับส่งข้อมูลได้ตลอดเวลา ซึ่งถ้าเป็นการรับส่งข้อมูลด้วยระบบ RS232 แบบที่ใช้สายสัญญาณนั้นจะเกิดความยุ่งยากในการติดตั้งสายสัญญาณเป็นอย่างมาก

แต่อย่างไรก็ตามการรับส่งข้อมูลโดยใช้อากาศเป็นตัวกลางในการสื่อสารนั้น ก็มีข้อจำกัดบางประการเหมือนกัน โดยเฉพาะอย่างยิ่ง เรื่องความน่าเชื่อถือของข้อมูลที่รับส่งกัน ซึ่งมีโอกาสผิดพลาดหรือสูญหายได้เหมือนกัน เนื่องจากในการลำเลียงข้อมูลนั้นไม่ได้ใช้สายสัญญาณเป็นตัวกลางในการรับส่งข้อมูล แต่ใช้อากาศเป็นตัวกลางในการรับส่งข้อมูลแทน ซึ่งมีโอกาสที่ข้อมูลจะ

เกิดการรบกวนจากสัญญาณอื่นๆ ที่มีย่านความถี่ใกล้เคียงกันแล้วทำให้ข้อมูลผิดเพี้ยนไปได้บ้างเหมือนกัน ซึ่งระบบการจัดการข้อมูลของเครื่อง ET-RF24G V1.0 นั้น มีระบบการเข้ารหัสและถอดรหัสข้อมูลที่มีความน่าเชื่อถืออยู่ในเกณฑ์ที่จัดว่าดี โดยข้อมูลแต่ละ Byte ที่มีการรับส่งกันนั้น จะมีการตรวจสอบความถูกต้องของข้อมูลให้ด้วยแล้ว โดยข้อมูลที่รับได้จากด้าน RF นั้นรับประกันได้ว่าเป็นข้อมูลที่มีความถูกต้องแน่นอน แต่อย่างไรก็ตามการรับส่งข้อมูลนั้นมีโอกาสผิดพลาดในเรื่องของการสูญหายของข้อมูลบ้างเหมือนกัน เนื่องจากกลไกในการรับส่งข้อมูลของเครื่อง ET-RF24G V1.0 นั้น จะมีการตรวจสอบข้อมูลทุก Byte ที่รับได้จาก RF เสมอ ซึ่งถ้าพบว่ามีผิดพลาดเกิดขึ้นจะทิ้งข้อมูล Byte นั้นไป ซึ่งผู้ใช้ควรมีกลไกในการตรวจสอบข้อมูลที่รับส่งกันว่าครบถ้วนหรือไม่ด้วย ซึ่งหากพบว่ามี การสูญหายของข้อมูลเกิดขึ้นก็ให้ร้องขอให้มีการส่งข้อมูลนั้นซ้ำนั้นๆ ใหม่อีกครึ่งหนึ่ง ก็จะสามารถแก้ไขปัญหาดังกล่าวได้

## 2.6 โหมดการทำงาน

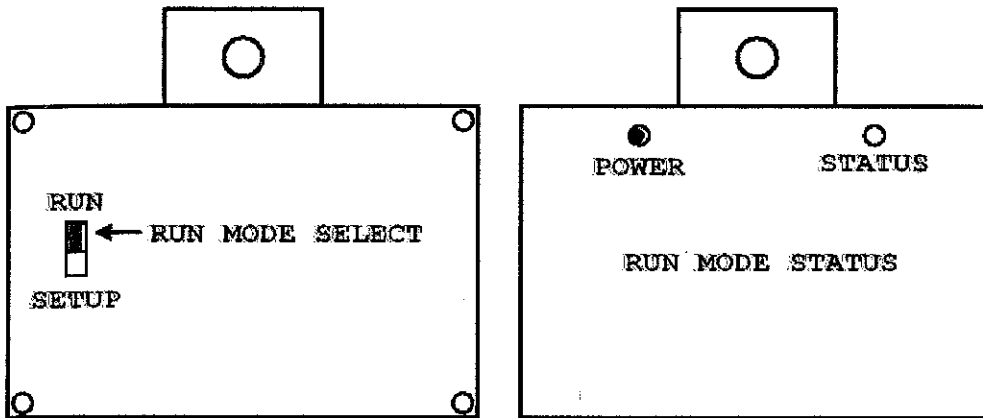
สำหรับโหมดการทำงานของ ET-RF24G V1.0 นั้นจะแบ่งออกเป็น 2 โหมด ด้วยกัน โดยการกำหนดโหมดการทำงานของ ET-RF24G V1.0 นั้นจะการทำผ่าน Switch เล็กโหมดซึ่งอยู่ด้านใต้กล่อง โดยการเลือกโหมดการทำงานนั้นจะต้องกระทำให้เสร็จเรียบร้อยก่อนการจ่ายไฟให้กับ ET-RF24G V1.0 ด้วยเสมอ เนื่องจากการทำงานของเครื่อง ET-RF24G V1.0 นั้นจะทำการตรวจสอบโหมดการทำงานของเครื่องจาก Switch เล็กโหมดเฉพาะในช่วงของการจ่ายไฟเลี้ยงให้เครื่องเริ่มต้นทำงานครั้งแรก (Power-ON) เท่านั้น ซึ่งการเปลี่ยนแปลงตำแหน่งการทำงานของ Switch เล็กโหมด หลังจากทำการจ่ายไฟให้กับ ET-RF24G V1.0 ไปแล้ว จะไม่มีผลต่อการทำงานของเครื่องแต่อย่างใด โดยการทำงานของเครื่อง ET-RF24G V1.0 นั้นจะมี LED แสดงสถานการณ์ทำงานของเครื่องจำนวน 2 หลอด คือ LED POWER ซึ่งเป็น LED สีแดง โดยที่ LED POWER นี้จะติดสว่างให้เห็นตลอดเวลาที่มีการจ่ายไฟเลี้ยงให้เครื่องทำงานอยู่ ส่วน LED อีกดวงหนึ่งนั้นจะเป็น LED สีเขียว ใช้แสดงสถานการณ์ทำงานของเครื่องซึ่งเรียกว่า LED STATUS โดย LED STATUS นี้จะเกิดการกะพริบตามจังหวะของการรับส่งข้อมูลกันในแต่ละครั้งโดยในสถานะปกตินั้น ถ้าเครื่องทำงานอยู่ใน RUN MODE หลอด LED STATUS จะดับอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูล แต่ถ้าตัวเครื่องทำงานอยู่ใน SETUP MODE หลอด LED STATUS จะติดอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูล โดยโหมดการทำงานของ ET-RF24G V1.0 จะมีอยู่ด้วยกัน 2 โหมด คือ

### 2.6.1 การใช้งานเครื่อง ET-RF24G V1.0 ใน Run Mode

การใช้งานใน Run Mode ซึ่งเป็นโหมดของการใช้งานตามปกติของเครื่อง โดยเมื่อเครื่อง ET-RF24G V1.0 เข้าทำงานในโหมดนี้แล้ว จะสังเกตเห็นหลอดไฟแสดงสถานการณ์ทำงาน หรือ



LED STATUS ดับอยู่ แต่เมื่อมีการรับหรือ ส่งข้อมูลเกิดขึ้น สถานการณ์ทำงานของ LED STATUS จึงจะกระพริบตามจังหวะของการรับส่งข้อมูลนั้นๆ แต่ยังไม่มีการรับส่งข้อมูลกัน LED STATUS จะดับอยู่ตลอดเวลา



รูปที่ 2.6 แสดงการเลือกโหมดการทำงาน สำหรับใช้งานปกติ (Run Mode)

สำหรับการทำงานใน Run Mode นั้น จะแบ่งลักษณะการทำงานออกเป็น 3 แบบด้วยกัน โดยลักษณะการทำงานนี้ จะถูกกำหนดไว้แล้วใน Configuration ของเครื่องใน Setup Mode ดังนั้นก่อนการใช้งานเครื่องในครั้งแรกจะต้องทำการกำหนดค่า Configuration ต่างๆ ให้เรียบร้อยเสียก่อน โดยเมื่อเครื่อง ET-RF24G V1.0 เริ่มต้นเข้าทำงานใน Run Mode แล้วมันจะทำการอ่านค่า Configuration ที่เก็บไว้ออกมา เพื่อใช้เป็นเงื่อนไขในการทำงานตามค่าที่ได้กำหนดไว้ โดยลักษณะการทำงานใน Run Mode แบ่งออกเป็นดังนี้

#### 2.6.1.1 การทำงานแบบ RF Receive Only

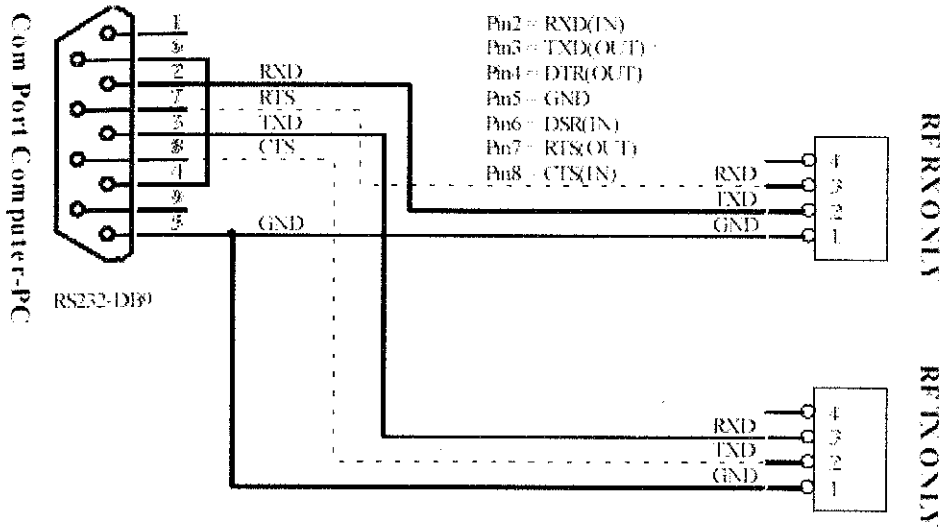
เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้ จะเป็นการรอรับข้อมูลความถี่แบบ GFSK จากด้าน RF แล้วเปลี่ยนเป็นข้อมูลอนุกรมส่งออกไปทางขา TX (Transmit) ของ RS232 โดยการทำงานจะวนรอบอยู่เช่นนี้ไปตลอด ซึ่งในการใช้งานเครื่อง ET-RF24G V1.0 ในโหมดนี้จะต้องนำสัญญาณ TX (Transmit) ไปต่อกับขาสัญญาณ RX (Receive) ของอุปกรณ์ด้านตรงข้าม (RS232 ของคอมพิวเตอร์ PC) โดยในโหมดนี้ การทำงานของขาสัญญาณ RX ด้าน RS232 ของเครื่อง ET-RF24G V1.0 จะถูกเปลี่ยนหน้าที่เป็นสัญญาณ CTS (Clear To Send) สำหรับใช้ตรวจสอบความพร้อมในการส่งข้อมูลไปให้อุปกรณ์ด้านตรงข้ามแทน ซึ่งในการใช้งานจะต้องนำสัญญาณนี้ไปต่อเข้ากับสัญญาณ RTS (Ready To Send) ของอุปกรณ์ด้านตรงข้าม โดยเครื่อง ET-RF24G V1.0 จะทำการตรวจสอบสถานะของสัญญาณ RX ซึ่งในโหมดนี้เปรียบเสมือน CTS ว่ามีค่าเป็น "0" หรือไม่ โดยถ้าพบว่าเป็น "0" จึงจะส่งข้อมูลออกไปให้

ทางขา TX แต่ถ้าพบว่าสถานะของขาสัญญาณนี้มีค่าเป็น "1 แสดงว่าอุปกรณ์ด้านตรงข้ามยังไม่พร้อมรับข้อมูลก็จะรอจนกว่าจะพบว่าสถานะของสัญญาณดังกล่าวมีค่าเป็น "0 จึงจะส่งข้อมูลออกไปได้ โดยเครื่อง ET-RF24G V1.0 จะสามารถจัดเก็บข้อมูลไว้ใน Buffer เพื่อรอการส่งได้สูงสุด 64 Byte เท่านั้น ซึ่งถ้าในระหว่างที่รอความพร้อมอยู่นั้น มีข้อมูลด้าน RF ส่งเข้ามาเกินกว่า 64 Byte จะทำให้ข้อมูลที่เกินมานั้นสูญหายไป

#### 2.6.1.2 การทำงานแบบ RF Transmit Only

เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้จะมีลักษณะตรงกันข้ามกับ RF Receive Only กล่าวคือ เครื่อง ET-RF24G V1.0 จะทำหน้าที่รอรับข้อมูลจากขา RX (Receive) ด้าน RS232 แล้วเปลี่ยนเป็นข้อมูลแบบ GFSK ส่งออกไปทางด้าน RF โดยการใช้งานเครื่องในโหมดนี้ จะต้องนำสัญญาณ TX (Transmit) ซึ่งเป็นขาส่งข้อมูลจาก RS232 ของอุปกรณ์ด้านตรงข้ามมาต่อเข้ากับขา RX (Receive) ของเครื่อง ET-RF24G V1.0 ส่วนขาสัญญาณ TX จะถูกเปลี่ยนหน้าที่เป็น (Ready To Send) เพื่อใช้แสดงสถานะความพร้อมในการรับข้อมูลจากด้าน RS232 ซึ่งในการใช้งานจะต้องนำสัญญาณ TX ซึ่งในขณะนี้เปรียบเสมือนกับ RTS นำไปต่อกับสัญญาณ CTS (Clear To Send) ของอุปกรณ์ด้านตรงข้าม เพื่อใช้ในการตรวจสอบความพร้อมในการรับข้อมูลของเครื่อง ET-RF24G V1.0 ด้วย โดยถ้าเครื่อง ET-RF24G V1.0 พร้อมรับข้อมูลจาก RS232 มันจะส่งสัญญาณ RTS ให้มีค่าเป็น "0" รอไว้ และเมื่อใดก็ตามที่การรับข้อมูลทางด้านของ RS232 มีจำนวนข้อมูลที่ยังไม่สามารถเปลี่ยนเป็น GFSK เพื่อส่งออกไปทางด้าน RF ได้ทันจนเกือบจะเต็ม Buffer แล้วเครื่อง ET-RF24G V1.0 จะทำการส่งสัญญาณ RTS ให้เป็นค่าเป็น "1" ออกไปบอกให้อุปกรณ์ด้านตรงข้ามทราบเพื่อจะได้หยุดการส่งข้อมูลออกมา โดยอุปกรณ์ด้านตรงข้ามจะต้องหยุดการส่งข้อมูลและรอจนกว่าสถานะของสัญญาณ RTS จะกลับเป็น "0" จึงจะเริ่มส่งข้อมูลออกมาใหม่ ซึ่งหลักจากที่เครื่อง ET-RF24G V1.0 ส่งสัญญาณ RTS ด้วยค่า "1" ออกไปแล้ว จะยังคงสามารถรับข้อมูลได้เพิ่มเติมอีกไม่เกิน 16 Byte เท่านั้น ซึ่งถ้าอุปกรณ์ด้านตรงข้ามยังส่งข้อมูลต่อเนื่องมาอีกจนเกินขนาดของ Buffer ที่เครื่อง ET-RF24G V1.0 จะรับไว้ได้จะทำให้ข้อมูลที่เกินมานั้นเกิดการสูญหายได้

โดยเราสามารถนำเครื่อง ET-RF24G V1.0 จำนวน 4 ชุด มาต่อใช้งานร่วมกัน เพื่อใช้งานในการรับส่งข้อมูลกันแบบ Full Duplex โดยแบ่งการใช้งานออกเป็น 2 ด้าน คือ ทางต้นทาง และปลายทาง ด้านละ 2 ชุด โดยแต่ละด้านให้กำหนดหน้าที่การทำงานเป็น RF Receive Only 1 ชุด และ RF Transmit Only อีก 1 ชุด



รูปที่ 2.7 แสดงสายสัญญาณ RS232 เพื่อใช้กับ ET-RF24G ในโหมด RF Receive Only และ RF Transmit Only

### 2.6.1.3 การทำงานแบบ RF Auto Direction

เป็นการทำงานชนิด 2 ทิศทาง แบบ Half Duplex หรือ ผลัดกันรับผลัดกันส่ง ซึ่งสามารถรับส่งข้อมูลระหว่างต้นทาง และ ปลายทางได้ โดยใช้เครื่อง ET-RF24G V1.0 ด้านละ 1 ชุด เท่านั้น เพียงแต่การรับส่งข้อมูลแบบนี้จะไม่สามารถส่งข้อมูลสวนทางกันได้เหมือนกับแบบ Full Duplex แต่จะต้องใช้วิธีการผลัดกันรับข้อมูลและส่งข้อมูลแทน โดยเมื่อฝ่ายรับทำการรับข้อมูลได้จนครบแล้วจึงจะสลับหน้าที่เป็นฝ่ายส่งเพื่อส่งข้อมูลย้อนกลับไป

โดยในโหมดนี้ เครื่อง ET-RF24G V1.0 จะทำหน้าที่เป็นฝ่ายรับ และฝ่ายส่งข้อมูลแบบอัตโนมัติ โดยในสภาวะปกติจะอยู่ในสภาวะของการรอรับข้อมูล ทั้งด้าน RF และ RS232 ซึ่งถ้าพบว่ามีข้อมูลส่งเข้ามาทางด้านของ RF ก็จะนำข้อมูลนั้นส่งออกไปทางด้านขา TX ของ RS232 ทันที และในทำนองเดียวกัน ถ้าพบว่ามีข้อมูลส่งเข้ามาทางด้าน RX ของ RS232 มันก็จะทำการรับข้อมูลนั้นจาก RS232 พร้อมกับเปลี่ยนทิศทางของอุปกรณ์ RF จากการรอรับข้อมูลทำหน้าที่เป็นตัวส่งข้อมูลแทน เพื่อทำการส่งข้อมูลที่รับได้จาก RS232 ออกไปทาง RF ในทันที ซึ่งหลังจากที่เครื่อง ET-RF24G V1.0 ทำการสลับโหมดการทำงานของอุปกรณ์ด้าน RF จากการรอรับเป็นการส่งและทำการเริ่มต้นส่งข้อมูลออกไปทางด้าน RF เรียบร้อยแล้ว มันจะวนกลับไปตรวจสอบการรับข้อมูลจากด้าน RS232 อีกว่ายังมีข้อมูลส่งเข้ามาอีกหรือไม่ ถ้าพบว่ามีข้อมูลส่งเข้ามาก็จะทำการแปลงข้อมูลนั้นเพื่อส่งออกไปยังด้าน RF ต่อไปอีกจนกว่าการส่งข้อมูลด้าน RS232 จะสิ้นสุด

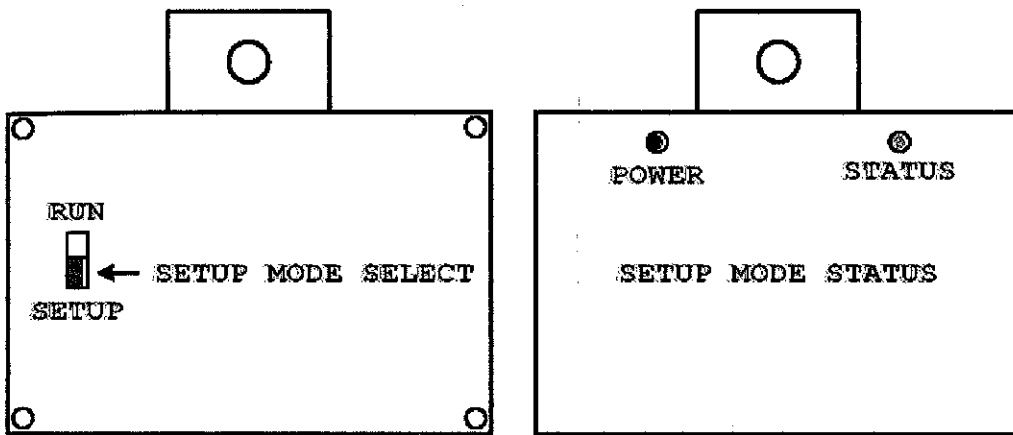
ลง ซึ่งข้อมูลด้าน RS232 ที่ส่งเข้ามานั้น ควรส่งอย่างต่อเนื่องโดยเมื่อเครื่อง ET-RF24G V1.0 ทำการส่งข้อมูลแต่ละ Byte ออกไปทางด้าน RF เรียบร้อยแล้วมันจะวนรอบรอรับข้อมูล Byte ถัดไปจาก RS232 ภายในเวลา 2.5 ms ถ้าไม่พบข้อมูลส่งเข้ามาอีกภายในระยะเวลาดังกล่าวมันจึงจะทำการเปลี่ยนหน้าที่ของอุปกรณ์ด้าน RF ให้กลับมาทำหน้าที่เป็นการรอรับข้อมูลตามเดิม โดยในขณะที่อุปกรณ์ด้าน RF ถูกกำหนดให้เป็นฝ่ายส่งข้อมูลอยู่นั้น จะไม่สามารถทำการรับข้อมูลจาก RF ได้ ซึ่งถ้ามีการส่งข้อมูลเข้ามาในขณะนั้นก็จะไม่สามารถรับได้ โดยค่าเวลาที่จะใช้ในการสลับโหมดการทำงานของ RF จากฝ่ายส่งข้อมูลให้เป็นฝ่ายรับข้อมูลจะมีค่าเป็น 2.5 ms ดังนั้นเมื่อฝ่ายรับสามารถรับข้อมูลได้ครบหมดแล้วก่อนที่จะทำการส่งข้อมูลเพื่อตอบกลับไปยังฝ่ายตรงข้ามนั้น ควรทำการหน่วงเวลาไว้ไม่น้อยกว่า 3 ms นับจากรับข้อมูล Byte สุดท้ายได้เรียบร้อยแล้วจึงเริ่มต้นส่งข้อมูล Byte แรกย้อนกลับไป ซึ่งถ้าฝ่ายรับทำการส่งข้อมูลตอบกลับไปยังฝ่ายตรงข้ามเร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้ทัน

สำหรับการใช้งานเครื่อง ET-RF24G V1.0 ในโหมด RF Auto Direction นี้ การรับ และส่ง ข้อมูลด้าน RS232 จะไม่มีการตรวจสอบความพร้อมของฝ่ายรับและส่ง ด้วยสัญญาณทางไฟฟ้า (CTS/RTS) เหมือนกับการใช้งานใน 2 โหมดที่ผ่านมาแล้ว โดยเมื่อมันสามารถรับข้อมูลจาก RF ได้ ก็จะมีการส่งข้อมูลนั้นออกทางขา TX (Transmit) ของ RS232 ในทันที โดยไม่สนใจว่า อุปกรณ์ที่ต่อไว้ด้าน RS232 จะพร้อมส่งข้อมูลหรือไม่ ซึ่งถ้าด้าน RS232 ไม่พร้อมรับข้อมูลก็จะทำให้ข้อมูล Byte นั้นสูญหายไปทันที ซึ่งในการใช้งานนั้น ผู้ใช้ควรกำหนดค่าความเร็วในการรับส่งข้อมูลด้าน RS232 ที่จะใช้เครื่อง ET-RF24G V1.0 ทุกๆ ตัวด้วยค่าความเร็วที่เท่ากันด้วย เพื่อให้การรับและส่งข้อมูลเกิดความสัมพันธ์กันอย่างเหมาะสม

สำหรับความสามารถในการรอรับข้อมูลจาก RS232 ของเครื่อง ET-RF24G V1.0 ในโหมดนี้ จะสามารถรับข้อมูลได้อย่างต่อเนื่องสูงสุด ไม่เกิน 64 Byte ต่อเนื่องกันนั้น ควรทำการแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดชุดละไม่เกิน 64 Byte ซึ่งหลังจากทำการส่งข้อมูลอย่างต่อเนื่องไปได้ 1 ชุด (64Byte) แล้วควรทำการหน่วงเวลาไว้ชั่วขณะหนึ่งอย่างน้อย 1 ms แล้วจึงเริ่มส่งข้อมูลชุดถัดไป สลับกับการหน่วงเวลาอย่างนี้เรื่อยๆ เพื่อให้เครื่อง ET-RF24G V1.0 สามารถนำข้อมูลที่รับได้จากด้าน RS232 ส่งออกทางด้าน RF ได้ทัน ซึ่งทำการส่งข้อมูลอย่างต่อเนื่องโดยไม่มีการหน่วงเวลาเลยอาจทำให้ข้อมูลบาง Byte เกิดการสูญหายไป

## 2.6.2 การใช้งานเครื่อง ET-RF24G V1.0 ใน Setup Mode

การใช้งานเครื่อง ET-RF24G V1.0 ใน Setup Mode ซึ่งเป็นโหมดสำหรับใช้กำหนดค่า Configuration ต่างๆ สำหรับควบคุมการทำงานของเครื่อง ET-RF24G V1.0 ที่จะใช้ในขณะที่เครื่องทำงานอยู่ใน Run Mode โดยในการ Setup ค่า Configuration ต่างๆ นั้นจะกระทำร่วมกับโปรแกรม "ET\_RF24G\_V1.EXE" ของ อีทีที ซึ่งเมื่อเครื่อง ET-RF24G V1.0 เข้าทำงานในโหมด Setup แล้ว จะสังเกตเห็นหลอดไฟแสดงสถานะการทำงาน หรือ LED STATUS ติดสว่างค้างอยู่ตลอดเวลา แต่เมื่อมีการตั้งอ่านหรือเขียนข้อมูลกับบอร์ด สถานะการทำงานของ LED STATUS จึงจะกระพริบตามจังหวะของการรับส่งข้อมูล แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED STATUS จะติดค้างอยู่ตลอดเวลา



รูปที่ 2.8 แสดงการเลือกโหมดการทำงาน สำหรับกำหนดค่า Configuration (Setup Mode)

ซึ่งการกำหนดค่า Configuration ให้กับ ET-RF24G V1.0 นั้น จะต้องกระทำในขณะที่ยังตัวเครื่องทำงานอยู่ใน Setup Mode เท่านั้น (เลือก Switch กำหนดโหมดไว้ทางด้าน Setup แล้วจ่ายไฟให้เครื่องเริ่มต้นทำงาน) โดยค่าของ Configuration ต่างๆ นั้นจะถูกใช้สำหรับเป็นเงื่อนไขในการทำงานของ ET-RF24G V1.0 ในขณะที่อยู่ใน Run Mode ดังนั้น ก่อนการเริ่มต้นใช้งานเครื่องในครั้งแรกนั้น จึงจำเป็นอย่างยิ่งที่จะต้องทำการกำหนดค่าของ Configuration ต่างๆ ให้ถูกต้องและตรงกับความต้องการที่จะใช้งานเสียก่อน โดยเมื่อทำการกำหนดค่าตัวเลือกต่างๆ ของ Configuration เรียบร้อยแล้ว ก็สามารถเปลี่ยนโหมดการทำงานของตัวเครื่องกลับเป็น Run Mode พร้อมกับการปิดไฟที่จ่ายให้กับตัวเครื่อง (Power-OFF) ช่วงขณะหนึ่ง จากนั้นจึงเริ่มต้นจ่ายไฟให้กับตัวเครื่องใหม่ (Power-ON) ก็สามารถใช้งาน ET-RF24G V1.0 ตามค่าของ Configuration ที่กำหนดไว้แล้วจะถูกเก็บไว้ภายในตัวเครื่องอย่างถาวร ถึงแม้ว่าจะไม่ได้ทำการจ่ายไฟให้กับตัวเครื่องแล้วก็ตาม ดังนั้นเมื่อทำการกำหนดค่า Configuration ต่างๆ เรียบร้อย ถ้าไม่มีการ

เปลี่ยนแปลงเงื่อนไขการทำงานของตัวเครื่องต่างไปจากเงื่อนไขเดิมที่ได้กำหนดไว้แล้ว ก็ไม่จำเป็นต้องทำการกำหนดค่า Configuration ใหม่อีกแต่อย่างไร โดยทุกๆครั้งที่เริ่มต้นจ่ายไฟเข้าเครื่องในครั้งแรกนั้น การทำงานของ ET-RF24G V1.0 จะเป็นไปตามเงื่อนไขที่กำหนดไว้ใน Configuration เสมอทุกๆครั้ง โดยคุณสมบัติของ Configuration ต่างๆ นั้นมีดังนี้

