

อธิบดีมหาวิทยาลัย



สำนักหอสมุด



ตู้นิรภัยสแกนลายนิ้วมือ

THE FINGERPRINT-SCAN SAFE

นายปรีวรรต วงศ์วรรณทร รหัส 51383959
นายเทอดศักดิ์ ทองสุข รหัส 51384864

สำนักหอสมุด มหาวิทยาลัยพระจอมเกล้าธนบุรี
วันลงทะเบียน 24 ส.ค. 2561
เลขทะเบียน 19221009
เลขเรียกหนังสือ ปร

ปี 4617
2554

ปฏิญานี้พินธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยพระจอมเกล้าธนบุรี

ปีการศึกษา 2554

ชื่อหัวข้อโครงการ	ผู้นิรภัยสแกนลายนิ้วมือ
ผู้ดำเนินโครงการ	นายปวิวรรต วงศ์วรรณตร รหัส 51383959
	นายเทอดศักดิ์ ทองสุข รหัส 51384864
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร. ยงยุทธ ชนบดีเฉลิมรุ่ง
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

บทคัดย่อ

โครงการนี้นำเสนอผู้นิรภัยสแกนลายนิ้วมือ โดยประยุกต์ใช้ความรู้ทางด้านคอมพิวเตอร์ ไมโครคอนโทรลเลอร์ และอิเล็กทรอนิกส์จะเปิดตู้ออกโดยอาศัยการสแกนลายนิ้วมือ ในโครงการนี้ ได้ออกแบบผู้นิรภัยสแกนลายนิ้วมือให้จำกัดคนที่ใช้งานเฉพาะบุคคลที่ได้ทำการลงทะเบียนและเก็บข้อมูลไว้ในฐานข้อมูล

การทำงานของผู้นิรภัยสแกนลายนิ้วมือนี้ เริ่มจากการสมัครลายนิ้วมือส่วนบุคคลโดยสแกนลายนิ้วมือกับ Finger scan ที่ต่อกับคอมพิวเตอร์ แล้วคอมพิวเตอร์จะทำการเพื่อเก็บลายนิ้วมือส่วนบุคคลและข้อมูลส่วนบุคคลไว้ในฐานข้อมูล การใช้งานผู้นิรภัยสแกนลายนิ้วมือทำการโดยสแกนลายนิ้วมือกับ Finger scan ที่ต่อกับคอมพิวเตอร์แล้วคอมพิวเตอร์ทำการประมวลผลเพื่อเปรียบเทียบกับฐานข้อมูล ถ้าลายนิ้วมือที่ได้ทำการสมัครไว้แล้วตรงกับฐานข้อมูล คอมพิวเตอร์จะส่งสัญญาณไปยังบอร์ดไมโครคอนโทรลเลอร์ เพื่อให้บอร์ดไมโครคอนโทรลเลอร์จะส่งสัญญาณลจอกิจจากบอร์ดเพื่อไปส่งงานให้รีเลย์ทำงาน โดยการจ่ายกระแสไฟฟ้าให้กลอนแม่เหล็กไฟฟ้าทำงานเพื่อเปิดตู้

Project title The Fingerprint-Scan Safe
Name Mr. Pariwat Wongworanate ID. 51383959
 Mr. Terdsak Tong sug ID. 51384864
Project advisor Associate Professor Yongyut Chonbodecchalemroong, Ph.D.
Major Electrical Engineering
Department Electrical and Computer Engineering
Academic 2011

Abstract

This project presents the fingerprint-scan safe by applying knowledge of computers microcontrollers and electronic circuits. The safe can be opened by fingerprint only and is designed for restricted people who had registered and had been collected data in the database only.

The operations of the safe are firstly from applying for a personal fingerprint by a Finger Scan interfaced with a computer, then the computer collecting personal fingerprints and personal data into the database. The use of fingerprint-scan safe is done by scanning a fingerprint with the Finger Scan interfaced with the computer. After that the computer will compare the incoming fingerprint with the fingerprints in the database. If the incoming fingerprint and the application's database match, the computer will send a signal to the microcontroller board to command it to send a signal, logic 1, to order the relay to work by letting the current from the power supply flowing into a magnetic lock to open the safe.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความกรุณาเป็นอย่างยิ่งจากรองศาสตราจารย์ ดร. ชงยุทธ ชนบทิเฉลิมรุ่งเป็นอาจารย์ที่ปรึกษาโครงการและให้ความกรุณาในการให้คำปรึกษาทุกเรื่องที่เกี่ยวข้องกับการทำโครงการนี้ นอกจากนี้ยังมี ดร. พรพิศุทธิ์ วรจิรันตน์ และ ดร. สุพรรณนิกา วัฒนาเป็นอาจารย์กรรมการโครงการและให้คำแนะนำในการปรับปรุงโครงการ คณะผู้ดำเนินโครงการขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ในด้านต่างๆ ที่เป็นประโยชน์ ทั้งที่เกี่ยวข้องในด้านของสาขาอาชีพและที่เกี่ยวข้องในด้านของการดำเนินชีวิตในสังคมให้กับคณะผู้ดำเนินโครงการ

สุดท้ายนี้ คณะผู้ดำเนินโครงการขอกราบขอบพระคุณบิดา มารดา คณาจารย์และผู้มีพระคุณทุกท่านที่ได้อบรมสั่งสอน ให้ความรู้ ให้คำแนะนำ ให้คำปรึกษาที่มีคุณค่า แก่คณะผู้ดำเนินโครงการด้วยดีเสมอมา

คณะผู้ดำเนินโครงการ

นายปวีรรัต

วงศ์วรรณ

นายเทอดศักดิ์

ทองสุข

สารบัญ

หน้า

ใบรับรองปริญญาโท.....	ก
บทคัดย่อภาษาไทย.....	ข
บทคัดย่อภาษาอังกฤษ.....	ค
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 แผนการดำเนินงาน.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	3
1.6 งบประมาณของโครงการ.....	3
บทที่ 2 ทฤษฎีและหลักการทำงาน.....	4
2.1 หลักการทำงานของไมโครคอนโทรลเลอร์ Arduino.....	4
2.2 Fingerprint System.....	7
2.3 ระบบกลอนแม่เหล็กไฟฟ้า (Electric Strike).....	11
2.4 OPTO-Relay.....	16
2.5 ลำโพงบี๊ซเซอร์ (Buzzer).....	17
บทที่ 3 วิธีการดำเนินงาน.....	19
3.1 การศึกษาการดำเนินงาน.....	20
3.2 การออกแบบและสร้างชิ้นงาน.....	20
3.3 การอธิบายส่วนต่างๆของวงจร.....	23

สารบัญ (ต่อ)

	หน้า
บทที่ 4 การทดสอบและผลการเปิด-ปิดตู้ที่ได้.....	37
บทที่ 5 บทสรุป.....	42
5.1 สรุปผลการดำเนินงานโครงการ.....	42
5.2 ปัญหาและแนวทางแก้ไข.....	42
5.3 แนวทางในการพัฒนาโครงการ.....	42
5.4 ข้อเสนอแนะ.....	42
5.5 ข้อจำกัดของ โครงการตู้รักษาสแกนลายนิ้วมือ.....	43
เอกสารอ้างอิง.....	44
ภาคผนวก ก คู่มือการใช้งาน ตู้รักษาสแกนลายนิ้วมือ.....	45
ภาคผนวก ข วิธีการใช้งาน ไมโครคอนโทรลเลอร์ตระกูล ATMEGA 328P.....	60
ภาคผนวก ค โปรแกรมระบบการเปิด-ปิดตู้รักษาสแกนลายนิ้วมือ.....	73
ภาคผนวก ง Data Sheet LM 7812.....	164
ประวัติผู้ดำเนินโครงการ.....	168

สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงการทดสอบการเปิดตู้นิรภัยสแกนลายนิ้วมือที่ได้ทำการลงทะเบียนไว้.....	38
4.2 แสดงการทดสอบการเปิดตู้นิรภัยสแกนลายนิ้วมือที่ไม่ได้ทำการลงทะเบียนไว้.....	39
4.3 แสดงผลการสอบการทำงานของลำโพงเมื่อทำการเปิดตู้นิรภัยสแกนลายนิ้วมือที่ได้ทำการลงทะเบียนไว้.....	40
4.4 แสดงผลการสอบการทำงานของลำโพงเมื่อทำการเปิดตู้นิรภัยสแกนลายนิ้วมือที่ได้ไม่ทำการลงทะเบียนไว้.....	41



สารบัญรูป

รูปที่	หน้า
2.1 ภาพแสดงบอร์ด Arduino ทั่วๆ ไป.....	4
2.2 แสดงรูปวงจรของ Arduino.....	5
2.3 ตารางขาในบอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P.....	6
2.4 แสดงตัวอย่างเครื่อง Finger scan.....	7
2.5 ภาพแสดงลายก้นหอย.....	8
2.6 ภาพแสดงลายมัดหวาย.....	8
2.7 ภาพแสดงลายโค้ง.....	8
2.8 แสดงตัวอย่างของจุดสังเกต.....	8
2.9 แสดงตัวอย่าง กลอนประตูแม่เหล็กไฟฟ้า.....	11
2.10 ภาพของโซลินอยด์รูปแบนหนึ่ง.....	12
2.11 ภาพแสดงเส้นทางที่ฟลักซ์แม่เหล็กเดินทางฟลักซ์แม่เหล็กสร้างโดยแม่เหล็กไฟฟ้าและเดินผ่านเหล็กและอากาศ.....	12
2.12 ภาพแสดงรูปแกนเหล็กแบบไม่มีช่องอากาศกับมีช่องอากาศ (Air gap).....	13
2.13 รูปวงจรแม่เหล็กวงจรจริงและวงจรสมมูลย์.....	13
2.14 ภาพแสดงจุดรับ ไฟ +12V กระแสตรงจากวงจรเรกติไฟเออร์.....	14
2.15 เป็นภาพแสดงอุปกรณ์ภายในกลอนแม่เหล็กไฟฟ้า (Electric Strike).....	14
2.16 ภาพแสดงบริเวณที่เกิดการเหนี่ยวนำไฟฟ้าทำให้กลอนแม่เหล็กไฟฟ้าสามารถเปิดออกได้....	15
2.17 ภาพแสดงเมื่อกลอนแม่เหล็กไฟฟ้า (Electric Strike) เปิดออก.....	15
2.18 ภาพแสดง OPTO-Relay.....	16
2.19 Buzzer แบบ Diaphragm-Peizo.....	17
2.20 แผ่น Diaphragm ของลำโพงแบบเปียโซ.....	17
2.21 ลำโพงแบบเปียโซเมื่ออยู่ในกล่องกำทอน (Resonance box).....	18
3.1 แผนผังลำดับการทำงานของระบบการเปิด-ปิดตู้নিরภัยสแกนลายนิ้วมือด้วยการสแกนลายนิ้วมือ.....	19
3.2 ตู้เอกสารเดิม.....	20
3.3 แสดงการแบ่งส่วนภายในของตู้และส่วนที่ 1 ที่เป็นส่วนไว้ใช้เก็บสิ่งของ.....	21
3.4 แสดงส่วนที่ 2 ที่เป็นส่วนไว้เก็บวงจร.....	21
3.5 แสดงวงจรตู้নিরภัยสแกนลายนิ้วมือ.....	22

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.6 แสดงตัวอย่างหน้าต่าง โปรแกรม.....	23
3.7 วงจรเรียงกระแสแบบบริดจ์	24
3.8 (ก) แสดงการทำงานของวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์เมื่ออินพุตเป็นซีกลบ	25
3.8 (ข) แสดงการทำงานของวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์เมื่ออินพุตเป็นซีกบวก	25
3.9 แสดงแรงดันที่ประกอบด้วยค่าไฟตรง V_{DC} และค่าไฟสลับ V_{ripple}	26
3.10 บล็อกไดอะแกรมของแหล่งจ่ายไฟตรงคงค่าแรงดัน	28
3.11 วงจรแหล่งจ่ายไฟโดยใช้ MC78xx	29
3.12 วงจรและแรงดันไฟสลับที่จุดต่างๆของวงจรเรียงกระแสแบบบริดจ์	30
3.13 ขณะจ่ายแรงดันไฟสลับไบอัสตรงให้ไดโอด D_2 , D_4	30
3.14 ขณะจ่ายแรงดันไฟสลับไบอัสตรงให้ไดโอด D_1 , D_3	30
3.15 ภาพแสดงวงจรเรียงกระแสแบบเต็มคลื่นก่อนและหลังใส่ตัวเก็บประจุ.....	31
3.16 ภาพแสดง IC regulator 7812	31
3.17 วงจรเรียงกระแสของตู้วิทยุสแกนลายนิ้วมือ	32
3.18 ภาพแสดงวงจรของ OPTO-RELAY ขนาด 1 ช่อง.....	32
3.19 ภาพแสดงวงจรทั้งหมดของตู้วิทยุสแกนลายนิ้วมือประกอบด้วยวงจร Regulator วงจรOPTO-RELAY และArduino ต่อกับ Computer และลำโพง Buzzer.....	35
3.20 ภาพแสดงการต่ออุปกรณ์ทั้งหมดของตู้วิทยุสแกนลายนิ้วมือ.....	36

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

เนื่องด้วยกลุ่มผู้จัดทำโครงการมีความคิดเห็นว่าเอกสารหรือสิ่งของสำคัญเป็นสิ่งที่ทุกคนอยากเก็บไว้ให้ปลอดภัย โดยผู้เก็บเอกสารหรือผู้เก็บสิ่งของสำคัญเป็นที่นิยมโดยทั่วไป เนื่องจากเอกสารหรือสิ่งของสำคัญเป็นสิ่งที่ทุกคนให้ความสำคัญใน การเก็บรักษาให้ปลอดภัย ปัจจุบันผู้เก็บเอกสารมีหลากหลายรูปแบบ เช่น ผู้เก็บเอกสารที่ทำจากโลหะ ผู้เก็บเอกสารจากไม้ เป็นต้น แต่ปัญหาที่เกิดขึ้นคือจะทำอย่างไรให้ผู้เก็บเอกสารหรือสิ่งของสำคัญนั้นให้มีความปลอดภัยมากขึ้น จำกัดการใช้งานได้เฉพาะบุคคล และยังสามารถแสดงวัน เดือน ปี เวลาในการใช้งานผู้เอกสารได้ ดังนั้น กลุ่มผู้จัดทำจึงได้คิดทำตู้เก็บเอกสารสแกนลายนิ้วมือ โดยการประยุกต์ หลักการทำงานของไมโครคอนโทรลเลอร์ บอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P ทำการประมวลผลข้อมูลลายนิ้วมือเพื่อความสะดวกสบายในการเก็บเอกสารจึงใช้การเปิดตู้เอกสาร โดยการสแกนลายนิ้วมือ และมีเวลาแสดงผลว่าทำการเปิดตู้

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาหลักการทำงานของไมโครคอนโทรลเลอร์ บอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P

1.2.2 เพื่อศึกษาการเขียน โปรแกรม Arduino, Visual Basic, Microsoft Access

1.2.3 เพื่อศึกษาการทำงานของวงจรรขยายและวงจรรองสัญญาณแบบต่างๆ

1.2.4 เพื่อศึกษาการทำงานของระบบกลอนแม่เหล็กใช้สำหรับเปิดตู้ โดยจะทำการเปิดเมื่อสแกนลายนิ้วมือที่ได้รับการบันทึกไว้ใน Microsoft Access

1.2.5 เพื่อศึกษาการเก็บลายนิ้วมือของผู้ใช้งาน วัน/เดือน/ปีเวลาที่ใช้งานใน Microsoft Access

1.3 ขอบเขตของโครงการงาน

1.3.1 การควบคุมประมวลผลด้วยไมโครคอนโทรลเลอร์บอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA

1.3.2 เขียนโปรแกรมสแกนลายนิ้วมือโดย Visual Basic

1.3.3 เก็บข้อมูลลายนิ้วมือใน Microsoft Access

1.3.4 แสดงผลโดยการเมื่อสแกนลายนิ้วมือที่ได้บันทึกใน Microsoft Access แล้วผู้จะทำการเปิดออกถ้าลายนิ้วมือตรงกับที่เก็บไว้ใน Microsoft Access

1.4 แผนการดำเนินงานโครงการงาน

กิจกรรม	ระยะเวลาดำเนินงาน					
	มิ.ย.	ก.ค.	ส.ค.- ก.ย.	ต.ค.- ธ.ค.	ม.ค.- มี.ค.	เม.ย.- พ.ค.
1. ศึกษาข้อมูลและวิเคราะห์ขอบเขตของโครงการงาน	←→					
2. ศึกษาคุณลักษณะและหลักการดำเนินงาน		←→				
3. ศึกษาและออกแบบวงจรขยายและวงจรกรองสัญญาณแบบต่างๆ	←→	←→				
4. ศึกษาข้อมูลไมโครคอนโทรลเลอร์ AVR รุ่น Duemilanove 2009 ATMEGA 328P			←→	←→		
5. ทำการสร้างตู้นิรภัยสแกนลายนิ้วมือ				←→	←→	
6. ทำการทดสอบตู้นิรภัยสแกนลายนิ้วมือ				←→	←→	
7. ปรับปรุงแก้ไข					←→	
8. จัดทำรูปเล่มรายงาน						←→

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 สามารถทำการเขียนโปรแกรมในไมโครคอนโทรลเลอร์บอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P ไปใช้กับ คิวน์ริกซ์สแกนลายนิ้วมือได้

1.5.2 สามารถเขียนโปรแกรมใน Visual Basic 6.0 แล้วรับส่งข้อมูล Visual Basic 6.0 ไปยังบอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P แล้วสามารถเปิด-ปิดคิวน์ริกซ์สแกนลายนิ้วมือโดยการสแกนลายนิ้วมือได้

1.5.3 มีความรู้ความเข้าใจเกี่ยวกับวงจรขยายและวงจรกรองสัญญาณแบบต่างๆ และสามารถนำมาประยุกต์ใช้งานจริงได้

1.5.4 ได้คิวน์ริกซ์สแกนลายนิ้วมือที่เปิด-ปิดคิวน์ริกซ์สแกนลายนิ้วมือได้

1.6 งบประมาณของโครงการ

1.6.1 ค่าวัสดุอุปกรณ์	3,000 บาท
1.6.2 ค่าวัสดุอื่นๆ	1,000 บาท
1.6.3 ค่าถ่ายเอกสารและจัดทำรูปเล่ม	1,000 บาท
รวมเป็นเงิน	<u>5,000</u> บาท

(ห้าพันบาทถ้วน)

หมายเหตุ: เฉลี่ยทุกรายการ

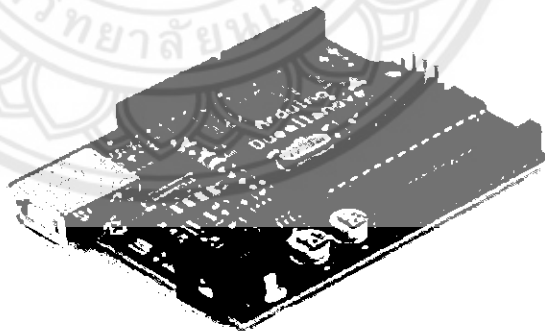
(ได้รับการสนับสนุนจากคณะวิศวกรรมศาสตร์เป็นจำนวนเงิน 2,000 บาท)

บทที่ 2

ทฤษฎีและหลักการทำงาน

2.1 หลักการทำงานของไมโครคอนโทรลเลอร์ Arduino

Arduino (อ่านว่า อา-ดู-อิ-โน้ หรือ อาดูอิน) คือไมโครคอนโทรลเลอร์ ชนิดหนึ่ง เป็นแบบที่เรียกว่า Open Hardware กล่าวคือ Arduino อุปกรณ์ที่มีแบบส่วนประกอบเป็นมาตรฐานที่เปิดเผยเป็นเครื่องมือที่สามารถทำให้คอมพิวเตอร์สามารถรับส่งสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพ ตัวบอร์ดออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยว และมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้บอร์ดทำงาน Arduino สามารถประยุกต์ทำเครื่องใช้รับสัญญาณจากสวิตช์ หรือ เซนเซอร์ และควบคุมหลอดไฟมอเตอร์หรืออุปกรณ์อื่นๆ Arduino เป็นได้ทั้งแบบทำงานอิสระหรือทำงานติดต่อกับโปรแกรมที่ทำงานบนเครื่องพีซีแสดงตัวอย่างในรูปที่ 2.1 ภาพแสดงบอร์ด Arduino ทัวๆ ไปรูปที่ 2.2 แสดงรูปร่างของ Arduino และรูปที่ 2.3 ตารางขาในบอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P



รูปที่ 2.1 ภาพแสดงบอร์ด Arduino ทัวๆ ไป

Arduino

2	PD0/RXD/PCINT16	PB0/CP1/CLK0/PCINT0	14
3	PD1/TXD/PCINT17	PB1/OC1A/PCINT1	15
4	PD2/INT0/PCINT18	PB2/ \overline{SS} /OC1B/PCINT2	16
5	PD3/INT1/OC2B/PCINT19	PB3/MOSI/OC2A/PCINT3	17
6	PD4/T0/XCK/PCINT20	PB4/MISO/PCINT4	18
11	PD5/T1/OC0B/PCINT21	PB5/SCK/PCINT5	19
12	PD6/AIN0/OC0A/PCINT22	PB6/TOSC1/XTAL1/PCINT6	9
13	PD7/AIN1/PCINT23	PB7/TOSC2/XTAL2/PCINT7	10
21	AREF	PC0/ADC0/PCINT8	23
20	AVCC	PC1/ADC1/PCINT9	24
		PC2/ADC2/PCINT10	25
		PC3/ADC3/PCINT11	26
		PC4/ADC4/SDA/PCINT12	27
		PC5/ADC5/SCL/PCINT13	28
		PC6/ \overline{RESET} /PCINT14	1

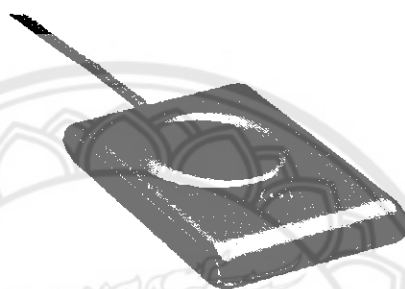
ATMEGA368P

2=pin0	13=pin7	23=pin14(A0)
3=pin1	14=pin8	24=pin15(A1)
4=pin2	15=pin9	
5=pin3	16=pin10	
6=pin4	17=pin11	
11=pin5	18=pin12	
12=pin6	19=pin13	

รูปที่ 2.3 ตารางขาในบอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P

2.2 Fingerprint system

Fingerprint system หรือ ระบบเครื่องสแกนลายนิ้วมือ คือ ระบบที่ต้องการใช้ลายนิ้วมือของบุคคลที่ได้รับอนุญาต หรือ บุคคลที่ได้ทำการบันทึกลายนิ้วมือลงในตัวเครื่องสแกนลายนิ้วมือ ที่ใช้ควบคุมระบบการทำงานต่างๆแล้ว บุคคลที่ไม่ได้ทำการบันทึกลายนิ้วมือไว้ ก็จะไม่สามารถเข้าใช้งานในระบบที่ต้องการมีการสแกนลายนิ้วมือนี้ได้รูปที่ 2.4 แสดงตัวอย่างเครื่อง Finger scan



รูปที่ 2.4 แสดงตัวอย่างเครื่อง Finger scan

ข้อดีของการใช้ลายนิ้วมือในการระบบตัวบุคคล

ลายนิ้วมือของแต่ละคนจะเริ่มปรากฏขึ้นตั้งแต่เป็นตัวอ่อนอายุ 3 ถึง 4 เดือนในครรภ์มารดา เป็นผิวหนังส่วนที่มีสัน (Ridge) และมีร่อง (Furrow) เอาไว้ใช้สำหรับอำนวยความสะดวกในการหยิบจับสิ่งของ สันและร่องที่ปรากฏนี้ จะมีคุณลักษณะที่สำคัญ 2 ประการคือ

ไม่มีการเปลี่ยนแปลงรูปแบบตามกาลเวลา (แต่อาจจะเปลี่ยนขนาดได้) ลายนิ้วมือจะไม่เปลี่ยนแปลงรูปแบบ (Permanence) ตั้งแต่แรกเกิดจนถึงกระทั่งวันที่เราตาย แต่ก็อาจจะเปลี่ยนแปลงขนาดได้ตามขนาดของร่างกาย

มีรูปแบบเฉพาะในแต่ละบุคคลการที่ลายนิ้วมือมีรูปแบบเฉพาะในแต่ละบุคคล (Individuality) เป็นคุณสมบัติที่สำคัญอย่างหนึ่งของลายนิ้วมือ นับตั้งแต่เริ่มมีการใช้เก็บ และเปรียบเทียบลายนิ้วมือโดยวิธีสมัยใหม่ มีมาร้อยกว่าปีแล้ว ยังไม่มีการตรวจพบว่ามีกรณีเหมือนกันของลายนิ้วมือ

ลายนิ้วมือของแต่ละคนนั้นจะมีลักษณะเฉพาะมากจนกระทั่งแม้แต่ (Identical Twin) ก็ยังมีลายนิ้วมือที่แตกต่างกัน (แต่มีรูปแบบ DNA ที่เหมือนกัน) อย่างไรก็ตามรูปแบบของลายนิ้วมือ นั้นก็จะมีลักษณะการคล้ายกันของคนในครอบครัว หรือพูดได้อีกอย่างหนึ่งว่า รูปแบบของลายนิ้วมือมี

การถ่ายทอตกันทางพันธุกรรมลายนิ้วมือมีแบบใหญ่ๆที่แสดงในรูปที่ 2.5 ภาพแสดงลายก้นหอย
2.6 ภาพแสดงลายมัดหวาย 2.7 ภาพแสดงลายโค้ง

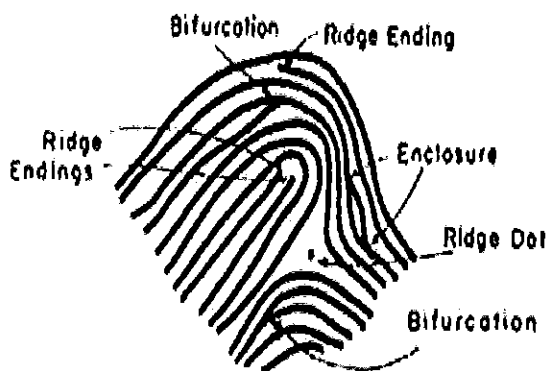


รูปที่ 2.5 ภาพแสดงลายก้นหอย รูปที่ 2.6 ภาพแสดงลายมัดหวาย รูปที่ 2.7 ภาพแสดงลายโค้ง

การจดจำลายนิ้วมือ สำหรับการตรวจดูลายนิ้วมือว่าเป็นลายนิ้วมือเดียวกันหรือไม่นั้น มีหลากหลายวิธีด้วยกัน แต่วิธีที่เป็นที่นิยมมากที่สุด ได้แก่ การเปรียบเทียบจุดสังเกตเล็กๆ (Minutiae) จุดที่ว่านี้คือจุดที่เส้นลายนิ้วมือ (Ridge) มาบรรจบกัน หรือ แยกออกจากกัน เป็นจุดจบของเส้นและจุดที่ว่ามีของแต่ละคนจะไม่เหมือนกัน

รายการของจุดสังเกต (ตัวอย่างตามรูปที่ 2.8) ได้แก่

- Bifurcation เป็นจุดที่แยกเส้นออกเป็นหลายเส้น
- Divergence เป็นจุดที่เส้น 2 เส้นที่อยู่ขนานกัน แยกออกจากกัน หรือ เข้ามารวมกัน
- Enclosure เป็นลักษณะที่เส้นแยกออกจากกัน และ กลับมารวมกันเป็นเส้นเดียวกันใหม่
- Endings เป็นตำแหน่งที่เส้นนั้นๆ สิ้นสุดลง



รูปที่ 2.8 แสดงตัวอย่างของจุดสังเกต

ในการบันทึกลายนิ้วมือจะมีการบันทึกตำแหน่งของจุดสังเกตเหล่านี้ไว้ปกติแล้วจะบันทึกได้ประมาณ 30-40 จุดต่อ 1 ลายนิ้วมือ และนำข้อมูลมาเปรียบเทียบกับลายนิ้วมือที่สแกนเข้ามาว่าตรงกันหรือไม่

การเก็บข้อมูลลายนิ้วมือลงฐานข้อมูลขั้นแรกจะเก็บตัวอย่างลายนิ้วมือเพื่อบันทึกลงฐานข้อมูล ดังนี้

- เก็บภาพลายนิ้วมือแต่ละนิ้ว โดยการเก็บภาพบายนิ้วมือจะทำการสแกนแต่ละนิ้วมากกว่า 1 ครั้ง เนื่องจากการเก็บภาพแต่ละภาพอาจมีการคลาดเคลื่อนจากตำแหน่งเดิมและการลง
- นำหนักไม่เท่ากัน ทำการค้นหาจุดรายละเอียดบนเส้นลายนิ้วมือ โดยการค้นหาจุดสังเกตเล็กๆ (Minutiae) ของแต่ละภาพที่เป็นนิ้วเดียวกัน ทำการแปลงข้อมูลลายนิ้วมือเป็นการสร้างรหัสแทน
- บันทึกรหัสข้อมูลลายนิ้วมือลงในฐานข้อมูล โดยมีการเก็บแบบข้อมูล การบันทึกแบบ 1:1 หนึ่งคนใช้ลายนิ้วมือบันทึกได้ 1 ลายนิ้วมือต่อ 1 USER การตรวจสอบแบบ 1:1 คือ ผลการตรวจสอบลายนิ้วมือนั้นไม่ว่าจะใช้นิ้วใดก็ตาม (ที่บันทึกไว้แล้ว) ผลอาจออกมาจะเป็นหนึ่งคนนั้นเท่านั้น
- วิธีค้นหาลายนิ้วมือ โดยการทำการแปลงข้อมูลลายนิ้วมือเป็นรหัส และนำค่ารหัสที่ได้ไปเปรียบเทียบกับค่ารหัสลายนิ้วมือที่เก็บไว้ในฐานข้อมูล เมื่อค่ารหัสตรงหรือคลาดเคลื่อนต่างกันน้อยกว่าที่กำหนด ก็จะแสดงบุคคลที่เป็นเจ้าของลายนิ้วมือนั้น
- โปรแกรม Fingerprint ของ SDK ย่อมาจาก Software Development Kit ชุดเครื่องมือที่สมบูรณ์ในการพัฒนาซอฟต์แวร์ผ่านการเขียน โปรแกรมภาษาต่าง ๆ เช่น Visual Basic และ Visual C++ SDK ได้รวมเอาชุดคำสั่ง DLL (สำหรับ VC) ชุดควบคุม OCX (สำหรับ VB, Delphi) สามารถใช้ในชุดของ SDK ในการสร้างโปรแกรมโดยใช้สถาปัตยกรรมของ DirectShow มีการจับภาพที่ให้คุณภาพสูงและการเล่นไฟล์มัลติมีเดียสตรีมบนแพลตฟอร์มของ Microsoft Windows แทนที่การเขียนโค้ดในการสร้างโปรแกรมขึ้นมา เช่นฟอร์ม, ปุ่มคำสั่งและเมนู API (Application Programming Interface) อินเตอร์เฟซนี้ช่วยในการสามารถเรียกฟังก์ชันที่เหมาะสมและชุดข้อมูลคำสั่งจากใน SDK

- ในส่วนของชุด SDK ที่ใช้ในทำงานผู้สแกนลายนิ้วมืออยู่ในส่วนการทำงานของModule และClass Modules ในส่วนของโปรแกรม Visual Basic Studio6.0 (ในส่วน Module Class Module คือใช้ประกาศตัวแปร Public ใช้แทน Dim จะประกาศทีเดียว เห็นทั้งหมด แล้วสามารถเขียน Sub และ Function ที่ใช้งานบ่อยๆ ไว้ทีเดียว ไม่ต้องทำทุกฟอร์ม สำหรับโปรแกรมที่ใช้ความละเอียด จะได้ไม่สับสนตัวแปร และ Function)

ชุด SDK มีความสำคัญคือ

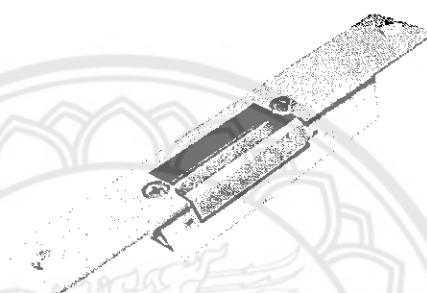
1. ใช้สำหรับติดต่อ Fingerprint กับ Computer
2. ใช้สำหรับการสแกนลายนิ้วมือ
3. ใช้ในการเก็บข้อมูลใน Microsoft Access
4. ใช้ในการเก็บรูปแบบลายนิ้วมือของผู้ใช้งานต่างๆในรูปแบบเช่นกันหอม มัดหวาย เป็นต้น
5. ใช้ในการเชื่อมต่อ Microsoft Jet ole 4.0, ODBC



2.3 ระบบกลอนแม่เหล็กไฟฟ้า (Electric Strike)

ระบบกลอนแม่เหล็กไฟฟ้า ตัวระบบกลอนแม่เหล็กไฟฟ้าใช้กระแสใช้งาน 200mA ที่ 12 โวลต์ กระแสตรง โซลินอยด์ทำงานต่อเนื่อง โซลินอยด์สามารถใช้ไฟฟ้าอย่างต่อเนื่อง

หลักการคือ ถ้าประตูอยู่ในสถานะปิดอยู่ ระบบกลอนจะทำการล็อกกลอนแม่เหล็กไฟฟ้า เมื่อต้องการเปิดกลอนแม่เหล็กไฟฟ้า โดยจะต้องมีการจ่ายไฟเข้าไปกลอนแม่เหล็กไฟฟ้าเพื่อกระทำการปลดล็อกกลอนแม่เหล็กไฟฟ้าซึ่งแสดงในรูปที่ 2.9 ระบบกลอนประตูแม่เหล็กไฟฟ้า



รูปที่ 2.9 แสดงตัวอย่าง กลอนประตูแม่เหล็กไฟฟ้า

ขดลวดแม่เหล็กไฟฟ้า (Electromagnetic coil)

ขดลวดแม่เหล็กไฟฟ้า หรือ อังกฤษ (Electromagnetic coil) หรือขดลวดหรือ คอยล์ เกิดขึ้นเมื่อตัวนำ (ลวดทองแดงเต็มเส้นหุ้มฉนวน) พันรอบแกนหรือขึ้นรูปเป็นวง เพื่อการเหนี่ยวนำหรือสร้างสนามแม่เหล็กไฟฟ้า เมื่อกระแสไฟฟ้าผ่านขดลวดจะสร้างสนามแม่เหล็ก วงรอบหนึ่งวงของลวดทองแดงจะเรียกว่าหนึ่งรอบ คอยล์จะประกอบด้วยขดลวดหนึ่งรอบหรือมากกว่า ใช้ในวงจรอิเล็กทรอนิกส์, ขั้วต่อเรียกว่าแท๊ป ขดลวดมักจะถูกเคลือบด้วยน้ำมันวานิชหรือเทปฉนวนเพื่อป้องกันกระแสไฟฟ้ารั่วอีกชั้นหนึ่งและผูกมัดไว้กับที่

หลักการทั่วไป

วัสดุที่ใช้เป็นขดลวดตัวนำไฟฟ้าขึ้นอยู่กับนำไปประยุกต์ใช้ แต่ในทุกกรณีในแต่ละรอบที่พันจะต้องทำให้เป็นฉนวนไฟฟ้าออกจากกันเพื่อให้แน่ใจว่ากระแสจะไหลในทุกกรอบของขดลวด. สำหรับกำลังไฟฟ้าขนาดเล็กและการใช้ในงานสัญญาณที่กระแสที่ใช้มีค่าต่ำและความต่างศักย์ระหว่างรอบที่อยู่ติดกันมีขนาดเล็ก ขดลวดที่พันมักจะทำจากลวดแม่เหล็กเคลือบเช่นลวด Formvar หรือแปลงไฟฟ้าขนาดใหญ่ในการทำงานที่แรงดันไฟฟ้าสูงอาจจะพันด้วยแถบตัวนำทองแดงสีเหลืองฉนวนกระดาษชุบน้ำมันและบล็อกของ Pressboard

โซลินอยด์ (Solenoid)

โซลินอยด์ เป็นขดลวดพันเป็นเกลียวแน่น ทางด้านกายภาพ โซลินอยด์หมายถึงเส้นลวดที่ยาว ม้วนเป็นวงเล็กๆ มักจะพันรอบแกนโลหะเมื่อกระแสไฟฟ้าผ่านจะก่อให้เกิดสนามแม่เหล็กที่สม่ำเสมอในบริเวณของพื้นที่(ที่จะต้องทำการทดลองต่อไป) โซลินอยด์มีความสำคัญเพราะสามารถสร้างสนามแม่เหล็กที่ควบคุมได้และสามารถนำมาใช้เป็นแม่เหล็กไฟฟ้าได้

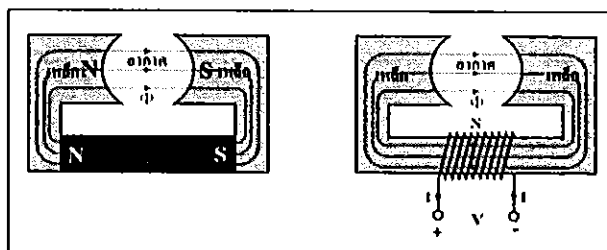
ทางด้านวิศวกรรม โซลินอยด์ยังอาจหมายถึงความหลากหลายของอุปกรณ์แปลงสัญญาณ (Transducer) ที่แปลงพลังงานให้เป็นการเคลื่อนไหวเชิงเส้น คำนี้อาจถูกใช้อ้างถึงว่าตัวโซลินอยด์ก็ได้ประกอบด้วยขดลวดกลไกแม่เหล็กไฟฟ้าเพื่อควบคุมการทำงานของทั้งวาล์วนิวเมติกหรือไฮดรอลิก, หรือเป็นสวิตช์ โซลินอยด์เป็นรีเลย์ชนิดหนึ่งที่ใช้ควบคุมสวิตช์ไฟฟ้าเช่นสวิตช์สตาร์ทในรถยนต์ เป็นต้น



รูปที่ 2.10 ภาพของ โซลินอยด์รูปแบบหนึ่ง

หลักการวงจรแม่เหล็กของตัวอย่าง กลอนประตูปower

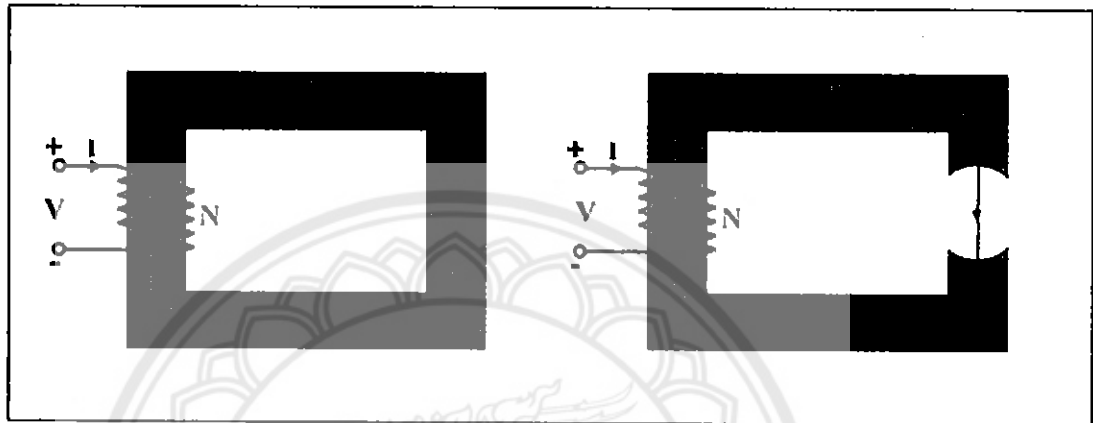
2.3.1 ส่วนประกอบของวงจรแม่เหล็กในกลอนประตูปower วงจรแม่เหล็กหมายถึงเส้นทางที่ฟลักซ์แม่เหล็กเดินทางผ่านวัสดุชนิดต่างๆอย่างครบวงจร โดยฟลักซ์แม่เหล็กอาจสร้างจากแม่เหล็กถาวรหรือขดลวดแม่เหล็ก



รูปที่ 2.11 ภาพแสดงเส้นทางที่ฟลักซ์แม่เหล็กเดินทางฟลักซ์แม่เหล็กสร้างโดยแม่เหล็กไฟฟ้าและเดินผ่านเหล็กและอากาศ

ส่วนประกอบของวงจรแม่เหล็กไฟฟ้า

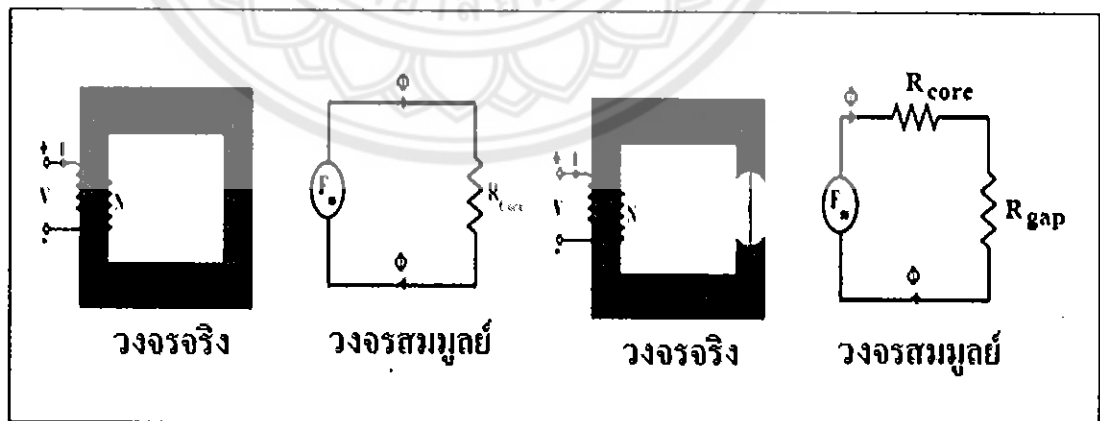
แม่เหล็กไฟฟ้า ประกอบด้วยขดลวดพันรอบแกนเป็นจำนวนรอบ (N) จำนวนหนึ่ง โดยขดลวดมีกระแสไฟฟ้า (I) ไหลผ่าน เพื่อกำเนิดแรงดันแม่เหล็ก (F_m) สำหรับสร้างฟลักซ์แม่เหล็ก (Φ) ให้เกิดขึ้นในแกน (Core) ของขดลวดอาจจะเป็นแกนเหล็กทั้งหมด