ET-RF 2.4 GHz (RS232 <--> Wireless Converter) Version 1.00

SETUP CONFIG MODE | TEST RUN MODE | HELP MODE

**User RS232 Baudrate**

1200 BPS

2400 BPS

4800 BPS

9600 BPS

19200 BPS

**Setup Communication Port**

Com Port Select

Status

BIOS Code

**RF Data Rate**

250 Kbps Data rate

1 Mbps Data Rate

**RF Operation Mode**

RF Receive Only

RF Transmitt Only

RF Auto Direction

**RF Power Gain**

-20 dBm (Min)

-10 dBm

-5 dBm

+0 dBm (Max)

**Select RXD ID Code**

RF Receiver ID Code

**Select TXD ID Code**

RF Transmitter ID Code

**Select RF Frequency Channel**

RF Frequency Channel

รูปที่ 2.9 แสดงรูปโปรแกรมที่ใช้สำหรับกำหนดค่า Configuration ของ ET-RF24G V1.0

- User RS232 Baudrate ใช้สำหรับกำหนดค่าความเร็วในการรับส่งข้อมูลทางด้าน RS232 ของตัวเครื่อง ในขณะที่ทำงานอยู่ใน Run Mode ซึ่งสามารถกำหนดได้ 5 ค่าคือ

- 1200 BPS
- 2400 BPS
- 4800 BPS
- 9600 BPS
- 19200 BPS

- **RF Data Rate** ใช้สำหรับกำหนดความเร็วในการรับส่งข้อมูลทางด้าน RF ของ ET-RF24G V1.0 ซึ่งจะต้องกำหนดให้เครื่อง ET-RF24G V1.0 ทุกๆตัว ที่จะนำมาใช้ติดต่อสื่อสารกัน มีค่าอัตราความเร็วในการรับส่งข้อมูลด้าน RF หรือ RF Data Rate นี้มีค่าเท่ากันทั้งหมด ซึ่งถ้ากำหนดค่าความเร็วต่างๆกันจะไม่สามารถรับส่งข้อมูลกันได้ ซึ่งค่าอัตราความเร็วในการส่งข้อมูลนี้จะมีผลต่อระยะทางการรับส่งข้อมูลด้วย ซึ่งถ้าใช้ความเร็วในการส่งสูง (1Mbps) จะทำให้รัศมีการรับส่งข้อมูลได้ระยะทางสั้นลง แต่ถ้าใช้ความเร็วในการรับส่งข้อมูลที่ช้าลง (250Mbps) จะทำให้ได้รัศมีการรับส่งไกลขึ้น โดยค่า RF Data Rate สามารถกำหนดได้ 2 ค่า คือ

- 250 Kbps
- 1 Mbps

- **RF Operation Mode** ใช้สำหรับกำหนดโหมดการทำงานของ ET-RF24G V1.0 ซึ่งสามารถกำหนดหน้าที่การทำงานได้ 3 แบบ ด้วยกันคือ

- RF Receive Only เป็นการกำหนดให้ ET-RF24G V1.0 ทำหน้าที่เป็นฝ่ายรับข้อมูลทางด้าน RF เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 และส่งออกไปทางด้านขา TX ของ RS232 ตลอดเวลา
- RF Transmit Only เป็นการกำหนดให้ ET-RF24G V1.0 ทำหน้าที่เป็นฝ่ายรับข้อมูลทางด้าน RS232 จากขา RX เพื่อเปลี่ยนเป็นข้อมูลแบบ GFSK และส่งออกไปทางด้าน RF ตลอดเวลา
- RF Auto Direction เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ โดยในโหมดการทำงานนี้เครื่อง ET-RF24G V1.0 จะรอตรวจสอบข้อมูลทั้งจากด้าน RS232 และด้าน RF อยู่ตลอดเวลา โดยถ้าได้รับข้อมูลจากด้าน RS232 ก็จะทำการแปลงแล้วส่งออกทางด้าน RF จากนั้นก็จะกำหนดให้ด้าน RF กลับมาเป็นฝ่ายรับข้อมูลตามเดิม และเมื่อได้รับข้อมูลจากด้าน RF ก็จะแปลงเป็นข้อมูลแล้วส่งออกไปทางด้าน RS232 โดยอัตโนมัติ

- **RF Power Gain** เป็นการกำหนดกำลังส่งของวงจร RF Power ที่ใช้ในการส่งข้อมูล โดยค่า +0dBm เป็นค่ากำลังส่งสูงสุด ส่วน -20dBm เป็นค่ากำลังส่งต่ำสุด โดยสามารถกำหนดได้ 4 ระดับคือ
  - -20dBm (กำลังส่งต่ำสุด)
  - -10dBm
  - -5dBm
  - +0dBm (กำลังส่งสูงสุด)
- **RXD ID Code** เป็นรหัส ID Code ของเครื่อง ET-RF24G V1.0 ในโหมดของการรับข้อมูลจาก RF โดยเมื่อเครื่อง ET-RF24G V1.0 ด้านส่งจะทำการส่งข้อมูลออกไปทาง RF นั้นจะมีการระบุหมายเลข ID Code ของด้านรับรวมไปกับชุดข้อมูลด้วยเสมอ โดยเมื่อเครื่อง ET-RF24G V1.0 ที่อยู่ทางด้านรับทำการรับข้อมูลจากด้าน RF ได้ อันดับแรกมันจะทำการเปรียบเทียบรหัส ID Code ที่รวมมากับข้อมูลที่ได้รับมาได้ว่าตรงกับรหัสของ RXD ID Code ที่กำหนดไว้ไปในตัวมันหรือไม่ ซึ่งถ้าถูกต้องก็จะแยกเอาเฉพาะส่วนของข้อมูลที่ได้รับเข้ามาได้เพื่อนเปลี่ยนแปลงเป็นข้อมูลแบบ RS232 แล้วส่งออกไปทางด้าน TX ของ RS232 แต่ถ้ารหัส ID Code ที่รับมาได้ไม่ตรงกับรหัส RXD ID Code ที่กำหนดไว้เครื่อง ET-RF24G V1.0 จะทิ้งข้อมูลชุดนั้นไปทันที โดยค่า RXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)
- **TXD ID Code** เป็นรหัส ID Code ปลายทางที่จะส่งข้อมูลไปหา โดยที่เครื่อง ET-RF24G V1.0 ที่ถูกกำหนดให้ทำหน้าที่เป็นฝ่ายส่งข้อมูลนั้น เมื่อมันสามารถรับข้อมูลจาก RS232 ได้แล้ว มันจะทำการนำเอาข้อมูลนั้นไปเข้ารหัสรวมกับ TXD ID Code ที่กำหนดไว้ แล้วส่งออกไปทางด้าน RF โดยรหัสของ TXD ID Code นี้หมายถึง รหัส RXD ID Code ของฝ่ายรับที่ต้องการส่งข้อมูลไปหาตัวเอง โดยค่า TXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก(00H-FFH)
- **RF Frequency Channel** เป็นการกำหนดค่าของช่องความถี่ที่จะใช้ในการรับส่งข้อมูลกัน โดยสามารถเลือกกำหนดช่องความถี่ได้สูงสุดมากถึง 125 ช่อง (0-124) โดยการที่เครื่อง ET-RF24G V1.0 จะทำการรับส่งข้อมูลกันได้นั้นจะต้องกำหนดช่องความถี่ที่ตรงกัน และใช้อัตราความเร็ว RF Data Rate ที่เท่ากันด้วย ซึ่งที่สามารถเลือกกำหนดช่องความถี่ RF Frequency



Channel ได้นั้น จะมีประโยชน์เป็นอย่างมากในกรณีที่มีการใช้งานเครื่อง ET-RF24G V1.0 จำนวนหลายๆกลุ่ม ในบริเวณพื้นที่ใกล้เคียงกัน โดยให้กำหนดช่องความถี่ของ ET-RF24G V1.0 กลุ่มที่จะสื่อสารข้อมูลร่วมกันไว้ที่ช่องความถี่เดียวกัน ส่วนกลุ่มอื่นๆ ก็ให้เลือกกำหนดช่องความถี่ที่แตกต่างกันออกไป เพื่อลดปัญหาการรบกวนกัน

## 2.7 ข้อแนะนำในการกำหนดค่า Configuration

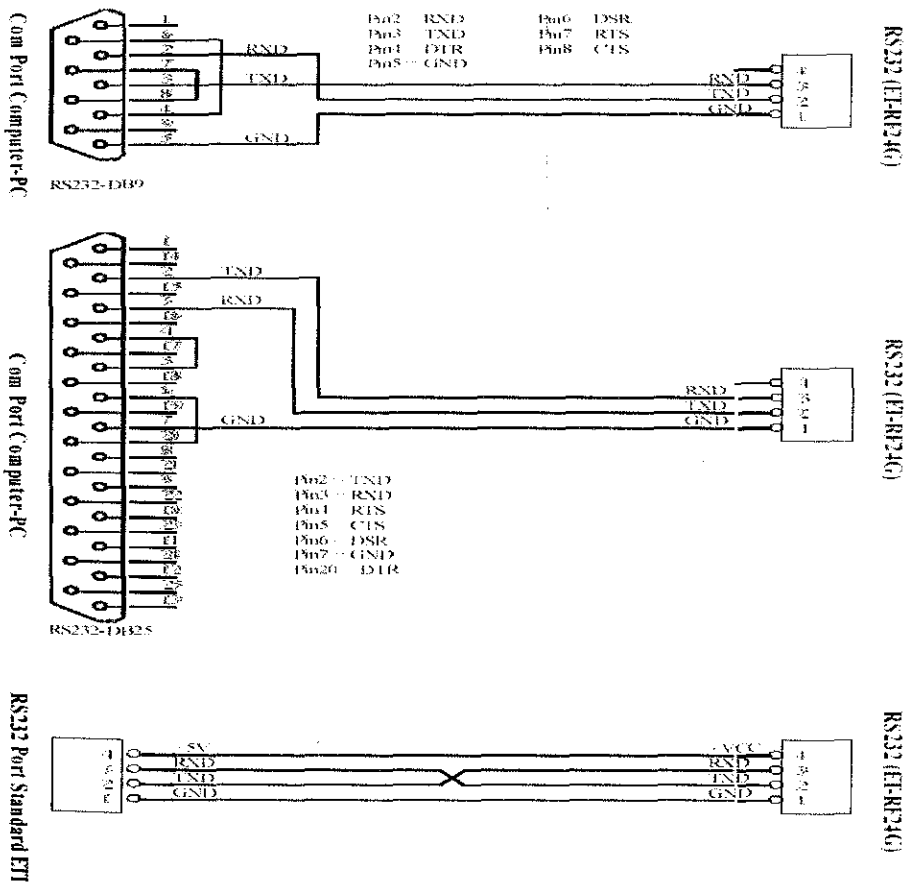
การกำหนดค่า Configuration ให้กับเครื่อง ET-RF24G V1.0 นั้น สามารถเลือกกำหนดได้ตามความต้องการและจุดประสงค์ของการใช้งาน โดยแต่ละโหมดของการใช้งานนั้นจะมีค่า Configuration ที่เหมาะสมต่างกันซึ่งขอแนะนำวิธีการกำหนดค่า Configuration ดังแนวทางต่อไปนี้

- ความเร็วในการรับส่งข้อมูลด้าน RS232 หรือ User RS232 Baud rate ที่ความเร็ว 19200 Bps นั้นเหมาะสมกับการใช้งาน ET-RF24G V1.0 แบบ Receive Only หรือ Transmit Only ซึ่งมีการตรวจสอบความพร้อมของสัญญาณในการรับส่งข้อมูลกันด้วย แต่ถ้าต้องการใช้งานเครื่อง ET-RF24G V1.0 ในโหมด Auto Direction นั้น ควรกำหนดค่า User RS232 Baud rate ไว้ที่ความเร็วไม่เกิน 9600 Bps จะดีที่สุด และควรกำหนดค่า Baud rate ของทั้งสองฝ่ายให้มีค่าเท่ากันด้วย
- ค่าความเร็วของการรับส่งข้อมูลด้าน RF หรือ RF Data Rate ที่สามารถรับส่งข้อมูลกันได้ระยะทางไกลมากที่สุด และมีโอกาสผิดพลาดน้อยที่สุด คือ 250Kbps
- ค่า RF Power Gain ที่ดีที่สุดคือ 0dBm ซึ่งเป็นค่ากำลังส่งสูงสุด ซึ่งจะทำให้สามารถส่งข้อมูลได้ระยะทางไกลที่สุด แต่ถ้าระยะการรับส่งข้อมูลไม่ไกลกันมาก และมีการใช้งานเครื่อง ET-RF24G V1.0 จำนวนหลายๆ กลุ่มในพื้นที่ใกล้เคียงกัน ก็อาจทำการลดกำลังส่งให้ต่ำลงเพื่อลดปัญหาการรบกวนกันหรือกำหนดช่องความถี่ RF Frequency Channel ให้ห่างกันมากๆ
- ในกรณีที่มีการใช้เครื่อง ET-RF24G V1.0 หลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ควรกำหนดช่องความถี่ในการใช้งาน หรือ RF Frequency Channel ให้ห่างกันด้วยเพื่อป้องกันการรบกวนกัน
- การใช้งานเครื่อง ET-RF24G V1.0 แบบ Auto Direction นั้น ถ้ามีการส่งข้อมูลจำนวนมากๆ ควรจัดแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดข้อมูลชุดละไม่เกิน 64 Byte โดยในการส่งข้อมูลแต่ละชุดนั้นให้ทำการส่งข้อมูลอย่างต่อเนื่องโดยให้ข้อมูลแต่ละ Byte มีระยะเวลาห่างกันไม่เกิน 2.5 mS เนื่องจากถ้าข้อมูลขาดหายไปนานกว่านี้ เครื่อง ET-RF24G V1.0 จะทำการเปลี่ยนโหมดของการส่งข้อมูลกลับเป็นโหมดของการรับข้อมูลแทน ซึ่งเมื่อมีการส่งข้อมูล Byte ถัดไปมาอีกก็จะต้องเสียเวลาในการ

สลับโหมดจากฝ่ายรับข้อมูลให้เป็นฝ่ายส่งข้อมูลอีก ซึ่งจะทำให้ประสิทธิภาพในการจัดส่งข้อมูลลดลงเนื่องจากต้องเสียเวลาในการสลับโหมดการทำงานของวงจรรภาค RF อยู่ตลอดเวลา โดยที่เมื่อทำการจัดส่งข้อมูลครบ 64 Byte แล้วให้ทำการหน่วงเวลาไว้ชั่วขณะหนึ่ง ประมาณ 1mS-2mS แล้วจึงส่งข้อมูลชุดถัดไปอีกอย่างนี้เรื่อยๆ จะทำให้การรับส่งข้อมูลมีประสิทธิภาพสูงสุด

- การใช้งานเรื่อง ET-RF24G V1.0 แบบ Auto Direction นั้น ควรหน่วงเวลาในการสลับโหมดจากฝ่ายของการรับข้อมูลเป็นฝ่ายส่งข้อมูล อย่างน้อยที่สุด 3mS-5mS ซึ่งถ้าส่งข้อมูลย้อนกลับด้วยเวลาที่เร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้

การเชื่อมต่อสัญญาณ RS232

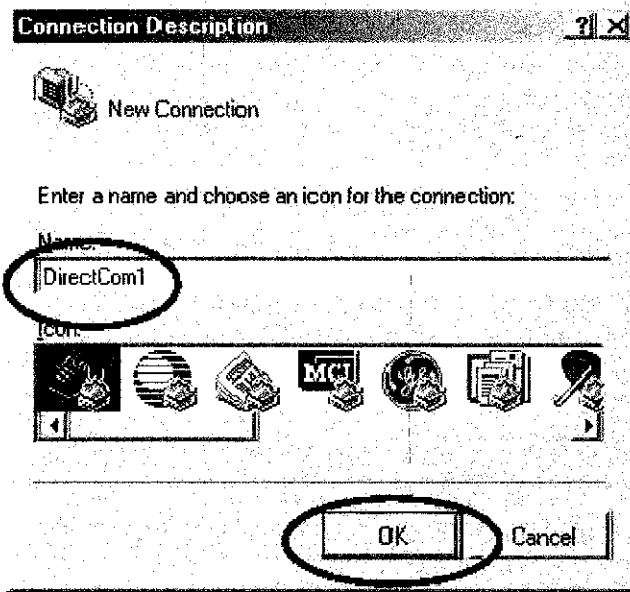


รูปที่ 2.10 แสดงแผนผังการต่อสาย RS232 เพื่อใช้งานกับ ET-RF24G V1.0 ในโหมด Auto Direction

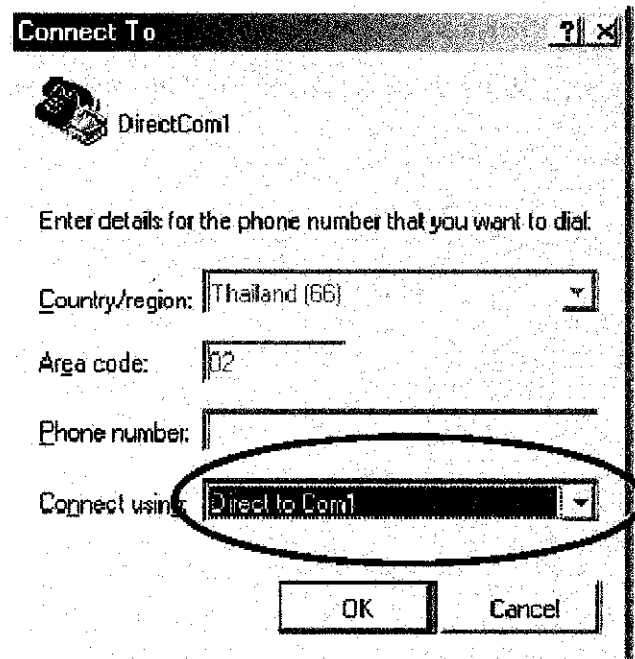
## ตัวอย่างการใช้งาน

สำหรับตัวอย่างการใช้นั้น จะขอแสดงให้เห็นโดยใช้คอมพิวเตอร์ PC เป็นอุปกรณ์การทดลอง โดยในที่นี้จะขอเลือกใช้โปรแกรมสำเร็จรูปสำหรับการใช้ในการสื่อสารของ Windows ซึ่งก็คือ Hyper Terminal ซึ่งมีวิธีการใช้งานดังต่อไปนี้

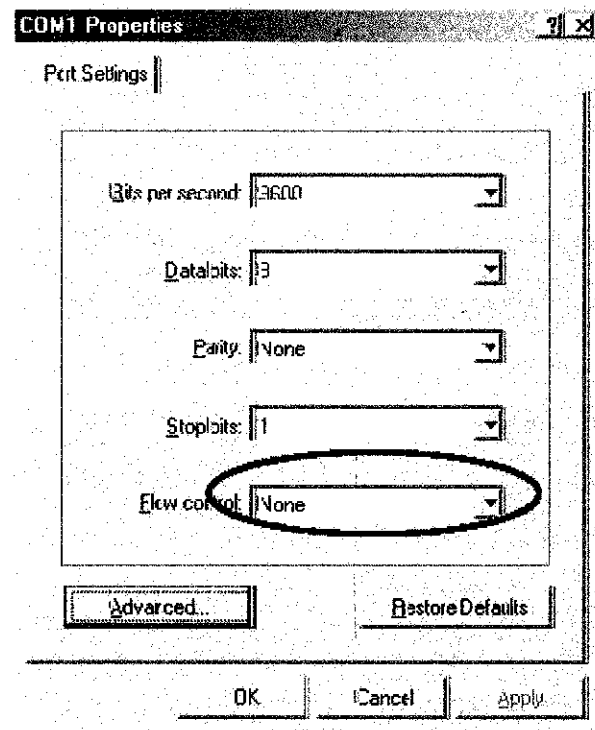
1. เรียกใช้โปรแกรม Hyper Terminal ของ Windows โดยเรียกจาก Star → Programs → Accessories → Communications → Hyper Terminal ซึ่งจะได้ผลดังรูป



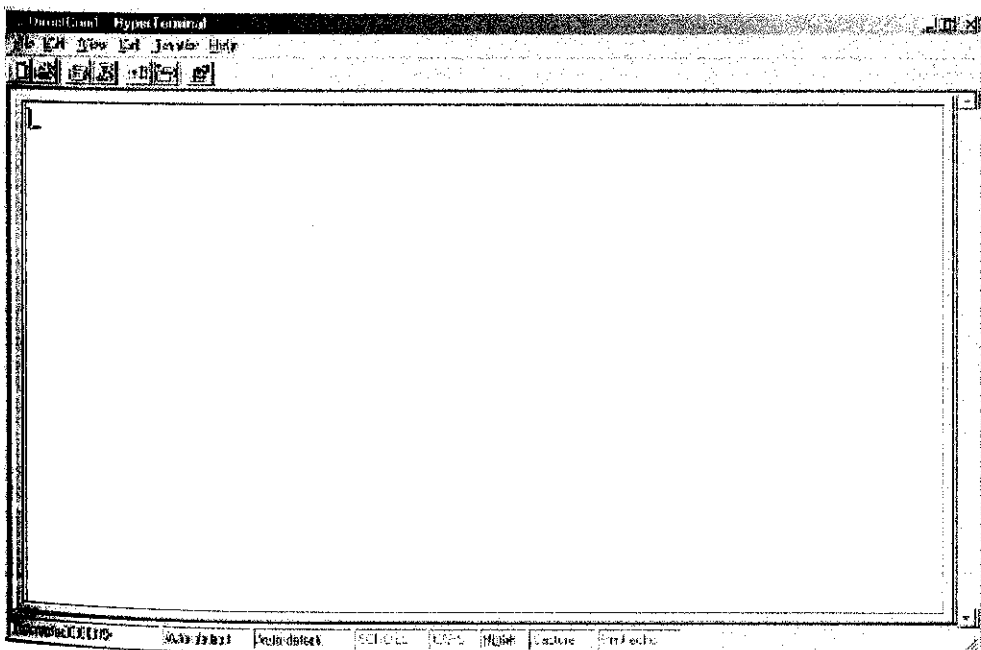
2. ให้เลือกกำหนดชื่อสำหรับการเชื่อมต่อ ซึ่งสามารถกำหนดได้เองตามต้องการ โดยในตัวอย่างจะกำหนดเป็น DirectCom1 จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป



3. ให้เลือกกำหนดการเชื่อมต่อเป็น Direct to Com1 ซึ่งถ้าเครื่องคอมพิวเตอร์ที่ใช้เป็น Com-port อื่นที่ไม่ใช่ Com1 ก็ให้เลือกให้ตรงกับความจริง จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป



4. ในขั้นตอนนี้ จะใช้สำหรับกำหนดคุณสมบัติของพอร์ตอนุกรม RS232 โดยให้เลือก Bit per second = 9600, Data Bit = 8, Parity = None, Stop Bit = 1 ส่วน Flow Control ให้เลือกเป็น None จากนั้นเลือก OK ซึ่งจะเข้าสู่หน้าต่างโปรแกรมหลักของ Hyper Terminal ดังรูป



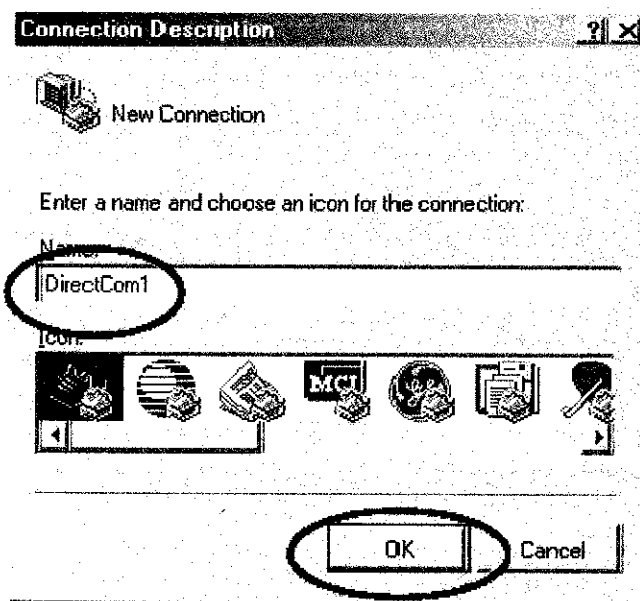
### ข้อสังเกตในการกำหนด Configuration

- ค่า RF Frequency Channel ต้องกำหนดให้ตรงกันทั้ง 2 ตัว
- ค่า RF Data Rate ต้องกำหนดให้ตรงกันทั้ง 2 ตัว
- ค่า RXD ID Code ของตัวที่ 1 ต้องตรงกับ TXD ID Code ของตัวที่ 2
- ค่า TXD ID Code ของตัวที่ 1 ต้องตรงกับ RXD ID Code ของตัวที่ 2

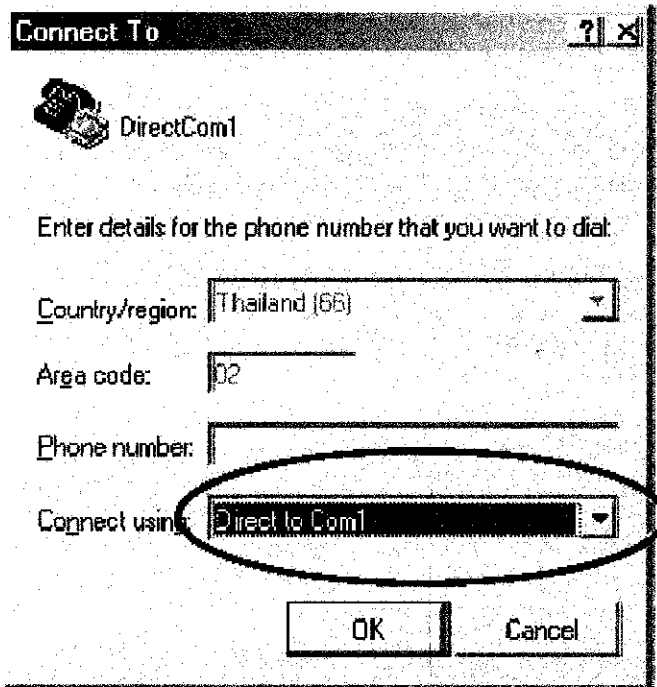
สำหรับการทดสอบการทำงานด้วย Hyper Terminal นั้นให้ทดลองกดคีย์ใดๆ ในขณะที่ Run โปรแกรม Hyper Terminal อยู่ โดยจะสังเกตเห็นตัวอักษรจากแป้นพิมพ์ของฝ่ายส่งข้อมูล จะถูกส่งออกไปแสดงผลที่หน้าจอโปรแกรม Hyper Terminal ของอีกฝ่ายหนึ่งในทันที

สำหรับการทดสอบการใช้งาน ตามตัวอย่างนี้ สามารถเรียกใช้โปรแกรมสำเร็จรูปที่จัดการเรื่องการสื่อสารอนุกรมของ Windows ซึ่งก็คือ Hyper Terminal ได้ทันที โดยในการใช้งานนั้นสามารถกระทำได้ดังขั้นตอนต่อไปนี้

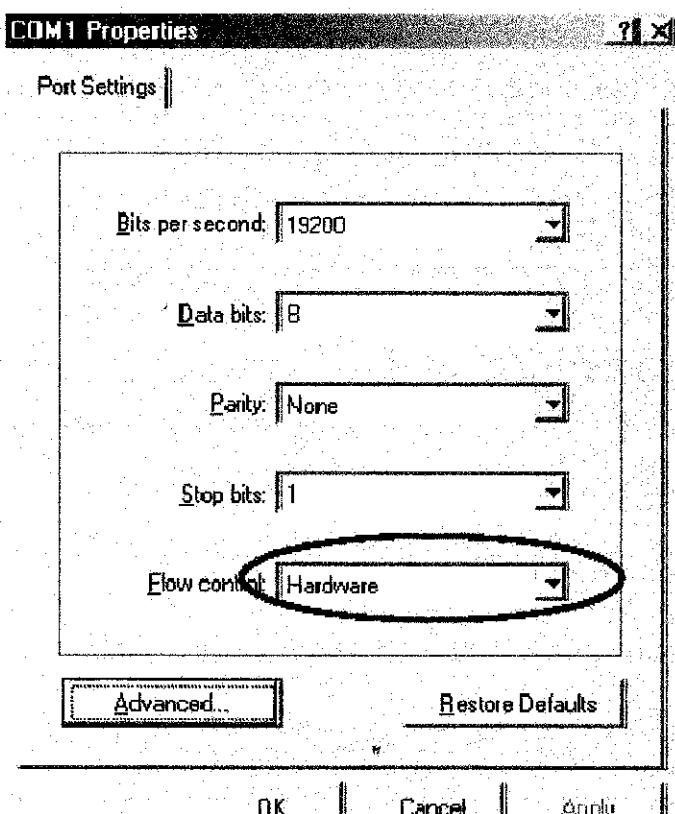
1. เรียกใช้โปรแกรม Hyper Terminal ของ Windows โดยเรียกจาก Star → Programs → Accessories → Communications → Hyper Terminal ซึ่งจะได้ผลดังรูป



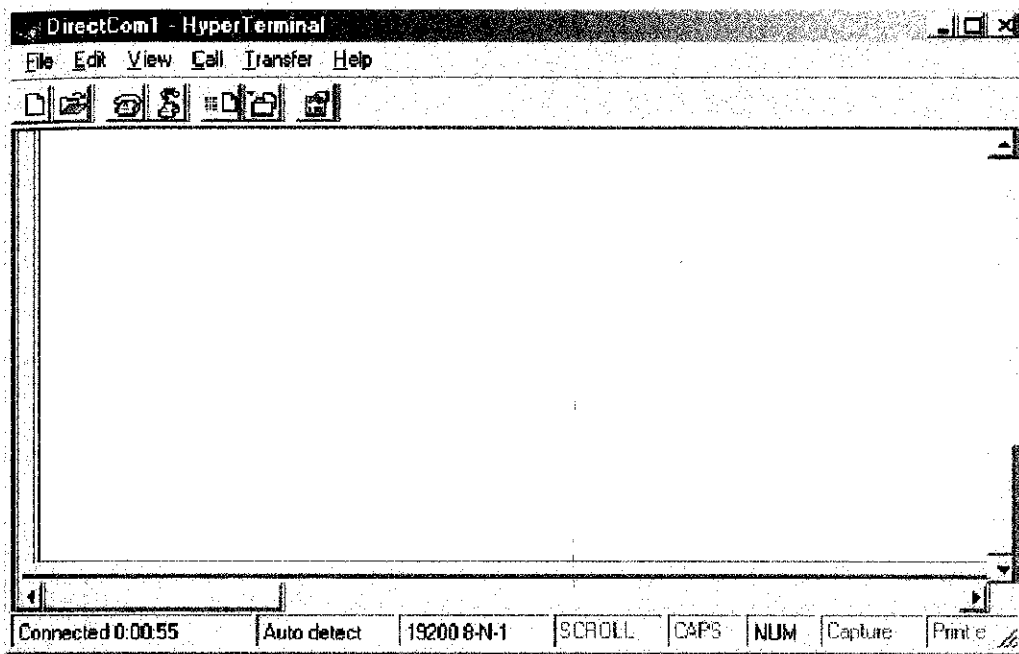
2. ให้เลือกกำหนดชื่อสำหรับใช้ในการเชื่อมต่อ ซึ่งสามารถกำหนดได้เองตามต้องการ โดยในตัวอย่างจะกำหนดเป็น DirectCom1 จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป



3. ให้เลือกกำหนดการเชื่อมต่อเป็น Direct to Com1 ซึ่งถ้าเครื่องคอมพิวเตอร์ที่ใช้เป็น Comport อื่นที่ไม่ใช่ Com1 ก็ให้เลือกให้ตรงกับความจริง จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป

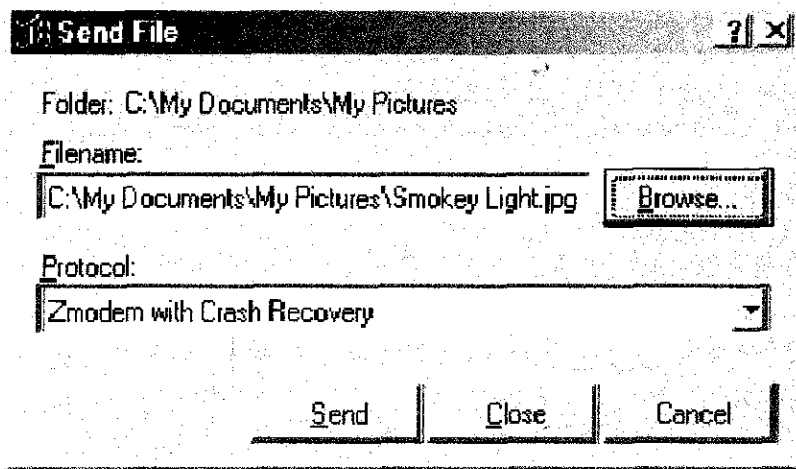


4. ในขั้นตอนนี้จะใช้สำหรับกำหนดคุณสมบัติของพอร์ตอนุกรม RS232 โดยให้เลือก Bit per second = 9600, Data Bit = 8, Parity = None, Stop Bit = 1 ส่วน Flow Control ให้เลือกเป็น None จากนั้นเลือก OK ซึ่งจะเข้าสู่หน้าต่างโปรแกรมหลักของ Hyper Terminal ดังรูป

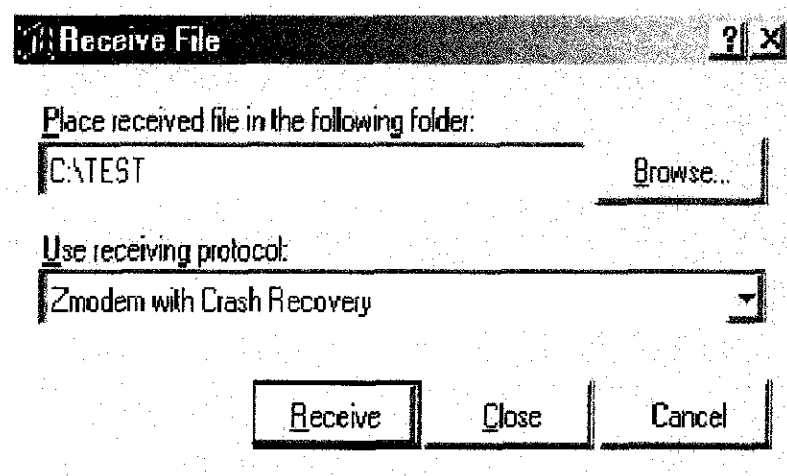


5. ในขั้นตอนนี้สามารถทำการรับส่งข้อมูลระหว่างทั้ง 2 ฝ่ายได้แล้ว ซึ่งสามารถทดสอบได้โดยการกดคีย์ใดๆ จากฝ่ายหนึ่ง ซึ่งตัวอักขระคีย์นั้นๆ จะถูกส่งไปแสดงผลยังฝั่งตรงข้ามทันที และในที่นี้เราจะทำการทดสอบการรับและส่งไฟล์ โดยใช้ Protocol สำเร็จรูปของ Hyper Terminal ซึ่งมีให้เลือกใช้มากมายหลาย Protocol โดยต้องกำหนด Protocol ให้ตรงกันทั้งฝ่ายส่งและฝ่ายรับ ซึ่งในขั้นตอนของการทดสอบนั้นต้องกำหนดให้ฝ่ายหนึ่งเป็นฝ่ายรับและให้อีกฝ่ายหนึ่งเป็นฝ่ายส่ง ซึ่งในที่นี้จะขอแนะนำให้ทดสอบโดยเลือกใช้ Protocol ของ Zmodem with Crash Recovery ซึ่งมีวิธีการทดสอบการรับส่งข้อมูลดังนี้

ทางด้านฝ่ายส่งให้ทำการเลือกกำหนดไฟล์ที่จะส่งจากเมนูคำสั่ง Transfer → Send File จากนั้นให้เลือกกำหนดชื่อและที่อยู่ของไฟล์ที่ต้องการจะส่ง โดยคลิกเมาส์ที่ปุ่ม Browse พร้อมกับกำหนดชื่อและที่อยู่ของไฟล์ตามต้องการ จากนั้นให้เลือกกำหนด Protocol ของการรับส่งข้อมูลเป็น Zmodem with Crash Recovery แล้วเลือกคลิกกดเมาส์ที่ปุ่ม Send เพื่อทำการเริ่มต้นส่งข้อมูลดังรูป

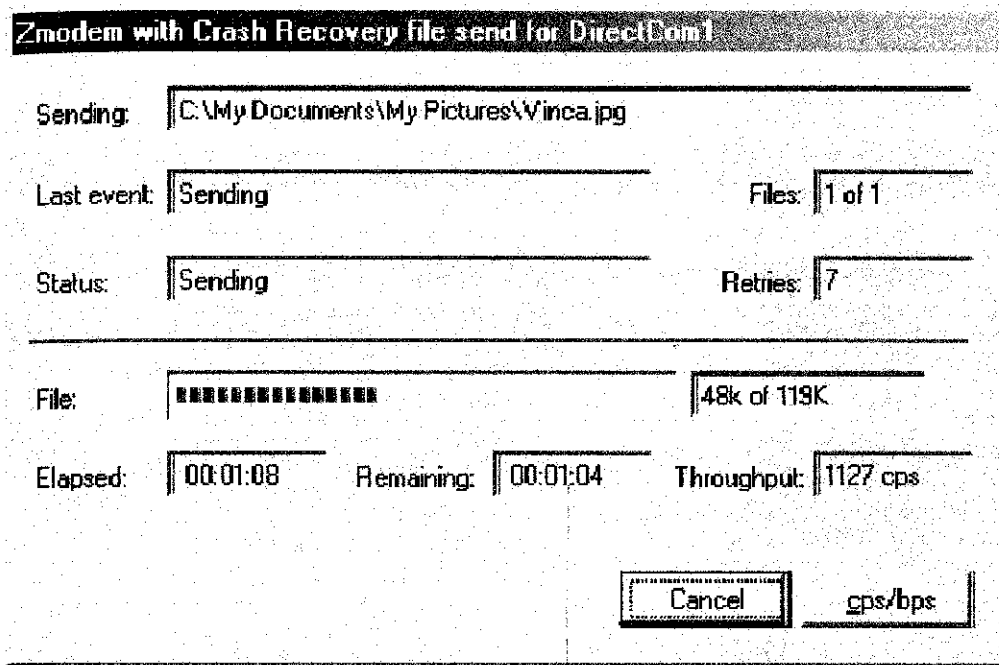


สำหรับในด้านที่เป็นฝ่ายรับข้อมูลนั้นก็ให้เลือกกำหนดการทำงานให้เป็นฝ่ายรับ โดยกำหนดจากเมนูคำสั่งของ Transfer → Receive File จากนั้นให้เลือกกำหนดตำแหน่งของ Folder สำหรับใช้บันทึกไฟล์ที่รับได้จากฝ่ายส่ง โดยการเลือกจากปุ่ม Browse แล้วเลือกกำหนด Folder ที่ต้องการ ส่วนชื่อนั้นไม่ต้องกำหนด โดยโปรแกรม Hyper Terminal จะตั้งให้เองตามชื่อไฟล์จริงที่ส่งมา และในส่วนของ Protocol ที่ใช้นั้นก็ต้องกำหนดให้ตรงกับทางด้านส่ง คือ Zmodem with Crash Recovery จากนั้นให้เลือก Receive เพื่อให้โปรแกรมรองรับไฟล์จากด้านส่ง



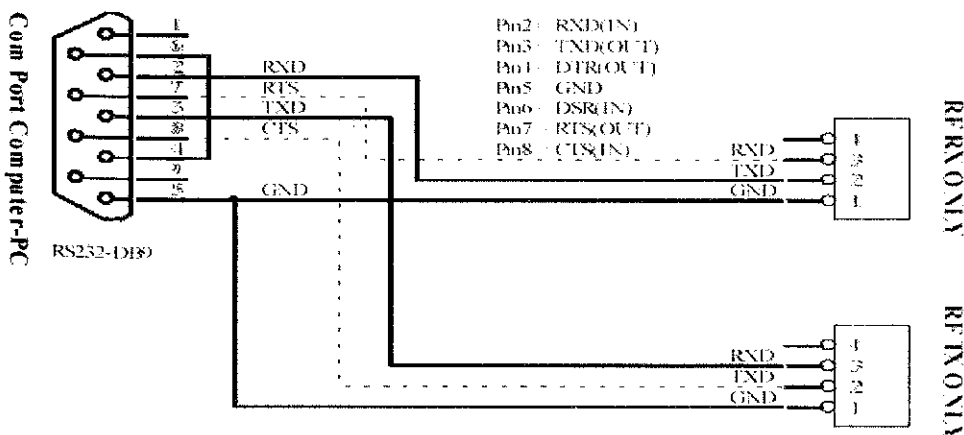


โดยในขณะที่มีการรับส่งข้อมูลกันอยู่นั้น โปรแกรม Hyper Terminal ทั้ง 2 ด้านจะแสดงสถานะการทำงานให้ทราบอยู่ตลอดเวลา ดังรูป



โดยให้รอก่อนว่าการทำงานจะเสร็จสมบูรณ์ ซึ่งหน้าต่างที่แสดงสถานะการทำงานของโปรแกรมจะถูกปิดไปเองโดยอัตโนมัติหลังจากทำการรับส่งข้อมูลกันเสร็จเรียบร้อยแล้ว

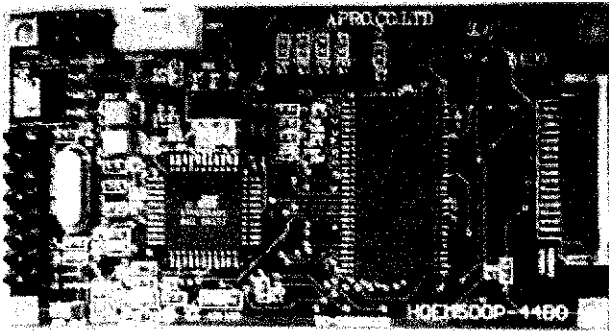
โดยในการทดสอบการทำงานของโปรแกรมตามตัวอย่างนี้ จะต้องกำหนดรูปแบบการสื่อสารของ RS232 ให้มีการตรวจสอบความพร้อมในการรับส่งข้อมูลกันด้วยสัญญาณทาง Hardware ด้วย โดยเลือกกำหนดรูปแบบการสื่อสารของ RS232 ในหัวข้อ Flow Control เป็น Hardware พร้อมกับต่อสายสัญญาณดังวงจรต่อไปนี้



รูปที่ 2.11 แสดงวงจรของสายที่ใช้สำหรับทดสอบการรับส่งข้อมูลแบบ Full Duplex

## 2.8 BioPrima OEM2000P-44b0 Fingerprint Identification Board

1. เครื่อง OEM2000P-44b0 ประกอบด้วย ตัวประมวลผลลายนิ้วมือ และ ตัวรับ input



รูปที่ 2.12 OEM2000P-44b0 module



รูปที่ 2.13 Optical Sensor

2. ลักษณะของเครื่อง OEM2000P-44b0

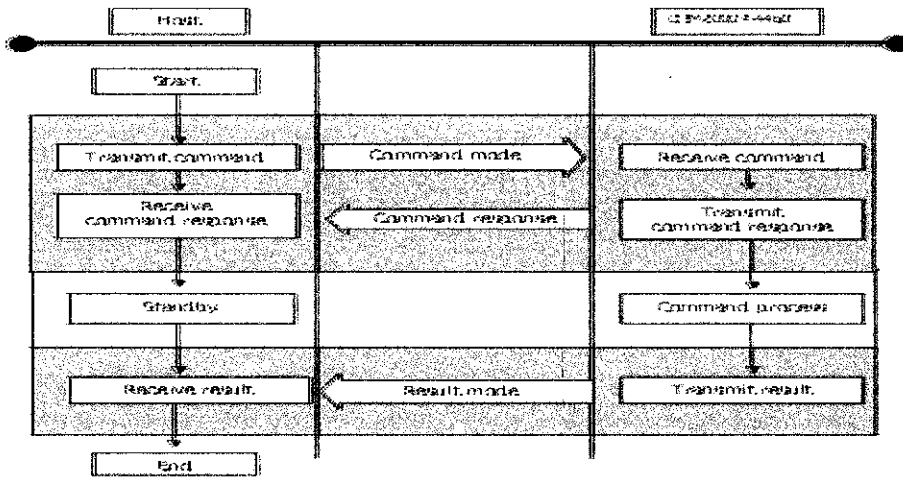
Item	Feature	
Mode of combination(O/F)	RS232C ( 3 line ) Baudrate=9600/19200/38400bps/115200bps Parity=NONE StopBit=1 FlowControl=NONE	
Size of fingerprint enroll data	1404 Byte	
Max. fingerprint capacity	2,000 fingerprints	
Matching mode	1 : N / 1 : 1	
Enrolling mode	Thrice	
Pass Rate value	1 ~ 5	
FAR	< 0.0001% ( in case pass rate is 3 )	
PAR	< 0.01% ( in case pass rate is 3 )	
Enrollment speed	< 3 sec	
Verification speed (500 prints)	< 1 sec	
Operating voltage	DC 5V	
Operating current	200 mA	
Operating Environment	Temperature	0°C ~ +60°C
	Humidity	20% ~ 80%
Storage Environment	Temperature	0°C ~ +30°C
	Humidity	10% ~ 80%
Size of board	76 x 41 x 12 mm ( W x H x D )	

3. Communication Protocol

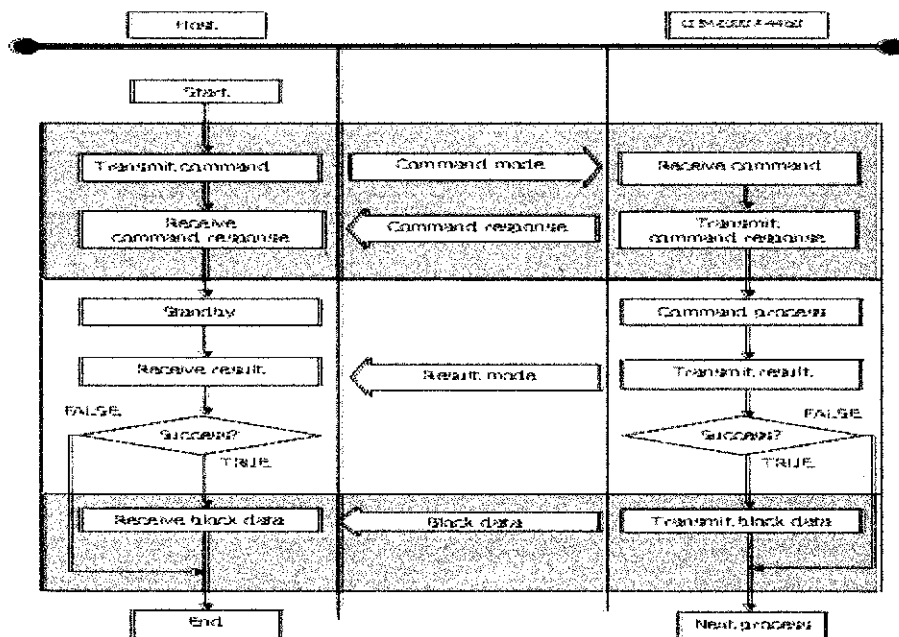
แบ่งออกเป็น 3 โหมดคือ

- 1.) โหมดคำสั่ง
- 2.) โหมดรับ
- 3.) โหมดส่ง

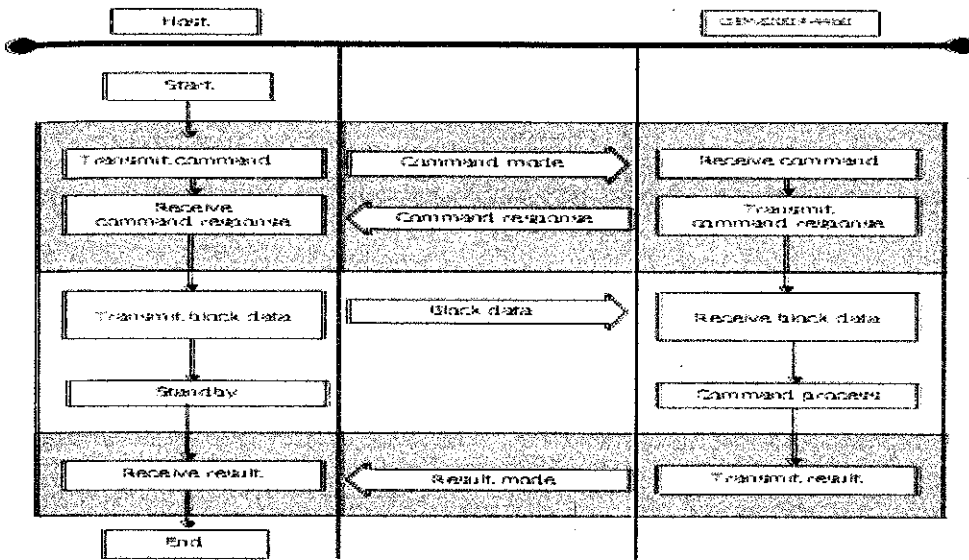
โหมดคำสั่ง



โหมดรับ



โหมดส่ง



### บทที่ 3

#### การทดสอบโปรแกรม

#### 3.1 การเก็บลายนิ้วมือ

เริ่มต้นเก็บลายนิ้วมือของกลุ่มนักศึกษาตัวอย่างจำนวน 30 คน โดยการเก็บทีละคนเรียงตามลำดับรหัสประจำตัวนักศึกษา จัดเก็บข้อมูลไว้ในโปรแกรมไมโครซอฟต์ แอคเซส (Microsoft Access) ได้ผลดังต่อไปนี้

Contacts			
sID	stuID	First Name	Last Name
1	B4700463	นายคมกฤช	ไสดาศรี
2	B4700739	นางสาวจุฬาลักษณ์	เข็มสว่าง
3	B4700913	นายชัยวัฒน์	แสนบุญศิริ
4	B4701989	นางสาวนุชนาฏ	ฝาเพ็ญ
5	B4702009	นายนุชา	แสงมณี
6	B4702399	นายบุญญวีร์	ชื่อตรง
7	B4702689	นางสาวพิชญา	ชัยปัญญา
8	B4703983	นางสาววิลาสินี	ธนะสำเร็จ
9	B4704270	นางสาวศจี	สิงห์สง่า
10	B4704454	นายศักรินทร์	เขาว์ขุนทด
11	B4704911	นายสิงหนเดช	วชิรเพชรปราวณี
12	B4705055	นางสาวสุจิตรา	โหมดhirัญ
13	B4705581	นางสาวอภิญา	อินทร์นอก
14	B4705888	นายเอกชัย	ชมระกา
15	B4705895	นายเอกราษฎร์	พลแสง
16	B4706212	นายกิตติกร	ศุภเสถียรกุล
17	B4706472	นายโกวิท	ภิรมย์กิจ
18	B4706588	นายคมกฤช	วงษ์แก้ว

Contacts			
sID	stuID	First Name	Last Name
19	B4706953	นางสาวจุฑารัตน์	ชัยมานิตย์
20	B4707301	นายชัยพฤกษ์	พลพวก
21	B4707776	นายदनัย	อัศสานึก
22	B4707936	นายทงศักดิ์	งามเจริญ
23	B4708063	นายทศพล	เตตานัง
24	B4709084	นางสาวปทุมวดี	แก้วดี
25	B4709152	นางสาวประทุมพร	อาโคม
26	B4709558	นางสาวผกามาศ	ขันขุนทด
27	B4710783	นางสาวรวมพร	นฤมลโรจน์
28	B4711971	นายศุภกิจ	ลักขุธ
29	B4712497	นายลำราญ	สันทาลุนัย
30	B4713043	นางสาวสุวรรณา	ประดิษฐรา

### ตารางที่ 3.1 แสดงข้อมูลลายนิ้วมือของนักศึกษากลุ่มตัวอย่าง

จากตารางจะเห็นได้ว่ามีคอลัมน์ทั้งหมด 4 คอลัมน์ คือ

1. sID หมายถึง หมายเลขของแต่ละบุคคลที่ถูกจับคู่กับลายนิ้วมือของคนนั้นๆ
2. stuID หมายถึง รหัสประจำตัวนักศึกษา
3. First Name หมายถึง ชื่อของนักศึกษา
4. Last Name หมายถึง นามสกุลของนักศึกษา

เมื่อทำการจัดเก็บลายนิ้วมือของนักศึกษากลุ่มตัวอย่างครบแล้ว ต่อไปจะเป็นการทดสอบการทำงานของโปรแกรม

### 3.2 ทดสอบการทำงานของโปรแกรม

ทำการสอบโปรแกรมกับนักศึกษาในกลุ่มตัวอย่างจำนวน 30 คน ที่ได้เก็บลายนิ้วมือไว้แล้ว เป็นจำนวน 3 วัน ได้ผลดังนี้

#### สรุปผลการเข้าห้องเรียน

เลขที่	รหัส	ชื่อ	ทั้งหมด	มา	ขาด
1	B4700463	นายคมกฤช ไสดาศรี	3	1	2
2	B4700739	นางสาวจุฬาลักษณ์ เข็มสว่าง	3	3	0
3	B4700913	นายชัยวัฒน์ แสนบุญศิริ	3	3	0
4	B4701989	นางสาวนุชนาฏ ฝาเพ็ยม	3	2	1
5	B4702009	นายนุชา แสงมณี	3	2	1
6	B4702399	นายบุญญวีร์ ชื่อดวง	3	3	0
7	B4702689	นางสาวพิชญา ชัยปัญญา	3	3	0
8	B4703983	นางสาววิลาลินี ธนะสำเร็จ	3	3	0
9	B4704270	นางสาวศจี สิงห์สง่า	3	2	1
10	B4704454	นายศักรินทร์ เซาว์ขุนทด	3	2	1
11	B4704911	นายสิงหนเดช วชิรเพชรปราณี	3	3	0
12	B4704911	นางสาวสุจิตรา ไหมคหิรัญ	3	3	0
13	B4705581	นางสาวอภิญา อินทร์นอก	3	3	0
14	B4705888	นายเอกชัย ชมระกา	3	3	0
15	B4705895	นายเอกนรินทร์ พลแสง	3	3	0
16	B4706212	นายกิตติกร ศุภเสถียรกุล	3	3	0
17	B4706472	นายโกวิท ภิรมย์กิจ	3	3	0
18	B4706588	นายคมกฤช วงษ์แก้ว	3	3	0
19	B4706953	นางสาวจุฑารัตน์ ชัยมานิตย์	3	3	0
20	B4707301	นายชัยพฤกษ์ พลพวก	3	3	0
21	B4707776	นายदनัย อัครานิก	3	1	2
22	B4707936	นายทองศักดิ์ งามเจริญ	3	1	2
23	B4708063	นายทศพล เตตานัง	3	2	1
24	B4709084	นางสาวปทุมวดี แก้วดี	3	1	2
25	B4709152	นางสาวประทุมพร อาโคม	3	-	-

เลขที่	รหัส	ชื่อ	ทั้งหมด	มา	ขาด
26	B4709558	นางสาวผกามาศ ชันขุนทด	3	1	2
27	B4710783	นางสาวรวมพร นุญกุลโรจน์	3	1	2
28	B4711971	นายศุภกิจ ลักษวุธ	3	1	2
29	B4712497	นายสำราญ สันทาลุนัย	3	1	2
30	B4713043	นางสาวสุพรรณภา ประดิษฐุรา	3	1	2

ตารางที่ 3.2 แสดงผลการเข้าห้องเรียน



## รายงานแสดงเวลาเข้า

Ident			
id	sID	idDate	idTime
1	11	18/3/2551	19:27:15
2	16	18/3/2551	19:27:15
3	18	18/3/2551	19:27:15
4	19	18/3/2551	19:27:15
5	6	18/3/2551	19:27:15
6	4	18/3/2551	19:27:15
7	5	18/3/2551	19:27:15
8	23	18/3/2551	19:27:15
9	27	18/3/2551	19:27:15
10	9	18/3/2551	19:30:09
11	1	18/3/2551	19:30:09
12	21	18/3/2551	19:30:09
13	26	18/3/2551	19:30:09
14	22	18/3/2551	19:30:09
15	23	18/3/2551	19:30:09
16	28	18/3/2551	19:30:09
17	29	18/3/2551	19:30:09
18	17	18/3/2551	19:33:17
19	8	18/3/2551	19:33:17
20	20	18/3/2551	19:33:17
21	15	18/3/2551	19:33:17
22	14	18/3/2551	19:33:17
23	13	18/3/2551	19:33:17
24	12	18/3/2551	19:33:17