รูปที่ 2.12 ภาพแสดงรูปแกนเหล็กแบบ ไม่มี Airgap กับมีช่องอากาศ (Air gap)

2.3.2 วงจรจริงและวงจรสมมูลย์ของวงจรแม่เหล็กไฟฟ้า

วงจรแม่เหล็กที่ประกอบด้วยแกนแม่เหล็กทั้งหมดและแกนเหล็กกับช่องอากาศ (Air gap)



รูปที่ 2.13 รูปวงจรแม่เหล็กวงจรจริงและวงจรสมมูลย์

แกนของขดลวดที่เป็นแกนเหล็ก จะมีค่าความต้านทานแม่เหล็กต่ำและเขียนวงจรสมมูลย์ เช่นเดียวกับวงจรไฟฟ้า โดยมีแรงดันแม่เหล็ก (F_m) เป็นแหล่งจ่ายความต้านทานแม่เหล็กของแกนเหล็ก R_{Core} เป็นความต้านทานของวงจรและมีฟลักซ์แม่เหล็ก (Φ) เดินทางในวงจร

2.3.3 การทำงานของระบบกลอนแม่เหล็กไฟฟ้า (Electric Strike)

รูปที่ 2.14 เป็นภาพแสดงจุดรับไฟ +12 V กระแสตรงจากวงจรเรกติไฟเออร์เพื่อเปิดตู้หนีภัย สแกนลายนิ้วมือ

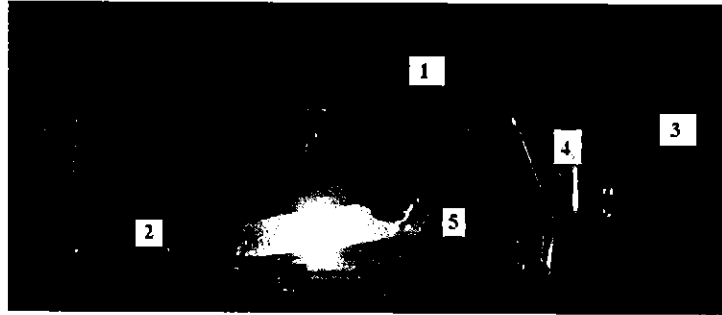


จุดรับไฟ +12V เพื่อเปิด
กลอนแม่เหล็กไฟฟ้า

รูปที่ 2.14 ภาพแสดงจุดรับไฟ +12 V กระแสตรงจากวงจรเรกติไฟเออร์

ภาพแสดงอุปกรณ์ภายในกลอนแม่เหล็กไฟฟ้าแสดงในรูปที่ 2.15 ประกอบด้วย

1. ตัวกลอนสำหรับเปิดปิดประตู
2. สายไฟสำหรับรับ +12 V กระแสตรงเพื่อเปิดกลอนแม่เหล็กไฟฟ้า
3. ขดลวดโซลินอยด์
4. กลไกในการปลดล็อกกลอนแม่เหล็กไฟฟ้า
5. กลไกในการล็อกกลอนแม่เหล็กไฟฟ้า (เดี่ยล็อกตัวกลอนสำหรับเปิดปิดประตู)



รูปที่ 2.15 เป็นภาพแสดงอุปกรณ์ภายในกลอนแม่เหล็กไฟฟ้า (Electric Strike)

เมื่อมี +12 V กระแสตรงไหลเข้ากลอนแม่เหล็กไฟฟ้า (Electric Strike) แล้วแม่เหล็กจะเกิดการเหนี่ยวนำไฟฟ้าตรงจุดในรูปที่ 2.16 แล้วระบบกลอนแม่เหล็กไฟฟ้าสามารถเปิดออกได้แสดงในรูป 2.17



รูปที่ 2.16 ภาพแสดงบริเวณที่เกิดการเหนี่ยวนำไฟฟ้าทำให้กลอนแม่เหล็กไฟฟ้าสามารถเปิดออกได้



รูปที่ 2.17 ภาพแสดงเมื่อกลอนแม่เหล็กไฟฟ้า (Electric Strike) เปิดออก

2.4 OPTO-Relay

ระบบ OPTO-Relay คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่ตัด-ต่อวงจร คล้ายกับสวิตช์ โดยใช้หลักการหน้าสัมผัสการทำงานของรีเลย์เมื่อทำการจ่ายไฟให้กับตัวรีเลย์ แล้วจะทำให้หน้าสัมผัสติดกันเป็นวงจรปิด และตรงข้ามทันทีเมื่อไม่ได้จ่ายไฟให้กับตัวรีเลย์ ก็จะเป็นวงจรเปิด ไฟที่ใช้จ่ายให้กับตัวรีเลย์ไฟที่ได้มาจากวงจรเรียงกระแส (Rectifier Circuit) ดังนั้นทันทีที่จ่ายไฟให้กับตัวรีเลย์ ก็จะทำให้รีเลย์ทำงาน

แล้วมี OPTO อุปกรณ์สวิตต์ควบคุมด้วยแสง (OPTO-Isolator) หรือที่เรียกว่าออปโตคัปเปิลเลอร์ (OPTO-Coupler) เราสามารถนำมาประยุกต์ใช้ในการออกแบบควบคุมไฟฟ้า โดยใช้หลักการ กระแสต่ำควบคุมกระแสสูง และแรงดันไฟฟ้าต่ำควบคุมแรงดันไฟฟ้าสูง ข้อดีที่สำคัญของ ออปโตคัปเปิลเลอร์ (OPTO-Coupler) นี้ คือกราวด์แยกออกจากกันทำให้ไม่มีการรบกวนกันทั้งสองฝั่งที่ต้องการความมีเสถียรของการควบคุมหรือใช้กับความถี่ต่างๆ ที่ต้องการควบคุมได้แม่นยำ OPTO-Relay ภาพแสดงที่รูปที่ 2.18



รูปที่ 2.18 ภาพแสดง OPTO-Relay

2.5 ลำโพงบีซเซอร์

บีซเซอร์ (Buzzer) คือลำโพงแบบแม่เหล็กหรือ แบบเพียโซที่มีวงจรถ้าเนิดความถี่ (Oscillator) อยู่ในตัว เมื่อป้อนแรงดันสามารถกำเนิดเสียงได้ด้วยตัวเอง แต่ไม่สามารถเปลี่ยนความถี่ของเสียงได้ ลำโพงขนาดเล็กที่ใช้ต่อกับไมโครคอนโทรลเลอร์

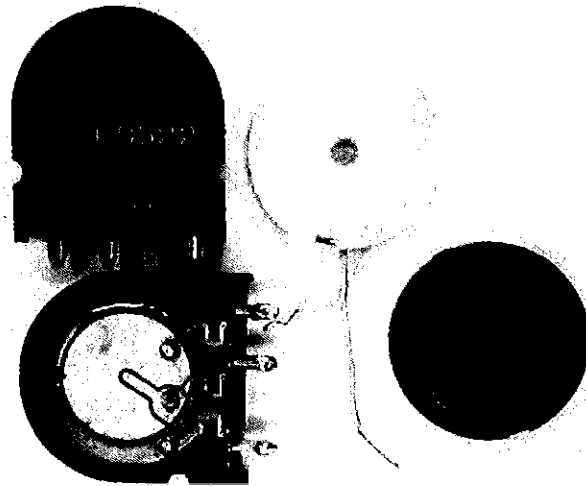
- มีความต้านทานสูงสามารถต่อกับไมโครคอนโทรลเลอร์ได้โดยตรง
- แผ่นลำโพงแบบเพียโซ มีคุณสมบัติเมื่อป้อนแรงดันไฟฟ้าจะทำให้เกิดการบิดงอไปมาทำให้เกิดเสียงขึ้น
- คุณสมบัติของเสียงขึ้นอยู่กัขนาดของแผ่น Piezo และกล่องกำทอน (Resonance box)



รูปที่ 2.19 Buzzer แบบ Diaphragm - Piezo



รูปที่ 2.20 แผ่น Diaphragm ของลำโพงแบบเพียโซ



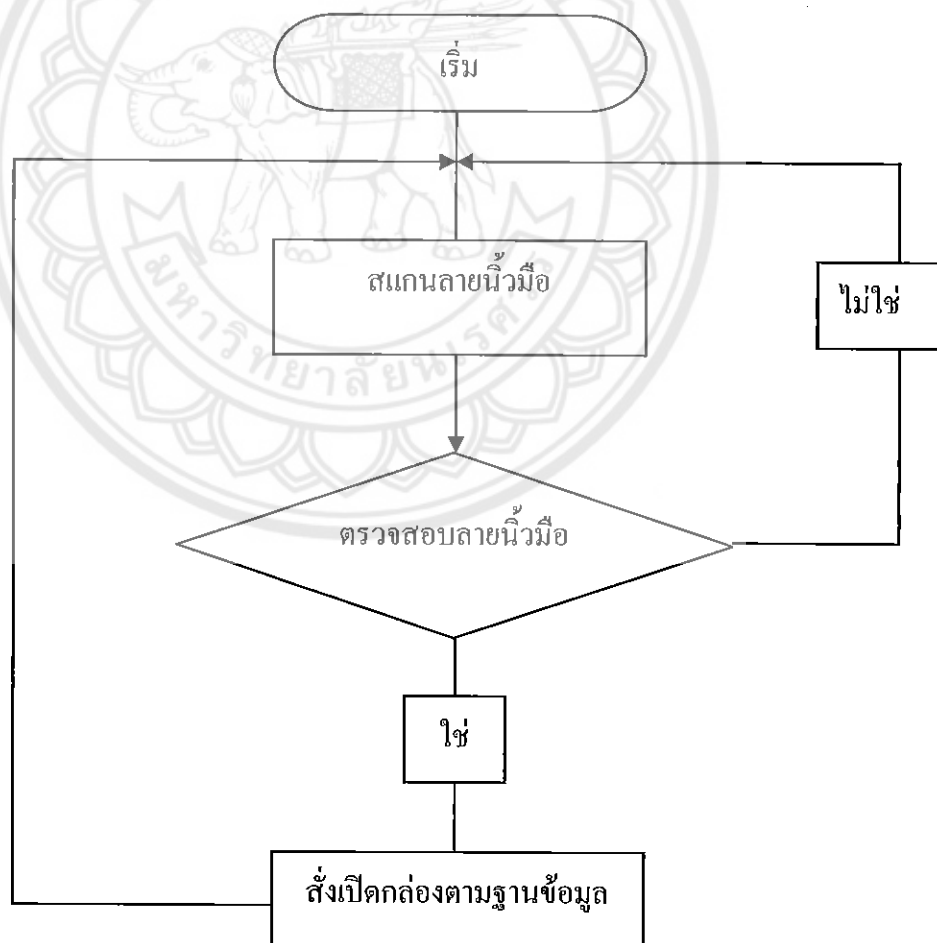
รูปที่ 2.21 ลำโพงแบบเปียโซเมื่ออยู่ในกล่องกำทอน (Resonance box)



บทที่ 3

วิธีการดำเนินงาน

วิธีการดำเนินการออกแบบและสร้างระบบการควบคุมการปิดเปิดตู้ด้วยไมโครคอนโทรลเลอร์ เพื่อใช้การเปิด-ปิดตู้เก็บเอกสาร โดยการใช้แม่เหล็กเป็นการเหนี่ยวนำ เพื่อให้การเปิด-ปิดตู้เป็นไปตามข้อมูลที่เก็บไว้ใน Microsoft Access เมื่อลายนิ้วมือตรงกับค่าที่เก็บไว้ โปรแกรม Visual basic จะส่งสัญญาณไปที่ไมโครคอนโทรลเลอร์ บอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P เพื่อจะสั่งให้ Relay ทำการส่งไฟที่ไปเลี้ยงในกลอนแม่เหล็ก ทำให้ประตูตู้เก็บเอกสารเปิดออก (ตามแผนผังลำดับการทำงานของระบบการเปิด-ปิดตู้ nirky สแกนลายนิ้วมือด้วยการสแกนลายนิ้วมือรูปที่ 3.1



รูปที่ 3.1 แผนผังลำดับการทำงานของระบบการเปิด-ปิดตู้ nirky สแกนลายนิ้วมือด้วยการสแกนลายนิ้วมือ

3.1 การศึกษาการทำงาน

การศึกษาการทำงานของ บอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P และ จัดทำระบบการควบคุมการเปิด-ปิดตู้ด้วยการสแกนลายนิ้วมือ กลุ่มผู้จัดทำโครงการ ได้สังเกตเห็นว่าการสแกนลายนิ้วมือเป็น Password ที่ดี โดยจะสามารถเปิด-ปิดตู้ได้เฉพาะบุคคลที่ได้บันทึกค่าไว้ในฐานข้อมูล ส่วนบุคคลอื่นที่ไม่ได้บันทึกค่าไว้ในฐานข้อมูลก็จะไม่สามารถทำการเปิดตู้ได้ สามารถศึกษาวิธีการใช้งานการเปิด-ปิดตู้ด้วยการสแกนลายนิ้วมือได้จาก ภาคผนวก ก

3.2 การออกแบบและสร้างชิ้นงาน

ในการออกแบบและสร้างระบบควบคุมการเปิด - ปิดตู้ด้วยการสแกนลายนิ้วมือ ให้สามารถใช้งานได้เฉพาะบุคคลตามที่ออกแบบไว้ ได้มีการออกแบบและสร้างชิ้นงานทั้งในส่วนฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software)

3.2.1 การออกแบบและสร้างชิ้นงานในส่วนฮาร์ดแวร์

ในส่วนของฮาร์ดแวร์ จากเดิมเป็นตู้เอกสารธรรมดา (รูปที่ 3.2) ที่สามารถเปิด-ปิดใช้งานได้หลายบุคคล ในแต่ละครั้งบุคคลใครก็สามารถเปิดใช้งานได้ แต่เมื่อกลุ่มผู้จัดทำได้นำระบบการสแกนลายนิ้วมือเพื่อมาเป็น Password ในการเปิด-ปิดตู้ มีการเก็บข้อมูลเวลา ในการเปิด-ปิดตู้ ทำให้สามารถจำกัดบุคคลที่ใช้งานตู้ได้ และทำให้มีความปลอดภัยมากขึ้น

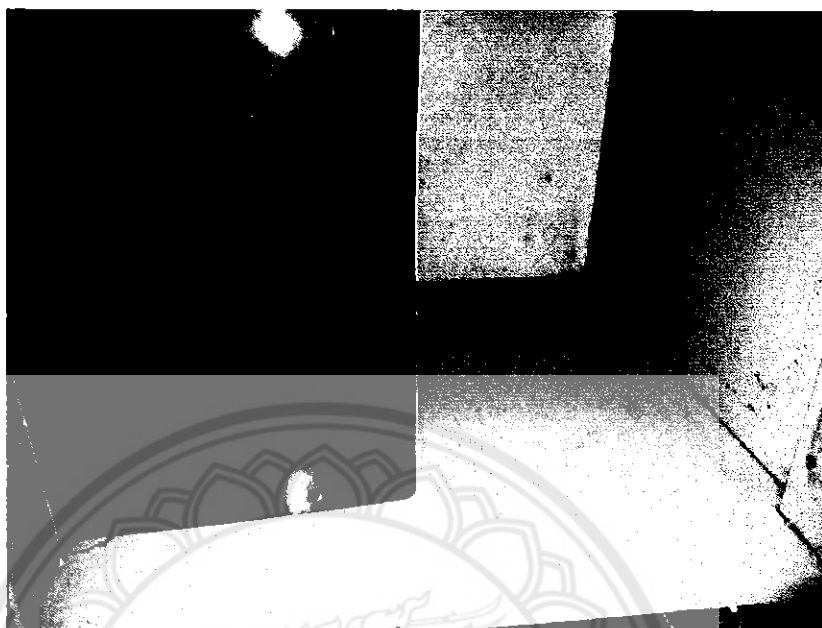


รูปที่ 3.2 ตู้เอกสารเดิม

ได้ออกแบบตู้โดยแบ่งส่วนภายในของตู้เป็น 2 ส่วน (รูปที่ 3.3) โดยส่วนที่ 1 ที่เป็นส่วนไว้ใช้เก็บสิ่งของหรือเก็บเอกสารจุดต่างๆเฉพาะบุคคล และส่วนที่ 2 (รูปที่ 3.4) เป็นส่วนที่ไว้เก็บวงจรอีกส่วนหนึ่งแยกจากส่วนเก็บสิ่งของหรือเก็บเอกสารเพื่อที่จะเป็นการเพิ่มระบบความปลอดภัย

นิรภัยสแกนลายนิ้วมือเพิ่มมากขึ้นถ้าเกิดไฟดับจะสามารถซ่อมแซมได้โดยเปิดตู้ในส่วนของที่เก็บ

วงจร



รูปที่ 3.3 แสดงการแบ่งส่วนภายในของตู้และส่วนที่ 1 ที่เป็นส่วนไว้ใช้เก็บสิ่งของ



รูปที่ 3.4 แสดงส่วนที่ 2 ที่เป็นส่วนไว้เก็บวงจร

3.2.1.1 การออกแบบวงจร

การออกแบบวงจร (รูปที่ 3.5 แสดงวงจรตู้นิรภัยสแกนลายนิ้วมือ) เริ่มต้นจากวงจรที่จะนำมาควบคุมกลอนแม่เหล็กไฟฟ้าในการเปิดปิดตู้นิรภัยสแกนลายนิ้วมือ โดยการรับสัญญาณจากการสแกนลายนิ้วมือเมื่อสแกนลายนิ้วมือ Finger scan ที่ต่อกับคอมพิวเตอร์จะรับข้อมูลเพื่อไปเปรียบเทียบกับข้อมูลใน Microsoft Access เมื่อลายนิ้วมือตรงกับข้อมูลที่เก็บไว้ แล้วจะส่งสัญญาณไปยังบอร์ด Arduino แล้วจะรับค่าที่ส่งมาจากฐานข้อมูล (Logic 1 ถ้าลายนิ้วมือถูกต้อง) แล้วส่งสัญญาณออกจาก Pin (ขาในบอร์ด Arduino) ที่ได้ตั้งค่าไว้ ไปยัง Relay และให้ขา NO ของ Relay ทำงาน ทำให้มีการจ่ายไฟไปยังกลอนแม่เหล็ก ทำให้ตู้เปิดออก แรงดันไฟได้จากเมื่อแรงดันจาก 220V AC ไหลเข้าห้อมแปลงชนิดแปลงแรงดันลง (Step-down Transformer) เป็น 12V DC



รูปที่ 3.5 แสดงวงจรตู้นิรภัยสแกนลายนิ้วมือ

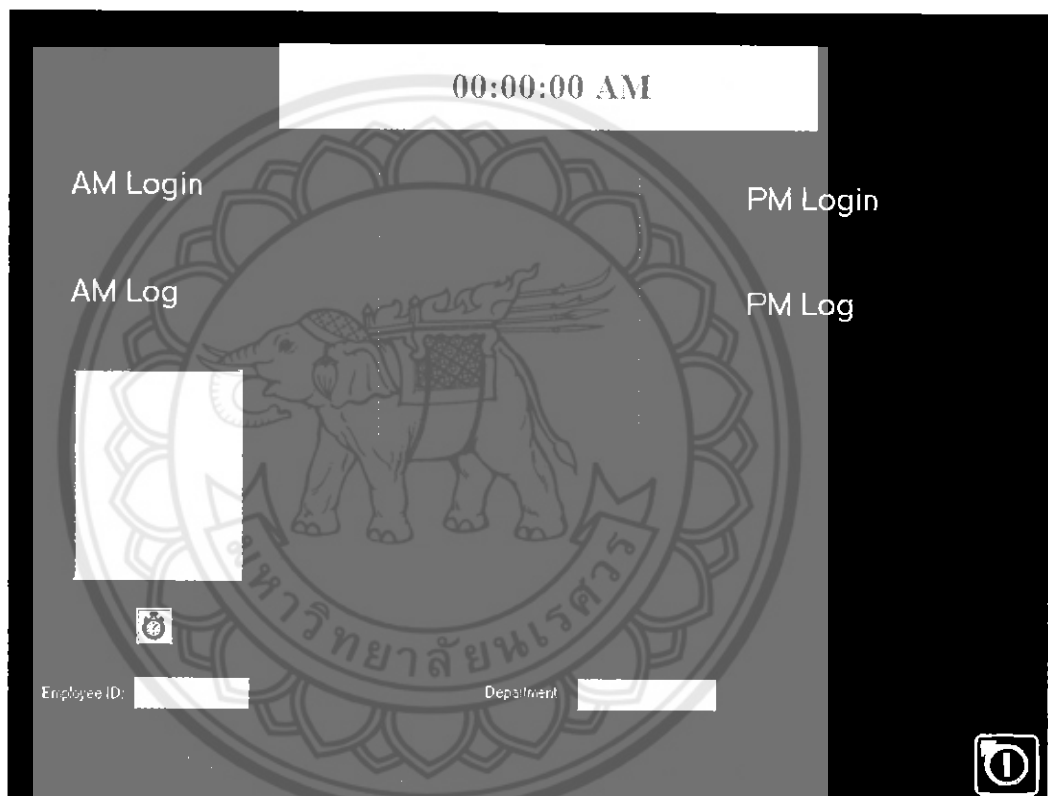
3.2.2 การออกแบบซอฟต์แวร์ (Software)

การทำงานของไมโครคอนโทรลเลอร์บอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P จะถูกสั่งการด้วยโปรแกรม Visual Basic Studio 6.0 ดังแสดงไว้ในแผนผังลำดับการทำงาน ของระบบการเปิด-ปิดตู้ด้วยการสแกนลายนิ้วมือ (รูปที่ 3.1)

เริ่มต้นโปรแกรมด้วยการ Run โปรแกรม Visual Basic Studio 6.0 เพื่อเปิดหน้าต่างโปรแกรม (รูปที่ 3.6) ขึ้นมา จากนั้นสแกนลายนิ้วมือผ่าน Finger scan แล้วไมโครคอนโทรลเลอร์ บอร์ด Arduino รุ่น

Duemilanove 2009 ATMEGA 328P จะรับค่ามานำมาเทียบกับข้อมูลในฐานข้อมูลถ้าตรงจะส่งสัญญาณไปยัง OPTO-Relay เพื่อเปิดตู้นิรภัยสแกนลายนิ้วมือ

หลังจากที่ได้ทำการออกแบบระบบการเปิด-ปิดตู้เอกสารด้วยการสแกนลายนิ้วมือ ทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์เรียบร้อยแล้ว จึงได้นำมาทดสอบใช้งานเพื่อสังเกตว่าการทำงานของระบบการเปิด-ปิดตู้เอกสารด้วยการสแกนลายนิ้วมือทำงานได้อย่างถูกต้องและมีข้อผิดพลาดเกิดขึ้นหรือไม่



รูปที่ 3.6 แสดงตัวอย่างหน้าต่างโปรแกรม

3.3 การอธิบายการทำงานส่วนต่างๆของวงจร

เป็นการอธิบายการทำงานส่วนต่างๆของวงจรเรกติไฟเออร์ (Rectifier) หรือวงจรเรียงกระแสวงจรเรียงกระแส วงจร OPTO-Relay และ Arduino ติดต่อกับคอมพิวเตอร์

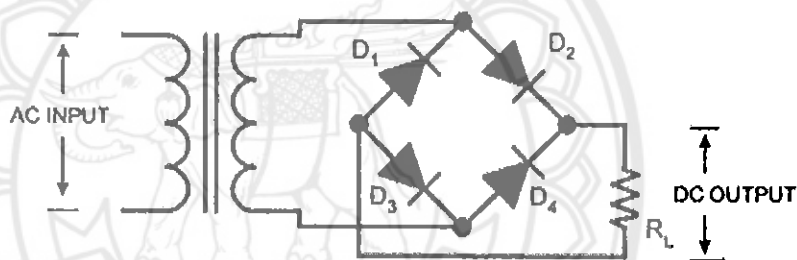
3.3.1 วงจรเรกติไฟเออร์ (Rectifier) หรือวงจรเรียงกระแสวงจรเรียงกระแส

วงจรเรียงกระแส (Rectifier Circuit) (รูปที่ 3.13) คือวงจรไฟฟ้าที่มีคุณสมบัติในการแปลงไฟฟ้ากระแสสลับ (AC Voltage) ให้เป็นสัญญาณไฟฟ้ากระแสตรง (DC Voltage) ที่ขั้วขาออก

อุปกรณ์ที่นิยมใช้ในการทำวงจรเรียงกระแสคือไดโอด โดยอาศัยคุณสมบัติของไดโอดที่ยอมให้กระแสไหลผ่านทางเดียววงจรเรียงกระแสมีอยู่ 2 ชนิด คือวงจรเรียงกระแสแบบครึ่งคลื่นและวงจรเรียงกระแสแบบเต็มคลื่น ในการอธิบายวงจรเรียงกระแสแบบเต็มคลื่นวงจรเรียงกระแสแบบนี้จะใช้ไดโอด 4 ตัวต่ออยู่ในวงจรเพื่อป้องกันไม่ให้ไดโอดสามารถนำกระแสได้ในแต่ละครึ่งไซเคิลของกระแสสลับไดโอดทั้ง 4 ตัวทำหน้าที่เป็นตัวจ่ายกระแสให้กับ R_L (ความต้านทาน โหลด) ตัวละครึ่งไซเคิลและจ่ายในทิศทางเดียวกัน

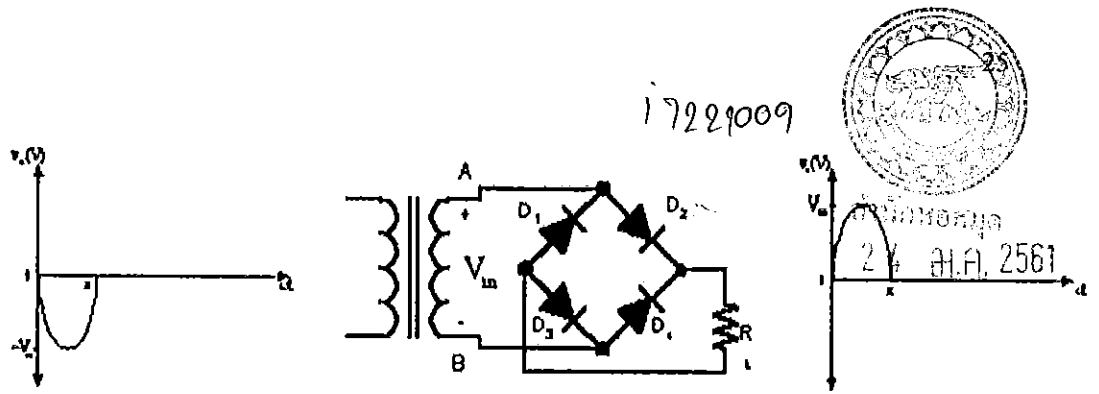
วงจรเรียงกระแสแบบบริดจ์ (Bridge Rectifier Circuit)

วงจรเรียงกระแสแบบเต็มคลื่น (Full Wave) วงจรเรียงกระแสแบบบริดจ์ (Bridge Rectifier Circuit) แรงดันไฟสลับจะต่อเข้ากับสองมุมของวงจรบริดจ์และเอาต์พุตจะถูกนำออกที่สองมุมที่เหลือดังในรูปที่ 3.7

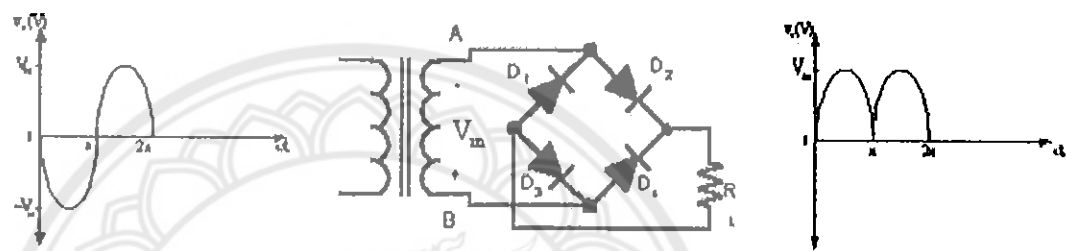


รูปที่ 3.7 วงจรเรียงกระแสแบบบริดจ์

ในแต่ละครึ่งไซเคิลของวงจรอินพุตสมมุติว่าเมื่อขั้ว A ของขดทุติยภูมิมีค่าเป็นบวก และขั้ว B มีค่าเป็นลบจึงเหมือนกับครึ่งไซเคิลถูกป้อนเข้าทางขดปฐมภูมิของหม้อแปลง ไดโอด D2 และ D3 จะอยู่ในลักษณะไบอัสตรงดังนั้นกระแสจึงไหลครบวงจรจากขั้ว A ผ่านไดโอด D2 ความต้านทานโหลดและ ไดโอด D3 แล้วกลับเข้าสู่ขั้ว B ของหม้อแปลงดังรูปที่ 3.8 ก. และเมื่อแรงดันไฟสลับเปลี่ยนข้างเป็นขั้วบวกที่ขั้ว B และเป็นลบที่ขั้ว A การนำกระแสของไดโอดจะเปลี่ยนไปโดยเริ่มจากจุด B ของขดทุติยภูมิผ่าน D4 ความต้านทานโหลดและ D1 กลับเข้าสู่ขั้ว A ของหม้อแปลง ทิศทางแรงดันตกคร่อม โหลดจะมีทิศทางเดียวกับตอนแรกคือ มีขั้วบวกอยู่ทางด้านบน ดังรูปที่ 3.8 ข. ดังนั้นการนำกระแสไดโอดจะเกิดสลับกันทีละสองตัว D2 กับ D3 และ D1 กับ D4



รูปที่ 3.8 (ก) แสดงการทำงานของวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์เมื่ออินพุตเป็น
ซีกลบ



รูปที่ 3.8 (ข) แสดงการทำงานของวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์เมื่ออินพุตเป็น
ซีกบวก

ค่าพารามิเตอร์ในวงจรเรียงกระแสแบบเต็มคลื่น

ในวงจรเรียงกระแสแบบครึ่งคลื่น จะได้สัญญาณเอาต์พุตเพียงครึ่งไซเคิล แต่ในวงจรเรียงกระแสแบบเต็มคลื่น จะได้สัญญาณเอาต์พุตทุกรูปคลื่นของสัญญาณอินพุต ดังนั้นค่าแรงดันหรือค่ากระแสย่อมได้มากกว่าแบบครึ่งคลื่น การคำนวณหาค่าโดยใช้สมการเดียวกันแต่จะให้ผลลัพธ์ดังนี้

$$I_{dc} = \frac{2I_m}{\pi} = 0.636I_m \quad (3.1)$$

$$V_{dc} = \frac{2V_m}{\pi} = 0.636V_m \quad (3.2)$$

$$I_{rms} = \frac{I_m}{\sqrt{2}} \quad (3.3)$$

$$V_{rms} = \frac{V_m}{\sqrt{2}} \quad (3.4)$$

แรงดันย้อนกลับ (Peak Inverse Voltage – PIV)

PIV เป็นค่าแรงดันย้อนกลับสูงสุดที่ตกคร่อมไดโอด ขณะที่ได้รับไบอัสกลับไดโอดที่ใช้ จะต้องไม่แรงดันทลาย (Breakdown) สูงกว่า PIV

PIV ของวงจรเรียงกระแสแบบเต็มคลื่นแบบบริดจ์เท่ากับ V_m

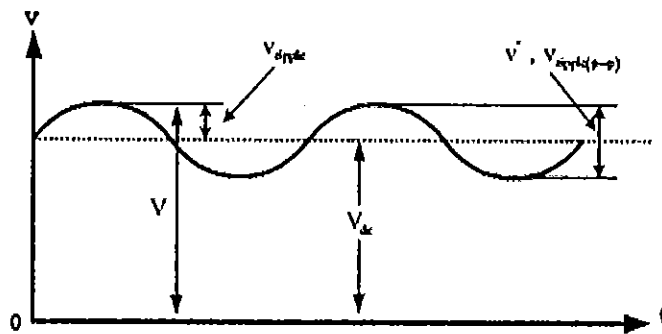
ตัวประกอบค่าระลอก (Ripple Factor)

เนื่องจากสัญญาณเอาต์พุตของวงจรเรียงกระแส มีทั้งองค์ประกอบไฟตรง และ องค์ประกอบไฟสลับ ค่าของไฟตรงหมายถึงค่าเฉลี่ย ค่าไฟสลับเป็นเสมือนระลอกคลื่นที่แปรขึ้นลง รอบ ๆ ค่าเฉลี่ยนี้ อัตราส่วนของค่าไฟสลับต่อไฟตรงเรียกว่า ตัวประกอบค่าระลอกและมีนิยามดังนี้

$$\begin{aligned} \text{ตัวประกอบค่าระลอก (r)} &= \frac{\text{ค่าอาร์เอ็มเอสของส่วนประกอบสลับ}}{\text{ค่าของส่วนประกอบกระแสตรง}} \\ &= \frac{V_r(\text{rms})}{V_{DC}} \\ &= \frac{I_r(\text{rms})}{I_{DC}} \end{aligned}$$

หมายเหตุ

$I_r(\text{rms}), V_r(\text{rms})$ หมายถึงค่ายังผลขององค์ประกอบไฟสลับเท่านั้น ส่วน $I'_{r(\text{rms})}, V'_{r(\text{rms})}$ หมายถึงค่ายังผลของสัญญาณออก (รวมไฟตรงและไฟสลับ) ให้ $V_{\text{ripple(p-p)}}$ เป็นสัญญาณดังรูปที่ 3.9 V_{DC} เป็นค่าเฉลี่ยและ V_{ripple} เป็นองค์ประกอบไฟสลับหรือเรียกว่าแรงดันระลอก (Ripple Voltage)



รูปที่ 3.9 แสดงแรงดันที่ประกอบด้วยค่าไฟตรง V_{DC} และค่าไฟสลับ V_{ripple}

จากรูปที่ 3.9 จะได้ความสัมพันธ์ดังนี้

$$\text{ตัวประกอบค่าระลอก } r = \frac{V_{r(\text{rms})}}{V_{\text{DC}}} = \frac{I_{r(\text{rms})}}{I_{\text{DC}}} \quad (3.5)$$

ค่า $V_{r(\text{rms})}$ จะหาค่าได้ดังนี้

$$\begin{aligned} V_{r(\text{rms})} &= \sqrt{\frac{1}{T} \int_0^T (V - V_{\text{dc}})^2 dt} \\ &= \sqrt{\frac{1}{T} \int_0^T (V^2 - 2VV_{\text{dc}} + V_{\text{dc}}^2) dt} \\ &= \sqrt{\frac{1}{T} \int_0^T V^2 dt - \frac{1}{T} \int_0^T 2VV_{\text{dc}} dt + \frac{1}{T} \int_0^T V_{\text{dc}}^2 dt} \\ &= \sqrt{V_{\text{rms}}^2 - 2V_{\text{dc}} \left(\frac{1}{T} \int_0^T V dt \right) + \frac{(V_{\text{dc}}^2)}{T} \int_0^T dt} \\ &= \sqrt{V_{\text{rms}}^2 - 2V_{\text{dc}}^2 + V_{\text{dc}}^2} \\ V_{r(\text{rms})} &= \sqrt{V_{\text{rms}}^2 - V_{\text{dc}}^2} \quad (3.6) \end{aligned}$$

ค่า V ในสมการที่ (3.6) คือแรงดันที่ประกอบด้วยค่าไฟตรง V_{DC} และค่าไฟสลับ V_{ripple} แสดงในรูปที่ 3.9

นำค่า $V_{r(\text{rms})}$ ที่หาได้ไปแทนในสมการที่ (3.5) เพื่อหาค่าระลอกคลื่นได้ดังนี้

$$r = \frac{V_{r(\text{rms})}}{V_{\text{dc}}} = \frac{\sqrt{V_{\text{rms}}^2 - V_{\text{dc}}^2}}{V_{\text{dc}}} \quad (3.7)$$

หรือ

$$r = \frac{I_{r(\text{rms})}}{I_{\text{dc}}} = \frac{\sqrt{I_{\text{rms}}^2 - I_{\text{dc}}^2}}{I_{\text{dc}}} \quad (3.8)$$

สำหรับวงจรเรียงกระแสแบบเต็มคลื่น

$$\begin{aligned} r &= \frac{\sqrt{\left(\frac{V_m}{\sqrt{2}}\right)^2 - \left(\frac{2V_m}{\pi}\right)^2}}{\frac{2V_m}{\pi}} \\ &= \frac{V_m \sqrt{0.5 - 0.405}}{0.636V_m} \\ &= \frac{0.307V_m}{0.635V_m} \\ &= 0.483 \end{aligned}$$

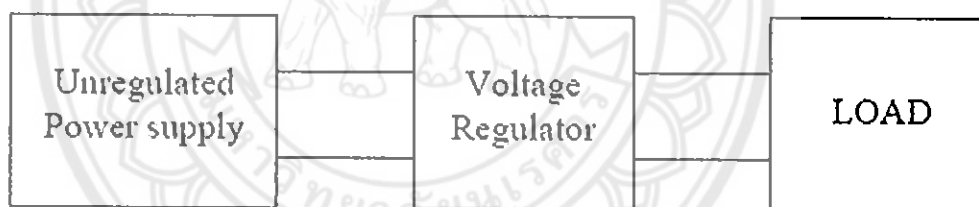
วงจรรักษาระดับแรงดัน (Voltage Regulator Circuits)

วงจรรักษาระดับแรงดัน คือ วงจรที่ต่อระหว่างแหล่งจ่ายไฟตรงไม่คงค่า (Unregulated Power Supply) กับ โหลด มีหน้าที่จ่ายไฟตรงให้กับ โหลดและสามารถรักษาแรงดันให้คงตัวขณะที่

1. โหลดเปลี่ยน คือ กระแสขาออกเปลี่ยนแปลง
2. แรงดันของแหล่งจ่ายไฟตรงไม่คงค่า เปลี่ยนแปลงทั้งนี้รวมถึงระลอกคลื่นด้วย (ripple)
3. อุณหภูมิของวงจรเปลี่ยนแปลง

จากบล็อกไดอะแกรม จะเห็นได้ว่าสัญญาณที่ออกมาจากเอาต์พุตของวงจรกรองแรงดัน คือ บล็อกแรกในรูปที่ 3.10 ซึ่งยังไม่เรียบเท่าที่ควรยังมีการเปลี่ยนแปลงขึ้นๆลงๆในลักษณะคล้ายสัญญาณฟันเลื่อย เพราะว่าตัวเก็บประจุจะเก็บและคายประจุอยู่ตลอดเวลา ถ้านำวงจรนี้ไปใช้งานเลยก็ได้แต่ต้องใช้กับงานที่ไม่ต้องการความเรียบของแรงดันจากแหล่งจ่ายไฟมากนัก

แต่ถ้าเป็นงานที่ต้องการความเรียบของแรงดันมาก วงจรนี้จะใช้งาน ไม่ได้ จะแก้ไขได้โดยใช้วงจรเรกกูเลเตอร์ ซึ่งจะทำหน้าที่ปรับแต่งสัญญาณที่ไม่ราบเรียบ ให้เรียบจนเกือบเป็นไฟกระแสตรง



รูปที่ 3.10 บล็อกไดอะแกรมของแหล่งจ่ายไฟตรงคงค่าแรงดัน

วงจรเรกกูเลเตอร์โดยใช้ไอซี

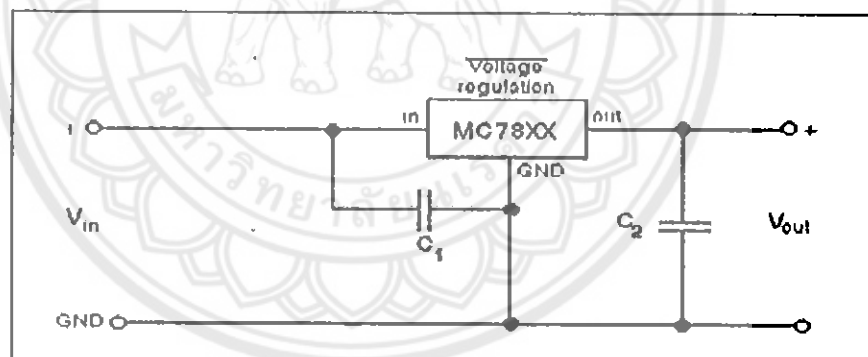
วงจรแหล่งจ่ายไฟสมัยใหม่ที่นิยมกันมาก คือวงจรเรกกูเลเตอร์โดยใช้ไอซี เพราะการออกแบบวงจรง่าย ราคาถูก ขนาดเล็ก และให้คุณภาพสูง ในการออกแบบวงจรเรกกูเลเตอร์โดยใช้ไอซีในลักษณะต่างๆ คือ วงจรเรกกูเลเตอร์โดยใช้ IC หลายขา วงจรเรกกูเลเตอร์โดยใช้ IC 3 ขา แบบแรงดันเอาต์พุตคงที่บวกและลบ และวงจรเรกกูเลเตอร์โดยใช้ไอซี 3 ขาแบบปรับค่าแรงดันเอาต์พุต ค่าที่ได้ของวงจรเรกกูเลเตอร์โดยใช้ไอซี จะได้แรงดันอินพุตมาจากวงจรเรกติไฟเออร์ที่ผ่านการกรองแรงดันมาแล้ว โดยวงจรเรกกูเลเตอร์จะทำหน้าที่ปรับแต่งแรงดันให้เรียบขึ้น และรักษาระดับแรงดันให้คงที่ตลอดการใช้งาน

วงจรเรกกูเลเตอร์ของโครงการตู้รีเลย์สแกนลายนิ้วมือ ใช้ไอซี 3 ขา แบบแรงดันเอาต์พุตคงที่ MC7812 คือ ไอซีตระกูล MC78xx ที่ให้แรงดันแบบบวกคงที่ขนาด 12 V

วงจรเรกกูเลเตอร์พื้นฐานโดยใช้ไอซีตระกูล MC78xx

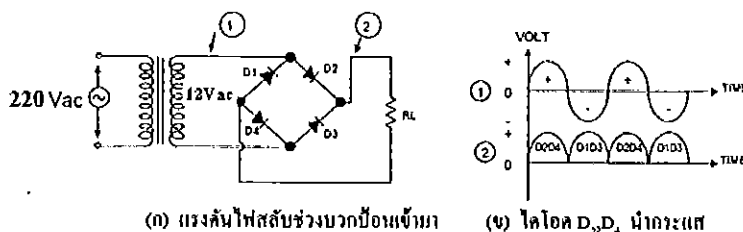
ในรูปที่ 3.11 แสดงวงจรมาตรฐานของแหล่งจ่ายไฟแบบแรงดันเอาต์พุตคงที่ที่ใช้ MC78xx เป็นวงจรเรกกูเลเตอร์ จะเห็นได้ว่าขา in ของ MC78xx จะต่อกับไฟบวก ส่วนขา GND ของ MC78xx จะต่อกับ GND ของ V_{in} ส่วน C_1 ใส่ไว้เพื่อป้องกันสัญญาณรบกวนภายในไอซี ซึ่งมักใช้ค่า $1 \mu\text{F}$ แบบแทนทาลัมหรือ $0.1 \mu\text{F}$ แบบเซรามิก ส่วน C_2 มีไว้เพื่อป้องกันสัญญาณรบกวนซึ่งมักใช้ค่า $1 \mu\text{F}$ แบบแทนทาลัมหรือ $0.1 \mu\text{F}$ แบบเซรามิก