Ident			
id	sID	idDate	idTime
25	7	18/3/2551	19:33:17
26	10	18/3/2551	19:33:17
27	2	18/3/2551	19:33:17
28	3	18/3/2551	19:33:17
29	2	19/3/2551	11:22:51
30	3	19/3/2551	11:22:51
31	6	19/3/2551	11:22:51
32	7	19/3/2551	11:22:51
33	8	19/3/2551	11:22:51
34	11	19/3/2551	11:22:51
35	12	19/3/2551	11:22:51
36	13	19/3/2551	11:22:51
37	14	19/3/2551	11:22:51
38	15	19/3/2551	11:22:51
39	16	19/3/2551	11:22:51
40	17	19/3/2551	11:22:51
41	18	19/3/2551	11:22:51
42	19	19/3/2551	11:22:51
43	20	19/3/2551	11:22:51
44	12	19/3/2551	12:52:38
45	11	20/3/2551	11:22:14
46	13	20/3/2551	11:22:14
47	14	20/3/2551	11:22:14
48	15	20/3/2551	11:22:14
49	16	20/3/2551	11:22:14

Ident			
id	sID	idDate	idTime
50	17	20/3/2551	11:22:14
51	18	20/3/2551	11:22:14
52	19	20/3/2551	11:22:14
53	20	20/3/2551	11:22:14
54	2	20/3/2551	12:44:47
55	3	20/3/2551	12:44:47
56	4	20/3/2551	12:44:47
57	5	20/3/2551	12:44:47
58	6	20/3/2551	12:44:47
59	7	20/3/2551	12:44:47
60	8	20/3/2551	12:44:47
61	9	20/3/2551	12:44:47
62	10	20/3/2551	12:44:47
63	30	20/3/2551	12:44:47
64	24	20/3/2551	13:15:47
65	24	20/3/2551	13:16:04

ตารางที่ 3.2 แสดงเวลาเข้าแบบยังไม่ได้เรียง sID

จากตารางจะเห็นว่ามียอดคอลลัมน์ทั้งหมด 4 คอลลัมน์

1. id แสดงถึง ลำดับของหมายเลข sID
2. sID แสดงถึง หมายเลขของแต่ละบุคคลที่ถูกจับคู่กับลายนิ้วมือของคนนั้นๆ
3. idDate แสดงถึง วันที่นักศึกษากรู่มตัวอย่างเข้าเรียน
4. idTime แสดงถึง เวลานั้นนักศึกษากรู่มตัวอย่างเข้าเรียน

## รายงานแสดงเวลาเข้า

เรียงตาม sID

Ident			
id	sID	idDate	idTime
1	1	18/3/2551	19:30:09
2	2	18/3/2551	19:33:17
3	2	19/3/2551	11:22:51
4	2	20/3/2551	12:44:47
5	3	18/3/2551	19:33:17
6	3	19/3/2551	11:22:51
7	3	20/3/2551	12:44:47
8	4	18/3/2551	19:27:15
9	4	20/3/2551	12:44:47
10	5	18/3/2551	19:27:15
11	5	20/3/2551	12:44:47
12	6	18/3/2551	19:27:15
13	6	19/3/2551	11:22:51
14	6	20/3/2551	12:44:47
15	7	18/3/2551	19:33:17
16	7	19/3/2551	11:22:51
17	7	20/3/2551	12:44:47
18	8	18/3/2551	19:33:17
19	8	19/3/2551	11:22:51
20	8	20/3/2551	12:44:47
21	9	18/3/2551	19:30:09
22	9	20/3/2551	12:44:47
23	10	18/3/2551	19:33:17
24	10	20/3/2551	12:44:47

Ident			
id	sID	idDate	idTime
25	11	18/3/2551	19:27:15
26	11	19/3/2551	11:22:51
27	12	18/3/2551	19:33:17
28	12	19/3/2551	11:22:51
29	13	18/3/2551	19:33:17
30	13	19/3/2551	11:22:51
31	14	18/3/2551	19:33:17
32	14	19/3/2551	11:22:51
33	14	20/3/2551	11:22:14
34	15	18/3/2551	19:33:17
35	15	19/3/2551	11:22:51
36	15	20/3/2551	11:22:14
37	16	18/3/2551	19:27:15
38	16	19/3/2551	11:22:51
39	16	20/3/2551	11:22:14
40	17	18/3/2551	19:33:17
41	17	19/3/2551	11:22:51
42	17	20/3/2551	11:22:14
43	18	18/3/2551	19:27:15
44	18	19/3/2551	11:22:51
45	18	20/3/2551	11:22:14
46	19	18/3/2551	19:27:15
47	19	19/3/2551	11:22:51
48	19	20/3/2551	11:22:14
49	20	18/3/2551	19:33:17

Ident			
id	siD	idDate	idTime
50	20	19/3/2551	11:22:51
51	20	20/3/2551	11:22:14
52	21	18/3/2551	19:30:09
53	22	18/3/2551	19:30:09
54	23	18/3/2551	19:27:15
55	24	20/3/2551	13:15:47
56	26	18/3/2551	19:30:09
57	27	18/3/2551	19:27:15
58	28	18/3/2551	19:30:09
59	29	18/3/2551	19:30:09
60	30	20/3/2551	12:44:47

ตารางที่ 3.3 แสดงเวลาเข้าแบบเรียง siD

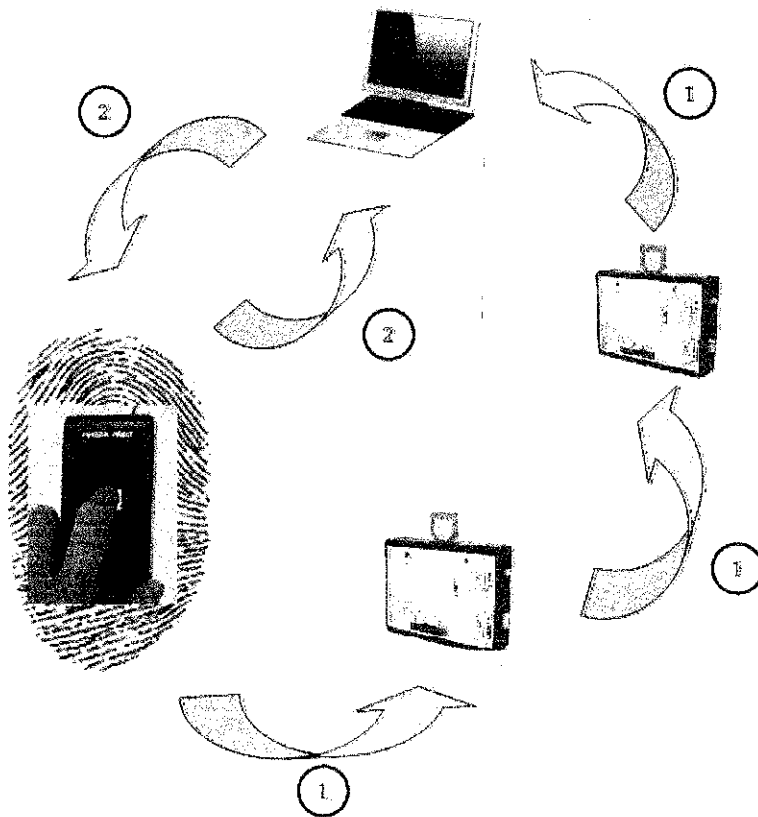
จากตารางจะเห็นว่ามียอดคอลลัมน์ทั้งหมด 4 คอลลัมน์

1. id แสดงถึง ลำดับของหมายเลข siD
2. siD แสดงถึง หมายเลขของแต่ละบุคคลที่ถูกจับคู่กับลายนิ้วมือของคนนั้นๆ
3. idDate แสดงถึง วันที่นักศึกษาในกลุ่มตัวอย่างเข้าเรียน
4. idTime แสดงถึง เวลานั้นนักศึกษาในกลุ่มตัวอย่างเข้าเรียน

## บทที่ 4

### ขั้นตอนการทำงาน

4.1 การทำงานของFINGER PRINT OEM2000P-44b0เมื่อทำงานเชื่อมอุปกรณ์แบบไร้สาย  
FINGER PRINT OEM2000P-44b0 เป็นอุปกรณ์ที่พัฒนาเพื่อใช้เก็บข้อมูลและตรวจสอบลายนิ้วมือเพื่อใช้ในการยืนยันตัวบุคคลและสามารถทำงานร่วมกับ RF Wireless (RF 2.4 GHz) เพื่อส่งข้อมูลการพิมพ์ลายนิ้วมือแบบไร้สาย โดยมีรูปแบบการทำงานซึ่งแสดงด้วยรูปภาพดังนี้

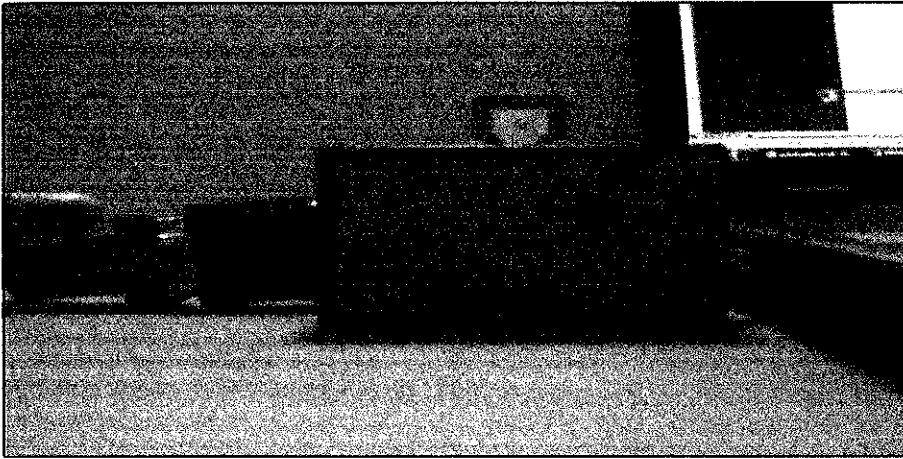


รูปที่ 4.1 แสดงรูปแบบการทำงาน

ขั้นตอนการทำงานจะประกอบด้วย ในส่วนแรกการจัดเก็บลายนิ้วมือลงในเครื่องอ่านลายนิ้วมือตามลูกศรที่ 1 ในแต่ละลายนิ้วมือจะมีการแสดงเอกลักษณ์ (Identification หรือ ID) ลำดับ ดังนั้นจะนำ การแสดงเอกลักษณ์ (Identification หรือ ID) มาบันทึกลงในคอมพิวเตอร์ และส่วนที่สองตามลูกศรที่ 2 แสดงขั้นตอนการแต่นิ้วและประมวลผลเพื่อยืนยันลายนิ้วมือ

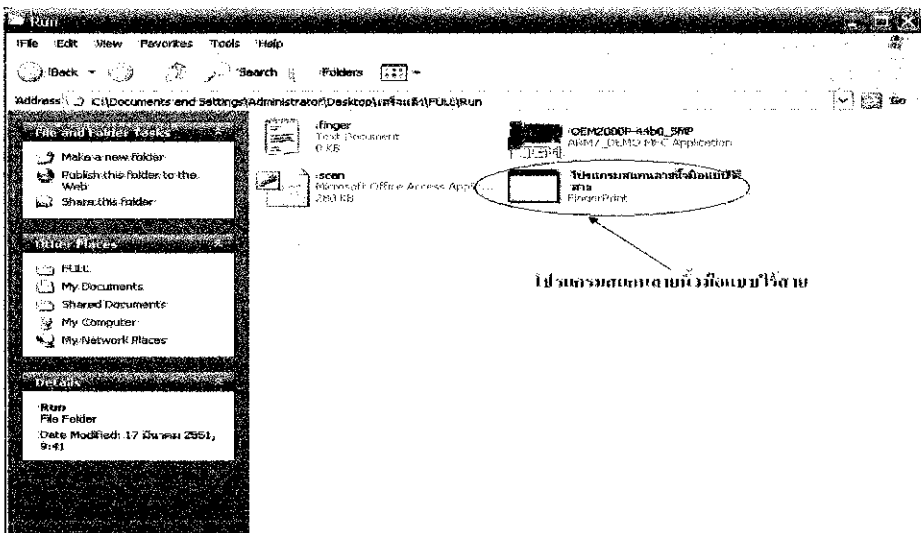
## 4.2 ขั้นตอนการใช้งานโปรแกรมอ่านลายนิ้วมือแบบไร้สาย

- ทำการลงโปรแกรมวิซวล สตูดิโอ 2005 (Visual Studio 2005) และ โปรแกรม ไมโครซอฟต์ แอคเซส (Microsoft Access)
- ก่อนการใช้งานโปรแกรมถ้าเป็น แลปทอป (Laptop) ต้องทำการลงโปรแกรม เพื่อแปลง Universal Serial Bus (USB) port เป็น Serial Port ก่อนเพื่อต่อกับ ตัวรับสัญญาณแบบไร้สาย
- ทำการต่อสาย Universal Serial Bus (USB) port to Serial Port กับ แลปทอป (Laptop) และเชื่อมต่อกับตัวรับสัญญาณแบบไร้สาย



### 4.2.1 เปิดการใช้งาน ตั้งค่าโปรแกรม และ การจัดเก็บลายนิ้วมือ (Enroll)

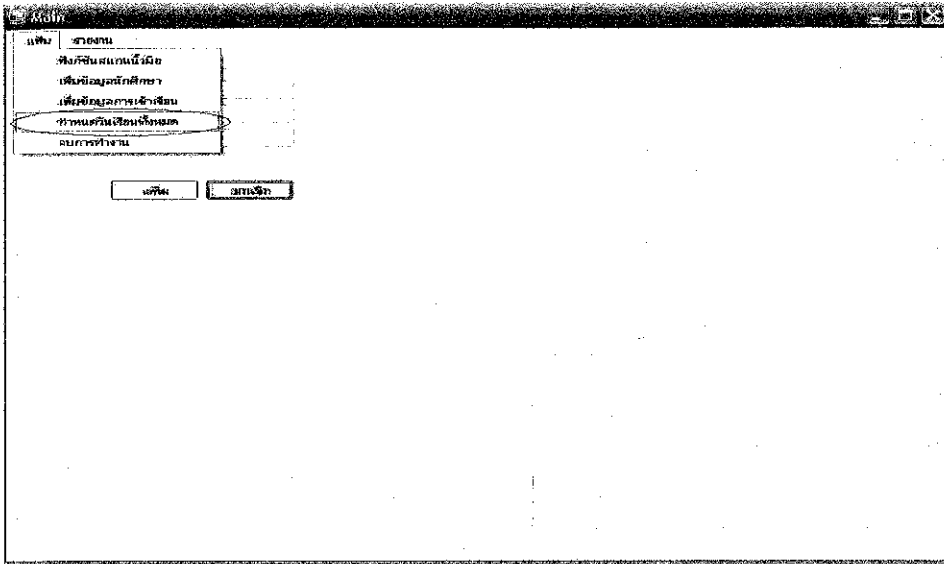
1. เปิดโปรแกรมสแกนลายนิ้วมือแบบไร้สายโปรแกรมจะสามารถใช้งานได้ทันที



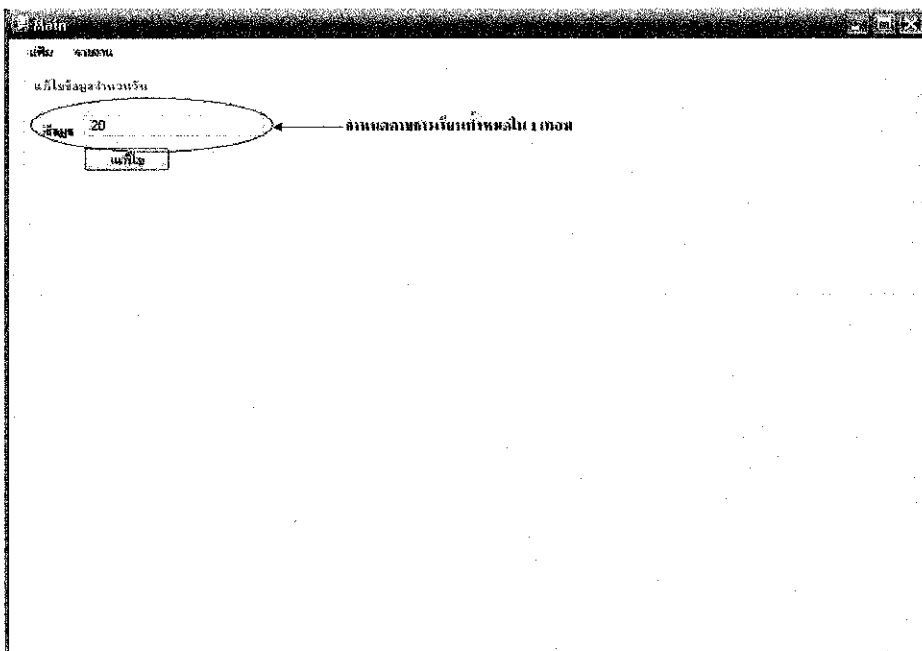
รูปที่ 4.2 โปรแกรมสแกนลายนิ้วมือแบบไร้สาย



2. ไปที่ปุ่ม เลือกกำหนดวันเรียนทั้งหมด ดังรูปที่ 4.3 เพื่อกำหนดคาบเรียนทั้งหมด  
 ดังเช่น รูปที่ 4.4

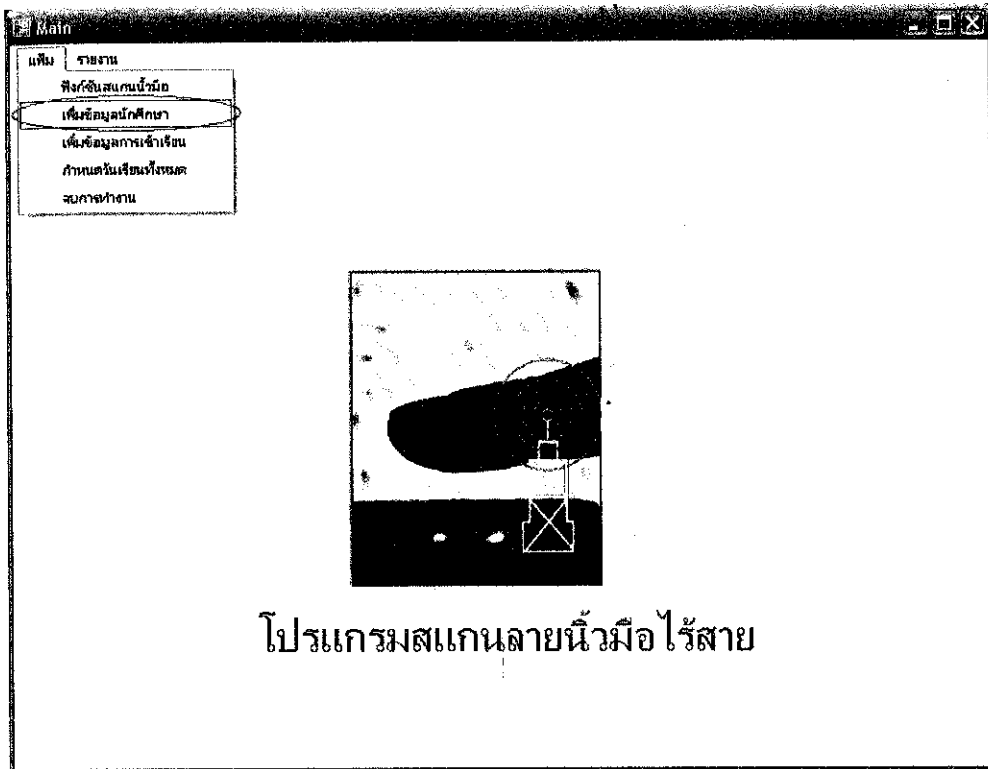


รูปที่ 4.3 การกำหนดคาบเรียนทั้งหมดในแต่ละเทอม



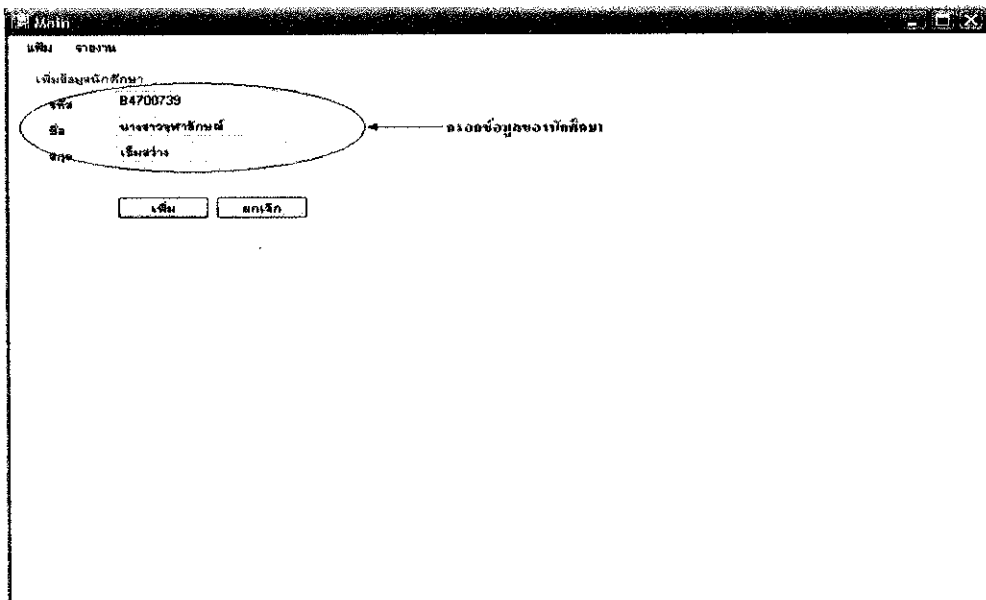
รูปที่ 4.4 กรอกคาบเรียนทั้งหมดลงในช่องข้อมูล

3. ไปที่ปุ่มเพิ่มและเลือก เพิ่มข้อมูลนักศึกษา เพื่อเป็นการเก็บ ID ชื่อและรหัสนักศึกษา



รูปที่ 4.5 เลือกเพิ่มข้อมูลนักศึกษา

4. กรอกข้อมูลลงใน รหัส ชื่อ และนามสกุล ดังรูป



รูปที่ 4.6 กรอกข้อมูลของนักศึกษา

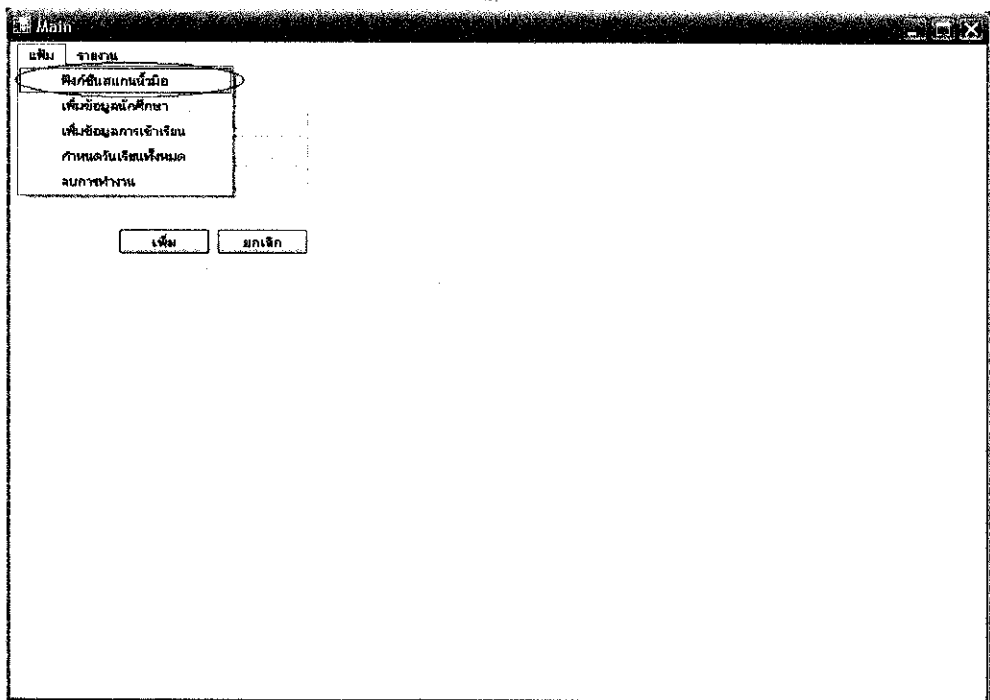
5. เลือก  เพื่อเป็นการเก็บรายชื่อนักศึกษา

6. แสดง ID Enroll รายชื่อ และรหัสของนักศึกษาทั้งหมดที่ทำการรอกกลงไปในชั้นตอนที่ 4

ID Enroll	รหัสนักศึกษา	ชื่อ	นามสกุล
1	B4700463	นายคมกฤษ	ไธสงค์
2	B4700739	นางสาวจุฬาลักษณ์	เขินสว่าง
3	B4700913	นายวีรวัฒน์	แสนบุญศิริ
4	B4701889	นางสาวนุชนาฏ	ผาเทียน
5	B4702009	นายอนุชา	แสงผณี
6	B4702399	นายปัญญาวิทย์	ชื่อตรง
7	B4702689	นางสาวศิษญา	ชัยปัญญา
8	B4703983	นางสาววิลาสินี	ธนะสำเร็จ
9	B4704270	นางสาวศรณี	สิงห์สง่า
10	B4704454	นายศรินทร์	เขารัฐนพ
11	B4704911	นายสิงหนะ	วชิรเพชรปราชดี
12	B4705055	นางสาวสุจิตรา	โหมคศิริ
13	B4705581	นางสาววชิรญา	ชินทร์นอก
14	B4705888	นายเอกชัย	ชมระกา
15	B4705885	นายเอกนรินทร์	พลแสง
16	B4706212	นายกิตติกร	ศุภเสถียรกุล
17	B4706472	นายโกวิท	ภิรมย์กิจ
18	B4706569	นายคมกฤษ	วงษ์แก้ว
19	B4706953	นางสาวจุฬารัตน์	ชัยพาณิชย์
20	B4707301	นายวีรพงศ์	พลทวง
21	B4707776	นายคณัย	ธิตสาณิก
22	B4707936	เกยพนงศักดิ์	งานเจริญ
23	B4708063	นายพศพล	เทศานัง

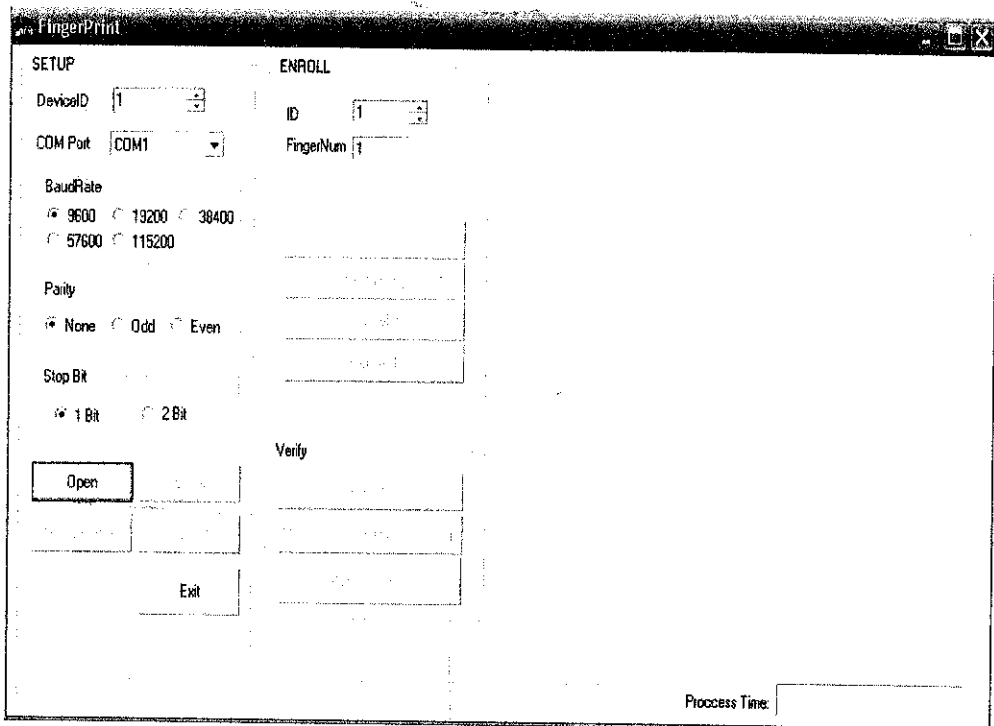
รูปที่ 4.7 แสดงข้อมูลนักศึกษา

7. ไปที่แฟ้มเลือก ฟังก์ชันสแกนนิ้วมือ



รูปที่ 4.8 เลือกฟังก์ชันสแกนนิ้วมือ

## 8. แสดงตัวโปรแกรม Fingerprint



รูปที่ 4.9 โปรแกรม Fingerprint

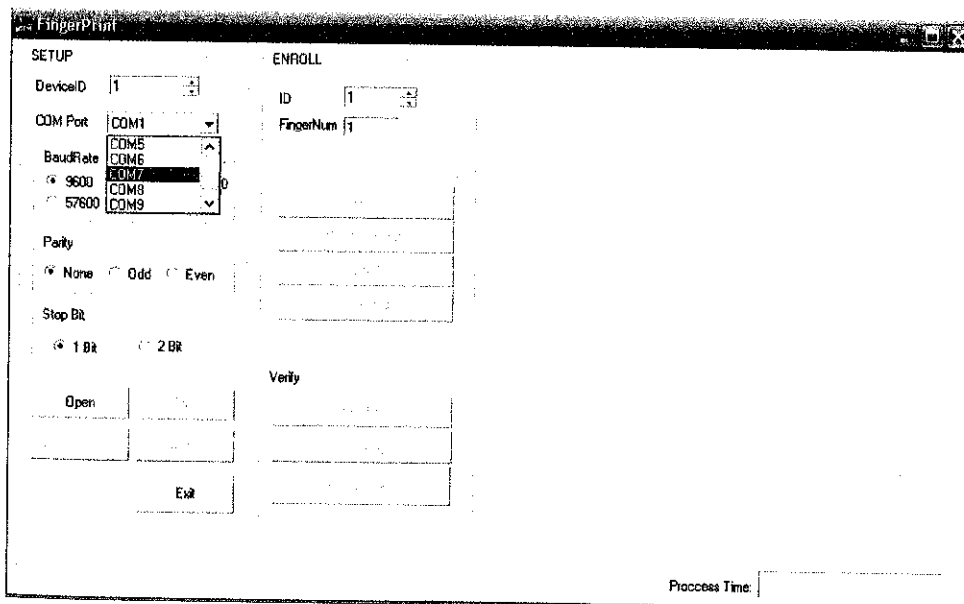
## 9. เลือก comport ให้ตรงกับคอมพิวเตอร์ที่ใช้ดังรูปที่

และใช้ค่า Device = 1

Baud Rate = 9600

Parity = None

Stop Bit = 1 Bit

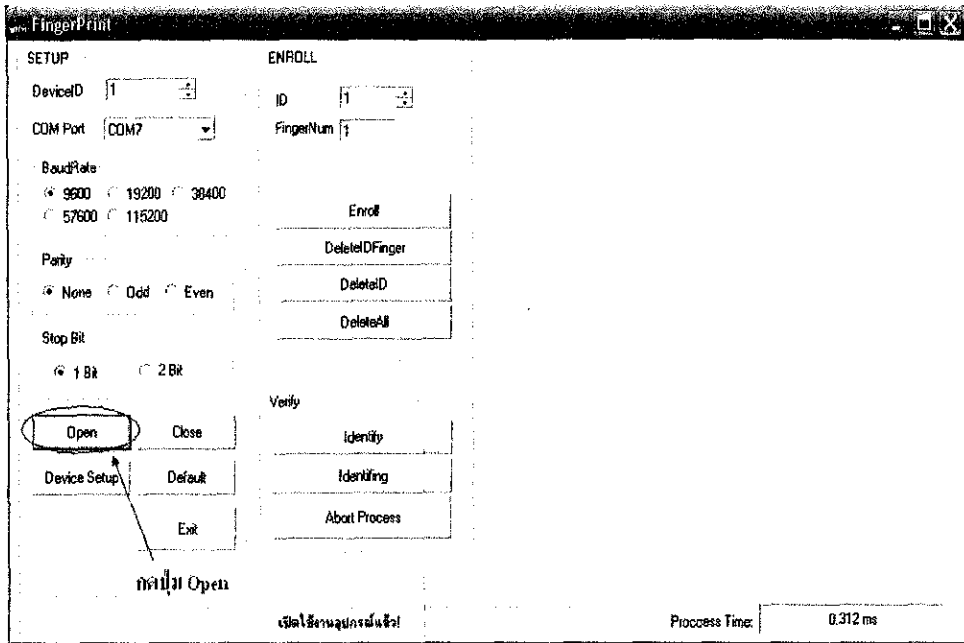


รูปที่ 4.10 โปรแกรม Fingerprint แสดงการตั้งค่าต่างๆของโปรแกรม

10. เลือกปุ่ม



ตัวโปรแกรมจะแสดงดังรูปที่



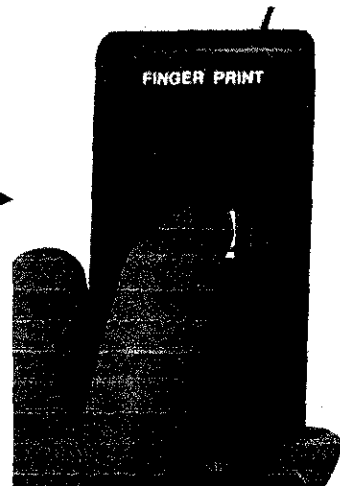
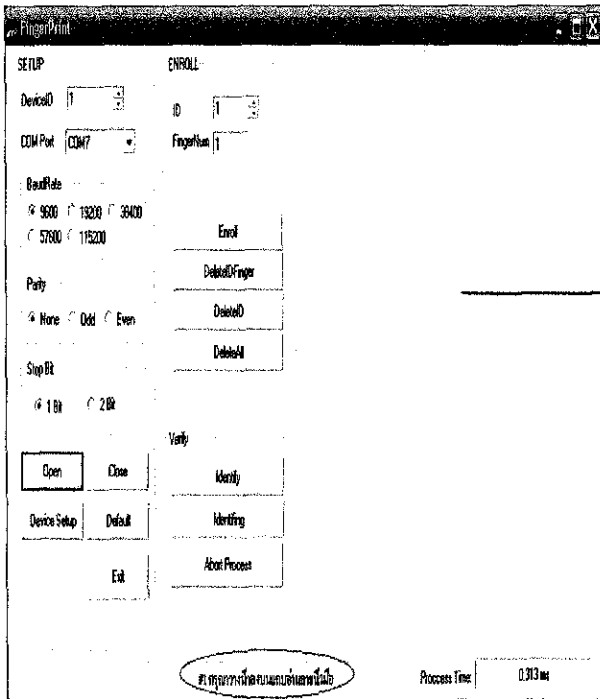
รูปที่ 4.11 ทำการเปิดโปรแกรมเพื่อใช้งาน

11. เลือกปุ่ม

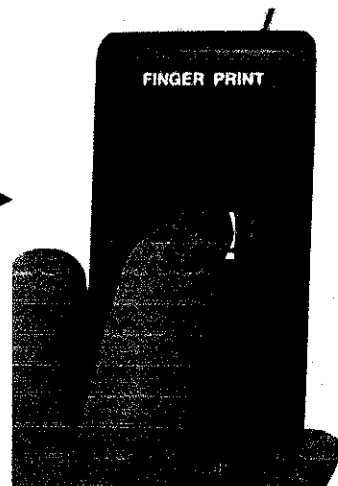
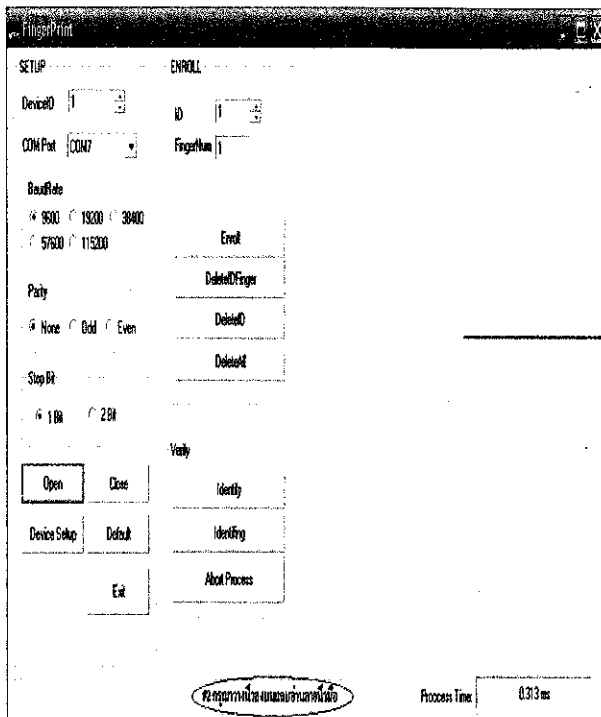
Enroll

เพื่อเก็บลายนิ้วมือทั้ง 3 ครั้งและต้อง

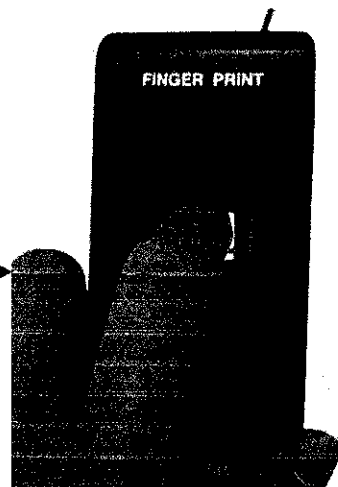
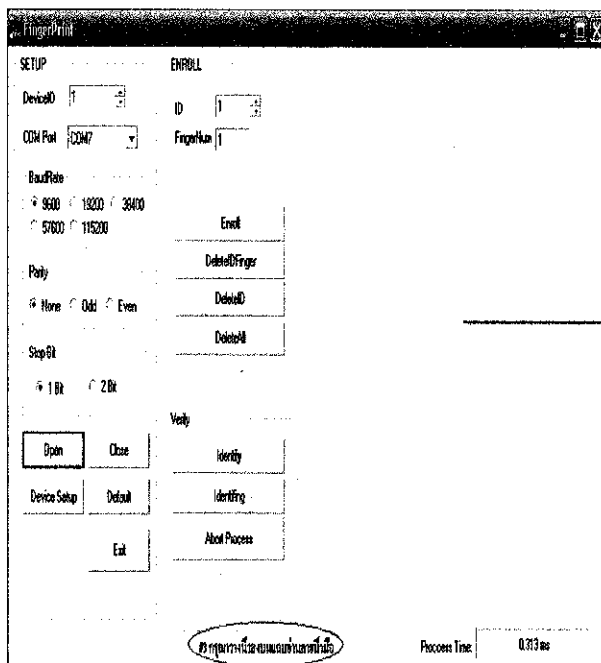
ใช้ ID ตรงกับรายชื่อของนักศึกษาที่เก็บไว้ในขั้นตอนที่ 6.



รูปที่ 4.12 วางลายนิ้วมือที่แถบอ่านลายนิ้วมือครั้งที่ 1



รูปที่ 4.13 วางลายนิ้วมือที่แถบอ่านลายนิ้วมือครั้งที่ 2

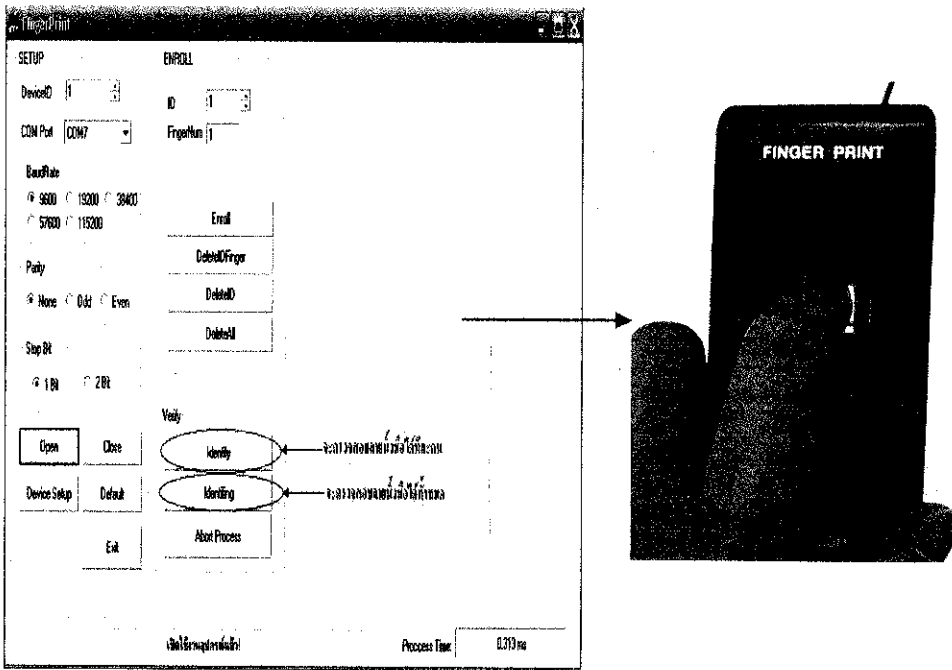


รูปที่ 4.14 วางลายนิ้วมือที่แถบอ่านลายนิ้วมือครั้งที่ 3

หมายเหตุ ถ้าวางลายนิ้วมือลงบนแถบอ่านลายนิ้วมือซ้ำประมาณ 5 วินาที จะหมดเวลาจะต้องทำการเลือกปุ่ม Enroll อีกครั้งเพื่อบันทึกลายนิ้วมือ

4.3 ขั้นตอนการตรวจสอบลายนิ้วมือ (Identification)

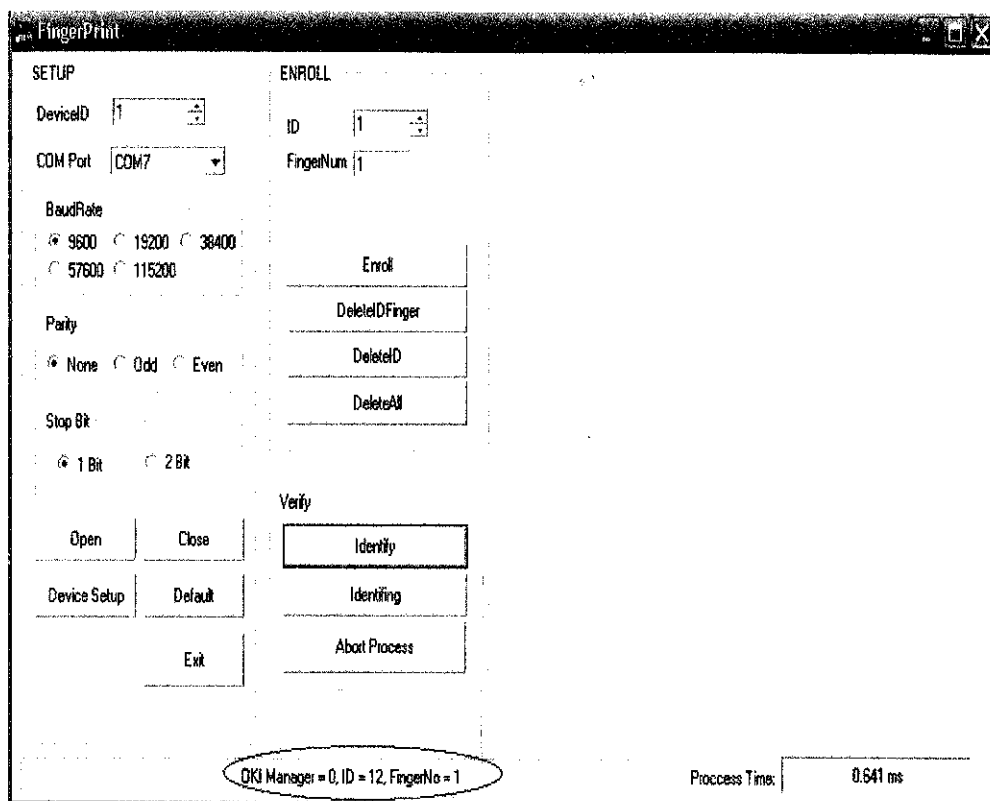
12. ขั้นตอนการพิมพ์ลายนิ้วมือเพื่อตรวจสอบการเข้าเรียนของนักศึกษาในแต่ละคาบ  
 การเรียนโดยเลือกที่ปุ่ม **Identify** หรือ **Identifying**



รูปที่ 4.15 ทำการตรวจสอบลายนิ้วมือ

Identify	จะสามารถตรวจสอบลายนิ้วมือได้ที่ละคน
Identifying	จะสามารถตรวจสอบลายนิ้วมือได้ทั้งหมด

13. ทำการพิมพ์ลายนิ้วมือของนักศึกษาที่ต้องการตรวจสอบรายชื่อเข้าห้องเรียนที่แถบอ่านลายนิ้วมือเพียง 1 ครั้ง เครื่อง FINGER PRINT OEM2000P-44b0 ทำการประมวลผลลายนิ้วมือว่าเป็นลายนิ้วมือที่อยู่ในหน่วยความจำหรือไม่ ถ้าประมวลแล้วพบว่าเป็นนิ้วที่ได้ทำการจัดเก็บไว้แล้ว เครื่อง FINGER PRINT OEM2000P-44b0 ตรวจพบลายนิ้วมือโปรแกรมก็จะแสดงให้เห็นว่าลายนิ้วมือนี้ตรงกับ ID ลายนิ้วมือของนักศึกษาคนใด



รูปที่ 4.16 แสดงการตรวจพบลายนิ้วมือของนักศึกษาที่ได้บันทึกไว้ว่าตรงกัน

ทำเช่นนี้จนครบทุกคนที่เข้าห้องเรียน

#### หมายเหตุ

1. ในส่วนของโปรแกรมสแกนลายนิ้วมือแบบไร้สายนั้น ถ้าข้อมูลของนักศึกษามีจำนวนมาก โปรแกรมอาจจะมีการหน่วงเวลาในการตรวจสอบรายชื่อเกิดขึ้นแต่จะสามารถนำโปรแกรมไปพัฒนาเพื่อแก้ปัญหานี้ได้
2. ในการตรวจสอบรายชื่อโดยให้ปุ่ม Identifying เมื่อนักศึกษาคนแรกพิมพ์นิ้วมือนักศึกษาคนถัดมาต้องทำการพิมพ์ลายนิ้วมือภายในเวลา 5 วินาที
3. ความแม่นยำของเครื่องอ่านลายนิ้วมือในการทดลองของโครงการนี้มีค่าประมาณร้อยละ 95%

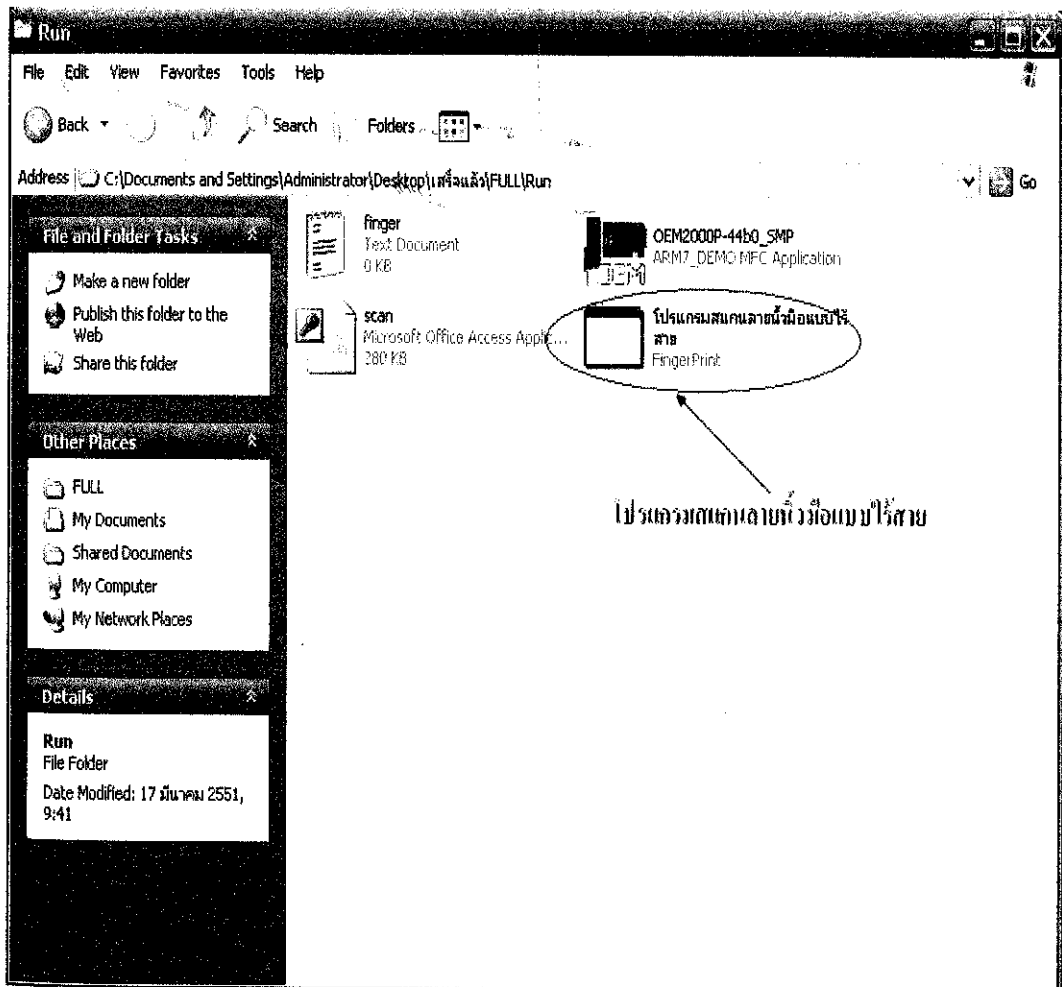


## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผล

การทำงานนี้ใช้ Biometric ที่เกี่ยวกับลายนิ้วมือมาใช้ในการทำงาน และ ประกอบกับการใช้เทคโนโลยีไร้สายเข้ามาช่วยโดยการเก็บลายนิ้วมือของกลุ่มนักศึกษาตัวอย่าง และทดลองตรวจสอบลายนิ้วมือของนักศึกษาในคาบเรียนนั้น เวลาที่ใช้ในการตรวจสอบน้อยกว่า ความปลอดภัยของเครื่องอ่านลายนิ้วมือมีมาก และความปลอดภัยของเครื่องคอมพิวเตอร์ที่ใช้มีมากกว่าการใช้เครื่องอ่านลายนิ้วมือซึ่งต่อกับคอมพิวเตอร์โดยไม่ผ่านอุปกรณ์ไร้สาย และอาจารย์ผู้สอนจะสามารถตรวจสอบรายชื่อการเข้าเรียนของรายวิชาได้จาก โปรแกรมสแกนลายนิ้วมือแบบไร้สาย



รูปที่ 5.1 แสดงการเข้าโปรแกรมเพื่อคุณผลตารางสรุปการเข้าห้องเรียน

## 1. ไปที่รายงาน เลือกรายงานสรุปผลการเข้าห้องเรียน

Main

เพิ่ม รายงาน

รายงานสรุปผลการเข้าห้องเรียน

รายงานข้อมูลนักศึกษา

รายงานแสดงเวลาเข้าออก

### สรุปผลการเข้าห้องเรียน

เลข	ชื่อ	ทั้งหมด	ท	ขาด
1	นายคมกฤษ โสภศรี	20	2	18
2	นางสาวจุฬาลักษณ์ เข็มสว่าง	20	4	16
3	นายชัยวัฒน์ แสงบุญศิริ	20	3	17
4	นางสาวนุชพภาพ ศาเพ็ญ	20	3	17
5	นายสุธา แสงเมธี	20	3	17
6	นายบุญญาวีร์ ช็อคตรง	20	5	15
7	นางสาวพิชญา ชัยบุญญา	20	4	16
8	นางสาววิลาสินี ธนะจำเริญ	20	5	15
9	นางสาวศศิ สิงห์สำ	20	4	16
10	นายศักดิ์รินทร์ เขียวขุนทด	20	3	17
11	นายสิงหนะช วรรณเพชรปราชญ์	20	4	16
12	นางสาวสุจิตรา โทมศิริบุญ	20	3	17
13	นางสาวอภิญญา สิงห์นอก	20	3	17
14	นายเกษมชัย ธนระกา	20	4	16
15	นายเกษมรินทร์ พลแสง	20	4	16
16	นายกิตติกร สุภเสถียรภูก	20	4	16
17	นายไกรวิฑ วัฒนชัยกิจ	20	3	17
18	นายคมกฤษ วัฒนแก้ว	20	4	16
19	นางสาวจุฬาลักษณ์ ชัยมาพันธ์	20	4	16
20	นายชัยพัฒน์ พลพวก	20	3	17
21	นายณัฏฐ วัฒนสาทิัก	20	1	19
22	นายทองศักดิ์ งามเจริญ	20	1	19
23	นายทศพล เสดาหัง	20	2	18
26	นางสาวผกาภาศ วัฒนขุนทด	20	1	19

รูปที่ 5.2 แสดงสรุปผลการเข้าห้องเรียนของนักศึกษา

## 2. ไปที่รายงาน เลือกรายงานแสดงเวลาเข้าเรียน

Main

รายงาน

รายงานสรุปผลการเข้าห้องเรียน  
รายงานข้อมูลนักศึกษา  
รายงานแสดงเวลาเข้าออก

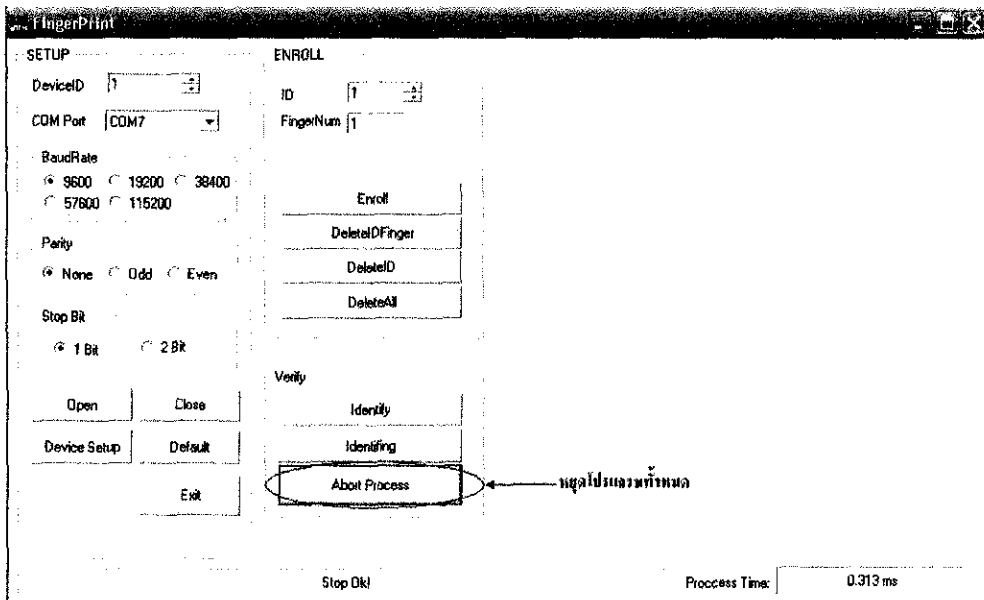
### รายงานแสดงเวลาเข้า

ชื่อ	วันที่เข้า	เวลาเข้า
นางสาวจุฑารัตน์	18/3/2551	19:33:17
นางสาวจุฑารัตน์	18/3/2551	19:33:17
นางสาวจุฑารัตน์	18/3/2551	18:27:15
นางสาวจุฑารัตน์	18/3/2551	11:22:51
นางสาวจุฑาลักษณ์	18/3/2551	19:27:15
นางสาวจุฑาลักษณ์	18/3/2551	19:33:17
นางสาวจุฑาลักษณ์	18/3/2551	19:33:17
นางสาวจุฑาลักษณ์	18/3/2551	11:22:51
นางสาวบุษนภ	18/3/2551	19:27:15
นางสาวบุษนภ	18/3/2551	19:33:17
นางสาวบุษนภ	18/3/2551	19:33:17
นางสาวภกมาศ	18/3/2551	19:30:09
นางสาวศิษญา	18/3/2551	19:33:17
นางสาวศิษญา	18/3/2551	19:33:17
นางสาวศิษญา	18/3/2551	11:22:51
นางสาวศิษญา	18/3/2551	19:27:15
นางสาวจวพร	18/3/2551	19:27:15
นางสาวจวพร	18/3/2551	19:30:09
นางสาววิลาสินี	18/3/2551	19:30:09
นางสาววิลาสินี	18/3/2551	19:27:15
นางสาววิลาสินี	18/3/2551	19:33:17
นางสาววิลาสินี	18/3/2551	11:22:51
นางสาววิลาสินี	18/3/2551	19:33:17
นางสาวศศิ	18/3/2551	19:33:17