การคงค่าแรงดันที่ขา out ของ MC78xx แรงดันที่ขา in และขา out ของ MC78xx จะต้องมีความต่างกันพอสมควร โดยดูจากค่า $V_{in} - V_{out}$ ใน Data Sheet เช่น MC7812 ค่า $V_{in} - V_{out} = 2.5\text{V}$ ฉะนั้นแรงดันที่ขา in ต้องป้อนมากกว่า 14.5V ขึ้นไป แต่ต้องน้อยกว่าค่า $V_{in(\text{max})}$ ซึ่งจาก Data Sheet IC regulator LM7812 มีค่าเท่ากับ 35V (ภาคผนวก ง หน้า 167)



รูปที่ 3.11 วงจรแหล่งจ่ายไฟโดยใช้ MC78xx

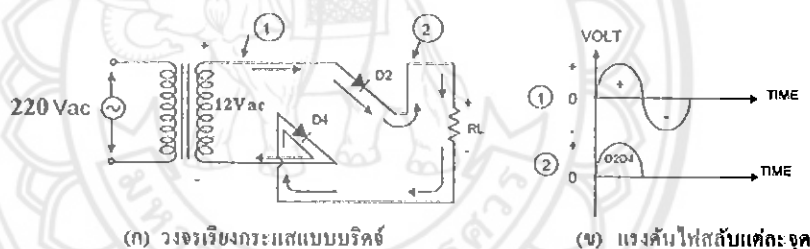
โครงการตู้รีเลย์สแกนลายนิ้วมือนี้ ใช้วงจรเรกติไฟเออร์เต็มคลื่นแบบบริดจ์ จากรูปที่ 3.12 แสดงวงจรเรียงกระแสแบบบริดจ์ และแรงดันไฟสลับที่จุดต่างๆของวงจร โดยแรงดันไฟสลับที่ได้จะผ่านไดโอดที่ทำหน้าที่เรียงกระแส ครั้งละ 2 ตัว D_2, D_4 ทำงานครั้งหนึ่ง และ D_1, D_3 ทำงานอีกครึ่งหนึ่งสลับกันไป



รูปที่ 3.12 วงจรและแรงดันไฟสลับที่จุดต่างๆของวงจรเรียงกระแสแบบบริดจ์

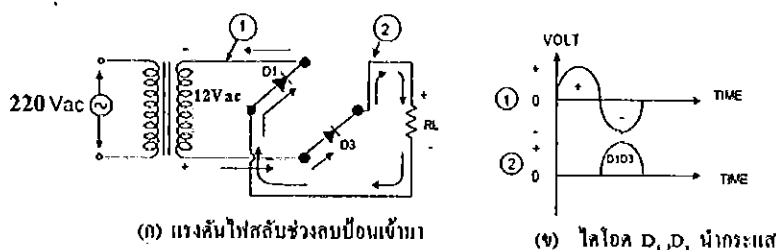
สามารถอธิบายการทำงานแต่ละครั้งได้ดังนี้

จากรูปที่ 3.13 เมื่อจ่ายแรงดันไฟสลับเข้ามาที่ขดลวดของหม้อแปลงทางด้านทุติยภูมิ ไฟเข้าด้านบนขดลวดด้านล่างลบ จ่ายไปอัสให้ไดโอดทั้ง 4 ตัว ไดโอดที่ได้รับไปอัสตรงคือ ไดโอด D_2, D_4 มีกระแสขั้วบวกด้านบน ผ่านไดโอด D_2 ผ่านโหลด R_L ผ่านไดโอด D_4 ครบวงจรที่ขั้วลบด้านล่างได้แรงดันเอาต์พุตเป็นไฟสลับช่วงบวก



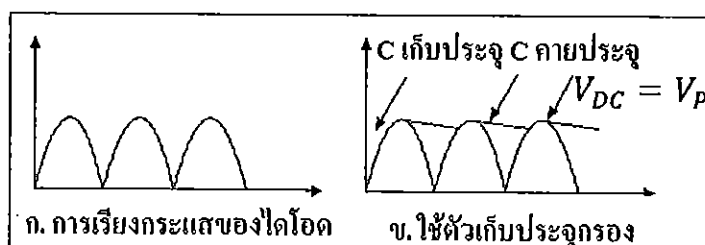
รูปที่ 3.13 ขณะจ่ายแรงดันไฟสลับไปอัสตรงให้ไดโอด D_2, D_4

จากรูปที่ 3.14 เมื่อจ่ายแรงดันไฟสลับเข้ามาที่ขดลวดของหม้อแปลงทางด้านทุติยภูมิ ไฟเข้าด้านบนขดลวดด้านล่างบวก จ่ายไปอัสให้ไดโอดทั้ง 4 ตัว ไดโอดที่ได้รับไปอัสตรงคือ ไดโอด D_1, D_3 มีกระแสขั้วบวกด้านล่าง ผ่านไดโอด D_1 ผ่านโหลด R_L ผ่านไดโอด D_3 ครบวงจรที่ขั้วลบด้านบนได้แรงดันเอาต์พุตเป็นไฟสลับช่วงบวก



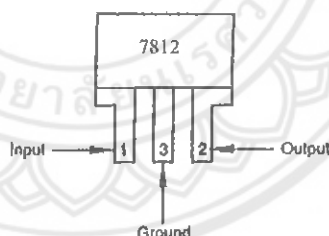
รูปที่ 3.14 ขณะจ่ายแรงดันไฟสลับไปอัสตรงให้ไดโอด D_1, D_3

หลังจากได้เอาต์พุตออกจากวงจรเรียงกระแสแบบบริดจ์แล้ว จะไหลเข้าตัวเก็บประจุเพื่อกรองกระแสให้เรียบขึ้น ดังรูปที่ 3.15



รูปที่ 3.15 ภาพแสดงสัญญาณวงจรเรียงกระแสแบบเต็มคลื่นก่อนและหลังใส่ตัวเก็บประจุ

โดยเริ่มจากเข้า C2 Capacitor ชนิด Electrolyte ขนาด 220 μ F 50V (รูปที่ 3.17) แล้วเข้าตัว C1 Capacitor ชนิด Ceramic ขนาด 3300PF 50V (รูปที่ 3.17) เนื่องจากไฟที่ได้ตอนนี้มีค่าประมาณ 16V DC มีค่าเกิน และไม่เหมาะสมกับการนำมาใช้งานกับกลอนแม่เหล็กไฟฟ้า ดังนั้นจึงต้องใช้ IC Regulator 7812 (รูปที่ 3.16) เพื่อให้ได้แรงดันประมาณ 12V DC (Full Wave) โดยเริ่มจากเข้าขา Input ของ IC Regulator 7812 แล้วออกมาที่ขา Output ของ IC Regulator 7812



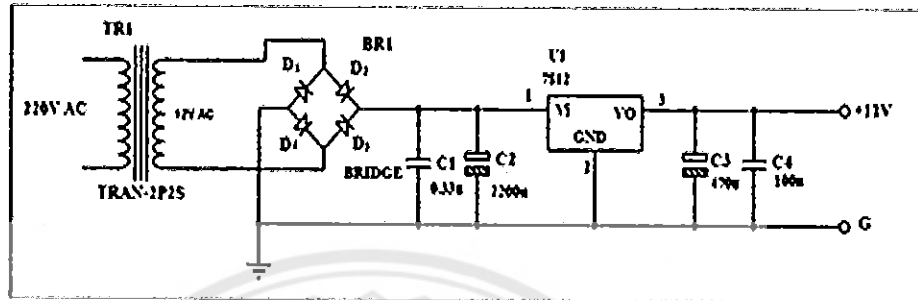
รูปที่ 3.16 ภาพแสดง IC regulator 7812

หลังจากไฟออกมาที่ขา Output ของ IC regulator 7812 แล้วมาเข้าที่ C3 Capacitor ชนิด Electrolyte ขนาด 470 μ F 35V (รูปที่ 3.17) จากนั้นเข้า C4 Capacitor ชนิด Multilayer ขนาด 0.1 μ F 50V (รูปที่ 3.17) เพื่อกรองกระแสให้เรียบมากขึ้น

วงจรเรียงกระแสตัวนี้รับสแกนลายนิ้วมือใช้ Capacitor คนละชนิดกัน และมีค่าต่างกัน C1 เป็น Capacitor ชนิด Ceramic ซึ่งเหมาะสมกับการใช้ลดสัญญาณรบกวน เป็นตัวเก็บประจุที่มีเสถียรภาพสูง กระแสรั่วต่ำ เป็นตัวเก็บประจุชนิดไม่มีขั้วทำให้ไม่ต้องคำนึงเวลาใช้งาน ข้อเสียคือค่าความจุต่ำ การสูญเสียมาก C2 เป็น Capacitor ชนิด Electrolyte แบบมีขั้ว ซึ่งเหมาะสมกับการใช้

งานที่ต้องการชาร์จไฟบ่อยๆ และเมื่อต้องปล่อยประจุครั้งละมากๆ จะเกิดความร้อนน้อย มีค่าความจุสูง เหมาะสมกับการใช้กรองกระแสให้เรียบ ข้อเสียคือกระแสสูง

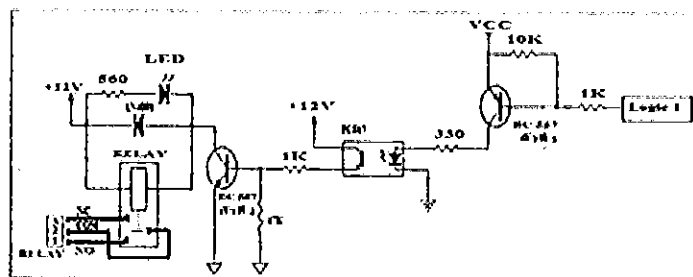
การนำตัวเก็บประจุมาต่อขนานกัน มีผลทำให้การเก็บประจุรวมมีค่ามากขึ้น อัตราทนแรงดันสูงสุดของวงจร มีค่าเท่ากับตัวที่มีอัตราทนแรงดันได้น้อยที่สุด



รูปที่ 3.17 วงจรเรียงกระแสของตู้นิรภัยสแกนลายนิ้วมือ

3.3.2 วงจร OPTO-Relay

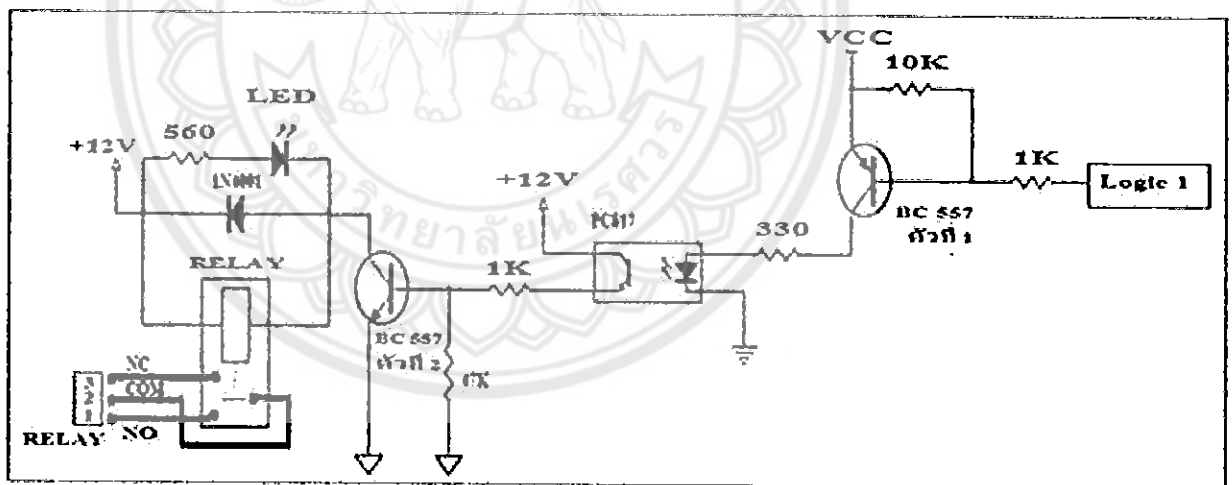
วงจร OPTO-Relay แสดงในรูปที่ 3.18 เป็นรูปแสดงการทำงานของวงจร OPTO-Relay ของตู้นิรภัยสแกนลายนิ้วมือ วงจร OPTO-Relay จะทำงานเมื่อสแกนลายนิ้วแล้วบอร์ด Arduino ได้ส่งสัญญาณ Logic 1 (จะส่ง Logic 1 กรณีที่ลายนิ้วมือได้บันทึกในฐานข้อมูล) เข้ามาที่ OPTO-RELAY แล้วไหลเข้าความต้านทาน 1K โอห์ม แล้วกระแสไฟไหลเข้าขา Bias ของทรานซิสเตอร์เบอร์ BC 557 (ตัวที่ 1) แล้ว VCC เท่ากับ 5V จะไหลมาที่ความต้านทาน 330 โอห์มแล้วไดโอดจะเปล่งแสงออกมาเพื่อให้ OPTO PC 817 ทำงาน โดยที่ตัว OPTO PC 817 จะทำหน้าที่เป็นตัวป้องกันไฟด้าน 12V กับไฟด้าน 5V ไม่ให้เกี่ยวข้องกัน เมื่อ OPTO PC 817 ทำงานแล้ว +12 V จะเริ่มทำงานแล้ว +12V ไหลมาที่ความต้านทาน 1K โอห์มแล้วมาที่ขา Bias ของทรานซิสเตอร์เบอร์ BC 547 (ตัวที่ 2) แล้วมาที่ LED ตัว LED เป็นตัวแสดงสถานะว่า OPTO-RELAY ทำงานแล้วกระแสไฟไหลลงคอยลิ่งเพื่อไม่ให้ไฟ 12V ทำงานไหลเข้าขา NO กับ COM ให้กลอนแม่เหล็กไฟฟ้าเปิดออกในรูปที่ 3.18 ภาพแสดงวงจรของ OPTO-RELAY ขนาด 1 ช่อง



รูปที่ 3.18 ภาพแสดงวงจรของ OPTO-RELAY ขนาด 1 ช่อง

3.3.2 วงจร OPTO-Relay

วงจร OPTO-Relay แสดงในรูปที่ 3.17 เป็นรูปแสดงการทำงานของวงจร OPTO-Relay ของตู้นิรภัยสแกนลายนิ้วมือ วงจร OPTO-Relay จะทำงานเมื่อสแกนลายนิ้วมือแล้วบอร์ด Arduino ได้ส่งสัญญาณ Logic 1 (จะส่ง Logic 1 กรณีที่ลายนิ้วมือได้บันทึกในฐานข้อมูล) เข้ามาที่ OPTO-RELAY แล้วไหลเข้าความต้านทาน 1K โอห์ม แล้วกระแสไฟไหลเข้าขา Bias ของทรานซิสเตอร์เบอร์ BC 557 (ตัวที่ 1) แล้ว VCC เท่ากับ 5V จะไหลมาที่ความต้านทาน 330 โอห์มแล้วไดโอดจะเปล่งแสงออกมาเพื่อให้ OPTO PC 817 ทำงาน โดยที่ตัว OPTO PC 817 จะทำหน้าที่เป็นตัวป้องกันไฟด้าน 12V กับไฟด้าน 5V ไม่ให้เกี่ยวข้องกัน เมื่อ OPTO PC 817 ทำงานแล้ว +12 V จะเริ่มทำงานแล้ว +12V ไหลมาที่ความต้านทาน 1K โอห์มแล้วมาที่ขา Bias ของทรานซิสเตอร์เบอร์ BC 547 (ตัวที่ 2) แล้วมาที่ LED ตัว LED เป็นตัวแสดงสถานะว่า OPTO-RELAY ทำงานแล้วกระแสไฟไหลลงคอยลิ่งเลยเพื่อให้ไฟ 12V ทำงานไหลเข้าขา NO กับ COM ให้กลอนแม่เหล็กไฟฟ้าเปิดออกในรูปที่ 3.13 ภาพแสดงวงจรของ OPTO-RELAY ขนาด 1 ช่อง



รูปที่ 3.17 ภาพแสดงวงจรของ OPTO-RELAY ขนาด 1 ช่อง

3.3.3 Arduino ติดต่อกับคอมพิวเตอร์

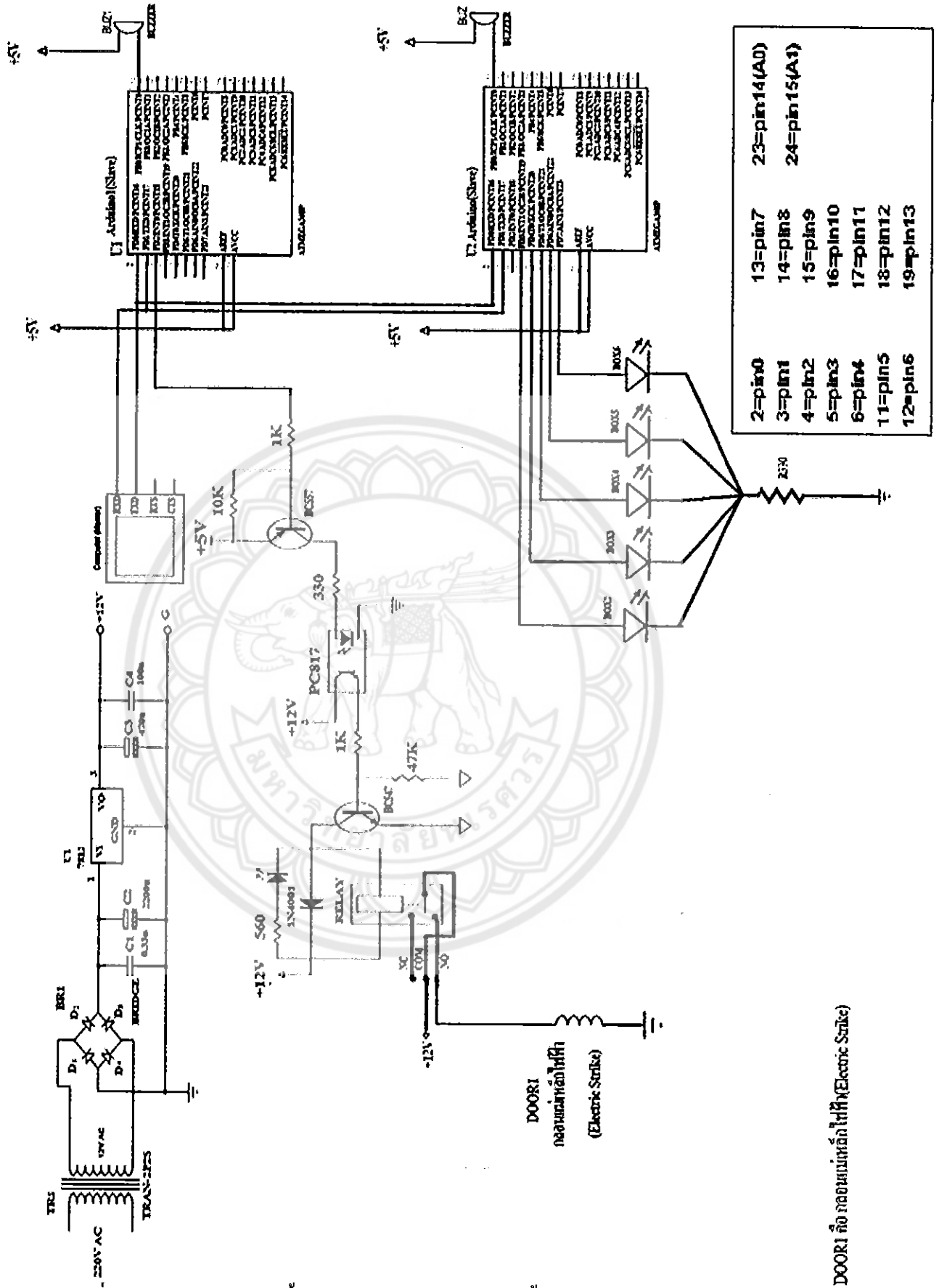
Arduino เป็นเครื่องมือที่จะทำให้คอมพิวเตอร์สามารถรับส่งสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกโดย Arduino จะเป็นตัวกลางที่คอยรับคำสั่งจากคอมพิวเตอร์หรือเมาส์เตอร์ถ้าเกิดสแกนลายนิ้วมือถูกต้องคอมพิวเตอร์จะประมวลผลแล้วเซ็คลายนิ้วมือในฐานข้อมูลถ้าลายนิ้วมือถูกต้องจะส่งสัญญาณไปยัง Arduino และจะมีลำโพง Buzzer เกิดเสียงขึ้นเมื่อคู้นริภัสสแกนลายนิ้วมือเปิดขึ้นรูปที่ 3.18 เป็นภาพที่แสดงวงจรทั้งหมดของคู้นริภัสสแกนลายนิ้วมือและรูปที่ 3.15 ภาพแสดงการต่ออุปกรณ์ทั้งหมดของคู้นริภัสสแกนลายนิ้วมือ

การสื่อสารระหว่างคอมพิวเตอร์กับ Arduino และในคู้สแกนลายนิ้วมือนี้การสื่อสารจะเป็นการสื่อสารทางเดียว จะส่งจากคอมพิวเตอร์ เข้าไปสั่งงาน Arduino สำหรับการสื่อสารภายนอก Arduino จะใช้พอร์ตที่เรียกว่า Serial หรือ ภาษาไทยเรียกพอร์ตอนุกรม พอร์ต Serial เป็นพอร์ตการสื่อสารการสื่อสารแบบอนุกรมหรือ Serial เป็นส่งข้อมูล โดยใช้เทคนิคการเลื่อนข้อมูล (Shift Bit) ส่งไปที่ละบิตบนสายสัญญาณเส้นเดียวโดยการส่งข้อมูลแบบ Serial และตั้งค่าความเร็วในการรับส่งสัญญาณให้เท่ากันหรือ เรียกว่าตั้งค่า Baud rate รูปแบบของ ข้อมูลจากที่ส่งผ่าน Serial จะมีการเพิ่ม Start bit และ Stop bit เข้าไปเพิ่มจากข้อมูลเดิม

- บิตเริ่มต้น (Start bit) จะมีขนาด 1 บิต
- บิตข้อมูล (Data bit) 8 บิต ข้อมูลที่จะส่ง
- บิตภาวะคู่หรือคี่ (Parity bit) มีขนาด 1 บิต ใช้ตรวจสอบข้อมูล ถ้าข้อมูลที่ได้รับ ไม่สมบูรณ์ นำเข้าค่ามา Check กับ Parity bit จะได้ค่าไม่ตรงกัน
- บิตหยุด (Stop bit) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล
- การติดต่อสื่อสารระหว่างคอมพิวเตอร์กับ Arduino จะมีโมดูล Hardware และมี Library ภายใน ที่ใช้สำหรับติดต่อ Serial ในส่วนของ Hardware ที่ต้องการรับส่งข้อมูลและทำการตั้งค่าอัตราเร็วในการส่งให้เท่ากันหรือการตั้งค่า Baud rate ก็จะสามารถรับส่งข้อมูลกับคอมพิวเตอร์

คำสั่งของ Arduino ที่ใช้ใน Serial port ในการสื่อสารข้อมูล

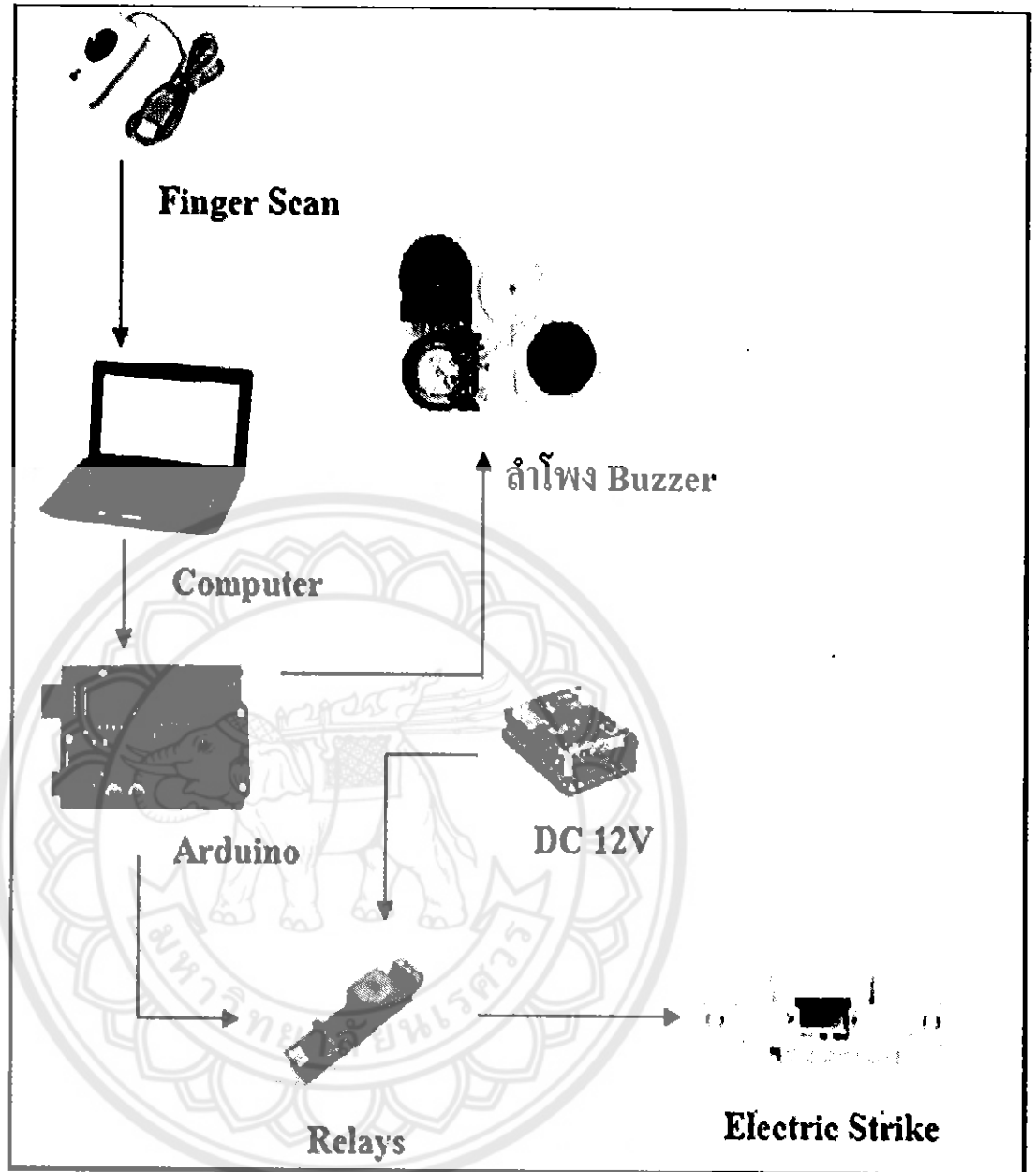
- void serial.begin(rate) กำหนดอัตราบอร์คของการรับส่งข้อมูล หน่วยเป็นบิตต่อวินาที (bits per second :bps baud rate)
- int serial.available() ใช้ตรวจสอบว่า buffer รับข้อมูลไว้หรือไม่ โดยจะคืนค่าจำนวนไบต์ที่อยู่ในบัฟเฟอร์
- int serial.read() ส่งค่าที่รับจากพอร์ตอนุกรมออกมา (Serial)



รูปที่ 3.18 ภาพแสดงวงจรทั้งหมดของตู้นิรภัยสแกนลายนิ้วมือประกอบด้วยวงจร Regulator วงจร OPTO-RELAY และ Arduino ต่อกับ Computer และลำโพง Buzzer

DOORI คือ กอนเนทท์ไฟฟ้า(Electric Strike)

DOORI
กอนเนทท์ไฟฟ้า
(Electric Strike)



รูปที่ 3.19 ภาพแสดงการต่ออุปกรณ์ทั้งหมดของตู้নিরক্ষসแกนลายนิ้วมือ

บทที่ 4

การทดสอบและผลการเปิดตู้เอกสารที่ได้

ในบทนี้ จะกล่าวถึงการทดสอบ ผลการทดสอบ และผลการเปิดตู้นิรภัยสแกนลายนิ้วมือที่ได้ของระบบการเปิดตู้ด้วยการสแกนลายนิ้วมือ ในการวัดประสิทธิภาพการทำงานของระบบการเปิดตู้ด้วยการสแกนลายนิ้วมือโดยได้ออกแบบการทดสอบไว้ดังนี้

1. ทดสอบวัดประสิทธิภาพและความถูกต้องในการทำงานของระบบการเปิดตู้นิรภัยด้วยการสแกนลายนิ้วมือ

1.1 ทดสอบจากลายนิ้วมือที่ได้ทำการลงทะเบียนไว้

1.2 ทดสอบจากลายนิ้วมือที่ไม่ได้ทำการลงทะเบียนไว้

2. ทดสอบความถูกต้องในการทำงานของลำโพงเมื่อมีการสแกนลายนิ้วมือ

2.1 ทดสอบจากลายนิ้วมือที่ได้ทำการลงทะเบียนไว้

2.2 ทดสอบจากลายนิ้วมือที่ไม่ได้ทำการลงทะเบียนไว้

4.1 ทดสอบการทำงานจากลายนิ้วมือที่ได้ทำการลงทะเบียนและไม่ได้ทำการลงทะเบียนไว้

ในการทดสอบนี้จะเป็นการทดสอบความถูกต้องในการทำงานของระบบเปิดตู้เอกสารด้วยการสแกนลายมือ โดยใช้การเปิดตู้ด้วยการสแกนลายนิ้วมือตามจำนวนครั้งที่กำหนด แล้วสังเกตว่ามีการทำงานที่ผิดพลาดเกิดขึ้นหรือไม่

4.1.1 ผลการทดสอบการทำงานจากลายนิ้วมือที่ได้ทำการลงทะเบียนและไม่ได้ทำการลงทะเบียนไว้

ผู้จัดทำได้ทำการทดสอบการเปิดตู้ด้วยการสแกนลายนิ้วมือจากลายนิ้วมือที่ได้ทำการลงทะเบียนและไม่ได้ทำการลงทะเบียนไว้ทั้งหมด 10 ครั้ง โดยแสดงผลการทดสอบจากลายนิ้วมือที่ได้ทำการลงทะเบียนและไม่ได้ลงทะเบียนไว้ในตารางที่ 4.1 และ 4.2

ตารางที่ 4.1 แสดงการทดสอบการเปิดตู้เอกสารจากถาดนิ้วมือที่ได้ทำการลงทะเบียนไว้

ครั้งที่	ตู้เอกสาร (BOX)					
	BOX 1	BOX 2	BOX 3	BOX 4	BOX5	BOX6
1						
2	O	O	O	O	O	O
3	O	O	O	O	O	O
4	O	O	O	O	O	O
5	O	O	O	O	O	O
6	O	O	O	O	O	O
7	O	O	O	O	O	O
8	O	O	O	O	O	O
9	O	O	O	O	O	O
10	O	O	O	O	O	O

*หมายเหตุ: สัญลักษณ์ในตาราง O แทนการทำงานที่ถูกต้อง (ในกรณีนี้คือ สามารถเปิดตู้ได้)

X แทนการทำงานที่ไม่ถูกต้อง (ในกรณีนี้คือ ไม่สามารถเปิดตู้ได้)

จากผลการทดสอบ ที่ได้ดังแสดงในตารางที่ 4.1 พบว่า การทำงานทั้งหมดตลอดช่วงเวลาการทดสอบ การเปิดตู้เอกสารด้วยการสแกนถาดนิ้วมือจากถาดนิ้วมือที่ได้ทำการลงทะเบียนไว้ จากทั้งหมด 6 ตู้ ทำงานครบทุกตู้ทำงานได้อย่างปกติ

ตารางที่ 4.2 แสดงการทดสอบการเปิดตู้เอกสารจากถาดนิ้วมือที่ไม่ได้ทำการลงทะเบียนไว้

ครั้งที่	ตู้เอกสาร (BOX)					
	BOX 1	BOX 2	BOX 3	BOX 4	BOX5	BOX6
1						
2	X	X	X	X	X	X
3	X	X	X	X	X	X
4	X	X	X	X	X	X
5	X	X	X	X	X	X
6	X	X	X	X	X	X
7	X	X	X	X	X	X
8	X	X	X	X	X	X
9	X	X	X	X	X	X
10	X	X	X	X	X	X

*หมายเหตุ: สัญลักษณ์ในตาราง O แทนการทำงานที่ถูกต้อง (ในกรณีนี้คือ สามารถเปิดตู้ได้)

X แทนการทำงานที่ไม่ถูกต้อง (ในกรณีนี้คือ ไม่สามารถเปิดตู้ได้)

จากผลการทดสอบ ที่ได้ดังแสดงในตารางที่ 4.2 พบว่า การทำงานทั้งหมดตลอดช่วงระยะเวลาการทดสอบ การเปิดตู้เอกสารด้วยการสแกนลายนิ้วมือจากถาดนิ้วมือที่ไม่ได้ทำการลงทะเบียนไว้ จากทั้งหมด 6 ตู้ ทำงานครบทุกตู้ทำงานได้อย่างปกติ

4.2 ทดสอบการทำงานของลำโพงเมื่อทำการสแกนลายนิ้วมือ

ในการทดสอบนี้จะเป็นการทดสอบความถูกต้องในการทำงานของลำโพงเมื่อทำการสแกนลายมือ เพื่อใช้ในการเปิดตู้ด้วยการสแกนลายนิ้วมือจากถาดนิ้วมือที่ได้ทำการลงทะเบียนและไม่ได้ทำการลงทะเบียนไว้ตามจำนวนครั้งที่กำหนด แล้วสังเกตว่ามีการทำงานที่ผิดพลาดเกิดขึ้นหรือไม่

4.2.1 ผลการทดสอบการทำงานของลำโพงเมื่อทำการสแกนลายนิ้วมือ

ผู้จัดทำได้ทำการทดสอบการทำงานของลำโพงว่ามีเสียงเตือนเมื่อทำการเปิดตู้ด้วยการสแกนลายนิ้วมือจากถาดนิ้วมือที่ได้ทำการลงทะเบียนและไม่ได้ทำการลงทะเบียนไว้ทั้งหมด 10 ครั้ง โดยแสดงผลการทดสอบจากถาดนิ้วมือที่ได้ทำการลงทะเบียนและไม่ได้ลงทะเบียนไว้ในตารางที่ 4.3 และ 4.4

ตารางที่ 4.3 แสดงผลการสอบการทำงานของลำโพงเมื่อทำการเปิดตู้จากถาดนิ้วมือที่ได้ทำการลงทะเบียนไว้

ครั้งที่	ตู้เอกสาร (BOX)					
	BOX 1	BOX 2	BOX 3	BOX 4	BOX5	BOX6
1						
2	O	O	O	O	O	O
3	O	O	O	O	O	O
4	O	O	O	O	O	O
5	O	O	O	O	O	O
6	O	O	O	O	O	O
7	O	O	O	O	O	O
8	O	O	O	O	O	O
9	O	O	O	O	O	O
10	O	O	O	O	O	O

*หมายเหตุ: สัญลักษณ์ในตาราง O แทนการทำงานที่ถูกต้อง (ในกรณีนี้คือ มีเสียงเตือนจากลำโพง)
X แทนการทำงานที่ไม่ถูกต้อง (ในกรณีนี้คือ ไม่มีเสียงเตือนจากลำโพง)

จากผลการทดสอบ ที่ได้ดังแสดงในตารางที่ 4.3 พบว่า การทำงานทั้งหมดตลอดช่วงเวลาการทดสอบทำงานของลำโพงเมื่อทำการเปิดตู้เอกสารด้วยการสแกนถาดนิ้วมือจากถาดนิ้วมือที่ได้ทำการลงทะเบียนไว้ จากทั้งหมด 6 ผู้ทำงานครบทุกผู้ทำงานได้อย่างปกติ

ตารางที่ 4.4 แสดงผลการสอบการทำงานของลำโพงเมื่อทำการเปิดตู้จากถายนิ้วมือที่ได้ไม่ทำการลงทะเบียนไว้

ครั้งที่	ตู้เอกสาร (BOX)					
	BOX 1	BOX 2	BOX 3	BOX 4	BOX5	BOX6
1						
2	X	X	X	X	X	X
3	X	X	X	X	X	X
4	X	X	X	X	X	X
5	X	X	X	X	X	X
6	X	X	X	X	X	X
7	X	X	X	X	X	X
8	X	X	X	X	X	X
9	X	X	X	X	X	X
10	X	X	X	X	X	X

*หมายเหตุ: สัญลักษณ์ในตาราง O แทนการทำงานที่ถูกต้อง (ในกรณีนี้คือ มีเสียงเตือนจากลำโพง)

X แทนการทำงานที่ไม่ถูกต้อง (ในกรณีนี้คือ ไม่มีเสียงเตือนจากลำโพง)

จากผลการทดสอบ ที่ได้ตั้งแสดงในตารางที่ 4.4 พบว่า การทำงานทั้งหมดตลอดช่วงเวลาการทดสอบการทำงานของลำโพงเมื่อทำการเปิดตู้เอกสารด้วยการสแกนถายนิ้วมือจากถายนิ้วมือที่ไม่ได้ทำการลงทะเบียนไว้ จากทั้งหมด 6 ตู้ ทำงานครบทุกตู้ทำงานได้อย่างปกติ

บทที่ 5

บทสรุป

จากผลการทดลองระบบการเปิด-ปิดประตูตู้นิรภัยสแกนลายนิ้วมือในบทที่ 4 นำมาสรุปและมีข้อเสนอแนะดังนี้

5.1 สรุปผลการดำเนินงานโครงการ

จากโครงการเรื่องตู้นิรภัยสแกนลายนิ้วมือสามารถหาข้อสรุปจากผลการดำเนินงานดังนี้

1. ระบบการเปิด-ปิดประตูตู้นิรภัยสแกนลายนิ้วมือสามารถนำมาประยุกต์ใช้งานกับสำนักงานหรือบริษัทได้และใช้สำหรับจำกัดบุคคลได้
2. ทำให้การเก็บเอกสารมีการจำกัดบุคคลที่จะใช้งานและมีความปลอดภัยมากยิ่งขึ้น
3. สามารถเช็ค ชื่อ วัน/เดือน/ปีจำนวนครั้งที่เปิด-ปิดตู้นิรภัยสแกนลายนิ้วมือ

5.2 ปัญหาและแนวทางการแก้ไข

จากการทำโครงการตู้เปิด-ปิดตู้นิรภัยสแกนลายนิ้วมือปัญหาและอุปสรรคต่างๆ ที่ทำให้การดำเนินงานมีความล่าช้า ดังนี้

1. ศึกษาความรู้ในการออกแบบสร้างวงจรต่างๆ ในตู้นิรภัยสแกนลายนิ้วมือ
2. ศึกษาความรู้ในเรื่องของ Microcontroller ของ Arduino และ Visual Basic 6.0

5.3 แนวทางในการพัฒนาโครงการ

1. พัฒนาในการใช้วัสดุอุปกรณ์ที่ใช้ในการสร้างตู้นิรภัยสแกนลายนิ้วมือเพื่อให้ความแข็งแรงและได้มาตรฐานมากยิ่งขึ้น
2. เพิ่มระบบป้องกันต่างๆ เพื่อให้ความปลอดภัยในตู้นิรภัยสแกนลายนิ้วมือ
3. พัฒนาในด้านการบำรุงรักษา ให้ตู้นิรภัยสแกนลายนิ้วมือนั้นไม่ต้องการการบำรุงรักษามากนัก

5.4 ข้อเสนอแนะ

สำหรับจำนวนตู้เอกสาร สามารถติดตั้งเพิ่มเติมได้อีก เมื่อมีความต้องการที่จะใช้งานเพิ่มได้

5.5 ข้อจำกัดของโครงการตู้นิรภัยสแกนลายนิ้วมือ

1. ถ้าเป็นลายนิ้วมือเดียวกันจะทำการเปิด User ที่สมัครไว้ครั้งแรกเนื่องจากฐานข้อมูลของ Microsoft Access จะดึงข้อมูล User ที่สมัครไว้ครั้งแรกมาเปิดตู้นิรภัยสแกนลายนิ้วมือ
2. ตู้นิรภัยสแกนลายนิ้วมือใช้ 1 User ต่อ 1 ตู้เพื่อป้องกันการเปิดตู้พร้อมกันที่หลายๆตู้
3. สามารถเปลี่ยนลายนิ้วมือได้ที่ใช้สมัครตู้นิรภัยสแกนลายนิ้วมือได้โดยแก้ไขฐานข้อมูล Microsoft Access
4. โครงการตู้นิรภัยสแกนลายนิ้วมือใช้บอร์ด Arduino รุ่น Duemilanove 2009 ATMEGA 328P แต่ละบอร์ดของ Arduino จะมี Serial Port ของ Digital Port = 8 Port ของ Analog Port = 6 Port บอร์ด Arduino 1 บอร์ดมีจำนวน Serial Port ทั้งหมด Digital Port รวมกับ Analog Port = 8 Port + 6 Port = 14 Port ตู้โดยจะแบ่ง 1 Port สำหรับต่อลำโพง Buzzer เพื่อให้เกิดเสียงสัญญาณเวลาเปิดตู้นิรภัยสแกนลายนิ้วมือโครงการตู้นิรภัยสแกนลายนิ้วมือใช้ 1 Port ต่อ 1 ตู้เพื่อใช้รหัสแอสกี (ASCII: American Standard Code Information Interchange) 1 รหัสต่อ 1 ตู้ ดังนั้นโครงการตู้นิรภัยสแกนลายนิ้วมือต่อตู้นิรภัยสแกนลายนิ้วมือได้ 13 ตู้
5. ในโครงการตู้นิรภัยสแกนลายนิ้วมือใช้ 1 บอร์ดต่อ 1 ตู้เพื่อเพิ่มระบบป้องกันความปลอดภัยมากยิ่งขึ้น
6. การลงทะเบียนใช้ตู้นิรภัยสแกนลายนิ้วมือ 1 User ต่อ 1 ลายนิ้วมือนั้น