รูปที่ 5.3 แสดงผลการเข้าห้องเรียนของนักศึกษาแต่ละคน

3. เมื่อใช้งานเรียบร้อยแล้วต้องการหยุดโปรแกรมให้เลือกที่ปุ่ม

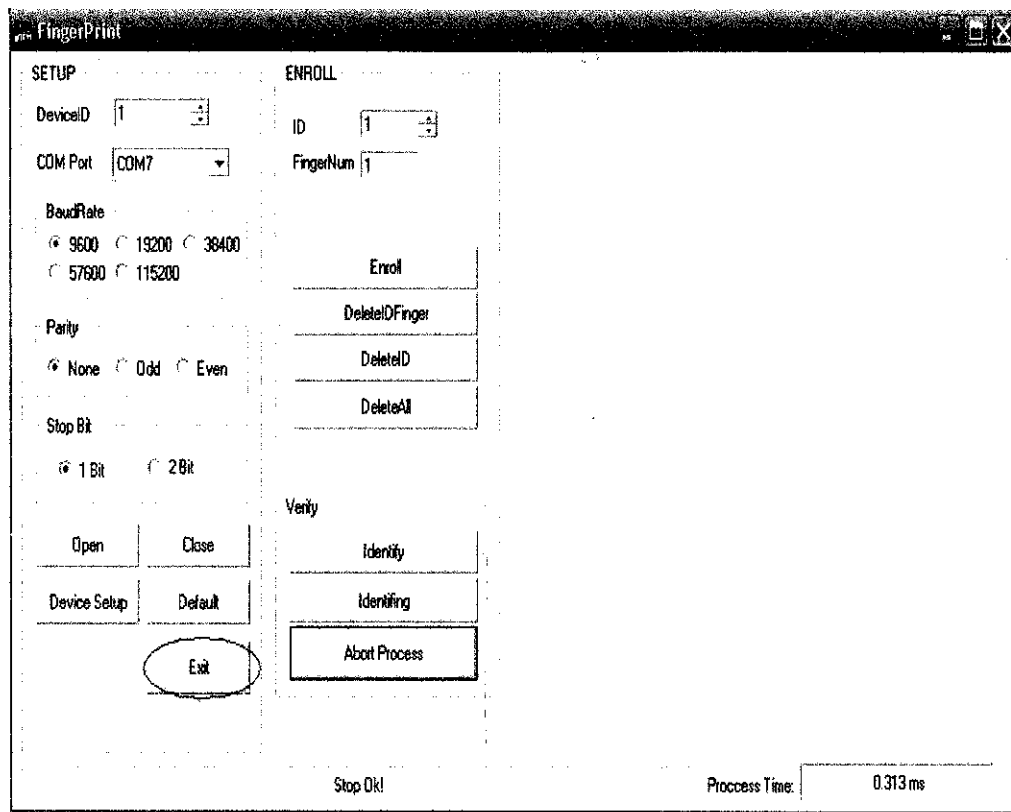
Abort Process



รูปที่ 5.4 กดเพื่อหยุดการทำงานที่โปรแกรมทำอยู่

4. ออกจากโปรแกรมเลือกที่ปุ่ม

Exit



รูปที่ 5.5 ออกจากโปรแกรม

## 5.2) ขั้นตอนการลบข้อมูล เมื่อจำเป็นต้องลบลายนิ้วมือ

กดปุ่ม  เพื่อใช้ลบเฉพาะ ID ใด ID หนึ่งเท่านั้น ซึ่งข้อมูลจะเก็บอยู่ในเครื่อง OEM500P

กดปุ่ม  เพื่อใช้ลบข้อมูล ID ที่เก็บไว้ในเครื่อง OEM500P ทั้งหมด

### 5.3 ปัญหาและอุปสรรคในการทำโครงการ

1. เครื่องอ่านลายนิ้วมือที่จะนำมาพัฒนาจากการเชื่อมต่อกับเครื่องคอมพิวเตอร์แบบ Universal Serial Bus (USB) มาเป็นเชื่อมต่อแบบไร้สายนั้นไม่สามารถทำการต่อกับชุดส่งสัญญาณแบบไร้สาย ได้จึงต้องทำการเปลี่ยนเครื่องอ่านลายนิ้วมือมาเป็นแบบ RS232 คือเครื่อง FINGER PRINT OEM2000P-44b0 ซึ่งทางผู้จัดทำต้องทำการหาข้อมูลและสั่งซื้อเครื่องเป็นเวลานาน

2. ผู้จัดทำมีความรู้ความเข้าใจเกี่ยวกับการเขียนโปรแกรมวิชวล สตูดิโอ 2005 (Visual Studio 2005) โปรแกรม C++ และโปรแกรมไมโครซอฟต์ แอคเซส (Microsoft Access) น้อยจึงต้องทำความเข้าใจเรียนรู้เกี่ยวกับการเขียนโปรแกรมเป็นเวลานานและต้องอาศัยความรู้จากผู้มีความชำนาญในเรื่องนี้คอยให้คำแนะนำ

3. เครื่องอ่านลายนิ้วมือมีความซับซ้อนในการใช้งานมากทางผู้จัดทำมีความรู้ความเข้าใจเกี่ยวกับการใช้งานน้อยเกินไป จึงต้องใช้เวลาในการศึกษาคุณสมบัติของเครื่องนาน และเมื่อนำไปใช้งานจริงจึงเกิดปัญหาบ้างในบางครั้งการส่งข้อมูลผ่านทางเทคโนโลยีไร้สายในบางครั้งการรับส่งข้อมูลนั้นมีโอกาสผิดพลาดในเรื่องของการสูญหายของข้อมูลบ้างเหมือนกันเนื่องจากกลไกในการรับส่งข้อมูลของเครื่อง ET-RF24G V1.0 นั้น จะมีการตรวจสอบข้อมูลทุก Byte ที่รับได้จาก RF เสมอ ซึ่งถ้าพบว่ามีความผิดพลาดเกิดขึ้นจะทิ้งข้อมูล Byte นั้นไป ซึ่งผู้ใช้ควรมีกลไกในการตรวจสอบข้อมูลที่รับส่งกันว่าครบถ้วนหรือไม่ด้วย ซึ่งหากพบว่ามีการสูญหายของข้อมูลเกิดขึ้นก็ให้ร้องขอให้มีการส่งข้อมูลนั้นซ้ำนั้นๆ ใหม่อีกครั้งหนึ่ง ก็จะสามารถแก้ไขปัญหาดังกล่าวได้

### 5.4 ข้อจำกัดในการทำโครงการ

1. อุปกรณ์ที่ใช้ในการทำโครงการนี้ คือ เครื่องอ่านลายนิ้วมือซึ่งจะสามารถเก็บลายนิ้วมือได้มากที่สุด 2,000 ลายนิ้วมือ
2. สามารถใช้งานแบบไร้สายห่างจากเครื่องคอมพิวเตอร์เป็นระยะมากที่สุด 180 เมตร (ในทึกลางแจ้ง) และ ระยะ 50 เมตร (ในตัวอาคาร)
3. แหล่งจ่ายไฟให้กับเครื่องอ่านลายนิ้วมือ และเครื่อง Wireless จะใช้งานได้ต่อเนื่องได้มากที่สุดประมาณ 1 ชั่วโมง

### 5.5 ข้อเสนอแนะ

1. สามารถพัฒนาการแสดงผลการตรวจสอบลายชื่อในรูปแบบตารางที่ให้อ่านดูได้ง่ายและสะดวกยิ่งขึ้น
2. สามารถพัฒนาแหล่งจ่ายไฟให้สามารถจ่ายไปได้มากและนานกว่า

## ภาคผนวก ก

## โปรแกรม

## 1. โปรแกรมเพื่อติดต่อกับฐานข้อมูล (DBAccess)

```
Imports System.Data.OleDb

Public Class DBAccess

    Private _strConnectionString As String =
"Provider=Microsoft.JET.OLEDB.4.0;data source=scan.mdb"

    Private _objConnection As OleDbConnection
    Private _objCommand As OleDbCommand
    Private _iRowAffect As Integer
    Private _objOleDbDataReader As OleDbDataReader

    Public Function ExecuteCommand(ByVal strSql As String) As Integer

        _objConnection = New OleDbConnection()
        _objConnection.ConnectionString = _strConnectionString
        _objCommand = New OleDbCommand()

    Try
        _objConnection.ConnectionString = _strConnectionString
        If _objConnection.State = ConnectionState.Open Then
            _objConnection.Close()
        End If
        _objConnection.Open()
        _objCommand.CommandType = CommandType.Text
        _objCommand.CommandText = strSql
        _objCommand.Connection = _objConnection
        _iRowAffect = _objCommand.ExecuteNonQuery()
    End Try
End Class
```

```
Catch ex As Exception
    _iRowAffect = -1
    MessageBox.Show(ex.Message)
Finally
    _objConnection.Close()
End Try
Return _iRowAffect
End Function

Public Function ExecuteReader(ByVal strSql As String) As OleDbDataReader

    _objConnection = New OleDbConnection()
    _objConnection.ConnectionString = _strConnectionString
    _objCommand = New OleDbCommand()

    Try
        _objConnection.ConnectionString = _strConnectionString
        If _objConnection.State = ConnectionState.Open Then
            _objConnection.Close()
        End If
        _objConnection.Open()
        _objCommand.CommandType = CommandType.Text
        _objCommand.CommandText = strSql
        _objCommand.Connection = _objConnection
        _objOleDbDataReader = _objCommand.ExecuteReader()
        Return _objOleDbDataReader
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
    Return _objOleDbDataReader
```

```

End Function

Public ReadOnly Property ConnectionString()
    Get
        Return _strConnectionString
    End Get
End Property

End Class

```

#### 4. โปรแกรมที่อ่านข้อมูลจาก Text File (ReadDaFile)

```

Imports System.Data.OleDb

Public Class DBAccess
    Private _strConnectionString As String = "Provider=Microsoft.JET.OLEDB.4.0;data
source=scan.mdb"

    Private _objConnection As OleDbConnection
    Private _objCommand As OleDbCommand
    Private _iRowAffect As Integer
    Private _objOleDbDataReader As OleDbDataReader

    Public Function ExecuteCommand(ByVal strSql As String) As Integer

        _objConnection = New OleDbConnection()
        _objConnection.ConnectionString = _strConnectionString
        _objCommand = New OleDbCommand()

    Try
        _objConnection.ConnectionString = _strConnectionString
        If _objConnection.State = ConnectionState.Open Then

```



```
        _objConnection.Close()
    End If
    _objConnection.Open()
    _objCommand.CommandType = CommandType.Text
    _objCommand.CommandText = strSql
    _objCommand.Connection = _objConnection
    _iRowAffect = _objCommand.ExecuteNonQuery()

Catch ex As Exception
    _iRowAffect = -1
    MessageBox.Show(ex.Message)
Finally
    _objConnection.Close()
End Try
Return _iRowAffect
End Function

Public Function ExecuteReader(ByVal strSql As String) As OleDbDataReader

    _objConnection = New OleDbConnection()
    _objConnection.ConnectionString = _strConnectionString
    _objCommand = New OleDbCommand()

    Try
        _objConnection.ConnectionString = _strConnectionString
        If _objConnection.State = ConnectionState.Open Then
            _objConnection.Close()
        End If
        _objConnection.Open()
        _objCommand.CommandType = CommandType.Text
        _objCommand.CommandText = strSql
```

```

        _objCommand.Connection = _objConnection
        _objOleDbDataReader = _objCommand.ExecuteReader()
        Return _objOleDbDataReader
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
    Return _objOleDbDataReader
End Function

Public ReadOnly Property ConnectionString()
    Get
        Return _strConnectionString
    End Get
End Property

End Class

```

## 5. DBManage

```

Imports System.Data.OleDb

Public Class DBManage
    Private _objDbAccess As New DBAccess
    Private _strConnectionString As String = _objDbAccess.ConnectionString

    Public Function GetDataTable(ByVal strSql As String) As DataTable
        _objDbAccess = New DBAccess
        _strConnectionString = _objDbAccess.ConnectionString
        Dim _objDataAdapter As New OleDbDataAdapter(strSql, _strConnectionString)
        Dim _objDataSet As New DataSet
        Dim _objDataTable As DataTable
    End Function

```

```

On Error Resume Next
_objDataAdaptor.Fill(_objDataSet)
_objDataTable = _objDataSet.Tables(0)
Return _objDataTable

```

```
End Function
```

```
Public Function ExecuteInsert(ByVal strSql As String) As Boolean
```

```

    Dim _iRowAffect As Integer
    _objDbAccess = New DBAccess
    _strConnectionString = _objDbAccess.ConnectionString
    _iRowAffect = _objDbAccess.ExecuteCommand(strSql)
    If _iRowAffect > 0 Then
        Return True
    Else
        Return False
    End If
End Function

```

```
End Class
```

## 6. โปรแกรมสำคัญ (frmMain)

```
Public Class frmMain
```

```

    Private Const VIEW_MAIN = 1
    Private Const VIEW_TIME_RECORD = 2
    Private Const VIEW_STUDENT_INFO = 3
    Private Const VIEW_SUMARY_REPORT = 4
    Private Const VIEW_TIME_REPORT = 5

```

```

Private Const VIEW_STUDENT_REPORT = 6
Private Const VIEW_EDIT_TOTAL_DAY = 7
Private _iCurView As Integer
Dim sID As Long

Private Sub btnAddStdData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddStdData.Click
    Dim _objStudent As New Student
    Dim _objDBManage As New DBManage
    Dim _boolError As Boolean
    _objStudent.Id = txtStudentId.Text
    _objStudent.Name = txtStudentName.Text
    _objStudent.LastName = txtStudentLastName.Text
    If txtStudentId.Text = "" Then
        MessageBox.Show("กรุณาป้อนรหัสนักศึกษา!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    ElseIf txtStudentName.Text = "" Then
        MessageBox.Show("กรุณาป้อนชื่อนักศึกษา!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    ElseIf txtStudentLastName.Text = "" Then
        MessageBox.Show("กรุณาป้อนนามสกุลนักศึกษา!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    Else
        If MessageBox.Show("คุณต้องการเพิ่มข้อมูลนักศึกษาใหม่ ใช่หรือไม่?", "ยืนยันการ
เพิ่มข้อมูลนักศึกษา", MessageBoxButtons.YesNo, MessageBoxIcon.Question) =
System.Windows.Forms.DialogResult.Yes Then
            _boolError = _objDBManage.ExecuteInsert("INSERT INTO
Contacts(stuID,FirstName,LastName) VALUES('" + _objStudent.Id + "','" +
_objStudent.Name + "','" + _objStudent.LastName + "')")
            If _boolError = True Then

```

```

        MessageBox.Show("เพิ่มข้อมูลเรียบร้อยแล้ว!", "ผลการทำงาน",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
    Else
        MessageBox.Show("ไม่สามารถเพิ่มข้อมูลได้!", "ผลการทำงาน",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
End If
End If
End Sub

Private Sub btnStudentInfoCancel_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnStudentInfoCancel.Click
    txtStudentId.Clear()
    txtStudentName.Clear()
    txtStudentLastName.Clear()
End Sub

Private Sub btnAddTimeRecord_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnAddTimeRecord.Click
    Dim _objStdTimeRecord As New StudentTimeRecord
    Dim _objDBManage As New DBManage
    Dim _objDBAccess As New DBAccess
    Dim _objOleDbDataReader As OleDb.OleDbDataReader
    Dim _boolError As Boolean
    Dim _iStuID As Integer
    _objStdTimeRecord.StudentId = txtTimeRecordStudentId.Text

    If txtTimeRecordStudentId.Text = "" Then
        MessageBox.Show("กรุณาป้อนรหัสนักศึกษา!", "ผลการทำงาน",
        MessageBoxButtons.OK, MessageBoxIcon.Information)
    Else

```

```

_objOleDbDataReader = _objDBAccess.ExecuteReader("SELECT sID FROM
Contacts WHERE stuID LIKE " + _objStdTimeRecord.StudentId + """)
If _objOleDbDataReader.HasRows Then
    While _objOleDbDataReader.Read()
        _iStuID = Integer.Parse(_objOleDbDataReader("sID").ToString())
    End While
End If

_objStdTimeRecord.DateIn = dtpStudentTimeIn.Value

Try
    _objStdTimeRecord.TimeIn =
DateTime.Parse(txtTimeIn.Text).ToLongTimeString()
    If MessageBox.Show("คุณต้องการเพิ่มข้อมูลการเข้าเรียน ใหมหรือไม่?", "ยืนยัน
การเพิ่มข้อมูลการเข้าเรียน", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
= System.Windows.Forms.DialogResult.Yes Then
        If _iStuID <= 1 Then
            MessageBox.Show("ไม่พบรหัสนักศึกษาในฐานข้อมูล!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Else
            _boolError = _objDBManage.ExecuteInsert("INSERT INTO
Ident(sID,idDate,idTime) VALUES(" + _iStuID.ToString() + "," +
_objStdTimeRecord.DateIn + "," + _objStdTimeRecord.TimeIn + ")")
            If _boolError = True Then
                MessageBox.Show("เพิ่มข้อมูลเรียบร้อยแล้ว!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information)
            Else
                MessageBox.Show("ไม่สามารถเพิ่มข้อมูลได้!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information)
            End If
        End If
    End If
End If

```

```
Catch ex As Exception
```

```
    MessageBox.Show("กรุณาป้อนข้อมูลเวลาเข้าเรียนให้ถูกต้อง!", "ผลการทำงาน",  
    MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
End Try
```

```
End If
```

```
End Sub
```

```
Private Sub btnTimeRecordCancel_Click(ByVal sender As System.Object, ByVal e  
As System.EventArgs) Handles btnTimeRecordCancel.Click
```

```
    txtTimeRecordStudentId.Clear()
```

```
    txtTimeIn.Clear()
```

```
End Sub
```

```
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
    UpdateView(VIEW_MAIN)
```

```
    Dim _objDataReaderFile As New ReadDataFile
```

```
    _objDataReaderFile.ReadFile()
```

```
End Sub
```

```
Private Sub ExitMenu_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles ExitMenu.Click
```

```
    Application.Exit()
```

```
End Sub
```

```
Private Sub TimeRecordEachPersonReportToolStripMenuItem_Click(ByVal sender  
As System.Object, ByVal e As System.EventArgs) Handles  
StudentReportToolStripMenuItem.Click
```

```
    UpdateView(VIEW_STUDENT_REPORT)
```

```

Dim _objDBManage As New DBManage
Dim _objDataTable As New DataTable
_objDataTable.Clear()
_objDataTable = _objDBManage.GetDataTable("SELECT sID, stuID, FirstName,
LastName FROM Contacts;")
grdStudentReport.DataSource = _objDataTable

With grdStudentReport
    .AllowUserToAddRows = False
    .AllowUserToResizeColumns = False
    .AllowUserToResizeRows = False
    .AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill
    .AutoSizeRowsMode =
DataGridViewAutoSizeRowsMode.AllCellsExceptHeaders
    .Columns(0).HeaderCell.Value = "ID Enroll"
    .Columns(1).HeaderCell.Value = "รหัสนักศึกษา"
    .Columns(2).HeaderCell.Value = "ชื่อ"
    .Columns(3).HeaderCell.Value = "นามสกุล"
    .EditMode = DataGridViewEditMode.EditProgrammatically
    .Font = New Font(grdReportTime.Font.Name, 10.0!)
    .MultiSelect = False
    .RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True
    .RowHeadersVisible = False
    .SelectionMode = DataGridViewSelectionMode.FullRowSelect
End With
End Sub

Private Sub SummaryReportToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SummaryReportToolStripMenuItem.Click
    Dim _objDataReaderFile As New ReadDataFile

```



```
_objDataReaderFile.ReadFile()
```

```
UpdateView(VIEW_SUMMARY_REPORT)
```

```
Dim _objDBManage As New DBManage
```

```
Dim _objDataTable As New DataTable
```

```
Dim _strQuery2 As String = "SELECT Contacts.sID, Contacts.stuID,  
[FirstName]+' '+[LastName] AS Name, Course.TotalDay, Count(Ident.sID) AS  
Present, [TotalDay]-Count([Ident.sID]) AS Absense "
```

```
_strQuery2 += "FROM Course, Contacts INNER JOIN Ident ON Contacts.sID =  
Ident.sID " + _
```

```
"GROUP BY Contacts.sID, Contacts.stuID, [FirstName]+' '+[LastName],  
Course.TotalDay;"
```

```
_objDataTable = _objDBManage.GetDataTable(_strQuery2)
```

```
grdSummaryReport.Columns.Clear()
```

```
grdSummaryReport.DataSource = _objDataTable
```

```
With grdSummaryReport
```

```
.AllowUserToAddRows = False
```

```
.AllowUserToResizeColumns = False
```

```
.AllowUserToResizeRows = False
```

```
.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill
```

```
.AutoSizeRowsMode =
```

```
DataGridViewAutoSizeRowsMode.AllCellsExceptHeaders
```

```
.Columns(0).HeaderCell.Value = "เลขที่"  
.Columns(1).HeaderCell.Value = "รหัส"  
.Columns(2).HeaderCell.Value = "ชื่อ"  
.Columns(3).HeaderCell.Value = "ทั้งหมด"  
.Columns(4).HeaderCell.Value = "มา"  
.Columns(5).HeaderCell.Value = "ขาด"  
  
.Columns(0).Width = "30"  
.Columns(1).Width = "60"  
.Columns(2).Width = "80"  
.Columns(3).Width = "25"  
.Columns(4).Width = "25"  
.Columns(5).Width = "5"  
  
.EditMode = DataGridViewEditMode.EditProgrammatically  
.Font = New Font(grdReportTime.Font.Name, 8.25!)  
.MultiSelect = False  
.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True  
.RowHeadersVisible = False  
.SelectionMode = DataGridViewSelectionMode.FullRowSelect  
  
End With  
End Sub  
  
Private Sub TimeRecordFormToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
TimeRecordFormToolStripMenuItem.Click  
    txtTimeRecordStudentId.Text = ""  
    txtTimeIn.Text = ""  
    UpdateView(VIEW_TIME_RECORD)  
  
End Sub
```

```
Private Sub StudentInformationFormToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles
```

```
StudentInformationFormToolStripMenuItem.Click
```

```
    gbStudentInfo.Text = "เพิ่มข้อมูลนักศึกษา"
```

```
    btnAddStdData.Visible = True
```

```
    btnUpdate.Visible = False
```

```
    txtStudentId.Enabled = True
```

```
    txtStudentId.Text = ""
```

```
    txtStudentName.Text = ""
```

```
    txtStudentLastName.Text = ""
```

```
    UpdateView(VIEW_STUDENT_INFO)
```

```
End Sub
```

```
Private Sub UpdateView(ByVal iView As Integer)
```

```
    _iCurView = iView
```

```
    Select Case iView
```

```
        Case VIEW_MAIN
```

```
            panelStudentInfo.Visible = False
```

```
            panelTimeRecord.Visible = False
```

```
            panelStudentReport.Visible = False
```

```
            panelSummaryReport.Visible = False
```

```
            panelTimeReport.Visible = False
```

```
            lblSummaryReport.Visible = False
```

```
            lblTimeReport.Visible = False
```

```
            lblStudentInfoReport.Visible = False
```

```
            panelEditTotalDay.Visible = False
```

```
        Case VIEW_STUDENT_INFO
```

```
            panelStudentInfo.Visible = True
```

```
            panelTimeRecord.Visible = False
```

```
            panelStudentReport.Visible = False
```

```
            panelSummaryReport.Visible = False
```

panelTimeReport.Visible = False  
lbiSummaryReport.Visible = False  
lbiTimeReport.Visible = False  
lbiStudentInfoReport.Visible = False  
panelEditTotalDay.Visible = False

#### Case VIEW\_TIME\_RECORD

panelStudentInfo.Visible = False  
panelTimeRecord.Visible = True  
panelStudentReport.Visible = False  
panelSummaryReport.Visible = False  
panelTimeReport.Visible = False  
lbiSummaryReport.Visible = False  
lbiTimeReport.Visible = False  
lbiStudentInfoReport.Visible = False  
panelEditTotalDay.Visible = False

#### Case VIEW\_SUMMARY\_REPORT

panelStudentInfo.Visible = False  
panelTimeRecord.Visible = False  
panelStudentReport.Visible = False  
panelSummaryReport.Visible = True  
panelTimeReport.Visible = False  
lbiSummaryReport.Visible = True  
lbiTimeReport.Visible = False  
lbiStudentInfoReport.Visible = False  
panelEditTotalDay.Visible = False

#### Case VIEW\_STUDENT\_REPORT

panelStudentInfo.Visible = False  
panelTimeRecord.Visible = False  
panelStudentReport.Visible = True  
panelSummaryReport.Visible = False  
panelTimeReport.Visible = False

```
lblSummaryReport.Visible = False  
lblTimeReport.Visible = False  
lblStudentInfoReport.Visible = True  
panelEditTotalDay.Visible = False
```

```
Case VIEW_TIME_REPORT
```

```
panelStudentInfo.Visible = False  
panelTimeRecord.Visible = False  
panelStudentReport.Visible = False  
panelSummaryReport.Visible = False  
panelTimeReport.Visible = True  
lblSummaryReport.Visible = False  
lblTimeReport.Visible = True  
lblStudentInfoReport.Visible = False  
panelEditTotalDay.Visible = False
```

```
Case VIEW_EDIT_TOTAL_DAY
```

```
panelStudentInfo.Visible = False  
panelTimeRecord.Visible = False  
panelStudentReport.Visible = False  
panelSummaryReport.Visible = False  
panelTimeReport.Visible = False  
lblSummaryReport.Visible = False  
lblTimeReport.Visible = False  
lblStudentInfoReport.Visible = False  
panelEditTotalDay.Visible = True
```

```
End Select
```

```
End Sub
```

```
Private Sub frmMain_Resize(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Resize
```

```
panelStudentInfo.Location = New Point(10, MenuStrip1.Top +  
MenuStrip1.Height + 10)
```

```
panelStudentInfo.Size = New Size(Me.ClientSize.Width - 20,  
Me.ClientSize.Height - panelStudentInfo.Top - 10)
```

```
panelTimeRecord.Location = New Point(10, MenuStrip1.Top +  
MenuStrip1.Height + 10)
```

```
panelTimeRecord.Size = New Size(Me.ClientSize.Width - 20,  
Me.ClientSize.Height - panelTimeRecord.Top - 10)
```

```
panelSummaryReport.Location = New Point(10, +MenuStrip1.Top +  
MenuStrip1.Height + 10)
```

```
panelSummaryReport.Size = New Size(Me.ClientSize.Width - 20,  
Me.ClientSize.Height - panelSummaryReport.Top - 10)
```

```
panelEditTotalDay.Location = New Point(10, +MenuStrip1.Top +  
MenuStrip1.Height + 10)
```

```
panelEditTotalDay.Size = New Size(Me.ClientSize.Width - 20,  
Me.ClientSize.Height - panelSummaryReport.Top - 10)
```

```
grdSummaryReport.Location = New Point(10, +MenuStrip1.Top +  
MenuStrip1.Height + 10)
```

```
grdSummaryReport.Size = New Size(panelSummaryReport.Width - 20,  
panelSummaryReport.Height - grdSummaryReport.Top - 10)
```

```
lblSummaryReport.Location = New Point((((panelSummaryReport.Width - 20) / 2) -  
(lblSummaryReport.Width / 2), MenuStrip1.Height)
```

```
panelTimeReport.Location = New Point(10, +MenuStrip1.Top +  
MenuStrip1.Height + 10)
```

```
panelTimeReport.Size = New Size(Me.ClientSize.Width - 20,  
Me.ClientSize.Height - panelTimeReport.Top - 10)
```

```
grdReportTime.Location = New Point(10, +MenuStrip1.Top +
MenuStrip1.Height + 10)
```

```
grdReportTime.Size = New Size(panelTimeReport.Width - 20,
panelTimeReport.Height - grdReportTime.Top - 10)
```

```
lblTimeReport.Location = New Point((((panelSummaryReport.Width - 20) / 2) -
(lblSummaryReport.Width / 2), MenuStrip1.Height)
```

```
panelStudentReport.Location = New Point(10, +MenuStrip1.Top +
MenuStrip1.Height + 10)
```

```
panelStudentReport.Size = New Size(Me.ClientSize.Width - 20,
Me.ClientSize.Height - panelStudentReport.Top - 10)
```

```
grdStudentReport.Location = New Point(10, +MenuStrip1.Top +
MenuStrip1.Height + 10)
```

```
grdStudentReport.Size = New Size(panelStudentReport.Width - 20,
panelStudentReport.Height - grdStudentReport.Top - 10)
```

```
lblStudentInfoReport.Location = New Point((((panelSummaryReport.Width - 20) /
2) - (lblSummaryReport.Width / 2), MenuStrip1.Height)
```

```
End Sub
```

```
Private Sub TimeRecordEachPersonReport_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
```

```
TimeRecordEachPersonReport.Click
```

```
UpdateView(VIEW_TIME_REPORT)
```

```
Dim _objDBManage As New DBManage
```

```
Dim _objDataTable As New DataTable
```

```
_objDataTable = _objDBManage.GetDataTable("SELECT Contacts.FirstName,
Ident.idDate, MID(Ident.idTime, 1, 10) FROM Contacts INNER JOIN Ident ON
Contacts.sID = Ident.sID ORDER BY Contacts.FirstName;")
```

With grdReportTime

```
.DataSource = _objDataTable
.AllowUserToAddRows = False
.AllowUserToResizeColumns = False
.AllowUserToResizeRows = False
.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill
.AutoSizeRowsMode =
```

DataGridViewAutoSizeRowsMode.AllCellsExceptHeaders

```
.Columns(0).HeaderCell.Value = "ชื่อ"
.Columns(1).HeaderCell.Value = "วันที่เข้า"
.Columns(2).HeaderCell.Value = "เวลาเข้า"
.EditMode = DataGridViewEditMode.EditProgrammatically
.Font = New Font(grdReportTime.Font.Name, 10.0!)
.MultiSelect = False
.RowsDefaultCellStyle.WrapMode = DataGridViewTriState.True
.RowHeadersVisible = False
.SelectionMode = DataGridViewSelectionMode.FullRowSelect
```

End With

End Sub

```
Private Sub CallToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CallToolStripMenuItem.Click
```

```
System.Diagnostics.Process.Start("OEM2000P-44b0_SMP.exe")
TimeRecordEachPersonReportToolStripMenuItem_Click(sender, e)
```

End Sub

```
Private Sub grdStudentReport_CellDoubleClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
grdStudentReport.CellDoubleClick
```

```
UpdateView(VIEW_STUDENT_INFO)
```



```

gbStudentInfo.Text = "แก้ไขข้อมูลนักศึกษา"
btnAddStdData.Visible = False
btnUpdate.Visible = True
txtStudentId.Text =
grdStudentReport.Rows(e.RowIndex).Cells(1).Value.ToString()
txtStudentName.Text =
grdStudentReport.Rows(e.RowIndex).Cells(2).Value.ToString()
txtStudentLastName.Text =
grdStudentReport.Rows(e.RowIndex).Cells(3).Value.ToString()
sID = grdStudentReport.Rows(e.RowIndex).Cells(0).Value.ToString()
txtStudentId.Enabled = False

End Sub

Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUpdate.Click

    Dim _objDBAccess As New DBAccess
    Dim _boolError As Boolean

    Dim _strSQL As String = "UPDATE Contacts SET FirstName = " +
txtStudentName.Text.ToString() + ", LastName=" +
txtStudentLastName.Text.ToString() + " WHERE stuID =" +
txtStudentId.Text.ToString() + ""
    If txtStudentId.Text = "" Then
        MessageBox.Show("กรุณาป้อนรหัสนักศึกษา!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    ElseIf txtStudentName.Text = "" Then
        MessageBox.Show("กรุณาป้อนชื่อนักศึกษา!", "ผลการทำงาน",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    ElseIf txtStudentLastName.Text = "" Then

```

```

    MessageBox.Show("กรุณาป้อนนามสกุลนักศึกษา!", "ผลการทำงาน",
    MessageBoxButtons.OK, MessageBoxIcon.Information)
Else
    If MessageBox.Show("คุณต้องการแก้ไขข้อมูลนักศึกษาใหม่ ใช่หรือไม่?", "ยืนยัน
    การแก้ไขข้อมูลนักศึกษา", MessageBoxButtons.YesNo, MessageBoxIcon.Question) =
    System.Windows.Forms.DialogResult.Yes Then
        _boolError = _objDBAccess.ExecuteCommand(_strSQL)
        If _boolError = True Then
            MessageBox.Show("แก้ไขข้อมูลเรียบร้อยแล้ว!", "ผลการทำงาน",
            MessageBoxButtons.OK, MessageBoxIcon.Information)
        Else
            MessageBox.Show("ไม่สามารถแก้ไขข้อมูลได้!", "ผลการทำงาน",
            MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End If
End If
End Sub

Private Sub กำหนดความเร่งรทั้งหมดToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles กำหนดความเร่งรทั้งหมด
ToolStripMenuItem.Click
    UpdateView(VIEW_EDIT_TOTAL_DAY)

End Sub

Private Sub btnEditTotalDay_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEditTotalDay.Click
    Dim _objDBAccess As New DBAccess
    Dim _boolError As Boolean

```

```

If txtTotalDay.Text = "" Then
    MessageBox.Show("กรุณาป้อนจำนวนวัน", "ผลการทำงาน",
    MessageBoxButtons.OK, MessageBoxIcon.Information)
Else

    Dim _strSQL As String = "UPDATE Course SET TotalDay = " +
    txtTotalDay.Text.ToString() + ""

    If MessageBox.Show("คุณต้องการแก้ไขข้อมูลวันทั้งหมด ใช่หรือไม่?", "ยืนยันการ
    แก้ไขข้อมูลวันทั้งหมด", MessageBoxButtons.YesNo, MessageBoxIcon.Question) =
    System.Windows.Forms.DialogResult.Yes Then

        _boolError = _objDBAccess.ExecuteCommand(_strSQL)

        If _boolError = True Then

            MessageBox.Show("แก้ไขข้อมูลเรียบร้อยแล้ว!", "ผลการทำงาน",
            MessageBoxButtons.OK, MessageBoxIcon.Information)

        Else

            MessageBox.Show("ไม่สามารถแก้ไขข้อมูลได้!", "ผลการทำงาน",
            MessageBoxButtons.OK, MessageBoxIcon.Information)

        End If

    End If

End If

End Sub

Private Sub MenuStrip1_ItemClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ToolStripItemClickedEventArgs) Handles
MenuStrip1.ItemClicked

End Sub

End Class

```

## 7. โปรแกรมเพื่อติดต่อกับ Serial Port และกำหนดBaudrate (rs232)

```

/*****
/* File Name : RS232.cpp
/* Detail :
/* -----
/* History
/* Rev          Date          Detail
/* Rev.01  2006-09-01      Create
/*
/* Copyright (C) 2006 APRO TECHNOLOGY CO.,LTD All Rights Reserved.
/*****
/*****/
/* include          */
/*****/

#include "stdafx.h"
#include <windows.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include "rs232.h"

/*****/
/* Global Variable          */
/*****/

HANDLE          idComDev          = (HANDLE)0xFFFFFFFF;

//:=====
=====

int                                     //:RTN:
InitPort(                                     //:CAL:
    int          p_PortNum          //:IN :COM Port No.
,   DWORD          p_BaudRate          //:IN :BaudRate

```

```

, int                p_DataBit        //:IN :DataBit
, int                p_Parity         //:IN :Parity
, int                p_StopBit       //:IN :StopBit
){
    int                w_sts           = 0;
    BOOL              w_bsts          = FALSE;
    DWORD             w_lasterr;
    char               w_portstr[64];
    DCB                w_dcb;
    COMMTIMEOUTS      w_timeouts;

    wsprintf( w_portstr, "COM%d", p_PortNum );

    idComDev = CreateFile( w_portstr,
                           GENERIC_READ |
GENERIC_WRITE,
                           0,
                           NULL,
                           OPEN_EXISTING,
                           FILE_ATTRIBUTE_NORMAL,
                           NULL );

    if ( idComDev == INVALID_HANDLE_VALUE ) {
        w_sts = -1;
        goto L_EXIT;
    }

    w_bsts = GetCommState( idComDev, &w_dcb );
    if ( w_bsts != TRUE ) {
        w_sts = -2;
        goto L_EXIT;
    }
}

```

```
w_dcb.BaudRate           = p_BaudRate;
w_dcb.fBinary            = 1;
w_dcb.fOutxCtsFlow       = 0;
w_dcb.fOutxDsrFlow      = 0;
w_dcb.fDtrControl        = 0;
w_dcb.fDsrSensitivity    = 0;
w_dcb.fTXContinueOnXoff  = 0;
w_dcb.fOutX              = 0;
w_dcb.fInX               = 0;
w_dcb.fErrorChar         = 0;
w_dcb.fNull              = 0;
w_dcb.fRtsControl        = 0;
w_dcb.fAbortOnError      = 0;
w_dcb.fDummy2            = 0;//17;
w_dcb.XonLim             = 0;//100;
w_dcb.XoffLim            = 0;//100;
w_dcb.ByteSize           = p_DataBit;
w_dcb.Parity             = p_Parity;
w_dcb.StopBits           = p_StopBit;
w_dcb.XonChar            = 0;//0x11;
w_dcb.XoffChar           = 0;//0x13;
w_dcb.ErrorChar          = 0x00;
w_dcb.EofChar            = 0x00;
w_dcb.EvtChar            = 0x00;
w_bsts = SetCommState( hComDev, &w_dcb );
if ( w_bsts != TRUE ) {
    w_sts = -3;
    goto L_EXIT;
}
```

```
w_timeouts.ReadIntervalTimeout          = MAXDWORD;
w_timeouts.ReadTotalTimeoutMultiplier  = 0;
w_timeouts.ReadTotalTimeoutConstant    = 0;
w_timeouts.WriteTotalTimeoutMultiplier = 0;
w_timeouts.WriteTotalTimeoutConstant   = 3000;
```

```
w_bsts = SetCommTimeouts( idComDev, &w_timeouts );
```

```
if ( w_bsts != TRUE ) {
    w_sts = -4;
    goto L_EXIT;
}
```

```
w_bsts = SetupComm( idComDev, 4096, 4096 );
```

```
if ( w_bsts != TRUE ) {
    w_sts = -5;
    goto L_EXIT;
}
```

```
w_bsts = PurgeComm( idComDev, PURGE_TXCLEAR | PURGE_RXCLEAR );
```

```
if ( w_bsts != TRUE ) {
    w_sts = -6;
    goto L_EXIT;
}
```

```
w_bsts = SetCommMask( idComDev, EV_RXCHAR );
```

```
if ( w_bsts != TRUE ) {
    w_sts = -7;
    goto L_EXIT;
}
```

```
Sleep(200);
```

```

L_EXIT:
    if ( w_bsts != TRUE ) {
        w_lasterr = GetLastError();
        ClosePort();
    }
    return w_sts;
}

//:=*=====*=====*=====
=====

int                                     //:RTN:
PortWrite(                               //:CAL:
    LPSTR                                //:IN :
    p_pString,
) {
    int                                     w_sts = 0;
    BOOL                                    w_bsts;
    int                                     w_NumToWrite;
    DWORD                                  w_NumWrite;
    DWORD                                  w_lasterr;
    if ( idComDev == (HANDLE)0xFFFFFFFF ) {
        w_sts = -1;
        goto L_EXIT;
    }
    w_NumToWrite = strlen( p_pString );
    if ( w_NumToWrite == 0 ) {
        w_sts = -2;
        goto L_EXIT;
    }

    w_bsts = WriteFile( idComDev, p_pString, w_NumToWrite, &w_NumWrite, NULL
);

```



```

if ( w_bsts != TRUE ) {
    w_lasterr = GetLastError();
    w_sts = -3;
    goto L_EXIT;
}
else {
    w_sts = w_NumWrite;
}

L_EXIT:
    return w_sts;
}

////////////////////////////////////
//Use this for binary data that may include NULL characters
//:=====*:=====*
=====

int
PortWriteNum(
    LPSTR                p_pString        //:IN :
    , int                p_NumToWrite //:IN :
){
    int                w_sts;
    BOOL                w_bsts;
    DWORD                w_NumWrite;
    DWORD                w_lasterr;

    if ( idComDev == (HANDLE)0xFFFFFFFF ) {
        w_sts = -1;
        goto L_EXIT;
    }

    w_bsts = WriteFile( idComDev, p_pString, p_NumToWrite, &w_NumWrite, NULL
);

```

```

if ( w_bsts != TRUE ) {
    w_lasterr = GetLastError();
    w_sts = -2;
    goto L_EXIT;
}
else {
    w_sts = w_NumWrite;
}

L_EXIT:
    return w_sts;
}

//:=====
=====

int
PortRead(
    LPSTR                p_pString        //:IN :
,   int                 p_NumToRead     //:IN :
){
    int                 w_sts;
    BOOL                w_bsts;
    DWORD               w_NumRead;
    DWORD               w_lasterr;

    if ( idComDev == (HANDLE)0xFFFFFFFF ) {
        w_sts = -1;
        goto L_EXIT;
    }

    w_bsts = ReadFile( idComDev, p_pString, p_NumToRead, &w_NumRead, NULL
);

    if ( w_bsts != TRUE ) {
        w_lasterr = GetLastError();

```

```

        w_sts = -2;
        goto L_EXIT;
    }
    else {
        w_sts = w_NumRead;
    }
L_EXIT:
    return w_sts;
}

//:=====
=====

void
ClearInBuf(void)
{
    if ( idComDev == (HANDLE)0xFFFFFFFF )
        return;
    if (idComDev >= 0) {
        PurgeComm( idComDev, PURGE_TXABORT | PURGE_RXABORT |
                    PURGE_TXCLEAR |
                    PURGE_RXCLEAR );
    }
}

//:=====
=====

void
ClearReadBuf(void)
{
    if ( idComDev == (HANDLE)0xFFFFFFFF )
        return;
    if (idComDev >= 0) {
        PurgeComm( idComDev, PURGE_RXCLEAR );
    }
}

```

```

    }
}

//:=====
=====

void
ClosePort(
    void
){
    if ( idComDev == (HANDLE)0xFFFFFFFF )
        return;

    CloseHandle( idComDev );
    idComDev = (HANDLE)0xFFFFFFFF;
}

//:=====
=====

void
DisableHandshaking(
    void
){
    DCB w_dcb;
    GetCommState( idComDev, &w_dcb );
    w_dcb.fDtrControl = 0; /* DTR Flow control */
    w_dcb.fRtsControl = 0; /* Rts Flow control */
    SetCommState( idComDev, &w_dcb );
}

//:=====
=====

BOOL                                     //:RTN:
IsReceiveInQue(                          //:CAL:
    DWORD p_RcvCnt                       //:IN :
){

```

```

BOOL                w_bsts                = FALSE;
DWORD               w_error                = 0;
COMSTAT             w_ComStat;
int                 tt;
LARGE_INTEGER fr, st, et;
LONGLONG gap;
memset( &w_ComStat, 0, sizeof(w_ComStat) );

if( QueryPerformanceFrequency(&fr) == FALSE )    return FALSE;
if( QueryPerformanceCounter(&st) == FALSE )      return FALSE;

while(1)
{
    ClearCommError( idComDev, &w_error, &w_ComStat );
    if ( w_ComStat.cbInQue < p_RcvCnt ) {
        if( QueryPerformanceCounter(&et) == FALSE )    return
FALSE;

        gap = et.QuadPart - st.QuadPart;
        tt = (long)( gap / fr.QuadPart );
        if(tt > 10)
            return FALSE;

        //Sleep(5);
        continue;
    }
    else {
        break;
    }
}

w_bsts = ( w_ComStat.cbInQue >= p_RcvCnt ) ? TRUE : FALSE;

```

```

return w_bsts;
}
//.EOF

```

## 8. โปรแกรมเพื่อกำหนดค่าให้กับ Input (ARM7\_DEMODlg)

```

/*****
/* File Name : ARM7_DEMODlg.cpp
/* Detail :
/* -----
/* History
/* Rev      Date          Detail
/* Rev.01   2007-02-12    Create
/*
/* Copyright (C) 2007 APRO TECHNOLOGY(Bangkok) CO.,LTD All Rights
Reserved.
*****/

// ARM7_DEMODlg.cpp : implementation file
//
/*****/
/* include */
/*****/

#include "stdafx.h"
#include "ARM7_DEMO.h"
#include "ARM7_DEMODlg.h"
#include "SF_CalTime.h"
#include "rs232.h"
#include <iostream>
#include <fstream>
#include <time.h>

```

```

using namespace std;

#ifdef _DEBUG

#define new DEBUG_NEW

#undef THIS_FILE
static char THIS_FILE[] = __FILE__;

#endif

/*****
/*   Define                               */
*****/

#define LD_DATA_VALID                1
#define WM_KITMSG WM_USER+3
#define MULTI_PROCESS()

{

    MSG        msg;

    while ( PeekMessage(&msg, NULL, 0, 0, PM_REMOVE) ) {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }

}

/*****
/*   Global Variable                       */
*****/

static BYTE  Iv_ReadFPImage256[GD_FPIMA_W][GD_FPIMA_H];

```

```

static BYTE Iv_Temp1FP[GD_ENROLLDATASIZE];
static BYTE Iv_TempMaxFP[GD_ENROLLDATASIZE*GD_MAX_FPNUMBER];
static BYTE Iv_TempDB[GD_ENROLLDATASIZE*GD_MAX_USERS];
static ST_TIMECOUNT Iv_tc;

/*****
/*      Function      */
*****/

//:=*=====*=====*=====
=====

static
UINT
SS_MyThreadProc(
    LPVOID          ,      p_pParam
)
{
    CARM7_DEMODlg*   w_pObject      =
(CARM7_DEMODlg*)p_pParam;
    w_pObject->Identify();

    return TRUE;
}

//:=*=====*=====*=====
=====

static
UINT
SS_MyThreadProc1(
    LPVOID          ,      p_pParam
)
{
    CARM7_DEMODlg*   w_pObject      =
(CARM7_DEMODlg*)p_pParam;
    w_pObject->Verify();

    return TRUE;
}

```



```
////////////////////////////////////  
// CAboutDlg dialog used for App About  
  
class CAboutDlg : public CDialog  
{  
public:  
    CAboutDlg();  
  
    // Dialog Data  
   //{{AFX_DATA(CAboutDlg)  
    enum { IDD = IDD_ABOUTBOX };  
    //}}AFX_DATA  
  
    // ClassWizard generated virtual function overrides  
   //{{AFX_VIRTUAL(CAboutDlg)  
    protected:  
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV  
support  
    //}}AFX_VIRTUAL  
  
    // Implementation  
    protected:  
    ////{{AFX_MSG(CAboutDlg)  
    //}}AFX_MSG  
    DECLARE_MESSAGE_MAP()  
};  
CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)  
{  
    ////{{AFX_DATA_INIT(CAboutDlg)  
    //}}AFX_DATA_INIT  
}
```

```

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CAboutDlg)
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
   //{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////

// CARM7_DEMODlg dialog

CARM7_DEMODlg::CARM7_DEMODlg(CWnd* pParent /*=NULL*/)
    : CDialog(CARM7_DEMODlg::IDD, pParent)
{
    {{{AFX_DATA_INIT(CARM7_DEMODlg)
    m_bManager = FALSE;
    m_bBmpBinary = 0;
    m_strTime = _T("");
    m_strMsg = _T("");
    m_nTimeOut = 5;
    m_nSecurity = 1050;
    m_dwProductID = 1;
    m_nBaudRate = 0;
    m_nParity = 0;
    m_nStopBit = 0;
}}}

```

```
m_bDoubleCheck = TRUE;
m_strPrefix = _T("a");
m_lblFPNum = _T("");
//m_subID = 1;
//}}AFX_DATA_INIT
// Note that LoadIcon does not require a subsequent DestroyIcon in
Win32
m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
```

```
void CARM7_DEMODlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    ///{{AFX_DATA_MAP(CARM7_DEMODlg)
    DDX_Control(pDX, IDC_COMBO_COM, m_cmbCOM);
    DDX_Control(pDX, IDC_OPEN, m_btnOpen);
    DDX_Control(pDX, IDC_CLOSE, m_btnClose);
    DDX_Control(pDX, IDC_VERIFYIDFPDATA, m_btnVerifyIDFPData);
    DDX_Control(pDX, IDC_DEFAULT, m_btnDefault);
    DDX_Control(pDX, IDC_WRITEPID, m_btnWritePID);
    DDX_Control(pDX, IDC_READPID, m_btnReadPID);
    DDX_Control(pDX, IDC_SETTHRESHOLD, m_btnSetThreshold);
    DDX_Control(pDX, IDC_INPUTFPDATA, m_btnInputFPData);
    DDX_Control(pDX, IDC_IDENTIFYFPDATA, m_btnIdentifyFPData);
    DDX_Control(pDX, IDC_EnrollFPData, m_btnEnrollFPData);
    DDX_Control(pDX, IDC_VERIFYFPDATA, m_btnVerifyFPData);
    DDX_Control(pDX, IDC_CHANGEPORT, m_btnChangePort);
    DDX_Control(pDX, IDC_STOP, m_btnStop);
    DDX_Control(pDX, IDC_Verifing, m_btnVerifing);
    DDX_Control(pDX, IDC_ISPRESSFINGER, m_btnIsPressFinger);
    DDX_Control(pDX, IDC_IDENTIFYING, m_btnIdentifying);
```

```
DDX_Control(pDX, IDC_READFPDATAID, m_btnReadFPDataID);
DDX_Control(pDX, IDC_WRITEFPDATAID, m_btnWriteFPDataID);
DDX_Control(pDX, IDC_VERIFYIDIMAGE, m_btnVerifyIDImage);
DDX_Control(pDX, IDC_VERIFYID, m_btnVerifyID);
DDX_Control(pDX, IDC_SETTIMEOUT, m_btnSetTimeout);
DDX_Control(pDX, IDC_SEARCHID, m_btnSearchID);
DDX_Control(pDX, IDC_SEARCHFINGER, m_btnSearchFinger);
DDX_Control(pDX, IDC_CHECKMANAGER, m_btnCheckManager);
DDX_Control(pDX, IDC_CHECKID, m_btnCheckID);
DDX_Control(pDX, IDC_DELETEID, m_btnDeleteID);
DDX_Control(pDX, IDC_SPIN_DEVICEID, m_SpinDeviceID);
DDX_Control(pDX, IDC_EDIT_DEVICEID, m_EditDeviceID);
DDX_Control(pDX, IDC_WRITEFPDB, m_btnWriteFPDB);
DDX_Control(pDX, IDC_WRITEFPDATA, m_btnWriteFPData);
DDX_Control(pDX, IDC_VERIFYIMAGE, m_btnVerifyImage);
DDX_Control(pDX, IDC_VERIFY, m_btnVerify);
DDX_Control(pDX, IDC_SPIN_ID, m_SpinID);
DDX_Control(pDX, IDC_SPIN_FPNO, m_SpinFPNo);
DDX_Control(pDX, IDC_READFPDB, m_btnReadFPDB);
DDX_Control(pDX, IDC_READFPDATA, m_btnReadFPData);
DDX_Control(pDX, IDC_CHECKIDFINGER, m_btnCheckIDFinger);
DDX_Control(pDX, IDC_IMAGEWRITE, m_btnImgWrite);
DDX_Control(pDX, IDC_IMAGEREAD, m_btnImgRead);
DDX_Control(pDX, IDC_IDENTIFYIMAGE, m_btnIdentifyImg);
DDX_Control(pDX, IDC_IDENTIFY, m_btnIdentify);
DDX_Control(pDX, IDC_ENROLLIMAGE, m_btnEnrollImg);
DDX_Control(pDX, IDC_ENROLLCOUNT, m_btnEnrollCount);
DDX_Control(pDX, IDC_ENROLL, m_btnEnroll);
DDX_Control(pDX, IDC_EDIT_ID, m_EditID);
DDX_Control(pDX, IDC_EDIT_FPNO, m_EditFPNo);
DDX_Control(pDX, IDC_DELETEALL, m_btnDeleteAll);
```

```

DDX_Control(pDX, IDC_DELETE, m_btnDelete);
DDX_Control(pDX, IDC_CAPTUREIMG, m_btnCaptureImg);
DDX_Control(pDX, IDC_FRAME, m_Frame);
DDX_Check(pDX, IDC_CHECK_MANAGER, m_bManager);
DDX_Radio(pDX, IDC_RADIO_BMP, m_bBmpBinary);
DDX_Text(pDX, IDC_TIME, m_strTime);
DDX_Text(pDX, IDC_MESSAGE, m_strMsg);
DDX_Text(pDX, IDC_EDIT_TIMEOUT, m_nTimeOut);
DDX_Text(pDX, IDC_EDIT_SECURITY, m_nSecurity);
DDX_Text(pDX, IDC_EDIT_PRODUCTID, m_dwProductID);
DDX_Radio(pDX, IDC_RADIO1, m_nBaudRate);
DDX_Radio(pDX, IDC_RADIO4, m_nParity);
DDX_Radio(pDX, IDC_RADIO7, m_nStopBit);
DDX_Check(pDX, IDC_DOUBLECHECK, m_bDoubleCheck);
DDX_Text(pDX, IDC_FPENROLLMENTNUM, m_lbIFPNum);
//DDX_Text(pDX, IDC_EDIT1, m_subID);
//}}AFX_DATA_MAP
}

```

```

BEGIN_MESSAGE_MAP(CARM7_DEMODlg, CDialog)
//{{AFX_MSG_MAP(CARM7_DEMODlg)
ON_WM_SYSCOMMAND()
ON_WM_PAINT()
ON_WM_QUERYDRAGICON()
ON_BN_CLICKED(IDC_OPEN, OnOpen)
ON_BN_CLICKED(IDC_CLOSE, OnClose)
ON_BN_CLICKED(IDC_IMAGEWRITE, OnImagewrite)
ON_BN_CLICKED(IDC_IMAGEREAD, OnImageread)
ON_BN_CLICKED(IDC_ENROLL, OnEnroll)
ON_BN_CLICKED(IDC_IDENTIFY, OnIdentify)
ON_BN_CLICKED(IDC_VERIFY, OnVerify)