เอกสารอ้างอิง

- [1] หนังสือเรียนรู้ เข้าใจ ใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino
- [2] ความรู้เกี่ยวกับอิเล็กทรอนิกส์ <http://www.electoday.com/>
- [3] บอร์ด Arduino <http://arduino.cc/en/Main/arduinoBoardDuemilanove>
- [4] Code Program <http://www.g2gnet.com/>
- [5] Visual Basic <http://visual-studio-basic.blogspot.com/>
- [6] ภาพสื่อวิดีโอการสอน Visual basic6 , Arduino, Microsoft access จาก <http://www.youtube.com/>





ส่วนประกอบต่างๆของระบบ

1. Microcontroller duemilanove w/ATmega328	ตัวประมวลผลสำหรับสั่งเปิด
Microcontroller ATmega168	Electric Strike
Rs232	ตัวประมวลผลหลักสามารถ รับแรงดันที่ 5V
2. Finger Print	ตัวเรียงสัญญาณและรับ- สัญญาณดิจิทัล
3. Electric Strike	ตัวสแกนลายนิ้วมือ
4. เครื่องสำรองไฟฟ้า UPS	DC 12 V/200mA
5. Power supply line	220v, ขนาดแบตเตอรี่ 850VA/425 W.
6. ตู้สำหรับควบคุมวงจร	220V
7. ตู้สำหรับเปิด/ปิดเก็บของ	
8. หม้อแปลง	500mA 220V / 0 - 12V
9. Voltage regulator circuit	
C เซรามิก	3300PF 50V (332)
C อิเล็กโทรไลต์	470uF 35V
C อิเล็กโทรไลต์	220uF 50V
C มัลติเมเยอร์	0.1uf 50V
IC 7812	
10. หลอดไฟ LED	
11. สายไฟฟ้า	
12. ชุด OPTO-Relay ของ ETT	
Transistor BC557	
Transistor BC547	
R 47k	
R 1k	

R 560Ω

ไดโอด 1N4001

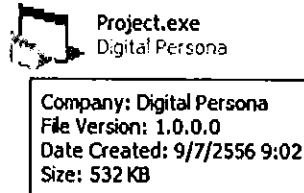
OPTO PC 817



การใช้งานตู้นิรภัยสแกนลายนิ้วมือ

การลงทะเบียนเพื่อใช้งานตู้นิรภัยสแกนลายนิ้วมือ

1. เริ่มต้นการเปิดเข้าโปรแกรม Project.exe ตามรูปที่ 1



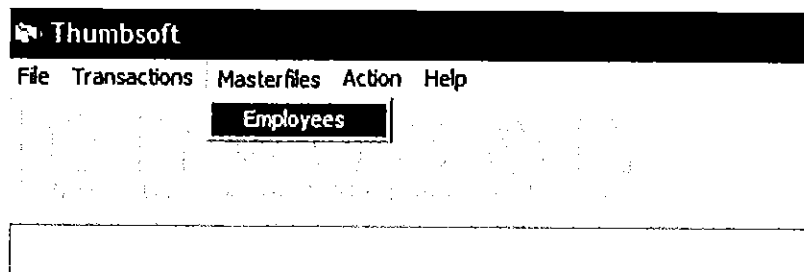
รูปที่ 1 ภาพแสดงตัวโปรแกรม Project.exe

2. รูปตัวโปรแกรมหน้าหลักเมื่อกดเข้ามาตามรูปที่ 2



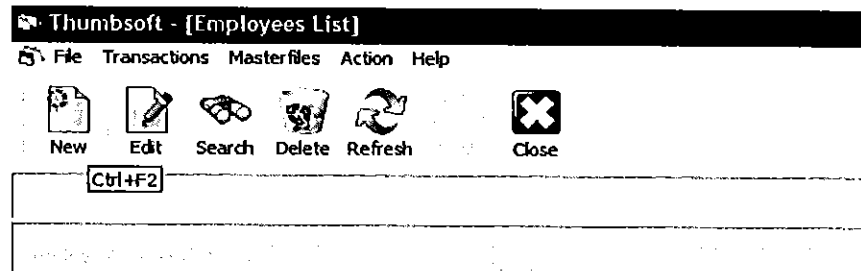
รูปที่ 2 ภาพแสดงรูปตัวโปรแกรมหน้าหลัก

3. กดเข้า Master files แล้วเลือก Employees ตามรูปที่ 3



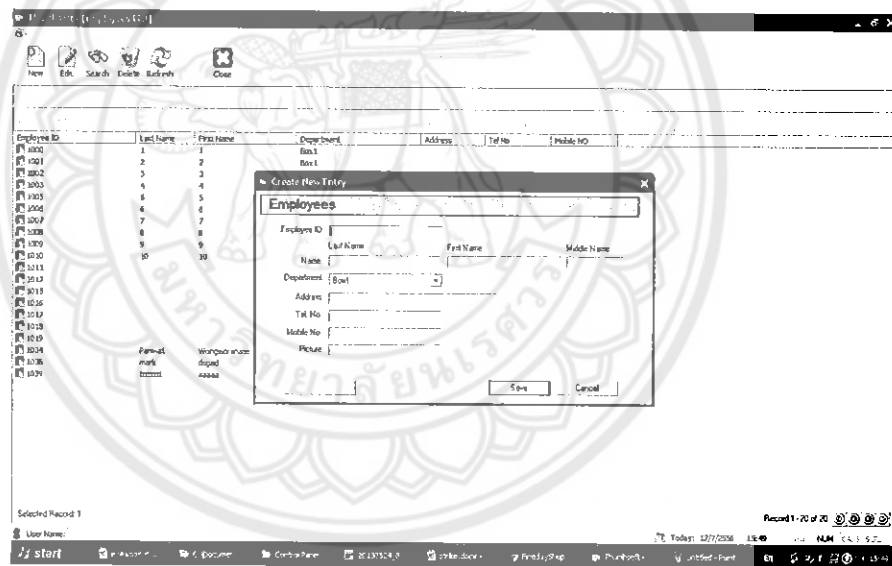
รูปที่ 3 ภาพแสดงหน้าจอที่ทำการสมัครตู้นิรภัยสแกนลายนิ้วมือ

4. กดเข้าที่ New เพื่อทำการสมัครใช้ตู้นิรภัยสแกนลายนิ้วมือตามรูปที่ 4



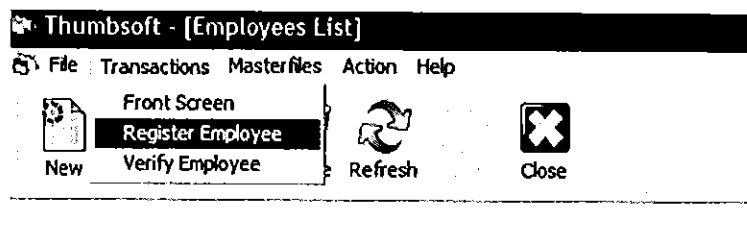
รูปที่ 4 ภาพแสดงเพื่อทำการสมัครสแกนลายนิ้วมือ

5. ทำการกรอกข้อมูล EmployeeID, Name, First name, Middle name, Department, Address, Tel No., Mobile No., Picture ตามรูปที่ 5



รูปที่ 5 ภาพแสดงเพื่อทำการกรอกข้อมูลส่วนบุคคล

6. เมื่อทำการกรอกข้อมูลเสร็จแล้วให้เลือก File>>Register Employee เพื่อทำการบันทึกถายนิ้วมือตามรูปที่ 6



รูปที่ 6 ภาพแสดงหน้าจอเข้าสู่หน้าจอสแกนลายนิ้วมือ

7. หน้าจอ Register User เพื่อทำการบันทึกถายนิ้วมือ ตามรูปที่ 7



รูปที่ 7 ภาพแสดงหน้าจอ Register User

8. เมื่อเข้าหน้า Register User ตรงช่อง Employee ID ที่ให้เข้าหน้า Employee List เพื่อจดจำเลข Employee ID ที่วงไว้ในกรอบสีน้ำเงินรูปที่ 8 เพื่อนำไปกรอกในช่อง Employee ID ในกรอบสีแดง ในรูปที่ 9 จำลองสมัครบุคคลสำหรับเปิดตู้สแกนลายนิ้วมือขึ้นมาในกรอบสีแดงรูปที่ 8 Test 01 Employee ID = 1044

Employee ID	Last Name	First Name	Department
1000	1	1	Box1
1001	2	2	Box1
1002	3	3	Box1
1003	4	4	Box1
1005	5	5	Box1
1006	6	6	Box1
1007	7	7	Box1
1008	8	8	Box1
1009	9	9	Box1
1010	10	10	Box1
1011			Box1
1012			Box1
1015			
1016			
1017			
1018			
1019			
1036	mark	dsged	Box1
1039	ttttttt	aaaaa	Box1
1044	Test01	Test01	Box1

รูปที่ 8 ภาพแสดง Employee ID, Test 01 ที่จำลองขึ้นมาเพื่อสมัครตู้นิรภัยสแกนลายนิ้วมือ

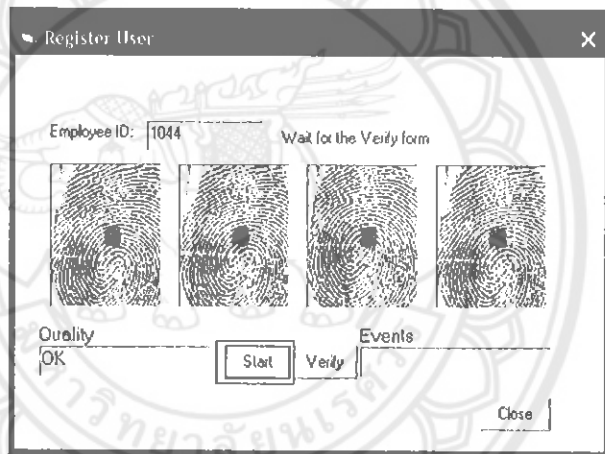
รูปที่ 9 ภาพแสดงกรอบ Employee ID ที่นำตัวเลข Employee ID หน้าจอ Employee List มากรอกใส่ในช่อง Employee ID

9. เมื่อใส่ Employee ID แล้วให้ทำการสแกนลายนิ้วมือกับตัว Finger Scan รูปที่ 10



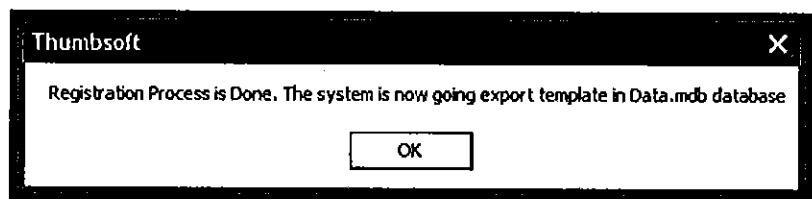
รูปที่ 10 ภาพแสดงตัว Finger scan

10. เมื่อกรอกรหัส Employee ID แล้วกดปุ่ม Start ในกรอบสีแดงตามรูปที่ 11 แล้วทำการสแกนลายนิ้วมือ 4 ครั้งเพื่อทำการเก็บบันทึกลายนิ้วมือบุคคลที่ใช้ศูนย์วิจัยสแกนลายนิ้วมือ



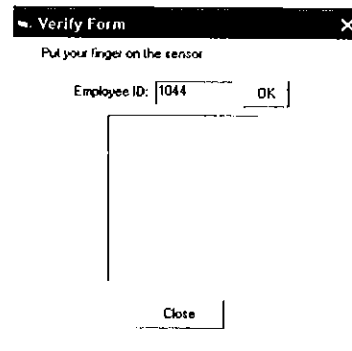
รูปที่ 11 ภาพแสดงหน้าจอที่ทำการเก็บลายนิ้วมือ 4 ครั้ง

11. จะมีหน้าจอขึ้นมาเมื่อทำการสแกนลายนิ้วมือเสร็จ 4 ครั้งตามรูปที่ 12 ลายนิ้วมือที่ได้ทำการสแกนไว้แล้วจะถูกเก็บไว้ไฟล์ที่ชื่อว่า Data.mdb



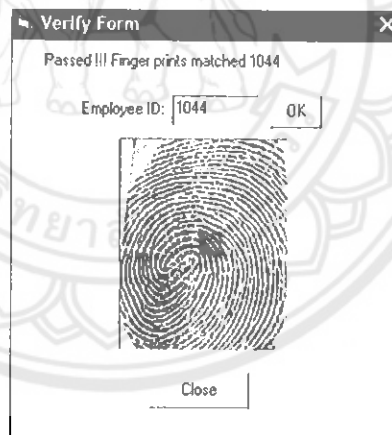
รูปที่ 12 ภาพแสดงการเก็บลายนิ้วมือที่ทำการสแกนไว้ในไฟล์ Data.mdb

12. หน้าจอสำหรับการ Verify ลายนิ้วมือเพื่อตรวจสอบลายนิ้วมืออีกครั้ง ตามภาพที่ 13



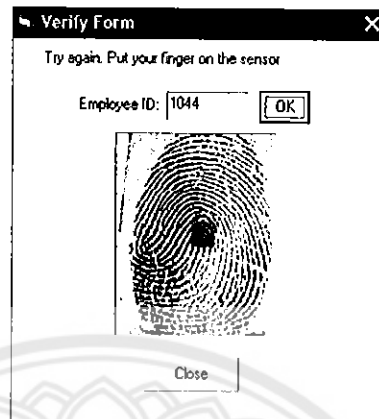
รูปที่ 13 ภาพแสดงหน้าจอ Verify อีก 1 ครั้ง

13. เมื่อทำการ Verify ลายนิ้วมือผ่านจะมีข้อความขึ้นว่า Passed!!! Finger Prints matched 1044 ตามรูปที่ 14 (ตรงหมายเลข 1044 คือหมายเลขเฉพาะบุคคลที่อยู่ในหน้าจอ Employee List ตามกรอบสีน้ำเงินที่ รูปที่ 8)



รูปที่ 14 ภาพแสดงลายนิ้วมือที่ทำการ Verify ผ่าน

14. ถ้าเกิด Verify ลายนิ้วมือไม่ผ่านจะมีข้อความขึ้นว่า Try again.Put your finger on the sensor เพื่อทำการลอง Verify ลายนิ้วมืออีกครั้งตามรูปที่ 15

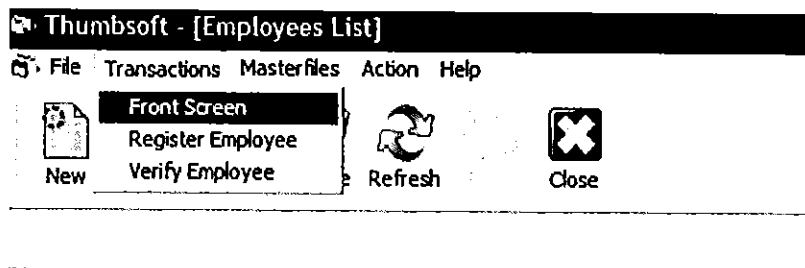


รูปที่ 15 ภาพแสดงการ Verify ลายนิ้วมือไม่ผ่าน



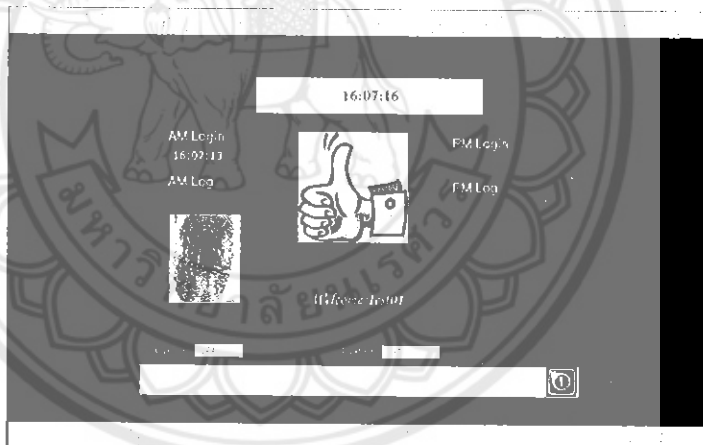
การเปิดตู้นิรภัยสแกนลายนิ้วมือ

1. เข้า Transaction>>Front Screen เพื่อทำการสแกนลายนิ้วมือเพื่อเปิดตู้ตามรูปที่ 1



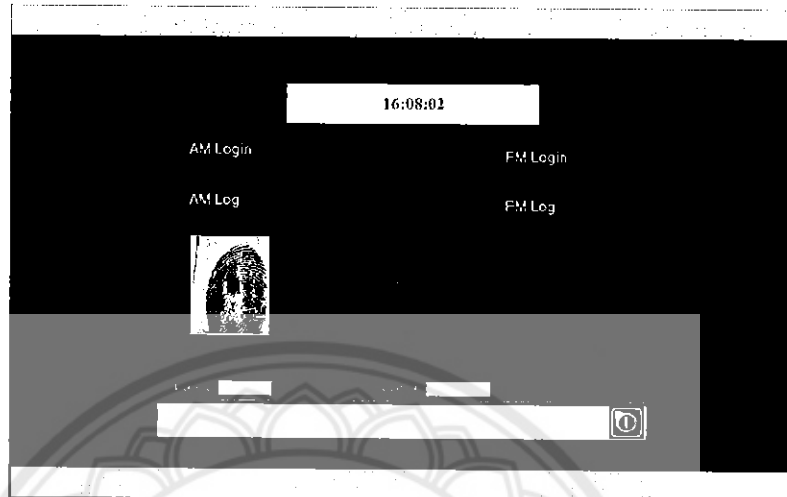
รูปที่ 1 ภาพแสดงการเข้าหน้าจอ Front Screen

2. ทำการสแกนลายนิ้วมือ เพื่อสแกนลายนิ้วมือถูกต้องจะมีข้อความขึ้น Passed!!! Finger Prints matched ตามรูปที่ 2



รูปที่ 2 ภาพแสดงเมื่อสแกนลายนิ้วมือผ่าน

2.1 กรณีสแกนลายนิ้วมือไม่ผ่านจะมีข้อความขึ้นว่า Try again. Put your finger on the sensor ให้ลองทำการสแกนลายนิ้วมือใหม่เพื่อเปิดตู้ ตามรูปที่ 2.1

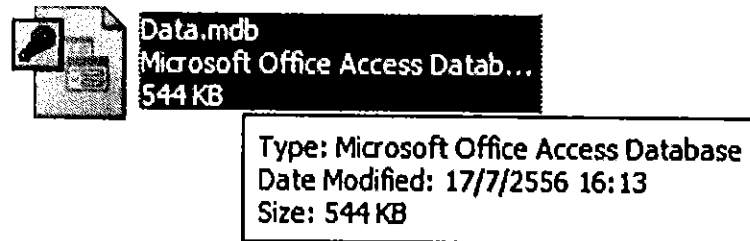


รูปที่ 2.1 ภาพแสดงเมื่อสแกนลายนิ้วมือไม่ผ่าน



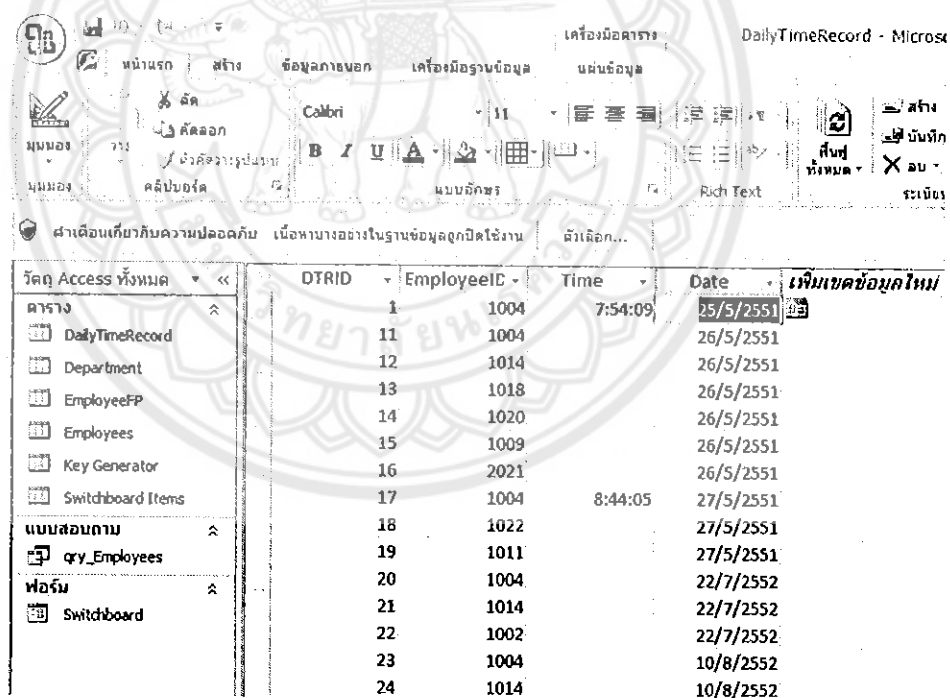
ข้อมูลต่างๆใน Data.mdb

1. ตัวโปรแกรม Data.mdb รูปที่ 1



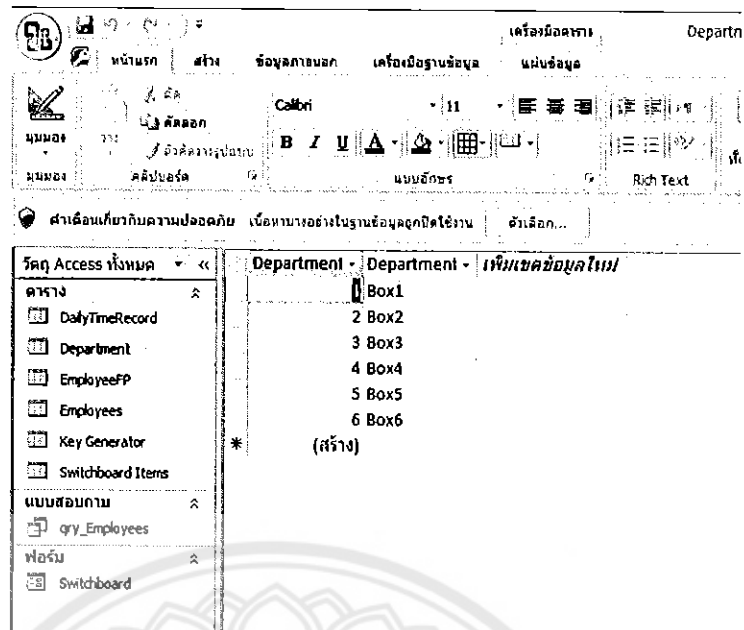
รูปที่ 1 ภาพแสดง Data.mdb

2. หน้าจอในตัวโปรแกรม Data.mdb ส่วนของ DailyTimeRecord ที่ทำการบันทึกวัน/เดือน/ปี เวลา Employee ID ที่ทำการเปิดตู้สแกนลายนิ้วมือ ตามรูปที่ 2



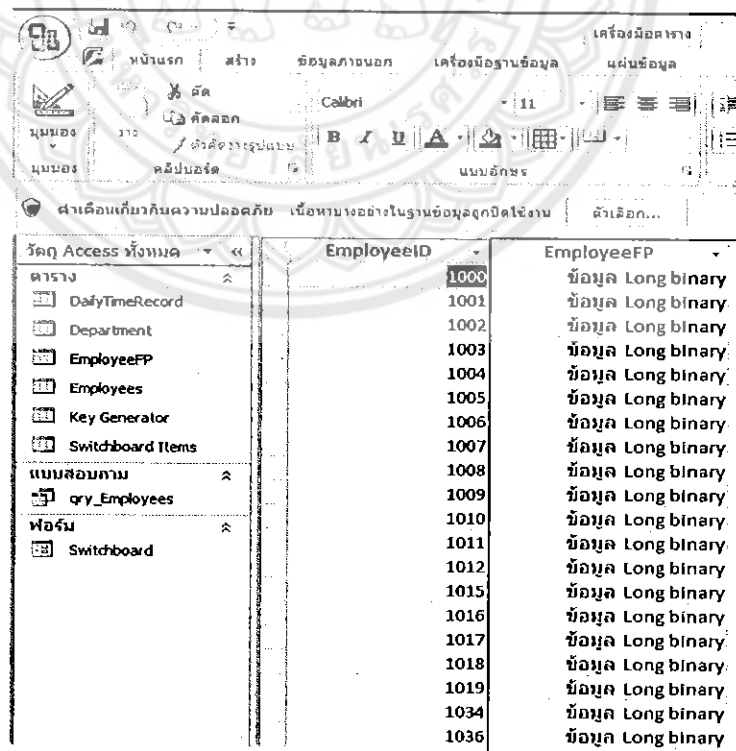
รูปที่ 2 ภาพแสดงหน้าจอ Data.mdb ส่วนของ DailyTimeRecord ใน Data.mdb

3. หน้าจอในส่วนของ Department เป็นหน้าจอที่ใช้สำหรับที่ต้องการจะเพิ่มจำนวนตู้เพิ่มขึ้นหรือต้องการลดจำนวนตู้ เปลี่ยนชื่อตู้ ลบตู้ ตามรูปที่ 3



รูปที่ 3 ภาพแสดงในส่วนหน้าจอของ Department ใน Data.mdb

4. หน้าจอในส่วนของ EmployeeFP เป็นหน้าจอในส่วนที่เก็บข้อมูลลายนิ้วมือที่เก็บข้อมูลเป็นแบบ Long binary ในกรอบสีแดงตามรูปที่ 4



รูปที่ 4 ภาพแสดงข้อมูลลายนิ้วมือของแต่ละบุคคลใน EmployeeFP ใน Data.mdb

5. ในส่วนหน้าจอของEmployee เป็นหน้าจอที่เก็บ Employee ID, Last name, First name, Middle name, Department, Address, Tel No., Mobile No., Picture, วัน/เดือน/ปีที่สมัคร, วัน/เดือน/ปี ที่ทำการแก้ไขตามรูปที่ 5

EmployeeID	LastName	FirstName	MiddleName	Department	Address	TelNo	MobileNo	Picture	DateAdded
1000 1	1			1				Images/1000.j	
1001 2	2			1					
1002 3	3			1					
1003 4	4			1					
1005 5	5			1					
1006 6	6			1					
1007 7	7			1					
1008 8	8			1					
1009 9	9			1				Images/1009.j	
1010 10	10			1					
1011				1					
1012				1					
1015				0					
1016				0					
1017				0					

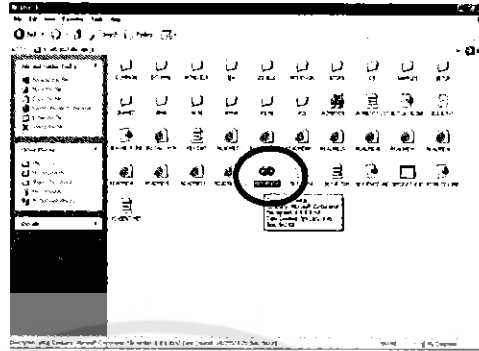
รูปที่ 5 ภาพแสดงหน้าจอส่วน Employee ใน Data.mdb



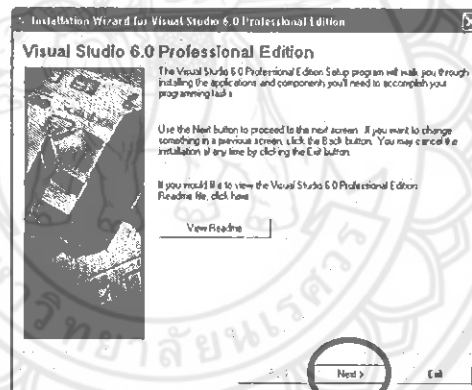
วิธีการใช้งานไมโครคอนโทรลเลอร์ตระกูล ATMEGA 328P

การลงโปรแกรม VB 6.0 Enterprise

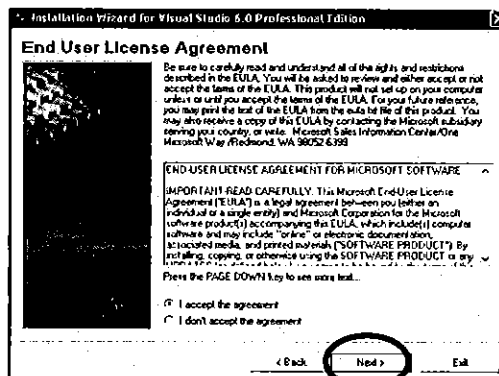
1. Unzip ไฟล์ VisuaalStudio6D1.rar จะได้ VisualStudio6D1.iso จากนั้นแตกไฟล์ VisualStudio.iso อีกครั้ง เลือกไฟล์ SETUP.EXE



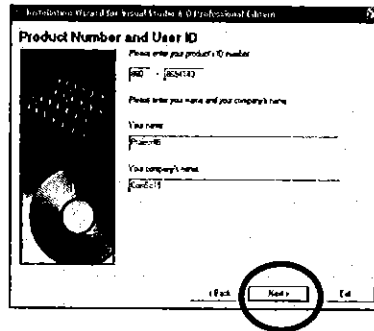
2. คลิก Next>



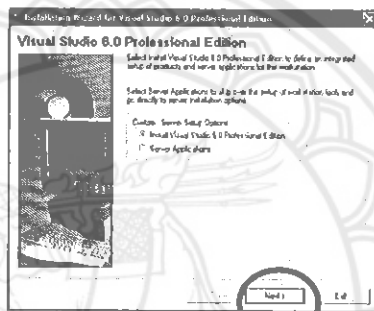
3. เลือกที่ I accept the agreement เพื่อยอมรับเงื่อนไข และข้อตกลงต่างๆ จากนั้นเลือกที่ NEXT>



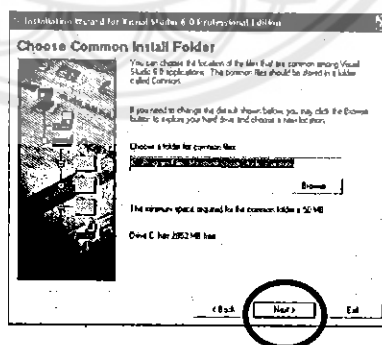
4. ใส่หมายเลขของ Product's ID, ชื่อ, และหน่วยงานจากนั้นกด NEXT>



5. เลือกที่ Install Visual Studio 6.0 Professional Edition จากนั้นเลือกที่ NEXT>



6. เลือก Drive ที่ต้องการจัดเก็บโปรแกรมไว้ โดยเลือกที่ Browse จากนั้นกดเลือก Drive ที่ต้องการ จากนั้นกด NEXT>

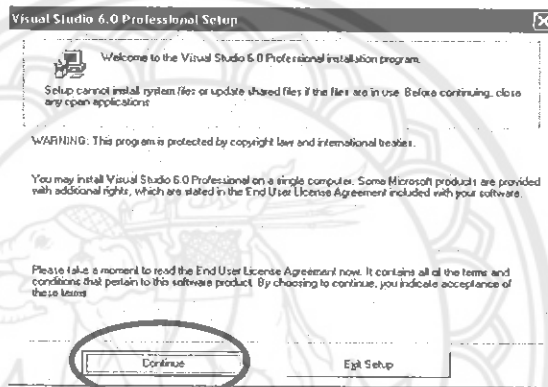


7. จากนั้นเริ่มการทำการติดตั้งโปรแกรม

Microsoft Visual Studio 6.0 Setup

Starting Visual Studio 6.0 Setup, please wait...

8. จากนั้นเลือกที่ Continue เพื่อทำการติดตั้ง

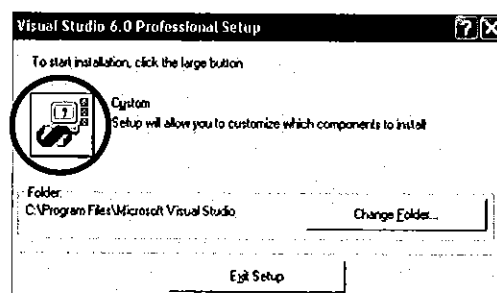


9. จากนั้นระบบจะทำการค้นหา Components ต่างๆที่จำเป็นในการติดตั้ง

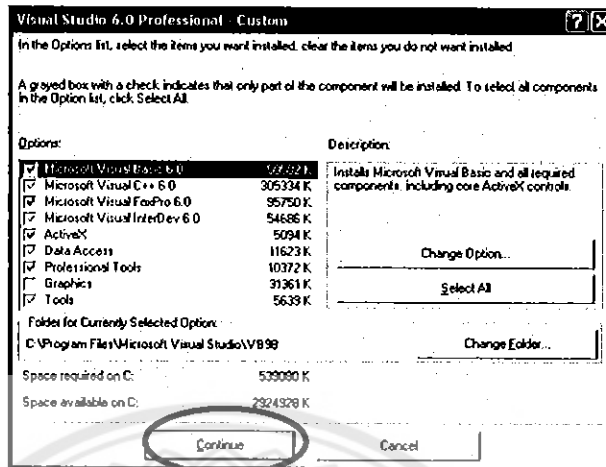


Setup is searching for installed components...

10. เลือกที่ Custom เพื่อติดตั้งโปรแกรม



11. ก่อนการติดตั้งให้เลือกที่ Microsoft Visual Basic 6.0 ก่อนเป็นการบอกว่าจะลงโปรแกรม Microsoft Visual Basic 6.0



12. จากนั้นเลือก OK เพื่อติดตั้งโปรแกรม

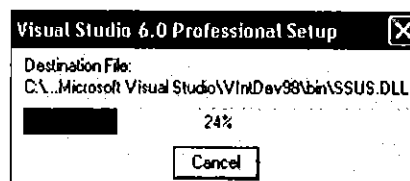


13. ทำการตรวจสอบพื้นที่ในหน่วยความจำว่าเพียงพอต่อการติดตั้งหรือไม่

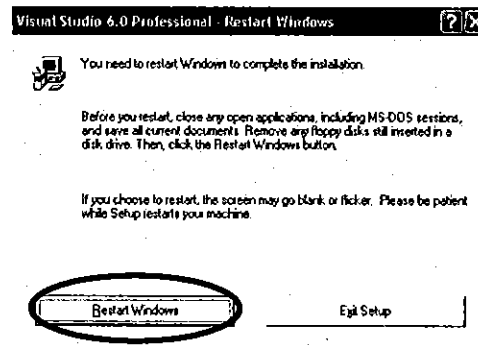


Setup is checking for necessary disk space...

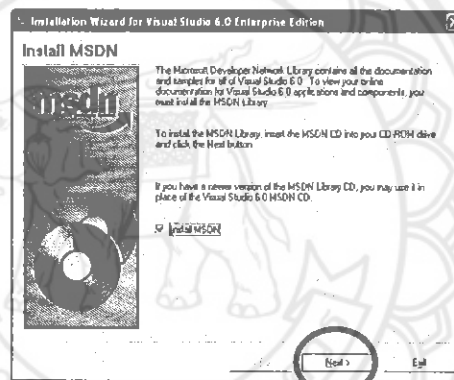
14. ทำการติดตั้งโปรแกรม



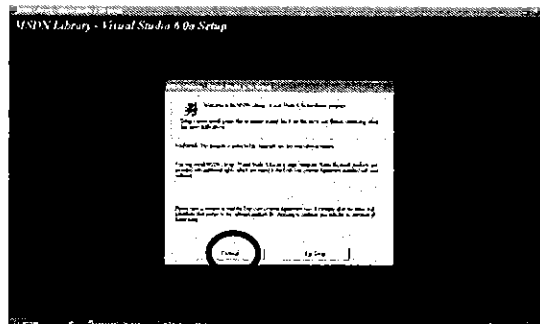
15. เมื่อทำการติดตั้งเสร็จแล้วให้เลือก Restart Windows เพราะระบบจะให้ทำการ Restart ก่อน โปรแกรมจึงจะเริ่มทำงานได้



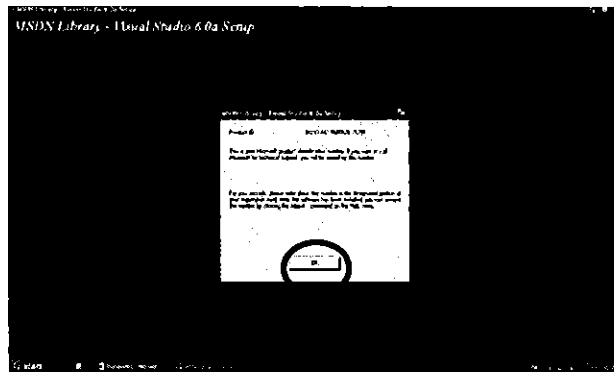
16. เมื่อ Restart เสร็จแล้ว จะปรากฏหน้าต่างเพื่อทำการติดตั้ง MSDN ต่อ หากต้องการติดตั้งให้คลิกที่ปุ่ม NEXT> แต่ถ้าหากไม่ต้องการให้กด EXIT



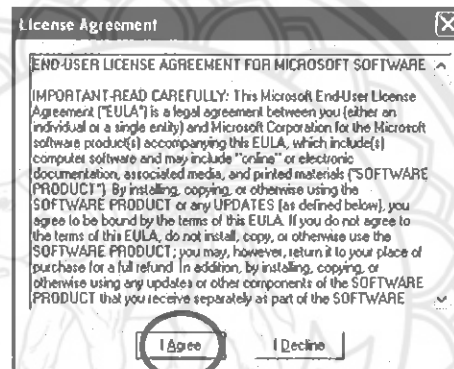
17. หากต้องการติดตั้ง MSDN จะมีหน้าต่างให้เลือกว่าจะติดตั้งที่ใด ให้เลือก Browse ไปที่โฟลเดอร์ VB_2 จากแผ่น CD หลังจากนั้นจะปรากฏหน้าจอด้านล่างให้กด Continue



18. หลังจากนั้นคลิก OK



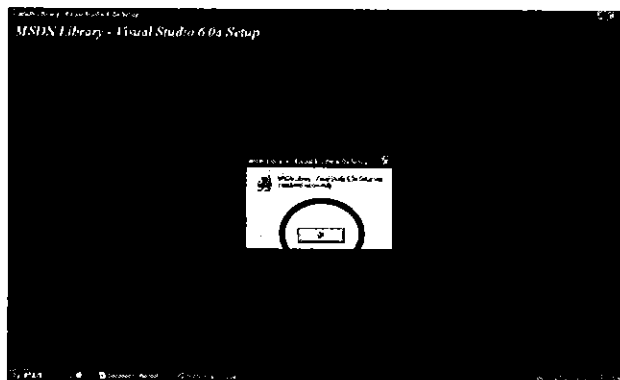
19. จะปรากฏหน้าต่าง License Agreement ให้คลิกปุ่ม I Agree



20. จากนั้นให้คลิกปุ่ม Typical



21. จนขึ้นหน้าจอด้านล่าง ให้คลิกปุ่ม OK เป็นการสิ้นสุดการติดตั้ง

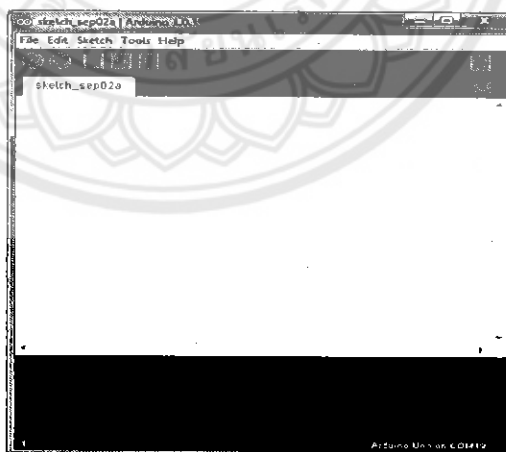


การลงโปรแกรม Arduino

1. เปิดโปรแกรม Arduino ที่ได้โหลดมาจาก <http://arduino.cc/en/Main/Software>

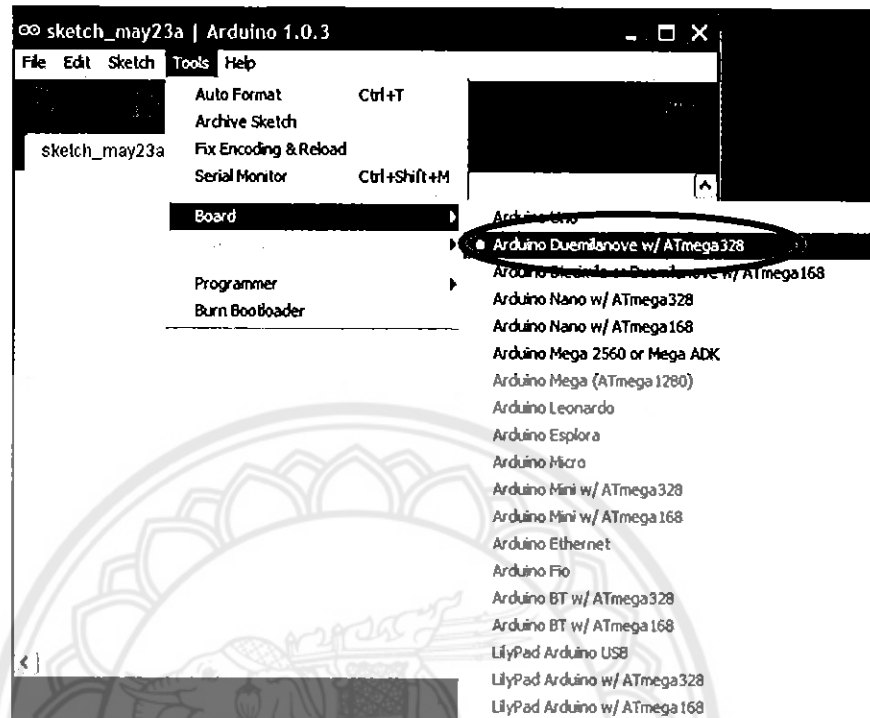


2. เมื่อเปิดโปรแกรมแล้วจะพบกับหน้าต่างของ IDE ดังรูป



3. ไปที่ Tools->Board แล้วเลือกให้ตรงกับบอร์ดที่ใช้งานสำหรับ Arduino Duemilanove w/aTmega328 เลือก Arduino_Duemilanove w/aTmega328 เลือกกำหนดบอร์ดใช้งานและกำหนดการติดต่อกับบอร์ด ก่อนที่จะทำการอัปเดตโปรแกรมเข้ากับบอร์ดทดลองเราต้องทำการเลือก

บอร์ดให้ตรงกับการใช้งาน (รุ่นหรือเบอร์ IC ให้ตรง) สามารถกำหนดได้ดังนี้ไปที่เมนู Tool > Board จากนั้นเลือกบอร์ดให้ตรงกับรุ่นที่ใช้งาน (ไม่จำเป็นต้องเหมือนในตัวอย่าง)



การสร้างงานใหม่

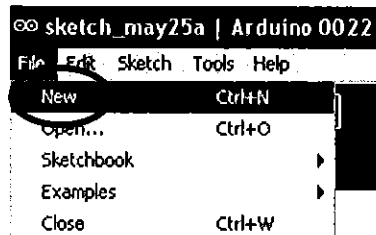
สิ่งที่จะต้องทำเป็นอันดับแรกในการเขียน โปรแกรมคือการหาที่เก็บ Code และกำหนดชื่อ (ชื่อ งานนั้นๆ) เมื่อเราเข้าโปรแกรม Arduino ในตัวโปรแกรมจะกำหนดชื่อมาให้ไว้แล้ว ในภาพจะเป็น ชื่อ sketch_may25a ถ้าต้องการชื่อดังกล่าวก็สามารใช้ได้ เลย หากไม่ต้องการหรือเปลี่ยนชื่ออื่น สามารถทำได้โดยการสร้างไฟล์งานใหม่ ทำได้ดังนี้



ตัวโปรแกรมจะกำหนดชื่อไว้เสมอ

ไปที่เมนู File > New จะได้หน้าต่างใหม่ขึ้นมา

การสร้างงานใหม่

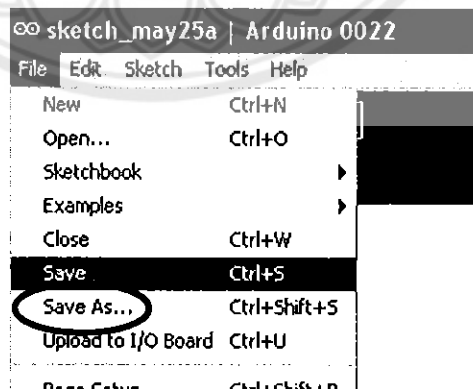


หน้าต่างงานใหม่



เมื่อได้หน้าต่างใหม่แล้วให้ทำการ Save ไปที่เมนู File > Save แล้วตั้งชื่อใหม่แล้วเก็บใน Folder ที่ต้องการ (ในตอนนี้จะตั้งชื่อ test 1)

การ Save งานใหม่

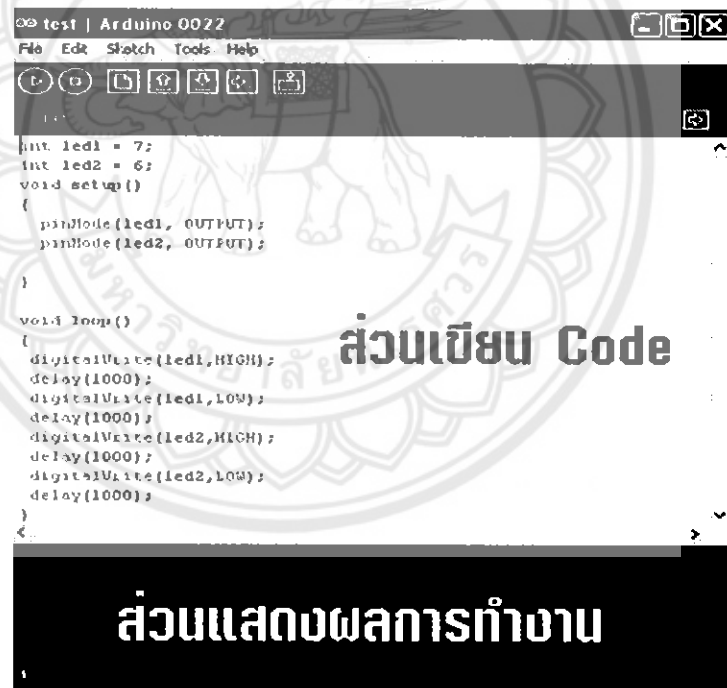


ชื่อไฟล์งานใหม่ คือ test 1



เริ่มเขียนโปรแกรม

เมื่อทำการกำหนดชื่อ และ ที่เก็บของ Code ได้แล้วเราก็มาทำการเขียนโปรแกรมโดยเขียนในส่วนของ Text Edit ได้เลย ในส่วนนี้จะมียู่ 2 ส่วนคือ ส่วนการเขียนโปรแกรมและส่วนแสดงผลการทำงาน

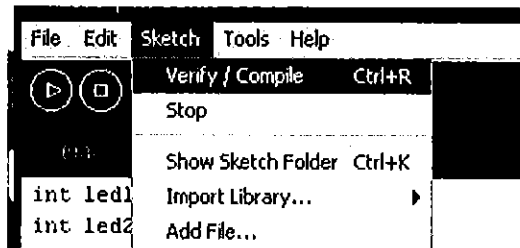


แสดงส่วนเขียน Code และ ส่วนแสดงผลการทำงาน

Verify/Compile

เมื่อทำการเขียนโปรแกรมตามที่ต้องการแล้วเราจะมาทำการ คอมไพล์เลอร์เพื่อดูว่า Code ที่เขียนไว้เขียนผิดหลัก หรือไม่ โดยในส่วนนี้สามารถเรียกใช้งาน โดยไปที่เมนู Sketch > Verify

Compile (Ctrl+R) แล้วดูผลการทำงาน



การเรียกใช้งานในส่วนการ Verify/Compile

การดูผลการทำงานสามารถดูได้ในส่วนท้ายดังนี้

เขียน Code ถูกหลักไวยากรณ์ ผ่าน

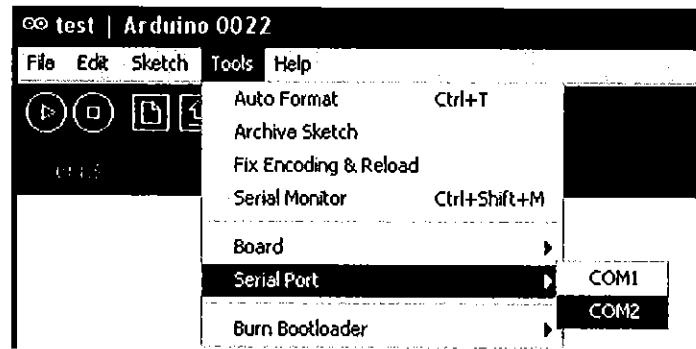
```
digitalWrite(led2,LOW);
delay(1000);
}
<
>
main.cpp
Binary sketch size: 1082 bytes (of a 30720 byte maximum)
1
```

เขียน Code ไม่ถูกหลักไวยากรณ์ ไม่ผ่าน

```
>
<
'led2' was not declared in this scope
6
```

การกำหนดบอร์ดที่ใช้ในการทดลองให้ตรง

เมื่อกำหนดบอร์ดแล้วให้กำหนด Port ที่ใช้ในการติดต่อสื่อสารกำหนดได้ดังนี้ ไปที่เมนู Tool > serial จากนั้นให้เลือก Port ที่ใช้ในการติดต่อ (บอร์ด Arduino จะทั้งแบบ port อนุกรม และแบบใช้ USB หรือใช้ตัวแปลง USB)

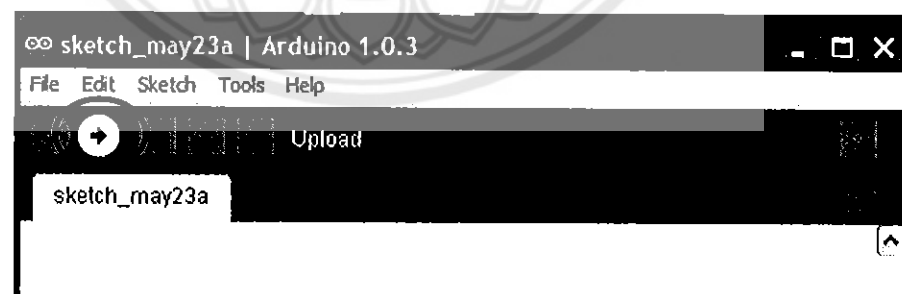


การกำหนดการเชื่อมต่อการอัปโหลดโปรแกรมลงในบอร์ดทดลอง เมื่อเขียน Code กำหนดส่วนต่างแล้วจะทำการอัปโหลด Code ลงในบอร์ดทดลอง จะใช้เมนู Upload ดังรูป



การเบิร์นโปรแกรมลงในบอร์ดทดลอง

เมื่อเขียน Code กำหนดส่วนต่างๆแล้วจะทำการอัปโหลด Code ลงในบอร์ดทดลอง จะใช้เมนู Upload ดังรูป



การ Upload

เมื่อทำการ Upload แล้วก็สามารถดูผลการทำงานของวงจรที่ได้ออกแบบไว้ได้แล้ว



ภาคผนวก ค

โปรแกรมระบบการเปิด - ปิดตู้ nirภัยสแกนลายนิ้วมือ

Code Visual Basic ในส่วนของ Forms

Frmemployee (การบันทึกข้อมูลส่วนบุคคล)

Option Explicit

Public State As FormState 'Variable used to determine on how the form used

Public PK As Long 'Variable used to get what record is going to edit

Dim HaveAction As Boolean 'Variable used to detect if the user perform some action

Dim RS As New Recordset

Private Sub DisplayForEditing()

On Error GoTo err

With RS

txtEntry(0).Text = .Fields("EmployeeID") (ส่วนของช่องกรอกข้อมูลของ EmployeeID)

txtEntry(1).Text = .Fields("LastName") (ส่วนของช่องกรอกข้อมูลของ LastName)

txtEntry(2).Text = .Fields("FirstName") (ส่วนของช่องกรอกข้อมูลของ FirstName)

txtEntry(3).Text = .Fields("MiddleName") (ส่วนของช่องกรอกข้อมูลของ MiddleName)

dcDepartment.BoundText = .Fields!["DepartmentID"]

txtEntry(4).Text = .Fields("Address") (ส่วนของช่องกรอกข้อมูลของ Address)

txtEntry(5).Text = .Fields("TelNo") (ส่วนของช่องกรอกข้อมูลของ TelNo)

txtEntry(6).Text = .Fields("MobileNo") (ส่วนของช่องกรอกข้อมูลของ MobileNo)

txtEntry(7).Text = .Fields("Picture") (ส่วนของช่องกรอกข้อมูลของ Picture)

End With

Exit Sub

err:

If err.Number = 94 Then Resume Next

prompt_err err, Name, "DisplayForEditing"

Screen.MousePointer = vbDefault

End Sub

Private Sub cmdCancel_Click()

Unload Me

End Sub

Private Sub ResetFields()

clearText Me

txtEntry(0).SetFocus

End Sub

Private Sub cmdSave_Click()

On Error GoTo err

If is_empty(txtEntry(0), True) = True Then Exit Sub


```

If is_empty(txtEntry(1), True) = True Then Exit Sub
If is_empty(txtEntry(2), True) = True Then Exit Sub
CN.BeginTrans

If State = adStateAddMode Or State = adStatePopupMode Then
    RS.AddNew(Codeส่วนของเพิ่มข้อมูลใหม่)
    RS.Fields("EmployeeID") = PK (Codeส่วนของเพิ่มข้อมูล EmployeeID)
    RS.Fields("AddedByFK") = CurrUser.USER_PK
Else
    RS.Fields("DateModified") = Now (Codeส่วนของการดัดแปลงข้อมูลใหม่)
    RS.Fields("LastUserFK") = CurrUser.USER_PK
End If

With RS
    .Fields("LastName") = txtEntry(1).Text
    .Fields("FirstName") = txtEntry(2).Text
    .Fields("MiddleName") = txtEntry(3).Text
    .Fields("DepartmentID") = dcDepartment.BoundText
    .Fields("Address") = txtEntry(4).Text
    .Fields("TelNo") = txtEntry(5).Text
    .Fields("MobileNo") = txtEntry(6).Text
    .Fields("Picture") = txtEntry(7).Text
    .Update
End With

CN.CommitTrans

HaveAction = True

If State = adStateAddMode Then
    MsgBox "New record has been successfully saved.", vbInformation (ข้อความเตือนเมื่อบันทึกข้อมูลผู้ใช้ใหม่)
    If MsgBox("Do you want to add another new record?", vbQuestion + vbYesNo) = vbYes Then
        (ข้อความเตือนเมื่อต้องการเพิ่มข้อมูลผู้ใช้งาน)
        ResetFields
    Else
        Unload Me
    End If
Else
    MsgBox "Changes in record has been successfully saved.", vbInformation (ข้อความเตือนเมื่อบันทึกข้อมูลผู้ใช้
    ถูกต้อง)

```

```

        Unload Me
    End If
    Exit Sub
err:
    CN.RollbackTrans
    prompt_err err, Name, "cmdSave_Click"
    Screen.MousePointer = vbDefault
End Sub

Private Sub cmdUsrHistory_Click()
    On Error Resume Next
    Dim tDate1 As String
    Dim tDate2 As String
    Dim tUser1 As String
    Dim tUser2 As String
    tDate1 = Format$(RS.Fields("DateAdded"), "MMM-dd-yyyy III:MM AMPM")
    tDate2 = Format$(RS.Fields("DateModified"), "MMM-dd-yyyy HH:MM AMPM")
    tUser1 = GetValueAt("SELECT PK,CompleteName FROM tbl_SM_Users WHERE PK = " & RS.Fields("AddedByFK"),
"CompleteName")
    tUser2 = GetValueAt("SELECT PK,CompleteName FROM tbl_SM_Users WHERE PK = " & RS.Fields("LastUserFK"),
"CompleteName")
    MsgBox "Date Added: " & tDate1 & vbCrLf & _
        "Added By: " & tUser1 & vbCrLf & _
        "" & vbCrLf & _
        "Last Modified: " & tDate2 & vbCrLf & _
        "Modified By: " & tUser2, vbInformation, "Modification History"
    tDate1 = vbNullString
    tDate2 = vbNullString
    tUser1 = vbNullString
    tUser2 = vbNullString
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    ' If KeyAscii = 13 And blnRemarks = False Then
    '     SendKeys "{TAB}"
    ' End If
End Sub

Private Sub Form_Load()

```

```

RS.CursorLocation = adUseClient

RS.Open "SELECT * FROM qry_Employees WHERE EmployeeID = " & PK, CN, adOpenStatic, adLockOptimistic
bind_dc "SELECT * FROM Department", "Department", dcDepartment, "DepartmentID", True

'Check the form state

If State = adStateAddMode Or State = adStatePopupMode Then

    Caption = "Create New Entry"

    cmdUsrHistory.Enabled = False

    GeneratePK

Else

    Caption = "Edit Entry"

    DisplayForEditing

End If

End Sub

'Procedure used to generate PK
Private Sub GeneratePK()

    PK = getIndex("Employees")

End Sub

Private Sub Form_Unload(Cancel As Integer)

    If HaveAction = True Then

        frmEmployeesList.RefreshRecords

    End If

    Set frmEmployees = Nothing

End Sub

Private Sub lblTitle_Click()

End Sub

Private Sub txtEntry_GotFocus(Index As Integer)

    HLText txtEntry(Index)

End Sub

Private Sub txtEntry_KeyPress(Index As Integer, KeyAscii As Integer)

    If Index = 0 Then KeyAscii = isNumber(KeyAscii)

End Sub

```

Frmemployeeslist (การบันทึกข้อมูลส่วนบุคคล)

```

Dim CURR_COL As Integer

Dim rsEmployees As New Recordset (Codeส่วนของการกรอกข้อมูลใหม่)

Dim RecordPage As New clsPaging (Codeส่วนของการเรียกใช้งานClass Moduleของ
clsPaging)

```

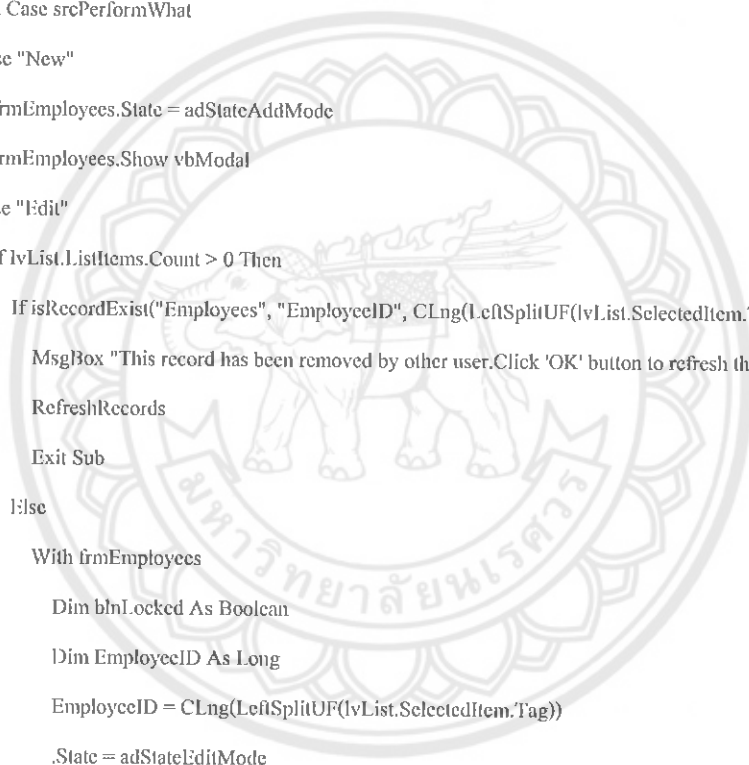
```

Dim SQLParser As New clsSQLSelectParser
clsSQL.SelectParser)

'Procedure used to filter records
Public Sub FilterRecord(ByVal srcCondition As String)
    SQLParser.RestoreStatement
    SQLParser.wCondition = srcCondition
    ReloadRecords SQLParser.SQLStatement
End Sub

Public Sub CommandPass(ByVal srcPerformWhat As String)
    On Error GoTo err
    Select Case srcPerformWhat
        Case "New"
            frmEmployees.State = adStateAddMode
            frmEmployees.Show vbModal
        Case "Edit"
            If IvList.ListItems.Count > 0 Then
                If isRecordExist("Employees", "EmployeeID", CLng(LeftSplitUF(IvList.SelectedItem.Tag))) = False Then
                    MsgBox "This record has been removed by other user.Click 'OK' button to refresh the records.", vbExclamation
                    RefreshRecords
                    Exit Sub
                Else
                    With frmEmployees
                        Dim bnlLocked As Boolean
                        Dim EmployeeID As Long
                        EmployeeID = CLng(LeftSplitUF(IvList.SelectedItem.Tag))
                        .State = adStateEditMode
                        .PK = EmployeeID
                        .Show vbModal
                    End With
                End If
            End If
        Case "Search"
            With frmSearch
                Set .srcForm = Me
                Set .srcColumnLeaders = IvList.ColumnLeaders
                .Show vbModal
            End With
    End Select

```



```

Case "Delete"
    If lvList.ListItems.Count > 0 Then
        If isRecordExist("Employees", "EmployeeID", CLng(LeftSplitUF(lvList.SelectedItem.Tag))) = False Then
            MsgBox "This record has been removed by other user.Click 'OK' button to refresh the records.", vbExclamation
            RefreshRecords
            Exit Sub
        Else
            Dim ANS As Integer

            ANS = MsgBox("Are you sure you want to delete the selected record?" & vbCrLf & vbCrLf & "WARNING: You
cannot undo this operation.", vbCritical + vbYesNo, "Confirm Record Delete")

            Me.MousePointer = vbHourglass
            If ANS = vbYes Then
                DelRecwSQL "Employees", "EmployeeID", "", True, CLng(LeftSplitUF(lvList.SelectedItem.Tag))
                RefreshRecords
                MsgBox "Record has been successfully deleted.", vbInformation, "Confirm"
            End If
            ANS = 0
            Me.MousePointer = vbDefault
        End If
    Else
        MsgBox "No record to delete.", vbExclamation
    End If
Case "Refresh"
    RefreshRecords
Case "Print"
    ' With mdiMain.CR
    '     .Reset: mdiMain.InitCrys
    '     .ReportFileName = App.Path & "\Reports\rpt_Customers_Profile.rpt"
    '     .WindowTitle = "Customer Profile"
    '     .ParameterFields(0) = "prmBusName;" & CurrBiz.BUSINESS_NAME & ";True"
    '     .PageZoom 100
    '     .Action = 1
    ' End With
Case "Close"
    Unload Me
End Select
Exit Sub

```

```

'Trap the error
err:
  If err.Number = -2147467259 Then
    MsgBox "You cannot delete this record because it was used by other records! If you want to delete this record" & vbCrLf &
    _
    "you will first have to delete or change the records that currently used this record as shown bellow." & vbCrLf & vbCrLf &
    & _
    err.Description, , "Delete Operation Failed!"
  Else
    MsgBox err.Number & " " & err.Description
  End If
  Me.MousePointer = vbDefault
End Sub
Public Sub RefreshRecords()
  SQLParser.RestoreStatement
  ReloadRecords SQLParser.SQLStatement
End Sub
'Procedure for reloading records
Public Sub ReloadRecords(ByVal srcSQL As String)
  '-In this case I used SQL because it is faster than Filter function of VB
  '-when handling millions of records.
  On Error GoTo err
  With rsEmployees
    If .State = adStateOpen Then .Close
    .Open srcSQL
  End With
  RecordPage.Refresh
  FillList 1
Exit Sub
err:
  If err.Number = -2147217913 Then
    srcSQL = Replace(srcSQL, "", "", , , vbTextCompare)
    Resume
  ElseIf err.Number = -2147217900 Then
    MsgBox "Invalid search operation.", vbExclamation
    SQLParser.RestoreStatement
    srcSQL = SQLParser.SQLStatement
  End If

```

```

        Resume
    Else
        prompt_err err, Name, "ReloadRecords"
    End If
End Sub

Private Sub btnClose_Click()

    Unload Me
End Sub

Private Sub btnFirst_Click()

    If RecordPage.PAGE_CURRENT <> 1 Then FillList 1
End Sub

Private Sub btnLast_Click()

    If RecordPage.PAGE_CURRENT <> RecordPage.PAGE_TOTAL Then FillList RecordPage.PAGE_TOTAL
End Sub

Private Sub btnNext_Click()

    If RecordPage.PAGE_CURRENT <> RecordPage.PAGE_TOTAL Then FillList RecordPage.PAGE_NEXT
End Sub

Private Sub btnPrev_Click()

    If RecordPage.PAGE_CURRENT <> 1 Then FillList RecordPage.PAGE_PREVIOUS
End Sub

Private Sub btnRecOp_Click()

    frmCustomerRecOp.Show vbModal
End Sub

Private Sub Form_Activate()

    mdiMain.ShowTBButton "ttttt"
End Sub

Private Sub Form_Deactivate()

    mdiMain.HideTBButton "", True
End Sub

Private Sub Form_Load()

    'Set the graphics for the controls

    With mdiMain

        'For listview

        Set lvList.SmallIcons = .i16x16

        Set lvList.Icons = .i16x16

        btnFirst.Picture = .i16x16.ListImages(3).Picture

        btnPrev.Picture = .i16x16.ListImages(4).Picture
    End With
End Sub

```

```

        btnNext.Picture = .i16x16.ListImages(5).Picture
        btnLast.Picture = .i16x16.ListImages(6).Picture
        btnFirst.DisabledPicture = .i16x16g.ListImages(3).Picture
        btnPrev.DisabledPicture = .i16x16g.ListImages(4).Picture
        btnNext.DisabledPicture = .i16x16g.ListImages(5).Picture
        btnLast.DisabledPicture = .i16x16g.ListImages(6).Picture
    End With

    With SQLParser
        .Fields = "EmployeeID, LastName, FirstName, Department, Address, TelNo, MobileNo, EmployeeID"
        .Tables = "qry_Employees"
        .SortOrder = "EmployeeID ASC"
        .SaveStatement
    End With

    If rsEmployees.State = 1 Then rsEmployees.Close
    rsEmployees.CursorLocation = adUseClient
    rsEmployees.Open SQL.Parser.SQLStatement, CN, adOpenStatic, adLockReadOnly
    With RecordPage
        .Start rsEmployees, 75
        FillList 1
    End With

End Sub

Private Sub FillList(ByVal whichPage As Long)
    RecordPage.CurrentPosition = whichPage
    Screen.MousePointer = vbHourglass
    Me.Enabled = False

    Call pageFillListView(lvList, rsEmployees, RecordPage.PageStart, RecordPage.PageEnd, 15, 2, False, True, , , ,
"EmployeeID")

    Me.Enabled = True
    Screen.MousePointer = vbDefault
    SetNavigation
    'Display the page information
    lblPageInfo.Caption = "Record " & RecordPage.PageInfo
    'Display the selected record
End Sub

```

FrmFrontScreen (หน้าจอสมแกนลายนิ้วมือ)

Private WithEvents op As FPGetTemplate

(ประกาศตัวแปล)


```

Dim tp As FPTemplate
Dim RS As New Recordset
Dim vp As FPVerify
Dim nStartTimer As Single
Dim nEndTimer As Single
Private Sub VerifyFP()

If op Is Nothing Then
Set op = New FPGetTemplate
End If
op.Run
End Sub
Private Sub cmbHistory_Click()
Form1.Show vbModal
End Sub
Private Sub Form_Load()
CenterControl Me.Frame1
RS.Open "SELECT * FROM EmployeeFP", CN, adOpenKeyset, adLockOptimistic
Call VerifyFP
MSComm1.Settings = "9600,N,8,1"
MSComm1.CommPort = 3
MSComm1.InputLen = 1
MSComm1.PortOpen = True
MSComm1.RThreshold = 1
End Sub
Private Sub Form_Resize()
picLine1.Width = Me.ScaleWidth
picLine2.Width = Me.ScaleWidth
picLine2.Top = Me.ScaleHeight - picLine2.Height
' gradLine2.width = Me.ScaleWidth
' gradLine3.width = Me.ScaleWidth
picLine1.Left = Me.Left
Option Explicit
Public State As FormState 'Variable used to determine on how the form used

```

(tp คือ fingerprinttemplate แสดงผลProfile นิ้วมือ)
(RS คือ คำสั่งบันทึกค่า)
(Vp คือ FingerPrintVerify ตัวแปลเปรียบเทียบลายนิ้วมือ)
(nStartTimer As Single ตัวแปลเวลาเก็บเป็นจำนวนเต็ม)
(nEndTimer As Single ตัวแปลเวลาเก็บเป็นจำนวนเต็ม)
(ชุดคำสั่ง เปรียบเทียบลายนิ้วมือ โดยถ้าลายนิ้วมือไม่ต้องให้เริ่ม
คำสั่ง FPGetTemplate คือหาลายนิ้วมือเปรียบเทียบอีกครั้ง)

(ชุดคำสั่ง โหลดหน้า CenterControl โดยเรียกตัวแปลต่างๆมา
เช่น EmployeeFP ชุดคำสั่งในกรอบ สั่งให้ VB ส่งสัญญาณออก
ที่ CommPort 3)

```

Public PK          As Long 'Variable used to get what record is going to edit
Dim HaveAction    As Boolean 'Variable used to detect if the user perform some action
Dim RS            As New Recordset

Private Sub DisplayForEditing()
    On Error GoTo err
    With RS
        txtEntry(0).Text = .Fields("EmployeeID")           (ส่วนของช่องกรอกข้อมูลของ EmployeeID)
        txtEntry(1).Text = .Fields("LastName")            (ส่วนของช่องกรอกข้อมูลของ LastName)
        txtEntry(2).Text = .Fields("FirstName")          (ส่วนของช่องกรอกข้อมูลของ FirstName)
        txtEntry(3).Text = .Fields("MiddleName")         (ส่วนของช่องกรอกข้อมูลของ MiddleName)
        delDepartment.BoundText = .Fields!["DepartmentID"]
        txtEntry(4).Text = .Fields("Address")             (ส่วนของช่องกรอกข้อมูลของ Address)
        txtEntry(5).Text = .Fields("TelNo")              (ส่วนของช่องกรอกข้อมูลของ TelNo)
        txtEntry(6).Text = .Fields("MobileNo")          (ส่วนของช่องกรอกข้อมูลของ MobileNo)
        txtEntry(7).Text = .Fields("Picture")           (ส่วนของช่องกรอกข้อมูลของ Picture)
    End With
    Exit Sub
err:
    If err.Number = 94 Then Resume Next
    prompt_err err, Name, "DisplayForEditing"
    Screen.MousePointer = vbDefault
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub ResetFields()
    clearText Me
    txtEntry(0).SetFocus
End Sub

Private Sub cmdSave_Click()
    On Error GoTo err
    If is_empty(txtEntry(0), True) = True Then Exit Sub

```

FrmLocate (ติดต่อกับ Microsoft Jet OLEDB4.0, ODBC)

```

Option Explicit

Private Sub cmdBrowse_Click()
    dlgOpen.ShowOpen           (ส่วนของการเรียกใช้งานของcommon dialog)
    txtPath.Text = dlgOpen.FileName

```

```

End Sub

Private Sub cmdCancel_Click()
    If CN.State = 1 Then CN.Close
    End
End Sub

Private Sub cmdOk_Click()
    If Trim(txtPath.Text) = "" Then
        'do nothing
    Else
        DBPath = txtPath.Text
        SetINI "Thumbsoft Settings", "Path", DBPath
        Unload Me
        'If InvalidDB Then frmLogin.show 1
    End If
End Sub

Private Sub Form_Load()
    With dlgOpen
        .Filter = "All Files (*.*)|*.| MS Access Files (*.mdb)|*.mdb|"
        (การเปิดไฟล์ของMicorsofe accessที่มีนามสกุลไฟล์.mdb)
    End With
End Sub

Private Sub optJet4_Click()
End Sub

Private Sub txtPath_Change()
End Sub

```

FrmRegister (หน้าจอบันทึกลายนิ้วมือ)

```

Dim WithEvents op As FPRRegisterTemplate          (เรียกใช้งานของ FPRRegisterTemplate)
Dim RS As New Recordset                          (RS คือ คำสั่งบันทึกค่าของลายนิ้วมือ)
Dim cursample As Integer
Dim register As FPTemplate                       (การสแกนลายนิ้วมือของfinger scan)
Dim PK As Long
Private Sub Save_and_Load_Verify_Form()
On Error GoTo err
    Dim bvariant As Variant
    Dim PK As Long
    blob_write = Null

```

```

bvariant = Null

If register Is Nothing Then
    lblEvents.Caption = ""
    MsgBox "Nothing Registered !!"
    Exit Sub
End If

register.Export bvariant

blob_write = bvariant

CN.BeginTrans

RS.Open "SELECT * FROM EmployeeFP", CN, adOpenKeyset, adLockOptimistic
RS.Filter = "[EmployeeID] = " & txtEmployeeID.Text
PK = txtEmployeeID.Text

If RS.RecordCount < 1 Then
    RS.AddNew
    GeneratePK
    RS("EmployeeID") = PK
End If

RS("Employee:P") = blob_write
RS.Update

Dim rsEmployees As New Recordset
rsEmployees.Open "SELECT * FROM Employees WHERE EmployeeID = " & txtEmployeeID.Text, CN, adOpenKeyset,
adLockOptimistic

If rsEmployees.RecordCount < 1 Then
    rsEmployees.AddNew
    rsEmployees("EmployeeID") = PK
    rsEmployees.Update
End If

CN.CommitTrans

rsEmployees.Close

RS.Close

With frmVerify
    .txtEmployeeID = txtEmployeeID.Text
    .Show vbModal
End With

Exit Sub

err:

```

```

    CN.RollbackTrans
    MsgBox err.Description, vbCritical
End Sub

Private Sub CmdClose_Click()
    Unload Me
End Sub

Private Sub Command1_Click()
    frmVerify.Show 1
End Sub

Private Sub Form_Load()
    cursample = 0
    Set op = New FPRegisterTemplate
    (การเรียกใช้งานFinger scanเพื่อสแกนลายนิ้วมือ4
    ครั้ง)
    For i = 0 To 3
        picSample(i).Picture = Nothing
        Label6(i).Visible = False
    Next i
End Sub

Private Sub op_Done(ByVal pTemplate As Object)
    lblEvents.Caption = ""
    Set register = Nothing
    Set register = pTemplate
    MsgBox "Registration Process is Done. The system is now going export template in Data.mdb database"
    Call Save_and_Load_Verify_Form
End Sub

Private Sub op_SampleQuality(ByVal Quality As DpSdkEngLib.AISampleQuality)
    Select Case Quality
    Case AISampleQuality.Sq_Good
        lblQuality.Caption = "OK"
        cursample = cursample + 1
        Label6(cursample - 1).Visible = False
        If cursample <> 4 Then
            Label6(cursample).Visible = True
        End If
    Case AISampleQuality.Sq_LowContrast
        lblQuality.Caption = ""
    Case AISampleQuality.Sq_NoCentralRegion

```

```

        lblQuality.Caption = ""
    Case AISampleQuality.Sq_None
        lblQuality.Caption = ""
    Case AISampleQuality.Sq_NotEnoughPtr
        lblQuality.Caption = ""
    Case AISampleQuality.Sq_TooDark
        lblQuality.Caption = ""
    Case AISampleQuality.Sq_TooLight
        lblQuality.Caption = ""
    Case AISampleQuality.Sq_TooNoisy
        lblQuality.Caption = ""
    End Select
    lblEvents.Caption = ""
End Sub

Private Sub op_SampleReady(ByVal pSample As Object)
    If cursample < 3 Then
        Label3.Caption = "Yes, put your finger again on the sensor to acquire sample #" & Str(cursample + 1)
        Label3.Refresh
    Else
        Label3.Caption = "Wait for the Verify form"
        Label3.Refresh
        Me.MousePointer = 2
    End If
    pSample.PictureOrientation = Or_Portrait
    pSample.PictureWidth = picSample(cursample).Width / Screen.TwipsPerPixelX
    pSample.PictureHeight = picSample(cursample).Height / Screen.TwipsPerPixelY
    picSample(cursample).Picture = pSample.Picture
    lblEvents.Caption = "Sample ready"
End Sub

Private Sub picSample_Click(Index As Integer)
    Dim i As Integer
    cursample = 0
    For i = 0 To 3
        picSample(i).Picture = Nothing
        Label6(i).Visible = False
    Next i
    Label6(cursample).Visible = True

```

```

Set op = Nothing

Set op = New FPRRegisterTemplate           (การเรียกใช้งานของFingerscan)

op.Run

IblQuality.Caption = ""

IblEvents.Caption = ""

End Sub

Private Sub start_cmd_Click()

    If txtEmployeeID.Text = "" Then       (ใส่เลขEmployee ID)
        MsgBox "Please enter Employee ID", vbExclamation      (ข้อความเตือนให้ใส่Employee ID)
        Exit Sub
    End If

    picSample_Click (0)

End Sub

'Procedure used to generate PK

Private Sub GeneratePK()

    PK = getIndex("Employees")

End Sub

FrmSearch (หน้าจอค้นหาผู้ใช้ตุลแกนลายนิ้วมือ)

Option Explicit

Public srcColumnHeaders As ColumnHeaders 'Source column headers

Public srcNoOfCol As Long

Public srcForm As Form 'Source form

Private Sub cmbFields_Change()

End Sub

Private Sub cmbOperation_Click(Index As Integer)

    If Index = 0 Then

        If cmbOperation(Index).ListIndex = 7 Then

            dtpDate(0).Visible = True

            dtpDate(1).Visible = True

            txtFilter(0).Visible = False

        Else

            txtFilter(0).Visible = True

            dtpDate(0).Visible = False

            dtpDate(1).Visible = False

        End If
    End If

```

```

Else
    If cmbOperation(Index).ListIndex = 7 Then
        dtpDate(2).Visible = True
        dtpDate(3).Visible = True
        txtFilter(1).Visible = False
    Else
        txtFilter(1).Visible = True
        dtpDate(2).Visible = False
        dtpDate(3).Visible = False
    End If
End If

End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOk_Click()
    'Verify
    If cmbOperation(0).ListIndex <> 7 Then If txtFilter(0).Text = "" Then txtFilter(0).SetFocus: Exit Sub
    On Error GoTo err
    Dim strFilter As String
    'Initialize the fields
    strFilter = Replace(cmbFields.Text, "/", "") 'ex. City/Town for tblCustomer
    strFilter = Replace(cmbFields.Text, " ", "")
    strFilter = "[" & strFilter & "]"
    'Initialize the operation used
    'First operation
    Select Case cmbOperation(0).ListIndex
        Case 0: strFilter = strFilter & " LIKE %" & txtFilter(0).Text & "%"
        Case 1: strFilter = strFilter & " = " & txtFilter(0).Text & ""
        Case 2: strFilter = strFilter & " <> " & txtFilter(0).Text & ""
        Case 3: strFilter = strFilter & " > " & txtFilter(0).Text & ""
        Case 4: strFilter = strFilter & " >= " & txtFilter(0).Text & ""
        Case 5: strFilter = strFilter & " < " & txtFilter(0).Text & ""
        Case 6: strFilter = strFilter & " <= " & txtFilter(0).Text & ""
        Case 7: strFilter = strFilter & " BETWEEN #" & dtpDate(0).Value & "# AND #" & dtpDate(1).Value & "#"
    End Select
    If cmbOperation(1).Text <> "" Then

```



```

'-Second operation
If Option1.Value = True Then
    strFilter = strFilter & " AND "
Else
    strFilter = strFilter & " OR "
End If
Select Case cmbOperation(1).ListIndex
    Case 0: strFilter = strFilter & " LIKE '%" & txtFilter(1).Text & "%"
    Case 1: strFilter = strFilter & " = " & txtFilter(1).Text & ""
    Case 2: strFilter = strFilter & " <> " & txtFilter(1).Text & ""
    Case 3: strFilter = strFilter & " > " & txtFilter(1).Text & ""
    Case 4: strFilter = strFilter & " >= " & txtFilter(1).Text & ""
    Case 5: strFilter = strFilter & " < " & txtFilter(1).Text & ""
    Case 6: strFilter = strFilter & " <= " & txtFilter(1).Text & ""
    Case 7: strFilter = strFilter & " BETWEEN #" & dtpDate(2).Value & "# AND #" & dtpDate(3).Value & "#"
End Select
End If
'InputBox "", , strFilter
'Pass the condition to filtered records
srcForm.FilterRecord strFilter
'Clear used variables
strFilter = vbNullString
Unload Me
Exit Sub
err:
If err.Number = -2147352571 Then
    MsgBox "Invalid search operation.", vbExclamation
    Unload Me
ElseIf err.Number = 3001 Then
    Resume Next
Else
    prompt_err err, "frmFilter", "cmdOk_Click"
End If
End Sub
Private Sub Form_Load()
    'Initialize values
    dtpDate(0).Value = Date

```

(ข้อความเมื่อต้องการหาผู้ใช้งานผู้สแกนลานี้ว่ามี)

```

dtpDate(1).Value = Date
dtpDate(2).Value = Date
dtpDate(3).Value = Date
'Set the images for the controls
With mdiMain
    Image1.Picture = .i16x16.ListImages(7).Picture
    Image2.Picture = .i16x16.ListImages(7).Picture
End With
Dim i As Integer
If srcNoOfCol = 0 Then srcNoOfCol = srcColumnHeaders.Count
For i = 1 To srcNoOfCol
    If srcColumnHeaders(i).Text <> "" Then cmbFields.AddItem srcColumnHeaders(i).Text
Next i
j = 0
cmbFields.ListIndex = 0
cmbOperation(0).ListIndex = 0
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Set frmSearch = Nothing
End Sub
Private Sub Frame1_DragDrop(Source As Control, X As Single, Y As Single)
End Sub
Private Sub Label1_Click(Index As Integer)
End Sub
Private Sub txtFilter_GotFocus(Index As Integer)
    HLText txtFilter(Index)
End Sub

```

FrmVerify (ตัวแปลเปรียบเทียบลายนิ้วมือ)

Private WithEvents op As FPGelTemplate	(fp คือ fingerprinttemplate แสดงผลProfile นิ้วมือ)
Dim tp As FPTemplate	(การเรียกใช้งานFinger Scan)
Dim RS As New Recordset	(RS คือ คำสั่งบนที่ค่าของข้อมูลผู้ใช้งาน)
Dim vp As FPVerify	(Vp คือ FingerPrintVerify ตัวแปลเปรียบเทียบลายนิ้วมือ)
Dim nStartTimer As Single	(ตัวแปลเวลาเก็บเป็นจำนวนเต็ม)
Dim nEndTimer As Single	(ตัวแปลเวลาเก็บเป็นจำนวนเต็ม)

```

Private Sub VerifyFP()
    (จุดคำสั่ง เปรียบเทียบลายนิ้วมือ โดยถ้าลายนิ้วมือไม่ต้องให้เริ่มคำสั่ง
    FPGetTemplate คือหาลายนิ้วมือเปรียบเทียบอีกครั้ง)

    If op Is Nothing Then
        Set op = New FPGetTemplate
    End If

    op.Run
End Sub

Private Sub CmdClose_Click()
    Unload Me
End Sub

Private Sub cmdOk_Click()
    Call VerifyFP
End Sub

Private Sub Form_Activate()
    If txtEmployeeID.Text = "" Then
        Label1.Caption = "Please Enter Employee ID"
        Exit Sub
    End If
End Sub

Private Sub Form_Load()
    RS.Open "SELECT * FROM EmployeeFP", CN, adOpenKeyset, adLockOptimistic
    Call VerifyFP
End Sub

Private Sub Form_Unload(Cancel As Integer)
    RS.Close
    Set op = Nothing
    Set vp = Nothing
End Sub

```

MdiMain (หน้าของส่วนสแกนลายนิ้วมือ)

```

Public Sub InitCrys()
Private Sub MDIForm_Load()
    DBPath = GetINI("Thumbsoft Settings", "Path") 'get path from file
    If Trim(DBPath) = "" Or IsNull(DBPath) Then
Jump here:
        frmLocate.Show 1 'browse database
    End If

```

```

If OpenDB = vbRetry Then GoTo JumpHere
mdiMain.HideTButton "", True
End Sub

Private Sub mnuEmployees_Click()
    LoadForm frmEmployeesList
End Sub

Private Sub mnuExit_Click()
    Unload Me
End Sub

Private Sub mnuFrontScreen_Click()
    frmFrontScreen.Show vbModal
End Sub

Private Sub mnuRegisterUser_Click()
    frmRegister.Show vbModal
End Sub

Private Sub mnuVerifyUser_Click()
    frmVerify.Show vbModal
End Sub

Public Sub HideTButton(ByVal srcPatern As String, Optional srcAllButton As Boolean)
    If srcAllButton = True Then srcPatern = "|||||"
End Sub

Private Sub tbMenu_ButtonClick(ByVal Button As MSCometLib.Button)
    ActiveForm.CommandPass Button.Key
End Sub

```

Code Visual Basic ในส่วนของ Modules ในส่วนของ Modules

ModADO (Module ตรวจสอบถูกหรือผิดของลายนิ้วมือ)

```

Option Explicit

Public Function OpenDB() As Boolean           (การเรียกใช้งานfunction)
    Dim isOpen As Boolean                    (Boolean คือการตรวจสอบลายนิ้วมือว่าถูกหรือผิด)
    Dim ANS As VbMsgBoxResult
    isOpen = False
    On Error GoTo err
    Do Until isOpen = True
        CN.CursorLocation = adUseClient

```

```

CN.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & DBPath & ";Persist Security Info=False;Jet
OLEDB:Database Password=jaypee"

isOpen = True

Loop

OpenDB = isOpen

Exit Function

err:

ANS = MsgBox("Error Number: " & err.Number & vbCrLf & "Description: " & err.Description, _
vbCritical + vbRetryCancel)

If ANS = vbCancel Then

OpenDB = vbCancel

ElseIf ANS = vbRetry Then

OpenDB = vbRetry

End If

End Function

Public Sub CloseDB()

'Close the connection

CN.Close

Set CN = Nothing

End Sub

'Function that return the current index for a certain table

Public Function getIndex(ByVal srcTable As String) As Long

On Error GoTo err

Dim RS As New Recordset

Dim RI As Long

RS.CursorLocation = adUseClient

RS.Open "SELECT * FROM [Key Generator] WHERE TableName = '" & srcTable & "'", CN, adOpenStatic,
adLockOptimistic

RI = RS.Fields("NextNo")

CN.BeginTrans

RS.Fields("NextNo") = RI + 1

RS.Update

CN.CommitTrans

getIndex = RI

srcTable = ""

RI = 0

Set RS = Nothing

```

```

Exit Function

err:
    "Error when incounter a null value

    If err.Number = 94 Then
        getIndex = 1
        Resume Next
    Else
        MsgBox err.Description
    End If

    CN.RollbackTrans

End Function

Function used to get the sum of fields

Public Function getSumOfFields(ByVal sTable As String, ByVal sField As String, ByRef sCN As ADODB.Connection, Optional
inclField As String, Optional sCondition As String) As Double

    On Error GoTo err

    Dim RS As New ADODB.Recordset
    RS.CursorLocation = adUseClient

    If sCondition <> "" Then sCondition = " GROUP BY " & inclField & " HAVING(" & sCondition & ")"
    If inclField <> "" Then inclField = "," & inclField

    RS.Open "SELECT Sum(" & sTable & "." & sField & ") AS fTotal" & inclField & " FROM " & sTable & sCondition, sCN,
adOpenStatic, adLockOptimistic

    If RS.RecordCount > 0 Then

        RS.MoveFirst

        Do While Not RS.EOF

            getSumOfFields = getSumOfFields + RS.Fields("fTotal")

            RS.MoveNext

        Loop

    Else

        getSumOfFields = 0

    End If

    Set RS = Nothing

Exit Function

err:

    "Error when incounter a null value

    If err.Number = 94 Then getSumOfFields = 0: Resume Next

End Function

'Procedure used to generate DSN

```

```

Public Sub GenerateDSN()
Open App.Path & "\rplCN.dsn" For Output As #1
    Print #1, "[ODBC]"
    Print #1, "DRIVER=Microsoft Access Driver (*.mdb)"
    Print #1, "UID=admin"
    Print #1, "UserCommitSync=Yes"
    Print #1, "Threads=3"
    Print #1, "SafeTransactions=0"
    Print #1, "PageTimeout=5"
    Print #1, "MaxScanRows=8"
    Print #1, "MaxBufferSize=2048"
    Print #1, "FIL=MS Access"
    Print #1, "DriverId=25"
    Print #1, "DefaultDir=" & App.Path & "\Data"
    Print #1, "DBQ=" & App.Path & "\Data\Data.mdb"
Close #1
End Sub
'Procedure used to remove DSN
Public Sub RemoveDSN()
On Error Resume Next
Kill App.Path & "\rplCN.dsn"
End Sub

```

ModAPI (Module ส่วนติดต่อ FingerPrint)

```

Option Explicit
'API Declarations and Constant
'For tracking mouse cursor position
Public Declare Function GetCursorPos Lib "user32" _
    (lpPoint As POINTAPI) As Long
Public Type POINTAPI
    X As Long
    Y As Long
End Type
'For memory status
Public Declare Sub GlobalMemoryStatus Lib "kernel32" _
    (lpBuffer As MEMORYSTATUS)
Private Type MEMORYSTATUS 'Type variable for memory info

```

dwLength As Long
 dwMemoryLoad As Long
 dwTotalPhys As Long
 dwAvailPhys As Long
 dwTotalPageFile As Long
 dwAvailPageFile As Long
 dwTotalVirtual As Long
 dwAvailVirtual As Long

End Type

'Use to get the window top,left,right and bottom position

Public Declare Function GetWindowRect Lib "user32" _

(ByVal hwnd As Long, _
 lpRect As RECT) As Long

Public Type RECT 'Type variable for window rect

Left As Long
 Top As Long
 Right As Long
 Bottom As Long

End Type

Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" _

(ByVal hwnd As Long, _
 ByVal wParam As Long, _
 ByVal lParam As Long, _
 lParam As Any) As Long

Public Const WM_CLOSE = &H10 'Message use to perform close

Public Const WM_ACTIVATE = &H16 'Message use to perform activate

'Use to set the parent

Public Declare Function SetParent Lib "user32" _

(ByVal hWndChild As Long, _
 ByVal hWndNewParent As Long) As Long

'Use for setting windows on top

Public Declare Function SetWindowPos Lib "user32" _

(ByVal hwnd As Long, _
 ByVal hWndInsertAfter As Long, _
 ByVal X As Long, _
 ByVal Y As Long, _
 ByVal cx As Long, _


```

        ByVal cy As Long, _
        ByVal wFlags As Long) As Long
Public Const HWND_TOPMOST = -1
Public Const HWND_NOTOPMOST = -2
Public Const SWP_NOSIZE = &H1
Public Const SWP_NOMOVE = &H2
Public Const SWP_NOACTIVATE = &H10
Public Const SWP_SHOWWINDOW = &H40
'API for opening a browser
Public Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" _
    (ByVal hwnd As Long, _
    ByVal lpOperation As String, _
    ByVal lpFile As String, _
    ByVal lpParameters As String, _
    ByVal lpDirectory As String, _
    ByVal nShowCmd As Long) As Long
Public MEM_STAT As MEMORYSTATUS
'API used to change the form border
Public Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As
Long) As Long
Public Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As
Long, ByVal dwNewLong As Long) As Long
'Related constant (see API used to change the form border)
Public Const WS_CAPTION = &HC0000
Public Const WS_THICKFRAME = &H40000
Public Const WS_DLGFRAME = &H40000
Public Const GWL_STYLE = (-16)
Public Const GWL_EXSTYLE = (-20)
Public Const SWP_FRAMECHANGED = &H20
Public Const SWP_NOZORDER = &H4
Public Const SWPFlags As Long = SWP_FRAMECHANGED Or SWP_NOMOVE Or SWP_NOZORDER Or SWP_NOSIZE
'Function used to change the form border (see API used to change the form border)
Public Function FormStyle(ByRef srcHWND As Long, ByVal Bit As Long, ByVal Value As Boolean) As Boolean
    Dim nStyle As Long
    nStyle = GetWindowLong(srcHWND, GWL_STYLE)
    If Value Then
        nStyle = nStyle Or Bit
    
```

```

Else
    nStyle = nStyle And Not Bit
End If

Call SetWindowLong(srcHWND, GWL_STYLE, nStyle)

SetWindowPos srcHWND, 0, 0, 0, 0, SWPFlags

FormStyle = (nStyle = GetWindowLong(srcHWND, GWL_STYLE))

End Function

Public Sub OpenURI(urlADD As String, sourceHWND As Long)

    Call ShellExecute(sourceHWND, vbNullString, urlADD, "", vbNullString, 1)

End Sub

Public Function OnTop(ByRef hwnd As Long, Optional BringOnTop As Boolean = True) As Long

    If BringOnTop = True Then

        SetWindowPos hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOACTIVATE Or SWP_SHOWWINDOW Or
        SWP_NOMOVE Or SWP_NOSIZE

    Else

        SetWindowPos hwnd, HWND_NOTOPMOST, 0, 0, 0, 0, SWP_NOACTIVATE Or SWP_SHOWWINDOW Or
        SWP_NOMOVE Or SWP_NOSIZE

    End If

End Function

```

ModDSN (Module ส่วนติดต่อ Microsoft Access)

```

Private Const ODBC_ADD_SYS_DSN = 4 ' Add data source
Private Const ODBC_CONFIG_DSN = 2 ' Configure (edit) data Source
Private Const ODBC_REMOVE_DSN = 3 ' Remove data source
Private Const vbAPINull As Long = vbNull ' NULL Pointer

Function Declare

Private Declare Function SQLConfigDataSource Lib "ODBC32.DLL" _
    (ByVal hwndParent As Long, ByVal fRequest As Long, _
    ByVal lpszDriver As String, ByVal lpszAttributes As String) _
    As Long

Public Sub createDSN()

    'Creating the DSN

    #If Win32 Then

        Dim intRet As Long

    #Else

        Dim intRet As Integer

    #End If

```

```

Dim strDriver As String
Dim strAttributes As String
strDriver = "Microsoft Access Driver (*.mdb)"
strAttributes = strAttributes & "DESCRIPTION=" & "VT Marketing DSN " & Chr$(0)
strAttributes = strAttributes & "DSN=" & "VT" & Chr$(0)
strAttributes = strAttributes & "PWD)=" & "jaypee" & Chr$(0)
strAttributes = strAttributes & "UID=" & "admin" & Chr$(0)
strAttributes = strAttributes & "DBQ=" & DBPath & Chr$(0)
intRet = SQL.ConfigDataSource(vbAPINull, ODBC_ADD_SYS_DSN, _
strDriver, strAttributes)
' DSN created
End Sub

```

ModEncrypt (Module การเข้ารหัสของลายนิ้วมือผู้ใช้งานผู้สแกนลายนิ้วมือ)

```

Option Explicit
Public Function DeCode(vText As String) As String
    On Error GoTo ErrHandler
    Dim CurSpe As Integer
    Dim varLen As Integer
    Dim varChr As String
    Dim varFin As String
    CurSpe = CurSpe + 1
    varLen = Len(vText)
    Do While CurSpe <= varLen
        DoEvents
        varChr = Mid(vText, CurSpe, 3)
        Select Case varChr
            'lower case
            Case "coc"
                varChr = "a"
            Case "wer"
                varChr = "b"
            Case "ibq"
                varChr = "c"
            Case "am7"
                varChr = "d"
            Case "pm1"

```

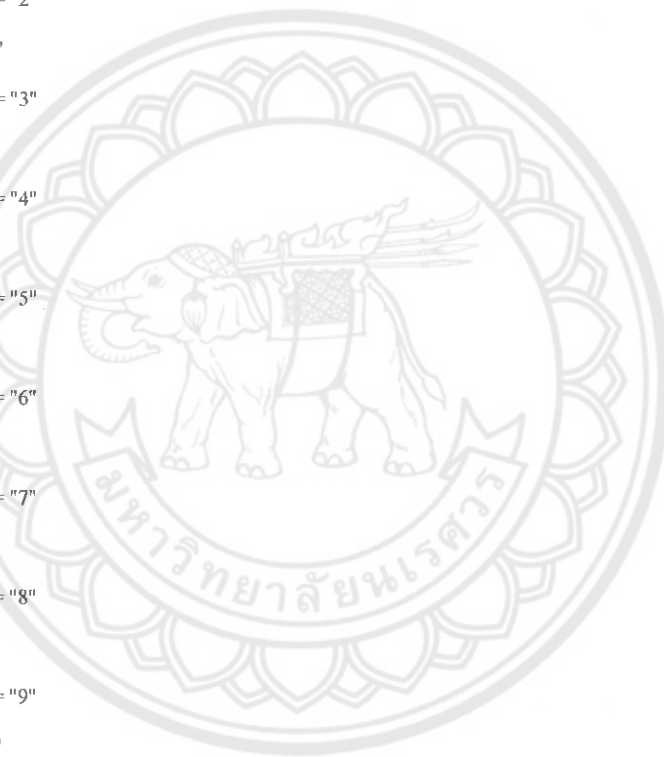
```
varChr = "e"  
Case "mop"  
varChr = "f"  
Case "9v4"  
varChr = "g"  
Case "qu6"  
varChr = "h"  
Case "zxc"  
varChr = "i"  
Case "4mp"  
varChr = "j"  
Case "f88"  
varChr = "k"  
Case "qe2"  
varChr = "l"  
Case "vbi"  
varChr = "m"  
Case "qwt"  
varChr = "n"  
Case "pl5"  
varChr = "o"  
Case "l3s"  
varChr = "p"  
Case "c%l"  
varChr = "q"  
Case "w$w"  
varChr = "r"  
Case "6a@"  
varChr = "s"  
Case "!2&"  
varChr = "t"  
Case "(=c"  
varChr = "u"  
Case "wvi"  
varChr = "v"  
Case "dp0"  
varChr = "w"
```



```

Case "wS-"
    varChr = "x"
Case "vn&"
    varChr = "y"
Case "c*4"
    varChr = "z"
'numbers
Case "aq@"
    varChr = "1"
Case "902"
    varChr = "2"
Case "2.&"
    varChr = "3"
Case "/w!"
    varChr = "4"
Case "|pq"
    varChr = "5"
Case "ml|"
    varChr = "6"
Case "t?"
    varChr = "7"
Case ">^s"
    varChr = "8"
Case "<s^"
    varChr = "9"
Case ",&c"
    varChr = "0"
'caps
Case "S)c"
    varChr = "A"
Case "-gt"
    varChr = "B"
Case "lp*"
    varChr = "C"
Case "1" & Chr(34) & "r"
    varChr = "D"
Case ">:"

```



```
    varChr = "E"  
Case "@+x"  
    varChr = "F"  
Case "v^a"  
    varChr = "G"  
Case "jeE"  
    varChr = "H"  
Case "aP0"  
    varChr = "I"  
Case "{=1"  
    varChr = "J"  
Case "cWv"  
    varChr = "K"  
Case "cDe"  
    varChr = "L"  
Case "*,!"  
    varChr = "M"  
Case "fW" & Chr(34)  
    varChr = "N"  
Case ".,?T"  
    varChr = "O"  
Case "%<8"  
    varChr = "P"  
Case "@:a"  
    varChr = "Q"  
Case "&c$"  
    varChr = "R"  
Case "WnY"  
    varChr = "S"  
Case "{Sh"  
    varChr = "T"  
Case "_%M"  
    varChr = "U"  
Case "}'$"  
    varChr = "V"  
Case "QIU"  
    varChr = "W"
```



```
Case "Im^"
    varChr = "X"
```

```
Case "|P"
    varChr = "Y"
```

```
Case ">#"
    varChr = "Z"
```

'Special characters

```
Case "\ & Chr(34) & "]"
    varChr = "!"
```

```
Case "cY,"
    varChr = "@"
```

```
Case "x%B"
    varChr = "#"
```

```
Case "a*v"
    varChr = "$"
```

```
Case "'&T"
    varChr = "%"
```

```
Case ";%R"
    varChr = "^"
```

```
Case "eG_"
    varChr = "&"
```

```
Case "Z/c"
    varChr = "*"
```

```
Case "rGV"
    varChr = "("
```

```
Case "]*!"
    varChr = ")"
```

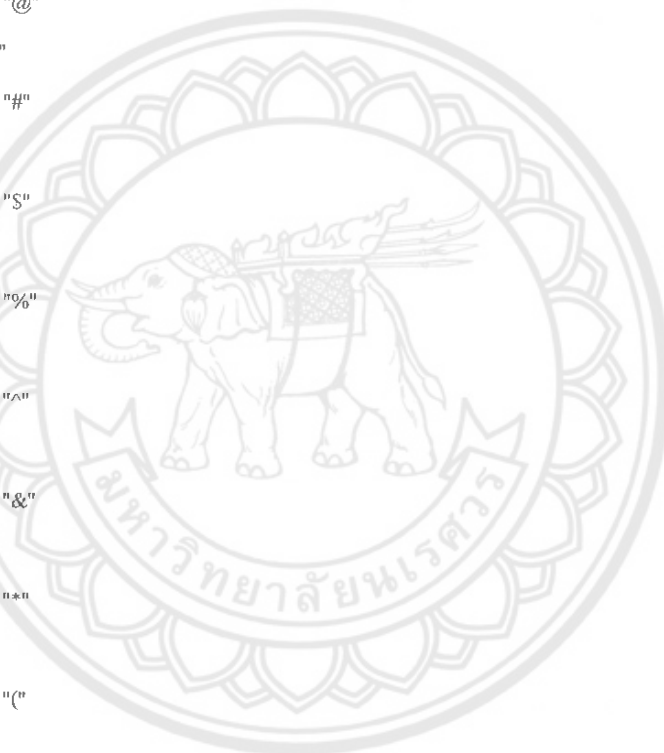
```
Case "@B*"
    varChr = "_"
```

```
Case "+He"
    varChr = "-"
```

```
Case "&|D"
    varChr = "="
```

```
Case "(:#"
    varChr = "+"
```

```
Case "SIW"
    varChr = "["
```



```

Case "QB"
    varChr = "]"
Case "{D>"
    varChr = "{"
Case "+c%"
    varChr = "}"
Case "(s:"
    varChr = ":"
Case "^a("
    varChr = ";"
Case "16."
    varChr = ""
Case "s.*"
    varChr = Chr(34)
Case "&?W"
    varChr = ";"
Case "GPQ"
    varChr = "."
Case "SK*"
    varChr = "<"
Case "R1.^"
    varChr = ">"
Case "40C"
    varChr = "/"
Case "?#9"
    varChr = "?"
Case "_?/"
    varChr = "\"
Case "(@"
    varChr = "|"
Case "#B"
    varChr = " "
End Select

varFin = varFin & varChr
CurSpc = CurSpc + 3
DoEvents
Loop

```



```

DeCode = varFin
Exit Function
ErrorHandler:
    Dim ErrNum, ErrDesc, ErrSource
    ErrNum = err.Number
    ErrDesc = err.Description
    ErrSource = err.Source
    MsgBox "Error# = " & ErrNum & vbCrLf & "Description = " & ErrDesc & vbCrLf & "Source = " & ErrSource, vbCritical +
vbOKOnly, "Program Error!"
    err.Clear
    Exit Function
End Function
Public Function Encode(vText As String)
    On Error GoTo ErrorHandler
    Dim CurSpc As Integer
    Dim varLen As Integer
    Dim varChr As String
    Dim varFin As String
    varLen = Len(vText)
    Do While CurSpc <= varLen
        DoEvents
        CurSpc = CurSpc + 1
        varChr = Mid(vText, CurSpc, 1)
        Select Case varChr
            'lower case
            Case "a"
                varChr = "coe"
            Case "b"
                varChr = "wer"
            Case "c"
                varChr = "ibq"
            Case "d"
                varChr = "am7"
            Case "e"
                varChr = "pm1"
            Case "f"
                varChr = "mop"

```

```
Case "g"  
    varChr = "9v4"  
Case "h"  
    varChr = "qu6"  
Case "i"  
    varChr = "zxc"  
Case "j"  
    varChr = "4mp"  
Case "k"  
    varChr = "f88"  
Case "l"  
    varChr = "qe2"  
Case "m"  
    varChr = "vbn"  
Case "n"  
    varChr = "qwt"  
Case "o"  
    varChr = "pl5"  
Case "p"  
    varChr = "l3s"  
Case "q"  
    varChr = "c%l"  
Case "r"  
    varChr = "w$w"  
Case "s"  
    varChr = "6a@"  
Case "t"  
    varChr = "!2&"  
Case "u"  
    varChr = "(=c"  
Case "v"  
    varChr = "wvT"  
Case "w"  
    varChr = "dp0"  
Case "x"  
    varChr = "w$-"  
Case "y"
```



```

varChr = "vn&"
Case "z"
varChr = "c*4"
'numbers
Case "1"
varChr = "aq@"
Case "2"
varChr = "902"
Case "3"
varChr = "2.&"
Case "4"
varChr = "/w!"
Case "5"
varChr = "|pq"
Case "6"
varChr = "ml|"
Case "7"
varChr = "t?"
Case "8"
varChr = ">^s"
Case "9"
varChr = "<s^"
Case "0"
varChr = ";&c"

```

Loop

Encode = varFin

Exit Function

ErrorHandler:

```
Dim ErrNum, ErrDesc, ErrSource
```

```
ErrNum = err.Number
```

```
ErrDesc = err.Description
```

```
ErrSource = err.Source
```

```
MsgBox "Error# = " & ErrNum & vbCrLf & "Description = " & ErrDesc & vbCrLf & "Source = " & ErrSource, vbCritical +
vbOKOnly, "Program Error!"
```

```
err.Clear
```

Exit Function

End Function

ModFunction (Module การทำงานของ FingerPrint)

Option Explicit

Public Function FormatRS(ByVal srcField As Field, Optional AllowNewLine As Boolean) As String

Dim strRet As String

With srcField

If AllowNewLine = True Then

strRet = srcField

Else

strRet = Replace(srcField, vbCrLf, " , , , vbTextCompare)

End If

If srcField.Type = adCurrency Or srcField.Type = adDouble Then

If srcField.Type = adCurrency Then

strRet = FormatS(srcField, "#,##0.00")

ElseIf srcField.Type = adDate Then

strRet = FormatS(srcField, "MMM-dd-yyyy")

Else

strRet = srcField

End If

End With

FormatRS = strRet

strRet = vbNullString

End Function

'Function that will format return a generated id

Public Function GenerateID(ByVal srcNo As String, ByVal src1stStr As String, ByVal src2ndStr As String) As String

If Len(src2ndStr) <= Len(srcNo) Then

GenerateID = src1stStr & srcNo

Else

GenerateID = src1stStr & Left\$(src2ndStr, Len(src2ndStr) - Len(srcNo)) & srcNo

End If

End Function

'Function used to check if the record exist or not.

Public Function isRecordExist(ByVal sTable As String, ByVal sField As String, ByVal sStr As String, Optional isString As

Boolean) As Boolean

Dim RS As New Recordset

RS.CursorLocation = adUseClient

If isString = False Then

```

RS.Open "Select * From " & sTable & " Where " & sField & " = " & sStr, CN, adOpenStatic, adLockOptimistic
Else
    RS.Open "Select * From " & sTable & " Where " & sField & " = " & sStr & "'", CN, adOpenStatic, adLockOptimistic
End If
If RS.RecordCount < 1 Then
    isRecordExist = False
Else
    isRecordExist = True
End If
Set RS = Nothing
End Function
'Function used to check if the Ascii is a number or not (return 0 if number)
Public Function isNumber(ByVal sKeyAscii) As Integer
    If Not ((sKeyAscii >= 48 And sKeyAscii <= 57) Or sKeyAscii = 8 Or sKeyAscii = 46) Then
        isNumber = 0
    Else
        isNumber = sKeyAscii
    End If
End Function
'Function used to check if the record exist in Flex grid
Public Function isRecExistInFlex(ByVal srcFlexGrd As MSFlexGrid, ByVal srcWhatCol As Integer, ByVal srcFindWhat As
String) As Boolean
    isRecExistInFlex = False
    Dim i As Long
    For i = 1 To srcFlexGrd.Rows - 1
        If srcFlexGrd.TextMatrix(i, srcWhatCol) = srcFindWhat Then isRecExistInFlex = True: Exit For
    Next i
    i = 0
End Function
'Function used to check if the record exist in Flex grid
Public Function getFlexPos(ByVal srcFlexGrd As MSFlexGrid, ByVal srcWhatCol As Integer, ByVal srcFindWhat As String)
As Integer
    Dim r As Long, ret As Integer
    ret = -1 'Means not found
    For r = 0 To srcFlexGrd.Rows - 1
        If srcFlexGrd.TextMatrix(r, srcWhatCol) = srcFindWhat Then ret = r: Exit For
    Next r

```

```

    getFlexPos = ret
    r = 0: ret = 0
End Function

'Function used to left split user fields
Public Function LeftSplitUF(ByVal srcUF As String) As String
    If srcUF = "*~*~*~*~*" Then LeftSplitUF = "": Exit Function

    Dim i As Integer
    Dim t As String

    For i = 1 To Len(srcUF)
        If Mid$(srcUF, i, 7) = "*~*~*~*~*" Then
            Exit For
        Else
            t = t & Mid$(srcUF, i, 1)
        End If
    Next i

    LeftSplitUF = t
    i = 0
    t = ""
End Function

'Function used to right split user fields
Public Function RightSplitUF(ByVal srcUF As String) As String
    If srcUF = "*~*~*~*~*" Then RightSplitUF = "": Exit Function

    Dim i As Integer
    Dim t As String

    For i = (InStr(1, srcUF, "*~*~*~*~*", vbTextCompare) + 7) To Len(srcUF)
        t = t & Mid$(srcUF, i, 1)
    Next i

    RightSplitUF = t
    i = 0
    t = ""
End Function

'Function that return true if the control is empty
Public Function is_empty(ByRef sText As Variant, Optional UseTagValue As Boolean) As Boolean
    On Error Resume Next

    If sText.Text = "" Then
        is_empty = True

        If UseTagValue = True Then

```

```

    MsgBox "The field '" & sText.Tag & "' is required.Please check it!", vbExclamation
Else
    MsgBox "The field is required.Please check it!", vbExclamation
End If
sText.SetFocus
Else
    is_empty = False
End If
End Function

```

'Function used to change the yes/no value

```

Public Function changeYNValue(ByVal srcStr As String) As String
    Select Case srcStr
        Case "Y": changeYNValue = "1"
        Case "N": changeYNValue = "0"
        Case "1": changeYNValue = "Y"
        Case "0": changeYNValue = "N"
    End Select
End Function

```

'Function used to change the true/false value

```

Public Function changeTFValue(ByVal srcStr As String) As String
    Select Case srcStr
        Case "True": changeTFValue = "1"
        Case "False": changeTFValue = "0"
        Case "1": changeTFValue = "True"
        Case "0": changeTFValue = "False"
    End Select
End Function

```

'Function that return true if the control is numeric

```

Public Function is_numeric(ByRef sText As String) As Boolean
    If IsNumeric(sText) = False Then
        is_numeric = False
        MsgBox "The field required a numeric input.Please check it!", vbExclamation
    Else
        is_numeric = True
    End If
End Function

```

'Function that return the value of a certain field

Public Function GetValueAt(ByVal srcSQL As String, ByVal whichField As String) As String

```
Dim RS As New Recordset
RS.CursorLocation = adUseClient
RS.Open srcSQL, CN, adOpenStatic, adLockReadOnly
If RS.RecordCount > 0 Then GetValueAt = RS.Fields(whichField)
Set RS = Nothing
```

End Function

'Convert string to number

'I create this instead of val() coz val return incorrect value

'ex. Try to see the output of val("3,800")

'It did not support characters like , and etc.

Public Function toNumber(ByVal srcCurrency As String, Optional RetZeroIfNegative As Boolean) As Double

```
If srcCurrency = "" Then
    toNumber = 0
Else
    Dim retValue As Double
    If InStr(1, srcCurrency, ",") > 0 Then
        retValue = Val(Replace(srcCurrency, ",", "", , , vbTextCompare))
    Else
        retValue = Val(srcCurrency)
    End If
    If RetZeroIfNegative = True Then
        If retValue < 1 Then retValue = 0
    End If
    toNumber = retValue
    retValue = 0
```

End If

End Function

'Function that return the count of the rows in the table

Public Function getRecordCount(ByVal srcTable As String, Optional srcCondition As String, Optional isFormatted As Boolean) As String

```
If srcCondition <> "" Then srcCondition = " " & srcCondition
Dim RS As New Recordset
RS.CursorLocation = adUseClient
RS.Open "SELECT COUNT(PK) as TCount FROM " & srcTable & srcCondition, CN, adOpenStatic, adLockReadOnly
If isFormatted = True Then
    getRecordCount = Format$(RS![TCount], "#,##0")
```



```

Else
    getRecordCount = RS![TCount]
End If

Set RS = Nothing
End Function

'Function that will return a currenct format
Public Function toMoney(ByVal srcCurr As String) As String
    toMoney = Format$(IIf(Trim(srcCurr) = "", 0, srcCurr), "#,##0.00")
End Function

'Function used to determine if the object has been set
Public Function isObjectSet(srcObject As Object) As Boolean
    On Error GoTo err
    'I use tag because almost all controls have this
    srcObject.Tag = srcObject.Tag
    isObjectSet = True
    Exit Function
err:
    isObjectSet = False
End Function

'Function used to get the end day number of a certain month
Public Function getEndDay(ByVal srcDate As Date) As Byte
    Dim h1 As String
    h1 = Format(srcDate, "mm")
    On Error GoTo err
    Select Case h1
        Case Is = "01": getEndDay = 31
        Case Is = "02": getEndDay = Day(h1 & "/29/" & Format(srcDate, "yy"))
        Case Is = "03": getEndDay = 31
        Case Is = "04": getEndDay = 30
        Case Is = "05": getEndDay = 31
        Case Is = "06": getEndDay = 30
        Case Is = "07": getEndDay = 31
        Case Is = "08": getEndDay = 31
        Case Is = "09": getEndDay = 30
        Case Is = "10": getEndDay = 31
        Case Is = "11": getEndDay = 30
        Case Is = "12": getEndDay = 31
    
```

```

End Select

hl = ""

Exit Function

err:

If err.Number = 13 Then getI:ndDay = 28: hl = "" 'Day if encounter not a left-year

End Function

Public Function getUnitID(ByVal sUnit As String) As Long

Dim RS As New ADODB.Recordset

Dim sql As String

sql = "SELECT UnitID From Unit WHERE ((Unit)=" & Replace(sUnit, "'", """) & "')"

RS.Open sql, CN, adOpenDynamic, adLockOptimistic

If Not RS.EOF Then

getUnitID = RS!UnitID

Else

getUnitID = 0

End If

RS.Close

Set RS = Nothing

End Function

Function GetINI(strMain As String, strSub As String) As String

Dim strBuffer As String

Dim lngLen As Long

Dim lngRet As Long

strBuffer = Space(100)

lngLen = Len(strBuffer)

lngRet = GetPrivateProfileString(strMain, strSub, vbNullString, strBuffer, lngLen, App.Path & "\config.txt")

GetINI = Left(strBuffer, lngRet)

End Function

```

ModProcedure (Module ขั้นตอนการทำงานส่วนต่างๆของ โปรแกรมตู้สแกนลายนิ้วมือ)

```

Option Explicit

Public Declare Function WritePrivateProfileString Lib "kernel32" Alias "WritePrivateProfileStringA" (ByVal lpApplicationName
As String, ByVal lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As String) As Long

Public Sub LoadForm(ByRef srcForm As Form)

```

On Error Resume Next

```
srcForm.Show
srcForm.WindowState = vbMaximized
srcForm.SetFocus
```

End Sub

Procedure used to custom move the recordset cursor

```
Public Sub customMove(ByRef sRS As Recordset, ByVal isNum As Boolean, ByVal findStr As String, ByVal sField As String)
```

```
    If sRS.RecordCount < 1 Then Exit Sub
```

```
    Dim old_pos As Long
```

```
    sRS.MoveFirst
```

```
    old_pos = sRS.AbsolutePosition
```

```
    If isNum = True Then
```

```
        sRS.Find sField & "=" & findStr
```

```
    Else
```

```
        sRS.Find sField & "=" & findStr & ""
```

```
    End If
```

```
    If sRS.EOF Then sRS.AbsolutePosition = old_pos
```

```
    old_pos = 0
```

End Sub

'This code is also available in .NET version with ADO.NET

'Procedure used to fill list view

```
Public Sub FillListView(ByRef sListView As ListView, ByRef sRecordSource As Recordset, ByVal sNumOfFields As Byte,
```

```
ByVal sNumIco As Byte, ByVal with_num As Boolean, ByVal show_first_rec As Boolean, Optional srcHiddenField As String)
```

```
    Dim X As Variant
```

```
    Dim i As Byte
```

On Error Resume Next

```
sListView.ListItems.Clear
```

```
If sRecordSource.RecordCount < 1 Then Exit Sub
```

```
sRecordSource.MoveFirst
```

```
Do While Not sRecordSource.EOF
```

```
    If with_num = True Then
```

```
        Set X = sListView.ListItems.Add(, , sRecordSource.AbsolutePosition, sNumIco, sNumIco)
```

```
    Else
```

```
        Set X = sListView.ListItems.Add(, "" & sRecordSource.Fields(0), sNumIco, sNumIco)
```

```
    End If
```

```
    If srcHiddenField <> "" Then X.Tag = sRecordSource.Fields(srcHiddenField)
```

```
    For i = 1 To sNumOfFields - 1
```

```

If show_first_rec = True Then
    If with_num = True Then
        If sRecordSource.Fields(CInt(i) - 1).Type = adDouble Then
            X.SubItems(i) = FormatRS(sRecordSource.Fields(CInt(i) - 1))
        Else
            X.SubItems(i) = "" & FormatRS(sRecordSource.Fields(CInt(i) - 1))
        End If
    Else
        If sRecordSource.Fields(CInt(i)).Type = adDouble Then
            X.SubItems(i) = FormatRS(sRecordSource.Fields(CInt(i)))
        Else
            X.SubItems(i) = "" & FormatRS(sRecordSource.Fields(CInt(i)))
        End If
    End If
Else
    X.SubItems(i) = "" & FormatRS(sRecordSource.Fields(CInt(i) + 1))
End If
Next i
sRecordSource.MoveNext
Loop
i = 0
Set X = Nothing
End Sub

'Procedure used to prompt unexpected errors
Public Sub prompt_err(ByVal sError As ErrObject, ByVal ModuleName As String, ByVal OccurIn As String)
    MsgBox "Error From: " & ModuleName & vbNewLine & _
        "Occur In: " & OccurIn & vbNewLine & _
        "Error Number: " & sError.Number & vbNewLine & _
        "Description: " & sError.Description, vbCritical, "Application Error"
    'Save the error log (The save error log will be display later on in the program)
    Open App.Path & "\Error.log" For Append As #1
        Print #1, Format(Date, "MMM-dd-yyyy") & "~~~~~" & Time & "~~~~~" & sError.Number & "~~~~~" &
sError.Description & "~~~~~" & ModuleName & "~~~~~" & OccurIn
    Close #1
End Sub

'Procedure used to delete record with SQL

```

```
Public Sub DelRecwSQL(ByVal sTable As String, ByVal sField As String, ByVal sString As String, ByVal isNumber As
Boolean, ByVal snum As Long)
```

```
    If isNumber = True Then
```

```
        CN.Execute "DELETE FROM " & sTable & " WHERE " & sField & " = " & snum
```

```
    Else
```

```
        CN.Execute "DELETE FROM " & sTable & " WHERE " & sField & " = " & sString & ""
```

```
    End If
```

```
End Sub
```

'Procedure used to fill the listview in paging method

```
Public Sub pageFillListView(ByRef sListView As ListView, ByRef sRecordSource As Recordset, ByVal pos_start As Long,
ByVal pos_end As Long, ByVal sNumOfFields As Byte, ByVal sNumIco As Byte, ByVal with_num As Boolean, ByVal
show_first_rec As Boolean, Optional match_field As String, Optional match_str As String, Optional match_ico As Byte, Optional
srcHiddenField As String)
```

```
    Dim X As ListItem
```

```
    Dim i As Byte, c As Long, old_pt As Long
```

```
    sListView.ListItems.Clear
```

```
    If sRecordSource.RecordCount < 1 Then Exit Sub
```

```
    sRecordSource.AbsolutePosition = pos_start
```

```
    On Error Resume Next
```

```
    old_pt = Screen.MousePointer
```

```
    Screen.MousePointer = vbHourglass
```

```
    DoEvents
```

```
    Do
```

```
        If match_field = "" Then
```

```
            If with_num = True Then
```

```
                Set X = sListView.ListItems.Add(, , "" & sRecordSource.AbsolutePosition, sNumIco, sNumIco)
```

```
            Else
```

```
                Set X = sListView.ListItems.Add(, , "" & FormatRS(sRecordSource.Fields(0)), sNumIco, sNumIco)
```

```
            End If
```

```
        Else
```

```
            If sRecordSource.Fields(match_field) = match_str Then
```

```
                If with_num = True Then
```

```
                    Set X = sListView.ListItems.Add(, , "" & sRecordSource.AbsolutePosition, match_ico, match_ico)
```

```
                Else
```

```
                    Set X = sListView.ListItems.Add(, , "" & FormatRS(sRecordSource.Fields(0)), match_ico, match_ico)
```

```
                End If
```

```
            Else
```

```

    If with_num = True Then
        Set X = sListView.ListItems.Add(, "" & sRecordSource.AbsolutePosition, sNumIco, sNumIco)
    Else
        Set X = sListView.ListItems.Add(, "" & FormatRS(sRecordSource.Fields(0)), sNumIco, sNumIco)
    End If
End If
End If
End If

If srcHiddenField <> "" Then
    X.Tag = sRecordSource.Fields(srcHiddenField) & "*~~~~~*" & c + pos_start
Else
    X.Tag = c + pos_start
End If

For i = 1 To sNumOfFields - 1
    If show_first_rec = True Then
        If with_num = True Then
            X.SubItems(i) = "" & FormatRS(sRecordSource.Fields(CInt(i) - 1))
        Else
            X.SubItems(i) = "" & FormatRS(sRecordSource.Fields(CInt(i)))
        End If
    Else
        X.SubItems(i) = "" & FormatRS(sRecordSource.Fields(CInt(i) + 1))
    End If
Next i

If sRecordSource.AbsolutePosition >= pos_end Then
    Exit Do
Else
    sRecordSource.MoveNext
    c = c + 1
End If

Loop

Screen.MousePointer = old_pt
i = 0: c = 0: old_pt = 0

Set X = Nothing
End Sub

'Procedure used to highlight text when focus
Public Sub HL.Text(ByRef sText)
    On Error Resume Next

```

```

With sText
    .SelStart = 0
    .SelLength = Len(sText.Text)
End With
End Sub

'Procedure used to bind data combo
Public Sub bind_dc(ByVal srcSQL As String, ByVal srcBindField As String, ByRef srcDC As DataCombo, Optional
srcColBound As String, Optional ShowFirstRec As Boolean)

    Dim RS As New Recordset
    RS.CursorLocation = adUseClient
    RS.Open srcSQL, CN, adOpenStatic, adLockOptimistic
    With srcDC
        .ListField = srcBindField
        .BoundColumn = srcColBound
        Set .RowSource = RS
        'Display the first record
        If ShowFirstRec = True Then
            If Not RS.RecordCount < 1 Then
                .BoundText = RS.Fields(srcColBound)
                .Tag = RS.RecordCount & "*****" & RS.Fields(srcColBound)
            Else
                .Tag = "0*****0"
            End If
        End If
    End With
    Set RS = Nothing
End Sub

'Procedure used to bind data list
Public Sub bind_dl(ByVal srcSQL As String, ByVal srcBindField As String, ByRef srcDL As DataList, Optional srcColBound As
String, Optional ShowFirstRec As Boolean)

    Dim RS As New Recordset
    RS.CursorLocation = adUseClient
    RS.Open srcSQL, CN, adOpenStatic, adLockOptimistic
    With srcDL
        .ListField = srcBindField
        .BoundColumn = srcColBound
        Set .RowSource = RS
    End With

```

```

'Display the first record
If ShowFirstRec = True Then
    If Not RS.RecordCount < 1 Then
        .BoundText = RS.Fields(srcColBound)
        .Tag = RS.RecordCount & "*~~~~~*" & RS.Fields(srcColBound)
    Else
        .Tag = "0*~~~~~*0"
    End If
End If
End With
Set RS = Nothing
End Sub

'Procedure used to clear the text content
Public Sub clearText(ByRef sForm As Form)
    Dim Control As Control
    For Each Control In sForm.Controls
        If (TypeOf Control Is TextBox) Then Control = vbNullString
    Next Control
    Set Control = Nothing
End Sub

'Procedure used to clear the text content
Public Sub LockInput(ByRef sForm As Form, ByVal bolLock As Boolean, Optional bolTabStop As Boolean)
    On Error Resume Next
    Dim Control As Control
    For Each Control In sForm.Controls
        Control.Locked = bolLock
    Next Control
    Set Control = Nothing
End Sub

'Procedure that will change the value at once
Public Sub ChangeValue(ByRef srcCN As Connection, ByVal srcTable As String, ByVal srcField As String, ByVal srcValue As
String, Optional isNumber As Boolean, Optional srcCondition As String)
    If srcCondition <> vbNullString Then srcCondition = " " & srcCondition
    If isNumber = True Then
        srcCN.Execute "UPDATE " & srcTable & " SET " & srcField & " = " & srcValue & " " & srcCondition
    Else
        srcCN.Execute "UPDATE " & srcTable & " SET " & srcField & " = " & srcValue & "" & " WHERE " & srcCondition
    End If
End Sub

```



```

End If
End Sub

Public Sub FillFlex(ByRef srcFlex As MSFlexGrid, ByVal srcSQL As String, ByVal srcNoOfCol As Integer)

    Dim RS As New Recordset

    RS.CursorLocation = adUseClient

    RS.Open srcSQL, CN, adOpenStatic, adLockReadOnly

    If RS.RecordCount < 1 Then Exit Sub

    RS.MoveFirst

    Dim i As Long, c As Long

    srcFlex.Rows = (srcFlex.Rows + RS.RecordCount) - 1

    For i = 1 To RS.RecordCount
        For c = 0 To srcNoOfCol - 1
            srcFlex.TextMatrix(i, c) = RS.Fields(c)
        Next c
        RS.MoveNext
    Next i

    i = 0
    c = 0
    Set RS = Nothing
End Sub

'Procedure used to search in listview
Public Sub search_in_listview(ByRef slistView As ListView, ByVal sFindText As String)

    Dim tmp_listview As ListItem

    Set tmp_listview = slistView.FindItem(sFindText, lvwSubItem)

    If Not tmp_listview Is Nothing Then

        tmp_listview.EnsureVisible

        tmp_listview.Selected = True

    End If

End Sub

'Procedure used to center form
Public Sub centerForm(ByRef sForm As Form, ByVal sHeight As Integer, ByVal sWidth As Integer)

    sForm.Move (sWidth - sForm.Width) / 2, (sHeight - sForm.Height) / 2

End Sub

'Procedure used to center object horizontal
Public Sub center_obj_horizontal(ByVal sParentObj As Variant, ByRef sMoveObj As Variant)

    sMoveObj.Left = (sParentObj - sMoveObj.Width) / 2

End Sub

```

'Procedure used to center vertical

```
Public Sub center_obj_vertical(ByVal sParentObj As Variant, ByRef sMoveObj As Variant)
```

```
    sMoveObj.Top = (sParentObj.Height - sMoveObj.Height) / 2
```

```
End Sub
```

```
Public Sub SetINI(strMain As String, strSub As String, strvalue As String)
```

```
    WritePrivateProfileString strMain, strSub, strvalue, App.Path & "\config.txt"
```

```
End Sub
```

ModPublicVar (Module ส่วนของการเก็บลายนิ้วมือผู้ใช้งานผู้สแกนลายนิ้วมือ)

```
Option Explicit
```

```
Public CurrUser As USER_INFO
```

```
Public DBPath As String
```

```
Public Enc As New clsBlowfish (การเรียกใช้งาน clsBlowfish)
```

```
Public CurrBiz As BUSINESS_INFO
```

```
Public CN As New Connection (การเรียกการคิดต่อใหม่ของFinger Scan)
```

ModVarType (Module ส่วนเก็บข้อมูลส่วนบุคคลผู้ใช้งานผู้สแกนลายนิ้วมือ)

'Variable structure for user

```
Public Type USER_INFO
```

```
    USER_PK As Long
```

```
    USER_NAME As String
```

```
    USER_ISADMIN As Boolean
```

```
End Type
```

'Enumerator for form state

```
Public Enum FormState
```

```
    adStateAddMode = 0
```

```
    adStateEditMode = 1
```

```
    adStatePopupMode = 2
```

```
    adStateViewMode = 3
```

```
End Enum
```

```
Public Type BUSINESS_INFO
```

```
    BUSINESS_NAME As String
```

```
    BUSINESS_ADDRESS As String
```

```
    BUSINESS_CONTACT_INFO As String
```

```
End Type
```

CodeVisual Basic ในส่วนของ Class Modules

ClsBlowfish (Class Modules ตักษณะการเก็บลายนิ้วมือ)

Option Explicit

Private Declare Sub CopyMem Lib "kernel32" Alias "RtlMoveMemory" (Destination As Any, Source As Any, ByVal Length As Long)

Event Progress(Percent As Long)

Private Const Rounds = 16

Private m_pBox(0 To Rounds + 1) As Long

Private m_sBox(0 To 3, 0 To 255) As Long

Private m_KeyValue As String

Private m_RunningCompiled As Boolean

Private m_bytIndex(0 To 63) As Byte

Private m_bytReverseIndex(0 To 255) As Byte

Private Const k_bytEqualSign As Byte = 61

Private Const k_bytMask1 As Byte = 3

Private Const k_bytMask2 As Byte = 15

Private Const k_bytMask3 As Byte = 63

Private Const k_bytMask4 As Byte = 192

Private Const k_bytMask5 As Byte = 240

Private Const k_bytMask6 As Byte = 252

Private Const k_bytShift2 As Byte = 4

Private Const k_bytShift4 As Byte = 16

Private Const k_bytShift6 As Byte = 64

Private Const k_lMaxBytesPerLine As Long = 152

Private Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (ByVal Destination As Long, ByVal Source As Long, ByVal Length As Long)

Private Sub Initialize64()

m_bytIndex(0) = 65 'Asc("A")

m_bytIndex(1) = 66 'Asc("B")

m_bytIndex(2) = 67 'Asc("C")

m_bytIndex(3) = 68 'Asc("D")

m_bytIndex(4) = 69 'Asc("E")

m_bytIndex(5) = 70 'Asc("F")

m_bytIndex(6) = 71 'Asc("G")
m_bytIndex(7) = 72 'Asc("H")
m_bytIndex(8) = 73 'Asc("I")
m_bytIndex(9) = 74 'Asc("J")
m_bytIndex(10) = 75 'Asc("K")
m_bytIndex(11) = 76 'Asc("L")
m_bytIndex(12) = 77 'Asc("M")
m_bytIndex(13) = 78 'Asc("N")
m_bytIndex(14) = 79 'Asc("O")
m_bytIndex(15) = 80 'Asc("P")
m_bytIndex(16) = 81 'Asc("Q")
m_bytIndex(17) = 82 'Asc("R")
m_bytIndex(18) = 83 'Asc("S")
m_bytIndex(19) = 84 'Asc("T")
m_bytIndex(20) = 85 'Asc("U")
m_bytIndex(21) = 86 'Asc("V")
m_bytIndex(22) = 87 'Asc("W")
m_bytIndex(23) = 88 'Asc("X")
m_bytIndex(24) = 89 'Asc("Y")
m_bytIndex(25) = 90 'Asc("Z")
m_bytIndex(26) = 97 'Asc("a")
m_bytIndex(27) = 98 'Asc("b")
m_bytIndex(28) = 99 'Asc("c")
m_bytIndex(29) = 100 'Asc("d")
m_bytIndex(30) = 101 'Asc("e")
m_bytIndex(31) = 102 'Asc("f")
m_bytIndex(32) = 103 'Asc("g")
m_bytIndex(33) = 104 'Asc("h")
m_bytIndex(34) = 105 'Asc("i")
m_bytIndex(35) = 106 'Asc("j")
m_bytIndex(36) = 107 'Asc("k")
m_bytIndex(37) = 108 'Asc("l")
m_bytIndex(38) = 109 'Asc("m")
m_bytIndex(39) = 110 'Asc("n")
m_bytIndex(40) = 111 'Asc("o")
m_bytIndex(41) = 112 'Asc("p")
m_bytIndex(42) = 113 'Asc("q")

```

m_bytIndex(43) = 114 'Asc("r")
m_bytIndex(44) = 115 'Asc("s")
m_bytIndex(45) = 116 'Asc("t")
m_bytIndex(46) = 117 'Asc("u")
m_bytIndex(47) = 118 'Asc("v")
m_bytIndex(48) = 119 'Asc("w")
m_bytIndex(49) = 120 'Asc("x")
m_bytIndex(50) = 121 'Asc("y")
m_bytIndex(51) = 122 'Asc("z")
m_bytIndex(52) = 48 'Asc("0")
m_bytIndex(53) = 49 'Asc("1")
m_bytIndex(54) = 50 'Asc("2")
m_bytIndex(55) = 51 'Asc("3")
m_bytIndex(56) = 52 'Asc("4")
m_bytIndex(57) = 53 'Asc("5")
m_bytIndex(58) = 54 'Asc("6")
m_bytIndex(59) = 55 'Asc("7")
m_bytIndex(60) = 56 'Asc("8")
m_bytIndex(61) = 57 'Asc("9")
End Sub
Public Function Decode64(sInput As String) As String
    If sInput = "" Then Exit Function
    Decode64 = StrConv(DecodeArray64(sInput), vbUnicode)
End Function
Public Function DecodeArray64(sInput As String) As Byte()
    If m_bytReverseIndex(47) <> 63 Then Initialize64
    Dim bytInput() As Byte
    Dim bytWorkspace() As Byte
    Dim bytResult() As Byte
    Dim IInputCounter As Long
    Dim IWorkspaceCounter As Long
    bytInput = Replace(Replace(sInput, vbCrLf, ""), "=", "")
    ReDim bytWorkspace(LBound(bytInput) To (UBound(bytInput) * 2)) As Byte
    IWorkspaceCounter = LBound(bytWorkspace)
    For IInputCounter = LBound(bytInput) To UBound(bytInput)
        bytInput(IInputCounter) = m_bytReverseIndex(bytInput(IInputCounter))
    Next IInputCounter

```

```

For IInputCounter = LBound(bytInput) To (UBound(bytInput) - ((UBound(bytInput) Mod 8) + 8)) Step 8
    bytWorkspace(IWorkspaceCounter) = (bytInput(IInputCounter) * k_bytShift2) + (bytInput(IInputCounter + 2) \ k_bytShift4)
    bytWorkspace(IWorkspaceCounter + 1) = ((bytInput(IInputCounter + 2) And k_bytMask2) * k_bytShift4) +
    (bytInput(IInputCounter + 4) \ k_bytShift2)
    bytWorkspace(IWorkspaceCounter + 2) = ((bytInput(IInputCounter + 4) And k_bytMask1) * k_bytShift6) +
    bytInput(IInputCounter + 6)
    IWorkspaceCounter = IWorkspaceCounter + 3
Next IInputCounter
Select Case (UBound(bytInput) Mod 8):
    Case 3:
        bytWorkspace(IWorkspaceCounter) = (bytInput(IInputCounter) * k_bytShift2) + (bytInput(IInputCounter + 2) \
    k_bytShift4)
    Case 5:
        bytWorkspace(IWorkspaceCounter) = (bytInput(IInputCounter) * k_bytShift2) + (bytInput(IInputCounter + 2) \
    k_bytShift4)
        bytWorkspace(IWorkspaceCounter + 1) = ((bytInput(IInputCounter + 2) And k_bytMask2) * k_bytShift4) +
    (bytInput(IInputCounter + 4) \ k_bytShift2)
        IWorkspaceCounter = IWorkspaceCounter + 1
    Case 7:
        bytWorkspace(IWorkspaceCounter) = (bytInput(IInputCounter) * k_bytShift2) + (bytInput(IInputCounter + 2) \
    k_bytShift4)
        bytWorkspace(IWorkspaceCounter + 1) = ((bytInput(IInputCounter + 2) And k_bytMask2) * k_bytShift4) +
    (bytInput(IInputCounter + 4) \ k_bytShift2)
        bytWorkspace(IWorkspaceCounter + 2) = ((bytInput(IInputCounter + 4) And k_bytMask1) * k_bytShift6) +
    bytInput(IInputCounter + 6)
        IWorkspaceCounter = IWorkspaceCounter + 2
End Select
ReDim bytResult(LBound(bytWorkspace) To IWorkspaceCounter) As Byte
If LBound(bytWorkspace) = 0 Then IWorkspaceCounter = IWorkspaceCounter + 1
CopyMemory VarPtr(bytResult(LBound(bytResult))), VarPtr(bytWorkspace(LBound(bytWorkspace))), IWorkspaceCounter
DecodeArray64 = bytResult
End Function
Public Function lEncode64(ByRef sInput As String) As String
    If sInput = "" Then Exit Function
    Dim bytTemp() As Byte
    bytTemp = StrConv(sInput, vbFromUnicode)
    Encode64 = EncodeArray64(bytTemp)

```

```

End Function

Public Function EncodeArray64(ByRef bytInput() As Byte) As String

    On Error GoTo ErrorHandler

    If m_bytReverseIndex(47) <> 63 Then Initialize64

    Dim bytWorkspace() As Byte, bytResult() As Byte

    Dim bytCrLf(0 To 3) As Byte, lCounter As Long

    Dim lWorkspaceCounter As Long, lLineCounter As Long

    Dim lCompleteLines As Long, lBytesRemaining As Long

    Dim lpWorkSpace As Long, lpResult As Long

    Dim lpCrLf As Long

    If UBound(bytInput) < 1024 Then

        ReDim bytWorkspace(LBound(bytInput) To (LBound(bytInput) + 4096)) As Byte

    Else

        ReDim bytWorkspace(LBound(bytInput) To (UBound(bytInput) * 4)) As Byte

    End If

    lWorkspaceCounter = LBound(bytWorkspace)

    For lCounter = LBound(bytInput) To (UBound(bytInput) - ((UBound(bytInput) Mod 3) + 3)) Step 3

        bytWorkspace(lWorkspaceCounter) = m_bytIndex((bytInput(lCounter) \ k_bytShift2))

        bytWorkspace(lWorkspaceCounter + 2) = m_bytIndex(((bytInput(lCounter) And k_bytMask1) * k_bytShift4) +
        ((bytInput(lCounter + 1)) \ k_bytShift4))

        bytWorkspace(lWorkspaceCounter + 4) = m_bytIndex(((bytInput(lCounter + 1) And k_bytMask2) * k_bytShift2) +
        (bytInput(lCounter + 2) \ k_bytShift6))

        bytWorkspace(lWorkspaceCounter + 6) = m_bytIndex((bytInput(lCounter + 2) And k_bytMask3))

        lWorkspaceCounter = lWorkspaceCounter + 8

    Next lCounter

    Select Case (UBound(bytInput) Mod 3):

        Case 0:

            bytWorkspace(lWorkspaceCounter) = m_bytIndex((bytInput(lCounter) \ k_bytShift2))

            bytWorkspace(lWorkspaceCounter + 2) = m_bytIndex((bytInput(lCounter) And k_bytMask1) * k_bytShift4)

            bytWorkspace(lWorkspaceCounter + 4) = k_bytEqualSign

            bytWorkspace(lWorkspaceCounter + 6) = k_bytEqualSign

        Case 1:

            bytWorkspace(lWorkspaceCounter) = m_bytIndex((bytInput(lCounter) \ k_bytShift2))

            bytWorkspace(lWorkspaceCounter + 2) = m_bytIndex(((bytInput(lCounter) And k_bytMask1) * k_bytShift4) +
            ((bytInput(lCounter + 1)) \ k_bytShift4))

            bytWorkspace(lWorkspaceCounter + 4) = m_bytIndex((bytInput(lCounter + 1) And k_bytMask2) * k_bytShift2)

            bytWorkspace(lWorkspaceCounter + 6) = k_bytEqualSign

```

Case 2:

```

bytWorkspace(IWorkspaceCounter) = m_bytIndex((bytInput(lCounter) \ k_bytShift2))
bytWorkspace(IWorkspaceCounter + 2) = m_bytIndex(((bytInput(lCounter) And k_bytMask1) * k_bytShift4) +
((bytInput(lCounter + 1)) \ k_bytShift4))
bytWorkspace(IWorkspaceCounter + 4) = m_bytIndex(((bytInput(lCounter + 1) And k_bytMask2) * k_bytShift2) +
((bytInput(lCounter + 2)) \ k_bytShift6))
bytWorkspace(IWorkspaceCounter + 6) = m_bytIndex(bytInput(lCounter + 2) And k_bytMask3)

```

End Select

IWorkspaceCounter = IWorkspaceCounter + 8

If IWorkspaceCounter <= k_I_MaxBytesPerLine Then

```
EncodeArray64 = Left$(bytWorkspace, InStr(1, bytWorkspace, Chr$(0)) - 1)
```

Else

```
bytCrLf(0) = 13
```

```
bytCrLf(1) = 0
```

```
bytCrLf(2) = 10
```

```
bytCrLf(3) = 0
```

```
ReDim bytResult(LBound(bytWorkspace) To UBound(bytWorkspace))
```

```
lpWorkSpace = VarPtr(bytWorkspace(LBound(bytWorkspace)))
```

```
lpResult = VarPtr(bytResult(LBound(bytResult)))
```

```
lpCrLf = VarPtr(bytCrLf(LBound(bytCrLf)))
```

```
ICompleteLines = Fix(IWorkspaceCounter / k_I_MaxBytesPerLine)
```

```
For lLineCounter = 0 To ICompleteLines
```

```
CopyMemory lpResult, lpWorkSpace, k_I_MaxBytesPerLine
```

```
lpWorkSpace = lpWorkSpace + k_I_MaxBytesPerLine
```

```
lpResult = lpResult + k_I_MaxBytesPerLine
```

```
CopyMemory lpResult, lpCrLf, 4&
```

```
lpResult = lpResult + 4&
```

```
Next lLineCounter
```

```
lBytesRemaining = IWorkspaceCounter - (ICompleteLines * k_I_MaxBytesPerLine)
```

```
If lBytesRemaining > 0 Then CopyMemory lpResult, lpWorkSpace, lBytesRemaining
```

```
EncodeArray64 = Left$(bytResult, InStr(1, bytResult, Chr$(0)) - 1)
```

End If

Exit Function

ErrorHandler:

```
Erase bytResult
```

```
EncodeArray64 = bytResult
```

End Function


```
Private Static Sub DecryptBlock(Xl As Long, Xr As Long)
```

```
    Dim i As Long, j As Long, K As Long
```

```
    K = Xr
```

```
    Xr = Xl Xor m_pBox(Rounds + 1)
```

```
    Xl = K Xor m_pBox(Rounds)
```

```
    j = Rounds - 2
```

```
    For i = 0 To (Rounds \ 2 - 1)
```

```
        Xl = Xl Xor f(Xr)
```

```
        Xr = Xr Xor m_pBox(j + 1)
```

```
        Xr = Xr Xor f(Xl)
```

```
        Xl = Xl Xor m_pBox(j)
```

```
        j = j - 2
```

```
    Next
```

```
End Sub
```

```
Private Static Sub EncryptBlock(Xl As Long, Xr As Long)
```

```
    Dim i As Long, j As Long, Temp As Long
```

```
    j = 0
```

```
    For i = 0 To (Rounds \ 2 - 1)
```

```
        Xl = Xl Xor m_pBox(j)
```

```
        Xr = Xr Xor f(Xl)
```

```
        Xr = Xr Xor m_pBox(j + 1)
```

```
        Xl = Xl Xor f(Xr)
```

```
        j = j + 2
```

```
    Next
```

```
    Temp = Xr
```

```
    Xr = Xl Xor m_pBox(Rounds)
```

```
    Xl = Temp Xor m_pBox(Rounds + 1)
```

```
End Sub
```

```
Public Sub EncryptByte(byteArray() As Byte, Optional Key As String)
```

```
    Dim Offset As Long, OrigLen As Long, LeftWord As Long, RightWord As Long, CipherLen As Long, CipherLeft As Long,
```

```
    CipherRight As Long, CurrPercent As Long, NextPercent As Long
```

```
    If (Len(Key) > 0) Then Me.Key = Key
```

```
    OrigLen = UBound(byteArray) + 1
```

```
    CipherLen = OrigLen + 12
```

```
    If (CipherLen Mod 8 <> 0) Then CipherLen = CipherLen + 8 - (CipherLen Mod 8)
```

```
    ReDim Preserve byteArray(CipherLen - 1)
```

```
    Call CopyMem(byteArray(12), byteArray(0), OrigLen)
```

```

Call CopyMem(byteArray(8), OrigLen, 4)

Call Randomize

Call CopyMem(byteArray(0), CLng(2147483647 * Rnd), 4)

Call CopyMem(byteArray(4), CLng(2147483647 * Rnd), 4)

For Offset = 0 To (CipherLen - 1) Step 8
    Call GetWord(LeftWord, byteArray(), Offset)

    Call GetWord(RightWord, byteArray(), Offset + 4)

    LeftWord = LeftWord Xor CipherLeft

    RightWord = RightWord Xor CipherRight

    Call EncryptBlock(LeftWord, RightWord)

    Call PutWord(LeftWord, byteArray(), Offset)

    Call PutWord(RightWord, byteArray(), Offset + 4)

    CipherLeft = LeftWord

    CipherRight = RightWord

    If (Offset >= NextPercent) Then
        CurrPercent = Int((Offset / CipherLen) * 100)

        NextPercent = (CipherLen * ((CurrPercent + 1) / 100)) + 1

        RaiseEvent Progress(CurrPercent)

    End If Next

Call CopyMem(OrigLen, byteArray(8), 4)

If (CipherLen - OrigLen > 19) Or (CipherLen - OrigLen < 12) Then Call err.Raise(vbObjectError, , "Incorrect size descriptor in
Blowfish decryption")

Call CopyMem(byteArray(0), byteArray(12), OrigLen)

ReDim Preserve byteArray(OrigLen - 1)

If CurrPercent <> 100 Then RaiseEvent Progress(100)

ErrorHandler:

End Sub

Private Static Function f(ByVal x As Long) As Long

    Dim xb(0 To 3) As Byte

    Call CopyMem(xb(0), x, 4)

    If (m_RunningCompiled) Then f = (((m_sBox(0, xb(3)) + m_sBox(1, xb(2))) Xor m_sBox(2, xb(1))) + m_sBox(3, xb(0))) Else
f = UnsignedAdd((UnsignedAdd(m_sBox(0, xb(3)), m_sBox(1, xb(2))) Xor m_sBox(2, xb(1))), m_sBox(3, xb(0)))

    End Function

Private Static Sub GetWord(LongValue As Long, CryptBuffer() As Byte, Offset As Long)

    Dim bb(0 To 3) As Byte

    bb(3) = CryptBuffer(Offset)

    bb(2) = CryptBuffer(Offset + 1)

```

```

bb(1) = CryptBuffer(Offset + 2)
bb(0) = CryptBuffer(Offset + 3)
Call CopyMem(LongValue, bb(0), 4)
End Sub

Private Static Sub PutWord(LongValue As Long, CryptBuffer() As Byte, Offset As Long)
    Dim bb(0 To 3) As Byte
    Call CopyMem(bb(0), LongValue, 4)
    CryptBuffer(Offset) = bb(3)
    CryptBuffer(Offset + 1) = bb(2)
    CryptBuffer(Offset + 2) = bb(1)
    CryptBuffer(Offset + 3) = bb(0)
End Sub

Private Static Function UnsignedAdd(ByVal Data1 As Long, Data2 As Long) As Long
    Dim x1(0 To 3) As Byte, x2(0 To 3) As Byte, xx(0 To 3) As Byte, Rest As Long, Value As Long, a As Long
    Call CopyMem(x1(0), Data1, 4)
    Call CopyMem(x2(0), Data2, 4)
    Rest = 0
    For a = 0 To 3
        Value = CLng(x1(a)) + CLng(x2(a)) + Rest
        xx(a) = Value And 255
        Rest = Value \ 256
    Next
    Call CopyMem(UnsignedAdd, xx(0), 4)
End Function

Private Function UnsignedDel(Data1 As Long, Data2 As Long) As Long
    Dim x1(0 To 3) As Byte, x2(0 To 3) As Byte, xx(0 To 3) As Byte, Rest As Long, Value As Long, a As Long
    Call CopyMem(x1(0), Data1, 4)
    Call CopyMem(x2(0), Data2, 4)
    Call CopyMem(xx(0), UnsignedDel, 4)
    For a = 0 To 3
        Value = CLng(x1(a)) - CLng(x2(a)) - Rest
        If (Value < 0) Then
            Value = Value + 256
            Rest = 1
        Else
            Rest = 0
        End If
    Next
End Function

```

```

        xx(a) = Value
    Next
    Call CopyMem(UndsignedDel, xx(0), 4)
End Function

Public Property Let Key(New_Value As String)
    Dim i As Long, j As Long, K As Long, dataX As Long, data1 As Long, datar As Long, Key() As Byte, KeyLength As Long
    Class_Initialize
    If (m_KeyValue = New_Value) Then Exit Property
    m_KeyValue = New_Value
    KeyLength = Len(New_Value)
    Key() = StrConv(New_Value, vbFromUnicode)
    j = 0
    For i = 0 To (Rounds + 1)
        dataX = 0
        For K = 0 To 3
            Call CopyMem(ByVal VarPtr(dataX) + 1, dataX, 3)
            dataX = (dataX Or Key(j))
            j = j + 1
            If (j >= KeyLength) Then j = 0
        Next
        m_pBox(i) = m_pBox(i) Xor dataX
    Next
    data1 = 0: datar = 0
    For i = 0 To (Rounds + 1) Step 2
        Call EncryptBlock(data1, datar)
        m_pBox(i) = data1
        m_pBox(i + 1) = datar
    Next
    For i = 0 To 3
        For j = 0 To 255 Step 2
            Call EncryptBlock(data1, datar)
            m_sBox(i, j) = data1
            m_slBox(i, j + 1) = datar
        Next
    Next
End Property

Private Sub Class_Initialize()

```

On Local Error Resume Next

m_RunningCompiled = ((2147483647 + 1) < 0)

m_pBox(0) = &H243F6A88

m_pBox(1) = &H85A308D3

m_pBox(2) = &H13198A2E

m_pBox(3) = &H13707344

m_pBox(4) = &HA4093822

m_pBox(5) = &H299F31D0

m_pBox(6) = &H82EFA98

m_pBox(7) = &HEC4E6C89

m_pBox(8) = &H452821E6

m_pBox(9) = &H38D01377

m_pBox(10) = &HBE5466CF

m_pBox(11) = &H34190C6C

m_pBox(12) = &H1C0AC29B7

m_pBox(13) = &HC97C50DD

m_pBox(14) = &H3F84D5B5

m_pBox(15) = &HB5470917

m_pBox(16) = &H9216D51D9

m_pBox(17) = &H8979F131B

m_sBox(0, 0) = &HD1310BA6

m_sBox(1, 0) = &H98DFB5AC

m_sBox(2, 0) = &H2FFD72DB

m_sBox(3, 0) = &HD01ADFB7

m_sBox(0, 1) = &HB8E1AFED

m_sBox(1, 1) = &H6A267E96

m_sBox(2, 1) = &HBA7C9045

m_sBox(3, 1) = &H1F12C7F99

m_sBox(0, 2) = &H24A19947

m_sBox(1, 2) = &HB3916CF7

m_sBox(2, 2) = &H801F2E2

m_sBox(3, 2) = &H858EFC16

m_sBox(0, 3) = &H636920D8

m_sBox(1, 3) = &H71574E69

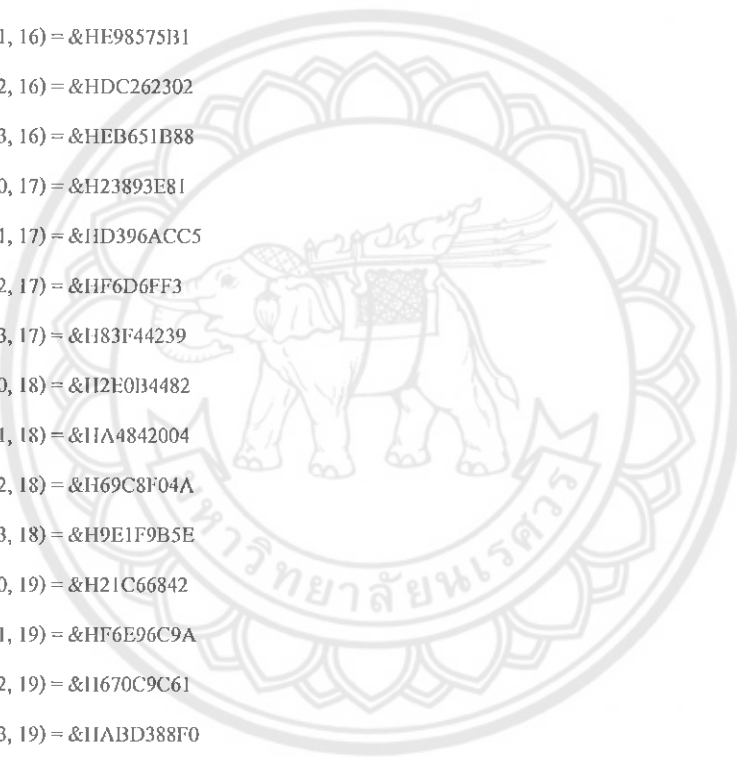
m_sBox(2, 3) = &HA458FEA3

m_sBox(3, 3) = &HF4933D7E

m_sBox(0, 4) = &HD95748F

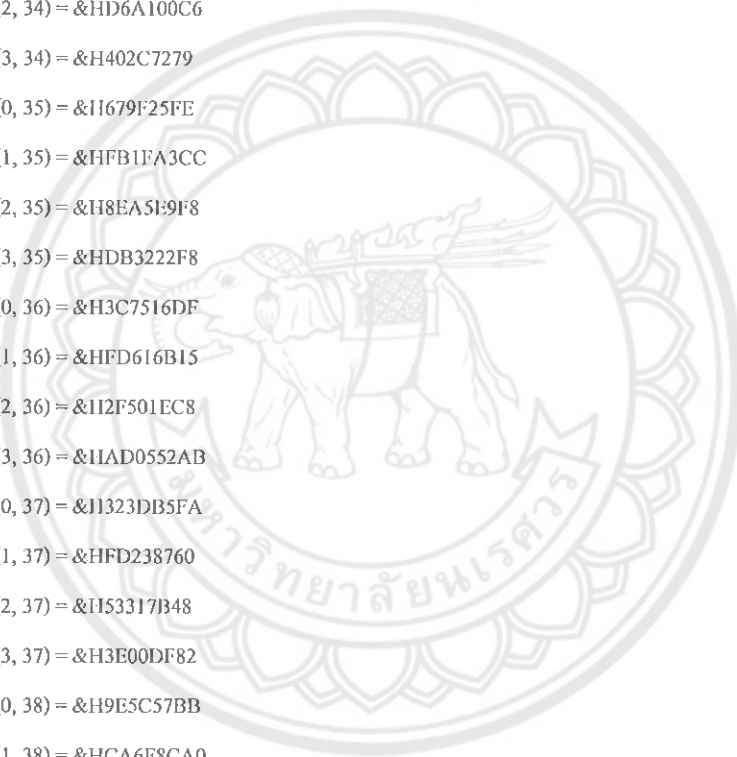
m_sBox(1, 4) = &H728EB658
m_sBox(2, 4) = &H718BCD58
m_sBox(3, 4) = &H82154AFE
m_sBox(0, 5) = &H7B54A41D
m_sBox(1, 5) = &HC25A59B5
m_sBox(2, 5) = &H9C30D539
m_sBox(3, 5) = &H2AF26013
m_sBox(0, 6) = &HC5D1B023
m_sBox(1, 6) = &H286085F0
m_sBox(2, 6) = &HCA417918
m_sBox(3, 6) = &HB8DB38EF
m_sBox(0, 7) = &H8E79DCB0
m_sBox(1, 7) = &H603A180E
m_sBox(2, 7) = &H6C9E0E8B
m_sBox(3, 7) = &H1B01E8A3E
m_sBox(0, 8) = &HD71577C1
m_sBox(1, 8) = &HBD314B27
m_sBox(2, 8) = &H78AF2FDA
m_sBox(3, 8) = &H55605C60
m_sBox(0, 9) = &HE65525F3
m_sBox(1, 9) = &HAA55AB94
m_sBox(2, 9) = &H57489862
m_sBox(3, 9) = &H63E81440
m_sBox(0, 10) = &H55CA396A
m_sBox(1, 10) = &H2AAB10B6
m_sBox(2, 10) = &HB4CC5C34
m_sBox(3, 10) = &H1141E8CE
m_sBox(0, 11) = &HA15486AF
m_sBox(1, 11) = &H7C72E993
m_sBox(2, 11) = &HB3EE1411
m_sBox(3, 11) = &H636FBC2A
m_sBox(0, 12) = &H2BA9C55D
m_sBox(1, 12) = &H7418311F6
m_sBox(2, 12) = &H1CE5C3E16
m_sBox(3, 12) = &H9B87931E
m_sBox(0, 13) = &HAFD6BA33
m_sBox(1, 13) = &H6C24CF5C

m_sBox(2, 13) = &117A325381
m_sBox(3, 13) = &H28958677
m_sBox(0, 14) = &113B814898
m_sBox(1, 14) = &H6B4BB9AF
m_sBox(2, 14) = &HC4BFE81B
m_sBox(3, 14) = &H66282193
m_sBox(0, 15) = &H61D809CC
m_sBox(1, 15) = &11F321A991
m_sBox(2, 15) = &H487CAC60
m_sBox(3, 15) = &115DFC8032
m_sBox(0, 16) = &HEF845D5D
m_sBox(1, 16) = &HE98575B1
m_sBox(2, 16) = &HDC262302
m_sBox(3, 16) = &HEB651B88
m_sBox(0, 17) = &H23893E81
m_sBox(1, 17) = &11D396ACC5
m_sBox(2, 17) = &11F6D6FF3
m_sBox(3, 17) = &H83F44239
m_sBox(0, 18) = &112E0B4482
m_sBox(1, 18) = &11A4842004
m_sBox(2, 18) = &H69C8F04A
m_sBox(3, 18) = &H9E1F9B5E
m_sBox(0, 19) = &H21C66842
m_sBox(1, 19) = &HF6E96C9A
m_sBox(2, 19) = &11670C9C61
m_sBox(3, 19) = &11ABD388F0
m_sBox(0, 20) = &116A51A01D2
m_sBox(1, 20) = &HD8542F68
m_sBox(2, 20) = &H960FA728
m_sBox(3, 20) = &HAB5133A3
m_sBox(0, 21) = &116EEF0B6C
m_sBox(1, 21) = &H137A3BE4
m_sBox(2, 21) = &11BA3BF050
m_sBox(3, 21) = &H7EFB2A98
m_sBox(0, 22) = &HA1F1651D
m_sBox(1, 22) = &H39AF0176
m_sBox(2, 22) = &1166CA593E



m_sBox(3, 22) = &H82430E88
m_sBox(0, 23) = &H8CEE8619
m_sBox(1, 23) = &H456F9FB4
m_sBox(2, 23) = &H7D84A5C3
m_sBox(3, 23) = &H3B8B5FBE
m_sBox(0, 24) = &HE06F75D8
m_sBox(1, 24) = &H85C12073
m_sBox(2, 24) = &H401A449F
m_sBox(3, 24) = &H56C16AA6
m_sBox(0, 25) = &H4ED3AA62
m_sBox(1, 25) = &H363F7706
m_sBox(2, 25) = &H1BFEDF72
m_sBox(3, 25) = &H429B023D
m_sBox(0, 26) = &H37D0D724
m_sBox(1, 26) = &HD00A1248
m_sBox(2, 26) = &HDB0FEAD3
m_sBox(3, 26) = &H49F1C09B
m_sBox(0, 27) = &H75372C9
m_sBox(1, 27) = &H80991B7B
m_sBox(2, 27) = &H25D479D8
m_sBox(3, 27) = &HF6E8DEF7
m_sBox(0, 28) = &HE3FE501A
m_sBox(1, 28) = &HB6794C3B
m_sBox(2, 28) = &H976CE0BD
m_sBox(3, 28) = &H4C006BA
m_sBox(0, 29) = &HC1A94FB6
m_sBox(1, 29) = &H409F60C4
m_sBox(2, 29) = &H5E5C9EC2
m_sBox(3, 29) = &H196A2463
m_sBox(0, 30) = &H68FB6FAF
m_sBox(1, 30) = &H3E6C53B5
m_sBox(2, 30) = &H1339B2EB
m_sBox(3, 30) = &H13B52EC6F
m_sBox(0, 31) = &H6DFC511F
m_sBox(1, 31) = &H9B30952C
m_sBox(2, 31) = &HCC814544
m_sBox(3, 31) = &HAF5EBD09

m_sBox(0, 32) = &HBEE3D004
m_sBox(1, 32) = &HDE334AFD
m_sBox(2, 32) = &H660F2807
m_sBox(3, 32) = &H1192E4BB3
m_sBox(0, 33) = &HC0CBA857
m_sBox(1, 33) = &H45C8740F
m_sBox(2, 33) = &HD20B5F39
m_sBox(3, 33) = &HB9D3FBDB
m_sBox(0, 34) = &H5579C0BD
m_sBox(1, 34) = &H1A60320A
m_sBox(2, 34) = &HD6A100C6
m_sBox(3, 34) = &H402C7279
m_sBox(0, 35) = &H1679F25FE
m_sBox(1, 35) = &HFB1FA3CC
m_sBox(2, 35) = &H8EA519F8
m_sBox(3, 35) = &HDB3222F8
m_sBox(0, 36) = &H3C7516DF
m_sBox(1, 36) = &HFD616B15
m_sBox(2, 36) = &H12F501EC8
m_sBox(3, 36) = &H1AD0552AB
m_sBox(0, 37) = &H1323DB5FA
m_sBox(1, 37) = &HFD238760
m_sBox(2, 37) = &H53317B48
m_sBox(3, 37) = &H3E00DF82
m_sBox(0, 38) = &H9E5C57BB
m_sBox(1, 38) = &HCA6F8CA0
m_sBox(2, 38) = &H1A87562E
m_sBox(3, 38) = &HDF1769DB
m_sBox(0, 39) = &H1D542A8F6
m_sBox(1, 39) = &H287EFC3
m_sBox(2, 39) = &HAC6732C6
m_sBox(3, 39) = &H8C4F5573
m_sBox(0, 40) = &H1695B27B0
m_sBox(1, 40) = &HBBCA58C8
m_sBox(2, 40) = &HE1FFA35D
m_sBox(3, 40) = &HB8F011A0
m_sBox(0, 41) = &H10FA3D98



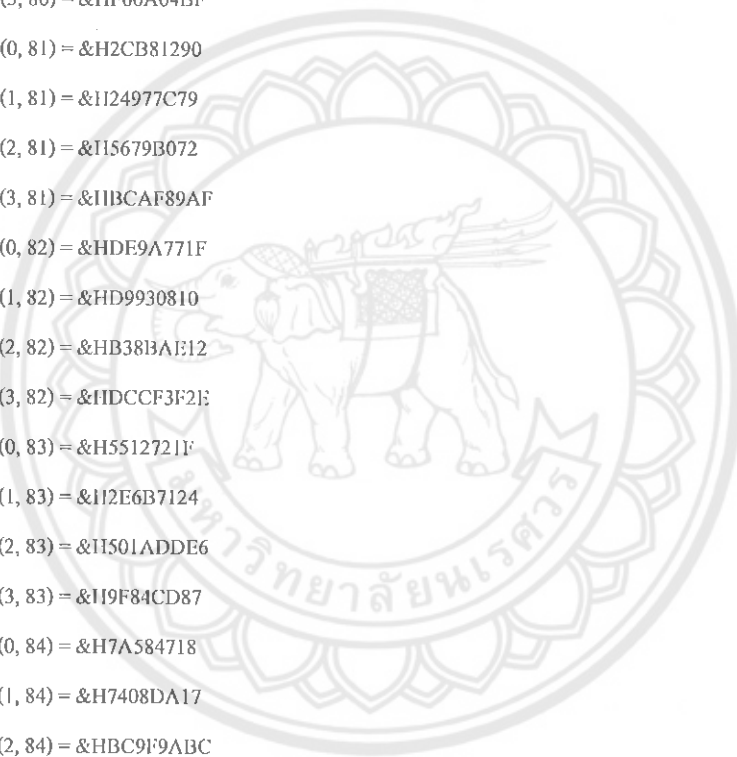
m_sBox(1, 41) = &HFD2183B8
m_sBox(2, 41) = &H14AFCB56C
m_sBox(3, 41) = &H2DD1D35B
m_sBox(0, 42) = &H19A53E479
m_sBox(1, 42) = &HB6F84565
m_sBox(2, 42) = &HD28E49BC
m_sBox(3, 42) = &H4BF139790
m_sBox(0, 43) = &HE1DDF2DA
m_sBox(1, 43) = &HA4CB7133
m_sBox(2, 43) = &H62FB1341
m_sBox(3, 43) = &HCCE4C6E8
m_sBox(0, 44) = &HEF20CADA
m_sBox(1, 44) = &H136774C01
m_sBox(2, 44) = &HD0719FFE
m_sBox(3, 44) = &H2BF11FB4
m_sBox(0, 45) = &H95DBDA4D
m_sBox(1, 45) = &HAE909198
m_sBox(2, 45) = &HEAAD8E71
m_sBox(3, 45) = &H6B93D5A0
m_sBox(0, 46) = &HD08ED1D0
m_sBox(1, 46) = &H1AFC725E0
m_sBox(2, 46) = &H8F3C5B2F
m_sBox(3, 46) = &H8F7594B7
m_sBox(0, 47) = &H8F16F21B
m_sBox(1, 47) = &HF2122B64
m_sBox(2, 47) = &H8888B812
m_sBox(3, 47) = &H900DF01C
m_sBox(0, 48) = &H14FAD5EA0
m_sBox(1, 48) = &H688FC31C
m_sBox(2, 48) = &HD1CFF191
m_sBox(3, 48) = &HB3A8C1AD
m_sBox(0, 49) = &H2F2F2218
m_sBox(1, 49) = &HBE0E1777
m_sBox(2, 49) = &H1EA752DFE
m_sBox(3, 49) = &H8B021FA1
m_sBox(0, 50) = &HE5A0CC0F
m_sBox(1, 50) = &HB56F74E8

m_sBox(2, 50) = &H18ACF3D6
m_sBox(3, 50) = &HCE891E299
m_sBox(0, 51) = &HB4A84FF0
m_sBox(1, 51) = &HFD13E0B7
m_sBox(2, 51) = &H7CC43B81
m_sBox(3, 51) = &HD2ADA8D9
m_sBox(0, 52) = &H165FA266
m_sBox(1, 52) = &H80957705
m_sBox(2, 52) = &H193CC7314
m_sBox(3, 52) = &H211A1477
m_sBox(0, 53) = &HE6AD2065
m_sBox(1, 53) = &H77B5FA86
m_sBox(2, 53) = &HC75442F5
m_sBox(3, 53) = &HFB9D35CF
m_sBox(0, 54) = &HEBCDAF0C
m_sBox(1, 54) = &H7B3E89A0
m_sBox(2, 54) = &HD6411BD3
m_sBox(3, 54) = &H1AE1E7E49
m_sBox(0, 55) = &H250E2D
m_sBox(1, 55) = &H2071B35E
m_sBox(2, 55) = &H226800BB
m_sBox(3, 55) = &H57B8E0AF
m_sBox(0, 56) = &H2464369B
m_sBox(1, 56) = &HF009B91E
m_sBox(2, 56) = &H5563911D
m_sBox(3, 56) = &H59DFA6AA
m_sBox(0, 57) = &H78C14389
m_sBox(1, 57) = &HD95A537F
m_sBox(2, 57) = &H207D5BA2
m_sBox(3, 57) = &H2E5B9C5
m_sBox(0, 58) = &H83260376
m_sBox(1, 58) = &H6295CFA9
m_sBox(2, 58) = &H111C81968
m_sBox(3, 58) = &H4E734A41
m_sBox(0, 59) = &H1B3472DCA
m_sBox(1, 59) = &H7B14A94A
m_sBox(2, 59) = &H1B510052

m_sBox(3, 59) = &H9A532915
m_sBox(0, 60) = &HD60F573F
m_sBox(1, 60) = &HBC9BC6E4
m_sBox(2, 60) = &H2B60A476
m_sBox(3, 60) = &H81E67400
m_sBox(0, 61) = &H8BA6FB5
m_sBox(1, 61) = &I1571BE91F
m_sBox(2, 61) = &HF296EC6B
m_sBox(3, 61) = &I12A0DD915
m_sBox(0, 62) = &HB6636521
m_sBox(1, 62) = &HE7B91F9B6
m_sBox(2, 62) = &HFF34052E
m_sBox(3, 62) = &HC5855664
m_sBox(0, 63) = &I153B02D5D
m_sBox(1, 63) = &IA99F8FA1
m_sBox(2, 63) = &I18BA4799
m_sBox(3, 63) = &H6E85076A
m_sBox(0, 64) = &H4B7A70E9
m_sBox(1, 64) = &HB5B32944
m_sBox(2, 64) = &HDB75092E
m_sBox(3, 64) = &HC4192623
m_sBox(0, 65) = &HAD6EA6B0
m_sBox(1, 65) = &H49A7DF7D
m_sBox(2, 65) = &H9CEE60B8
m_sBox(3, 65) = &H81FEDB266
m_sBox(0, 66) = &H1CAA8C71
m_sBox(1, 66) = &H699A17FF
m_sBox(2, 66) = &H5664526C
m_sBox(3, 66) = &HC2B19FE1
m_sBox(0, 67) = &H193602A5
m_sBox(1, 67) = &I175094C29
m_sBox(2, 67) = &IA0591340
m_sBox(3, 67) = &HE4183A3E
m_sBox(0, 68) = &H3F54989A
m_sBox(1, 68) = &H5B429D65
m_sBox(2, 68) = &H6B8FE4D6
m_sBox(3, 68) = &I199F73FD6

m_sBox(0, 69) = &HA1D29C07
m_sBox(1, 69) = &11EFE830F5
m_sBox(2, 69) = &114D2D38E6
m_sBox(3, 69) = &11F0255DC1
m_sBox(0, 70) = &H4CD12086
m_sBox(1, 70) = &H8470EB26
m_sBox(2, 70) = &H6382E9C6
m_sBox(3, 70) = &H21ECC5E1
m_sBox(0, 71) = &H9686B3F
m_sBox(1, 71) = &H3EBAEFC9
m_sBox(2, 71) = &113C971814
m_sBox(3, 71) = &H6B6A70A1
m_sBox(0, 72) = &11687F3584
m_sBox(1, 72) = &H52A0E286
m_sBox(2, 72) = &HB79C5305
m_sBox(3, 72) = &HAA500737
m_sBox(0, 73) = &H3E07841C
m_sBox(1, 73) = &H7FDEAE5C
m_sBox(2, 73) = &118E7D44EC
m_sBox(3, 73) = &H5716F2B8
m_sBox(0, 74) = &11B03ADA37
m_sBox(1, 74) = &HF0500COD
m_sBox(2, 74) = &HF01C1F04
m_sBox(3, 74) = &H200B3FF
m_sBox(0, 75) = &HAE0CF51A
m_sBox(1, 75) = &H3CB574B2
m_sBox(2, 75) = &H25837A58
m_sBox(3, 75) = &HDC0921BD
m_sBox(0, 76) = &HD19113F9
m_sBox(1, 76) = &H7CA92FF6
m_sBox(2, 76) = &H94324773
m_sBox(3, 76) = &H22F54701
m_sBox(0, 77) = &H3AE5E581
m_sBox(1, 77) = &1137C2DADC
m_sBox(2, 77) = &11C8B57634
m_sBox(3, 77) = &H9AF3DDA7
m_sBox(0, 78) = &HA9446146

m_sBox(1, 78) = &HFD0030E
m_sBox(2, 78) = &HECC8C73E
m_sBox(3, 78) = &HA4751E41
m_sBox(0, 79) = &HE238CD99
m_sBox(1, 79) = &H3BEA0E2F
m_sBox(2, 79) = &H3280BBA1
m_sBox(3, 79) = &H1183EB331
m_sBox(0, 80) = &H4E548B38
m_sBox(1, 80) = &H4F6DB908
m_sBox(2, 80) = &H6F420D03
m_sBox(3, 80) = &HF60A04BF
m_sBox(0, 81) = &H2CB81290
m_sBox(1, 81) = &H24977C79
m_sBox(2, 81) = &H15679B072
m_sBox(3, 81) = &H3CAF89AF
m_sBox(0, 82) = &HDE9A771F
m_sBox(1, 82) = &HD9930810
m_sBox(2, 82) = &HB38BAE12
m_sBox(3, 82) = &HDCCF3FE
m_sBox(0, 83) = &H5512721F
m_sBox(1, 83) = &H2E6B7124
m_sBox(2, 83) = &H501ADDE6
m_sBox(3, 83) = &H9F84CD87
m_sBox(0, 84) = &H7A584718
m_sBox(1, 84) = &H7408DA17
m_sBox(2, 84) = &HBC9F9ABC
m_sBox(3, 84) = &HE94B7D8C
m_sBox(0, 85) = &HFC7AEC3A
m_sBox(1, 85) = &HDB851DFA
m_sBox(2, 85) = &H63094366
m_sBox(3, 85) = &HC464C3D2
m_sBox(0, 86) = &HEF1C1847
m_sBox(1, 86) = &H3215D908
m_sBox(2, 86) = &HDD433B37
m_sBox(3, 86) = &H24C2BA16
m_sBox(0, 87) = &H12A14D43
m_sBox(1, 87) = &H2A65C451



m_sBox(2, 87) = &1150940002
m_sBox(3, 87) = &H133AF4DD
m_sBox(0, 88) = &H71DFF89E
m_sBox(1, 88) = &H10314E55
m_sBox(2, 88) = &H81AC77D6
m_sBox(3, 88) = &H5F11199B
m_sBox(0, 89) = &H43556F1
m_sBox(1, 89) = &H1D7A3C76B
m_sBox(2, 89) = &H13C11183B
m_sBox(3, 89) = &H15924A509
m_sBox(0, 90) = &HF28FF6ED
m_sBox(1, 90) = &H97FF1FBFA
m_sBox(2, 90) = &H9EBA1BF2C
m_sBox(3, 90) = &H1E153C6E
m_sBox(0, 91) = &H86E34570
m_sBox(1, 91) = &H1EAE96FB1
m_sBox(2, 91) = &H860E5E0A
m_sBox(3, 91) = &H15A3E2AB3
m_sBox(0, 92) = &H771FE71C
m_sBox(1, 92) = &H4E3D06FA
m_sBox(2, 92) = &H2965DCB9
m_sBox(3, 92) = &H99E71D0F
m_sBox(0, 93) = &H803E891D6
m_sBox(1, 93) = &H5266C825
m_sBox(2, 93) = &H2E4CC978
m_sBox(3, 93) = &H9C10B36A
m_sBox(0, 94) = &H1C6150EBA
m_sBox(1, 94) = &H94E2EA78
m_sBox(2, 94) = &HA5FC3C53
m_sBox(3, 94) = &H1E0A2DF4
m_sBox(0, 95) = &H1F2F74EA7
m_sBox(1, 95) = &H361D2B3D
m_sBox(2, 95) = &H11939260F
m_sBox(3, 95) = &H19C27960
m_sBox(0, 96) = &H5223A708
m_sBox(1, 96) = &HF71312B6
m_sBox(2, 96) = &H1EBADFE6E

m_sBox(3, 96) = &HEAC31F66
m_sBox(0, 97) = &HE3BC4595
m_sBox(1, 97) = &HA67BC883
m_sBox(2, 97) = &HB17F37D1
m_sBox(3, 97) = &H18CFF28
m_sBox(0, 98) = &HC332DDFF
m_sBox(1, 98) = &H1BE6C5AA5
m_sBox(2, 98) = &H65582185
m_sBox(3, 98) = &H68AB9802
m_sBox(0, 99) = &HEFECA50F
m_sBox(1, 99) = &HDB2F953B
m_sBox(2, 99) = &H2AEF7DAD
m_sBox(3, 99) = &H5B6E2F84
m_sBox(0, 100) = &H1521B628
m_sBox(1, 100) = &H29076170
m_sBox(2, 100) = &HECDD4775
m_sBox(3, 100) = &H619F1510
m_sBox(0, 101) = &H13CCA830
m_sBox(1, 101) = &HEB61BD96
m_sBox(2, 101) = &H334FE1E
m_sBox(3, 101) = &HAA0363CF
m_sBox(0, 102) = &HB5735C90
m_sBox(1, 102) = &H4C70A239
m_sBox(2, 102) = &HD59E9E0B
m_sBox(3, 102) = &HCBAADE14
m_sBox(0, 103) = &HEECC86BC
m_sBox(1, 103) = &H60622CA7
m_sBox(2, 103) = &H9CAB5CAB
m_sBox(3, 103) = &HB2F3846E
m_sBox(0, 104) = &H648B1EAF
m_sBox(1, 104) = &H19BDF0CA
m_sBox(2, 104) = &HA02369B9
m_sBox(3, 104) = &H655ABB50
m_sBox(0, 105) = &H40685A32
m_sBox(1, 105) = &H3C2AB4B3
m_sBox(2, 105) = &H319E19D5
m_sBox(3, 105) = &HC021B8F7

m_sBox(0, 106) = &119B540B19
m_sBox(1, 106) = &11875FA099
m_sBox(2, 106) = &1195F7997E
m_sBox(3, 106) = &H623D7DA8
m_sBox(0, 107) = &HF837889A
m_sBox(1, 107) = &H97E321D77
m_sBox(2, 107) = &H11ED935F
m_sBox(3, 107) = &1116681281
m_sBox(0, 108) = &HE358829
m_sBox(1, 108) = &HC7E61FD6
m_sBox(2, 108) = &H96D1DFA1
m_sBox(3, 108) = &117858BA99
m_sBox(0, 109) = &H57F584A5
m_sBox(1, 109) = &H1B227263
m_sBox(2, 109) = &H9B83C31F
m_sBox(3, 109) = &H1AC24696
m_sBox(0, 110) = &HCDB30AEB
m_sBox(1, 110) = &H532E3054
m_sBox(2, 110) = &118FD948E4
m_sBox(3, 110) = &116DBC3128
m_sBox(0, 111) = &H58EBF2EF
m_sBox(1, 111) = &H34C6FFEA
m_sBox(2, 111) = &HFE28ED61
m_sBox(3, 111) = &HEE7C3C73
m_sBox(0, 112) = &115D4A14D9
m_sBox(1, 112) = &HE864B7E3
m_sBox(2, 112) = &1142105D14
m_sBox(3, 112) = &H203E13F0
m_sBox(0, 113) = &H45FEF2B6
m_sBox(1, 113) = &HA3AAA13EA
m_sBox(2, 113) = &HDB6C4F15
m_sBox(3, 113) = &HFACB4FD0
m_sBox(0, 114) = &11C742F442
m_sBox(1, 114) = &HEF6ABBB5
m_sBox(2, 114) = &11654F3B1D
m_sBox(3, 114) = &H41CD2105
m_sBox(0, 115) = &HD81E799E

m_sBox(1, 115) = &H86854DC7
m_sBox(2, 115) = &HE44B476A
m_sBox(3, 115) = &H3D816250
m_sBox(0, 116) = &HCF62A1F2
m_sBox(1, 116) = &H5B8D2646
m_sBox(2, 116) = &HFC8883A0
m_sBox(3, 116) = &I1C1C7B6A3
m_sBox(0, 117) = &H7F1524C3
m_sBox(1, 117) = &H69CB7492
m_sBox(2, 117) = &H47848A0B
m_sBox(3, 117) = &H5692B285
m_sBox(0, 118) = &H95BB1F00
m_sBox(1, 118) = &HAD19489D
m_sBox(2, 118) = &f11462B174
m_sBox(3, 118) = &H23820E00
m_sBox(0, 119) = &H58428D2A
m_sBox(1, 119) = &I1C55F5EA
m_sBox(2, 119) = &H1DADF43E
m_sBox(3, 119) = &H233F7061
m_sBox(0, 120) = &H3372F092
m_sBox(1, 120) = &H8D937E41
m_sBox(2, 120) = &I1D65FECF1
m_sBox(3, 120) = &H6C223BDB
m_sBox(0, 121) = &H7CDE3759
m_sBox(1, 121) = &HCBEE7460
m_sBox(2, 121) = &I14085F2A7
m_sBox(3, 121) = &HCF77326E
m_sBox(0, 122) = &HA6078084
m_sBox(1, 122) = &H19F8509E
m_sBox(2, 122) = &HE8EFD855
m_sBox(3, 122) = &I161D99735
m_sBox(0, 123) = &I1A969A7AA
m_sBox(1, 123) = &HC50C06C2
m_sBox(2, 123) = &H5A04ABFC
m_sBox(3, 123) = &H800BCADC
m_sBox(0, 124) = &H9E447A2E
m_sBox(1, 124) = &I1C3453484

m_sBox(2, 124) = &HFDD56705
m_sBox(3, 124) = &HE1E9EC9
m_sBox(0, 125) = &HDB73DBD3
m_sBox(1, 125) = &I1105588CD
m_sBox(2, 125) = &H675FDA79
m_sBox(3, 125) = &HE3674340
m_sBox(0, 126) = &HC5C43465
m_sBox(1, 126) = &H713E38D8
m_sBox(2, 126) = &H3D28F89E
m_sBox(3, 126) = &HF16DFF20
m_sBox(0, 127) = &H153E21E7
m_sBox(1, 127) = &H8FB03D4A
m_sBox(2, 127) = &IE6E39F2B
m_sBox(3, 127) = &HDB83ADI7
m_sBox(0, 128) = &IE93D5A68
m_sBox(1, 128) = &H948140F7
m_sBox(2, 128) = &HF64C261C
m_sBox(3, 128) = &H94692934
m_sBox(0, 129) = &I1411520F7
m_sBox(1, 129) = &H7602D4F7
m_sBox(2, 129) = &I1B3CF46B2E
m_sBox(3, 129) = &HD4A20068
m_sBox(0, 130) = &IID4082471
m_sBox(1, 130) = &H3320F46A
m_sBox(2, 130) = &H43B7D4B7
m_sBox(3, 130) = &H500061AF
m_sBox(0, 131) = &I1E39F62E
m_sBox(1, 131) = &I97244546
m_sBox(2, 131) = &I14214F74
m_sBox(3, 131) = &HBF8B8840
m_sBox(0, 132) = &H4D95FC1D
m_sBox(1, 132) = &H96B591AF
m_sBox(2, 132) = &H70F4DDD3
m_sBox(3, 132) = &H66A02F45
m_sBox(0, 133) = &I1BFBC09EC
m_sBox(1, 133) = &H3BD9785
m_sBox(2, 133) = &H7FAC6DD0

m_sBox(3, 133) = &H31CB8504
m_sBox(0, 134) = &H96EB27B3
m_sBox(1, 134) = &H55FD3941
m_sBox(2, 134) = &HDA2547E6
m_sBox(3, 134) = &HABCA0A9A
m_sBox(0, 135) = &H28507825
m_sBox(1, 135) = &H530429F4
m_sBox(2, 135) = &HA2C86DA
m_sBox(3, 135) = &H1E9B66DFB
m_sBox(0, 136) = &H68DC1462
m_sBox(1, 136) = &HD7486900
m_sBox(2, 136) = &H680EC0A4
m_sBox(3, 136) = &H27A18DEE
m_sBox(0, 137) = &H4F3F7EA2
m_sBox(1, 137) = &H1E887AD8C
m_sBox(2, 137) = &H1B58CE006
m_sBox(3, 137) = &H17AF4D6B6
m_sBox(0, 138) = &HAAACE1E7C
m_sBox(1, 138) = &HD3375FEC
m_sBox(2, 138) = &HCE78A399
m_sBox(3, 138) = &H406B2A42
m_sBox(0, 139) = &H20FE9E35
m_sBox(1, 139) = &H1D9F385B9
m_sBox(2, 139) = &H1EE39D7AB
m_sBox(3, 139) = &H3B124E8B
m_sBox(0, 140) = &H11DC9FAF7
m_sBox(1, 140) = &H4B6D1856
m_sBox(0, 150) = &H257B7834
m_sBox(1, 150) = &H602A9C60
m_sBox(2, 150) = &HDF8E8A3
m_sBox(3, 150) = &H11F636C1B
m_sBox(0, 151) = &H1E12B4C2
m_sBox(1, 151) = &H2F1329E
m_sBox(2, 151) = &HAF664FD1
m_sBox(3, 151) = &HCAD18115
m_sBox(0, 152) = &H6B2395E0
m_sBox(1, 152) = &H333E92E1

m_sBox(2, 152) = &H3B240B62
m_sBox(3, 152) = &H1EEBEB922
m_sBox(0, 153) = &H185B2A20E
m_sBox(1, 153) = &H1F6BA0D99
m_sBox(2, 153) = &HD1E720C8C
m_sBox(3, 153) = &H21DA2F728
m_sBox(0, 154) = &HD0127845
m_sBox(1, 154) = &H95B794FD
m_sBox(2, 154) = &H647D0862
m_sBox(3, 154) = &HE7CCF5F0
m_sBox(0, 155) = &H5449A36F
m_sBox(1, 155) = &H877D48FA
m_sBox(2, 155) = &H1C39DFD27
m_sBox(3, 155) = &HF33E8D1E
m_sBox(0, 156) = &H1A476341
m_sBox(1, 156) = &H992EFF74
m_sBox(2, 156) = &H3A6F6EAB
m_sBox(3, 156) = &HF4F8FD37
m_sBox(0, 157) = &HA812DC60
m_sBox(1, 157) = &HA1EBDDF8
m_sBox(2, 157) = &H1991BE14C
m_sBox(3, 157) = &H1DB6E6B0D
m_sBox(0, 158) = &H1C67B5510
m_sBox(1, 158) = &H6D672C37
m_sBox(2, 158) = &H2765D43B
m_sBox(3, 158) = &HDC1D0E804
m_sBox(0, 159) = &HF1290DC7
m_sBox(1, 159) = &H1CC00FFA3
m_sBox(2, 159) = &HB5390F92
m_sBox(3, 159) = &H1690FED0B
m_sBox(0, 160) = &H667B9FFB
m_sBox(1, 160) = &H1CEDB7D9C
m_sBox(2, 160) = &HA091CF0B
m_sBox(3, 160) = &HD9155EA3
m_sBox(0, 161) = &HBB132F88
m_sBox(1, 161) = &H1515BAD24
m_sBox(2, 161) = &H7B9479BF

m_sBox(3, 161) = &11763BD6EB
m_sBox(0, 162) = &H37392fB3
m_sBox(1, 162) = &HCC115979
m_sBox(2, 162) = &H8026E297
m_sBox(3, 162) = &HF42E312D
m_sBox(0, 163) = &H6842ADA7
m_sBox(1, 163) = &HC66A2B3B
m_sBox(2, 163) = &1112754CCC
m_sBox(3, 163) = &H782EF11C
m_sBox(0, 164) = &H6A124237
m_sBox(1, 164) = &HB79251E7
m_sBox(2, 164) = &H6A1B3BE6
m_sBox(3, 164) = &H4BFB6350
m_sBox(0, 165) = &H1A6B1018
m_sBox(1, 165) = &H11CAEDFA
m_sBox(2, 165) = &113D25BDD8
m_sBox(3, 165) = &HE2E1C3C9
m_sBox(0, 166) = &1144421659
m_sBox(1, 166) = &HA121386
m_sBox(2, 166) = &11D90CEC6E
m_sBox(3, 166) = &HD5A1B1A2A
m_sBox(0, 167) = &H64AF674F
m_sBox(1, 167) = &HDA86A85F
m_sBox(2, 167) = &HBEBFE988
m_sBox(3, 167) = &1164E4C3FE
m_sBox(0, 168) = &119DBC8057
m_sBox(3, 186) = &117C927C24
m_sBox(0, 187) = &H86E3725F
m_sBox(1, 187) = &H724D9DB9
m_sBox(2, 187) = &H1AC15B134
m_sBox(3, 187) = &11D39EB8FC
m_sBox(0, 188) = &11ED545578
m_sBox(1, 188) = &1181CA5B5
m_sBox(2, 188) = &HD83D7CD3
m_sBox(3, 188) = &H4DA1D0FC4
m_sBox(0, 189) = &H1E50EF5E
m_sBox(1, 189) = &11B161E6F8

```

m_sBox(2, 189) = &11A28514D9
m_sBox(3, 189) = &116C51133C
m_sBox(0, 190) = &116FD5C7E7
m_sBox(1, 190) = &H56E14FC4
m_sBox(2, 190) = &H362ABFCE
m_sBox(3, 190) = &HDDC6C837
m_sBox(0, 191) = &HD79A3234
m_sBox(1, 191) = &1192638212
m_sBox(2, 191) = &H670EFA8E
m_sBox(3, 191) = &11406000E0
m_sBox(0, 192) = &H3A39CE37
m_sBox(1, 192) = &11D3FAF5CF
m_sBox(2, 192) = &HABC27737
m_sBox(3, 192) = &H5AC52D1B
m_sBox(0, 193) = &H5CB06791
m_sBox(1, 193) = &H4FA33742
m_sBox(2, 193) = &HD3822740
m_sBox(3, 193) = &H99BC9BBE
m_sBox(0, 194) = &11D5118E9D
m_sBox(1, 194) = &11BF0F7315
m_sBox(2, 194) = &11D62D1C7E
m_sBox(3, 194) = &HC700C47B
m_sBox(0, 195) = &HB78C1B6B
m_sBox(1, 195) = &H21A19045
m_sBox(2, 195) = &11B26EB1BE
m_sBox(3, 195) = &H6A366EB4

```

End Sub

Public Function EncryptFile(InFile As String, OutFile As String, Overwrite As Boolean, Optional Key As String, Optional OutputIn64 As Boolean) As Boolean

Class_Initialize

If FileExist(InFile) = False Then

EncryptFile = False

Exit Function

End If

If FileExist(OutFile) = True And Overwrite = False Then

EncryptFile = False

```

Exit Function
End If
Dim Buffer() As Byte, FileO As Integer
FileO = FreeFile
Open InFile For Binary As #FileO
ReDim Buffer(0 To LOF(InFileO) - 1)
Get #FileO, , Buffer()
Close #FileO
Call EncryptByte(Buffer(), Key)
If FileExist(OutFile) = True Then Kill OutFile
FileO = FreeFile
Open OutFile For Binary As #FileO
If OutputIn64 = True Then
Put #FileO, , EncodeArray64(Buffer())
Else
Put #FileO, , Buffer()
End If
Close #FileO
EncryptFile = True
Erase Buffer(): Key = "": InFile = "": OutFile = ""
Exit Function
ErrorHandler:
Erase Buffer(): Key = "": InFile = "": OutFile = ""
EncryptFile = False
End Function
Public Function DecryptFile(InFile As String, OutFile As String, Overwrite As Boolean, Optional Key As String, Optional
IsFileIn64 As Boolean) As Boolean
On Error GoTo ErrorHandler
If FileExist(InFile) = False Then
DecryptFile = False
Exit Function
End If
If FileExist(OutFile) = True And Overwrite = False Then
DecryptFile = False
Exit Function
End If
Dim Buffer() As Byte, FileO As Integer

```



```

FileO = FreeFile
Open InFile For Binary As #FileO
    ReDim Buffer(0 To LOF(FileO) - 1)
    Get #FileO, , Buffer()
Close #FileO
If IsFileIn64 = True Then Buffer() = DecodeArray64(StrConv(Buffer(), vbUnicode))
Call DecryptByte(Buffer(), Key)
If FileExist(OutFile) = True Then Kill OutFile
Open OutFile For Binary As #FileO
    Put #FileO, , Buffer()
Close #FileO
DecryptFile = True
Erase Buffer(): Key = "": InFile = "": OutFile = ""
Exit Function
ErrorHandler:
    Erase Buffer(): Key = "": InFile = "": OutFile = ""
    DecryptFile = False
End Function
Private Function FileExist(FilePath As String) As Boolean
    On Error GoTo ErrorHandler
    Call FileLen(FilePath)
    FileExist = True
Exit Function
ErrorHandler:
    FileExist = False
End Function

```

ClsPaging (Class Modules ส่วนติดต่อของ Finger Scan กับ Microsoft Visual Basic Studio 6.0)

```

Option Explicit
Private m_PAGE_CURRENT As Long 'Store current page
Private m_PAGE_PREV As Long 'Store the previous page
Private m_PAGE_NEXT As Long 'Store the next page
Private m_PAGE_TOTAL As Long 'Store the total page count
Private m_Recordset As Recordset 'Reference recordset where we will perform the paging
Private m_PageBy As Long 'Store the how many record display per page

```

```

Private m_PageStart      As Long 'Store the starting record position
Private m_PageEnd        As Long 'Store the ending record position
Private PageInformation  As String 'Store the page information

'Return the current position
Public Property Get CurrentPosition() As Long

    MsgBox m_PAGE_CURRENT

    CurrentPosition = m_PAGE_CURRENT

End Property

'Set the current position
Public Property Let CurrentPosition(ByVal whichPage As Long)

    m_PAGE_CURRENT = whichPage

    PageRecordset

End Property

'Proccedure used to refresh pages
Public Sub Refresh()

    PageRecordset

End Sub

'Start the class and get all the input needed
Public Sub Start(ByRef srcRecordset, ByVal srcPageBy As Long)

Set m_Recordset = srcRecordset

    m_Recordset.CacheSize = srcPageBy

    m_PageBy = srcPageBy

End Sub

Private Sub Class_Terminate()

    'Clear all variables

    m_PAGE_CURRENT = 0

    m_PAGE_NEXT = 0

    m_PAGE_PREV = 0

    m_PAGE_TOTAL = 0

    m_PageBy = 0

    m_PageStart = 0

    m_PageEnd = 0

    PageInformation = vbNullString

    Set m_Recordset = Nothing

End Sub

'Procedure for paging the recordset
Private Sub PageRecordset()

```

```

If m_Recordset.RecordCount < 1 Then
    m_PAGE_NEXT = 0
    m_PAGE_PREV = 0
    m_PAGE_TOTAL = 1
    m_PageStart = 0
    m_PageEnd = 0
    PageInformation = "0 - 0 of 0"
    Exit Sub
End If

'Initialize the paging variables
m_PAGE_TOTAL = Fix(m_Recordset.RecordCount / m_PageBy)
If InStr(1, (m_Recordset.RecordCount / m_PageBy), ".") > 0 Then
    m_PAGE_TOTAL = m_PAGE_TOTAL + 1
End If
If m_PAGE_TOTAL = 0 Then m_PAGE_TOTAL = m_PAGE_TOTAL + 1
'Page the records
m_PAGE_NEXT = m_PAGE_CURRENT + 1
m_PAGE_PREV = m_PAGE_CURRENT - 1
If m_PAGE_TOTAL = 1 Then
    m_PageStart = 1
    m_PageEnd = m_Recordset.RecordCount
Elseif m_PAGE_CURRENT = 1 And m_PAGE_TOTAL > 1 Then
    m_PageStart = 1
    m_PageEnd = m_PageBy
Elseif m_PAGE_CURRENT = m_PAGE_TOTAL And m_PAGE_CURRENT > 1 Then
    m_PageStart = ((m_PAGE_CURRENT - 1) * m_PageBy) + 1
    m_PageEnd = m_Recordset.RecordCount
Else
    m_PageStart = ((m_PAGE_CURRENT - 1) * m_PageBy) + 1
    m_PageEnd = (m_PAGE_NEXT - 1) * m_PageBy
End If

'Set the page information
PageInformation = m_PageStart & " - " & m_PageEnd & " of " & m_Recordset.RecordCount
End Sub

'Return the current page
Public Function PAGE_CURRENT() As Long
    PAGE_CURRENT = m_PAGE_CURRENT

```

```

End Function

'Return the previous page
Public Function PAGE_PREVIOUS() As Long
    PAGE_PREVIOUS = m_PAGE_PREV
End Function

'Return the next page
Public Function PAGE_NEXT() As Long
    PAGE_NEXT = m_PAGE_NEXT
End Function

'Return the page total
Public Function PAGE_TOTAL() As Long
    PAGE_TOTAL = m_PAGE_TOTAL
End Function

'Return the starting record position
Public Function PageStart() As Long
    PageStart = m_PageStart
End Function

'Return the ending record position
Public Function PageEnd() As Long
    PageEnd = m_PageEnd
End Function

'Return the page information
Public Function PageInfo() As String
    PageInfo = PageInformation
End Function

```

ClsSQLSelectParser (Class Modulesการเรียกใช้งานFingerprint)

```

Option Explicit

'Variables for SQL statement
Private m_Fields      As String
Private m_Tables      As String
Private m_wCondition  As String
Private m_GroupOrder  As String
Private m_hCondition  As String
Private m_SortOrder   As String

'Variables for storing saved statement

```

```

Private def_Fields      As String
Private def_Tables     As String
Private def_wCondition  As String
Private def_GroupOrder As String
Private def_hCondition  As String
Private def_SortOrder  As String

'Return the fields
Public Property Get Fields() As String
    Fields = m_Fields
End Property

'Set the fields
Public Property Let Fields(ByVal srcFields As String)
    m_Fields = srcFields
End Property

'Return the tables
Public Property Get Tables() As String
    Tables = m_Tables
End Property

'Set the tables
Public Property Let Tables(ByVal srcTables As String)
    m_Tables = srcTables
End Property

'Return the where condition
Public Property Get wCondition() As String
    wCondition = m_wCondition
End Property

'Set the where condition
Public Property Let wCondition(ByVal srcwCondition As String)
    m_wCondition = srcwCondition
End Property

'Return the group order
Public Property Get GroupOrder() As String
    GroupOrder = m_GroupOrder
End Property

'Set the group order
Public Property Let GroupOrder(ByVal srcGroupOrder As String)
    m_GroupOrder = srcGroupOrder

```

```

End Property

'Return the having condition
Public Property Get hCondition() As String
    hCondition = m_hCondition
End Property

'Set the having condition
Public Property Let hCondition(ByVal srcCondition As String)
    m_hCondition = srcCondition
End Property

'Return the sort order
Public Property Get SortOrder() As String
    SortOrder = m_SortOrder
End Property

'Set the sort order
Public Property Let SortOrder(ByVal srcSortOrder As String)
    m_SortOrder = srcSortOrder
End Property

'Return the SQL statement
Public Property Get SQLStatement() As String
    'Required statement
    SQLStatement = "SELECT " & m_Fields & " FROM " & m_Tables
    'Optional statement
    If m_wCondition <> "" Then SQLStatement = SQLStatement & " WHERE " & m_wCondition
    If m_GroupOrder <> "" Then SQLStatement = SQLStatement & " GROUP BY " & m_GroupOrder
    If m_hCondition <> "" Then SQLStatement = SQLStatement & " HAVING " & m_hCondition
    If m_SortOrder <> "" Then SQLStatement = SQLStatement & " ORDER BY " & m_SortOrder
End Property

'Save the statement
Public Sub SaveStatement()
    def_Fields = m_Fields
    def_Tables = m_Tables
    def_wCondition = m_wCondition
    def_GroupOrder = m_GroupOrder
    def_hCondition = m_hCondition
    def_SortOrder = m_SortOrder
End Sub

'Restore the statement

```

```

Public Sub RestoreStatement()
    m_Fields = def_Fields
    m_Tables = def_Tables
    m_wCondition = def_wCondition
    m_GroupOrder = def_GroupOrder
    m_hCondition = def_hCondition
    m_SortOrder = def_SortOrder

```

```
End Sub
```

```
Private Sub Class_Terminate()
```

```
    'Clear used variables
```

```
    m_Fields = vbNullString
```

```
    m_Tables = vbNullString
```

```
    m_wCondition = vbNullString
```

```
    m_GroupOrder = vbNullString
```

```
    m_hCondition = vbNullString
```

```
    m_SortOrder = vbNullString
```

```
    def_Fields = vbNullString
```

```
    def_Tables = vbNullString
```

```
    def_wCondition = vbNullString
```

```
    def_GroupOrder = vbNullString
```

```
    def_hCondition = vbNullString
```

```
    def_SortOrder = vbNullString
```

```
End Sub
```

Code Microcontroller Arduino

```

void setup()
{
    start serial port at 9600 bps and wait for port to open
    Serial.begin(9600);
    pinMode(2, OUTPUT);    digital sensor is on digital pin 2
    send a byte to establish contact until receiver responds
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);

```

```

pinMode(6, OUTPUT);
pinMode(7, OUTPUT);
pinMode(8, OUTPUT);
digitalWrite(2, LOW);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
digitalWrite(5, LOW);
digitalWrite(6, LOW);
digitalWrite(7, LOW);
digitalWrite(8, HIGH);
}
void loop()
{
  if we get a valid byte, read analog ins
  if (Serial.available() > 0) {
    char inChar = (char)Serial.read();
    if (inChar == 97) {a
      Serial.print(inChar);
      digitalWrite(2, HIGH);           //turn the LED on (HIGH is the voltage level)
      digitalWrite(8, LOW);           //turn the Buzzer off (Low is the voltage
level)
      delay(5000);                     //wait 5 sec
      digitalWrite(2, LOW);             //turn the LED off (LOW is the voltage level)
      digitalWrite(8, HIGH);           //turn the Buzzer on (High is the voltage
level)
    }
    else if (inChar == 98) {b
      Serial.print(inChar);
      digitalWrite(3, HIGH);           //turn the LED on (HIGH is the voltage level)
      digitalWrite(8, LOW);
      delay(5000);                     //wait 5 sec
      digitalWrite(3, LOW);            //turn the LED off (LOW is the voltage level)

```



```

digitalWrite(8, HIGH);
}
else if (inChar == 99) {c
  Serial.print(inChar);
  digitalWrite(4, HIGH);           //turn the LED on (HIGH is the voltage level)
  digitalWrite(8, LOW);
  delay(5000);                     //wait 5 sec
  digitalWrite(4, LOW);           //turn the LED off (LOW is the voltage level)
  digitalWrite(8, HIGH);
}
else if (inChar == 100) {d
  Serial.print(inChar);
  digitalWrite(5, HIGH);           //turn the LED on (HIGH is the voltage level)
  digitalWrite(8, LOW);
  delay(5000);                     //wait 5 sec
  digitalWrite(5, LOW);           //turn the LED off (LOW is the voltage level)
  digitalWrite(8, HIGH);
}
else if (inChar == 101) {e
  Serial.print(inChar);
  digitalWrite(6, HIGH);           //turn the LED on (HIGH is the voltage level)
  digitalWrite(8, LOW);
  delay(5000);                     //wait 5 sec
  digitalWrite(6, LOW);           //turn the LED off (LOW is the voltage level)
  digitalWrite(8, HIGH);
}
else if (inChar == 102) {f
  Serial.print(inChar);
  digitalWrite(7, HIGH);           //turn the LED on (HIGH is the voltage level)
  digitalWrite(8, LOW);
  delay(5000);                     //wait 5 sec
  digitalWrite(7, LOW);           //turn the LED off (LOW is the voltage level)

```

```
digitalWrite(8, HIGH);  
}  
else  
{  
}  
}  
}
```





LM78XX Series Voltage Regulators

Check for Samples: LM7805C, LM7812C, LM7815C

FEATURES

- Output Current in Excess of 1A
- Internal Thermal Overload Protection
- No External Components Required
- Output Transistor Safe Area Protection
- Internal Short Circuit Current Limit
- Available in the Aluminum TO-3 Package

DESCRIPTION

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

VOLTAGE RANGE

- LM7805C: 5V
- LM7812C: 12V
- LM7815C: 15V

Connection Diagrams

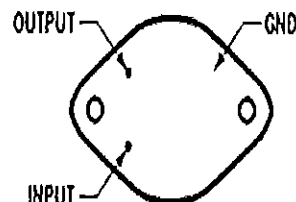


Figure 1. Metal Can Package
TO-3
Aluminum
Bottom View

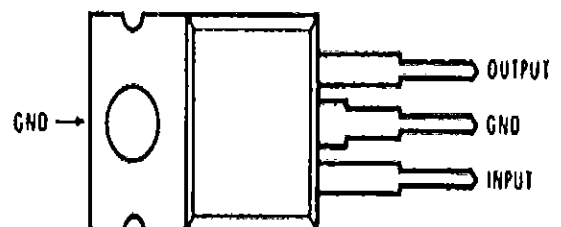
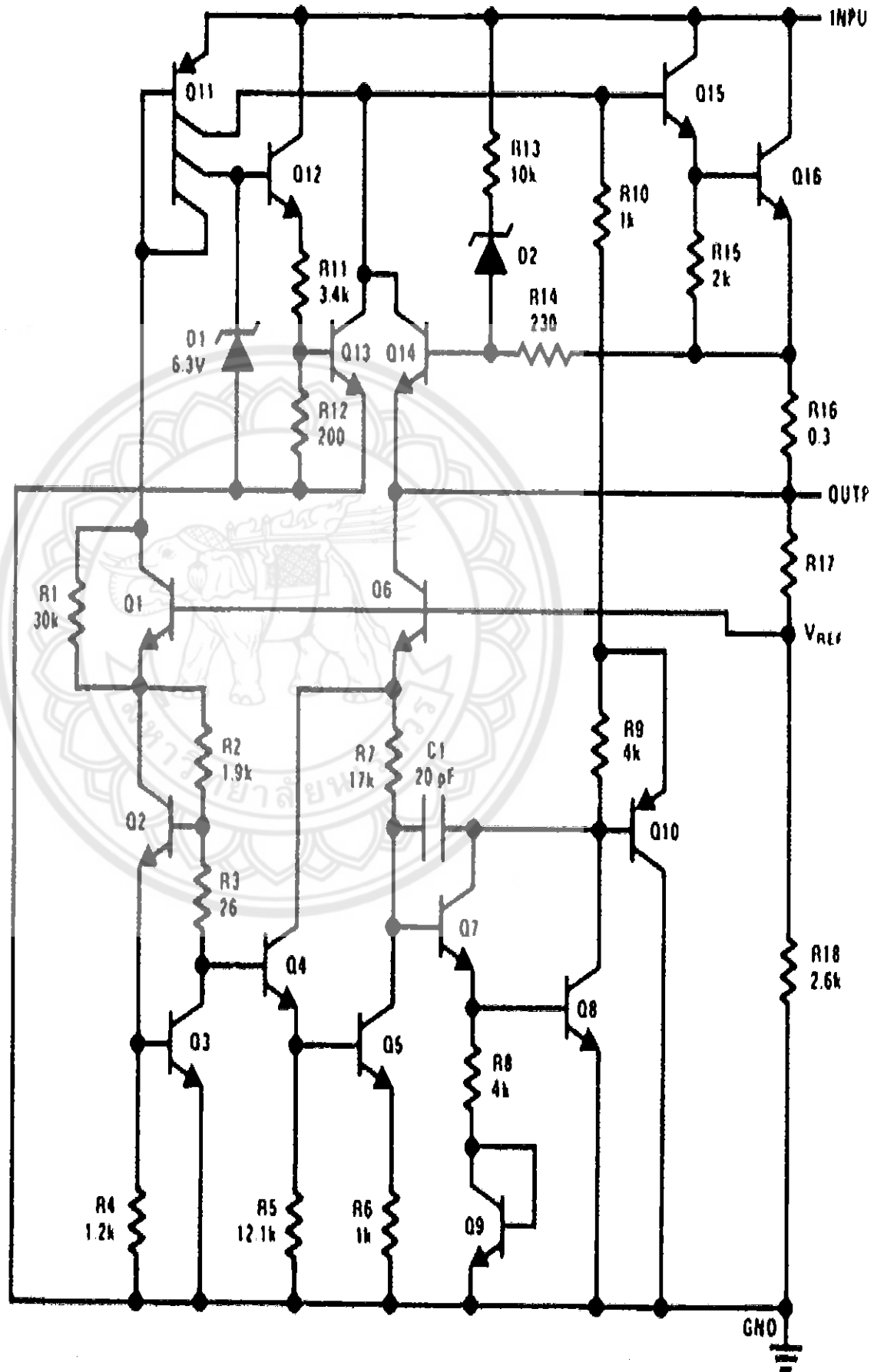


Figure 2. Plastic Package
TO-220 (NDE)
Top View
See Package Number NDE0003B

SCHEMATIC DIAGRAM

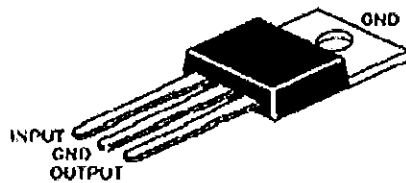


3-TERMINAL POSITIVE VOLTAGE REGULATOR

LM7812

TO-220

Plastic Package



The Voltages Available allow these Regulators to be used in Logic Systems, Instrumentation, Hi-Fi Audio Circuits and other Solid State Electronic Equipment

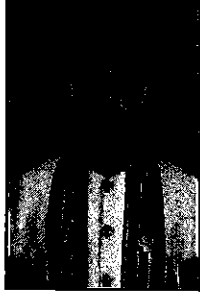
ABSOLUTE MAXIMUM RATINGS ($T_s=25^\circ\text{C}$)

DESCRIPTION	SYMBOL	VALUE	UNIT
Input Voltage	V_{IN}	35	V
		40	
Power Dissipation	P_O	15	W
Operating Temperature	$T_{J=0}$	-20 to +80	$^\circ\text{C}$
Storage Temperature Range	T_{stg}	-55 to +150	$^\circ\text{C}$

ELECTRICAL CHARACTERISTICS ($T_s=25^\circ\text{C}$ unless specified otherwise) $V_{IN}=19\text{V}$, $I_o=100\text{mA}$, $T_s=25^\circ\text{C}$

DESCRIPTION	SYMBOL	TEST CONDITION	MIN	TYP	MAX	UNIT
Output Voltage	V_o	$I_o=5\text{mA} \sim 1.5\text{A}$	11.5		12.5	V
		$V_{IN}=15 \sim 27\text{V}$, $P_O 15\text{W}$				
Line Regulation	R_{REGV}	$V_{IN}=14.5 \sim 30\text{V}$			120	mV
Load Regulation	R_{REGI}	$I_o=5\text{mA} \sim 1.5\text{A}$			120	mV
Quiescent Current	I_Q				8.0	mA
Quiescent Current Change	I_Q	$V_{IN}=14.5 \sim 30\text{V}$			1.0	mA
		$I_o=5\text{mA} \sim 1\text{A}$			0.5	mA
Input Voltage	V_{IN}		14.5		30	V
Ripple Rejection Ratio	R_{RR}	$V_{IN}=15 \sim 25\text{V}$, $f=120\text{Hz}$	55			dB
Max Output Current	I_{OM}	$T_s=25^\circ\text{C}$		2.2		A
Output Voltage Drift	V_{IT}	$I_o=5\text{mA}$, $T_s=0 \sim 125^\circ\text{C}$		-0.8		mV/ $^\circ\text{C}$
Output Noise Voltage	V_{NO}	$f=10\text{Hz} \sim 100\text{kHz}$		10		μV
Short Circuit Current Limit	I_{SC}	$T_s=25^\circ\text{C}$		2.0		A

ประวัติผู้ดำเนินโครงการ



ชื่อ นายปวิวรรต วงศ์วรรณ

ภูมิลำเนา 13/407 หมู่ 11 ต.ต้นธงชัย อ.เมืองลำปาง จ.ลำปาง

ประวัติการศึกษา

- จบระดับชั้นมัธยมจากโรงเรียนมัธยมวิทยาลัยลำปาง
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี

สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail: Markz_hiso_pariwat_ee@hotmail.com



ชื่อ นายเทอดศักดิ์ ทองสุข

ภูมิลำเนา 13/151 หมู่ 11 ต.ต้นธงชัย อ.เมืองลำปาง จ.ลำปาง

ประวัติการศึกษา

- จบระดับชั้นมัธยมจากโรงเรียนอัสสัมชัญลำปาง
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี

สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail: terdsakt51@email.nu.ac.th