```

ON\_BN\_CLICKED(IDC\_DELETE, OnDelete)  
ON\_BN\_CLICKED(IDC\_DELETEALL, OnDeleteall)  
ON\_BN\_CLICKED(IDC\_ENROLLCOUNT, OnEnrollcount)  
ON\_BN\_CLICKED(IDC\_CHECKIDFINGER, OnCheckIDFinger)  
ON\_BN\_CLICKED(IDC\_CAPTUREIMG, OnCaptureimg)  
ON\_BN\_CLICKED(IDC\_ENROLLIMAGE, OnEnrollimage)  
ON\_BN\_CLICKED(IDC\_IDENTIFYIMAGE, OnIdentifyimage)  
ON\_BN\_CLICKED(IDC\_VERIFYIMAGE, OnVerifyimage)  
ON\_BN\_CLICKED(IDC\_WRITEFPDATA, OnWritefpdata)  
ON\_BN\_CLICKED(IDC\_READFPDATA, OnReadfpdata)  
ON\_BN\_CLICKED(IDC\_WRITEFPDB, OnWritefpdb)  
ON\_BN\_CLICKED(IDC\_READFPDB, OnReadfpdb)  
ON\_BN\_CLICKED(IDC\_SETTIMEOUT, OnSettimeout)  
ON\_BN\_CLICKED(IDC\_VERIFYID, OnVerifyid)  
ON\_BN\_CLICKED(IDC\_SEARCHID, OnSearchid)  
ON\_BN\_CLICKED(IDC\_SEARCHFINGER, OnSearchfinger)  
ON\_BN\_CLICKED(IDC\_CHECKID, OnCheckID)  
ON\_BN\_CLICKED(IDC\_CHECKMANAGER, OnCheckManager)  
ON\_BN\_CLICKED(IDC\_DELETEID, OnDeleteid)  
ON\_BN\_CLICKED(IDC\_VERIFYIDIMAGE, OnVerifyidimage)  
ON\_BN\_CLICKED(IDC\_WRITEFPDATAID, OnWritefpdataid)  
ON\_BN\_CLICKED(IDC\_READFPDATAID, OnReadfpdataid)  
ON\_BN\_CLICKED(IDC\_ISPRESSFINGER, OnIspressfinger)  
ON\_BN\_CLICKED(IDC\_Verifing, OnVerifing)  
ON\_BN\_CLICKED(IDC\_IDENTIFING, OnIdentifing)  
ON\_BN\_CLICKED(IDC\_STOP, OnStop)  
ON\_BN\_CLICKED(IDC\_CHANGEPORT, OnChangeport)  
ON\_BN\_CLICKED(IDC\_INPUTFPDATA, OnInputfpdata)  
ON\_BN\_CLICKED(IDC\_EnrollFPData, OnEnrollFPData)  
ON\_BN\_CLICKED(IDC\_IDENTIFYFPDATA, OnIdentifyfpdata)  
ON\_BN\_CLICKED(IDC\_VERIFYFPDATA, OnVerifyfpdata)

```

ON_BN_CLICKED(IDC_SETTHRESHOLD, OnSetThreshold)
ON_BN_CLICKED(IDC_WRITEPID, OnWritepid)
ON_BN_CLICKED(IDC_READPID, OnReadpid)
ON_BN_CLICKED(IDC_DEFAULT, OnDefault)
ON_MESSAGE(WM_KITMSG, OnKitMsg)
ON_BN_CLICKED(IDC_VERIFYIDFPDATA, OnVerifyidfpdata)
ON_BN_CLICKED(IDC_RADIO1, OnRadio1)
ON_BN_CLICKED(IDC_RADIO2, OnRadio2)
ON_BN_CLICKED(IDC_RADIO4, OnRadio4)
ON_BN_CLICKED(IDC_RADIO5, OnRadio5)
ON_BN_CLICKED(IDC_RADIO6, OnRadio6)
ON_BN_CLICKED(IDC_RADIO7, OnRadio7)
ON_BN_CLICKED(IDC_RADIO8, OnRadio8)
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CARM7_DEMODlg message handlers

BOOL CARM7_DEMODlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)

```

```
{
    CString strAboutMenu;
    strAboutMenu.LoadString(IDS_ABOUTBOX);
    if (!strAboutMenu.IsEmpty())
    {
        pSysMenu->AppendMenu(MF_SEPARATOR);
        pSysMenu->AppendMenu(MF_STRING,
IDM_ABOUTBOX, strAboutMenu);
    }
}

// Set the icon for this dialog. The framework does this automatically
// when the application's main window is not a dialog
SetIcon(m_hIcon, TRUE);           // Set big icon
SetIcon(m_hIcon, FALSE);        // Set small icon

//.Init controll
//.ID
m_SpinID.SetBuddy(&m_EditID);
m_SpinID.SetRange32(1,100000);
m_SpinID.SetPos(1);
//.Finger Number
m_SpinFPNo.SetBuddy(&m_EditFPNo);
m_SpinFPNo.SetRange(1,10);
m_SpinFPNo.SetPos(1);
//.Device ID
m_SpinDeviceID.SetBuddy(&m_EditDeviceID);
m_SpinDeviceID.SetRange(1,16);
m_SpinDeviceID.SetPos(1);
//.Button
SS_EnableBtn(FALSE);
```



```
m_btnOpen.EnableWindow(TRUE);
//.Combo Box
char          w_strCom[8];
int           i;
for( i=0; i<10; i++){
    sprintf( w_strCom, "COM%d", i+1 );
    m_cmbCOM.AddString( w_strCom );
    m_cmbCOM.SetCurSel( 0 );
}
m_cWinThread = NULL;
m_bContinue = FALSE;
m_IsOpen = FALSE;
return TRUE; // return TRUE unless you set the focus to a control
}
```

```
void CARM7_DEMODlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFFF) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}
```

```
// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.
```

```
void CARM7_DEMODlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (WPARAM)
dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        //CRect rect;
        //m_Frame.GetClientRect(&rect);
        m_DIB.Draw(m_Frame.GetDC());
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the user drags
```

```

// the minimized window.
HCURSOR CARM7_DEMODlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

//:=====
=====

BOOL
CARM7_DEMODlg::
PreTranslateMessage(
    MSG*                pMsg
)
{
    switch(pMsg->message){
case WM_KEYDOWN:
    switch(pMsg->wParam){
        case VK_ESCAPE: return TRUE;
        case VK_RETURN: return TRUE;
    }
    }

    return CDialog::PreTranslateMessage(pMsg);
}

//:=====
=====

```

## 6.1 การเปิดพอร์ต

```

void
CARM7_DEMODlg::
OnOpen(
    BOOL                w_bsts;

```

```
int w_nInParam;
int w_COM =
m_cmbCOM.GetCurSel()+1;
//FILE
ofstream myfile;
myfile.open ("finger.txt",ios::ate);
myfile.close();

ClosePort();
UpdateData();
if(m_cWinThread != NULL){
    TerminateThread(m_cWinThread->m_hThread, 0);
    ::WaitForSingleObject(m_cWinThread->m_hThread,INFINITE);
}
m_bContinue = FALSE;
//.Open COM port
w_bsts = m_ClaOEMCtrl.OpenComPort( w_COM, (BYTE)m_nBaudRate,
(BYTE)m_nParity, (BYTE)m_nStopBit );
if ( w_bsts != TRUE ){
    MessageBeep( MB_ICONHAND );
    m_strMsg.Format("Open COM Port%d Failed.", w_COM );
    UpdateData(FALSE);
    goto L_EXIT;
}
else{
    m_strMsg.Format( "Open COM Port%d OK!", w_COM );
}
w_nInParam = ( 1 << 16 );
SF_StartTime(&lv_tc);
//.Open the device
```

```

        w_bsts = m_ClaOEMCtrl.Open( m_SpinDeviceID.GetPos(), w_nInParam
);
        SF_EndTime(&lv_tc);
        if( w_bsts != TRUE ){
            m_strMsg.Format( "Error : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError() ) );
            MessageBeep( MB_ICONHAND );
        }
        else{
            MessageBeep(MB_ICONASTERISK);
            m_strMsg.Format("เปิดใช้งานอุปกรณ์แล้ว!");
            SS_EnableBtn(TRUE);
        }
        m_nSecurity = 1050;
        m_IsOpen = TRUE;
        //View FP enrollment number
        OnEnrollcount();
        m_strTime.Format( "%d.%03ld ms", lv_tc.Sec, lv_tc.MiliSec );
        UpdateData(FALSE);
L_EXIT:
        return;
    }

```

## 6.2 การปิดพอร์ต

```

void
CARM7_DEMODlg::
OnClose(
){
    if(m_cWinThread != NULL){
        TerminateThread(m_cWinThread->m_hThread, 0);
    }
}

```

```

        ::WaitForSingleObject(m_cWinThread->m_hThread,INFINITE);
    }
    if( !m_ClaOEMCtrl.Close() ){
        m_strMsg.Format( "Error : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError() ) );
        MessageBeep( MB_ICONHAND );
        goto L_EXIT;
    }
    if ( !m_ClaOEMCtrl.CloseComPort() ){
        MessageBeep( MB_ICONHAND );
        m_strMsg.Format( "Error : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError() ) );
        goto L_EXIT;
    }
    else{
        MessageBeep(MB_ICONASTERISK);
        m_strMsg = "ปิดพอร์ทเรียบร้อยแล้ว!";
    }
    m_IsOpen = FALSE;
L_EXIT:
    SS_EnableBtn(FALSE);
    m_btnOpen.EnableWindow();
    UpdateData(FALSE);
}

```

### 6.3 การกดปุ่ม Enroll แล้วพิมพ์ลายนิ้วมือ 3 ครั้ง

```

void
CARM7_DEMODlg::
OnEnroll(
){

```

```

        gstUSERINFO          wgstUSERINFO;
        DWORD                w_dwUserInfo;
        DWORD                w_Endtime;
        BOOL                 w_bsts;
int    i;

        SS_EnableBtn( FALSE );
        UpdateData();
        SS_MakeUserInfo( &w_dwUserInfo );
        if( !m_bDoubleCheck ){
                w_dwUserInfo &= 0x0FFFFFFF;
        }
        w_bsts = m_ClaOEMCtrl.EnrollStartN( w_dwUserInfo );
        if( w_bsts != TRUE ){
                m_strMsg.Format( "ผิดพลาด : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError() ) );
                MessageBeep( MB_ICONHAND );
                goto L_EXIT;
        }
        for( i=0; i<3; i++ ){
                m_strMsg.Format( "#%d : กรุณาวางนิ้วลงบนแถบอ่านลายนิ้วมือ...",
i+1 );

                UpdateData( FALSE );
                w_Endtime = GetTickCount() + m_nTimeOut *1000 ;
                for( ;; ) {
                        w_bsts = m_ClaOEMCtrl.CaptureImage();
                        if( w_bsts == TRUE ){
                                break;
                        }
                        else if( m_ClaOEMCtrl.GetLastError() != IMAGE_ERR ){
                                m_strMsg.Format( "ผิดพลาด : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError() ) );

```

```

        MessageBeep( MB_ICONHAND );
        goto L_EXIT;
    }

    if ( GetTickCount() > w_Endtime ) {
        break;
    }
}

if( w_bsts != TRUE ){
    MessageBeep( MB_ICONHAND );
    m_strMsg = "หมดเวลา";
    goto L_EXIT;
}

else{
    MessageBeep( MB_ICONASTERISK );
    m_strMsg = "นำนิ้วออกจากเครื่องอ่าน!";
    UpdateData( FALSE );
}

w_bsts = m_ClaOEMCtrl.EnrollProNImage( i+1 );

if( w_bsts != TRUE ){
    m_strMsg.Format( "ผิดพลาด : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError() ) );
    MessageBeep( MB_ICONHAND );
    goto L_EXIT;
}

//.Take off finger
do{
    w_bsts = m_ClaOEMCtrl.IsPressFinger();
    if( w_bsts != TRUE
    && m_ClaOEMCtrl.GetLastError() != IMAGE_ERR ){
        m_strMsg.Format( "ผิดพลาด : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError() ) );

```



```

        MessageBeep( MB_ICONHAND );
        goto L_EXIT;
    }
    Sleep( 10 );
} while( w_bsts == TRUE );
} //for
SF_StartTime( &lv_tc );
w_bsts = m_ClaOEMCtrl.EnrollEndN();
SF_EndTime( &lv_tc );
if( w_bsts == TRUE ){
    MessageBeep( MB_ICONASTERISK );
    m_strMsg.Format("Enroll OK!! ID = %d, Finger No. = %d",
m_SpinID.GetPos(), m_SpinFPNo.GetPos() );
    SS_UpdateUser();
}
else if( m_ClaOEMCtrl.GetLastError() > GD_ERR_FPDOUBLE ){
    m_ClaOEMCtrl.ReleaseUserInfoParam( &wgstUSERINFO,
m_SpinID.GetPos() );
    MessageBeep( MB_ICONASTERISK );
    m_strMsg.Format(
        "The finger is already enrolled! Manager = %d, ID = %d,
FingerNo = %d"
        , wgstUSERINFO.Manager
        , wgstUSERINFO.UserID, wgstUSERINFO.FP_NO
    );
}
else{
    m_strMsg.Format( "Error : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError() ) );
    MessageBeep( MB_ICONHAND );
}

```

```

//View FP enrollment number
OnEnrollcount();
m_strTime.Format( "%d.%03ld ms", lv_tc.Sec, lv_tc.MilliSec );
L_EXIT:
UpdateData( FALSE );
SS_EnableBtn( TRUE );
return;
}

```

#### 6.4 การตรวจสอบลายนิ้วมือ

```

void
CARM7_DEMODlg::
OnIdentify(
){
    gstUSERINFO          wgstUSERINFO;
    BOOL                  w_bsts;
    DWORD                 w_Endtime;
    time_t date; // Make a time_t object that'll hold the date
    time(&date); // Set the date variable to the current date
    //FILE
    ofstream myfile;

    SS_EnableBtn( FALSE );
    UpdateData();
    m_strMsg = "กรุณาวางนิ้วลงบนแถบอ่านลายนิ้วมือ...";
    UpdateData( FALSE );
    w_Endtime = GetTickCount() + m_nTimeOut *1000 ;
    for( ;; ) {
        w_bsts = m_ClaOEMCtrl.CaptureImage();
    }
}

```

```

        if( w_bsts == TRUE ){
            m_strMsg = "นำนิ้วออกจากเครื่องอ่าน!";
            MessageBeep( MB_ICONASTERISK );
            PostMessage( WM_KITMSG,0,0 );
            break;
        }
        else if( m_ClaOEMCtrl.GetLastError() != IMAGE_ERR ){
            m_strMsg.Format( "ผิดพลาด : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError()) );
            MessageBeep( MB_ICONHAND );
            goto L_EXIT;
        }
        if ( GetTickCount() > w_Endtime ){
            break;
        }
    }
    if( w_bsts != TRUE ){
        MessageBeep( MB_ICONHAND );
        m_strMsg = "Time Out";
        UpdateData( FALSE );
        SS_EnableBtn( TRUE );
        goto L_EXIT;
    }
    SF_StartTime( &lv_tc );
    w_bsts = m_ClaOEMCtrl.IdentifyImage( &wgstUSERINFO );
    SF_EndTime( &lv_tc );
    if( w_bsts == TRUE ) {
        myfile.open ("finger.txt",ios::app);
        myfile << wgstUSERINFO.UserID << " " << ctime(&date);
        myfile.close();
        MessageBeep( MB_ICONASTERISK );
    }

```

```

        m_strMsg.Format( "OK! Manager = %d, ID = %d, FingerNo =
%d"
                        , wgstUSERINFO.Manager
                        , wgstUSERINFO.UserID
                        , wgstUSERINFO.FP_NO );
    }
    else if( m_ClaOEMCtrl.GetLastError() == GD_ERR_FPNOTIDENTIFY ){
        m_strMsg = "ไม่พบลายนิ้วมือ!!";
        MessageBeep( MB_ICONHAND );
    }
    else{
        m_strMsg.Format( "ผิดพลาด : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError()) );
        MessageBeep( MB_ICONHAND );
    }
    m_strTime.Format("%d.%03ld ms", lv_tc.Sec, lv_tc.MilliSec);
    UpdateData( FALSE );
    SS_EnableBtn( TRUE );
L_EXIT:
    return;
}

```

### 6.5 การลบข้อมูลลายนิ้วมือจากเครื่องอ่านลายนิ้วมือ

```

void
CARM7_DEMODlg::
OnDelete(
){
    DWORD                w_dwUserInfo;
    BOOL                 .w_bsts;
    SS_EnableBtn( FALSE );
}

```

```

        UpdateData();
        SS_MakeUserInfo( &w_dwUserInfo );
        SF_StartTime( &lv_tc );
        w_bsts = m_ClaOEMCtrl.DeleteIDFinger( w_dwUserInfo );
        SF_EndTime( &lv_tc );
        if( w_bsts != TRUE ) {
            m_strMsg.Format( "Error : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError()) );
            MessageBeep( MB_ICONHAND );
        }
        else {
            MessageBeep( MB_ICONASTERISK );
            m_strMsg.Format( "Delete OK!\nID=%d,FingerNo=%d:"
                                , m_SpinID.GetPos()
                                , m_SpinFPNo.GetPos()
                                );
        }
        //View FP enrollment number
        OnEnrollcount();
        m_strTime.Format( "%d.%03ld ms", lv_tc.Sec, lv_tc.MiliSec );
        UpdateData( FALSE );
        SS_EnableBtn( TRUE );
    }
    //:=*=====*=====*=====
    =====
    void
    CARM7_DEMODlg::
    OnDeleteall(
    ){
        BOOL
        , w_bsts;
        SS_EnableBtn( FALSE );

```

```

SF_StartTime( &lv_tc );
w_bsts = m_ClaOEMCtrl.DeleteAll();
SF_EndTime( &lv_tc );
if( w_bsts != TRUE ) {
    MessageBeep( MB_ICONHAND );
    m_strMsg.Format( "Error : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError()) );
}
else {
    MessageBeep( MB_ICONASTERISK );
    m_strMsg = "DeleteAll OK!";
}
//.View FP enrollment number
OnEnrollcount();
m_strTime.Format( "%d.%03ld ms", lv_tc.Sec, lv_tc.MiliSec );
UpdateData( FALSE );
SS_EnableBtn( TRUE );
}

```

#### 6.6 การหมดเวลาพิมพ์ลายนิ้วมือ (5 วินาที)

```

void
CARM7_DEMODlg::
OnSettimeout(
){
    UpdateData();
    MessageBeep( MB_ICONASTERISK );
    m_strMsg = "Set TimeOut OK!";
    m_strTime = "";
    UpdateData( FALSE );
}

```

## 6.7 โปรแกรมอ่านลายนิ้วมือต้องตั้งค่าจุดบนลายนิ้วมือ

```
void
CARM7_DEMODlg::
OnSetThreshold(
){
    int                w_bsts;
    SS_EnableBtn(FALSE);
    UpdateData( TRUE );
    SF_StartTime(&lv_tc);
    w_bsts = m_ClaOEMCtrl.SetThreshold( m_nSecurity );
    SF_EndTime(&lv_tc);
    if( w_bsts != TRUE ){
        m_strMsg.Format( "Error : %s",
m_ClaOEMCtrl.GetLastErrString(m_ClaOEMCtrl.GetLastError()) );
        MessageBeep( MB_ICONHAND );
    }
    else{
        MessageBeep( MB_ICONASTERISK );
        m_strMsg = "SetThreshold OK!";
    }
    m_strTime.Format( "%d.%03ld ms", lv_tc.Sec, lv_tc.MilliSec );
    UpdateData( FALSE );
    SS_EnableBtn( TRUE );
}
```

## ภาคผนวก ข

### การใช้งานและการดูแลรักษาเครื่องอ่านลายนิ้วมือ

#### การใช้งานเครื่องอ่านลายนิ้วมืออย่างถูกวิธี

##### 1. พยายามแตะให้กว้างครอบคลุมพื้นที่มากที่สุด

ในการแตะอุปกรณ์เพื่อลงทะเบียนหรือลงเวลา ผู้ใช้ควรวางปลายนิ้วไว้ที่ส่วนบนสุดของกระจกรับภาพของอุปกรณ์ และแตะนิ้วลงไปให้สัมผัสกระจกให้ครอบคลุมพื้นที่มากที่สุด เพราะอุปกรณ์จะอ่านลายนิ้วมือจากส่วนที่สัมผัสกับกระจกเท่านั้น ดังนั้นยิ่งผู้ใช้แตะพื้นผิวได้กว้างเท่าใด ก็จะเป็นการดีสำหรับการประมวลผลข้อมูลลายนิ้วมือได้อย่างแม่นยำ

##### 2. วางนิ้วให้ขนานกับแนวราบ

บ่อยครั้งที่ผู้ใช้อุปกรณ์จะแตะอุปกรณ์ในลักษณะเดียวกับการกดปุ่ม เหมือนกับการที่นิ้วลงไปอยู่ที่อุปกรณ์ ซึ่งเป็นลักษณะการใช้งานที่ไม่ถูกต้อง เพราะในลักษณะนั้นส่วนปลายนิ้วเท่านั้นที่สัมผัสกับกระจก ซึ่งจะทำให้ลายนิ้วมือที่อ่านได้มีคุณภาพต่ำ ลักษณะที่ถูกต้องในการสัมผัส อุปกรณ์จะต้องวางนิ้วขนานกับแนวราบลงไปกับกระจกรับภาพของอุปกรณ์

##### 3. ไม่ควรกดแรงเกินไป

ลายนิ้วมือที่อุปกรณ์อ่านได้นั้นมีที่มาจากรายละเอียดและความตื้นลึกบนพื้นผิวของลายนิ้วมือของบุคคลนั้นๆ การกดอุปกรณ์แรงเกินไปจะทำให้ลายนิ้วมือที่อ่านได้ไม่แม่นยำและถูกต้องน้อยลง ในทางกลับกันหากแตะเบาเกินไป จะทำให้นิ้วสัมผัสกับอุปกรณ์น้อยเกินกว่าจะอ่านข้อมูลได้ การใช้งานที่ถูกต้องคือผู้ใช้จะต้องวางนิ้วมือลงกับอุปกรณ์ให้พอดีและกดลงไปเบาๆ

##### 4. เล็บยาว

บ่อยครั้งที่ผู้ใช้ที่มีเล็บยาวจะมีปัญหากับการใช้เครื่องอ่านลายนิ้วมือ เนื่องจากผู้ใช้จะใช้ปลายเล็บวางไว้ที่ส่วนบนสุดของกระจกรับภาพ แทนที่จะเป็นปลายนิ้ว ซึ่งวิธีการที่ถูกต้องคือ จะต้องวางนิ้วให้เล็บเลยกระจกรับภาพไปทางด้านบน และให้ปลายนิ้วส่วนที่เป็นเนื้อวางอยู่ที่ขอบบนของกระจกพอดี

##### 5. อ่าหมนิ้ว

ในบางครั้งผู้ใช้อาจจะคิดว่าการที่อุปกรณ์จะอ่านลายนิ้วมือได้ดีจะต้องกดแล้วเอียงนิ้วมือซ้าย-ขวาในลักษณะของการพิมพ์ลายนิ้วมือบนกระดาษ ซึ่งเป็นวิธีการที่ไม่แนะนำอย่างยิ่ง เนื่องจากการกดในลักษณะเช่นนั้นจะทำให้ข้อมูลที่ได้ไม่มีความแม่นยำเท่าที่ควร และจะไม่ได้ผลที่ดีกว่าเมื่อเทียบกับการกดนิ้วลงไปนิ่งๆบนกระจกรับภาพ



## 6. มือแห้ง

ถึงแม้ว่าเราจะพบปัญหานี้น้อยมากๆ อย่างไรก็ตามในบางกรณีความชื้นที่อยู่บนนิ้วมือก็เป็นส่วนที่สำคัญอย่างมากสำหรับการอ่านลายนิ้วมือ ผู้ใช้ที่มีนิ้วมือที่แห้งมากเกินไปอาจมีปัญหาในการใช้อุปกรณ์ ซึ่งวิธีการแก้ปัญหาสำหรับผู้ใช้คือ ให้ผู้ใช้ใช้นิ้วถูกับฝ่ามืออีกข้างสักครู่ จนรู้สึกว่ามีผิวหนังขึ้นอยู่บ้าง และลองกดดูอีกครั้งหนึ่ง

### การป้องกันปัญหาที่อาจเกิดขึ้น

#### 1. มือสกปรก

กรุณาตรวจสอบให้แน่ใจว่านิ้วมือของท่านสะอาดและไม่เปียกน้ำก่อนการใช้งานทุกครั้ง ไม่ควรมีฝุ่นสกปรกที่อาจหลงเหลือติดอยู่บนเครื่องอ่านลายนิ้วมือ

#### 2. แสงแดด

การวางเครื่องอ่านลายนิ้วมือไว้ในที่มีแสงแดดจ้า อาจรบกวนการทำงานของเครื่องอ่านลายนิ้วมือได้

#### 3. แอลกอฮอล์และน้ำยาทำความสะอาด

แอลกอฮอล์และน้ำยาทำความสะอาด จะทำลายแผ่นฟิล์มบางๆบนเครื่องอ่านลายนิ้วมือ

#### 4. น้ำตาล

น้ำตาลที่อยู่บนนิ้วมือเมื่อสัมผัสกับเหงื่อ จะกัดกร่อนแผ่นฟิล์มบางๆบนเครื่องอ่านลายนิ้วมือ

## การทำความสะอาดเครื่องอ่านลายนิ้วมือ

การทำความสะอาดจะขึ้นกับปริมาณการใช้งานเครื่องอ่านลายนิ้วมือ อาจต้องมีการทำความสะอาดบ้าง เนื่องจากฝุ่นละอองและเหงื่อที่อาจตกหล่นอยู่จนทำให้ช่องอ่านลายนิ้วมือมีไว้

### วิธีที่แนะนำในการทำความสะอาด

ในการทำความสะอาดฝุ่นละอองที่ตกหล่นอยู่ ให้ใช้สก็อตเทปชิ้นเล็กๆ ด้านที่มีกาวเหนียวติดลงบนช่องอ่านลายนิ้วมือแล้วดึงออก



รูปที่ 15 แสดงวิธีทำความสะอาด

ถ้าต้องการเช็ดทำความสะอาด ต้องใช้ผ้านุ่มที่ไม่มีขุยหลุดออกมาแต่น้ำหรือน้ำยาเช็ดกระจกที่เป็นส่วนผสมของแอมโมเนียเช็ดเบาๆ

### ข้อควรระวังในการทำความสะอาด

อย่าใช้กระดาษเช็ดช่องอ่านลายนิ้วมือ อย่าจุ่มหรือเทน้ำลงบนเครื่องอ่านลายนิ้วมือ และแอลกอฮอล์จะทำลายฟิล์มบางๆบนเครื่องอ่านลายนิ้วมือได้

## เอกสารอ้างอิง

- [1.] พร้อมเลิศ หล่อวิจิตร.คู่มือเรียน visual basic 2005.กรุงเทพฯ. โปรวิชั่น, 2549
- [2.] สัจจะ จรัสรุ่งรวีวร.คู่มือเรียน visual basic 2005 ฉบับสมบูรณ์.นนทบุรี:2549
- [3.] ณาตยา ฉาบนาค.Microsoft Access 2003.กรุงเทพฯ: เอส.พี.ซี.
- [4.] ANIL K. JAIN and SHARATHCHANDRA PANKANTI, สรรพฤทธิ์ มฤคทัต. วารสารเทคโนโลยีโทรคมนาคม. กุมภาพันธ์ 2550, หน้า 9-15.
- [5.] นายวิชโรตม ประระมะ, นางสาววาทีณี สุมาลัย, นายอนุวัฒน์ นราวัฒน์.โครงการตรวจสอบรายชื่อด้วยเครื่องอ่านลายนิ้วมือ: ปีการศึกษา 2549

## ประวัติผู้เขียน

นางสาวจุฬาลักษณ์ เข้มสว่าง เกิดวันที่ 19 พฤษภาคม พ.ศ.2529 ภูมิลำเนาอยู่บ้านเลขที่ 259/1 ถนนลำราญราษฎร์ ตำบลโนนสูง อำเภอโนนสูง จังหวัดนครราชสีมา จบการศึกษามัธยมศึกษาตอนต้นและตอนปลายจากโรงเรียนโนนสูงศรีธานี ปีการศึกษา 2546 ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา

นายโกวิท ภิรมย์กิจ เกิดวันที่ 24 มีนาคม พ.ศ. 2529 ภูมิลำเนาอยู่บ้านเลขที่ 408 หมู่ที่ 14 ตำบลศิลา อำเภอเมืองขอนแก่น จังหวัดขอนแก่น จบการศึกษามัธยมศึกษาตอนต้นและตอนปลายจากโรงเรียนกัลยาณวัตร ปีการศึกษา 2546 ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา

นางสาวนุชนาฏ ฝาเพ็ญ เกิดวันที่ 8 มิถุนายน 2529 ภูมิลำเนาอยู่บ้านเลขที่ 24 หมู่ที่ 1 ตำบลเขาน้อย อำเภอลำสนธิ จังหวัดลพบุรี จบการศึกษามัธยมศึกษาตอนต้นและตอนปลายจากโรงเรียนมารีย์วิทยา นครราชสีมา ปีการศึกษา 2546 ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา

